Z6110X0035:
Introduction to Cloud Computing
- Economics of IaaS Cloud Providers

Lecturer: Prof. Zichen Xu

Acknowledge Dr. Adel Nadjaran Toosi and congratulations on his Doctorate Defense

Outline

- Background
- Motivations, Research Problem and Thesis Focus
- Solutions
  - Federation of Clouds
    - Resource Provisioning Policies to Increase Profit
    - Financial Option Market Model
  - Single Cloud Provider
    - Revenue Management with Optimal Capacity Control
    - An Auction Mechanism for a Cloud Spot Market
    - Spot instance pricing as a Service
- Conclusions and Future Directions

Outline

- Background
- Motivations, Research Problem and Thesis Focus
- Solutions
  - Federation of Clouds
    - Resource Provisioning Policies to Increase Profit
    - Financial Option Market Model
  - Single Cloud Provider
    - Revenue Management with Optimal Capacity Control
    - An Auction Mechanism for a Cloud Spot Market
    - Spot instance pricing as a Service
- Conclusions and Future Directions

Cloud Computing

Allowing businesses to outsource their IT facilities to cloud providers
Avoid expensive up-front investments of establishing their own infrastructure
Long-held dream of computing as a utility
On-demand delivery of IT services
Customers pay for what they use
Virtualized resources

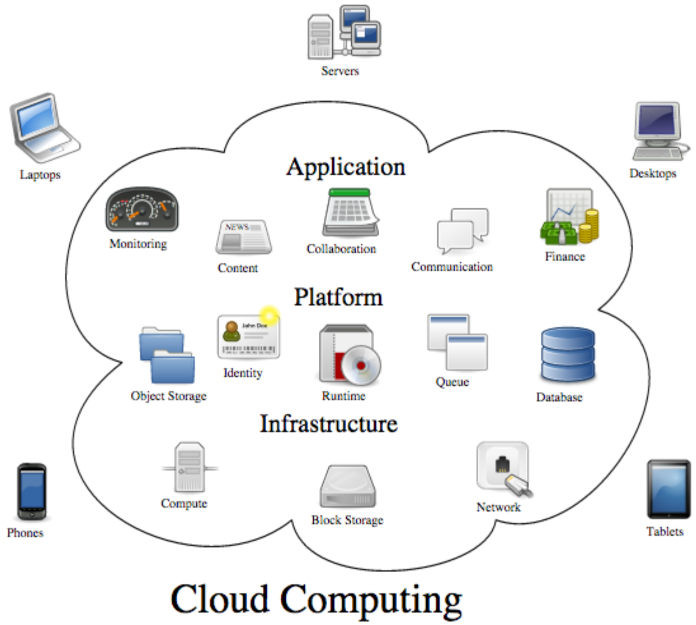


Figure: http://en.wikipedia.org/wiki/Cloud_computing

Types of Service Models

Software as a Service (SaaS)

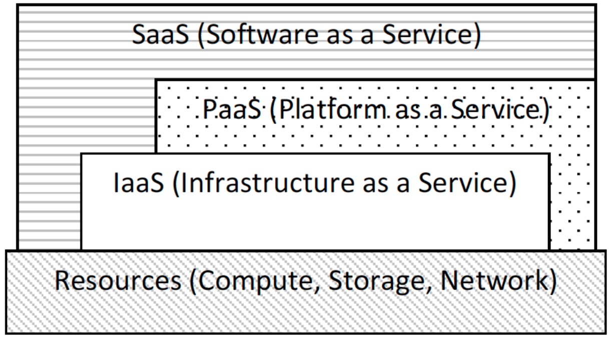
- It provides applications and software to the customer in utility-based model
- Accessible from a thin client interface such as a Web browser
- Example: Salesforce.com
  - Customer relationship management (CRM) as a service.

Platform as a Service (PaaS)

- It provides programming languages and tools to
  deploy application onto the cloud infrastructure
- Example: Google App Engine
  - facilities to build an reliable and scalable application

Infrastructure as a Service (IaaS)

- It provides capabilities for the customers to provision computational resources such as processing, storage, network, and other fundamental computing resources
- Virtual Machines (VMs)
- Example: Amazon EC2/S3

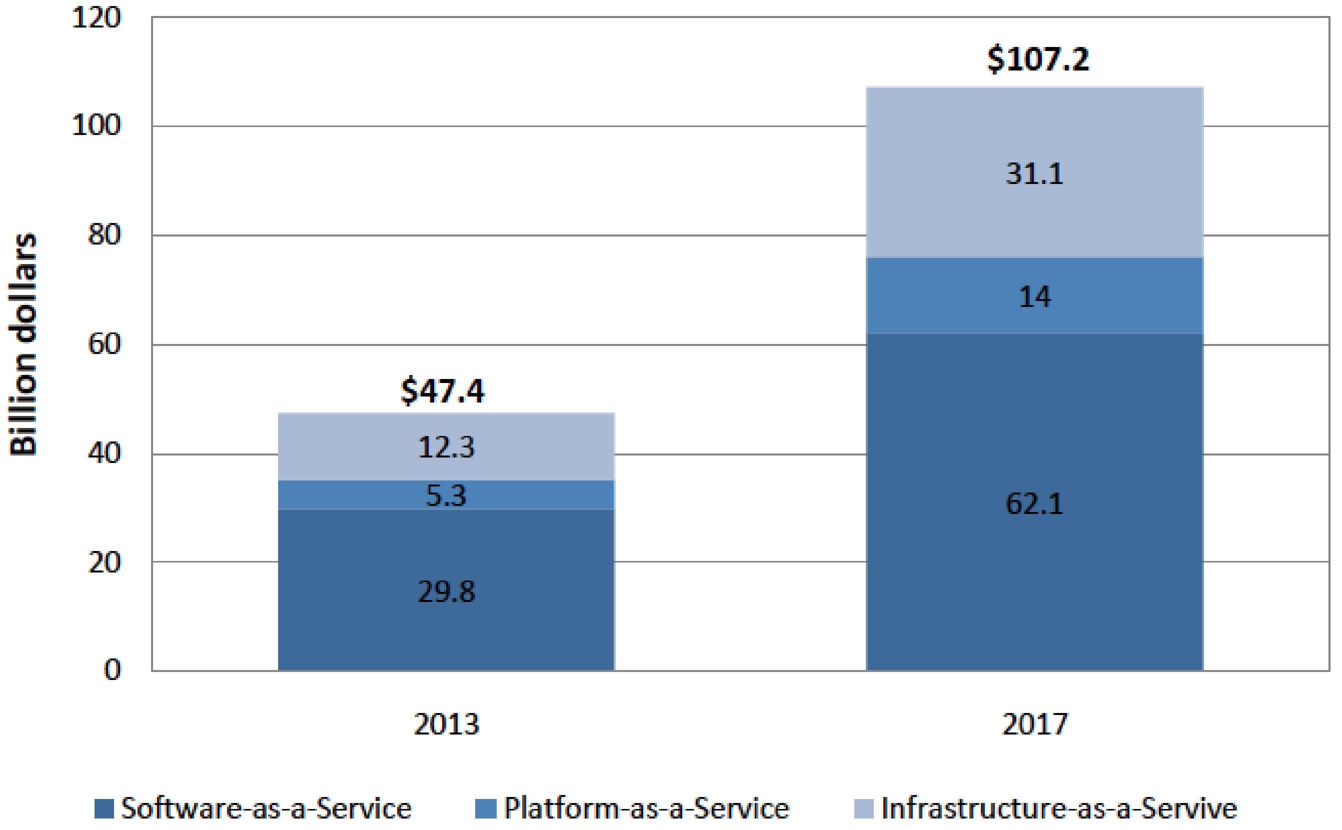


Cloud Service Stack

Outline

- Background
- Motivations, Research Problem and Thesis Focus
- Solutions
  - Federation of Clouds
    - Resource Provisioning Policies to Increase Profit
    - Financial Option Market Model
  - Single Cloud Provider
    - Revenue Management with Optimal Capacity Control
    - An Auction Mechanism for a Cloud Spot Market
    - Spot instance pricing as a Service
- Conclusions and Future Directions

International Data Corporation (IDC) forecast



Worldwide public IT cloud services spending in billion dollars by service model

Motivation (1)

The primary focus of designers in traditional distributed system:

- Improving system performance in terms of response time, throughput system, and utilization;
- Extensively studied in Cluster and Grid Computing in the past 15 years.

Cloud Computing and Computational Economy

- New objective functions: price, cost, and budget

☑Money

This has shifted a traditional distributed system into a two-party computation:

- Cloud providers and Cloud customers
- Pricing and cost as the bridge

Motivation (2)

There is a large body of research devoted to

- Minimizing cost
- Cloud customers

Relatively less work has been done

- Maximizing profit
- The provider's side

Techniques and mechanisms that help providers to maximize their profit while still selling their services competitively.

Research Problem

This thesis tackles the research challenges arising from the following topic:

“ *Designing market and economics-inspired algorithms and resource allocation mechanisms to maximize IaaS cloud providers' profit honoring the QoS constraints associated w SLA.*”

Focus and Solution Classification

We focus on the

- profit maximization problem of IaaS cloud providers.

We develop techniques and mechanisms for two main different situations:

- When a provider acts solely using their in-house resources to serve customers
- When it participates in a cloud federation and benefits from outsourcing requests

Background (Pricing)

Pricing definition

The process of determining the rate or fee the provider will receive in exchange for offering services or selling reso

Pricing Strategies

Marginal-cost pricing, Market-oriented pricing, etc.

Pricing Factors

Cost of Service

Market Competition

Value to the Customers

Pricing Models

Usage-based (Pay-as-you-go or consumption-based)

Subscription-based (reservation contract or prepaid scheme)

Demand-oriented

Background

Non-storable or perishable commodity

- Cloud services similar to many other services, are perishable in nature, and cannot be stored for future

IaaS providers offer computational services

- CPU cycles, network bandwidth, and memory space
- Virtual Machine (VM) instances
- VMs are non-storable (perishable)
  - if not utilized at a specific point in time
  - no value and waste associated underlying data center capacity

To maximize revenue, IaaS provider adopts differentiated pricing pla with different Quality of Service (QoS)

- Usage-based, subscription-based and demand-oriented

Background (Common Pricing Models)

On-demand pay-as-you-go instances (usage-based)

- Customers pays for instances based on actual usage
- No long-term commitments
- Billing cycle (e.g., hourly)
- Fixed-rate basis

Reserved instances (subscription-based)

- Also known as reservation plan
- Reservation fee (premium)
- After which the usage is either free (e.g., GoGrid) or heavily discounted (e.g., Amazon EC2)
- Guaranteed availability.

Spot instances (demand-oriented)

- Also known as spot market
- Customers submits their bids
- Providers reports a market-wide clearing price
- Terminated by the provider if spot market’s clearing price rises above the customer’s bid

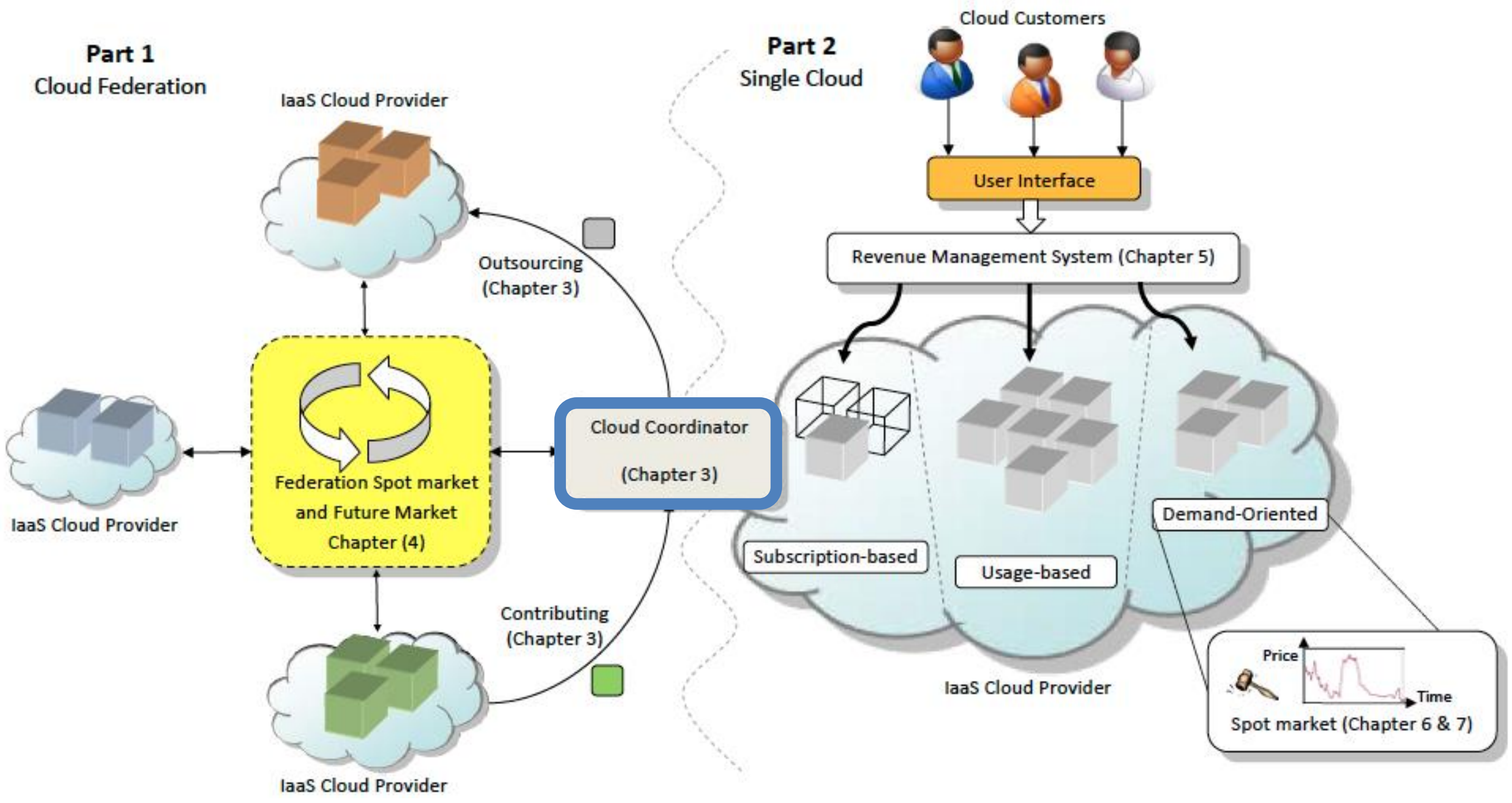
Background (Cloud Federation)

- Resources available in a single data center are limited
  - A large demand may put pressure in the data center capacity
  - One possible source for additional resources is idling resources from other providers
- Cloud Federation is a collection of individual Cloud providers, which collaborate by trading resources (e.g. computing, storage)

Outline

- Background
- Motivations, Research Problem and Thesis Focus
  - Background
    - Pricing and Dynamic Pricing
    - Cloud Federation
- Solutions
  - Federation of Clouds
    - Resource Provisioning Policies to Increase Profit
    - Financial Option Market Model
  - Single Cloud Provider
    - Revenue Management with Optimal Capacity Control
    - An Auction Mechanism for a Cloud Spot Market
    - Spot instance pricing as a Service
- Conclusions and Future Directions

Solutions



Part 1
Cloud Federation
IaaS Cloud Provider
Outsourcing
(Chapter 3)
IaaS Cloud Provider
Federation Spot market
and Future Market
Chapter (4)
Cloud Coordinator
(Chapter 3)
Contributing
(Chapter 3)
IaaS Cloud Provider
Part 2
Single Cloud
Cloud Customers
User Interface
Revenue Management System (Chapter 5)
Subscription-based
Usage-based
Demand-Oriented
IaaS Cloud Provider
Price
Time
Spot market (Chapter 6 & 7)

Resource Provisioning Policies to Increase Profit

Derived from publication:

Adel Nadjaran Toosi, Rodrigo N. Calheiros, Ruppa K. Thulasiram, and Rajkumar Buyya, "**R**
provisioning policies to increase IaaS provider's profit in a federated cloud environm
Proceedings of the 13th IEEE International Conference on *High Performance Comput*
Communications (HPCC'11), Banff, Canada, Sep. 2011, pp. 279–287.

