Performance, Sizing, Tips & Tools: The Latest and the Greatest for i5/OS

Common Luxemburg
September 28, 2006

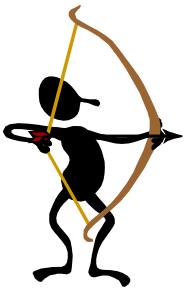
ON DEMAND BUSINESS™

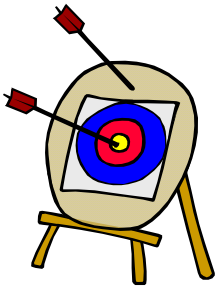
What is Performance?

- Definition:
  - Performance: The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage [IEEE-610.12]
- In general:
  - Predictability and timeliness of response is what performance is about
  - A “faster” system is not always enough; for example, a real-time system requires repeatable results
- A quote from the Institute of Computer Capacity Management (ICCM):
  - “A manager's goal should always be to strike the right balance between system function, processing costs, people costs, and performance. This is why the technical aspects of performance can never be entirely divorced from organizational politics.”

Successful Systems have at least Three Characteristics

- The right function
- Operational integrity
- Acceptable performance





All three are necessary or the solution would fail.

VICTORIA'S SECRET

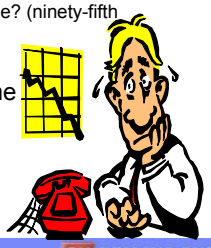
boo.com

An Example of a Throughput Disaster

- Application
  - Monthly production schedule creation
  - Target: 16 million items to be planned in an 8-hour batch window
- Testing
  - Unit and integration tests: Function runs correctly.
  - System test: Two weeks before going live, the system was still running after 48 hours.
  - Progress analysis indicated the likely time to completion: **154 years**.
- Resolution
  - System went live four months late.
  - The project incurred significant extra costs.

