ECE 6930-004 Fault Tolerance and Reliability in High-Performance Computing

INTRODUCTION AND SYLLABUS

DR. JON CALHOUN

24 AUGUST 2017

Who am I?

Dr. Jon Calhoun

Title: Assistant Professor of Electrical and Computer Engineering

Office: 221-C Riggs

Education:

- 2017 Ph.D. Computer Science from University of Illinois at Urbana-Champaign
- 2012 B.S. Mathematics with Minor in Statistics from Arkansas State University
- 2012 B.S. Computer Science from Arkansas State University

Who am I?

Dr. Jon Calhoun

Research:

- Fault tolerance of HPC systems
- Application based fault tolerance
- Large scale fault injection techniques
- Failure analysis tools
- Approximate computing and data transfer
- Scalability of computational science and engineering applications

Software:

- FlipIt – Compiler based fault injector
- FaultSight – Fault analysis tool
- SDCProp – Corruption propagation tracking tool

Who are you?

How long have you been at Clemson? Where are you from?

Research interests, background, advisor?

Why are you taking the course? Expectations?

Course goals

Understand:

- Current challenges to reliability of HPC systems
- Options to mitigate the challenges

Develop:

- Ability to read and critique papers
- Presentation skills
- Project skills

Bonus:

- Publication at workshop, conference, or journal

Prerequisites

Required:

- ECE 3220 – Introduction to Operating Systems
- ECE 3290 – Computer Systems Structures

Suggested:

- ECE 4730 – Introduction to Parallel Systems

Parallel experience is not required!
Canvas survey to poll background
(Helps me adjust my examples)

Course materials

This course will post all relevant material to Canvas

- Assignments
- Papers
- Grades
- Project submission

Canvas Link:

- https://clemson.instructure.com/courses/35155

Class requirements and grading

Scale:

- A: (90%, 100%]; B: [80%, 90%); C: [70%, 80%); D: [60%, 70%); F: [0%, 60%)

Assignment	Undergraduate	Graduate
Paper and Presentation Reviews	10%	10%
Homework	10%	10%
Paper Presentations	25%	25%
Discussion Participation	10%	10%
Final Exam	25%	10%
Project*	20%	35%

**Undergraduate projects no more than 4 students.*
Graduate projects no more than 2 students. Graduate projects are expected to be more rigorous.

Paper presentations

- One paper assigned per class
- One student will present the paper and lead the discussion
- Presentation focuses on the paper, but may contain supplemental material
- Presentation should highlight major ideas in a way that fosters discussion
- Example discussion topics include:
  - New ideas and techniques
  - Comparisons with other papers
  - Strengths of paper
  - Shortcomings of paper
  - How you would have done things differently?
- Assume all students have read the paper

Paper presentations

Post presentation to Canvas by 5pm on the day prior to your presentation

- I will send an individual assignment to submit the file PowerPoint or PDF preferred

Give yourself enough time to read the paper, follow up papers, and key references

Practice, Practice, PRACTICE!!1

Grading:

- Clarity of slides/presentation
- Understanding of paper/topic
- Quality of discussion
- Time management

Presentation critiques

Submit to Canvas by 5pm on day prior to next paper being discussed

Presentation critiques (~5 sentences):

- Positives of the presentation/discussion
- How could the presentation be improved

Be constructive with comments

- Comments will be shared anonymously with the presenter

Paper reviews

Submit to Canvas by 5pm on day prior to paper being discussed

Paper reviews (~3 sentences per bullet):

- Summary of the paper
- Strengths of the paper
- Shortcomings of the paper
- How would you improve on this work?
- A question you would like discussed

The questions will be sent to the presenter prior to class

You can also ask the question during the discussion

Review grading

Paper:

- Summary (1 point)
- Strengths (1 point)
- Weakness (1 point)
- Improvement (2 points)
- Question (2 points)

Presentation:

- Positives (1 points)
- Improvement (2 points)

Class participation

Class is more fun when there is a discussion

We are all friends here

Ask your question… ask all the questions!!!

