

MySQL for Oracle DBAs

Philipp Michaly
Trivadis GmbH
München

Schlüsselworte

MySQL, Oracle, DBAs

Einleitung

Opensource Datenbanken halten seit geraumer Zeit vermehrt Einzug in Enterprise IT Umgebungen. Sie erweisen sich in vielzähligen Projekten als attraktive Alternative zu den etablierten kommerziellen RDBMS. Dabei überzeugen Sie nicht alleine durch den günstigen Preis, sondern bieten auch aus technischer Sicht viele interessante Möglichkeiten.

In diesem Manuskript wird der Fokus auf die MySQL Enterprise Edition gelegt und beleuchtet, wie es sich mit der MySQL Architektur verhält, wie die Lizenzpolitik sich gestaltet, aber auch wie die Marktforschung die gegenwärtige und zukünftige Relevanz von Open Source RDBMS beurteilt.

Zum Abschluss werden noch einige MySQL Best Practices vorgestellt, die im Rahmen eines Proof of Concept angewandt wurden.

1. Überlegungen zu Open Source Datenbanken im Allgemeinen

Als Consultant für Oracle und MySQL Datenbanken, ist der Autor täglich bei unterschiedlichsten Kunden in verschiedensten Branchen tätig. Da bleibt es nicht aus, dass man mit den Kunden über die Zukunft der aktuellen Oracle RDBMS ins Gespräch kommt. Bei einigen Kunden bemerkt man in diesen Gesprächen, dass seitens der Fachbereiche und auch des IT Managements immer wieder Opensource Datenbanken ins Gespräch gebracht werden oder bereits zur Implementierung angedacht sind. Bei anderen Kunden sind Opensource Datenbanken wiederum schon Normalität, sei es ein paralleler Betrieb zur Oracle RDBMS oder gar ein Projekt um komplett zu Opensource Datenbanken zu migrieren. Die Erfahrungen und Eindrücke des Autors sind natürlich subjektiv und daher nicht repräsentativ zu verstehen, spiegeln aber möglicherweise eine Stimmung in der Branche wider.

Was sagen also aktuelle Marktforschungsstudien zu diesem Thema?

Von Gartner gibt es aus dem Jahr 2015 eine sehr aufschlussreiche Studie zu diesem Gebiet:

“The State of Open-Source RDBMSs, 2015”

Gartner kommt bei dieser Studie zu dem Resultat, dass Opensource Datenbanken Stand 2015 einen Reifegrad erreicht haben, der es ermöglicht kommerzielle RDBMS zu ersetzen und dabei bemerkenswerte Einsparungen beim TCO zu erzielen. Es wird eine Empfehlung an IT Management, DBAs und Software Entwicklungs Management ausgesprochen schnellstmöglich auf Open Source Datenbanken zu migrieren.

Gartner kommt in dieser Studie zu zwei weiteren Annahmen, diesmal das Jahr 2018 betreffend: Die Autoren der Studie nehmen an, dass für das Jahr 2018 bereits 70% aller neuen in-house Applikationen

auf Open Source Datenbanken basieren werden, sowie 50% aller existierenden kommerziellen RDBMS bereits zu Open Source RDBMS migriert wurden, oder sich in Migrationen befinden werden.

Die Prognosen gehen von einer massiven Verschiebung des Datenbankmarktes aus, in Form einer Abwendung von kommerziellen Lösungen hin zu Opensource Lösungen.

Unabhängig davon wie sehr man die Einschätzung der Autoren dieser Studie persönlich teilt, erscheint es dennoch ratsam sich mit Open Source Datenbanken auseinanderzusetzen; auch als Oracle DBA.

1.1 MySQL Enterprise Edition

War der Blickwinkel bisher allgemeiner gehalten, wird der Fokus nun auf die MySQL Enterprise Edition gelegt. Es gibt einige gute Argumente sich für dieses ODBMS zu entscheiden:

Enterprise Support:

Für Unternehmen mit einer tief verankerten Oracle DB Umgebung gibt es die Möglichkeit, in Form der MySQL Enterprise Edition, eine Opensource DB einzusetzen ohne sofort alle Enterprise Support Modelle über Bord werfen zu müssen. Mit der MySQL Enterprise Edition steht der gewohnte Oracle 24/7 Support zur Verfügung, ebenso auch Zugriff auf alle MOS Ressourcen. Man kann also weiterhin kritische Datenbanken hosten und im Fehlerfall eine Eskalationskette mit klaren Reaktionszeiten erwarten.

