

PROPOSTA DE ARQUITETURA ORIENTADA A MICROSERVIÇOS PARA APLICAÇÕES DE INTERNET DAS COISAS INDUSTRIAL

JEFERSON A. BIGHETI, SÉRGIO L. RISSO, MARCOS R. CALDIERI

SENAI – Lençóis Paulista - SP

E-mails: jefersonbigheti@terra.com.br, sergio.risso@sp.senai.br, mrcaldieri@gmail.com

EDUARDO P. GODOY

Universidade Estadual Paulista (Unesp), Sorocaba - SP

E-mails: epgodoy@sorocaba.unesp.br

Abstract — The Industry 4.0 (I4.0) is a new concept of production that requires a jointly use of industrial automation and information technologies in order to obtain greater efficiency, quality and productivity. The I4.0 implementation is tied to the application of recent technologies such as the Industrial Internet of Things (IIoT) and Cloud Computing. In this context, the paradigm of collaborative automation through the use and sharing of services provides a flexible, distributed and integrated architecture using communication networks. This paper focuses on the proposal of a cloud-based service-oriented architecture (SOA) for IIoT applications. A review about SOA and its industrial application is presented, as well as on the composition of services. The proposed microservices oriented architecture is discussed, considering its characteristics, operational details, advantages and potential of application in the area of IIoT.

Keywords – Cloud computing, Microservices, SOA, Networked Control System.

Resumo — A Indústria 4.0 (I4.0) é um novo conceito de produção que demanda o uso conjunto de tecnologias de automação industrial e da informação visando a obtenção de maior eficiência, qualidade e produtividade. A implementação da I4.0 está atrelada à aplicação de tecnologias recentes como a Internet das Coisas Industrial (IIoT) e a Computação em Nuvem. Nesse contexto, o paradigma da automação colaborativa através do uso e compartilhamento de serviços proporciona uma arquitetura flexível, distribuída e integrada através de redes de comunicação. Este trabalho foca na proposta de uma arquitetura orientada a serviços (SOA) em nuvem para aplicações de IIoT. Uma revisão sobre SOA e sua aplicação industrial é apresentada, bem como sobre a composição de serviços. A arquitetura baseada em microserviços proposta é discutida, considerando suas características, detalhes operacionais, vantagens e potenciais de aplicação na área de IIoT.

Palavras-chave — Computação em nuvem, Microserviços, SOA, Sistema de controle via rede

1 Introdução

As indústrias estão evoluindo para melhorar seus processos e dessa forma obter uma maior eficiência e qualidade dentro de menores custos e tempos. As máquinas estão incorporando capacidade de comunicação entre si e conosco utilizando o conceito da Internet das Coisas (IoT). Expressões como Big Data, descentralização, virtualização, digitalização e IoT Industrial (IIoT) são conceitos relacionados à Indústria 4.0, também chamada de Quarta Revolução Industrial, a qual representa uma evolução dos sistemas produtivos atuais a partir da convergência entre novas tecnologias de automação industrial e tecnologia da informação (TI) (LU, 2017). Na Indústria 4.0 a maioria das tecnologias necessárias para a sua implementação já existem. Entre elas podem-se citar o protocolo IPv6, os sistemas ciber-físicos (CPS – *Cyber Physical System*), o RFID, o uso de virtualização de sistemas e serviços a partir de softwares, a IoT, a Computação em nuvem (*Cloud Computing*) e Big Data (ZUEHLKE, 2010; STANKOVIC, 2014, LEE et al., 2015, HEGAZY & HEFEEDA, 2015). O grande desafio, portanto, é promover a integração entre essas tecnologias, visando a obtenção de uma nova realidade produtiva, onde tudo estará conectado para que as melhores decisões de produção, custo e segurança sejam tomadas, sob demanda e em tempo real.

Muitas pesquisas têm focado o desenvolvimento nesse sentido com soluções que permitam a integração dos equipamentos da automação a serviços de TI que estão armazenados na nuvem. Hegazy et al (2015) apresentam um novo serviço na nuvem aplicado em automação industrial onde os controladores são hospedados em dois servidores em nuvem, fisicamente separados. Nesta proposta, diversas aplicações como controle em malha fechada e virtualização de controladores são realizadas na nuvem. Nikolaidis et al. (2015) propõem a interação entre dois conceitos: a utilização de controladores nas nuvens com dispositivos baseados em IoT. A proposta de Chen et al. (2010) apresenta a utilização de *Robot as a Service* ou RAAS, sendo este um serviço na nuvem para acesso a hardware e software de um robô. Em Wu et al. (2013), o qual mostrou em seu estudo, o conceito de Cloud Manufacturing, que é um modelo de produção utilizando a definição de computação em nuvem. Por fim Jammes et al (2014) descreve as Arquitetura Orientada a Serviço (SOA) promissoras que podem ser utilizadas em Processo de Monitoramento e Controle com uso do protocolo como CoAP e OPC UA entre outros para acesso a sensores e atuadores.

