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

Jacques Sauvé*, Filipe Marques*, Antão Moura*, Marcus Sampaio†, João Jornada† and Eduardo Radziuk†
*Universidade Federal de Campina Grande, Brazil
Email: {jacques,filipetm,antao,sampaio}@dsc.ufcg.edu.br
†Hewlett-Packard-Brazil
Email: {joao.jornada,eduardo.radziuk}@hp.com

***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.**

I. 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.

Within the general problem of 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. [1], from a response time point of view, e.g. [2] or, more recently, from a business perspective, e.g. [3]. 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-Driven IT Management (BDIM) [4], [5], [6]. BDIM takes Service Management (SM) 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? BDIM 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, in turn, on the business and the impact of load on performance (response time) and, in turn, 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 II informally discusses the design problem using a BDIM approach while section III formalizes it; section IV considers an application of the method through a full numerical example; section V discusses related work; conclusions are provided in section VI.

II. Informal Problem Description

Infrastructure must be designed for an e-commerce site. The approach taken was first suggested in [6] and aims to minimize monthly financial outlays as calculated by the infrastructure cost plus the business loss incurred due to the imperfect infrastructure. Thus, the 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, may possibly even be 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 tier and a data 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

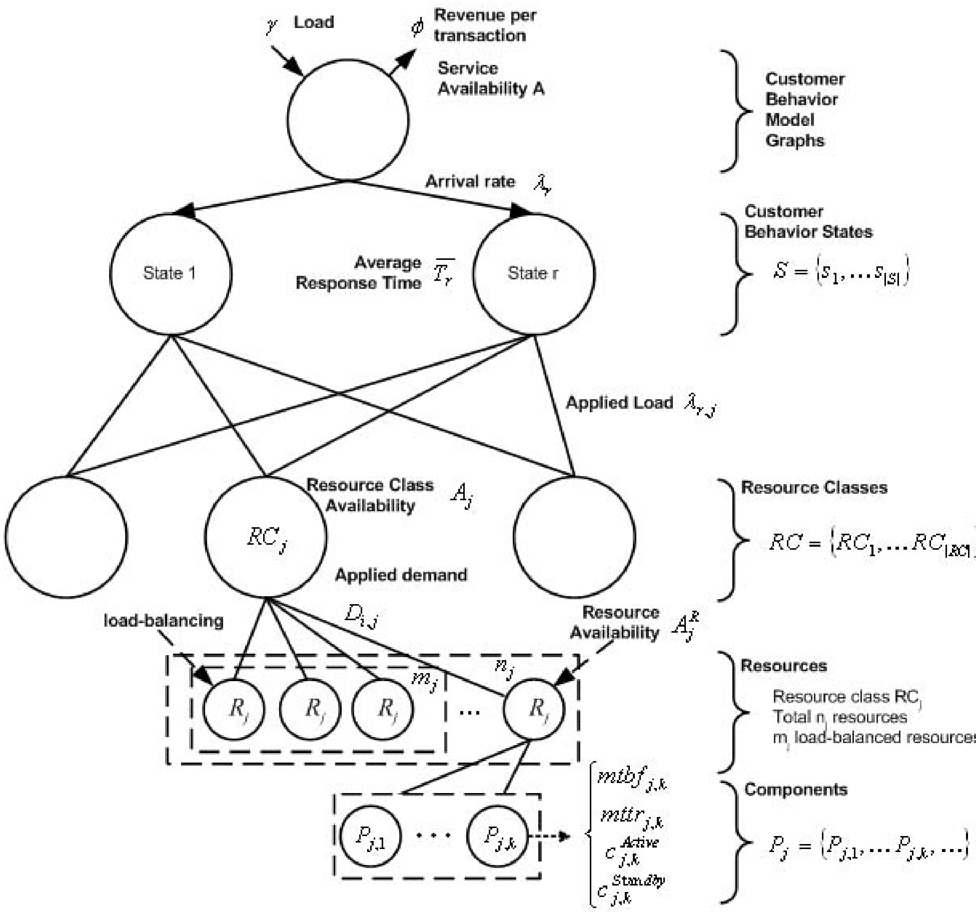

Fig. 1. Model entities

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.

