White Paper

Steel-Belted Radius High Availability

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
408 745 2000 or 888 JUNIPER
www.juniper.net

Part Number: 200170 -001 Feb 2006

Executive Summary

Steel-Belted Radius High Availability (SBR HA) provides a stateless high-availability AAA platform for customers requiring scalability and very high reliability for RADIUS-based policy management software.

The SBR HA platform offloads certain aspects of SBR operation, such as IP address pool management and session tracking functions, from SBR servers to a distributed, highly fault-tolerant database cluster shared by all SBR servers. In this architecture, each database node has its own memory and permanent storage, while the actual database is replicated across multiple nodes within the cluster. This platform provides both vertical and horizontal scalability.

Introduction to SBR/High Availability

SBR HA is designed to provide 99.999% availability for the AAA server infrastructure using a distributed node architecture which has no single point of failure. The system consists of multiple nodes that can be distributed across machines and regions to ensure continuous availability in case of node or network failure.

Key Benefits

The key benefits of the SBR HA platform are:

- **99.999% Availability** – provided by a fault tolerant architecture with no single point of failure using a distributed node-based architecture with fast fail-over.
- **Horizontal and Vertical Scalability** – provided to incrementally scale the platform to meet current and future performance, reliability and throughput requirements. The SBR servers that interact with the database cluster may be organized into groups of servers.
- **Performance** – provided by the SBR HA platform database cluster that keeps all data in memory for transaction processing and fast fail-over purposes and limits I/O bottlenecks by asynchronously writing transactions logs to disk, thereby managing tens of thousands of distributed transactions per second.
- **Manageability** – provided by a unified management domain that presents the user with a complete virtual environment even though that unified virtual resource is actually constructed from a number of separate elements, which may reside in multiple data centers. This lowers costs of administration, service provisioning, hardware or software upgrades, and provides a pool of resources for maintenance activities.
- **Readily Integrated** – provided by support for an extensive list of NAS devices currently supported by SBR and a core competence in the RADIUS protocol, Juniper's SBR HA platform is a truly vendor-agnostic RADIUS product. Juniper Networks has deployed over 1,300 SBR servers to 250+ Operator and Service Provider customers and has extensive experience scaling and migrating RADIUS solutions.

AAA System Architecture

SBR HA allows certain aspects of SBR operation to be performed in a manner that maintains no state in the SBR server. Instead, relevant state information is offloaded to a database cluster that can be shared by a number of SBR servers.

The database cluster consists of multiple hosts (called cluster nodes). The cluster is configured for redundancy (no single cluster node failure will jeopardize the cluster) and can be horizontally and vertically scaled to achieve the necessary throughput.

Juniper Networks has extensive experience with capacity planning for SBR servers in operator networks. Hooking each SBR server to the database cluster does not materially change the performance capability of the SBR server: while it no longer needs to maintain its own current sessions table (savings), it must now communicate with the database cluster (cost).

