



IBM ITSO Poughkeepsie - the zSeries Redbooks people

**WebSphere** software

# (Simple) Performance Management Methodology

Holger Wunderlich  
wunderl@us.ibm.com

[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

Copyright IBM Corp 2001



## [performance]

[performance] noun

The performance of a person or machine is how well they do a piece of work or activity.

Some athletes take drugs to improve their performance.

These cars have a reputation for poor performance.

High-performance (Fast, powerful and easy to control) cars are the most expensive.

(Cambridge International Dictionary of English)



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Objectives

- Performance Management
  - ▶ Capacity planning & sizing
  - ▶ Load Injection
  - ▶ Reference Workloads
  - ▶ Major Tuning Steps
  - ▶ Monitoring & Tracing



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## zSeries stands for Quality...

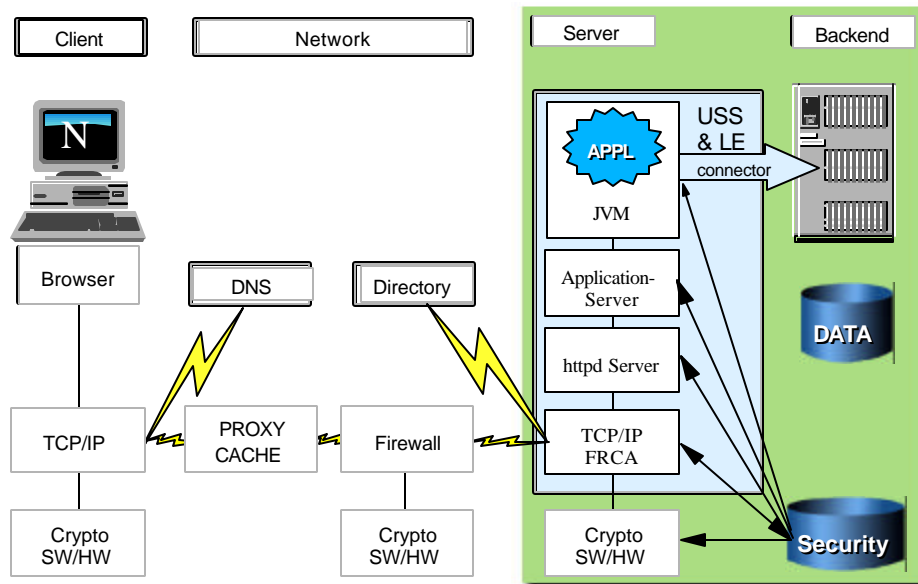
- Quality of service...
  - ▶ More Security
  - ▶ More Reliability
  - ▶ Accounting
  - ▶ Transaction Security
  - ▶ All this is not for free...
  - ▶ In fact it costs instructions

zSeries has proved the last years that it is able to deliver highest transaction rates and highest quality of service at one time.

