

Computação Distribuída e Serverless Computing

Grupo 2: Artur Jalles, Bianca Veloso, Matheus Melo, Heitor Campos e

Samuel Xavier

Link do Slide:

https://www.canva.com/design/DAGgS8wVVxg/eLOmRsb5Fu_xvFGIPyShow/edit?utm_content=DAGgS8wVVxg&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

O que é computação sem servidor?

A computação sem servidor é um modelo de execução de computação em nuvem que aloca recursos de máquina conforme usados. Em um modelo sem servidor, os desenvolvedores podem criar e executar aplicativos sem gerenciar servidores e pagar apenas pela quantidade exata de recursos usados. Em vez disso, o provedor de serviços de nuvem é responsável por provisionar, gerenciar e escalonar a infraestrutura de nuvem que executa o código do aplicativo.

O nome pode ser enganoso, mas [sem servidor](#) não significa "sem servidores". Em vez disso, os apps sem servidor abstraem o trabalho de infraestrutura de rotina associado ao desenvolvimento de aplicativos. Você não tem visibilidade sobre as máquinas que executam seus aplicativos, não pode configurá-los e não precisa gerenciá-los ou escaloná-los. Em outras palavras, você paga pelo serviço de um servidor, e não pelo servidor em si.

Do ponto de vista do desenvolvimento, é como se não houvesse servidores. Os desenvolvedores escrevem o código, o implantam na produção e o provedor de nuvem cuida do resto.

Como funciona a computação sem servidor?

Quando se trata de criar aplicativos, os desenvolvedores normalmente têm uma série de outras tarefas de gerenciamento de servidor que precisam executar para implantar o código, como definir recursos para processamento e armazenamento, aplicar patches, carregar balanceamento e muito mais. Todos esses itens extras levam a tempos de provisionamento mais longos e mais sobrecarga operacional que, no fim, desacelera as equipes de desenvolvimento.

A computação sem servidor tenta fornecer uma experiência "sem servidor" invisível para os desenvolvedores, eliminando a necessidade de pensar nos servidores ou em qualquer coisa que um aplicativo possa precisar executar. Em vez disso, o provedor de serviços faz todo o trabalho nos bastidores para garantir que você tenha os recursos para executar seu código e atender aos requisitos sem ser cobrado por capacidade ociosa.

Veja uma analogia simples para entender como a computação sem servidor funciona

Pense na computação sem servidor como levar água para sua casa. Você pode cavar o próprio poço, testar a qualidade da água, instalar e manter todo o encanamento externo necessário para executá-lo na casa. Ou você pode se conectar ao principal sistema de abastecimento da sua cidade. Basta ligar a torneira de água e pegar a quantidade de água que quiser exatamente quando precisar, e a cidade enviará uma fatura mensal com a quantidade exata de água que você consome.

Da mesma forma, os provedores de nuvem ativam automaticamente os recursos de infraestrutura e os ambientes de execução necessários para executar seus aplicativos sem servidor e reduzem automaticamente o escalonamento a zero quando a execução é concluída. Normalmente, você é cobrado com base nos recursos

durante o tempo de execução. No entanto, isso pode variar de acordo com o serviço e o fornecedor.

Computação sem servidor x outros tipos de computação em nuvem

Mas não é o caso de outros tipos de computação em nuvem não fornecem recursos sob demanda com pagamento por utilização? É verdade que modelos de serviço na nuvem como [Infraestrutura as a Service \(IaaS\)](#) e [Plataforma como serviço \(PaaS\)](#) também oferecem acesso sob demanda a recursos de computação.

No entanto, a infraestrutura em nuvem alocada para você permanece ativa mesmo que não seja usada. Também é sua responsabilidade escalonar a capacidade do servidor ou configurar seus aplicativos para aumentar ou diminuir a escala em resposta à demanda do usuário.

Por outro lado, as arquiteturas sem servidor são orientadas a eventos. Nesse modelo, o provedor só fornece recursos quando um evento aciona a execução do seu código e faz o escalonamento automático e instantâneo das solicitações.

