Retrospective:
A Look Back at 20+ Years of Parallel Computing Education
(*in 15 Minutes?!?*)

Joel C. Adams
Department of Computer Science
Calvin University

1996-97: Beginnings

- Calvin's CS department identified *Parallel Computing* as a gap in our computing curriculum
- Because my dissertation area was *Distributed Systems*, I was drafted to create an advanced elective course (i.e., for 3rd- or 4th-year students), to be taught for the first time during 1997-98

Summer 1997: Colgate Workshop

Chris Nevison (Colgate) and Nan Schaller (RIT) led an NSF-sponsored faculty development workshop on parallel computing, covering:

- Different parallel computing hardware platforms (SIMD, MIMD)
- Parallel algorithms
- Hands-on use of parallel software platforms:
  - Parallaxis (for SIMD systems)
  - MPI or PVM for MIMD systems
- Course Materials: syllabi, assignments, slide decks, etc.
- Excellent model for creating a new course!

1998: Parallel Computing, v.1

- First *Parallel Computing* course offered at Calvin
  - Basically a customized/expanded version of the Colgate workshop
- Text: *Parallel Programming with MPI* (Pacheco)
- Topics: MIMD/MPI, SIMD/Parallaxis, parallel hardware models, different network topologies, parallel algorithms, analysis and time-complexity, scalability, Amdahl's law, Gustafson's law, history, …
- 6 Hands-On Labs+Projects:
  - 3 SIMD using Parallaxis (simulator running per workstation)
  - 3 MIMD using MPI (distributed processes running on 32-node NoW)
- Problem: Students were unable to experience speedup / scalability b/c they were competing for CPU cycles on single-core workstations…

1998-99: NSF Proposal #1

- Applied for NSF Major Research Instrumentation (MRI) grant to build a Beowulf cluster at Calvin
- Turned down for various reasons, including:
  - All existing Beowulf clusters were at national labs or graduate school labs
  - Reviewers were skeptical that an undergraduate institution could build one

MBH'99: Calvin's First Beowulf Cluster

- Mark Ryken's Senior Project
- 9 Nodes: 25MHz 486 PCs
  - Head node
  - 8 compute nodes
- 100Mbps Ethernet
  - Star+Hypercube Topology
- CPU-bound
  - Too slow to be useful
- Good proof-of-concept / prototype system

