



MySQL Performance Tuning

Frankfurter Datenbanktage
13. März 2013

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com



Über FromDual GmbH

- FromDual bietet neutral und unabhängig:
 - Beratung für MySQL und Galera
 - Support für MySQL und Galera
 - Remote-DBA Dienstleistungen
 - MySQL Schulungen
- Partner der Open Database Alliance (ODBA.org)
- Oracle Silver Partner (OPN)



www.fromdual.com

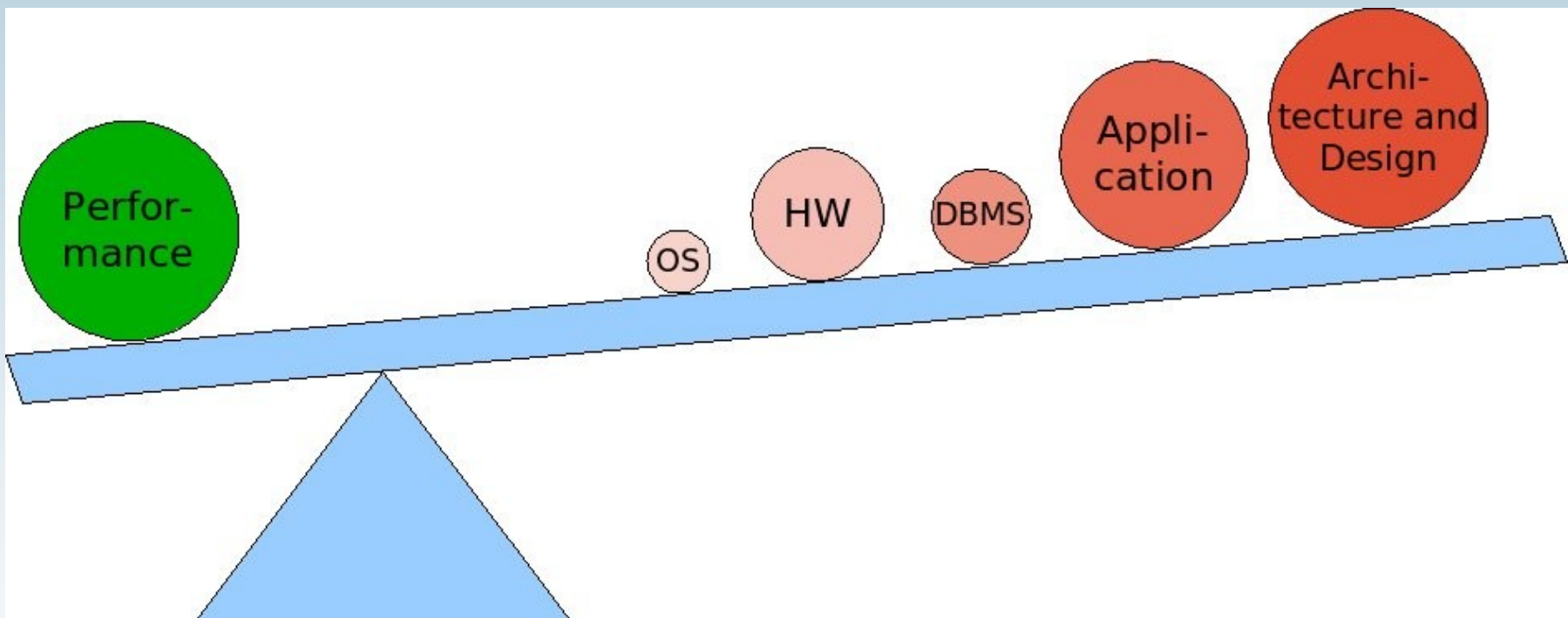
Inhalt

Performance Tuning

- › Messen
- › Hardware & O/S
- › MySQL Konfiguration
- › Applikationstuning
- › Architektur

Der Weg

- Ausgangslage: Kunde schreit, weil er ein Performance Problem hat!
 - Viele Wege führen nach Rom!



