MySQL Cluster – Wann brauche ich das?

Mario Beck
Oracle
Berlin

Schlüsselworte
MySQL, MySQL Cluster, Hochverfügbarkeit, Skalierbarkeit, NoSQL

Einleitung
MySQL bietet für höchste Anforderungen an die Verfügbarkeit und Skalierbarkeit die MySQL Cluster Carrier Grade Edition. Mit dieser shared-nothing Cluster Architektur für MySQL Datenbanken sind herausragende Leistungen erzielbar. Wir betrachten die Besonderheiten von MySQL Cluster, Einsatzszenarien und Praxisbeispiele. Außerdem werden Alternativen zu MySQL Cluster diskutiert, um MySQL Cluster in verschiedene Hochverfügbarkeitsszenarien einordnen zu können. Nach dieser allgemeinen Einführung wird die aktuelle Version des MySQL Cluster technisch detailliert vorgestellt. Dabei wird auch ein Ausblick auf den aktuellen Entwicklungsstand gegeben.

Allgemeines zu Hochverfügbarkeit

Höhere Verfügbarkeitsgrade sorgen für eine wesentliche Verkürzung von Ausfallzeiten, denen Anwendungen im Verlauf eines Jahres unterliegen. Höhere Verfügbarkeitsgrade werden erreicht, indem Systeme mit höheren Redundanz- und Fehlertoleranzgraden bereitgestellt werden. Da für höhere Redundanzgrade jedoch zusätzliche Hardware- und Softwarekomponenten bereitgestellt werden müssen, steigen auch die Gesamtkosten und die Komplexität des Systems. Darüber hinaus sind größere Investitionen im Hinblick auf IT-Mitarbeiter, Prozesse und Dienste erforderlich.

Um hochverfügbare Datenbankdienste bereitzustellen, können verschiedene Architekturen eingesetzt werden, die sich durch die jeweils gebotene Betriebszeit unterscheiden. Diese Architekturen lassen sich in drei Hauptkategorien gliedern:

- Datenbankreplikation (wird zumeist über ein Cluster lose gekoppelter Server implementiert).
- Eng gekoppelte Cluster und virtualisierte Systeme.
- Shared-Nothing-Cluster mit geografischer Replikation.

Wie in der Abbildung unten gezeigt, bieten diese Architekturen unterschiedliche Verfügbarkeitsgrade. Da eine höhere Verfügbarkeit potenziell mit höheren Kosten und mehr Komplexität einhergeht, müssen diese Faktoren gegeneinander abgewägt werden.

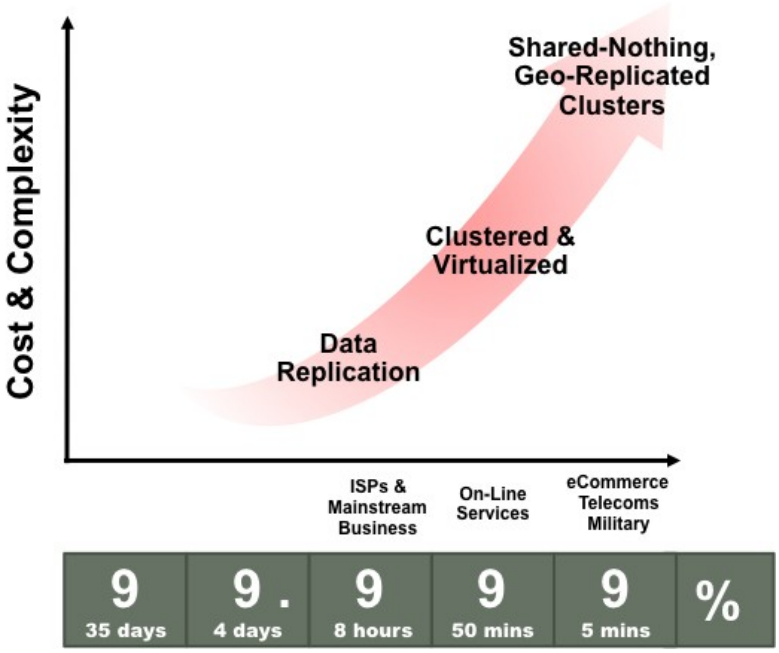

Abb. 1: Verhältnis zwischen Hochverfügbarkeitsarchitekturen und erreichbarer Verfügbarkeit

Replikation

Replikation ist der gängigste Ansatz zur Bereitstellung von Hochverfügbarkeit für MySQL und ein systemeigenes Feature der Datenbank, das standardmäßig ohne komplexe Add-Ons oder Optionen zur Verfügung steht.