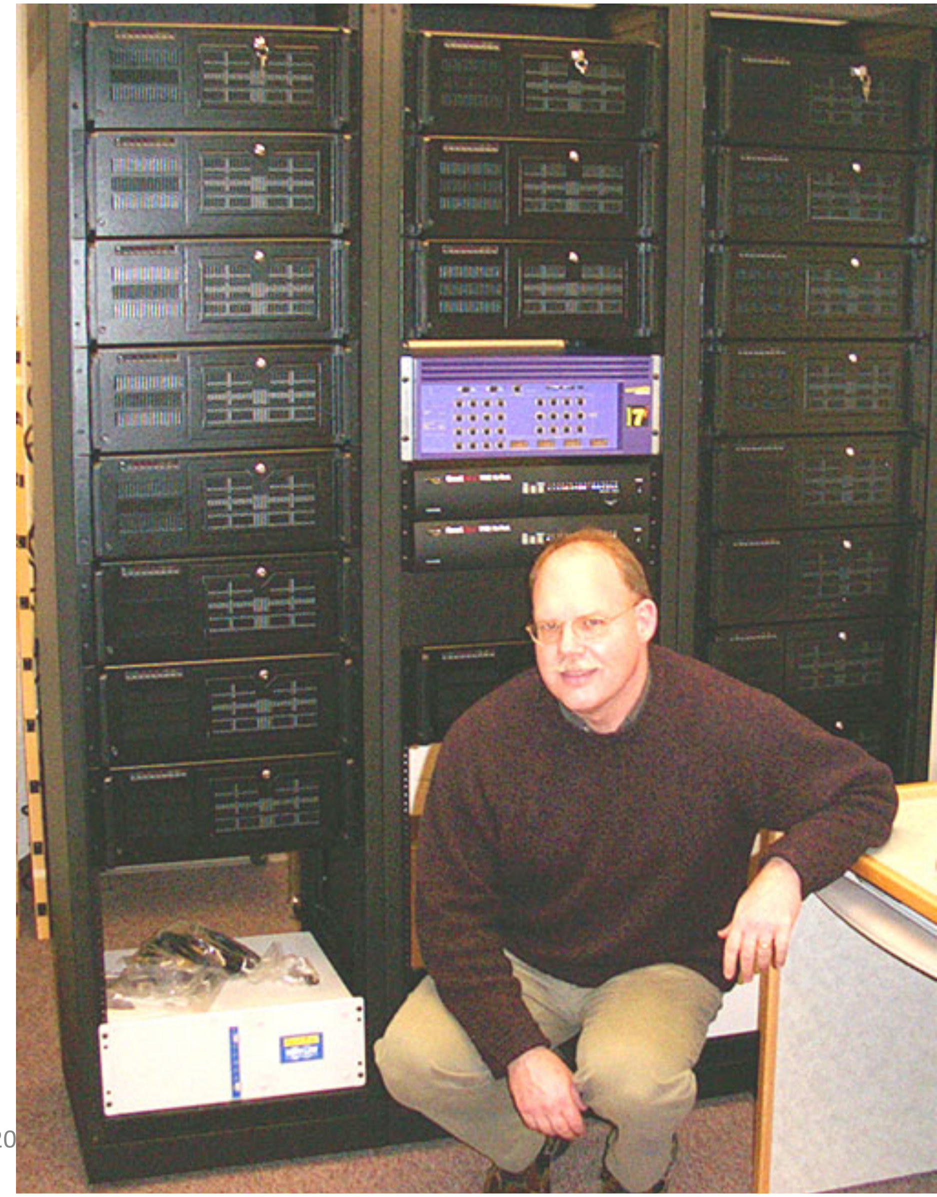
2000: Parallel Computing, v.2

Changes to v.1:

- Minor tweaks to fix issues:
  - Perl script to randomize MPI host files
  - Updated versions of Parallaxis and MPICH

2000: Ohm

- NSF-MRI funded Beowulf Cluster
- 18 Nodes: 1GHz AMD Athlon-64s
  - Two head nodes
  - 16 compute nodes
- Gigabit Ethernet
  - Star+Hypecube Topology
- HPC: 10.4 Gflops (R_{max})



2002: *High Performance Computing*, v.1

HPC replacement for our *Parallel Computing* course, using Ohm

- MPI segment uses Ohm
  - Students develop/debug MPI programs on NoW
  - Transfer/run/time them on Ohm to test scalability using 1,2,4,8,16,32 processes
  - Students get to experience parallel speedup and scalability
- Feedback:
  - Students wanted more of the course devoted to MIMD, MPI, Ohm
  - SIMD seemed to be generally declining in importance
- NoW workstations refreshed: 1GHz AMD Athlon-64 CPUs (same as Ohm)
  - Ohm faster b/c on it, students aren't competing for CPU cycles

2003: *HPC, v.2*

Significant Revision:

- First half of the course: Distributed multiprocessing and MPI as primary introduction to parallelism
- Last half: Symmetric Multiprocessors (SMPs) increasingly affordable, so replaced SIMD/Parallaxis material with SMP/OpenMP material

Summer 2004: NCSI Workshop

- National Computational Science Institute + Shodor Foundation
- Instructors: Paul Gray, David Joiner, Tom Murphy, Charlie Peck
- Host: Henry Neeman
- MPI-based science examples
  - Monte Carlo Forest Fire simulation
  - Galaxy simulation
  - …
- *LittleFe*: Cluster in a suitcase
  - Ultimately led to *LittleFe Buildouts*

