Noname manuscript No.
(will be inserted by the editor)

Small Compute Clusters for Large-Scale Data Analysis

Nathan Regola · **Nitesh V. Chawla**

Received: date / Accepted: date

Abstract Enterprises of all sizes are dealing with an increasing amount of electronic data that is essential to their business operations. Since there are many options when moving beyond a single server, organizations must consider a range of investment options such as purchasing a small cluster of machines or leasing capacity from cloud providers.

Given the complexity (and cost) of configuring larger scale systems (such as distributed databases), we believe that revisiting more generic batch computing systems provides several advantages that are worth considering and benchmarking. For example, leveraging low cost servers, avoiding the data transformation process, and the time necessary to load and move data to a database are important considerations when selecting a workflow or technology. This paper presents benchmark results that compare MySQL to basic Unix tools on Condor and Oracle Grid Engine (OGE) supported by distributed filesystems for high throughput access to data using a real world data set. The benchmark results should be largely generalizable to other business analytic tasks. These results should aid information technology (IT) managers facing difficult decisions on which software stack to utilize for large, dynamic, datasets where database setup and loading may not be feasible within the time constraints imposed by business needs.

Keywords Data Analytics · Cloud Computing · Performance · Business Intelligence and Knowledge Management · Distributed Storage

Nathan Regola · Nitesh V. Chawla
Department of Computer Science and
Interdisciplinary Center for Network Science and Applications
The University of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556
Tel.: 1-574-631-7095
Fax: 1-574-631-9260
E-mail: {nregola, nchawla}@nd.edu

1 Introduction

Modern organizations increasingly leverage vast amounts of electronic data for decision making. These efforts span the collection, management, and analysis of data. This is occurring because hard drive capacity has been growing exponentially from 1980 to the present, while simultaneously reducing the cost to store a gigabyte of data. These two trends have enabled enterprises to store all types of business data at an even lower cost. For example, according to news reports, Wal-Mart's original data warehouse was 320GB, grew to 500TB in 2004, and in 2007 was around 4PB, which is over 13,000 times the size of the original data warehouse. With this surge in enterprise data volume, smaller organizations with significant data storage and analysis needs are forced to select tools and technologies from an ever increasing number of vendors. This problem is exacerbated when their needs grow beyond a single server. However, before leasing cloud compute instances or building a local cluster, it is important for any potential customers to know which software stack is appropriate in order to run the cluster cost effectively and return results as fast as possible.

One industrial partner of our research lab at the University of Notre Dame is a small to mid-sized business with a single server data warehouse that is used to make real-time pricing decisions and support a small team of analysts. Since there are many options when moving beyond a single server, they must consider a range of investment options such as purchasing a small cluster or leasing capacity from cloud providers. However, before leasing cloud compute instances, it is important for any potential customers to know which software stack is appropriate in order to run the cluster cost effectively and return results as fast as possible. The partner's dataset consists of sales data (several TB) and the associated lookup tables to track common products.

Given that many small and medium businesses may not have the expertise to select the correct software stack or simply cannot afford expensive distributed databases, we present a performance comparison of data manipulation methods on a small cluster, comparing them to a MySQL database. Obviously, utilizing simple Unix tools, a distributed file system, and a batch computing environment is only one possible approach to manipulating large datasets and we plan to compare this approach to other systems in later work.

Given the complexity of configuring larger scale systems, such as distributed databases, revisiting more generic batch computing systems provides several advantages that are worth considering and benchmarking. First, many businesses may want to leverage low cost commodity tools and hardware. Secondly, datasets may not always lend themselves to the strict conformance of SQL tables. If the dataset is not already in a database, then a table schema must be defined, primary and foreign keys must be created, security roles must be defined, etc. Thirdly, the data must be loaded. In the case of this dataset, it took us almost one week to restore the database backups so that we could access the entire dataset. All of these steps consume valuable time that could be used for analysis. If a dataset exists solely to perform basic analysis, or for correlation with other datasets (for example, to match customers to data sources such as Facebook, Twitter, etc.) then the analyst may be able to simply copy the dataset to distributed storage and begin computation, using shell scripts for analytic purposes. This approach may be necessary if the dataset would become stale by the time that the database load is complete.

