

Uma Plataforma Escalável para Desenvolvimento de Aplicações de IoT

Jorge Pereira¹, Thais Batista¹, Flavia C. Delicato², Paulo F. Pires²

¹Departamento de Informática (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)
Natal - RN

²Departamento de Ciência da Computação (DCC)
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro - RJ

jorgepereirasb@gmail.com, thais@ufrnet.br, fdelicato@gmail.com,
paulo.f.pires@gmail.com

Abstract. *The Internet of Things (IoT) is an emergent paradigm that connects physical devices to the digital world, allowing the building of a myriad of applications. These applications need to be built and executed in a scalable way to support many connected devices as well as to store and process the huge amount of data they produce. Additionally, IoT applications also need to deal with several different protocols. In this context, this paper presents the EcoCIT platform, a scalable middleware platform that supports the integration of IoT devices to the Internet, as well as the development and execution of IoT applications with scalability requirements using on-demand computing services provided by cloud computing platforms.*

Resumo. *A Internet das coisas (IoT) é um paradigma emergente que conecta dispositivos físicos ao mundo digital, permitindo a construção de uma miríade de aplicações. Desenvolver aplicações para de IoT não é uma tarefa trivial. Tais aplicações precisam ser construídas atendendo requisitos de escalabilidade a fim de suportar um grande número de dispositivos conectados, além de armazenar e processar a enorme quantidade de dados produzida. Além disso, as aplicações precisam lidar com protocolos distintos. Nessa perspectiva, o presente artigo apresenta a EcoCIT, uma plataforma de middleware escalável que provê suporte para a integração de dispositivos de IoT à Internet, bem como ao desenvolvimento e execução de aplicações de IoT com requisitos de escalabilidade através do uso serviços computacionais providos sob demanda por plataformas de computação em nuvem.*

1. Introdução

A *Internet das Coisas* (do inglês, *Internet of Things –IoT*) é um paradigma emergente que abrange uma infraestrutura de hardware, software e serviços que conectam objetos físicos, denominados como *coisas*, à Internet (Commission of the European Communities 2008), permitindo o surgimento de uma miríade de aplicações que poderão se beneficiar dos novos tipos de dados, serviços e operações disponíveis. Segundo (Walport 2014), em 2014 existiam cerca de 14 bilhões de dispositivos

conectados à Internet e estima-se que, em 2020, esse número chegará a 100 bilhões. Devido ao grande número de dispositivos, a IoT caracteriza-se por uma enorme *heterogeneidade*, uma vez que os dispositivos e os componentes de software que os gerenciam são desenvolvidos por diversos fabricantes e sem padronização. Outra característica marcante desse paradigma é a *dinamicidade*. Em IoT, os objetos podem tornar-se indisponíveis devido a vários fatores, tais como: mobilidade dos dispositivos, ausência de rede, disponibilidade energética, etc.

Devido a essas características, desenvolver aplicações para IoT envolve, em muitos casos, lidar com diversos dispositivos, protocolos e formatos de dados que não seguem uma padronização comum, bem como usar um número desconhecido de dispositivos que podem ou não estar disponíveis. Nesse contexto, plataformas de *middleware* surgem como soluções promissoras para prover interoperabilidade e gerenciar a crescente variedade de dispositivos associados às aplicações, bem como o consumo de dados por parte dos usuários finais (Teixeira, et al. 2011). Dada a *dinamicidade* desse ambiente, tais plataformas precisam ter a capacidade de assimilar um número crescente de dispositivos e requisições e funcionar corretamente, mesmo em situações de uso intenso (Pires, Delicato, et al. 2015). Para satisfazer esse requisito, as plataformas devem ajustar a quantidade de recursos computacionais consumidos de acordo com a necessidade. Nesse contexto, autores como (Gubbi, et al. 2013) (Pires, Delicato, et al. 2015) (Cavalcante, et al. 2016) afirmam que o paradigma de *computação em nuvem* é um candidato natural para complementar a IoT, no sentido de prover escalabilidade e serviços sob demanda para este ambiente altamente dinâmico. A integração entre IoT e computação em nuvem dá origem ao paradigma conhecido como *Nuvem das Coisas* (do inglês, *Cloud of Things – CoT*) (Aazam, et al. 2014).

