Performance Testing
Why and How?

Sudheer Parpatakam

Agenda

- Why Performance Testing?
- Performance Testing Workflow
- HP LoadRunner Solutions

Why Load Test Your Application?

Prevents costly failures of mission-critical applications.

Assures performance and functionality under real-world conditions.

Locates potential problems before your customers do.

Reduces development time.

Reduces infrastructure costs.

Types of Performance Testing

Component Testing	Find the behavior and performance of each tier.
Load Testing	Determine whether the system handles anticipated real-world load.
Stress Testing	Find system's breaking point; measure whether system's environment is properly configured for unexpected, high-transaction volume.
Volume Testing	Check stability of system under extended periods of load.

Examples of Performance Test Objectives

Application response time

- How long does it take to complete a task?

Configuration sizing

- Which configuration provides the best performance level?

Regression

- Does a new version of the software adversely affect response time?

Reliability

- How stable is the system under a heavy work load?

Capacity planning

- At what point does performance degradation occur?

Bottleneck identification

- What is the cause of the performance degrading?

Performance Testing Expert Workflow

- Establish Goals
- Gather System Usage Information
- Analyze System Under Test

Defining Goals

Why start with conceptual goals?

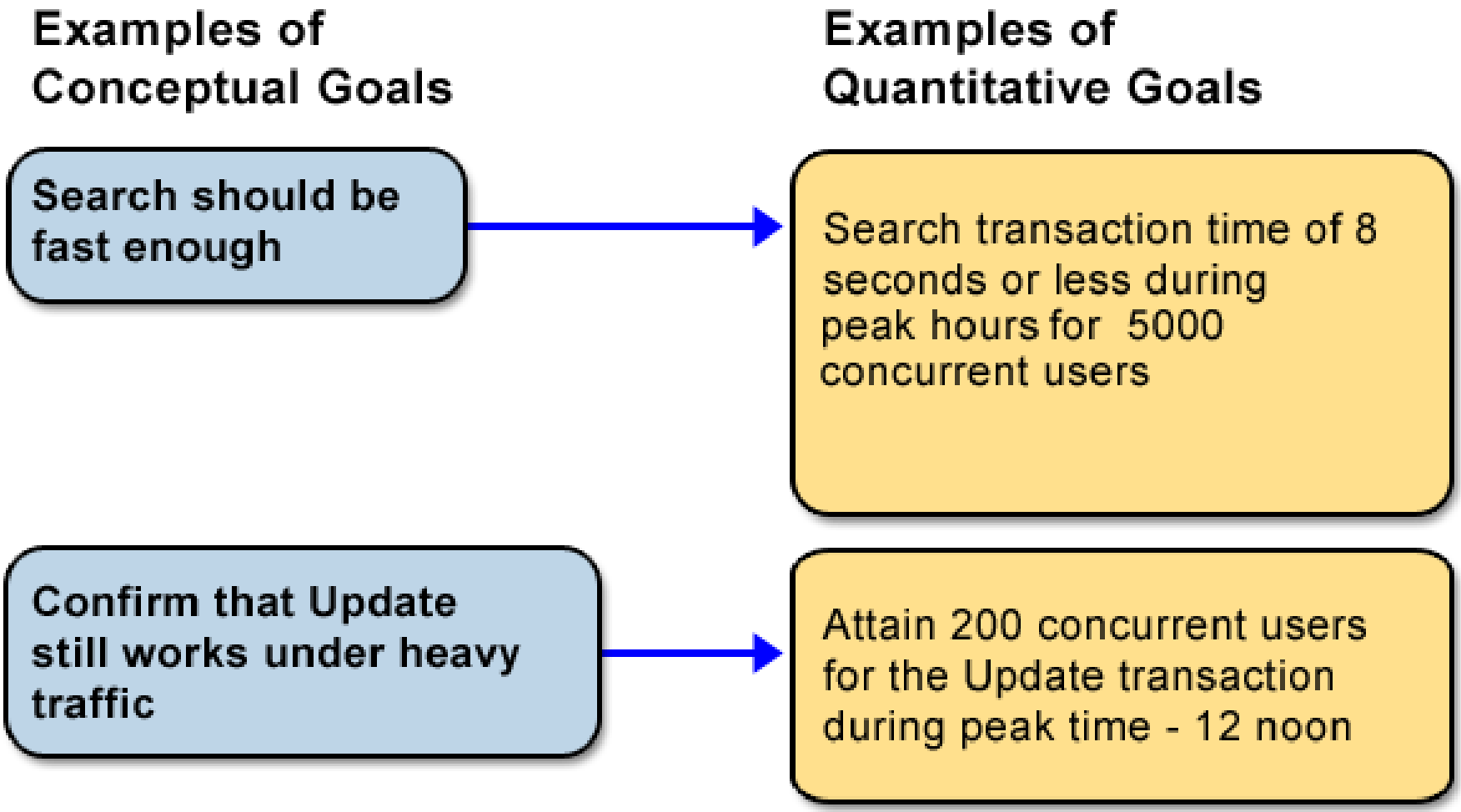
Conceptual goals should outline all of your load test objectives.

Examples of conceptual goals:

- A high-priority example is the responsiveness of a “Search” function. Are we able to get search results within a reasonable time?
- A second example is the system administrator’s concern that the “Update” transaction functions during heaving usage.

In the initial stages, jotting down goals that can’t be measured will allow later filtering to create more focused goals.

