ERICSSON

CUDB High Availability

FACILITY DESCRIPTION

7/155 34-HDA 104 03/10-V1 Uen A

Contents

1 Introduction

This document provides a description for the High Availability (HA) feature of the Ericsson Centralized User Database (CUDB).

1.1 Scope

The purpose of this document is to describe the structure, configuration, and mechanisms that the CUDB system uses to maintain HA in the system. The HA feature is responsible for system protection and the quick automatic recovery from interruptions and failures. Key features of HA include redundancy, replication, reconciliation and the handling of split situations.

The document consists of two major parts. First, a short introduction of the CUDB infrastructure and logic architecture is provided, which is followed by a detailed description of the mechanisms and policies that HA offers.

1.2 Revision Information

Rev. A

This document is based on 7/155 34-HDA 104 03/9 with the following changes:

- Removed obsolete information.
- Virtualization terminology updates throughout the document.
- Structural rearrangements throughout the document.
- Section 2.1 on page 5: Added note regarding the Advanced Network Protection Value Package.
- Section 3.1.1 on page 7 and Section 3.1.4 on page 12: Updated the LDAP Counter Process information in Figure 1, Figure 4, and in the relevant list item. Added Traffic Control to SC blades in Figure 1 and Figure 4, and updated description with information on Traffic Control.
- Section 3.1.4 on page 12: Updated the Notification Process and Figure 5. Added the Notification Process and updated Figure 6.
- Section 3.1.7.3 on page 31: Updated information regarding Storage Performance Monitoring Function.
- Section 3.3.1 on page 41: Updated description of Data Distribution Supported in PLDB.
- Section 3.3.2.1 on page 43: Updated section with additional information on PLDB mastership movement.
- Section 3.3.3.3 on page 46: Updated description.
- Section 3.3.4 on page 47: Updated scenarios for cases when Slave PLDB or Master PLDB is down.
- Section 3.3.5.2 on page 66: Updated node recovery information related to symmetrical split situations.
- Section 3.3.6 on page 74: Added a new bullet to explain when master election is needed.
- Section 4.1 on page 75: Updated description.

1.3 Target Groups

This document is intended for system administrators and users working with the HA feature.

1.4 Prerequisites

Users of this document must have knowledge and experience of the following:

- CUDB system and data architecture.
- Infrastructure and SW components of the CUDB node.

1.5 Typographic Conventions

Typographic conventions can be found in the following document:

- *Typographic Conventions*

2 Overview of High Availability

High availability is a system property defined as the probability of providing service upon request. A system has high availability if its probability of availability is beyond 99.999%, or if the system unavailability does not exceed 5.26 minutes per year.

CUDB is a distributed database system which provides data and service access in high availability. The system is designed to ensure HA on several levels, aiming that in case of failure in any of its hardware or software components, the overall CUDB system (or the data stored in it) is not compromised, and the database can continue providing as much service and data consistency as possible.

2.1 Architecture

From a top level perspective, the HA feature of CUDB is built upon two basic features:

- **Single Point of Access**

  CUDB is a distributed database system made up of connected CUDB nodes. Each node provides a Single Point of Access to any data stored across the distributed system. Therefore, even in case a node or multiple CUDB nodes experience complete system failure, application clients can still access CUDB through any available CUDB nodes to retrieve and modify data stored in the system.

- **Geographical Data Redundancy**

  Each piece of data stored in CUDB is replicated at least in two CUDB nodes located in different network site locations (ensuring 1+1 data redundancy, also known as double geographical redundancy). Also, if needed, data can even be replicated in three CUDB nodes located in three different remote network sites (ensuring 1+1+1 data redundancy, also known as triple geographical redundancy).

  Note:

  - The Triple Geographical Redundancy feature can only be used if the Advanced Network Protection Value Package is available.
  - Standalone configuration without geographical redundancy is supported for Customer Trial, Customer Test, and Ericsson Internal systems.

Besides these features, the whole system and each of its internal components have been designed to withstand single failures on each level. To better

understand the different built-in resilience and redundancy mechanisms, these levels are defined as follows:

- **HA at CUDB Node Level**

  Each CUDB node provides a set of HA node functions, performed by a set of processes running on a highly available platform. These processes are running on a set of redundant infrastructure components to ensure the high availability of the CUDB nodes.

  However, a series of multiple failures could still result in a severe or complete failure at CUDB node level. In that case, additional HA mechanisms are triggered on the CUDB system level to maintain the rendering service, even if one or several CUDB nodes are down, or taken out of the system.

- **HA at CUDB System Level**

  The CUDB System ensures data availability and data access service in case of multiple failures at the CUDB node level, resulting in a complete failure in the CUDB node or in any DS Unit contained in any of the nodes.

  In case of DS Unit or PLDB failures, the service availability of each data partition is ensured by means of the CUDB geographical redundancy. At system level, supervision mechanisms exist that monitor the service availability of every database replica and CUDB node. This supervision functionality is known as Data Availability Coordination (DAC), and allows dynamic system reconfiguration to avoid failures at CUDB system level.

For a detailed description of the node and system level features, see Section 3 on page 7. For further information on the CUDB system and data architecture, refer to *CUDB Technical Product Description*, Reference [1].

3 Detailed Description of High Availability

This section contains detailed information about the HA feature. HA of the CUDB system is realized on two levels: node level, and system level. For more information on node level availability, see Section 3.1 on page 7. For more information on system level availability, see Section 3.3 on page 41.

3.1 Node Level Availability

This section provides information on the functions and processes providing HA on the node level.

3.1.1 Infrastructure Availability for CUDB Systems Deployed on Native BSP 8100

The hardware realization of the CUDB nodes is based on the Blade Server Platform (BSP). A CUDB node is a fully equipped hardware cabinet hosting 1-3 Evolved Generic Ericsson Magazines (EGEM2), populated by Generic Ericsson Processor (GEP) blade servers and other infrastructure boards. For further details on the CUDB hardware, refer to ‘‘BSP Hardware Description’’ documentation in the BSP 8100 CPI.

Two infrastructure boards (CMX boards) are located in the first magazine or subrack, providing external routing connectivity to the site switches/routers. The rest of the magazines or subracks also contain two infrastructure CMX boards, and all CMX boards in all subracks provide internal Layer 2 (L2) backplane switching connectivity to the blade servers in the subrack.

Additionally, two support boards (SCX boards) are located in each magazine or subrack providing the execution environment for the switching and routing management functions.

The CUDB node topology consists of a double star configuration on CMX boards. The CMX board pair that constitute the double star is located on the main or first subrack. Every CMX board on additional subracks in the cabinet is connected to one CMX board of the main subrack of that cabinet through one 10GE port on the front, in order to extend the switching backplane to the upper subracks.

CUDB routers balance incoming traffic over the GEP blade servers hosting Virtual IP Address (VIP) Front End (FE) instances. The VIP FEs are placed in the first (main) subrack but if the cabinet holds 20 or more GEP blade servers, additional VIP FEs are placed in the second subrack.

Figure 1 shows the CUDB internal topology of a node with blades installed in three subracks.

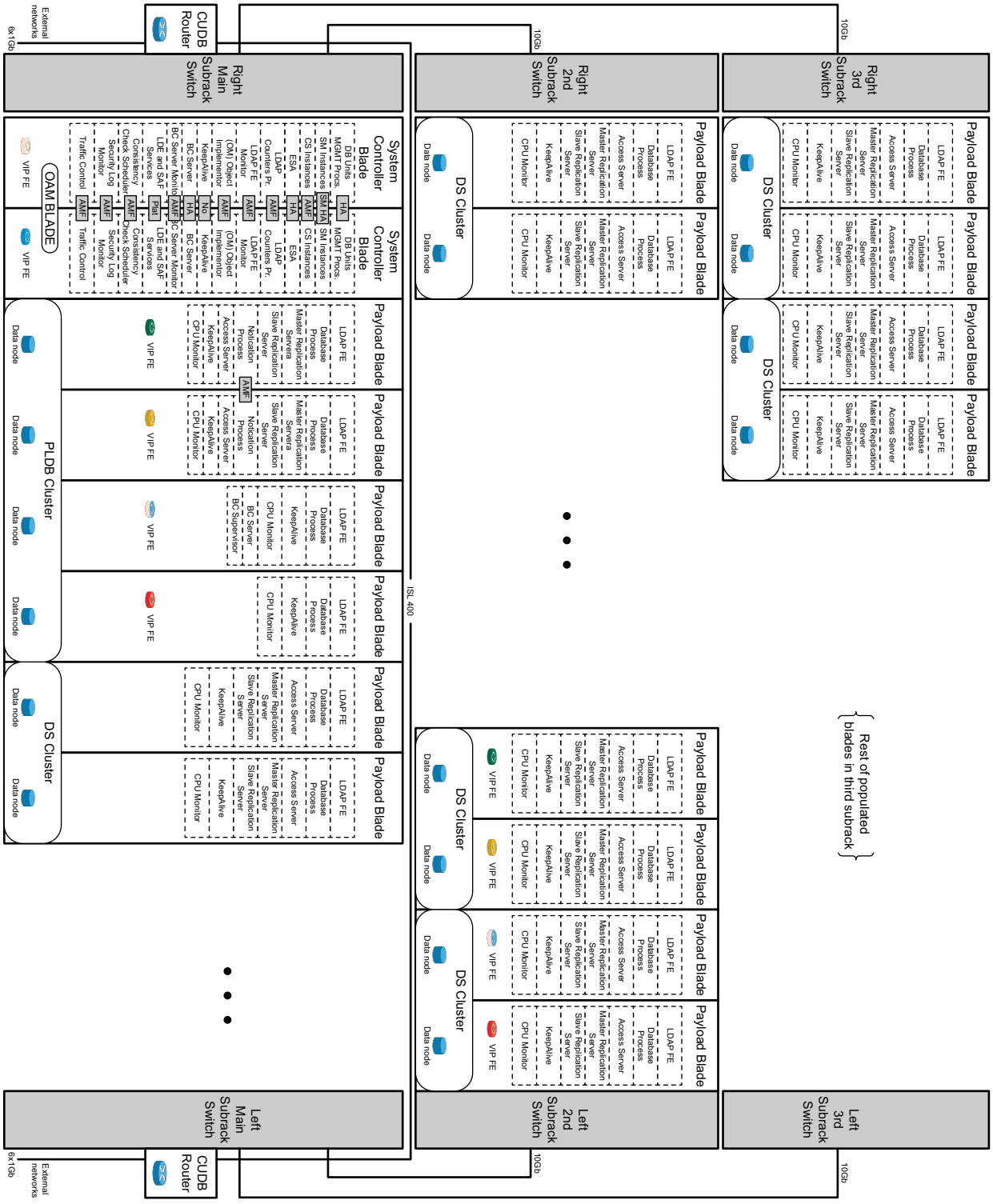


Figure 1 CUDB Node Topology

Figure 1 also provides a generic overview of the relationship between the blade processors and the running processes they have. Even though the position of a blade in a slot does not generally imply its role within the system, some specific rules are available regarding the physical position of a blade and its functionality established during installation. These rules are as follows:

- CUDB router devices are always located in the main subrack.
- CUDB System Controller (SC) blades are placed in the first two slots of the main subrack.
- PLDB blades are located after the SC blades in the main subrack.
- Blades assigned to the same DS cluster unit are placed in consecutive slots.
- Blades running Virtual IP FEs are located on the first subrack but cabinets holding 20 GEP5 blades or more, require increased Virtual IP processing capacity, so additional Virtual IP FEs are located on the second subrack as well.

The operating system of the blades in the CUDB node is the Linux Distribution Extension (LDE) distribution, allowing to see the CUDB node as a Linux blade cluster. From the LDE perspective, two types of blades are available: SCs and Payload Nodes. The SC blades centralize the management functions in the Linux cluster, and two of them are available for redundancy purposes.

The following sections describe HW resiliency in more detail.

3.1.2 Infrastructure Availability for CUDB Systems Deployed on Cloud Environment

In case the CUDB system is deployed on a cloud environment, the infrastructure availability is provided by the cloud infrastructure. Refer to the cloud infrastructure documentation for more information.

In general, the deployment of CUDB Virtual Machines (VMs) across compute hosts is not predefined and does not generally imply its role within the system. To ensure the high availability of the CUDB system, failure domains are used.

“Failure domain”, in this context, is a group of VMs impacted when an underlying infrastructure experiences problems. System Controllers (SCs) and payload VMs are deployed over different failure domains. When deploying a CUDB node in a cloud infrastructure, the following rules are applied:

- No two payload VMs of the same DS Unit are deployed in the same failure domain.
- SCs are not deployed in the same failure domain.
- Only one BC server is deployed in one failure domain.

This deployment ensures that in case of an infrastructure failure impacting one failure domain, the system still remains fully functional.

3.1.3 Network Links Resiliency

In case the CUDB system is deployed on cloud infrastructure, VM network links resiliency is provided by the cloud infrastructure. Refer to the cloud infrastructure documentation for more information.

In case the CUDB system is deployed on native BSP 8100 hardware, every blade server in a subrack is connected to the infrastructure boards in that subrack. CMX boards are connected through a pair of 10GE links on the backplane (one link against each CMX board).

Additionally every blade server in a subrack is connected to the support SCX boards through a pair of 1GE links on the backplane (one link against each SCX board) during the PXE Boot phase. Once this phase is finished, the 1GE links are deactivated and cannot be accessed.

Blade network links use Active/StandbyL2 resiliency mechanism based on the Linux Bonding Driver. Therefore, no link aggregation exists at the blade network interfaces. In steady state, all blades have the active links to the right LAN side of the double star switching topology.

The interface bonding switchover is based upon link supervision using Address Resolution Protocol (ARP). All blades are sending periodical ARP requests to two configured and different ARP targets. The targets (usually the infrastructure boards in the main subrack) are configured during LDE installation.

The bonding switchover condition is triggered when there is no response to a configured number of ARP packets from both ARP targets. Typically, the bonding active link to the standby interface in the bonding configuration is forced to change after three missed replies from both infrastructure boards. The time delay between retries is 100 ms.

After the recovery of the active link is detected again by ARP monitoring, the bonding drivers enable the active link towards the left LAN side again.

Figure 2 shows how server bonding failover takes place in case of a Single Blade configuration.

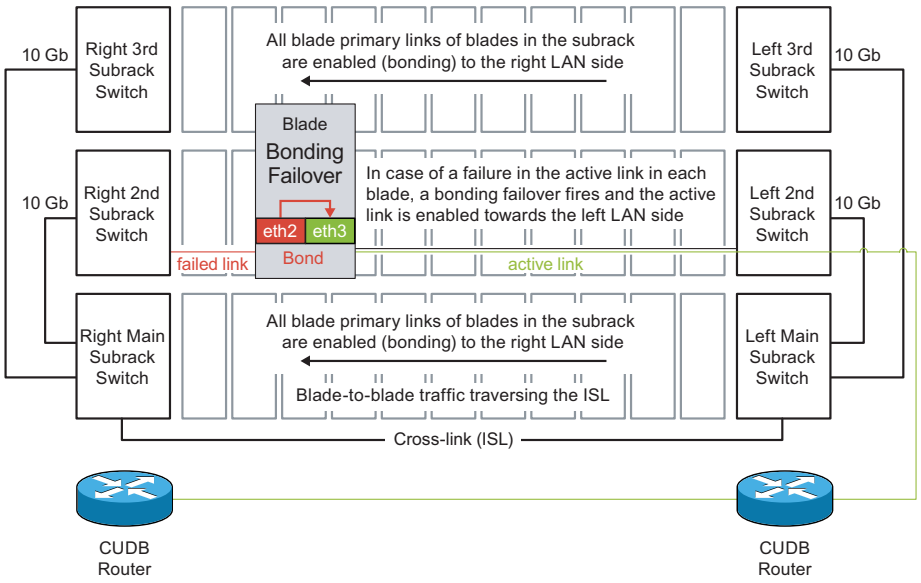


Figure 2 Single Blade Server Bonding Failover

In steady state, if all blade servers have active links in the primary interface in bonding configuration, no traffic flow is present over the main subrack cross-link (for example, no traffic is present through the inter-switch link, or ISL for short). However, in case of single blade servers, network link failures or a failure in the switch of the active LAN side causes blade-to-blade traffic to traverse on the cross-link.

In case of failure in an infrastructure board in the right or active LAN side, all blade servers bonded interfaces on that subrack fail over to the standby links that are connected to the left LAN side. The reason of this is that no ARP supervision is available to the ARP targets through the link connected to the left LAN side.

Figure 3 shows how bonding failover takes place in case of an infrastructure board failure in the main subrack.

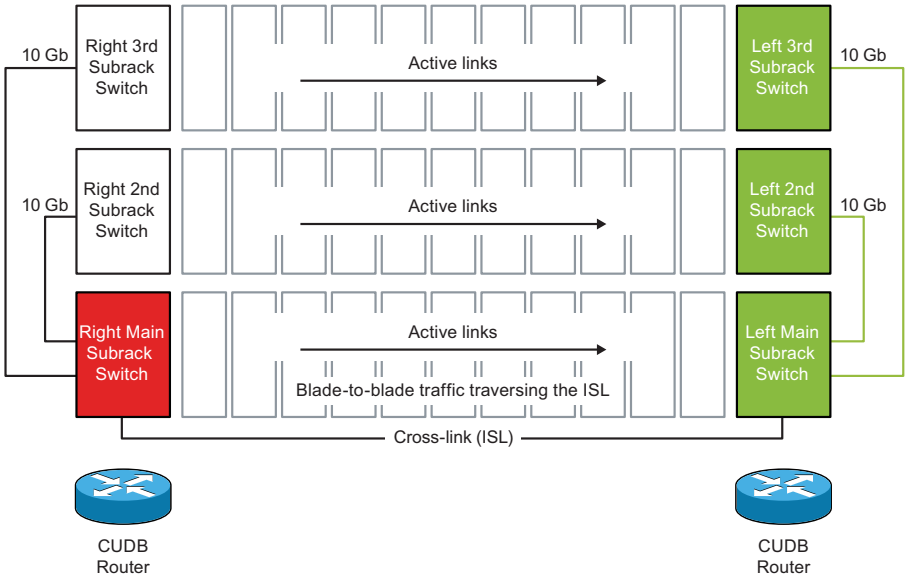


Figure 3 Main Subrack SCX Failure and Bonding Failover

3.1.4 Server Resiliency

All CUDB node functions withstand the failure of any single blade or VM in the CUDB node. However, multiple failures in two or more different blade or VM servers within a CUDB node are withstood only if the faulty ones do not share any redundant functions. For example, simultaneous failures in a single SC and a single PLDB (or DS blades not belonging to the same cluster unit) do not necessarily bring the CUDB node or a cluster unit down, as long as their redundant pairs are still running.

However, a simultaneous failure in two blades or VMs hosting a redundant service (such as two SCs, or two DS/PLDB payloads of the same DS/PLDB instance) can result in a complete or partial CUDB node failure. In this case, the proper HA mechanisms are triggered at the CUDB system level to restore the service in a different CUDB node.

The supported single blade or VM failures are as follows:

SC Failure

SCs host mainly LDE and Core MiddleWare (MW) services along with the CUDB OAM processes. Figure 4 shows the main CUDB services and processes running in the SCs, along with their associated redundancy modes.

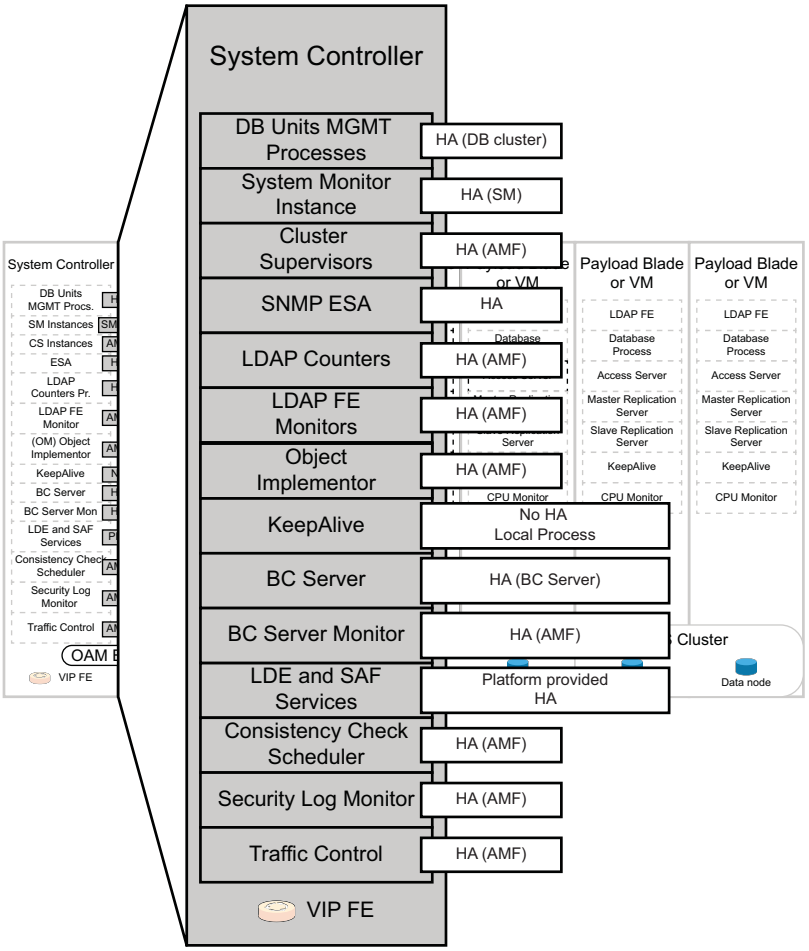


Figure 4 Main CUDB Services, Processes and Associated Redundancy Modes of the SCs

The services, processes, and redundancy modes of the SCs are as follows:

- **Database Units Management Processes**

  Each DSG/PLDB cluster in the node runs one instance of Database Management processes. See Section 3.1.7.2 on page 26 for more information on the architecture of cluster units.

