# Algorithms and Computation in Signal Processing

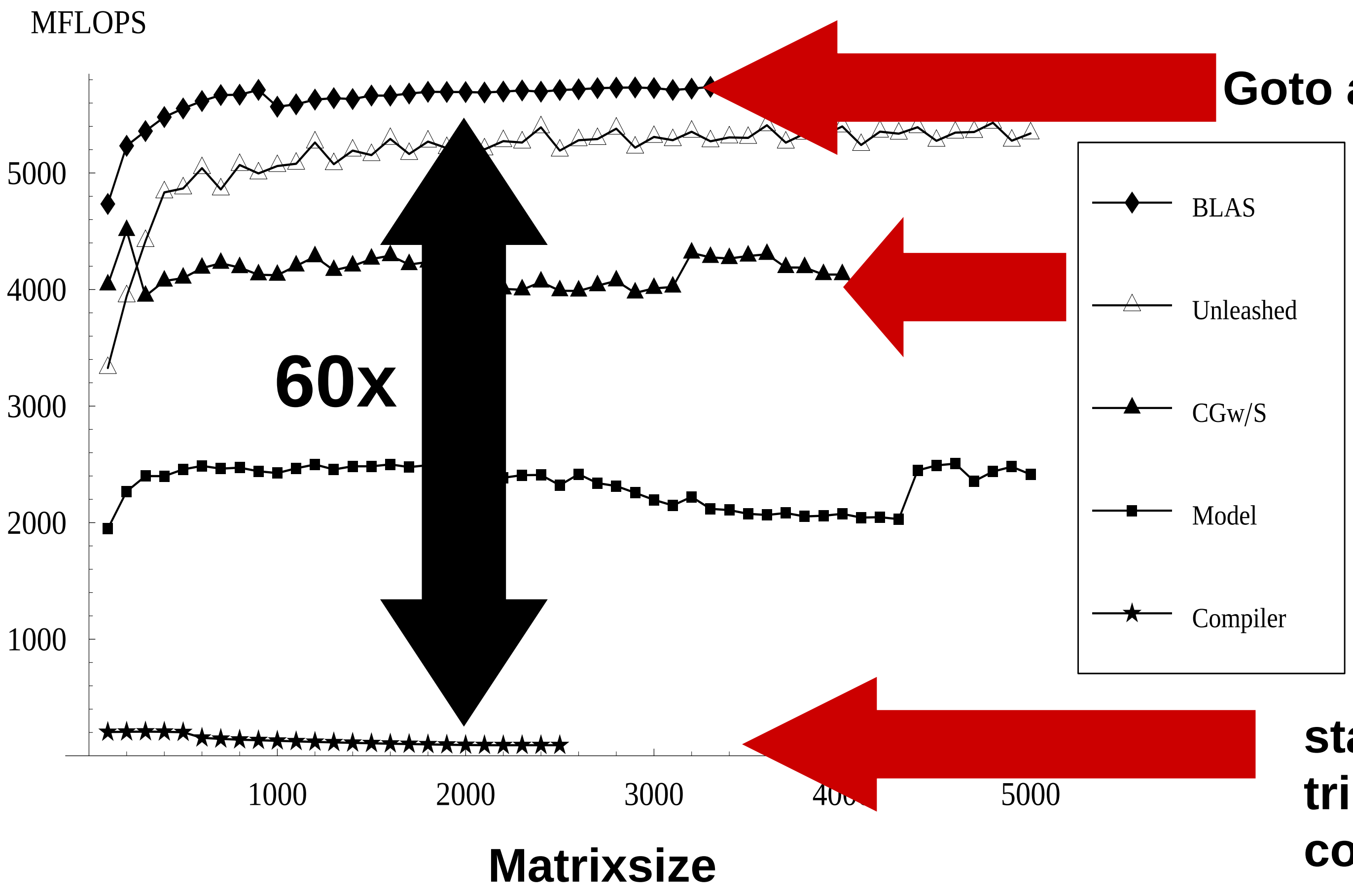
special topic course 18-799B
spring 2005
6th Lecture Jan. 27, 2005

