CS 5413: High Performance Systems and Networking

Hakim Weatherspoon
Assistant Professor, Dept of Computer Science
CS 5413: High Performance Systems and Networking
January 25, 2017
Goals for Today

• Background
• Why take this course?
• How does this class operate?
• Class details
Who am I?

• Prof. Hakim Weatherspoon
  – (Hakim means Doctor, wise, or prof. in Arabic)
  – Background in Education
    • Undergraduate University of Washington
      – Played Varsity Football
        » Some teammates collectively make $100’s of millions
        » I teach!!!
    • Graduate University of California, Berkeley
      – Some class mates collectively make $100’s of millions
      – I teach!!!

– Background in Operating Systems
  • Peer-to-Peer Storage
    – Antiquity project - Secure wide-area distributed system
    – OceanStore project – Store your data for 1000 years
  • Network overlays
    – Bamboo and Tapestry – Find your data around globe
  • Tiny OS
    – Early adopter in 1999, but ultimately chose P2P direction
The promise of the Cloud

– A computer utility; a commodity
– Catalyst for technology economy
– Revolutionizing for health care, financial systems, scientific research, and society
• The promise of the Cloud

– ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

NIST Cloud Definition
The promise of the Cloud

- ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

NIST Cloud Definition
• The promise of the Cloud
  – ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.  

• Requires fundamentals in distributed systems
  – Networking
  – Computation
  – Storage

NIST Cloud Definition
High Performance Networks

• How to optimize a global network of data centers?
High Performance Networks

- How to optimize a global network of data centers?
  - E.g. Need to optimize movement of data between DCs
High Performance Networks

• How to optimize a global network of data centers?
  – E.g. Investigate novel data center designs

  [ToN 2013 and ANCS 2012; best paper]

Core Switch (CS)

Aggregate Switch (AS)

ToR

...
High Performance Networks

• How to optimize a global network of data centers?

• Rack-scale computers
  – Rack is now the new computing block
  – New hardware trends allows to create a rack containing ~100s-1000 SoC

HP Moonshot  Boston Viridis  AMD SeaMicro
SW Programming the L1/L2 of the Network

• Dataplane programming (e.g. P4) via P4FPGA [SOSR’16, http://p4fpga.org]

• Physical Layer programming via SoNIC [NSDI’13,’14, IMC’14, SIGCOMM 2016]
SW Programming the L1/L2 of the Network

- Dataplane programming (e.g. P4) via P4FPGA [SOSR’16, http://p4fpga.org]
- Physical Layer programming via SoNIC [NSDI’13,’14, IMC’14, SIGCOMM 2016]
SW Programming the L1/L2 of the Network

• **P4**: Programming Protocol-Independent Packet Processors
  [SOSR’16]

• Use P4 to describe many different network applications

• Compile P4 to many different FPGA platforms

\[ \text{P4} \rightarrow \text{P4 HLIR} \rightarrow \text{intermediate representation} \]

\[ \text{p4c-fpga} \rightarrow \text{BSV(Bluespec System Verilog)} \rightarrow \text{Verilog} \rightarrow \text{FPGA firmware} \]
High Performance Networks

SW Programming the L1/L2 of the Network

• E.g. Move Consensus into the network
  – Consensus as a Service (CAANS)
  – Consensus protocols are the foundation for fault-tolerant systems

• Key: Low latency and high throughput
SW Programming the L1/L2 of the Network

• **Datacenter Time Protocol** [SIGCOMM 2016]
  • Synchronization protocol in the PHY
    – Each physically link is already synchronized!
    – No protocol stack overhead
    – No network overhead
    – Scalable: peer-to-peer and decentralized
High Performance Networks

SW Programming the L1/L2 of the Network

- **Datacenter Time Protocol** [SIGCOMM 2016]
- **Synchronization protocol in the PHY**
  - Bounded and precise synchronization
  - Bounded by 4 oscillator ticks (25ns) peer-to-peer
  - Bounded by 150ns for an entire datacenter
    - No clock differs by more than 150ns
  - Free – No network traffic: Use the PHY!
Network Latency Control via Rack-scale computing

- Assuming synchronized time, schedule every packet
- Every node is allocated a full time slot to a single destination
- No two nodes will be able to communicate with the same destination at the same time

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Node 2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Node 3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Node 4</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Node 5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Benefits

• No network contention
• Full bisection bandwidth
  – Direct connect topology
  – Route through one random intermediate node
• Bounded latency
• Low power

Node 1  2  3  4  5
Node 2  3  4  5  1
Node 3  4  5  1  2
Node 4  5  1  2  3
Node 5  1  2  3  4
High Performance Storage

- Large organizations considering using the cloud
  - New York Times
  - Netflix
  - Nintendo
  - Cornell
  - Library of Congress

- The more data you have, the harder it is to move
  - Switching providers entails paying for bandwidth *twice*
  - Inhibits opportunistic migration
High Performance Storage

• How hard is it to move a PetaByte?

