Automated Empirical Optimizations of Software and the ATLAS project*

Software Engineering Seminar

Pascal Spörri

*R. Clint Whaley, Antoine Petitet and Jack Dongarra. Parallel Computing, 27(1-2):3-35, 2001.

Is Search Really Necessary to Generate High-Performance BLAS?

Kamen Yotov and Xiaoming Li and Gang Ren and Maria Garzaran and David Padua and Keshav Pingali and Paul Stodghill
PROCEEDINGS OF THE IEEE, VOL. 93, NO. 2, FEBRUARY 2005

INTRODUCTION

BLAS (Basic Linear Algebra Subprograms)

- **Level 1**

  Vector operations

$$\boldsymbol{y} \leftarrow \alpha \boldsymbol{x} + \boldsymbol{z}$$

- **Level 2**

  Matrix-Vector operations

$$\boldsymbol{y} \leftarrow \alpha \boldsymbol{A} \boldsymbol{x} + \boldsymbol{z}$$

- **Level 3**

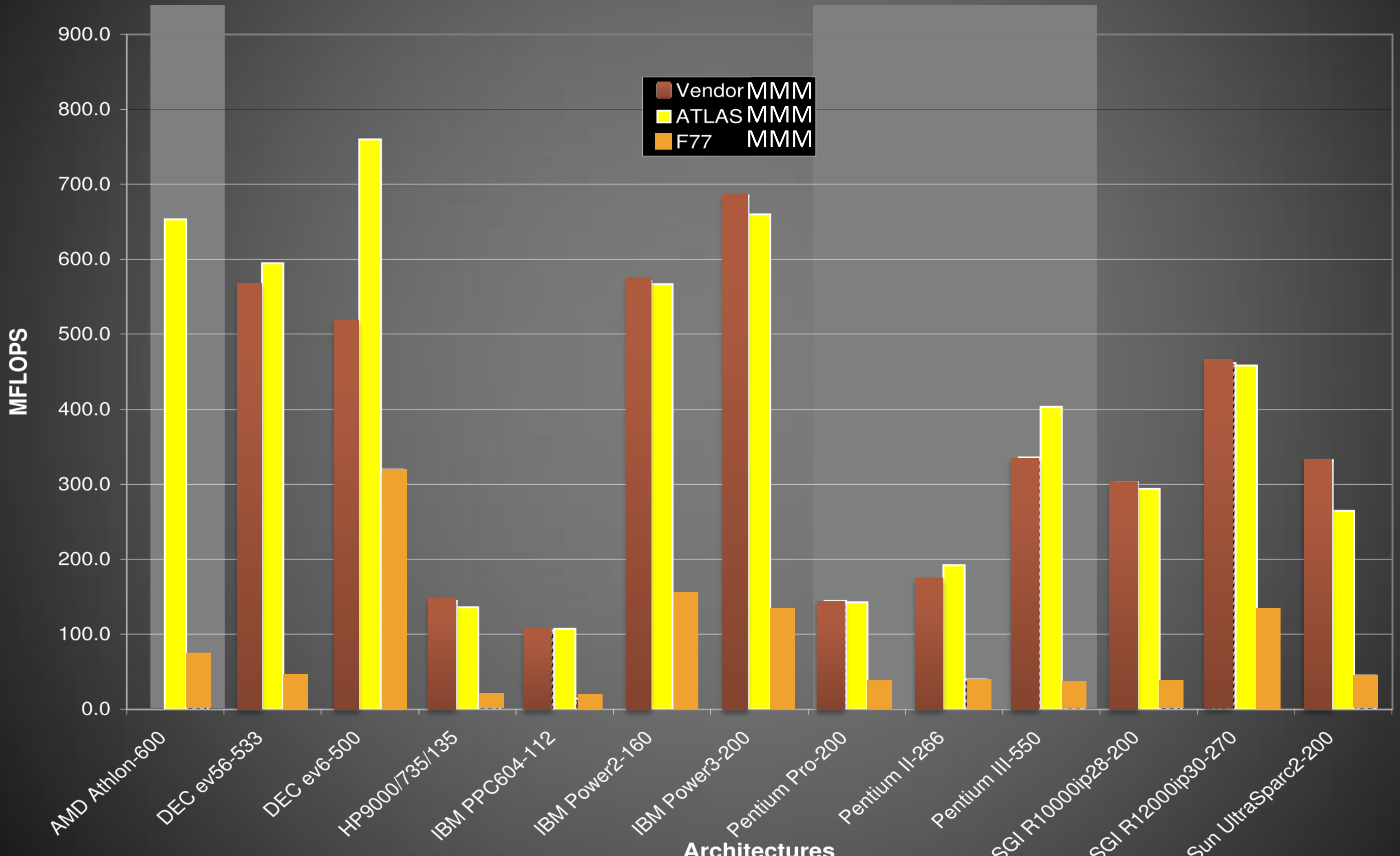
  Matrix-Matrix operations

$$\boldsymbol{D} \leftarrow \alpha \boldsymbol{A} \boldsymbol{B} + \beta \boldsymbol{C}$$

ATLAS
(Automatically Tuned Linear Algebra Software)

- Implements BLAS
- Applies empirical optimization techniques to source code to generate an optimized library
- Fully automatic
- Produces ANSI-C code

