



MySQL-Server im Teamwork - Replikation und Cluster

DOAG-Konferenz, 2015-Nov-17

Jörg Brüche

Senior Support Engineer, FromDual GmbH

joerg.bruehe@fromdual.com



FromDual GmbH

Support



Beratung



remote-DBA



Schulung



Zur Person

- **Entwicklung verteiltes SQL-DBMS:**
Unix-Portierung,
Anschluss Archivierungs-Tools (ADSM, NetWorker)
- **MySQL Build Team:**
Release-Builds inkl. Tests, Paketierung, Skripte, ...
- **DBA:**
MySQL für eine Web-Plattform
(Master-Master-Replikation)
- **Support-Ingenieur (FromDual):**
Support + Remote-DBA für MySQL / MariaDB / Percona
mit oder ohne Galera Cluster

Inhalt

MySQL Server: Architektur

Binlog

Replikation

Galera Cluster

Fazit

Allgemeines

- **Konzepte, nicht Details
„der Wald, nicht die Bäume“**
- **MySQL 5.5 / 5.6 (aktuelle GA-Versionen)**
- **Übertragbar von MySQL (Oracle) auf
Percona und MariaDB**
- **Nicht anwendbar auf „embedded“ MySQL**
- **Nicht betrachtet: NDB = „MySQL Cluster“**