Resource Provisioning Policies in Cloud

Current resource management strategies hinder providers market potential by *limiting the amount of resources* allocated to requests

- QoS is met in a conservative way
- Over-provisioning of datacenter capacity
  - average demand of the system is several times smaller than the peak demand

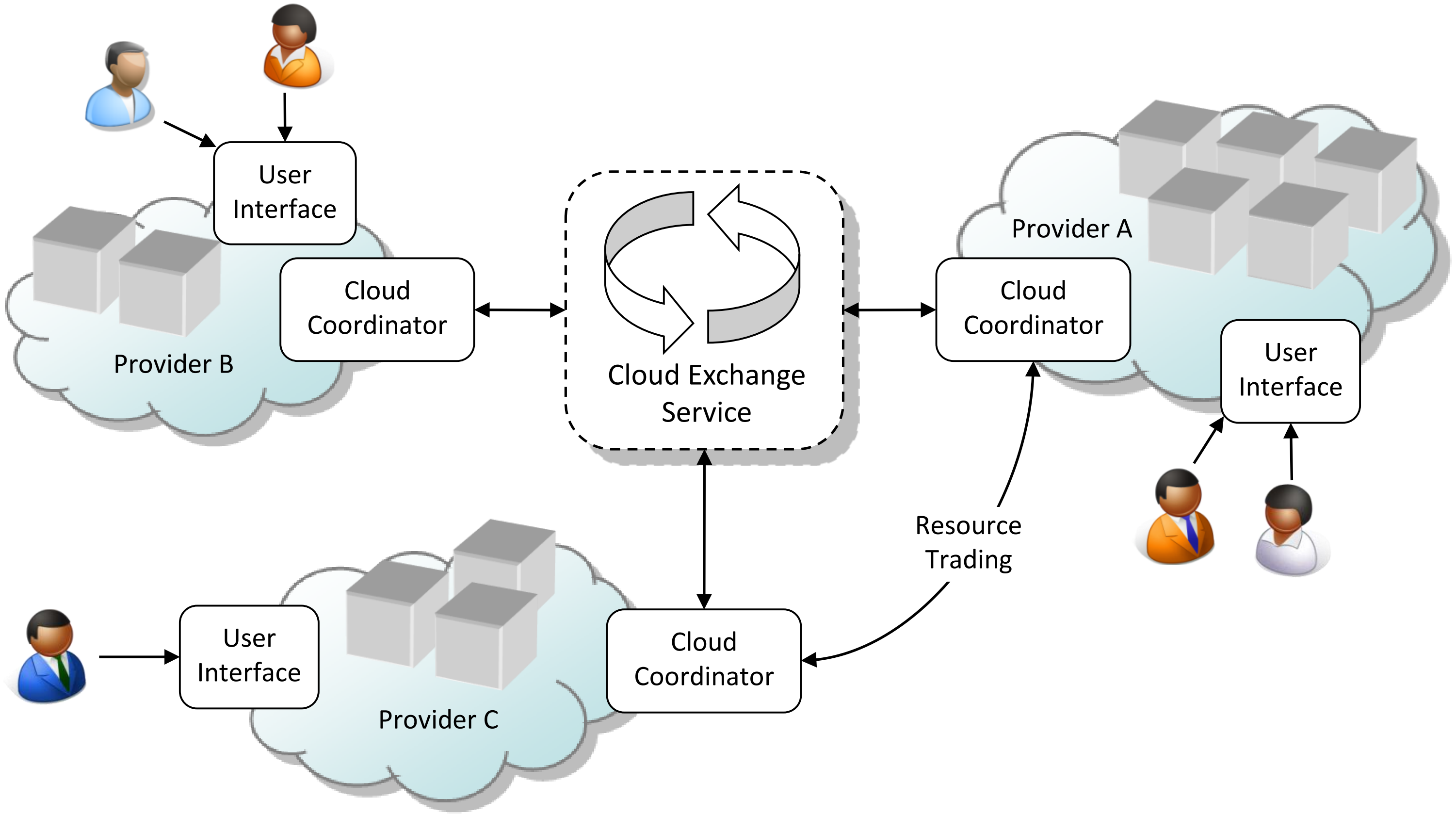
By exploiting cloud federation potentials, providers are able to *dynamically increase the available resources* to serve requests

Outsourcing requests or *renting part of idling resources* to other providers is not a trivial issue

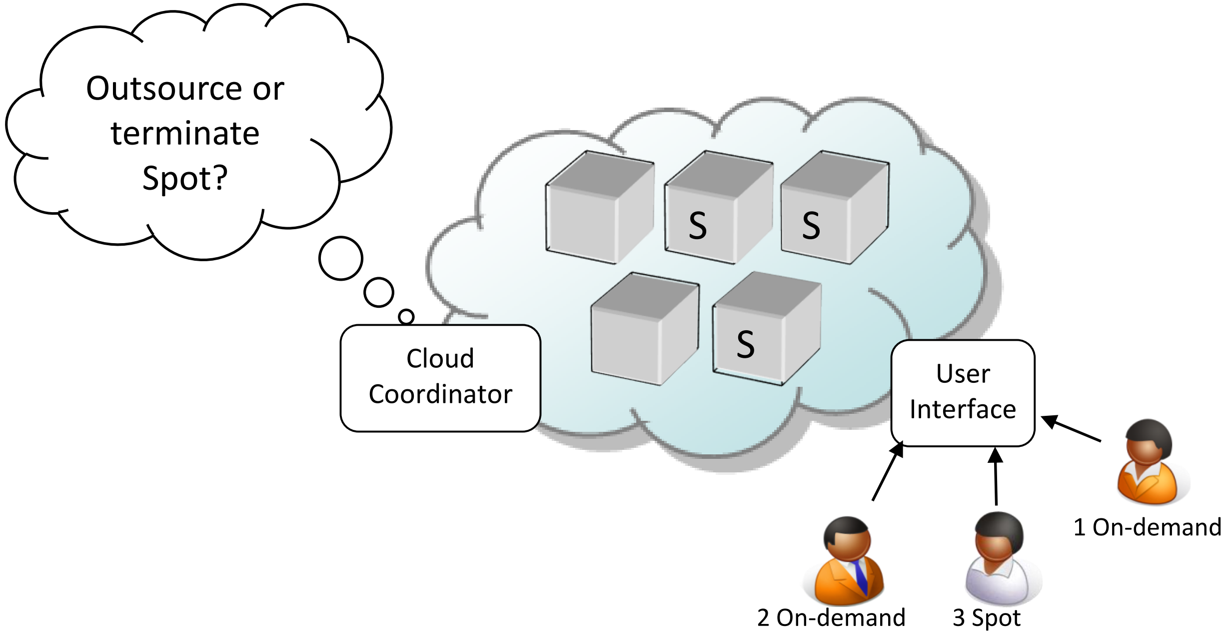
Research Objectives

Study the *impact of federation* as a mechanism for maximizing cloud provider's profit, utilization and reputation
Leverage federation potentials by creating *resource provisioning mechanisms* for Cloud providers
Investigate the effect of applying different *resource trading paradigms* to facilitate federation of providers

System Model



Outsourcing decision scenario



Resource provisioning policies for federated-aware providers in the presence of terminable local VMs

Policies

Non Federated Totally In-house (NFTI)

Termination of spot VMs with lowest bid

If action does not release enough resources for the new on demand request the request will be rejected

Federation-Aware Outsourcing Oriented (FAOO)

Fully utilized provider firstly checks the Cloud exchange service for available resources by other members

It outsources the request to the provider that offers the cheapest price

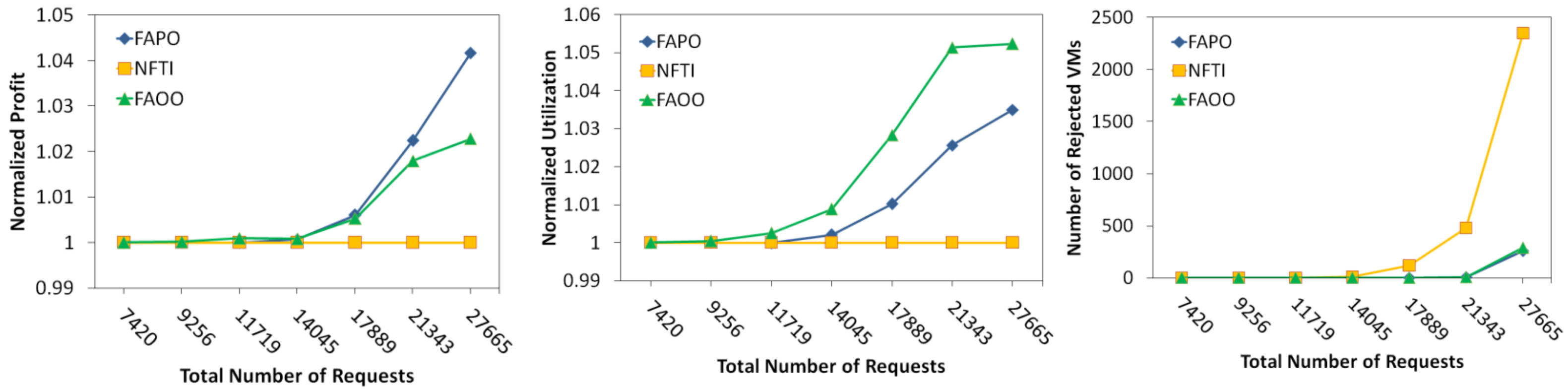
Federation-Aware Profit Oriented (FAPO)

Based on analytical analysis of instant profit, it decides between outsourcing and termination of spot VMs

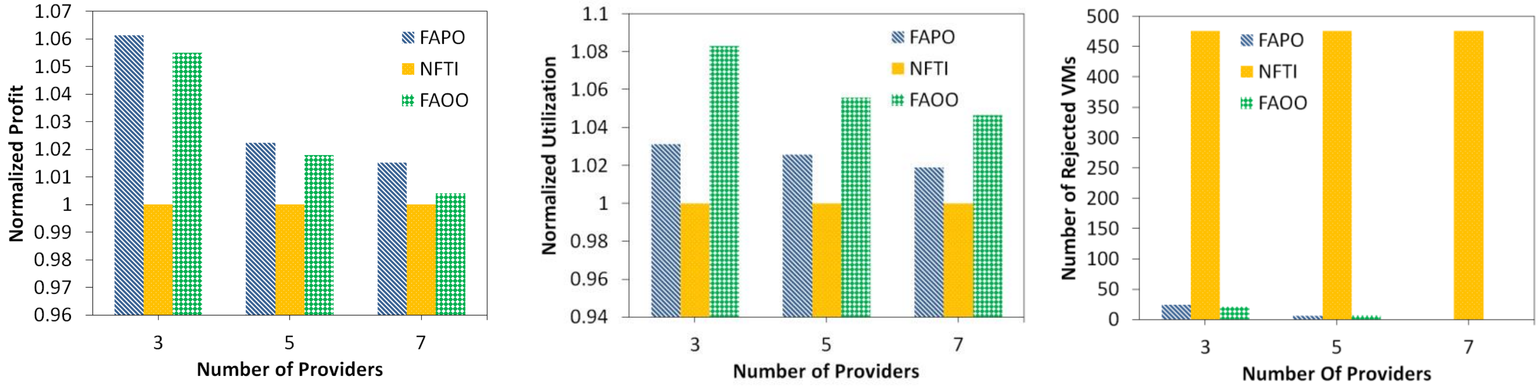
Performance Evaluation - Setup

- Simulation study with CloudSim
  - CloudSim: A discrete simulator for modelling and simulation of Cloud Computing
- The VM configuration is inspired by Amazon EC2 instances
  - One VM type (Small Instances:1 CPU core, 1.7 GB RAM, 1 EC2 Compute Unit, and 160 GB of local storage)
- Each datacenter 128 servers, and each one supports 8 VMs.
- Due to lack of publicly available workload models and real traces of IaaS Clouds, Lublin workload model (one week long simulation)
  - Each experiment is carried out 20 times
  - Average of the results is reported.
- Bidding Algorithm:
  - A uniformly-distributed random value between the minimum of bid \$0.020 and maximum of \$0.085(on-demand price)
  - The minimum price is set in such a way that the value offered by customers is still enough to cover operational costs of serving the request

Results

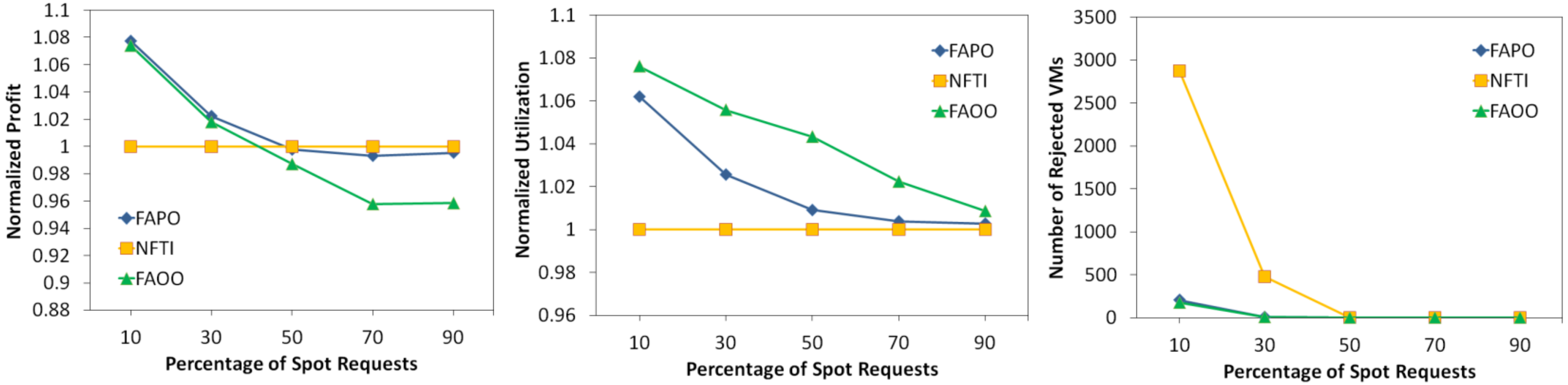


Impact of load on (a) Profit (b) Utilization (c) Number of rejected on-demand VMs, for a provider with different policies.