Embora a integração entre os dois ambientes (IoT e cloud) seja o ponto chave na concepção do paradigma emergente de CoT, atualmente a maioria das iniciativas existentes nessa direção ainda estão em estado inicial de desenvolvimento e/ou não foram completamente validadas (Cavalcante, et al. 2016) (Nastic, et al. 2014). Aplicações nesse contexto devem refletir a quantidade de recursos computacionais demandada pelos usuários e dispositivos de IoT e, ativamente, ajustar essa demanda à quantidade de recursos alocada na infraestrutura de nuvem subjacente (Serrano, et al. 2013). No entanto, gerenciar tal elasticidade no contexto de CoT ainda é um desafio (Nastic, et al. 2014) (Serrano, et al. 2013).

Nesse contexto, apresentamos a *EcoCIT* (*Ecosystem of Cloud and Internet of Things*), uma plataforma de middleware para CoT que oferece suporte à conexão de dispositivos de IoT à Internet, bem como ao desenvolvimento e execução de aplicações de IoT com requisitos de escalabilidade através do uso de serviços computacionais providos sob demanda por plataformas de computação em nuvem. A *EcoCIT* surgiu como uma evolução da plataforma *EcoDiF* (Ecosistema Web de Dispositivos Físicos) (Pires, Cavalcante, et al. 2014). Apesar de oferecer suporte a conexão de dispositivos físicos à Internet e ao desenvolvimento aplicações, em seu projeto, a *EcoDiF* não levou em conta, como requisitos fundamentais, questões relacionadas à escalabilidade. Como consequência, a plataforma é capaz de suportar apenas um número limitado de dispositivos, usuários e aplicações conectadas a mesma. Além disso, a plataforma não é capaz de suportar a enorme quantidade de dados que os dispositivos de IoT podem

gerar. Na *EcoCIT* a quantidade de recursos computacionais alocada na infraestrutura de nuvem subjacente é ajustada dinamicamente em resposta ao aumento ou diminuição da carga de trabalho demandada pelos dispositivos, aplicações e usuários conectados a mesma. Além disso a plataforma utiliza SGDBs escaláveis e distribuídos para gerenciar os dados produzidos pelos dispositivos e aplicações conectadas.

O resto desse artigo está estruturado como segue: A seção 2 descreve a *EcoDiF*, plataforma que serviu como base para construir a *EcoCIT*; a seção 3 apresenta a arquitetura e o funcionamento da *EcoCIT*; a seção 4 apresenta um experimento computacional que compara a escalabilidade da *EcoCIT* em relação à *EcoDiF*, mostrando que, de fato, a nova plataforma tem escalabilidade superior à sua predecessora; a seção 5 apresenta trabalhos relacionados; e a seção 6 conclui o trabalho.

2. EcoDiF

A *EcoDiF* é uma plataforma de middleware que integra dispositivos de IoT e os conecta à Internet, fornecendo funcionalidades de controle, visualização e armazenamento de dados em tempo real. A plataforma provê uma interface bem definida, baseada em REST (Fielding e Taylor 2002), para expor as funcionalidades dos dispositivos na Web, provendo padronização e simplificação do processo de desenvolvimento de aplicações, além de minimizar barreiras no tocante à incompatibilidade entre diferentes fabricantes, protocolos proprietários e formatos de dados. A *EcoDiF* provê ainda suporte à execução de aplicações que possam fazer uso das funcionalidades e dados providos pelos dispositivos a ela conectados.

Em sua arquitetura a *EcoDiF* conta com os seguintes módulos: i) *Módulo de Conexão de Dispositivos*, que visa facilitar a conexão de dispositivos físicos à plataforma. Esse módulo oferece um API, baseada em REST, por meio da qual os dispositivos podem compartilhar seus dados através da plataforma; ii) *Módulo de Manipulação de Dados*, que gerencia o acesso aos dados produzidos pelos dispositivos conectados; iii) *Módulo de Visualização e Gerenciamento*, o qual provê uma interface Web para permitir aos usuários monitorar o estado e localização de seus dispositivos, bem como visualizar dados históricos armazenados; iii) *Módulo de Colaboração*, que visa facilitar a colaboração entre usuários da *EcoDiF*, permitindo realizar buscas por dispositivos e aplicações a partir de seus respectivos *metadados* (tipo, usuário, localização, etc); iv) *Módulo de Armazenamento*, que consiste de dois repositórios básicos, um para armazenamento de dados utilizando uma base de dados relacional, e outro para armazenamento de scripts de aplicações em um sistema de arquivos; v) *Módulo de Segurança*, gerencia a autenticação e autorização dos usuários conectados à plataforma; vi) *Módulo de Serviços Comuns*, envolve serviços de infraestrutura providos pela plataforma, tais como, gerenciamento de ciclo de vida de aplicações, transações, etc; e por fim, vii) *Módulo de Aplicações*, que provê um modelo e um ambiente para programação e execução de aplicações que fazem uso dos dados disponíveis na *EcoDiF*. Essas aplicações são construídas como *mashups* (Guinard et al., 2009) e utilizando a linguagem EMMML (*Enterprise Mashup Markup Language*). Os *mashups* são implementadas como scripts escritos em EMMML e executadas em um motor (*engine*) de execução que processa tais scripts.

