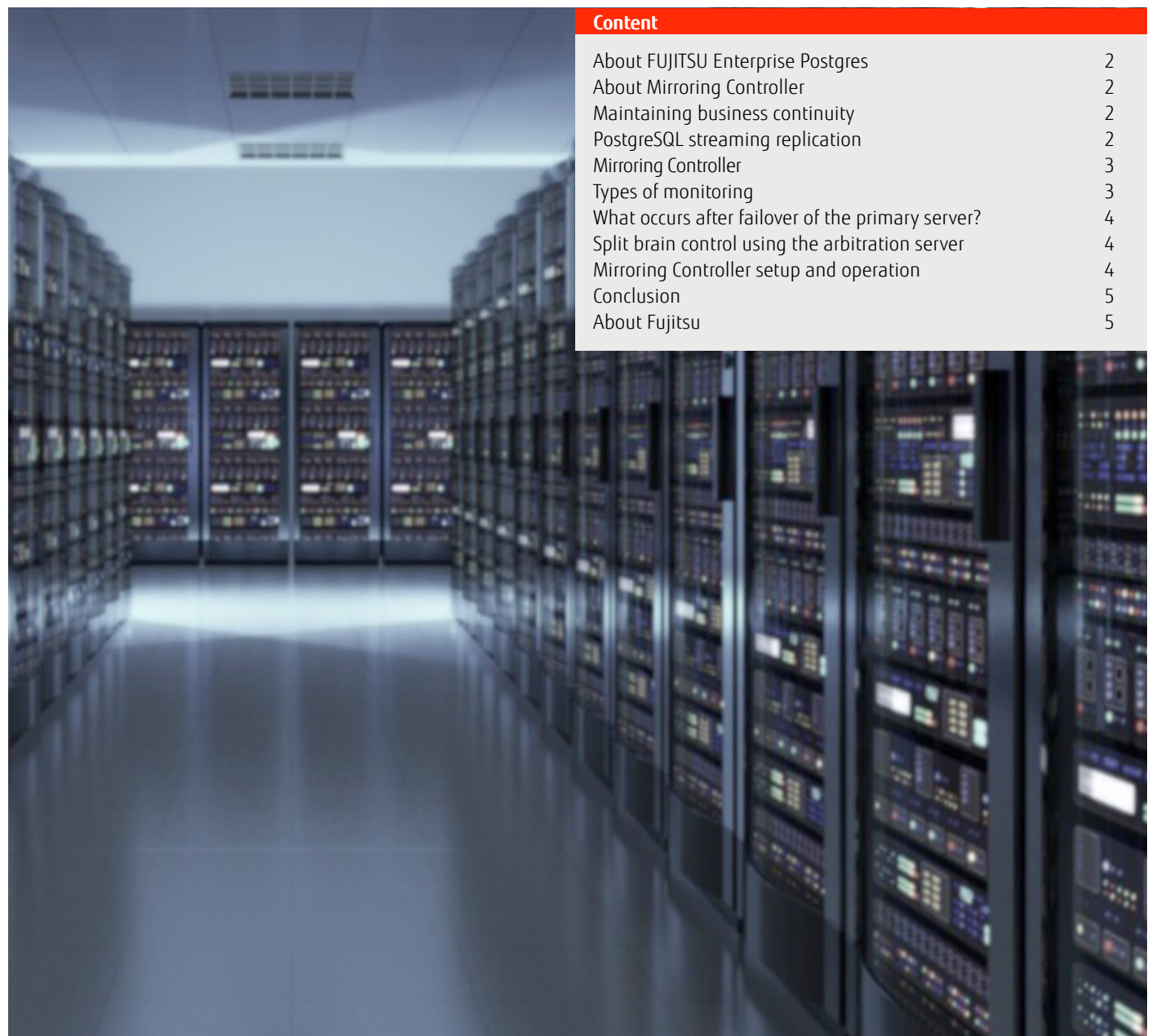


FUJITSU Enterprise Postgres Mirroring Controller

In today's demanding world of digital business, customers expect your IT services to be always available. Any time your IT service is down is costly to your business – your bottom line and your reputation will both take a hit. This drives the need for fault tolerant, high-availability database management systems such as FUJITSU Enterprise Postgres.



