Postgres NoSQL: Combining Developer Productivity with Enterprise Data Integrity

An EnterpriseDB White Paper

For DBAs, Developers & Database Architects
July 2014

EDB
ENTERPRISEDB

Introduction

Application developers have become the newest culprits in bringing so-called “shadow IT” into the infrastructure. Their tech tools of choice are NoSQL-only database solutions.

Driven by a need to deliver new types of applications faster and respond more directly to business and operations departments employing agile development methods, application developers are seizing on solutions that enable them to work faster. Using NoSQL-only solutions to address specialized demands, they’re adding silos of data to the enterprise environment by spinning up clusters in the cloud or on-premise for new specialized applications based on unstructured and ungoverned data (e.g. social and mobile).

It’s not an unreasonable reaction to rising business pressures. Developers argue they need greater control and flexibility and agility—they can’t wait weeks or even days for new columns, tables and relationships to be created and normalized as the needs of applications evolve.

Developers vs. DBAs

NoSQL-only solutions, however, pose a host of challenges, complexities and even serious risks. This pits application developers against database professionals—the database architects, DBAs and IT executives charged with maintaining data flow, stability and integrity throughout the enterprise.

Postgres solves this dilemma. Advances in NoSQL capabilities alongside longstanding relational database features have given Postgres the ability to support the new unstructured data types and programming methods common to many NoSQL products. In Postgres, you can even combine unstructured data with relational tables, all while maintaining ACID compliance and centralized business processing rules and logic.

It's a matter of doing more with Postgres than ever before.

Postgres Overtakes NoSQL Databases

Postgres is unique in the database world in that it was developed with expansion in mind, making it easy to incorporate new data types, indexing schemes, languages, and much more without de-stabilizing or compromising existing features. As demand for NoSQL capabilities emerged, the open source community behind the PostgreSQL project simply built these new capabilities into the database. In fact, the Postgres community was ahead of curve on one key NoSQL technology – key/value stores – releasing the capability years before the leading key/value store NoSQL-only solution emerged.

What's more, Postgres outperforms the single largest NoSQL-only solution, document database MongoDB. For further validation, the Benchmark tests with the framework and scripts were released to the public for validation and additional testing. [See Addendum for performance results and where to find more information.]

Outperforming MongoDB on common workloads – involving 10 million or 50 million records, the size of a typical MongoDB deployment – is important for Postgres users to understand. Because despite the proliferation of new database technologies, new vendors, fabulous case studies and big venture capital funding rounds, most Postgres users don't need MongoDB or a specialized key/value solution because Postgres can handle most NoSQL use cases as well or better.

Further, there are some other good reasons why enterprises should use Postgres for most of their data challenges and reserve NoSQL-only solutions for very specialized use cases. If not handled properly, and many are not, applications using NoSQL-only solutions can erode the long-term value of enterprise data.

- According to Gartner[1], b**y 2017, 50% of data stored in NoSQL DBMSs will be damaging to the business** due to a lack of applied information governance policies and programs.

1 Does Your NoSQL DBMS Result in Information Governance Debt?, by Nick Heudecker and Ted Friedman, December 3, 2013

Let's examine three primary benefits to using Postgres for building applications with NoSQL technologies.

Ensure Data Integrity

For starters, NoSQL-only databases are not ACID compliant. Some claim ACID compliance in a single document, but achieving but achieving the robust data integrity that enterprises typically require across their data set requires complex application development.

Further, NoSQL-only solutions only store data. They don't *process* data. Data must be brought to the application for analysis. The application (and hence each individual application developer) is responsible for efficiently accessing data, implementing business rules, and for data consistency. Further complicating the issue is the fact that each NoSQL database product uses a different representation for its data and its data access/manipulation language so organizations may find themselves with multiple but incompatible NoSQL solutions.

Pushing the processing out of the database and into the applications—especially in the absence of ACID compliance—is problematic for two primary reasons:

- **Data silos**. Enterprises using NoSQL-only solutions are battling a proliferation of data silos because each application requires its own data store and enterprise data becomes fragmented and loosely governed. This makes it difficult or impossible for enterprises to effectively consolidate, trust and make use of data stored in NoSQL-only database silos. Further, making changes to individual data objects requires identifying and updating across multiple applications.

- **Loss of control**. Enterprises lose control over application logic because each developer is using their solution's individual processing language and applying their own preferred techniques. This can further erode the quality and usability of data from No-SQL only applications.

As a relational database, Postgres supports stored procedures and triggers for implementing business logic for supporting applications. This performs the processing in the database, providing a single

source of business logic truth, easier maintenance of the logic and better performance.

Increase Developer Productivity

Application development cycles have gone down to weeks or days. The ability to reuse code and tap stored functions speeds development. However, NoSQL-only solutions cannot support stored procedures, and NoSQL solutions do not represent one single technology, inhibiting the ability to reuse code, establish standards and find talented resources.

Postgres supports developers in three ways:

- **Structured and unstructured data type support.** Developers using Postgres are no longer bound by the confines of a rigid data model. With JSON/JSONB for supporting and processing document data and the HStore data type for key/value pairs, Postgres provides the flexible data models that developers need to build applications capable of evolving with changing business objectives.

- **Flexibility.** Postgres supports unstructured data stores but then enables developers to apply schema rules to selected data according to business needs. Postgres is also capable of providing enhanced performance using the same eventual consistency techniques used by NoSQL solutions.

