IBM Software Group

Performance Problem Determination

Stacy Joines

STSM, ISSW Enablement Team

RTP, NC

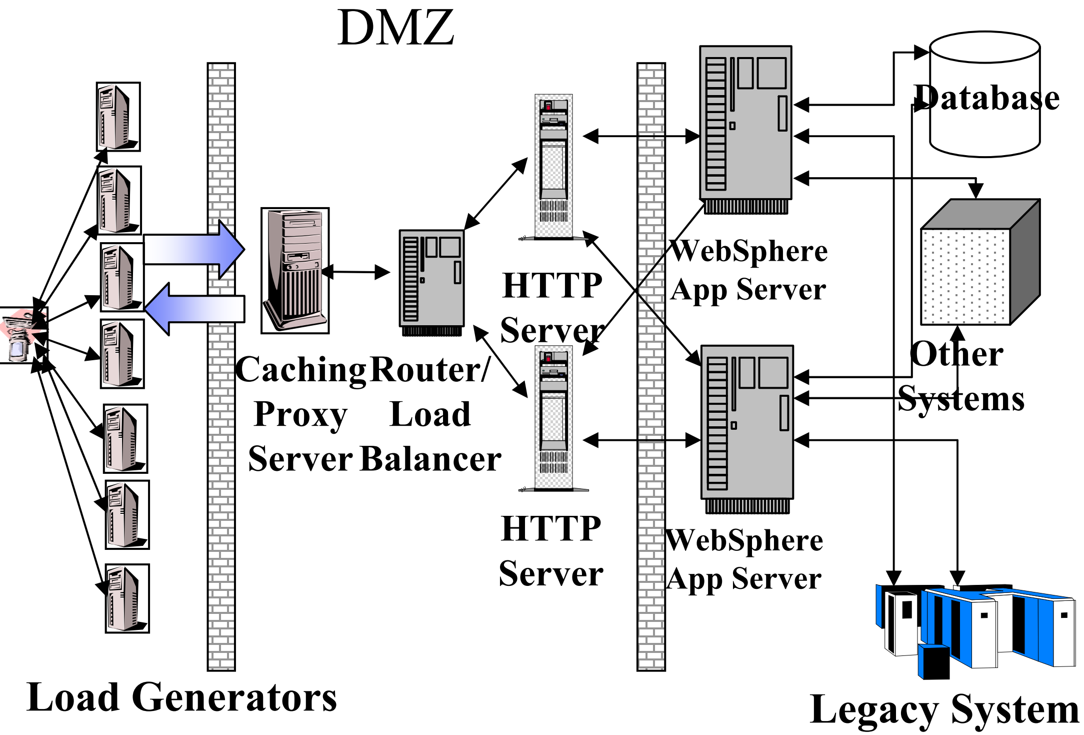
© 2005 IBM Corporation

WebSphere Performance
Bottleneck Patterns and Tuning

- Many Web site performance issues present similar symptoms
  - Underutilization
  - Bursty behavior
  - Full CPU utilization
  - Others…
- Key symptom patterns let us
  - Look for a known set of issues matching this pattern
  - Employ the correct tests and supporting tools to isolate the cause
  - Review problem data in context
- WebSphere Application Server performance
  - WAS "problems" often result from other Web site issues
  - We routinely use WAS to isolate non-WAS issues