Este artigo apresenta uma proposta de Arquitetura Orientada a Microserviços (MOA) com o framework *Molecular* para automação de processos com o uso da tecnologia IIoT. O grande desafio desta abordagem é

promover a integração entre as tecnologias da automação com as de TI, equipamentos e sistemas alocados em diferentes níveis hierárquicos dos sistemas industriais. Nesse sentido, um paradigma recente tem sido o da automação colaborativa através do uso e compartilhamento de serviços para obtenção de uma arquitetura flexível, escalável, interoperável, distribuída e totalmente integrada através de redes industriais (JAMES et al., 2014). Diante desse contexto, esta pesquisa foca no estudo e o desenvolvimento de uma arquitetura em nuvem baseada em serviços para aplicações de IIoT.

2 Arquitetura Orientada a Serviços (SOA) para Automação Industrial

Soluções baseadas em Ethernet Industrial permitem ao usuário a integração dos equipamentos industriais a serviços de TI fornecidos por servidores (DECOTIGNIE, 2005). Dessa forma, tornou-se possível disponibilizar informações provenientes dos sistemas de automação através de protocolos com suporte ao Ethernet TCP/IP em aplicações Web. Essa interconexão de dados e infraestruturas de rede de TI e de automação de forma segura e confiável representa uma tendência para o desenvolvimento de soluções inovadoras e integradoras na área industrial com a IIoT e a Indústria 4.0 (LU, 2017).

Processos industriais, bem como muitas outras infraestruturas críticas dependem de sistemas SCADA e SDCD para executar suas funcionalidades complexas. A IIoT é complementar a esses sistemas, utilizando as informações provenientes dos sistemas SCADA e SDCD como fontes de dados. Sistemas SCADA tem foco em monitoramento e controle, enquanto a IIoT busca analisar os dados para aumentar a produtividade e eficiência (KULKARNI, 2016). Colombo et al. (2014) discutem o conceito de uma arquitetura de automação baseada no uso da Computação em Nuvem e de serviços (Web Services) para comunicação e realização das tarefas entre os diferentes componentes e equipamentos de um sistema industrial. Os autores atestam que arquiteturas orientadas a serviço (SOA) são consideradas um caminho promissor para concepção do modelo de fábrica do futuro.

No entanto, existe a necessidade e potencial para melhorar as funcionalidades e minimizar problemas de integração através de abordagens colaborativas de sistemas e serviços (COLOMBO et al., 2014). Nesse sentido é importante considerar os próximos passos para a evolução dos sistemas industriais através da IIoT e Indústria 4.0, de forma que consigam atender os desafios recentes como grau de descentralização e independência dos sistemas. Nesse sentido, a proposta de arquiteturas SOA possui grande potencial, a partir da disponibilização de serviços em nuvem alocados em diferentes dispositivos, gateways ou sistemas, os quais facilitam e padronizam as interações entre eles. Leitão et al. (2016) apresentam um estudo sobre novas arquiteturas SOA para aplicações industriais.

A evolução das arquiteturas de automação e controle industriais, Figura 1, evidencia a tendência para a convergência do uso das tecnologias de automação e da informação. A primeira arquitetura era centralizada e com comunicação ponto a ponto entre equipamentos. A segunda e terceira incorporaram o conceito de redes de comunicação, permitindo a distribuição do controle, redução de custos e facilidade de comissionamento e manutenção. A terceira incorporou as chamadas arquiteturas de controle via redes (NCS), com o uso de redes cabeadas e sem fio em aplicações de controle e provendo a descentralização do controle, modularização e flexibilização.

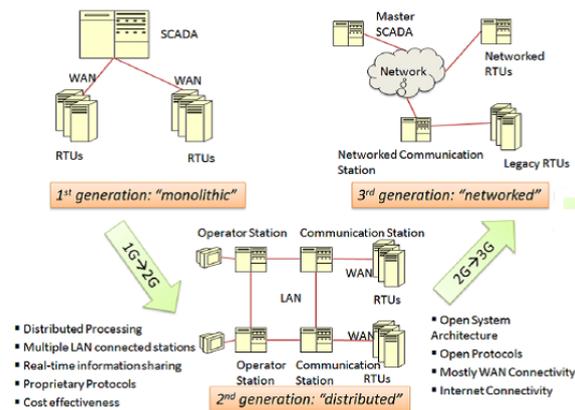


Figura 1 – Evolução das Arquiteturas de Automação e Controle (Colombo et al., 2014)

A Figura 2 ilustra a quarta e atual arquitetura é baseada nos sistemas ciber-físicos (CPS) avançados pela total integração demandada pela IIoT e Indústria 4.0. Nessa nova proposta de arquitetura, um sistema de armazenamento de informações em nuvem será compartilhado pelos equipamentos e sistemas, tornando possível o uso de serviços (Web Services) padronizados para comunicação entre os mesmos.

