IEEE WCCI 2020
IEEE World Congress on Computational Intelligence
Virtual Conference – July 19-24, 2020

A Demand-driven, Proactive Tasks Management Model at the Edge

Anna Karanika[1], Panagiotis Oikonomou[1], Kostas Kolomvatsos[2], Thanasis Loukopoulos[3]

[1] Dept. of Informatics and Telecommunications. University of Thessaly, Lamia, Greece

[2] Dept. of Informatics and Telecommunications. University of Athens Panepistimiopolis, Athens, Greece

[3] Dept. of Computer Science and Biomedical Informatics. University of Thessaly, Lamia, Greece.

Introduction

- Tasks offloading is a key research topic in
  - Edge Computing (EC)
  - Internet of Things (IoT)
- Tasks to EC Nodes mapping
- Processing at the EC
  - Latency minimization
  - network traffic minimization
- Tasks offloading and Mobile Cloud Computing (MCC)
  - Latency increased

Motivation

- The advent of the IoT and EC defines new requirements for tasks management
  - Dynamic environment
  - Intense variability regarding EC nodes status
- Users mobility
  - spatio-temporal requirements
  - Complexity increment
- Real time responses

Motivation

- Our goal:

  investigate tasks offloading methodologies to decide when a task should be executed locally or be offloaded to other peer nodes

- Distributed environment
- Users mobility
- EC nodes exchange
  a) tasks demand and
  b) contextual information

- Optimize
  - Load balancing
  - Latency
  - Execution time

Contributions

1) A georeferenced task management scheme where computation offloading is decided based on data present at every node and tasks demand.

2) A **Fuzzy Logic Controller (FLC)** to indicate when a task should be offloaded or not.

3) An extensive experimental evaluation that reveals the pros and cons of the proposed approach

System Model (1)

- We assume an EC scenario with computational resources (EC nodes) spanning across different geographically distributed locations

- EC nodes connectivity
  - IoT devices
  - Cloud

- EC nodes characteristics
  - Collect and store data
  - Execute tasks

System Model (2)

- We consider that tasks exhibit specific characteristics like
  - complexity
  - demand

- Task Demand
  - the number of users/devices asking for their execution.

- Users mobility
  - Tasks Demand Vector (TDV) $TDV = \{e_1^t, e_2^t, \ldots, e_M^t\}$
  - EC nodes update their TDV

a) keep the execution locally for popular tasks
b) Offload non popular tasks

Georeferenced Tasks Demand Management

Tasks Demand Indicator

- Incoming TDVs could exhibit different information
  - The *i*th task may be requested at n_j but not at n_k
- The locality of the demand is important
  - EC nodes keep the TDVs only for peers being in a distance *p*

$$[\{TDV_1^1, TDV_2^1, \dots, TDV_D^1\}$$

$$\{TDV_1^2, TDV_2^2, \dots, TDV_D^2\}$$

$$\dots$$

$$\{TDV_1^w, TDV_2^w, \dots, TDV_D^w\}]$$

n_j

1. Monitor local TDV
2. Receive incoming TDVs
3. Kernel Density Estimator (KDE)

Local Demand Indicator (LDI)

Group Demand Indicator (GDI)

Georeferenced Tasks Demand Management

Uncertainty Driven Decision Making Model

Local Demand Indicator (LDI_{ei})

Group Demand Indicator (GDI_{ei})

FLC

Task Offloading Indicator (TOI_{ei})

$LDI_{ei} \rightarrow 1$ $GDI_{ei} \rightarrow 1$ $TOI_{ei} \rightarrow 1$

$LDI_{ei} \rightarrow 0$ $GDI_{ei} \rightarrow 0$ $TOI_{ei} \rightarrow 0$

FUZZY LOGIC RULE BASE

No	LDI_{e_i}	GDI_{e_i}	TOI_{e_i}
1	Low	Low or Medium	Low
2	Low	High	High
3	Medium	Low	Low
4	Medium	Medium or High	Medium
5	High	Low or Medium	Low
6	High	High	Medium

