

Skalering og tuning af MySQL

A large, faint, light gray outline of the MySQL fish logo is visible on the left side of the slide, serving as a background watermark.

Carsten Pedersen

Coordinator of Development, Certification Manager

MySQL AB

LinuxForum 6. marts 2004

Overview

- Tuning – Hvordan får vi serveren til at yde mere?
 - Hvor findes problemet?
 - Tuning af forespørgsler
 - Re-design af tabeller
 - Tuning af serveren
- Skalering – Hvordan kan vi udnytte extra hardware?
 - Replikering
 - Clustering

Tuning – Generelle råd

- Brug indexes
- Overvej at tilgå tabeller i en anden rækkefølge
- Indexes er allerede sorteret – brug det til din fordel!
- Undgå at scanne hele tabeller
- Hent kun det antal rækker du skal bruge
- Hent kun de kolonner du skal bruge
- Undgå joins hvis du kan

Tuning – Indsamle informationer

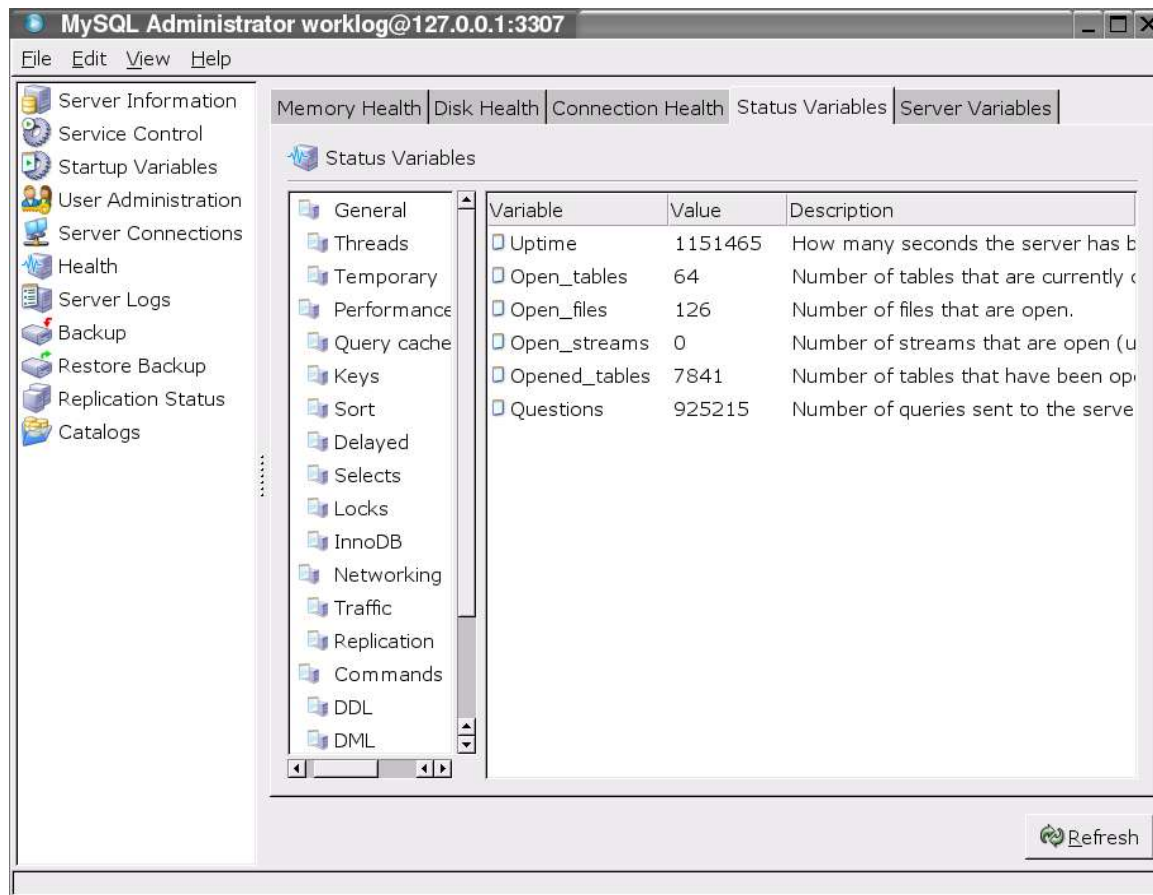
- Start med at samle informationer
 - `SELECT` ct. UPDATE/INSERT/DELETE (general log)
 - Lang svartid (slow log)
 - `long_query_time` (default: 10)
 - Status

```
mysql> status

...

Threads: 262  Questions: 2996999629
Slow queries: 1342  Opens: 270  Flush tables: 1
Open tables: 259
Queries per second avg: 132.093
```

