

Blick in die Werkzeug-Kiste eines MySQL-DBAs

Lenz Grimmer
Oracle Deutschland B.V. & Co KG
Hamburg

Keywords:

MySQL, Administration, Werkzeuge, Tools, Unix, Linux, Open Source, Replikation, Backup, Sicherheit, Benutzerverwaltung, Überwachung

Einführung

Die MySQL Distribution enthält verschiedene Hilfsmittel zur Administration des MySQL Servers auf der Kommandozeile. Diese Werkzeuge lassen allerdings viele Wünsche offen – viele administrative Aufgaben können damit nur sehr mühselig oder gar nicht bewerkstelligt werden.

Glücklicherweise wurden diese Lücken in der Zwischenzeit von diversen anderen Open-Source Projekten ausgefüllt. Diese stellen eine Vielzahl von Möglichkeiten bereit, das Leben eines MySQL-Administrators einfacher zu gestalten und bieten meist noch viel zusätzliche Funktionalität an.

Dieser Vortrag gibt einen Überblick und eine kurze Einführung in die beliebtesten und nützlichsten Werkzeuge zur MySQL-Administration. Es werden sowohl die bekanntesten Toolkits als auch einzeln verfügbare Werkzeuge vorgestellt, mit Links auf die Project-Homepages, auf denen sich üblicherweise weiterführende Informationen und die Downloads befinden. Viele dieser Tools sind auch bereits Bestandteil der gängige Linux-Distributionen und damit schnell verfügbar.

Es wird beschrieben, mit welchen Werkzeugen sich bestimmte Aufgaben wie z.B. Benutzerverwaltung, die Durchführung von Backups und Security-Audits sowie die Einrichtung und Administration von Replikationskonfigurationen besser bewerkstelligen lassen.

Dabei liegt der Schwerpunkt auf Werkzeugen, die für Linux/Unix unter einer Open-Source Lizenz verfügbar sind, da dies üblicherweise die Plattform der Wahl ist auf der MySQL im Produktionsbetrieb eingesetzt wird. Nicht Bestandteil dieses Vortrags sind „Bordwerkzeuge“ des Betriebssystems wie z.B. `vmstat`, `iostat` oder `top`, da sie keinen direkten MySQL-Bezug haben. Sie sind jedoch für eine erste Einschätzung der Lage hilfreich und gehören auf jeden Fall ebenfalls zum Repertoire eines erfahrenen MySQL-Administrators.

Generell sind freie Tools für die Windows-Plattform rar – die in diesem Vortrag vorgestellten auf Perl oder Python basierenden Werkzeuge *könnten* laufen, dies müßte jedoch im Einzelfall erfragt oder ausprobiert werden.

Unter anderem werden in diesem Vortrag die folgenden Werkzeuge und -kompilationen näher vorgestellt:

- maatkit
- Openark-kit
- Kontrollkit

- Securich
- xtrabackup
- mylvmbackup

Der Vortrag versteht sich als eine „Führung durch den Süßwarenladen“ – Sie müssen selbst entscheiden und abwägen, was Ihre Anforderungen sind und welche der Tools dafür am ehesten für Sie in Frage kommen.

Eine schamlose Schleichwerbung vorweg sei mir gestattet: es gibt natürlich auch viele kommerzielle/proprietäre Lösungen in diesem Umfeld, wie z.B. den MySQL Enterprise Monitor, der viele der in diesem Vortrag vorgestellten Themenbereiche abdeckt und mit Hilfe einer Web-basierten GUI leicht zugänglich macht.

Maatkit

Maatkit (<http://maatkit.org/>) ist die wahrscheinlich bekannteste Kollektion von Tools für MySQL. Mit über 25 verschiedenen Werkzeugen deckt Maatkit praktisch alle Aspekte der MySQL-Administration ab. Die Tools sind ausschließlich in der Scriptsprache Perl entwickelt, zu jeder Komponente gehört eine umfangreiche Test Suite sowie ausführliche Dokumentation. Besonderer Wert wird auf möglichst minimale externe Abhängigkeiten gelegt – jedes Programm ist für sich eigenständig und kann separat von der Projekt-Homepage heruntergeladen und direkt verwendet werden.

Unterstützt wird die Entwicklung von der Firma Percona, bei der Baron Schwartz und Daniel Nichter als Consultants und Entwickler angestellt sind. Maatkit besitzt eine sehr aktive Entwickler-Community und das Projekt veröffentlicht in regelmäßigen Abständen neue Versionen ihrer Kollektion. Dabei sind sie auch nicht scheu, alte Zöpfe abzuschneiden und Dinge von Grund auf neu zu implementieren, wenn es als sinnvoll erachtet wird.

