

IBM Research Report

Capacity Planning Tools for Web and Grid Environments

Sugato Bagchi, Eugene Hung*, Arun Iyengar, Norbert Vogl, Noshir Wadia*

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

*IBM Santa Teresa Laboratory
San Jose, CA 95141



Research Division

Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Capacity Planning Tools for Web and Grid Environments

(Invited Paper)

Sugato Bagchi¹, Eugene Hung², Arun Iyengar¹, Norbert Vogl¹, and Noshir Wadia²

{bagchi,eyhung,aruni,vogl,noshir}@us.ibm.com

¹ IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

² IBM Santa Teresa Lab, San Jose, CA 95141, USA

Abstract

A key aspect in managing resources for customer sites is to predict and assess the load associated with a site in order to figure out how best to allocate resources for the site over time and to efficiently schedule tasks. The cost associated with the site and return on investment are also key parameters. This paper describes work we have done in developing tools for answering these critical questions. The tools use both analytical models and discrete event simulations to predict performance and analyze costs needed for handling a customer workload while satisfying the service level objectives. These tools provide capacity and load planning, performance simulation, and cost and financial analyses. Our tools have been used successfully by several major customers, and those experiences have shaped how the tools have evolved over time.

Categories and Subject Descriptors

C.4 [Computer Systems Organization]: Performance of systems; B.8 [Hardware]: Performance and reliability; D.4.8 [Software]: Operating Systems—*Performance*; I.6 [Computing Methodologies]: Simulation and modeling

General Terms

Performance, Design

Keywords

capacity planning, grid computing, performance modeling, Web performance

1 Introduction

Capacity planning is critically important for managing computer systems. System administrators need to be able to predict how load

will vary over time. Loads tend to fluctuate over time. As a workload changes, the system requirements for handling the workload may change as well. It is important to have enough hardware to handle workload increases and surges. If a workload decreases after a peak, it would be desirable to have the ability to reduce the hardware servicing the workload; this would allow the hardware to be used for other purposes.

Having tools for predicting capacity is important. For a given workload, the tools should be able to predict the hardware and software requirements needed to handle it. This paper presents an overview of a set of capacity planning tools developed at IBM. The capacity planning tools have been successfully deployed by multiple customers.

The capacity planning tools are particularly applicable to commercial applications which are transactional in nature. They have been quite successful with a number of customers serving large amounts of dynamic Web data. Our tools are also designed for grid environments in which workloads may be continually changing and load may be shifted among multiple processors.

Grid computing [10] has made it possible to virtualize aspects of IT infrastructure design by mediating the linkage between application workloads and infrastructure hardware such as servers and storage. This technical advance is in alignment with business interests for an on-demand operating environment that is responsive to unexpected surges in workload demands and resilient to resource outages.

These business requirements and technology advances have outpaced the IT modeling, design, and valuation tools that are currently in use. For example, the shift from dedicated to virtual and flexible computing resources places new requirements for IT capacity planning and resource optimization. The increased business focus on designing responsive and resilient infrastructure calls for the need to simulate random events and their impact on the infrastructure and to measure these characteristics. Along with these technical considerations, the shift from fixed ownership costs to variable usage-based costs requires a tighter coupling between technical design and financial analysis of the returns from the infrastructure investment.

This paper describes two sets of tools for capacity planning and performance analysis. The first set of tools, known as the On Demand Performance Advisor (OPERA), uses analytical models to predict performance. It can predict solutions quickly and can be used for both offline and online capacity planning due to the fast response times. The second set of tools, known as Grid Value at Work, uses discrete event simulation. The use of discrete event simulation can result in a higher degree of accuracy than analytical

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Valuetools'06 October 11-13, 2006, Pisa, Italy
Copyright 2006 ACM 1-59593-504-5 ...\$5.00

models. However, more information must be supplied to the discrete event simulator, and response times are considerably slower. The two approaches are complementary. The output of OPERA can be imported into Grid Value at Work for simulation and cost analysis.

A key differentiating feature of our tools compared with other tools is that our tools make use of considerably more information regarding customer workloads as well as performance characteristics of real platforms. This makes our tools useful in environments in which the application is known but limited information about specific characteristics of the workload are known.

In the next section, we describe the key features of OPERA. We then describe Grid Value at Work. This is followed by a results section where we quantify how well our tools have worked for real customer workloads.

2 Key Features in a Capacity Planning Tool

We now describe key features of the On Demand Performance Advisor (OPERA). OPERA was first released in 2001 and has undergone several enhancements since then. OPERA includes a performance estimator which displays performance results in sufficient detail to allow users to assess the adequacy of a given configuration for their requirements, and to provide insight into where the bottlenecks are likely to occur. This allows the performance advisor to be useful for planning capacity, evaluating infrastructure/workload changes, estimating scalability, and reducing site costs.

Another key feature of OPERA is the configuration estimator which can determine a near-optimal machine configuration for a given workload and hardware brand. If the site consists of multiple tiers, the advisor enables users to adjust the number of tiers as necessary. For a Web-based system, a two-tier architecture might contain Web servers in the first tier with application servers in the second tier. A three-tier system might contain a third tier of database servers.

Figure 1 illustrates a typical customer configuration which OPERA is designed for. In this example, clients are communicating with a three-tier server over the Web. The first tier, the Web presentation server, serves content to the clients. The second tier, comprising Web application servers, processes requests received by the Web presentation server and generates client responses. The Web application server may invoke the database tier. Each tier may contain multiple processors. The edge server could be from a content distribution network.