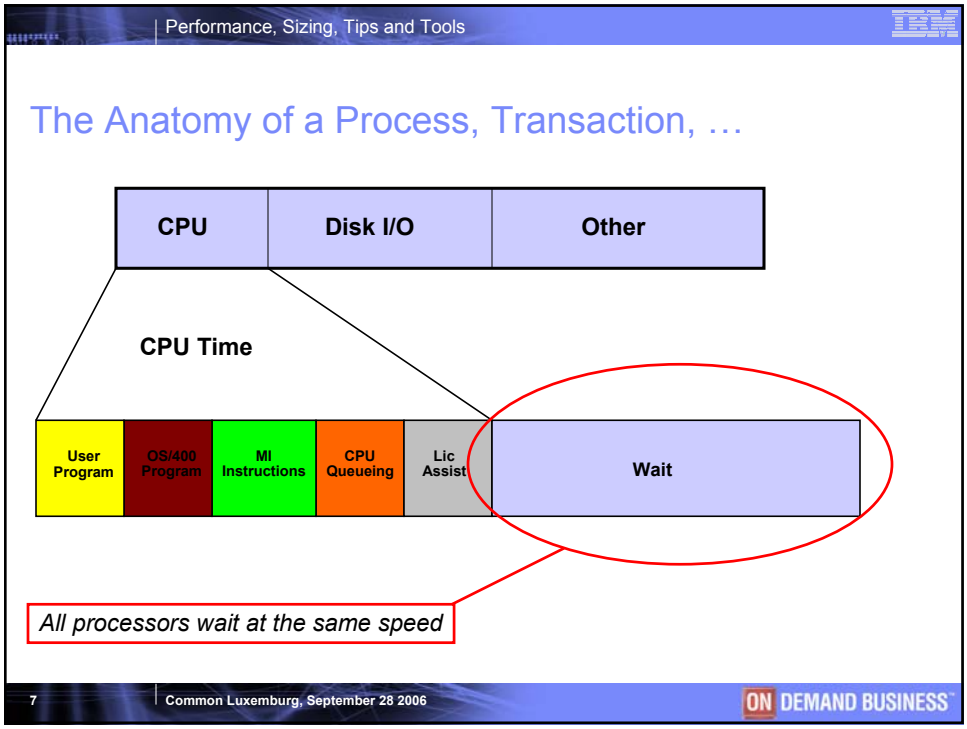
Understand the Requirements

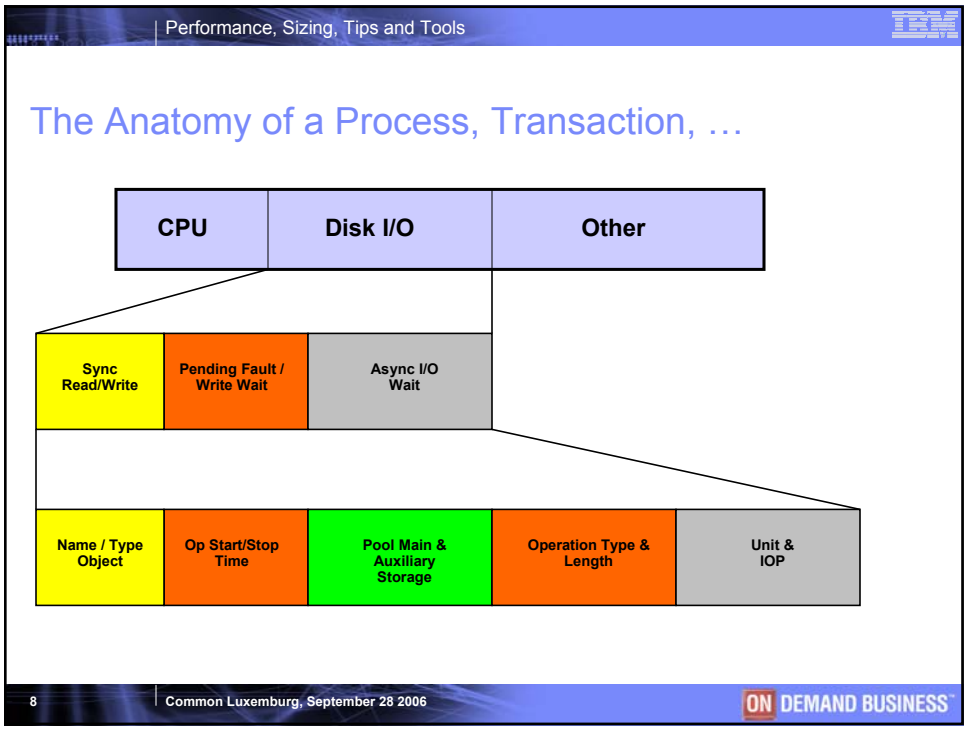
- A simple performance statement, such as “a response time of less than one second,” is insufficient. It should always contain several components, such as:
  - What is the performance requirement? (Response time? Transaction time?)
  - When is it to be delivered (in defined service hours)?
  - Where is it to be measured? (On which systems, using which tools?)
  - For what workload is it to be valid? (Are test transactions available?)
  - For which users or systems? (And how many?)
  - How can you define the percentile to achieve the given value? (ninety-fifth percentile? average?)
- Remember: In the e-business world, response time can equal competitive advantage



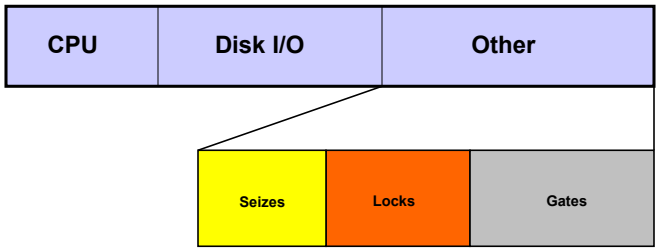
Remember that the Users of the System are Human