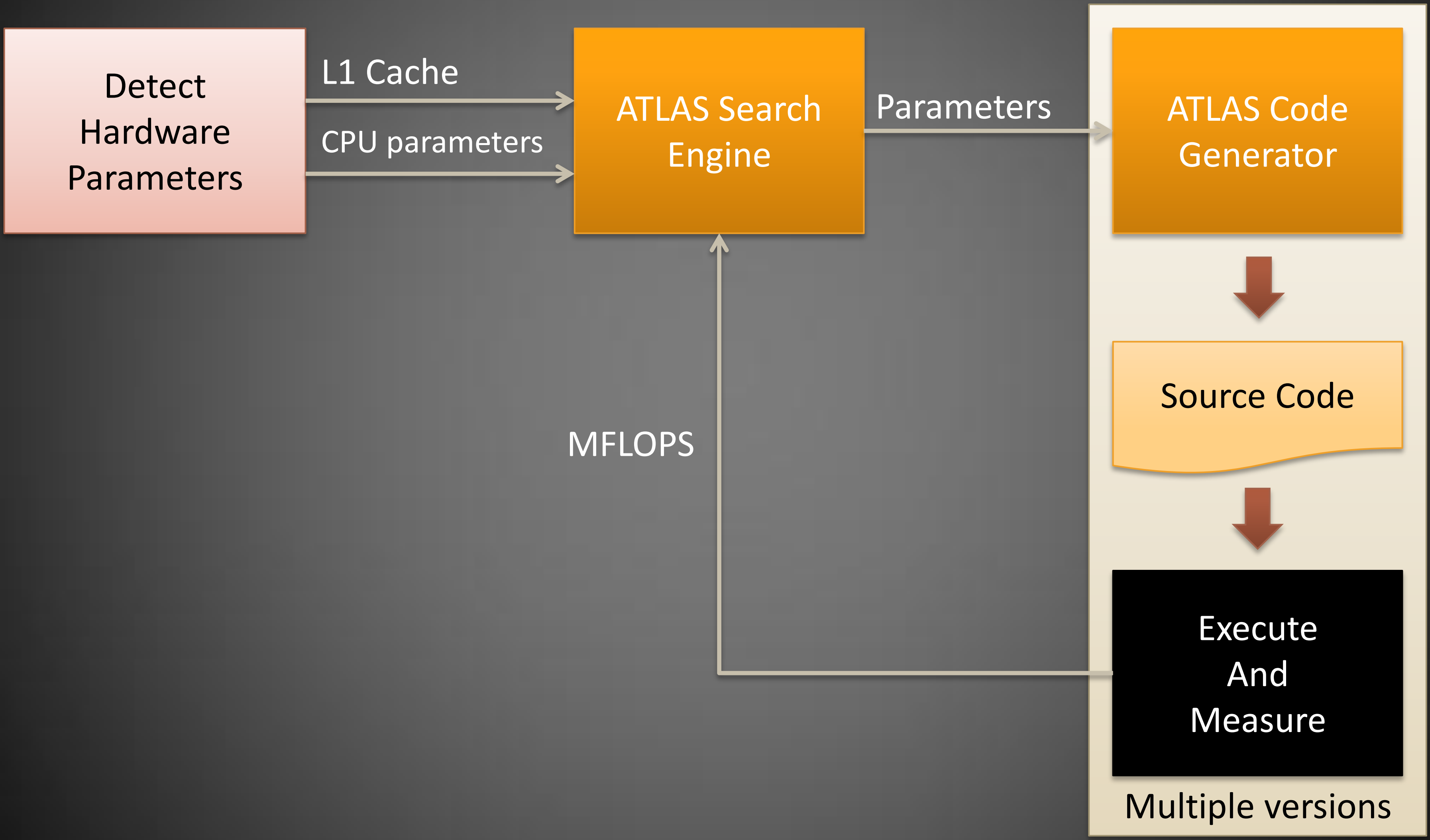
ATLAS Matrix-Matrix Multiplication



"Automated Empirical Optimization of Software and the ATLAS project" by R. Clint Whaley, Antoine Petitet and Jack Dongarra. Parallel Computing, 27(1-2):3-35, 2001.

