

# CS15-319 / 15-619

## Cloud Computing

Recitation 11

November 4<sup>th</sup> and November 6<sup>th</sup>, 2014

# Announcements

- Encounter a general bug:
  - Post on Piazza
- Encounter a grading bug:
  - Post Privately on Piazza
- Don't ask if my answer is correct
- Don't post code on Piazza
- Search before posting
- Post feedback on OLI

# Piazza Questions

- `sudo ndbd --initial`

[ndbd] ERROR -- Could not connect to management server, error: ''


- Solution: Allow inbound at port 1186 in the Management node.

- `ndb_mgmd -f`

[MgmtSrvr] ERROR -- Could not determine which nodeid to use for this node.

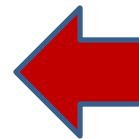
- Solution: Use privateIP in the config.ini

# Module to Read

- UNIT 5: Distributed Programming and Analytics Engines for the Cloud
  - Module 16: Intro to distributed programming for the Cloud 
  - Module 17: Distributed analytics engines: MapReduce
  - Module 18: Distributed analytics engines: Pregel
  - Module 19: Distributed analytics engines: GraphLab
  - Quiz 5: Distributed Programming and Analytics Engines for the Cloud

# Project 3

- Files vs. Databases
  - File vs. Database
- Vertical Scaling in Databases
  - Vertical Scaling
- Horizontal Scaling in Databases
  - Horizontal Scaling
- Working with NoSQL: DynamoDB / Hbase
  - Write Scalability
  - Provisioned Databases



# Amazon RDS



Amazon RDS

- Relational Database Service;
- MySQL, Oracle, MS SQL server, other;
- AWS manages the DB for you!

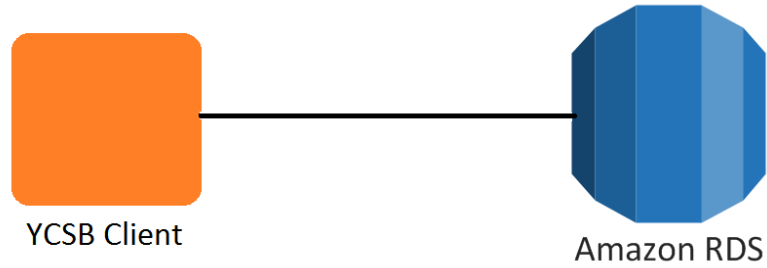
# Amazon DynamoDB



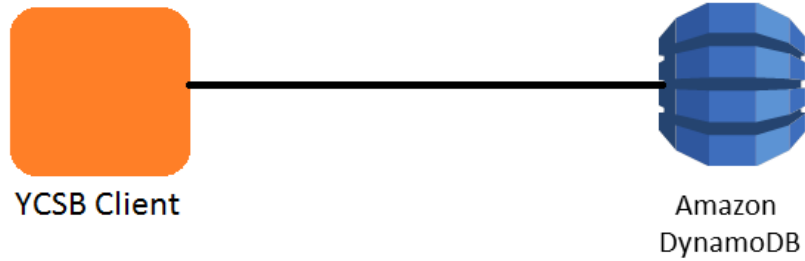
- Managed NoSQL Database Service;
- Limited functionality. No guarantee of ACID properties;
- Expensive!

