Isolamento de carga de trabalho usando ordem aleatória de fragmentação

Colm MacCárthaigh

Isolamento de carga de trabalho usando ordem aleatória de fragmentação

Hoje, o Amazon Route 53 hospeda muitas das maiores empresas e sites mais populares do mundo, mas seu início foi muito mais humilde.

Assumindo a hospedagem de DNS

Pouco tempo depois que a AWS começou a oferecer serviços, os clientes da AWS deixaram claro que desejavam poder usar nossos serviços Amazon Simple Storage Service (S3), Amazon CloudFront e Elastic Load Balancing na "raiz" do domínio deles, ou seja, para nomes como "amazon.com" e não apenas para nomes como "www.amazon.com".

Isso pode parecer muito simples. No entanto, devido a uma decisão de design no protocolo DNS, tomada na década de 1980, é mais difícil do que parece. O DNS tem um recurso chamado CNAME que permite que o proprietário de um domínio transfira uma parte desse domínio para outro provedor hospedar, mas não funciona no nível raiz ou superior de um domínio. Para atender às necessidades de nossos clientes, teríamos que hospedar os domínios de nossos clientes na verdade. Quando hospedamos o domínio de um cliente, podemos retornar qualquer conjunto atual de endereços IP para o Amazon S3, Amazon CloudFront ou Elastic Load Balancing. Esses serviços estão constantemente expandindo e adicionando endereços IP, portanto, também não é algo que os clientes possam codificar com facilidade nas configurações de domínio.

Hospedar o DNS não é uma tarefa pequena. Se o DNS estiver com problemas, uma empresa inteira poderá ficar offline. No entanto, depois que identificamos a necessidade, decidimos resolvê-la da maneira típica da Amazon: com urgência. Criamos uma pequena equipe de engenheiros e começamos a trabalhar.

Processando ataques DDOS

Pergunte a qualquer provedor de DNS qual é o maior desafio e eles lhe dirão que está lidando com ataques de negação de serviço distribuída (DDOS). O DNS é criado com base no protocolo UDP, o que significa que as solicitações de DNS são falsificadas em grande parte da Internet do oeste selvagem. Como o DNS também é uma infraestrutura crítica, essa combinação o torna um alvo atraente para atores inescrupulosos que tentam extorquir negócios, "inicializadores" que pretendem desencadear interrupções por várias razões e o ocasional causador de problemas equivocado que parece não perceber que está cometendo um crime grave com consequências pessoais reais. Independentemente do motivo, todos os dias há milhares de ataques DDOS cometidos contra domínios.

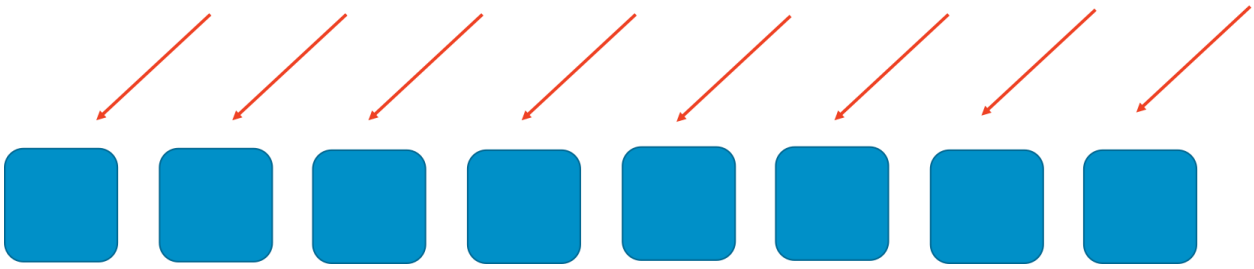
Uma abordagem para atenuar esses ataques é usar grandes volumes de capacidade do servidor. Embora seja importante ter uma boa linha de base de capacidade, essa abordagem não é realmente escalável. Todo servidor adicionado por um provedor custa milhares de dólares, mas os invasores podem adicionar mais clientes falsos por centavos se estiverem usando botnets comprometidos. Para os provedores, adicionar grandes volumes de capacidade do servidor é uma estratégia perdida.

No momento em que criamos o Amazon Route 53, o estado da arte em defesa do DNS eram dispositivos de rede especializados que podiam usar uma variedade de truques para "eliminar" o tráfego a uma taxa muito alta. Tínhamos muitos desses dispositivos na Amazon para nossos serviços DNS internos existentes e conversamos com fornecedores de hardware sobre o que mais estava disponível. Descobrimos que comprar equipamentos suficientes para cobrir totalmente todos os domínios do Route 53 custaria dezenas de milhões de dólares e incluiria meses em nossa programação para que eles fossem entregues, instalados e ficassem operacionais. Isso não se encaixava com a urgência de nossos planos ou com nossos esforços para sermos frugais, por isso nunca avaliamos seriamente a possibilidade de usá-los. Precisávamos encontrar uma maneira de gastar recursos apenas na defesa de domínios que realmente estão enfrentando um ataque. Voltamos ao antigo princípio de que a necessidade é a mãe da invenção. Nossa necessidade era criar rapidamente um serviço DNS de classe mundial com 100% de tempo de atividade, usando uma quantidade modesta de recursos. Nossa invenção foi a fragmentação aleatória.

