

# Datenbankelastizität leicht gemacht: zimory®scale

Christian Link  
zimory  
Berlin

## Schlüsselworte

Query-Routing, Datenbank-Replikation, Oracle, MySQL, Elastizität, Cloud, VM, Datenbank Middleware

## Einleitung

Mit zimory®scale bietet zimory eine Datenbank Middleware, die zum einen bestehende Oracle-Datenbanken schemaweise in Satellitendatenbanken repliziert und zum anderen den mit JDBC angebundenen Datenbankanwendung eine transparente Schicht für die Datenbankzugriffe bietet.

Durch intelligente Algorithmen werden die Datenbankzugriffe analysiert und ein Großteil der lesenden Zugriffe auf die Satellitendatenbanken umgeleitet.

Dies führt zu einer spürbaren Entlastung der Masterdatenbank.

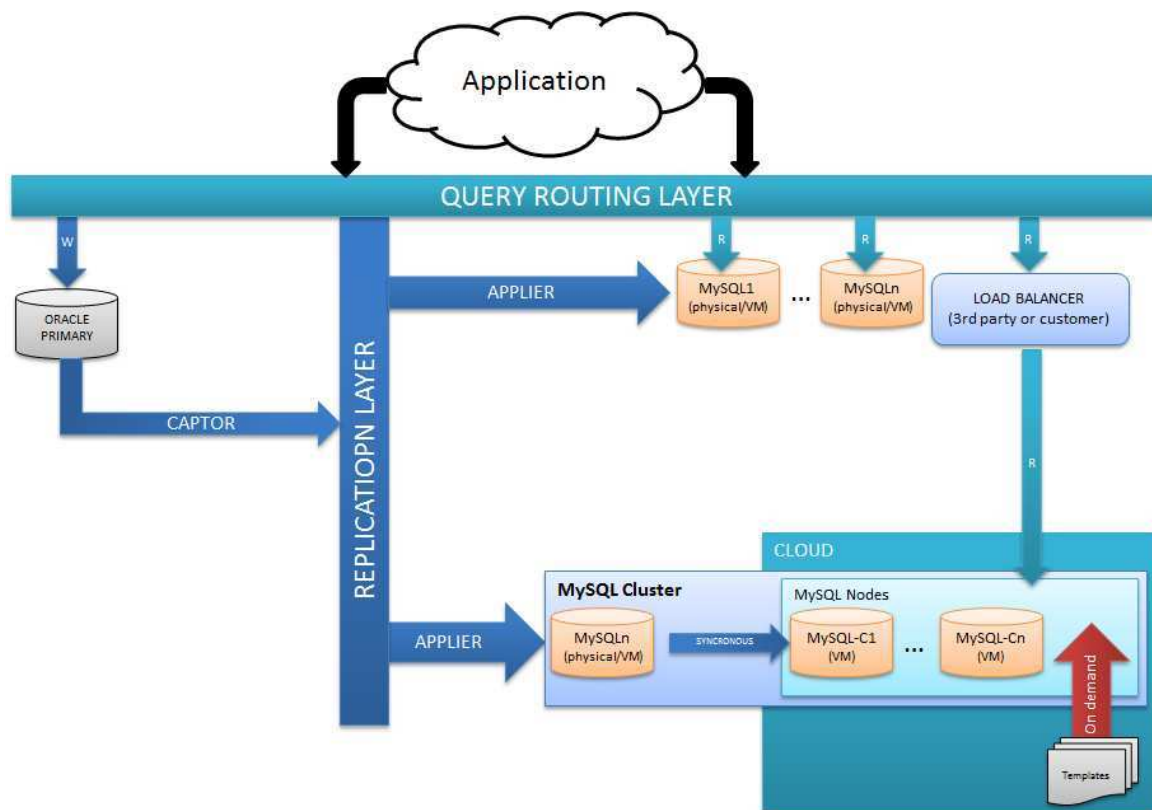


Abb. 1: Architektur

## Replikation

zimory@scale basiert auf Oracles LogMiner, liest alle Changes direkt aus der Master-Datenbank aus und fügt diese via JDBC in die Satelliten-Datenbanken ein.

Aus diesem Grund können die Datenbank-Versionen und –Edition der Master- und Satelliten-Datenbanken unterschiedlich sein, z.B. Enterprise Edition (Master) und Standard-Edition (Satellite). Ebenso können die Satelliten-Datenbanken mit einem nicht-Oracle-System betrieben werden, z.B. MySQL.

