



MySQL Branches und Forks

Oliver Sennhauser, DOAG Datenbank Community

MySQL, die Datenbank, die das Internet antreibt, gehört heute zu Oracle. Ihren Anfang hat die Firma in der ersten Hälfte der 1990er-Jahre genommen.

Gesucht wurde eine Firma, die einen einfachen SQL-Layer für C-ISAM-Dateien schreibt. Drei gewitzte, finnische Entwickler trauten sich diese Aufgabe zu und führten in wenigen Monaten diesen Auftrag aus. Das Ganze scheint ein Erfolg gewesen zu sein, denn einige Jahre später entstand daraus eine Firma namens „MySQL“. „My“ ist der Name der Tochter eines der Entwickler, daher stammt auch der Name der Firma.

Über die Zeit wurde das Produkt immer umfangreicher und mächtiger. Der limitierte Funktionsumfang der ISAM-Storage-En-

gine wurde erweitert und es entstand die MyISAM-Storage-Engine, die heute immer noch ab und zu genutzt wird. In dieser Zeit wurde MySQL auch unter die Open-Source-Lizenz GPL v2 gestellt. Als weitere Storage-Engine hat die Berkeley-Datenbank (BDB) Eingang in MySQL gefunden, ein transaktionaler Tabellentyp, der im Unterschied zur MyISAM-Engine, die ausschließlich Table-Level-Locks macht, Block-Level-Locks beherrscht.

Kurz nach der Jahrtausendwende hat MySQL während des großen Internet-Booms Eingang in den sogenannten

„LAMP-Stack“ (Linux-Apache-MySQL-PHP) gefunden. Man brauchte zu der Zeit eine einfache, preiswerte SQL-Datenbank, um die Bedürfnisse der boomenden Internet-Industrie befriedigen zu können. Da war MySQL genau zur richtigen Zeit am richtigen Ort – der Hauptgrund für den sensationellen Erfolg dieses Produkts.

Ebenfalls in diese Zeit fällt eine weitere Storage-Engine, die ebenfalls eine kleine, finnische Firma entwickelt hat, die InnoDB-Storage-Engine. Sie war ursprünglich als kleiner, schlanker Oracle-Ersatz gedacht. Diese Storage-Engine er-

füllt die ACID-Anforderungen (Atomicity, Consistency, Isolation, Durability), ist daher Crash-sicher und skalierte dank ihres zeilenbasierten Lockings wesentlich besser als die dazumal gängige MyISAM-Storage-Engine.

Einem unaufhaltsamen Aufstieg von MySQL stand nichts mehr im Wege. Man plante bei MySQL sogar bereits einen Börsengang, zuerst an der Stockholmer Börse, dann aber mit etwas mehr Finanzappetit an der Technologiebörse NASDAQ in den USA. Doch wie so oft im Leben kommt es manchmal anders, als man denkt: Im November 2005 kündigte Oracle den Kauf von Innobase OY, der Firma hinter der InnoDB-Storage-Engine, für eine nicht näher genannte Summe an. Das Herzstück wurde aus dem MySQL-Konglomerat herausgekauft. Schlechte Karten für einen Börsengang. Dieser Tag ging in die MySQL-Annalen auch als der InnoDB-Black-Friday ein.

Das MySQL-Management ließ sich dadurch nur kurz beeindrucken. Mit einigen geschickten Schachzügen konnte man das InnoDB-Problem umgehen und es gelang im April 2008, MySQL für eine Milliarde US Dollar an Sun Microsystems zu verkaufen. Der Enthusiasmus war groß und die MySQL-Mitarbeiter wurden bei Sun freudig begrüßt. Doch die Freude währte nur kurz: Nach zahlreichen Gerüchten übernahm Oracle ein knappes Jahr später Sun Microsystems. MySQL gehörte damit ebenfalls zu Oracle. InnoDB und MySQL waren somit wieder unter demselben Dach vereint.

