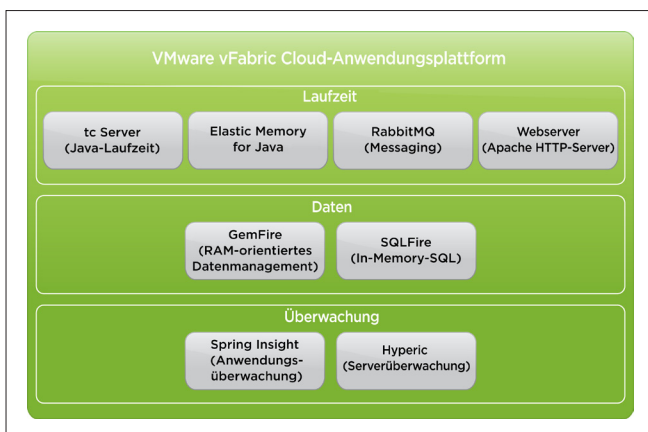


VMware vFabric GemFire

Elastisches In-Memory-Datenmanagement

MODERNES DATENMANAGEMENT

- Extrem schneller, hochverfügbarer und skalierbarer Datenzugriff für moderne Anwendungen – Ausführungsgeschwindigkeit um 4- bis 40-mal schneller
- Parallele Datenverarbeitung für extrem hohen Durchsatz
- Konsequenz auf minimale Latenzen ausgelegt
- Datenreplikation über mehrere Knoten und Cluster für hohe Verfügbarkeit
- Datenpartitionierung über Regionen zur horizontalen Skalierung
- Zuverlässige Ereignisbenachrichtigung: Anwendung ist bei Änderungen an den Daten automatisch auf dem neuesten Stand
- Bereitstellung von Ergebnissen für Dauerabfragen mit verlässlich niedriger Latenz
- Festplattenspeicherung nach „Shared-Nothing“-Modell: Plattenausfall auf einem Knoten führt nicht zu Datenverlust
- Gemeinsame Nutzung von Daten zwischen Anwendungen auf Basis von Java, C++ und C#
- WAN-Skalierung bei gleichbleibender Performance, Zuverlässigkeit und Datenkonsistenz



VMware vFabric ist die beste Plattform zur Ausführung von Java-Anwendungen auf Spring-Basis und eignet sich zudem hervorragend für virtuelle Infrastrukturen.

Übersicht

VMware vFabric™ GemFire® ermöglicht schnellen, sicheren, zuverlässigen und skalierbaren Datenzugriff für moderne Enterprise- und Cloud-Anwendungen. Als Kernkomponente der Cloud-Anwendungsplattform VMware vFabric ermöglicht GemFire die Bereitstellung datenintensiver Anwendungen bei höchster Leistung und ständiger Verfügbarkeit – unabhängig von der Anwenderzahl.

Moderne Anwendungen sind immer stärker weborientiert und müssen dynamischer mit größeren Datenmengen umgehen. Demzufolge stellen sie auch neue Anforderungen an das Datenmanagement. Hohe Laufzeiten für den Datenabruf aus einer Datenbank waren bei Webanwendungen der ersten Generation vielleicht noch vertretbar. Für neuere Anwendungen ist jedoch ein anderer Ansatz erforderlich, der nicht nur an die veränderten Rahmenbedingungen angepasst ist, sondern sich auch für die Bereitstellung in virtuellen Infrastrukturen eignet.

GemFire verbindet die Datenmanagement-Funktionalität einer Datenbank mit einem Leistungsniveau, wie es nur mit In-Memory-Technologie möglich ist. Damit eignet es sich hervorragend für moderne Anwendungen, die einen Echtzeit-Datenzugriff erfordern oder besonders komplexe Anforderungen bezüglich des Datenmanagement stellen.

Die wichtigsten Highlights

In-Memory-Datenmanagement mit GemFire vermeidet hohe Transaktionskosten für Datenbankzugriffe (CPU-Zyklen, Netzwerkverkehr, Datenbanklatenzen usw.)

Entwicklerproduktivität

- Überarbeitete APIs erleichtern die Entwicklung, Codebeispiele für den schnellen Einstieg werden mitgeliefert
- Deklarative Konfiguration der GemFire-Dateninfrastruktur aus Spring – dadurch sicherer, zuverlässiger und skalierbarer Datenzugriff aus Enterprise-Anwendungen bei minimalem Aufwand
- Integrierte Ausnahmefehlerbehandlung gibt Ausnahmen aus GemFire an Spring weiter; dadurch wird ein einheitlicher, transparenter Umgang mit Fehlern möglich
- Integration mit den Transaktionsmanagement-Funktionen des beliebten Spring-Frameworks ermöglicht die schnelle Entwicklung sicherer, zuverlässiger, skalierbarer transaktionsorientierter Enterprise-Anwendungen und erleichtert den Support
- Hibernate-Cachemodul für schnelles, skalierbares, verteiltes Level-2-Caching bei Anwendungen, die objektrelationales Mapping mit Hibernate nutzen

Persistenz wie bei einer Datenbank

- Persistenz auf parallelen Datenträgern nach dem „Shared-Nothing“-Prinzip kombiniert hohe Performance auch in stark skalierten Umgebungen mit Hochverfügbarkeit im gesamten Cluster und Stabilität im Fehlerfall
- Anbindung an Back-End-Datenquellen synchron (Read-Through, Write-Through) oder asynchron (Write-Behind)
- Unterstützung für Dauerabfragen – Anwendungen werden automatisch mit Daten aus dem RAM aktualisiert, dadurch keine Laufzeiten für Datenbankzugriffe

Hohe Effizienz auch im Cloud-Maßstab

- Das PDX-Format von GemFire (Portable Data eXchange) ist ein sprachübergreifendes Datenformat, das helfen kann, die Kosten beim Verteilen und Serialisieren/Deserialisieren von Datenobjekten zu reduzieren.
- Dynamische Datenpartitionierung im System sorgt für gleichmäßige Lastverteilung und hohe Skalierbarkeit
- Transaktionen nach dem Co-Location-Prinzip, zwei- bis dreimal höhere Performance für Thin Clients
- WAN-Unterstützung zur übergreifenden Skalierung für Remote-Standorte

- Mehrere Verwaltungsoptionen: Befehlszeilen-Utility, JMX-Agent (Java Management Extensions) und GFMon (Monitoring-Tool mit grafischer Benutzeroberfläche)
- HTTP-Sitzungsmanagementmodul übernimmt eigenständig die Verwaltung des HTTP-Sitzungsstatus für tc Server oder Tomcat-Server. Das vorkonfigurierte HTTP-Sitzungsmanagement von GemFire wird automatisch mit tc Server gestartet und ermöglicht eine hohe Performance und Skalierbarkeit auch bei Webanwendungen mit hoher Anwenderlast oder sehr umfangreichen Sitzungen.

Bereitstellungsoptionen

- GemFire wird in einem Java Runtime Environment (JRE) im 32- bzw. 64-Bit-Modus unter Windows, Linux und Solaris ausgeführt.
- Client werden auf der Basis von C++, C# .Net und Java unterstützt.
- GemFire unterstützt neben Peer-to-Peer- und Client/Server-Architekturen auch einen Betrieb über mehrere Standorte im WAN.