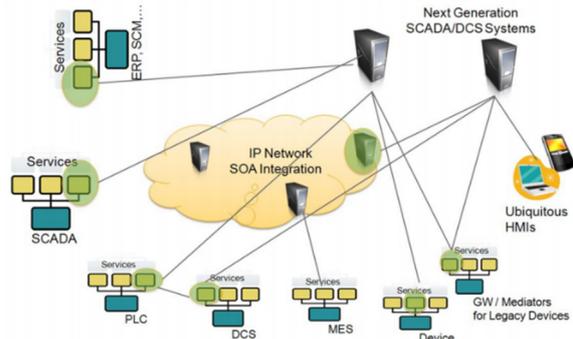


Figura 2 – Nova Arquitetura de Automação e Controle (Colombo et al., 2014)

Os serviços podem ser acessados por aplicações, sistemas e outros serviços independentemente de sua alocação (hardware, software e rede de comunicação), propiciando uma arquitetura colaborativa. Portanto, a consolidação das tecnologias de automação e TI compatíveis com a ideia de uma arquitetura SOA em nuvem para aplicações IIoT norteiam esta pesquisa.

3 Arquiteturas Orientadas a Serviços (SOA)

No contexto computacional, os Serviços são módulos de negócio que possuem interfaces de conexão que são invocadas via mensagens (DUSTDAR e PAPAZOGLU, 2008). Essas interfaces disponibilizam recursos sem que a implementação do serviço seja conhecida. Na tecnologia da informação os serviços são virtuais. Esses serviços são processos independentes que são descritos, disponibilizados, localizados e invocados por meio de redes de comunicação. Essas inovações dos serviços por meio de redes de comunicação variam de simples respostas a requisições de processos até a execução de procedimentos de negócios complexos, os quais requerem uma comunicação entre provedores e consumidores de serviços virtuais.

Dustdar e Papazoglou (2006) afirma que os serviços virtuais estão associados à computação orientada a serviços, onde esses serviços são elementos fundamentais para o desenvolvimento de aplicações e soluções que utilizam as plataformas de computação distribuída. Os serviços virtuais permitem que as organizações exponham suas principais competências programaticamente através da Internet (ou Intranet) usando linguagens e protocolos padrões como XML ou JSON por meio de redes de comunicação.

Muitas pesquisas têm focado no desenvolvimento de arquiteturas orientadas a serviços (*SOA – Service Oriented Architecture*) com a promessa de oferecer uma arquitetura para suportar a propagação e a utilização de serviços virtuais reutilizáveis (XIAO, et al., 2016; JAMMES et al 2014). Para Sheng et al. (2014), a SOA é uma arquitetura para organizar infraestruturas e aplicações em um conjunto de serviços passíveis de interação. Neste contexto, as informações e os recursos ficam disponíveis para todos os clientes como serviços independentes que são acessados de um modo padrão. Nas pesquisas de Souit (2013) e Sheng et al., (2014), a SOA estabelece uma arquitetura que permite que os serviços sejam publicados, descobertos e consumidos por aplicações ou outros serviços. Para Xiao et al. (2016), a SOA é a integração e a colaboração de recursos distribuídos com o conceito de serviço. Na SOA os serviços podem ser invocados de forma independente por qualquer cliente (externos ou internos) para processar funções ou trabalhar de forma coordenada para desenvolver novas funcionalidades para os processos existentes.

Os Serviços também podem ser construídos considerando a invocação de outros serviços existentes. Esta abordagem é conhecida como composição de serviços, apontada como um dos principais aspectos da computação orientada a serviço que contribui para o reuso dos serviços. Segundo Dustdar e Papazoglou (2008) as técnicas mais utilizadas para composição de serviços são a orquestração e a coreografia.

3.1 ORQUESTRÇÃO DE SERVIÇOS

A orquestração representa a composição de serviços para criar um novo serviço ou para resolver uma tarefa de um processo de negócio. Neste caso, sempre

há a figura de um processo central (mestre), sendo um serviço ou uma atividade de negócio que controla e coordena a chamada de outros serviços para compor uma função de maior granularidade. Neste tipo de composição de serviço, cada serviço participante não tem conhecimento de que faz parte de uma composição de serviço de maior hierarquia, com exceção do processo mestre. Somente o processo mestre detém a inteligência sobre a aplicação completa, e a execução é, portanto, centralizada. Para Sheng et al. (2014), a orquestração pode ser considerada uma abordagem baseada no modelo workflow, onde um processo central controla o fluxo de dados entre nós e os serviços (*Web Services*) envolvidos, coordenando a execução das diferentes operações.

Apesar de facilitar a implementação, administração e monitoramento, orquestração possui limitações de escalabilidade. A orquestração de serviços Web se dá por meio do padrão WS-BPEL (*Web Services – Business Process Execution Language*) que também inclui o gerenciamento de transações entre serviços, incluindo a manipulação de erros, assim como a descrição de todo o processo. Em Souit (2013), a orquestração através de SOA fornece flexibilidade da aplicação considerando as características heterogêneas dos componentes de um sistema produtivo.