- **System Monitor (SM) Instance**

  SM instances run on SCs in active/active redundancy mode, using their own HA mechanisms not integrated to the Availability Management Framework (AMF) service.

- **Cluster Supervisors**

  Each SC runs one instance of cluster supervisor per DSG/PLDB cluster in the CUDB node.

- **Ericsson SNMP Agent (ESA)**

  Each SC runs one ESA process configured in active/active redundancy model using built-in HA mechanism.

- **LDAP Front End (FE) Monitors**

  The Lightweight Directory Access Protocol (LDAP) FE Monitors are integrated to the AMF HA service, and work in active/standby mode.

- **LDAP Counter Process**

  The LDAP Counter process runs in one-active mode integrated with the AMF HA service.

- **Object Implementor (OI)**

  The OI works in active/standby mode integrated with the AMF HA service.

- **KeepAlive**

  This process monitors the running services and processes in the local blade or VM. Therefore, it is not configured to use redundancy in other blades or VMs. The processes are monitored by the cron daemon which restarts them in case they are not running.

- **Blackboard Coordinator (BC) Server Instance**

  The BC server instances run in the SCs, but depending on the deployment, it may also be executed on payloads as well. The BC server instances running on the CUDB nodes of a CUDB site constitute a BC cluster with its own mechanism of high availability for its cluster service.

  Distribution of BC servers among different blades or VMs of the CUDB Nodes of a given site follows the pattern explained in Table 1.

  For more information about BC clustering solution, see Section 3.2 on page 34.

Table 1 BC Servers Deployment Options for a BC Cluster in a Site

Nodes in the site	Node 1	Node 2	Node 3	Node 4	Node 5
1	SC1 SC2 PL_2_5				
2	SC1 SC2 PL_2_5	SC1 SC2			

Nodes in the site	Node 1	Node 2	Node 3	Node 4	Node 5
3	SC1 SC2	SC1 SC2	SC1		
4	SC1 SC2	SC1	SC1	SC1	
>4	SC1	SC1	SC1	SC1	SC1

- **BC Server Monitor**

  This process is in charge of supervising the BC server instances running in the node. The process has two instances that work in active/standby redundancy mode, relying on the process HA framework provided by the Core MW platform through the AMF service. The active process monitors the BC servers and is responsible for raising and clearing alarms related to the BC server processes.

- **VIP FE instance**

  See Section 3.1.6.4 on page 23 for more details on VIP resiliency mechanisms.

- **Consistency Check Scheduler**

  Each SC runs one instance of CUDB Consistency Check Scheduler that controls execution of CUDB Consistency Check tasks on the CUDB node. The instances work in active/standby redundancy mode relying on the process HA framework provided by the Core MW platform through the AMF service.

- **Security Log Monitor**

  The Security Log Monitor is in charge of monitoring the status and configuration of the **Centralized Security Event Logging** function. Two instances exist, each on a separate SC (SC_2_1 and SC_2_2). One instance assumes an active role, while the other takes on the standby role, all under the control of the AMF service. Unlike other active/standby common task distributions, the standby instance of the Security Log Monitor is not only sleeping until it becomes active, but it also has the responsibility of deleting the configuration of the `syslog` configuration file.

- **Traffic Control**

  The Traffic Control is in charge of monitoring the configuration of traffic blocking rules in the configuration data model and provisioning VIP address activation and deactivation requests towards the eVIP Dynamic Traffic Management (DTM) server. Two instances exist, each on a separate SC (SC_2_1 and SC_2_2). One instance assumes an active role, while the other takes on the standby role, all under the control of the AMF service.

Simultaneous failures in the SCs result in CUDB node downtime, triggering the subsequent HA actions and mechanisms at system level. See CUDB Node is down in Section 3.3.4 on page 47 for more information.

PLDB Blade or VM Failure

The PLDB cluster size (that is, the number of blades or VMs) varies depending on the dimensioning of the CUDB system deployed. The minimum PLDB size consists of two blades or VMs

The PLDB payloads run different processes and services than the SCs. See Figure 5 and the list below for information on these processes, services, and their associated redundancy modes.

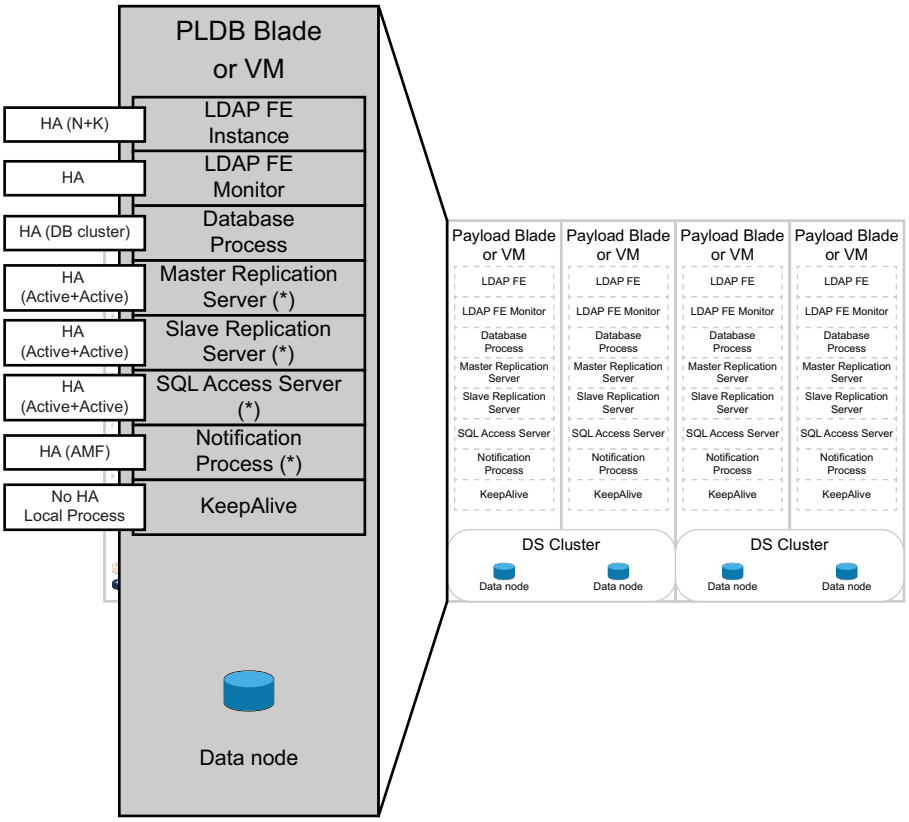


Figure 5 Main CUDB Services, Processes and Associated Redundancy Modes of the PLDB Blades or VMs

The services, processes and redundancy modes of the PLDB blades or VMs are as follows:

- **LDAP FE instance**

  The redundancy of LDAP FE follows an `n+k` pattern not just on the PLDB payloads, but also on the CUDB node level. See Section 3.1.7.1 on page 25 for more information on the redundancy model for LDAP FEs.

- **LDAP Front End (FE) Monitors**

  The LDAP FE Monitors run independently on each PLDB payload and (in contrast to the LDAP FE Monitor process on SCs that handles alarms) are responsible for the continuous operation of the LDAP FE running on the same payload. The high availability of the service is provided by KeepAlive.

- **Database Process**

  See Section 3.1.7.2 on page 26 for more details on the architecture of cluster units.

- **PLDB (Master) Replication Server**

  The first two PLDB payloads house two master replication servers, working in active/active mode.

- **PLDB (Slave) Replication Server**

  The first two PLDB payloads also house two slave replication servers, working in active/active mode.

- **PLDB SQL Access Server**

  The first two PLDB payloads also house two PLDB data access servers, working in active/active mode.

- **Notification Process**

  The notification process runs on all the payload blades or VMs of the node, but it works only on two of them in active/standby mode.

- **KeepAlive**

  This process is monitoring the running services and processes in the local blade or VM. Therefore, it is not configured to use redundancy in other ones. The processes are monitored by the `cron` daemon which restarts them in case they are not running on the blade or VM.

Simultaneous failures in the PLDB payloads can result in the complete failure of the PLDB cluster. If the PLDB cluster is out of service, the entire CUDB node is declared unavailable to process traffic, triggering subsequent HA actions and mechanisms at system level. See CUDB node is down in Section 3.3.4 on page 47.

DS Blade or VM Failure

The processes and services running on the DS payloads are similar to the ones running on the PLDB payloads (although minor differences exist between the two). See Figure 6 for the processes and services of the DS.

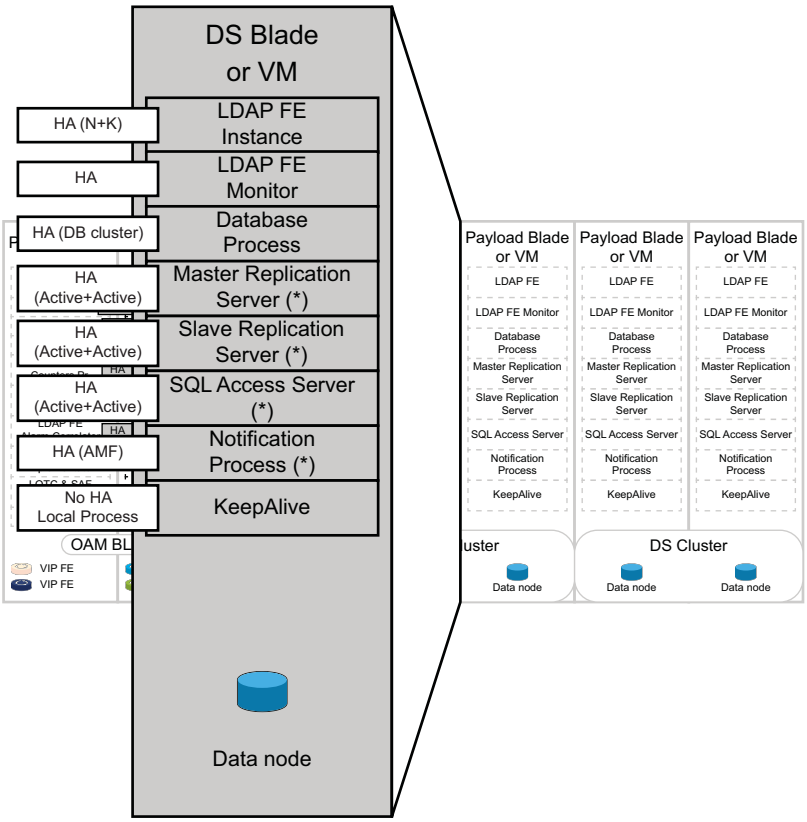


Figure 6 Main CUDB Services, Processes and Associated Redundancy Modes of the DS Blades or VMs

The services, processes and redundancy modes of the DS blades or VMs are as follows:

- **LDAP FE instance**

  The redundancy of LDAP FEs is defined on the CUDB node level, not at the DS level. See Section 3.1.7.1 on page 25 for more information.

- **LDAP Front End (FE) Monitors**

  The LDAP FE Monitors run independently on each DS payload and (in contrast to the LDAP FE Monitor function on SCs that handle alarms) are responsible for the continuous operation of LDAP FE running on the same payload. High availability of the service on each DS is provided by `KeepAlive`.

- **Database Process**

See Section 3.1.7.2 on page 26 for more information on the architecture of cluster units.

- **DS (Master) Replication Server**

  Two master replication servers are running on both DS payloads, working in active/active mode.

- **DS (Slave) Replication Server**

  Two slave replication servers are running on both DS payloads, working in active/active mode.

- **DS SQL Access Server**

  Two SQL data access servers are running on both DS payloads, working in active/active mode.

- **Notification Process**

  The notification process runs on all the payload blades or VMs of the node, but it works only on two of them in active/standby mode.

- **KeepAlive**

  This process is monitoring the running services and processes in the local blade or VM. Therefore, it is not configured to use redundancy in other ones. The processes are monitored by the `cron` daemon which restarts them in case they are not running.

Simultaneous failures in the DS blades or VMs belonging to the same DS Unit can result in failures of the geographical data redundancy feature operating at system level. See Section 3.3.3 on page 43 for more information.

3.1.5 Node Networking Resilience

In case the CUDB system is deployed on cloud infrastructure, node network resilience is provided by the cloud infrastructure. Refer to the cloud infrastructure documentation for more information.

CUDB nodes are deployed in network sites, and are therefore connected to the network site infrastructure. The CUDB infrastructure boards in the main subrack are the only CUDB hardware components connected to the external site switches, or directly to the site routers (the CMX infrastructure boards are electrically connected to a built-in optical/electrical converter located under the main subrack which connects with the site switches/routers using optical wiring).

The built-in optical/electrical converter, named APP, is duplicated and contains internal redundancy mechanisms that guarantee high availability of the optical/electrical conversion function.

For network ports/links and L2 redundancy, the CUDB router devices have interface ports configured for all the external VLANs using Link Aggregation (for example interface trunking, Link Aggregation Control Protocol, or LACP for short). Refer to *CUDB Node Network Description*, Reference [2] for further details on network configuration.

For redundancy at L3 level, the CUDB router pairs can be configured to use Virtual Router Redundancy Protocol (VRRP) or Bidirectional Forwarding Detection (BFD). These redundancy configurations are described below in more detail.

VRRP-based Router Redundancy

CUDB node router devices are deployed in pairs, and are configured by default to use VRRP as HA mechanism.

VRRP works with the paradigm of a Virtual IP floating on the physical interfaces of the redundant pair of routers. The floating IP address is set to the router which is configured as the master router in the VRRP cluster. If the master router fails, the backup router takes over the virtual IP address.

VRRP uses ARP-based monitoring. The CUDB infrastructure boards in the main subrack have a VRRP address configured for all the external networks to which the CUDB nodes are connected. Supervision between the router pair is done both through the external and internal sides (site switches and CUDB infrastructure boards in the main subrack, respectively).

A router failure triggers the other unit to take over the virtual routers, and then distribute the traffic to the VIP FEs.

Figure 7 shows the VRRP-based router redundancy logic.

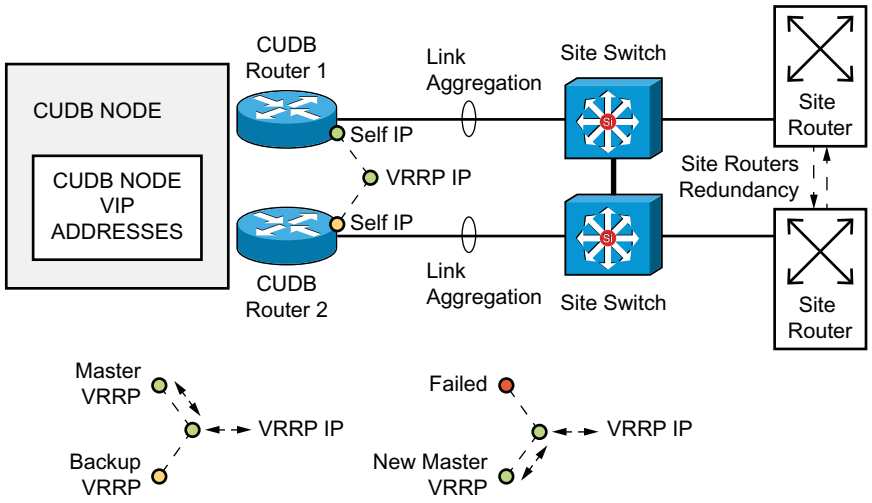


Figure 7 CUDB Node VRRP routers redundancy

BFD-based Router Redundancy

As an alternative to VRRP, CUDB routers can also be configured to use BFD protocol for fastrack failure detection between the site routers (L3 endpoints terminating BFD sessions) and the CUDB routers. BFD is a hello protocol that allows to test bidirectional communication by using packets of very small size (24 bytes on top of UDP+IP headers). Due to this low overhead, BFD timers are commonly in milliseconds range, allowing rapid failure detection between adjacent routers. BFD is focused exclusively on path-failure detection (that is, no other existing hello protocol duty mechanisms are used). In that sense, BFD is solely designed as an agent for other applications/clients (such as routing protocols) requiring fast-failure detection.

In the CUDB router, there are two possible BFD configuration alternatives:

- Single BFD Session
- Multiple BFD Session

In case of Single BFD, each router establishes BFD sessions towards one site router and tracks the availability of that path. The client provided with the tracking information from the BFD agents is the router forwarding engine (statics routes table), which enables a secondary IP route over the other CUDB router in case a failure is detected in any of the BFD monitored links towards the site routers. See Figure 8 for Single BFD-based solution.

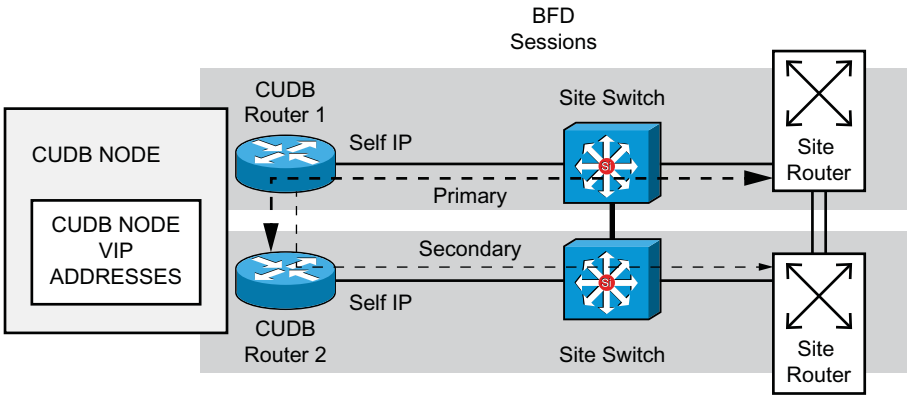


Figure 8 Single BFD-based L3 Redundancy

In case of Multiple BFD, from every CUDB router there is a static route with high priority towards each Site router tracked by BFD, and there is a secondary route with lower priority also monitored by BFD towards the diagonal router. In case a router loses the high priority BFD session toward the Site Router, the lower priority BFD session toward the cross site router is taking over the role. The client provided with the tracking information from the BFD agents is the router forwarding engine (statics routes table), which enables a secondary IP route over the other CUDB router in case a failure is detected in all of the BFD monitored links towards the site routers. See Figure 9 for Multiple BFD-based solution.

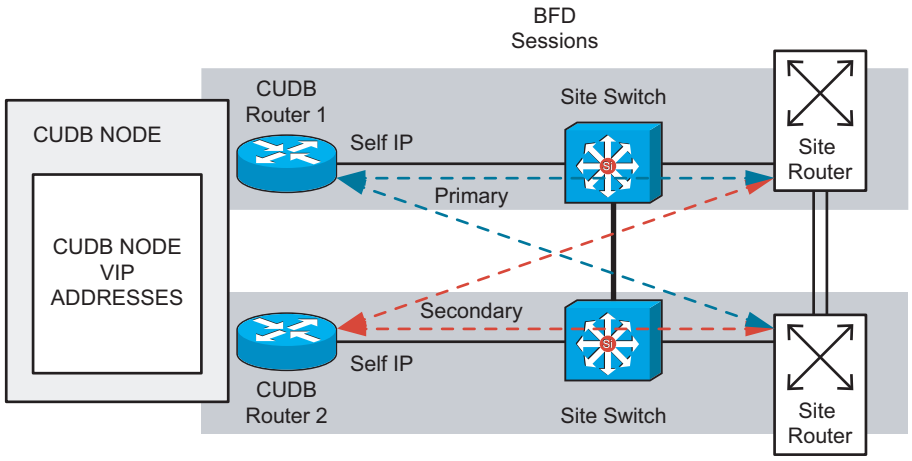


Figure 9 Multiple BFD-based L3 Redundancy

Refer to *RFC 5880: Bidirectional Forwarding Detection*, Reference [40], for more information on BFD.

3.1.6 Platform Availability

This section provides information on the platform-level availability features of CUDB.

3.1.6.1 LDE

LDE is the operating system executed on all nodes. In addition to a Linux OS, LDE also provides the `Cluster Wide User Management` service. The service synchronizes users and groups between the blades or VMs, ensuring that all blades or VMs have the same view.

3.1.6.2 Core Middleware

Core Middleware is an implementation of standards specified by the Service Availability Forum (SAF) and designed to provide HA.

3.1.6.3 COM

Common Operation and Management (COM) is aligned with the Ericsson OAM Architecture.

3.1.6.4 VIP Redundancy

In case the CUDB system is deployed on native BSP 8100 hardware, then, as mentioned earlier, CUDB routers are infrastructure components providing external connectivity. Their role is to route traffic in and out the CUDB node blades and the external routers.

In case the CUDB system is deployed on cloud infrastructure, then the connection of the VMs to the external infrastructure is provided by the cloud infrastructure.

In both deployment options, the incoming traffic load distribution towards the blades or VMs of the internal cluster is provided by the VIP software component.

VIP is a cluster SW function that makes it possible to address services in the cluster blades or VMs with a single IP address representing the entire cluster. An external application client can use the VIP address to send a request to the cluster irrespective of which payload processor should actually serve the request.

Configured CUDB VIP addresses (such as traffic VIPs, CUDB VIPs, OAM VIPs, and so on) are announced through OSPF from the blades or VMs running VIP FE instances as part of the VIP cluster function. CUDB routers act as VIP Gateway Routers for the VIP traffic, receiving the Open Shortest Path First (OSPF) link updates from the VIP FEs. Each VIP address is announced with the same metric from each VIP FE, therefore each CUDB router has as many equal-cost paths as payloads running VIP FE to route incoming requests

towards cluster VIP addresses. CUDB routers use the Equal Cost Multipath (ECMP) protocol to evenly balance traffic over the available VIP FE links.

OSPF link supervision is performed by using the BFD protocol between the CUDB routers and the VIP FEs for faster link failure detection (see Figure 10).

