Isolement de la charge de travail à l'aide du partitionnement aléatoire

Colm MacCárthaigh

Isolement de la charge de travail à l'aide du partitionnement aléatoire

Aujourd'hui, Amazon Route 53 héberge un très nombre des plus grandes entreprises et sites Web les plus populaires au monde, mais ses débuts sont bien plus modestes.

Prise en charge de l'hébergement DNS

Peu de temps après qu'AWS a commencé à offrir des services, les clients AWS ont clairement indiqué qu'ils souhaitaient pouvoir utiliser nos services Amazon Simple Storage Service (S3), Amazon CloudFront et Elastic Load Balancing à la « racine » de leur domaine, c'est-à-dire des noms tels que « amazon.com » et pas seulement pour des noms comme « www.amazon.com ».

Cela peut sembler très simple. Cependant, en raison d'une décision de conception dans le protocole DNS prise dans les années 80, c'est plus difficile qu'il ne paraît. Le protocole DNS possède une fonctionnalité appelée CNAME qui permet au propriétaire d'un domaine de transférer une partie de son domaine vers un autre fournisseur, mais elle ne fonctionne pas au niveau racine ou supérieur d'un domaine. Pour répondre aux besoins de nos clients, nous devions héberger les domaines de nos clients. Lorsque nous hébergeons le domaine d'un client, nous pouvons renvoyer l'ensemble des adresses IP actuelles pour Amazon S3, Amazon CloudFront ou Elastic Load Balancing. Ces services ne cessent de s'étendre et d'ajouter des adresses IP. Par conséquent, ce n'est pas quelque chose que les clients pourraient facilement coder en dur dans leurs configurations de domaine non plus.

Ce n'est pas une mince affaire que d'héberger DNS. Si DNS rencontre des problèmes, une entreprise entière peut être hors ligne. Cependant, après avoir identifié le besoin, nous avons décidé de le résoudre de la manière habituelle chez Amazon : en urgence. Nous avons formé une petite équipe d'ingénieurs et nous nous sommes mis au travail.

Gestion des attaques par déni de service distribué (DDOS)

Demandez à tout fournisseur DNS quel est son plus gros défi et il vous dira que c'est la gestion des attaques par déni de service distribué. DNS repose sur le protocole UDP, ce qui signifie que les requêtes DNS peuvent être usurpées sur une grande partie de l'« Internet sauvage ». Comme DNS est aussi une infrastructure critique, cette combinaison en fait une cible attrayante pour des acteurs peu scrupuleux qui tentent d'extorquer des entreprises, des « booters » qui cherchent à provoquer des pannes pour diverses raisons, ainsi que des créateurs de nuisances parfois mal avisés qui ne semblent pas s'en rendre compte. Ils commettent un crime grave avec des conséquences personnelles réelles. Quelle que soit la raison, chaque jour, des milliers d'attaques DDOS sont commises contre des domaines.

Une approche pour réduire ces attaques consiste à utiliser d'énormes volumes de capacité serveur. Bien qu'il soit important de disposer d'une bonne base de référence de capacité, cette approche n'est pas vraiment adaptée. Chaque serveur ajouté par un fournisseur coûte des milliers d'euros, mais les attaquants peuvent ajouter plus de faux clients pour quelques

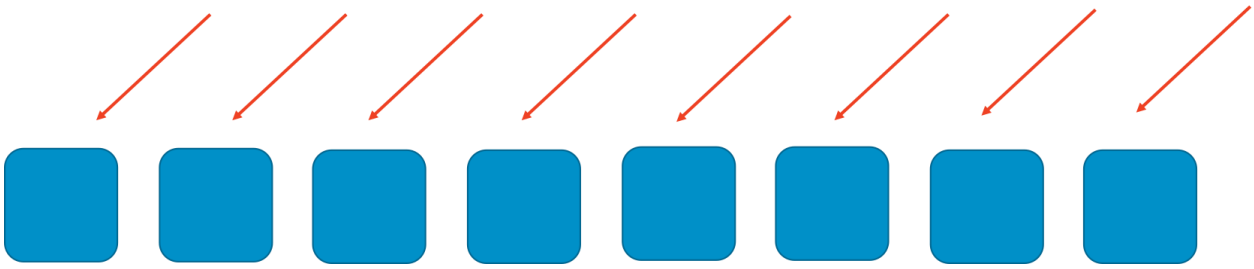
centimes s'ils utilisent des réseaux de zombies compromis. Pour les fournisseurs, ajouter d'énormes volumes de capacité serveur est une stratégie perdante.

