Do the middle letters of "OLAP" stand for Linear Algebra ("LA")?

J.N. Oliveira
(joint work with H. Macedo)

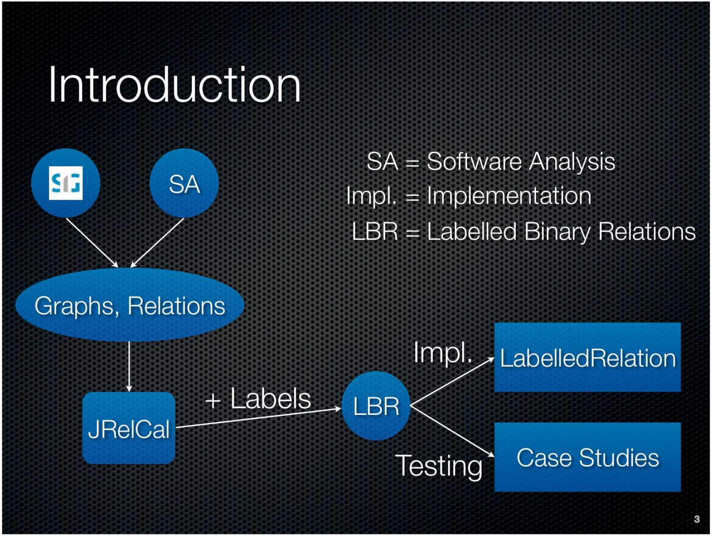
HASLab/Universidade do Minho
Braga, Portugal

SIG
Amsterdam, NL
26th May 2011

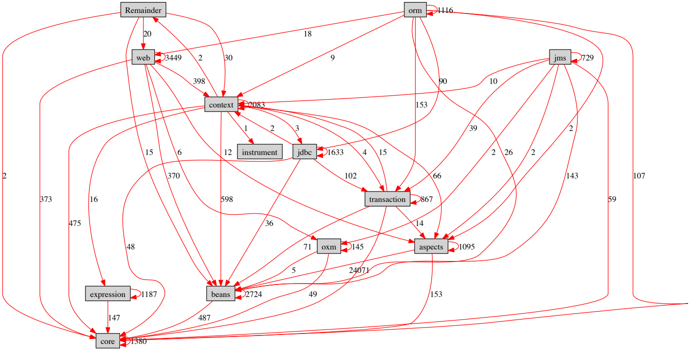
Context

- **HASLab** is a research group at Minho University in Braga, Portugal
- The group has been concerned with developing techniques for *high assurance software*
- **HASLab** - **SIG** collaboration on a regular basis since 2007
- **SIG** contributes with knowledge transfer in the area of **software quality**
- **HASLab** does so in the area of formal analysis, modelling and verification.

A previous collaborative project



On-going collaborative project



Mining call-graphs for software **architecture** quality profiling.

Motivation

Clearly:

- Need to **quantify** over relationships
- Raw information too fine-grained
- Too much information involved in data mining
- Need to make sense of **huge** data banks

Need for data summarizing techniques.

Motivation

In fact:

- Data summaries unveil **trends** hidden in raw data
- One sees the "**big picture**"

State-of-the -art:

- "OLAP" stands for *On Line Analytical Processing*
- Proprietary solutions (IBM, Oracle, MS)
- Calls for parallelism
- Expensive.

Can **parallel OLAP** be made more widely accessible?

OLAP's "Hello World"

As generated in MS Excel (choose Data > PivotTableReport):

Raw data

Model	**Year**	**Color**	**Sales**
Chevy	1990	Red	5
Chevy	1990	Blue	87
Ford	1990	Green	64
Ford	1990	Blue	99
Ford	1991	Red	8
Ford	1991	Blue	7

"How many vehicles were sold per color and model?"

CTAB

Sum of Sales	Model		
Color	Chevy	Ford	Grand Total
Blue	87	106	193
Green		64	64
Red	5	8	13
Grand Total	92	178	270

Pivot table

Three dimensions — *Model*, *Year*, *Color* — and one measure — *Sales*. Summarizing over *Year*.