➔ **MySQL Server: Architektur**

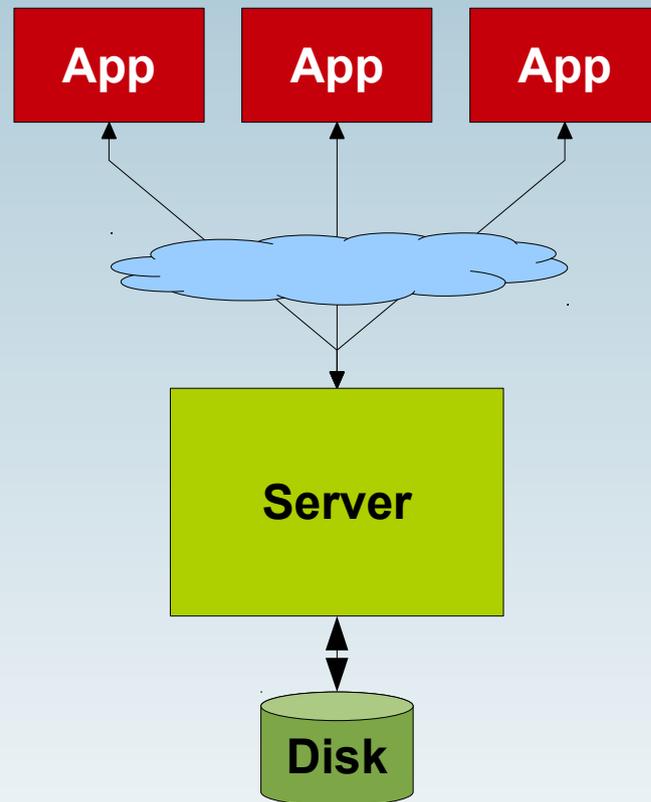
Binlog

Replikation

Galera Cluster

Fazit

Client-Server-DBMS



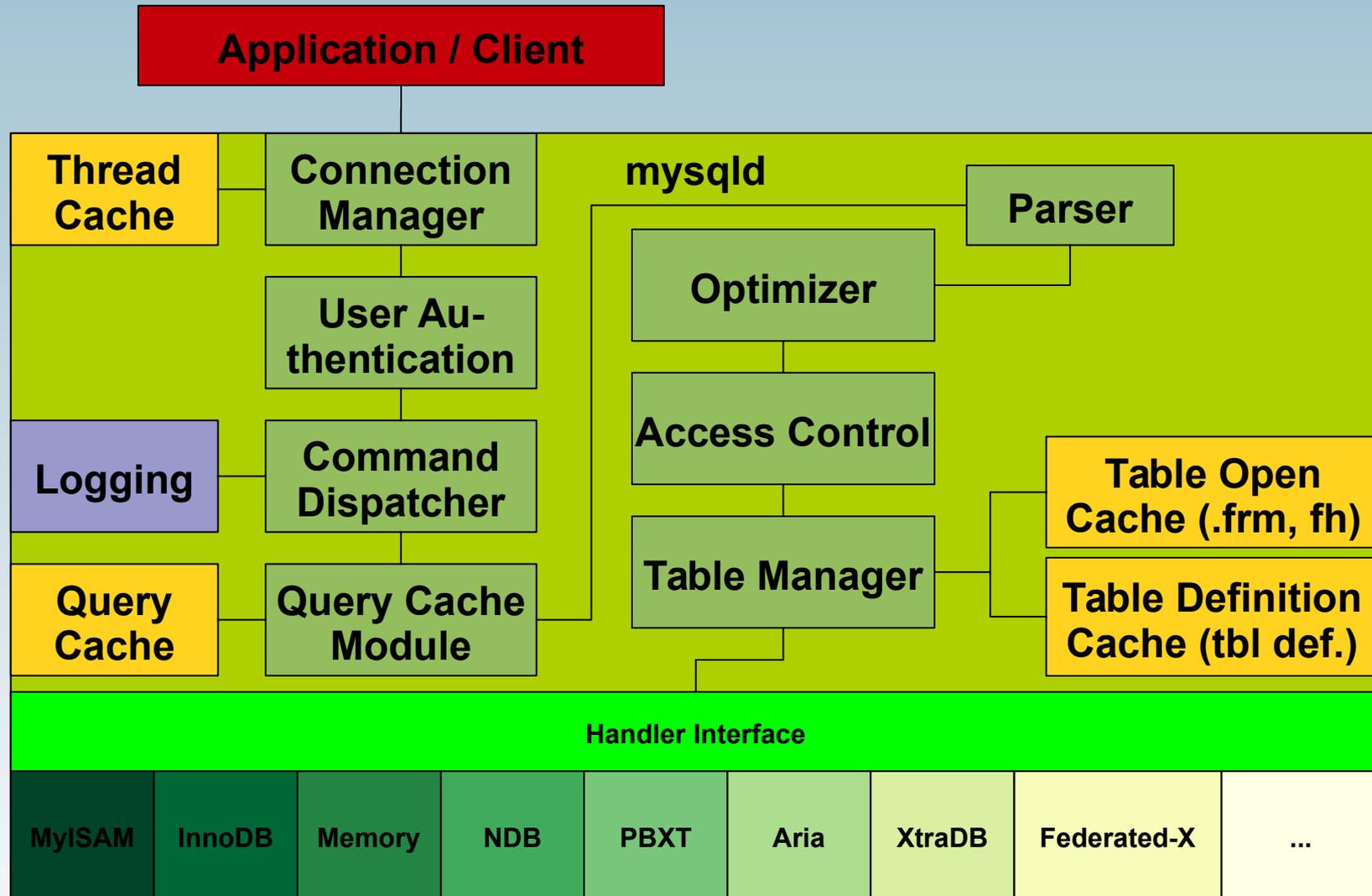
Client (Applikation)
lokal oder remote

Socket, LAN oder Internet

Server ist eigener Prozess
Multi-threaded:
1 Thread je Session

Platte / SSD, lokal oder SAN

Server intern



MySQL Server: Architektur

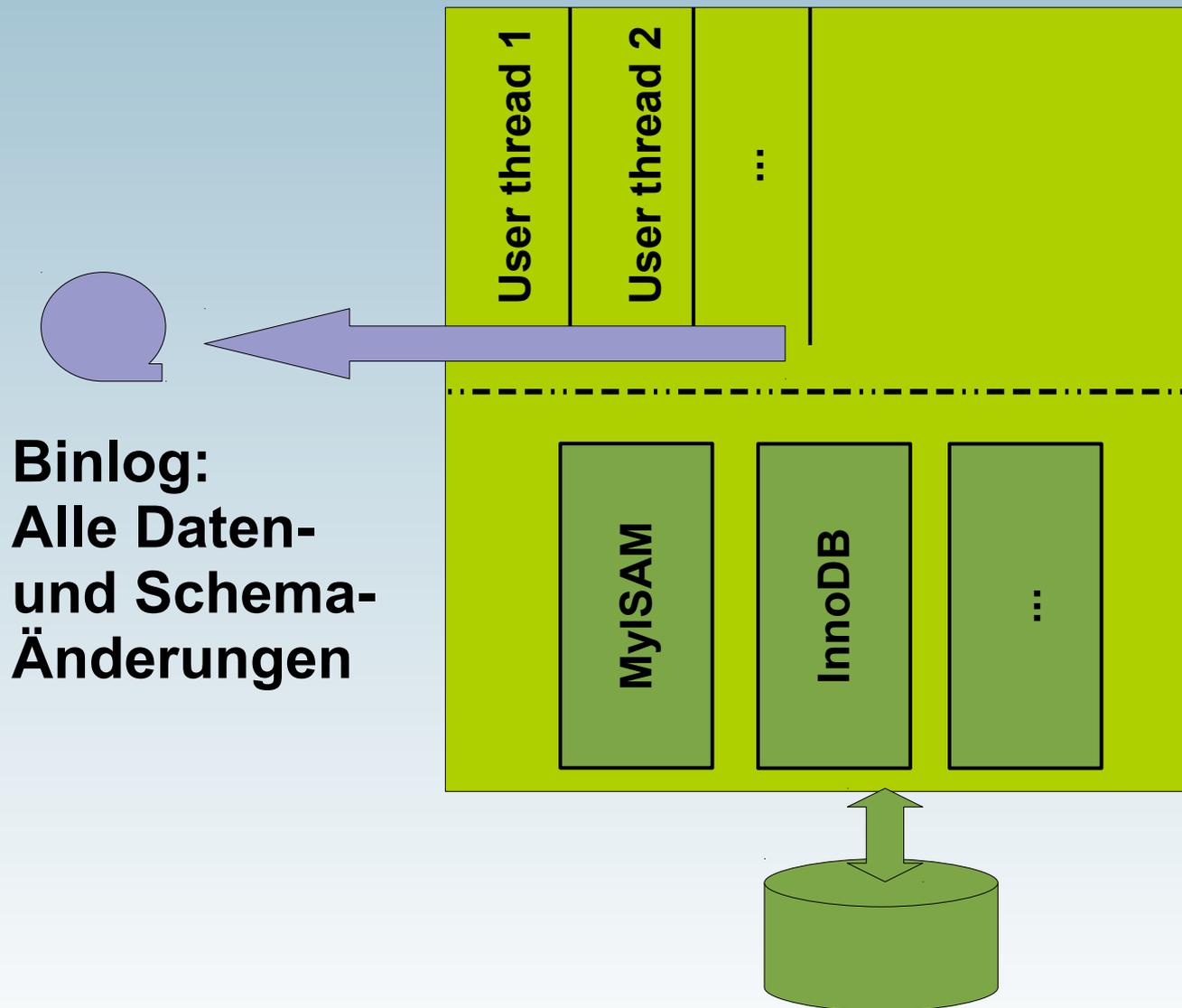
➔ **Binlog**

Replikation

Galera Cluster

Fazit

Ebenen + Binlog



SQL-Ebene:

- Parser
- Optimizer
- Privilegien
- Query Cache
- ...

Handler Interface

Datei-Ebene:

- Tabellen-Handler
- InnoDB:
 - Satz-Zugriffe
 - Satz-Sperren
 - Recovery
- ...

Binlog

- **Alle ausgeführten Daten-Änderungen**
- **Alle ausgeführten Schema-Änderungen**
- **Zeitstempel**
- **Unabhängig von Tabellen-Handler**
- **Formate „statement“, „row“ und „mixed“**
- **Zwingend für Point-in-Time-Recovery „PITR“**
- **Segmente mit konfigurierbarer Größe**
- **Fortlaufend nummeriert**

