Carrier Grade Service Execution Environment

Delivering Continuously Available Value- Added Services:

Sun GlassFish Communications Server and
MySQL Cluster Carrier Grade Edition

A MySQL® Technical White Paper by Sun Microsystems

Table of Contents

1 INTRODUCTION 3

2 CARRIER GRADE SERVICE EXECUTION ENVIRONMENTS 3

2.1 High Availability 4

2.2 High Throughput & Instant Scalability 4

2.3 Minimizing Latency 4

2.4 Event Driven Architecture 5

3 SUN GLASSFISH COMMUNICATIONS SERVER + MYSQL CLUSTER CARRIER GRADE EDITION 6

3.1 Benefits 7

3.2 High Availability 7

3.2.1 Geographical Redundancy 9

3.3 Horizontal Scaling and High Throughput 10

3.3.1 Increased throughput via efficient use of the network 12

3.3.2 Minimizing Latency 12

3.4 Management of Carrier Grade Service Execution Environment 13

4 CONCLUSION 14

5 REFERENCES 15

1 Introduction

The convergence of previously separate communications networks onto a single IP-based network is revolutionizing today's telecommunications industry. Convergence promises the opportunity for Communications Service Providers (CSPs) to reach new markets with compelling value-added communications services, delivered over fixed and wireless networks via Service Delivery Platforms.

Service Delivery Platforms are designed to accelerate service deployment and reduce cost per subscriber by laying the foundation for consistent value-added service creation, execution and management. By taking advantage of Service Delivery Platforms in the deployment of services over converged networks, CSPs can differentiate themselves from their traditional competitors to increase ARPU and reduce customer churn. They can also combat new market entrants arriving from the world of the Internet which is more accustomed to the rapid delivery of new services, based on open source and open standards technology.

Any opportunity comes with challenges, which must be addressed by CSPs as they seek to exploit the market opportunity presented by convergence. New services delivered over converged networks comprise ever-richer applications that can be delivered to more users requiring greater levels of customization. As a result, one of the most critical factors in successful service adoption is the need for massively scalable and continuously available service execution and data management environments, with real-time performance capabilities.

Highly available application servers and carrier-grade databases are at the core of any Service Delivery Platform. The database is especially critical as more CSPs seek to consolidate subscriber and service data into single unified stores. The benefits of such unified stores cannot be underestimated when considering their value in delivering operational efficiency, time to market for new services, integration with legacy environments and interfaces to OSS/BSS platforms.

Investments in SDPs represent one of the most important strategic decisions CSPs will make as they attempt to unlock the opportunity presented by network convergence. Ensuring the enabling service execution environment is highly available and scalable while capable of delivering real time performance will be the determining factor in success or failure. In this paper we explore how the Sun GlassFish Communications Server, developed under the SailFin project, and MySQL Cluster Carrier Grade Edition enables CSPs to cost-effectively address the core requirements of SDP deployments in the delivery of new services over converged networks.

2 Carrier Grade Service Execution Environments

The Service Execution Environment is a key layer of any SDP, and the application servers and databases are, in turn, key components of this layer.

To support the rapid adoption and monetization of new services over converged networks, the applications servers and database need to offer carrier grade capabilities, which are defined by the following attributes.

2.1 High Availability

Providing continuous service availability requires a platform with proven capabilities to withstand hardware and software failures, as well as accommodating live upgrades to both underlying infrastructure components and data structures.

Any failures of hardware or software must be instantly detected and the affected services failed over immediately to surviving nodes. The time to recovery needs to include both the business logic and service data being restored to an operational state. Service state should be persisted and capable of being distributed geographically so that users do not experience service disruption. Self-healing mechanisms should allow recovered nodes to re-join the environment as quickly as possible without manual intervention.

As services evolve, any upgrades to the underlying infrastructure must be accommodated without service downtime. Such upgrades can include adding or removing capacity from the service execution environment, or updating the underlying hardware and software components.

Adding new services to an existing environment potentially requires changes to existing data structures. Such changes should not incur downtime so that existing services continue to run, while new services can be deployed quickly and efficiently.

By employing the mechanisms discussed above, the service execution environment is capable of providing continuous availability in the event of failures or system upgrades.

2.2 High Throughput & Instant Scalability

In addition to high availability, high-performance is a critical requirement in order to accommodate the massive volumes of communication service requests and database transactions typically found in the communications industry. Telecom performance requirements are usually in the range of tens of thousands of requests per second, executed with millisecond response times. These performance requirements need to be met while still maintaining continuous availability, even in the event of a network or node failure.

While the rapid deployment of new services is critical to any CSP, it is almost impossible to predict their level of adoption. To reduce investment and financial exposure, it is prudent for any infrastructure supporting the service to start small, but allow for rapid and low cost scalability in the event of mass-market adoption.