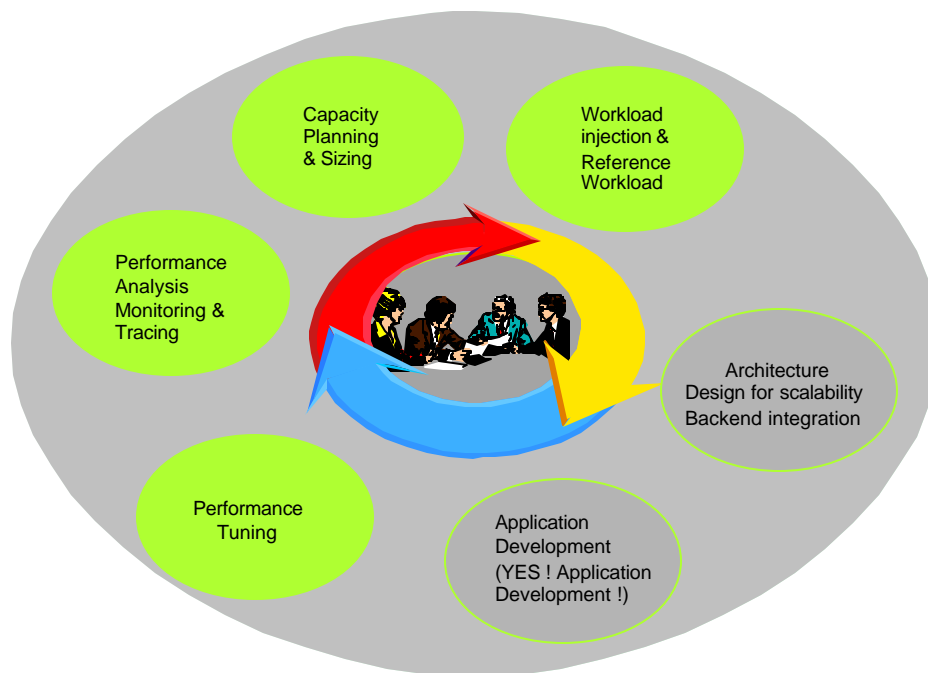


[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Main components of web application end-to-end performance



## Performance Management Tasks



## Performance Management

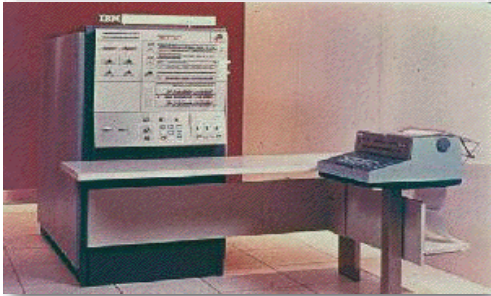
- ▶ performance design
- ▶ capacity planning
- ▶ sizing



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Historical Performance Design...

...when we talked about split seconds and human productivity

- asm programming
  - ips/ics/opt tuning
  - I/O reduction
  - some science
- 
- A photograph of an early computer system, likely from the 1960s or 1970s. It features a large, dark-colored mainframe unit with a control panel and a keyboard. A small, light-colored terminal or printer is connected to the mainframe by a cable. The system is situated in a room with a tiled floor and a plain wall.
- known application
  - known tx rate
  - few computation resources in a very controlled network
    - ▶ good system skills
    - ▶ easy derived capacity and hardware planning

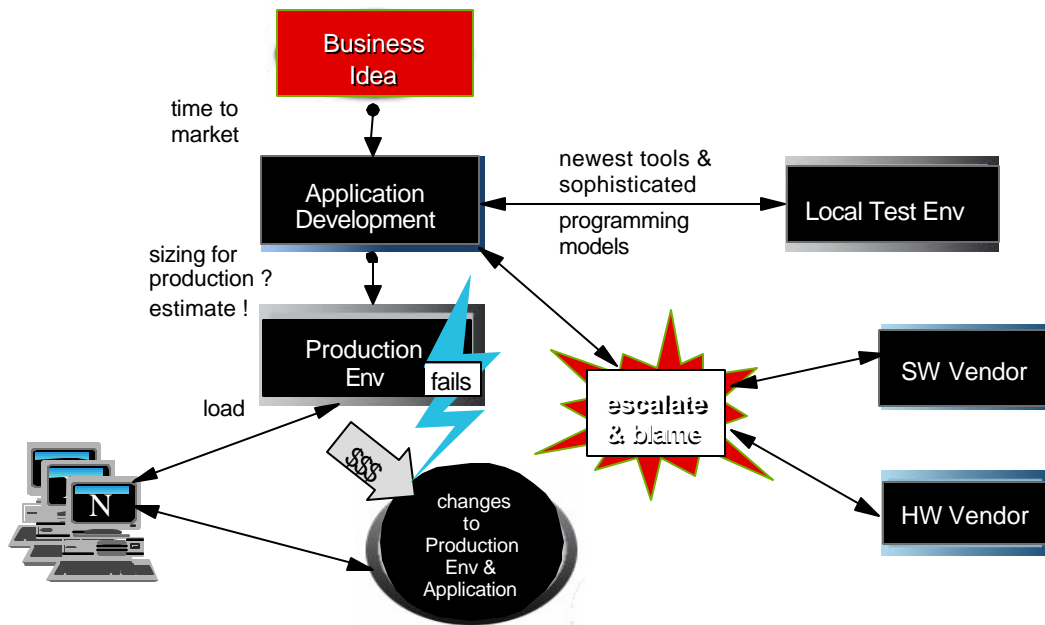
- then came WLM & Hardware
- SLAs have been matched to Goals



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## WWW Performance design...

