

Maximizing Social Welfare in Cloud Computing

Nahum Shimkin





Department of Electrical Engineering, Technion

Joint work with

Ishai Menache (MIT & Microsoft Research), Asu Ozdaglar (MIT)

2nd Israeli-Dutch Workshop on Queueing Theory
EURANDOM, September 2010

Cloud Computing Background

- **Cloud computing** has been gaining prominence recently, with the advent of major systems from Amazon, IBM, Google, Microsoft, and many others.
- The cloud computing paradigm allows on-demand network access to shared pools of computing resources: virtual servers, applications and software.
- Benefits for users include:
 -  economy of scale, efficient use of resources
 -  central & transparent management
 -  “utility computing”: resources on demand, pay per use
 -  no infrastructure investment

Major Modalities of Cloud Computing

- **Infrastructure as a Service (IaaS):**
Offers access to virtual servers and storage on-demand
- **Software as a Service (SaaS):**
Offers specific applications and development tools that run on the cloud.

IaaS Example: Amazon EC2 (Elastic Compute Cloud)



Purchasing options:

- On-demand Instances (pay per use)
- Reserved Instanced (yearly subscription + low usage cost)
- Spot Instances (bid for unused capacity)

| Standard On-Demand Instances | Linux/UNIX Usage | Windows Usage |
|---------------------------------|------------------|-----------------|
| Small (Default) | \$0.095 per hour | \$0.13 per hour |
| Large | \$0.38 per hour | \$0.52 per hour |
| Extra Large | \$0.76 per hour | \$1.04 per hour |
| High-Memory On-Demand Instances | Linux/UNIX Usage | Windows Usage |
| Double Extra Large | \$1.34 per hour | \$1.58 per hour |
| Quadruple Extra Large | \$2.68 per hour | \$3.16 per hour |
| High-CPU On-Demand Instances | Linux/UNIX Usage | Windows Usage |
| Medium | \$0.19 per hour | \$0.31 per hour |
| Extra Large | \$0.76 per hour | \$1.24 per hour |

Pricing is per instance-hour consumed for each instance type, from the time an instance is launched until it is terminated.

System Management Goals

- Possible objectives for system management:
 1. Revenue maximization *(profit-centric)*
 2. Optimizing performance metrics *(system-centric)*
 3. Maximizing social welfare *(user-centric)*
- Our focus here will be on **social welfare maximization**.
- Social optimality is especially relevant for:
 -  Cloud computing offered as a public service
 -  In-house clouds computing (private clouds)

Market-based Resource Management

- A variety of economic models and market-based resource management schemes have been devised over the years, both in the context of performance and QoS optimization, as well as in the context of social welfare maximization.
- Early work, related to parallel and distributed systems, includes:
 - Ferguson, Yemini & Nikolau '88: auction-based load balancing
 - Kurose & Simha, '89: bid-based resource allocation.

Market-based Resource Management (2)

A sample of more recent work in the context of grid and cluster management systems, includes the following economic mechanisms:

- **Commodity markets** (fixed or variant prices)
[MOSIX, '00; G-Commerce, '01; Nimrod/G, '02; Libra, '04]
- **Bargaining**
[CATNET '03; Ocean '03]
- **Auctions**
[WALRAS, '93; Gridmarket, '03; Bellagio, '04; Tycoon '04]
- **Bid-based proportional resource sharing**
[D'Agents, '98; REXEC, '00; Chun & Kuller, '00]

Bid-based Mechanisms

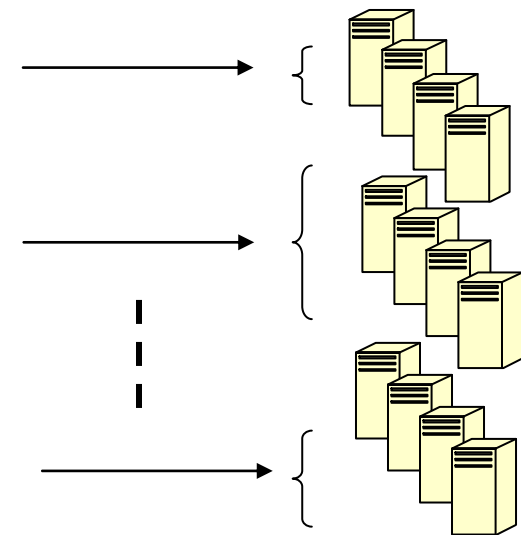
- Bid based mechanisms have attracted much attention in the context of rate control in communication networks, following the influential work of Kelly and co-workers ['97,'98].
- Here, an entire resource (bandwidth) is divided among the users, in proportion to their bids. This leads to optimization of a social cost function. This mechanism can also be viewed as *adaptive, congestion dependent pricing*.
- Yolken & Bambos ('09) consider a bid-based capacity allocation mechanism, in a utility computing context. In their formulation users submit stationary streams of jobs, and are allocated capacity proportional to their bids with no resource sharing.
- In the present work we focus on *fixed pricing* schemes, closer to the classical economic theory of competitive equilibrium. Further, we identify each user with a single job (or application), and emphasize aspects of resource-dependent computation times,

