IoTMan Agent: Avaliação Qualitativa e Comparação de Ferramentas de Monitoramento para IoT

Dener Ottolini, Alexandre Heideker, Carlos Kamienski

[1]Universidade Federal do ABC (UFABC)

Abstract. *IoT environments are complex, with many moving parts and significant amounts of data. Traditional or homegrown monitoring approaches cannot scale or provide insights in these environments, lacking system observability. In this article we analyze qualitatively well know and established monitoring tools in terms of IoT support. This article also proposes a new IoT monitoring tool named IoTMan Agent, explaining how the tool can meet IoT specific demands and provide the understanding needed to manage the complexity of an IoT solution considering the heterogeneity and distributed nature of such systems.*

Resumo. *Ambientes de IoT são complexos, com muitas partes móveis e quantidades significativas de dados. Abordagens de monitoramento tradicionais ou domésticas podem não ser escaláveis, ou fornecer entendimento destes ambientes, não provendo observabilidade do sistema. Neste artigo analisamos qualitativamente ferramentas de monitoramento bem conhecidas e estabelecidas em termos de suporte à IoT. Este artigo também propõe uma nova ferramenta de monitoramento de IoT denominada IoTMan Agent, explicando como a ferramenta pode atender demandas específicas de IoT e fornecer o entendimento necessário para gerenciar a complexidade de uma solução de IoT considerando a heterogeneidade e natureza distribuída de tais sistemas.*

1. INTRODUÇÃO

Com o crescimento de aplicações inteligentes que adotam os conceitos de Internet das Coisas (*Internet of Things* - IoT), surge também a necessidade de garantir que os serviços, infraestruturas e aplicações consigam acompanhar este crescimento de forma escalável, confiável e com uma qualidade de serviço aceitável. Para garantir que estas necessidades sejam atendidas, é essencial que técnicas apropriadas para monitoramento e gerenciamento de recursos sejam implementadas.

Ferramentas de monitoramento permitem a coleta de dados através de diferentes abordagens. Um grande número de ferramentas com este propósito pode ser encontrado, tornando difícil a identificação de uma solução robusta, escalável, que traga as necessidades de monitoramento de sistemas tradicionais em combinação com as especificidades de ambientes IoT. Dentro do próprio contexto de IoT, uma ferramenta que atende uma solução inteligente pode não ser aplicável a outras, uma vez que aplicações inteligentes costumam apresentar grande heterogeneidade [Zyrianoff et al. 2020].

Diferentes provedores de serviços de tecnologia oferecem ferramentas proprietárias para monitoramento e gerenciamento de seus componentes, seja de *hardware* ou *software*, tornando-as geralmente dependentes de suas respetivas plataformas, arquiteturas e tecnologias. Em muitos cenários não é possível a customização ou exportação

dos dados de monitoramento que permitiriam a comparação entre diferentes ambientes e análise de estados, consumo e disponibilidade.

Este artigo analisa qualitativamente as características de diferentes ferramentas de monitoramento, sejam proprietárias ou de código livre, de se adaptarem a ambientes de IoT. Exploramos as características, necessidades e restrições de ambientes IoT e suas especificidades quando comparados a sistemas tradicionais, descrevendo como cada uma das plataformas opera para atender estas demandas.

Após a análise qualitativa e levantamento de requisitos, apresentamos uma ferramenta de monitoramento desenvolvida especificamente para operação em ambientes de IoT, o IoTMan Agent. A ferramenta foi desenvolvida com o propósito de oferecer a capacidade de monitoramento para diferentes ambientes e dispositivos, independentemente da plataforma, arquitetura ou infraestrutura. Além das métricas de sistemas comumente monitoradas, como uso de CPU e memória RAM, o IoTMan Agent oferece ainda a capacidade de analisar e monitorar a disponibilidade de serviços, conectividade entre componentes, fluxo fim-a-fim e o estado de processos e contêineres Docker [1].

O IoTMan Agent é uma extensão de trabalhos realizados em [Heideker et al. 2019], que apresenta um modelo inicial de ferramenta de monitoramento, e [Silva et al. 2022] onde uma proposta de arquitetura de gerenciamento desenvolvida especificamente para sistemas IoT é apresentada e denominada IoTManA (IoT Management Architecture). A IoTManA propõe um modelo multicamadas que pode ser aplicada em diferentes ambientes de IoT, independente da infraestrutura ou plataforma adotada, devido a sua característica genérica. Nesta abordagem, cada elemento que compõe o ambiente gerenciado (i.e. *hardware*, *software*, conexões, *endpoints*, sistemas externos) é representado como uma entidade virtual que considera sistemas distribuídos com elementos de borda, escalabilidade, heterogeneidade, consciência de contexto, disponibilidade e observabilidade. A representação virtual do objeto pode ser então compreendida como uma entidade no ambiente monitorado, fazendo com que o próprio sistema se torne também uma entidade com seus próprios parâmetros na solução implementada.

A implementação da IoTManA é apresentada em [Silva et al. 2022] denominada IoTManS (IoT Management System), um sistema desenvolvido com o propósito de demonstrar como a IoTManA pode ser implementada para gerenciar ambientes de IoT. o IoTManS visa garantir a disponibilidade de dados coletados pelos sensores, a correta operação da infraestrutura - componentes de *hardware* e *software*, comunicação entre diferentes componentes, sendo também capaz de analisar fluxos de dados e qualidade de dados. O IoTMan Agent é um módulo do IoTManS que atua como ferramenta de monitoramento e gerenciamento para que as técnicas implementadas pela IoTManA e IoTManS possam ser aplicadas no sistema gerenciado. Os trabalhos anteriores focam na apresentação da arquitetura, modelagem e implementação da IoTManA e IoTManS, não dando destaque ao funcionamento do agente. Este trabalho visa detalhar o funcionamento do IoTMan Agent, seu modo de operação e explanar como pode atender às necessidades de ambientes IoT, comparando-o a ferramentas de monitoramento populares.

