



Utility-based Uploading Strategy in Cloud Scenarios

Ziqi Wan, Jie Wu, and [Huanyang Zheng](#)

Temple University, Philadelphia, PA, USA



Outline

1. Introduction

2. Problem Formulation

3. Models

4. Analysis

5. Simulation Results

6. Conclusions and Future Work

Introduction

- Why do we need to upload tasks to Cloud?
 - Limitation of our own devices



- Powerful Cloud



User's view

- What really matters for users?
 - Time
 - Money



- What does a good scheduler need to do?
 - Balance Time and Money





Outline

1. Introduction

2. Problem Formulation

3. Models

4. Analysis

5. Simulation Results

6. Conclusions and Future Work



Problems

- How to save money for users?
 - We rent devices in the Cloud together!
 - However, we need to wait and find enough people to share
- With a limited budget
 - If you want to go quickly, go alone
 - If you want to go far, go together
- Key issue
 - Should I wait for others?
 - How long should I wait?

Simplification

- Balance customers' satisfaction and data center cost
- Simplification (time vs cost):
 - Task Finishing Time \rightarrow Customers' Satisfaction
 - Running Machines Cost \rightarrow Data Center Cost





Outline

1. Introduction

2. Problem Formulation

3. Models

4. Analysis

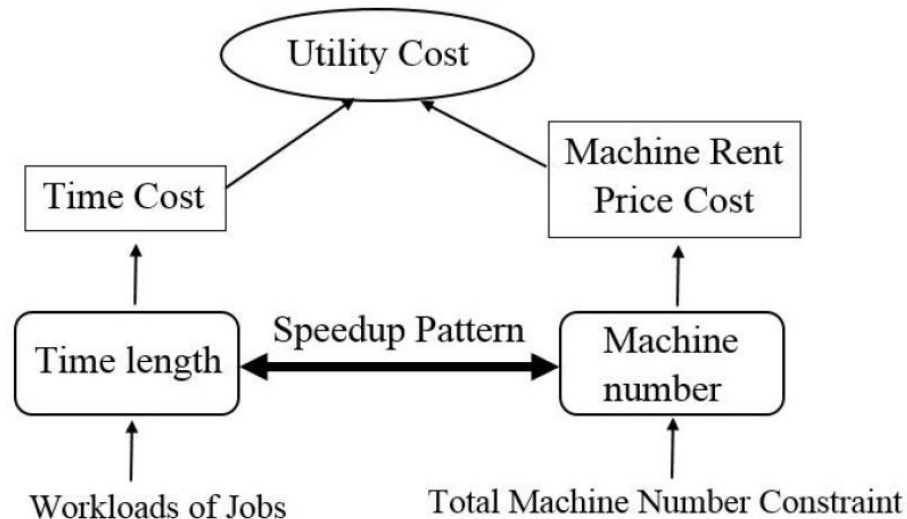
5. Simulation Results

6. Conclusions and Future Work



Utility-based VM Model

- Utility Cost Model



- Our objective for users:
 - minimize the utility cost.



Utility-based VM Model

- Utility Cost Model

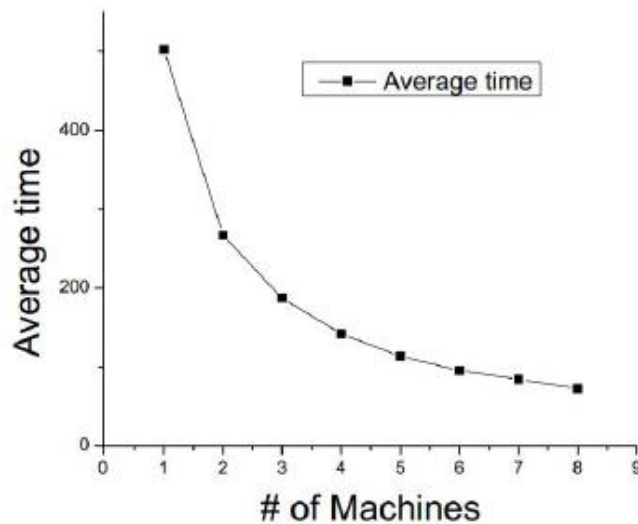
$$U = f(t) - C$$

- $f(t)$ is the time cost, i.e., $f(t) = b - at$
- C denotes the money cost
- Utility decreases with respect to time

