Zwischenvortrag Skript

Ja danke – also jetzt gehen wir weiter mit den Eigenschaften

Wir betrachten nämlich die 4 hier: Clusterfähigkeit, Integrationskomplexität, Skalierbarkeit und Replizierbarkeit

Clusterfähigkeit geht ja darum, ob eine Datenbank fähig ist, sich in mehrere Server oder Instanzen zusammen verbinden und als eine einzige Datenbank herzustellen

Integrationskomplexität trifft dazu, wie aufwendig ist es, die Datenbank zu installieren und in eine Anwendung zu integrieren

Mit Skalierbarkeit versteht man, wie fähig ist die Datenbank, mit variabler Anzahl Zugriffen und großer Datenmengen umgehen kann – also ob die Datenbank mit abweichende Workloads (Read, Write, Update usw.) schnell oder gut adaptieren kann

Zuletzt bedeutet Replizierbarkeit hier, die Verteilung und Erstellung von Kopien derselben Daten - was natürlich sehr wichtig für die Verfügbarkeit den Daten, Latenz und Skalierbarkeit die Datenbank ist

Lassen wir mal dann die Eigenschaften von unseren gewählten Datenbanken schauen:

Wir fangen mit Bibliotheken an: nach unserer Definition sind Bibliotheken inhärent nicht Clusterfähig und nicht Replizierbar, da während ihrer Anwendung nur eine Instanz existiert.

Und die beide Datenbanken hier, MapDB und MVStore, haben die gleiche Integrationskomplexität: die dürfen direkt eingebunden sein, z.B. als eine Maven Dependency, und ihre Skalierbarkeit basieren auf den Java Collections oder Threads, da die ja in Java geschrieben wurden

Danach gehen wir zu dem skalierbaren Prozesse: die 3 Datenbanken hier sind alle clusterfähig

Die Erste: Voldemort hat Auto-clustering feature mit consistent hasing und multi-version concurrency control

Bei Aerospike ist es auch automatisiert, aber mit Master-Slave Zuweisung zwischen die Nodes, mit synchronisierter Replikation und Rack-Awareness

Mit Riak erfolgt Clustering nicht automatisch, aber multi-cluster ist immer noch möglich und man kann mehrere Nodes oder sogar mehrerer Clusters auf dem gleichen Computer hosten

Die 3 Datenbanken haben also unterschiedliche Integrationskomplexität

Voldemort kann man mit ein bat Programm oder über Kommandozeilen gestartet werden, und Aerospike ganz bequem mit Docker, bei Riak braucht man aber bei Windows zusätzliche Virtualisierung, um die Datenbank überhaupt zu starten

Skalierbarkeit ist bei Voldemort/Aerospike und Riak auch ein bisschen anders

Bei Voldemort und Aerospike sind Daten auf den Server/Nodes repliziert, und die Workloads dürfen auf Server oder Cores geteilt werden – also horizontal skalieren. Riak hat aber ihr eigenes System, der Riak Ring: dieser Ring ist ein 160-bit binäre Hash, was dann von allen Vnodes (virtuale Nodes) verteilt werden; Workloads werden an die Vnodes gleich verteilt, in dieser Ring, was dann von ihren zugehörige „reale“ Nodes behandelt werden

Zu unsere letzte Kategorie kommen wir auf der Mischform: RocksDB + Rocksplicator

RocksDB, was auf LevelDB basiert, ist nativ eine Bibliothek – mit zusätzliche Codes, Rocksplicator, ist aber Clustering möglich mit Asynchron, Semi-synchron oder synchron replizierbar; Integrationskomplexität bleibt niedrig, entweder mit Docker oder direkt einbinden (wieder z.B. Maven Dependency) und dank Rocksplicator kann man es auf mehreren Cores laufen lassen, was die Workloads verteilt werden können

Damit bin ich fertig mit den Eigenschaften der Datenbanken, und ich gebe jetzt weiter an meinem Kollege, Alexander