



IT-Dienstleistungen
E-Learning Systeme
Content Management
Individuelle Software



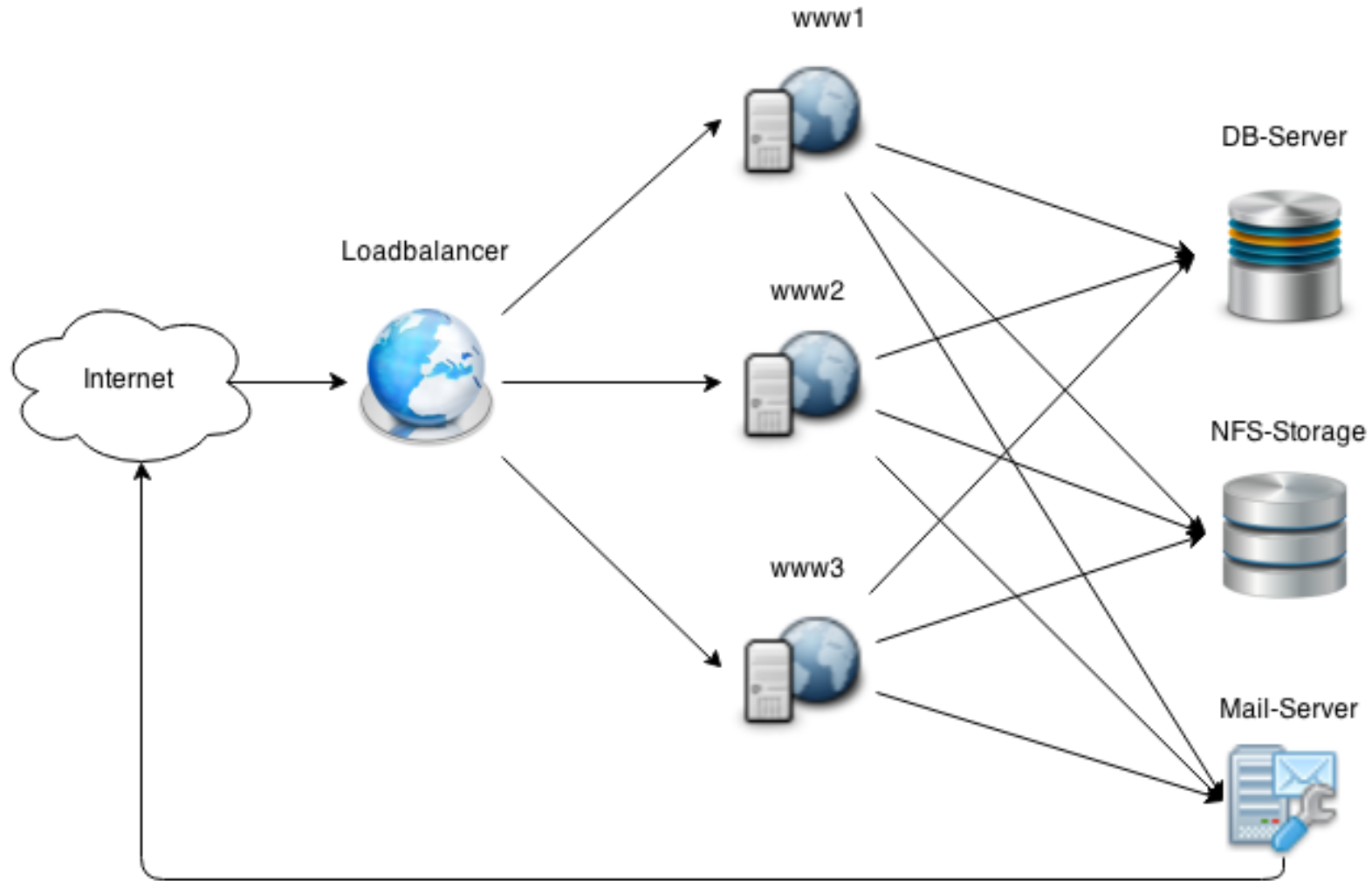
ILIAS aus der Krise skalieren

ILIAS-Conference September 2020

IT-Dienstleistungen aus Aachen

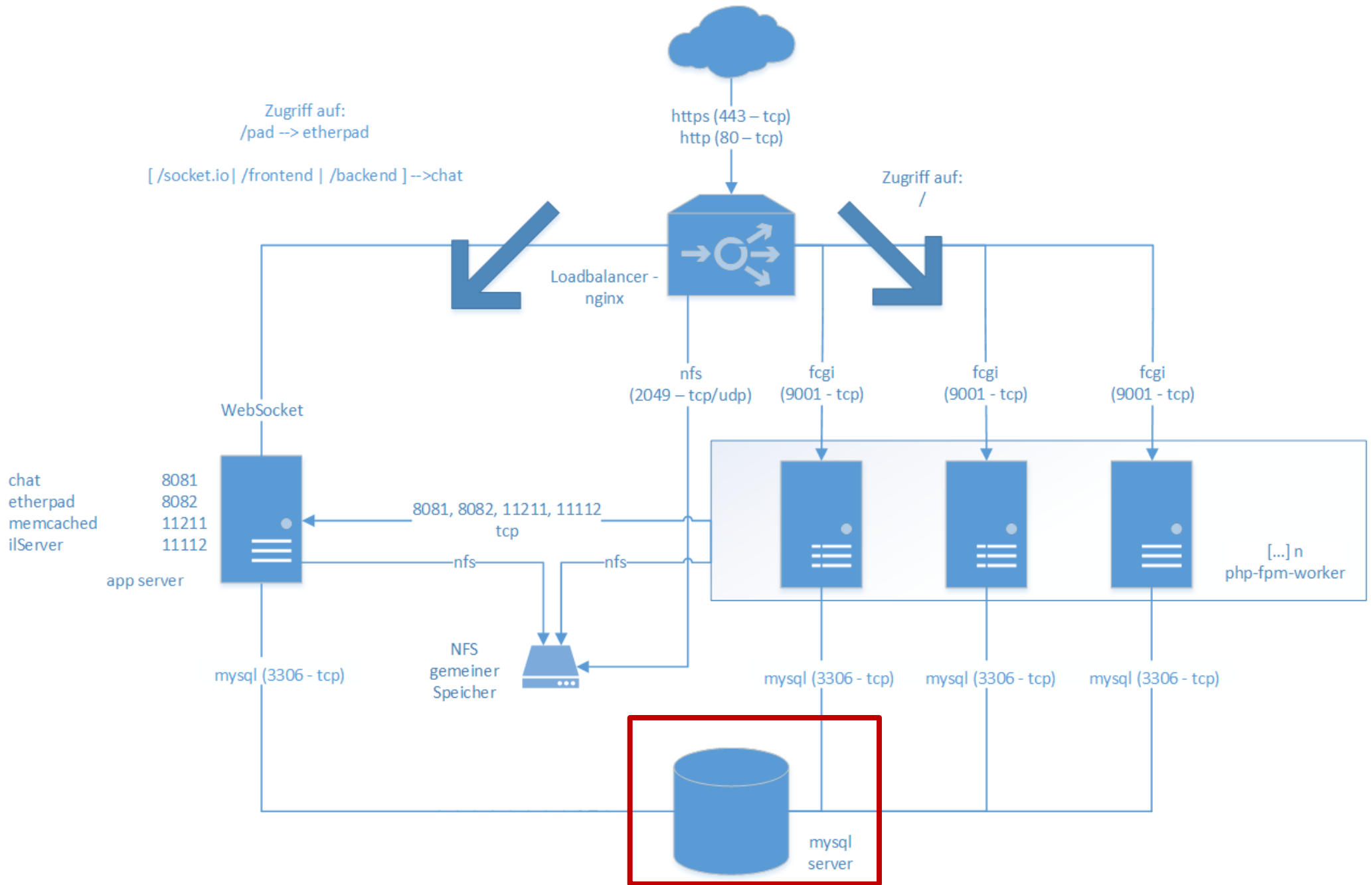
- Gründung im Juni 2000, ILIAS-Dienstleistungen/Entwicklung seit 2001
- Bereits zuvor Kooperation der drei Gründer und heutigen Vorstände
- 27 Mitarbeiter
- Ausbildungsbetrieb (Fachinformatiker, Mathematisch-Technische Softwareentwickler)
- Basis der Geschäftstätigkeit: Umsetzung von IT-Lösungen auf Basis von Open Source Software, Gründungsmitglied des ILIAS e.V.

