

Strategische Datenflexibilität

Sowohl herkömmliche Datenbankanbieter als auch Cloud-Provider setzen auf komprimierte Modelle der Datenschicht, die Kompromisse eingehen. Glücklicherweise bietet Redis Enterprise einen dritten Weg - das Beste aus beiden Welten.

Kurzfassung

Die Wahl der richtigen Datenbank ist eine kritische Entscheidung, die über Jahre und sogar Jahrzehnte hinweg Auswirkungen auf das gesamte Unternehmen und die gesamte Technologiekette hat. Leider ist die Wahl der Datenbank oft schwierig und komplex, da die vollen Auswirkungen erst deutlich werden, wenn die Datenbank in großem Maßstab oder über das ursprüngliche Datenmodell hinaus eingesetzt wird.

Ironischerweise macht die begrenzte Anzahl von Optionen die Wahl der Datenbank noch schwieriger. Die Datenbankbranche ist weitgehend in zwei Lager gespalten. Die einen plädieren für ein monolithisches relationales Datenbankmodell, die anderen drängen darauf, für jede Anwendung eine Vielzahl von zweckgebundenen Datenbanken zu verwenden. Stellen Sie sich das erste Lager wie einen Hot-Dog-Wagen vor: Sie bieten nur eines an - Hot Dogs. Das zweite Lager ist wie ein Food Court: Sie wählen aus einer Reihe von Optionen, die eventuell zusammenpassen oder auch nicht.

Relationale Datenbanken = Hotdogs

Die Hotdog-Anbieter setzen auf ein einziges Datenmodell (relational) und eine einzige Art von Software (DBMS). Unabhängig von der tatsächlichen Art der gespeicherten und abgefragten Daten müssen diese so gebogen und geformt werden, dass sie in das relationale Modell passen. Das Entwicklungsteam muss diese zusätzliche Softwareschicht schreiben, um eine Vielzahl von Datenproblemen in relationale/tabellarische Probleme umzuwandeln. Im Betrieb führt dies oft zu Leistungs- oder Skalierungsproblemen, da die Datenbank gezwungen wird, Dinge zu tun, für die sie eigentlich nicht ausgelegt ist. Egal, wie viele weitere Zutaten sie oben drauf packen, im Endeffekt werden hier doch nur Hotdogs angeboten.

“ Dieser Wagen verkauft Hotdogs. Auch wenn man Ketchup, Senf oder sogar Essiggurken hinzufügt, ist es immer noch ein Hotdog. Lassen Sie sich nichts anderes einreden. ”



“ Redis Enterprise bietet einen dritten Weg, der das Beste aus beiden Welten bietet. ”



Cloud-Datenbanken = Food-Courts

Die Food-Court-Fraktion ist der Ansicht, dass selbst für eine einzige Anwendung viele Datenmodelle erforderlich sind, und jeder mehrere, speziell entwickelte Datenbanken verwenden sollte. Diese Datenbanken sind über ein Netzwerk miteinander verbunden, was ein anfälliges, langsames Wirrwarr miteinander verbundener Dienste ergibt. Entwickler müssen dabei eine verwirrende Vielzahl von Abfragesprachen, Verbindungsmethoden und Client-Bibliotheken erlernen. Außerdem müssen sie sich Gedanken über Synchronisationsprobleme und Fehlertoleranzen bei der Verwaltung von Daten über mehrere Dienste hinweg machen. Die Betriebsteams müssen nicht nur eine, sondern mehrere Datenbankplattformen konfigurieren, skalieren, sichern und warten.

Beide Möglichkeiten sind kompromittiert. Sie sind beide konfliktanfällig und bergen Probleme mit der Leistung und Wartungsfreundlichkeit. Was noch wichtiger ist: Es ist gar nicht wahr, dass Sie sich zwischen einer der beiden Möglichkeiten entscheiden müssen. Es gibt noch einen anderen Weg, der Ihnen helfen kann, das Wirrwarr zu vermeiden und alles in ein einziges Datenmodell einzupassen.

Redis Enterprise = ein dritter Weg

Redis Enterprise präsentiert einen dritten Weg - eine einzige Betriebsschnittstelle, in der Sie mehrere Datenbanken einsetzen können. Die Datenbanken in Redis Enterprise bieten Module, die die schlüsselbasierte Funktionalität von Redis um spezifischere Datenmodelle wie Graphen, Suche, Dokumente und Zeitreihen sowie probabilistische, ereignisgesteuerte und KI-gestützte Funktionen erweitern. Entwickler haben über Standard-Redis-Client-Bibliotheken eine Schnittstelle zur Datenbank. Anstelle eines Hotdogwagens oder eines Food Courts bietet Redis Enterprise ein komplettes Menü mit gut durchdachten Optionen aus einem einzigen Restaurant.

Einführung

Das Herzstück der meisten modernen Software sind Daten. Die Entscheidung für eine Datenbank ist eine der wichtigsten technologischen Entscheidungen für Technologieführer. Diese Entscheidung wurde durch die Entwicklung der Speicherung selbst beeinflusst. Die frühesten Datenspeicher waren physisch, buchstäblich Löcher in Karten oder Papierstreifen. Im Laufe der Jahre sind wir zu magnetischen Medien und schließlich zu Festkörperspeichern übergegangen. Heute geht es nicht mehr nur darum, wie Bits auf physischen Medien gespeichert werden, sondern auch darum, wie die Daten modelliert und in nutzbare Informationen umgewandelt werden.

Ein Großteil der alten Datenbanksoftware wurde jedoch in einer Zeit entwickelt, in der die heute verfügbaren Abstraktionsebenen noch nicht existierten - Software konnte nur auf physischen Medien bereitgestellt werden.

