Message-Oriented Middleware for Edge Computing Applications

Thomas Rausch
Distributed Systems Group
TU Wien, Austria
rausch@dsg.tuwien.ac.at

ABSTRACT

Edge computing is an emerging paradigm which aims to leverage the ever increasing amount of computational resources at the edge of the network to satisfy the stringent quality of service (QoS) requirements of many modern Internet of Things (IoT) scenarios. This PhD thesis explores challenges and solutions of message-oriented middleware (MOM) for edge computing applications. In particular, we focus on QoS optimization and message delivery guarantees under the constraints of geographic dispersion, client mobility, dynamic resource availability, and privacy policies.

KEYWORDS

edge computing; internet of things; message-oriented middleware

1 INTRODUCTION

Cloud-based message-oriented middleware (MOM) is a widely-used technology for facilitating device communication in large-scale IoT applications [1]. However, for IoT scenarios where low end-to-end latencies between data producers and consumers is a critical requirement [6], latencies incurring from routing messages from the edge of the network to the cloud may be impractical [7].

Edge computing is considered a key enabler for such scenarios. Edge computing aims to provide highly responsive mobile computing and IoT applications by leveraging computational resources placed at the edge of the network, close to where data are generated [8]. These resources could also be used by MOM to enhance QoS for device communication in IoT scenarios, as illustrated in Figure 1.

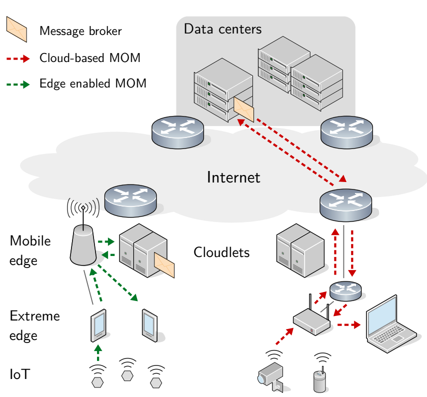

Figure 1: Message delivery in edge computing architecture

When designing MOM for edge computing applications, we are faced with numerous challenges. In this work, we aim to examine the following challenges in particular:

Consolidating geographically dispersed communication: Communication in edge computing applications is not limited to devices in close proximity. In publish–subscribe communication scenarios, subscribers to channels may be at any geographic location. In IoT data analytics applications, data are processed and filtered at the edge, and subsequently handed off to enterprise clouds for storage or further analytics [5, 6]. To support such applications, edge computing MOM must be able to provide low-latency communication for devices in close proximity, and, at the same time, facilitate messages distribution to geographically dispersed locations.

Mobility and dynamic resource availability: In many edge computing applications, clients as well as resources are mobile. A use-case from the military domain, where soldiers are equipped with mobile devices that communicate with so-called *tactical cloudlets* highlights this [5]. These cloudlets are edge resources hosted on vehicles, drones, or other mobile platforms in close proximity to field personnel. Client mobility and volatile resources in these tactical environments make it extremely challenging to provide message delivery guarantees and resilient coordination mechanisms.

Data privacy: The centralization of IoT systems has caused growing concern about data privacy [7]. In the edge computing vision, user-defined privacy policies specify which data leave the edge, or what type of pre-processing has to be performed before handing them off to the cloud. In MOM, this also concerns message routing. When passing messages through a network of brokers, privacy policies may constrain message delivery by, e.g., excluding certain geographic regions.

2 RELATED WORK

Although publish–subscribe MOM is a well researched topic [3], and individual problems, such as message routing through complex overlay networks, have received a lot of attention in the past decades, few efforts have been made to engineer holistic solutions for MOM for edge computing. In particular, these approaches often lack concrete solutions for QoS monitoring and elastic resource provisioning, two key requirement of edge computing MOM.

Some open pub/sub brokers provide basic mechanisms for enabling edge computing, but rely on static configurations to do so. The popular Message Queue Telemetry Transport (MQTT) brokers Mosquitto or HiveMQ provide the concept of bridging, where brokers can be deployed at the edge and pre-configured to forward specific topics to centralized brokers. Clearly these approaches are limited in their operational capability and cannot deal with dynamic resource availability and client mobility.

Scalability and load-balancing techniques for pub/sub middleware focus mainly on cloud-based systems. Commercial systems, such as Amazon IoT, leverage the immense computational capacities of the companies data centers to deliver transparently scaling, centralized message brokers. In research, prototypes such as Dynamoth [4] focus on elastically provisioning a network of replicated brokers in the cloud to maintain QoS during varying message load. As such, these cloud-based systems disregard client proximity or latencies incurring from link usage.

The closest related work to ours we are aware of is by An et al. [2] who present PubSubCoord, a cloud-based coordination system for a distributed broker network, focusing on wide-area communication.

3 APPROACH

We propose a pub/sub MOM for edge computing applications based on the MQTT protocol which has proliferated as a standard pub/sub platform for IoT scenarios [1]. Our system orchestrates a network of brokers and dynamically reconfigures the network at runtime to optimize QoS. To that end, the system implements the following core mechanisms, where we synthesize existing and novel approaches.

