About PostgreSQL

OpenERP makes extensive use of the open source database PostgreSQL. It is one of several layers of the technology “stack” in which OpenERP leverages robust, freely available, open source software in order to provide greater value to its customers, as listed in the table.

<table>
<thead>
<tr>
<th>OpenERP Application</th>
<th>Other ERP Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Reporting</td>
<td>OpenReports</td>
</tr>
<tr>
<td></td>
<td>Crystal Reports</td>
</tr>
<tr>
<td></td>
<td>Microsoft Access</td>
</tr>
<tr>
<td></td>
<td>Java, Microsoft.Net</td>
</tr>
<tr>
<td>Dev Tools/App Servers</td>
<td>Qt, C++</td>
</tr>
<tr>
<td></td>
<td>Microsoft Access</td>
</tr>
<tr>
<td>Database</td>
<td>PostgreSQL</td>
</tr>
<tr>
<td></td>
<td>Oracle®, Microsoft® SQL Server</td>
</tr>
<tr>
<td>Operating System</td>
<td>Linux</td>
</tr>
<tr>
<td></td>
<td>Windows®, UNIX</td>
</tr>
</tbody>
</table>
| Total Cost of Component Software: | No License Fees | $$$$$

So what is PostgreSQL?

Is it really comparable to enterprise-class database software from companies such as Oracle and Microsoft? For most applications, including OpenERP, the answer is a resounding “YES”.

PostgreSQL History

Like many other software products – open and closed – PostgreSQL’s roots are in the academic environment. It began as a project called Ingres developed at the University of California at Berkeley (1977-1985). The Ingres code was enhanced by Relational Technologies/Ingres Corporation, from which one of the first commercially successful relational database servers was produced. Computer Associates later purchased Ingres Corporation. Later at Berkeley, Michael Stonebraker led a team to develop an object-relational database server called Postgres (1986-1994). The Postgres code was assimilated by Illustra Corporation and was developed into a commercial product. Illustra was later purchased by Informix and integrated into Informix's Universal Server. Stonebraker served as CTO of Informix for several years.

In 1995, two Berkeley graduate students added SQL capabilities to Postgres. A year later, the core steering group for the now-active PostgreSQL open source project came together and began issuing regular source code releases of the software. Today, the PostgreSQL development community consists of thousands of developers worldwide, guided by a five-member core steering group, similar to the way that Linux creator Linus Torvalds oversees and screens new enhancements for Linux.

PostgreSQL, now at Release 8.0, has matured into a product with significant transactions throughout, complex queries, commercial-grade SQL support, and complex data types.
PostgreSQL Features

Following is a partial list of distinguishing features (PostgreSQL website www.postgresql.org):

- Fully ACID compliant
- ANSI SQL 92 and 99 compliant
- Referential Integrity
- Replication, allowing the duplication of the master database to multiple slave machines
- Native interfaces for ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby
- Rules
- Views
- Triggers
- Full Unicode support
- Sequences
- Inheritance
- Outer Joins
- Sub-selects
- An open API
- Stored Procedures
- Native SSL support
- Seven procedural languages, used by OpenERP for all ERP business logic
- Hot stand-by and failover
- Better than row-level locking
- Functional and Partial indexes
- Native Kerberos authentication
- Support for UNION, UNION ALL and EXCEPT queries
- Loadable extensions offering SHA1, MD5, XML, and other functionality
- Tools for generating portable SQL to share with other SQL-compliant systems
- Extensible data type system providing for custom, user-defined data types and rapid development of new data types
- Cross-database compatibility functions for easing the transition from other, less SQL-compliant RDBMS

Benchmarks

In 2001, independent tests pitted PostgreSQL against three widely published database systems. The databases tested were:

- PostgreSQL version 7.0
- MySQL (another open source database) version 3.22
- Oracle version 8i
- Microsoft SQL Server version 7

Out of a need to be able to compare different databases, running under different operating systems, and on different hardware platforms, over time the industry has embraced several benchmark tests which can level the playing field — allowing databases to be compared “apples to apples”. Two of the more prominent tests used to evaluate database performance are the AS3AP and the TPC-C.
ANSI SQL Standard Scalable and Portable (AS3AP)

This database benchmark was designed with several goals in mind: to be scalable enough that it can be used on both large and small platforms (which are comprised of the hardware, database, and operating system); to indicate an “equivalent database size” which is the maximum database size capable of running this benchmark for the indicated platform in under 12 hours; and to determine a “cost per megabyte” for the “equivalent database”. The transactions performed by this benchmark give a good assessment of the raw transaction-processing power of the platform and its ability to scale.

Following are the results of the AS3AP testing with 100 concurrent users of the system:

MySQL enjoys a reputation for great speed in applications such as “quick read” Web site data retrievals. But as we can see, after about 10 concurrent users, performance fell off sharply. Oracle and Microsoft SQL Server both underperformed, but were fairly consistent. PostgreSQL’s performance was a surprise, as it had long been considered the slow and stodgy (if reliable) open source database in the field; with the multitude of performance enhancements in the version 7.x family, it was now the fastest as well.
Transaction Processing Performance Council (TPC)

Probably the most accepted Database Benchmark in the industry is the TPC-C. TPC-C denotes Version C of the benchmark designed by the TPC. Unlike some of the other benchmarks which merely judge processing power (speed), Version C of the TPC benchmark is meant to emulate the on-line transaction processing (OLTP) typical in a real-world business environment. In particular, it mimics an order-entry system. Like a real-world order-entry system, the transactions processed in this benchmark mimic the management, selling, and distribution of a product or service typical in any industry. This benchmark results in two metrics: the first is the total number of transactions the database accomplished in a minute (a measure of throughput); the second metric is the number of transactions per minute divided by the Total Cost of Ownership (TCO) of the database, operating system, hardware, maintenance, and licenses for a total of five years. This metric is thus a measure of price performance (efficiency).

Following are the results of the TPC-C testing with 100 concurrent users of the system:

As we can see, in a very transaction-intensive test environment, PostgreSQL held its own with the two commercial offerings. It must be noted that MySQL was not able to run this test at the time, as it has limited support for true ACID transactions. Subsequent versions have added transaction support, but at a substantial cost in performance.

Copyright (c) 2001 by Command Prompt, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at http://www.opencontent.org/openpub/).