Au moment où nous avons créé Amazon Route 53, le nec-plus-ultra de la défense DNS consistait en des dispositifs réseau spécialisés qui pouvaient utiliser quantité d'astuces pour « nettoyer » le trafic très rapidement. Bon nombre de ces dispositifs chez Amazon étaient destinés à nos services DNS internes existants et nous avons discuté avec les fournisseurs de matériels de ce qui était disponible. Nous avons réalisé que l'achat d'un nombre suffisant de dispositifs pour couvrir entièrement chaque domaine de Route 53 coûterait des dizaines de millions d'euros et allongerait notre planning de plusieurs mois pour les livrer, les installer et les faire fonctionner. Cela ne correspondait pas à l'urgence de nos plans ni à nos efforts de frugalité, et nous ne l'avons donc jamais envisagé sérieusement. Nous devions trouver un moyen de ne consacrer que des ressources à la défense des domaines confrontés à une attaque. Nous nous sommes tournés vers le vieux principe selon lequel la nécessité est la mère de l'invention. Notre devions créer rapidement un service DNS de classe mondiale, 100 % disponible, en utilisant un faible nombre de ressources. Nous avons donc inventé le partitionnement aléatoire.

Qu'est-ce que le partitionnement aléatoire ?

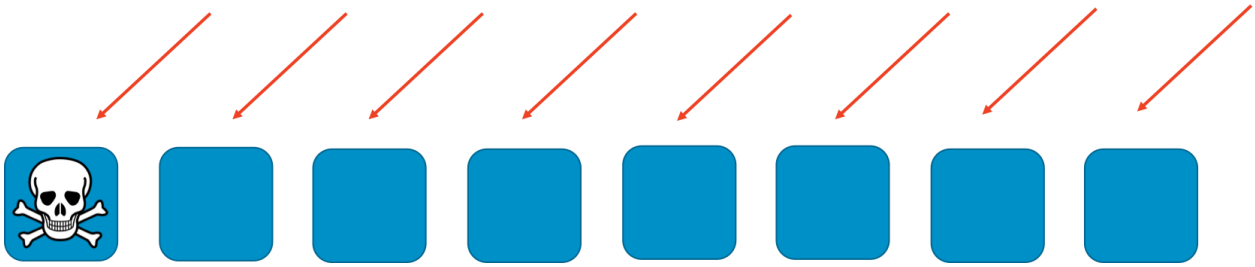
Le partitionnement aléatoire est quelque chose de simple mais puissant. C'est même plus puissant qu'on ne le pensait. Nous l'avons utilisé maintes et maintes fois, et c'est devenu un modèle essentiel qui permet à AWS de fournir des services rentables multi-locataires offrant à chaque client une expérience avec un locataire unique.

Pour voir comment fonctionne le partitionnement aléatoire, vous devez d'abord examiner comment rendre un système plus évolutif et résistant par le biais du partitionnement ordinaire. Imaginez un système ou un service évolutif horizontalement composé de huit agents. L'image suivante montre les agents et leurs demandes. Les agents peuvent être des serveurs, des files d'attente ou des bases de données, quelle que soit la « chose » qui constitue votre système.