ILIAS im Cluster



Corona ?

- 3-fache bis 5-fache Nutzungsintensität der ILIAS Instanzen
- DB-Server mit > 100.000 Queries/s
- Viele Limits gerissen auch rohe Betriebssystem-Parameter
 - File-Handles
 - Netzwerk-Connections
 - Anzahl und Frequenz Client-Ports bei Socket-Connections (tcp_fin_timeout)
 - Bond von 2xGigabit Netzwerkkinterfaces zur DB am Limit



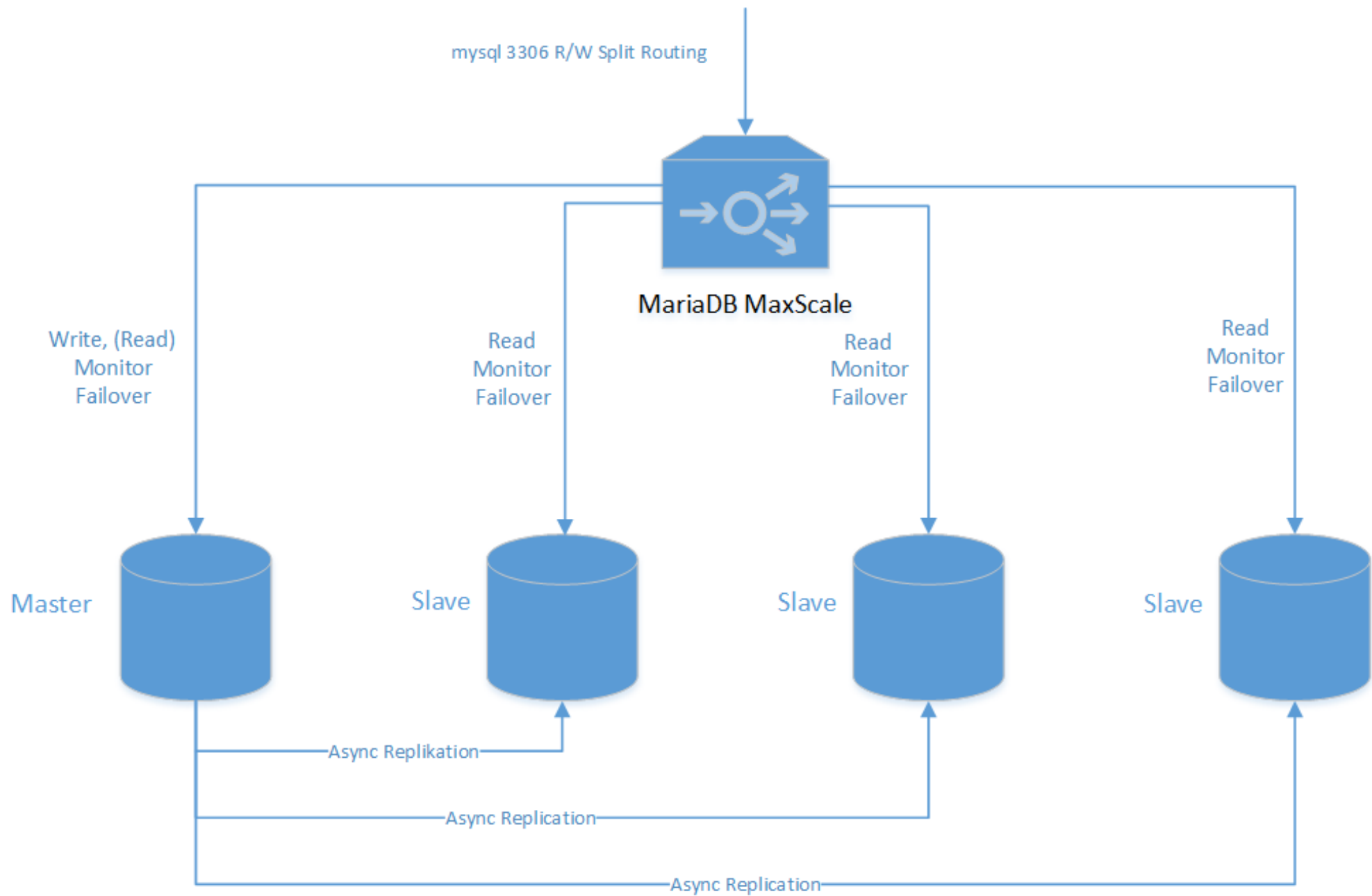
Skalierbarkeit

VERTIKAL („SCALE UP“)