MySQL Server: Architektur

Binlog

➔ **Replikation**

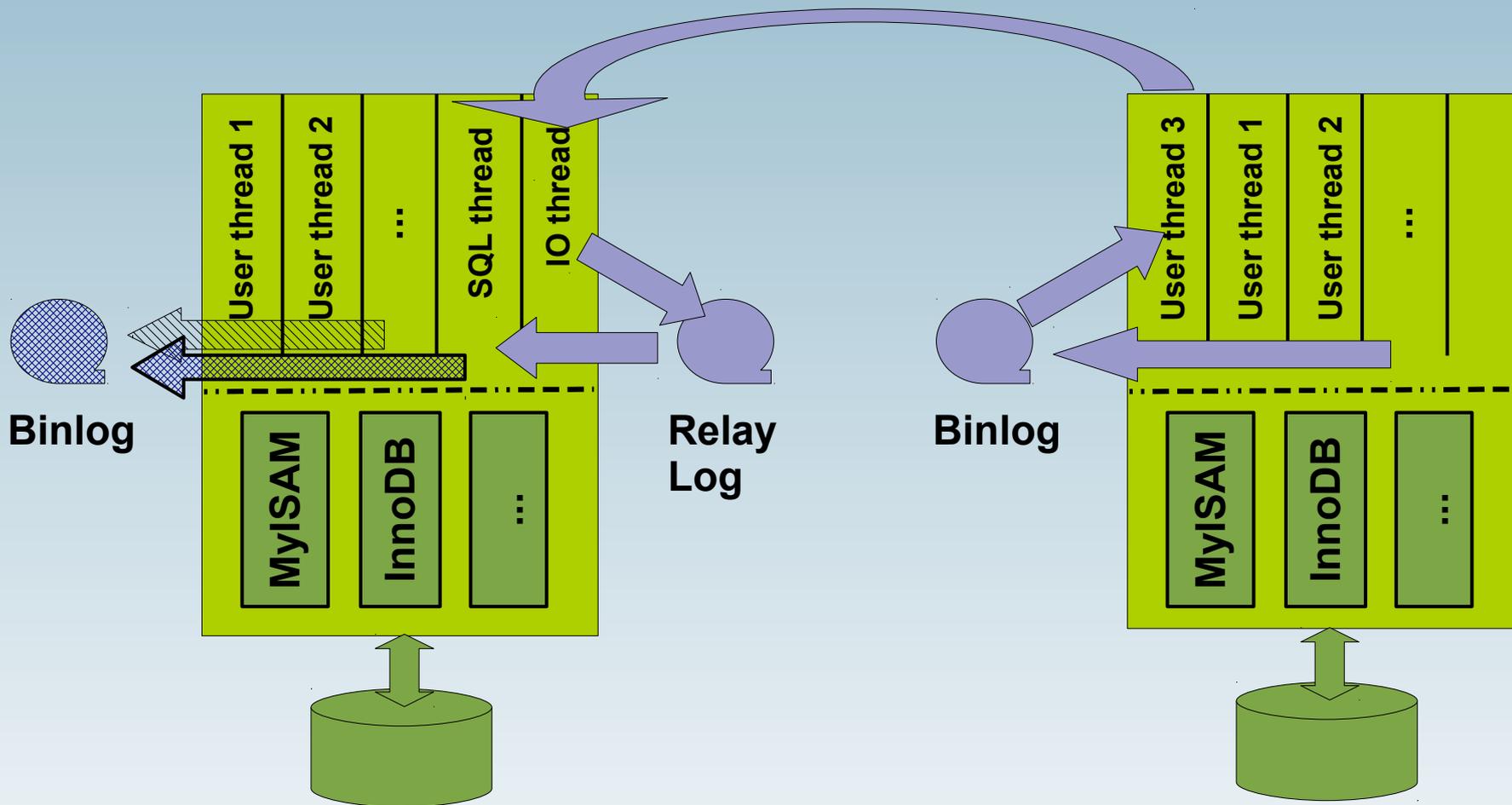
Galera Cluster

Fazit

Replikation bei MySQL

- Anwendungen kommunizieren mit „Master“
- „Master“ protokolliert alle Änderungen
- „Slave“ hat identischen Anfangszustand
- Slave holt alle Änderungen vom Master und wendet sie bei sich an
- Replikation läuft asynchron
- Slave stoppt Replikation bei Abweichung

Slave holt Binlog vom Master



Slave:

“log-bin = FILE”, sonst kein Binlog

“log_slave_updates = 1” für Weiterleitung

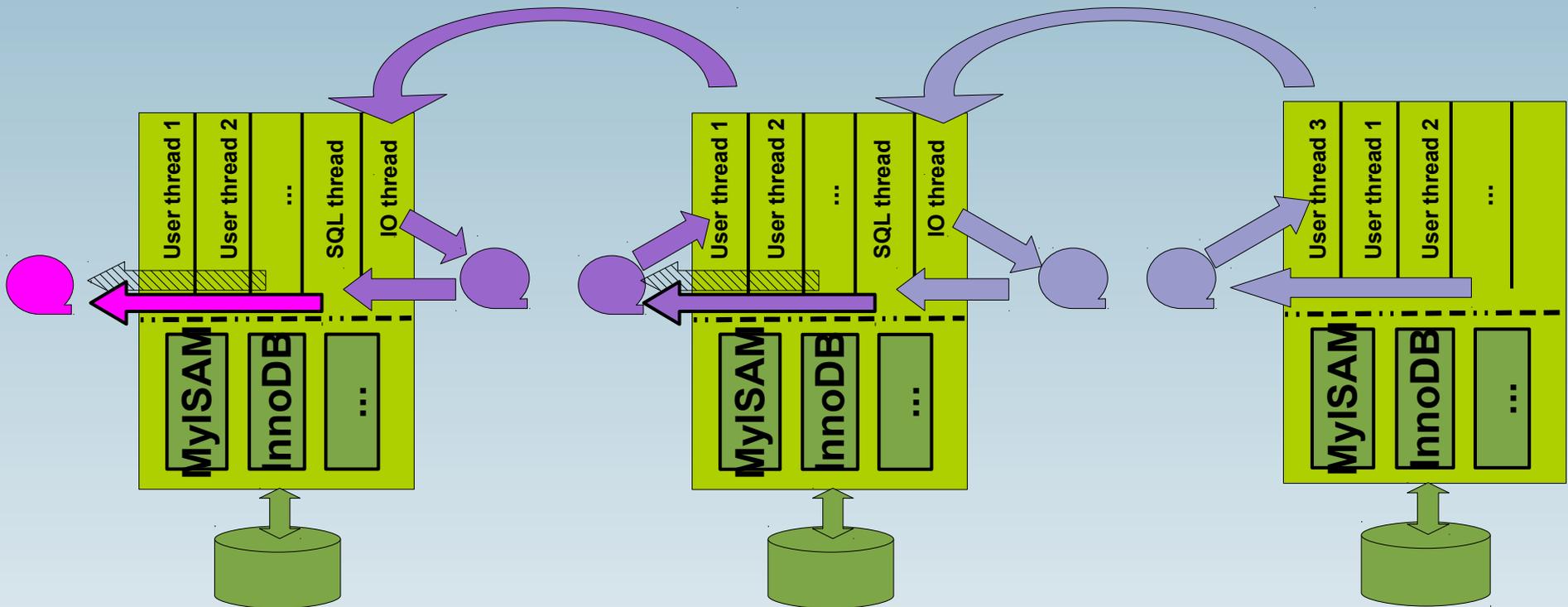
Master:

“log-bin = FILE”, sonst kein Binlog
(keine Master-Funktion)

Typische Anwendungen

- **„High Availability“**
- **Geo-Redundanz**
- **Höhere Lese-Last unterstützen**
(= „read scale-out“)
- **Read-Only-Instanz(en)**
für z.B. Backup oder Reports
- **Verzögerte Replikation ist möglich**
- **Filterung (nach DB oder Tabelle) ist möglich**