Através de técnicas de monitoramento baseada em eventos, o IoTMan Agent

[1] https://www.docker.com/

analisa estados de aplicações e infraestruturas tratando cada componente monitorado como uma entidade única. O agente é capaz de observar o comportamento individual de cada entidade e analisar estas informações para formação de contextos, de estruturas de informação que fornecem que auxiliam no entendimento de cada componente e seu impacto na plataforma monitorada e nos demais componentes através de uma abordagem que utiliza um gráfico acíclico direcionado (Directed Acyclic Graph - DAG).

As principais contribuições deste trabalho são:

- Analisa e compara diferentes ferramentas de monitoramento com foco na sua implementação em ambientes de IoT;
- Propõe e exemplifica o funcionamento de uma ferramenta de monitoramento desenvolvida especificamente para operação em ambientes de IoT.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os fundamentos e trabalhos relacionados. Na seção 3 são apresentadas as plataformas avaliadas, seu critério de seleção e por fim, a avaliação qualitativa. A seção 4 apresenta a ferramenta IoTMan Agent e de que forma atende aos requisitos esperados para o funcionamento de uma ferramenta de monitoramento para IoT. Por fim, a seção 5 apresenta a conclusão.

2. Fundamentos e estado da arte

Entender o funcionamento e estado de plataformas, aplicações e sistemas permite aos desenvolvedores, gerentes e fornecedores a melhor compreensão de suas aplicações, infraestruturas, demandas, diagnóstico de falhas, qualidade de serviços (QoS). A compreensão do ambiente monitorado auxilia na previsão de futuras demandas, permitindo que haja a adequação das infraestruturas e serviços garantindo sua continuidade e estabilidade [Hauser and Wesner 2018]. Nesta seção, identificamos as características de uma ferramenta de monitoramento para que efetue estas funções, avaliando as características para operação em sistemas tradicionais e ambientes de IoT. É realizada ainda uma análise de trabalhos relacionados.

2.1. Ferramentas de Monitoramento

Ferramentas de monitoramento proprietárias são geralmente fortemente ligadas às suas plataformas de operação, ou até exclusivas como o AzureWatch[2] e AWS CloudWatch[3]. Quando consideramos a operação em sistemas tradicionais, ferramentas de monitoramento devem apresentar as seguintes características básicas [Yahia et al. 2021][Hauser and Wesner 2018]:

- Adaptável: apresenta capacidade de operar em diferentes ambientes;
- Oportuna: fornece os dados no momento em que há intenção de seu uso, reportando as leituras e eventos em tempo apropriado para tratamento e visualização;
- Autônoma: pode se auto gerenciar e se adaptar às possíveis variações da plataforma sem interrupção no fornecimento de dados;
- Resiliente, confiável e disponível: uma ferramenta é resiliente quando pode se adaptar e não interromper o serviço frente às mudanças ou falhas de componentes. A ferramenta é confiável quando pode executar sua função satisfatoriamente dentro de um intervalo de tempo com um SLA[4] aceitável;

[2]https://azure.microsoft.com/pt-br/

[3]https://aws.amazon.com/

[4]Service Level Agreement

- Precisa: apresenta precisão quanto as medidas fornecidas, refletindo a realidade;
- Portável: deve conseguir mover-se entre ambientes heterogêneos;
- Escalável: por se tratar de sistemas distribuídos, dinâmicos e elásticos, que podem adicionar dinamicamente novos objetos a serem monitorados, a ferramenta deve ter escalabilidade, permitindo que acompanhe o crescimento do ambiente monitorado.

Quando aplicadas à IoT, ferramentas de monitoramento deverão ainda considerar os seguintes aspectos [Recommendation ITU-T Y.4702 2016][Silva et al. 2022] :

- Dispositivos restritos: muitos dispositivos de IoT (sensores e atuadores) dispõem de limitados recursos de *hardware*. A ferramenta não deve demandar de seus agentes recursos mínimos para execução que são superiores aos recursos computacionais totais de dispositivos IoT ou que afetem seu desempenho e operação;
- Conectividade: dada a quantidade de elementos que compõem um ambiente IoT, é desejável que ferramentas de monitoramento possam identificar a conectividade entre dois componentes de maneira a entender de que forma um componente monitorado pode afetar os demais;
- Heterogeneidade: dispositivos e componentes de ambientes IoT são altamente heterogêneos, com diferentes tecnologias, protocolos, fornecedores e plataformas. Desta forma, uma ferramenta de monitoramento deverá ser adaptável para monitoramento de ambientes heterogêneos;
- Customização: dada a natureza heterogênea de ambientes IoT, é desejável que ferramentas permitam um nível de customização da forma como os dados são monitorados. Por exemplo, permitir que o usuário defina a interação da ferramenta com o componente monitorado ou forneça documentação e parâmetros para criação e modificação de agentes para interação com soluções desenvolvidas internamente.
- Operação distribuída: ambientes IoT operam geograficamente distribuídos e uma ferramenta de monitoramento deve ser capaz de coletar informações de diferentes pontos de observação;
- Operação não intrusiva: será não intrusiva se a sua utilização não demandar modificações na plataforma monitorada ou impactar diretamente no desempenho do sistema. Esta característica permite ainda o monitoramento de ambientes não controlados pelo agente de monitoramento, como APIs externas. Para adoção em ambientes IoT, o monitoramento não intrusivo pode fornecer métodos de coleta de dados em dispositivos restritos;
- Método de expansão: como novas tecnologias são criadas diariamente, é desejável que uma ferramenta de monitoramento de IoT permita a expansão de suas capacidades de maneira dinâmica, sem a necessidade de alteração de todo o ambiente. Podemos categorizar modelos de expansão em expansões realizadas pelo desenvolvedor, onde as capacidades da ferramenta dependem da intervenção do desenvolvedor através de atualizações e novas versões; e expansão dinâmica, onde são utilizados métodos como adoção de *plugins* que permitem a criação de novas funções na aplicação sem a necessidade de atualizações complexas ou do envolvimento do desenvolvedor;
- Visualização: É desejável que a ferramenta de monitoramento forneça uma interface gráfica amigável e de fácil utilização que permita a observabilidade do sistema monitorado que considere a complexidade de ambientes IoT;