Performance Indicators

- Metric 1) Number of correct decisions Δ

loss function $\lambda(C,R_s)$

C=1 → s executed locally
C=0 → s is offloaded

$R = \langle IT_s, RT_s, d_s \rangle$

Execute task locally : $\lambda(C=1, IT_s)=0$
$\lambda(C=1, RT_s)=0$

Offload task : $\lambda(C=0, IT_s)=MGT$
$\lambda(C=0, RT_s)=RST$

Performance Indicators

- Metric 1) Number of correct decisions Δ

loss function $\lambda(C,R_s)$

$$Cost_t(C) = \sum_{r \in R}^{|R|} (C, r), R \in \{IT_s, RT_s, d_s\}$$

C=1 → s executed locally
C=0 → s is offloaded

Correct decision:

$R = \langle IT_s, RT_s, d_s \rangle$

$$\boxed{\begin{array}{l} Cost_s(C=1) < cost_s(C=0) \&\& \ FLC \rightarrow local\ execution \\ Cost_s(C=1) > cost_s(C=0) \&\& \ FLC \rightarrow offloading\ action \end{array}}$$

Execute task locally : $\lambda(C=1, IT_s)=0$
$\lambda(C=1, RT_s)=0$

Offload task : $\lambda(C=0, IT_s)=MGT$
$\lambda(C=0, RT_s)=RST$

Performance Indicators

- Metric 2) average time τ required to take a decision
  - the time spent to estimate KDE
  - the time spent for FLC to produce the final TOI

Experimental Setup

Real world workload

142 users
4500 web services
64 time slots

$N \in \{50,100,500,1000\}$
$W \in \{10\%,50\%,100\%\}$
$E \in \{5,10,50,100\}*10^3$
$T \in \{0.5,0.7\}$
100 iterations for each experiment