Titan tech boom, randy katz, 2008
High Performance Storage

• All my valuable data/computation is in the cloud
  Am I locked in to one provider forever?
  – The more data you have, the harder it is to move

• RACS: Redundant Array of Cloud Storage
  – Collaboration with the Internet Archive and IBM
  – [SOCC 2010]; See Also [EuroSys 2007, FAST 2009, FAST 2013]
High Performance Storage

• All my valuable data/computation is in the cloud. Am I locked in to one provider forever?
  – The more data you have, the harder it is to move.

• RACS: Redundant Array of Cloud Storage
  – Collaboration with the Internet Archive and IBM
  – See Also [EuroSys 2007, FAST 2009, FAST 2013]
High Performance Storage

Estimated Cost of Switching Cloud Providers

- Single Provider
- RACS (4,5)
- RACS (6,7)
- RACS (8,9)

Cost in $K

Date

High Performance Storage

- **RACS**: How do I optimize storage globally
  - Collaboration with Internet Archive / IBM
- **Gecko**: How do I optimize storage locally
  - Collaboration with Google and Microsoft

<table>
<thead>
<tr>
<th>Relative Storage</th>
<th>$n/k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Upload Bandwidth</td>
<td>$n/k$</td>
</tr>
<tr>
<td>Relative Download Bandwidth</td>
<td>$1$</td>
</tr>
</tbody>
</table>
High Performance Computation

• Can I compute in the cloud if some of my data is in a vault at home or on another provider
• Xen-Blanket and VirtualWire
  • Collaboration with IBM
  • [HotCloud 2012, EuroSys 2012]
High Performance Computation

- Can create your own *Cloud-within-a-Cloud*

- Migrate computation among different cloud providers
High Performance Computation

- Can create your own *Cloud-within-a-Cloud*

Demo: http://supercloud.cs.cornell.edu

- Migrate computation among different cloud providers
My Contributions

• Cloud Networking
  – Wireless DC in ANCS 2012 (best paper) and NetSlice in ANCS 2012
  – Bifocals in IMC 2010 and DSN 2010
  – Maelstrom in ToN 2011 and NSDI 2008
  – Chaired Tudor Marian (2010), Ki Suh Lee (2016), Han Wang (2017) PhD

• Cloud Computation & Vendor Lock-in
  – Supercloud in SoCC 2016, OSR 2015 / IEEE Internet Computing-2013
  – Xen-Blanket in EuroSys-2012 and HotCloud-2011
  – Overdriver in VEE-2011
  – Chaired Dan William (2012) and Zhiming Shen (~2017) PhD

• Cloud Storage
  – Isotope/Gecko in FAST 2015, 2013 / HotStorage 2012
  – RACS in SOCC-2010
  – SMFS in FAST 2009
  – Chaired Lakshmi Ganesh (2011) and Ji Yong Shin PhD 2016
Goals for Today

• Be brief!
• Background on course and Professor
• Why take this course?
• How does this class operate?
• Class details
Why take this course

• Learn about systems abstractions, principles, and artifacts that have lead to the high performance systems and networks we see in the cloud,
• Understand attributes of systems research that is likely to have impact,
• Become comfortable navigating the state of the art in systems and networking,
• Gain experience in thinking critically and analytically about systems research, and
• Acquire the background needed to work on cloud and data center problems currently under study at Cornell and elsewhere.
Who is the course “for”?

