Colin Garrett (ST Onsite)

From: Colin Garrett (S&T Onsite)
Sent: Wednesday, March 31, 1999 5:49 PM
To: Doug Hiller; David Kubelka
Subject: Suggestions on Benchmarking Paper

Gentlemen --

I've had a closer look at our rough draft, as well as at other MS white papers and various introductory material on benchmarking. We are off to a good start but it seems to me we still have a lot of work to do on organization and content before we are ready to move on to a more detailed level of editing. I think we are missing things that should be included and that the things we have are not organized to best advantage.

I suggest the plan shown below for organizing the paper. This plan has two chief features:

- It's oriented toward selling the customer on our services by highlighting the problems they are facing or may face, and then explaining how we can solve or prevent them.
- It tells a story, starting at the beginning of the benchmarking process and moving through to the successful conclusion

Title (suggestion): "PLS Benchmarking for Software Developers"

- **Introduction** (In the Intro, we'll state the main problems customers have in regard to scaling up their applications, how PLS benchmarking can help them solve them, the customer scenarios where benchmarking is useful, and the other topics to be covered in the paper).
- **Why Benchmarking?** (We'll flesh out discussion of why customers would benefit from benchmarking, the various ways it can benefit them, various customer scenarios that might call for it, the trouble they can get in without it)
- **The Lab Engagement** (Here we lead prospective customer through the benchmarking process at PLS as shown below)
  - **Eligibility** (who is eligible--David's p. 5)
  - **Preparation** (what info the customer has to provide, setting objectives for the lab engagement)
  - **In the Lab** (what happens once we start the engagement, how long it takes, what you do. Discuss our **Environment** (made up of <u>Software</u> [discussion of Loadrunner], <u>Hardware</u>, and <u>Network)</u>; Environment subsection covers material on David's p. 2. In general, talk up the strengths of our environment and how it can benefit customer)
  - **Results** (discuss what sort of data is generated and what you give customer after this is all over)
- **Following Up** (discuss how customer acts on the data we've given him, emphasize that benchmarking is an ongoing process; state that many former customers have found it beneficial to come back for more)
- **Conclusion** (sum it up)

What do you think? If this sounds good then **I suggest David take his first draft and recast into this format, and also that he write new material as necessary to fill up the new sections.** (Don't worry about making it polished, just throw in the info as best you can, with lots of details, and I will fix it up.) To help you along, David, I have prepared templates for the **Introduction** and **Environment** sections. These are attached below and include yellow-highlighted guidelines suggesting what sort of sentences you should write where. I suggest you take these, fill in correct info, and incorporate them into your second draft.

Introduction and
Environment P...

Final Notes:

- Sorry to throw it back at you in so rough a form but I think it would be better if we beef up and reorganize the information, which only you can do.
- If you know of areas that should be covered but which aren't included above, tell me and I will find a place to put them.
- If you think anything I've included above is unnecessary, let me know and I'll cut it.
- These proposed heading names are tentative as regards exact wording. I'll make them sound better before we're

done. Your suggestions encouraged on headings and everything else.

Doug, I hope this is helpful and the direction you want to go. If I missed your drift on what's desired, please let me know and I'll try some alternate approaches… I'll be in touch Friday.

Colin Garrett
Technical Editor / Enterprise Customer Supportability Center

Prodbro.pdf
(get adobe reader)

David's
rough

Microsoft

PLS Computing Laboratory

Service Description

title - benchmarking

The Premier Leveraged Services Computing Laboratory

White Paper

Abstract

The PLS Lab Services are part of the offerings that Premier Leveraged Services provides. This white paper describes the status quo of these services. Lab services at the moment provide a benchmarking environment for software developers. These benchmarks should test and ensure scalability of large-scale applications and should help software developers to successfully deploy Microsoft technologies in their applications. Our focus is on three tier applications with the database services provided by Microsoft SQL Server and the business services provided by the Internet Information Server (IIS).

what are the 3 tiers

which is this about?

Lab or Benchmark

rewrite to reflect benchmarking goal?

document aims to inform customers of, interest them in lab services?

