

Datacenter Marketing System

Rui Zhou

Shu Zhang

Final Project - CSCI 2950-u - Spring 2013

Abstract

In this report we present the auction-based marketing system we build for datacenter applications. Motivated by similar mechanism of auctions for resource in other computer systems, we aim to let users bid their network resources, mainly bandwidth and latency of their desired flows. Different from existing bidding systems like Google's marketing system[4], our system is based on SDN network and utilizes the advantage of the central controller and global network information. Another thing we focus on is the allocation algorithm for the auctioneer. We investigated some algorithms and evaluated their performances.

1 Introduction

Resource allocation is an important topic in system related area, especially in data center where network resources heavily influence the performance of the application running on the datacenter. Since the network resource is limited and application expense is infinite, an idea of launching auctions on datacenters for applications to bid resources comes to existence.

2 Related Work

In this section, we investigate existing paradigms of data centers (mainly private datacenters and public datacenters) and their marketing and pricing methods. Also, we introduce the existing techniques of SDN which is the base of our system.

2.1 Datacenter Markets

There are generally two types of data centers, one is for-profit, public cloud datacenters, such as Amazon Web Services[1], the other is private cloud datacenters, like those operated by Google or Facebook. Although auction mechanisms could be applied to both, different scenarios give diverse answers because of their business models. The following two paragraphs talk about the assumed behavior for the two types of data centers in this paper and our discussion are based upon these behaviors.

In a public cloud service, bidders are customers who use the cloud service. Take Amazon EC2 for example[2], Amazon revises prices for instances for all kinds of types. Three basic types are On-Demand Instances, Reserved Instances, which allows user to reserve a period of time for future uses and Spot Instances[3] whose values fluctuate and allow customers to bid on them. Along this path, Amazon created other services like data transfer, elastic IP addresses, elastic load balancing, etc and makes pricing policies for them. From Amazon's view, its pricing policies are so carefully revised so that they could maximize its profits. If a customer wants to bid for Amazon's resources, its knowledge is pretty limited, and Amazon hides the data center details from customers and only expose pricing history as hints for users to bid, and the budget of buyers are real currency.

In a private datacenter running inside IT companies, the customers are internal teams of companies who use the datacenter resources for the sake of their team and the companies. The operator's goal might be minimizing the cost while meeting business objectives, or may be efficient use of network resources. The users (teams), according to their priorities, are allocated different amounts of budget, which usually is virtual currency. Teams with more budget might use the datacenter resources in a higher share. The private datacenter might perform as a autonomy market, whose prices of goods are totally determined by the market or self-adapted by the system. Bidders, with higher possibility, will be provided more detailed view of the datacenter conditions, such as dynamic traffic between servers or racks and static network topology, so that they could formulate their bidding policies more sensible.

2.2 Existing Pricing Mechanisms

We found that the value of goods which are to be traded in the market is the core issue when letting a market run. People may argue that in an auction, people could win the bidding by giving the highest amount of money per unit of resource. But sometimes it might be far more complex. Prices, in some cases, are the reflection of the condition of the market, so it is better to set a baseline price for resources so that people who bid lower than that baseline will never win the auction, even if they have the highest price. But in some cases, prices do need not to be considered and the simple first-price or second-price auctions policy could perform well. Also, complex cases like combinatorial pricing[5] is also a must-to-think issue. Also, The pricing methods for auctions in private and public datacenters are pretty different. This is because the goals of markets in either datacenters are different.

2.2.1 Public Cloud Pricing

Amazon[1] has set models for pricing in public cloud. AWS supports buy-at-once pricing for on-demand instances as well as auctions for resources for spot instances. For on-demand services, prices for instances are relatively higher and the good thing is that they are guaranteed to be provided to the buers. For Spot instances[3], Amazon has set the base prices for instances. The prices are much lower (\$0.007 per hour versus \$0.060 per hour for on-demand instances). The prices are given by Amazon according to their costs and virtual equipments with different capacities will have different prices. Resources are provided in a combinatorial way, since it is meaningless to bid a single CPU without any memory or storage. But in some scenarios resources are not sold in a combinatorial way, such as Elastic Load Balancing which only charges by the amount of data to be processed by the load balancer.

2.2.2 Private Cloud Pricing

Unlike AWS, prices per unit of resources in the private cloud datacenters probably will change in a dynamic way. Systems like D'Agents[7] and Google's Planet-wide Clusters[4] has adopted dynamic pricing and price of resources are updated periodically. Both the aforementioned systems uses dynamic prices to reflect the changing demand-supply relation. For dynamic pricing, the prices are the minimum payment. But in order to maximize the utility of total resources, the sealed-bid second-price auctions [8] could be used so as to guarantee that as long as there are more than one bidders in a certain auction round, there will always be a winner. The dynamic minimum prices in this case, serve as an indication for bidders to price their bid requests. The prices in [4] could change in an self-adaptive way. In short, 'hot' resources will be more expensive and a function ($g(x,p)$) calculates the price increment after each auction round, basing on the exceeding of demands over supplies. Interestingly, AuYoung, et al [6] discovered that auctions might take some time as bidder's patience falls down when waiting for gaining the resource. So there should be another 'buy-it-now' pricing mechanism provided for users who don't want to wait. The buy-it-now pricing short-

circuits the price discovery process of the auction, and the price is determined by a historical function which takes the historical auction/buy-it-now prices for parameters.

2.2.3 OpenFlow

2.2.4 Floodlight

2.3 Software Defined Network

2.3.1 OpenFlow

2.3.2 Floodlight

3 System Design and Algorithm

3.1 Precondition

4 Implementation

5 Evaluation

6 Conclusions

It should be easy to write your report in LaTeX, and it's a great tool to learn. It almost certainly came with your Linux installation, and can be very easily installed in Cygwin and on the Mac (through the excellent MacTeX distribution).

References

- [1] Amazon Web Services: <http://aws.amazon.com>.
- [2] Amazon EC2 Pricing: <http://aws.amazon.com/ec2/pricing>.
- [3] EC2 Spot Instances: <http://aws.amazon.com/ec2/spot-instances>.
- [4] Stokely, Murray, et al. "Using a market economy to provision compute resources across planet-wide clusters." Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on. IEEE, 2009.
- [5] Nisan, Noam. "Bidding and allocation in combinatorial auctions." Proceedings of the 2nd ACM conference on Electronic commerce. ACM, 2000.
- [6] AuYoung, Alvin, et al. Practical market-based resource allocation. University of California, San Diego, 2010.
- [7] Bredin, Jonathan, David Kotz, and Daniela Rus. "Market-based resource control for mobile agents." Proceedings of the second international conference on Autonomous agents. ACM, 1998.
- [8] Vickrey, William. "Counterspeculation, auctions, and competitive sealed tenders." The Journal of finance 16.1 (1961): 8-37.

- [9] Ausubel, Lawrence M., and Peter Cramton. "Auctioning many divisible goods." *Journal of the European Economic Association* 2.23 (2004): 480-493.
- [10] Cramton, Peter, and Lawrence M. Ausubel. "The clock-proxy auction: A practical combinatorial auction design." (2006): 115-138.