Performance Assessment

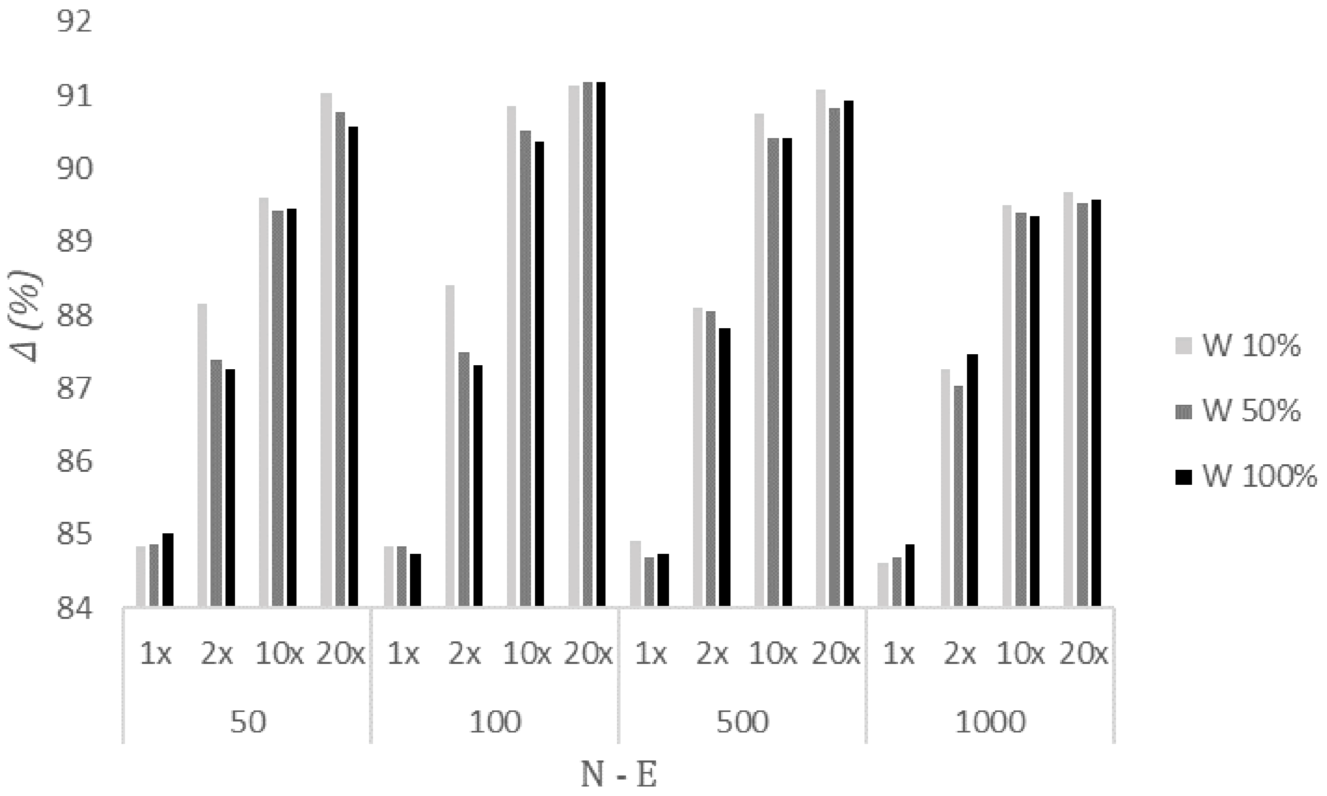

Fig. 1. Percentage of correct decisions; k = 10%

Performance Assessment

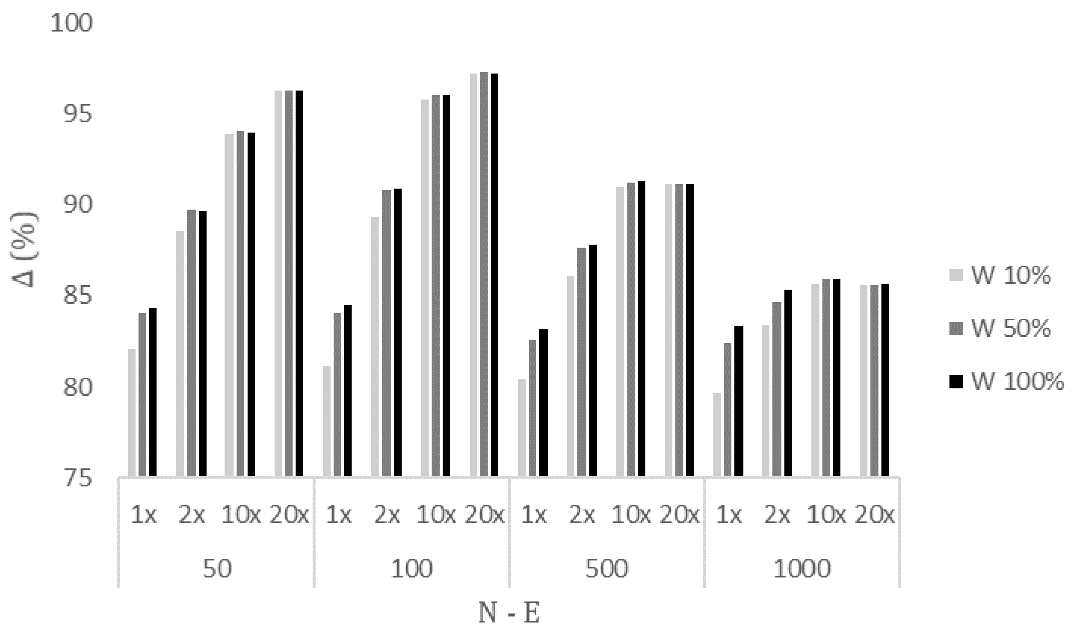

Fig. 2. Percentage of correct decisions; T=0.5