Sans aucun partitionnement, la flotte d'agents gère tout le travail. Chaque agent doit pouvoir traiter n'importe quelle demande. C'est excellent pour l'efficacité et la redondance. En cas de défaillance d'un agent, les sept autres peuvent absorber le travail, de sorte que le système a besoin d'une marge de capacité relativement faible. Cependant, un gros problème se pose si des défaillances peuvent être déclenchées par un type particulier de demande ou par un flot

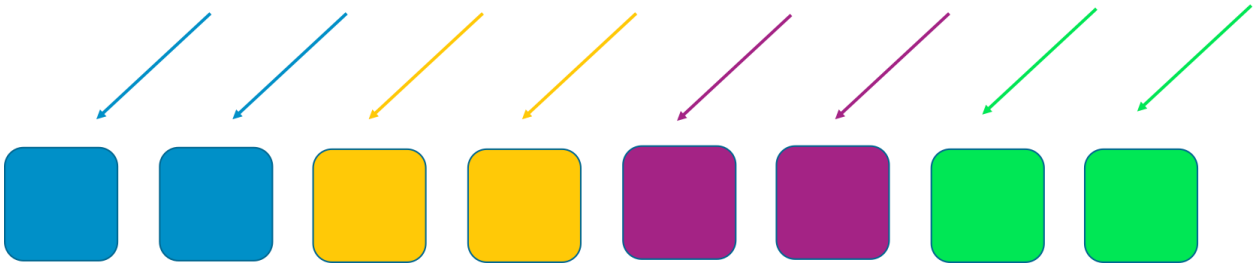
de demandes, tel qu'une attaque DDOS. Les deux images suivantes montrent la progression d'une telle attaque.



Le problème éliminera le premier agent impacté, mais se répercutera en cascade au fur et à mesure que les agents restants prendront la relève. Le problème peut très rapidement éliminer tous les agents et l'ensemble du service.

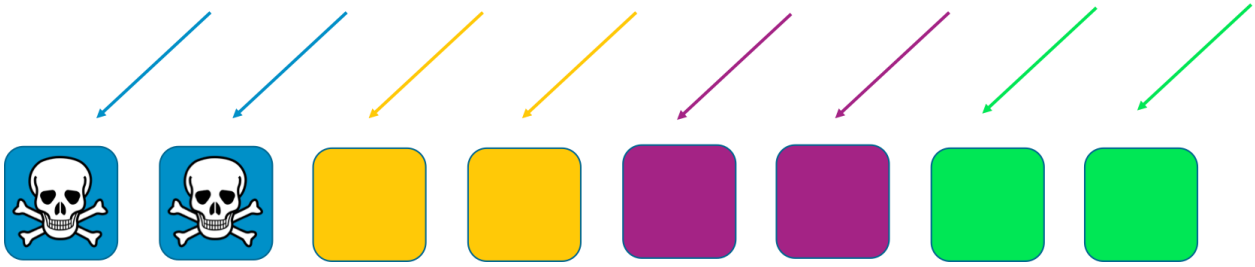
L'ampleur de l'impact de ce type d'échec est « tout et tout le monde ». Tout le service tombe en panne. Tous les clients sont affectés. Comme on dit en ingénierie de disponibilité : ce n'est pas optimal.

Avec le partitionnement normal, nous pouvons être plus performants. Si nous divisons la flotte en quatre partitions d'agents, nous pouvons remplacer l'efficacité par l'ampleur de l'impact. Les deux images suivantes montrent comment le partitionnement peut limiter l'impact d'une attaque DDOS.



Dans cet exemple, chaque partitionnement dispose de deux agents. Nous répartissions les ressources, telles que les domaines des clients, entre les partitions. Nous avons toujours la

redondance, mais comme il n'existe que deux agents par partition, nous devons garder une plus grande marge de capacité dans le système pour gérer les défaillances. En conséquence, l'ampleur de l'impact est considérablement réduite.