CONTENTS

help team
- Set bmk objectives
- dont try [illegible] to test, prove

BENCHMARK OBJECTIVE

The benchmark should (1) verify design decisions made in the development process or better yet could (2) help to make such decisions. Measurements provide a means for a comparative reference. Benchmarks performed should be measurable and repeatable. This can and should make performance testing an integral part of the development cycle and quality assurance. Meaning the lab engagement for benchmarking is a repetitive event that occurs at different points in the development cycle. *

Benchmarking is the tool to improve product quality and usability substantially. Good performance is a requirement for user acceptance.

Another purpose for benchmarks is to establish baselines for potential customers. Benchmarks can serve as reference installations to document new performance boundaries. We encourage reviews, audits by independent consultants.

Exactly ~~[illegible]~~ who are ~~[illegible]~~ these reviews done on?

rewrite
—
what does bmk do?
- scenarios

- this is why you should use the facility

- necessary step - can't live w/o bmking

allocating adjustments
- network-wide

- code works
is it performing adequately?

- cost in hardware + software

- all the time: problems w/ installations, SR escalations

- we provide env they don't have

- scalability

- impact on corporate services

- retaining customers
└ those who get benchmarked come back

- new functionality

- verify product enhancements

- performance testing as important as functionality testing.

List of the kinds of simulation

LAB OFFERINGS

topics
- simulating (virtual) users
- Loadrunner

diff kinds of simulation

Virtual users are simulated by a "load driver" software. Currently we are using a product by Mercury Interactive – Loadrunner. Virtual users simulate real users on different levels of a multi-tier application. The most complete simulation is to drive the graphical user interface of the client application. This kind of simulation has a very limiting restriction insofar as there will by only one virtual user per agent machine. The agent machine acts in behave of a client workstation and runs the load driver software. Our current configuration allows for 100 agent machines, which means that there will be only 100 virtual users to put load on the business and data services. If the application has the presentation layer and the business services on the client tier, provisions should be established, so that business services can be driven independently from the GUI (COM).

Agent machines can run virtual users that connect to IIS servers via HTTP. We can provide an agent configuration and a Loadrunner license for **7000** web users.

For virtual users that connect directly to SQL server either through DB-Lib or ODBC we can provide a configuration for up to **10000** users.

As "**S**ystem **U**nder **T**est", SUT we have multi-processor machines with up to 4 Pentium II Xeon 400 MHz CPU's with raid-controllers and storage-arrays that make large database configurations possible . We will try to keep our server offering current and are looking forward to 8-Way systems with the new Pentium III Xeon which will support up to 32GB main memory. Benchmarking will enable our customer's products to take advantage of the scalability features of Microsoft's SQL Server 7 together with Windows 2000 on such high-end systems.

tiers of business functions

Today most request are for database workloads, nevertheless we do encourage engagements that include stressing the business functions, the middle tier. This middle tier today can be driven with HTTP requests, in the future we hope to be able to provide the same functionality for DCOM as soon as Loadrunner supports this kind of scenario.

We plan on upgrading our capability on simulating users according to user demand. For Web scenarios we will soon be in the 30,000 to 50,000-user range.

Our networking environment in the lab consists of a 100Mbit switched network that will soon be extended with a 1GBit switched network with an ATM backbone.

sh'd list examples first then stated objectives ?

EXAMPLES

cost perf

Cost and price-performance comparison is not a stated objective of benchmarks that we help to perform; nevertheless can such a benchmark be used to establish a hardware baseline for a well-known configuration.

For example, a defined workload, known to be representative for real world installations can be scripted for the use with Loadrunner. Running this workload can then reveal figures such as, growth of database per day/month/year. What is the impact on the response times of a daily full database backup, what is the impact on performance of permanent backup of the database transaction log to either hard disk or tape device? How many tape devices do I need, to finish the full database backup in the time windows of low system load?

Database physical design decisions can be verified, example: by adding a specific index one query statement will improve significantly, but will the insert transactions in a multi-user environment stay within my performance constraints after adding this index?

RECOMMENDED BEST PRACTISES

Lab engagements are usually conducted during two weeks. We will assist with formulating the objectives for the benchmark and creating a project plan for the duration of the engagement. A detailed project plan will allow us to phase in specialists out of product support or if necessary product development, for the technologies used at the right point in time. It will also allow us to help keep the benchmark on track.

