The size of astronomical observational data is increasing yearly. For example, while Atacama Large Millimeter/submillimeter Array is expected to generate 200 TB raw data every year, Large Synoptic Survey Telescope is estimated to produce 15 TB raw data every night. Since the increasing rate of computing is much lower than that of astronomical data, to provide high performance cloud computing (HPC) resources together with scientific data will be common in the next decade. However, the installation and maintenance costs of a HPC system can be burdensome for the provider. I note public cloud computing for an alternative way to get sufficient computing resources inexpensively. I build Hadoop and Hive clusters by utilizing a virtual private server (VPS) service and Amazon Elastic MapReduce (EMR), and measure their performances. The VPS cluster behaves differently day by day, while the EMR clusters are relatively stable. Since partitioning is essential for Hive, several partitioning algorithms are evaluated. In this poster, I report the results of the benchmarks and the performance optimizations in cloud computing environment.

1. Introduction

- The size of astronomical observational data is getting bigger and bigger year by year.
- High Performance Computing (HPC) resources are required to process such massive data.
- A HPC system is too expensive and also requires a large physical space.
- Large project teams can afford it, and can lend those computing resources together with their data.
- Should small project teams and individual persons give up sharing their relatively big data and knowledge with astronomical communities?
- Why not cloud computing?
- It is available at low cost.
- We need not take care of the hardware and its physical configuration.
- Is cloud computing applicable to astronomical data?

2. Hadoop and Hive

- Hadoop is an open-source software framework for distributed file systems and data processing.
- Hadoop is designed to run on a cluster of standard PCs, where hardware failures are much more common than well-maintained HPC systems; automatic error recovery functionality (re-execute of failed jobs on other nodes) is implemented.
- Hive is a SQL-like distributed database system running on Hadoop clusters. A Hive database essentially consists of a large number of or huge text files.
- Hive and Hadoop are designed to process a set of moderate large files in parallel.
- A massive number of small files exhaust resources to manage their metadata on HDFS, the native distributed file system of Hadoop.
- To the contrary, a small number of huge files are inefficient since they lead to intensive file I/O and data transportation between nodes over the network.
- A small degree of parallelism makes the time required to schedule at the initialization stage much longer.
- We need to divide datasets into pieces of appropriate size.
- Different from standard RDBMSs, Hive does not manage datasets by indexes.
- Instead, datasets can be organized by "partitions".
- On the HDFS level, partitions correspond to directories.
- Partitions seem to be one of keys of a table.
- Files to be read (processed) in a query are narrowed down by specifying the values of the partitions.

3. What are the problems?

- Hadoop and Hive are designed to process a set of moderate large files in parallel.
- A massive number of small files exhaust resources to manage their metadata on HDFS, the native distributed file system of Hadoop.
- To the contrary, a small number of huge files are inefficient since they lead to intensive file I/O and data transportation between nodes over the network.
- A small degree of parallelism makes the time required to schedule at the initialization stage much longer.
- We need to divide datasets into pieces of appropriate size.
- Different from standard RDBMSs, Hive does not manage datasets by indexes.
- Instead, datasets can be organized by "partitions".
- On the HDFS level, partitions correspond to directories.
- Partitions seem to be one of keys of a table.
- Files to be read (processed) in a query are narrowed down by specifying the values of the partitions.

4. Benchmark Strategies

- Data files for 2MASS Catalog Server Kit² (196 GB, 470,992,970 rows) are used as test data.
- Two columns are appended:
  - healpix_id: HEALPix ID with \(N_{side} = 2^5\) of the source position.
  - healpix_partition: HEALPix ID with \(N_{side} = 2^5, 2^4, \ldots, 2^1\) of the source position to calculate the partition ID.
- A benchmark program is written in Java.
  1. Querying positions and search radius (5'-5") are randomly chosen with uniform distributions by Mersenne Twister. The random seed is fixed at a certain value.
  2. The range of healpix_partition is specified by the HEALPix library.
  3. The angular distances are calculated for the rows in healpix_partition based on Yamachi (2011).
  4. The average magnitudes of J, H, K-bands are calculated with the built-in function AVG() in Hive for the sources within the given search radius.
- Above mimic an use case to cut out desired data cubes from 3-dimensional high resolution all-sky images.

5. Results of VPS

- I setup a Hadoop/Hive cluster consisting of 8 nodes by utilizing "Small Plan" provided by GMO CLOUD K.K., a Japanese company.
- There is one NameNode (master node) and 7 DataNodes (slave nodes).
- Pure Apache Hadoop and Hive distributions are used, i.e., Tez is not installed.
- The construction of a partition tree is performed on a workstation due to the limitation of the memory size of VPS, then the partitions are transported to the nodes.
- Figure 1 represents the distributions of searching time with \(N_{side}\) of 2\(^k\) measured on different days.
- The performance of the cluster differs by day.

6. Results of Amazon EMR

- I use Amazon EMR for an IaaS solution.
- Clusters with 1 NameNode and 3 DataNodes are created, each of which has 4 CPU cores and 15 GB RAM and costs $0.385 an hour.
- A cluster is created every time when a new set of the benchmark is started.
- Physical files of the database are stored on Amazon S3, which is a cloud storage service, since files on HDFS are lost when the cluster is terminated.
- VM instances and database files locate in the Tokyo region.
- Tez is enabled.
- Figure 2 shows the distributions of searching time with \(N_{side}\) of 2\(^k\) executed on different cluster instances, suggesting that no difference is observed.
- Figure 3 represents the dependence of mean searching time on \(N_{side}\). As \(N_{side}\) increases, the mean searching time decreases. \(N_{side} \geq 2^3\) are measured due to insufficient memory.
- A typical query is distributed to only 3 nodes since the range of healpix_partition is \(\leq 3\) for \(N_{side} \geq 2^3\).
- A wide range of healpix_partition makes the time required to schedule at the initialization stage much longer.

7. Nested Partitions

- The introduction of healpix_id modulo 16 (\(=\text{healpix}_\text{mod}\)) into the \(N_{side}\) and partitioning the dataset into \((=\text{healpix}_\text{part})\) reduce an effective file size of one partition to that in the \(N_{side}\) case.
- So this approach is expected to make the searching time in the \(N_{side}\) case same as that in the \(N_{side}\) case.
- But an application of this method to EMR does not change the searching time at all.

8. Future Work

- Identification of parameters determining the degree of parallelism
- Checking if a larger number of partitions are possible

References

- Yamachi, C. 2011, PASP, 123, 1234