Auf jede einzelne Komponente dieser Sammlung einzugehen würde leicht einen eigenen, mehrstündigen Vortrag füllen. Daher wird dieser Vortrag lediglich ein paar ausgewählte Tools hervorheben.

Openark Kit

Openark Kit (<http://code.openark.org/forge/openark-kit/>) von Shlomi Noach ist eine weitere populäre und nützliche Sammlung von Werkzeugen zur Erleichterung der MySQL-Administration. Zur Zeit besteht die veröffentlichte Kollektion aus ca 13 Programmen, die in der Scriptsprache Python implementiert wurden. Diese unterliegen der BSD-Lizenz, sind ausführlich dokumentiert und werden aktiv weiterentwickelt.

Im Subversion-Repository befinden sich weitere hilfreiche Scripts, die noch in der Entwicklung sind. Wie auch bei Maatkit würde eine Behandlung aller Komponenten den Rahmen dieses Vortrages sprengen. Daher werden nur einige dieser Tools aufgeführt, es empfiehlt sich jedoch, alle Komponenten von Openark Kit zu evaluieren.

Kontrollkit

Kontrollkit (<http://kontrollsoft.com/software-kontrollkit/>) ist die dritte „große“ Tool-Kollektion für MySQL, die man als MySQL-Admin kennen sollte. Früher auch unter dem Namen „Monolith Toolkit“ bekannt, ist Kontrollkit mittlerweile auf ca. 15 eigenständige Werkzeuge angewachsen. Weiterentwickelt werden sie von Matt Reid, der auch der Kopf hinter dem freien web-basierten MySQL-

Monitoring Tool „Kontrollbase“ ist. Die Tools sind in unterschiedlichen Sprachen geschrieben, von Shell über Perl zu Python ist alles vertreten. Interessant sind u.a. Scripts zum Erstellen von parallelen Backups oder die Überwachung der MySQL-Replikation.

Hack MySQL Kit (hmk)

Als letzter im Bunde soll auch das unter der GPL stehende Hack MySQL Kit (<http://hackmysql.com/>) nicht unerwähnt bleiben. Es wurde ursprünglich von Daniel Nichter entwickelt, der das Projekt mittlerweile allerdings zu Gunsten von Maatkit eingestellt hat. Nichtsdestotrotz ist HMK erwähnenswert, da die 4 Werkzeuge (in Perl und C implementiert) sehr sauber dokumentiert sind und viele praktische Beispiele enthalten. Da es sich nur um vier Tools handelt, will ich sie hier kurz näher vorstellen:

`mysqlreport` verarbeitet die Ausgabe von `SHOW STATUS` zu einem Report, der ein Verständnis dafür geben soll, wie gut ein MySQL-Server läuft und konfiguriert ist. Es ist eine bessere Alternative dazu, die Ausgabe von `SHOW STATUS` selbst zu interpretieren.

`mysqlsla` verarbeitet, filtert, analysiert und sortiert MySQL Slow-, General, und Binärlogs und erstellt daraus einen anpaßbaren Report der Abfragen und ihrer Meta-Eigenschaften. Die Ausgabe des Reports erfolgt wahlweise in menschen- oder maschinenlesbarer Form, z.B. um sie an andere Skripte wie `mk-query-profiler` zur weiteren Analyse weiterzureichen.

Der Zweck von `mysqlidxchk` ist, MySQL Datenbanken und Tabellen auf nicht verwendete Indizes zu untersuchen. Es benötigt ein oder mehrere MySQL Slow- oder General Logs, um damit die bei Abfragen verwendeten Indizes mit den tatsächlich vorhandenen zu vergleichen.

`mysqlsniffer` ist ein `tcpdump`-Clone, speziell zum Auslesen/Mitschneiden und Überwachen des MySQL TCP/IP Verkehrs. Er ist in C geschrieben und verwendet die `pcap` Bibliothek.

Benutzerkonten-Verwaltung

Insbesondere in Mehrbenutzer-Umgebungen (z.B. einer Entwicklungsabteilung) kommt es doch häufiger vor, daß neue Benutzerkonten in MySQL angelegt werden müssen. Das Erstellen, Duplizieren und Verwalten von Benutzerkonten mit den `CREATE USER` oder `GRANT/REVOKE` ist mühselig. Auch ist es nur mit Umständen möglich, ein Benutzerkonto temporär zu sperren, ohne es komplett zu entfernen. Weiterhin unterstützt MySQL von Haus aus keine Rollen oder Benutzergruppen, jeder Benutzer muß seine Rechte und Privilegien individuell zugeteilt bekommen.

