Workload-Isolation mit Shuffle-Sharding

Colm MacCárthaigh

Workload-Isolation mit Shuffle-Sharding

Auf Amazon Route 53 befinden sich heute viele der weltweit größten Unternehmen und beliebtesten Websites. Die Anfänge waren jedoch weitaus bescheidener.

DNS-Hosting in Angriff nehmen

Nicht lange nach dem Beginn des Serviceangebots von AWS haben AWS-Kunden klargestellt, dass sie unsere Services Amazon Simple Storage Service (S3), Amazon CloudFront und Elastic Load Balancing am „Stamm“ ihrer Domain nutzen möchten, d. h. Namen wie „amazon.com“ und nicht nur für Namen wie „http://www.amazon.com/“.

Das mag sehr einfach erscheinen. Aufgrund einer Entwurfsentscheidung im DNS-Protokoll aus den 1980er-Jahren ist es jedoch schwieriger, als es scheint. DNS verfügt über eine Funktion namens CNAME, mit der der Besitzer einer Domain einen Teil ihrer Domain an einen anderen Provider zum Hosten auslagern kann. Sie funktioniert jedoch nicht auf der Stamm- oder obersten Ebene einer Domain. Um die Anforderungen unserer Kunden zu erfüllen, müssen wir die Domains unserer Kunden tatsächlich hosten. Wenn wir die Domain eines Kunden hosten, können wir alle aktuellen IP-Adressen für Amazon S3, Amazon CloudFront oder Elastic Load Balancing zurückgeben. Diese Services werden ständig erweitert und fügen IP-Adressen hinzu. Daher können Kunden ihre Domain-Konfigurationen auch nicht so einfach fest programmieren.

Das Hosten von DNS ist keine leichte Aufgabe. Wenn DNS Probleme hat, kann ein ganzes Unternehmen offline sein. Nachdem wir den Bedarf festgestellt hatten, machten wir uns daran, ihn auf die für Amazon typische Weise zu lösen – und zwar dringend. Wir haben ein kleines Team von Ingenieuren zusammengestellt und uns an die Arbeit gemacht.

Umgang mit DDOS-Angriffen

Fragen Sie jeden DNS-Anbieter nach der größten Herausforderung und erfahren Sie, dass er mit DDOS-Angriffen (Distributed Denial of Service) umgeht. DNS baut auf dem UDP-Protokoll auf, was bedeutet, dass DNS-Anforderungen in weiten Teilen des Wildwest-Internets fälschbar sind. Da DNS auch eine kritische Infrastruktur ist, ist diese Kombination ein attraktives Ziel für skrupellose Akteure, die versuchen, Unternehmen zu erpressen, für „Booter“, die aus verschiedenen Gründen Ausfälle auslösen möchten, und für den gelegentlich fehlgeleiteten Störer, der dies nicht zu bemerken scheint Sie begehen ein schweres Verbrechen mit echten persönlichen Konsequenzen. Aus welchem Grund auch immer, jeden Tag werden Tausende von DDOS-Angriffen auf Domänen ausgeführt.
Ein Ansatz zur Abwehr dieser Angriffe besteht darin, große Mengen an Serverkapazität zu verwenden. Obwohl es wichtig ist, eine gute Basiskapazität zu haben, lässt sich dieser Ansatz nicht wirklich skalieren. Jeder Server, den ein Anbieter hinzufügt, kostet Tausende von Dollar, aber Angreifer können mehr gefälschte Clients für ein paar Cent hinzufügen, wenn sie kompromittierte Botnets verwenden. Für Anbieter ist das Hinzufügen großer Serverkapazitäten ein Verlust.

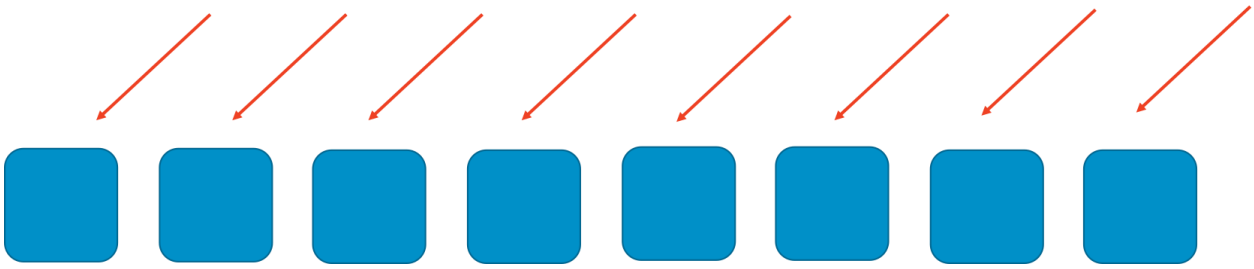
Zu der Zeit, als wir Amazon Route 53 bauten, war der Stand der Technik für die DNS-Verteidigung eine spezialisierte Netzwerk-Appliance, die mit einer Vielzahl von Tricks den