Dadas as funcionalidades providas pela plataforma, a *EcoDiF* possui quatro perfis de *stakeholders*: i) *fabricantes de dispositivos*, responsáveis por desenvolver drivers que irão realizar a comunicação entre os dispositivos e a *EcoDiF* através da API REST provida pela plataforma; ii) *provedor de dados*, que são os proprietários dos dispositivos, que disponibilizam os dados por eles aferidos; iii) *desenvolvedores de aplicações*, que constroem as aplicações que fazem uso dos dados disponíveis na plataforma; e, por fim, iv) *consumidores de dados*, os usuários que interagem com a plataforma para buscar e utilizar informações disponíveis.

Apesar de oferecer diversas facilidades para integração de dispositivos de IoT e desenvolvimento de aplicações, a *EcoDiF* foi projetada para executar sobre um único servidor. Além disso, o sistema de armazenamento por ela utilizado é centralizado, ou seja, todos os dados residem em uma única máquina. Esses fatores limitam sua capacidade de gerenciar grandes volumes de dados e atender a um grande número de dispositivos, usuários e aplicações. Como consequência de executar sobre um único servidor e de usar tecnologias de armazenamento centralizadas, incrementar a capacidade de processamento, memória, rede e armazenamento da plataforma implica transferi-la para uma máquina mais robusta ou adicionar manualmente novos recursos (processadores, pentes de memória, discos de armazenamento, placas de rede, etc) ao servidor existente. A fim de superar essas limitações esse trabalho apresenta a plataforma *EcoCIT*.

3. EcoCIT

Como explicado, a *EcoCIT* surgiu como uma evolução da *EcoDiF*. Em sua arquitetura, a *EcoCIT* herdou diversas funcionalidades e módulos da *EcoDiF*, alguns dos quais foram alterados a fim superar suas limitações relacionadas à escalabilidade. O componente de software resultante das alterações realizadas sobre a *EcoDiF* foi denominado *EcoCIT Middleware*. O *EcoCIT Middleware* é, no entanto, apenas um dos componentes da arquitetura da *EcoCIT*. Ele provê todas as funcionalidades originalmente providas pela *EcoDiF*. Adicionalmente, a arquitetura da *EcoCIT* conta com cinco outros componentes que contribuem para lhe conferir escalabilidade e elasticidade. A visão geral da arquitetura da *EcoCIT* e o detalhamento de cada um dos componentes é descrita na seção 3.1. As alterações realizadas sobre os módulos da *EcoDiF* para a concepção do *EcoCIT Middleware* são descritas na seção 3.2.

3.1 Arquitetura da EcoCIT

A arquitetura da *EcoCIT* é composta por diversos componentes organizados de maneira distribuída. A arquitetura faz uso de recursos computacionais providos por uma plataforma de nuvem que oferece serviços a nível de *IaaS* (Infraestrutura como Serviço, do inglês, *Infrastructure as a Service*). Para permitir que a *EcoCIT* possa, de fato, fazer uso da elasticidade que o ambiente de computação em nuvem pode prover e, com isso, atender a um número grande de usuários e dispositivos a ela conectados, sua arquitetura permite que diversas máquinas possam trabalhar de maneira agrupada sob a forma de um *cluster*. A carga de trabalho (requisições realizadas à plataforma) é distribuída entre as máquinas que fazem parte do *cluster*. A capacidade do *cluster* pode ser incrementada e reduzida em tempo de execução através da adição e remoção de servidores.

