RC23095 (W0401-146) January 30, 2004
Mathematics

IBM Research Report

Service Level Agreements for Web Hosting Systems

Alan J. King, Mark S. Squillante
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Service Level Agreements for Web Hosting Systems

Alan J. King and Mark S. Squillante

December 3, 2003

1 Introduction

In an accelerating trend, corporations of all sizes are outsourcing their IT infrastructure to service companies. Basic services offered range from shelf-space rental, electricity, air conditioning and network bandwidth to the provision and maintenance of servers, storage, middleware, help centers and deskside support. These are the first steps toward a "computing utility", in which computing services are delivered "on-demand" like energy, communication, transportation, and other commodity services.

The economic pressures underlying this trend are many. On the customer side, oversubscription to accomodate peak usage requirements substantially increases fixed costs, and isolated information technology (IT) departments find it difficult to match the economies of scale that service providers have in plant, equipment refresh rates, personnel skill development, and software infrastructure. On the provider side, Moore's Law improvements on cost and performance together with the steady enrichment and standardization of middleware make it possible to supply a network in which excellent computing resources can be distributed cheaply and securely. But Moore's law also cuts the other way. The slowdown in equipment refresh rates for both hardware and software, the inexorable decrease in unit profits, and the huge fixed costs of technical leadership strongly incent the industry leaders to move to a services provision model with its steady payments and opportunity for increased profits through bundling.

The future of the computing utility is today very much in the making. There is evidence that the market may desire specialized solution offerings like human resources, payroll, procurement, supply chain, email, Web-hosting, data storage, numerically intensive computing, et cetera, all of which may depend on predictable contracts for the underlying computing infrastructure. Commoditized computing elements together with very high global bandwidth capacity may encourage the development of exchanges like those in the energy or telecommunications industries. Finally, as the industry confronts the days of reckoning for the current wave of outsourcing agreements, competitive pressures will undoubtedly lead to industry-wide standards for service agreements.

1.1 Requirements for Service Contracts

Current approaches to managing contracts for computing utilities are necessarily naive, since there is very little in the way of practical experience to guide the parties. A straightforward outsourcing contract simply takes over the customer's entire computing infrastructure, including personnel, under a business case that is based on anticipated savings from labor economies and Moore's law effects. The risks in these contracts are addressed largely by extending the contract duration and consequently postponing the day of reckoning with the customer's cost reduction and QoS expectations. Slightly more sophisticated agreements may contemplate the shared usage of network bandwidth, hosting facilities, and possibly, service personnel.

A computing utility is more complex than an electric or a telephone utility. Peak loads are often many orders of magnitude greater than average loads. Performance of service personnel or installations also vary over many orders of magnitude for a multitude of interacting causes. Performance measurements are not standard or easily described, measured, or predicted. Even the unit of usage is not standardized. Finally, and perhaps most importantly, actual payment for services is made on behalf of users by the customer — who will require evidence that their users are being well treated. This evidence will necessarily be statistical in nature, and will be averaged over long time intervals in order to develop reasonable Quality of Service (QoS) estimates.

This chapter discusses the requirements and features of a Service Level Agreement (SLA) between the computing utility and the customer in such a complex service environment. It is a discussion paper that is intended to outline basic issues and models. Section 2 presents basic concepts of workload measurements, charging, accomodating Quality of Service requirements, and so forth. Section 3 discusses issues concerning the management of portfolios of SLA contracts. Section 4 is a study of the issues arising in a particular type of contract in which customers purchase a committed level of service and pay a differential rate for bursts above this level. Finally, Section 5 envisions an SLA in which customers may place bids for service above the base level.

1.2 Preface and Acknowledgements

The research represented in the chapter is drawn from the results of a few years of work and discussions among our colleagues in IBM Research to investigate technical issues concerning on-demand computing under service level agreements. Some of these sections are based on material that has been published in the academic literature. Parts of Section 1 are adapted from [2]. Section 5 is adapted from the paper that appeared in the proceedings of the annual PIMS Industrial Problem Solving Workshop [4]. Finally, we would like to thank Kaan Katircioglu for his contributions to Section 4.

2 Service level agreements

The computing utility and a customer enter into a contract, called the service level agreement (SLA), that specifies properties of the Quality of Service (QoS) performance measurements and the contract incentives and billing. Typically, these are long-term agreements that include test-bed measurements and phase-in periods to collect the sorts of data needed to develop some of the elements of the SLA. In addition, periodic monitoring and assessment must take place to adapt the infrastructure and SLA to inevitable changes in workloads over time.

