

An Examination on Database Servers with the Cluster Architecture and Some Consideration on Enhancing PGP Public Key Servers with Clustered PCs¹

Jiahong Wang
Iwate Prefectural University

In section one, we describe the requirement for enhancing a PGP public key server using the Personal Computer (PC) cluster. Then in section two, we examine the free database management systems available to us, focusing on whether they can be used for a PC cluster system and their performance comparison. Since it is decided that the PostgreSQL is used (because it is the only one that can satisfies our requirements stated in section one), and the PostgreSQL cannot used for our purpose directly, we examine the technique of implementing a PC cluster database system using PostgreSQL in section three.

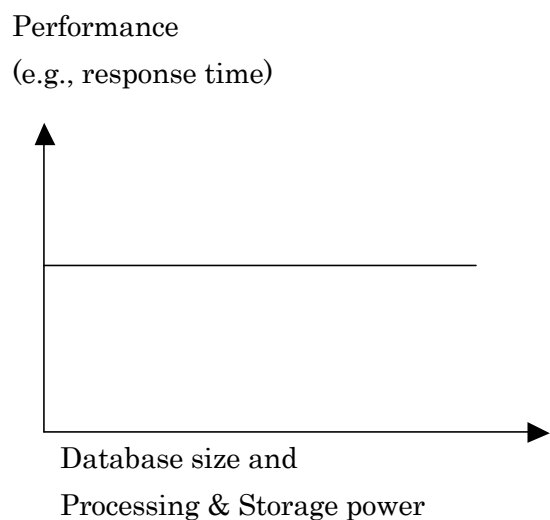
1. The Requirements for Enhancing PGP Public Key Servers with Clustered PCs

1.1 High Scalability

The system should be capable to expand its ability by adding additional processing and storage power. Especially, linear scaleup and linear speedup are expected.

- Linear Scaleup:

Scaleup measures whether a constant performance (e.g., response time) can be maintained as adding a proportional number of processors and disks increases the workload. If a database system can maintain a sustained performance for a linear increase in both database size and processing and storage power, we say it has the property of

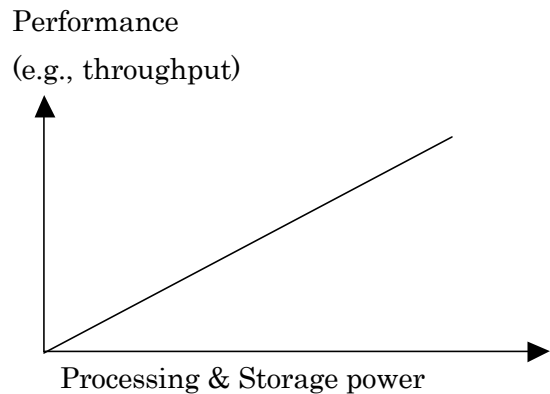


¹ COMM Internal Note 011 / PGP-005

linear scaleup.

- Linear Speedup:

Speedup measures whether adding additional processors and disks to a system results in a corresponding increase in performance (e.g., the throughput). If a system has a linear increase in performance for a constant database size and linear increase in processing and storage power, we say it has the property of linear speedup.



1.2 High Availability

The system should be highly available. Here the high availability means it automatically allows one node to take over the responsibilities of another node that has failed.

1.3 Transaction Processing

If the function of only the traditional PGP key server is required, transaction processing may be unnecessary. However, Electronic Commerce (EC) will benefit from the transaction processing ability, and in the near future, it is expected that the EC is brought into our public key server, transaction processing should be supported.

1.4 Low Cost

The system should be costless. A cluster of PCs should be used to satisfy the requirements stated above. All the PCs should be locally available via a high speed of network. In addition, the database management system should be free

2. Free Database Management Systems

The top two database management systems available to us are MySQL and PostgreSQL. There is also a lesser-used database system called mSQL.

(1) PostgreSQL (ORDBMS): Limitations of PostgreSQL:

- Maximum size for a database: unlimited (60GB databases exist)
- Maximum size for a table: 64 TB on all operating systems
- Maximum size for a row: unlimited in 7.1 and later
- Maximum size for a field: 1GB in 7.1 and later
- Maximum number of rows in a table: unlimited
- Maximum number of columns in a table: 1600
- Maximum number of indexes on a table: unlimited

(2) MySQL (A storage tool)

(3) MiniSQL (A storage tool)

2.1 How MySQL Compares to PostgreSQL

In some cases MySQL is faster than PostgreSQL. This may be because (1) MySQL provides a limited set of features; (2) MySQL is coded to be multi-threaded while PostgreSQL uses processes; Context switching and access to common storage areas is much faster between threads than between separate processes.

