UNIVERSIDADE DO MINHO
Escola de Engenharia
Departamento de Informática
Licenciatura em Engenharia de Sistemas e Informática

Relatório de Estágio

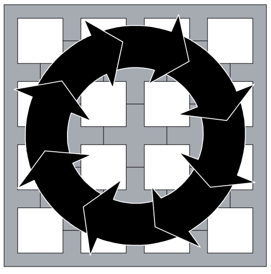
Aplicação da Tecnologia de Agentes a Problemas de Distribuição de Carga em Sistemas Distribuídos

Manuel Alcino Pereira da Cunha

Supervisor:
Engenheiro Orlando Belo

Agosto 1996

Aplicação da Tecnologia de Agentes a Problemas de Distribuição de Carga em Sistemas Distribuídos



Aos meus pais

Agradecimentos

- Ao Engenheiro Orlando Belo, pela motivação, apoio e dedicação ao longo de todo o estágio.
- Ao Luís Paulo, pelas conversas enriquecedoras.
- Ao restante grupo de Engenharia de Computadores, especialmente ao João Luis, ao João Miguel e ao Miguel Monteiro, por me aturarem.
- À minha família, à Sara e restantes amigos, por todo o apoio.

Resumo

A distribuição da carga é uma das características essenciais entre os elementos de processamento disponíveis nos sistemas distribuídos. A eficiência destes sistemas está fortemente correlacionada com a eficiência dos mecanismos de distribuição utilizados. Com a interligação de diversas organizações, através de redes eficientes e economicamente acessiveis, surgiu a possibilidade de rentabilizar os recursos em execesso numa organização através da sua comercialização para outras organizações. No entanto, os mecanismos tradicionais de distribuição não consideram explicitamente os aspectos monetários e de não intrusão nas suas políticas, não estando assim adaptados para serem utilizados nestas condições. Neste relatório apresentam-se os resultados de um trabalho de investigação, relacionado com o estudo da aplicação de técnicas de sistemas multi-agente, nomeadamente técnicas de cooperação por partilha de tarefas, à concepção de uma nova estratégia para o problema da distruibuição de carga. Com a atribuição da gestão dos elementos de processamentos a agentes computacionais, com requisitos próprios e distintos entre si, e com a aplicação de modelos do comportamento humano ao controlo do comportamento cooperativo desses agentes, desenvolveu-se uma estratégia para aplicação em sistemas distribuídos multi-organizacionais, que permite prever questões como a propriedade dos nodos de processamento, o pagamento pela utilização dos recursos desses nodos e a flexibilidade de adaptação às características físicas de cada organização e ao carácter dos seus gestores.

Contents

List of Figures

List of Tables

Chapter 1

Introdução

A preferência do mercado informático por sistemas distribuídos, em detrimento dos grandes sistemas centralizados, é cada vez mais uma realidade. O aumento da oferta de processadores com grande capacidade e baixo custo e o avanço verificado na tecnologia de comunicações, permitiram o desenvolvimento de sistemas distribuídos com um desempenho comparável ao dos grandes sistemas centralizados, mas com um custo substancialmente menor. Analogamente, o mercado, anteriormente muito restrito, de máquinas de elevado desempenho, também foi fortemente abalado pelo aparecimento de máquinas massivamente paralelas, baseadas nos mesmos princípios dos sistemas distribuídos mas com um custo menor para desempenhos equivalentes ou mesmo superiores. No entanto, para utilizar eficazmente todo o potencial dos sistemas distribuídos e paralelos, é necessário proceder à correcta distribuição das tarefas a executar pelos diversos processadores disponíveis, de tal forma que seja maximizada a utilização da capacidade total de processamento. Esta tarefa, normalmente designada por distribuição de carga é vital, se quisermos evitar, por exemplo, que um determinado processador esteja sobrecarregado com trabalho enquanto que um outro possa estar parado. A distribuição de carga só não faria sentido se os processadores tivessem todos a mesma capacidade e se todos fossem submetidos a uma carga idêntica, situação muito pouco provável de acontecer. A sua importância é tal, que o desempenho global de um sistema distribuído está fortemente correlacionado com a eficiência do algoritmo de distribuição de carga utilizado. Por várias vezes foi demonstrado que, até a utilização de algoritmos tão simples como a distribuição aleatória das tarefas pelos processadores, traz grandes benefícios em relação à não realização de qualquer tipo de distribuição [Shivaratri et al., 1992, Zhou, 1988, Eager et al., 1986].

Ultimamente tem surgido o interesse na partilha de recursos entre diversas organizações.

Este interesse surge em consequência de uma tendência existente no meio industrial, onde se verifica o aparecimento de empresas pequenas, dinâmicas e livres de burocracia organizacional, que procuram formar alianças de curto prazo para poderem satisfazer uma procura maior e mais diversificada [Sandholm and Lesser, 1995a]. Estas alianças, quando aplicadas ao domínio da distribuição de carga, podem ser utilizadas para se atingirem economias de escala, nomeadamente, através da aquisição de menos recursos e da utilização dos recursos de outras organizações em momentos de maior necessidade. No entanto, a distribuição de carga neste contexto levanta questões tão importantes como, por exemplo, a necessidade de garantir o direito à propriedade dos recursos ou a possibilidade de pagamento pela utilização dos mesmos. Nenhum dos mecanismos de distribuição de carga desenvolvidos nos últimos anos está preparado para funcionar neste contexto porque não consideram explicitamente os aspectos monetários e de propriedade nas suas decisões.

Se pretenderem ser bem sucedidas no mercado competitivo em que estão inseridas, as entidades de distribuição de carga devem apresentar algumas características que "humanizem" o seu comportamento, tais como a sociabilidade, o oportunismo, a autonomia e a capacidade de reação. Estas são também as características que distinguem os agentes de software dos restantes paradigmas de programação actuais [Jennings and Wooldridge, 1996], e cuja necessidade levou à tentativa de utilizar este paradigma no desenvolvimento de uma nova estratégia de distribuição de carga. Mais concretamente, dada a distribuição espacial dos recursos e a natureza multi-organizacional do sistema, será a utilizada a tecnologia de sistemas multi-agente. O principal problema no desenvolvimento destes sistemas é a definição de uma estratégia efectiva de cooperação entre os agentes. Neste caso, essa definição é ainda mais problemática dado que os agentes podem pertencer a organizações distinctas, cujos interesses são também distinctos ou até mesmos conflituosos.

Estrutura do Relatório

- Capítulo 1 - **Introdução**. Introdução ao trabalho desenvolvido no estágio e apresentação da estrutura do relatório.

- Capítulo 2 - **Distribuição de Carga**. Descrição do problema da distribuição de carga, das razões para a sua existência num sistema multi-organizacional, e dos cuidados que se devem ter na construção de um novo mecanismo de distribuição.

- Capítulo 3 - **Abordagem Convencional**. Apresentação de vários algoritmos de distribuição de carga ditos convencionais, tentando analisar o seu comportamento, identificando os problemas que apresentam e procurando definir critérios de classificação para possibilitar o seu enquadramento e comparação com o novo mecanismo desenvolvido.

- Capítulo 4 - **Abordagem Multi-Agente**. Apresentação da nova estratégia concebida utilizando um sistema multi-agente, indicando em que medida resolve os problemas das abordagens convencionais.

- Capítulo 5 - **Simulador de Estratégias de Distribuição de Carga**. Apresentação de uma plataforma de simulação de estratégias de distribuição de carga, desenvolvida devido à necessidade de comparar a nova estratégia com as convencionais.

- Capítulo 6 - **Conclusões e Trabalho Futuro**. Principais conclusões do trabalho efectuado e ideias para trabalho futuro.

Chapter 2

Distribuição de Carga

A definição do que é um mecanismo de distribuição de carga está praticamente explícita no próprio conceito, embora esta, só por si, não esclareça ninguém acerca desta funcionalidade. Daí que, sempre que se tente defini-lo, se acabe também por descrever as razões que motivam a sua existência. Eventualmente, estes conceitos acabam por se confundir, sendo, normalmente, considerados em conjunto como uma definição alargada do mecanismo de distribuição de carga. No entanto, julga-se que o seu significado é distinto e que merecem uma descrição individualizada, embora a definição seja bastante simples, podendo, desde já, ficar esclarecida: a distribuição de carga consiste na redistribuição das tarefas, a executar em todo o sistema, pelos diversos nodos de processamento por forma a melhorar o desempenho global do mesmo.

Para além da descrição das razões que motivam a existência da distribuição de carga, também se apresentam alguns dos cuidados que se devem ter aquando do desenvolvimento de um mecanismo que implemente esta funcionalidade. A inclusão deste conjunto de cuidados nesta fase do relatório deve-se ao facto de estes deverem ser tomados na concepção de um qualquer mecanismo de distribuição.

2.1 Motivação

Um sistema distribuído real caracteriza-se por nele existirem, simultaneamente, nodos muito sobrecarregados com tarefas e nodos cuja carga é muito pequena ou mesmo nula [Becker, 1995, Shivaratri et al., 1992]. Poderia conseguir-se um grande aumento do desempenho global do sistema se existisse um mecanismo que transferisse algumas das tarefas dos nodos mais sobrecarregados, onde o tempo de execução dessas tarefas é muito elevado, para aqueles com pouca carga, onde as tarefas poderão tirar partido da sua capacidade de processamento, que doutra forma seria desaproveitada [Shivaratri et al., 1992]. Um sistema é heterogéneo quanto à capacidade de processamento de cada um dos seus nodos e quanto à carga a que esses nodos estão sujeitos. Assim, é muito frequente que a situação de desequilíbrio, anteriormente descrita, aconteça frequentemente, havendo necessidade de existir um mecanismo de distribuiçao de carga que a corrija. Várias vezes foi demonstrado [Shivaratri et al., 1992, Zhou, 1988] que, até um mecanismo tão simples como a distribuição aleatória das tarefas pelos diversos nodos existentes, traz grandes melhorias de desempenho em relação à não utilização de qualquer mecanismo de distribuição. Um pouco surpreendentemente, em [Zhou, 1988] concluiu-se que a distribuição de carga traz benefícicios para todos os nodos do sistema, mesmo para aqueles originalmente com pouca carga.

Uma outra razão que motiva a existência deste mecanismo de distribuição passa pelas limitações de recursos dos nodos de processamento. Uma tarefa pode ser submetida para execução num nodo que não possua os recursos necessários para a executar, quer em termos de quantidade, quer em termos de variedade. Esses recursos poderão existir num outro nodo do sistema, não se justificando, neste caso, a não execução da tarefa. Mais uma vez esta situação pode facilmente ser resolvida com um mecanismo de distribuição. Estas limitações nas necessidades de recursos podem originar que algumas tarefas apenas possam ser executadas num determinado nodo da rede, não podendo pois participar no mecanismo de distribuição dada a sua imobilidade. No entanto, em [Zhou, 1988] demonstrou-se que, mesmo quando 50-70% das tarefas são imóveis, a existência de um mecanismo de distribuição continua a ser benéfica.

No sistema subjacente a este estágio, constituído por várias organizações onde se prevê o pagamento pela utilização de recursos, para além dos dois motivos já apontados existem outros inerentes do facto de se tratar de um verdadeiro sistema comercial. Todas as tarefas executadas nos nodos de processamento de uma organização representam um custo para essa organização. Poderão existir tarefas, que por motivos específicos, sejam mais "baratas" de executar em nodos de outras empresas, mesmo sendo possíveis de executar internamente com um desempenho aceitável. No entanto, como o objectivo de maximização do lucro de uma empresa passa pela minimização dos custos de operação, essas tarefas deverão ser alocadas aos nodos que ofereçam os menores custos, mesmo que pertençam a outras empresas.

A situação inversa à descrita também pode ser apontada como motivação para a distribuição de carga. Uma organização tem interesse em executar o maior número de tarefas exteriores, sem prejudicar a execução das suas, pois essa execução será acompanhada do respectivo pagamento, ajudando à maximização dos ganhos da empresa e, consequentemente, à maximização do lucro. Quanto mais não seja, estes ganhos poderão, mais tarde, ser usados para comprar capacidade de processamento a outras organizações. Este motivo também pode ser visto de outra forma: uma organização pode ter interesse em realizar o maior número de tarefas nas suas máquinas como forma de rentabilizar o investimento que foi feito na sua compra.

2.2 Cuidados a se ter na Concepção de Mecanismos de Distribuição

Apresento agora alguns cuidados que se devem ter aquando da concepção de um mecanismo de distribuição. Estes cuidados são genéricos para qualquer mecanismo, estando intrinsecamente relacionados com o próprio conceito de distribuição de carga. Alguns dos cuidados que se apresentam, nomeadamente no tocante à partilha de recursos entre organizações e ao comércio electrónico, apenas se justificam num sistema distribuído com as características apresentadas no capítulo anterior.

Esta enumeração também tem por objectivo fornecer meios para analisar a validade de alguns dos mecanismos já desenvolvidos, a apresentar no capítulo seguinte.

2.2.1 Actualidade da Informação

As decisões tomadas por cada elemento do mecanismo de distribuição de carga deverão ser baseadas em informação actualizada do sistema ou pelo menos consistente com o estado deste. Se esta consistência não for garantida, então o mecanismo deverá estar preparado para trabalhar com informação possivelmente inconsistente ou incompleta e ter mecanismos para prever, usando essa informação, o estado correcto dos componentes do sistema distribuído [Riedl and Richter, 1996].

A não utilização de informação suficientemente actualizada ou a previsão incorrecta do estado dos nodos de processamento podem originar situações indesejáveis, nomeadamente [Jacqmot and Milgrom, 1993]:

- Um nodo de processamento pode, num dado instante de tempo, estar com uma carga baixa e portanto apto a receber tarefas para executar. Os restantes nodos, sabendo esta informação, começam a transferir tarefas para esse nodo. Se o novo estado da carga não for actualizado, pode dar-se o caso de os nodos continuarem essas transferências, pois ainda consideram que o nodo está com carga baixa, acabando por provocar a sua sobrecarga;

- Complementarmente, um nodo pode, num dado momento, estar sobrecarregado, e, se não houver cuidado em ir actualizando a informação sobre esse nodo, pode dar-se o caso de os restantes nodos lhe retirarem continuamente tarefas, acabando este por ficar com pouca, ou mesmo nenhuma, carga.

Ambas as situações são indesejáveis se quisermos maximizar o desempenho global do sistema.

Uma das causas deste problema é o assumir por parte dos autores, e.g. [Eager et al., 1986], que os atrasos existentes na comunicação de informação entre dois nodos são negligenciáveis, ou seja, assumem que a informação que chega a um nodo acerca do estado de outro nodo representa, no instante de chegada, o estado real do nodo que enviou a mensagem. Dadas as características do sistema subjacente, composto por várias organizações que não desejam ver os seus nodos sobrecarregados, assume-se que não se devem negligenciar estes atrasos na entrega de informação, devendo existir mecanismos para lidar com esta situação.

Este cuidado com a actualidade da informação também se deve ter em relação às infra-estruturas de comunicações e não só em relação aos nodos de processamento. Por questões de simplificação apenas se referiu e continurá o caso dos nodos de processamento.

2.2.2 Estabilidade

A instabilidade de um mecanismo de distribuição pode ser vista de duas perspectivas [Shivaratri et al., 1992]:

- Quando a taxa a longo prazo de chegada de tarefas ao sistema é maior que a taxa a que o sistema as executa, a fila de espera do sistema cresce indefinidamente. Neste caso, se na realidade o sistema tiver capacidade para executar todas tarefas, diz-se que o mecanismo de distribuição do sistema é instável;

- Se existe alguma probabilidade do mecanismo de distribuição poder realizar, indefinidamente, transferências inúteis de tarefas, então esse mecanismo é instável.

Para prever o primeiro caso de instabilidade é necessário garantir que a carga imposta ao sistema pelo mecanismo de distribuição seja mínima, quer em termos de necessidade de recursos de processamento, quer em termos de número de mensagens a ocupar a rede de comunicações [Jacqmot and Milgrom, 1993, Ramamritham et al., 1989, Shivaratri et al., 1992, Theimer and Lantz, 1989]. Costuma-se dizer que o sistema deve ocupar os seus recursos a executar as tarefas e não a distribuí-las.

Mais globalmente á que ter o cuidado de o mecanismo ser realmente efectivo, ou seja, dado um conjunto específico de condições, o desempenho do sistema usando um mecanismo de distribuição deve ser superior ao desempenho do mesmo sistema sem qualquer forma de distribuição de carga [Shivaratri et al., 1992]. Existem alguns cuidados essenciais para garantir a eficiência do mecanismo:

- Evitar a transferência de tarefas quando o benefício que se obtem pela sua execução no nodo remoto é inferior ao que se teria se fosse executada localmente [Jacqmot and Milgrom, 1993];

- Encorajar a transferência da tarefa que trará mais benefícios ao ser executada num nodo remoto [Jacqmot and Milgrom, 1993];

- Maximizar o número de tarefas executadas localmente, nomeadamente quando o sistema está sobrecarregado e os atrasos são bastante superiores ao normal [Kremien and Kramer, 1992]. [Zhou, 1988] defende mesmo que a execução remota de uma pequena percentagem das tarefas é suficiente para se conseguir uma distribuição de carga eficiente;

- Evitar que o mecanismo de distribuição reaja a pequenas flutuações na carga do sistema. Se esta precaução não for tomada pode acontecer que os nodos gastam a maior parte dos seus recursos a distribuir as tarefas [Eager et al., 1986].

Para garantir a estabilidade em relação ao segundo ponto, o mecanismo deve evitar, obviamente, transferências inúteis de tarefas. Uma transferência diz-se inútel quando provoca a sobrecarga do nodo destino. Um caso extremo de instabilidade é o conhecido "efeito ping-pong", que acontece quando um nodo A transfere uma tarefa para um nodo B, que, por ficar sobrecarregado, a transfere outra vez para o nodo A [Jacqmot and Milgrom, 1993]. Este tipo de instabilidade é particularmente indesejável em sistemas que executam aplicações em tempo-real, onde as tarefas têm prazos de execução que devem obrigatoriamente ser cumpridos [Ramamritham et al., 1989].

2.2.3 Escalabilidade

É desejável que um mecanismo de distribuição de carga seja escalável ao tamanho do sistema distribuído onde exerce as suas funções. Diz-se, normalmente, que um mecanismo é escalável se o seu desempenho é independente do tamanho do sistema ou pelo menos independente até um certo tamanho de sistema. Esta independência existe se a taxa de execução das tarefas se mantem superior à taxa de chegada das mesmas ao sistema [Kremien, 1995, Özden et al., 1993].

Outra definição mais abrangente, também apresentada em [Kremien, 1995], diz que um mecanismo só é escalável se, para além de ser independente do tamanho do sistema, não é sensível às características físicas dos recursos do sistema, como latência ou largura de banda das comunicações.

Quanto ao tamanho do sistema, [Özden et al., 1993] afirmam que se devem considerar sistemas com vários milhares de nodos, pois organizações de média e grande dimensão possuem, normalmente, essa capacidade de processamento. Como esta afirmação foi feita considerando a distribuição de carga num contexto organizacional, e o sistema subjacente a este estágio prevê a existência de várias organizações, pode-se assumir um tamanho virtualmente ilimitado. Assim, o mecanismo deverá ser infinitamente escalável, ou seja, o seu desempenho deverá ser independente do tamanho do sistema.

2.2.4 Políticas de Partilha

Um mecanismo de distribuição não deve fixar os tipos de políticas de partilha dos nodos de processamento de uma organização. Deve, pelo contrário, fornecer mecanismos que suportem um conjunto alargado de políticas, deixando a especificação das mesmas para os proprietários dos nodos de processamento [Özden et al., 1993].

Um mecanismo também deve ser não intrusivo [Özden et al., 1993]. Um mecanismo não intrusivo é aquele que respeita as políticas de partilha definidas em cada nodo de processamento. Sem esta garantia, nenhuma organização colocará os seus recursos à disposição de outras organizações, mesmo sabendo que a utilização desses recursos é paga.

2.2.5 Comércio Electrónico

O sistema distribuído apresentado no capítulo anterior prevê o pagamento pela utilização dos recursos das organizações. Esta movimentação de capitais obriga à existência de certos cuidados que garantam a segurança das transacções, nomeadamente no respeitante a [Costa et al., 1995, Simão, 1995]:

- Privacidade. O conteúdo das transações efectuadas não pode ser conhecido por terceiros, mesmo que tenham acesso ilícito aos canais de comunicação;

- Autenticação. É necessário garantir que uma mensagem foi criada por quem afirma que o fez, eliminando a possibilidade que uma entidade forje mensagens em nome de outra;

- Integridade. Não deverá ser possível alterar o conteúdo de uma mensagem sem que essa alteração seja detectada;

- Não repudiação. Consiste em tornar tecnicamente e, eventualmente, legalmente demonstrável que uma dada transacção, com um determinado valor, ocorreu de facto entre duas entidades. Serve para garantir que uma entidade não possa alterar à posteriori o conteúdo de uma transacção ou mesmo negar a sua existência.

2.2.6 Outros Cuidados

Para além dos cuidados referidos anteriormente, um mecanismo de distribuição de carga deverá ter em atenção os seguintes aspectos:

- Transparência para os utilizadores. A execução remota deverá ser completamente transparente para as tarefas do sistema e consequentemente para o utilizador que as criou. Para isso, é necessário minimizar os atrasos iniciais na execução das tarefas, e não deve ser preciso alterar o código das tarefas existentes pelo facto de existir um mecanismo de carga no sistema [Theimer and Lantz, 1989];

- Expansibilidade. Deverá ser possível incluir novos nodos de processamento, ou linhas de comunicações, no sistema distribuído sem ser necessário fazer alterações explícitas nos componentes do mecanismo de distribuição de carga. O mesmo é desejável quando se retiram recursos do sistema;

- Tolerância a falhas. Em alguns trabalhos, e.g. [Zhou, 1988], considera-se que a tolerância a falhas não é uma característica essencial de um mecanismo de distribuição, devido ao facto de este não ser um mecanismo vital do sistema, cuja falta pode ser tolerada o tempo suficiente para ser reposto. No entanto, devido à possibilidade de distribuição de tarefas essenciais, que especificam os recursos e prazos de execução de que necessitam, a tolerância a falhas torna-se imprescindível para evitar o risco da não execução dessas tarefas. Assim, o funcionamento do mecanismo de distribuição de carga não deverá ser posto em causa pelo facto de um ou mais nodos de processamento deixarem de funcionar [Theimer and Lantz, 1989]. Esta tolerância também deverá ser extendida, com as devidas reservas, às quebras nas linhas de comunicações que interligam os nodos.

Chapter 3

Abordagem Convencional

Neste capítulo analisa-se algumas abordagens convencionais utilizadas na concepção de mecanismos de distribuição de carga. As abordagens apresentadas, assim como os respectivos mecanismos que as implementam, foram escolhidas tentando formar um leque de abordagens suficientemente representativo do trabalho previamente desenvolvido. Como a compreensão destas abordagens, assim como a sua análise e comparação, é bastante prejudicada se não existir uma terminologia consistente, ou mais genericamente, se não existir um termo comum de comparação [Jacqmot and Milgrom, 1993, Riedl and Richter, 1996], apresentar-se-á uma metodologia que facilita a comparação qualitativa dos mesmos através da sua decomposição em componentes funcionais. Complementarmente, e baseado nesta divisão funcional, apresentar-se-ão alguns parâmetros de classificação de mecanismos de distribuição de carga.

Poderá parecer estranha a inclusão neste capítulo das metodologias de comparação e classificação. Supostamente, a sua validade em relação a qualquer mecanismo de distribuição de carga seria razão suficiente para incluir essas metodologias no capítulo anterior, onde se fala genericamente do problema da distribuição de carga. No entanto, essas metodologias foram concebidas tendo por base os mecanismos desenvolvidos no passado, não havendo garantias que essa validade se mantenha com o surgimento de novas abordagens.

Os mecanismos apresentados não foram desenvolvidos tendo por base um sistema distribuído multi-organizacional. No entanto, a sua análise vai ser feita considerando a sua possível aplicação ao modelo apresentado, pois só assim poderão ser julgados em relação à nova abordagem desenvolvida. Os problemas, ou mais correctamente, as limitações que se apontarão a estes mecanismos deverão, por isso, ser vistas no contexto da sua aplicação ao sistema definido anteriormente e não no seu contexto original.

Para além desta análise qualitativa também se deveria apresentar uma análise quantitativa dos mecanismos apresentados, quer em termos de desempenho, quer em termos de eficiência. Essa análise não é aqui apresentada pois é necessário desenvolver uma plataforma de teste que permita a sua comparação num sistema distribuído multi-organizacional e que, como se verá, difere muito da abordagem convencional à distribuição de carga. O desenvolvimento dessa plataforma de teste é apresentado no capítulo 5.

3.1 Componentes de um Mecanismo de Distribuição de Carga

A maior parte dos trabalhos apresentados sobre distribuição de carga baseiam-se no conhecido modelo das quatro regras ou das quatro políticas. Assim, todos os mecanismos de distribuição devem incluir as seguintes políticas [Shivaratri et al., 1992]:

- **Política de transferência**, que determina quando é que um nodo de processamento está em condições de participar numa transferência de carga, quer como originador dessa carga quer como receptor;
- **Política de selecção**, que determina, caso um nodo seja escolhido pela política de transferência para transferir carga, qual a tarefa que será escolhida;
- **Política de localização**, que é responsável por encontrar um parceiro para a transferência de carga, depois de um nodo ter sido escolhido como originador ou receptor de carga pela política de transferência;
- **Política de informação**, que decide quando é que a informação acerca do estado dos outros nodos vai ser obtida, de quem é que vai ser obtida e que informação é que vai ser obtida.

Existem alguns trabalhos que apresentam decomposições diferentes, mas que na sua maior parte constituem variações deste modelo. Por exemplo, em [Zhou, 1988] inclui-se a política de selecção na política de transferência, considerando que ao procurar uma tarefa para ser transferida, caso esta seja encontrada, estamos automaticamente a considerar um nodo em condições de participar numa transferência. No entanto, considera-se que este agrupar das duas políticas não é muito correcto pois não prevê o facto de um nodo poder estar em condições de participar numa transferência de carga como receptor da mesma. Existem outros trabalhos onde os autores agrupam as políticas acima referida em políticas mais abrangentes:

- em [Jacqmot and Milgrom, 1993] considera-se que a descrição de um mecanismo de distribuição de carga consiste apenas na descrição da forma como a informação e a transferências de tarefas são tratadas. Na realidade, o tratamento da informação corresponde à política de informação acima referida, enquanto que o tratamento das transferências agrupa as outras três políticas, acabando pois por diferir muito pouco do modelo apresentado;

- em [Lüling et al., 1992] decompõe-se os mecanismos da distribuição em duas partes: a parte de decisão e a parte de migração. Na primeira decide-se se um nodo deve ou não manter uma tarefa e na segunda escolhe-se um parceiro para transferir a tarefa. Basicamente a parte de decisão corresponde às políticas de transferência e de selecção, enquanto que a parte de migração corresponde à política de localização. A política de informação está dividida pelas duas partes.

No entanto, existem outras abordagens bastante diferentes na modelação de mecanismos de distribuição de carga. Em [Riedl and Richter, 1996], por exemplo, propõe-se uma decomposição em três modelos: o modelo da carga, o modelo das acções e o modelo da solução. O primeiro descreve a realidade física do sistema, nomeadamente a carga a que o sistema está sujeito e a forma como está distribuída pelo sistema, e os possíveis estados em que esse sistema pode estar. O modelo das acções define quais as acções que cada componente do sistema pode realizar para cada estado do sistema. O modelo da solução determina, dado o estado do sistema definido pelo modelo da carga, e as possíveis acções a tomar definidas pelo modelo das acções, qual a melhor estratégia a seguir para se atingirem os objectivos da distribuição de carga. Este modelo é mais genérico que o modelo das quatro políticas, na medida em que situa o mecanismo de distribuição no contexto do sistema para que foi construído e não limita o tipo de acções de distribuição às transferências de tarefas entre nodos. No entanto, não se concorda com os autores quando dizem que o modelo das quatro políticas é específico para mecanismos de distribuição de carga onde todos os nodos fazem acções de distribuição: nada impede que as políticas possam prever acções diferentes para cada nodo.

Embora o modelo proposto em [Riedl and Richter, 1996] seja mais genérico, optou-se, neste estágio, por comparar os mecanismos tendo por base o modelo das quatro políticas dada a sua melhor compreensão, maior facilidade de utilização e, principalmente, maior aceitação. Outra razão para o detrimento do modelo proposto em [Riedl and Richter, 1996] é o facto de que a maior generalidade que o caracteriza não traz vantagens para a comparação que pretendo efectuar, pois o modelo da carga é igual para todos os mecanismos e as acções de distribuição usadas consistem apenas em transferências de tarefas entre nodos.

3.2 Classificação de Mecanismos de Distribuição de Carga

Nesta secção vou apresentaram-se alguns dos parâmetros usados mais frequentemente na classificação de mecanismos de distribuição. Exceptuando alguns de carácter geral, estes parâmetros estão normalmente relaccionados com as políticas que constituem um mecanismo de distribuição. Esta metodologia de classificação não foi adoptada na sua totalidade por nenhum dos trabalhos consultados, sendo antes uma compilação das diversas metodologias utilizadas nesses trabalhos. O objectivo desta compilação é fornecer uma base de classificação e comparação o mais alargada possível mesmo à custa de alguma redundância nos seus parâmetros.

As primeiras secções apresentadas, dizem respeito aos parâmetros de caracterização global de um mecanismo de distribuição de carga, estando pois reflectidas em todas as políticas do mesmo. Os parâmetros de classificação específicos de cada política serão apresentados em secções com os respectivos nomes.

3.2.1 Estáticos vs Dinâmicos

Muitos dos trabalhos mais antigos concentravam-se na distribuição estática da carga. Nestes trabalhos, as decisões para transferir uma tarefa eram feitas deterministicamente ou probabilisticamente sem levar em consideração o estado corrente do sistema [Zhou, 1988]. Esta abordagem é bastante simples e efectiva quando a carga pode ser suficientemente bem caracterizada à priori, o que quase nunca acontece nos sistemas distribuídos reais, onde as tarefas que executam num nodo tem uma distribuição de chegada aleatória e diferem bastante entre si quanto aos recursos necessários. O seu carácter estático não permite que se adaptem às flutuações de carga no sistema e por isso tem sido progressivamente abandonada como estratégia de balanceamento genérica, continuando, no entanto, a ser muito usada em sistemas paralelos dedicados, como meio de distribuir as tarefas de algumas aplicações cujas necessidades de recursos e de comunicações entre nodos são bem conhecidas.

As abordagens dinâmicas têm o potencial de superar as estáticas devido ao facto de usarem informação sobre o estado do sistema para melhorarem a qualidade das suas decisões [Shivaratri et al., 1992]. Ao não assumirem um pré-conhecimento da carga do sistema, estas abordagens optam por fazer a distribuição dinamicamente conforme as tarefas vão surgindo nos nodos [Zhou, 1988]. Embora os mecanismos dinâmicos imponham uma maior carga sobre o sistema que os estáticos, este aumento é normalmente compensado devido à maior qualidade da distribuição. Naturalmente, também existem algumas formas intermédias entre abordagens completamente estáticas e completamente dinâmicas, como por exemplo, mecanismos estáticos que são inicializados por algum componente dinâmico que reage a mudanças no estado da carga, ou combinações de componentes estáticos e dinâmicos [Riedl and Richter, 1996]. Em consonância com praticamente todos os trabalhos recentes sobre distribuição de carga, apenas se irá considerar neste estudo comparativo abordagens com algum tipo de dinamismo.

3.2.2 Centralização vs Descentralização

Os mecanismos de distribuição de carga diferem no grau de centralização com que são concebidos. Nas abordagens centralizadas, um dos nodos é responsável por todas as actividades de distribuição, desde a recolha de informação à selecção dos nodos candidatos para receber as tarefas. Pelo contrário, nas abordagens completamente distribuídas, todas as tarefas da distribuição de carga estão distribuídas por todos os nodos do sistema. Entre os dois extremos, centralizada e completamente distribuída, existem muitas combinações usadas pelas abordagens de distribuição de carga, como por exemplo:

- a organização hierárquica [Shivaratri et al., 1992], onde os nodos estão agrupados segundo uma árvore, havendo para cada conjunto de nodos um responsável central pela distribuição, que por sua vez também tem um responsável central, e assim sucessivamente;

- a proposta por [Becker, 1995], onde os nodos se encontram agrupados em conjuntos, denominados por *clusters*, cada um com um responsável central pela distribuição, mas onde os diversos responsáveis distribuem a carga entre si de forma distribuída.

As diversas abordagens são apresentadas na figura 3.1, onde as máquinas com componentes de distribuição de carga estão assinaladas com as letras DC e onde as linhas de interligação entre os nodos representam ligações lógicas e não físicas. Aliás, quando se discute este conceito é necessário estabelecer uma distinção clara entre conceitos físicos e lógicos [Riedl and Richter, 1996]. Na maior parte dos casos, os conceitos lógicos coincidem com os físicos. Naturalmente, mecanismos fisicamente centralizados também são logicamente centraliza-

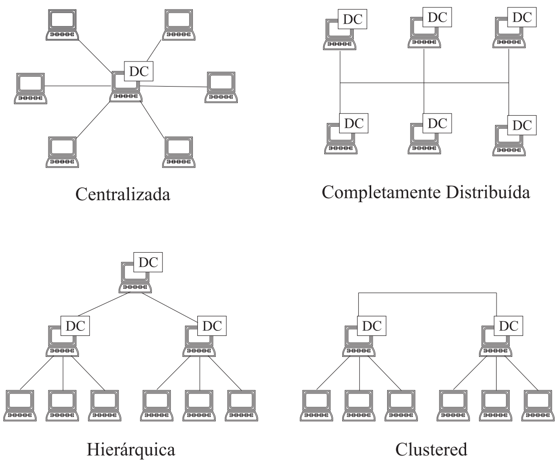


Figure 3.1: Centralização vs Descentralização: Abordagens Possíveis

dos. O contrário é que não é necessariamente verdade: pode haver um componente centralizado que é implementado fisicamente de forma distribuída.

3.2.3 Distribuição Global vs Distribuição Local

A distribuição de carga não tem necessariamente que ser feita tendo em consideração todos os nodos do sistema, ou seja, cada nodo pode, em cada instante, estar a distribuir a sua carga para um conjunto particular de nodos não necessariamente igual ao conjunto total de nodos. Dado isto, [Lüling et al., 1992] classificaram os mecanismos de distribuição como sendo locais ou globais. Num mecanismo local, cada nodo exerce a distribuição de carga num conjunto que inclui os seus vizinhos e ele próprio, enquanto que num mecanismo global este conjunto equivale ao total dos nodos de processamento.

Esta classificação é, no entanto, um pouco limitativa, na medida em que restringe a distribuição não global à distribuição entre os vizinhos e ele próprio. Por exemplo em [Kremien et al., 1993] é usada uma abordagem em que cada nodo distribui a sua carga para um subconjunto de nodos do sistema que pode variar ao longo do tempo e que não corresponde necessariamente ao conjunto dos seus vizinhos. Assim, passa-se a considerar distribuição de carga local quando cada nodo responsável por actividades de distribuição apenas considera um subconjunto dos nodos de processamento nessas mesmas actividades.

