https://doi.org/10.26552/com.C.2008.4.53-57

Svetlana Andjelic – Slobodan Obradovic – Branislav Gacesa *

A PERFORMANCE ANALYSIS OF THE DBMS – MySQL vs PostgreSQL

Abstract: Modern communications among business subjects imply an exchange of a lot of data and information. To enhance that process in all its aspects, electronic data storing and processing is mandatory and enabled by an application of a DBMS (Date Base Managment System). This paper shows the performances analysis of the two most popular open source DBMSs - MySQL and PostgreSQL. First, some characteristics of these DBMSs are shortly described. Then the applied procedure of the performed testing is described. That is, the query (select, insert, delete and order by) execution times were measured for the both DBMSs mentioned, and the results shown in tabular and graphical forms. The goal of this paper is to make the choice of an adequate DBMS easier for future users.

Key words: open source DBMS, MySQL, PostgreSQL, ACID

1. Introduction

Nowadays, it is virtually impossible to imagine the functioning of a larger system without some sort of electronic data storage. DBMS is certainly the most often used data management and permanent storage system.

It is interesting to note that the United Nations (UN) have recommended to their members to use the open source software, especially in the areas of health protection, education and international commerce. According to UN, open source software is the most adequate means for the development of their members.

In the paper, some of the basic characteristics of the most often used open source DBMS – MySQL and PostgreSQL – are described first. Second, more important part of this paper is dedicated to the description of the testing performed on those DBMS and the exposition of the results obtained.

2. The basic characteristics of the DBMS – MySQL & PostgreSQL

2.1. MySQL

MySQL is an open source database management system, conceived and implemented to rival the MS SQL. It reaches that aim to a degree, especially so when less voluminous and simpler tasks are considered.

MySQL is renowned for its speed and reliability. That's why it's one of the most popular database management systems on the web. Of the data base access technologies, there exist drivers for ODBC, JDBC i OleDB, as well as libraries for C++, Delphi, Perl, Python, PHP and TCL.

The biggest downside to this system is not checking of the referential integrity. External keys are supported by syntax, which can be misleading, because they are not actually applied. The current version neither supports views, nested queries nor stored procedures.

The run-time environment for PHP MySQL contains two significant flaws, which enable abusers to take control of the server through the memory_limit function, thus circumventing the security mechanisms in the strip_tags function. In the meantime, PHP group has anounced its first final environment version of generation 5.0. The most important new features are the building-in of the Zend Engine II library with a new object model, rewamped support for XML based on libxml2 library, built-in support for SOAP and the new MySQLi add-on for working with version 4.1 of the MySQL server. The latest versions are MySQL 5.1 through 5.1.21-beta and MySQL 6.0 Falcon. Falcon has been specially developed for systems that are able to support larger memory architectures and multi-threaded or multi-core CPU environments.

2.2. PostgreSQL

PostgreSQL is an object-relational database management system (ORDBMS), based on POSTGRES version 4.21 developed in the University of California Computer Sciences Department at Berkley.

The first demo version became operative in 1987. Berkley's POSTGRES was officially completed with version 4.2. In 1994, Andrew Yu and Jolly Chen implemented the POSTGRES interpreter for SQL. Under its new name, Postgres95, it was published on the web as an open-source version of the original POSTGRES Berkley code in 1996. It supports a large part of the SQL standard and provides for many modern characteristics: complex queries,

* **Svetlana Andjelic[1], Slobodan Obradovic[2], Branislav Gacesa[3]**
[1]The Railway College of Vocational Studies, Belgrade, Serbia, E-mail: angeo@verat.net or svetangela@gmail.com
[2]Megatrend University, Belgrade, Serbia
[3]Telekom Srpske A. D., SRJ Internet Srpske, Banja Luka, Bosna and Herzegovina

external keys, triggers, views and transactional integrity. Newer versions are 8.1 & 8.2, and 8.2.4 is the latest.