A arquitetura da *EcoCIT* está estruturada conforme ilustra a Figura 1. Um *cluster* composto por diversas máquinas virtuais (VM1, VM2, ... , VMn) cada uma das quais contém uma instancia do *VM Monitor*, *EcoCIT Middleware* e *Oauth Server*. Tais máquinas realizam o processamento das requisições dos usuários e dispositivos conectados à plataforma, enquanto uma outra máquina hospeda o componente *VM Manager*, que realiza o balanceamento da carga de trabalho entre as diversas instâncias do *EcoCIT Middleware* e gerencia a adição e remoção de máquinas virtuais ao *cluster* conforme a demanda, de modo que a quantidade de recursos alocada na infraestrutura de nuvem subjacente seja proporcional à necessidade demandada pelos usuários. Os dados estruturados gerenciados pela plataforma são armazenados no *NewSQL DB*, enquanto que os dados menos estruturados produzidos pelos dispositivos são armazenados no *NoSQL DB*. Ambos são SGBDS escaláveis e distribuídos.

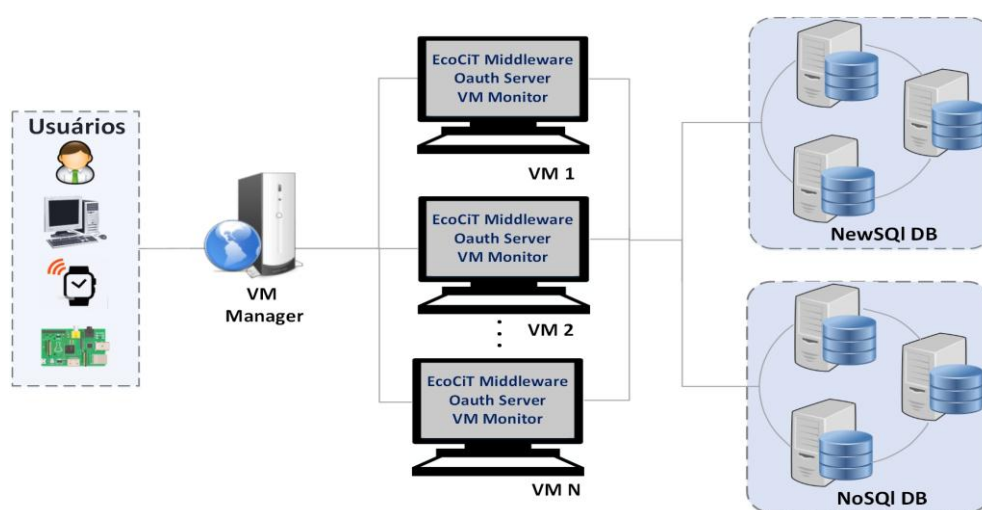


Figura 1. Visão geral da arquitetura da EcoCIT

Nessa arquitetura, o componente *EcoCIT Middleware* provê funcionalidades de controle e gerenciamento de *feeds*, dispositivos, usuários e aplicações. Essas funcionalidades podem ser acessadas via interface Web ou via API REST. A autenticação dos usuários que fazem uso da API REST provida pela plataforma é gerenciada pelo componente *Oauth Server*, um servidor de autorização que gerencia o acesso à API da *EcoCIT* com base no protocolo *Oauth* (Jones e Hardt 2012). As *Virtual Machines (VMs)*, por sua vez, são responsáveis por alojar uma instância do *EcoCIT Middleware*, *Oauth Server* e do *VM Monitor*, permitindo a execução dos mesmos.

O *VM Monitor* é responsável por monitorar o consumo de recursos computacionais (processamento, memória e rede) em cada uma das *VMs* instanciadas e por emitir alertas ao *VM Manager* quando for verificado que alguma das máquinas está sobrecarregada ou ociosa. O *NewSQL DB* é um SGBD escalável e distribuído do tipo *NewSQL* que permite armazenar as informações estruturadas gerenciadas pela *EcoCIT*, tais como os dados dos usuários, aplicações, dispositivos conectados, etc. Já o *NoSQL DB* é um SGBD do tipo *NoSQL* utilizado para armazenar os dados gerenciados pelo *Oauth Server* e os dados produzidos pelos dispositivos conectados à plataforma. Por ser um SGBD altamente escalável, robusto e capaz de armazenar e recuperar os dados produzidos pelos sensores de forma rápida e eficiente, o SGBD *Apache Cassandra* foi escolhido para fazer o papel do *NoSQL DB*. Para fazer o papel do *NewSQL DB* foi

utilizado o SGBD *NuoDB*, além de ser um SGBD escalável e distribuído o *NuoDB* permite armazenar dados estruturados mantendo as propriedades de atomicidade, consistência, isolamento e durabilidade em todas as suas transações.

