Efficiently Rationing Resources for Grid and P2P Computing *

Ming Chen, Yongwei Wu, Guangwen Yang, Xuezheng Liu
cm01@mails.tsinghua.edu.cn, wuyw@tsinghua.edu.cn,
ygw@mail.tsinghua.edu.cn, xuezhengliu00@mails.tsinghua.edu.cn

Dept. of Computer Science and Technology, Tsinghua University

1 Introduction

As Grid and P2P computing become more and more popular, many schedule algorithms based on economics rather than traditional pure computing theory have been proposed. Such algorithms mainly concern balancing resource supply and resource demand via economic measures. As we know, fairness and efficiency are two conflicting goals. In this paper, we argue that overbooking resources can greatly improve usage rates of resources and simultaneously reduce responsive time of tasks by shortening schedule time especially under extreme overload, while maintaining schedule principles. This is accomplished by scheduling more eligible tasks above resource capacity in advance, therefore overbooking the resource. We verify our claim on Grid Market[1].

2 Enhance Grid Market with Overbooking

2.1 Grid Market[1]

There are two types of participants in the market: resource suppliers and resource consumers. Suppliers compete to sell resources while consumers contend to buy resources. There exists a market transaction center where all orders from suppliers and consumers are matched based on order type and match algorithm. Running pricing algorithms, software agents on behalf of suppliers and consumers automatically post orders.

Resource market periodically uses price-driven continuous double auction process to match consumers' bid orders (buy orders) and suppliers' ask orders (sell orders). Bid orders and ask orders can be submitted at anytime during the trading duration. At the end of a match period, if there are open bids and asks that match or are compatible in terms of price and requirements, a trade is executed. Bids are ranked from highest to lowest according to bid prices while asks are sorted from lowest price to highest price. The match process starts from

* This project is supported by the National Natural Science Foundation of China under Grant No. 60373004, No. 60373005, and No. 60273007, and by the National High Technology Development Program of China under Grant No. 2002AA104580

the beginning of ranked bids and asks. If prices are equal, match priorities are based on the principle of time first and quantity first.

Grid Market proposed two pricing algorithms: consumer pricing algorithm and supplier pricing algorithm. The consumer pricing function is: $P_{bid}(t) = \alpha + \beta \Delta t$, where α, denoting base price, and β, expressing price elasticity, are consumer-specific coefficients and t is the time parameter. The supplier function is: $P_{asked}(t) = \alpha - \beta \Delta t$, where α, denoting base price, and β, expressing price elasticity, are supplier-specific coefficients and t is the time factor. These two functions automatically make temporal differences between bid prices and ask prices to converge to clear the market.

2.2 Overbooking

Consider such a scenario in a basic Grid Market. A resource is released by a consumer and is open to compete by potential consumers. If the resource's sell price is higher than any buy prices posted by consumers, the resource is inevitably idle until ask price and bid price gravitate towards each other and meet or cross finally. The wasted negotiation time may be negligible in most light-loaded cases, but it may cause severe service bottleneck in heavy load situation. And the existing of idle time decreases resource utility rates and prolongs task's responsive time, which is the sum of negotiation time and service time.

As a widely used technology, overbooking can improve resource utility rates. The basic idea of overbooking goes as follows. A resource can be assigned loads more than its stated capacity temporarily and, using priority-based algorithm, serves them at its best. So, the resource works in high gear and no resource is squandered. Nevertheless, in the long-term the resource's capability should not be overwhelmed by average loads to maintain service quality.

Therefore, we introduce overbooking into Grid Market. A resource keeps a waiting queue in which successfully bidders for it are lined up. A new winning consumer will be queued in when a consumer frees the entire/partial resource. The length of waiting queue depends on three parameters. The first is the match interval. A relative short waiting queue is enough to preserve utility usage rates for frequenter matches relative to service time, which can effectively shorten necessitated negotiation time. The second is the number of a resource's servants, which determines the number of consumers that can simultaneously served by the resource. The number of candidates in the queue must be as many as, or more than the number of servants of corresponding resource. Otherwise, until succeeding consumers successfully win out, fractional resource released by served consumers may be unoccupied. The third is the ratio of negotiation time to service time. The higher the ratio is, the longer the waiting queue should be to ensure the pipeline's feeding. While service time is relatively fixed, negotiation time relies on several factors in market wide scope: consumers' starting prices, elasticities, and ceilings, and suppliers' starting prices, elasticities, and floors. In common practices, we expect that the ratio of the waiting queue's length to the number of servants of a resource should predominately account for the efficiencies and we take 1/1, say the length of the queue is equal to the number of corresponding servants, as a rough estimation for the suitable length. Adaptation policy can be applied here.