Scale-out allows for just such an incremental approach to increase capacity. Throughput can be increased by scaling out the infrastructure on multiple processing nodes to handle increasing numbers of simultaneous service requests and database operations, using low cost, commodity hardware and open source, open standards software.

2.3 Minimizing Latency

Communications services are characterized by the requirement for predictable and fast response times to user or network requests. To achieve such requirements, it is necessary to use real time components such as carrier grade databases.

Response times are dictated by the processing times of sub-systems within the network, I/O operations and the speed of network connections. Carrier Grade databases have been able to achieve low latency by using main memory data stores to limit disk I/O, and use asynchronous operations to write log files to disk. The ability to batch operations before sending to nodes for processing across a network also helps to reduce the impact of network latency.

2.4 Event Driven Architecture

Communication services are inherently asynchronous and event-driven. The communications core network generates various events that must be efficiently handled by the service execution environment.

A Java EE application server that supports Java Message Service API (JMS) and provides an Enterprise Service Bus handles this requirement effectively. JMS is a messaging standard that allows Java EE applications and components to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous.

Open Enterprise Service Bus (Open ESB) hosts a set of pluggable component containers, which integrate various types of IT assets. A message oriented asynchronous publish/subscribe based model is able to deliver events to their intended destination efficiently.

Used together, these components provide an event-driven architecture that will support carrier grade requirements for handling events efficiently, e.g., with predictable response times with low latencies.

3 Sun GlassFish Communications Server + MySQL Cluster Carrier Grade Edition

The Sun GlassFish Communications Server, developed under the SailFin project, and MySQL Cluster Carrier Grade Edition allow CSPs to cost effectively build out a Service Execution Environment for the delivery of continuously available and highly available value added services, delivered over converged IP networks.

Sun GlassFish Communications Server is a SIP application server and represents the outputs of an on-going joint development by Sun Microsystems and Ericsson. Sun GlassFish Communications Server provides Java EE 5 platform compatibility, industry standard Web Services interoperability, service composition with OpenESB, high availability for converged SIP and Java EE applications and industry leading performance [3][4].

MySQL Cluster Carrier Grade Edition (CGE) is the industry's only true real-time, fault-tolerant database with the flexibility of a relational database and the low TCO of open source. Its “shared-nothing” distributed architecture ensures no single point of failure while delivering millisecond response times and 99.999% data availability, even when serving tens of thousands of transactions per second. MySQL Cluster Carrier Grade Edition (CGE) has enabled users to accelerate time to market while reducing risk and provide complete control over performance and scalability as business demands shift, at a fraction of the cost of proprietary solutions [1][2].

