Resource management from Cloud to Edge systems

Massimo Canonico - University of Piemonte Orientale

Who we are

- Distributed Computing System group

Prof. Cosimo Anglano

Dott. Marco Guazzone

Dott. Massimo Canonico

Research interests

- Computing platforms
  - Cloud computing
  - Edge and Fog computing
  - Femtocloud computing
- What we have used
  - Game theory
  - Fuzzy controller
- Efficient resource management
  - Fault-tolerant and energy-aware algorithms
  - Cloud federation

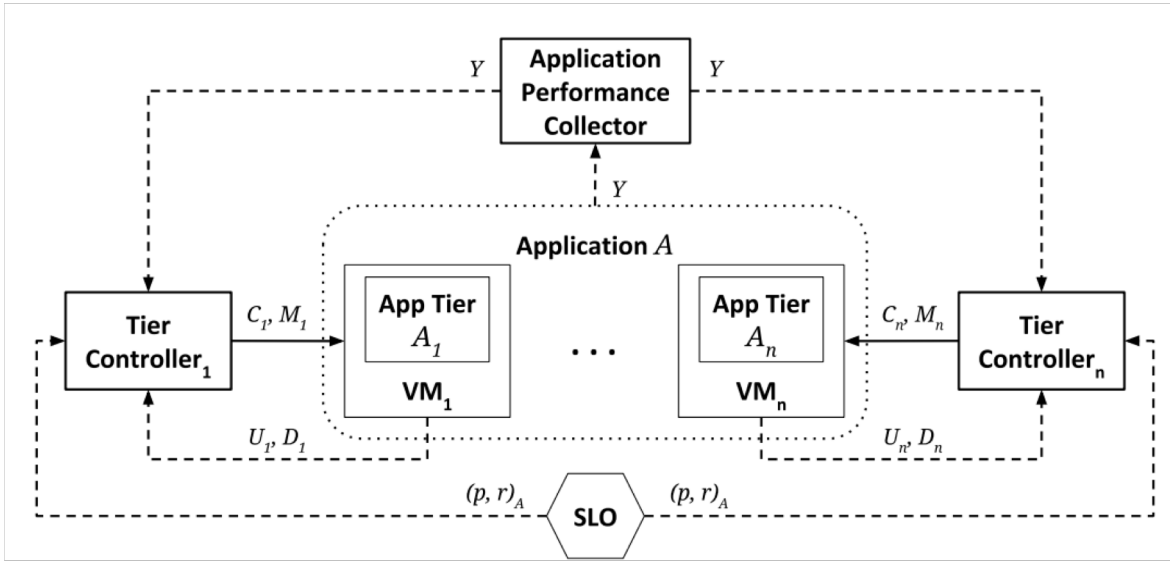
Fault-tolerant and energy-aware algorithms

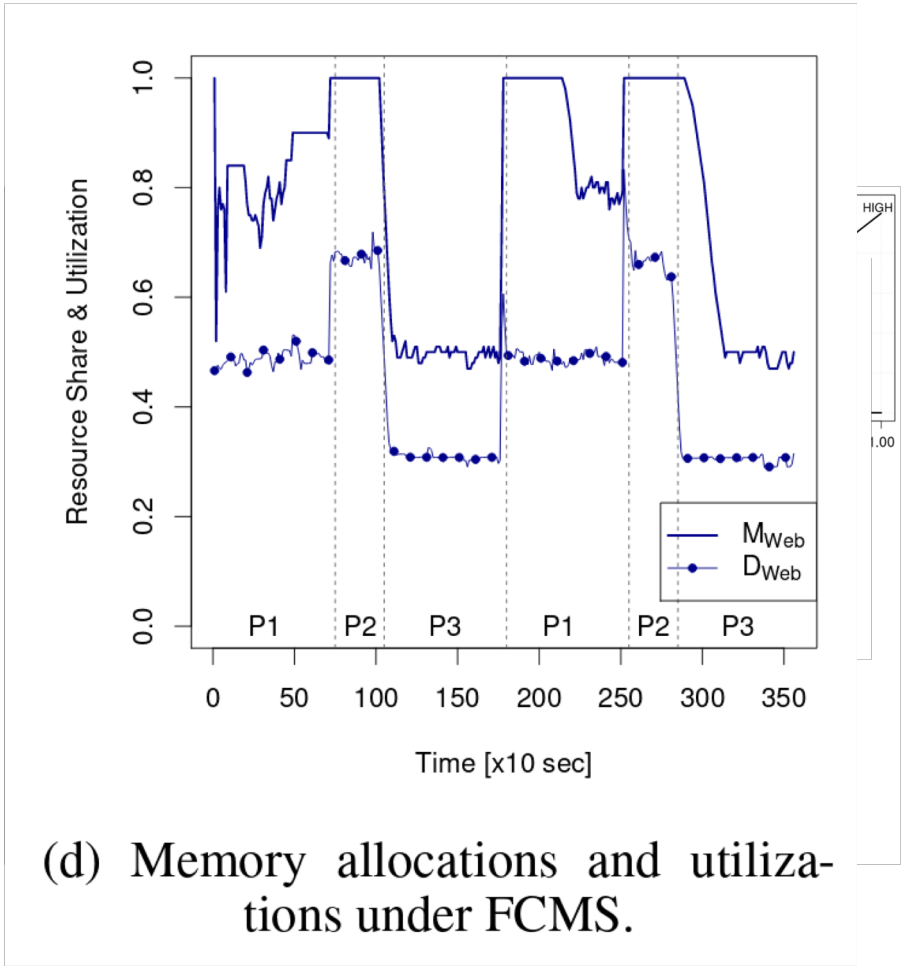
- Knowledge-free algorithms
  - Forecast with error
- Fault tolerant
  - Replication
  - Checkpoint
- Energy-aware
  - Consolidation: Switch-off/Switch-on resource
  - Cost Vs Benefits