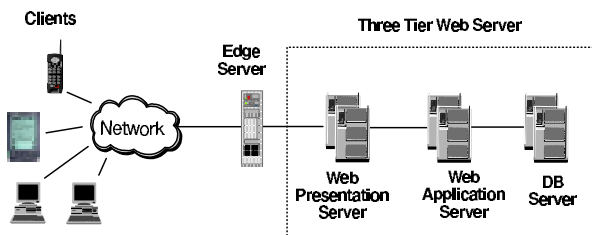


Figure 1. A three-tiered system commonly used for Web sites.

Note that the overall system may also contain other components not shown. For example, there could be load balancers at multiple places such as in front of the Web presentation server tier. Several clients could be behind a proxy with a proxy cache reducing the

latency for fetching remote contents. Caches could exist at multiple places within the system.

OPERA also contains a hardware and software library with performance characteristics of several different hardware platforms, operating systems, and software such as IBM's WebSphere. This library allows performance characteristics to be predicted for different platforms. It also allows users to determine how performance is affected after switching from one hardware platform to another. The presence of the library allows users to predict performance without having to enter performance characteristics of the hardware or operating system platforms being used.

OPERA has a number of predefined workload types for commercial applications, including online shopping, online trading, online banking, business-to-business, inventory management, online brokerage, online auctions, and a number of others. Users who have workloads falling into these categories can use OPERA's predefined workloads to get reasonably accurate results without having to enter detailed characteristics of their individual workloads. For workloads which don't fall into one of these categories, users can define their own workload types.

OPERA also has features for handling numerically intensive scientific workloads in addition to commercial transactional workloads. It can also handle multiple concurrent workloads which vary over time.

2.1 Methodology used by OPERA

OPERA is typically deployed by using the following steps (see Figure 2):

1. *Identify workload pattern of application.* Typical customers have transactional workloads which are high-volume and growing, and contain significant amounts of dynamic data. Beyond that, other characteristics must be considered, such as transaction complexity, data volatility, and security. If the application workload is similar to one of the predefined workload types, then the closest pattern should be selected. Otherwise, a user-defined pattern characterizing the data is selected.
2. *Measure performance of current site.* To plan for the future, data from the present needs to be fed into OPERA. Site characteristics such as volumes (e.g., hits, page views, transactions, searches), arrival rates, response times by class, user session time, number of concurrent users, and processor/disk utilization should be measured or estimated and fed into OPERA.
3. *Analyze trends and set performance objectives.* In this step, trends are analyzed to estimate future workloads. After analysis, objectives are set for each metric that needs sizing along with any new metric that applies to future requirements.
4. *Model infrastructure alternatives.* The user has the option of entering a specific hardware and software configuration (either current or hypothetical) to get an estimate of the configuration's performance. Alternatively, the user can ask OPERA to estimate the optimal configuration, with appropriate best-practices rules applied.

OPERA forecasts performance using analytic models based on enhanced M/M/k queuing models [5]. In this technique, users are iteratively added to the system in an incremental fashion. Then, for

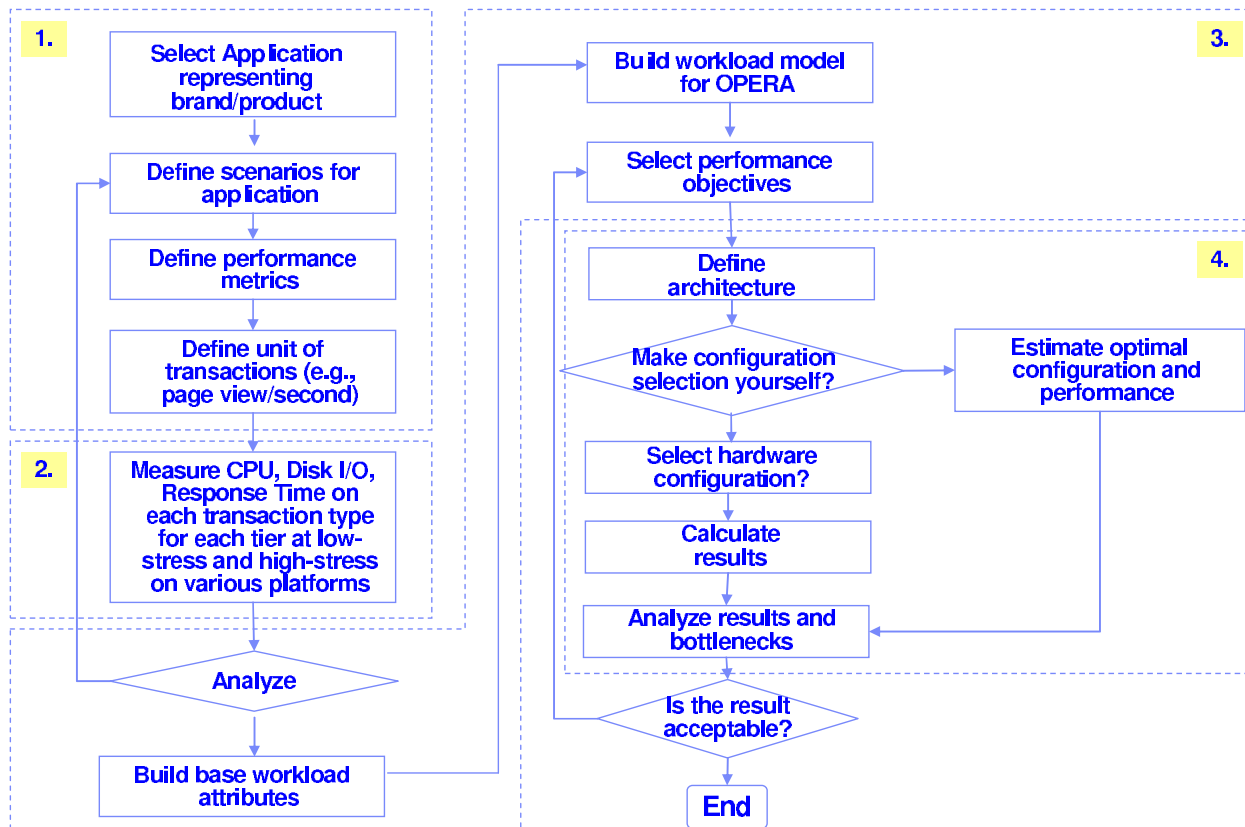


Figure 2. A typical methodology for using OPERA.

each server in the topology, the delay d of that server is the sum:

$$d = T_S + T_W$$

where T_S is the service time calculated from the measured data, and then estimated for the target server based on the established relationships between the measured and target systems; and T_W is the wait time obtained from the M/M/k queueing model. In a system with multiple tiers, each tier is treated separately with a different M/M/k queueing model. Then, the total response time is calculated by adding up the delays for each tier.