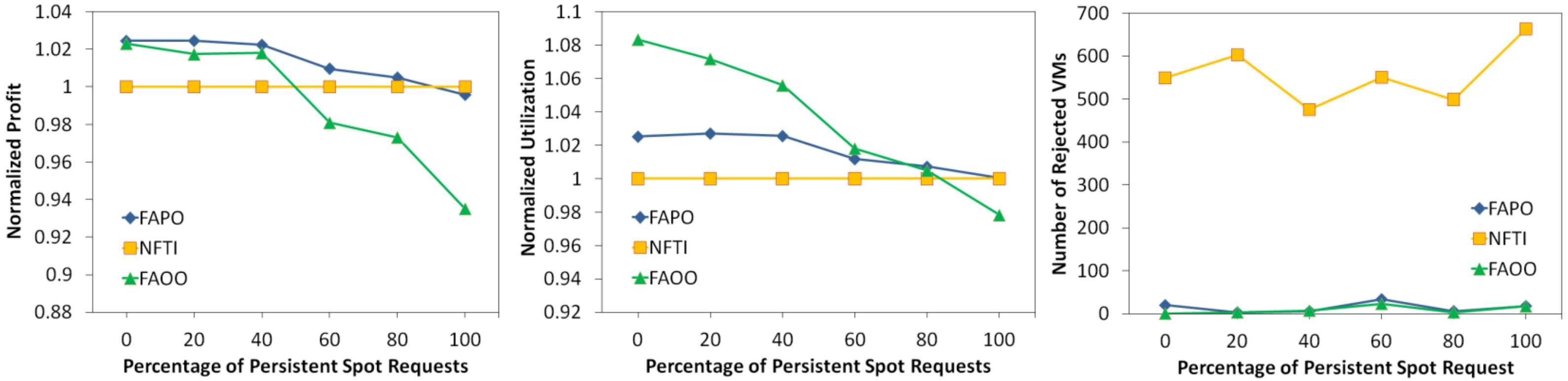


Impact of number of providers on (a) Profit (b) Utilization (c) Number of rejected on-demand VMs for a provider with different policies.

Results(cont.)

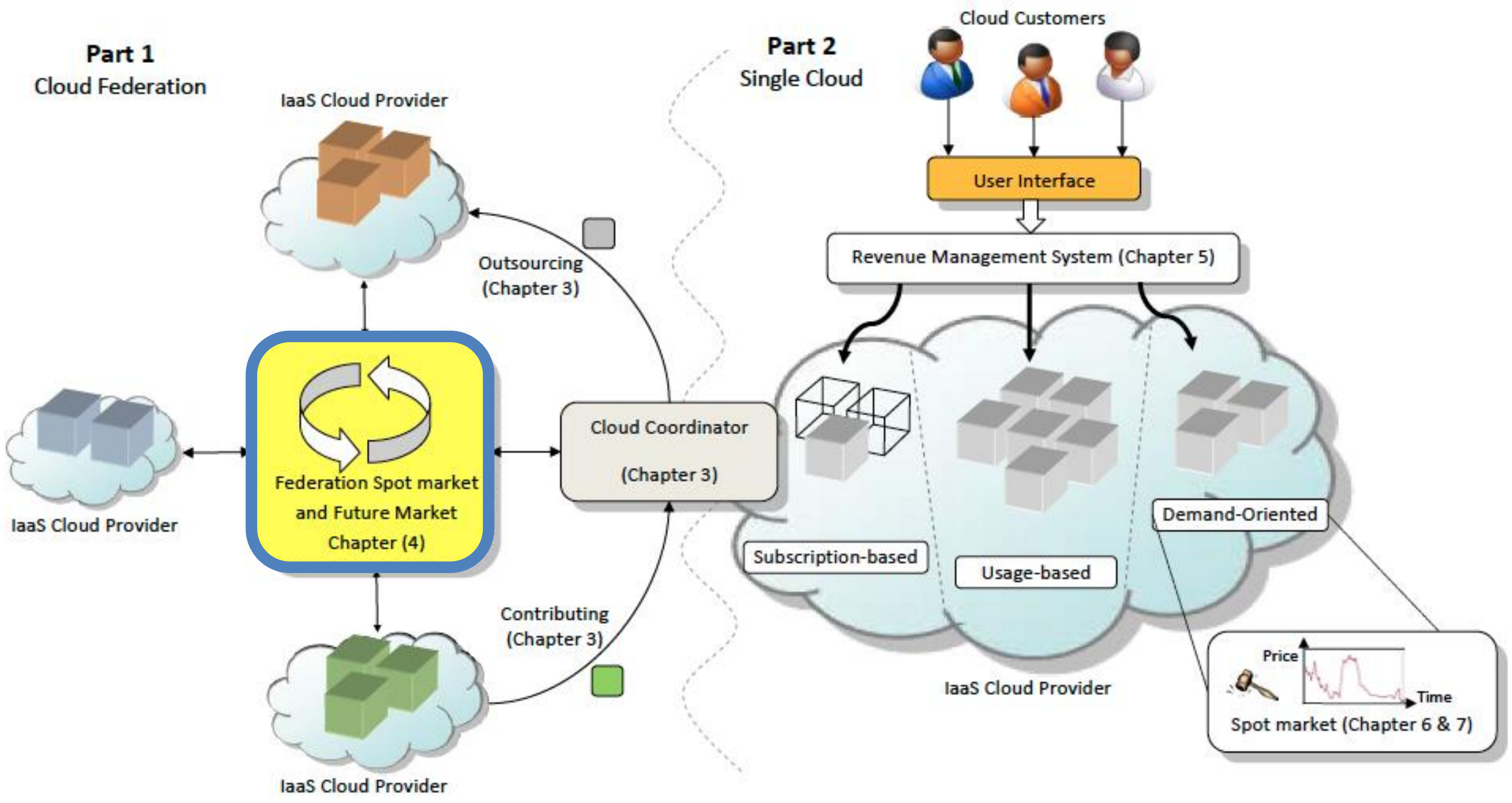


Impact of percentage of spot requests on (a) Profit (b) Utilization (c) Number of rejected on-demand VMs, for a provider with different policies.



Impact of percentage of persistent spot requests on (a) Profit (b) Utilization (c) Number of rejected on-demand VMs for a provider with different policies.

Solutions


Part 1
Cloud Federation
IaaS Cloud Provider
Outsourcing
(Chapter 3)
IaaS Cloud Provider
Federation Spot market
and Future Market
Chapter (4)
Cloud Coordinator
(Chapter 3)
Contributing
(Chapter 3)
IaaS Cloud Provider
Part 2
Single Cloud
Cloud Customers
User Interface
Revenue Management System (Chapter 5)
Subscription-based
Usage-based
Demand-Oriented
IaaS Cloud Provider
Price
Time
Spot market (Chapter 6 & 7)

Financial Option Market Model

Derived from publication:

Adel Nadjaran Toosi, Ruppa K. Thulasiram, and Rajkumar Buyya, “**Financial option market m**
federated cloud environments,” in Proceedings of the 5th IEEE/ACM International Confer
Utility and Cloud Computing (UCC’12), Chicago, Illinois, USA, Nov. 2012, pp. 3–12.

Reserved capacity is not fully utilized!

Huge and unpredictable variation in the load over time for Cloud applications, e.g. we
applications.

Economic advantage of using Reserved Instances in comparison with On-Demand Instances, even if they are not fully utilize

- Amazon EC2: Using Heavy Utilization Reserved Instances, you can ***save up*** to 54% for a 1-year term and 71% for a 3-year term vs. ru Demand Instances.
- Reserved Instance Marketplace

All together :

The pattern of utilization at user side causes reserved instances not
deployed at all times

Using Reserved Capacity for On-demand Requests

We are interested to explore the opportunity:

Additional cash flow by releasing underutilized capacity of the reserved instances to the on-demand request
If the unreserved part of the data center experiences high utilization, providers are able to accommod
demand requests on the underutilized reserved capacity of the data center.

Risk of SLA violation!
Cloud cooperation is a possible solution

Risks in Cloud Federation Market

Two more risks:

- a risk of being unable to acquire required resources in the market.
  - Short selling resources without having a good knowledge of usage loads and hence violating the Q
- Future price variations in the federation market using past price history

A financial option-based market model is introduced for federation of Cloud providers

which helps providers increase their profit and mitigate the risks (risks of violating QoS or paying extra

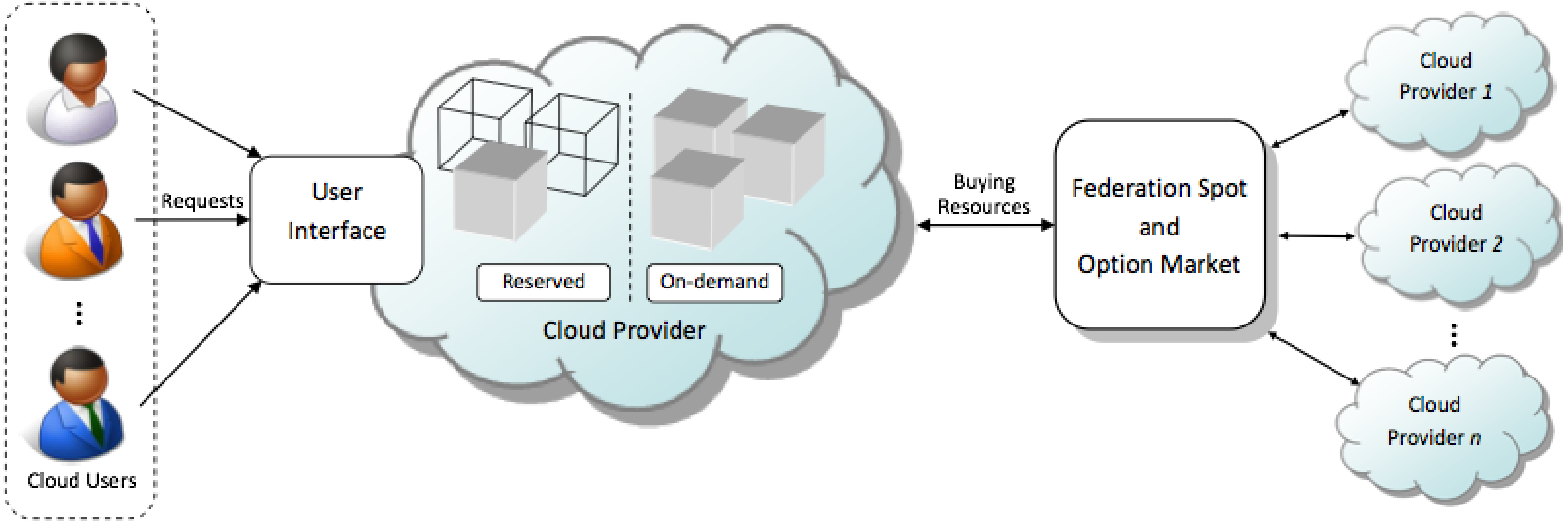
Financial Option Contract

A financial option is a contract for a future.
Transaction between two parties:

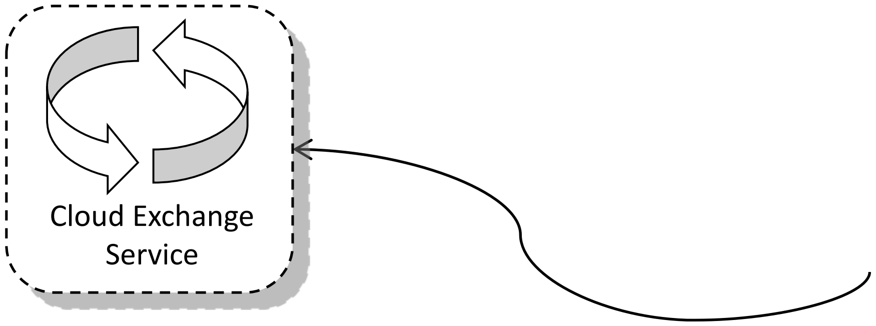
Holder and seller of the contract.

A financial option gives the holder the right, *but not the obligation*, to
(or to sell) an underlying asset at a certain price, called the *strike pri*
(exercise price), within a certain period of time, called *maturity date*
(expiration date).
The seller is obligated to fulfill the transaction.
As a compensation the seller collects an upfront payment at the begin
of the contract, called *premium*.

The System Model



When to buy option?



Policies

Baseline In-house Isolated Pool Policy (IIP)

- Provider works independently, without participating in the federation.
- Isolated pools of physical nodes for on-demand and reserved instances .