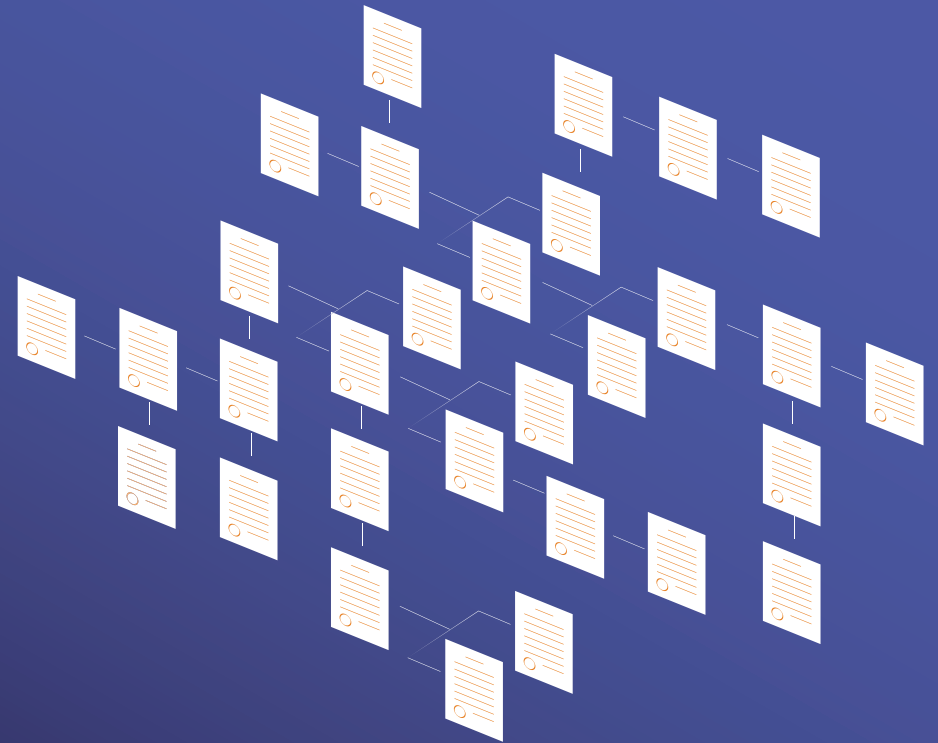
Es gibt immer noch eine große Anzahl von Datenbanken alter Bauart, die auch auf nicht veralteten Systemen installiert sind. Diese Datenbanken basieren auf einem einzigen, monolithischen, relationalen Modell. Dieses Modell ist zwar nach wie vor nützlich für die Modellierung einiger Datenprobleme, aber es stellt (inzwischen) eine generische Speichermaschine dar, die zu oft anstelle geeigneter Datenmodellierungstechniken verwendet und fehlverwendet wird.

Cloud-Anbieter hingegen lassen Maschinen ganz hinter sich und stellen speziell entwickelte Datenbanken bereit, die ausschließlich über ihren Clouddienst verfügbar sind. Diese verbinden Sie dann „einfach“ mit Ihrer Anwendung. Sie benötigen eine Datenbank mit einem anderen Modell? Legen Sie dann „einfach“ eine neue Datenbank an und stellen Sie eine Verbindung zu ihr her.

Entscheidend ist, dass sowohl die Cloud als auch alte Datenbanken grundlegende Kompromisse beinhalten. In vielen Fällen führen sie zu einem zyklischen Wechsel zwischen verschiedenen technologischen Ansätzen. Der wiederholte Versuch, die Beschwerden einer bestimmten Gruppe von Kunden zu lindern, geht letztendlich auf Kosten anderer Kunden.

Für alle Deutlichkeit: Kompromisse in Ihrer Datenbankstrategie können zu internen Problemen (Zufriedenheit und Leistung der Mitarbeiter, Langlebigkeit der Projekte) und externen Problemen (Leistung und Stabilität) führen. Glücklicherweise bietet die dritte Option einen Mittelweg, der die Fallstricke von Legacy-Datenbanken und Cloud-Anbietern vermeidet. Eine strategische Option, die flexibel genug ist, um sich an neue Datenmodelle anzupassen und einen einfachen und stabilen Betrieb zu ermöglichen.

“ Ein Großteil der alten Datenbanksoftwares wurde in einer Zeit entwickelt, in der die heute verfügbaren Abstraktionsebenen einfach noch nicht existierten. “





Positionen in der Datenbankindustrie

Angesichts der beiden konkurrierenden Ansätze für das Datenbankdesign ist es nicht verwunderlich, dass ein Großteil der Datenbankindustrie in zwei Lager fällt: altbewährte Datenbankanbieter und Cloud-Anbieter. Jedes Lager sieht die Landschaft anders und setzt sich für ganz unterschiedliche Lösungen ein.

Altbewährte Datenbankanbieter

Schon früh hat man in der Informatik darüber nachgedacht, wie man Daten in einem standardisierten Format speichern kann. Charles W. Bachman entwickelte 1963 den Integrierten Datenspeicher bei General Electric, die erste Datenbank, die wir heutzutage als solche erkennen würden. Einige Jahre später, 1970, schrieb Edgar Codd „Ein relationales Datenmodell für große gemeinsam genutzte Datenbanken“, welches als Vorlage für das relationale Datenbankmodell diente. Larry Ellison griff die Ideen aus dieser Abhandlung auf und nutzte sie als Grundlage für die 1979 veröffentlichte Oracle-Datenbank.

Das relationale Modell erwies sich (für die damalige Zeit) als sehr flexibel und wurde schließlich zum De-facto-Datenbankstandard. Das relationale Modell und die Datenbank wurden in den meisten Gesprächen funktional gleichwertig. Mit diesem leistungsstarken Modell können Sie viele verschiedene Dinge über eine einzige Schnittstelle erledigen.

Egal, ob es sich um ein einfaches Projekt für einige wenige Benutzer oder um ein komplexes unternehmensweites Datenprojekt handelte, das relationale Modell war die Antwort. Natürlich wurden unzählige Watts Energie verschwendet, als riesige relationale Datenbanken eingeführt und gewartet wurden, ohne dass je komplexere Operation durchgeführt wurden, als das Nachschlagen von Elementen anhand eines Primärschlüssels in einer einzigen Tabelle. Die Absicht war nicht, verschwenderisch zu sein, aber jahrzehntelang waren relationale Datenbanken die einzige brauchbare, bekannte und gut verstandene Option, so dass sie für so ziemlich alles eingesetzt wurden.

