History of PostgreSQL*

Bruce Momjian
pgman@candle.pha.pa.us

Jan 03, 2002

PostgreSQL is an object-relational database under active development on the Internet. You can learn more by visiting *http://www.postgresql.org*.

1 Introduction

PostgreSQL is the most advanced open source database server. In this article, you will learn about databases, open source software, and the history of POSTGRESQL.

Three basic office productivity applications exist: word processors, spreadsheets, and databases. Word processors produce text documents critical to any business. Spreadsheets are used for financial calculations and analysis. Databases are used primarily for data storage and retrieval. You can use a word processor or spreadsheet to store small amounts of data. However, with large volumes of data or data that must be retrieved and updated frequently, databases are the best choice. Databases allow orderly data storage, rapid data retrieval, and complex data analysis.

*This article is contributed by Bruce Momjian based on the excerption from his book, **PostgreSQL: Introduction and Concepts**, published by Addison-Wesley. Further information about the book can be obtained from *http://www.postgresql.org/docs/awbook.html*. — **FSM**

2 University of California at Berkeley

POSTGRESQL's ancestor was Ingres, developed at the **University of California at Berkeley** (1977-1985). The Ingres code was later enhanced by *Relational Technologies/Ingres Corporation*, which produced one of the first commercially successful relational database servers. (Ingres Corporation was later purchased by Computer Associates.) Also at Berkeley, *Michael Stonebraker* led a team to develop an object-relational database server called **Postgres** (1986-1994). *Illustra* took the Postgres code and developed it into a commercial product. (Illustra was later pur-

chased by Informix and integrated into Informix 's Universal Server.) Two Berkeley graduate students, **Jolly Chen** and **Andrew Yu** , subsequently added SQL capabilities to Postgres. The resulting project was called **Postgres95** (1994-1995). The two later left Berkeley, but Chen continued maintaining Postgres95, which had an active mailing list.

3 Development Leaves Berkeley

In the summer of 1996, it became clear there was great demand for an open source SQL database server, and a team formed to continue development. Marc G. Fournier of Toronto, Canada, offered to host the mailing list and provide a server to host the source tree. One thousand mailing list subscribers were moved to the new list. A server was configured, giving a few people login accounts to apply patches to the source code using cvs. (cvs sychronizes access by developers to shared program files.)

Jolly Chen has stated, "This project needs a few people with lots of time, not many people with a little time." Given the 250,000 lines of C code, we understood what he meant. (C is a popular computer language first developed in the 1970s.) In the early days, four people were heavily involved: *Marc Fournier* in Canada; *Thomas Lockhart* in Pasadena, California; *Vadim Mikheev* in Krasnoyarsk, Russia; and me in Philadelphia, Pennsylvania. We all had full-time jobs, so we participated in the effort in our spare time. It certainly was a challenge.

Our first goal was to scour the old mailing list, evaluating patches that had been posted to fix various problems. The system was quite fragile then, and not easily understood. During the first six months of development, we feared that a single patch might break the system and we would be unable to correct the problem. Many bug reports left us scratching our heads, trying to figure out not only what was wrong, but how the system even performed many functions.

We had inherited a huge installed base. A typical bug report came in the following form: "When I do this, it crashes the database." We had a long list of such reports. It soon became clear that some organization was needed. Most bug reports required significant research to fix, and many reports were duplicates, so our TODO list included every buggy SQL query. This approach helped us identify our bugs, and made users aware of them as well, thereby cutting down on duplicate bug reports.

Although we had many eager developers, the learning curve in understanding how the database worked was significant. Many developers became involved in the edges of the source code, like language interfaces or database tools, where things were easier to understand. Other developers focused on specific problem queries, trying to locate the source of the bug. It was amazing to see that many bugs were fixed with just one line of C code. Because Postgres had evolved in an academic environment, it had not been exposed to the full spectrum of real-world queries. During that period, there was talk of adding features, but the instability of the system made bug fixing our major focus.

