Uma Arquitetura de Microsserviços Orientada a Eventos para Integração de Dados Frequentemente Atualizados Oriundos de Bases de Dados Poliglotas

An Event-Oriented Microservice Architecture for the Integration of Frequently Updated Data from Polyglot Databases

Luís H. N. Villaça
Programa de Pós Graduação em Sistemas de Informação – PPGI
Universidade Federal do Estado do Rio de Janeiro (Unirio)
Petrobras Transporte SA
luis.villaca@uniriotec.br

Leonardo G. Azevedo
Programa de Pós Graduação em Sistemas de Informação – PPGI
Universidade Federal do Estado do Rio de Janeiro (Unirio)
IBM Research
azevedo@uniriotec.br, lga@br.ibm.com

RESUMO

Em uma arquitetura de microsserviços, soluções são criadas por equipes focadas em domínios específicos que desenvolvem independentemente serviços distribuídos na rede. Um desafio neste cenário é a integração de dados provenientes de fontes distintas, como bancos de dados NoSQL e relacionais, e mesmo dados não estruturados. Este trabalho propõe uma nova abordagem arquitetural, orientada a eventos, por meio de uma plataforma de *streaming*. Essa proposta será avaliada e comparada a outras abordagens orientadas a evento publicadas na literatura com base em critérios dos modelos ISO 25010 e ISO 25012.

Palavras-Chave

Microsserviços, Arquitetura orientada a eventos, Persistência Poliglota

ABSTRACT

In a microsservice architecture, solutions are created by teams focused on specific domains, which independently develop services distributed on the network. A challenge in this scenario is integrating data from distinct sources, such as NoSQL and relational databases, and even unstructured data. This work presents a new event-driven architectural approach to address this problem through a streaming platform. This proposal will be evaluated and compared to other event-oriented approaches available in the literature based on criteria from ISO 25010 and ISO 25012 models.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBSI 2018 June 4^{th} – 8^{th}, 2018, Caxias do Sul, Rio Grande do Sul, Brazil

Copyright SBC 2018.

CCS Concepts

•**Information systems** → **Distributed retrieval; Mediators and data integration;** •**Computer systems organization** → *Heterogeneous (hybrid) systems;*

Keywords

Microservices, Event-Driven Architecture, Polyglot Persistence

1. INTRODUÇÃO

A arquitetura de microsserviços apresenta como uma de suas características o "Gerenciamento de dados descentralizados" [6], no qual serviços têm autonomia para definir o modelo de persistência dos dados que ele manipula.

Este trabalho propõe uma arquitetura orientada a eventos destinada a resolver com maior eficiência desafios que surgem quando buscamos consolidar informações a partir de serviços que utilizam diferentes tecnologias de armazenamento de dados, especialmente num cenário em que informações são frequentemente atualizadas em determinadas fontes de dados (como sensores que coletam dados de telemetria). Neste caso, eventos de atualização ocorrem por meio de dados gerados continuamente, a uma taxa de incidência alta. Estas atualizações podem ser consideradas como *streams* - cuja natureza de processamento (*stream processing*) visa proporcionar a análise e tratamento, com baixa latência, de um grande volume de dados [12].

A solução proposta nesta dissertação parte do princípio de que, mantendo-se uma visão consolidada a cada ocorrência de alteração identificada, obtém-se eficiência em operações de consulta integrada. Isto dado que informações são continuamente correlacionadas a um custo aceitável de perda de consistência, quando as integrações ocorrem tão rápido quanto o processamento dos eventos, e sem que o grau de desacoplamento entre os componentes seja comprometido.

Em trabalho anterior [14], identificamos e analisamos as principais estratégias na literatura para a integração de dados em microsserviços com bases poliglotas. Dentre estas, duas são orientadas a eventos e se destacam por apresentarem

como vantagens o desacoplamento e a eficiência na integração de dados atualizados nas diferentes fontes de informações. Outras propostas foram identificadas na literatura com implementações no contexto deste trabalho. Para fins de comparação, estes estudos, juntamente com a proposta deste trabalho, serão avaliados considerando o mesmo cenário de negócio e considerando os critérios dos modelos ISO 25010 e ISO 25012.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os conceitos fundamentais. A Seção 3 apresenta o problema de pesquisa. A Seção 4 apresenta os trabalhos relacionados às estratégias atuais. A Seção 5 apresenta a proposta. A Seção 6 apresenta os critérios de avaliação. A Seção 7 apresenta o cronograma de atividades desta dissertação. Finalmente, a Seção 8 apresenta a conclusão.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os principais conceitos necessários para o entendimento deste trabalho.

