

A Scalable Multiagent Architecture for Retrieving and Updating Vinculated Information

Uma Arquitetura Escalável para Recuperação e Atualização de Informações Vinculadas

Rocha, V.*; Brandão, A. A. F.*

*Escola Politécnica - Universidade de São Paulo

E-mail: vmoreira@ime.usp.br, anarosa.brandao@poli.usp.br

Abstract—Recently, there has been an explosive growth in the use of wireless devices, mainly due to the decrease in cost, size, and energy consumption. Researches into different domains, such as Internet of Things and Video on Demand, have focused on how to continuously monitor these devices considering both scalability and efficiency while searching and updating the devices information. For this, a combination of an efficient distributed structure and data aggregation method is used, allowing a device to manage a group of devices, minimizing the number of transmissions and saving energy. However, scalability is still a key challenge when the group is composed of a large number of devices. In this work, we propose a scalable architecture that distributes the data aggregation responsibility to the devices of the boundary of the group, and creates new agents to manage groups, and the interaction among them, when a agent is overloaded. Experimental results showed the viability of adopting this architecture if compared with the most widely used approaches.

Keywords— Multiagent systems; MAS; DHT; VoD; IoT.

Resumo— A imensa quantidade de dispositivos móveis conectados à Internet tem provocado um crescimento na geração e armazenamento distribuído de informações, como a localização dos dispositivos ou os arquivos que compartilham. Nesse contexto, a busca deve ser eficiente para trazer os resultados rapidamente e escalável para suportar o volume crescente de tráfego de dados gerado pela atualização e busca dessas informações. As alternativas atuais combinam tecnologias bem estabelecidas e amplamente conhecidas para tratar questões relacionadas à eficiência: agregação de dados para atualizar as informações e Tabelas de *Hash* Distribuídas (DHT) para buscar a informação. Entretanto, ainda existe o problema da escalabilidade quando um dispositivo da estrutura se sobrecarrega pelo recebimento e processamento das informações do grupo de dispositivos que gerencia. Este projeto de doutorado visa desenvolver uma arquitetura multiagentes que localize de forma eficiente e escalável as informações armazenadas de forma distribuída nos dispositivos, independente do domínio em que for utilizada. Para aumentar a escalabilidade, tanto o método de agregação quanto a estrutura distribuída foram estendidas. No caso da agregação, a responsabilidade da coleta e atualização foi distribuída para todos os dispositivos que fazem parte da fronteira do grupo. No caso da DHT, as estruturas internas foram estendidas para distribuir as requisições entre os diversos dispositivos. A utilização da arquitetura, no domínio de sistemas de Vídeo sob Demanda (VoD) e de sistemas de sensores sem fio, no contexto de Internet das Coisas, mostrou-se viável tanto na escalabilidade quanto na eficiência na recuperação e atualização de informações.

Palavras-chave— Sistemas multiagentes; DHT; VoD; IoT.

I. INTRODUÇÃO

Nos últimos anos houve um aumento significativo na utilização de dispositivos sem fio conectados a Internet, devido principalmente à diminuição do custo, tamanho e consumo de energia desses dispositivos. De acordo com o relatório da Gartner, em 2016 haverá mais de 4.9 bilhões desses dispositivos, chegando a 25 bilhões em 2020 [1]. Como consequência desse aumento, houve uma explosão na produção de informação. Diariamente são produzidos milhares de Petabytes de informação em fotos, vídeos, documentos, entre outros.

