

Folien zum Workshop NDO

Sammlung von Übersichten

Bernd Horn

DB-Engines MySQL

Engine	Heap	InnoDB	MyISAM
Geschwindigkeit	Sehr hoch	OK	Ca 5-10% schneller als InnoDB (ungesicherte Aussage) + 5-10% wenn „Festes Zeilenformat?“
Crashrecovery	Nein; Daten weg bei Restart MySQL/Server	Ja	Nein
Transaktionen	Nein	Ja	Nein
Fremdschlüssel-Beschränkungen	Nein	Ja	Nein
Gleichzeitiges Lesen und Schreiben	Nein	Ja	Nur, wenn in der Tabelle nichts gelöscht ist (Lücken).
Backup	Struktur ...	Online möglich; bremst Performance für die Dauer des Backups Andere Engines separat sichern ...	Blockiert Tabellen/Updates oder ist nicht konsistent.
Backup-Befehl (mit Dump).		mysqldump --single-transaction	mysqldump --lock-tables
Online-Backup (Betrieb geht weiter - gebremst)	Nein	Ja	Nein

Engines MySQL – Verwendung NDO

Engine	Heap	InnoDB	MyISAM
Verwendung für aktuelle Stati	Evtl. OK Je nach Verwendung (z.B. nur NagVis).	OK	OK
Verwendung für historische Daten	NEIN	OK	OK
Bremseffekt auf Startup Nagios	Gering	Ja	Ja – geringer als InnoDB?

DB-Eigenschaften NDO - kurz

- InnoDB als Default-Engine (genau so wie bei MySQL5).
- Je Nagios-Installation eine Anmeldung in der Datenbank für Wartungs-Aufgaben (Delete ..) und Nagios-Daten.
- Bremst Nagios; Abhängigkeiten:
 - Performance MySQL
 - Gewählte Engine (Heap ist sehr schnell)
 - DB-Cleanup historischer Daten durch NDO bremst periodisch, auch aufgrund nur einer DB-Connection !
 - Offenbar werden nicht ALLE historischen Tabellen von der NDO bereinigt
 - Truncate kann periodisch >alles< löschen
 - Verwendung von Merge-Tabellen?
 - Ab Version 5.1 können unter InnoDB „Partitions“ verwendet werden
 - ...

MySQL-Konfiguration

- QueryCache
 - Abfragen – Wiederholung
 - Gültigkeit des Cache erlöscht beim Schreiben von Daten in die betroffenen Tabellen
- Puffer
 - Jede „Engine“ hat eigene Optionen
 - Nur für Indizes
 - Verschiedene Puffer
 - Sollte sich am Volumen der DB's orientieren
- Verbindungen
 - Zahl der Verbindungen einstellen auf die zu erwartende Zahl von Verbindungen
- Offene Tabellen
 - Je Session und Zahl der verwendeten Tabellen.
- Fehlersuche und Performance-Analyse:
 - Log aller Befehle:
 - log
 - Log aller „langsamen“ Abfragen
 - log_slow_queries
 - long_query_time=1

Installation NDO

- Gemäß Beschreibung der NDO-Utills
- + Ausführen eines zusätzlichen DB-Skripts zur Anpassung der Tabellen an die Version 1.4b7.
 - Lage: [Auspack-Verzeichnis]\db\mysql-mods-1.4b7.sql
 - Ausführen:
 - Mysql aufrufen
 - In NDO-DB wechseln
 - Source [Auspack-Verzeichnis]\db\mysql-mods-1.4b7.sql

MySQL – 5 zu 4

- Verwendung mehrerer Indizes pro Select möglich
- Subqueries möglich

Konfiguration Nagios.cfg für NDO

- Bei welchen Events soll der Broker angesprochen werden?
 - `event_broker_options=-1`

- Einbindung des Event-Brokers
 - `broker_module=/usr/local/nagios/bin/ndomod-3x.o`
`config_file=/usr/local/nagios/etc/ndomod.cfg`

Konfiguration ndomod.cfg

- Nagios-Instanz?
 - instance_name=default
- Übergabe an NDO2DB definieren (TCP-Socket; NULL Security)
 - output_type=tcpsocket
 - output=127.0.0.1
 - tcp_port=5668
- Puffer, falls die Verbindung zur NDO stockt:
 - output_buffer_items=5000
 - buffer_file=/usr/local/nagios/var/ndomod.tmp
- Optionen zur Verbindung an NDO2DB
 - reconnect_interval=15
 - reconnect_warning_interval=15
- Welche Daten sollen verarbeitet werden?
 - data_processing_options=-1
Dieser Bitschlüssel ist anders als der der Brokeroptions in nagios.cfg
 - config_output_options=3

Konfiguration Ndo2DB.cfg

- Eigener Service – muß passen zu NDOMOD.cfg
 - socket_type=tcp
 - socket_name=/usr/local/nagios/var/ndo.sock
 - tcp_port=5668
 - ndo2db_user=nagios
 - ndo2db_group=nagios
- Datenbank-Verbindung
 - db_pass=nagios
 - db_servertype=mysql
 - db_host=192.168.141.130
 - db_port=3306
 - db_name=ndo
 - db_prefix=nagios_
 - db_user=nagios
- Was soll wann gelöscht werden? (1440 = 1 Tag, 10080 = 1 Woche)
 - max_timedevents_age=1440
 - max_systemcommands_age=10080
 - max_servicechecks_age=10080
 - max_hostchecks_age=10080
 - max_eventhandlers_age=44640

