

Performance Evaluation 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 license to spend depending on the scale of capability to handle their transactions. Therefore this research would like to represent the distributed database methodology that can be scalable to improve performance the database system to meet business requirements.

To implement the distributed database methodology, researcher will use an open source DMBS named MySQL Cluster as research's tool. MySQL Cluster deploys on distributed database technology that can be scaled the performance dynamically on the PC Clustering computers. MySQL Cluster can provide higher performance with significantly lower cost than enterprise DBMSs based on PC Clustering computers. This research focuses on the small and medium of enterprise businesses in Thailand which their incomes are less than one and a half million dollar per year. Most of their budget have been spent on productions rather than invested on information technology section.

Therefore SMEs businesses in Thailand can utilize this research's information to make their plans for the database management system to meet the requirements of their businesses.

Key-Words: - High Performance Computing, PC Clustering Computers, Database, MySQL Cluster, Distributed Database, Distributed Processing.

1 Introduction

Typically, SMEs businesses [1] in Thailand are very importance segments which can produce several billion dollars per year. SMEs businesses will effect increasing of the GDP of the country moderately. Most of SMEs businesses in Thailand are quit small or locally businesses that some of them may not ever deploy information technology to improve their businesses such as e-commerce or e-business. SMEs businesses in Thailand have limited to invest on information management system. The SMEs businesses may dynamically growth or slow down depending on the present environments of economic. Enterprise information systems are very expensive budget to be invested with high risk of return in term of SMEs businesses in Thailand. Currently some SMEs businesses that utilize e-commerce are growing up continuously because the SMEs businesses may not familiar with e-commerce at the first time but later that they will

find that businesses which deploy information technology can run their businesses in all time at any where as 24/7 businesses. They can promote their products on the internet technology therefore their customers can find the products that may match with the customer's needed. If all things are working efficiently the related businesses can grow up in terms of number of volumes or number of income.

Even though information technology may useful for the SMEs businesses, but the businesses will have to invest on IT resources such as faculties, hardware, software licenses, time, and budget to utilize the information technology. In case of the SMEs businesses which deploy information technology, most of them store and manage their information on centralization approach using DBMSs as illustrated in Figure 1. Generally, 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 simply to manage and maintain for database management administrators. On the other hand if there are higher volumes of requests, the only one database server may not be 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 licenses. Not only cost of the new hardware and software licenses but also the cost of migration from the previous system to the new higher performance database server which is also expensive and take time to implement. This approach would easy and less of investment at the beginning time of implementation but in the future this would be difficult to scalable and expensive of investment.

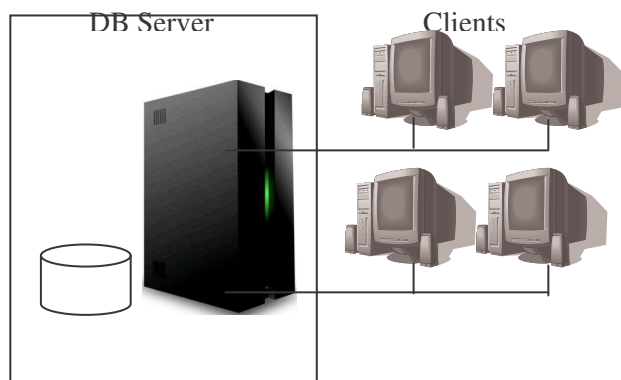


Figure 1: Typical Database Management.

Another approach to improve the performance of database server is distributed processing [2, 3]. In this approach there may have more than one processing database server that each server handles those requests independently in parallel methodology. The performance of this approach depends on many factors such as hardware capabilities, software configurations and also loads balancing technology that would be deployed with this approach. Even though this approach can use share-nothing storage that means each processing database server stores their own data on their own storage but the performance would be downgrade due to the synchronization of data of each processing database server that will consume time and resources significantly. To improve the

processes of the synchronization between each processing database server the data must be stored on shared storage which is accessed and manipulated by each database servers depend on users' requests. But 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, high cost of maintenance and so forth. This approach is not only widely used in commercial DBMSs such as Oracle database or Microsoft database but also can be used in some open source DBMSs such as MySQL or MySQL Cluster.

