How to Write Fast Code

18-645, spring 2008
8th Lecture, Feb.11th

Instructor: Markus Püschel

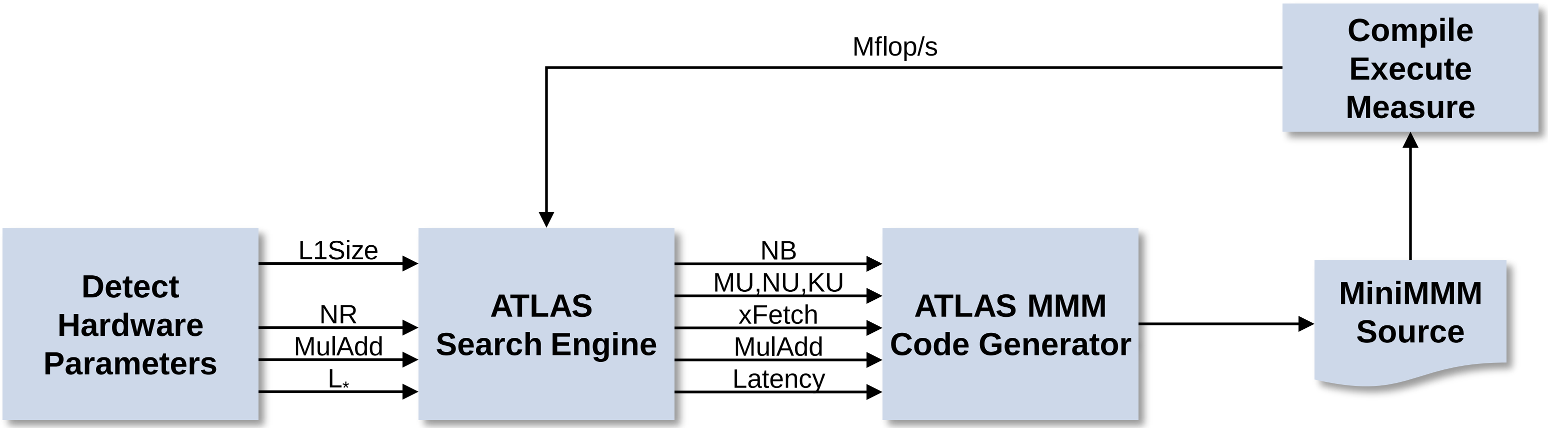
TAs: Srinivas Chellappa (Vas) and Frédéric de Mesmay (Fred)

Today

- **ATLAS: Principles**
- **Model-based ATLAS**

- K. Yotov, X. Li, G. Ren, M. Garzaran, D. Padua, K. Pingali, P. Stodghill, **Is Search Really Necessary to Generate High-Performance BLAS?,** Proceedings of the IEEE, 93(2), pp. 358–386, 2005. Link.

Last Time: ATLAS



- Blocks MMM into mini-MMMs
- Searches for fastest (highest-performance) mini-MMM
- Choices encoded by parameters (N_B, M_U, N_U, …)
- Parameter space bounded through microarchitecture parameters for example: $N_B \leq$ sqrt(cache size)

How it Worked: From Triple Loop to …

```
// MMM loop-nest
for i = 0:$N_B$:N-1
  for j = 0:$N_B$:M-1
    for k = 0:$N_B$:K-1
```

- *ij or ji depending on N and M*
- *Blocking for cache*

```
      // mini-MMM loop nest
      for i' = i:$M_U$:i+$N_B$-1
        for j' = j:$N_U$:j+$N_B$-1
          for k' = k:$K_U$:k+$N_B$-1
```

- *Blocking for registers*

```
            // micro-MMM loop nest
            for k'' = k':1:k'+$K_U$-1
              for i'' = i':1:i'+$M_U$-1
                for j'' = j':1:j'+$N_U$-1
```

- *unrolling*
- *scalar replacement*
- *add/mult interleaving*
- *skewing*

Search parameters: N_B, M_U, N_U, K_U, L_S

Principles used in ATLAS Optimization

- **Optimization for memory hierarchy = increasing locality**
  - Blocking for cache, blocking for registers
  - Done by loop tiling and loop exchange

