Adapting Market-Oriented Scheduling Policies for Cloud Computing

Mohsen Amini Salehi
Rajkumar Buyya

Cloud Computing and Distributed Systems (CLOUDS) Laboratory,
Department of Computer Science and Software Engineering,
The University of Melbourne, Australia
mohsena,raj@csse.unimelb.edu.au

THE UNIVERSITY OF MELBOURNE

MELBOURNE CLOUDS LAB

Introduction

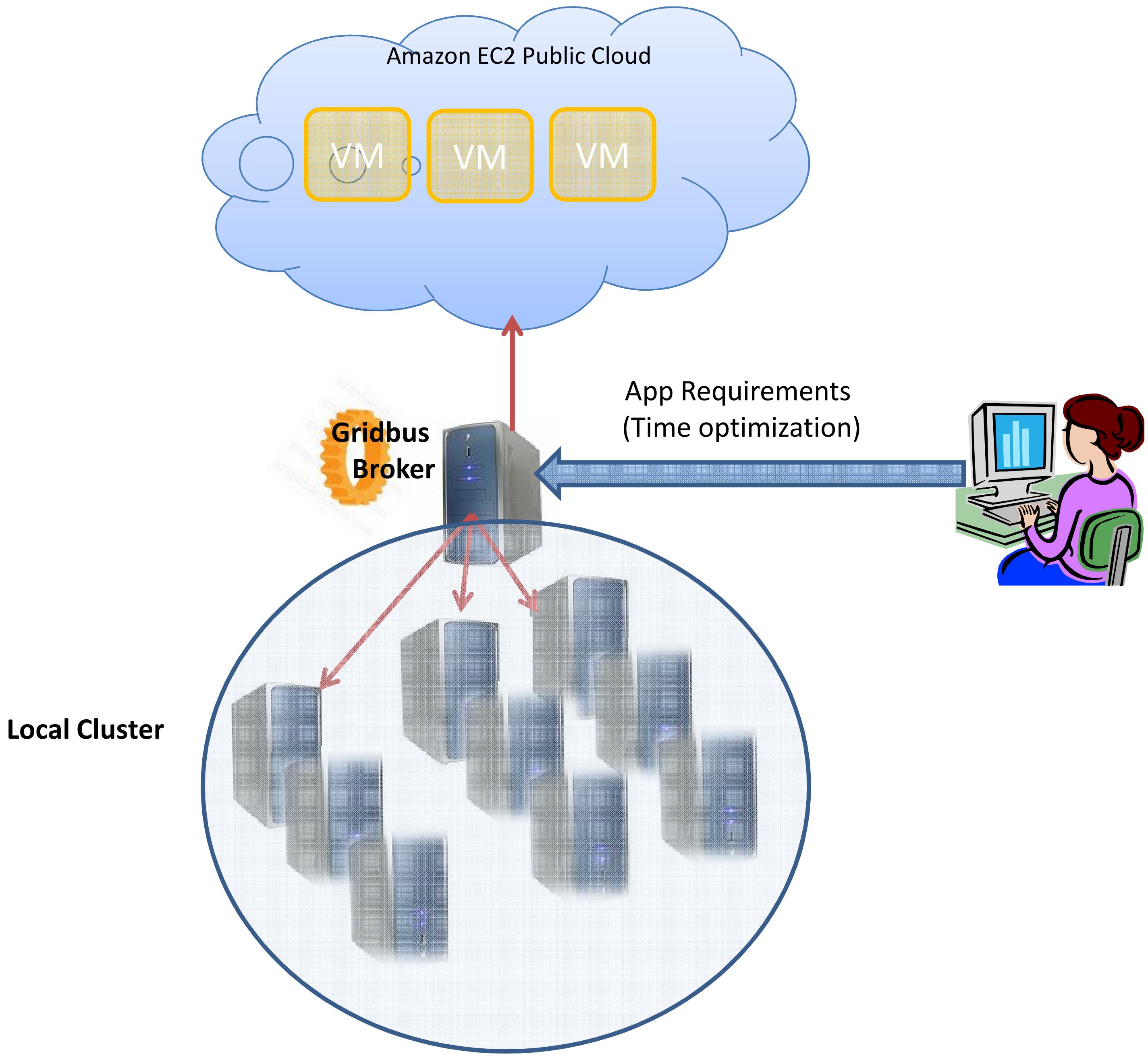
- In High Performance Computing (HPC), providing adequate resources for user applications is crucial.
- User-level brokers work on behalf of users and provide access to diverse resources with different interfaces.
- Recently, commercial Cloud providers(IaaS) offer computational power to users

