Predicting QoS and energy-consumption in the FOG

A.Brogi, M.Danelutto, D.De Sensi, A.Ibrahim, M.Torquati

Through the fog workshop, January 15th, 2016
Computer Science Department, University of Pisa, Italy

Outline

- Our view of FOG computing
- The need of *autonomic management* for optimizing QoS and power consumption
- A simple use-case scenario: preliminary thoughts and results

Our view of FOG architecture

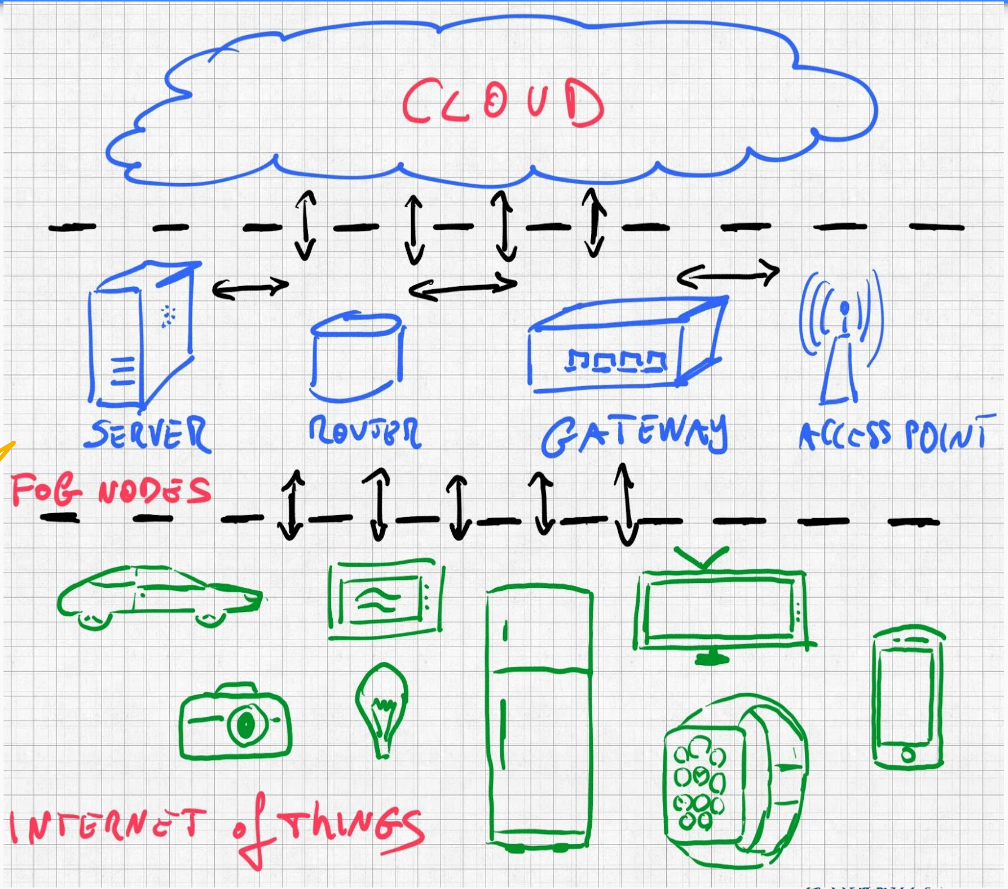
Dynamic distributed architecture:

- using very different types of interconnection networks
- unreliable system, including devices running on batteries

Extremely heterogeneous architecture

- sensors, mobile devices, PC/laptops, hosts, cloud

We aim at targeting the problem of dynamic resources allocation for the "FOG NODES" layer