Por fim, o *VM Manager*, tem o papel de gerenciar o *cluster* de *VMs*, além de distribuir a carga de trabalho (requisições) entre as diversas *VMs* pertencentes ao *cluster*. O *VM Manager* é composto internamente por dois componentes: *i)* uma aplicação JAVA que consome as métricas de uso de recursos computacionais monitoradas pelo *VM Monitor*, e com base nelas, decide quando é necessário instanciar uma nova *VM* para dividir com as demais a carga de trabalho, ou quando, por motivos de ociosidade dos recursos, é necessário desativar ou deletar parte das *VMs* pertencentes ao *cluster*; e *ii)* pelo balanceador de carga, que gerencia a distribuição das requisições HTTP dos usuários entre as diversas instâncias do *EcoCIT Middleware*. Para permitir que o estado das conexões (*sessões*) dos usuários que estão conectados à plataforma via interface Web não seja perdido quando uma *VM* é deletada, as sessões dos usuários são replicadas entre todas as máquinas pertencentes ao *cluster*.

A interação entre o *VM Manager* e a infraestrutura de nuvem subjacente a fim de instanciar e deletar VMs ocorre por meio de drivers desenvolvidos especificamente para a plataforma de nuvem usada para implantar a *EcoCIT*. Os drivers são componentes plugáveis que são anexados à plataforma. Atualmente existem drivers para as plataformas *Amazon EC2* e *OpenStack*. Informações referentes ao driver a ser utilizado, quantidade máxima de *VMs* que podem ser instanciadas e configuração de hardware das mesmas são especificadas em um arquivo de configuração utilizado pelo *VM Manager*.

3.2 EcoCIT Middleware

Como mencionado, o *EcoCIT Middleware* é um componente resultante das modificações realizadas sobre a *EcoDiF*. No total, três módulos da *EcoDiF* foram alterados para a concepção do *EcoCIT Middleware*: *Módulo de Armazenamento*, *Módulo de Aplicações* e *Módulo de Segurança*. Além dessas alterações, o antigo sistema de armazenamento usado pela *EcoDiF* foi removido para dar espaço aos SGBDs do tipo *NewSQL* e *NoSQL*.

Na *EcoDiF*, o *Módulo de Armazenamento* utiliza sistema de arquivos e um SGBD centralizado como suporte ao armazenamento de informações. Como consequência, a quantidade de inserções e consultas simultâneas realizadas sobre o banco de dados, bem como a quantidade de informações que podem ser armazenadas é bastante limitada. Na concepção do *EcoCIT Middleware* esse módulo foi alterado a fim de fazer uso dos sistemas de armazenamento providos pelo *NewSQL DB* e *NoSQL DB*, que são capazes de operar de maneira distribuída e escalar horizontalmente (com a adição de novos nós). Da mesma forma, na *EcoDiF* o *Módulo de Segurança* faz uso do antigo SGBD centralizado utilizado pela plataforma para consultar informações referentes aos usuários e gerenciar a autenticação e autorização dos usuários que acessam a plataforma via interface WEB. Com a remoção desse SGBD no processo de concepção da *EcoCIT*, esse módulo foi alterado para passar a fazer uso do *NewSQL DB*.

Por fim, na *EcoDiF* o *Módulo de Aplicações* utilizava sistemas de arquivos para armazenar os *mashups* criados pelos desenvolvedores de aplicações. Na *EcoCIT* as máquinas virtuais que dão suporte à execução da plataforma são criadas e deletadas

dinamicamente, dessa forma, não seria viável armazenar essas aplicações no sistema de arquivos das *VMs*. Portanto, o *Módulo de Aplicações* foi alterado para que passe a armazenar esses arquivos no *NewSQL DB*.