We will help our customers to get around the "lab syndrome" which is: Lab results are unrealistic and can not be duplicated in a production environment." To circumvent this problem it is necessary to picture the real life workload into a Loadrunner scenario. Consisting of the scripts that were produced by recording business transactions, and the distribution of these scripts to agent machines with the proper definition of frequency, think time and iteration.

We will review and analyze performance logs from scenario runs. Help isolate bottlenecks and propose improvements to the design or implementation of the application.

ELIGABILITY

Premier level customers can request lab engagements. Requests should be directed to the technical account manager (TAM) or the application development consultant (ADC) who will help to fill in a lab request to asses the benchmark requirements. Requests should be submitted at least 30 days in advance. The process can be seen on http://www.microsoft.com/permier/I_don't_know where

Microsoft® Premier leveraged services ®

Microsoft

Benchmarking for Software Developers

Service Offering

A Problem-Solution and -Prevention Service of the Premier Computing Lab

White Paper

Abstract

The omission of load and performance testing from the development cycle frequently leads to serious problems when the attempt is made to scale the application up to a larger user community. To assist developers in preventing and solving scalability problems, Premier Leveraged Services offers Benchmarking for Software Developers, a service in which customers bring their application to a lab engagement at a Premier Computing Lab, where special software is used to simulate loads of up to several thousand users and Microsoft staff work with the customer to identify and eliminate performance bottlenecks. The paper describes scenarios in which benchmarking can benefit the customer, key benchmarking concepts, how a customer sets up and prepares for a lab engagement, which members of the customer's team should attend, the three phases of a lab engagement, the unique software, hardware, and network resources needed for benchmarking, the Premier Computing Lab's ample resources in these areas, and the Premier Computing Lab's consulting profile. The paper also discusses following up on a lab engagement, the usefulness of return visits, and the possibility of having a benchmark certified by an independent auditor for marketing purposes.

CONTENTS

INTRODUCTION

Scalability has been recognized as the Achilles' heel of even the most talented software developers. Developers frequently create functionally rich applications that work well for a relatively small number of users but which, when scaled up to a larger user community, reveal unexpected problems that negatively impact system performance and customer satisfaction.

To assist developers in solving and preventing scalability problems, Premier Leveraged Services offers Benchmarking for Software Developers, a service in which members of the Premier Computing Lab conduct lab engagements with developers using special benchmarking software to simulate loads on their applications of many thousands of users. These loads typically reveal performance problems—ranging from flaws in the application's underlying architecture to relatively minor matters—which can then be isolated, diagnosed, and corrected through the joint efforts of Microsoft staff and the customer. The goal of these efforts is to improve the robustness, capacity, marketability, and value of the customer's application.

Effective benchmarking is a complex activity requiring unique skills and specialized hardware and software. The Premier Computing Lab offers both to developers interested in improving the scalability of their applications, including access to Microsoft technical experts who are uniquely positioned to help customers get the most out of their Microsoft software.

This white paper outlines scenarios in which an application under development might benefit from benchmarking, touches on the concepts behind benchmarking, describes the Premier Computing Lab engagement, and discusses the unique resources of the Premier Computing Lab, including hardware, software, and network resources as well as our consulting profile. It also includes suggestions for following up on the lab engagement.

WHY BENCHMARKING?

While functional regression testing is an accepted part of the development cycle, load and performance testing generally are not. Even when developers do consider load issues in the development cycle, the effectiveness of their work in this area tends to be mixed because multi-user performance is difficult to predict without the use of specialized simulation software, and the testing of multi-user performance requires a set of skills that is normally not present in the developer community.

The absence of careful benchmarking is a frequent cause of problems with installations and scale-ups—as well as of customer dissatisfaction and developer embarrassment. Benchmarking can help developers prevent or solve many of the problems associated with implementing and upscaling their applications.