ARCHITECTURE

ATLAS Architecture

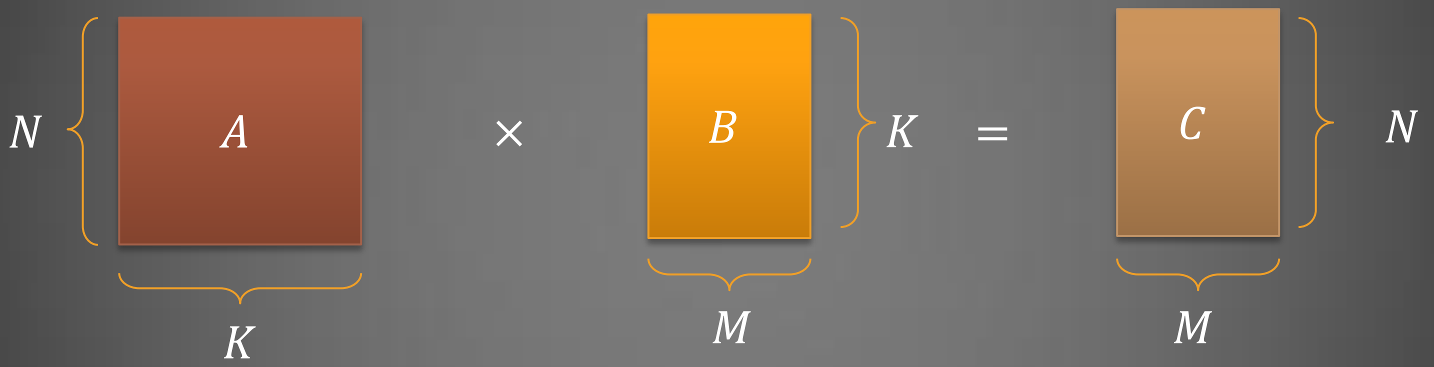


Kamen Yotov and Xiaoming Li and Gang Ren and Maria Garzaran and David Padua and Keshav Pingali and Paul Stodghill
PROCEEDINGS OF THE IEEE, VOL. 93, NO. 2, FEBRUARY 2005

ATLAS CODE GENERATOR

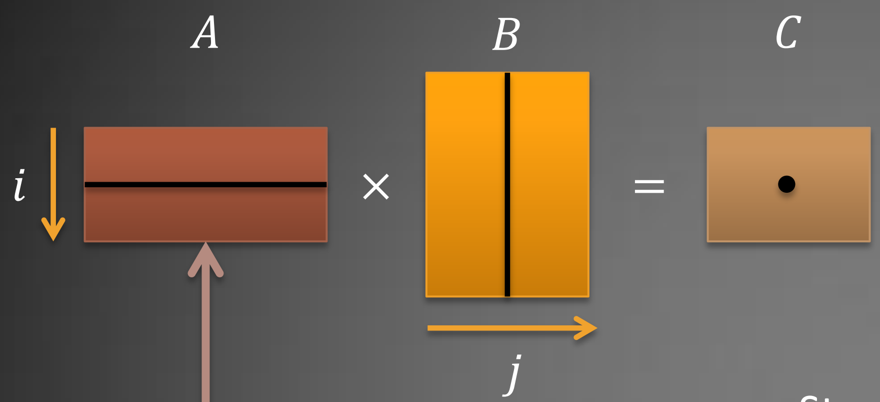
ATLAS Optimizations

- Case: Matrix-Matrix multiplication



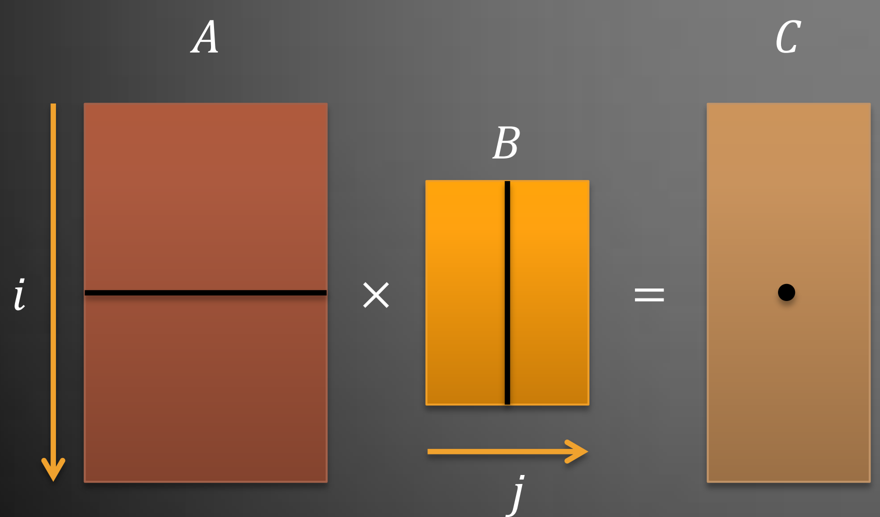
$$
\begin{aligned}
&for\ \ i \in [0:1:N-1] \\
&\qquad for\ j \in [0:1:M-1] \\
&\qquad\qquad for\ k \in [0:1:K-1] \\
&\qquad\qquad\qquad C_{i,j} = C_{i,j} + A_{j,k} \times B_{k,j}
\end{aligned}
$$

Loop Ordering



$$\begin{aligned}&for\ \ \boldsymbol{j} \in [0:1:N-1]\\&\quad for\ \boldsymbol{i} \in [0:1:M-1]\\&\qquad for\ \boldsymbol{k} \in [0:1:K-1]\\&\qquad\quad C_{i,j} = C_{i,j} + A_{j,k} \times B_{k,j}\end{aligned}$$