3.2.4 Objectivos

Os objectivos de um mecanismo de distribuição de carga estão presentes em todas as políticas do mesmo, sendo, por isso, este parâmetro de carácter geral. Embora o objectivo da distribuição de carga seja, genericamente, aumentar o desempenho do sistema através da transferência de tarefas dos nodos mais sobrecarregados para os menos sobrecarregados, a quantificação deste desempenho assume formas muito variadas em cada um dos mecanismos de distribuição desenvolvidos. Estas diferentes formas de quantificar o desempenho de um sistema refletem-se nos diferentes objectivos específicos que cada mecanismo procura alcançar. Consequentemente, estes objectivos deverão ser usados como mais um parâmetro de classificação e comparação de mecanismos de distribuição de carga. Em [Riedl and Richter, 1996] destacam-se os seguintes objectivos:

- Bons desempenhos do ponto de vista das tarefas:
  - Grande desempenho em termos de pequenos tempos de espera, pequenos tempos de serviço ou pequenos tempos de resposta;
  - Predictabilidade dos tempos de resposta;
  - Disponibilidade de serviços;
  - Robustez do sistema;
- Distâncias curtas (fisicamente) entre as tarefas e os dados, i.e., grande afinidade;
- Bons desempenhos do ponto de vista dos recursos:
  - Grande desempenho global, em particular grande débito de tarefas executadas;
  - Grande desempenho local, como por exemplo grande percentagem de acertos na *cache* ou pequenos tempos de comunicação;
- Situações equilibradas, em particular:
  - Distribuição equalizada da carga;
  - Uso ajustado dos recursos disponíveis;
- Boa utilização dos recursos:
  - Tempos pequenos de não utilização dos recursos, ou o contrário;

- Pouco uso dos recursos, e.g., pouco tráfego nas linhas.

No contexto do sistema distribuído subjacente a este estágio poderão ser enunciados mais alguns objectivos, nomeadamente:

- Garantir os recursos e prazos especificados pelas tarefas mais críticas;

- Maximizar o lucro obtido com a distribuição de carga, quer minimizando os custos de execuções remotas, quer maximizando os proveitos que advêm de executar tarefas de outras organizações.

Um mecanismo de distribuição pode não perseguir apenas um destes objectivos, mas sim um conjunto determinado de objectivos, devendo neste caso estar definida a forma em como cada um contribuí para a decisão final.

3.2.5 Política de Transferência

Como é a política de transferência que determina se um nodo está em condições de participar numa transferência de tarefas, é ela que também determina o início da mesma. Uma das grandes distinções entre mecanismos de distribuição de carga diz respeito a quem inicia as transferências de carga. Assim, temos três tipos de mecanismos [Shivaratri et al., 1992]:

- os iniciados pela origem, quando a actividade de distribuição é iniciada por um nodo com excesso de carga (origem) que tenta transferir tarefas para um nodo com pouca carga (receptor);

- os iniciados pelo receptor, quando essa mesma actividade é iniciada por um nodo com pouca carga que tenta encontrar um nodo sobrecarregado que lhe transfira algumas tarefas; e

- os iniciados simetricamente, quando tanto nodos origem como nodos receptores iniciam a transferência de carga.

Outra questão importante relacionada com o início da transferência é saber quando é que é activada a política de transferência, ou seja, quando é que se inicia o mecanismo de decisão que determina a transferência de uma tarefa. [Kremien and Kramer, 1992] distingue três possibilidades de activação:

- periódica, quando se decide periodicamente se um nodo está em condições de participar numa transferência;
- após mudança de estado, quando essa decisão se toma após uma mudança no estado do sistema, sendo a mais frequentemente usada a chegada de uma nova tarefa; e
- após mudança de estado com periodicidade mínima, numa combinação das duas anteriores, ou seja, a decisão é tomada após uma mudança de estado apenas se já decorreu um determinado intervalo desde a última decisão.

Finalmente, há o tipo de política propriamente dito, ou seja, como é que um nodo decide se está em condições de participar numa transferência. De entre os mais variados tipos de política destacam-se os seguintes pela sua frequência de utilização [Shivaratri et al., 1992]:

- políticas tipo *threshold*, quando a decisão é tomada comparando a carga actual do nodo com um valor de referência: nas políticas iniciadas pela origem a carga tem de ser superior ao valor de referência, enquanto que nas iniciadas pelo receptor a carga deve ser inferior; e
- políticas relativas, quando a decisão passa pela comparação da carga com a dos restantes nodos: por exemplo, um nodo pode considerar-se receptor se a sua carga for inferior em pelo menos um determinado δ à de um outro nodo, ou se for o nodo com a carga mais baixa do sistema.

3.2.6 Política de Selecção

Uma das primeiras questões que se colocam ao se conceber um novo mecanismo de distribuição de carga é se as tarefas poderão ou não ser transferidas a meio da execução. Assim há uma distinção entre mecanismos com:

- transferências preemptivas, quando é possível interromper uma tarefa a meio da sua execução para ser transferida para outro nodo; esta operação é geralmente bastante custosa, na medida em que a transferência do estado da tarefa, que pode ser bastante grande e complexo, é normalmente muito difícil [Shivaratri et al., 1992]; e
- transferências não preemptivas, quando não é possível interromper as tarefas a meio da sua execução, ou seja, só se podem transferir tarefas que ainda não começaram a executar; embora neste caso não seja necessário transferir o estado da tarefa, continua a

ser necessário transferir a informação relativa ao contexto em que a tarefa iria executar, como por exemplo a localização da directoria de trabalho do utilizador ou os privilégios herdados pela tarefa [Shivaratri et al., 1992].

Outra questão importante diz respeito à possibilidade de transferir uma tarefa que já tenha sido transferida para o nodo. Existem três hipóteses de actuação:

- a tarefa é considerada como qualquer tarefa que acaba de ser criada no nodo e portanto pode ser sempre redistribuída; esta atitude pode originar que uma tarefa seja transferida ilimitadamente acabando por nunca executar;
- a tarefa tem de ser sempre executada independentemente de o nodo ter ficado sobrecarregado com a sua chegada; e
- a tarefa apenas pode sofrer um número limitado de transferências, ou seja, após sofrer a sua última transferência possível, a tarefa tem obrigatoriamente de ser executada pelo nodo receptor dessa transferência.

As políticas seguidas na escolha da tarefa para transferir são muito diversificadas. Algumas das utilizadas são, por exemplo, escolher a tarefa [Shivaratri et al., 1992]:

- a mais pequena, pois implica menos custos de transferência;
- a que se vai executar mais tempo, para compensar os custos da transferência;
- a que tem menos dependência dos recursos do nodo onde foi criada; e
- a que está há mais tempo à espera, para minimizar tempos de espera.

3.2.7 Política de Localização

Existem várias políticas usadas na escolha de um parceiro para uma transferência, como por exemplo, escolher:

- aleatoriamente, a política mais simples mas ao mesmo tempo das mais eficazes;
- o nodo com menos carga, no caso de políticas iniciadas pela origem, ou o nodo com mais carga, no caso de políticas iniciadas pelo receptor;

- aleatoriamente entre um conjunto de nodos cuja carga difere pelo menos um determinado valor Δ;
- aleatoriamente entre um conjunto de nodos cuja carga esteja abaixo, ou acima, de um determinado valor de referência;
- por sondagem, quando um nodo pergunta a outro para ver se está na disposição de aceitar a tarefa em questão; e
- por leilão, pede-se a todos os outros nodos para dizerem quais as condições disponíveis para executarem uma determinada tarefa, escolhendo-se depois o que oferece as condições mais favoráveis;

3.2.8 Política de Informação

Globalmente, há duas formas de obter a informação acerca do estado da carga no sistema [Riedl and Richter, 1996]:

- É obtida através de medidas comunicadas à origem da distribuição da carga; ou
- É obtida a partir de conjecturas que diferem da informação comunicada, quer porque a informação disponível está incompleta, quer porque se assume que esta difere da realidade física.

Em ambos os casos podem existir duas formas de comunicar a informação [Kremien and Kramer, 1992]:

- A origem da distribuição pede a informação quando esta se torna necessária; ou
- Os nodos de processamento disseminam a informação relativa ao seu estado. Esta política subdivide-se de acordo com a altura em que a informação é disseminada:
  - Disseminação periódica. Periodicamente os nodos enviam a informação relativa ao seu estado para os nodos responsáveis pela distribuição de carga;
  - Disseminação após mudança de estado. Neste caso a informação é disseminada sempre que o estado da carga do nodo varia para além de um certo grau [Shivaratri et al., 1992]. Neste caso a classificação deverá ser acompanhada pelo número de estados existentes para graduar a carga de um nodo. A frequência das mudanças de estado pode ser razoavelmente controlada variando o número dos mesmos [Kremien and Kramer, 1992];

– Combinação das anteriores. Nos casos em que a informação é disseminada aquando de mudanças de estado, se o sistema estiver com a carga bastante flutuante poderá haver um execesso de mensagens. Para prevenir esta situação, alguns sistemas usam uma política mista: a informação é disseminada quando o estado muda apenas se já decorreu um determinado intervalo desde a última disseminação [Kremien and Kramer, 1992].

Uma das características mais importantes dos mecanismos de distribuição de carga é a medida de carga usada para caracterizar o estado de cada nodo de processamento [Shivaratri et al., 1992, Kunz, 1991]. Para o sistema ter um desempenho razoável é necessário que estas medidas se correlacionem bem com os objectivos pretendidos com a distribuição de carga. Algumas das medidas mais utilizadas são [Kunz, 1991]:

- o número de tarefas em execução;
- o tamanho da memória livre disponível;
- a frequência de mudanças de contexto do CPU;
- a frequência de invocação de rotinas do sistema;
- a média da carga num determinado período de tempo; e
- a quantidade de tempo livre de CPU.

Existem mecanismos de distribuição de carga onde os nodos não utilizam informação relativa aos outros nodos. Nestes casos pode-se classificar a política de informação como sendo local, passando apenas a fazer sentido classificar a mesma quanto à medida de carga utilizada.

3.2.9 Adaptabilidade

A forma como o mecanismo de distribuição de carga procura alcançar os seus objectivos pode variar conforme o estado da carga do sistema. Assim, é possível distinguir entre mecanismos com [Riedl and Richter, 1996]:

- uma única e estática estratégia;
- uma única estratégia mas com variação dinâmica dos seus parâmetros; e
- várias estratégias diferentes.

Os mecanismos correspondentes aos dois últimos casos são, normalmente, designados por adaptativos. Esta adaptabilidade pode verificar-se em qualquer das políticas e parâmetros apresentados acima.

Se por exemplo, uma política de distribuição funciona melhor em certas condições do sistema, e outra funciona melhor noutras condições, um mecanismo adaptativo simples poderia escolher entre essas duas políticas conforme o estado do sistema. Quando um sistema está uniformemente sobrecarregado, de tal forma que não se poderiam obter melhorias de desempenho por se fazer distribuição de carga, um mecanismo não adaptativo continuaria a tentar distribuir, provocando a degradação das condições do sistema. Pelo contrário, um mecanismo adaptativo, ao verificar esse estado, poderia optar por deixar de distribuir tarefas e consequentemente não piorava o estado do sistema [Shivaratri et al., 1992].

3.2.10 Síntese dos Parâmetros de Classificação Adoptados

Os parâmetros apresentados dividem-se em parâmetros genéricos de classificação e os respeitantes às políticas que compõem um mecanismo de distribuição de carga. A síntese desses parâmetros é apresentada na tabela 3.1. A escolha da forma tabular para sintetizar os parâmetros deve-se à facilidade de leitura que apresenta, que será de grande utilidade na classificação de cada um dos mecanismos a seguir apresentados.

Não aparece nas tabelas nenhuma entrada respeitante à adaptabilidade dos mecanismos, pois essa adaptabilidade pode-se verificar em qualquer dos parâmetros, não sendo suficiente classificar o mecanismo como adaptativo ou não adaptativo. Aquando da classificação de um mecanismo de distribuição de carga, sempre que este seja adaptativo em relação a um determinado parâmetro, essa adaptabilidade deverá ser referida na entrada respectiva.

3.3 Alguns Mecanismos Convencionais

3.3.1 Distribuição Aleatória com Redistribuição Limitada

A Distribuição Aleatória com Redistribuição Limitada é um dos mecanismos de distribuição mais simples que alguma vez foi concebido. Porventura deverá ser impossível atribuir a origem deste mecanismo a um autor em particular, mas a versão apresentada é a descrita em [Eager et al., 1986], uma das mais referênciadas na literatura.

A parte mais complicada deste mecanismo é porventura a sua política de transferência, uma política do tipo *Threshold*. Sempre que uma nova tarefa é criada num nodo verifica-se se

Estratégia global	Dinâmico vs Estático	
	Centralizado vs Distribuído	
	Global vs Local	
Objectivos	Tipo	Pequenos tempos de resposta
		Grande afinidade
		Distribuição equalizada, ...
Transferência	Quem inicia	Origem
		Receptor
		Simétrico
	Quando se inicia	Periódica
		Após mudança de estado
		Após mudança de estado com periodicidade mínima
	Tipo	*Threshold*
		Relativa, ...
Selecção	Quando se transfere	Preemptiva
		Não preemptiva
	Redistribuição	Impossível
		Limitada
		Ilimitada
	Tarefa escolhida	Mais pequena
		Maior tempo de execução, ...
Localizaçao	Qual o nodo	Aleatório
		Leilão
		Sondagem, ...
Informação	Tipo	Local
		Medidas comunicadas
		Conjecturas a partir de informação comunicada
	Obtenção (não local)	A pedido
		Disseminada periodicamente
		Disseminada após mudança de estado
		Disseminada após mudança de estado com uma periodicidade mínima
	Medida de carga	Número de tarefas em execução
		Tamanho da memória livre disponível, ...

Table 3.1: Síntese dos Parâmetros de Classificação de Mecanismos de Distribuição de Carga

o número de tarefas que o nodo contém é superior a um determinado valor de referência. Caso o seja, o nodo transfere essa tarefa para um outro nodo escolhido aleatoriamente. O número de transferências que a tarefa pode sofrer está limitado a um determinado valor como forma de prevenir a instabilidade do mecanismo. Para uma síntese das características ver tabela 3.2.

Estratégia global	Dinâmico	
	Completamente distribuído	
	Global	
Objectivos	Tipo	Pequenos tempos de resposta médios
Transferência	Quem inicia	Origem
	Quando se inicia	Após mudança de estado: chegada de tarefa
	Tipo	*Threshold*
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Limitada
	Tarefa escolhida	Tarefas criadas recentemente
Localizaçao	Qual o nodo	Aleatório
Informação	Tipo	Local
	Medida de carga	Número de tarefas em execução

Table 3.2: Classificação da Abordagem Aleatória com Redistribuição Limitada

Embora simples, este mecanismo é de grande eficiência principalmente porque não necessita de trocas de informação entre nodos. A sua reconhecida eficiência originou que este mecanismo seja frequentemente usado como mecanismo de referência em estudos comparativos de abordagens à distribuição de carga, e.g. [Kremien and Kramer, 1992, Shivaratri et al., 1992, Zhou, 1988], embora, ocasionalmente, seja utilizado sem possibilidade de redistribuição de tarefas, ou seja com limite de transferências igual a um.

Como já foi dito anteriormente, o mecanismo aleatório agora apresentado é dos mais referenciados na literatura. No entanto, existem outros mecanismos aleatórios. Por exemplo, em [Lüling et al., 1992] é apresentado um mecanismo com iniciação periódica, onde a cada intervalo de tempo uma certa quantidade de tarefas, escolhidas aleatoriamente, são transferidas para um nodo, também escolhido aleatoriamente.

3.3.2 Distribuição Simétrica Adaptativa e Estável

Em [Shivaratri et al., 1992] é apresentado um mecanismo adaptativo no respeitante à sua política de transferência e que alterna, automaticamente em função da carga do sistema, entre as estratégias de iniciação pela origem, iniciação simétrica e iniciação pelo receptor.

Nesta abordagem, a informação recolhida a pedido aquando da tentativa de localização de um nodo para uma transferência de tarefas é, ao contrário do normal, guardada e usada para classificar os nodos do sistema como nodos origem, quando sobrecarregados, nodos receptores, quando com pouca carga, e nodos OK, quando a carga está em valores intermédios, sendo usados nesta classificação dois valores de referência. Cada nodo mantém três listas correspondentes aos nodos em cada um dos estados. As funções de consulta e actualização destas listas estão concebidas para serem escaláveis em relação ao tamanho do sistema. Inicialmente cada nodo considera todos os outros como receptores, ou seja, estão todos na lista dos nodos receptores, estando as listas dos nodos origem e OK vazias.

<table>
<tr><td rowspan="3">Estratégia global</td><td colspan="2">Dinâmico</td></tr>
<tr><td colspan="2">Completamente distribuído</td></tr>
<tr><td colspan="2">Global</td></tr>
<tr><td>Objectivos</td><td>Tipo</td><td>Pequenos tempos de resposta médios</td></tr>
<tr><td rowspan="3">Transferência</td><td>Quem inicia</td><td>Adaptativo entre origem, simétrico e receptor</td></tr>
<tr><td>Quando se inicia</td><td>Após mudança de estado: chegada ou saída de tarefa</td></tr>
<tr><td>Tipo</td><td>Dois valores de threshold</td></tr>
<tr><td rowspan="3">Selecção</td><td>Quando se transfere</td><td>Adaptativa entre Não Preemptiva e Preemptiva</td></tr>
<tr><td>Redistribuição</td><td>Ilimitada</td></tr>
<tr><td>Tarefa escolhida</td><td>Adaptativa entre tarefa criada mais recentemente e não definida</td></tr>
<tr><td>Localizaçao</td><td>Qual o nodo</td><td>Sondagem com limite</td></tr>
<tr><td rowspan="3">Informação</td><td>Tipo</td><td>Medidas comunicadas</td></tr>
<tr><td>Obtenção</td><td>A pedido</td></tr>
<tr><td>Medida de carga</td><td>Número de tarefas em execução</td></tr>
</table>

Table 3.3: Classificação da Abordagem Simétrica Adaptativa e Estável

A política de transferência é activada sempre que chega uma nova tarefa ou quando acaba uma tarefa. É uma política do tipo *threshold*, que, como já foi dito, usa dois valores de referência para, em função do número de tarefas em execução, classificar o nodo como sendo um nodo origem, receptor ou OK. Se o nodo for um nodo origem a política de localização sonda os nodos da lista de receptores para determinar se ainda estão nesse estado. O nodo sondado aproveita a mensagem para actualizar o estado do nodo origem, retirando-o da lista em que se encontra e colocando-o na lista dos nodos origem; depois, informa-o do seu estado. Em função da resposta, se o nodo sondado continuar a ser um nodo receptor, a tarefa é transferida para esse nodo. Caso contrário, actualiza-se o estado do nodo sondado e continua-se o processo até que um nodo receptor seja encontrado, se atinja um determinado limite de sondagens ou a lista de nodos receptores fique vazia. Nestes casos a tarefa é executada localmente.

Se o nodo for um nodo receptor, começa uma operação de sondagem para encontrar um nodo origem. Neste caso, os nodos são sondados na seguinte ordem:

- da cabeça para a cauda da lista de nodos origem, a informação mais actualizada é usada primeiro;
- da cauda para a cabeça da lista de nodos OK, a informação mais antiga é usada primeiro, na esperança de que o nodo já seja um receptor; e
- da cauda para a cabeça da lista de nodos receptores, mais uma vez, a informação mais antiga é usada primeiro.

O procedimento de sondagem é semelhante ao activado quando o nodo é um nodo origem, ou seja, os nodos sondados aproveitam para actualizar a informação sobre o nodo receptor e depois enviam o seu estado para este. Se o nodo sondado for um nodo origem então este transfere uma tarefa para o nodo receptor. Ao receber a resposta o nodo receptor actualiza o estado do nodo sondado. A sondagem para se o nodo deixa de ser receptor ou se se atinge um determinado limite no número de sondagens possíveis.

Quando um nodo é origem, a política de selecção apenas considera as tarefas recentemente criadas. Quando o nodo é receptor podem ser transferidas tarefas do nodo remoto de forma preemptiva. A política de selecção não está especificada neste caso, mas aparentemente nada impede que, tarefas anteriormente transferidas para o nodo não possam ser transferidas de novo preemptivamente. Assim, a possibilidade de redistribuição é ilimitada.

A grande vantagem deste mecanismo resulta da adaptabilidade conseguida em função da carga do sistema. Quando a carga do sistema aumenta, a probabilidade de um nodo origem encontrar um nodo receptor é muito baixa. Assim, conforme se forem fazendo sondagens vão-se retirando todos os nodos da lista de receptores, acabando por cessar a actividade por parte dos nodos origem devido a não haver nodos para sondar. Desta forma, evita-se sobrecarregar ainda mais o sistema com actividades de distribuição quando este já está sobrecarregado. Essa actividade só é reiniciada quando a carga baixa e começam a aparecer nodos receptores a fazer sondagens. Um resumo das características desta abordagem é apresentada na tabela 3.3.

3.3.3 Distribuição Centralizada com Restrição no Número de Actualizações

O mecanismo de distribuição de carga centralizado típico prevê que todos os nodos de processamento enviem, periodicamente, para o nodo encarregue da distribuição o valor da sua carga. Sempre que uma nova tarefa é criada num nodo é enviado um pedido de distribução para o serviço central que devolve a identificação do nodo mais apto a receber a tarefa, ou seja, o nodo com menos carga. O nodo onde a tarefa foi criada envia-a depois para o nodo seleccionado. Este mecanismo tem um pequeno problema que consiste no facto de os nodos enviarem indiscriminadamente o seu estado para o serviço central de distribuição, mesmo que esse valor se mantenha em relação à última mensagem. Grandes melhorias de desempenho, quer ao nível da rede de interligação, quer ao nível do servidor, seriam conseguidas se o número destas mensagens de actualização fosse restringido.

O mecanismo centralizado aqui apresentado, proposto em [Theimer and Lantz, 1989], procura aumentar o desempenho do sistema de distribuição ao nível da escalabilidade com a diminuição dessa taxa de actualizações. Efectivamente, o sistema será mais escalável se houver uma menor ocupação dos recursos partilhados por todos os nodos, nomeadamente, da rede de interligação e do nodo encarregue da distribuição. Assim, a primeira melhoria proposta consiste em apenas enviar o valor da carga se esta variou em mais de um Δ desde a última vez que actualizou o seu valor. Esta política de informação corresponde à muito utilizada política de disseminação após mudança de estado com uma periodicidade mínima, sugerida por exemplo em [Kremien and Kramer, 1992] com objectivos idênticos.

A segunda melhoria introduzida consiste em restringir o número de nodos que enviam o seu estado para o servidor. A política de selecção do melhor nodo para receber uma tarefa consiste em escolher o nodo com menos carga. Com esta política de selecção apenas seria necessário actualizar a informação relativa aos nodos com menos carga. Assim, foi criado um valor de

Estratégia global	Dinâmico	
	Centralizado	
	Global	
Objectivos	Tipo	Pequenos tempos de resposta médios
Transferência	Quem inicia	Origem
	Quando se inicia	Após mudança de estado: chegada de tarefa
	Tipo	Distribuí todas as tarefas
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Impossível
	Tarefa escolhida	A que solicitou distribuição
Localizaçao	Qual o nodo	Nodo com menos carga
Informação	Tipo	Medidas comunicadas
	Obtenção	Disseminada após mudança de estado com uma periodicidade mínima Só disseminam os nodos com carga abaixo de um *threshold* que varia dinâmicamente
	Medida de carga	Não definida

Table 3.4: Classificação da Abordagem Centralizada com Restrição no Número de Actualizações

threshold que serve para restringir as actualizações aos nodos cuja carga não o exceda. Este valor é disseminado regularmente e cada nodo apenas envia o seu estado para o servidor central se a sua carga estiver abaixo desse valor. A razão do envio regular do valor de *threshold* é o facto de este variar dinamicamente em função da carga do sistema. Quando existem muitos nodos com carga baixa diminui-se o *threshold* e quando existem muitos nodos com carga alta é aumentado. Estas modificações são feitas na tentativa de manter constante o número de nodos a actualizar o seu estado. Sempre que um nodo é escolhido para receber uma tarefa o servidor coloca a sua carga a 100% para evitar ser imediatamente seleccionada para receber outra tarefa. A síntese das restantes características deste mecanismo é apresentada na tabela 3.4.

Um dos problemas das abordagens centralizadas é a sua não tolerância a falhas no nodo servidor. Para resolver este problema os autores propõem a existência de tantos nodos *watch* *dogs* quantas as falhas que se pretendam toleradas. Assim que um destes nodos detecta uma falha no nodo servidor, uma nova cópia do mecanismo de distribuição é instânciada e tenta-se reconstruir a informação perdida enviando para todos os nodos um pedido para enviarem imediatamente o seu estado. Também existem mecanismos para prevenir a activação simultânea de diferentes instâncias do mecanismo de distribuição pelos diversos *watch dogs*.

3.3.4 Distribuição Iniciada pelo Receptor com Reserva de Tarefas

Este mecanismo foi retirado de [Zhou, 1988] e constitui o único mecanismo desta análise onde as actividades de distribuição são iniciadas pelos nodos receptores. A síntese das características do mesmo pode ser consultada na tabela 3.5. A sua descrição mais detalhada será agora apresentada.

Sempre que uma tarefa é concluída num nodo verifica-se se a carga local, medida através do número de tarefas presentemente em execução, está abaixo de um determinado valor de *threshold*. Caso isso se verifique então começa um processo de sondagem para encontrar nodos que estejam sobrecarregados, para se proceder à transferência de tarefas para o nodo receptor. Dadas as dificuldades em implementar um mecanismo de transferência preemptiva de tarefas, o autor apenas considera que se podem transferir tarefas recentemente criadas. Dada esta restrição, o processo de sondagem não pode simplesmente procurar um nodo sobrecarregado e proceder à transferência de uma das tarefas desse nodo. Para contornar este problema este mecanismo procede à reserva de tarefas, ou seja, sempre que um nodo está em condições de receber tarefas reserva as mesmas nos nodos sobrecarregados, e, quando forem criadas tarefas nesses nodos, elas serão transferidas, não preemptivamente, para o nodo que fez a reserva. Mais concretamente, um nodo receptor sonda, aleatoriamente, os restantes nodos do sistema e sempre que encontra um nodo com carga acima de um valor de *threshold* coloca uma reserva de tarefas nesse nodo. Existe um limite no número de reservas que se podem efectuar para evitar que o processo de sondagem proceda indefinidamente. Sempre que um nodo recebe uma reserva de tarefa, esta é colocada no topo de uma pilha de reservas para que as transferências se façam para os nodos que reservaram tarefas mais recentemente, ou seja, para aqueles cuja probabilidade de continuarem com pouca carga é superior. Os nodos sobrecarregados não transferem qualquer tarefa recentemente criada. Para que uma tarefa seja elegível para ser transferida é necessário que o seu tempo estimado de execução seja superior a um determinado *threshold*. Esta política de selecção deriva do pressuposto de que não vale a pena transferir tarefas que executem pouco tempo, pois o tempo perdido em todo o processo de distribuição anularia todas as vantagens que dele poderiam advir.

Estratégia global	Dinâmico	
	Completamente Distribuído	
	Global	
Objectivos	Tipo	Pequenos tempos de resposta médios
Transferência	Quem inicia	Receptor
	Quando se inicia	Após mudança de estado: saída de tarefa
	Tipo	*Threshold*
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Impossível
	Tarefa escolhida	Tarefas recentemente criadas com tempo de execução superior a um *threshold*
Localizaçao	Qual o nodo	Sondagem
Informação	Tipo	Medidas comunicadas
	Obtenção	A pedido
	Medida de carga	Número de tarefas em execução

Table 3.5: Classificação da Abordagem Iniciada pelo Receptor com Reserva de Tarefas

3.3.5 Distribuição Escalável com Partições

A escalabilidade em mecanismos que englobem todos os nodos do sistema nunca poderá ser ilimitada. Só a necessidade de um nodo recolher informação acerca de todos os outros nodos implica que, a partir de um certo tamanho do sistema, esse nodo não faça outra coisa para além de actualizar informação. Em [Kremien et al., 1993] é proposta uma abordagem à distribuição de carga que procura resolver este problema: cada nodo do sistema apenas interage com uma parte desse sistema correspondente a uma partição. A síntese das características deste mecanismo pode ser consultada na tabela 3.6.

Cada nodo pode, em cada instante, estar num de três estados: sobrecarregado, com média carga ou com pouca carga. Estes estados são obtidos através de dois valores de *threshold* aplicados à média da utilização do CPU num intervalo de tempo passado. Sempre que este estado muda é enviada uma mensagem de actualização para todos os nodos da sua partição. A informação acerca dos nodos da sua partição é mantida numa *cache* local, correspondendo o seu tamanho ao tamanho da partição.

Estratégia global	Dinâmico	
	Completamente Distribuído	
	Local	
Objectivos	Tipo	Pequenos tempos de resposta médios
Transferência	Quem inicia	Origem
	Quando se inicia	Após mudança de estado: chegada de tarefa
	Tipo	Dois valores de *threshold* para a utilização do CPU e um para as tarefas em execução
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Impossível
	Tarefa escolhida	A que solicitou distribuição
Localizaçao	Qual o nodo	Sondagem
Informação	Tipo	Medidas comunicadas
	Obtenção	Disseminação após mudança de estado para os nodos do domínio
	Medida de carga	Utilização de CPU e Número de tarefas em execução (a última só é usada em decisões locais)

Table 3.6: Classificação da Abordagem Escalável com Partições

Sempre que uma tarefa é criada decide-se se será executada local ou remotamente. A execução remota apenas acontece se o nodo está sobrecarregado e se o número de tarefas presentemente em execução exceder um determinado *threshold*. Estas duas medidas são conjugadas porque a última, devido à sua actualidade, dá uma indicação precisa do estado futuro do nodo evitando que uma tarefa seja executada remotamente se a carga do nodo está a baixar. Esta atitude demasiado restritiva em relação à execução remota de tarefas deve-se à crença dos autores em relação à estabilidade de um mecaninsmo de distribuição: um mecanismo é estável se tiver uma pequena percentagem de execuções remotas de tarefas.

A escolha do nodo para executar uma tarefa é feita através de sondagem aos nodos da *cache* que estejam num estado com pouca carga. O pedido de execução remota é acompanhado da tarefa para, caso o nodo esteja efectivamente com pouca carga, a sua execução começar imediatamente. Se o nodo não estiver no estado indicado pela *cache* a tarefa é recusada. Se não houverem mais candidatos para sondar a tarefa é executada localmente.

É do interesse de um nodo manter na sua partição o maior número possível de nodos com um estado oposto ao seu, ou seja, os nodos sobrecarregados preferem interactuar com nodos com pouca carga e estes com os nodos sobrecarregados. Os nodos com carga média não iteragem com outros nodos, ou seja, tem uma partição vazia. Esta é a regra utilizada para variar dinamicamente o tamanho da partição. Estas variações podem ocorrer sempre que chega uma mensagem de actualização de estado, quando o nodo muda de estado ou periodicamente. O protocolo usado para a gestão das partições garante a inclusão mútua de nodos com estados opostos, ou seja se um nodo k está na partição do nodo j então o nodo j também está na partição do nodo k. Também é garantida a imunidade a falhas de nodos e perdas de mensagens. O protocolo tenta manter o tamanho das partições entre dois limites. Quando muitos nodos são retirados da partição (por já não haver oposição de estados) e é atingido o limite inferior então são escolhidos aleatoriamente alguns nodos do sistema para a integrar, recomeçando todo o processo.

3.3.6 Distribuição Usando um Autómato Estocástico com Aprendizagem

A versão deste mecanismo aqui apresentada foi retirada de [Kunz, 1991], embora não tenha sido concebida pelo autor deste artigo. Esta é mais uma abordagem completamente distribuída, ou seja, existe uma cópia idêntica do mecanismo de distribuição em todos os nodos do sistema. Cada nodo periodicamente verifica o seu estado, classificando-se como sobrecarregado ou não sobrecarregado em referência a um determinado valor de *threshold*, sendo este estado difundido para os restantes nodos. No artigo de onde foi retirado este mecanismo pretende-se analisar o comportamento do mesmo em função do uso de diferentes medidas de carga. A medida que o autor provou ser mais eficiente consiste no número de tarefas em execução no nodo. Como tal, assume-se esta como a medida utilizada, devendo o valor de *threshold* estar especificado em número de tarefas.

O mecanismo de distribuição é activado sempre que uma nova tarefa é criada no nodo. Se o nodo não estiver sobrecarregado ou todos os nodos estiverem sobrecarregados a tarefa

Estratégia global	Dinâmico	
	Completamente Distribuído	
	Global	
Objectivos	Tipo	Pequenos tempos de resposta médios
Transferência	Quem inicia	Origem
	Quando se inicia	Após mudança de estado: chegada de tarefa
	Tipo	Adaptativa entre *Threshold* e não distribuir quando o sistema está sobrecarregado
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Impossível
	Tarefa escolhida	A que activou a transferência
Localizaçao	Qual o nodo	Automato estocástico com aprendizagem
Informação	Tipo	Medidas comunicadas
	Obtenção	Disseminação periódica
	Medida de carga	Número de tarefas em execução

Table 3.7: Classificação da Abordagem Usando um Autómato Estocástico com Aprendizagem

é executada localmente. Nos restantes casos a tarefa é executada remotamente num nodo determinado através de um autómato estocástico com aprendizagem. Este autómato serve para escolher qual a melhor acção a ser executada de entre um conjunto de acções possíveis. Neste caso as acções são da forma "executar a tarefa no nodo x", onde x identifica um nodo remoto. A forma do autómato escolher a melhor acção é agora descrita. Existe uma probabilidade de escolha atribuída a cada acção possível. Inicialmente todas estas probabilidades são idênticas. Cada vez que é escolhida uma acção observa-se a reação do nodo remoto à sua escolha. Esta reação é usada para actualizar as probabilidades associadas à acção respectiva, acabando cada nodo por aprender quais as melhores acções a ser tomadas.

Este conceito geral de autómato foi extendido com a introdução da noção de estado do autómato. O autómato acima descrito usa apenas um vector de probabilidades. Com esta noção é usado um vector de probabilidades diferente para cada estado do autómato. Este estado é determinado em função do estado do sistema. Sempre que uma tarefa vai ser distribuída determina-se, primeiro o estado do sistema, depois o estado correspondente do autómato e, finalmente, utiliza-se o vector de probabilidades correspondente a esse estado tal como foi apresentado no parágrafo anterior. Num sistema com n nodos existem $n-1$ estados possíveis para o autómato. A forma como se determina o estado actual é a seguinte: o nodo i percorre sucessivamente a informação relativa aos estados dos nodos $(i+j+1) mod 5$ para $j = 0, \ldots, n-2$ até ser encontrado um nodo não sobrecarregado. O valor de j determina qual o estado do autómato e, consequentemente, qual o vector de probabilidades usado para escolher a acção a tomar. A classificação deste mecanismo é apresentada na tabela 3.7.

3.3.7 Distribuição de Tarefas com Prazos de Execução e Requisitos de Recursos

Finalmente, apresenta-se um mecanismo de distribuição concebido para distribuir tarefas em sistemas que operam em tempo real [Ramamritham et al., 1989]. Neste sistema as tarefas são caracterizadas por possuirem prazos de execução que devem ser garantidos. Para permitir a distribuição destas tarefas é necessário que sejam submetidas para execução com uma descrição dos recursos que necessitam. Esta lista de recursos é confrontada com a disponibilidade dos recursos do nodo onde foi criada para se saber se a tarefa poderá ou não ser garantida. Por garantida entende-se que, não importando o que possa acontecer exceptuando falhas de sistema, a tarefa será executada dentro do prazo estabelecido. O objectivo do mecanismo de distribuição de carga é, neste caso, maximizar o número de tarefas que são garantidas.

Quando uma tarefa não pode ser garantida localmente o mecanismo de distribuição entra em funcionamento para tentar encontrar um nodo que a possa garantir. Este mecanismo baseia-se na informação enviada pelos outros nodos acerca dos excedentes de recursos que possuem. Periodicamente, cada nodo calcula o tempo total que cada recurso esteve livre num determinado intervalo passado, e envia um vector com informação relativa aos excedentes de todos os recursos para os restantes nodos. Para minizar a carga imposta no sistema pelo mecanismo de distribuição, esta informação apenas é enviada para um conjunto limitado de nodos. Este conjunto consiste nos nodos que mais tarefas lhe enviaram para executar no já mencionado intervalo de tempo. Esta decisão baseia-se na suposição de que os nodos que mais tarefas lhe enviaram no passado continuarão a sê-lo no futuro.

A política de localização de um nodo remoto onde a tarefa possa ser executada combina duas estratégias: a estimativa do nodo com mais excedentes e a técnica de leilão da tarefa.

