A MULTI-AGENT BASED APPROACH FOR LOAD DISTRIBUTION IN MULTI-ENTERPRISE ENVIRONMENTS

Alcino Cunha
alcino@di-ia.uminho.pt

Orlando Belo
obelo@di.uminho.pt

Departamento de Informática, Escola de Engenharia, Universidade do Minho
Largo do Paço, 4709 Braga Codex, PORTUGAL

KEYWORDS: Multi-Agent Systems, Load Distribution, Negotiation, Dynamic Organizational Structures.

ABSTRACT

This paper presents and discusses an automated contracting mechanism for load distribution in multi-enterprise environments. Within each enterprise, tasks are allocated by a multi-agent system whose agents develop cooperative processes in order to determine the best nodes for execution. When profitable, a negotiation process is started as a way to select a remote enterprise to execute a specific task. The use of a dynamic organizational structure, the concepts of resource oriented contract and virtual node, the dynamic definition of the task market extent and the possibility to precisely define the internal behavior of the agents, provide the developed mechanism with the flexibility to adapt to the specificities and dynamic characteristics of almost every enterprise and organizations environments.

1 INTRODUCTION

Load distribution is usually seen in a single enterprise context. However, it is possible to increase its performance by taking advantage of economies of scale emerging from short term alliances between different enterprises. Each enterprise has a set of tasks and a set of resources that it can use to handle its own activities. A task can be transferred from one enterprise to another if one knows that it cannot be executed in its current location or if it can be more efficiently handled in another enterprise environment. In order to agree upon the monetary compensation that must be payed for the execution of the transfered task, a mechanism for automated contracting is needed.

Several mechanisms have been designed for load distribution inside an organization (e.g. [3, 14, 8, 12, 5]). It is not simple to apply those mechanisms on a multi-enterprise environment due to the following difficulties:

- none of them deals with the monetary aspects needed to calculate compensations;
- the selection of the destination for a task transference is not agreed upon by mutual selection;
- none of them considers the communication costs in its decisions; and,
- it is not possible to explicitly define more than one goal to be achieved by load distribution.

In order to deal with such difficulties, a new load distribution mechanism was developed based on a multi-agent approach. Software agents were used because they have some of the essential characteristics that one finds in a multi-enterprise environment, namely [4]: autonomy, social ability, responsiveness and pro-activeness.

Agents involved in automated contracting in a multi-enterprise environment must behave in a self-interested fashion and make decisions in real-time. This decision making process is constrained to the bounded rationality of the agents, which exists due to [6]: the limited communication bandwidth; the limited processing power; the self-interested behavior of the agents which restricts the sharing of information; and finally, the dynamic characteristics of the environment. Mechanisms for automated contracting between bounded rational self-interested agents have already been proposed in [9, 10]. What makes our work innovative is the use of a multi-agent approach, applied not only between different enterprises, but also within each single enterprise. The reasons for not having a single agent for each enterprise are related to the ones usually presented in favor of multi-agent systems technology, namely: the spatial distribution of the resources, the increase in efficiency through parallelization, and the increase in reactivity, fault tolerance, and extensibility.

2 THE MULTI-AGENT ARCHITECTURE

Within an enterprise one may find two kinds of agents: *allocation agents*, that control the enterprise resources, allocate tasks and develop the negotiation process with other enterprises; and *routing agents*, that are responsible for the routing of messages between allocation agents and from an outside enterprise to the specific allocation agent in charge of the negotiation with it. Each processing node contains one routing agent.

2.1 Dynamic Organizational Structure

The responsibilities of the allocation agents are determined by a dynamic organizational structure, which specifies their roles, positions and communication patterns. Such a structure provides the means to reduce the complexity of the control problem and constrains the decisions range, giving the basis for a coherent cooperation [1]. An allocation agent is characterized by a *3-tuple* $< Location, Resources, Enterprises >$, where *Location* defines the node where the agent is located, *Resources* defines the set of nodes and communication lines controlled by the agent, and *Enterprises* defines the set of enterprises which witch the agent interacts. In order to increase efficiency it was defined that two agents cannot have the same location and that one resource or organization can be controlled by more than one agent.