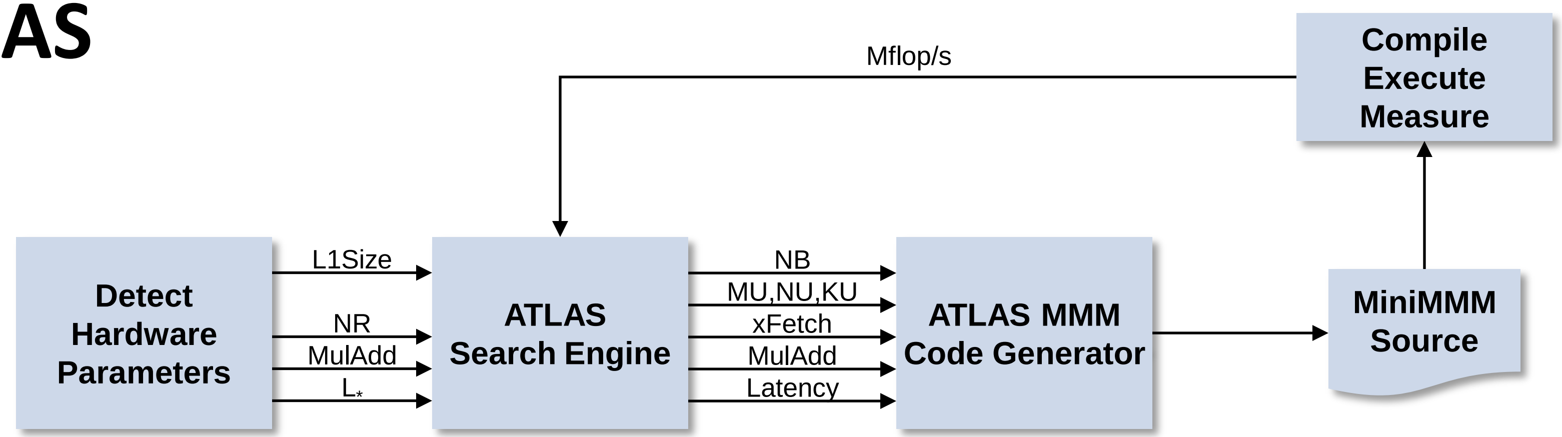
- **Fast basic blocks for small sizes (micro-MMM):**
  - Loop unrolling (reduce loop overhead)
  - Scalar replacement (enables better compiler optimization)
  - Add/mult interleaving and skewing (instruction level parallelism)

- **Search for the fastest over a relevant set of algorithm/implementation alternatives**

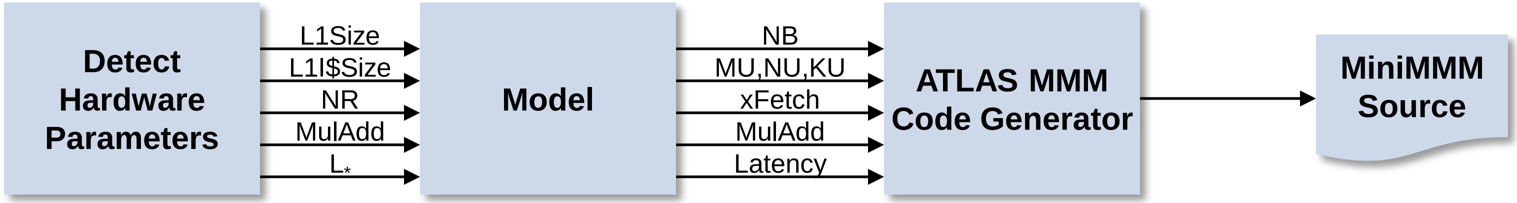
MMM: So far

- We learned a set of optimization techniques for the memory hierarchy
- But there are degrees of freedom
- Practical problem: How to choose them without implementing search?
- Scientific problem: How to choose them from an understanding of the microarchitecture?

