

Controle e Monitoramento Remoto de Máquinas Industriais por meio de *Smart Gateways* na Web das Coisas*

Paulo Henrique da Silva², Marlon Cordeiro Domenech^{1,2} †, Leonardo Ronald P. Rauta², Rodrigo Cândido da Silva², Michelle S. Wangham^{1,2}

¹Laboratório de Sistemas Embarcados e Distribuídos

²4Vision Lab – Universidade do Vale do Itajaí (UNIVALI) – SC – Brasil

{paulohenrique, marloncdomenech, leonardorauta}@edu.univali.br,

rcandidosilva@gmail.com, wangham@univali.br

Abstract. *Remote monitoring and control are essential in industrial environments. This work describes a remote monitoring and control solution for industrial machines, through the use of Smart Gateways connected to the Internet. Smart Gateway provides machines' resources as RESTful Web Services, sends monitored data to a Cloud-based storage service and also enables monitoring and control using mobile devices and web applications. Proposed solution was evaluated in a case study through the implementation of a prototype. Results obtained in functional and performance tests confirm the feasibility of the proposed solution.*

Resumo. *O controle e o monitoramento são essenciais em ambientes industriais. Este trabalho descreve uma solução de monitoramento e controle remoto de máquinas industriais, por meio de Smart Gateways conectados à Internet. Além de disponibilizar na forma de Serviços Web RESTful os recursos das máquinas, um Smart Gateway envia dados de monitoramento para um serviço de persistência na nuvem e permite o controle e o monitoramento via dispositivos móveis e aplicações Web. O uso da solução proposta em um estudo de caso (máquina real) foi analisado por meio da implementação de um protótipo. Os resultados obtidos nos testes funcionais e de desempenho confirmam a viabilidade da solução proposta.*

1. Introdução

A necessidade de informação sobre as máquinas industriais é de extrema importância para que os gestores das empresas possam tomar ciência do cenário corrente de seu chão de fábrica. [Ramamurthy et al. 2010] afirmam que o controle e monitoramento remotos são alguns dos mais importantes mecanismos para o aumento da produção e que, recentemente, tem havido muito interesse das empresas por implantá-los.

Uma forma de permitir a comunicação entre máquinas industriais heterogêneas, garantindo também o controle e monitoramento remoto destas, é por meio da Internet das

*Projeto financiado pelo CNPq (RHAE 459623/2013-3) e pela Microsoft Research ("PaaS for smart machines monitoring and control").

†Bolsista CAPES.

Coisas (*Internet of Things - IoT*). Segundo [Atzori et al. 2010], a ideia básica da IoT consiste na presença de uma diversidade de coisas (objetos) que interagem e cooperam entre si afim de atingir um objetivo comum, por exemplo, o compartilhamento de informações, utilizando métodos de endereçamento único e protocolos de comunicação padronizados.

A integração entre sensor-atuador-Internet forma a base tecnológica para o conceito de ambientes inteligentes (*smart environments*), nos quais a informação gerada pode ser compartilhada entre diversas plataformas e aplicações, sendo possível o controle e monitoramento de determinados dispositivos [Gubbi et al. 2013]. O conceito de ambientes inteligentes engloba diferentes tecnologias, tais como redes de sensores e sistemas embarcados, trabalhando em conjunto para rastrear estados dos dispositivos, como localização, temperatura e movimentos [Atzori et al. 2010].

A Computação em Nuvem completa o conceito de IoT no intuito de prover o sensoriamento ubíquo [Gubbi et al. 2013]. Em uma Nuvem de Coisas (*Cloud of Things - CoT*), é possível a execução de tarefas computacionais pesadas, pois o processamento e a tomada de decisão são facilitados pelas capacidades computacionais da Nuvem, enquanto que os dispositivos podem focar no sensoriamento e atuação. Em uma CoT, é possível oferecer Sensores como um Serviço e Atuadores como um Serviço [Parwekar 2011].

Máquinas industriais integram um vasto número de sistemas embarcados, conectados ou não a uma rede de comunicação. Estas máquinas podem ter sensores e atuadores que possibilitam operações de monitoramento e controle [Kirubashankar et al. 2009]. Na literatura, se reconhece o potencial da tecnologia de IoT para criação de ambientes industriais inteligentes por meio de *Smart Gateways*. Esse *Smart Gateway* serve como uma ponte para integrar máquinas industriais heterogêneas entre si e com sistemas na Internet.