Ferramentas de monitoramento devem coletar métricas que representam os estados do sistema monitorado para gerar informação. Aqui listamos algumas métricas básicas que a ferramenta deverá ser capaz de monitorar: 1) contêineres - dada a popularização e facilidade para implementação e desenvolvimento fornecido pela utilização de contêineres como o Docker[5], a adoção de aplicações em execução em contêineres em ambientes distribuídos são largamente utilizadas em soluções de IoT. É desejável que a ferramenta seja capaz de monitorar aplicações em execução dentro de contêineres; 2) métricas de sistema: coleta de dados de uso de CPU, memória, armazenamento, RX/TX de interfaces de rede e demais métricas de sistemas; 3) informações de rede: coleta de dados de estado de rede, conectividade e série histórica; 4) conectividade entre dispositivos: como ambientes IoT podem contar com milhares de dispositivos, entender o fluxo de dados e a conectividade entre elementos que formam este fluxo pode fortalecer a observabilidade dos ambientes monitorados; 5) processos e aplicações: é desejável que a ferramenta de monitoramento tenha capacidade de monitorar o estado de processos e aplicações, analisando sua disponibilidade e desempenho.

2.2. Trabalhos Relacionados

Existem várias propostas de ferramentas na literatura, que, no entanto, focam em problemas específicos. O trabalho realizado em [Brattstrom and Morreale 2017] propõe uma ferramenta totalmente não intrusiva que utiliza SNMP (*Simple Network Management Protocol*) para comunicação com os dispositivos monitorados. Em [Nagano et al. 2021] uma nova ferramenta denominada motch é apresentada com uma abordagem que fornece a ferramenta de monitoramento no modelo *Software as a Service* (SaaS) de maneira invasiva. O artigo [Raposo et al. 2018] propõe uma arquitetura para monitoramento focada em redes de maneira não intrusiva. Aplicando técnicas ainda menos intrusivas, um método que não depende de ferramenta de monitoramento com interação direta ao dispositivo é apresentado em [Santos et al. 2019] com uma proposta de monitoramento de redes IoT através da análise do fluxo de rede com o intuito de focar na identificação do conteúdo das mensagens sendo transmitidas, criando uma ferramenta de validação de identidade de origens e destinos de pacotes para aumento de segurança na transmissão.

A utilização de dados coletados por ferramentas de monitoramento pode ser observada em [Shahid et al. 2019], que aplica técnicas de inteligência artificial para detecção de ataques e identificação de vulnerabilidades. [Košťál et al. 2019] explora métodos de monitoramento de dispositivos IoT utilizando *blockchain* para aumento da segurança. Comparações de ferramentas podem ser encontradas em [de C. Silva et al. 2018], que realiza uma análise e comparação qualitativa de diferentes ferramentas de monitoramento de rede IoT, focando apenas no monitoramento de redes e não considerando demais componentes. Em [Mota et al. 2018] é realizada a análise de performance de diferentes protocolos aplicados em ferramentas IoT, como SNMP e MQTT (*Message Queuing Telemetry Transport*), focando apenas em protocolos e não considerando ferramentas.

3. Análise qualitativa de ferramentas de monitoramento

Através de análise pela plataforma de revisão de *softwares* Gartner [Insights 2022], selecionamos as 17 ferramentas de monitoramento melhor avaliadas em relação a seus modos

[5]https://www.docker.com/

de operação, métricas que são capazes de monitorar, sua adequação para atender os requisitos esperados de uma ferramenta de monitoramento, como apresentado na Seção 2, e de que forma elas se adaptam para a operação em ambientes de IoT.

A primeira observação desta avaliação é que todas as ferramentas estudadas apresentam capacidades e características de operação de monitoramento para sistemas tradicionais, como operação de maneira distribuída, permitem escalabilidade, fornecem um método de visualização de dados, e permitem a coleta de métricas de sistemas, redes, aplicações e processos.

Das ferramentas analisadas, apenas duas são de código livre, Zabbix e Nagios. Zabbix é mantido pela Zabbix LLC[6] e conta com uma comunidade ativa que constantemente auxilia em seu aprimoramento. O único requisito não atendido para monitoramento em IoT é a consideração por dispositivos restritos. Entretanto, o Zabbix oferece maneiras de contornar este problema através da criação de *endpoints* onde o dispositivo poderá publicar suas informações diretamente, substituindo a necessidade de um agente mas atribuindo esta responsabilidade para o desenvolvedor do dispositivo. Seu catálogo conta com mais de 4000 *plugins* que podem ser customizados para expansão das capacidades da ferramenta. O Nagios [7] conta também com uma grande comunidade que suporta e mantêm a aplicação. A ferramenta pode ser implementada em variadas plataformas e sua expansão através de *plugins* permite uma grande flexibilidade para a adequação a novos componentes. Nagios apresenta a mesma característica do Zabbix em relação à dispositivos com recursos restritos, sendo necessárias adaptações para seu funcionamento.

