Computer science (CS) aspects of the fast multipole method

Rich Vuduc
Aparna Chandramowlishwaran (UC Irvine)
Jee Choi (Georgia Tech), Kamesh Madduri (Penn State)
+ many collaborators!

Georgia Tech College of Computing
Computational Science and Engineering

hpcgarage

Far field?

Computer science (CS) aspects of the fast multipole method

Rich Vuduc

Aparna Chandramowlishwaran (UC Irvine)

Jee Choi (Georgia Tech), Kamesh Madduri (Penn State)

+ many collaborators!

Georgia Tech College of Computing

Computational Science and Engineering

hpcgarage

Rahimian
NYU/Courant
Lashuk Tufts
Biros UT Austin
Chandramowlishwaran
UC Irvine
← A CS casualty of the FMM

There is good news and bad news.‡

‡ … with CS research questions, not math questions

Bad news?

Bad news? **We may never be truly done.**[‡]

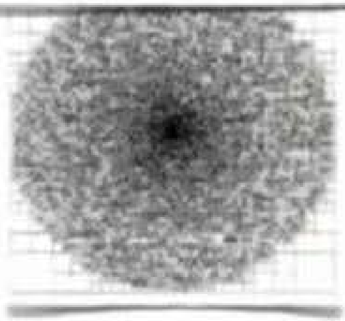
‡ At least not until we close the **performance engineering gap**.

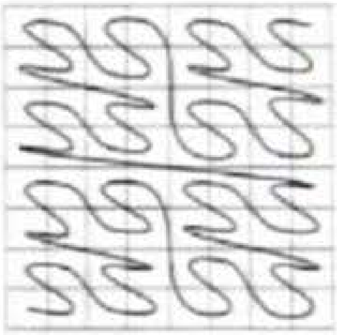
A Parallel Hashed Oct-Tree N-Body Algorithm

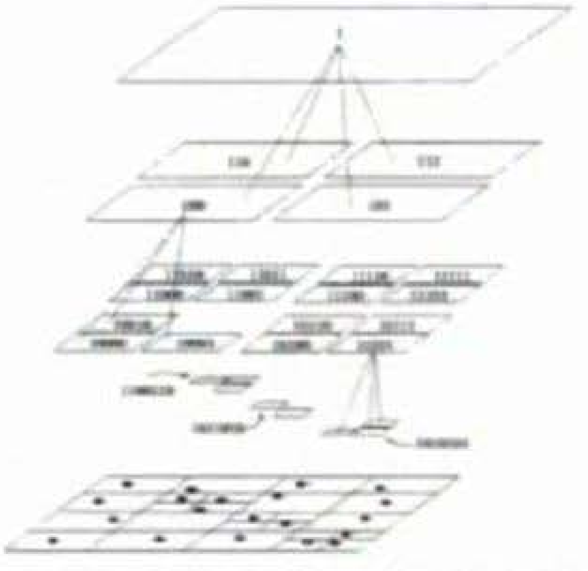
Michael S. Warren*
Theoretical Astrophysics
Mail Stop B288
Los Alamos National Laboratory
Los Alamos, NM 87545

John K. Salmon
Physics Department
206-49
California Institute of Technology
Pasadena, CA 91125

computation stage	*time* (sec)
Domain Decomposition	7
Tree Build	7
Tree Traversal	33
Communication During Traversal	6
Force Evaluation	54
Load Imbalance	7
Total (5.8 Gflops)	114







At later stages of the calculation the system becomes extremely clustered (the density in large clusters of particles is typically 10^6 times the mean density). The number of interactions required to maintain the same accuracy grows moderately as the system evolves. At a slightly increased error bound of 4×10^{-3}, the number of interactions in the clustered system is 2.6×10^{10} per timestep.

computation stage	*time* (sec)
Domain Decomposition	19
Tree Build	10
Tree Traversal	55
Data Communication during traversal	4
Force Evaluation	60
Load Imbalance	12
Total (4.9 Gflops)	160

Network scaling

$$T_{compute} + T_{network} + T_{memory}$$

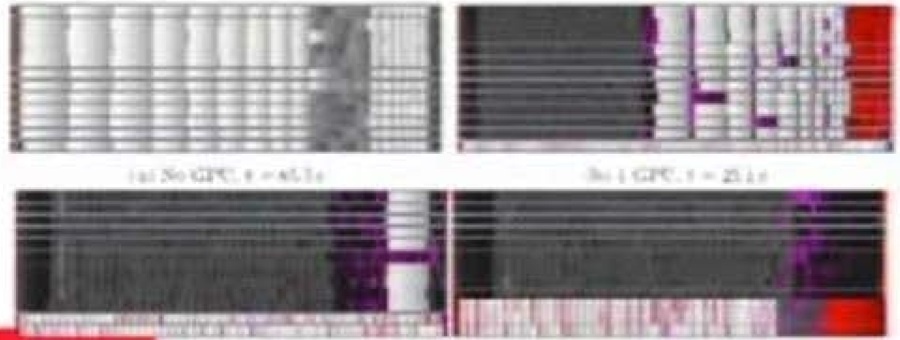
PROVABLY GOOD PARTITIONING AND LOAD BALANCING ALGORITHMS FOR PARALLEL ADAPTIVE N-BODY SIMULATION*

