Federated Cloud Pricing(Draft)

When there is no federation among clouds, it is difficult to switch to other cloud service providers after it is involved in the current cloud service provider. The data is locked-in. The federation cloud makes it easily for users to switch among different cloud service providers.

The cost of a cloud provider changes with the price of power dynamically. There are papers discussing how to make use of the dynamic power price in different places to reduce the cost of clouds. The federation cloud also promotes this reduction.

The question is how the federation cloud should price its users. The pricing policy should achieve the maximum profit for the federation cloud. Meanwhile, to make an individual cloud provider willing to participate in the federation, the profit expectation should be increased after the federation.

1 Problem Formulation

The federation cloud has K cloud providers. Cloud provider k has $M_k, 1 \leq k \leq K$ distributed data centers, denoted by $\mathcal{D}^k = \{D^{k1}, \ldots, D^{kM_k}\}$. We assume data centers of the same cloud provider have homogeneous servers. Let S^k denote the server in cloud provider k. Each data center D^{ki} has N^{ki} servers. Consider the system operates in slotted time $t \in 0, 1, 2, \ldots$.

1.1 The request model

The federation cloud provides V types of VMs (as the different types of instances in AWS). We consider the requests for the same type of VMs as the same type of jobs. In the time slotted system in this paper, a VM may need to be hosted for different time slots according to users' workloads. As the user can divide its workloads into different VMs, we assume a user requests a VM one slot by one slot. Hence, we model a VM hosted one time slot as a job. We differentiate the cloud service for the same type of jobs in terms of the responsive time. Let H denote the number of SLA levels. For SLA level h, the responsive time is bounded by T_h. We assume the VM requests are generated from users in R regions. Let $\mathcal{A}_{rvh}(t)$ denote the potential requests in region r for VM type v with SLA h that arrive at the beginning of time slot t. Let $A_{rvh}(t) = |\mathcal{A}_{rvh}(t)|$. We denote the price the federation cloud offers for VM type v with SLA h by $p_{vh}, 1 \leq h \leq H, 1 \leq v \leq V$. We assume the distribution of potential requests' valuation for the service satisfies the cumulative distribution function (CDF) $F_{vh}(u)$, u is potential requests' valuation. Hence, $[1 - F_{vh}(p_{vh})]$ portion of potential requests will request service. The actual request arrival rates are $A^c_{rvh}(t) = [1 - F_{vh}(p_{vh})] \cdot A_{rvh}(t)$. These requests may be served at different data centers.

1.2 The request routing model

Let $\alpha_{rvh}^{ki}(t)$ denote the portion of requests from region r for VM type v with SLA h routed to cloud provider k's data center i. Those requests are queued in a queue $\mathcal{Q}_{rvh}^{ki}$ in the federation cloud for service. Let $Q_{rvh}^{ki}(t)$ denote the queue backlog at time slot t. We denote the round trip delay for requests from region r to cloud provider k's data center i by d_r^{ki}.

1.3 Server operation model

The federation cloud can select a data center $D^{ki}, 1 \leq k \leq K, 1 \leq i \leq M_k$ among K cloud providers to serve a job. Let $N^{ki}(t)$ be the active servers at time slot t at data center i of cloud provider k. Each server of cloud provider k can run a combination of V types of VMs. We define the capacity region of a server as the convex hull of the set of available combination of V types of VMs that can be run on a server [1]. Let Λ^k be the capacity region of a server in cloud provider k. Then, the capacity region of data center D^{ki} is $\mathcal{C}^{ki} = N^{ki}(t) \cdot \Lambda^k$. Hence, at each time slot, the federation cloud control the number of activated servers $N^{ki}(t), 1 \leq k \leq K, 1 \leq i \leq M_k$ to adjust each data center's capacity region. Let $\mu_{rvh}^{ki}(t)$ denote the service rate that data center i of cloud provider k serve requests from region r for type v VM with SLA h. $(\sum_{r=1}^{R}\sum_{h=1}^{H}\mu_{r1h}^{ki}(t), \ldots, \sum_{r=1}^{R}\sum_{h=1}^{H}\mu_{rVh}^{ki}(t)) \in \mathcal{C}^{ki}(t)$. D^{ki} consumes a total power of $P^{ki}(N^{ki}(t))$.

1.4 The profit model

We consider the price for the power consumption at data centers. We ignore the one time investment. Let $c^{ki}(t)$ denote the price of power at data center i of cloud provider k at time slot t. The total power cost at time slot t is $CP(t) = \sum_{k=1}^{K}\sum_{i=1}^{M_k} c^{ki}(t) \cdot P^{ki}(N^{ki}(t))$. The revenue at time slot t is $R(t) = \sum_{h=1}^{H}\sum_{r=1}^{R}\sum_{v=1}^{V} A_{rvh}^{c}(t) \cdot p_{vh}$. The profit at time slot t is $P(t) = R(t) - CP(t) = \sum_{h=1}^{H}\sum_{r=1}^{R}\sum_{v=1}^{V} A_{rvh}^{c}(t) \cdot p_{vh} - \sum_{k=1}^{K}\sum_{i=1}^{M_k} c^{ki}(t) \cdot P^{ki}(N^{ki}(t))$.