Dentre as opções proprietárias, a PRTG Network Monitor[8] oferece monitoramento de infraestruturas de *hardware* e *software*, com foco em monitoramento de redes, permitindo a personalização dos componentes monitorados. A ferramenta permite o monitoramento de contêineres e conectividade entre entidades monitoradas, mas não considera o monitoramento de dispositivos restritos. O método de expansão é dependente do desenvolvedor.

O VMWARE vRealize Operations[9] é focado no monitoramento de soluções e ambientes fornecidos por outros produtos da própria empresa, não considerando dispositivos restritos e limitada a plataformas específicas. Similarmente, o OnCommand Insight[10] também é direcionado para soluções específicas e não operando de maneira não intrusiva, não considerando dispositivos restritos e não sendo compatível com contêineres. Em ambas, a expansão é dependente do desenvolvedor.

As ferramentas Dynatrace[11] e DataDog[12] oferecem um catálogo de compatibilidade com mais de 500 componentes de *software* e infraestrutura compatíveis, considerando contêineres e ambientes heterogêneos. Sua expansão é dependente do desenvolvedor e não considera operação em dispositivos restritos, não sendo ainda capaz de operar de maneira não intrusiva.

[6]https://www.zabbix.com/
[7]https://www.nagios.org/
[8]https://www.paessler.com/
[9]https://www.vmware.com/br/products/vrealize-operations.html
[10]https://www.netapp.com/data-management/oncommand-insight/
[11]https://www.dynatrace.com/
[12]https://www.datadoghq.com/

O SCOM [13], não se limita a ambientes Windows, sendo capaz de operar em outras plataformas como Linux. O agente não pode monitorar contêineres ou dispositivos com recursos computacionais restritos. Por ter um escopo fechado, a ferramenta conta com funções sofisticadas como a organização dos componentes em forma de árvores de decisão para auxiliar na identificação de falhas. A expansão é dependente do desenvolvedor.

As ferramentas Icinga GmbH[14], OpManager[15], LogicMonitor[16], SolarWinds[17] e WhatsUpGold[18] tem propostas semelhantes, direcionadas ao monitoramento de ambientes de nuvem. As ferramentas não consideram fatores como heterogeneidade, dispositivos restritos e métodos para expansão de funcionalidades. A SolarWinds oferece algumas funcionalidades a mais em relação às demais, como capacidade de testar conectividade entre dois componentes monitorados e monitoramento de contêineres. A OpManager oferece o maior catálogo de compatibilidade e permite um nível mínimo de customização não presente nas demais. Nos três casos, as ferramentas necessitam de intervenção do desenvolvedor para expansão de funcionalidades.

As ferramentas Checkmk[19] e eG Enterprise [20] atendem boa parte dos requisitos de uma ferramenta de monitoramento, não sendo capaz apenas de operar em dispositivos restritos e de monitorar contêineres. Diferente da eGEnterprise que depende da expansão por parte do desenvolvedor, a Checkmk opera através de *plugins* e é capaz ainda de interagir com outros agentes.

Algumas ferramentas são direcionadas para problemas específicos. O Virtual Wisdom[21] é voltado para ambientes full-stack, com funções aperfeiçoadas para monitoramento de aplicações e cargas de trabalhos. A ferramenta não é capaz de operar de maneira não intrusiva. Em contraste com o Virtual Wisdom, o MicroFocus SiteScope[22] tem como premissa prover apenas monitoramento não intrusivo através de acessos SSH para coleta de métricas. De certa forma, esta solução considera coleta de métricas em dispositivos restritos, mas demanda configurações prévias dos ambientes assim como um gerenciamento de chaves de segurança que pode se tornar complexo ao considerarmos a escalabilidade de soluções IoT.

A ferramenta apresentada neste artigo, IoTMan Agent (IoTManager Agent), foi desenvolvida especificamente para operar em ambientes de IoT, sendo capaz de monitorar e gerenciar os seus diferentes aspectos considerando todas as suas características e complexidades apresentadas na Seção 2. A ferramenta pode coletar métricas de sistema, *hardware*, *software*, contêineres, considerando a implementação em ambientes heterogêneos, sendo ainda capaz de operar em dispositivos restritos nativamente (sem a necessidade de adequações), operando através de *plugins* e permitindo ainda a configuração e gerenciamento totalmente remotos, sem a necessidade de interação direta com o agente ou com

[13]https://docs.microsoft.com/pt-br/system-center/scom/
[14]https://icinga.com/
[15]https://www.manageengine.com/br/network-monitoring/
[16]https://www.logicmonitor.com/
[17]https://www.solarwinds.com/
[18]https://www.whatsupgold.com/
[19]https://checkmk.com
[20]https://www.eginnovations.com/
[21]https://www.virtana.com/products/virtualwisdom/
[22]https://www.microfocus.com/pt-br/products/

