SAND2010-5501P

Programming and Run-Time Models for Heavily Threaded Systems

Hardware Panel

Scott Hemmert
Sandia National Laboratories
Scalable Computer Architectures
kshemme@sandia.gov

July 28, 2010

Sandia is a Multiprogram Laboratory Operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy Under Contract DE-ACO4-94AL85000.

Likely Consequences of the Massive Multi-core Future

- Dramatically increasing levels of parallelism to exploit
  - Can be good or bad depending on your programming model
  - Likely to have multiple threads per core (unsure if this is 1's, 10's or 100's)
- No socket wide cache coherency
- More depth to memory hierarchy
  - Near and far memory
- Less memory capacity per compute
- Limiting factor on large scale machines will be data movement
  - I would be wary of programming models that don't allow easy programmer control of locality
  - I would highly recommend a model that forces programmers to think about locality (at least for problem areas with locality)

GPGPUs: A Cautionary Tale

- Historically, HPC has been able to ride the commodity CPU market
  - It has become evident that the commodity CPU market will not get us to exascale
- Naturally, we looked for another commodity market to ride to exascale
  - The only logical choice: GPUs
- Unfortunately, the commodity GPU market will not get us to exascale
  - To be generally useful, GPUs will need to become more flexible
    - Leading to higher overheads and lower efficiency
  - Commodity GPU market will not drive the required resiliency or memory capacity
- Take home message: GPUs will be a useful tool for HPC, but they shouldn't distract us from pursuing other options

What We Can't Give You (and why)

- Efficient networks for 8-byte messages (laws of nature)
- Efficient networks for BSP-style communications (cost/power)
- Same compute/memory capacity ratios of today (cost)
  - Not necessarily true for small systems, but
  - Even if we can afford the capacity we can't provide high capacity at high bandwidth
- Reliability (physics)
  - May be able to provide some level of resiliency, at the cost of increased power

Prime Areas for Investment

- Memory system
  - More channels
  - More bandwidth
  - In-memory operations
- Network
  - Make network match communication semantics
    - Yes, it Is possible to support MPI reasonably at exascale
  - More efficient topologies
  - More efficient links (power saving states)
- On socket memory hierarchy
  - Cache vs. scratchpad
- System level “power exchange”

Hardware Support for Run-Time Systems

- Network hardware support for thread activation
  - Run-time system components must communicate across nodes
  - Message reception in current networks occurs by recognizing change in memory
    - Leads to polling
  - Need hardware mechanism to block/unblock threads on network events
  - Active message model only makes sense with hardware support
    - Waiting until there's nothing to do to notice incoming messages is bad
- More advanced network functions (eureka, dynamic hierarchy)
- More sophisticated mode switch / protection hardware
- Hardware performance information
  - Dynamic resource management decisions will need performance info
  - Current performance counters only capture a subset of what is needed
- Thread scheduling
  - Hardware support for efficient scheduling
  - Must be flexible (programmable?)
  - Should allow for operating on groups of threads