PostgreSQL, on the other hand, provides more features than MySQL. Especially, PostgreSQL provides more powerful and robust transaction support than MySQL. If you develop an application for highly complex business use, PostgreSQL would be a better candidate.

Many performance comparison results on MySQL and PostgreSQL can be found. Two of them are given below. (Unfortunately, they are contradictory.)

(1) Performance Comparison by the GreateBridge, a leading provider of solutions based on PostgreSQL. ("<http://www.angelfire.com/country/aldev0/pgsql/GreatBridge.html>")

➤ See Figure 1

(2) Performance Comparison by MySQL group ("<http://www.mysql.com/>):

➤ See Figure 2.

2.2 How about the Support to Clustering

We found that PostgreSQL cannot be directly used for our purpose unless it is modified properly. On the other hand, MySQL is ready for clustering use. For example, it is said that SteelEye (<http://www.steeleye.com>) has delivered an Enterprise-Grade High Availability Platform for Linux-Based MySQL Database Servers. Key features include flexible multi-node configuration options, packaged support for a variety of applications, databases, server and networking platforms in shared-data as well as shared-nothing environments.

2.3 The Conclusion

The PostgreSQL seems more proper candidate for our purpose. This is because (1) it is decided that the transaction processing should be fully supported; (2) The cluster system should be implemented without so much cost. As a result, we have to modify PostgreSQL so that it can support the PC cluster system for our purpose.

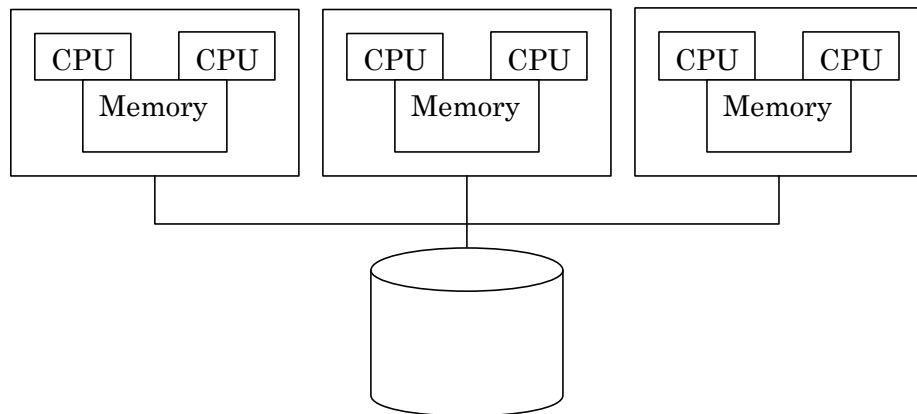
3. The Architecture Available to Constructing PC Clusters

Two architectures are available. The one is the shared-disk architecture, and the other is the shared-nothing architecture. The following gives a brief description of them.

3.1 Shared-Disk Architecture: it has the following characteristics

- Each node consists of one or more CPUs and associated memory; Each CPU accesses its local memory only. That is, memory is not shared between nodes.

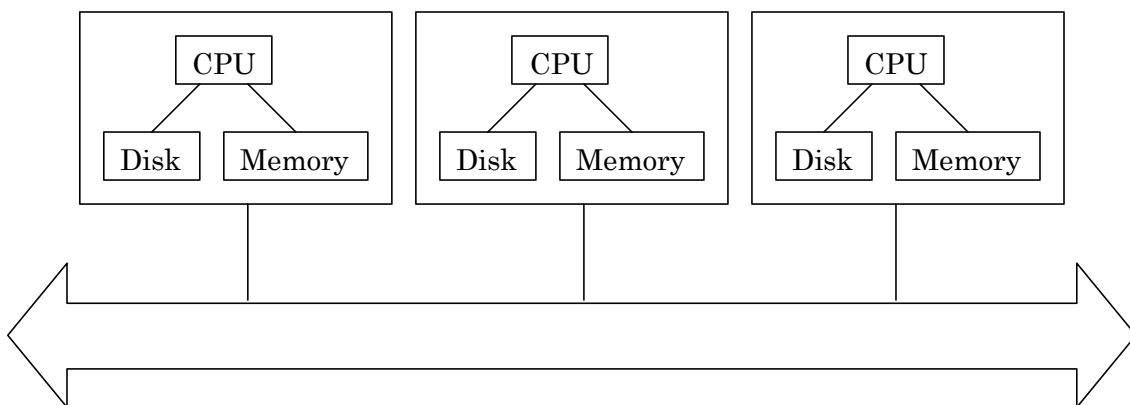
- All nodes share the disks



- Advantages:
 - It provides high availability. All data is accessible even if one node dies.
 - It provides a similar programming environment to a non-clustered system. As a result, getting a shared-disk database to market can be faster than getting the shared-nothing system.
- Disadvantages:
 - Inter-node synchronization is difficult.
 - There is operating system overhead of running shared disk software

