Data Collection Guideline for CUDB

DOCUMENT LIST

3/1543-HDA 104 03/10-V1 Uen A

Contents

1 Introduction

The purpose of this document is to instruct what troubleshooting data is to be collected and enclosed in a Customer Service Request (CSR) in case a problem is experienced with the Ericsson Centralized User Database (CUDB).

1.1 Revision Information

Rev. A

This document is based on 3/1543-HDA 104 03/9 with the following changes:

- Terminology and hardware information updates throughout the document.
- Section 3.1.4 on page 11: Updated command in Step 4.

1.2 Related Information

Definition and explanation of acronyms and terminology, trademark information, and typographic conventions can be found in the following documents:

- *CUDB Glossary of Terms and Acronyms*, Reference [1]
- *Trademark Information*, Reference [2]
- *Typographic Conventions*, Reference [3]

1.3 Prerequisites

This section describes the prerequisites for performing the data collection procedure.

1.3.1 Documents

Before starting this procedure, ensure that the following information or documents are available:

- This document.
- *CUDB Troubleshooting Guide*, Reference [4] and *CUDB Node Logging Events*, Reference [5].
- All the documents referenced in the procedures below and listed in the References section.

1.3.2 Conditions

Before starting the data collection procedure, ensure that the following conditions are met:

- The CUDB, database cluster, and BSP 8100 usernames and passwords are available.
- Log in to the CUDB node Command Line Interface (CLI). Refer to the *CUDB CLI* section of *CUDB System Administrator Guide*, Reference [6] for information.
- Log in to the blade or Virtual Machine (VM) of the CUDB node (see Section 1.3.2.2 on page 2 for more information).
- Obtain the local and global Data Store (DS) and DS Unit Group (DSG) ID allocation (see Section 1.3.2.3 on page 3 for more information).
- Obtain the global DSG ID for channel replication (see Section 1.3.2.4 on page 4 for more information).
- In case the CUDB system is deployed on native BSP 8100 hardware, refer to the "Ericsson Command-Line Interface User Guide" document in the BSP 8100 CPI for more information.

The below sections describe how to establish the connections and fetch the required information listed above.

1.3.2.1 Naming Conventions

Several instructions in this document require the execution of a command manually. In such cases, the CUDB prompt is indicated as follows:

```
CUDB<node_id> SC_2_x#
```

In the above output, `<node_id>` represents the ID of the node to update, while `SC_2_<x>` represents the active System Controller (SC).

Note: Use the `cudbHaState` command as shown below to check the active controller for COM:

```
CUDB<node_id> SC_2_<x># cudbHaState | grep COM | grep ACTIVE
```

The output of the command must look similar to the below example:

```
CUDB<node_id> SC_2_<x># cudbHaState | grep COM | grep
ACTIVE
COM is assigned as ACTIVE in controller SC-2
```

1.3.2.2 Connecting to the Blade or Virtual Machine of the CUDB Node

In case the CUDB system is deployed on native BSP 8100 hardware, connection to the GEP3 or GEP5 boards is available either through serial connection, or through the CUDB CLI. In case the CUDB is deployed on a virtual infrastructure, connection to the VMs is available only through CUDB CLI.

Connecting from an External Server with Serial Connection

Perform the following procedure to log in to a GEP3 or GEP5 board of the CUDB node through serial connection:

1. Take note of the following default serial port parameters:
   - Baud rate: 115200.
   - Data bits: 8.
   - Parity: None.
   - Stop bits: 1.
   - Flow control: None.
2. Log in to the GEP3 or GEP5 board. After a successful login, the prompt of the CUDB node is displayed.

Connecting through the CUDB CLI

Perform the following procedure to log in to the blade or VM of the CUDB node through the CUDB CLI:

1. Log in to the CUDB CLI as described in the *CUDB CLI* section of *CUDB System Administrator Guide*, Reference [6].
2. Establish an SSH connection to the blade or VM as follows:
   - Use **`ssh PL_2_<x>`** to connect to the payload blade or VM.
   - Use **`ssh DS<x>_<x>`** to connect to the DS.
   - Use **`ssh PL<x>`** to connect to the PLDB.