Tabela 1. Análise comparativa de ferramentas de monitoramento

	A	B	C	D	E	F	G	H	I	J	K	L	M
PRTG Network Monitor	X	X	X	X		X	X	X	X		dev	X	X
OpManager	X	X	X				X	X	X	X	dev	X	X
ZabbiX	X	X	X	X		X	X	X	X	X	plugins	X	X
DataDog	X	X	X			X	X	X		X	dev	X	X
SolarWinds	X	X	X	X			X	X	X		dev	X	X
Nagios IX	X	X	X			X	X	X	X	X	plugins	X	X
vRealize Operations		X	X				X	X			dev	X	X
SCOM		X	X			X	X	X	X		dev	X	X
Dynatrace	X	X	X			X	X	X	X		dev	X	X
WhatsUpGold		X	X				X	X	X		dev	X	X
Microfocus SiteScope	X	X	X				X	X	X		dev	X	X
Checkmk	X	X	X				X	X	X		dev	X	X
Icinga		X	X				X	X	X		dev	X	X
OnCommand Insight		X	X			X	X	X			dev	X	X
Virtual Winsdom	X	X	X			X	X	X			dev	X	X
LogicMonitor	X	X	X				X	X	X		dev	X	X
eG Enterprise		X	X			X	X	X	X		dev	X	X
IoTMan Agent	X	X	X	X	X	X	X	X	X	X	plugins	X	X

o dispositivo para que mudanças e atualizações sejam repassadas para o ambiente. A descrição completa do ambiente de criação é apresentada em [Silva et al. 2022], onde toda a arquitetura e funcionamento do sistema de gerenciamento para IoT é apresentado. Neste trabalho, apresentamos na sessão 4 de que maneira o IoTMan Agent atende aos requisitos de uma aplicação de monitoramento de ambientes IoT.

A Tabela 1 resume as avaliações das ferramentas avaliadas em relação às características apresentadas e de que forma a ferramenta as atende. As características avaliadas são: a) monitoramento de contêiner; b) monitoramento de sistema; c) monitoramento de redes; d) monitoramento da conectividade entre dois componentes; e) consideração de operação em dispositivos restritos; f) consideração de questões de heterogeneidade; g) consideração de questões de escalabilidade; h) operação distribuída; i) monitoramento de forma não intrusiva; j) customização de *endpoints* e aplicações monitoradas; k) abordagem da questão de expansão de funcionalidades; l) fornecimento de métodos de visualização dos dados; e m) monitoramento de processos e aplicações.

Algumas das ferramentas avaliadas apresentam muitas das características necessárias para operação em ambientes de IoT. As plataformas Zabbix, Nagios, Datadog, Dynatrace e Checkmk apenas não consideram o monitoramento em ambientes de poucos recursos computacionais, sendo as mais próximas de atenderem a todos os requisitos. Entretanto, de todas as ferramentas avaliadas, apenas o Nagios e Zabbix fornecem meios de contornar esta restrição. Por se tratar de soluções de código livre que operam através de *plugins*, é possível a adaptação da ferramenta para operação em dispositivos restritos. Portanto, avaliamos as ferramentas Nagios e Zabbix como as mais próximas de se adequarem ao monitoramento de ambientes IoT. Das ferramentas avaliadas, a IoTMan Agent é a única ferramenta que atende a todos os requisitos necessários para monitoramento de ambientes IoT.

4. IoTMan Agent - Arquitetura e Funcionamento

Criado durante a implementação do projeto SWAMP[23], que desenvolveu uma solução baseada em IoT para o gerenciamento inteligente de sistemas de irrigação de precisão [Zyrianoff et al. 2020], o IoTManS (IoT Management System)[Silva et al. 2022] divide suas funções em módulos: IoTManager Server - responsável pela gerência e visualização de dados da plataforma; IoTMan Agent - ferramenta de monitoramento de sistemas; IoT Entity Editor - ferramenta capaz de realizar operações CRUD de entidades em uma plataforma baseada em FIWARE[24]; IoT Sensor Setup - ferramenta que realiza a configuração automatizada de sensores para ingresso na plataforma de IoT. A aplicação de todos os módulos permitem o monitoramento e configuração de dispositivos, entidades, infraestruturas e gerência de uma plataforma de IoT.

A ferramenta de monitoramento IoTMan Agent (IoTManager Agent) opera em multiplataformas, ambientes de nuvem e névoa computacional, *bare metal* e dispositivos IoT. Além das capacidades já apresentadas de monitoramento tradicionais de métricas de sistema e processos, a ferramenta tem capacidade de analisar a disponibilidade de aplicações através de testes de checagem de saúde de cada aplicação monitorada, não apenas checando um *endpoint*, mas sendo capaz de consumir a aplicação com testes de maneira a garantir sua disponibilidade, e ainda checar a conectiva de entre dois componentes através da correlação do estado de cada ponta da conexão, sendo possível ainda a adoção de *sniffers* para detecção da disponibilidade de conexão entre os componentes.

Uma vez coletados, os dados são publicados para um FIWARE Context Broker Orion, utilizado por ser capaz de gerenciar simultaneamente um grande número de mensagens de estrutura complexa que contêm informações de contexto. A cada mensagem, o Orion encaminha uma notificação para o IoTMan Server onde os dados serão analisados, tratados e armazenados de maneira estruturada em um banco de dados não relacional. As informações podem então ser utilizadas de maneira contextual para análise, tomadas de decisão e auto gerenciamento da plataforma. Apesar da utilização da plataforma FIWARE para implementação do IoTMan Server e IoTMan Agent, as ferramentas podem ser implementadas em outras plataformas, não sendo tecnologicamente dependentes do FIWARE. O estudo apresentado em [Silva et al. 2022] demonstra uma implementação realizada em uma plataforma ThingsBoard para o gerenciamento de dados. A adoção da abordagem cliente/servidor permite ao IoTMan Agent operar de maneira distribuída, sendo possível a adoção de múltiplos agentes e servidores. Por fornecer esta característica de portabilidade, o desempenho da ferramenta é diretamente relacionado ao desempenho da plataforma IoT utilizada. O trabalho apresentado em [Silva et al. 2022] analisa o desempenho da ferramenta com a utilização de diferentes plataformas IoT.