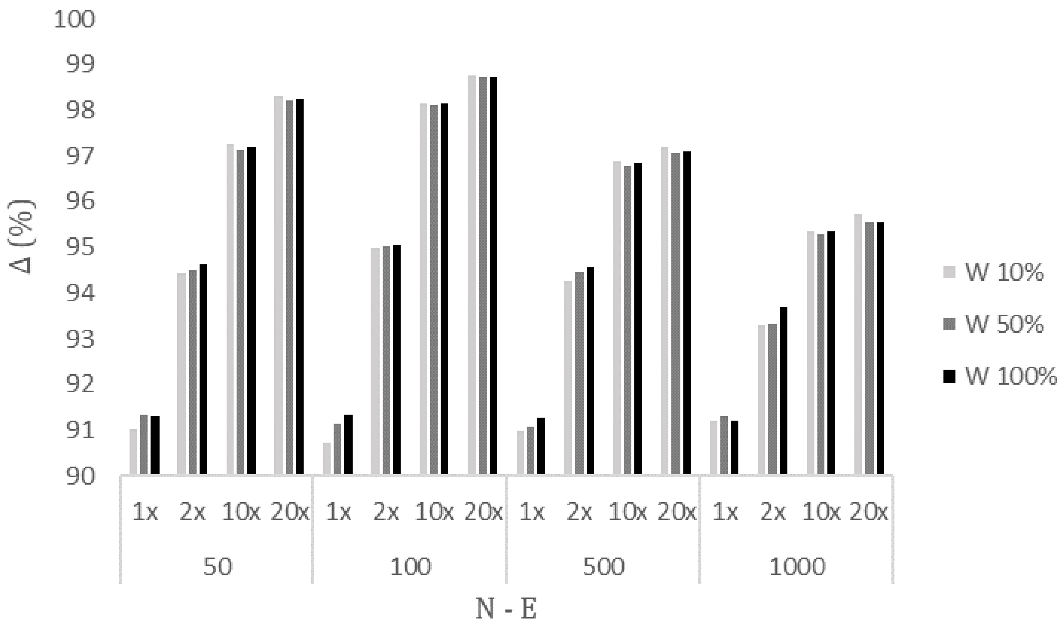

Fig. 3. Percentage of correct decisions; T=0.7

Performance Assessment

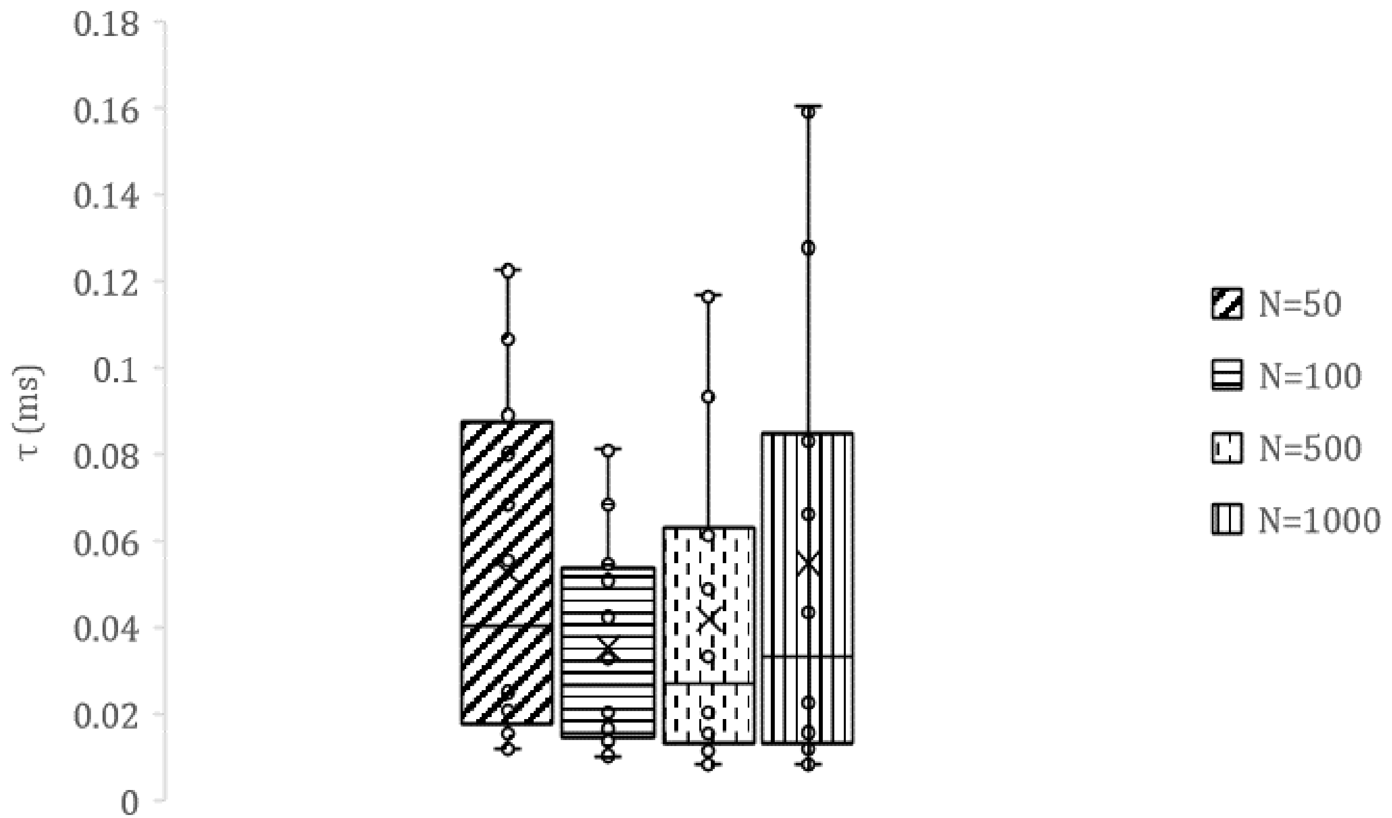

Fig. 4. Execution Time

Performance Assessment