Bei der Replikation kopiert oder dupliziert eine Datenbank in einem lose gekoppelten Cluster Änderungsvorgänge von einem zu einem anderen physischen Standort oder System (in der Regel vom Master- zum Slave-System). Dies erfolgt zur Erhöhung der Verfügbarkeit und Skalierbarkeit einer Datenbank. Zudem können Benutzer auch Datensicherungen oder analytische Abfragen über die Slave-Systeme flexibel durchführen, um die Master-Systeme von diesen Aufgaben zu entlasten.

Bei einer Hochverfügbarkeitsbereitstellung werden Datenbankaktualisierungen von einem Master- auf einen Slave-Server repliziert, um bei einem Ausfall des Master-Servers (aufgrund eines Fehlers, Absturzes oder von Wartungsarbeiten) ein Failover auf den Slave-Server zu ermöglichen. MySQL ermöglicht eine Replikation sowohl intern im Rechenzentrum als auch rechenzentrenübergreifend und somit eine rasche Wiederherstellung nach einem Notfall. Die MySQL Replikation umfasst keinen automatischen Failover-Mechanismus, der jedoch mithilfe verschiedener Mechanismen auf Anwendungs- oder Betriebssystemebene implementiert werden kann.

Es gibt zwei Replikationsmodi in MySQL: asynchron und semi-synchron. (Später wird auch synchrone Replikation mit MySQL Cluster diskutiert.)

Failover Cluster

Zum Erzielen höherer Verfügbarkeitsgrade bei sehr wichtigen Anwendungen müssen Systeme unter

Verwendung von Heartbeat-Mechanismen und Clusterressourcen-Managern in enger gekoppelten Failover-Clustern bereitgestellt werden. Diese Mechanismen überwachen Hardware-, Betriebssystem-, Netzwerk und Datenbankprozesse und sorgen bei Erkennen einer Störung für ein automatisches Failover auf Standby-Server. Durch die Bildung von Clustern wird in einer Infrastruktur für Redundanz gesorgt und die Anzahl singulärer Fehlerquellen verringert, wodurch Ausfallzeiten verkürzt werden und ein Schutz vor Datenverlusten gewährleistet ist.

Wenngleich diese Technologien mit eigenen Cluster-Mechanismen arbeiten, können sie auch mit der MySQL Replikation kombiniert werden, um bei Bedarf horizontal skalierte Slave-Systeme und geografische Redundanz bereitzustellen. Benutzer kommen dadurch in den Genuss von MySQL Mastern mit hoher Verfügbarkeit, die mit der MySQL Replikation gekoppelt werden, um Abfragen auf mehrere Systeme zu verteilen.

Oracle zertifiziert und unterstützt derzeit die folgenden Failover- und Cluster-Lösungen für MySQL Server, die nachfolgend beschrieben werden:

- Oracle VM Template für MySQL Enterprise Edition
- Windows Server-Failoverclustering für MySQL Enterprise Edition
- Solaris Cluster für MySQL Enterprise Edition

Zum Gewährleisten einer vollständig automatischen Wiederherstellung nach einem Failover sollten Benutzer die Speicher-Engine InnoDB (ab MySQL 5.5 die Standard-Engine) nutzen. Dadurch ist sichergestellt, dass bei einem Failover Transaktionsdaten vollständig wiederhergestellt werden können.

MySQL Cluster

Die zuvor vorgestellten Methoden zum Erreichen einer hohen Verfügbarkeit erfüllen die Anforderungen an die Betriebszeit vieler Anwendungen. Es gibt jedoch Kategorien von Diensten, die stark auf Transaktionen ausgelegt und aktualisierungsintensiv sind und eine nahezu kontinuierliche Verfügbarkeit erfordern. Beispiele sind E-Commerce- und Finanztransaktionen, Fakturierung, Benutzerzugriffs-/Authentifizierungssysteme, Netzwerkinfrastrukturanwendungen und netzwerkinterne Telekommunikationsdienste.

In diesen Fällen bietet MySQL Cluster die Vertrautheit und Benutzerfreundlichkeit des herkömmlichen MySQL Servers mit einer Verfügbarkeit von 99,999 % (d. h. weniger als 5 ½ Minuten Ausfallzeit pro Jahr) gekoppelt mit einer automatischen Partitionierung für eine hohe Schreibleistung und kurze Wartezeiten. MySQL Cluster hat sich in Umgebungen mit den höchsten Ansprüchen an die Verfügbarkeit bewährt und stellt bei Ausfällen, Notfällen und geplanten Wartungsvorgängen Dienste weiter zur Verfügung.