Replikations-Kaskade



- Empfehlung: „read-only = 1“ auf Slave
- mehrere Slaves an einem Master möglich

Einträge im Binlog

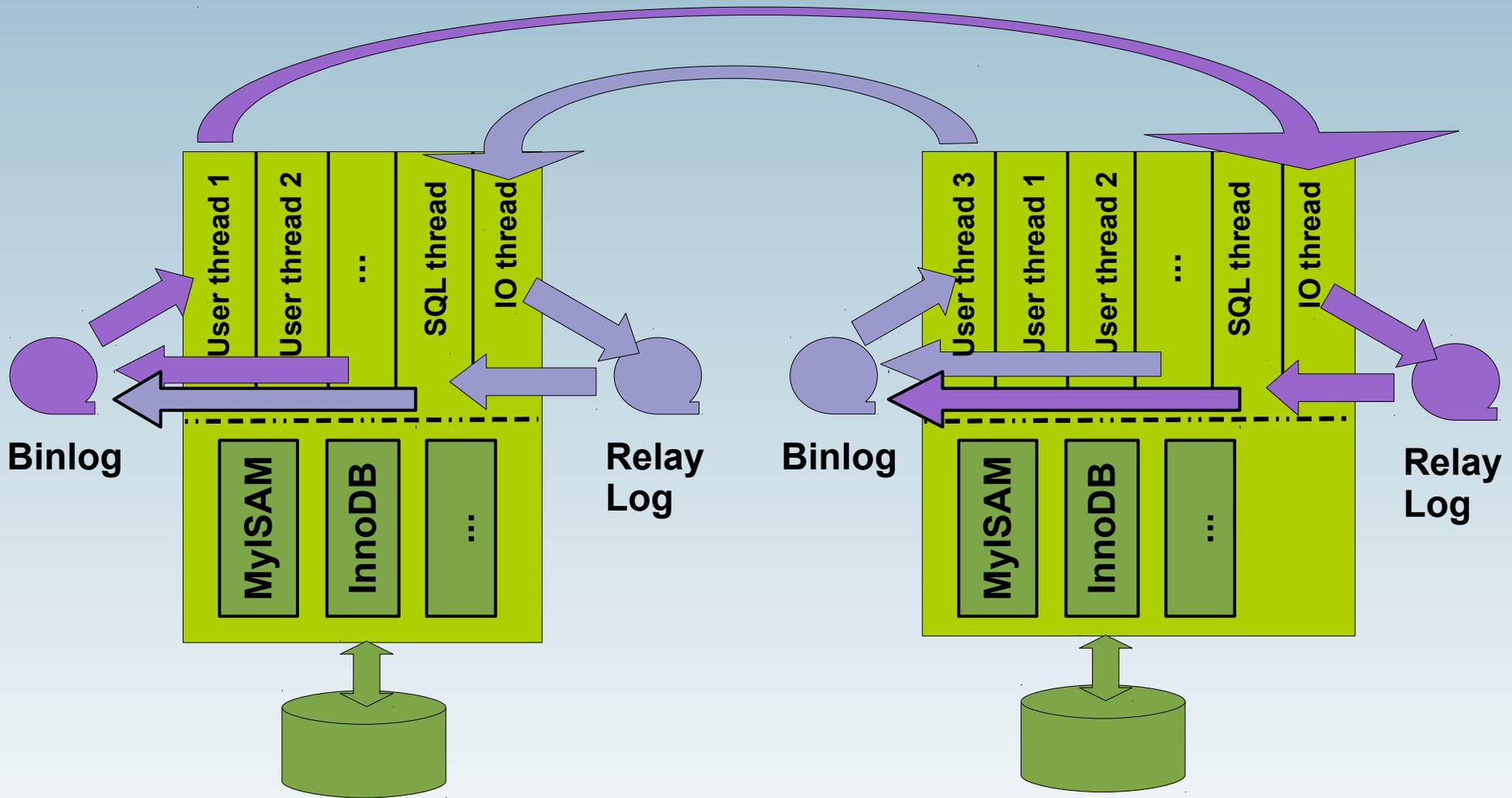
Ursprünglich:

- Identifikation durch Filename und Position
- Replikation: „change master to ...“ mit Host, Port, User, Password, File, Position
- Siehe auch „mysqldump --master-data“

Ab MySQL 5.6:

- GTID = „Global Transaction ID“
- Replikation: „change master to ...“ mit Host, Port, User, Password und „auto_position = 1“

Master-Master-Replikation



- Überlappende Änderungen sind fatal!

Anmerkungen zur Replikation

- **Master-Master ist umstritten, Vorsicht!**
- **Replikation erhöht den Lese-Durchsatz, aber nicht/kaum den Schreib-Durchsatz**
- **Replikation hat File-IO und Netzlast**
- **Format „row“ ist effizienter, aber weniger lesbar**
- **Mit MySQL 5.7 ist Multi-Master-Replikation möglich**
- **Große Installation: booking.com**
- **Lese-Tipp (Giuseppe Maxia, August 2015): datacharmer.blogspot.de**