...when WWW got a new meaning



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Performance Management Sizing

- Two approaches
  - ▶ scientific
    - get many informations
    - put it into an tool
    - come out with an estimate of processor load
  - ▶ empiric
    - run an prototype or 'like' workload
    - take your measurements
    - do your performance projections

Results will be in both cases not accurate. Don't blame us, please. It's very difficult to size for the Internet. Are you sizing for DoS-attacks too ?



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## scientific sizing approach

- We give you a questionnaire
- You give us the answers
- We put into an tool
  - ▶ And give you CPU & memory requirements
  - ▶ Questionnaire available for IBMers + BPs

number connections

page size

concurrent clients

% of SSL

programming model

security model

backend integration

software levels

service levels

response time expectations

visits

encryption key length

SSL session time-outs

proxy

firewall

how many visits

how many pages per visit

session tracking

%static %dynamic

backends

used connectors

CTG level

commarea usage

JDBC or SQLJ

IMS connect / OTMA

middle tier

JSP used

hardware used

complexity of tx

complexity of application

contact your IBM representative



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## empiric / pragmatic sizing approach

- Scientific approach is still OK for static pages, common programming models
- For Application Servers its sometimes better to derive the sizing from test runs against your early application
  - ▶ Establish regularly test runs on same architecture than production
  - ▶ Programmers see bottlenecks early
  - ▶ You get an early feeling on the resource consumption
  - ▶ Concerns can be raised early
  - ▶ Usually performance gets better tuning testing
- If an early code is not available in time, start with a common test suite that covers at least the same components than your application (EJB, DB2, LDAP)
- Get estimates on the number of requests and expected response time (SLA !)



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

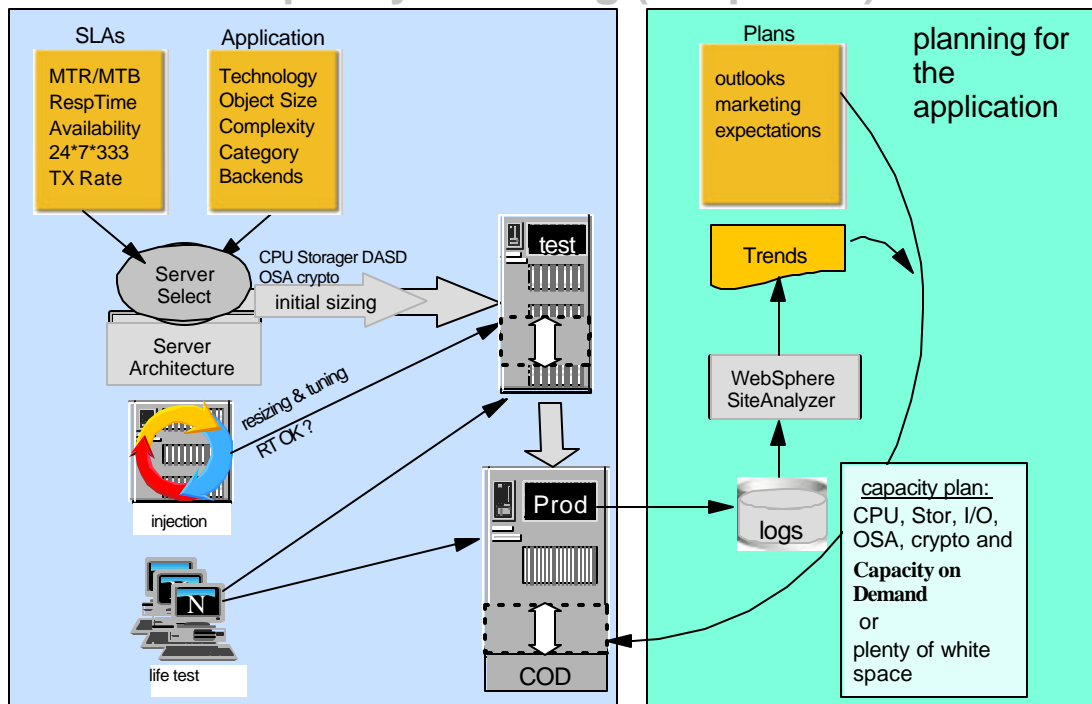
## pragmatic sizing approach flow

- Setup a controlled environment according to general sizing guidelines
- Install reference workload or early application
- Run simulated workloads
- Record data
  - Pathlength
  - Throughput
  - Response time
- You calculate/estimate resource requirements from your measurements and the expected transaction rate / response time