PostgreSQL has native programming interfaces for C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, among others, and exceptional documentation. PostgreSQL supports international character sets, multibyte character encodings, Unicode, and is locale-aware for sorting, case-sensitivity, and formatting. PostgreSQL runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. PostgreSQL is fully ACID compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages).

Although not the fastest, the PostgreSQL database has been characterised as the most advanced on many tests. Inspired by Oracle, from the very beggining it supported transactions, triggers, referential integrity and matched procedures (in contrast to the MySQL). The recommendations for its use refer more to the proven quality and robustness than to the performances themselves. Among others, PostgreSQL users include UNICEF, Cisco and American Chemical Society.

2.3. MySQL vs PostgreSQL

The next table (Table 1) contains comparative data among two most popular "Open Source" database today (MySQL and PostgreSQL).

MySQL vs PostgreSQL [1] Table 1

	PostgreSQL 8.0	MySQL 4.1	MySQL 5.0
Operating System	Windows, Linux, all BSDs, HP-UX, AIX, OS X, Unixware, Netware...	Linux, Windows, FreeBSD, MacOS X, Solaris, HP UX, AIX, and other	Linux, Windows, FreeBSD, MacOS X, Solaris, HP UX, AIX, and other
ANSI SQL compliance	ANSI-SQL 92/99	Possible	Possible
Sub-selects	Yes	Yes	Yes
Transactions	Yes	Yes InnoDB tables	Yes
Database replication	Yes	Yes	Yes
Foreign key support	Yes	Yes InnoDB tables	Yes
Views	Yes	No	Yes
Stored procedures	Yes (pl/SQL)	No	Yes
Triggers	Yes	No	Yes
Unions	Yes	Yes	Yes
Full joins	Yes	No	No
Constraints	Yes	No	No
Cursors	Yes	No	Partial
Procedural languages	Yes	No	Yes
Vacuum	Yes	Yes	Yes
Different table types	No	Yes	Yes
ODBC	Yes	Yes	Yes
JDBC	Yes	Yes	Yes
Other APIs	Most of languages	Most of languages	Most of languages
IPv6 support	Yes	No	No

3. Test procedure

The benchmark itself consists of following steps and procedures:

- generates a set of alerts;
- connects to databases;
- creates tables inserts data;
- perform SELECT operations;
- deletes all data from the table;

When creating, adding to, editing and deleting data from the base (SELECT, INSERT, ORDER and DELETE), the system sends various warnings that trigger certain processes. In Figure 1 these processes are shown for both MySQL versions, as well as for the PostgreSQL database. The diagram was made according to the four different types of tables in bases, including the fsync PSQL option.

4. Hardware and software characteristics

Tests were performed using Intel SR2200, Xeon 2.4 GHz with 1 Gb RAM. The software was:

- operating system - Gentoo Linux 2006.1 with Linux 2.6.14 kernel,
- datebases:
  - PostgreSQL 8.0.1
  - MySQL 4.1.9
  - snort 1.8 (DB v100-103)
  - ACID 0.9.6b10

Alert sets were generated using open source "Nmap" and "NessusD" vulerability scanners and tools.

In computer science, ACID (*Atomicity, Consistency, Isolation, Durability*) is a set of properties that guarantee that database transactions are processed reliably. In the context of databases, a single logical operation on the data is called a transaction. [1]

- *Atomicity* – refers to the ability of the DBMS to guarantee that either all of the tasks of a transaction are performed or none of them are. For example, the transfer of funds can be completed or it can fail for a multitude of reasons, but atomicity guarantees that one account won't be debited if the other is not credited.
- *Consistency* – property ensures that the database remains in a consistent state before the start of the transaction and after the transaction is over (whether successful or not).