III. Problem Formalization

This section formalizes the infrastructure design problem. The analysis uses results from reliability theory, queueing theory and extends a novel business impact model presented in [6].

A. The Design Optimization Problem

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

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, data 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

TABLE I

Notational summary for problem definition

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

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:	$C(\Delta T) + L(\Delta T)$, the total financial impact on the business over the time period ΔT
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$

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

TABLE II
NOTATIONAL SUMMARY FOR PROBLEM DEFINITION

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 $k^{\underline{th}}$ 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	The site availability
A_j	The availability of resource class RC_j
A_j^R	The availability of an individual resource R_j from class RC_j
$mtbf_{j,k}$	The Mean-Time-Between-Failures of component $P_{j,k}$
$mttr_{j,k}$	The Mean-Time-To-Repair of component $P_{j,k}$

B. Characterizing the Infrastructure

In this section, expressions for the infrastructure cost $C(\Delta T)$ and for site availability, A, are developed. Site availability will be used in a later section to derive an expression for loss, $L(\Delta T)$. Please refer to Table II for a notational summary.

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 $P_j = \{P_{j,1}, \cdots, P_{j,k}, \cdots\}$ of components, all of which must be operational for the resource to also be operational. As an example, a single Web server can 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 $c_{j,k}^{standby}$ 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 ΔT can be calculated as the sum of individual cost for all components.

$$C(\Delta T) = \Delta T \cdot \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{active} + \sum_{l=1}^{n_j - m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{standby} \right)$$

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 [7]. For service to be available, all resource classes it uses must be available. Thus:

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

TABLE III
NOTATIONAL SUMMARY FOR PROBLEM DEFINITION

Symbol	Meaning
T^{DEF}	The response time threshold after which customer defection occurs
$B(y)$	The 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
α_j	Speedup factor for resources in resource class RC_j
$T_r(y)$	Cumulative distribution of response time for requests in state r
NZ^{RG}	Set of states from the RG CBMG that have non-zero average number of visits

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:

$$A_j = \sum_{k=m_j}^{n_j} \left[\binom{n_j}{k} \cdot \left(A_j^R\right)^k \cdot \left(1 - A_j^R\right)^{n-k} \right]$$

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, all of which must be operational for the resource to be operational. Thus:

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

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.

C. The Response Time Performance Model

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

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

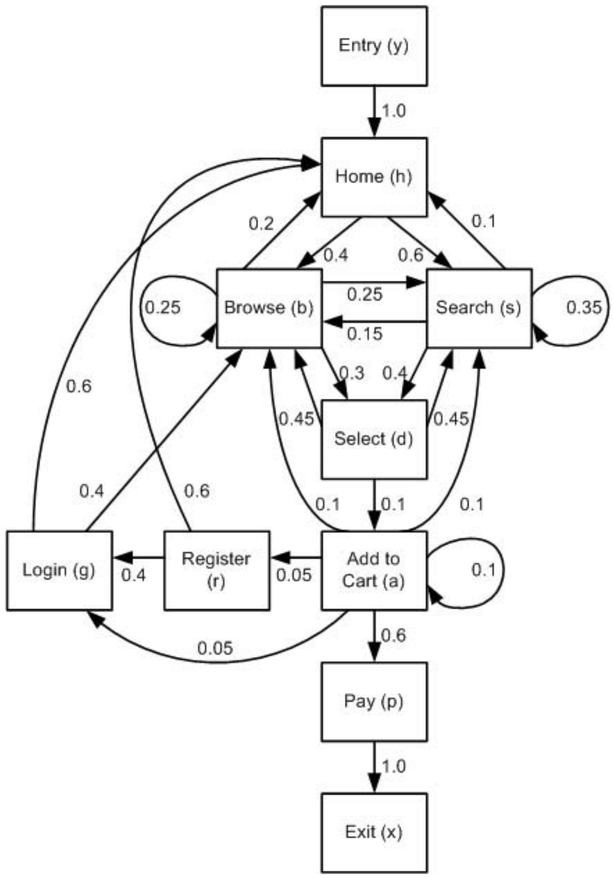

Fig. 2. CBMG for the e-commerce site

visits to the site's pages. The mathematical development that follows is (initially) based on the Customer Behavior Model Graph (CBMG) [8], that allows one to accurately model how customer-initiated sessions accessing a web site impose load on the IT infrastructure. The use of the CBMG model will then be extended to include business impact.