Securich

Diese Einschränkungen in der Benutzerverwaltung bewegten Darren „MySQL Preacher“ Cassar dazu, Securich (<http://www.securich.com/>) zu entwickeln. Dieses Werkzeug ist besonders in Umgebungen hilfreich, in denen „echte“ Benutzer (z.B. Entwickler oder Hosting-Kunden) direkt mit der Datenbank arbeiten müssen und sich die Benutzerkonten häufiger ändern.

Für die Implementierung wählte Darren einen interessanten Ansatz – Securich wurde komplett mit Hilfe von Stored Routines innerhalb der MySQL Datenbank realisiert. Dadurch ist das Tool sehr portabel, es läuft auf allen Plattformen auf denen auch MySQL läuft. Stored Routines sind seit MySQL 5.0 verfügbar, so daß die Voraussetzungen/Anforderungen sehr niedrig sind (MySQL 5.1 wird empfoh-

len). Securich bietet eine ganze Reihe von Funktionen, um die Administration von Benutzerkonten leichter zu gestalten. Anstatt neue Konten direkt mit `CREATE USER` oder dem `GRANT`-Kommando zu erstellen, ruft der MySQL-Administrator die entsprechende Stored Routine mit den benötigten Parametern auf.

Mit Hilfe von Securich ist es möglich, Rollen mit bestimmten Privilegien zu definieren. Diese Rollen können dann einzelnen Benutzerkonten zugewiesen werden. So ist es z.B. sehr leicht möglich, eine Rolle „Entwickler“ zu erstellen, die nur über Privilegien auf den Entwicklungsdatenbanken verfügt. Einem neuen Entwickler können so sehr einfach die erforderlichen Rechte zugewiesen werden.

Weiterhin lassen sich Benutzerkonten zu Gruppen zusammenfassen, die wiederum bestimmte definierte Privilegien besitzen können. Bestehende Benutzerkonten können außerdem „geclont“ werden, d.h. ein neuer Benutzer „erbt“ alle Privilegien eines anderen bei der Erstellung. Auch der Entzug von Privilegien oder das vollständige Blockieren eines Kontos ist mit wenigen Handgriffen erledigt.

Darüberhinaus bietet Securich einige Methoden zur Verbesserung der Kennwort-Sicherheit. Im Gegensatz zu MySQL können bestimmte Regeln festgelegt werden, denen ein Benutzerkennwort entsprechen muß. So können z.B. die Mindestlänge und die Komplexität (Groß- und Kleinschreibung, Sonderzeichen) festgelegt werden, die ein Kennwort haben muß. Securich kann weiterhin eine Kennwort-Historie führen, so daß bereits verwendete Paßwörter abgewiesen werden, falls der Anwender sie erneut benutzen will. Auch ein Ablaufdatum kann für Paßwörter definiert werden, so daß Anwender gezwungen sind, in regelmäßigen Abständen ihr Paßwort zu ändern. Für die Änderung des Kennworts ist der Anwender nicht auf die Hilfe des Administrators angewiesen – er kann einfach selbst die entsprechende Securich-Routine aufrufen.

Neben der Variante, die Stored Routines direkt auf der Kommandozeile aufzurufen gibt es darüberhinaus auch noch eine web-basierte Variante. Darren arbeitet an SAM-My (<http://code.google.com/p/sam-my/>), ein in PHP geschriebenes Frontend.

Oak-block-account

Das aus Shlomi Noach's OpenArk Kit stammende Werkzeug bietet eine in MySQL lang vermißte Funktionalität, nämlich das temporäre Sperren von bestimmten Benutzerkonten ohne die Privilegien des Kontos verändern zu müssen. Dabei ist es möglich, den Zugang eines Benutzers von allen Hosts oder nur einem bestimmten Host zu sperren. Das gesetzte Kennwort bleibt dabei erhalten und wird beim Aufheben der Sperrung wieder auf den ursprünglichen Wert zurückgesetzt. Beim Setzen einer Kontosperrung kann der Administrator bestimmen, ob eine eventuell bestehende Verbindung zur Datenbank ebenfalls sofort getrennt werden soll.

Dies ist hilfreich, um z.B. eine „Amok laufende“ Applikation von der Datenbank abzuklemmen, oder um Benutzerkonten zu sperren, die zu einem späteren Zeitpunkt wieder aktiviert werden sollen. Oak-block-account erfordert `UPDATE`-Privilegien auf der `mysql.user` Tabelle, welche üblicherweise dem root-Benutzer vorbehalten sind.