Our view of parallel FOG applications

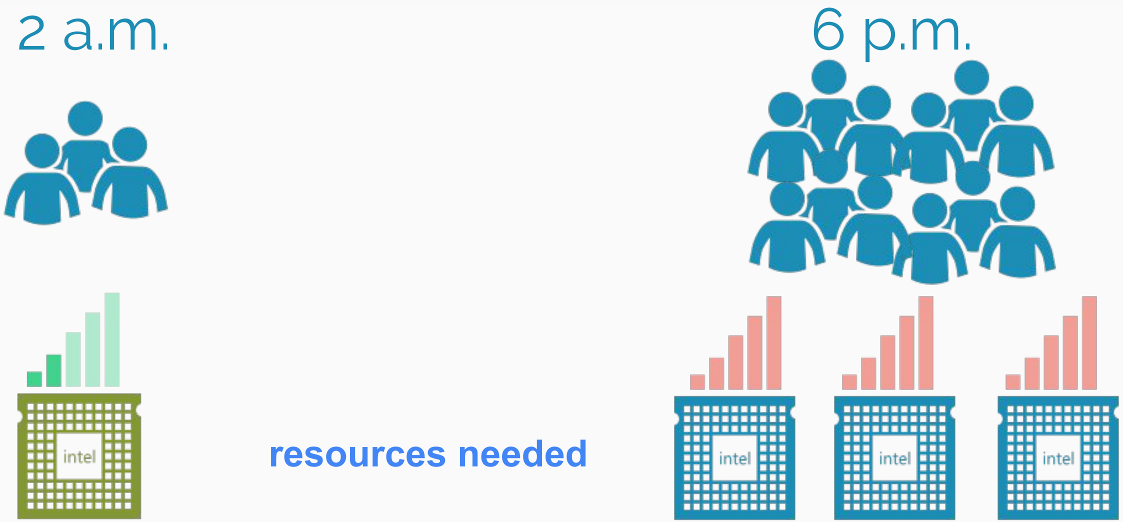
Main characteristics:

- dynamic workload distribution
- dynamic numbers of devices appearing and disappearing

Our approach:

- parallel structure of the application modelled (exclusively) with
  - hierarchical compositions of
  - parallel patterns
  - with autonomic control

Autonomic management needs



2 a.m.

6 p.m.

resources needed

Dimensioning the system resources for the worst case scenario may be unfeasible and too costly

- How many FOG nodes?
- How many resources to use on each node (cores, clock frequency)?

Use case scenario: Network applications

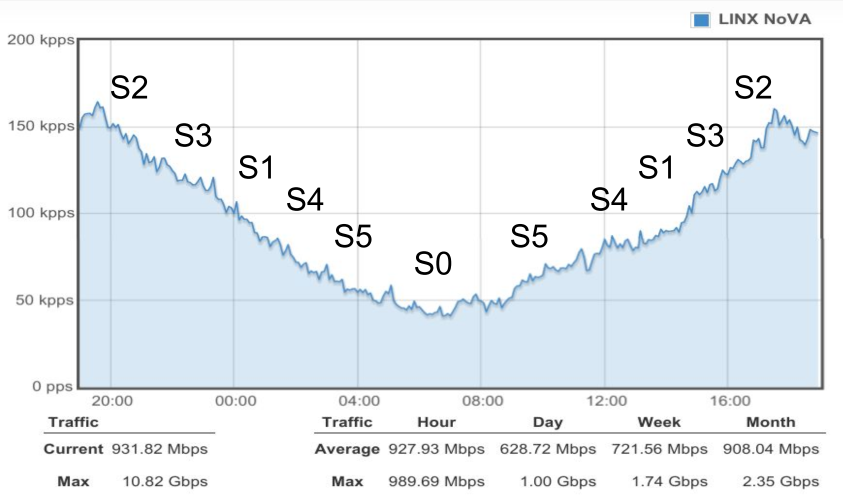
- We are interested in those applications where a set of different workloads $W_1 \dots W_n$ correspond to different phases of the FOG application
- Different phases have different requirements in terms of performance and power consumption (+ performance → + power consumption)

- **Goal:** to dynamically adapt/reconfigure the system resources in order to minimize power consumptions and/or execution time

  - Possible application scenarios:
    - streaming hot-spots
    - network packets analysis
    - .....