4 Avaliação

A fim de verificar se as modificações realizadas sobre a arquitetura da *EcoDiF* para a concepção da *EcoCIT* de fato tornam a nova plataforma mais escalável em comparação com sua predecessora, um experimento computacional foi conduzido. O experimento consistiu em estressar ambas as plataformas simulando diversos dispositivos e usuários acessando sua API REST simultaneamente. Utilizando a ferramenta para testes de sistemas Web *JMeter*, foram realizados sucessivos testes de carga em ambas as plataformas. Os testes de carga simularam um grupo de usuários que realizam um número fixo de requisições à plataforma a fim de atualizar e consumir dados providos por dispositivos de IoT. O número inicial de requisições simultâneas realizadas a cada plataforma foi fixado em 100. Ao longo dos testes esse número foi incrementado gradualmente em 100 unidades até chegar ao limite suportado por ambas as plataformas. Esse limite é alcançado quando parte das requisições falham. Requisições cujo tempo de resposta foram superiores a 12 segundos ou que retornaram um *status code* diferente de 200 (*status code OK*) foram consideradas como falhas.

Ambas as plataformas foram instaladas em *VMs* instanciadas sobre a plataforma de computação em nuvem *Amazon EC2*. Como explicado, a arquitetura da *EcoDiF* não foi projetada para executar sobre múltiplos servidores, no máximo a plataforma é capaz de fazer uso de duas máquinas, uma para suportar a plataforma propriamente dita e outra para suportar o SGBD centralizado utilizado pela mesma. Portanto, para a instalação da *EcoDiF* duas *VMs* foram utilizadas. Para a instalação da *EcoCIT* foram usadas 2 *VMs* para a instalação do SGBD *Apache Cassandra*, 1 *VM* para instalação da versão community do SGBD *NuoDB*, 1 *VM* para a instalação do *VM Manager* e, por fim, inicialmente foi instanciada pelo *VM Manager* 1 *VM* contendo uma instancia do *EcoCIT Middleware*, *VM Monitor* e *Oauth Server*. O limite máximo de máquinas a serem instanciadas pelo *VM Manager* foi definido em 3. Assim, até duas outras *VMs* poderiam ser instanciadas pelo *VM Manager* e adicionadas ao *cluster*, conforme a necessidade.

Todas as *VMs* usadas para a instalação de ambas as plataformas possuíam mesmas configurações de hardware e sistema operacional. Cada *VM* possuía 2 CPUS virtuais, 4 GB de memória RAM, 1 placa de rede virtual e utilizavam o S.O. *Ubuntu Server* 16.4. A *VM* utilizada para realizar os testes de carga possuía as seguintes configurações: 4 CPUS virtuais, 8 GB de memória RAM, 1 placa de rede virtual e fazia uso do sistema operacional *Windows Server* 2016. Os resultados obtidos com a realização do experimento são sumarizados no gráfico da Figura 2.