Respond to the presenter's questions

Homework

1-3 during the semester

Mixture of “pen and paper” style homeworks and machine problems

Very little coding mostly running code and analyzing results

Due 5 pm on due date (~1-2 weeks after assigned)

Project

Undergraduates:
- Group 4 or less

Graduates:
- Group of 2 or less

Project can either:
- Explore new idea
- Validate prior work

Graded based on:
- Proposal
- Progress report
- Final report

More details coming soon

Final exam

Exact format is TBA

Current thoughts are an extended homework

- Not harder just more work

I will update you when it is finalized

Course polices

Late assignments:

- Deduction of 1 letter grade per day late
- Will update you on late assignment submission
  - Still trying to master Canvas

Cheating:

- DON’T DO IT
- All assignments are individual except for the project. Do your own work.

Office hours and contact info

Office hours:

- Tuesday 11-12 a.m.
- Additional office hours can be arranged via email

Email Policy:

- **calhou3@clemson.edu**
- To receive a response, emails must be sent from your Clemson University email address and must include the course number
- I will respond within 24-hours during weekdays unless on travel
- Longer latencies may occur on weekends

Course outline

Part 1: Introduction to HPC and reliability

Part 2: Checkpoint-restart

Part 3: Soft errors

Part 4: Approximate computing and lossy compression

Homework 0

Select your first paper from the options shown here

Date	Paper/Topic	Presenter
8/24	Introduction/Syllabus/What is HPC	Calhoun
8/29	Basic Fault Tolerance Concepts	Calhoun
8/31	Toward Exascale Resilience	Calhoun
9/5	Lessons Learned From the Analysis of System Failures at Petascale: The Case of Blue Waters	
9/7	Basics of Checkpoint-restart	Calhoun
9/12	A Job Pause Service under LAM/MPI+BLCR for Transparent Fault Tolerance	
9/14	Evaluation of Simple Causal Message Logging for Large-Scale Fault Tolerant HPC Systems	
9/19	Design, Modeling, and Evaluation of a Scalable Multi-level Checkpointing System	
9/21	MCRENGINE: A Scalable Checkpointing System Using Data-Aware Aggregation and Compression	

Any questions?

ECE 6930-004 Fault Tolerance and Reliability in High-Performance Computing

WHAT IS HIGH-PERFORMANCE COMPUTING?

DR. JON CALHOUN

24 AUGUST 2017

Background

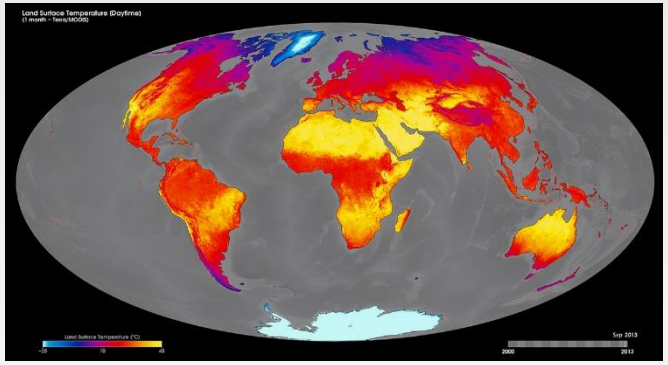
Any parallel experience?

What does parallel or high-performance computing mean to you?

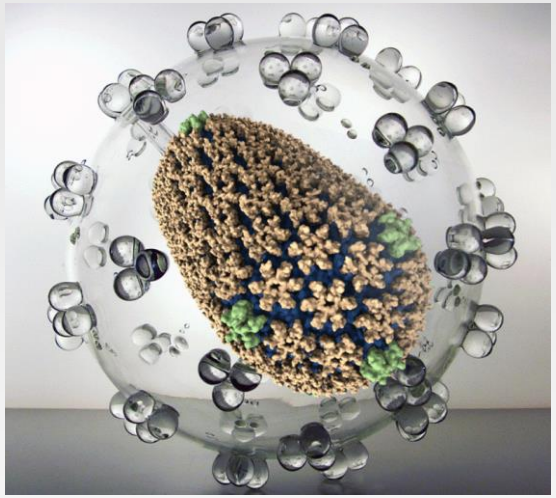
What is high-performance computing?