Eine parallele Replikation in Oracle- und nicht-Oracle-Datenbanken ist ebenfalls möglich.

In der Roadmap ist die Unterstützung weiterer Datenbank-Engines wie PostgreSQL und noSQL-Datenbanken geplant.

Die nötigen Datenkonvertierungen und SQL-Übersetzungen finden on-the-fly in der Replikationsschicht statt, es findet kein Zwischenspeichern auf Platte statt.

Für das Einfügen der Daten in die Satelliten-Datenbanken steht pro Satelliten-Datenbank eine eigene Verbindung bereit, so dass die Satelliten sich nicht gegenseitig beeinflussen können.

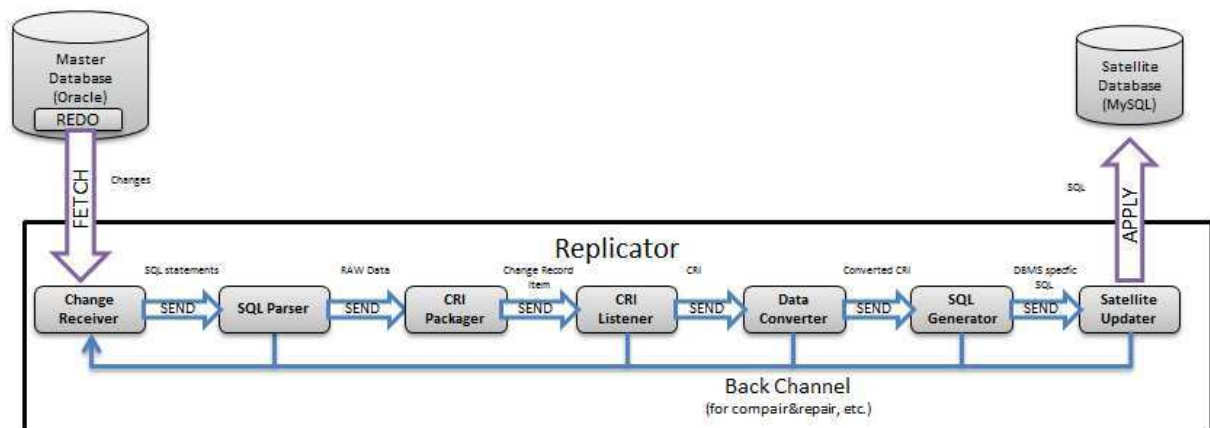


Abb.2: Replikationsschicht

Die Replikationsschicht bietet verschiedene Modi an.

So kann bei anstehenden massiven Datenänderungen (BULK CHANGES) der Modus „BULK“ aktiviert werden, welcher für das Verarbeiten von großen, sequentiellen Datenmengen optimiert wurde.

Treten Replikationsfehler auf und betreffen diese nur *unkritische* Tabellen, kann der Administrator den Modus „SKIP“, welcher diese Fehler ignoriert, auswählen.

Bei Bedarf kann die Kommunikation zwischen den Datenbanken auch verschlüsselt und komprimiert werden.

Die Daten der Master-Datenbank werden schemawise repliziert, wobei durch Nutzung von Blacklists bestimmte Tabellen oder Statements ausgeblendet werden können.

So kann z.B. durch das Auslassen von DELETE- und TRUNCATE-Anweisungen eine Satelliten-Datenbank als Archiv betrieben werden.

## Query Routing

Neben der reinen Datenbank-Replikation erweitert zimory@scale eine Datenbank-Umgebung mit einer transparenten Schicht, welche die Datenbankzugriffe der Applikationen analysiert und mit Hilfe von intelligenten Algorithmen auf die Master-Datenbank (alle Schreibzugriffe) und die Satelliten-Datenbanken (lesende Zugriffe) verteilt.

Durch den Einsatz mehrerer Satelliten-Datenbanken ist so eine enorme Entlastung der Master-Datenbank möglich.

Voraussetzung hierfür ist jedoch eine Datenbanknutzung mit mehrheitlich lesenden Zugriffen der Applikationen.

Werden die Satelliten-Datenbanken als VMs und als MySQL-Cluster, z.B. Percona, in einer Cloud betrieben, kann durch Erhöhung oder Verringerung optimal auf wechselnde Anforderungen der Applikationen, z.B. saisonale Lastspitzen, reagiert werden.

In ruhigen Phasen können Kosten durch das Abschalten von Satelliten-Datenbanken gespart werden.