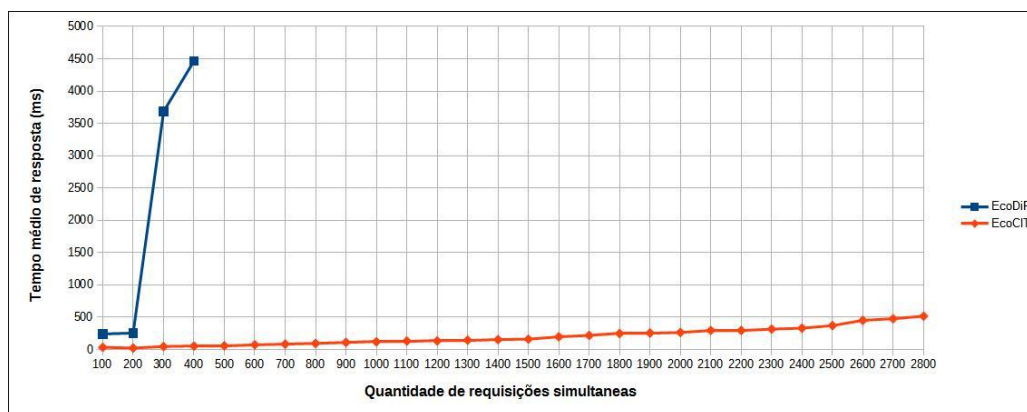


Figura 2. Resultado dos testes realizados sobre as plataformas EcoCIT e EcoDiF

Como se pode observar, a *EcoDiF* foi capaz de suportar até 400 requisições simultâneas. Acima desse número parte das requisições falharam devido ao fato de as *VMs* utilizadas para suportar a plataforma e o SGBD relacional utilizado pela mesma terem exaurido toda sua capacidade de processamento e memória. Dada a sua capacidade de executar de maneira distribuída e de dividir a carga de trabalho entre diversas *VMs*, a *EcoCIT* foi capaz de suportar um número bem maior de requisições simultâneas, chegando a 2800. Além disso, o tempo de resposta às requisições realizadas a *EcoCIT* foi bem menor que o tempo de resposta das requisições à *EcoDiF*. Esses resultados mostram que as alterações realizadas sobre a arquitetura da *EcoDiF* para a concepção da *EcoCIT* tiveram impacto positivo na escalabilidade da plataforma. Também é importante ressaltar que a arquitetura da *EcoCIT* permite escalar, a fim de que sua capacidade vá além do que foi explorado nesse experimento. Diferente da *EcoDiF*, sua arquitetura permite, em tempo de execução, adicionar novos nós aos SGBDs utilizados, tornando as consultas mais rápidas. Além disso, também é possível definir, no arquivo de configuração do *VM Manager*, um número maior de *VMs* que podem ser instanciadas e adicionadas ao *cluster* para atender requisições dos usuários.

5 Trabalhos relacionados

Como discutido anteriormente, desenvolver aplicações de IoT não é uma tarefa trivial, pois para isso frequentemente os desenvolvedores precisam lidar com questões relacionadas a escalabilidade e com um grande número de dispositivos heterogêneos. Esses desafios têm motivado o desenvolvimento de plataformas de middleware que oferecem facilidades para o desenvolvimento de aplicações de IoT. A título de exemplo, o trabalho proposto por (Lopes, et al. 2013) apresenta o *EXEHDA-UC*, uma arquitetura distribuída para suporte ao gerenciamento da aquisição, armazenamento e processamento dos dados produzidos por dispositivos de IoT. A arquitetura conta com componentes para aquisição de informações provenientes dos dispositivos, consulta a dados históricos sensorizados e consulta em tempo real a informações provenientes de sensores. A arquitetura também conta com um módulo que gerencia regras para manipulação e dedução de informações com base nos dados armazenados.

A plataforma *ManIoT* (Antunes, et al. 2016) permite integrar e gerenciar dispositivos de IoT à internet, oferecendo seus dados e funcionalidades de maneira padronizada, via REST, para facilitar o desenvolvimento de aplicações. Além de facilitar a integração e gerenciamento dos dispositivos, a plataforma também provê funcionalidades de: armazenamento de dados históricos aferidos pelos dispositivos; agendamento de ações a serem realizadas sobre os dispositivos (por exemplo, ligar ou desligar um aparelho); comunicação assíncrona, a fim de notificar à aplicações a ocorrência de ventos detectados pelos dispositivos, e etc.

O *Kaa*¹ é uma plataforma de middleware *open-source*, flexível e capaz de escalar de maneira horizontal. Ele oferece suporte para a gerenciamento e conexão de dispositivos de IoT à Internet, armazenamento e processamento de dados, além de oferecer APIs para desenvolvimento de aplicações que façam uso dos dados providos pelos dispositivos. O *Kaa* tem como foco acelerar o desenvolvimento de aplicações no domínio de IoT, oferecendo diversas abstrações que podem ser utilizadas pelos desenvolvedores para tal.

A *EcoCIT* tem como diferencial em relação às duas primeiras propostas supracitadas a capacidade de escalar de maneira horizontal a fim de suportar um grande número de dispositivos e usuários conectados. Como diferencial em relação a todas as propostas supracitadas, a *EcoCIT* tem a capacidade de fazer uso de serviços providos sob demanda por plataformas de computação em nuvem.

6 Considerações Finais

Este trabalho apresentou a *EcoCIT*, uma plataforma de middleware escalável para suporte ao desenvolvimento de aplicações de IoT, que é uma evolução da *EcoDiF*. Foram apresentadas as modificações realizadas sobre a arquitetura da *EcoDiF* para a concepção de *EcoCIT* e sua integração com plataformas de computação em nuvem. Essas alterações permitem à *EcoCIT* gerenciar um maior volume de dados proveniente de dispositivos de IoT, também permitem que a plataforma possa escalar de maneira horizontal, ou seja, através da adição de novas máquinas à sua arquitetura e a consumir recursos computacionais providos sob demanda por plataformas de computação em nuvem. O experimento computacional apresentado na seção 5 mostra que, de fato, essas alterações tiveram impacto positivo na escalabilidade da plataforma.