Technische Reife:

MySQL Datenbanken bewähren sich bereits seit Jahren auch im Enterprise und High-End Umfeld, sie wird von Google, Facebook, und vielen weiteren Konzernen eingesetzt. Die stete Weiterentwicklung des MySQL Kernels wird aus **technischer** Sicht von der Community ebenfalls sehr positiv gesehen. Zum Beispiel im Bereich des Cost Based Optimizers sind in den letzten Jahren deutliche Fortschritte erzielt worden.

Enterprise Tools:

Mit der Enterprise Edition kommen noch die MySQL Enterprise Tools hinzu, sie erweitern den MySQL Funktionsumfang erheblich. Zu erwähnen wäre da zum Beispiel MySQL Enterprise Backup, mit dem wesentlich flexiblere und performantere Backups erstellt werden können. Auf einige weitere Enterprise Tools wird in der Präsentation eingegangen.

2. MySQL Architektur

Die Architektur von MySQL zeichnet sich durch ihre Modularität aus. Dadurch lässt sie sich granular an die jeweiligen Anforderungen anpassen. Man kann die Datenbank dadurch entweder hochgradig spezialisieren, indem nicht benötigte Module deaktiviert werden, oder man profitiert durch die Vielzahl an vorhandenen Plugins und implementiert eine vielseitig nutzbare Datenbank. Eine weitere Besonderheit ist das „Shared Nothing Prinzip“. Dieses besagt, dass niemals mehr als **eine** MySQL Instanz auf die Daten zugreifen darf, da MySQL den Zugriff auf Datafiles durch verschiedene Instanzen nicht synchronisieren kann. Somit ist ein Active-Active Cluster mit MySQL technisch nicht möglich.

Für Hochverfügbarkeitslösungen sind dafür verschiedene Replikationstechniken vorgesehen auf die noch später eingegangen wird.

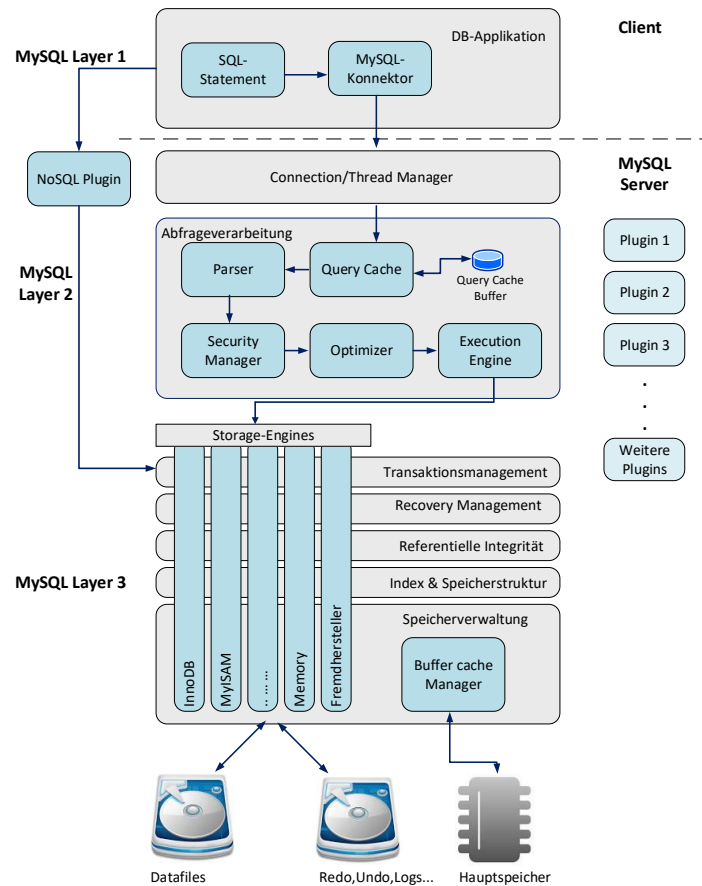


Abbildung 1: MySQL Schaubild Architektur

2.1 Beschreibung der Architektur

2.1.1 Konnektoren (MySQL-Layer 1)

Konnektoren ermöglichen die Konnektivität der Clients zur Datenbank, dabei gibt es für die wichtigsten Programmiersprachen spezifische Treiber (Konnektoren), aber auch APIs für einen direkten Zugriff. Zum Beispiel Connector/ODBC, Connector/NET, Connector/J