Benchmarking Scenarios

Benchmarking has a potential role to play in many parts of the application-development cycle as well as during and after installation. Listed below are scenarios in which a developer might find it useful to arrange for a benchmarking engagement with the Premier Computing Lab:

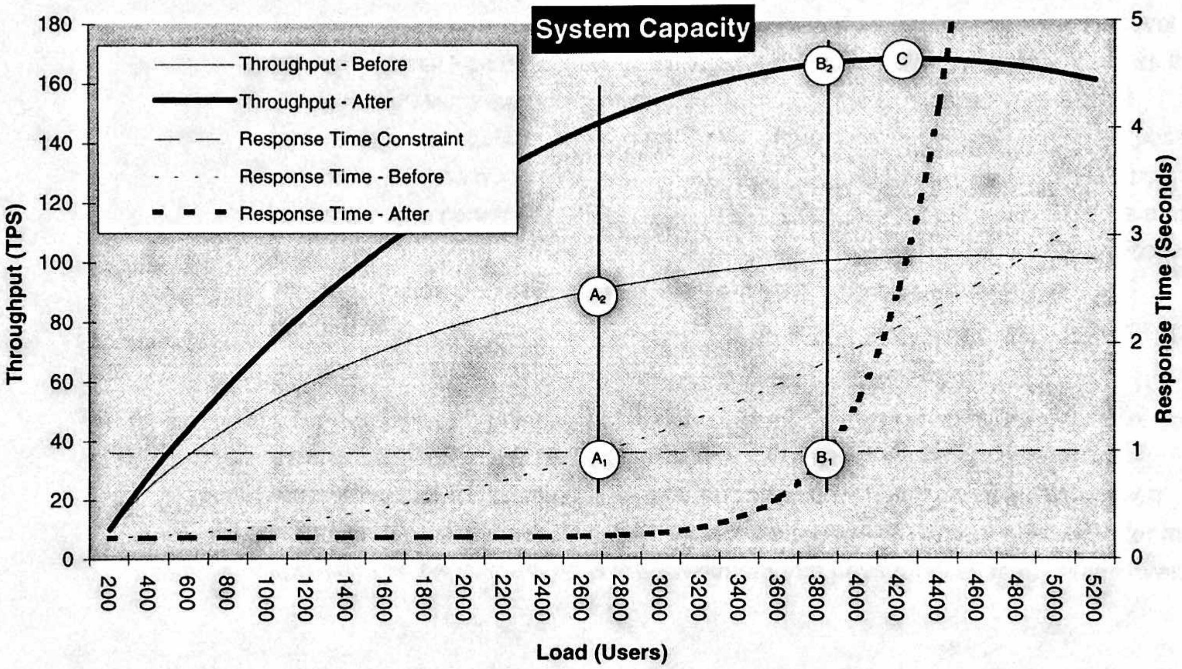
- **A point in the development cycle has been reached where performance needs to be verified.** For example, in a database application, the logical and physical design is complete and stored procedures have been written to access the database, but the application's behavior in a multi-user environment remains unknown.
- **A new implementation of an underlying technology needs to be validated against the application's goals.** For example, you have upgraded from Microsoft SQL Server 6.5 to SQL Server 7.0 and need to test your application against the new technology.
- **The application architecture has been redesigned and needs to be verified.** For example, the client has evolved toward a thin client, and business functions have been migrated from the client applications into a Web server.
- **You need to identify and eliminate bottlenecks.** Requirements for an existing application have changed. Installations for a certain number of users worked according to user expectations, but when more users were added, performance becomes unacceptable.
- **The need for a new performance baseline arises.** For example, potential customer interest exists for a 10,000-user installation, but the biggest reference installation is only a couple of thousand users. Solution: Perform a 10,000-user benchmark to show scalability and document configuration requirements. Have the benchmark audited by an independent third party.
- **A support incident is proving resistant to the traditional problem-solving methods.** For example, a computer running SQL Server becomes unresponsive if multiple long-running transactions are executed. The Premier Computing Lab can model an environment (including hardware and software components) to isolate the problem and help you move forward toward a solution.

Benchmarking Concepts

Benchmarking operates on the theory that computer applications are inherently affected by certain performance constraints, but that within those constraints it is possible to adjust certain variables to optimize performance and even stretch the outer limits of the possible. Doing this depends on accurately observing the original underlying performance patterns and understanding the relationships among the variables that affect performance so that you can manipulate them to positive effect. Before we continue, it might be helpful to define a few key benchmarking terms:

- **Throughput** is the amount of work the system performs in a given unit of time, usually transactions per second (TPS).
- **Workload** is a measure of system capacity, measured here in number of users.
- **Response time** is the amount of time the computer takes to perform a given operation.
- **Response time constraint** is the maximum acceptable amount of time for completing an operation, usually determined by the developer's marketing department.
- **Knee capacity** is the point at which the throughput stops increasing linearly (the resulting curve on the graph being knee-shaped) and response times go up exponentially.