IaaS Provider

- Provide resources in an on-demand manner
- charge users in a pay-as-you-go fashion
- An instance: Amazon EC2!
- In fact, there is a trade-o between spending budget to get resources from IaaS providers and running the application on local resources.

Problem Statement

- How scheduling policies inside the broker can benefit from resources supplied by the IaaS providers while:
  - User has limited deadline and budget.
  - There are some (but not enough) local resources available for free!
  - End user does not have any knowledge about the application execution time.
  - user has time minimization or cost minimization preference


Amazon EC2 Public Cloud
VM
VM
VM
App Requirements
(Time optimization)
Gridbus
Broker
Local Cluster

DBC (Deadline Budget Constraint) scheduling

- Time Optimization:
  - minimizing time, within time and budget constraints.
- Cost Optimization:
  - minimizing cost, within time and budget constraints

Related Work

- Deploying Cloud resources at resource provisioning level
- Deploying Cloud resources at broker (user) level

Related Work…

Proposed Policy	Use Non-Cloud Resources	User Constrains	User Transparency	Scheduling Level	Goal
Open Nebula	Local	No	Yes	System-level	Handling peak load
Llorenete et al.	Local and grid (Globus enabled)	No	Yes	System-level	Provision extra resources to handle peak load
Assuncao et al.	local	Yes (Budget)	Yes	System-level	Handling peak load
Vazquez et al	Local and grid (Globus enabled)	No	No	User-level	Federating several providers from Grid and Cloud
Silva et al	No	Yes (Budget)	No	User-level	Run Bag-Of-Task application on Cloud
Time and Cost Optimization (this paper)	Local	Yes (Budget and Deadline)	Yes	User-level	Minimizing completion time and incurred cost within a deadline and budget

Proposed Policy

- Time Optimization Policy
- Cost Optimization Policy

Time Optimization Policy

– Input: deadline, totalBudget,resourceCost

1. *budgetPerHour =totalBudget = (deadline - currentTime);*
2. *reqCounter =budgetPerHour = resourceCost;*
3. *RequestResource(reqCounter);*
4. *availResources + = AddAsServer();*
5. *DoAccounting();*
6. *while TaskRemained() = True do*
   1. *SubmitTask(availResources);*
7. *Terminate(reqCounter);*

Cost Optimization Policy

– *Input: deadline, totalBudget,resourceCost*

1. *SetAvailBudget(totalBudget);*
2. *while TaskRemained() = True do*
    1. *if availResources > 0 then*
        1. *SubmitTask(availResources);*
        2. *estimation = EstimateCompletionTime();*

