Optimal Bidding in Spot Instance Market

Presenter: Jian Zhao

April 12, 2012

Background

- **Spot Instance Market**
  - Amazon introduced spot instance market on Dec. 2009 to utilize the idle resources of Amazon EC2 efficiently.
  - Price changes dynamically according to the supply and demand of cloud resources.
  - When the spot price falls below users' bids, the requests are accepted and users are charged by the spot price. When the spot price exceeds users' bids, the instance will be terminated.

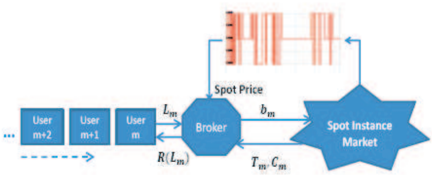
Background

- **Spot Instance Market**
  - Amazon introduced spot instance market on Dec. 2009 to utilize the idle resources of Amazon EC2 efficiently.
  - Price changes dynamically according to the supply and demand of cloud resources.
  - When the spot price falls below users' bids, the requests are accepted and users are charged by the spot price. When the spot price exceeds users' bids, the instance will be terminated.
- **Cloud Service Broker(CSB)**
  - An intermediation layer serves between cloud users and cloud infrastructure providers, e.g., RightScale.
  - Cloud Service Brokers can maximize their profit by leveraging the opportunistic spot instances.

Background

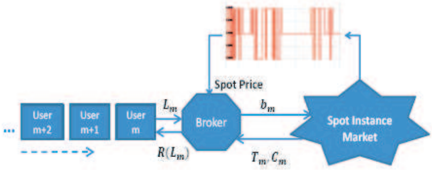
- **Spot Instance Market**
  - Amazon introduced spot instance market on Dec. 2009 to utilize the idle resources of Amazon EC2 efficiently.
  - Price changes dynamically according to the supply and demand of cloud resources.
  - When the spot price falls below users' bids, the requests are accepted and users are charged by the spot price. When the spot price exceeds users' bids, the instance will be terminated.
- **Cloud Service Broker(CSB)**
  - An intermediation layer serves between cloud users and cloud infrastructure providers, e.g., RightScale.
  - Cloud Service Brokers can maximize their profit by leveraging the opportunistic spot instances.
- **Cloud User**
  - Spot instances are suitable for compute-oriented, delay-tolerant job requests, e.g., Big Data analytics such as Hadoop, biological data processing, scientific batch computing.

How could cloud service broker maximize the time average profit?



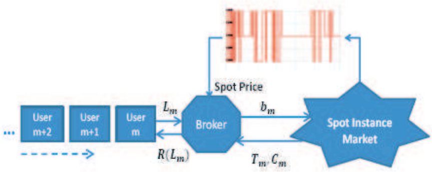
- **Model**
  - Size of the m-th job, $L^{MIN} \leq L_m \leq L^{MAX}, \forall m$;

How could cloud service broker maximize the time average profit?



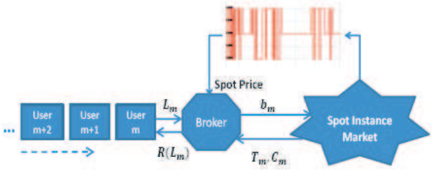
- **Model**
  - Size of the m-th job, $L^{MIN} \leq L_m \leq L^{MAX}, \forall m$;
  - Cloud service broker charges user m by $R(L_m)$, $R()$ is an arbitrary bounded function.

How could cloud service broker maximize the time average profit?



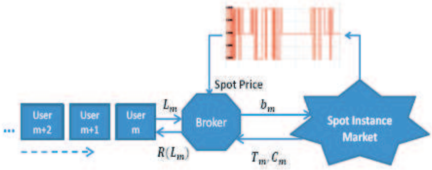
- **Model**
  - Size of the m-th job, $L^{MIN} \leq L_m \leq L^{MAX}, \forall m$;
  - Cloud service broker charges user m by $R(L_m)$, $R()$ is an arbitrary bounded function.
  - Based on the random job size L_m and spot price S_m, cloud service broker selects a bid b_m for spot instance

How could cloud service broker maximize the time average profit?



- **Model**
  - Size of the m-th job, $L^{MIN} \leq L_m \leq L^{MAX}, \forall m$;
  - Cloud service broker charges user m by $R(L_m)$, $R()$ is an arbitrary bounded function.
  - Based on the random job size L_m and spot price S_m, cloud service broker selects a bid b_m for spot instance
  - The cloud provider charges C_m for T_m time length of a job.

How could cloud service broker maximize the time average profit?