The SLA specifies how the QoS measurements are to be taken, the customer's estimate of normal and peak loads, and how the customer is charged or compensated. More formally, an SLA should contain three types of provisions:

- Satisfactory versus unsatisfactory QoS measurements;
- Normal versus exceptional workloads;
- Notifications, actions required or allowed, and charging under various loads and QoS conditions.

The first type of provision reflects the customer's desire to obtain satisfactory levels of QoS. The second type addresses the provider's need to be protected against unexpected demands on the resource pool. The third sets out the general obligations and permissions of the parties to the contract.

Provisions of the SLA will generally be defined using terms from the following three categories:

1. QoS Metric: utilization, response time, and/or throughput measured at various points in the service network;
2. Statistical Measures: averages or expectations, peaks, quantiles, moments, tail probabilities, etc., together with the sample space over which such measures are calculated;
3. Financial/contractual: charging rates, penalties, circumstances under which actions may be taken and/or prices and units may be changed.

Examples of such provisions are: customer must be able to successfully ping the Web site 99.99% of the time, measured response times between certain network nodes must be less than 3 seconds 95% of the time, and so forth, where the sample space for these statistics is commonly defined as the QoS statistic's average value during each five minute interval in the billing period, usually one month.

2.1 Workload Measurement

At the foundation of the SLA is the definition of the workload and its corresponding metrics and statistical measurements. At the rawest level, one can examine http logs to estimate request arrival and file size distributions as a proxy for the workload. This type of workload analysis is probably useful in forecasting utilization rates. But it very likely does not say much about the user experience, since the most obvious type of user QoS metric is the response time. In today's complex middleware environments, the detailed analysis of what level of equipment provision contributed to what part of the response-time measurement is very difficult, if not impossible. Moreover, many http logs actually record the departure time rather than the arrival time for each request.

A straightforward approach that does not require middleware session modeling is to define a simple set of standard user sessions and use them to "ping" the system periodically. Metrics such as response times and throughput measurements can be generated. In addition one can monitor utilization levels on key pieces of equipment (servers, routers). Correlations between utilization levels and standard user session performance statistics can be drawn over time, or derived in a testbed environment. Then, provided the workload mix does not change, one can set utilization *QoS thresholds* as a proxy for response time QoS.

A combination of the simple session ping records and the equipment utilization logs may be sufficient to construct a time series of workload QoS measurements for the purposes of an SLA. Otherwise, more sophisticated methods will likely be required.

2.2 Charging for Resources

In the computing utility service model the promise is that resources may be brought online when required to maintain a QoS. The canonical resources in this type of thinking are a rack-mounted CPU or a network attached disk pack. But there are other types of resources involved in the computing utility. There are the various elements of the network, from the Internet itself through the network access router to the local area network, and even the internal network serving the rack itself. There are service personnel of all flavors, from system administrators to Web-page programmers. There are backup facilities, tape robots, the subsystems of the building housing the hosting facility itself, and so forth.

One of the key distinctions is whether the resource is "assigned" to the user or whether the resource is "shared". The natural charging unit for assigned resources is the *utilization*, the metric for the percentage of the resource actually used. The adoption of this charging unit protects the customer from paying for unused capacity, which is important when the service provider controls the assignment of resources. For shared resources, the natural charging unit is the *throughput*. Throughput is a metric for the number of bytes transmitted or the number of jobs processed.

For both styles of charging, one also must consider the quality of the service received. In Web-hosting, the universally adopted metric is the response-time. In the case of utilization charging, response-time can be accomodated by imposing an upper bound on the utilization threshold — meaning that the user may not be charged for utilization levels that degrade the agreed response time. When charging is done on a per-transaction basis, then compensation may be based on the actual response time achieved.

Customers who are assigned resources actually receive a more complex service than just the usage of a particular piece of equipment for a time interval. They also benefit from a resource shared across all customers: namely, an inventory of equipment sufficient to permit customers to get resources "on-demand". As with many service examples, someone renting equipment actually purchases a bundle of services. Although the charging unit is essentially related to utilization, the pricing structure will include compensation for all these services.

2.3 QoS-adjusted charging

When the system does not satisfy the agreed QoS, then the SLA must address this in some way. One natural way to do this for throughput charging is to impose a penalty on the service provider, such as a penalty charge for each throughput unit that was handled with an unsatisfactory QoS. In the case of utilization charging, it would be more natural to handle this in terms of utilization QoS thresholds.