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Server Capacity Planning (simplified)



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## sizing

Don't forget... it's not only that simple web page (0,1m)

it's about instructions...

if sizing for an application you might add CPU resources for

- + security
- + tracing
- + peaks
- + logging / accounting
- + reporting & mining
- + encryption
- + high availability (redundancy)
- + workload classification and local clones (WLM)
- + concurrent backups

and might subtract if running in a

- mixed workload environment

have you planned a staging server ?



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)



IBM ITSO Poughkeepsie - the zSeries Redbooks people

**WebSphere** software

## Performance Management

- Load Injection
  - Controlled Environment
  - Load Injector
  - Workloads



Tip in front

No fancy tools necessary for heavy applications

Holger Wunderlich  
wunderl@us.ibm.com

[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

Copyright IBM Corp 2001





## Load Injection controlled environment

- change management
- test runs without mixed workloads
- LPAR guarantee
- controlled changes to the applications
- or to the application
- protocol of the test runs
  - ▶ again: pathlength, throughput, response time, parameters
- don't change the load generators after adjustment



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Installation & Usage

- Det DAT from
  - ▶ <http://www.alphaworks.ibm.com/tech/dat>
- Run the install.exe on your load generation PC
  - ▶ It's Java and runs on any platform
- Run the program



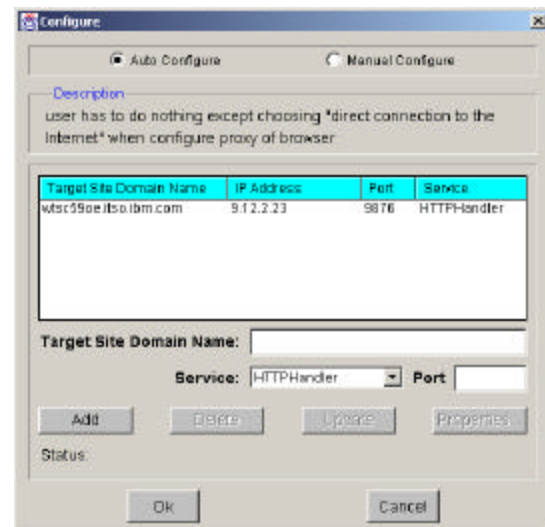
Dat.Ink



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Installation & Usage

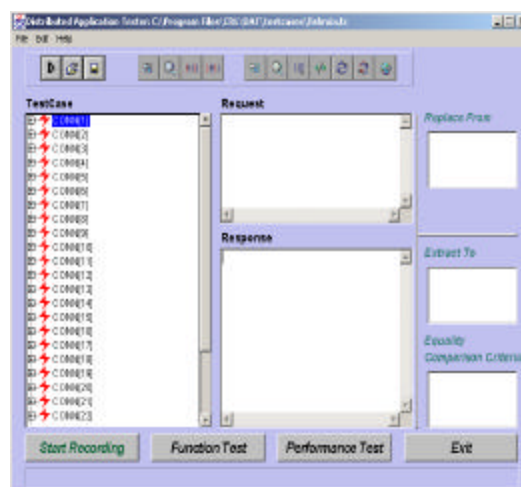
- Pull down file/configuration
- Add your webserver here
  - ▶ IP address not allowed
- Add the ssl port too
- Click auto configuration
- Click ok



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Testcase

- To generate a testcase
  - ▶ Put DAT in recording mode
  - ▶ Start your browser
  - ▶ Load the webpage referring to your reference workload (pingServlet)
  - ▶ Stop recording
  - ▶ Save as PingServlet.tc
  - ▶ Repeat until you have all testcases recorded



