

Neu: MySQL Cluster 7.3 GA

Jürgen Giesel und Mario Beck, ORACLE Deutschland B.V. & Co. KG

Seit Juni 2013 steht MySQL Cluster 7.3 als Produktions-Release zur Verfügung. Zu den großen Neuheiten zählen eine „node.js“-NoSQL-Schnittstelle, eine grafische Benutzeroberfläche für den Auto-Installer sowie die Unterstützung von Fremdschlüsseln. Damit können Anwender mit den schnellen Innovationszyklen bei Web-, Cloud-, sozialen und mobilen Diensten weiter Schritt halten.

MySQL Cluster ist eine skalierbare, ACID-konforme, transaktionale Echtzeitdatenbank, die 99,999 Prozent Verfügbarkeit mit den niedrigen Gesamtbetriebskosten einer Open-Source-Lösung verbindet. Sie basiert auf einer verteilten Multi-Master-Architektur ohne singuläre Fehlerquellen und wird horizontal auf Standard-Hardware skaliert, um lese- und schreibintensive Arbeitslasten mit Auto-Sharding zu verarbeiten.

Ursprünglich als eingebettete Telekommunikations-Datenbank für netzinterne Anwendungen entworfen, die Carrier-Grade-Verfügbarkeit und Echtzeitleistung erfordern, wurde MySQL Cluster schnell um neue Funktionen erweitert, die es unter anderem für lokale oder in der Cloud bereitgestellte Web-, Mobil- und Unternehmens-Anwendungen einsetzbar machen. Die neuen Funktionen von MySQL Cluster betreffen folgende Bereiche:

- Automatische Installation von MySQL Cluster
- Fremdschlüssel

- JavaScript-Treiber für „Node.js“
- Leistungsverbesserungen
- MySQL 5.6: Erweiterungen der Replikation
- MySQL Cluster Manager: Zentralisierte Sicherung und Wiederherstellung

Automatische Installation von MySQL Cluster

Hohe Priorität hat in diesem Release die wesentlich einfachere und schnellere Bereitstellung eines Clusters, der genau auf die Anwendung und Umgebung des Nutzers zugeschnitten ist – dieser soll sich darauf konzentrieren, die Vorteile von MySQL Cluster in der Anwendung zu nutzen, statt darüber nachdenken zu müssen, wie die Datenbank installiert, konfiguriert und gestartet werden kann. Die gesamte Steuerung der automatischen Installation erfolgt über eine einfach zu bedienende Web-Anwendung und führt durch folgende Schritte:

- Eingabe der voraussichtlichen Art der Arbeitslast (beispielsweise „Echt-

zeitanwendung mit vielen Schreibvorgängen“) zusammen mit der Liste der Hosts, auf denen der Cluster ausgeführt werden soll.

- Das Installationsprogramm stellt automatisch Verbindungen mit den einzelnen Hosts her, um die verfügbaren Ressourcen zu ermitteln (Betriebssystem, Arbeitsspeicher und CPU-Kerne).
- Dann wird eine Topologie vorgeschlagen (also welche Knoten/Prozesse den Cluster bilden und auf welchen Hosts sie ausgeführt werden sollten). Der Benutzer kann diese Topologie akzeptieren oder verändern.
- Basierend auf allen bereitgestellten und ermittelten Informationen werden Konfigurations-Einstellungen vorgeschlagen – auch diese kann der Benutzer akzeptieren oder verändern.
- Zum Schluss kann das Installationsprogramm optional den Cluster bereitstellen und starten sowie den Status der Knoten anzeigen, während der Cluster in Betrieb genommen wird.

Fremdschlüssel

MySQL Cluster 7.3 unterstützt nativ die Nutzung von Fremdschlüsseln. Das Entwurfsziel war, die Implementierung von Fremdschlüsseln in InnoDB möglichst genau nachzustellen. So können Benutzer leichter zu MySQL Cluster übergehen, wenn dies sinnvoll ist, und ihre vorhandenen Kenntnisse der MySQL-Entwicklung bei der Erstellung neuer Anwendungen weiter nutzen. Auf einige Ausnahmen sei hier hingewiesen:

- InnoDB unterstützt keine „No-Action“-Einschränkungen; MySQL Cluster unterstützt diese.
- InnoDB kann das Erzwingen von Fremdschlüssel-Einschränkungen mit dem Parameter „FOREIGN_KEY_CHECKS“ aussetzen; MySQL Cluster ignoriert diesen Parameter zurzeit.
- Man kann keine Fremdschlüssel zwischen zwei Tabellen einrichten, wenn die eine mit MySQL Cluster gespeichert ist und die andere mit InnoDB.
- Primärschlüssel für MySQL-Cluster-Tabellen lassen sich nicht über Fremdschlüssel-Einschränkungen ändern.
- In MySQL Cluster sind Fremdschlüssel-Einschränkungen für partitionierte Tabellen zulässig.