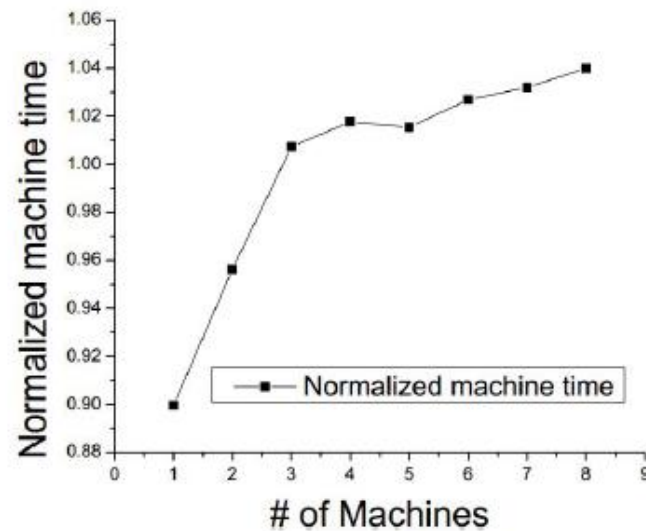


Utility-based VM Model (Cont'd)

- More VMs -> run jobs faster.
- However, the processing speed does not linearly increase with the number of machines.



(a) Average Time



(b) Normalized Machine Time

Figure 5.5: Pentomino



Outline

1. Introduction

2. Problem Formulation

3. Models

4. Analysis

5. Simulation Results

6. Conclusions and Future Work



Analysis

- Arrival rate estimation
 - Learn the pattern
- Fixed Arrival Rate
 - Easy to solve.
 - No one comes, go alone
 - Crowded users, go together
- Decision under dynamic Arrival
 - Set an observation window
 - Then apply the former idea



Analysis

- In some cases, users only care about the time, and pay little attention to the rent price.
 - minimize the time cost first, then consider minimizing the machine rent price.
- In some cases, users only care about the Price
- In general, we want to maximize the utility.
 - We provide a greedy algorithm to make a balance between performance and time complexity.



