A Novel Cloud Based Elastic Framework for Big Data Preprocessing

Omer Dawelbeit and Rachel McCrindle
Overview

- Introduction
- Cloud based elastic framework
- Motivation
- Major Components
- Workload distribution
- Processing steps
- Experiments and results
- Discussion
- Conclusion and future work
Introduction

• Big Data is data that is too big, too fast or too hard to process using traditional tools.

• The Primary aspects of Big Data are characterized in terms of three dimensions (Volume, Variety and Velocity).

• Cloud computing is an emerging paradigm which offers resource Elasticity and Utility Billing.

• Cloud computing resources include: VMs, cloud storage and interactive analytical big data services (e.g. Google Bigquery).
Cloud Based Elastic Framework

- Entirely based on cloud computing.
- Elastic, hence able to dynamically scale up or down.
- Extendible, such that tasks can be added or removed.
- Tracks the overall cost incurred by the processing activities.
- Capable of both preprocessing and analyzing Big Data.

**Big Data processing pipeline**

- Data collection
- Data curation
- Data integration and aggregation
- Data storage using Cloud Storage and BigQuery
- Data analysis and interpretation using BigQuery
Motivation

• Analytical big data services can analyze massive datasets in seconds (e.g. 1 terabyte in 50s).
• Can handle the analysis and storage of textual based structured and semi-structured big data.
• Data curation, transformation and normalization can be handled using an entirely parallel approach.
• Some tasks do not naturally fit the MapReduce paradigm (map/reduce, task chaining, complex logic, data streaming).
• Frameworks such as Hadoop utilizes a fixed number of computing nodes during processing.
• Cloud computing elasticity can be utilized to scale up and down VMs as needed.
Major Components

- Coordinator VM.
- Processor VMs.
- Processor VM Disk Image
- Job/Work description.
- Processing program and tasks.
- Workload Distribution function.
- Cloud storage.
- Analytical big data service.
- Program input via VM metadata.
Workload Distribution

• Task processing is entirely parallel, so processors do not need to communicate with each other.

• Work is distributed using bin packing to ensure each processor is fairly loaded.

• Items to partition can be files to process or analytical queries to run against Bigquery.
Coordinator

- Receives requests for work.
- Monitors nodes once a task is done, another task is started.
- Starts the required number of nodes, supplying tasks as metadata.
- Partitions the work using bin-packing algorithm.
- Terminates them once the work is done.
- The overall cost of resource usage is tracked.

Metadata Server

- Wait for next task
- Set metadata as FREE
- Complete task, and upload/update data
- Set metadata as BUSY
- Read Metadata
- Choose task to execute
- Read input data from Cloud Storage or Bigquery
- Read startup script

Processor

- Choose task to execute
- Set metadata as BUSY
- Read input data from Cloud Storage or Bigquery
- Complete task, and upload/update data
- Set metadata as FREE
- Wait for next task

Bigquery/Cloud Storage
Experiment

- Experiment conducted on the Google Cloud Platform:
  - Compute Engine: Up to 10 processors of type n1-standard2 VMs each with 2 virtual cores, 10 GB disk and 7.5 GB of main memory.
  - Cloud Storage

- DBpedia* dataset is used:
  - Structured extract from Wikipedia
  - Contains 300 Million statements
  - Total size is 50.19 GB
  - Compressed size is 5.3GB
  - Data is in NTriple RDF format:
    <http://dbpedia.org/resource/AccessibleComputing>
    <http://xmlns.com/foaf/0.1/isPrimaryTopicOf>

* [http://wiki.dbpedia.org/Datasets](http://wiki.dbpedia.org/Datasets)
Results

- Num. of processor VMs vs. Runtime (min)
- Runtime (min) vs. Un-compressed file size in GB
- Cost ($) vs. Runtime (min)
Discussion

• Preprocessed 50GB of data in 11 minutes using 8 VMs.
• For our data, the processing is CPU bound (80% processing, 20% I/O).
• Processing time is proportional to the size of the data assigned to the VM.
• The overall runtime is constraint by the time required to process the largest file.
• Input files can be split further to enable equal workload allocation.
• Only 9% to 20% of the overall runtime is spent in transferring the files to and form cloud storage.
Conclusion and future work

• We have developed a novel cloud based framework for Big Data preprocessing.
• Our framework is lightweight, elastic and extendible.
• Makes use of cloud storage and analytical big data services to provide a complete pipeline for big data processing.
• We have extended the processing to executing analytical queries against Bigquery.
• We plan to use the framework for processing social media datasets.
• The implementation for our framework is open source and can be downloaded from [http://ecarf.io](http://ecarf.io)
Thank you

- Any questions?