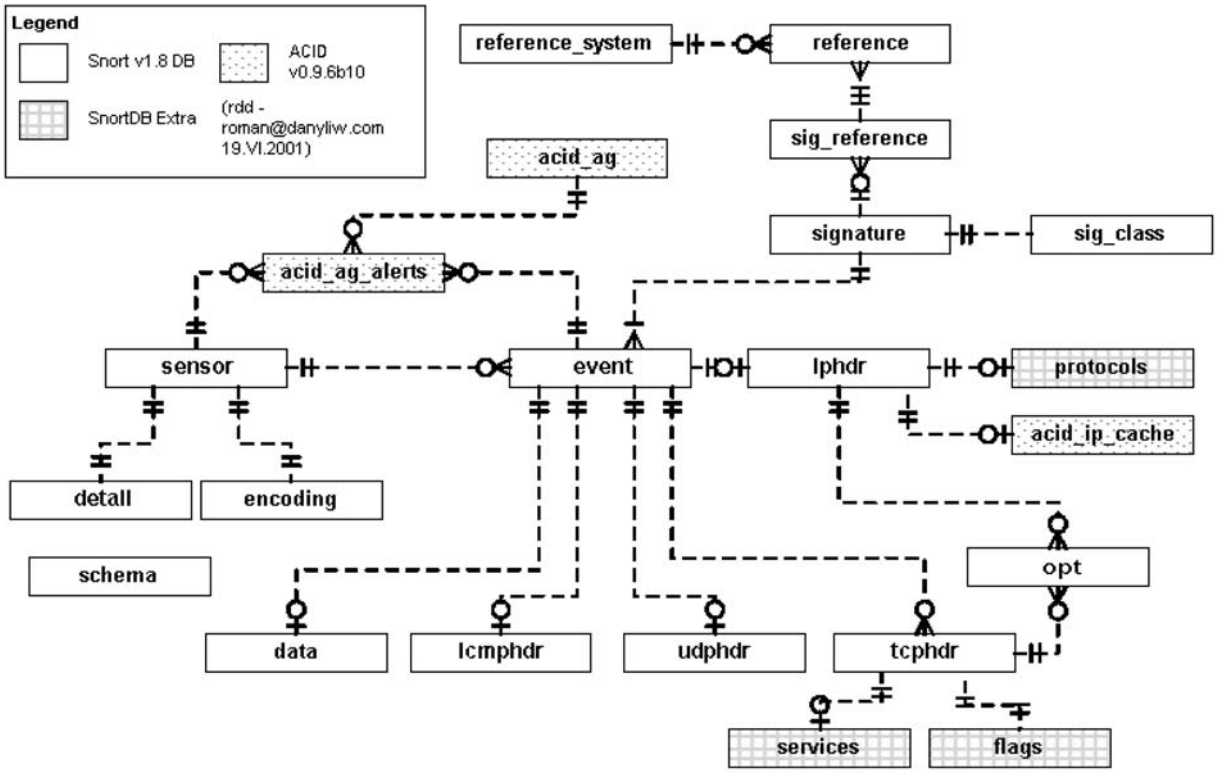


Fig. 1. Logical Database diagram – snort 1.8 (DB v100-103) and ACID 0.9.6b10 [1]

- *Isolation* – refers to the ability of the application to make operations in a transaction appear isolated from all other operations. This means that no operation outside the transaction can ever see the data in an intermediate state; for example, a bank manager can see the transferred funds on one account or the other, but never on both–even if he ran his query while the transfer was still being processed.
- *Durability* – refers to the guarantee that once the user has been notified of success, the transaction will persist, and not be undone. This means it will survive system failure, and that the database system has checked the integrity constraints and won't need to abort the transaction.

5. Test results

The results which follow were obtained by direct measuring, and obeying the previously described test procedure.

5.1. INSERT clause

Inserting pregenerated set of 50000 records to both MyISAM and InnoDB MySQL table types and PostgreSQL with fsck enabled and disabled (commit after each insert)

Table 2 shows timed results while executing "INSERT" statement against four different MySQL and PostgreSQLdatabase table.

Tabular display of the time needed for execution of the INSERT clause Table 2

INSERT	MyISAM-v4	InnoDB-v4	MyISAM-v5	InnoDB-v5	PgSQL	PgSQL
					fsync=true	fsync=false
5000	3006	34745	1674	42897	17540	4825
10000	4967	71402	2235	85964	14052	8772
15000	7474	103469	3525	128697	21873	13064
20000	9996	143009	4739	168609	110359	17374
25000	12524	170862	5897	212718	101442	21974
30000	15034	186316	6996	261772	51390	26148
35000	17515	119604	8266	297953	49103	30484
40000	20040	58373	9512	337577	158071	35112

The following graph (Graph 1) contains a graphical display of the results shown in Table 2. X-axis shows the number of records, while y-axis depicts time in milliseconds.