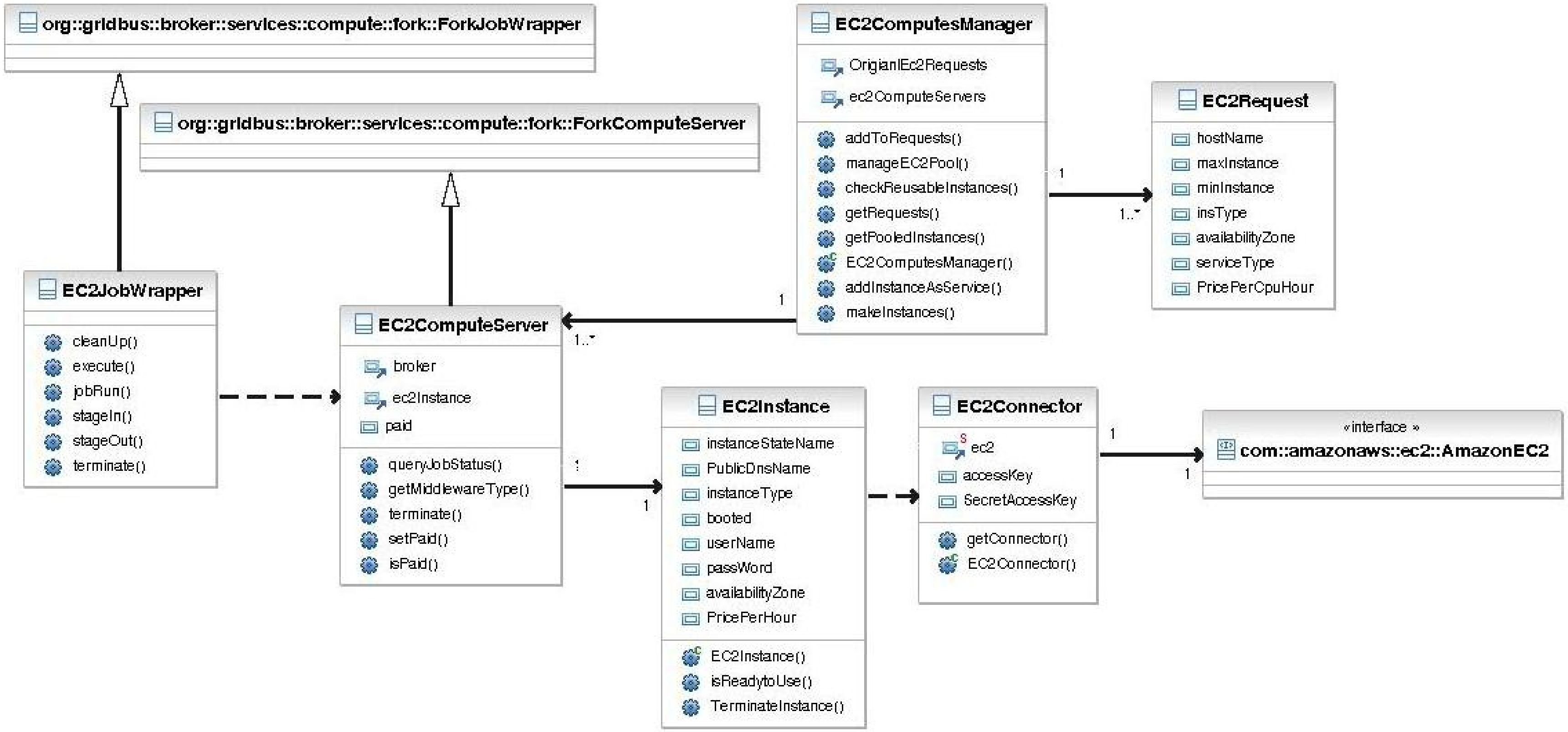
    else

        1. *estimation = deadline + 1;*
    1. *if estimation > deadline then*
        1. *availBudget = GetAvailBudget();*
        2. *if availBudget >= resourceCost then*
        3. *RequestResource(1);*
        4. *availResources + = AddAsServer();*
        5. *DoAccounting();*