Web Site Performance Issues

Using WebSphere to Isolate Problems

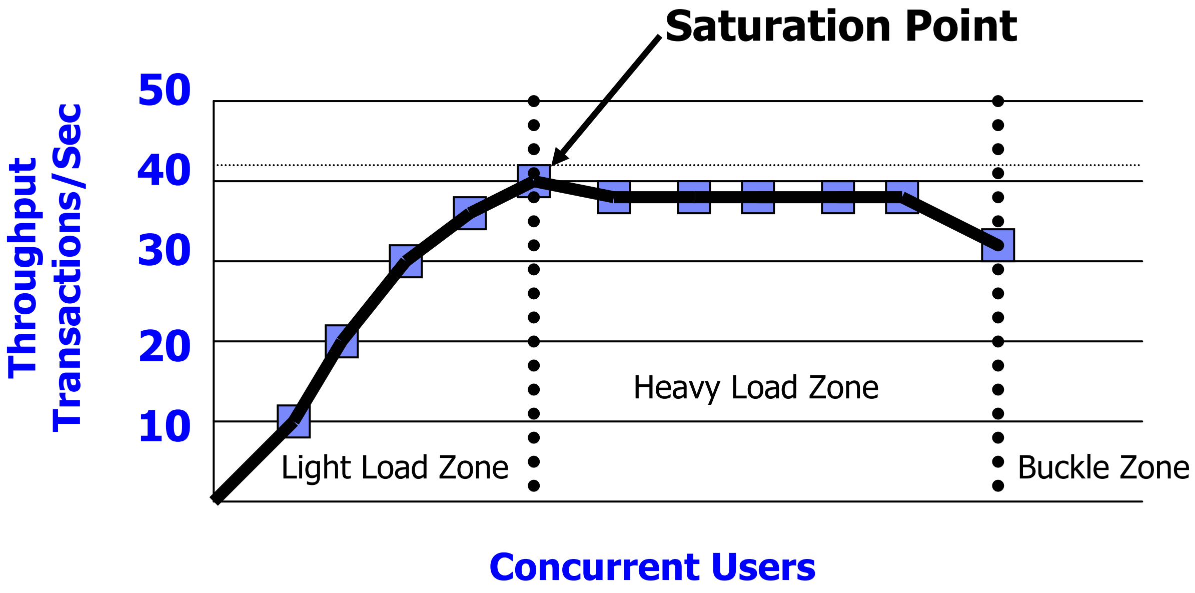


- Three major Web site tuning areas:
  - Web applications
  - Web site hardware
  - Remote systems
- WebSphere in the Web site middle
  - Executes the Web application
  - Receives from the “edge”
  - Interacts with resources at
    - Mid-tier
    - Backend
- Use WebSphere to isolate problems

Throughput Saturation

- At the Saturation Point
  - Additional load does **not** yield additional throughput

- Maximum Throughput is a saturation point
  - 100% CPU utilization (or less for very large systems)



Web Site Bottleneck Patterns

Collecting Data

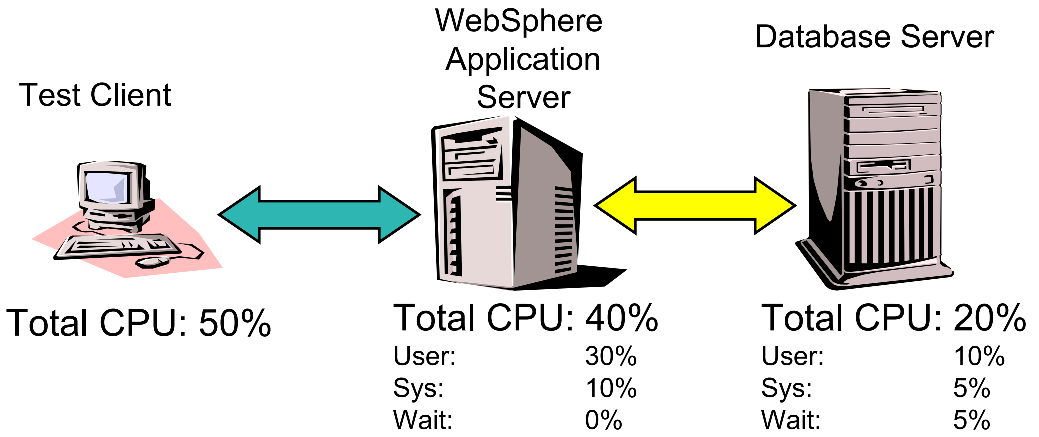
- How do we know there's a problem?
    - Throughput data
    - Response time data
    - Loading data
- What is the Web site doing?
    - CPU utilization data
    - Resource Analyzer information
    - The more data, the better….
- Use this data to launch further analysis