We present benchmark results that compare MySQL to basic Unix tools on a Condor[5] and OGE[8] (formerly SGE) cluster supported by distributed filesystems for high throughput access to data. The benchmark results should be largely generalizable to other business analytic tasks and should aid IT managers facing difficult decisions on which software stack to utilize for large, dynamic datasets where database setup and loading may not be feasible within the time constraints imposed by business needs. Future work will add other technologies such as MapReduce/Hive and distributed databases such as Greenplum to determine which technology is able to process the dataset most efficiently (from the data load phase though the analysis phase), considering the system's usability from an analyst's perspective and the cost of the system.

2 Method

The experimental setup used transactional data for the tests (all of the transactional data shares the same attributes and represents over 90% of the dataset). For the tests of the MySQL database, we scripted queries and measured the time to query the transactional data and output results. We exported the same source data that the MySQL database relies on to text files, and then placed this data on two distributed file systems, a Panasas [9] 700 series storage system and Chirp [10]. Figure 1 shows a simplified view of the network in our campus's high performance computing center.

Panasas is a distributed, high performance, network based storage system that services our campus compute cluster. Each compute node is able to access this storage system over Gigabit Ethernet. Since the storage is shared between hundreds of compute nodes, the storage performance may vary over time (depending on the load). To compensate for the variability in load on the storage system, we repeated the experiments thirty times and present the average results[1].

Chirp is an open source, distributed file system that supports strong access controls, is simple to deploy, requires no special privileges, and can be attached transparently to ordinary Unix applications. A recent addition to Chirp [3] enables Chirp to be used as a front end to access Hadoop file systems, while providing strong access control for datasets. This is an important contribution, because basic Hadoop file systems do not support strong authentication and authorization. On our campus, one such file system is approximately 158TB, allowing us to benchmark this for use as distributed storage for Condor jobs. This type of approach could be attractive to many organizations, as Condor uses idle CPU cycles from workstations and dedicated Chirp/Hadoop servers can be built at relatively low cost compared to high end storage systems such as Panasas. One particularly useful feature of Chirp is the ability to use tickets for distributed access to data. These tickets allow a user to generate tickets with their identity and send them with compute jobs for access to data, without allowing the entire compute cluster to access restricted data. We found this quite powerful, as we did not have to use complex authentication and authorization mechanisms to restrict access to data, such as Kerberos [7] or Globus [2].

Twenty-five files, representing approximately 124GB of transactional data, were selected for testing on the various systems since twenty-five concurrent tasks can usually be completed in a reasonable amount of time on both Condor and OGE/Panasas. We generally submitted jobs to the scheduler and did not perform extensive optimizations in order to make a comparison between "typical" performance on each system, since these batch computing systems are almost always shared between many users. We worked with the developers of the Hadoop addition to Chirp to fix several bugs that we encountered, and we were unable to run 86 tasks concurrently when we began testing. We expect that our campus Hadoop cluster will be upgraded in the near future with higher performance networking, allowing us to perform higher scalability testing.

[1] The distribution is non-Gaussian.