3.2 COREOGRAFIA DE SERVIÇOS

Na coreografia, os serviços já estão pré-determinados antes da sua execução. Quando um serviço é acionado e envia uma mensagem, outros serviços podem estar programados antecipadamente para receber ou não essa mensagem e dispararem outras ações. Na composição de serviços por coreografia, não existe a figura de um processo mestre que controla e coordena os demais processos. Neste tipo de composição, cada processo envolvido tem o conhecimento de que faz parte de uma composição de serviços (ex: aplicação) e que precisa interagir com outros serviços/aplicações de maneira ordenada para que a composição resultante tenha sucesso. Cada serviço sabe quando atuar, com quais outros serviços interagir, quais operações deve executar e quais mensagens deve trocar, sendo a execução, portanto, descentralizada.

Fattori et al. (2011) apresentam a coreografia de serviço como colaborativa e permite que cada parte envolvida possa descrever seus serviços na interação. A coreografia representa uma descrição global do comportamento de cada um dos serviços participantes da interação, o que é definido pela troca pública de mensagens, regras de interação e acordos entre dois ou mais processos de negócios (SHENG et al., 2014).

Para Souit (2013), a coreografia é a interação entre componentes distribuídos sem a existência de uma entidade controlando a lógica de colaboração, como acontece no mecanismo de orquestração. A coreografia é normalmente associada com interações que ocorrem entre múltiplos serviços Web. A coreografia é suportada pelo padrão WS-CDL (*Web Services Choreography Description Language*) (SHENG et al., 2014).

3.3 MICROSERVIÇOS

Variante da SOA, os microserviços são um estilo arquitetônico em que as aplicações são decompostas em serviços acoplados, oferecendo modularidade, tornando as aplicações mais fáceis de desenvolver, testar, implantar e, o mais importante, alterar e manter. Fowler (2014) descreve todas as características que definem a estrutura de microserviço. Xiao et al. (2014) descrevem um microserviço como um aplicativo flexível, escalável, interoperável, distribuído e totalmente integrado através de redes.

A arquitetura baseada em microserviços (MOA) surge como uma alternativa ao tradicional padrão arquitetural monolítico. Newman (2015) define os microserviços como pequenos serviços autônomos que trabalham juntos. Esses serviços rodam em seus próprios processos e se comunicam através de mecanismos de comunicação de mensagens, como o *Representational State Transfer (REST)*. Os microserviços são implantados e escalados de forma independentes, podem ser escritos em diferentes linguagens e utilizam diferentes recursos para armazenamento de dados.

As principais características que definem um microserviço são (FOWLER, 2014; NEWMAN, 2015):

- ✓ Cada microserviço implementa um único recurso de negócios ou funcionalidade;
- ✓ Um microserviço é o suficiente simples para ser desenvolvido e mantido por um único programador;
- ✓ Microserviços são executados em processos separados, comunicando-se por meio de padrões de mensagens ou APIs bem definidas;
- ✓ Microserviços não compartilham armazenamentos de dados. Cada microserviço é responsável por gerenciar seus próprios dados;
- ✓ Microserviços têm bases de código separadas e não compartilham código-fonte. No entanto, eles podem usar bibliotecas de utilitários comuns;
- ✓ Cada microserviço pode ser implantado e atualizado de forma independente de outros serviços.

Numa arquitetura MOA, a saída de um serviço é usada como uma entrada para outro em uma orquestração de serviços independentes. Ao ser agnóstico de dispositivos e plataformas, os microserviços fornecem flexibilidade para o sistema desenvolvido. Os microserviços têm diferentes formas de comunicação e processamento sendo mais um dos atributos que os distinguem do SOA tradicional.

4 Framework Moleculer

O framework *Moleculer* é uma estrutura de desenvolvimento com a linguagem JavaScript para aplicações com arquiteturas orientadas a microserviços (MOA) (MOLECULER, 2018). O framework é de código aberto (*open source*) e roda sobre a plataforma Node.js. A Figura 3 apresenta a arquitetura MOA do framework *Moleculer*, na qual os serviços são executados em nós individuais que se comunicam via protocolo de comunicação (*transporter*).

Na arquitetura MOA do *Moleculer*, todos os microserviços são iguais, não havendo nenhum tipo de hierarquia ou prioridade. Uma grande vantagem desse framework é que todos os microserviços desenvolvidos possuem disponível um recurso automático de registro e descoberta. Dessa forma, todos os serviços existentes são informados após a criação de um novo serviço ou a disponibilização de uma nova funcionalidade em um serviço. Outro recurso importante é o balanceamento automático de carga, o qual tem função de distribuir dinamicamente a carga da comunicação entre os microserviços uniformemente.

