COST-AWARE TASK SCHEDULING IN FOG-CLOUD ENVIRONMENT

Tina Samizadeh Nikoui
Tina Samizadeh Nikoui, Amir Masoud Rahmani, Ali Balador and Zeinab Bakhshi
COST-AWARE TASK SCHEDULING IN FOG-CLOUD ENVIRONMENT

Name: Tina Samizadeh Nikoui

Tina Samizadeh Nikoui received her BS degree and MS degree in computer engineering from Islamic Azad University, Tehran. Currently, she is a full-time PhD Candidate in Computer Engineering-Software Systems at Science and Research Branch - Islamic Azad University, Tehran. Her research interests are distributed systems, cloud computing, internet of things and fog computing.
INTRODUCTION

WHY WE NEED FOG COMPUTING

- Large amounts of data
- Delay-sensitive applications
- Performance

Fog & Edge computing

- Bringing resources closer to the user
- Providing an intermediate layer between device & cloud layers

Cloud Computing

- Lower latency
- Security and Privacy
- Higher QoS

Fog Computing
Task scheduling is a decision-making process and deals with the allocation of resources to tasks over a defined period of time in order to optimize one or more objectives [3].

Evolutionary algorithms such as particle swarm optimization (PSO), genetic algorithms (GA), bees life algorithm (BLA) and ant colony optimization (ACO) can be applied.

Due to global optimization and search ability of genetic algorithm, it is widely used to solve task scheduling problems.
## RELATED WORKS

### 1. GA-based scheduling algorithm
- Decrease Makespan and financial cost.
- Weighting coefficients for flexibility.
- Not considering Time-sensitivity of tasks and deadline.

### 2. Cost-aware scheduling algorithm
- Increase the profit of service providers.
- Collaboration between multiple fog nodes is not addressed.

### 3. ACO-based scheduling algorithm
- Time-constrained workflows.
- PSO generates solutions based on problem objectives.
- Min-Min algorithm is used to resolve conflicting tasks.
- Not considering waiting time, which has a critical impact on response time.

---


A Task request is forwarded to MasterFogNode.

MasterFogNode estimates the completion time of tasks with respect to capability of available resources and requirements of tasks.

Considering the processing capability of available resources and requirements of tasks, task may be rejected. Otherwise it will be sent to corresponding resources.

Each node should process receiving tasks and send results back to MasterFogNode.

Corresponding responses will be sent to the requester.
CHROMOSOME ENCODING

Each possible solution in GA is represented by a chromosome (individual).

We use integer coding that includes the resource identifications.

A set of chromosome in a generation is called a population.

The first generation of chromosome (initial population) is generated randomly.
CHROMOSOME ENCODING

{task_0, task_1, task_2, …., task_n}

Chromosome encoding

{Node_0, Node_1…, Node_m}

Task_0 ↦ Node 3
**Fitness Function**

**Objectives:**
- Cost ▼
- Success rate ▲

**Fitness Function**

\[ f(x) = \frac{\text{Cost}}{\text{Success Rate}}. \]

**Response Time**
- Waiting time [8]
- Data transmission time [9]
- Processing time
- Propagation Time [10]

**Cost [11]**
- Cost of processing
- Cost of network bandwidth

**Number of tasks that are successfully completed within the predefined deadline**

---

Tasks are assumed as BoT [7]

Master Fog node uses Cost-Aware Genetic algorithm to assign tasks to fog and cloud nodes.

---

The iFogSim simulator that is suitable for resource management in fog environment, is used for evaluation. [12]

We extended iFogSim by adding some classes for the genetic algorithm and new scheduling approaches.

Round-Robin (RR) is a simple, but well-known algorithm that is one of the most common algorithms for resource allocation.[13]

Minimum Response Time assigns each tasks to the resource with minimum completion time is used for comparison. [14]

Cost-aware genetic-based (CAG) task scheduling algorithm improves the cost efficiency of real-time applications with hard deadlines in fog-cloud environments.


We investigate the performance of CAG, RR and MCT algorithms in terms of Cost and Success Rate.

**Cost per Instruction**

The financial cost that is spent for a single instruction.

This metric is calculated by dividing the total financial cost to total number of instructions for successful tasks.

**Success rate**

The fraction of tasks that are completed within deadline.

This metric is calculated by dividing the number of successfully completed tasks to total number of tasks.

**Cost**

The financial cost that is spent for execution of tasks.
The experimental evaluation shows that

1. CAG algorithm has higher success rate than the RR and MRT algorithms.
2. ↑ number of computational resources ↓ response time

The underlying reasons are

1. Lower latency of added resources (fog nodes) (shorter latency of fog layer)
2. RR and MRT algorithms does not consider a global perspective of problem.
RESULTS AND DISCUSSIONS- COST

The experimental evaluation shows that

1. RR has higher cost than the others.
2. MRT compared to CAG has higher cost in most cases.

The underlying reasons are

1. Number of rejected tasks which can not be completed in deadline.
2. Instructions that are belonged to missed tasks has a negative impact on cost.
RESULTS AND DISCUSSIONS- COST PER INSTRUCTION

Considering Cost per Instruction, it can be concluded that a trade off between cost and success rate can be achieved using CAG algorithm.
CONCLUSIONS

- Efficient task scheduling is a critical challenge for cloud-fog systems.
- Genetic-based task scheduling algorithm
- Trade-off between throughput and cost
- Hard real-time applications.

- Using iFogSim simulator for evaluation.
- CAG performs better than RR and MRT algorithms in terms of financial cost and success rate.

FUTURE WORKS

- Evaluate the performance under realistic workloads and applications.
- Workflow scheduling and considering dependent tasks
Thank You

CSI/CPSSI International Symposium on Real-Time and Embedded Systems and Technologies (RTEST)
June 10-11, 2020
Institute for Research in Fundamental Sciences (IPM), Tehran, Iran