The figure below illustrates the performance of a system before and after benchmarking, and shows how, after benchmarking, the shortening of code paths and elimination of contention problems have produced substantial increases in system efficiency. A discussion follows.



Reading from the left and bottom of the graph, and looking at the solid curved lines, the figure shows the **effect that an increase in the number of users (load) has on throughput**. Typically, as the number of people using a system increases from zero, throughput increases linearly, rising as the number of users rises. After a point, however, the increasing number of users causes contention problems, and throughput begins to level off (this is the knee capacity); if the number of users increases sufficiently, throughput can even decline. As shown in the graph, **system throughput is greater after benchmarking than before**. This improvement was attained by using benchmarking to identify performance bottlenecks and then shortening the code path and reducing contention problems to eliminate them. Additional boosts in efficiency can be obtained by tuning the underlying system layers (for example, the Web server, database server, or operating system).

Reading from the right and bottom of the graph, and looking at the curved dashed lines, the figure shows the **relationship between the number of users and the system response time in seconds**. As the number of users increases, response time also tends to increase. As the graph shows, this is true both before and after benchmarking. Indeed, in one regard response time before benchmarking is faster than response time after benchmarking: when there are 4,200 or more users on the system. However, taking into account the response time constraint, set at one second in the illustration and represented by the dashed horizontal line, **performance after benchmarking is seen to be clearly superior where it counts**: In the "before" system, the system can support only 2,600 users before the response time crosses the one-second mark (Point A_1), disallowing the addition of more users. But in the "after" system, 3,800 users can join the system before the constraint is reached (Point B_1).

Normally, maximum throughput is not as important as usable capacity, the level of throughput that can be achieved while adhering to certain constraints, such as the one-second response time constraint depicted in the illustration. The goal of benchmarking is to increase the efficiency of the system by getting the best possible ratio between the maximum throughput (Point C) and throughput as limited by the response time constraint (Points A_2 and B_2). Efficiency for the system before and after benchmarking can be calculated by dividing A_2 by C and B_2 by C, respectively. The result shows an efficiency of ~50 percent before benchmarking and ~95 percent after, illustrating that it is possible to almost double system efficiency through benchmarking.

It is perhaps worth pointing out that system efficiency is ultimately much more than a matter of graphs and percentages. The system has a cost per user to the enterprise, and if it can be made to support twice as many users within a given response time constraint, the cost is cut in half. Results like this show why, for most applications, benchmarking is an investment that pays for itself many times over.

THE LAB ENGAGEMENT

The heart of the PLS Benchmarking for Software Developers service is the lab engagement, in which a team from the development company travels to a Microsoft Premier Computing Lab for an intensive session of performance-bug–busting in close cooperation with Microsoft staff. These sessions differ in focus depending on where the application is in the development cycle; however, all have the same basic goal: Trying to optimize the application's performance for a given hardware configuration. Each engagement is individually scoped and the time required varies with the application and objectives. Most last about two weeks; almost all are hard-driving "workouts" featuring long days and on-duty weekends.

The course of a typical Premier Computing Lab benchmarking engagement is described below.

Setting Up an Engagement

Premier customers set up lab engagements through their Application Development Consultants (ADCs) or Technical Accountant Managers (TAMs), who will act as their hosts when they're at Microsoft. Engagements should be arranged approximately 60 days in advance. The long lead time is not time lost for the average Premier customer, who has plenty to do to get ready for the engagement.