MySQL Cluster ist eine transaktionale ACID-konforme Echtzeitdatenbank mit Unterstützung für die Skalierung von Schreibvorgängen, die eine Verfügbarkeit von 99,999 % mit den niedrigen Gesamtbetriebskosten einer Open-Source-Lösung verbindet. MySQL Cluster hat eine verteilte Multi-Master-Architektur ohne singuläre Fehlerquelle und kann horizontal auf Standardhardware skaliert werden, um lese- und schreibintensive Arbeitslasten zu verarbeiten. Der Zugriff erfolgt über SQL- und NoSQL-Schnittstellen.

Das Echtzeitdesign von MySQL Cluster liefert vorhersagbare Reaktionszeiten von wenigen Millisekunden sowie die Fähigkeit, Millionen von Vorgängen pro Sekunde zu verarbeiten.

Unterstützung für im Arbeitsspeicher oder auf Festplatte abgelegte Daten, automatische Datenpartitionierung mit Lastverteilung und die Möglichkeit des Hinzufügens von Knoten zu einem laufenden Cluster ohne Ausfallzeit ermöglichen eine lineare Skalierbarkeit der Datenbank, sodass selbst unvorhergesehene Arbeitslasten problemlos verarbeitet werden können.

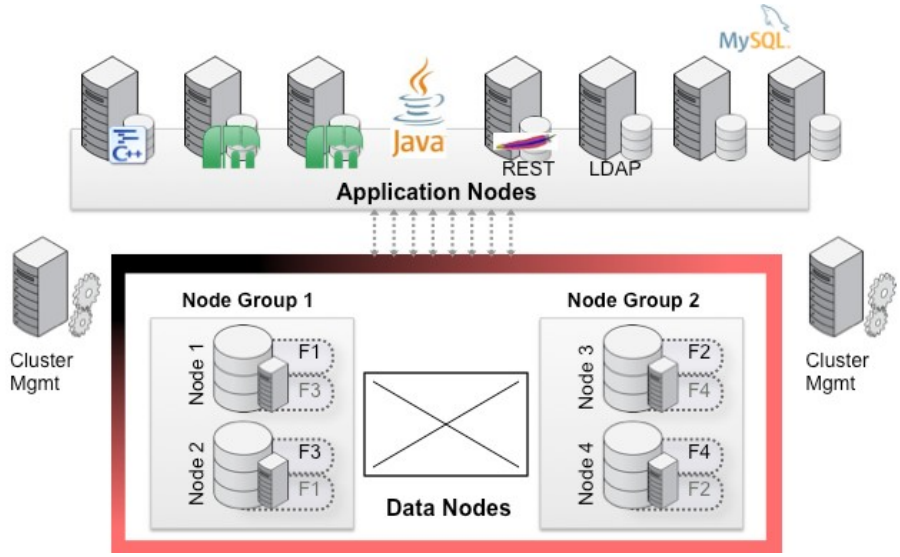

Abb. 2: Dank der verteilten Architektur werden singuläre Fehlerquellen vermieden.

MySQL Cluster umfasst drei Knotentypen, die zusammen Hochverfügbarkeit für die Anwendung gewährleisten:

Datenknoten dienen zum Verwalten der Speicherung von und des Zugriffs auf Daten. Tabellen werden automatisch datenknotenübergreifend partitioniert, die auch unbemerkt die Lastverteilung, Replikation und Selbstreparatur sowie das Failover übernehmen. Zusätzliche Heartbeat- oder Ressourcen-verwaltungs-Middleware ist nicht erforderlich, da alle diese Funktionen direkt in MySQL Cluster integriert sind.

Anwendungsknoten verbinden die Anwendungslogik mit den Datenknoten. Der Anwendung werden mehrere APIs zur Verfügung gestellt. MySQL bietet eine standardmäßige SQL-Schnittstelle und Konnektivität mit allen führenden Webentwicklungssprachen und Frameworks. Ferner gibt es auch eine Palette von NoSQL-Schnittstellen, z. B. Memcached1, REST/JSON, C++ (NDB-API), Java und JPA.