Überwachung

Im Folgenden werden einige Werkzeuge vorgestellt, die eine Überwachung des MySQL-Servers zur Laufzeit ermöglichen. Hierbei handelt es sich um standalone-Lösungen, die Statistiken und Laufzeit-

Parameter erfassen und darstellen können. Es gibt darüberhinaus noch eine Unmenge an weiteren Monitoring-Lösungen mit MySQL-Unterstützung aus dem Open-Source Umfeld – die folgende Übersicht ist bei weitem nicht vollständig.

Innotop

Ähnlich wie das System-Überwachungswerkzeug `top` stellt Innotop den Zustand eines oder mehrerer MySQL-Server auf der Kommandozeile in einem sich stetig aktualisierenden Fenster dar. Das Tool wurde ursprünglich von Baron Schwartz entwickelt, der sich beim Design sehr stark von Jeremy Zawodny's `mytop` inspirieren ließ. Innotop geht jedoch weit über das hinaus, was `mytop` an Parametern und Werten anzeigt. Ein besonderer Schwerpunkt liegt hierbei (wie auch schon der Name vermuten läßt) auf der Überwachung der InnoDB Speicher-Engine.

Das in Perl geschriebene Werkzeug ist sehr anpaß- und erweiterbar und verfügt über eine Plugin-Schnittstelle, die es leicht macht, eigene Erweiterungen zu erstellen. Innotop überwacht unter anderem den Zustand der MySQL Replikation, aktuell ausgeführte SQL-Abfragen sowie InnoDB-relevante Informationen wie z.B. laufende Transaktionen, offene Tabellen, auf locks wartende Threads, Verletzungen von Fremdschlüssel-Beziehungen, Speicher-Puffer, Log-Dateien, I/O-Operationen, Auslastung und vieles mehr.

Mycheckpoint

Ebenfalls aus der Hand von Shlomi Noach, dem Autor von OpenArk Kit, stammt das MySQL-Überwachungswerkzeug `mycheckpoint` (<http://code.openark.org/forge/mycheckpoint/>). Das unter der BSD-Lizenz stehende und in Python entwickelte Script legt einen besonderen Schwerpunkt auf der leichten Verfügbarkeit der Überwachungsdaten und arbeitet SQL-basiert: Charts, Reports und erweiterte Metriken werden „on the fly“ mit Hilfe von VIEWS generiert – es werden keine externen Programme zur Diagnose und Auswertung der Daten benötigt.

Hierbei können sowohl lokale als auch entfernte Hosts über das Netzwerk überwacht werden, die Daten werden zentral in einer MySQL-Tabelle gespeichert. Dabei überwacht `mycheckpoint` auch bestimmte Parameter des (Linux-) Betriebssystems wie CPU (Auslastung), Speicher (RAM und Swap) und System-Auslastung (Page I/O). Weiterhin können benutzerdefinierte SQL-Abfragen hinzugefügt werden. Herz von `mycheckpoint` ist eine Tabelle mit mehreren hundert Spalten, welche die meisten der GLOBAL STATUS, GLOBAL VARIABLES, MASTER STATUS und SLAVE STATUS Metriken enthalten. Die Aufzeichnung der Daten erfolgt über ein einzelnes INSERT-Kommando, welches den Zustand des Systems zu diesem Zeitpunkt festhält. Dabei besteht keine Notwendigkeit, daß diese Messung von `mycheckpoint` selbst durchgeführt wird – die Werte können prinzipiell über jedes beliebige Tool oder Kommando in die Tabelle geschrieben werden.

Wenn die Daten einmal aufgezeichnet sind ist es einfach, daraus komplexere Darstellungen und Diagnosen, wie z.B. Charts (via Google charts) oder HTML-Reports zu erstellen. Diese Reports können auch direkt über den eingebauten http-Server zugänglich gemacht werden. Alternativ können auch simple SELECT-Abfragen über die Daten ausgeführt werden. Es ist auch möglich, Alarm-Bedingungen zu definieren um z.B. via E-Mail auf neue oder behobene Warnhinweise hingewiesen zu werden.

Datensicherung/Backup

Die Sicherung sowie die Wiederherstellung der Datenbanken und Tabellen eines MySQL-Servers gehört mit Sicherheit zu den am häufigst wiederkehrenden und wichtigsten Aufgaben eines verantwortungsvollen MySQL-Administrators. Im folgenden werden einige Werkzeuge vorgestellt, die diese Aufgabenstellung deutlich erleichtern und beschleunigen.

