

Compreendendo o Desempenho de Gerenciadores de Contexto para Internet das Coisas

Ivan Zyrianoff, Fabrizio Borelli, Alexandre Heideker,
Gabriela Biondi, Carlos Kamienski

Universidade Federal do ABC (UFABC)

ivan.dimitry@aluno.ufabc.edu.br, fabrizio.borelli@ufabc.edu.br
{alexandre.heideker,gabriela.biondi,cak}@ufabc.edu.br

Resumo. *Sistemas de Gerenciamento Sensíveis ao Contexto tem sido propostos para realizar a tomada de decisão automática para a Internet das Coisas. Embora escalabilidade seja uma característica imprescindível nesses sistemas, não existem resultados abrangentes de desempenho disponíveis. Esse artigo apresenta resultados de avaliação de desempenho de diferentes arquiteturas sensíveis ao contexto e apresenta a ferramenta SenSE, que gera dados sintéticos de sensores. Os resultados mostram que diferentes escolhas de arquitetura impactam no desempenho e na escalabilidade do sistema e que é factível a tomada de decisão em tempo real em um ambiente com dezenas de milhares de sensores enviando dados continuamente.*

Abstract. *Context-Aware Management Systems have been proposed in the last years to perform automatic decision making for the Internet of Things. Although scalability is an indispensable feature for those systems, there are no comprehensive results reporting their performance. This paper shows results of a performance analysis study of different context-aware architectures and introduces the SenSE platform for generating sensor synthetic data. Results show that different architectural choices impact system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousands of sensors that continuously transmit data.*

1 Introdução

A geração e o uso eficiente de energia estão atualmente em evidência, pelo impacto ambiental e uso sustentável dos recursos naturais. Como a demanda mundial por energia tende a crescer, surge o desafio de usar soluções tecnológicas para gerenciar seu uso de maneira mais eficiente. Cidades inteligentes aceleram o crescimento econômico sustentável e a qualidade de vida com um gerenciamento adequado dos recursos naturais através de um processo participativo de governança [Caragliu et. al 2011]. A Internet das Coisas (IoT) desempenha um papel fundamental na implantação de cidades inteligentes, viabilizando a coleta de dados de uma enorme quantidade de sensores, a análise de dados, tomada de decisões e o envio de comandos para atuadores que alteram o comportamento dos sistemas. A computação baseada em contexto oferece a inteligência necessária à IoT, junto com big data e outras tecnologias, permitindo a modificação dos comportamentos na velocidade necessária sem a intervenção humana direta [Osello et. al 2013].

Sistemas de gerenciamento sensíveis aos contexto para Cidades Inteligentes baseado em IoT recebem uma quantidade imensa de dados provenientes de sensores de diversos tipos e funções e devem tomar decisões para enviar comandos a atuadores para que o ambiente sofra alterações automáticas. Para que o sistema seja capaz de tratar desses

dados e enviar respostas em tempo real, principalmente quando o ambiente gerenciado pode ser tão reduzido quanto uma sala ou tão grande quanto uma cidade, alto desempenho e escalabilidade são fundamentais. Embora escalabilidade seja uma característica imprescindível nesses sistemas, não existem resultados abrangentes de desempenho disponíveis.

Este artigo apresenta uma avaliação de desempenho de gerenciadores de contextos para ambientes de IoT, usando o sistema IMPReSS [Kamienski et. al 2016] como estudo de caso, no qual o impacto de diferentes módulos e implementações foram avaliados. Para realizar a avaliação de desempenho foi desenvolvido o simulador SenSE, que gera dados sintéticos de diferentes categorias de sensores. Considerando que a literatura não apresenta avaliações de desempenho de sistemas sensíveis ao contexto para IoT, esse artigo apresenta três contribuições principais: 1) A metodologia de avaliação, que foi desenvolvida especialmente para esse artigo; 2) O simulador de sensores SenSE, necessário para testar a escalabilidade da solução; 3) Os resultados da avaliação de desempenho considerando diferentes arquiteturas de gerenciadores de contexto, que mostram a escalabilidade da solução.

