Chapter 13

Publish/Subscribe-Based Service Bus for IoT Services

Yang Zhang
Beijing University of Posts and Telecommunications, China

ABSTRACT

In IoT (Internet of Things) scenarios, lots of things and services are connected and coordinated each other. In our work, we first propose a service-oriented publish/subscribe middleware as a construction base of distributed, ultra-scale, and elastic service bus for IoT applications. The IoT services in our solution are then aware of underpinning service communication fabric, where they are event-driven, their interfaces are defined by underlying event topics, their behaviors are specified by event relations, and they can cooperate with the service communication fabric to complete distributed service coordination.

INTRODUCTION

In some applications, the communication foundation should cooperate with the computation systems. The cloud computing in data centers is based on this idea to realize elastic and scalable computation with good utility. Especially in IoT (Internet of Things) scenarios, there are lots of things and real-time services to connect and coordinate each other such that the cooperation becomes very important for supporting IoT services’ high scalability and interconnection. In the work of Bakken et al. (2011), a special publish/subscribe middleware was proposed for smart grids with adjusting communication fabric for upper IoT services. Software-defined networks (SDN) (OME Committee, 2012) was such efforts with allowing applications to control all network operations. The two examples did not consider redesigning upper applications. In our work, the communication fabric for IoT services acts as a service bus. For a service bus and IoT service systems over it, the cooperation involves three problems:

1. Whether the architecture of service bus supports the upper applications to easily utilize its fine-grained functionalities and flexibly adjust it for their changes.
2. Whether the interaction functionalities can be separated from the upper application and be assumed by the communication fabric to realize the decoupling-based scalability for application systems.

DOI: 10.4018/978-1-5225-2157-0.ch013
3. Whether the upper applications can be aware of the underpinning communication fabric to adapt themselves such as adjusting their behavior and moving.

Compared with GridStat and SDN rethinking communication foundation, the work of (Li et al. 2010; Hensa, et al. 2014) re-thought upper applications for accommodating to the underpinning. They partitioned a business process into multiple pieces, which were deployed and executed according to the underpinning’s states. But they did not consider the two-layer cooperation.

We think that the two-layer redesigning is an appropriate way to realize the services’ high concurrency and scalability. In our work, we firstly design the underlying communication fabric which works on the service (SOAP) protocol. Then, distributed service containers are designed. These two components construct our distributed publish/subscribe-based service bus for IoT services, where service routing such as sending service invocations to targets is supported, its routing and management capabilities are encapsulated as application interfaces for bus clients to access to its functionalities. Our bus is called UMS (Unified Message Space).

The remainder of the paper is structured as follows. Section 2 describes the whole architecture of our service bus. Section 3 describes the service-oriented publish/subscribe middleware. In Section 4, the service’s behavior coordination functionalities are integrated into the middleware with supporting service-oriented programming. In Section 5, the service’s authorization control is also combined with the communication fabric to act as an example of integrating service properties. In Section 6, we present applications and experiments for our solutions. Section 7 describes the related work. Finally, conclusions are drawn in Section 8.

Unified Message Space

We propose a service-oriented publish/subscribe middleware, and then we try to create a publish/subscribe-based service bus to converge distributed IoT services deployed near their managed physical devices, called Unified Message Space (UMS). The task of our publish/subscribe middleware is to unify the management of transmission of time-synchronous sensed IoT data and IoT services’ operations; support the interoperation among all IoT participants; and provide a coordinate mechanism for IoT resources and IoT services. Only the publish/subscribe functionality cannot complete the task. The publish/subscribe middleware and the service runtime environment together assume the event-driven service bus, i.e., UMS, where event names and event relations can be used to define IoT service interfaces for converging service capabilities and IoT data; coordination logic can be cut off from the services as an independent modeling unit such that it can be integrated into the communication fabric to realize cross-layer coordination in the middleware; and collaborating business processes can be transparently managed without being aware of their deployment sites.

Our publish/subscribe-based service bus, UMS, is often used and deployed in an IoT application, illustrated in Figure 1. The Figure 1 depicts such IoT (Internet of Things) scenarios. The access agent gets raw data from sensors and uses resource models to translate it into events. These events enter the unified message space, and can be subscribed by all services. IoT services also use the unified message space to publish events which could be sent to the sensors as instructions. The coordination interfaces are executed in distributed service container and use pub/sub interfaces as basic activities which are realized through the local layer. Figure 2 illustrates the publish/subscribe middleware, where local layer provides services to connect local clients, and all local layers at different sites are connected each other.