Nesse contexto, diversas pesquisas têm se focado em criar tecnologias que permitam realizar uma busca por essas informações de forma eficiente, para trazer os resultados rapidamente, e escalável para suportar o volume crescente de tráfego de dados gerado pela atualização e busca dessas informações. Dentre as alternativas atuais mais utilizadas, duas se destacam em tratar essas questões: agregação de dados (*Data Aggregation*) [2] para atualizar as informações e a estrutura denominada Tabela de *Hash* Distribuída (DHT) [3] para buscar a informação. Na agregação de dados, escolhe-se um dispositivo responsável por coletar e gerenciar as informações de todos os dispositivos que formam um grupo, minimizando a quantidade de informações trafegadas. Na estrutura distribuída, os dispositivos se organizam para recuperar, de forma eficiente, o dispositivo escolhido para agregar dados, quando outros dispositivos querem se unir a um grupo ou encontrar uma determinada informação. No entanto, ainda existem problemas em aberto, especificamente no ponto único de falha e na escalabilidade.

Por um lado, o problema do ponto único de falha surge quando o dispositivo escolhido interrompe sua responsabilidade de agregar os dados devido a um defeito (e.g., pelo consumo total da energia, perda do sinal em certas regiões, entre outras) que devem ser consideradas no comportamento do dispositivo. Como consequência, as informações do grupo que gerencia ficam inacessíveis até que um novo dispositivo seja escolhido. Por outro lado, o problema da escalabilidade surge quando o grupo aumenta a quantidade de membros, seja porque novos dispositivos se uniram ao grupo ou porque diver-

os grupos se uniram entre si. Esse incremento faz com que o dispositivo fique sobrecarregado na coleta de informações até o ponto em que não consegue mais gerenciar o grupo. Como consequência, as informações do grupo ficam inacessíveis inclusive quando um novo dispositivo é escolhido.

Esse trabalho de doutorado visa desenvolver uma arquitetura escalável para a atualização de informações advindas de um grupo de dispositivos, mantendo a eficiência das alternativas atuais quando realizada uma busca. A arquitetura é composta de duas camadas, a de dispositivos e a de agentes. A camada de dispositivos consiste em grupos de dispositivos conectados. Para manter a escalabilidade no processo de atualização de informação, o método de agregação foi estendido, onde cada grupo atribui a responsabilidade por agregar e enviar suas informações aos dispositivos da borda. A camada multiagentes consiste em agentes responsáveis por gerenciar as informações dos grupos e por verificar se existe alguma interação com outros grupos, tais como uniões e separações. Para manter a escalabilidade no processo de atualização de informações, a estrutura DHT foi estendida, onde os agentes podem criar novos agentes (agentificando os dispositivos) para distribuir a responsabilidade de gerenciamento do grupo. No fim, apresentamos os resultados obtidos através de simulações da arquitetura que foi implementada no domínio dos VoD [4] e de sensores de localização no contexto de Internet das Coisas [5].

II. CENÁRIOS

A arquitetura a ser desenvolvida poderá ser implantada em diferentes domínios. A característica comum desses domínios é que a informação a ser gerenciada está vinculada a outras informações. A seguir serão mostrados os dois cenários analisados nesse trabalho. O primeiro cenário são os sistemas de VoD, que permitem aos usuários assistirem um vídeo, cujo conteúdo foi particionado em segmentos. Nesse contexto, um segmento está fortemente vinculado ao segmento seguinte e ao anterior (pois os usuários que visualizaram um segmento tendem a assistir o próximo).

O segundo cenário é composto por sistemas de localização de sensores, que permitem monitorar o movimento destes em determinadas regiões do mapa. Nesse contexto, uma região está fortemente vinculada às regiões que a circundam (pois os sensores podem se movimentar entre elas).

III. TRABALHOS RELACIONADOS

Nesta seção são revisados os trabalhos que utilizam a agregação de dados para escolher o responsável por gerenciar um grupo e os trabalhos que utilizam a DHT para recuperar de forma eficiente esse responsável.