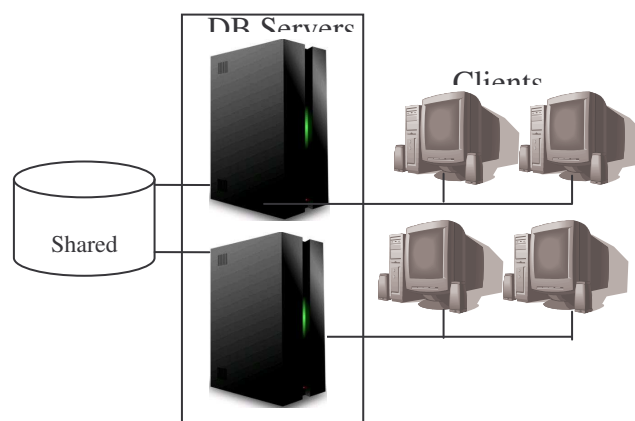


Figure 2: Distributed Database Processing using shared storage.

The other approach 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 via fast network connections. Those cluster computers are connected together as a high performance computer. The distributed DBMSs will be handling those pieces of data as one unit. Users do not need to know where the exactly data will be stored on which cluster nodes. There are several advantages of this approach such as higher processing performance, more memory capacity, higher network bandwidth, higher I/O bandwidth and also scalability of the system. 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 instead of purchasing a high performance storage server. On the other hand, there are some drawbacks of this approach such as the complexity of configurations of the system and complexity of distributed algorithm of this approach. This research will implement the distributed data using MySQL Cluster 7.0 as DBMSs which the latest version of MySQL Cluster at this time.

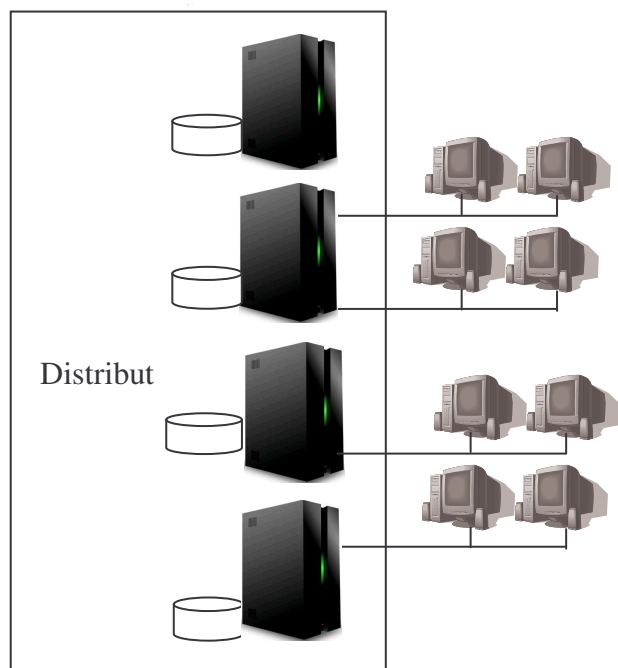


Figure 3: Distributed Data and distributed processing.

In some system they can utilize both distributed database processing and also distributed data approaches to deploy the advantages of each approach. Some of the benefits of the system that utilize both approaches are higher performance in term of capability of processing or scalability of storage capacity. But there are some of the drawbacks of the mixed system are the complexity of the configurations and the cost of the implementation of the system.

2 Related work

Hasham Pathan had published his research with titled “MySQL Cluster Database 7: Performance Benchmark”. In the research he was not only benchmark the performance of MySQL Cluster

Database 7.0 but also MySQL Cluster Database 6.3. The research illustrated that MySQL Cluster Database 7.0 had significantly higher performance than MySQL Cluster Database 6.3. Highlight of his results are:

The result of four data storage nodes was 251,000 transactions per minute which was more than four times improvement over the MySQL Cluster 6.3 release.

The result of two data storage nodes was 143,000 transactions per minute which was more than four times improvement over the MySQL Cluster 6.3 release.

The architecture of his research in case of two data storage nodes which were running on two Sun Fire X4450 system with eight processor cores per data nodes. The MySQL server nodes were running on a combination of Sun Fire x4450 systems and Sun Fire x4600 systems.

In case of four data storage nodes, four Sun Fire X4450 system were used to deploy the data nodes which each data node used eight cores. The MySQL server nodes were running on a combination of two Sun Fire x4600, one Sun Fire x4240 and four Sun Fire x4450 systems. In both cases all machines were connected using gigabit Ethernet.