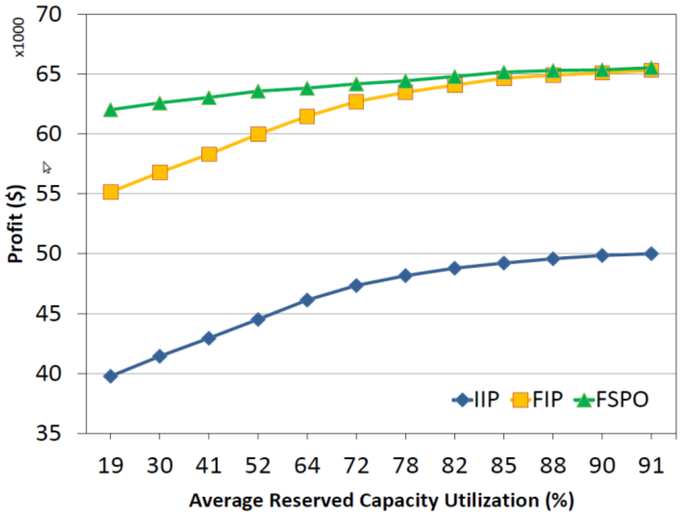
Baseline Federated Isolated Pool Policy (FIP)

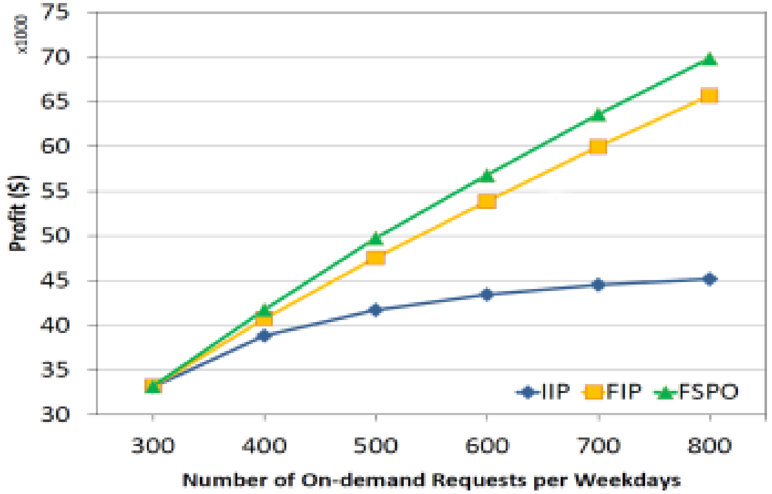
- If the provider is not able to serve on-demand requests locally it outsource them through federation spot market.
- To be always cost-effective, if the spot price in the federation market is higher than the local on-demand price then the provider rej on-demand request.
- The pool of physical nodes for reserved and on-demand instances are isolated to prevent rejection of reserved requests.

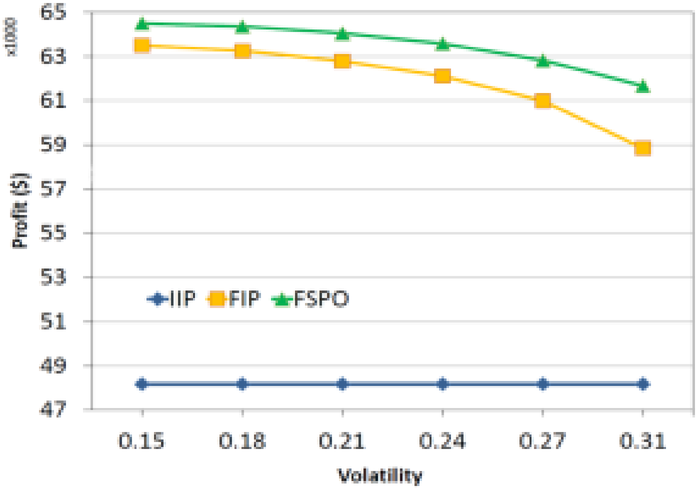
Federated Shared Pool Option-Enabled Policy (FSPO)

- The provider accommodates excess on-demand requests in the underutilized reserved capacity.
- The provider buys an option whenever he accommodates on-demand requests in the reserved capacity.
- If a reserved request comes in and the provider is not able to serve it locally, the option is exercised and the reserved request is out the strike price of the option.

Experimental Results







Number of Request Rejections

NUMBER OF ON-DEMAND (O), RESERVED (R), REJECTED ON-DEMAND (RO), REJECTED RESERVED (RR), OUTSOURCED ON-DEMAND (OO), AND OUTSOURCED RESERVED(OR) REQUESTS FOR THE PROVIDER WITH POLICIES.

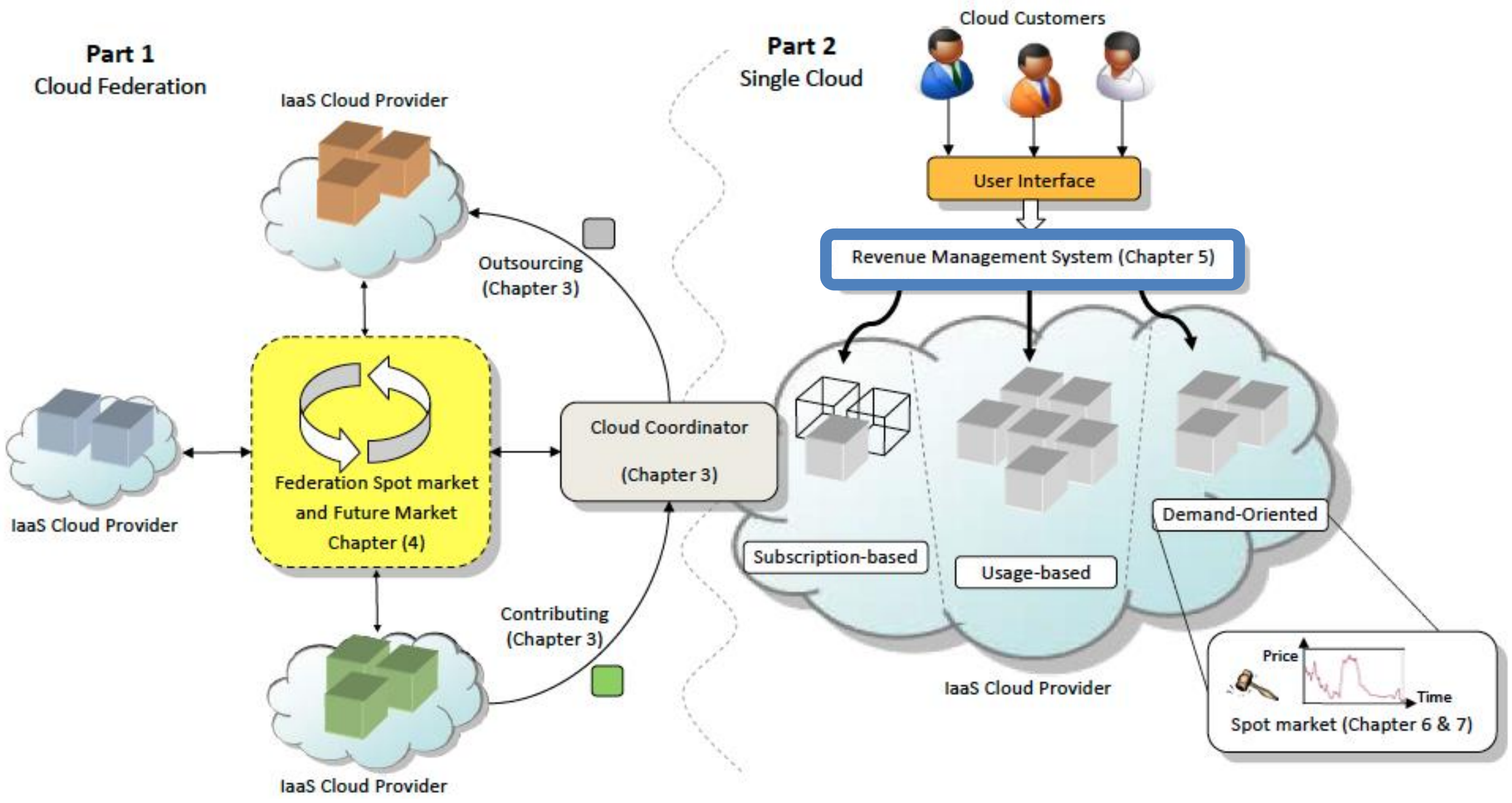
Policy	O	R	RO	RR	OO	OR
IIP	106340	36590	42450	**0**	0	0
In-house+ Shared Pool	106340	36590	19188	**6636**	0	0
FIP	106340	36590	219	**0**	42230	0
Federated+ Shared Pool	106340	36590	111	**38**	19078	6598
FSPO	106340	36590	111	**0**	19078	6636

Effect of Maturity Time

IMPACT OF OPTION MATURITY TIME ON PROVIDER'S PROFIT USING FSPO.

Maturity time (Day)	Bought Option	Profit ($)
7	3086	63588
10	2612	63588
30	1913	63583
60	1745	63578
90	1701	63574

Solutions



Part 1
Cloud Federation
IaaS Cloud Provider
Outsourcing
(Chapter 3)
IaaS Cloud Provider
Federation Spot market
and Future Market
Chapter (4)
Cloud Coordinator
(Chapter 3)
Contributing
(Chapter 3)
IaaS Cloud Provider
Part 2
Single Cloud
Cloud Customers
User Interface
Revenue Management System (Chapter 5)
Subscription-based
Usage-based
Demand-Oriented
IaaS Cloud Provider
Price
Time
Spot market (Chapter 6 & 7)

Revenue Management with Optimal Capacity Control

Derived from:

Adel Nadjaran Toosi, Kurt Vanmechelen, Ramamohanarao Kotagiri, and Rajkumar Buyya, "I
Maximization with Optimal Capacity Control in Infrastructure as a Service Cloud M
IEEE Transactions on Cloud Computing (TCC), 2014, in

IaaS providers' various pricing plans (or markets)

- on-demand pay-as-you-go
  - Generates highest revenue per unit capacity
  - Demand uncertainty
- Reservation (subscription)
  - Risk-free income from reservations
  - Guaranteed cash flow through long-term commitments
  - Compensate for the demand uncertainty of on-demand pay-as-you-go
  - Generates lower revenue per unit capacity
  - The provider is liable to offer guaranteed availability to honor SLA
- Spot market
  - Selling spare capacity in the data center
  - Attract price-sensitive users that are capable of tolerating service interruptions
  - Without being exposed to the risks resulting from overbooking capacity

Problem definition

The use of multiple pricing plans introduces:

Non-trivial trade-offs in revenue maximization

Our main **research question** is the following:

"with limited resources available, and considering the dynamic and stochastic nature of customers' demand, how can expected revenue be maximized through an optimal allocation of capacity to each pricing plan?"

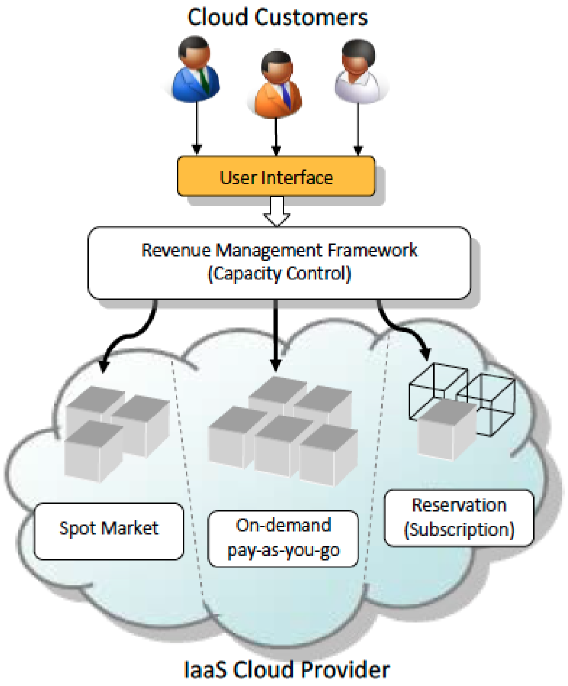
Assumptions:

Customers do not fully utilize their reserved capacity in the reservation's lifetime

We do not allow accommodating on-demand requests in underutilized reserved capacity

Creates the risk of SLA violations.

Provider accommodates spot instances in the underutilized reserved capacity of the data center



Our Solution

We frame our algorithmic contributions

- Within a revenue management framework
- Supports the three presented pricing plans
- Incorporates an admission control system for requests of the reservation plan.

We formulate the optimal capacity control problem

- As a finite horizon Markov Decision Process (MDP)
- A stochastic dynamic programming technique
  - To compute the maximum number of reservation contracts the provider must accept
  - For a large capacity provider the stochastic dynamic programming technique is computationally prohibitive.
- We therefore present two algorithms to increase the scalability of our solution

Contributions

Proposed Algorithms

- Optimal Algorithm
  - A stochastic dynamic programming technique
  - As a finite horizon Markov Decision Process (MDP).
- Pseudo Optimal Algorithm
  - Based on optimal algorithm
  - Only increases the spatial and temporal granularity of the problem
  - To solve it in a time suitable for practical online decision making
- Heuristic Algorithm
  - Sacrifices accuracy to an acceptable extent to increase scalability
  - Through a number of simplifying assumptions on
    - Reserved capacity utilization and
    - The lifetime of on-demand requests.