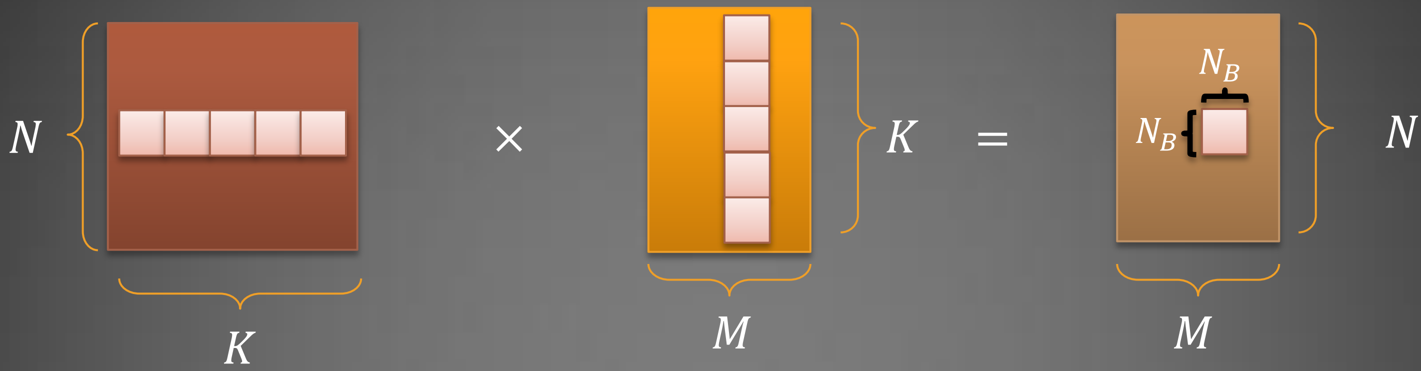
Store ***A*** in Cache



$$\begin{aligned}&for\ \ \boldsymbol{i} \in [0:1:N-1]\\&\quad for\ \boldsymbol{j} \in [0:1:M-1]\\&\qquad for\ \boldsymbol{k} \in [0:1:K-1]\\&\qquad\quad C_{i,j} = C_{i,j} + A_{j,k} \times B_{k,j}\end{aligned}$$

Store ***B*** in Cache

1st Level Blocking



$for\ \ i \in [0: \boldsymbol{N_B}: N-1]$
$for\ j \in [0: \boldsymbol{N_B}: M-1]$
$for\ k \in [0: \boldsymbol{N_B}: K-1]$

N_B is choosen in such that the working set fits into L_1

$for\ j' \in [j: 1: j + N_B - 1]$
$for\ i' \in [i: 1: i + N_B - 1]$
$for\ k' \in [k: 1: k + N_B - 1]$

$$C_{i',j'} = C_{i',j'} + A_{j',k'} \times B_{k',j'}$$

2nd Level Blocking

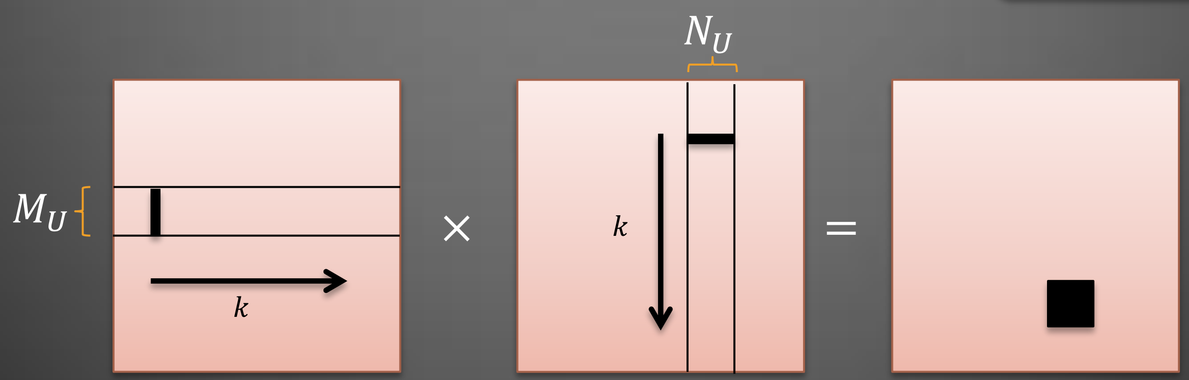
$for\ \ i \in [0{:}\,\boldsymbol{N_B}{:}\,N-1]$
$\quad for\ j \in [0{:}\,\boldsymbol{N_B}{:}\,M-1]$
$\quad\quad for\ k \in [0{:}\,\boldsymbol{N_B}{:}\,K-1]$

$for\ j' \in [j{:}\,\boldsymbol{N_U}{:}\,j+N_B-1]$
$for\ i' \in [i{:}\,\boldsymbol{M_U}{:}\,i+N_B-1]$
$for\ k' \in [k{:}\,\boldsymbol{K_U}{:}\,k+N_B-1]$

$for\ k'' \in [k'{:}\,1{:}\,k'+K_U-1]$
$for\ j'' \in [j'{:}\,1{:}\ j'+N_U-1]$
$for\ i'' \in [i'{:}\,1{:}\ i'+M_U-1]$
$\quad C_{i'',j''} = C_{i'',j''} + A_{j'',k''} \times B_{k'',j''}$

$M_U + N_U + M_U \times N_U \leq N_R$

Unroll Loop



Graphic from “How To Write Fast Numerical Code: A Small Introduction”
Srinivas Chellappa, Franz Franchetti, and Markus Püschel

Scalar Replacement

- Replace array accesses with scalars

Stored in memory

```
double t[2];
for (i=0; i<8; i++) {



    t[0] = x[2*i] + x[2*i+1];
    t[1] = x[2*i] - x[2*i+1];
    y[2*i] = t[0] * D[2*i];
    y[2*i+1] = t[0] * D[2*i];
}
```

Store intermediate results in registers

```
double t0, t1, x0, x1, D0;
for (i=0; i<8; i++) {
    x0 = x[2*i];
    x1 = x[2*i+1];
    D0 = D[2*i];
    t0 = x0 + x1;
    t1 = x0 - x1;
    y[2*i] = t0 * D0;
    y[2*i+1] = t1 * D0;
}
```