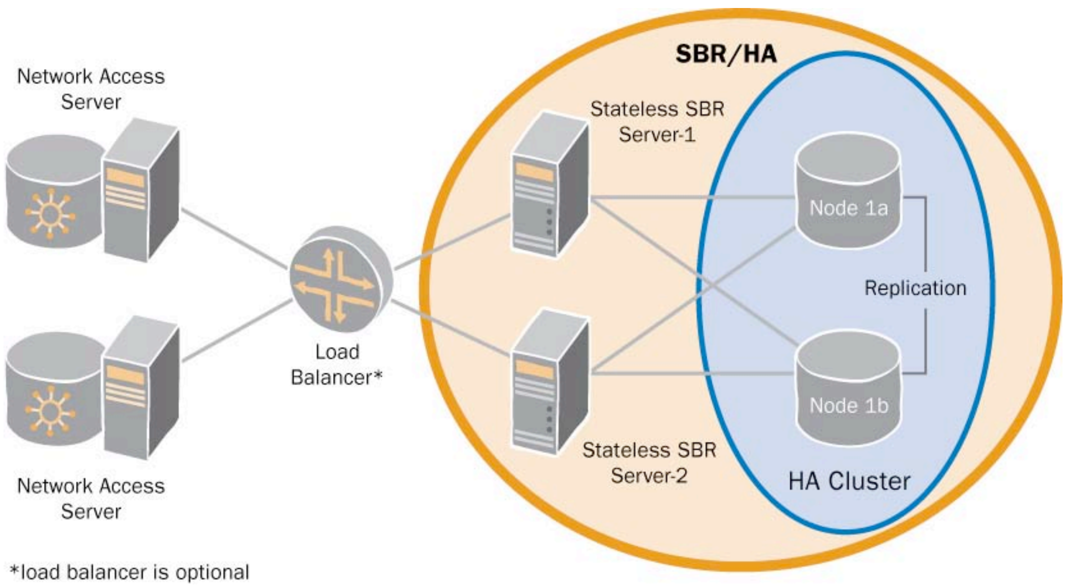

Figure 1: Architectural Overview of SBR HA Deployment[1]

Figure-1 illustrates a four-system SBR HA architecture: the minimal deployment of systems required to achieve fault tolerance. Data stored within the cluster is replicated across each node allowing each to contain identical data.

The database cluster uses a SQL-based design to store data, providing multiple tables for IP address pool information as well as a Current Sessions Table (CST), which acts as a dynamic sessions database of all active subscriber sessions across the network.

1 Network Access Server is used here as a generic reference: all devices which support IETF RFCs 2865 & 2866 as a RADIUS client are implied. This includes GGSN; PSDN; Home Agent; BRAS; VPN (LAC/LNS); Edge Router; Access Point; Access Controller; and other devices.

Cluster Schema

While the database cluster schema is extensible for customization, the default schema has been designed to address standard requirements for the majority of customers, both wireless and wire-line.

The database cluster uses a relational, SQL-based design, where different tables store distinct groupings of data. There are two primary groupings:

- IP address pools
- current sessions data

IP Address Pools

As part of sessions management, SBR HA stores all IP address pools centrally within the cluster. This information is distributed across all cluster nodes for fault tolerance and high availability. RADIUS authentication and accounting traffic may be sent to any stateless SBR server and IP addresses will be allocated dynamically to subscribers during successful authentication.

There are three tables in the cluster which store IP pools information:

- IP Pools Table
- IP Ranges Table
- IP Address Table

None of these tables require customization. They are created during installation, provisioned during SBR HA setup and designed to store their respective types information in memory and (optionally) on disk for fast retrieval during the authorization process.

Operations personnel will be able to query the IP Address Table to verify what addresses are in use and how many addresses remain available for any specific IP pool or pools. Scripting capabilities will allow for stale session cleanup.

Current Session Data

As a related part of sessions management, SBR HA stores session data centrally within the cluster in the Current Sessions Table (CST). This information is distributed across all cluster nodes for fault tolerance and high availability. Since requirements as to what information will need to be stored in this table may be customer specific, the CST is extensible.

As noted previously, the default CST schema is designed to address the majority of customer requirements. Drivers for the CST schema design are as follows.

Operations

- manual or scripted queries to verify information about sessions as part of customer care services
- manual or scripted queries to verify information about sessions as part of user tracking for security

Applications

Various operator services involve systems/applications on the IP network which need to correlate distinct pieces of information in order to function. Since SBR HA receives detailed attributes for each session it serves and maintains state for this information centrally, it is the logical source for other applications to query for correlative and verification purposes.

For example, A video gateway may know an identifier for a given subscriber (e.g. MSISDN, IMSI or username depending on the network), but it does not know the subscriber's current IP address. This gateway can query the SBR HA CST in real time to correlate these two pieces of information about an active session.

Figure-2 below illustrates the general schema design along with usage examples.