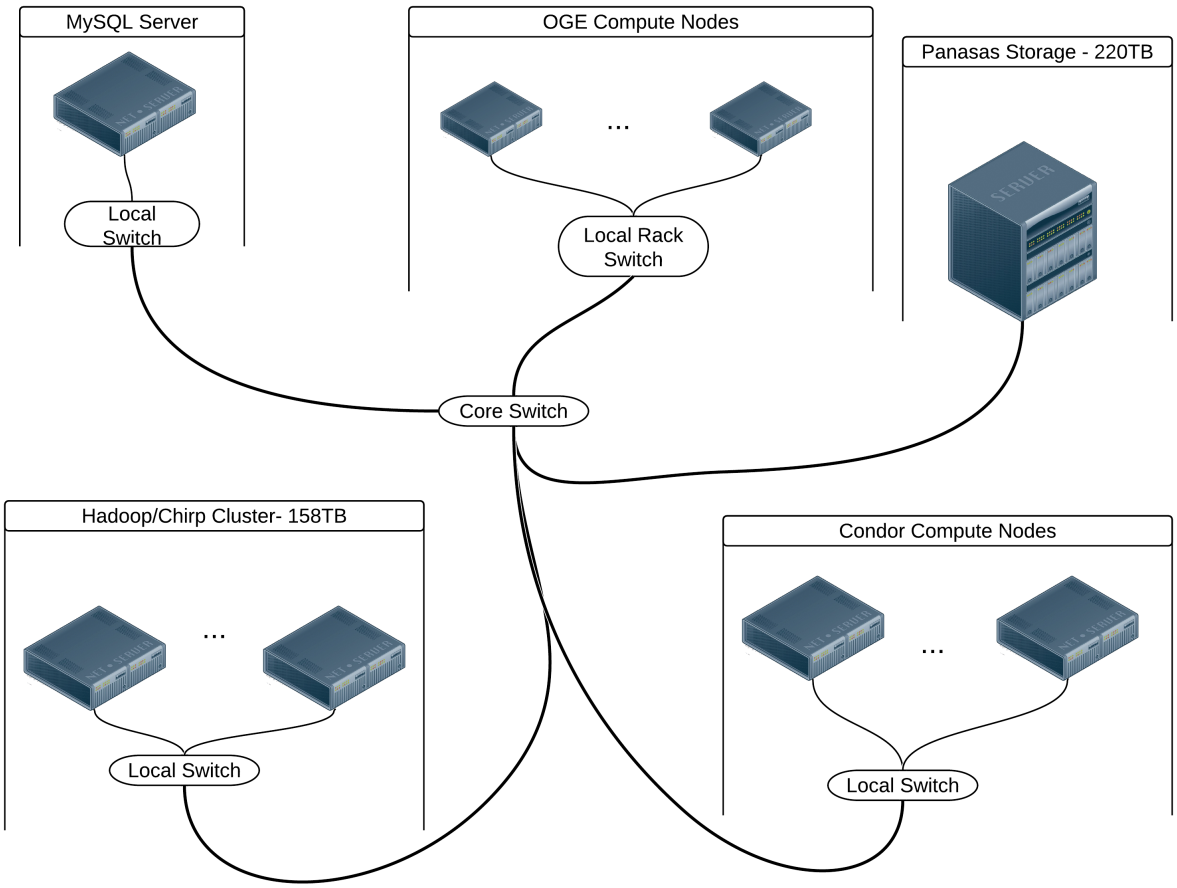

Fig. 1 Experimental setup for the benchmarks, showing the placement of OGE compute nodes, Condor nodes, Chirp/Hadoop distributed storage, the Panasas storage system, and the MySQL server. Thin lines represent 1 gigabit Ethernet connections. Thicker lines represent higher bandwidth connections (10Gbps+). Servers are typically connected to a local switch (usually in the same rack), and this local switch is connected to the core switch. The Panasas system on our campus has over 70Gbps of bandwidth available to the core switch.

System	Runs	Notes	Elapsed Time	Result Time
MySQL	3	SQL Query, restart between each run	(1 node) 348.9s	348.94s
MySQL hardware/Panasas	3	Unix Tools, single job	(1 node) 4345.6s	4345.6s
OGE/Panasas	30	Unix Tools, multiple jobs per node	(25 jobs) 21402.33s	856.0s
OGE/Panasas	30	Unix Tools, single job per node	(25 jobs) 17896.56s	715.8s
Condor/Chirp	10	Unix Tools, multiple jobs per node	(25 jobs) 28760s	1150.4s

Table 1 Total computation time for a basic query in MySQL and using Unix tools on the MySQL server hardware (using the same Panasas storage as OGE), OGE/Panasas and Chirp/Condor. MySQL does not need to read the entire table to retrieve specific columns. The minimum elapsed time for an OGE/Panasas job was 5228.42s and the maximum time was 21212.62s, over thirty runs (achieved using a single job per node). The minimum elapsed computation time for a Condor job was 12894s and the maximum was 44143.49s over ten runs. The server with MySQL hardware is higher performance hardware than the OGE compute nodes (OGE/Panasas), explaining the faster performance running the same job from the same storage system.

3 Results

Table 1 shows the compute times to select two columns of data from a large dataset on various systems. While it is faster to reserve eight cores, and use one to two cores for computation (OGE/Panasas with a single job per node) in order to maximize the network bandwidth that is available to read the file from network storage, it is usually not cost effective. This is the case because idling at least six cores wastes hardware and power since they are performing no

useful work. The combination of OGE/Panasas with multiple jobs per node provided reasonable performance while cost effectively leveraging the available hardware. Further optimizations to the scheduling of data intensive tasks may yield increased performance with Condor and Chirp.