Hasham Pathan used DBT2 as the testing benchmark in his research. Database Test 2 (DBT2) is an open source benchmark developed by Open Source Development Labs (OSDL). DBT2 can simulate a typical Online Transaction Processing (OLTP) application that performs transactions with around ten to twenty SQL statements per transactions within five distinct transaction types. The DBT2 benchmark can be used as a good indicator in regards to the expected performance of any applications that perform fairly simple transactions and execute these transactions in a repetitive fashion. To simulate OLTP applications which are typical real-time applications, both DBT2 and MySQL Cluster were configured as an in-memory database. DBT2 can be downloaded from www.iclastron.com.

The results of his research were very impressive. The architecture of the systems was very high performance machines which were about nine Sun Fire machines. The cost of those machines and other peripherals would be more than one million dollars. This might be suitable for enterprise systems. On the other hand, our research was focusing on the low cost systems such as PC computers to implement the higher performance distributed database system.

Furthermore our research can be used for small and medium of enterprise businesses to make plans of database workload to meet their business's requirements.

3 Hardware Implementation

In this research, we had used ten PC computers that were in the computer laboratory as illustrated in Figure 4. The specifications of each computer were a single chip 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 illustrated in Figure 4. Presently, each computer was cost approximately less than six hundred dollars. Therefore the total cost of the entire system would approximately six thousands and five hundreds dollar. All of computers are connected using two eight-port 1000 Mbps switching hubs with CAT5e wired cable connections.

The system was a closed system that prevented other factors that might affect the results of the experimental. Even though these computers did not have high performance as new PC computers or high performance servers but there were several advantages such as significantly lower cost of hardware and software licenses also the simplicity to configuration of PC computers. Furthermore the PC clustering could be scaled up the capability depended on the required capacity. Fortunately we did not have to spend on PC computers due to the machines are in computer laboratory. We only spent on two eight-port 1000 Mbps switching hubs because the current switching hubs in this computer laboratory were only 100 Mbps.

In the distributed data approach, the network bandwidth is one of the very importance factors that will affect the results of the testing experimental. Even though presently the 1000 Mbps is standard and widely add-on in most PC computers but the wired cable is only CAT5e that has maximum 350 Mbps of network capacity. That means we may not utilize the full capacity of 1000 Mbps connections because of the limitation of capability of CAT5 wired cable. 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, Infiniband or 10 Gbps-Ethernet network systems.

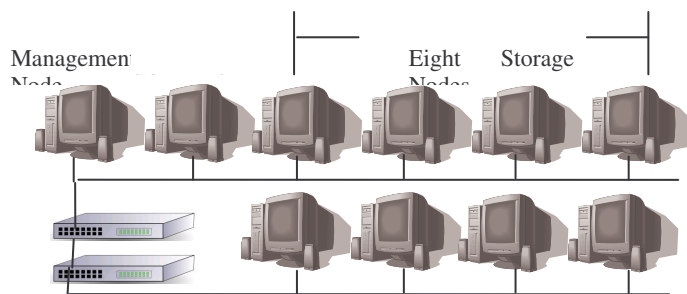


Figure 4: System design in this research.

4 Software Implementation

We decided to use Red Hat Enterprise Linux 5 [5] as our operating system for the system. RHEL5 is one of operating system that has reliable, potential, stable, secure and so forth. We did 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 may 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 the system can be updated patches or packages via internet that can be very useful for system administrator to fix or upgrade the system software.

Red Hat is one of the world's leading open source technology solutions provider which under GNU General Public License. Therefore we did not have to spend for the software license as proprietary operating system. We may optionally have to pay for subscription for update or fix the system when needed.

Previously we had researched and discovered the MySQL Cluster 5.0 for the past four years. Therefore we have some experiences and familiars with MySQL Cluster.

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 nodes 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 significantly faster than disk-based approach. MySQL Cluster 7.0 also supports up to eight threads of data nodes in parallel that is very suitable for present processors multi-thread or multi-core era.

There are some researches papers represented that the new version of MySQL Cluster 7.0 has greatly high performance than previous versions.

5 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 network. The performance of this new evaluation was significantly higher than the previous evaluation.

To evaluate the performance of distributed database [8, 9, 10], we had used SysBench [11] as our benchmark in this research. There are some other benchmark tools including proprietary or open source tools. Even though there are many of open source benchmark tools but some of them were not work in our environment that may cause of our mistaken of the system configurations. There are also many of commercial benchmark tools either but we had 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 the database under intensive load. Primarily written for MySQL server benchmarking, SysBench will be further extended to support multiple database backend, 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 to utilize the distributed data methodology approach.

