ECE 574 – Cluster Computing
Lecture 14

Vince Weaver

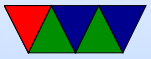
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

12 March 2019

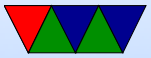
Announcements

- HW#7 will be posted.
- HW#6 was due

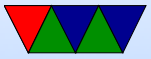


ICS Reviews

- Why I'm behind on grading
- 2019 International Conference on Supercomputing (Phoenix, AZ)
- Note this is *not* Supercomputing (ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis) (deadline soon)
- Also not ISC (International Supercomputing Conference), Europe

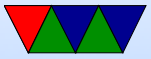


- On program committee. Give rundown on what that means. Rhode Island PC Meeting
- Interesting topics. One addressing skid issues we had with earlier hw. Lots on slurm-style scheduling. Processor-variability. Auto-vectoring code. Hard-drive reliability.

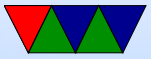


MPI Debugging (HW#6) notes

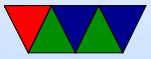
- MPI is *not* shared memory
- Picture having 4 nodes, each running a copy of your program *without* MPI.
  Also picture the various MPI routines as a network socket (or web browser query).
  Things initialized the same in all will have same values, no need to initialize.
  Things initialized in only one node will need to be somehow broadcast for the values to be the same in all.



- Problems debugging memory issues.
  Valgrind should work, but Debian compiles MPI with checkpoint support which breaks Valgrind :(
  Mpirun supposed to have -gdb option, doesn't seem to work.
- What does work is `mpiexec -n num xterm -e gdb ./your_app` but this depends on you running X11 plus logging into Haswell-EP with X forwarding (-Y) enabled
- The bug most people hit is improper bounds, leading to segfault. You can debug that with printfs of your bounds
- MPI does give useful error messages sometimes

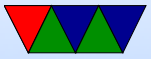


- Some of the problem is malloc/calloc

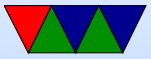


Other MPI Notes

- `MPI_Gather( sendarray, 100, MPI_INT, rbuf, 100, MPI_INT, root, comm);`
  rbuf ignored on all but root
- All collective ops are blocking by default, so you don't need an implicit barrier
- MPI_Gather(), same as if each process did an MPI_Send() and the root note did in a loop MPI_Receive() incrementing the offset.

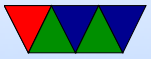


- `MPI_Gather()` aliasing
  cannot gather into same pointer, will get an aliasing error
  Can use `MPI_IN_PLACE` instead of the send buffer on rank0.
  Why is this an error? Partly because you cannot alias in Fortran. Just avoids potential memory copying errors.
  What happens if your gathers overlap?
- Can you handle non-even buffer sizes with MPI_Gather?
  No. Two options.
  - One, just handle in one of other threads (either master



or send/receive from other)

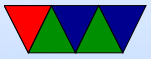
- Two, use `MPI_Gatherv()` where you specify the displacement and sizes of what you want to gather



Reliability in HPC

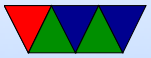
Good reference is a class I took a long time ago, CS717 at Cornell:

`http://greg.bronevetsky.com/CS717FA2004/Lectures.html`

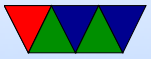


Sources of Failure

- Software Failure
  - Buggy Code
  - System misconfiguration
- Hardware Failure
  - Loose wires
  - Tin whiskers (lead-free solder)
  - Lightning strike
  - Radiation
  - Moving parts wear out

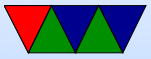


- Malicious Failure
  - Hacker attack



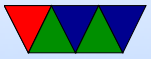
Types of fault

- Permanent Faults – same input will always result in same failure
- Transient Faults – go away, temporary, harder to figure out



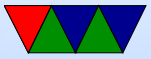
What do we do on faults?

- Detect and recover?
- Just fail?
- Can we still get correct results?