Preparation

During the time after an engagement is scheduled, but before the customer's team arrives at the laboratory, numerous steps must be taken by both the customer and PLS. These steps usually include some or all of the following:

- **The ADC or TAM, working with the customer, completes and submits a Lab Request Form and a Project Plan.** These address such issues as how many computers will be needed for the engagement, what tests will be run, and what specialists will be needed and when.
- **The ADC or TAM and the customer, typically with input from the customer's marketing or sales departments, define the objectives for the lab engagement**, including what kinds of benchmarks to include, the desired transaction mix, the desired number of users, the number of transactions to be performed per some given unit of time, and the expected system configuration (size of database server, number of application servers, and so on).
- **The customer, advised by their ADC or TAM, if necessary, arranges for the use of appropriate load-driver software**. PLS has a business arrangement with Mercury Interactive for making its Loadrunner load-driver software available to customers once a year (with certain restrictions) for Premier Computing Lab engagements, but a variety of suitable benchmarking products are on the market.
- **The customer writes approximately 10 to 20 "business transactions,"** or scripts, reproducing the steps a user would typically take (including think time) in using the application to perform the tasks specified by the marketing department. For example, for an airline reservation and ticketing application,

the scripts might include various scenarios in which a ticketing agent books and sells a ticket, some for a smoking seat, some for a nonsmoking seat, some involving a special meal, and so on.

The Customer's Team

Who at the customer company should be on the team that comes to the Premier Computing Lab for the lab engagement? This varies with the company, the application, and the goals of the engagement. The amount of time each person needs to spend at the lab usually varies as well, with some team members necessary the entire time and others for much shorter periods.

To allow the members of the customer's team to make their travel plans, the project plan shows who is needed at the lab and when. Following is a list of the personnel typically required at a Premier lab engagement, with a description of the responsibilities of each:

- The **writer of the transaction scenarios** or scripts. This person's presence is required the whole time because these scenarios almost always need adjustment during benchmarking. This person's role typically includes incorporating requests from marketing, making sure that the load is generated in a realistic way, collecting the results of the test runs, and explaining the difference in runs to the people involved.
- The **database administrator** creates different database configurations and studies the effects caused by various modifications, for example, how response time changes for an online backup.
- The **quality assurance team** configures and deploys the application, provides tools, and learns what additional testing has to be incorporated in their quality assurance procedures.
- Someone from the **marketing staff** might be necessary to set the goals; for example, specifying what transaction mix to use in a 10,000-user benchmark—how many from sales, how many from marketing, and so on.
- In smaller companies, the **application developer** him or herself might come to the lab to be available to incorporate changes into the application in response to bottlenecks or other problems found in multi-user environments

In the Lab

When the appointed day arrives, the customer's team travels to the lab city, checks into a hotel, is greeted by their ADC or TAM, and begins their lab engagement. Most engagements are divided into three phases:

- **Phase 1:** Installing the customer's application on the lab computers and restoring the database, if necessary.
- **Phase 2:** Confirming that all of the scripts work in a multi-user environment or adjusting them until they do.
- **Phase 3:** Finding and fixing bottlenecks. This might involve: doing test runs;

studying the results; isolating problems; and bringing in specialists from Microsoft Product Support or Microsoft Developer Support to help find workarounds or identify aspects of the application that need to be adjusted in order to meet performance goals.

As previously stated, testing for functionality (regression testing) is a standard part of the development cycle, but load and performance testing generally are not. The Premier Computing Lab operates under the assumption that below-expectations performance is a bug. A chief focus of all Premier Computing Lab benchmarking engagements, but especially those involving applications being subjected to benchmarking for the first time, is the identification and repair of performance bugs. One key to success in these tasks, as described in the next section, is having the optimal benchmarking lab environment.

THE LAB ENVIRONMENT

Effective load testing requires unique software, hardware, and network resources, as well as the contributions of specially trained personnel. Together these resources comprise the lab environment. The Microsoft Premier Computing Lab is equipped with a wealth of benchmarking resources, which are being constantly updated and expanded to keep at the leading edge of technological developments.

A well-equipped benchmarking lab is probably beyond the reach of the majority of software-development companies. Load-driver software sells at premium prices, and load testing requires the use of servers powerful and numerous enough to handle a load equivalent to that of a major corporation—something prohibitively expensive for most development firms, especially given the typical need pattern for these resources in the development cycle (intensively but for brief periods). The Premier Computing Lab, by using its lab resources fulltime in benchmarking for Premier customers, is able to distribute the expense, making the process cost-effective for all concerned.