Estratégia global	Dinâmico	
	Completamente Distribuído	
	Global	
Objectivos	Tipo	Maximizar o número de tarefas executadas dentro do prazo
Transferência	Quem inicia	Origem
	Quando se inicia	Após mudança de estado: chegada de tarefa
	Tipo	Distribuí sempre que não pode executar localmente uma tarefa sem falhar o prazo
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Impossível
	Tarefa escolhida	Tarefa que não possa ser executada localmente sem falhar o seu prazo
Localizaçao	Qual o nodo	Estimado e leilão
Informação	Tipo	Medidas comunicadas
	Obtenção	Disseminação periódica para os nodos que mais tarefas enviaram para o nodo em causa
	Medida de carga	Excedentes de recursos

Table 3.8: Classificação da Abordagem para Distribuição de Tarefas com Prazos de Execução e Requisitos de Recursos

Quando uma tarefa T não pode ser garantida localmente, o nodo começa por escolher k nodos com suficiente excedente nos recursos necessitados por T. Se o nodo mais bem cotado tiver uma probabilidade de garantir a tarefa superior a um determinado valor, a tarefa é imediatamente enviada para esse nodo. Ao mesmo tempo é enviado um pedido de oferta para os restantes $k-1$ nodos. Este pedido de oferta também contem a identificação do nodo escolhido, caso exista algum. Quando um nodo recebe um pedido de oferta, esta é calculada por forma a reflectir a probabilidade de a tarefa poder ser garantida nesse nodo. Esta oferta é depois enviada para o nodo escolhido, caso exista algum, ou para o nodo onde a tarefa foi criada quando este nodo não existe.

Quando é inicialmente escolhido um nodo para executar a tarefa, devido à estimativa de que este possui os recursos suficientes para a executar, este começa por verificar se realmente consegue executar a tarefa dentro do prazo determinado. Se o conseguir, todas as ofertas que entretanto lhe vão chegar serão ignoradas. Caso contrário, a tarefa será enviada para o nodo que fizer a melhor oferta, sendo o nodo onde a tarefa foi criada notificado dessa escolha. Quando nenhum nodo é escolhido imediatamente para executar a tarefa, a escolha da melhor oferta é feita pelo nodo onde foi criada. Por razões de eficiência, cada pedido de oferta é acompanhado por um valor de oferta mínimo, para evitar mensagens desnecessárias, e por um prazo limite para a entrega das ofertas. Este prazo é calculado em função do prazo de execução da tarefa e do tempo necessário para a sua execução, para evitar prosseguir a tentativa de distribuição quando a tarefa já não pode ser garantida.

Quando um nodo faz uma oferta os recursos não ficam reservados para a tarefa em causa, pois esta pode não lhe ser atribuída e essa reserva limitaria as ofertas para a execução de outras tarefas do sistema, limitando o desempenho do mesmo. A não existência desta reserva pode provocar que, quando a tarefa chega ao nodo escolhido para ser executada, já não existam os recursos para a executar, não podendo assim ser garantida a sua execução. Quando uma tarefa não é garantida no nodo onde foi criada, nem no nodo escolhido inicialmente por estimativa, nem no nodo escolhido por leilão para a executar, ela é pura e simplesmente rejeitada e não é executada. Neste caso, resta ao utilizador a hipótese de a invocar de novo com um prazo de execução mais alargado ou de escolher outra tarefa que necessite de menos recursos.

Devido à relativa complexidade deste mecanismo, esta apresentação não abordou alguns aspectos importantes, que poderão ser consultados no artigo de onde foi retirado, nomeadamente: a heurística de escalonamento local das tarefas, a forma como são estimadas as probabilidades dos nodos garantirem as tarefas e a forma como são calculadas as ofertas. Acredita-se, no entanto, que a compreensão do funcionamento global do mecanismo não foi afectada por estas omissões. O resumo das características do mecanismo e sua classificação poderão ser consultadas na tabela 3.8.

3.4 Análise Qualitativa e Comparativa

3.4.1 Escalabilidade

Esta é provavelmente a questão mais delicada na concepção de um mecanismo de distribuição para o sistema subjacente a este estágio. Como já foi dito, pretende-se distribuir as tarefas entre diversas organizações e não apenas dentro de uma organização. Isto implica que o mecanismo a desenvolver seja ilimitadamente escalável, ou mais concretamente, que o seu desempenho seja absolutamente independente do tamanho do sistema.

A primeira questão normalmente colocada quando se pretende conceber um sistema escalável é se ele deve ser centralizado ou distribuído. Esta questão é também a uma das mais polémicas pois as opiniões entre os autores divergem frequentemente, havendo uns que consideram os sistemas centralizados mais escaláveis, e.g. [Theimer and Lantz, 1989, Zhou, 1988], enquanto que outros defendem que os sistemas distribuídos é que são mais escaláveis, e.g. [Shivaratri et al., 1992, Kremien and Kramer, 1992, Kunz, 1991]. Algumas destas opiniões são formuladas empiricamente, não sendo demonstradas pelos autores. Por exemplo, é frequente afirmar-se que os mecanismos distribuídos são mais escaláveis que os centralizados, porque estes possuem um ponto de congestionamento que, eventualmente, originará perdas de desempenho quando se expandir o sistema [Shivaratri et al., 1992]. Esta é uma opinião empirica que, como se verá, não corresponde à realidade.

Mesmo quando os autores demonstram as suas afirmações, estas estão condicionadas aos sistemas sobre os quais os autores realizaram as suas experiências e ao que cada um deles considera ser um sistema escalável. Como exemplo do primeiro caso temos a demonstração efectuada por [Theimer and Lantz, 1989], onde conclui que os sistemas centralizados são mais escaláveis que os distribuídos, mas que pode ser considerada um pouco duvidosa pois o autor concebeu o seu mecanismo num ambiente de programação onde os processos ficam bloqueados sempre que enviam um mensagem até que o processo destino envie uma mensagem de resposta. Isto implica que, para actualizar a informação do estado da carga nos N nodos do sistema, sejam necessárias $2 * (N-1)$ mensagens num mecanismo centralizado e $(N-1)^2$ num mecanismo distribuído, com a vantagem evidente para o primeiro. A maior parte dos ambientes de programação distribuída, e.g. PVM [Geist et al., 1994], não obriga a este mecanismo de reposta, sendo, por isso, as conclusões tiradas específicas para este caso.

Existem autores que defendem que para um mecanismo ser escalável não necessita que o seu desempenho seja independente do tamanho do sistema, mas apenas de ser independente até a um determinado tamanho do sistema, afirmando, por isso, que os mecanismos por eles apresentados são escaláveis. Esta é a razão pela qual muitos autores afirmam que os mecanismos centralizados são escaláveis [Kremien and Kramer, 1992]. Este autor defende que a independência total do tamanho do sistema é desejável mas que a escalabilidade de um mecanismo não tem correspondência com o seu grau de distribuição, mas sim com a quantidade de informação que utiliza. Complementarmente afirma que a escalabilidade só se consegue se apenas se utilizar informação parcial e que mecanismos onde os nodos que desempenham actividades de distribuição recolhem informação acerca de todos os nodos do sistema nunca poderão ser escaláveis.

A relação entre a escalabilidade e a quantidade de informação utilizada deve ser vista de outra perspectiva. Dois dos factores mais influentes na eficiência final de um mecanismo de distribuição são, por um lado, a sobrecarga imposta no sistema pelo mecanismo de distribuição, obviamente menor quando apenas se recolhe informação parcial, e por outro, a qualidade das decisões tomadas, que é tanto melhor quanto maior for a quantidade de informação recolhida. Assim, um mecanismo pode ver o seu desempenho degradar-se, quer porque a sobrecarga sobre o sistema se torna excessiva, quer porque a distribuição deixa de ser eficiente. Um mecanismo só será ilimitadamente escalável se utilizar informação parcial e ao mesmo tempo garantir que nessa informação se encontra aquela necessária para tomar as melhores decisões. Esta é a filosofia seguida em mecanismos como os apresentados nas secções 3.3.3, 3.3.5 e 3.3.7. Mecanismos em que o nodo receptor para uma tarefa é determinado aleatoriamente ou com base em sondagens a nodos também escolhidos aleatoriamente, são normalmente escaláveis pois contrabalanceiam as decisões menos conseguidas com uma política de informação apenas baseada em dados locais ou em muito pouca informação. Estes mecanismos baseiam-se na sua simplicidade para obter desempenhos semelhantes a outros muito mais complexos, nomeadamente no tocante à recolha de informação acerca do sistema [Eager et al., 1986].

Existe, no entanto, uma razão para que os mecanismos centralizados, mesmo com recolha parcial de informação, dificilmente sejam ilimitadamente escaláveis e portanto não possam ser utilizados no sistema subjacente a este estágio. O nodo central, para além de recolhar a informação, também recebe os pedidos para distribuir tarefas de todos os nodos do sistema. Se não houver uma limitação nestes pedidos o nodo não pode ser ilimitadamente escalável. Este é o caso do mecanismo apresentado na secção 3.3.3. Uma estratégia semelhante à da política de informação poderia ser utilizada, ou seja, apenas receber pedidos dos nodos com mais necessidade de distribuição. No entanto, conforme o sistema aumenta é natural que aumentem os nodos com necessidades de distribuição, havendo um conflito evidente com a necessidade de restringir os pedidos. De todos os mecanismos apresentados apenas os apresentados nas secções 3.3.3 e 3.3.6 não cumprem minimamente os requisitos para serem escaláveis. O primeiro porque utiliza uma abordagem centralizada sem restrição de pedidos de distribuição e o segundo porque utiliza uma estratégia de informação global.

Para além das abordagens completamente distribuída e centralizada, também se podem utilizar outras intermédias, como a hierárquica e o *clustering*, que, devido a utilizarem implicitamente políticas de informação locais ou parciais, são boas candidatas na concepção de mecanismos de distribuição escaláveis. Embora não tenha apresentado nenhum mecanismo que utilize uma delas, muitos sistemas reais, e.g. [Becker, 1995], utilizam estas abordagens com sucesso. Em [Zhou, 1988] demonstra-se que não se conseguem melhorias no desempenho de um sistema de distribuição a partir de um determinado tamanho de sistema. Tendo em conta este pressuposto não interessa que um mecanismo seja escalável para além desse tamanho, afirmando o autor que se deve optar por fazer a distribuição por *clusters* sensivelmente com esse tamanho, apenas se transferindo tarefas entre *clusters* em casos que compensem fortemente. Embora o pressuposto de que não se conseguem ganhos de desempenho a partir de um determinado tamanho de sistema seja um pouco duvidoso (dado o reduzido número de mecanismos analisados nesse trabalho, no qual não está incluído, por exemplo, nenhum que utilize informação parcial ao mesmo tempo que tenta maximizar a utilidade retirada dessa informação), é natural que a eficiência de um sistema de distribuição diminua com o aumento do tamanho do sistema, e que a utilização de abordagens intermédias entre a centralização e a distribuição total seja mais compensadora. No entanto, há falta de trabalhos de investigação que validem esta afirmação.

3.4.2 Tolerância a Falhas

O grau de centralização de um mecanismo de distribuição também assume grande importância na questão da tolerância a falhas. É normalmente aceite, e ao mesmo intuitivo, que os mecanismos centralizados são mais vulneráveis devido à existência de um único nodo encarregue da distribuição e cuja falha não é tolerada pelo sistema [Shivaratri et al., 1992, Kremien and Kramer, 1992, Theimer and Lantz, 1989].

Os autores que defendem a utilização de mecanismos centralizados assumem que a tolerância se pode conseguir através da instalação de réplicas do algoritmo central de distribuição em diversos nodo que monotorizam o nodo central e que, quando detectam que este deixa de reponder, assumem o seu papel [Theimer and Lantz, 1989]. Nestas situações é necessário existir um protocolo que garanta que apenas um dos nodos fica encarregado da distribuição. Quando não se considere que a tolerância a falhas seja uma característica essencial, o possível atraso provocado por esta mudança de nodo central pode ser facilmente tolerado pelos utilizadores [Zhou, 1988]. No sistema subjacente a este estágio, devido à existência de tarefas essenciais, que tem de ser executadas dentro de um determinado prazo, este tempo de recuperação pode ser crítico e deve ser minimizado.

Outra questão importante quando se procede à mudança do nodo central é a recuperação da informação perdida. Devido à constante mutação do estado do sistema esta não é, normalmente, uma questão crítica, pois os mecanismos de distribuição apenas se baseiam no passado recente do sistema e o novo nodo recuperará rapidamente a informação necessária para funcionar plenamente [Zhou, 1988]. No entanto, se for necessária informação relativa a um passado mais distante, esta pode tornar-se numa questão crítica, passando a haver necessidade de manter a coerência entre a informação presente no nodo central e nas suas réplicas para que a recuperação a uma falha não seja crítica. Como no sistema subjacente a este estágio se prevê a existência de comércio de recursos entre nodos, poderá haver a necessidade de, por exemplo, guardar informação acerca de transações passadas para saber os nodos com que se consegue negociar mais facilmente, sendo esta informação importante em decisões futuras e, caso se utilizasse uma abordagem centralizada, deveria ser replicada.

Basicamente, o grau de centralização utilizado deverá ser ponderado em relação à simplicidade desejada para a concepção do mecanismo de distribuição [Theimer and Lantz, 1989]. É mais ou menos intuitivo que, para se conseguir um elevado grau de tolerância a falhas num mecanismo centralizado, será necessário construir um mecanismo de recuperação extremamente complicado, enquanto que essa mesma tolerância se consegue automaticamente num sistema completamente distribuído. Quando a tolerância a falhas é essencial, como no caso do sistema subjacente a este estágio, a vantagem vai claramente para os mecanismos distribuídos.

3.4.3 Intrusão

Nenhum dos mecanismos apresentados foi desenvolvido para distribuir carga entre sistemas com diferentes proprietários. Consequentemente, nenhum deles está preparado para respeitar as políticas de partilha definidas pelas organizações, nomeadamente, no que diz respeito à percentagem de tarefas remotas executadas nos seus nodos. No entanto, existem mecanismos mais ou menos predispostos para funcionar nessas condições, e cujas características são analisadas de seguida.

Mais uma vez, o grande problema é a escolha entre uma abordagem centralizada ou distribuída. Neste caso, as abordagens completamente centralizadas ficam praticamente excluídas devido ao problema de escolher onde vai ficar localizado o nodo central. Se este problema já se põe dentro de uma organização [Theimer and Lantz, 1989], entre organizações ele é ainda maior. Nenhuma organização deseja disponiblizar um dos seus nodos para gerir a distribuição pois, dadas as dimensões do sistema em questão, isso implicaria dedicar pelo menos um nodo exclusivamente a esse serviço. Poderia alegar-se que se esse serviço fosse pago então o problema estaria resolvido. No entanto, note-se que se isso acontecesse a organização onde o nodo estivesse localizado deteria o monopólio do serviço e, mais grave ainda, teria acesso a informação de todas as organizações e controlaria todas as decisões de distribuição. Tal situação muito dificilmente seria aceite pelas restantes organizações.

Dadas essas condicionantes, o grau máximo de centralização possível é quando existe um nodo central em cada organização responsável pelas actividades de distribuição da mesma. Neste caso, a distribuição entre os nodos responsáveis das organizações deverá ser feita de modo distribuído pelas mesmas razões acima apontadas. Esta abordagem corresponde à distribuição por *clusters* apresentada na secção 3.2.2.

O único trabalho consultado no domínio da intrusão, [Özden et al., 1993], chega a conclusões semelhantes, mas não considera questões monetárias e organizacionais. Apenas é considerado o problema da sobrecarga nos recursos, ou seja, considera que ninguém deseja dedicar exclusivamente alguns dos seus nodos ou das suas infra-estruturas de comunicação para a distribuição de carga. Julgo que esse trabalho sofre do mesmo problema que os trabalhos realizados no domínio da escalabilidade, que é fazer variar o grau de centralização para determinar qual a melhor abordagem para um mecanismo não ser intrusivo. Como no caso da escalabilidade, considero que essa análise deveria ser feita variando o grau de informação recolhida e de pedidos de distribuição atendidos.

Para além do grau de centralização requerido também existe outra questão fundamental para garantir a não intrusão de um mecanismo de distribuição: a sua política de localização. Podemos facilmente distinguir entre dois tipos de políticas de localização: as que decidem localmente para onde transferir uma tarefa e que impõem essas tarefas aos nodos remotos, e as que sondam primeiro esses nodos remotos para saber se querem ou não receber essa tarefa. Embora a maior parte das políticas de sondagem, o sejam apenas para saber o estado do nodo remoto, acabando por lhe impor na mesma as tarefas, podem, no entanto, facilmente ser adaptadas para permitirem a rejeição. Quanto mais não seja, o nodo remoto pode simplesmente não responder ou falsear a resposta para não receber tarefas. Nesta situação encontram-se os mecanismos apresentados nas secções 3.3.2, 3.3.4, 3.3.5 e 3.3.7. De entre os restantes, o caso mais flagrante de imposição é o mecanismo apresentado na secção 3.3.1 onde os nodos remotos são escolhidos aleatoriamente.

3.4.4 Actualidade da Informação

Na secção 2.2.1 assumui-se que os atrasos na entrega de mensagens não deveriam ser negligenciados, ou seja, que não se deve assumir que a informação recebida numa mensagem representa realmente o estado do processador que a enviou. Este cuidado é importante para se evitar, por exemplo, sobrecarregar um nodo cujo estado se tenha alterado repentinamente. Nos casos dos sistemas onde os nodos tem proprietários diferentes este cuidado deve ser redobrado pois o mecanismo de distribuição não pode ser intrusivo. Nestes sistemas deve-se evitar o uso de informação desactualizada e também de informação probabilística [Özden et al., 1993].

A única forma segura de lidar definitivamente com a informação desactualizada é permitir que um nodo rejeite uma tarefa depois de esta já lhe ter sido enviada para execução. Esta estratégia é facilmente conseguida usando uma política de localização por sondagem onde a tarefa a ser executada é enviada junto com o pedido de informação [Kremien and Kramer, 1992]. Se o nodo remoto estiver em condições de a executar pode começar logo a fazê-lo, pois já possui a tarefa, sendo essa decisão tomada em função do estado real do nodo nesse instante.

Nenhum dos outros mecanismos apresentados utiliza mecanismos para lidar com esta situação, mas entre eles existem políticas de informação que originam informação mais ou menos actualizada. A obtenção de informação actualizada implica uma contrapartida ao nível do tempo de espera das tarefas para iniciarem a sua execução. É intuitivo, que quando as decisões de distribuição se baseiam em informação contida localmente, podem ser tomadas mais rapidamente, evitando-se os tempos de espera necessários para requisitar informação aos restantes nodos [Kremien and Kramer, 1992]. Por outro lado, se a informação for requisitada na altura de decidir, normalmente quando uma tarefa é criada, o atraso vai-se reflectir no tempo de espera da mesma, sendo o mecanismo mais lento [Kremien and Kramer, 1992, Zhou, 1988].

Se dermos mais importância à transparência para o utlizador, deverão ser minimizados os tempos de espera, através da utilização de políticas de informação que garantam a existência da mesma na altura de decisão, nomeadamente, políticas de dissiminação ou não utlização de qualquer informação remota. Como exemplo temos os mecanismos das secções 3.3.1, 3.3.3 e 3.3.6. Se a prioridade for para a actualidade deverão ser utilizadas políticas de sondagem ou leilão, como as utlizadas nos restantes mecanismos apresentados. Os mecanismos mais eficazes nos dois pontos são os que utilizam mecanismos híbridos, onde se utiliza informação local para decidir quais os melhores nodos para sondar. Ver, por exemplo, os mecanismos apresentados nas secções 3.3.2, 3.3.5 e 3.3.7. As abordagens utilizadas pelos últimos são boas candidatas para o sistema subjacente a este estágio, dada a importância que se deve ter tanto na actualidade de informação como na transparência para as tarefas.

3.4.5 Estabilidade

Como já foi dito, a estabilidade de um mecanismo pode ser vista de duas perspectivas [Shivaratri et al., 1992]. Primeiro, o mecanismo deve garantir que a taxa de execução de tarefas se mantenha superior à taxa de chegada das mesmas ao sistema. Segundo, não deve haver a possibilidade de o mecanismo executar indefinidamente transferência inúteis de tarefas, ou seja, transferências para nodos sobrecarregados.

Para garantir a estabilidade em relação ao primeiro caso deve-se evitar que o mecanismo não sobrecarregue demasiado o sistema com tarefas de distribuição. Esta questão já foi analisada aquando da escalabilidade, na secção 3.4.1, sendo esta a razão pela qual se diz que a escalabilidade é uma pré-condição para a estabilidade [Kremien and Kramer, 1992]. A solução para este ponto passa, principalmente, como já tinha dito, pela recolha de informação parcial acerca do sistema.

Outra garantia necessária é realizar transferências de tarefas apenas quando realmente elas compensarem. Uma das soluções para este caso é adoptar uma política de selecção que apenas escolha tarefas com um grande ratio entre custo de processamento e custo de transferência

[Eager et al., 1986]. Por exemplo, a escolha de tarefas recentemente criadas, e que ainda não começaram a executar, permite utilizar transferências não preemptivas, muito menos custosas que quando se tem de transferir preemptivamente uma tarefa a meio da sua execução. Com a utilização de transferências não preemptivas consegue-se a maior parte dos benefícios possíveis com a distribuição e são especialmente atractivas em sistemas de ficheiros partilhados frequentes na maior parte das organizações [Kremien and Kramer, 1992]. Esta estratégia, devido à sua simplicidade e eficácia, é utlizada em praticamente todos os mecanismos de distribuição. A política de selecção pode, no entanto, ser mais restritiva. Por exemplo, no mecanismo apresentado na secção 3.3.4, apenas se transferem tarefas cujo tempo de execução se prevê superior a um determinado *threshold*. Outra política eficaz é não permitir transferir tarefas que já foram alguma fez transferidas [Kremien and Kramer, 1992]. A utilização de políticas de selecção apertadas, que reduzam a percentagem de execuções remotas, é uma pré-condição para a estabilidade de um mecanismo [Kremien and Kramer, 1992], e não reduz os benefícios da distribuição de carga, pois a maior parte destes se consegue à custa da distribuição de uma pequena percentagem de tarefas [Zhou, 1988].

A escolha de quem inicia a transferência de tarefas é outras das decisões que influencia muito a estabilidade de um mecanismo. É normalmente aceite que os mecanismos iniciados pelo receptor (como o apresentado na secção 3.3.4) são mais estáveis que os iniciados pelo nodo origem da tarefa [Shivaratri et al., 1992, Theimer and Lantz, 1989, Kremien and Kramer, 1992]. Nestes mecanismos são os nodos com pouca carga que requisitam tarefas aos restantes. Quando a carga global do sistema está muito elevada, os nodos deixam automaticamente de requisitar tarefas, uma situação vantajosa pois não se conseguiriam melhorias de desempenho com essas transferências. Consequentemente, o mecanismo de distribuição deixa de sobrecarregar o sistema, já de si sobrecarregado. Pelo contrário, nos mecanismos iniciados pela origem, os nodos transferem tarefas quando ficam sobrecarregados, ou seja, quando o sistema estiver globalmente sobrecarregado, a maior parte dos nodos estão, inutilmente, a tentar transferir tarefas, acabando o mecanismo de distribuição por sobrecarregar ainda mais o sistema. O problema com a utlização de mecanismos iniciados pelo receptor é que estes apenas são eficazes quando se utilizam transferências preemptivas [Zhou, 1988] muito mais custosas acabando, normalmente, por se perder nas transferências a eficiência que se ganhava por utlizar esta abordagem.

Outra razão para não se utilizarem mecanismos iniciados pelo receptor é a constatação de que na maior parte dos sistemas distribuídos os nodos se encontram normalmente parados, sucedendo que, ocasionalmente, alguns deles fiquem sobrecarregados [Theimer and Lantz, 1989]. Neste tipo de sistemas os mecanismos iniciados pela origem são mais eficientes porque não impõem tanta sobrecarga quando o sistema está com carga baixa (nesta situação os mecanismos iniciados pelo receptor estão a tentar obter tarefas, sobrecarregando o sistema) e porque reagem mais depressa a existência de um pico de carga num dos nodos (ele próprio encarrega-se de distribuir a carga em excesso, enquanto que no outro caso é necessário que um dos receptores descubra primeiro que o nodo está sobracarregado). Este último ponto é muito mais crítico, pois no primeiro caso, como o sistema está com pouca carga, o facto de o mecanismo de distribuição sobrecarregar um pouco o sistema não é determinante. Mais uma vez as soluções intermédias são as mais eficazes. Com a utilização de abordagens adaptativas entre iniciação pela origem e pelo receptor, como a utilizada no mecanismo apresentado na secção 3.3.2, consegue-se facilmente obter o melhor das duas estratégias.

Para garantir a estabilidade de um mecanismo de distribuição em relação ao segundo ponto, ou seja, impedir que haja probabilidade de se efectuarem indefinidamente transferência inúteis de tarefas é necessário haver o cuidado de se utilizar informação actualizada nas decisões de distribuição, para minimizar a possibilidade de transferir uma tarefa para um nodo sobrecarregado que depois a rejeita, enviando-a de volta para o nodo origem (dando origem ao efeito "ping-pong") ou que a redistribuí, caso seja permitido, para outro nodo qualquer, podendo originar que uma tarefa esteja continuamente a ser transferida sem nunca ser executada. As implicações necessárias para se garantir a actualidade da informação já foram apresentadas na secção 3.4.4.

3.4.6 Comércio Electrónico

Nenhum dos mecanismos apresentados e todos os consultados em artigos de distribuição de carga, prevê o pagamento pela utilização de recursos em nodos remotos. Aliás, este é o maior problema apresentado pelas estratégias convencionais de distribuição de carga quando utilizadas no contexto do sistema subjacente a este estágio. Poderia alegar-se que facilmente se poderia implementar um mecanismo de taxação à posteriori em qualquer mecanismo de distribuição, por exemplo, através do registo da ocupação dos recursos por outras organizações e da cobrança periódica dessa utilização. No entanto, com este tipo de abordagem, a política de localização de um nodo remoto não considera explicitamente os custos de execução da tarefa nesse nodo, não actuando como um verdadeiro agente económico na procura da melhor relação preço / qualidade, ou mais concretamente, do nodo que oferece melhor relação entre os recursos oferecidos e o preço desses recursos. Dado o objectivo de integrar o mecanismo de distribuição numa verdadeira organização comercial, apenas interessam os mecanismos que consideram explicitamente os custos de execução, e, portanto, nenhum dos mecanismos conhecidos pode ser utilizado neste caso.

As abordagens que mais facilmente se adaptam à utilização num sistema comercial são as de sondagem ou leilão, pois as respostas aos pedidos de execução de uma tarefa podem incluir facilmente os custos dessa execução, cabendo ao nodo que originou o pedido decidir qual o nodo onde vai executar essa tarefa. Como a utilização destas abordagens já é recomendada dada a necessidade de não intrusão, elas são excelentes candidatas a ser utilizadas num sistema com as características apresentadas.

3.4.7 Nodos de Processamento Heterogéneos

Normalmente, os mecanismos de distribuição não consideram a heterogeneidade dos elementos de processamento, e quando o consideram fazem-no apenas em relação ao CPU. Esta abordagem baseia-se na constatação de que o CPU é o recurso com mais contenção [Zhou, 1988], e, portanto, o único que deve ser considerado pelas actividades de distribução. No entanto, se apenas se pretende diferenciar os nodos em relação à sua capacidade de CPU, e se assumirmos que as tarefas não têm prazos de execução, não são necessárias grandes alterações aos mecanismos já desenvolvidos, que podem ser concebidos assumindo um sistema homogéneo, sendo depois apenas necessário escalar as medidas de carga em relação às capacidades dos diversos processadores [Eager et al., 1986].

Genericamente, se considerarmos que em todos os nodos existem os recursos necessários para executar qualquer tarefa e que as tarefas não têm prazos de execução definidos, não há necessidade de considerar explicitamente os recursos presentes em cada nodo. No entanto, esta não é a situação que se verifica nos sistemas reais, onde se constata que existem tarefas que não podem ser executadas em alguns nodos, quer porque neles não existem os tipos de recursos que necessitam, quer porque neles não existem as quantidades desses recursos necessárias. Por exemplo, uma tarefa pode necessitar de um ficheiro que apenas existe na sua rede local ou de uma grande quantidade de memória, apenas disponível em algumas máquinas e logo não é transferivel para qualquer nodo do sistema. Em [Zhou, 1988] já se tinha alertado para esta situação no contexto dos mecanismos de distribuição genéricos, mas apenas em relação aos casos de tarefas que tem necessariamente de ser executadas nos nodos de origem, aliás, a situação mais frequente.

A alegação de que, devido à não especificação dos recursos necessários pelas tarefas, não se justifica a sobrecarga por se considerar os diferentes recursos existentes em cada nodo, não deve ser considerada válida devido à possibilidade de o sistema aprender com as sucessivas execuções de uma tarefa as suas necessidades de recursos. No sistema subjacente a este estágio, essa alegação é ainda menos válida, pois também se pretendem abranger as aplicações que interagem em tempo real, que, devido aos prazos de execução que apresentam, devem especificar minimamente os recursos de que necessitam. Num sistema destas características devem-se considerar explicitamente os recursos que cada nodo possui, não só em qualidade mas também em quantidade, para se garantir que uma tarefa é alocada a um nodo que lhe pode fornecer as quantidades de recursos que necessita. Para além disso, os mecanismos de distribuição devem possuir formas de reservar os recursos necessários por uma tarefa com prazo de execução, para impedir que outras tarefas utilizem esses recursos, com o consequente atraso em relação ao prazo estabelecido.

A existência de diversos recursos que devem ser considerados também implica que se deixe de utilizar apenas uma medida de carga relativa à utilização do CPU. Embora em [Kunz, 1991] se tenha demonstrado que o número de tarefas em execução é a medida de carga mais bem sucedida em mecanismos de distribuição, na medida em que reflete melhor o estado do CPU como recurso com mais contenção, e que combinações com outras medidas não são susceptíveis de melhorar o desempenho destes mecanismos, tal demonstração foi feita tendo por base o pressuposto de que as tarefas podem executar em qualquer nodo e, consequentemente, não se aplica neste caso. A especificação por parte das tarefas dos recursos necessários obriga a que se conheça o estado dos nodos em relação a todos os recursos e não apenas em relação ao CPU, o que implica que se utilizem várias medidas de carga para caracterizar um nodo do sistema. Similarmente, também se deve adoptar uma medida que caracterize o estado das linhas de interconexão entre os nodos.

Devido às necessidades das aplicações que interagem em tempo real, os mecanismos de distribuição desenvolvidos para sistemas que suportam estas aplicações são os únicos que, normalmente, consideram explicitamente os recursos de cada nodo de processamento. Este é o caso também dos mecanismos apresentados, onde o único que cumpre este requisito é o mecanismos apresentado na secção 3.3.7 (desenvolvido especificamente para sistemas que suportam interacção em tempo real). No entanto, mesmo neste mecanismo, não se consideram as infra-estruturas de comunicação como um dos recursos limitados do sistema, que também deve ser sujeito a reserva, nomeadamente, quando uma tarefa com prazo de execução precisa de comunicar com outra para conseguir terminar. Mesmo num sistema de carácter genérico, e pelas razões já apontadas, devem-se considerar os diferentes recursos de cada nodo, para garantir que todas as tarefas são correctamente executadas.

3.4.8 Carga Heterogénea

Quando se fala em carga heterogénea, normalmente apenas se refere a diferença entre os tempos médios de execução das tarefas em diferentes nodos de processamento, ou seja, um sistema com carga heterogénea é aquele sistema em que existem nodos continuamente mais sobrecarregados do que outros. Este tipo de carga heterogénea é típica de redes de estações de trabalho [Kremien et al., 1993, Theimer and Lantz, 1989, Zhou, 1988, Shivaratri et al., 1992], mas nem sempre foi considerada nos mecanismos de distribuição. Por exemplo, trabalhos como [Eager et al., 1986] assumem erradamente que a carga do sistema é homogénea a longo prazo, sendo o objectivo da distribuição de carga lidar com as diferenças de carga a curto prazo. Este pressuposto está na origem de muitos mecanismos de distribuição onde a política de localização escolhe aleatoriamente os nodos remotos para onde enviar uma tarefa ou para sondar. Efectivamente, se a carga de todos os nodos for idêntica a longo prazo, a probabilidade de se encontrar um nodo com pouca carga também é idêntica para todos os nodos o que garante que, por exemplo, um mecanismo por sondagem, com escolha aleatória dos nodos a sondar, não necessite de efectuar muitos pedidos para encontrar um nodo com pouca carga [Eager et al., 1986]. No entanto, num sistema com carga heterogénea as probabilidades de um nodo estar com pouca carga são diferentes para cada nodo do sistema e ao longo do tempo, provocando que os mecanismos com componentes de localização aleatórias, como os das secções 3.3.1 e 3.3.4, não sejam tão bem sucedidos em sistemas reais. A política de localização de um mecanismo de distribuição para um sistema com carga heterogénea deve considerar nas suas decisões alguma informação relativa ao estado do sistema.

Existe, no entanto, outra forma de heterogeneidade menos referida, que é o facto de cada tarefa poder necessitar de recursos diferentes e em quantidades diferentes, e o facto de haver tarefas que especificam esses recursos e outras, a maior parte, que não o fazem. O problema dos diferentes recursos já foi abordado na secção 3.4.7, chegando-se à conclusão que é necessário considerar explicitamente os recursos de cada nodo e haver mecanismos de reserva para esses recursos. No entanto, um mecanismo de distribuição genérico não deve assumir o conhecimento dos recursos das tarefas à priori [Kunz, 1991]. Dadas as características do sistema subjacente a este estágio, um mecanismo de distribuição de carga desenvolvido para esse sistema deve suportar indiscriminadamente tarefas que especifiquem os seus recursos e tarefas que não o façam ou apenas o façam parcialmente. Esta abordagem flexível não é adoptada por nenhum dos mecanismos apresentados, nem mesmo pelo mecanismo apresentado na secção 3.3.7 que também não considera as necessidades de comunicações das tarefas.

3.4.9 Objectivos da Distribuição

Um dos grandes problemas dos mecanismos de distribuição de carga convencionais é considerarem explicitamente apenas um objectivo nas suas decisões. Normalmente, o objectivo mais considerado é o de minimizar os tempos médios de resposta das tarefas, visto ser o objectivo que melhor se aplica em sistemas distribuídos genéricos [Kunz, 1991]. No entanto, no contexto do sistema subjacente a este estágio existem mais dois objectivos que um mecanismo de distribuição deve considerar em simultâneo:

- a maximização do número de tarefas executadas dentro dos prazos especificados e com os recursos requisitados; e

- a maximização do lucro da organização, através da minimização dos custos e maximização dos proveitos inerentes às tarefas de distribuição de carga.

Devido à existência de diversos objectivos também é necessário existir a possibilidade de se definirem explicitamente prioridades e relações de troca entre eles.

Outra questão importante, relacionada com o objectivo de um mecanismo de distribuição de carga, diz respeito à integração das máquinas paralelas e de sistemas distribuídos compostos por redes de estações de trabalho. Uma das grandes diferenças que caracteriza os mecanismos de distribuição de carga para estes dois sistemas é precisamente o objectivo que perseguem. Ao contrário dos sistemas distribuídos, onde genericamente se procura diminuir o tempo de execução médio das tarefas, nas máquinas paralelas esse objectivo é, normalmente, equalizar a carga de todos os nodos de processamento disponíveis. Na realidade, este último apenas consiste num caso particular do primeiro, ou mais concretamente, consiste num objectivo que já indica como se deve atingir a diminuição do tempo médio de execução. De facto, dadas as características das máquinas paralelas, nomeadamente, das suas infra-estruturas de comunicação muito rápidas, e da carga a que normalmente estão sujeitas, constituída por aplicações muito longas, que garantem a existência de tarefas para todos os nodos, esse objectivo representa efectivamente uma forma eficaz de se atingir a diminuição do tempo médio das aplicações.