A CBMG consists of a set S of states and transitions between states occuring with particular probabilities. 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.

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 [8]. 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 Pay state, 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. Observe that S, f and all transition probabilities for these graphs can be obtained automatically from web server logs.

As shown in [8], 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:

$$V_1^{RG} = 1$$

$$V_r^{RG} = \sum_{i=1}^{|S|} \left(V_i \cdot p_{i,r}^{RG}\right)$$

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 $B\left(T^{DEF}\right)$, the probability that customers will defect due to response time exceeding T^{DEF} while navigating. 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. Let the cumulative distribution of response time in state r be $T_r(y) = \Pr\left[\widetilde{T}_r \leq y\right]$, where $\widetilde{T}_r$ is the random variable corresponding to the response time seen by the customer in state r. $T_r\left(T^{DEF}\right)$ is the probability that there will be no defection in a visit to state r. Thus, since defection will *not* occur if all response times are within the threshold, we can say:

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

In order to find $T_r\left(T^{DEF}\right)$, the IT services are modeled using a multi-class open queueing model. Open queueing 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 queueing model. Standard queueing theory [9] can be used to solve this model by considering the arrival rate of requests corresponding to state r as $\lambda_r = \gamma \cdot \left(f \cdot V_r^{RG} + (1-f) \cdot V_r^{NRG}\right)$ transactions per second. Observe that, in this analysis, some simplifications are made to make mathematical treatment feasible. The assumptions are:

1) Poisson arrivals are assumed (this is a reasonable for stochastic processes with large population) and also exponentially distributed service times. This assumption

TABLE IV
TRANSITION PROBABILITIES IN NRG CBMG

	y	h	b	s	g	p	r	a	d	x
Entry (y)		1.00								
Home (h)			0.55	0.40						0.05
Browse (b)		0.10	0.50	0.20					0.10	0.10
Search (s)		0.10	0.15	0.40					0.25	0.10
Login (g)		0.60	0.30							0.10
Pay (p)										1.00
Register (r)		0.50			0.40					0.10
Add to Cart (a)			0.40	0.30	0.05		0.05	0.05	0.10	0.05
Select (d)			0.45	0.40				0.05		0.10

is necessary to find the probability distribution function of response time using Laplace transforms [9]. This assumption is frequently made when analyzing performance [10]. Also, since the numerical results will be used primarily to *compare* designs, one expects little sensitivity to particular distributions.

2) 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 reduced by a factor of m_j [8].

Details of the full mathematical development can be found in [11].

D. The Business Impact Model

The key expressions to be used in estimating business loss have been determined in the last sections. These are site availability, A, and the defection probability for revenue-generating sessions, $B\left(T^{DEF}\right)$. 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 always low, this would also be the revenue-generating throughput (sessions end without defection and produce 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 ΔT as: $L\left(\Delta T\right) = \Delta X \cdot \phi \cdot \Delta T$.

Loss has two components: loss due to unavailability and loss due to high response time. Thus, we have: $L\left(\Delta T\right) = \left(\Delta X^A + \Delta X^T\right) \cdot \phi \cdot \Delta T$ 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 $1 - A$:

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

On the other hand, when the site is available, loss occurs when response time is slow and this occurs with probability A:

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

The above results are combined to yield:

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

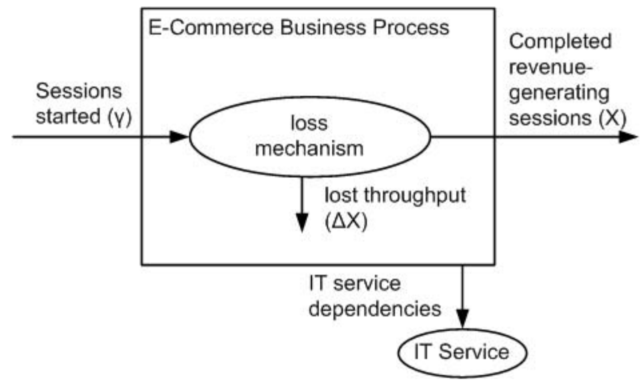

Fig. 3. E-commerce Business Loss

This concludes our analysis.

