Optimal Design of E-Commerce Site Infrastructure from a Business Perspective

Abstract

A methodology for designing data center infrastructure for E-commerce sites is developed. It differs from existing methodologies in that it evaluates and compares alternative designs from a business perspective, that is, by evaluating the business impact (financial loss) imposed by imperfect infrastructure. The methodology provides the optimal infrastructure that minimizes the sum of provisioning costs and business losses incurred during failures and performance degradations. A full numerical example design is provided and results are analyzed. The use of the method for dynamically provisioning an adaptive infrastructure is briefly discussed.

1. Introduction

The problem addressed in this paper is that of infrastructure design for Information Technology (IT) services that cater to business processes that are heavily dependent on IT. An example of such a business process is that supported by an e-commerce site: IT services are the main technology support in such a context and any failure or performance degradation in the IT infrastructure can profoundly affect business operations.

When designing infrastructure to provision IT services, the work reported here concentrates on the data center, that part of the infrastructure most easily controllable by the service provider. Current approaches in data center design usually either consider the problem from a reliability point of view, e.g. [2], from a response time point of view, e.g. [4] or, more recently, from a business perspective, e.g. [5]. The last approach is more novel and merits some discussion.

A new area of academic research – and also of the practitioner's art – is termed Business Impact Management (BIM) [6,7,8]. BIM takes Service Level Management (SLM) to a new maturity level since metrics meaningful to the customer are used to gauge IT effectiveness rather than technical metrics such as availability and response time. This is the crucial departure that the present work takes on most past efforts.

In the present study, infrastructure design aims to decide how many and what kind of resource components should be used to provision IT services. Clearly, adding more fail-over servers will improve service availability and adding more load-balanced servers will lower response time. But what values of availability or of response time should the designer aim for? How does one combine requirements on availability and requirements on response time into coherent design decisions? BIM answers this question as follows: the impact of any IT infrastructure imperfection should be gauged in terms of its impact on business as captured by *business metrics*. The design decisions should then be evaluated in terms of the business impact caused by the resulting design.

This paper provides a concrete business impact model that includes the impact of IT component failures on service availability and hence on the business and the impact of load on performance (response time) and hence on the business. Using this impact model, the problem of designing optimized IT infrastructure is formally defined and solved analytically.

The rest of the paper is structured as follows: section 2 informally discusses the approach while section 3 formalizes it; section 4 considers an application of the method through a full numerical example; section 5 discusses related work; conclusions are provided in section 6.

2. Informal Problem Description

Infrastructure must be designed for an e-commerce site. The main approach is to minimize monthly financial outlays as calculated by the infrastructure cost plus the business loss incurred due to the imperfect infrastructure. Thus, our approach uses a business perspective in the design process through a business impact model. Two kinds of imperfections present in the IT infrastructure are considered, both generating business loss. The first is that components may fail, rendering the service unavailable part of the time. The second is that the load imposed on the infrastructure components results in delays, with the possibility of customers defecting due to overlarge delays.

Sessions visiting the site are divided into two types: revenue-generating sessions where, at some point during the visit, some revenue will accrue to the site's owner; in the second type of session, customers may visit pages on the site, maybe even adding items to a shopping cart, but end up desisting before generating revenue.

The infrastructure itself consists of several tiers, say a web tier, an application server tier and a database tier. Each tier is served by a load-balanced cluster with a certain number of machines, sufficient to handle the applied load. Varying this number of machines affects response time and thus the business loss due to customer defections. Furthermore, additional machines are available in standby mode to improve site availability and hence reduce business losses due to service unavailability.

The problem studied here is to choose the best infrastructure configuration (number and type of machine in each tier's load-balanced cluster and the number of

standby machines), that is, the configuration that minimizes monthly cost plus business losses.

3. Problem Formalization

This section formalizes the infrastructure design problem. The analysis uses results from reliability theory, queuing theory and develops a novel business impact model extending the presented in [8].

3.1. The Design Optimization Problem

Let us first define the design problem to be solved. Please refer to Table 1 for a notational summary and to Figure 1 for a summary of the entities involved.

