High-performance Matrix Computations

Prof. **Paolo Bientinesi**

pauldj@aices.rwth-aachen.de

Performance

Quality of an algorithm

Metrics?

- Execution time
- Complexity
- Stability
- Accuracy
- Memory usage
- Memory/network accesses
- ...

Execution time

Pros? Cons?

- It says how fast an algorithm is
- “Does not measure the memory usage”
- “It might depend on the input”
- “It depends on the processor”
- **It does not measure how well the algorithm takes advantage of a given architecture**

- Quality of the algorithm IN RELATION to the potential of the architecture
- Potential of an architecture?

Theoretical Peak Performance

TPP = #cores * frequency * #flops/cycle

- what is frequency? GHz
- what is a cycle?
  processor can initiate a new operation
- what is a flop?
  floating point operation

TPP is unattainable. Why?

Back to performance

- Theroretical peak performance $\rightarrow$ unattainable
- Practical peak performance? (practical = attainable)
- DGEMM (BLAS)
  Double GEneral Matrix Matrix multiply
  Basic Linear Algebra Subroutines
- Peak performance $\equiv$ DGEMM
- How to compute DGEMM's performance?

Performance

$$\text{Perf} = \frac{\#\text{ops}}{\text{exec. time}}$$

- What is #ops?
- What is #ops in your algorithm?
- What if the algortihm is iterative?
  “iterative algorithm” != “loop-based”

$$\text{Efficiency} = \frac{\text{Perf}}{\text{Peak Perf}}$$

Time vs. Performance

- Can you cheat time?
- Can you cheat performance?
- “fast” flops vs. “slow” flops
- $5n \log n$ vs. $2n^2$

USE BOTH!

To GEMM or not to GEMM

“GEMM makes it really hard to win with better algorithms that don’t use it.”

“You can use GEMM stupidly and still win because
on most processors GEMM is the speed-of-light.”

“GEMM is holding back algorithmic innovation for tensor computations.”
J.H.