QoS-oriented service selection in Things-as-a-Service architectures

Enzo Mingozi
University of Pisa, IT

ENSIMAG – Grenoble, November 17, 2015
Outline

• IoT horizontal platforms: centralized vs. distributed approach
  – the EU FP7-BETaaS solution: Things-as-a-Service on top of a "local cloud"

• Quality of Service support for M2M applications
  – the BETaaS QoS framework

• Optimized Thing Service selection
  – Exploit multiple TS equivalence to maximize system lifetime: the Agent Bottleneck Generalized Assignment Problem

• Ongoing & future work
Outline

• IoT horizontal platforms: centralized vs. distributed approach
  – the EU FP7-BETaaS solution: Things-as-a-Service on top of a "local cloud"

• Quality of Service support for M2M applications
  – the BETaaS QoS framework

• Optimized Thing Service selection
  – Exploit multiple TS equivalence to maximize system lifetime: the Agent Bottleneck Generalized Assignment Problem

• Ongoing & future work
Aknowledgement

- EU-FP7 BETaaS (2012-15)
  Building the Environment for the Things as a Service

http://www.betaas.eu
The BETaaS platform

• A reference framework enabling interoperable (horizontal) M2M application deployment
  – A distributed runtime environment based on a “local cloud” of gateways for the services to be deployed so as to fulfill high-level M2M applications
  – Content-centric Things-as-a-Service layered model
  – Built-in support for non-functional requirements (**Quality of Service**, big data management, dependability, security)
  – Seamless integration of existing IoT/M2M systems

• Implementation based on distributed OSGi
  – [https://github.com/BETaaS/](https://github.com/BETaaS/): setup, development tools, tutorials
IoT: still a vision, or real already?
Vertical platforms

- Designed to serve one **single purpose**
- **Inefficient**: each device is dedicated to a single application
- Operate in **isolation**: no (or very limited) cooperation
Horizontal approach

Business Application

Existing Infrastructure

dedicated devices

Business Application

Existing Infrastructure

dedicated devices

Business Application

Existing Infrastructure

dedicated devices

Business Application

Existing Infrastructure

dedicated devices

Service infrastructure

Converged Network Infrastructure

devices

devices

devices

Horizontal approach

Cloud-based centralized platforms

Business Application

Business Application

Business Application

Cloud

IoT/M2M system

IoT/M2M system

IoT/M2M system

Cloud-based centralized platforms

Open Source
OpenIoT

Proprietary
Xively

Is the cloud always appropriate?

Time/space locality not exploited
Getting data into and out of the cloud
(Low) latency requirements
Security
**BETaaS approach**

- **Move/distribute the intelligence to the edge!!!**
  - *BETaaS gateways*: network devices, set-top boxes, RSUs, ...
  - Gateways cooperate to form a runtime (distributed) platform
Use case: Smart Home
Use case: Smart City

BETaaS approach

• Move/distribute the intelligence to the edge!!!

Data storage and processing close (in space and time) to where it is generated

Reduced latency

Resource pooling/optimization

Business Application

Business Application

Business Application

GW

GW

GW

IoT/M2M system

IoT/M2M system

IoT/M2M system
“Local cloud” of gateways

• The set of computational resources hosting the BETaaS runtime environment
Existing IoT/M2M systems

- Heterogeneous physical devices and protocols
- Several data formats and structures
- No common semantic for resource description
Enable integration

Standardize access through a common interface and **data representation**

Provide a basic set of functionalities by the plugged-in IoT/M2M system
Realize integration: TaaS model

Seamless service-oriented access to things irrespectively of the location

Common semantics to enable context-aware lookup

Support for non-functional requirements (e.g., QoS)
Thing service equivalence

- Different things may provide overlapping information or equivalent functionalities
  - Redundant sensors
    - Industrial automation scenario, e.g., for dependability
    - Large scale deployments (weather info sensors, LoRA, ...)
  - Presence information at home: infrared sensors, smart camera, smart thermostat, temperature+humidity, Google, ...
  - Road vehicle detection: infrared sensors, smart camera, ...
- Equivalence is inferred by semantic reasoning based on contextual information by the Context Manager inside the TaaS layer of BETaaS
  - In the current release, it depends on the service type and location
Manage M2M services built on top of TaaS
Outline

- IoT horizontal platforms: centralized vs. distributed approach
  - the EU FP7-BETaaS solution: Things-as-a-Service on top of a "local cloud"

- Quality of Service support for M2M applications
  - the BETaaS QoS framework

- Optimized Thing Service selection
  - Exploit multiple TS equivalence to maximize system lifetime: the Agent Bottleneck Generalized Assignment Problem

- Ongoing & future work
BETaaS functional view

- Built-in support for extended features
  - Context-aware lookup
  - QoS, Security, Big Data, Virtualization
Quality of Service support

- M2M application scenarios may have very different and unique requirements
- Provide services with associated guarantees on performance
  - Allow applications to negotiate a Service Level Agreement (SLA)
    - QoS model
  - Enforce QoS through the efficient management of resources
    - Resource reservation and optimized allocation
  - Monitor SLA fulfillment

QoS model

• Classes of service
  – **Real-Time**: Applications with hard response time requirements (deterministic)
    • E.g., Surveillance system, industrial control, healthcare
  – **Assured Services**: Applications with soft response time requirements (e.g., probabilistic)
    • E.g., Road traffic alerts, Vehicle tracking
  – **Best-Effort**: Applications with no time requirements
    • E.g., Meter data collection

• SLA templates defined accordingly
QoS negotiation

- **SLA negotiation**
- **Admission control** based on QoS requirements
- **QoS-based resource reservation**
- **Authorize** Thing Service invocation

