University of Glasgow

School *of* Computing Science

Essence: Pervasive & Distributed Computing

Optimal Load-Aware Task Offloading in Mobile Edge Computing

Odysseas Polycarpou, Christos Anagnostopoulos, Kostas Kolomvatsos

School of Computing Science, University of Glasgow

WORLD CHANGING GLASGOW

8th CDSR Conference; 23-25 May 2021

Table of Contents

Introduction – Mobile Edge Computing and Task Offloading

- Mobile Edge Computing (MEC) is a recently emerged computing paradigm
- Brings the user **closer** to the Edge of the network
- **Offloading tasks** from Mobile Nodes to MEC Servers to be executed for efficiency

- **Challenge**: Seek the **best** candidate server at the **best** time to offload tasks

Context Description

- We tackle the problem of Task Offloading in a MEC environment
- Many MEC servers are **geo-distributed** and their **CUP load** is observed (fixed intervals)

- A Mobile Node (e.g., mobile device, vehicle) passes through MEC Server sequences
- We seek the **best candidate server** at the **best possible time** to offload our tasks (e.g., application specific tasks like image classification, data cleaning, outliers detection…)

- We introduce and implement three **Time-optimized Sequential Decision Making Models**
- Real data sets of ~4000 load observations for experimentation

Related Work & Motivation

- Several studies tackle the problem using different assumptions and parameters
- We rely on the principles of the **Optimal Stopping Theory (OST)**
- Differences are:
  - We apply our OST Models in sequential decision making
  - We focus on the MEC Server CPU Load at each time instance to make our decision
  - We take in account time as well
- We aim to make task execution **faster** by selecting the best possible server to offload the tasks

OST in Computing Science

- OST refers to the task of finding the best time to **stop** and **take** a particular decision/action in order to optimize an objective function.
- We adopt OST: **when** to offload the tasks

- Let a sequence of N observable servers {N1, N2, N3,…} and a sequence of the corresponding reward functions.
- A **Reward** function determines how efficient our Task Offloading decision is; (the higher the reward the better our decision)

Task-Offloading Sequential Decision Models

University of Glasgow

School of Computing Science
Essence: Pervasive & Distributed Computing

Random Model

- Let a fixed, pre-defined probability p>0.
- The mobile node decides on offloading its tasks **with probability p.**

OST Secretary Model

- Observes the first 37% of the sequence (sample) N
- Finds the **best** server in this sample (benchmark) which refers to the minimum load.
- Offloads the tasks at the **next** available server **better** than the benchmark.

OST House Selling Model

- Let a discount factor r in (0,1).
- Compute **optimal decision values** based on the r factor using Dynamic Programming
- If the current load **exceeds** the current optimal decision value, then offload the tasks to the associated server.

Implementation

- Development using Python (Jupyter Notebook)
- Reads .csv files in a streaming mode.
- Made dynamic to allow the application to specify:
  - Size of server sequence N
  - Probability p (Random Model)
  - r factor (House Selling Model)

- We compare against the **Optimal Solution**, where the optimal server in *known* a-priori.


LIVE DEMO

Performance Assessment– Optimizing House Selling

- Discount factor r where House Selling Model *beats* Secretary Model (r=0.015)
- Gets better and better by minimizing the r factor
- But: r > 0 (cannot be 0)
- The r factor depends on the different N size sequences
- For demonstration, N=200

	Model	Optimal Means	Offloading Means	Mean Load Difference
1	Random(P = 0.05)	37.8482	42.5971	4.7489
2	Random(P = 0.1)	37.8482	42.2957	4.4475
3	Random(P = 0.2)	37.8482	42.709	4.8608
4	Random(P = 0.3)	37.8482	43.6277	5.7795
5	Random(P = 0.5)	37.8482	42.1202	4.272
6	Secretary	37.8482	39.0581	1.2099
7	House Selling	37.8482	39.024	1.1758

Conclusions

- We apply Optimal Stopping Theory (OST) models to address the problem of Load-aware Task Offloading in Mobile Edge Computing.

- Our OST models achieve reward *very close* to the Optimal solution
- Optimized the House Selling Model
- Can outperform other baseline solutions in terms of CPU load

- **Future Agenda**: Investigate network conditions (e.g., link availability, connectivity, latency) when taking sequential decisions
- Turn the model to be **context-aware (e.g., CPU load, requests, uplink bandwidth, data freshness, application-specific deadline, etc.)**

University of Glasgow

School *of* Computing Science

Essence: Pervasive & Distributed Computing

Thank You!

Odysseas.Polycarpou@gmail.com

Christos.Anagnostopoulos@glasgow.ac.uk

Essence Lab: http://www.dcs.gla.ac.uk/essence/