Two complete sets of calculations are performed. The base set of calculations uses built-in (or user-provided) measured data for CPU and disk I/O. A more conservative base-plus-contingency set of calculations is also performed by adding a user-supplied contingency factor to each of the measured data values before the calculation. Both results are then transformed to the target configuration using built-in scaling coefficients taken from industry standard benchmarks and measurements.

At each step the calculated results are compared against the user-selected performance target(s) to determine if a target has been reached or if an early resource depletion (CPU or disk bandwidth) has occurred in any component of the infrastructure being evaluated. Resource depletion events signal the need for configuration adjustments and bring all calculations to a stop. OPERA provides the user with information about which tier constitutes the bottleneck as well as several performance metrics. Results can be displayed using both tables and graphs.

OPERA also has modified algorithms based on G/M/K queueing models to estimate the performance impact of burstiness. Burstiness indicates the relative ratio between peak rates and average rates. It may be caused by unpredictable events such as major stock market swings or special events such as Christmas or Valentine's Day. The user can specify a burst-to-peak ratio to increase the accuracy of OPERA's calculations.

3 Performance Modeling and Analysis using a Simulation Tool

We now describe the Grid Value at Work simulation tool. While OPERA uses analytical models, Grid Value at Work uses discrete event simulation. This requires more information about customer request distributions but often results in greater prediction accuracy at the cost of a longer execution time. Both approaches are important for customers. Grid Value at Work is an infrastructure modeling and analysis tool for IT consultants and architects. The tooling environment supports a number of analyses modules which are selectively used by IT consultants and architects, depending on the situation of the customer along the grid sales, delivery, and deployment cycle.

The Grid Value at Work modeling environment is designed for the Eclipse platform [9] with a common XML data model which forms the basis for integration between the various analysis capabilities that are developed as Eclipse plug-ins. Figure 3 shows the analysis plug-ins that have been developed. These and additional plug-ins can be developed and used independent of each other. A common user interface may optionally be available for the analyses. Both the

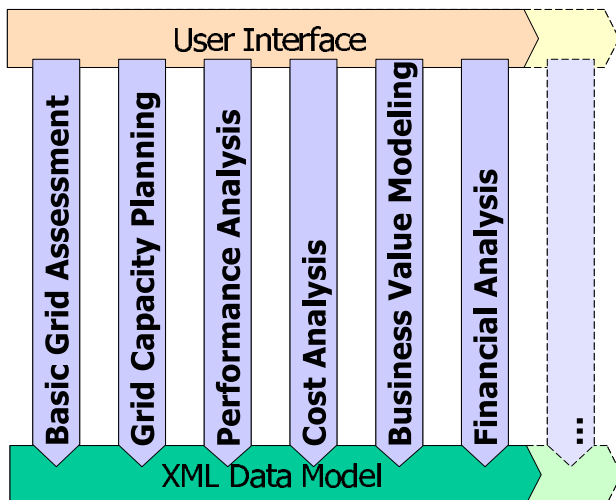


Figure 3. Grid Value at Work Architecture

grid data model and user interface are extensible to accommodate future analysis modules. In addition, other tools could be loosely coupled with Grid Value at Work by designing them to generate an output in a format compatible with the grid data model.

3.1 Grid Capacity Planning

Capacity planning results from OPERA are transformed into the Grid Value at Work data model in order to implement the capacity planning capability for grid environments. This may then be evaluated with the Performance Simulation plug-in under various conditions, such as grid scheduling policies, workload surges, resource outages, and non-exponential probability distributions, that were not considered by the queueing analytics used in OPERA. Another benefit of integrating OPERA with Grid Value at Work is the availability of infrastructure costing capabilities using the Cost Analysis plug-in.

3.2 Performance Simulation

The objective of the performance simulator analysis module of Grid Value at Work is to estimate the performance of processing application workloads on a grid. Simulation can be performed once the grid infrastructure capacity has been defined in terms of the number and types of resources on the grid and their availability schedules. This information could be generated from a capacity planning analysis or through manual estimation and design. The resource requirements and expected arrival rates for the application workloads must also be known.

With this information, a discrete-event simulation of the grid can be performed to test various elements of its design:

1. The type of IT resources (e.g., computational servers, database servers, storage and network capacities) available on the grid: their quantities and availability profile over time.
2. The grid resource allocation policies that determine how many grid resources to allocate for the workloads over any time interval.
3. The grid scheduling and dispatch policies that determine the

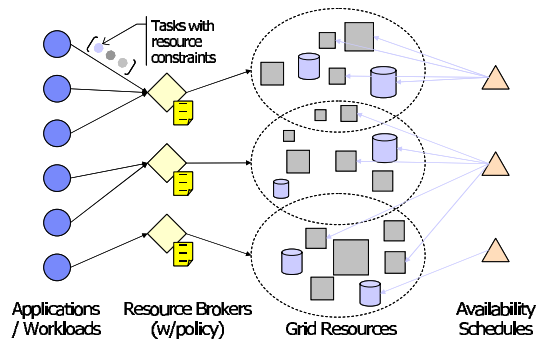


Figure 4. Operational Flows in a Grid Environment

resources to which each arriving job should be routed for processing.