A. Agregação de Dados

Pesquisas recentes no método de agregação de dados têm se focado em como explorar o método para apoiar a descoberta e atualização de informações de um grupo de dispositivos de forma eficiente e escalável. Gallucio et al. [6] propõe que um dispositivo (denominado *Group Master* ou GM) seja o responsável por coletar, agregar e enviar as informações de

todos os dispositivos (denominados *slaves*) que residem dentro da sua área de cobertura. Essa abordagem foi avaliada em [7], onde um grande número de simulações foram realizadas em diferentes cenários, como logística e de movimentação de pessoas. Por outro lado, Tang et al. [8] propõe agrupar os dispositivos em setores retangulares pré-definidos (e não pela área de cobertura como nos trabalhos anteriores), utilizando uma estrutura de árvore distribuída, denominada ECH-Tree, para localizá-los de forma eficiente. Nosso trabalho se diferencia na distribuição da responsabilidade da agregação entre os dispositivos que formam a borda do grupo.

B. DHT

Pesquisas recentes na estrutura DHT têm se focado em como escolher o GM responsável por um setor para evitar ter que substituí-lo constantemente. A substituição é necessária pois cada ação realizada pelo dispositivo (e.g., coleta de informações) consome energia até o ponto de deixá-lo inutilizável. Para uma DHT, evitar a substituição de um membro é crítico, pois esse comportamento tem como consequência a reorganização da estrutura, gastando recursos enquanto é realizada. O sistema *Chord for Sensor Networks* (CSN) [9] propõe escolher aleatoriamente um GM dentre os dispositivos que residem em um determinado setor. A substituição do GM é realizada antes do dispositivo consumir toda sua energia. Quando isso ocorre, o sistema escolhe outro aleatoriamente, que será o novo GM (processo denominado de rotação). O sistema *Tiered Chord System* (T-Chord) [10], por outro lado, analisa as propriedades de todos os dispositivos que residem em um determinado setor (propriedades como energia, estabilidade na comunicação, entre outras), escolhendo aquele que tiver o maior valor. Entretanto, o sistema não permite a rotação do GM, que pode deixar indisponíveis as informações de um setor caso o GM consuma toda sua energia. Nosso trabalho se diferencia por permitir que vários GM possam gerenciar um determinado setor, evitando o ponto único de falha e sobrecarga do GM.

IV. ARQUITETURA

A Figura 1 apresenta a arquitetura, composta de duas camadas, a de dispositivos e a de agentes. Na camada de dispositivos, aqueles que pertencem à borda do grupo estabelecem conexões entre si para decidir quais serão responsáveis por agregar e enviar as informações do grupo para o agente que o gerencia (dispositivos d_1 , d_2 e d_3 do grupo S_1). Na camada multiagentes, cada agente monitora o comportamento do seu grupo (agente a_1 gerencia o grupo S_1). Nesta camada, os agentes estabelecem relacionamento entre si, trocando informações sobre seus grupos, i.e., se estão se unindo (o agente a_2 troca informações com o agente a_3 sobre a união dos seus grupos S_2 e S_3) ou divisão (agente a_4 troca informações com os agentes a_2 e a_3 sobre a divisão do grupo S_4).

A. Camada de Dispositivos

A camada de dispositivos consiste de dispositivos interconectados onde cada um tem seu próprio movimento, mas se

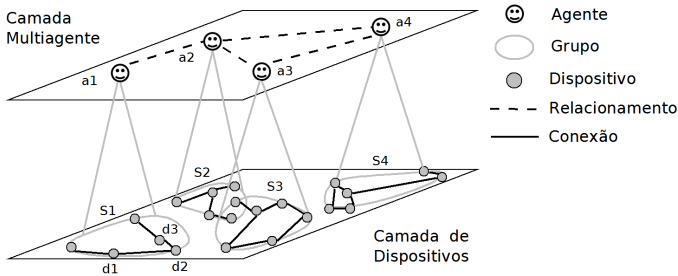


Figura 1. Arquitetura e suas camadas.

visto como um todo, eles se movimentam em grupos. A Figura 2 mostra o detalhe de um grupo dessa camada, utilizando o sistema de coordenadas \mathbb{R}^2 por simplicidade. Na figura, o grupo é composto por uma grande quantidade de dispositivos. A borda do grupo está representada como o conjunto de dispositivos dH_i , com i variando de 1 a 18, onde, para cada par consecutivo de dispositivos ($dH_i, dH_{i+1 \bmod N}$), existe uma conexão entre eles. A fronteira do grupo é o conjunto de dispositivos dB_k , com k variando de 1 a 16, que residem na zona de influência (área de raio r) de algum dispositivo pertencente à borda (e.g., a localização do dB_3 reside na zona de influência do dH_2).

