
Wie moderne Datenbanken herausfordernde Anwendungen meistern



Vorwort

Das Bild von Datenbanken als Speicher, die Unternehmensdaten zentral im eigenen Rechenzentrum aufbewahren, hat schon lange ausgedient. Eine zeitgemäße Datenbank muss an vielen Standorten gleichzeitig funktionieren – auch in der Cloud und am Edge, wo es auf eine zeitkritische Verarbeitung ankommt. Diese Verteilung darf die Komplexität des gesamten Systems nicht unnötig steigern. Es muss also zum Beispiel Mechanismen geben, die die Vorverarbeitung und Synchronisierung der Daten intelligent, sicher und transparent erledigen. Um die IT-Fachabteilungen zu entlasten, soll dann auch noch die Übertragung möglichst vieler Aufgaben und Verantwortungsbereiche an einen erfahrenen Service-Dienstleister möglich sein. Klingt fantastisch, ist aber realistisch: Denn ein solches System ist heute bereits erhältlich!

Die Herausforderungen heutiger Anwendungsszenarien lassen sich gut am Beispiel der Kreuzfahrtbranche illustrieren. Der Markt dieser ständig beliebter werdenden Reiseform wird von großen Konzernen beherrscht, deren schwimmende Hotels jedes Jahr Hunderttausende Passagiere befördern. Das Publikum ist international und erreicht die Starthäfen in der Regel mit dem Flugzeug. Die logische Konsequenz: Auch die Buchung einer Kreuzfahrtreise muss von überall aus in der Welt möglich sein.

Ein fiktives Kreuzfahrtunternehmen, nennen wir es einfach „Cruise Ship World“, möchte sich nun von seinen Mitbewerbern abheben und eine komfortable App entwickeln, die potenzielle Passagiere vor und während ihrer Reise optimal unterstützt. Dazu sollen Kunden beispielsweise weltweit ihre Reisen in Echtzeit buchen können. Da die Zahl der Kabinen auf den Schiffen des Konzerns begrenzt ist, müssen die Buchungen zu diesem Zweck weltweit synchronisiert werden, damit es nicht zu Doppelbelegungen kommt. Jeder Interessent sollte dabei sofort erkennen können, welche Kabinen auf welchem Schiff zu welchem Datum noch frei sind. Gleichzeitig sollte die App aber auch offline funktionieren, denn der satellitengestützte Internet-Zugang auf Kreuzfahrtschiffen ist in der Regel nur gegen Aufpreis erhältlich. Nicht jeder Reisende bucht dieses Angebot dazu. Jeder Gast sollte jedoch dennoch die Möglichkeit haben, zumindest seine persönlichen Daten und das Wunschdatum für die nächste Reise mit diesem oder einem anderen Schiff einzugeben. Die Synchronisation mit dem weltweiten Buchungssystem des Anbieters erfolgt dann, sobald das Schiff wieder in den Bereich der Mobilfunknetze gelangt.



Schnell erkennt man bei „Cruise Ship World“, dass sich diese Aufgaben mit den herkömmlichen Datenbank-Architekturen nicht bewältigen lassen. Ursprünglich hatte das Entwicklerteam ein zentrales Datenbanksystem am Hauptsitz des Unternehmens vorgesehen, das sämtliche Buchungen aus allen Teilen der Welt bearbeiten sollte. Doch die hohen Latenzzeiten machen das unmöglich. Hinzu kommt, dass die daraus resultierenden, oft sehr langsamen Reaktionszeiten der App ein positives Nutzererlebnis verhindern: Das System wirkt träge und ist anfällig für Fehler.

Als Lösung erarbeitet das IT-Team ein Konzept, das vor allem auf die Public-Cloud setzt: Passend zu den Ländern aus denen die allermeisten Kunden kommen, sucht es sich für das Deployment der Datenbanken diejenigen Verfügbarkeitszonen der Cloud-Anbieter aus, die diese Länder abdecken. Denn nur, indem man die Daten ganz nah beim Kunden hält, ist es der IT-Abteilung möglich, Response-Zeiten im Submillisekunden-Bereich zu realisieren. Durch kurze Reaktionszeiten lässt sich dann für die Kunden ein optimales Ergebnis bei der Bedienung der App erreichen.

