Cloud Pricing

Avinash Mohan and Rahul R

1 Motivation

Cloud computing is the realization of Internet based sharing of processing resources and data to devices on demand. It allows for an abstraction of physical infrastructure, which allows for large scale, efficient, cost effective deployment of on-demand configurable network access to a pool of resources such as servers, networks, applications and services. Most cloud computing platforms uses pricing as the inherent mechanism for resource allocation, to maximize available capacity utilisation as well as ensuring QoS/QoE guarantees for individual customers.

Cloud-computing services like Amazon EC2 utilizes an auction type pricing model called Spot pricing that allows users to bid for the unused capacity available in the system. In spot pricing, the spot price is dynamic and varies depending on current supply and demand fluctuations. At every time instant, the users specify the instance type, the number of computing instances they want to run, and bid on unused Amazon EC2 capacity. All those users with bids greater than the current spot price are then allowed to run their instances uninterrupted until the spot prices exceed their original bids, at which instant the interrupted subscribers can withdraw their jobs the from the cloud (non-persistent user) or can revise their bids (persistent user) for job completion. Auction mechanisms like Spot pricing are a first step towards market-pricing on a large scale computing resources based on supply and demand. However, Spot pricing cannot ensure guarantees on fair allocation of resources and truthful bidding.

Our goal in this research work is to formulate and investigate the problem of designing a Dominant-strategy incentive-compatible(DSIC) on-line auction mechanism/payment scheme for the cloud infrastructure. We aim to design pricing mechanisms, that stabilize all job queues, while providing subscribers with guarantees (commensurate with pricing) on expected time for job completions with low variance.

2 The pricing model

We consider a cloud service auction setting, where the service provider offer m types of instances. These instances are characterized by service requirements(eg., CPU/memory requirements, OS etc). We consider a slotted time system with discrete time slots $t \in \{1, 2, ...\}$. The cloud infrastructure is assumed to have fixed capacity, i.e., it can support a maximum of K_i instances of type i. We assume that the number of users at time instant t, is $N(t)$.

User valuations: Each user has a private valuation $v_i \in \mathbb{R}^+$ for each instance of type i. Thus the sum valuation of user j at any time instant, $V_j = \sum_{i=1}^{m} k_j^i v_j^i$ where, k_j^i denotes the number of units of instance type i desired by the j^{th} user. We will assume that there exists a valuation/preference distribution $\Phi \in \Delta(\mathbb{R}^{N(t)})$, i.e, $P(V_j \leq v) = \Phi(\infty, \ldots, \infty, v, \infty, \ldots, \infty)$. Φ is assumed to be common knowledge among all the users. At the beginning of each time instant, each user submits a total bid $B_j = \sum_{i=1}^{m} k_j^i b_j^i$, where, b_j^i denotes the j^{th} users bid for instance of type i. Since cloud services like Amazon EC2 services publish the spot-pricing history for upto 90 days, it is reasonable to assume that the user bids B_j is also a function of the entire past history of payment to the cloud provider.

In this work we will consider only the single-minded user, i.e, the user is willing to pay his valuations v_j^i and V_j of the desired bundle only if he receives the whole desired bundle; otherwise he is willing to pay 0 for any other bundle,i.e, $V_j = 0$. All the customers are assumed to be persistent with some deadline (maximum time for service completion) requirements.

Payment mechanism: Let P_j represent the amount of money bidder j is required to pay after each the completion of each auction. We assume the bidder is charged 0 if he loses the auction.

User's utility: Let $U_j = V_j - P_j$ represent the utility of player j, which he intends to maximize.

Revenue: The instantaneous revenue earned by the service provider is the sum of payments made by all the users ,i.e, $R(t) = \sum_{j=1}^{N(t)} P_j$. The service provider wishes to maximize his expected revenue, $R = \mathbb{E}[R(t)]$, where the expectation is taken over the payment scheme and the joint distribution of job arrival process and user valuations.

3 Problem Statement

Resource allocation problem: The auctioneer/service provider needs to choose a subset of bids from the submitted bids so as to maximize revenue as well as average capacity utilization, while ensuring that all the job queues remains stable. Moreover the choosing of bids must be done such that it is optimal for each user to bid his true valuations.

Payment: Having selected the users for service the auction mechanism should specify how much each user has to pay for his bundle. The payment mechanism should be dynamic and should reflect the current supply vs demand of the market, as well as the past history of bids.

QoE guarantees: Assuming a known prior joint distribution on job size and valuations, the auction mechanism should be able to provide Quality of Experience (QoE) guarantees on expected time taken for service completion as well as ensure job queue stability.

3.1 Justification

Existing works on mechanism design does not address the persistent users. In the case of persistent users with deadline requirements, it is possible that the user might be willing to place an instantaneous bid more than his private valuation to meet QoE requirement. Modelling such behaviour and providing QoE guarantees for the user has not been addressed yet. In research work focused on maximizing revenue to the cloud service provider, existing work ignores the strategic aspects of user bidding and assumes i.i.d. distribution of users private valuations. This ignores the dynamic nature of the auction mechanism which we aim to capture with our model.

4 Plan

In the next one month and a half, we aim to formulate the model fully and to find provably DSIC mechanism for auction pricing. We also aim to incorporate different pricing mechanisms for the on-demand service allocation problem and do a comparative study.