- **Model**
  - Size of the m-th job, $L^{MIN} \leq L_m \leq L^{MAX}, \forall m$;
  - Cloud service broker charges user m by $R(L_m)$, $R()$ is an arbitrary bounded function.
  - Based on the random job size L_m and spot price S_m, cloud service broker selects a bid b_m for spot instance
  - The cloud provider charges C_m for T_m time length of a job.
  - The profit per unit time in serving job m for cloud service broker is $(R(L_m) - C_m)/T_m$.

Profit Maximization Problem

CSB optimizes the time average profit

- The time average profit maximization problem:

$$\max \lim_{M\to\infty} \frac{\sum_{m=1}^{M}(R(L_m) - C_m)}{\sum_{m=1}^{M} T_m}$$

s.t. (average cost requirement.)

$$\lim_{M\to\infty} \frac{\sum_{m=1}^{M} C_m}{\sum_{m=1}^{M} T_m} \leq \alpha$$

and (average job computation rate requirement.)

$$\lim_{M\to\infty} \frac{\sum_{m=1}^{M} L_m}{\sum_{m=1}^{M} T_m} \geq \beta$$

Profit Aware Dynamic Bidding Algorithm

Apply Lyapunov optimization:

Apply Lyapunov optimization:

Construct two virtual queues for the two constraints respectively:

- virtual queue $Y(m)$ for cost requirement:

$$Y(m+1) = [Y(m) - \alpha T_m]^+ + C_m \qquad \lim_{M\to\infty} \frac{\sum_{m=1}^{M} C_m}{\sum_{m=1}^{M} T_m} \leq \alpha$$

Profit Aware Dynamic Bidding Algorithm

Apply Lyapunov optimization:

Construct two virtual queues for the two constraints respectively:

- virtual queue $Y(m)$ for cost requirement:

$$Y(m+1) = [Y(m) - \alpha T_m]^+ + C_m \qquad \lim_{M\to\infty} \frac{\sum_{m=1}^{M} C_m}{\sum_{m=1}^{M} T_m} \leq \alpha$$

- virtual queue $Z(m)$ for job computation rate requirement

$$Z(m+1) = [Z(m) - L_m]^+ + \beta T_m. \qquad \lim_{M\to\infty} \frac{\sum_{m=1}^{M} L_m}{\sum_{m=1}^{M} T_m} \geq \beta$$

Profit Aware Dynamic Bidding Algorithm

Apply Lyapunov optimization:

Construct two virtual queues for the two constraints respectively:

- virtual queue $Y(m)$ for cost requirement:

$$Y(m+1) = [Y(m) - \alpha T_m]^+ + C_m \qquad \lim_{M\to\infty} \frac{\sum_{m=1}^{M} C_m}{\sum_{m=1}^{M} T_m} \le \alpha$$

- virtual queue $Z(m)$ for job computation rate requirement

$$Z(m+1) = [Z(m) - L_m]^+ + \beta T_m. \qquad \lim_{M\to\infty} \frac{\sum_{m=1}^{M} L_m}{\sum_{m=1}^{M} T_m} \ge \beta$$

Define a counter for the time average profit before serving the m-th job.

$$\rho'(m) = \frac{\sum_{r=1}^{m-1}(R(L_r) - C_r)}{\sum_{r=1}^{m-1} T_r}, m = 2, 3, \ldots$$

Profit Aware Dynamic Bidding Algorithm

- **Algorithm:**
  For each job m, the CSB observes $(L_m, S_m, Y(m), Z(m))$, selects the bid $b_m^\star$ which maximize:

$$VR(L_m) + Z(m)L_m - (V + Y(m)) \cdot E(C_m(S_m, b_l, L_m)) - \\ (V\rho'(m) - \alpha Y(m) + \beta Z(m)) \cdot E(T_m(S_m, b_l, L_m))$$

  The intuition is to minimally upper-bound $\Delta - V\bar{P_m}$, Δ is the Lyapunov drift to measure the change of total queue backlogs.

Profit Aware Dynamic Bidding Algorithm

- **Algorithm:**
  For each job m, the CSB observes $(L_m, S_m, Y(m), Z(m))$, selects the bid $b_m^\star$ which maximize:

$$VR(L_m) + Z(m)L_m - (V + Y(m)) \cdot E(C_m(S_m, b_l, L_m)) - \\ (V\rho'(m) - \alpha Y(m) + \beta Z(m)) \cdot E(T_m(S_m, b_l, L_m))$$

  The intuition is to minimally upper-bound $\Delta - V\bar{P_m}$, Δ is the Lyapunov drift to measure the change of total queue backlogs.
- **Performance of Algorithm**

Profit Aware Dynamic Bidding Algorithm

- **Algorithm:**
  For each job m, the CSB observes $(L_m, S_m, Y(m), Z(m))$, selects the bid $b_m^\star$ which maximize:

$$VR(L_m) + Z(m)L_m - (V + Y(m)) \cdot E(C_m(S_m, b_l, L_m)) -$$
$$(V\rho'(m) - \alpha Y(m) + \beta Z(m)) \cdot E(T_m(S_m, b_l, L_m))$$

  The intuition is to minimally upper-bound $\Delta - V\bar{P_m}$, Δ is the Lyapunov drift to measure the change of total queue backlogs.