Num sistema distribuído normal esse objectivo não poderia ser utilizado, pois caracterizam-se por estarem sujeitos a uma carga baixa na maior parte dos nodos, ocasionalmente interrompida por um pico de carga num deles. A tentativa de equalização ou de balanceamento de carga neste caso provocaria piores resultados, pois as transferências de tarefas necessárias, muito mais custosas que nas máquinas paralelas, acabariam por contrariar os efeitos benéficos que se poderiam atingir [Kremien and Kramer, 1992, Theimer and Lantz, 1989].

Acho que, sendo o objectivo de equalização da carga um caso particular do objectivo mais genérico de diminuição do tempo médio de resposta das tarefas, deve-se usar apenas este último num sistemas que pretenda integrar indiscriminadamente máquinas paralelas e sistemas distribuídos vulgares. Cabe ao mecanismo de distribuição decidir qual a estratégia a adoptar para atingir esse objectivo. Esta decisão deverá levar em consideração as diferenças que caracterizam os diversos componentes do sistema e a carga a que está ser sujeito em cada momento, adoptando a melhor estratégia para esse caso. Eventualmente, se a carga for muito elevada e as capacidades de comunicação elevadas, poderá adoptar-se uma estratégia de equalização.

3.4.10 Resumo

Resumindo, os maiores problemas que caracterizam os mecanismos de distribuição convencionais estudados quando aplicados a um sistema distribuído multi-organizacional são os seguintes:

- nenhum possibilita o comércio dos recursos existentes numa organização;
- nenhum garante integralmente a não intrusão;
- nenhum integra indiscriminadamente tarefas que especifiquem os recursos necessários, tarefas sem essa especificação ou com uma especificação parcial dos mesmos e nenhum considera as necessidades de comunicação entre elas;
- a maior parte não considera explicitamente os diferentes recursos existentes nos nodos e nenhum considera a capacidade da infra-estrutura de comunicações; e

- nenhum considera explicitamente a possibilidade de existirem vários objectivos a atingir com a distribuição de carga.

De entre as diversas estratégias utilizadas pelos mecanismos apresentados, as que parecem mais promissoras para aplicar neste sistema são as seguintes:

- recolha de informação parcial acerca do sistema, tentando garantir que nela se incluam os nodos com melhores características para receberem tarefas;
- utilização de um mecanismo completamente distribuído ou organizado por *clusters*;
- utilização de políticas de localização por sondagem ou leilão em oposição às que impõem as tarefas aos nodos remotos; e
- utilização de mecanismos adaptativos.

Estas conclusões foram tiradas considerando os trabalhos consultados, não se excluindo a hipótese de que exista algum mecanismo que cubra os problemas mencionados. No entanto, considero que as referências consultadas constituem uma base suficientemente alargada e diversificada e que a maior parte das conclusões tiradas são válidas para a distribuição de carga em geral.

Chapter 4

Abordagem Multi-Agente

A distribuição de carga é, essencialmente, um problema de alocação de recursos. Dados o modelo do sistema distribuído físico, onde estão explicitas as características dos recursos existentes no sistema, e o modelo do sistema distribuído lógico, que caracteriza as tarefas e suas interdependências, a distribuição de carga resume-se ao problema da alocação, ou seja, do mapeamento sistemático do sistema lógico no sistema físico [Kremien and Kramer, 1992].

O planeamento distribuído da alocação de recursos constituí uma das principais áreas de interesse na concepção de aplicações multi-agente [Lesser, 1995]. A disperção espacial dos recursos, a necessidade de reacção em tempo real e o aumento da robustez do sistema, motivam a distribuição do controlo por um conjunto de agentes cooperativos [Sandholm and Lesser, 1995b]. Estes coordenam as suas actividades para evitar conflitos na utilização dos recursos e para maximizar a taxa de execução das tarefas. Este problema de alocação, evidente quando consideramos os recursos internos de uma organização, também se coloca quando existe comércio electrónico de recursos entre diferentes organizações, sendo necessário, neste caso, providenciar os agentes com mecanismos para contratação automática.

O principal problema associado à concepção de um sistema multi-agente para distribuição de carga num sistema multi-organizacional é a necessidade dos agentes assumirem um comportamento misto: benevolente quando se relacionam com agentes da mesma organização e egoísta em contratos com outras organizações [Sandholm and Lesser, 1995b]. Embora se tenham desenvolvido muitas estratégias de cooperação, quer para agentes benevolentes quer para agentes egoístas, não temos conhecimento de estratégias que integrem simultaneamente os dois comportamentos. Dada a necessidade da sua existência, foi concebida uma nova estratégia de cooperação com essas características.

A estratégia desenvolvida não é especifica para o problema da distribuição de carga, podendo ser aplicada em qualquer problema de alocação de recursos num sistema multi-organizacional. Esta generalidade torna-se especialmente atractiva ao constatarmos que a maior parte dos problemas resolvidos computacionalmente podem ser modelados como um problema de alocação de recursos [Wellman, 1995].

4.1 Porquê um Sistema Multi-Agente?

Abstraíndo-nos da noção de agente, comecemos por explicar o porquê da necessidade de existirem diversas entidades de controlo. A primeira razão deriva do facto de existirem diversas organizações. O controlo centralizado da alocação de recursos implicaria a existência de uma entidade imparcial à qual as organizações delegariam a responsabilidade de zelar pelos seus interesses. Esta imparcialidade numa sociedade puramente comercial é, como se pode constatar na vida real, impossível de se conseguir. Assim, a primeira distribuição do controlo surge ao nível organizacional, ou seja, deverá existir pelo menos uma entidade por cada organização que controle a alocação de recursos dessa organização.

Dentro de cada organização também existe a necessidade de distribuir o controlo por diversas entidades. Algumas razões para essa distribuição são:

- **Capacidade de comunicações limitada**. Um problema de planeamento distribuído, como é o da alocação de recursos, origina, normalmente, um problema de interpretação distribuída, devido à necessidade de saber o estado dos diversos nodos do sistema [Durfee et al., 1989]. Estando estes nodos distribuídos espacialmente, a existência de um controlador central levaria à necessidade de existir uma infra-estrutura de comunicação com capacidade suficiente para se proceder à recolha de toda a informação em tempo útil. Sendo as capacidades de comunicação bastante limitadas, quando comparadas com o tamanho das organizações envolvidadas, torna-se necessário distribuír o controlo por diversas entidades distribuídas espacialmente.

- **Maior rapidez**. O planeamento da utilização dos recursos é uma actividade que também consome recursos, estando, consequentemente, a sua rapidez condicionada às características dos mesmos. Se este planeamento for distribuído poderemos ter as diversas entidades a utilizar diferentes recursos, aumentando-se a rapidez do próprio processo de planeamento devido à sua paralelização. Esta razão é ainda mais importante quando a alocação tem de se realizar em tempo-real, como no caso da distribuição de carga.

- **Maior tolerância a falhas**. Garantir a robustez num sistema centralizado é muito mais dificil do que num sistema distribuído onde é automaticamente conseguida.
- **Maior reactividade**. Estando o planeamento distribuído as respostas aos pedidos de execução de tarefas são obtidas mais rapidamente. A maior proximidade entre a origem da tarefa e a entidade de planeamento leva a uma menor latência na transmissão da informação e, consequentemente, a respostas mais rápidas [Sandholm and Lesser, 1995b].
- **Expansibilidade**. Sendo o planeamento distribuído, torna-se mais simples a inclusão de novos recursos na organização, sendo suficiente atribuir o controlo desses recursos a uma nova entidade de planeamento [Smith, 1980].

O facto de algumas destas razões serem equivalentes às apresentadas a favor da utilização de estratégias distribuídas para distribuição de carga no capítulo 3, apenas reforça o relacionamento entre a área de distribuição de carga e a área mais genérica de alocação de recursos.

Esclarecida a necessidade da utilização de multiplas entidades de controlo, falta justificar porque decidimos aplicar tecnologia de agentes à concepção dessas entidades. Embora a utilização de agentes como paradigma de programação esteja a crescer rapidamente, um dos problemas que ainda ocupa muito tempo aos investigadores da área é a definição do próprio conceito de agente. A maior parte das definições apresentadas são muito simples e genéricas para poderem abarcar toda a flexibilidade do paradigma. Muitos definem um agente como sendo simplesmente uma entidade de software capaz de executar determinadas tarefas. Outros, definem-no como sendo uma entidade capaz de simular o comportamento humano na execução de alguma tarefa.

Uma definição mais elaborada apresenta um agente como sendo um programa autónomo, capaz de controlar as suas próprias decisões e acções, baseado na sua percepção do ambiente que o rodeia e que tenta atingir um determinado conjunto de objectivos [Jennings and Wooldridge, 1996]. No entanto, esta definição ainda não permite uma clara diferenciação entre este paradigma e outros mais tradicionais, principalmente a programação distribuída e a programação orientada por objectos. Aliás, só com esta definição praticamente todas as estratégias de distribuição de carga apresentadas no capítulo 3 poderiam ser caracterizadas como estratégias baseadas em agentes.

Para reforçar a singularidade deste paradigma de programação, [Jennings and Wooldridge, 1996] apresenta um conjunto de características comportamentais que um programa deve minimamente apresentar para poder ser caracterizado como um agente. Estas características "humanizam" o comportamento do agente, dando crédito aos investigadores que o apresentam como uma tentativa de reprodução do comportamento humano. Apresentamos agora estas características, descrevendo em que medida elas são desejáveis para a resolução de um problema de alocação de recursos num sistema multi-organizacional:

- **Autonomia**. Um agente deve ser capaz de executar a maioria das suas tarefas sem a intervenção directa de humanos ou outros agentes, e deve ter um determinado grau de controlo sobre as suas acções e estado interno. A capacidade de comunicação limitada, apontada como uma das razões para a existência de diversas entidades de controlo, volta a ser apresentada para justificar a necessidade de autonomia no funcionamento dessas entidades. A resolução independente da maior parte dos problemas e a utilização de comunicação selectiva usando uma semântica expressiva permitem reduzir a dependência da infra-estrutura de comunicações. A independência do controlo humano é absolutamente necessária, fazendo parte da especificação inicial do problema. Outras características do ambiente, como a possibilidade de falhas técnicas e o seu carácter dinâmico, também justificam a necessidade de autonomia como forma de garantir o funcionamento do sistema em situações de instabilidade e transição.

- **Sociabilidade**. Um agente deve ser capaz, quando apropriado, de interagir com outros agentes e humanos como forma de avançar o estado da resolução dos seus problemas e para ajudar outros agentes com as suas actividades. A sociabilidade é necessária a dois níveis [Sandholm and Lesser, 1995b]: dentro da organização, onde o agente deve agir benevolamente para com os outros agentes como forma de se conseguir uma melhor utilização dos recursos disponíveis; e no relacionamento com outras organizações, onde o agente deve assumir um comportamento orientado pelos interesses da sua organização, respeitando, no entanto, as regras da sociedade económica onde está inserido. É principalmente a necessidade de interagir numa sociedade comercial, onde existe a movimentação de capitais reais, que implica um grande salto na complexidade comportamental dos programas de software e que marca uma grande diferença entre esta abordagem e as apresentadas anteriormente.

- **Capacidade de reacção**. Um agente deve monitorizar o ambiente que o rodeia e responder atempadamente a mudanças que nele ocorram. Esta característica é fundamental num ambiente altamente dinâmico, caracterizado por mudanças nas quantidades de recursos disponíveis, pelo aparecimento / desaparecimento de organizações, pelas variações ao nível da carga de tarefas para executar ou por mudanças nos preços propostos para os recursos necessários. O comportamento de uma entidade responsável pela alocação de recursos deve adaptar-se às caracteristicas transitórias do ambiente. Já no capítulo 3 se tinha apresentado a necessidade desta adaptatividade em sistemas de distribuição de carga dentro de uma organização, sendo portanto mais necessária num ambiente de complexidade acrescida pelo aparecimento de diversas organizações.

- **Oportunismo**. Um agente não deve reagir apenas em resposta a uma mudança no ambiente, mas deve apresentar um comportamento oportunístico, orientado pelos seus objectivos, devendo tomar a iniciativa quando apropriado. A capacidade de aproveitar correctamente oportunidades dita o sucesso de um gestor nos competitivos mercados reais. A necessidade de incorporar esta característica numa entidade de software destinada a operar num mercado económico artificial é, por razões similares, absolutamente necessária.

Com estas características julgamos ter tornado clara a validade da aplicação da tecnologia de agentes, nomeadamente de sistemas multi-agente, à resolução de problemas de alocação de recursos e, particularmente, do problema da distribuição de carga num sistema multi-organizacional. Grande parte das razões apresentadas são compativeis com as apresentadas por muitos investigadores da área de multi-agentes quando justificam a utilização desta tecnologia para aplicações de alocação de recursos, e.g. [Durfee et al., 1989, Lesser, 1995], ou de comércio electrónico e contratação automática, e.g. [Sandholm and Lesser, 1995b, Jennings and Wooldridge, 1996]. Mesmo ao nível da distribuição de carga, este trabalho não é inovador quanto à utilização da tecnologia de multi-agentes: ver por exemplo [Shaerf et al., 1995].

4.2 Estratégias de Cooperação

Para se conseguir uma resolução efectiva de problemas usando um sistema multi-agente é necessário definir uma estratégia de cooperação entre as diversas entidades. A necessidade de cooperação nestes sistemas surge porque os agentes estão a resolver sub-problemas que são interdependentes, quer devido à contenção no acesso aos recursos, quer devido à existência de relações entre os objectivos dos sub-problemas [Lesser, 1995]. Mais especificamente, no caso particular da alocação de recursos num sistema multi-organizacional, as razões para os agentes de uma mesma organização cooperarem são:

- **A inexistência de recursos suficientes num agente**. A um dado instante um agente pode não conseguir alocar os recursos necessários a uma tarefa porque os recursos que controla estão ocupados. Este agente necessita de cooperar com outros agentes para descobrir qual deles pode executar essa tarefa. Mesmo que o agente consiga executar a tarefa, ela pode ser executada mais eficientemente por outro agente.
- **A necessidade de maximizar a utilidade colectiva**. Cada agente resolve parte do problema de alocação, não possuindo uma visão completa do estado do sistema. Como nem sempre os interesses individuais correspondem aos interesses colectivos, é necessária a cooperação para obter a informação que permita a cada agente compreender o estado global do problema e, consequentemente, tomar as melhores decisões para evuluir para um novo estado.

As razões para a cooperação entre agentes de diferentes organizações são:

- **A inexistência de recursos suficientes numa organização**. Se após cooperarem internamente, os agentes de uma organização concluem que não conseguem executar a tarefa, então podem cooperar com agentes de outras organizações com o intuito de saber se esses a conseguem executar. Neste caso, a cooperação toma a forma de negociação dada a necessidade de se acordar quanto à retribuição monetária a atribuir pela execução da tarefa. Como sucedia internamente, mesmo que uma organização consiga executar uma tarefa, pode compensar executá-la noutra organização.
- **Pode existir interesse em adoptar uma estratégia de mercado comum**. Uma organização pode ter uma atitude competitiva em relação a outra organização, passando a maximização da sua utilidade pela diminuição da utilidade da outra [Sandholm and Lesser, 1995b]. Se outra organização partilhar essa atitude então poderão associar-se para conseguirem uma estratégia de competição mais eficaz. Noutra prespectiva, poderemos ter associações, denominadas por "carteis", formadas com o objectivo de manipular os preços de determinados recursos [Pindyck and Rubinfeld, 1992].

A definição de uma estratégia de cooperação levanta algumas questões, nomeadamente definir [Lesser, 1995]: a forma como os agentes vão coordenar a resolução dos seus sub-problemas; os mecanismos de negociação necessários para que um conjunto de agentes atinjam um estado aceitável; técnicas para deteção e resolução de conflitos; protocolos de comunicação entre os agentes; e mecanismos que garantam que os agentes podem manter a sua autonomia ao mesmo tempo que contribuem para a melhoria do estado global do sistema.

Existem duas abordagens para a concepção de estratégias destas cooperação [Durfee, 1992]:

- **Partilha de tarefas**. Quando um agente possui uma tarefa que não consegue executar tenta alocar essa tarefa a outros agentes. Geralmente, as estratégias que seguem esta abordagem baseiam-se em esquemas de contratação para alocar as tarefas. Um dos trabalhos pioneiros nesta área é o *Contract Net protocol* [Smith, 1980]. Nesta estratégia os anúncios de contratos são feitos quer por agentes com tarefas que desejam alocar a outros agentes, quer por agentes que desejam que lhes sejam atribuídas tarefas. O agente que anuncia um contrato escolhe de entre todas as ofertas a mais satisfatória, atribuíndo ao agente que a propôs a tarefa. A grande diferença em relação as estratégias de leilão, frequentes em distribuição de carga, é que o acordo final é estabelecido por selecção mútua, não havendo imposição de tarefas. Esta é uma abordagem *top-down* à resolução distribuída de problemas [Durfee et al., 1989].

- **Partilha de resultados**. Os agentes partilham resultados finais e parciais com o objectivo de determinarem a solução final para o problema. A partilha de tentativas de resultados parciais leva a um melhor conhecimento do estado da resolução do problema, permitindo aos agentes detectar interdependências entre as tarefas em execução e incompatibilidades entre os resultados apresentados. Um dos exemplos mais conhecidos da utilização desta abordagem é a estratégia *functionally accurate, cooperative (FA/C)* [Durfee et al., 1987] para a resolução distribuída de problemas. Esta abordagem implementa uma filosofia *bottom-up* [Durfee et al., 1989].

Se todos os agentes do sistema tivessem conhecimento dos objectivos, acções e interações de todos os outros agentes, seria possivel saber o que cada agente estaria a fazer em cada momento e o que pretenderia fazer no futuro. Nestas condições poderiam-se evitar todos os conflitos e esforços desnecessários e a cooperação seria perfeita [Jennings, 1996]. Infelizmente, a obtenção deste conhecimento não é possivel devido à racionalidade limitada dos agentes. Esta limitação deriva de [Lesser, 1995]:

- existir uma largura de banda de comunicação limitada, que, normalmente, impede a comunicação de toda a informação necessária;
- os agentes possuirem uma capacidade de processamento limitada, que limita o número de operações de *packing* / *unpacking* de mensagens e impede o assimilamento de toda a informação comunicada;
- os agentes serem heterogéneos, o que dificulta a partilha de informação;
- existirem diferentes organizações, cujos agentes agem de acordo com os seus interesses particulares, o que pode levar ao desejo de não partilhar informação;
- o ambiente ter um carácter dinâmico que dificulta a manutenção da actualidade da informação e a previsão de acções futuras.

Como forma de contornar esta racionalidade limitada, as estratégias básicas de cooperação (partilha de tarefas ou partilha de resultados) são normalmente complementadas com outras estratégias, de entre as quais destacamos as seguintes:

- **Estruturação organizacional**. Consiste na atribuição a cada agente de um determinado papel e padrão de comunicação por forma a organizar a resolução de um problema. Esta estruturação reduz a complexidade dos agentes pois limita o espectro das decisões de controlo que podem tomar. Esta atribuição não deve ser demasiado rígida por forma a não limitar demasiado a latitude de decisão dos agentes, necessária para reagir a situações inesperadas, nem deve ser fixa durante toda a actividade do sistema [Lesser, 1995].
- **Controlo local sofisticado**. Quanto mais sofisticado for um agente acerca dos seus planos para resolução de problemas, do estado das suas suposições, e das implicações das suas acções nos planos e suposições dos outros agentes, mais fácil é conseguir uma cooperação efectiva entre os agentes [Lesser, 1990]. Esta sofisticação só se consegue integrando a compreensão das acções e suposições dos outros agentes com a compreensão da resolução local do problema, de tal forma que as decisões de coordenação façam parte das decisões locais e não constituam uma camada separada, acima da resolução local do problema [Durfee et al., 1989].

- **Técnicas teóricas**. A utilização de modelos formais do comportamento permite derivar as consequências das acções de um agente no comportamento dos outros agentes, a influência que as alterações no meio-ambiente tem nos agentes, deduzir as intenções de um agente a partir do seu comportamento ou a utilização de técnicas de cooperação sem recurso a comunicação. A aplicabilidade destes modelos é, actualmente, bastante limitada, pois é necessário recorrer a simplificações acerca dos agentes e do seu conhecimento para permitir a sua utilização em tempo útil. Por exemplo, a maior parte dos modelos concebidos até agora não assumem a racionalidade limitada dos agentes.

- **Utilização de convenções**. A utilização de convenções, leis a que todos os agentes obedecem, para ditar a forma como os agentes interagem serve, similarmente à estruturação organizacional, para limitar o espectro das decisões que se podem tomar, reduzindo a complexidade dos agentes. A grande vantagem das convenções reside na sua utilidade para guiar o comportamento dos agentes em situações de grande instabilidade e incerteza no ambiente que os rodeia [Jennings, 1996]. As convenções constituem uma espécie de protocolo para interações sem comunicação [Durfee, 1992].

A abordagem utilizada na concepção da estratégia de cooperação do sistema multi-agente para alocação de recursos num ambiente multi-empresarial foi a partilha de tarefas. Com efeito, esta abordagem é uma forma natural de se atingirem os objectivos da cooperação, através da distribuição de tarefas para aumentar o paralelismo de execução e maximizar a utilização dos recursos disponiveis [Durfee et al., 1989]. A utilização de uma abordagem de partilha de recursos não se justifica, nem é viável, neste caso porque:

- as tarefas constituem entidades indivisiveis cuja execução não dá origem a quaiquer resultados parciais;

- as tarefas são independentes entre si, não se justificando a procura de interdependências entre as mesmas que caracteriza essa abordagem;

- a partilha de resultados entre agentes de diferentes organizações, com interesses distinctos e particulares, não é viável.

Seguindo o exemplo de praticamente todas as estratégias desenvolvidas segundo esta abordagem será usada uma variação do *Contract Net protocol*, sendo utilizadas algumas das técnicas acima referidas como forma de contornar a racionalidade limitada dos agentes.

4.3 Arquitectura do Sistema

Quando se propõe uma estratégia de cooperação entre agentes é necessário referir três aspectos fundamentais [Genesereth and Ketchpel, 1994]:

- a arquitectura do sistema;
- o protocolo de comunicação; e
- a estrutura interna dos agentes.

Destes três aspectos, o protocolo de comunicação entre agentes de diferentes organizações é o unico que deve ser cumprido para se atingir uma cooperação efectiva. Quanto aos restantes aspectos, cada organização é livre para adoptar a estratégia que entender, desde que o protocolo de comunicação seja cumprido. No entanto, neste estágio é proposta uma solução integrada de cooperação, suficientemente genérica para ser aplicada em qualquer problema de alocação automática de recursos, e suficientemente flexivel para se adaptar às necessidades particulares de cada organização.

Assumindo-se que uma organização é constituída por um conjunto de nodos e um conjunto de linhas de comunicações. Em cada nodo concentram-se um conjunto de recursos que são utilizados para executar as tarefas. As tarefas também são originadas pelos nodos. As linhas de comunicações servem para transferir tarefas entre nodos e para encaminhamento das mensagens originadas por essas tarefas. Este modelo, característico dos sistemas distribuídos, pode ser aplicado a outros problemas de alocação de recursos. Por exemplo, um sistema de produção organizado por células de fabrico também pode ser visto como uma rede, onde cada nodo corresponde a uma célula, as linhas correspondem aos veículos para transporte, as mensagens às matérias primas e produtos finais e as tarefas a actos produtivos. Para simplificar, vamos assumir que todos os nodos são capazes de executar os agentes a seguir apresentados, e que as linhas de comunicação são capazes de transmitir as mensagens trocadas por estes agentes. No caso do exemplo apresentado, seria necessário que em cada célula de fabrico existisse um computador e entre estes uma rede local para transmissão de mensagens. Embora tenha ficado claro a aplicação do modelo a outros casos de alocação de recursos, a partir de agora apresentaremos como exemplo o caso da distribuição de carga. Um exemplo de um sistema multi-organizacional é apresentado na figura 4.1.

Na arquitectura desenvolvida existem dois tipos de agentes:

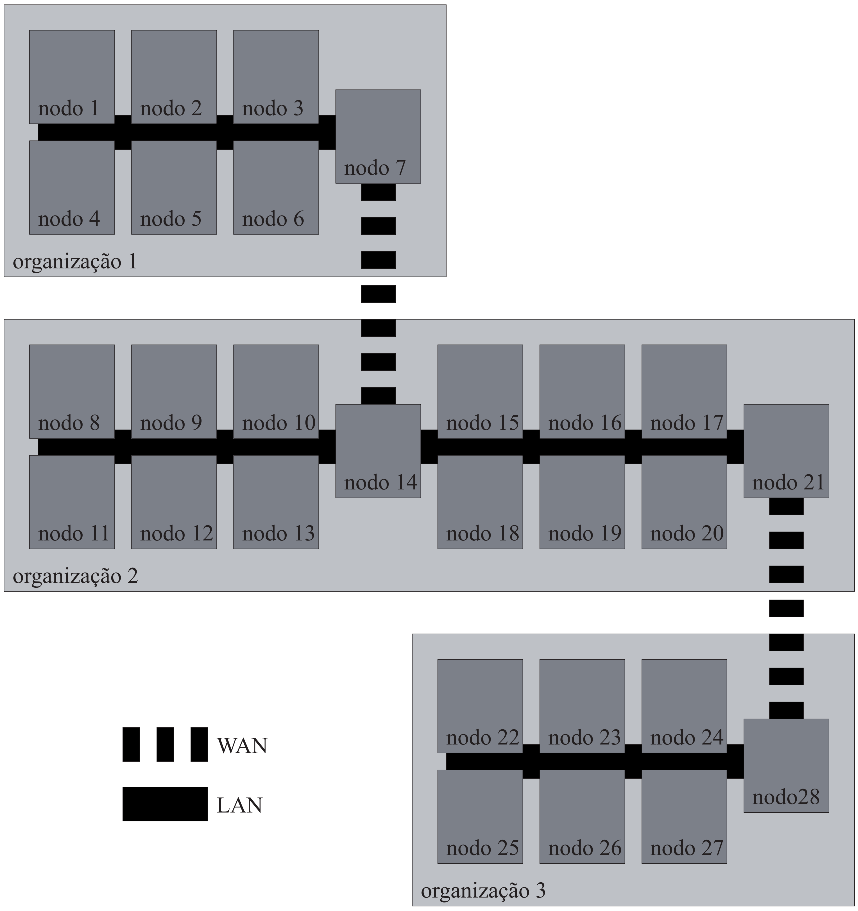


Figure 4.1: Exemplo de Sistema Distribuído Multi-Organizacional

- **Agentes de alocação**. São os agentes principais do sistema. São responsáveis pela gestão dos recursos da organização, pela atribuição desses recursos às tarefas, pela contratação automática de recursos de outras organizações, pela venda desses recursos e pela gestão das linhas de comunicação.

- **Agentes de controlo**. Responsáveis pelas operações de inicialização do sistema, pelo encaminhamento de mensagens entre agentes de alocação e pelo encaminhamento das tarefas geradas num nodo para os agentes de alocação respectivos. Existe um agente de controlo em cada nodo do sistema.

Um dos problemas principais que se colocou na concepção da arquitectura do sistema foi a definição da quantidade e organização dos agentes de alocação. Dependendo das características de cada organização, poderia ser preferivel adoptar uma arquitectura completamente distribuída, com um agente de alocação por cada nodo, uma arquitectura centralizada, com um agente a controlar todos os nodos, ou uma arquitectura intermédia. Outro problema importante consistia em decidir se a gestão de um nodo deveria ser atribuída apenas a um agente ou se deveria ser distribuída por todos os agentes. Para flexibilizar a arquitectura adoptou-se uma estruturação organizacional dinâmica, caracterizada pela existência de um número variável de agentes de alocação, sendo a gestão dos recursos distribuída pelos agentes também dinamicamente. Esta organização varia durante o processo de alocação de acordo com as características do ambiente e os objectivos definidos pela organização. A mudança de organização é regida por um protocolo especial apresentado na secção 4.6.

Em cada instante, uma organização é composta por N agentes de alocação, sendo cada agente caracterizado pelo seguinte tuplo:

$$Agente_i =< Localizacao_i, Nodos_i, Linhas_i, Organizacoes_i > \tag{4.1}$$

$Localizacao_i$ corresponde à identificação do nodo onde está presentemente localizado o agente. $Nodos_i$ e $Linhas_i$ correspondem a subconjuntos dos nodos e linhas da organização. $Organizacoes_i$ corresponde a um subconjunto das organizações do sistema.

Em cada instante também se devem verificar simultaneamente as seguintes condições:

$$\forall i \neq j, org(Agente_i) = org(Agente_j) Localizacao_i \neq Localizacao_j \tag{4.2}$$

$$\forall i \neq j, org(Agente_i) = org(Agente_j) Nodos_i \cap Nodos_j = \emptyset \tag{4.3}$$

$$\forall i \neq j, org(Agente_i) = org(Agente_j) Linhas_i \cap Linhas_j = \emptyset \tag{4.4}$$

$$\forall i \neq j, org(Agente_i) = org(Agente_j) Organizacoes_i \cap Organizacoes_j = \emptyset \tag{4.5}$$

O predicado $org(x)$ devolve o identificador da organização a que pertence x, que tanto pode ser um agente, um nodo ou uma linha.

A condição 4.2 representa a impossibilidade de dois agentes de alocação estarem localizados no mesmo nodo. Com efeito, dado que cada agente está preparado para funcionar qualquer que seja o tuplo que o caracteriza e dado que os parâmetros são sempre dijuntos, os dois agentes i e j podem sempre ser substituídos por um agente k com as seguintes características:

$$\begin{aligned} Agente_k &= < Localizacao_k, Nodos_i \cup Nodos_j, Linhas_i \cup Linhas_j, \\ &\quad Organizacoes_i \cup Organizacoes_j > \end{aligned}$$

$$Localizacao_k = Localizacao_i = Localizacao_j$$

A existência de um só agente aumenta a rapidez do processo de alocação, pois para se obter um plano óptimo de utilização dos recursos geridos pelos dois agentes era necessário a existência de comunicação entre ambos, através do protocolo próprio, enquanto que um só agente consegue o mesmo plano sem necessidade de comunicar. O atraso pelas comunicações não se deve à latência no meio de comunicação, dado que estão no mesmo nodo, mas ao assincronismo entre os *schedulers* dos dois agentes: um agente pode pretender um resultado que o outro agente só vai obter passado algum tempo. Este processo de redução do paralelismo interno de um nodo, através da junção dos processos equivalentes que nele executam, é uma técnica à muito utilizada na área de programação paralela.

As condições 4.3 e 4.4 atribuem a gestão de cada recurso de uma organização a apenas um nodo. Segundo [Shaerf et al., 1995] deveria existir uma separação clara entre agentes de alocação e recursos. Essa prespectiva dificilmente resultaria neste sistema pois é necessário, em determinados contextos, evitar conflitos no acesso aos recursos. Sendo um sistema multi-organizacional, onde a utilização de recursos de outras organizações implica a existência de um contracto que explicita em que quantidades esses recursos serão utilizados, estando o não cumprimento do mesmo sujeito a contrapartidas monetárias, é necessário existir um controlo perfeito da ocupação dos recursos para evitar que uma utilização exagerada leve ao não cumprimento de algum contrato. Para se conseguir este controlo, sem que haja apenas um nodo a controlar um recurso, seria necessário em cada contrato estimar em que medida todos os nodos pretendiam ocupar os recursos em causa, o que levaria a constantes difusões de informação. Existindo apenas um gestor para cada recurso, todo o processo se desenrola muito mais rapidamente pois a informação necessária está localmente acessivel. Por outro lado, se não existisse apenas um agente por recurso, também haveria problemas na identificação de qual o agente que deveria escalonar cada nova tarefa.

Dado que as linhas de comunicação são normalmente partilhadas por diversos nodos e devido à natureza dos acessos às mesmas, muito rápidos para se proceder a uma negociação prévia, as funções dos agentes que as gerem são ligeiramente diferentes das de gestão dos nodos. Cabe ao agente de alocação que controla uma linha definir a taxação a que será sujeita quando acedida por outras organizações. Esta taxação não é negociada a cada mensagem sendo definida periodicamente em função da procura externa dessas linhas e do uso que a organização detentora pretende fazer delas. Os únicos contratos feitos sobre as linhas são contratos a longo prazo (ver secção 4.4).

Finalmente, a condição 4.5 determina que o relacionamento com uma determinada organização seja feito apenas por um agente de alocação. Esta condição deriva um pouco da observação da realidade empresarial, onde o controlo do relacionamento com um determinado cliente está a cabo de um só gestor de conta. Esta política é necessária porque, na negociação de cada novo contrato, é desejável ter acesso a todo o historial do relacionamento com a organização em causa. Esta informação permite saber a "qualidade" dessa organização como cliente / fornecedor e ajuda a determinar a melhor estratégia de negociação. Se qualquer agente podesse fazer contractos com qualquer organização duas situações poderiam ocorrer: ou passado não era levado em conta, correndo-se o risco de fazer sucessivos maus contratos; ou seria necessário proceder à difusão da informação relativa ao historial dessa organização (com os consequente atrasos). Não se pode esquecer que nestas situações os atrasos nas respostas a contratos podem significar perdas de oportunidades e consequentes perdas de dinheiro. Com esta condição também se evitam algumas situações de fraude evidente, como situações de especulação, em que uma organização pega numa tarefa que lhe tinha sido contratada por uma organização e volta a contratá-la a outro agente da mesma organização (que antes talvez não a pudesse executar) ganhando algum lucro fácil sem nada fazer.

Esta condição não implica que uma tarefa originária num determinado agente não possa ser atribuída a uma organização controlada por outro agente. Quando um agente fica com uma tarefa através de um contrato pode sempre contratá-la de novo, por exemplo para outra organização, desde que se cumpram as condições do contracto original. Outro aspecto importante é a forma como uma organização contacta outra organização para proceder a uma negociação. Dado o facto da estrutura organizacional ser dinâmica torna-se bastante complicado uma organização contactar directamente o agente que presentemente se relaciona com ela. A solução para este problema passa pelos agentes de controlo, pois estes possuem uma localização fixa. Ao estilo dos protocolos de correio electrónico, optou-se por atribuír uma porta (endereço de escuta) específica em cada nodo para efeitos de contratação automática. Esta porta deveria estar definida numa versão real do protocolo e ser idêntica para todos os nodos envolvidos no processo de contratação. Quando uma organização deseja fazer um contracto com outra envia a mensagem em causa (ver secção 4.4), endereçada à porta referida, para qualquer dos nodos da outra organização. O agente de controlo presente nesse nodo, que monitoriza essa porta, capta a mensagem e, como tem conhecimento da estruturação organizacional actual, redirecciona a mensagem para o agente que trata dos contratos com essa organização. Esta política, para além de resolver os acessos remotos, apresenta outra vantagem: cada organização é livre de escolher qual o nodo de contacto, alargando-se assim o mercado de uma organização a organizações que, caso fossem obrigadas a contactar um nodo particular, não o fizessem devido aos custos de acesso.

Para finalizar a apresentação da arquitectura do sistema falta apenas referir o conceito de nodo virtual. Este conceito deriva da necessidade de uma organização fazer alocações rápidas de tarefas para outra organização. Efectivamente, pode dar-se o caso de uma organização estar permanentemente com sobrecarga de tarefas enquanto que outra deseja vender parte da capacidade dos seus recursos pois não os consegue utilizar completamente. No entanto, nem todas as tarefas podem ser sujeitas ao processo normal de contratação entre organizações, que por ser demorado pode não compensar se, por exemplo, a duração da tarefa fôr muito pequena. Para estas situações, seria vantajoso para as duas organizações existir a possibilidade de efectuar um contracto a longo prazo que permitisse a posterior transferência de tarefas sem negociação. O processo de contratação escolhido para estas situações consiste em fazer um contracto orientado aos recursos (ver secção 4.4) do qual resulta a possibilidade de uma organização utilizar uma certa capacidade dos recursos de um nodo de outra organização. Estes contractos, ao contrário dos orientados à tarefa, são válidos até denúncia por uma das partes e taxados por tempo de utilização. A melhor forma de gerir esses recursos contratados, consiste em criar um nodo virtual no nodo remoto real cuja capacidade é igual à capacidade dos recursos contratados. Estes nodos, são geridos como os restantes nodos da organização, ou seja, são-lhes instalados agentes de controlo e são geridos por um determinado agente de alocação. O acesso aos nodos virtuais pode ser também facilitado através da realização de um contrato a longo prazo sobre as linhas que ligam a organização a esses nodos. Esta estratégia tem várias vantagens, nomeadamente:

- **facilita a gestão dos contratos a longo prazo**, pois o acesso aos recursos contratados é feito de forma transparente por todos os agentes de alocação da organização; e
- **permite adoptar políticas de expansão de mercado**, pois ao se expandir fisicamente a organização, novas organizações poderão passar a interessar-se por negociar com ela devido aos menores custos de acesso.