2005: Fulbright in Iceland

… teaching Distributed Computing

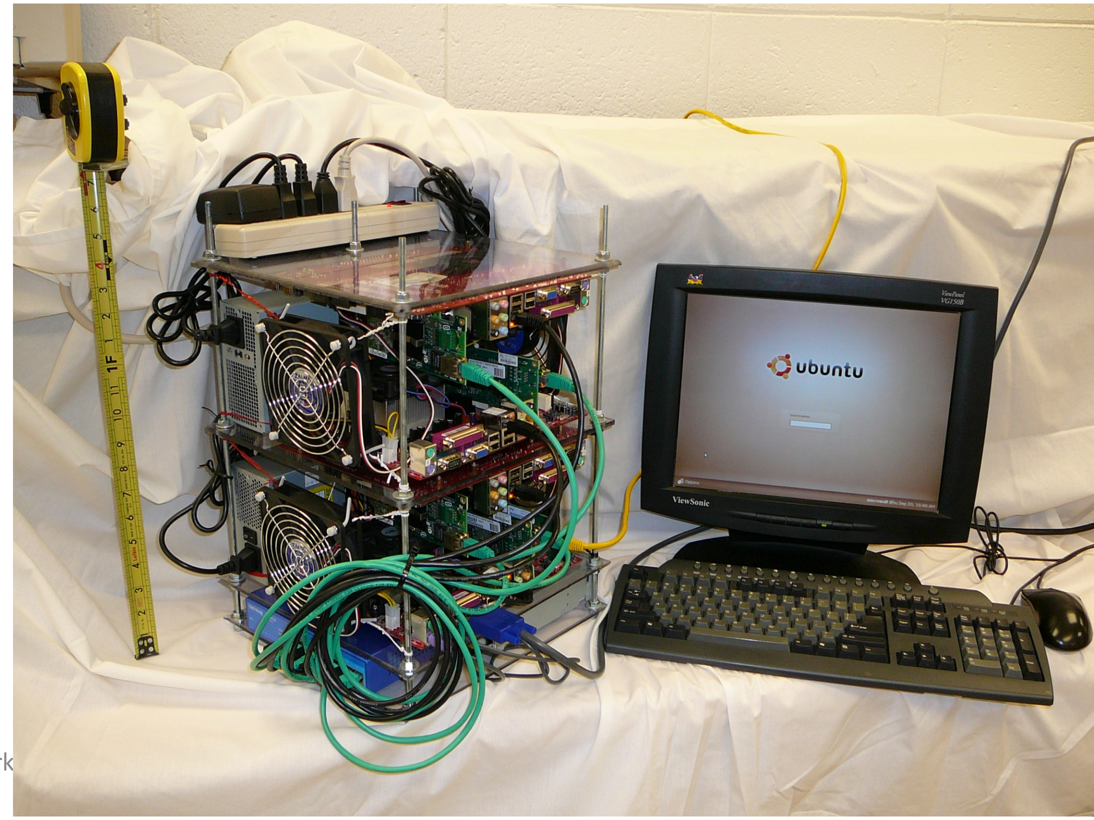
- Built *Sleipnir*
  - 6-node Beowulf cluster
  - 3.8 GHz Pentium 4 CPUs
  - Gigabit Ethernet
  - 20.25 Gflops (R_{MAX})!
- Some non-English speakers
  - Minimalist MPI-stmt demos
    → *Patternlets*

Fall 2005: *HPC, v.3*

- Changed textbook:
  - *Parallel Programming in C with MPI and OpenMP* (Quinn)
- Replaced less interesting content with material from 2004 NCSI workshop
  - Programming Project: Monte Carlo Forest Fire Simulation
- Replaced some lectures with Patternlet-based demos
  - Iceland MPI examples
  - New OpenMP examples
- NoW workstations refreshed: 35 x 3.2GHz Athlon-64s
  - Students notice NoW is faster than Ohm
  - 4-year-old cluster is showing its age b/c of Dennard Scaling