Apesar dos benefícios alcançados com a concepção da *EcoCIT*, a plataforma possui ainda algumas limitações. Algumas dessas dizem respeito ao componente *VM Manager* que, diferentemente dos outros componentes da arquitetura (*NewSQL DB*, *NoSQL DB* e do *cluster* de *VMs* utilizado para processar as requisições dos usuários), que podem ter sua capacidade incrementada em tempo de execução através da adição de novos nós, não existe na *EcoCIT* uma estratégia que permita incrementar em tempo de execução a capacidade de processamento, memória e rede do *VM Manager* em casos de necessidade. Portanto, um dos trabalhos futuros consiste em propor uma estratégia para superar essa limitação. Outro trabalho futuro consiste em propor mecanismos de tolerância a falhas para a *EcoCIT* a fim de tornar a plataforma mais resiliente.

¹ **Kaa**: <https://www.kaaproject.org/>

Agradecimentos

Esse trabalho foi parcialmente financiado pelo CNPq e pelo Instituto Nacional de Ciência e Tecnologia para Engenharia de Software (INES)². Thais Batista, Flavia C. Delicato e Paulo F. Pires são bolsistas de produtividade do CNPq.

Referências

- Antunes, Josué B., e outros. “ManIoT: Uma Plataforma para Gerenciamento de Dispositivos da Internet das Coisas.” Workshop de Gerência e Operação de Redes e Serviços (WGRS). Salvador, Bahia, 2016. 3-16.
- Cavalcante, Everton, et al. “On the interplay of Internet of Things and Cloud Computing: A systematic mapping study.” *Computer Communications* 89-90 (Março 2016): 17–33.
- Commission of the European Communities. “Early Challenges regarding the “Internet of Things”.” Bruxelas, 2008.
- Fielding, Roy T., e Richard N. Taylor. “Principled design of the modern Web architecture.” *ACM Transactions on Internet Technology (TOIT)*, 2002: 115-150 .
- Gubbi, Jayavardhana et al. “Internet of Things (IoT): A vision, architectural elements, and future directions.” *Future Generation Computer Systems* 29, n. 7 (Setembro 2013): 1645–1660. Kitchenham, B.A., e S. Charters. “Guidelines for Performing Systematic Literature Reviews in Software Engineering.” University/University of Durham, Keele, 2007.
- Lopes, Joao L. B., et al. “Uma arquitetura distribuída direcionada à consciência de contexto na Computação Ubíqua.” *Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP)*. Porto Alegre, 2013. 2022-2031.
- Nastic, Stefan et al. “Provisioning Software-defined IoT Cloud Systems.” *Int. Conf. on Future Internet of Things and Cloud*. Washington, USA: IEEE Computer Society, 2014. 288-295.
- Pires, Paulo F., et al. “A platform for integrating physical devices in the internet of things.” *Embedded and Ubiquitous Computing (EUC)*. Milano:, 2014. 234-241.
- Pires, Paulo F., e outros. “Plataformas para a Internet das Coisas.” *Minicurso do Simpósio Brasileiro de Rede de Computadores (SBRC)*, Vitória - ES, 2015.
- Serrano, Martin, et al. “Resource Optimisation in IoT Cloud Systems by Using Matchmaking and Self-management Principles.” *Cáp. 11 em THE FUTURE INTERNET*, edição: Alex Galis e Anastasius Gavras. Berlin: Springer, 2013.
- Teixeira, T., S. Hachem, V. Issarny, e N Georgantas. “Service oriented middleware for the Internet of Things: A perspective.” *4th European Conference on Towards a Service-Based Internet*. Poznan, Poland: LNCS, 2011. 220-229.
- Walport, Mark. “The Internet of Things: making the most of the Second Digital Revolution.” *UK Government Chief Scientific Adviser*, 2014.
- Guinard, D., Trifa, V. “Towards the Web of Things: Web mashups for embedded devices”, *Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web*, 2009.

² <http://www.ines.org.br/>