Store for reuse (annotation pointing to `D0 = D[2*i];`)

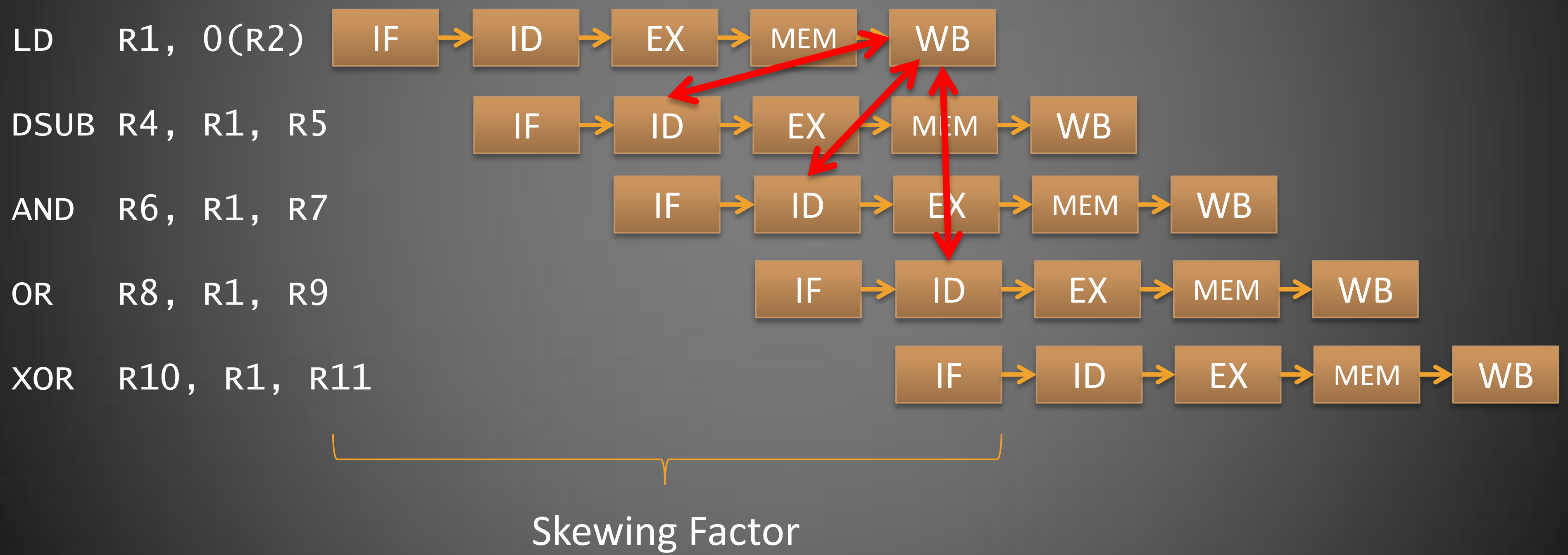
Scalar Replacement

```
a11 = A[1][1]
a12 = A[1][2]
a13 = A[1][3]
a14 = A[1][4]
…
b11 = B[1][1]
b12 = B[1][2]
b13 = B[1][3]
b14 = B[1][4]
…
```

```
c11 =  a11*b11
c11 += a12*b21
c11 += a13*b31
…
c12 =  a11*b12
c12 += a12*b22
c12 += a13*b32
…

C[1][1] = c11
C[1][2] = c12
C[1][3] = c13
```