A maioria dos dispositivos (máquinas industriais) que estão atualmente conectados à Internet utilizam softwares proprietários, o que torna onerosa a criação de aplicações que integram dados e recursos providos por diferentes dispositivos [Guinard et al. 2011]. Para garantir interoperabilidade, existe uma tendência em tratar a Internet das Coisas como Web das Coisas (*Web of Things - WoT*), na qual os padrões e protocolos abertos da Web são empregados para prover o compartilhamento de informação [Zeng et al. 2011].

O objetivo deste artigo é descrever uma solução que visa prover o monitoramento e controle remoto de máquinas industriais, por meio de *Smart Gateways* conectados à Internet. Um *Smart Gateway* é responsável por: (1) disponibilizar, na forma de Serviços Web RESTful, os recursos das máquinas industriais; (2) enviar os dados monitorados para um Serviço Web de persistência de dados na Nuvem; e (3) possibilitar o controle remoto das máquinas industriais por aplicações web hospedadas na Nuvem ou aplicativos de dispositivos móveis.

2. Trabalhos Relacionados

Esta seção apresenta quatro trabalhos relacionados que tratam do monitoramento remoto em ambientes industriais. Estes trabalhos foram analisados considerando as seguintes características: (1) se estes fazem uso de Serviços Web RESTful; (2) se permitem a comunicação *Machine-to-Machine* (M2M); (3) se fazem uso do ambiente Web; (4) se proveem o controle remoto de máquinas industriais; (5) se fazem uso de um serviço de persistência em Nuvem; e, (6) se usam *Smart Gateways*.

Em [Zecevic 1998], é descrita uma solução que permite que usuários acessem os dados obtidos por um sistema SCADA (*Supervisory Control and Data Acquisition*), por meio de um navegador Web. O acesso é possível pois uma aplicação hospedada no servidor Web da Intranet da empresa, ao ser requisitada por um usuário via navegador Web, se conecta ao servidor que armazena os dados do SCADA e os disponibiliza ao usuário. Apesar de ser uma aplicação intermediária entre o usuário e os dados do SCADA, esta aplicação não oferece as mesmas funcionalidades que o *Smart Gateway*. Não há armazenamento de dados históricos na Nuvem e os recursos das máquinas não são disponibilizados para controle remoto, pois apenas dados em tempo real do SCADA estão acessíveis.

Em [Polônia 2011], é apresentada uma arquitetura de software para aplicações de SCADA que utiliza a Web como plataforma. A solução é baseada em uma Arquitetura Orientada a Recursos (*Resource Oriented Architecture - ROA*) que, por meio de Serviços Web RESTful, permite o monitoramento e controle remoto de máquinas (sistema SCADA RESTful). A arquitetura permite uma integração do SCADA com a Web de uma maneira mais transparente e interoperável do que se obtém com sistemas SCADA tradicionais, baseados em RPC (*Remote Procedure Call*) ou em arquiteturas SOA (*Service Oriented Architecture*). Como o SCADA tem acesso direto às máquinas industriais, por meio de seus protocolos específicos, é possível considerar o sistema SCADA RESTful como um *Smart Gateway*. Apesar de possível a comunicação M2M, esta questão não é tratada. Além disso, a integração com a Nuvem não é abordada e a comunicação autônoma entre máquinas heterogêneas não é possível neste cenário.

Em [Mahmood e Al-Naima 2011], é proposto um sistema de monitoramento e controle remoto de plantas industriais petroleiras via Internet. Os equipamentos industriais das diversas plantas petroleiras são conectados a um servidor web central, o qual disponibiliza aos usuários acesso aos recursos de monitoramento e controle destes equipamentos. O acesso é feito por meio do navegador Web do usuário, o qual carrega uma aplicação do tipo *applet* para executar as operações nos equipamentos. Devido ao uso de *applets*, a solução proposta não possibilita a comunicação M2M. Além disso, não há persistência dos dados monitorados para análise de dados históricos.