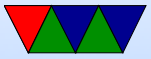
Metrics

- MTBF – mean time before failure
- FIT (failure in Time)
  One failure in billion hours. 1000 years MTBF is 114FIT.
  Zero error rate is 0FIT but infinite MTBF Designers just FIT because additive.
- Nines. Five nines 99.999% uptime (5.25 minutes of downtime a year)
  Four nines, 52 minutes. Six nines 31 seconds.
- Bathtub curve

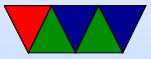


Architectural Vulnerability factor

- Some bit flips matter less
- (branch predictor) others more (caches) some even more (PC)
- Parts of memory that have dead code, unused values

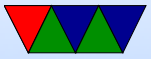


Things you can do Hardware

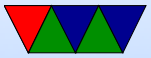


Hardware Replication

- Lock step – Have multiple machines / threads running same code in lock-step Check to see if results match. If not match, problem. If replicated a lot, vote, and say most correct is right result.
- RAID – (redundant array of inexpensive disks)
- Memory checksums – caches, busses
- Power conditioning, surge protection, backup generators, UPS

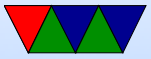


- Hot-swappable redundant hardware

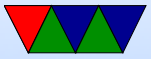


Lower Level

- Replicate units (ALU, etc)
- Replicate threads or important data wires
- CRCs and parity checks on all busses, caches, and memories

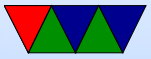


Lower-Level Problems



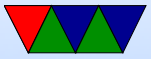
Soft errors/Radiation

- Chips so small, that radiation can flip bits. Thermal and Power supply noise too.
- Soft errors – excess charge from radiation. Usually not permanent.
- Sometime called SEU (single event upset)

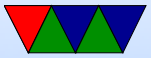


Radiation

- Neutrons: from cosmic rays, can cause "silicon recoil" Can cause Boron (doped silicon) to fission into Li and alpha.
- Alpha particles: from radioactive decay
- Cosmic rays – higher up you are, more faults Denver 3-5x neutron flux than sea level. Denver more than here. Airplanes. Satellites and space probes are radiation-hardened due to this.

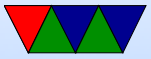


- Smaller devices, more likely can flip bit.



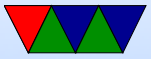
Shielding

- Neutrons: 3 feet concrete reduce flux by 50%
- alpha: sheet of paper can block, but problem comes from radioactivity in chips themselves



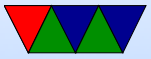
Case Studies

- "May and Woods Incident" first widely reported problem. Intel 2107 16k DRAM chips, problem traced to ceramics packaging downstream of Uranium mine.
- "Hera Problem" IBM having problem. ^{210}Po contamination from bottle cleaning equipment.
- "Sun e-cache" Ultra-SPARC-II did not have ECC on cache for performance reasons. High failure rate.



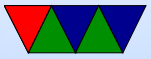
Hardware Fixes

- Using doping less susceptible to Boron fission
- Use low-radiation solder
- Silicon-on-Insulator
- Double-gate devices (two gates per transistor)
- Larger transistor sizes
- Circuits that handle glitches better.
- Memory fixes
  - ECC code
  - spread bits out. Right now can flip adjacent bits, flip



too many can't correct.

- Memory scrubbing: going through and periodically reading all mem to find bit flips.

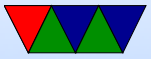


Extreme Testing

- Single event upset characterization of the Pentium MMX and Pentium II microprocessors using proton irradiation", IEEE Transactions on Nuclear Science, 1999.

- Pentium II, took off-shelf chip and irradiated it with proton. Only CPU, rest shielded with lead. Irradiate from bottom to avoid heatsink

- Various errors, freeze to blue screen. no power glitches or "latchup" 85% hangs, 14% cache errors no ALU or



FPU errors detected.