Content

About FUJITSU Enterprise Postgres	2
About Mirroring Controller	2
Maintaining business continuity	2
PostgreSQL streaming replication	2
Mirroring Controller	3
Types of monitoring	3
What occurs after failover of the primary server?	4
Split brain control using the arbitration server	4
Mirroring Controller setup and operation	4
Conclusion	5
About Fujitsu	5

About FUJITSU Enterprise Postgres

FUJITSU Enterprise Postgres is an exceptionally reliable and robust relational database for organisations that require strong performance, security, and high availability. FUJITSU Enterprise Postgres is fully compatible with PostgreSQL, the world's most advanced open source relational database management system, while extending PostgreSQL functionality with a number of enhanced enterprise features.

About Mirroring Controller

A feature unique to FUJITSU Enterprise Postgres is Mirroring Controller, which continually monitors your system and seamlessly switches online transaction processing to an alternate server to ensure business continuity when an abnormality is detected.

Maintaining business continuity

Database outages occur because of power supply disruption, hardware failure, network failure, site failure, software component error, and human error. When such outages occur, it is imperative that business operations be redirected without delay to an alternate system that contains current, replicated data.

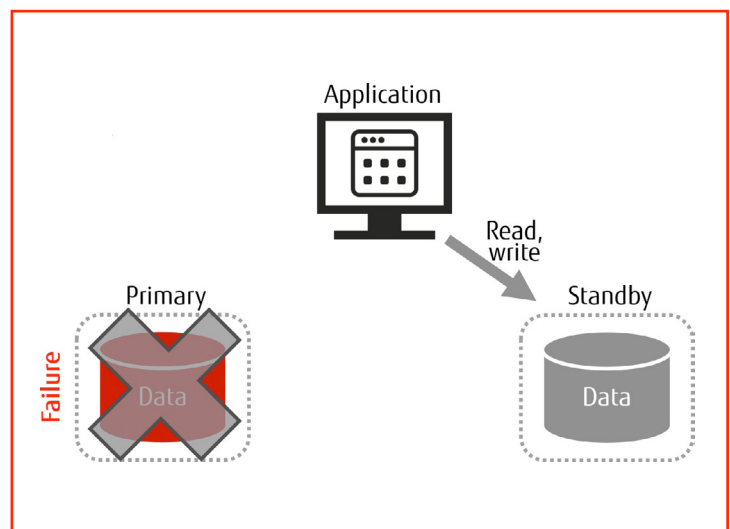
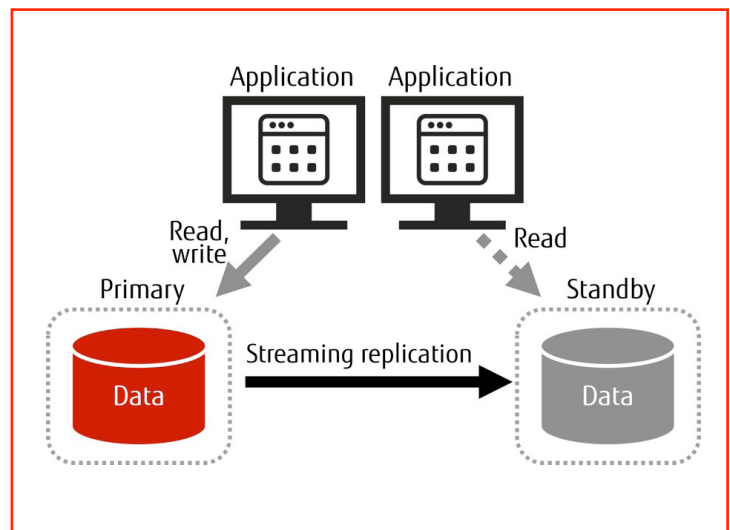
FUJITSU Enterprise Postgres was designed from the outset for high-availability, mission-critical systems, which, by definition, require maximum uptime. And this is what Mirroring Controller provides by proactively monitoring your system and automatically performing failover to an alternate system should an issue occur; this is functionality that is not provided in PostgreSQL itself.

PostgreSQL streaming replication

Databases that operate within mission-critical systems need to run in a redundant configuration where it is crucial that stability can be resumed without irreversible loss of data. In such a configuration, PostgreSQL databases are replicated using the streaming replication feature of PostgreSQL. This requires a replication cluster that contains at least two servers. In its simplest form, there will be a primary (master) server and a standby server.