Com este conceito, existe a possibilidade de a porta de contacto de um nodo ter de ser partilhada por dois agentes de controlo de diferentes organizações. Essa situação é resolvida com um pequeno *router* intermédio que encaminha a mensagem para o agente em causa. O desvio das mensagens por parte da organização detentora do nodo real é uma possibilidade, mas que no entanto não lhe traz grandes vantagens pois:

- não pode consultar as mensagens porque não possui a chave para as desencriptar (ver secção 4.7);
- se a organização que criou o nodo virtual não receber as mensagens, não verá vantagens por ter criado esse nodo, acabando eventualmente por rescindir o contrato (situação que também prejudicaria a organização detentora).

Uma das características mais evidentes desta arquitectura é a sua flexibilidade de adaptação às características de cada organização e às características dinâmicas do ambiente. Esta flexibilidade pode ser comprovada através dos exemplos apresentados nas figuras 4.2 e 4.3, onde são apresentados duas estratégias de gestão dos recursos internos da organização 1 da figura 4.1, nomeadamente, uma estratégia completamente distribuída e outra centralizada.

$$agente_1 =< nodo_1, \{nodo_1\}, \{\}, \{\} >$$

$$agente_2 =< nodo_2, \{nodo_2\}, \{\}, \{\} >$$
$$agente_3 =< nodo_3, \{nodo_3\}, \{\}, \{\} >$$
$$agente_4 =< nodo_4, \{nodo_4\}, \{\}, \{\} >$$
$$agente_5 =< nodo_5, \{nodo_5\}, \{linha_1\}, \{\} >$$
$$agente_6 =< nodo_6, \{nodo_6\}, \{\}, \{organizacao_1\} >$$
$$agente_7 =< nodo_7, \{nodo_7\}, \{\}, \{organizacao_2\} >$$

Figure 4.2: Estruturação Organizacional Completamente Distribuída

$$agente_1 =< nodo_7, \{nodo_1, nodo_2, nodo_3, nodo_4, nodo_5, nodo_6, nodo_7\},$$
$$\{linha_1\}, \{organizacao_1, organizacao_2\} >$$

Figure 4.3: Estruturação Organizacional Centralizada

4.4 Protocolo de Cooperação Entre Organizações

A arquitectura apresentada na secção 4.3 implica que os agentes de alocação assumam, simultaneamente, dois comportamentos distintos: benevolente, quando se relacionam com agentes da mesma organização, e egoísta, baseando-se nos interesses particulares da sua organização, quando se relacionam com agentes de outras organizações. Consequentemente, os protocolos de cooperação são ligeiramente diferentes, sendo apresentados em secções separadas. Nesta secção é apresentado o protocolo de cooperação entre agentes de diferentes organizações.

4.4.1 Limitações do *Contract Net Protocol*

Como já foi referido na secção 4.2, a maior parte dos protocolos para partilha de tarefas assemelham-se a um processo normal de contratação. Um dos primeiros protocolos deste género foi o *Contract Net protocol* (CNP) [Smith, 1980], no qual se basearam praticamente todos os protocolos seguintes. O CNP original apresenta algumas limitações quando aplicado a um processo de contratação automática num sistema multi-organizacional, pois não considera explicitamente aspectos tão importantes como a racionalidade limitada dos agentes e o facto de estes agirem de acordo com os seus interesses particulares [Sandholm and Lesser, 1995a].

Uma das extensões mais importantes ao CNP original é o protocolo apresentado em [Sandholm and Lesser, 1995a], onde, como forma de contornar os aspectos referidos, se procederam às seguintes alterações:

- **Inclusão dos aspectos monetários**. Como os agentes agem motivados pelos seus interesses apenas executam tarefas de outras organizações mediante pagamento monetário. Assim, o protocolo foi extendido com a inclusão de campos para explicitar o pagamento requirido / oferecido pela execução de uma tarefa e com mensagens para efectuar pagamentos.

- **Contratação simultânea de várias tarefas e apresentação de várias alternativas**. Pode haver vantagem em contratar simultaneamente várias tarefas devido à possibilidade de existirem economias de escala pela sua execução em conjunto. A proposta de alternativas é desejável porque muitas vezes os resultados pretendidos por um agente podem ser obtidos através da execução de tarefas diferentes. Na alocação de recursos esta situação também se verifica: por exemplo, para um agente que pretenda ver uma peça executada, é indiferente se ela é executada através de uma máquina CNC ou manualmente desde que se cumpram os prazos e padrões de qualidade pretendidos.

- **Possibilidade de fazer contrapropostas**. Como os agentes possuem uma racionalidade limitada, as propostas feitas num dado instante podem não corresponder à proposta ideal pois o agente está sujeito a pressões temporais, e como não pode calcular a melhor proposta num tempo infinitesimal (racionalidade perfeita) tem de apresentar uma proposta apenas satisfatória. Com a possibilidade de existirem contrapropostas, o agente tem uma oportunidade para rever a proposta inicial, podendo mesmo aproveitar o tempo desde a última proposta para continuar os seus cálculos e melhorar a proposta inicial. Esta estratégia também permite reagir a alterações no meio-ambiente, uma das causas da racionalidade limitada dos agentes.

- **Utilização de compromissos variáveis**. Num processo de negociação o compromisso significa que um agente fica obrigado a cumprir um contrato enquanto espera que o outro agente aceite ou rejeite a sua proposta [Sandholm and Lesser, 1995b]. Se a outra parte aceitar, ambos ficam obrigados a cumprir o contracto. Se rejeitar, o primeiro agente fica livre do seu compromisso. Na maior parte dos protocolos o comprisso dá-se num ponto fixo. Por exemplo no CNP original, quando um agente faz uma oferta fica automaticamente comprometido a cumprir o contracto se o agente que fez o anúncio a aceitar. Numa situação de racionalidade limitada esta situação não é desejável pois um agente, mais uma vez devido a pressões temporais, pode fazer uma oferta que se venha a revelar contra os seus interesses. O conceito de compromisso variável consiste na possibilidade de um compromisso poder ser quebrado á custa do pagamento de uma multa de descompromisso. Esta multa é acordada por ambas as partes, podendo ser variável como forma de variar o grau de compromisso - uma multa mais pesada garante que os agentes não quebram os compromissos enquanto que uma multa leve não garante a conclusão da tarefa.

- **Possibilidade de os pagamentos e multas de descompromisso serem funções do tempo, ou do acontecimento de certos eventos de negociação ou do meio-ambiente**. Estes acontecimentos devem ser observáveis por ambas as partes envolvidas no negócio. Esta possibilidade permite, por exemplo, que o pagamento por uma tarefa decresça com a passagem do tempo como forma de incentivar o envio rápido de respostas. Também permite a não existência de um prazo fixo para entrega de ofertas, dando mais liberdade ao processo de decisão dos agentes.

Um dos aspectos mais inovadores desta extensão ao CNP original é o conceito de compromisso variável. Este conceito apresenta diversas vantagens, nomeadamente [Sandholm and Lesser, 1995a]:

- através da especificação de baixas multas de situações de quebra de compromisso, percorrer um maior número de soluções para o problema de alocação, pois existem mais agentes a fazer propostas devido ao pouco risco do negócio;
- uma maior flexibilidade para processo de decisão dos agentes, porque a determinação da viabilidade de um contracto pode continuar após o envio de uma proposta devido à possibilidade de, mais tarde, quebrar o compromisso;
- variar o grau do risco envolvido no negócio através da variação da multa envolvida;
- efectuar contractos condicionais através da especificação de multas de quebra de compromisso dependentes de eventos futuros.

Exte protocolo, que os autores apresentam como englobando quase todas as extensões feitas ao CNP original, tem o problema de não permitir a negociação de contratos a longo pazo, sendo esta a única extensão relevante que lhe foi feita. Estes contractos, quando negociados por recursos, tal como foram apresentados na secção 4.3, têm três vantagens:

- permitem o aproveitamento de situações de contínuo acordo entre duas organizações quanto à transferência de tarefas entre elas, sendo a gestão destas transferências (sem negociação prévia) bastante simplificada devido ao conceito de nodo virtual (apenas possivel se estes contractos forem orientados aos recursos);
- permitem fazer contratos a longo prazo sobre a utilização de linhas de comunicação; e
- permitem que organizações com recursos disponíveis anunciem facilmente a sua disponibilidade a outras organizações.

A última vantagem apresentada já existia, quer no CNP original quer na extensão proposta em [Sandholm and Lesser, 1995a], embora com uma viabilidade duvidosa no contexto presente. No primeiro caso temos, mais uma vez, uma inadequação a sistemas multi-organizacionais, com a impossibilidade de apresentar o custo a que pretende disponibilizar os recursos. No segundo, como as propostas são orientadas às tarefas, um agente que pretenda disponibilizar os seus recursos tem duas hipóteses: ou sabe que tipo de tarefas os outros agentes pretendem executar num dado instante, ou apresenta uma lista com todas as tarefas que consegue executar. Nenhuma destas hipóteses é muito viável. Se o agente apresentar os recursos que tem disponiveis, os restantes agentes podem aproveitar para fazer um contrato a longo prazo, ou, se pretenderem apenas negociar uma tarefa, podem verificar facilmente se os recursos disponibilizados são suficientes para a executar e fazer uma contraproposta (já não orientada aos recursos, mas orientada para essa tarefa).

4.4.2 Estrutura das Mensagens

A sintaxe das mensagens existentes neste protocolo é apresentada na figura 4.4 utilizando notação BNF [Gries, 1981, páginas 304–309]. Os símbolos delimitados por "/ /" correspondem a campos de informação que devem ser preenchidos usando uma linguagem comum a todos os nodos [Smith, 1980]. Esta linguagem deve ser suficientemente expressiva por forma a flexibilizar o processo de negociação. Na figura 4.5 é apresentado o diagrama de transição de estados referente a uma negociação.

Proposta

Esta mensagem, para além das propostas, também é utilizada para fazer contrapropostas, para aceitar propostas e para terminar negociações. Quem envia a mensagem deve especificar se está no negócio como contrator, a querer comprar recursos, ou como contratado, a querer vender recursos. Quando se inicia uma negociação esta mensagem é utlizada para especificar quais as alternativas de contratação com que um agente se deseja compremeter. Cada alternativa é composta por um conjunto de items que tanto podem ser tarefas individuais, conjuntos de recursos ou linhas de comunicações. No caso das linhas, a especificação corresponde à identificação da linha que pretende utilizar. No caso dos recursos, a especificação é uma lista de recursos e respectivas capacidades pretendidas ou oferecidas. Estes recursos devem ser todos do mesmo nodo, não podendo o agente contratado alocar partes desses recursos a nodos diferentes. Como os items são atómicos e independentes entre si, o agente contratado pode alocá-los a nodos diferentes, ou voltar a contratar sub-conjuntos desses items para outros agentes.

Para cada item são apresentados o pagamento prometido caso seja aceite a proposta, a multa por descompromisso que pagará caso rejeite uma oferta válida e a multa de descompromisso que a outra parte deve pagar caso renuncie a um contrato previamente aceite. Estes campos podem ser funções do tempo ou de eventos de negociação e do meio-ambiente observáveis por

```
<mensagem> ::= <proposta> | <renuncia> | <conta> | <pagamento>

<proposta> ::= PROPOSTA <tipo> <id_negociacao> <id_mensagem> <resposta_a>
                        <origem> <destino> <termina> <alternativas>
<tipo> ::= CONTRATOR | CONTRATADO
<id_negociacao> ::= NEGOCIACAO /identificador_necociacao/
<id_mensagem> ::= MENSAGEM /identificador_mensagem/
<resposta_a> ::= RESPOSTA_A /identificador_mensagem/
<origem> ::= ORIGEM /identificador_agente/
<destino> ::= DESTINO /identificador_agente/
<termina> ::= TERMINA <flag>
<flag> ::= SIM | NAO
<alternativas> ::= <alternativa> {<alternativa>}
<alternativa> ::= ALTERNATIVA /identificador_alternativa/ <item> {<item>}
<item> ::= <tarefa> | <recursos> | <linha>
<tarefa> ::= TAREFA /especificacao_tarefa/
                    /pagamento_prometido/
                    /multa_descompromisso_prometida/
                    /multa_descompromisso_requerida/
<recursos> ::= RECURSOS /especificacao_recursos/
                        /pagamento_prometido_por_unidade_tempo/
                        /multa_descompromisso_prometida/
                        /multa_descompromisso_requerida/
<linha> ::= LINHA /especificacao_linha/
                  /pagamento_prometido_por_unidade_tempo/
                  /multa_descompromisso_prometida/
                  /multa_descompromisso_requerida/

<renuncia> ::= RENUNCIA <id_mensagem> <origem> <destino> <mensagem_aceitacao>
                        <que_recursos> <fim_contrato>
<mensagem_aceitacao> ::= MENSAGEM /identificador_mensagem/
<que_recursos> ::= RECURSOS /especificacao_recursos/ |
                   LINHA /especificacao_linha/
<fim_contrato> ::= FIM /instante_fim_utilizacao/

<conta> ::= CONTA <id_mensagem> <origem> <destino> <mensagens_cobradas> <valor>
<mensagens_cobradas> ::= <msg> {<msg>}
<msg> ::= MENSAGEM /especificacao_mensagem/
<valor> ::= VALOR /quantidade_monetaria/

<pagamento> ::= PAGAMENTO <id_mensagem> <origem> <destino> <tipo_pagamento>
                          <referente_a> <valor>
<tipo_pagamento> ::= CONTRATO | DESCOMPROMISSO | CONTA
<referente_a> ::= REFERENTE_A /identificador_mensagem/
```

Figure 4.4: Sintaxe do Protocolo de Cooperação Entre Organizações

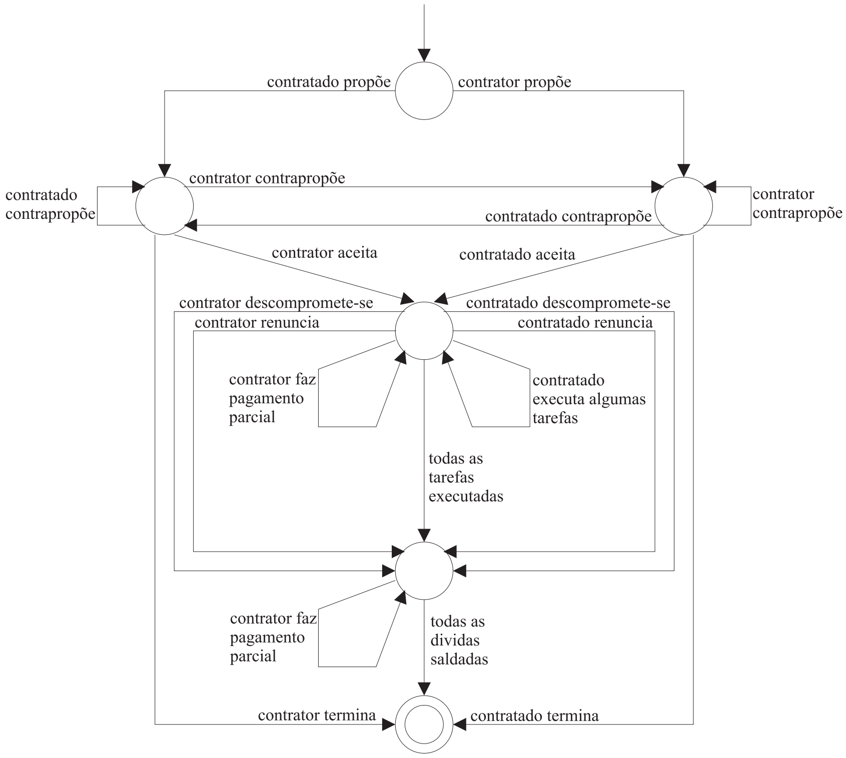


Figure 4.5: Diagrama de Transição de Estados de Uma Negociação

ambas as partes. Por exemplo, o pagamento pode ser função da duração de execução da tarefa - quanto mais rápida for executada mais o agente está disposto a pagar.

Um agente aceita automaticamente uma das alternativas propostas apresentando uma alternativa onde todos os items verificam as seguintes condições: os requisitos apresentados no campo de especificação são cumpridos, o pagamento requirido não excede o pagamento prometido, a multa de descompromisso requerida não excede a prometida e a prometida não é inferior á requirida. Quando se verificam estas condições, ou seja, quando uma das alternativas foi aceite pela outra parte, o agente contrator compromete-se automaticamente a pagar o pagamento especificado e só pode cancelar o contrato se pagar a multa de descompromisso respectiva.

Se alguma das condições referidas não se verificar a oferta assume o carácter de uma contraproposta, que a outra parte pode aceitar ou fazer outra contraproposta, podendo este processo seguir *ad infinitum*. Quando um agente não aceita nenhuma das alternativas, fazendo apenas contrapropostas, o outro agente fica descomprometido das suas propostas anteriores. O mesmo se passa quando uma das alternativas é aceite em relação às restantes alternativas. Neste caso o agente fica automaticamente descomprometido das restantes alternativas que propôs, não podendo a outra parte mais tarde exigir o pagamento de qualquer multa. Qualquer das partes pode, antes de alguma alternativa ter sido aceite, terminar o processo de negociação não havendo lugar a qualquer pagamento de multas de descompromisso.

No protocolo apresentado em [Sandholm and Lesser, 1995a] existe um prazo limite para a apresentação de ofertas, a partir do qual um agente fica descomprometido das alternativas apresentadas. Este prazo não é incluído pois pode sempre ser substituído por uma multa de descompromisso cujo valor é nulo a partir do prazo pretendido.

Renúncia

Esta mensagem, não proposta por [Sandholm and Lesser, 1995a], surge em consequência do aparecimento dos contratos a longo prazo. Como o fim destes não é especificado no contrato, é necessário existir uma mensagem para que uma das partes possa indicar que já não está interessada na continuação do contrato. Esta mensagem identifica os recursos em causa através da especificação da mensagem que causou a aceitação automática e da identificação do item correspondente aos recursos.

O último campo indica em que instante o agente vai deixar de utilizar os recursos, ou seja, quando se vai descomprometer do contrato. Este descomprometimento também está sujeito a multa. Normalmente, é conveniente para o agente contratado, que a outra parte avise com uma certa antecedência que vai rescindir o contrato, e, se esse aviso for feito com bastante antecedência, até pode não haver qualquer multa. Uma forma de se conseguir isto consiste em fazer com que a multa de descompromisso seja função da diferença entre o instante de chegada da mensagem de aviso e o instante de fim de contrato. Ambos os eventos são observáveis pelos dois agentes e portanto é uma função válida.

Conta

As linhas de comunicação podem ser utilizadas de duas formas: ou faz-se um contrato a longo prazo podendo-se depois enviar o número de mensagens que se quiser (desde que não se exceda a capacidade contratada), ou enviam-se as mensagens normalmente, sendo estas sujeitas à taxação em vigor fixada pelos agentes que gerem as linhas. Podemos assumir que, quando se envia uma mensagem, se faz automaticamente um contrato orientado à tarefa, sem negociação prévia, ficando ambas as partes comprometidas a cumpri-lo: uma a enviar a mensagem, e outra a pagar a taxação em vigor.

Periodicamente, os agentes que gerem linhas de comunicações enviam as contas relativas à utilização das suas linhas. Nesta mensagem aparecem descriminadas todas as mensagens que foram transmitidas nas suas linhas e o respectivo custo.

Pagamento

Existem três tipos de pagamentos neste protocolo: os referentes aos contratos acordados, as multas de descompromisso e as contas referentes à utilização de linhas de outras organizações. Qualquer dos pagamentos pode ser feito em prestações. No primeiro e segundo casos, o campo identificador de mensagem refere-se à mensagem que causou a aceitação automática do contrato. No terceiro refere-se à identificação da mensagem de apresentação de conta.

4.4.3 Redução do Congestionamento de Mensagens

Quando se define um protocolo de cooperação existe sempre o cuidado de se apresentar técnicas para a redução das mensagens em circulação. No caso do CNP original essa redução era conseguida à custa do *focused addressing* que consistia em especificar na mensagem de anúncio quais as condições que o agente de destino deveria verificar para enviar uma oferta. Neste caso, este mecanismo não é viável já que os agentes não agem de forma benévola. A decisão de enviar uma oferta é apenas baseada no seu interesse individual, não importando se essa mensagem vai provocar a saturação do outro agente [Sandholm and Lesser, 1995a].

Os factores que influenciam directamente o número de mensagens enviadas são os custos das comunicações, quanto mais mensagens se enviam mais se paga, e as multas de descompromisso, quanto mais baixas forem mais propostas se podem enviar. O primeiro ponto é relativamente fácil de controlar cabendo a cada organização controlar o acesso às suas linhas através de variações nos preços desses acessos. O segundo é ligeiramente mais complicado, pois quando uma gente faz uma proposta é livre de especificar qual o valor prometido para a multa de descompromisso. Nestes casos a única hipótese consiste em os agentes procederem a uma monitorização mútua, em que se controlam quais os agentes que enviam regularmente propostas com compromissos muito baixos, devendo esses agentes ser penalizados. Estas penalizações podem consistir em cortar relações com esse agente, que, desta forma, perde as vantagens que advêm da existência de um sistema multi-organizacional. Em qualquer protocolo de cooperação entre agentes egoístas, esta monitorização mútua e respectivas penalizações em futuros encontros é, normalmente, a única forma de garantir que as regras de bom comportamento sejam cumpridas por todos os agentes [Durfee, 1992].

4.5 Protocolo de Cooperação Dentro da Organização

O protocolo adoptado para a cooperação dentro da organização também é baseado no esquema de contratação do CNP. No entanto, como neste caso os agentes agem de forma benévola, já podem ser adoptadas técnicas existentes no CNP original como o *focused addressing*, atribuições directas de tarefas e pedidos de informação. Embora o protocolo que rege a mudança da estrutura organizacional também faça parte do protocolo de cooperação, é apresentado separadamente na secção 4.6 devido às grandes diferenças com o protocolo para alocação de recursos.

4.5.1 Funcionalidades Principais

Embora deixem de existir as preocupações derivadas de os agentes agirem de acordo com os seus interesses individuais, continua a ser necessário ter em consideração a racionalidade limitada dos mesmos. Embora este seja principalmente um problema a ser resolvido internamente por cada agente, é necessário que o protocolo seja suficientemente poderoso para lidar com essa limitação. As principais funcionalidades deste protocolo e diferenças em relação ao utilizado entre organizações (secção 4.4) são:

- **Não utilização de contrapartidas monetárias.** Como os agentes agem em função do bem colectivo não se justifica a existência de contrapartidas monetárias pela execução das tarefas ou pela quebra de compromissos. Transferências monetárias internas mantêm a utilidade da organização ao mesmo nível, pois esta só varia quando há transferências de ou para outras organizações. Se existissem, apenas poderiam levar à diminuição dessa utilidade devido à ocupação desnecessária de recursos: linhas de comunicação e capacidade de processamento para tratar as mensagens.

- **Utilização de compromisso fixo.** A utilização do compromisso variável na cooperação entre organizações visava, por um lado garantir que os agentes não eram sobrecarregados com mensagens inúteis, e por outro permitir a renúncia de contratos previamente acordados cuja manutenção deixou de ter interesse dadas as mudanças no contexto. Ambas as vantagens podem ser conseguidas entre agentes benevolentes sem se recorrer ao compromisso variável, com a consequente diminuição de complexidade. Todas as acções de um agente tem em vista o bem colectivo. Assim, apenas são enviadas as mensagens que se prevê originarem uma contribuição positiva para a utilidade geral. Quanto ao segundo aspecto, se um agente verifica que a execução de uma tarefa não se está a revelar positiva, simplesmente termina essa execução e anuncia um novo contracto para essa tarefa. Neste contexto é indiferente qual o agente a contratar uma tarefa e o cancelar do contrato apenas devolveria essa responsabilidade ao agente original.

- **Utilização de um protocolo de negociação com apenas três fases: anúncio, oferta e atribuição.** A possibilidade de fazer contrapropostas também não se justifica. A decisão de esquecer um anúncio feito anteriormente é pacífica, bastando ao agente enviar uma mensagem de terminação para todos os agentes envolvidos e voltar a fazer novo contrato. A contraproposta por parte do agente que faz a oferta também não faz sentido pois o anúncio já foi feito tendo em conta o bem colectivo.

- **Utilização de *focused addressing* para restringir o número de mensagens.** Entre agentes benevolentes já se justifica utilizar esta técnica para garantir que apenas serão enviadas ofertas com um mínimo de qualidade. Também representa um meio para os agentes lidarem com a sua racionalidade limitada: se quiserem reduzir o número de mensagens a processar fazem um *focused addressing* mais restrito.

- **Aparecimento de atribuições directas de tarefas.** Quando um agente prevê, ou sabe, que um agente pode executar uma tarefa mais satisfatoriamente que todos os outros, pode optar por lhe atribuir imediatamente essa tarefa sem negociação prévia. Esta facilidade, em conjunto com a possibilidade de os agentes anunciarem a sua disponibilidade, constitui o método interno priveligiado para se proceder à transferência rápida de tarefas. Os contratos a longo prazo orientados aos recursos continuam a existir por questões de compatibilização com o processo de contratação externo.

- **Existência de trocas de informação.** Dada a benevolência é agora possível aos agentes fazerem trocas de informação. Estas são vantajosas para: permitirem aos agentes saberem o estado dos outros agentes e assim criarem uma melhor imagem no nível de satisfação actual da organização; possibilitar a transferência entre agentes de recursos transferíveis, como por exemplo ficheiros num sistema distribuído ou materiais num sistema de produção, permitindo assim que os agentes alarguem o leque das tarefas que podem realizar. Esta última vantagem permite que agentes recém chegados ao sistema possam adquirir os recursos para se integrarem no processo de resolução, constituindo um forte contributo para a expansibilidade do sistema.

- **Utilização de respostas negativas.** Uma forma de diminuir a limitação da racionalidade dos agentes consiste em diminuir o número de situações que este tem de prever em cada nova acção. Assim, se um agente, em vez de esperar um determindado tempo até deixar de considerar uma situação, for imediatamente avisado de que ela não ocorrerá, sairá fortemente beneficiado no seu processo de decisão. Por exemplo, existem agentes que não podem responder a um anúncio devido ao *focused addressing*. O agente que faz o anúncio, por estar a contar com eles, pode atrasar a decisão quando, se esses agentes avisassem quanto à impossibilidade de resposta, poderia ser muito mais rápida aumentando a satisfação resultante do processo. Assim, foi criada uma mensagem especial, que permite a um agente enviar uma resposta negativa a uma mensagem enviada por outro agente.

As funcionalidades apresentadas, à excepção dos contratos múltiplos, das alternativas e dos contratos orientados aos recursos, estavam já previstas no CNP original.

4.5.2 Estrutura das Mensagens

A sintaxe das mensagens utilizadas é apresentada na figura 4.6. Mais uma vez, os símbolos delimitados por "/ /" correspondem a campos de informação que devem ser preenchidos usando uma linguagem comum a todos os nodos. Não é apresentado o diagrama de transição de estados, porque devido à não existência de contrapropostas e de descompromissos, o protocolo tornou-se bastante simples, sendo constituído maioritariamente por esquemas pergunta / resposta. Este protocolo é bastante mais claro que o anterior, e por isso apenas se procedem a pequenos esclarecimentos de alguns aspectos relevantes.

Anúncio

Esta é a mensagem que dá inicio a um processo normal de alocação. O campo elegibilidade é utilizado para se fazer o *focused addressing*. Neste campo são especificadas as condições que um agente deve verificar para fazer uma oferta sobre a alternativa em causa. A elegibilidade pode ser função do tempo ou de eventos observáveis por todas as partes envolvidas. A utilização mais frequente para este campo consiste na definição de um prazo limite para enviar ofertas. No CNP

```
<mensagem> ::= <anuncio> | <oferta> | <atribuicao> | <atribuicao_directa> |
               <termina> | <renuncia> | <resposta_negativa> | <informacao> | <conta>

<anuncio> ::= ANUNCIO <id_negociacao> <id_mensagem> <origem> <destino> <alternativas>
<id_negociacao> ::= NEGOCIACAO /identificador_necociacao/
<id_mensagem> ::= MENSAGEM /identificador_mensagem/
<origem> ::= ORIGEM /identificador_agente/
<destino> ::= DESTINO /identificador_agente/
<alternativas> ::= <alternativa> {<alternativa>}
<alternativa> ::= ALTERNATIVA /identificador_alternativa/ <item> {<item>}
                              <elegibilidade> <informacao_requerida>
<item> ::= <tarefa> | <recursos> | <linha>
<tarefa> ::= TAREFA /especificacao_tarefa/
<recursos> ::= RECURSOS /especificacao_recursos/
<linha> ::= LINHA /especificacao_linha/
<elegibilidade> ::= ELEGIBILIDADE /especificacao_elegibilidade/
<informacao_requerida> ::= REQUER /especificacao_informacao_requerida/

<oferta> ::= OFERTA <id_negociacao> <id_mensagem> <resposta_a> <origem>
                    <destino> <ofertas_alternativas>
<resposta_a> ::= RESPOSTA_A /identificador_mensagem/
<ofertas_alternativas> ::= <oferta_alternativa> {<oferta_alternativa>}
<oferta_alternativa> ::= ALTERNATIVA /identificador_alternativa/
                                     <informacao_requerida> <informacao_fornecida>
<informacao_fornecida> ::= FORNECE /informacao_fornecida/

<atribuicao> ::= ATRIBUICAO <id_negociacao> <id_mensagem> <resposta_a> <origem>
                            <destino> <alternativa_escolhida> <informacao_fornecida>
<alternativa_escolhida> ::= ALTERNATIVA /identificador_alternativa/

<atribuicao_directa> ::= ATRIBUICAO_DIRECTA <id_mensagem> <origem> <destino> <tarefa>

<termina> ::= TERMINA <id_negociacao> <id_mensagem> <origem> <destino>

<renuncia> ::= RENUNCIA <id_mensagem> <origem> <destino> <mensagem_aceitacao>
                        <que_recursos> <fim_contrato>
<mensagem_aceitacao> ::= MENSAGEM /identificador_mensagem/
<que_recursos> ::= RECURSOS /especificacao_recursos/ | LINHA /especificacao_linha/
<fim_contrato> ::= FIM /instante_fim_utilizacao/

<resposta_negativa> ::= RESPOSTA_NEGATIVA <id_mensagem> <resposta_a> <origem>
                                          <destino> <justificacao>
<justificacao> ::= JUSTIFICACAO /especificacao_justificacao/

<informacao> ::= INFORMACAO <id_mensagem> <origem> <destino>
                            <informacao_requerida> <informacao_fornecida>

<conta> ::= CONTA <id_mensagem> <origem> <destino> <organizacao> <mensagens_cobradas> <valor>
<organizacao> ::= ORGANIZACAO /identificador_organizacao/
<mensagens_cobradas> ::= <msg> {<msg>}
<msg> ::= MENSAGEM /especificacao_mensagem/
<valor> ::= VALOR /quantidade_monetaria/
```

Figure 4.6: Sintaxe do Protocolo de Cooperação Dentro da Organização

original existia, desnecessariamente, um campo específico para este fim. No campo informação requirida, é especificada a informação que deve ser enviada na oferta. Este campo deixa à liberdade dos agentes a decisão do tipo de oferta a enviar, não a limitando, por exemplo, a um simples valor numérico. Também serve, normalmente, para actualizar a informação relativa aos parâmetros que influenciam a medida de utilidade do sistema, para que o agente possa decidir a atribuição do contrato com uma visão mais correcta do estado da organização.

Oferta

Esta mensagem só é enviada em resposta a uma mensagem de anúncio. Um agente pode fazer ofertas sobre alternativas diferentes em mensagens e instantes diferentes, desde que cumpra a eligibilidade respectiva. No campo informação fornecida é feita a oferta na forma pedida. O campo informação requirida especifica a informação que o agente anunciante, caso aceite a oferta, deve enviar para que a tarefa possa ser executada. É normalmente utilizado para pedir recursos transferíveis, como por exemplo, no caso da distribuição de carga, o código fonte da tarefa a executar. Obviamente, na informação requirida não podem ser pedidos items especificados na condição de elegibilidade. O agente que faz uma oferta fica automaticamente comprometido a executar a tarefa caso o agente anunciante a aceite.

Atribuição

Esta mensagem é utilizada para avisar um agente de que a sua oferta foi escolhida. A partir deste instante, a responsabilidade da execução dessa tarefa fica a cabo do agente escolhido. Caso este não a consiga executar ou verifique que já não compensa ser ele a executá-la, deve iniciar um novo processo de contratação como anunciante.

Atribuição Directa

Esta mensagem é utilizada nas seguintes situações:

- quando um agente sabe qual o agente que consegue executar a tarefa com maior aumento de satisfação para a organização;
- quando um agente deseja transferir uma tarefa com a qual, por ser demasiado curta, não compense perder tempo de negociação; e

- pelos agentes de controlo, para transferir as tarefas criadas nos seus nodos para os agentes de alocação encarregados de gerir esses nodos.

Os agentes destinatários não podem recusar estas tarefas, podem sim, anunciar um contrato para as mesmas ou voltar a transferi-las usando a mesma mensagem.

Termina

Esta mensagem é utilizada pelo agente anunciante para cancelar um anúncio de contrato. Como já foi dito, é utilizada quando verifica que o contexto mudou de tal forma que o anúncio feito anteriormente já não faz sentido. Se, entretanto, já tiverem sido enviadas ofertas, estas serão ignoradas, ficando os agentes que as fizeram ao corrente da situação quando receberem a mensagem de terminação.

Renúncia

Tal como anteriormente esta mensagem serve para indicar que um contrato a longo prazo deixou de interessar. A semântica é idêntica à apresentada no protocolo de comunicação entre organizações.

Resposta Negativa

A resposta negativa é utilizada pelos agentes principalmente em quatro situações:

- quando a informação pedida não é possuida pelo agente;
- quando o agente não pode responder a um anúncio devido a não cumprir a elegibilidade especificada;
- quando a oferta feita por outro agente foi recusada; e
- quando não pode fazer uma oferta devido a estar demasiado ocupado.

A última situação, apresentada em [Smith, 1980], é de grande utilidade dado o contexto de racionalidade limitada dos agentes. Esta mensagem também pode ser utilizada noutras situações, desde que estejam previstas na linguagem comum aos nodos.

Informação

Esta mensagem é utilizada para requisitar e transferir informação e recursos transferíveis entre agentes de alocação e entre estes e os agentes de controlo. Como o dinheiro electrónico é também um recurso transferivel, pode ser movimentado dentro da organização usando esta mensagem. Outra utilização frequente para esta mensagem é a apresentação de disponibilidade por parte dos nodos, nomeadamente, especificando-se as capacidades de recursos disponíveis.

Conta

Esta mensagem surge em consequência da mensagem com o mesmo nome apresentada na secção 4.4. Como, devido à estruturação organizacional, as relações com uma determinada organização só podem ser feitas por um determinado agente de alocação, que não tem obrigatoriamente que controlar todas as linhas que essa organização pode utilizar, é necessário transferir-se as contas relativas à utilização das linhas para os agentes respectivos. Esta mensagem foi criada com essa finalidade e possui uma semântica semelhante à apresentada no protocolo de comunicação entre organizações.

4.6 Protocolo de Controlo da Estruturação Organizacional

O protocolo de controlo da estruturação organizacional é constituído apenas por três mensagens, cuja sintaxe é apresentada na figura 4.7. Ao contrário dos outros protocolos, onde descrevemos cada mensagem individualmente, optamos por apresentar os eventos principais de mudança de organização, indicando a forma como cada um deles está prevista no protocolo. Esta opção tem como objectivo provar a validade do protocolo. Relembrar que o conceito de nodo inclui os nodos reais e os virtuais.