4.1. Modo de operação

O IoTMan Agent opera com dois subagentes, IoTMan Agent Sys e IoTMan Agent App. IoTMan Agent Sys: desenvolvido em C++, o agente de monitoramento opera de forma intrusiva e é capaz de analisar dados de uso de memória, CPU e ainda monitoramento de processos, sendo capaz ainda de monitorar o uso de CPU e memória de contêineres Docker. O agente se mostrou leve a ponto de ser capaz de operar em dispositivos com

[23]http://www.swamp-project.org
[24]https://www.fiware.org/

menor capacidade de processamento, como Raspberry PI e Arduino, sem afetar as demais aplicações, e versátil a ponto de poder operar em ambientes de névoa e nuvem computacional. IoTMan App: o agente de monitoramento de aplicações pode operar de maneira não intrusiva e pode também ser implementado em contêiner Docker, aumentando sua portabilidade. Desenvolvido em *typescript* sob um *framework* NodeJS[25].

A Figura 1 ilustra como o IoTMan Agent opera em um ambiente de IoT em uma abordagem distribuída que considera nuvem e elementos de borda. A ferramenta pode monitorar aplicações, indicado pelo número 1, verificando seu status e disponibilidade. A disponibilidade pode ser checada através do consumo da aplicação, como uma requisição ao banco de dados se o componente monitorado for um a aplicação de banco de dados. Este recurso foi adicionado considerando que aplicações podem retornar falsos positivos em uma checagem de status. Desta forma, aumentamos a confiabilidade da informação coletada.

O monitoramento de comunicações, indicado pelo número 2 na Figura 1, pode determinar se a conexão entre dois elementos está ativa e operacional através da checagem do estado de cada um dos componentes que formam a conexão, sendo possível ainda a implementação de *sniffers* para garantia de que há a troca de informações entre os dois componentes. O agente pode também monitorar conexões entre sistemas e infraestruturas, indicado pelo número 3 na Figura 1, coletando métricas como latência e disponibilidade da conexão. O monitoramento não invasivo, indicado pelo número 4 na Figura 1, permite que um agente seja implementado de maneira remota e monitore uma aplicação externa, *endpoints*, ou qualquer outro componente que seja compatível com este tipo de monitoramento.

Dispositivos IoT podem ser monitorados diretamente através do agente intrusivamente, ou ainda utilizando técnicas de monitoramento de *clusters* de dispositivos, que permitem que um único dispositivo, ou um grupo de dispositivos, represente o estado dos demais dispositivos que fazem parte de um mesmo *cluster*[Shao et al. 2018].

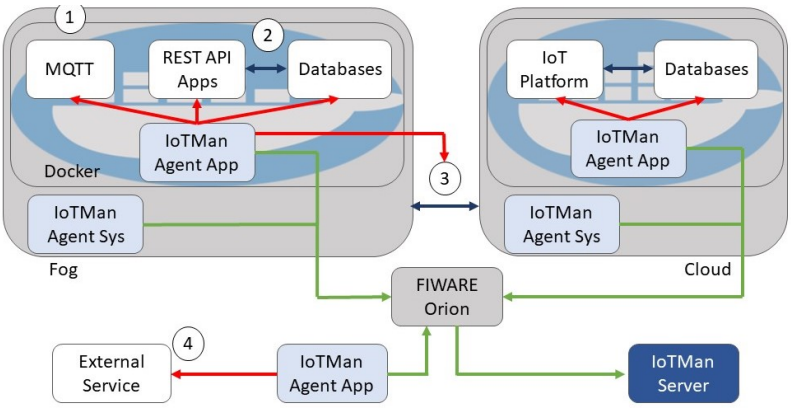

Figura 1. Arquitetura do agente de monitoramento

A implementação do agente de monitoramento é realizada através de um arquivo de configuração em formato JSON que apenas identifica o endereço do IoTMan Server que contém as configurações e sua identidade. Com estas informações, o agente inicia

[25]https://nodejs.org/en/

seus processos e busca no IoTMan Server uma entidade de configuração, um objeto JSON que fornece as informações completas sobre quais componentes devem ser monitorados, de que forma serão monitorados, a frequência e o endereço para os quais as informações de cada componente devem ser enviadas.

A entidade de configuração fornece informações para o agente sobre quais métricas serão monitoradas, o tipo de entidade, o endereço do Orion Context Broker que receberá as informações desta entidade e o ID da entidade. Para trazer ainda mais versatilidade para a ferramenta, o arquivo de configuração dispõe de um campo que indica a frequência com que este arquivo de configuração deverá ser lido pelo agente de monitoramento. Desta forma, é possível realizar alterações no agente, como atualizações e mudanças de configurações, sem a necessidade de acesso direto, trazendo a característica de auto gerência. É possível ainda o envio de diferentes ações e comandos para que o agente realize alterações no ambiente monitorado, além da execução de *scripts* pré-configurados. Esta característica permite o envio de informações, comandos e até mesmo solicitação de atualizações de *plugins* para agentes que operam de modo isolado, e que não podem ser acessados diretamente - como agentes operando em redes internas que não tem endereços públicos ou opções de NAT (*Network Address Translation*);

Em ambientes com limites de conexão, o agente pode ser configurado para armazenamento local de dados, que podem então ser coletados através de interação direta com o dispositivo ou encaminhando uma série histórica de dados ao invés de dados pontuais.