MySQL Administrator



The screenshot shows the MySQL Administrator window titled "MySQL Administrator worklog@127.0.0.1:3307". The interface includes a menu bar (File, Edit, View, Help) and a left-hand navigation pane with categories like Server Information, Service Control, Startup Variables, User Administration, Server Connections, Health, Server Logs, Backup, Restore Backup, Replication Status, and Catalogs. The main area is divided into tabs: Memory Health, Disk Health, Connection Health, Status Variables (selected), and Server Variables. Under the Status Variables tab, there is a tree view on the left with categories such as General, Threads, Temporary, Performance, Query cache, Keys, Sort, Delayed, Selects, Locks, InnoDB, Networking, Traffic, Replication, Commands, DDL, and DML. The right pane displays a table of variables:

Variable	Value	Description
<input type="checkbox"/> Uptime	1151465	How many seconds the server has b
<input type="checkbox"/> Open_tables	64	Number of tables that are currently c
<input type="checkbox"/> Open_files	126	Number of files that are open.
<input type="checkbox"/> Open_streams	0	Number of streams that are open (u
<input type="checkbox"/> Opened_tables	7841	Number of tables that have been op
<input type="checkbox"/> Questions	925215	Number of queries sent to the serve

A "Refresh" button is located at the bottom right of the main content area.

Tuning – Brug af indexes

- Flere informationer – er disse værdier “store”?
 - `Handler_read_first` \Rightarrow Many full index scans
 - `Handler_read_next` \Rightarrow Full index scans, range selects
 - `Handler_read_rnd` \Rightarrow Many full table scans, range selects
- Overvej:
 - Bruges indexes optimalt?

Tuning – Caches

- ...og hvad med disse?
 - `Opened_tables` \Rightarrow Increase `table_cache`
 - `Key_reads` \Rightarrow Increase `key_cache`

EXPLAIN SELECT ...

```
mysql> explain select Language, SUM(Percentage * Population) / 100 as Speakers
-> from Country, CountryLanguage where Country.Code =
CountryLanguage.Country and
-> Continent = 'Europe' group by Language \G
***** 1. row *****
      table: CountryLanguage
      type: ALL
possible_keys: PRIMARY
      key: NULL
      key_len: NULL
      ref: NULL
      rows: 984
      Extra: Using temporary
***** 2. row *****
      table: Country
      type: eq_ref
possible_keys: PRIMARY
      key: PRIMARY
      key_len: 3
      ref: CountryLanguage.Country
      rows: 1
      Extra: where used
2 rows in set (0.00 sec)
```

EXPLAIN SELECT - type: ...

- `system`: kun een række (konstant værdi; meget sjælden)
- `const`: Kun een række, som kan læses i starten af forespørgslen
- `eq_ref`: Een række per kombination af rækker fra tidligere tabeller
- `ref`: Flere rækker per kombination af rækker fra tidligere tabeller
- `range`: Index bruges til at udvælge et del-resultat
- `index`: Hele indexet bliver scannet
- `ALL`: Hele tabellen bliver scannet

Flere gode råd om forespørgsler

- Sørg for at indexes kan bruges!
`WHERE Population*1.05 > 100000;` Æ **slow** (no index)
`WHERE Population > 100000/1.05;` Æ **fast** (indexed)
- Brug udvidet form af INSERT
 - `INSERT INTO t VALUES (1,2), (3,4), (5,6);`
 - Ja, det er standard SQL!
- Udskyd DELETE / UPDATE hvis du kan
- REPLACE er hurtigere end DELETE+INSERT
- Brug LIMIT når du kan

Brug den rette tabeltype

- **MyISAM**
 - Hurtigere ved SELECT, få processer som opdaterer
 - Brug packed MyISAM til R/O data
- **HEAP**
 - Godt til midlertidige tabeller
 - Skrives ikke til disk; ekstremt hurtig
- **InnoDB**
 - Låser på rækkeniveau
 - Hurtigere ved mange processer der opdaterer samtidigt