    else

    1. *if estimation < (deadline) then*
        1. *Terminate(1);*

System Implementation



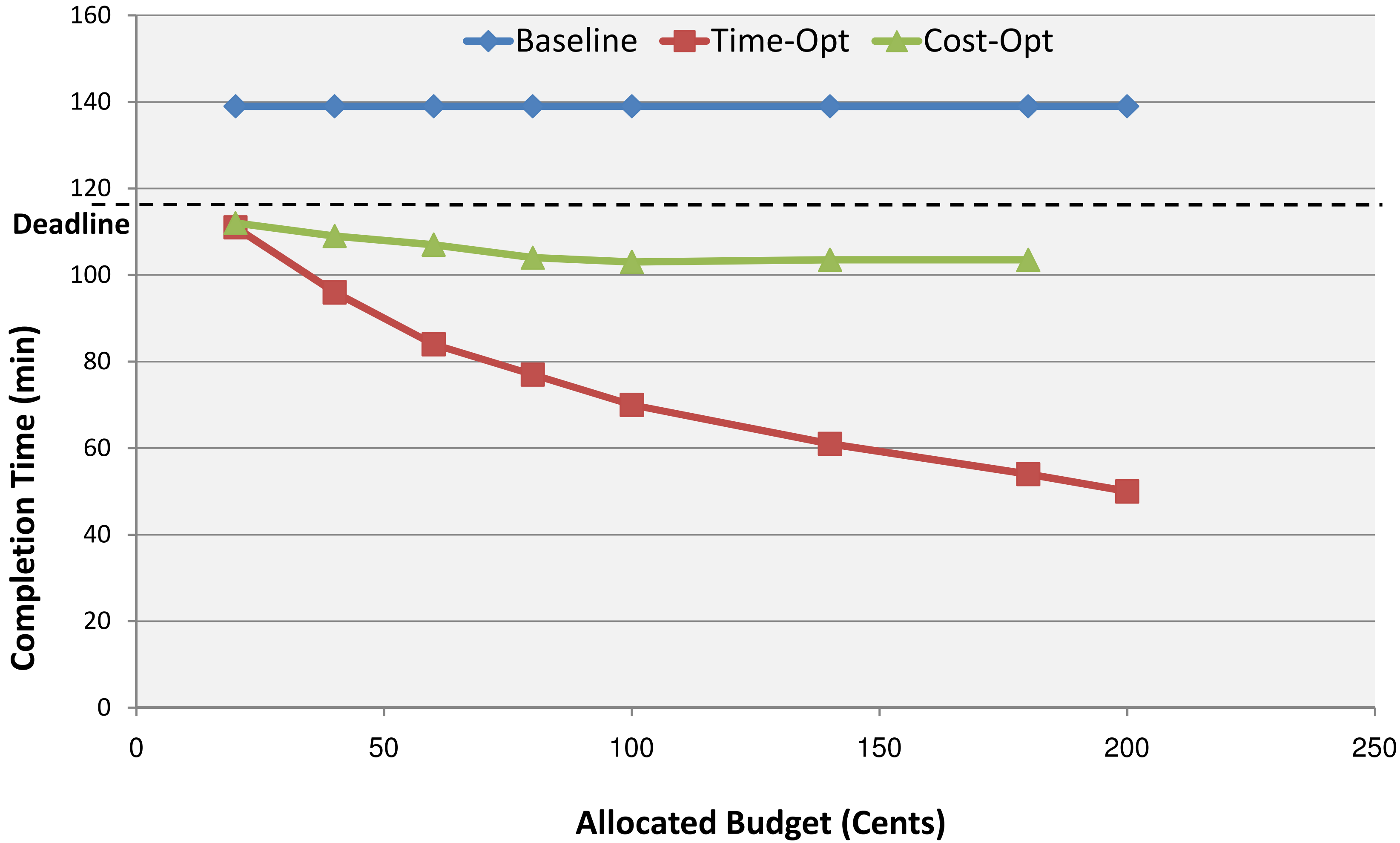
Experiment Results

- Experiment Setup
  - Cluster (Snowball.cs.gsu.edu) as the local resource
    - The cluster has 8 Pentium III (XEON 1.9 GHZ) CPU, 1GB RAM,
    - Linux operating system
  - Amazon EC2 as the IaaS provider
- Parameter Sweep Application (PSA)
  - Pov-Ray [7]
    - We configured Pov-Ray to render images with the same size.
    - Therefore, we ensure that the execution time is almost the same for all the tasks.

Exp#1. Impact of Budget Spent on Completion Time

- render 144 images
- two hours (120 minutes) as the deadline.
- no cost is assigned to the cluster