Nesta camada, para manter a escalabilidade, somente os dispositivos que pertencem à fronteira serão os responsáveis por executar determinados processos relacionados com o comportamento do grupo. Nesse contexto, primeiro analisaremos como determinar se um dispositivo é da fronteira e, depois, quais são os processos que estes dispositivos devem executar.

Cada dispositivo do grupo deve determinar se pertence à fronteira em dois momentos, (i) quando se une ao grupo; (ii) e durante seu ciclo de vida. No primeiro caso, quando um dispositivo d se une ao grupo, o agente responsável por ele lhe proverá uma lista L de dispositivos que estão localizados próximos a d e que já pertencem à fronteira ou à borda (o agente conhece todos os dispositivos da fronteira e da borda do grupo que gerencia). Logo, d seleciona de L aqueles dispositivos d_i que pertencem à borda e verifica se algum deles reside na sua zona de influência. Se isso ocorre, o dispositivo d se declara ao agente como pertencente à fronteira. No segundo caso, durante o ciclo de vida, quando d muda sua posição, este deve perguntar aos dispositivos que residem na sua zona de influência se algum deles pertence à borda. Se isso acontecer, o dispositivo se declara ao agente como pertencente à fronteira.

O dispositivo que pertencer à fronteira deve ser responsável por executar os seguintes processos: (1) atualizar a localização e movimento da fronteira e agregar a localização dos dispositivos que residem na sua zona de influência, enviando essa informação ao agente responsável pelo grupo; (2) perceber se a fronteira está se desunindo, tentando evitar a desconexão da borda e avisando dessa situação ao agente responsável; (3) eliminar as conexões com os dispositivos que saíram da sua zona de influência e estabelecer conexões com os dispositivos que entraram na zona; (4) agentificar um dispositivo para que gerencie o grupo, caso não haja um agente cumprindo essa

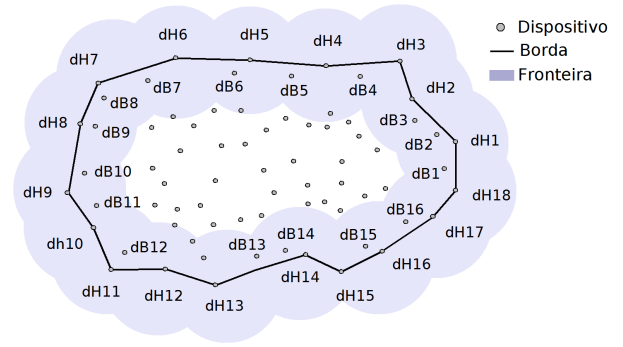


Figura 2. Camada de dispositivos.

função. Cabe destacar que os processos 1 ao 3 podem ser realizados em paralelo.

B. Camada de Agentes

A camada multiagentes consiste em agentes responsáveis por gerenciar grupos de dispositivos. Cada agente possui dois tipos de relacionamentos com outros agentes: sequenciais e de saltos. Relacionamentos sequenciais são estabelecidos para manter a escalabilidade, permitindo aos agentes próximos entre si (i.e., aqueles cujos grupos estão próximos de se unirem) trocarem informações. Relacionamentos de saltos são estabelecidos para manter a eficiência e escalabilidade, permitindo aos agentes acessar a localização de grupos que estão fora do alcance dos relacionamentos sequenciais.

