Various MySQL High Availability (HA) Solutions

Percona Live MySQL Conference, London, Oct 24th and 25th, 2011

Oli Sennhauser

Senior MySQL Consultant at FromDual GmbH

oli.sennhauser@fromdual.com

About FromDual GmbH (LLC)

- FromDual provides neutral and independent:
  - Consulting for MySQL (on-site and remote)
  - Remote-DBA Services / MySQL Operations
  - Premium Support (ex. MySQL Basic and Silver)
  - Training for MySQL
- Consulting partner of the Open Database Alliance (ODBA.org)
- Oracle Silver Partner (OPN)
- More information you can find at:

  http://www.fromdual.com

Contents

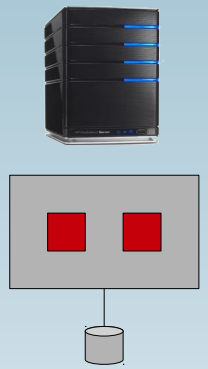
Various MySQL High Availability (HA) Solutions

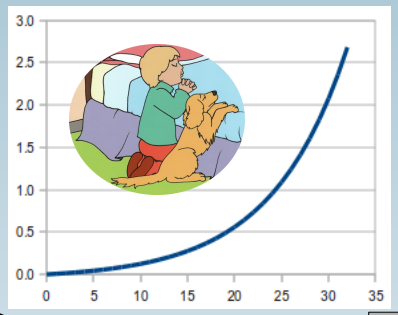
- MySQL Replication / MySQL Scale-Out
- High-Availability with Replication
- Master-Master Replication
- Active/passive fail-over with SAN
- Active/passive fail-over with DRBD
- Galera (synchronous) Replication
- MySQL Cluster
- Tungsten Replicator

