MySQL Hochverfügbarkeitslösungen

Lenz Grimmer ‹lenz@grimmer.com›

http://lenzg.net/ | Twitter: @lenzgr

2010-09-23 | OpenSource Trends Days | Germany

$ whoami

1998

2002

2008

2010

Agenda

- Konzepte & Aspekte
- MySQL Replikation
- Disk replikation (DRBD)
- Shared Storage (SAN/NAS)
- Heartbeat/Pacemaker (Linux-HA)
- MySQL Cluster
- Weitere HV-Applikationen

Einzelner MySQL Server

MySQL Client

SQL (SELECT, INSERT, UPDATE, DELETE)

MySQL Storage Engines (InnoDB, MyISAM, PBXT...)

Disk Storage (XFS, ReiserFS, JFS, ext3...)

Warum Hochverfügbarkeit?

- Irgendetwas kann und wird ausfallen
- Notwendigkeit von Wartungsarbeiten
- Ausfallzeiten sind teuer
- Vorher einplanen: HV nachträglich hinzuzufügen ist schwierig

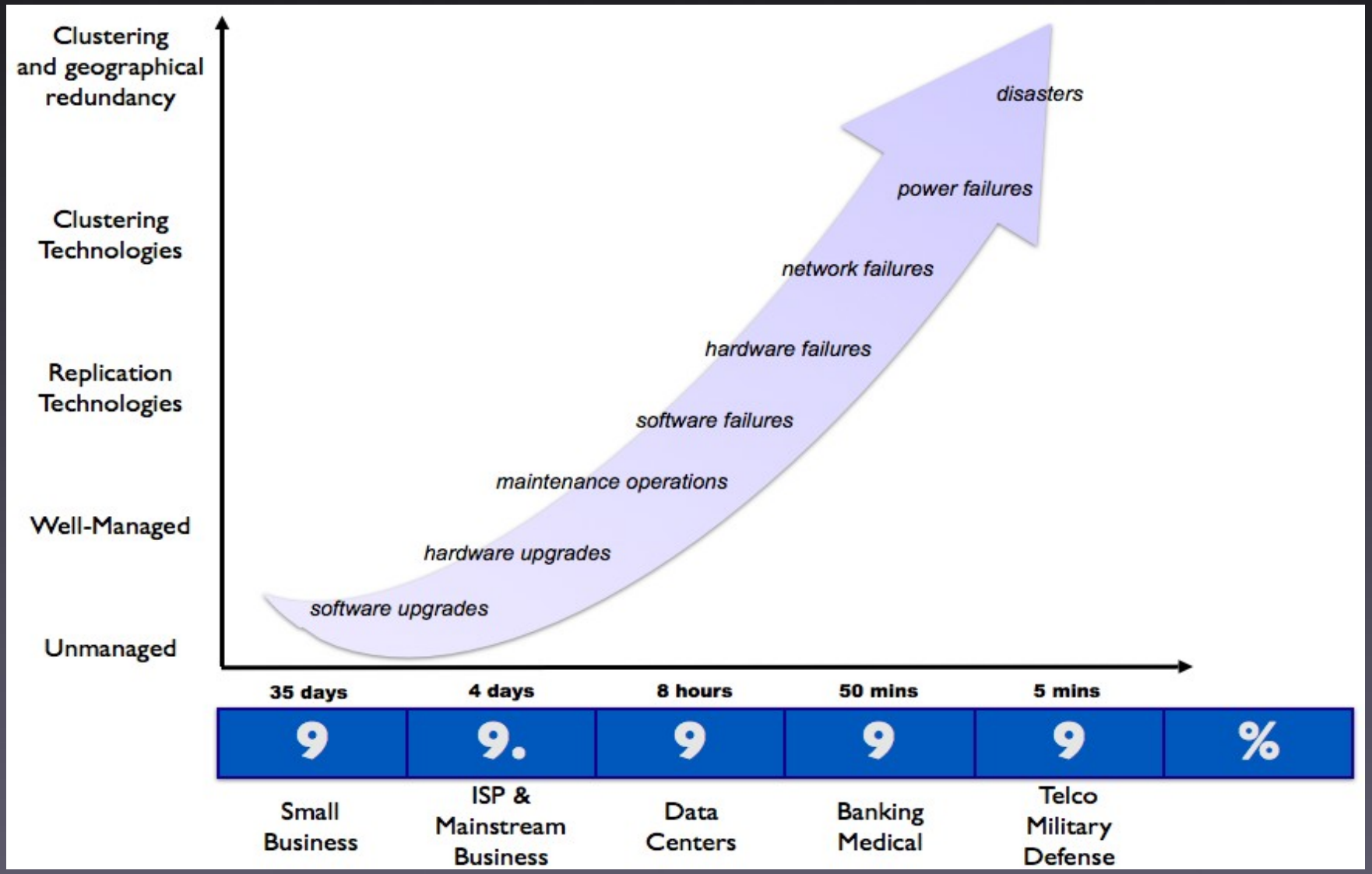
Eliminierung des SPOF

- Was ausfallen wird
  - Festplatten
  - Lüfter
- Was ausfallen kann
  - Absturz der Applikation
  - OOM-Situationen, Kernel-Panics
  - Netzwerkverbindungen/-kabel
  - Stromversorgung

Was ist HV-Clustering?

- Redundanz
- Ein Dienst fällt aus → ein anderer übernimmt
- Übernahme der IP Addresse und des Dienstes
- Failover vs. Failback vs. Switchover
- **Nicht geeignet** für mehr Leistung/Durchsatz

Hochverfügbarkeitslevel



Regeln zur Hochverfügbarkeit

- Auf Ausfälle vorbereitet sein
- Sicherstellen daß keine **wichtigen** Daten verloren gehen
- Keep it simple, stupid (KISS)
- Komplexität ist der Feind der Zuverlässigkeit
- **Automatisieren** was sinnvoll ist
- Häufig **Tests** durchführen!