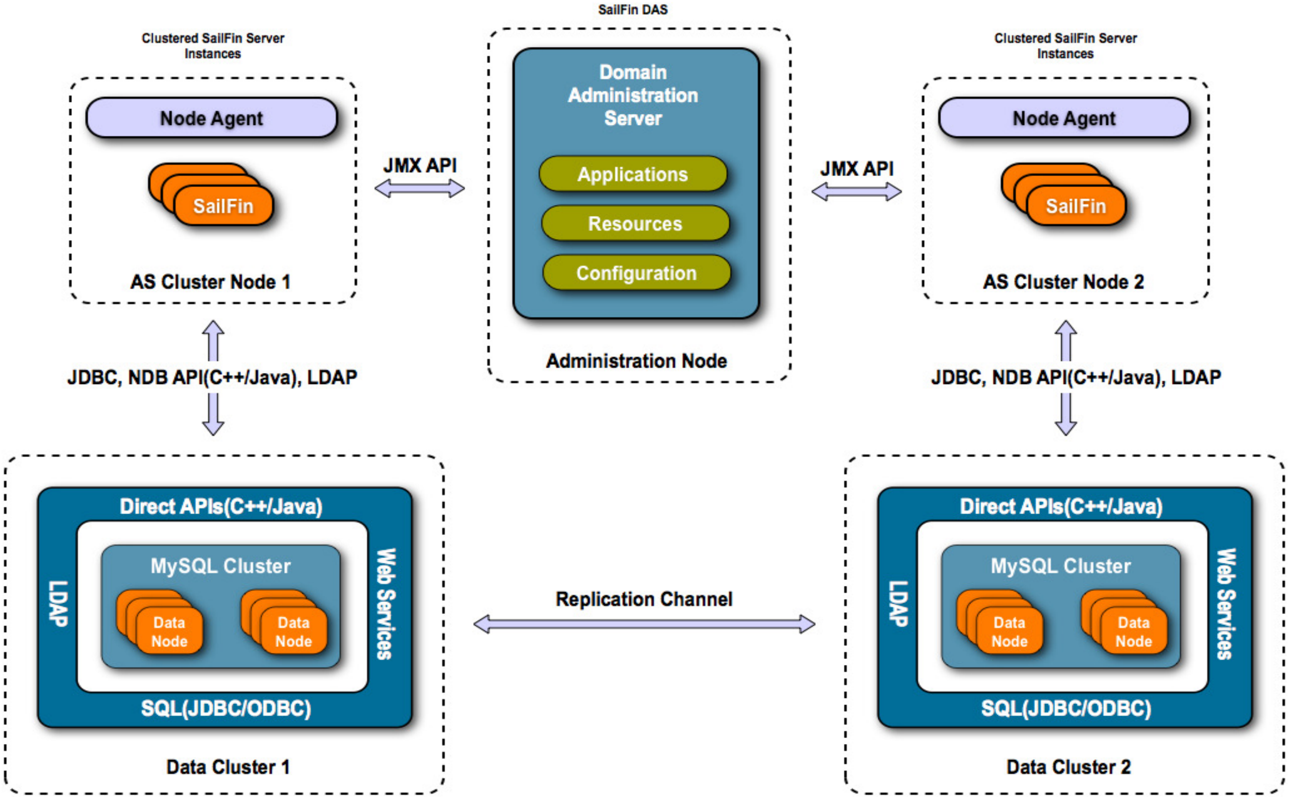

Figure 1 Sun GlassFish Communications Server (SailFin) + MySQL Cluster System

Application developers can easily integrate applications using their preferred database-independent method. MySQL Cluster Carrier Grade Edition provides multiple, simultaneous data access interfaces susing a variety of means including SQL (JDBC/ODBC), Direct APIs (C++/Java), LDAP, and Web Services. Such an approach ensures CSPs have access to a consolidated data store serving multiple new and legacy applications.

MySQL Cluster CGE automatically and transparently distributes data across multiple data nodes. It delivers fast failover times ensuring applications can recover quickly. Cluster nodes automatically restart, recover, and configure themselves in case of failure providing a “zero admin, self healing” high availability solution.

The shared-nothing architecture means applications can start small and make incremental investments to increase capacity as demand grows. Typical response times are in the range of a few milliseconds and the database can be scaled to handle tens of thousands of transactions per second.

3.1 Benefits

When deployed together, Sun GlassFish Communications Server and MySQL Cluster Carrier Grade Edition inherently provide high availability, load balancing and horizontal scaling capabilities.

For example, combining Sun GlassFish Communications Server’s Converged Load Balancing and Data Centric Rules with the ability of MySQL Cluster to partition data, service requests can be directed to a specific server instance and shard of data, (i.e. database partition) [1][3]. Such an approach thus improves overall throughput and provide lower service response times.

Carrier Grade Characteristics	Sun GlassFish Communications Server	MySQL Cluster CGE
High Availability	Cluster of instances; in-memory session data replication	Shared-nothing architecture; No single point of failure; Sub-second failover; at least 2 replicas within one cluster; Geographical Replication between clusters
High Throughput	Dynamic clustering with Shoal/JXTA; scale by adding more clustered instances; Data Centric Rules	Linear scalability; scaling out by adding Data and Application Nodes using COTS hardware; transaction and operations batching; user defined partitioning/distribution aware clients
Low Latency	JVM tuning (garbage collection, heap size etc); tuning threadpools; socket buffers	Native NDB APIs provides predictable and low latency access; real time extensions (locking threads to CPU, real time priorities); in-memory tables avoiding disk I/O; Scalable Coherent Interface transporters
Failures & Self Healing	Built in converged load balancing and failover; self healing using management rules; memory/cpu overload protection	Automatic load balancing; automatic failover and node recovery; arbitration protocol to avoid “split brain” issues
Online Upgrades & General Maintenance	GUI/CLI admin(DAS); JMX, rolling upgrade	CLI console; Online: backup, schema change, adding nodes and rolling node upgrades

3.2 High Availability

When combined, Sun GlassFish Communications Server and MySQL Cluster CGE provide continuous service execution availability by implementing a redundant service layer through clustered

SIP application server instances and a fault tolerant data storage layer provided by redundant MySQL Cluster nodes.

Sun GlassFish Communications Server provides runtime dynamic clustering and in-memory replication of HTTP and SIP requests and session data (both HTTP/SIP session data and stateful session bean data) across multiple application server instances.