MariaDB

Bis zu diesem Zeitpunkt war die Welt für die MySQL-Nutzer noch einfach und in Ordnung. Es gab nur eine Quelle der Wahrheit: MySQL aus dem Hause Oracle. Doch einigen Mitarbeitern, den ursprünglichen Gründern von MySQL, passte diese Entwicklung nicht (nachdem man eine Milliarde US Dollar abkassiert hatte). Da MySQL unter der Open-Source-Lizenz GPL v2 steht, ist jedermann dazu berechtigt, den öffentlich zugänglichen Quellcode von MySQL zu nehmen, einzusehen, zu ändern und sogar wieder zu verteilen, sofern man sich an die Regeln der GPL-v2-Lizenz hält. Dies wurde auch von besagten Entwicklern prompt getan.

Sie nahmen den MySQL-Quellcode und brachten ihn, ursprünglich nur leicht modifiziert, unter dem neuen Namen „MariaDB“ heraus. Diesen Vorgang bezeichnet man technisch auch als „Branchen“.

An diesem Punkt stellt sich die Frage, wie Oracle oder eine Firma MariaDB Geld mit MySQL verdient, um die Entwickler, Supporter, QA-Tester, Berater, Trainer, Manager, Verkäufer, Marketing-Leute und das ganze restliche Fußvolk zu bezahlen? Bei einer klassischen Software-Firma (nicht Open Source) geschieht dies primär über das Lizenzgeschäft und sekundär über das Support-Abo-Geschäft. Weitere Dienstleistungen wie Schulung und Beratung sind bedingt durch ihre relativ geringen Gewinnmargen von 20 bis 40 Prozent nicht sonderlich lukrativ und daher nicht so beliebt. Außerdem skalieren die letztgenannten Bereiche nicht besonders gut und werden eher als Kundenbindungs-Instrument gesehen. Sie sind quasi nur Enabler für das eigentliche Lizenz- und Support-Abo-Business.

Was die Mehrheit der MySQL-Nutzer nicht weiß, ist, dass Oracle das Recht hat, MySQL auch unter einer Oracle-proprietären Nicht-GPL-Lizenz zu vertreiben. Dieses Recht hat seinen Ursprung schon bei der alten Firma MySQL, die sich in weiser Voraussicht von allen Community-Kontributoren eine Abtretung des Verwertungsrechts in einem sogenannten „Contributor License Agreement“ (CLA) hat zusichern lassen. Daher ist Oracle heute die einzige Firma, die für MySQL auch Geld durch Lizenzen schöpfen darf. Alle anderen Mitbewerber müssen sich ihren Umsatz durch Dienstleistungen im MySQL-Umfeld erwirtschaften.

Die Theorie besagt, dass Open-Source-Firmen ihren Ertrag nicht durch Lizenzen erwirtschaften (mit einer Marge von gegen 90 Prozent), sondern ausschließlich durch Dienstleistungen wie Support-Abos (Marge 60 bis 80 Prozent), Schulung (Marge 40 Prozent) und Beratung (Marge 20 Prozent); ein gewisser Nachteil im hart umkämpften Software-Markt.

Somit haben die Firma MariaDB und weitere Dienstleister im MySQL-Umfeld heute nur die Möglichkeit, mit Support-Subskriptionen, Schulung und Beratung Geld zu verdienen. Es ist daher klar, dass diese Firmen sich, wenn möglich, auf das lukrativste Geschäft konzentrieren,

die Support-Subskriptionen. Mit dem dadurch erwirtschafteten Geld müssen nicht nur die Mitarbeiter entlohnt, sondern es muss zusätzlich auch noch in die Weiterentwicklung des Produkts investiert werden. Da das Lizenzgeschäft entfällt, muss entsprechend effizienter mit den Ressourcen gehaushaltet werden, um mit den Mitbewerbern, die nicht aus der Open-Source-Szene kommen, mithalten zu können.