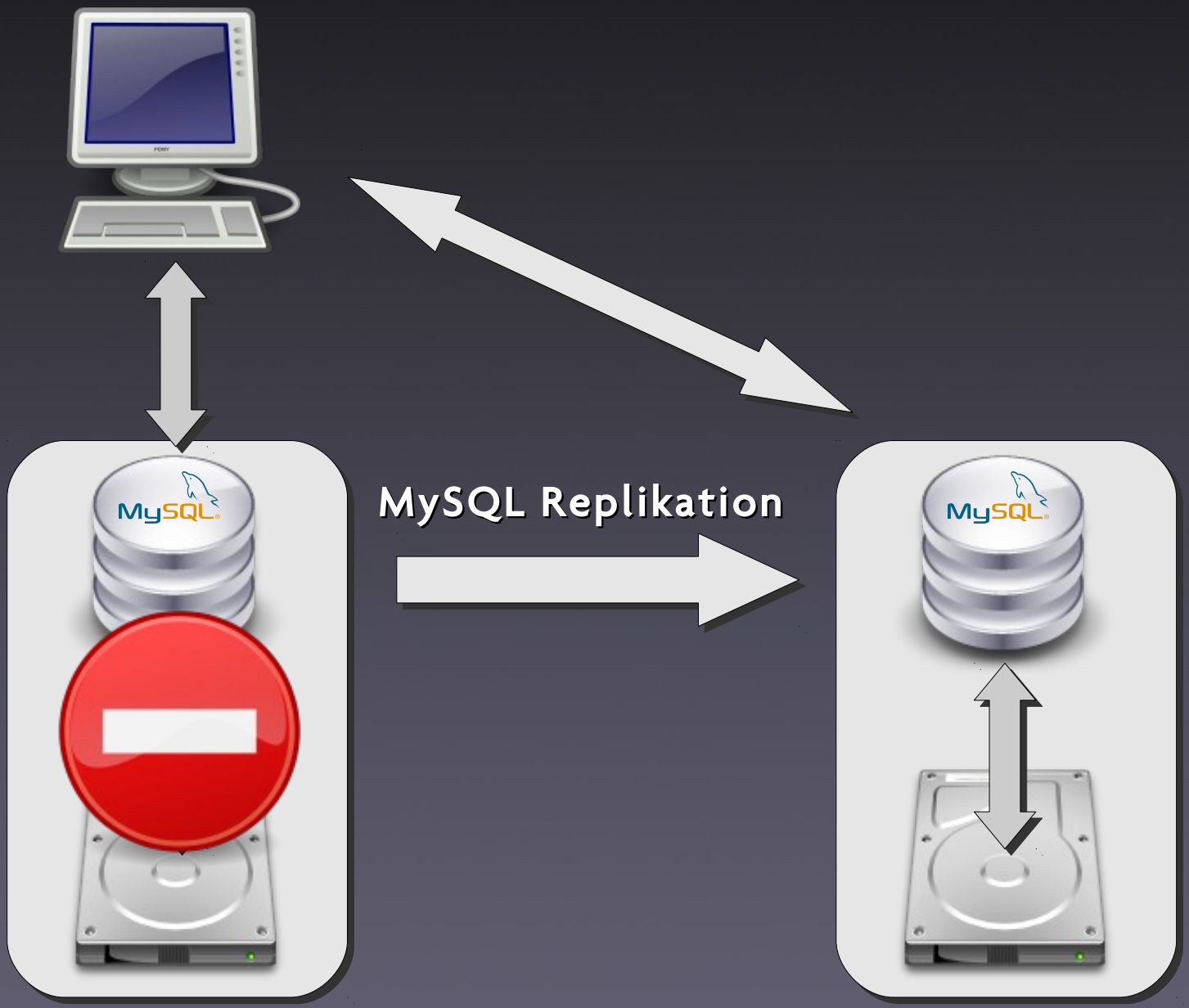
HA Komponenten

- Heartbeat
  - Überwachung der hochverfügbar zu machenden Dienste
  - Kann einzelne Server, Dienste, Netzwerkverbundungen etc. prüfen
- HA Monitor
  - Konfiguration der Dienste/Applikationen
  - Regelt ordnungsgemäßen Start und Beenden
  - Erlaubt manuelle Übernahme
- Replikation oder Shared storage (SAN)

Split-Brain

- Ausfall der Kommunikation → separate Cluster-Partitionen
- “Split-brain” Situation: mehr als eine Partition will die Kontrolle über ihren Teil des Clusters übernehmen
- http://linux-ha.org/BadThingsWillHappen
- Vermeidung mittels “Fencing“ oder Moderation/Arbitrierung