To avoid bottlenecks and contention situations, the decision to change the organizational structure is distributed among the agents. This distribution may cause some conflicts due to the communication latencies. A specific protocol was designed to rule the changes in the organizational structure and to deal with conflict detection and consequent resolution. This protocol is presented in Figure 1 in an informal notation.

```
APPEARING MESSAGE
    message id
    sending agent id
    location node
    controlled nodes
    controlled communication lines
    controlled organizations

DISAPPEARING MESSAGE
    message id
    sending agent id
    deadline

CONFLICT RESOLUTION MESSAGE
    message id
    sending agent id
    receiving agent id
    conflict 1
        conflicting item
        expected marginal utility
    (...)
    conflict n
```

Figure 1: Changing Protocol

Every change in the organizational structure implies two kinds of messages: a disappearing and an appearing one. These messages are broadcasted by each allocation agent that wants to change its competences. A deadline field specifies the maximum time that the agent requires to make the change. If it does not reappear before this deadline, the remaining agents are free to occupy its own competences. When a conflict occurs, the involved agents should exchange conflict resolution messages, where, for each conflicting item, each one of them specifies the increase in the utility value that it expects to occur if it keeps the conflicting item. The agent with the highest expected marginal utility is the winner of the conflict.

The decision to change the organizational structure is made every time the enterprise satisfaction decreases more than a parameterized value, or after a certain period is passed since the last decision. This process is very costly, both in communication and computation, and should be tuned so that reorganization is undertaken infrequently [2].

2.2 The Utility Function

The measure of the satisfaction level of an enterprise, and the explicit definition of its goals, are achieved with the utility function

$$U(M, P_1, \ldots, P_{Np}),$$

where M stands for the available money and $P_1, \ldots, P_{Np}$ represent the parameters which influence the enterprise satisfaction level. M must always exist since agents have the need to quantify the money that the enterprise is willing to tradeoff for the execution of a specific task. All the internal decisions of the agents are based on marginal utility calculations.

To compute precisely the enterprise level of satisfaction, the allocation agents must broadcast their local values of the utility function parameters. In order to control the actualization rate, a special parameter Max_Age was introduced. This parameter specifies the maximum possible age for each agent's view of the utility value. The age of a parameter is defined as

$$Age_i(P) = t_{act} - t_{obt}(P),$$

where i stands for the agent that is assessing the age, t_{act} for the actual time, and $t_{obt}(P)$ for the last time when P was obtained by the agent i. Considering N as the number of allocation agents, M^k and P_j^k as the values of, respectively, the available money and the value of the parameter j in the allocation agent k, it should be true that

$$\forall i \quad \frac{\sum_{k=1}^{N}(Age_i(M^k) + \sum_{j=1}^{Np} Age_i(P_j^k))}{(Np+1) \times N} \leq Max\_Age.$$

By changing the value of Max_Age we achieve a trade-off between the actuality of the data and the saturation of the interconnection network.

3 THE AUTOMATED CONTRACTING MECHANISM

3.1 Negotiation Protocols

The main problem with the followed approach is the need for the agents to assume, simultaneously, two types of behavior: *self-interested*, when dealing with agents from other enterprises, and *cooperative*, within the same enterprise [11]. To solve this problem two communication protocols are simultaneously used. Inside the enterprise, the agents cooperate trough a slight variation of the original Contract Net protocol (CNP) [13]. For negotiations between different enterprises the CNP extension proposed in [10] is used. The original CNP can not be used in this context because it did not deal explicitly with the self-interestness of the agents. The major innovation of this extension is the use of multiple levels of commitment with penalties associated to potential commitment breaking.