Instructor: Markus Pueschel

TA: Srinivas Chellappa

# Code Generation for MMM (ATLAS)

# The Problem: Matrix-matrix Multiplication



Goto assembly code

ATLAS
generated code

standard:
triple loop +
compiler
optimization

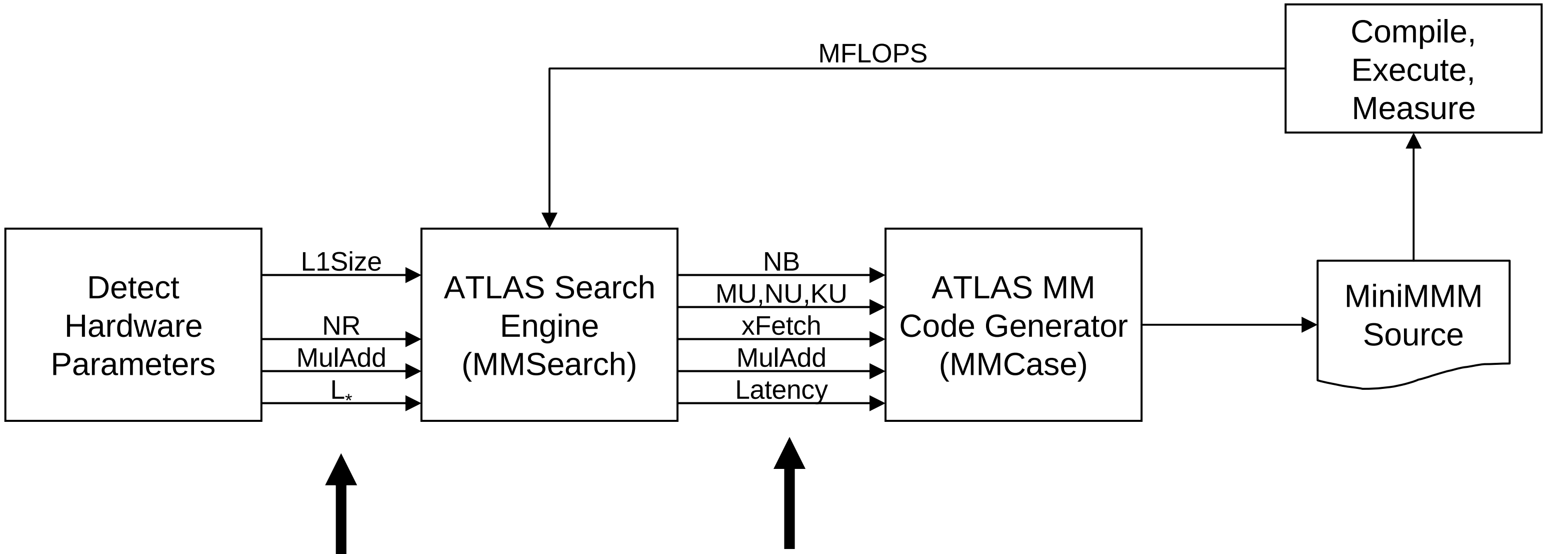
Now we will learn how it works

*graph: Pingali, Yotov, Cornell U.*

# ATLAS

- Successor of PhiPAC, Generator for BLAS link
  (Whaley, Petitet, Dongarra)
- People can also contribute handwritten code
- The generator uses empirical search over implementation alternatives to find the fastest implementation
- We focus on BLAS3 MMM
- Search only over $2n^3$ algorithms
  (cost equal to direct method)

# ATLAS Architecture


MFLOPS
Compile, Execute, Measure
Detect Hardware Parameters
L1Size
NR
MulAdd
$L_*$
ATLAS Search Engine (MMSearch)
NB
MU,NU,KU
xFetch
MulAdd
Latency
ATLAS MM Code Generator (MMCase)
MiniMMM Source

**Search parameters:**

- **span search space**
- **determine code**
- **found by orthogonal line search**

**Hardware parameters:**

- **L1Size: size of L1 data cache**
- **NR: number of registers**
- **MulAdd: fused multiply-add available?**
- **$L_*$ : latency of FP multiplication**

*source: Pingali, Yotov, Cornell U.*

# How ATLAS Works

- Blackboard

# Search in ATLAS

- Search strategy:
  Orthogonal line search = fix all parameters except one and search for the optimal value for this parameter

- Optimize parameters in this order
  - $N_B$
  - $M_U$, $N_U$
  - $K_U$
  - $L_s$
  - …

- Details in paper distributed in class

# Principles in ATLAS Code Generation

- Optimization for memory hierarchy = increasing locality (Blocking for cache, blocking for registers)
- Fast basic blocks for small sizes (micro-MMM):
    - Loop unrolling (reduce loop overhead)
    - Scalar replacement (enables better compiler optimization)
    - Add/mult interleaving (better throughput)
    - Skewing (better instruction level parallelism)
- Search for the fastest over a relevant set of algorithm/implementation alternatives