Daten, die auf völlig unterschiedliche Art und Weise modelliert wurden, wurden ebenfalls in die relationale Datenbank eingefügt, was zu Schwierigkeiten bei der Modellierung führte. Die Abbildung von

“ Unzählige Watt an Energie wurden verschwendet, als riesige relationale Datenbanken eingeführt und gewartet wurden, ohne dass je komplexere Operation durchgeführt wurden, als das Nachschlagen von Elementen anhand eines Primärschlüssels in einer einzigen Tabelle. ”

Diagramm Daten auf eine relationale Datenbank beispielsweise führt zu Indizierungsproblemen und massiven Leistungseinbußen. Ebenso sind Zeitreihendaten eine schreiboptimierte, aggregationslastige Arbeitslast, die viele relationale Datenbanken einfach nicht in Echtzeit verarbeiten können.

Bei der Speicherung von Dokumenten in einer relationalen Datenbank wird versucht, diese Art von schemafreien Freiformdaten in ein stark typisiertes und strukturiertes Tabellenformat zu pressen. Alle diese Datenmodelle sind völlig legitim, aber die Allgegenwart von relationalen Datenbanken hat die Komplexität und die Modellanpassung in die Anwendung hinein verlagert. Das Ergebnis war oft eine relationale Datenbank, die Daten enthielt, auf die nur eine bestimmte Anwendung zugreifen konnte, und eine überlastete Datenbank, die damit kämpfte, mit einem Nutzungsmuster Schritt zu halten, für das sie nicht konzipiert war.

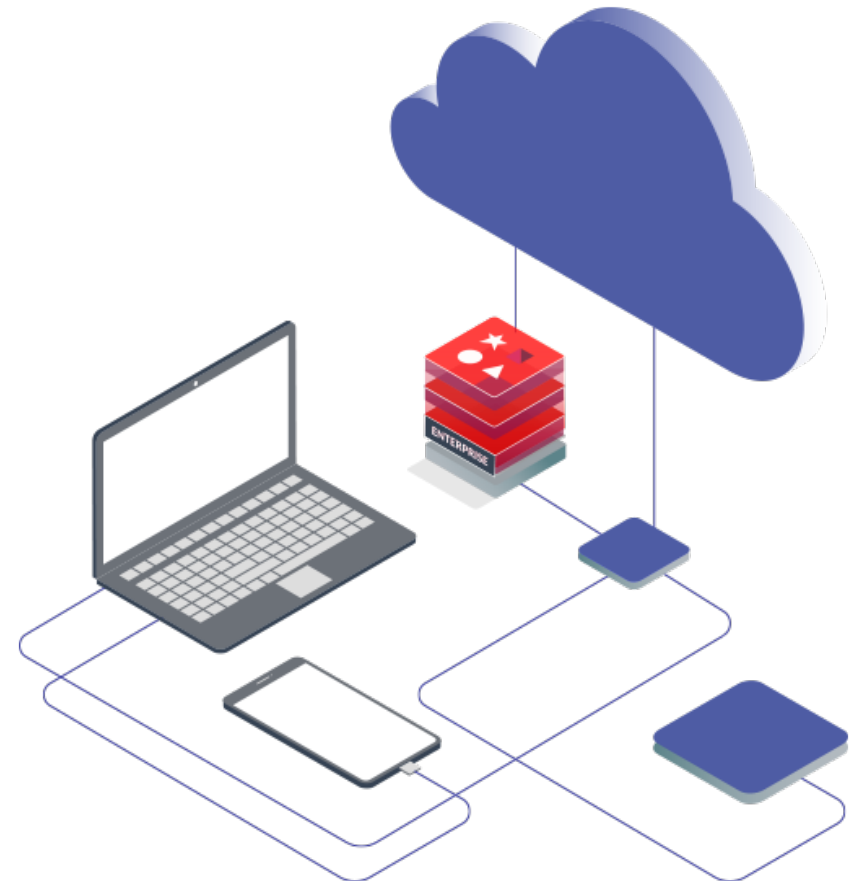
Auch hier soll nicht gesagt werden, dass das relationale Modell völlig ungeeignet ist. Wie bei jedem Datenmodell gibt es Probleme, für die relationale Datenbanken die optimale Wahl sind. Anwendungen mit stark relationalen Daten und/oder solchen, die nicht effektiv denormalisiert werden können, eignen sich nach wie vor hervorragend für diese Art von Datenbanken. Aber so wie mit einem Hammer alles wie ein Nagel aussieht, erscheinen mit einer relationalen Datenbank alle Probleme relational.

Fairerweise muss man sagen, dass die Entscheidung für eine einzige Datenbankplattform einige betriebliche Vorteile mit sich bringt. Die Fachleute, die diese Plattformen betreiben, haben schnell und gründlich gelernt, wie man sie stabil halten kann. Da es sich um große, komplexe Software handelt, können relationale Datenbanksysteme auf Unternehmensebene betrieben werden, anstatt speziell für eine einzelne Software eingesetzt zu werden, wodurch Komplexität und Fachwissen zentralisiert werden. Der Einheitsansatz kann zwar zu Entwicklungsproblemen führen, aber er schafft auch einen einzigen, bekannten Vektor, der operationalisiert werden kann.

Die Wurzeln dieser Systeme stammen jedoch aus einer Zeit vor der Geburt vieler Menschen, die sie betreiben - vor dem modernen Internet, geschweige denn vor dem Cloud Computing. Zugegeben, in dieser Softwareklasse gibt es, wenn überhaupt, nur noch wenig Quellcode, der aus den späten 1970er Jahren stammt, aber die zugrundeliegenden Konzepte stammen aus einer komplett anderen Zeit.

