Resource Provisioning Policies to Increase Iaas Provider’s Profit in a Federated Cloud Environment

Presenter: Shi Shengkai
Department of Computer Science
The University of Hong Kong

Cloud computing

- Long-held dream of computing as utility
- Virtualized resources
- Customers pay as they use
- Delivery of IT services
  - ***Infrastructure as a service (Iaas)***
  - Platform as a service (Paas)
  - Software as service (Saas)

Resource provisioning in cloud

- Demand is very uneven
  - Average demand of the system is several times smaller than the peak demand
- A limited amount of resources
  - Reject new requests
  - Relax Quality of Service (QoS)

Cloud federation

- Cloud federation is a collection of individual cloud providers, which collaborate by trading resources.
- Desired feature of cloud
  - Illusion of infinite computing resources
- By exploiting cloud federation potentials, providers are able to dynamically increase the available resources to serve requests

Cloud federation motivation

- During peak times, obtain extra resources from other members (outsourcing)
  - Avoid losing customers
  - Avoid losing reputation by violating QoS
- During lower times, lease idle resources (contributing to the federation)
  - Avoid wasting resources

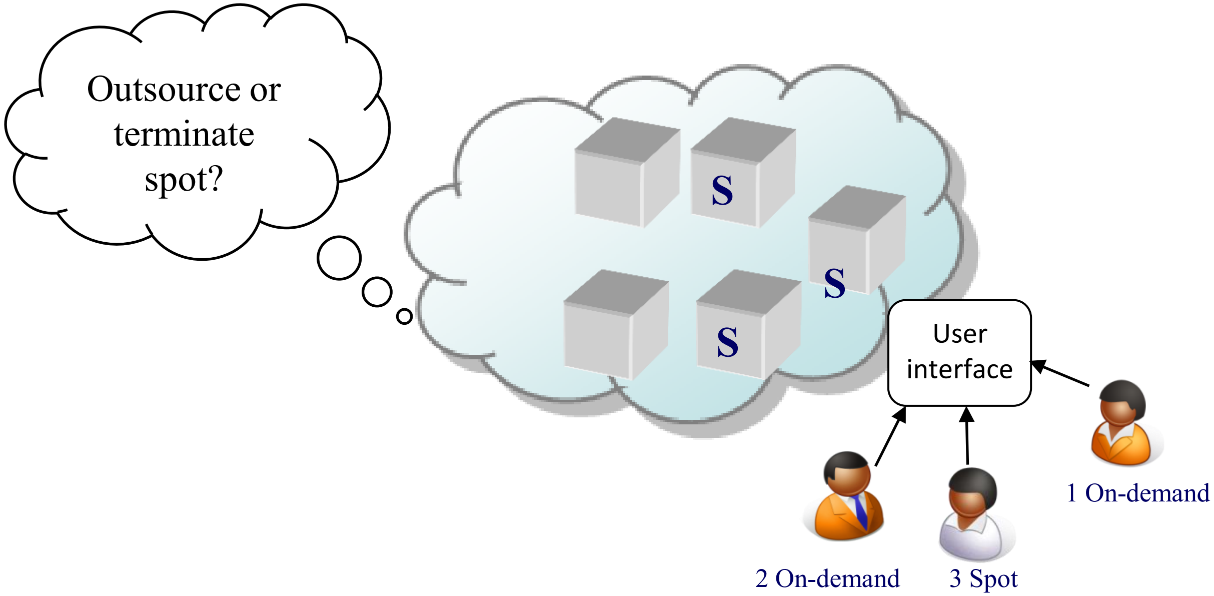
Federation-aware resource provisioning

- How providers can exploit federation to dynamically increase their data center capacity?
- When providers should sell and buy resources and what are the proper contracts and pricing schemes?
- What types of high-level infrastructure and mechanisms are required to outsource extra demands and contribute under-utilized capacity to federation members?

Spot instance

- Spot instances can significantly lower your computing costs for time-flexible, interruption-tolerant tasks.
- Spot instances and on-demand instances only differ in their pricing model and the possibility of being interrupted when the spot price exceeds your max bid.