1.5 The profit maximization problem

$Q_{rvh}^{ki}(t)$ is the queue backlog of requests from region r for VM type v with SLA s to data center i of cloud provider k queued at the federation cloud at time slot t. The request queue dynamics as follows:

$$Q_{rvh}^{ki}(t+1) = \max[Q_{rvh}^{ki}(t) - \mu_{rvh}^{ki}(t), 0] + \alpha_{rvh}^{ki}(t) \cdot A_{rvh}^{c}(t)$$

We omit the one-time investment when considering the profit of federation cloud. The time average profit of the federation cloud is:

$$\overline{P(t)} \triangleq \lim_{t\to\infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} E\{P(t)\}.$$

The profit maximization problem is formulated as follows:

$$
\begin{aligned}
\text{max: } & \overline{P(t)} \\
\text{subject to: } & N^{ki}(t) \le N^{ki}, 1 \le k \le K, 1 \le i \le M_k, \\
& \sum_{k=1}^{K}\sum_{i=1}^{M_k} \alpha_{rvh}^{ki}(t) = 1, \\
& (\sum_{r=1}^{R}\sum_{h=1}^{H} \mu_{r1h}^{ki}(t), \ldots, \sum_{r=1}^{R}\sum_{h=1}^{H} \mu_{rVh}^{ki}(t)) \in \mathcal{C}^{ki}(t) \\
& \text{Queues } Q_{rvh}^{ki}(t) \text{ are stable.}
\end{aligned}
$$

The profit maximization problem is for the federation cloud to choose appropriate price $p_{vh}(t)$, the request routing $\alpha_{rvh}^{ki}(t)$, the number of activated servers $N^{ki}(t)$, the service rate $\mu_{rvh}^{ki}(t)$.

1.6 The profit allocation among cloud providers

The above optimization problem is to achieve the maximum time average social welfare of the federation cloud, $\overline{P(t)}$. The next problem is how to allocate this profit to individual cloud providers participating in the federation. One natural method is to allocate the profit according to the jobs they complete. Hence, the profit allocated to cloud provider k is $P^k(t) = R^k(t) - CP^k(t) = \sum_{r=1}^{R}\sum_{v=1}^{V}\sum_{h=1}^{H}\sum_{i=1}^{M_k} \alpha_{rvh}^{ki}(t) \cdot A_{rvh}^{c}(t) \cdot p_{vh}(t) - \sum_{i=1}^{M_k} c^{ki}(t) \cdot P^{ki}(N^{ki}(t))$.

1.7 The fairness problem in profit allocation

In the social welfare maximization problem, the objective function is to maximize $\overline{P(t)}$, which is equal to maximize $\sum_{k=1}^{K} P^k(t)$ at every time slot. Whether this is a fairness profit allocation is a problem. In [2], the condition of proportional fairness is given.

To achieve the proportional fairness among different cloud providers, we need to change the objective function to [3]:

$$
\begin{aligned}
& \sum_{k=1}^{K} \ln P^k(t) \\
& = \sum_{k=1}^{K} \ln\{\sum_{r=1}^{R}\sum_{v=1}^{V}\sum_{h=1}^{H}\sum_{i=1}^{M_k} \alpha_{rvh}^{ki}(t) \cdot A_{rvh}^{c}(t) \cdot p_{vh}(t) - \sum_{i=1}^{M_k} c^{ki}(t) \cdot P^{ki}(N^{ki}(t))\}
\end{aligned}
$$

2 Bounded Response Delay

The federation cloud offers different SLA levels, i.e., different responsive time for requests. T_h denotes the responsive time for SLA h. This means the federation cloud needs to bound the queue delay for requests. For requests generated from region r for SLA h served by data center i of cloud provider k, after subtracting the round trip delay d_r^{ki}, the queue delay should be no more than $T_h - d_r^{ki}$. We apply the ϵ-persistent service queue technique.

For each queue $Q_{rvh}^{ki}(t)$, define a virtual queue $Z_{rvh}^{ki}(t)$ with initial backlog $Z_{rvh}^{ki}(0) = 0$, the queue update:

$$Z^{ki}_{rvh}(t+1) = \max[Z^{ki}_{rvh}(t) + 1_{Q^{ki}_{rvh}(t)>0}(\epsilon^{ki}_{rvh} - \mu^{ki}_{rvh}(t)) - 1_{Q^{ki}_{rvh}(t)=0}\mu_{max}, 0]$$

With $\mu^{ki}_{rvh}(t) \leq \mu_{max}$.