Datenverkehr mit einer sehr hohen Rate „bereinigen“ konnte. Wir hatten viele dieser Geräte bei Amazon für unsere vorhandenen internen DNS-Dienste und sprachen mit Hardwareanbietern darüber, was noch verfügbar war. Wir haben herausgefunden, dass der Kauf von Appliances, die für die vollständige Abdeckung jeder einzelnen Route 53-Domain ausreichen, mehrere zehn Millionen US-Dollar kosten und unseren Zeitplan um Monate verlängern würde, um sie auszuliefern, zu installieren und betriebsbereit zu machen. Das passte nicht zur Dringlichkeit unserer Pläne oder zu unseren Bemühungen, sparsam zu sein, und deshalb haben wir sie nie ernsthaft in Betracht gezogen. Wir mussten einen Weg finden, um nur Ressourcen für die Verteidigung von Domänen auszugeben, die tatsächlich angegriffen werden. Wir haben uns dem alten Prinzip zugewandt, dass die Notwendigkeit die Mutter der Erfindung ist. Unsere Notwendigkeit bestand darin, schnell einen erstklassigen DNS-Dienst mit 100 Prozent Verfügbarkeit und geringem Ressourceneinsatz aufzubauen. Unsere Erfindung war Shuffle Sharding.

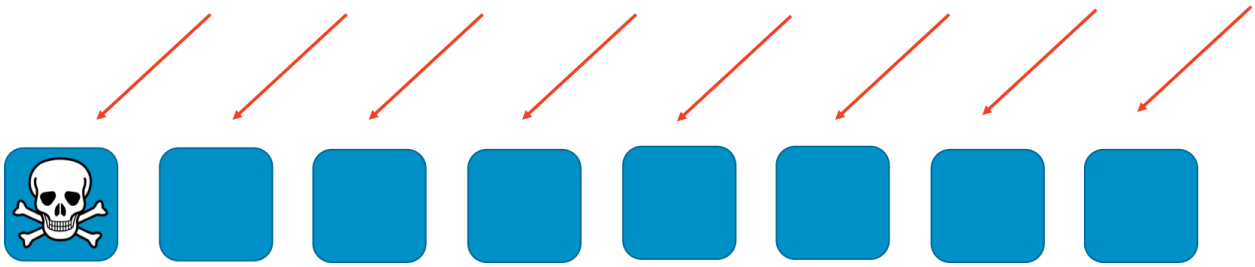
Was ist Shuffle Sharding?

Shuffle Sharding ist einfach, aber leistungsstark. Es ist sogar noch leistungsstärker als wir zuerst gemerkt haben. Wir haben es immer wieder verwendet und es wird zu einem Kernmuster, das es AWS ermöglicht, kostengünstige mandantenfähige Services bereitzustellen, die jedem Kunden ein Mandantenerlebnis bieten.

Um zu sehen, wie Shuffle-Sharding funktioniert, müssen Sie zunächst überlegen, wie ein System durch gewöhnliches Sharding skalierbarer und stabiler gemacht werden kann. Stellen Sie sich ein horizontal skalierbares System oder einen horizontal skalierbaren Dienst vor, der aus acht Mitarbeitern besteht. Das folgende Bild zeigt die Mitarbeiter und ihre Anforderungen. Die Mitarbeiter können Server, Warteschlangen oder Datenbanken sein, unabhängig davon, was Ihr System ausmacht.



