Performance Prediction and Simulation

Nathan Binkert

How Do We Model Large Networks?

- Large: networks with 25M to 100M nodes
- Parallel simulation necessary!
- Software: SMP systems
  - On a 16 core system: 1.6M nodes per core
  - On the order of 1KB/node
  - Parallelism from multiple experiments
  - Too many compute-hours for a single experiment

How Do We Model Large Networks?

- Software: cluster systems
  - 10,000s of nodes per core
  - MB/node
  - Interconnect latency?
  - May be required just for memory capacity
- Hardware: FPGA
  - Realistically requires software too
  - Potentially much faster
  - Hard to model buffers for millions of nodes
  - Too many man-hours

How Do We Model Large Networks?

- Model inter-chip network separately
  - Impact on workloads and accuracy
- Model smaller networks faithfully
  - Workloads are fundamentally not the same
- Use the smaller, more accurate network to help tune the larger, less accurate one.

- Do we need to model the cores in detail to understand the network?

Main Challenges

- Workloads!
  - Are synthetic workloads good enough?
  - If not, what do we run?
  - How do we even model it?
    - Not enough RAM/compute
- Validation
  - How do we know we're getting the right answer?
  - Gut feeling may not be sufficient as systems get more complicated.
  - Need to use sound programming constructs
    - Force models to behave like real things to prevent bogus results.
    - E.g. no magic access to variables in other elements

Promising Technologies

- Multicore is both a blessing and a curse
  - Simulation gets more cores
  - Need to simulate more cores
    - Do we really need to simulate the cores?
  - Architectural simulation needs advances in parallel programming just like everyone else.
- Low power techniques reduce simulation time
  - Clock gating (don't simulate)
  - Smaller buffers (faster data structures)
- Hybrid software/FPGA simulation

Will complexity doom us to build what we can model/predict?

- We can only build what we can imagine
- We only build what we can model

- But, start with very crude models and refine them.
  - Excel is often good enough to get initial numbers on a wacky idea
  - Higher fidelity is nice, but not required