Trends in parallel computing and their implications for extreme-scale parallel coupled cluster

Workshop on Parallelization of Coupled Cluster Methods

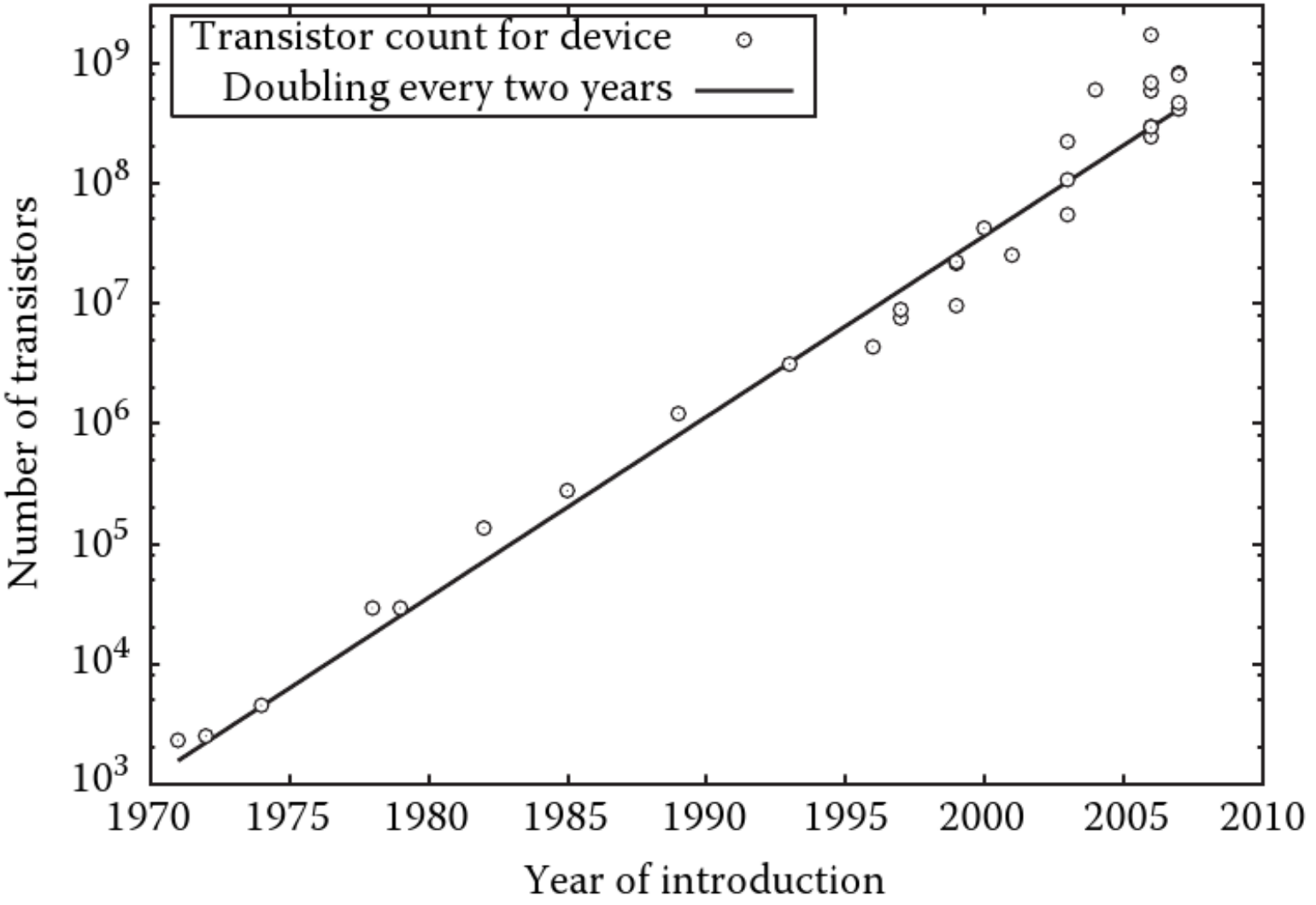
February 23 to 24, 2008
St. Simons Island, GA

