Performance Evaluation of Distributed Database on PC Cluster Computers using MySQL Cluster

Sorapak Pukdesree, Vitalwonhyo Lacharoj, Parinya Sirisang

Abstract— **MySQL database is one of the most deployed DBMSs in both personnel and businesses. MySQL database can be deployed as personnel database, small and medium business and including enterprise organizations. MySQL database also has a version of distributed processing and distributed data, named MySQL Cluster. MySQL Cluster intends to be used in higher performance processing and can be scaled capability of capacity. The objective of this research was to evaluate the performance of scalability of MySQL Cluster on PC Cluster computers. We had used Benchw benchmark as the performance evaluating tool. The results of the research showed the relations between time of processing of each query of Benchw benchmark and the scalability of number of storage nodes. The results also showed that increasing the number storage nodes trended to improve processing time of each query of Benchw benchmark. Therefore the research is advantages for personnel or businesses that can plan the capability of database system to meet their business requirements. Furthermore, we had deployed MySQL Cluster on personnel computers that can reduce cost of hardware budgets dynamically and also can reduce cost of software license significantly.**

***Index Terms*— High Performance Computing, Cluster Computers, Database, MySQL Cluster, Distributed Database.**

I. Introduction

Most of small or medium DBMSs store and manage their information on centralization approach as shown in Figure 1. Normally, the information can be shared by many of users. Users can manipulate information by making requests to the centralized database server [2, 3]. All users' requests are handled by only one database server then the results will send back to the users. This approach is easy to manage and maintain for database management administrator. Otherwise if there are higher volumes of requests, the only one database server may not handle those requests or the results of each request may be waited for a long period of time. To handle this situation there may have to spend high budget for a high performance database server that have higher processing capacity, more memory, more advanced I/O and also more high speed network. Furthermore they have to spend on the cost of proprietary software license.

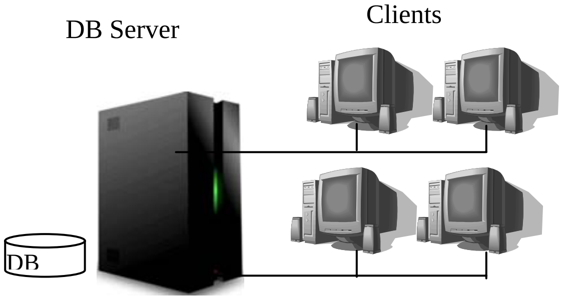

Figure 1: Typical Database Management

Another approach to improve the performance of database server is distributed processing [2, 3]. In this approach there are more than one processing database servers that each server handles those requests independently in parallel. Normally the information is stored on shared storage which is accessed and manipulated by each database servers depend on users' requests. But in this approach organizations will have to spend expensive cost of the shared storage such as dedicated SAN storage technology, several high performance servers, high cost of software license and so forth. This approach is not only widely used in commercial DBMSs such as Oracle database server or MS SQL server but also be used in some open source DBMSs such as MySQL server.

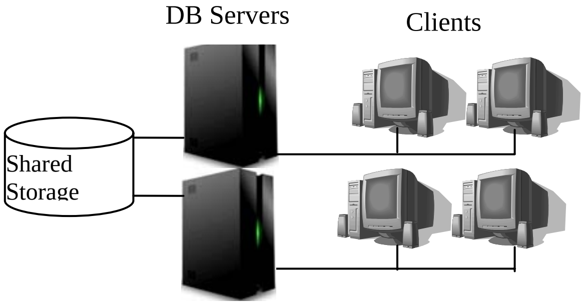

Figure 2: Distributed Database Processing using shared storage

The other approaches such as distributed data environment [2, 3] that information will be decomposed into small pieces then distribute them to be stored on several data

Manuscript received April 21, 2010. This work was supported in part by Bangkok University and 2b-cert Company.

Sorapak Pukdesree is a chairperson of Software Engineering, School of Computer Science and Technology, Bangkok University, Thailand. (e-mail: sorapak@gmail.com).