Neben der Nutzung der Public-Cloud-Dienste von Hyperscalern wie AWS, Google oder Microsoft Azure sieht das Team zusätzlich die Einbeziehung einer Private Cloud oder des eigenen Rechenzentrums für die Datenbanken vor. Bei diesem hybriden Ansatz könnte „Cruise Ship World“ dort diejenigen Informa-

tionen vorhalten, die nicht direkt von der App genutzt, sondern durch Weiterverarbeitung aus ihnen generiert werden. Wichtige Daten für Anwendungen wie beispielsweise das Reporting, die Umsatzplanung oder die Verwaltung von Schiffen und Personal verbleiben so in einem separierten Bereich nahe am Unternehmen.

Database-Replikation zum Hyperscaler

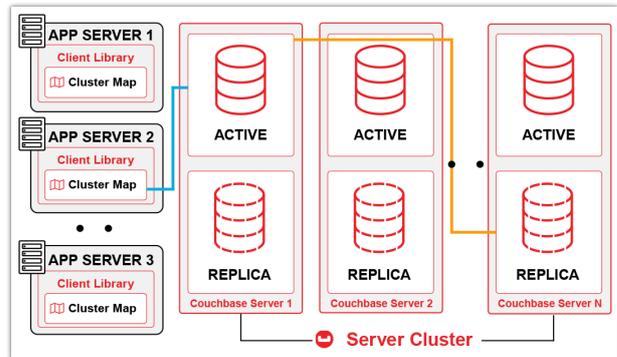
Um die Belastung durch die laufenden Workloads zu verringern oder im Notfall ständig auf eine aktuelle Eins-zu-eins-Kopie einer Datenbank zugreifen zu können, werden Datenbanken häufig von den Rechenzentren on premise zu den Cloud-Diensten der großen Hyperscaler wie AWS, Google und Azure repliziert. Das kann in festgelegten Abständen geschehen. Zudem ist sowohl eine Replikation der Daten zum gleichen Datenbanksystem in der Cloud möglich wie auch zum System eines anderen Herstellers. Im zweiten Fall spielen häufig finanzielle Überlegungen in puncto Lizenzkosten eine Rolle.

Doch wie lässt sich das Problem der Synchronisierung lösen? Da die Kunden in den verschiedenen Regionen auf die gleichen Daten zugreifen, in diesem Fall also auf die Buchungsinformationen der einzelnen Kreuzfahrtschiffe, müssen die Datenbanken ständig synchronisiert und miteinander abgeglichen werden.

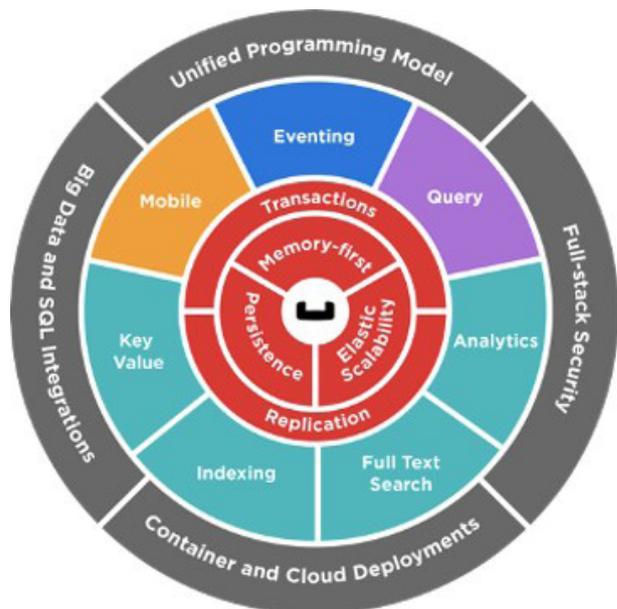
Wie Couchbase als übergreifende Lösung funktioniert