Quantifying Load Testing Goals



Gathering System Usage Information

Why *gather system usage information?*

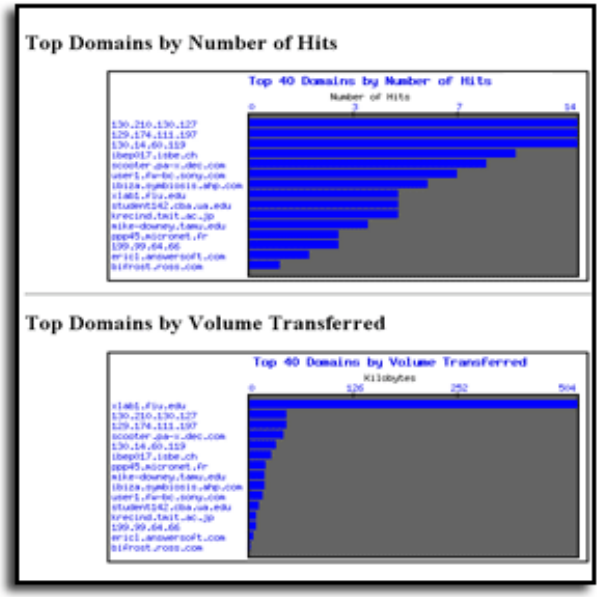
Allows you to:

- Decide which business processes to test.
- Isolate peak loads and peak load times.
- Document user actions and valid input data for each business process.

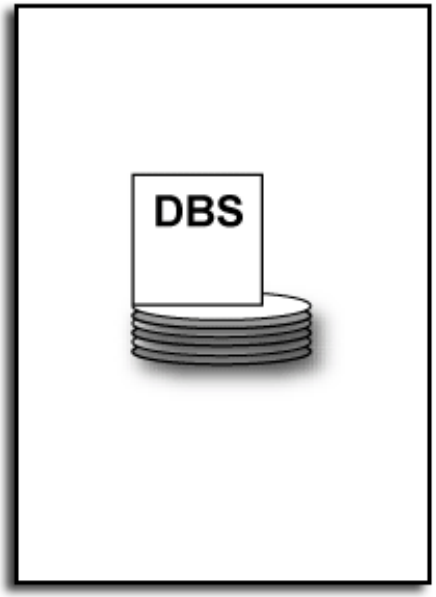
Where *can we get information?*

- Using the site, first-hand
- Consulting with administrators
- Consulting with executives
- Researching competitor's sites

Obtaining Customer Locations



Use web site's domain report log



Consult database of customer addresses

Create estimated model (if application is not yet live)

Analyzing the System Under Test

Why *gather system usage information?*

- Allows you to setup monitors.
- Allows you to effectively coordinate with backend experts.
- Provides system information that helps isolate performance problems.

Where *can we get information?*

- System Administrators
- Backend experts
- Application experts
- Database Administrators

Identifying Business Processes to Test

Mission-Critical Business Process

- Business Processes that are crucial in day-to-day operations.

Heavy Throughput

- Heavy throughput business processes may not be mission-critical but are very popular.

Dynamic Content

- Dynamic content consists of sever requests that are customized for each user.

Business Processes to Test: Mission-Critical Transactions

- **Mission-critical**

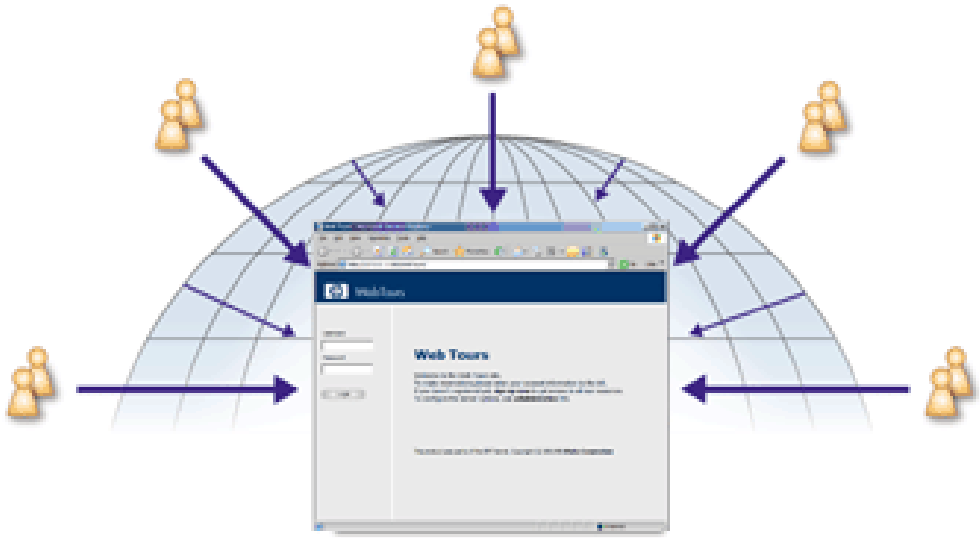
Examples:

E-commerce transactions
Downloading orders

Get info from:

Application Experts
Management Team

Business Processes to Test: Heavy Throughput



- Mission-critical
- **Heavy throughput**

Examples:

Home page viewing
Super Bowl ads
Accessing a login screen

Business Processes to Test: Dynamic Content

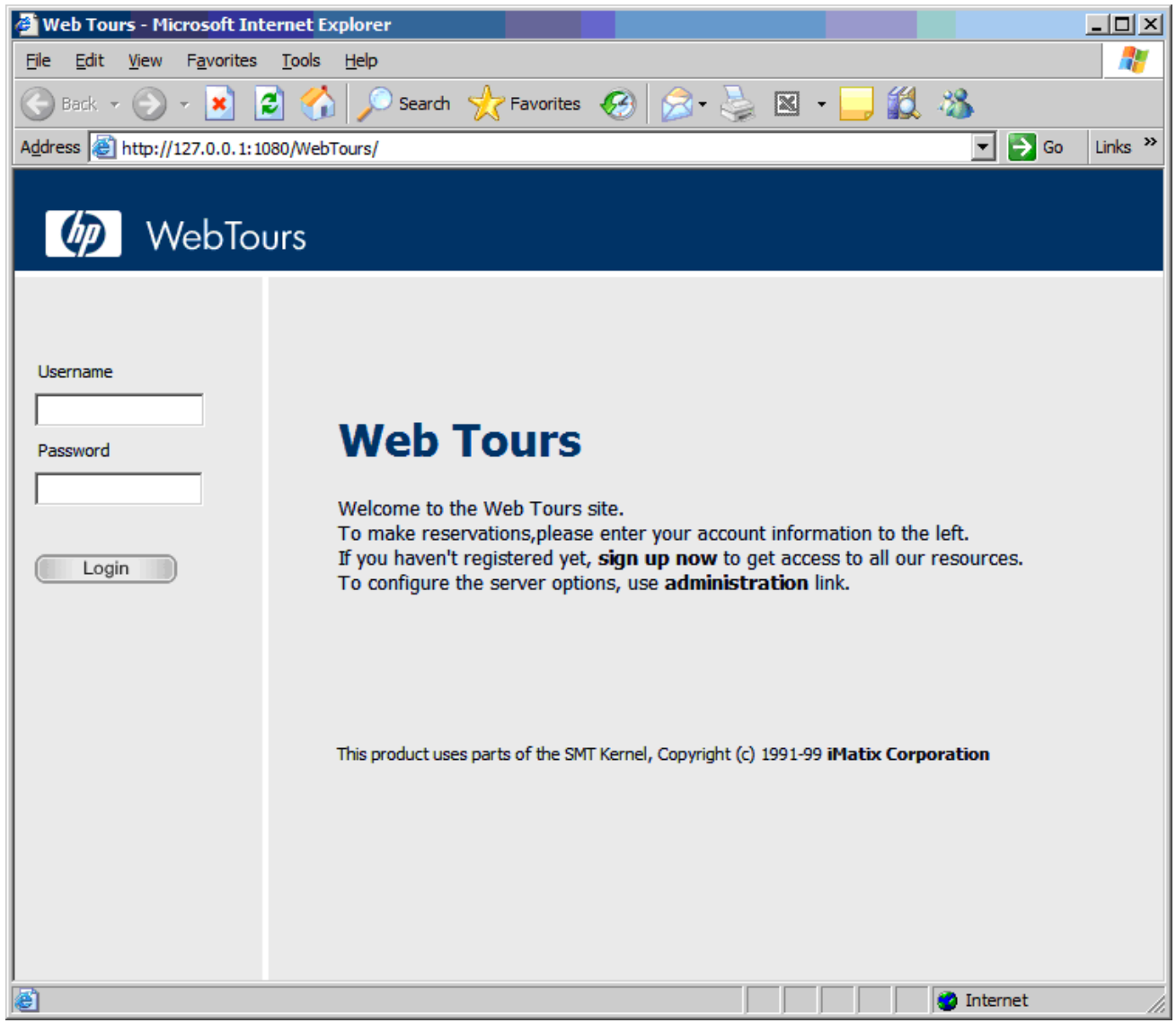
- Misson-critical
- Heavy throughput
- Dynamic content

Examples:
Dynamic page generation
Streaming media -
hear streaming audio
view streaming video

Business Process Profile

Business Process Name	Typical Day	Peak day	Dynamic content	Mission critical	Test
Sign in	70/hr	210/hr	Light	High	?
Create new account	10/hr	15/hr	Moderate	Low	?
Search for flights	130/hr	180/hr	Moderate	Moderate	?
View flight booking	20/hr	30/hr	Moderate	High	?
Purchase ticket	40/hr	90/hr	Heavy	High	?

Documenting User Steps and Input Data



Determining Valid Test Data to Use

Sign-in Business Process

User Actions	Test Data
Type user name and password	username: pattyr password: test
	username: paulb password: rain
	username: ericc password: brazil
	username: joee password: jojo
	username: franzo password: fober

Valid Test Data Sources

Valid input data comes from three sources:

Master Data

- Also known as Application Data
- Data is resident in the application's database
  - Examples: ID numbers and passwords

User-Generated Data

- Originates with the user
  - Examples: new unique ID or email address

External Data

- Data is unknown before the application is run
  - Examples: confirmation and purchase order numbers

Defining Concurrency

Concurrency

A set of users acting upon an application in a similar manner at the same time.

Application Level

- How many users are active on the system?