Redundanz mit MySQL Replikation



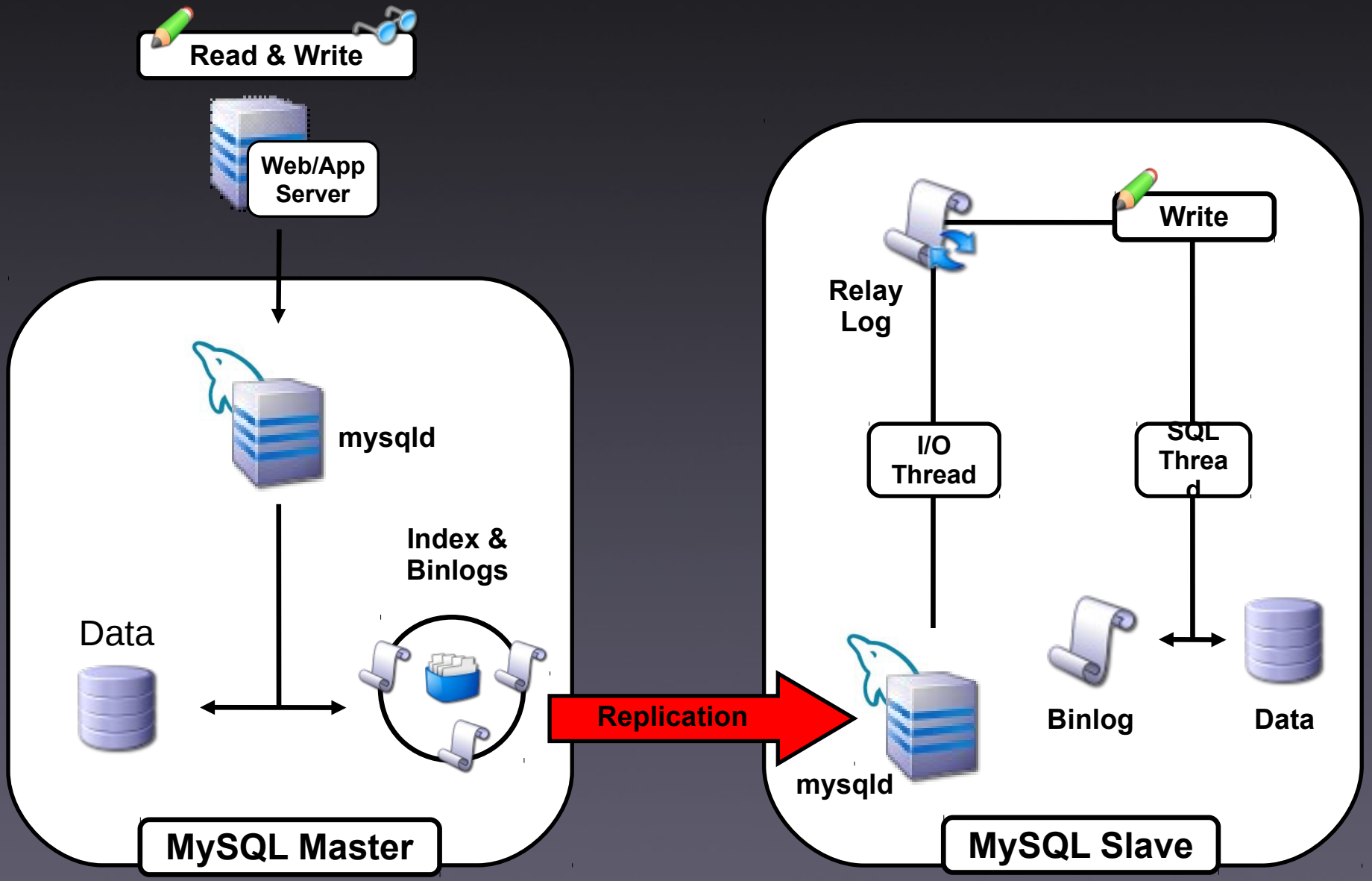
MySQL Replikation

- Unidirektional
- Anweisungs- oder zeilenbasiert (MySQL 5.1)
- Bestandteil des MySQL-Servers
- Einfach zu bedienen und einzurichten
- Ein Master, viele Slaves
- Asynchron – Slaves können nachlaufen
- Neu in MySQL 5.5: Semisync Replication

MySQL Replikation (2)

- Master verwaltet Binärlogs & index
- Replikation auf Slave ist (noch) single-threaded
  http://forge.mysql.com/wiki/ReplicationFeatures/ParallelSlave
- Kein automatisiertes Fail-over
- Kein Heartbeat, kein Monitoring

MySQL Replikation - Überblick



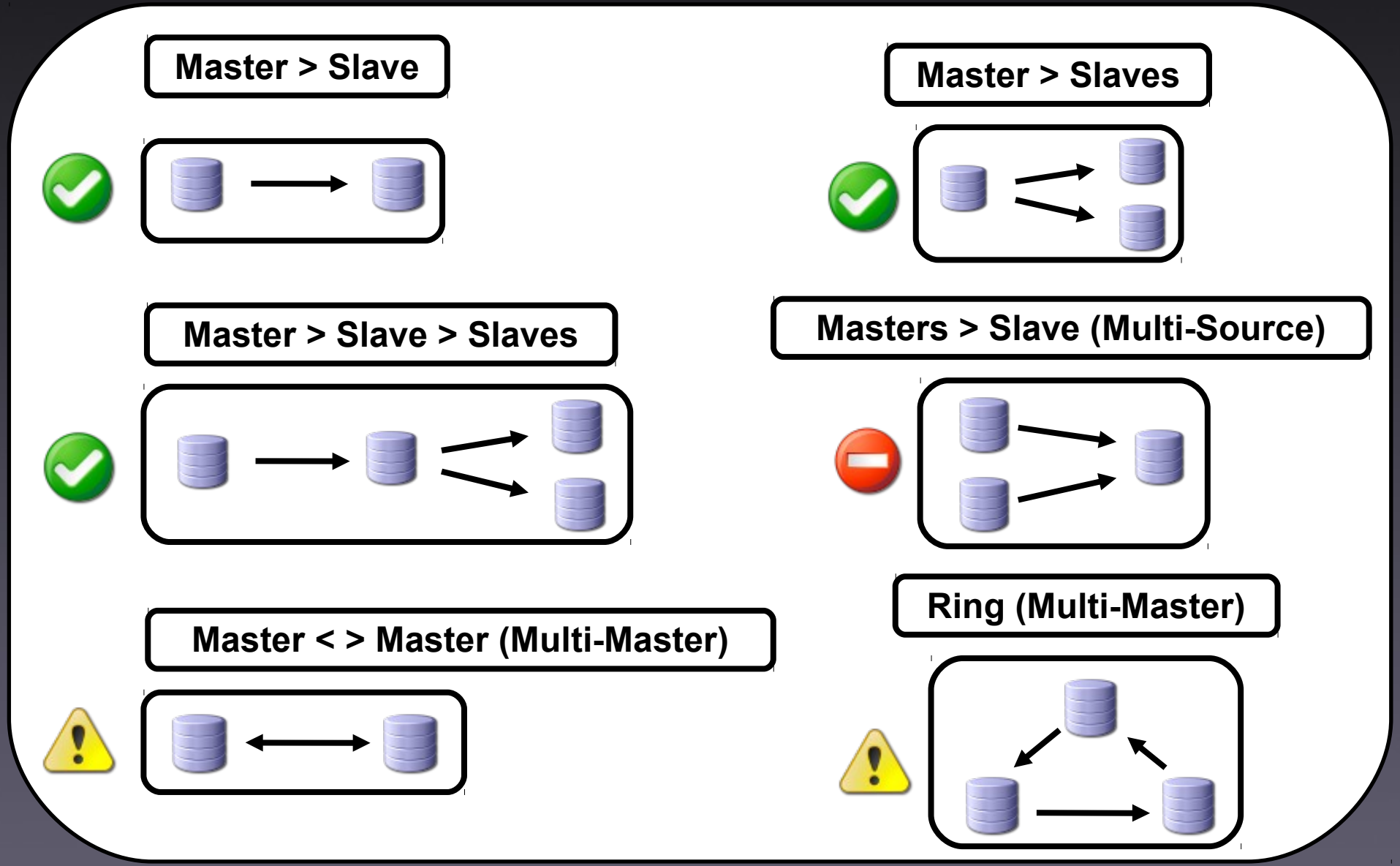
Anweisungsbasierte Replikation

- Pro
  - Bewährt (seit MySQL 3.23 verfügbar)
  - Kleinere Logdateien
  - Auditing der tatsächlichen SQL-Anweisungen
  - Kein Primärschlüssel bei replizierten Tabellen erforderlich
- Kontra
  - Nicht-deterministische Funktionen und UDFs
  - LOAD_FILE(), UUID(), USER(), FOUND_ROWS() (RAND() and NOW() gehen)