High-performance computing (HPC) focuses massive compute capabilities for exploration and solving of previously intractable computational problems

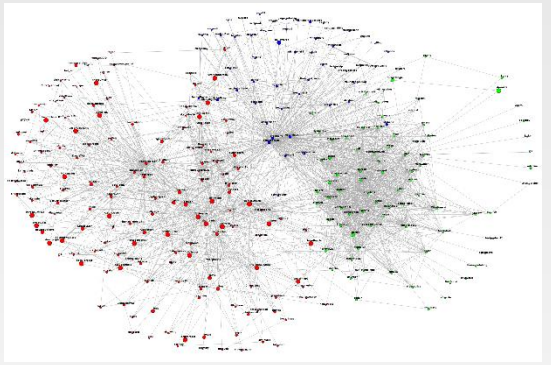
HPC is an essential tool of science and discovery in many fields of science, engineering, and industry



Land surface temperature [NASA]



HIV capsid [Beckman Institute]



Twitter social graph [Olin College]

"Why does high-performance computing matter? Because science matters! Discovery matters! … HPC extends our reach, putting more knowledge, more discovery, and more innovation within our grasp" - Dr. Shishir Pandya NASA Aerospace Engineer

High-performance computing (HPC)

Applications are the driving force behind HPC system design

HPC systems seek to be general by supporting applications from many different domains:

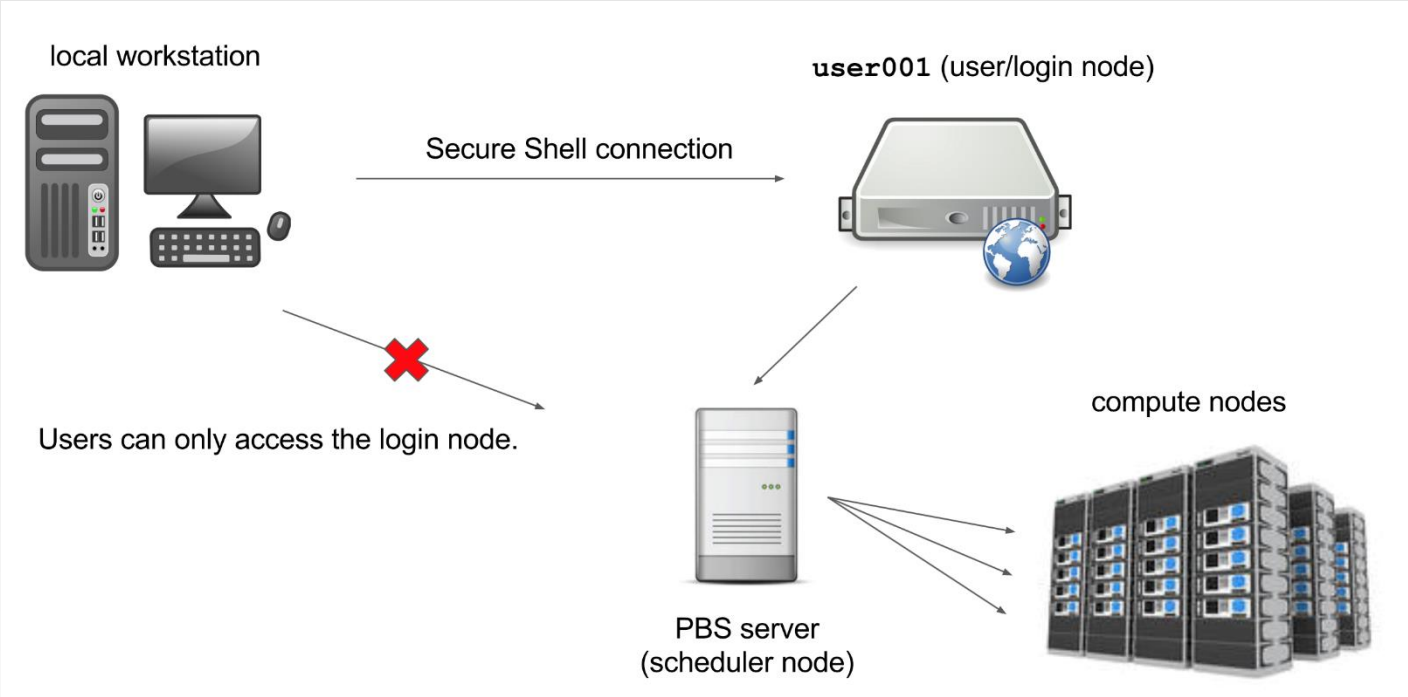
- Climate
- Physics
- Mechanical engineering
- Finance
- Biology