Symbol	Meaning
RC	Set of resource classes in IT infrastructure (e.g. tiers)
RC_j	j^{th} resource class
n_j	Total number of resources (machines) in RC_j
m_j	Total number of load-balanced machines in RC_j
E	Any time period over which cost and loss are evaluated. Typically a month.
$C(E)$	The infrastructure cost over the time period E
$L(E)$	The financial loss over the time period E due to imperfections in the infrastructure

Table 1: Notational summary for problem definition

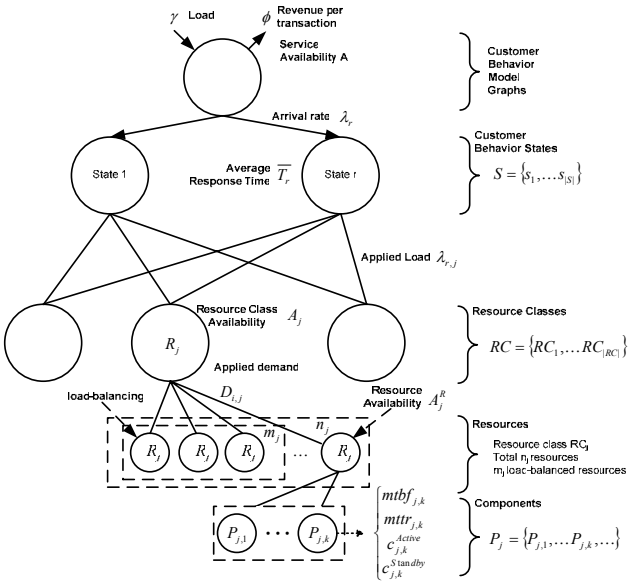

Figure 1: Model entities

The infrastructure provisioning the e-commerce site is made up of a set RC of resource classes. For example, the resource classes could correspond to tiers (web tier, application tier, database tier). Resource class RC_j is provisioned with a total of n_j machines, of which m_j make up a load-balanced cluster while the rest are standby machines. The load-balanced machines enable the tier to handle the input load while the standby machines provide the required availability. The design problem can be posed as an optimization problem as follows:

Find:	For each resource class RC_j, the total number of machines n_j and the number of load-balanced machines m_j
By minimizing:	$C(E)+L(E)$, the total financial impact on the business over the time period E
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$

One must now derive expressions for $L(E)$ and $C(E)$, which we now proceed to do.

3.2. Characterizing the Infrastructure

In this section, expressions for the infrastructure cost $C(E)$ and for site availability, A, are developed. Let us first define the design problem to be solved. Site availability will be used in a later section to derive an expression for loss, $L(E)$. Please refer to Table 2 for a notational summary.

Symbol	Meaning
R_j	An individual resource in RC_j
P_j	The set of components that make up resource R_j
$P_{j,k}$	The k^{th} component in P_j
$c_{j,k}^{Active}$	The cost rate of component $P_{j,k}$ if active
$c_{j,k}^{Standby}$	The cost rate of component $P_{j,k}$ if on standby
A	Site availability
A_j	Availability of resource class RC_j
A_j^R	Availability of an individual resource R_j from class RC_j
$mtbf_{j,k}$	Mean Time Between Failures of component $P_{j,k}$
$mttr_{j,k}$	Mean Time To Repair of component $P_{j,k}$

Table 2: Notational summary for infrastructure

As mentioned previously, the infrastructure used to provision the e-commerce site consists of a set of resource classes, $\{RC_1, \ldots, RC_{|RC|}\}$. Class RC_j consists of a cluster of IT resources. This cluster has a total of n_j identical individual resources, up to m_j of which are load-balanced and are used to provide adequate processing power to handle incoming load. The resources that are not used in a load-balanced cluster are available in standby (fail-over) mode to improve availability.

An individual resource $R_j \in RC_j$ consists of a set $P_j = \{P_{j,1}..., P_{j,k},...\}$ of components, all of which must be operational for the resource to also be operational. As an example, a single Web server could be made up of the following components: server hardware, operating system software and Web software. Individual components are subject to faults as will be described later.

Determining infrastructure cost. Each infrastructure component $P_{j,k}$ has a cost rate $c_{j,k}^{Active}$ when active (that is, used in a load-balanced server) and has a cost rate $c_{j,k}^{S\tan dby}$ when on standby. These values are cost per unit time for the component and may be calculated as its total cost of ownership (TCO) divided by the amortization period for the component. The cost of the infrastructure over a time period of duration E can be calculated as the sum of individual cost for all components.

$$C(E) = E \cdot \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{Active} + \sum_{l=1}^{n_j - m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{S\tan dy} \right)$$

Determining service availability. Recall that IT components making up the infrastructure can fail, producing unavailability and hence business loss. In order to calculate business loss, one needs to evaluate the availability A of the site. This is done using standard reliability theory [1]. For service to be available, all resource classes it uses must be available. Thus:

$$A = \prod_{j \in RC} A_j$$

where A_j is the availability of resource class RC_j. Since this resource class consists of a cluster of n_j individual resources, and since service will be available and able to handle the projected load when at least m_j resources are available for load-balancing, one has, from reliability theory:

$$A_j = \sum_{k=m_j}^{n_j} \binom{n_j}{k} (A_j^R)^k (1 - A_j^R)^{n-k}$$

where A_j^R is the availability of an individual resource R_j from class RC_j. This individual resource is made up of a set P_j of components. Thus:

$$A_j^R = \prod_{k \in P_j} \left[\frac{mtbf_{j,k}}{(mttr_{j,k} + mtbf_{j,k})} \right]$$

where $mtbf_{j,k}$ and $mttr_{j,k}$ are, respectively, the Mean-Time-Between-Failures (MTBF) and Mean-Time-To-Repair (MTTR) of component $P_{j,k}$. Observe that values from MTBF can be obtained from component specifications or historical logs whereas values for MTTR will typically depend on the type of service contract available (gold, silver, etc.).

