**F15 15619 Project Phase 1 Report**

**Team Name:**

**Members (First Name, Last Name, Andrew ID):**

**Philip Lee, plee2**

**Performance Data and Configurations**

|  |  |
| --- | --- |
| Best Configuration and | Results |
| Number and type of instances | Q1:  Q2H:  Q2M: |
| Cost per hour |  |
| Queries Per Second (QPS) | INSERT HERE: (Q1,Q2H,Q2M)  score[ ]  tput [ ]  latcy [ ]  corr [ ]  error [ ] |
| Rank on the scoreboard: | Q1:  Q2H:  Q2M: |

**Rubric:**

**Each unanswered question = -5%**

**Each unsatisfactory answer = -2%**

**[Please provide an insightful, data-driven, colorful, chart/table-filled, and interesting final report. This is worth a quarter of the grade for Phase 1. Use the report as a record of your progress, and then condense it before sharing it with us. Questions ending with “Why?” need evidence (not just logic)]**

**Task 1: Front end**

**Questions**

1. Which front end framework did you use? Explain why you used this solution. [Provide a small table of special properties that this framework/platform provides]
2. Explain your choice of instance type and numbers for your front end system.
3. Explain any special configurations of your front end system.
4. Did you use an ELB for the front-end? Why, or why not? Condense your experience with ELB in the next few sentences. Talk about load-balancing in general and why it matters in the cloud.
5. Did you explore any alternatives to ELB? List a few of these alternatives. What did you finally decide to use? (if possible) Provide some graphs comparing performance between different types of systems.
6. Did you automate your front-end instance? If yes, how? If no, why not?
7. Did you use any form of monitoring on your front-end? Why or why not? If you did, show us the results.
8. What was the cost to develop the front end system?
9. What are the best reference URLs (or books) that you found for your front-end? Provide at least 3.

[Please submit the code for the frontend in your code ZIP file]

**Task 2: Back end (database)**

**Questions**

1. Describe your schema. Explain your schema design decisions. Would your design be different if you were not using this database? How many iterations did your schema design require? Also mention any other design ideas you had, and why you chose this one? Answers backed by evidence (actual test results and bar charts) will be valued highly.
2. What was the most expensive operation / biggest problem with your DB that you had to resolve for Q2? Why does this problem exist in this DB? How did you resolve it? Plot a chart showing the improvements with time.
3. Explain (briefly) **the theory** behind (at least) 3 performance optimization techniques for databases. How are each of these implemented in MySQL? How are each of these implemented in HBase? Which optimizations only exist in one type of DB? How can you simulate that optimization in the other (or if you cannot, why not)? Use your own words (paraphrase).
4. Plot a graph showing results with/without each individual optimization that you used. Extremely impressive will be a timeline of rps v/s submission id (mentioning which optimization was in use at that time).
5. Would your design work if your web service also implemented insert/update (PUT) requests? Why or why not?
6. Which API/driver did you use to connect to the backend? Why? What were the other alternatives that you tried?
7. How did you profile the backend? If not, why not? Given a typical request-response for each query (q1-q2) what percentage of the overall latency is due to:
   1. Load Generator to Load Balancer (if any, else merge with b.)
   2. Load Balancer to Web Service
   3. Parsing request
   4. Web Service to DB
   5. At DB (execution)
   6. DB to Web Service
   7. Parsing DB response
   8. Web Service to LB
   9. LB to LG

How did you measure this? A 9x2 table is one possible representation.

1. Say you are at any big tech company (Google/Facebook/Twitter/Amazon etc.). List one concrete example of an application/query where they should be using NoSQL versus one where they should be using an RDBMS. Both examples should be based on the same company (you choose).
2. What was the cost to develop your back end system?
3. What were the best resources (online or otherwise) that you found. Answer for both HBase and MySQL.

[Please submit the code for the backend in your code ZIP file]

**Task 3: ETL**

1. For each query, write about:
   1. The programming model used for the ETL job and justification
   2. The number and type of instances used and justification
   3. The spot cost for all instances used
   4. The execution time for the entire ETL process
   5. The overall cost of the ETL process
   6. The number of incomplete ETL runs before your final run
   7. Discuss difficulties encountered
   8. The size of the resulting database and reasoning
   9. The size of the backup
2. What are the most effective ways to speed up ETL?
3. Did you use EMR? Streaming or non-streaming? Which approach would be faster and why?
4. Did you use an external tool to load the data? Which one? Why?
5. Which database was easier to load (MySQL or HBase)? Why?

[Please submit the code for the ETL job in your code ZIP file]

**General Questions**

1. Would your design work as well if the quantity of data would double? What if it was 10 times larger? Why or why not?
2. Did you attempt to generate load on your own? If yes, how? And why?
3. Describe an alternative design to your system that you wish you had time to try.
4. Which was/were the toughest roadblock(s) faced in Phase 1?
5. Did you do something unique (any cool optimization/trick/hack) that you would like to share with the class?