OLAP Cubes

For huge raw data sets

- such **cross tabulation** summaries (vulg. "**pivot tables**") take too long to generate
- The same tabulation likely to be required by different people in the organization
- Solution: generate all possible summaries overnight so that businessmen can have all pivot tables afresh the day after.
- Build a "**cube**" with all such multi-dimension projections.

```
Chevy 1990 Blue 87
Chevy 1990 Red 5
Ford 1990 Blue 99
Ford 1990 Green 64
Ford 1991 Blue 7
Ford 1991 Red 8
------------------
Chevy 1990 ALL 92
Ford 1990 ALL 163
Ford 1991 ALL 15
Chevy ALL Blue 87
Chevy ALL Red 5
Ford ALL Blue 106
Ford ALL Green 64
Ford ALL Red 8
ALL 1990 Blue 186
ALL 1990 Green 64
ALL 1990 Red 5
ALL 1991 Blue 7
ALL 1991 Red 8
------------------
Chevy ALL ALL 92
Ford ALL ALL 178
ALL 1990 ALL 255
ALL 1991 ALL 15
ALL ALL Blue 193
ALL ALL Green 64
ALL ALL Red 13
------------------
ALL ALL ALL 270
```

OLAP —- which theory behind?

OLAP:

- Cross tabulations are matrices (2-dim)
- What about OLAP cubes?
- Why SQL, GROUPBY, and so on?

Parallel solutions:

- MS Excel spreadsheet users may legitimately ask:

  Is the generation of pivot tables in Excel actually taking advantage of the underlying multi-core hardware?
  How parallel is such a construction?

Inspiration: relational algebra

Projecting over some attributes is a standard operation in relation algebra:

Table

A	B	C	D
a1	b1	c1	d1
a2	b1	c2	d2
a3	b1	c1	d3

"Only interested in B and C"

$\pi_{B,C}$

B	C
b1	c1
b1	c2

Projected table

Relational projection

Given T, a set of tuples:

$$\pi_{B,C}\, T \quad = \quad \{(t[B], t[C]) \mid t \in T\}$$

Pointwise relational:

$$b(\pi_{B,C}\, T)c \quad \Leftrightarrow \quad \langle \exists\ t\ :\ t \in T :\ b = t[B] \wedge c = t[C] \rangle$$

Pointfree relational:

$$\pi_{B,C}\, T \quad = \quad f_B \cdot [\![T]\!] \cdot f_C^{\circ}$$

where $f_X\, t = t[X]$ and $[\![T]\!] = \{(t, t) \mid t \in T\}$ — a coreflexive binary relation.

Types ("Relations as arrows")

For $T \in set\ A$, a set of tuples:

$$\begin{tikzcd} A \arrow[d, "f_B"'] & A \arrow[l, "[\![T]\!]"'] \arrow[d, "f_C"] \\ B & C \arrow[l, "\pi_{B,C}\,T"] \end{tikzcd}$$

In general: R **any** binary relation and f, g **arbitrary** functions in

$$\pi_{g,f} R \;=\; g \cdot R \cdot f^{\circ} \qquad \begin{tikzcd} A \arrow[d, "g"'] & B \arrow[l, "R"'] \arrow[d, "f"] \\ C & D \arrow[l, "\pi_{g,f} R"] \end{tikzcd} \tag{1}$$

Question

How to project data without loosing **quantitative** information in measure columns such as eg. *Sales* in

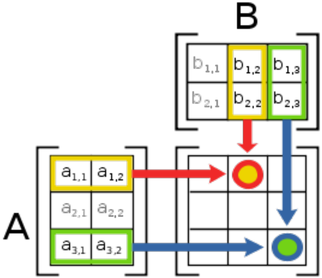
Model	**Year**	**Color**	**Sales**
Chevy	1990	Red	5
Chevy	1990	Blue	87
Ford	1990	Green	64
Ford	1990	Blue	99
Ford	1991	Red	8
Ford	1991	Blue	7

Clearly:

- Relational projection needs to take quantities into account
- Weighted graphs?
- Call them a proper name: we need **matrices**!