SHANG-HUA TENG†

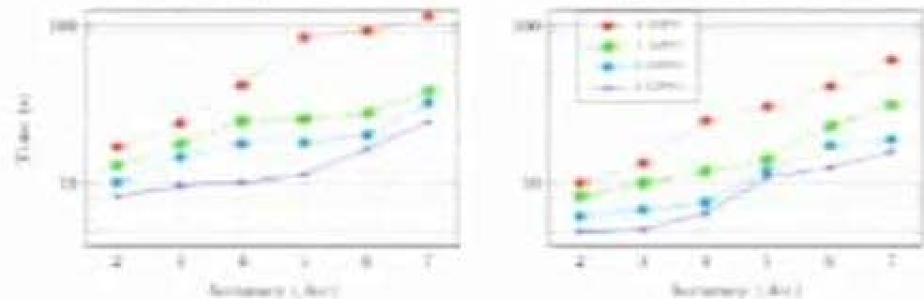
THEOREM 5.1. *Let G be a weighted N-body communication graph (for either BH or FMM) of a set of particles at $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ in $\mathbb{R}^d$ ($d = 2$ or 3). If P is μ-nonuniform, then G can be partitioned into two equally weighted subgraphs by removing at most $O(n^{1-1/d}(\log n + \mu)^{1/d})$ nodes, or by removing edges of at most $O(n^{1-1/d}(\log n + \mu)^{2+1/d})$ total edge weights.*

Recursively applying our partitioning theorem, we can analyze the quality of the recursive bisection scheme for p-way partitioning. (See Simon and Teng [27] for unstructured meshes.)

COROLLARY 5.2. *If G is a (weighted) N-body communication graph for particles that are μ-nonuniform, then G can be partitioned into p equally weighted subgraphs such that the total weight of the removed edges is bounded by $O(p^{1/d}n^{1-1/d}(\log n + \mu)^{2+1/d})$.*

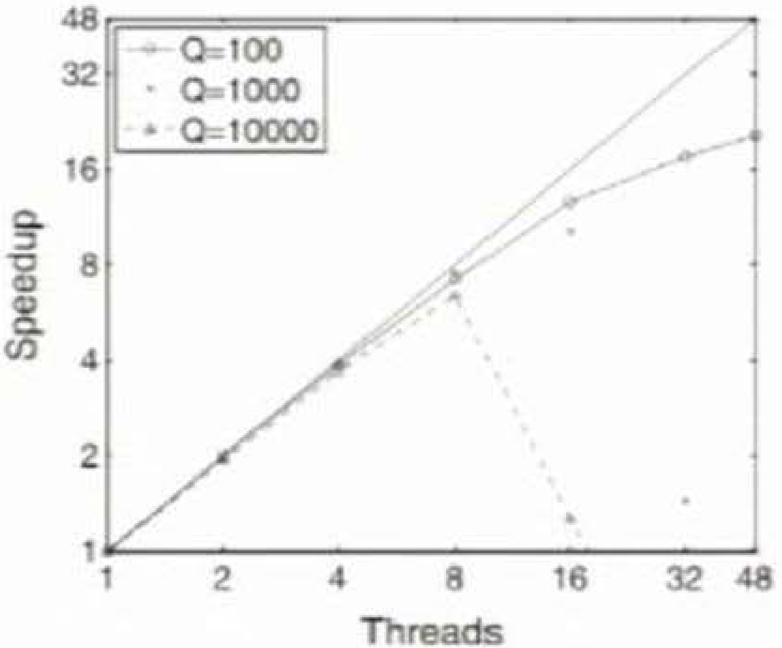


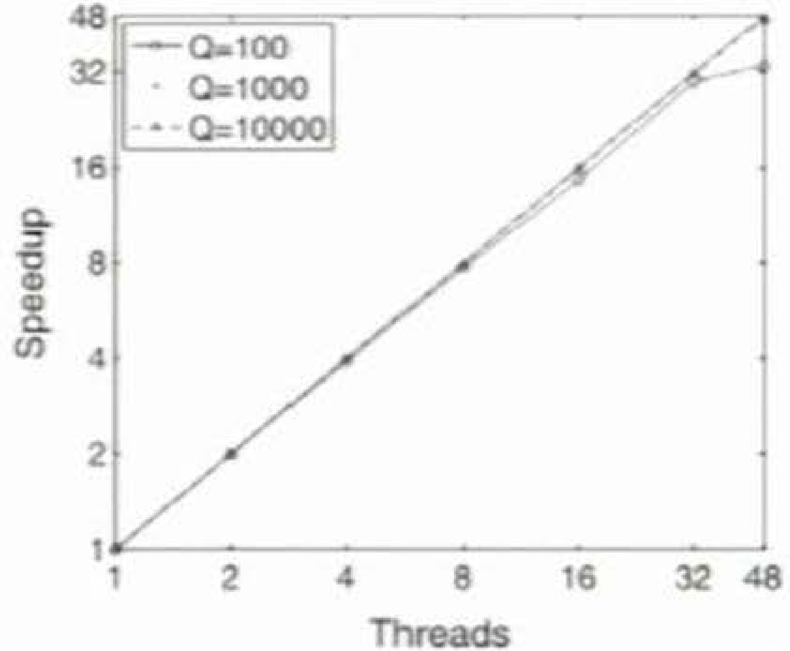
Task-based FMM for heterogeneous architectures