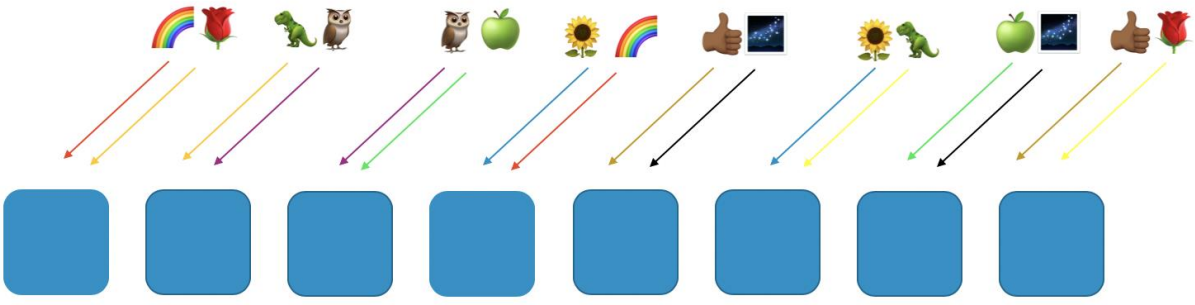


Dans cet environnement de partitionnement, l'ampleur de l'impact est réduite par le nombre de partitions. Ici, avec quatre partitions, si un client rencontre un problème, la partition qui les héberge peut être impactée, ainsi que tous les autres clients de cette dernière. Cependant, ce partitionnement ne représente qu'un quart du service total. Un impact de 25 % est beaucoup mieux qu'un impact de 100 %. Avec le partitionnement aléatoire, nous pouvons faire excessivement mieux.

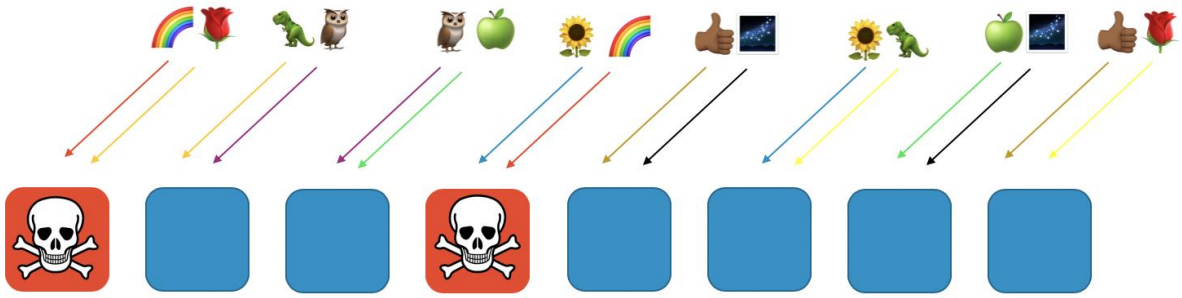
Avec le partitionnement aléatoire, nous créons des partitions virtuelles de deux agents chacune, et nous affectons nos clients ou nos ressources, ou ce que nous voulons isoler, à l'une de ces partitions virtuelles.

L'image suivante montre un exemple de partitionnement aléatoire avec huit agents et huit clients, chacun étant affecté à deux agents. Normalement, nous devrions disposer de beaucoup plus de clients que d'agents, mais il est plus facile de suivre à une échelle plus petite. Nous allons nous concentrer sur deux clients : le client Arc-en-ciel et le client Rose.

Dans notre exemple, nous attribuons le client Arc-en-ciel au premier et au quatrième agents. La combinaison de ces deux agents constitue la partition aléatoire de ce client. Les autres clients utiliseront différentes partitions virtuelles, avec leur propre combinaison d'agents. Par exemple, le client Rose est également affecté au premier agent, mais son autre agent est le huitième.



Si le client Arc-en-ciel affecté aux agents 1 et 4 connaît un problème (comme une demande dangereuse ou une saturation de demandes), ce problème aura une incidence sur cette partition virtuelle, mais n'affectera pas pleinement les autres partitions aléatoires. En fait, l'un des agents d'une autre partition aléatoire sera affecté, tout au plus. Si les demandeurs sont tolérants aux pannes et peuvent contourner ce problème (avec des tentatives, par exemple), le service peut continuer sans interruption pour les clients ou les ressources des partitions restantes, comme illustré ci-dessous.