As the previous subsection noted, QoS concerns may limit the desirable utilization to a threshold fraction that is somewhat less than one. Scaling the utilization by this number produces a metric, the *QoS-adjusted utilization*, which takes the value 1 for "full" utilization — that is, for the highest utilization consistent with the QoS. This scaling can be tuned to reflect the different performance characteristics of different classes of equipment. The customer would be charged for no more than the QoS-adjusted utilization, or even impose a penalty term for the size of the utilization violation above the QoS-adjusted threshold.

For shared resources, when the system happens to be out of compliance with the QoS then the throughput charges may be scaled back or even converted into a penalty to the service provider. This can be viewed as a *QoS-adjusted throughput.* This makes intuitive sense to customers who sign on for a service only to find that it is busy precisely when they wish to use it.

2.4 Normal versus Exceptional workloads

Workloads in computing systems are highly variable. Workload bursts measuring many magnitudes higher than "normal" are common, and the SLA must address these circumstances. There are two aspects to this issue. The first is the provider's need to get some information from the customer about how much equipment might be required to provide the agreed QoS. The second is the customer's desire to obtain service even when the Web site is bursting above the contracted levels.

2.4.1 Variable service level charging schemes

Providers often structure their computing utility service charges into a committed service level charge plus a variable service charge. The customer selects a committed service level at the beginning of the month. The provider agrees to provide the committed level of service at the agreed QoS.

In a utilization charging scheme, the service level is most naturally described in terms of the number of QoS-adjusted utilization resource units. For example, the user could specify a committed service level of 100 CPU-equivalents of a certain Pentium grade. In throughput charging schemes the user could specify a committed service level in terms of a throughput rate.

If the customer bursts above the committed service level, then the provider agrees to provide that level of service on a “best-effort” basis. The provider may assign new resources if they are available. If not then the load is handled on the available equipment. At the end of the month, the customer pays a variable charge based on some cumulative measurement of the workload above the committed level.

The advantage of this scheme is that it provides information to the provider concerning the customer’s own quantile estimate of usage, in addition to protecting the provider from QoS charges. The disadvantage is that the customer perceives that they must pay a higher rate for the less-desirable best-effort service.

There are many types of cumulative measurements that can be used for the variable charges. In section 4 we compare two types of measurements: the peak load, and the average load above the committed level. This analysis shows that the ratio between the committed and variable charging rates is important in determining the customer’s choice of committed level.

In the most commonly used version of this scheme, the variable charge is related to the 95% peak. For the provider this scheme has a natural association with the size of equipment needed to support peak bursts. However, the analysis also shows that such a scheme could result in highly volatile monthly payments. Unfortunately, the apparently fairer charging scheme (averaged usage above committed level) requires such a high ratio between the committed and the variable charging rate to achieve reasonable committment levels that this scheme is probably unsustainable in the marketplace.

2.4.2 Price directed allocation for variable service

An alternative method to allocating resources for variable, sometimes bursty, workloads is to devise a method in which customers pay variable rates for additional service above the committed level. The outcome for the customer is similar to the variable service level charging schemes, in that they have to pay a variable charge for service above the committed level, but the context is completely different since customers will be competing among themselves for the additional resources.

The default version of this scheme is one in which the provider keeps an inventory of excess equipment on hand, and brings it into service at a fixed rate on a first-come-first-served basis.

A slightly more sophisticated version of this allocation scheme would envision two classes of service, say gold and silver, with rates that differ. Customers would indicate which service scheme they wished, and then resources would be allocated by the service provider using revenue maximization concepts. This type of system is explored in the operational chapters.

Spot markets for resources could develop in which customers bid for resources in a local market to obtain additional services. It could even allow customers to trade their excess committed service levels among themselves. A brief sketch of such a system is contained in section 5. Finally, contracts enabling future resource reservation, and possibly even options could also be envisioned.

3 SLA Portfolio and Feasibility Planning

A Web-hosting facility at any given time will have a portfolio of contracts, each with their associated revenue and resource allocation histories. Additionally, the collection of resources in the facility that supports the contract portfolio (routers, servers, disk packs, etc.) can also be viewed as a resource portfolio. One major question that arises in this portfolio perspective is whether or not the resource portfolio is sufficient to serve the workload of the contract portfolio over, say, the weekly cycle of user demand.