4. The workload job arrival rates and patterns over time. Alternate design scenarios, created by varying any combination of these elements, may be evaluated and compared on the basis of the outputs from the simulation analysis.

Figure 4 shows the components in the simulation model of a generalized grid environment.

Application workloads send jobs for processing on the grid. Each job consists of a sequence of tasks with specific resource constraints. These tasks are sent to a resource broker which uses its scheduling policy to decide how to route it to a resource for processing. The availability of a resource is subject to an availability schedule which specifies how many resource units are available at any point in time for processing a certain type of task. The simulation outputs provide various performance metrics on the components of the grid:

- For grid resources:
 1. the utilization of the available resources on the grid,
 2. the number of jobs processed by each resource, and
 3. the average processing time.
- For application workloads:
 1. the average and peak response time for the jobs from each application workload,
 2. the number of jobs processed and aborted due to lack of available resources.

See [2] for a detailed description of the grid simulation environment including its unique requirements and challenges.

3.3 Cost Analysis

Cost analysis estimates the cost of ownership and usage of the grid infrastructure being designed. It could be used for cost comparison of alternate grid and non-grid infrastructure designs against the client's current infrastructure environment and its upgrade plans. The degree of detail in the cost model is expected to vary widely across clients. Some may want detailed costing of each software installed on servers while others want a high-level rule of thumb for server costs. Regardless of the degree of detail, costs have the following characteristics:

- *capital or expense*: this is used in the financial analysis to understand whether there is a depreciable asset backing up the cost or whether it is a business expense.
- *fixed or variable*: specifies whether the cost is dependent on another metric (e.g., number of servers, people, etc.) or the cost is fixed (e.g., an enterprise-wide license for DB2).
- *one-time or periodic*: specifies whether the cost is incurred just once, as with an outright purchase of an asset, or periodic (e.g., monthly lease of equipment, salaries, maintenance fees).

The cost inputs are defined in terms of cost drivers and cost elements. A Cost Driver is a factor that influences the variable cost of acquiring or operating a resource. A trivial cost driver is the number of such resources. Other cost drivers could be people (e.g., system administrators) required to manage IT resources, the person-hours required for grid design, the annual energy (kilowatts) consumed by these resources, and the floor space occupied by them, etc. A Cost Element describes a unit of cost that is associated with acquiring or operating the IT infrastructure and its associated cost drivers. It can be a high-level aggregate cost (e.g., annual cost of ownership of a “Wintel” server) or as detailed a line-item as necessary to match client requirements or the data availability. For example, the per-unit cost of purchasing a pSeries 650 server could be \$26,895, the annual fully-burdened cost for a system administrator could be \$150,000, the hourly cost of system integration could be \$250, etc.

The cost analysis plug-in combines these cost drivers and elements to compute the actual cost line-items for each infrastructure scenario by year. These are then rolled up into a combined cost per year as well as their net present value. Cost analysis can be performed on multiple scenarios at the same time. Optionally, the cost of each scenario may also be compared against a “base” scenario (for example the as-is IT environment). A negative cost in the cost comparison for a scenario signifies cost savings compared to the base.

3.4 Business Value Modeling

Business value analysis estimates the financial value that could be added to the business as a result of improvements in the performance of applications running on the grid. The performance improvement may be estimated by the performance simulator described above or predicted through other means. Application performance may be defined in terms of increase in job throughput, decrease in job response time, or both. The financial value could be specified in terms of savings in the operational costs of the business process impacted by the application or in terms of profits from additional revenue.

The calculation of financial value requires the development of a business value model that links application performance to financial value. The inputs to the model are application workload metrics such as response time and throughput as well as other business process metrics. These inputs impact other business process metrics, whose values can be calculated from the values of their input metrics. Ultimately, these chains of impact lead to impacts on financial metrics such as operating costs and revenue, from which the impact on the net income of the business can be estimated.

Users of the tool are expected to use existing business value models or develop their own based on application and industry-specific knowledge and assumptions. The tool can model multiple scenarios of business values, reflecting alternate views of future external and internal conditions as well as alternate strategies of deriving busi-

ness value from an improvement in the application performance. For example, the reduced time taken to perform a computation could be leveraged either in terms of faster response time or being able to perform more computations in the same time as before, but with an improvement in the quality or accuracy of the output.

3.5 Financial Analysis

The financial analysis module develops a business case for moving from one IT infrastructure scenario to another. It summarizes the IT cost and value analyses done by the Cost Analysis and Business Value Modeling plug-ins described above. The capital IT investment required for the “to-be” scenario is determined and compared with the benefits from that investment in terms of (a) savings in IT expenses and (b) added value at the business level. From this, a set of financial metrics are calculated, which will enable a financial decision-maker to compare the grid investment with other potential investments that could be made.

The financial metrics for measuring returns from a project may be classified into two groups based on the means used to recognize income.

1. The *accounting measure of income* uses generally accepted accounting principles (GAAP), account statement measures and standards. In this category, the financial metrics that aid decision making about an investment are:
 - *Return on Capital (ROC)*: This is the return from the project, measured as earnings before interest and taxes (EBIT), as a ratio of the average book value of the total capital investments in the project.
 - *Economic Value Added (EVA)*: This is a measure of the value added to the firm as a result of investing in the project.
2. The *cash flow* measure, calculated as the difference between the actual cash inflows and outflows. In this category, the financial metrics that aid decision making about an investment are:
 - *Return on Investment (ROI)*: This is the ratio of all the cash benefits over all the cash investments, over the duration of the project. It does not discount future cash flows into present value terms. That is, a dollar spent today is considered to be worth the same as a dollar to be spent in the future.
 - *Payback Period*: This is a measure of the time taken for the benefits from the investment to cover the initial investment. This measure too does not discount future cash flows.
 - *Net Present Value (NPV)*: This is the difference in the present values of all cash inflows due to the project and all the cash outflows. The present value of a future cash flow is determined by discounting it based on how far it is into the future and a discount rate which is a measure of the project risk and interest rates. A project with a positive NPV should be considered for acceptance.
 - *Internal Rate of Return (IRR)*: This is the discount rate at which the NPV of the project becomes zero. This rate may be compared by the decision-maker to the business cost of capital to decide whether the project should be accepted.