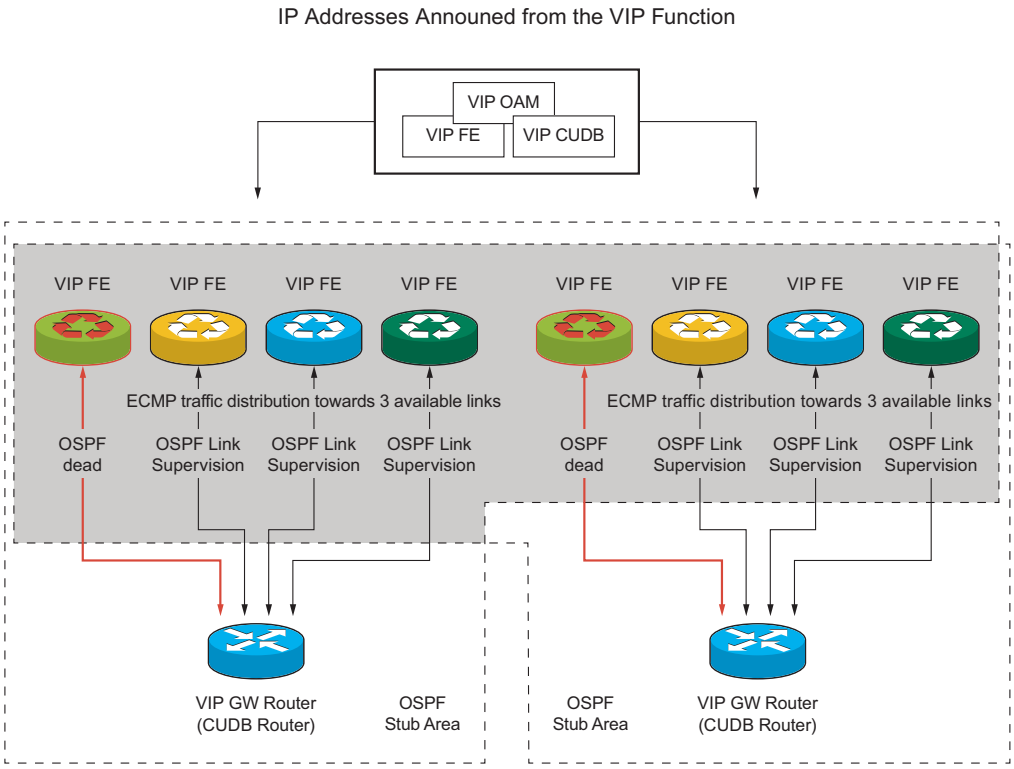


Figure 10 VIP FE Supervision

OSPF links rely on BFD supervision. When BFD detects a link failure, the unresponsive VIP FE is considered out of service, and the incoming traffic is balanced over the three available links through ECMP as shown in Figure 11.

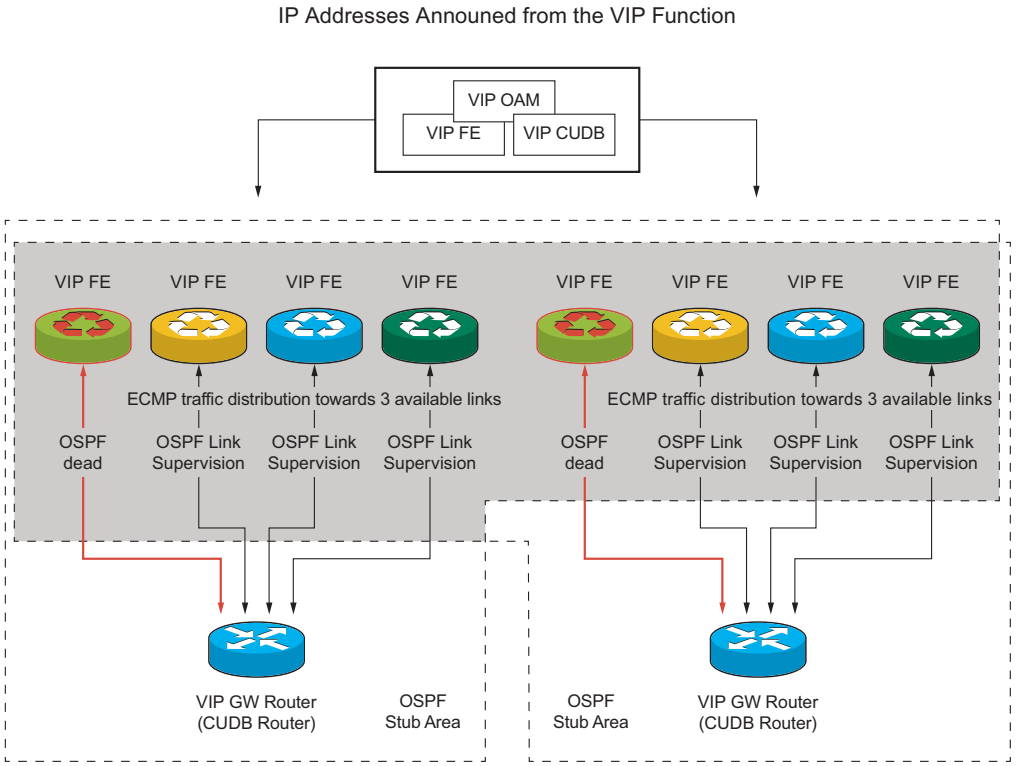


Figure 11 VIP FE Failure

For each CUDB VIP address, the number of running VIP FEEs and their location varies.

3.1.7 CUDB Node Function Availability

The main functions of the CUDB node are provided by a set of processes running on a set of redundant infrastructure pieces. The high availability of these processes is ensured by the HA functions provided by either the platform, or internally by other means.

Regardless of the HA methods, the system must guarantee the main CUDB node functions provided by these processes in case of both infrastructure and SW failures. To ensure this, the processes providing the same functions are redundantly executed in different infrastructure resources following different redundancy setups, as listed in the following subsections.

3.1.7.1 LDAP Access and Processing Function

The LDAP traffic access and processing is provided at each node by a set of LDAP FE processes handling a set of incoming connections from the platform VIP function. The incoming LDAP connections are distributed by the VIP among the available LDAP FEs according to a least-connection load balancing

algorithm. Refer to *CUDB Node Network Description*, Reference [2] for more information.

Only one LDAP FE process is executed on each payload (out of all the payloads assigned). These processes are internally monitored, and also restarted if get blocked or crashed. They are deployed following an all active `n+k` redundancy setup, where `n` stands for the number of processes needed to handle the nominal traffic rate of the CUDB node, while `k` refers to the desired redundancy level defined by `redundancyLevel` parameter (refer to *CUDB Node Configuration Data Model Description*, Reference [3]). This means that a maximum of `k` number of processes can fail to maintain the optimum level of service. Refer to *CUDB Deployment Guide*, Reference [4] for further deployment information.

When all LDAP FE processes are down, LDAP traffic is rejected and is not accepted again until `n` process are up and running, following redundancy setup explained above.

The alarms raised by the system in case of an LDAP FE process failure can be as follows:

- When one of the LDAP FE processes fails, an `LDAP Front End, Server Down` alarm is raised.
- When `k` number of processes fail, an `LDAP Front End, Processing Redundancy Lost` alarm is raised.
- When `k+1` number of processes fail, an `LDAP Front End, Processing Capacity Below Minimum` alarm is raised.

In all cases above, and each case an LDAP FE process fails, failure is logged with the following information: `(warning)<monitor_thread>: LDAP FE at <ldapfe>:<ldapfe_port> is now down.`

3.1.7.2 Data Store Function

The Data Store (DS) function provides storage for the different data stored in CUDB, and is composed by a set of databases deployed in the node. From a functional perspective, two types of databases are deployed: PLDBs and DS Units. The type of database deployed determines the internal architecture of the data storage solution, as well as the possible differences from the perspective of availability. Refer to *CUDB Data Distribution*, Reference [5] for further details on the CUDB database types.

The database cluster consists of different processes, each performing a different function. Failure of one or more of these degrades the performance and functionality of the database, or can even make it out of service. The main functions provided by the different processes are outlined below.

DS Unit Storage

Two DS Unit database processes are running within the database cluster, each of them executed on a different payload. The processes provide data storage services with an active-active redundancy setup. The architecture of the service is shown in Figure 12.

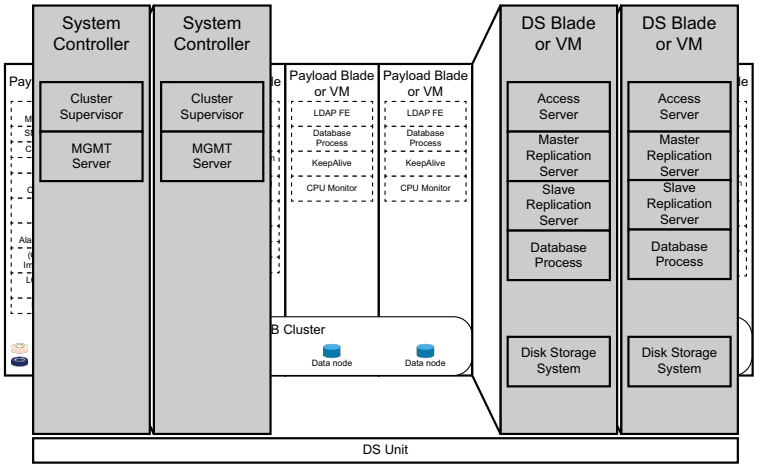


Figure 12 DS Unit Architecture

Both processes store the whole data set, which means that the entire data is replicated in both processes, with seamless access provided through both processes to reach them.

The possible failure scenarios and their respective availability measures are as follows:

- If one of the database processes fail, the surviving process keeps providing the database service, but the DS Unit is labeled as degraded by the monitoring function of the node. At the same time, the system raises a `Storage Engine, DS Cluster Node Down` alarm.

- If both database processes fail, the DS Unit is considered out of service by the monitoring function of the node, and the system raises a pair of `Storage Engine, DS Cluster Node Down` alarms.

  On system level, this is perceived as a DS Unit failure, and therefore results in raising a `Storage Engine, DS Cluster Down` alarm. See Slave DS Unit down and Master DS Unit down in Section 3.3.4 on page 47 for more information.

The process failures outlined above are resulting either from blade or VM failure, or process crash. If the failure is caused by a process crash, the `Cluster Supervising` function of the node is in charge of restarting it. Failure of this restart is logged with the following message: `(error) - ndbd process restart for <store_id> in host <host_ip> not completed.`

PLDB Storage

The PLDB is composed of an even number of database processes, each running on different payloads. Each pair of processes forms a group, within the main PLDB storage. See Figure 13 for an overview of the PLDB architecture.

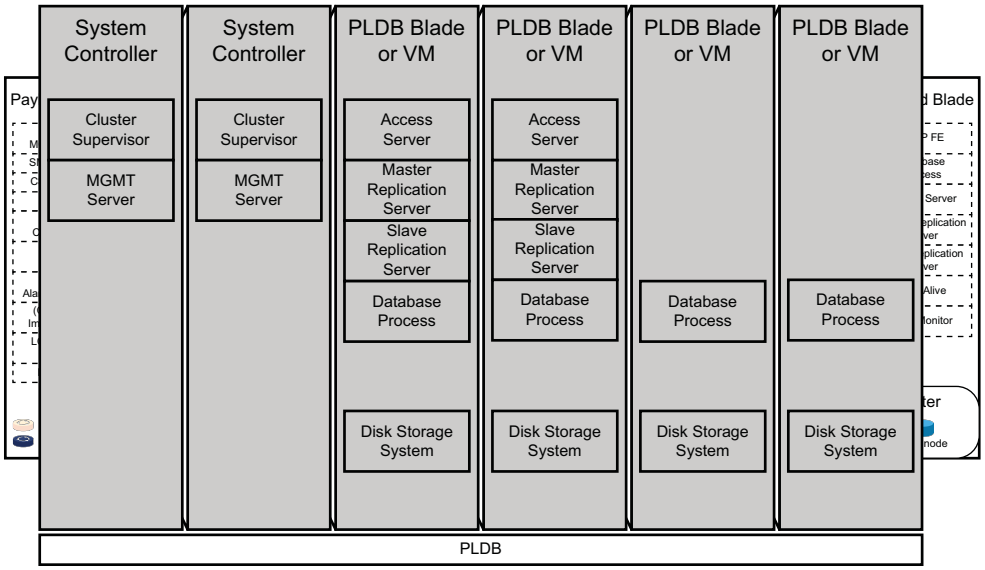


Figure 13 PLDB Architecture

The data stored in PLDB is partitioned in as many chunks as the number of existing groups. Each chunk is then assigned to a group. Within each group, both processes store the entire data chunk, which means that the entire data is replicated in both processes, with seamless access provided through both processes to reach them.

Note: In case of minimum setup with two payloads, there are only two database processes which make up a unique group and contain the whole PLDB data. This can be considered in the following failure scenarios for the recommended setup.

The possible failure scenarios and their respective availability measures are as follows:

- If one of the database processes fails in the group, the PLDB is labeled as degraded by the monitoring function of the node, and the system raises the `Storage Engine, PLDB Cluster Node Down` alarm.
- If multiple database processes fail, but all belong to different groups, the situation is the same as in the previous scenario, with the difference that it affects more than one group. The PLDB is labeled as degraded by the monitoring function of the node. The system also raises as many `Storage Engine, PLDB Cluster Node Down` alarms as the number of failed processes.

- If multiple database processes fail in the same group, the group stops providing service. The system raises as many `Storage Engine, PLDB Cluster Node Down` alarms, as the number of failed processes.

  As part of the stored data is not accessible in such case, the whole PLDB is considered out of service, and the system raises a `Storage Engine, PLDB Cluster Down` alarm. See Slave PLDB Unit Down and Master PLDB Unit Down in Section 3.3.4 on page 47 for more information.

The process failures outlined above are resulting either from blade or VM failure, or process crash. In case of a process crash, the `Cluster Supervising` function of the database is in charge of restarting it. The following text is logged: `(error) - ndbd process restart for <store_id> in host <host_ip> not completed.`

Geographical Redundancy Replication

Two replication processes are running, each on different payloads with active-active redundancy setup. Both processes can establish and maintain a replication channel with its mates in a geographically redundant cluster.

The processes in one cluster act as replication servers, while the other processes in the remote cluster act as replication clients. Two server-client channels are available, one of them being used as the primary channel, while the other (normally not used and not started) kept as secondary channel.

The replication setup applies to both the DS Units and PLDBs. If the DS Units have two payloads, the replication server processes are running in the same ones as the database processes. In case of the PLDB payloads, the replication server processes are running on the first two PLDB blades or VMs out of all the ones assigned to the database processes.

The possible failure scenarios and their respective availability measures are as follows:

- If the server or client processes in charge of secondary channel fail, the primary channel can still function without interruption. Therefore, no further actions are taken.
- If the server or client processes in charge of the primary channel fail, the secondary channel is started by the secondary client process, and is promoted to primary channel. No further actions are taken.
- If both the primary and secondary channels fail, the whole replication service fails. When Cluster Supervisor tries to recover replication, alarm `Storage Engine, Replication Channels Down in DS` or the `Storage Engine, Replication Channels Down in PLDB` is triggered. Further actions on the system level are only taken if both channel failures are due to simultaneous process failures in the master DS Unit or the master PLDB acting as replication servers. In such case, the affected DS Unit or PLDB is considered out of service, and the following actions take place:

– If the PLDB master is down, the system raises a `Storage Engine, PLDB Cluster Down` alarm. See Master PLDB Unit down in Section 3.3.4 on page 47 for more information.

– If the DS Unit master is down, the system raises a `Storage Engine, DS Cluster Down` alarm. See Master DS Unit Down in Section 3.3.4 on page 47 for more information.

See Section 3.3.3 on page 43 for more information on Geographical Redundancy.

Database Management

Database management is available for every DS Unit or PLDB. This type of process manages the rest of the processes within the database cluster by providing configuration data, starting and stopping services, ordering cluster backup, and so on.

Two database management processes are running at the same time, one on each SC with active-active redundancy setup.

The possible failure scenarios and their respective availability measures are as follows:

- If one of the processes fail, the other still functions without interruption. Therefore, no further actions are taken, but a `Storage Engine, PLDB Cluster Node Down` or `Storage Engine, DS Cluster Node Down` alarm is raised.

- If both processes fail, the whole management service fails, as no information is provided for the database cluster monitoring function. A pair of `Storage Engine, PLDB Cluster Node Down` or `Storage Engine, DS Cluster Node Down` alarms are raised. See Master DS Unit down, Slave DS Unit down, Master PLDB is down, and Slave PLDB is down in Section 3.3.4 on page 47 for more information.

  On system level, the whole cluster is considered out of service, while the affected DS Unit or PLDB is considered out of service. In such case, the following actions take place:

  – If the PLDB is down, the system raises a `Storage Engine, PLDB Cluster Down` alarm. See Master PLDB is down and Slave PLDB is down in Section 3.3.4 on page 47 for more information.

  – If the DS Unit is down, the system raises a `Storage Engine, DS Cluster Down` alarm. See Master DS Unit down and Slave DS Unit down in Section 3.3.4 on page 47 for more information.

SQL Access

SQL Access is available for every DS Unit or PLDB. This type of process provides internal SQL access to the data stored in the PLDB or a specific DS Unit. The service is used by other functions in the system, such as Application Counter.

Two SQL Access processes are running at the same time, one on each PLDB payload with active-active redundancy setup.

This redundancy setup applies to both DS Units and PLDBs. In case of DS Units with two blades or VMs, the SQL Access server processes are running on the same two payloads as the database processes. In case of the PLDB blades or VMs, the SQL Access server processes are running on the first two PLDB payloads out of all the ones assigned to the database processes.

The possible failure scenarios and their respective availability measures are as follows:

- If one of the processes fail, the other still functions without interruption. Therefore, no further actions are taken, but a `Storage Engine, PLDB Cluster Node Down` or `Storage Engine, DS Cluster Node Down` alarm is raised.

- If both processes fail, the whole SQL access service fails. A pair of `Storage Engine, PLDB Cluster Node Down` or `Storage Engine, DS Cluster Node Down` alarms are raised. See Master DS Unit down, Slave DS Unit down, Master PLDB is down, and Slave PLDB is down in Section 3.3.4 on page 47 for more information.

  No further actions are taken on system level.

Data Store Function in Maintenance Mode

Data Store in Maintenance Mode is a special scenario affecting an entire DS Unit or PLDB, and is available when a specific DS Unit or PLDB in a CUDB node is set to maintenance mode either manually, or due to a failed restoration or restart of the Data Store.

In such case, the PLDB or DS Unit is considered out of service, and a `Storage Engine, DS Cluster in Maintenance Mode` or `Storage Engine, PLDB Cluster in Maintenance Mode` alarm is raised. See Master DS Unit down, Slave DS Unit down, Master PLDB is down, and Slave PLDB is down in Section 3.3.4 on page 47 for more information.

3.1.7.3 Monitoring Function

The monitoring functions available as part of the HA feature are listed below.

Cluster Supervising Function

The Cluster Supervising function is further described in Section 3.2 on page 34:

Two instances of the Cluster Supervising function are running for each PLDB and DS Unit database cluster configured in the node, each of them on a different SC with active-standby redundancy setup.

HA for the Cluster Supervising function is provided by means of SAF AMF, ensuring that one active and one standby instance are always running in the two SCs. In case the active instance dies or just does not reply to the AMF heartbeats, the standby instance receives a status change to become active. Refer to SAF AIS, Reference [41] for more information on SAF AMF.

If both the active and standby process instances fail, the affected DS or PLDB is considered out of service. See Master DS Unit down, Slave DS Unit down, Master PLDB is down, and Slave PLDB is down in Section 3.3.4 on page 47 for more information.

System Monitoring Function

The System Monitoring function is further described in Section 3.2 on page 34.

System Monitoring is performed with two processes, each running on a different SC with active/active redundancy setup.

If one instance of the process fails, the `KeepAlive` process attempts to restart it. If this process fails to restart the System Monitor instance, the error is logged in the system.

If both instances of System Monitoring fail, the entire CUDB node is considered out of service by the system. See CUDB Node is down in Section 3.3.4 on page 47 for more information.

LDAP FE Monitor

LDAP FE monitoring is provided by the following processes:

- LDAP FE Monitor running on SCs:

  The LDAP FE Monitor is a HA process monitoring the LDAP FE processes. It raises the `LDAP Front End, Server Down` alarm if needed. Depending on the number of monitored alarms, the monitor can raise an `LDAP Front End, Processing Redundancy Lost` alarm, an `LDAP Front End, Processing Capacity Below Minimum` alarm, or both. HA is provided by means of SAF AMF, ensuring that the function runs with an active and a standby instance, one in each of the SCs.

  If the active instance fails or does not reply to the AMF heartbeats, the standby instance receives a change of status to become active. Refer to SAF AIS, Reference [41] for more information on SAF AMF.

If both the active and standby instances fail, the function is considered out of service. No specified alarm is available for such scenario, and no further node or system level actions are taken.

- LDAP FE Monitors running on payloads:

  The LDAP FE Monitors work on a per payload blade or VM basis: one instance runs on each payload. Each process keeps checking the existence and basic functionality of the LDAP FE process running on the corresponding payload.

  If one instance of the process fails, the `KeepAlive` process attempts to restart it. If `KeepAlive` fails to restart the LDAP FE monitor instance on the payload, the error is logged in the system.

Storage Performance Monitoring Function

The CUDB system monitors the health of the local storage system on the SC and payload blades or VMs with the Storage Performance Monitoring function. In case infrastructure-related storage errors are detected, the function raises the alarm *Server Platform, Storage Performance Degradation Detected*, Reference [33]. For a CUDB system deployed on native BSP 8100 hardware, if the error is detected on a payload blade, then the blade is shutdown to minimize the impact in the corresponding DS or PLDB unit.