- People's senses are analog not digital.
  - For example, water temperature - freezing, cold, warm, hot, boiling
- Interaction perception:
  - People treat computers as people too.
  - The steps in people’s sense of time are as follows:
    - Instantaneous = below 0.2 seconds (for example, typing)
    - Immediate = Sub-second (like-minded colleague / aggressive stranger)
    - Fast = under 2 seconds (frequent or standard request)
    - Normal = 3 seconds (typical conversation)
    - Extended = 5 seconds (slow speaker - ideally needs to make sound in the long pauses to show they are still speaking)
    - Too long = about 17 seconds (Other party gives up, saying “Come and tell me when you have an answer”)
- Performance does not mean blazing speed. It means an acceptable and predictable level of responsiveness. Aim for consistency!


Performance, Sizing, Tips and Tools
The Anatomy of a Process, Transaction, …
CPU
Disk I/O
Other
CPU Time
User Program
OS/400 Program
MI Instructions
CPU Queueing
Lic Assist
Wait
All processors wait at the same speed
7
Common Luxemburg, September 28 2006
ON DEMAND BUSINESS


Performance, Sizing, Tips and Tools
The Anatomy of a Process, Transaction, …
CPU
Disk I/O
Other
Sync Read/Write
Pending Fault / Write Wait
Async I/O Wait
Name / Type Object
Op Start/Stop Time
Pool Main & Auxiliary Storage
Operation Type & Length
Unit & IOP
8
Common Luxemburg, September 28 2006
ON DEMAND BUSINESS

The Anatomy of a Process, Transaction, …

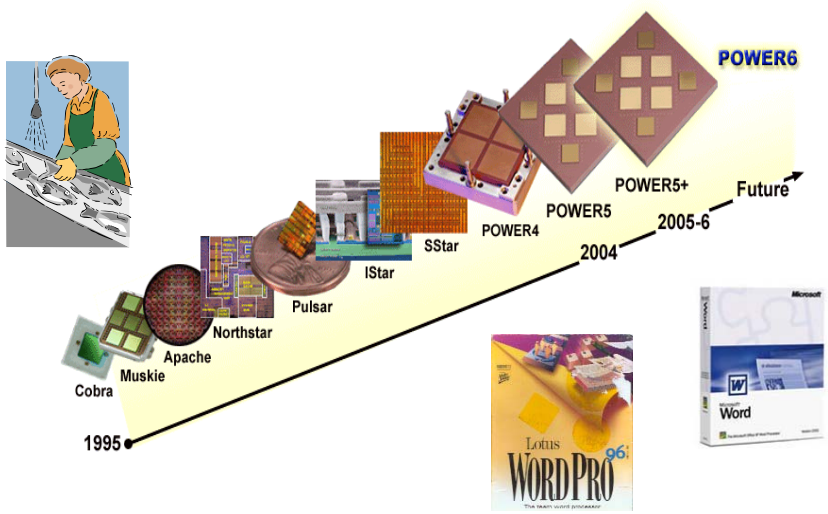


Components and their Performance Characteristics

What's out there?

- Processors – can somebody define it?
- Memory – does it scale? How much do I need?
- Disk units – how do they behave?

A Processor can be Lots of Things …



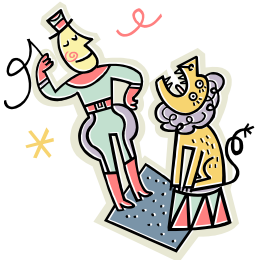
What is a Processor?

We like to say:

We have the fastest 4-way in the industry!

But what is a 4-way?

The answer to this seemingly simple question, is a little like…..



intel

A Three

AMD

Ring

Circus

Sun microsystems

Why the Confusion?

IBM has Power4 and Power5 processor chips that contain 2 cores on the chip – We naturally called this 2 processors, because each core acts largely independently of the other.