3.3. The Response Time Performance Model

Since business loss occurs for high values of response time – defection typically occurs when response time reaches 8 seconds [9] – this section uses queuing theory to obtain an expression for $B(T^{DEF})$, the probability that response time has exceeded T^{DEF}, the defection threshold, and that revenue-generating customers will therefore defect. Please refer to Table 3 for a notational summary.

Symbol	Meaning
T^{DEF}	Response time threshold after which customer defection occurs
$B(y)$	Probability that response time is greater than y
S	The set of states in the Customer Behavior Model Graph. Each state represents a particular interaction with the e-commerce site (browse, search, etc.)
γ	The rate at which sessions are initiated at the site
f	The fraction of sessions that generate revenue (type RG sessions)
$p_{i,r}^{RG}$	Probability of going from state i to state r in the RG CBMG
$p_{i,r}^{NRG}$	Probability of going from state i to state r in the NRG CBMG
V_r^{RG}	Average number of visits to state r in RG CBMG
V_r^{NRG}	Average number of visits to state r in NRG CBMG
λ_r	Arrival rate of requests to IT infrastructure in state r
$D_{r,j}$	Demand in seconds applied by each transaction from state r on resources from resource class RC_j
α_j	Speedup factor for resources in resource class RC_j
$\mu_{r,j}$	Service rate at a class RC_j resource for transactions from state r
$\lambda_{r,j}$	Arrival rate of requests to a class RC_j resource in state r
$\rho_{r,j}$	Utilization of class RC_j resources in processing transactions from state r
ρ_j	Total utilization of class RC_j resources
$T_r(y)$	Cumulative distribution of response time for requests in state r
$\overline{T}_r$	Average response time for requests in state r
NZ^{RG}	Set of states from the RG CBMG that have non-zero average number of visits

Table 3: Notation for response time analysis

In order to assess response time performance, one must model the load applied to the IT resources. Access to the e-commerce site consists of sessions, each generating several visits to the site's pages. The mathematical development that follows is (initially) based on the Customer Behavior Model Graph (CBMG) [9], that allows one to accurately model how customer-initiated sessions accessing a web site impose load on the IT infrastructure. A CBMG consists of a set *S* of states and probabilities of moving between states. Each state typically represents a web site page that can be visited and where a customer interacts with the e-commerce site. As an example, consider Figure 2 that shows the states and the transition probabilities for a simple but typical e-commerce site. The customer always enters through the Home state and will then Browse (with probability 0.4) or Search (with probability 0.6). The Select state represents viewing the details of a product and the other states are self-explanatory.