Na sequência desse artigo a seção 2 apresenta trabalhos relacionados e a seção 3 apresenta uma visão de gerenciamento sensível ao contexto para IoT. As seções 4 e 5 apresentam a metodologia e os resultados de avaliação, respectivamente. A seção 6 discute os resultados e as lições aprendidas e finalmente a seção 7 tece conclusões.

2 Trabalhos Relacionados

Em [Perera et al. 2014] foi apresentado um amplo estudo sobre computação sensível ao contexto em IoT, incluindo comparações de diversas características técnicas de soluções comerciais existentes. Porém, o estudo comparativo foi restrito às características, não envolvendo a avaliação de desempenho de nenhuma das ferramentas.

Em [Mongiello et al 2016] é proposto um *middleware* para IoT se adapta de acordo com o contexto, usando o paradigma de programação reflexiva. Apesar de apresentar um *middleware* para IoT, esta proposta se difere da deste artigo em dois pontos: (1) não faz o gerenciamento do contexto, mas sim uma adaptação de acordo com o contexto; e (2) não apresenta uma avaliação de desempenho para mostrar o comportamento em situações de alta latência.

A construção de um servidor de banco de dados para um *middleware* de IoT é apresentada em [Paethong et al. 2016], com desempenho satisfatório e custo acessível. Esta pesquisa faz a avaliação de desempenho comparando o MongoDB e o MySQL em operações de CRUD. Apesar de avaliar o impacto do uso de banco de dados, o trabalho não considera o processamento destes por um gerenciador de contexto.

Finalmente em [Aman et al. 2016] é apresentado um estudo sobre a paralelização dos sistemas de *middleware* para IoT, através de uma avaliação de desempenho que considera o uso da tecnologia *multicore*. Porém, os autores não consideram em sua avaliação o impacto dos diversos componentes de um gerenciador de contexto. Não foram encontrados na literatura trabalhos diretamente relacionados à avaliação de desempenho de gerenciadores de contexto.

3 Gerenciamento de Contexto em IoT

A arquitetura de um gerenciador de contexto em sistemas de IoT deve conter alguns módulos básicos para que seu propósito seja atingido. A implementação de cada um desses componentes pode ser realizada com diferentes combinações de ferramentas ou

com implementações específicas (*hardcoded*). A Figura 1 apresenta uma visão simplificada da arquitetura de um gerenciador de contexto em sistemas de IoT (outros módulos foram omitidos para prezar pela clareza).

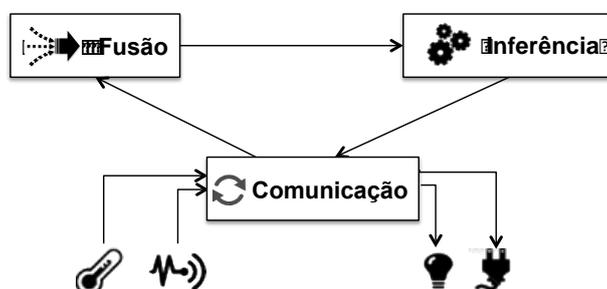


Figura 1. Arquitetura de Gerenciamento de Contexto

- **Fusão:** Realiza a fusão de dados, usando um conjunto de técnicas para combinar dados de múltiplas fontes. Seu propósito é obter dados mais precisos, aumentando a sua granularidade. A fusão é um passo importante para diminuir a carga de dados para a inferência, mas esse módulo não é obrigatório na arquitetura.
- **Inferência:** O motor de Inferência infere consequências lógicas de um conjunto de fatos. Quando invocado pela Fusão com um conjunto de parâmetros, ele faz o casamento com um conjunto de regras. Caso uma regra seja encontrada, ações são executadas que enviam comandos a atuadores.
- **Comunicação:** Receber os dados dos sensores e envia comandos aos atuadores.

Nesse artigo é utilizada a implementação de gerenciador de contexto do IMPReSS [Kamiński et. al 2015], que usa os mesmos componentes da Figura 1 sendo que para Fusão, Inferência e Comunicação são utilizados Esper¹, Drools² e broker MQTT Mosquitto³ respectivamente.

4 Cenário: Ambientes de IoT em Cidades Inteligentes

O ambiente dos experimentos foi baseado na modelagem de Cidades Inteligentes criada na cidade Pádua na Itália [Zanella et. al., 2014].