Zeilenbasierte Replikation

- Pro
  - Alle Veränderungen können repliziert werden
  - Verfahren ähnlich zu anderen DBMSen
  - Erfordert weniger Locks für bestimmte INSERT, UPDATE oder DELETE Anweisungen
- Kontra
  - Mehr Daten müssen gelogged werden
  - Logfile-Größe (Auswirkungen auf Backup/Restore)
  - Replizierte Tabellen benötigen expliziten Primärschlüssel
  - Mögliche Ergebnis-Differenzen bei Bulk-INSERTs

Replikationstopologien



Master-Master-Replikation

- Zwei Server: sowohl Master als auch Slave zueinander
- Vereinfacht Failover
- **Nicht geeignet** um Schreiblast zu verteilen
- **Nicht** auf beide Master schreiben!
- Sharding oder Partitionierung (z.B. MySQL Proxy) eignen sich besser

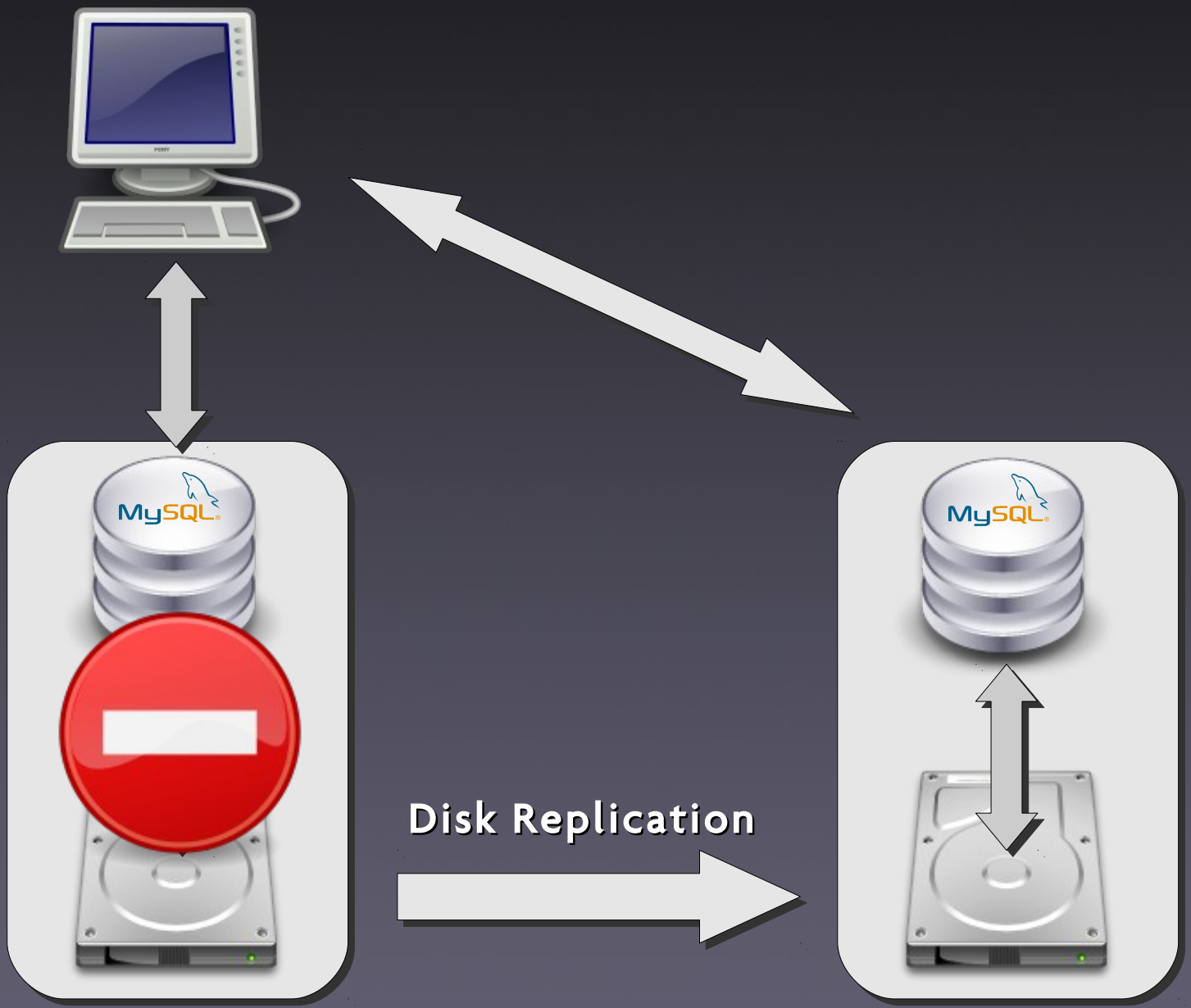
MySQL Replikation als eine HV-Lösung

- Was passiert wenn der Master ausfällt?
- Was passiert wenn der Slave ausfällt?
- Ist das Hochverfügbarkeit?

Replikation & Hochverfügbarkeit

- Kombiniert mit Heartbeat
- Übernahme der virtuellen IP-Adresse
- Slave wird zum Master befördert
- Nebeneffekt: Lastverteilung & Backup
- Failback kompliziert
- Keine automatische Konfliktauflösung
- Korrektes Failover muß gescriptet werden