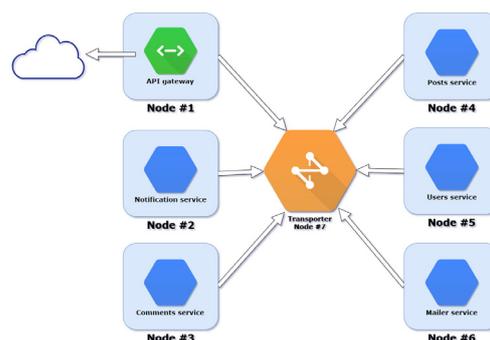


Figura 3: Arquitetura orientada a microserviços do *Moleculer*

Na arquitetura MOA da Figura 3, os quadrados de cor azul representam os nós de gerenciamento do *Moleculer* para suporte ao desenvolvimento, descoberta e gerenciamento dos (micro) serviços. Cada nó pode conter um ou mais serviços e é responsável pelo gerenciamento dos seus dados (possui seu próprio banco de dados). Cada serviço disponibilizado na arquitetura é representado com um hexágono azul. Os serviços podem oferecer e executar diferentes tarefas que são chamadas de ações (ex: um serviço de aquisição de dados pode oferecer uma ação de aquisição de dados de entrada e uma ação de atualização de dados de saída). O hexágono verde representa o microserviço de gateway, o qual tem a função de interface de conexão dos serviços internos (hexágonos azuis) com aplicações externas através de chamadas via nuvem por meio do protocolo REST. A comunicação entre os microserviços é realizada via um serviço de transporte (*transporter*), representado pelo hexágono laranja. A troca de informações entre os microserviços utilizando o *transporter* é realizado pelo módulo *Serializers*. Os principais recursos para microserviços que compõem o framework são (MOLECULER, 2018):

- ✓ Conceito de requisição-resposta;
- ✓ Arquitetura baseada em eventos;
- ✓ Suporte a middlewares para Node.js;
- ✓ Suporte ao armazenamento em cache de variável;
- ✓ Diversas opções de comunicação (*transporters*): TCP/IP, NATS, MQTT e etc;
- ✓ Diversas opções de serializers: JSON, MsgPack, Protocol Buffer e etc;
- ✓ Suporte ao desenvolvimento de múltiplos serviços em um mesmo nó;
- ✓ Suporte nativo para o registro de serviços;
- ✓ Descoberta automática de serviços;
- ✓ API para interface com aplicações externas.

5 Arquitetura baseada em Microserviços para aplicações de IIoT

Este artigo apresenta uma nova proposta de arquitetura orientada a microserviços (MOA) para aplicações de IIoT. A estrutura geral da arquitetura proposta pode ser vista na Figura 5. A arquitetura usa como base o framework *Moleculer* e discute o desenvolvimento de diferentes microserviços para realização de atividades relacionadas à IIoT. Esta proposta de arquitetura busca suportar as seguintes aplicações relacionadas à IIoT a partir da criação de serviços de: aquisição de dados (DAQ), monitoramento remoto (dashboard), otimização (KPIs), controle de processo, identificação de sistemas, detecção de falhas ou anomalias e mecanismos de redundância. No entanto, a arquitetura pode ser facilmente expandida, através da criação de novos serviços, para permitir novas aplicações como virtualização, simulação de sistemas, segurança da informação, realidade aumentada e análise de dados usando técnicas de Big Data.

O principal componente operacional da arquitetura MOA é o *Service Broker*, um contêiner de microserviços, que é o responsável pela configuração dos nós como: nome, opções de comunicação, registro de (micro)serviços, descoberta automática de serviços, cache de variáveis, entre outros recursos. No *Service Broker* é definido qual *transporter* será utilizado para comunicação entre os serviços e as configurações adicionais desse *transporter* como o endereço IP e porta de comunicação. Cada microserviço é composto por:

- ✓ *Name*: nome do microserviço;
- ✓ *Settings*: configurações do microserviço, como porta de comunicação. Essas configurações são enviadas durante o procedimento de descoberta do serviço;
- ✓ *Actions*: ações ou funções que compõem o funcionamento do microserviços, podendo ser chamadas internamente por outros microserviços ou externamente via API REST;

- ✓ *Methods*: métodos são funções privadas do microserviço que somente são executadas internamente;
- ✓ *Events*: eventos podem ser disparados conforme execução das ações do microserviço. Cada microserviço publica um evento sempre que necessário e outros microserviços podem se inscrever em eventos publicados.