![Diagram showing the relationship between SLA negotiation, QoS negotiation, and TaaS](image-url)
WS-Agreement
Service invocation

- Select at runtime which thing to allocate the service
QoS manager

Outline

• IoT horizontal platforms: centralized vs. distributed approach
  – the EU FP7-BETaaS solution: Things-as-a-Service on top of a "local cloud"

• Quality of Service support for M2M applications
  – the BETaaS QoS framework

• Optimized Thing Service selection
  – Exploit multiple TS equivalence to maximize system lifetime: the Agent Bottleneck Generalized Assignment Problem

• Ongoing & future work
Problem formulation

Service requests

1

Things

1

$p_j$ time period

$p_j = t_{ij}$

$m_{ij} = 1$

Service requests

1

Things

1

$p_j = t_{ij}$

$m_{ij} = 1$

Service requests

1

Things

1

$p_j = t_{ij}$

$m_{ij} = 1$
Problem formulation

We use a common reference time period $h$ to compare energy costs:

$$h = lcm(p_j)$$

$$u_{ij} \triangleq \frac{t_{ij}}{p_j}$$

$$f_{ij} \triangleq \left( \frac{h}{p_j} \cdot c_{ij} \right) / b_i$$

fraction of available energy needed by service $j$ on thing $i$ to complete all executions over the common reference period $h$
Problem formulation

Problem variables

\[ x_{ij} = 1 \text{ if request } j \text{ is allocated to thing } i, \; 0 \text{ otherwise} \]

\[ \sum_j u_{ij} x_{ij} \]

allocated utilization

\[ \sum_j f_{ij} x_{ij} \]

total energy cost of the allocation

energy

utilization

Thing \( i \)
Problem formulation

Constraints? **Schedulability**

\[ \sum_{j} u_{ij}x_{ij} < v_i \]

\[ v_i = s_i \left( \frac{1}{2^{s_i}} - 1 \right) \]

\[ s_i = \sum_{j} x_{ij} \]
Problem formulation

Objective? Minimize the maximum total energy cost per thing

\[
\min \left( \max_i \sum_j f_{ij} x_{ij} \right)
\]

Bottleneck Generalized Assignment Pr.

Minimax objective

\[
\max_i f_{ij} x_{ij}
\]

Task BGAP [Martello, Toth 1995]

\[
\max_i \sum_j f_{ij} x_{ij}
\]

Agent BGAP


Solution

• Solving for minimax

1. heuristically search for a feasible solution of a value lowest than a threshold $\theta$

2. heuristically search for the lowest value $\theta$ for which a feasible solution is found
Solution

- Solving for minimax

1. heuristically search for a feasible solution of a value lowest than a threshold $\theta$

2. heuristically search for the lowest value $\theta$ for which a feasible solution is found
Find a **good** feasible solution (inspired by TBGAP)

Set a **fixed** priority $p_{ij}$ measuring the **desirability** of allocating service $j$ to thing $i$

Many choices available

$$p_{ij} = u_{ij} \quad \text{Largest job first}$$
$$p_{ij} = f_{ij} \quad \text{Highest (energy) cost first}$$
$$p_{ij} = -f_{ij} \quad \text{Lowest (energy) cost first}$$
Solution

Find the next request to allocate $j^*$

Calculate, for each $j$

$$F_j = \{ p_{ij} : m_{ij} = 1 ; c_i + u_{ij} < v_i ; e_i + f_{ij} < \theta \}$$

$$\delta_j = \max F_{ij} - \max_2 F_{ij}$$

Allocate $j^* = \arg\max \delta_j$ to $i^* = \arg\max F_{j^*}$
Issue with fixed priorities

Example

\[ p_{ij} = -f_{ij} \]

Fixed priorities (desirability) may lead to increase the energy consumption on the bottleneck

The reason is that desirability depends on the current working solution, i.e., should be dynamic
Dynamic priorities

Set priorities based on residual energy on each thing according to the current solution

\[ p_{ij} = \theta - (e_i + f_{ij}) \]

RTTA: Real-Time Thing allocation algorithm
Numerical evaluation

- C++ implementation of RTTA vs. standard optimization solver (IBM iLOG CPLEX)
- Randomly generated input

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>10 – 100 s - step 10 s</td>
</tr>
<tr>
<td>Initial battery level</td>
<td>50 – 25 mJ - step 5 mJ</td>
</tr>
<tr>
<td>Execution cost</td>
<td>$2 \cdot 10^{-4} – 6 \cdot 10^{-4}$ mW</td>
</tr>
<tr>
<td>Execution time</td>
<td>7 – 22.5 ms</td>
</tr>
</tbody>
</table>
Numerical results

- Residual battery (average) after one period
  - RTTA aims at maximizing the minimum

50 things, 500 requests

100 things, 500 requests
Numerical results

- Residual battery (distribution) after one period

50 things, 500 requests

100 things, 500 requests
Numerical results

- Computation time
Outline

- IoT horizontal platforms: centralized vs. distributed approach
  - the EU FP7-BETaaS solution: Things-as-a-Service on top of a "local cloud"

- Quality of Service support for M2M applications
  - the BETaaS QoS framework

- Optimized Thing Service selection
  - Exploit multiple TS equivalence to maximize system lifetime: the Agent Bottleneck Generalized Assignment Problem

- Ongoing & future work
Split allocation

\[
\begin{align*}
QoS & - oriented service selection in Things-as-a-Service architectures \\
\text{– ENSIMAG Grenoble-INP – November 17, 2015}
\end{align*}
\]
QoS support integration in FIWARE
References


Thanks!

Enzo Mingozi
Associate Professor @ University of Pisa
e.mingozi@iet.unipi.it