MATe: Multiagent Architecture for Taming e-Devices (Extended Abstract)

Vladimir Rocha
Escola Politécnica
University of São Paulo
vmoreira@ime.usp.br

Anarosa Alves Franco Brandão
Escola Politécnica
University of São Paulo
anarosa.brandao@usp.br

ABSTRACT

In recent years, an explosive growth has been observed in the use of wireless devices, mainly due to the decrease in cost, size, and energy consumption. Researches in the Internet of Things have focused on how to continuously monitor these devices in different scenarios, such as vehicle, weather, and biodiversity tracking, considering both scalability and efficiency while searching and updating their information. For this, current alternatives use a combination of a widely recognized method, called data aggregation, and a widely adopted distributed structure, called Distributed Hash Table, which minimize the number of transmissions and save energy. However, scalability is still a key challenge when the group comprises a large number of devices. In this paper, we propose a scalable architecture that distributes the data aggregation responsibility to the devices of group frontier, and creates agents to manage groups and the interaction among them. Experimental results showed the viability of adopting this architecture if compared to the most widely used approaches.

Keywords

Multiagent System, IoT, Data Aggregation, DHT

1. INTRODUCTION

The Internet of Things (IoT) is an emerging paradigm for integrating *things* (real world devices) to create new value-added applications [1]. A wide range of value-added tracking applications (such as in environmental [11], vehicle [9], biodiversity [7], and people [2] scenarios) requires continuously monitoring a large number of devices deployed in some area of interest.

Current alternatives combine a widely recognized method, called data aggregation (DA) [4], and a widely used distributed structure, called Distributed Hash Table (DHT) [10] to monitor these devices. The method consists in choosing one device for collecting and managing information from a group of devices, minimizing the information transmission and saving energy. The DHT is used for efficiently retrieving the chosen device. However, scalability in the chosen device is still an open issue when the group is composed of a large number of devices.

In this paper, we propose MATe, a scalable architecture for discovering and for updating the location for a group of devices. Negotiation and autonomy are capabilities explored by the proposed architecture. In that, a multiagent layer was built above the device layer to deal with scalability while maintaining the efficiency of current alternatives. In such a layer, each agent has a goal to improve the global scalability. For this, we added capabilities to an agent for negotiating mergings with other agents, for perceiving if it is necessary to create new agents, and for monitoring the behavior of devices it is responsible.

2. PROPOSED ARCHITECTURE

The overall architecture, composed of two layers, is shown in Figure 1. In the device layer, a group of devices forms a *swarm*, the devices that bound it are the *border*, and the devices at a certain distance from the border are the *frontier*. Devices in the frontier establish connections[1] among them to decide which ones are responsible for aggregating and for sending the swarm information to the agent that manages it (devices d_1 and $d_2 \in$ border B_1, and d_1, d_2 and $d_3 \in$ swarm of B_1). In the multiagent layer, each agent is responsible for monitoring the border behavior and interacting with other agents to exchange information about the borders they monitor. Borders' behaviors can be of three types: neutral, splitting or merging. We say that a border is splitting when it is disconnected (agent a_4 interacting with agent a_2 and a_3 about B_4 border split). On the other hand, a border is merging when two borders are intersecting (agent a_2 interacting with agent a_3 about B_2 and B_3 borders merge). A border is neutral if it is neither splitting nor merging.

In the device layer, to support scalability, only devices d_i belonging to the frontier are responsible for executing the following processes: (D1) updating the location of the border; (D2) perceiving if the border is splitting; (D3) updating the connections with other devices; (D4) agentifying the device.

In **D1**, each device d_i is responsible for collecting and storing the location of its neighbors, and for sending this information to the agent. Then, this responsibility is rotated among devices in the frontier. In **D2**, each device d_i is responsible for maintaining the border connected. If some disconnection occurs, d_i tries to reconnect the border from

[1]Channel established when they are in a transmission range.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

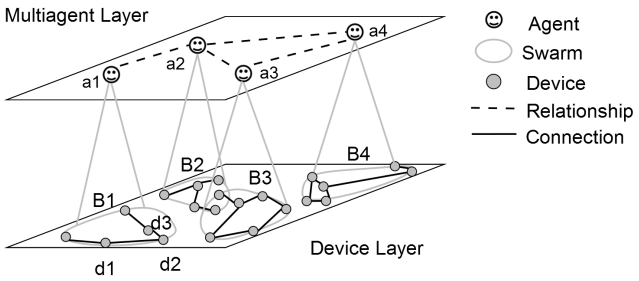


Figure 1: Overall Architecture.

its neighbors and, if it could not do it, d_i will notify the agent that there is a border split. In **D3**, each device d_i must find the devices, located at the swarm, that will be its neighbors. In **D4**, when an agent perceives that scalability is being compromised, it decides that some device it monitors must be agentified. Note that agentifying a device means that it remains running as such in the device layer, but it also acts as an agent in the multiagent layer because it was empowered with agent capabilities such as negotiation and autonomy.

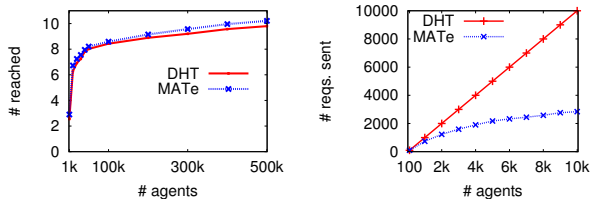
In the multiagent layer, each agent a is responsible for executing the following processes: (A1) receiving the location of the frontier; (A2) perceiving and negotiating about mergings; (A3) perceiving splitting and requesting the device to agentify; (A4) perceiving scalability issues, requesting the device to agentify and splitting.