MMM — matrix matrix multiplication

From the Wikipedia:

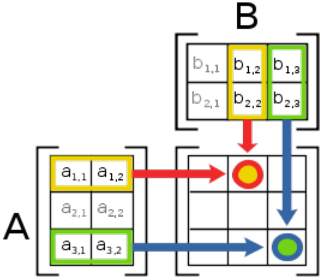


Index-wise definition

$$C_{ij} = \sum_{k,j=1,1}^{2,3} A_{ik} \times B_{kj}$$

MMM — matrix matrix multiplication

From the Wikipedia:



Index-wise definition

$$C_{ij} = \sum_{k,j=1,1}^{2,3} A_{ik} \times B_{kj}$$

Hiding indices i, j, k:

$$\begin{tikzcd} 3 & 2 \arrow[l, "A"'] & 3 \arrow[l, "B"'] \arrow[ll, bend left, "A \cdot B"] \end{tikzcd}$$

Index-free

$$C = A \cdot B$$

"Matrices as Arrows"

Given

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}_{m \times n} \qquad m \xleftarrow{A} n$$

$$B = \begin{bmatrix} b_{11} & \dots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nk} \end{bmatrix}_{n \times k} \qquad n \xleftarrow{B} k$$

define

$$\begin{tikzcd} m & n \arrow[l, "A"'] & k \arrow[l, "B"'] \arrow[ll, bend left, "A \cdot B"] \end{tikzcd}$$

Projection (relations) → tabulation (matrices)

Types:

$$\begin{tikzcd} n \arrow[d, "t_B"'] & n \arrow[l, "[\![T]\!]_M"'] \arrow[d, "t_C"] \\ |B| & |C| \arrow[l, "ctab_{B,C;M}\,T"] \end{tikzcd}$$

where

- n — number of rows in raw data collection T
- $|B|$ — set of values in column B of T
- $|C|$ — set of values in column C of T
- $[\![T]\!]_M$ — diagonal matrix storing all measures in column M of T
- t_X — "Membership matrix" of (non-metric) column X.

Details

Each column A in T "is" a function which tells, for each row, which value of $|A|$ can be found in such column, which *matricizes* into:

$$t_A : |A| \leftarrow n$$
$$a\ t_A\ r = \begin{cases} 1 & \text{if } T(r, A) = a \\ 0 & \text{otherwise} \end{cases}$$

Diagonal construction for measure column M:

$$[\![T]\!]_M : n \leftarrow n$$
$$j[\![T]\!]_M i = \begin{cases} T(j, M) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

"Hello World" illustration

Recall raw data example:

Model	**Year**	**Color**	**Sales**
Chevy	1990	Red	5
Chevy	1990	Blue	87
Ford	1990	Green	64
Ford	1990	Blue	99
Ford	1991	Red	8
Ford	1991	Blue	7

"Hello World" illustration

$$|Model| \xleftarrow{t_{Model}} 6$$

$$t_{Model} = \begin{array}{r|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline Chevy & 1 & 1 & 0 & 0 & 0 & 0 \\ Ford & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

$$|Color| \xleftarrow{t_{Color}} 6$$

$$t_{Color} = \begin{array}{r|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline Blue & 0 & 1 & 0 & 1 & 0 & 1 \\ Green & 0 & 0 & 1 & 0 & 0 & 0 \\ Red & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

Counting

Typewise, the composition of matrices $|Color| \xleftarrow{t_{Color}} 6$ and $6 \xleftarrow{t^{\circ}_{Model}} |Model|$ makes sense and yields

$$t_{Color} \cdot t^{\circ}_{Model} = \begin{array}{r|cc} & Chevy & Ford \\ \hline Blue & 1 & 2 \\ Green & 0 & 1 \\ Red & 1 & 1 \end{array} \qquad (2)$$

Matrix $t_{Model} \cdot t^{\circ}_{Color}$ (counting)

counting sale records — corresponds to formula $t_A \cdot [\![T]\!] \cdot t^{\circ}_B$ where the middle matrix is the identity.