Aparecimento de Nodo

Esta situação acontece nas seguintes situações: quando um novo nodo é acrescentado ao sistema; quando o sistema é iniciado; e quando um nodo recupera após uma falha. Quando aparece um novo nodo, a única acção que o gestor de sistema deve tomar caso deseje incluir esse nodo no sistema de alocação de recursos, consiste em pôr a funcionar um agente de controlo para esse nodo. A primeira tarefa deste agente consiste em perguntar a outro agente de controlo, ele sabe quais os nodos do sistema e que em cada nodo existe um agente, qual a estruturação organiza-

```
<mensagem> ::= <aparecimento> | <desaparecimento> | <conflito>

<aparecimento> ::= APARECIMENTO <id_mensagem> <id_agente> <localizacao>
                                {<nodo>} {<linha>} {<organizacao>}
<id_mensagem> ::= MENSAGEM /identificador_mensagem/
<id_agente> ::= AGENTE /identificador_agente/
<localizacao> ::= LOCALIZACAO /identificador_nodo/
<nodo> ::= NODO /identificador_nodo/
<linha> ::= LINHA /identificador_linha/
<organizacao> ::= ORGANIZACAO /identificador_organizacao/

<desaparecimento> ::= DESAPARECIMENTO <id_mensagem> <id_agente> <prazo>
<prazo> ::= PRAZO /especificacao_prazo/

<conflito> ::= CONFLITO <id_mensagem> <origem> <destino> <conflitos>
<origem> ::= ORIGEM /identificador_agente/
<destino> ::= DESTINO /identificador_agente/
<conflitos> ::= <localizacao_conflituosa> | <recursos_conflituosos>
<localizacao_conflituosa> ::= LOCALIZACAO /identificador_nodo/ /utilidade/
<recursos_conflituosos> ::= {<nodo_conflituoso>} {<linha_conflituosa>}
                           {<organizacao_conflituosa>}
<nodo_conflituoso> ::= NODO /identificador_nodo/ /utilidade/
<linha_conflituosa> ::= LINHA /identificador_linha/ /utilidade/
<organizacao_conflituosa> ::= ORGANIZACAO /identificador_organizacao/ /utilidade/
```

Figure 4.7: Sintaxe do Protocolo de Controlo da Estruturação Organizacional

cional actual. Para esta tarefa recorre-se à mensagem de troca de informação apresentada na secção 4.5.

O agente de controlo está encarregado de redireccionar as mensagens provenientes de organizações exteriores e de atribuir as tarefas geradas no seu nodo ao agente de alocação respectivo. Uma das regras protocolares do comportamento deste agentes consiste, sempre que este não conhece o destinatário de uma mensagem, em criar um novo agente de alocação no seu nodo que passe a ser esse destinatário. Assim, quando surge um novo nodo, se na estruturação actual esse nodo não fôr gerido por ninguém, quando surgir a primeira tarefa o agente de controlo respectivo criará um novo agente de alocação localizado no seu nodo e encarregado de gerir esse nodo. Situação semelhante acontece quando chega uma mensagem de uma organização que nenhum nodo gere. Neste caso é criado um agente encarregado apenas de gerir essa organização. Embora este procedimento possa criar momentaneamente uma organização pouco eficiente, este problema desaparecerá gradualmente devido à estruturação organizacional dinâmica.

Quando um novo agente de alocação é criado começa por questionar outro agente, semelhantemente ao agente de controlo, acerca da estruturação actual. A sua tarefa seguinte consiste em difundir uma mensagem de aparecimento para todos os agentes, de alocação e controlo, onde especifica qual a sua localização, os nodos, linhas e organizações que gere. Esta mensagem tem o objectivo de dar a conhecer aos outros agentes a existência de um novo agente de alocação.

Situações de Conflito

Qualquer situação de aparecimento de um novo agente é susceptivel de provocar um conflito devido à existência de agentes a gerirem os mesmos recursos, linhas ou organizações, ou localizados no mesmo nodo. Esta situação acontece principalmente devido à latência das comunicações - quando um agente aparece, a informação a esse respeito leva algum tempo a chegar a todos os agentes e, por isso, é muito provavel que, se outro agente aparcer entretanto, não obtenha ainda a informação actualizada, e tente ocupar parte da organização já ocupada pelo outro. A situação de conflito quanto à localização também pode acontecer porque um agente de controlo decide criar um novo agente no seu nodo quando já lá estava outro.

Em última instância, o conflito é detectado quando a mensagem de aparecimento é difundida e chega aos agentes afectados pelo aparecimento do novo agente (o conflito pode não se verificar apenas entre dois agentes). Quando o agente detecta que está em situação conflituosa difunde uma mensagem de desaparecimento temporário, para que os restantes agentes não comuniquem com ele durante a resolução do conflito. Esta mensagem de desaparecimento temporário consegue-se através do campo prazo da mensagem de desaparecimento. Este campo especifica o instante até ao qual os restantes agentes devem considerar o agente como estando vivo, mas inoperacional. Se até esse instante não surgir nenhuma mensagem de aparecimento do mesmo agente, então ele é considerado morto, podendo a parte da organização gerida por ele passar a ser gerida por outro agente.

Durante o estado de desaparecimento temporário, os agentes conflituosos trocam entre si mensagens do tipo conflito, onde, para cada parte da organização em conflito, cada agente especifica o ganho de utilidade previsto para a organização se ele ficar com a gestão dessa parte. O agente que apresentar maior ganho fica com agestão da parte em causa, devendo o outro agente passar-lhe toda a informação que possui acerca dessa parte (usando a mensagem para troca de informação). Quando os conflitos estiverem resolvidos os agentes difundem uma mensagem de aparecimento com as suas novas competências. Obviamente que estas mensagens podem originar novos conflitos, recomeçando-se o processo.

Os agentes exteriores ao conflito também o detectam quando recebem as mensagens de aparecimento respectivas. Enquanto um agente se aperceber que esse conflito existe não entra em contacto com os agentes envolvidos. Os agentes de controlo durante este período retêm as mensagens destinadas a esses agentes, e, caso um dos agentes em conflito seja o gestor do seu nodo, executam as tarefas originárias nesse nodo localmente. A situação de conflito desaparece com a chegada das mensagens de aparecimento ou com o fim do prazo de desaparecimento, voltando a situação à normalidade. Quando um agente desaparece, temporária ou definitivamente, os restantes agentes não detectam esse evento imediatamente, podendo, por isso, exitir mensagens em circulação endereçadas a agentes desaparecidos. Como o encaminhamento é feito pelos agentes de controlo, o primeiro que detectar que uma mensagem não tem destino envia uma resposta negativa ao agente que originou a mensagem, especificando se o agente desapareceu definitivamente ou temporariamente. Neste último caso, o agente que originou a mensagem pode reenviá-la quando receber a mensagem de aparecimento. O desaparecimento de um agente de alocação é detectado sempre, pelo menos pelo agente de controlo localizado no mesmo nodo onde se encontrava antes de desaparecer.

Desaparecimento

Um agente de alocação pode desaparecer definitivamente quando se apercebe que a sua existência não está a ser benéfica para a organização. Nestes casos difunde uma mensagem de desaparecimento com o prazo igual ao instante de tempo actual. Nos nodos que ficarem sem gestor, os agentes de controlo encarregar-se-ão automaticamente de criar novos agentes de alocação.

Junção de Agentes

A junção de dois agentes pode fazer-se de duas formas: os agentes podem acordar previamente, através de mensagens de troca de informação, que se devem juntar; ou um dos agentes muda-se para o nodo do outro, originando um conflito, cujo processo normal de resolução levará ao aparecimento de um só agente. No primeiro caso, os dois agentes desaparecem temporariamente, reaparecendo apenas um deles com as competências de ambos. No segundo caso temos um conflito ao nível da localização, cujas mensagens de resolução apenas apresentam um valor de utilidade. O agente que perder o conflito transfere todas as suas competências para o outro.

Fragmentação

A fragmentação de um agente implica o seu desaparecimento temporário seguido do aparecimento de um novo agente e do reaparecimento do antigo.

Desaparecimento de Nodo

Quando existe uma falha num nodo provocando o seu desaparecimento inesperadamente, as acções desencadeadas dependem do facto de ser nele se encontrar, ou não, uma agente de alocação. Se nele não se encontrar nenhum agente de alocação a resolução é simples: quando o agente que o geria detectar essa falha, retira-o da sua lista de competências. Quando o nodo regressar, o novo agente de controlo criará um novo agente de alocação. No segundo caso, quando qualquer outro agente detectar a seu desaparecimento difunde uma mensagem de desaparecimento definitivo em nome do agente desaparecido. Provavelmente, serão difundidas várias dessas mensagens, não constituíndo esta situação qualquer problema. Nos nodos que ficarem sem gestão serão automaticamente criados novos agentes de alocação.

Outras Mudanças

Qualquer mudança nas competências de um agente, como a localização ou as linhas, nodos e organizações geridas, faz-se difundindo uma mensagem de desaparecimento temporário seguida de uma mensagem de aparecimento quando o agente já se encontrar no novo estado.

4.7 Segurança nos Contratos e Transacções

Embora não se possa construir um sistema completamente seguro, é já possivel conseguir um mínimo de garantia nas transações efectuadas. A segurança neste sistema é composta por duas partes: a segurança das mensagens transferidas e a segurança quanto aos pagamentos e transação dos bens. A primeira parte diz respeito a todos os protocolos enquanto que a segunda apenas diz respeito ao protocolo de cooperação entre organizações.

A garantia dos protocolos internos à organização consegue-se utilizando, simultaneamente, criptografia de chaves secretas e funções de *hash* seguras [Simão, 1995]. Na criptografia, os dados transmitidos são transformados segundo um algoritmo de criptação, parametrizado com constante designada por chave de criptação. Na recepção, para recuperar os dados originais utiliza-se um algoritmo de decriptação parametrizado por uma constante designada chave de decriptação. Estes algoritmos tornam muito dificil, praticamente impossivel em tempo útil, a obtenção da informação transmitida sem que se conheça a chave de decriptação. Dentro da organização pode-se utilizar uma chave secreta, apenas conhecida pelos agentes da organização, para a transmição da informação. Por questões de simplificação pode ser utilizada a mesma chave para criptar e para decriptar, denominando-se, neste caso, o algoritmo de criptografia simétrica.

Esta técnica só por si não garante a integridade dos dados. Para isso, utilizam-se em conjunto funções de *hash* seguras. Uma função de *hash* segura é uma função de *hash* normal que tem a propriedade de não ser invertível, ou seja, dado o resultado não é possivel saber a entrada. Se esta função for aplicada à mensagem a ser transmitida obtem-se um valor que a caracteriza sem a revelar. Se esse valor for transmitido juntamente com a mensagem, ambos criptados, o receptor sabe se a mensagem foi ou não corrompida aplicando a mesma função de *hash* à mensagem e comparando o resultado obtido com o transmitido.

Embora esta estratégia de segurança não garanta a não repudiação por parte do receptor (ver secção 2.2.5), esta não é necessária pois os agentes agem benevenolamente entre si. A privacidade, autenticação e integridade são plenamente garantidas.

Entre organizações também se poderia utilizar esta técnica para garantir a segurança nas mensagens transmitidas. No entanto, seria necessário criar uma chave secreta para cada duas organizações que desejassem comunicar, chave esta que deveria ser transmitida antes de iniciar qualquer comunicação. Este último aspecto inviabiliza esta técnica pois a mensagem de transmição da chave não poderia ser encriptada, podendo a chave ficar acessivel a terceiros. Assim, entre organizações recorre-se a outra técnica de criptografia, que utiliza chaves públicas e privadas, e ao conceito de assinatura digital para transmitir a informação [Simão, 1995]. As funções de *hash* seguras continuam a ser utilizadas para garantir a integridade.

Nesta técnica, cada organização possui duas chaves: uma pública livremente distribuída pela rede e uma privada que apenas a organização conhece. Estas chaves têm a propriedade de serem inversas, ou seja, o que uma cripta a outra decripta e vice-versa. Os algorítmos que utilizam estas chaves denominam-se assimétricos porque utilizam chaves diferentes para criptar e decriptar. Quando uma organização deseja transmitir uma mensagem a outra, cripta-a com a chave pública da organização destino. Como apenas esta conhece a chave de decriptação é garantida a privacidade da transmissão. Para garantir a integridade volta-se a utilizar as funções de *hash* seguras. Embora a antenticidade do receptor não levante problemas, pois apenas este pode decriptar a mensagem, é necessário garantir a autenticidade de quem envia. Esta consegue-se com o conceito de assinatura digital, que não é mais do que uma marca que apenas a organização que assina poderia fazer. As assinaturas digitais implementam-se também à custa das chaves públicas e privadas. A organização que envia uma mensagem, antes de a criptar com a chave pública do receptor, acrescenta-lhe a sua identificação criptada com a sua chave privada. Esta informação é depois decriptada pelo receptor utilizando a chave pública da origem para verificar se a identificação é a esperada. Como uma chave pública apenas decripta informação criptada com a chave privada respectiva, está garantida a autenticidade da origem. A assinatura digital também garante a não repudiação por parte da origem, pois apenas essa organização poderia ter criptado uma informação decriptada com a sua chave pública.

Esta técnica apresenta como único defeito a não garantia de não repudiação por parte do receptor. Esta garantia é importantissima no protocolo apresentado, pois sem ela um agente pode esquivar-se do pagamento de multas de descompromisso alegando não ter recebido as mensagens.

O mecanismo de pagamento que melhor se aplica a este sistema é o dinheiro electrónico [Costa et al., 1995]. O dinheiro electrónico é uma sequência especial de *bits*, adquirida num banco virtual (usando cartões de crédito, cheques electrónicos ou dinheiro real), que uma vez emitido representa um valor que pode ser gasto no comércio ligado ao mesmo sistema. O protocolo de pagamento com dinheiro electrónico baseia-se na utilização mista de chaves públicas e secretas, e caracteriza-se pelas seguintes fases:

- o cliente envia ao fornecedor uma mensagem, encriptada com a chave pública do fornecedor, contendo a quantia de dinheiro electrónico em causa e uma chave secreta do cliente gerada para essa transação;
- o fornecedor envia uma mensagem ao banco virtual, encriptada com a chave pública do banco, contendo o dinheiro recebido e uma chave secreta do vendedor gerada para essa transação;
- o banco, após verificar a autenticidade do dinheiro, devolve novo dinheiro ao fornecedor, encriptando-o com a chave secreta que tinha recebido; e
- o fornecedor envia um recibo ao cliente, com a descrição do bem e valor pago, encriptando-

o com a chave secreta recebida do cliente.

O protocolo de pagamento apresentado tem o defeito de não garantir a transferência do bem propriamente dito. Para se conseguir essa garantia teria de se usar um protocolo muito mais complicado, em que o dinheiro adquirido no banco já vem identificado com o bem que com ele será comprado. No entanto, esse protocolo, devido á sua lentidão, seria de aplicação bastante limitada num sistema de alocação de recursos. Neste caso, essa garantia é ainda mais difícil pois os bens são tarefas a ser executadas, sendo, por exemplo, também muito difícil a uma organização provar que concluiu uma tarefa.

As técnicas apresentadas garantem apenas uma segurança parcial no sistema, pois não garantem a não repudiação por parte do receptor e a transferência dos bens comprados. Mais uma vez o bom funcionamento do sistema baseia-se na confiança entre as partes envolvidas, sendo a único mecanismo de garantir o bom comportamento a utilização de "vinganças particulares". Como já foi dito na secção 4.4.3 a única defesa de um agente que se sinta lesado é prejudicar o lesador nas futuras negociações, cortando relações com ele, reduzindo-lhe assim as oportunidades de negócio. Nenhum agente lesará constantemente todos os outros agentes, pois acabaria por ficar isolado, perdendo todas as vantagens inerentes à alocação de recursos num sistema multi-organizacional.

4.8 Agente de Alocação

A apresentação da arquitectura do sistema e dos protocolos de cooperação, ou seja, do mundo exterior aos agentes, não é suficiente para se compreender como é que os agentes cooperam. Para se estudar correctamente uma estratégia de cooperação é necessário examinar a estrutura interna dos agentes envolvidos [Jennings, 1996]. Assim, nesta secção vamos apresentar uma possivel estrutura interna de um agente de alocação, para que se possa entender melhor as motivações que levam este agente a cooperar. Apenas são apresentados alguns dos aspectos principais que influenciam o seu comportamento, estando a estrutura apresentada longe de uma implementação real.

4.8.1 Arquitectura do Agente

Como se pode ver na figura 4.8, um agente de alocação é constituído por três módulos:

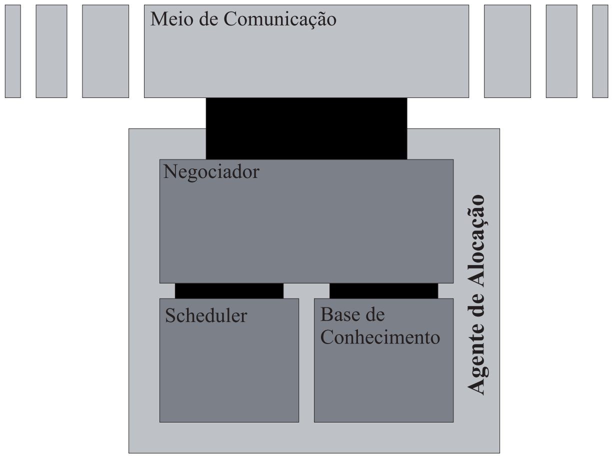


Figure 4.8: Arquitectura de um Agente de Alocação

- **Negociador.** Responsável por gerir o seu domínio, maximizando a satisfação global da organização. Trata de todo o processo negocial, nomeadamente, dos anúncios de contratos, da apresentação de ofertas relativas a contratos anunciados por outros agentes, e da decisão de quais o agentes que vão ser contratados para executar uma tarefa. Na negociação com outras organizações, também é responsável pela apresentação de contrapropostas e por efectuar os pagamentos. Também lhe compete decidir quanto à mudança da estruturação organizacional.

- ***Scheduler.*** Cálcula, heuristicamente, os planos de execução de tarefas pedidos pelo módulo de negociação.

- **Base de Conhecimento.** Armazena toda a informação necessária ao funcionamento do agente, nomeadamente, a estruturação organizacional actual, a descrição da ocupação actual dos nodos e linhas geridos pelo agente, as estatísticas de execução de tarefas e os resultados das negociações passadas.

Embora seja necessária uma interface com o gestor do sistema, para permitir a definição dos objectivos da organização e a parametrização do funcionamento do agente, essa interface não pode ser feita directamente com o agente de alocação devido ao carácter dinâmico da estruturação organizacional, que impede a fácil localização dos agentes. Esta interface é feita através dos agentes de controlo, com localização fixa e conhecida, que depois difundem os comandos do gestor para todos os agentes de alocação.

Como a estruturação organizacional é dinâmica não é possivel ao gestor do sistema configurar individualmente cada agente. Essa configuração individual levantaria problemas quando os agentes se juntassem ou fragmentassem. As configurações são difundidas para todo os agentes, adoptando estes, pelo menos momentaneamente, um comportamento idêntico no que se refere aos aspectos configurados.

4.8.2 Função de Utilidade

Como as actividades dos agentes de alocação tem por objectivo maximizar a satisfação da organização é necessário existir uma medida para essa satisfação. A microeconomia, ciência que estuda o comportamento e interação das diversas entidades existentes num mercado, utiliza o conceito de função de utilidade para esse efeito [Pindyck and Rubinfeld, 1992]. Uma função de utilidade estabelece a relação entre os parâmetros que influenciam a satisfação duma entidade e um valor numérico que permite medir e comparar essa satisfação. Este conceito constitui uma ferramenta valiosa na análise do comportamento de uma entidade comercial. Assim, para um determinado instante t, a utilidade $\mathcal{U}(t)$ da organização é definida por

$$\mathcal{U}(t) = \mathcal{F}(D(t), P_1(t), \ldots, P_{Np}(t)), \tag{4.6}$$

onde $P(t)$ representa o valor no instante t de um qualquer parâmetro que influencia o grau de satisfação da organização, Np representa o número desses parâmetros e $D(t)$ representa o dinheiro existente na organização no instante t, sendo este o único parâmetro que deve obrigatoriamente existir na função de utilidade. Esta obrigatoriedade deve-se ao facto de a organização estar envolvida num mercado, onde as transacções de recursos se fazem a troco de dinheiro, existindo a necessidade de os agentes saberem, em cada instante, quanto a dinheiro a organização está disposta a dar ou receber em troca de uma determinada quantidade de recursos, ou seja, qual a quantidade de dinheiro que deixará a organização pelo menos no mesmo nível de satisfação. Como o nível de satisfação é dado pela função de utilidade é necessário que a quantidade de dinheiro existente seja um parâmetro dessa função.

Definida a função de utilidade o objectivo dos agentes é maximizar colectivamente o valor dessa função de utilidade. No entanto, os agentes apenas possuem uma vista parcial do estado do sistema não podendo calcular o valor global da utilidade. Para que tal seja possivel, é necessário existir comunicação entre os agentes. Numa primeira abordagem, parecia ser suficiente a comunicação do valor da utilidade calculado localmente. No entanto, esta abordagem é bastante limitativa porque, sendo $\mathcal{U}^i(t)$ o valor da função de utilidade no agente i no instante t e N o número de agentes de alocação, seria necessário que a função de utilidade verificasse a propriedade

$$\mathcal{U}(t) = \sum_{i=1}^{N} \mathcal{U}^i(t),$$

o que raramente acontece.

A opção adoptada consiste em comunicar os parâmetros da função de utilidade. Embora menos limitativa continua a impôr algumas restrições, agora ao nível dos parâmetros. Sendo $P_j^i(t)$ o valor do parâmetro j no agente i no instante t e $P_j(t)$,o valor do mesmo parâmetro em toda a organização, deve-se verificar o seguinte:

$$\forall j \quad P_j(t) = \sum_{i=1}^{N} P_j^i(t). \tag{4.7}$$

O parâmetro $D(t)$ verifica sempre esta condição, dado que a quantidade de dinheiro existente na organização é igual à soma das quantidades existentes nos diversos agentes. Um exemplo de parâmetro que não pode ser utilizado é a média de um conjunto de valores. No entanto, neste caso, e em muitos outros, é possivel dar a volta ao problema utilizando como parâmetro não a média, mas sim a soma dos valores envolvidos e o número desses valores. Estes dois parâmetros já respeitam a condição 4.7.

Definido o que vai ser transmitido, falta definir quando se fará essa transmissão. É necessário definir uma política de transmissão que permita estabelecer uma contrapartida entre o número de mensagens em circulação e a actualidade dos parâmetros. Para tal definine-se o conceito de idade de um determinado dado. Sendo D um dado qualquer, t_{act} o instante de tempo actual e t_{obt} o instante em que D foi obtido, temos que $\mathcal{I}(D)$, a idade de D, define-se como:

$$\mathcal{I}(D) = t_{act} - t_{obt} \tag{4.8}$$

Recorrendo à equação 4.8 é agora possivel definir a idade do valor da função de utilidade num determindado agente i como:

$$\mathcal{I}(\mathcal{U}^i(t)) = \frac{\sum_{k=1}^{N}(\mathcal{I}(D^k(t)) + \sum_{j=1}^{Np} \mathcal{I}(P_j^k(t)))}{(Np+1) \times N} \tag{4.9}$$

Com a equação 4.9 é possivel parametrizar o comportamento dos agentes, definindo que:

$$\forall i \quad \mathcal{I}(\mathcal{U}^i(t)) \leq Idade \tag{4.10}$$

Variando-se o valor de $Idade$, varia-se o número de actualizações que os agentes deverão fazer para actualizar o valor da função de utilidade - quanto mais elevado for $Idade$ menos actualizações terão de ser feitas e vice-versa. Uma das técnicas que se podem usar para reduzir o tráfego na rede consiste em aproveitar o campo informação requirida das mensagens de anúncio e oferta (figura 4.6) para pedir os parâmetros. Outra possibilidade consiste em tornar o valor de $Idade$ adaptativo, podendo cada agente estabelecer qual o seu valor. Se se verificar que o valor dos parâmetros se mantêm estáveis podemos tornar o valor de $Idade$ mais elevado sem prejuízo na actualidade. Uma forma de medir a estabilidade dos parâmetros consiste em calcular o seu desvio padrão - quanto menor mais estáveis são os dados e mais alto pode ser o valor de $Idade$. Para ajustar este valor dinâmicamente apenas precisamos de definir $Idade$ como uma função do desvio padrão.

4.8.3 Anúncios de Contratos

Quando surge uma nova tarefa, quer criada localmente, quer através de uma atribuição de contrato, é necessário decidir se ela vai ser executada localmente, se vamos anunciar um contrato interno à organização, se vamos anunciar um contrato para outras organizações e se decidir-se que vai ser anunciado um contrato, é preciso decidir também para que agentes vai ser enviado o anúncio. Esta decisão depende principalmente dos seguintes factores: da influência que a execução da tarefa tem no grau satisfação da organização e do custo das comunicações necessárias para fazer o anúncio. Esta decisão é semelhante à existente na microeconomia para definir a extensão do mercado de um produto. A extensão deste mercado é tanto maior quanto maior for o ratio entre o custo do produto e seu custo de transporte [Pindyck and Rubinfeld, 1992]. Neste caso, vamos definir dinamicamente a extensão do mercado para cada nova tarefa. Se esse mercado for nulo então a tarefa é executada localmente.

O primeiro passo para determinar a extensão do mercado consiste em determinar a influência que a execução da tarefa tem no grau satisfação da organização. Como os anúncios podem incluir diversas tarefas vamos considerar que T representa o conjunto das tarefas que um agente pretende anunciar. A influência da execução de T vai ser medida como $\Delta_{adi}\mathcal{U}(T)$, que representa a utilidade marginal por se adicionar T ao conjunto de tarefas executadas localmente. Sendo T_{act} o conjunto das tarefas actualmente em execução das quais o agente não se pode descomprometer (tarefas internas da organização), T_{desc} o conjunto das tarefas em execução das quais o agente se pode descomprometer (mediante o pagamento da multa respectiva) e T_{ofer} o conjunto das tarefas que podem ser atribuidas ao agente em consequência das ofertas feitas por este (sem esquecer que uma oferta para outra organização, quando não aceite, implica o recebimento de uma multa de descompromisso), então $\Delta_{adi}\mathcal{U}(T)$ está aproximadamente limitado inferiormente por

$$\Delta^{-}_{adi}\mathcal{U}(T) = \min_{T'_{desc} \subseteq T_{desc}, T'_{ofer} \subseteq T_{ofer}} [\mathcal{U}(T_{act} \cup T'_{desc} \cup T'_{ofer} \cup T) - \mathcal{U}(T_{act} \cup T'_{desc} \cup T'_{ofer})],$$

e superiormente por

$$\Delta^{+}_{adi}\mathcal{U}(T) = \max_{T'_{desc} \subseteq T_{desc}, T'_{ofer} \subseteq T_{ofer}} [\mathcal{U}(T_{act} \cup T'_{desc} \cup T'_{ofer} \cup T) - \mathcal{U}(T_{act} \cup T'_{desc} \cup T'_{ofer})].$$

Estas expressões são baseadas em [Sandholm, 1993]. Nesse trabalho apenas se considera o custo de execução e, também não se considera a possibilidade de o agente se descomprometer da execução de uma tarefa. O parâmetro t foi excluído para melhorar a legibilidade e compreensão das expressões. Os valores obtidos são aproximações, porque não incluem as novas tarefas e anúncios que possam surgir. A análise dessas possibilidades é problemática porque para cada nova tarefa que possa surgir seria necessário estimar o seu mercado, a possibilidade de ser aceite, etc. Essa análise é impraticável, e a utlidade marginal que assim seria calculada provavelmente cairia dentro dos valores obtidos, pois estes já correspondem a situações limite bastante improváveis de acontecer.

O valor da expressão $\mathcal{U}(Tarefas)$ é calculado em duas fases. Primeiro, é definido pelo *scheduler* um plano para a execução de todas as tarefas contidas em $Tarefas$, que maximize o valor da função de utilidade definida na expressão 4.6. Depois, calcula-se o valor da utilidade incluindo os custos de descompromisso resultantes de tarefas deixadas de executar e de ofertas não consideradas.

Para que o *scheduler* possa fazer um plano de execução, é necessário haver uma estimativa dos recursos consumidos pelas diversas tarefas. Esta estimativa, se for uma tarefa essencial, deve ser fornecida em conjunto com o pedido de submissão. Quando não é uma tarefa essencial é necessário recorrer à memória do agente e, em função dos recursos consumidos no passado, calcular uma estimativa de consumo.

Devido a limitações de espaço e por razões de eficiência, não é possivel considerar todo o passado do agente, sendo necessário limitar a memória do mesmo. O tamanho da memória do agente pode ser definido como um intervalo de ΔT instantes de tempo, que corresponde à memorização dos eventos ocorridos entre $t_{act} - \Delta T$ e t_{act}. ΔT constitui um parâmetro muito importante do funcionamento do agente, na medida em define o seu comportamento ao nível da capacidade de reação. Quanto maior for ΔT maior a memória do agente, sendo considerados eventos mais antigos, que provavelmente estão desactualizados, sendo o agente incapaz de reagir a mudanças rápidas no ambiente. Quanto menor for ΔT, maior capacidade de reação. De notar que ter maior capacidade de reação não é necessariamente melhor, pois muitas das alterações do ambiente são passageiras e às quais não é boa política reagir. É necessário descobrir o ponto de equilibrio que garante o melhor funcionamento.

Para estimar a quantidade de um recurso necessária para executar uma tarefa, começa-se por calcular a média e o desvio padrão da quantidade desse recurso consumida no passado. Considerando que existem N amostras da quantidade consumida, $X_1, \ldots, X_N$, então temos que a média é calculada como

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N},$$

e o desvio padrão como [Fernandes, 1993]

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (X_i - \overline{X})^2}.$$

Se nenhuma execução constar na memória do agente, considera-se a média e desvio padrão do consumo desse recurso em todas as outras tarefas. Se apenas constar uma execução, a média corresponde ao consumo verificado nessa execução, sendo o desvio padrão o do consumo desse recurso em todas as outras tarefas. Em ambos os casos, considera-se $N = 2$ como forma de diminuir a confiança na estatistica a seguir obtida. Para que estas situações não sejam muito frequentes ΔT deve ser bastante superior ao intervalo médio entre surgimentos da mesma tarefa. Obtidos $\overline{X}$ e s, se $N < 30$ e as amostras seguem distribuições normais, podemos afirmar com uma confiança de $100 \times \alpha\%$, que μ, uma estimativa para a quantidade de recurso necessária, verifica [Fernandes, 1993]

$$\mu < \overline{X} + t_{1-\alpha,N-1} \times \frac{s}{\sqrt{N}},$$

sendo $t_{1-\alpha,N-1}$ o ponto crítico da distribuição t - Student, com $N-1$ graus de liberdade, à direita do qual se encontra uma probabilidade (área) de $1-\alpha$. Este valor é facilmente calculado pois a distribuição t - Student encontra-se tabelada. Dentro deste intervalo podemos apontar a pior estimativa μ, o valor que será considerado, como sendo

$$\mu = \overline{X} + t_{1-\alpha,N-1} \times \frac{s}{\sqrt{N}}. \tag{4.11}$$

Se $N \geq 30$, mesmo que as amostras não sigam uma distribuição normal, podemos afirmar com uma confiança de $100 \times \alpha\%$, que

$$\mu < \overline{X} + z_{1-\alpha} \times \frac{s}{\sqrt{N}},$$

sendo $z_{1-\alpha}$ o ponto crítico da distribuição normal, com parâmetros 0 e 1, à direita do qual se encontra uma probabilidade (área) de $1 - \alpha$. A pior estimativa para μ é

$$\mu = \overline{X} + z_{1-\alpha} \times \frac{s}{\sqrt{N}}. \tag{4.12}$$

Como se $N \geq 30$ a estimativa calculada em 4.12 é independente da distribuição que seguem as amostras, sendo por isso muito mais fiável, é conveniente que ΔT seja suficientemente grande para permitir, na maior parte dos casos, a existência desse número de amostras.

O valor de α $(0 < \alpha < 1)$ representa mais um parâmetro de grande importância no comportamento do agente, na medida em que representa a atitude deste perante o risco. Para valores de α próximos de 0 temos um agente com grande gosto pelo risco e para valores de α próximos de 1 temos um agente mais cuidadoso. Por exemplo na estimativa 4.12, um valor de $\alpha = 0.5$ implica um risco médio pois $\mu = \overline{X}$. Mais uma vez, é de referir que a atitude perante o risco não está directamente relacionada com a eficiência do agente.