A separate business case could be built for each of the alternate “to-

be” infrastructure scenarios being considered, allowing an objective comparison between them on the basis of the financial metrics of the business cases.

4 Results and Usage Scenarios

The capacity planning tools described in this paper have been used successfully by several major customers mostly in the area of designing highly accessed Web sites which serve considerable amounts of dynamic data. The tools have been a significant factor in how these customers have chosen to design their systems. We have also done performance tests to determine how accurately the performance tools model actual workloads. Predicted values for throughput and system utilization are generally off by less than 10%. When request rates are varied across a single configuration, average response times are generally off by less than 10%. When a multi-tiered system is used in which the number of servers in different tiers is varied considerably, we do see some larger than expected variations in actual response times.

Figure 5 compares actual performance to the performance predicted by OPERA for two different customers. Both had a Web server and application server tier. The second customer also had a database tier. In the first two sets of bars, transactions per second and number of users represent the number of transactions per second and number of users the system could accommodate while meeting the necessary service level objectives. The next three bars represent response times, percent utilization of the Web server tier, and percent utilization of the application tier during steady state execution of the customer application. In the second graph, the last pair of bars represent the percent utilization of the database tier. The graphs indicate that OPERA is usually off by less than 10%.

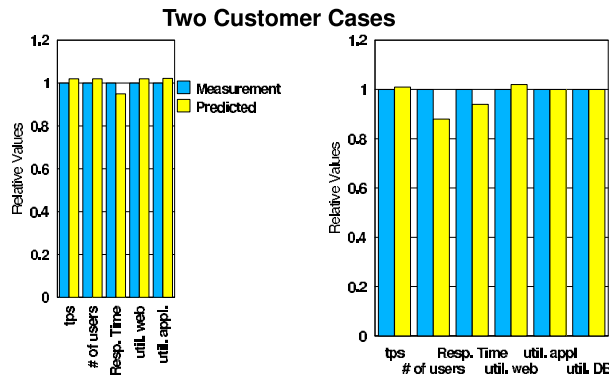


Figure 5. Comparison of actual and predicted performance measurements for two different customer workloads. The text clarifies the meaning of the bars.

We now give a set of more detailed results. Figure 6 shows the architecture used to perform the experiments. Requests are directed from a load balancer to a Web server front end which is executing servlets. The requests may then be passed to a business logic tier which is executing Enterprise JavaBeans (EJB). The business logic tier may access the database tier. The throughput for both the Web server (servlet) tier and the business logic tier are shown in the figure. Both throughputs scale linearly with the number of nodes.

Figure 7 shows the actual and predicted CPU utilization percentages for both the business logic and Web server tiers as well as the actual and predicted average response times and page throughputs

as the number of clients making requests to the site is scaled. Both figures show that OPERA does a reasonably accurate job of predicting performance.

Figure 8 show the measured and predicted CPU utilization for the business logic (abbreviated BLT in the figure) and Web tiers as well as average response times and page throughput as the number of nodes is scaled. While OPERA generally did a good job of predicting performance, there were a few anomalies in the measured response times which resulted in a greater difference from the predicted response time than expected.

4.1 Grid Value at Work Usage Scenario

We now present a representative scenario illustrating the use of Grid Value at Work. The client is a national retail bank. Its marketing and sales department mines data about consumer household credit history and purchasing patterns to identify those likely to accept one of the bank’s product offerings.

Faced with increasing demand for data-mining, the bank has 3 scenarios before it:

1. It can do nothing, keeping their existing IT infrastructure and thereby forgoing the benefits, if any, from increasing the data mining throughput.
2. Upgrade their existing IT servers to handle the increased throughput.
3. Create a grid of existing desktop computers in the bank.

The bank wants to develop business cases for the transition options (1 to 2 and 1 to 3) based on the infrastructure costs for each scenario and the value add from increasing the data mining throughput. In order to estimate the throughput increase, the IT performance of each scenario must be simulated.

The data-mining application is performed with the SAS statistical software package that reads in consumer records from a database and then performs a statistical analysis. The application can therefore be divided into two tasks: accessing a database and SAS analysis, each requiring a different type of IT resource for its processing.

The first task retrieves 1000 rows per consumer from the database. The existing database system has a service rate of 10000 rows/second. We use a custom performance benchmark of transactions/second (TPS) and give the existing database system a rating of 10000. The second task runs the statistical analysis on SAS engines (computers). The analysis for each consumer takes a second to perform on a server (Sun Fire V480) with an industry standard SPECfp2000 benchmark of 637. The SPECfp2000 benchmark was chosen because the analysis is CPU intensive with floating point operations. The analysis for each consumer may be done independently. Therefore, the task may be split up into any number of parallel sub-tasks.

The IT infrastructure for running the data mining operations consists of a 2-CPU Sun Fire V480 server that performs the SAS analysis. This has a SPECfp2000 benchmark rating of 637 per CPU which is appropriate for the compute-intensive floating point workload it processes as the SAS Engine.

The consumer data is maintained in a Sun Enterprise 10000 database server. As previously mentioned, this can support 10000 transactions/second.