Pivot Table Calculation

The outcome of cross-tabulation

$$ctab_{Color,Model;Sales} \quad = \quad t_{Color} \cdot [\![\, T \,]\!]_{Sales} \cdot t^{\circ}_{Model} \tag{3}$$

solely using matrix operations is the desired pivot table:

$$t_{Color} \cdot [\![\, T \,]\!]_{Sales} \cdot t^{\circ}_{Model} \quad = \quad \begin{array}{r|cc} & \mathit{Chevy} & \mathit{Ford} \\ \hline \mathit{Blue} & 87 & 106 \\ \mathit{Green} & 0 & 64 \\ \mathit{Red} & 5 & 8 \end{array} \tag{4}$$

Grand Totals (ALL) still missing

- Easily obtained via *"bang" matrices* ($!_A$)
- Matrix counterpart of the "bang" function (unique function to singleton type), that is, matrix $1 \xleftarrow{!_A} |A|$ wholly filled up with 1s.

Using matrix **block notation** cross tabulation (with totals) becomes:

$$ctab_{A,B;M} : |A| + 1 \leftarrow |B| + 1$$

$$ctab_{A,B;M} = \left[\frac{t_A}{!}\right] \cdot [\![T]\!]_M \cdot \left[\frac{t_B}{!}\right]^{\circ} \qquad (5)$$

"Hello World" illustration

$$\left[\frac{t_{Color}}{!}\right] \cdot [\![T]\!]_{Sales} \cdot \left[\frac{t_{Model}}{!}\right]^{\circ} = \begin{array}{r|ccc} & \mathit{Chevy} & \mathit{Ford} & \text{ALL} \\ \hline \mathit{Blue} & 87 & 106 & 193 \\ \mathit{Green} & 0 & 64 & 64 \\ \mathit{Red} & 5 & 8 & 13 \\ \text{ALL} & 92 & 178 & 270 \end{array}$$

Sum of Sales	Model		
Color	Chevy	Ford	Grand Total
Blue	87	106	193
Green		64	64
Red	5	8	13
Grand Total	92	178	270

Incremental OLAP (proving things)

- Let T be yesterday's raw data and T' be the today's data.
- Assume that T has remained the same (no updates, no deletes).
- Let $T'' = T; T'$ denote the two data sources appended. Then the following facts hold:

$$t''_A = [t_A | t'_A] \tag{6}$$
$$t''_B = [t_B | t'_B] \tag{7}$$
$$[\![T; T']\!]_M = [\![T]\!]_M \oplus [\![T']\!]_M \tag{8}$$

where $\oplus$ denotes the direct sum of two matrices.

Let us prove that cross tabulation is incremental:

$$ctab_{A,B;M}(T; T') = ctab_{A,B;M}\, T + ctab_{A,B;M}\, T' \tag{9}$$

Calculational proof

$$
\begin{array}{ll}
& ctab_{A,B;M}(T;T') \\
\Leftrightarrow & \quad \{ \ \text{(5); totals off with no loss of generality} \ \} \\
& t''_A \cdot [\![T;T']\!]_M \cdot (t''_B)^\circ \\
\Leftrightarrow & \quad \{ \ \text{(6) ; (7) and (8)} \ \} \\
& [t_A | t'_A] \cdot ([\![T]\!]_M \oplus [\![T']\!]_M) \cdot [t_B | t'_B]^\circ \\
\Leftrightarrow & \quad \{ \ \text{absorption} \ \} \\
& [t_A \cdot [\![T]\!]_M | t'_A \cdot [\![T']\!]_M] \cdot \left[\dfrac{t_B^\circ}{(t'_B)^\circ} \right] \\
\Leftrightarrow & \quad \{ \ \text{divide \& conquer matrix multiplication} \ \} \\
& t_A \cdot [\![T]\!]_M \cdot t_B^\circ + t'_A \cdot [\![T']\!]_M \cdot (t'_B)^\circ \\
\Leftrightarrow & \quad \{ \ \text{(5) twice} \ \} \\
& ctab_{A,B;M} T + ctab_{A,B;M} T'
\end{array}
$$

OLAP Cube (parallel) construction

Thus far

Summary generation in "human readable" format: cross-tabulations are 2D charts.