2006: Microwulf

- Gigabit Ethernet becomes on-board NIC motherboard standard
- Dual-core CPUs released by AMD and Intel
- Personal, portable HPC cluster
  - 4 dual-core nodes:
    - 3.8GHz AMD Athlon-64 X2 CPUs
  - 11” x 12” x 17”
- 26.25 Gflops (R_{MAX})
- $2,470 in Jan 2006
  - $94/Gflop
- $1,255 in Aug 2006
  - $48/Gflop
- Aided by senior Tim Brom



Fall 2007: *HPC, v.4*

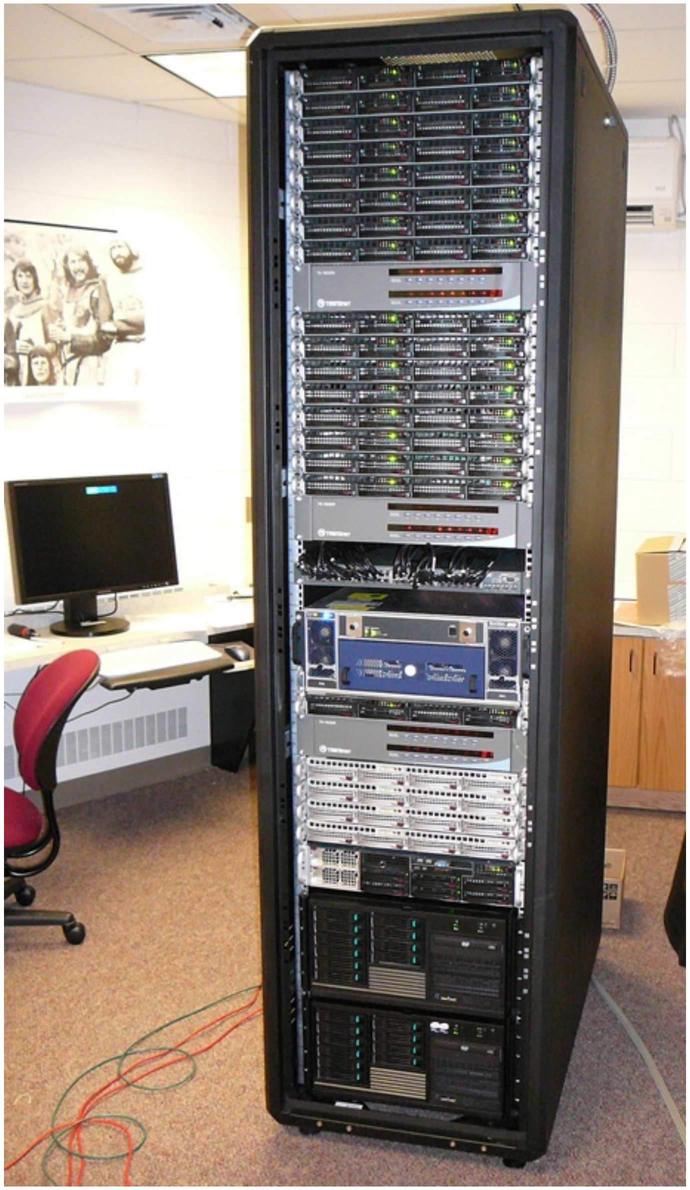
Changes from v.3:

- Replaced slower Ohm with faster Microwulf
  - 1-8 processes: Microwulf faster than NoW
  - Microwulf's scalability limited to 8 processes
    - NoW faster for 16+ processes, *if lab is empty*
- Added coverage of MPI-2 and OpenMP 2.0 features
  - Unit on MPI-2's Parallel I/O mechanism
  - Unit on OpenMP 2.0's user-defined reduction operations

2008: Dahl

NSF-MRI funded HPC cluster

- 45 nodes
  - Two 2.2GHz Intel quad-core Xeon CPUs
- Interconnect
  - Infiniband data network
  - Gigabit Ethernet admin network
- Storage
  - Dual 20TB RAID arrays
- 3.7 Tflops (R_{PEAK})
- Major research upgrade
  - 20+ publications generated