We started the test focus on the term of the number of processed transactions in the specific of

time. The results of SysBench testing with read/write operations were illustrated in table 1. The result of two data storage machines with four threads of SysBench was 74,625 transactions with improvement the performance of 74.04%. The result of four data storage machines with eight threads of SysBench was 128,654 transactions with improvement the performance 200.05%. The result of eight data storage machines with sixteen threads was 127,353 transactions with improvement the performance 197.02%. The result was illustrated that when we increased more storage nodes and more MySQL threads, the number of succeed transactions trended to grow up as illustrated in Figure 5. The best performance ratio of this test was 200.05% by using four data storage nodes with eight threads of SysBench. In case of eight data storage nodes with sixteen threads of SysBench, the performance was slightly downgrade than four storage nodes with eight threads of SysBench. We had analyzed that even though we had totally sixteen cores of processors but the MySQL Cluster itself supports only eight threads on data nodes. 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.

Figure 6 also illustrated the average number of transactions per second that can be executed successfully within a specific of time. By using two storage nodes, the performance can improve the average approximately two times. By using four storage nodes, the performance can improve the average approximately four times. Finally by using eight storage nodes, the performance can improve the average approximately four times. The results of average number of transactions per second were corresponding with the result of the number of transactions with a specific of time.

Table 1: SysBench Testing Result (Read/Write Operations)

# Storage nodes	# Threads	# Transactions (300 seconds)	Performance Ratio	Average Transactions per second
1	2	42877		142.92
2	4	74625	+74.04 %	248.74
4	8	128654	+200.05 %	428.82

8	16	127353	+197.02 %	424.49
---	----	--------	-----------	--------

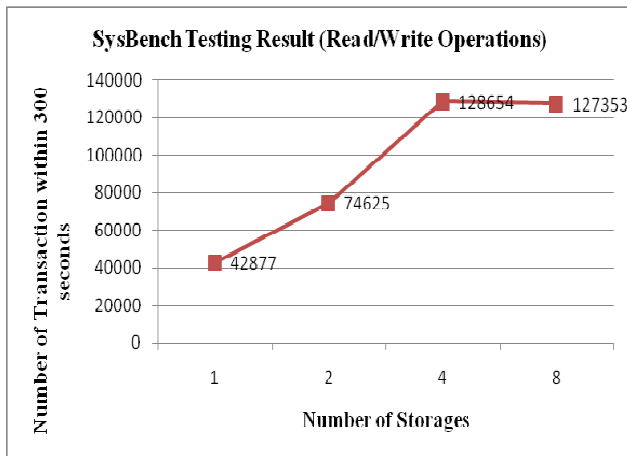


Figure 5: Testing Result (Read/Write Operations)

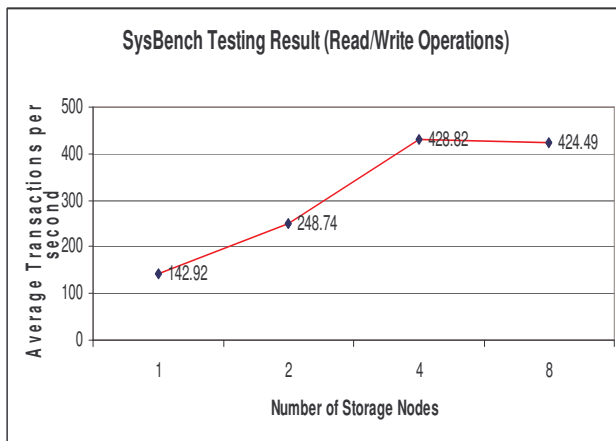


Figure 6: Testing Result (Average Transactions per second- Read/Write Operations)

The results of SysBench Testing with read only operations were illustrated in table 2. The result of two data storage machines with four threads of SysBench as 99,713 transactions with improvement the performance 87.69%. The result of four data storage machines with eight threads of SysBench was 177,816 transactions with improvement the performance 234.71%. The result of eight data storage machines with sixteen threads of SysBench was 176,234 transactions with improvement the performance 231.73%. The results were illustrated that when we increased more data storage nodes and more MySQL threads, the number of succeed transactions trended to grow up as illustrated in Figure 7. The best performance ratio of the test was 234.71% by using four data storage nodes with eight

threads of SysBench. In case of eight data storage nodes with sixteen threads of SysBench, the performance was slightly downgrade than four storage nodes with eight threads. We had analyzed that even though we had totally sixteen cores of processors but the MySQL Cluster itself supports only eight threads on data nodes. Therefore only eight cores were active; the other cores were not active for MySQL Cluster. But this case might improved the network capacity in term of parallel accessing to multiple of storage nodes.