Sammeln

- **Zusammentragen von Fakten:**
 - **Wie sieht das Problem aus?**
 - DB ist plötzlich, manchmal oder schon immer langsam?
 - Was ist genau langsam?
 - Wurde ein neuer Release eingespielt?
 - Hat jemand „rumgefummelt“?
 - **Haben wir historische Messdaten?**
 - **Am besten wenn:**
 - man das Problem (gezielt) simulieren kann
 - es vorhersagbar oder periodisch auftritt
- **Finde das Muster!**



Ressourcen

- Finde den Flaschenhals / die limitierende Ressource:
- Zum Glück „nur“:
 - CPU
 - Speicher (RAM)
 - I/O (IOPS, Durchsatz)
 - Netzwerk (FpS, Durchsatz)
- Aber wie?



Messen CPU

- Was tut mein System?

vmstat

```
# vmstat 1
procs -----memory----- ---swap-- ----io---- -system-- ----cpu----
r  b   swpd   free   buff   cache   si   so    bi    bo    in   cs  us  sy  id  wa
1  0   96148  56096  35936  548792   0   0     0    656   379  343  5  38  57  0
0  0   96148  56096  35936  548792   0   0     0     0   260  357  5  34  61  0
0  0   96148  56096  35936  548792   0   0     0     0   306  399  9  29  62  0
3  0   96148  49192  35940  549808   0   0   1020    0   289  431  91  4  3  2
1  0   96148  47424  35944  551572   0   0   896     0   310  378  98  2  0  0
1  0   96148  45656  35944  553344   0   0   896     0   260  359  98  1  0  1
2  0   96148  43948  35944  555112   0   0   896     0   280  355  97  3  0  0
1  0   96148  42056  35952  556884   0   0   904     0   260  374  99  0  0  1
1  0   96148  40288  35984  558672   0   0   896   3772   312  398  97  3  0  0
1  0   96148  38520  35984  560424   0   0   896     0   259  365  97  1  0  2
```

- Welcher Prozess braucht denn CPU?

top

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
26251	mysql	20	0	3107m	296m	6936	S	64	3.8	0:02.43	mysqld
23819	root	20	0	0	0	0	S	10	0.0	0:35.73	kworker/0:1
26288	mysql	20	0	238m	137m	2256	S	10	1.7	0:00.36	mysql
1154	root	20	0	199m	49m	23m	S	5	0.6	3:49.37	Xorg
2013	oli	20	0	414m	108m	24m	S	2	1.4	1:39.69	skype
2109	oli	20	0	423m	107m	24m	S	2	1.4	1:39.35	skype
23171	oli	20	0	496m	29m	19m	S	1	0.4	0:05.57	konsole

Messen: Speicher (RAM)

- **free / top:**

```
#free
                total    used      free  shared  buffers  cached
Mem:           1036016  983864    52152        0    35484  547432
-/+ buffers/cache: 400948    635068
swap:          4202112    96148  4105964
```

- **ps**

```
# ps -eo user,pid,%cpu,%mem,vsz,rsz,comm --sort -vsz | \
  egrep 'mysql|COMMAND'
USER          PID %CPU %MEM    VSZ   RSZ COMMAND
mysql         1361  0.0  1.5 108368 16444 mysqld
mysql         1210  0.0  0.1   4536  1956 bash
mysql         1289  0.0  0.1   4060  1444 safe_mysqld
mysql         1204  0.0  0.1   4048  1404 su
```


Messen I/O

vmstat

```
# vmstat 1
procs ---swap-- ----io---- ----cpu----
 r  b   si   so   bi    bo  us  sy  id  wa
 0  0    3    3   94   143 21  21 56  2
 0  0    0    0    0     4  9  37 54  0
```

iostat (→ sysstat package)

```
# iostat -x 1
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5.88    0.00  34.31   2.94    0.00  56.86

Device:            r/s    w/s  rkB/s  wkB/s  await  svctm   %util
hda                 0.00   0.00   0.00   0.00   0.00   0.00   0.00
hdc                 0.00   2.94   0.00  23.53  14.67  12.00   3.53
```

