CS15-319 / 15-619
Cloud Computing

Recitation 10
October 28th and October 30th, 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
• **OLI does not show timer for Quiz! You have to maintain your own timer!**
Project 3, Module 3 Reflections

• Cost is different between running a fixed number of instances vs autoscaling

• Custom Metrics

  • Queries per Second = Queries / (Uptime1 – Uptime2)
  • Transactions Per Second (TPS)
    • Factor in the two queries used to find the number of queries and uptime.
    • *sysbench* transaction equals to 16 queries
  • Send TPS information to CloudWatch using API
Project 3, Module 3 Reflections

• Horizontal Scaling of Databases

• Horizontal Scaling of SQL Database is not easy to implement.
Project 3, Module 3 Problems

• The `/etc/init.d` directory contains a number of start/stop scripts for various services on your system.

• `/etc/rc.local` file runs after all other init level scripts have run. You can put commands that you want to have issued upon startup.

• **cron:** enable users to execute commands or scripts automatically at a specified time/date.
Project 3, Module 3 Problems

@1640 - Unable to SSH to AMI

- When copying mysql backup, it should go to /storage/mountpoint;
- Some students copied to root folder, and when they executed ‘chown’, it caused mysql to own the root folder, thus making ubuntu user unable to access the folder that contains the key (*.pem) file.
Module to Read

• UNIT 4: Cloud Storage
  – Module 12: Cloud Storage
  – Module 13: Case Studies: Distributed File Systems
  – Module 14: Case Studies: NoSQL Databases
  – Module 15: Case Studies: Cloud Object Storage
  – Quiz 4: Cloud Storage
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
  – DynamoDB vs. HBase
Sharding

• Horizontal partitioning (i.e. partitioning of the rows)
Sharding

• Steps

1. Configure management node;

2. Configure database nodes;

3. Testing.
YCSB Tests

1\textsuperscript{st}: Single Node

YCSB Client  \rightarrow  MS SQL

Single MySQL Server

2\textsuperscript{nd}: MySQL Cluster

MySQL Management Node

YCSB Client  \rightarrow  MS SQL

MySQL Data Node/API Node 1  \rightarrow  MySQL Data Node/API Node 2  \rightarrow  MySQL Data Node/API Node 3
# Upcoming Deadlines

## Project 3:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Scalability</td>
<td>Checkpoint, Available Now</td>
</tr>
<tr>
<td>DynamoDB vs. HBase</td>
<td>Checkpoint, Not yet assigned</td>
</tr>
</tbody>
</table>

## Unit 4:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 15: Case Studies: Cloud Object Storage</td>
<td>Available Now, Due date TBD by instructor</td>
</tr>
<tr>
<td>Quiz 4: Cloud Storage</td>
<td>Checkpoint, Due 10/30/14 11:59 PM</td>
</tr>
</tbody>
</table>
TWITTER ANALYTICS: THE 619 PROJECT
How's the 619 Project?

FEAR
Be afraid. Be very afraid.
<table>
<thead>
<tr>
<th>Phase Score</th>
<th>Q1 Besc</th>
<th>Q1 Correctness</th>
<th>Q1 HTTP Error Rate</th>
<th>Q1 Throughput</th>
<th>Q1 Latency</th>
<th>Q2 Besc</th>
<th>Q2 Correctness</th>
<th>Q2 HTTP Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11</td>
<td>100.00</td>
<td>0.00</td>
<td>1670.7</td>
<td>55</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>100.00</td>
<td>0.00</td>
<td>2827.8</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>10</td>
<td>47</td>
<td>100.00</td>
<td>0.00</td>
<td>7064.6</td>
<td>13</td>
<td>2</td>
<td>92.00</td>
<td>0.01</td>
</tr>
<tr>
<td>13</td>
<td>67</td>
<td>100.00</td>
<td>0.00</td>
<td>10001.0</td>
<td>9</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>14</td>
<td>73</td>
<td>100.00</td>
<td>0.00</td>
<td>10587.0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>14</td>
<td>72</td>
<td>100.00</td>
<td>0.00</td>
<td>10571.4</td>
<td>8</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>14</td>
<td>71</td>
<td>100.00</td>
<td>0.00</td>
<td>10576.6</td>
<td>9</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>15</td>
<td>74</td>
<td>100.00</td>
<td>0.00</td>
<td>11059.0</td>
<td>8</td>
<td>2</td>
<td>45.00</td>
<td>0.04</td>
</tr>
<tr>
<td>15</td>
<td>78</td>
<td>100.00</td>
<td>0.00</td>
<td>11649.2</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0.17</td>
</tr>
<tr>
<td>15</td>
<td>78</td>
<td>100.00</td>
<td>0.00</td>
<td>11671.2</td>
<td>8</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>15</td>
<td>75</td>
<td>100.00</td>
<td>0.00</td>
<td>11309.8</td>
<td>8</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>16</td>
<td>84</td>
<td>100.00</td>
<td>0.00</td>
<td>12542.2</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0.27</td>
</tr>
<tr>
<td>16</td>
<td>83</td>
<td>100.00</td>
<td>0.00</td>
<td>12487.0</td>
<td>7</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>16</td>
<td>83</td>
<td>100.00</td>
<td>0.00</td>
<td>12492.0</td>
<td>7</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>17</td>
<td>84</td>
<td>100.00</td>
<td>0.00</td>
<td>12664.5</td>
<td>7</td>
<td>1</td>
<td>7.00</td>
<td>0.01</td>
</tr>
<tr>
<td>17</td>
<td>87</td>
<td>100.00</td>
<td>0.00</td>
<td>13037.0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>17</td>
<td>89</td>
<td>100.00</td>
<td>0.00</td>
<td>13366.9</td>
<td>7</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>17</td>
<td>88</td>
<td>100.00</td>
<td>0.00</td>
<td>13282.6</td>
<td>7</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>17</td>
<td>86</td>
<td>100.00</td>
<td>0.00</td>
<td>12835.6</td>
<td>7</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>17</td>
<td>84</td>
<td>100.00</td>
<td>0.00</td>
<td>12632.5</td>
<td>7</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>18</td>
<td>73</td>
<td>100.00</td>
<td>0.00</td>
<td>10969.8</td>
<td>8</td>
<td>9</td>
<td>44.00</td>
<td>12.67</td>
</tr>
<tr>
<td>18</td>
<td>79</td>
<td>100.00</td>
<td>0.00</td>
<td>11807.1</td>
<td>8</td>
<td>4</td>
<td>45.00</td>
<td>0.02</td>
</tr>
<tr>
<td>18</td>
<td>84</td>
<td>100.00</td>
<td>0.00</td>
<td>12564.3</td>
<td>7</td>
<td>3</td>
<td>66.00</td>
<td>0.02</td>
</tr>
<tr>
<td>18</td>
<td>93</td>
<td>100.00</td>
<td>0.00</td>
<td>14029.3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0.03</td>
</tr>
<tr>
<td>18</td>
<td>90</td>
<td>100.00</td>
<td>0.00</td>
<td>13538.1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>18</td>
<td>90</td>
<td>100.00</td>
<td>0.00</td>
<td>13487.7</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>19</td>
<td>83</td>
<td>100.00</td>
<td>0.00</td>
<td>12461.5</td>
<td>7</td>
<td>5</td>
<td>72.00</td>
<td>13.12</td>
</tr>
</tbody>
</table>
Phase 1 Score Distribution