Outline

- System & user model
- The user-optimization problem
- Optimal social welfare
- Socially optimal pricing
- Economic context
- On-line price adjustment
- Finite resources
- Some relations to profit maximization

1. The Basic Model

- We consider a large computing facility that offers computing resources to incoming users, each with his own job.
- Existing jobs are executed *concurrently*.
- Resources allocated to each job are determined based on some known mechanism.
- The service (execution) time of each job generally depends on the resources allocated to it.



Arrivals

- Potential users (and their jobs) arrive as a Poisson process with rate Λ .
- Users may differ in their types, which define their service requirements and personal preferences.
- Let $i \in I$ denote the type parameter. The set of types is allowed to be continuous.
- The type of each arriving user is drawn randomly according to a probability distribution on I with density $f_I(i)$. Thus, potential arrivals of different types are independent Poisson Processes with rates distributed according to $\bar{\lambda}(i) = \Lambda f_I(i)$.
- As users may balk upon arrivals, the *actual* arrival rates will be smaller. The respective rate distribution is denoted $\lambda(i)$.

Resources and Service Durations

- Arriving users of type i are allocated a certain amount of resources, represented by a real number $z_i > 0$.
- The job execution time may depend z_i . Let $T_i(z)$ denote the mean execution time for type i jobs, given resources z .

Note:

- The dependence of the service duration on the applied resources should be especially significant for batch-type jobs, such as scientific computing and business analytics, that can be effectively parallelized.
- Our model is especially geared towards such applications, However, it also supports fixed-durations applications, such as customer service over the cloud.

Service Duration Model

- A reasonable model for service time scaling is given by:

$$T_i(z) = \frac{D_i}{z} + a_i \quad (a_i > 0, D_i \geq 0)$$

- This coincides with *Amdhal's law*, often used in the parallel processing literature.
- More generally, we impose the following:

Assumption 1: Each $T_i(z)$ is a (weakly) *convex decreasing* function of z , with $T_i(\infty) > 0$.

Pricing

- We consider here simple per-unit pricing:

$$\text{Charge} = P z \tau, \quad \text{where } \tau = \text{actually service time}$$

- Therefore, for a type- i user that employs z resources,

$$\text{Expected charge} = P z T_i(z)$$

Individual Utilities

- The utility of a balking user is set to 0 by default.
- The utility function of a served user takes the form

$$U_i(z) = V_i(z) - PzT_i(z)$$

where $V_i(z)$ is the user's *value of service* with resources z .

- *Example:* For delay-sensitive applications,

$$V_i(z) = V_i - c_i(T_i(z))$$

- *Assumption 2:* $V_i(z)$ is strictly concave increasing.

An Additional Assumption

- *Assumption 3:* For each i ,

$$\frac{V_i(z)'}{(zT_i(z))'}, \text{ is strictly decreasing in } z.$$

This property can be seen to hold in the following cases:

- $zT_i(z)$ is a convex function of z .
E.g., for $T_i(z) = a + D/z$ (Lindhal's law), is linear.
- Delay-dependent value:: $V_i(z) = V_i - c_i(T_i(z))$, with c_i a convex function.

Steady-State System Load

- Consider the system in steady state, with given effective arrivals rates $\lambda(i)di$ and resource allocations (z_i) .
- We suppose here that *there are sufficient resources to accommodate all arrivals*. Hence, each job type i effectively forms an independent M/G/ ∞ queue.
- With arrival rate $\lambda(i)$ and mean service time $T_i(z_i)$, the expected number of type- i jobs in the system at steady-state is obtained by Little's law:

$$N(i) = \lambda(i)T_i(z_i)$$

- Therefore, the expected resource occupancy is given by:

$$Y = \int_i N(i)z_i di = \int_i \lambda(i)T_i(z_i)z_i di$$

We refer to Y as the *system load*.