Discussed below are the software, hardware, and networking resources, as well as the human assets, that the Premier Computing Lab can bring to bear on benchmarking the applications of Premier customers.

Note: All numbers describing the capacity of our systems and offerings, such as the number of virtual users supported, can be expected to shift upward dramatically moving forward. The pace of change in this field is fast. Our lab refresh cycle is between six and nine months, and we try to stay within six months of the technology currently on the market.

Software

Benchmarking depends on the use of load-driver software to simulate corporate-sized loads on the application under test. Customers have wide latitude in choosing which load-driver to use in their lab engagement. The “house” load-driver of the Premier Computing Lab is currently Mercury Interactive’s effective Loadrunner program. The following paragraphs describe how Loadrunner works, but the information given will be applicable generally to other benchmarking load generators as well.

Loadrunner works by enabling a benchmark specialist to simulate a large number of users, called virtual users, on a single computer, known as an agent machine. In tests of an application's database layer, for example, the current version of Loadrunner can simulate 1,000 users per agent machine; thus, using 10 agent machines, it is possible to simulate a load on the DB of 10,000 users.

Loadrunner also permits testing on all three layers of an application: the database layer (at the bottom), the business functions layer (the middle tier), and the presentation layer (the top tier, and also known as the GUI layer). Premier Computing Lab staff will make recommendations concerning what layer(s) to test, based on the application architecture and the goals of the lab engagement. Among the considerations taken into account in making these recommendations are the following:

- Testing the database layer is the simplest, most common, and most effective kind of testing. Because all problems in the database layer replicate to the other layers, this is considered the best place to begin in benchmarking a new application. Note that it is possible to start database benchmarking even before the application has been written.
- Testing the presentation layer is the most complete form of testing (as it also tests the two lower layers); however, it is highly inefficient because it imposes a limit of one virtual user per agent machine.
- Sometimes a blend of methods is used. For example, if on a given test run you are simulating 10,000 users in a test of the database layer, it can be informative to set up a GUI-layer test on one agent machine to get a feel for the performance experienced by the "10,001st" user.
- Currently, in testing business functions, Loadrunner is limited to the Web server interface (HTTP requests); however, Mercury Interactive plans to introduce Component Object Model (COM) capability to Loadrunner in the near future, simplifying benchmarking for applications in which the business services are based on a distributed COM (DCOM) server.
- If the application locates the presentation and business services layers both on the client tier, provisions should be made to allow the business services to be driven independently of the GUI (COM).

The Premier Computing Lab can currently provide an agent configuration and Loadrunner license for 7,000 Web users.

For virtual users that connect directly to SQL Server, either through DB-Lib or ODBC, we can currently provide a configuration for up to 10,000 users.

As stated previously, these numbers can be expected to increase dramatically moving forward.

Hardware

State of the art hardware, and lots of it, is essential for benchmarking your applications effectively and ensuring they are compatible with the latest high-end systems that your customers might be running. To see how your application performs under a load equivalent to what it would face if used by a major corporation, it is necessary to have the server and memory resources equivalent to what would be deployed at such an organization.

The Premier Computing Lab has the hardware resources necessary to handle large, corporate-sized database configurations. So that customers of all sizes can be assured of having access to a facility that meets their needs, the Premier Computing Lab offers four types of lab configurations:

- Large-scale lab for simulating between 2,000 and 7,500 users
- Medium-scale lab for simulating between 500 and 2,000 users

- Mini-lab for simulating between 50 and 500 users
- "Physical" lab, for running one virtual user per agent machine; 100 agent machines are available

Reflecting our commitment to stay abreast of the latest in hardware developments, we will be acquiring computers with Intel's new 8-way technology, with the 8 Pentium III Xeon processor, as soon as they become available. These computers will support up to 32 gigabytes of main memory. Benchmarking your applications on such high-end systems will ensure that they can take advantage of the scalability features of Microsoft SQL Server 7.0 and Windows 2000—and that they remain attractive to potential customers as business upgrades to these more powerful systems.

Note: We have an active hardware acquisitions program, but we do not make purchases geared toward specific customers or engagements. When customers arrange for lab engagements, the hardware available to them will be what we have "on the shelf" at the time of their visit.