To address this question of feasibility planning, one must turn to a different class of models than those considered in the control and optimization chapters, although this class of models can certainly exploit those in the other chapters. First of all, the problem involves considerably longer time horizons and predicting the arrival and service characteristics in the workload processes with limited uncertainty over a weekly (or longer) time horizon can be very difficult, if not impossible. Secondly, it can be very difficult to build a control model with uncertainty that operates over multiple time scales spanning a wide range. In this section we propose a method to address this question that is based on robust optimization and that exploits the models and methods developed in the control and optimization chapters.

Related issues concern the addition of a new contract, any changes to existing contracts, the addition of new services, the investment in new resources to existing portfolios, and so on. The proposed approach can be employed in an analogous manner to address these related issues.

3.1 Forecasting

Web loads are notoriously non-stationary and bursty processes. Despite this, many commercial Web sites show predictable patterns of usage with daily, weekly and monthly cycles [5, 6, 7]. In addition, a pool of Web sites with independent sources of usage-variability will be less variable in the aggregate because of laws of large numbers effects.

For planning periods of a day, week or month, it is reasonable to divide the period into stationary epochs during which the Web sessions seem to have predictable or forecastable arrival and service statistics of the type required for operational management. During these stationary epochs, models and methods such as those in the control and optimization chapter can be used to allocate resources to the workloads.

3.2 Robust Allocation

If the amount of variability in the epochal workload statistics will be very large, then the error distribution of the forecast epochal workloads is an important source of uncertainty that should be incorporated into the portfolio and feasibility planning. One specific way to conveniently capture this type of uncertainty is through the robust optimization model of Bertsimas and Sims [1], in which one solves for a solution that is feasible in a probabilistic sense, where the probability distribution is generated by the (independent) variation of each forecast error about its forecast mean.

During each stationary epoch, one can build a model in which the allocation of servers to Web sites is an variable of optimization. More specifically, this model would provide an estimate of the profit as a function $R(I(t), L(t))$ where $L_j(t)$ denotes the forecasted workload for Web site j during epoch t and $I_{ij}(t)$ denotes the assignment of resource i to Web site j during epoch t. We must have

$$\sum_j I_{ij}(t) \leq b_t \tag{1}$$

where b_t is the total number of servers available during epoch t (which could also be a forecasted number). This model would be developed based on the corresponding models and methods employed as part of the operational management of the system in order to have an accurate representation of the system behavior. By further exploiting the models and methods of the control and optimization chapters for this purpose, one can determine the maximum profit estimate obtained under the optimal scheduling of the workload on the servers with this objective. If there are, say, four epochs in a workday and two epochs in each weekend day, then there would be a total of 24 such models and 24 sets of resource allocation variables to be determined. The transitions between epochs could also be handled in a similar manner.

Because the variable expense of a Web server system is negligible and fixed costs dominate, it seems reasonable to develop a plan that meets (and hopefully exceeds) a revenue target τ. Then the objective could be modeled as the surplus revenue over target, which should be maximized. More specifically, we could seek to maximize the surplus revenue over target

$$\sum_t R(I(t), L(t)) \; - \; \tau \tag{2}$$

subject to this surplus being non-negative and to the resource constraint in (1) and the underlying constraints of the operational management all being satisfied. Of course, the solution of this optimization problem should exploit whatever properties are known for the functions $R(I(t), L(t))$, such as linearity in the case of many utilization-based scenarios and convexity in the case of many throughput-based scenarios with response-time guarantees. This is a model that is similar in style to the portfolio optimization example in [1].

The advantage of the robust optimization model over more traditional stochastic programming models, as in [3], is that the resulting optimization problem only increases linearly in the number of sources of uncertainty. The interpretation of the solution is that it generates a robust allocation that maximizes the risk-adjusted expected revenue which is feasible with a certain level of confidence that can be bounded from below. This would likely suffice for the type of planning system at hand.

3.3 Adding a new resource

The robust allocation model may not be feasible with sufficiently high confidence. In this case one must allow the model to add resources. One must then address the question of how to expense the new resource. One possibility would be to assign an amortization schedule for the resource and require that each added resource subtract its assigned amortization from the total revenue generated by the workloads.

3.4 Adding a new contract

Once the weekly plan has been made with a certain confidence, then one can address the question of whether to add an additional contract, with its forecasted workloads, to the contract portfolio. In some cases the weekly variation of the contract will be such that it complements the existing portfolio, and so the additional revenue will come with no additional expense. Obviously, attracting such a contract will be very much in the interest of the operator, and so pricing discounts might be appropriate. In other cases, the variation will require additional resources and so the operator can use the robust planning model to determine whether this additional contract will be profitable or not.