Problem statement



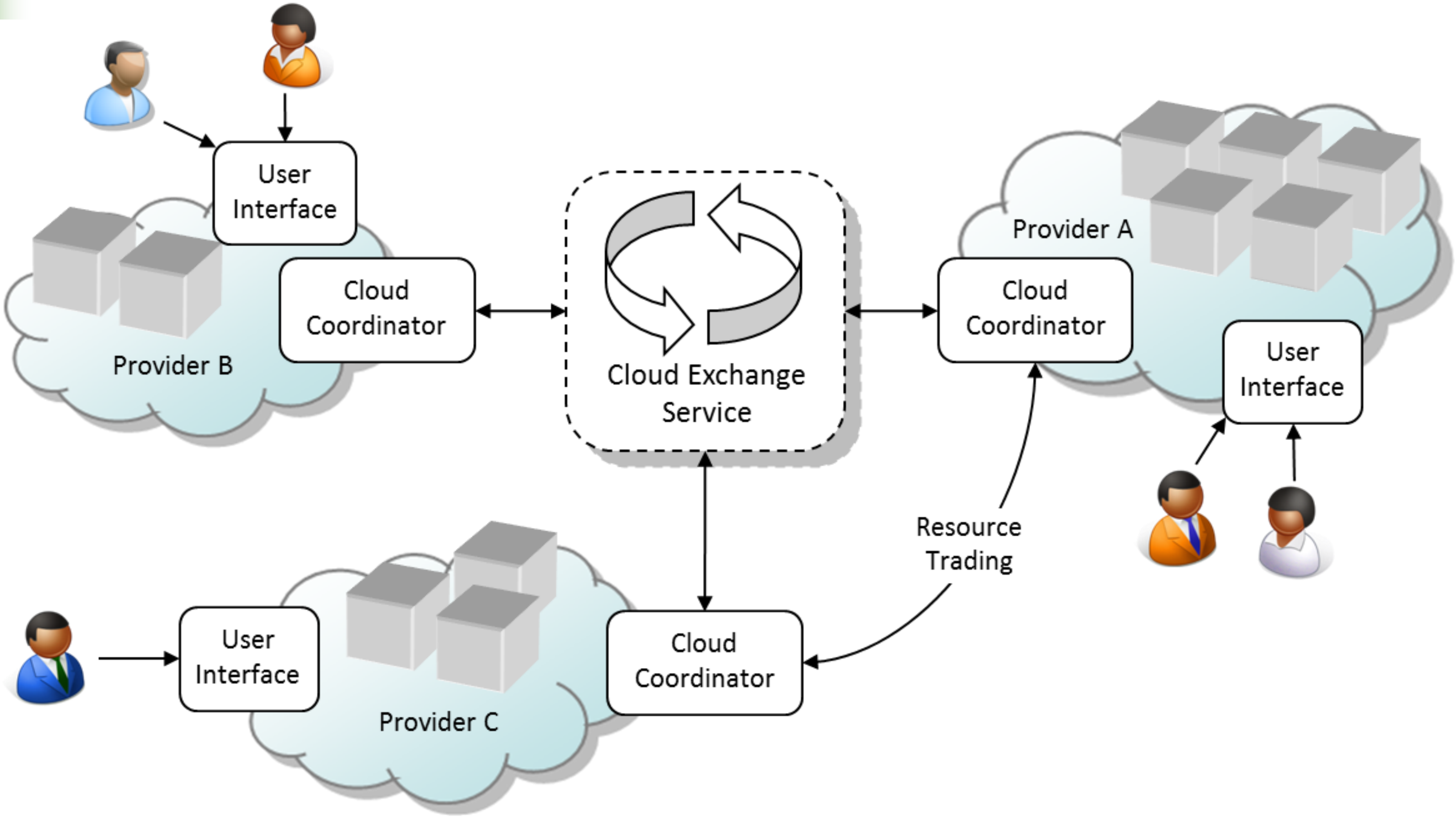
System model

Interaction between customers and providers

- Each federated cloud provider owns a data center and serves a number of customers.
- The user request has to be entirely served in one data center.
- Only on-demand and spot VM requests are considered.
- Consolidation of requests is not considered.
- Outsourcing is only considered for on-demand requests.

System model

Interaction between cloud federation and providers



System model

Interaction between cloud federation and providers

- Cloud exchange
  - Information service directory
  - Available resources from the members of federation
- Cloud coordinator
  - Decision on allocating additional resources from another cloud provider
  - Publishing idle capacity shares with cloud exchange
  - Resources pricing for contributing capacity

System Model

Interaction between cloud federation and providers

- Federation Level Agreement (FLA)
  - Instant federation price of a resource per hour

$$F = \frac{M_p - M_{idle}}{M_p}(F_{\max} - F_{\min}) + F_{\min}$$

    - M_p : total capacity
    - M_{idle}: idle capacity of the provider data center
    - F_{max} : the on-demand VM price to customers
    - F_{min} : the minimum profitable price for the provider