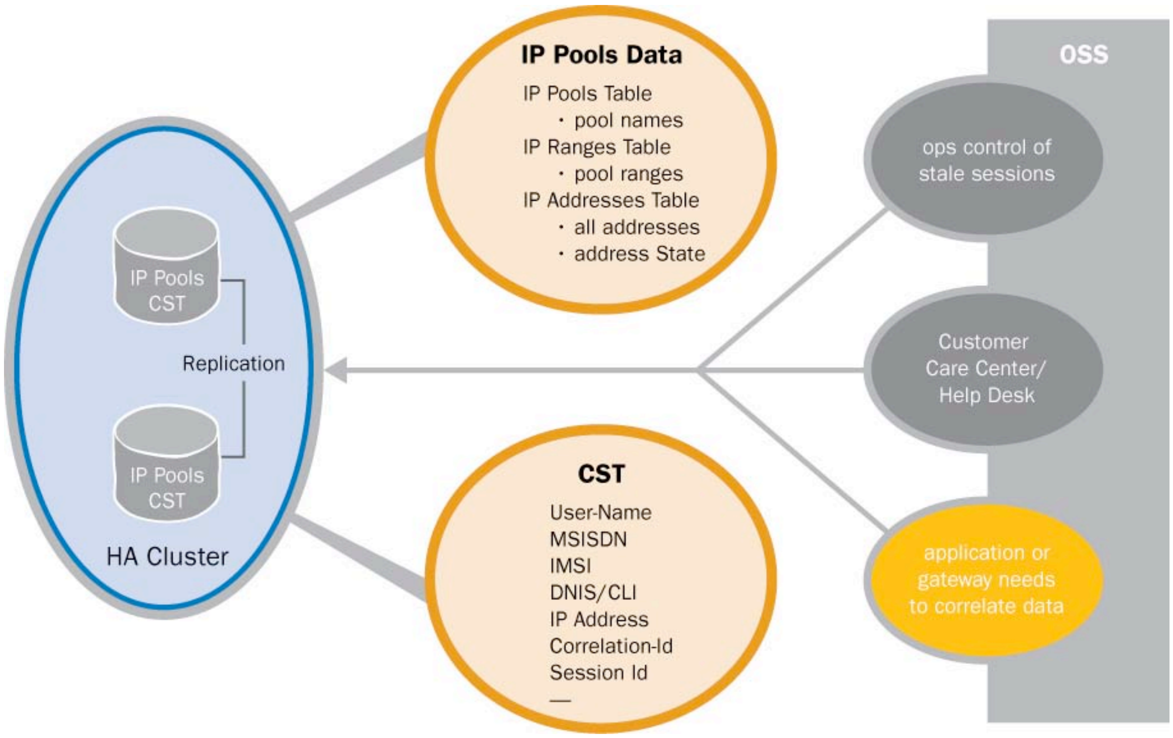

Figure-2: database schema and usage illustration

Scaling the AAA Architecture

The AAA system architecture described previously illustrates a four-system architecture comprised of two stateless SBR servers deployed with a database cluster of one node group with two nodes. The four-system deployment represents the minimum architecture required to achieve fault tolerance.

The SBR HA platform will scale to meet the increased capacity requirements associated with vertical and horizontal service growth. The architecture may be scaled horizontally by deploying additional node groups within the cluster and/or by deploying additional stateless SBR servers to front-end the cluster.

Scaling the Cluster

The HA database cluster consists of one or more node groups. Each node group maintains unique portions of the database. The number of node groups in a cluster determines the level of performance of the cluster. A cluster with four node groups stores one fourth of each table's rows on each node group. Node groups are akin to striped drives in a RAID volume.

Each node group consists of one or more nodes. The number of nodes per node group determines the level of redundancy of the cluster. A system deployed with two nodes per node group will survive the failure of a single node – one with three nodes per node group would survive the failure of any two nodes. Nodes in a node group are akin to mirrored drives in a RAID volume.