Bottleneck Pattern

Underutilization

Test Run #1

- X Users
- 7 txns/sec
- 5 sec response time



Test Client

Total CPU: 50%

WebSphere Application Server

Total CPU: 40%

User:	30%
Sys:	10%
Wait:	0%

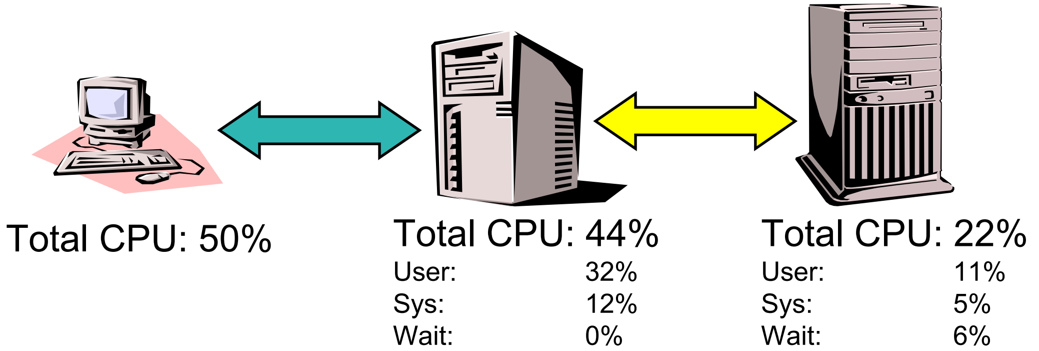
Database Server

Total CPU: 20%

User:	10%
Sys:	5%
Wait:	5%

Test Run #2

- 2X Users
- 7 txns/sec
- 10 sec response time



Total CPU: 50%

Total CPU: 44%

User:	32%
Sys:	12%
Wait:	0%

Total CPU: 22%

User:	11%
Sys:	5%
Wait:	6%

Underutilization

Symptoms and Considerations

- **Symptoms:**
  - **Increasing users does not increase machine utilization**
  - **Transaction rates stabilize, but response time increases**
- **The usual suspects:**
  - **Limited resources**
    - **Undersized resource, improper configuration, etc.**
  - **Application problems**
    - **Synchronization**
  - **No traffic**
    - **Test client not producing sufficient load**
    - **“Edge” equipment problems**
  - **SMP scalability**
    - **Cannot fully utilize a large system with one JVM**
- **Use WebSphere to begin problem isolation process**
  - **Find out what the Web application is doing (thread dump)**
  - **Monitor key resources such as heap and database connections (TPV)**
  - **Check logs**

Underutilization

Case Study: Mid-Tier DB Tuning

- Internal HR site
  - Poor throughput and response time
  - Sun 24-way system shared with other applications
  - Connecting to backend Oracle database
    - Same database used successfully for other tests
    - But installed at the time on different hardware
- Tools and tests:
  - Tivoli Performance Viewer
    - Showed connection pool utilization very high
    - One connection per active thread under full load
    - Extremely long hold time per thread (seconds)
- Resolution
  - Focused tuning on database
  - Found missing index overlooked in database port
  - Response times dropped from 17 secs to sub-second

Underutilization

Case Study: Test Client Tuning

- European retailer e-Commerce site
  - Poor throughput and response time
  - AIX 6-way system
  - Connecting to massive DB2 mainframe database

- Tools and tests:
  - Tivoli Performance Viewer
    - Very little activity on threads or connection pools
  - Thread dump
    - Showed almost no active threads
  - Network Protocol Analyzer
    - Showed low traffic volumes between test client and the Web site

- Resolution
  - Focused on front end componentry
  - Determined test environment inadequately equipped
  - Also think times and script scenarios not in line with customer activity

Underutilization

Case Study: Site Equipment

- European bank
  - Home banking Web site
  - Clustered AIX systems
    - Identical WAS installations on each
  - Poor throughput, utilization from only one machine in the cluster