1.3.2.3 Obtaining Local and Global DS and DSG ID Allocation

Perform the following steps to obtain the local and global DS-DSG ID allocation:

1. Log in to the node for which the local and global mapping is determined.
2. Execute the following command:

   CUDB*<node_id>* SC_2_*x*# **`cudbManageStore -a -o status`**

The output of the command must be as follows:

```
CUDB<node_id> SC_2_x# cudbManageStore -a -o status
cudbManageStore stores to process:  pl ds1 (in
dsgroup1)
ds2 (in dsgroup2) ds3 (in dsgroup3) ds4 (in dsgroup4)
ds5 (in dsgroup5) ds6 (in dsgroup6) ds7 (in dsgroup7)
ds8 (in dsgroup8).
Store pl in dsgroup 0 is alive and reporting status
ACTIVE(2).
Store ds1 in dsgroup 1 is alive and reporting status
ACTIVE(2).
Store ds2 in dsgroup 2 is alive and reporting status
ACTIVE(2).
Store ds3 in dsgroup 3 is alive and reporting status
ACTIVE(2).
Store ds4 in dsgroup 4 is alive and reporting status
ACTIVE(2).
Store ds5 in dsgroup 5 is alive and reporting status
ACTIVE(2).
Store ds6 in dsgroup 6 is alive and reporting status
ACTIVE(2).
Store ds7 in dsgroup 7 is alive and reporting status
ACTIVE(2).
Store ds8 in dsgroup 8 is alive and reporting status
ACTIVE(2).
cudbManageStore command successful.
```

In this specific case, the local DS 1 belongs to the global DSG 1.

Refer to *CUDB Subscription Reallocation*, Reference [7] for more information on DSG ID allocation.

1.3.2.4 Obtaining Global DSG ID for Channel Replication

Execute the `cudbSystemStatus` command from one of the SCs as follows to obtain the global DSG IDs:

```
CUDB<node_id> SC_2_x# cudbSystemStatus -r
```

The output of the command must be similar to the below example:

```
CUDB<node_id> SC_2_x# cudbSystemStatus -r

Execution date:  Tue Sep 4 22:20:01 CEST 2012

Checking Replication Channel in the System
Node 109 :
DSG0 is Master
Replication in DSG1(Node=109--Chan=1)....  OK -- Delay = 0
Replication in DSG2(Node=109--Chan=1)....  OK -- Delay = 0
```

```
Replication in DSG3(Node=109--Chan=1)....  OK -- Delay = 0
Replication in DSG4(Node=109--Chan=1)....  OK -- Delay = 0
Replication in DSG5(Node=109--Chan=2)....  OK -- Delay = 0
Replication in DSG6(Node=109--Chan=2)....  OK -- Delay = 0
Replication in DSG7(Node=109--Chan=1)....  OK -- Delay = 0
[-E-] Replication in DSG8(Node=109)....  NOK
Node 110 :
Replication in DSG0(Node=110--Chan=1)....  OK -- Delay = 0
DSG1 is Master
DSG2 is Master
DSG3 is Master
DSG4 is Master
DSG5 is Master
DSG6 is Master
DSG7 is Master
DSG8 is Master
```

In this specific case, the global DSG ID 8 has problems with the replication in node 109.

2 Workflow

The workflow for collecting troubleshooting data is as follows:

1. Collect mandatory data that is needed in connection with any problems experienced. See Section 3 on page 9 for more information on collecting mandatory data.

2. Collect other useful information if available within an acceptable amount of time and effort.

3 Mandatory Data

This section describes how to collect data that is mandatory for every type of problems related to CUDB.

The data described in this section must always be included in a CSR.