Use case scenario: Network applications

Input: a set of possible "Solutions" (S0, S1, S2, S3, S4, S5, S6, S7, ….) all able to sustain a given input rate with a given power cost



Traffic	
Current	931.82 Mbps
Max	10.82 Gbps

Traffic	Hour	Day	Week	Month
Average	927.93 Mbps	628.72 Mbps	721.56 Mbps	908.04 Mbps
Max	989.69 Mbps	1.00 Gbps	1.74 Gbps	2.35 Gbps

Output: find a suitable subset of "Solutions" that provides the desired QoS and minimize the power cost

Option 1:

Experimentally trying out the different configurations

not feasible

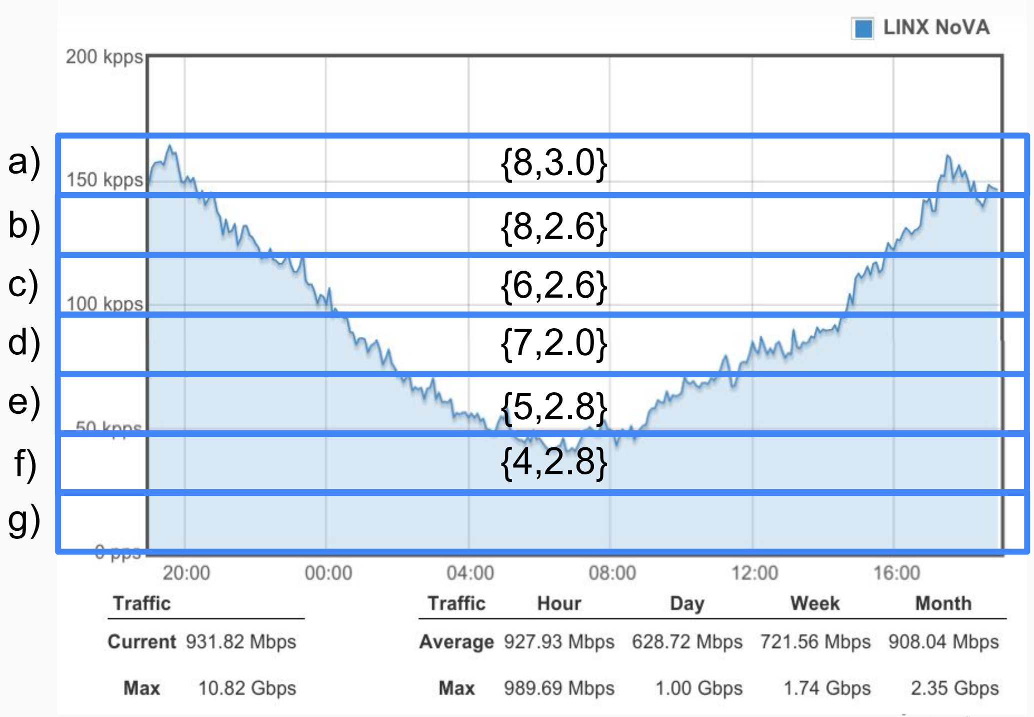
Option 2:

Using a probabilistic simulation tool

Use case scenario: Network applications

- Configuration {C, F} = {number of Cores, Frequency of the cores}
- We know the cost of each solution as well as the cost for the transition among solutions

	Prob.	*Rate*	*Configuration of the solutions*
a)	12%	> 150	**{8, 3.0}**, {7, 3.3}
b)	14%	> 125	**{8, 2.6}**, {7, 3.0}
c)	14%	> 100	**{6, 2.6}**, {6, 2.6}, {5, 3.0}
d)	29%	> 75	**{7, 2.0}**, {6, 2.4}
e)	24%	> 50	**{5, 2.8}**, {5, 2.8}
f)	7%	> 25	**{4, 2.8}**, {3, 3.0}
g)	0	> 0	-