Storage Performance Monitoring consists of two processes running on each blade or VM. One runs continuously, the other one runs periodically, both are driven by the `KeepAlive` process. If this process fails to start or restart the monitoring process instances, the error is logged in the system.

3.1.7.4 Node OAM Function

The CUDB Operation and Maintenance Function consists of the following subfunctions:

- LDAP Counters
- CUDB Fault Management
- CUDB Performance Management
- CUDB Configuration and Node Management
- CUDB Software Management
- CUDB Application Logging

The Fault Management and Performance Management functions are supported by the Ericsson SNMP Agent (ESA). ESA centralizes the alarms sent from all the components of a CUDB node, and gathers them in an unique SNMP flow.

There are two ESA instances running, each one in a different SC server. ESA is installed in both SCs because high availability. Both ESA instances are running at the same time, but alarms are sent to any of them, not to both. These alarms are internally copied between them for persistency.

If the ESA instance fails, it is automatically restarted by the system. However, if the SC where ESA is running fails, there is still a running ESA instance in the redundant SC.

The **Configuration and Node Management**, **Software Management**, and **Application Logging** features are provided by the platform, therefore their HA characteristics are also inherited from the platform (see Section 3.1.6 on page 22 for more information).

The only exception within the **CUDB Configuration and Node Management** function is the CUDB Object Implementor, which is in charge of validating configuration updates. It consists of two process instances, each running on one of the two SCs with an active-standby redundancy setup. In case the active instance dies or does not reply to the AMF heartbeats, the standby instance receives a change of status to become active. Refer to SAF AIS, Reference [41] for more information on SAF AMF.

3.1.7.5 Application Notification Function

Application Notification is supported by the monitoring function in charge of monitoring the configured changes and events in the master DS Units of the CUDB node. When detected, these events are processed to send the notification towards the configured application FE. Refer to *CUDB Notifications*, Reference [6] for further information on CUDB notifications.

Application Notification is a high availability process. HA is provided by means of SAF AMF, ensuring that the function runs with an active and a standby instance on two payloads.

In case the active instance dies or does not reply to the AMF heartbeats, the standby instance receives a change of status to become active. Refer to SAF AIS, Reference [41] for more information on SAF AMF.

3.2 Data Availability Coordination Function

The Data Availability Coordination (DAC) function is in charge of monitoring data storage and system availability to facilitate CUDB in taking the appropriate decisions and actions to ensure high availability in case of failures.

The main components of DAC are as follows:

- BC cluster: Provides availability data for the coordination framework and leadership election.

- System Monitoring (SM) : Ensures the high availability of the CUDB system at any time.
- Cluster Supervising : Ensures the high availability of the data storage clustering service.

The components are described below in more detail.

3.2.1 Blackboard Coordination Cluster Service

The BC cluster service allows coordination between distributed processes through a shared hierarchical name space (similar to a replicated file system). See Table 1 for more information on how the processes are distributed among the blades or VMs.

The name space is stored in the memory, and persistently to the disk storage system. This assures the high persistency, high throughput and low latency of the BC cluster.

The high availability of the BC cluster is provided by all the BC servers composing the cluster. Therefore, the BC Cluster service is available as long as the majority of the BC servers composing the BC Cluster are available. This means that a single failure in one of the BC servers does not affect the service provided.

Figure 14 illustrates the BC clustering solution described above.

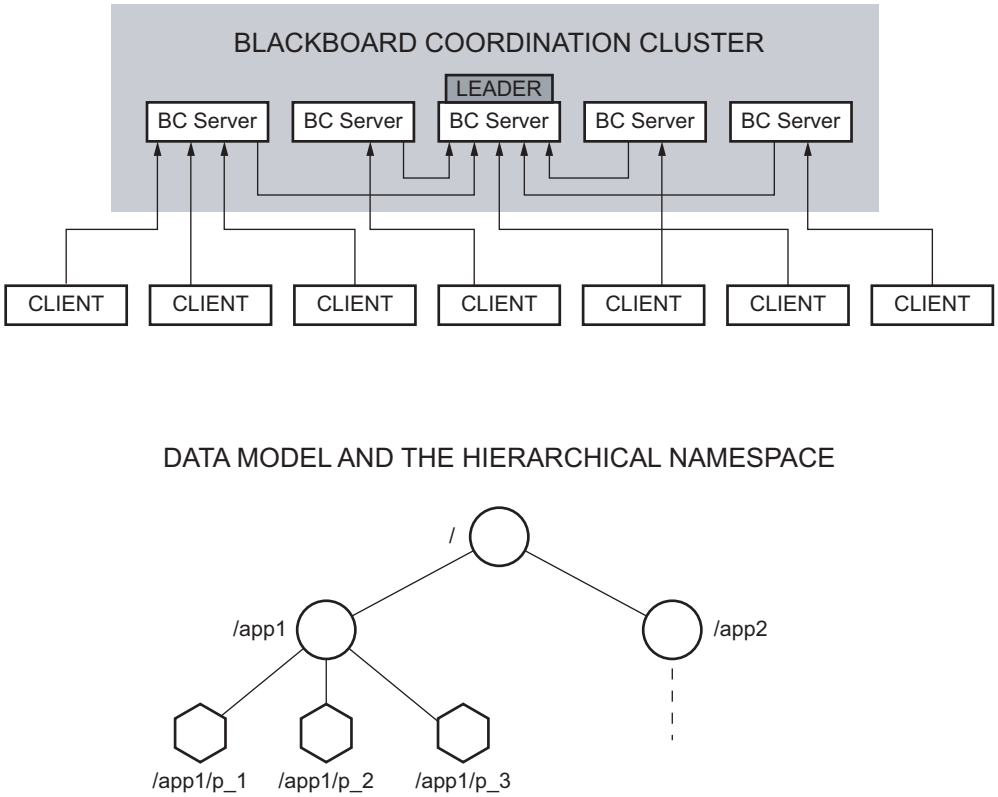


Figure 14 BC Clustering Solution

The information shared by the BC cluster provides the client processes a complete view on the CUDB system.

Apart from coordinating information sharing between client processes, the BC cluster also supports the system in other important tasks, such as leadership election among the client processes.

Each CUDB site is served by one BC cluster, no matter how many nodes are deployed on the site. The operational principles of the BC clusters depend on whether they operate on node level, or system level. These principles are described below.

BC Cluster Principles on Node Level

From the perspective of a single CUDB node, the following principles are applicable:

- Both System Monitor instances (in active/active modes) are connected to the site BC cluster.
- Both Cluster Supervising instances (in active/standby modes) are connected to the site BC cluster.

Figure 15 below illustrates the above principles.

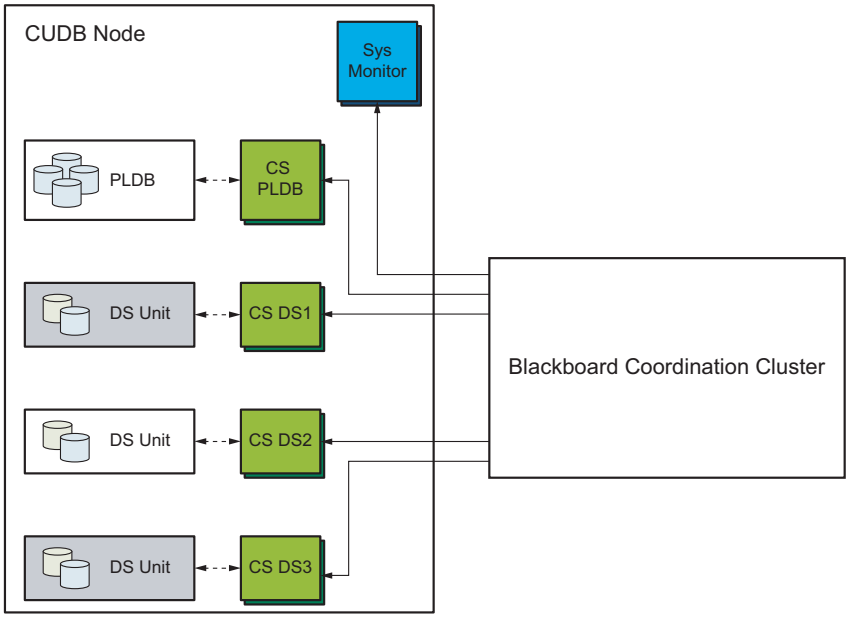


Figure 15 BC Cluster from Node Perspective

BC Cluster Principles on System Level

From the perspective of the CUDB system, the following principles are applicable:

- The System Monitor leader instance is connected to all remote BC clusters in other CUDB sites to report local incidents occurring in other sites.
- Some site-level System Monitor instances are connected to remote BC clusters to improve site availability incident reporting.

Figure 16 below illustrates the above principles.

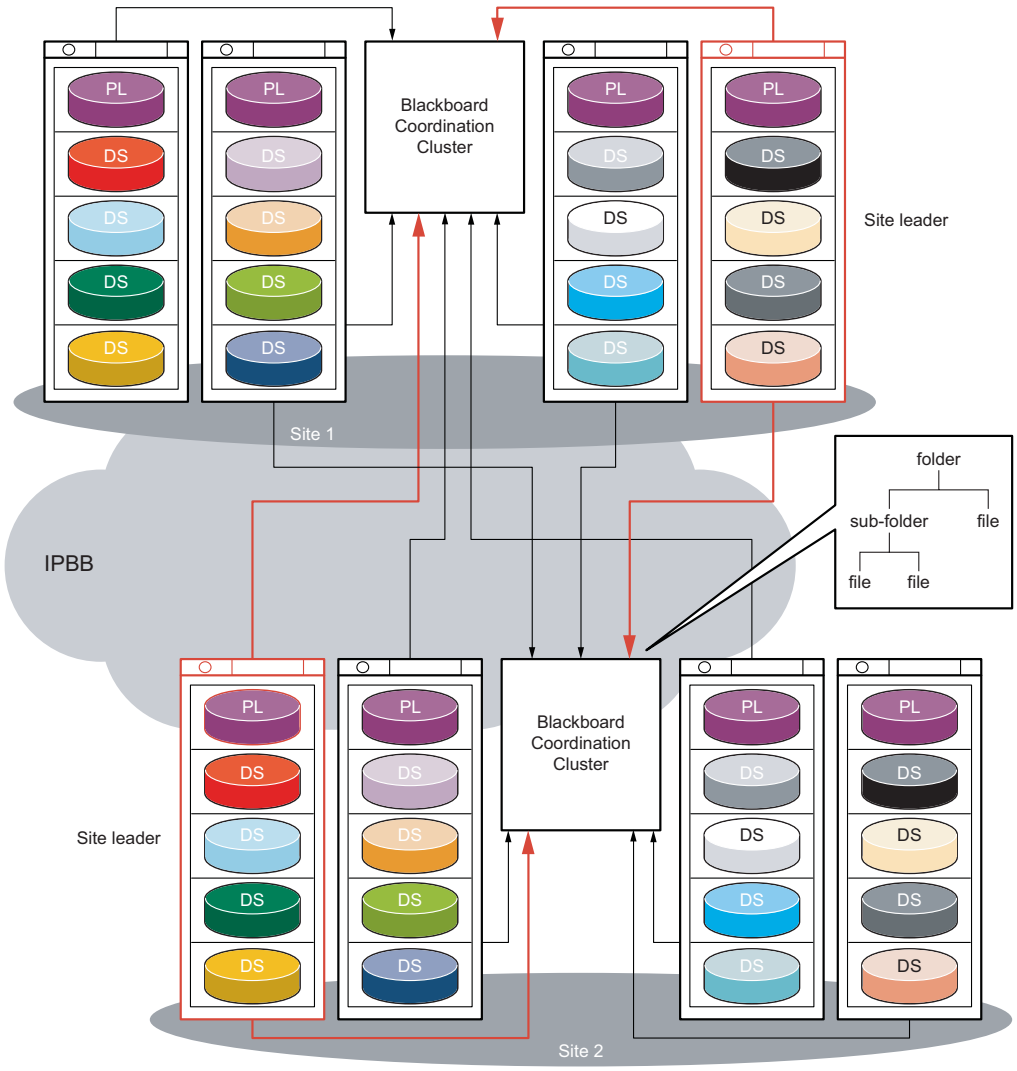


Figure 16 BC Clustering from System Perspective

3.2.2 Cluster Supervising Service

The cluster supervising service is in charge of monitoring and checking the availability of the CUDB data storage system and taking the necessary actions if high availability is compromised. The service performs the following actions:

- It manages and supervises the different processes of database clusters located in the DS Units or PLDBs.
- It uses the BC cluster to report the status of the database cluster.
- It manages the replication channels according to the information received from the BC cluster (such as master-slave situation and decisions). Each

cluster supervising instance is subscribed to the changes occurring in the master PLDB and master DSGs in the BC cluster.

3.2.3 System Monitoring Service

The System Monitoring (SM) processes perform the following actions depending on the Node ID of their node, and whether they are elected as leader SM:

- All SM processes perform the following actions:
  - Publish in-site BC Cluster runtime status information (such as the amount of free memory, degradation state) about all accessible clusters.
  - Broadcast the collected BC Cluster runtime status information to other software components within the CUDB node that cannot access the BC Cluster.
  - Electing one of the SM instances of the CUDB site as the SM leader (or in other words, site leader).
- The SM processes of the two nodes with the lowest Node ID in the site also perform the following action:
  - Establish auxiliary inter-site connections to all remote BC Clusters in the current partition.

    Note: If one of these processes is the site leader, then it keeps both the auxiliary connection as well as the site leader connection to remote BC Clusters.
- The SM leader (or site leader) performs the following actions:
  - Connects to local and remote BC Clusters.
  - Subscribes to changes occurring in the status of site resources.
  - Updates status changes for the local and remote BC Clusters.
  - Handles site incidents.
  - Coordinates PLDB and DSG master election with all site BC Clusters.
  - Elects new PLDB and DSG master by using the status and replication information received from all BC Clusters.
  - Updates the memory usage and status of the site-based DS in remote BC Clusters.
  - Checks for network partition incidents in case of inter-site connectivity problems.

  - Runs health checks in background to detect and fix potential wrong decisions.
  - Raises and clears alarms.

- The SM partition leader is the SM leader running in the site with the lowest site id within a network partition, and it performs the following actions:
  - Handles incidents triggered by connectivity problems between sites.
  - If needed, also handles master reelections for the PLDB and DSGs in the partition after network partition incidents.
  - When Provisioning Assurance is configured, it handles the re-provisioning request towards the Provisioning Gateways after non-manual master changes.

The below example describes how system monitoring operates in case a master DS is down:

1 The cluster supervising function reports the situation for the BC cluster of the corresponding site.

2 The SM leader of the site where the incident happened receives the information, since it is subscribed to such events.

3 The SM leader notifies all the BC clusters of the system (that is, all clusters on all sites) that a new master election is needed in the affected DSG.

4 When Provisioning Assurance is configured, the SM partition leader will notify all the BC clusters of the system that normal provisioning is locked.

5 The cluster supervising functions receive this notification, as they are subscribed to such events.

6 The cluster supervising functions in the nodes where this DSG is allocated report the replication information for their BC clusters.

7 The SM leader receives this information, since it is subscribed to such data. It uses this information to appoint a new master for the affected DSG.

8 The appointment decision is published for all BC clusters.

9 When Provisioning Assurance is configured, the SM partition leader will send a message to the Provisioning Gateways to request re-provision of the provisioning orders since the specified time.

10 The cluster supervising functions and SM functions also receive this information, as they are subscribed to such events.

11 The cluster supervising functions use this information to set up replication channels to the newly-appointed master.

12 The SM function in every node propagates this decision in their respective node for those components which have no access to the BC cluster).

13 When Provisioning Assurance is configured, once the Provisioning Gateways finish the re-provision, the SM partition leader will notify all the BC clusters of the system that normal provisioning is unlocked.

3.3 System Level Availability

The CUDB system consists of a set of CUDB nodes coordinated and interacting with each other and providing seamless access to all data stored in the system. To achieve this, the CUDB system is logically and structurally divided to two internal tiers offering different functions: the Processing Layer (PL) and the Data Store (DS) Layer. On top of them, the DAC functions are running, providing high availability for the entire system in case of multiple incidents (described later). Refer to *CUDB Technical Product Description*, Reference [1] for more information on the CUDB architecture.

As these layers provide different functions, they require different availability measures. The detailed description of system level availability measures and services is outlined in the below subsections.

3.3.1 Processing Layer System Functions

The PL has two major system functions:

- **LDAP Traffic Access and Processing**

  LDAP data access and traffic processing is a system function distributed between the different CUDB nodes in the system. By means of their LDAP FEs, every CUDB node provides LDAP access and processing capabilities to reach any data in the whole CUDB System.

  On system level, every CUDB node is providing a Single Point of Access (SPoA) to the whole database, which means that every piece of data can be reached through any CUDB node of the system. Therefore, a single CUDB node failure does not prevent application FEs from accessing CUDB, because in such cases, they can connect to the system through a different CUDB node.

- **Data Distribution Supported in PLDB**

  CUDB distributes data in different Data Stores (DS) which are deployed either on the same or different CUDB nodes, located in the same or different CUDB sites. Data distributed among several CUDB sites implies the geographical distribution of data. The distribution information together with the identity data and any other data is stored in PLDB. Refer to *CUDB Data Distribution*, Reference [6] for further information on data distribution.

Due to performance reasons, PLDB is replicated in at least one CUDB node in each CUDB site of the system. However, this replication scheme requires to keep all the replicas synchronized. Due to this reason and the asynchronous nature of replication, a master-slave replication model is followed where one PLDB is elected as master (receiving read-write operations), while the rest of the PLDBs become slaves (replicating updates from the master and receiving read-only operations).

With the functions outlined above, the PL availability of the CUDB system ensures that PLDB data is available, if the following conditions are met:

- The availability of the master and slave PLDBs is secured.
- The availability of the CUDB nodes hosting the master and slave PLDBs is secured.
- The availability of the CUDB sites hosting the related CUDB nodes is secured.

Any failure, or the combination of failures in the above availability services can result in the following failures:

- Master PLDB failure.
- Slave PLDB failure.
- CUDB node failure, CUDB site failure, or both, impacting the master PLDB.

3.3.2 Data Storage System Functions

Data availability on the CUDB system level is ensured by Geographical Redundancy.

The Geographical Redundancy HA feature replicates a DS in one or two CUDB nodes, each located in a different CUDB site. Replication to one node results in double replication, while replication to two other nodes results in triple data replication.

Due to the double or triple replication scheme, the necessity to keep all the replicas synchronized, and the asynchronous nature of replication, a master-slave replication model is followed where one DS Unit is elected as master (receiving read-write operations) while the rest of the DS Units (one or two) become slaves (replicating updates from the master and receiving read-only operations).

Therefore, the DS availability of the CUDB system ensures that all partition of data is available if the following conditions are met:

- The availability of the master or slave DS Units hosting the actual piece of data is secured.

- The availability of the CUDB nodes hosting the master and slave DS Units is secured.
- The availability of the CUDB sites hosting the CUDB nodes is secured.

The CUDB system supervises the availability of the master replica for each DSG and selects a new master in case the current master becomes unavailable. Any failure, or the combination of failures in the above availability services can result in the following failures:

- DS master failure.
- CUDB node failure, CUDB site failure, or both, impacting several DS masters in each affected node.

3.3.2.1 Automatic Mastership Change

The Automatic Mastership Change (AMC) function aims to return the master role of PLDB or DSGs to the replica instance configured with the highest priority, in case the mastership was previously removed from it and it is now ready for service again. When AMC is enabled, it checks if the PLDB and DSG masters are located on the highest priority replica of each group. In case AMC detects that the master PLDB or any master DSGs are not located on the highest priority replica of their specific group (due to any failure in the PLDB or DSG replica, CUDB node, or CUDB site or if the master is changed with the `cudbDsgMastershipChange` command), it moves it there, provided that the highest priority replica of the group is healthy.

AMC uses configured time slots during which it checks the location of the PLDB and DSG master replicas, and changes the mastership if it is not assigned to the replica with the highest priority. If the PLDB of the node with the highest priority replica is a slave, and is not replicating from its master, then AMC will not move the mastership to that replica to avoid future master movements in case the PLDB needs to be restored from a master backup.

For more information on configuring the AMC function, refer to *CUDB System Administrator Guide*, Reference [7].

3.3.3 Geographical Redundancy

The geographical redundancy feature replicates a DS in one or two CUDB nodes, each located in a different CUDB site. Replication to one node results in double replication, while replication to two other nodes results in triple data replication, ensuring high data availability. Refer to *CUDB Data Distribution*, Reference [6] for more information on data distribution.

Two possible configurations are available when configuring geographical redundancy of data:

- **Double Geographical Redundancy**

In this case, data located on each DSG is stored on two DS Units.

- **Triple Geographical Redundancy**

  In this case, data located on each DSG is stored on three DS Units.

All DSGs in the CUDB system must have the same redundancy level configured (in other words, all of them must possess the same number of DS slave replicas).

The functions provided by the redundancy configurations listed above are described in the following subsections.

3.3.3.1 Double Geographical Redundancy Configuration

Double geographical redundancy (also known as 1+1 redundancy) allows having two DSs per partition, one master DS and one slave DS replica, each one hosted in a different CUDB site. See Figure 17 for an illustration of double geographical redundancy.

