THIAGO ADRIANO

SOFTWARE ARCHITECTURE

ARQUITETURA DE SOFTWARE

AULA 04

SUMÁRIO

O QUE VEM POR AÍ?

Nesta aula você vai aprender alguns pontos que vão ajudar na escalabilidade e resiliência da sua aplicação.

HANDS ON

No primeiro vídeo desta aula, os professores apresentaram algumas formas de monitoramento da sua aplicação, como a utilização do health check e como ele pode ajudar a monitorar a saúde da sua aplicação, como, por exemplo, se ela está conectada ao banco de dados.

Também comentamos sobre o Zabbix e como ele pode ajudar neste monitoramento com o disparo de notificações.

Na outra aula em vídeo, os docentes apresentaram uma ferramenta de monitoramento e disponibilidade de sistemas, como o Prometheus e o Grafana.

Eles implementaram as bibliotecas necessárias para monitorar o projeto que estão desenvolvendo e, utilizando o Docker, criaram containers para demonstrar o exemplo real de monitoramento. Vale lembrar que temos uma disciplina completa sobre dockers e containers! Estude-a para compreender mais sobre o assunto!

No vídeo final desta aula, os docentes demonstraram como escalar uma aplicação utilizando um serviço em cloud chamado Azure Container Apps, um PaaS para Kubernetes.

SAIBA MAIS

ESCALABILIDADE, DISPONIBILIDADE E DESEMPENHO

Escalabilidade, disponibilidade e desempenho são três conceitos importantes relacionados ao desenvolvimento e operação de sistemas de software.

A seguir, vamos detalhar cada um deles para que você possa firmar o seu conhecimento nesses pilares.

ESCALABILIDADE

Escalabilidade de software se refere à capacidade de um sistema de software de crescer e se adaptar para lidar com um aumento na demanda. Isso significa que um software escalável deve ser capaz de lidar com um grande volume de usuários, dados, e tráfego de rede sem prejudicar seu desempenho ou confiabilidade.

Existem várias estratégias que podem ser usadas para tornar um software escalável, como:

- Arquitetura escalável: a arquitetura do software deve ser projetada para suportar o aumento de demanda sem afetar o desempenho. Isso pode incluir o uso de sistemas distribuídos e balanceamento de carga.
- Uso de tecnologias modernas: tecnologias modernas, como computação em nuvem e contêineres, podem ajudar a aumentar a escalabilidade de um software, permitindo que ele seja facilmente dimensionado e gerenciado.
- Banco de dados escalável: um banco de dados escalável é capaz de lidar com grandes quantidades de dados e usuários simultaneamente. Isso pode incluir a utilização de bancos de dados NoSQL e tecnologias de cache.
- Monitoramento e otimização: monitorar o desempenho do software e fazer otimizações frequentes é fundamental para garantir a escalabilidade a longo prazo.

Falando sobre monitoramento e otimização, existem várias ferramentas disponíveis para esta finalidade. São algumas:

- New Relic: é uma plataforma de monitoramento e análise de desempenho que oferece visibilidade em tempo real sobre o desempenho do aplicativo, infraestrutura e usuários finais.
- Nagios: é um sistema de monitoramento de código aberto que monitora serviços de rede, servidores e dispositivos de rede em tempo real. Ele também oferece alertas quando ocorrem problemas.
- Splunk: é uma plataforma de análise e inteligência de dados que permite monitorar e analisar grandes quantidades de dados de diferentes fontes. Ele pode ser usado para monitorar logs de aplicativos, infraestrutura e segurança.
- AppDynamics: é uma plataforma de monitoramento e análise de desempenho que ajuda a otimizar o desempenho de aplicativos empresariais. Ele monitora o desempenho em tempo real e identifica problemas que afetam a experiência do usuário.
- VisualVM: é uma ferramenta de monitoramento de desempenho de código aberto para aplicativos Java que fornece informações detalhadas sobre a utilização de memória, threads, classes e outros recursos do sistema.

Essas são apenas algumas das muitas ferramentas disponíveis para monitoramento e otimização de software.