2.1 Microsserviços

Na arquitetura de microsserviços, sistemas são compostos por serviços que colaboram entre si com base em sete princípios [16]: (i) Desenvolvimento de unidades autônomas, com responsabilidades específicas; (ii) Uso de práticas de desenvolvimento orientadas ao negócio; (iii) Uso de diretrizes da Cloud como distribuição e elasticidade; (iv) Entrega contínua descentralizada; (v) Uso *containers* leves; (vi) Uso de abordagens automatizadas para configuração, desempenho e gerenciamento de falhas; (vii) Uso de diferentes estratégias de programação e persistência.

Persistência poliglota [11] implica em escolher o melhor modelo de persistência para cada tarefa a ser realizada, considerando que cada serviço deve ser projetado para resolver com maior eficiência seus problemas específicos.

2.2 Plataforma de Streaming e CEP

Uma plataforma de *streaming* atua como um sistema de mensageria [8], por meio de publicação e assinatura para a distribuição de dados gerados frequentemente e de forma contínua. Esta plataforma é composta por produtores, consumidores e *brokers*, que podem ser distribuídos na rede para melhorar o desempenho do sistema. Implementações dessa plataforma trazem benefícios, como eficiência (por exemplo, publicação em lote, compactação de dados históricos e APIs (Application Program Interface) que permitem processamento paralelo e avaliação tardia de operações) e usabilidade (por exemplo, visão de tabela para resgatar o estado atual de objetos).

A partir da demanda gerada devido ao volume desses dados e da necessidade de seu processamento em tempo próximo ao real surge o conceito de processamento de eventos complexos, ou CEP (Complex Event Processing) [3], cujo objetivo é gerar informações úteis a partir do tratamento das informações continuamente coletadas.

Alguns estudos demonstram eficiência no uso de microsserviços, classificados como orientados a eventos (Seção 4), aliados à utilização de plataforma de streaming [4, 17]. Entretanto, estes não foram empregados em bases de dados poliglotas.

3. PROBLEMA

Esta dissertação objetiva tratar o problema de integração de dados provenientes de microsserviços que empregam persistência poliglota e cujos dados são atualizados frequentemente. Como exemplo de cenário real, dados relacionados à manutenção preventiva dos equipamentos que suportam a operação dos meios de transporte de combustíveis de uma importante empresa para o transporte e a logística de combustível no Brasil são armazenados por sistemas distintos (tais como sistemas de gestão na rede corporativa, sistemas supervisórios na rede de automação, dados de telemetria captados por sensores) empregando diferentes tecnologias de bancos de dados. O desafio então surge da necessidade de correlacionar, em tempo próximo ao real, as informações provenientes destes sistemas permitindo consultas e visualizações destes dados de forma integrada.

Este problema vem sendo reiteradamente apontado como desafio pela academia e pela indústria. Soluções publicadas apresentam uma série de aspectos positivos e negativos como, por exemplo, aumento de disponibilidade, complexidade e alto grau de acoplamento). A solução para tratar este problema deve promover eficiência e simplicidade na integração de informações geradas em grande volume de dados.

Para tratar este problema será necessário abordar os desafios de se correlacionar sistemas distribuídos com essa natureza. De acordo com o teorema CAP [2] - que trata de consistência (C), disponibilidade (A) e tolerância de partição (P), ao menos um desses aspectos será comprometido em qualquer configuração com sistemas que compartilham dados. Como as partições de rede são inerentes a sistemas distribuídos, temos que equilibrar a disponibilidade e a consistência. O volume e a variedade de dados oriundos de fontes distintas de informação, ditas poliglotas, pode impactar ambos aspectos, dificultando sua integração.