Data Hazards



Pipeline Scheduling

Interleave mul and add sequences

mul_1
mul_2
…
mul_{L_S}

Skewing factor L_S

add_1
mul_{L_S+1}
add_2
mul_{L_S+2}
add_3
…

Pipeline Scheduling

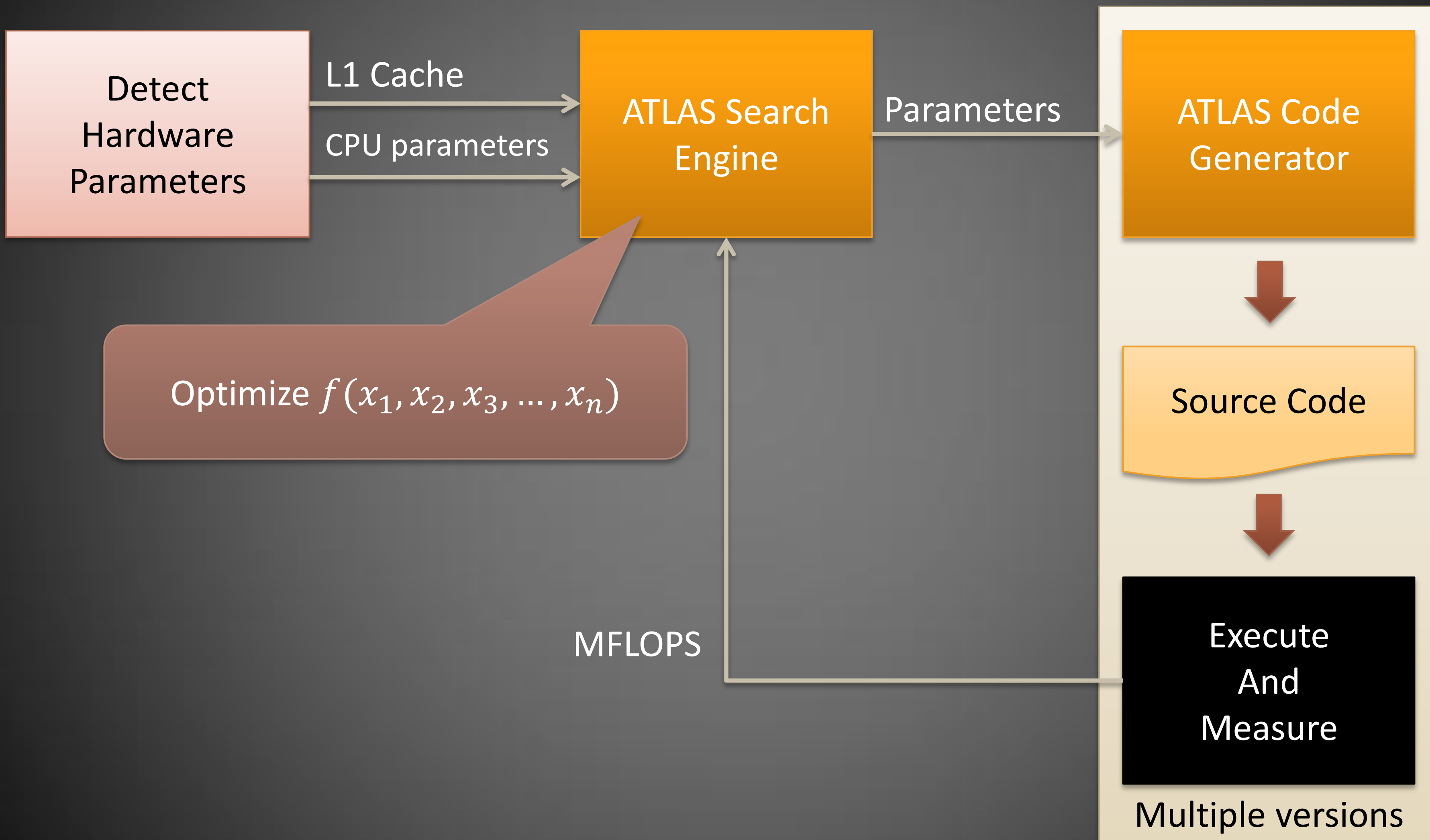
```
a11 = A[1][1]
a12 = A[1][2]
a13 = A[1][3]
a14 = A[1][4]
…
b11 = B[1][1]
b12 = B[1][2]
b13 = B[1][3]
b14 = B[1][4]
…
```

```
c11 =  a11*b11
c12 =  a11*b12
…
c11 += a12*b21
c12 += a12*b22
…
c11 += a13*b31
c12 += a13*b32
…

C[1][1] = c11
C[1][2] = c12
C[1][3] = c13
```

EMPIRICAL OPTIMIZATION IN ATLAS

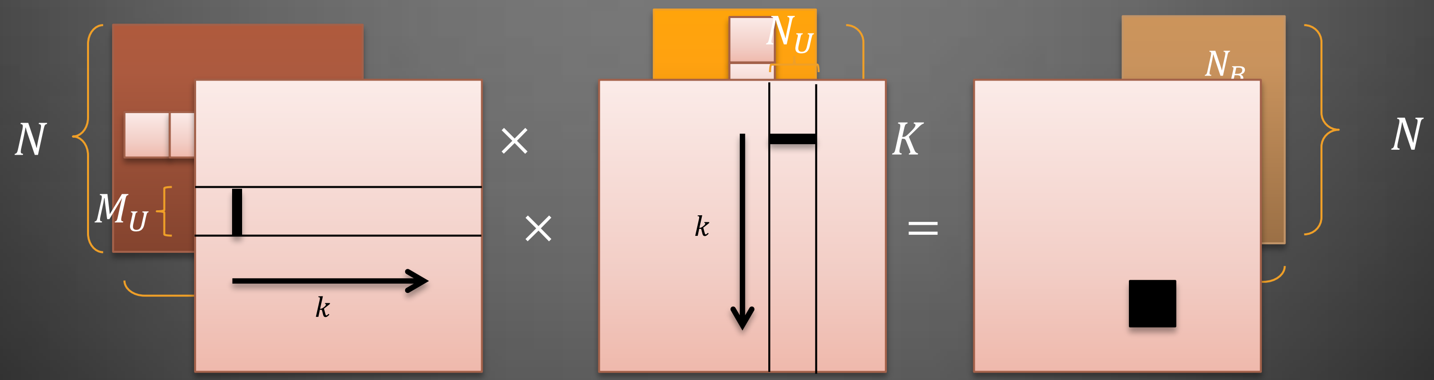
ATLAS Architecture



Kamen Yotov and Xiaoming Li and Gang Ren and Maria Garzaran and David Padua and Keshav Pingali and Paul Stodghill
PROCEEDINGS OF THE IEEE, VOL. 93, NO. 2, FEBRUARY 2005

