

Modelagem de Dados baseado em Computação em Névoa para Cidades Inteligentes

Gabriela Oliveira Biondi

Universidade Federal do ABC (UFABC)

Gabriela.biondi@ufabc.edu.br

Resumo: As Cidades Inteligentes visam aplicar recursos de Internet das Coisas (Internet of Things - IoT) para melhorar a qualidade de vida de uma comunidade como um todo. O objetivo deste resumo é apresentar uma modelagem híbrida de dados baseada em computação em nuvem para cidades inteligentes.

1. Introdução

O conceito de Cidades Inteligentes (do inglês Smart Cities) visa aplicar recursos de IoT não apenas para melhorar a qualidade de vida de um indivíduo, mas sim de uma comunidade como um todo, criando soluções que promovam cidadania, sustentabilidade, integração social, inclusão e participação, através do uso de tecnologias inovadoras [1].

Essas soluções geralmente são baseadas na geração, armazenamento e processamento de um grande volume de dados. Além do desafio do processamento em tempo hábil, existe também o desafio de fazê-lo de forma eficiente, a um custo aceitável e com possibilidade de rastreamento. Para isso, é preciso combinar a integridade dos tradicionais modelos relacionais SQL com a versatilidade dos modelos NoSQL [2], usando cada um deles onde é mais conveniente (modelo relacional na nuvem e modelo NoSQL na névoa).

O objetivo deste resumo é apresentar uma modelagem de dados baseada em computação em nuvem para cidades inteligentes, usando modelagem NoSQL nas esferas locais e modelagem relacional na esfera global.

2. Proposta

Uma Cidade Inteligente precisa ter uma modelagem de dados que apresente as seguintes quatro características:

Genérica, pois diferentes cenários precisam ser implementados sem que novas entidades sejam criadas e associadas ao modelo existente. Ou seja, é preciso construir uma modelagem que seja composta por entidades genéricas, que possam se adaptar para funcionar nos diferentes cenários de uma Cidade Inteligente.

Flexível, pois 90% dos dados gerados em uma Cidade Inteligentes são "desestruturados" (uma das características previstas em um Big Data [3]), vindos de diversas fontes e formas possíveis. Eles podem ser dados de GPS, tweets com conteúdo e sentimento, dados visuais como fotos e vídeos, dados de sensores, entre outras coisas. Neste contexto, a modelagem de dados precisa considerar diferentes paradigmas para armazenar, e posteriormente processar, todos esses dados de forma otimizada. Em outras palavras, é necessário mesclar o uso da modelagem relacional tradicional com modelagens NoSQL.

Performática, pois os cenários típicos de uma Cidade Inteligente exigem baixa latência, mesmo perante ao grande volume de dados que é gerado. Neste contexto, os paradigmas NoSQL oferecem vantagens, pois ao abrir mão da integridade relacional construída através de amarrações com chaves estrangeiras, ganhamos desempenho no armazenamento e nas consultas.

Rastreável, pois é natural que se deseje auditar eventos ocorridos no passado para compreender os fatos ou para aprender com eles. Neste contexto, o paradigma relacional oferece vantagens, pois permite relacionar registros através de chaves estrangeiras. Para atender esta característica, além de usar algoritmos de rastreamento nos dados, também podemos usar algoritmos de aprendizado de máquina, como o SVM (do inglês Support Vector Machine), por exemplo, para prever eventos futuros.

Para garantir a primeira característica, que diz que uma modelagem de dados para cidades inteligentes precisa ser genérica, este trabalho de pesquisa propõe o uso de um conjunto de onze entidades genéricas que se relacionam entre si: Place, Subject, Resource, Service, Consumable, Action, Decision, Schedule, Fusion, Context e Result.

O armazenamento dos dados dessas onze entidades precisa garantir a segunda característica, ser flexível. Para isso, é preciso considerar o formato do dado e seu propósito para a escolha do melhor paradigma de modelagem. Neste contexto a modelagem de dados proposta por este trabalho divide os dados em duas esferas: local, com dados pertinentes a uma região específica; e global, com dados pertinentes a cidade como um todo. Na esfera local é usado uma modelagem NoSQL e na esfera global é usado uma modelagem relacional, garantindo assim que diferentes formatos de dados possam ser armazenados adequadamente, de acordo com a sua natureza. Desta forma também garantimos a terceira e quarta característica, a performance (NoSQL) e a rastreabilidade (SQL), respectivamente.