Vitalwonhyo Lacharoj is a student of Computer Science, School of Computer Science and Technology, Bangkok University, Thailand. (e-mail: vitalwonhyo@gmail.com).

Parinya Sirisang received a Bachelor degree in Computer Science, School of Computer Science and Technology, Bangkok University, Thailand. Presently he is working as Network (e-mail: jnosis.s@hotmail.com).

storage nodes in the cluster computers system. Those cluster computers are connected together as a high performance computer. The distributed DBMSs will handle those pieces of data as one unit. Users do not need to know where the exactly data will be stored on cluster nodes. There are several advantages of this approach such as higher processing performance, more memory capacity, higher network bandwidth and also higher I/O bandwidth. In this approach there are several storage nodes that each storage node will handle their own pieces of data in parallel. Furthermore, each of them has its own network interface card the handles network bandwidth. The best case of network bandwidth can improve the system by the product of number of storage nodes and bandwidth of each network interface. Finally this approach can be deployed on PC computer clusters that can reduce the budget of the system significantly. This research will implement the distributed data using MySQL Cluster 7.0 as DBMSs which the latest version of MySQL Cluster.

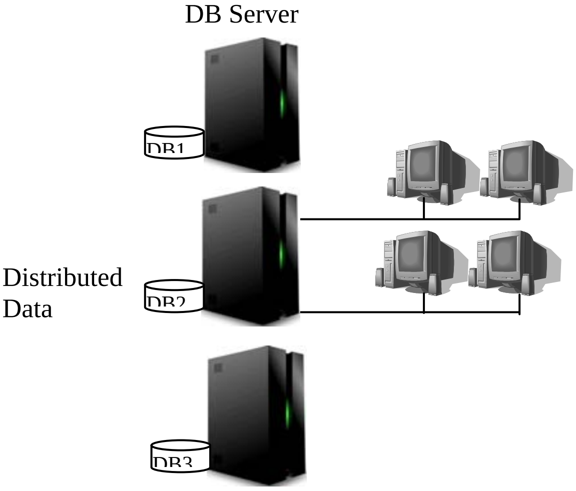

Figure 3: Distributed Data and distributed processing.

II. Hardware Implementation

In this research, we have used ten PC computers that are in the computer laboratory as shown in Figure 4. The specifications of each computer are a single Intel Core 2 Duo E6400, 2 GB-667 MHz of RAM, 250 GB-SATA II of hard drive and on boarded 1000 Mbps of network interface as shown in Figure 4. All of computers is connected using two 1000 Mbps eight-port switching hubs. The system is a closed system that prevents other factors that may affect the results of the experimental. Even though these computers do not have high performance as new PC computers but this is our environment. In the distributed data approach, the network bandwidth is very importance factor that will affect the results of the testing experimental. Presently the 1000 Mbps is standard and widely use in most PC computers and also wired cable is CAT5e that has maximum 350 Mbps of network capacity. In the next research we may use CAT6, CAT6e or CAT7 that has 550 Mbps or 1000Mbps to improve the network capacity. Unfortunately we do not have planned to use very high speed networks as Myrinet or Infiniband network system.

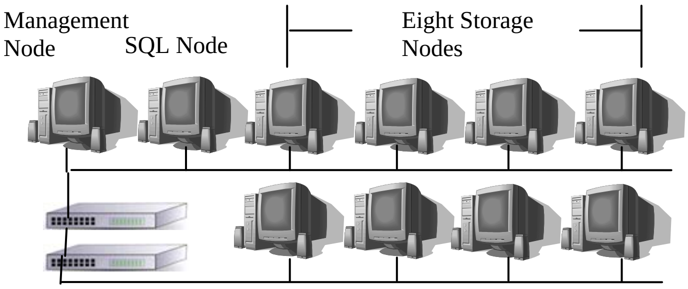

Figure 4: System design in this research.