In the "hot standby" configuration below, updated transaction records (write-ahead log (WAL) records) are continuously streamed from the primary database to the standby database. One added advantage of this configuration is that the standby database with near real-time data can be utilised to generate complex on-line analytic processing (OLAP) reports during normal business hours, with little impact on the primary database.

In this configuration, if the primary server fails or is unavailable, online transaction processing can be switched to the standby server to achieve and maintain business continuity.



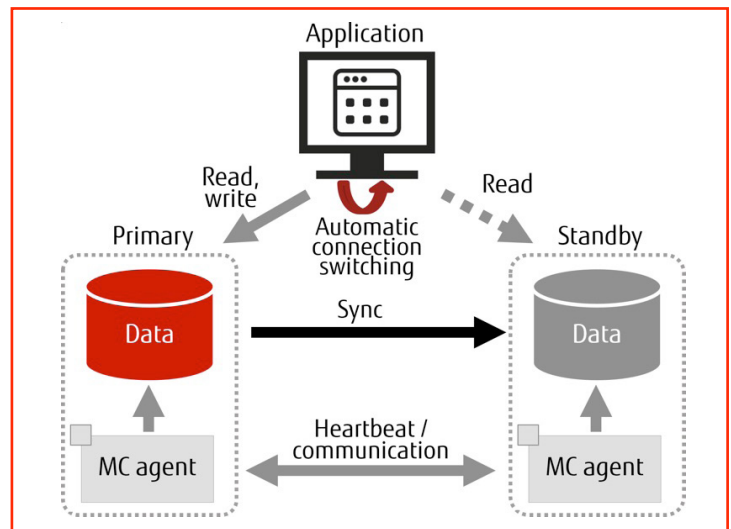
Mirroring Controller Overview

Mirroring Controller performs two main functions:

1. The Mirroring Controller agents constantly monitor your operating system, server, disk, network, and database, and notify you if something is amiss.
2. If an abnormality is detected, Mirroring Controller performs automatic switch/disconnect – if the primary server fails, processing is switched to the standby server to ensure operational continuity, and the primary server is disconnected.

You have the option of switching from the primary server to the standby server manually, or setting the Mirroring Controller to do so automatically, ensuring there are no interruptions to your database operations.

Mirroring Controller is installed on each database server. The Mirroring Controller agents communicate with each other and continually monitor server processes and operating system, network, and disk state – they even monitor the actual Mirroring Controller processes.



Types of monitoring

Operating system or server failures, and no-response state

By generating a heartbeat between the Mirroring Controller agents on each server, operating system or server errors are detected and acknowledged between the relevant servers.

The optimal operating method for environments where database multiplexing mode is performed can be selected from the following:

- Use the arbitration server to perform automatic degradation (switch/disconnect) The arbitration server objectively determines the status of database servers, then isolates and degrades from the cluster system the ones with an unstable status.
- Call the user exit (user command) that will perform the degradation decision, and perform automatic degradation
- Notification messages
Mirroring Controller outputs messages to the system log when an abnormality is detected. This ensures that a split brain will not occur due to a heartbeat abnormality – however, automatic switching will not be performed if the primary server operating system or server fails or becomes unresponsive.

Database process failures, and no-response state

Mirroring Controller periodically accesses the database processes and checks the status. A process error is detected by monitoring whether an access timeout occurs.

Disk failure

Mirroring Controller periodically creates files on the disks below. A disk error is detected when an I/O error occurs.

- Disk on which the database is stored
- Disk on which the transaction logs are stored
- Disk on which the tablespaces are stored

Streaming replication issue

Mirroring Controller detects streaming replication issues (log transfer network and WAL send/receive processes) by periodically accessing the PostgreSQL system views.

Mirroring Controller process failure and no response

In order to continue the monitoring process on Mirroring Controller, Mirroring Controller process failures and no responses are also monitored. If an issue is detected, Mirroring Controller is automatically restarted by the monitoring process.

What occurs after failover of the primary server?