DISPONIBILIDADE E OBSERVABILIDADE

Disponibilidade e observabilidade são duas métricas importantes no contexto de sistemas de computação e tecnologia da informação.

Disponibilidade refere-se à capacidade de um sistema ou serviço estar disponível para uso em um determinado momento. Em outras palavras, a disponibilidade mede se um sistema está funcionando corretamente e se pode ser acessado pelos usuários quando necessário.

A disponibilidade é geralmente medida em termos de tempo de atividade, ou seja, o tempo em que um sistema está disponível em relação ao tempo total.

Observabilidade, por outro lado, refere-se à capacidade de observar e medir o comportamento de um sistema ou serviço em tempo real. A observabilidade permite monitorar e rastrear o desempenho e o estado de um sistema, bem como identificar e solucionar problemas rapidamente.

A observabilidade é geralmente medida em termos de métricas de desempenho, como tempo de resposta, taxa de erros, utilização de recursos e outras métricas relevantes.

Em geral, disponibilidade e observabilidade são complementares e trabalham juntas para garantir que um sistema esteja funcionando de maneira confiável e eficiente.

Um sistema altamente disponível pode não ser muito útil se não puder ser monitorado e observado para detectar e solucionar problemas, enquanto um sistema altamente observável pode não ser muito útil se não estiver disponível quando necessário. Por isso, é importante que as equipes de tecnologia da informação monitorem tanto a disponibilidade quanto a observabilidade de seus sistemas e serviços.

DESEMPENHO

O desempenho é a medida de quão bem um sistema ou componente executa uma tarefa em relação às expectativas ou requisitos estabelecidos. Em outras palavras, o desempenho é a capacidade de um sistema ou componente de realizar suas funções de maneira eficiente e eficaz.

No contexto de sistemas de computação e tecnologia da informação, o desempenho pode ser medido em várias dimensões, incluindo:

- Tempo de resposta: o tempo que leva para um sistema ou componente responder à uma solicitação do usuário.
- Taxa de transferência: a quantidade de dados que podem ser transferidos de um sistema ou componente em um determinado período.
- Utilização de recursos: a quantidade de recursos do sistema, como processamento, memória e armazenamento, usada para executar uma tarefa ou conjunto de tarefas.

- Confiabilidade: a capacidade de um sistema ou componente de funcionar sem falhas ou interrupções por longos períodos de tempo.
- Escalabilidade: a capacidade de um sistema ou componente de lidar com um aumento no número de usuários, dados ou solicitações sem perda de desempenho.

O desempenho é uma preocupação crítica em muitos setores da tecnologia, incluindo serviços de internet, jogos eletrônicos, computação de alto desempenho e muitos outros.

As pessoas desenvolvedoras e engenheiros(as) de sistemas estão constantemente procurando maneiras de melhorar o desempenho dos sistemas existentes e criar sistemas que possam lidar com as crescentes demandas dos usuários e dos negócios.

Uma delas é a estratégia de cache que é usada para melhorar o desempenho de um aplicativo, armazenando temporariamente dados que são frequentemente acessados em memória cache para evitar buscar esses dados novamente do local original, como um banco de dados ou sistema de arquivos. Isso reduz o tempo necessário para buscar e processar esses dados, melhorando a velocidade de resposta do aplicativo.

Algumas estratégias comuns de cache incluem:

- Cache de página inteira: nesta estratégia as páginas do aplicativo são armazenadas inteiramente em cache, incluindo HTML, CSS, JS e outros recursos. Isso pode melhorar significativamente o tempo de carregamento da página para os usuários.
- Cache de banco de dados: dados frequentemente acessados em um banco de dados, como informações de perfil do usuário, podem ser armazenados em cache para reduzir a quantidade de solicitações de banco de dados necessárias.
- Cache de objeto: nesta estratégia, objetos complexos ou recursos frequentemente usados são armazenados em cache. Isso pode incluir arquivos de imagem, objetos JSON ou outras informações.