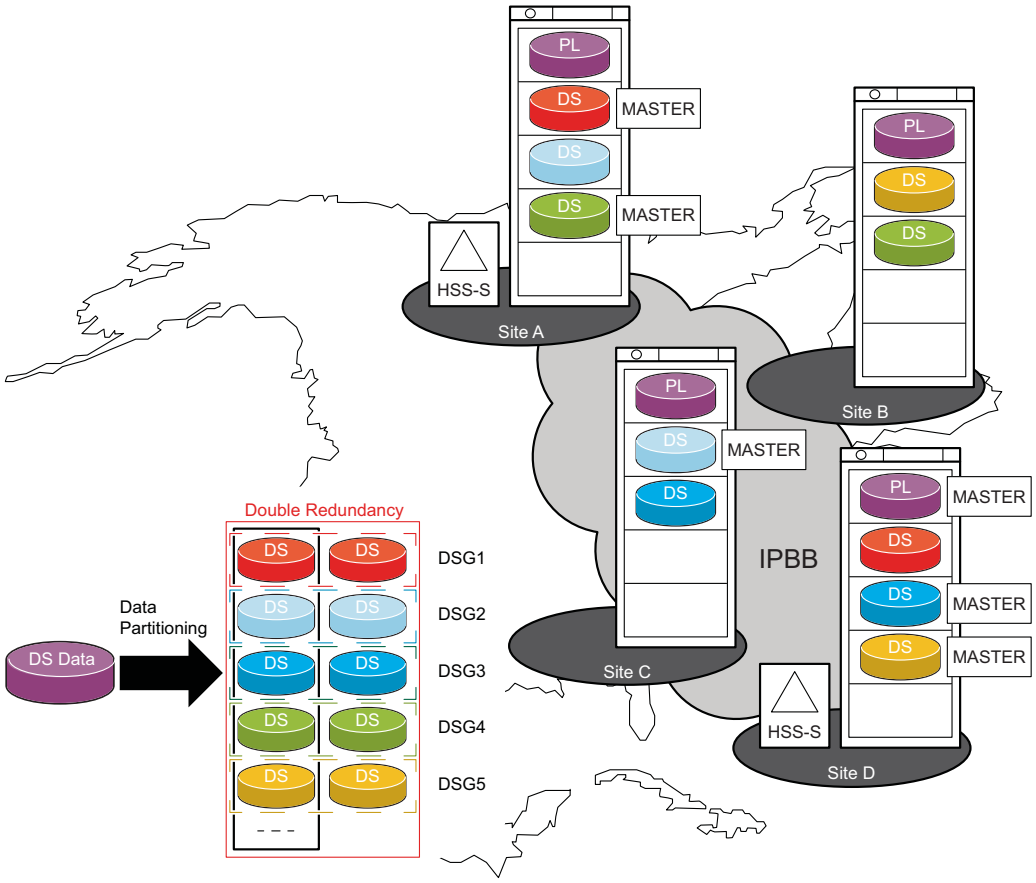


Figure 17 Double Geographical Redundancy Deployment

3.3.3.2 Triple Geographical Redundancy Configuration

Triple geographical redundancy (also known as 1+1+1 redundancy) allows having three DS replicas per partition, one master DS and two slave DS replicas, each one hosted by a different node in a different CUDB site. See Figure 18 for an illustration of triple geographical redundancy.

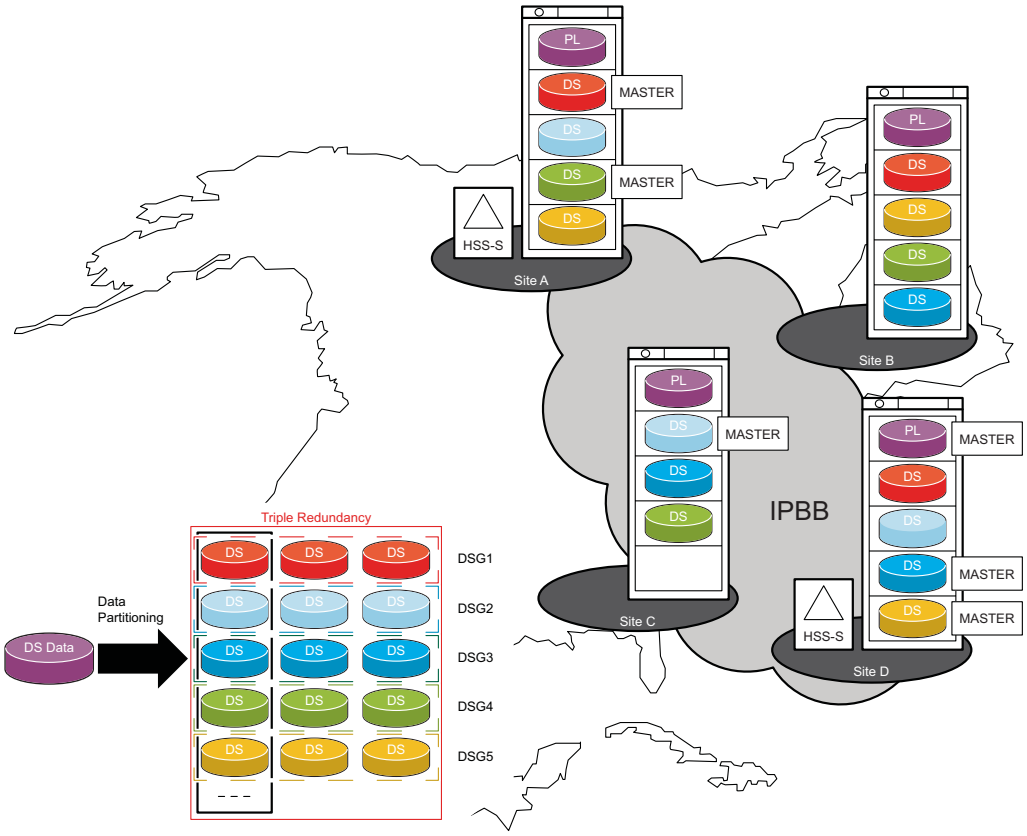


Figure 18 Triple Geographical Redundancy Deployment

3.3.3.3 Inter Database Cluster Replication

A CUDB system consists of more than one CUDB node, therefore the PLDB is redundant: the system houses more than one copy of PLDB data and every CUDB site must host at least one copy of PLDB. A CUDB system also consists of more than one CUDB site, therefore it can have up to three copies of the same DSG data. The synchronization of these copies is supported by inter database cluster replication.

The different DS Units (or PLDBs) use **asynchronous replication**. Write operations are executed in the master replica for a group, then the data changes ripple through to the rest of the replicas in the same DSG or PLDB group. Slave replicas take updates from the master replicas asynchronously.

Replication Channels Availability

The line of communication established between the replication servers (one for each of the masters and slaves) in a clustered database is referred to as the replication channel. The CUDB system has high availability in replication channels.

For fault tolerance, the replication channels are established in pairs from every slave replica to its master replica. While one of the replication channels is active, the other is on standby. If the active replication channel fails, the standby channel takes over the active role automatically.

Replication is handled and monitored by the Cluster Supervising Function on CUDB node level (see Section 3.1.7.3 on page 31). Failure of both replication channels does not trigger any CUDB system availability measures, and requires no other administrative action than dealing with alarms.

If desired, replication can be secured. Refer to *CUDB Security and Privacy Management*, Reference [8] for more information on securing databases.

Replication Delay Monitoring

The replication delay between each PLDB or DS slave replica and its master replica is continuously monitored. If the delay exceeds the threshold of `replicationTimeDelayAlarmThreshold`, the alarm `Storage Engine, Replication Delay Too High In DS` or `Storage Engine, Replication Delay Too High In PLDB` is raised. For more information on these alarms, refer to *Storage Engine, Replication Delay Too High In DS*, Reference [9] and *Storage Engine, Replication Delay Too High In PLDB*, Reference [10], respectively. The delay is expressed as the estimated time needed for the slave replica to catch up with the master replica. Replication delay monitoring is only active if the replication is up. It is performed on each slave replica by the associated Cluster Supervising function.

For more information about `replicationTimeDelayAlarmThreshold` parameter, refer to *CUDB Node Configuration Data Model Description*, Reference [3].

If a mastership change takes place while this alarm is raised, a problem can arise during re-establishing replication channels. See Section 3.3.5.4 on page 73 for further information on the possible actions to perform in that case.

3.3.4 System Resiliency to Multiple Failures

As described earlier, the CUDB system HA deals with the availability of the DS and PLDB master databases, the availability of the CUDB Nodes hosting those master databases, and the availability of the CUDB sites hosting those CUDB nodes.

The DAC supervises the availability of the master replica for each database, and selects a new master in case the configured master becomes unavailable.

Any failure, or the combination of failures in the availability services can result in the following failures:

- Master DS or PLDB failure.
- Slave PLDB failure.
- CUDB node failure, CUDB site failure, or both, impacting several master DSs or PLDBs in the affected nodes.

Based on the above information, the response of the CUDB system to component failures depends on the type of failure, and the geographical redundancy configuration. The possible failures, and the data availability responses are described below.

Note: Alarms related to node failures and the events described below are listed in Section 3.1.7.2 on page 26. Refer to *CUDB LDAP Data Access*, Reference [11] for further information on the specific LDAP error codes.

Double or Triple Geographical Data Redundancy

In case of double and triple geographical redundancy, the behavior of the CUDB system is mostly the same. The only difference stems from the different number of slave DSs (double geographical redundancy has one slave DS on each DSG, while triple redundancy has two).

If the master DS Unit is down and the mastership change process is launched, a master election algorithm is performed. See Section 3.3.6 on page 74 for more details on this process.

The initial redundancy configuration is shown on Figure 19.

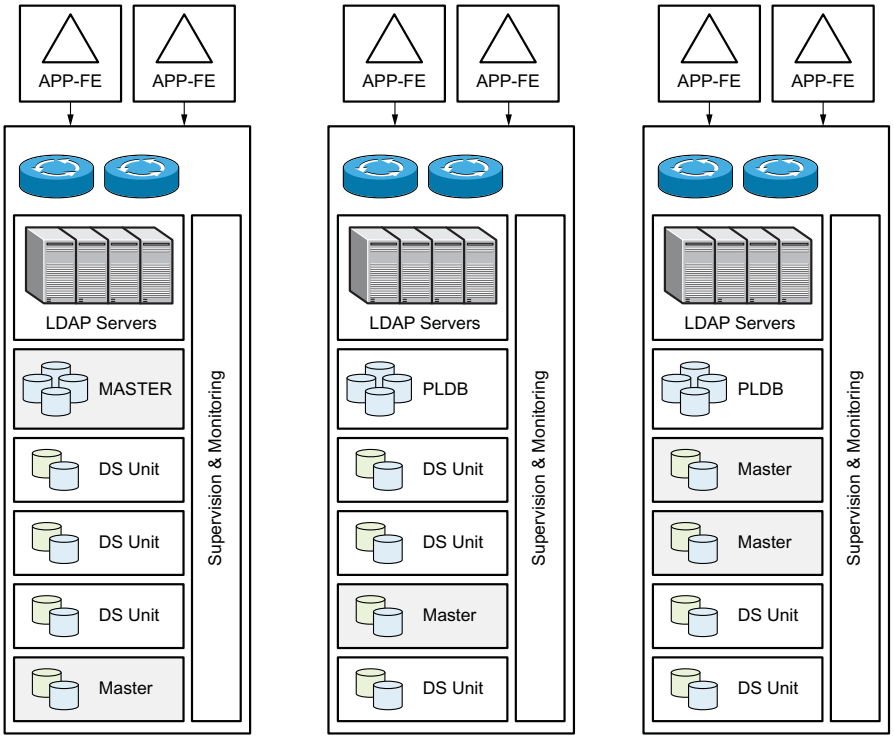


Figure 19 Initial Geographical Data Redundancy Configuration

The different failures that may occur are as follows:

- **Slave DS Unit is down**

  No actions are taken at system level. The availability of this data partition is not interrupted, since all traffic is directed to its master DS Unit replica. See Figure 20 for an illustration.

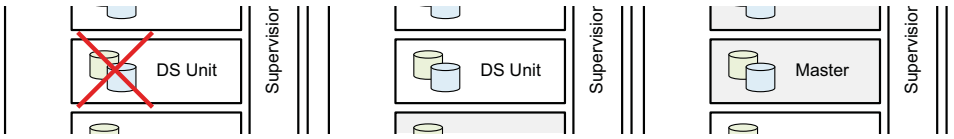


Figure 20 Slave DS Unit Down in Geographical Redundancy Configuration

- **Master DS Unit is down**

  In case the master DS fails, the mastership change process is launched to choose a new master for the data partition. In case of double geographical redundancy, the single slave replica of the DSG is elected as the new master. In case of triple geographical redundancy, the master election algorithm is initiated (see Section 3.3.6 on page 74 for more information). During the mastership change process, the traffic directed to this DSG

in any CUDB node responds with an LDAP error. See Figure 21 for an illustration of the Master DS Unit failure, and Figure 22 for an illustration of the mastership change process.

As mentioned in Section 3.3.3.3 on page 46, in case a high replication delay exists before such mastership change takes place, a problem can arise during re-establishing the replication channel in the slave nodes for the impacted DSG. For more information, see Section 3.3.5.4 on page 73.

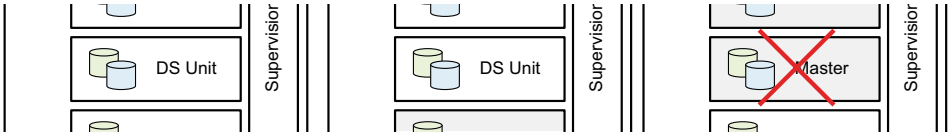


Figure 21 Master DS Unit Down in Geographical Redundancy Configuration

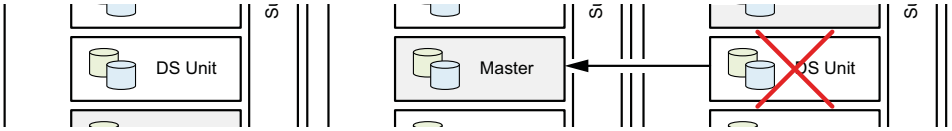


Figure 22 Mastership Change Process During Master DS Unit Down Failure

- **Slave PLDB is down**

  If a PLDB is down in a CUDB node, the full node is considered unavailable (see Figure 23). All LDAP operations in the affected CUDB node respond with an LDAP error. All master DSs hosted in this node become inaccessible. Therefore, the procedure outlined at Master DS Unit is down applies for each affected master DS (see Figure 24). The DSs in the node that become slaves start, and continue replication from the new master. If the PLDB that is down is the only PLDB in a CUDB Site, all master DSs hosted in CUDB nodes without PLDB in that site will become inaccessible, applying the same procedures as previous statement.

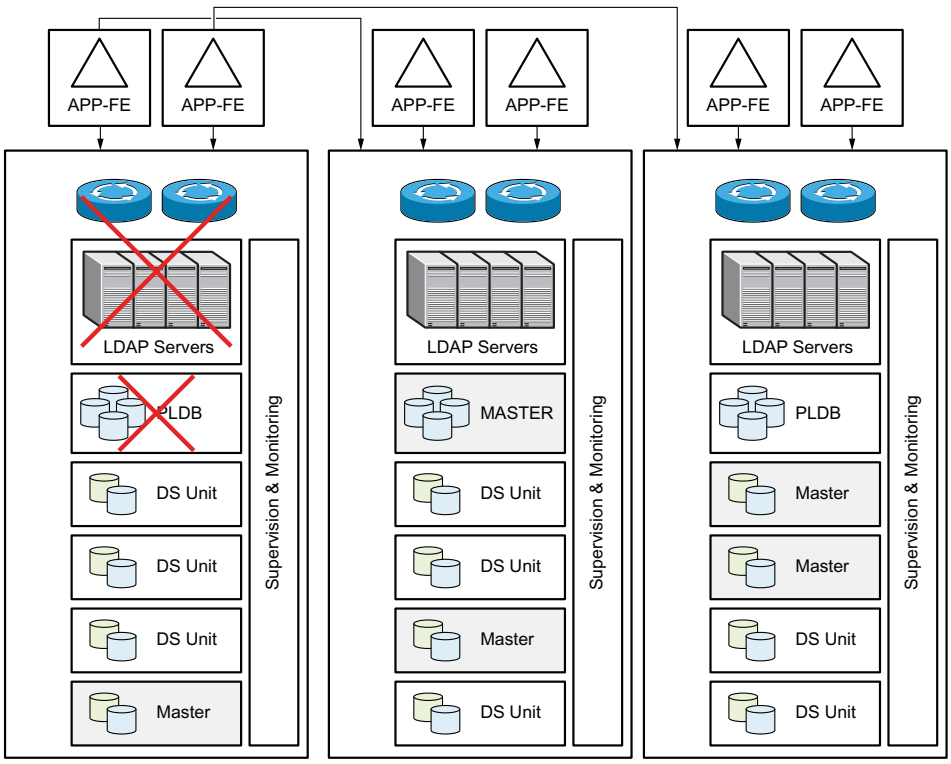


Figure 23 Slave PLDB Down Failure

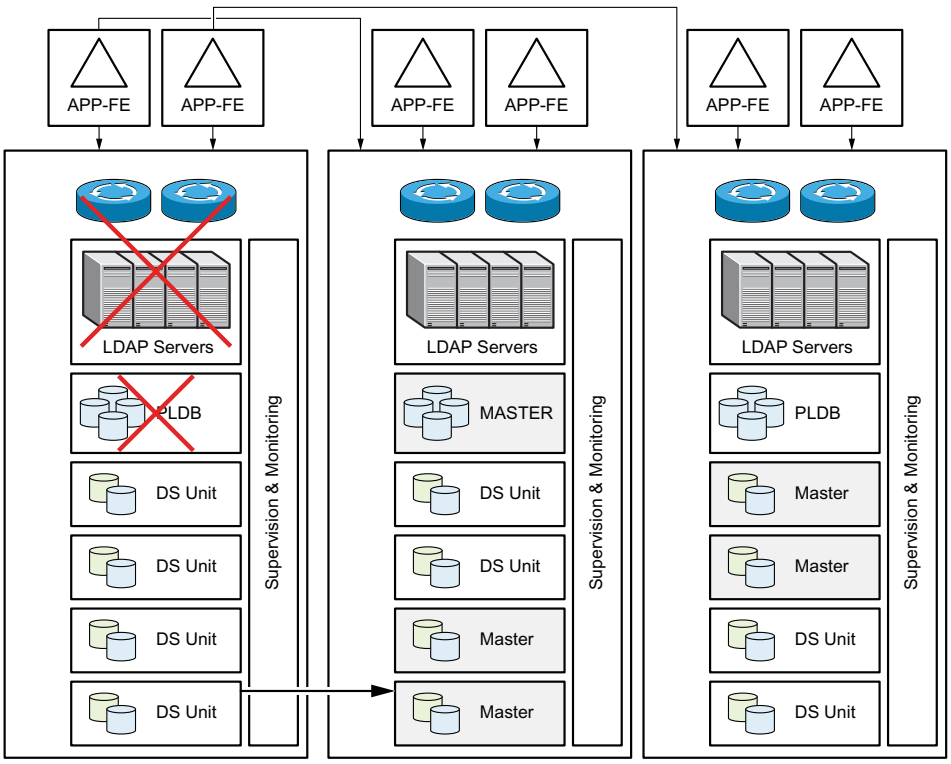


Figure 24 Mastership Change Process During Slave PLDB Down Failure

- **Master PLDB is down**

  As stated in Slave PLDB is down, when a PLDB is down in a CUDB node, the full node is considered unavailable, and the detailed procedure applies (see Figure 25). Additionally, a mastership change procedure is performed, and a new master PLDB is elected from the slave PLDB replicas present in the system (see Figure 26). During this mastership change, traffic is rejected with an LDAP error code. If the Master PLDB that is down is the only PLDB in a CUDB Site, all master DSs hosted in CUDB nodes without PLDB in that site will become inaccessible, applying the same procedures as in section Slave PLDB is down.

  As mentioned in section Section 3.3.3.3 on page 46, in case a high replication delay exists before such mastership change takes place, a problem can arise during re-establishing the replication channel in the slave nodes for the PLDB. For more information, see Section 3.3.5.4 on page 73.

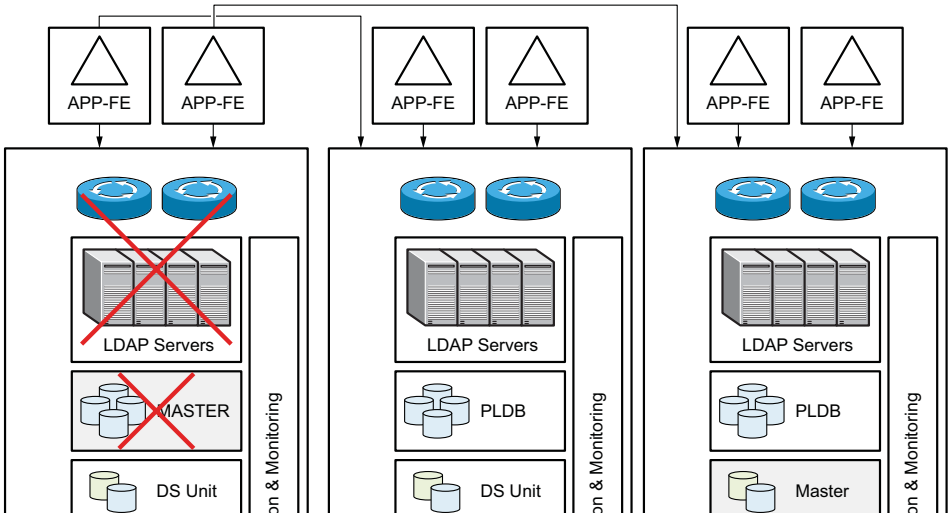


Figure 25 Master PLDB Down Failure

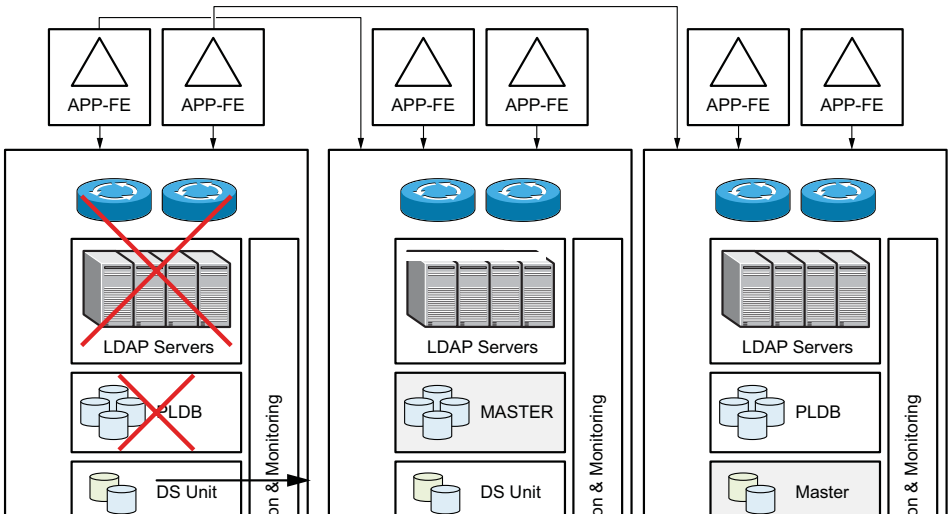


Figure 26 Mastership Change Process During Master PLDB Down Failure

- **DSG down**

  This situation is the result of multiple failures in all the DSs in the group. The data partition hosted in the failing DSG becomes unavailable. All LDAP operations to this data partition in any CUDB node respond with an LDAP error, while the rest of operations in other CUDB nodes continue operating without interruption. Degraded service performance of the CUDB system is expected.