JavaScript-Treiber für „Node.js“

„Node.js“ ist eine Plattform, mit der schnelle, skalierbare Netzwerk-Anwendungen (normalerweise Web-Anwendungen) mit JavaScript entwickelt werden können. „Node.js“ ist dafür entworfen, dass ein einzelner Thread Millionen von Client-Verbindungen in Echtzeit bedienen kann. Dies wird durch eine asynchrone, ereignisgesteuerte Architektur erreicht. Es entspricht der Funktionalität von MySQL Cluster, sodass die beiden Lösungen hervorragend zusammenpassen.

Der NoSQL-Treiber von MySQL Cluster für „Node.js“ wurde als Modul für die V8-Engine implementiert. Er bietet eine native, asynchrone JavaScript-Schnittstelle für „Node.js“, über die Abfragen direkt an MySQL Cluster gesendet und Ergebnismen-

gen empfangen werden können, ohne Transformationen in SQL. Zusätzlich kann man auch festlegen, dass der Treiber SQL verwendet, sodass dasselbe API für InnoDB-Tabellen verwendet werden kann.

Mit dem JavaScript-Treiber von MySQL Cluster für „Node.js“ können Architekten JavaScript vom Client auf dem Server wiederverwenden, bis hin zu einer verteilten, fehlertoleranten, transaktionalen Datenbank. Dies unterstützt hoch skalierbare Echtzeitsysteme:

- Verarbeitung von Streaming-Daten aus digitalen Werbungs- und Benutzerverfolgungs-Systemen
- Spiele und soziale Netzwerkseiten – als Grundlage der Back-End-Infrastruktur von Diensten für mobile Geräte

JavaScript ist mit „Node.js“ Teil eines wachsenden Portfolios von NoSQL-APIs für MySQL-Cluster, zu dem bereits „Memcached“, „Java“, „JPA“ und „HTTP/REST“ gehören. Natürlich können Entwickler weiterhin SQL nutzen, um komplexe Abfragen auszuführen und das umfangreiche Umfeld von Konnektoren, Frameworks, Hilfswerkzeugen und Kompetenzen zu nutzen.

Leistungsverbesserungen

MySQL Cluster 7.3 baut auf den enormen Verbesserungen der Leistung und Skalierbarkeit in MySQL Cluster 7.2 auf. In diesem Release wurde zwei Bereichen besondere Beachtung geschenkt. Der erste ist die Verbesserung des Durchsatzes jeder Anwendungs-Verbindung zur Datenbank, mit dem noch höhere Skalierbarkeit ermöglicht wird. Der zweite Bereich ist die Verarbeitung von SQL-Abfragen auf dem MySQL-Server, mit der Abfragen beschleunigt und die Last der Cluster-Knoten verringert werden können, was wiederum den Durchsatz erhöht.

Die Leistungsfähigkeit von MySQL Cluster wird besonders deutlich, wenn so viele parallele Vorgänge angeboten werden wie möglich. Um dies zu erreichen, sollte Parallelität auf jeder Ebene konfiguriert werden: Es sollten mehrere Anwendungs-Threads Arbeit

an den MySQL-Server (oder das sonstige API) senden und es sollte mehrere MySQL-Server sowie mehrere Verbindungen zwischen dem MySQL-Server (oder Knoten des sonstigen API) und den Datenknoten geben.

Jede Verbindung zu den Datenknoten verbraucht eine der 256 verfügbaren Knoten-IDs. In einigen Szenarien kann dies eine Grenze für die Skalierbarkeit des Clusters darstellen. In MySQL Cluster 7.3 ist der Durchsatz über jede dieser Verbindungen wesentlich höher. Dies bedeutet, dass weniger Verbindungen (und somit weniger Knoten-IDs) erforderlich sind, um dieselbe Arbeitslast zu bewältigen. Dies wiederum bedeutet, dass dem Cluster mehr API-Knoten und Datenknoten hinzugefügt werden können, um die Kapazität und Leistung noch weiter zu skalieren. Vergleichstests haben gezeigt, dass der Durchsatz pro Verbindung bis zu sieben Mal höher ist.