5.2. SELECT clause

Inserting pregenerated set of 50000 records to both MyISAM and InnoDB MySQL table types and PostgreSQL with fsck enabled and disabled (commit after each insert)

Table 3 shows timed results while executing "SELECT" statement against four different MySQL and PostgreSQLdatabase table.

The following graph (Graph 3) shows timed results while executing "ORDER BY" statement against four different MySQL and PostgreSQLdatabase table. X-axis shows the number of records, while y-axis depicts time in milliseconds.

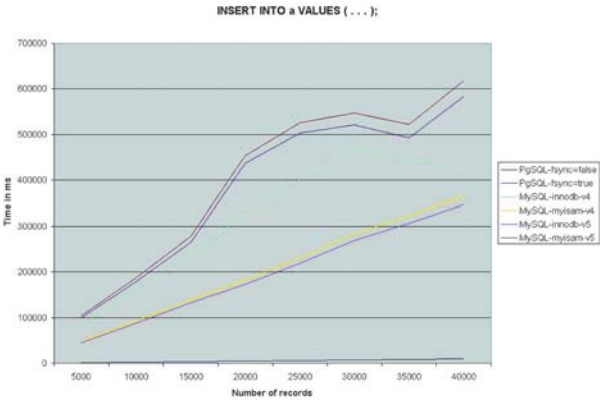
5.4. DELETE clause

Inserting pregenerated set of 50000 records to both MyISAM and InnoDB MySQL table types and PostgreSQL with fsck enabled and disabled (commit after each insert).

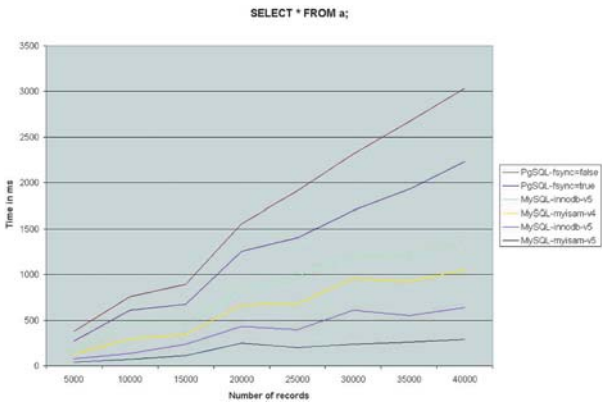
Table 4 shows timed results while executing "DELETE" statement against four different MySQL and PostgreSQLdatabase table.

Tabular display of the time needed for execution of the SELECT clause Table 3

SELECT	MyISAM-v4	InnoDB-v4	MyISAM-v5	InnoDB-v5	PgSQL	PgSQL
					fsync=true	fsync=false
5000	45	42	44	34	108	104
10000	157	174	71	66	141	150
15000	110	113	114	122	218	219
20000	240	156	248	182	429	300
25000	288	303	201	193	420	506
30000	352	251	235	375	489	614
35000	378	285	262	286	721	738
40000	415	316	293	347	864	805



Graph 1. INSERT clause



Graph 2. SELECT clause

The following graph (Graph 2) contains a graphical display of the results shown in Table 3. X-axis shows the number of records, while y-axis depicts time in milliseconds.