3.1 Data Collection

To collect the required data, perform the following procedure:

1. Collect all application and platform logs with the `cudbCollectInfo` command.
2. Make a copy of the log archive directory for later use. See Section 3.1.2 on page 11 for more information.
3. Collect the SAF logs with the `cmw-collect-info` command.
4. Collect additional core dump files. See Section 3.1.4 on page 11 for more information.
5. Collect additional logs required for the CSR. See Section 3.1.5 on page 12 for more information.
6. Collect all recent activities that can have an impact on the behavior of the CUDB system. See Section 3.1.6 on page 12 for more information.
7. Collect logs using the `cudbGetLogs` script. Refer to *CUDB Logchecker*, Reference [8] for more information.

The exact procedures to perform for the above points are described in the following sections in more detail.

3.1.1 Collecting Application and Platform Logs

The `cudbCollectInfo` script collects logs from the whole CUDB system, including the following:

- `ddci`: Consistency Check logs.
- `ndb_error_reporter`: Network database cluster logs.
- `collect_info`: Core MW and system log files for each blade or VM from the active SC.
- `sm`: System Monitor (SM) logs.

- `esa`: Ericsson SNMP Agent (ESA) logs.
- `cluster_config`: Cluster configuration log.
- `measurement`: CUDB performance and counter measurements.
- `ldapfe`: Lightweight Directory Access Protocol (LDAP) Front End (FE) logs.
- `system_status`: System status logs.
- `bc_server`: Blackboard Coordination Server (BC Server) logs.
- `evip`: Evolved Virtual IP (eVIP) logs.
- `bc_client`: Data stored in the BC cluster.
- `upgrade`: Logs stored in the `cudbUpgradeWorkDir` directory.

Perform the following steps to collect the above logs:

1. Log in to the CLI of the CUDB node as described in the *CUDB CLI* section of *CUDB System Administrator Guide*, Reference [6].
2. Execute `cudbCollectInfo` as follows:

   `CUDB<node_id> SC_2_x#` **`cudbCollectInfo`**

   Note: The execution of `cudbCollectInfo` takes approximately 5-20 minutes.
3. Copy the resulting file from the CUDB node. The file must be located in the following folder:

   `/local/tmp/cudb_collect_info_<date>-<time>.c`

The output of the command must be similar to the below example:

```
CUDB143 SC_2_1# cudbCollectInfo

Creating /local/tmp/cudb_collect_info_20130711-171410/143
...  OK
Creating /local/tmp/cudb_collect_info_20130711-171410/144
...  OK
Creating /local/tmp/cudb_collect_info_20130711-171410/145
...  OK

Waiting 143 to finish ...  OK
Waiting 144 to finish ...  OK
Waiting 145 to finish ...  OK

CREATING ARCHIVE ...  OK
Encrypting archive ...  OK
```

```
Removing unencrypted archive ...  OK

REMOVING RAW DATA ...  OK
```

Note: The output file has a `.c` extension if the default encryption is enabled. If the default encryption is disabled, the file extension is `.tar.gz` instead. The GNU Privacy Guard (GPG) encryption of the `tar.gz` file can be disabled by executing the `cudbCollectInfo -e` command.

The size of log files is generally between 100-900 MB. For example, in a three-node system with 8 DSGs, the size of the collected information is approximately 600 MB, which includes all collected logs from all three nodes.

For further information about the use of the command, execute it with the help switch (`cudbCollectInfo -h`), or refer to *CUDB Node Commands and Parameters*, Reference [9].

3.1.2 Copying Log Archive Files

Copy the log archive files to an external repository (for example, with the `scp` command), then send the relevant ones to support if requested. The log archive files are located on the SC in the following directory:

```
/local/cudb_logarchive/
```

3.1.3 Collecting SAF Logs

Perform the following steps to collect the SAF logs:

1. Log in to the CLI of the CUDB node as described in the *CUDB CLI* section of *CUDB System Administrator Guide*, Reference [6].

2. Execute the `cmw-collect-info` command as follows:

   CUDB<*node_id*> SC_2_*x*# **cmw-collect-info <*filename*>**

Note: To extract all logs related to the platform, send the command from an SC to generate a compressed file in the current directory.

3.1.4 Collecting Additional Core Dump Files

The `cudbCollectInfo` command lists the core dumps only. Perform the following steps to collect additional dump files.