4.1 Gerador de tráfego – SenSE (*Sensor Simulation Environment*)

Apesar de existirem diversas propostas de *middlewares* e sistemas de tempo real para IoT e Cidades Inteligentes, não existe uma plataforma de testes adequada para comprovar a eficiência e escalabilidade desses softwares, principalmente em cenários envolvendo milhares de dispositivos conectados. Para a realização da avaliação de desempenho foi desenvolvido o SenSE (*Sensor Simulation Environment*) [Zyrianoff et. al 2017], um gerador de dados sintéticos de sensores de código aberto⁴. O objetivo do SenSE é simular o tráfego de dados gerado em ambientes complexos que geram grande quantidade de dados simultaneamente, enviando esses dados para os sistemas através de protocolos IoT como o MQTT (usado nesse artigo).

O estudo de sensores em ambientes IoT identificou dois grandes grupos de sensores que se diferenciam pelo modo de gerar os dados:

¹ <http://www.espertech.com/>

² <https://www.drools.org>

³ <https://mosquitto.org/>

⁴ <https://github.com/ivanzy/SenSE-Sensor-Simulation-Environment>

- Sensores movidos por tempo: enviam dados periodicamente para reportar um estado (ex.: um sensor de qualidade do ar que envia dados a cada 30 minutos);
- Sensores movidos por evento: enviam dados caso o seu estado atual seja alterado (ex.: um sensor de presença que só envia dados caso alguma presença seja detectada).

A ferramenta SenSE permite criar vários tipos de sensores com base em distribuições de probabilidade, sendo possível criar tanto sensores movidos por tempo quanto por evento. Além disso é possível definir o tipo de sensor (ex.: temperatura ou qualidade do ar), a periodicidade no envio de dados (ex.: a cada 30 segundos ou a cada 6 horas), o tipo de dado a ser enviado (ex.: inteiro, booleano, dentro outros), bem como o intervalo de valores desse tipo de dado (ex: um sensor de temperatura envia dados com valor mínimo de 25 e máximo de 30, para graus Celsius), além de outras configurações específicas. O SenSE é capaz de simular dezenas de milhares de sensores, de diversas configurações, simultaneamente e em uma única execução. A Figura 2 mostra uma interface de configuração do SenSE.

Figura 2. SenSE (Sensor Simulation Environment)

O SenSE utiliza filas de tarefas para simular uma grande quantidade de sensores movidos por tempo. Cada tipo de sensor possui uma fila e eles são instanciados em um tempo aleatório seguindo a distribuição Uniforme entre o tempo zero e a periodicidade daquele tipo de sensor. Por exemplo, sensores de qualidade do ar enviam dados com periodicidade de 30 minutos. Em uma simulação de 10.000 sensores de qualidade do ar, todos iniciam o envio de dados na primeira meia hora do experimento.

É complexo modelar sensores movidos a evento, já que é necessário uma mudança no seu estado gerada por algum fator externo. Considerando um cenário de Cidades Inteligentes, foram modelados sensores cujo acionamento é causado pelo comportamento humano e a modelagem é feita para simular pessoas chegando à um evento (ex.: pessoas chegando a um estádio) e a chegada de um indivíduo modifica o estado do sensor, fazendo com que uma mensagem seja enviada.

4.2 Ambiente de Avaliação

A modelagem dos sensores considera cinco tipos de sensores movidos por tempo e apenas um movido por evento, caracterizados na Tabela 1. O objetivo é submeter o sistema a uma carga semelhante à encontrada em ambientes de computação urbana reais, já que os tipos de sensores escolhidos e seus tempos de envio se baseiam em um cenário real [Zanella et. al., 2014]. A avaliação de desempenho utiliza uma carga de trabalho de dezenas de milhares de sensores, para avaliar a escalabilidade da solução.