3 Dynamic Algorithm Design

We design a dynamic algorithm based on the Lyapunov optimization framework. Let $\Theta(t) = [Q(t), Z(t)]$ be the vector of all queues in the system. Define the Lyapunov function as:

$$L(\Theta(t)) = \frac{1}{2}[\sum_{r=1}^{R}\sum_{v=1}^{V}\sum_{h=1}^{H}\sum_{k=1}^{K}\sum_{i=1}^{M_k}(Q^{ki}_{rvh}(t)^2 + Z^{ki}_{rvh}(t)^2)]$$

The one-slot conditional Lyapunov drift is:

$$\begin{aligned}\Delta(\Theta(t)) = E\{L(\Theta(t+1)) - L(\Theta(t))|\Theta(t)\} = E\{\frac{1}{2}\sum_{r=1}^{R}\sum_{v=1}^{V}\sum_{h=1}^{H}\sum_{k=1}^{K}\sum_{i=1}^{M_k}[Q^{ki}_{rvh}(t)^2 + \mu^{ki}_{rvh}(t)^2 + \alpha^{ki}_{rvh}A^{c}_{rvh}(t)^2 \\ +2Q^{ki}_{rvh}(t)[\alpha^{ki}_{rs}A^{c}_{rvh}(t) - \mu^{ki}_{rvh}(t)] \\ +Z^{ki}_{rvh}(t)^2 + \{1_{Q^{ki}_{rvh}(t)>0}[\epsilon^{ki}_{rvh} - \mu^{ki}_{rvh}(t)] - 1_{Q^{ki}_{rvh}(t)=0}\mu_{max}\}^2 \\ +2Z^{ki}_{rvh}(t)[1_{Q^{ki}_{rvh}(t)>0}[\epsilon^{ki}_{rvh} - \mu^{ki}_{rvh}(t)] - 1_{Q^{ki}_{rvh}(t)=0}\mu_{max}]]\}\end{aligned}$$

The one-slot drift plus penalty is:

$$\Delta(\Theta(t)) - V[\sum_{h=1}^{H}\sum_{r=1}^{R}\sum_{v=1}^{V}A^{c}_{rvh}(t)\cdot p_{vh} - \sum_{k=1}^{K}\sum_{i=1}^{M_k}c^{ki}(t)\cdot P^{ki}(N^{ki}(t))]$$

4 Models in References

The models in references:

First, the work does not consider the pricing problem of cloud computing service. In this type of work, some [4] do not consider the specific virtual machines the workload runs in. Some [1] consider the specific virtual machine requests as the jobs.

[4] considers a "cloud workload factoring" game, a user's workload is modeled as a job with a size. The job's size is the computation time of the job when the computation rate is 1. The user can choose to move how many fraction of his job to cloud computing service.

[1] consider the requests for different types of VM as different jobs. And the time slots that VMs are hosted for is the job size. The queue backlog is measured by the total time slots that the VM requests ask for. Different servers can run different combinations of virtual machines due to the its configuration. The paper defines a concept of cloud's capacity, which is the set of feasible configurations for different types of VMs. Preemptive and Non-preemptive algorithms for VMs are given. The preemptive algorithm is the server-by-server MaxWeight allocation. The non-preemptive algorithm groups T time slots into a

super time slot, a myopic MaxWeight allocation is proposed and proved to support the job arrival rate λ, $(1+\epsilon)\frac{T}{T-S_{max}}\lambda \in \mathcal{C}$, $\mathcal{C}$ is the capacity region of the data center.

Second, the work considers the pricing problem of cloud computing service. This type of work models virtual machines as the products of cloud providers. The cloud providers allocate resources to virtual machines. These resources are the CPU, memory, storage of the cloud providers. A server can simultaneously run a limited number of virtual machines.

[5] considers the pricing problem in spot instance market, the auction of m types of virtual machines. The auction is executed in every round. This paper does not consider how much time the bidders occupy the virtual machines after winning the auction. This paper just specifies how many virtual machines a bidder needs in one round. This paper proposes a truthful auction.

[6] models the different types of spot instances as different products, whose prices are different and related to the demand according to the demand curve. This paper assumes cloud providers can do preempting. Hence, it can divide the length of an instance into separate time slots. This paper analyzes how to allocate the resources to different types of VMs to maximize the revenue.

[7] consider the bidding in spot market of cloud service. This paper models one type of virtual machines, which can be seen as one product of the cloud service.

References

[1] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters," in *Proc. of INFOCOM*, March 2012.

[2] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of Operational Research Society*, vol. 49, pp. 237–252, 1998.

[3] D. Bertsimas, V. F. Farias, and N. Trichakis, "A Characterization of the Efficiency-Fairness Tradeoff," July 2010.

[4] A. Nahir, A. Orda, and D. Raz, "Workload Factoring with the Cloud: A Game-Theoretic Perspective," in *Proc. of INFOCOM*, March 2012.

[5] Q. Wang, K. Ren, and X. Q. Meng, "When Cloud Meets eBay: Towards Effective Pricing for Cloud Computing," in *Proc. of IEEE INFOCOM*, March 2012.

[6] Q. Zhang, E. Gurses, R. Boutaba, and J. Xiao, "Dynamic Resource Allocation for Spot Markets in Clouds," in *Proc. of Hot'ICE*, 2011.

[7] Y. Song, M. Zafer, and K. W. Lee, "Optimal Bidding in Spot Instance Market," in *Proc. of IEEE INFOCOM*, March 2012.