Redundanz mit Festplatten-Replikation

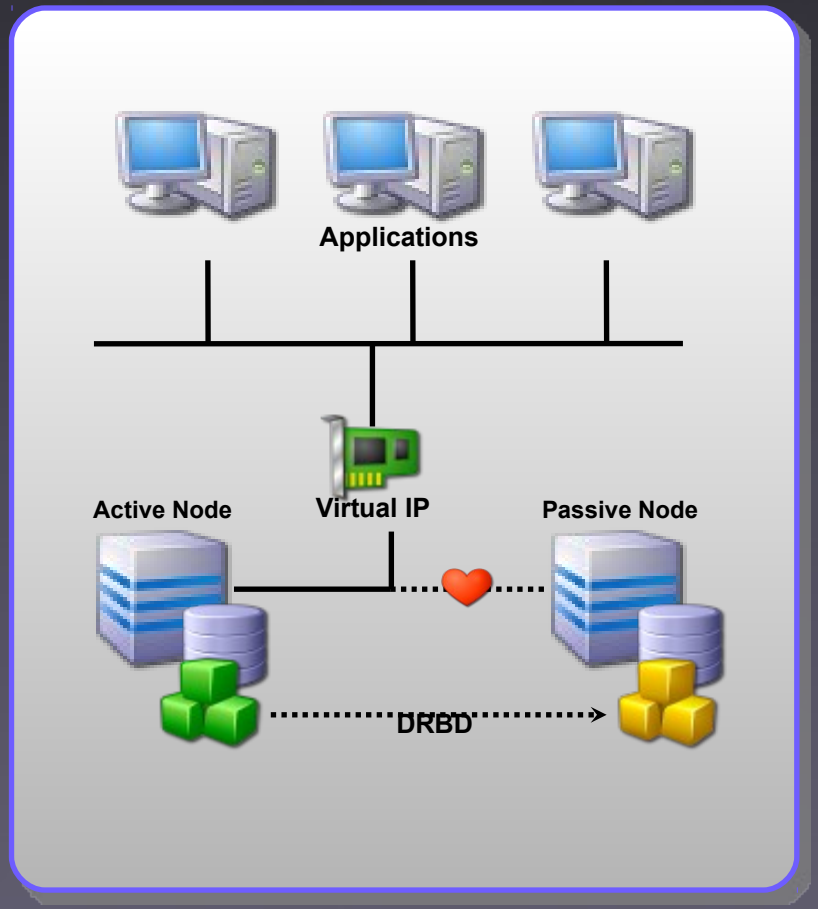


DRBD

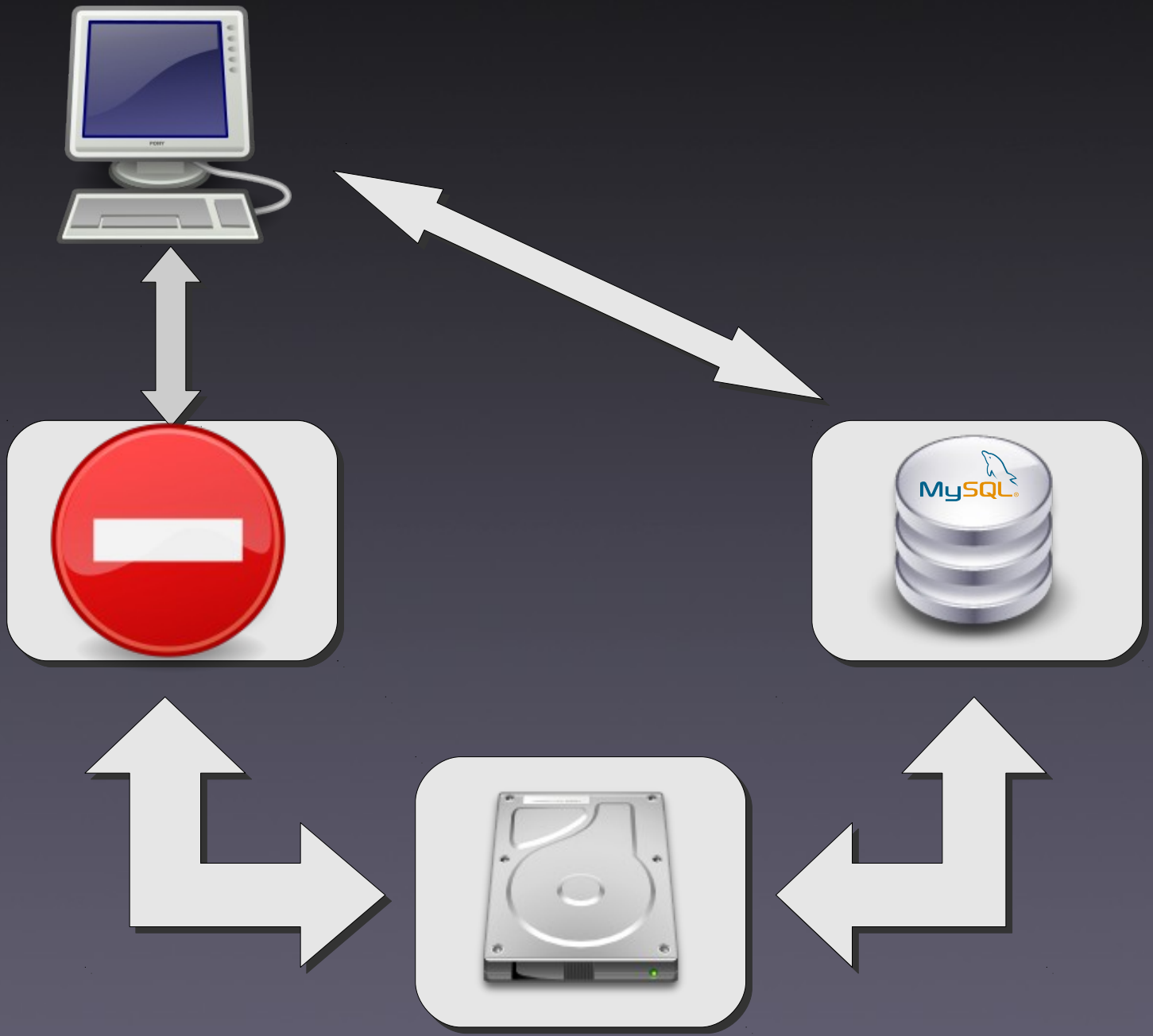
- Distributed Replicated Block Device
- “RAID-1 über das Netzwerk”
- Synchrone/asynchrone Block-Replizierung
- Automatische Resynchronisierung
- Applications-agnostisch
- Kann lokale I/O-Fehler maskieren
- Aktiv/passiv-Konfiguration vorgegeben
- Dual-primary Modus (benötigt ein Cluster Dateisystem wie GFS or OCFS2)
- http://drbd.org/

DRBD im Detail

- DRBD repliziert Datenblöcke zwischen zwei Plattenpartitionen
- DRBD kann mit Linux-HA und anderen HV-Lösungen gekoppelt werden
- MySQL läuft normal auf dem Primärknoten
- MySQL ist nicht aktiv auf dem Sekundärknoten
- DRBD ist nur für Linux verfügbar