Pseudo Optimal

Algorithm 1 Pseudo Optimal Algorithm

Input: t, l_t^r, l_t^o, i_t
Output: $maxrev$
dp $\leftarrow \{-1\}$ ▷ matrix **dp** is used for memoization and all cells are initialized with -1.
function $V(t, l_t^r, l_t^o, i_t)$
  if $\mathbf{dp}[t][l_t^r][l_t^o][i_t] \neq -1$ **then**
    return $\mathbf{dp}[t][l_t^r][l_t^o][i_t]$
  end if
  if $t = \tau$ **then**
    $\mathbf{dp}[t][l_t^r][l_t^o][i_t] = 0$
    return 0
  end if
  $maxrev \leftarrow 0$
  for $r_t \leftarrow 0$ **to** $\min(C - l_t^r - l_t^o, d_t^r)$ **do**
    $rev \leftarrow 0$
    $l_{t+1}^r \leftarrow l_t^r + r_t - e_t^r$
    $o_t \leftarrow \min(C - l_t^r - l_t^o - r_t, d_t^o)$
    $s_t \leftarrow \min(C - (l_t^r + r_t)u_{i_t} - l_t^o - o_t, d_t^s)$
    $\lambda \leftarrow (\tau - t)/\tau$
    $\gamma(\varsigma_t, r_t) \leftarrow B\lambda r_t \varphi + BT(\alpha p(l_t^r + r_t)u_{i_t} + p(l_t^o + o_t) + \beta p s_t)$
    for $l_{t+1}^o \leftarrow 0$ **to** $l_t^o + o_t$ **do**
      for $i_{t+1} \leftarrow 0$ **to** $|U|$ **do**
        $P(\varsigma_{t+1}|\varsigma_t, r_t) \leftarrow Bin(l_{t+1}^o; l_t^o + o_t, q) \times P(u_t = u_{i_{t+1}})$
        $rev \leftarrow rev + \gamma(\varsigma_t, r_t) + P(\varsigma_{t+1}|\varsigma_t, r_t) \times V(t+1, l_{t+1}^r, l_{t+1}^o, i_{t+1})$
      end for
    end for
    if $rev \geq maxrev$ **then**
      $maxrev \leftarrow rev$
    end if
  end for
  $\mathbf{dp}[t][l_t^r][l_t^o][i_t] \leftarrow maxrev$
  return $maxrev$
end function

Details of *Optimal Algorithm*

Computational Complexity

$$O(\tau \times C^5 \times |U|^2)$$

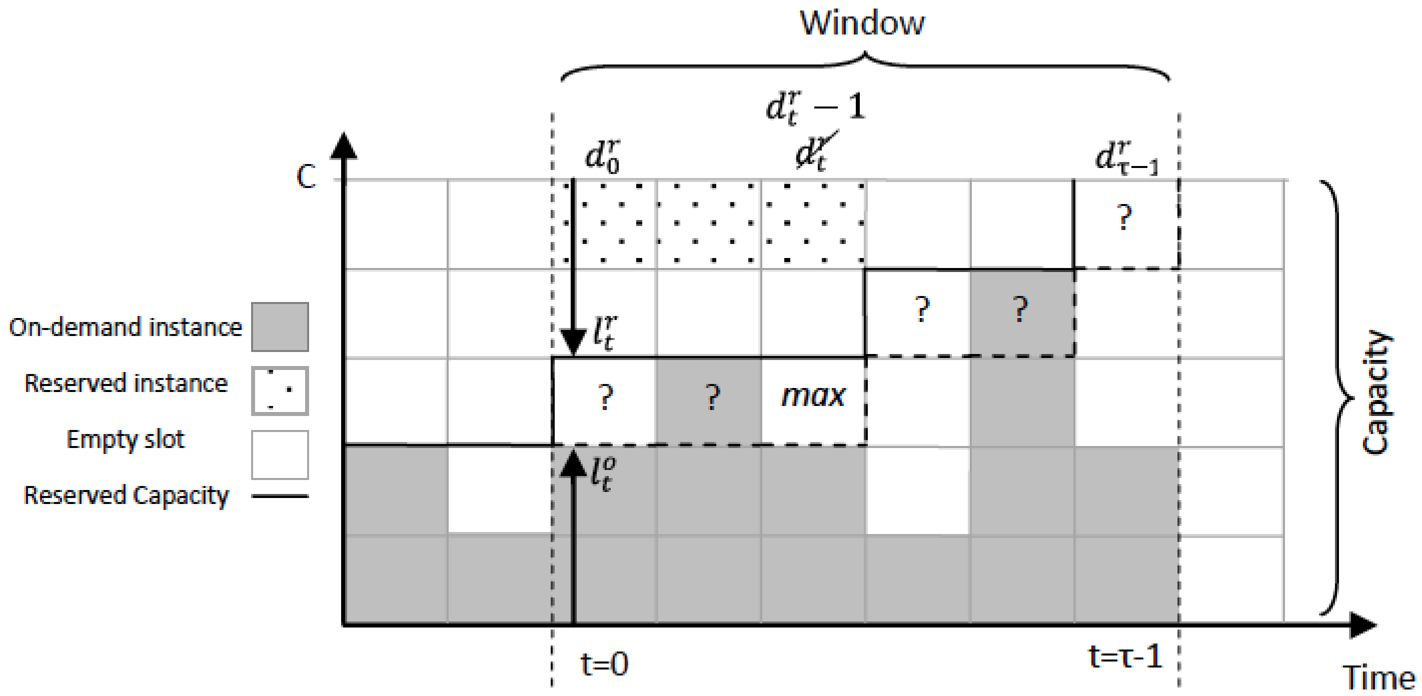
B: The nu… instances per block of capacity

e.g., B = 100 VMs

T: The number of billing cycles per time slot

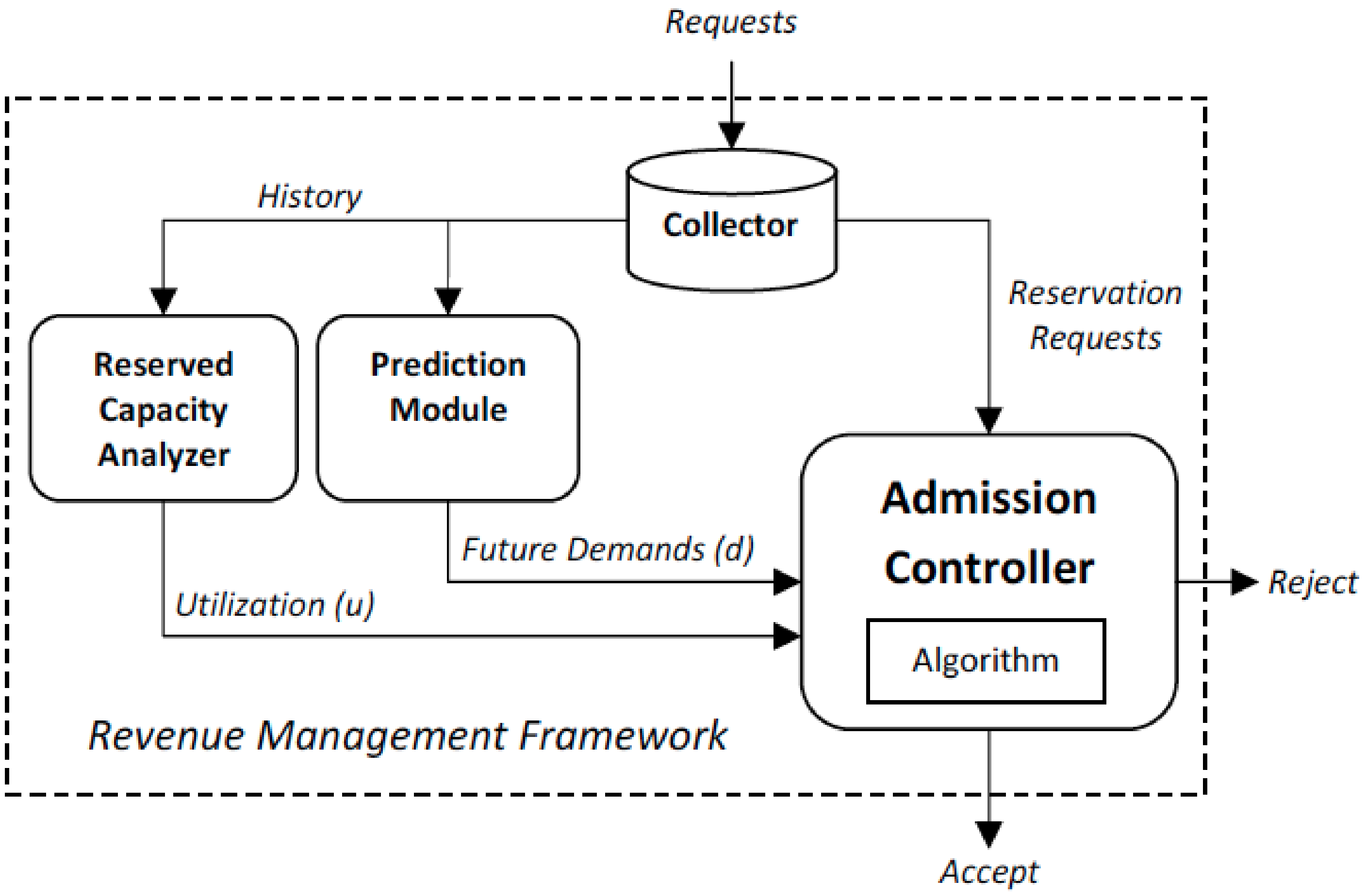
e.g., T = 168 hours.

The heuristic algorithm



Computational Complexity: $O(\tau \times C)$

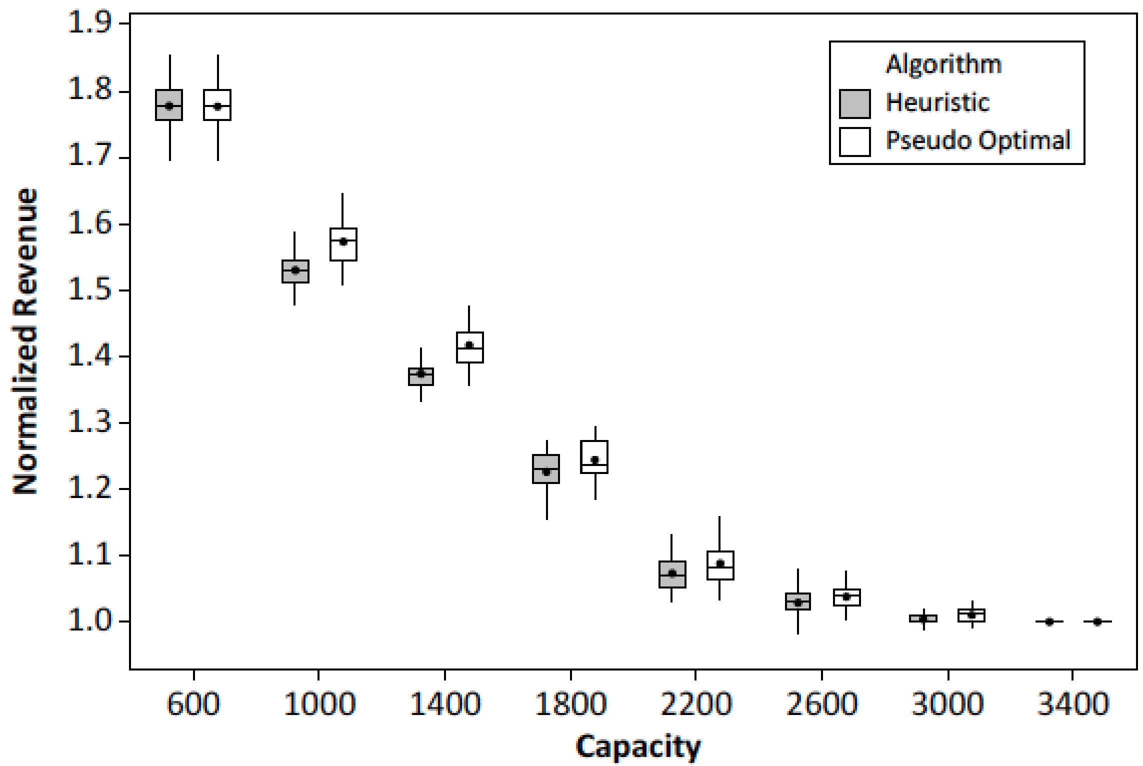
Key modules of the revenue management framework



Performance Evaluation

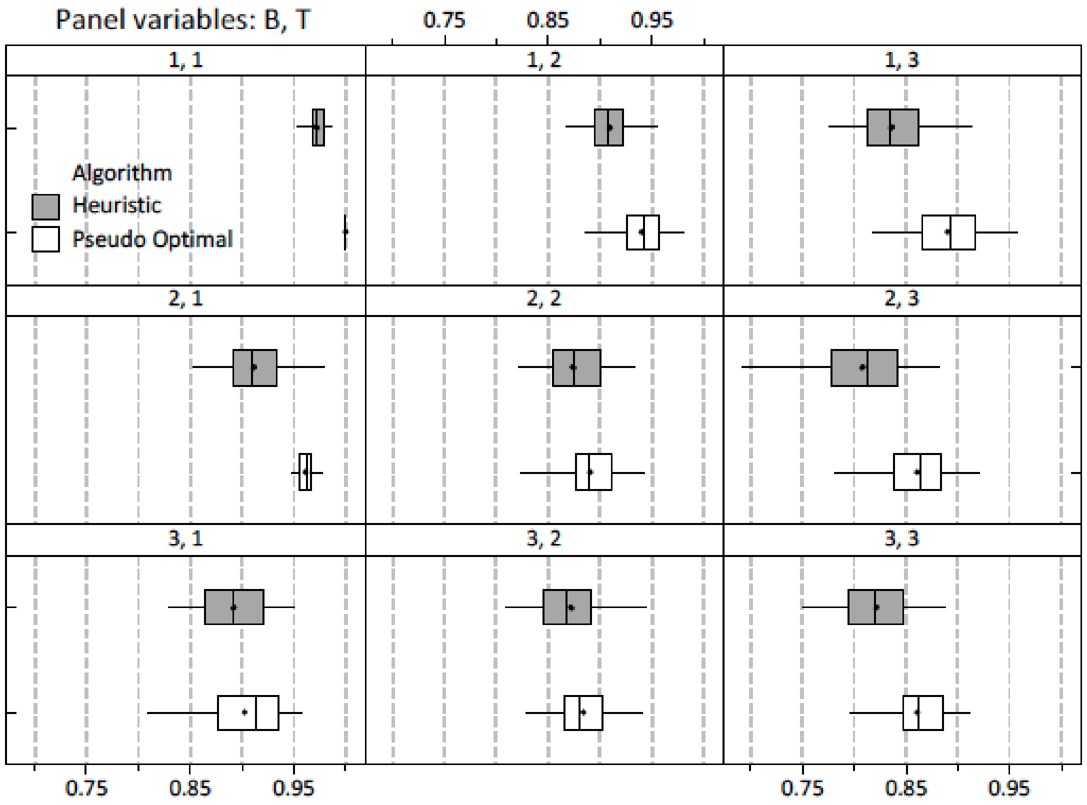
- We evaluate our proposed framework through:
  - Large-scale simulations (12 months)
  - Driven by cluster-usage traces that are provided by Google.
- No publicly available workload traces of real-world IaaS clouds:
  - We propose a scheduling algorithm that generates VM requests based on the user resource usage in these traces.
- Pricing conditions that are aligned with those of Amazon EC2
- Benchmark Algorithm:
  - no admission control referred to as *no-control*
- Simulation Environment
  - Extended CloudSim (pricing plans and the proposed revenue management framework)

Experimental Results (1)