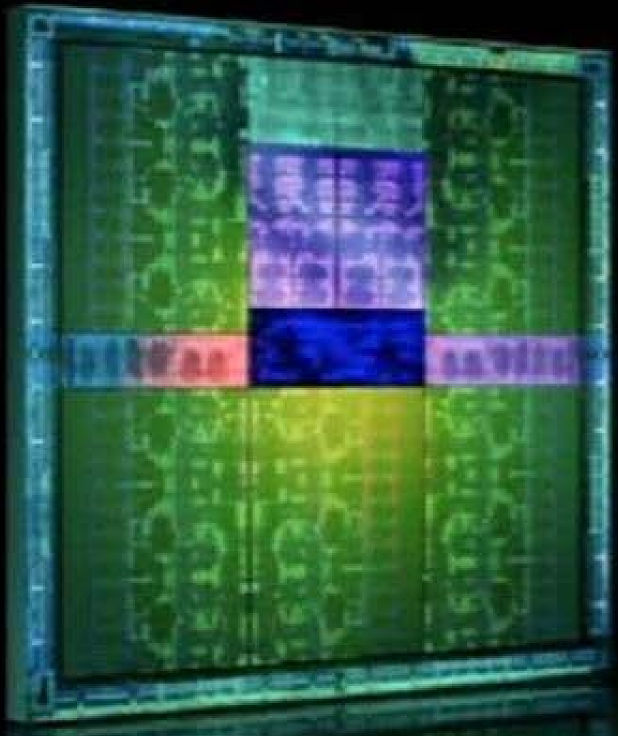


Data-driven execution of fast multipole methods

Hatem Ltaief[1,*,†] and Rio Yokota[2]







The World's Most Powerful GPU

2688	4500	7.1
CUDA Cores	Gigaflops	Billion Transistors

Fast multipole methods on graphics processors

Nail A. Gumerov *, Ramani Duraiswami

Perceptual Interfaces and Reality Laboratory, Computer Science and UMIACS, University of Maryland, College Park, United States
Fantalgo, LLC, Elkridge, MD, United States

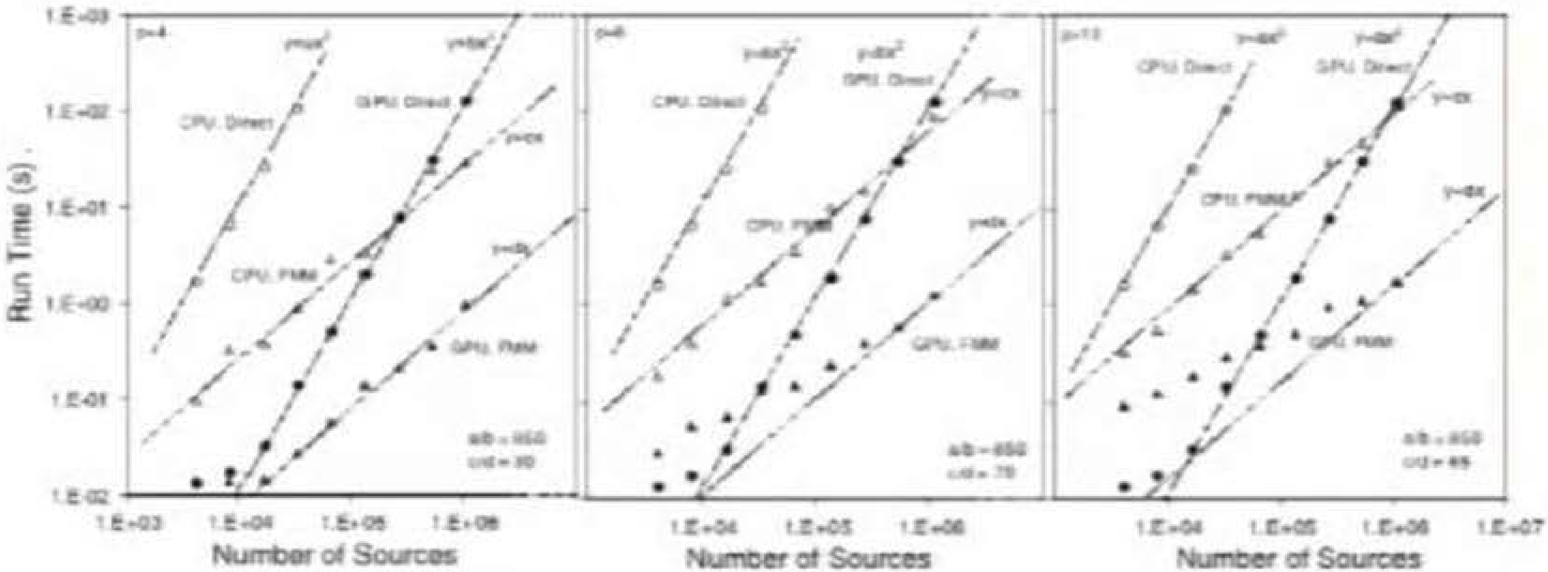
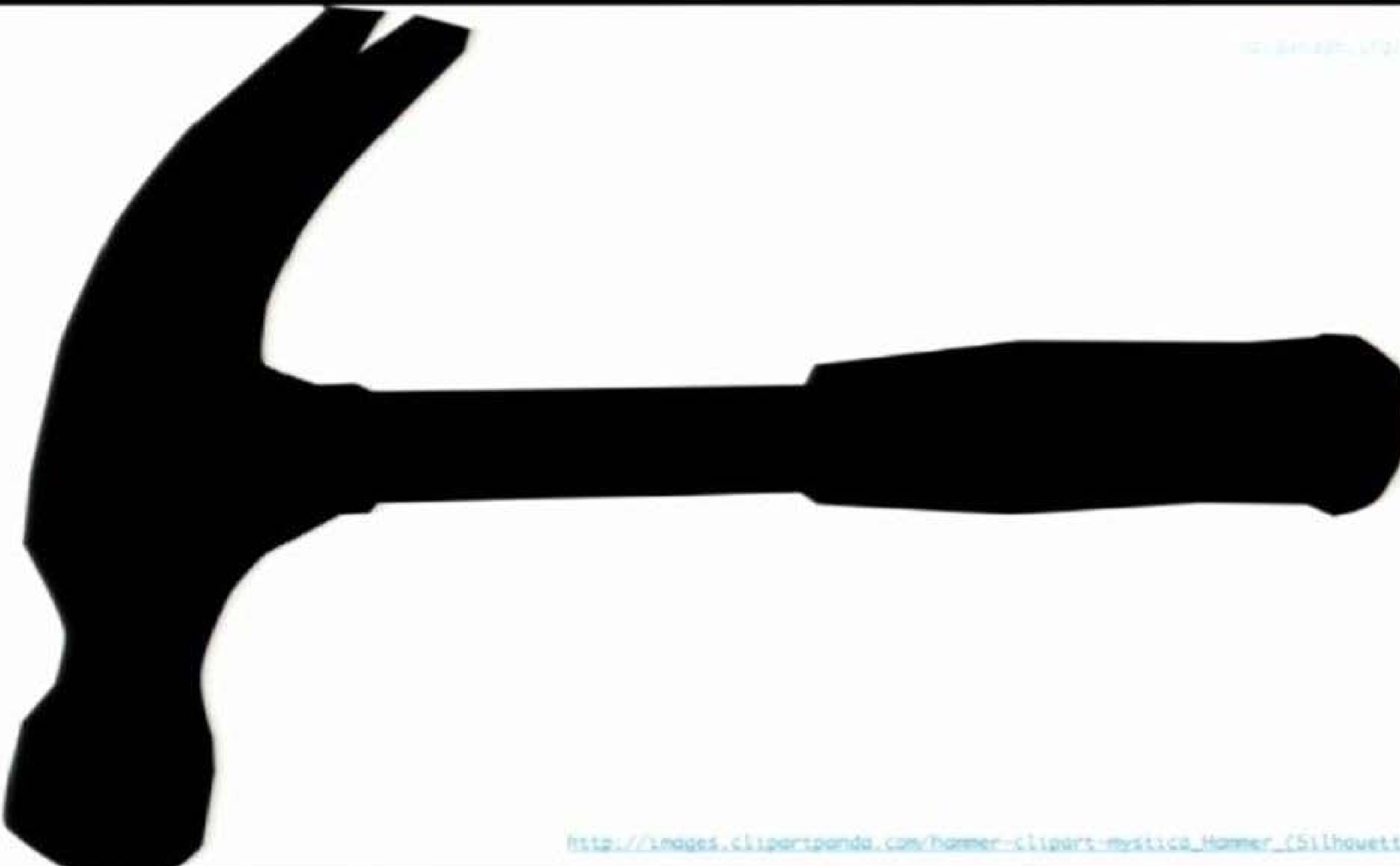