HPC system architecture

User sends job to job scheduler to be run

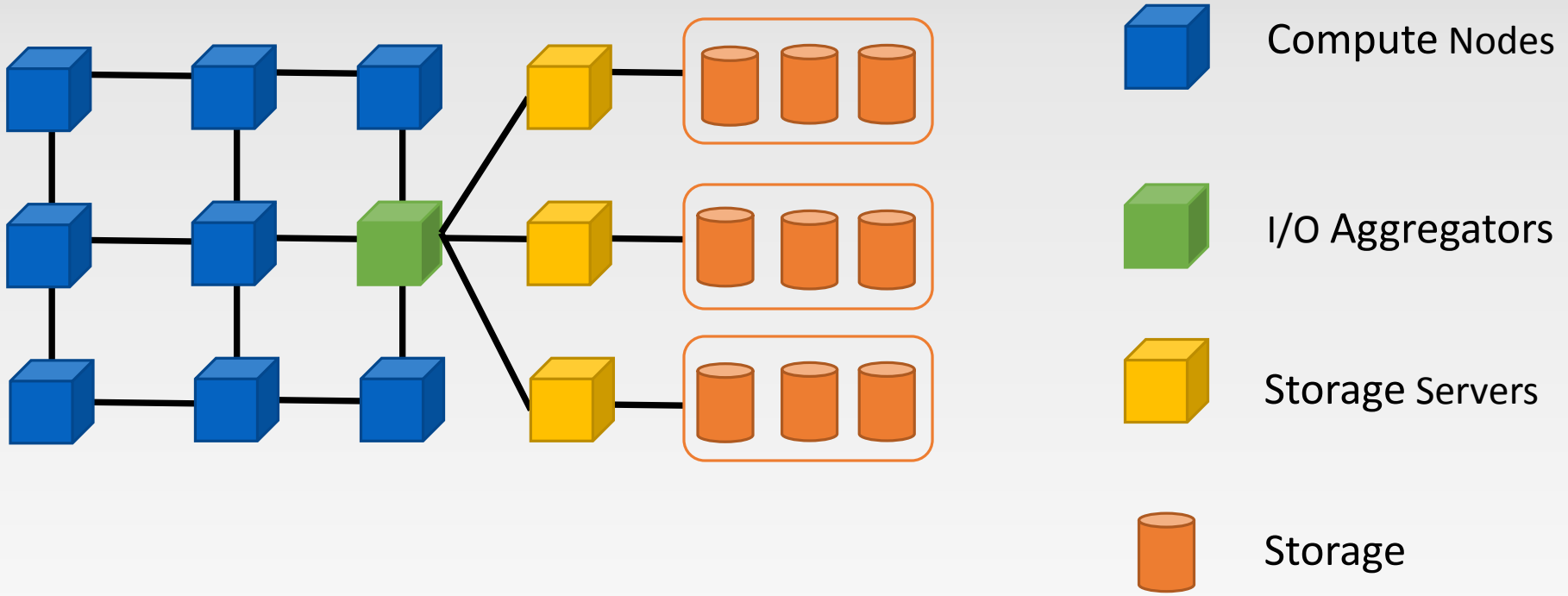
Scheduler ensures all jobs are run

Only running applications interact with the “machine”



HPC frontend [CCIT]

The “machine”



Failures can strike anywhere inside the system
This course will focus primarily on failures impacting the compute nodes