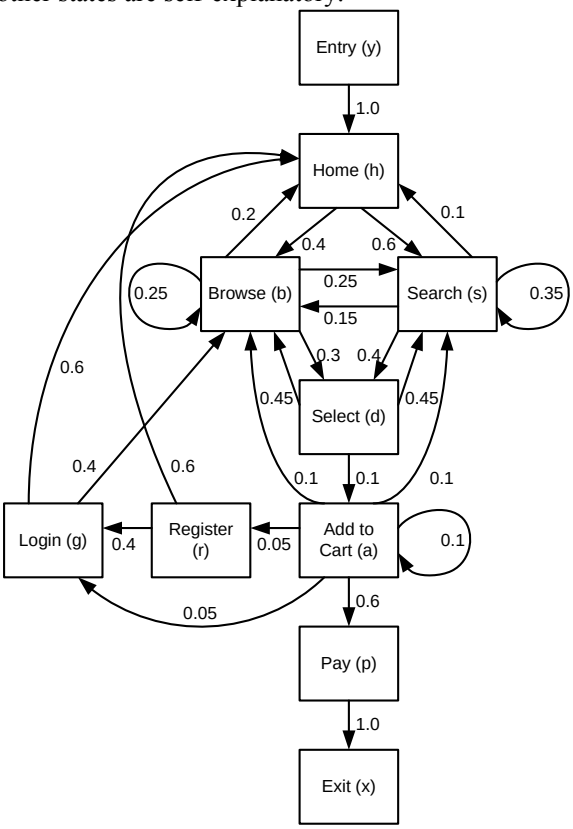

Figure 2: CBMG for the e-commerce site

Some of these states are revenue-generating (for example, a state "Pay" where the customer pays for items in a cart). Sessions are initiated at a rate of γ sessions per second. For our purposes, we divide the sessions into two types: type RG sessions generate revenue while type NRG sessions do not. Customer behavior for each session type is modeled by means of its own CBMG [9]. The particular CBMG shown in Figure 2 is an example applicable to type RG sessions since the Pay state is visited with non-zero probability. For type NRG sessions, the CBMG will include the same states but with different probabilities. For example, there will be no path leading to the state Pay, the only revenue-generating state in this particular graph. The fraction of sessions that are revenue-generating is denoted by *f*. The transition probability matrices have elements $p_{i,r}^{RG}$, the probability of going from state *i* to state *r* in the RG CBMG, and $p_{i,r}^{NRG}$ for the NRG CBMG. As shown in [9], flow equilibrium in the graph can be represented by a set of linear equations that can be solved to find the average number of visits per session to state *r*. The set of equations to be solved for the RG CBMG is:

$$V_1^{RG} = 1$$

$$V_r^{RG} = \sum_{i=1}^{|S|} V_i \times p_{i,r}^{RG}$$

In this set of equations, the average number of visits in the RG CBMG is V_r^{RG}. The situation for the NRG CBMG is similar and the average number of visits is V_r^{NRG}.

We now need to find $B(T^{DEF})$. In order to find this probability, the IT services are modeled using a multi-class open queuing model. Open queuing models are adequate when there is a large number of potential customers, a common situation for e-business. Since, in each state, the demands made on the IT infrastructure are different, each state in the CBMG represents a traffic class in the queuing model. Let us examine state *r*. The arrival rate of requests corresponding to this state is $\lambda_r = \gamma\left(f \cdot V_r^{RG} + (1-f) \cdot V_r^{NRG}\right)$ transactions per second. Transactions demand service from all resource classes. Demand applied by each transaction from state *r* on resources from resource class RC_j is assumed to be $D_{r,j}$ seconds. In fact this is the service demand if a "standard" processing resource is used in the class RC_j resources. In order to handle the case of more powerful hardware, assume that a resource in class RC_j has a processing speedup of α_j compared to the standard resource. Thus, service time for a transaction is $D_{r,j} / \alpha_j$ and the service rate at a class RC_j resource for transactions from state *r* is:

$$\mu_{r,j} = \frac{\alpha_j}{D_{r,j}}$$

Finally, since there are m_j identical load-balanced parallel servers used for processing in resource class RC_j, response time is calculated for an equivalent single server with input load [9]:

$$\lambda_{r,j} = \frac{\lambda_r}{m_j}$$

Thus the utilization $\rho_{r,j}$ of class RC_j resources in processing transactions from state *r* is:

$$\rho_{r,j} = \frac{\lambda_{r,j}}{\mu_{r,j}} = \frac{\lambda_r D_{r,j}}{m_j \alpha_j}$$

The total utilization ρ_j of class RC_j resources due to transactions from all states is:

$$\rho_j = \sum_{r=1}^{|S|} \rho_{r,j}$$

Observe that, when load is so large that any $\rho_j \geq 1$, then we have $B\left(T^{DEF}\right) = 1$, since response time is very high for saturated resources.

Now, in order to find $B\left(T^{DEF}\right)$ when $\rho_j < 1$, let us find the cumulative distribution of response time, $T_r(y) = \Pr\left[\widetilde{T}_r \leq y\right]$. Here, $\widetilde{T}_r$ is the random variable corresponding to the response time seen by the customer in state *r*. Since the transactions must (potentially) use resources from all resources classes, the total response time for a transaction from state *r* is the sum of $|RC|$ random variables, one for each resource class. In order to find the probability distribution of a sum of random variables, one may multiply the Laplace transforms of the distribution function [10]. In order to make mathematical treatment feasible, assume Poisson arrivals (this is a reasonable for stochastic processes with large population) and exponentially distributed service times. From queuing theory, the Laplace transform of response time (waiting time plus service time) for a single-server queue is:

$$T^*(s) = \frac{a}{s+a}$$

where $a = \mu(1-\rho)$, μ is the service rate and ρ is the utilization. Recall that input load from several states are going to the same resource class. Thus, for the combination of resource classes used by transactions in state *r*, we have:

$$T_r^*(s) = \prod_{j \in RC} \frac{a_{r,j}}{s + a_{r,j}}$$

where

$$a_{r,j} = \mu_{r,j}(1-\rho_j)$$

Inverting the transform yields the probability density function of response time, which is integrated to find the cumulative probability distribution function (PDF) of response time, $T_r(y)$.