Curtis Janssen

SAND Number: 2008-1166C
Unlimited Release

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

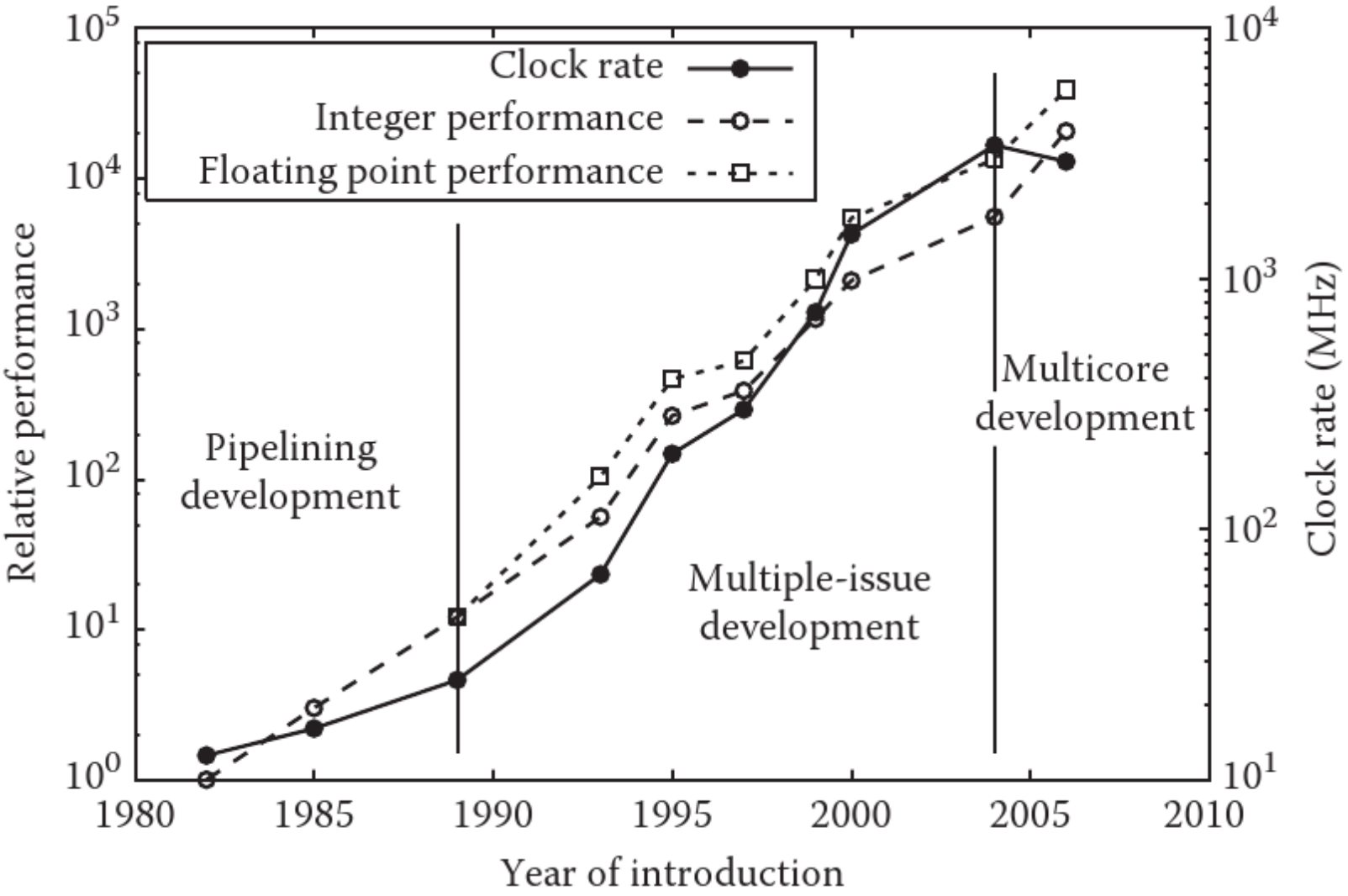
Cost of semiconductor production continues to follow Moore's law



From: C. L. Janssen, I. B. Nielsen, *Parallel Computing in Quantum Chemistry*, CRC Press, May 2008.