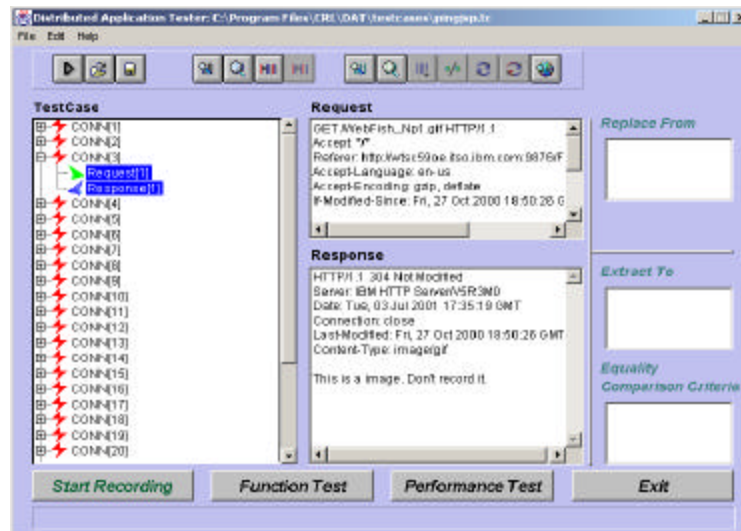
[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Installation & Usage

### ■ Hint:



when you open one of the connections DAT will show you the http request and response headers



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Installation & Usage

### ■ Hint:



if you now pulldown file/view detailer you will see your http request... in detail

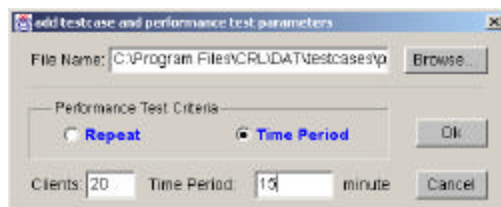
maximize the window to get access to the scrollbars !



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Installation & Usage

- To do an performance run against your installation now push performance test
- Pull down testcase and add the required run
- You can add various testcases and save later as a group (good for overall throughput)
- Select the number of clients (maybe 20-30)
- Select a fixed time period instead of repeats
- Press start and the run begins



Testtools are not real users

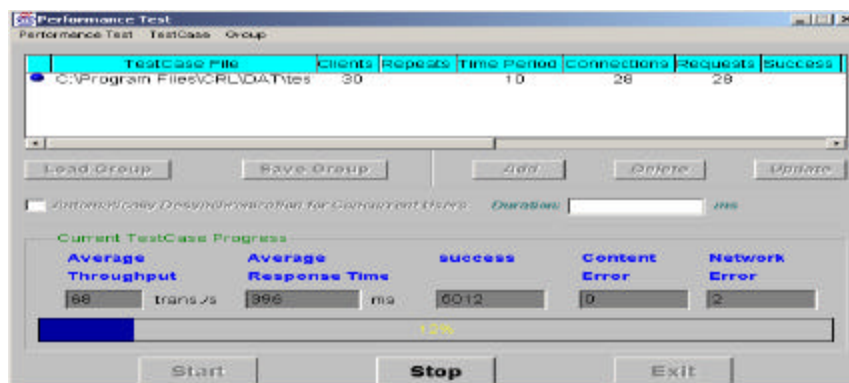
Minimum, ROT: 10% of users logged on, 10% logged on are active  
Generally peak rate is 2 X to 4 X average



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Installation & Usage

- After the run DAT will show the performance data if you scroll to the right.
- Record now all the performance data together with essential system data (CPU time...) in a worksheet



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## DAT Summary

- Now you can always check if your system change was successful
- There are many numbers that can be used to get insight of your system throughput
- To begin just have a look at
  - ▶ CPU Time / Pathlength
  - ▶ Average responsetime
  - ▶ Number of successful transactions
  - ▶ Errors
- After a loadrun always check system and server logs if something unusual happened!



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Workloads

- Your Application
- Static
  - ▶ Homegrown, download from the internet
- Dynamic
  - ▶ /usr/lpp/was35/AppServer/hosts/default\_host/examples/servlets/
    - SimpleJSPBean.class
    - SimpleJSPServlet.class
  - ▶ Sessiontracking sample supplied (using JDBC, DB2 and connection pooling)
  - ▶ Hello World Servlet
- Enterprise Workloads
  - ▶ see WSPBSA