Exp#1. Impact of Budget Spent on Completion Time

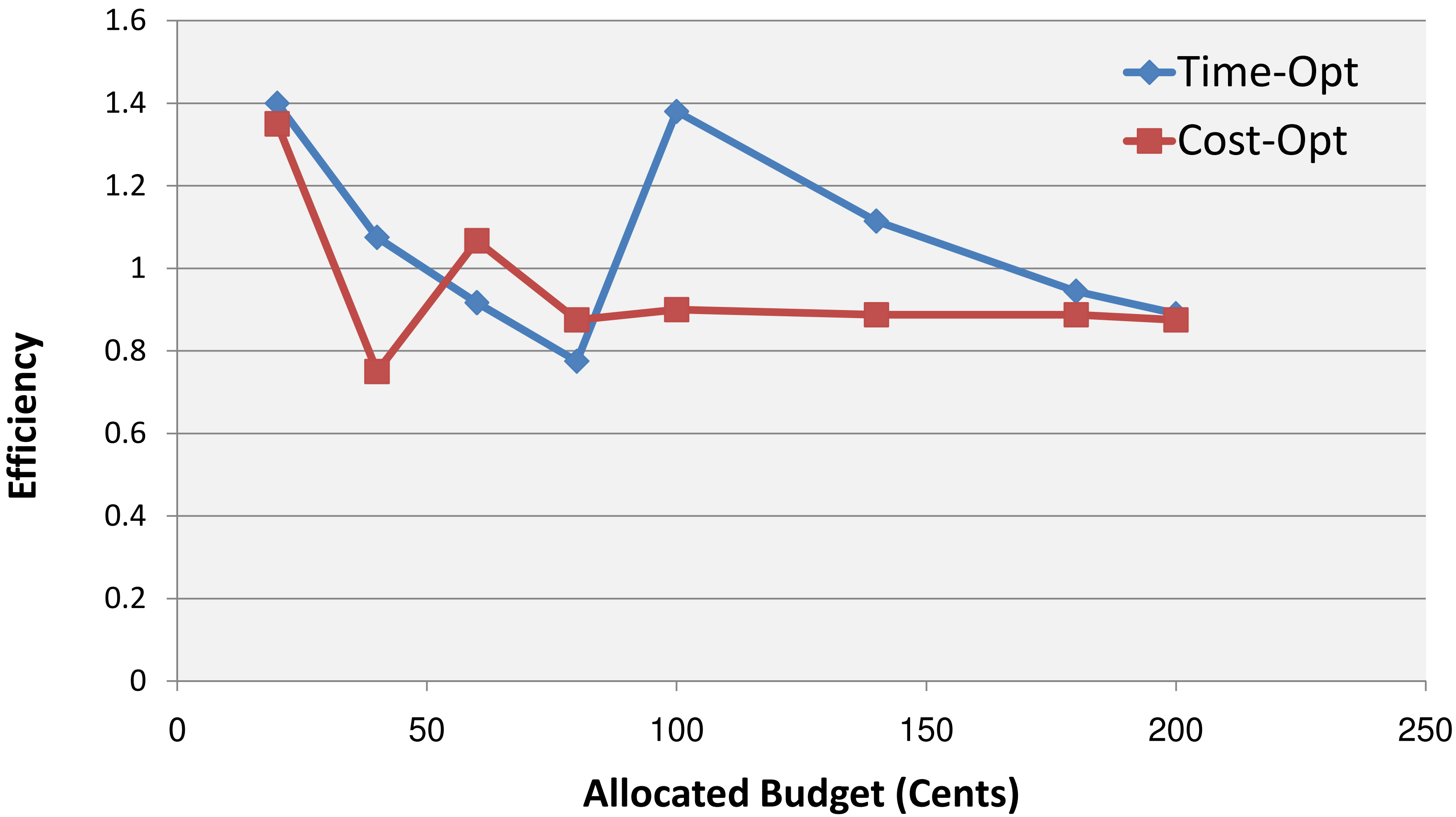


Exp#2. Efficiency of the Time Optimization and Cost Optimization Policy

$$efficiency = (C_l - C_{lc})/S$$

- C_l:
  - Completion time by deploying local resources.
- C_{lc}:
  - Completion time by using both local and cloud resources
- S:
  - Spent budget