QoS monitoring & proximity detection: We implement a lightweight monitoring protocol to continuously monitor network QoS between nodes. The system maintains the current state of the network as a graph data structure that allows us to reason about the network at runtime. We determine proximity of nodes based on different distance metrics such as latency and routing hops.

Dynamic bridging: We extend the static bridging approach of existing MQTT systems to enable efficient message dissemination to geographically dispersed locations. Clients may connect to any broker in the network and subscribe to arbitrary topics. An overlay management component maintains distributed subscription tables that brokers access to known which messages to forward.

Network reconfiguration: A network reconfiguration engine reconfigures client–broker connections at runtime based on data gathered by our monitoring protocol. Its purpose is to optimize QoS while maintaining balanced load between brokers that provide similar QoS. We will experiment with different metrics (latency, packet loss, bandwidth, etc.) and devise load-balancing strategies that address the challenges of edge computing infrastructure.

Elastic resource provisioning: Using proximity and QoS data in combination with a resource discovery mechanism, the system can identify regions in the network where additional brokers may alleviate network usage and optimize QoS. We will examine how existing elasticity mechanisms for MOM, such as the one proposed in [4], are feasible and how they can be extended to suit our needs.

Privacy policy enforcement: We provide facilities (e.g., in the form of a DSL) to define custom privacy policies, which the system enforces by augmenting message delivery routes and constraining resource provisioning efforts. We examine how reconfiguration and provisioning strategies have to be extended to enable this.

4 EVALUATION

To verify the efficacy of the approach, we will perform experiments in both simulated environments as well as real-world testbeds. The evaluation will focus on measuring QoS the system can guarantee under geographic dispersion, client mobility and different privacy policies. We will also evaluate the cost of our proposed mechanisms in terms of network communication overhead and resource consumption. To that end, we have begun the development of a reusable tool suite for evaluating pub/sub systems[1].

As testbed, we have at our disposal the IoT infrastructure created in the CPS/IoT Ecosystem project[2]. In the context of this project, there are also several application scenarios (including, smart farming, autonomous vehicle networks, and smart production systems) that will serve as evaluation use-cases for our approach.

5 PRELIMINARY RESULTS

To date, we have begun the implementation of a proof-of-concept prototype[3] that coordinates a network of MQTT gateways and brokers, and reconfigures client–broker connections at runtime. A QoS monitoring protocol allows proximity detection based on network latency. We deployed the prototype in an Amazon EC2 testbed spanning three different regions to evaluate the system under geographic dispersion. Experiments show that we can provide near real-time communication between clients in close proximity, and allow message delivery to geographically dispersed locations, even in the face of client and resource mobility. Next steps include implementing delivery guarantees and elastic broker deployment.

6 CONCLUSION

Without MOM that is expressly designed for edge computing applications, many requirements of modern IoT scenarios will remain unsatisfied. The aim of this thesis is to examine how the challenges of edge computing affect the design of MOM. The infrastructure and scenarios of the CPS/IoT Ecosystem project provide a real-world testbed for a comprehensive evaluation of our approach.

ACKNOWLEDGMENTS

This thesis is under the supervision of Schahram Dustdar, and is partly supported by the Austrian Federal Ministry of Science within the CPS/IoT Ecosystem project and by TU Wien research funds.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials* 17, 4 (2015), 2347–2376.

[2] K. An, S. Khare, A. Gokhale, and A. Hakiri. 2017. An Autonomous and Dynamic Coordination and Discovery Service for Wide-Area Peer-to-peer Publish/Subscribe: Experience Paper. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems (DEBS'17)*.

[3] PT. Eugster, PA. Felber, R. Guerraoui, and AM. Kermarrec. 2003. The Many Faces of Publish/Subscribe. *ACM Comput. Surv.* 35, 2 (jun 2003), 114–131.

[4] J. Gascon-Samson, FP. Garcia, B. Kemme, and J. Kienzle. 2015. Dynamoth: A Scalable Pub/Sub Middleware for Latency-Constrained Applications in the Cloud. In *2015 IEEE 35th International Conference on Distributed Computing Systems.*

[5] G. Lewis, S. Echeverria, S. Simanta, B. Bradshaw, and J. Root. 2014. Tactical Cloudlets: Moving Cloud Computing to the Edge. In *2014 IEEE Military Communications Conference.* IEEE, 1440–1446.

[6] S. Nastic, T. Rausch, O. Scekic, S. Dustdar, M. Gusev, B. Koteska, M. Kostoska, B. Jakimovski, S. Ristov, and R. Prodan. 2017. A Serverless Real-Time Data Analytics Platform for Edge Computing. *IEEE Internet Computing* 21, 4 (2017), 64–71.

[7] M. Satyanarayanan. 2017. The Emergence of Edge Computing. *Computer* 50, June (2017), 30–39.

[8] W. Shi and S. Dustdar. 2016. The Promise of Edge Computing. *Computer* 49, 5 (may 2016), 78–81.

[1] https://hyde.infosys.tuwien.ac.at/emma/pubsub-benchmark
[2] http://cpsiot.at
[3] https://hyde.infosys.tuwien.ac.at/emma/emma