Cloud federation

CloudTUI-FTS

Fuzzy controller





(d) Memory allocations and utilizations under FCMS.

Game theory

Fog Internet Provider (FIP)

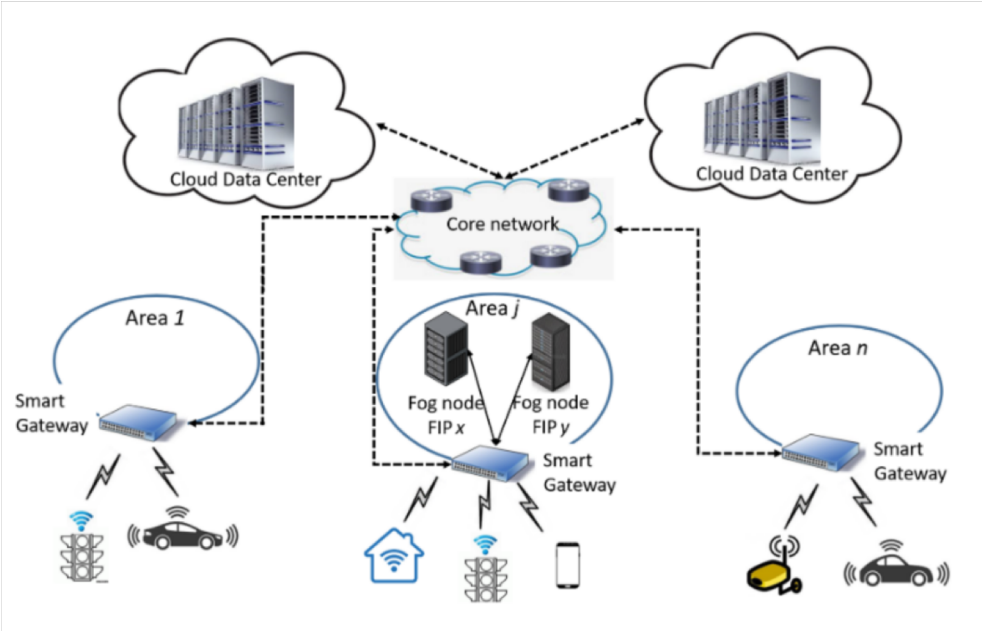
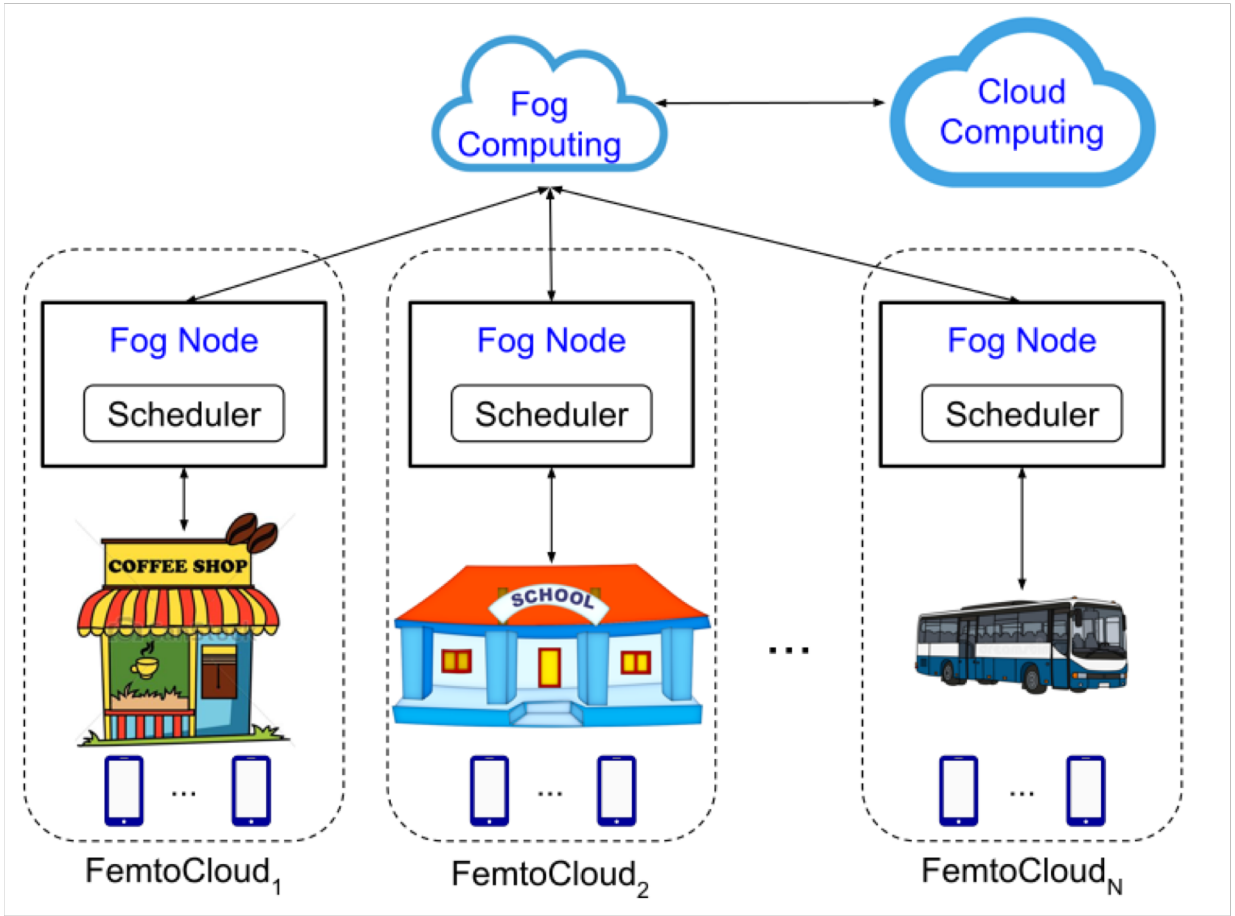