Figure 8 also illustrated the average number of transactions per second that can be executed successfully within a specific of time. By using two storage nodes, the performance can improve the average approximately two times. By using four storage nodes, the performance can improve the average approximately four times. Finally by using eight storage nodes, the performance can improve the average approximately four times. The results of average number of transactions per second were corresponding with the result of the number of transactions with a specific of time.

Table 2: SysBench Testing Result (Read Only Operations) focused on number of Transactions within 300 seconds.

# Storage nodes	# Threads	# Transactions (300 seconds)	Performance Ratio	Average Transactions per second
1	2	53,125		177.08
2	4	99,713	+87.70%	332.36
4	8	177,816	+234.71%	592.70
8	16	176,234	+231.73%	587.42

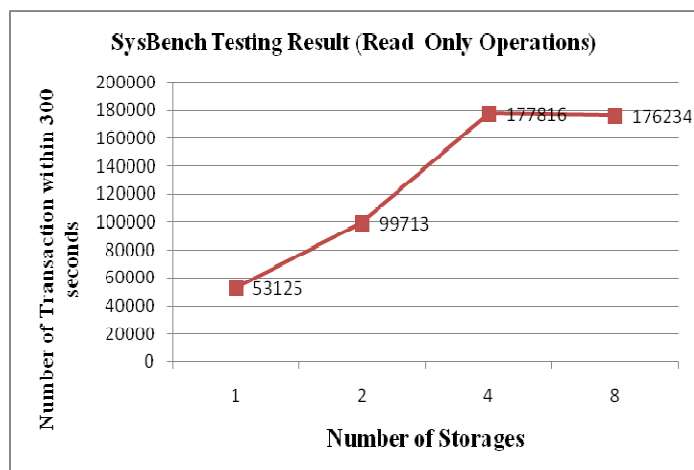


Figure 7: SysBench Testing Result (Read Only Operations)

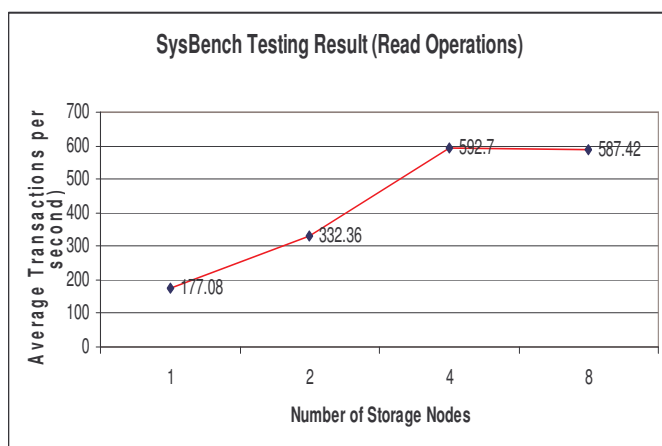


Figure 8: Testing Result ((Average Transactions per second- Read Only Operations)

Next we focused the test in the term of the number of processed requests in the specific of time. The results of SysBench testing with read/write operations were illustrated in table 3. The result of two data storage machines with four threads of SysBench was 814,663 requests with improvement the performance 74.04%. The result of four data storage machines with eight threads of SysBench was 2,444,426 requests with improvement the performance 200.05%. The result of eight data storage machines with sixteen threads of was 2,419,707 requests with improvement the performance 197.02%. The result illustrated that when we increased more storage nodes and more MySQL threads, the number of succeed transactions trended to grow up as illustrated in Figure 7. The best performance ratio of the test was 200.05% by using four data storage nodes with eight threads of SysBench. In case of eight data storage nodes with

sixteen threads of SysBench, the performance was slightly downgrade than four storage nodes with eight threads of SysBench. We had analyzed that even though we had totally sixteen cores of processors but the MySQL Cluster itself supports only eight threads on data nodes. 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 3: SysBench Testing Result (Read/Write Operations) focused on number of Requests within 300 seconds.