We are now ready to find $B(T^{DEF})$. Customer defection will occur and cause business loss only in the revenue-generating sessions. Let NZ^{RG} represent the set of states from the RG CBMG that have non-zero average number of visits. The crucial fact to be understood is that if the response time in *any* of the states in NZ^{RG} exceeds the threshold T^{DEF}, then defection will occur; in other words, a customer defects when any page access becomes too slow. Put differently, defection will *not* occur if all response times are within the threshold. We can thus say:

$$B\left(T^{DEF}\right) = 1 - \prod_{r \in NZ^{RG}} T_r(T^{DEF})$$

Finally, one may desire to evaluate the average response time for each state, possibly to be included in an SLA. Average response time may be found from the Laplace transform as follows:

$$\overline{T}_r = -\left.\frac{dT_r^*(s)}{ds}\right|_{s=0}$$

Average response time to the site (over all states) is the weighted average of $\overline{T}_r$ using the average number of visits as weights.

3.4. The Business Impact Model

The key expressions to be used in determining business loss have been determined in the last sections. These are site availability, *A*, and the defection probability for revenue-generating sessions, $B(T^{DEF})$. These are now combined to calculate business loss.

Revenue-generating sessions are initiated at a rate of $f \cdot \gamma$ sessions per second. If availability were perfect and response time were always low, this would also be the revenue-generating throughput (sessions ending without defection and producing revenue). However, due to IT imperfections (see Figure 3), the actual throughput is *X* transactions per second, with $X < f \cdot \gamma$. Let the average revenue per completed revenue-generating session be φ. The lost throughput in transactions per second is ΔX. Thus, one may express the business loss over a time period *E* as: $L(E) = \Delta X \cdot \varphi \cdot E$.

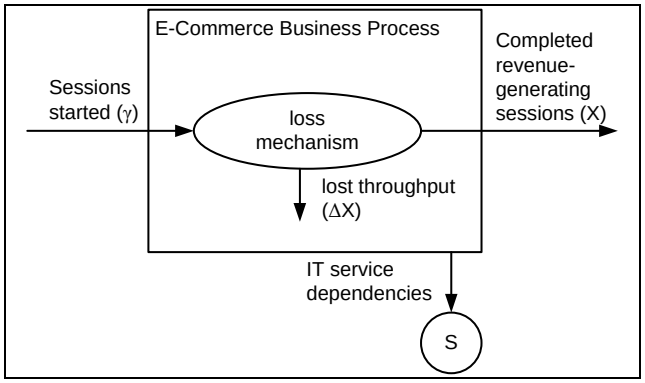

Figure 3: E-commerce Business Loss

Loss has two components: loss due to unavailability and loss due to high response time. Thus, we have: $L(E) = (\Delta X^A + \Delta X^T) \cdot \varphi \cdot E$ where ΔX^A is the throughput lost due service unavailability and ΔX^T is the throughput lost due to high response time (customer defections).

When the site is unavailable, throughput loss is total and this occurs with probability *1- A*:

$$\Delta X^A = f \cdot \gamma (1 - A)$$

where ΔX^A is the loss attributable to site unavailability, $f \cdot \gamma$ is the rate of revenue-generating sessions incident on the site and A is the site availability. On the other hand, when the site is available, loss occurs when response time is slow and this occurs with probability A:

$$\Delta X^T = f \cdot \gamma \cdot B(T^{DEF}) \cdot A$$

where ΔX^T is loss attributable to high response time and $B(T^{DEF})$ is the probability that the site response time is larger that some threshold T^{DEF} in any state visited by a revenue-generating session.

The above results are combined to yield:

$$L(E) = f \cdot \gamma \cdot \left(1 - A + B\left(T^{DEF}\right) \cdot A\right) \cdot \varphi \cdot E$$

4. An Example E-Commerce Site Design

The purpose of this section is to use the above results and exercise the IT infrastructure design process for a representative e-commerce site. The values for all input parameters are meant to be typical for current technology [2,9,11].

