EE542 - Reading Assignment – 10

Toward a Distributed Data Flow Platform for the Web of Things

Presenter: Boyang Xiao

USC id: 3326-7302-74

Email: boyangxi@usc.edu

Index

- Introductions
- IoT data flow architecture
- Toward distributed data flow
- Q&A

Introductions

- Background
  - Today practitioners are creating new tools and platforms to ease the development of the class of IoT applications. To provide more flexibility while maintaining ease of use, several platforms [4, 5, 7, 11, 14] provide a data flow programming paradigm where computer programs are modeled as directed graphs connecting networks of ‘black box’ nodes that exchange data along connected arcs. Two web-based data flow systems in particular; the WoTKit Processor [5], and Node-RED [11], begin to address interactive IoT scenarios like these.
- This paper will:
  - provide an overview of data flow architectures and describe the WoTKit and Node-RED systems
  - outline authors’ proposed distributed architecture, and describe an early prototype system the authors have developed based on Node-RED
  - provide context for this work by describing existing interactive web-based platforms that can be used in IoT applications

IoT data flow architecture

- The two systems described here provide such an environment for building real time IoT applications, where users can arguably more easily move between program design and implementation and reduce development time:
- WoTKit Processor:
  - WoTKit Processor is a multi-user service bundled with the WoTKit platform that allows users to process sensor data and react to real time updates from sensors and other external systems.
  - The WoTKit Processor includes input and output modules to send and receive data to external systems such as WoTKit-hosted sensors, twitter feeds and email.
  - The Processor's pipes are expressed using JSON documents. The visual editor on the browser generates these JSON representations, which are sent to the server for storage and execution.

IoT data flow architecture

- Node-RED
  - Node-RED is a web-based tool for connecting hardware devices and APIs, who provides a browser-based flow. It is implemented in JavaScript using the Node.js framework, taking advantage of Node's built in event model, and native support for JavaScript on both the client editor and the server.
  - Nodes in a flow inherit from the Node base class. A Node is also subclass of an EventEmitter in the Node.js event API that implements the observer design pattern to maintain subscriber lists defined by wires, and emits events to downstream nodes.
  - Node-RED's flows are similar to the Processor's Pipes in that they are expressed using JSON. One difference is that "wires" are not separate objects, but are arrays associated with each node connecting it to a downstream node.

Toward distributed data flow

- Because operations in a data flow can run as soon as data is available, and share no state with each other, operations are inherently parallel and can run independently.
- The authors propose a cloud-based platform called ‘WoT Flow’ that takes advantage of this property of data flow programs and the open source Node-RED system to provide an execution engine suitable for both multi-user cloud environments and individual devices.
- On deployment and execution of a flow, a portion of the overall flow, a subflow, will be executed on the cloud service, while other subflows will be distributed and executed on devices.

Toward distributed data flow

- Data Flow Extensions
  - To support distributed flows, the data flow program model used by Node-RED or the Processor needs to be extended in a number of ways:
    1. First, flows should have an owner. This allows the system to associate access and control of a flow to a specific user and their devices, a requirement for any multi-user system
    2. Second, it must be possible to mark nodes as ‘device’ nodes, ‘server’ nodes, or ‘mobile’ nodes. Device nodes are those that rely on local device connectivity, or the specific capabilities of a device.
    3. Similarly, we must extend a simple flow model to include different types of arcs or wires. In a distributed flow, the wires between nodes are not all local connections in the same execution engine, but may involve the transfer of data between servers and devices over a local or wide area network.
- Work to Date
  - The WoT Flow system is at the early stages of development. We are currently finalizing the distributed flow model by modifying Node-RED to support distributed flows.
  - To date we have modified certain nodes in Node-RED by adding a device id property. This property specifies the device where the node should execute.

Q&A

- **Question:** What do WoTKit Processor and Node-Red have in common and what are their differences?
- **Answer:**
  - Similarities:
    - Both of them use a drag and drop visual editor to generate data flows using JSON. The generated flows (pipes) consist of nodes (modules) connected via wires. On execution, modules are instantiated in memory and execute code as they receive data.
  - Differences:
    - Node-RED hosts a single set of connected nodes, and can only manage one flow on the system. The WoTKit Processor never access local sensors or OS services directly since the execution engine is server-based.
    - The WoTKit Processor pipe data flow model has a separate section for wires allowing programs to iterate through nodes and wires separately, while wires in the Node-RED model are associated directly with nodes.

Thanks for watching!

Presenter: Boyang Xiao

USC id: 3326-7302-74

Email: boyangxi@usc.edu