2008-09: Parallel Coverage Elsewhere

- Dual- and quad-core CPUs ubiquitous
  - Multicore is the new hardware standard / foundation on which software runs
  - *All* students need more multithreading experience
- CS2 (*Data Structures*) taught in C++
  - Easy to add parallelism / multithreading using OpenMP
  - Lab exercise: speed up Matrix addition, transpose
    - Record times and plot time vs number of threads
  - Project: speed up other Matrix ops
- CS 232 (*Operating Systems & Networking*)
  - Already covering shared- and distributed-memory concurrency
  - Added new material on thread performance, false sharing, etc.

Fall 2009: *HPC, v.5*

- Dahl replaces Microwulf
  - First half: MPI using Dahl; test scalability using 1,2,4,8,16,32,64,128,256 processes
  - Second half: OpenMP using Dahl; test scalability using 1,2,4,8,16 threads
    - All of Dahl's nodes have at least 8 cores
- NoW workstations refreshed: 3.6GHz Intel i7 CPUs (50% faster than Dahl's CPUs)
  - MPI:
    - 1-8 processes: NoW beats Dahl
    - 16 processes: even
    - 32+ processes: Dahl beats NoW
  - OpenMP:
    - 1-4 threads: NoW beats Dahl
    - 8-16 threads: Dahl beats NoW

2010: TCPP Early Adopter Program

Early Adopter award; as dept chair:

- Used funds as "carrots" for colleagues to add 1-week's parallel content to:
  - CS 212: *Algorithms*
    - Parallel algorithm design, specific parallel algorithms, distributed graph algorithms, …
  - CS 214: *Programming Languages*
    - Shared memory parallelism, including lab comparing performance of Ada, C++, Ruby
  - CS 262: *Software Engineering*
    - Engineering of distributed mobile apps using cloud-service APIs
  - ENGR 220: *Intro to Computer Architecture*
    - Multicore processor and cache organization, bus organization, memory controllers, etc.

All are CS core courses, ensuring all Calvin CS majors get extensive exposure to PDC topics across the curriculum

Fall 2011: *HPC, v.6*

Changes from v.5:

- Added coverage of GPUs, CUDA and OpenCL
  - Students were able to complete the CUDA exercises
  - They struggled with the OpenCL exercises
- Reduced OpenMP material to make room
  - Not an issue b/c OpenMP was getting more coverage in the core

2013: ACM/IEEE CS 2013 Curriculum

Thanks to involvement of NSF/IEEE TCPP representatives:

- New PDC knowledge area
- Recommended all CS majors receive significant exposure to PDC, especially shared-memory / multithreading / multicore topics

Fall 2013: *HPC, v.7*

Significant course overhaul:

- Parallel design patterns as a unifying theme
- New text: *Introduction to Parallel Programming* (Pacheco)
- 1-week module on Pthreads
  - Hands-on Pthread exercise (implement parallel reduction pattern)
- 1-week module on heterogeneous computing
  - Hands-on MPI+OpenMP exercise
  - Previous OpenMP material reduced to 1 week
- 1-week module on co-processors (Parallella, Xeon Phi)
  - Hands-on Xeon Phi exercise
  - Previous OpenCL material reduced to 1 lecture

Fall 2015: *HPC, v.8*

Changes from v.7:

- Incorporated TSGL visualizations so students could *see* parallel design pattern behaviors (e.g., parallel loop) in real-time
- Added 1 week's coverage of "Big Data" concepts + technologies (MapReduce / Hadoop / HDFS):
  - Hands-on: Hadoop exercise
  - Compressed coverage of HPC history to make room
- A few of 7-year old Dahl cluster's nodes fail

Fall 2017: *HPC, v.9*

Changes from v.8:

