

What should we do next?

Robert W. Numrich

Minnesota Supercomputing Institute
Minneapolis, MN USA
rwn@msi.umn.edu

June 4, 2008

Back to the future

- ▶ Serious scientific computing is not cheap.
- ▶ It requires a tightly-integrated system with global load/store instructions and fast synchronization primitives.
 - ▶ semaphores, test-n-set
- ▶ Object-oriented languages solve the data distribution problem (Fortran 2003, C++, Java, C#).
 - ▶ Abstract distributed data structures
 - ▶ Abstract methods
 - ▶ Concrete, extensible data structures and methods for specific architectures
- ▶ Simple extensions to existing languages provide sufficient expressivity (CAF, UPC, Titanium).

Forward to the future

- ▶ On a loosely integrated system, without a global address space, few programming models work well.
 - ▶ How easily can you substitute a different communication model into the MPI-API?
 - ▶ Suppose the new machines require interleaving computation with communication. Can you disentangle your application from the MPI-API?
- ▶ On a tightly integrated system with a global address space, many programming models work well.
 - ▶ Liberates imagination and creativity.
 - ▶ Expose the memory hierarchy and the synchronization primitives to the programmer.
 - ▶ Computational scientists do understand their problems.
- ▶ A compiler that yields 5% of peak on processor one, kills you before you ever get to processor two.
- ▶ A compiler that yields 50% of peak, allows you to concentrate on your problem instead of the arcane features of the chip.

My wish list

- ▶ A strategic national initiative needs to **invest** in high-performance scientific computing.
- ▶ Hardware vendors need to build high-performance, but balanced, hardware more cost-effectively.
 - ▶ Give me something like a Cray-1 on a chip.
 - ▶ flops are not the problem; memory is, and always has been, the problem (von Neumann, collected works, vol. 5? ca. 1946).
- ▶ Compiler technology needs to generate high-performance code for single processors.
- ▶ Compiler technology needs to be more nimble to keep up with new processor designs and to incorporate new programming models in a timely manner.

Final thoughts

- ▶ Don't trust anyone over 30!
- ▶ "Question everything, including everything I just said."
Emerson, *The American Scholar*, 1837.
Lib^{NewNew} F--
- ▶ The 21st Century is yours. Figure it out.