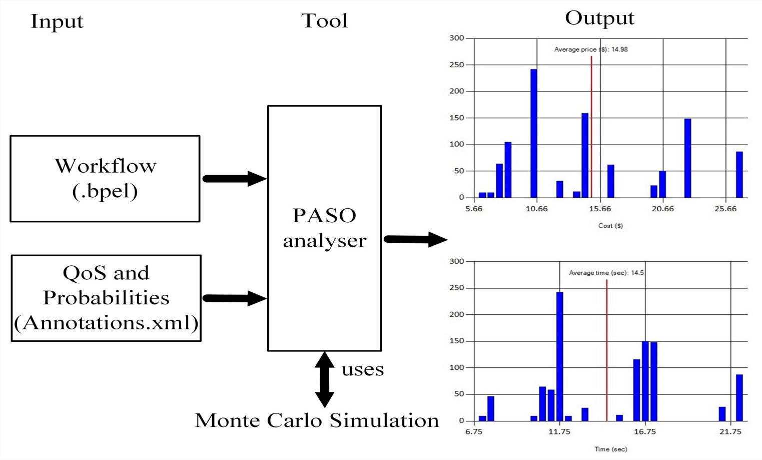


Traffic	
Current	931.82 Mbps
Max	10.82 Gbps

Traffic	Hour	Day	Week	Month
Average	927.93 Mbps	628.72 Mbps	721.56 Mbps	908.04 Mbps
Max	989.69 Mbps	1.00 Gbps	1.74 Gbps	2.35 Gbps

Probabilistic Analyser of Service Orchestration (PASO)

- PASO can probabilistically predict the QoS of a workflow
- Open-source application developed in F# .Net



Why PASO?

Can address several challenges in predicting QoS

1. Different results of service invocations
2. Non-determinism in the workflow
3. Correlation in parallel branches
4. Complex dependency structure

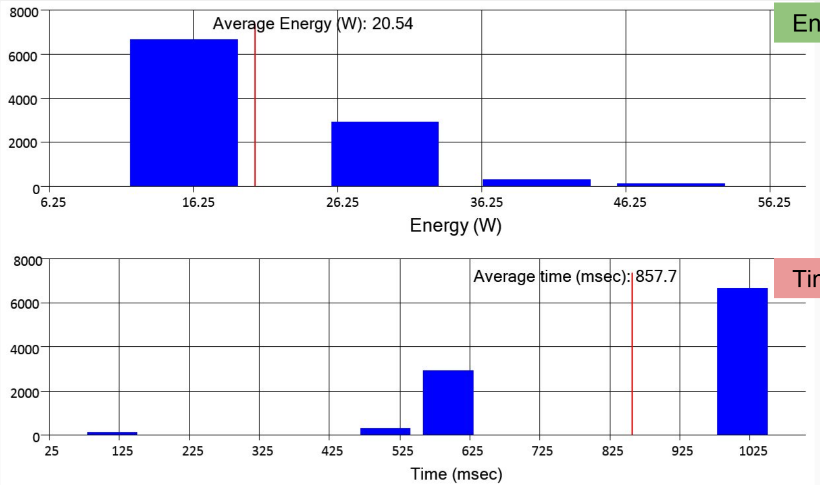
Attributes:

- Service Time (ms)
- Energy (W)
- ..

Source code for the PASO analyser is available at https://github.com/upi-bpel/paso

L. Bartoloni, A. Brogi, and A. Ibrahim, Probabilistic prediction of the QoS of service orchestrations: A truly compositional approach,ICSOC 2014, LNCS 8831, pp. 378–385, November 3-6, 2014