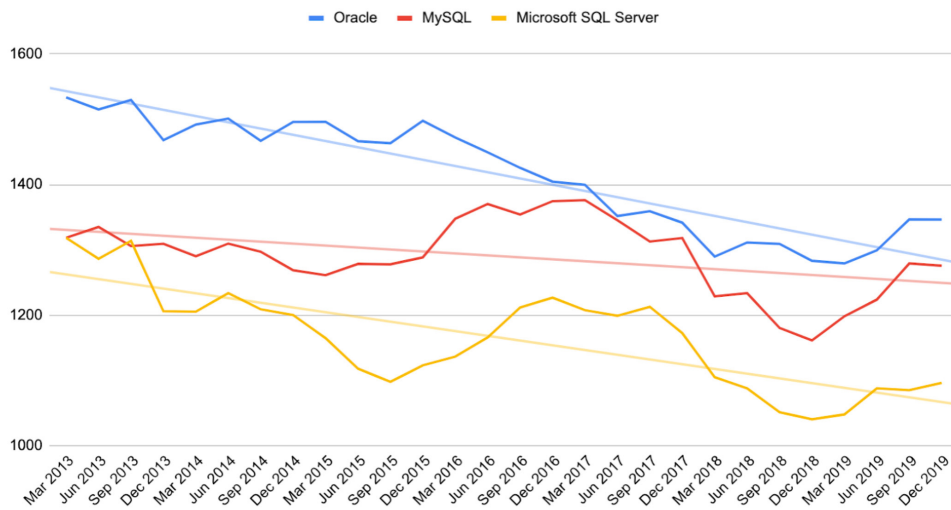
Die Einführung der Cloud

Als Unternehmen begannen, Software in die Cloud zu verlagern, versuchten viele, dies zu tun, ohne ihre Anwendungen neu zu archivieren (bekannt als „Lift-and-shift“). Leider entsprachen die Annahmen, die auf architektonischer Ebene für viele Softwarekomponenten getroffen wurden, nicht der drastisch veränderten Cloud-Umgebung - RDBMS waren da keine Ausnahme. Jahrzehntelange Low-Level- und Mikro-Optimierungen hatten Software hervorgebracht, die in lokalen Umgebungen mit akzeptabler Leistung lief. In einer virtualisierten Umgebung ergaben sich jedoch neue Herausforderungen: Geringere Leistung von Netzwerkgebundenem Speicher, kurzlebige Rechenressourcen und fehlende Automatisierung machten den Weg in die Cloud steinig. Die ersten Lift-and-Shift-Ausführungen zeigten also, dass Betriebsexperten nicht nur Experten für massive Datenbanksoftware sein müssen, sondern auch für die von ihnen verwendete Cloud. Die Welt hatte sich verändert.



“ Herkömmliche Datenbankanbieter versuchten, eine Datenbank für ein Unternehmen bereitzustellen, während Cloud-Anbieter versuchen, viele Datenbanken für eine einzige Anwendung bereitzustellen. ”

Top 3 Relationale Datenbanken DB-Engines Scores, Quarterly 2013-2019



Die moderne Ära des Cloud Computing hat die Popularität der herkömmlichen Datenbanken beeinträchtigt. Während relationale Datenbanken in der Rangliste von DB-Engines.com immer noch die ersten drei Plätze belegen, ist der Trend eindeutig rückläufig. Seit DB-Engines 2013 mit dem Tracken der Bewertungen begonnen hat, ist die Beliebtheit der besten relationalen Modelle laut der **DB-Engines-Bewertungsmethodik insgesamt zurückgegangen**.

Cloud-Schatten: Relationale Datenbanken bleiben beliebt... vorerst

Cloud-Anbieter

Am anderen Ende des Spektrums sitzen die Cloud-Anbieter. Angesichts ihren Produktlinien scheinen die Cloud-Anbieter die Idee abzulehnen, dass das relationale Modell die Standardmethode zum Speichern und Abrufen von Daten ist. Die verschiedenen Datenmodelle, die in den Datenbankprodukten der Cloud-Anbieter vertreten sind, legen nahe, dass sich die Daten Ihrer Anwendung nicht nach der verfügbaren Datenbank richten sollten, sondern dass Sie stattdessen die richtige(n) Datenbank(en) für Ihre Anwendung auswählen sollten.

Die Cloud-Anbieter unterstützen NoSQL, ein Begriff, der bereits 1998 geprägt wurde, aber erst seit 2009 seine aktuelle Bedeutung trägt. NoSQL-Datenbanken konzentrierten sich ursprünglich auf die Bereitstellung eines einzigen Datenmodells zur Unterstützung einer kürzeren Liste von Anwendungsfällen. Sie sind speziell für ein einziges Datenmodell konzipiert. Jede Anwendung von ausreichender Komplexität hat jedoch wahrscheinlich ein breites Spektrum von Daten, die auf unterschiedliche Weise modelliert werden sollten, so dass eine einzige Anwendung mehrere verschiedene Datenbanken erfordern kann. Während herkömmliche Datenbankanbieter eine einzige Datenbank für ein Unternehmen bereitstellten, können Cloud-Anbieter viele Datenbanken für eine einzige Anwendung bereitzustellen.

Mehrere Datenbanken in einem einzigen Projekt können helfen, die Probleme der Entwickler zu lösen. Relationale Datenbanken erzeugen Ergebnisse, bei denen sich Entwickler auf Bibliotheken oder ihren eigenen Code verlassen müssen, um Daten so zu ordnen und zu trennen, dass sie in das relationale Modell passen (das **objekt-relationale Impedanzfehlpassungsproblem**). Die Eingaben und Ausgaben von NoSQL-Datenbanken ähneln in der Regel der internen Datenmodellierung des Anwendungscode. Für Entwickler, die mit zweckgebundenen Datenbanken arbeiten, ist der Anteil der Zeit, der auf das Schreiben von relevantem, problemlösendem Code entfällt, größer als bei relationalen Datenbanken, bei denen, die Daten noch in eine nutzbare Form gebracht werden müssen.

Die Verwendung mehrerer zweckbestimmter Datenbanken in einer einzigen Anwendung hat jedoch auch ihre Nachteile. Wie bei jedem anderen Werkzeug sind auch hier Fertigkeiten und Techniken erforderlich; jede Datenbank erfordert von den Entwicklern nicht nur das Erlernen der konzeptionellen Grundlagen der Datenbank und des Modells, sondern auch der zugehörigen Bibliotheken. Zum Beispiel ist das Verständnis der Verbindungsparameter für Datenbanken oft eine nicht triviale Aufgabe, die bei unsachgemäßer Handhabung zu Leistungs- oder Stabilitätsproblemen führen kann.

Obwohl sie cloudbasiert sind, stellen mehrere Datenbanken auch eine betriebliche Herausforderung dar. Eine dieser Herausforderungen besteht in der Verwaltung einer großen Anzahl von Cloud-Diensten in Ihrem Unternehmen. Da jede Anwendung von vielen Datenbanken bedient wird, müssen sich Ihre Betriebsteams mit mehr verschiedenen Softwarekomponenten auseinandersetzen. Cloud-Dienste sind einfacher zu warten als lokale Dienste, aber die betriebliche Wartung bleibt variabel und ist nicht gleich null. Einige Datenbanken führen beispielsweise nahtlose Upgrades durch, während andere einen Eingriff erfordern. Einige Datenbanken müssen speziell für Ihren Kontext konfiguriert werden, während andere perfekt mit den Standardeinstellungen laufen.