# Tasks

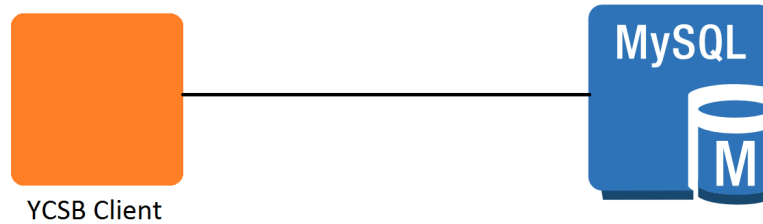
**Task 1:**



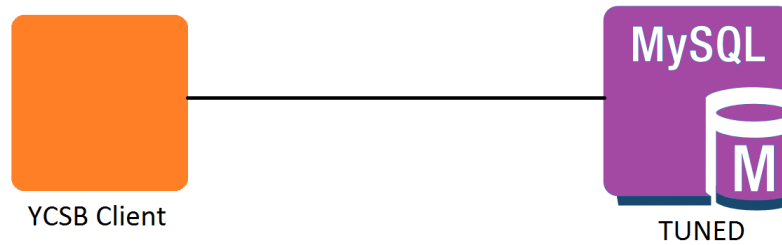
**Task 2:**



**Task 3:**



**Task 4:**



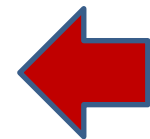


# Upcoming Deadlines

- Project 3:

<u>Project 3</u>		
<a href="#">Files vs. Databases</a> <a href="#">(Gradebook)</a> <a href="#">(Learning Dashboard)</a>		
File vs. Database	<a href="#">Checkpoint</a>	<a href="#">Ended 10/12/14 11:59 PM</a>
<a href="#">Vertical Scaling in Databases</a> <a href="#">(Gradebook)</a> <a href="#">(Learning Dashboard)</a>		
Vertical Scaling	<a href="#">Checkpoint</a>	<a href="#">Ended 10/19/14 11:59 PM</a>
<a href="#">Horizontal Scaling in Databases</a> <a href="#">(Gradebook)</a> <a href="#">(Learning Dashboard)</a>		
Horizontal Scaling	<a href="#">Checkpoint</a>	<a href="#">Ended 10/26/14 11:59 PM</a>
<a href="#">Working with NoSQL: DynamoDB / HBase</a> <a href="#">(Gradebook)</a> <a href="#">(Learning Dashboard)</a>		
Write Scalability	<a href="#">Checkpoint</a>	<a href="#">Ended 11/2/14 11:59 PM</a>
Provisioned Databases	<a href="#">Checkpoint</a>	<b>Available Now</b> <a href="#">Due 11/9/14 11:59 PM</a>

- Project 619 Phase 2: 11:59 PM Thursday 2014/11/06





# TWITTER ANALYTICS: THE 619 PROJECT

# Leaderboard (as of 2 PM on Nov 6)

<u>Q1</u>	<u>Q2</u>	<u>Q3</u>	<u>Q4</u>	<u>Overall</u>
apt143	Darwin	PHD	xnoobs	FDU
cloudreaper	xnoobs	xnoobs	Amazombies	xnoobs
FDU	cloudreaper	Echo	phoenix	PHD

**Exciting stuff !!!**

---

# Phase 2 Report [VERY IMPORTANT]

- Start early
- Document your steps



© Scott Adams, Inc./Dist. by UFS, Inc.

- Identify and isolate the performance impact of each change you make
- Document your ideas and experiments