- Added material on Apache Spark to end-of-course "Big Data" module
  - Better/higher performance than MapReduce/Hadoop
  - Modular software architecture (data streams, ML, SQL, …)
- About 50% of Dahl's 45 nodes had failed

2018: Borg (*Anita*, not Star Trek)

NSF-MRI funded HPC cluster for handling Big Data

- Nodes: 3.2GHz Xeon Gold CPUs, 16 cores/node, 96GB RAM, …
  - Head node
  - Virtualization node (VMs offering different web services)
  - 20 compute nodes
  - "Big Data" / GPU node: 768GB RAM, 4 Nvidia Titan V GPUs
  - Dual file-server nodes: 110TB tiered storage
- Interconnect:
  - 100Gbps Ethernet data network
  - 10Gbps admin network
- Support for research + new Data Science program

Fall 2019: *HPC, v.10*

Changes from v.9:

- Used Borg in place of Dahl:
  - Borg: 20 x 3.2GHz Xeon Gold compute nodes, 100Gbps Ethernet
  - NoW: 35 x 3.7GHz i7 workstations, Gigabit Ethernet
  - 33 students bog down the NoW; exclusive-access Borg usually faster
- Coprocessors:
  - Coprocessors were disappearing (Adapteva: ~~Parallella~~; Intel: ~~Xeon Phi~~)
  - Replaced coprocessor material with expanded coverage of CUDA/GPUs

HPC Topic Evolution: 1998-2019

Wk	1998	2000	2002	2003	2005	2007	2009	2011	2013	2015	2017	2019
1	Parallaxis	Parallaxis	Parallaxis	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI
2	Parallaxis	Parallaxis	Parallaxis	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI
3	Parallaxis	Parallaxis	Parallaxis	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI
4	Parallaxis	Parallaxis	Parallaxis	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI
5	Parallaxis	Parallaxis	Parallaxis	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI
6	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI	MPI
7	MPI	MPI	MPI	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	Pthreads	Pthreads	Pthreads	Pthreads
8	MPI	MPI	MPI	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP
9	MPI	MPI	MPI	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP
10	MPI	MPI	MPI	OpenMP	OpenMP	OpenMP	OpenMP	OpenMP	Hetero	Hetero	Hetero	Hetero
11	MPI	MPI	MPI	OpenMP	OpenMP	OpenMP	OpenMP	CUDA	CUDA	CUDA	CUDA	CUDA
12	History	History	History	History	History	History	History	OpenCL	XeonPhi	XeonPhi	XeonPhi	CUDA
13	History	History	History	History	History	History	History	History	History	Big Data	Big Data	Big Data

Observations

- Major hardware innovations are like earthquakes that send shockwaves through the software (and educational) communities
  - There will undoubtedly be more to come; change is inevitable…
- Hardware continues to change, but the rate of change is slowing down…
  - Dennard Scaling has ended; will Moore's Law be next?
  - GPU-computing has helped reduce energy-consumption: what is the next step?
- Software continues to change, trying to keep up with the hardware…
  - MPI, OpenMP standards continue to evolve
  - GPUs: CUDA, OpenCL, OpenACC, or OpenMP?
  - Languages: Chapel, Julia, Scala, …?
- A modern NoW is a pretty good multiprocessor (for small-sized classes)!

Conclusions

- Good news: there are some points of stability:
  - *Parallel Design Patterns* are a stable body of knowledge
  - Technologies with official standards tend to be stable (or at least backwards-compatible)
- Good students are an invaluable resource:
  - They are curious & energetic
  - Find what motivates them and turn them loose!
- Faculty development workshops can be an invaluable resource:
  - Knowledgeable instructors with experience you can leverage
  - Materials for getting started, or to 'refresh' a course you've taught before
  - A supportive community of people interested in the workshop's topic

Finally

See the paper for many more details that there wasn't time to cover…

Links:

- Email: adams@calvin.edu
- HPC course materials: https://cs.calvin.edu/courses/cs/374/

Many thanks to Calvin, NSF, the TCPP, and *you* for your kind attention!