Latenzzeit und Skalierung

Zusätzlich zu den Auswirkungen auf die Gesamtkomplexität, die mehrere Datenbanken pro Anwendung haben, wird auch die Latenz beeinträchtigt. Darüber hinaus kann es schwierig sein, die Gesamtlatenzzeit in einem komplexen System in den Griff zu bekommen. Wenn ein einzelner Vorgang einer Anwendung über mehrere Datenbanken läuft, wirkt sich die kumulative Latenz aller Datenbanken und der Anwendung auf die Gesamtleistung der Anwendung aus. Während eine einzelne Datenbankplattform viele Vorgänge in einer einzigen Anfrage ausführen kann, kann es in der Cloud vorkommen, dass jeder Vorgang in einer separaten Datenbank stattfindet. Der jeweilige Hin- und Rückweg von und zur Anwendung trägt in erheblichem Maße zur Gesamtlatenzzeit bei. So können synchrone Operationen viele Roundtrip-Netzsprünge erfordern (was die Latenz erhöht), während andere parallelisiert werden können, so dass alle Netzsprünge zur gleichen Zeit erfolgen (in diesem Fall wäre die Gesamtlatenz die höchste Latenz der Gruppe).

Ein weiterer Aspekt ist die Skalierung. Jede Datenbank, die die Anwendung unterstützt, kann unterschiedlich skaliert werden, aber die Anwendung funktioniert immer nur so gut wie ihr schwächstes Glied. Selbst mit einer optimierten Infrastruktur für jede Datenbank ist die Betriebsstrategie komplex, da die Leistung bei der einen oder anderen Datenbank bei der Skalierung auf ein Plateau stößt.

“ Für jede Datenbank müssen die Entwickler nicht nur die konzeptionelle Grundlage der Datenbank und des Modells, sondern auch die der zugehörigen Bibliotheken erlernen. ”



Die Gefahr der zu kurzen Decke

Stellen Sie sich vor, Sie versuchen, ein Bett mit einer Decke zu machen, die nicht lang genug ist, um die ganze Matratze zu bedecken. Wenn die Decke das Fußende erreicht, liegt der obere Teil frei. Wenn man sie nach oben zieht, bedeckt sie nicht mehr das Fußende. Wenn die Decke in die Mitte des Bettes legt, liegen sowohl Kopfende als auch Fußende frei. Alle Optionen sind unzufriedenstellend.

In ähnlicher Weise bieten die altbekannten Datenbankanbieter eine einheitliche organisatorische Bereitstellung und Skalierung, aber ihre unpassenden Modelle müssen durch zusätzliche Entwicklungskomplexität behoben werden (siehe *unten links*). Cloud-Anbieter stellen eine Reihe von zweckgebundenen Datenbanktypen zur Verfügung, was leider zu betrieblicher Komplexität führt, da mehrere Datenbanken pro Anwendung verbunden, skaliert und ausgeführt werden müssen (siehe *unten rechts*).

Letztlich sind beide Ansätze unzureichend. Durch die Verlagerung der Komplexität auf Entwicklungs- oder Betriebsteams wird die Gesamtkomplexität nicht verringert. Der enorme technische Aufwand, der erforderlich ist, um eine Anwendung von einem einzigen Datenmodell in einer herkömmlichen Datenbank auf viele speziell entwickelte Datenbanken in der Cloud zu verlagern, könnte tatsächlich Zeitverschwendung sein, wenn dadurch die Komplexität nicht wirklich reduziert, sondern lediglich einem anderen Team zugewiesen wird.

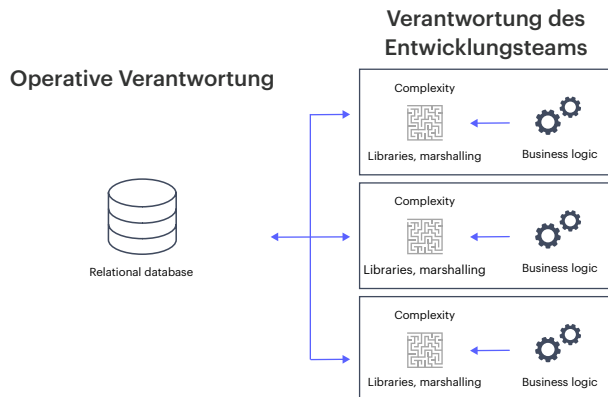
Noch schlimmer ist, dass die Migration einer Anwendung in die Cloud einen endlosen Zyklus von Refactoring nach sich ziehen kann. Nehmen wir zum Beispiel eine Anwendung, die auf einer Vielzahl von Datenmodellen gut und stabil läuft, aber in eine relationale Datenbank gezwängt ist. Die Anwendung mag zwar gut laufen, aber der zugrunde liegende Code ist langwierig und schwer zu warten - ein Großteil davon



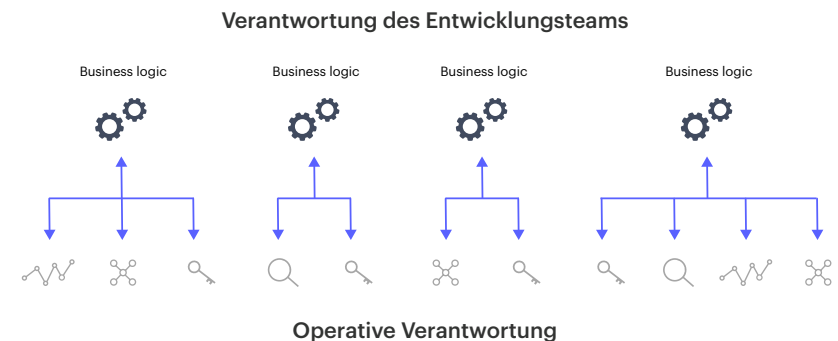
betrifft nicht die Geschäftslogik, sondern eher objektrelationale Impedanzabweichungsprobleme. Der Umfang des Codes erschwert das Hinzufügen neuer Funktionen und Korrekturen.