Cada agente desta camada é responsável por executar os seguintes processos: (1) dado uma localização l , enviada por um dispositivo que quer se unir a um grupo, procurar o agente que gerencia o grupo localizado em l ; (2) monitorar o comportamento do grupo, eliminando aqueles dispositivos da fronteira e da borda que não estejam cumprindo com suas responsabilidades; (3) receber a localização dos dispositivos da fronteira, atualizando a área coberta pelo grupo; (4) perceber se o grupo está se unindo com outros grupos, utilizando para isso os relacionamentos sequenciais, e unindo-os caso seja necessário; (5) perceber se o grupo está se desunindo, tentando prevenir a divisão; (6) dividir o grupo em dois, caso não seja possível prevenir a divisão no passo anterior. Cabe destacar que os processos 2, e 4 ao 6 podem ser realizados em paralelo.

V. RESULTADOS EXPERIMENTAIS

A escalabilidade e a eficiência da arquitetura foram testadas para os dois cenários mencionados na Seção II.

A. Video sob Demanda

No VoD, a eficiência foi analisada no tempo necessário para um membro da DHT responder às requisições pela busca de um determinado segmento de vídeo, onde o vídeo foi dividido em 1000 segmentos de 1024 kb cada um. Nesse contexto, em um determinado momento, todos os nós que pertenciam à DHT enviaram uma requisição de busca pelo mesmo segmento. A Figura 3 compara a alternativa da DHT (onde o nó tenta responder todas as requisições, diminuindo

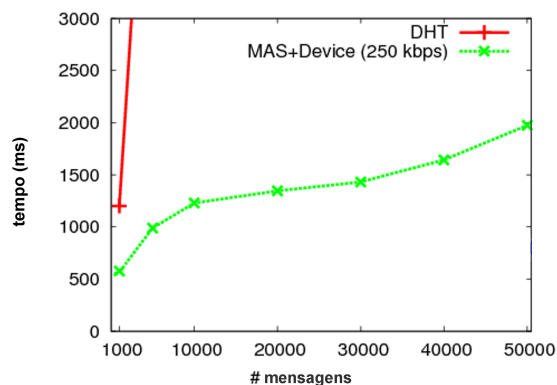


Figura 3. Escalabilidade nas mensagens atendidas.

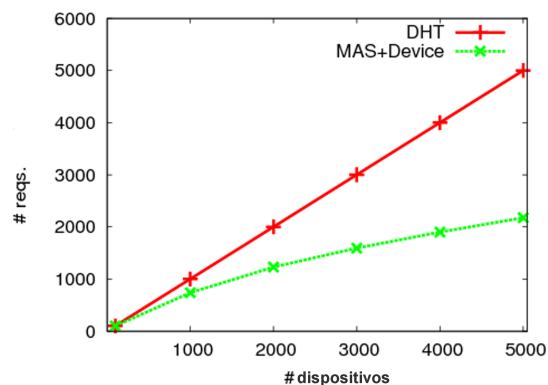


Figura 4. Escalabilidade nas mensagens enviadas.

sua eficiência quando sua largura de banda para respondê-las fica sobrecarregada) com a nossa (que cria novos agentes para distribuir as requisições e, portanto, a carga na largura de banda). A figura mostra que a DHT consegue atender 1.200 mensagens em 1 segundo, enquanto a nossa alternativa, em 2 segundos, consegue atender 50.000. Este resultado foi publicado em [4].

B. Localização de Sensores

Na localização de sensores, a escalabilidade foi analisada pela quantidade de mensagens de atualização enviadas pelo grupo comparada com as enviadas pela fronteira. Para isso, foi simulado o comportamento exibido por um grupo de aves que voam em uma determinada área, conhecido como bando (*flocking* em inglês). As diferentes formas que pode ter um bando foram obtidas da execução de um modelo 2D definido por Craig Reynolds [11]. A Figura 4 mostra que o uso da fronteira diminui a quantidade de mensagens recebidas pelo agente (que está gerenciando o grupo) se comparada com a estratégia usada pelas alternativas atuais, onde todos os dispositivos do grupo enviam mensagens ao agente. Por exemplo, no grupo onde a soma dos seus membros totaliza 5.000 dispositivos, nossa arquitetura supera os outros trabalhos, enviando somente 2.175 mensagens ao invés das 5.000 (correspondendo a uma mensagem por dispositivo), com uma redução de aproximadamente 57%. Por outro lado, a nossa proposta utiliza a fronteira somente depois do grupo ter uma forma. Antes disso, a arquitetura utiliza a mesma estratégia das outras alternativas, onde todos enviam mensagens ao agente. Este resultado foi publicado como capítulo de livro em [5].

