

xeround

Cloud Database

Part 1 - Technology

Table of Contents

Preface.....	3
Xeround database virtualization technology	3
Virtualized layering.....	3
Virtual partitioning	4
Data replicas.....	4
Quorum based decisions	5
Continuity	5
Low latency.....	5
Scalability and elasticity	6
Availability and resiliency	6
Replication and geographical disparity	7
ACID Compliance	7
Standard interfaces	7
SQL Optimization for distributed computing	7
Summary.....	8

Preface

Cloud technology is currently the leading trend in IT evolution. It addresses the need for supplying IT infrastructure as an OTS (Off the Shelf) product, and allows new economic models, which are more efficient than those that are based on fixed infrastructure.

Cloud technology allows organizations high flexibility in adapting to changing needs. It relies on virtualization, which includes the virtualization of physical servers into virtual servers, virtualization of storage and networking, and virtualization of application frameworks and databases.

Xeround provides a unique solution to database virtualization, which offers a high level of elasticity. Xeround enables databases to grow and shrink as needed, occupying variable sizes of resources – from very small databases that occupy a fraction of a server, up to sizes that span any number of servers.

Xeround database virtualization technology

Xeround's database virtualization is designed for distribution as a core capability, based on the latest distribution theories and concepts. Rather than just another smart sharding or partitioning method, Xeround uses strong distribution techniques, assuming that networking is always required when dealing with distributed services, and therefore it has to play a key role in optimization. This approach is the key to the success of Xeround's database virtualization technology in delivering on its promise.

Virtualized layering

The Xeround database instance is comprised of three layers: Load balancing, SQL engine, and data store. The three layers are fully virtualized and enable scale out, failover and high-availability.

The **Virtualized Load Balancer** is the gateway to the database instance; it deals with the specific level of load balancing required by databases. It maintains

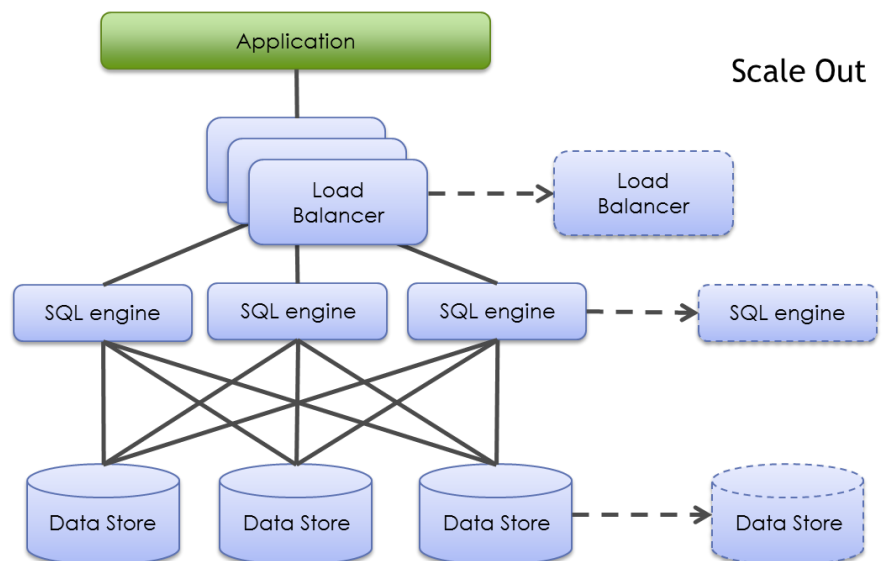


Figure 1

transaction stickiness and balances the load between the SQL front-end instances. It also deals with failover of SQL engines.

Virtualized SQL engine is the MySQL façade implementation. This layer is responsible for breaking a SQL statement into distributed parallel plan of execution, gathering the results from the different processes and assembling the final result for the application.

The **Virtualized data store instances** are distributed data stores, which maintain the data in its different representations for the database. These instances implement DHT, Distributed BTree and Object Store, which are the basic data services required by the database.

As seen in Figure 1, an application (or applications) can be connected to any or all of the load balancers. Each load balancer is connected to the number of SQL engines that are available for the database, and all the SQL engines are connected to all the data stores allocated to the database.

The number of load balancers, SQL engines and data stores are not fixed, and depend on the requirements and size of the specific database. All the layers are virtual, elastic and can scale as needed. For example, if more space is needed but traffic is low, only additional virtualized data stores are added, while the number of virtualized SQL engines and virtualized load balancers may be reduced (as long as high-availability is maintained). Elasticity is maintained automatically for every layer, and the database re-balances itself upon changes.

Virtual partitioning

Xeround introduces the concept of “virtual” partitions, where data partitions are decoupled from the physical resources. Xeround massively partitions the data into hundreds up to thousands of self-managed partitions, which are allocated to the available resources in a well-balanced manner. Whenever a change occurs, either in the case of data growth or with available resources, the virtual partitions are autonomously re-allocated to the physical resources. The virtual partitioning mechanism inherits the advantages of the ‘**Shared Nothing**’ architecture, yet overcomes the disadvantages of partition events.