Mk-parallel-dump / mk-parallel-restore

Diese beiden Werkzeuge sind Bestandteil der Maatkit-Kollektion und dienen der schnellen und komfortablen Erstellung von logischen Sicherungskopien der MySQL-Tabellen. Im Gegensatz zum „klassischen“ `mysqldump` werden bei `mk-parallel-dump` die Tabellen parallel in mehreren Threads gesichert, was einen deutlichen Geschwindigkeitsvorteil mit sich bringt. Pro Datenbank wird ein Unterverzeichnis angelegt, in dem sich pro Tabelle eine SQL-Datei befindet. Die eigentliche Sicherung findet im Hintergrund entweder via `mysqldump` oder `SELECT INTO OUTFILE` statt.

Diese Dumps können „on the fly“ komprimiert werden und lassen sich bei Bedarf in Teile definierter Größe aufsplitten. Beim Aufsplitten verwendet das Tool keine feste Größe, sondern orientiert sich an den indizierten Daten und versucht, eine sinnvolle Aufteilung vorzunehmen. `Mk-parallel-dump` unterstützt reguläre Ausdrücke, mit denen sich sehr detailliert festlegen läßt, welche Tabellen/Datenbanken gesichert oder übersprungen werden sollen. Die Sicherung kann auch auf bestimmte Speicher-Engines (z.B. MyISAM oder InnoDB) begrenzt werden. Ebenfalls wird die aktuelle Position im Binärlog des Servers festgehalten, was für die Wiederherstellung mit Hilfe von Binärlogs (point-in-time recovery) oder dem Aufsetzen eines MySQL Replikations-Slaves sehr hilfreich ist.

Mydumper

Ähnlich wie Maatkit's `mk-parallel-dump` versucht das in C geschriebene `mydumper` (<https://launchpad.net/mydumper/>) die Sicherung von MySQL-Tabellen dadurch zu beschleunigen, in dem der Inhalt von mehreren Tabellen parallel gesichert wird. `Mydumper` verwendet dafür jedoch nicht `mysqldump` im Hintergrund, sondern benutzt die MySQL Client-Bibliothek `libmysqlclient` um mit dem Server Kontakt aufzunehmen.

Ein besonderer Augenmerk wurde auf die leicht zu verarbeitende Ausgabe dieses Werkzeugs gelegt, was die Integration beispielsweise in Shell-Skripts vereinfachen soll. Wichtig: `mydumper` sichert nur die **Inhalte** der Tabellen, es handelt sich also um reine **Daten-Dumps**. Um die Schema-Definitionen zu sichern, können Sie z.B. `mysqldump --no-data` verwenden. Um die Auswahl der zu sichernden Datenbanken und Tabellen steuern zu können unterstützt `mydumper` reguläre Ausdrücke. Weiterhin versucht `mydumper` die Konsistenz des Backups durch setzen eines globalen Write Locks zu gewährleisten und führt die Sicherung als Transaktion mit konsistentem Snapshot durch. Auch die Binärlog-Position des Masters und eventueller Slaves wird im Snapshot festgehalten.

Xtrabackup

Eine weitere Entwicklung aus dem Hause Percona ist das Online-Backuptool `xtrabackup` (<https://launchpad.net/percona-xtrabackup/>). Das unter der GNU GPL stehende Programm ermöglicht eine Online (nicht-blockierende) Sicherung von InnoDB (und XtraDB) Tabellen im laufenden Betrieb des MySQL-Servers. Seit April 2010 unterstützt das Programm auch Windows als Betriebssystemplattform. Die Backups können komprimiert und als Stream auf Band oder an über das Netzwerk einen entfernten Rechner übertragen werden. Auch die Erstellung inkrementeller Backups ist möglich.

Primär ist Xtrabackup für die Sicherung von InnoDB-Tabellen gedacht. Mit Hilfe des Wrapper-Scripts `innobackupex` lassen sich auch MyISAM-Tabellen zusammen mit InnoDB-Tabellen sichern (unter Verwendung des Kommandos `FLUSH TABLES WITH READ LOCK`).

MyLvmbackup

Das unter der GPL stehende Perl-Script `mylvmbackup` (<http://www.lenzg.net/mylvmbackup/>) wird vom Autor dieses Vortrags betreut und weiterentwickelt. Es verwendet die Snapshot-Funktionalität des Linux Volume-Managements LVMv2, um konsistente Abbilder eines MySQL-Datenverzeichnisses zu erstellen. MyISAM-Tabellen werden vorab mit Hilfe des Kommandos `FLUSH TABLES WITH READ LOCK` in einen konsistenten Zustand versetzt. Im Anschluß daran wird der Snapshot erstellt und die Tabellen wieder freigegeben. Üblicherweise dauert dieser Vorgang nur wenige Sekunden an, so daß Applikationen meist nicht in ihrem Ablauf gestört werden.