O projeto IoT@Work tem como objetivo reduzir custos operacionais de configuração, funcionamento e manutenção de soluções fabris, por meio da redução dos tempos de interrupção decorrentes de reconfiguração e mudanças nos sistemas. Assim, foram desenvolvidas tecnologias para viabilizar o conceito de "*plug&work*" em redes industriais, visando permitir o ingresso de equipamentos na rede e a obtenção/fornecimento de funcionalidades sem intervenção humana [IOT@WORK 2013]. Dentre as tecnologias desenvolvidas destacam-se: (i) o serviço de diretório unificado: permite aos dispositivos descobrir e acessar outros dispositivos/serviços da rede por meio de uma API RESTful, viabilizando também seu monitoramento remoto; (ii) O serviço de notificação de eventos: um *middleware* que atua como um conector entre geradores e consumidores de eventos, permitindo a troca de informações eficiente entre os dispositivos; e (iii) o Processamento de Eventos Complexos: permite que dados monitorados disparem processos automatizados de atuação [Imtiaz et al. 2013]. Contudo, apesar de ser focado em ambientes industriais e permitir a comunicação M2M, o projeto não aborda a publicação de dados dos dispositivos em uma plataforma de Nuvem por meio de *Smart Gateways*.

A Tabela 1 compara os trabalhos relacionados e a solução proposta (descrita na

Seção 3). Nenhum dos trabalhos relacionados abordou o uso de Nuvem para a persistência de dados de monitoramento e o provimento de aplicações Web que possibilitem o controle remoto de dispositivos em um ambiente industrial.

Tabela 1. Comparação dos trabalhos

	Zecevic 1998	Polônia 2011	Mahmood e Al-Naima 2011	IoT@Work 2013	Este Trabalho
API REST	Não	Sim	Não	Sim	Sim
M2M	Não	-	Não	Sim	Sim
Ambiente Web	Sim	Sim	Sim	Sim	Sim
Controle Remoto	Não	Sim	Sim	Sim	Sim
Uso de Cloud	Não	Não	Não	Não	Sim
Uso de Smart Gateway	Não	Sim	Não	Não	Sim

3. Solução Proposta

Devido a restrições como segurança, fatores geográficos e custo de instalação e operação, geralmente, máquinas industriais não possuem conectividade com a Internet por meio do protocolo IP (*Internet Protocol*). Nesses casos, é possível a utilização de um dispositivo intermediário entre a máquina e a Internet, o *Smart Gateway*. O *Smart Gateway* é capaz de interpretar as mensagens oriundas das máquinas e as direcionar para o destino desejado, seja este uma base de dados, um servidor ou mesmo outro dispositivo ou máquina.

Máquinas industriais geralmente possuem uma arquitetura já definida, testada e validada, o que dificulta alterações de conectividade. Sendo assim, o software do *Smart Gateway* proposto neste trabalho está embarcado em um hardware separado da máquina, porém, capaz de se comunicar com a mesma. Isso possibilita sua expansão para o funcionamento com diferentes máquinas sem alterar a sua arquitetura ou efetuar retrabalhos.

Na solução proposta, o *Smart Gateway* faz uso dos conceitos de WoT, pois utiliza protocolos Web já difundidos na Internet, o que favorece a interoperabilidade entre os dispositivos e os sistemas finais da Internet. O *Smart Gateway* é responsável por disponibilizar os recursos monitorados por meio de Serviços Web RESTful para os usuários e para outras aplicações na Internet, além de interfacear com cada recurso monitorado na máquina, através da linguagem específica de cada um. Portanto, além de ter o papel de ponte entre recursos e usuários ou aplicações na Internet, a solução proposta abstrai os recursos presentes na máquina para usuários ou aplicações que precisam acessá-los.

O uso de Serviços Web RESTful se justifica devido a interoperabilidade e a diminuição da complexidade entre as partes de uma aplicação distribuída através do uso das características inerentes ao protocolo HTTP, como autenticação, autorização, criptografia, compressão e *caching*. Além disso, no estilo arquitetural REST, o mapeamento dos métodos básicos do protocolo HTTP, como o *Get*, *Post*, *Put* e *Delete* como métodos de aplicação visam o aumento da simplicidade e redução do acoplamento entre as pontas do sistema distribuído.