III. Software Implementation

We decided to use Red Hat Enterprise Linux 5 [5] for our operating system. RHEL5 is one of the most reliable, potential, stable, secure and so forth. We do not test the system on others open source operating system such as fedora or FreeBSD. Because in our assumption, we are going to deploy the system in SMEs businesses that they do not want to get risk or unreliable situations. Therefore we should not deploy unstable operating system. RHEL5 also provides support for their customers via subscription and also be updated patches or packages via internet that can very useful for system administrator to fix or upgrade the system software.

In this research we use MySQL Cluster 7.0 [7] as our distributed DBMS which is the latest version at this time. MySQL Cluster 7.0 provides many advance features as enterprise DBMSs such as HA or online duplication of database. We installed only required packages on each type of MySQL Cluster components, for example management node, SQL node and storage nodes. MySQL Cluster supports both disk-based and in-memory database. In this research we use in-memory database approach because this type provides greatly high responsiveness than disk-based approach. The access time of in-memory database is greatly faster than disk-based approach. MySQL Cluster 7.0 also supports up to eight threads in parallel that is very suitable for present processors multi-thread or multi-core era.

IV. System Evaluation

We started Benchw [11] benchmark testing using Query0 which the results were represented in table 1 and Figure 5. The result of two storage nodes was 1.688 seconds. The result of four storage nodes was 1.088 seconds that improved the processing time to 35.54502%. The result of eight storage nodes was 0.772 seconds that improved the processing time to 54.26540%.

Table 1: Benchw Testing Result of Query0

Number of Storage Nodes	Type of Query	Time (Seconds)	Performance Ratio
2	0	1.688	
4	0	1.088	+ 35.54502%
8	0	0.772	+ 54.26540%

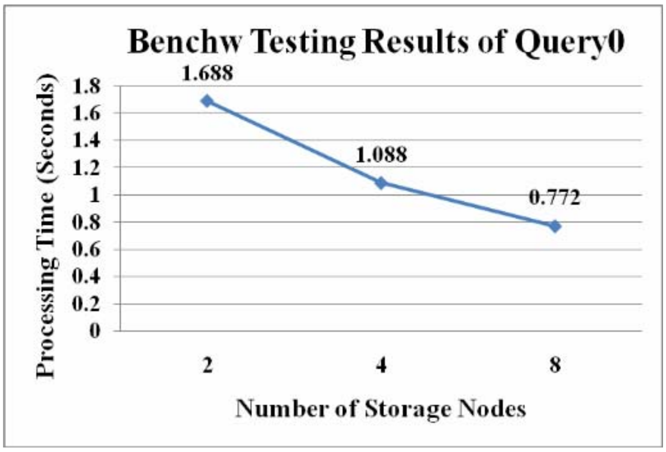

Figure 5: Result of Query0 of Benchw Testing.

The results of Benchw benchmark testing with Query1 were represented in table 2 and Figure 6. The result of two storage nodes was 1.689 seconds. The result of four storage nodes was 1.076 seconds that improved the performance 36.29366%. The result of eight storage nodes was 0.772 seconds that improved the performance 54.29248%.

Table 2: Benchw Testing Result of Query1

Number of Storage Nodes	Type of Query	Time (Seconds)	Performance Ratio
2	1	1.689	
4	1	1.076	+ 36.29366%
8	1	0.772	+ 54.29248%

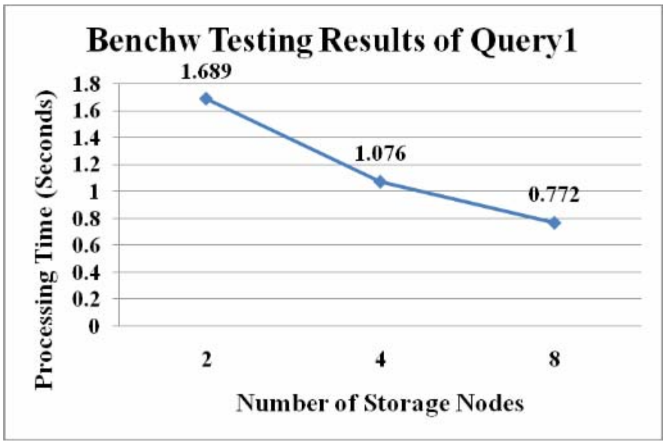

Figure 6: Result of Query1 of Benchw Testing.