IV. 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 site has a revenue-generating CBMG as shown in Figure 2. For the non-revenue-generating CBMG, we do not provide a figure similar to Figure 2 but the transition probabilities are shown in Table IV. In practice, these transition probabilities for a given site can be gathered from web server log files. Solving these CBMGs as shown in section III-C yields the average number of visits (V_r^{RG} and V_r^{NRG}) shown in Table V. Observe that, for RG sessions, the Pay state is always visited (probability 1.0) whereas it is never visited in NRG sessions (probability 0.0). The IT infrastructure consists of three resource classes: web tier, application tier and database tier. For our study, the parameters shown in Table VI and Table VII are used, except where otherwise noted. The values for all input parameters are meant to be typical for current technology and were obtained from [1], [8], [10]. In Table VI, 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)). Table VII shows the average demand (in milliseconds) imposed by a transaction on the various tiers for each customer behavior state; recall that the actual service times are random variables with exponential distribution.

TABLE V
AVERAGE NUMBER OF VISITS TO EACH STATE

State (r)	RG Session (V_r^{RG})	NRG Session (V_r^{NRG})
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 VI
PARAMETERS FOR EXAMPLE SITE

Parameters	Values
T^{DEF}	8 seconds
ϕ	\$1 per transaction
γ	14 transactions per second
f	25%
ΔT	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)

Let us first try to design the site infrastructure in an ad hoc fashion, without business considerations, as is typically performed by an infrastructure designer. This is done by trying to minimize cost while maintaining reasonable service availability and response time. In the discussion that follows, a particular design is represented by the tuple (n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) which indicates the number of machines (total and load-balanced) in each of the three tiers. 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 a single 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.

TABLE VII
SERVICE DEMAND IN MILLISECONDS IN ALL TIERS

	CBMG state							
Tier	**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

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 III-D (see Table VIII). In that table, each line represents a different infrastructure design alternative; the first column indicates the infrastructure design being considered; the second column is the cost of this infrastructure; the third column represents the business loss (in \$) due to customer defections; the fourth is the business loss due to service unavailability; the fifth is the total financial commitment (cost plus business losses); finally, the last column indicates how much the business loses by adopting that particular infrastructure compared to the optimal one (described in the last line). All financial figures are monthly values.

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 (last column). Observe that an over-design can also be suboptimal. In this case, business loss can 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 [6].

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., $C(\Delta T)+L(\Delta T)$) 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

TABLE VIII
COMPARING INFRASTRUCTURE DESIGNS

Infrastructure design (n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})	Cost ($)	Response Time Loss ($)	Unavailability Loss ($)	Cost + Loss ($)	Cost of choosing wrong ($)
(5,5,2,5,5,2)	24,430	4,964,417	1,422,755	6,411,602	6,361,129
(6,6,3,5,5,2)	31,715	5,851,498	55,929	5,939,142	5,888,669
(7,7,4,5,5,2)	39,000	5,886,685	1,712	5,927,397	5,876,924
(8,9,5,6,6,3) (optimal)	**48,351**	**754**	**1,368**	**50,473**	**0**