Sandia National Laboratories

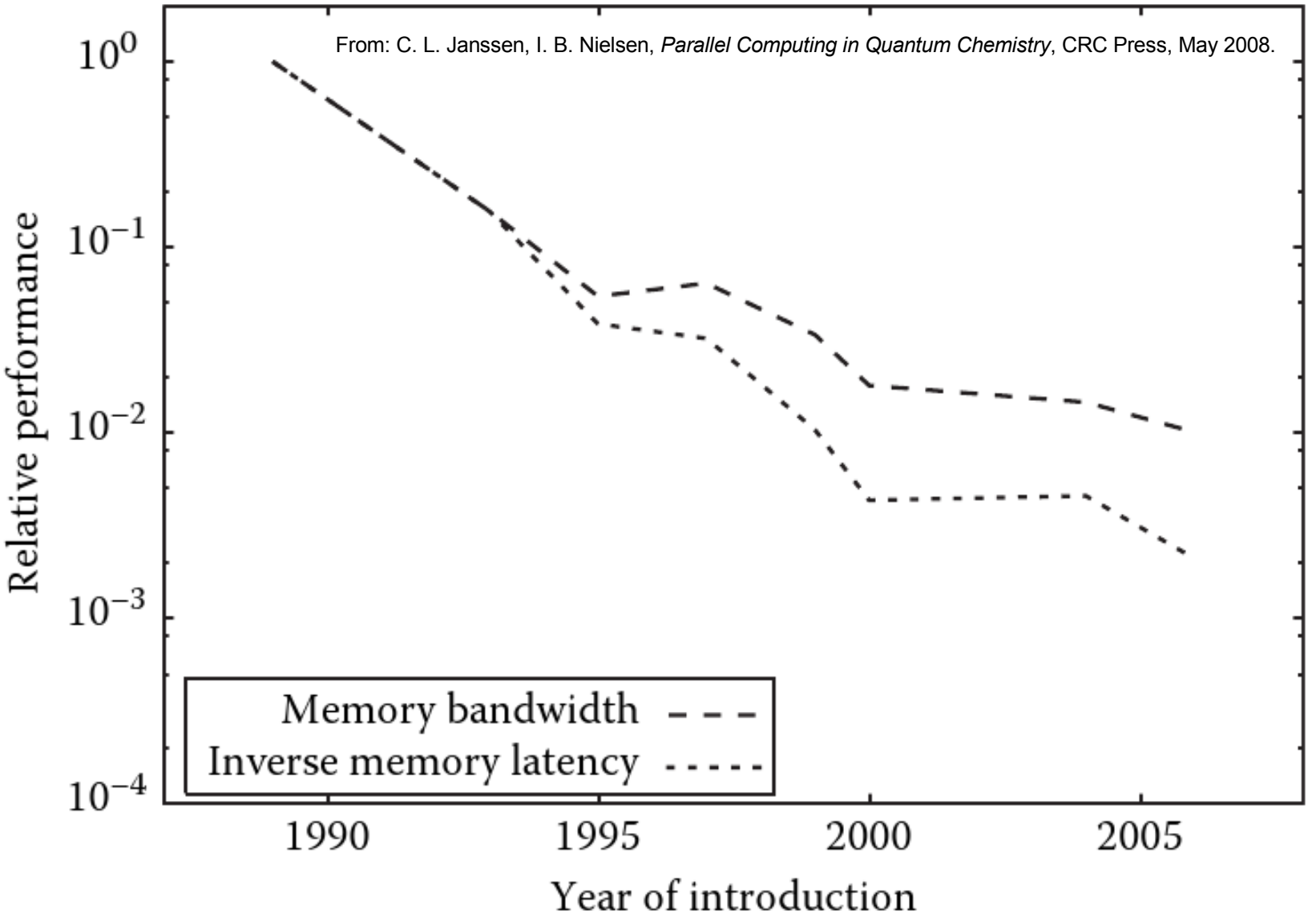
Moore's law has affected performance in a variety of ways