Real Customer Example Online Trading

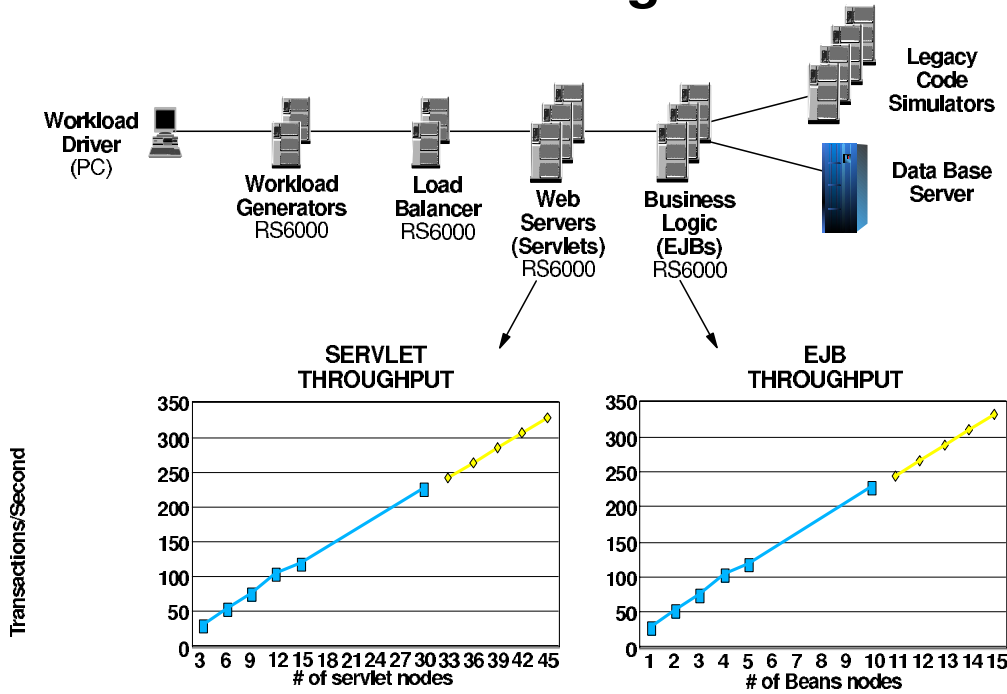


Figure 6. On-line trading system used for comparing actual to predicted performance.

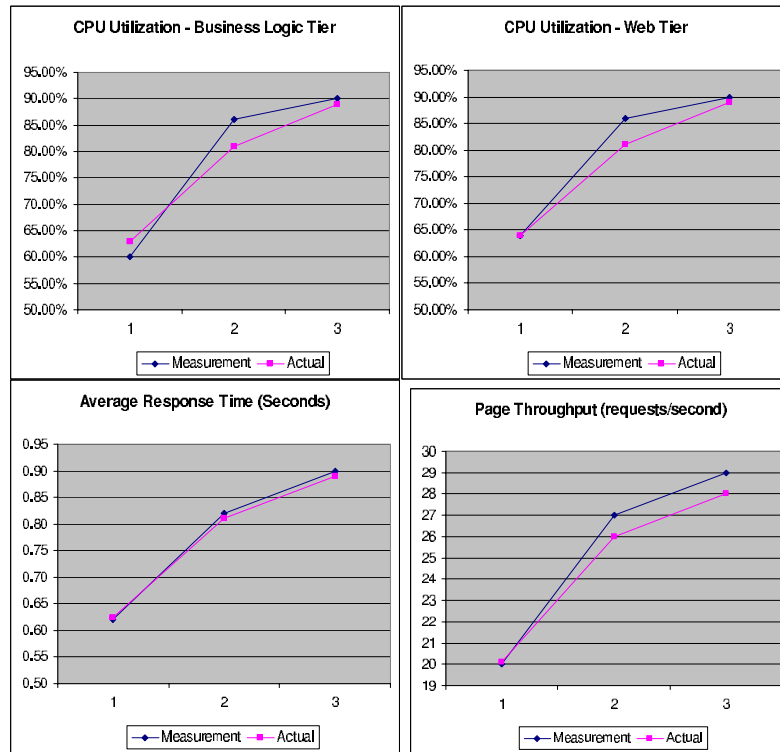


Figure 7. Actual and predicted performance metrics as the number of clients is scaled by factors of 2 and 3. The X-axis represents the normalized number of clients.