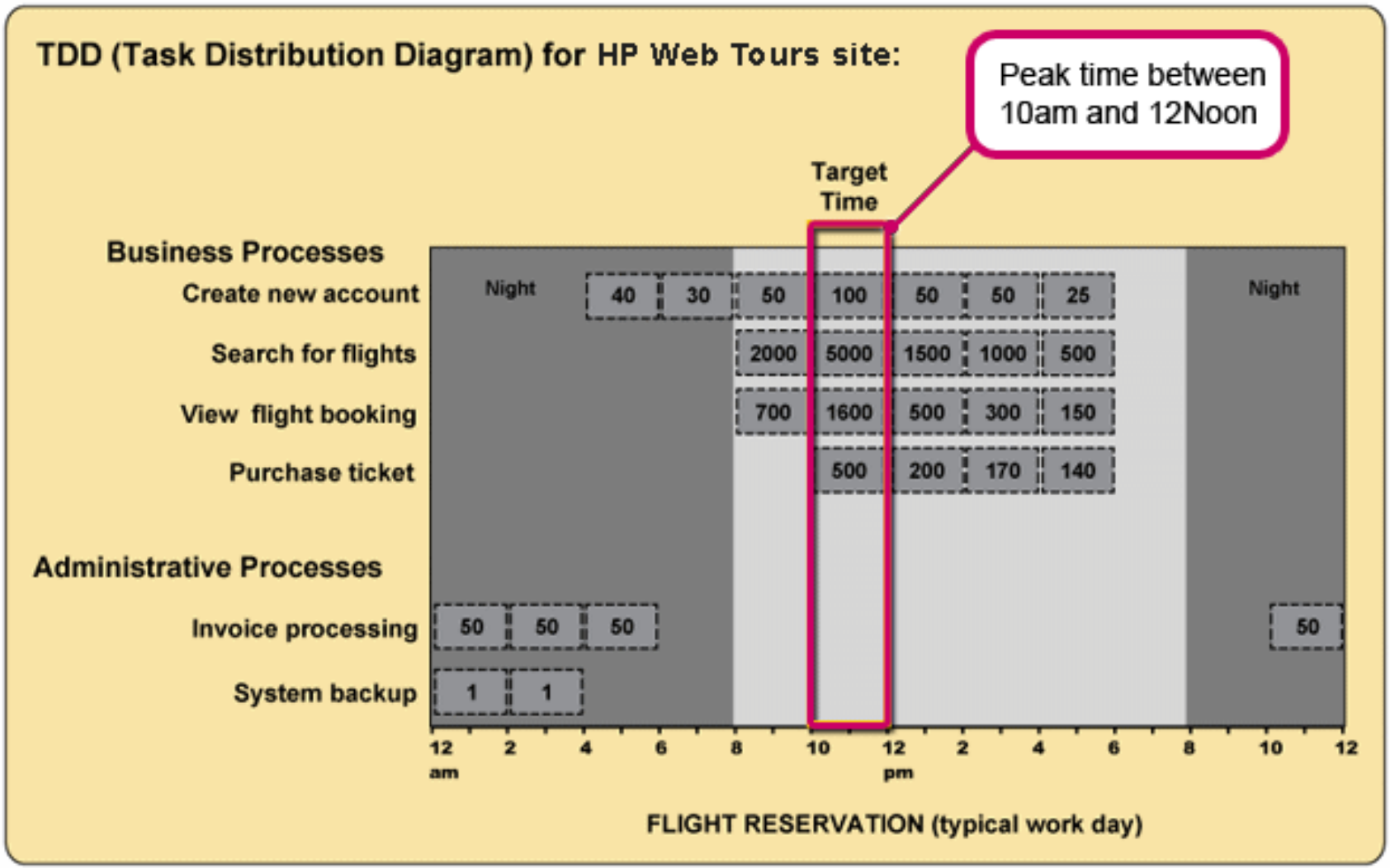
Business Process Level

- How many users are buying tickets?

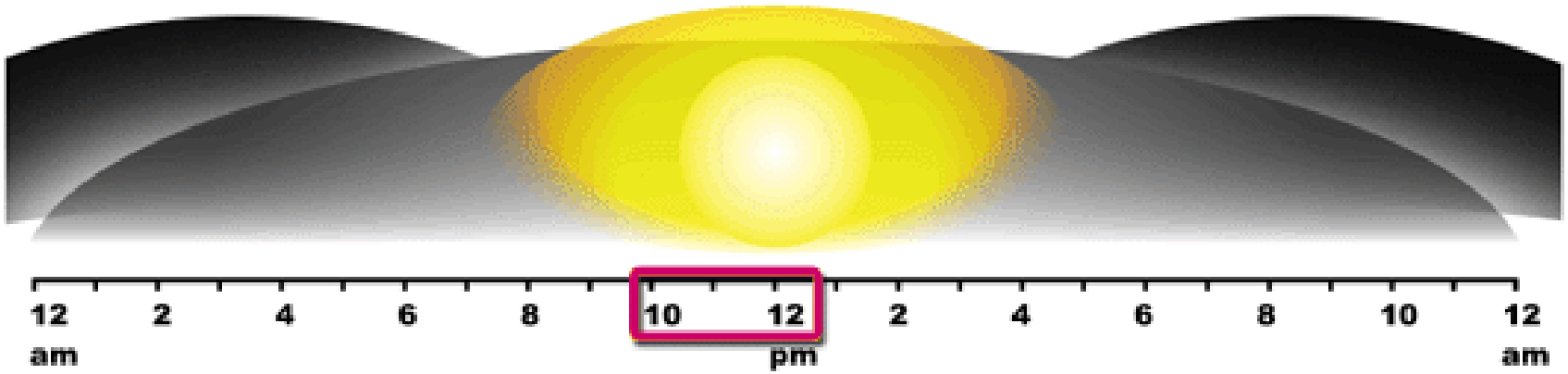
Transaction Level

- How many users are buying tickets NOW?

Application Concurrency



Business Process Concurrency



Business Processes	Number of Users (between 10am and 12 noon)
Create new account	100
Search for flights	5000
View flight book	1600
Purchase ticket	500

User concurrency changes based on the time of day.

Transaction Concurrency

How many transactions will need to be run per minute if a load test has to be run for two hours with 5000 users, assuming an average transaction length of five minutes?

Determine how many transactions run per minute:

- 120 min / 5 min = 24 iterations for each user
- 5000 users X 24 iterations = 120,000 transactions
- 120,000 transactions / 120 minutes = 1000 transactions per minute

Apply the transactional concurrency to the application:

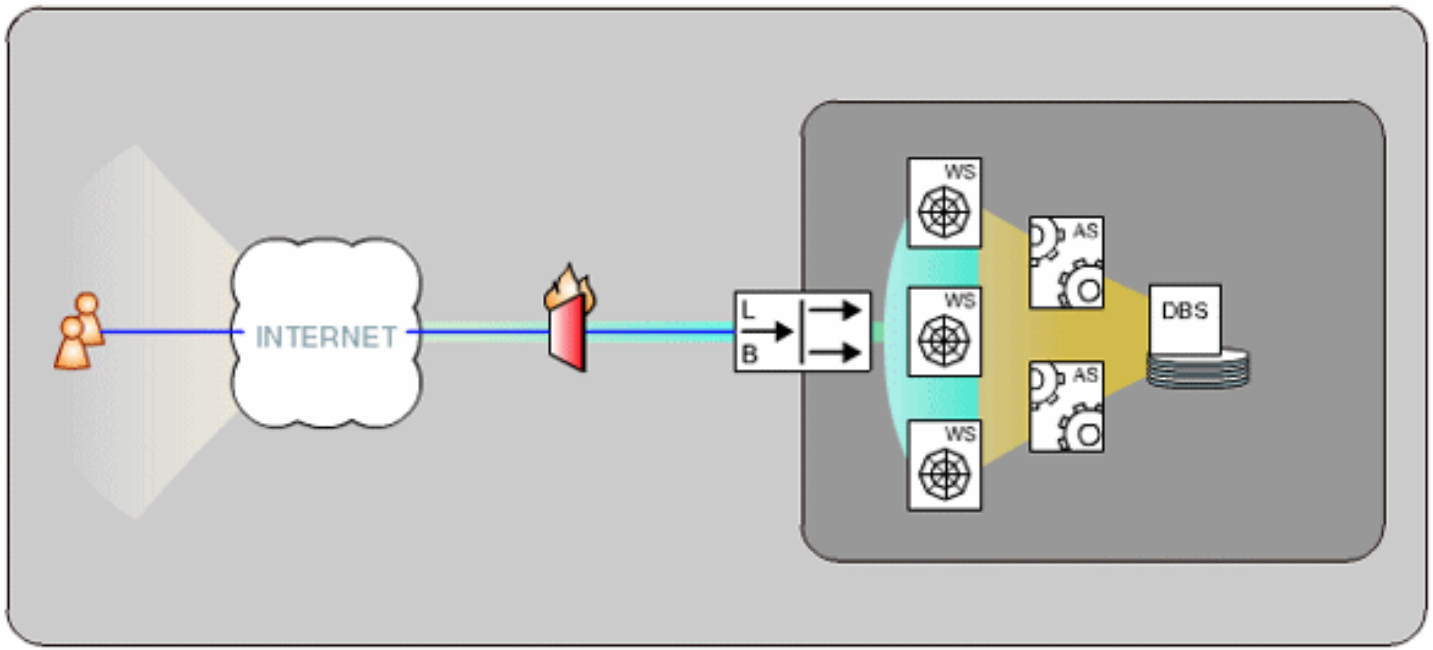
- The test is run during the 10-12 AM time slot
- The test should consist of 5000 users running 24 iterations
- The system must be able to handle 1000 transactions per minute

Understanding the System Components

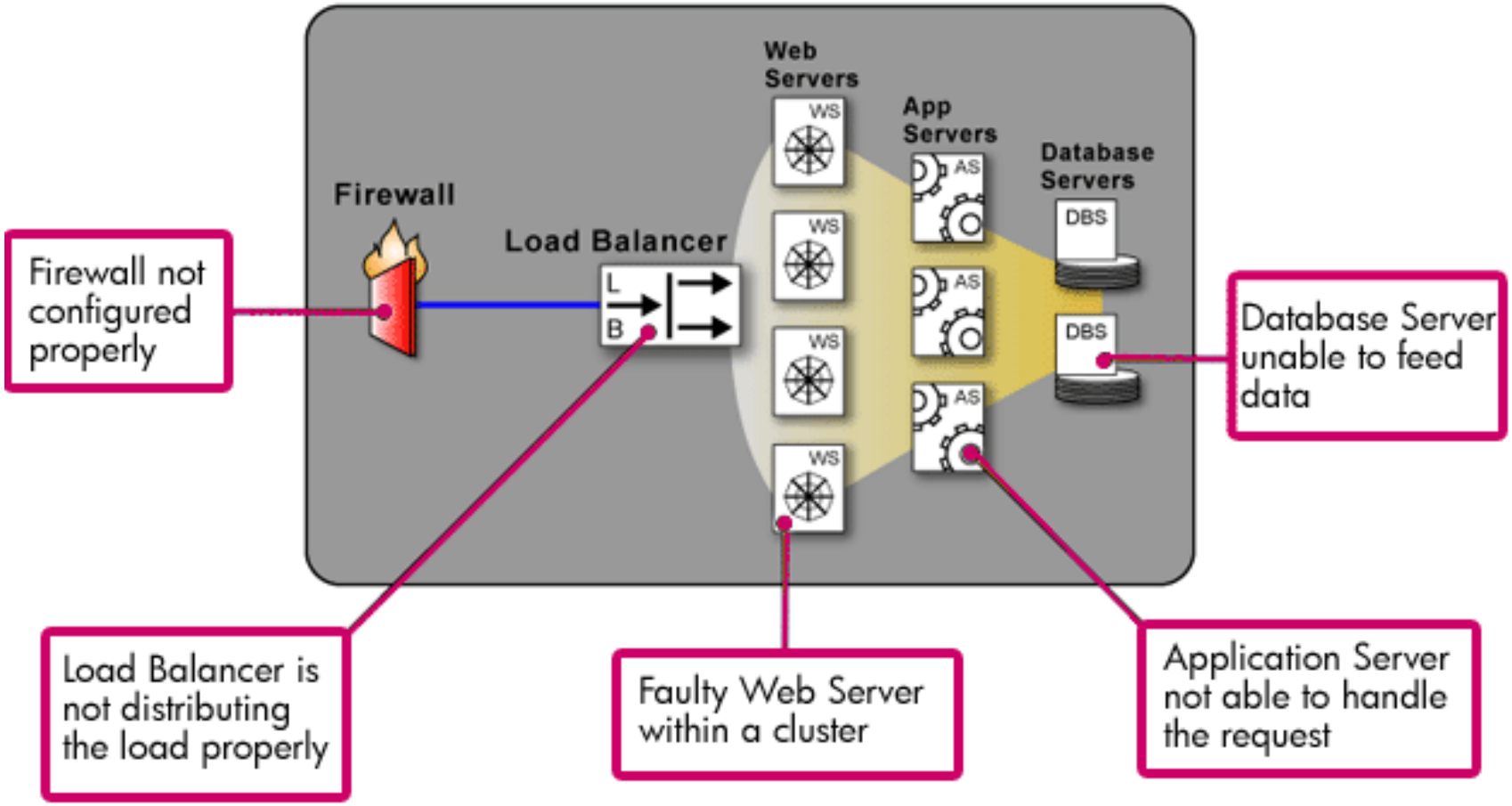
Resources	Type	OS
Web Server	☑ Microsoft IIS ☐ Apache ☐ Netscape	Windows2003
Application Server	☑ Broadvision ☐ SilverStream ☐ Microsoft ASP ☐ Allaire ColdFusion ☐ IBM WebSphere ☐ ATG Dynamo ☐ Ariba Buyer ☐ BEA WebLogic	Windows2003
Database	☑ Microsoft SQL Server ☐ Oracle	Windows2003
Middleware	☐ BEA Tuxedo	Windows2003
Firewall	☐ CheckPoint Firewall-1	Windows2003
SNMP Devices	☑ SNMP Monitors ☐ Cisco Works	Windows2003
Streaming Media	☑ Real Server ☐ Windows Media Server	Windows2003