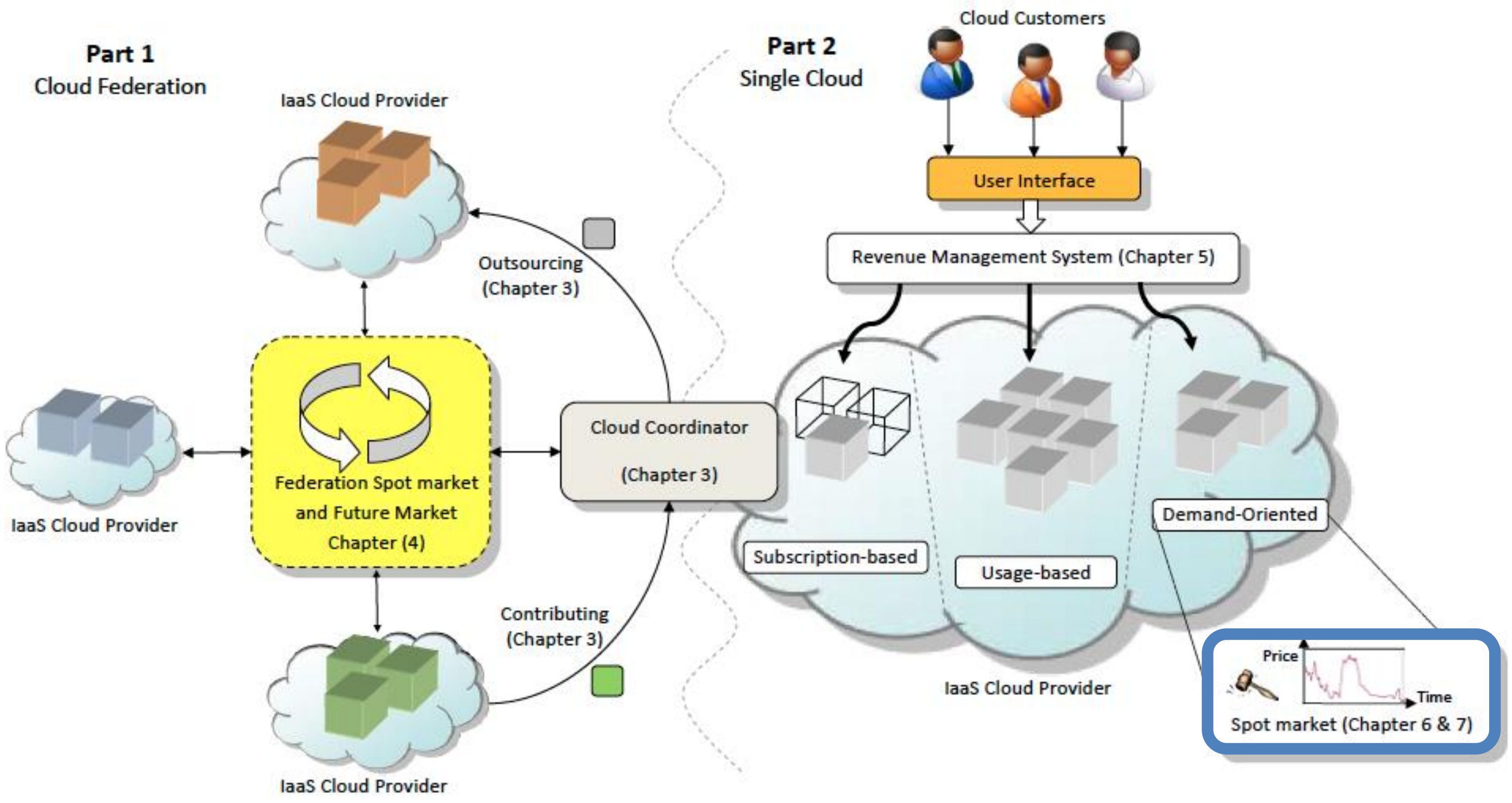
The revenue performance of the proposed revenue management framework under different algorithms normalized to the outcome of no-control algorithm
(B = 100 and T = 75).

Experimental Results (2)



The revenue performance of the pseudo optimal and heuristic algorithms with different values of B and T. All values are normalized to the outcome of the optimal solution.

Solutions



Part 1
Cloud Federation
IaaS Cloud Provider
Outsourcing
(Chapter 3)
IaaS Cloud Provider
Federation Spot market
and Future Market
Chapter (4)
Cloud Coordinator
(Chapter 3)
Contributing
(Chapter 3)
IaaS Cloud Provider
Part 2
Single Cloud
Cloud Customers
User Interface
Revenue Management System (Chapter 5)
Subscription-based
Usage-based
Demand-Oriented
IaaS Cloud Provider
Price
Time
Spot market (Chapter 6 & 7)

An Auction Mechanism for Cloud Spot Markets

Derived from:

Adel Nadjaran Toosi, Kurt Vanmechelen, Farzad Khodadadi, and Rajkumar Buyya, "**An Auction Mechanism for Cloud Spot Markets**," *ACM Transactions on Autonomous and Adaptive Systems,* Volume 11, Issue 1, 2016.

Motivation

Computational resources sold by a cloud provider can be characterized:

As a non-storable or perishable commodity

Demand for computational resources is:

non-uniform over time

These motivates the use of dynamic forms of pricing in order to optimize revenue

Well-designed auction mechanisms are particularly effective

Why Auctions?

Incentivize users to reveal their true valuation of the service (i.e., report the price they are willing to pay for resources)
Ensure resources are allocated to those who value them the most
Correctly price resources in line with supply and demand conditions by creating competition among buyers.

Background

Auction:

A common market mechanism with a set of rules determining prices and resource allocations on basis of bids subm
from the market participants.

Auctions can be in assorted shapes and have different characteristics:

Single dimensional (e.g., only bid price) or multi-dimensional (e.g., bid price plus quantity),
Single-sided (e.g., only customers submit bids) or double-sided (e.g., double auction),
Open-cry or sealed-bid,
Single-unit (e.g., a single good or service) or multi-unit (e.g., multiple units of the good),
Single item (e.g., one type of service) or multi-item (e.g., combinatorial auction).

Goals in designing auction:

Truthfulness, Revenue maximization, Allocative efficiency, Fairness

Background

- Optimal mechanism design
  - When goal of mechanism is to maximize the profit or revenue for the seller (provider)
- Optimal mechanism design can be categorized in
  - Bayesian optimal mechanism design
    - the valuations of the participants in the auction are drawn from a known prior distribution
    - based on the seminal work by Myerson
  - Prior free optimal mechanism
    - Determining the prior distribution is not practical, convenient or even possible in advance
    - Digital Goods: Competitive Framework by Goldberg and Hartline

Problem Definition and Goals

Amazon Web Services (AWS)

- Spot Instances (Auction-like mechanism)
  - Customers communicate their bids a VM instance hour to AWS.
  - AWS reports a market-wide spot price at which VM instance use is charged,
  - while terminating any instances that are executing under a bid price that is lower than the market price.

AWS has revealed no detailed information regarding

- Their auction mechanism and
- The calculation of the spot price.

The design of an auction mechanism

- Efficient, fair, and profit-maximizing
- Open research challenge,
- Of great interest to cloud providers.

Contributions

We design an auction mechanism

aimed at maximizing profit from selling the spare capacity of non-storable resources ava

in cloud data centers

A multi-unit, online recurrent auction, two-dimensional bid domain

Extension of Consensus Revenue Estimate (CORE) mechanism

Work by Goldberg and Hartline

Envy-free, truthful with high probability and generates near optimal profit for the provider.

Contributions (cont.)

Benchmark Algorithm: An optimal auction mechanism (HTAOPT)

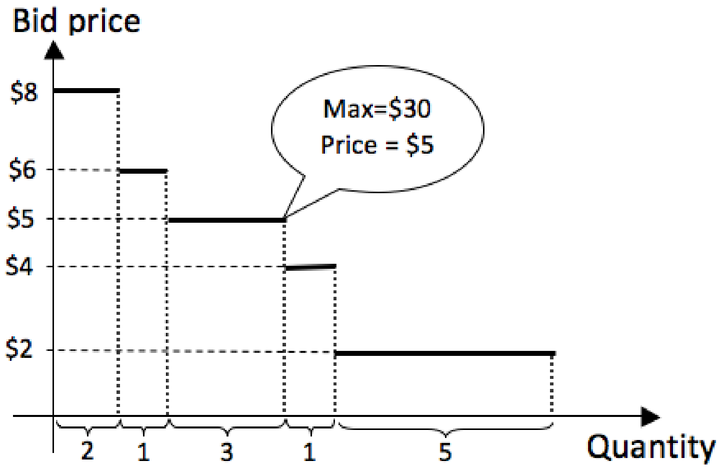
- Dynamic programming
- To quantify the efficiency loss caused by the lack of information on the amount of time a bidder wants to run a VM.

A method for dynamically computing

- A reserve price
  - The lowest possible price that the provider accepts.
- Based on a coarse grained data center power usage model

Research Problem

Optimal Single Price Auction: $\mathcal{F}(\mathbf{d})$



The order-indeper

Optimal order-ind

- Not single price (i.e., not envy-free)
- It is not fair

$\mathcal{F}(\mathbf{d}_{-i})$

How to compute a single price for an order-independent auction while attaining the revenue of the optimal auction?

Consensus Revenue Estimate

We are interested in a mechanism that provides us with a sufficiently accu $\mathcal{F}(\mathbf{d})$ estimate of

While the estimate is constant on d_{-i} for all i (i.e $\mathcal{F}(\mathbf{d}_{-i})$ ieves consensus).

$\mathcal{F}(\mathbf{d})$ is limited by a constant fraction of

it is possible to pick a good estimate

We consider a limit on maximum quantity (r)

Let r denote the supremum of the number of requested units in **d**

Let m be the number of sold units in *F*

$$\frac{m-r}{m}\mathcal{F}(\mathbf{d}) \leq \mathcal{F}(\mathbf{d}_{-i}) \leq \mathcal{F}(\mathbf{d})$$

We use Consensus Re[illegible]

Revenue Extraction Mechanism

Cost sharing mechanism proposed by Moulin and Shenkar
Extracts a target revenue from the set of bidders if this is possible
It finds the *k* bidders with the highest bid values that allow for the extraction of *R* (target revenue)
Given a target revenue *R*, the revenue extraction mechanism is truthful for the price dimension but not for the quantity dimension.

Reserve Price

The reserve price for most perishable goods and services is considere
low at their expiration time

The reserve price for flight seats is theoretically negligible.

A significant part of the service cost in cloud data centers is related
power consumption of physical servers.

We propose a method for calculating dynamic reserve prices based on
cost model that incorporates data center PUE (load, outside temperat
and electricity cost.

The Online Ex-CORE Auction

Algorithm 3 The Online Ex-CORE Auction

Input: $\mathbf{d}$, p_{cur}, p_{optprv} ▷ $\mathbf{d}$ is the list of orders, sorted in descending order of bids, p_{cur} is current market price, p_{optprv} is the optimal single price in the previous round.

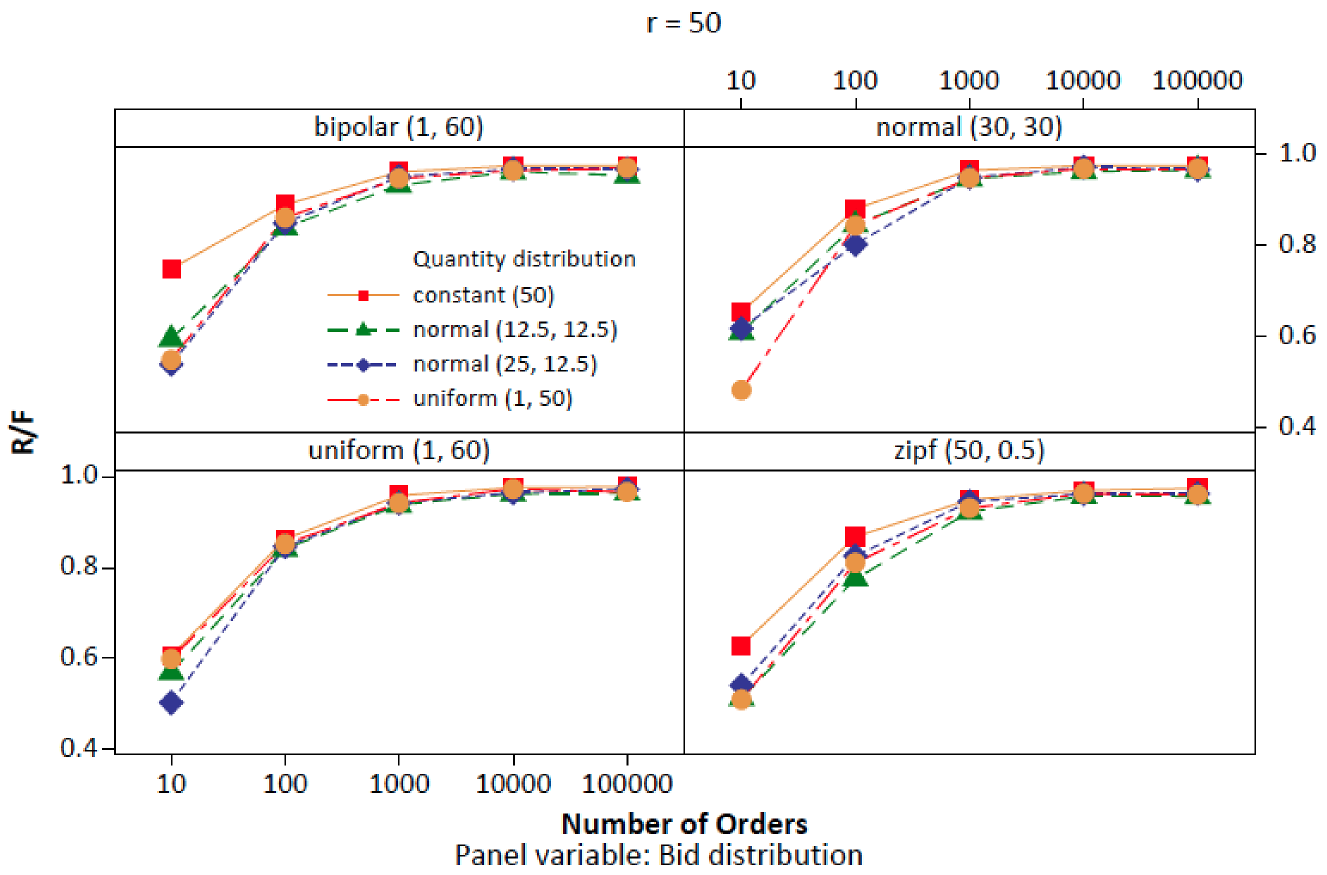
Output: p ▷ Sale Price

$p_{opt} \leftarrow opt(\mathbf{d})$
if $p_{opt} = p_{optprv}$ **then**
    return p_{cur}
end if
$r \leftarrow$ the largest r_i in $\mathbf{d}$
$m \leftarrow \operatorname{argmax}_{\sigma_i(\mathbf{d})} \; b_i \sigma_i(\mathbf{d})$
if $m \leq r$ **then**
    return p_{opt} ▷ single optimal price
else
    $\rho \leftarrow \frac{m}{m-r}$
    Find c in $\rho\, ln(c) + \rho - c = 0$
    $u \leftarrow rnd(0,1)$ ▷ chosen uniformly random on [0,1]
    $l \leftarrow \lfloor log_c(\mathcal{F}(\mathbf{d})) - u \rfloor$
    $R \leftarrow c^{(l+u)}$
    $j \leftarrow$ the largest k such that $\frac{R}{\sigma_k(\mathbf{d})} \geq b_k$
    return $\frac{R}{\sigma_j(\mathbf{d})}$
end if

