Speed, Low Latency and Parallel Programming

Jeffrey M. Birnbaum

60East Technologies, LLC

The Big Three

- MultiCore and ManyCore
  - Think Parallel is a skill and mindset
  - Lock/Wait Free codes require understanding
- Flash in the server
  - Super fast storage
  - Huge cheap memory footprint (TB's)
- RDMA Over Ethernet
  - Low Latency
  - High Throughput
  - Only way to keep up with data production rates

The Choices

- X86 CPUs
  - Programming skill is critical
  - Software is easier to maintain and evolve
- FPGA
  - Low Jitter
  - Hard to maintain and evolve
- GPU
  - Tools get better every day
  - Memory movement is still an issue
- MIC
  - Software leverage and lots of cores
  - Too new to know

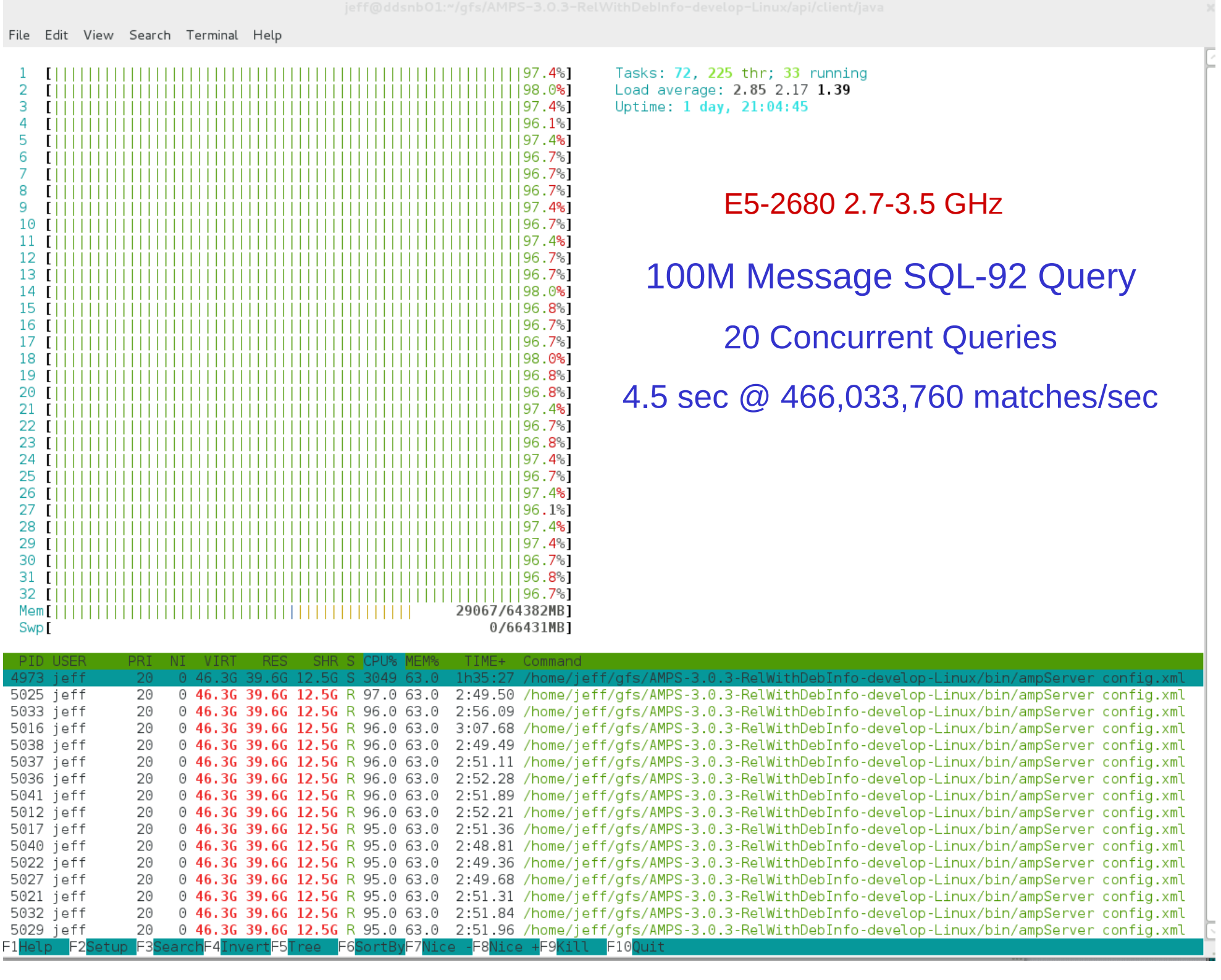
Comparison of 1S and 2S Sandy Bridge
100M Message Parallel In-Memory SQL-92 Query

I7-3960X – 12 Cores @ 3.3 GHz with Turbo to 3.9 GHz

Operation	Elapsed Time	Matches/Sec/Core/GHz	Matches/Sec
1 Query	0.91 secs	2,309,842	115,228,131
20 Concurrent Queries	11.5 secs	8,147,440	182,361,040
Complete reindex (parse,write,match)	5.5 secs	194,830	19,065,018

E5-2680 – 32 Cores @ 2.7 with Turbo to 3.5 GHz

Operation	Elapsed Time	Matches/Sec/Core/GHz	Matches/Sec
1 Query	0.42 secs	2,229,115	249,660,952
20 Concurrent Queries	4.5 secs	8,141,100	466,033,760
Complete Reindex (parse,write,match)	2.3 secs	208,050	45,590,260



E5-2680 2.7-3.5 GHz

100M Message SQL-92 Query

20 Concurrent Queries

4.5 sec @ 466,033,760 matches/sec

The Takeaway

- Exploit parallelism in any way possible
- Pick the solution that works best for you whether it be CPUs, FPGAs or GPUs
- The trade offs depend on what you are trying to achieve, low jitter vs ease of change
- Design and plan for even more parallelism than you have today, i.e. forward scaling
- Invest in experimentation and don't rely on past knowledge