# Storage nodes	# Threads	# Read/write requests	Average Requests per second
1	2	814,663	2,715.45
2	4	1,417,875	4,726.02
4	8	2,444,426	8,147.65
8	16	2,419,707	8,065.27

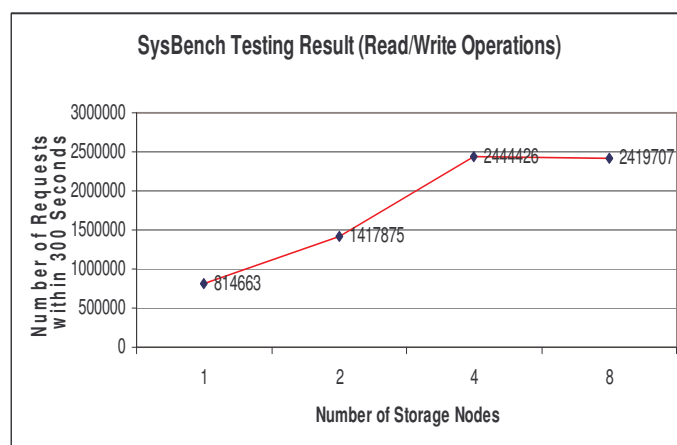


Figure 7: SysBench Testing Result (Read/Write Operations) focused on number of Requests within 300 seconds.

Finally we focused the test in the term of the number of processed requests in the specific of time. The results of SysBench testing with read/write operations are illustrated in table 4. The result of two data storage machines with four threads of SysBench was 1,3959,823 requests with improvement the performance 87.70%. The result of

four data storage machines with eight threads of SysBench was 2,489,424 requests with improvement the performance 234.71%. The result of eight data storage machines with sixteen threads of was 2,467,276 requests with improvement the performance 231.73%. The result illustrated that when we increased more storage nodes and more MySQL threads, the number of succeed transactions trend to grow up as illustrated in Figure 7. The best performance ratio of the test was 234.71% by using four data storage nodes with eight threads of SysBench. In case of eight data storage nodes with sixteen threads of SysBench, the performance was slightly downgrade than four storage nodes with eight threads of SysBench. 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 4: SysBench Testing Result (Read Only Operations) focused on number of Requests within 300 seconds.

# Storage nodes	# Threads	# Read/write requests	Average Requests per second
1	2	743,750	2479.13
2	4	1,395,982	4653.09
4	8	2,489,424	8297.78
8	16	2,467,276	8223.86

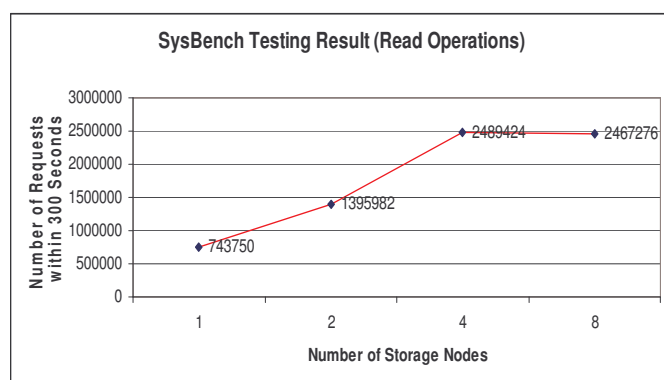


Figure 8: SysBench Testing Result (Read Only Operations) focused on number of Requests within 300 seconds.

6 Conclusion

The objective of this research was to evaluate the distributed database approach that can improve the performance of database system. SMEs businesses in Thailand do not desire 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 desire to take any risks from information technology part.

We evaluated the distributed database system using SysBench benchmark tool. We have tested two types of operations as read/write and read-only. The result illustrated that when we increased the number of data 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 data storage nodes to eight data storage machines. But if we have opportunity to configure more data 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 in Thailand) wired connection to improve the network capacity that may affect the system improvement. We also plan to use more benchmark tools such as DBT2 to evaluate more aspects of the system performance. SMEs businesses or other organizations may use this information to plan their database system to meet the requirements.

There were some bottlenecks of this research such as there was only one SQL node which will response and execute all requests from all clients. Even though MySQL Cluster can support up to eight threads on data nodes but in the situations that are high work load that may not suitable and this would be the bottleneck of this system. Furthermore the research implemented up to eight data storage nodes with only one SQL node. We also had assumed that CAT5e wired cable connections was one of the system bottlenecks. Therefore we had changed the wired cable connections from CAT5e to CAT6 but the results of the test were a little improvement.