Benchmark Algorithms

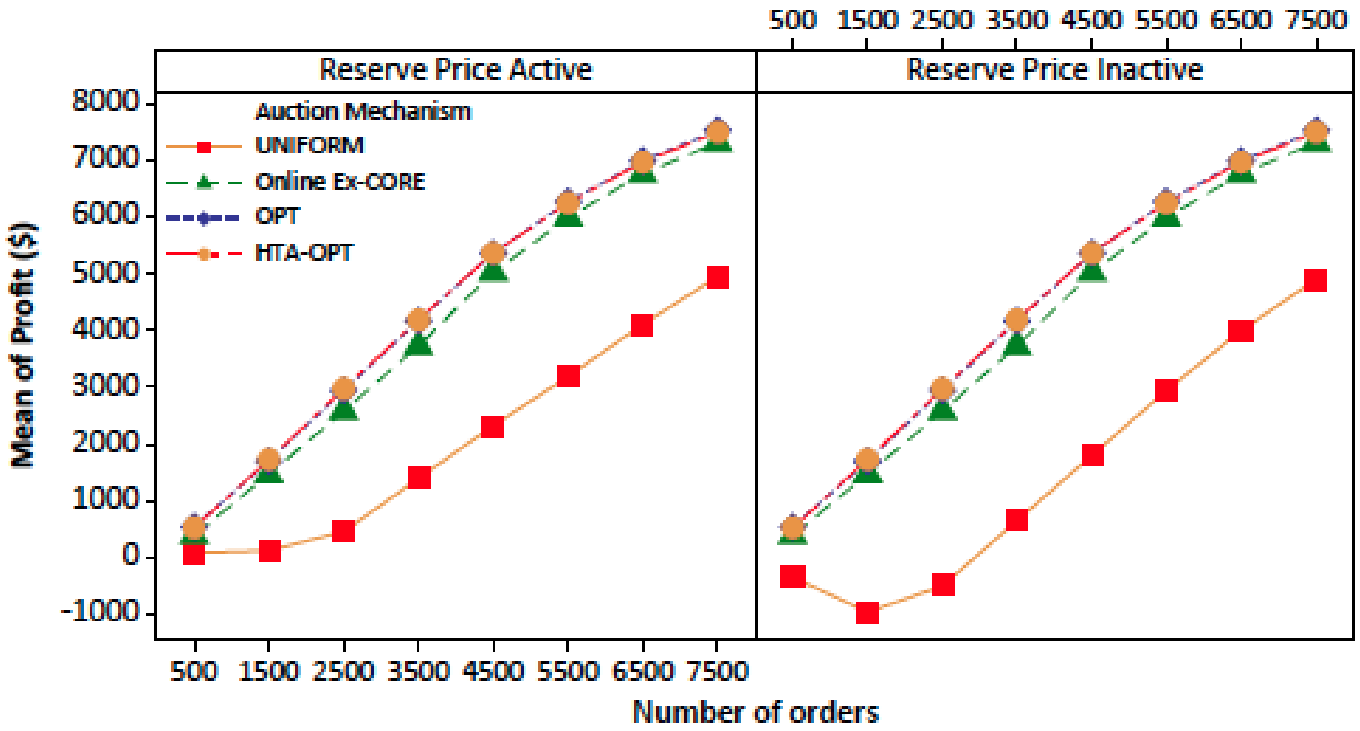
Optimal Single Price Auction (OPT)
Holding Time Aware Optimal Auction (HTA-OPT)
Uniform Price Auction

Performance Evaluation (Single Round)



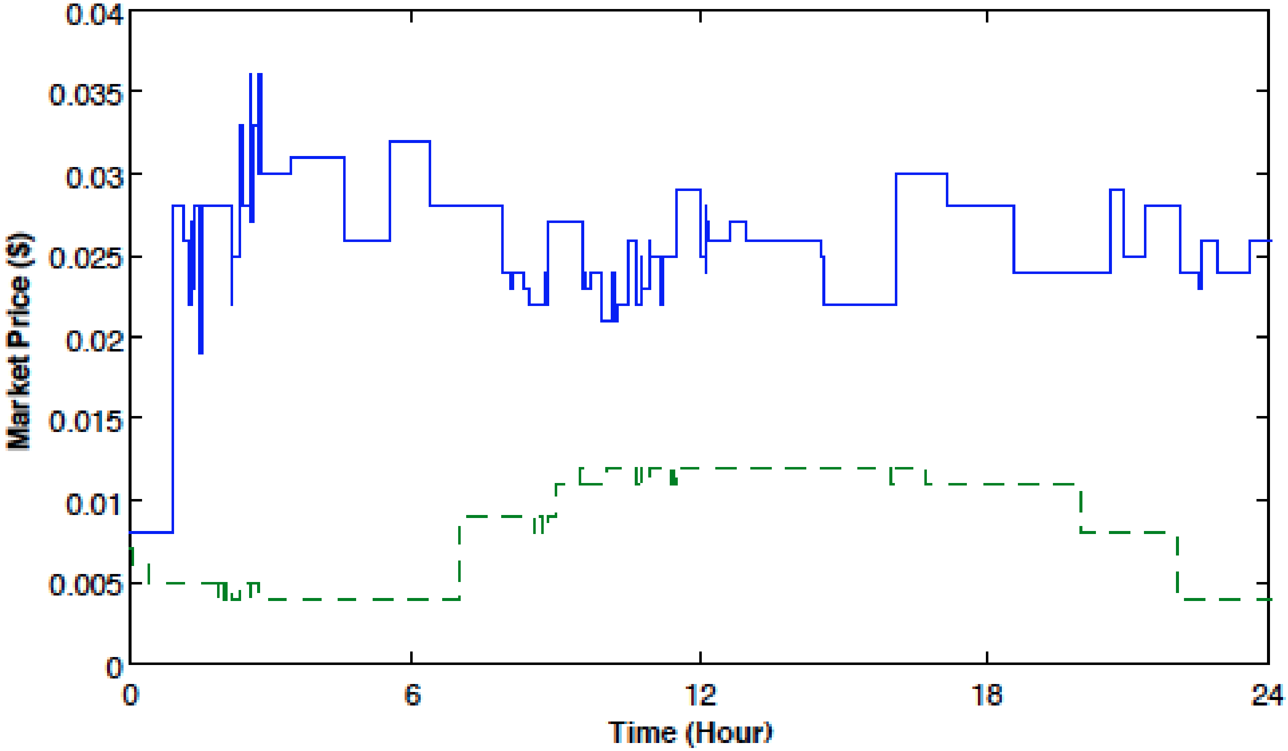
Ratio of gained revenue by the Ex-CORE auction to optimal auction under different distribution of orders

Performance Evaluation (Online)



Average profit gained and with different auction mechanisms.

Results



Reserve price (green dashed line) and spot market price generated by online Ex-CORE (blue solid line) in a sample simulation run when the number of orders is 1500.

SipaaS: Spot instance pricing as a Service

Derived from:

Adel Nadjaran Toosi, Farzad Khodadadi, and Rajkumar Buyya, "***SipaaS*: Spot instance pricing as a Service and its Implementation in OpenStack**," Software Software: Practice and Experience, Wiley Press, 2014 (in preparation).

Spot instance pricing as a Service

The implementation of the proposed auction mechanism

by identifying a framework called Spot instance pricing as a Service (SipaaS).

SipaaS:

is an open source project offering a set of web services to price and sell VM instances in a spot market.

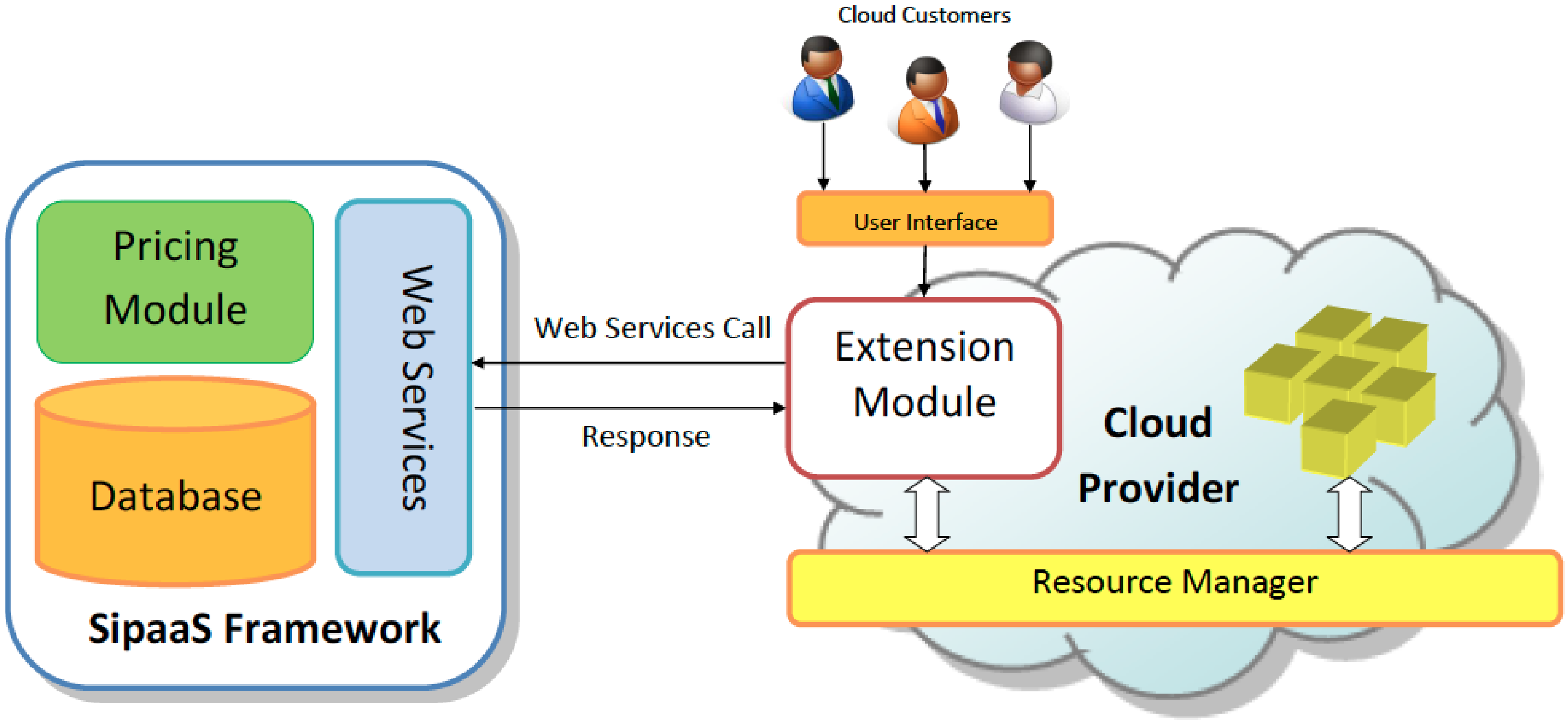
Utilizing SipaaS

Cloud providers require installation of add-ons in their existing platform to make use of the framework.

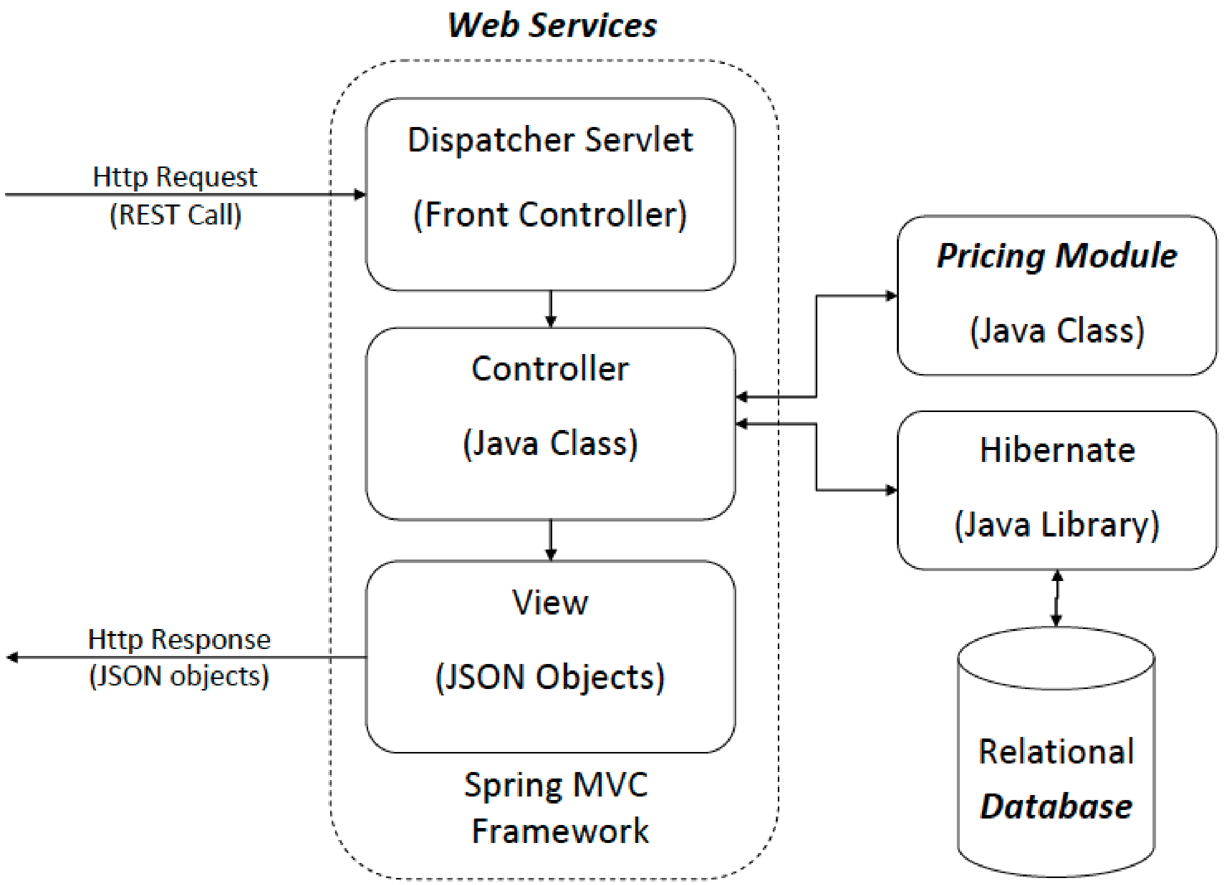
An extension to the Horizon –the OpenStack dashboard project

- To employ SipaaS web services and to add a spot market environment to OpenStack.