Devido à utilização de Serviços Web RESTful para envio desses recursos do *Smart Gateway* para a Nuvem, foi adotado o padrão JSON, em função da sua formatação ser em mensagens de texto, o que facilita sua manipulação. Além disso, este também possui

uma sintaxe menos verbosa que o padrão XML e é facilmente integrada em diversas linguagens de programação (p.e. JavaScript). Segundo [Nurseitov et al. 2009], o padrão JSON utiliza menos recursos computacionais se comparado com o padrão XML, o que o torna mais leve para ser executado em sistemas embarcados.

A Figura 1 apresenta a solução proposta, a qual é composta por um *Smart Gateway* capaz de enviar, periodicamente, os recursos de uma máquina industrial para um Serviço Web RESTful na Nuvem (responsável por persistir estes dados) e disponibilizá-los para acesso por usuários e outras máquinas, além de permitir a atuação nos recursos desta máquina. Na Figura 1 são mostrados os componentes do *Smart Gateway*, a comunicação deste com a máquina industrial, o envio dos recursos da máquina para o Serviço Web na Nuvem e o acesso de aplicativos de dispositivos móveis a esses recursos.

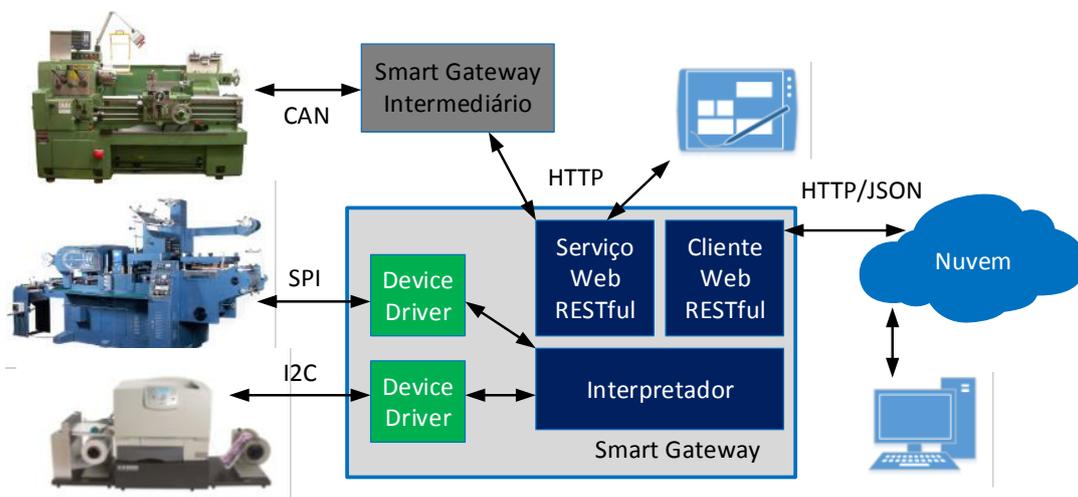


Figura 1. Solução proposta

A comunicação entre o *Smart Gateway* e a máquina industrial pode ser feita utilizando diferentes protocolos. A escolha do protocolo de comunicação depende da conectividade da máquina industrial com o mundo externo, por meio de um protocolo padrão (p.e. I2C e SPI) ou proprietário. Para facilitar a interoperabilidade entre as máquinas e o *Smart Gateway*, este possui diversos módulos à nível de sistema operacional (SO), responsáveis por abstrair tais protocolos. Esses módulos podem ser caracterizados como *Device Drivers*, pois realizam a comunicação entre o SO e a máquina (*device*), conforme mostra a Figura 1.

Para que o *Smart Gateway* seja capaz de disponibilizar e enviar os recursos da máquina para o Serviço Web na Nuvem, este precisa de um Serviço Web RESTful e de um Cliente Web RESTful. O Serviço Web RESTful é responsável por disponibilizar os recursos da máquina, os quais podem ser requisitados por meio do método HTTP GET e também permite a atuação na máquina por meio do método HTTP POST. Já o Cliente de Serviço Web RESTful é responsável por enviar os recursos para um serviço de recebimento de dados na Nuvem, utilizando para isso o método HTTP POST. Esse envio é feito periodicamente e pode ser parametrizado conforme a necessidade da aplicação.