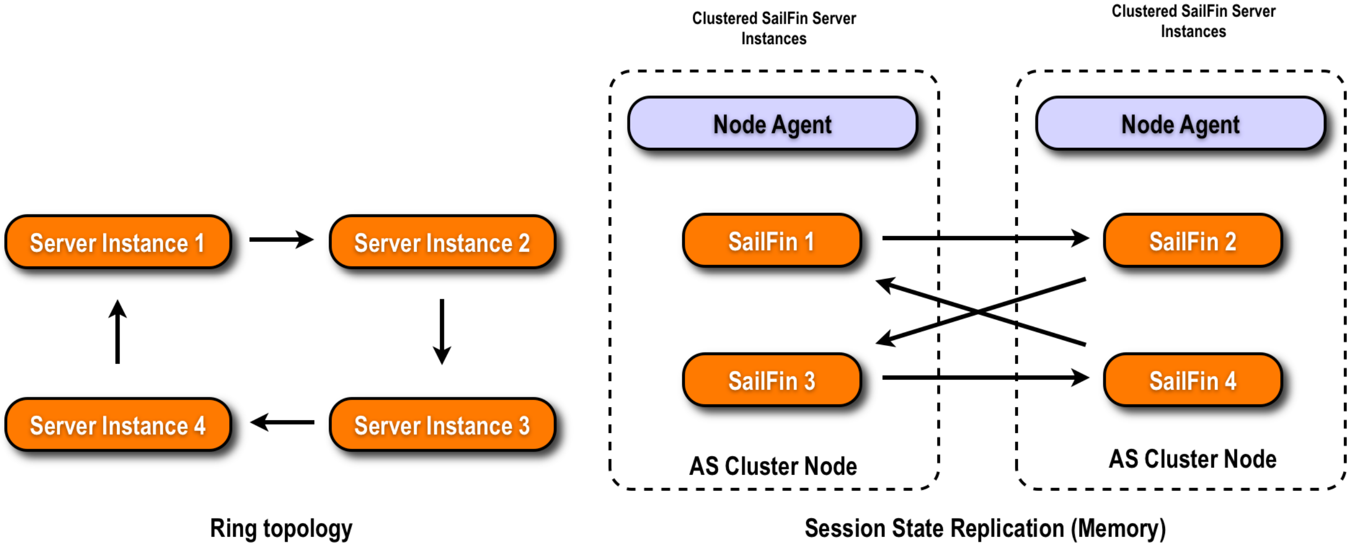

Figure 2 Sun GlassFish Communications Server (SailFin) in-memory clock-wise (based on instance name) replication

Sun GlassFish Communications Server's converged load balancer accepts both HTTP/HTTPS and SIP/SIPS requests and forwards them to application service instances. If an instance fails, becomes unavailable (due to network faults), or becomes unresponsive, the load balancer re-directs requests to existing, available machines.

The load balancer can also recognize when a failed instance has recovered and redistribute the load accordingly. An Application Server instance can be a dedicated load balancer or each instance in a cluster can participate in load balancing, in which case the cluster is described as self-load-balancing.

By distributing the workload among multiple physical machines, the load balancer increases overall system throughput. It also provides higher availability through failover of HTTP and SIP requests.

MySQL Cluster Carrier Grade Edition provides a shared-nothing architecture with no single point of failure and automatic failure detection with sub-second failover time. Synchronous replication is used to propagate transaction information to all the appropriate database nodes so applications can automatically fail over to another node quickly and always have access to the latest version of the data.

When failed nodes are detected they are automatically restarted using a node recovery protocol. This protocol provides the restarting nodes with the necessary data from the surviving nodes in order to become current and active participants in the cluster.

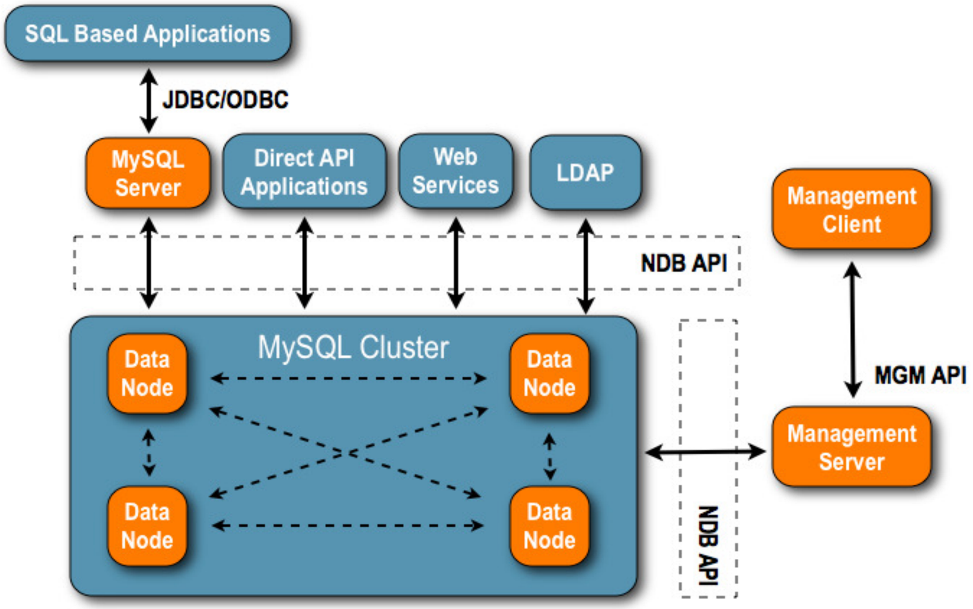

Figure 3 MySQL Cluster Components

MySQL Cluster CGE supports online schema upgrades, such as adding and removing tables or indexes, as well as changing table schemas themselves, e.g. adding or removing columns. In MySQL Cluster CGE, adding and removing tables and indexes are performed as normal operations on the active cluster, and do not require any rolling upgrade of the cluster. Such features are critical when CSPs seek to deploy new services, as these will inherently require updates to the underlying data structures.