Higher dimensions

Generation of cubic and hypercubic data summaries also captured by our typed LA approach.

We have to introduce some notion of dimension **product**.

Khatri-Rao matrix product

Given matrices $n \xleftarrow{A} m$ and $p \xleftarrow{B} m$, build the Khatri-Rao product of A and B,

$$
\begin{tikzcd}
n \times p \\
m \arrow[u, "A \odot B"']
\end{tikzcd}
\qquad
\begin{tikzcd}
n & n \times p \arrow[l, "fst"'] \arrow[r, "snd"] & p \\
& m \arrow[ul, "A"] \arrow[u, "A \odot B"'] \arrow[ur, "B"'] &
\end{tikzcd}
$$

as follows,

$$
\begin{aligned}
u \odot v &= u \otimes v \\
[A_1|A_2] \odot [B_1|B_2] &= [A_1 \odot B_1 | A_2 \odot B_2]
\end{aligned}
\tag{10}
$$

where u, v are column-vectors and A_i, B_i are suitably typed matrices.

Example: measures in Khatri-Rao

As an example of Khatri-Rao product operation, consider row vector

$$s = \begin{bmatrix} 5 & 87 & 64 & 99 & 8 & 7 \end{bmatrix}$$

of type $1 \xleftarrow{s} 6$ capturing the transposition of the *Sales* column. Then Khatri-Rao product $s \odot id$ is the corresponding diagonal matrix:

$$6 \xleftarrow{s \odot id} 6 = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 87 & 0 & 0 & 0 & 0 \\ 0 & 0 & 64 & 0 & 0 & 0 \\ 0 & 0 & 0 & 99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 \end{bmatrix} \tag{11}$$

"Hello World" illustration

Back to our running example, recall projections

$$t_{Model} : |Model| \longleftarrow 6$$

$$t_{Model} = \begin{array}{r|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline Chevy & 1 & 1 & 0 & 0 & 0 & 0 \\ Ford & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

and

$$t_{Color} : |Color| \longleftarrow 6$$

$$t_{Color} = \begin{array}{r|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline Blue & 0 & 1 & 0 & 1 & 0 & 1 \\ Green & 0 & 0 & 1 & 0 & 0 & 0 \\ Red & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

Pairing up dimensions

The Khatri-Rao product of t_{Model} and t_{Color} is matrix

		1	2	3	4	5	6
Chevy	*Blue*	0	1	0	0	0	0
Chevy	*Green*	0	0	0	0	0	0
Chevy	*Red*	1	0	0	0	0	0
Ford	*Blue*	0	0	0	1	0	1
Ford	*Green*	0	0	1	0	0	0
Ford	*Red*	0	0	0	0	1	0

of type $|Model| \times |Color| \longleftarrow n$.

It tells in which rows the particular pairs of values turn up.

In other words, this matrix is the projection $t_{Model \times Color}$ of the Cartesian product of the two dimensions. In general:

$$t_{A \times B} \quad = \quad t_A \odot t_B$$

All dimensions together

$$
\begin{array}{ll}
& t_{Model \times Year \times Color} \\
= & t_{Model} \odot t_{Year} \odot t_{Color} \\
= & \begin{array}{r|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\mathit{Chevy}\ 1990\ \mathit{Blue} & 0 & 1 & 0 & 0 & 0 & 0 \\
\mathit{Chevy}\ 1990\ \mathit{Green} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Chevy}\ 1990\ \mathit{Red} & 1 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Chevy}\ 1991\ \mathit{Blue} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Chevy}\ 1991\ \mathit{Green} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Chevy}\ 1991\ \mathit{Red} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Ford}\ 1990\ \mathit{Blue} & 0 & 0 & 0 & 1 & 0 & 0 \\
\mathit{Ford}\ 1990\ \mathit{Green} & 0 & 0 & 1 & 0 & 0 & 0 \\
\mathit{Ford}\ 1990\ \mathit{Red} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Ford}\ 1991\ \mathit{Blue} & 0 & 0 & 0 & 0 & 0 & 1 \\
\mathit{Ford}\ 1991\ \mathit{Green} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathit{Ford}\ 1991\ \mathit{Red} & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
\end{array}
\tag{12}
$$

is the projection capturing the dimensional part of raw-data table.