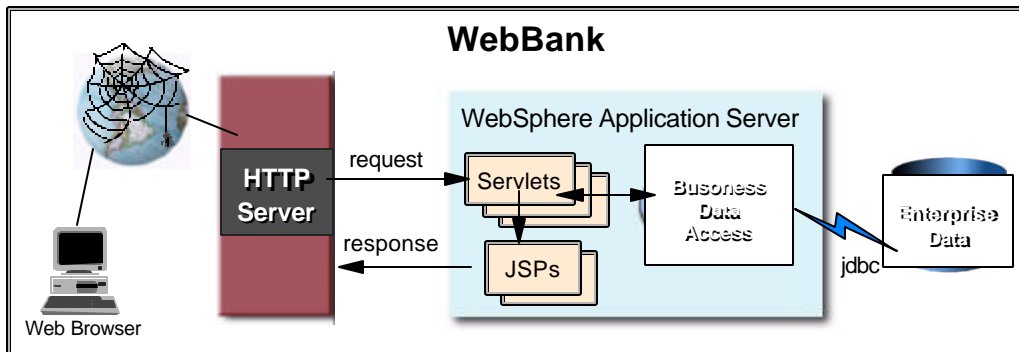


[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## WebSphere Performance Benchmark Sample application, Trade2

- WebSite
  - ▶ [http://www-4.ibm.com/software/webervers/appserv/wpbs\\_download.html](http://www-4.ibm.com/software/webervers/appserv/wpbs_download.html)
- For installation tips on trade2 write me an email
- For installation tips on WSPBSA wait for the coming Redbook

### TRADE2



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Performance Management

### Tune & Analyze



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## What can you achieve ?

### ■ German Customer

- ▶ Complex WebSphere SE application, highly OO, many JSPs
- ▶ Migration from WAS 3.02 to WAS 3.5 reduced CPU-consumption to 41%-44% of the original CPU-consumption. The numbers depend slightly on the application code.
- ▶ Several quick changes in the application code reduced CPU-consumption contributes to 46% of the original CPU-consumption
- ▶ Both sum up to 79% reduction of CPU-consumption compared to the original application under WAS3.02.



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Tuning base vocabulary

- response time
  - ▶ time spent in client
  - ▶ time spent in network
  - ▶ time spent in servers
- throughput
- resource utilization
  - ▶ path length
  - ▶ waiting times

Tuning (within mixed workloads) is

- ▶ Analyzing runtime
- ▶ Avoiding of unnecessary cycles
- ▶ Assigning resources to workloads
- ▶ Restricting workloads
- ▶ Making adjustments and ordering hardware!

Matching performance to SLAs



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## First of all: Resources

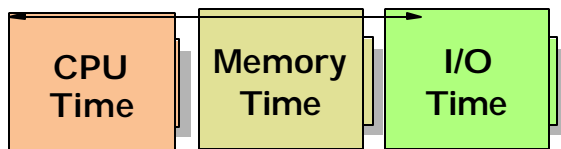


- Resources & Queues

- ▶ CPU
- ▶ Memory
- ▶ I/O
- ▶ HW - Ftr
- ▶ Bandwidth
- ▶ Services

- There is interaction:  
e.g.

- ▶ Memory access costs CPU
- ▶ I/O is for free on zSeries ...but slow and can affect queues
- ▶ Same for external services



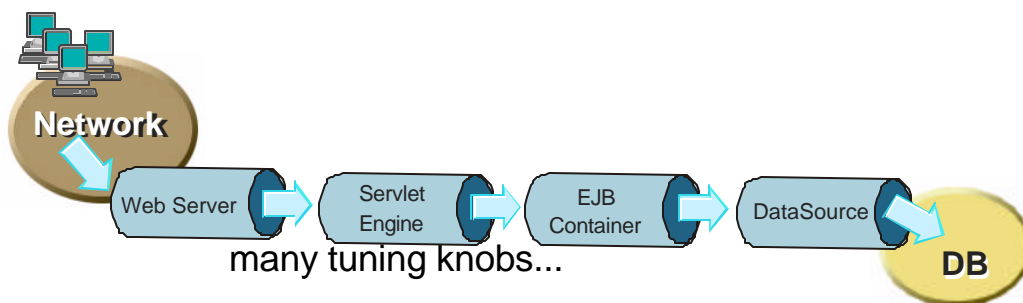
- ▶ Slow network can lead to excessive lockings in TX scope



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Queuing

- Resources are a problem if they are queued
- Queues are virtually not there if you don't queue
- There are open and closed queues
  - ▶ closed queues a brilliant mechanisms to control the loading of your servers