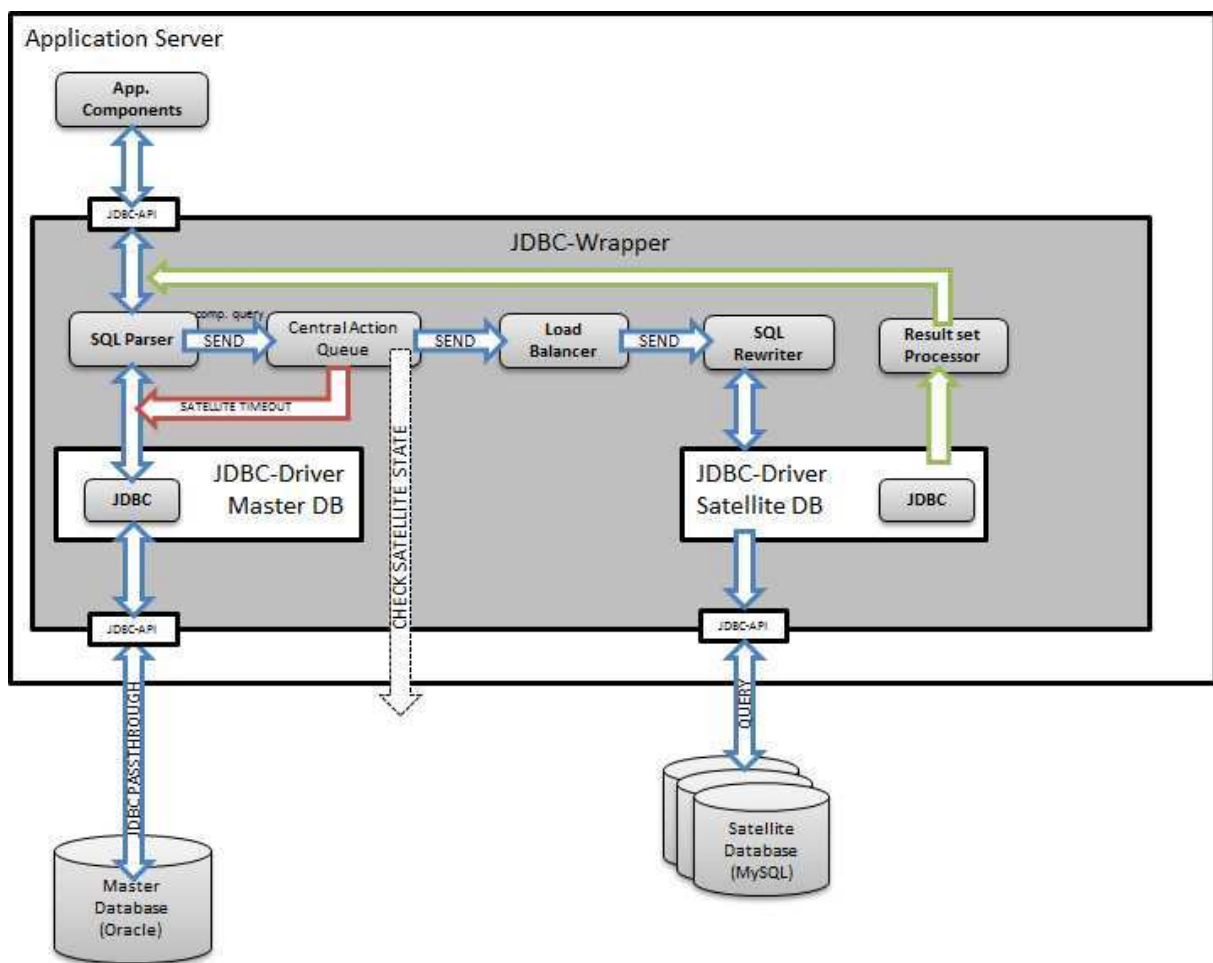


Abb. 3: Query Routing

## Administration

Der Einsatz von zimory@scale ist ohne Eingriffe in die Master-Datenbank möglich.

Das System läuft auf einem eigenen Server und greift über das Netzwerk auf die Master-Datenbank und die Satelliten-Datenbanken zu.

Mittels einer Konsole kann das System verwaltet und die Replikation eingerichtet, erweitert und überwacht werden.