3.2 Shared-Nothing Architecture: it has the following characteristics:

- Each CPU accesses its local memory and disk, and if a database is located on that disk, access depends entirely on the CPU that owns it.



- Advantages
 - It provides for high scalability, and system growth is practically unlimited.
 - Failure is local: if one node fails, the others stay up.
- Disadvantages
 - Partitioning a database is difficult. If it is not performed properly, the data

skew may damage system performance.

- More overheads are required for a process to work on a disk belonging to another node.

3.3 Shared Nothing /Shared Disk Combined Systems

In fact, a combined system of shared-nothing system and shared-disk system can be very advantageous. It brings together the advantages of shared nothing system and shared disk system, and overcomes their respective limitations.

3.4 The Conclusion

It is decided that the shared-nothing system is first implemented, and then the shared nothing /shared disk combined systems. The reason is as follows. Compared with a shared-disk system, a shared-nothing system is more easily implemented on a clustered PC system. In addition, high scalability is very important to our application, and a shared-nothing system has better scalability.

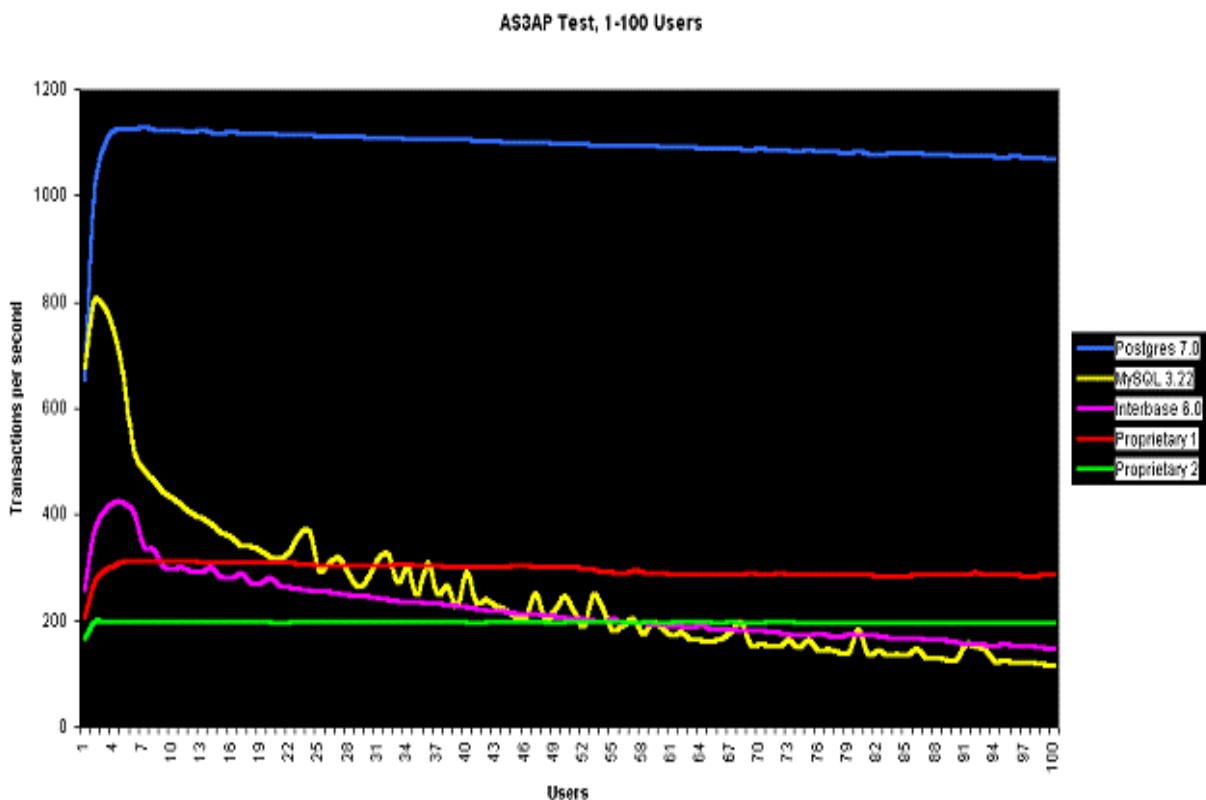


Figure 1: Performance Comparison of PostgreSQL and MySQL (Provided by the GreateBridge, <http://www.angelfire.com/country/aldev0/pgsql/GreatBridge.html>)