MAKE A QUANTITATIVE, DATA-DRIVEN REPORT

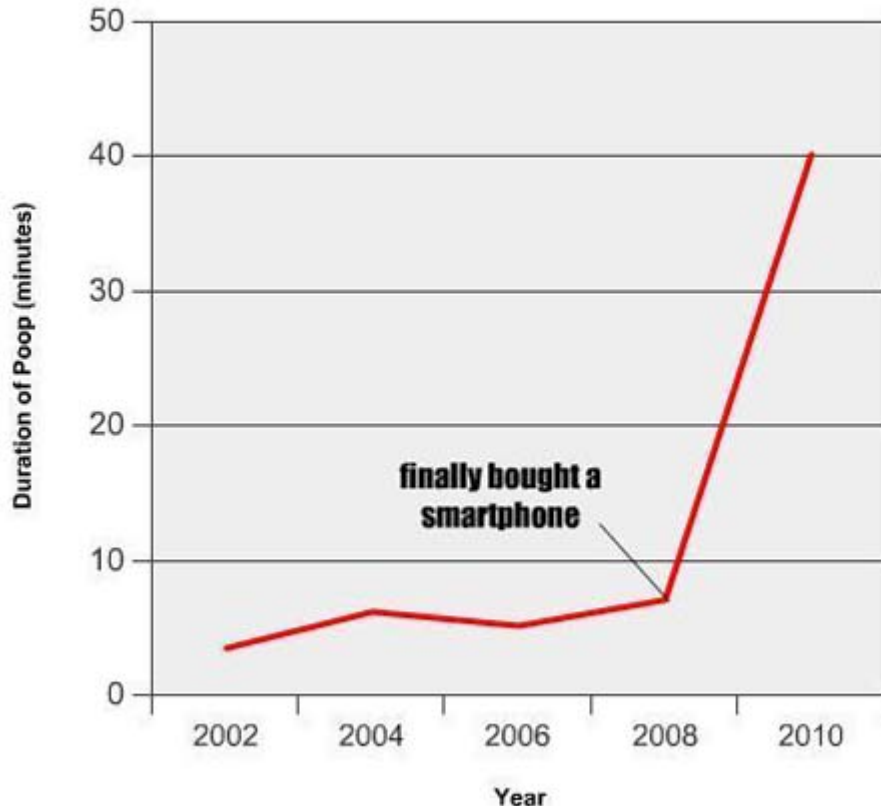
---

# Good reports contain Tables

<u>Idea</u>	<u>Expected Result</u>	<u>Submission id</u>	<u>Observed Result</u>	<u>Inference</u>
Use a Godzilla caching proxy server	200% improvement in throughput (based on reference[1])	10022, 10024	No improvement (8k rps)	Caching proxy did not help as the front end server we use already caches responses
Use ZoomZoom Web Server	1 million rps (based on YouTube video [2])	11211-11217	25k rps	LG is not fast enough
...	...	...	...	...

