Why hybrid? Why now?

September 13, 2009

Bronis R. de Supinski

Center for Applied Scientific Computing

Science & Technology Principal Directorate - Computation Directorate

Is the sky falling now?

- ASCI Blue-Pacific (circa 1998)
  - 1464 Nodes
  - 4-way SMP nodes (5856 cores total)
  - Limit of 512 MPI tasks
- Hybrid model needed?
  - Yes: For full machine runs
  - No: Most jobs used under 512 tasks
  - Limited performance of hybrid model

- ASC Sequoia (circa 2012)
  - Over 100,000 nodes
  - More than 1.5 million cores, 6 million threads
  - MPI required to support full system jobs
- Hybrid model needed?
  - Yes: Limited MPI efficiency for full machine runs
  - No: Most jobs will not span entire system
- What is different this time?

Circumstances today are different

- MPI Limitations are different
  - Lower relative messaging rates
  - Core counts make MPI memory use too high
- Limitations are not just from MPI
  - Less memory and memory bandwidth per core
  - Overall required concurrency levels are much higher
    - Likely to exceed concurrency of existing parallelization
    - Multiplicative advantages of multiple strategies

- Architectural changes make hybrid models attractive
  - Multicore nodes will bring threading support
    - Support for fast thread creation and synchronization
    - More innovative features, like transactional memory, likely
  - GPUs and other accelerators may proliferate
    - Likely on-chip integration of GPU & general purpose CPU
    - Does any one model apply to systems with accelerators?

Are hybrid models evil?

Why is this:

better than this?

Are hybrid models evil? NO!

- Hybrid models allow separation of concerns
  - Can largely isolate parallelization at multiple levels
  - Target each level of hardware
- No magic compiler is coming to save the day
  - It never has and it never will

A brief foray into which models to hybridize

- MPI must be included
  - Too much invested in it already
  - Suits distributed problem well (most people are wrong)
- Other models to include?
  - Must acknowledge threading; I favor OpenMP
  - What about accelerators? Can the compiler help?