System Model



SipaaS Architecture



List web services

Web Service Name	Input parameter(s)	Output
register	name	credentials
unregister	accesskey,secretkey	status
regvmtype	accesskey,secretkey,type	status
unregvmtype	accesskey,secretkey,type	status
setavailables	accesskey,secretkey,vmtype,quantity	price
setmaxqty	accesskey,secretkey,vmtype,quantity	status
setreserveprice	accesskey,secretkey,vmtype,value	price
setmaxprice	accesskey,secretkey,vmtype,value	status
addorder	accesskey,secretkey,vmtype,quantity,bid,ref	price
updateorder	accesskey,secretkey,quantity,ref	price
removeorder	accesskey,secretkey,ref	price
pricehistory	accesskey,secretkey,vmtype,fromdate,todate	price(s)

OpenStack and Horizon

OpenStack

An open-source cloud management platform

Developed to control large pools of compute, storage, and networking resources in a data center.

Horizon

The OpenStack dashboard project

A web based user interface to OpenStack services

Provides administrators with management capabilities

Extensions for Horizon

To add spot market facilities to OpenStack,

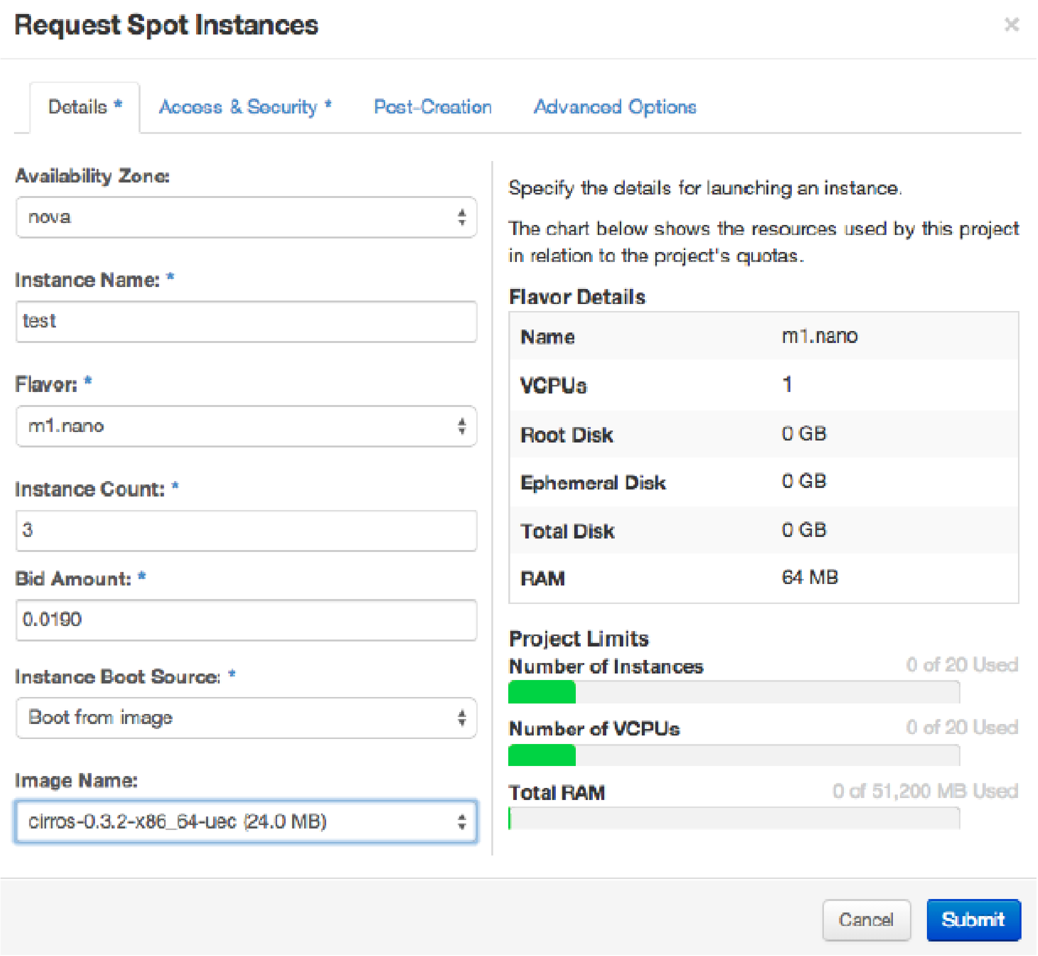
- we extended Horizon to be capable of using the services provided by SipaaS.

We added a new panel through which system administrators are capable of enabling spot market support

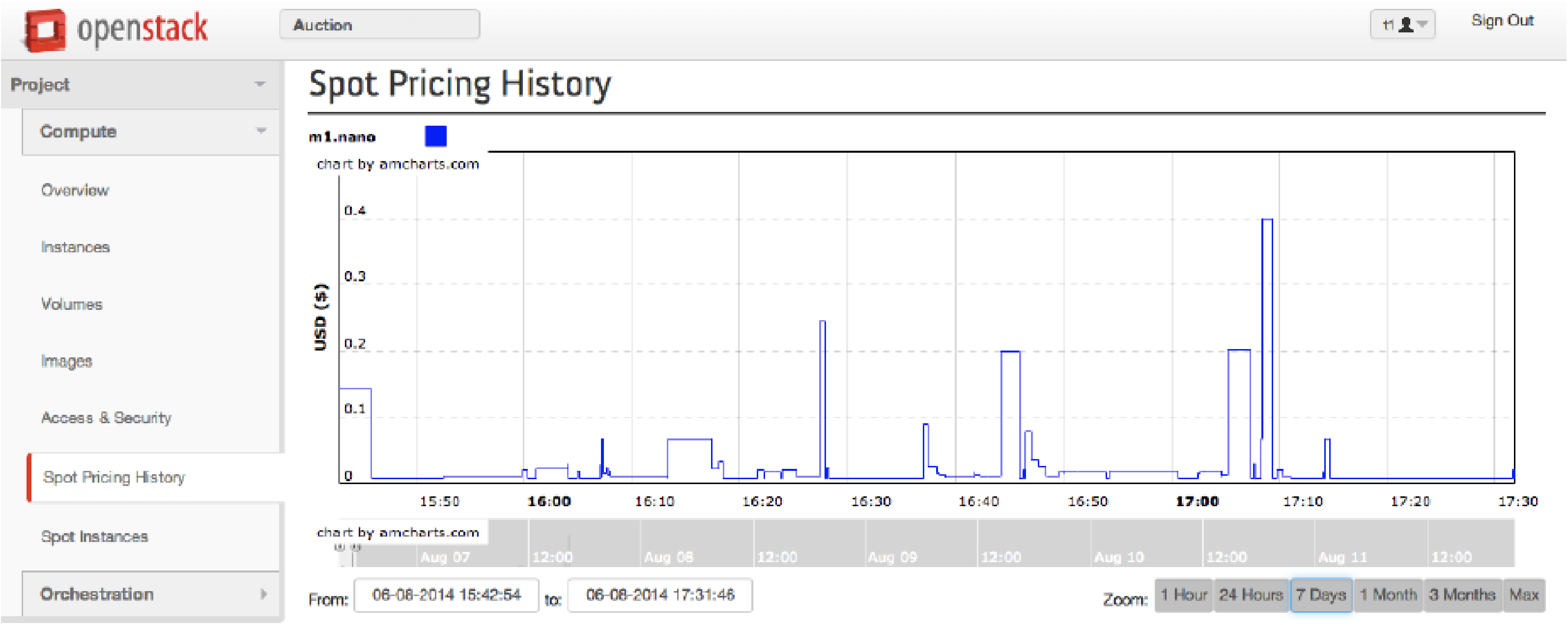
- Maximum and minimum amount of bid price for users,
- Number of available VMs for allocation,
- Maximum number of VMs a user can request.

We added panels for requesting spot instances and viewing spot market price history

Requesting spot instances web page



Spot Pricing History Webpage



Evaluation and Validation

We conducted 20-minute experiment with 10 participants (i.e., spot market users).

Participants are selected from a group of professional cloud users who have satisfactory level of knowledge about the spot market.

Participants are divided into two groups of five:

(i) Group T or truthful bidders and

(ii) Group C or counterpart bidders who have the freedom to misreport their true private values to maximize their utility

User	Price Value ($)	Quantity
T1, C1	0.0691	2
T2, C2	0.0092	1
T3, C3	0.0475	1
T4, C4	0.0232	2
T5, C5	0.0184	1

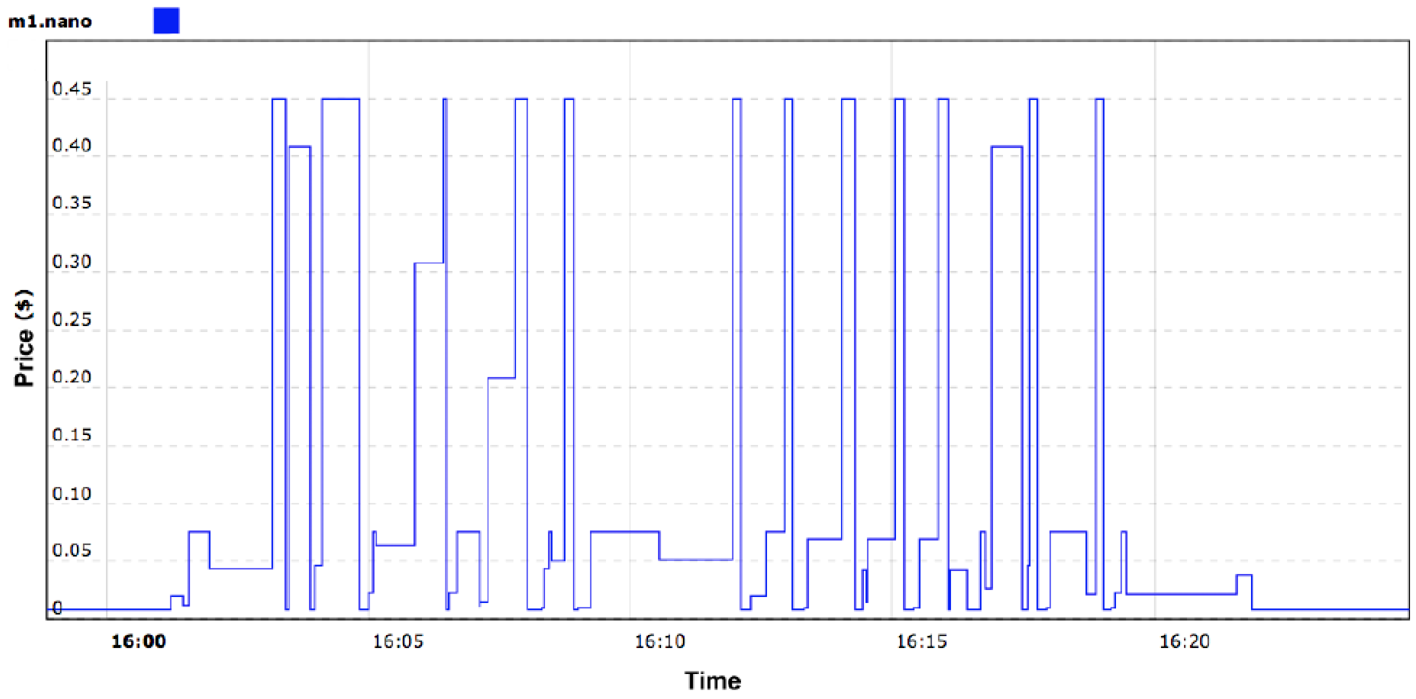
Utility Function

- Utility function for one time slot instance usage, formulated as below:

$$u(r,b) = \begin{cases} (qv - rp)x\,, & \text{if } b \geq p \text{ and } r \geq q; \\ 0\,, & \text{otherwise.} \end{cases}$$

- r: requested number of instances
- b: bid price value
- q: true private quantity
- v: true private price value
- p: spot market price at the time of order submission
- x: A Boolean value describing whether the order is accepted or not, respectively.

Results



Spot market price fluctuation during the experiment.

Results (Cont.)

User	Total Cost ($)	Number of Full Time Slots	Utility Value ($)
T1	1.2964	16	**0.9148**
C1	1.8216	17	0.5278
T2	0.0000	0	**0.0000**
C2	10.0227	18	-9.8571
T3	0.1896	6	0.0954
C3	0.2280	8	**0.1520**
T4	0.0436	1	**0.0030**
C4	3.6810	5	-3.4490
T5	0.0000	0	**0.0000**
C5	0.0738	2	-0.0370

Total cost, the number of full time slots usage, and utility value of experiment participants.

Take Home Message

- Think problems at a higher level
- Computation problem is not only a technical solution but also an economic mechanism
- We need to learn to formulate the problem, model it, and understand it in real experiments

Future directions

Advanced Resource Provisioning Policies

Power Saving, Future load prediction

Option Trading Strategies

Portfolio, Bulk trading, Selling options

Game Theoretical Analysis of Cloud Federation

Selfish provider and its own profit, Social welfare

Customer Diversion in Revenue Management Framework

Bumped by the admission control mechanism

Revenue Management with Overbooking

Underutilized reserved capacity

Multi-dimensional Truthful Mechanism Design

Two-dimensionally truthful (bid price and quantity)