An Empirical Study of Distributed Database on PC Cluster Computers

SORAPAK PUKDESREE, VITALWONHYO LACHAROJ, PARINYA SIRISANG
School of Science and Technology
Bangkok University
Rama IV rd., Klongtoey
THAILAND

sorapak@gmail.com, vitalwonhyo@gmail.com, jnosis.s@hotmail.com
http://lily.bu.ac.th/~sorapuck.p/

Abstract—Presently, Information is very importance aspect to be recognized on every application. Modern organizations have stored and managed their information using database management system. The proprietary DBMSs Software is very expensive to spend and also depends on scale of their transactions.

This research focuses on the small medium of enterprise businesses in Thailand that their incomes are less than one and a half million dollar per year. Most of their budget has been spent on productions rather than invested on information technology section. Therefore this research would like to represent that the open source DMBS such as MySQL Cluster can provide high performance with lower cost than enterprise DBMSs. It deploys on distributed database technology that can be scaled the performance dynamically on the PC clustering computers. Therefore SMEs businesses in Thailand can utilize this research to make their plans for the database management system to meet the requirements of businesses.

Keywords—High Performance Computing, Cluster Computers, Database, MySQL Cluster, Distributed Database.

I. INTRODUCTION

Typically, SMEs businesses [1] in Thailand have limited to invest on information management system. The SMEs businesses may dynamically growth or slow down depending on the present environment of economic. Enterprise information systems are very expensive budget to be invested with high risk of return in term of SMEs businesses in Thailand.

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.

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 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.

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.
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

Previously we have researched distributed database using MySQL Cluster 5.0 [6], the previous version, on the old PC machines. Those machines had specifications with a single core on single processor Pentium 4 1.8 GHz processor, 256 MB SDRAM single channel, 20 GB-IDE HDD and one 100 Mbps fast Ethernet NIC. The performance of the new evaluation was greatly higher than the old evaluation.

To evaluate the performance of distributed database [8, 9, 10], we have used SysBench [11] as our benchmark. There are some other benchmark tools including proprietary or open source. Even though there are many of open source benchmark tools but some of them were not work in our environment. There are also many of commercial benchmark tools either but we have chosen SysBench. SysBench is a database benchmark tool developed by MySQL that supports both common and distributed database. SysBench is a modular, cross-platform and multi-threaded benchmark tool for evaluating OS parameters that are important for a system running a database under intensive load. Primarily written for MySQL server benchmarking, SysBench will be further extended to support multiple database backends, distributed benchmarks and third-party plug-in modules. The operations within SysBench include alter-table, large table, connect, create, insert, select and transaction. In this research, we have customized some part of scripts of SysBench to support MySQL Cluster when created the table on the database. We have changed default database engine from MyISAM to NDBCLUSTER.

The results of SysBench testing with read/write operations are shown in table 1. The result of two machines with four threads was 74625 transactions with improve the performance 74.04%. The result of four machines with eight threads was 128654 transactions with improve the performance 200.05%. The result of eight machines with sixteen threads was 127353 transactions with improve the performance 197.02%. The result is shown that when we increased more storage nodes and more MySQL threads, the number of succeed transactions trend to grow up as shown in Figure 5. The best performance ratio of the test was 200.05% by using four storage nodes with eight threads. In case of eight storage nodes with sixteen threads, the performance was slightly downgrade than four storage nodes with eight threads. We have analyzed that even though we have totally sixteen cores of processors but the MySQL Cluster itself supports only eight threads. Therefore only eight cores were active; the other cores were not active for MySQL Cluster. But this case might improve the network capacity in term of parallel accessing to multiple of storage nodes.

Table 1: SysBench Testing Result (Read/Write
The results of SysBench Testing with read only operations are shown in Table 2. The result of two machines with four threads was 99713 transactions with improve the performance 87.70%. The result of four machines with eight threads was 177816 transactions with improve the performance 234.71%. The result of eight machines with sixteen threads was 176234 transactions with improve the performance 231.73%. The result is shown that when we increased more storage nodes and more MySQL threads, the number of succeed transactions trend to grow up as shown in Figure 6. The best performance ratio of the test was 234.71% by using four storage nodes with eight threads. In case of eight storage nodes with sixteen threads, the performance was slightly downgrade to four storage nodes with eight threads. We have analyzed that even though we have sixteen cores of processors but the MySQL Cluster itself support only eight threads. Therefore only eight cores were active; the other cores were not active for MySQL Cluster. But this case might improve the network capacity in term of parallel accessing to multiple of storage nodes.

V. CONCLUSION

The objective of this research is to evaluate the distributed database approach that can improve the performance of database system. SMEs businesses in Thailand do not want to spend their budget in those expensive enterprise database management systems. But using an open source DBMSs, they may take a risk in some situations or in the future and there are not provides support for the users. SMEs businesses absolutely do not want to take any risks from information technology part.

We evaluated the system using SysBench benchmark tool. We have tested two types of operations as read/write and read-only. The result shown that when we increased the number of storage nodes which data was stored, the number of succeed transactions was improve gratefully and also improved the average number of succeed transactions per second. 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 to CAT6, CAT6e or CAT7 (if available) wired connection to improve the network capacity that may affect the system improvement. We also plan to use more benchmark tools to evaluate more aspects of the system performance. SMEs businesses or others may use this information to plan their system to meet the requirements.

VI. ACKNOWLEDGMENT

This research was supported by Bangkok University who has supported facilities, budget and PC computers in the computer laboratory for this research and also the previous supported research papers. Furthermore we would like to appreciate 2b-cert Company [12] who supported two eight-port 1000 Mb switching hubs, CAT5e and CAT6 wired cables and knowledge from expert faculties.

VII. REFERENCES


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