Redundanz mit Shared Storage



Replikation vs. SAN

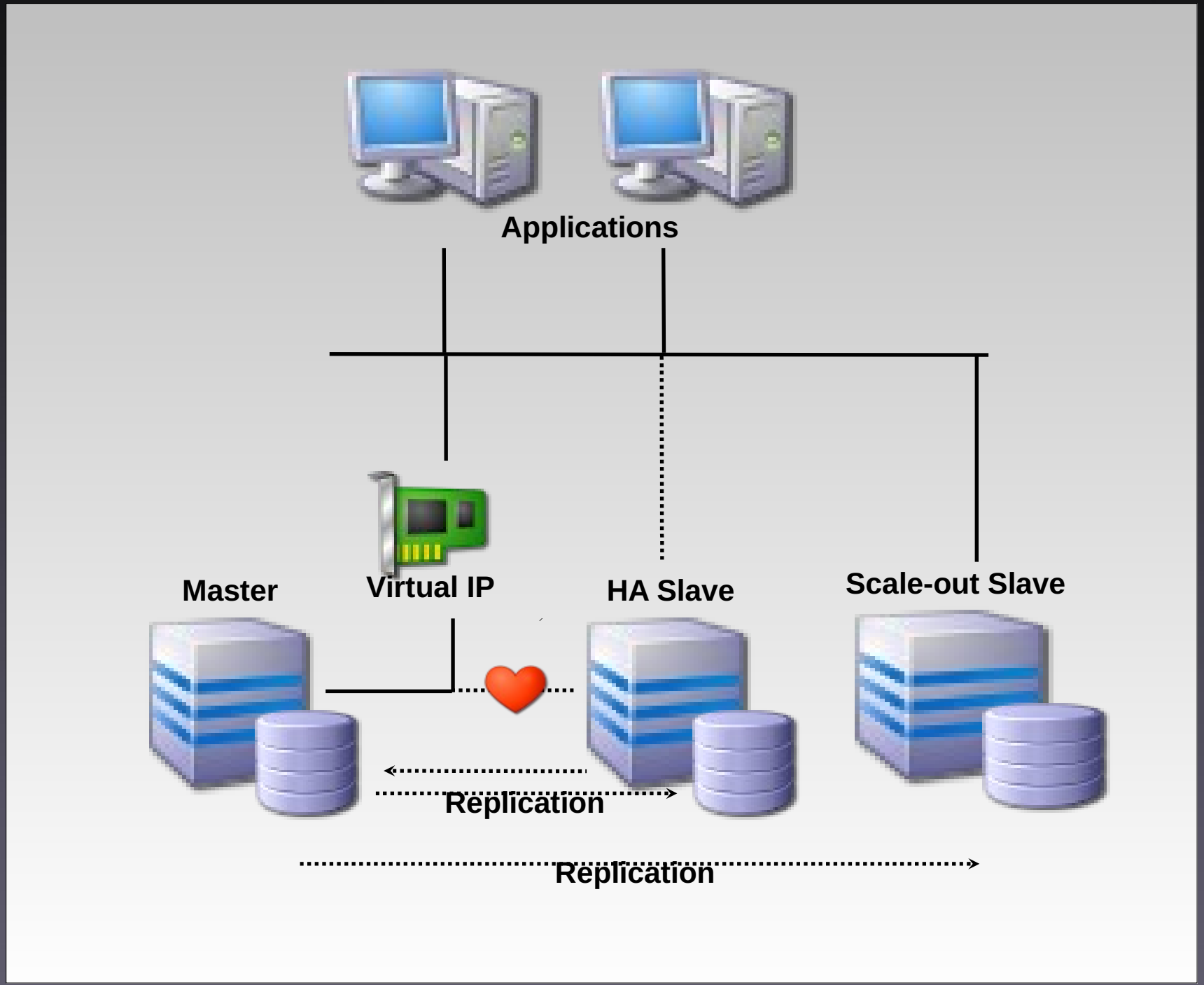
- Konsistenz der replizierten Daten
- Synchrone vs. asynchrone Replikation
- SAN kann zum SPOF werden
- “Split brain”-Situationen
- SAN/NAS I/O Overhead

Heartbeat/Pacemaker (Linux-HA)

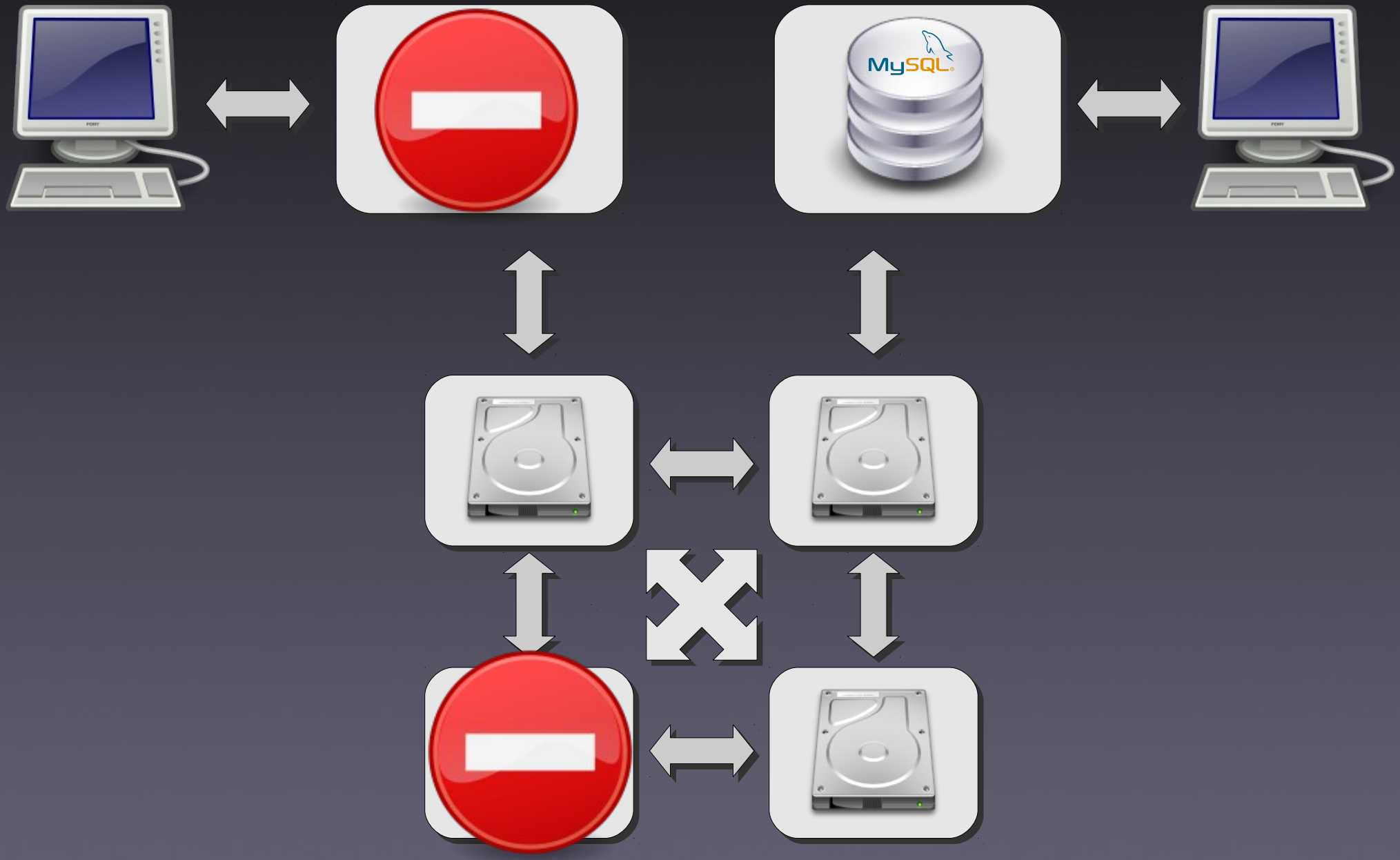
- Unterstützt 2 oder mehr Knoten (v2)
- Wenig Abhängigkeiten/ Anforderungen
- Überwachung von Resourcen
- Aktiver Fencing-Mechanismus: STONITH
- Erkennung eines Knotenausfalls in Sekundenbruchteilen
- http://linux-ha.org/