4 Analysis of Customer choice of Committed Service Levels

This brief section examines the service provider's behavior and revenue implications under two versions of a contractual scheme in which customers select a "committed service level" c_0 with charge $r_0 c_0$ and pay a variable charge r_1 for service levels above c_0. The service is labeled bandwidth, but it could be for any service with a variable user demand.

One version applies rate r_1 to the total usage above c_0

$$r_0 c_0 + (r_1/T)\sum_{t=1}^{T}\{\max[0, C_t - c_0]\} \tag{3}$$

where C_t is the sequence of sampled bandwidth rates, measured by sampling bandwidth usage rates every five minutes. We may call this the CB-AB pricing method.

A second version levies rate r_1 for the variable peak load, so that the monthly bill to the customer is

$$r_0 c_0 + r_1 \max[0, P_{95} - c_0] \tag{4}$$

where P_{95} is the 95% quantile, measured by rank ordering the sample sequence C_t and choosing P_{95} so that it is the smallest number that is greater than 95% of the sample. Call this the CB-PB scheme. (There are many philosophical objections to this pricing method. For one thing it makes a difference to the service provider whether the peak is narrow or wide. But it is in very wide usage.)

4.1 Modeling Customer Choice

Under both charging schemes, the customer is faced with making a choice of c_0. What information will the customer take into consideration?

The customer will have to make some assumptions about the (related) distributions of the peaks P_{95} and the paths $C = \{C_t\}$, $t = 1, \ldots, T$. Let F_P be the distribution function for the peak load, that is,

$$F_P(\alpha) = \Pr\{P_{95} \leq \alpha\} \tag{5}$$

One may as well make the assumption that the C_t are independent and identically distributed and that the time average is equal to the space average, so that

$$(1/T) \sum_{t=1}^{T} \{\max[0, C_t - c_0]\} \approx \int \max[0, x - c_0] dF_C(x) \tag{6}$$

where F_C is the distribution function for C_1.

The customer will also need to make assumptions about their self-imposed penalty d_1 for best effort processing (slow response time, for example, or possibly no response at all). Assume that the available bandwidth is so large that there is no effective upper bound to the choice of committed bandwidth c_0.

Under the PB-CB pricing scheme, the customer's choice can be represented by solving the following optimization:

$$\min_{c_0} r_0 c_0 + \int (r_1 + d_1) \max[0, x - c_0] dF_P(x) \tag{7}$$

The second term is the expected monthly variable charge for loads above the committed level.

Under the CB-AB pricing scheme the customer's choice can be represented by

$$\min_{c_0} r_0 c_0 + \int (r_1 + d_1) \max[0, x - c_0] dF_C(x) \tag{8}$$

Both of these optimization problems have the same form, so we can analyze the customer choice framework under the assumption that the distribution function F could be either F_C or F_P.

The customer's choice is assumed to be unconstrained optimization. The answer is found by taking derivatives and setting equal to zero. The main observation that helps this procedure is to observe that the derivative of the integral is equal to

$$\begin{aligned} \tfrac{d}{dc_0}\textstyle\int_{c_0}^{\infty}(r_1+d_1)(x-c_0)dF(x) &= \tfrac{d}{dc_0}(r_1+d_1)[\textstyle\int_{c_0}^{\infty} xdF(x) - c_0\int_{c_0}^{\infty} dF(x)] \\ &= (r_1+d_1)[-c_0F'(c_0) - [1-F(c_0)] + c_0F'(c_0)] \\ &= (r_1+d_1)[F(c_0)-1] \end{aligned} \tag{9}$$

It follows that c_0 should be chosen to solve

$$0 = r_0 + (r_1+d_1)[F(c_0)-1] \tag{10}$$

The customers choice is the familiar newsboy solution. One can show that

$$\bar{c}_0 = \begin{cases} 0 & \text{if } d_1 < r_0 - r_1, \\ F^{-1}\left(1 - \frac{r_0}{(r_1+d_1)}\right) & \text{if } d_1 \geq r_0 - r_1. \end{cases} \tag{11}$$

The customer's choice of c_0 is essentially the $1 - \frac{r_0}{(r_1+d_1)}$ quantile estimate for the distribution F.