Outline

1. Introduction

2. Problem Formulation

3. Models

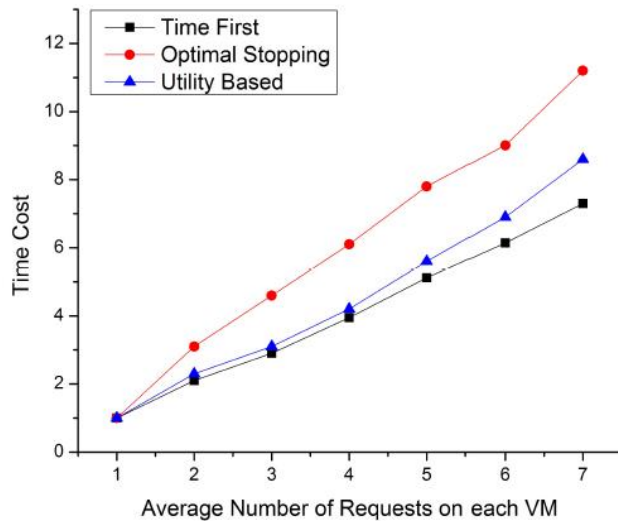
4. Analysis

5. Simulation Results

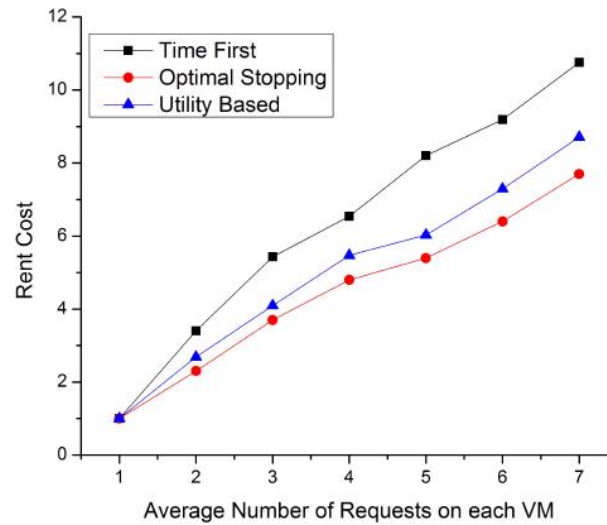
6. Conclusions and Future Work



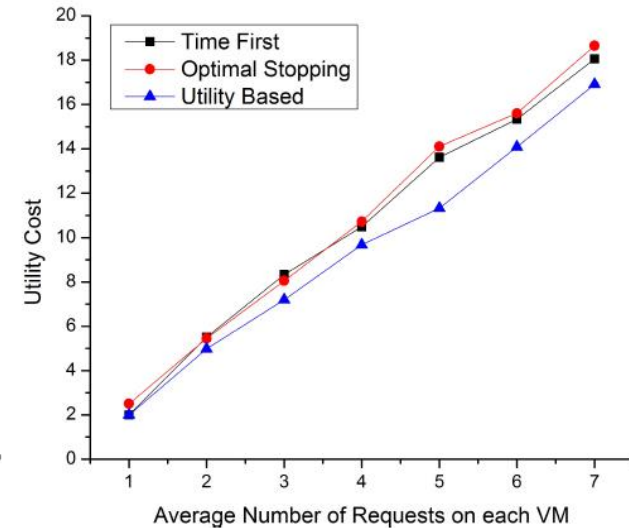
Simulations



(a) Time Cost



(b) Rent Cost

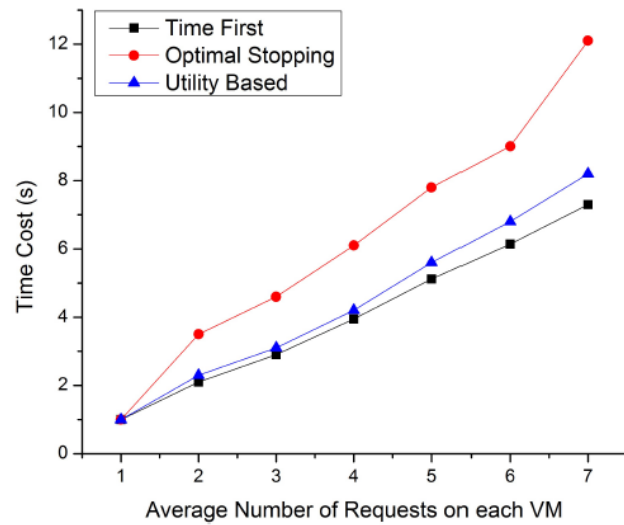


(c) Utility Cost

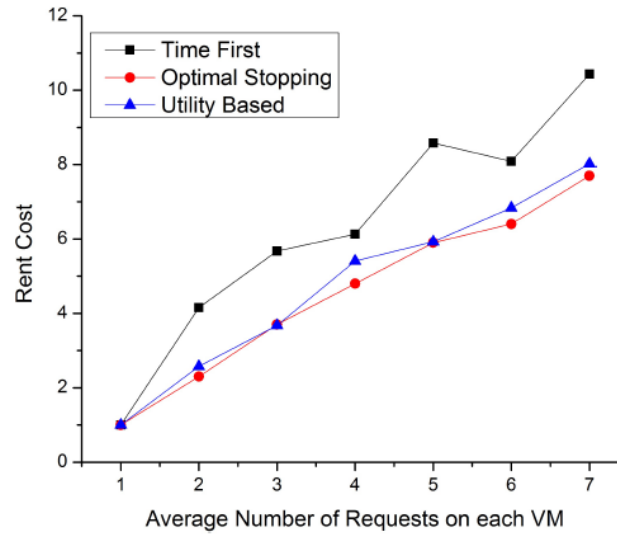
Simulation results of 3 algorithms under hybrid speedup pattern.



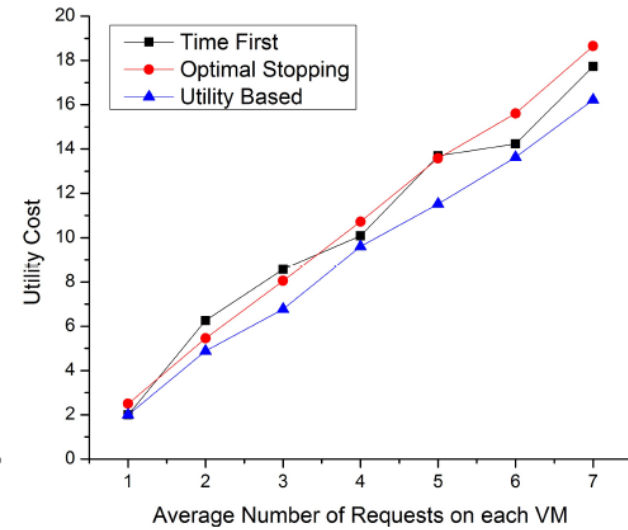
Simulations (cont'd)



(a) Time Cost



(b) Rent Cost



(c) Utility Cost

Trace-based results of 3 algorithms under hybrid speedup pattern.



Outline

1. Introduction

2. Problem Formulation

3. Models

4. Analysis

5. Simulation Results

6. Conclusions and Future Work



Conclusions

- We consider the design and analysis of utility-based scheduler in the cloud environment. Unlike all existing works, we propose the notion of utility for the Virtual Machine management.
- The model presented here opens the door for an in-depth study of how to schedule in the presence of phase overlapping. There are a wide variety of open questions remaining with respect to the design of algorithms that minimize response time.

Thank you!

Questions?

huanyang.zheng@temple.edu