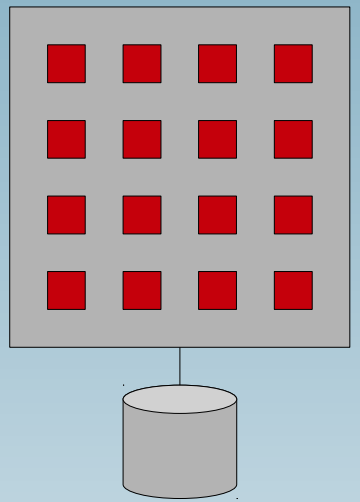
MySQL Scale-Out vs Scale-Up

Scale-Up

- Costs
- MySQL Design
- Physical bottlenecks

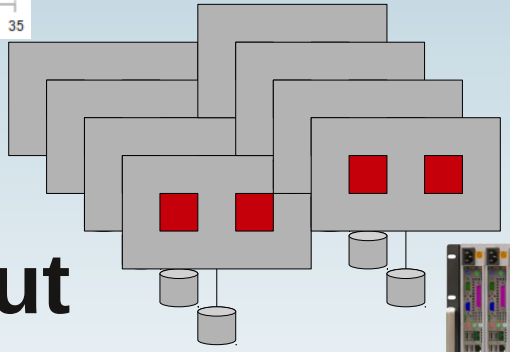




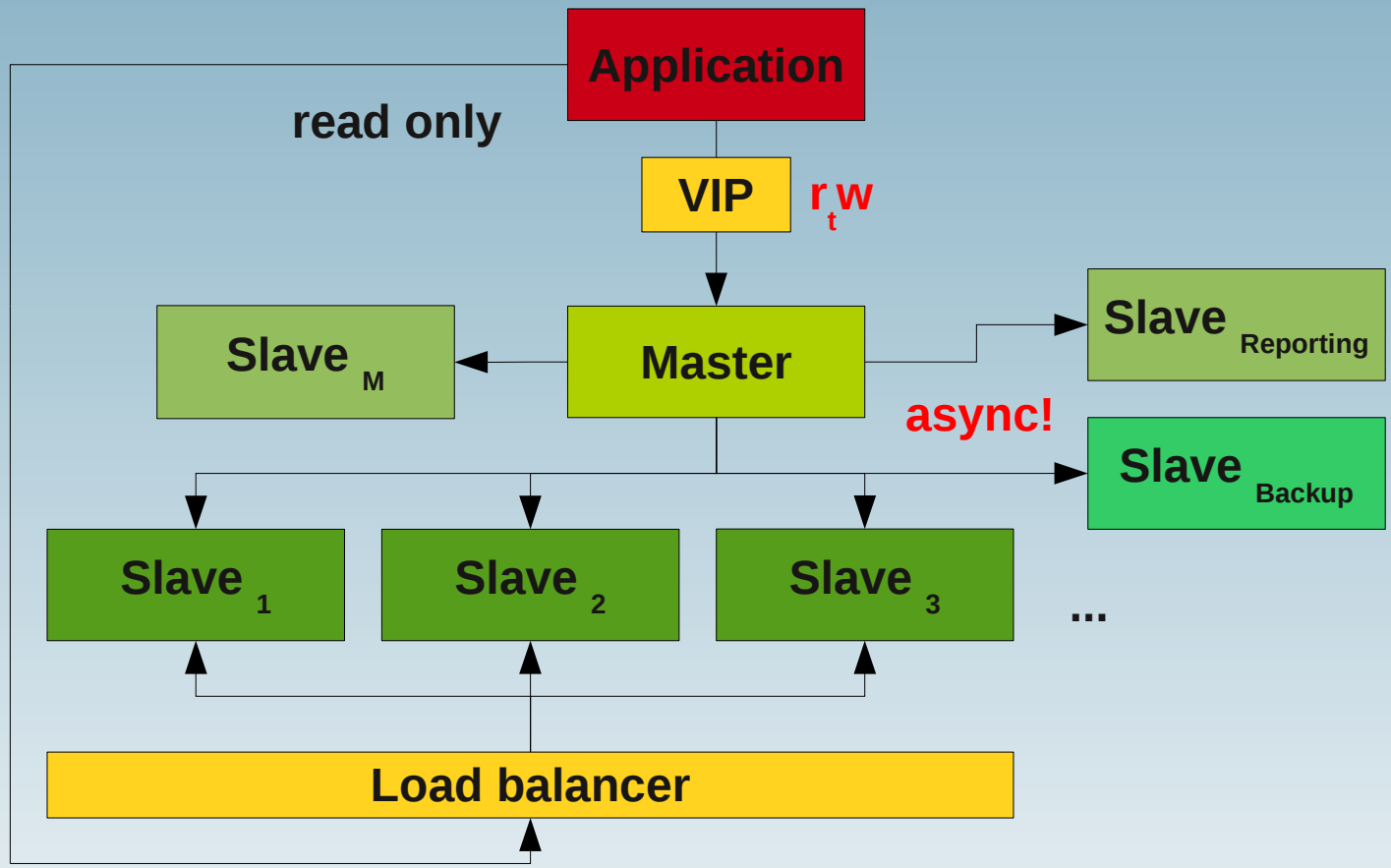