Mapping Business Processes to Infrastructure Components

Business Process	Web Servers	App Servers	Database Servers
Search	X	X	X
Home Page	X		



Monitoring Application Components



About the Test Environment

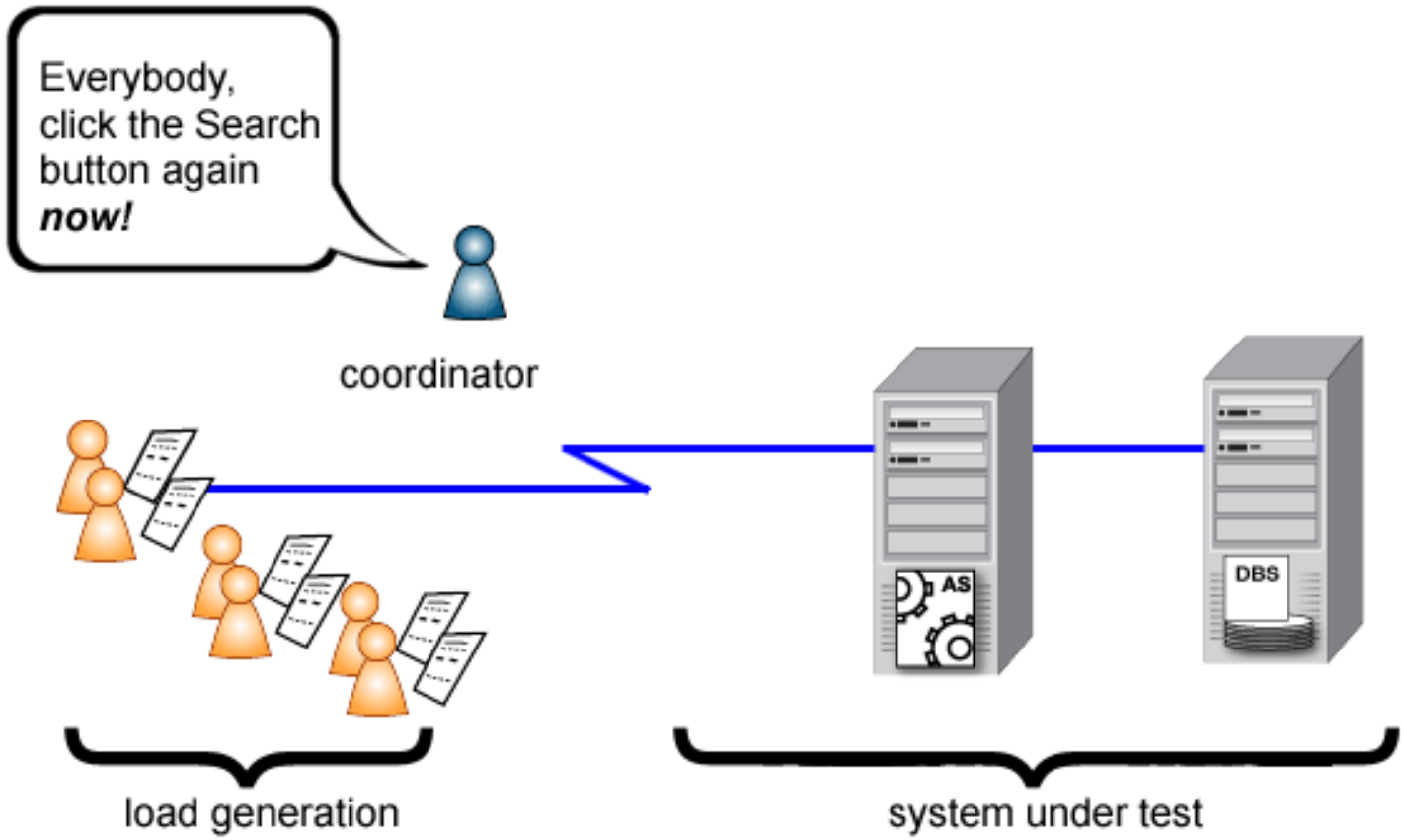
A test environment:

- Allows data to be written, read, and destroyed without affecting production users.
- Allows test system to be rebooted during test runs without affecting production users.