Of course, we also have SMT, which runs two processor threads on each core, making it look like each core is two processors to the software……

Intel figured out that IBM's 1999 HMT (hardware multithreading) on Power3 was a good thing, and came out with hyperthreading, that allows multiple processor threads to interleave on a single core. Recently, they have come out with Dual-core "Processors"

With both dual cores and hyperthreading, they have the ability to match words with Power5 - - - as soon as they start shipping it on more than workstations

Sun has introduced UltraSPARC IV with Chip Multithreading

Which means they have dual (independent) core chips that they call single processors that have no actual multithreading in the same sense that IBM and Intel have – not unlike Power4 but very unlike Power5

AMD has introduced "dual-core processors" and has begun to talk about "multi-core processors"

Although their verbiage still sometimes says things like "With multi-core technology, your computing load can be spread across two or more processors" (http://multicore.amd.com/WhatIsMC/)

Consider the 4-way "Processor"

Technology	Chips	Processors	Cores	Threads
IStar/SStar (w/HMT)	4	4	4	8
Power4	2	4	4	4
Power5 (w/SMT)	2	4	4	8
Intel-single core (w/hyperthread)	4	4	4	8
Intel-dual core (w/HT)	4	4	8	16
AMD-single core	4	4	4	4
AMD-dual core	4	4	8	8
Sun SPARC3	4	4	4	4
Sun SPARC4	4	4	8	8
Sun Niagara	4	4	32	128

What IBM says

IBM - z	A processor is a core
IBM - i	A processor is a core
IBM - x	A processor is whatever our processor providers proclaim it to be. We cannot sell our solutions using different vocabulary than others using the same technology
IBM - p	A processor is a core, but we should really call it a “processor-core”
IBM - SWG	We would like to say a processor is a core, but -- may need to follow definitions of processor manufacturers -- agree to use processor as a chip for two OpenPower systems -- not all cores are created equal -- we have a challenge getting equal payment for equal performance, when processors and cores all perform differently
Mostly	We will deal with cores as the primary units of computer processing.

Performance Analysis: What must be done?

- Be careful where you step
- Know your architecture and your tools
  - Some tools will report multithreading as individual processors, even though the threads are not using independent resources
    - Tools that report processor threads as processors may yield different analysis impressions depending on whether multithreading is active or not
  - Most tools will report cores as processors, even though the product is advertised with processor chips
- Of course, virtualization of processors throws entirely different variables into the discussion

Capacity Planning: What must be done?

- Be careful where you step
- For capacity planning – must pay attention to
  - Processor Cores
  - Processor Threads
  - And the relative strengths of these to deliver performance for a particular workload
- Don't predict response time (or batch run time) improvement based on rPerf or CPW comparisons between Power4 and Power5
  - These are reasonable indicators of commercial multitasking capacity, but they are NOT good predictors of single transactions, since Power5 results will include SMT

Conclusion

- Capacity Planning now requires
  - The understanding of the unique components that make up a "processor"
  - The understanding of how a workload might use those components
  - The understanding of how a workload might change with a different set of components
- Software design (at least kernel, but probably middleware) now requires
  - The understanding of which components are available to use
  - Multiple optimization paths depending on this discovery
  - Possible just-in-time microcode loads of "on-demand accelerators"
- Sometimes
  - An application will directly use all the components
  - The kernel or firmware may "elect" to use some components on the application's behalf
  - The components won't be used
- And for processor performance analysis, we might just use …

Another Kind of Processor

Settings that affect Processor Behavior

- Processor multithreading set at system level (QPRCMLTTSK system value)
  - Options are “Off” - “0” - , “On” - “1” - or “System controlled” - “2”
  - “System controlled” is valid for any hardware
  - Value change only takes effect after IPL
- When using partitioning, a number of virtual processors can be defined at the partition definition level:
  - Virtual processors are the whole number of concurrent operations which the operating system can utilize
  - ROT: Balance virtual processors to processor units:
    - If 1 processor or less, define 1 virtual processor
    - If 2 “ “ “ , define 2 virtual processors
    - Imbalance between virtual and real processors can slow down batch processing
- Performance Tools capture and report all aspects of processor capacity used