No ambiente de execução dos experimentos o SenSE envia dados simulados de sensores para uma plataforma IoT (nesse caso, um gerenciador de contexto) que realiza

algum tipo de processamento e transmite a informação para os atuadores. Foi desenvolvida uma aplicação *Sink* que abstrai um conjunto de atuadores, recebendo os dados processados e salvando os mesmos. Além disso, o SenSE foi configurado para enviar um *timestamp* em cada mensagem e o *Sink* marca o *timestamp* de chegada da mensagem e realiza uma operação de subtração para descobrir o tempo de processamento de cada mensagem pela plataforma IoT. Perdas da rede foram abstraídas, pois o intuito é avaliar o desempenho e escalabilidade da plataforma IoT isolada, e uma falha iria gerar dúvidas se o problema foi da plataforma ou da rede.

Tabela 1. Lista de sensores utilizados nos experimentos

Tipo de Sensor	Taxa de envio (pacotes)	Categoria
Qualidade do ar	1 mensagem a cada 30 min por sensor	Tempo
Barulho	1 mensagem a cada 30 min por sensor	Tempo
Estrutura de edifícios	1 mensagem a cada 10 min por sensor	Tempo
Congestionamento (trânsito)	1 mensagem a cada 10 min por sensor	Tempo
Gestão de resíduos	1 mensagem a cada 60 min por sensor	Tempo
Iluminação	1 mensagem quando alguém é detectado	Evento

Para a execução dos experimentos, o gerenciador de contexto foi executado em um servidor dedicado com configuração de processador Intel(R) Xeon(R) CPU E3-1240 V2 @ 3.40GHz, 450 GB HD de disco, 8GB de memória RAM e o sistema operacional Ubuntu 14.04. Em uma máquina virtual localizada em outro servidor estava sendo executado o SenSE e o *Sink*. A configuração da VM é Intel Xeon E5520 2.27 Ghz Octa Core (Virtual), possui 11GB de RAM e 80GB de HD.

4.3 Fusão de Dados e Inferência de Regras

Nos experimentos realizados com o módulo de Fusão há um critério de fusão para cada tipo de sensor apresentado na Tabela 1 que retransmite a mensagem. Neste caso específico, o módulo responsável pela fusão dos dados não agrega mensagens. Nos experimentos foi utilizado o Esper como módulo de Fusão.

Foram criadas 6 regras para esse cenário, uma para cada tipo de sensor apresentado na Tabela 1. Experimentos preliminares mostraram que o número de regras não apresenta impacto significativo no desempenho do motor de regras (nesse caso, o Drools). Além disso, o objetivo é avaliar a plataforma como um todo e não somente o módulo do motor de inferência. A inferência é realizada sobre a mensagem recebida do módulo anterior - dependendo da arquitetura a mensagem pode vir do módulo de Fusão ou diretamente de um pré-processamento *hardcoded* realizado na mensagem do sensor, acionando uma regra que irá gerar uma ação e fará com que seja enviada uma mensagem para um atuador (no caso, para o *Sink*).

5 Metodologia de Avaliação

5.1 Métricas

Com base nos dados obtidos, é calculado o tempo médio total que uma mensagem leva para ser processada no gerenciador de contexto inteiro, obtida pela média de tempo que cada mensagem leva para percorrer todos os componentes do sistema, até ser recebida pelo atuador. Também é registrada a taxa de entrega do sistema, com o objetivo de verificar se a plataforma avaliada possui alguma taxa de perda de mensagens em picos de processamento.

5.2 Experimentos

Os experimentos de avaliação de desempenho medem o tempo de resposta e a taxa de entrega de mensagens entre uma ação e a reação do contexto no cenário controlado. O SenSE foi utilizado para simular diversas quantidades de sensores diferentes enviando mensagens, de modo a tornar possível a compreensão do funcionamento do gerenciador de contexto, em um ambiente de cidades inteligentes.

Para compreender o impacto de diferentes tipos de sensores foram realizados experimentos apenas com sensores movidos por tempo, apenas com sensores movidos a evento e experimentos com os dois tipos de sensores juntos. A Tabela 2 mostra os fatores e níveis utilizados nessa avaliação. Cada experimento foi replicado trinta vezes e o intervalo de confiança assintótico foi calculado com nível de confiança de 99%.