Policies

- No 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

Policies

- Federation-Aware Profit Oriented (FAPO)
  - Based on analytical analysis of instant profit, it decides between outsourcing and termination of spot VMs

Federation-Aware Profit Oriented (FAPO)

- P(t), the instant profit of the provider in time t:

$$P(t) = R(t) - C(t)$$

  - R(t): revenue at time t
  - C(t): cost at time t

Federation-Aware Profit Oriented (FAPO)-revenue

- R(t) can be obtained as follow:

$$R(t) = R_o(t) + R_s(t) + R_{fed}(t) + R_{out}(t)$$

  - Ro(t): revenue of on-demand VMs at time t
  - Rs(t): revenue of spot VMs at time t
  - Rfed(t): revenue of contributed VMs to federation those local resources used by other members of the federation
  - Rout(t): revenue of outsourced VM requests

Federation-Aware Profit Oriented (FAPO)-revenue

$$R_s(t) = vm_s(t) \cdot F_s(t)$$

$$R_o(t) = vm_o(t) \cdot F_o$$

$$R_{out}(t) = vm_{out}(t) \cdot F_o$$

$$R_{fed}(t) = \sum_{i=1}^{vm_{fed}(t)} F_{fed_i}$$

- F_o: the on-demand resource price per resource per hour
- $F_s(t)$: the price of the spot VMs at time t
- $vm_o(t)$: the number of on-demand VMs running locally
- $vm_{out}(t)$: the number of outsourced VMs
- $vm_s(t)$: the number of running spot VMs

Federation-Aware Profit Oriented (FAPO)-cost

- C(t) can be obtained as follow:

$$C(t) = C_p(t) + C_{out}(t)$$

  - CP(t) is the operational cost
  - Cout(t) is the cost of outsourced VMs that a provider pays to federation members hosting its requests:

$$C_{out}(t) = \sum_{i=1}^{vm_{out}(t)} F_{out_i}$$

  - Fout-i is the price per resource per which is paid for each outsourced vmi.

Federation-Aware Profit Oriented (FAPO)

- Putting all the above equations together

$$\Downarrow$$

$$P(t) = vm_s(t) \cdot F_s(t) + vm_o(t) \cdot F_o + vm_{out}(t) \cdot F_o + \sum_{i=1}^{vm_{fed}(t)} F_{fed_i} - \sum_{i=1}^{vm_{out}(t)} F_{out_i} - C_p(t)$$

Federation-Aware Profit Oriented (FAPO)

- FAPO policy has two choices for incoming *n* on-demand VMs arriving at time *t*, and local infrastructure can only accommodate m VMs (m < n).

Federation-Aware Profit Oriented (FAPO)

1. Terminate the n-m spot VMs

$$\begin{aligned} P_1(t') = {} & (vm_o(t)+n)\cdot F_o + vm_{out}(t)\cdot F_o - C_p(t) \\ & + (vm_s(t)-(n-m)-k)\cdot F_s(t) + k\cdot F_s(t') \\ & + \sum_{i=1}^{vm_{fed}(t)} F_{fed_i} - \sum_{i=1}^{vm_{out}(t)} F_{out_i} \end{aligned}$$

2. Outsource the new request

$$\begin{aligned} P_2(t') = {} & vm_o(t)\cdot F_o + vm_{out}(t)\cdot F_o \\ & + n\cdot F_o - C_p(t) + vm_s(t)\cdot F_s(t) \\ & + \sum_{i=1}^{vm_{fed}(t)} F_{fed_i} - \sum_{i=1}^{vm_{out}(t)} F_{out_i} - n\cdot F_{offer} \end{aligned}$$

$$\Longrightarrow \quad P_1(t')-P_2(t') = k\cdot F_s(t')-(n-m+k)\cdot F_s(t)+n\cdot F_{offer}.$$

$P_1(t') - P_2(t') \geq 0$ ☑ | $P_1(t') - P_2(t') < 0$ ☑

Performance evaluation-setup

- Simulation study with CloudSim
- 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 data centre has 128 servers, and each server supports 8 VMs.

Performance evaluation-setup

- 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

Evaluation parameters

- System load
  - The arrival rate of requests has been selected to adjust the load of a provider
  - *aarr* parameter of the Lublin workload model between 8.2 and 6.4
- Number of providers
  - 3,5,7

Performance metrics

- Profit

$$Profit(\Delta t) = Revenue(\Delta t) - Cost_{out}(\Delta t)$$

- Utilization

$$Utilization(\Delta t) = \frac{\sum_{i=1}^{vm} runtime(vm_i)}{vm_{max} \cdot \Delta t}$$

- Number of rejected on-demand requests

Results

- Results for Profit and Utilization are the normalized values for each metric using the result obtained for the NFTI policy as the base value.