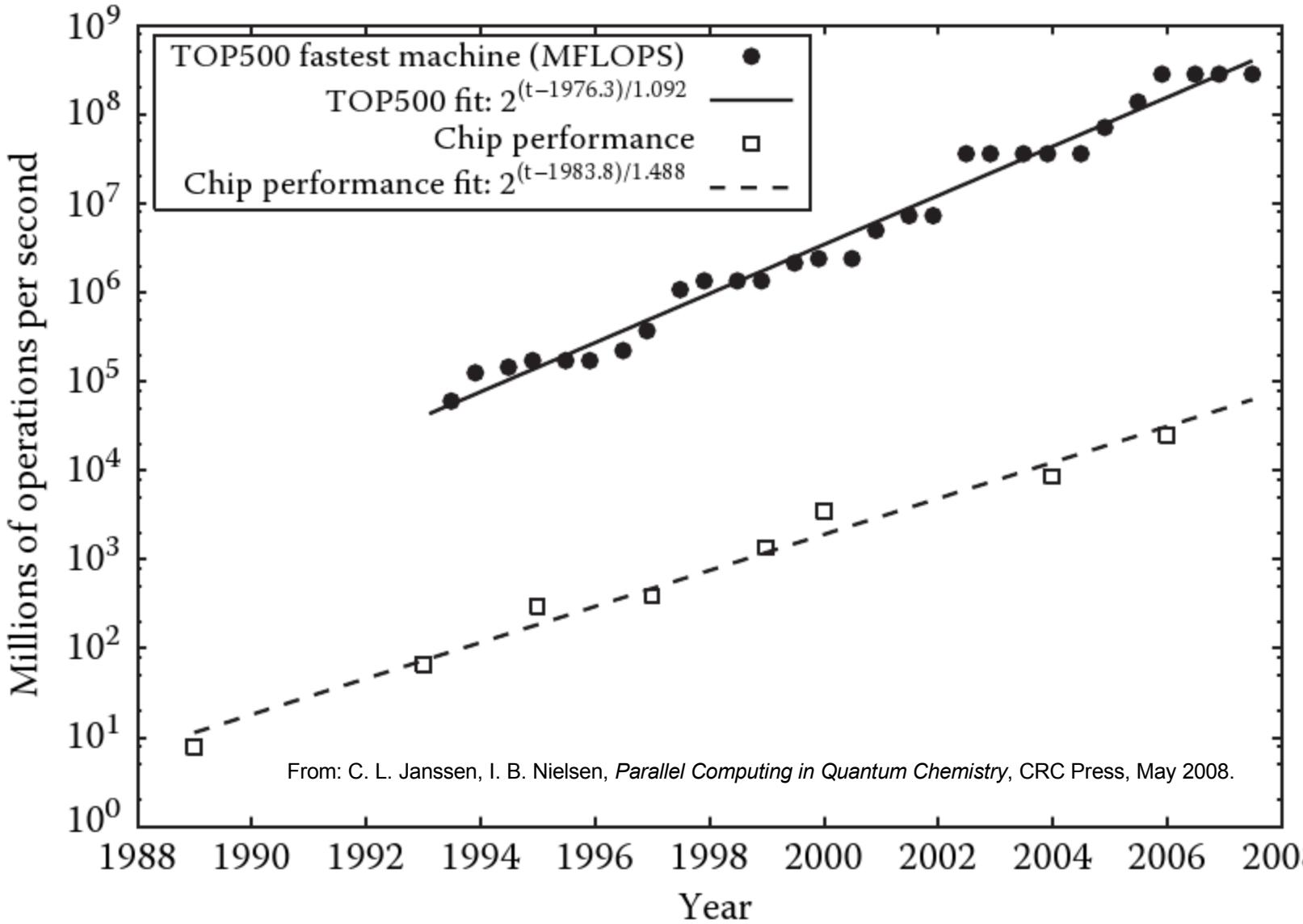
From: C. L. Janssen, I. B. Nielsen, *Parallel Computing in Quantum Chemistry*, CRC Press, May 2008.

Sandia National Laboratories

These improvements in speed are not matched by latency and bandwidth



Parallel machines increase performance by faster chips and more chips



From: C. L. Janssen, I. B. Nielsen, *Parallel Computing in Quantum Chemistry*, CRC Press, May 2008.

Sandia National Laboratories

Net result: future parallel environments very different from today's.

- Multiple levels of memory (both local and remote)
  - Latency to speed gap will continue to widen. (Stacked chips might help us, though.)
- More components, and failure rate proportional to number of components
  - Can we assume future machines are 100% reliable?
- Even if crashes avoided what about “brown outs”?
  - Thermal throttling, cores going offline, ECC recovery, memory banks going offline.
  - Nodes might be heterogeneous from a performance perspective

Efficient utilization of future machines will be hard.

We also must consider how the application will be transformed.

- We all agree that high accuracy quantum methods are important, but how best employed at such scale?
  - Canonical CC methods?
  - Reduced scaling CC methods?
  - Periodic CC methods?
  - More robust CC methods?
  - All of the above.
- With petaflop machine, 5 yr lifetime, several MW power, several FTEs support: cost to run a one week job on entire machine > $1,000,000.
- Obligated to provide the most impactful science possible on such large and expensive machines.

Reduced scaling methods present tremendous parallelization challenges

- **Canonical MP2, parallelizing only the *O(N^5)* step:**

$$t_{MP2}(N,p) \approx A\frac{N^5}{p} + BN^4 \qquad S_{max,MP2} = \frac{AN+B}{B}$$

- **Local MP2, parallelizing most of the *O(N)* steps:**

$$t_{LMP2}(N,p) \approx C\frac{N}{p} + DN \qquad S_{max,LMP2} \approx \frac{C+D}{D}$$

In a local method, data is more irregular, making load balancing more difficult—this is where current widely practiced programming models break down.

Current programming models do not allow human effort to scale up

- MPI+Remote Get/Put/Accum has worked up to now
- MPI+Remote Get/Put/Accum+threads will help us scale up — but there are problems
  - MPI is difficult, threads are difficult, hybrid is difficult[2]
  - Results in a fragile environment, expensive to develop, debug, and obtain portable performance
- Memory hierarchy is deep, but imperative programming languages encourage random access
  - Need to think in terms of new models akin to dataflow
  - Combination of better general runtimes plus domain specific tools are needed

Key Points

- Getting good scaling to > 100,000 will be hard.
- Current methods must expand and adapt to the types of problems that will be solved at that scale.
- Must reexamine current programming models to find best way to allow human scalability—entire software life cycle must be considered.