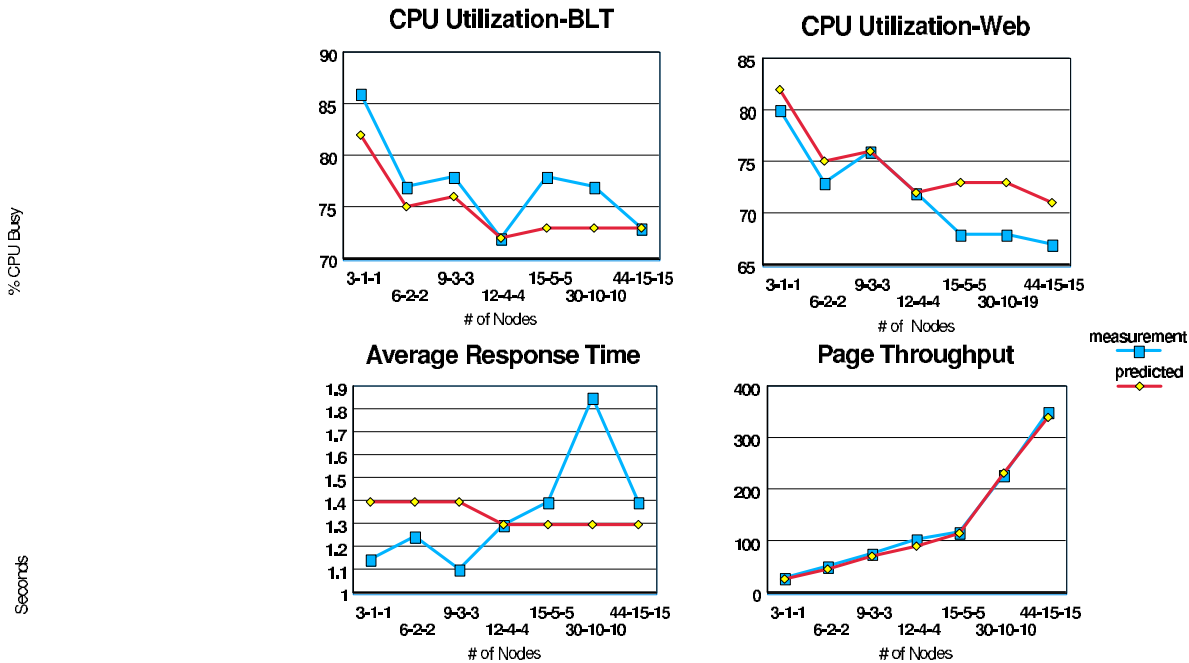


Figure 8. Actual and predicted performance as the number of nodes is varied. The triplets below each graph represent the number of nodes in each of the three tiers.

For future infrastructure scenarios, we are looking at two alternatives: using a set of Wintel desktops for the SAS analysis or upgrading the existing Sun Fire server to a newer Sun server. In either case, the database server will not be changed because it is not yet being perceived as a bottleneck. The Wintel desktops are a mix of Pentium IIIs and Pentium 4s with an average SPECfp2000 of 450. The Sun upgrade being considered by the bank is a 4-CPU Sun Fire V880 with a SPECfp2000 rating of 923 per CPU.

Based on these client requirements, we need to develop a grid model with enough information for performing the following analyses:

- *performance simulation* - for quantifying the throughput improvements
- *cost analysis* - for comparing infrastructure costs in each scenario
- *business value analysis* - for estimating financial impact of increased throughput
- *financial analysis* - for developing business cases for infrastructure transitions

The objective of the simulation is to evaluate the performance in terms of response time of the data-mining application and the utilization of the computing resources. As required by the client, we consider two levels of throughput: the current workload of 1500 arrivals every 15 minutes for the as-is system and the high-volume workload of 5000 arrivals every 15 minutes for the future IT infrastructure scenarios.

The following table shows the results of simulation of workload arrivals over a 24 hour period for the three IT scenarios. The grid scenario has 10 desktop servers shared over a grid while the non

grid scenarios have one dedicated server as described above.

Table 1. Grid Simulation Results

Scenario	Volume (per day)	Response Time (seconds)	Server Util
As-is	142500	900	82.4%
Desktop Grid	470000	1207.76	77.5%
Non-grid Upgrade	470000	1362.65	94.3%

The impact of varying the number of desktop servers on the grid can be seen in Figure 9. This figure may be used to determine the appropriate number of desktops to be used, based on the desired response time of the workload or maximum utilization levels of the desktops. The response time decreases as the number of desktops is increased because more workloads can be processed in parallel. However, after 50 desktops, there is no further improvement in throughput. This is because there is not enough of the arriving workload (assuming a minimum split size of 100 rows/server) to be allocated to all the available desktops.

Once the technical performance characteristics of the grid environment is determined for each infrastructure scenario being considered and we receive validation about their technical feasibility, we can use the Grid Value at Work financial plug-ins for estimating and comparing their costs and contributions to business value. Cost analysis calculates the capital investment required for each to-be scenario (including grid middleware and implementation costs) and the difference in IT expenses compared to the as-is expenses. The added business value from increasing the throughput of data-mining was evaluated by the bank marketing experts. Table 2 summarizes the results of these analyses in terms of the typical financial measures and ratios used to evaluate competing business case scenarios, as described in Section 3.5.

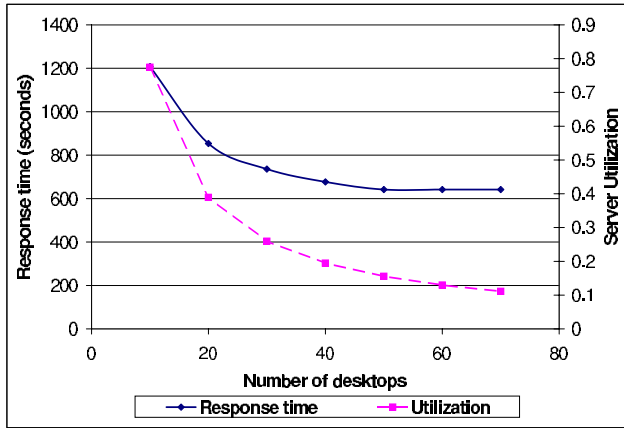


Figure 9. Plot of workload response time and server utilization vs. number of servers on a grid

Table 2. Business Case Comparison: Grid vs. Non-grid upgrade

Scenario	ROI	Payback Period	NPV	IRR	ROC	EVA
Desktop Grid	307%	23 mo.	\$111,379	109%	38%	\$22,305
Non-grid Grid	182%	35 mo.	\$50,246	40%	20%	\$12,073

5 Related Work

COMPASS is a performance modeling and analysis tool designed to assist users in capacity planning [17]. Daniel Menasce and his colleagues have written a number of books on capacity planning including [14]. The Transaction Processing Performance Council (TPC, www.tpc.org) and Standard Performance Evaluation Corporation (SPEC, www.spec.org) maintain a number of standard benchmarks which can be used for testing the performance of a variety of systems. TPC-W and SPECweb2005 are particularly relevant for Web-related workloads. Two other benchmarks representative of auction sites and bulletin boards, modeled after ebay.com and slashdot.org respectively, are described in [1].