Os serviços da arquitetura MOA oferecem uma funcionalidade principal, podendo executar diferentes ações relacionadas àquela funcionalidade. De forma resumida, a arquitetura da Figura 5 é composta dos seguintes microserviços e suas respectivas ações:

- ✓ **Microserviço de Aplicação**: são os serviços que oferecem as funcionalidades principais para o sistema como Monitoramento, Controle de Processos, Otimização e etc. Cada um desses microserviços pode oferecer diferentes ações relacionadas à uma funcionalidade. Por exemplo, o microserviço de Controle de Processos oferece diferentes algoritmos para cálculo das ações de controle como PID, Preditivo MPC e etc;
- ✓ **Microserviço Gateway (API)**: serviço de conexão padronizada de aplicações/requisições externas aos microserviços da arquitetura;
- ✓ **Microserviço M2M**: serviço de comunicação em rede com máquinas e sistemas. Semelhante a um middleware, este serviço é responsável pela comunicação usando diferentes redes e protocolos (Modbus TCP/IP, CoAP, MQTT, etc) e disponibilização desses dados aos microserviços da arquitetura. Esse microserviço disponibiliza duas ações: aquisição (amostragem) de dados de entrada e atualização de dados de saída;
- ✓ **Microserviço DAQ**: serviço de aquisição dados de módulos de hardware alocados no processo. Essa aquisição de dados é transparente aos microserviços. Também possui duas ações: aquisição de dados de entrada e atualização de dados de saída;

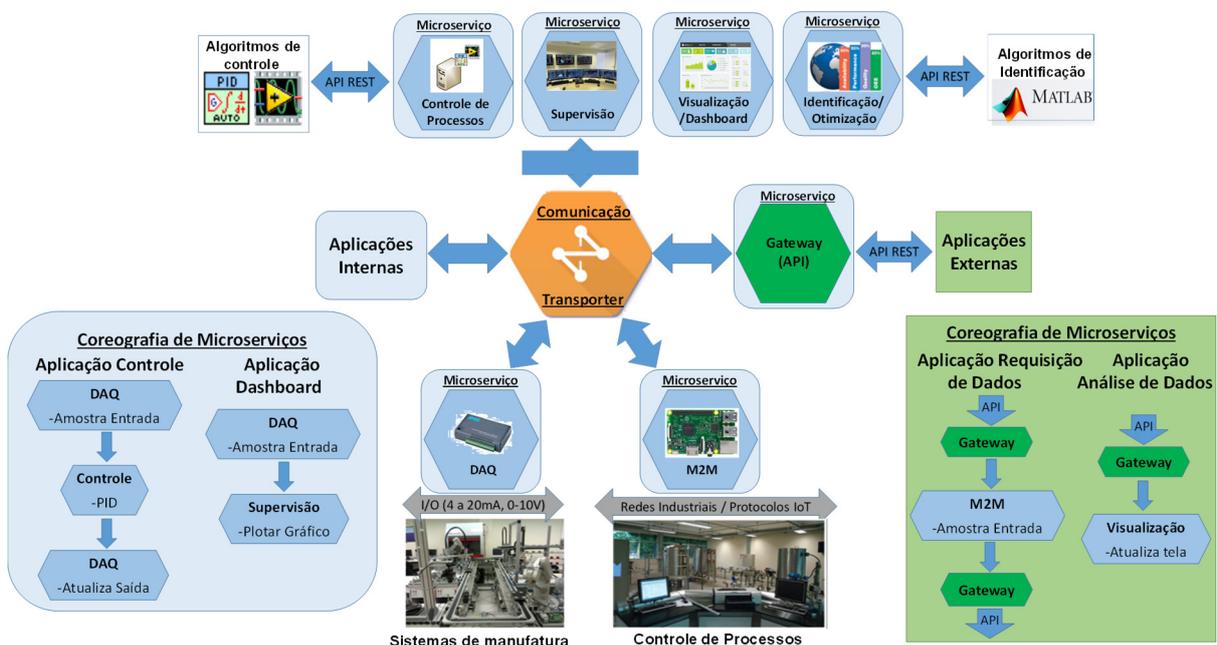


Figura 5- Arquitetura Orientada a Microserviços para aplicação IIoT

Na Figura 5, as aplicações internas são desenvolvidas em Node.js e coreografam os microserviços diretamente, conforme sequência definida, usando comunicação via *transporter*. Aplicações externas podem ser desenvolvidas em qualquer plataforma, mas sempre necessitam acessar e coreografar os microserviços via comunicação REST (API) através do microserviço Gateway.

Considerando a aplicação interna “Controle” da Figura 5, dois *Service Broker* devem ser definidos, sendo um para cada microserviço: DAQ e Controle. O microserviço DAQ é composto por duas ações: “Amostra Entrada” e “Atualiza Saída”. O microserviço Controle é composto por uma ação de controle da malha do processo. A coreografia ou sequência de execução dos serviços é predeterminada conforme a aplicação necessária. Neste caso da aplicação Controle, inicialmente o microserviço de aquisição de dados (DAQ) é executado num hardware embarcado na planta para realizar a aquisição de dados (ex: sinal 4 a 20 mA de um sensor) de um canal de entrada e transmitir esse dado digital via *transporter* para o microserviço de Controle. Perceba que neste momento, a ação de amostrar uma entrada está sendo executada no microserviço DAQ. Adicionalmente esta ação de amostrar uma entrada pode ser configurada para execução aperiódica (sob demanda) ou periódica (sequencialmente de acordo com um intervalo de tempo). Ao receber a comunicação do sinal de entrada com a variável de interesse, o microserviço Controle executa uma ação, no caso o algoritmo de controle “PID”, e calcula um novo valor do sinal de controle para ser aplicado ao atuador da malha. Perceba que no caso do microserviço Controle, cada algoritmo de controle diferente pode ser cadastrado como uma “Ação” do microserviço, fornecendo flexibilidade e modularidade para o desenvolvimento e implementação. O sinal de controle calculado é então transmitido via *transporter* ao microserviço DAQ. Este microserviço executa agora a ação de atualizar uma saída ou canal de saída (ex: sinal 4 a 20 mA de um atuador) do hardware para atuação sobre a planta.