Heartbeat/Pacemaker (Linux-HA) (2)

- Richtlinien-basierte Resourcenverwaltung, Abhängigkeiten & Einschränkungen
- Zeitbasierte Regeln
- Unterstützt sehr viele Applikationen (incl. MySQL)
- Grafische Oberfläche


Applications
Virtual IP
Master
HA Slave
Scale-out Slave
Replication
Replication

Redundanz mit MySQL Cluster



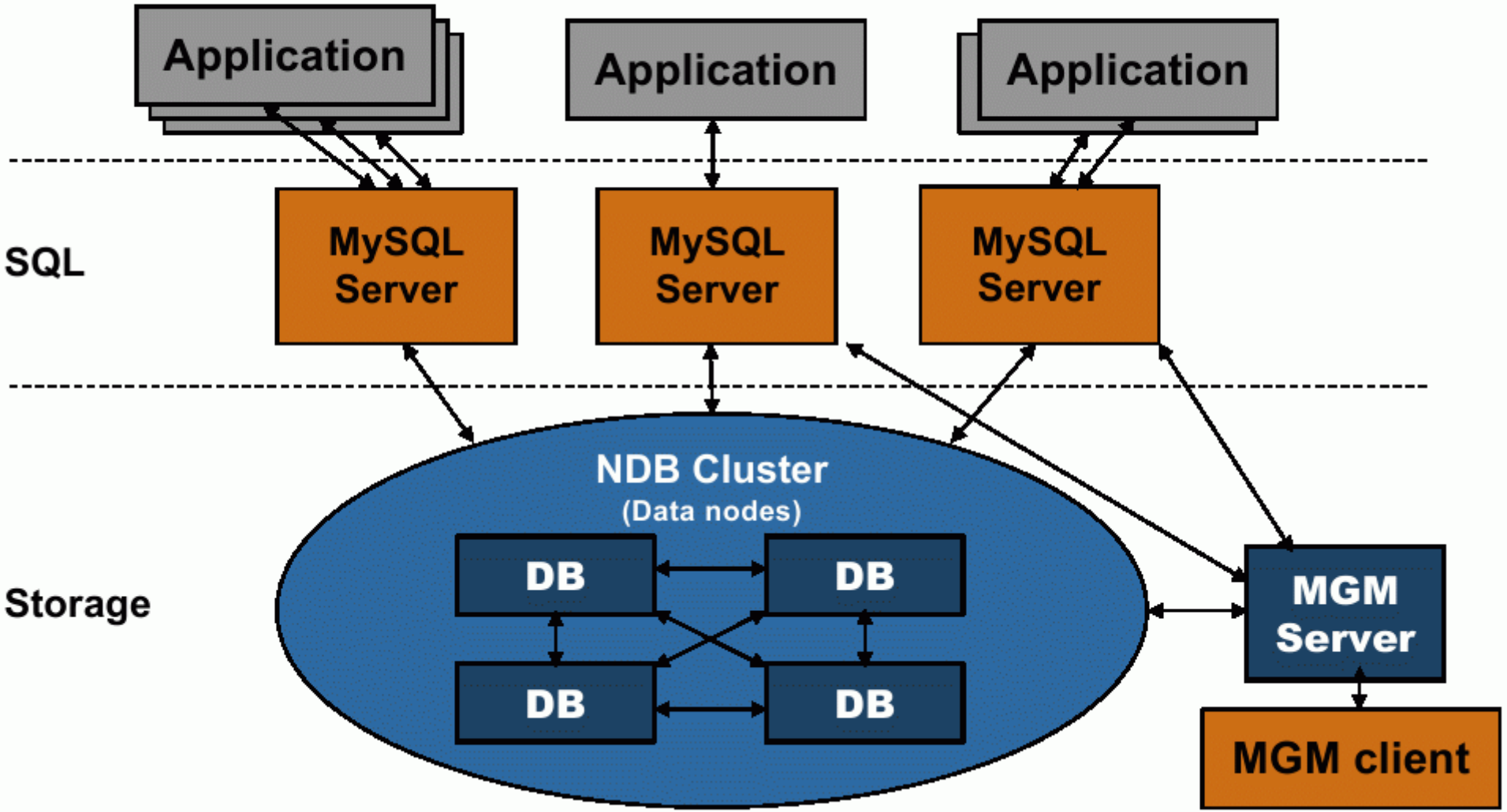
MySQL Cluster

- “Shared-nothing”-Architektur
- Automatische Partitionierung
- Verteilte Fragmente
- Synchrone Replikation
- Schnelles, automatisches Fail-Over der Datenknoten
- Automatische Resynchronisierung
- Transparent für MySQL-Applikationen
- Unterstützt Transaktionen
- http://mysql.com/products/database/cluster/

