Interactive Analytical Processing in Big Data Systems, BDGS: A Scalable Big Data Generator Suite in Big Data Benchmarking, Study about DataSet

Aayush Agrawal

May 23, 2014
Outline

1. Interactive Analytical Processing in Big Data Systems
2. Towards a BigData BenchMark
3. BDGS: A Scalable Big Data Generator Suite
4. Methodology of BDGS:
5. Synthetic Data Generator:
6. References
Introduction:

MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster.

LOGICAL VIEW:

- **Map Function:** Map takes one pair of data with a type in one data domain, and returns a list of pairs in a different domain:
  \[
  \text{Map}(k_1, v_1) \rightarrow \text{list}(k_2, v_2)
  \]

- **Reduce Function:** It produces a collection of values in the same domain:
  \[
  \text{Reduce}(k_2, \text{list}(v_2)) \rightarrow \text{list}(v_3)
  \]
Key findings during the study:

1. There is a new class of MapReduce workloads for interactive, semi-streaming analysis that differs considerably from the original MapReduce use case targeting purely batch computations.

2. There is a wide range of behavior within this workload class, such that we must exercise caution in regarding any aspect of workload dynamics as "typical".

3. Query-like programatic frameworks on top of MapReduce such as Hive and Pig make up a considerable fraction of activity in all workloads we analyze.

4. Some prior assumptions about MapReduce such as uniform data access, regular diurnal patterns, and prevalence of large jobs no longer hold.
By Job Data Sizes: Most jobs have input, shuffle, and output sizes in the MB to GB range. Thus, benchmarks of TB and above captures only a narrow set of input, shuffle, and output patterns.

Fig: Datasize for each workload [?]
Skews in Data Frequency: The observation challenges the design assumption in HDFS that all data sets should be treated equally, i.e., stored on the same medium, with the same data replication policies.

Fig: Zipf distribution of all workloads [?]
Access Temporal Locality: A possible cache eviction policy is to evict entire files that have not been accessed for longer than a workload specific threshold duration.

Fig: Data re-access interval [?]
Workload Variation With Time:

1. Weekly Time Series:

Fig: Weekly time graph for workload CC-a (utilization slot)
Workload Variation With Time:

- Burstiness: Peak to average ratio is a common way to measure burstiness. They extend the concept of peak-to-average ratio to quantify burstiness.

Fig: Weekly time graph for workload CC-a (utilization slot)
Time Series Correlation: They compute three correlation values: between the time varying vectors \( \text{jobsSubmitted}(t) \) and \( \text{dataSizeBytes}(t) \), between \( \text{jobsSubmitted}(t) \) and \( \text{computeTimeTaskSeconds}(t) \), and between \( \text{dataSizeBytes}(t) \) and \( \text{computeTimeTaskSeconds}(t) \), where \( t \) represents time in hourly granularity, and ranges over the entire trace duration.

Fig: Correlation between diff. time series [?]
Towards a BigData BenchMark:

Some challenges associated with building a TPC-style benchmark for MapReduce and other big data systems.

- Data Generation: A good benchmark should stress the system with realistic conditions in all areas like data access pattern, skews in frequency.
- Mixing MapReduce and query-like frameworks: The heavy use of query-like frameworks on top of MapReduce indicates that future cluster management systems need to efficiently multiplex jobs.
- Empirical models: It is necessary for a benchmark to assume an empirical model of workloads.
- Workload Suites: If not a single set is able to represent a workload then we have to make a small suite of workload class that create a range of workload.
Introduction:

Data explosion is an irresistible trend that data are generated faster than ever. IDC forecasts this speed of data generation will continue and it is expected to increase at an exponential level over the next decade. The challenges of capturing, storing, indexing, searching, transferring, analyzing, and displaying Big Data bring fast development of big data systems. In this paper, we introduce Big Data Generator Suite (BDGS), a comprehensive tool developed to generate synthetic big data while preserving 4V (volume, variety, velocity, veracity) properties.
Requirements:

Big data generators should scale up or down a synthetic data set (volume) of different types (variety) under a controllable generation rate (velocity) while keeping important characteristics of raw data (veracity).

- **Volume:** Big data generators should be able to generate data whose volume ranges from GB to PB, and can also scale up and down data volumes to meet different testing requirements.

- **Velocity:** Big data applications can be divided into in three types: online analytic, online service and realtime analytic. Data processing speed and data updating frequency should be maintained.
Requirements:

- **Variety**: Since big data come from various workloads and systems, big data generators should support a diversity of data types (structured, semi-structured and unstructured) and sources (table, text, graph, etc).

- **Veracity**: In synthetic data generation, the important characteristics of raw data must be preserved.
Methodology of BDGS:

It represents an overview of BDGS that is implemented as a component of our big data benchmark suite BigDataBench.

Fig: Architecture of BDGS [?]
Methodology of BDGS:

BDGS can generate synthetic data while preserving the important characteristics of real data sets and can also rapidly scale data meeting the requirements.

- Text Generator
- Graph Generator
- Table Generator
Text Generator:

It applies latent dirichlet allocation (LDA) as the text data generation model. This model can keep the veracity of topic distributions in documents as well as the word distribution under each topic.

Fig: Text Generator [?]
A graph consists of nodes and edges that connect nodes. For the graph data sets, namely Facebook Social Graph and Google Web Graph, we use the kronecker graph model in our graph generator.
E-commerce transaction and personal resumes come under the table dataset. PDGF uses XML configuration files for data description and distribution, thereby simplying the generation of different distributions of specified data sets. The process of resume generator:

- Randomly generate a string as the name of a resume.
- Randomly choose fields from email, telephone, address, date of birth, home place, institute, title, research interest, education experience, work experience and publications, where each field follows the bernoulli probability distribution.
- For each field: if the field has sub fields, then randomly choose its sub fields following the bernoulli probability distribution. else assign the content of the field using the multinomial probability distribution.