The results of Benchw benchmark testing with Query2 were represented in table 3 and Figure 7. The result of two storage nodes was 293.035 seconds. The result of four storage nodes was 324.416 seconds that degraded the performance 10.70896%. The result of eight storage nodes was 342.631 seconds that degraded the performance 16.92494%.

Table 3: Benchw Testing Result of Query2

Number of Storage Nodes	Type of Query	Time (Seconds)	Performance Ratio
2	2	293.035	
4	2	324.416	-10.70896%
8	2	342.631	-16.92494%

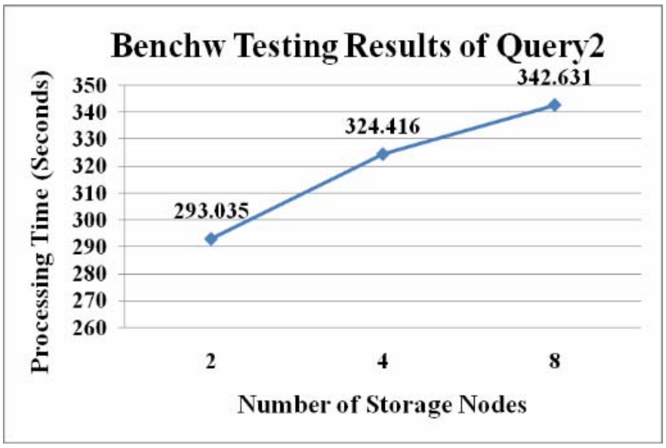

Figure 7: Result of Query2 of Benchw Testing.

The results of Benchw benchmark testing with Query3 were represented in table 4 and Figure 8. The result of two storage nodes was 130.939 seconds. The result of four storage nodes was 149.886 seconds that degraded the performance 14.47097%. The result of eight storage nodes was 170.655 seconds that degraded the performance 30.33268%.

Table 4: Benchw Testing Result of Query3

Number of Storage Nodes	Type of Query	Time (Seconds)	Performance Ratio
2	3	130.938	
4	3	149.886	-14.47097%
8	3	170.655	-30.33268%

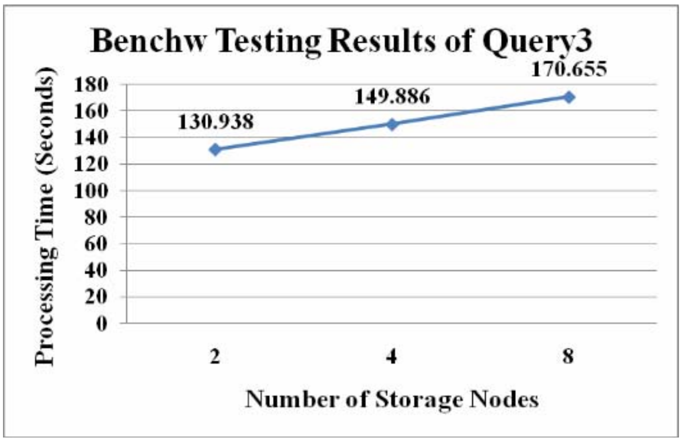

Figure 8: Result of Query3 of Benchw Testing.

Finally, the results of Benchw benchmark testing with Query4 were represented in table 5 and Figure 9. The result of two storage nodes was 17.199 seconds. The result of four storage nodes was 10.957 seconds that improved the performance 36.29281%. The result of eight storage nodes was 7.807 seconds that improved the performance 54.60783%.