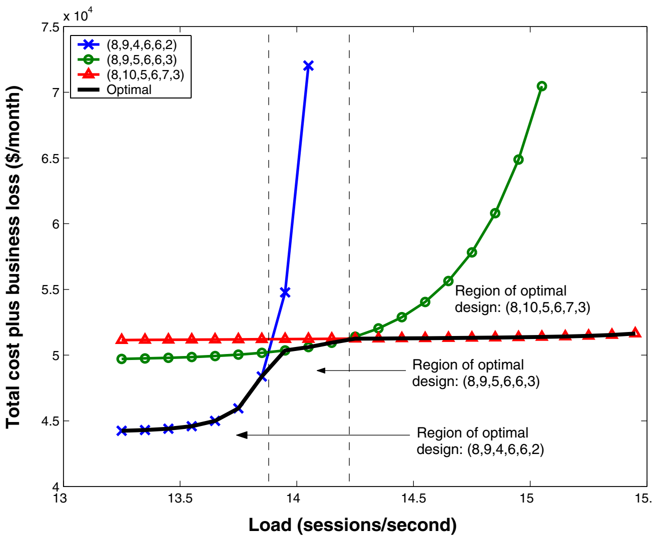

Fig. 4. Sensitivity of total cost plus loss due to load

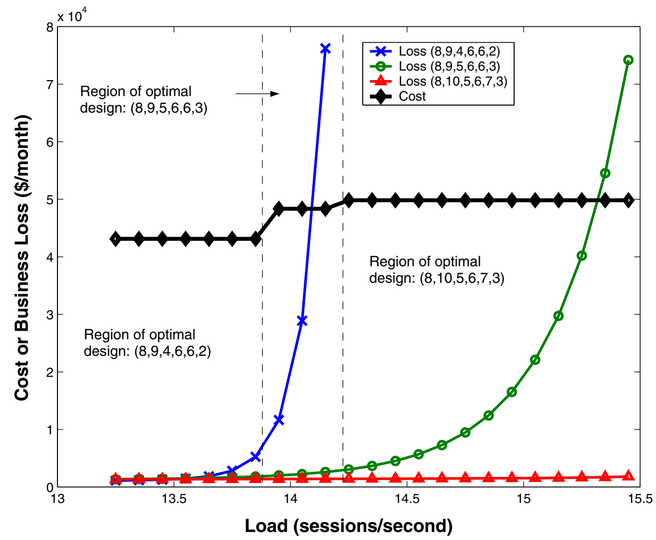

Fig. 5. Sensitivity of cost and loss due to load

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.

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). Since the load applied to the system varies with time, two situations can occur: in a static provisioning scenario, one can use the data shown in Figure 6 to choose the "best" design over the expected range of load. This will not be an optimal design for all load values but the designer has a tool to evaluate the cost of over-designing to handle load surges. On the other hand, in an adaptive infrastructure scenario, a dynamic provisioning algorithm can be used to trigger infrastructure changes to keep the design optimal at all load levels. The dashed lines in Figure 6 show where dynamic provisioning must trigger and the heavy black line shows the response time that is attained, a low value for all load levels.

V. Related Work

In the area of infrastructure design, [1] describes a tool – AVED – used for capacity planning to meet performance and

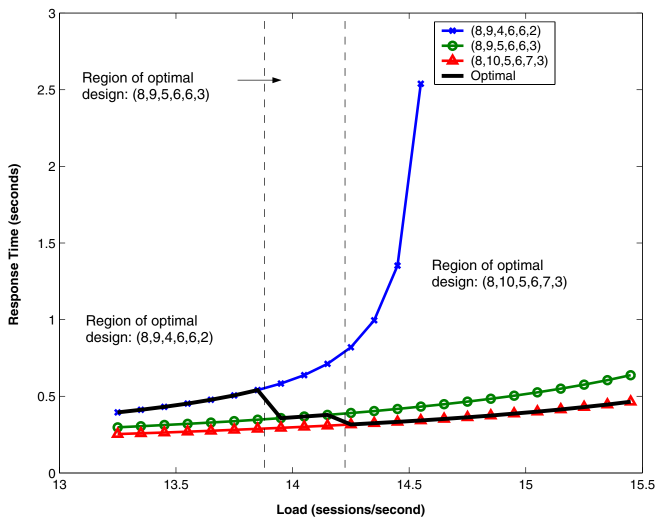

Fig. 6. Response time for various designs

availability requirements and [12] describes a methodology for finding minimum-cost designs given a set of requirements. Similarly, [2] 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 [1], [12].

Finally, [3] 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.

An initial version of the business impact model presented here appeared in [6]. The current work adds a different customer behavior model (CBMG [8]) and a new analysis of customer defection, as well as new conclusions concerning the sensitivity of the optimal design to changes in applied load. Furthermore, dynamic provisioning considerations are discussed here.

VI. Conclusions

In summary, a method was discussed to design IT infrastructure from a business perspective. The method is novel in that three types of metrics are considered and combined – 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 [6], etc.