O IoTMan Agent utiliza *plugins* para realizar o monitoramento. Desta forma, para adicionar compatibilidade com uma nova aplicação, basta criar um novo *plugin*. Um arquivo JavaScript deverá ser incluído no diretório *Plugins* da aplicação junto a um arquivo JSON que tem estrutura de chave-valor contendo informações de: nome do *plugin*; método de requisição; nome e caminho do arquivo JavaScript para importação da função de validação. Cada função de validação foi desenvolvida através da consulta de documentação de cada *software* compatível com a versão atual do IoTManager. Esta abordagem permite ainda a adição de novos *plugins* sem interrupção da aplicação, dadas as características de módulos que adicionam essa capacidade ao *framework* NodeJS.

Em termos de escalabilidade, o trabalho experimental apresentado em [Silva et al. 2022] demonstra que o IoTManager é capaz de gerenciar mais de 5000 componentes sendo monitorados simultaneamente e enviando dados a cada 60 segundos. Nos mesmos experimentos, o IoTMan Agent se mostrou leve o bastante para operar em dispositivos restritos, demandando baixo uso de CPU e memória, podendo ainda consumir poucos recursos de rede com a utilização de protocolos como o Ultralight, que permite o envio de apenas alguns bytes ao invés de objetos JSON complexos.

4.2. Interação com IoTManager Server

A aplicação de *backend* do IoTManager (IoTManager Server) centraliza e analisa os dados recebidos pelos múltiplos agentes e módulos, permitindo o armazenamento de dados históricos, visualização dos dados e a adoção de ferramentas de gerenciamento de eventos.

Adotando o contexto de eventos, o IoTManager Server é capaz de analisar os dados assim que os recebe através da utilização de regras. Regras de monitoramento são capazes de analisar um valor específico de determinada entidade e compará-la com um

valor limite (*threshold*) que será utilizado como métrica para acionar ou não a regra, sendo possível a definição de um número de ocorrências mínimas. Um exemplo que mostra a versatilidade desta opção é um cenário onde o agente tem uma conexão de baixa qualidade e deve enviar medições a cada 1 minuto. Caso a baixa qualidade cause interrupção na transmissão, o *backend* dispara um evento de erro informando que a mensagem que era esperada para este intervalo de tempo não foi recebida. Uma vez que elevamos o valor de ocorrências para 3, por exemplo, a ferramenta apenas dispara o evento quando este ocorrer 3 vezes, reduzindo assim os falsos positivos.

Ao acionar uma regra, um evento em formato JSON é gerado informando: tipo e ID do evento, necessários para o padrão NGSI; o tipo do evento (falha, erro, alerta, entre outros); a entidade que acionou a regra; o ID da regra que foi acionada; a mensagem (se houver); ação que será disparada pelo gerenciador de eventos, devendo incluir o tipo de ação e o alvo da ação, se aplicável, podendo incluir uma ação do tipo "Notificar usuário"e qual usuário notificar e de que forma, se por email ou um alerta na *dashboard*.

Os eventos são encaminhados para o gerenciador de eventos do IoTManager Server, uma aplicação desenvolvida em NodeJS capaz de receber as mensagens no padrão NGSI através de uma API REST e realizar as ações necessárias através da análise do evento gerado. Assim como o agente de monitoramento de aplicações, o Event Handler é capaz de operar através de *plugins*, permitindo assim a fácil expansão para diferentes eventos com o decorrer das necessidades.

O IoTManager Server organiza os elementos monitorados em forma de grafo direcionado acíclico (DAG), que permite a organização de todas as entidades monitoradas, bem como enriquecendo a observabilidade do sistema através do uso de ferramentas de checagem de dependências entre componentes, i.e. um *software* executado em um servidor é dependente da disponibilidade do servidor no qual é executado. Um estudo sobre estas funcionalidades é apresentado em [Silva et al. 2022].

Uma *dashboard* de visualização de métricas coletadas pelos agentes foi desenvolvida para a visualização dos dados monitorados e a organização das aplicações em forma de grafos. A Figura 2 mostra a *dashboard* desenvolvida para visualização dos componentes monitorados representados em forma de grafo. As conexões e dependências entre as entidades são organizadas em forma de DAGs, que estabelecem o fluxo de dados fim a fim desde o sensor até a nuvem. Esta abordagem permite a identificação não apenas de todo o fluxo de dados dentro do ambiente monitorado, mas também a identificação e isolamento de falhas através da identificação de dependências entre elementos. Cada elemento pode ser expandido e tem seus próprios dados de CPU, memória, últimas mensagens, além de dados estatísticos das amostras coletadas.

5. Conclusão e Trabalhos futuros

Ferramentas de monitoramento tem um papel essencial na garantia de disponibilidade, qualidade de serviço e detecção de falhas. Uma vez que ambientes de IoT apresentam características específicas quando comparadas a sistemas tradicionais. Através da análise qualitativa apresentada neste artigo, identificamos que as ferramentas de monitoramento disponíveis no mercado e na academia não atendem estas necessidades.