MySQL Server: Architektur

Binlog

Replikation

➔ **Galera Cluster**

Fazit

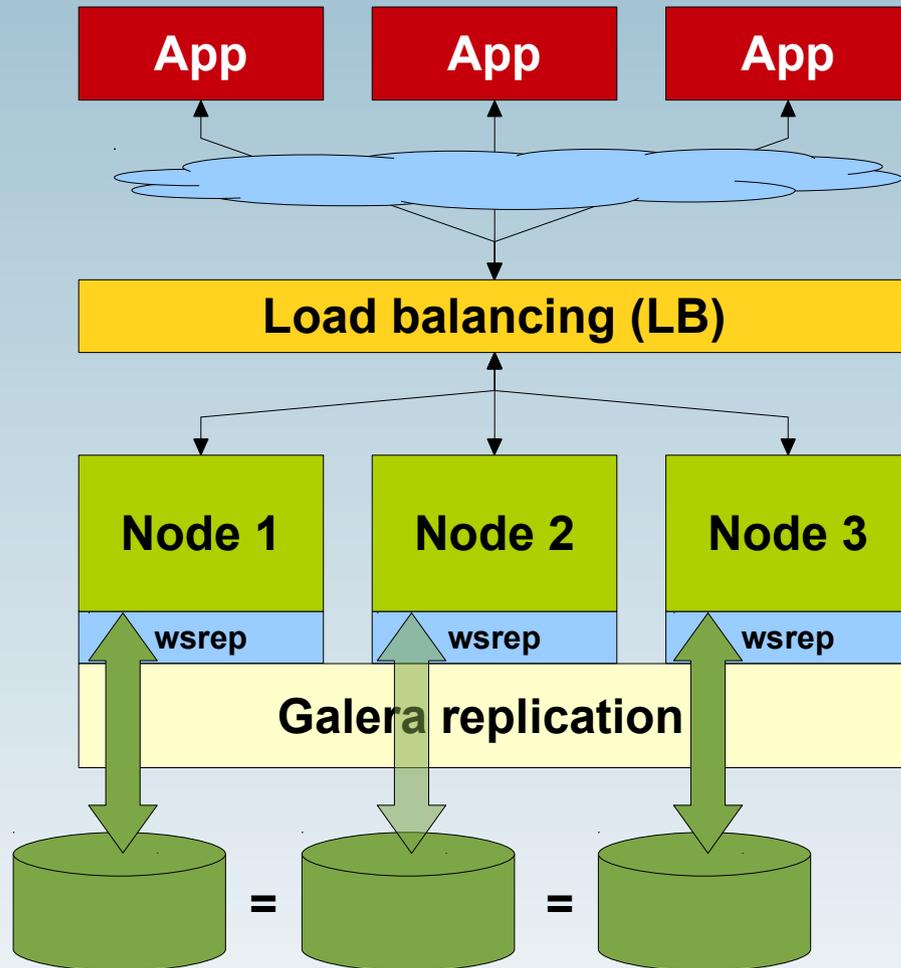
Schwächen der Replikation

- **Asynchron**
- **Asymmetrisch**
- **Nur ein Schreib-Knoten**
- **Paralleles Schreiben verursacht Abbruch**
- **HA braucht Failover nach Knoten-Ausfall**
- **Jeder Knoten ist SPOF, Ausfall erzwingt Struktur-Änderung**
(Erleichterung in 5.7 durch Multi-Master)
- **Dynamische Änderungen sind schwierig**

Bessere Alternative

- **Symmetrischer Cluster**
- **Synchrone Übertragung**
- **Schreibzugriffe überall möglich**
- **Verteilte Konflikt-Analyse und -Behebung**
- **HA durch Kontinuität nach Knoten-Ausfall**
- **Dynamischer Eintritt / Austritt möglich**

Galera Cluster



Inklusive Ausfall-Erkennung und Redirection für HA

“Working Set Replication”