Messen Netzwerk

- `watch -d -n 1 'ifconfig'`
- `iptraf`

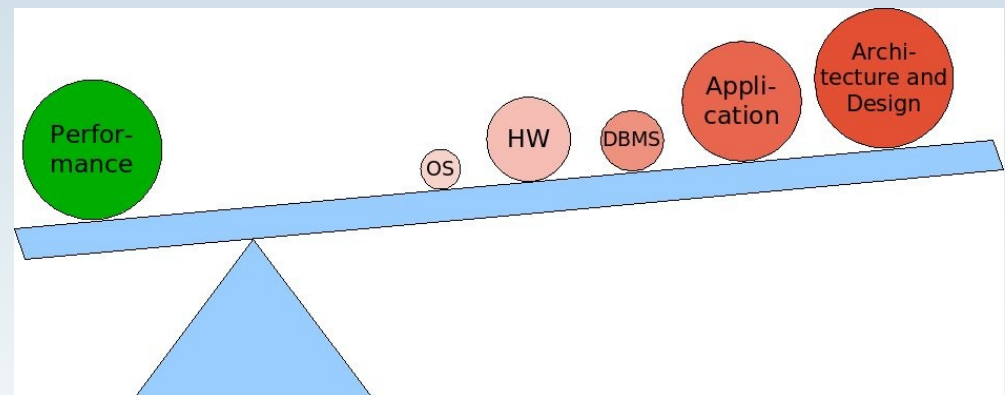
Messen Zusammenfassung

- **CPU**
 - Welcher Prozess
 - wie viele Cores?
 - meist SQL Abfragen
- **Speicher**
 - Welcher Prozess
 - Swapping?
 - Über- oder Unterallozierung von DB Caches!
- **I/O**
 - welches Device?
 - Durchsatz oder IOPS
 - Random oder Sequential I/O
 - Read oder Write?
 - Caches zu klein, tmp Tabellen?
- **Netzwerk**
 - Errors / Drops?
 - Durchsatz

Stellschrauben

- CPU
 - Schnellere Cores?
 - Mehr Cores
- Speicher
 - Mehr RAM?
- I/O
 - RAID-5 :-(
• SAN :-(
• RAID-10, viele Spindeln, SSD?
• Batterie gepufferter I/O Cache!
- Netzwerk
 - ≥ 1 Gbit
- OS
 - Neue 64-bit Kernel ($\geq 2.6.3x$)
 - XFS
 - \rightarrow I/O Scheduler
 - noop
- Virtualisierung :-(

Datenbank Tuning



MySQL Tuning

- Welche Storage Engine verwenden Sie zur Zeit?
- Welchen MySQL Release? (→ 5.1 und neuer)
- Zur Zeit: ca. 330 MySQL Parameter
 - aber nur ca. 9 davon sind primär signifikant!
 - **Grob-Tuning**
- Alle anderen nur nach ausführlichem Benchmarken
 - **Fine-Tuning**

MySQL Grob-Tuning

- **key_buffer_size**
 - ca. 25 – 33% vom RAM auf dedizierter Maschine
 - `SHOW STATUS LIKE 'Key_blocks_%';`
- **table_open_cache**
 - Running connections x used tables → 2 – 4k ist nicht ungewöhnlich! Siehe `Open[ed]_tables`.
- **table_definition_cache**
 - Siehe `Open[ed]_table_definitions` → 512 – 4096 ist nicht ungewöhnlich!
- **query_cache_type/query_cache_size**
 - Nicht zu gross machen (≤ 128 M), bei sehr hoher Concurrency aber schädlich!