Multi-dimensional summaries

Multidimensional cross-tabulations are obtained via the same formula (5) just by supplying higher-rank projections, for instance

		1990	1991	ALL
Chevy	*Blue*	87	0	87
Chevy	*Green*	0	0	0
Chevy	*Red*	5	0	5
Ford	*Blue*	99	7	106
Ford	*Green*	64	0	64
Ford	*Red*	0	8	8
	ALL	255	15	270

corresponding to $A = Model \times Color$ and $B = Year$ in (5).

Computing the whole cube

General formula

$$\bigominus_{i=0}^{\#D} \Big(\bigominus_{j \in \binom{D}{i}} \big(\bigodot_{d \in j} t_d \cdot [\![T]\!]_{Unit} \cdot !^{\circ} \big) \Big) \tag{13}$$

for

- *Unit* is the chosen measure (quantitative/numerical attribute),
- $\bigodot_i A_i$ iterates $A_1 \odot A_2$ to more than two arguments — mind that

$$! \odot A \quad = \quad A \quad = \quad A \odot ! \tag{14}$$

  holds
- $\ominus_i A_i$ is the n-ary extension of the vertical blocking combinator $\left[\frac{A_1}{A_2}\right]$

MATLAB script

```
function C = Cube(proj,dnum,ndim,lines)
    C = [];
    for i=1:ndim
        ind = nchoosek(1:ndim,i);
        for j=1:size(ind,1)
            C = [C ; kr(proj{ind(j,:)}) * dnum * bang(lines)'];
        end
    end
    C = lift([C; bang(lines) * dnum * bang(lines)']);
end
```

By running

```
>> Cube({m,y,c},d,3,6)
```

in MATLAB, where variables m, y, c and d respectively hold t_{Model}, t_{Year}, t_{Color}, $[\![T]\!]_{Sales}$ we will obtain the cube previously shown.

Summary

Summing up:

- A **no-SQL** approach to data mining
- Formal **semantics** implicit in LA encoding
- Other OLAP operations such as **roll-up** easy to implement in typed LA.

Moreover:

- All constructions in the approach **embarrassingly parallel** (Foster, 1995).
- Projection and diagonal matrices are **sparse**, therefore calling for suitably optimization in a parallel environment (Williams et al., 2009).

Promises **inexpensive** parallel implementation of OLAP/data mining in multi-core, **lap-top** machines, eg. on top of MS Excel or OpenOffice.

Putting ideas on paper

Draft paper

Details in `http://alfa.di.uminho.pt/~hmacedo/wiki/doku.php?id=blog:2011:0411_do_the_middle_letters`

We regard this as a practical application of the typed LA approach we are developing under the **"matrices as arrows"** motto (Macedo and Oliveira, 2010)

Ian Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995. ISBN 0201575949.

H.D. Macedo and J.N. Oliveira. Matrices As Arrows! A Biproduct Approach to Typed Linear Algebra. In *Mathematics of Program Construction*, volume 6120 of *Lecture Notes in Computer Science*, pages 271–287. Springer, 2010.

Samuel Williams, Leonid Oliker, Richard Vuduc, John Shalf, Katherine Yelick, and James Demmel. Optimization of sparse matrix-vector multiplication on emerging multicore platforms. *Parallel Comput.*, 35:178–194, March 2009. ISSN 0167-8191. doi: 10.1016/j.parco.2008.12.006. URL `http://portal.acm.org/citation.cfm?id=1513001.1513318`.