Um dieses Problem zu lösen, sieht das Entwicklungsteam z. B. eine Graphdatenbank, eine Zeitreihendatenbank und eine Suchdatenbank, die große Teile der bestehenden Codebasis überflüssig machen würden. Die Umstellung auf diese speziell entwickelten Datenbanken erscheint sinnvoll. Es erfolgt ein Re-Engineering und die nächste Version der Anwendung wird so geschrieben, dass sie mit diesen drei neuen Datenbanken kommunizieren kann.

Das Entwicklungsteam arbeitet schnell an neuen Funktionen... aber die Produktionsumgebung ist jetzt langsam und instabil. Das Betriebsteam sieht Probleme mit der Betriebszeit und hat Schwierigkeiten mit der Fehlerbehebung bei mehreren neuen Datenbanken. Es wird schwierig, das genaue Problem zu diagnostizieren, da es sich bei den in der Cloud bereitgestellten Lösungen um Black Boxes handelt, die nur einen begrenzten Einblick in die problematischen Datenbanken bieten. Das Betriebsteam beginnt sich für den Wechsel zu einer anderen Datenbankstrategie einzusetzen (höchstwahrscheinlich zurück zu einer relationalen Datenbank), die stabiler und leistungsfähiger ist. Der Kreislauf von Problemen und Kosten beginnt erneut, wie wenn man die Decke auf dem Bett hin und her zieht, weil sie einfach nicht passt.



Die Komplexität bleibt auf der Seite des Entwicklungsteams, da es relationale Daten so abbilden muss, dass sie das wahre Modell der Anwendung (Geschäftslogik) widerspiegeln.



Die Komplexität der Anwendung liegt in der Verbindung mit und der Pflege von mehreren, speziell entwickelten Datenbanken, wie sie von Cloud-Anbietern empfohlen werden.

“ Eine strategische Option ist es, einen Weg zu finden, der das Beste aus beiden Welten vereint. Ein Weg, der die Koexistenz mehrerer nativer Datenmodelle bei gleichzeitiger vorhersehbarer Skalierung ermöglicht. ”



Strategische Flexibilität mit Redis Enterprise

Bislang wurden in diesem Dokument nur zwei Optionen behandelt, so unzureichend sie auch sein mögen. Sowohl der RDBMS-for-everything-Ansatz als auch der Ansatz mit mehreren Datenbankmodellen in der Cloud funktionieren nur auf taktischer Ebene, lösen ein Problem auf reglementierte Weise, verlagern die Schwierigkeiten innerhalb der Organisation und bereiten die Organisation letztlich auf zyklische, teure Re-Engineering-Projekte vor.

Eine strategischere Option ist es, einen Weg zu finden, der das Beste aus beiden Welten vereint. Ein Weg, der die Koexistenz mehrerer nativer Datenmodelle bei gleichzeitiger vorhersehbarer Skalierung ermöglicht. Ein Weg, der es Entwicklern ermöglicht, eine Bibliothek zu verwenden, eine Verbindung zu einem Endpunkt herzustellen und auf mehrere Modelle zuzugreifen. Dieser strategische, flexible Weg existiert - und er führt zu Redis Enterprise.

Redis Enterprise ist eine Datenbankplattform, die auf Redis basierend aufgebaut ist. Redis bietet ohne Erweiterungen ein Key-Value-Datenzugriffsmodell mit dem Zusatz von Datenstrukturen wie Hash-Maps, Listen, sortierten Mengen, Streams und mehr. Redis kann mit Modulen erweitert werden, die der Datenbank völlig neue Funktionen verleihen. Diese Module implementieren native Datenmodelle wie Graphen, Volltextsuche, Dokumentenspeicherung und Zeitreihen sowie zusätzliche Funktionen rund um künstliche Intelligenz und ereignisgesteuertes Scripting.

Die Vorteile des Redis Enterprise-Ansatzes lassen sich auf drei Schlüsselfaktoren reduzieren:

1. Eine operative Schnittstelle, mit mehreren Datenmodellen
2. Eine einfache, einheitliche Schnittstelle
3. In der Cloud entstanden und entwickelt

Eine operative Schnittstelle, mit mehreren Datenmodellen

Redis Enterprise bietet das Beste aus beiden Welten - es bietet eine einzige operative Schnittstelle, auf der viele Datenmodelle auf einer einzigen Datenebene koexistieren können. Redis Enterprise wurde aus der Notwendigkeit heraus entwickelt, dass Anwender in Unternehmen ihre Redis-Instanzen operationalisieren können. Das Redis-Ökosystem hat ein robustes, steckbares Modulsystem entwickelt, das tief in das Herz von Redis eingreift und es Redis ermöglicht, über das eingebaute Schlüssel/Wert-Modell hinauszugehen und andere Datenmodelle zu hosten. Dieser Multi-Modell-Ansatz ermöglicht es, die operativen Merkmale von Redis zu übernehmen und gleichzeitig leistungsstarke neue Funktionen hinzuzufügen. Wenn Sie skalieren, brauchen Sie nur mehr Infrastruktur zu Redis Enterprise hinzuzufügen.

Das Aktivieren eines Moduls in Redis Enterprise ist so einfach wie das Klicken auf eine Schaltfläche in einer Benutzeroberfläche, aber es verändert komplett die Art und Weise, wie Daten gespeichert und abgefragt werden. Redis Labs hat eine Reihe von Modulen entwickelt, die eine Reihe verschiedener Datenmodelle ermöglichen, darunter:



RedisSearch

Volltextsuche und Sekundärindexierung



RedisGraph

Cypher-kompatible Graphen-Engine



RedisJSON

JSON-Dokumentenspeicher



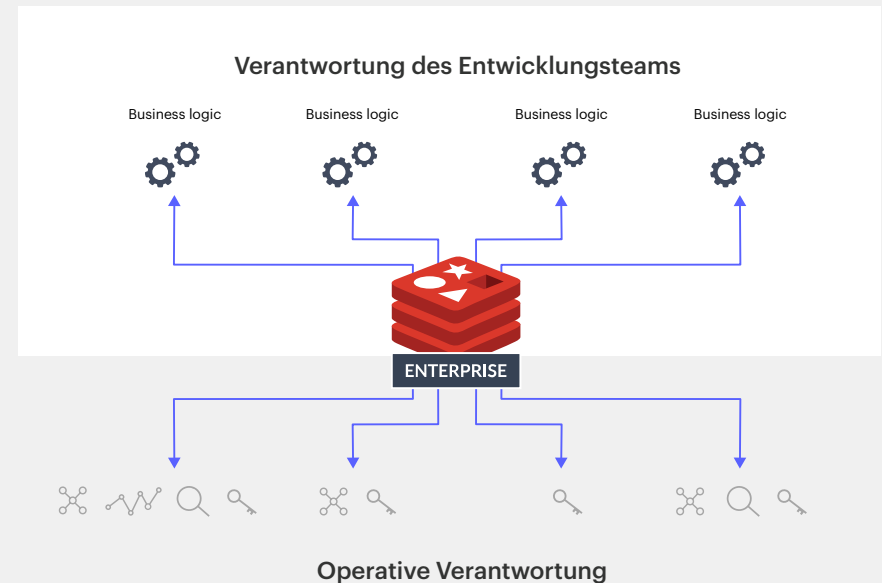
RedisTimeSeries

Aufzeichnung von Datenproben in großen Mengen sowie Zusammenführung und Reduzierung von Proben während des Betriebs

Die Module sind optional, Sie können nur die Module aktivieren, die Sie für ein bestimmtes Projekt benötigen. Redis Enterprise ist mandantenfähig, d. h. Sie können Datenbanken für individuelle Bedürfnisse mit spezifischen Modulen erstellen oder Datenbanken mit vielen Modulen, die zusammenarbeiten, so dass Sie sich auf die spezifischen Fähigkeiten konzentrieren können, die Sie für Ihre Anwendung benötigen.

Diese Module ermöglichen zusammen mit der in Redis integrierten Key-Value-Datenstruktur eine breite Palette von Anwendungen. Es ist wichtig zu verstehen, dass es sich bei diesen Modulen nicht um Fähigkeiten

“ Redis Enterprise bietet das Beste aus beiden Welten - es bietet eine einzige operative Schnittstelle, auf der viele Datenmodelle auf einer einzigen Datenebene koexistieren können. ”



Redis Enterprise ermöglicht eine elegante Trennung von Entwicklung und Betrieb, ohne die Flexibilität des Datenmodells zu beeinträchtigen.

“ Redis Enterprise ist Cloud-unabhängig. Redis Labs ist kein Cloud-Anbieter, daher können Sie die Cloud- und Bereitstellungsoptionen - in der Cloud oder vor Ort - wählen, die für Ihr Unternehmen und Ihre Anwendungen am besten geeignet sind. “



handelt, die auf dem Key-Value-Modell aufbauen, sondern um Modelle, die auf die Interna von Redis zugreifen, um Daten im Speicher zu speichern und auf der Festplatte zu halten. Viele dieser Modelle können in derselben Datenbank aktiviert werden, so dass es beispielsweise möglich ist, eine Volltextsuche über Graphknoten durchzuführen oder JSON-Dokumente neben Zeitreihendaten zu speichern. Da die verschiedenen Datenmodelle auf der gleichen Datenebene existieren, sind sogar Transaktionen zwischen den Modellen möglich.

Eine vereinfachte, einheitliche Schnittstelle

Da eine Redis-Datenbank mehrere Modelle aus der gleichen -Datenebene bedienen kann, wird sie zu einem einzigen Verbindungspunkt. Alle Verbindungseigenschaften sind in einer einzigen zusammengefasst, und es gibt keine Leistungseinbußen oder zusätzliche Latenzzeiten für die Verwendung mehrerer nativer Datenmodelle. Darüber hinaus müssen Entwickler nur eine einzige programmatische Schnittstelle für alle verschiedenen Datenmodelle verstehen - alles läuft auf das Redis-Protokoll hinaus. So erhalten Sie die Vorteile der Verwendung separater Datenbanken in einem einzigen Paket.

In der Cloud entstanden und entwickelt

Redis war schon immer eine Lösung für die Cloud. Redis Enterprise erweitert die Lösung um ein Gesamtpaket, das sich der Cloud-Umgebung bewusst ist und Funktionen wie Zonen- und Rack-Bewusstsein, Überwachung und zweigeteilte Daten- und Verwaltungsebenen bietet.

Da Redis den Systemspeicher für die primäre Datenspeicherung nutzt, entsteht eine Betriebsdynamik, die sich gut für den Betrieb in jeder Cloud oder in mehreren Clouds eignet. Redis Enterprise unterstützt auch Redis on Flash, wodurch SSD-basierter Speicher auf Cloud-Instanzen als Erweiterung von DRAM fungieren kann, was eine hohe Leistung von sehr großen Datensätzen auf derselben Infrastruktur zu erschwinglichen Kosten ermöglicht. Als In-Memory-Datenbank ist Redis bereit für die nächsten Generationen der Datenspeicherung, da sich die Branche von rotierenden Festplatten und blockbasiertem Flash-Speicher zu byteadressierbarem, dauerhaftem Speicher weiterentwickelt.

Bezeichnenderweise ist Redis Enterprise Cloud-unabhängig. Redis Labs ist kein Cloud-Anbieter. Sie können also die Cloud wählen, die für Ihr Unternehmen und Ihre Anwendungen am besten geeignet ist. Dies verringert die Abhängigkeit von der Cloud und ermöglicht Hybrid-Cloud- und Multi-Cloud-Bereitstellungen. Darüber hinaus bietet Redis Labs eine Vielzahl von Bereitstellungsmethoden: Die Auswahl reicht von einer vollständig verwalteten Cloud-Umgebung bis hin zu lizenzierten On-Premises-Implementierungen.

Redis Enterprise: eine strategische Lösung

Die Lösung eines einzelnen Problems ist oft relativ unkompliziert. Entscheidungen zu treffen, die mehrere Probleme lösen, ist in der Regel eine größere Herausforderung. Wenn dann noch die Notwendigkeit besteht, mehrere aktuelle Probleme zu lösen und gleichzeitig mehrere Probleme für die Zukunft anzugehen, können die Dinge extrem komplex werden. Dies ist das Herzstück der strategischen Problemlösung. Manchmal werden Datenbankentscheidungen leichtfertig getroffen, manchmal sind sie schon vor Beginn eines Projekts festgelegt. Beides ist keine strategische Entscheidung.