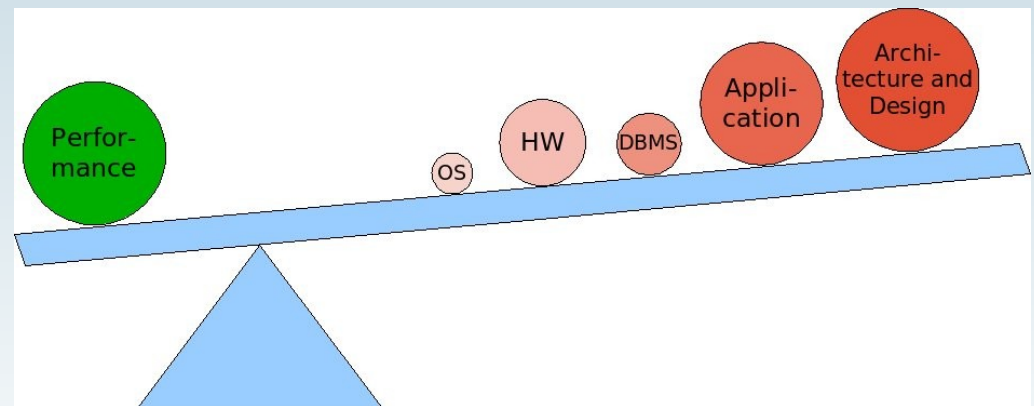
InnoDB Grob-Tuning

- **innodb_buffer_pool_size**
 - ca. 80% vom RAM auf dedizierter Maschine
 - `SHOW STATUS LIKE 'Innodb_buffer_pool_pages%';`
- **innodb_buffer_pool_instances**
 - Ab 5.5, bei grossem Buffer Pool und hoher Konkurrenz
- **innodb_log_file_size**
 - Grösser = schneller, aber längere Recovery Zeiten → 2 x 256 M
- **innodb_flush_log_at_trx_commit**
 - 0, 2 für Performance, 1 für Sicherheit
- **sync_binlog**
 - `!= 0` → langsam(er)

Weitere Hilfe

- **Wie messen?**
 - **SHOW GLOBAL STATUS;**
 - **SHOW ENGINE INNODB STATUS\G**
- **MySQL Database Health Check:**
<http://www.fromdual.com/mysql-database-health-check>
- **MySQL Doku, Server Status Variablen:**
<http://dev.mysql.com/doc/refman/5.5/en/server-status-variables.html>

Applikations-Tuning



Applikations-Tuning

- **Index Tuning**
 - **Primary Key (InnoDB)**
 - Lokalität der Daten
 - Länge der Sekundärindices!
 - (partiell) Redundante Indices weg
 - Indices mit einer geringen Kardinalität weg!?!
- **Query Tuning**
 - **SHOW PROCESSLIST;**
 - **Slow Query Log (ab 5.1 dynamisch!)**
 - **log_queries_not_using_indexes = 1**
 - **mysqldumpslow -s t slow.log > slow.log.profile**
 - **EXPLAIN SELECT ...**
 - **SHOW PROFILE**