- „Relaxation of Constraints“

Scale-Out

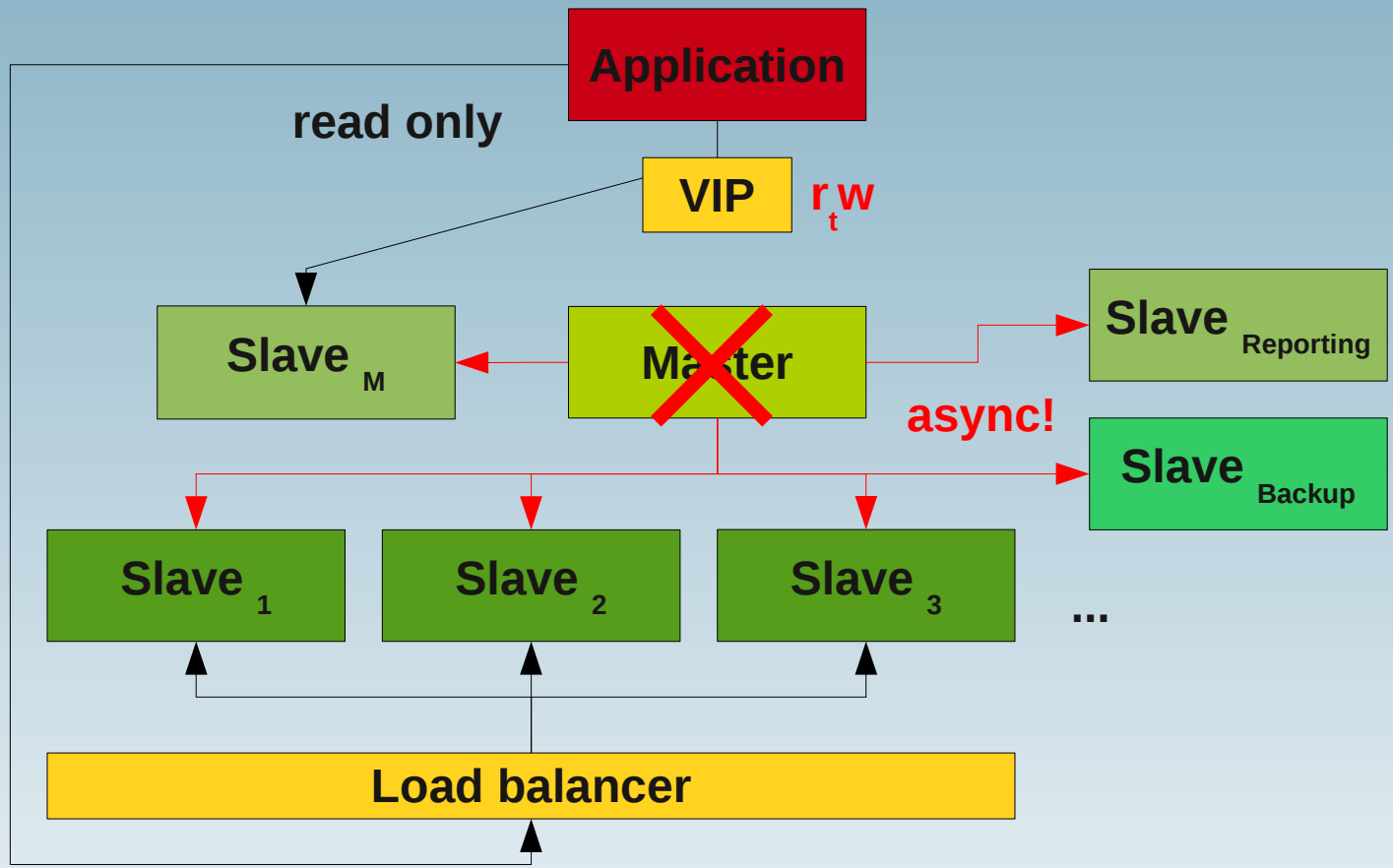


High-Availability with Replication



- **Fail-over?**

Replication fail-over



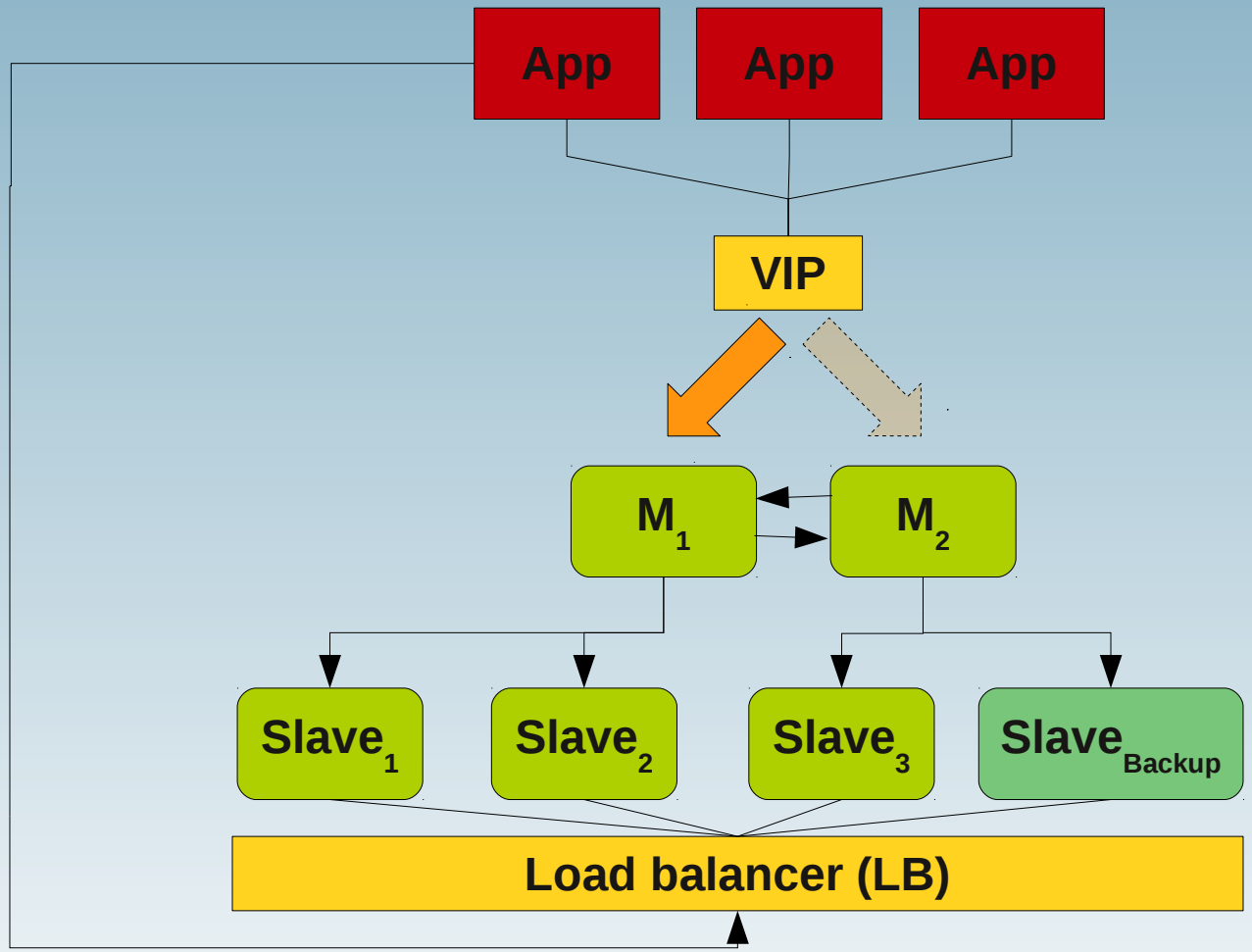