Der Snapshot wird im Anschluß daran an anderer Stelle ins Dateisystem eingebunden und kann nun mit Hilfe „herkömmlicher“ Werkzeuge wie `tar`, `rsync` oder `rsnap` gesichert werden. Die Archiv-Dateien werden automatisch mit Zeitstempeln versehen. Das Script ist sehr flexibel konfigurierbar und automatisiert diesen Prozeß vollständig.

Auditing/Sicherheit

Von Zeit zu Zeit ist es sinnvoll, die sichere Konfiguration eines MySQL-Servers zu überprüfen, um eventuelle Sicherheitsrisiken auszuschließen. Das folgende Werkzeug leistet wertvolle Hilfestellung dabei.

Oak-security-audit

Aus Shlomi Noach's OpenArk Kit stammt dieses kleine Helferlein, daß man am besten per Cron-Job in regelmäßigen Abständen laufen läßt. Oak-security-audit prüft die Konfiguration eines MySQL-Servers auf folgende sicherheitsrelevante Aspekte ab:

Benutzerkonten

- Anonyme Benutzerkonten (z.B. "@'localhost')
- Überprüfung auf wildcard hosts (“<username>@%”)
- Nicht-lokale root-Konten
- Konten mit leeren oder identischen Passwörtern

Privilegien

- Meldet nicht-root Konten mit Administrator-Privilegien (z.B. SUPER, SHUTDOWN)
- Meldet nicht-root Konten mit weitreichenden datenverändernden Privilegien (z.B. DROP, ALTER)
- Meldet nicht-root Konten mit datenverändernden Privilegien für alle Schemata
- Meldet nicht-root Konten mit Schreibrechten auf die `mysql` Datenbank

Allgemein

- Überprüft sicherheitsrelevante `sql_mode`
- Prüft, ob pre-MySQL 4.1 Authentifizierung (“old passwords”) verwendet wird
- Prüft, ob `networking` aktiviert ist
- Prüft, ob die `test`-Datenbank existiert

Performance

Die Analyse der Laufzeitparameter eines MySQL-Servers kann recht komplex sein. Um MySQL optimal auf die System-Umgebung und die Anforderungen der Applikation einzustellen bedarf es einiges an Erfahrung. Die folgenden Werkzeuge leisten wertvolle Hilfestellung bei dieser Aufgabe.

MySQL Tuning Primer

Das MySQL Tuning Primer Shell-Skript von Matthew Montgomery (<https://launchpad.net/mysql-tuning-primer/>) verwendet die Informationen aus `SHOW STATUS LIKE...` und `SHOW VARIABLES LIKE...` um einen Überblick über die aktuellen Parameter eines MySQL-Servers zu erhalten. Basierend auf diesen Werten versucht es dann, sinnvolle Empfehlungen zum Optimieren diverser Server-Konfigurationsvariablen zu geben. Die Ausgabe der Werte wird übersichtlich und farbig formatiert.

Im Moment spricht das Skript Empfehlungen für die folgenden Variablen aus:

- Slow Query Log
- Max Connections
- Worker Threads
- Memory Usage
- Key Buffer
- Query Cache
- Sort Buffer
- Joins
- Temp Tables
- Table (Open & Definition) Cache
- Table Scans (read_buffer)
- Table Locking
- InnoDB Status

MySQL Tuner

Die Entwickler des Perl-Skripts `mysqltuner` (<https://launchpad.net/mysqltuner/>) wurden vom MySQL Tuning Primer inspiriert. Es untersucht ebenfalls die Konfiguration und Parameter eines MySQL-Servers mit Hilfe von `SHOW GLOBAL VARIABLES` und `SHOW GLOBAL STATUS`, geht allerdings noch einen Schritt weiter. Zur Berechnung der Empfehlungen werden neben den Server-Variablen auch noch einige Parameter des Betriebssystems hinzugezogen, z.B. die Größe des Hauptspeichers (RAM und swap), sowie die Architektur (32 oder 64 bit).

Das Format der Ausgabe ist konfigurierbar; neben einer gut lesbar aufbereiteten Version ist es alternativ möglich, den Inhalt als CSV-Datei ausgeben zu lassen. MySQL Tuner kann weiterhin im „offline mode“ betrieben werden, bei dem der MySQL-Server nicht direkt kontaktiert und die Werte abgefragt werden, sondern diese Information aus einer Textdatei ausgelesen wird. Dies ist z.B. bei der Unterstützung von Kunden hilfreich, wenn man keinen direkten Zugriff auf das System hat, den Kunden aber um die Ausgabe dieser Kommandos bitten kann. Generell läßt sich feststellen, daß das Skript deutlich schneller zu Werke geht als das original Tuning Primer Shellskript.