- Hinzufügen von Ressourcen zu einem Rechner/VM
- Erhöhung (V)CPU, RAM
- Leistungsfähigere Hardware
- Optimaler Skalierungsfaktor
- Limitiert insbesondere in Virtualisierung
- Evtl. Betriebsunterbrechung bei Skalierung
- Transparent für Software

HORIZONTAL („SCALE OUT“)

- Hinzufügen zusätzlicher Rechner/VM
- Loadbalancer/Verteiler notwendig
- Theoretisch unlimitiert
- Durch Loadbalancer meist ohne Betriebsunterbrechung bei Skalierung
- Viele „kleinere“ Einheiten - virtualisierungsfreundlich
- Skalierungsfaktor niedriger
- Transparent für Software ???



Voraussetzungen

- VM für MaxScale mit moderaten Anforderungen (z.B. 4-8 vCPU, 8GB)
- MariaDB > 10.3.x
- Bereits bestehende Replikation basierend auf GTID
- Konfigurationsanpassungen proxy-protocol-networks, session_track_system_variables, bin-log auch auf Slaves
- Durchsicht mysql-Benutzertabelle auf user@NETZ/MASKE
- Ab 3 MariaDB-Server – Lizenzkosten !

Transparent für Software ?

- MaxScale bietet unterschiedliche „Router“ (siehe <https://mariadb.com/kb/en/maxscale-24-router/>)
- Ein „Router“ wird unter Nutzung eines „Service“ über einen „Listener“ als TCP-Port bereitgestellt
- Die DB-Server selber werden von einem „Monitor“ überwacht
- Relevant für das skalierbare ILIAS System ist der „Readwritesplit“ Router
- Der Zugangspunkt zum Service kann wie zur MariaDB auf TCP-Port 3306 stattfinden
- Authentifizierung mit mysql Benutzer und Passwort erfolgt via proxy-protocol
- Ein MaxScale kann in ein bestehendes Master-Slave System zwischengeschaltet werden
- Keine Anpassungen an der Software ausser Zugangs-IP/Port notwendig

Keine Angst vor Inkonsistenzen

„causal_read“ stellt konsistente Ergebnisse sicher

Query:

```
INSERT INTO test.t1 (id) VALUES (1); # Must be routed to master
SELECT * FROM test.t1 WHERE id = 1; # Can be routed to slave
```

Führt zu Abfragen:

```
INSERT INTO test.t1 (id) VALUES (1); # To master and remember GTID
SET @maxscale_secret_variable=(
  SELECT CASE
    WHEN MASTER_GTID_WAIT('0-3000-8', 3) = 0 THEN 1
    ELSE (SELECT 1 FROM INFORMATION_SCHEMA.ENGINES)
  END);
SELECT * FROM test.t1 WHERE id = 1; # To slave, wait at most 3s, else ask master
```

Slaves werden „read_only“ betrieben

Benefit Hochverfügbarkeit

- Monitoring und „auto_failover“ schaltet automatisch einen Slave->Master und alle anderen Slaves werden auf den neuen Master konfiguriert
- Ist ein Server in der Replikation hinterher mit „Lag“ wird er nicht verwendet
- Manuell lassen sich Server in „maintenance“ schalten
- Manueller Switch Master/Slave: „maxctrl call command mariadbmon switchover MariaDB-Monitor mariadb02 mariadb01“
- Damit sind auch weitgehend alle Wartungsarbeiten unterbrechungsfrei machbar

```

Service:          Read-Write-Service
Router:          readwritesplit
State:           Started

```

```

use_sql_variables_in:    all
slave_selection_criteria: ADAPTIVE_ROUTING
master_failure_mode:     fail_instantly
max_slave_replication_lag: 5
retry_failed_reads:      true
strict_multi_stmt:       false
strict_sp_calls:         false
prune_sescmd_history:    false
disable_sescmd_history:  false
max_sescmd_history:      50
master_accept_reads:     true
connection_keepalive:    120
causal_reads:            true
causal_reads_timeout:    2
master_reconnection:     true
delayed_retry:           false
delayed_retry_timeout:   10