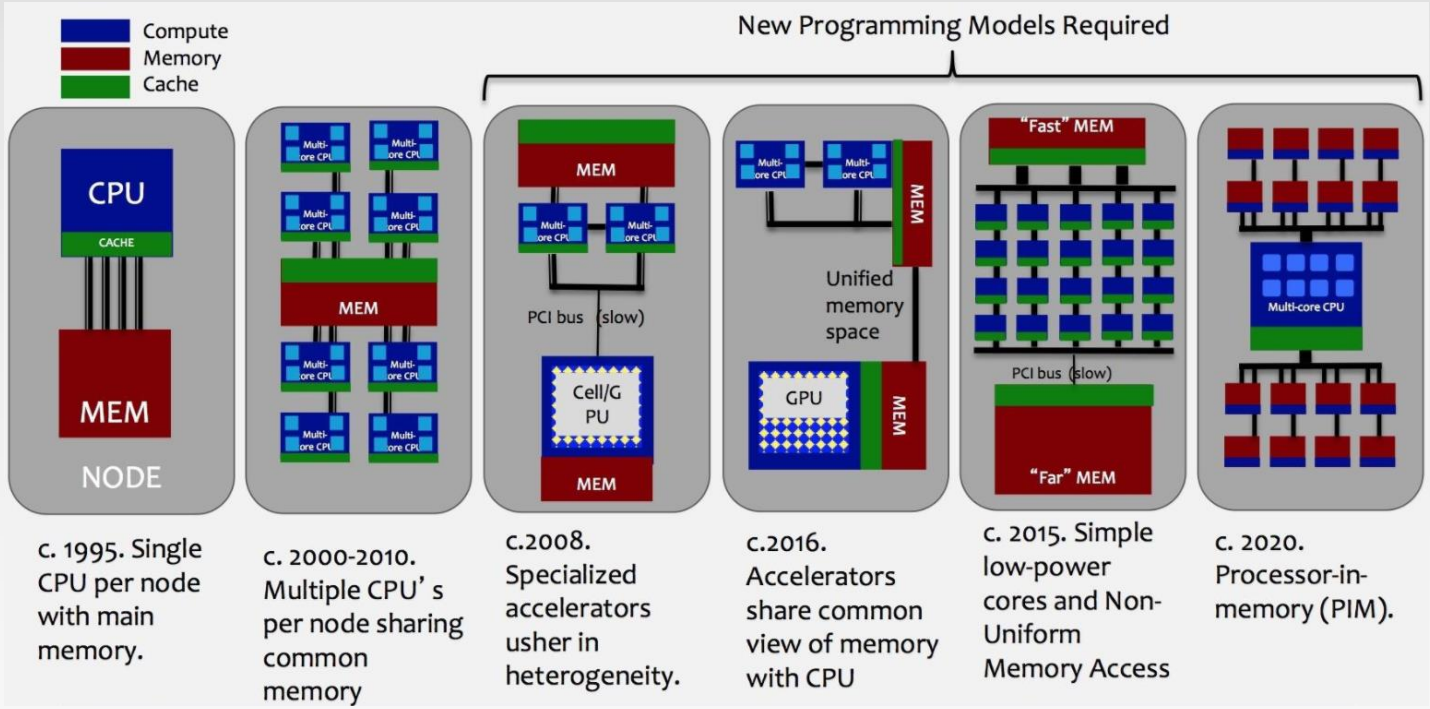
Compute nodes

Changed drastically in the last decade

Increasing emphasis on many smaller cores

More memory levels

Complexing of compute node increasing



HPC compute node evolution [LLNL]

How do applications make use of the system?

Many HPC applications simulate physical phenomena:

- Tornados
- Airplanes
- Cells

Simulations evolve a 3D domain though time:

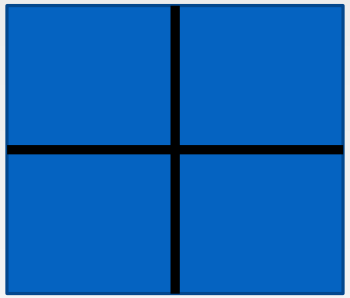
```
for t = 0 to T
     update_variables();
```

How to map the problem to the system?

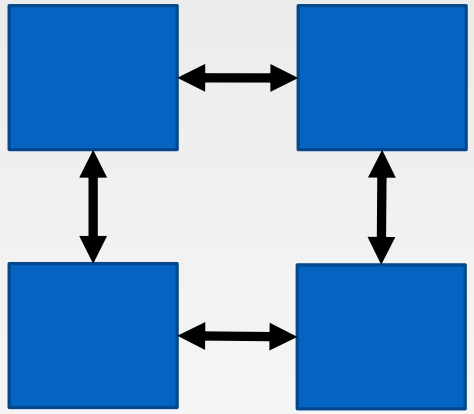
Most important question of HPC application design

Standard approach distribute regions of physical domain among parallel processes

Original domain



Domain partitioned amongst 4 processes



Each process must communicate to update variables in its region

Bulk synchronous model

```
for t = 0 to T
      communicate();
      update_variables();
```

Each process updates local region before sending data for next iteration

Creates two phases:

- Communication
- Computation

Communication can be local (shared-memory) or remote(distributed memory)

Going forward

There are other programming models, but this course will mainly stick to bulk synchronous

In the coming weeks we will discuss:

- Define a common vocabulary
- How failure strikes current systems
- Expected trends on next generation systems
- How to recover from failure of a process
- How to recover from corrupted results
- How to optimize performance with *corrupted* results

Any questions?

Backup Slides