- Cache de sessão: informações específicas da sessão do usuário, como preferências, histórico de navegação ou dados do carrinho de compras, podem ser armazenadas em cache para melhorar o desempenho do aplicativo.
- Cache de CDN: uma rede de entrega de conteúdo (CDN) pode ser usada para armazenar arquivos de mídia em cache, como imagens e vídeos, em servidores distribuídos em todo o mundo, para que os usuários possam acessá-los a partir de um servidor mais próximo, reduzindo o tempo de resposta e melhorando o desempenho.

Veja a figura 1 – “Exemplo de fluxo de cache” que demonstra a estratégia de cache:

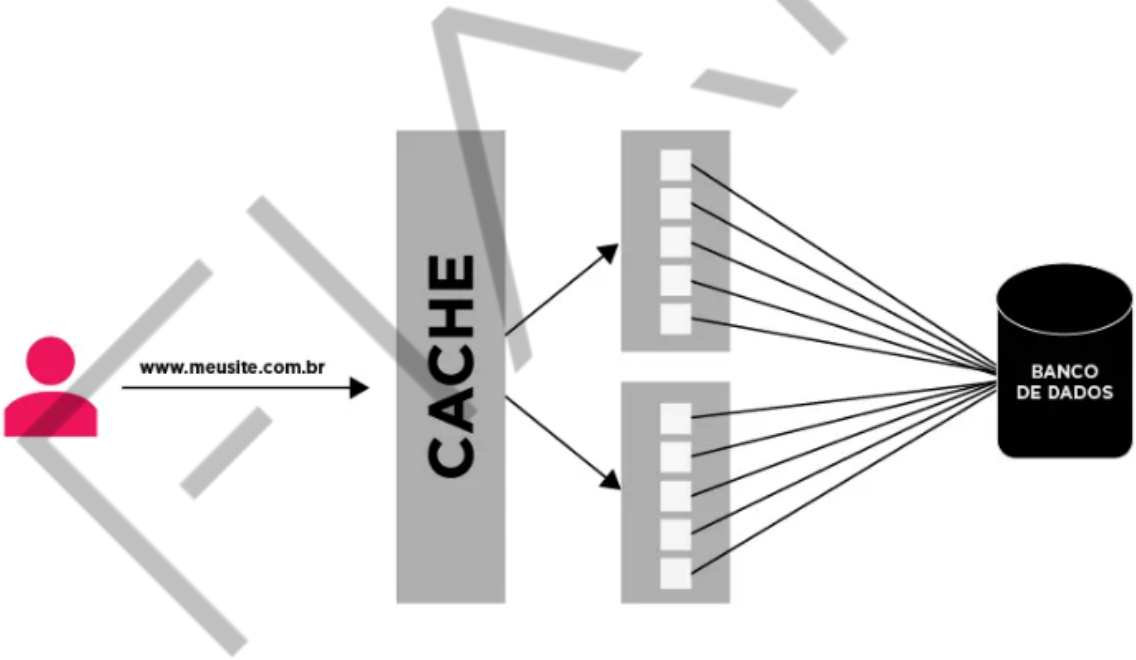

Figura 1 – Exemplo de fluxo de cache
Fonte: elaborada pelo autor (2023)

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, você aprendeu sobre a importância de monitorar o desempenho da nossa aplicação, o que é um sistema escalável, e o que é disponibilidade e observabilidade.

Não perca as oportunidades de interagir com os docentes e colegas em nossa comunidade do Discord. Venha tirar dúvidas, trocar ideias, assistir às lives e muito mais. Estamos te esperando!

REFERÊNCIAS

FOWLER, S. J. **Microsserviços Prontos Para a Produção: Construindo Sistemas Padronizados em uma Organização de Engenharia de Software**. São Paulo: Novatec Editora, 2017.

MAJORS, C.; FONG-JONES, L.; MIRANDA, G. **Observability Engineering: Achieving Production Excellence**. Califórnia: O'Reilly Media, 2022.

PALAVRAS-CHAVE

Observabilidade. Escalabilidade. Desempenho.