The site has a revenue-generating CBMG as shown in Figure 2. The non-revenue-generating CBMG has transition probabilities shown in Table 4. The transition probabilities for a given site can easily be gathered from web server log files. These CBMGs yield the average number of visits shown in Table 5. Observe that, for RG sessions, the Pay state is always visited whereas it is never visited in NRG sessions. The IT infrastructure consists of three resource classes: web tier, application tier and database tier. Furthermore, the parameters shown in Table 6 and Table 7 are used, except where otherwise noted. In the Table 6, tuples such as (a,b,c) represent parameter values for the three resource classes (web, application, database); furthermore, each resource is made up of three components: (hardware (hw), operating system (os), application software (as)).

	y	h	b	s	g	p	r	a	d	x
Entry (y)		1.00								
Home (h)			.55	.40						.05
Browse (b)		.10	.50	.20					.10	.10
Search (s)		.10	.15	.40					.25	.10
Login (g)		.60	.30							.10
Pay (p)										1.0
Register (r)		.50			.40					.10
Add to Cart (a)			.40	.30	.05		.05	.05	.10	.05
Select (d)			.45	.40				.05		.10

Table 4: Transition probabilities in NRG CBMG

State	RG Session	NRG Session
Entry	1.000	1.000
Home	1.579	1.780
Browse	2.325	4.248
Search	3.300	3.510
Login	0.167	0.005
Pay	1.000	0.000
Register	0.083	0.003
Add to cart	1.667	0,069
Select	2.250	1.309
Exit	1.000	1.000

Table 5: Average number of visits

Parameters	Values
T^{DEF}	8 seconds
φ	$1 per transaction
γ	14 transactions per second
f	25%
E	1 month
α_j	(1,1,3)
$c_{j,k}^{Active}$ (($/month)	hw =(1100, 1270, 4400) os=(165, 165, 165) as=(61, 35, 660)
$c_{j,k}^{Standby}$ ($/month)	hw =(1000, 1150, 4000) os=(150, 150, 150) as=(55, 30, 600)
$\left(A_{web}^R, A_{as}^R, A_{db}^R\right)$	(99.81%, 98.6%, 98.2%) (these values are calculated from appropriate MTBF and MTTR values)

Table 6: Parameters for example site

	h	b	s	g	p	r	a	d
Web tier	50	20	30	70	50	30	40	30
Application tier	0	30	40	35	150	70	40	25
Database tier	0	40	50	65	60	150	40	30

Table 7: Demand in milliseconds

Let us try to design the site infrastructure in an ad hoc fashion. This is done by trying to minimize cost while maintaining reasonable service availability and response time. The cheapest infrastructure here is (n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})=(1,1,1,1,1,1). However, this design cannot handle the applied load (average response time is very high) due to saturation of the servers in all tiers. In order to handle the load and make sure that no server is saturated, the design must use (n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})=(5,5,2,5,5,2). There are 5 servers in the web and application tiers and 2 servers in the database tier. This design has a monthly cost of $24430, average response time of 1.76 s. and service availability of 84.32%. Since this value for availability is typically considered inadequate, the designer may add 1 standby server in each tier, yielding a design with infrastructure (6,6,3,5,5,2),

monthly cost of $31715, average response time of 1.76 s. and service availability of 99.38%. If this value of unavailability is still considered inadequate – and one may well ask how the designer is supposed to know what value to aim for – then an additional standby server may be added to each tier, yielding a design with infrastructure (7,7,4,5,5,2), monthly cost of $39000, average response time of 1.76 s. and service availability of 99.98%. There the designer may rest. We will shortly show that this is not an optimal design.

The problem is that none of the above design decisions take business loss into account. It is instructive to discover the values for loss for the above designs as well as for the optimal design which minimizes the sum of cost plus loss as shown in section 3.4 (see Table 8).

Table 8: Comparing infrastructure designs

Infra	Cost	Response Time Loss	Unavail. Loss	Cost + Loss	Cost of choosing wrong
(5,5,2,5,5,2)	24430	4964417	1422755	6411602	6361129
(6,6,3,5,5,2)	31715	5851498	55929	5939142	5888669
(7,7,4,5,5,2)	39000	5886685	1712	5927397	5876924
(8,9,5,6,6,3) (optimal)	**48351**	**754**	**1368**	**50473**	**0**

For the optimal design (8,9,5,6,6,3), the average response time is 0.26 s., availability is 99.98%. It has lowest overall cost+loss, and the table clearly shows the high cost of designing in an ad hoc fashion: a wrong choice can cost millions of dollars per month. Observe that an over-design can also be suboptimal. In this case, business loss could be quite low, but as a result of an over-expensive design.

It is interesting to note that the importance of the site revenue should (and does) affect infrastructure design. For example, by reducing per transaction revenue from $1.00 to $0.10, the optimal design is no longer (8,9,5,6,6,3) but (8,9,3,6,7,2), with monthly cost $38516, total monthly loss $2467, average response time 0,26 s. and availability 99.87%; as expected, a site generating less revenue merits less availability (99.87% rather than 99.98%). In other scenarios, response time rather than availability could be the main metric affected. Additional scenarios concerning the importance of per transaction revenue are discussed in a previous report using a simplified version of the impact model presented here [8].