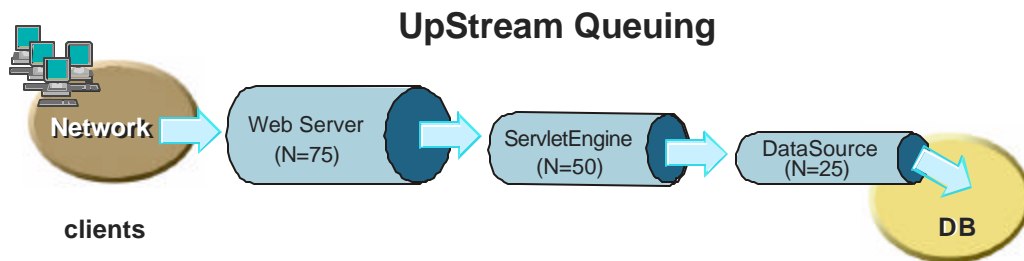


[www.redbooks.ibm.com](http://www.redbooks.ibm.com)



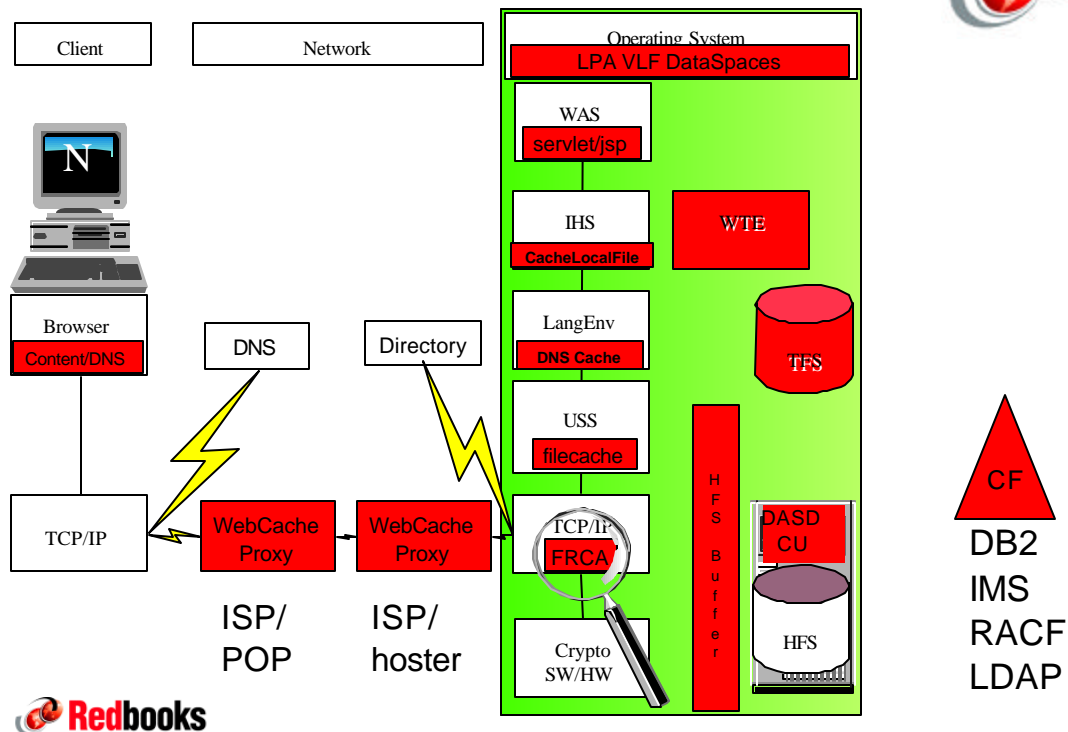
## Queuing

- keep queues outside of WAS
- if you see queuing problems without cpu saturation you might put in another WAS instance



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Caching (moving work from i/o to memory/cpu)



## Tuning general guidelines

- Know what to tune for (categorize & understand your application)
- Minimize I/O,Cache (where appropriate)
- Check best practice for your software
  - ▶ WebSphere, IHS, USS, Java, DB2...
- Software service level
- Keep a record on what you have done, at least
  - Pathlength
  - Responsetime
  - Throughput
  - Parameters



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## TOP 10 most common mistakes