Considerando a aplicação externa “Requisição de Dados” da Figura 5, dois *Service Broker* também seriam definidos para os microserviços Gateway e M2M. O microserviço Gateway realiza a interface entre aplicações externas e os microserviços da arquitetura via comunicação REST (API). A coreografia ou sequência de execução dos serviços é predeterminada, onde inicialmente uma requisição de dados (variáveis de interesse) feita por uma aplicação externa é recebida pelo microserviço Gateway via API REST. Após a requisição dos dados, o microserviço Gateway se comunica via *transporter* com microserviço M2M para a aquisição dos dados requeridos do processo/planta. O microserviço M2M é executado num hardware embarcado na planta, realizando a interface com um protocolo de rede para a aquisição dos dados (ex: dado de um sensor) e a transmissão (retorno) desse dado via *transporter* para o microserviço Gateway. Perceba que a ação de amostrar uma entrada está sendo executada no microserviço M2M. Esta mesma ação de

amostrar uma entrada (ou outra ação, por exemplo, de atualizar uma saída) pode ser configurada no microserviço M2M para interface e comunicação com diferentes protocolos de rede industriais e da IoT. Finalmente, o microserviço Gateway envia os dados solicitados via REST (API) para uso pela aplicação externa.

Buscando a validação operacional da arquitetura baseada em microserviços para aplicações de IIoT, testes foram realizados com a implementação de uma malha de controle de nível e supervisão de uma variável da planta de processos usando os microserviços de Controle, M2M, Gateway e Supervisão. Nesta implementação, uma aplicação externa utiliza os microserviços Controle e Supervisão para realizar o Controle (PID) e a Supervisão/monitoramento (plotagem de gráfico no tempo das variáveis) da variável do processo usando o software LabVIEW. O microserviço M2M foi usado para aquisição da variável do processo (PV) e atualização do sinal de controle na planta (MV) usando o protocolo de IoT CoAP. O *transporter* utilizado para comunicação entre os microserviços foi o NATS (padrão de alto desempenho de troca de mensagens distribuído baseado em fila). O microserviço Gateway foi usado para comunicação dos dados via REST com a aplicação externa criada no LabVIEW. A malha de controle regulatório foi projetada com um período de atualização/discretização de 1s (aquisição do sensor, cálculo do controle e atuação sobre a planta). Nestes testes o objetivo não foi avaliar o desempenho de controle, mas somente verificar a capacidade de controlar e manter uma malha de controle de nível da planta estável usando a arquitetura MOA.

A arquitetura MOA proposta oferece os seguintes benefícios de aplicação: 1. Aumento da resiliência: usando microserviços toda a aplicação é descentralizada e desacoplada em serviços que atuam como entidades separadas. 2. Escalabilidade: A escalabilidade é um aspecto chave dos microserviços. Como cada serviço é um componente separado, permite-se expandir uma única funcionalidade (ação) ou serviço sem ter que dimensionar toda uma aplicação. 3. Disponibilidade: os serviços críticos para a aplicação podem ser implantados em vários nós ou servidores para aumentar a disponibilidade e o desempenho de um serviço sem impactar o desempenho de outros serviços. 4. Flexibilidade de desenvolvimento: usando microserviços, o desenvolvimento da aplicação não fica limitado a um único software ou linguagem. Dessa forma, é possível escolher a ferramenta certa para cada tarefa. 5 – Facilidade de desenvolvimento e modularidade: a criação dos microserviços é mais simples e rápida por executarem funcionalidades específicas, fornecendo maior modularidade para a aplicação.

6 Conclusões e Trabalhos Futuros

Este trabalho apresentou uma revisão sobre arquiteturas orientadas a serviço (SOA) e sua aplicação industrial. Além disso, discutiu as características e vantagens de uma proposta de arquitetura orientada a microserviços em nuvem para aplicações de Internet das

Coisas Industrial (IIoT). Detalhes operacionais da proposta como o framework utilizado, a comunicação e composição de serviços por coreografia foram explicados. A integração das tecnologias de automação e TI usando uma arquitetura SOA permitirá o uso e compartilhamento de micros serviços para obtenção de uma arquitetura flexível, escalável, interoperável, distribuída e conectada em rede, de acordo com os requisitos das novas aplicações relacionadas à Indústria 4.0 e IIoT.

