8	16	127353	+197.02 %	424.49

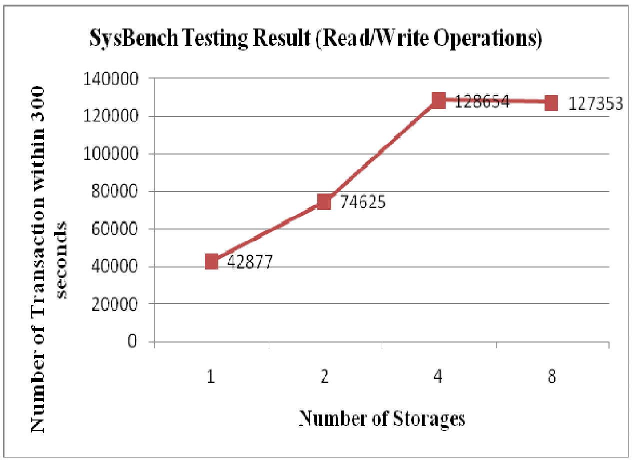

Figure 5: Testing Result (Read/Write Operations)

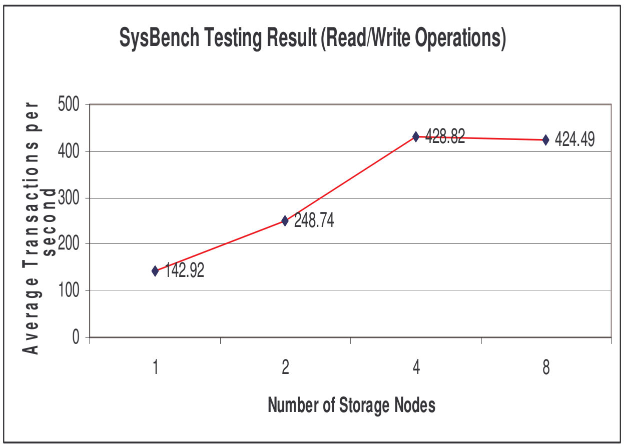

Figure 6: Testing Result (Average Transactions per second- Read/Write Operations)

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

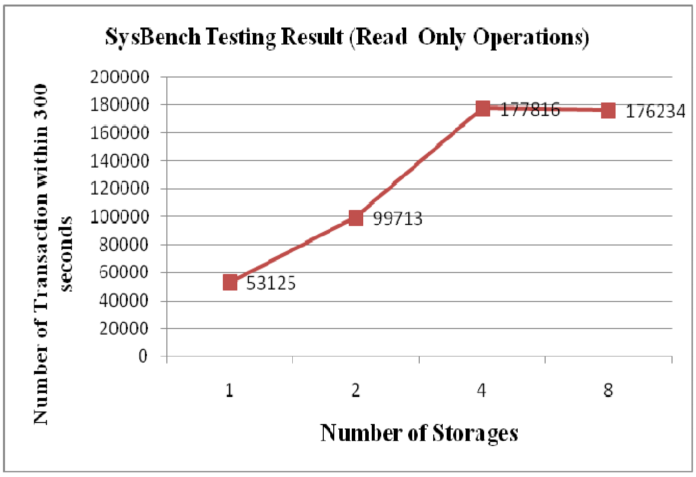

Figure 7: SysBench Testing Result (Read Only Operations)

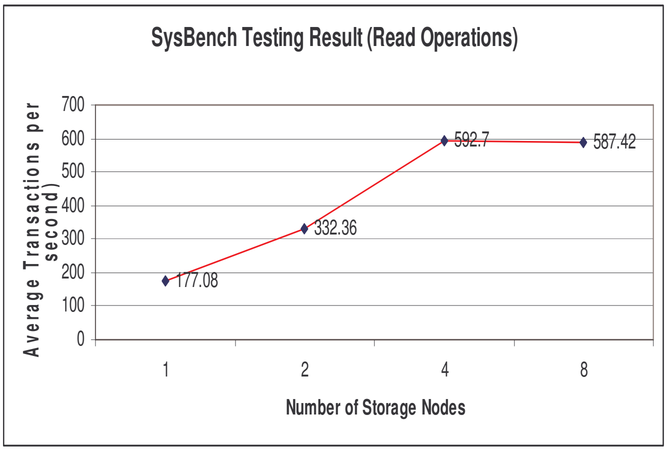

Figure 8: Testing Result ((Average Transactions per second- Read Only Operations)

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

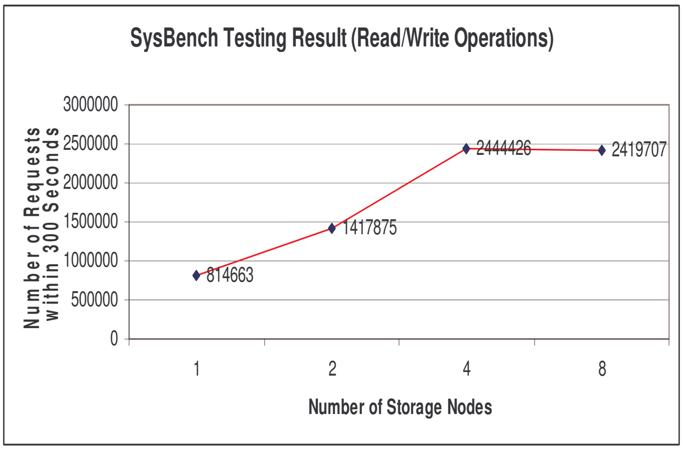

Figure 7: SysBench Testing Result (Read/Write Operations) focused on number of Requests within 300 seconds.

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

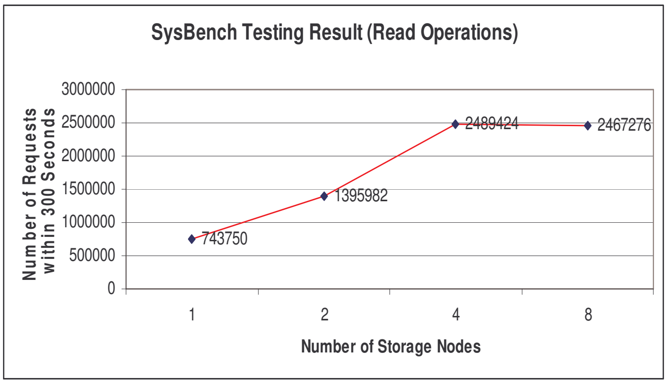

Figure 8: SysBench Testing Result (Read Only Operations) focused on number of Requests within 300 seconds.

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 Russakovskii, 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, 2rd 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. Taufer and T.Stricker. Accurate Performance evaluation, Modelling and Prediction of a Message Passing Simulation Code based on Middleware. In Proc. Of the SC98 Conference, Oralando, 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 1. Prentice Hall 1999.

[46] 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.

[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.

****