Der Schlüssel zur Wahrheit

- **EXPLAIN**
- **Der Schlüssel zur Wahrheit!**
- **Query Tuning: x schneller möglich!**

```
EXPLAIN SELECT * FROM test;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	test	ALL	NULL	NULL	NULL	NULL	261369	

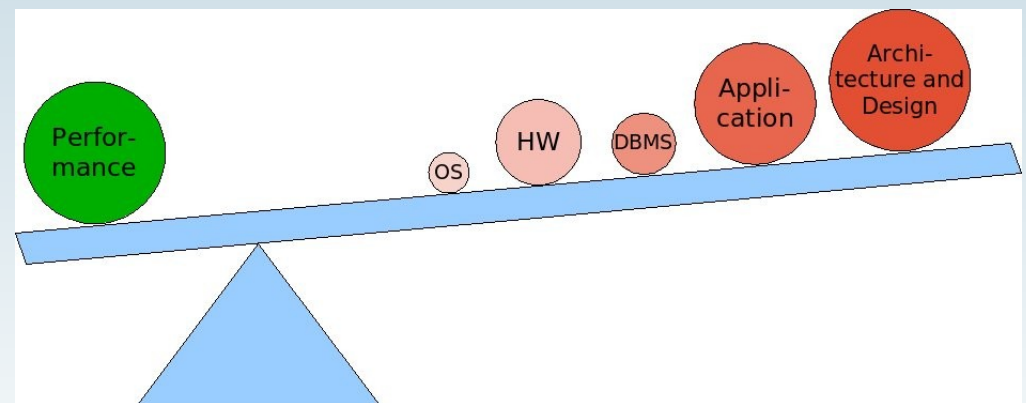
- **EXPLAIN Output Format:**

<http://dev.mysql.com/doc/refman/5.5/en/explain-output.html>

Applikations-Tuning

- **Loslassen!**
 - **Spalten, welche nicht gebraucht werden → weg (char(0) oder ausNULLen)!**
 - **Alte Daten → weg (archivieren)!**
- **Schema Tuning**
 - **mysqldump --no-data > structure_dump.sql**
 - **Richtige Datentypen und richtige Längen!**
 - **int(1) → 4 byte int!**