Para a realização dos experimentos, foi instanciada a mesma quantidade de sensores movidos por tempo apresentados na Tabela 1. Ou seja, se o experimento é feito com 25.000 sensores e existem cinco tipos diferentes de sensores movidos a tempo, então serão instanciados 5.000 sensores de cada tipo. O envio de mensagens dos sensores movidos por evento é gerado por uma distribuição de Poisson, e o parâmetro λ (lambda) define a taxa de chegada de pessoas a um evento simulado. Foi observado que a taxa de chegada de pessoas em uma evento típico (ex: chegada de pessoas à um estádio) varia em três momentos: a) Início: antes e instantes após o início, com taxa de chegada moderada, uma vez que as pessoas chegam com antecedência ou um pouco atrasados ao evento em tempos diferentes; b) Meio: durante o evento a taxa de chegada é pequena, pois poucos indivíduos entram e saem do evento; c) Fim: após o término do evento a taxa de chegadas é alta pois todos os indivíduos costumam sair ao mesmo tempo. Por exemplo, quando a taxa de chegada é X indivíduos por segundo no Início, no Meio é diminuída para X/10 e no Fim ela assume 3X.

Tabela 2. Fatores e Níveis dos Experimentos

Fator	Níveis
Número de sensores movidos por tempo	0, 25.000, 50.000
Taxa de chegada (por segundo) para sensores movidos por evento	0 – 6 – 12
Configuração do gerenciador de contexto	C1, C2, C3, C4

Como um dos objetivos da avaliação é identificar o impacto de diferentes módulos de software em plataformas de gerenciamento de contexto em ambientes IoT, foram realizados experimentos para avaliar diferentes arquiteturas possíveis (Figura 3). Foram realizados experimentos com e sem o módulo de Fusão e experimentos com o módulo de Inferência desempenhado pelo Drools (C2 e C4) e por uma solução específica *hard coded* por um trecho de código em Java (C1 e C3). A configuração padrão da plataforma avaliada é a C4.

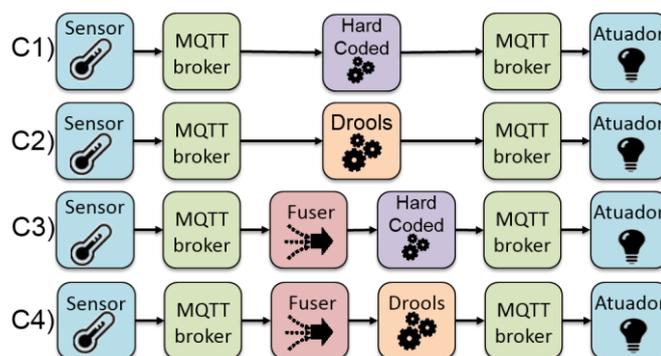


Figura 3. Configurações da arquitetura do gerenciador de contexto

Cada experimento possui um tempo fixo de 5 minutos e visa simular um janela de tempo de um ambiente de cidades inteligentes, com diversos sensores enviando mensagens. Em experimentos nos quais existem sensores movidos a evento, o tempo dos experimentos é dividido em três períodos diferentes de envio de mensagem. O evento simulado acontece em 2/3 do tempo do experimento, ou seja, 3min e 20s.

6 Resultados

Os resultados apresentados nessa seção estão divididos por categoria de sensor, tempo e evento. Não são apresentados resultados referentes à taxa de entrega pois a plataforma não perdeu mensagens em nenhum experimento. As legendas dos gráficos referem-se as configurações do gerenciador de contexto denotadas na Figura 3.

6.1 Impacto de Sensores Movidos por Tempo

Os resultados para o tempo médio que uma mensagem leva para ser processada pelo gerenciador de contexto quando submetido a um cenário somente com sensores movidos por tempo são apresentados na Figura 4. É possível observar o efeito do aumento de carga do número de sensores enviando mensagem ao sistema na plataforma, bem como o impacto desse aumento nas diferentes configurações avaliadas.

Na maioria das configurações, não houve aumento significativo do tempo médio entre os experimentos realizados com 25.000 sensores e 50.000 sensores. Além disso, os intervalos de confiança são pequenos se comparados à média. Além disso, o maior tempo de processamento é cerca de 140 ms, considerado suficiente para que a plataforma interaja com as pessoas em um cenário real.