5.3. ORDER BY clause

Inserting pregenerated set of 50000 records to both MyISAM and InnoDB MySQL table types and PostgreSQL with fsck enabled and disabled (commit after each insert)

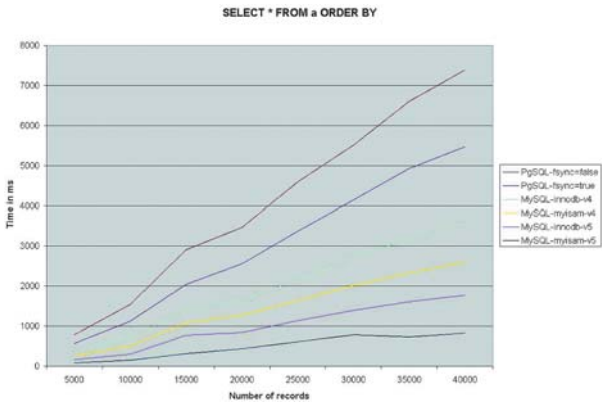
The following graph (Graph 4) contains a graphical display of the results shown in Table 4. X-axis shows the number of records, while y-axis depicts time in milliseconds.

6. Conclusion

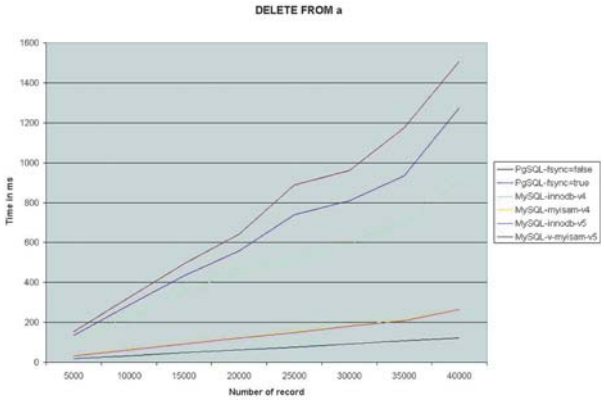
Modern communications among business subjects imply exchange of a lot of data and information. To enhance that process

Tabular display of the time needed for execution of the DELETE clause Table 4

DELETE	MyISAM-v4	InnoDB-v4	MyISAM-v5	InnoDB-v5	PgSQL	PgSQL
					fsync=true	fsync=false
5000	1	78	1	4	23	19
10000	2	166	2	8	56	40
15000	3	259	3	11	79	60
20000	3	330	3	15	104	82
25000	3	370	4	19	216	150
30000	4	405	4	21	220	151
35000	4	490	4	25	230	241
40000	5	664	6	28	341	235



Graph 3. ORDER BY clause



Graph 4. DELETE clause

an electronic data storing and processing is mandatory and enabled by the application of a DBMS.

From the test results shown one concludes that the MySQL/MyISAM DBMS performs better than the PostgreSQL.

In the case where, regardless of the integrity of data, the speed of query execution is important, one should choose MySQL. Contrarywise, PostgreSQL provides greater consistency and smaller robustness of the base as well as greater security from the system failure.

The general conclusion cannot be reached, but depending on the type of the server and types of clients to access the base, some recommendations can be given. If both the server and the clients are Linux (and transactions are needed), Postgres is recommended. In the case of Linux/Unix server and Windows clients, then Oracle certainly is recommended, since ODBC support for MySQL is bad, and nonexistent for PostgreSQL.

Before choosing one of the DBMSs, the criteria relevant to the problem should be laid down.

It is to be presumed that the performances of MySQL and PostgreSQL will converge in the future, aiming to create a fast, secure and optimally robust DBMS.

References

[1] ANDJELIC, S., OBRADOVIC, S., GACESA, B.: *A Performance Analysis of the Open Sorce Database*, XIII conference YUINFO 2007, Kopaonik, Serbia, ISBN 978-86-85525-02-5
[2] KLINE, K. E., KLINE, D., HUNT, B.: *SQL in Nutshell,* O'Reilly, 2004
[3] WORSLEY, J. C., DRAKE, J. D.: *Practical PostgreSQL*, O'Reilly, 2002
[4] RUSSELL, J. DYER, T.: *MySQL In A Nutshell*, O'Reilly, 2005
[5] HORN, J. W., GREY, M.: *MySQL: Essential Skills*, McGraw-Hill Professional, 2004
[6] http://support.summersault.com/why_postgresql.html
[7] http://www.postgresql.org
[8] http://www.mysql.com