TABLE 2. PARAMETERS USED IN THE EXPERIMENTAL SCENARIOS. SUBSCRIPTS i AND j TAKE VALUES ON THE SET $\{1, 2, 3\}$.

	Parameter	Value
$\|App(i)\|$	Number of applications associated to FIP i	1
$E_{i,j}$	Electricity price for FIP i in area j	0.0001 \$/Wh
$\|FN(i,j)\|$	Number of fog nodes for FIP i in area j	3
$L_{i,j}$	Penalty rate for FIP i and application j	0.022 \$/h
m	Number of FIPs	3
n	Number of applications	3
Q_i	Max request processing time for application i	0.7 sec
$R_{i,j}$	Revenue rate for FIP i and application j	0.0022 \$/h
$U_{i,j}$	CPU demand for any VM j and fog node i	0.05
$W_j^{\max}$	Max power consumption of fog node j	200 W
$W_j^{\min}$	Idle power consumption of fog node j	100 W
τ_j	Request processing time of any VM j	0.5 sec

Femtocloud



Algorithm 1: The WQR-UD scheduling algorithm.

```
1  procedure Schedule(T,D,τ_r)
       Input: task set T, device set D, replication threshold τ_r.
2      t ← GetOldestTaskWithLowestNumReplicas(T,τ_r)
3      d ← GetIdleDevice(D)
4      if t ≠ nil and d ≠ nil then
5          if CheckpointExist(t) then
6              RunTaskReplicaFromCheckpoint(t, d)
7          else
8              RunTaskReplica(t, d)
9          end
10         IncrNumTaskReplicas(t)
11     end
12 end
13 procedure Main(T,D,e,τ_r)
       Input: task set T, device set D, event e, replication threshold τ_r.
14     if EventType(e) = NewTask then
15         t ← GetTask(e)
16         InsertTask(T, t)
17     else if EventType(e) = TaskDone then
18         t ← GetTask(e)
19         RemoveTaskReplicas(t, T)
20     else if EventType(e) = DeviceIdle then
21         d ← GetIdleDevice(e)
22         InsertDevice(D, d)
23     else if EventType(e) = DeviceGone then
24         RemoveDevice(D, d)
25         t ← GetTask(e)
26         if t ≠ nil then
27             DecrNumTaskReplicas(t)
28         end
29     end
30     Schedule(T,D,τ_r)
31 end
```