- ▶ Inappropriate Software Levels
  - ▶ IEEE/WAS combination
- ▶ Over engineered Application
  - ▶ handmade frameworks, no high tx design, too much OO
- ▶ JIT disabled
- ▶ Servlet Reloading
- ▶ USS-SAF misconfiguration
  - ▶ IRRUMAP, IRRGMAP, SURROGAT, RACLIST
- ▶ Traces in IHS & WAS
  - ▶ -vv, jvm trace, MVS trace, jdbc trace
- ▶ Memory Usage in JVM & GC
- ▶ No Connection pooling
- ▶ Inefficient backend connection
- ▶ Base tuning
  - ▶ IPS, ICS,Resources, HW enable, LAngeage Environment
- ▶ Network Infrastructure
  - ▶ Edge Server, Caching, Bandwith



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)



## WebSphere software

# Performance Management

- Monitoring
- Tracing



**Tip in front**  
It even affects what you  
measure. Take care.

Holger Wunderlich  
wunderl@us.ibm.com

[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

Copyright IBM Corp 2001



## Monitoring & Tracing

- Monitoring external, Tracing internal
- To find problems in the server infrastructure usually tools like RMF are the right choice
- To find problems in the application profilers are the right choice
- The best is not to look at it... keep traces off !
- MVS
  - ▶ CTRACE
  - ▶ SMF
- TCP/IP
  - ▶ VTAM Trace
- WebServer
  - ▶ LE TRACE
  - ▶ ServerTrace
- WebSphere
  - ▶ JDBC Trace
  - ▶ JavaTrace
  - ▶ Object Level Trace
  - ▶ Jinsight, Jprobe
  - ▶ WebSphere Trace



## Simplified methodology for performance management / Summary

- make zSeries environment ready and controlled
- put load injectors in place
- get sample applications installed
- give developers access & instructions to place their applications on zSeries
- run load simulation
  - ▶ with reference workload for IS tuning and sizing
  - ▶ with real application for application tuning and sizing (regularly)
- analyze: trace & monitor, collecting data
- capacity management & performance tuning



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## Additional Information

- Redbooks: [www.redbooks.ibm.com](http://www.redbooks.ibm.com)
  - ▶ WebSphere V3 Performance Tuning Guide, SG24-5657-00
  - ▶ IBM San Francisco Performance Tips and Techniques, SG24-5368-0, See Chapter 9
- Whitepapers:  
[www.ibm.com/software/webserver/appserv/whitepapers.html](http://www.ibm.com/software/webserver/appserv/whitepapers.html)
  - ▶ WebSphere Application Server Development Best Practices for Performance and Scalability
  - ▶ IBM WebSphere Application Server Standard and Advanced Editions - A Methodology for Production Performance Tuning
  - ▶ zOS eBusiness tuning  
[www-1.ibm.com/servers/eserver/zseries/ebusiness/perform.html](http://www-1.ibm.com/servers/eserver/zseries/ebusiness/perform.html)
- Bookstores
  - ▶ "Java TM Performance and Scalability Volume 1, Server-Side Programming Techniques", by Dov Bulka, First Printing May, 2000, Addison Wesley
  - ▶ "Practical Queuing Analysis" by Mike Tanner, pub. McGraw-Hill in their IBM series. ISBN 0-07-709078-0.
  - ▶ Patrick Killelea, Web Performance Tuning O'Reilly



[www.redbooks.ibm.com](http://www.redbooks.ibm.com)

## More Info

- "A Scalable System for Consistently Caching Dynamic Web Data "  
(postscript, by Jim Challenger, Arun Iyengar and Paul Dantzig).
  - ▶ In Proceedings of IEEE INFOCOM'99, New York, March 1999.
  - ▶ <http://www.research.ibm.com/people/i/iyengar/infocom2.ps>
- "A Scalable and Highly Available System for Serving Dynamic Data at Frequently Accessed Web Sites " (Jim Challenger, Arun Iyengar and Paul Dantzig).
  - ▶ In Proceedings of ACM/IEEE Supercomputing '98, Orlando
  - ▶ [http://www.supercomp.org/sc98/TechPapers/sc98\\_FullAbstracts/Challenger602/index.htm](http://www.supercomp.org/sc98/TechPapers/sc98_FullAbstracts/Challenger602/index.htm)