SCC 411— Coursework

Introduction

Your coursework is to complete a Data Science project. This entails designing, implementing, and executing a successful Big Data pipeline. To do so you will be working within small groups to demonstrate the key elements and roles within the Big Data pipeline we have discussed in previous lectures.

The Project

Your team has been tasked by Google to understand the operational behaviour of one of their datacenter clusters. They would like to better understand their user and software behaviour that executes within their cluster. The engineers have done a decent job in cleaning the data up and removing most anomalies, however have provided it in a somewhat fractured manner.

The good news is that they have also provided a schema that you will have interpret from the raw data. You are also required to prepare a pitch presentation (10 minutes) and a demonstration (10 minutes). This should also include an end-to-end demonstration of the work.

The Data

The data you will be working on is a lightly modified version of operational trace data from one of Google's production clusters (12,500+ servers). The trace data provided captures a wide selection of characteristics pertaining to task resource utilization, submissions patterns, as well as software and hardware failures.

A detailed trace schema describing these many of the attributes in the data can be found at:

https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md

The data is approximately 3 GB, and can be downloaded from: https://livelancsac-my.sharepoint.com/:f:/g/personal/garragha_lancaster_ac_uk/Eq9r2Em2tEVCqyCtOHuBvSIB5-1KH4qy7S87U_zJyAhRlQ?e=rJMiD6

Question structure

Questions are subdivided into four main areas spanning data architecting, pre-processing, analytics, and visualisation. All questions are broken down into three parts: *easy-intermediate, challenge,* and *creative.* Easy-intermediate will be activities that will have a definite answer. Challenge questions will take you a longer amount of time to tackle. Hence, you are expected to only tackle one challenge per question category where there are multiple. Creative questions can fall into the category of technically difficult, something more lightweight but informative, or a combination of both.

If you are unsure about the above paragraph, kindly let us know and we can explain this to you.

Words of Wisdom

For a number of the more advanced questions, we expect you to use your own initiative/creativity to tackle problems. You will find some parts of this project difficult; a large hint we can give you is that you will likely face challenges in terms of operating a distributed infrastructure, as well as the raw data itself. Hence, there are number of methods to address precise problems; we will leave it up

to you how you interpret this. For example, it might not be wise to dive immediately into the entire data and instead work on a smaller data sub-set before tackling the entire dataset.

This is particularly true as to how you might go about approaching your project and teamwork. Questions pertaining to design and optimization, data pre-processing, analytics, and visualization will be decided by yourselves and teammates. We're not only giving you opportunities to demonstrate your technical depth, but also giving you experience and reflections on managing data projects.

Tools of the Trade
You are welcome to use programming languages and programs that you feel comfortable with (C/C++/Java, RStudio, Tableau, Python, etc.) However, this doesn't give you free reign to download libraries off the Internet to automate the entire procedure for you. Please exercise good judgement — we will appraising contributions precise contributions and knowledge.

As mentioned in lecture, we would recommend that you investigate with automating parts of your system for purposes of convenience. Be careful with this however, as if you are not careful it may lead to unintended consequences.

Better Together
These questions are not fully isolated from one another and should instead interweave into everyone's work. For example, an analytics questions could very likely be visualized in an interesting manner, or the pre-processing is used to reduce the necessary computational/storage requirements.

You should also be considering the project critical path (so that all of you are not waiting on a single team member to get Hive rebuild/restarted, or wrestling with the networking). A lot of the questions depends on conducting analysis, so would advise instead of independently working on your role, to tackle things together, or in smaller groups.

The Unexpected
You will likely find problems/activities of interest that are not explicitly mentioned in the questions below. I would recommend that you include these in some form (presentation, demo, quiz). If you are unsure about any of the questions or what might constitute as 'interesting' feel free to ask us.

For example, given most groups will have access to 10-12 VMs each, our suggestion is to first investigate building a smaller cluster, ensure that it is operating properly, and then attempt to connect all VMs.