Table 5: Benchw Testing Result of Query4

Number of Storage Nodes	Type of Query	Time (Seconds)	Performance Ratio
2	4	17.199	
4	4	10.957	+ 36.29281%
8	4	7.807	+ 54.60783%

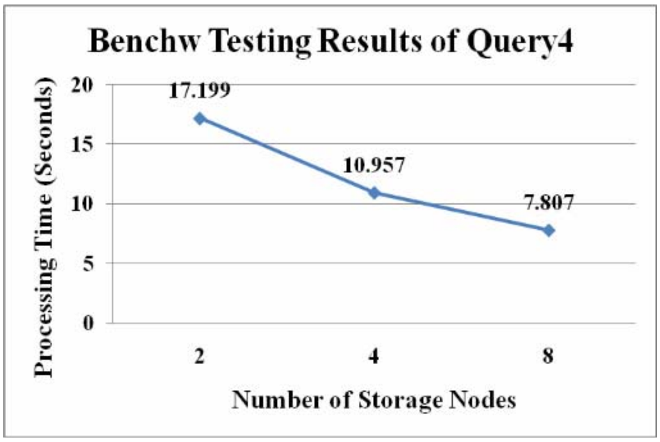

Figure 9: Result of Query4 of Benchw Testing.

The results represented that when we increased more storage nodes, the processing time for test operations trended to use less processing time. The maximum ration that improved performance was 54.60783% by using eight storage nodes on Query4. On the other hand, the worst case from the test that degraded performance was 30.33268% by using eight storage nodes on Query3.

V. Conclusion

The objective of this research was to evaluate the distributed database approach that can improve the performance of database system.

We evaluated the system using Benchw benchmark tool. We had tested four of query types by increasing the number of storage nodes from two to eight nodes. The results represented that when we increased the number of storage nodes which data was stored, the time of processing was improve gratefully. The evaluation may be limited by the maximum number of storage nodes to eight machines. But if we have opportunity to configure more storage nodes and also to improve some other factors that would affect the system performance. Furthermore, from the same system designed we have plan to change from CAT5e wired connection cable to CAT6, CAT6e or CAT7 (if available) wired connection cable to improve the network capacity that may affect the system improvement. Because other hardware specifications such as processor, memory, HDD, network interface and network switching hubs are quite sufficiently at this time except the network cable with CAT5e. We also plan to use more benchmark tools to evaluate more aspects of the system performance. Finally the personal enterprise or others that use MySQL Cluster as DBMSs can use this information to plan their system to meet the requirements.

Acknowledgment

We would like to appreciate our sponsor, Bangkok University who has supported this research and also the previous supported research papers. Furthermore we would like to appreciate 2b-cert Company who supported two gigabit switching hubs, wired cables and some knowledge from expert faculties.

References

[1] Bill Scheela, Thawatchai Jittrapanun, “Venture Capital in Thailand (Report for OSMESP & Thai VC Association)”, Office of SMEs Promotion (OSMESP), May 2007.

[2] Thomas M. Connolly and Carolyn E. Begg, “Database systems : a practical approach to design, implementation, and management”, 4th edition. ,Harlow : Addison-Wesley, c2005.

[3] Peter Rob, Carlos Coronel, “Database systems : design, implementation, and management”, 8th edition, Australia : Thomson/Course Technology, 2009.

[4] Mary Edie Meredith, Duc Vianney , “Linux 2.6 Performance in the Corporate Data Center”, Open Source Development Labs (OSDL)

[5] Deployment Guide 5.1 Red Hat Enterprise Linux, Red Hat Inc.,

[6] Sorapak Pukdesree and Vitalwonhyo Lacharoj, “Evaluating of Distributed Database on PC Cluster Computers”, WEAS Transactions on Information Science & Application, Issue10, Volumn 3, Octomber 2006, Pages 1963-1967.

[7] Sun Microsystems Inc., “MySQL 5.1 Reference Manual Including MySQL Cluster NDB 6.X/7.X Reference Guide MySQL Cluster Database”.

[8] Sun Microsystems Inc., “MySQL Cluster Evaluation Guide Getting the most out of a MySQL Cluster evaluation”.

[9] Sun Microsystems Inc., “MySQL Performance Benchmark Measuring MySQL’s Scalability and Throughput”.

[10] Mikael Ronström, “Performance Guide for MySQL Cluster”.

[11] Mark Kirkwood, “Benchw”, http://sourceforge.net.