A única configuração que destoa é sem utilização do módulo de Fusão, mas com o uso do Drools fazendo a inferência de regras (C2). Nesse caso, a plataforma exige poder de processamento do motor de inferência, mas como as otimizações de desempenho do Esper não estão presentes, ocorre um crescimento do tempo de processamento. Esse comportamento repete-se nos experimentos posteriores.

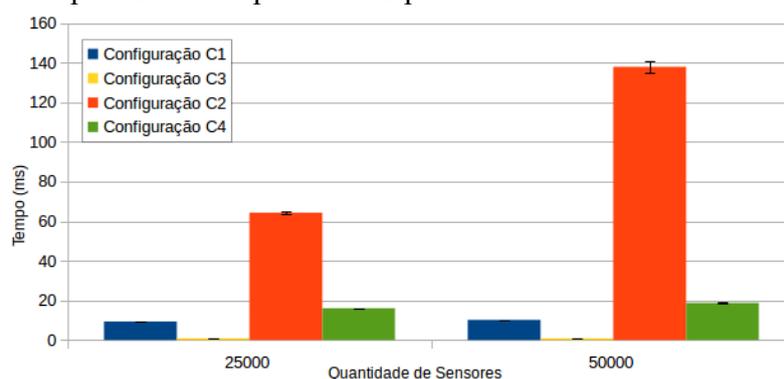


Figura 4. Impacto de sensores movidos por tempo

Analisando a configuração sem a utilização do Drools e sem o módulo de Fusão nota-se bons resultados, já que os módulos que exigem mais processamento foram retirados e implementados com soluções simples e pontuais. Experimentos com apenas o módulo responsável pela fusão e com a solução *hard coded* de inferência de regras apresentam o menor tempo de processamento. Isto é esperado, já que o Esper implementa diversas otimizações de desempenho e não existe outra aplicação que exija processamento, como um motor complexo de inferência de regras.

Experimentos com a plataforma completa (C4) indicam que o gerenciador de contexto é adequado para funcionar com sensores movidos por tempo, apresentando um

baixo tempo de resposta - inferior a 20 ms - em ambas cargas de trabalho. O baixo crescimento na comparação entre 25.000 e 50.000 sensores indica que a plataforma é escalável para sensores dessa categoria.

6.2 Impacto de Sensores Movidos por Evento

A Figura 5 mostra o tempo médio que uma mensagem leva para percorrer a plataforma em um cenário apenas com sensores movidos por eventos. É possível observar o efeito do aumento da taxa de chegada de indivíduos por segundo na plataforma, bem como o impacto desse aumento nas diferentes configurações testadas.

Neste tipo de experimento, o tráfego de mensagens não é constante como nos experimentos envolvendo apenas sensores movidos por tempo. Não houve mudanças significativas quando se aumenta a carga de trabalho, indicando que o sistema é escalável para esse tipo de sensor.

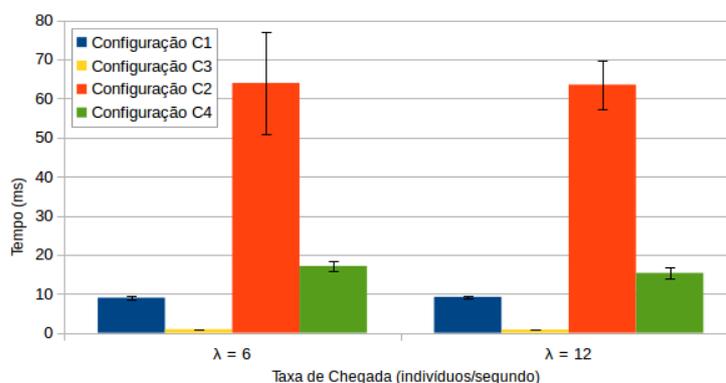


Figura 5. Impacto de sensores movidos por evento

Os experimentos que não utilizam o Esper e utilizam o Drools motor de inferência de regras (configuração C3) se destacam dos demais, inclusive no comportamento nos experimentos com sensores movidos por tempo. O motivo continua sendo as otimizações de desempenho presentes no Esper. O intervalo de confiança maior é esperado devido a pouca regularidade desses sensores quando comparado aos sensores movido por tempo.