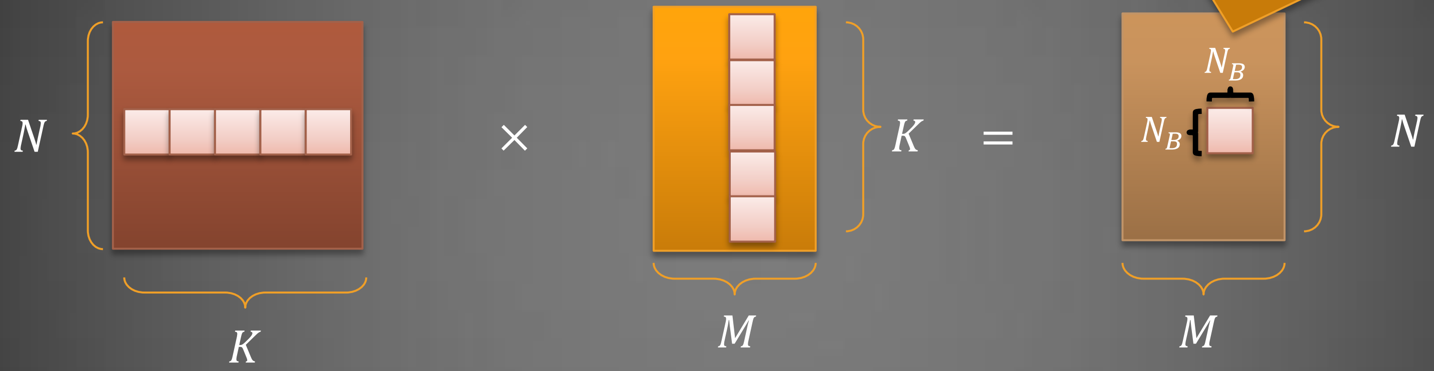
Optimization Order

1. Find best block size for outer loop
2. Find best block sizes for inner loop
3. Find best skewing factor
4. Find best parameters for scheduling of loads
5. Additional parameters



Search for best Outer Loop Size

Restrict search space $16 \leq N_B \leq \min(80, \sqrt{L_1\ Size})$



- N_B must be a multiple of 4
- Use fastest version

Try with and without unrolling the inner loop

DISCUSSION

Comparison to PhiPAC

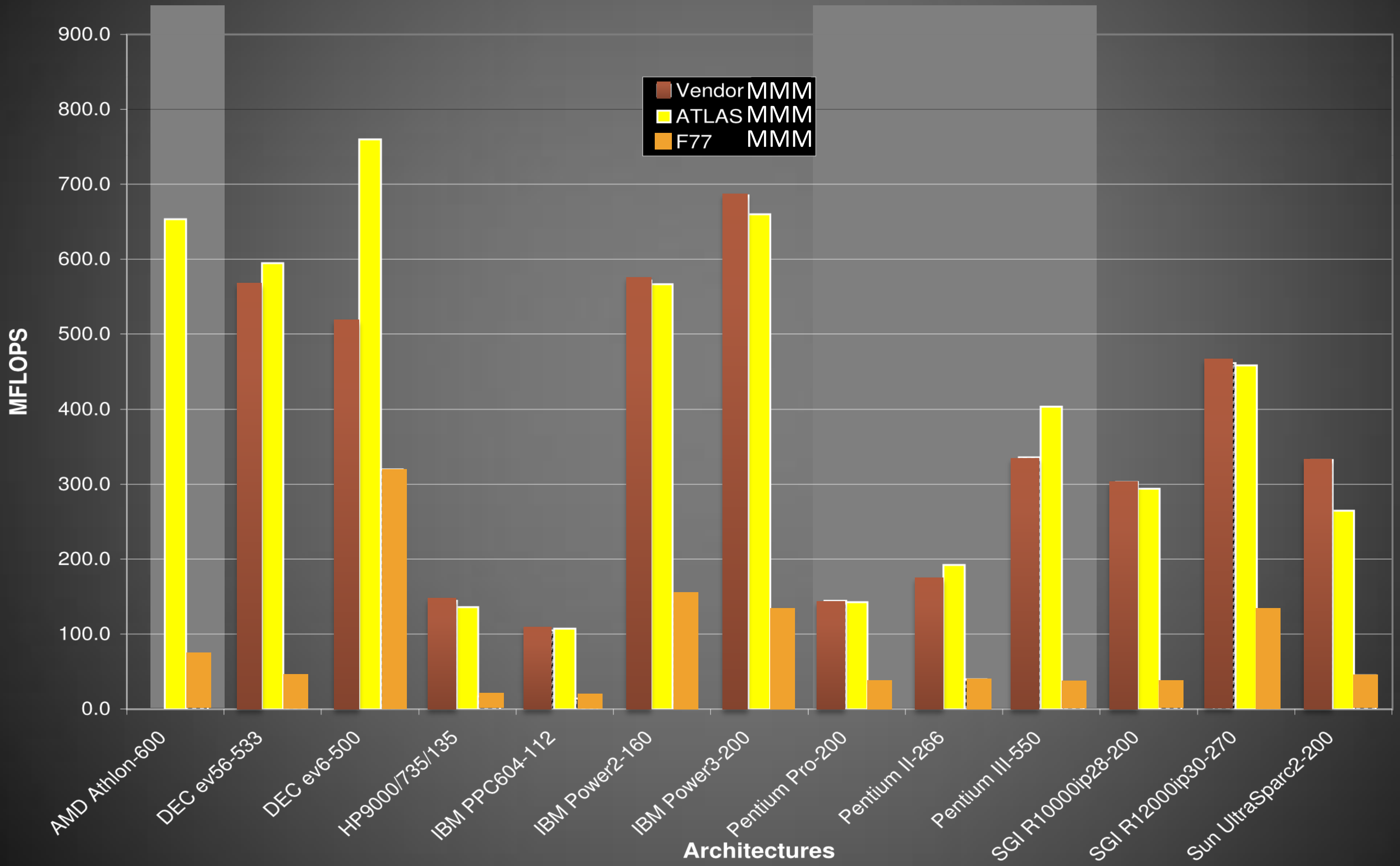
PhiPAC

- Coding methodology to write fast code
- Precursor for ATLAS
- Specialized Code Generator for BLAS Matrix-Matrix Multiplication
- Optimizes parameters for inner and outer loop

