ECE 574 – Cluster Computing
Lecture 20

Vince Weaver

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

vincent.weaver@maine.edu

15 April 2025

Announcements

- HW#9 was posted (OpenCL), Due Friday
- Don't forget Project Status due Thursday, April 17th
- Don't forget midterm exam next week on Tuesday, April 22nd

Python in HPC

- C/C++/Fortran have somewhat steep learning curves
- Is it possible to use a higher level language?
- Python is popular
  - Can do MPI
  - Recent CUDA support
- Downside is Python can be 100x slower

Python Tradeoffs

- Pros
  - East to use/program
  - Lots of libraries/tools
  - External C libraries can provide fast code
- Cons
  - 100x slower
  - GIL (global interpreter lock) makes parallel code hard
  - Complex package dependencies on python version and libraries

◦ High disk usage – Often end up install multiple copies of large python libraries

Python related Tools

- Jupyter Notebooks – create documents containing code, data, visualizations
- Ipython (?)
- Spyder – Python IDE for scientific software

Python Package Managers

- Conda – python package manager (anaconda is the full distribution, miniconda minimal version)
- virtualenv

Ways to Improve Python Speed / External Libraries

- Most of these call out to code written in C for speed
- Numpy – fast matrix/array manipulation
- Scipy – more HPC code, DSP, image manipulation
- Pytorch – fast AI routines, including CUDA support

Ways to Improve Python Speed / JIT

- Compile interpreted python to machine code / executables
- Cython
- pypy – fast replacement for cython
- Numba – add $@jit$ note in code after profiling

Python, More traditional HPC

- Can profile python (snakeviz)
- Regular parallel programming limited by GIL global lock (they are working on removing it)
- MPI – can write MPI code using python

Big Data

- Until now compute or network bound systems
- What if we want to do a lot of disk/IO? Big Data?
- There are often trends in Computer Research
  - Takes a while to trickle down to UMaine, funny watching how administration will launch an initiative for the new "hot topic" after it has already peaked and is on the downswing elsewhere
  - Big Data was the previous hot thing, and sure enough UMaine just finalized its effort for it

- The current big thing is AI. I thought that had peaked too, but this time the Chat-GPT stuff came in at the last minute as a bit of a curve-ball so maybe it will stick around longer

Can you name big data datasets?

- Physics (particle accelerators)
- Astronomy
- Genomics
- AI Training
- Web search
- Social media
- Streaming (youtue, netflix)

Big Data Examples

- Where is Data Used a lot?
  - Google
  - Worldwide LHC Computing Grid (WLCG) as of 2012 25 petabytes of data/year (petabyte=1000 terabytes) 300 GByte/s of data incoming
- Big Data is about big data sets (Terabytes?) and trying to get useful information out of them. Data Mining.
- "Big Data: Astronomical or Genomical": PLos Biology, 7 July 2015. (as per IEEE Spectrum

December 2015). Twitter: 1-17PB/year. Astronomy, 1,000PB/year, YouTube 1,000-2,000 PB/year, Genomics 2,000-40,000PB/year

Big Data Challenges

- Data capture
- Data storage
- Data analysis / Visualization
- Data search / Querying
- Data sharing
- Data transfer
- Updating
- Information privacy (?) afterthought?

the Big Data Vs

- Volume – quantity, terabytes? Petabytes?
- Variety – more than just lists of numbers
- Velocity – speed the data is generated
- Veracity – GIGO (garbage in/garbage out)
- Value – Usefulness
- Variability –

Big Data

- A buzzword?
- How big is big?
- Terabytes?
- Too big for one machine?
- In general if fits in RAM ($< 32GB$) or fits on disk ($< 10TB$) you are better off just using a database or similar
- Once it won't fit on one machine, and you want to use a cluster, things get complicated.

Note on Hard Disk Sizes

- Grew exponentially, sort of like DRAM and CPU sizes
- Trailed off about 10 years ago
- For average machine has leveled off in the 1T - 10T size

Big Data Optimizations

- Key idea is to *move computation to the data*, rather than vice-versa

Big data challenges/astronomy

- https://www.computerworld.com/article/2972251/massive-telescope-array-aims-for-black-hole-gets-gu html
- Black hole "picture"
- From radio-wave interferometry
- Telescopes scattered all over world, including Antarctica
- Hard drives fail on mountain tops! (not enough air) use helium-filled ones instead
- Over 5 days, each telescope collected 900TB of data
- 1000-2000 hard drives, about 9PB

- How data sent? Hard-drives shipped to Massachusetts
- Had to wait for spring in Antarctica to ship out those
- 800 core cluster to analyze

Types of Storage

- Hard Disks
  - spinning rust – can be slow latency wise
  - Traditional vs Advanced features

    Shingled (SMR) Disks

    Perpendicular (PMR) Disks

    `https://www.youtube.com/watch?v=xb_PyKuI7II`

    Helium

    Caches
  - SSD – faster, why?

Use Flash? Reads vs Writes, wear leveling, Lose data over time if not powered up?

- Other flash / SD cards
- Memristors/Phase-Change/Optane/XPoint Non-volatile RAM
- Tape – robot tape libraries, etc
- Optical – CD/DVD/Blueray

RAID

- Redundant Array of (Independent / Inexpensive) Disks
- Patterson Gibson and Katz 1987: replace expensive mainframe disks with arrays of relatively cheap desktop drives
- RAID0: striping, spreading across multiple disks, can increase performance, increases size of disk, bad things happen if one drive fails
- RAID1: mirroring – same thing written to both drives can increase performance as either drive can answer

request

- RAID2: hamming code, each bit on separate drive. Not really used.
- RAID3: byte-level striping with parity. not common
- RAID4: block-level striping with dedicated parity.
- RAID5: block-level striping with distributed parity.
  can handle failure of single disk, can rebuild based on parity.
  Not recommended, as you have to read entirety of all other disks to rebuild, likely to fail other disks if all of same vintage

- RAID6: block-level striping with double parity. Recommended
- Hybrid: RAID10 = RAID1 + RAIDO
- Software vs Hardware
- Some filesystems include RAID like behavior: ZFS, GPFS, brfs, xfs

RAID5/6 further notes

- raid5, you need at least three drives. Data is striped across first two drives, third drive gets the bytes XOR with each other
- If a drive fails, you can xor the data from the two remaining drives to get the missing data
- This does waste some space. Also rebuilding can be stressful on remaining drives and cause them to fail
- RAID6 has an extra drive for redundancy, but reduces capacity

- Modern systems often do RAID10 (RAID1+0) which is striping plus full backup. Wastes half the space, but doesn't require XOR

Non-RAID

- nearline storage – not offline but also not fully online
- MAID – Massive Array of Idle Drives
  - Write once, read occasionally.
  - Data that you want to save, but really don't access often so taking seconds to recall is OK.
  - What kind of data? Backups? Old Facebook pictures?
  - Old science data?