A *Figura 1* apresenta uma arquitetura mostrando a relação entre os componentes de software de um gerenciador de contexto para cidades inteligentes e a modelagem de dados proposta neste resumo. Elementos usados:

- **SQL Operações e Previsões:** realiza o armazenamento de dados da nuvem através de um SGBD relacional.
- **NoSQL:** realiza o armazenamento de dados da névoa através de um SGBD não relacional.
- **Fuser:** realiza a fusão de dados, ou seja, a utilização de técnicas para combinar dados de múltiplas fontes.
- **Reasoner:** infere consequências lógicas de um conjunto de fatos para disparar uma regra, onde uma ou mais ações são executadas para alterar a configuração de dispositivos.
- **MQTT:** encapsula a comunicação dos recursos, interfaceando o Gerenciador de Comunicação e os sensores.
- **Buffer:** armazena os dados temporariamente, enquanto o SGBD relacional faz uma gravação assíncrona.
- **Tracker:** usa algoritmos que rastreamento para fazer consultas complexas no banco de dados relacional para descobrir a ligação entre todos os acontecimentos na cidade.
- **IA:** usa algoritmos de aprendizado de máquina, como o SVM, para fazer a predição de situações que podem ocorrer na cidade, baseando-se nos dados históricos armazenados na plataforma.
- **Sensores:** Os sensores foram emulados usando o SenSE¹, um gerador de dados de IoT open-source.

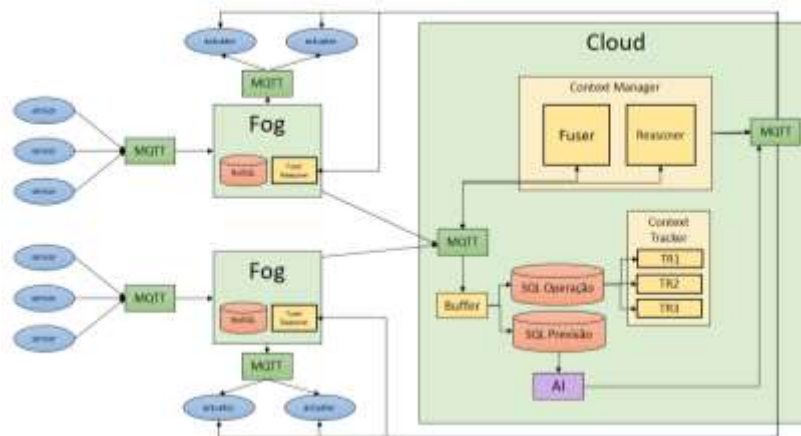


Figura 1: Arquitetura de um Gerenciador de Contexto que usa a modelagem proposta

Esta arquitetura faz uso de recursos de software na nuvem e na névoa. O armazenamento SQL Operação em nuvem garante que os dados sejam genéricos e posteriormente rastreáveis, enquanto que o armazenamento NoSQL na névoa garante a flexibilidade dos dados usados no modelo. O buffer usado no armazenamento SQL garante que não será criado um gargalo para fazer um armazenamento online, pois este processo pode ser feito em batch. Desta maneira, garantimos as características todas as quatro características sugeridas nesta pesquisa: rastreável, flexível, genérica e performática. Além disso, as entidades de precisão no armazenamento SQL da nuvem permite usar os dados históricos que o gerenciador de contexto recebeu, armazenou e processou, para prever situações futuras e se preparar para elas. É possível usar esses dados para auto ajustar as fusões e regras da névoa, com base nos resultados das previsões. Assim, garantimos que a plataforma de auto adapte a situações adversas. O módulo de AI (Artificial Inteligente – Inteligência Artificial) fará uso de algoritmos de Aprendizado de Máquina [4], como SVM (Support Vector Machine – Máquina de Vetores de Suporte) por exemplo, para fazer a previsões.

3. Conclusão

As modelagens NoSQL foram indispensáveis na esfera local, pois elas possuem um armazenamento e processamento rápido, garantindo assim a baixa latência. A modelagem relacional foi indispensável, pois ela garante a integridade dos registros através do uso de chaves estrangeiras, permitindo assim a rastreabilidade dos eventos.

4. Referências

- [1] Monroy-Hernández, A., Farnham, S., Kiciman, E., Counts, S., Choudhury, M., "Smart Societies: From Citizens as Sensors to Collective Action", ACM Interactions Magazine, 20(4), pp 16-19, August 2013.
- [2] Leavitt, N., "Will NoSQL Databases Live Up to Their Promise?", IEEE Computer, 43(2), 12-14, Fevereiro 2010.
- [3] Dumbill, E., "Making Sense of Big Data", Big Data Journal, 1(1), March 2013.
- [4] Biondi, G., Prati, R., "Setting Parameters for Support Vector Machines using Transfer Learning", Journal of Intelligent & Robotic Systems, 2015.

¹ github.com/ivanzy/SenSE-Sensor-Simulation-Environment