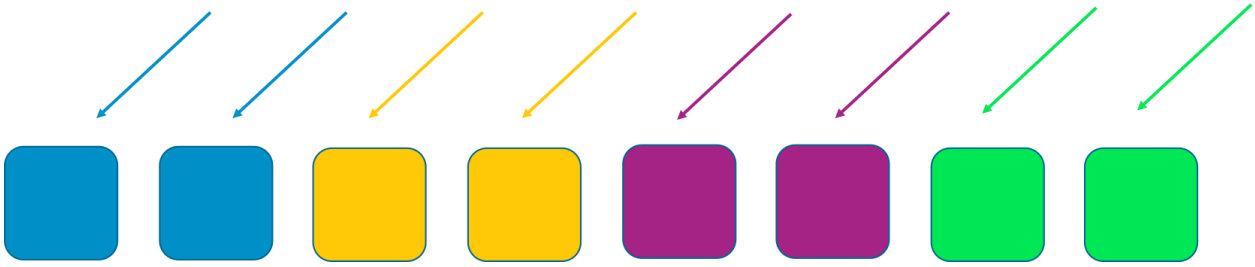
Ohne Sharding erledigt die Arbeiterflotte die ganze Arbeit. Jeder Mitarbeiter muss in der Lage sein, jede Anfrage zu bearbeiten. Dies ist für die Effizienz und Redundanz von Vorteil. Wenn ein Arbeiter ausfällt, können die anderen sieben die Arbeit aufnehmen, so dass im System relativ wenig Durchhangkapazität benötigt wird. Ein großes Problem tritt jedoch auf, wenn Fehler durch eine bestimmte Art von Anforderung oder durch eine Flut von Anforderungen ausgelöst werden können, z. B. durch einen DDOS-Angriff. Die folgenden beiden Bilder zeigen den Verlauf eines solchen Angriffs.



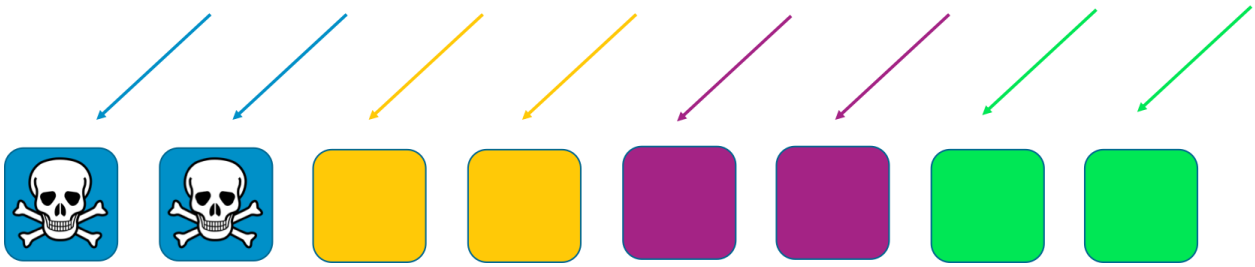
Das Problem wird den ersten betroffenen Arbeiter ausschalten, dann aber durch die anderen Arbeiter kaskadieren, während die verbleibenden Arbeiter übernehmen. Das Problem kann sehr schnell alle Mitarbeiter und den gesamten Service ausschalten.

Der Umfang der Auswirkungen für diese Art von Fehler ist „alles und jeder“. Der ganze Service fällt aus. Jeder Kunde ist betroffen. Wie wir im Availability Engineering sagen: Es ist nicht optimal.

Mit regelmäßigem Sharding können wir es besser machen. Wenn wir die Flotte in vier Arbeiter-Shards aufteilen, können wir die Effizienz gegen den Wirkungsbereich tauschen. Die folgenden zwei Bilder zeigen, wie Splitter die Auswirkungen eines DDOS-Angriffs begrenzen können.



In diesem Beispiel hat jeder Shard zwei Arbeiter. Wir teilen Ressourcen wie z. B. Kundendomänen auf die Shards auf. Wir haben immer noch Redundanz, aber da es nur zwei Arbeiter pro Shard gibt, müssen wir mehr Kapazitätsreserven im System halten, um eventuelle Ausfälle zu beheben. Im Gegenzug wird der Wirkungsbereich erheblich reduziert.