Neben den beiden wichtigsten Branchen MySQL und MariaDB gibt es noch zahlreiche weitere Branchen und Forks. Viele davon sind allerdings nicht mehr relevant oder gar existent und daher meist nur für Software-Archäologen von Interesse. Eine spannende Entwicklung ist zudem, dass einige große Web-Firmen (Facebook, Google etc.) einen eigenen MySQL-Branch für ihre eigenen Bedürfnisse pflegen. Auch im fernen Asien scheint es Firmen zu geben, die ihre eigenen MySQL-Branches haben (wie AliSQL), die wir hier im Westen, wenn überhaupt, höchstens vom Namen her kennen. Spannend wäre noch zu wissen, von was wir noch, sprach- und schriftbedingt, keine Kenntnis haben.

Einige dieser Entwicklungen von Fremdfirmen fließen auch wieder zurück zu den eigentlichen Software-Entwicklungshäusern (Upstream) mit dem Wunsch, dass diese Fremd-Entwicklungen dann dort in den Softwarehäusern gepflegt und weiterentwickelt werden. Dies nennen einige Exponenten der MySQL-Szene dann auch „Community Contributions“. InnoDB und BDB waren die ersten namhaften Beiträge der Community. Heute sind es vor allem Firmen wie Facebook, Google, Booking.com, Alibaba etc., die solche Beiträge leisten.

Das Problem dabei ist, dass diese Firmen ihr Hauptgeschäft nicht im Entwickeln von Software sehen und daher ihre Beiträge gerne an die Software-Hersteller abgeben möchten. Diese sehen sich jedoch mit dem zusätzlichen Wartungsaufwand konfrontiert, der aber nur für wenige Endnutzer wirklich relevant ist. Die meisten MySQL-Nutzer spielen nicht in derselben Liga wie Facebook und Alibaba – also keine Einkünfte durch das Lizenzgeschäft, aber zusätzlichen Aufwand durch die Pflege von Fremd-Entwicklungen. Das geht voraussichtlich auf die Dauer nicht auf.

Das Open-Core-Modell

MySQL verfügt über zahlreiche APIs (Schnittstellen) für Erweiterungen. Diese bieten die Möglichkeit, MySQL unabhängig von der Kernfunktionalität modular zu erweitern. Ein Beispiel ist das Storage-Engine-API, das eine definierte Schnittstelle für das Anbinden weiterer Storage-Engines anbietet. Storage-Engine-Plug-ins, die dieses API nutzen, sind InnoDB, RocksDB, Connect, Federated-X etc. Daneben umfasst MySQL auch APIs für Volltext-Suche, Replikation, Auditing, Authentifizierung, Passwortvalidierung, Query Rewriting, Keyring-Anbindung etc. Für die meisten davon sind auch bereits Plug-ins verfügbar.

Dies führt zu einem neuen Geschäftsmodell in der Open-Source-Welt: dem Open-Core-Modell. Ein bereits zu MySQL-Zeiten praktizierter Ansatz, Nutzer der kostenlosen MySQL-Datenbank mit subskriptionspflichtigen, kommerziellen Erweiterungen zum Abschluss eines lukrativen Support-Abos zu motivieren. Das

daraus resultierende Produkt nennt sich MySQL Enterprise Server.

Erweiterungen für solche Enterprise-Kunden sind MySQL Enterprise Monitor für das Überwachen der MySQL-Datenbank; das MySQL Enterprise Backup für das Sichern und Zurückspielen von großen Datenbeständen; MySQL Enterprise Security mit Encryption, Audit, SQL Firewall und dem MySQL Enterprise Thread Pool für Anwendungen, die Tausende von Verbindungen gleichzeitig ermöglichen müssen. Die Mitbewerber versuchen natürlich, die spannenden Erweiterungen nachzubauen, um die Gunst der Kunden nicht zu verlieren.