- **Centralized data management.** As new data types emerged, Postgres contributors developed capabilities for working with them. With Foreign Data Wrappers, developers can integrate external structured and unstructured data within Postgres and enable Postgres to read and write SQL queries to foreign data sources. There are FDWs for MongoDB, CouchDB, MySQL, Redis, Neo4j and even Twitter and more.

Postgres also allows developers to utilize JSON oriented document syntax directly inside SQL statements complete with a large supply of functions for manipulating JSON data and converting it back and forth with relational data. The ability to create unstructured data stores and combine components with relational tables on the fly is a powerful capability—unheard of in a relational database. At the same time,

Postgres enables enterprises to centralize the business logic around these applications and adhere to standards while enabling developers to reuse code and deliver apps faster.

Extend the Value of Current Resources

In any IT context, new solutions bring new challenges. They require new skill sets, maintenance requirements, operational processes and for applications, deployment, tuning and upgrade demands. NoSQL-only solutions may help developers get up and running faster but the downstream costs on the overall environment and enterprise data strategies are significant.

Consider:

- More data silos mean managing a more fragmented data infrastructure with inhibitions in the flow and trust of data.
- Deploying, maintaining and upgrading applications are more complicated and time-consuming without standardized and easy-to-maintain business logic or processing languages.
- More solutions mean more vendors, even if the software is open source. New software means specialized consultants and support resources.
- Using a single standards-based SQL relational database to solve evolving Web 2.0 challenges means adding to an existing skill set by learning how to do more with Postgres, and realizing greater value from an existing resource–your enterprise database.

Conclusion

Postgres has responded rapidly and aggressively to emerging demands on the database. It's possible to do more than ever thought possible of a standard, enterprise database. And using an ACID-compliant relational database brings benefits across the enterprise, from the executives to the application developers. It's merely time to learn new Postgres skills, not buy a whole new database.

Addendum – Performance Comparison

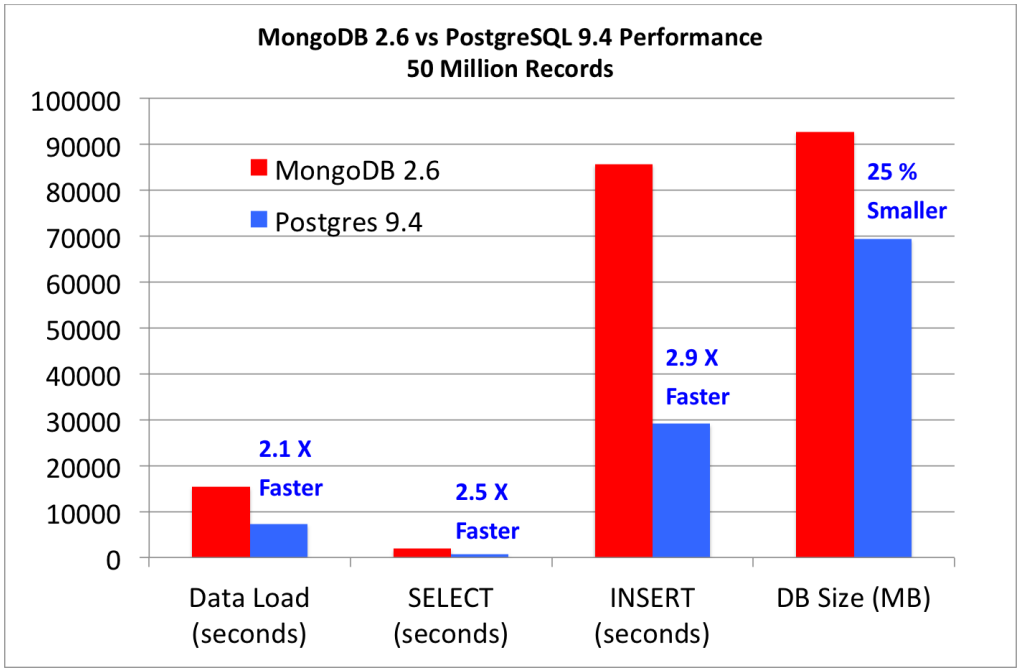
EDB has started to conduct comparative evaluations to help users correctly assess Postgres' NoSQL capabilities.

The initial set of tests compared MongoDB v2.6 to Postgres v9.4 beta, on single machine instances. Both systems were installed on Amazon Web Services M3.2XLARGE instances with 32GB of memory.

Figure one shows the results of our findings:

- Ingestion of high volumes of data was approximately 2.1 times faster in Postgres
- MongoDB consumed 33% more the disk space
- Data inserts took almost 3 times longer in MongoDB
- Data selection took more than 2.5 times longer in MongoDB than in Postgres

Figure 1: Relative Performance Comparison of MongoDB 2.6 with PostgreSQL 9.4



Get Started Today. Let EnterpriseDB help you build and execute your game plan. Contact us at +1-877-377-4352 or +1-781-357-3390, or send an email to sales@enterprisedb.com to get started today on your dependence.

About EnterpriseDB

EnterpriseDB is the leading worldwide provider of Postgres software and services that enable enterprises to reduce their reliance on costly proprietary solutions and slash their database spend by 80 percent or more. With powerful performance and security enhancements for PostgreSQL, sophisticated management tools for global deployments and Oracle compatibility, EnterpriseDB software supports both mission and non-mission critical enterprise applications. More than 2,500 enterprises, governments and other organizations worldwide use EnterpriseDB software, support, training and professional services to integrate open source software into their existing data infrastructures. Based in Bedford, MA, EnterpriseDB is backed by top-tier venture capitalists and strategic investors like Red Hat and IBM. For more information, please visit http://www.enterprisedb.com/.

20140826