Let us now examine the expected revenue consequences for the service provider, assuming that each customer accurately projects their distribution. Label the customers by $j = 1, \ldots, J$. Each customer chooses $\bar{c}_0^j$ as in (11) with parameter d_1^j and distribution function F^j. The expected revenue for the service provider is

$$\begin{aligned} \textstyle\sum_j r_0\bar{c}_0^j + \sum_j r_1 \int \max[0, x-\bar{c}_0^j]dF^j(x) &= \textstyle\sum_j \left[r_0\bar{c}_0^j + r_1\int_{\bar{c}_0^j} xdF^j(x) - r_1\bar{c}_0^j\left(1 - F^j(\bar{c}_0^j)\right)\right] \\ &= \textstyle\sum_j \left[r_0\bar{c}_0^j + r_1\int_{\bar{c}_0^j} xdF^j(x) - r_0\bar{c}_0^j(\frac{r_1}{r_1+d_1^j})]\right] \\ &= \textstyle\sum_j \frac{r_0 d_1^j}{r_1+d_1^j}\bar{c}_0^j + r_1\sum_j\int_{\bar{c}_0^j} xdF^j(x). \end{aligned} \tag{12}$$

In equation (11) the j-th customer's choice of $\bar{c}_0^j$ depends on the relationship between the customer's penalty rate d_1^j for usage above the committed service level, and the difference $r_0 - r_1$ between the charging rates. Notice that if $d_1^j \approx 0$ then the revenue will be dominated by the second term. Let us now examine the various relationships and their consequences for the provider's revenue streams.

At one extreme one can set $r_0 \gg r_1$. Then the customers with $d_1^j < r_0 - r_1$ have $\bar{c}_0^j = 0$, which places them in the best effort service state all of the time. The expected revenue is

$$r_1 \sum_j \int_0 xdF^j(x). \tag{13}$$

In the CB-PB charging scheme, the expected revenue is r_1 times the sum of these customers' expected peaks P_{95}^j. The CB-AB revenue is r_1 times the sum of their expected usage $E\{C^j\}$.

At the other extreme, one can set $r_0 \ll r_1$. All customers will have $d_1^j > 0 > r_0 - r_1$ and hence $\bar{c}_0^j \gg 0$. The expected revenue is

$$\sum_j \left[\frac{r_0 d_1^j}{r_1 + d_1^j} \bar{c}_0^j + r_1 \int_{\bar{c}_0^j} x dF^j(x) \right]. \tag{14}$$

Specifically, suppose that $r_0 = r_1/5$, and suppose that d_1^j is very small compared to r_1. Then the customer's newsboy choice $\bar{c}_0^j$ can be interpreted as the $1 - 1/5 = .80$ quantile. Then the revenue term will be dominated by

$$r_1 \sum_j \int_{\bar{c}_0^j} x dF^j(x). \tag{15}$$

In the CB-PB charging scheme, this is r_1 times the conditional expected value of the P_{95} peak given that this peak is above its 80% quantile. In the CB-AB charging scheme, this can be interpreted as the conditional expected value of usage given that usage is above its 80% quantile. As d_1^j becomes more significant, the quantile level at which these conditional expectations are taken will rise but the revenue will still be dominated by the second term.

4.2 Conclusions

The equations (13) and (15) highlight the fact that the two-rate contractual scheme can be interpreted as a single-rate scheme r_1 with a required quantile level $1 - r_1/r_0$. These equations also allows us to compare the different versions of the charging schemes. For the service provider, of course, it is important to be paid for the equipment outlays needed to accomodate peak demands. The scheme chosen should bear some relationship to peak demands. Which scheme, CB-PB or CB-AB, is better in this regard?

The CB-PB case purports to obtain revenue for peak demands. Therefore, in this scheme it seems that one ought to set $r_0 \gg r_1$ and obtain revenue proportional to the expected value of the sample peak P_{95} service measurement, as in (13). The impression that CB- PB is unfair to users is quite correct in charging schemes where $r_1 > r_0$ as one can see from (15). In this setting the users are being charged proportional to the expected value of the peak service measurement P_{95} given that that peak is above the $1 - \frac{r_0}{r_1 + d_1}$-quantile, however that peak will be below this level most of the time.

On the other hand, while it may seem that the CB-AB revenue might not relate to peak demands, in fact it does. If one sets $r_1 \gg r_0$ then equation (15) shows that revenue will be proportional to the conditional expected value of service usage given that it is above the $1 - \frac{r_0}{r_1 + d_1}$-quantile. This will be quite similar to the P_{95} for sufficiently high ratios of r_1/r_0. However, these ratios are much higher than the market expectations (typically $r_1 \approx 1.5 r_0$). One way around this is to require the customer to select a quantile level directly, instead of implicitly as in the two-rate scheme.

r_1/r_0	1	2	3	4	5	6	7	8	9	10
c_0 level	60	102	114	121	126	129	132	135	137	139
c_0 as quantile	0.09	0.524	0.677	0.756	0.803	0.836	0.859	0.877	0.891	0.909
CB-PB	149	197	220	235	244	249	252	252	250	246
CB-AB	101	124	133	138	142	145	147	149	151	153

Table 1: Comparison of PB and AB charging.