Network

The chief concern regarding networking in a benchmarking lab is that the network be fast enough so that it never be the cause of any bottlenecks. Any bottlenecks encountered should originate in the application, so that they can be isolated and fixed and the application improved. To achieve this objective, the Premier Computing Lab supports LAN speeds of 100 megabits per second (Mbps) to agents and workstations, and of 1 gigabit per second (Gbps) to servers and lab interconnections.

An exception to the above rule regarding network bottlenecks is in the case of Web-based applications where the developer wants to see how their product functions in the environment of the Internet, where network bottlenecks are common. To meet this need, the Premier Computing Lab plans to develop the means of simulating the Internet environment, creating bottlenecks such as those experienced by users accessing the application through ISDN, ASDL, and other WAN connections.

Our Consulting Profile

In a benchmarking lab, even the best software, hardware, and networking resources are only as good as the people deploying them. Performance and load testing are specialized skills demanding special training and experience, and the skillset needed to fix bottlenecks, once identified, potentially spans every aspect of the underlying technology. The Premier Computing Lab, in its consulting profile, offers customers both breadth and depth. Among the Microsoft staff available to aid lab visitors are:

- **Lab Specialists,** including Lab Technicians, experts in the process known as "buildup and teardown" (loading the customer's application and database onto the lab computers at the beginning of the lab engagement and removing them at the end) and Performance Specialists, experts in the performance aspects of many technologies.

- **Relationship Managers,** including the customer's TAM and/or ADC.
- **Product Specialists,** including experts in the various Microsoft product areas who can be called in as needed to suggest approaches to fixing specific, identified bottlenecks, with an escalation path to Microsoft product development.

FOLLOWING UP

The Premier Computing Lab, as previously stated, regards the lab engagement as a kind of intensive bug-busting session targeting performance bugs. As a result of this focus, the principal take-home benefit the software developer derives from the engagement is to have identified and fixed a number of performance problems and to have pinpointed areas where further work needs to be done once the developer returns to their home office. The larger benefit, of course, is to improve the application, increasing sales and customer satisfaction.

Obviously, "improving the application" is not something that happens once in the development cycle, but is rather a recurring concern as the product evolves. For this reason, the Premier Computing Lab encourages return visits focused on making the product more and more competitive, and in fact the lab's experience has been that most developers who have come to the lab once eagerly arrange for subsequent visits. As an illustration of how lab engagements after the first might be useful, many developers devote their first visit to testing their application's database services and their second to testing its business functions.

Auditing

Another potential use of a follow-up lab visits is for third-party auditing, in which the customer arranges for a third-party auditor to attend the lab engagement. This usually happens in a follow-up visit because the initial visit is devoted to making sure the product does in fact perform as desired. The auditor can certify the various performance benchmarks (number of users, response time, and so forth), and the development company can then use the successful audit in marketing their application.

CONCLUSION

This paper described PLS Benchmarking for Software Developers, a service designed to help developers solve and prevent scalability problems in their software applications. The service, as described in the paper, centers on "lab engagements" at a Premier Computing Lab, where special load-driver software is used to simulate loads of up to several thousand users on the customer's application, allowing Microsoft staff and the customer's team to identify and eliminate performance bottlenecks.

The paper described the benefits of benchmarking, including scenarios in which benchmarking can benefit the customer and some key benchmarking concepts. It discussed the Premier Computing Lab lab engagement, including how to set up and prepare for an engagement, which members of the customer's team should attend, and the three phases of the lab engagement. The lab environment was also discussed, with an emphasis on the considerable software, hardware, and network resources available at the Premier Computing Lab, as well as its strong consulting profile and the unique position enjoyed by Microsoft personnel in assisting customers in getting the most out of Microsoft technologies. Finally, the paper addressed the issue of following up on the customer lab engagement, including the benefits of returning for subsequent visits and the option of having a benchmark certified by an independent auditor as an aid in marketing.

Premier Leveraged Services and the Premier Computing Lab are dedicated to helping our customers be successful in using Microsoft technologies so they can be self-sufficient in doing so. To partner with us in improving the scalability of your applications, and to take advantage of the benchmarking service and resources described in this white paper, you are cordially invited to contact your ADC or TAM at your convenience to arrange for a lab engagement at the Premier Computing Lab.