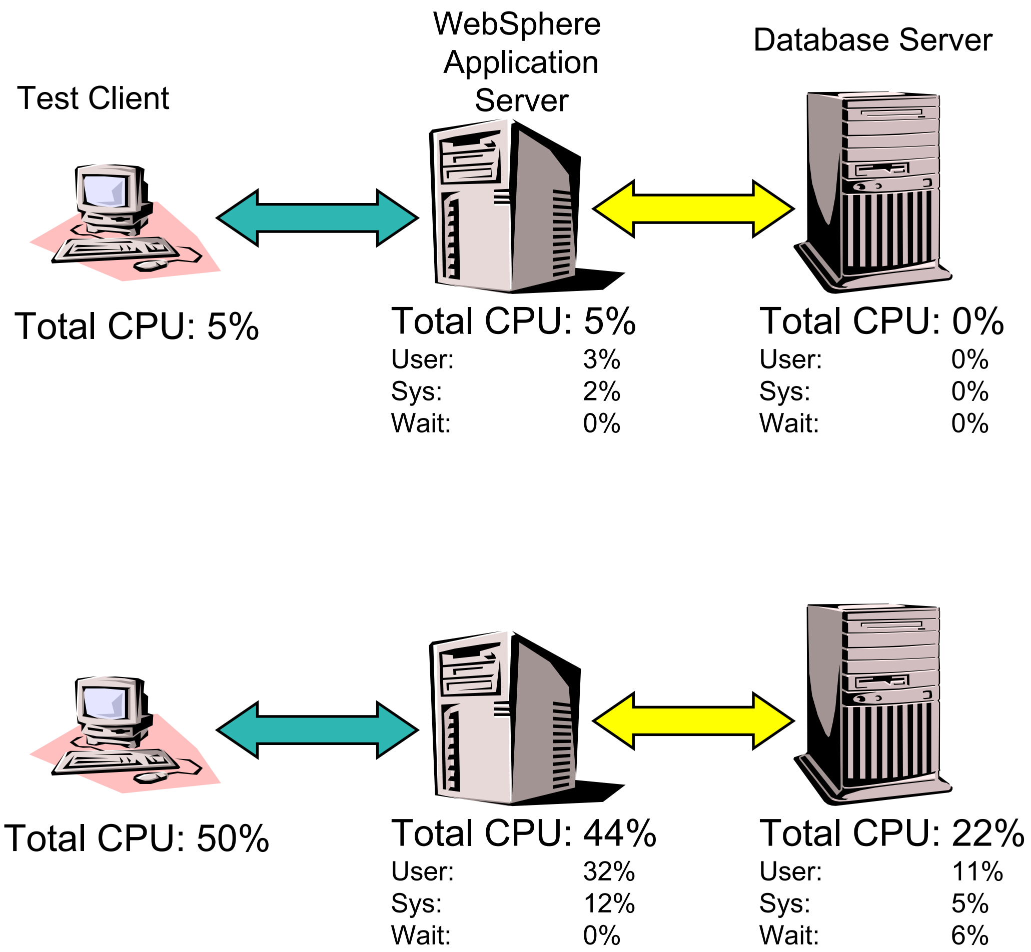
- Tools and tests:
  - Thread dump of poorly performing machine JVM
    - Showed almost no active threads

- Resolution
  - Focused on HTTP Server and other front-end components
  - Found a misconfigured router
    - Router original configured for production site
    - Throttled incoming traffic to prevent denial-of-service attacks
    - One of the throttled IP addresses was that of the afflicted server
  - Resetting router gave us even loading

Bottleneck Pattern

Bursty Behavior

- **Large system fluctuations**
  - **CPUs either**
    - **Near idle**
    - **Busy**
- **Cycles low to high during run**
- **Corresponding fluctuations in**
  - **Throughput**
  - **Response times (maybe)**



Test Client

WebSphere Application Server

Database Server

Test Client: Total CPU: 5%

WebSphere Application Server: Total CPU: 5%
User: 3%
Sys: 2%
Wait: 0%

Database Server: Total CPU: 0%
User: 0%
Sys: 0%
Wait: 0%

Test Client: Total CPU: 50%

WebSphere Application Server: Total CPU: 44%
User: 32%
Sys: 12%
Wait: 0%

Database Server: Total CPU: 22%
User: 11%
Sys: 5%
Wait: 6%

Bursty Behavior

Symptoms and Considerations

- Symptoms:
  - The systems do not run consistently under load
  - Transaction rates and response times fluctuate correspondingly

- Indicates an intermittent resource deprivation
  - Burstiness is the most difficult problem to diagnose

- The usual suspects:
  - Third-party interference
    - Machine back-ups, increased network traffic, etc.
  - Buffer and cache management
    - Periodic (and expensive) cache refreshes
  - Timeout settings
  - Misconfiguration

- Use WebSphere to begin problem isolation process
  - Find out what the Web application is doing during idle times (thread dump)
  - Monitor key resources such as heap and database connections (TPV)