Para disponibilizar os recursos da máquina (sensores, atuadores e estados), é ne-

cessária uma representação virtual desta. Essa representação é criada pelo *Smart Gateway* com base em informações enviadas pela própria máquina, as quais são interpretadas com base em uma semântica específica (p.e. os processos que a máquina executa). Deste modo, a aplicação que interpreta as informações da máquina e atualiza a representação virtual desta é chamada de Interpretador, conforme mostrado na Figura 1. Assim, é possível que o *Smart Gateway* conheça os recursos que precisam ser disponibilizados para a *Cloud* e para as aplicações dos dispositivos móveis.

Assim, a cada alteração nos recursos da máquina e, conseqüentemente, na representação virtual que o *Smart Gateway* mantém desta, o *Smart Gateway* gera um registro de que determinado recurso foi alterado, precedido por um *timestamp* para indicação de quando o evento ocorreu. Esses registros são armazenados em memória e, após um tempo configurado pelo administrador, são enviados para o Serviço Web de persistência de dados na *Cloud*.

O acesso aos recursos da máquina na Nuvem permite, por exemplo, o monitoramento da máquina e emissão de relatórios sobre o seu uso e identificação de possíveis gargalos na linha de produção. Já o acesso aos recursos da máquina por meio de dispositivos móveis, feito diretamente no *Smart Gateway* permite, por exemplo, monitorar o funcionamento da máquina em tempo real.

Por meio do monitoramento da máquina, além de possibilitar ao gerente de produção a emissão de relatórios e identificação de gargalos em sua linha de produção, o fabricante da máquina também pode utilizar a solução para identificar possíveis pontos de falha de seus equipamentos, acelerar o processo de manutenção da máquina e também gerenciar seu processo de garantia, sendo possível verificar se ela foi utilizada de forma incorreta, ou não. Além de poder atuar remotamente na máquina.

A solução proposta permite a comunicação autônoma (M2M) entre máquinas industriais no chão de fábrica. Um exemplo seria o caso de duas máquinas industriais A e B, em que A depende de B para realizar seu trabalho. Ao perceber um problema de funcionamento, B avisa A sobre este. A é capaz de inferir a respeito das conseqüências do problema e pode decidir em reduzir seu ritmo de trabalho, para economia de energia. Para viabilizar o cenário M2M, a solução proposta considera que o *Smart Gateway* pode ser colocado com duas funções no ambiente industrial: (i) representar algumas máquinas na Web ou (ii) representar apenas uma máquina. A Figura 1 apresenta as duas situações, em que um *Smart Gateway* intermediário representando apenas uma máquina (à esquerda) se comunica com o *Smart Gateway* que representa duas máquinas (acessíveis via I2C e SPI). Ambos os *Smart Gateways* comunicam-se entre si (M2M) por meio de Serviços Web RESTful. Isso permite utilizar diversos *Smart Gateways* para gerir o chão de fábrica.

4. Estudo de Caso e Implementação

Para verificar a viabilidade da solução proposta, uma prova de conceito foi desenvolvida por meio de um protótipo do *Smart Gateway* conectado a uma máquina industrial real. Conforme mostrado na Figura 2, a máquina industrial já possui um sistema SCADA (instalado em um notebook que a acompanha) que tem acesso ao hardware da máquina. O SCADA gera um arquivo de log que descreve todas as atividades realizadas pela máquina, sejam estas automáticas ou provenientes de comandos do operador. Para não alterar a arquitetura e o funcionamento da máquina industrial, foi criada uma aplicação (chamada

Monitor), que acessa os arquivos de log e faz a interface entre o *Smart Gateway* e o SCADA, semelhante ao *Device Driver* descrito na Figura 1.