We have set up a column for free database management system and application, this article is the first one for it. Bruce Momjian will contribute more articles on PostgreSQL for upcoming issues —**FSM**

4 PostgreSQL Global Development Team

In late 1996, we changed the name of the database server from Postgres95 to **POSTGRESQL**. It is a mouthful, but honors both the Berkeley name and its SQL capabilities. We started distributing the source code using remote cvs, which allowed people to keep up-to-date copies of the development tree without downloading an entire set of files every day.

Releases occurred every three to five months. Each period consisted of two to three months of development, one month of beta testing, a major release, and a few weeks to issue sub-releases to correct serious bugs. We were never tempted to follow a more aggressive schedule with more releases. A database server is not like a word processor or game, where you can easily restart it if a problem arises. Instead databases are multiuser, and lock user data inside the database, so they must be as reliable as possible.

Development of source code of this scale and complexity is not for the novice. We initially had trouble interesting developers in a project with such a steep learning curve. However, over time, our civilized atmosphere and improved reliability and performance helped attract the experienced talent we needed.

Getting our developers the knowledge they needed to assist with POSTGRESQL was clearly a priority. We had a TODO list that outlined what needed to be done, but with 250,000 lines of code, taking on any item was a major project. We realized developer education would pay major benefits in helping people get started. We wrote a detailed flowchart of the database modules. (All the files mentioned in this chapter are available as part of the POSTGRESQL distribution, or at *http://www.postgresql.org/docs.*) We also wrote a developers' FAQ (Frequently Asked Questions), answering the most common questions of POSTGRESQL developers. With this information, developers became more productive at fixing bugs and adding features.

Although the source code we inherited from Berkeley was very modular, most Berkeley coders used POSTGRESQL as a test bed for research projects. As a result, improving existing code was not a priority. Their coding styles were also quite varied.

We wrote a tool to reformat the entire source tree in a consistent manner. We wrote a script to find functions that could be marked as static or unused functions that could be removed completely. (A static function is used by only one program file.) These scripts are run just before each release. A release checklist reminds us of the items to be changed for each release.

As we gained knowledge of the code, we were able to perform more complicated fixes and feature additions. We redesigned poorly structured code. We moved into a mode where each release had major new features, instead of just bug fixes. We improved SQL conformance, added sub-selects, improved locking, and added missing SQL functionality. A company was formed to offer telephone support.

The Usenet discussion group archives started touting us. At one time, we had searched for POSTGRESQL and found that many people were recommending other databases, even though we were addressing user concerns as rapidly as possible. One year later, many people were recommending us to users who needed transaction support, complex queries, commercial-grade SQL support, complex data types, and reliability — clearly our strengths. Other databases were rec-

ommended when speed was the overriding concern. Red Hat's shipment of POSTGRESQL as part of its Linux distribution quickly expanded our user base. (GNU/Linux is a popular UNIX-like, free operating system.)

Today, every release of PostgreSQL is a major improvement over the last. Our global development team has mastery of the source code we inherited from Berkeley . In addition, every module is understood by at least one development team member. We are now easily adding major features, thanks to the increasing size and experience of our worldwide development team.

5 Open Source Software[1]

PostgreSQL is [free] open source software. The term "open source software" often confuses people. With commercial software, a company hires programmers, develops a product, and sells it to users. With Internet communication, however, new possibilities exist. Open source software has no company. Instead, capable programmers with interest and some free time get together via the Internet and exchange ideas. Someone writes a program and puts it in a place everyone can access. Other programmers join and make changes. When the program is sufficiently functional, the developers advertise the program's availability to other Internet users. Users find bugs and missing features and report them back to the developers, who, in turn, enhance the program.