Bursty Utilization

Case Study: Web Application

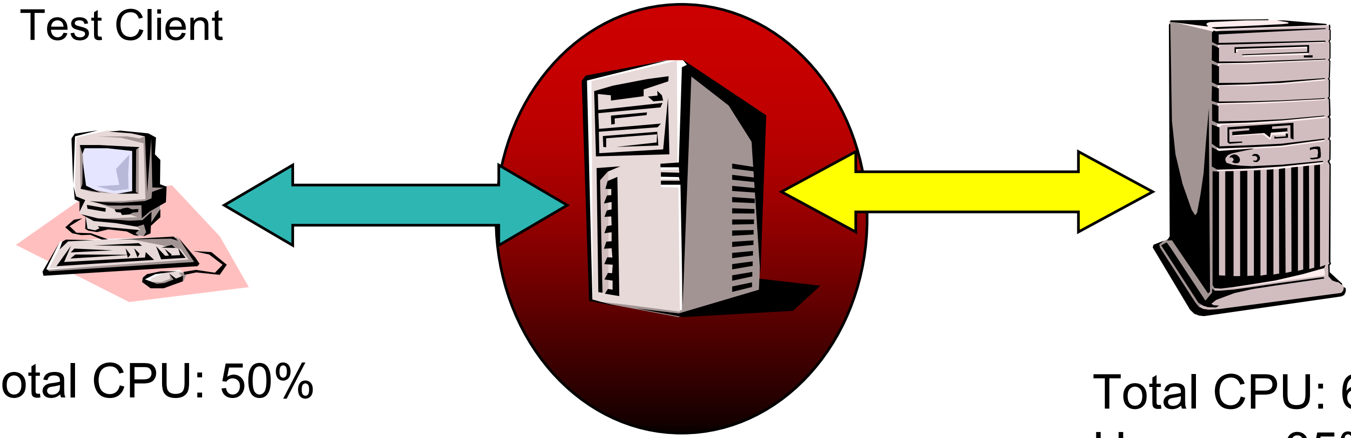
- B2E site
  - AIX 6-way machines
  - Runs consistently stalled at 5 minute intervals during tests
- Tools and tests:
  - Thread dump
    - Showed most threads fully engaged, but waiting for a lock to clear
    - One thread performing a database query
- Resolution
  - Focused tuning on application internals
  - Found application was reloading a very large cache from database
    - Setup to reload every 5 minutes
    - Part of test process called for restarting the app server before each test
  - Reset reload to occur once a day solved bursty utilization problem

Bursty Utilization
Case Study: Host Systems

- Brokerage Trading Site
  - AIX 4-way machines connected to complex mainframe trading system
  - Runs consistently oscillated
- Tools and tests:
  - Thread dump
    - Showed threads engaged
    - Threads waiting on query to mainframe system
- Resolution
  - Focused tuning on mainframe application
  - Mainframe specialist found problem in buffers in mainframe software
  - Resolved oscillation in the short term
    - Returned when system went to higher loads/more application servers
    - Additional tuning required

Bottleneck Pattern

CPU Busy



Test Client

WebSphere Application Server

Database Server

Total CPU: 50%

Total CPU: 98%	
User:	68%
Sys:	30%
Wait:	0%

Total CPU: 60%	
User:	35%
Sys:	10%
Wait:	15%

CPU Busy

Considerations

- CPU processing capacity exhausted
  - Throughput maximized
- Also indicates
  - Sufficient traffic arriving to drive CPU to full utilization
  - Sufficient remote backend resource availability
- Focus on the application to reduce CPU burden
  - Code profiling, javacore traces, etc. to find bottlenecks
  - Activate "-verbosegc" to monitor garbage collection cycles
  - Test individual code paths under load to find slow components
- WebSphere Application Server apps tend to be CPU constrained
  - Our goal is to drive the CPU to full utilization

CPU Busy
Case Study: Web Application

- Dot com media site
    - AIX machines
    - CPU 100% utilized
    - Poor throughput and response times
- Tools and tests:
    - Thread dump
        - Showed threads fully engaged in String manipulations
- Resolution
    - Focused tuning on application internals
    - Smarter String management
        - String buffers, better concatenation strategy
    - Improved throughput and response times

Using Patterns
Performance PD

- What is the best training approach for Performance issues?
  - Mentoring
    - Hindered by time constraints
    - Lacks breadth of learning
  - Tool education
    - Focuses on "how" vs. a "why" and "when" approach
    - May collect data without understanding its meaning
  - A pattern analysis approach gives the learner broad skills
    - Symptoms, tests, tools, and analysis techniques
    - Provides a methodology to support mentored learning

- Benefits
  - Faster training cycle
  - Focus on the Web site as a system
  - Use WebSphere's role in the site to isolate problem areas
  - Save money

Summary

- WebSphere’s location in most Web sites
    - Allows us to isolate the problem areas with a few tests
- Bottlenecks tend to follow a set of patterns
    - For each pattern, identify:
        - The most common causes
        - A set of checklists/decision trees to:
            - Isolate these common problems
            - Isolate problem areas requiring advanced tuning
    - Use these patterns as the basis for training
        - Follow with tool education and mentoring