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

Introdução

- **Pioneiros**: Netflix iniciou a transição de monolítica para microserviços em 2009.
- **Oficialmente cunhado**: Termo oficialmente cunhado em 2011, após um ano 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.

Desafios e Benefícios

Desafios dos Microserviços:

- Identificação de limites de microserviços.
- Orquestração de serviços complexos.
- Manutenção da consistência de dados.
- Aumento do consumo de recursos.

Benefícios dos Microserviços:

- Melhor escalabilidade horizontal.
- Facilidade de manutenção independente.
- Maior tolerância a falhas.

Motivação para o Estudo

- **Pequenas Empresas**: Avaliar benefícios reais da migração para microserviços.
- **Escalabilidade Vertical vs. Horizontal**: Efetividade para sistemas menores.
- **Evidências Empíricas**: Preencher a lacuna de conhecimento sobre os benefícios reais.

Objetivo do Estudo

- 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.

Perguntas da Pesquisa

- **(RQ1)** Qual a diferença de desempenho entre aplicações monolíticas e microserviços?
- **(RQ2)** Qual abordagem arquitetural oferece melhor escalabilidade?
- **(RQ3)** Em quais circunstâncias a tecnologia de implementação (Java vs C# .NET) tem vantagens e desvantagens?

Arquitetura Monolítica

- **Definição**: Aplicações construídas como uma única unidade.
- **Simplicidade**: Fáceis de testar, implantar, depurar e monitorar.
- **Desvantagens**: Complexidade crescente torna difícil modificar e manter.
- **Desempenho**: Melhor em máquina única devido à comunicação intra-processo.

Arquitetura de Microserviços

- **Definição**: Decompõe o domínio de negócios em pequenos serviços autônomos.
- **Autonomia**: Independência facilita manutenção e desenvolvimento.
- **Escalabilidade**: Escala bem horizontalmente.
- **Desafios**: Orquestração, consistência de dados e gestão de transações.
- **Desempenho**: Pode ter overhead devido à comunicação entre processos (IPC), mas compensa em sistemas de alta demanda.

Metodologia

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

Configuração dos Experimentos

- **Máquinas**:
  - Local: PC com Windows 10, Intel Core i7, 32 GB RAM.
  - Azure: Instâncias variadas (pequenas a grandes).
- **Ferramentas**:
  - 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.

Conclusão

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

Discussão

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