The example in Table 1 compares the two schemes. It supposes that C_1 follows a lognormal distribution with mean 100 and standard deviation 30, that $r_0 = 1$, that $d_1 = 0.1$, and that the sample peak distribution F_P is replaced by placing probability one at the true peak $P_{95} = 149$. The table compares the amount of revenue collected by the provider under the CB-PB and CB-AB schemes.

One can see that the CB-PB charging scheme grows very rapidly as r_1 increases. Its peak value is approximately 152 which is reached around $r_1 = 8$. The CB-AB charging scheme grows much more slowly. At $r_1 = 8r_0$ it crosses the revenue for the CB-PB value. One sees that the CB-AB scheme does eventually resemble the CB-PB charges with high enough ratio r_1/r_0, but it is a much more stable curve. When one considers the volatility of the variable part of the provider's revenue, one may speculate that the CB-AB revenue is more stable. This conclusion would depend on a careful analysis of the variance of the sampled peak P_{95} versus that of the sampled conditional expectation.

5 Customer Directed Allocation in Service Level Agreements

In this section we describe an agreement between the provider and the clients in which the service level can be increased at the direction of the customer by providing an explicit bid for additional service above a base level. We also sketch a simple algorithm for the hosts problem of allocating resources to changing workloads.

5.1 Customer Directed Service Level Agreement

The customer directed service level agreement has the following components.

1. Base Service Level L, representing the maximum number of servers L to be allocated to a specific class of requests based on the parameters offered by the client.

2. Per-Unit Bid B, representing the variable rate the client agrees to pay for adding servers beyond Base Service Level.

Service up to the customer's Base Service Level L is guaranteed. Requests that exceed L are satisfied if possible when the per-unit bid equals or exceeds the current spot market price, which is the maximum of the bids B over the set of customer's requesting additional service. The host can impose a minimum variable charge M (i.e., cost + economic profit) and customers that wish service beyond L must supply a bid $B \geq M$. Whether the bid is 0, M, or B, it reflects the nature of the customer:

- if the customer wants no service beyond its base level, then its implicit bid is 0;
- if the customer wants service beyond its base level, then its implicit bid is M;
- if the customer wants requests beyond its base level to be completed, then its explicit bid is B.

Finally, the provider must satisfy a basic response time type QoS constraint for allocated servers or pay a penalty charge. It should be noted that the contract is designed so that the customer may change B at any time.

5.2 Resource Allocation

The host reallocates servers by considering the revenue implications of such a move. Consider a family of servers sharing the load of one Web site. When the number of requests for a Web site causes the probability of a large response time to the customer, we say the family of servers *is going red* or enters in a *critical phase.*

To complicate matters, it is not possible to reallocate a server instantaneously. To reallocate a server, we must first let its active threads die out, remove the existing environment, and install the new environment. Only then can it be reallocated to a new customer (this can take on the order of 5 minutes). Finding an optimal solution through dynamic programming is an extremely difficult task due to the long-time horizon in this problem (24 hours) and the short intervals on which decisions are made. This leads to a problem of such large magnitude that an exact solution is impractical.

Instead, various threshold algorithms can be used to get good solutions. We give an example of such a scheme below. We will make our decisions based on three important values, namely

- the probability of a server family going red,
- the expected cost rate C incurred from going red,
- the expected revenue rate R for providing service beyond the customer's required level.

Note that C and R are both non-negative values and cannot be both zero at the same time for a particular family of servers. This is because C is non-zero when we have gone red as a result of not providing the resources required in the SLA, whereas R is non-zero when we have gone red as a result of traffic being so high that the level of resources agreed to in the SLA is insufficient. As mentioned above, it can take about 5 minutes for a server being moved to come on-line in its new family. However, the server does not immediately stop contributing to its original family. One can approximate that it continues to work for approximately 1/3 of a 5-minute interval, after which it is removed from its family. So, for 2/3 of a 5-minute interval, it is not active in any family. This reflects the period in which it is shutting down and being rebooted.