Operating Costs

- The system operating costs increase with the load on the system. We suppose the operating costs can be represented (at least approximately) as a function of the average load:

$$C_{\text{oper}} = C_0(Y)$$

-
- *Assumption 4:* C_0 is a strictly convex increasing function of Y (at least beyond some value of the load Y).
- *Remark:* Depending on the problem and time scale considered, C_0 can also embody the amortized investment in infrastructure required to support the load Y (possible under some QoS requirements).

2. Individual Optimality

- Given the price P , the utility maximization problem faced by an arriving user of type i is

$$\max\{0, \max_{z \geq 0} U_i(z)\}$$

where

$$U_i(z) = V_i(z) - PzT_i(z)$$

- Proposition 1:*

The optimization problem $\max_{z \geq 0} U_i(z)$ has a unique solution z_i^* .

Individual Optimality (cont'd)

- *Proof (essence):* The first-order condition for an internal solution is $U_i(z)$, or equivalently $\frac{V_i(z)'}{(zT_i(z))'} = P$. But, by assumption, the LHS is continuous (existence) and strictly decreasing (uniqueness). ■
- The decision procedure for this user is then clear:
 - ✚ Solve for z_i^* , and compute $U_i(z_i^*)$.
 - ✚ If $U_i(z_i^*) < 0$, balk. Else, if $U_i(z_i^*) > 0$, enter service with resources z_i^* .
 - ✚ For concreteness, in the neutral case $U_i(z_i^*) = 0$ we choose to enter.

3. Social Welfare

- The social welfare (or social utility) function is given by:

$$W = \int_i V_i(z_i) \lambda(i) di - C_0(Y)$$

where $Y = \int_i \lambda(i) T_i(z_i) z_i di$. The decision variables are the effective arrival rate $\lambda(i) \in [0, \bar{\lambda}(i)]$ and resource allocations $z_i \geq 0$.

- To determine the best possible social welfare, we consider an omniscient central controller, who knows the user characteristics and preferences, and can set their choices accordingly.

Optimal Social Welfare

- Under some additional regularity conditions on the type distribution, we obtain the following result.
- *Theorem 2:* There exists an optimal solution $\{\lambda^*(i), z^*(i)\}$ to the social welfare maximization problem. A particular solution is uniquely defined by the following set of first-order conditions:

$$z^*(i) \in \arg \max_{z \geq 0} U_i(z, \rho),$$

$$U_i(z, \rho) \triangleq V_i(z) - \rho z T_i(z), \quad \rho = C'_0(Y^*)$$

$$\lambda^*(i) = \bar{\lambda}(i) \text{ if } U_i(z^*(i), \rho) \geq 0, \text{ else } \lambda^*(i) = 0$$

- *Remark:* The above-mentioned regularity conditions are meant to ensure that $U_i(z^*(i), \rho) = 0$ cannot hold simultaneously for a set of types of positive measure.

- *Proof idea:* The stated conditions are essentially the KKT conditions for the optimization problem, which are *necessary* conditions for optimality here (note that the problem is not convex jointly convex in $\{\lambda(i), z(i)\}$, nor is it convex in $\{z(i)\}$ alone unless $zT_i(z)$ is convex). We next show that the load Y^* is monotone decreasing in ρ , hence $\rho = C'_0(Y^*)$ has a single solution. This implies that there exists (essentially) a single stationary point, which must therefore be a global extremum (a maximum in this case).

4. Socially Optimal Pricing

- Consider now the system under individually-optimal decisions of the users, with a fixed per-unit price P . The following is our central result.
- *Theorem 3:* There exists a unique price P_0 for which the individually optimal solution maximizes the social welfare. P_0 is the unique solution to the equation $P = C'_0(Y(P))$.
- *Proof idea:* Using $P = C'_0(Y^*)$ from Proposition 2, it is easily seen the optimality conditions of that Proposition and the equations of individual optimality coincide. By uniqueness, the two solutions coincide.

5. Economic Context

- The observation that pricing induces social welfare maximization is a classical one in the economic literature.
- The classical formulation of the problem assumes a finite set of buyers, $\{1, \dots, n\}$ is a static setting, where each chooses the amount x_i of goods to buy so as to maximize the personal utility $V_i(x_i) - Px_i$. The social welfare is $\sum_i V_i(x_i) - C(\sum_i x_i)$.
- The model proposed here examines more closely the effects of user dynamics, and in particular the effect of resource-dependent service times.