MySQL Cluster CGE also supports online software upgrades, which are performed using rolling restarts.

In addition, MySQL Cluster CGE allows a backup to be made on a running cluster. The online backup is a consistent cluster-wide snapshot of the database that can be archived and copied to a secure, remote location.

3.2.1 Geographical Redundancy

By leveraging MySQL Cluster CGE's replication mechanisms, services deployed on Sun GlassFish Communications Server will be able to survive site wide outages or site-specific performance degradations.

MySQL Cluster CGE supports multiple replication topologies like Active-Active or Active-Standby. Active-Active replication saves hardware and administration costs over Active-Standby replication, as all clusters play an active role in serving user requests, while simultaneously acting as a standby for another cluster.

An Active-Active Ring is a cluster topology used for both high availability and high performance, based on Active-Active replication. Client applications can connect directly to the relevant active cluster during normal operation, and switch over to the standby cluster in the event of non-availability of the active cluster.

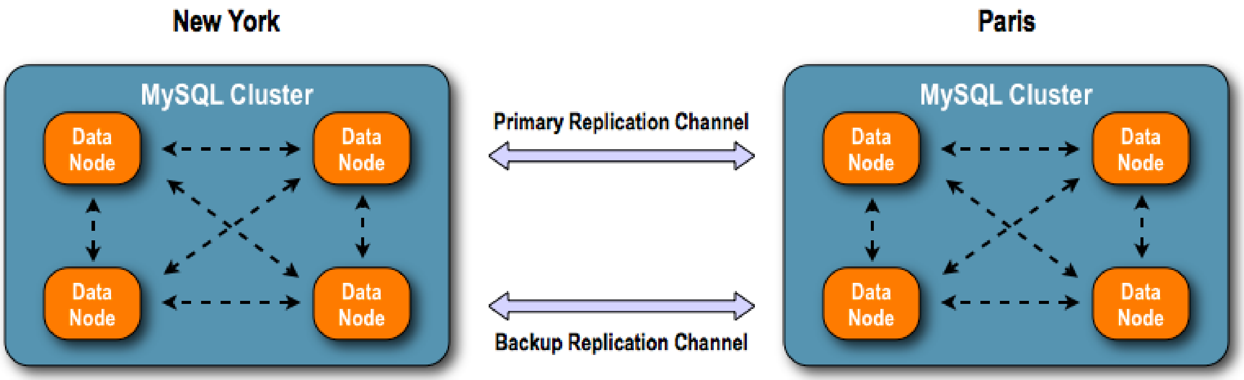

Figure 4 MySQL Cluster Geographical Replication

In the event of a cluster failure, the recovered cluster can be merged back into the Active-Active Ring. As geographical replication is asynchronous in MySQL Cluster CGE, any data at the failed cluster that may had not been replicated before failure can be extracted using binary log tools, and then subsequently applied to its (currently running) standby cluster.

3.3 Horizontal Scaling and High Throughput

Scalability of service execution is achieved by allowing for the addition of application server instances to a Sun GlassFish Communications Server cluster, thus increasing the throughput of the system. The Sun GlassFish Communications Server load balancer plug-in distributes requests to the available server instances within the cluster. No disruption in service is required as an administrator adds more server instances to a cluster [5][10].