- **Performance of Algorithm**
  (a) Both the virtual queues are stable, hence, the two constraints are satisfied.

Profit Aware Dynamic Bidding Algorithm

- **Algorithm:**
  For each job m, the CSB observes $(L_m, S_m, Y(m), Z(m))$, selects the bid $b_m^\star$ which maximize:

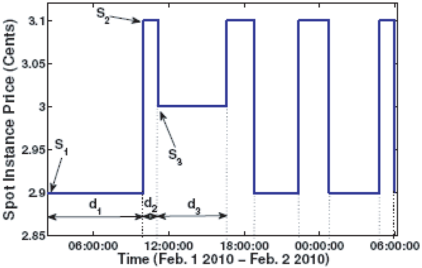
$$VR(L_m) + Z(m)L_m - (V + Y(m)) \cdot E(C_m(S_m, b_l, L_m)) - \\ (V\rho'(m) - \alpha Y(m) + \beta Z(m)) \cdot E(T_m(S_m, b_l, L_m))$$

  The intuition is to minimally upper-bound $\Delta - V\bar{P_m}$, Δ is the Lyapunov drift to measure the change of total queue backlogs.

- **Performance of Algorithm**
  (a) Both the virtual queues are stable, hence, the two constraints are satisfied.
  (b) $\rho' \geq \rho^\star - \frac{B}{VT^{MIN}}$, ρ' is the solution from PADB algorithm. $\rho^\star$ is the optimum solution.

Random Event Distribution: Spot Price and Job Size

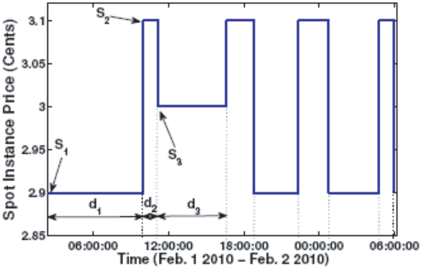
Spot Price Model:



- $(S_1, d_1) \rightarrow (S_2, d_2) \rightarrow \ldots \rightarrow (S_n, d_n)$. Use semi-markovian process to model the spot price.

Random Event Distribution: Spot Price and Job Size

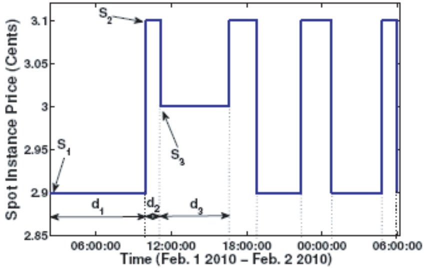
Spot Price Model:



- $(S_1, d_1) \rightarrow (S_2, d_2) \rightarrow \ldots \rightarrow (S_n, d_n)$. Use semi-markovian process to model the spot price.
- the state transition probability: $q_{i,j,k} = \Pr(S_{n+1} = s_j, S_n = s_i, \tau_n = k)$.

Random Event Distribution: Spot Price and Job Size

Spot Price Model:



- $(S_1, d_1) \to (S_2, d_2) \to \ldots \to (S_n, d_n)$. Use semi-markovian process to model the spot price.
- the state transition probability: $q_{i,j,k} = \Pr(S_{n+1} = s_j, S_n = s_i, \tau_n = k)$.
- Empirical Estimator for $q_{i,j,k}$: N_i is the number of occurrences of price $s_i \in \mathcal{S}$; $N_{i,j}^k$ is the number of transitions from price $s_i \in \mathcal{S}$ to $s_j \in \mathcal{S}$ with sojourn time of $k \in \mathcal{T}$.

$$\hat{q_{i,j,k}} = \frac{N_{i,j}^k}{N_i}.$$

Random Event Distribution: Spot Price and Job Size

Job Size: A bounded-pareto distribution: $L^{MIN} < L < L^{MAX}$. The probability density function:

$$f(x) = \frac{\sigma(L^{MIN})^{\sigma} x^{-\sigma-1}}{1 - (\frac{L^{MIN}}{L^{MAX}})^{\sigma}}.$$

Conclusions

- How the spot instance is priced is not discussed. It uses a semi-markovian model to capture the spot price characterization.