Exemplos de uso e casos de uso de computação sem servidor

Na computação em nuvem sem servidor, geralmente há dois tipos de serviços: funções como serviço (FaaS) e back-end como serviço (BaaS).

O FaaS fornece os recursos de computação necessários para executar a lógica do aplicativo em resposta às solicitações. Esses elementos de lógica (ou funções) são executados em contêineres totalmente gerenciados pelo provedor de serviços de nuvem. Os aplicativos sem servidor normalmente são divididos em funções únicas que realizam uma ação em resposta a um evento.

O BaaS oferece toda a funcionalidade de back-end de aplicativos da Web ou móveis como serviço, como serviços de autenticação, gerenciamento de banco de dados, armazenamento em nuvem, notificações push e hospedagem. O BaaS também elimina a necessidade de gerenciar servidores, contêineres ou máquinas virtuais.

No desenvolvimento, o termo "sem servidor" é geralmente usado para descrever ofertas FaaS. No entanto, é importante observar que, tecnicamente, o FaaS é um subconjunto da tecnologia sem servidor. A computação sem servidor inclui qualquer tipo de serviço em que o gerenciamento, a configuração, o escalonamento e o faturamento do servidor são abstraídos do usuário final. Isso pode incluir bancos de dados, armazenamento, streaming de eventos, mensagens e gateways de API.

Em geral, a computação sem servidor ainda é usada principalmente para tarefas simples e de curta duração. Alguns dos exemplos de computação sem servidor mais comuns incluem:

- Como integrar com serviços e APIs de terceiros
- Execução de tarefas programadas (por exemplo, relatórios diários, backups ou lógica de negócios)
- Automação do processo de TI, como remover o acesso automaticamente, iniciar verificações de segurança de conformidade ou enviar aprovações
- Processamento de dados em tempo real para dados estruturados e não estruturados
- Como automatizar pipelines de integração e entrega contínuas (CI/CD)
- Back-ends da API REST
- Ações com base em acionadores ou execução de tarefas agendadas (por exemplo, relatórios diários, backups ou lógica de negócios)

- Processamento assíncrono
- Notificações em tempo real ou programadas

Vantagens da computação sem servidor

Melhor produtividade dos desenvolvedores

Os desenvolvedores sem servidor se concentram em escrever códigos e otimizar a lógica de negócios em vez de se preocupar com provisionamento, gerenciamento ou escalonamento de infraestrutura.

Escalonabilidade imediata

Arquiteturas sem servidor são escalonadas verticalmente ou verticalmente por natureza. Ao contrário do escalonamento automático em outros tipos de computação em nuvem, que requer regras de ajuste, o escalonamento automático sem servidor é instantâneo e pode ser escalonado para zero.

Preços flexí

Os modelos de preços sem servidor geralmente são cobrados com base no uso real e no tempo necessário para executar uma função. Você também não paga pela capacidade inativa, o que ajuda a reduzir os custos operacionais associados à execução e ao gerenciamento dos seus próprios servidores.

Implantação mais rápida

Com a computação sem servidor, as equipes de DevOps não têm tempo para definir a infraestrutura necessária para integração, teste, entrega ou implantação de código. Eles simplesmente escrevem e implantam na produção.

Use qualquer idioma

Os ambientes sem servidor são compatíveis com qualquer linguagem ou framework, o que permite que as equipes desenvolvam nessa linguagem ou com o framework (Go, Python, Java, Node.js, .NET e muito mais) que se sentem mais à vontade.

Desvantagens da computação sem servidor

Uma das maiores desvantagens da computação sem servidor é que ela ainda é uma tecnologia relativamente nova. Por isso, ele ainda não é adequado para atender a todos os casos de uso em potencial.

Além disso, a natureza intencionalmente temporária do recurso sem servidor e a capacidade dele de reduzir a zero o tornam inadequado

para determinados tipos de aplicativos. Ele não foi criado para executar códigos por longos períodos e nem sempre pode atender a aplicativos com requisitos rigorosos de baixa latência, como aplicativos de serviços financeiros.