- **CUDB Node is down**

  If the failing CUDB Node is hosting the PLDB master replica, the system behaves as described on Master PLDB is down. However, if the failing

CUDB node is hosting a PLDB slave replica, the system behaves as described on Slave PLDB is down.

The CUDB node is considered unavailable in the following cases:

– The Local System Monitoring function decides that the local node is down due to the following events:

  • Both database processes of the same group in the PLDB are down.

  • Both replication servers in PLDB are down.

  • Both management servers in PLDB are down.

  This is communicated to the rest of the CUDB nodes in the system.

– The entire CUDB node is unavailable due to the following events:

  • Both System Monitor processes fail at SW level.

  • Infrastructure failures in both SCs or in the PLDB blades or VMs ending up in a PLDB down situation.

  • Failure in the SCX or CMX boards in the first subrack (in case the CUDB system is deployed on native BSP 8100 hardware).

  • Massive infrastructure failure.

  If any of the failed SM processes is an SM leader, that is the crashed CUDB node runs the SM leader, an SM leader failover takes place and an SM running on another CUDB node on the same site of the crashed CUDB node takes the SM leader role.

  In any of the above cases, the CUDB node is considered unavailable by the SM leader of the site (that is, the SM leader hosted in one of the remaining CUDB nodes of the affected site). If no other CUDB nodes are located on the site, the site is considered down, and a partition incident is detected by the rest of the SM leaders in the rest of the CUDB sites.

  The `Control, Remote Node Unreachable` alarm is raised in the CUDB node hosting the SM leader of the site in case more than one CUDB node is present in the site when the affected CUDB node becomes unavailable.

  If no other CUDB nodes are located on the site, the alarm `Control, Remote Site Unreachable` is raised by the SM leaders on the rest of the sites.

The recovery process from the various failures are as follows:

– **Recovery from PLDB down**

After recovering from PLDB down, the full CUDB node is considered recovered. All the DS masterships have already been reassigned to other DSs in other CUDB nodes. If the failed PLDB was operating as master, it is also reassigned to another PLDB in another CUDB node. No mastership restoration is performed in any case. The recovered PLDB and the rest of the DSs become slaves, and have to establish replication from the new masters. Replication lag is expected in the new slaves.

– **Recovery from DS Unit down**

  No actions are taken at system level. If the recovered DS was master before failing, no mastership restoration is needed. The newly elected master DS continues operation. The recovered DS establishes replication from the new master. Replication lag is expected.

– **Recovery from CUDB node down**

  The CUDB node can recover from the above situation. The SMs in the node start reporting in the BC cluster, so they become visible again for the rest of the SMs in the system, especially for the SM leaders. From the point of view of the CUDB system, no actions are taken since the procedure is the same as the one outlined for Recovery from PLDB downand Recovery from DS Unit down.

- **CUDB site failure**

  The site failures relevant for CUDB system availability are as follows:

  – Site is down: Multiple failures result in **all** the CUDB nodes of a site are down or the BC cluster of the site is down.

  – Site is isolated: Isolation occurs if the SM leader of a site cannot contact the BC clusters in other sites, or the SM leaders of other sites are not reporting in the BC cluster of the affected site. The possible scenarios are as follows:

    - The SM leader of a site cannot contact the BC clusters in other sites. This prevents the SM leader of the site to report status on the other BC clusters.

    - The SM leader of Site A can contact the BC cluster in Site B, but detects that there is no SM leader on site B reporting to the BC cluster of site A.

      Auxiliary information (such as other SMs on site B reporting for site A) can be used by the SM leader of Site A to detect if SM leader election is ongoing on Site B, and wait before the partition incident is raised.

  Site failure is perceived in different ways, depending on the type of failure and the location where it occurs:

- Site isolation splits the CUDB system into two or more smaller systems. Each of these systems perceives the isolation in a different way, and takes a set of decisions to guarantee as much service as possible, maintaining full data consistency.

- Site down is perceived as an isolation by the rest of the sites. Obviously, the site unavailable does not take any measures.

In all cases, the systems or sites failing to communicate with other sites (and therefore perceiving isolation) cannot be sure of the reason of the isolation, and if the other sites are really down at all. Therefore, the sites attempt to continue providing service. Then (based on the total number of sites in the system, and the number of sites visible) each systems defines the current split situation, and decides the measures to take. See Section 3.3.5 on page 56 for more information.

- **CUDB Site Recovery**

  See Section 3.3.5 on page 56 for more information on site recovery.

3.3.5 CUDB System Split

A CUDB system can suffer either network or site failures that can split the system in different subdivisions or partitions, or cause single site failures. The set of actions carried out by the system to control these situations is known as Split Management. The purpose of this section is to describe how the potential split situations are managed by the CUDB system.

CUDB configuration states in which site the individual CUDB nodes are located. Site information is also added to the CUDB system, so all the CUDB nodes are associated with a site. A site is considered visible from other sites as long as the SM leaders of other sites can connect and report to the BC cluster of the site. By using site information and the shared information in the BC cluster of their sites, SM leaders receive accurate information about the CUDB system status.

The three possible scenarios managed by this function are *Majority*, *Minority* and *Symmetrical Split*. More information is provided on them in the following sections.

Site visibility is always perceived from the SM leader hosted in one of the CUDB nodes of a site. In case the system is split into more than two subdivisions, each SM leader is able to communicate with a set of CUDB sites, but unable to detect another set of sites, no matter if this latter set is also split in other divisions. Therefore, every SM leader considers itself belonging to a group (the group of visible sites) and compares that with the group of not visible sites.

In such situation, incidents and decisions that affect the group or the partitions of the group are taken only by one of the SM leaders of the partition (being considered as the SM partition leader).

3.3.5.1 Split Situations

When a failure occurs in the system or in the interconnecting network, the CUDB System (as perceived from the SM leader), can be split in two different ways:

- Asymmetrical split situation: in this case, the group of visible sites and the group of not visible sites have a different number of sites.
- Symmetrical split situation: in this case, the group of visible sites and the group of not visible sites have the same number of sites.

Although it is already specified that this view is perceived from the perspective of a single SM entity, the following must also be added:

Provided that the SM and BC functions work properly in sites X, Y, and Z, the IP back-bone interconnecting the CUDB sites must fulfill the following conditions at all times:

- Condition 1: `X<)Y` must be bijective, that is if `X<)Y` is **true**, then `Y<)X` is also **true**. Likewise, if `X<)Y` is **false**, then `Y<)X` is also **false**.
- Condition 2: `X<)Y<)Z` is transitive, that is if `X<)Y` is **true** and `Y<)Z` is **true**, then `X<)Z` is also **true**. Likewise, if `X<)Y` is **true**, but `Y<)Z` is **false**, then `X<)Z` must also be **false**.

Note: `X<)Y` (read as "site X sees site Y") is a relationship between site X and Y which (provided that the SM and BC functions are working properly on both sites), can take one of the following two values:

- If the SM of site X can report in the BC of site Y, then `X<)Y` is **true**.
- If the SM of site X cannot report in the BC of site Y, then `X<)Y` is **false**.

Under normal circumstances the IP back-bone fulfills the above conditions, but if for any reasons any of the above conditions are not met, asymmetric partition scenarios can arise. In these scenarios the CUDB system tries to maximize the service assigning the masterships to the sites with greater visibility, that is, the sites that are reachable by the maximum number of sites, thus minimizing potential data loss and database inconsistencies.

Condition 1 might not be met if a link in the IP back-bone is congested in one direction, but not in the opposite one.

Condition 2 might not be met if the IP back-bone (or a part of it) interconnecting the CUDB sites has a mesh topology, and therefore if the link between two sites fail, working links can still be available between those sites and a third site.

More details on the above-mentioned scenarios are provided below. See Figure 27 for an illustration of the different split scenarios.

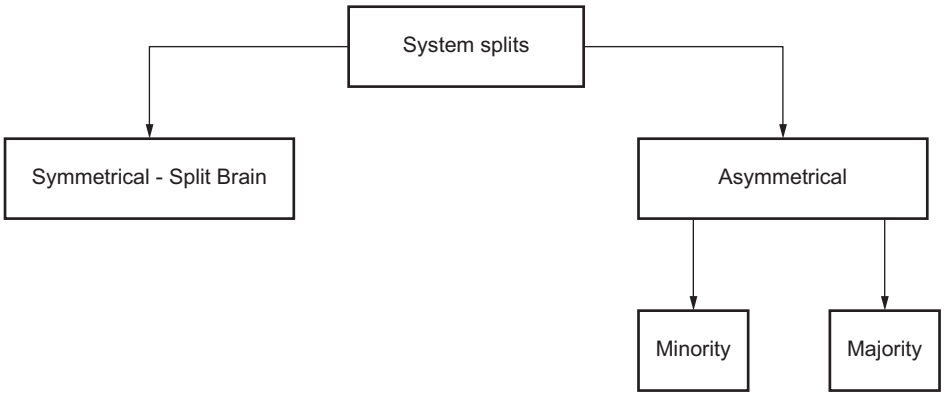


Figure 27 System Split Scenarios

Asymmetrical Split Situations

In an asymmetrical split situation, the majority takes over the master replicas in the system, whereas the minority will relinquish all their masterships. These situations can overcome in most deployments with more than two sites. The details of the majority and minority situation are listed below.

- **Subsystem in a Majority Situation**

  Subsystem in a majority situation is the normal situation of a CUDB system. In such situations, the SM leader in a given site is connected and exchanging information with BC clusters of at least more than a half of the total CUDB sites in the system. Auto-removed sites described below this section are not taken into account to compute the total number of sites in the system.

  However, even in the case there is a majority group, it is possible to lose some subscriber partitions if all replicas of a complete DSG are in the unreachable part. The system returns an LDAP error code for operations requesting profiles from that partition. Refer to *CUDB LDAP Data Access*, Reference [11] for further information on LDAP Access in case of system split. CUDB provisioning can be done in the majority group of the system.

  In a split situation where the network is subdivided in two subnetworks that make a majority and a minority situation for a particular DSG, the subnetwork in majority situation must select the master replica from their own DSG replicas. The process is carried out through the Master Election Algorithm (see Section 3.3.6 on page 74 for more information).

  As shown in Figure 28, the system allows `Add-delete`, `Modify` and `Search` operations for both Provisioning and Traffic users. Refer to *CUDB LDAP Data Access*, Reference [11] for more information on LDAP access.

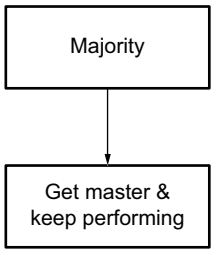


Figure 28 System Behavior in Majority Situation

- **Subsystem in a Minority Situation**

  A CUDB partition is in a minority situation when that partition includes less than half of the sites in the CUDB system. Auto-removed sites described below in this section are not taken into account to compute the total number of sites in the system. In minority partitions the following two scenarios are possible:

  – For every DSG present in the minority partition there is at least one replica that is not hosted in this minority partition. In this case, the SM partition leader in the minority partition assumes that there is another partition in a majority situation (that is a partition that includes more than half of the sites in the CUDB system) and that the majority partition is able to serve all traffic. In this situation, the CUDB nodes in the minority partition do the following:

    - release their masters replicas,
    - stop processing traffic, sending an LDAP error code back to the applications, urging them to reconnect to other CUDB nodes,
    - raise alarms to warn about the situation.

  – All replicas of one or more DSGs are fully hosted in the minority partition (that is, all CUDB sites where there is a replica of that DSG, or those DSGs, are in the minority partition). In such scenario, the assumption that the majority partition is able to serve all traffic does not hold, because it is not able to serve traffic for any fully hosted DSG in the minority partition. In this situation, the CUDB nodes in the minority partition do the following:

    - keep masters for any fully hosted DSG in the minority partition,
    - process traffic normally, according to the rules described in *CUDB LDAP Data Access*, Reference [11].

  Behavior in both scenarios above are shown in Figure 29.

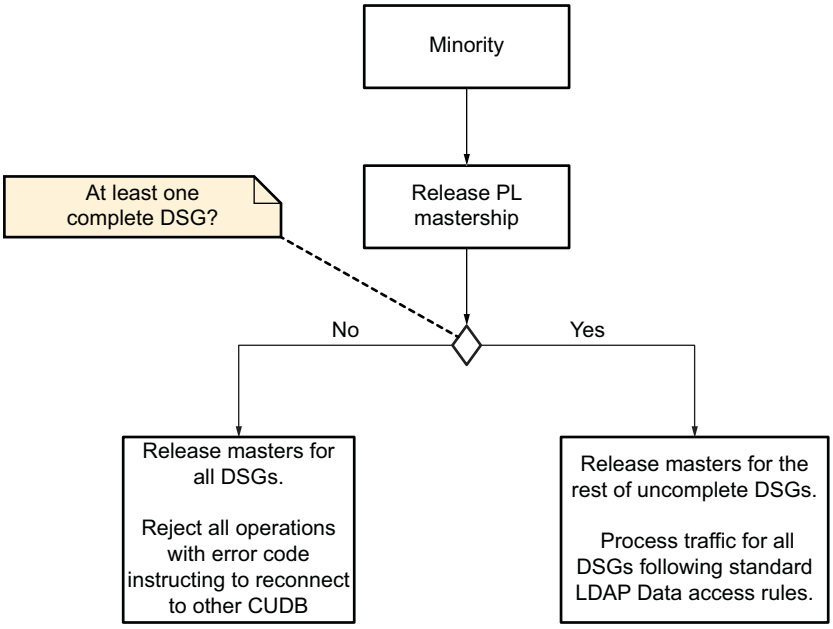


Figure 29 System Behavior in Minority Situation

Figure 30 illustrates a majority or minority scenario within a CUDB system.

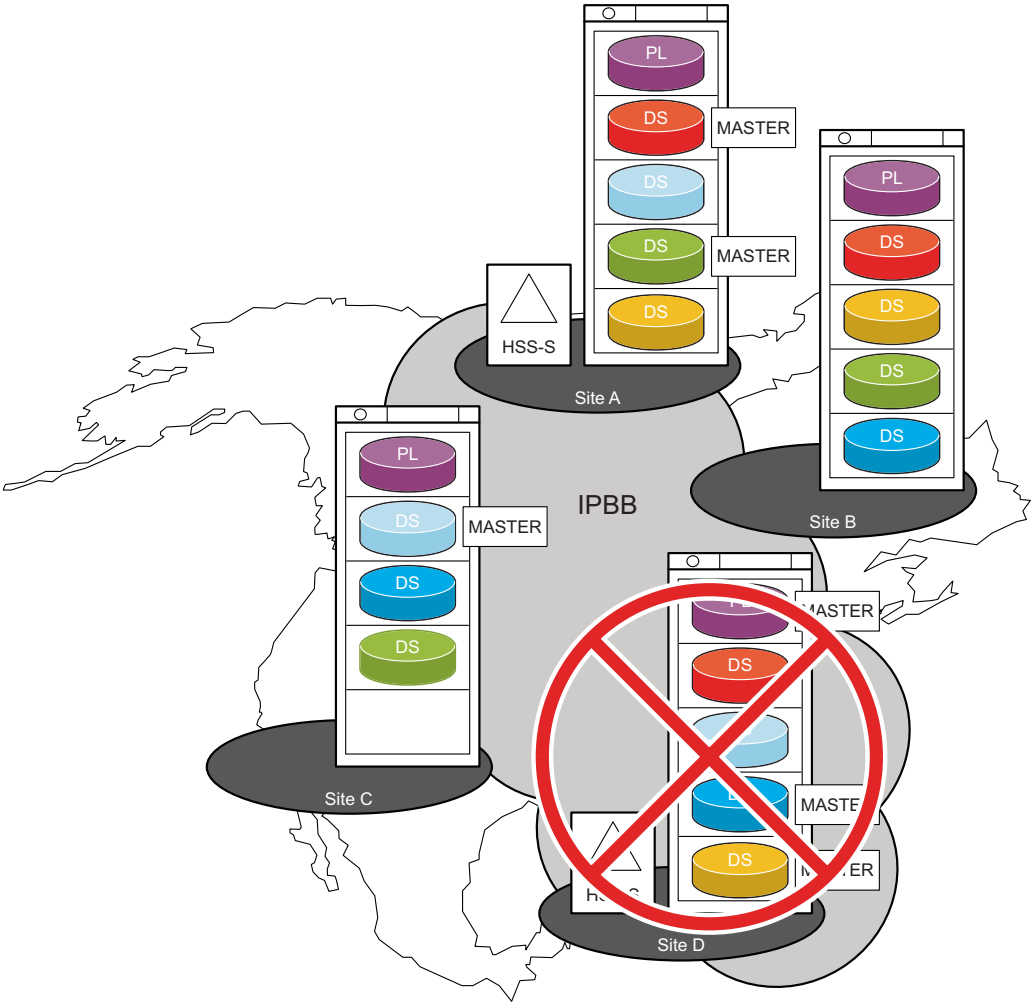


Figure 30 Majority and Minority Scenario in the System. Site D is in Failure or Isolation.

Figure 31 shows the resulting situation after applying the control measures described above.

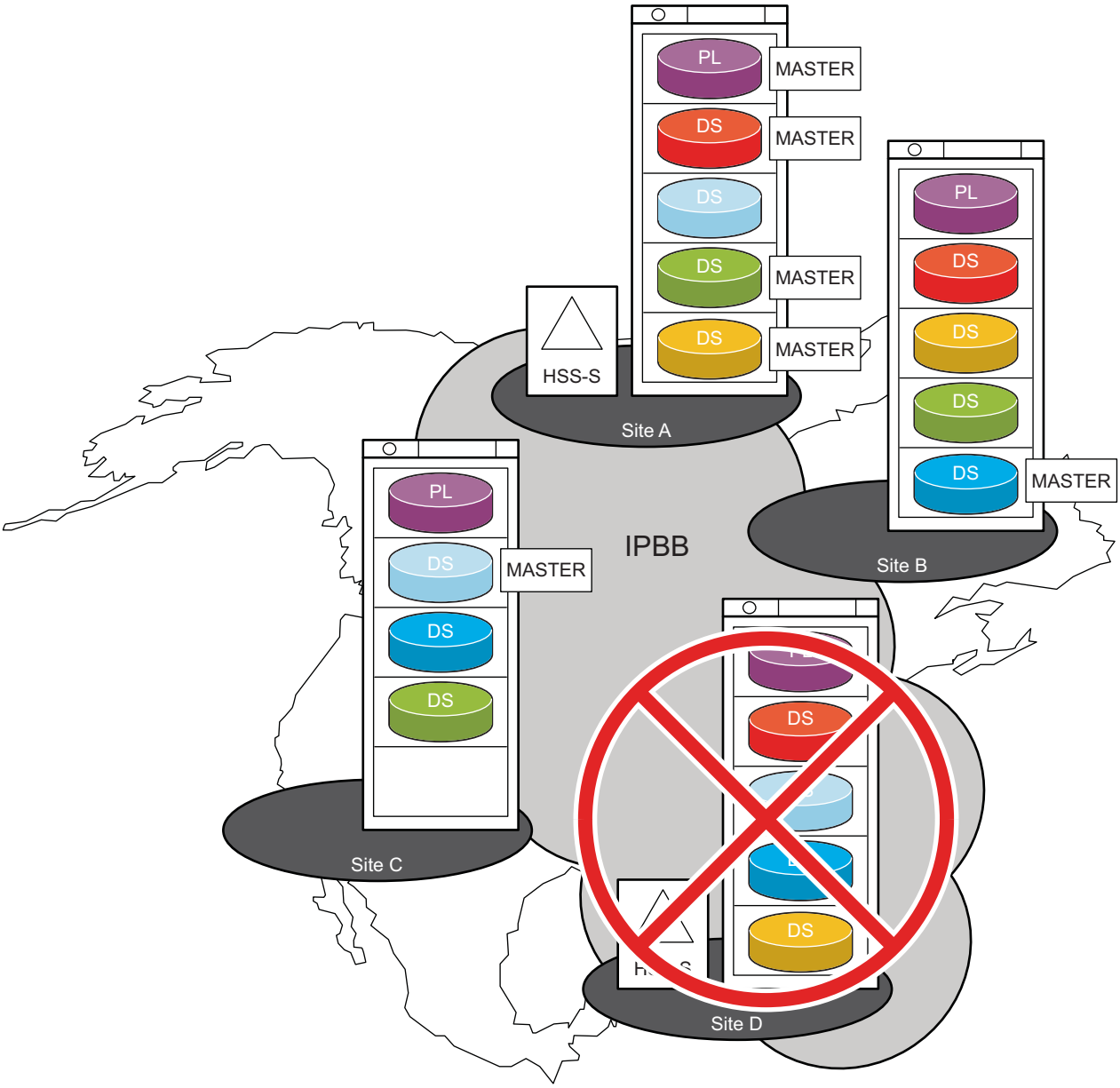


Figure 31 Majority and Minority Scenario in the System After Master Reassignment for PLDB and DSs

- **Auto-removed Sites**

  When a site is unreachable by the SM leaders forming a majority group, it is marked as auto-removed by the SM partition leader of the majority group. When a site is marked as auto-removed, it is not part of the system anymore, so it is not taken into account when computing the total number of sites. This is reported by the logging function of the SM leader, and by a number of alarms depending on the cause of the site being unreachable.