Finally, we can show how sensitive the optimal design, IT metrics and business metrics are to variations in input load. This is an important consideration since the design procedure assumes a fixed value for input load (γ) while, in practice, this load varies over time. Consider Figure 4 which shows the total cost plus loss (i.e., $C(E)+L(E)$) as load varies. The load values (γ) are divided in three regions: the first design is (8,9,4,6,6,2) and is optimal for all values of load in the left region (γ=13.25 to 13.85); the middle region (γ=13.85 to 14.15) has an optimal design of (8,9,5,6,6,3) with an additional database server; the right region (γ=14.25 to 15.40) has an optimal design of (8,10,5,6,7,3) with an additional application server. Four curves are shown in the figure; the first (blue, cross marker) shows cost plus loss when using the design that is optimal for the left region; similarly, the second curve (green, circle marker) shows cost plus loss when using the design that is optimal for the middle region; the third curve (red, triangle marker) shows the situation for the design that is optimal for the right region. Finally, the heavy black curve simply follows the bottom-most curve in any region and represents the optimal situation in all regions, using three different infrastructure designs, one for each region.

Three major conclusions can be reached from this figure. First, an optimal design remains optimal for a range of load. Although some of these ranges are wider than others, the width of the ranges lends some hope that a static infrastructure design may be optimal or close to optimal even in the presence of some variation in load. The second conclusion is that, in the presence of larger load variations, an infrastructure design can quickly become suboptimal; an example is the leftmost optimal design (8,9,4,6,6,2) which quickly accumulates heavy losses at loads greater than γ=13.85). In this case, dynamic provisioning can be used to introduce a new infrastructure configuration at appropriate times to reduce business losses (scaling up) or to reduce infrastructure costs (scaling down), as appropriate. The third major conclusion is that it appears that the business impact model described in this paper can be used as one of the mechanisms for dynamic provisioning since it captures appropriate load transition points for reprovisioning using a business perspective. Further investigations will be conducted concerning this point in the future.

Additional interesting details can be seen in Figure 5 which shows individual components of cost and business loss for the three data center designs described above. Costs clearly go up (from left to right) as designs use more resources, although the increase in cost is more than offset by the reduction in loss offered by better designs. Finally, Figure 6 shows response time for the three designs as well as the optimal response time (heavy black line) when dynamic provisioning triggers in the optimal design at all load levels.