VI. CONCLUSÃO

Para aumentar a escalabilidade de sistemas para recuperação e atualização de informações vinculadas, foi apresentada uma arquitetura em camadas que utiliza técnicas inovadoras que estendem tanto o método de agregação de dados quanto a DHT. Para isso, na camada de dispositivos, somente os dispositivos que pertencem à fronteira do grupo são responsáveis por coletar e agregar as informações dos dispositivos que residem na sua área de influência, evitando que todos os membros

do grupo o façam. Na camada de agentes, os agentes são responsáveis por gerenciar as informações desses grupos, por responder requisições de busca por essas informações e por se comunicarem com outros agentes para trocar as interações que possam ocorrer entre seus grupos, como uniões e divisões. Quando o agente fica sobrecarregado pelas responsabilidades assumidas, novos agentes são criados para gerenciar o grupo, distribuindo a carga entre eles e aumentando a escalabilidade do sistema. Dentre os próximos passos, no cenário de VoD, pretende-se estudar como a união de grupos pode melhorar a eficiência no tempo total para baixar um vídeo.

REFERÊNCIAS

- [1] GARTNER. *Hype Cycle for the Internet of Things*. 2015. Disponível em: <<http://www.gartner.com/doc/3098434>>.
- [2] REN, F. et al. Attribute-Aware Data Aggregation Using Potential-Based Dynamic Routing in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, v. 24, n. 5, p. 881–892, 2013.
- [3] STOICA, I. et al. Chord: A scalable peer-to-peer lookup service for internet applications. In: *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. [S.l.]: ACM, 2001. (SIGCOMM '01), p. 149–160.
- [4] ROCHA, V.; BRANDÃO, A. A. F. Towards conscientious peers: Combining agents and peers for efficient and scalable video segment retrieval for VoD services. *EAAI*, v. 45, p. 180 – 191, 2015.
- [5] ROCHA, V.; BRANDÃO, A. A. F. A Scalable Multiagent Architecture for Monitoring Biodiversity Scenarios. In: ADAMATTI, D. (Ed.). *Multi-Agent Based Simulations Applied to Biological and Environmental Systems*. Hershey, PA: IGI Global, 2016. cap. 4.
- [6] GALLUCCIO, L. et al. On the potentials of object group localization in the Internet of Things. In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. [S.l.: s.n.], 2011. p. 1–9.
- [7] D'ORO, S. et al. Exploiting Object Group Localization in the Internet of Things: Performance Analysis. *IEEE Transactions on Vehicular Technology*, v. 64, n. 8, p. 3645–3656, 2015. ISSN 0018-9545.
- [8] TANG, J. et al. An energy efficient hierarchical clustering index tree for facilitating time-correlated region queries in the Internet of Things. *Journal of Network and Computer Applications*, v. 40, p. 1–11, 2014.
- [9] ALI, M.; UZMI, Z. Csn: a network protocol for serving dynamic queries in large-scale wireless sensor networks. In: *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*. [S.l.: s.n.], 2004. p. 165–174.
- [10] ALI, M.; LANGENDOEN, K. A Case for Peer-to-Peer Network Overlays in Sensor Networks. In: *International Workshop on Wireless Sensor Network Architecture*. [S.l.: s.n.], 2007. p. 56–61.
- [11] REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 21, n. 4, p. 25–34, ago. 1987.