The next time a node has to decide if it is in a majority, minority, or a symmetrical split situation, the auto-removed sites will not be part of the total number of nodes.

When an auto-removed site goes back to service, it is included again in the list of sites taken into account by the SM leaders in a majority partition to compute the total number of sites in the system.

Symmetrical Split Situation

A site or set of sites is in a symmetrical split situation if the SM partition leader is connected to and exchanging information with exactly half of the total CUDB sites (or BC clusters of these sites) in the system. Auto-removed sites are not taken into account to compute the total number of sites in the system.

Figure 32 illustrates such a split situation with two nodes. (of which one is down), while Figure 33 shows the same situation with an IP Backbone failure.

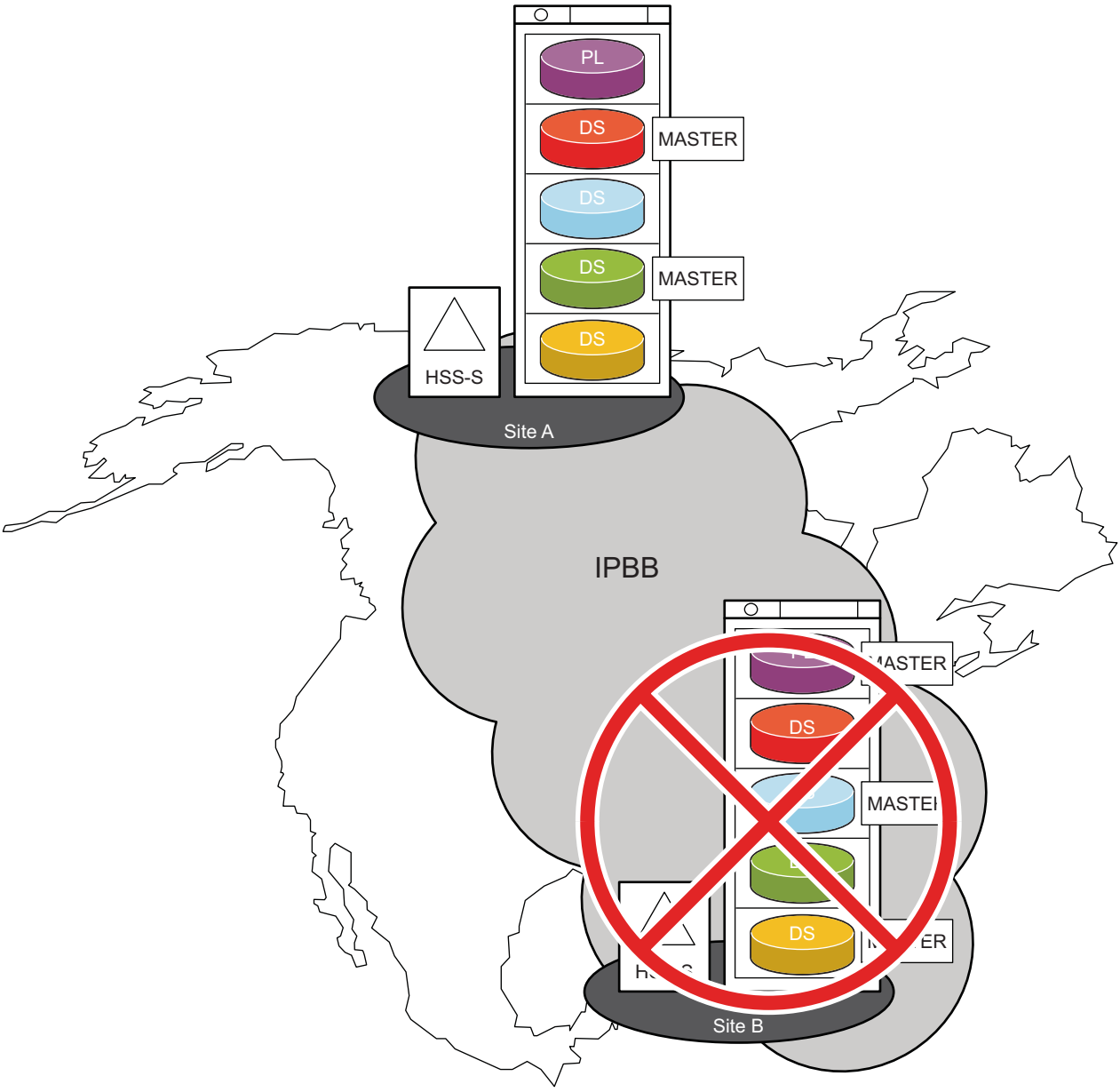


Figure 32 Symmetrical Split Situation with Two Nodes (One Node is Down)

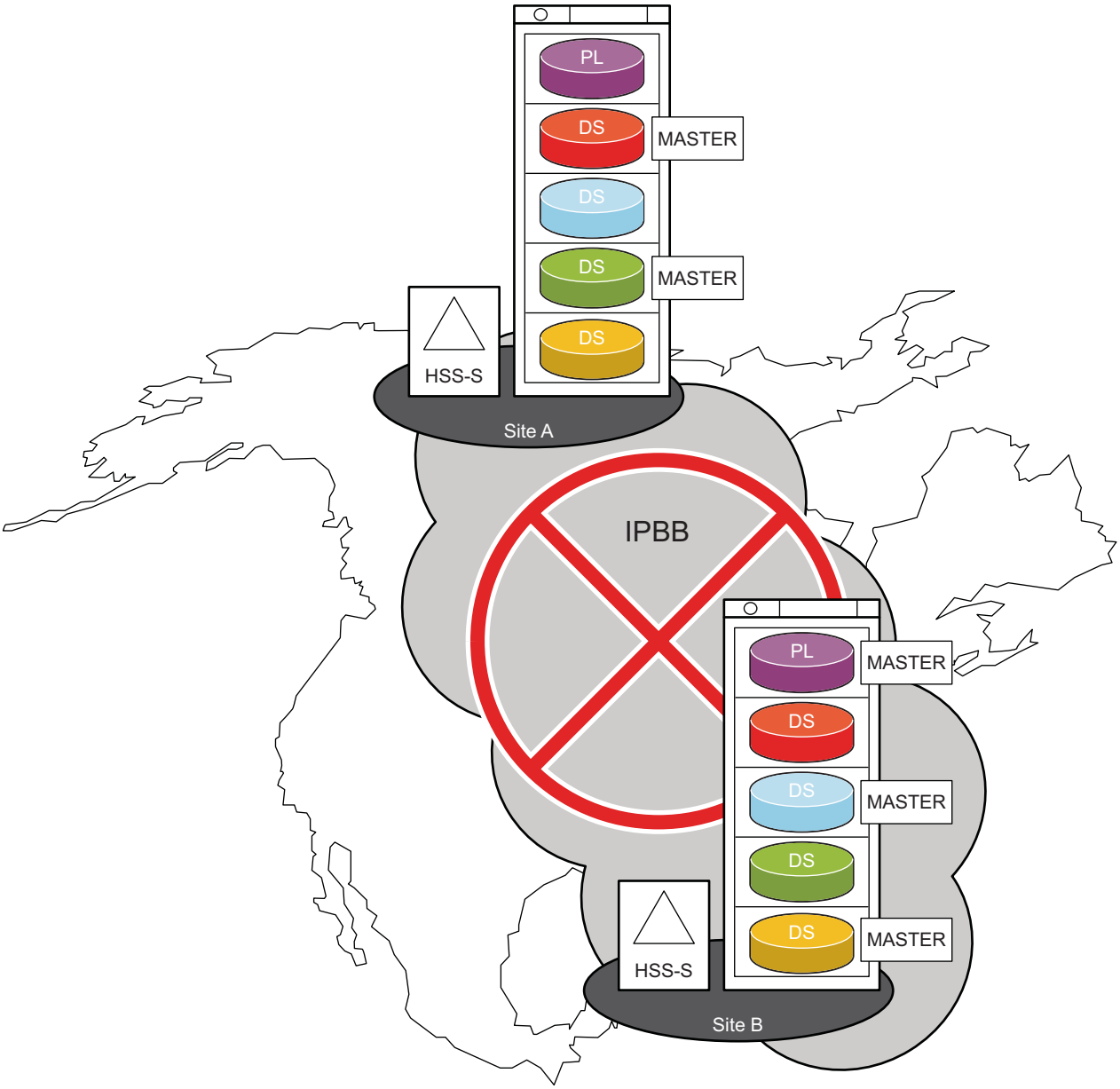


Figure 33 Symmetrical Split Situation with Two Nodes (IP Backbone Communication Failure)

In case of such split situations, the `Control, Potential Split Brain Detected` alarm is raised.

In a symmetrical split, the CUDB system is split in two equal halves regarding site visibility. In this scenario, the SM partition leader assumes that the CUDB sites in the unreachable part of the system are down, and then take the mastership of all the DSGs with an available slave DS Unit, and whose master DS Units were previously hosted in the unreachable part of the system.

If the two halves of the system are actually alive but can not connect to each other due to a network failure, both parts of the system take mastership independently. As a consequence of this situation, a master DS Unit for the same DSG can be reachable in each isolated partition of the system.

The PLDB mastership is not reassigned, and provisioning operations (`create`, `update` and `delete`) performed by a provisioning user are only allowed in the partition hosting the Master PLDB Replica, in order to minimize potential provisioning data inconsistencies.

After the symmetrical split situation is over, provisioning operations are unlocked for the partitions not hosting the Master PLDB Replica. Refer to *CUDB LDAP Data Access*, Reference [11] and *CUDB LDAP Interwork Description*, Reference [12] for more information on provisioning operations.

3.3.5.2 Recovery Procedures after System Split

This section provides information about the possible split situations in the CUDB system on different deployments, and the recovery procedures followed by the system. Deployments are characterized by the number of sites and the type of geographical redundancy configured. These features define the system behavior in a system split situation. Refer to *CUDB Deployment Guide*, Reference [4] for more information on the various deployment types.

The probability of a site becoming unreachable due to the hard failure of all nodes is inversely proportional to the number of nodes in the site, and the number of BC servers in the BC cluster of the site. Similarly, the probability of a whole site going down is smaller as the number of nodes in the site increases.

If more than one node is located on a site, the following scenarios apply in case of a hard site failure:

- **Recovery of at least one node within the site**

  The site is up again if the BC cluster is recovered through node recovery (that is, more than half of the BC servers in the BC cluster are recovered), indicating that the symmetrical split or majority/minority split situation is over or changed.

- **Subsequent node recoveries after the first one**

  Subsequent recoveries are handled as normal node management operations, specified at the list of recovery methods.

In case of network isolation, the number of nodes per site is not significant: the whole site is isolated.

The possible main scenarios are as follows:

Majority/Minority (Asymmetrical) Situations

The behavior of the site(s) in such cases is as follows:

- The site(s) in minority because of network isolation or node failure release their PLDB and DS masterships. The only exceptions are the self-contained partitions, which are served normally in minority partitions.

- The site(s) in majority take the mastership of the all DSGs (both for the currently owned DSGs, and also for the ones released by the minority partitions). The PLDB mastership is reassigned if the mastership was originally in the minority group. Traffic and provisioning are not affected.

The behavior of the site(s) during recovery is as follows:

- No master reassignation is performed.

- All the database clusters (PLDB and DS) in the recovering site(s) become slaves, and start replication from already running masters in the surviving sites (except for the self-contained partitions which did not suffer from mastership changes).

Replication lag is expected. When replication is restarted, the replication lag is gradually reduced by automatic cluster replication, which can fail due to several reasons. Then, a previous backup and restore operation is required to catch the slaves up with the running masters and to restart automatic replication. Refer to *Control, Potential Split Brain Detected*, Reference [13] for more information about manual recovery.

Symmetrical Split Situation

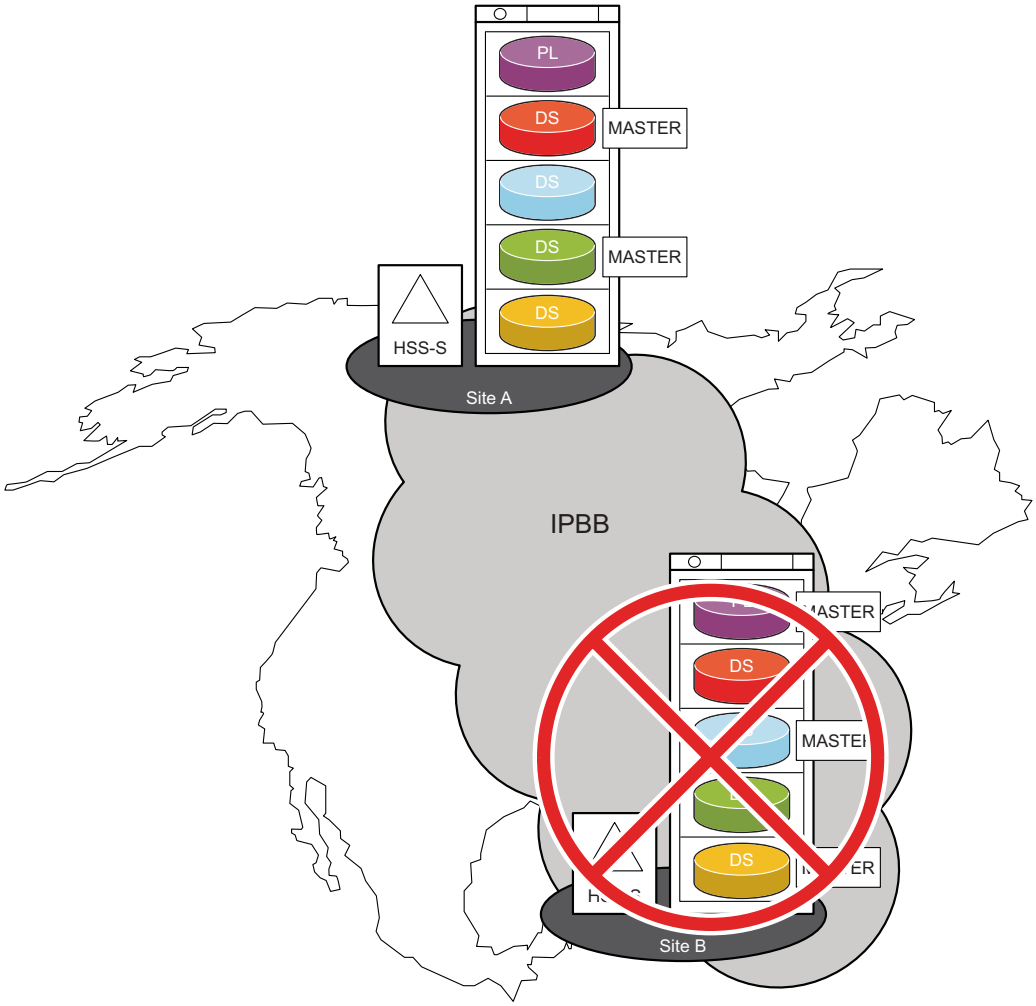
In case of deployments with two sites, only symmetrical split situations can occur.

Note: The explanation of symmetrical split situation demonstrates a system with two sites, which is the deployment with the highest probability to get into symmetrical split situation.

The recovery procedure during a symmetrical split situation is the following:

Both split sides take over DS mastership. The PLDB mastership is not reassigned, and the split side not hosting the PLDB master replica locks write provisioning to simplify reconciliation later. Further recovery procedures depend on whether the symmetrical split situation occurred due to a site failure or due to a site communication failure.

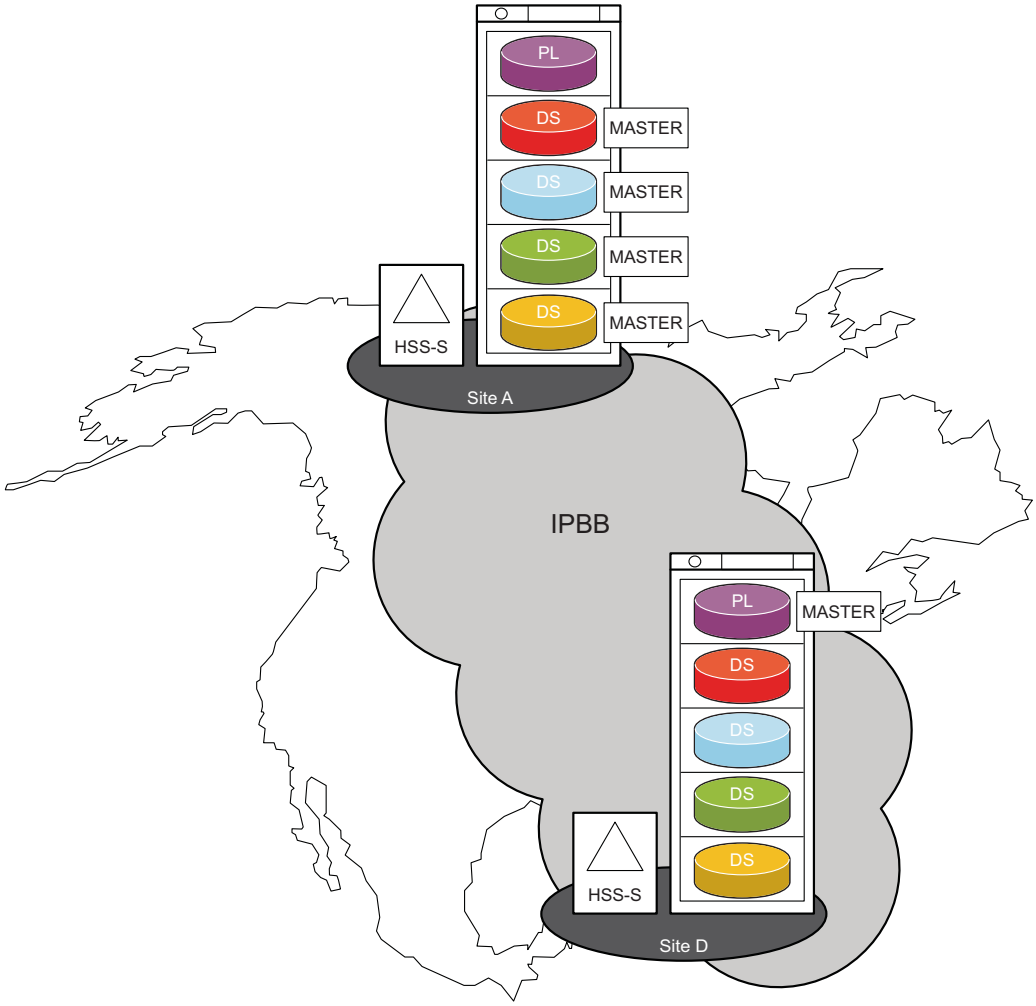
Site failure: See the figure below for the illustration of site failure.



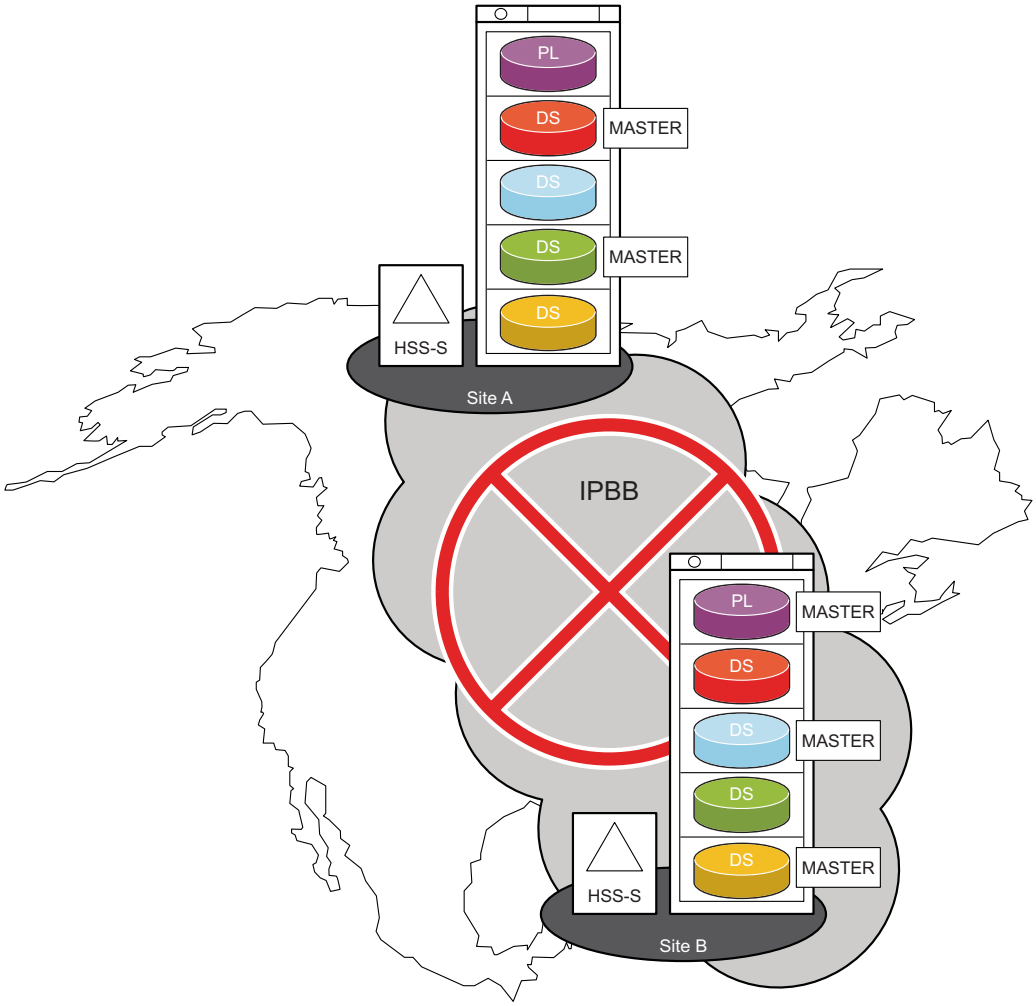
The behavior is the same for all the node recoveries in the site:

- The site is recovered and considered visible. The split situation is over, and the `Control, Potential Split Brain Detected` alarm is ceased.
- PLDB mastership remains where it originally was. If the local PLDB was master, then mastership continues there after recovery. No replication lags are expected, as provisioning was locked on the surviving side.
- At the same time, the mastership for the DSGs has been lost in the recovering nodes. During one of the node recoveries, all former master DSs become slaves. Replication lag is expected due to the new masters.

