Large-scale systems subgroup out-brief

Sarah Michalak (LANL), discussion leader
Dennis Abts (Google), Nathan DeBardeleben (LANL), Greg Bronevetsky (LLNL), John Daly (DoD), Armando Fox (Berkeley), Jon Stearley (SNL), David Walker (Princeton)
Guest stars: Ravi Iyer (UIUC), Will Jones (Coastal Carolina Univ.)

Disclaimer

- These are not (yet) in “challenge question” form. They represent key research agenda foci.

Key points:

1. Large shared-memory apps are dead
2. App-level abstractions must encapsulate *operations* as well as *functionality*
3. Dilation factors vs. “mean time to X” should be metrics of merit for challenges

Goal: exascale reliability—2 paths

1. Reinvent the world
   - today's HW, OS, apps, ... weren't designed for extreme scale; they must be replaced
2. Endgame of "Google approach"
   - conceal/contain HW failures through software architecture, telemetry analysis, machine learning
   - endgame: catastrophe (100's racks or whole warehouse failure) doesn't stop computation

- Approaches will yield complementary insights, pursue in parallel

1. Abstractions for operations as well as functionality

- Value in MapReduce paper was not the functional abstraction, but *operational* one
  - M/R infrastructure handles many common failures, reschedules failed work, tries to find spatial locality, ...
  - sophisticated tuning, relies heavily on analyzing telemetry from multiple layers
  - yet mostly invisible to *users* of M/R—Google view is must scale # of programmers as well as HW size
- What other subsystems/abstractions look at?

2. Large-shared-mem apps are dead

- Popular abstraction: everyone reads/writes an arbitrarily large shared data structure
- Datacenter-scale commercial apps discovered they can't afford this abstraction
  - MTTFA, energy use, overengineered redundancy, inability to hide NUMA performance gap
- Their approach: rearchitect apps to *what is buildable* (shared-nothing clusters)
  - Stateless protocols, locality-awareness, different storage subsystems optimized for different tasks
  - Heavy use of cross-layer debugging/monitoring to find bottlenecks, performance failures, errors

Role for new algorithms research

- *New algorithms research to reformulate some HPC problems*
- Example: instead of 1 NLP model, build N models, sync'd every T. Prove bounds on model drift as function of T.
-
- Goal: algorithms should better fit the lower spatial & temporal locality of clusters

3. Dilation factors vs. MTTFA

- Endgame of “Google approach”: application never fatal-aborts (but see next slide)
- But resuming from coarse-grained checkpoint increases completion time, energy use
- *Time dilation & energy dilation* become metrics of merit for benchmark (subject to calculational correctness)
- For apps that tolerate variable precision answer (e.g. convergence, confidence interval, probabilistic bound), also *precision dilation*
- Product of all 3 is target for improvement

Cross-datacenter reliability

- Last major obstacle to “nonstop” for commercial apps
  - lightning or regional power failures still stop app
- significant *energy savings* from eliminating heavy power redundancy in datacenter
  - capital cost =~20% of datacenter cost (for ~50K-node DC; Hamilton et al. 2008)
  - operating overhead =~ 10-12% of ingress power
  - (compare: up to 40% power spent in recovery/fault management at hardware level)