Exp#2. Efficiency of the Time Optimization and Cost Optimization Policy

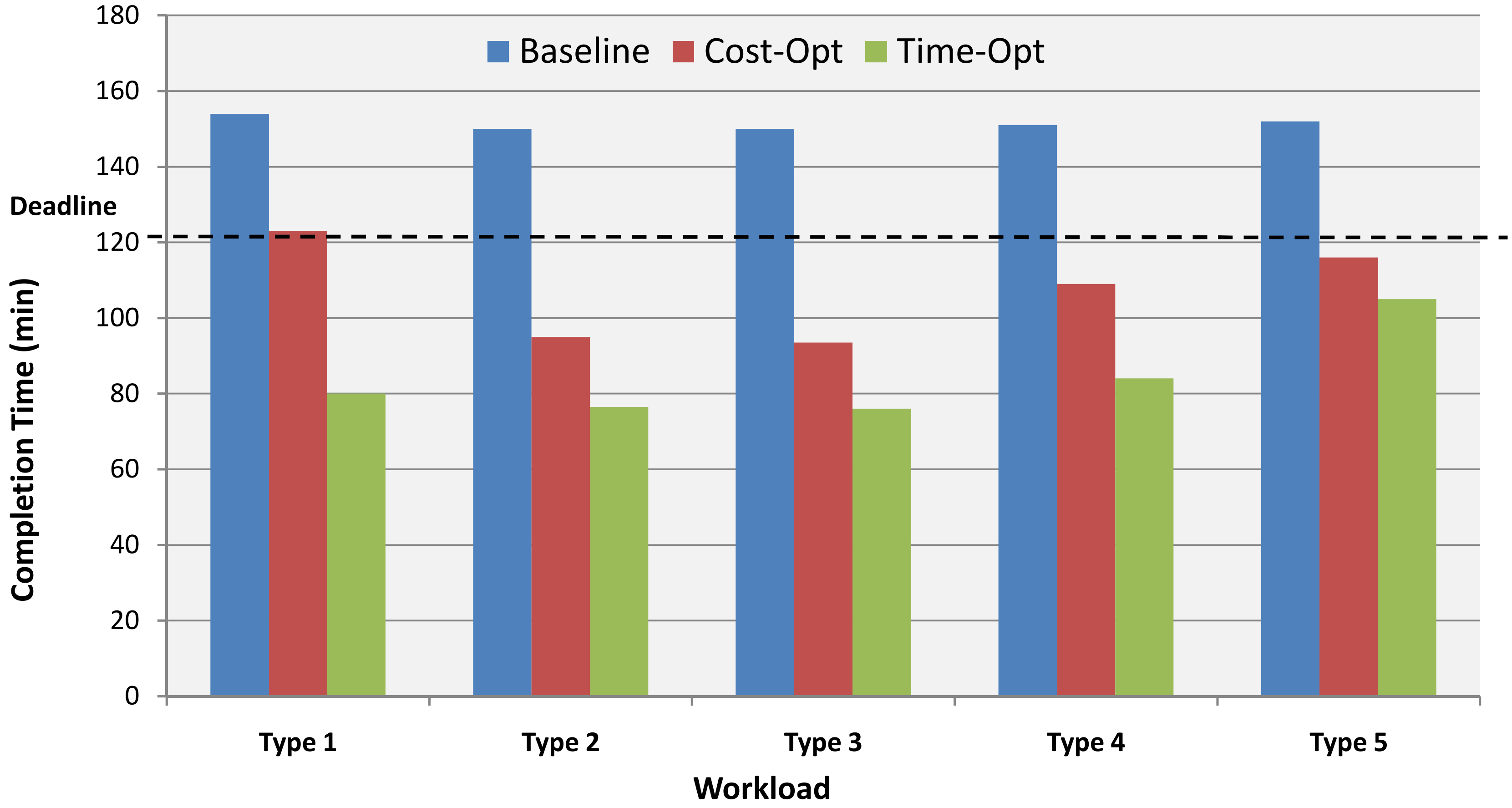


Exp#3. Impact of the Time Optimization and Cost Optimization Policies on the Completion Time of Different Workload Types.

Different workload types used in the experiment

Workload	No of Tasks	Task Time (minutes)
Type 1	32	38
Type 2	64	18.75
Type 3	128	9.37
Type 4	256	4.65
Type 5	512	2.34

Exp#3. Impact of the Time Optimization and Cost Optimization Policies on the Completion Time of Different Workload Types.



Conclusion and Future Work

- In this paper, two market-oriented scheduling policies are proposed to increase the computational capacity of the local resources by hiring resources from an IaaS provider.
- Time Optimization policy, completion time reduces almost linearly by increasing the budget.
- In the Cost Optimization the completion time does not improve after a certain budget (100 cents in our experiments).
- Efficiency of the Time Optimization and Cost Optimization policies can potentially increase by increasing the budget.
- We observed that different workload types can get completed before the deadline and within the budget using the proposed policies.

- **Questions?**