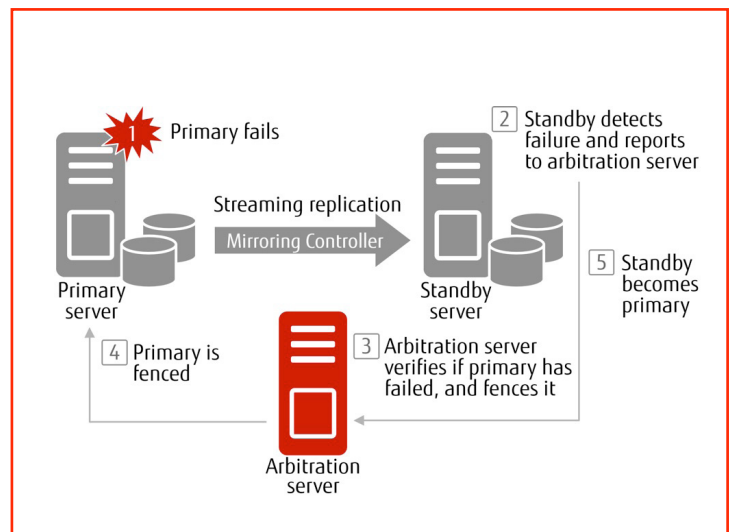
- The primary server is stopped
- Mirroring Controller triggers failover
- Mirroring Controller setup is removed from both servers
- The standby instance is promoted to a standalone instance and both servers are removed from the replication cluster. The Mirroring Controller setup is removed from both standby and master instances

Split brain control using the arbitration server

“Split brain” refers to a scenario in which two nodes in a cluster both believe they are the primary (active) node. This may occur, for example, when a standby server becomes the primary, and the original primary server comes online again. If each server considers it is active and commits any transactions that the other does not, then you cannot easily resynchronise the databases and “merge” the information on each server.

This is where the Mirroring Controller “arbitration server” comes into play. In this master-standby configuration, a third server – the arbitration server – is added to evaluate the status of the primary server reported by the standby server or vice versa, and take the appropriate action, such as fencing the malfunctioning server.

This is where the Mirroring Controller “arbitration server” comes into play. In this master-standby configuration, a third server – the arbitration server – is added to evaluate the status of the primary server reported by the standby server or vice versa, and take the appropriate action, such as fencing the malfunctioning server.



Mirroring Controller setup and operation

Mirroring Controller is included in the FUJITSU Enterprise Postgres Advanced Edition. Mirroring Controller can be set up and operated using the WebAdmin GUI management tool or via commands.

Using WebAdmin

- Set up Mirroring Controller for the master (primary) instance and standby instance
- Display the Mirroring Controller status
- Start and stop Mirroring Controller
- Enable/disable automatic fail-over
- Delete the Mirroring Controller setup

Using commands

- mc_arb: Start, stop and display the status of the Mirroring Controller arbitration process
- mc_ctl: Start and stop Mirroring Controller, switch/disconnect the server, or display the server status

```
$ mc_arb start -M /mcarb_dir/arbiter1
```

Conclusion

Mirroring Controller further enhances high-availability operations. It provides a mechanism that not only detects system abnormalities but acts on them to ensure the business continuity required by mission-critical systems.

Contact us

If you have any questions about Mirroring Controller or how to implement a high-availability database system, please contact our support team on +612 9452 9191.

Further reading

Download our white paper, "High Availability - Solutions and Implementations" from www.postgresql.fastware.com/white-papers

About Fujitsu

Fujitsu is the 5th largest IT service provider in the world, offering a full range of technology products, solutions and services. Around 160,000 Fujitsu employees support customers in over 100 countries.

Published by

Fujitsu Australia Software Technology Pty Ltd
14 Rodborough Road, Frenchs Forest NSW 2086, Australia
© FUJITSU AUSTRALIA SOFTWARE TECHNOLOGY PTY LTD 2017
Document number 14975a. 2018-10-02 WW EN

Copyright 2018 FUJITSU AUSTRALIA SOFTWARE TECHNOLOGY. Fujitsu, the Fujitsu logo and Fujitsu brand names are trademarks or registered trademarks of Fujitsu Limited in Japan and other countries. Other company, product and service names may be trademarks or registered trademarks of their respective owners. All rights reserved. No part of this document may be reproduced, stored or transmitted in any form without prior written permission of Fujitsu Australia Software Technology. Fujitsu Australia Software Technology endeavours to ensure the information in this document is correct and fairly stated, but does not accept liability for any errors or omissions.