We will introduce subscripts to reflect when the parameter is measured. A subscript of 1 indicates the parameter is measured 5 minutes from now, a subscript of 2 indicates 10 minutes from now, and so on. We will introduce a superscript of +1 or -1 to our parameter P to indicate the probability of going red given the addition or subtraction of a server from the family, respectively; i.e., P_1^{-1} indicates the probability of going red 5 minutes from now given that we have removed a server from the family. For each family of servers, we have created the following measures:

$$\text{Need} = P_1 \cdot C_1 + P_2 \cdot C_2 + (1 - P_1^{+1})R_1 + (1 - P_2^{+1})R_2 \tag{16}$$

Note that due to the mutually non-zero relationship of C and R mentioned above, either the first two terms above are zero, or the second two terms are zero. If the first two terms are zero, this indicates that a traffic level higher than agreed to in the SLA would push us into red, and if the last two terms are zero, this indicates that we might fall into a penalty situation. Thus, *Need* can reflect either a possibility to make extra revenue (if action is taken), or the possibility of paying penalties (if action is not taken), depending on which terms are zero. The higher the *Need* of a family is, the more money that can be lost or earned by adding a server to that family.

$$\begin{aligned}\text{Availability} &= \tfrac{2}{3}P_{1/3}^{-1} \cdot C_{1/3} + P_1^{-1} \cdot C_1 \\ &\quad + P_2^{-1} \cdot C_2 + \tfrac{2}{3}(1 - P_{1/3})R_{1/3} + (1 - P_1)R_1 + (1 - P_2)R_2\end{aligned} \tag{17}$$

Availability is closely related to *Need*, but there are two significant differences. The first is that the superscripts reflect that we are considering removing a computer from the family, as opposed to adding one. The second difference is that there are two extra terms. These terms reflect the fact that the server will be removed from the family after 1/3 of the first 5 minute interval. *Availability* is intended to measure the amount of penalties that will be paid, or revenue lost if we move a server from that family.

In order to decide when to take action and move a server from one family to another, we use the following heuristic:

1. Calculate the *Need* and *Availability* for every family of servers.

2. Compare the largest *Need* value with the smallest *Availability* value. If the *Need* value exceeds the *Availability* value, one server is taken from

the family corresponding to the *Availability* value and given to the family corresponding to the *Need* value.

3. If a server was told to move, go back to step 1 (note: the probabilities will change as the number of servers used to make the calculations will be different). Terminate the loop if no server was told to move in the last iteration.

The above iteration loop should be performed on a frequent basis. We suggest about every 15 seconds. This is only one possible heuristic, and we have yet to actually compare it in simulation with an optimal solution. However, it has the obvious advantage of requiring considerably less computation than a long-time horizon dynamic program, which allows it to be performed very often. This allows us to react nearly instantaneously to a predicted critical situation. The P, C and R values are obtained from forecasts provided from the router control level.

References

[1] D. Bertsimas and M. Sim. The price of robustness. Technical report, MIT, 2002.

[2] M. Bichler, J. Kalagnanam, K. Katircioglu, A. J. King, R. D. Lawrence, H. S. Lee, G. Y. Lin, and Y. Lu. Applications of flexible pricing in business-to-business electronic commerce. *IBM Systems Journal*, 41(2):287–302, 2002.

[3] J. R. Birge. Stochastic programming computation and applications. *INFORMS Journal on Computing*, 9:111–133, 1997.

[4] A. King, M. Begen, M. Cojocaru, E. Fowler, Y. Ganjali, J. Lai, T. Lee, C. Navasca, and D. Ryan. Web hosting service level agreements. In *Proceedings of the Fifth PIMS Industrial Problem Solving Workshop*, 2001. http://www.pims.math.ca/publications/proceedings/.

[5] Z. Liu, M.S. Squillante, C.H. Xia, S.-Z. Yu, L. Zhang, N.M. Malouch. Traffic profiling, clustering and classification for commercial Internet sites. In *Proceedings of the Tenth International Conference on Telecommunication Systems, Modeling and Analysis*, October 2002.

[6] Z. Liu, M.S. Squillante, C.H. Xia, S.-Z. Yu, L. Zhang, N.M. Malouch, P.M. Dantzig. Analysis of measurement data from sporting event web sites. In *Proceedings of the IEEE GLOBECOM Internet Performance Symposium*, November 2002.

[7] Z. Liu, M.S. Squillante, C.H. Xia, S.-Z. Yu, L. Zhang, N.M. Malouch. Profile-based traffic characterization of commercial Web sites. In *Proceedings of the Eighteenth International Teletraffic Congress Conference*, August-September 2003.