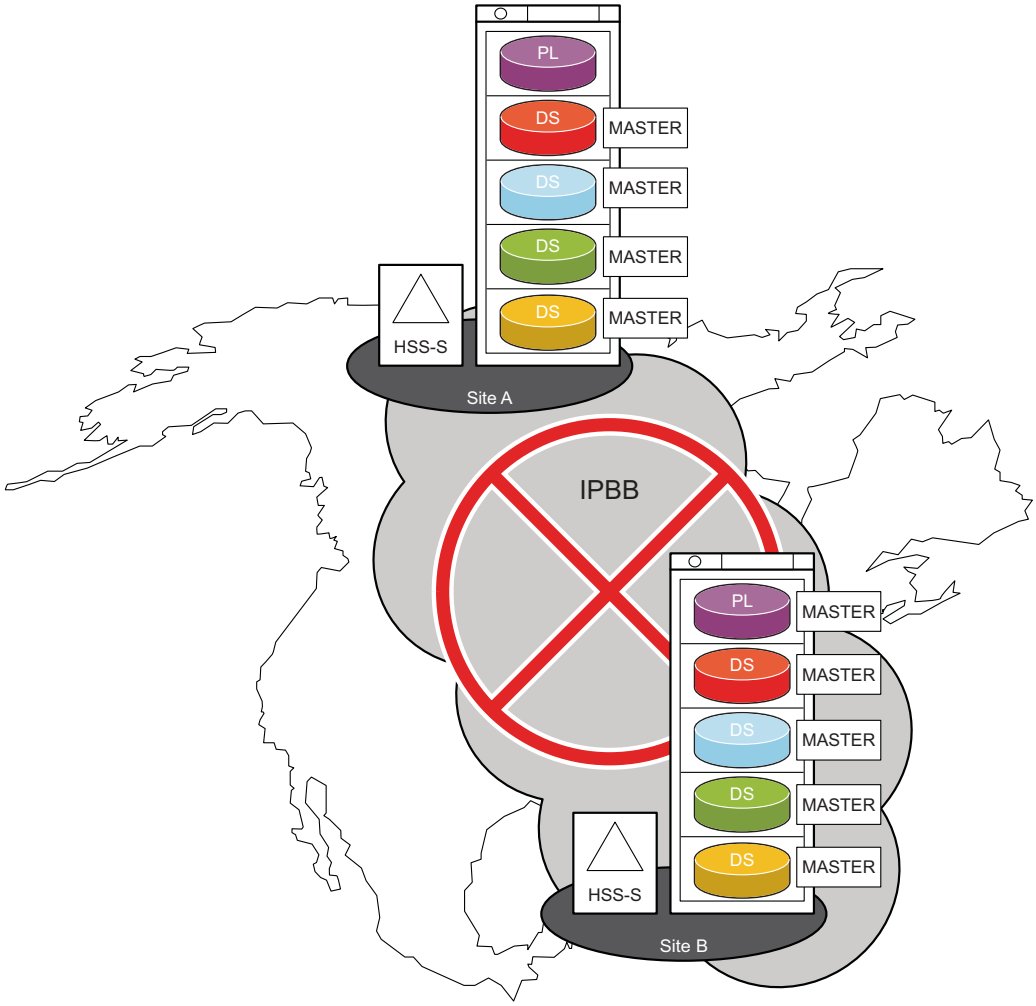
The final system set-up after the recovery is shown in the picture below:



Site communication failure means that the nodes are unable to communicate with each other. The failure is shown with the picture below.



During the failure, both sites in the two isolated partitions apply service continuity or split management actions resulting in double mastership for each DSG. The PLDB mastership is not modified and keeps hosted in the same site. This is shown in the picture below.

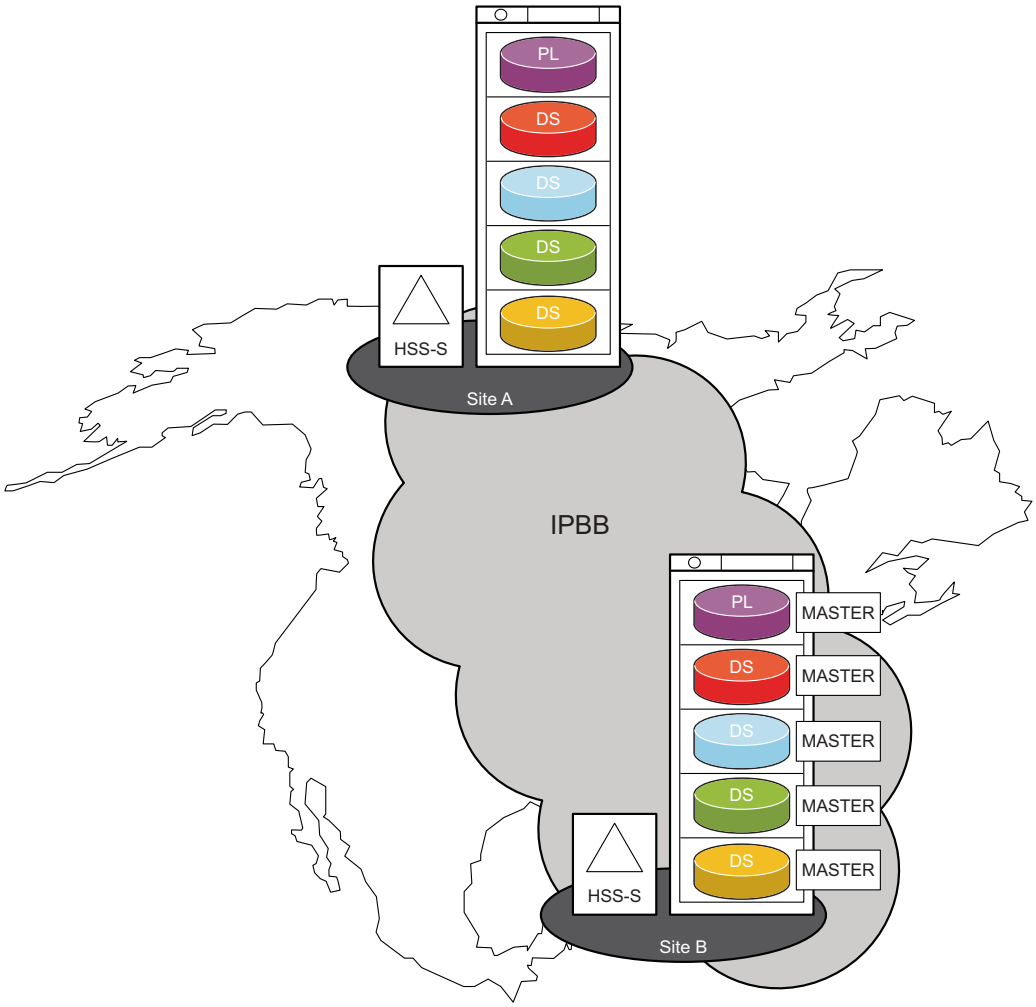


Node recovery in this case happens as follows:

- On communication recovery, the split situation is over. The `Control, Potential Split Brain Detected` alarm is ceased.
- In case of PLDB, no replication lag is expected, as provisioning was stopped.
- The mastership for the DSGs was duplicated, therefore a new unique master is elected with the master election algorithm (see Section 3.3.6 on page 74 for more information).
- Once the new master is elected for each DSG, the other replicas become slaves. As mastering was duplicated, both clusters received updates and did not replicate. Therefore, no replication lag is expected, but the databases are different, and consequently the slave replicas cannot automatically get in sync with their master replicas.

In case the Automatic Handling of Network Isolation function is enabled, the diverging database versions are merged together into the new master. After that, if the Self-Ordered Backup and Restore function is enabled, the slave replicas are automatically restored from a backup taken from the new master. Otherwise, they must be resynchronized manually. Refer to *CUDB Data Storage Handling*, Reference [39] for more information.

The final system set-up after the recovery is shown in the picture below:



3.3.5.3 Service Continuity for Asymmetrical Split Scenarios

In case of site or node failures, or due to connectivity problems in the transport network, the CUDB system topology can be dynamically reconfigured, aiming to provide service from a majority partition. In case failures lead to asymmetrical split situations, the CUDB system tries to maximize service by placing master replicas on the majority partition, and disabling traffic services from the nodes

residing in the minority partition, which return an error code that signals clients to try switch-over to the majority side of the system.

Catering service from the biggest partition is usually the best option. However, it can occur that the default behavior of the system must be corrected to enable service also on minority partitions. The CUDB system provides specific commands to allow a partition in minority to accept and process traffic requests (or provisioning) in case one of the following scenarios occur:

- Provide service from a network-isolated minority site (even if another network partition in majority already exists). For example, in a three-site CUDB system deployment, when one of the sites is isolated from the rest of the system, the isolated site enters minority state and stops processing traffic, rejecting the incoming requests. The execution of the `cudbServiceContinuity` command in any of the nodes of the isolated (and in-minority) site will force the site to take up DSG masters, allowing the nodes in the minority partition to continue processing traffic. The PLDB master is not taken in the minority partition: therefore, provisioning traffic is still not allowed. The service continuity mode is automatically canceled when the minority partition regains connectivity to the rest of the system. If the `automaticServiceContinuity` parameter (refer to *CUDB Node Configuration Data Model Description*, Reference [3]) is set in the nodes of the isolated (and in-minority) site, that site will take up DSG masters automatically.
- Provide service when there is only one site surviving and is in a minority partition. In a situation where there is only one site surviving, in a two- or three-site CUDB system deployment, there is an option to enable traffic and also provisioning for all the DSGs available on the surviving site.

  Warning!

  Make sure that the rest of the sites are indeed out of service, and that the issue is not caused by communication problems.

  The `cudbTakeAllMasters` command can be used in any of the nodes on the surviving site, which forces the site to elect both a PLDB master and all the DSG masters included in the surviving site.

  Note: The command can be executed only if the surviving site is in Minority. Also, consider that the recovery of the entire system from this situation is not automatic. Refer to *CUDB System Split Partial Recovery Procedure*, Reference [14] for more information about this recovery process.

3.3.5.4 Replication Lag

In many cases, the replication lag is automatically caught up by the Cluster Geographical Replication mechanism. However, manual resynchronization is

necessary if the nodes that contain the slave partitions raise the *Storage Engine, Unable to Synchronize Cluster in PLDB, Major*, Reference [15] and *Storage Engine, Unable to Synchronize Cluster in DS, Major*, Reference [16] alarms.

In these cases, the slave partitions must be resynchronized manually from the master partitions, and backup and restore is necessary. Refer to *Control, Potential Split Brain Detected*, Reference [13] for more information on manual synchronization.

3.3.6 Master Election Algorithm

Master PLDBs or DSs must be selected in four cases:

- The current master of any PLDB or DSG is going down. In this case, the Master Election Algorithm promotes one of the slave replicas to master. The selected slave replica is the one whose replication update state is the closest to the master (that is, the one which has the smallest replication lag).

  In case the replication lag in both replicas is the same, the one with the higher priority is selected. Refer to *CUDB Node Configuration Data Model Description*, Reference [3] for more information on the priority parameter and replica priorities.

- A majority system split situation occurred. If the majority group in a system split situation contains two slave replicas for a DSG, a new master replica must be elected for the majority. The selection criteria to follow is the same as above.

- Recovery after a dual mastership situation occurred. During symmetrical split situations, two or more masters exist, one in each of the isolated systems. During recovery, the system is unified and the system must elect a unique master from one of those masters.

  In this case, the Master Election Algorithm keeps the DSG masters in the site which hosted the PLDB Master replica during the symmetrical split situation.

- There is not any available PLDB replica in a CUDB site. If the CUDB site contains any CUDB node without PLDB, new master replica must be elected for all master DSs hosted by these CUDB nodes.

3.3.7 Manual DS Master Change

In some cases, a DS mastership change might be required. Refer to *CUDB System Administrator Guide*, Reference [7] for more information.

4 Operation and Maintenance

This section provides information on operating and maintaining the CUDB high availability functions and services.

4.1 Configuration

This section provides information on configuring the various CUDB high availability processes and services, such as geographical redundancy, site identifiers and provisioning conditions.

4.1.1 Geographical Redundancy Configuration

Information related to geographical redundancy is stated in the `CudbDsGroup`, `CudbLocalDS` and `CudbRemoteDs` configuration classes. Refer to *CUDB Node Configuration Data Model Description*, Reference [3] for more information on these parameters.

Replication is automatically handled by the CUDB, and requires no administrative actions other than managing raised alarms. No action needed to fix replication issues other than following the procedures described in *CUDB Node Fault Management Configuration Guide*, Reference [17].

4.1.2 Geographical Redundancy Upgrade

This procedure can be performed by Ericsson personnel only. Contact the next level of maintenance support in case the system needs to be expanded.

4.1.3 Setting Site Identifier for a CUDB Node

Site identifiers are configured with the `siteId` attribute in the `CudbLocalNode` class, indicating the site where the local node is located.

Refer to *CUDB Node Configuration Data Model Description*, Reference [3] for more information about these classes.

4.1.4 Provisioning Condition for an LDAP User

Provisioning conditions are configured with the `isProvisioningUser` attribute in the `CudbLdapUser` class, indicating if the LDAP user is a provisioning user.

Refer to *CUDB LDAP Interwork Description*, Reference [12] for further information.

4.2 Fault Management

Refer to the following alarm documents for more information on the alarms that can occur.

- *Storage Engine, DS Cluster Down*, Reference [18].
- *Storage Engine, DS Cluster Node Down*, Reference [19].
- *Storage Engine, DS Cluster in Maintenance Mode*, Reference [20].
- *Storage Engine, PLDB Cluster Down*, Reference [21].
- *Storage Engine, PLDB Cluster Node Down*, Reference [22].
- *Storage Engine, PLDB Cluster in Maintenance Mode*, Reference [23].
- *Storage Engine, No Available Master Replica for DS*, Reference [24].
- *Storage Engine, No Available Master Replica for PLDB*, Reference [25].
- *Control, Potential Split Brain Detected*, Reference [13].
- *Control, Remote Node Unreachable*, Reference [26].
- *Control, Remote Site Unreachable*, Reference [27].
- *Control, Blackboard Coordination Server Down*, Reference [28].
- *Control, Blackboard Coordination Cluster Down*, Reference [29].
- *Storage Engine, Replication Channels Down in DS*, Reference [30].
- *Storage Engine, Replication Channels Down in PLDB*, Reference [31].
- *Storage Engine, Unable to Synchronize Cluster in DS, Major*, Reference [16].
- *Storage Engine, Unable to Synchronize Cluster in PLDB, Major*, Reference [15].
- *Storage Engine, High Load in DS*, Reference [32].
- *LDAP Front End, Server Down*, Reference [34].
- *LDAP Front End, Processing Redundancy Lost*, Reference [35].
- *LDAP Front End, Processing Capacity Below Minimum*, Reference [36].

Refer to *CUDB Node Fault Management Configuration Guide*, Reference [17] for further information about alarms.

In addition to the faults listed above, two additional fault management scenarios exist. These scenarios are listed below.

4.2.1 Management of Geographical Redundancy

In some cases, the replication processes between clusters could fail. Refer to *CUDB Troubleshooting Guide*, Reference [37] for further information on troubleshooting the issue.

4.3 Performance Management

This section is not applicable to this feature.

4.4 Security

This section describes the security measures applicable to the CUDB High Availability feature. In case of HA, security issues mostly affect the Geographical Redundancy feature.

The CUDB system allows to configure secure database replication traffic between the master and slave replication servers. This feature is optional, and can be configured by parameter settings. The secure communication is established by using SSL/TLS (Secure Socket Layer/Transport Layer Security) certificates previously issued and signed by Certification Authority (CA).

4.5 Logging

Table 2 lists the logged messages related to the High Availability feature.

Table 2 LDAP FE Logs

Severity	Message Information	Trigger Event
Error	`(error) - LDAP FE is unreachable`	The LDAP FE Monitor detected that LDAP FE is not reachable through LDAP.
Error	`(error) - LDAP server responded with an error: <error>`	The LDAP FE responded to the LDAP FE Monitor with the specified error.
Error	`(error) - ndbd process restart for <store_id> in host <host_ip> not completed.`	A database cluster process has stopped, and the Cluster Supervising function attempts to restart it.

Severity	Message Information	Trigger Event
Error	(error) - Slave mysqld server belonging to store *<store_id>* at address *<server_ip>* seems to be dead.	A slave database server became unavailable.
Info	(info) - LDAP FE process has been started	The LDAP FE Monitor started the LDAP FE process.
Info	(info) - LDAP FE process has been terminated	The LDAP FE Monitor successfully terminated the LDAP FE process with a TERM signal.
Info	(info) - LDAP FE process has been killed	The LDAP FE Monitor terminated the LDAP FE process with a KILL signal.
Warning	(warning) - Provisioning locked due to an split brain situation.	Provisioning is locked due to symmetrical split.
Warning	(warning) - Provisioning unlocked.	Provisioning resumed.
Warning	(warning) - Site *<site_id>* set as auto-removed for node *<node_id>*.	An unreachable site has been marked as auto-removed by the majority group of nodes.
Warning	(warning) - Site *<site_id>* set as non auto-removed for node *<node_id>*.	An auto-removed site has been rediscovered by the majority group of nodes.
Warning	(warning) - *<monitor_thread>*: LDAP FE at *<ldapfe>*:*<ldapfe_port>* is now down.	An LDAP FE process has failed.
Warning	(warning) - SystemMonitor is NOT running. KeepAlive will start it.	The System Monitor process has stopped.
Warning	(warning) - LDAP FE process is not running	The LDAP FE Monitor detected that LDAP FE is not running.
Warning	(warning) - LDAP FE did not shut down in time	The LDAP FE Monitor could not terminate the LDAP FE process with a TERM signal.

Glossary

For the terms, definitions, acronyms and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [38].

Reference List

CUDB Documents

[1] *CUDB Technical Product Description*

[2] *CUDB Node Network Description*

[3] *CUDB Node Configuration Data Model Description*

[4] *CUDB Deployment Guide*

[5] *CUDB Data Distribution*

[6] *CUDB Notifications*

[7] *CUDB System Administrator Guide*

[8] *CUDB Security and Privacy Management*

[9] *Storage Engine, Replication Delay Too High In DS*

[10] *Storage Engine, Replication Delay Too High In PLDB*

[11] *CUDB LDAP Data Access*

[12] *CUDB LDAP Interwork Description*

[13] *Control, Potential Split Brain Detected*

[14] *CUDB System Split Partial Recovery Procedure*

[15] *Storage Engine, Unable to Synchronize Cluster in PLDB, Major*

[16] *Storage Engine, Unable to Synchronize Cluster in DS, Major*

[17] *CUDB Node Fault Management Configuration Guide*

[18] *Storage Engine, DS Cluster Down*

[19] *Storage Engine, DS Cluster Node Down*

[20] *Storage Engine, DS Cluster in Maintenance Mode*

[21] *Storage Engine, PLDB Cluster Down*

[22] *Storage Engine, PLDB Cluster Node Down*

[23] *Storage Engine, PLDB Cluster in Maintenance Mode*

[24] *Storage Engine, No Available Master Replica for DS*

[25] *Storage Engine, No Available Master Replica For PLDB*

[26] *Control, Remote Node Unreachable*

[27] *Control, Remote Site Unreachable*

[28] *Control, Blackboard Coordination Server Down*

[29] *Control, Blackboard Coordination Cluster Down*

[30] *Storage Engine, Replication Channels Down in DS*

[31] *Storage Engine, Replication Channels Down in PLDB*

[32] *Storage Engine, High Load in DS*

[33] *Server Platform, Storage Performance Degradation Detected*

[34] *LDAP Front End, Server Down*

[35] *LDAP Front End, Processing Redundancy Lost*

[36] *LDAP Front End, Processing Capacity Below Minimum*

[37] *CUDB Troubleshooting Guide*

[38] *CUDB Glossary of Terms and Acronyms*

[39] *CUDB Data Storage Handling*

Other Documents and Online References

[40] *Bidirectional Forwarding Detection. IETF RFC 5880* *http://www.rfc-editor.org/rfc/rfc5880.txt*

[41] *SAF AIS* *http://www.saforum.org/Service-Availability-Forum:-Application-Interface-Specification~217404~16627.htm*