We have planed to study in the next research paper that if we increase the number of SQL nodes

one by one the performance will be increase or may be drop. And how much of the performance will be increase or may be drop?

In the next research paper will benchmark the scalability of number of distributed processing on distributed data system which integrates both two methodologies.

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.

References

- [1] Alex Davies, MySQL Cluster: Two webserver setup. <http://dev.mysql.com/tech-resources/articles/mysql-cluster-for-two-servers.html>
- [2] Alexey Kopytov, "SysBench manual", www.sysbench.sf.net
- [3] Allan Packer, MySQL 5.4 Scaling on Nehalem with Sysbench. http://blogs.sun.com/allanp/entry/mysql_5_4_scaling_on.
- [4] Ameet S. Talwadker, Survey of performance issues in parallel database systems, *Journal of Computing Sciences in Colleges*, v.18 n.6, p.5-9, June 2003.
- [5] Amit P. Sheth, James A. Larson, Federated database systems for managing distributed, heterogeneous, and autonomous databases, *ACM Computing Surveys (CSUR)*, v.22 n.3, p.183-236, Sept. 1990 [doi>10.1145/96602.96604]
- [6] Amjad Umar, Distributed computing : a practical synthesis of networks, client-server systems, distributed applications, and open systems, Prentice-Hall 1993.
- [7] Arjen Lentz, An Introduction to MySQL Cluster Architecture and Use, MySQL AB, November 2006.
- [8] Artem Russakovsky, Setting up A MySQL Cluster, <http://beerpla.net/2008/03/26/setting-up-a-mysql-cluster/>, March 26th, 2008.
- [9] Attiya, Hagit, Distributed computing : fundamentals, simulations and advanced topics. , International ed. ,McGraw Hill 2000.
- [10] Bill Scheela, Thawatchai Jittrapanun, "Venture Capital in Thailand (Report for OSMESP & Thai VC Association)", Office of SMEs Promotion (OSMESP), May 2007.
- [11] CAT5e, <http://en.wikipedia.org/wiki/Cat5e>
- [12] CAT6, <http://en.wikipedia.org/wiki/Cat6>
- [13] David Bell and Jane Grimson, Distributed Database Systems, 1st edition, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1992, ISBN:0201544008.
- [14] Falko Timme, Setting Up Master-Master Replication With MySQL 5 On Debian Etch, <http://www.howtoforge.com/forums/showthread.php?t=23501>.
- [15] G.Pfister: In Search of Clusters. Prentice Hall PTR, NJ, 2nd Edition, NJ 1998.
- [16] GC. Fox. What have we learn from using real parallel machines, to solve real problem? In *Proc. Of The 3rd Conference on Hypercube concurrent computers and applications* page Vol2 pp 897-955, Pasadena, CA January 1999.
- [17] George Coulouris, Distributed Systems: concepts and design, 4th ed., Addison-Wesley, 2005.
- [18] Gomer Thomas and Teams, Heterogeneous distributed database systems for production use, *ACM Computing Surveys (CSUR) Volume 22*, Issue 3 (September 1990) Special issue on heterogeneous databases.
- [19] Hank Dietz, Paralle Processing using Linux. <http://yara.ecn.purdue.edu/~pplinux>. April 28, 1999.
- [20] Hasham Pathan, "MySQL Cluster Database 7: Performance Benchmark", April 2009.
- [21] Jay Pipes, Performance Tuning Best Practices, www.mysql.com.
- [22] Jeremy Zawodny, Introduction to MySQL Cluster, <http://jeremy.zawodny.com/blog/archives/001868.html>, April 14, 2004.
- [23] Johan Andersson, Mat Keep, Performance / Tuning Best Practices -MySQL Cluster, SUN Micro Systems.
- [24] Kris Buytaert, Building a High-Availability MySQL Cluster, Febuary 16, 2006.
- [25] M. T. Özsu and P. Valduriez, Principles of Distributed Database Systems (2nd edition), Prentice-Hall, 1999. ISBN 0-13-659707-6.
- [26] M. Tamer Ozsu, P. Valduriez, Principles of distributed database systems, Prentice-Hall, Inc., Upper Saddle River, NJ, 1991.
- [27] M. Tamer Özsu, DISTRIBUTED DATABASE SYSTEMS. Volume 24, Issue 8 August 1991, Distributed computing systems: separate resources acting as one, Pages: 68 – 78, IEEE Computer Society Press Los Alamitos, CA, USA, ISSN:0018-9162.
- [28] M. Tamer Özsu, Patrick Valduriez, Distributed and parallel database systems, *ACM Computing Surveys (CSUR) Volume 28*, Issue 1 (March 1996) Pages: 125 - 128, 1996, ISSN:0360-0300.
- [29] M. Tauber and T.Stricker. Accurate Performance evaluation, Modelling and Prediction of a Message Passing Simulation Code based on Middleware. In *Proc. Of the SC98 Conference*, Orlando, FL, November 1998.
- [30] Mary Edie Meredith, Duc Vianney, "Linux 2.6 Performance in the Corporate Data Center", Open Source Development Labs (OSDL)
- [31] Mikael Ronström, Performance Guide for MySQL Cluster, Sun Microsystems.
- [32] Mikiya Okuno, MySQL Cheat Sheet v1., <http://samurai-mysql.blogspot.com/2009/12/mysql-cheat-sheet-v10.html>, 2009-12-28.
- [33] MySQL 5.1 Reference Manual Including MySQL Cluster NDB 6.X/7.X Reference Guide MySQL Cluster Database, Sun Microsystems Inc.,
- [34] MySQL 5.1 Reference Manual, SUN Micro Systems. MySQL Cluster 7.1,