MySQL Report

`mysqlreport` ist Teil von Daniel Nichter's am "Hack MySQL Toolkit", das am Anfang des Vortrags bereits erwähnt wurde. Das Programm (ein sehr ausführlich dokumentiertes Perl-Skript) erstellt einen Report der wichtigsten MySQL Status-Werte. Im Gegensatz zum `SHOW STATUS` Kommando, das einfach über 100 Variablen als lange Liste auf dem Bildschirm ausgibt, interpretiert, formatiert und bereitet `mysqlreport` diese Werte zu einem leicht zu lesenden Report auf.

Die begleitende Dokumentation „The Guide to Understanding `mysqlreport`“ ist eine sehr nützliche, allgemeine Einführung in die Grundlagen des MySQL-Tunings und unbedingt lesenswert, wenn man sich näher mit der Materie auseinandersetzen will: <http://hackmysql.com/mysqlreportguide/>

Mext

Das unter der BSD-Lizenz stehende Shellskript `mext` ist Teil des Aspersa-Toolkits (<http://code.google.com/p/aspersa/>). Aspersa ist eine Sammlung einfacher, aber nützlicher Werkzeuge. Diese sind in diversen Skriptsprachen geschrieben, und darauf ausgerichtet, bestimmte Aufgaben/Tätigkeiten zu beschleunigen. Der Name kommt von "Helix Aspersa", der gefleckten Weinbergschnecke. Die meisten dieser Tools wurden von Consultants bei Percona für den persönlichen Gebrauch entwickelt.

`Mext` führt das Kommando `mysqladmin extended-status` aus und formatiert die Ausgabe. Es ist auch möglich, die gespeicherte Ausgabe dieses Kommandos aus einer Datei zu lesen. Die Ausgabe kann in inkrementeller, differentieller oder relativer Form erfolgen. So lassen sich Veränderungen der MySQL-Parameter zur Laufzeit sehr gut beobachten und analysieren, was insbesondere bei Problemen, die nur sporadisch auftauchen, hilfreich sein kann.

Replikation

MySQL-Replikation wird für verschiedene Zwecke eingesetzt, z.B. zur Erhöhung der Redundanz, Leistungssteigerung (horizontale Skalierung) oder zu Backup-Zwecken. Die Replikations-Technologie ist Bestandteil des MySQL-Servers und relativ simpel aufgebaut. Aufgrund ihrer asynchronen Natur ist es erforderlich, den Stand der Replikations-Server zu überwachen. Die folgenden Werkzeuge helfen, ein MySQL-Replikations-Setup zu überwachen.

Mk-heartbeat

`Mk-heartbeat` ist Teil von `Maatkit` und verwendet separate Heartbeat-Tabelle, um die ordnungsgemäße Funktion der MySQL-Replikation zu überwachen. Hierzu aktualisiert das Script in im Sekundentakt einen Zeitstempel in der Heartbeat-Tabelle auf dem Master und mißt mit Hilfe einer zweiten Instanz die Verzögerung, mit der dieser Zeitstempel auf einen Slave repliziert wird. Dies setzt voraus, daß die Systemzeit auf Master und Slave synchronisiert ist (z.B. mittels NTP).

Mk-table-checksum

`Maatkit's mk-table-checksum` generiert Prüfsummen über MySQL-Tabellen, um schnell zu überprüfen ob Tabellenkopien auf einem MySQL-Slave identisch zu denen auf dem Master-Server sind. Die Checksummen werden hierbei vom MySQL-Server mit Hilfe des Kommandos `CHECKSUM TABLE` selbst erstellt, dadurch hält sich der benötigte Netzwerkverkehr in Grenzen.

Dabei können Tabellen auf einem oder mehreren Servern parallel überprüft werden, das Script konsolidiert die Ausgaben und vergleicht die Ergebnisse. Zur Erstellung der Prüfsummen können verschiedene Algorithmen verwendet werden, standardmäßig wird CRC-32 eingesetzt. Je nach Tabellengröße und Anforderungen an die Laufzeit und Verlässlichkeit der Aussage können auch andere Methoden wie z.B. MD5 oder SHA1 verwendet werden.

oak-get-slave-lag

Dieses zum OpenArk Kit gehörige Tool liest den Status der Replikation auf einem Master-Server aus und iteriert dann durch `SHOW SLAVE STATUS`. `oak-get-slave-lag` verbindet sich direkt mit dem Slave-Server. Das Script gibt die Verzögerung in Sekunden aus und liefert einen Fehlercode, wenn ein bestimmter Schwellwert erreicht wird, d.h. ein Slave zu weit in der Replikation zurückgefallen ist.

oak-show-replication-status

Dieses Script zeigt an, wie weit Replikations-Slaves einem bestimmten Master hinterher eilen.