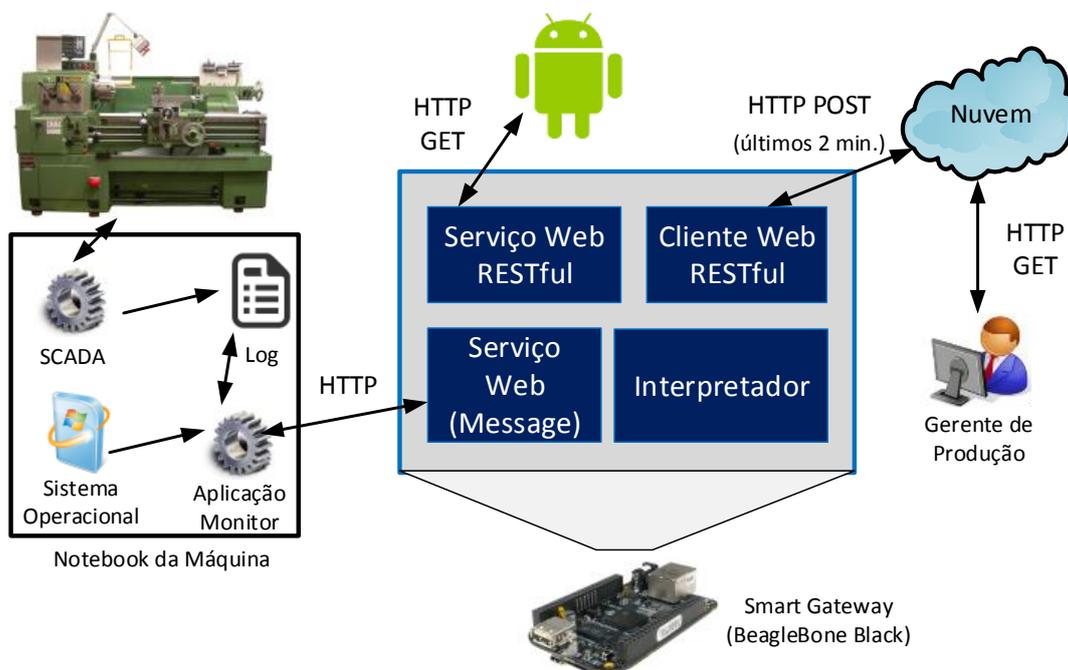


Figura 2. Cenário do Estudo de Caso

Toda vez que uma nova operação é realizada pela máquina ou ocorre qualquer alteração de configuração, é criado um novo registro no log. Diante disso, a aplicação Monitor recebe um aviso do SO do notebook informando que o arquivo do log foi alterado, conforme é mostrado na Figura 2. A aplicação Monitor obtém este novo registro no arquivo de log e o envia ao *Smart Gateway* por meio de uma mensagem HTTP POST para um Serviço Web de recebimento das mensagens de log (chamado *Message* na Figura 2). Vale ressaltar que a aplicação Monitor foi desenvolvida em função da restrição do ambiente do estudo de caso. Caso a máquina em questão não tivesse um notebook e um SCADA, seria possível utilizar interfaces como USB, CAN, I2C, dentre outras, para comunicação direta entre o *Smart Gateway* e a máquina industrial.

Ao receber uma mensagem da aplicação Monitor, o *Smart Gateway* a interpreta (por meio do Interpretador) e monta a representação virtual da máquina, composta pelos seus recursos (sensores, atuadores e estados). O *Smart Gateway* possui conhecimento dos processos (e seus estados) que são executados pela máquina e como ocorrem as suas transições, o que permite a este inferir o significado de uma mensagem do SCADA dentro de um contexto de operação específico.

Para obter algum recurso da máquina em tempo real, pode-se (i) fazer uma consulta ao recurso no *Smart Gateway* (via HTTP GET) usando, por exemplo, uma aplicação em um dispositivo móvel no chão de fábrica ou (ii) fazer a escrita (via HTTP POST) de algum recurso da máquina, permitindo que esta seja operada remotamente. O *Smart Gateway* também mantém um registro histórico das alterações nos recursos da máquina.

A cada alteração na representação virtual é gerado um registro precedido de um *timestamp*. Esses registros são armazenados em memória e, após 2 minutos, são enviados para o Serviço Web de persistência de dados na *Cloud*, conforme mostrado na Figura 2. Com base nestes dados, aplicações de *Data Mining* e diferentes tipos de SaaS (*Software as a Service*) para manipulação e visualização destes dados podem ser construídos.

O *Smart Gateway* e o Monitor foram desenvolvidos usando a linguagem Java, em sua versão 7 *update 45*, específica para ambientes embarcados que utilizam processadores ARM. A escolha do JAVA deu-se devido à portabilidade que este oferece. Como servidor de aplicação foi utilizado o Apache Tomcat na versão 8.0.9. Para implementação dos clientes e Serviços Web RESTful foi utilizado o framework Jersey, a implementação de referência em Java para APIs RESTful. O *Smart Gateway* foi embarcado em um BeagleBone Black, por ser uma plataforma de hardware aberta que atende às necessidades do projeto. Como plataforma de *Cloud* para hospedagem do Serviço Web de recebimento de dados de monitoramento, foi utilizada a plataforma Windows Azure.