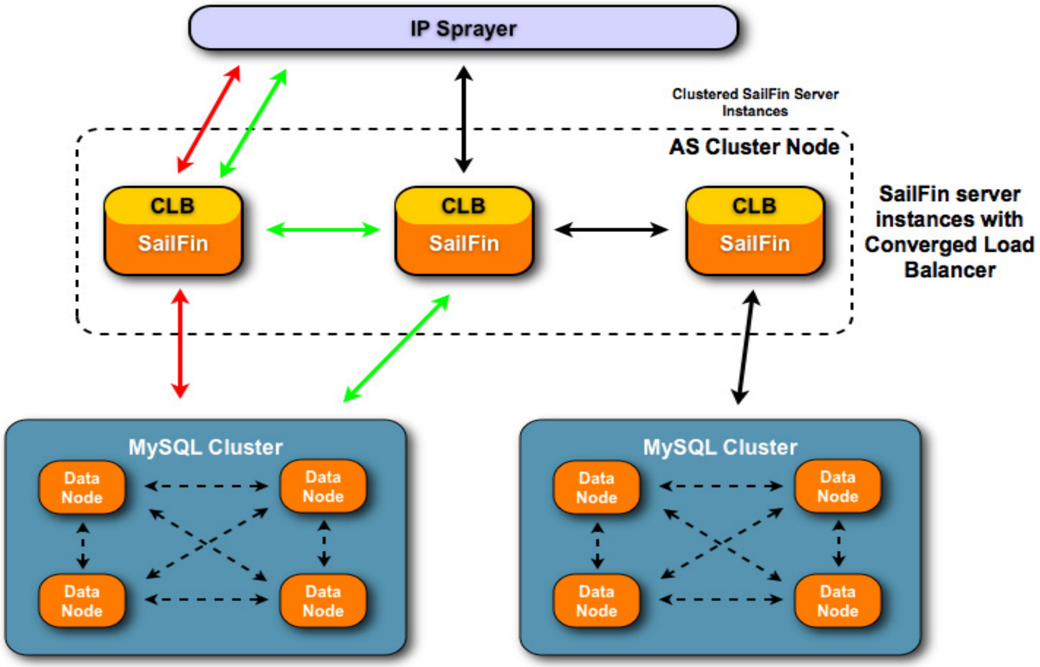

Figure 5 Sun GlassFish Communications Server (SailFin) Converged Load Balancer

Sun GlassFish Communications Server's built-in converged load balancer features a load-balancing mode using a consistent-hash algorithm which computes a hash-value based on the service request made. Combined with Data Centric Rules and using data partitioning, specific requests, i.e. those originating from a specific set of users can be directed to a specific server instance and database cluster thus improving the overall performance and service response times, while reducing network load.

To accommodate increasing application loads and data capacities, SQL and data nodes can be added on-line to a running MySQL Cluster

The MySQL Cluster CGE architecture enables it to scale in a linear fashion by leveraging data partitioning features. Using such partitioning, data can be efficiently accessed on a single Data Node without the need for intercommunication within the cluster in order to satisfy a result set or look up.

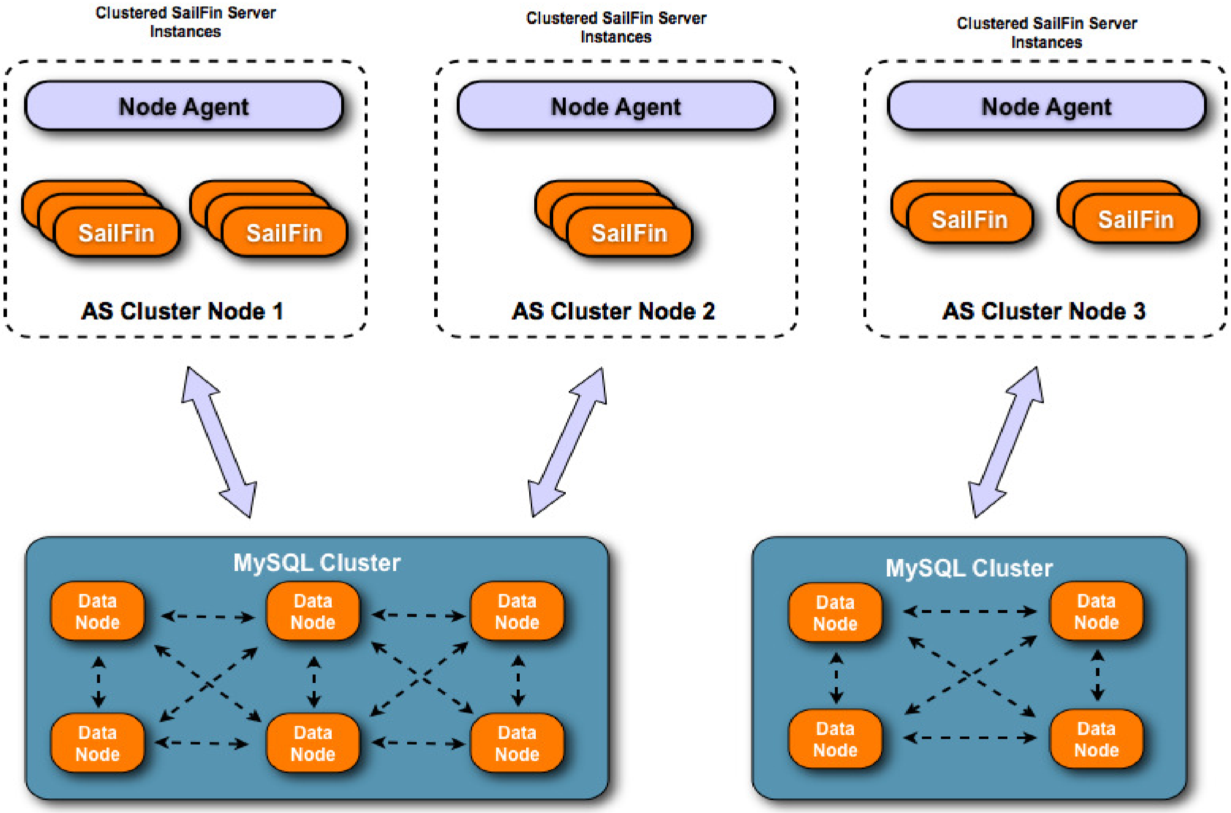

Figure 6 Horizontal Scaling of Sun GlassFish Communications Server (SailFin) and MySQL