Esclarecido o cálculo de $\mathcal{U}(Tarefas)$, voltemos ao cálculo de $\Delta_{adi}\mathcal{U}(T)$. Antes de apresentar a expressão para esta quantidade, façamos uma análise empírica da relação entre a atitude do agente perante o risco e o valor de $\Delta_{adi}\mathcal{U}(T)$. Como este valor representa a utilidade marginal por se executar mais um conjunto de tarefas localmente, é natural que seja um valor negativo. Quanto mais negativo for, maior a extensão do mercado em que interessa anunciar a tarefa. Um agente arriscador anunciará a tarefa a mais agentes na procura de obter uma melhor oferta, mas arriscando mais dinheiro devido aos custos do anúncio e multas de descompromisso. Pelo contrário, um agente mais cuidadoso fará o anúncio a menos agentes. Para se conseguir este comportamento, um agente que arrisca estimará o valor de $\Delta_{adi}\mathcal{U}(T)$ como sendo próximo de $\Delta^{-}_{adi}\mathcal{U}(T)$, e um agente cuidadoso como sendo próximo de $\Delta^{+}_{adi}\mathcal{U}(T)$. Após esta análise deduz-se a expressão para $\Delta_{adi}\mathcal{U}(T)$, como sendo:

$$\Delta_{adi}\mathcal{U}(T) = \Delta^{-}_{adi}\mathcal{U}(T) + \alpha \times (\Delta^{+}_{adi}\mathcal{U}(T) - \Delta^{-}_{adi}\mathcal{U}(T)) \tag{4.13}$$

Se definirmos $\mathcal{E}(\Delta^{A}\mathcal{U}(T))$ como a esperança de melhoria da utilidade por se anunciar o conjunto de tarefas T ao conjunto de agentes A, a função $\mathcal{F}(A)$ como

$$\mathcal{F}(A) = \mathcal{E}(\Delta^{A}\mathcal{U}(T)) - \Delta_{adi}\mathcal{U}(T),$$

$Agentes$ como o conjunto de todos os agentes de alocação (de todas as organizações) excluindo o agente que faz o anúncio, então o conjunto de agentes $\mathcal{M}(T)$ para os quais de deve anunciar T, ou seja o mercado para um anúncio, é definido por:

$$\mathcal{M}(T) = A \quad sse \quad A \subseteq Agentes \quad \wedge \quad \mathcal{F}(A) > 0 \quad \wedge$$

$$\forall B \subseteq Agentes, \mathcal{F}(B) > 0 \quad \mathcal{F}(A) \leq \mathcal{F}(B)$$

Se definirmos $\mathcal{E}(\Delta^{O}_{com\_oferta}\mathcal{U}(T))$ como a variação de utilidade esperada se o conjunto de agentes O fizer ofertas válidas, e $\Delta^{C}_{comunicacoes}\mathcal{U}(T)$ como a variação de utilidade devido aos custos de enviar o anúncio de T para todos os agentes de C, custos referentes à infraestrutura de comunicações, então $\mathcal{E}(\Delta^{A}\mathcal{U}(T))$ está limitado inferiormente por

$$\mathcal{E}^{-}(\Delta^{A}\mathcal{U}(T)) = \min_{O \subseteq A}(\mathcal{E}(\Delta^{O}_{com\_oferta}\mathcal{U}(T)) + \Delta^{A}_{comunicacoes}\mathcal{U}(T)),$$

e superiormente por

$$\mathcal{E}^{+}(\Delta^{A}\mathcal{U}(T)) = \max_{O \subseteq A}(\mathcal{E}(\Delta^{O}_{com\_oferta}\mathcal{U}(T)) + \Delta^{A}_{comunicacoes}\mathcal{U}(T)).$$

Como um agente que arrisca espera sempre um maior retorno do anúncio temos

$$\mathcal{E}(\Delta^{A}\mathcal{U}(T)) = \mathcal{E}^{+}(\Delta^{A}\mathcal{U}(T)) - \alpha \times (\mathcal{E}^{+}(\Delta^{A}\mathcal{U}(T)) - \mathcal{E}^{-}(\Delta^{A}\mathcal{U}(T))). \tag{4.14}$$

Se considerarmos que, dos agentes que respondem vamos escolher o que no passado nos ofereceu maiores variações de utilidade sendo este agente denomidado $Vencedor$, que $\Delta^{D}_{desc}\mathcal{U}(T)$ corresponde à variação de utilidade devido ao pagamento de multas de descompromisso aos agentes contidos em D, que $\Delta^{Vencedor}_{envia}\mathcal{U}(T)$ corresponde à variação de utilidade resultante dos custos de enviar T para o agente $Vencedor$ e de efectuar o respectivo pagamento, e que $\mathcal{E}(\Delta^{Vencedor}_{executa}\mathcal{U}(T))$ é a esperança de variação de utilidade se o agente $Vencedor$ executar T, então

$$\mathcal{E}(\Delta^{O}_{com\_oferta}\mathcal{U}(T)) = \max_{Vencedor \in O}(\mathcal{E}(\Delta^{Vencedor}_{executa}\mathcal{U}(T)) + \Delta^{Vencedor}_{envia}\mathcal{U}(T) + \Delta^{O-\{Vencedor\}}_{desc}\mathcal{U}(T)). \tag{4.15}$$

Considerando que na memória do agente existem N amostras, $X_1, \ldots, X_N$, correspondentes a antigos $\Delta^{Vencedor}_{executa}\mathcal{U}(T)$, então podemos calcular μ, uma estimativa para $\Delta^{Vencedor}_{executa}\mathcal{U}(T)$, através da equações 4.11 ou 4.12, se $N < 30$ ou $N \geq 30$, respectivamente. No entanto, esta estimativa deve ser alterada para permitir, por um lado, dar maior credibilidade às amostras mais recentes, e por outro, que o agente faça contratos com agentes com quem nunca se relacionou e que esqueça antigos maus contratos com alguns agentes, dando-lhes nova chance de serem contratados. Assim, considerando

$$I_{amostra} = \min_{1 \leq i \leq N} (\mathcal{I}(X_i)),$$

temos que

$$\Delta_{executa}^{Vencedor} \mathcal{U}(T) = \frac{\Delta T - I_{amostra}}{\Delta T} \times \mu.$$

Com esta equação ficou bem realçada a importância que tem o parâmetro ΔT. Se for muito grande, o agente demora mais tempo a esquecer um mau contrato, que pode ter sido uma excepção, mas, por outro lado, envia os anúncios para os agentes que no passado deram sistematicamente melhores resultados. Se for muito pequeno, o agente terá tendência a esquecer rapidamente os contratos feitos, adoptando o agente uma atitude mais exploradora (experimenta mais novos agentes), dado que $\Delta_{executa}^{Vencedor} \mathcal{U}(T)$ se aproxima rapidamente de 0 para todos os agentes tornando-os indiferentes aos olhos do agente que anuncia.

Até agora, não se especificou como devem ser calculados o pagamento prometido e as multas de descompromisso. Efectivamente, esses valores influenciam o número de agentes a responderem a um anúncio, mas a variação de utilidade final é, pelo menos teoricamente, independente desses valores, dado que são grandezas que tendem a anular-se mutuamente. Se considerarmos os casos extremos, verificamos empiricamente essa situação. Se o pagamento e a multa de descompromisso propostos forem muito altos e a multa de descompromisso requirida muito baixa, então muitos agentes responderão ao anúncio com ofertas válidas, havendo mais hipóteses de obter uma boa proposta. No entanto os ganhos com esta proposta acabarão por diluir-se devido à grande quantidade de multas de descompromisso a pagar. Se o pagamento e a multa de descompromisso propostos forem muito baixos e a multa de descompromisso requirida muito alta, poucos agentes responderão à oferta, mas qualquer oferta feita representará um ganho efectivo maior, devido ao baixo pagamento, à baixa multa de descompromisso e também porque temos a certeza que o outro agente não se descomprometerá.

Quando o agente faz um anúncio para para agentes da mesma organização, a condição de eligibilidade pode ser simplesmente

$$\Delta_{adi\_agente\_remoto} \mathcal{U}(T) > \Delta_{adi} \mathcal{U}(T), \tag{4.16}$$

para indicar que o agente remoto só deve fazer uma oferta se a variação na utilidade por executar as tarefas é superior à do agente que faz o anúncio, ou seja, se é mais vantajoso para a organização executar as tarefas nesse agente.

O anúncio de contratos a longo prazo é feito quando o agente decide criar um nodo virtual, isto é, quando decide alterar a estruturação organizacional, não correspondendo à situação aqui apresentada. As decisões de alteração da estruturação organizacional são discutidas na secção 4.8.7. A oferta de disponibilidade, que também corresponde a um anúncio, pode ser vista de duas perspectivas: como um desejo de alterar a estruturação organizacional, reduzindo a capacidade de um nodo, ou como uma forma de publicidade, pois quando um agente recebe uma oferta de disponibilidade, em que são referidos os recursos e os seus preços, pode estimar nesse instante novos valores para $\Delta_{executa}\mathcal{U}(T)$, que serão considerados na determinação de novos mercados, podendo levar à inclusão da organização que anuncia. A primeira perspectiva será discutida na secção 4.8.7. A análise da influência da publicidade fica para futuro trabalho de investigação.

4.8.4 Ofertas

Quando surge um anúncio para um conjunto de tarefas T, começamos por determinar $\Delta_{adi}\mathcal{U}(T)$ utilizando a expressão 4.13. Este valor representa o que o agente vai perder se executar as tarefas. A recompensa por executar estas tarefas é o pagamento prometido pelo agente anunciante. Para saber se a recompensa é suficiente é necessário determinar $\Delta_{pagamento}\mathcal{U}(T)$, a variação na utilidade causada pelo hipotético pagamento referente à execução de T. A expressão para o cálculo desse valor é

$$\begin{aligned}\Delta_{pagamento}\mathcal{U}(T) &= \mathcal{F}(Dinheiro_{act} + Dinheiro_{prometido}, P_1, \ldots, P_{Np}) - \\ &\quad \mathcal{F}(Dinheiro_{act}, P_1, \ldots, P_{Np}),\end{aligned}$$

onde $Dinheiro_{act}$ representa a quantidade de dinheiro existente actualmente e $Dinheiro_{prometido}$ o pagamento prometido pela execução de T. Assim, uma das condições para que o agente faça uma oferta é

$$\Delta_{pagamento}\mathcal{U}(T) \geq \Delta_{adi}\mathcal{U}(T). \tag{4.17}$$

Quando a condição 4.17 se verifica, o agente tem uma certa margem de manobra quanto ao pagamento a ser pedido. O valor mínimo que o agente pode pedir sem que prejudique a sua utilidade, obtem-se resolvendo a seguinte inequação em ordem a $Dinheiro_{minimo}$:

$$\Delta_{adi}\mathcal{U}(T) = \mathcal{F}(Dinheiro_{act} + Dinheiro_{minimo}, P_1, \ldots, P_{Np}) - \mathcal{F}(Dinheiro_{act}, P_1, \ldots, P_{Np}).$$

Se o agente é aquele que arrisca pede o máximo, ou seja, $Dinheiro_{prometido}$. Se é cuidadoso pede o mínimo, $Dinheiro_{minimo}$. Assim, $Dinheiro_{oferta}$ (a oferta efectuada pelo agente) pode ser definida como

$$Dinheiro_{oferta} = Dinheiro_{prometido} - \alpha \times (Dinheiro_{prometido} - Dinheiro_{minimo}).$$

Quando um se tratar de um anuncio feito por um agente da mesma organização, basta que o agente verifique a condição de eligibilidade especificada em 4.16 para que faça uma oferta. Esta deverá ser feita de acordo com o requirido pelo anunciante.

Para além da condição 4.17 também é necessário que a multa de descompromisso prometida seja satisfatória para o agente fazer uma oferta. Uma multa satisfatória deve aumentar com o tempo de tal forma que, se o agente anunciante se descomprometer no instante final da execução da tarefa, o valor total do pagamento será reavido. No entanto, podemos pedir uma multa de descompromisso mais baixa se soubermos que o agente raramente se descompromete. Assim, se t_{fim} for o instante final de execução de T, $Multa_{prometida}(t)$ for uma função do tempo que indica a multa prometida pelo anunciante e $P_{descompromisso}$ fôr a probabilidade de o agente anunciante quebrar o compromisso, então a outra condição para que o agente faça uma oferta pode ser definida como

$$\forall t_{act} \leq t \leq t_{fim} \quad P_{descompromisso} \times \frac{(t - t_{act}) \times Dinheiro_{oferta}}{t_{fim} - t_{act}} \leq Multa_{prometida}(t) \tag{4.18}$$

Para calcular $P_{descompromisso}$ recorremos, mais uma vez, à memória do agente. Sendo N o número de contratos feitos com o agente anunciante e X o número de vezes que ele se descomprometeu, então [Fernandes, 1993]

$$P_{descompromisso} = \frac{X}{N} + z_{1-\alpha} \times \sqrt{\frac{\frac{X}{N}(1 - \frac{X}{N})}{N}}.$$

Concluíndo, o agente faz uma oferta se se verificarem simultaneamente 4.17 e 4.18, sendo o pagamento requerido $Dinheiro_{oferta}$ e a multa de descompromisso requerida

$$Multa_{requerida}(t) = P_{descompromisso} \times \frac{(t - t_{act}) \times Dinheiro_{oferta}}{t_{fim} - t_{act}}.$$

4.8.5 Atribuições de Contratos

Quando chega uma oferta válida, é necessário verificar se o pagamento requirido compensa dado o novo estado do agente. Assim, começa-se por recalcular $\Delta_{adi}\mathcal{U}(T)$ (expressão 4.13). Depois, considerando A o agente que faz a oferta, O o conjunto de agentes que ainda não enviaram oferta, $\mathcal{E}(\Delta^O \mathcal{U}(T))$ a esperança de variação na utilidade devido às possiveis ofertas que se irão receber, e $\Delta^A_{aceita}\mathcal{U}(T)$ a variação na utilidade se se aceitar a oferta de A, então a oferta é aceite se

$$\Delta^A_{aceita}\mathcal{U}(T) > \Delta_{adi}\mathcal{U}(T),$$

e se

$$\Delta^A_{aceita}\mathcal{U}(T) > \mathcal{E}(\Delta^O \mathcal{U}(T)) + \Delta^{\{A\}}_{desc}\mathcal{U}(T).$$

Se estas condições não se verificarem a oferta é rejeitada. $\Delta^A_{aceita}\mathcal{U}(T)$, se A fôr um agente de outra organização calcula-se como

$$\begin{aligned}\Delta^A_{aceita}\mathcal{U}(T) &= (\mathcal{F}(Dinheiro_{act} - Dinheiro_{requerido}, P_1, \ldots, P_{Np}) - \\ &\quad \mathcal{F}(Dinheiro_{act}, P_1, \ldots, P_{Np})) + \Delta^A_{envia}\mathcal{U}(T),\end{aligned}$$

e se fôr um agente da mesma organização como

$$\Delta^A_{aceita}\mathcal{U}(T) = \Delta_{adi\_agente\_remoto}\mathcal{U}(T) + \Delta^A_{envia}\mathcal{U}(T).$$

Se definirmos $\mathcal{E}(\Delta^C_{com\_oferta}\mathcal{U}(T))$ (expressão 4.15) como a variação de utilidade esperada se o conjunto de agentes C fizer ofertas válidas então $\mathcal{E}(\Delta^O \mathcal{U}(T))$ está limitado inferiormente por

$$\mathcal{E}^-(\Delta^O \mathcal{U}(T)) = \min_{C \subseteq O}(\mathcal{E}(\Delta^C_{com\_oferta}\mathcal{U}(T)),$$

e superiormente por

$$\mathcal{E}^+(\Delta^O \mathcal{U}(T)) = \max_{C \subseteq O}(\mathcal{E}(\Delta^C_{com\_oferta}\mathcal{U}(T)).$$

Como um agente que arrisca espera sempre um maior retorno do anúncio, temos

$$\mathcal{E}(\Delta^O \mathcal{U}(T)) = \mathcal{E}^+(\Delta^O \mathcal{U}(T)) - \alpha \times (\mathcal{E}^+(\Delta^O \mathcal{U}(T)) - \mathcal{E}^-(\Delta^O \mathcal{U}(T))). \tag{4.19}$$

É de reparar que nestas expressões não se incluem os custos de ter enviado os anúncios nem os custos de descompromisso referentes a ofertas passadas. Incluir esses custos seria errado pois no instante actual eles já são irrecuperáveis. Vejamos o seguinte exemplo para uma melhor compreensão. Um agente decide anunciar uma tarefa a um só agente, custando-lhe esse anúncio 5 UM. Quando a oferta chega verifica-se que o custo de executar localmente a tarefa é de 20 UM e que o valor da oferta é de 18 UM. Como as 5 UM já estão consumidas a melhor decisão é aceitar a oferta, mesmo que no total o contrato tenha custado 23 UM. Este problema, que também se verifica em relação às multas de descompromisso já pagas, é no entanto resolvido na fase de anúncio do contrato com a definição dinâmica da extensão do mercado. Neste exemplo, assumindo que o agente previa que a oferta ia ser de 18 UM nunca teria sido feito o anuncio a esse agente, dado que a esperança de custo era de 23 UM.

4.8.6 O *Scheduler*

A função do *scheduler* é determinar $\mathcal{U}(Tarefas)$, ou seja, planos de execução de tarefas que maximizem a utilidade da organização. Estes planos são calculados heuristicamente. Devido às pressões temporais a que o agente está sujeito, e devido à sua racionalidade limitada que impede a obtenção instantânea de $\mathcal{U}(Tarefas)$, o *scheduler* deve estar preparado para fornecer resultados intermédios. Estes resultados representam sucessivas aproximações ao resultado óptimo. Uma forma de o agente determinar quando deve parar o cálculo, consiste em o *scheduler* atribuir a cada resultado intermédio um valor que indique a sua qualidade. Se este valor estiver contido no intervalo entre 0 e 1, tendo o resultado óptimo a qualidade 1, o agente pode parar o cálculo quando a qualidade for superior a α. Desta forma, um agente que arrisca utilizará resultados mais imprecisos (mas calculados em pouco tempo) e um agente cuidadoso utilizará resultados mais correctos (mas obtidos mais tarde).

A separação entre o *scheduler* e o coordenador deve-se, fundamentalmente, a três razões [Garvey et al., 1994]:

- **Modularidade**. É difcil encapsular todo o processo de decisão num só módulo, especialmente tratando-se de funcionalidade tão distintas como o controlo da cooperação e a elaboração de planos de execução. A separação em módulos que comunicam através de um interface bem definido, apresenta-se como uma solução viável, já que permite isolar as características particulares de cada problema.

- **Eficiência**. Este ponto está directamente relacionado com o anterior, já que a possibilidade de desenvolvimento dos módulos em separado permite, não só utilizar a tecnologia mais adequada em cada um dos módulos, mas, principalmente, proceder facilmente a continuas optimizações.

- **Reutilização**. Um *scheduler* que permite determinar planos de execução óptimos dados um conjunto de tarefas, um conjunto de recursos e uma função de utilidade (que permite medir a qualidade de cada plano), é util em muitos domínios para além da alocação distribuída de tarefas num sistema multi-agente. Se o protocolo de interface entre o *scheduler* e o coordenador fôr suficientemente genérico (ou mesmo standard), torna-se possivel a sua utilização noutros domínios, assim como, a incorporação nesta arquitectura de outros *schedulers* já desenvolvidos.

4.8.7 Quando As Coisas Não Correm Bem ...

Quando as coisas não estão a correr bem ao agente, existem duas hipóteses para tentar melhorá-las:

- **Mudar a estruturação organizacional**. Existem várias alternativas para o fazer, nomeadamente, mudar a sua localização, modificar o conjunto dos nodos geridos (incluindo a criação de nodos virtuais ou a redução da capacidade de um nodo através de um contrato a longo prazo), modificar o conjunto das linhas geridas, modificar o conjunto das organizações com que se relaciona, juntar-se a outro agente ou fragmentar-se em vários agentes.

- **Mudar o seu comportamento**. Existem três parâmetros que podem ser alterados para mudar o comportamento do agente. O parâmetro *Idade* permite variar a taxa de actualização dos parâmetros da função de utilidade, com a consequente variação da carga de mensagens na infra-estrutura de comunicações. ΔT permite variar o tamanho da memória do agente dando ao agente uma atitude mais ou menos exploradora e uma maior ou menor capacidade de reação. Finalmente, variando α variamos a atitude do agente perante o risco.

Para se efectuar uma mudança, é necessário ter em conta dois aspectos: como é que se sabe que as coisas não estão a correr bem e como é que decidimos qual a melhor mudança a efectuar. Ambos os casos vão ser aprofundados em trabalho futuro, sendo, no entanto, possível adiantar algumas ideias. O agente pode tentar mudar alguma coisa sempre que o valor da função de utilidade varie negativamente mais do que um determinado Δ deste a última tentativa de mudança. A escolha da mudança a fazer pode ser feita por tentativas, testando-se para cada uma delas se, caso se repetissem todos os eventos presentes na memória do agente, o grau de satisfação actual seria superior.

4.9 Agente de Controlo

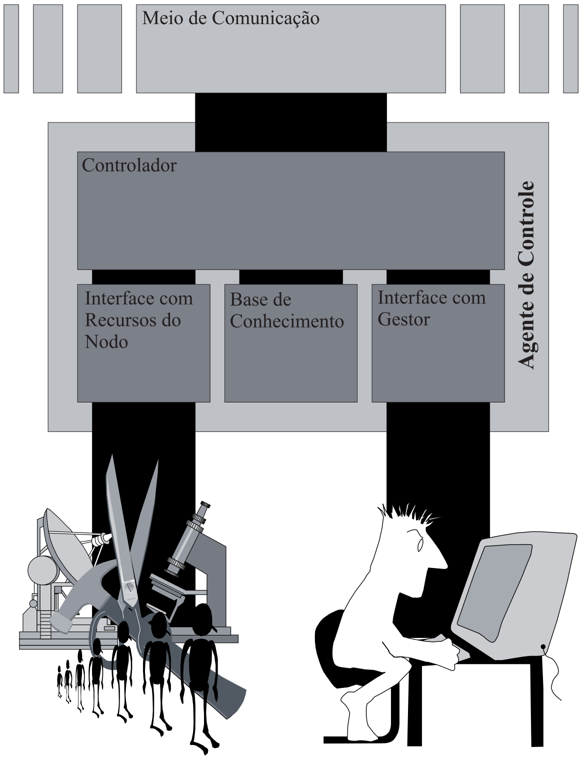


Figure 4.9: Arquitectura de um Agente de Controlo

Um agente de controlo é composto por quatro módulos (ver figura 4.9):

- **Controlador.** Responsável por encaminhar as mensagens que passam ou são originadas no nodo. Quando o destino de uma destas mensagens é desconhecido e é possível saber se a mensagem se refere a determinado recurso ou organização, cria um novo agente de alocação no nodo para receber essa mensagem. Também é responsável pelo envio de respostas negativas quando sabe que um agente desapareceu. Se o agente que gere o seu nodo está temporariamente desaparecido tenta executar localmente as tarefas criadas no nodo.

- **Interface com Recursos do Nodo.** Responsável pela monitorização dos recursos, encarregando-se da execução das tarefas enviadas pelo agente que gere o nodo dentro dos parâmetros por este definidos. No fim da execução recolhe as respectivas estatísticas. Também detecta o aparecimento de novas tarefas.

- **Base de Conhecimento.** Serve para guardar a estruturação organizacional actual.

- **Interface com Gestor.** Informa o gestor do estado do sistema e do nível de satisfação da organização. Recolhe as parametrizações definidas por este e transmite-as ao controlador para serem difundidas para todos os agentes.

Um dos melhoramentos que se pode fazer no sistema, consiste em os agentes de controlo puderem armazenar as estatísticas de execução das tarefas, para que, caso se dê uma falha no nodo do agente de alocação respectivo, não se perca informação. Os novos agentes de alocação, ao iniciarem as suas actividades, requisitariam essa informação aos agentes de controlo dos nodos que gerem. O armazenamento desta informação também permitiria ao controlador fazer uma pequena análise para determinar se compensava enviar a tarefa para o agente de alocação. Se a influência de uma tarefa na maximização da satisfação global é muito pequena, nem vale a pena perder o tempo a comunicá-la ao agente de alocação, sendo imediatamente executada.

4.10 Aplicação do Modelo à Distribuição de Carga

Aplicar esta estratégia de cooperação a um problema em particular consiste, principalmente, em definir a função de utilidade da organização e adaptar o módulo de interface com os recursos (agente de controlo) ao tipo de nodo utilizado no sistema.

No caso do problema de distribuição de carga subjacente a este estudo, a função de utilidade deverá ser:

$$\begin{aligned}\mathcal{U} &= \mathcal{F}(Dinheiro, NumeroTarefas, TempoTotalTarefas,\\ &\quad NumeroTarefasComPrazo, NumeroTarefasGarantidas)\end{aligned}$$

Estes parâmetros respeitam a condição 4.7 e permitem representar correctamente os objectivos definidos na secção 3.4.9: o primeiro representa o lucro da organização, os segundo e terceiro permitem definir o tempo médio de resposta das tarefas e os dois últimos a percentagem de tarefas garantidas (executadas dentro do prazo).

O módulo de interface com os recursos deve estar ligado directamente ao *scheduler* do sistema operativo, para permitir interceptar novas tarefas criadas, e para poder definir as quantidades de recursos activos atribuídas às tarefas.

Para concluir a apresentação da estratégia de distribuição desenvolvida, falta apenas referir em que medida é que resolve os problemas apontados na secção 3.4.10 e, em que medida segue as sugestões apontadas na mesma secção. Assim, a estratégia desenvolvida apresenta as seguintes características:

- possibilita o comércio dos recursos existentes na organização;
- previne a não intrusão, pois uma tarefa apenas é executada noutra organização com o consentimento desta e porque as organizações podem controlar o acesso às suas linhas de comunicação subindo ou baixando os seus custos;
- integra indiscriminadamente tarefas com e sem descrição dos recursos necessários, pois transforma o primeiro problema no segundo à custa da memorização das necessidades passadas de recursos das tarefas;
- considera explicitamente as capacidades dos diferentes recursos e infra-estruturas de comunicações existentes na organização;
- permite, recorrendo à definição de uma função de utilidade, considerar diversos objectivos a atingir com a distribuição de carga;
- contorna a necessidade de recolha de informação para a distribuição de tarefas recorrendo à memorização dos resultados de interações anteriores;
- a única informação que é necessário transmitir é referente ao estado da satisfação da função de utilidade nos diversos nodos, sendo a influência negativa destas transmissões bastante minimizada devido ao aproveitamento das mensagens de negociação para a transmitir, e devido à possibilidade de alterar dinâmicamente a taxa de actualização em função da estabilidade da informação;
- utiliza uma estruturação organizacional dinâmica que se adapta às características particulares de cada organização e às caracteristicas dinâmicas do ambiente envolvente;
- utiliza políticas de distribuição por negociação para possibilitar o comércio electrónico e a não imposição de tarefas; e
- a utilização extensiva de mecanismos adaptativos, principalmente com a inovação de aplicar essa adaptatividade à própria arquitectura do sistema.

Para uma melhor comparação com a abordagem convencional de distribuição de carga, também se apresenta na tabela 4.1 a classificação da estratégia desenvolvida segundo os parâmetros definidos no capítulo 3.

Estratégia global	Dinâmico	
	Estruturação Organizacional Dinâmica	
	Global	
Objectivos	Tipo	Minimização tempos de resposta, maximização tarefas garantidas e maximização do lucro
Transferência	Quem inicia	Adaptativo entre origem, receptor e simétrico
	Quando se inicia	Após mudança de estado: chegada de tarefa
	Tipo	Mercado definido dinâmicamente
Selecção	Quando se transfere	Não Preemptiva
	Redistribuição	Ilimitada
	Tarefa escolhida	Tarefas criadas recentemente
Localizaçao	Qual o nodo	Negociação
Informação	Tipo	Medidas comunicadas e conjecturas a partir da informação memorizada
	Obtenção	A pedido
	Medida de carga	Parâmetros influentes na função de utilidade

Table 4.1: Classificação da Estratégia de Distribuição de Carga Desenvolvida

Chapter 5

Simulador de Estratégias de Distribuição de Carga

A análise quantitativa de um novo mecanismo de distribuição de carga é um processo fundamental para se verificar se este atingiu os objectivos inicialmente propostos. Esta análise, sempre que possível, deve ser feita antes da implementação definitiva do mesmo, evitando-se, desta forma, o desperdício de recursos humanos e materiais que poderiam advir de uma análise incompleta ou errada. Devido à profusão de estratégias de distribuição concebidas nos últimos anos, algumas delas apresentadas no capítulo 3, esta análise torna-se ainda mais importante pois existe a possibilidade de haver um mecanismo cujo desempenho seja superior ao do novo mecanismo desenvolvido. Esta possibilidade leva à necessidade de se efectuar um estudo comparativo com o objectivo de determinar o real valor de um mecanismo de distribuição.

Para se efectuar esta análise escolheu-se o método da simulação devido à grande flexibilidade que apresenta, quer na definição do sistema de processamento subjacente, quer na definição da carga a que é sujeito. O simulador foi desenvolvido em PROLOG [Carlsson and Widen, 1993] devido à reconhecida capacidade desta linguagem para a definição de comportamentos algorítmicos complexos, necessária para modelar o agente de alocação de carga apresentado no capítulo 4.

5.1 Introdução

Existem dois grandes métodos para se efectuar a análise quantitativa à priori de um mecanismo de distribuição de carga:

- **O método analítico.** Neste método as estratégias de distribuição são modeladas através de fórmulas analíticas, assumindo-se a possibilidade de quantificar todas as variáveis fundamentais do sistema e também o conhecimento das expressões que definem a dependência relativa dessas variáveis [Rodrigues, 1988]. A maior parte dos modelos utilizados baseiam-se na utilização de redes de filas para representar o sistema de processamento e as tarefas em execução. Consultar [Eager et al., 1986, Theimer and Lantz, 1989] como exemplos da utilização deste método.

- **O método da simulação numérica.** Neste método não se assume o conhecimento detelhado das equações que relacionam as variáveis do sistema, assumindo-se apenas o conhecimento do funcionamento interno dos seus diversos componentes, ou seja, do mecanismo de distribuição. Os resultados são implicitamente obtidos como consequência da interação dos diversos componentes do sistema reproduzidos computacionalmente. Ver, como exemplos, [Ramamritham et al., 1989, Kremien et al., 1993, Kremien and Kramer, 1992, Zhou, 1988].

Existem autores que recorrem à construção de protótipos dos mecanismos desenvolvidos procedendo depois à análise do seu desempenho num sistema físico real. O sistema é sujeito a uma carga artificial, ou sintética, analisando-se depois o seu comportamento quando em presença de diversos mecanismos de distribuição. O protótipo não precisa, como o seu próprio nome indica, de incluir todas as funcionalidades necessárias num mecanismo de distribuição, nomeadamente as que garantem a robustez do mecanismo. No entanto, necessita que as funcionalidades essenciais de distribuição estejam já implementadas, não sendo, por isso, um verdadeiro método de análise à priori. Como exemplos de aplicação deste método ver, por exemplo, [Kunz, 1991, Lüling et al., 1992].

Cada um destes três métodos apresenta, simultaneamente, vantagens e desvantagens, nomeadamente:

- devido à dificuldade em encontrar as expressões que modelam a relação entre as diversas variáveis do sistema, o método analítico apenas pode ser aplicado na análise de estratégias de balanceamento muito simples e nunca em estratégias caracterizadas por um comportamento complexo, como, por exemplo, a apresentada no capítulo 4;

- a grande vantagem do método analítico consiste na rapidez de obtenção de resultados, mesmo quando a solução das expressões encontradas tem de ser obtida através de métodos numéricos iterativos;

- a simulação com carga artificial implica um grande esforço no desenvolvimento dos protótipos dos mecanismos em análise, pois estes têm de operar sobre um sistema real;

- este método também implica que o sistema de processamento seja dedicado exclusivamente ao processo de análise, situação fácil de se conseguir quando se trata de uma máquina paralela, mas muito difícil em sistemas distribuídos, devido à grande quantidade de utilizadores e devido ao grande número de tarefas do sistema operativo em execução;

- quer o método analítico, quer o método de simulação com carga artificial, são muito rígidos quanto possibilidade de analisar o mecanismo sobre diferentes configurações físicas, devido, respectivamente, à complexidade de modelar as expressões inerentes a configurações mais complexas, e à dificuldade ou impossibilidade de alterar fisicamente a configuração do sistema de processamento subjacente;

- o método da simulação numérica requer a concepção de um simulador que suporte os diversos mecanismos de distribuição em análise, sendo, no entanto, esta tarefa mais simples do que a construção dos protótipos para a simulação com carga artificial;

- este método é o mais flexível quanto à capacidade de analisar diferentes configurações do sistema subjacente, diferentes padrões de carga nesse sistema e diferentes configurações dos elementos de distribuição;

- o tempo requirido pela simulação numérica pode ser relativamente elevado se o número de entidades envolvidas na simulação for muito grande.

O método escolhido para a análise do mecanismo de distribuição desenvolvido foi o da simulação numérica, principalmente devido à flexibilidade que apresenta ao nível da configuração do sistema subjacente, da possibilidade de definir diferentes padrões de carga e diferentes configurações dos elementos de distribuição. Obviamente, que esta flexibilidade não está implicita no método mas no simulador respectivo. No entanto, é evidente que é muito mais fácil de se conseguir com este método do que com os outros dois.

Formalmente, o modelo de simulação que vai ser usado caracteriza-se por ser [Rodrigues, 1988]:

- **Abstracto.** Pois procede-se a uma modelação computacional da realidade.

- **Numérico.** Porque o modelo lida com valores numéricos, mas sem que se conheçam as expressões que relacionam as variáveis do sistema.

- **Dinâmico.** Pois representa a forma como o tempo modifica o sistema real.
- **Explícito.** Porque o comportamento interno dos suas entidades é conhecido e bem detalhado.
- **Estocástico.** Porque os parâmetros de funcionamento do sistema não são determinísticos, mas valores aleatórios provenientes quer de distribuições teóricas, quer de distribuições empíricas (histogramas de probabilidade).

5.2 Procedimento da Simulação

Por questões de flexibilidade, e devido ao facto da simulação poder ser um processo extremamente demorado, dividiu-se o processo em duas fases distintas (figura 5.1), nomeadamente:

- a simulação propriamente dita, onde a partir de uma determinada configuração, armazenada num ficheiro, se geram os resultados da mesma, sendo estes também armazenados em ficheiro; e
- a análise de resultados, feita mais tarde recorrendo aos ficheiros de resultados gerados pelo simulador.

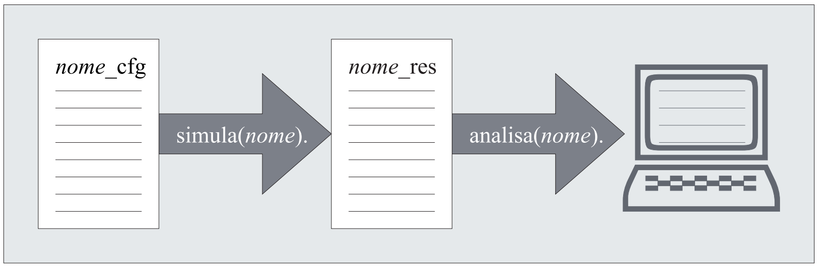


Figure 5.1: Pocedimento da Simulação

Entre outras vantagens, esta separação permite que o processo de simulação seja feito em *batch*, porque não necessita da presença do utilizador, quer devido à existência dos ficheiros de configuração, quer devido ao facto de a visualização dos resultados apenas ser feita mais tarde. Outra vantagem, que advém desta separação, consiste em não ter de se calcular todas as estatísticas durante o processo de simulação, o que o tornaria ainda mais lento e obrigaria à definição à priori de todas as estatísticas necessárias. Se assim não fosse, de todas as vezes que se pretende-se obter uma nova estatística seria necessário executar nova simulação.

Todas as rotinas necessárias no processo de simulação, nomeadamente o próprio simulador e as rotinas para análise estatística dos resultados, foram desenvolvidas na versão da linguagem PROLOG desenvolvida pela SICStus [Carlsson and Widen, 1993]. Esta linguagem não é particularmente rápida, especialmente na versão interpretada utilizada, não sendo por isso, à partida, uma grande candidata à construção de um simulador cujo grande problema é o tempo de execução. No entanto, esta linguagem apresenta grandes vantagens, nomeadamente:

- capacidade de modelar comportamentos algorítmicos complexos e grandes capacidades de inferência, necessárias para modelar a complexidade crescente dos mecanismos de distribuição de carga, nomeadamente, do agente computacional apresentado no capítulo 4, e modelar fielmente as características do sistema de processamento subjacente;
- pequenos tempos de desenvolvimento, necessários devido ao pouco tempo disponível para a concepção do simulador, e que permitem modelar um maior número de mecanismos de distribuição, tornando a análise mais efectiva.

A primeira etapa para se efectuar uma simulação consiste na elaboração do ficheiro de configuração do sistema em análise, descrito na secção 5.3, cujo nome deverá ter a seguinte sintaxe: *Nome_da_Simulacao*`_cfg`. Depois, dentro do interpretador de PROLOG, invoca-se o simulador através do predicado `simula(`*Nome_da_Simulacao*`)`, que gerará um ficheiro de resultados com o nome *Nome_da_Simulacao*`_res`. Este ficheiro poderá mais tarde ser analisado estatisticamente recorrendo a uma biblioteca de predicados de análise. O funcionamento interno do simulador é descrito na secção 5.4.

5.3 Ficheiro de Configuração

O ficheiro de configuração é uma das partes fundamentais do processo de simulação, porque expressa toda a flexibilidade do simulador e porque constitui a interface entre o simulador e o utilizador final. A sintaxe deste ficheiro é apresentada na figura 5.2, onde se recorre à notação BNF [Gries, 1981, páginas 304–309] devido à facilidade de compreensão que possui (os termos delimitados por "/ /" representam termos não terminais cuja expansão é trivial e não relevante para a especificação). Na figura 5.3 é apresentada a configuração correspondente ao sistema distribuído multi-organizacional da figura 4.1. Na figura 5.4 é apresentada a configuração correspondente a uma máquina paralela com 9 nodos e uma topologia em matriz. No primeiro exemplo não se efectua qualquer espécie de distribuição de carga. No segundo recorre-se à estratégia de distribuição aleatória tal como foi apresentada na secção 3.3.1.

Para facilitar o processo de simulação, o simulador procede a uma análise sintática e semântica do ficheiro de configuração, reportando os erros respectivos ao utilizador.

A seguir apresentam-se mais detalhadamente os diversos parâmetros que caracterizam as entidades presentes no ficheiro de configuração, explicando-se em que medida é que influenciam o seu comportamento durante a simulação. Também serão apresentadas as afirmações feitas quanto ao comportamento interno de cada entidade. Não é apresentada qualquer explicação em relação à entidade organização (grupo), pois esta é uma entidade perfeitamente passiva, correspondendo apenas a uma relação de dados.

5.3.1 Controlo da Simulação

Nesta secção apresentam-se os parâmetros dos predicados de controlo da simulação, comuns a todas as entidades da mesma, nomeadamente:

- **Semente.** Ajusta a semente do gerador de números aleatórios utilizado. É essencial para permitir efectuar diferentes simulações com a mesma configuração, aumentando assim o nível de confiança nos resultados da mesma.