Die Umsetzung einer Lösung, wie sie im Beispiel skizziert wird, stellt Unternehmen vor große Herausforderungen. Denn die Anforderungen sind hoch: Im Fokus steht ein Datenbanksystem, das auf mehreren unterschiedlichen Plattformen funktioniert. Es soll einerseits auf leichtgewichtigen Endgeräten wie Smartphones und Tablets laufen, und zwar sowohl im Online- wie auch im Offline-Modus. Gleichzeitig soll es aber auch als Serversystem dienen können und dabei Private- ebenso wie Public-Clouds unterstützen. Außerdem muss es auch eine Database-Replikation beherrschen, bei der die verschiedenen Instanzen eines Distributed-Database-Management-Systems ständig ihre Daten untereinander austauschen. Last, but not least sollten sich der Aufwand für die Entwicklung und die damit verbundenen Kosten in einem überschaubaren Rahmen halten.

Eines der wenigen Datenbanksysteme, das alle diese Anforderungen erfüllt, ist Couchbase von dem gleichnamigen US-amerikanischen Unternehmen. Das System liegt in drei Varianten vor: als Couchbase Server, Mobile und Couchbase Capella. Letzteres ist ein vollständig verwalteter Database-as-a-Service (DBaaS), der speziell für den Betrieb in einer Public-Cloud entwickelt wurde und beispielsweise auf AWS oder Google Cloud läuft. Hinzu kommt noch Couchbase Lite, eine Version für Embedded-Systeme. Sie benötigt lediglich drei MB Speicherplatz und lässt sich beispielsweise sogar schon auf einem Raspberry Pi betreiben. Ebenso wie die Mobile-Version können Unternehmen diese Variante sehr nah am Edge platzieren, um die Antwortzeiten so kurz wie möglich zu halten.



Couchbase eignet sich gleichermaßen für den Einsatz in der Cloud, auf Mobilgeräten und am Edge eines Netzwerks.



Generell handelt es sich bei Couchbase um eine NoSQL-Datenbank, also ein Not-only-SQL-System. Diese Datenbanken brechen mit den starren Strukturen relationaler Datenbanken und lassen sich deshalb flexibel einsetzen. Sie skalieren sowohl horizontal als auch vertikal und eignen sich beispielsweise auch für Big-Data-Anwendungen. Couchbase zwingt den Benutzer jedoch nicht dazu, eine irreversible Entscheidung zwischen einer relationalen und



einer NoSQL-Datenbank zu treffen. Denn das System umfasst auch Features, welche die Schemata und Tabellen relationaler Databases nachbilden.

Für einen Einsatzzweck, wie wir ihn oben skizziert haben, empfiehlt sich Couchbase auf jeden Fall, weil alle Varianten auf der gleichen Code-Basis aufbauen. App-Entwickler müssen also nicht mehrere unterschiedliche Versionen ihres Codes schreiben, sondern können ihn mit kleinen Modifikationen leicht an die jeweilige Variante anpassen. Couchbase macht ihnen das Leben noch leichter, da alle Zugriffe auf die Daten über die gleiche API (Application Programming Interface) laufen. Zudem bietet das System in allen Versionen SQL-Queries an, Entwickler müssen sich also nicht in eine proprietäre Abfragesprache einarbeiten, die sie noch nicht kennen. In der Praxis führen diese Merkmale dazu, dass sich Couchbase-Systeme in verhältnismäßig kurzer Zeit aufbauen und in den Live-Betrieb bringen lassen.

Die genannten Eigenschaften von Couchbase haben zudem noch den Vorteil, dass sich Datenbanken sehr einfach und schnell synchronisieren und replizieren lassen. Für ein verteiltes Datenbanksystem, wie es weiter oben vorgestellt wurde, ist das ein unschätzbare Vorteil.