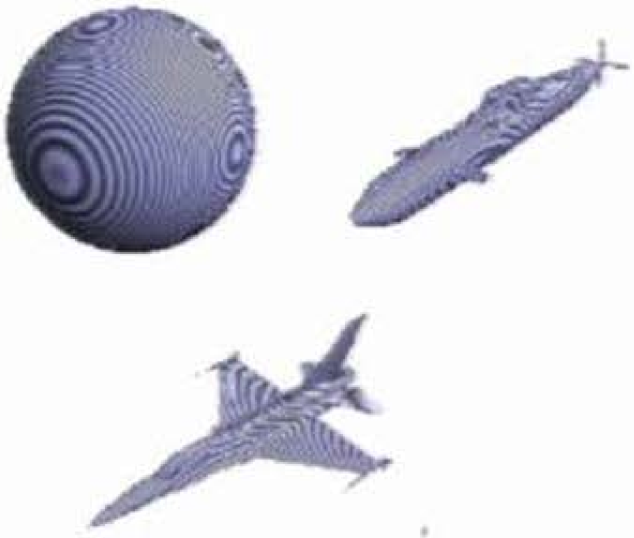
Fig. 11. FMM wall clock time (in seconds) for serial CPU code (one core of 2.67 GHz Intel Core 2 extreme QX is employed) and for GPU (NVIDIA GeForce 8800 GTX) for different truncation numbers p (potential+gradient). Also direct summation timing is displayed for both architectures. No SSE optimizations for the CPU were used.