2.1.2 Plugin Konzept (MySQL-Layer 2)

Das sehr flexible Plugin-System, mit dem auch die verschiedenen Storage-Engines eingehängt werden können, ist eine der großen Stärken von MySQL, so kann durch externe Plugins MySQL um interessante Features erweitert werden. Seit Release 5.6 werden neue Features sowie bereits

bestehende Teile des Kernels, die selten genutzt werden, in Plugins ausgelagert. Dies ermöglicht eine sehr hohe Flexibilität, da es möglich ist beliebige Module zu aktivieren oder zu deaktivieren. Am Beispiel des NoSQL-Memcached Plugin ist es sogar möglich auf direktem Wege mit der Storage Engine zu kommunizieren. Dabei werden die Module für die normalen SQL Abfragen umgangen, ohne dass es dabei zu Wechselwirkungen kommt. Somit ist es zum Beispiel möglich in einer Datenbank die Vorteile von NoSQL und SQL zu kombinieren.

2.1.3 Storage Engines (MySQL-Layer 3)

MySQL bietet über das modulare Storage-Engine Konzept die Möglichkeit verschiedenste Storage Engines einzubinden. Eine Storage Engine ist ein in sich abgeschlossenes Modul, das über das MySQL Plugin-System eingebunden werden kann. Dabei übernimmt das Modul die gesamte Speicherung von Tabellen und bietet auch einen strukturierten Zugriff auf die Daten für die Abfrageverarbeitung an (MySQL-Layer 2).

Es gibt Storage Engines für verschiedenste Anwendungsszenarien. Wir empfehlen als Storage Engine die Verwendung von InnoDB. InnoDB ist für MySQL die Standard Engine wenn es um Transaktionssicherheit geht und zugleich die wichtigste und am häufigsten eingesetzte Storage Engine.

2.2 Replikation

Es gibt bei MySQL 3 Replikationsverfahren, auf alle 3 Verfahren wird in kompakter Form eingegangen.

2.2.1 Asynchrone Replikation

MySQL stellt Mechanismen zur asynchronen Replikation zur Verfügung. Das bedeutet, dass der Master stets alle Änderungen unabhängig von seinen Slaves durchführt. Auch wenn der Datenstand eines Slaves nicht up to date ist, werden weiterhin Änderungen am Master durchgeführt; der Slave kann dadurch hinsichtlich der Datenaktualität etwas hinterher hängen. Man ist gut beraten den Slave auf einen Apply-Delay zu überwachen („Seconds behind“). Vor einem Switchover oder Failover muss sichergestellt sein, dass die Datenstände identisch sind.

Die Vorteile der asynchronen Replikation liegen im Bereich der Performanz. Da die Master Datenbank nicht auf die Slave Datenbank warten muss, ist die Schreibgeschwindigkeit bei diesem Replikationsmechanismus nicht beeinträchtigt.

2.1.1 Semi-Synchrone Replikation

Über ein Plugin kann zusätzlich die Semi-Synchrone Replikation aktiviert werden. Die Semi-Synchrone Replikation stellt einen guten Kompromiss zwischen der Synchronen – und Asynchronen Replikation dar.

Bei der Semi-Synchronen Replikation wartet der Master bis mindestens ein Slave die Events (in Form von Binlogs oder GTIDS) erhalten und gelogged hat. Der Master wartet nicht auf alle Slaves und fordert lediglich die Bestätigung ein, dass die Events empfangen wurden. Es wird nicht geprüft ob die Events appliziert wurden.

2.2.2 Synchrone Replikation

Bei der synchronen Replikation wird ein Schreibvorgang auf dem Master erst dann beendet („Committed“), wenn mindestens ein Slave die Events ausgeführt und committed hat. Dies erhöht einerseits die Datensicherheit, andererseits wird die Schreibgeschwindigkeit bei diesem Verfahren deutlich reduziert, da der Master für seine commits stets auf Bestätigung des Slaves warten muss. Der Schreibvorgang auf den Slaves kann im jeweiligen IO Thread nur seriell erfolgen. Die synchrone Replikation kann nur im MySQL Cluster eingesetzt werden.

3 Lizenzierung

Die Lizenzierung von MySQL Enterprise Edition weist die Besonderheit auf, dass keine Lizenzgebühren anfallen, sondern stattdessen eine Support Gebühr.