O IoTMan Agent foi desenvolvido especificamente para atender as necessidades de ambientes de IoT, considerando problemas como escalabilidade, gerenciamento dis-

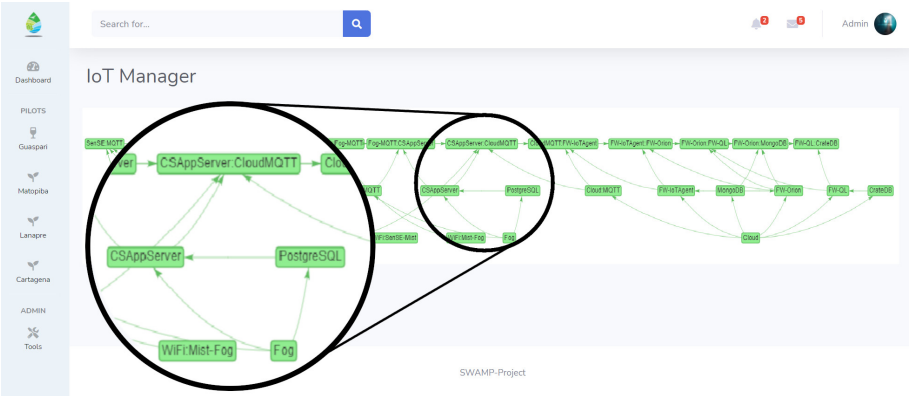

Figura 2. IoTManager Dashboard

tribuído, heterogeneidade e dispositivos restritos. Operando de maneira distribuída e com múltiplas formas de implementação (sys, app, contêiner, Docker), a ferramenta se adapta a diferentes cenários devido a sua característica de operar em módulos e de fornecer métodos para criação de novas funcionalidades, tornando-se capaz de monitorar diversas novas aplicações com um mínimo de esforço empregado e sem a necessidade de reconfiguração do sistema ou mudança do código da ferramenta. Seu modo de funcionamento permite que novas informações sejam passadas aos agentes, mesmo que operando de maneira isolada, ou seja, sem uma conexão direta que permita o envio destas informações para o agente.

Trabalhos futuros visam explorar e expandir as funcionalidades presentes no agente, adicionando capacidades de gerenciamento e orquestração de aplicações em sistemas distribuídos entre componentes de borda e nuvem computacionais. Uma vez que a ferramenta traga estas funcionalidades, as informações obtidas poderão enriquecer as operações de gerenciamento operadas pelo IoTManager, visando a criação de um ambiente de IoT que forneça características de auto-gerenciamento e que não seja vinculado a uma tecnologia específica, podendo ser implementada em diferentes ambientes de IoT.

Referências

Brattstrom, M. and Morreale, P. (2017). Scalable agentless cloud network monitoring. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 171–176.

de C. Silva, J., M. Pereira, P. H., de Souza, L. L., M. Marins, C. N., Marcondes, G. A., and Rodrigues, J. J. (2018). Performance evaluation of iot network management platforms. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 259–265.

Hauser, C. B. and Wesner, S. (2018). Reviewing cloud monitoring: Towards cloud resource profiling. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 678–685.

Heideker, A., Ottolini, D., Zyrianoff, I., Kleinschmidt, J., and Kamienski, C. (2019). Imaiot - infrastructure monitoring agent for iot: Um agente monitor de infraestruturas para ambientes de iot. In *Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 9–16, Porto Alegre, RS, Brasil. SBC.

Insights, G. P. (2022). It infrastructure monitoring tools reviews and ratings.

Košťál, K., Helebrandt, P., Belluš, M., Ries, M., and Kotuliak, I. (2019). Management and monitoring of iot devices using blockchain. *Sensors*, 19(4).

Mota, L. C., Moreno, E. D., and Ribeiro, A. L. (2018). A comparative analysis of protocols for iot network management. In *Proceedings of the Euro American Conference on Telematics and Information Systems*, EATIS '18, New York, NY, USA. Association for Computing Machinery.

Nagano, M., Arai, Y., Fujihashi, T., Watanabe, T., and Saruwatari, S. (2021). Design and implementation of device monitoring saas for diy-iot systems. In *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4.

Raposo, D., Rodrigues, A., Sinche, S., Sá Silva, J., and Boavida, F. (2018). Industrial iot monitoring: Technologies and architecture proposal. *Sensors*, 18(10).

Recommendation ITU-T Y.4702 (2016). Common requirements and capabilities of device management in the Internet of things. Standard, TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU.

Santos, L., Rabadão, C., and Gonçalves, R. (2019). Flow monitoring system for iot networks. In Rocha, Á., Adeli, H., Reis, L. P., and Costanzo, S., editors, *New Knowledge in Information Systems and Technologies*, pages 420–430, Cham. Springer International Publishing.

Shahid, M. R., Blanc, G., Zhang, Z., and Debar, H. (2019). Machine learning for IoT network monitoring. In *RESSI 2019: Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*, RESSI 2019 Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information, pages 1–3, Erquy, France.

Shao, X., Yang, C., Chen, D., Zhao, N., and Yu, F. R. (2018). Dynamic iot device clustering and energy management with hybrid noma systems. *IEEE Transactions on Industrial Informatics*, 14(10):4622–4630.

Silva, D., Heideker, A., Zyrianoff, I., Kleinschmidt, J., 1Soininen, J.-P., Roffia, L., and Kamienski, C. (2022). "A Management Architecture for IoT Smart Solutions: Design and Implementation". *"Journal of Network and Systems Management*, 30.

Yahia, H. S., Zeebaree, S., Sadeeq, M., Salim, N., Kak, S. F., Adel, A., Salih, A. A., and Hussein, H. A. (2021). Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. *Asian Journal of Research in Computer Science*, 8(2):1–16.

Zyrianoff, I., Heideker, A., Silva, D., Kleinschmidt, J., Soininen, J.-P., Salmon Cinotti, T., and Kamienski, C. (2020). Architecting and deploying iot smart applications: A performance–oriented approach. *Sensors*, 20(1).