Many more Settings that affect Processor Behavior

- Run priority attribute for jobs and system values that affect them:
  - Dynamically adjust job priorities within priority bands (QDYNPTYSCD)
  - Dynamically adjust job priorities for interactive jobs (QDYNPTYADJ): only valid on systems with # interactive and total processor capacity and requires QDYNPTYSCD to be on
- Time slice settings
- Activity settings (on memory pools; system-wide):
  - Threads versus jobs
- Job mixes in subsystems

Data and Methodology for CPU Utilization

- QAPMSYSTEM table CPU calculation fields for:
  - Partition (System) Capacity Information
    - SYSCTA: Total CPU time (in milliseconds) configured (guaranteed capacity)
    - SYSPTU: Total CPU time (in milliseconds) used
    - SYSUTA: Available CPU time (in milliseconds) that could have been used
  - SYIFTA: 5250 OLTP CPU time (in milliseconds) configured/Interactive CPU time available capacity configured for use by the partition
    - SYIFUS: Interactive CPU time (in milliseconds) charged (used) to 5250-OLTP capacity
    - SYIFTE: Elapsed time (in milliseconds) when 5250 OLTP threshold was exceeded
- Shared Processor Pool Information
  - SYSPLA: Total CPU time (in milliseconds) in shared processor pool
  - SYSPLU: Total CPU time (in milliseconds) used by the shared processor pool partitions

Total Configured CPU Util % = (SYSPTU/SYSCTA) * 100
Total Available CPU Util % = (SYSPTU/SYSUTA) * 100

Partition Sizing – CPU Utilization Considerations

- Dedicated processors = best performance
  - Example: Two 4 dedicated processor partitions yield more total CPW than single 8 processor partition
- For shared, micro partitioning best performance:
  - Assigned processors greater than or equal to 1
  - Virtual processors 2x of assigned processors if short CPU per transaction
  - Consider uncapped partitions, but...
- Less than expected partition performance, if
  - Less than one whole processor assigned to a partition and high CPU workload
  - Virtual processors 10x of assigned shared processors
  - Tasks run for longer than fraction of single processor. For example, a .6 processor and task runs for longer than 6 milliseconds
- Check: “Sizing IBM i5/OS Work on IBM System i5 Partitions” - SG24-6656

Partition Sizing based on CPW

- Size your workload requirement – determine CPW required
  - For example 1.100
- Calculate the CPW per processor on the selected model
  - For example 520 = 3.300
- Determine multiplier for processor efficiency
  - 3.300/1.100 = 3
- Micro partitioning efficiency reduction: you decide: multiply by 5% minimum, up to 40% maximum. The basic concept is that the smaller the micro-partition the higher the multiplication factor
  - Example: 3 * 0,05 = 0,15
- Final partition processor unit size required
  - 1.100 * (1,15)/3.300 = 0.38 processors

Capped versus Uncapped Partitioning

- Capped (Only option up to V5R2/POWER4)
  - Limited by the entitled capacity (Max 100%)
    - The entitled capacity is assured (.75 Processing Units = 7.5 MS)
    - Partition CANNOT use any unused capacity in the processor pool
    - CPU queuing algorithms work
- Uncapped May exceed the entitled capacity (>100% - based on entitled)
  - The entitled capacity is assured
  - Partition CAN use unused capacity in the processor pool
  - Upper limit of utilization determined by VIRTUAL PROCESSORS
  - Partition Weight Factor determines ratio of allocation of unused capacity – when more than one partition requests unused capacity. Be careful here, a weight of zero is actually a capped partition. Default of 128 for all partitions generally is not a good idea

What's out there?

- Processors – can somebody define it?
- Memory – does it scale? How much do I need?
- Disk units – how do they behave?

