

Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation

Análise de Desempenho e Escalabilidade

Jefferson Uchoa Ponte

Baseado no artigo de: Grzegorz Blinowski, Anna Ojdowska, Adam Przybyłek

Universidade Estadual do Ceará (UECE)

Engenharia de Software

Prof. Matheus Paixão

Junho de 2024

- Pioneiros: Netflix começou a transição de uma arquitetura monolítica para microserviços em 2009.
- 2011 oficialmente cunhado e anunciado na 33rd Degree Conference in Kraków
- Popularização: Ganhou tração em 2014 com a publicação de Lewis e Fowler e os relatos de sucesso da Netflix.
- Adesão de Empresas Globais: Amazon, eBay, Spotify, Uber, Airbnb, LinkedIn, entre outras, adotaram microserviços para lidar com o aumento da complexidade e demanda.

- Desafios dos Microserviços:
 - Identificação de limites de microserviços ideais.
 - Orquestração de serviços complexos.
 - Manutenção da consistência de dados e gestão de transações.
 - Aumento do consumo de recursos computacionais.
- Benefícios dos Microserviços:
 - Melhor escalabilidade horizontal.
 - Facilidade de manutenção e desenvolvimento independente.
 - Maior tolerância a falhas.

Motivação para o Estudo

- Pequenas Empresas: Avaliar se pequenas empresas realmente se beneficiam da migração para microserviços como as grandes empresas.
- Escalabilidade Vertical vs. Horizontal: Examinar a efetividade da escalabilidade vertical em comparação com a horizontal para sistemas de menor escala.
- Evidências Empíricas: Preencher a lacuna de conhecimento sobre os benefícios reais da migração para microserviços para sistemas com menor demanda.

Comparar a performance e escalabilidade das arquiteturas monolítica e de microserviços:

- Aplicação web de referência.
- Tecnologias: Java vs. C# .NET.
- Ambientes: Local, Azure Spring Cloud, Azure App Service.

- (RQ1) Qual a diferença de desempenho entre aplicações monolíticas e microserviços?
- (RQ2) Qual das duas abordagens de arquiteturas deverá ser escolhida para ter o melhor benefício em uma aplicação para ser escalada?
- (RQ3) Em quais circunstâncias a tecnologia de implementação usada (Java vs C# .NET) possui vantagens e desvantagens?

Arquitetura Monolítica

- Definição: Aplicações monolíticas são construídas como uma única unidade onde todas as funcionalidades são implementadas e implantadas juntas.
- Simplicidade: Mais fáceis de testar, implantar, depurar e monitorar.
- Desvantagens: À medida que a aplicação cresce, a complexidade aumenta, tornando mais difícil modificar e manter.
- Desempenho: Melhor em um ambiente de máquina única devido à comunicação intra-processo.

Arquitetura de Microserviços

- Definição: Decompõe o domínio de negócios em pequenos serviços autônomos e independentes que são implantáveis separadamente.
- Autonomia: Cada microserviço é independente, facilitando a manutenção e o desenvolvimento.
- Escalabilidade: Escala bem horizontalmente, permitindo adicionar mais instâncias conforme necessário.
- Desafios: Requer maior esforço em orquestração, manutenção de consistência de dados e gestão de transações.
- Desempenho: Pode ter overhead devido à comunicação entre processos (IPC), mas compensa em sistemas de alta demanda.

- Implementação de quatro versões da aplicação:
 - Monolítica em Java e C# .NET.
 - Microserviços em Java e C# .NET.
- Experimentos realizados em três ambientes de deployment:
 - Local
 - Azure Spring Cloud
 - Azure App Service
- Critérios de avaliação:
 - Performance em máquina única
 - Escalabilidade vertical (aumento de recursos na mesma máquina)
 - Escalabilidade horizontal (adição de instâncias)
 - Impacto da tecnologia (Java vs. C# .NET)

Configuração dos Experimentos

- Configuração das máquinas:
 - Local: PC com Microsoft Windows 10 Enterprise - Intel(R) Core(TM) i7-9850H CPU 2.60GHz, six physical, 12 logical cores, and 32 GB RAM (Dell Precision 7540).
 - Azure: Configurações variadas de instâncias (pequenas a grandes).
- Ferramentas de monitoramento e teste:
 - JMeter para simulação de carga.
 - Azure Monitor para métricas de desempenho.
- Cenários de teste:
 - Testes de carga com diferentes níveis de concorrência.
 - Medição de tempo de resposta, throughput e uso de recursos.

Resultados Principais

- Desempenho em Máquina Única: Monolíticas têm melhor desempenho.
- Java vs. .NET: Java melhor em máquinas poderosas, .NET melhor em máquinas menos potentes.
- Escalabilidade Vertical vs. Horizontal: Vertical é mais econômica no Azure.
- Limite de Instâncias: Aumento excessivo degrada a performance.
- Impacto da Tecnologia: Implementação (Java ou C# .NET) não afeta escalabilidade.

- Monolítica: Simplicidade, facilidade de teste, deploy, debug e monitoramento.
- Microserviços: Melhor para aplicações complexas e grandes, com desafios em comunicação e gestão de dados.

- Microserviços: Vantajosos para sistemas com alta demanda e complexidade.
- Monolíticas: Adequadas para pequenas empresas ou sistemas de menor escala.