Einen Schritt weiter hat sich MariaDB vor einigen Monaten gewagt. Das selbstentwickelte Router- und Proxy-Produkt MaxScale wurde in der neuesten Version unter die Business Source License (BSL) gestellt. Diese besagt, dass Installationen zu Testzwecken und für bis zu drei Installationen kostenlos sind. Größere Installationen sind kostenpflichtig und um den Gedanken von Open Source nicht zu vernachlässigen,

fallen diese BSL-Produkte nach einer Dauer von drei Jahren automatisch unter die GPL-v2-Lizenz. Der Gedanke dahinter ist: Der Kunde hat Investitionsschutz dank Open Source. Nach spätestens drei Jahren ist er nicht mehr an den Software-Hersteller gebunden und dieser hat die Möglichkeit, durch Lizenzgeschäfte mehr Umsatz zu generieren, um so mehr in die Software-Entwicklung zu investieren oder die Gewinne zu optimieren.

Ein Plug-in, das in den letzten sieben Jahren mehr und mehr Zuspruch gefunden hat, ist ein Produkt namens „Galera“, von einer kleinen, finnischen Firma namens „Codership“. Es implementiert eine synchrone Replikation (genauer gesagt eine virtuell synchrone Replikation) zwischen verschiedenen MySQL-Servern. So entsteht das erste Mal die Möglichkeit, parallel auf verschiedene MySQL-Knoten gleichzeitig zu schreiben und von ihnen zu lesen. Dieses Produkt, „Galera Cluster für MySQL“ genannt, ist insbesondere heute, wo zunehmend 7 x 24 Stunden Verfüg-

Flexible SLAs. Weil Ihre IT einmalig ist.



dbi FlexService für Datenbanken und Middleware: Ihr flexibles, kosteneffizientes und ISO 20000-zertifiziertes Service Management. Profitieren Sie insbesondere von unserem dedizierten Service Desk für validierte Systeme im Pharma-Bereich.

Phone +41 32 422 96 00 · Basel · Nyon · Zürich · dbi-services.com



Infrastructure at your Service.



barkeit verlangt wird, genau das, was sich viele Administratoren wünschen. Keine Nacht- und Wochenend-Arbeiten mehr. Wartungen können im laufenden Betrieb während der regulären Arbeitszeiten unter Last ausgeführt werden.

Fazit

MySQL hat eine bewegte Vergangenheit. Dank ihrer flexiblen Architektur und Erweiterbarkeit wird MySQL immer mehr und neue Funktionalität relativ schnell und einfach anbieten können. Dank der Open-Source-Lizenz GPL v2 dürfen sozusagen alle mitmischen, die dazu Lust und Laune haben. Stand heute gibt es zwei namhafte Player: MySQL von Oracle und MariaDB. Bei so vielen Freiheitsgraden stellt sich natürlich die Frage: Wo geht die Reise hin? Voraussichtlich wird auch hier der zweite Hauptsatz der Thermodynamik zum Tragen kommen. Salopp gesagt: Die Natur strebt das größtmögliche Chaos an ...

Bereits jetzt ist eine zunehmende Inkompatibilität zwischen den beiden Branchen MySQL und MariaDB erkennbar. Dies hat man zuerst bei der Replikation in der Version 5.5 gesehen. Später sind Änderungen am Optimizer und Parser hinzugekommen. Zahlreiche Erweiterung bei beiden Produkten tragen ebenfalls zur zunehmenden Inkompatibilität bei.

Man sieht dies auch an den verwendeten Formulierungen: Früher hieß es „Drop-in Replacement“, heute nur noch „File- oder Schnittstellen-kompatibel“ (was genau damit gemeint ist, wird nicht näher spezifiziert), und man findet zunehmend Hinweise zu Migrationsvorgängen. In naher Zukunft, also bei MariaDB 10.2 oder 10.3 und MySQL 8.0, muss damit gerechnet werden, dass diese beiden Produkte nicht mehr ohne Weiteres durch das jeweils andere substituiert werden können. Es wird dann, wie bei anderen Datenbank-Produkten auch, einen Migrationsprozess brauchen, der sich über die Zeit zunehmend aufwändiger gestalten wird.