- <http://www.mysql.com/products/database/cluster/>.
- [35] MySQL Cluster Evaluation Guide Getting the most out of a MySQL Cluster evaluation, Sun Microsystems Inc.
 - [36] MySQL Cluster Evaluation Guide Getting the most out of a MySQL Cluster evaluation. www.mysql.com, April 2000.
 - [37] MySQL Cluster for Web and E-Commerce Applications: Growing Revenues and Enhancing Customer Loyalty, An Oracle White Paper, April 2010.
 - [38] MySQL Cluster on Debian AMD64, <http://www.howtoconfig.com/database/mysql-database/mysql-cluster-on-debian-amd64.html>.
 - [39] MySQL Performance Benchmark Measuring MySQL's Scalability and Throughput, Sun Microsystems Inc.
 - [40] MySQL Performance Benchmarks Measuring MySQL's Scalability and Throughput, A MySQL® Technical White Paper, March 2005.
 - [41] Open Source Development Labs Database Test 2 User's Guide Version 0.21, Open Source Development Labs, Inc.
 - [42] Optimizing Performance of the MySQL Cluster Database, SUN Micro Systems. Pages: 237 - 266 Year of Publication: 1990 ISSN:0360-0300.
 - [43] Patrick Valduriez, Parallel database systems: open problems and new issues, Distributed and Parallel Databases, v.1 n.2, p.137-165, April 1993 [doi>10.1007/BF01264049].
 - [44] Peter Rob, Carlos Coronel, "Database systems : design, implementation, and management", 8th edition, Australia : Thomson/Course Technology, 2009.
 - [45] R.Buyya. "High performance Cluster Computing" Vol I. Prentice Hall 1999.
 - [46] Sorapak Pukdesree and Vitalwonhyo Lacharaj, "Evaluating of Distributed Database on PC Cluster Computers", WEAS Transactions on Information Science & Application, Issue10, Volumn 3, Octomber 2006, Pages 1963-1967.
 - [47] Stefano Cer, Distributed databases : principles and systems 1986 ,McGraw-Hill.
 - [48] T. Grabs, K.Bohm, and H.-J.Schek. High-level Parallisation in a Database Cluster: a Feasibiolity Study Using Document Services. In Proc. Of the International Condference on Data engineering (ICDE), Heidelberg, Germany, April 2001. IEEE Computer Society.
 - [49] Tae-Kyung Kim , Seung-Hyun Jung , Kyoung-Ran Kim , Jae-Soo Yoo , Wan-Sup Cho, HyperDB: a PC-based database cluster system for efficient OLAP query processing, Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems, November 19-21, 2007, Cambridge, Massachusetts.
 - [50] Thomas M. Connolly and Carolyn E. Begg, "Database systems : a practical approach to design, implementation, and management", 4th edition. ,Harlow : Addison-Wesley, c2005.