Server tuning – globalt

- Ofte unødvendigt!
- Brug binaries fra MySQL AB!
 - Gevinsten ved selv at kompilere er tvivlsom
 - Vi kompilerer mod en optimeret GLIBC, vi tester, vi har erfaringen
 - Din tid er bedre brugt på at se på EXPLAIN SELECT...
- Vigtige globale parametre :
 - `key_buffer_size`: RAM brugt på at holde index værdier
 - `table_cache`: Antal tabeller der kan holdes åbne på een gang

Server tuning pr. connection

- Også ofte unødvendigt, men afhænger af applikationer
- `record_buffer`
 - Buffer til scanning af tabeldata
- `sort_buffer`
 - Buffer til sortering ved `ORDER BY`/`GROUP BY`
 - Evt. yderligere plads tages fra disk
- `tmp_table_size`
 - Grænsen for, hvornår midlertidige tabeller skrives til disk

OS tuning

- Filsystem: Overvej ReiserFS
 - Generelt 10-20% hurtigere
 - Mange databaser / mange tabeller pr. database
- Antal åbne filer
 - `ulimit -n (/etc/security/limits.conf)`
 - `my.cnf:`

```
[mysqld.safe]
open-files-limit=8192
```
- Antal processer / tråde
 - `ulimit -u (/etc/security/limits.conf)`
- Ingen swap

Tuning af hardware

- Første prioritet: RAM (og mere RAM)
 - Større buffere, bedre tilgang til indexes
- Anden prioritet: CPU kraft
 - Store tabeller > 2G? Overvej 64-bit
 - 64-bit beregninger bruges mange steder internt
- Tredje prioritet: Netværket
 - Latency er lige så vigtigt som throughput
- Fjerde prioritet: Diske

Tuning af diske

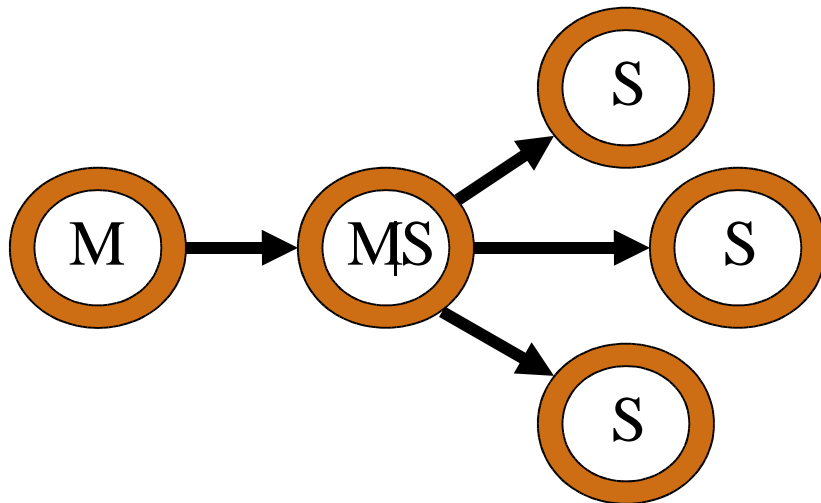
- Brug dedikerede diske
 - til logfiler og databaser, men ikke til den enkelte tabel
- Lav søgehastighed vigtigere end throughput
- Brug gerne RAID 0 eller RAID 1+0
 - men ikke RAID 5, og ikke 1+0 til data der nemt kan reableres
- Boot / mount:
 - async, noatime ved mount
 - hdparm -m16 -d1

Skalering

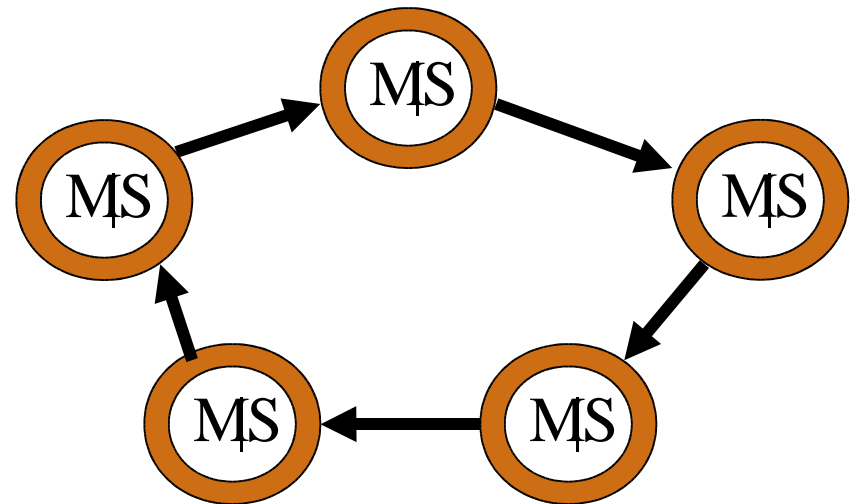
- Replikering
- Clustering

Replikering - typer

Master/Slave



Multi-Master



Replikering – lyn-HOWTO

Master:

my.cnf:

```
[mysqld]
log-bin
server-id=1
#Next is only required if the
#master is a slave as well
log-slave-updates
```

```
shell> mysqldump -master-data \
                > master.sql
```

```
mysql> GRANT REPLICATION SLAVE ON *.*
-> to rpl_user@host IDENTIFIED BY
-> 'rpl_pass';
```

Slave

my.cnf:

```
[mysqld]
server-id=2
```

```
shell> mysql < master.sql
```

```
mysql> CHANGE MASTER TO
-> MASTER_HOST='master host',
-> MASTER_USER='rpl_user',
->
MASTER_PASSWORD='rpl_pass',
-> MASTER_PORT=MySQL_port;
```

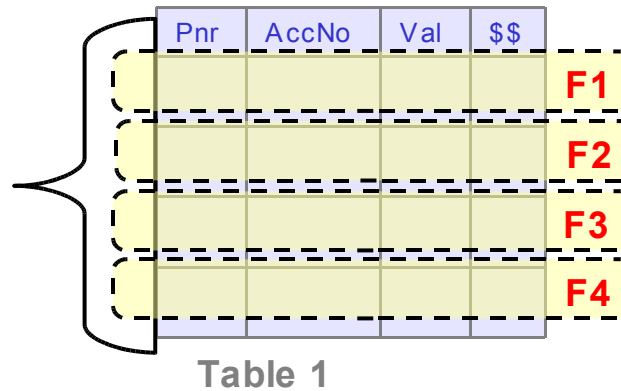
```
mysql> SLAVE START;
```

Replikering og clustering

- Replikering
 - Master/Slave
 - Asynkront
 - Vi ved ikke, hvornår data findes på en slave
 - Stadig et stort pres på master
 - Forøger, men garanterer ikke opetid
 - Vi ved ikke, om en master eller slave fejler
 - God til store distancer
- Clustering
 - Alle nodes er “master”
 - Synkront
 - Når data er gemt, er det gemt på alle nodes
 - Distribuerer load bedre
 - 99.999% opetid
 - Auto-detektering af fejl i andre nodes
 - Kræver at nodes står tæt

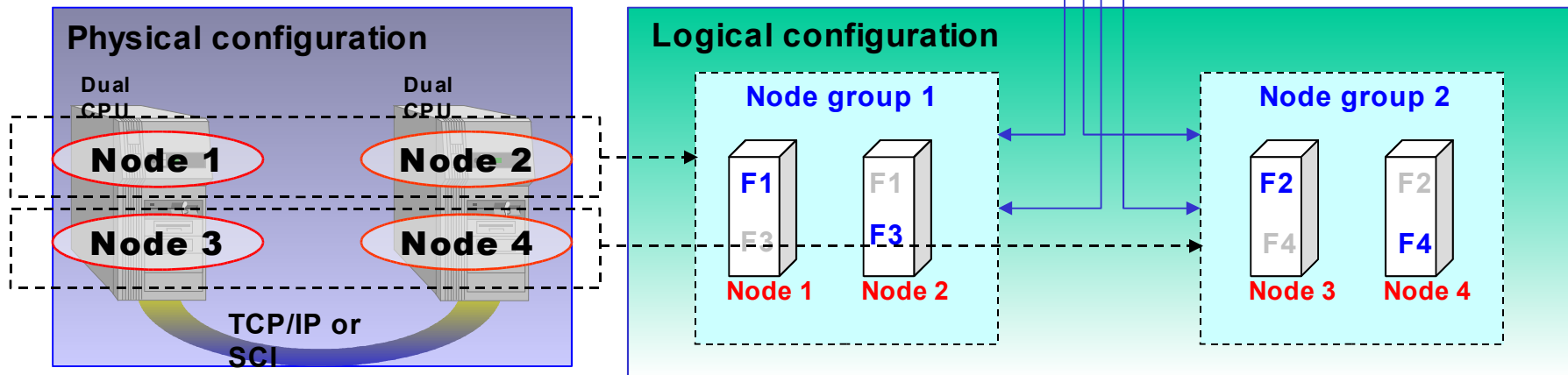
NDB clustering

- Horizontal fragmentation of Table 1 (4 fragments)
- Fragments distributed on nodes