A key differentiating feature of our tools compared with other tools is that our tools make use of considerably more information regarding customer workloads as well as performance characteristics of real platforms. This makes our tools useful in environments in which the application is known but limited information about specific characteristics of the workload are known.

There have also been a number of tools designed to generate representative Web workloads. SURGE generates Web workloads matching empirical measurements of server file size distribution, request size distribution, file popularity, embedded file references, temporal locality of reference, and idle periods of individual users [4]. SURGE is mainly applicable for static Web content, however. For many commercial Web sites, the real performance problems are created by dynamic data. Methods for generating Web traffic are studied in [3]. The S-client approach proposed in the paper can generate requests at a higher rate than the server can handle. Mercury LoadRunner (www.mercury.com/us/products/performance-center/loadrunner/) is a tool for testing performance that works by emulating clients to generate a workload to test an actual system.

It then monitors the performance of the system on the emulated workload in order to determine performance characteristics of the system. By contrast, our methodology focuses on estimating performance via modeling and simulations.

A considerable amount of research has been done in the area of Web workload characterization. The results from this research can be used to generate realistic workloads for benchmarking performance. A retrospective study of Web workload characterization over a ten year period is contained in [16]. Characterizations of highly accessed Web sites which serve considerable amounts of dynamic data are contained in [11, 8]. Models for capturing the characteristics of dynamic workloads are studied in [15].

Several papers have been published which analytically model different Web server architectures. An analytical model of a three tiered Web serving system consisting of a Web server, application server, and database server is presented in [13]. While the three tiers are similar to those depicted in Figure 1, the paper doesn't discuss how the number of servers can be varied to scale each tier.

Tools have been developed for modeling and simulation of grid infrastructure, but lack many of the requirements of commercial environments. GridSim [6] was designed for the modeling of heterogeneous, multi-tasking grid resources, calendar based resource provisioning, parallel application models, and resource scheduling. It has been used to evaluate the performance of various resource scheduling algorithms based on deadline and budget based constraints. The key drawback is the scalability of GridSim in tackling enterprise level simulation requirements where thousands to millions of grid tasks may have to be simulated.

SimGrid [12] has a heterogeneous and multi-tasking resource model. A key strength of SimGrid is to model the grid network topology and simulate the data flow over the available network bandwidth. However, it does not incorporate a realistic view of grid application workloads by not modeling job decomposition and task parallelization characteristics.

OptorSim [7] models dynamic provisioning and resource scheduling policies. It focuses on analyzing the interaction of these policies and estimating the resulting performance. However, its focus on the data aspect is at the expense of the other computational requirements that are also usually provided from a grid.

Common off-the-shelf simulators which provide good visualization and user interfaces are often used by IT designers and consultants. However, these packages are unable to model several of the key requirements for simulating grids, most notably, resource multi-tasking, task parallelization, resource scheduling and resource provisioning based on dynamic policies.

Acknowledgment

The authors would like to thank Eric Peng Ye, Daniel Dias, and Michael Ignatowski for their contributions to OPERA.

6 References

- [1] C. Amza et al. Specification and Implementation of Dynamic Web Site Benchmarks. In *Proceedings of the 5th IEEE Workshop on Workload Characterization*, 2002.

- [2] S. Bagchi. Simulation of Grid Computing Infrastructure: Challenges and Solutions. In *Proceedings of the 2005 Winter Simulation Conference*, December 2005.
- [3] G. Banga and P. Druschel. Measuring the Capacity of a Web Server. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS '97)*, December 1997.
- [4] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Proceedings of Sigmetrics '98*, June 1998.
- [5] S. Bose. *An Introduction to Queueing Systems*. Kluwer Academic Publishers, 2002.
- [6] R. Buyya and M. Murshed. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience*, 14:1175–1220, 2002.
- [7] D. Cameron et al. Evaluating scheduling and replica optimization strategies in OptorSim. In *Proceedings of the 4th International Workshop on Grid Computing (GRID'03)*, 2003.
- [8] J. Challenger, P. Dantzig, A. Iyengar, M. Squillante, and L. Zhang. Efficiently Serving Dynamic Data at Highly Accessed Web Sites. *ACM/IEEE Transactions on Networking*, pages 233–246, April 2004.
- [9] J. des Rivieres and J. Wiegand. Eclipse: A Platform for Integrating Development Tools. *IBM Systems Journal*, 43(2), 2004.
- [10] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. Grid Services for Distributed System Integration. *IEEE Computer*, pages 37–46, June 2002.
- [11] A. Iyengar, M. Squillante, and L. Zhang. Analysis and Characterization of Large-Scale Web Server Access Patterns and Performance. *World Wide Web*, 2(1,2), June 1999.
- [12] A. Legrand, L. Marchal, and H. Casanova. Scheduling distributed applications: The SimGrid simulation framework. In *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03)*, 2003.
- [13] X. Liu, J. Heo, and L. Sha. Modeling 3-Tiered Web Applications. In *Proceedings of MASCOTS 2005*, 2005.
- [14] D. Menasce and V. Almeida. *Capacity Planning for Web Services: Metrics, Models, and Methods*. Prentice-Hall, Inc., 2002.
- [15] W. Shi, E. Collins, and V. Karamcheti. Modeling object characteristics of dynamic web content. *Journal of Parallel and Distributed Computing*, 63(10), 2003.
- [16] A. Williams, M. Arlitt, C. Williamson, and K. Barker. *Web Content Delivery*, chapter Web Workload Characterization: Ten Years Later, pages 3–21. Springer, 2005.
- [17] L. Zhang, Z. Liu, A. Riabov, M. Schulman, C. Xia, and F. Zhang. A Comprehensive Toolset for Workload Characterization, Performance Modeling and On-line Control. In *Proceedings of Performance Tools 2003*, September 2003.