Nos demais experimentos obteve-se um tempo de resposta menor do que 20 ms, ideal para esse tipo de sensor que costuma demandar uma ação imediata da plataforma. O intervalo de confiança reduzido implica na pouca variação dos experimentos, indicando uma consistência nesse tempo de resposta, mesmo se tratando de sensores irregulares nos seus envios. O tempo médio de processamento para cada configuração adotada segue o mesmo comportamento observado na seção 6.1. Os experimentos com o módulo de Fusão e o Drools apresentaram um rápido tempo de resposta e um pequeno intervalo de confiança, indicando que a plataforma é capaz de lidar com esse tipo de tráfego e prover uma resposta no tempo adequado.

6.3 Impacto de Sensores Movidos por Tempo e por Evento

Os resultados para o tempo médio que uma mensagem leva para ser processada pelo gerenciador de contexto em um cenário com sensores movidos por evento e tempo são apresentados na Figura 6. Nota-se um destaque dos experimentos com Drools e sem o módulo responsável pela fusão. Isso acontece pois o Drools requer processamento, e nesse caso não há as otimizações presentes no Esper⁵. O tempo de processamento

⁵ <http://www.espertech.com/esper/performance.php>

manteve-se baixo para a maioria dos casos, apresentando um tempo de resposta inferior a 40 ms em todas as configurações. Os valores de tempo de processamento para as diversas configurações testadas seguem o padrão dos experimentos já apresentados.

Observa-se um pequeno crescimento no tempo de processamento conforme a carga de trabalho aumenta nos experimentos com a configuração C4. Esse padrão de crescimento é esperado, porém sua proporção reduzida indica que o sistema é escalável e portanto consegue lidar com diferentes tipos de sensores e prover uma rápida resposta.

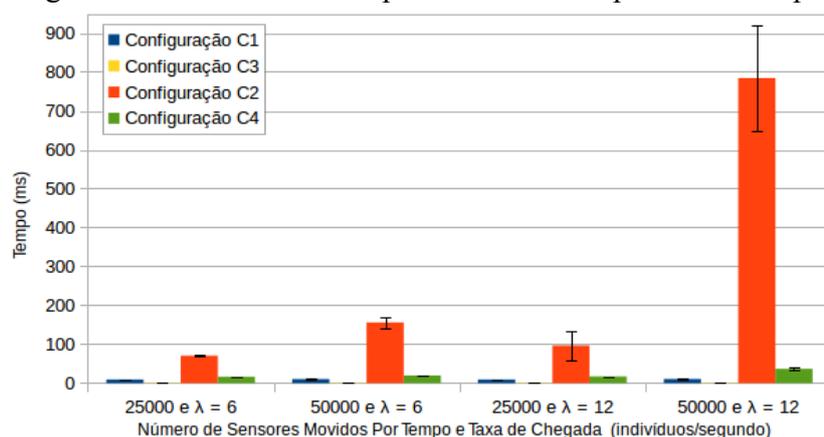


Figura 6. Impacto conjunto de sensores movidos por tempo e por evento

7 Discussão

Existem diversas propostas de sistemas de computação urbana que utilizam Internet das Coisas para controlar dispositivos em tempo real com gerenciamento sensível ao contexto. No entanto, não existem avaliações de desempenho abrangentes desses sistemas, não sendo possível comprovar a escalabilidade dessas plataformas. Alta escalabilidade deve ser uma característica essencial, uma vez que essas tais plataformas se destinam a automatizar ambientes com grande número de sensores e atuadores.

Desenvolvedores devem se atentar a questões de desempenho e escalabilidade nas escolhas referentes a arquitetura de gerenciadores de contexto, além de definir se irão utilizar ferramentas já existentes no mercado ou desenvolverão as próprias soluções específicas, estando cientes que essa escolha pode impactar no desempenho e flexibilidade da plataforma. Os experimentos apresentados evidenciam que diferentes configurações impactam significativamente no tempo de resposta da plataforma. Um exemplo é o impacto do módulo de Fusão, desempenhado pelo Esper, que torna a plataforma mais escalável e rápida. Já o Drools demonstrou ser menos escalável, mas obteve resultados satisfatórios.