Advantages / Disadvantages

- Simple „standard“ Set-up
- Master is a SpoF! (Single Point of Failure)
- If master fails → which Slave becomes new master?

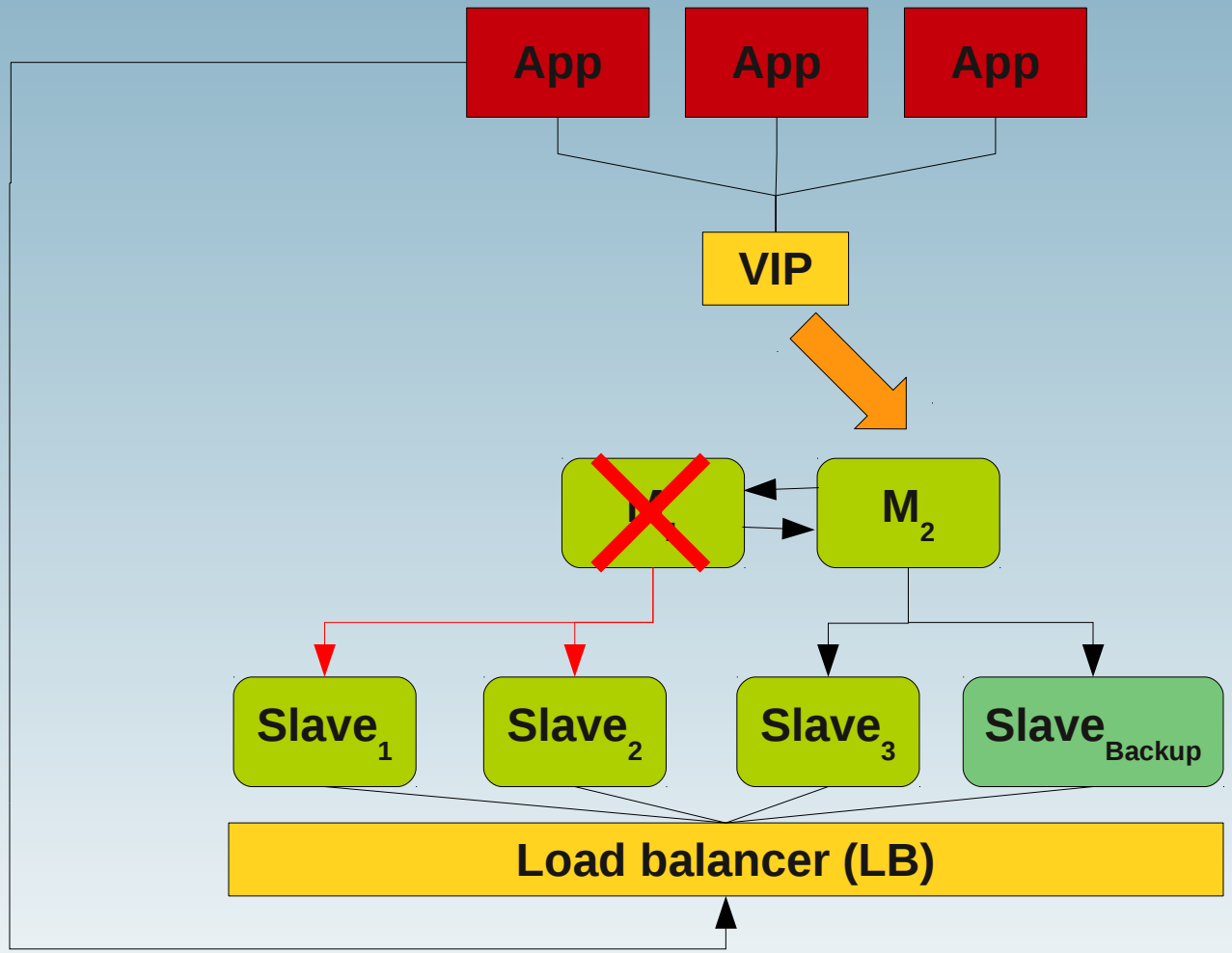
  Switch → a lot of work, delicate!