Testes operacionais iniciais da arquitetura realizados com a implementação de uma malha de controle e supervisão de uma variável de processo usando micros serviços validam a proposta e demonstram seu potencial para aplicações de IIoT. Trabalhos futuros focarão na avaliação do desempenho de aplicações da arquitetura MOA (quantificação da latência e análise temporal de requisições/respostas, gargalos com o uso de micros serviços e impacto da variação do número de requisições/resposta) e no desenvolvimento dos outros micros serviços propostos.

Agradecimentos

Os autores agradecem ao apoio para este trabalho: 2015/01303-2, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências Bibliográficas

- Chen Y., Du, Z. and Garcia-Acosta, M. (2010). Robot as a service in cloud computing. In: 5th IEEE International Symposium on Service Oriented System Engineering, 2010, pp. 151–158.
- Colombo, A. W., Karnouskos, S., Bangemann, T. (2014). Towards the Next Generation of Industrial Cyber-Physical Systems, In: Bangemann, T. et al. (eds), Ch 1. Springer International Publishing, Switzerland, pp. 01–22.
- Decotignie, J. D. (2005). Ethernet-based real-time and industrial communications. Proceedings of the IEEE, v. 93, n. 6, pp. 1102-1117.
- Dustdar, S.; Papazoglou, M. P. (2008). Services and Service Composition - An Introduction. Information Technology, v. 2, pp. 86-92.
- Fattori, C. C.; Junqueira, F.; Santos Filho, D. J. Dos; Miyagi, P. E. (2011). Service composition modeling using interpreted Petri net for system integration. In: 2011 IEEE International Conference on Mechatronics, pp. 696-701.
- Fowler, M. (2014). Microservices a definition of this new architectural term. March 2014. Disponível em: <<https://martinfowler.com/articles/microservices.html>>. Acesso em: fevereiro, 2018.
- Hegazy, T. and Hefeeda, M. (2015). Industrial Automation as a Cloud Service, IEEE Transactions on Parallel and Distributed Systems, v. 26, n. 10, pp. 2750-2763, Oct. 1.
- Jammes, F., Karnouskos, S., Bony, B., Nappey, P., Colombo, A. W., Delsing, J., Eliasson, J., Kyusakov, R., Stluka, P., Tilly, M., Bangemann, T. (2014), "Promising Technologies for SOA based Industrial Automation Systems", Industrial Cloud based Cyber Physical Systems: The IMC AESOP Approach. A. W. Colombo et al. (eds.), Springer, pp. 89-109.
- Kulkarni, A., IoT and SCADA: Complimentary technologies for Industry 4.0, (2016). Disponível: <<http://altizon.com/iotandscadacomplimentarytechnologieshatarestepingstonetoindustry40/>>. Acesso: 2016.
- Lee, J.; Bagheri, B. and Kao, H. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems”, Manufacturing Letters, v. 3, January 2015, pp. 18-23.
- Leitão, P.; Colombo, A.W.; Karnouskos, S. (2016). Industrial automation based on cyber-physical system technologies: Prototype implementations and challenges. Computers in Industry. v. 81, pp. 11-25.
- Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues, Journal of Industrial Information Integration, v. 6, pp. 1-10.
- Moleculer. (2018). Fast & powerful microservices framework for Node.js. Disponível em: <<https://moleculer.services/>>. Acesso em: Dezembro, 2017.
- Newman, S. (2015). Building Microservices. O’Reilly Media, Inc.. CA: Gravenstein, 2015.
- Nikolaidis, P.; Didic, A.; Mubeen, S.; Pei-Breivold, H.; Sandstrom, K. and Behnam, M. (2015). Applying Mitigation Mechanisms for Cloud-based Controllers in Industrial IoT Applications, Malardalen University, Sweden.
- Sheng, Q. Z.; Qiao, X.; Vasilakos, A. V.; Szabo, C.; Bourne, S.; Xu, X. (2014). Web services composition: A decade’s overview. Information Sciences, v. 280, pp. 218– 238.
- Souit, S. (2013). Orquestração de sistemas produtivos dispersos. Tese de Doutorado, Engenharia Mecânica, Universidade de São Paulo, São Paulo, 2013. Acesso em: 31 jan. 2018.
- Stankovic, J. A. (2014). Research Directions for the IEEE Internet of Things, Internet of Things Journal, v. 1, n. 1, pp. 3-9.
- Wu, D., Greer, M., Rosen, D., and Schaefer, D. (2013). Cloud manufacturing: Strategic vision and state-of-the-art, Journal of Manufacturing Systems, v. 32, n. 4, pp. 564–579.
- Xiao, Z.; Wijegunaratne, I.; Qiang, X.. (2016). Reflections on SOA and Microservices. In: 4th IEEE International Conference on Enterprise Systems, pp. 60-67.
- Zuehlke, D. (2010). Smart Factory—Towards a factory-of-things, Annual Reviews in Control, v. 34, n. 1, pp. 129-138.