

# WHITE PAPER

# **Heterogeneous Data Replication**

## Abstract

This white paper will give you an overview of SymmetricDS data replication, including its key features, modular architecture, and common use cases. Data replication can provide high availability, load distribution, and disconnected operation. It can ensure applications have reliable uptime and fast response times. It can improve performance by balancing load across databases or offloading work to a reporting database. SymmetricDS can enable a distributed workforce on laptops and mobile devices that are occasionally connected. An enterprise with disparate and heterogeneous systems can use replication as part of an integration strategy for enriching application databases, Data Warehouses, mobile devices, and cloud-based solutions.

Copyright © 2017 JumpMind, Inc. All rights reserved. JumpMind, JumpMind logo, SymmetricDS, and SymmetricDS logo are trademarks of JumpMind in the United States and other countries. All other trademarks are property of the respective owners.

#### SYMMETRICDS: HETEROGENEOUS DATA REPLICATION

#### INTRODUCTION

SymmetricDS is cross platform data replication that keeps databases and file systems in synchronization. Using web server technology, it builds on a familiar and proven platform for scaling to thousands of sources and targets. Data is replicated continuously in the background, enabling real-time access to information. It can also subset data, apply conditions, resolve conflicts, and transform data. With an emphasis on resilience and interoperability, it can run across different networks, operating systems, and databases.

#### **KEY FEATURES**

SymmetricDS offers a rich set of features with a flexible configuration for large scale deployment in a mixed environment of multiple systems.

**Data Replication** - Change data capture for relational databases and file replication for file systems can be periodic or real-time, with an initial load feature to fully populate a node.

**Cross Platform** – Support for most major databases (over 20 dialects) and operating systems, with options for deploying to the cloud and mobile devices.

**Automatic Recovery** - Data delivery is durable and low maintenance, withstanding periods of downtime and automatically recovering from a network outage.

**Secure and Efficient** - Communication uses a minimal, compressed data protocol designed for low bandwidth networks and streamed over HTTPS for encrypted transfer.

**Central Management** - Configure, monitor, and troubleshoot replication from a central web console where conflicts and errors can be investigated and resolved.

**Transformation** - Manipulate data during multiple phases to filter, subset, translate, merge, and enrich data.

**Conflict Management** - Enforce consistency of two-way synchronization by configuring rules for automatic and manual resolution.

**Extendable** - Scripts and Java code can be configured to handle events, transform data, and create customized behavior using available configuration and plug-in points.

**Flexible Deployment** - The software can be installed either as a self-contained server that stands alone, deployed to a web application server, or embedded within an application.