Bugs und Tips

- Filerotation: <http://www.nagios-portal.de/wbb/index.php?page=Thread&threadID=10011>
- Befüllung von Tabellen: <http://www.nagios-portal.de/wbb/index.php?page=Thread&threadID=9629>
- Update Programmstatus – nur bei Ext Commands:
http://www.nagvis.org/docs/general/often_reported_errors#nagiosdatanotuptodate
 - Workaround: Patch von „event.c“
- Zeitzonen müssen übereinstimmen
- Fehlerhafte Indizes: <http://www.nagios-portal.de/wbb/index.php?page=Thread&threadID=8459>
- Verwendung von HEAP-Tabellen: <http://www.nagios-portal.de/wbb/index.php?page=Thread&threadID=9357> und passenden „Data Processing Optionen“
- Löschen durch Index beschleunigen: <http://www.nagios-portal.de/wbb/index.php?page=Thread&threadID=8172>

Security NDO

- Nagios
 - Ist mit Sicherheitskonzept aufgebaut (CGI, Pipe, Benutzer und Gruppen für Service)
- NDOMod
 - Ist eingebettet in das Sicherheitskonzept Nagios
- Ndo2DB
 - **Ist bei Verwendung von TCP/IP von „jedem“ ansprechbar?**
 - Bei Verwendung der File- oder Socket-Schnittstelle greift die UNIX-Dateisicherheit.
- MySQL
 - Die Datenbank hat ein entsprechendes Sicherheitskonzept, das grundlegend auch in der Installation verwendet wird (per Empfehlung).

Performance Nagios - Aspekte

- Durch Verwendung der NDO werden zusätzliche Informationen exportiert.
 - Die Menge ist konfigurierbar; je weniger, desto weniger „Bremseffekt“.
- Der Eventbroker wartet mit der „Rückkehr zu den Nagios-Funktionen“ bis zum erfolgreichen Schreiben der Daten in die NDO
 - Bei File: Bis Anhängen der Datei
 - Bei Socket: Bis die Daten in der DB gelandet sind.
- DB-Server kann die „Wartezeiten“ des Eventbrokers ebenfalls erhöhen.
 - Eine hohe Last auf dem DB-Server ...
 - Ein langsamer DB-Server ...
 - Eine langsamere DB-Engine (aber: Recovery-Fähigkeiten von InnoDB im Crashfall ...)
 - Korrekte Speicher-Konfiguration (orientiert an Bedarf und Arbeitsspeicher, Puffer, Verbindungen, ...)
- Bei Verbindungen über TCP/IP kommen minimale Wartezeiten für die Versendung über das Netzwerk hinzu.
 - Viele Tausend „Winzigkeiten“ geben großes Delay.
 - Bei Netzwerken mit erhöhter Latenz (City-Netz? WAN?) kommt eine zusätzliche Verzögerung/Winzigkeit hinzu.
- Das Löschen der historischen Daten blockiert die DB-Connection des NDO2DB-Prozesses, so daß während der Löschzeit keine weiteren Anweisungen über diese Verbindung laufen.

Interims-Planung Ablauf Installation NDO

- Nagios/NDO-Modul schreibt in Textdatei
 - Pfad
- Verifizieren der Daten
 - Prüf-Katalog
- Konfiguration der DB (laut Original-Dokumentation + DB-Modifikations-Skript).
- Konfiguration des Ndo2DB.
 - Input per TCP/IP
 - MySQL-DB per TCP/IP
- Pipen der Textdatei in Ndo2DB
 - Es muß während des Laufs eine Verbindung im MySQL sichtbar sein. Zum Ende der Laufzeit wird die Verbindung wieder abgebaut.
 - Die Daten gemäß Katalog müssen in der DB ankommen.

Installationsnotizen zur Demo

Basis:

Knoppix

- lamp im Pfad /opt/lampp
- Nagios 3 im Pfad /usr/local/nagios

Voraussetzungen:

Libmysqlclient15 (DEB-Paket)

```
./configure --with-mysql-lib=/opt/lampp/lib/mysql/  
make
```

Anlegen einer Datenbank

```
create database nagios;  
use nagios;
```

Anlegen eines Benutzers

```
==> nagios
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON `nagios`. * TO  
'nagios'@'%' identified by 'nagios';
```

Kopieren der Dateien

```
cp src/ndomod-3x.o /usr/local/nagios/bin/ndomod.o  
cp config/ndomod.cfg /usr/local/nagios/etc
```

Editieren nagios.cfg

```
broker_module=/usr/local/nagios/bin/ndomod.o  
config_file=/usr/local/nagios/etc/ndomod.cfg  
event_broker_options=-1
```

Dito für NDO2DB

```
cp src/ndo2db-3x /usr/local/nagios/bin/ndo2db  
cp config/ndo2db.cfg /usr/local/nagios/etc  
/usr/local/nagios/bin/ndo2db -c /usr/local/nagios/etc/ndo2db.cfg
```