Results

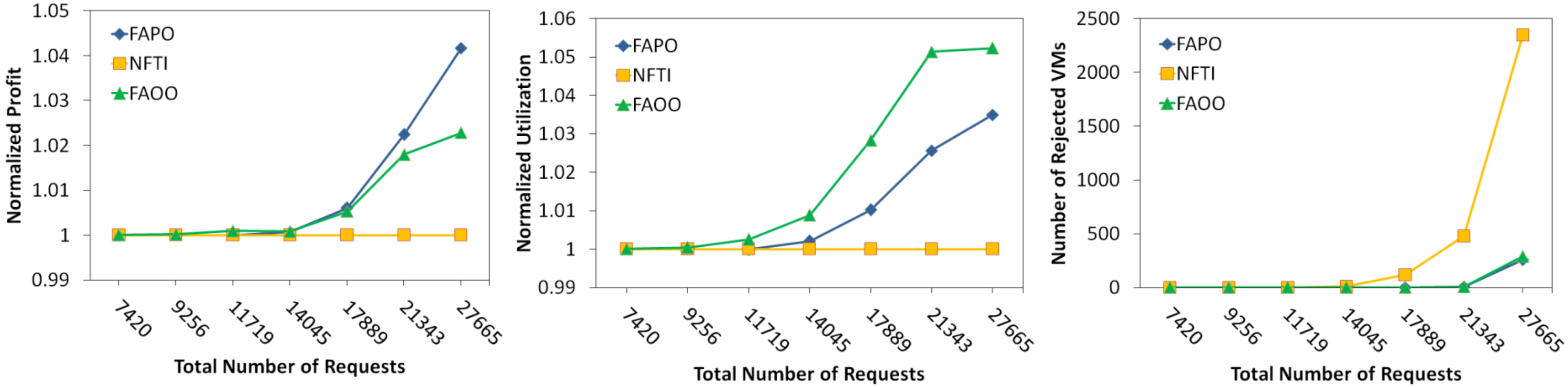


Figure: impact of load on (a) Profit (b) Utilization (c) Number of rejected on-demand VMs, for a provider with different policies.

Results

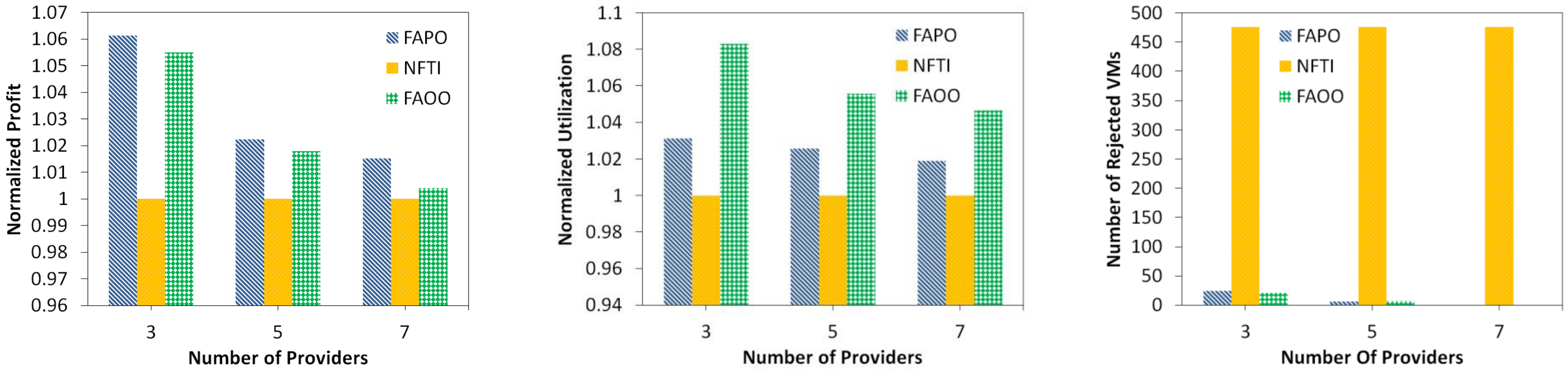


Figure: impact of number of providers on (a) Profit (b) Utilization (c) Number of rejected on-demand VMs for a provider with different policies.

Comments and inspirations

- Next generation cloud service
- Too much simplification
- Price model of federation price is too simple
- Lack of interoperability among providers
- How to guarantee the fairness?
- How to achieve the best geographical distribution of placements for requests？

Thank You !