MySQL Cluster

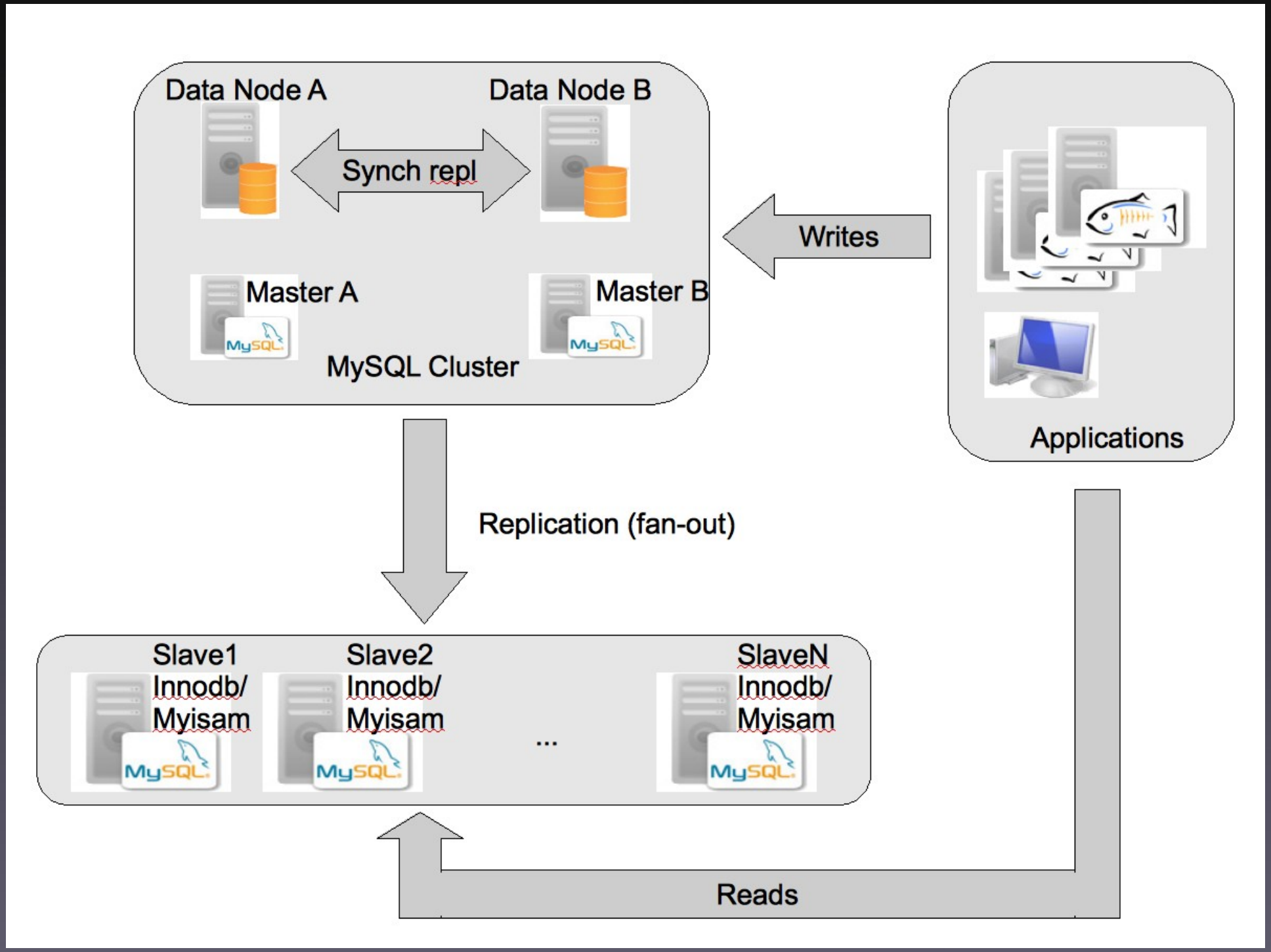
- In-memory indexes
- Nicht für alle Abfrage-Muster geeignet (JOINs über viele Tabellen, Range Scans)
- Keine Unterstützung für Fremdschlüssel
- Ungeeignet für lang laufende Transaktionen
- Netzwerk-Latenz wichtig
- Kann mit MySQL Replikation (RBR) kombiniert werden

Cluster Components



MySQL Cluster & Replikation

- MySQL Cluster
  - Einfacher failover von einem Master zum anderen
  - Schreiblast-Skalierung mittels multipler SQL-Knoten
- Asynchrone Replikation vom Cluster auf mehrere Slaves
- Leselast wird auf Slaves verteilt (InnoDB/MyISAM)
- Schnelles Einrichten weiterer Slaves (mit Cluster Online Backup)
- Leichtes Failover und schnelle Wiederherstellung



http://johanandersson.blogspot.com/2009/05/ha-mysql-write-scaling-using-cluster-to.html

Galera Replication

- Patch für InnoDB plus externe Bibliothek
- Synchrone Replikation
- Single- oder Multi-Master
- Multicast-Replikation
- HA plus Lastverteilung möglich
- Zertifikat-basierte Replikationsmethode (anstatt 2PC)
- http://codership.com/products/mysql_galera

MMM

- MySQL Master-Master Replication Manager
- Perl-Scripte zur Überwachung/Failover und Management
- Master-Master Replikationskonfiguration (ein aktiver beschreibbarer Master)
- Failover mittels Übernahme einer virtuellen IP-Adresse
- http://mysql-mmm.org/

Red Hat Cluster Suite

- HV und Lastverteilung
- Unterstützt bis zu 128 Knoten
- Unterstützt Shared Storage
- Überwacht Dienste, Dateisysteme und Netzwerk-Status
- Fencing (STONITH) oder distributed lock manager
- Synchronisierung der Konfiguration
- http://www.redhat.com/cluster_suite/

Solaris Cluster / OpenHA Cluster

- Provides failover and scalability services
- Solaris / OpenSolaris (Project Colorado)
- Kernel-level components plus userland
- Agents monitor applications
- Geo Edition to provide Disaster Recovery using Replication
- Open Source since June 2007
- http://www.opensolaris.org/os/community/ha-clusters/
- http://opensolaris.org/os/project/ha-mysql/

Flipper

- Perl Script zur Steuerung eines sich replizierenden Paars von MySQL-Servern (Master-Master)
- Automatisierter Switch-Over, manuell ausgelöst
- Kontrolliertes Herunterfahren einer Hälfte zu Wartungszwecken
- Die andere Hälfte übernimmt die Arbeit
- Verteilung von rollenbasierten IP Addressen ("nur-lesen", "beschreibbar") zwischen zwei Knoten im Master-Paar stellt sicher, daß beide Rollen verfügbar sind
- http://provenscaling.com/software/flipper/

Weitere Werkzeuge/Links

- Maatkit
  http://maatkit.sourceforge.net/
- Continuent Tungsten Replicator
  https://community.continuent.com/community/tungsten-replicator

Q & A

Fragen, Kommentare?

Vielen Dank!

Lenz Grimmer <lenz@grimmer.com>
Twitter: @lenzgr
http://lenzg.net/