Economic Context (2)

- Comparing the forms of this mathematical program to ours, it may be seen that $z_i T_i$ plays an analogous role to x_i . This points to the fact that the actual good being sold here is resource-hours. However, T_i itself does have an important independent role in the dynamic context.
- Another important observation is the independent role of balking (or effective arrival rate) in our model. The classical model assumes by default that $V_i(0) = 0$ (with V_i continuous), so that a decision of not participating is implied by the choice of $x_i = 0$). The situation is different in our case, as the choice of participating with $z_i = 0$ would lead to infinite computation time. Therefore, an independent set of decision variables must be considered to allow for non-participation, or balking.

6. Price Adjustment

- The implementation of the fixed pricing scheme does not require any private information on specific users. However, computation of the optimal price does require a detailed model of the user population.
- On-line price adjustment mechanisms may then be used to advantage. Such mechanisms have been well studied in the economic literature, and are often referred to as tatonnement processes. We briefly mention here a few variations.
- The basic goal is to iteratively approach the solution of the equation $P = C'_0(Y(P))$ that defines the socially-optimal price.

Price Adjustment (2)

- Continuous-time tatonnement gives in this case:

$$\frac{d}{dt} P(t) = \alpha \left(C'_0(Y(P(t))) - P(t) \right)$$

- The load Y is assumed to reach the steady state conditions (corresponding to $P(t)$ at every time instant.
- Convergence follows trivially from monotonicity, as the price is scalar here.
- A discrete-time version of this adjustment process reads

$$P_{k+1} = P_k + \alpha \left(C'_0(Y(P_k)) - P_k \right)$$

This again converges provided that the gain is small enough.

Price Adjustment (3)

- Stochastic-approximation type process can be devised to take account of the noise in the measurement of Y due to stochastic and transient effects. For example, consider

$$P_{k+1} = P_k + \alpha_k \left(C'_0(\hat{Y}_k) - P_k \right)$$

$$\text{where } \hat{Y}_k = \frac{1}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} Z(t) dt$$

- Under appropriate conditions this can be shown to follow the ODE described before, thereby both establishing convergence and formalizing the relevance of the continuous time process.

7. Finite Resources

- Actual systems are of course limited in their resources. This can lead to blocking, service denial and customer loss if the system is operated without sufficient margin. So far, this effect was not part of our model.
- A detailed incorporation of blocking and loss effects in the utility functions of our economic model appears hard for several reasons.
 - ✚ User behavior modeling: Blocked users have several options, including going elsewhere, waiting online, or coming back later. An exact model and its analysis appear complicated.
 - ✚ The economic effects of blocking and service denial are not only the direct ones, but also indirect such as loss of reputation. It is hard to quantify these in the utility equation.
 - ✚ System capacity is often planned with a view to congestion period, rather than to the normal (long-term average) operating conditions, on which our model is focused.

Finite Resources: Average Load Constraints

- We therefore propose the following modification to our model.
- Start by deriving an upper bound Y_{\max} on the allowed average load on the system. This can be derived, for example, to satisfy upper bounds on allowed blocking, using multiclass Erlang loss models.
- Add the constraint $Y \leq Y_{\max}$ to the mathematical program that defines the goal social utility maximization.
- Using the same analysis as before, it may be seen that:
- A fixed per-unit pricing induces the (modified, resource constrained) optimal social welfare.
- The optimal price will be the *larger* of the following two: the previously-computed price, namely the unconstrained solution of $P = C'_0(Y(P))$; and the solution of the constraint equation $Y(P) = Y_{\max}$

8. Profit Maximization

- Profit maximization is a major goal of commercial firms, and deserves a separate treatment.
- Here we point out a few known observations on the relation between profit maximization and social welfare optimization, that have been recovered in the context of our model:
- The maximal social welfare is an upper bound on the possible profit (using any economic mechanism).
- The maximal profit is not attained by fixed per-unit pricing (except for the very special case of a single user type).
- With fixed per-unit pricing, the profit-maximizing price is *higher* than the socially optimal one.
- Both objective functions are strictly increasing in P , up to the socially optimal price P^*
- For $P \geq P^*$, the profit may not be unimodal in P .

Concluding Remarks

- The model proposed here incorporated some aspects of dynamics and timing that are prevalent in on-line operations, within the classical economic theory of social welfare maximization using fixed per-unit pricing.
- The basic model developed here may provide a basis for additional work on economic aspects of cloud computing, considering in more detail aspects of profit and revenue, multiple resources and resource bundles, non-stationary demand, competition among firms, and effects of resource limitations, among other issues.

Thank You