ScalienDB and NoSQL in the Enterprise

An Introduction

Cloud computing is resulting in unprecedented data management workloads that are beyond the reach of traditional SQL databases. While SQL databases remain at the heart of corporate information systems, NoSQL has emerged to address these new challenges. This whitepaper describes how ScalienDB offers a best of breed Transactional NoSQL solution for the Enterprise.

New applications like Gmail or Dropbox, run by millions of users, need to store and quickly access large amounts of data in the datacenter. This whitepaper reviews the core requirements posed by such web-scale applications.

Scalability. Successful web and mobile products are used by millions of users. New users, arriving in flocks, continuously generate new data and load. The database must scale with this avalanche of information, keeping the website responsive, without downtime as the database is expanded by adding new servers. Unlike NoSQL databases, SQL databases were not designed with this degree of horizontal scalability in mind.

No single point of failure. Server and network equipment failure is common in the datacenter, but we don't want a server failure to take down the database and hence the entire website. The solution is to use a replicated database and store copies of data on several servers. This allows for individual servers to go down without the entire database going down, allowing for 24/7 availability of the website. SQL databases were designed to run on a single server, and do not inherently replicate data and handle failover scenarios. NoSQL databases, on the other hand, were designed with replication as a primary design consideration.

Agile and Continuous deployment. A web application can be updated hundreds of times a day to quickly test new ideas. To support this demanding release schedule, the database needs to accommodate schema changes, without downtime due to migration. SQL database with rigid relational schemas are not designed for handling this requirement, while NoSQL databases are a natural fit.

Commodity hardware. A database that can take advantage of low cost commodity servers can deliver benefits like scalability and fault tolerance. This requires a distributed database designed to work effectively across multiple servers, also called a shared-nothing architecture. This is a fundamental design principle of NoSQL.

Cloud computing. Providers like Amazon Web Services allow you to quickly launch on-demand virtual compute instances. These are basically another type of commodity hardware. As with physical commodity hardware, this calls for NoSQL database designed to run in a distributed environment like the cloud.

Open source. Open source simplifies the process of evaluating database products by enabling customers to download and test before committing. Once in production, open source access to the database source code allows customers to evaluate new database features transparently and download new upgrades on the fly.

NOSQL

NoSQL databases emerged in 2009 when new challenges in cloud data management were not being met by traditional SQL databases. However, NoSQL is not completely replacing SQL databases in the Enterprise. It is important for decision makers to understand the trade-offs involved.

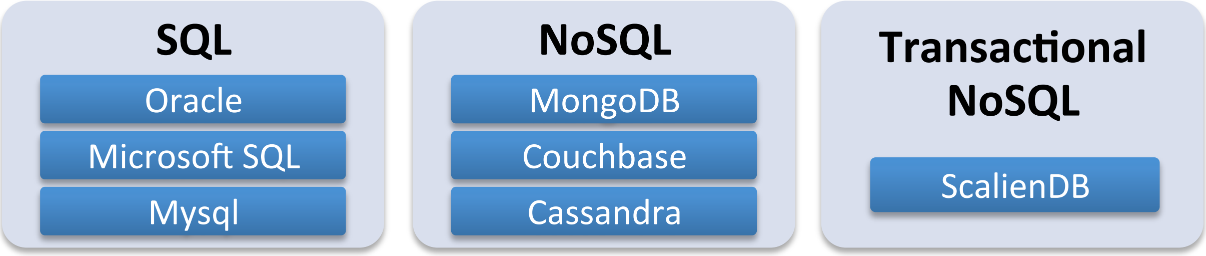
SQL databases have been the backbone of Enterprise computing for the past 25 years. They have delivered immeasurable benefits as computers and the Internet revolutionized the business landscape. It is clear that SQL databases are here to stay, as they are proven tools for a wide range of business use-cases that require transactions, normalization and complex queries, such as accounting and order management. Nevertheless, workloads fueled by the transition of consumer applications to the web have created a new class of workloads. These are characterized by the need for more relaxed schemas, scalability and availability, features not relevant when SQL database technology was being developed in the pre-Internet era.

SQL databases such as Oracle and Mysql were not designed for shared-nothing scalability and replication across commodity hardware. They do not fit well with continuous deployment and agile development practices, as they require a schema before you can add data. If you change your application often, you must change your schema often. On SQL databases this involves slow background processes that block the entire database and usually require application downtime of some sort.

Some of the strongest architectural features of SQL databases such as normalization and complex queries are impractical in a web application. For example, normalized relations in a large scale email application like Gmail would require all user emails to be stored in one email table, all contacts in one contacts table, resulting in huge, unmanageable tables. In cases like this, SQL databases ended up being used as a key-value database with the application developer handling sharding in the application. But this is a complicated, error-prone and hard to manage. ScalienDB takes care of this while retaining many of the benefits of SQL databases.

Unlike other NoSQL databases, **ScalienDB supports transactions and ACID**. This enables ScalienDB to function as a “system of record” primary database with high consistency and integrity properties, while also delivering scalability and having no single point of failure.

Based on this, we offer the following segmentation of the database market:



SQL

Oracle

Microsoft SQL

Mysql

NoSQL

MongoDB

Couchbase

Cassandra

Transactional NoSQL

ScalienDB

SCALIENDB

ScalienDB unites the strengths of SQL databases with the strengths of NoSQL databases. ScalienDB is designed from the ground up to be a **highly available and scalable Transactional NoSQL database**.

Here's a look at the key features of ScalienDB and how they benefit you.

Transactions. ScalienDB in the only open source NoSQL database to support transactions and hence is a "system of record" primary database. Unlike with other NoSQL databases, using ScalienDB there is no need to worry about data loss, data integrity or consistency.

Immediate Consistency. ScalienDB uses the Paxos algorithm for replication. This means that ScalienDB is *immediately consistent* unlike other NoSQL databases that are only *eventually consistent*.

Scalable. ScalienDB is a horizontally scalable distributed database. To store more data you add more servers instead of buying bigger, more expensive mainframe-like servers. ScalienDB supports both read scalability and write scalability, without sacrificing strong transactional semantics.

No single point of failure. ScalienDB is a replicated database without a single point of failure. Customers can chose to run at 3x, 5x or higher replication factors for increased reliability. Unlike other NoSQL databases, ScalienDB can tolerate failures up to a single remaining data server. This reduces operational headaches because the ops team doesn't have to be alerted for single failures.

Schemaless. ScalienDB is a schemaless key-value database with high control over data layout. For example, the customer can chose to compress data to trade speed for space. This gives maximum flexibility and fits in nicely with continuous integration and agile methodologies.

No downtime migration. Changing schema is not free even in a NoSQL database, as data has to be transformed from the old representation to the new one. But in a key-value NoSQL database like ScalienDB this can be performed on a user-by-user basis, on-demand, without downtime.

High performance. ScalienDB is an asynchronous high-performance C++ application. This allows ScalienDB, unlike other NoSQL databases, to supports thousands of concurrent connections while keeping CPU and memory footprint low, making it optimal for high volume websites in the cloud.

Open source. The ScalienDB server and client libraries for C#, Java and Python, and the source code are available for free to download from our website.

Strong roadmap. ScalienDB has been in development for more than 3 years and we're adding exciting new enhancements every week. Thanks to open source, you can evaluate and start using new ScalienDB versions on the fly.

SCALIEN

Scalien offer Enterprise class licensing, training, support, proof of concepts, and custom development for ScalienDB.

For more information email bizdev@scalien.com or call (617) 755-5881.