General Rules

- Catch 22:
  - Avoid ☒☚●ⓒⓐ⑦⑦⓪⑤✍ **Swapping** but swapping cannot be avoided
- If
  - You know your application and
  - You know your system then
  - Automatic performance adjustment could be an option you do not really want to follow unless you customize shared pool behavior
- Using other tools requires you to know the tool very well also
- Don't stack tools

Rules-of-Thumb

- Machine pool set at about 10% of available storage – faulting should be less than 5 per second
- Limit the number of pools to a maximum of 8
- Base pool preferably used for subsystem management only:
  - Small amount of memory needed
  - Requires changes to all IBM shipped subsystems
- Calculate activity tool based on number of **threads** to run concurrently
- Expert cache is very useful for transactional and DB intensive work
- Avoid using QTSEPOOL to move interactive jobs to the base pool at timeslice end:
  - Verify why jobs are going to timeslice end
- More memory is always a good option but does not fix bad application design
- Power5 or Power5+ run reasonably well from 6 GB per processor on

Tricks of the Trade

- Consider Journal Caching to limit the impact of journal wait time
  - Install option 42 of the OS
  - Set the JRNCACHE to *YES on the CHGJRN command
  - Commitment Control already uses some journal caching
- Consider Concurrent Write to optimize DB update or write time
  - CALL QDBENCWT '1' to enable or '0' to disable
  - Requires an IPL to be activated

What's out there?

- Processors – can somebody define it?
- Memory – does it scale? How much do I need?
- Disk units – how do they behave?

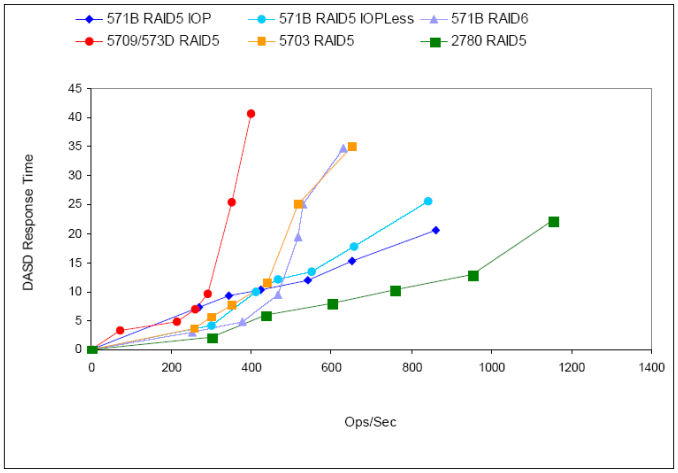
Topology

- Persistence of data flows from memory via I/O Processor to a disk controller (I/O Adapter) to a (set of) disk(s)
- Storage optimization done based on capacity used and number of I/Os per spindle
- Objects might be grouped for efficiency in ASPs, housed under a separate IOP/IOA
  - Journal receivers are prime candidates to do this if only deposits are done
  - Calculate efficiency of such a setup in the Component Report

Disk Configuration – All Disk Types

- Limit the number of disk controllers per bus – check with the rules for high capacity I/O adapters
- Disk capacity needs to be calculated based on:
  - Storage capacity needed:
    - Do not exceed 80% capacity utilization
    - Temporary space can grow rapidly
  - Number of I/Os expected:
    - Don't load disks for more than 35% busy
    - Large blocks or sequential ops will provide a good yield
- Disk utilization can be improved with ASP balancing or disk reorganization
- Faster disks are … faster
- Large memory on controllers or subsystems yields better performance but may cause data to be staged