Command	Description
version	Shows the Replicator version
help <command>	Shows help for a command
connect <IP/hostname>	Connects to a Replicator instance
disconnect	Disconnects from the Replicator instance
status	Shows the client status
addsat	Adds a satellite to the Replicator (interactive)
remsat <satellite_ID>	Removes a satellite from the Replicator
showsat <satellite_ID>	Shows properties about a satellite
showsat all	Shows properties about all satellites
start <satellite_ID>	Starts replication to a satellite (interactive)
start all	Starts replication to all satellites (interactive)
restart <satellite_ID>	Restarts replication to a satellite
restart all	Restarts replication to all satellites
stop <satellite_ID>	Stops replication to a satellite
stop all	Stops replication to all satellites
stop abort <satellite_ID>	Hard stops replication to a satellite
stop abort all	Hard stops replication to all satellites
stats <satellite_ID>	Shows statistics about a satellite
clearsat <satellite_ID>	Removes replicator related state from a satellite
bulk <enable/disable> <satellite_ID>	Toggles Batch Mode On or Off
createtables <satellite_ID>	Creates master db tables at the specified satellite.
createindexes <satellite_ID>	Creates indexes on specified satellite.
exit	Disconnects and exits from the console
quit	Disconnects and exits from the console
help	Shows help

Abb.4:Konsole

## Skalierbarkeit, Elastizität und Kosteneinsparung

Relationale Datenbanken sind in aller Regel statische, unflexible, auf Lastspitzen ausgelegte Systeme. Im Regelbetrieb wird hingegen meist nur ein Bruchteil der maximalen Performance benötigt, dennoch ist eine Vorabinvestition in Hardware und Lizenzen notwendig.

zimory@scale erweitert bestehende Datenbanken und nutzt für diese Erweiterung lizenzfreie Alternativsysteme, wie z.B. MySQL, welche auch als VM betrieben werden können.

Diese Erweiterungen können zur Laufzeit aus- oder zurückgebaut werden, so dass sich die Kosten für den Betrieb der Server/VMs immer an der tatsächlich benötigten Leistung orientiert.

Werden die Satelliten-Datenbanken zudem als MySQL-Cluster betrieben, können Administratoren sehr schnell und flexibel auf Lastspitzen reagieren und z.B. mit Percona MySQL bei Bedarf weitere Nodes in ein Cluster einfügen oder überschüssige Nodes entfernen.

Dies geschieht ohne Eingriff in die Master-Datenbank und hat keinen Einfluss auf den Betrieb der produktiven Datenbank.

Satelliten-Datenbanken können sowohl inhaltlich als auch in der Struktur (Indexe für Lese-Optimierung, etc.) von der Master-Datenbank unterscheiden.

So können diese Satelliten ggfs. deutlich kleiner dimensioniert werden und benötigen weniger (Speicher-)Ressourcen.

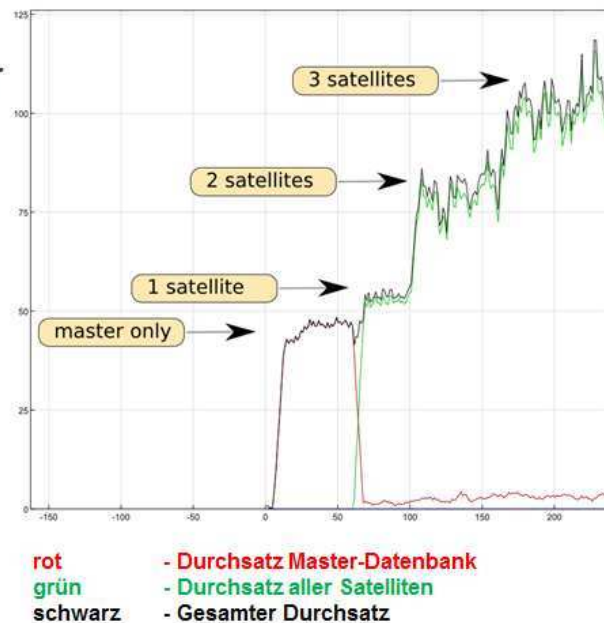
# Dynamische Datenbank-Skalierung

## Dynamische Datenbank-Skalierung zur Steigerung der Elastizität

- Ohne Downtime
- Flexibles Bereitstellen von Satelliten (> MySQL Cluster)
- Wahrung der Datenbankkonsistenz

### Beispiel (TPC-W Benchmark)

- 3% write
- 97% read



**Kontaktadresse:**

Christian Link  
zimory GmbH  
Revaler Straße 100  
D-10245 Berlin

Telefon: +49 (0) 30 6098507-35  
Fax: +49 (0) 30 6098507-99  
E-Mail: [christian.link@zimory.com](mailto:christian.link@zimory.com)  
Internet: [www.zimory.com](http://www.zimory.com)