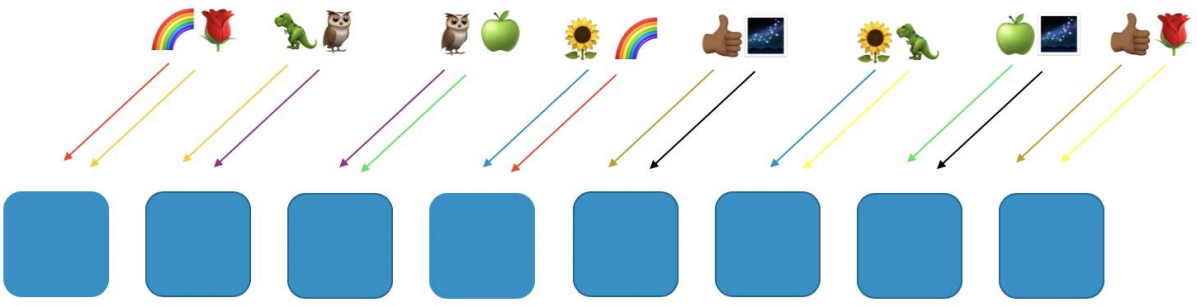


In dieser Shard-Welt verringert sich der Wirkungsbereich um die Anzahl der Shards. Wenn bei vier Shards ein Problem bei einem Kunden auftritt, sind möglicherweise der Host-Shard und alle anderen Kunden auf diesem Shard betroffen. Dieser Shard macht jedoch nur ein Viertel des gesamten Service aus. Ein Aufprall von 25 Prozent ist viel besser als ein Aufprall von 100 Prozent. Mit Shuffle-Sharding können wir es wieder exponentiell besser machen.

Mit Shuffle-Sharding erstellen wir virtuelle Shards mit jeweils zwei Arbeitern und weisen unsere Kunden oder Ressourcen oder was auch immer wir isolieren möchten, einem dieser virtuellen Shards zu.

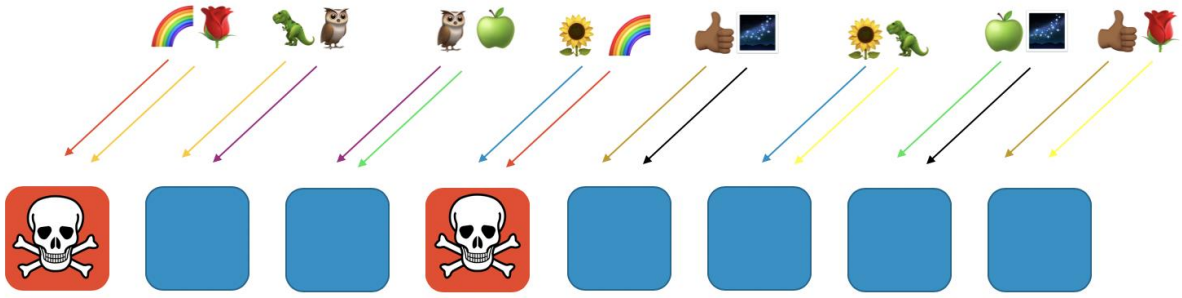
Das folgende Bild zeigt ein Beispiel für ein Sharding-Layout mit acht Mitarbeitern und acht Kunden, denen jeweils zwei Mitarbeiter zugewiesen sind. Normalerweise haben wir viel mehr Kunden als Arbeiter, aber es ist einfacher, mitzumachen, wenn wir die Dinge kleiner halten. Wir konzentrieren uns auf zwei Kunden - den Rainbow-Kunden und den Rose-Kunden.

In unserem Beispiel weisen wir den Rainbow-Kunden dem ersten und dem vierten Mitarbeiter zu. Die Kombination dieser beiden Arbeiter ergibt den Shuffle-Shard des Kunden. Andere Kunden werden verschiedene virtuelle Shards mit ihrem eigenen Mix aus zwei Arbeitern besuchen. Zum Beispiel wird der Rosenkunde auch dem ersten Arbeiter zugewiesen, sein anderer Arbeiter ist jedoch der achte Arbeiter.



Wenn der den Arbeitern eins und vier zugewiesene Regenbogenkunde ein Problem hat (z. B. eine giftige Anfrage oder eine Flut von Anfragen), wirkt sich dieses Problem auf diesen virtuellen Shard aus, aber auf keinen anderen Shard. Tatsächlich ist höchstens einer der Arbeiter eines anderen Shards betroffen. Wenn die Anforderer fehlertolerant sind und dies

umgehen können (z. B. mit Wiederholungsversuchen), kann der Dienst für die Kunden oder Ressourcen auf den verbleibenden Shards ohne Unterbrechung fortgesetzt werden, wie in der folgenden Abbildung dargestellt.