ATLAS

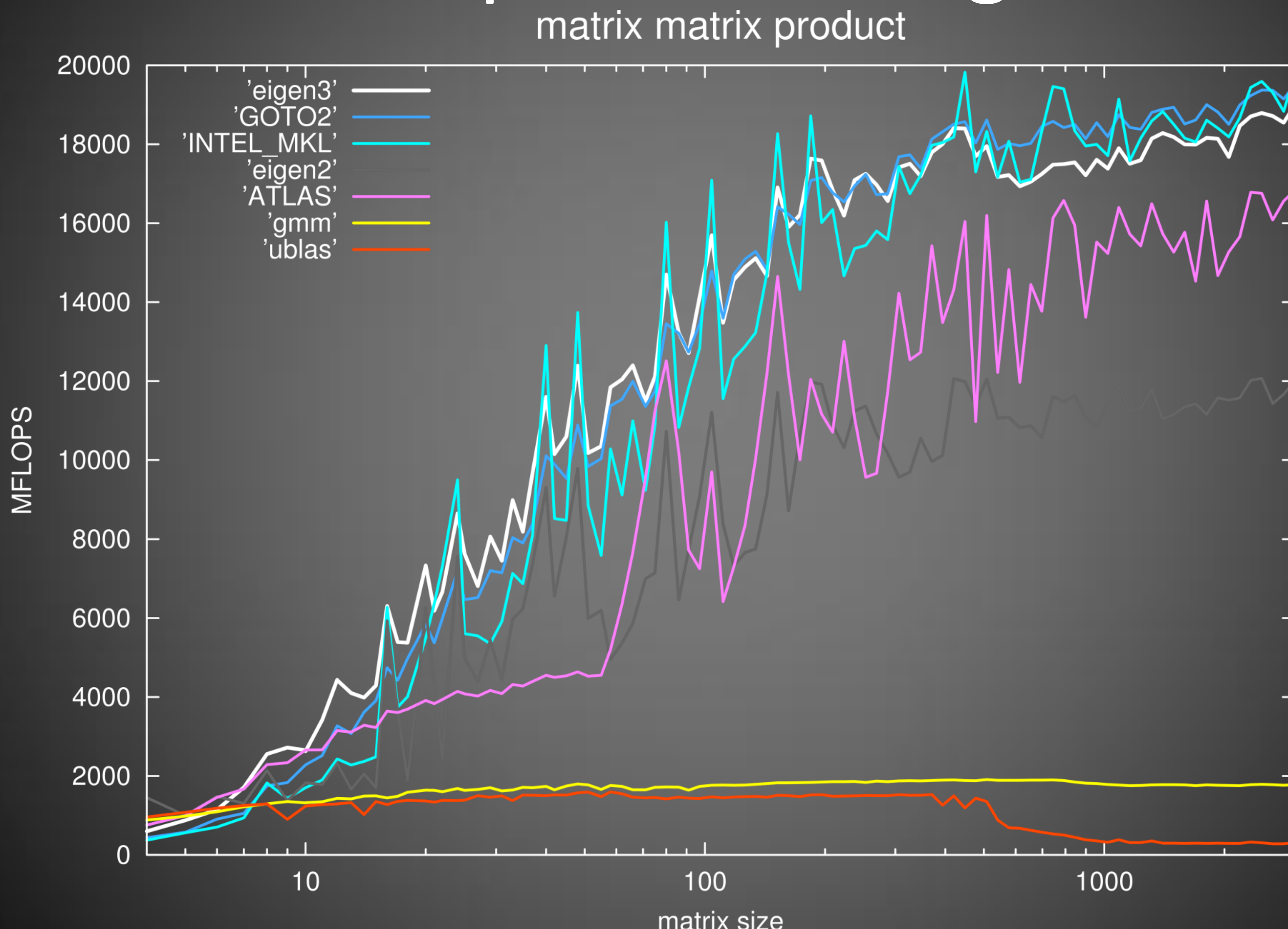
- Library generator
- Automatic generation of optimized BLAS
- Support for handcoded routines

ATLAS Matrix-Matrix Multiplication



"Automated Empirical Optimization of Software and the ATLAS project" by R. Clint Whaley, Antoine Petitet and Jack Dongarra. Parallel Computing, 27(1-2):3-35, 2001.

Comparison to eigen



http://eigen.tuxfamily.org/index.php?title=Benchmark

Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66GHz (x86_64)

Conclusion

Pro

- Fast method to generate an optimized library for a new platform
- Supports hand optimized code
- Implements BLAS

Contra

- Needs constant adjustment to support new architectures
- Outdated

Further Information

- ATLAS Project

  http://math-atlas.sourceforge.net/

- BLAS

  http://netlib.org/blas/