Eine Vergleich Lizenz/Support Kosten von Oracle SE/EE und MySQL Enterprise Edition.

Server Setup: X86, 2 Sockel, je CPU 6 cores

	Anzahl sockel/cores	Lizenz Metrik	Lizenz Gebühr	Jährl. pro Metrk	Lizenz pro Server	Jährl. Support	Gesamt 3 Jahres Kosten
Oracle Database Standard Edition 2	2	pro sockel	\$17,500	\$3,850	\$35,000	\$7,700	\$58,100
Oracle Database Enterprise Edition	12	0.5 pro Sockel	\$47,500	\$10,450	\$285,000	\$62,700	\$473,100
MySQL Enterprise Edition	n/a	Server	n/a	\$5,000	n/a	\$5,000	\$15,000

Tabelle 1: Lizenzierung Gegenüberstellung

Der Preisvorteil für die MySQL Enterprise fällt im 3 Jahres Vergleich gegenüber der Oracle EE enorm aus. Im Vergleich mit der Standard Edition erscheint der Preisvorteil nicht mehr so exorbitant, jedoch ist die SE im Funktionsumfang deutlich eingeschränkter und daher nicht 1:1 mit der MySQL Enterprise Edition vergleichbar.

4 MySQL Best Practices in einem Proof of Concept

Wie können MySQL Best Practices in einer Oracle geprägten IT Umgebung aussehen? Lassen sich Oracle Standards übernehmen? Einige der in diesem Projekt gewonnenen Erfahrungswerte werden in diesem Manuskript kurz dargestellt. Weitere Informationen auch zu Themen wie Architekturvarianten und werden in der Präsentation ausführlicher betrachtet.

4.1 Filesystem Architektur

Ein wichtiger Aspekt beim Aufbau und bei der Integration ist das Filesystem Layout. Die MySQL default File Locations bieten ein „gewisses“ Optimierungspotential.

Idealerweise steht uns mit OFA (Oracle Flexible Architecture) eine Strukturierungsmethodik zur Hand, die sich auch auf MySQL anwenden lässt.

Tabellarisch dargestellt wie eine OFA mit MySQL realisiert werden könnte:

Verzeichnis					
u00					
	app				
		mysql			
			admin	\$dbname	log
					socket
			etc	\$dbname.my.cnf	
			product		
				5.7.14	bin
				5.7.x	
				enterprise	
					agent
					weitere enterprise tool
			3rdPartyTool (e.g. TVDTOOLBOX)		
u01	mysqldata	\$dbname			
			db/schema1		
			db/schema2		
u02	binlog	\$dbname			
			binlogs		
	redo	\$dbname			
			redologs		
	relay	\$dbname			
			relaylogs		
u03	backup	\$dbname	full		
			incremental		
			incremental_redologs		

Tabelle 2: OFA for MySQL

4.2 Installation von MySQL mit Pre-built binaries (Tarball)

Im Rahmen des Proof of Concepts haben sich die Vorteile einer manuellen Installation gezeigt. Sie ermöglicht eine maximale Flexibilität, da alle relevanten Parameter, wie z.B. Zielverzeichnisse, manuell definiert werden können. Da jede Binary Installation in sich geschlossen ist, hat man auch die Möglichkeit verschiedene Versionsstände voneinander abgekoppelt zu betreiben. Es gibt dort keine Wechselwirkungen oder gemeinsame Abhängigkeiten.

Beispiel einer manuellen Installation:

```
bin/mysqld --initialize --user=mysql --basedir=/u00/app/mysql --datadir=/u01/mysqldata/$dbname
```

4.3 MySQL TCP Port

Im Rahmen des PoC hat sich die Empfehlung ergeben, nicht die default Port Range zu wählen. Es ist empfehlenswert die Port Range 33001 bis 33999 zu wählen.

Anhand des Präfix „33“ lassen sich MySQL Prozesse leicht identifizieren.

tcp6	0	0	:::33005	:::*	LISTEN	4211/mysqld
tcp6	0	0	:::33011	:::*	LISTEN	4251/mysqld

Aus Security Gründen ist es ohnehin empfehlenswert nicht den Default Port zu verwenden, da bietet sich diese Variante gut an.

Kontaktadresse:

Philipp Michaly

Trivadis GmbH

Lehrer-Wirth-Straße, 4

D- 81829 München

Telefon: +49 (0) 089 99275930

E-Mail philipp.michaly@trivadis.com

Internet: www.trivadis.com