- **Inicio de monitorização.** Utilizado pelos predicados de análise estatística dos resultados. Indica a partir de que instante serão consideradas as tarefas chegadas ao sistema, servindo para ignorar o período de inicialização em que o sistema ainda não atingiu um estado estacionário [Rodrigues, 1988], nomeadamente no respeitante ao número de tarefas no sistema.

- **Fim de simulação.** Indica quando termina a simulação.

- **Imprime tempo.** Indica de quanto em quanto tempo será comunicado o tempo actual de simulação, e de quanto em quanto tempo será feito o *garbage collection* das *stacks* do PROLOG [Carlsson and Widen, 1993].

- **Tamanho das mensagens de controlo.** Indica o tamanho (em bytes) das mensagens de controlo trocadas entre as entidades de simulação, como por exemplo, os pedidos de distribuição de tarefas ou as sondagens. Deve ser ponderado com cuidado para não desvirtuar os resultados finais.

```
<configuracao> ::= {<comentario>} <controlo> {<entidade>}

<comentario> ::= % /string/

<controlo> ::= <semente> <inicio> <fim> <monitorizacao> <mensagens_controle>
               <cabecalho>
<semente> ::= semente(/inteiro/, /inteiro/, /inteiro/).
<inicio> ::= inicio_simulacao(/inteiro/).
<fim> ::= fim_simulacao(/inteiro/).
<monitorizacao> ::= imprime_tempo(/inteiro/).
<mensagens_controle> ::= tamanho_mensagens_controle(/inteiro/).
<cabecalho> ::= cabecalho(/inteiro/).

<entidade> ::= <comentario> | <grupo> | <nodo> | <linha> | <carga> | <tarefa> |
               <agente>

<grupo> ::= grupo(/identificador/, "/string/").

<nodo> ::= nodo(/identificador/, /identificador/, <lista_identificadores>,
           <recursos_nodo>, (/inteiro/, /inteiro/)).
<lista_identificadores> ::= [] | [{/identificador/,} /identificador/]
<recursos_nodo> ::= [{<recurso_nodo>,} <recurso_nodo>]
<recurso_nodo> ::= (/identificador/, /inteiro/, <tipo>)
<tipo> ::= activo | passivo

<linha> ::= linha(/identificador/, /identificador/, /inteiro/).

<carga> ::= carga(/identificador/, <distribuicao1>, <lista_tarefas>).
<distribuicao1> ::= exactamente(/inteiro/) | expneg(/inteiro/)
<lista_tarefas> ::= [{<tarefa_carga>,} <tarefa_carga>]
<tarefa_carga> ::= (/inteiro/, /identificador/)

<tarefa> ::= tarefa(/identificador/, <distribuicao1>, <distribuicao2>,
             <recursos_tarefa>, (<distribuicao2>, <distribuicao2>), <distribuicao1>).
<distribuicao2> ::= <distribuicao1> | nao_tem
<recursos_tarefa> ::= [{<recurso_tarefa>,} <recurso_tarefa>]
<recurso_tarefa> ::= (/identificador/, <distribuicao1>, <tipo>)

<agente> ::= <agente_aleatorio> | <faltam os outros>

<agente_aleatorio> ::= aleatorio(/identificador/, /identificador/, /inteiro/,
                       /inteiro/, <lista_identificadores_nao_vazia>).
<lista_identificadores_nao_vazia> ::= [{/identificador/,} /identificador/]
```

Figure 5.2: Especificação em Notação BNF da Sintaxe de um Ficheiro de Configuração

```
% Controlo da simulacao
semente(27134, 9213, 17773).
inicio_monitorizacao(10000).
fim_simulacao(200000).
imprime_tempo(1000).
tamanho_mensagens_controle(100).
cabecalho(50).
% Grupos
grupo(organizacao1, "Organizacao XXX").
grupo(organizacao2, "Organizacao YYY").
grupo(organizacao3, "Organizacao ZZZ").
% Nodos
nodo(nodo1, organizacao1, [linha1], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo2, organizacao1, [linha1], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo3, organizacao1, [linha1], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo4, organizacao1, [linha1], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo5, organizacao1, [linha1], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo6, organizacao1, [linha1], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo7, organizacao1, [linha1, linha5], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 200000, carga1)], (100, 50)).
nodo(nodo8, organizacao2, [linha2], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo9, organizacao2, [linha2], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo10, organizacao2, [linha2], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo11, organizacao2, [linha2], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo12, organizacao2, [linha2], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo13, organizacao2, [linha2], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo14, organizacao2, [linha2, linha3, linha5], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 200000, carga1)], (100, 50)).
nodo(nodo15, organizacao2, [linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo16, organizacao2, [linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo17, organizacao2, [linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo18, organizacao2, [linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo19, organizacao2, [linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo20, organizacao2, [linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo21, organizacao2, [linha3, linha6], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 200000, carga1)], (100, 50)).
nodo(nodo22, organizacao3, [linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo23, organizacao3, [linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo24, organizacao3, [linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo25, organizacao3, [linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo26, organizacao3, [linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo27, organizacao3, [linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)], [(0, 200000, carga1)], (100, 50)).
nodo(nodo28, organizacao3, [linha4, linha6], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 200000, carga1)], (100, 50)).
% Linhas
linha(linha1, nodo7, 10000).
linha(linha2, nodo14, 10000).
linha(linha3, nodo21, 10000).
linha(linha4, nodo28, 10000).
linha(linha5, nodo7, 10000).
linha(linha6, nodo28, 10000).
% Tipos de carga
carga(carga1, expneg(1000), [(100, tarefa1)]).
% Tipos de tarefas
tarefa(tarefa1, expneg(700), nao_tem, [(cpu, exactamente(1000), activo)], (nao_tem, nao_tem), exactamente(100)).
```

Figure 5.3: Exemplo de Ficheiro de Configuração de um Sistema sem Distribuição de Carga

```
% Controlo da simulacao
semente(27134, 9213, 17773).
inicio_monitorizacao(1000).
fim_simulacao(20000).
imprime_tempo(1000).
tamanho_mensagens_controle(100).
cabecalho(50).
% Grupos
grupo(grupo1, "Organizacao XXX").
% Nodos
nodo(nodo1, grupo1, [linha1, linha3], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo2, grupo1, [linha1, linha2, linha4], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo3, grupo1, [linha2, linha5], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo4, grupo1, [linha3, linha6, linha8], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo5, grupo1, [linha4, linha6, linha7, linha9], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo6, grupo1, [linha5, linha7, linha10], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo7, grupo1, [linha8, linha11], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo8, grupo1, [linha9, linha11, linha12], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
nodo(nodo9, grupo1, [linha10, linha12], [(cpu, 1000, activo), (memoria, 20000, passivo)],
     [(0, 20000, carga1)], (100, 50)).
% Linhas
linha(linha1, nodo1, 20000).
linha(linha2, nodo1, 20000).
linha(linha3, nodo1, 20000).
linha(linha4, nodo1, 20000).
linha(linha5, nodo1, 20000).
linha(linha6, nodo1, 20000).
linha(linha7, nodo1, 20000).
linha(linha8, nodo1, 20000).
linha(linha9, nodo1, 20000).
linha(linha10, nodo1, 20000).
linha(linha11, nodo1, 20000).
linha(linha12, nodo1, 20000).
% Tipos de carga
carga(carga1, expneg(1000), [(50, tarefa1), (50, tarefa2)]).
% Tipos de tarefas
tarefa(tarefa1, expneg(700), nao_tem, [(cpu, exactamente(1000), activo)],
        (nao_tem, nao_tem), exactamente(100)).
tarefa(tarefa2, expneg(700), nao_tem, [(cpu, exactamente(1000), activo)],
        (exactamente(100), exactamente(100)), exactamente(100)).
% Agentes de Distribuição Aleatórios
aleatorio(aleatorio1, nodo1, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio2, nodo2, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio3, nodo3, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio4, nodo4, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio5, nodo5, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio6, nodo6, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio7, nodo7, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio8, nodo8, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
aleatorio(aleatorio9, nodo9, 1, 1, [nodo1, nodo2, nodo3, nodo4, nodo5, nodo6, nodo7, nodo8, nodo9]).
```

Figure 5.4: Exemplo de Ficheiro de Configuração de uma Máquina Paralela Com Distribuição Aleatória de Carga

- **Cabeçalho.** Indica o tamanho do cabeçalho de uma mensagem. Constitui o limite inferior no tamanho de uma mensagem que circula na rede. Depende do protocolo de comunicação que se pretenda simular.

5.3.2 Nodos

Os parâmetros que caracterizam um nodo são os seguintes:

- **Identificação.** Caracteriza univocamente cada nodo do sistema.

- **Organização a que pertence.** Serve para modelar um sistema multi-organizacional.

- **Linhas a que está ligado.** Define a topologia da rede de processamento e indica, caso o nodo esteja ligado a mais do que uma linha, se o nodo vai ser sujeito a sobrecarga devido ao encaminhamento de mensagens para outros nodos.

- **Recursos.** Define a heterogeneidade do sistema ao nível do sistema de processamento. Cada recurso caracteriza-se pela sua identificação (única dentro de cada nodo), pela capacidade respectiva e pelo tipo de recurso. Quanto ao tipo, os recursos podem ser activos ou passivos. Os recursos activos são partilhados pelas tarefas em execução no nodo (*time-sharing*). É possivel definir prioridades no escalonamento reccorrendo à reserva de uma percentagem de um recurso para uma determinada tarefa. Quando não existem reservas, os recursos activos são equitativamente distribuídos pelas tarefas em execução variando a percentagem atribuída com o número de tarefas. Os recursos passivos também podem ser partilhados (como, por exemplo, um ficheiro de leitura), sendo essa situação caracterizada pelo valor -1 presente no atributo capacidade. A capacidade de um recurso passivo requirida por uma determinada tarefa, é-lhe alocada durante o tempo total da sua execução.

- **Padrão de carga.** Serve para modelar o padrão de carga a que o nodo será sujeito. Permite a definição de um padrão variável ao longo do tempo, sendo cada elemento da lista constituído por um intervalo de tempo (instante inicial mais instante final) e pela identificação do padrão respectivo (ver definição de padrões de carga na secção 5.3.4). Esta possibilidade é importante para modelar a realidade dos sistemas distribuídos, caracterizados por períodos com carga muito baixa (nocturnos) seguidos por períodos muito sobrecarregados (horas de expediente). A possibilidade de modelar diferentes padrões para cada nodo é necessária, dado, por exemplo, o desfazamento que existe entre os picos de carga em sistemas afastados geográficamente ou entre nodos de um mesmo sistema. Define a heterogeneidade do sistema ao nível da carga.

- **Tempos de processamento de mensagens.** Indicam os tempos de CPU necessários para fazer, respectivamente, o *packing/unpacking* de mensagens originárias ou destinadas ao nodo e para fazer o encaminhamento de mensagens destinadas a outros nodos (relevante apenas quando o nodo está ligado a mais do que uma linha). Os valores indicados são factores que serão multiplicados pelo tamanho das mensagens, sendo o total obtido o tempo de processamento requirido para um processador com capacidade unitária (para facilitar a comparação entre os tempos dos diversos nodos). Estes parâmetros são específicos para cada nodo, de modo a reforçar a possibilidade de definição de sistemas heterogéneos.

5.3.3 Linhas de Comunicação

As linhas de comunicação são caracterizadas pelos seguintes parâmetros:

- **Identificação.** Caracteriza univocamente cada linha de comunicação.

- **Nodo responsável.** Define qual o nodo que controla o acesso e taxação na linha. Implicitamente também define qual a organização sua proprietária.

- **Capacidade.** Indica qual a velocidade da linha expressa em bytes por 1000 unidades de simulação. Se considerarmos que cada unidade de simulação corresponde a um milisegundo, então teremos bytes por segundo.

Convêm apresentar qual o processo de encaminhamento de mensagens utilizado no simulador. No início da simulação cada nodo e cada linha do sistema calcula as suas tabelas de encaminhamento. Estas tabelas indicam qual a linha (no caso dos nodos) ou qual o nodo (no caso das linhas) que uma mensagem deve seguir para atingir um outro nodo do sistema. Presentemente, essas tabelas são construídas tendo por base o caminho mais rápido (teoricamente) entre a entidade que constroi a tabela e os restantes nodos do sistema. Este processo não garante o funcionamento mais eficiente em todos os casos e deverá ser alvo de futuros melhoramentos.

O predicado disponibilizado para o envio de mensagens permite o *multi-cast*, ou seja, permite definir o destino da mensagem como um qualquer subconjunto de todas entidades existentes no sistema. Se apenas indicarmos uma entidade teremos comunicação ponto a ponto e se indicarmos todas as entidades teremos comunicação por difusão. Quando uma linha, ou nodo, tem de entregar uma mensagem as entidades destino são agrupadas de acordo com a próxima entidade por onde deverão ser encaminhadas. A mensagem original é depois partida em tantas mensagens quantas as próximas entidades determinadas, sendo essas mensagens encaminhadas para essas entidades, onde recomeça este processo. Com este procedimento garante-se que será gerado o número mínimo de mensagens possível para entregar a mensagem aos destinos.

5.3.4 Carga do Sistema

A definição da carga do sistema divide-se em duas partes: a definição de padrões de carga e a definição das tarefas que são utilizadas nesses padrões. Um padrão de carga caracteriza-se pelos seguintes parâmetros:

- **Identificação.** Caracteriza univocamente cada padrão de carga.
- **Tempo entre chegadas de tarefas.** Indica qual a distribuição teórica que modela o tempo médio de intervalo entre chegadas de tarefas a um nodo com este padrão de carga. Quando se pretender que o tempo entre chegadas seja determinístico utiliza-se o predicado `exactamente(`*`Valor`*`)`.
- **Histograma de tarefas.** Define, recorrendo a um histograma, que tipo de tarefas serão geradas. Para cada elemento da lista indica-se qual a percentagem de ocorrências desse tipo de tarefa.

Cada tipo de tarefa caracteriza-se pelos seguintes parâmetros:

- **Identificação.** Identifica univocamente cada tipo de tarefa.
- **Duração.** Indica qual a duração da tarefa (através de uma distribuição teórica) caso fosse executada num nodo de processamento com capacidades de recursos activos idênticas às apresentadas na lista de recursos da tarefa. O cálculo do tempo real de execução num qualquer nodo processa-se da seguinte forma: calcula-se qual o menor ratio entre os recursos activos atribuídos à tarefa (quer seja resultante da partilha, quer seja reservado) e os respectivos recursos activos presentes na lista de recursos da tarefa. Depois, esse factor é multiplicado pelo tempo de execução da tarefa, determinando-se assim o seu tempo real de execução. Este tempo é sistematicamente reavaliado conforme entram ou saiem tarefas da lista de execução do nodo.

- **Prazo de execução.** Indica qual o prazo que a tarefa tem para ser executada, contado a partir do instante de chegada da tarefa ao sistema. A não existência de qualquer prazo é indicada pelo parâmetro `nao_tem`. Caso exista prazo o agente de distribuição assume o conhecimento dos parâmetros de execução da tarefa.

- **Lista de recursos.** Indica quais os recursos que um nodo deve possuir para que a tarefa possa ser executada. As capacidades nos recursos activos não indicam a capacidade que deve ser alocada à tarefa, servindo apenas para determinar o tempo real de execução, tal como foi explicado anteriormente. A capacidade indicada nos recursos passivos deverá existir no nodo e será alocada à tarefa durante o tempo total da sua execução.

- **Necessidades de comunicação.** Define, respectivamente, o intervalo entre envios consecutivos de mensagens para o nodo originário da tarefa e o tamanho dessas mensagens. Quando é enviada uma mensagem deste tipo, a tarefa é retirada de execução, sendo escalonada de novo aquando da chegada de uma mensagem de resposta enviada pelo seu nodo originário. Os únicos atrasos induzidos no nodo originário consistem nos tempos de *unpacking* e *packing* da mensagem enviada pela tarefa e da mensagem de resposta, respectivamente.

- **Tamanho da tarefa.** Indica o tamanho da tarefa (código e dados) em bytes. É usado aquando das transferências de tarefas entre nodos para determinar o tamanho das mensagens.

5.4 Funcionamento do Simulador

O simulador desenvolvido segue uma filosofia *event-schedulling* [Rodrigues, 1988], ou seja, quando ocorre um evento, a entidade que o gerou deve proceder ao planeamento de todos os eventos futuros que são consequência desse evento. Nesta simulação o evento principal, que desencadeia todo o processo, é a chega de uma tarefa ao sistema. O planeamento das primeiras chegadas é feito no instante zero de simulação. Depois, durante o processo de tratamento de uma nova chegada, cada nodo planeia a próxima. Este processo de planeamentos futuros repete-se para todos os outros eventos.

Todas as entidades activas na simulação, ou seja, os nodos, as linhas de comunicação e os agentes de distribuição, foram desenvolvidas recorrendo a linguagem de padrões [Bratko, 1990].

Esta linguagem é ideal para descrever comportamentos reactivos, tal como os apresentados por estas entidades, que reagem ao surgimento de eventos a elas destinados. O interpretador de eventos apresentado em [Bratko, 1990] teve de ser ligeiramente alterado para permitir a convivência de padrões referentes a diferentes entidades (ver figura 5.5). Assim, a cada padrão foi acrescentado o identificador da entidade a que diz respeito, sendo essa entidade activada através do predicado `run(`***`Entidade`***`)`.

```
%
% Interpretador de Padroes Modificado
%

:- op(800, xfx, --->).
:- op(600, xfx, ::).

run(Identificador) :-
       Identificador :: Condition ---> Action,
       test(Condition), !,
       execute(Action, Identificador), !.

test([]).
test([Head|Tail]) :-
       call(Head),
       test(Tail).

execute([stop], _) :- !.
execute([], Identificador) :-
       run(Identificador).
execute([Head|Tail], Identificador) :-
       call(Head), !,
       execute(Tail, Identificador).
```

Figure 5.5: Interpretador de Padrões Modificado

O próprio *kernel* do simulador foi desenvolvido em linguagem de padrões (ver figura 5.6). Este *kernel* constitui o "motor" da simulação, estando encarregado da gestão dos eventos e do avanço do tempo. Mais concretamente, as suas funções são:

- **Inicializar a simulação.** As seguintes acções são realizadas no início da simulação: é inicializado o ambiente, é planeado o evento que origina o fim da simulação, planeado o primeiro evento de monitorização, e planeado o evento que marca o início de simulação em todas as entidades presentes. Cada entidade, após a detecção deste evento, inicializa o seu ambiente particular e, no caso dos nodos, planeia a primeira chegada de tarefas.

- **Monitorização.** Imprime o tempo actual de simulação e faz o *garbage collection* das pilhas do interpretador de PROLOG.

- **Despoletar eventos nos agentes.** Cada evento é caracterizado pela sua descrição, pelo instante em que vai ocorrer e pela lista de entidades que afecta. Quando se atinge o instante em que o evento vai ocorrer, é necessário activar todas as entidades que estão presentes na lista de entidades afectadas.

- **Retirar eventos que não afectam entidades.** Por questões de optimização, todos os eventos que já não vão afectar nenhuma entidade são retirados do ambiente.

- **Avanço do tempo.** Quando já não existe mais nenhum evento planeado para um dado instante o tempo de simulação é avançado para o tempo do próximo evento planeado. Como a filosofia seguida é a de *event-schedulling* garante-se que durante esse intervalo nada acontece.

- **Fim de simulação.** Quando o evento que marca o fim de simulação já foi despoletado em todas as entidades presentes, o próprio *kernel* termina, limpando o ambiente de todos os eventos planeados para depois desse instante.

Para garantir que a duração real da simulação cresça linearmente em relação ao tempo que se pretende simular é necessário garantir que, quer a base do interpretador de PROLOG, quer as suas pilhas de controlo, se mantenham com um tamanho sensivelmente constante ao longo da simulação. Quanto ao primeiro aspecto, optou-se por se armazar os resultados da simulação em ficheiro conforme vão sendo obtidos. Para garantir o segundo ponto há que eliminar a possibilidade de *backtracking* do interpretador. Esta é a razão da colocação dos "cuts" no interpretador de padrões da figura 5.5.

Por razões de espaço e de tempo não será apresentado o funcionamento detalhado de todas as entidades envolvidas na simulação. Apenas será apresentada a descrição de alguns predicados e mensagens essenciais ao desenvolvimento de agentes de distribuição, nomeadamente:

- `envia_mensagem((`*`Agente`*`, `*`Tipo`*`), `*`ListaDestino`*`, `*`Mensagem`*`, `*`Tam`*`).` Predicado usado para enviar uma mensagem a uma entidade remota. Os parâmetros indicam, respectivamente, a origem da mensagem, a lista de agentes destino (cada elemento é um tuplo formado pela identificação do agente e o seu tipo), a mensagem a enviar, e o tamanho da mesma. As mensagens quando chegam aos nodos de destino são escalonadas da mesma forma que os eventos normais.

- `distribuidor_presente(`*`Distribuidor`*`, `*`Tipo`*`).` Mensagem que um agente de distribuição de carga deve enviar a um nodo para lhe indicar que é encarregado pela distribuição das suas tarefas.

```
%
% Simulador
%

% Inicio simulacao
simulador ::
        [not(tempo(_))]
        --->
        [semente(X, Y, Z),
        setrand(rand(X, Y, Z)),
        assert(tempo(0)),
        assert(tarefa(0)),
        assert(tarefa_aux(-1)),
        todas_entidades(Todos),
        assert(evento(0, Todos, inicia_simulacao)),
        fim_simulacao(Fim),
        assert(evento(Fim, Todos, fim_simulacao)),
        imprime_tempo(Imp),
        assert(evento(Imp, simulador, imprime_tempo)),
        format('Inicio da simulacao.~n', [])].

% Monitorizacao
simulador ::
        [tempo(T),
        evento(T, simulador, imprime_tempo)]
        --->
        [retract(evento(T, simulador, imprime_tempo)),
        format('Tempo = ~w~n',[T]),
        imprime_tempo(Imp),
        NovoT is T + Imp,
        garbage_collect,
        assert(evento(NovoT, simulador, imprime_tempo))].

% Fim de simulacao
simulador ::
        [tempo(T),
        evento(T, [], fim_simulacao)]
        --->
        [retractall(evento(_, _, _)),
        retractall(tempo(_)),
        format('Fim da simulacao.~n', []),
        stop].

% Despoletar eventos nos agentes
simulador ::
        [tempo(T),
        evento(T, [(_, Tipo_agente)|_], _)]
        --->
        [run(Tipo_agente)].

% Limpeza de mensagens vazias
simulador ::
        [evento(T, [], Tipo)]
        --->
        [retract(evento(T, [], Tipo))].

% Avanco do tempo
simulador ::
        [tempo(T),
        not(evento(T, _, _))]
        --->
        [findall(X, evento(X, _, _), Tempos),
        min_list(Tempos, NovoT),
        replace(tempo(T), tempo(NovoT))].
```

Figure 5.6: *Kernel* do Simulador

Chapter 6

Conclusões e Trabalho Futuro

6.1 Conclusões

Neste estágio especificou-se uma nova estratégia de distribuição de carga, baseada em sistemas multi-agente, que permite resolver muitos dos problemas apresentados pelas abordagens convencionais quando aplicadas a um sistema distribuído multi-organizacional. Os principais problemas resolvidos são:

- a impossibilidade de comércio dos recursos de uma organização;
- a não garantia de intrusão;
- a não integração de tarefas com e sem especificação de recursos;
- a impossibilidade de considerar explicitamente os recursos existentes, nomeadamente as infra-estruturas de comunicação; e
- a impossibilidade de definir vários objectivos a atingir com a distribuição de carga.

Utilizando uma estruturação organizacional dinâmica, em que a gestão dos recursos da organização e das interações com outras organizações é dividida dinâmicamente entre os agentes e em que a localização dos próprios agentes pode mudar, conseguiu-se uma abordagem de distribuição flexível, adaptável às características particulares de cada organização e ao ambiente dinâmico que caracteriza o problema.

Para reduzir o tempo de negociação e viabilizar a transferência de pequenas tarefas entre duas organizações, especificou-se o conceito de contrato a longo prazo, orientado aos recursos, e apresentou-se o conceito de nodo virtual como forma de gerir, transparentemente e eficientemente, os recursos contratados desta forma. Os nodos virtuais também possibilitam às organizações um meio para expandirem o seu mercado. Outro conceito inovador, é definição dinâmica da extensão do mercado para a negociação de uma tarefa. Este conceito permite determinar transparentemente se uma tarefa deve ser executada localmente, dentro da organização, ou numa organização externa. Também constitui uma forma de combater a racionalidade limitada dos agentes através da dedicação da capacidade de processamento às tarefas mais influentes.

O comportamento interno dos agentes foi caracterizado em função de três parâmetros:

- $Idade$, que permite variar a taxa de actualização dos parâmetros da função de utilidade;
- ΔT, que permite variar o tamanho da memória do agente, dando-lhe uma atitude mais ou menos exploradora e uma maior ou menor capacidade de reação; e
- α, que define a atitude do agente perante o risco.

Nenhum dos parâmetros anteriores tem uma correlação evidente com a eficiência do sistema, mas permitem ao gestor definir as características do mesmo, podendo o mesmo reflectir a sua atitude o no comportamento dos agentes. Caso se permita aos agentes variar automáticamente estes parâmetros, teremos um novo meio conjuntamente com a estruturação organizacional, no sentido de dinamicamente aumentar a eficiência do sistema.

Para validar a abordagem apresentada foi desenvolvida uma plataforma de simulação para problemas de distribuição de carga. Esta plataforma permite especificar um sistema distribuído multi-organizacional com padrões dinâmicos de carga sendo genérica para qualquer estratégia de distribuição. Para além de permitir comparar a eficiência de diversas estratégias de distribuição, constitui uma ferramenta de apoio aos administradores de sistema, pois permite testar à priori os efeitos da introdução de uma nova estratégia. Neste caso em particular, é fundamental para o ajuste dos parâmetros que caracterizam o comportamento dos agentes.

6.2 Trabalho Futuro

O trabalho futuro a desenvolver recai sobre três áreas principais:

- validação da abordagem desenvolvida;
- aprofundamento do mecanismo de mudança da estruturação organizacional; e
- aprofundamento da especificação do comportamento dos agentes.

Validação

Para validar a abordagem desenvolvida é necessário implementar um protótipo dessa abordagem na plataforma de simulação desenvolvida. Para efeitos de comparação também teriam de ser implementadas mais algumas das estratégias convencionais apresentadas na secção 3. Uma comparação efectiva nunca seria conseguida dado que essas estratégias não estão preparadas para funcionar num ambiente multi-organizacional. Um dos aspectos interessantes a analisar na abordagem desenvolvida consiste na influência do comportamento no desempenho dos agentes: a relação entre os parâmetros $Idade$, ΔT e α e a qualidade do desempenho não é clara, sendo necessário proceder-se a um estudo de simulação aprofundado para quantificar essa relação. Um estudo semelhante foi efectuado em [Shaerf et al., 1995] com resultados supreendentes.

Dado que a abordagem concebida é genérica para problemas de alocação de recursos, deveria generalizar-se o simulador desenvolvido para testar qualquer problema de alocação e não apenas problemas de distribuição de carga. Para facilitar o processo de simulação também seria conveniente desenvolver uma ferramenta integrada, preferencialmente visual, que permita uma definição facilitada de configurações de sistemas e de consulta dos resultados. Outros aspectos importante são a obtenção de intervalos de confiança para os resultados de simulação e a determinação automática das durações óptimas da simulação e dos períodos de inicialização.

Estruturação Organizacional

Embora o protocolo de mudança da estruturação organizacional tenha sido especificado, falta definir como é que os agentes decidem quando e como alterar essa estruturação. Algumas das ideias a investigar são: mudanças após variações negativas do valor da função de utilidade em mais do que um determinado Δ deste a última tentativa de mudança; e decisão da nova estruturação baseada no que teria acontecido se se repetissem todos os eventos presentes na memória do agente.

Comportamento dos Agentes

Os aspectos principais a aprofundar quanto ao comportamento dos agentes são os conceitos de memória e esquecimento e a relação entre estes e a atitude do agente perante o risco. Neste momento utiliza-se um filtro que elimina todos os acontecimentos que ocorreram para além de um certo instante e, exceptuando o cálculo da esperança de variação na utilidade no relacionamento com um determinado agente, dá-se a mesma importância a todas as amostras de um acontecimento presentes na memória. É necessário investigar em que medida se pode abandonar o conceito de limite de memória, muito pouco realista, e utilizar a noção de esquecimento gradual para todos os dados, como forma de dar mais importância aos acontecimentos recentes no cálculo das previsões. Uma possibilidade para se atingir este objectivo consiste na utilização simultânea de médias exponenciais (médias parametrizadas com um valor que define o peso das amostras recentes) e do conceito de idade dos dados [Rodrigues, 1988]. Esta análise também tem como objectivo eliminar o parâmetro *Idade* do comportamento do agente, passando a taxa de actualização a ser definida em função da atitude do agente perante o risco e da qualidade actual da estimativa da utilidade.

Outros aspectos a investigar são:

- a sintaxe da linguagem comum para especificação dos parâmetros das mensagens;
- a possibilidade da utilização de publicidade para se alargar um mercado;
- a possibilidade de se formarem associações entre organizações; e
- as possibilidades de fraude e especulação no sistema e os meios de as combater.

Bibliography

[Becker, 1995] Becker, W. (1995). Dynamic balancing complex workload in workstation networks - challenge, concepts and experience. In Hertzberger, B. and Serazzi, G., editors, *High-Performance Computing and Networking*, Lecture Notes in Computer Science 919, pages 407–412. Springer.

[Bratko, 1990] Bratko, I. (1990). *PROLOG: Programming for Artificial Intelligence*. International Computer Science Series. Addison-Wesley, second edition.

[Carlsson and Widen, 1993] Carlsson, M. and Widen, J. (1993). *SICStus Prolog User's Manual*. Swedish Institute of Computer Science.

[Costa et al., 1995] Costa, J. F., Silva, A., and Delgado, J. (1995). Análise dos sistemas comerciais emergentes na internet. In *Conferência Nacional de WWW*.

[Durfee, 1992] Durfee, E. H. (1992). What your computer really needs to know, you learned in kindergarten. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 858–864.

[Durfee et al., 1987] Durfee, E. H., Lesser, V. R., and Corkill, D. D. (1987). Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36(11):1275–1291.

[Durfee et al., 1989] Durfee, E. H., Lesser, V. R., and Corkill, D. D. (1989). Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83.

[Eager et al., 1986] Eager, D. L., Lazowska, E. D., and Zahorjan, J. (1986). Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, SE-12(5):662–675.

[Fernandes, 1993] Fernandes, E. M. G. . (1993). *Estatística Aplicada.* Universidade do Minho.

[Garvey et al., 1994] Garvey, A., Decker, K., and Lesser, V. R. (1994). A negotiation-based interface between a real-time scheduler and a decision-maker. Technical Report 94-08, Department of Computer Science, University of Massachusetts.

[Geist et al., 1994] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V. (1994). *PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Network Parallel Computing.* The MIT Press.

[Genesereth and Ketchpel, 1994] Genesereth, M. R. and Ketchpel, S. P. (1994). Software agents. Technical Report WP032, Computer Science Department, Stanford University.

[Gries, 1981] Gries, D. (1981). *The Science of Programming.* Springer-Verlag.

[Jacqmot and Milgrom, 1993] Jacqmot, C. and Milgrom, E. (1993). A systematic approach to load distribution strategies for distributed systems. In Cosnard, M. and Puigjaner, R., editors, *Decentralized and Distributed Systems*, number A–39 in IFIP Transactions A: Computer Science and Technology, pages 291–303. Elsevier Science B.V. (North-Holland).

[Jennings and Wooldridge, 1996] Jennings, N. and Wooldridge, M. (1996). Software agents. *IEE Review*, pages 17–20.

[Jennings, 1996] Jennings, N. R. (1996). Coordination techniques for distributed artificial intelligence. In O'Hare, G. and Jennings, N., editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. Wiley Inter-Science.

[Kremien, 1995] Kremien, O. (1995). Scalability in distributed systems, parallel systems and supercomputers. In Hertzberger, B. and Serazzi, G., editors, *High-Performance Computing and Networking*, Lecture Notes in Computer Science 919, pages 532–541. Springer.

[Kremien and Kramer, 1992] Kremien, O. and Kramer, J. (1992). Methodical analysis of adaptive load sharing algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 3(6):747–760.

[Kremien et al., 1993] Kremien, O., Kramer, J., and Magee, J. (1993). Scalable load-sharing for distributed systems. In *HICSS-26*.

[Kunz, 1991] Kunz, T. (1991). The influence of different workload descriptions on a heuristic load balancing scheme. *IEEE Transactions on Software Engineering*, 17(7):725–730.

[Lesser, 1990] Lesser, V. R. (1990). An overview of dai: Viewing distributed ai as distributed search. *Journal of Japonese Society for Artificial Intelligence - Special Issue on Distributed Artificial Intelligence*, 5(4):392–400.

[Lesser, 1995] Lesser, V. R. (1995). Multiagent systems: An emerging subdiscipline of ai. *ACM Computing Surveys*, 27(3):340–342.

[Lüling et al., 1992] Lüling, R., Monien, B., and Ramme, F. (1992). A study on dynamic load balancing algorithms. Technical Report TR-001-92, Paderborn Center for Parallel Computing.

[Pindyck and Rubinfeld, 1992] Pindyck, R. S. and Rubinfeld, D. L. (1992). *Microeconomics*. Maxwell Macmillan International Editions, second edition.

[Ramamritham et al., 1989] Ramamritham, K., Stankovic, J. A., and Zhao, W. (1989). Distributed scheduling of tasks with deadlines and resource requirements. *IEEE Transactions on Computers*, 38(8):1110–1123.

[Riedl and Richter, 1996] Riedl, R. and Richter, L. (1996). Classification of load distribution algorithms. In *Fourth Euromicro Workshop on Parallel and Distributed Processing (PDP'96)*, pages 404–413. IEEE Computer Society Press.

[Rodrigues, 1988] Rodrigues, A. G. (1988). *Simulação*. Universidade do Minho.

[Sandholm, 1993] Sandholm, T. (1993). An implementation of the contract net protocol based on marginal cost calculations. In *Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, Washington D.C.

[Sandholm and Lesser, 1995a] Sandholm, T. and Lesser, V. R. (1995a). Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *First International Conference on Multiagent Systems (ICMAS - 95)*, pages 328–335, San Francisco.

[Sandholm and Lesser, 1995b] Sandholm, T. and Lesser, V. R. (1995b). On automated contracting in multi-enterprise manufacturing. In *Improving Manufacturing Performance in a Distributed Enterprise: Advanced Systems and Tools*, pages 33–42, Edimburgh, Scotland.

[Shaerf et al., 1995] Shaerf, A., Shoham, Y., and Tennenholtz, M. (1995). Adaptive load balancing: A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2:475–500.

[Shivaratri et al., 1992] Shivaratri, N. G., Krueger, P., and Singhal, M. (1992). Load distributing for locally distributed systems. *IEEE Computer*, pages 33–44.

[Simão, 1995] Simão, J. P. F. (1995). Segurança no www - panorâmica do estado da arte. In *Conferência Nacional de WWW*.

[Smith, 1980] Smith, R. G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113.

[Theimer and Lantz, 1989] Theimer, M. M. and Lantz, K. A. (1989). Finding idle machines in a workstation-based distributed system. *IEEE Transactions on Software Engineering*, 15(11):1444–1458.

[Wellman, 1995] Wellman, M. P. (1995). The economic approach to artificial intelligence. *ACM Computing Surveys*, 27(3):340–342.

[Özden et al., 1993] Özden, B., Goldberg, A. J., and Silberschatz, A. (1993). Scalable and non-intrusive load sharing in owner-based distributed systems. In *Proceedings of the Fifth IEEE Symposium on Parallel and Distributed Processing*, pages 690–699. IEEE Computer Society Press.

[Zhou, 1988] Zhou, S. (1988). A trace-driven simulation study of dynamic load balancing. *IEEE Transactions on Software Engineering*, 14(9):1327–1341.