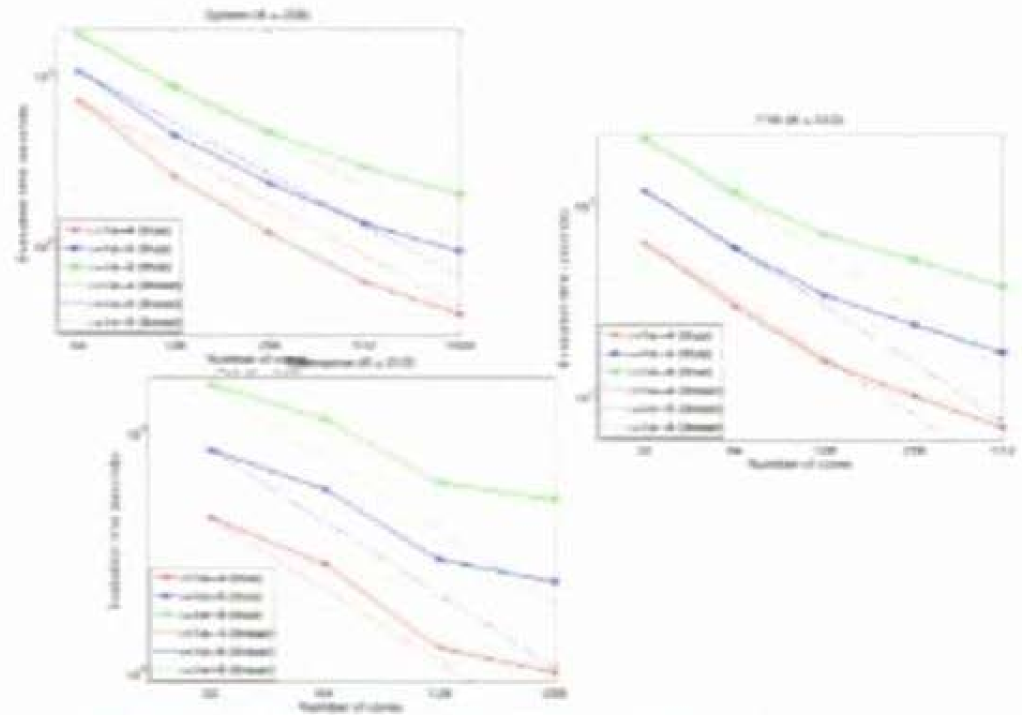


http://images.clipartpanda.com/hammer-clipart-mystica_Hammer_(Silhouette).png

A PARALLEL DIRECTIONAL FAST MULTIPOLE METHOD

AUSTIN R. BENSON*, JACK POULSON†, KENNETH TRAN‡,
BJÖRN ENGQUIST§, AND LEXING YING¶



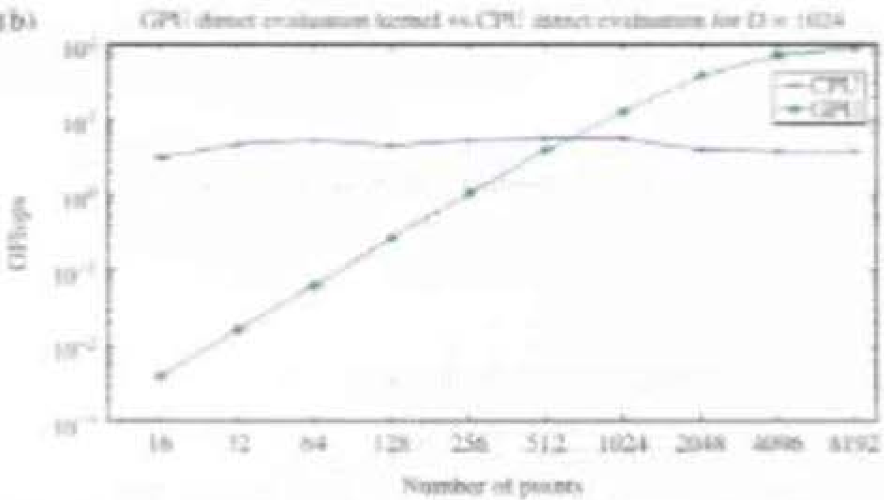


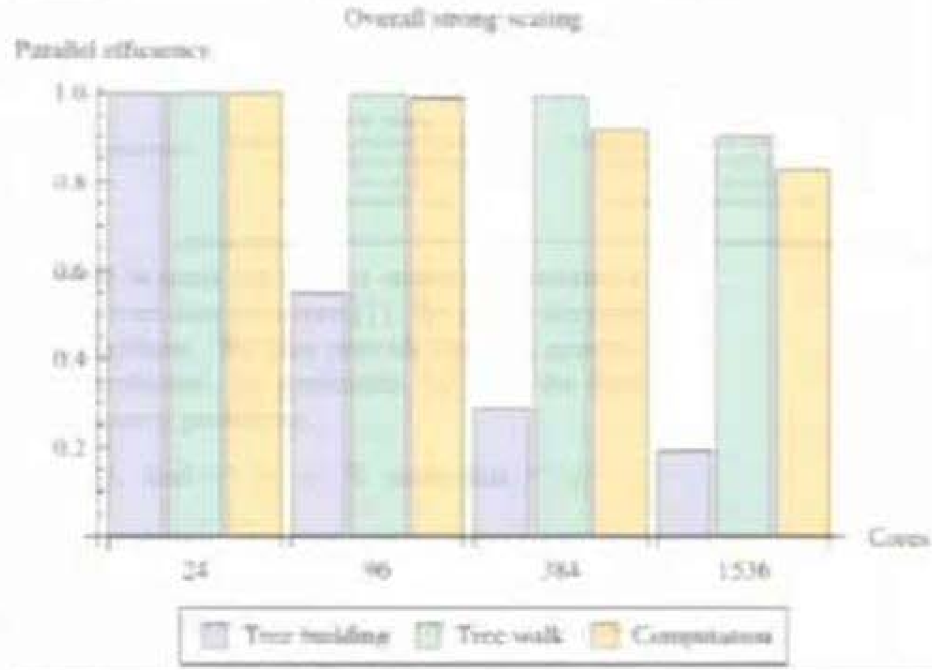
A Distributed Kernel Summation Framework for General-Dimension Machine Learning

Dongryeol Lee[1*], Piyush Sao[2], Richard Vuduc[2] and Alexander G. Gray[2]