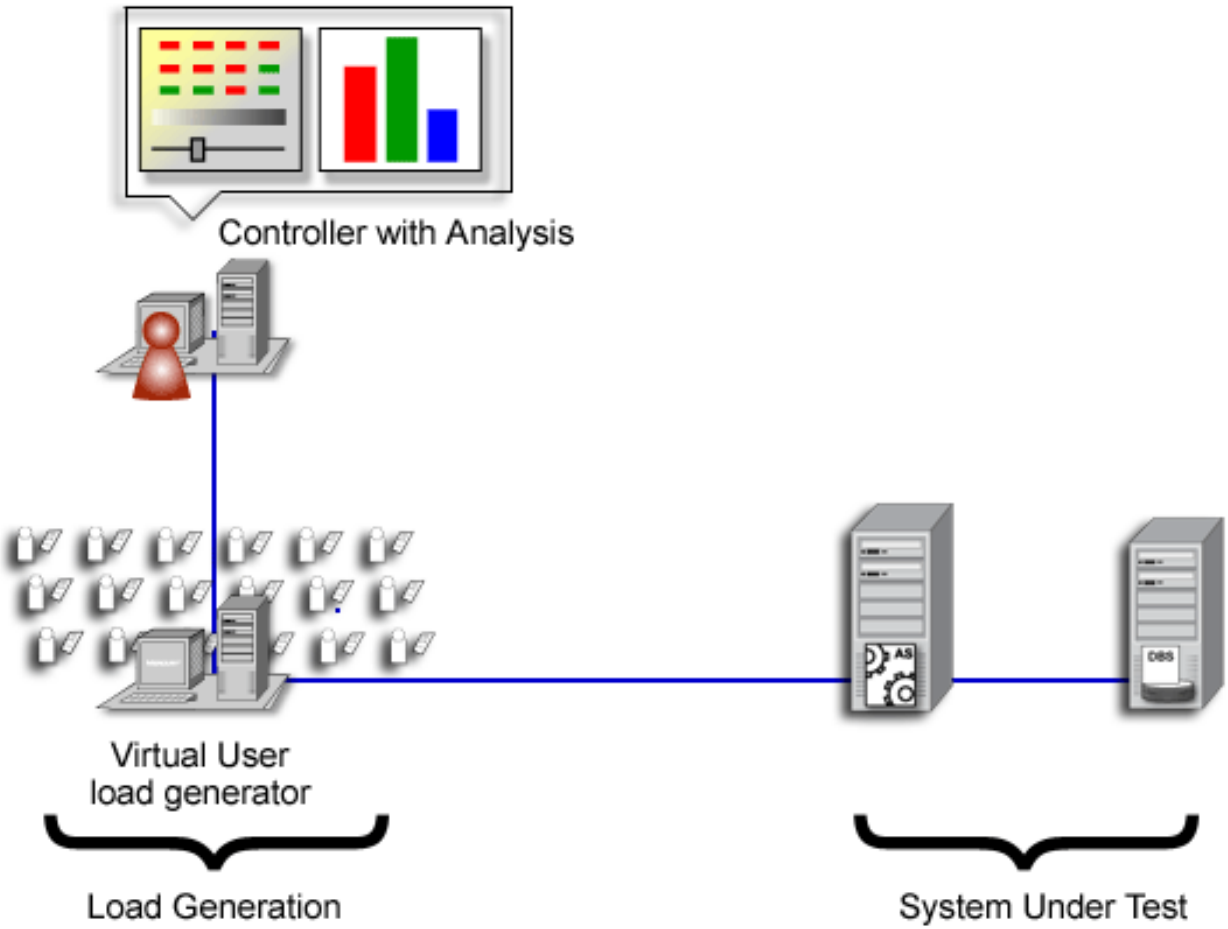
The test environment:

- Should mirror the production system.
- Needs business processes that are functioning correctly.
- Should include Benchmark runs.
- Must contain sufficient hardware to generate the load test.

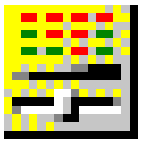
Manual Testing is Problematic



The LoadRunner Solution



LoadRunner's Load Testing Components



LR
CONTROLLER

The Controller is an administrative center for creating, maintaining, and executing **Scenarios**. The Controller assigns Vusers and load generators to Scenarios, starts and stops load tests, and performs other administrative tasks.

LR
LOAD GENERATORS

Load generators (also known as hosts) are used to run the Vusers that generate load on the application under test.