En d'autres termes, si tous les agents de Rainbow connaissent un problème ou subissent une attaque, les autres agents ne sont pas du tout touchés. Pour les clients, cela signifie que même si le client Rose et le client Tournesol partagent chacun un agent avec Arc-en-ciel, ils ne sont pas impactés. Le client Rose peut obtenir le service de l'agent 8, et le client Tournesol, de l'agent 6, comme le montre l'image suivante.

En cas de problème, nous pouvons encore perdre un quart de l'ensemble du service, mais la manière dont les clients ou les ressources sont attribués signifie que l'ampleur de l'impact avec le partitionnement aléatoire est bien meilleure. Avec huit agents, il existe 28 combinaisons uniques de deux agents, ce qui signifie qu'il existe 28 partitions aléatoires possibles. Si nous avons des centaines de clients, voire plus, et que nous affectons chaque client à une partition aléatoire, l'ampleur de l'impact d'un problème n'est que de $1/28^{e}$. C'est 7 fois moins qu'avec le partitionnement normal.

C'est très excitant de voir que les chiffres sont exponentiellement meilleurs avec l'augmentation du nombre d'agents et de clients. La plupart des problèmes de dimensionnement deviennent plus complexes dans ces dimensions, mais le partitionnement

aléatoire devient plus efficace. En fait, avec un nombre suffisant d'agents, il peut exister plus de partitions aléatoires que de clients, et chaque client peut être isolé.

Amazon Route 53 et le partitionnement aléatoire

Comment tout cela aide-t-il Amazon Route 53 ? Avec Route 53, nous avons décidé d'organiser notre capacité en un total de 2 048 serveurs de noms virtuels. Ces serveurs sont virtuels, car ils ne correspondent pas aux serveurs physiques hébergeant Route 53. Nous pouvons les déplacer pour gérer la capacité. Nous affectons ensuite chaque domaine de client à une partition aléatoire de quatre serveurs de noms virtuels. Avec ces valeurs, nous arrivons à un nombre faramineux de 730 milliards de partitions aléatoires possibles. Nous disposons d'un si grand nombre de partitions aléatoires que nous pouvons attribuer une partition unique à chaque domaine. En fait, nous pouvons aller plus loin et veiller à ce qu'aucun domaine client ne partage jamais plus de deux serveurs de noms virtuels avec un autre domaine de client.

Le résultat est étonnant. Si un domaine client est ciblé par une attaque DDOS, le trafic des quatre serveurs de noms virtuels qui lui sont affectés augmentera, mais les domaines des autres clients ne le remarqueront pas. Nous ne nous résignons pas à ce que le client ciblé passe une mauvaise journée. Le partitionnement aléatoire signifie que nous pouvons identifier et isoler le client ciblé dans une capacité d'attaque spéciale. De plus, nous avons également développé notre propre couche exclusive de nettoyeurs de trafic AWS Shield. Toutefois, le partitionnement aléatoire fait toute la différence pour garantir la continuité de l'expérience client de Route 53, même lorsque ces événements se produisent.

Conclusion

Nous avons continué à intégrer le partitionnement aléatoire dans beaucoup de nos autres systèmes. En outre, nous apportons des améliorations, telles que le partitionnement aléatoire récursif, dans lequel nous partitionnons des éléments à plusieurs niveaux, isolant ainsi le client du client. Le partitionnement aléatoire est extrêmement adaptable. C'est un moyen judicieux d'organiser les ressources existantes. De plus, comme il n'entraîne généralement aucun coût supplémentaire, c'est donc une grande amélioration que la frugalité et l'économie peuvent apporter.
Si vous souhaitez utiliser le partitionnement aléatoire vous-même, consultez notre bibliothèque open source Route 53 Infima. Cette bibliothèque comprend plusieurs implémentations différentes du partitionnement aléatoire qui peuvent être utilisées pour attribuer ou organiser des ressources.