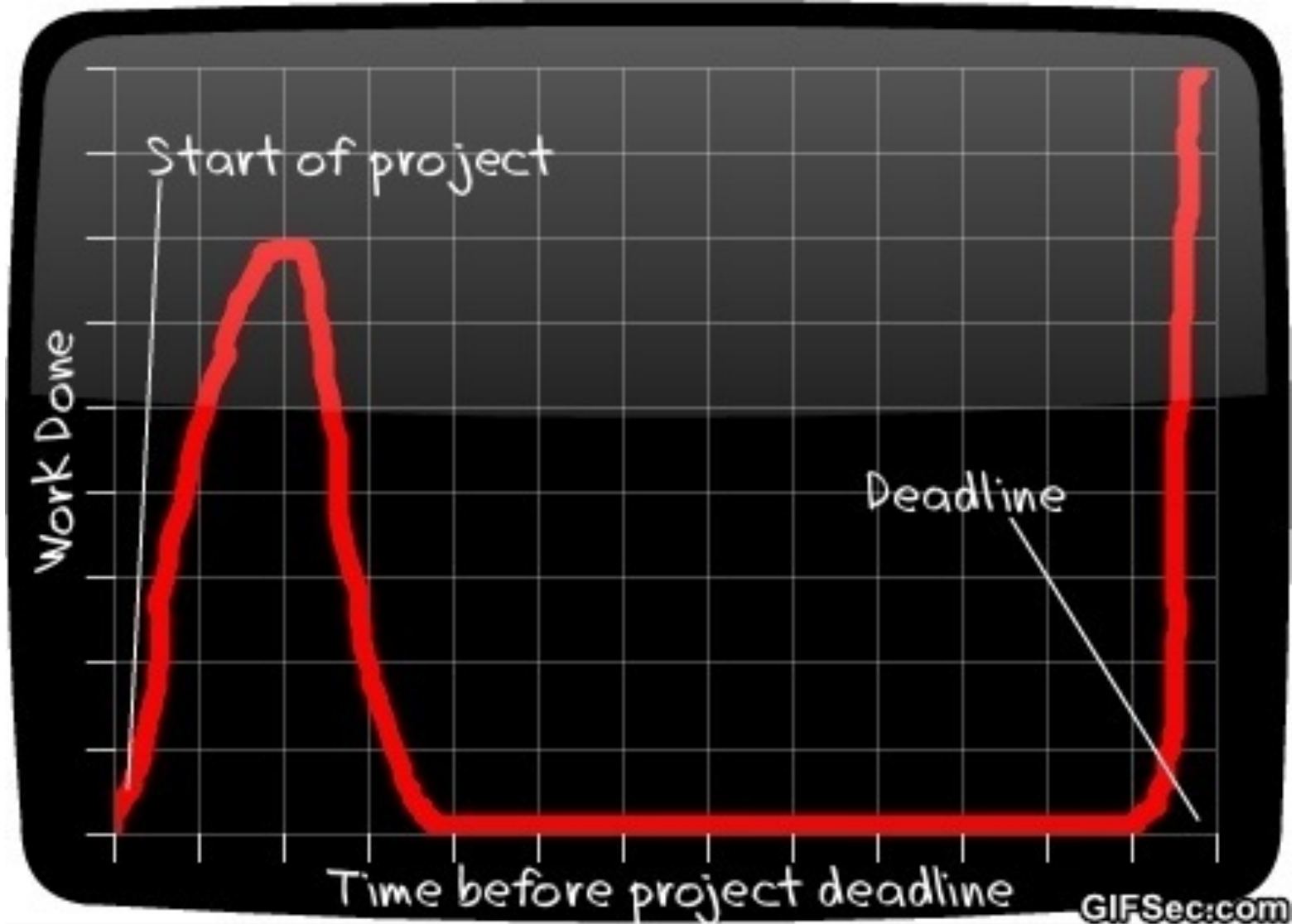
# Good reports contain Charts/Graphs

## Time Spent Pooping



**Illustrate the effectiveness of your optimization technique by tweaking parameters and plotting the delta**

# Warning : Live Test in < 66 hrs



# Warning : Live Test in < 66 hrs

- Predicting your performance in the Live Test using job submissions
  - Cache 22?
    - When to cache? What to cache? What not to cache?
  - Grading
    - Choose which Live Test you want to be scored on
-



# Live Test Logistics & Costs

- \$1.25 per hour per system
    - ~ \$5 for iteration 1 (first 3 hours)
    - ~ \$9 for iteration 2 (running for 6+ hours)
    - **Do you have \$15 left? Else, save money (autoscale(?) / spot (risky!!) / 10% penalty) !!!**
  - Submit your URLs by 11 PM tonight
  - Live Test Part 1 starts at midnight exactly
  - Live Test Part 2 starts at 3 AM (+ delta)
-

# Timeline (+/- delta) EST

Start Time	End Time	What's Happening
0000	0030	Warmup 1
0030	0100	Q1 MySQL
0100	0130	Q2 MySQL
0130	0200	Q3 MySQL
0200	0230	Q4 MySQL
0230	0300	<b><u>MySQL Mix</u></b>
0300	0330	Warmup 1
0330	0400	Q1 HBase
0400	0430	Q2 HBase
0430	0500	Q3 HBase
0500	0530	Q4 HBase
0530	0600	<b><u>HBase Mix</u></b>

# How to do well in a Live Test?

- Don't crash. If you do, recover fast!!!
  - Make smart trade-offs (focus on your score)
  - Hopefully you have simulated a Live Test  
(self-warmup) (no precaching) (if not do it NOW!!!) (sequential 20 minutes)
-

# Help Us Help You

- What do you need from us to help you score 100%?



# Suggestions / Improvements

- UI
- Favicon
- Design
- Report comments
- Features
- Bugs
- Feedback



<http://bit.ly/1roJsvU>

**Any questions?**

