The Case of the Missing Supercomputer Performance

Achieving Optimal Performance on the 8,192 Processors of ASCI Q

Fabrizio Petrini, Darren J. Kerbyson, and Scott Pakin

Proc. ACM/IEEE Conf. Supercomputing (SC), Phoenix, AZ, USA, November 2003.

Key Ideas (2003)

- "Waterfall" method of optimization
  Model, then measure, analyze, modify, repeat
- Use "microbenchmarking" to identify performance trends.
- Test with the smallest possible synthetic benchmarks.
- Combine synthetic benchmarks to cover system behavior.
  Model the universe of behavior, and close in on the (possibly complex) cause.
- Time-intensive, many false leads.

Yeah, so?

- Performance decayed only at 128+ processors
  Most of us weren't working on 128+ processors in 2003.
- Performance decayed *catastrophically*
  At 256 cores, 3 processors were faster than 4. At 512, 2 were faster.
  4 processes per node was a magic threshold for catastrophe.
- Shared resource; real-world runs are infrequent
  Affects testing methodology. We can't "throw more machines" at the problem—the model's broken.

Methodology Part 1

- Instruction pointer poll-based sampling
- Repeat in different configurations
- Look for performance shifts, rather than pure hot spots

  The `allreduce` collector is identified.

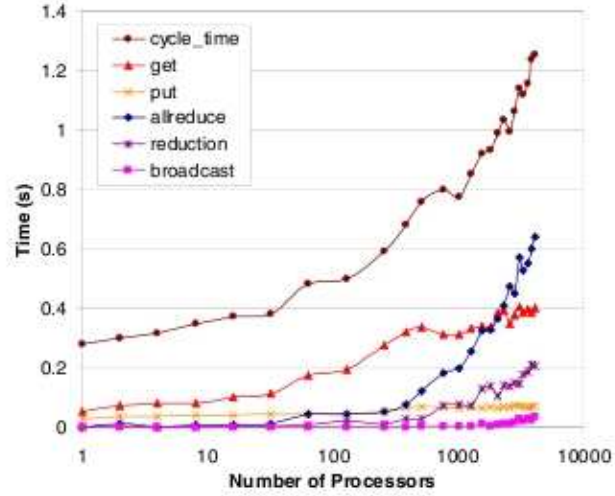

Figure 5: Profile of SAGE's cycle time

Figure 1: Scale fail

Methodology Part 2

- If it's not our code, it's someone else's.
- Model the gaps and match them up with other processes.
- {F,L,E,P} (ν, $\overline{length}$, X, nodes)
- "Accurate enough to closely model. . ."
  Gotta hand it to these pioneers. Life was tough in ought-and-three.
- `latencytop(8)` demonstration

  ©Arjan van de Ven, Intel Corporation <arjan@linux.intel.com>

Results Part 1

- For a barrier, the kingdom was lost
  Barrier-style synchronization meant any thread's slowdown was a total slowdown.
- More performance is lost to short, frequent noise on multiple nodes than to long, infrequent noise on fewer nodes.

Results Part 2

- Disturbance model integrated into simulator.
- Elimination of easily-culled noise predicted 2.2x improvement.
- Substantial performance loss requires a *resonance* (best result, IMHO).
- Pittsburgh team fudged magic constants to “work out” the same issue.

Critique

- Pg 2 - "The primary challenge is complexity . . . applications of hundreds of thousands of lines of code."
  Maybe you ought work on that before you add another 4,096 nodes.
- Pg 2 - "Nodes run a. . . operating system tuned for workstation workloads, not high-performance computing."
  I suppose that if the Good Lord wanted you to run a lightweight OS, He'd have miracled one onto the node for you.
- Pg 2 - "It ran LINPACK at 68%. Reasonable performance."
  No interest in how architectural changes feed back into algorithmic changes.
  No thought of stepping back and redesigning the model.
  Where's the one they built to throw away?
- Pg 10 - "obvious solution is to remove any type of noise"
  Then we give everyone their own pet rhinoceros!