SQL++ to easily query and access JSON

```
results = conn.query(
  "SELECT * FROM users WHERE addr.work.state=$s",
  s="California")
```

High-performance NoSQL database ops

```
conn.mutate_in('user-1234',
  op.remove('addr.billing'),
  op.replace('email', 'doug11@hotmail.com'))
```

Die verschiedenen Varianten von Couchbase für lokale Server, die Cloud und Mobilgeräte lassen sich dank einheitlicher APIs einfach synchronisieren.

Darüber hinaus weist Couchbase alle Merkmale eines vollwertigen Datenbank-Management-Systems auf. So unterstützt es in allen Versionen eine Volltextsuche für das schnelle Auffinden eines Datenbank-Assets. Hinzu kommt die Fähigkeit zum Aufbau eines Key-Value-Stores, also einer Schlüssel-Werte-Datenbank, die sich durch eine hohe Performance und flexible Skalierbarkeit auszeichnet. Auch Predictive Queries und ein Eventing-Service sind möglich.

Was den Eventing-Service von Couchbase auszeichnet

Beim Eventing-Service von Couchbase handelt es sich um ein Framework, das Datenänderungen in Echtzeit verarbeitet. Er ist in die Couchbase-Data-Plattform integriert und beobachtet sogenannte „Mutationen“ – mit diesem Begriff bezeichnet Couchbase Erstellungs-, Aktualisierungs-, Verfalls- oder Löschvorgänge. Wenn beispielsweise ein Kunde bei einer E-Commerce-Anwendung eine Bestellung aufgibt, lässt sich nachverfolgen, wie die Software einen neuen Datensatz für die Transaktion anlegt, dem Konto des Kunden eine neue Kauftransaktion hinzufügt und den Lagerbestand des Shops mit der neuen Bestellung aktualisiert. Jede dieser Änderungen verzeichnet Couchbase als ein Ereignis (englischer Ausdruck: event). Da Couchbase den Eventing-Service beinhaltet, benötigt der Anwender keine Tools von Drittanbietern für die Verwaltung und die Reaktion auf Datenmutationen.

Wie Couchbase Capella den Fachkräftemangel bekämpft

Doch nicht nur technische Herausforderungen bremsen viele Datenbankprojekte aus, es gibt noch andere Hürden. In immer mehr Regionen der Welt beklagt die Industrie zum Beispiel einen Fachkräftemangel. Es fehlen qualifizierte Ingenieure, Informatiker, Entwickler und andere Spezialisten, die das Wachstum der Firmen unterstützen und neue Maßnahmen umsetzen können.

Ein Projekt wie die Entwicklung einer App für die weltumspannende Buchung von Kreuzfahrten bindet jedoch nicht nur in der Entstehungsphase die Manpower mehrerer Fachkräfte. Auch für den Betrieb des Systems muss das Unternehmen in sein Kalkül einbeziehen, dass in seinen Rechenzentren ein zusätzlicher Bedarf an Administratoren und Entwicklern entsteht. Back-ups und Replikationen müssen eingerichtet, Patches und Updates eingespielt und neue Versionen weltweit ausgerollt werden. Die Systeme benötigen beim Betrieb von geschäftskritischen Anwendungen eine Überwachung rund um die Uhr. Das gilt auch für den Support, denn die User möchten die App zu jeder Tages- und Nachtzeit nutzen. Deshalb müssen die IT-Mitarbeiter für das Datenbanksystem in der Lage sein, ebenfalls rund um die Uhr Support zu leisten.

Auch die stärkere Belastung der Security-Spezialisten in Betrieben spielt eine Rolle, zudem müssen bei der Verarbeitung der Daten die rechtlichen Rahmenbedingungen in den einzelnen Ländern beachtet und von der IT umgesetzt werden. Bei Projekten in der genannten Größenordnung bedeutet das unter dem Strich einen nicht zu unterschätzenden personellen Zusatzaufwand, der sich auch in den Kosten niederschlägt. Firmen müssen aber noch eine Reihe weiterer Aspekte in Betracht ziehen: So entstehen etwa durch den Einsatz eines Datenbanksystems on premise zusätzliche Infrastruktur- und Operationskosten für den Betrieb im Rechenzentrum.