Mean (Average):
36.04444

Median:
$(24+25)/2 = 24.5$

Mode:
$17, 18, 20$
Congratulations Team FDU

- The early bird gets the worm

<table>
<thead>
<tr>
<th>Team</th>
<th>Phase Score</th>
<th>Q1 Best</th>
<th>Q1 Throughput</th>
<th>Q2 Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDU</td>
<td>174</td>
<td>127</td>
<td>19082.7</td>
<td>218</td>
</tr>
<tr>
<td>xnoobs</td>
<td>140</td>
<td>121</td>
<td>18174.2</td>
<td>151</td>
</tr>
<tr>
<td>WaterPig</td>
<td>98</td>
<td>111</td>
<td>16634.1</td>
<td>99</td>
</tr>
<tr>
<td>spartans</td>
<td>96</td>
<td>121</td>
<td>18148.3</td>
<td>124</td>
</tr>
<tr>
<td>LanXiang</td>
<td>95</td>
<td>106</td>
<td>15861.3</td>
<td>96</td>
</tr>
<tr>
<td>CMUETC</td>
<td>92</td>
<td>111</td>
<td>16590.0</td>
<td>141</td>
</tr>
<tr>
<td>aaa</td>
<td>90</td>
<td>113</td>
<td>16980.6</td>
<td>95</td>
</tr>
<tr>
<td>KuantumKoalas</td>
<td>90</td>
<td>110</td>
<td>16530.6</td>
<td>91</td>
</tr>
<tr>
<td>whatever</td>
<td>87</td>
<td>126</td>
<td>18925.4</td>
<td>139</td>
</tr>
</tbody>
</table>
Welcome to Phase 2

- Is Q1 easier? Why is the rps reduced?
- Is Q2 easier? Why is the rps reduced?
Q2 Recap

● Don’t understand encoding? See: https://piazza.com/class/hxs4b3o2lox5f9?cid=1595

● Still don’t understand encoding? Meet a TA!!!

● Question: Explain:

“q2.sample as only 499161 rows whereas 15619f14twitter-parta-aa has 500000 twweets”
Q3

- Use the ['retweeted_status']['user']['id'] and ['user']['id']

- Sort response numerically in order of user_id
- Ignore parentheses during sort

- What if user A retweeted two of user X's tweets?

- Q3 required rps is fairly high. Why?
Q4

- Hashtags, places are case sensitive
- Read the rules carefully to find location
- Ignore “location” field
- Q4 required rps is fairly low. Why?
Phase 2 Report [VERY IMPORTANT]

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

MAKE A QUANTITATIVE, DATA-DRIVEN REPORT
Live Test

- No benefit of database pre-caching
  - (unless you’re really smart)

- 30 minute warm-up
- 2 hours Q1-Q4
- 30 minutes mix-Q1Q2Q3Q4
  - Avoid bottlenecks

- Lather, rinse, repeat
Grading

- The report really matters!!
  - Dense, not long
  - What you tried and what you measured matters

- Choose which run you get graded on
  - BUT,
    - You cannot use that DB for Phase 3

- How do we decide which DB?
Suggestions / Improvements

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

Any questions?