4. TRABALHOS RELACIONADOS

Newman lista duas estratégias empregadas para integração de dados em microsserviços [9, 14] com características relacionadas a processamento de eventos:

- *Command-Query Responsibility Segregation (CQRS)*: Propõe uma "visão materializada" do banco de dados para buscas a fim de isolar as operações transacionais das de processamento de consultas;
- *Event Data Pump*: Propõe a publicação de dados relacionados a eventos de atualizações de dados em uma plataforma que permite que sejam capturados por componentes projetados para consolidação de informações.

Nesta dissertação, a classificação "Orientada a Eventos"[1] é utilizada, abrangendo *Event Data Pump* e o caso particular em que *CQRS* tem sua visão construída a partir de uma fila de eventos (processo denominado "Event Sourcing" [9]).

A partir de uma revisão sistemática nas bibliotecas digitais IEEE Xplore, ACM e Scopus (entre Jan/2013 e Fev/2018), 215 publicações distintas foram filtradas a partir da seguinte *string* de busca: *"(microservice) AND (architecture OR design OR project) AND (event* OR reactive OR data OR integration* OR consolidation* OR query OR queries OR cqrs OR stream*)"*. A partir da leitura de todos os resumos, 205 publicações

[1]Sistemas orientados a eventos são projetados para processar eventos à medida que ocorrem, permitindo que o sistema observe e reaja dinamicamente [5].

foram excluídas por não apresentarem afinidade com a proposta. Após a leitura de toda a publicação nos demais casos, os quatro estudos abaixo foram identificados como aplicáveis a este estudo:

- Viennot *et al.* [13] propõem Synapse, um ecosistema de microsservicos escalável no qual cada serviço complementa o modelo de dados dos outros por meio de mapeamentos ORM e estratégias de mensageria;
- Kookarinrat e Temtanapat [7] propõem uma arquitetura de microsserviços que utiliza um barramento de mensagens para desacoplar a comunicação entre seus componentes e mitigar cenários de SPF (Single Point of Failure);
- Richter *et al.* [10] avaliaram uma arquitetura de microsserviços com ênfase em confiabilidade segundo requisitos de conteinerização, comunicação via filas de mensagens, tolerância a falhas e alta disponibilidade;
- Zappatore *et al.* [15] propõem uma plataforma baseada em Mobile Crowd Sensing que permite coletar fluxos de dados de sensores embutidos em *smartphones* em grandes escalas geográficas e possibilita análises sobre os mesmos.

Estes trabalhos utilizam mecanismos diferentes do proposto para a publicação e tratamento dos eventos.

Conforme avaliação prévia [14], um dos aspectos que se destacou de forma negativa nestas estratégias orientadas a eventos foi a confiabilidade, especialmente quando relacionada ao grau de consistência dos dados na solução. Outro aspecto crítico observado foi a manutenibilidade, dado o nível de complexidade das soluções.

Neste trabalho, busca-se criar uma arquitetura que resolva o problema de integração de forma mais eficiente que as propostas existentes, mantendo o benefício de baixo acoplamento preconizado nas estratégias orientadas a eventos.

5. PROPOSTA DE SOLUÇÃO

Este trabalho tem por objetivo aprimorar a integração de dados oriundos de bases de dados poliglotas empregando uma arquitetura orientada a eventos, bem como minimizar o acoplamento entre os componentes de software. A abordagem proposta (Figura 1) usa uma plataforma de streaming, escolhida de acordo com as diretrizes de Stonebraker *et al.* [12], e possui os seguintes componentes:

- **Fontes de Dados**: Serviços ou microsserviços que efetuam operações transacionais em suas bases de dados;
- **Agentes Coletores de Dados**: Microsserviços que obtêm dados (por exemplo, tabelas, trilhas de auditorias, arquivos de *log*, sensores IoT, etc) e publicam eventos em uma plataforma de streaming;
- **Plataforma de Streaming**: Plataforma em que são tratados o enfileiramento e o processamento de eventos, além de questões como as relacionadas à sincronia e à ordenação.
- **Consolidadores**: Microsserviços que subscrevem informações providas pelos agentes, e, por meio de mecanismos de processamento de eventos complexos (Complex Event Processing - CEP), consolidam e publicam esse resultado como novos eventos (*streams*), promovendo reuso e desacoplamento entre as novas *streams* e as visões elaboradas pelos provedores a seus consumidores;
- **Provedor**: Microsserviço que disponibiliza serviços web (por exemplo, em REST/Json ou GraphQL) para prover informações consolidadas (tabelas ou visões) a partir dos *streams* publicados pelos consolidadores.
- **Consumidor**: Consome as informações providas pelos serviços do provedor.

