

Petaflops Algorithms: Summary and Recommendations

David H. Bailey

NERSC

Lawrence Berkeley Laboratory

dhbailey@lbl.gov



The Concurrency Challenge

- **Little's Law: Concurrency = memory latency x total memory bandwidth.**
- **Another reckoning: Concurrency = number of nodes x memory latency hiding per node.**
- **Result: 10^7 to 10^8 concurrent threads must be found in virtually every step of algorithms on petaflops systems.**
- **Do the algorithms to be used on future systems possess these extremely high levels of concurrency, even on large problems?**



Algorithms and Communication

- **Hierarchical algorithms.**
 - » Do there exist variants of key algorithms that possess a natural hierarchical structure, well matched to future system designs?
- **Latency tolerant algorithms.**
 - » Do there exist variants of key algorithms that permit very coarse-grained (large block) communication implementations?
- **Bandwidth-friendly algorithms.**
 - » Do there exist variants of key algorithms that minimize the total amount of data communicated?
- **To what extent can we trade latency for bandwidth as necessary to optimize an algorithm for a given system architecture?**



Numerical Issues

- **Numerical accuracy.**
 - » Can we utilize algorithms that are less numerically accurate, but which are better suited for parallel systems?
 - » To what extent can we remedy these errors by the use of a refinement step done with extra-precision arithmetic?
- **Numerical scalability.**
 - » For a wide class of 3-D physical simulation algorithms, computational work scales as n^4 , while memory scales as n^3 .
 - » Can we break this link between grid size and time step size?
 - » What can be done for algorithms, such as those that arise in computational chemistry, that have very unfavorable computational complexity (n^5 , n^7 , $n!$, etc.)?



Unstructured Methods

- **What are the communications requirements of dynamic mesh methods?**
- **Can advanced techniques reduce the communications requirements?**
- **Can algorithms that require dynamic load balancing be efficiently implemented on clusters and other high-latency systems?**
- **Can grid generation and grid partitioning algorithms be efficiently implemented on highly parallel systems?**



Performance Analysis and Modeling

- **The performance characteristics of key algorithms are not well understood, except in a few isolated instances.**
- **Why is the sustained performance of most algorithms so poor on existing systems?**
- **What is the principal performance bottleneck -- single node issues, concurrency limitations, or communication problems?**
- **How can the performance of key algorithms be projected on future high-end systems?**



Recommendations

- **More money.**