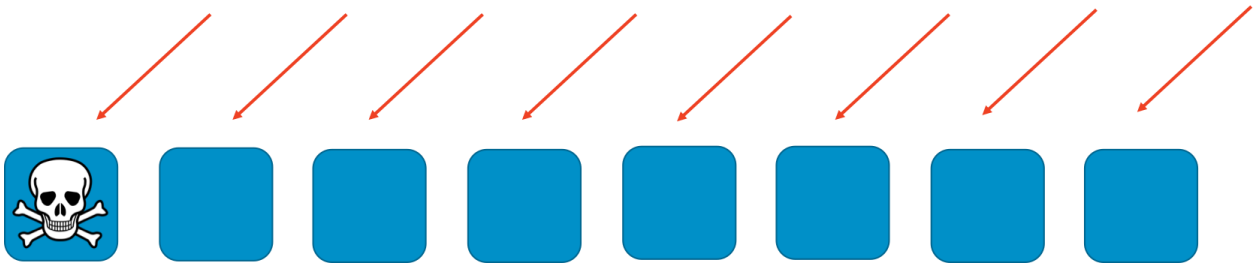
O que é fragmentação aleatória?

A fragmentação aleatória é simples, mas poderosa. É ainda mais poderosa do que imaginávamos a princípio. Nós a usamos repetidamente e tornou-se um padrão essencial que possibilita à AWS fornecer serviços de vários inquilinos econômicos que proporcionam a cada cliente uma experiência de inquilino único.

Para ver como a fragmentação aleatória funciona, primeiro considere como um sistema pode se tornar mais escalável e resiliente por meio da fragmentação comum. Imagine um sistema ou serviço escalável composto horizontalmente por oito funcionários. A imagem a seguir ilustra os trabalhadores e suas solicitações. Os trabalhadores podem ser servidores, filas ou bancos de dados, qualquer que "coisa" que componha seu sistema.



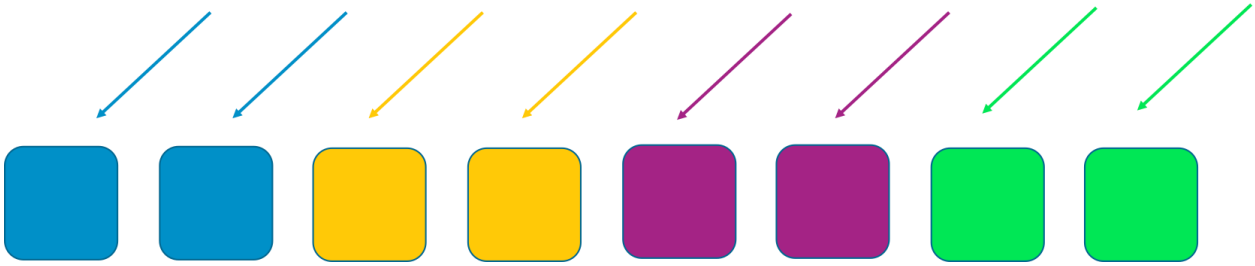
Sem nenhuma fragmentação, a frota de trabalhadores processa todo o trabalho. Cada trabalhador deve ser capaz de processar qualquer solicitação. Isso é ótimo para eficiência e redundância. Se um trabalhador falhar, os outros sete poderão absorver o trabalho, portanto, é necessária uma capacidade relativamente reduzida de folga no sistema. No entanto, um grande problema surge se as falhas puderem ser acionadas por um tipo específico de solicitação ou por uma enxurrada de solicitações, como um ataque DDOS. As duas imagens a seguir mostram a progressão desse ataque.



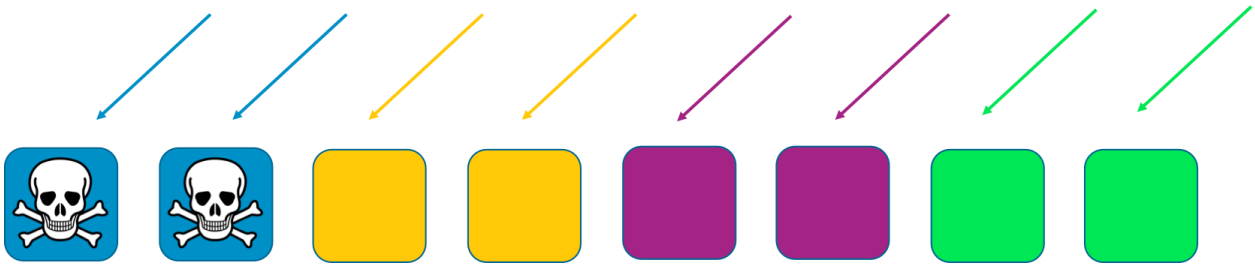
O problema eliminará o primeiro trabalhador afetado, mas continuará em cascata pelos outros trabalhadores à medida que os demais trabalhadores assumirem o controle. O problema pode eliminar rapidamente todos os trabalhadores e todo o serviço.