Verwaltungsknoten dienen zum Konfigurieren des Clusters und bieten Vermittlungsdienste bei einer Netzwerkpartition, um ein „Split-Brain“-Szenario zu vermeiden, was zu Dateninkonsistenzen führen würde.

Die verteilte Shared-Nothing-Architektur von MySQL Cluster wurde mit Sorgfalt entwickelt, um eine Ausfallsicherheit und Wiederherstellung zur Selbstreparatur sicherzustellen:

Die Daten in einem Datenknoten werden auf allen Knoten der Knotengruppe synchron repliziert. Wenn ein Datenknoten ausfällt, ist immer mindestens ein weiterer Datenknoten mit denselben Informationen verfügbar.

Bei Ausfall eines Datenknotens können der Server mit MySQL oder der Anwendungsknoten Transaktionen über die anderen Datenknoten in der Knotengruppe ausführen. Die Anwendung

wiederholt die Transaktion, woraufhin die verbleibenden Datenknoten die Anforderung erfolgreich erfüllen.

MySQL Cluster ermittelt Ausfälle umgehend und übergibt die Steuerung an die aktiven Clusterknoten – ohne Dienstunterbrechungen für die Clients. Bei einem Ausfall können die MySQL Cluster Knoten sich selbst reparieren, indem sie sich automatisch neu starten, wiederherstellen und dynamisch neu konfigurieren, was von der Anwendung vollständig unbemerkt erfolgt.

Um beim Ausfall eines einzelnen Verwaltungsservers weiterhin alle Verwaltungs- oder Vermittlungsfunktionen ausführen zu können, ist die Bereitstellung doppelter Verwaltungsserverknoten möglich.

Da singuläre Fehlerquellen bei diesem Cluster-Design minimiert werden, stellen Sie ein zuverlässiges und hochverfügbares System bereit. Der Ausfall eines Knotens hat keine Auswirkungen auf das Gesamtsystem.

Zusätzlich zur Hochverfügbarkeit, die auf Standortebene durch die redundante Architektur von MySQL Cluster erzielt wird, können Cluster mithilfe der geografischen Replikation zwischen Standorten für eine Wiederherstellung im Notfall repliziert werden. Die geografische Replikation wird über die standardmäßige asynchrone MySQL Replikation implementiert, wobei ein wichtiger Unterschied die Unterstützung der Aktiv/Aktiv-Replikation ist. Wenn beispielsweise Ihre Anwendungen versuchen, gleichzeitig dieselbe Zeile in verschiedenen Clustern zu ändern, kann die geografische Replikationsfunktion von MySQL den Konflikt erkennen und beheben. Dies stellt sicher, dass alle Standorte mithilfe aller aktiven Ressourcen Lese- und Schreibanforderungen aktiv erfüllen können, für Datenkonsistenz unter der Clustern sorgen können und eine Wiederherstellung nach einem Notfall ermöglichen.

Aufbauend auf der zuvor behandelten Shared-Nothing-Architektur können Benutzer Wartungsaufgaben an MySQL Cluster durchführen, ohne die Datenbank herunterfahren zu müssen. Alle folgende Vorgänge erfolgen online, was bedeutet, das Abfrage-, Einfüge-, Aktualisierungs- und Löschtransaktionen durchgängig verarbeitet werden:

- Skalieren des Clusters durch Hinzufügen neuer Daten-, Anwendungs- und Verwaltungsknoten
- Aktualisieren des Schemas mit neuen Spalten, Tabellen und Indizes
- Datenknotenübergreifendes Neupartitionieren von Tabellen zum Ermöglichen einer besseren Datenverteilung
- Durchführen eines Upgrades für die zugrunde liegende Hardware und das Betriebssystem bzw. Einspielen von Patches
- Durchführen eines Upgrades für MySQL Cluster mit vollständigen Online-Upgrades zwischen Releases bzw. Einspielen von Patches
- Vollständige Online-Datenbanksicherung, synchronisiert zwischen allen Datenknoten.

Aufgrund der oben beschriebenen Funktionen können mit MySQL Cluster geplante Wartungsfenster und ungeplante Ausfälle weiter reduziert werden, um die für die wichtigsten Anwendungen erforderliche Verfügbarkeit von 99,999 % sicherzustellen.

Kontaktadresse:
Mario Beck
Oracle Deutschland
Komturstraße 18a
D-12099 Berlin

Telefon:	+49 (0) 30-74 70 96 - 879
E-Mail	mario.beck@oracle.com
Internet:	www.mysql.de