5. Avaliação dos Resultados

Para avaliar o protótipo desenvolvido, foram executados dois tipos de testes: (i) testes funcionais de software e (ii) de desempenho. O cenário dos testes é composto pela aplicação Monitor instalada em um notebook, que está conectado ao *Smart Gateway* por um cabo USB 2.0. O *Smart Gateway* está conectado à Internet por meio de uma conexão FastEthernet. O SO utilizado no *Smart Gateway* foi o Angstrom¹ Linux.

O primeiro conjunto de testes verificou o atendimento aos requisitos funcionais e foi composto por um total de doze casos de testes com execuções reais da máquina, os quais foram executados corretamente. Por meio desses testes, foi possível garantir que a semântica específica dos processos de operação da máquina industrial estava sendo reproduzida corretamente na representação virtual da máquina que o *Smart Gateway* possui.

A etapa inicial dos testes de desempenho consistiu na identificação da frequência de geração de mensagens do SCADA. Verificou-se que essa geração não é determinística, sendo necessária uma aproximação por meio de uma função de densidade de probabilidade. Para isso, foram geradas 2040 mensagens de log de operações reais do SCADA e foi extraída a quantidade de ocorrências da geração de n mensagens/s. Desconsiderando os espaços de tempo em que nenhuma mensagem é gerada, em 79,29% dos casos o SCADA gerava 1 mensagem/s e, em 14,53% dos casos, eram geradas 2 mensagens/s.

Em seguida, foram executados testes visando mensurar o custo computacional de embarcar o software do *Smart Gateway* em uma BeagleBone Black (*footprint*). Os resultados dos testes são apresentados na Tabela 2. Foi mensurado o consumo de memória RAM do SO, da aplicação do *Smart Gateway* e do *Web Container*, bem como o consumo de memória secundária (no caso do BeagleBone Black uma memória *flash*) da aplicação do *Smart Gateway* e do SO². Conforme os dados da Tabela 2, é possível perceber que um hardware como o BeagleBone Black possui memória RAM e memória secundária suficientes para embarcar a solução proposta, uma vez que possui 4GB de memória *flash* e 512MB de memória RAM.

¹<http://www.angstrom-distribution.org/>

²Os testes realizados não permitiram identificar com precisão o consumo de memória flash atribuída ao Container Web.

Tabela 2. Consumo de memória RAM e memória secundária no *Smart Gateway*

	Memória RAM (MB)	Memória Secundária (MB)
Sistema Operacional	73	1330
Container Web	28	-
Apl. Smart Gateway	121	8

Tabela 3. Tempo de execução do *parsing* e o envio e recebimento de mensagens

	Parse	Envio de Mensagem
Média (ms)	6,31	37,01
Desvio Padrão	3,39	20,81

Ainda no escopo dos testes de desempenho, foram avaliados: (i) o tempo que o *Smart Gateway* levou para interpretar as mensagens recebidas do Monitor e alterar a representação virtual da máquina e (ii) o tempo de envio de uma mensagem do Monitor para o *Smart Gateway* e a respectiva resposta (logo, o teste (i) está contido no teste (ii)). Ambos os testes foram executados 160 vezes com uma taxa de envio de 1 mensagem/s, sendo extraídos a média e o desvio padrão dessas execuções. A Tabela 3 apresenta os resultados desses testes.

Conforme a Tabela 3, o tempo de interpretação das mensagens enviadas pelo Monitor foi, em média, de 6,31ms, enquanto que o desvio padrão foi de 3,39. No segundo caso de teste de desempenho, é considerado o tempo para o envio da mensagem de log do Monitor para o *Smart Gateway*, o processo de interpretação e a resposta do *Smart Gateway* para o Monitor. A média de 160 execuções foi de 37,01ms e o desvio padrão foi de 20,81. Os resultados para ambos os testes podem ser considerados aceitáveis, uma vez que o SCADA gera, na maioria das vezes (93,82%), entre 1 e 2 mensagens/s.