It sounds like an unworkable cycle, but in fact it has several advantages:

- A company structure is not required, so there is no overhead and no economic restrictions.
- Program development is not limited to a hired programming staff, but taps the capabilities and experience of a large pool of Internet programmers.
- User feedback is facilitated, allowing program testing by a large number of users in a short period of time.
- Program enhancements can be rapidly distributed to users.

[1] a misnomer of "free software". — FSM

6 Summary

This article has explored the long history of PostgreSQL, starting with its roots in university research. PostgreSQL would not have achieved its success without the Internet. The ability to communicate with people around the world has allowed a community of unpaid developers to enhance and support software that rivals commercial database offerings. By allowing everyone to see the source code and contribute to its ongoing development, PostgreSQL continues to improve every day.

About the Author Bruce Momjian is a co-founder of the PostgreSQL Global Development Group, and has worked on PostgreSQL for the past four years. He is the author of "PostgreSQL: Introduction and Concepts", published by Addison-Wesley.

Bruce is the vice-president of Database Development at Great Bridge LLC, a PostgreSQL support company based in Norfolk, Virginia. He has recently spoke at several free and open-source conferences in the US and internationally.

Prior to his involvement with PostgreSQL, Bruce

worked as a consultant in a company that developed custom relational database applications for some of the largest law firms in the US. Prior to this, he was a high school computer science teacher.

The PostgreSQL Global Development Group released PostgreSQL v7.2 in Jan 2002, now it is freely available for download from *http://www.postgresql.org/* and its mirror sites.

Also please try to get your CD-ROM subscription from them to fund their development.

PostgreSQL Commercial Technical Support are available from these companies:

- PostgreSQL, Inc.
  (*http://www.pgsql.com/*) Support for PostgreSQL, Database hosting and promotional materials.
- Software Research Associates
  (*http://osb.sra.co.jp/*) offers [free] Open Source Software Support — a range of services to help customers develop [free] Open Source Software based systems since April 1999.
- Red Hat
  (*http://www.redhat.com/software/database*) offers products, support, training, services and development for PostgreSQL.
- Cybertec Geschwinde &. Schvnig OEG
  (*http://postgres.cybertec.at/*) in Vienna (Austria) offers training courses, support, consulting, cost effective high-end systems and high-availability solutions worldwide.
- dbExperts
  (*http://www.dbexperts.com.br*) in Brazil offers training courses, specialized support for development and commercial products for PostgreSQL in portuguese language.
- Applinet
  (*http://www.applinet.nl/*) offers PostgreSQL consulting services in The Netherlands.
- Command Prompt, Inc.
  (*http://www.commandprompt.com/*) is a managed services and PostgreSQL support company. Located in the Pacific Northwest, they are the author of *"Practical PostgreSQL"* (*http://www.postgreSQL.info/*) from O'Reilly and have been in business for over 4 years. Their specialities are [GNU/Linux], Solaris and PostgreSQL support including custom programming. They are also the developers of the XML/PostgreSQL application server LXP.
- Lerner Communications Consulting
  (*http://www.lerner.co.il/*) Based in Modi'in, Israel, provides programming, support, and training for PostgreSQL and other [free] open-source technologies, including [GNU/Linux], Perl, Python, Apache, Zope, and OpenACS.
- DBAS
  (*http://www.dbas.com.ar*) in Buenos Aires, Argentina offers Database and development support, remote administration and monitoring services, training for database administrators and developers, and consulting for integration with other systems (migrations, interfaces and replication).
- Pate Consulting
  (*http://www.pateconsulting.com*), located in Texas, is a professional services company with over 6 years of Linux experience implementing solid solutions for businesses with Linux and other open source software. These solutions consist of Apache, PHP, PostgreSQL, and Samba. Their specialty is opening proprietary data to businesses using PostgreSQL, Apache, and PHP.
- RON's Datacom
  (*http://www.rons.net.cn*) Located in Wuhan, is the leading company to support free software community in China, offers Zenit Web Application Server, including the commerical support for enterprise-oriented applications.