Gegensteuern lässt sich mit Couchbase Capella. Wie zuvor bereits erwähnt, handelt es sich dabei um Database-as-a-Service (DBaaS), die Cloud-Variante von Couchbase. Damit profitiert das Unternehmen in vielerlei Hinsicht:

- Ganz grundsätzlich liegt die Verantwortung für das Datenbanksystem nun nicht mehr bei der Firma, sondern wechselt zum Datenbankbetreiber. Er kümmert sich um die Sicherheit und Stabilität des Systems. Das beginnt bereits beim Aufsetzen der Lösung, die perfekt nach den Angaben von Couchbase erfolgt. Mit Couchbase Capella wird anschließend das Einspielen von Updates und Patches ebenso automatisiert wie Upgrades auf neue Versionen.
- Beim Kunden entfällt die Investition in die Hardware – auch dieser Bereich liegt nun in den Händen des Datenbankbetreibers. Fragen zur Kompatibilität sowie zur benötigten Performance sind jetzt von ihm zu klären. In diesem Zusammenhang sollten Unternehmen auch beachten, dass für den Betrieb umfangreicher Datenbanksysteme ganze Servergruppen erforderlich sind, deren Kosten bei vielen Firmen außerhalb des Investitionsbudgets liegen. Auch die Ausgaben für professionelle Storage-Systeme, die höchste Anforderungen an die Datensicherheit erfüllen, liegen in vielen Fällen außerhalb der Reichweite von Firmen.
- Auch bei der Security spielen die Kosten eine wichtige Rolle, zudem muss das entsprechende Know-how vorhanden sein. Professionelle Rechenzentren, wie sie die Cloud-Anbieter betreiben, sind gegen physische Zugriffe mit modernster Überwachungstechnik und Wachpersonal abgesichert. Gleichzeitig sorgen Spezialisten für die Sicherheit der Systeme und Daten, damit Angriffe aus dem Netz ins Leere laufen.

Fazit

Trotzdem stellt auch eine DBaaS-Lösung wie Couchbase Capella kein Rundum-sorglos-Paket dar. Die Bedienung und Anpassung an das jeweilige Projekt verbleibt ebenso beim Anwender wie etwa die Überwachung der Leistungsdaten. Kunden können jedoch auf Wunsch problemlos zusätzliche Ressourcen buchen, während die Bereitstellung zusätzlich benötigter Hardware on premise oftmals etwas länger dauert.

Der Einsatz von Couchbase Capella entlastet sowohl die Entwickler wie auch die Administratoren, indem ihnen das System zahlreiche Routinejobs abnimmt. Stattdessen können sie sich auf wichtigere Aufgaben wie etwa die Entwicklung neuer Produkte und Features oder die Optimierung der Performance konzentrieren. Die wertvollen und teuren Fachkräfte werden dadurch erheblich effektiver eingesetzt.

Couchbase ist einmalig, denn der Kunde bekommt nicht nur eine leistungsfähige Datenbank, die sich nahezu beliebig skalieren lässt. Couchbase bringt bereits von Haus aus viele Komponenten und Features mit, die ansonsten über Tools von Drittanbietern hinzugefügt und aufwendig konfiguriert werden müssten. Dazu zählt der Key-Value-Store genauso wie die Volltextsuche, der Eventing-Service und die mobile Komponente. Die Software enthält bereits alles, muss nicht zusätzlich integriert werden und ist über die verbreitete Abfragesprache SQL einfach zugänglich. In Kombination mit der hohen Flexibilität von Couchbase und der umfassenden Cloud-Kompatibilität ergibt das eines der führenden Datenbanksysteme auf dem Markt.

Das US-Unternehmen Couchbase wurde 2011 gegründet. Die gleichnamige Backend-Plattform des Unternehmens basiert auf Open-Source-Technologie und kombiniert die Technologien SQL und NoSQL. Zur Klientel zählen Branchenschwergewichte wie AT&T, Disney, Ebay oder Verizon.



Contact

Gregor Bauer, Solutions Engineer, Couchbase,
gregor@couchbase.com