Data replicas

Within the virtual partition structure, Xeround keeps a (configurable) number of replicas of each data element, as illustrated in Figure 2. Keeping more than one copy ensures high-availability and resiliency.

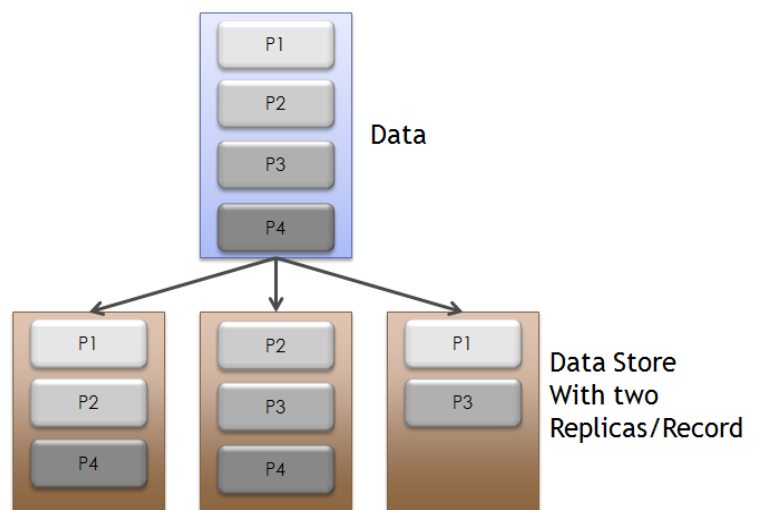


Figure 2

Quorum based decisions

Complementing the multiple replicas, all Read and Write operations are based on quorum voting to ensure consistency:

- Assuming three replicas and a read quorum of two, the system will issue three distributed read requests and will return a reply once it receives two identical results, without having to wait for the third reply.
- Assuming three replicas and a write quorum of two, the system will issue three distributed write requests and will consider the operation successful once it receives two confirmations. In this case, the two replicas will attempt to update the third replica asynchronously. If the update fails after a (configurable) number of retries, full data sync will occur to prevent data corruption.

Continuity

Xeround architecture and processes ensure that the service is Always-On. In addition to managing high-availability and auto-healing, Xeround guarantees continuous service during the following processes:

- **Schema changes** – The database treats any change to the schema, such as adding tables, columns or changing index definition as a Write operation; hence it is totally operational throughout the process.
- **Resource adjustment** – During addition or removal of physical resources, the virtual partitions reallocate autonomously, without affecting the service.
- **Scaling Process** – The database transparently performs any addition or removal of components to accommodate changes in size or throughput needs.

Low latency

One of the key challenges for a high-performance database management system is maintaining consistently low latency. Xeround can guarantee **predictable response times and low latency** for data reads and writes across the entire system. Reads and writes are processed in the same manner and are non-locking in nature. Unique to the industry, data managed by Xeround can be read and updated at the same speed. Xeround has been engineered so that both response time and latency remain unaffected while the database scales, even during unforeseen failures or outages.

Scalability and elasticity

Xeround is the only cloud-based database solution that offers linear scalability for both throughput and database size without any service downtime. When resources are added or removed on existing or new hardware, Xeround adjusts the layout of its virtual partitions on the fly, according to available resources. Figure 3 illustrates this scalability model.

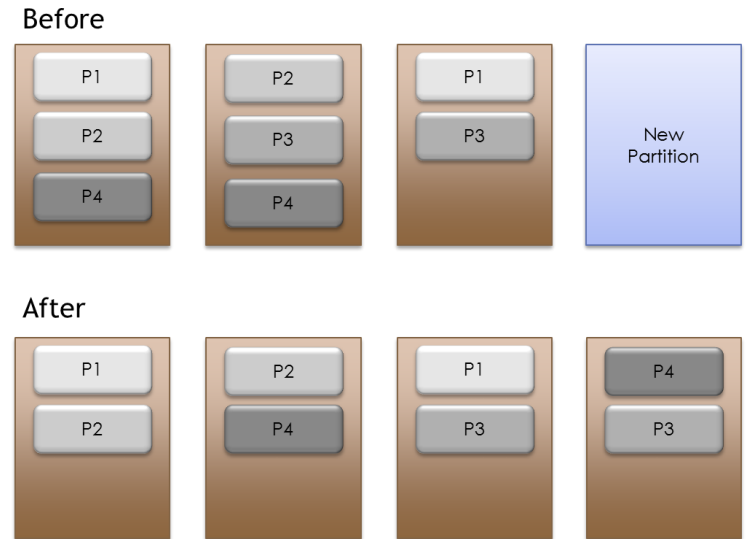


Figure 3

Availability and resiliency

Xeround's unique architecture ensures high-availability and full resiliency in case of software or hardware failure. Xeround has **no single point of failure** and is resilient to failure of any software component across data grid nodes, sites, networking elements, external data sources and virtualization technologies. Figure 4 illustrates this high-availability model.