3 Analysis and Simulation

3.1 Analysis

The system is modelled as a M/M/N queuing network[6]. Task streams of all consumers are bound into a single task stream as system input stream. We employs below equations[6] to theoretically analyze the resource utilization rate and responsive time of our system: $\rho = \frac{\lambda N_{consumers}}{\mu N_{suppliers}}$, where $N_{consumers}$ and $N_{suppliers}$ are the number of consumers and suppliers separatively and ρ is the system resource usage rate. $K = (\sum_{i=0}^{N-1} \frac{(N\rho)^i}{i!})/(\sum_{i=0}^{N} \frac{(N\rho)^i}{i!})$, $C = \frac{1-K}{1-\rho K}$. System responsive time is: $T_{rst} = \frac{C}{N} \frac{\mu}{(1-\rho)} + 1/\mu$.

3.2 Simulation

We use a event-driven prototype to explore the schedule efficiency of this algorithm in aspects of task responsive time and resource utilization rate varying elasticity coefficients, which should be the primary determinant factor in common settings (Figure 1 and Figure 2). To emphasize the algorithm's performance in bad settings, we set match interval to 2 time units, which greatly protracts the negotiation duration.

First, we can see from figures that our schedule algorithm is highly efficient: the theoretical curves (plotted according to ρ and T_{rst} respectively) are almost approximated by experiment curves when system's load is not high. Second, without overbooking, time burden due to *bargaining* between consumers and suppliers increases sharply as system approaches saturation and the degree of increased burden is negatively related to elasticity coefficients. The reason behind it is straightforward: bargaining time costs are neglectable relative to *'long'* arrival interval when system load is light, but it does matter in high load cases. These costs reduce resource utilization rates and increase responsive time. Third, overbooking greatly improves the algorithm's performance especially in high load environment. With the help of overbooking, responsive time curve and utility utilization rate curve all draw near their theoretical curves respectively. Finally, the heavier the time burden incurred by participants' parameters is, the more efficient the overbooking is.

4 Related Work

Spawn[2] employs Vickrey Auction[3]—second-price sealed auction—to allocate resources among bidders. Bidders receive periodical funding and use balance of fund to bid for hierarchical resources. Task-farming master program spans and

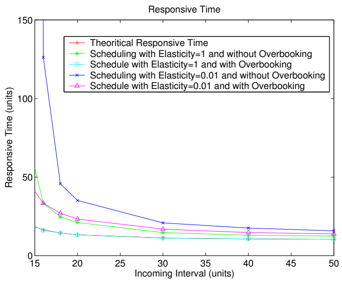

Fig. 1. Responsive Time (1 consumer vs. 1 supplier): $\alpha_{consumer} = 1$, $\alpha_{supplier} = 10$; $Ceiling_{consumer} = 7$, $Floor_{supplier} = 4$; $\mu = 1/10$

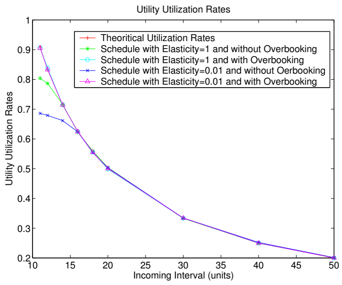

Fig. 2. Usage Rates (1 consumer vs. 1 supplier): $\alpha_{consumer} = 1$, $\alpha_{supplier} = 10$; $Ceiling_{consumer} = 7$, $Floor_{supplier} = 4$; $\mu = 1/10$

withdraws subtasks depending on its relative balance to its counterparts. Monte Carlo simulation applications are its main targets. Rexec/Anemone[4] implements proportional resource sharing in clusters. Users assign utility value to their applications and system allocates resources proportionally. Cost requirement is not its consideration. In JaWS (Java Web-computing System)[5], machines are assigned to applications via auction process in which highest bidder wins out. These above solutions don't make use of continuous double auction.

5 Conclusion

With the pervasion of Grid and P2P computing, arises a critical problem, efficiently and fairly allocating resources especially under extreme overload. In this paper, we contend that overbooking resources can greatly improve usage rates without disobeying algorithm-specific scheduling principle. Simulation results conducted on Grid Market enhanced by overbooking testify these claims.

References

1. Ming Chen, Guangwen Yang, Xuezheng Liu, *Gridmarket: A Practical, Efficient Market Balancing Resource for Grid and P2P Computing*, Grid and Cooperative Computing 2003, Shanghai, 2003.
2. Hogg, B. Huberman, J. Kephart, and W. Stornetta, *Spawn: A Distributed Computational Economy*, IEEE Transactions on Software Engineering, February 1992.
3. W. Vickrey, *Counter-speculation, auctions, and competitive sealed tenders*, Journal of Finance, Vol. 16, No. 1, pp. 9-37, March 1961.
4. B. Chun and D. Culler, *Market-based proportional resource sharing for clusters*, Technical Report CSD-1092, University of California, Berkeley, USA, Jan. 2000.
5. S. Lalis and A. Karipidis, *An Open Market-Based Framework for Distributed Computing over the Internet*, Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID 2000), Dec. 17, 2000, Bangalore, India.
6. Hock N C. *Queuing Modelling Fundamentals*, John Wiley & Sons Ltd., 1997