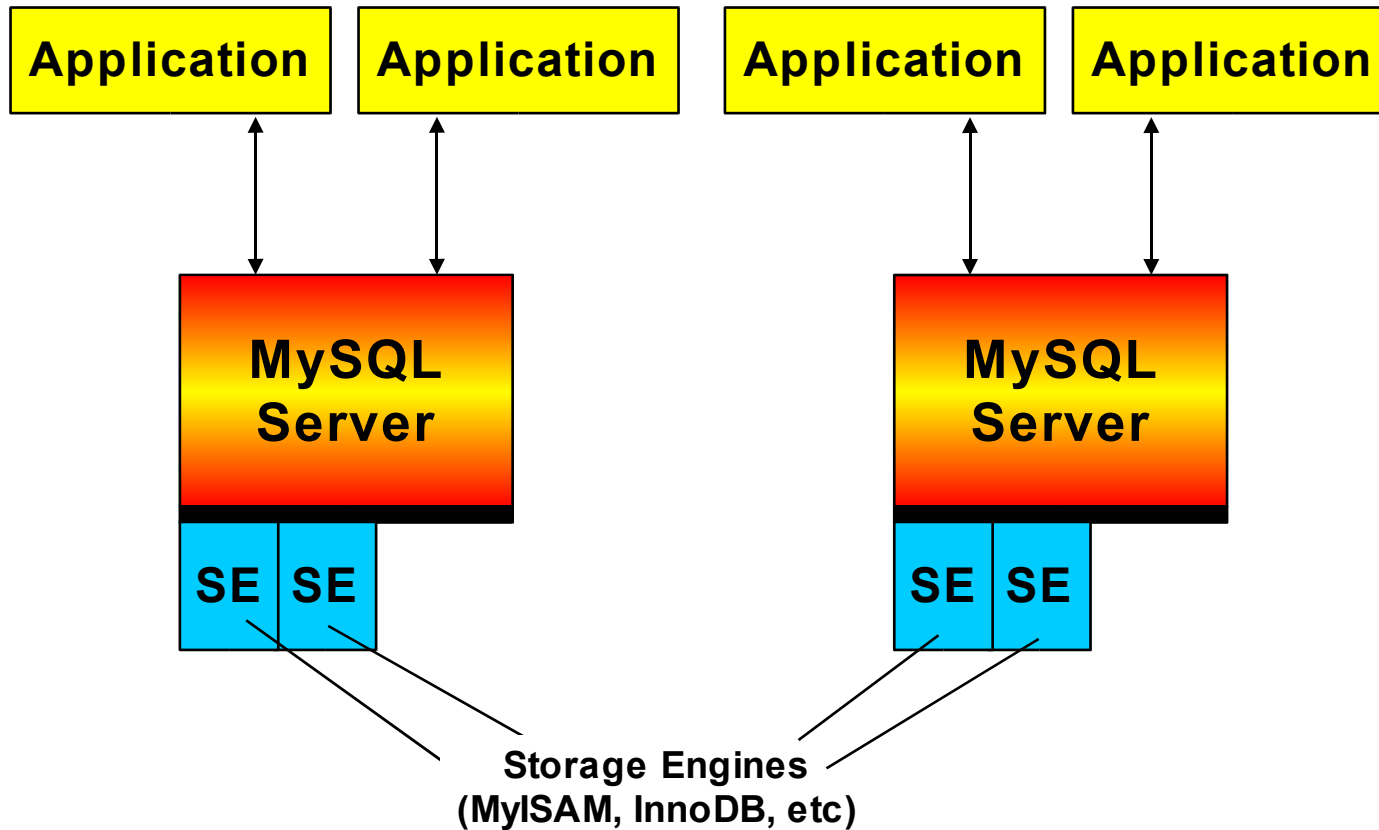
2 copies of data

Fx – primary replica
Fx – secondary replica



NDB Cluster : 4 node configuration on 2 dual processor machines

Clustering – et preview



Clustering - NDB