O escopo do impacto para esse tipo de falha é "tudo e todos". Todo o serviço cai. Todo cliente é afetado. Como dizemos na engenharia de disponibilidade: não é o ideal.

Com a fragmentação regular, podemos fazer melhor. Se dividirmos a frota em quatro fragmentos de trabalhadores, poderemos trocar a eficiência pelo escopo do impacto. As duas imagens a seguir mostram como a fragmentação pode limitar o impacto de um ataque DDOS.



Neste exemplo, cada fragmentação tem dois trabalhadores. Dividimos recursos, como domínios de clientes, entre os fragmentos. Ainda temos redundância, mas como existem apenas dois trabalhadores por fragmentos, precisamos manter mais capacidade de folga no sistema para lidar com qualquer falha. Em troca, o escopo do impacto é reduzido consideravelmente.

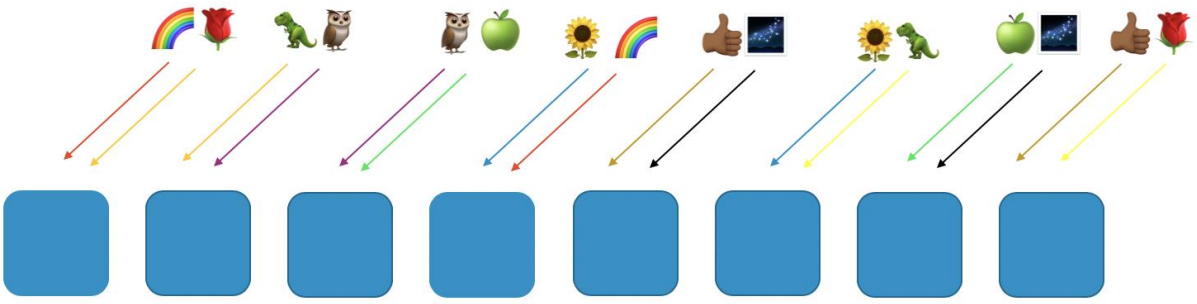


Nesse mundo de fragmentação, o escopo do impacto é reduzido pelo número de fragmentos. Aqui, com quatro fragmentos, se um cliente tiver um problema, o fragmento que os hospeda poderá ser impactado, assim como todos os outros nesse fragmento. No entanto, esse fragmento representa apenas um quarto do serviço geral. Um impacto de 25% é muito melhor do que um impacto de 100%. Com a fragmentação aleatória, podemos melhorar exponencialmente de novo.

Com a fragmentação aleatória, criamos fragmentos virtuais de dois trabalhadores cada um e atribuímos nossos clientes ou recursos, ou o que quisermos isolar, a um desses fragmentos virtuais.

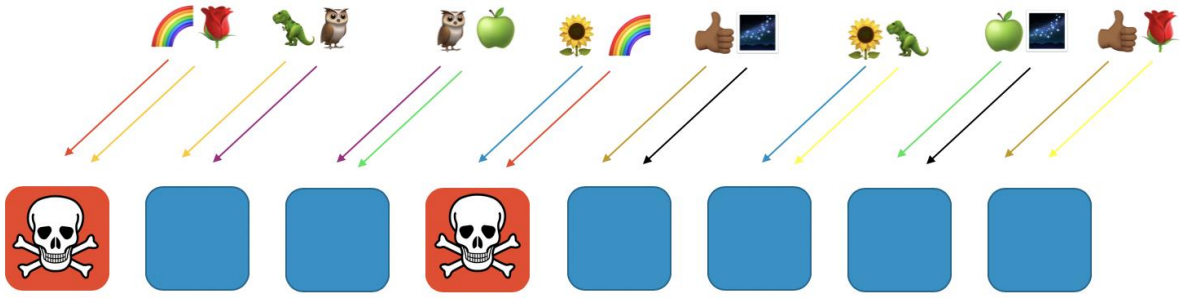
A imagem a seguir mostra um exemplo de layout de fragmentação aleatória com oito trabalhadores e oito clientes, cada um atribuído a dois trabalhadores. Normalmente, teríamos muito mais clientes que trabalhadores, mas é mais fácil acompanhar se mantivermos as coisas menores. Vamos nos concentrar em dois clientes: o cliente arco-íris e o cliente rosa.