Uma das contribuições do projeto é o SenSE, uma ferramenta que possibilita a realização de testes em plataformas desenvolvidas para cenários grandes e complexos de Internet das Coisas, tornando possível avaliar a escalabilidade desses sistemas, identificando possíveis gargalos, antes que sejam implementados em cenários reais.

O gerenciador de contexto avaliado é uma plataforma para automatização predial que visa tornar o sistema energeticamente eficiente. A plataforma se mostrou robusta, não perdendo nenhuma mensagem em nenhum experimento, mesmo com a carga de trabalho alta. Os resultados da avaliação mostram que a plataforma possui um tempo de resposta muito baixo, tanto em ambientes mais regulares (experimentos apenas com sensores movidos a tempo), quanto em ambientes em que o envio de mensagens era irregular (experimentos apenas com sensores movidos a evento). Quando a plataforma

foi submetida a experimentos nos quais os dois tipos de sensores enviaram mensagens, ela apresentou um bom comportamento, mesmo com dezenas de milhares de sensores enviando mensagens. Isso evidencia que o sistema é adequado para funcionar em ambientes de larga escala, onde há uma grande heterogeneidade de dispositivos conectados, uma das mais importantes características de redes IoT.

8 Conclusão

Esse artigo apresenta o impacto de diferentes escolhas de arquitetura em gerenciadores de contexto para redes IoT na escalabilidade e desempenho da plataforma, para isso é apresentado um Gerenciador de Contexto juntamente com uma avaliação quantitativa e qualitativa da plataforma com diferentes configurações. Foi desenvolvida a plataforma SenSE, capaz de simular sensores presentes em uma Cidade Inteligente. Os experimentos realizados com diversas cargas de trabalho mostraram que o gerenciador de contexto avaliado é apropriado para ambientes complexos de IoT, provendo uma rápida reposta quando solicitado. A avaliação de desempenho possibilitou compreender melhor o comportamentos de diversos componentes encontrados em gerenciadores de contexto, principalmente dos responsáveis pela realização de fusão de dados e do processamento de regras. Como trabalhos futuros, pretende-se implementar novas versões da arquitetura do gerenciador de contextos, inclusive distribuídas.

Referências

- Aman, M. et al (2016) “Parallelization in software systems used in wireless sensor networks and Internet of Things: Case study: Middleware systems”. WF-IoT 2016: 342-347.
- Caragliu, A., Del Bo, C., Nijkamp, P., “Smart Cities in Europe”, Journal of Urban Technology, 18:2, pp. 65-82, August 2011.
- Kamienski, C., Borelli, F., Biondi, G., Rosa, W., Pinheiro, I., Zyrianoff, I., Sadok, D., Pramudianto, F. (2015), “Context-Aware Energy Efficiency Management for Smart Buildings”, IEEE World Forum on Internet of Things (WF-IoT 2015).
- Kamienski, C., Jentsch M., Eisenhauer M., Kiljander J., Ferrera E., Rosengren P., Thestrup P., Souto E., Andrade W., Sadok D. (2017), “Application development for the Internet of Things: A context-aware mixed criticality systems development platform”, Computer Communication, vol. 104, pp. 1-16.
- Mongiello, M., et al (2016), “Context-Aware Design of Reflective Middleware in the Internet of Everything”. STAF Workshops 2016: 423-435.
- Osello A., et. al (2013), “Energy saving in existing buildings by an intelligent use of interoperable ICTs”, Energy Efficiency, 6(4), pp. 707-723.
- Paethong, P., Sato, M., Namiki, M. (2016), “Low-power distributed NoSQL database for IoT middleware”, Student Project Conference (ICT-ISPC), 2016 Fifth ICT International.
- Perera, C., et. al. (2014), "Context Aware Computing for The Internet of Things: A Survey", IEEE Comm. Surveys & Tutorials, 16(1), First Quarter 2014.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M. (2014), "Internet of Things for smart cities", IEEE Internet Things J., vol. 1, no. 1, pp. 22-32.
- Zyrianoff, I.; Borelli, F.; Kamienski, C.; “SenSE – Sensor Simulation Environment: Uma ferramenta para geração de tráfego IoT em larga escala”, 2017. SBRC 2017. Salão de Ferramentas.