The protocol proposed in [10] was further extended to allow the negotiation of long-term contracts. This contracts are resource oriented and the monetary compensation is negotiated per time unit. Some of their advantages are: 1) to allow task transference between two enterprises without a previous negotiation process; 2) the possibility to contract communication lines; and, 3) to provide a more effective way for enterprises with resource surplus to announce their availability.

In order to provide an efficient control of the resources contracted in a long-term way, the concept of virtual node was introduced. Every time a set of resources is contracted, a virtual node is created in the real node of the contracted enterprise. The capacity of this node is equal to the contracted capacity. Normal routing agents are installed in virtual nodes in order to allow their transparent control and inclusion in the $Resources$ set of the contractor agent as normal nodes. To simplify the access to a virtual node, the enterprise can also contract some communication lines (also in a long-term fashion) in order to link it to the real node.

Besides control transparency, the virtual node concept allows the enterprise to fiscally extend the market where it operates. This can be achieved in two ways: the organizational structure can change so that an allocation node is moved to the virtual node, providing a cheap access to more enterprises; or, new enterprises can contact the virtual node owner because of cheaper access to one of its nodes.

3.2 Dynamic Task Market Extent

Every time a task arrives to an allocation agent, it must decide if it will be immediately executed locally or if the task will announce a contract for the remote execution of the task. In the later case, it must also decide to what agents the announce should be sent. Both decisions are transparently made due to the concept of dynamic definition of the task market extent. This concept was retrieved from the microeconomics concept of product market extent [7]. The group of agents that constitute the market is derived from the task influence in the global level of satisfaction, the expected increases of satisfaction if the task were remotely executed, and the communication costs of sending the announce to the selected agents.

All marginal cost calculations are biased with the parameter α that represents the agents risk attitude. A low value for α leads to a risk seeking agent, that, for example, will send announces for other agents in expectation of a better offer. High α values will cause the agent to act more carefully. All estimations take into account the past events stored in the agent memory. Another behavior parameter is ΔM, the size of the agent's memory. Lower values for ΔM augment the agent reactivity and give him more exploratory characteristics. Higher values provide more secure and informed decisions. Both parameters can be used by the system manager to tune the system behavior according to his own preferences.

4 CONCLUSIONS

The automated contracting mechanism proposed in this paper is not specific for load distribution, but can be applied effectively to any resource allocation problem in a multi-enterprise context. It can be used in industry to allocate tasks in a distributed cell manufacturing environment and, occasionally, out-source some of them. The use of a dynamic organizational structure, the concept of resource oriented contract and virtual node, the dynamic definition of the task market extent and the possibility to define the internal behavior of the agents, provide this mechanism with the flexibility to adapt to the specificities and dynamic character of every enterprise and environment. Such flexibility is specially attractive because almost every computational problem can be modeled as a resource allocation problem [15].

REFERENCES

[1] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36(11):1275–1291, November 1987.

[2] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83, March 1989.

[3] Derek L. Eager, Edward D. Lazowska, and John Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, SE-12(5):662–675, May 1986.

[4] Nick Jennings and Michael Wooldridge. Software agents. *IEE Review*, pages 17–20, January 1996.

[5] O. Kremien, J. Kramer, and J. Magee. Scalable load-sharing for distributed systems. In *HICSS-26*, January 1993.

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

[7] Robert S. Pindyck and Daniel L. Rubinfeld. *Microeconomics.* Maxwell Macmillan International Editions, second edition, 1992.

[8] Krithi Ramamritham, John A. Stankovic, and Wei Zhao. Distributed scheduling of tasks with deadlines and resource requirements. *IEEE Transactions on Computers*, 38(8):1110–1123, August 1989.

[9] Tuomas Sandholm. 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., 1993.

[10] Tuomas Sandholm and Victor R. Lesser. 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, 1995.

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

[12] Niranjan G. Shivaratri, Phillip Krueger, and Mukesh Singhal. Load distributing for locally distributed systems. *IEEE Computer*, pages 33–44, December 1992.

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

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

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