Das Skript nimmt Verbindung zu einem Master Server auf, liest dessen Master-Status aus, lokalisiert die aktiven Slaves, verbindet sich dann mit diesen Slaves und liest ihren Status aus. Das Werkzeug berücksichtigt einen Schwellwert für die Verzögerung und berichtet die folgenden Informationen:

- Das aktuelle Master Logfile
- Wenn Slaves fehlen (siehe die Option `--expect-num-slaves`), wird eine Fehlermeldung ausgegeben.
- Für jeden aktiven Slave: sein Master Logfile, Verzögerung hinter dem Master in Sekunden und der Status.

Replikations-Wartung

Regelmäßiges Bereinigen/Löschen der Master-Logs ist notwendig, um freien Plattenplatz auf dem Master-Server zu behalten. Die MySQL Replikation wird jedoch unterbrochen, wenn Logs auf dem Master gelöscht werden, die noch nicht auf dem Slave repliziert wurden (z.B. wegen Verzögerung der Replikation oder temporären Ausfällen).

Oak-purge-master-logs

Löscht Master-Logdateien unter Rücksichtnahme auf den Stand der Replikations-Sklaven.

`oak-purge-master-logs` berücksichtigt den Kompromiß zwischen den Speicherplatzanforderungen auf dem Master und den Bedürfnissen der Replikation zu den Slaves. Das Tool kennt drei Betriebsarten:

`--pro-master`: Master bekommt die höhere Priorität. Master Logs werden gelöscht, selbst wenn einige oder alle Slaves noch nicht "up to date" sind. Wenn der Plattenplatz auf dem Master knapp ist, ist dies eventuell die richtige Einstellung.

`--pro-slaves`: Slaves bekommen die höhere Priorität: `oak-purge-master-logs` löscht nur Logs die von keinem Slave mehr benötigt werden. Dies stellt sicher, daß alle Slaves weiterhin sicher replizieren können. Dies ist die richtige Einstellung, wenn genügend Plattenplatz auf dem Master verfügbar ist.

`--pro-master` mit `--pro-slaves`: Die Kombination beider Optionen stellt sicher, daß zumindest ein Slave alle Daten repliziert hat, die auf dem Master gelöscht werden. Dies bedeutet daß zumindest ein Slave-Server noch weiter laufen kann, wenn `oak-purge-master-logs` gelaufen ist. Dies ist die richtige Einstellung, wenn genügend Platz auf dem Master verfügbar ist und zumindest von einem Slave-Server bekannt ist, daß er ordnungsgemäß funktioniert, während andere für einen längeren Zeitraum nicht mehr erreichbar sein könnten. Die Replikation auf einem dieser anderen Slaves wieder aufzusetzen ist relativ leicht und ohne Unterbrechung des Betriebs möglich, wenn ein laufender Replikations-Slave vorhanden ist.

mk-slave-move

Dieses Maatkit-Tool verschiebt einen MySQL-Slave innerhalb der Replikationshierarchie. Es kann Slave-Server unter Berücksichtigung der Replikations-Position voneinander trennen und wieder miteinander verbinden. Dies ist komfortabler und fehlerfreier als wenn man die dafür notwendigen Kommandos (z.B. `CHANGE MASTER TO . . .`) per Hand eingibt.

mk-slave-find

Dieses Skript aus der Maatkit-Kollektion ist hilfreich, wenn man sich schnell ein Bild von einer bestehenden (evtl. komplexen) MySQL-Replikationsstruktur machen will. Es befragt einen MySQL Replikations-Master nach den Slave-Servern, kontaktiert diese in rekursiver Form und erstellt eine baumartige Ansicht der Replikationshierarchie. Das Ausgabe-Format des Reports ist konfigurierbar.

Kontakt:

Lenz Grimmer

Oracle Deutschland B.V. & Co. KG

Döringweg 7a

22529 Hamburg

Phone: +49(0)40-4126 7308

Fax: +49(0)40-4126 7311

Email lenz.grimmer@oracle.com

Internet: <http://www.mysql.com/>