ATLAS



Model-Based ATLAS



- **Search for parameters replaced by model to compute them**
- **More hardware parameters needed**

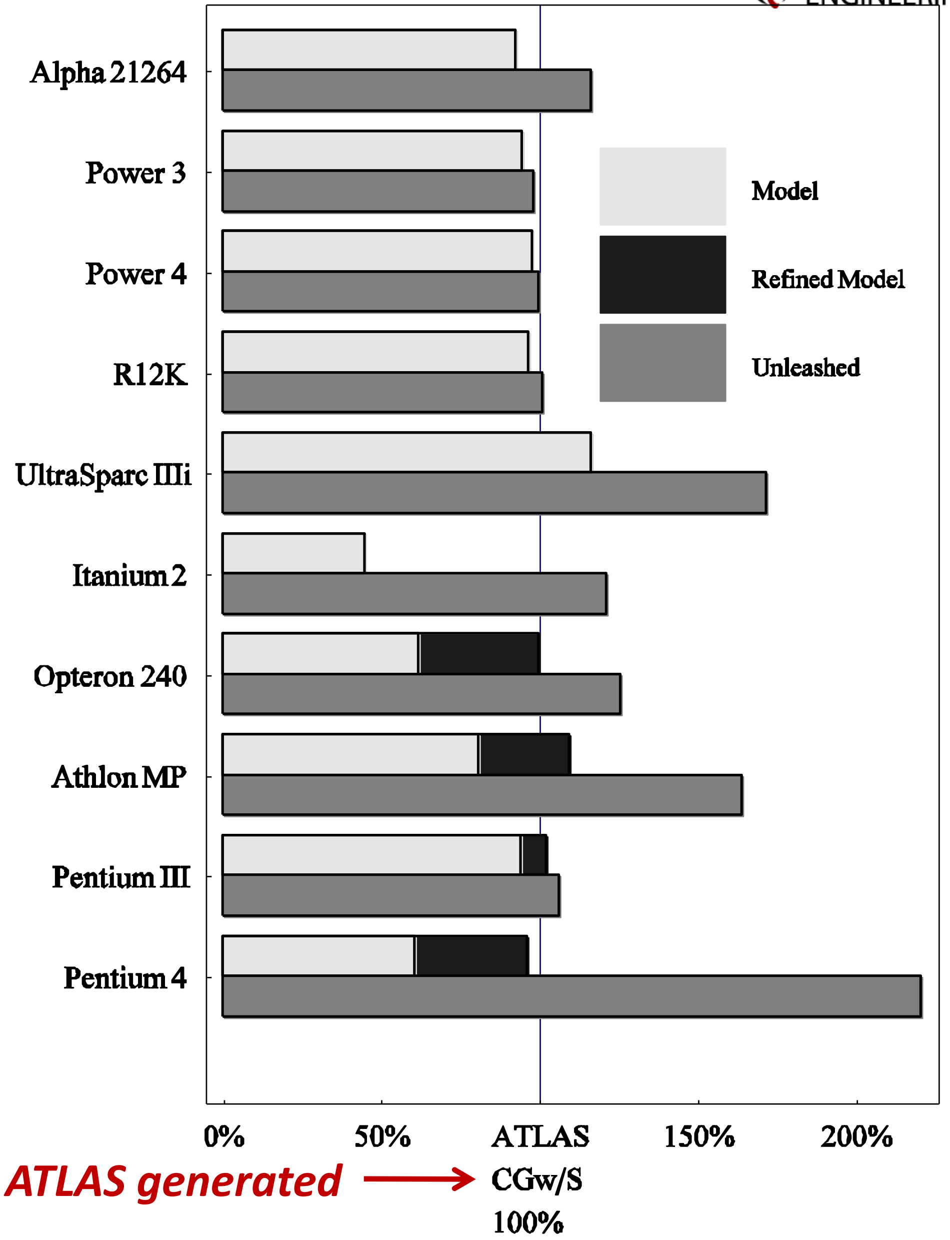
source: Pingali, Yotov, Cornell U.

Model-Based ATLAS: Details

- Blackboard

Experiments

- **Unleashed:** Not generated = hand-written contributed code
- **Refined model** for computing register tiles on x86
- Blocking is for L1 cache
- Blocking for L1 cache usually better code but problematic if MMM used as subroutine
- Model-based comparable to search-based (except Itanium)



ATLAS generated →

graph: Pingali, Yotov, Cornell U.