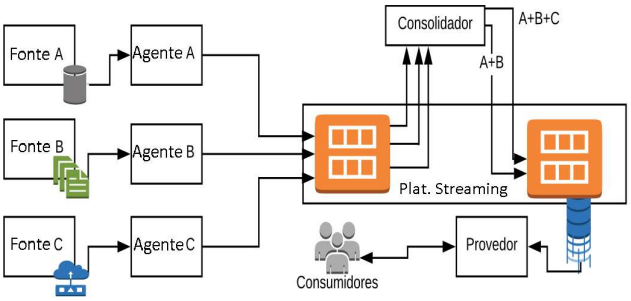

Figure 1: Componentes da Arquitetura Proposta

6. AVALIAÇÃO

Um conjunto de sensores mapeado pela equipe responsável pelas atividades de manutenção preventiva de uma importante empresa para o transporte e a logística de combustível no Brasil será utilizado para a validação da proposta desta dissertação.

A avaliação será realizada de duas formas: (i) avaliação quantitativa e qualitativa da qualidade de software segundo duas normas de qualidade de software; (ii) execução de experimentos comparativos com abordagens concorrentes.

O primeiro inclui critérios de confiabilidade (abrangendo disponibilidade), eficiência de desempenho, usabilidade, compatibilidade, manutenibilidade e portabilidade, conforme a norma ISO 25010. Esta é uma evolução do modelo ISO 9126, e foi considerada como o padrão mais importante para avaliar recursos de qualidade em soluções de software [1]. O fator consistência será considerado segundo a ISO 25012, que complementa a ISO25010.

Os experimentos avaliarão a eficácia e eficiência da proposta em relação às soluções existentes, considerando: (i) Tempos de processamento e de resposta: medidos entre a requisição do consumidor (usuário) e seu atendimento, e entre a geração do evento(a partir dos nós produtores) e sua atualização nas *streams* consolidadas; (ii) Taxa de ocorrência dos eventos gerados nos nós produtores: frequência dos eventos gerados (nos nós produtores); (iii) Assertividade nas respostas: precisão e completude dos registros retornados; (iv) Consumo de recursos de infraestrutura: número de nós, capacidade de processamento e armazenamento durante a execução das atividades; (v) Resiliência quando nós perdem a conexão: comportamento do sistema quando nós aleatoriamente perdem a conexão com a plataforma de streaming;

As avaliações qualitativas serão realizadas pelo autor, mas serão submetidas à apreciação de 10 profissionais especialistas, pesquisadores ou analistas com experiência mínima de 10 anos na área, e o percentual de aprovação da análise do autor será divulgado. Serão destacadas e justificadas eventuais vantagens e desvantagens da nova arquitetura.

7. CRONOGRAMA

A Figura 2 apresenta o cronograma desta dissertação. O estudo da literatura já foi feito em grande parte. O problema já foi definido bem como já foram encontrados os trabalhos relacionados e que serão comparados com a proposta desta dissertação. A proposta já foi definida. A implementação está iniciando. O cenário do estudo de caso já foi definido, bem como a forma de avaliação. Um artigo foi publicado no ACM SAC 2018 [14] (qualis A1) apresentando uma análise qualitativa das propostas existentes, destacando seus pontos fortes e fracos. Foram aceitos para publicação este WTD, que apresenta a proposta da dissertação, e um minicurso para lecionar os a construção de sistemas de informação empregando microsserviços.