1. Log in to the CLI of the CUDB node as described in the *CUDB CLI* section of *CUDB System Administrator Guide*, Reference [6].

2. List the core dump files of the `/cluster/dumps` directory (where SC and PL_2_5 write them) and the `/local2/dumps` directory (for the rest of the payload blades or VMs) with the following commands:

```
CUDB<node_id> SC_2_x# ls -lat /cluster/dumps/

CUDB<node_id> PL_2_x# ls -lat /local2/dumps/
```

Note: PL_2_x stands for all payload blades or VMs except PL_2_5.

The command output must be similar to the example below:

```
CUDB45 SC_2_1# ls -lat /cluster/dumps/
total 16672
drwxr-xr-x 27 root root 4096 Oct 24 15:57 ..
drwxr-xr-x 2 root root 8192 Oct 5 16:54 .
-rw------- 1 root root 11407360 Oct 5 16:54\
ncs_scap.14942.SC_2_2.core
-rw------- 1 root root 180662272 Oct 3 17:56
cudb_LdapFeMoni.15701.SC_2_2.core
-rw------- 1 root root 130125824 Oct 3 17:56\
cudb_LdapFeMoni.8446.SC_2_2.core
```

3. List the slapd gdb files of the /cluster/dumps directory (for any blade or VM) with the following command:

   ```
   CUDB<node_id> SC_2_x# ls -lat /cluster/dumps/ | grep
   gdb_slapd
   ```

4. Copy the core dump files and slapd gdb files whose timestamps are dated around the incident. Use the following command to do so (an external server is recommended for copying):

   ```
   SC_2_1# scp -l <bandwidthlimit_in_kbps> /path/to/file
   username@remote_host:/path/to/file
   ```

   The recommended value for *<bandwidthlimit_in_kbps>* is 10000.

3.1.5 Collecting Additional Logs

Use the following commands to collect additional logs:

```
tar -czvf /cluster/home/commandlog.tar.gz /var/log/*/comm
andlog* /var/log/*/kernel*

tar -czvf /cluster/home/prmaint.tar.gz /home/cudb/monitor
ing/preventiveMaintenance/

cluster alarm -a -l --full
```

3.1.6 Describing Recent Activities

The problematic behavior of the CUDB system may be caused by certain activities that occurred (or were executed) in the system days or weeks before the problem occurred.

Therefore, check if any of the activities listed below took place during the last 2-4 weeks before the problem appeared. If so, then try to obtain as much information, logs, and documents as possible to facilitate troubleshooting.

Table 1 shows the activities that must be taken into consideration when collecting additional information, if applicable.

Table 1 Recent Activities

Activity	CUDB Nodes / CUDB System Level	UDC Solution Level (FEs Connected to the CUDB System, Provisioning System, Other Nodes Connected)	IP Network (Site Routers, Switches, Backbone Connecting the CUDB Nodes)(1)	Other
Upgrade or Update	X	X	X	X
Configuration Change	X	X	X	X
Provisioning-related Activities	X	X		
Schema Change	X	X		
HW Replacement or Expansion	X	X	X	X
Change in the Amount of Subscribers in the System	X	X		
Script or Tool Execution	X	X	X	X

(1) In case the CUDB system is deployed on native BSP 8100 hardware.

4 Data Collection for Specific Problem Types

This section describes the data to be included in a CSR, depending on what type of problem is experienced.

4.1 BSP 8100

In case the CUDB system is deployed on native BSP 8100 hardware, perform the data collection procedure described in the ‘‘Data Collection Guideline for BSP’’ document in the BSP 8100 CPI, if the experienced problem is related to BSP 8100.

Glossary

For the terms, definitions, acronyms, and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [1].

Reference List

CUDB Documents

[1] *CUDB Glossary of Terms and Acronyms*

[2] *Trademark Information*

[3] *Typographic Conventions*

[4] *CUDB Troubleshooting Guide*

[5] *CUDB Node Logging Events*

[6] *CUDB System Administrator Guide*

[7] *CUDB Subscription Reallocation*

[8] *CUDB Logchecker*

[9] *CUDB Node Commands and Parameters*