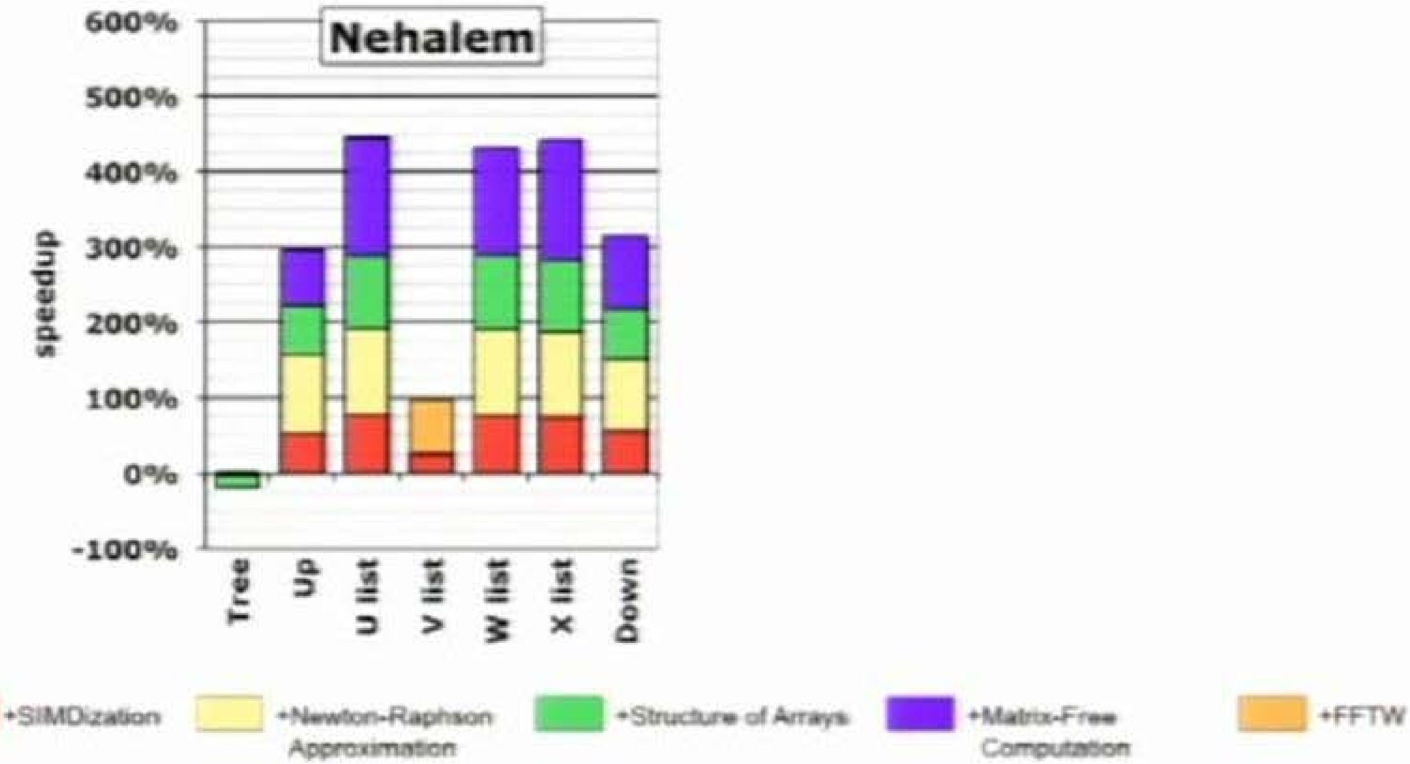
[1]GE Global Research, Schenectady, NY 12309, USA