We offer the following conclusions:

1) Ad hoc infrastucture design – a process in which business considerations are not formally taken into account – can yield suboptimal designs causing significant business loss.
2) Overdesign to satisfy very stringent SLA requirements can also yield suboptimal designs due to their high cost.
3) The approach taken here can also be used to choose appropriate Service Level Objectives (SLOs) when designing SLAs. Rather than choosing SLO values a priori, the values are simply *calculated* from the optimal design obtained by the process described here.
4) The optimal infrastucture depends on the importance of the business processes being serviced: a business process generating more revenue will merit larger outlays in infrastucture. The method presented here shows how much should be spent to provision services for each business process.
5) Infrastrucure designs are optimal over a range of input load; however, we have found that this range is typically small and that response time and resulting business losses can quickly grow when load varies significantly.
6) The method shown provides clear trigger points for dynamic provisioning and also offers a way of calculating what the infrastructure design should be for a given load.

In the future, we plan to develop new impact models applicable to business processes other than e-commerce (say, manufacturing, CRM, etc.). In addition, the approach can be used to investigate more general enterprise architecture scenarios; this will require the development of more holistic models that include the network and other components outside the data center, as well as more detailed enterprise architecture components. Finally, a fuller study of the use of business impact models in adaptive environments can be undertaken; this will be an expansion of the initial comments given here concerning dynamic provisioning.

Acknowledgments

We would like to acknowledge and thank the Bottom Line Project team (http://www.bottomlineproject.com/). This work was developed in collaboration with HP Brazil R&D.

References

[1] G. J. Janakiraman, J. R. Santos, and Y. Turner, “Automated multi-tier system design for service,” in *First Workshop on Design of Self-Managing Systems*, 2003.
[2] D. A. Menascé, D. Barbará, and R. Dodge, “Preserving QoS of e-commerce sites through self-tuning: a performance model approach,” in *EC ’01: Proceedings of the 3rd ACM conference on Electronic Commerce*. New York, NY, USA: ACM Press, 2001, pp. 224–234.
[3] D. Gilat, A. Landau, and A. Sela, “Autonomic self-optimization according to business objectives,” in *ICAC ’04: Proceedings of the First International Conference on Autonomic Computing (ICAC’04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 206–213.
[4] M. Sallé and C. Bartolini, “Management by contract,” in *NOMS 2004: IEEE/IFIP Network Operations and Management Symposium*. IEEE Computer Society, 2004, pp. 787–800.

[5] C. Bartolini and M. Sallé, “Business-driven prioritization of service incidents.” in *DSOM*, ser. Lecture Notes in Computer Science, A. Sahai and F. Wu, Eds., vol. 3278. Springer, 2004, pp. 64–75.

[6] J. P. Sauvé, F. T. Marques, J. A. B. Moura, M. C. Sampaio, J. F. H. da Jornada, and E. Radziuk, “SLA design from a business perspective,” in *16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, ser. Lecture Notes in Computer Science. Springer, 2005.

[7] K. S. Trivedi, *Probability and statistics with reliability, queuing, and computer science applications*. Prentice-Hall PTR, 1982.

[8] D. A. Menascé and V. A. F. Almeida, *Scaling for E-Business*. Prentice Hall, 2000.

[9] L. Kleinrock, *Queueing Systems, Volume I: Theory*. Wiley-Interscience, New York, 1976.

[10] D. A. Menascé, V. A. F. Almeida, and L. W. Dowdy, *Performance by Design: Computer Capacity Planning by Example*. Prentice-Hall PTR, 2004.

[11] J. P. Sauvé, F. T. Marques, J. A. B. Moura, M. C. Sampaio, J. F. H. da Jornada, and E. Radziuk, “Optimal design of e-commerce site infrastructure from a business perspective,” Tech. Rep., 2005. [Online]. Available: http://jacques.dsc.ufcg.edu.br/projetos/bl-hp/reports/004-2005.pdf

[12] D. Ardagna and C. Francalanci, “A cost-oriented methodology for the design of web-based it architectures,” in *SAC ’02: Proceedings of the 2002 ACM symposium on Applied computing*. New York, NY, USA: ACM Press, 2002, pp. 1127–1133.

URLs last checked in August, 2005.