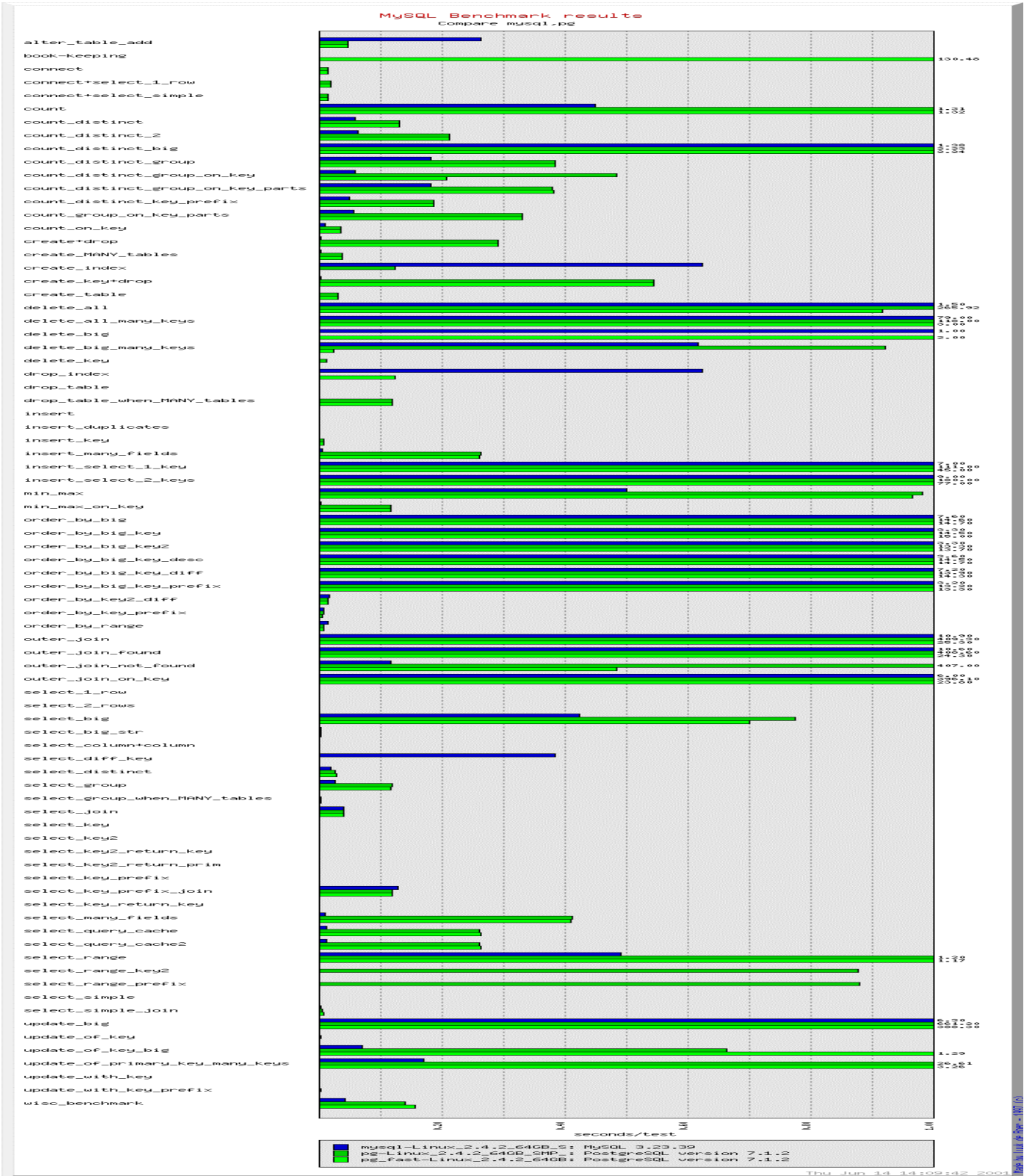


Figure2: Performance Comparison of PostgreSQL and MySQL (Provided by MySQL group, <http://www.mysql.com/>)