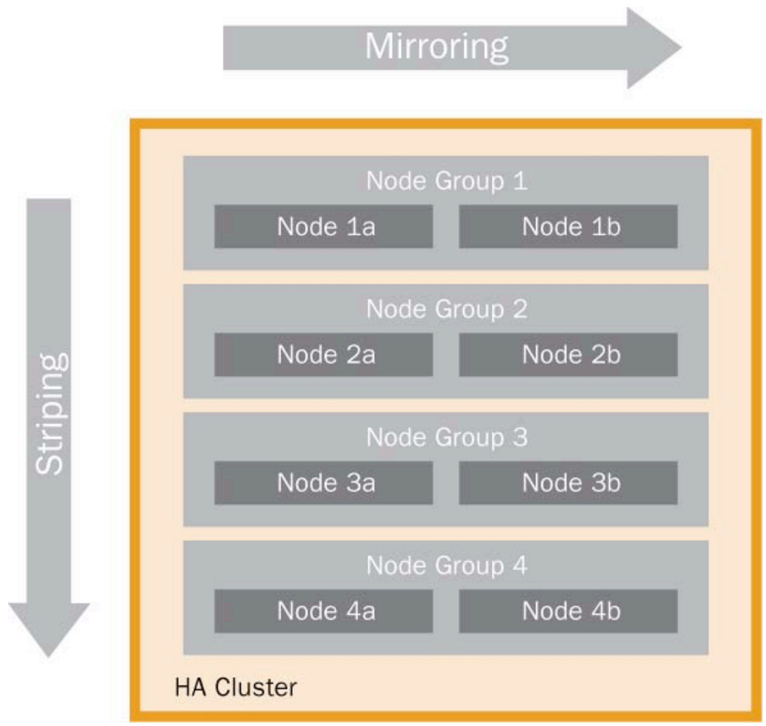

Figure 3: four node group/eight node cluster

Scaling SBR

The AAA architecture may also scale by deploying additional SBR servers to the network, while leveraging the same database cluster setup. Figure-4 illustrates two additional SBR servers deployed in Data Center 2.

It's critical to note that keeping the database cluster as a single node group with two nodes may require vertical scaling of the nodes in the cluster to accommodate for the increased traffic from the front-end SBR server infrastructure. See the Vertical Scaling section below for details.

Each grouping of SBR servers, located in different geographical sites as needed, represents a pool of servers.

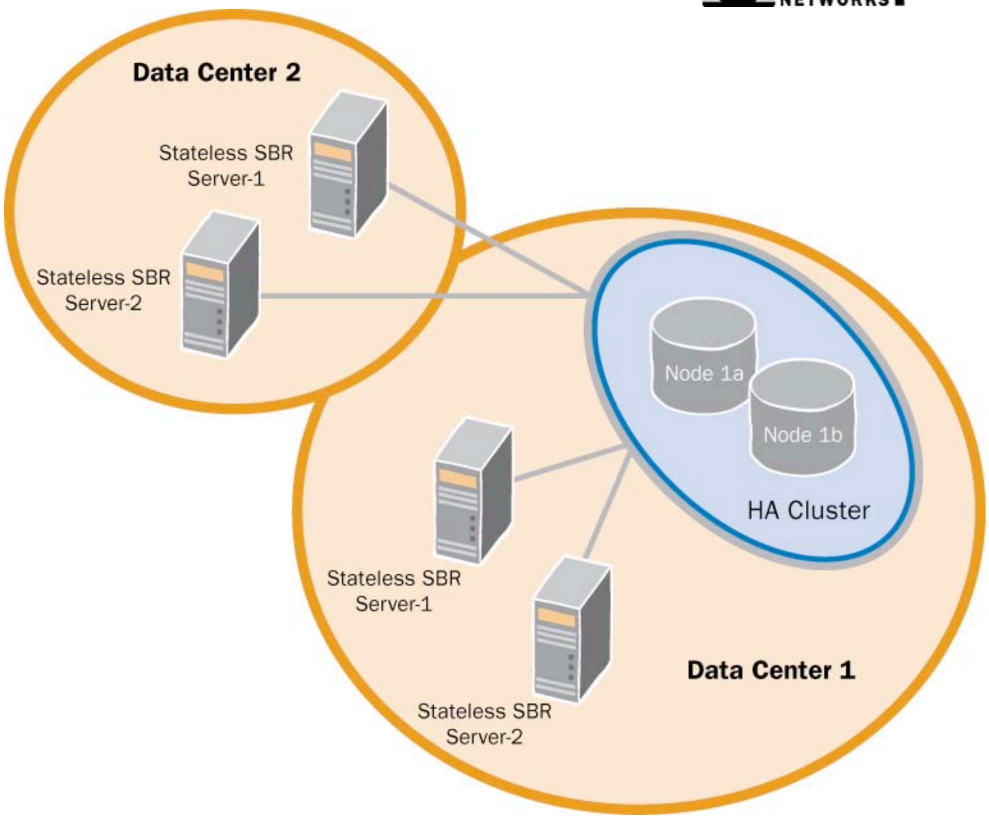

Figure 4: Scaling SBR

When additional capacity is needed, new SBR servers can be added to existing pools in an incremental fashion. The advantage of maintaining pools of stateless SBR servers becomes immediately apparent:

- The outage of any one of the SBR servers reduces the overall throughput a given pool is capable of achieving, but it does not affect the functionality that the pool of servers provides.
- A spare server can be kept on stand-by and can be added to whichever pool has experienced an outage – no migration of data is required to accomplish this.

Vertical Scaling

Any of systems which comprise a SBR HA deployment may be scaled vertically to provide increased capacity. All SBR HA components are designed to leverage system hardware resources to provide carrier-grade performance.