  There are tools to help (MMM ~~v1~~/v2, MHA, Tungsten, ...)
- Fail-over Site is already warm/hot!
- Works very well if r >> w
- Data inconsistencies (mk-table-check/sync)
- Delay Master/Slave
- Slave lagging (Slave as bottleneck)

Master-Master Replikation



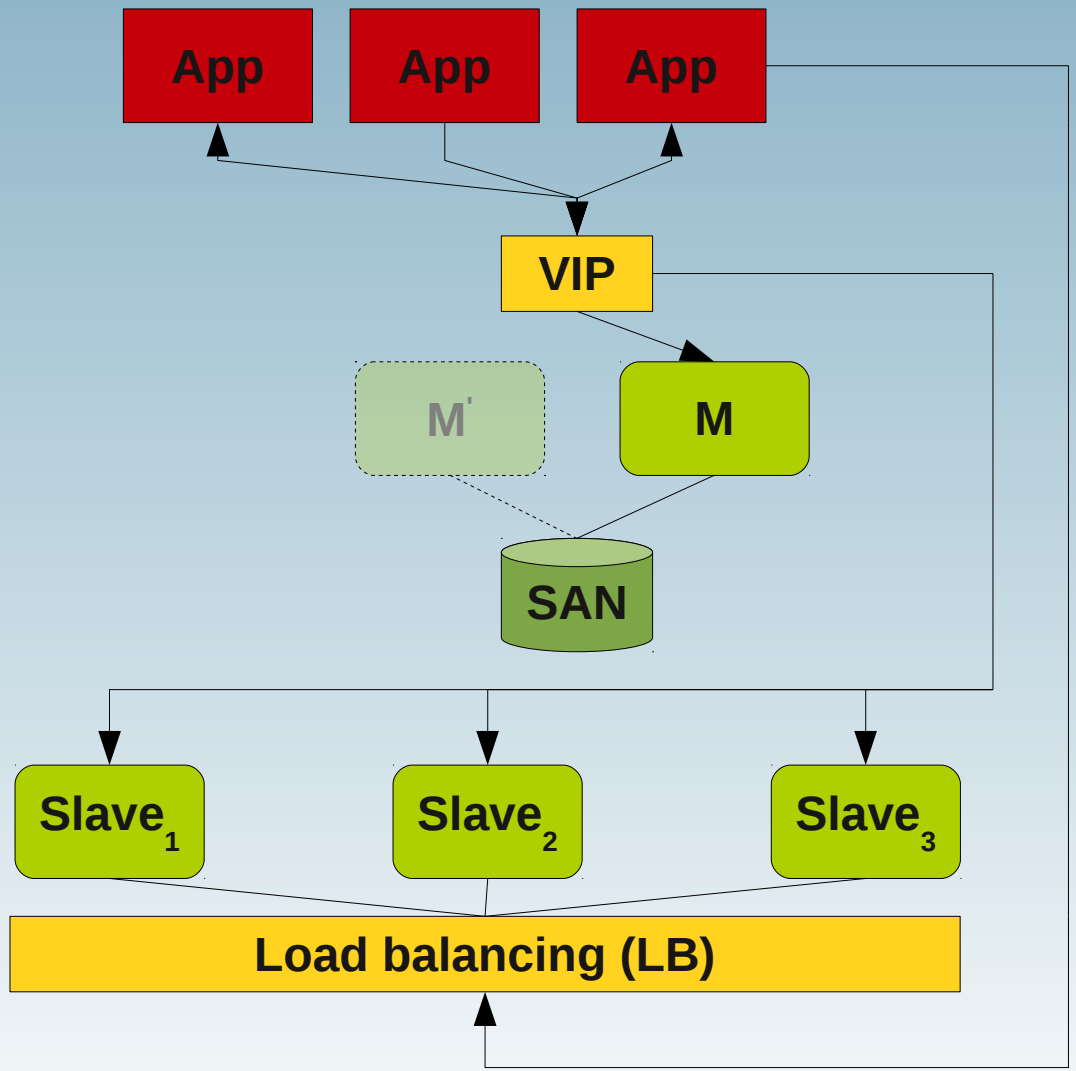
Master-Master Replication



Advantages / Disadvantages

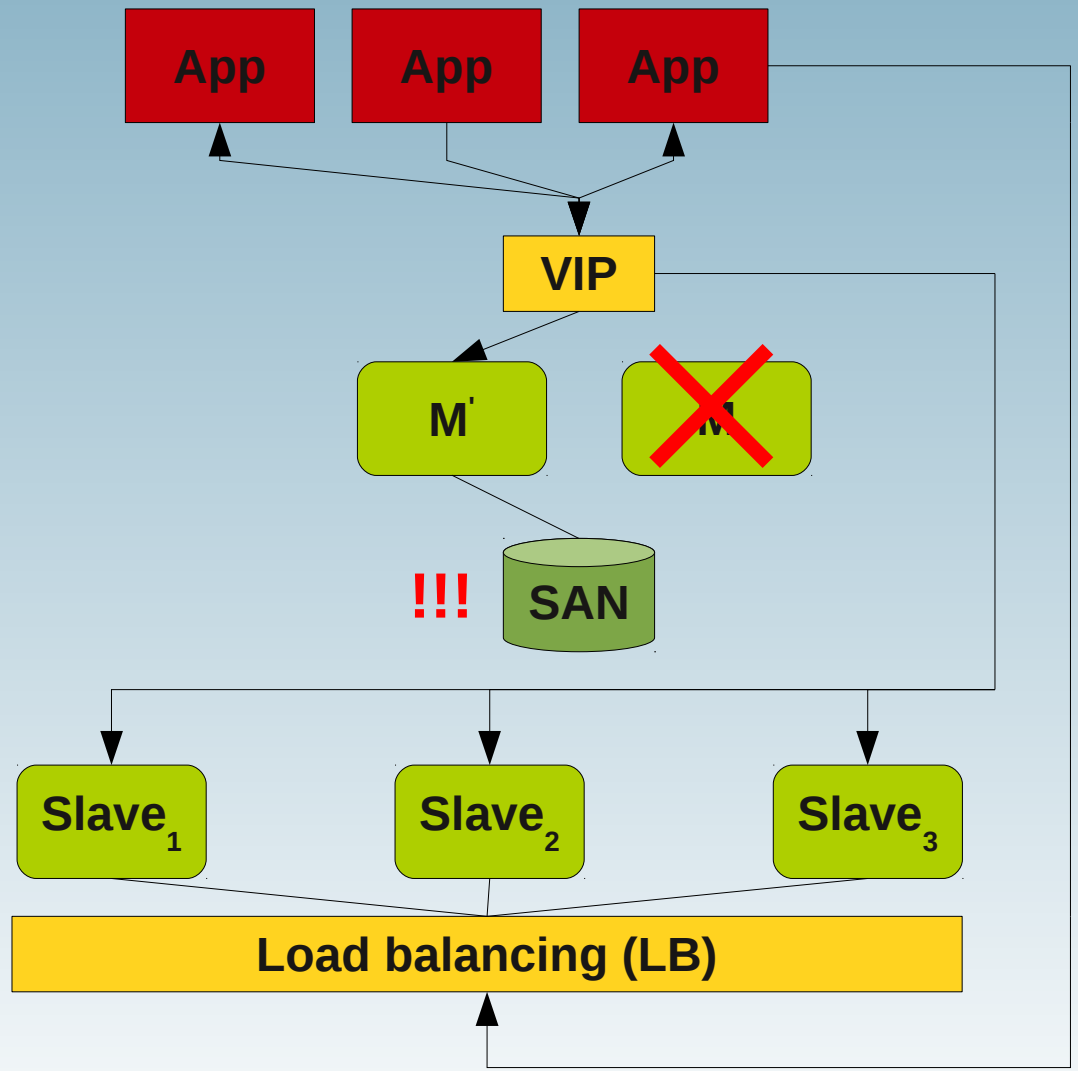
- Only slightly more complex than Master/Slave
- Careful when writing on both Masters!
- For a “balanced” system at least 2 Slaves are needed
- You will NOT get more I/O throughput!
- Data in-consitency possible because of asynchronous replication
- Fail-over Site is already warm/hot!
- Works very well if r >> w
- Data inconsistencies (mk-table-check/sync)
- If Master fails, half of the Slave are out of sync!
- A little more complicated to (re-)set-up
- Delay Master/Slave
- Slave lagging (Slave as bottleneck)

Active/passive fail-over with SAN



Active/passive fail-over with SAN

- SPOF!