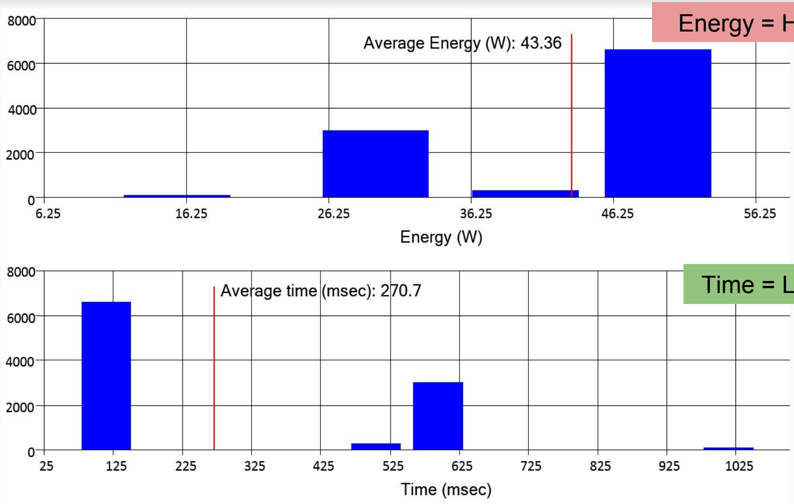
Preliminary results obtained using PASO



Energy = Low

Time = High

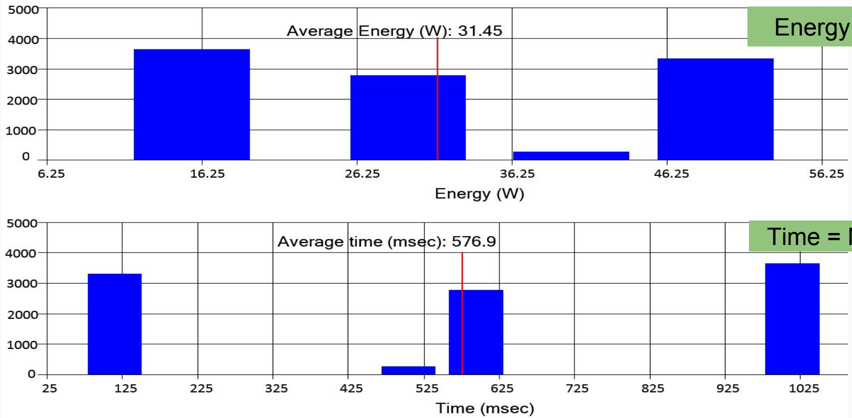
Configuration 1



Energy = High

Time = Low

Configuration 2



Energy = Medium

Time = Medium

Configuration 3

Conclusions and Future Work

Conclusions:

- We aim at targeting the problem of dynamic allocation of resources for the FOG nodes
- Our case study focuses on applications that have dynamic workload distributions
- Preliminary results produced with the PASO tool

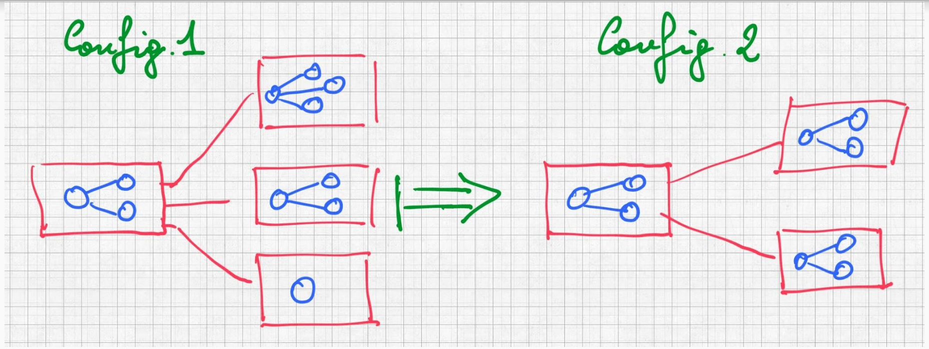
Future works:

- more experiments needed trying more complex and accurate functions for energy and time
- validation of the results

Thank you !

Any questions ?

Why using hierarchical parallel patterns ?



- .... because:
  - well-known parallel structure
  - simpler to manage and deploy
  - easier to model the execution behaviour
  - easier to reconfigure/adapt at run-time

- The autonomic hierarchical approach has been used in other contexts: distributed-systems, global computing, cloud, ...
  - we think this is the way to go for the FOG