In **A1**, when the frontier location is received (from D1), agent a updates the area covered by its swarm' frontier. In **A2**, if the result of interaction between agents a_i and a_j indicates some frontier intersection, they negotiate and decide by consensus which one will be responsible for managing the frontier resulting from the merge. Then, the elected agent updates its frontier and the other agent is eliminated from the multiagent layer. In **A3** when agent a receives notifications about the frontier splitting (from D2), it inspects its frontier in order to evaluate which device must be agentified (i.e., converted into agent). When the device is agentified into agent a_i (using D4), agents a and a_i negotiate about which one will be responsible for each of the swarms resulting from the split. In **A4** when agent a perceives that its scalability is compromised by the number of update requests (from D1), it decides about splitting its swarm into two in the same way as A3 does.

3. EXPERIMENTAL RESULTS

We evaluated the efficiency and scalability of the proposed architecture using simulations and measured in terms of number of messages exchanged by the layers to retrieve an information. The layers were implemented in Java, using PeerSim [3] as the simulation framework to evaluate the protocols developed for the Multiagent and Device layers.

The efficiency experiment measures the system efficiency when a device is joining it. It compares one of the most efficient structured solution for distributed systems (DHT) with MATe. Figure 2(a) shows the number of agents needed to find a swarm when a device performs a join request. We observe that, with 300000 agents, our system reaches 9.57 agents, which is approximately the same number of nodes reached by DHT (9.2 nodes). Note that these numbers are the same as those theoretically obtained in several DHT implementations [10, 8], which is $O(log\ n)$. Therefore, MATe performed as efficient as DHT.



(a) Efficiency on finding an agent.

(b) Scalability on using the frontier.

Figure 2: Efficiency and scalability results.

The scalability experiment measures the system scalability while comparing the number of messages exchanged among the swarm and the agent using one of the most scalable structured solution for distributed systems (DHT) with MATe. In the experiment, in order to have the frontier of a swarm, we analyze different swarm shapes, obtained from running the 2D flocking model defined by Craig Reynolds [5]. In Figure 2(b) we observe that, using the frontier decreases the number of messages received by the MATe' agent if compared with current alternatives in which all the devices of the swarm send the messages to a DHT node. For example, in a swarm of 10000 devices, our architecture outperforms other work, just sending 2839 messages instead of 10000 (one message for each device), which represents a reduction of approximately 71%, improving scalability.

4. CONCLUSIONS

We presented MATe, a layered architecture that deals with scalability issues raised in a specific class of distributed systems whose better solution adopts a combination of distributed hash table structure and data aggregation method. For such, MATe gives agent capabilities to DHT nodes to allow negotiation and autonomy for addressing the aforementioned problem. In addition, MATe's device layer improves the data aggregation scalability by limiting the responsibility for collecting, aggregating and sending neighbors information to the swarm frontier.

According to the experimental results, using the frontier of a swarm decreases the number of messages sent to the agent, increasing its scalability. Also, adding agent responsibilities to a DHT node maintains the efficiency of the structure while increasing its scalability. We implemented the architecture in a Video-on-Demand domain [6] and in a monitoring flocking behavior environment [7]. Currently, we are implementing the architecture in an intensive group location displacement, such as vehicle tracking.

REFERENCES

[1] E. Borgia. The Internet of Things vision: Key features, applications and open issues. *Computer Communications*, 54:1–31, 2014.

[2] K. Li, C. Yuen, and S. Kanhere. SenseFlow: An Experimental Study of People Tracking. In *Proceedings of the 6th ACM Workshop on Real World*

Wireless Sensor Networks, RealWSN '15, pages 31–34, New York, NY, USA, 2015. ACM.

[3] A. Montresor and M. Jelasity. PeerSim: A scalable P2P simulator. In *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P'09)*, pages 99–100, Seattle, WA, Sept. 2009.

[4] F. Ren et al. Attribute-Aware Data Aggregation Using Potential-Based Dynamic Routing in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(5):881–892, May 2013.

[5] C. W. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, Aug. 1987.

[6] V. Rocha and A. A. F. Brandão. Towards conscientious peers: Combining agents and peers for efficient and scalable video segment retrieval for VoD services. *EAAI*, 45:180 – 191, 2015.

[7] V. Rocha and A. A. F. Brandão. A Scalable Multiagent Architecture for Monitoring Biodiversity Scenarios. In D. Adamatti, editor, *Multi-Agent Based Simulations Applied to Biological and Environmental Systems*, chapter 4. IGI Global, Hershey, PA, 2016.

[8] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Middleware '01, pages 329–350, London, UK, UK, 2001. Springer-Verlag.

[9] H. Soleimani and A. Boukerche. SLA: Speed and Location Aware LTE Scheduler for Vehicular Safety Applications. In *Proceedings of the 13th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '15, pages 13–19, New York, NY, USA, 2015. ACM.

[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. 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*, SIGCOMM '01, pages 149–160, New York, NY, USA, 2001. ACM.

[11] A. Zenonos, S. Stein, and N. R. Jennings. Coordinating Measurements for Air Pollution Monitoring in Participatory Sensing Settings. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 493–501, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.