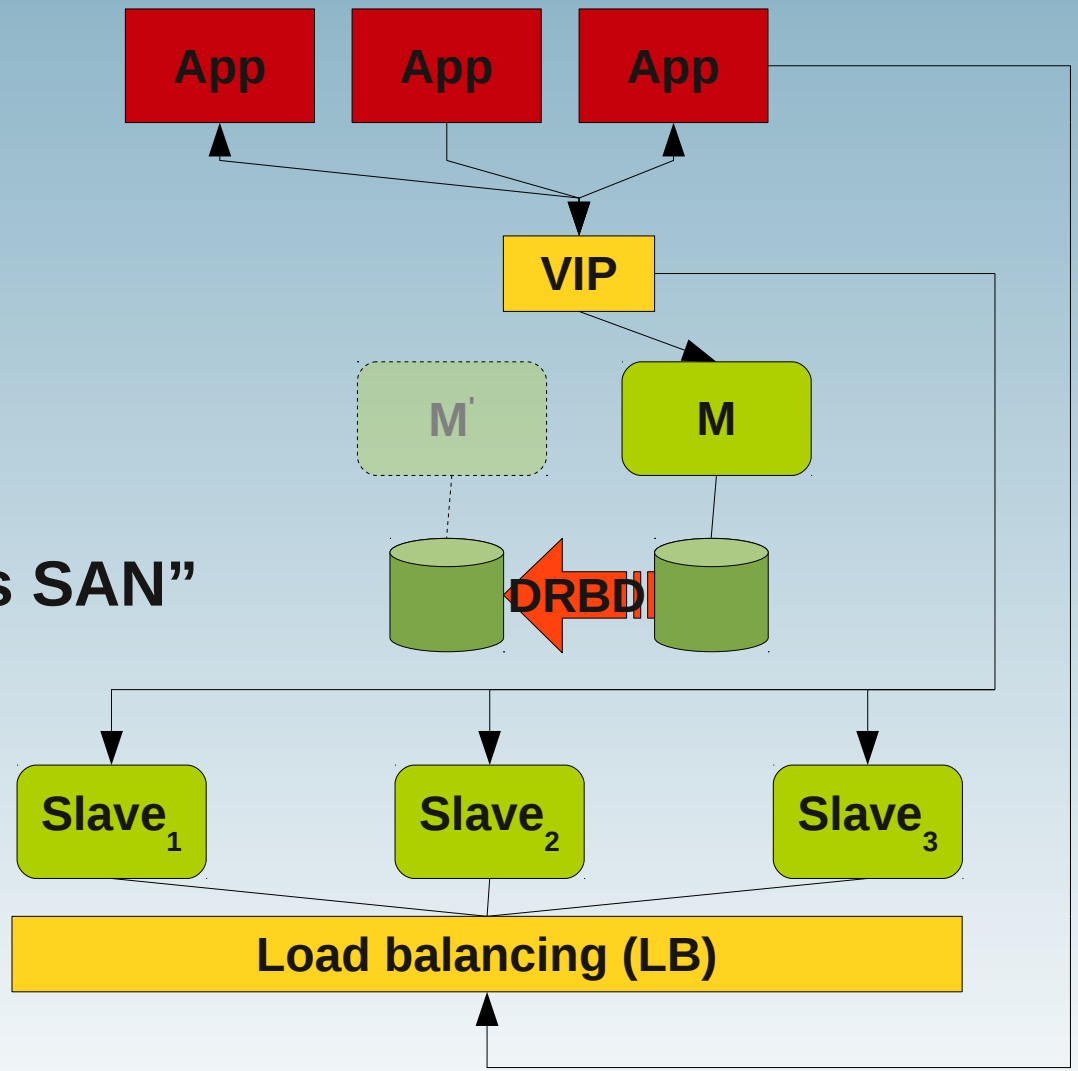


Advantages / Disadvantages

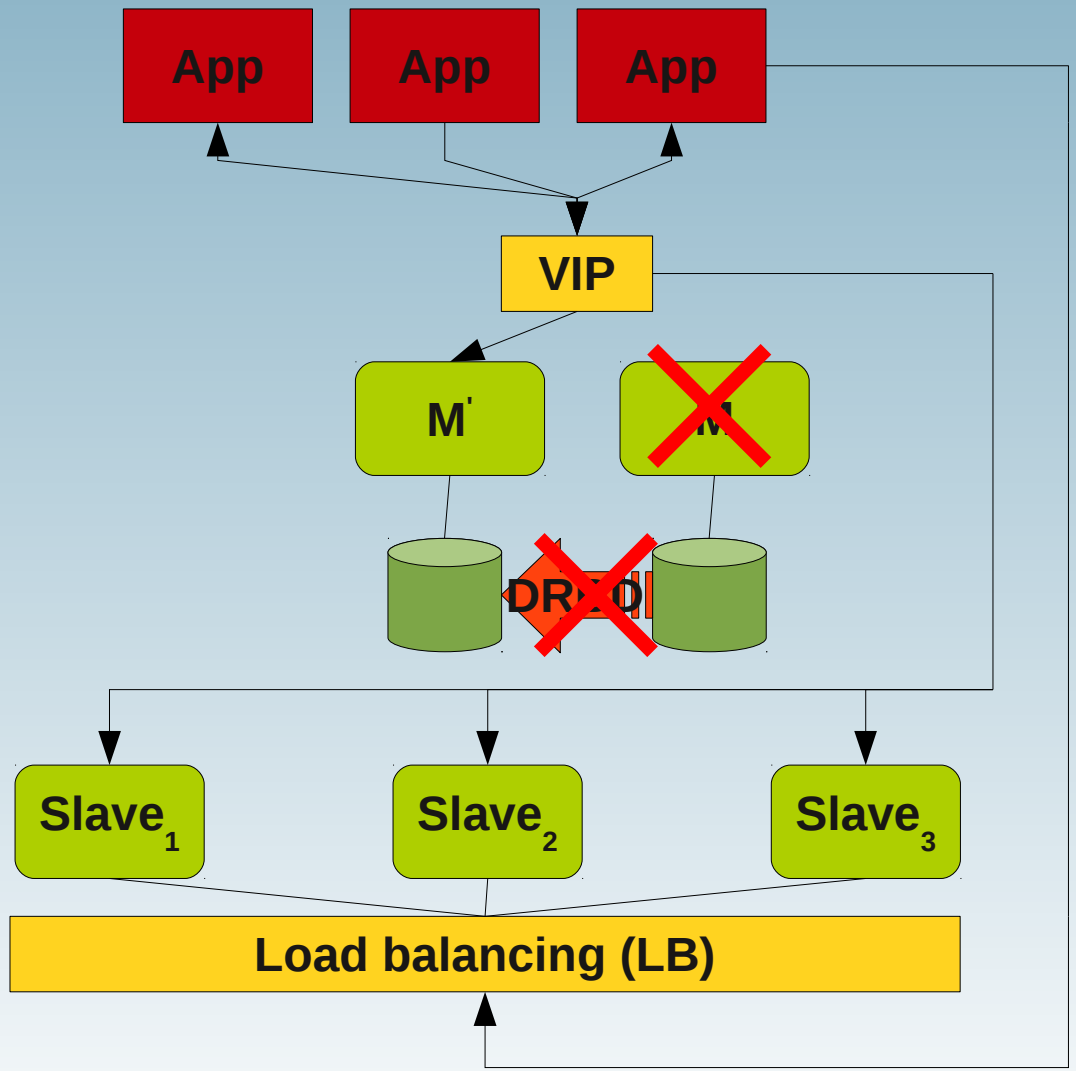
- Synchronous replication
- I/O throughput depends on SAN (I/O system)
- No data IN-consistencies possible
- SAN is a SpoF!
- Expensive if SAN is not available yet.
- SAN's are not easy to handle!
- Fail-over Site is still cold!
- Half of the hardware is idling
- Only one possible Data source
- Slaves are automatically and properly fail-overed
- Far more complex to set-up

Active/passive fail-over with DRBD

“Poor man's SAN”