In an Active-Active topology, all MySQL Clusters are used, as each cluster is responsible for updates to its data partition (that is, its range of data), while also acting as a standby for a different cluster's data partition. This way, database operations can be load balanced over different clusters, while cluster replication ensures that a cluster can be failed-over in the event of some problem.

A combination of clustered application servers and databases makes a very scalable system. One can scale the number of applications servers to handle more SIP/converged services and distribute processing needs between the application servers. Data accesses are load balanced between the different data nodes in the database cluster.

3.3.1 Increased throughput via efficient use of the network

In a clustered database, the throughput of the network interconnect is often a limiting factor in determining the database throughput. While certain applications may become CPU or memory-bound; often the throughput or response-time of an application can be improved by more efficient use of the network.

The cost of transporting the transaction between the client and the data nodes typically exceeds the cost of any individual operation. Tests have found[1] that there was a greatly increased execution overhead for transactions consisting of low numbers of operations. To reduce the proportional transport overhead, it's best to maximize the payload. The more operations per transaction, the lower the effective cost of each individual operation.

The NDB-API is provided as part of MySQL Cluster CGE to allow low-level access in defining how transactions and operations are buffered and batched before being sent over the network. Some of the NDB-API features that can be used to improve utilization of the network include [1][11]:

- Adaptive buffering of transactions
- User-defined partitioning and distribution-aware clients
- Batching operations in a transaction
- Batching transactions

3.3.2 Minimizing Latency

Performance tuning can significantly improve user experience for service response times and should ideally be performed in the following sequence [8]:

- Application tuning
- Application Server tuning
- High Availability Database tuning
- Java Runtime System tuning
- Operating System tuning

Sun GlassFish Communications Server provides many options to tune performance such as tuning the size of socket buffers, or client/server thread pools. Another tuning option is to use Sun GlassFish Communications Server's Overload Protection Manager that monitors system attributes like CPU and memory usage.

The OLP manager intercepts every request/response (SIP/HTTP) and only allows messages to be processed if the system is still within the pre-configured threshold limits. In the event of a sudden increase in load, the OLP manager is able to ensure that the Quality of Service (e.g., Calls per second) is maintained.

MySQL Cluster CGE is primarily leveraged as a main-memory database to ensure performance SLAs are met, although it also supports disk-based data. As a result of the in-memory design, MySQL Cluster CGE is able to limit disk-based I/O bottlenecks by asynchronously writing transaction logs to disk. Therefore, typical response times for Carrier Grade are in the range of a few milliseconds.

Another way in which telecom applications improve their performance is through the use of the native data access available through the NDB API. This API provides applications the fastest and lowest-level data access available via a non-SQL interface. This typically results in latencies for reads and writes in the order of a few milliseconds.

[1] http://www.mysql.com/why-mysql/white-papers/mysql_wp_ipl_multiplay.php

Improvements have also been made to the protocol used to communicate and satisfy requests between nodes. The improvements have resulted in lower latency and response times by making messages smaller and more efficient. An added benefit of these optimizations is for application nodes to now access data nodes over a WAN with acceptable response times.

3.4 Management of Carrier Grade Service Execution Environment

Monitoring and managing carrier grade Service Execution Environments potentially creates a significant system administration overhead which increases the costs of service delivery and reduces the ability to apply scarce admin/DBA resource to the development and deployment of new services. Both Sun GlassFish Communications Server and MySQL Cluster are designed to reduce this overhead with self-managing, self-healing architectures and simplified systems administration tools.

Sun GlassFish Communications Server's self-configuring design reduces the complexity of routine management tasks through automation. System Administrators can automate system monitoring during live run-time conditions to detect and automatically recover from failures. Even security threats are detected and self-protective actions are taken in response. The systems management framework has been developed using standard technologies.