Anders ausgedrückt: Während alle Arbeiter, die Regenbogen bedienen, möglicherweise ein Problem oder einen Angriff haben, sind die anderen Arbeiter überhaupt nicht betroffen. Für Kunden bedeutet dies, dass der Rosenkunde und der Sonnenblumenkunde zwar jeweils einen Arbeiter mit dem Regenbogen teilen, jedoch nicht betroffen sind. Der Rosenkunde kann von Arbeiter 8 bedient werden, und die Sonnenblume kann von Arbeiter 6 bedient werden, wie in der folgenden Abbildung dargestellt.

Wenn ein Problem auftritt, können wir immer noch ein Viertel des gesamten Service verlieren. Die Art und Weise, wie Kunden oder Ressourcen zugewiesen werden, bedeutet jedoch, dass das Ausmaß der Auswirkungen von Shuffle-Sharding erheblich besser ist. Bei acht Arbeitern gibt es 28 eindeutige Kombinationen von zwei Arbeitern, was bedeutet, dass es 28 mögliche Shuffle-Shards gibt. Wenn wir Hunderte oder mehr Kunden haben und wir jedem Kunden eine Shuffle-Shard zuweisen, beträgt der Umfang der Auswirkungen aufgrund eines Problems nur 1/28. Das ist siebenmal besser als normales Sharding.

Es ist sehr aufregend zu sehen, dass die Zahlen exponentiell besser werden, je mehr Mitarbeiter und Kunden Sie haben. Die meisten Skalierungsherausforderungen werden in diesen Dimensionen schwieriger, aber Shuffle-Sharding wird effektiver. In der Tat kann es bei genügend Arbeitnehmern mehr Shuffle-Shards geben als Kunden, und jeder Kunde kann isoliert werden.

Amazon Route 53 und Shuffle-Sharding

Wie hilft all dies Amazon Route 53? Mit Route 53 haben wir beschlossen, unsere Kapazität auf insgesamt 2048 virtuelle Nameserver zu verteilen. Diese Server sind virtuell, da sie nicht den physischen Servern entsprechen, auf denen Route 53 gehostet wird. Wir können sie bewegen, um die Kapazitäten zu verwalten. Anschließend ordnen wir jede Kundendomäne einem Shuffle aus vier virtuellen Nameservern zu. Mit diesen Zahlen gibt es unglaubliche 730 Milliarden mögliche Shuffle-Shards. Wir haben so viele mögliche Shuffle-Shards, dass wir jeder Domain einen eigenen Shuffle-Shard zuweisen können. Wir können sogar noch weiter gehen und sicherstellen, dass keine Kundendomäne jemals mehr als zwei virtuelle Nameserver mit einer anderen Kundendomäne gemeinsam nutzt.

Die Ergebnisse sind großartig. Wenn eine Kundendomäne für einen DDOS-Angriff vorgesehen ist, nehmen die dieser Domäne zugewiesenen vier virtuellen Nameserver im Datenverkehr zu, aber keine andere Kundendomäne bemerkt dies. Wir sind nicht damit zufrieden, dass der Zielkunde einen schlechten Tag hat. Shuffle-Sharding bedeutet, dass wir den anvisierten Kunden identifizieren und für spezielle Angriffskapazitäten isolieren können. Darüber hinaus haben wir eine eigene Schicht von AWS Shield-Datenverkehrsscrubbern entwickelt. Shuffle-Sharding trägt jedoch entscheidend dazu bei, dass das Kundenerlebnis auf der Route 53 auch während dieser Ereignisse reibungslos verläuft.

Fazit

Wir haben Shuffle-Sharding in viele unserer anderen Systeme integriert. Darüber hinaus haben wir uns Verfeinerungen ausgedacht, wie z. B. das rekursive Shuffle-Sharding, bei dem wir Objekte auf mehreren Ebenen sharden und so den Kunden eines Kunden isolieren. Shuffle Sharding ist enorm anpassungsfähig. Es ist eine clevere Möglichkeit, vorhandene Ressourcen zu ordnen. Darüber hinaus entstehen in der Regel keine zusätzlichen Kosten. Dies ist eine enorme Verbesserung, die durch Sparsamkeit und Sparsamkeit erreicht werden kann.

Wenn Sie Shuffle-Sharding selbst verwenden möchten, besuchen Sie unsere Open-Source-Bibliothek Route 53 Infima. Diese Bibliothek enthält verschiedene Implementierungen von Shuffle-Sharding, die zum Zuweisen oder Anordnen von Ressourcen verwendet werden können.