Vorzugsweise eigenes Netz

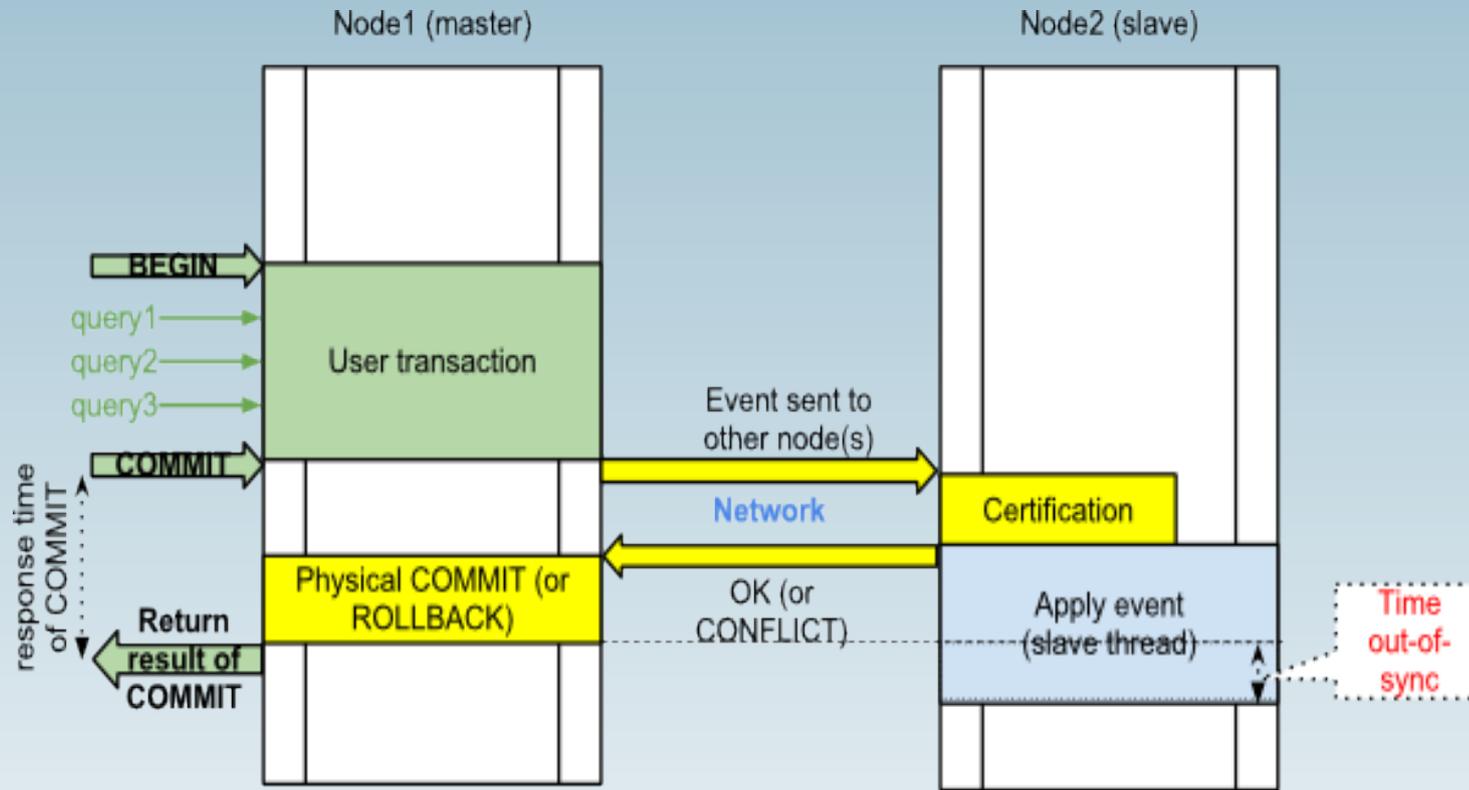
lokale Platten,
jeweils Daten komplett

“shared nothing” Architektur

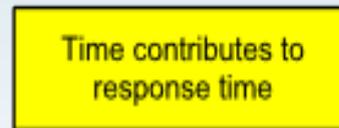
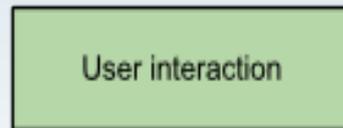
Eigenschaften von Galera (1)

- + Basiert auf InnoDB (wg. Transaktionen und Rollback)**
- + Überträgt auch Benutzer-Definitionen usw.**
- + Quasi-synchrone Übertragung beim Commit, Prüfung auf Konflikt-Freiheit, effizient**
- + Symmetrisch, HA ohne Server-Failover, Quorum**
- + Kein Transaktions-Verlust**
- + Scale-Out für Lesen, auch mehr Schreiben**
- + Dynamischer Eintritt / Austritt möglich, automatische Synchronisation**

Ablauf



Legend:



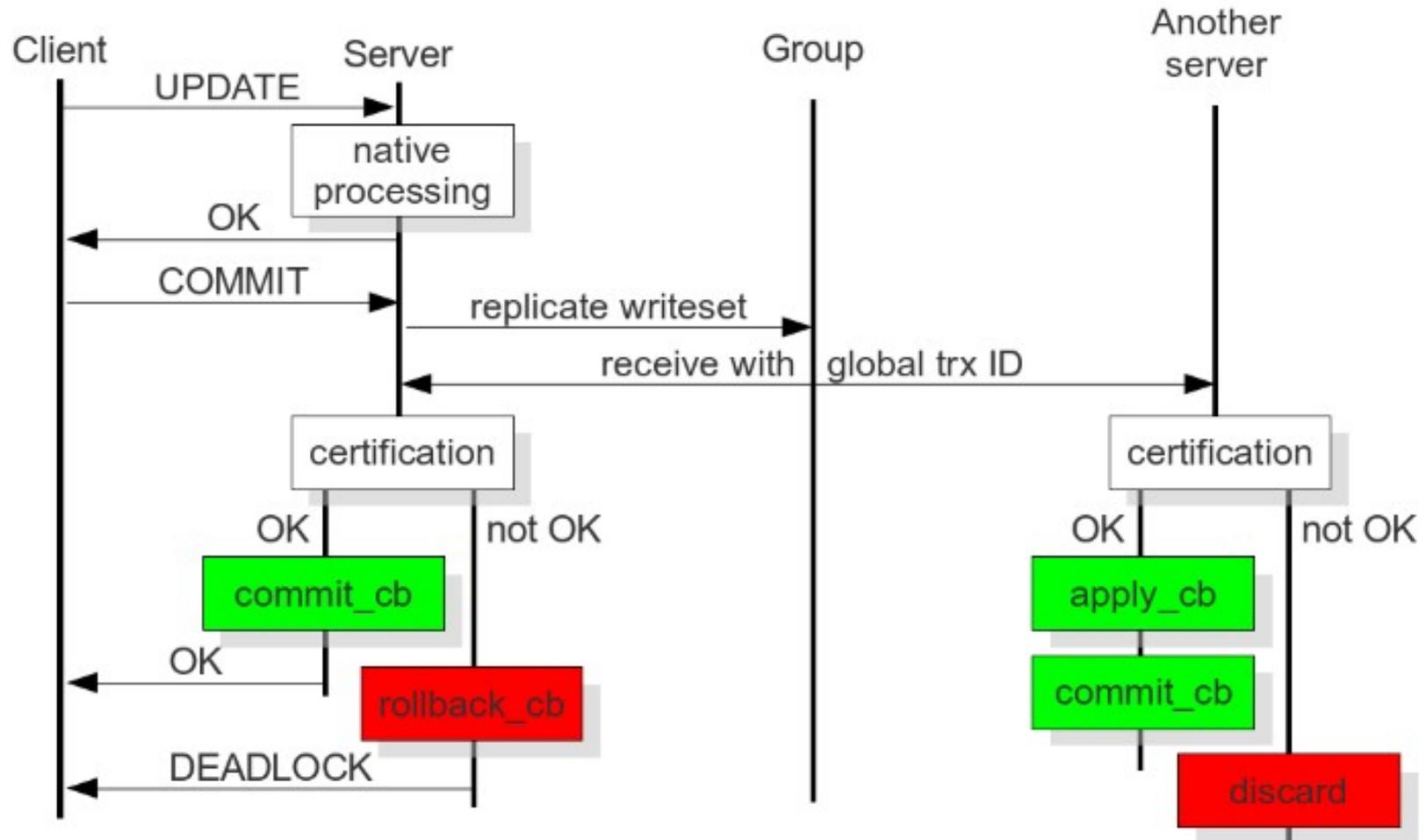
Graph by
Vadim Tkachenko
(Percona):

<http://www.mysqlperformanceblog.com/2012/01/19/percona-xtradb-cluster-feature-2-multi-master-replication/>

Eigenschaften von Galera (2)

- **Patch der MySQL-Quellen (Codership bietet Binaries)**
- **Vorsicht bei Hot Spots (Zeilen)**
- **Späte Konflikt-Erkennung (Prüfung erst bei Commit)**
- **Mindestgröße drei Knoten**
- **Synchronisations-Dauer bei großer DB (mysqldump -> xtrabackup oder rsync)**

Zertifizierung bei Commit



<http://galeracluster.com/documentation-webpages/certificationbasedreplication.html>

MySQL Server: Architektur

Binlog

Replikation

Galera Cluster

➔ **Fazit**

MySQL-Server im Teamwork

- **Alternativen: Replikation oder Cluster**
- **Redundanz bei Maschine und Storage**
- **HA**
- **Scale-Out, besonders für Lese-Last**
- **Instanzen für Reports, Analyse, Backup**
- **Daten lokal verfügbar (Filialen, ...)**

Vergleich (1)

Replikation	Galera
Standard	Zusatzprodukt
Alle Handler	InnoDB
Aufwärts-kompatibel	gleiche Versionen
Mind 2 Knoten	Mind 3 Knoten
HA durch Failover	HA ohne Änderung
<i>Kommunikation:</i>	
Hierarchisch, Kette	symmetrisch, parallel
asynchron	Quasi-synchron
Verzögerung möglich	sofort
Filtern möglich	alles

Vergleich (2)

Replikation	Galera
Lese-Scale-Out	Lese-Scale-Out
<i>Schreib-Knoten:</i> 1* Write	n* Write
Schreiben konst.	Schreiben erhöht
<i>Konflikt lokal:</i>	
Fehler bei Statement	Fehler bei Statement
<i>Schreib-Knoten:</i> n* Write	n* Write
<i>Konflikt verteilt:</i>	
Replikations-Abbruch	Rollback bei Commit

Vergleich (3)

Replikation	Galera
<i>kurze Unterbrechung:</i>	
Replikation fortsetzen	IST (inkrementeller Transfer)
<i>lange Unterbrechung:</i>	
Replikation fortsetzen	SST (kompletter Transfer)
<i>Struktur-Änderung:</i>	
Manuell / Zusatz-Tool	Automatisch / dynamisch
<i>Aufsetzen:</i>	
Schnappschuss, Master bleibt verfügbar	Komplett-Transfer, Donor tlw. Blockiert

Q & A



Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch – Stand 308

- **FromDual bietet neutral und unabhängig:**
 - **Beratung**
 - **Remote-DBA**
 - **Support für MySQL, Galera, Percona Server und MariaDB**
 - **Schulung**

www.fromdual.com/presentations