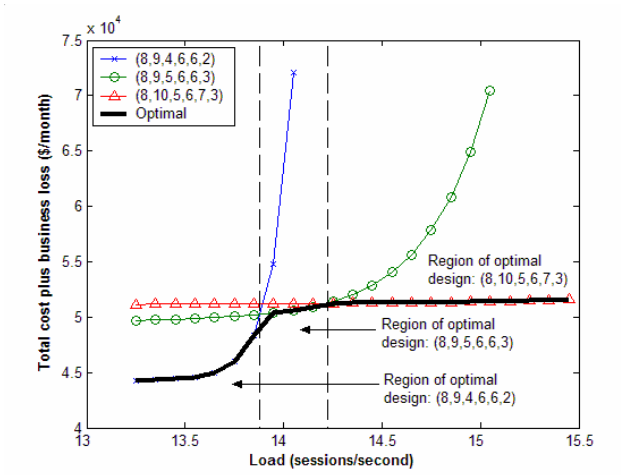

Figure 4: Sensitivity of total cost plus loss due to load

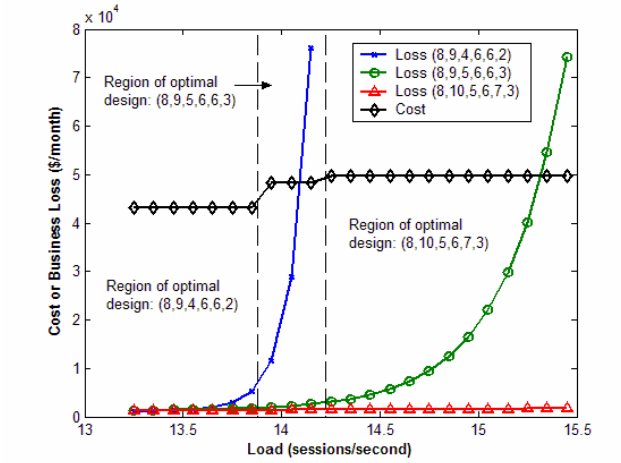

Figure 5: Sensitivity of cost and loss due to load

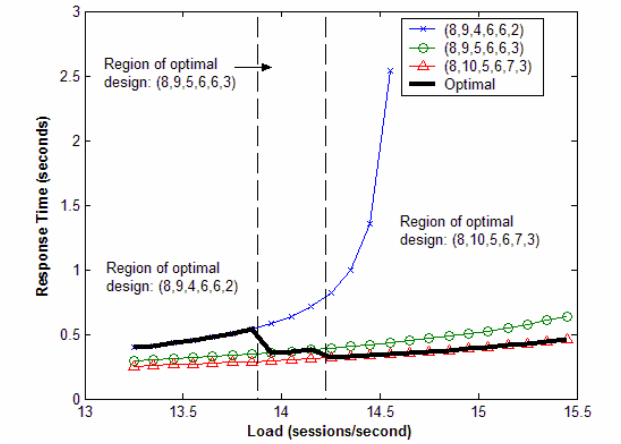

Figure 6: Response time for various designs

5. Related Work

In the area of infrastructure design, [2] describes a tool – AVED – used for capacity planning to meet performance and availability requirements and [3] describes a methodology for finding minimum-cost designs given a set of requirements. Similarly, [4] optimizes using IT level metrics. However, none of these references consider the problem of capacity planning from a business perspective, using business metrics. Furthermore, response time considerations are not directly taken into account in [2,3].

Finally, [5] considers the dynamic optimization of infrastructure parameters (such as traffic priorities) with the view of optimizing high-level business objectives such as revenue. It is similar in spirit to the work reported here, although the details are quite different and so are the problems being solved (the paper considers policies for resource allocation rather than infrastructure design). The model is solved by simulation whereas our work is analytical.

The business impact model presented here is detailed in [8]. The current work adds a different customer behavior model (CBMG [9]) and a new analysis of customer defection, as well as new conclusions concerning the sensitivity of the optimal design to changes in applied load.

6. Conclusions

In summary, a method was discussed to design IT infrastructure (this is also called *capacity planning*) from a business perspective. The method is novel in that three types of metrics are considered – availability, response time and financial impact – whereas most studies consider only one of the first two in isolation. The three metrics are tied through a business impact model, one of the main contributions of the present work. The method itself finds optimal data center infrastructure configurations by minimizing the total cost of the infrastructure plus the financial losses suffered due to imperfections. It is important to note that a business impact model such as the one discussed here can be used in other contexts to solve other IT-management-related problems such as incident management, Service Level Agreement (SLA) design [8], etc.

In the future, we plan to develop new impact models applicable to business processes other than e-commerce (say, manufacturing, CRM, etc.); additionally, more holistic models that include the network and other components outside the data center may be considered. Finally, a fuller study of the use of business impact models in adaptive environments can be undertaken; this would be an expansion of the initial comments given here concerning dynamic provisioning.

References

1 K. S. Trivedi, Probability & Statistics with Reliability, Queuing and Computer Science Applications, Prentice-Hall, 1982.

2 J. Janakiraman; J. R. Santos, Y. Turner; “Automated Multi-Tier System Design for Service Availability”; In Proceedings of the First Workshop on Design of Self-Managing Systems, June 2003.

3 D. Ardagna, C. Francalanci, “A Cost-oriented methodology for the design of Web based IT architectures”; In Proceedings of the 2002 ACM symposium on Applied Computing, 2004.

4 D. Menascé, D. Barbara, and R. Dodge, “Preserving QoS of E-commerce Sites Through Self-Tuning: A Performance Model Approach”; In Proceedings of 2001 ACM Conference on E-commerce, Tampa, 2001.

5 S. Aiber, D.Gilat, A. Landau, N. Razinkov, A. Sela, and S. Wasserkrug, “Autonomic Self-Optimization According to Business Objectives”; In Proceedings of the International Conference on Autonomic Computing, 2004.

6 M. Sallé and C. Bartolini, “Management by Contract”; In Proceedings of the 2004 IEEE/IFIP Network Operations and Management Symposium, Seoul, Korea, April 2004.

7 P. Mason, “A New Culture for Service-Level Management: Business Impact Management”; IDC White Paper.

8 J. P. Sauvé, F.T. Marques, J. A. B. Moura, M. C. Sampaio, J. F. H. Jornada, E. Radziuk; “SLA Design from a Business Perspective”, RT-DSC 003/2005, Department of Computing Systems (DSC), Universidade Federal de Campina Grande (UFCG), Brazil, http://jacques.dsc.ufcg.edu.br/projetos/bl-hp/reports/003-2005.pdf, 2005.

9 D. Menascé and V. Almeida, “Scaling for E-Business”, Prentice Hall PTR, 2000.

10 L. Kleinrock, Queuing Systems, Vol I: Theory, Wiley, New York, 1975.

11 D. Menascé, V. Almeida and L. Dowdy; “Performance by Design: Computer Capacity Planning by Example”, Prentice Hall PTR, 2004.