MySQL 5.6: Erweiterungen der Replikation

Die native Integration mit MySQL 5.6 ermöglicht es, InnoDB- und MySQL-Cluster-Speicher-Engines innerhalb einer MySQL-5.6-basierten Anwendung zu kombinieren – und die in MySQL 5.6 enthaltenen Neuheiten bezüglich der MySQL-Replikation zu nutzen. MySQL Cluster verwendet die MySQL-Replikation für die Implementierung der geografischen Replikation. Nutzer von MySQL Cluster 7.3 profitieren nun von:

- Prüfsummen für Replikations-Ereignisse
- Verringerter Größe des Binär-Logs
- Der Möglichkeit, Replikationen zeitlich verzögert vorzunehmen

MySQL Cluster Manager: Zentralisierte Sicherung und Wiederherstellung

MySQL Cluster unterstützt Online-Sicherungen (und die anschließende Wiederherstellung dieser Daten). MySQL Cluster Manager 1.2 vereinfacht den Prozess. Die Datenbank kann jetzt mit einem einzelnen Befehl gesichert werden, wobei wiederum die Daten auf jedem Datenknoten im Cluster gesichert werden: „mcm> backup clus-

```
mcm> list backups mycluster;
```

BackupId	NodeId	Host	Timestamp
1	1	grün	2012-07-31T06:41:36Z
1	2	braun	2012-07-31T06:41:36Z
1	3	grün	2012-07-31T06:41:36Z
1	4	braun	2012-07-31T06:41:36Z
1	5	lila	2012-07-31T06:41:36Z
1	6	rot	2012-07-31T06:41:36Z
1	7	lila	2012-07-31T06:41:36Z
1	8	rot	2012-07-31T06:41:36Z

Listing 1

ter mycluster;“. Der Befehl „list.“ kann identifizieren, welche Sicherungen im Cluster verfügbar sind (siehe Listing 1).

Man kann dann auswählen, welche dieser Sicherungen man wiederherstellen möchte, indem man beim Aufrufen des Wiederherstellungsbefehls die zugeordnete Sicherungs-ID angibt: „mcm> restore cluster -I 1 mycluster;“. Falls man die vorhandenen Inhalte der Datenbank entfernen muss, bevor man die Wiederherstellung durchführt, kommt die in MCM 1.2 eingeführte Option „—initial“ für den Befehl „start cluster“ zum Einsatz, um die Daten aus allen MySQL-Cluster-Tabellen zu entfernen.

Jürgen Giesel

Juergen.Giesel@oracle.com



Mario Beck

Mario.Beck@oracle.com



Oracle macht den Enterprise Manager 12c fit für die Multitenant-Option

Mit dem Oracle Enterprise Manager Cloud Control 12c Release 3 Plugin Update 1 (12.1.0.3) öffnet Oracle den Weg für das volle „Lifecycle Management“ von Database-as-a-Service (DBaaS) mit der neuen Multitenant-Option der neuen Datenbank 12c.

Hinsichtlich der Datenbank-Bereitstellung bietet das Release umfassende Auswahlmöglichkeiten in Sachen „Provisioning“, „Monitoring“, „Verwaltung“, „Backup“ und „Wiederherstellung“ von Datenbanken in der Cloud.

So sind gesamte Datenbanken, Instanz-Klone, Schemata oder sofort einsatzbereite DBaaS schnell erstellt. Der „DBaaS rapid start Kit“ enthält zudem eine Lösung, die es ermöglicht, Exadata als „out of the Box“-Plattform einzurichten. Sie erlaubt es Administratoren weiterhin, ihr DBaaS-Setup mit Skripten und Third-Party-Lösungen zu integrieren.

Für die Nutzung von „Pluggable Databases“ ist eine Reporting-Funktionalität vorgesehen, die unter anderem eine Messung der Workloads sowie eine Kosten-Analyse ermöglicht. Darüber hinaus können Administratoren auch die Leistungsfähigkeit und den Lebenszyklus von Container Databases administrieren und monitoren.

Ein Service-Katalog, in dem genehmigte Konfigurationen von „Pluggable Da-

tabases“ festgehalten werden, soll Unternehmen bei der Standardisierung helfen. Zudem sorgt ein rollen- und richtlinienbasiertes Management für die Einhaltung der Compliance und Governance-Anforderungen.

Das neue Release stellt auch „Testing as a Service“ zur Verfügung. Somit können zusätzlich zu den Lasttests auch Funktions- und Regressionstests über den Enterprise Manager in der Private or Public Cloud durchgeführt werden. Auch beim Patching bringt das Release Neues mit sich: Mit einem Out-of-Place-Patching lassen sich Datenbanken in einer RAC-Umgebung ohne Ausfallzeiten patchen.

Zusätzlich zu der bisherigen Unterstützung von Oracle ZFS Storage Appliances und Netapp sind nun auch generische ZFS-Dateisysteme nutzbar. So funktioniert die Lösung sowohl auf NAS- als auch auf SAN-Storage gleichmäßig. Darüber hinaus ist das Klonen von Datenbanken jeder Größe unter Verwendung von Snap Clone innerhalb von Minuten erledigt.

Auch für das Daten Lifecycle Management stellt das neue Release einen Advisor zur Verfügung, der „Heat Map“-Informationen über Tablespaces und Segmenten von häufig abgefragten oder geänderten Daten visualisiert und damit eine optimalere Nutzung von Ressourcen ermöglicht.