erview									Node Status							
93% Health	thy	(	D Processes	s active					Nodes 7 nodes syn	sing						
0 Outgoing	Errors		Process +	Running	Nodes	Batches	Rows	Last Run				Outgoing			Incoming	
0 Incoming	Errors		Routing	-	2	0	1	1 s ago	Node	Status	Rows	Batches	Last	Rows	Batches	Last
0 Offline No	odes		Push Handler	0	7	0	101	114 s ago	Columbus	e	89	0	3 m 53 s ago	1	0	25 s ago
0 Monitors I	Fired		Pull Handler	-	7	7	623	183 s ago	Tampa	o	89	0	3 m 3 s ago	1	0	29 s ago
			.oad Setup	-	1	0	0	1 s ago	Portland	0	89	0	4 m 28 s ago	1	0	52 s ago
								more	Atlanta	e	89	0	4 m 11 s ago	1	0	8 m 28 s agr
									NewYork	0	89	0	3 m 12 s ago	0	0	43 s ago
									Austin	0	89	0	4 m 24 s ago	0	0	7 m 56 s ag
ch Status	Add Source/	/Target Database	Add	Table(s) To Sync		≓ Load Existi	ng Data		Austin SanDiego Throughput	0	89 89	0	4 m 24 s ago 3 m 47 s ago	0	0	7 m 56 s ag 24 m 10 s a
ch Status Batches Changes f	Add Source/	/Target Database o 7 nodes when p	Add	Table(s) To Sync	Itches Inco	≓ Load Existi ming bles pushed to t	ng Data	7 nodes every 5s	Austin SanDiego Throughput Market Stats are a	2 2 Dut available since	89 89 21 h 46 m 20 s	0 0	4 m 24 s ago 3 m 47 s ago	0	0	7 m 56 s ag 24 m 10 s a Hours
Ch Status Batches Changes f Node	Add Source/ es Outgoing for 2 tables sent to Batch Id	/Target Database o 7 nodes when p Progress	III Add	Table(s) To Sync	atches Inco anges for 2 tab	Load Existi     Defining     Deles pushed to t     atch Id	ng Data this node from Progress	7 nodes every 5s Status =	Austin SanDiego Throughput Market Stats are a	2 2 Dut available since	89 89 21 h 46 m 20 s	0 0	4 m 24 s ago 3 m 47 s ago	0	0	7 m 56 s ag 24 m 10 s a Hours
Ch Status Batches Changes f Node Portland	Add Source/	/Target Database o 7 nodes when p Progress	Ulled every 5s	Table(s) To Sync Ba Chu Node NewYo	ntches Incc anges for 2 tab Bi rk	The second existing the second existing the second existing of the second existing e	ng Data this node from Progress	7 nodes every 5s Status = Loading	Austin SanDiego Throughput Market Stats are a	2 2 Dut available since	89 89 21 h 46 m 20 s	0 0 s ago Rected	4 m 24 s ago 3 m 47 s ago	0 0 Loaded	0	7 m 56 s ag 24 m 10 s a Hours
Ch Status Batchee Changes f Node Portland Atlanta	Add Source/	/Target Database	Ulled every 5s Status Waiting Waiting	Table(s) To Sync Ba Ch Node NewYo Atlant	anges for 2 tab Bi ork a	E Load Existi oming oles pushed to to atch Id 101	ng Data this node from Progress	7 nodes every 5s Status Loading Loading	Austin SanDiego Throughput Market Stats are a	2 2 Dut available since	89 89 21 h 46 m 20 :	0 0 s ago ted Routed	4 m 24 s ago 3 m 47 s ago	0 0	0	7 m 56 s ag 24 m 10 s ag Hours
ch Status Batcher Changes f Node Portland Atlanta Columbus	Add Source/	/Target Database	Add	Table(s) To Sync Ba Chi Node NewYo Atlant Columb	ttches Incc anges for 2 tat b Bi ork a uus	E Load Evisti oning bles pushed to t atch Id 101 101 49 10	ng Data this node from Progress	7 nodes every 5s Status = Loading Loading Loading	Austin SanDiego Throughput M Stats are a	© © Dut available since	89 89 21 h 46 m 20 :	0 0 s ago ted Routed	4 m 24 s ago 3 m 47 s ago	0 0	0	7 m 56 s ag 24 m 10 s ag Hours
Changes f Node Portland Atlanta Columbus NewYork	Add Source/	/Target Database	Add	Table(s) To Sync Back Children Childre	ttches Inco anges for 2 tab srk a uus tus tad	E Load Existing best pushed to 1 atch Id 101 1 49 1 49 1 49 1 49 1	ng Data this node from Progress	7 nodes every 5s Status Loading Loading Loading Loading Loading	Austin SanDiego Throughput Throughput Through Stats are a	C C Dut uvailable since	89 89 21 h 46 m 20 :	0 0 s ago	4 m 24 s ago 3 m 47 s ago	0 0	0	7 m 56 s ag 24 m 10 s ag Hours
ch Status Batcher Changes f Node Portland Atlanta Columbus NewYork SanDiego	Add Source/      Source/	/Target Database	Add     A	Table(s) To Sync	Itches Incc anges for 2 tak b Bi rrk Ita itus Ita itd a	E Load Existi Deles pushed to tatch Id 101 49 1 49 1 49 1 50 1	ng Data	7 nodes every 5s Status = Loading Loading Loading Loading Loading	Austin SanDlego Throughput Throughput Stats are a	C C Dut wailable since	89 89 21 h 46 m 20 :	0 0 s ago	4 m 24 s ago 3 m 47 s ago	0 0	0	7 m 56 s ag 24 m 10 s ag
ch Status Batches Changes Portland Atlanta Columbus NewYork SanDiego re •••	▲ Add Source/ es Outgoing for 2 tables sent to Batch Id 800 801 802 803 804	/Target Database	Add      diled every 5s      Status      Waiting      Waiting	Table(s) To Sync	tiches inco anges for 2 tab b Bi rrk ta 100 100 100 100 100 100 100 100 100 10	E Load Existi Deles pushed to 1 atch Id 49 1 49 1 50 1	ng Data	7 nodes every 5s Status E Loading Loading Loading Loading	Austin SanDiego Throughput Market Stats are of Stats are of 200 200 100	C C Dut wailable since	89 89 21 h 46 m 20 :	0 0 s ago	4 m24 s ago 3 m 47 s ago	0 0	0	7 m 56 s aç 24 m 10 s a

*Figure 1:* The web console simplifies configuration, monitoring, and troubleshooting from a central location.