Figure 4 illustrates this high-availability model.

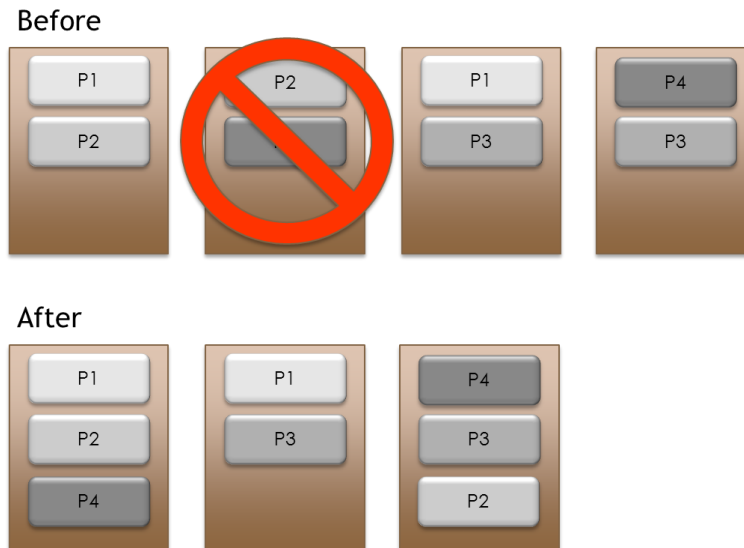


Figure 4

Replication and geographical disparity

Xeround supports distributed deployments where multiple copies of the database run at multiple locations, automatically taking care of synchronization in any combination of active and passive modes.

Figure 5 illustrates the Xeround partition model in a two-site deployment.

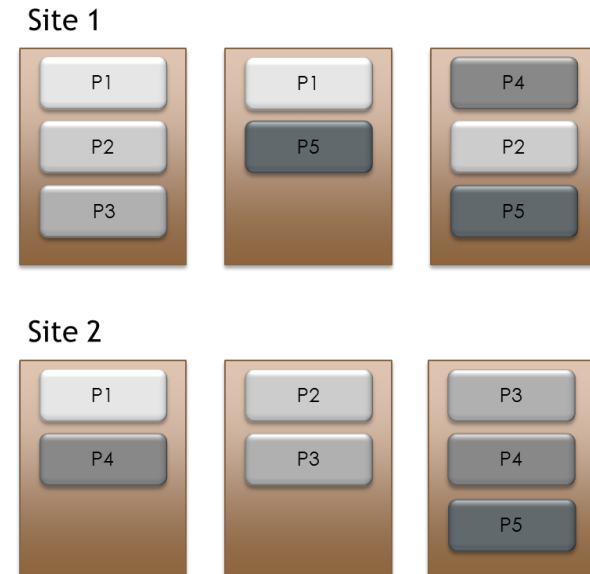


Figure 5

ACID Compliance

Xeround is fully ACID-compliant, providing full support for transactions, including distributed transactions spanning multiple partitions.

Standard interfaces

Xeround is accessible via standard SQL interfaces; it does not require code changes on the application front, nor does it impose any tie-in to any specific cloud vendor. The current version provides MySQL front-end compatibility, enabling transparent database transition. More front ends are being developed and will be available in coming Xeround versions.

SQL Optimization for distributed computing

Xeround Database optimizes the execution of SQL statements to get the maximum from the foundations it is built on. These services are distributed and highly parallel. In this sense, it is very different from MySQL, which is built more for I/O and its limitations. Xeround lays an emphasis on parallelizing processes and offloading tasks from the front end, and executing them at the data store

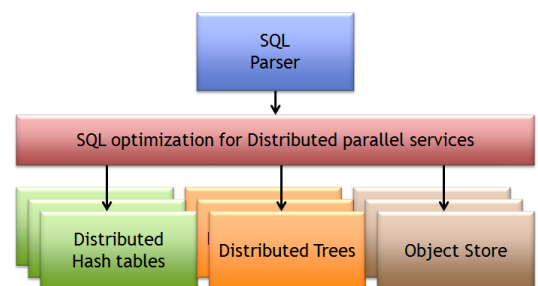


Figure 6

level whenever possible. This approach is in many ways similar to the Map/Reduce concept. This approach accelerates heavy processes such as large JOINS or scans of very large databases. The high level of parallelism is also visible in the throughput that the database can deal with, and the number of concurrent requests it can handle, which are much higher than standard databases.

Summary

Xeround offers an innovative and comprehensive database solution that addresses the challenges of cloud-based data management, emerging technologies and markets.

Xeround's solution, based on unique patented technology and solid development process, is capable of handling and exceeding any requirements of scale or extremely high-availability, without compromising functionality, ACID compliance and SQL support.

Combining virtual partitioning, data replicas and distributed architecture, the Xeround cloud database solution provides linear scalability, elasticity and high-availability, while its MySQL compatibility and its management layer allow an easy migration with no change in code or architecture.

Create your free cloud database at www.Xeround.com