Em nosso exemplo, atribuímos o cliente arco-íris ao primeiro e ao quarto trabalhador. A combinação desses dois funcionários compõe o fragmento aleatório desse cliente. Outros clientes irão para diferentes fragmentos virtuais, com sua própria mistura de dois trabalhadores. Por exemplo, o cliente rosa também é atribuído ao primeiro trabalhador, mas seu outro trabalhador é o oitavo trabalhador.



Se o cliente arco-íris atribuído aos trabalhadores um e quatro tiver um problema (como uma solicitação venenosa ou uma enxurrada de solicitações), esse problema afetará esse fragmento virtual, mas não afetará totalmente nenhum outro fragmento aleatório. De fato, no máximo um dos trabalhadores de outra fragmentação aleatória será afetado. Se os solicitantes forem

tolerantes a falhas e puderem solucionar isso (com novas tentativas, por exemplo), o serviço poderá continuar ininterrupto para os clientes ou recursos nos fragmentos restantes, como mostra a imagem a seguir.



Dito de outra forma, enquanto todos os trabalhadores que servem o arco-íris podem estar enfrentando um problema ou um ataque, os outros trabalhadores não são afetados. Para os clientes, isso significa que, embora o cliente rosa e o cliente girassol compartilhem um trabalhador com o arco-íris, eles não são afetados. O cliente rosa pode obter serviço do trabalhador oito e o girassol pode obter serviço do trabalhador seis, como pode ser visto na imagem a seguir.

Quando ocorre um problema, ainda podemos perder um quarto de todo o serviço, mas a maneira como os clientes ou os recursos são atribuídos significa que o escopo do impacto com a fragmentação aleatória é consideravelmente melhor. Com oito trabalhadores, há 28 combinações únicas de dois trabalhadores, o que significa que há 28 possíveis fragmentos aleatórios. Se temos centenas de clientes ou mais e atribuímos cada cliente a um fragmento aleatório, o escopo do impacto devido a um problema é de apenas $1/28^{o}$. Isso é sete vezes melhor que a fragmentação regular.

É muito emocionante ver que os números ficam exponencialmente melhores quanto mais trabalhadores e clientes você tiver. A maioria dos desafios de escalabilidade fica mais difícil nessas dimensões, mas a fragmentação aleatória fica mais eficaz. De fato, com trabalhadores suficientes, pode haver mais fragmentos aleatórios do que clientes, e cada cliente pode ser isolado.

Amazon Route 53 e fragmentação aleatória

Como tudo isso ajuda o Amazon Route 53? Com o Route 53, decidimos organizar nossa capacidade em um total de 2048 servidores de nomes virtuais. Esses servidores são virtuais porque não correspondem aos servidores físicos que hospedam o Route 53. Podemos movê-los para ajudar a gerenciar a capacidade. Em seguida, atribuímos todos os domínios do cliente a um fragmento aleatório de quatro servidores de nomes virtuais. Com esses números, existem impressionantes 730 bilhões de fragmentos possíveis. Temos tantos fragmentos aleatórios possíveis que podemos atribuir um fragmento aleatório exclusivo a cada domínio. Na verdade, podemos ir além e garantir que nenhum domínio do cliente compartilhe mais de dois servidores de nome virtual com qualquer outro domínio do cliente.

Os resultados são surpreendentes. Se um domínio do cliente for alvo de um ataque DDOS, os quatro servidores de nomes virtuais atribuídos a esse domínio terão um pico no tráfego, mas nenhum domínio de outro cliente notará. Não renunciamos ao cliente alvo que está tendo um dia ruim. A fragmentação aleatória significa que podemos identificar e isolar o cliente alvo para uma capacidade de ataque dedicada especial. Além disso, também desenvolvemos nossa própria camada proprietária de lavadores de tráfego da AWS Shield. No entanto, a fragmentação aleatória faz uma enorme diferença ao garantir que a experiência geral do cliente do Route 53 seja perfeita, mesmo enquanto esses eventos estão acontecendo.

Conclusão

Passamos a incorporar a fragmentação aleatória em muitos de nossos outros sistemas. Além disso, criamos refinamentos, como fragmentação aleatória recursiva, em que fragmentamos itens em várias camadas, isolando assim o cliente de um cliente. A fragmentação aleatória é extremamente adaptável. É uma maneira inteligente de organizar os recursos existentes. Também geralmente vem sem custo adicional, por isso é uma grande melhoria que a frugalidade e a economia podem gerar.

Se você estiver interessado em usar a fragmentação aleatória, confira nossa biblioteca do Route 53 Infima de código aberto. Esta biblioteca inclui várias implementações diferentes de fragmentação aleatória que podem ser usadas para atribuir ou organizar recursos.