[2]Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA



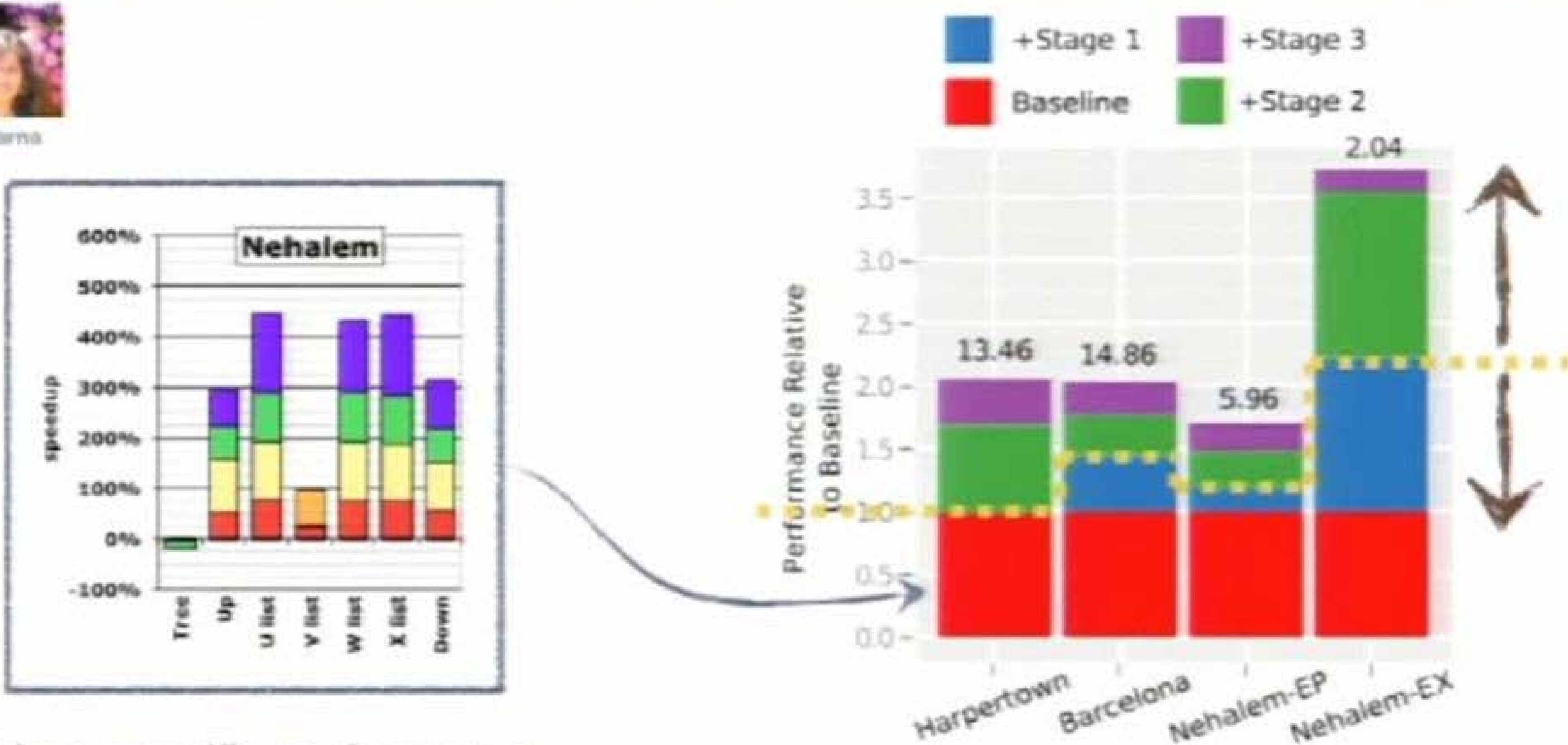


Aparna



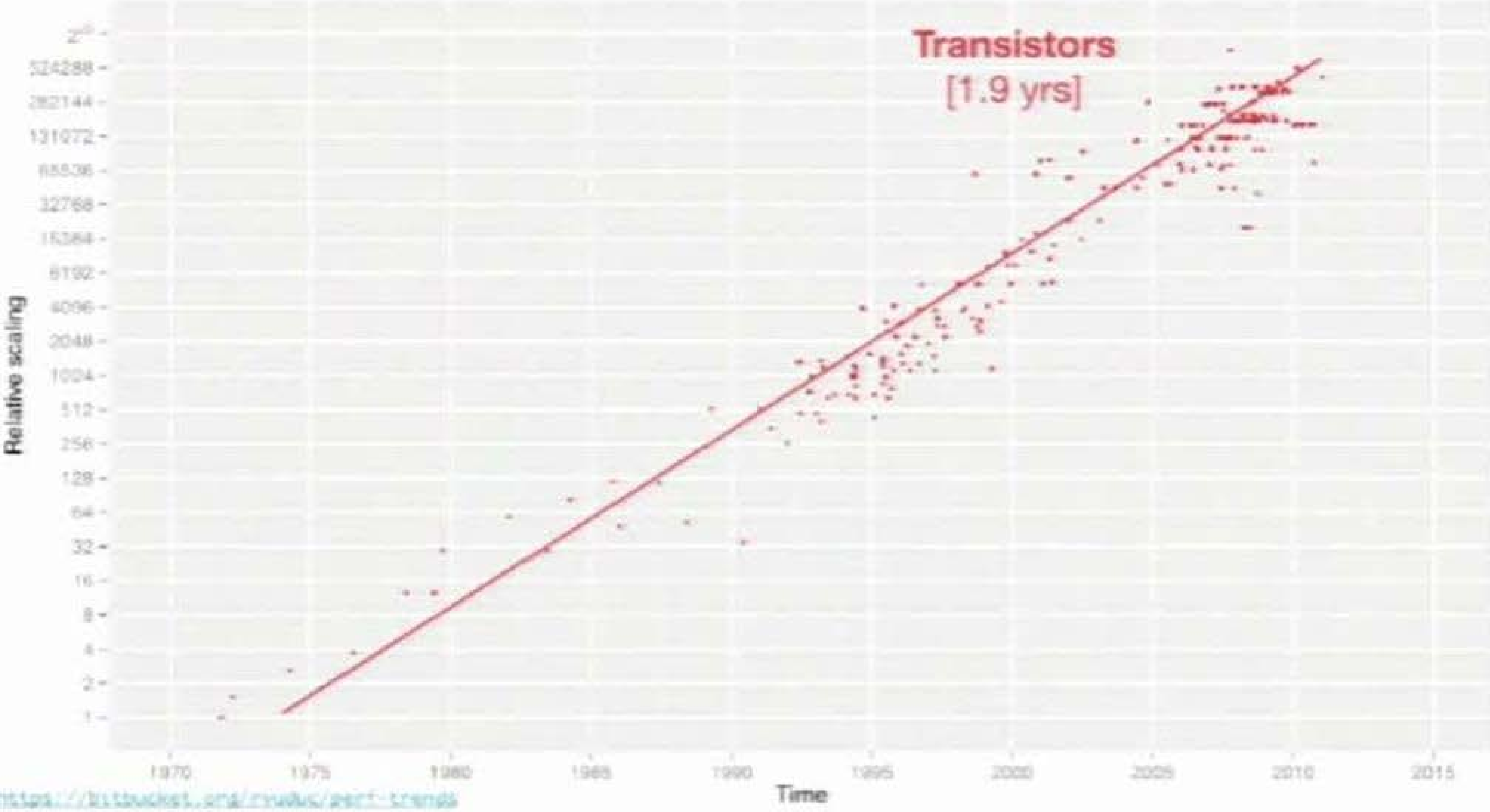
Problem-specific performance engineering