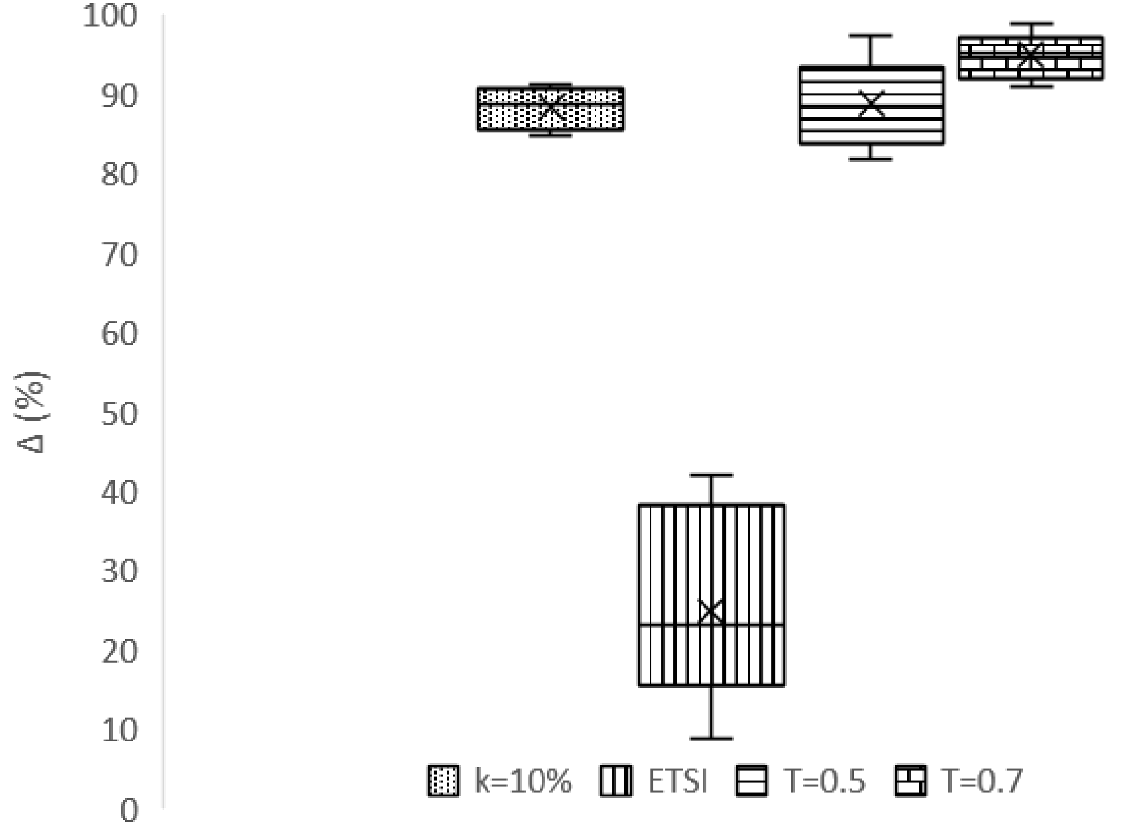

Fig. 5. Comparative Results

Future Work

- Conduct experiments in a real world IoT environment
- Optimal Stopping Theory
- Integrate Scheduling Component

Thank you!