Wer weiß heute noch, dass das Produkt Microsoft SQL Server auf dem Code von Sybase ASE Server basiert? Und wer würde heute noch wagen zu behaupten, dass diese beiden Datenbank-Systeme noch kompatibel zueinander sind? Genau diese Entwicklung steht uns bei MySQL und MariaDB ebenfalls bevor. Streng dem Marketing-Leitsatz folgend: „Differenziere dich von deinem Mitbewerber ...“



Oliver Sennhauser
oliver.sennhauser@doag.org

Oracle sagt: Fachabteilungen und IT-Teams müssen besser zusammenarbeiten

Laut der Oracle-Studie „IaaS: Building a business that innovates“ verhindern in vielen Betrieben bestehende Prozesse und Strukturen eine reibungslose interne Zusammenarbeit. Die Studie hat untersucht, wie Unternehmen in der EMEA-Region das Thema „Cloud Computing“ angehen und welche organisatorischen Faktoren signifikanten Veränderungen und Innovationen im Wege stehen. Befragt wurden 1.200 IT-Entscheider in mittleren bis großen Unternehmen. Die Ergebnisse:

- 65 Prozent der Befragten in Deutschland (62 Prozent in EMEA) sind der Ansicht, dass Fachabteilungen und IT-Teams zusammenarbeiten müssen, damit ein Enterprise-Cloud-Modell erfolgreich sein kann
- 45 Prozent (34 Prozent in EMEA) geben an, dass Schatten-IT (bei der Fachbereiche Technologien ohne Wissen der IT-Abteilung einkaufen) einen integrierten Cloud-Ansatz erheblich erschwert

- 49 Prozent (41 in EMEA) sagen, dass mangelndes Verständnis der Fachabteilungen für die Notwendigkeit von integrierten Cloud-Ressourcen eine der aktuellen IT-Infrastruktur-Herausforderungen für den Cloud-Einsatz ist.

Im aktuellen Plattform-Modell der Unternehmen gibt es einiges zu optimieren. Als besonders ineffizient nennen die Befragten die folgenden Bereiche: die Integration von verwendeten Services (50 Prozent in Deutschland und EMEA), die Bereitstellungszeit (53 Prozent in Deutschland, 49 Prozent in EMEA) und die schlechte Applikationsintegration (44 Prozent in Deutschland, 47 Prozent in EMEA). Alle drei Faktoren sind entscheidend für effektive und schnelle Innovationen.

Ein integrierter Cloud-Ansatz kann die Kollaboration verbessern. Dem stehen jedoch einige Hürden im Weg: 45 Prozent der Befragten in Deutschland (34 Prozent in EMEA) sehen das Management von

Schatten-IT als Hindernis. 31 Prozent (34 in EMEA) kämpfen mit Infrastrukturen, die nicht zusammenpassen. Auch größere Sicherheitsrisiken (36 Prozent in Deutschland, 34 Prozent in EMEA), erhöhte Kosten (30 Prozent in Deutschland, 32 Prozent in EMEA) und der ROI-Nachweis (26 Prozent in Deutschland, 31 Prozent in EMEA) werden als Hürden genannt.

Mark Borgmann, Cloud Sales Leader Oracle Deutschland kommentiert: „Verantwortlich für diese Probleme ist unter anderem eine Infrastruktur, die festgefahren, konfus und komplex ist. Das führt zu einer zersiedelten Organisation, die nicht schnell genug reagieren kann. Schlecht integrierte Systeme und abgegrenzte Fachabteilungen bremsen die Innovationsfähigkeit. Mit Infrastructure-as-a-Service (IaaS) können Unternehmen die Basis für echte Kollaboration schaffen. Sie ist das Fundament für Innovation, indem sie Ressourcen gemeinsam verfügbar macht und Silos auflöst.“