## ARCHITECTURE

The architecture for SymmetricDS consists of modular services that process change data from a source database and loads them into a target database. Each database is managed by a *node* running these services, which connects to the database using JDBC for creation of runtime objects and handling of change data. Nodes are configured inside a running instance of SymmetricDS. When communicating across a WAN, the best performance is achieved by installing an instance next to each database. The durability of the database is used to guarantee that data is replicated and acknowledged, despite any adverse network conditions, power loss, or server crashes.



*Figure 2:* Nodes connect to the nearby database with JDBC over the LAN and connect to remote nodes with HTTPS over the WAN.

#### SYMMETRICDS: HETEROGENEOUS DATA REPLICATION



*Figure 3:* The SymmetricDS architecture uses multiple phases to process change data, with each phase offering settings and plug-in points.

#### Capture

Change data capture occurs at the source database where changes are recorded by triggers into a runtime table. Each change includes a sequence number and transaction identifier to preserve the order and grouping of committed rows. Capture is configured for specific tables and columns, and rows can be matched with conditional values.

#### Route

Changes are grouped together into batches and assigned to targets that will receive them. Routing can distribute rows across multiple targets and send rows to a target based on matching values in a row or looking up values from another table.

## Extract

Batches are extracted from the database and written to a temporary staging area, which is compressed and encrypted. Staging is optional but recommended to improve performance because it avoids the overhead and lag of holding open the database and network resources.

## Transform

During extraction at the source or before loading at the target, transformation can be performed on change data to map, filter, translate, and modify rows, columns, and their

#### SYMMETRICDS: HETEROGENEOUS DATA REPLICATION

values. Data is transformed using built-in functions or custom scripts. The local database can be queried to enhance data.

#### Load

Outgoing batches from the source are transferred over HTTP/S to the target as an incoming batch to be loaded. Channels are configured to control the number of batches sent on the connection and how much bandwidth can be consumed. Communication can be initiated by either source or target using a "push" or "pull" method, allowing for traversal of firewalls. Batch data is committed along with a status, and the status is acknowledged to the source.

# **COMMON USE CASES**

SymmetricDS has a flexible configuration that allows databases to be replicated across a range of topologies from simple to complex. Common uses cases include:

- · Data consolidation at central location and distribution across branch locations
- · Consolidating disparate systems into a Data Warehouse
- Offloading an operational database to a reporting database
- Transforming data between different application databases
- · Backup and fail-over of critical business data
- · Cloud integration between providers and on-premise systems
- Disconnected operation on devices for a distributed workforce



*Figure 4:* Multiple use cases are possible, including 1-way, bi-directional, one source to multiple targets, multiple sources to one target, multi-master, and tiered replication.

#### PERFORMANCE

SymmetricDS includes features for performance scaling and fault tolerance. Each remote node is serviced with separate threads to run extract, transfer, and load phases in parallel. Additional threads can be configured for parallel loading of data by assigning tables to different channels. By deploying across multiple servers and enabling clustering, many simultaneous requests can be handled, with deployments in production supporting more than ten thousand (10,000) nodes. With a load balancer in front of the cluster, the central servers become highly available to remote nodes. Large networks of nodes can be grouped into tiers for more control and efficiency, with each group synchronizing data to the next tier.



*Figure 5:* SymmetricDS data replication performance with load increasing to a peak of 41,418 rows/second between two Oracle 11.2 databases. Tested with SymmetricDS 3.8 on Redhat Enterprise Linux 7 with 4-core, 64-bit CPU and 2 GB of memory.

## **ENABLING BUSINESS**

Cross platform data replication can be part of an integration strategy that improves access to critical business data. Real-time integration gives a competitive advantage through business intelligence and enabling new kinds of services. Customers expect self-service applications and web sites to include interaction with real-time data. An architecture with real-time information and on-demand processing can positively affect the decision-making process within an organization. SymmetricDS offers a flexible, feature-rich data replication solution that offers high value, low maintenance, and a diverse platform support that can work across a heterogeneous enterprise.