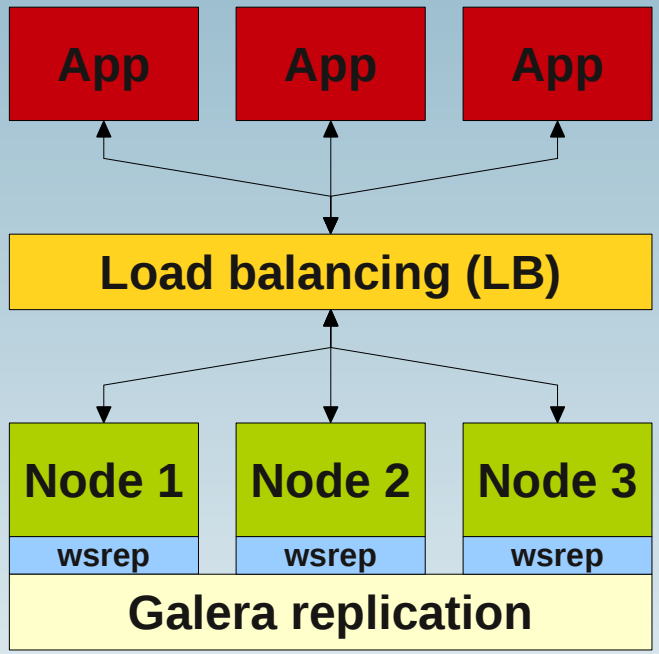
Active/passive fail-over with DRBD



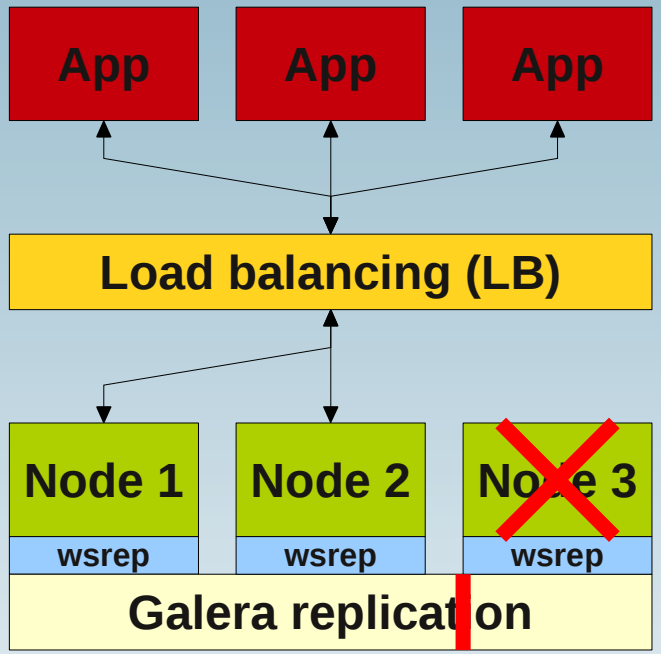
Advantages / Disadvantages

- Synchronous replication
- No data IN-consistencies possible
- I/O throughput lower
- Slaves are automatically and properly fail-overed
- Fail-over Site is still cold!
- Half of the hardware is idling
- Only one possible Data source
- Far more complex to set-up