Redis Enterprise bietet einen neuen Weg, der sowohl Entwicklungs- als auch Betriebsprobleme strategisch lösen kann, nicht nur heute, sondern auch in absehbarer Zukunft. Es verringert die Probleme der Datenmodelle in der Entwicklung sowie bei der Leistung und Stabilität im Betrieb und trägt dazu bei, eine Entscheidung zu treffen, die sich langfristig auszahlen wird.



Freiheit, Ihr eigenes Datenmodell mitzubringen

Anwendungen, die mit Redis Enterprise erstellt wurden, funktionieren besser, weil sie Daten so speichern, wie Ihre Anwendung sie nutzt. Redis Enterprise unterstützt eine Vielzahl von eingebauten, reichhaltigen Datenmodellen, die je nach Aufgabenstellung verwendet werden können. Als vollständige Implementierungen versucht Redis Enterprise nicht, Daten in unpassende Datenmetaphern zu pressen, was Ihnen hilft, Code und Komplexität auf Anwendungsebene zu reduzieren. Die Reduzierung des Codes und der Komplexität auf Anwendungsebene führt zu produktiveren Entwicklern, die sich auf Ihre Geschäftsprobleme konzentrieren und nicht versuchen, Daten in ein ungeeignetes Datenmodell zu pressen.



Integrierte betriebliche Flexibilität

Da Redis Enterprise die Koexistenz vieler Datenmodelle innerhalb desselben Clusters oder sogar desselben Datenraums ermöglicht, eliminiert es Verbindungen zwischen verschiedenen,

“ Redis Enterprise bietet einen neuen Weg, der sowohl Entwicklungs- als auch Betriebsprobleme strategisch lösen kann, nicht nur heute, sondern auch in absehbarer Zukunft. “



“ Die Datenbank-TCO sollte nicht nur den Preis pro Zeiteinheit, sondern auch die organisatorischen Auswirkungen berücksichtigen. Ihre Datenbank sollte so flexibel sein, dass sie von Haus aus mehr als ein einziges Datenmodell unterstützt und die Datenschicht optimiert und zukunftssicher macht. ”



unabhängigen Datenbanken. **Die Konsolidierung mehrerer Modelle unter einer Verbindung verringert die Brüchigkeit und Instabilität, die durch mehrere Verbindungen entstehen.** In Redis Enterprise können Sie sogar Transaktionen zwischen Datenmodellen durchführen, was bei einer Sammlung unabhängiger Datenbanken nicht möglich ist.



Kein Cloud-Lock-in

Da es nicht von einem Cloud-Anbieter stammt, ist **Redis Enterprise nicht an eine bestimmte Cloud gebunden - Sie können Ihre Daten bei Bedarf in eine völlig neue Cloud bringen oder sogar in einer hybriden oder Multi-Cloud-Konfiguration einsetzen.** Sie haben die volle Kontrolle über die Infrastruktur, auf der Redis Enterprise läuft, und können problemlos weitere Infrastrukturen zur Unterstützung eines Clusters hinzufügen. Innerhalb des Clusters können Sie isolierte oder kombinierte Datenbanken mit speziell entwickelten Datenmodellen erstellen.



Flexibilität der Infrastruktur

Die Gesamtbetriebskosten einer Datenbank sollten nicht nur den Preis pro Zeiteinheit, sondern auch die organisatorischen Auswirkungen berücksichtigen. Die von Ihnen gewählte Lösung sollte so flexibel sein, dass sie von Haus aus mehr als ein einziges Datenmodell unterstützt und die Datenschicht gegen unvorhergesehene Verschiebungen und Änderungen optimiert und zukunftssicher macht. **Redis Enterprise vereinfacht die Herausforderungen auf der Datenebene, indem es Entwicklern und Architekten die Freiheit und Flexibilität gibt, Anwendungen zu erstellen, die Leistung bieten, ohne die betriebliche Effizienz zu beeinträchtigen.** Dieses Gleichgewicht fördert die Interessen sowohl derjenigen, die die Software schreiben, als auch derjenigen, die mit dem Betrieb der Software betraut sind, und minimiert die Überlastung jedes einzelnen Teils der Organisation.

Sowohl die traditionellen Datenbankanbieter als auch die Cloud-Provider verkünden lautstark, dass sie den einzig wahren Weg kennen und der jeweils andere völlig falsch liegt. Beide Seiten üben berechnete Kritik an den Schwächen der Datenmodellierung und der operativen Komplexität. Zum Glück gibt es nicht nur zwei Optionen: Sie haben Wahlmöglichkeiten außerhalb dieser falschen Dichotomie. Weitere Informationen über die Vorteile von Redis Enterprise finden Sie unter redislabs.com/modules.

Über Redis Labs

Moderne Unternehmen sind auf die Leistungsfähigkeit von Echtzeitdaten angewiesen. Mit Redis Labs liefern Organisationen sofortige Erfahrungen auf höchst zuverlässige und skalierbare Weise.

Redis Labs ist die Heimat von Redis, der weltweit populärsten In-Memory-Datenbank und kommerzieller Anbieter von Redis Enterprise, der weltweit überlegene Leistung, unvergleichliche Zuverlässigkeit und beispiellose Flexibilität für Personalisierung, maschinelles Lernen, IoT, Suche, E-Commerce, Social- und Metering-Lösungen bietet.

Redis Labs, das in den führenden Analystenberichten zu NoSQL, In-Memory-Datenbanken, operativen Datenbanken und Database-as-a-Service (DBaaS) stets als führend eingestuft wird, vertrauen mehr als 7.400 Unternehmenskunden, darunter fünf Fortune-10-Unternehmen, drei der vier führenden Kreditkartenherausgeber, drei der fünf führenden Kommunikationsunternehmen, drei der fünf führenden Unternehmen im Gesundheitswesen, sechs der acht führenden Technologieunternehmen und vier der sieben führenden Einzelhändler.

Redis Enterprise, das als Dienst in öffentlichen und privaten Clouds, als herunterladbare Software, in Containern und für hybride Cloud-/On-premises-Implementierungen verfügbar ist, unterstützt beliebte Redis-Anwendungsfälle wie Hochgeschwindigkeitstransaktionen, Job- und Warteschlangenmanagement, Speichern von Benutzersitzungen, Echtzeit-Dateneinspielung, Benachrichtigungen, Inhaltscaching und Zeitreihendaten.

Folgen Sie uns:



redislabs.com