• MEng students
  – Students who have mastered 4410/4411
  – PhD students as well
  – Serious undergraduates

• MEng Project
  – Projects in this course can be used to satisfy MEng project requirements
Goals for Today

• Be brief!
• Background
• Why take this course?
• How does this class operate?
• Class details
How this class operates

• Instructor: Hakim Weatherspoon
  – hweather@cs.cornell.edu
  – Office Location: 427 Gates Hall

• TA: Vishal Shrivastav
  – vishal@cs.cornell.edu

• Lectures:
  – Three slots reserved a week,

  ***but lecture will be twice a week on average***
Course Help

• Course staff, office hours, etc:
  – http://www.cs.cornell.edu/courses/cs5413/2017sp

• MEng projects
  – http://www.cs.cornell.edu/courses/cs5413/2017sp/projects.htm
CS 5413: Overview

- **This is a hands-on course**
  - Objective: Have fun!
  - Build systems and networks

- **Prerequisite:**
  - Mastery of CS 4410 and 4411 material
    - Fundamentals of OS design
    - How parts of the OS are structured
    - What algorithms are commonly used
    - What are the mechanisms and policies used
    - Programming in C/C++

- **Class Structure**
  - Lecture/Readings/Reading questions
  - In-class labs/Take-home Homeworks
  - Course Project/BOOM
  - Prelim
CS 5413: Topics

• Overview
  – Cloud computing, and Internet vs Data Center Networks

• High Performance Networking Basics
  – Textbook networking vs Data Center Networks
  – Network protocol stack: TCP/IP protocol stack

• High Performance Data Center Systems & Networks
  – Basic Switching Technologies: 50Gb/s routers & NetFPGA
  – Data Center Topologies, Software Router Designs,
  – Alternative switching technologies, Data Center Transport
  – Software defined networking, virtual networks
  – Rack-scale networks and computers
  – Disaggregated Datacenters
  – Middleboxes, advanced topics
  – Data Center traffic and analysis
• Required reading is always *one* paper and/or book reading
  – Book reading provides basic background knowledge
  – Papers pulled from, best journals and conferences
    • TOCS, SOSP, OSDI, …

• Read papers before each class and bring notes
  – takes ~1 hr, write notes and questions

• Answer a paper related question and turn in *at least two hours* before beginning of class
  – Turn on online via Course Management System (CMS)
  – *No late reviews will be accepted*
• A question(s) related to the reading will be posted at usually a week before class.
• Each student is required to write a response to question
• Turn question response in online before class via CMS
  – Be succinct
    • Usually can answer question in two or three sentences or at most one paragraph
• 40 minute presentation
• All students are required to read ahead of time and participate!
• Counts in final grading.
• May have student (group) presentations
CS 5413: In-class Labs/Homeworks

• In-class Labs
  – Work in groups
  – ~once per week
  – Need a laptop
  – Designed to finish during class

• Take-home Homeworks
  – work in groups
  – 1-3 weeks per lab/homework
  – Topics
    • Building a network proxy (single-threaded, then multi-threaded)
    • Building a chatserver
    • Implement a programmable dataplane device or software-defined network (SDN) switch/controller

• Facilities
  – Laptops and local Fractus cloud
CS 5413: Project/BOOM

- One major project per group
  - Groups include three people
- Group formation – early February
- Initial selection of project topic – due mid-February
- Survey of area (related works)–late begin of February

- Midterm draft paper – late of March
- Peer reviews—due a week later

- Final demo/presentation–due begin of May
- Final project report – due a week later

Extra – BOOM

- http://boom.cornell.edu/students-and-faculty/project-guidelines/
- BOOM Submission – due March 31st
- BOOM Presentation – April 19th
• SoNIC: Software Network Interface Cards
• NetFPGA/P4FPGA

• Fractus: our very own (mini) cloud
• Amazon’s Cloud Infrastructure EC2/S3
• Emulab
• PlanetLab
• Cornell’s Center for Advanced Computing (CAC)
• ...
Academic Integrity

• Submitted work should be your own

• Acceptable collaboration:
  – Clarify problem, C syntax doubts, debugging strategy
  – You may use any idea from any other person or group in the class or out, provided you *clearly state what you have borrowed and from whom*.
  – If you do not provide a citation (i.e. you turn other people's work in as your own) that is *cheating*.

• Dishonesty has no place in any community
  – May NOT be in possession of someone else’s homework/project
  – May NOT copy code from another group
  – May NOT copy, collaborate or share homework/assignments
  – University Academic Integrity rules are the general guidelines

• Penalty can be as severe as an ‘F’ in CS 5413
Stress, Health and Wellness

• Need to pace yourself to manage stress
  – Need regular sleep, eating, and exercising

• Do **not** come to class sick (with the flu)!
  – Email me *ahead* of time that you are not feeling well
  – People not usually sick more than once in a semester
• Read one paper below and write review

• Check website for updated schedule