Assume full knowledge of data access patterns, algorithms, and code



Problem-specific performance engineering

Assume full knowledge of data access patterns, algorithms, and code

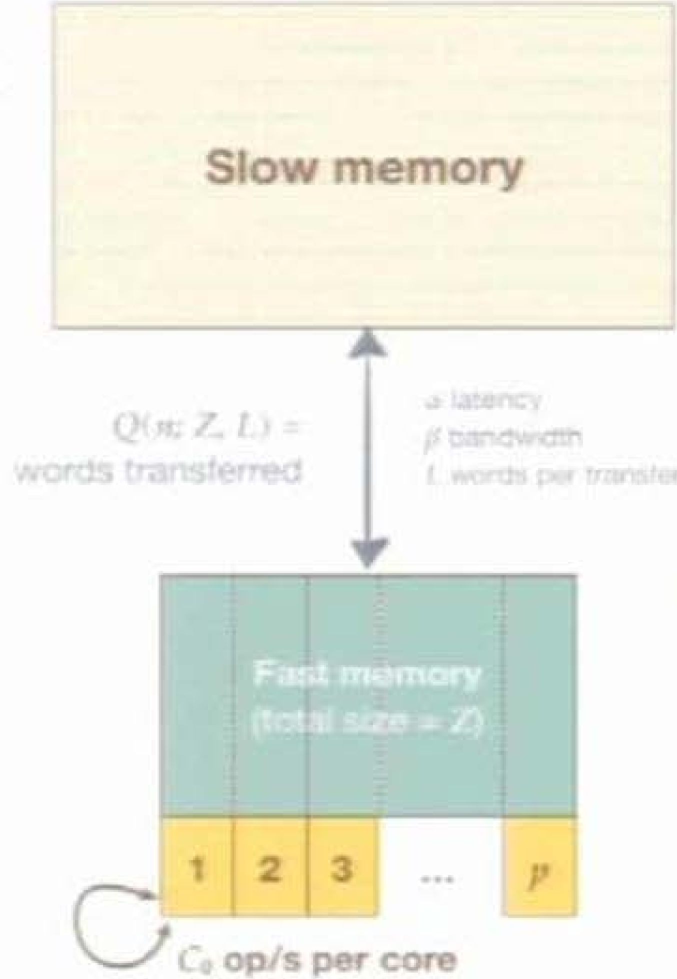


https://bitbucket.org/rvuduc/perf-trends

See Aparna's SPAA'12 brief announcement:

Communication analysis of the fast multipole method

Aparna



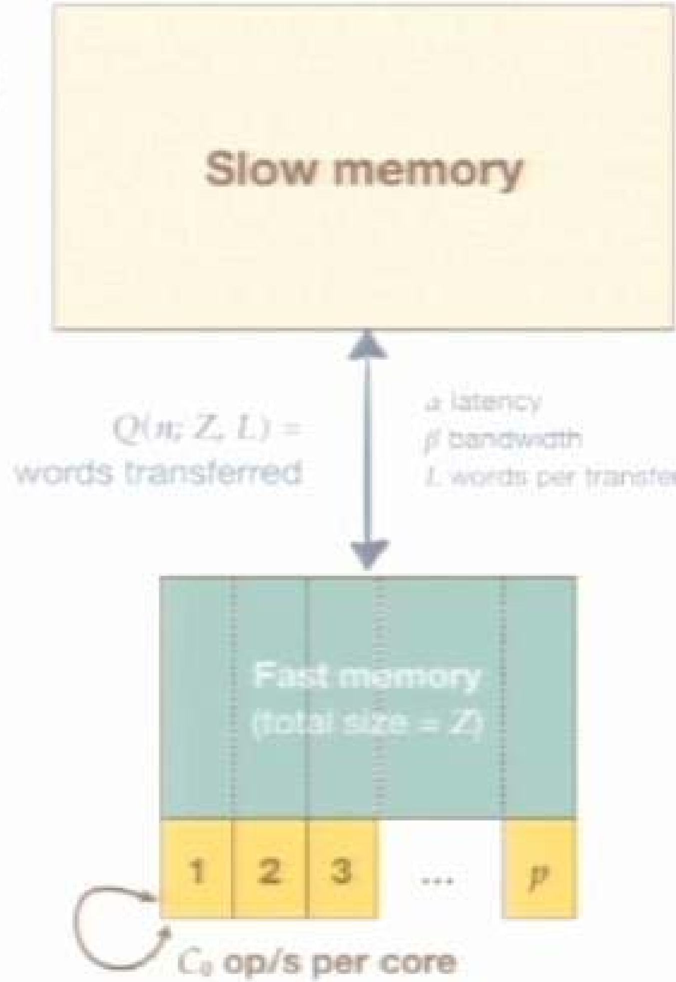
See Aparna's SPAA'12 brief announcement:
Communication analysis of the fast multipole method

Communication penalty:
processor balance
(a.k.a., flop:byte)

~ accuracy

$$T \propto \frac{n\Delta^{\frac{3}{2}}}{p \cdot C_0}\left(1 + (\text{const.})\frac{p \cdot C_0}{\beta}\right)$$

Min. time
(ops only)



A summary?

Despite tremendous progress, there is a hidden cost, caused by the lack of tools and *simple* techniques to make fast code persist.