 - **utf8 → nur wenn nötig**
 - **NULL or NOT NULL**

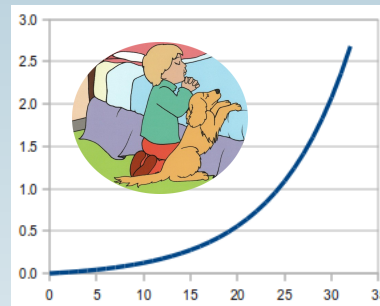
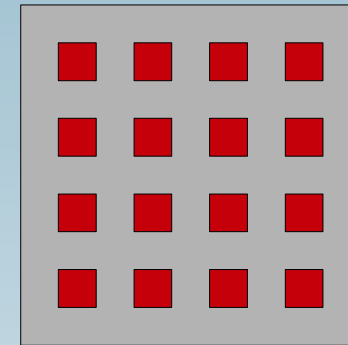
Architektur & Design



MySQL Scale-Out vs Scale-Up

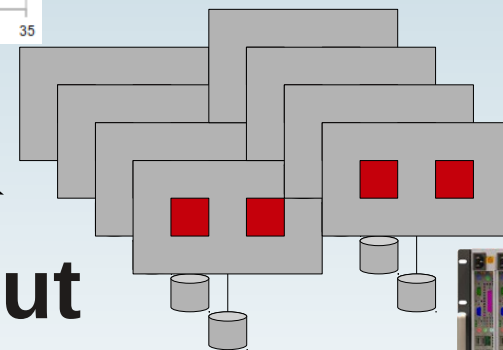
- Kosten
- MySQL Design
- Physische Flaschenhalse

Scale-Up



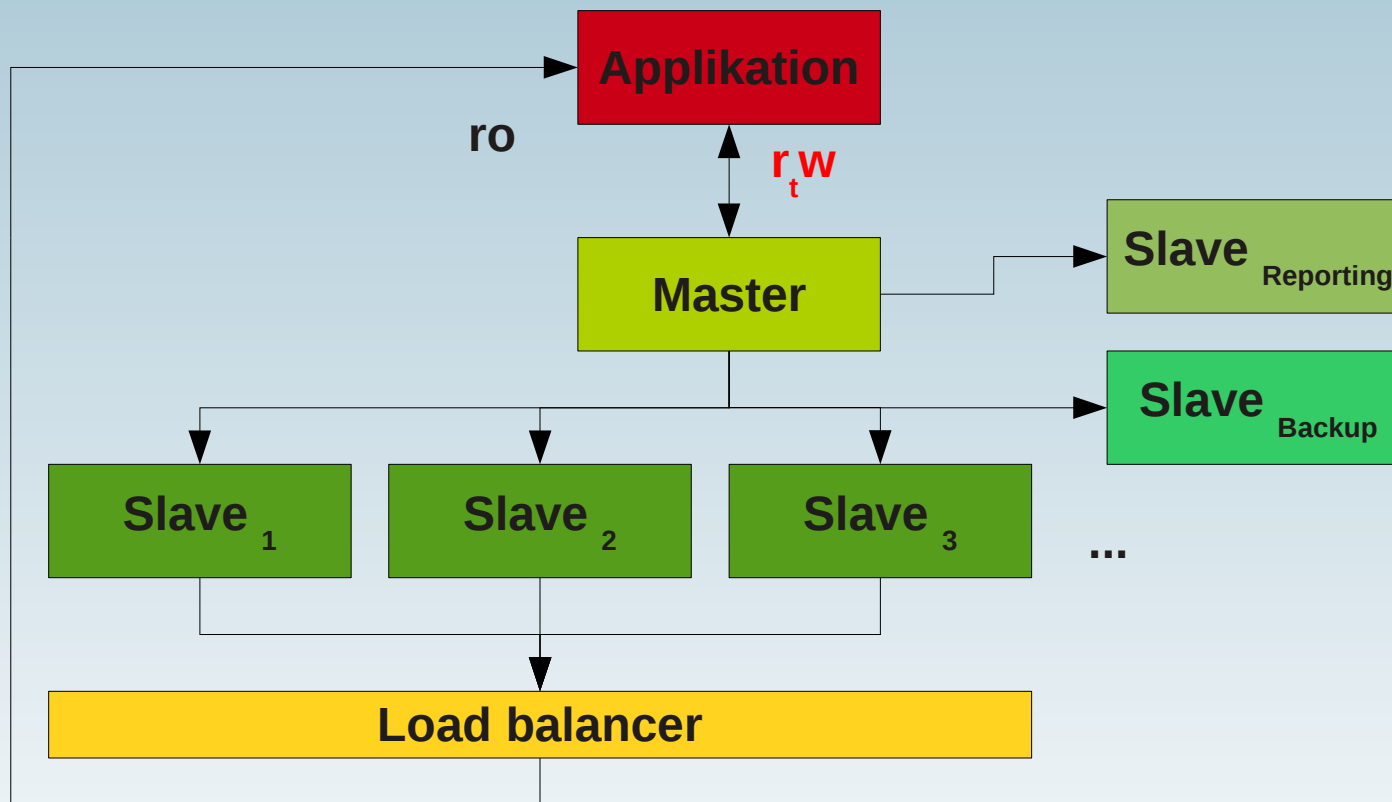
„Relaxation of Constraints“

Scale-Out



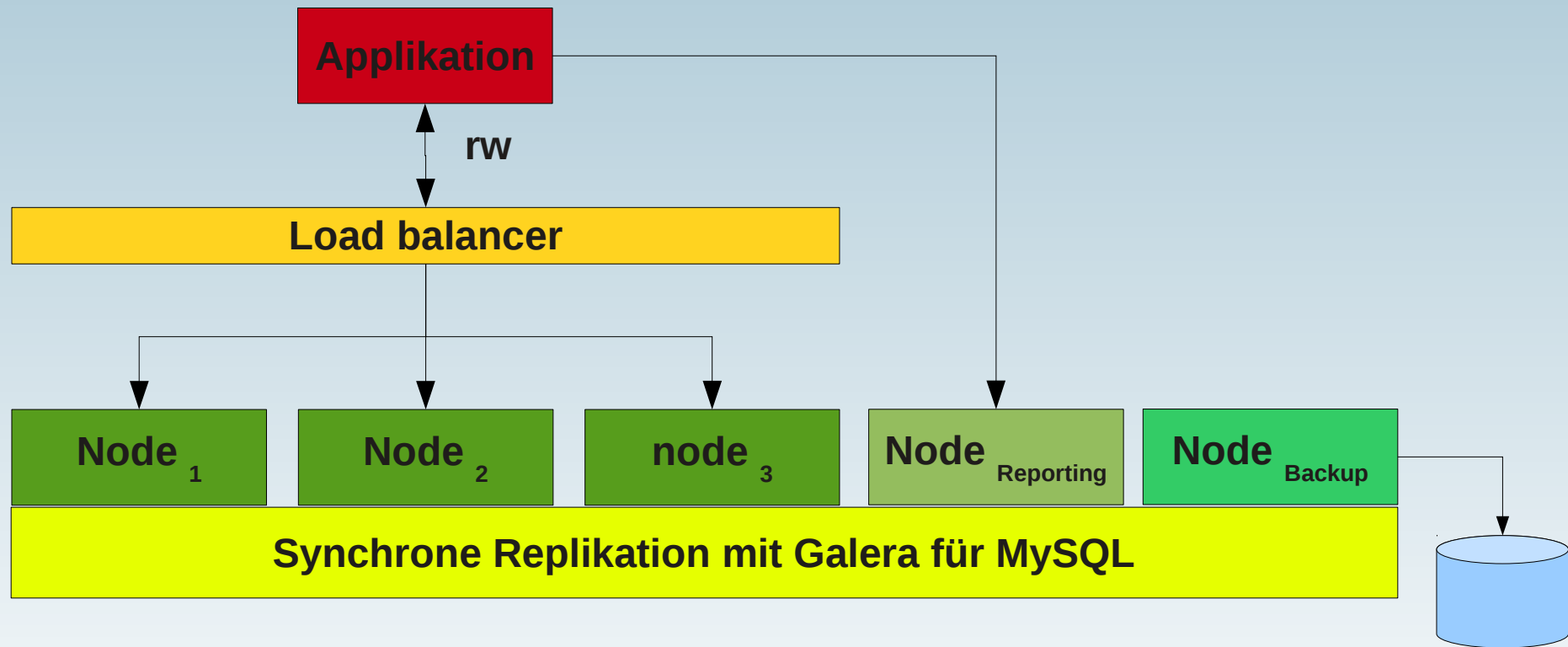
Der MySQL Scale-Out Ansatz

- Traditionell



Der MySQL Scale-Out Ansatz

- State-of-the-Art:



Q & A



www.fromdual.com



Fragen ?
Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

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

www.fromdual.com/presentations