Vertical scaling may be accomplished in the following ways:

- increasing system memory
- increasing the number of processors
- increasing processor speed

The following section reviews database throughput.

Database Throughput

An individual cluster node's performance is limited by CPU (rather than I/O). The scaling is almost linear based on CPU speed – a 1.5 GHz processor will allow a node to process 50% more than a 1GHz processor. While the database cluster processor is, for all intents and purposes, single threaded, it does use a separate thread to write database log files to disk. For this reason, it will always process dual processor hosts as cluster nodes: one processor will do the entire database processing, while the other one will allow the OS to run and the I/O to be completed.

As previously noted, rows in each table are evenly spread out over the node groups that make up the cluster. The unique primary key of any table can be used by a client application of the cluster (e.g. SBR) to determine which node group in the cluster is storing a particular record. In other words, looking up a record by its primary key only involves a single node in the cluster. Scans of tables for records meeting certain criteria (i.e. searching for sessions which are more than 2 hours old) does involve all the nodes in the cluster.

The SBR HA database schema is specifically designed to only use primary key lookups during transaction processing. Even operations such as phantom session sweeps, which would seem to require a scan of the current session table, have been implemented in a manner that removes the necessity for a scan. As a result, the SBR database cluster performance scales almost linearly based on the number of node groups in the cluster.

Database Persistence

The cluster node will replicate any data written to it to all other nodes in its node group. In the event of an outage of a single node, the other node(s) in the node group take over responsibility for requests that would have gone to the node that failed. When the failing node comes back online, it obtains up-to-date data it needs from the other nodes in the node group.

The database cluster can be configured to operate with or without data persistence. If configured to operate without data persistence, the cluster node will replicate data to other nodes in the group, but it will not write the data to disk. In this scenario, should the cluster stop operating (either due it being shut down or due to all the nodes in a single node group going offline), all data will be lost and the cluster will be have a blank database when the nodes comes back online. If the data being stored is of an ephemeral nature, this may be a desirable side effect – if it is not, persistence should be enabled.

When persistence is enabled, each cluster node should have twelve times the disk space of its database storage capacity enabled. In other words, a 16GB cluster node has 12GB of database storage capacity and should have 12 x 16GB = 192GB of disk space available for storing the log files related to the database.

Note that disabling persistence only increases the throughput of the cluster by about five percentage points. This is due to the fact that the cluster node is primarily CPU bound and all I/O processing happens on a separate processor.

Hardware Recommendations

The following are the hardware recommendations for the HA cluster nodes.

Sun Netra 440 servers[2], each of which is configured as follows:

Dual 1.6GHz UltraSPARC IIIi processors

8GB RAM

2 x Gigabit Ethernet

Solaris 9

Hard drive size will depend on capacity requirements

Cluster Performance

Juniper Networks has verified performance benchmarks for the following three database cluster sizes. A single session is comprised of three database transactions:

- database transaction from authentication + phantom
- database transaction from accounting Start
- database transaction from accounting Stop (delete session)

Performance benchmarks:

- Two dual 1.6 GHz UltraSPARC III running 64-bit Solaris configured as one node group with two nodes in the group: 3,780 sessions per second
- Four dual 1.6 GHz UltraSPARC III running 64-bit Solaris configured as two node groups with two nodes per group: 7,420 sessions per second
- Six dual 1.6 GHz UltraSPARC III running 64-bit Solaris configured as three node groups with two nodes per group: 10,920 sessions per second

The outage of a single node reduced throughput by the following percentages:

- In the two node scenario: by 40%
- In the four node scenario: by 20%
- In the six node scenario: by 13%

2 Netra 240 servers may be used however they are limited to 8GB of RAM and would need to be replaced should capacity requirements warrant additional memory.

Professional Services

Juniper Networks provides a full suite of professional services to meet the needs of its customers including project planning, configuration planning, installation, customization and deployment of the SBR HA solution.

Conclusion

Juniper Networks' SBR HA platform is a fault-tolerant, highly available and high performance solution for any operator requiring scalability and very high reliability for RADIUS-based policy management software.

Flexibility in sizing and scaling allow SBR HA to provide AAA services for any size network and Juniper Networks' extensive experience, strong technical support and professional services complete this solution.

For more information, please visit www.juniper.net