```

```

Number of router sessions:      3269207
Current no. of router sessions: 0
Number of queries forwarded:    565000917
Number of queries forwarded to master: 50504881 (8.94%)
Number of queries forwarded to slave: 514497422 (91.06%)
Number of queries forwarded to all: 6475744 (1.15%)
Number of read-write transactions: 0
Number of read-only transactions: 0
Number of replayed transactions: 0

```

Server	Total	Read	Write	Sess Avg:	dur	active	selects
mariadb01	279476718	228971837	50504881		123.433ms	15.41%	84
mariadb02	141205518	141206012	0		123.42ms	10.60%	42
mariadb03	163745557	163746449	0		123.419ms	12.04%	49

```
Started: Thu May 28 13:05:44 2020
```

```
Root user access: Enabled
```

```
Backend databases:
  [10.23.24.38]:3306 Protocol: MariaDBBackend Name: mariadb01
  [10.23.24.39]:3306 Protocol: MariaDBBackend Name: mariadb02
  [10.23.24.40]:3306 Protocol: MariaDBBackend Name: mariadb03

```

```
Total connections: 3269222
Currently connected: 0
```

```
root@zs-23-24-37:~# maxadmin show monitors
Name: MariaDB-Monitor
State: Running
Times monitored: 8273542
Sampling interval: 1000 milliseconds
Connect Timeout: 3 seconds
Read Timeout: 1 seconds
Write Timeout: 2 seconds
Connect attempts: 1
Monitored servers: [10.23.24.38]:3306, [10.23.24.39]:3306, [10.23.24.40]:3306
Automatic failover: Enabled
Failcount: 5
Failover timeout: 90
Switchover timeout: 90
Automatic rejoin: Disabled
Enforce read-only: Enabled
Enforce simple topology: Disabled
Detect stale master: Enabled

Server information:
-----

Server: mariadb01
Server ID: 1
Read only: No
Gtid current position: 0-1-43410984
Gtid binlog position: 0-1-43410984
No slave connections

Server: mariadb02
Server ID: 2
Read only: Yes
Gtid current position: 0-1-43410984
Gtid binlog position: 0-1-43410984
Slave connections:
  Host: [10.23.24.38]:3306, IO/SQL running: Yes/Yes, Master ID: 1, Gtid_IO_Pos: 0-1-43410984, R.Lag: 0

Server: mariadb03
Server ID: 3
Read only: Yes
Gtid current position: 0-1-43410984
Gtid binlog position: 0-1-43410984
Slave connections:
  Host: [10.23.24.38]:3306, IO/SQL running: Yes/Yes, Master ID: 1, Gtid_IO_Pos: 0-1-43410984, R.Lag: 0
```

Performance vs. Kapazität

Wie groß ist der Performancegewinn ?

Keiner bei gleichen eingesetzten Ressourcen !

Aber...

Die Grenzen einiger mir bekannter Installationen wurden erreicht...

Die Grenzen werden für MariaDB im Wesentlichen durch Anzahl CPU/Threads bestimmt - nicht Speicher/IO.

48 Kerne/96 Threads können eine Lösung sein, aber Ihre IT wird Sie hassen, weil das nicht in die Virtualisierungslösung/Management/Backup passt.

my.cnf Anpassungen für maxscale Betrieb

```
# Enable PROXY protocol for authentication

proxy-protocol-networks=::1, 127.0.0.1, localhost, 192.168.252.0/24

# Track last_gtid in session variables causal reads to add to default
value)
session_track_system_variables =
"autocommit,character_set_client,character_set_connection,character_set_r
esults,time_zone,last_gtid"
```


Snippets

```
/etc/sysctl.d/90-socket-tuning.conf:

# Recieve and Send Socket Mem Buffer Sizes
net.core.rmem_max=4194304
net.core.wmem_max=4194304

# Sockets (default 128)
net.core.somaxconn = 4096

# Increase the number of outstanding syn requests allowed.
# c.f. The use of syncookies.
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_syncookies = 1

# Widen local portrange
net.ipv4.ip_local_port_range = 2051 64512

# Shorter tcp_fin_fimeout (default 60)
net.ipv4.tcp_fin_timeout = 15
```