	2017												2018											
	01	02	03	04	05	06	07	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11	12	01
Estudo da literatura			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X							
Definição da proposta														X	X										
Implementação															X	X	X	X	X	X	X				
Realização do estudo																		X	X	X	X	X	X		
Análise dos dados																				X	X	X	X	X	
Escrita da monografia																						X	X	X	
Escrita de artigos + Minicurso de 8 horas						X	X	X	X				X	X	X	X	X	X					X	X	X
Defesa																									X

Figure 2: Cronograma de Atividades

8. CONCLUSÃO

A arquitetura de microsserviços preconiza uma série de práticas que proporciona autonomia a seus componentes. Entretanto, o nível de isolamento pretendido nessa abordagem pode dificultar a correlação de dados provenientes de microsserviços que compõe uma solução, especialmente num cenário de persistência poliglota em que um grande volume de dados é atualizado de forma frequente. O problema vem sendo reiteradamente discutido na academia, e soluções publicadas apresentam aspectos positivos e negativos relacionados, por exemplo, à disponibilidade, consistência, e grau de acoplamento.

Com base nos estudos existentes e nas inovações proporcionadas pelos sistemas gerenciadores de *streams*, esta dissertação tem o objetivo de construir uma nova solução com maior efetividade na integração de dados.

Como contribuições acadêmicas, este trabalho apresenta uma nova arquitetura para tratar microsserviços com persistência poliglota em um cenário de alta frequência na atualização de dados e faz uma avaliação comparativa com propostas existentes. Como contribuição tecnológica está a implementação da arquitetura proposta empregando tecnologias de software livre e métricas indicando sua efetividade.

9. REFERENCES

[1] B. Behkamal, M. Kahani, and M. K. Akbari. Customizing iso 9126 quality model for evaluation of b2b applications. *Information and software technology*, 51(3):599–609, 2009.

[2] E. A. Brewer. Towards robust distributed systems. In *PODC*, volume 7, 2000.

[3] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):15, 2012.

[4] A. Debski, B. Szczepanik, M. Malawski, S. Spahr, and D. Muthig. In search for a scalable & reactive architecture of a cloud application: Cqrs and event sourcing case study. *IEEE Software*, 2017.

[5] O. Etzion, P. Niblett, and D. C. Luckham. *Event processing in action*. Manning Greenwich, 2011.

[6] M. Fowler and J. Lewis. Microservices. https://martinfowler.com/articles/microservices.html, 2014. Acessado em: 02/02/2018.

[7] P. Kookarinrat and Y. Temtanapat. Design and implementation of a decentralized message bus for microservices. In *Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on*, pages 1–6. IEEE, 2016.

[8] J. Kreps, N. Narkhede, J. Rao, et al. Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, pages 1–7, 2011.

[9] S. Newman. *Building microservices: designing fine-grained systems*. "O'Reilly Media, Inc.", 2015.

[10] D. Richter, M. Konrad, K. Utecht, and A. Polze. Highly-available applications on unreliable infrastructure: Microservice architectures in practice. In *Software Quality, Reliability and Security Companion (QRS-C), 2017 IEEE International Conference on*, pages 130–137. IEEE, 2017.

[11] P. J. Sadalage and M. Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2012.

[12] M. Stonebraker, U. Çetintemel, and S. Zdonik. The 8 requirements of real-time stream processing. *ACM Sigmod Record*, 34(4):42–47, 2005.

[13] N. Viennot, M. Lécuyer, J. Bell, R. Geambasu, and J. Nieh. Synapse: a microservices architecture for heterogeneous-database web applications. In *Proceedings of the Tenth European Conference on Computer Systems*, page 21. ACM, 2015.

[14] L. H. Villaca, L. G. Azevedo, and F. Baiao. Query strategies on polyglot persistence in microservices. In *2018 ACM SIGAPP - a ser publicado*, 2018.

[15] M. Zappatore, A. Longo, and M. A. Bochicchio. Crowd-sensing our smart cities: a platform for noise monitoring and acoustic urban planning. 2017.

[16] O. Zimmermann. Microservices tenets. *Computer Science-Research and Development*, pages 1–10, 2016.

[17] T. Zschörnig, R. Wehlitz, and B. Franczyk. A personal analytics platform for the internet of things. 2017.