Backup Everything. Seriously.
Hopefully you've learnt in past few weeks how complex (and sometimes fragile) there types of systems can be. Unfortunately, it gets worse as we start adding more machines together. We strongly recommend you backup, extract data and automate things as much as possible. The worst thing that could happen is if someone accidently deletes your entire HDFS that contains all your analysis results a couple days before the deadline. You can avoid it if you try to work on your physical machine and use the VM for actually conducting the larger analytics only.

Given that the VMs share the underlying infrastructure do be cautious with your disk space, while it's true each machine has 25GB, this is the physical machine, hence other groups will be using this. We will hope that you will be able to moderate yourselves accordingly, but we'll step in to mitigate if we must.

Questions

1. Architect

1.1 Create a database schema of the data to be stored within Hive. This should include an Entity Relationship Diagram, however, can include additional lightweight material if appropriate.

12 Successfully distributed the HDFS datanodes across multiple VMs.

1.3 Successfully execute a MapReduce jobs across multiple VMs.

1.4 Challenge: Make a new table schema in Hive that allows you to minimize the computational complexity of queries for your analysis that combines multiple data sources together.

1.5 Creative: Design and illustrate a high-level overview of your data pipeline which would be informative for both engineers and potential clients with technical interests.

2. Pre-processing

2.1 Identify (and omit) outliers you come across in the data — very briefly justify this decision.

2.2 The data contains string hash values for obfuscating machine architectures, applications, and users (i.e. HofLGzk1Or/8IIdj2+Lqv0UGGvY82NLoni8+J/Yy0RU=). Convert all of these into numbers.

2.3 The trace data includes number scientific notation for representing very small numerical data. Convert this to a numbered value (e.g. 7e-054 0.00007).

2.4 Challenge: The trace schema for task_resource_usage describes that software within the system is defined by a combination of the jobID and taskID attributes (i.e. these two values together identify a unique process). Create a new singular column named processID to identify unique software (instead of always referring to jobID and taskID).

2.5 Creative: Produce software that automates questions 2.2 — 2.4. You might also want to explore means of interconnecting all other parts together (i.e. automatic generation of Hive questions, or automatic visualization of database outputs).

3. Analytics

3.1 Conduct a coarse-grain analysis of the system detailing the following:

- Total number of unique users and jobs daily (note: timestamps are measured in microseconds, and the trace data start at time 0 on 19:00 EDT on Sunday May 1, 2011).
- Average number of tasks per jobs, categorized by priority in ascending order.
- Total number of task failures (i.e. all tasks that FAIL).

3.2 Challenge: Calculate the total downtime of machines within the system (i.e. the total time duration between machines leaving and joining the cluster).
OR

3.3 Challenge: Identify tasks that took 50% longer to complete their execution (from scheduling to completion) in comparison to the average task duration of their respective job.
OR

3.4 Challenge: Create a probability distribution function of user job submission patterns with appropriate statistical tests to strengthen their validity (Anderson Darling test, KS test etc.)

3.5 Creative: Anything that you feel would be particular interesting, or exciting. For example, create a multi-parameter k-means or x-means cluster to explore user resource request patterns and actual task utilization for CPU.

4. Visualisation

4.1 Visualise all answers for the coarse-grain analytics presented in 3.1, plus an additional coarse-grant analytics not mentioned above.

4.2 Challenge: Provide a temporal visualisation of the number of tasks executing on the busiest machine (i.e. the machine that executes the most tasks within the trace timespan) at 5 minutes intervals. For further challenge, sub-divide each time interval by task priority.

4.3 Creative: This activity is quite open ended, and we would like you to use your imagination Visualization that are particularly interesting and/or interactive would be particularly striking informative, or interactive are welcome. For example, visualizing heatmaps for all machines based off their utilization, or quantifying the amount of computational time lost (which could be theoretically translated into energy costs).