If immediate results are needed, for example to provide input for realtime analytic processes, then a dedicated cluster will likely provide faster response time. In our experience, if compute capacity was available, OGE began executing the workload within one or two seconds, while Condor typically requires a small startup time before jobs begin to execute (approximately thirty seconds). Also, since other users may have priority on machines where Condor jobs execute, jobs may be kicked off and have to restart elsewhere. If realtime capability is required, then a dedicted OGE cluster will likely provide the fastest reponse time. If Condor is utilized, then priority should be given to users that require fast response times, and a dedicated cluster should be used to prevent jobs from being vacated by the machine owner.

The selection of a batch computing system (OGE or Condor) does not dictate that one must use a specific storage system. For example, it would be entirely possible to use Chirp/Hadoop file system nodes to support OGE jobs, or to use Panasas to support Condor jobs. We will likely explore these possibilities in later work. If high performance is the goal, then the Hadoop/Chirp could be configured to provide high performance. In our current environment, the Hadoop/Chirp file system is not configured for extreme performance while the Panasas system is, by design, a high performance storage appliance.

4 Related Work

Data mining and systems researchers have attempted to take advantage of new paradigms and hardware advances in order to analyze increasingly larger datasets. Researchers have been evaluating cloud computing as a modern approach to quickly analyze large datasets for some time. For example, Moretti et al.[6] examined the use of Chirp and Condor to build ensembles of classifiers for a large scale distributed environment.

Grossman et al.[4] built a distributed system composed of compute (named Sphere) and storage (named Sector) components that is optimized for performance over high bandwidth wide area networks. Sphere and Sector are able to outperform Hadoop in various benchmarks.

Bennett et al.[1] introduced a data intensive benchmark for cloud compute platforms and an accompanying workload generator as a follow up to Grossman et al.[4]. Bennett et al.[1] argues that Hadoop and MapReduce are designed for local data centers with known server locations. However, multiple cloud compute instances are not guaranteed to be in the same rack and systems such as Hadoop are designed with the assumption that the relative rack location of servers is known (in order to optimize data locality). They also show that in cloud environments, other paradigms are able to outperform MapReduce. Our benchmarks are an evaluation of basic Unix tools that utilize a variety of systems.

5 Conclusion

The paper presents a comparison of MySQL with Condor/Chirp and OGE/Panasas. OGE/Panasas provides the best performance because it has higher performance, yet more expensive storage. Condor/Chirp provides a lower cost alternative that uses scavenged CPU cycles and on average takes 57% longer to complete jobs in our environment. Future work will add distributed databases and MapReduce jobs to the environment in order to evaluate these systems for our workload. Eventually, we plan to test the highest performing systems in virtual environments since cloud providers utilize virtual machines to provide leased compute nodes.

References

1. Bennett, C., Grossman, R.L., Locke, D., Seidman, J., Vejcik, S.: Malstone: towards a benchmark for analytics on large data clouds. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10, pp. 145–152. ACM, New York, NY, USA (2010). DOI http://doi.acm.org/10.1145/1835804.1835826
2. Butler, R., Welch, V., Engert, D., Foster, I., Tuecke, S., Volmer, J., Kesselman, C.: A national-scale authentication infrastructure. Computer **33**(12), 60 – 66 (2000). DOI 10.1109/2.889094
3. Donnelly, P., Bui, P., Thain, D.: Attaching cloud storage to a campus grid using parrot, chirp, and hadoop. In: Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, pp. 488–495 (2010). DOI 10.1109/CloudCom.2010.74
4. Grossman, R., Gu, Y.: Data mining using high performance data clouds: experimental studies using sector and sphere. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08, pp. 920–927. ACM, New York, NY, USA (2008). DOI http://doi.acm.org/10.1145/1401890.1402000
5. Litzkow, M., Livny, M., Mutka, M.: Condor - a hunter of idle workstations. In: Proceedings of the 8th International Conference of Distributed Computing Systems (1988)
6. Moretti, C., Steinhaeuser, K., Thain, D., Chawla, N.V.: Scaling up Classifiers to Cloud Computers. In: Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on, pp. 472–481 (2008). DOI 10.1109/ICDM.2008.99
7. Neuman, B., Ts'o, T.: Kerberos: an authentication service for computer networks. Communications Magazine, IEEE **32**(9), 33 –38 (1994). DOI 10.1109/35.312841
8. Oracle: http://www.oracle.com/technetwork/oem/grid-engine-166852.html
9. Panasas: http://www.panasas.com
10. Thain, D., Moretti, C., Hemmes, J.: Chirp: a practical global filesystem for cluster and grid computing. Journal of Grid Computing **7**, 51–72 (2009). DOI 10.1007/s10723-008-9100-5