Por último, mas não menos importante, as vantagens de passar o gerenciamento de todos os recursos de computação para um provedor de terceiros também podem ter várias desvantagens. Embora ele libere tempo para se concentrar no código, também pode ser complicado migrar para um novo provedor de nuvem. Você também não tem visibilidade de onde ou como seu serviço é executado e tem controle limitado sobre como você escala, qual tipo de código de hardware é executado e situações de recuperação de desastres.

Ainda há muito a gostar quando se trata de computação sem servidor, especialmente para organizações que buscam ajudar os desenvolvedores a avançar mais rapidamente e criar uma maneira mais simples e econômica de criar nuvem baseados em aplicativos.

Também está claro que algumas das atuais fraquezas serão resolvidas com o tempo, à medida que a computação sem servidor continua a evoluir. Muito trabalho já está sendo feito em implementações de código aberto de sem servidor e maneiras de tornar os [serviços em nuvem mais abertos](#) para tornar as soluções portáteis em diferentes plataformas e ambientes.

Tipos de Serverless Computing

Function as a Service (FaaS)

Esse modelo permite que os desenvolvedores executem funções individuais em resposta a eventos, sem precisar gerenciar servidores ou infraestrutura. Cada função é executada de forma independente e escala automaticamente conforme a demanda.

Backend as a Service (BaaS)

Oferece um backend completo gerenciado, permitindo que os desenvolvedores foquem apenas no frontend e na lógica do aplicativo. Os serviços incluem autenticação, banco de dados, armazenamento, notificações, entre outros.

Tipos de computação distribuída

Sistemas de Cluster

Os sistemas de cluster são conjuntos de computadores interconectados que operam como um único sistema, sendo amplamente utilizados para cálculos científicos e processamento de grandes volumes de dados. Eles oferecem **alta disponibilidade e escalabilidade**, garantindo que o sistema continue funcionando mesmo em caso de falhas em alguns nós.

Cada **nó** pode ter hardware distinto, mas todos compartilham o mesmo sistema operacional para garantir compatibilidade. A conexão ocorre por redes de alta velocidade, como Ethernet, permitindo inclusão e remoção de nós sem interrupções. O **nó líder** gerencia as tarefas e distribui a carga de processamento.

Tipos:

- Cluster de Alta Disponibilidade
- Cluster para Balanceamento de Carga
- Cluster de Alto Desempenho
- Cluster de Processamento paralelo

Computação em Grade

A computação em grade é um modelo de computação distribuída que utiliza uma rede de computadores interconectados para processar grandes volumes de dados e realizar cálculos complexos. Diferente da supercomputação, que usa máquinas individuais de alto desempenho, a computação em grade aproveita o poder combinado de vários dispositivos distribuídos.

Tipos:

- Grade computacional

- Grade de limpeza
- Grade de dados

Computação em Nuvem

A computação em nuvem é um modelo que permite o fornecimento de serviços de computação — como servidores, armazenamento, bancos de dados, redes, software, análise e inteligência — pela internet. Esse modelo visa facilitar a inovação mais rápida, oferecer flexibilidade nos recursos e garantir economias de escala. Os usuários pagam apenas pelos serviços que utilizam, o que ajuda a reduzir custos operacionais e possibilita maior eficiência na gestão da infraestrutura.

Tipos:

- nuvem pública
- nuvem privada
- nuvem híbrida

Tipos de serviços:

- infraestrutura como serviço (IaaS)
- plataforma como serviço (PaaS)
- software como serviço (SaaS)
- computação sem servidor

Computação em Borda (Edge Computing)

A computação em borda é um paradigma que processa dados próximo à fonte onde são gerados, ao invés de em data centers distantes. Esse modelo visa reduzir a latência, garantindo uma resposta quase em tempo real.

É crucial para o aproveitamento de redes 5G, permitindo uma experiência mais eficiente e rápida com computação próxima aos usuários.

Setores que usam:

- Manufatura
- Veículos autônomos
- Energia

- Saúde