LR
ANALYSIS

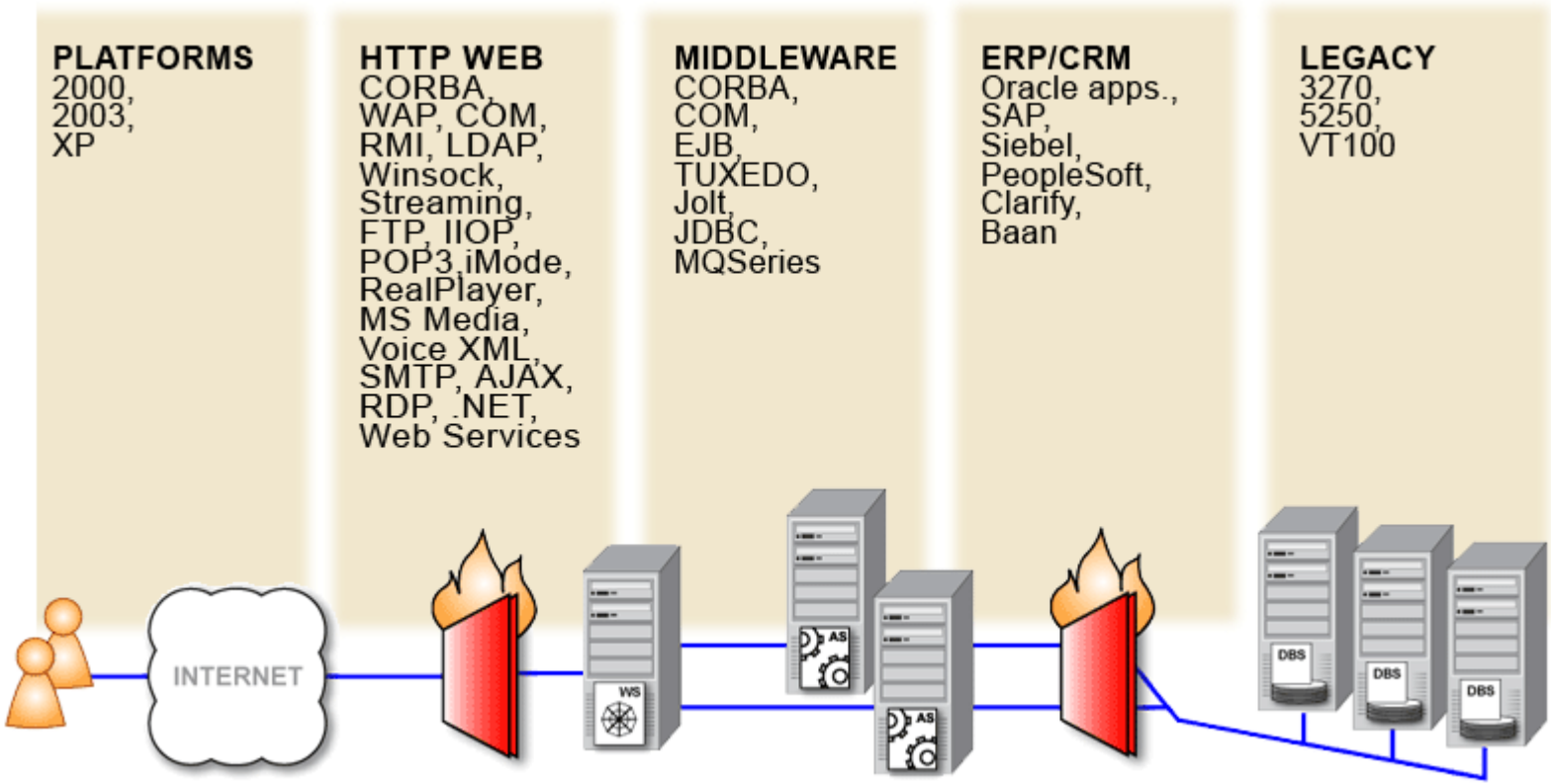
LR Analysis uses the load test results to create **graphs** and **reports** that are used to correlate system information, identify bottlenecks, and performance issues.

VuGen

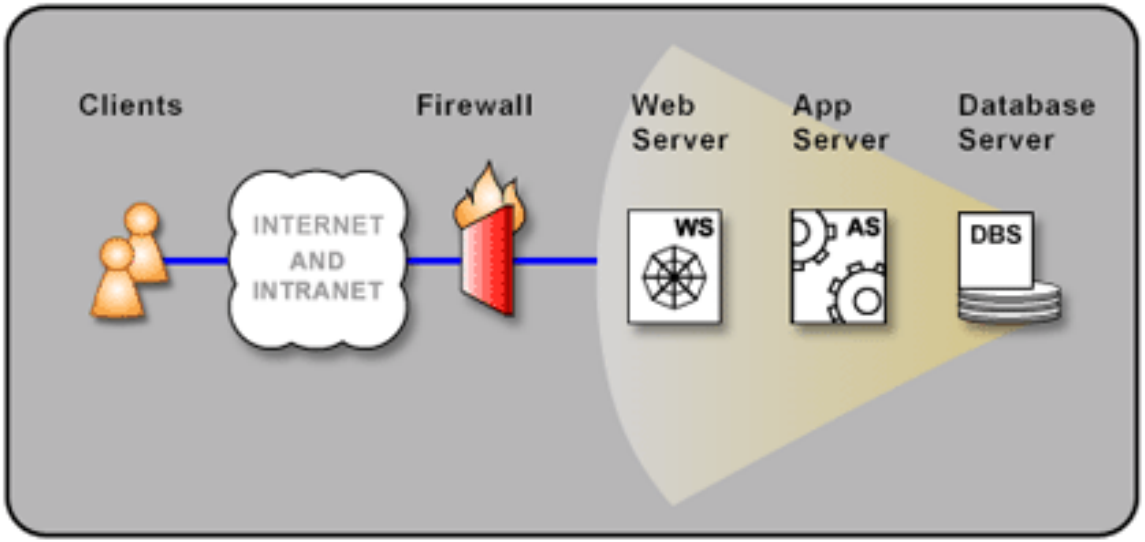
VuGen

Virtual User Generator (VuGen) – records Vuser scripts that emulate the steps of real users using the application.

Supported Protocols



Supported Performance Monitors



Sys. Resource Monitors
Windows,
UNIX, SNMP,
Antara Flame Thrower,
SiteScope

Network Monitors
Network Delay

Firewall Monitors
Check Point

Web Server Monitors
Apache,
iPlanet,
MS IIS

Database Monitors
DB2, SQL,
Oracle
Sybase

App. Server Monitors
Ariba, BroadVision,
ColdFusion,
WebLogic,
WebSphere,
Oralce9iAS,
SilverStream,
iPlanet (NAS)
Fujitsu INTERSTAGE,
MS Active Server Pages

ERP / CRM Server Monitors
SAP Portal,
SAPGUI,
Siebel Web,
PeopleSoft (Tuxedo)

App. Component Monitors
COM+

Benchmarking Run

To validate that there is enough test hardware available in the test environment, benchmark the business processes against the testing hardware.

Take a business process and execute a small number of users against the application.

- Validates the functionality of the business process
- Potentially exposes unknown data dependencies

Evaluate the testing infrastructure against the footprint.

- Do I have enough hardware to generate the user load?
- Do I have enough memory?
- Do I have enough CPUs?

Summary

- Performance Testing Objectives
- Define Goals, Gather Data & Analyze System
- HP LoadRunner Solution

Questions?


invent