Investigating IoT Cloud Systems with MobIoTSim

Attila Kertesz, Ph.D. and Tamas Pflanzner

University of Szeged, Hungary

keratt@inf.u-szeged.hu

2017.11.10.
IoT Cloud systems

Cloud
- Manage recourses
- Scalability
- Handle big data
- Access from everywhere

IoT Cloud features
- Connect devices
- Store the data
- Process
- Visualize
Research collaborations in ACROSS

- **IoT extension of DISSECT-CF:**
  - Liverpool John Moores University (Gabor Kecskemeti)
  - 2 STSMs
  - 3 conference papers, 1 book chapter
  - 1 journal paper

- **Development of MobIoTSim:**
  - University of Antwerp (Steven Latre)
  - 1 STSM
  - 1 conference paper

- **Legal aspects of IoT Cloud systems:**
  - University of Zurich (Burkhard Stiller)
  - 1 conference paper

- **Involvement in Cloud Federation Task Force**
  - 1 white paper, 1 book chapter
IoT extension of DISSECT-CF

DISSECT-CF: DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations

https://github.com/kecskemeti/dissect-cf
MobIoTSSim: hybrid simulations

- Android application
- Connects to the cloud
  - MQTT protocol with Eclipse PASO client
- Simulates many IoT devices
  - With authentication (ID and token)
  - Custom data generation (JSON format)
  - Custom frequency
  - Interactive start and stop of devices
MobIoTSim use cases

- **PaaS developer**
  - Create IoT applications
  - Learn the basics
  - No need to buy real devices

- **Test an IoT cloud application**
  - Generate different test data
  - Replay the test cases
MobIoTSim screenshots

https://github.com/sed-szeged/MobIoTSim
MobIoTSim options

- Cloud settings
  - Common part for all simulated devices
  - How to connect to the cloud
  - Cloud specific fields (generates a URL)

- Device settings
  - Specific for one device
  - Type
  - Authentication (ID, token)
  - Frequency
  - Random data generation, or trace loading

- Devices
  - Device list (Add / Edit / Delete)
  - Start / Stop
  - Warnings (Reactions from the cloud gateway)
  - Grouping
MobIoTSim and Bluemix

- Cloud application runs on Bluemix
  - Uses the native Bluemix MQTT broker (IoT service)
- Devices can be registered to the Bluemix database (ID, token)
- Visualisation
  - Receives data from each simulated device
  - Visualize sensor data in real time
MobIoTSim with Bluemix gateway

IBM Internet of Things Foundation

Use a different API key and Auth Token
Private gateway in Bluemix

- We developed an own, customized gateway for the IBM Bluemix platform

New features:
- Inclusion to Docker container
- Grouped device management, faster data management with paging
- CPU Stress loading for more realistic operation
- Advanced performance monitoring
GUI of the custom gateway
Resource usage in Watson

Device types:
- Total devices: 5
- MobIoTSimType: 3
- MobIoTSimWeather: 2

Data transferred:
- Data transferred today: 1406.7 MB
- This month: 5478.9 MB
- Previous month: 0.0 MB
Grafana resource usage charts

- Detailed usage info for Docker apps
Initial evaluation

- Thermostat template (temperature value)
- Device groups for: 10, 100, 250 devices
- 1 message per second
Detailed evaluation with Thermostat devices

- Device groups for: 100 and 450 devices
- 1 or 2 messages per second
Extension for stressing

- The reason for little load is that we performed no real data processing tasks with the gateway, though in real world we may do that

- Therefore we introduced stressing processes with parameterizable Fibonacci number generation

- We used the setting to count the 20th Fibonacci number upon each received message multiplied by a score representing the size of the message
Evaluation with CPU stressing

- Comparison of different data generation frequencies (1 or 0.5 seconds) with stressing
- Continuous measurements for 100, 250 and 450 devices
Detailed evaluation with Thermostat devices

- Device groups for: 10, 100, 250 and 450 devices
- 1 message per second
- Resource usage shares
Template for weather station devices

- We used this Weather template to create device groups of 100 and 450 simulated devices.
- In MobIoTSim we have a template to set up weather station parameters following the OpenWeatherMap format.
- We can also load previously saved OpenWeatherMap traces of certain cities.
- For this experiment we randomly picked weather data of cities (one city for one simulated device) from earlier traces.
Detailed evaluation with Weather station devices

- From the results we can see that as the number of devices grow, the resource utilization also gets higher.

<table>
<thead>
<tr>
<th>No. of devices</th>
<th>10</th>
<th>100</th>
<th>250</th>
<th>450</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU util. (%)</td>
<td>1.59</td>
<td>12.27</td>
<td>29.53</td>
<td>52.29</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>110.07</td>
<td>110.22</td>
<td>110.35</td>
<td>110.05</td>
</tr>
<tr>
<td>Network (B/s)</td>
<td>890.66</td>
<td>881.34</td>
<td>855.16</td>
<td>853.60</td>
</tr>
<tr>
<td>Message size (KB)</td>
<td>2468</td>
<td>24695</td>
<td>61666</td>
<td>11110</td>
</tr>
<tr>
<td>No. of messages</td>
<td>6000</td>
<td>60046</td>
<td>149940</td>
<td>270165</td>
</tr>
</tbody>
</table>
Detailed evaluation with Weather station devices

- On the left we can see the difference of the applied two templates in sizes.
- The chart on the right highlights CPU utilization comparison of the two device types.
MobIoTSim app evaluation

The start of simulating 200 devices in MobIoTSim
Future work

- Create our own IoT service (MQTT broker + device management)
- The whole system (IoT service + visualization) could be deployed to different clouds
- Just started researching MQTT brokers:

<table>
<thead>
<tr>
<th></th>
<th>VerneMQ</th>
<th>Moquette</th>
<th>RabbitMQ</th>
<th>ActiveMQ</th>
<th>JoramMQ</th>
<th>Mosquitto</th>
</tr>
</thead>
<tbody>
<tr>
<td>price</td>
<td>free</td>
<td>free</td>
<td>free</td>
<td>free</td>
<td>free</td>
<td>free</td>
</tr>
<tr>
<td>opensource</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>license</td>
<td>appache 2.0</td>
<td>appache 2.0</td>
<td>mozilla public</td>
<td>appache 2.0</td>
<td>lGPL</td>
<td>appache 2.0</td>
</tr>
<tr>
<td>cluster</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Linux</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>MacOS</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>na</td>
<td>na</td>
</tr>
<tr>
<td>Windows</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Language</td>
<td>Erlang</td>
<td>Java</td>
<td>Erlang</td>
<td>Java</td>
<td>Java</td>
<td>C</td>
</tr>
<tr>
<td>Embeddable</td>
<td>Erlang</td>
<td>Maven</td>
<td></td>
<td></td>
<td></td>
<td>yes (Maven)</td>
</tr>
<tr>
<td>Authentication</td>
<td>XML, JAAS (LDAP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><a href="https://github.com">https://github.com</a></td>
</tr>
<tr>
<td>Comments</td>
<td>AMQP (MQTT plugin)</td>
<td>scalability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Thank You for Your Attention!

Questions?

keratt@inf.u-szeged.hu

The research leading to these results has received funding from the European COST programme under Action identifier IC1304 (ACROSS), and was supported by the UNKP-17-4 New National Excellence Program of the Ministry of Human Capacities of Hungary, and by the Hungarian Government and the European Regional Development Fund under the grant number GINOP-2.3.2-15-2016-00037 ("Internet of Living Things").