Galera (synchronous) Replication



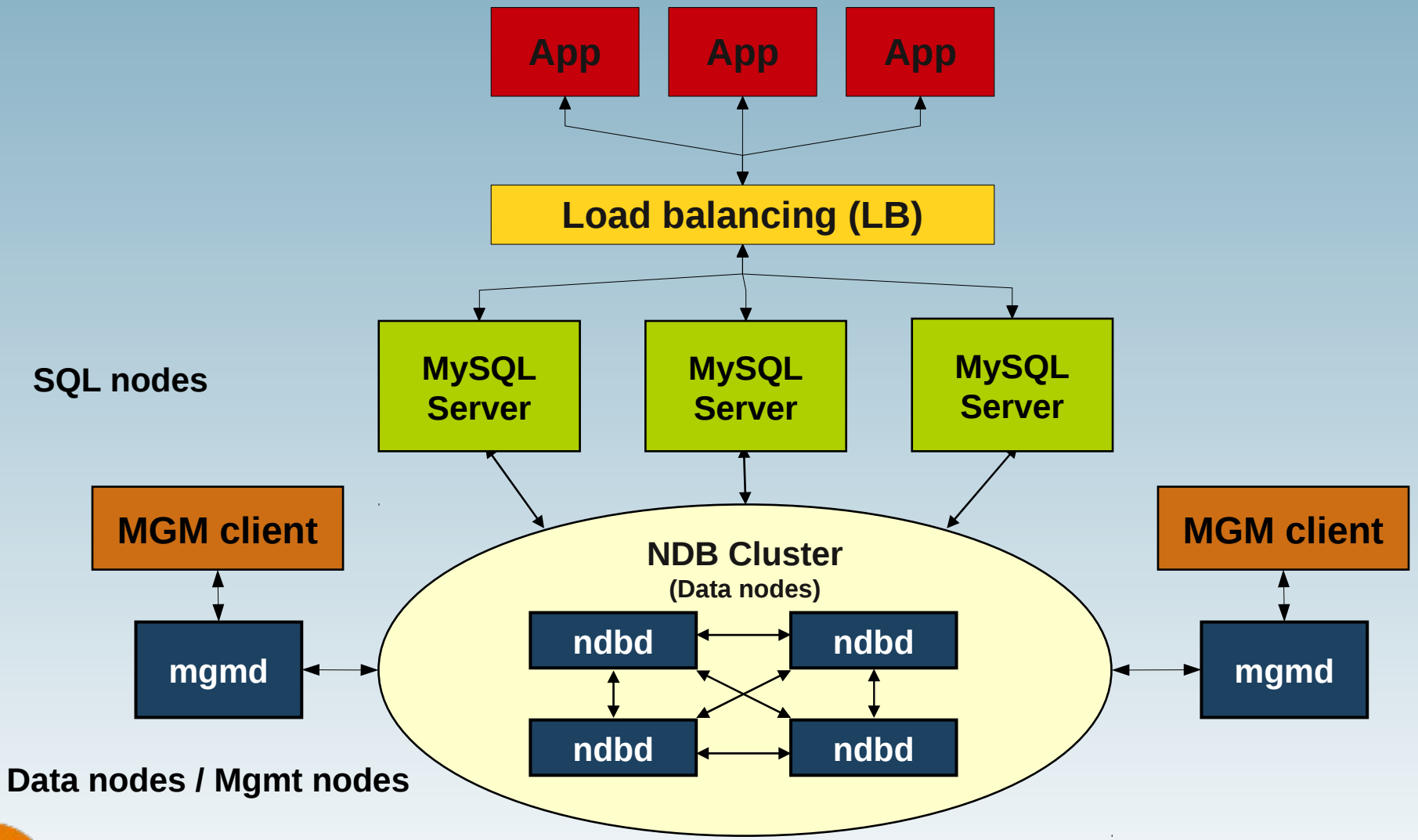
Galera (synchronous) Replication



Advantages / Disadvantages

- Synchronous replication
- Based on InnoDB SE (other SE theoretically possible)
- Active-active real multi-master topology
- Read and write to any cluster node
- Automatic membership control
- True parallel replication, on row level
- No slave lag
- No lost transactions
- Read AND write scalability (Read Scale-Out!)
- Patch off MySQL binaries (Codership provides binaries)
- Be aware of Hot Spots on rows
- Higher probability of dead locks
- Full sync blocks for writing → 3 nodes
- Initial sync for very big databases (>>50 Gbyte) with mysqldump

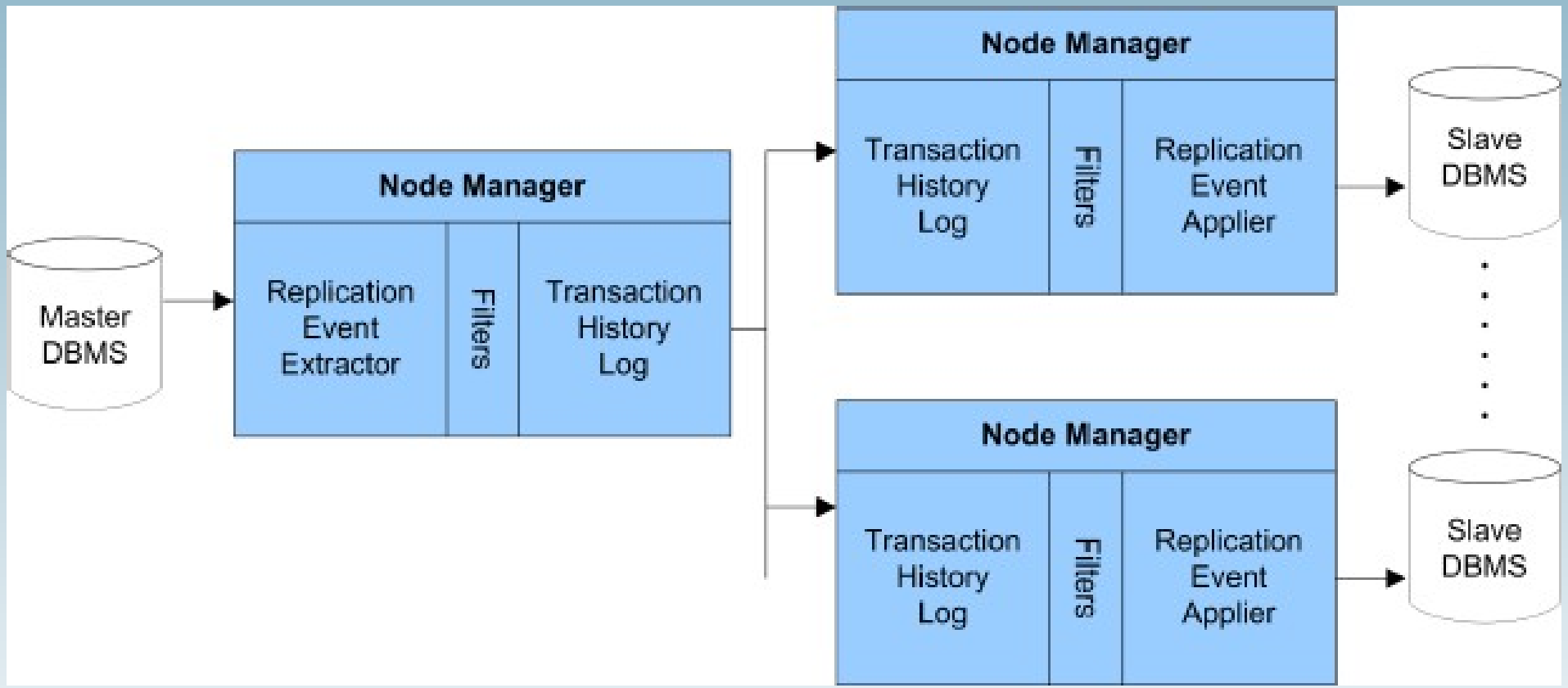
MySQL Cluster



Advantages / Disadvantages

- Synchronous replication
- No data IN-consistencies possible
- Extremely high throughput (if done correctly)
- Good for read AND write
- New beast to tame (MySQL Cluster != MySQL!)
- More complex to set-up and operate than normal MySQL
- Not for disaster fail-over (<= 7.1)
- Bad for complex queries (Joins, <= 7.1)
- At least 3 machines (better 4) are need.
- High demand on RAM and Network
- Only one possible Data source

Tungsten Replicator



Advantages / Disadvantages

- Similar to MySQL Replication
- Introduces Global Transaction-ID
- Asynchronous Replication
- Bases on MySQL Binary Log
- For: MySQL, PostgreSQL, Oracle and Drizzle
- Requires Java and Ruby?
- Multi Source Replication
- Failover should be easier to handle

Questions ?

Discussion ?