Self-Management [6] is implemented via a set of management rules. Different rules can be created and implemented for various event types such as:

- Lifecycle events
- Monitor events
- Log events
- Trace events
- Timer events
- Notification events

From a single administration console, Systems Administrators can also perform and monitor the following:

- Create and administer deployed instances and clusters as a single entity.
- Manage distributed deployments through centralized administration
- Dynamically grow or shrink a cluster by adding or removing application server instances
- Deploy and automatically update the load balancing plug-in which is responsible for monitoring cluster health and load balancing across available instances
- Provide remote secure management using a browser-based administration console and feature-equivalent scriptable command-line interface.
- Enhanced application monitoring, visualization and diagnosis.
- Built-in management rules and triggers can be expanded programmatically.
- Manage and monitor the application server leveraging existing JMX-enabled enterprise management tools

MySQL Cluster CGE includes easy to use tools for administering the clustered database environment. Command line tools provide monitoring of database nodes, controlling access to applications, and creating / restoring backups.

As discussed previously, MySQL CGE allows standard database management tasks to be performed on a live cluster, without downtime or disconnection of users. Back-ups can be taken on-line. Schema updates are supported and rolling restarts can be implemented to accommodate software upgrades. Using node recovery protocols built into MySQL CGE, any failed node can automatically rejoin the cluster when in a state to recover.

4 Conclusion

Converged services are important to attract new users and increase Average Revenue Per User. The infrastructure to handle these services need to provide carrier grade characteristics at every level in order to provide scalable, continuously available communications services.

Speed matters! Amazon found that every +100ms increase in response time cost them 1% in sales while Google found that +0.5 seconds response time in search page generation reduced the traffic by 20%. [2] [3]

Sun GlassFish Communications Server, developed under the SailFin project, combined with MySQL Cluster Carrier Grade Edition enables communications service providers to develop, deploy and maintain a highly available carrier grade service execution environment where predictable response times and low latency are critical for business success.

MySQL Cluster's shared-nothing distributed architecture with automatic sub-second failover provides high availability with synchronous replication so services can failover to another data node quickly. It's in-memory design with automatic and manual data partitioning allows for a very flexible model for increasing performance and lowering overall data request times.

Sun GlassFish Communications Server's built-in converged load balancer and in-memory replication provides high availability with automatic failover for deployed services. Converged load balancing using consistent-hash and Data Centric Rules ensures that service requests can be directed to a specific server instance and database cluster, thus helping to improve overall performance and lowering service response times.

The fusion of Sun GlassFish Communications Server and MySQL Cluster CGE addresses all of the needs to successfully provide a robust service execution and data management environment for the delivery of value added services over converged IP networks. Services can be independently scaled online in the Application Server and Database tiers while also providing continuous service availability that satisfy end users high expectations for service responsiveness and availability.

You can freely download both Sun GlassFish Communications Server and MySQL Cluster CGE under open source licenses to evaluate their capability in delivering continuously available services. You can freely develop and deploy next generation value added services on Sun GlassFish Communications Server and MySQL Cluster, and then access services to support you when your services become revenue generating and mission critical to your business.

Such an open approach enables CSPs to accelerate their time to market and reduce the risks associated with delivering Value Added Services over converged IP networks.

[2] Make Data Useful, Greg Linden, Stanford Data Mining 2006 presentation

[3] http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html

5 References

[1] Database Driven Development for Carrier Grade Systems white paper, MySQL AB 2007.

[2] Building Subscriber Databases with MySQL Cluster Carrier Grade Edition (For Converged Telecommunications Networks), Technical Whitepaper, MySQL AB, 2007.

[3] Project SailFin: Architecture, Applications, and Roadmap, Kristoffer Gronowski, Senior Specialist, Ericsson, Binod PG, Senior Staff Engineer, Sun Microsystems., TS-5866

[4] GlassFish presentation Alexis Moussine-Pouchkine, GlassFish Team, Sun Microsystems, May 2008

[5] Project Shoal - A Generic Clustering Framework, Shreedhar Ganapathy (GlassFish), Mohamed Abdelaziz (JXTA) , Jan 2007

[6] Sun Java System Application Server 9.1 Administration Guide

[7] Sun Java System Application Server 9.1 High Availability Administration Guide

[8] Sun Java System Application Server 9.1 Performance Tuning Guide

[9] Sun GlassFish Communications Server Administration Guide

[10] Sun GlassFish Communications Server High Availability Administration Guide

[11] The Adaptive Send Algorithm, MySQL Cluster API Developer's Guide, MySQL AB, http://mysql.bst.lt/doc/ndbapi/en/overview-adaptive-send.html

[12] High Performance Database Solutions For Multiplay Service Architectures, David Shepherd, Technical Whitepaper, IPL Ltd (UK), 2007.

[13] MySQL Cluster API Developer's Guide, MySQL AB, http://mysql.com/doc/ndbapi/en/overview-adaptive-send.html

[14] MySQL Cluster Manual, MySQL AB, http://dev.mysql.com/doc/refman/5.1/en/mysql- cluster.html

[15] MySQL Code Quality, MySQL AB, http://www.mysql.com/why-mysql/quality/

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN (9786) Web sun.com