Desse modo, ao comparar os resultados obtidos nos testes de desempenho com a condição real do ambiente industrial do estudo de caso, percebe-se que a solução proposta é viável para o cenário avaliado, diante da carga à que foi submetida a aplicação.

6. Conclusão

Disponibilizar os recursos de máquinas industriais via Internet visa facilitar o controle e o monitoramento destas e, com isto, aprimorar os processos industriais. Devido ao fato de muitas máquinas não possuírem conectividade com a Internet, o uso de *Smart Gateway* se mostra muito interessante. A solução proposta integra conceitos de IoT, computação em nuvem e automação de máquinas industriais e inova ao possibilitar, via *Smart Gateway*, o controle e o monitoramento remoto de máquinas industriais, o uso de serviço de persistência de dados baseado em Nuvem e o suporte a comunicação M2M.

Como trabalho futuro, pretende-se avaliar o desempenho do *Smart Gateway* diante das métricas de uso de CPU, consumo de energia elétrica, tamanho das mensagens transmitidas na rede e vazão da rede (*throughput*). Pretende-se ainda desenvolver uma solução que permita a criação e implantação dinâmica de aplicações Web customizadas para monitoramento e controle remoto de máquinas industriais. Ainda, é ressaltada a importância de tratar questões como segurança e confiabilidade de informações enviadas e recebidas pelas máquinas. Visto que sistemas industriais possuem requisitos mais

relacionados a sistemas de tempo real, do que sistemas de propósito geral, é preciso identificar os requisitos para controle (latência, completude e corretude), que são necessários na atuação sobre equipamentos industriais. Além disso, é importante para a confiabilidade das máquinas a existência de uma arquitetura de segurança focada principalmente nas questões de autenticação e de autorização, tanto de dispositivos quanto de pessoas.

Referências

- Atzori, L., Iera, A., e Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15):2787–2805.
- Gubbi, J., Buyya, R., Marusic, S., e Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.
- Guinard, D., Trifa, V., Mattern, F., e Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In Uckelmann, D., Harrison, M., e Michahelles, F., editors, *Architecting the Internet of Things*, pages 97–129. Springer Berlin Heidelberg.
- Intiaz, J., Dürkop, L., Trsek, H., Givehchi, O., Wisniewski, L., Shrestha, G. M., Zanella, A., Vivo, G., Rotondi, D., Piccione, S., Comolli, M., Corvino, F., Houyou, A., Fischer, K., Hans-Peter, H., Joachim, W., J., C., Periorellis, P., Kloukinas, C., Mahbub, K., e Krotsiani, M. (2013). Iot@work - d2.5 – integrated secure plug&work framework.
- IOT@WORK (2013). Project overview. <https://www.iot-at-work.eu/>. [Online; accessed 10-December-2014].
- Kirubashankar, R., Krishnamurthy, K., e Indra, J. (2009). Remote monitoring system for distributed control of industrial plant process. *Journal of Scientific and Industrial Research*, 68(10):858.
- Mahmood, M. K. e Al-Naima, F. M. (2011). An internet based distributed control systems: A case study of oil refineries. *Energy and Power Engineering*, 3:310–316.
- Nurseitov, N., Paulson, M., Reynolds, R., e Izurieta, C. (2009). Comparison of json and xml data interchange formats: A case study. *CAINE 2009*, 9:157–162.
- Parwekar, P. (2011). From internet of things towards cloud of things. In *Computer and Communication Technology (ICCCT), 2011 2nd International Conference on*, pages 329–333.
- Polônia, P. V. (2011). Proposta de arquitetura orientada a recursos para scada na web. Master's thesis, Universidade Federal de Santa Catarina.
- Ramamurthy, B., Bhargavi, S., e ShashiKumar, R. (2010). Development of a low-cost gsm sms-based humidity remote monitoring and control system for industrial applications. *International Journal of Advanced Computer Science and Applications*, 1(4).
- Zecevic, G. (1998). Web based interface to scada system. In *Power System Technology, 1998. Proceedings. POWERCON '98. 1998 International Conference on*, volume 2, pages 1218–1221 vol.2.
- Zeng, D., Guo, S., e Cheng, Z. (2011). The web of things: A survey. *Journal of Communications*, 6(6).