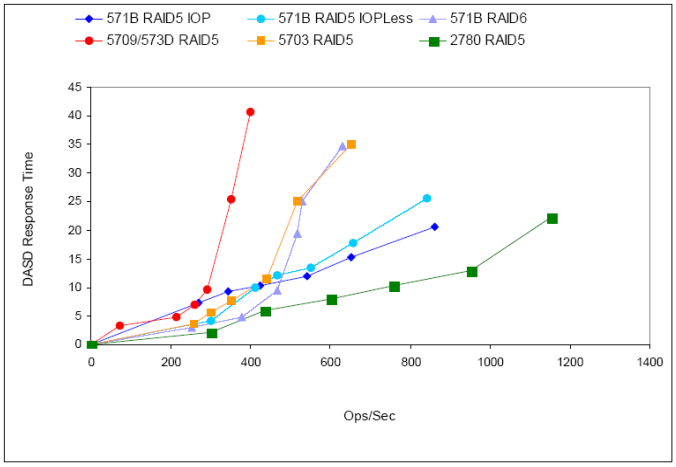
DASD Performance



5709/573D: 8 15 KRPM 35 GB – Others: 10 15 KRPM 35 GB

Performance, Sizing, Tips and Tools

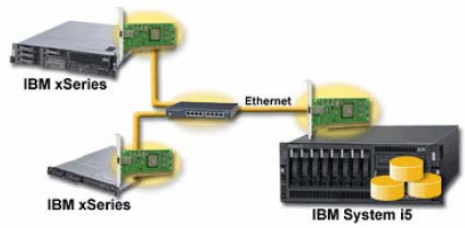
DASD Performance



5709/573D: 8 15 KRPM 35 GB – Others: 10 15 KRPM 35 GB

System i as a Storage Server

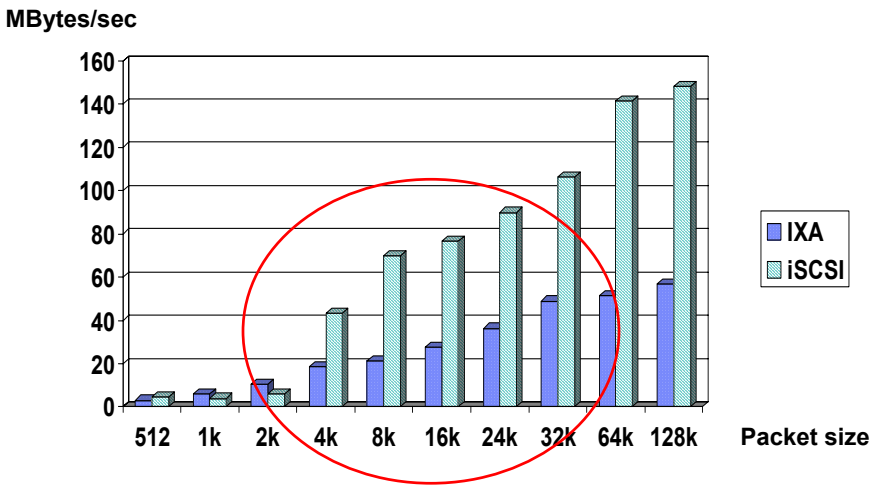
- Providing support to store storage associated with other Operating Systems, especially Microsoft or Linux based
- New kid on the block: iSCSI support for xSeries or Blade Center



iSCSI Support

- iSCSI support includes IXS and IXA capabilities:
  - All disks are virtual disks, enabling storage consolidation
  - Centralized server management
  - i5/OS tape, optical, and virtual Ethernet devices
- Extensive scalability range
  - Many System x-based servers through one iSCSI Host Bus Adapter (HBA) for a lower cost connectivity, up to a number of iSCSI HBAs per individual System x-based servers for scalable bandwidth
  - Each HBA can be shared by up to 8 integrated servers. In configurations with low to moderate virtual disk I/O rates this can result in a lower cost solution compared to an IXS or IXA option per integrated server.
- Capability to specify multiple virtual disk paths
- Lower incremental cost per integrated “server”
- Utilization of TCP/IP skills to install and configure
- Leveraging industry standards with iSCSI

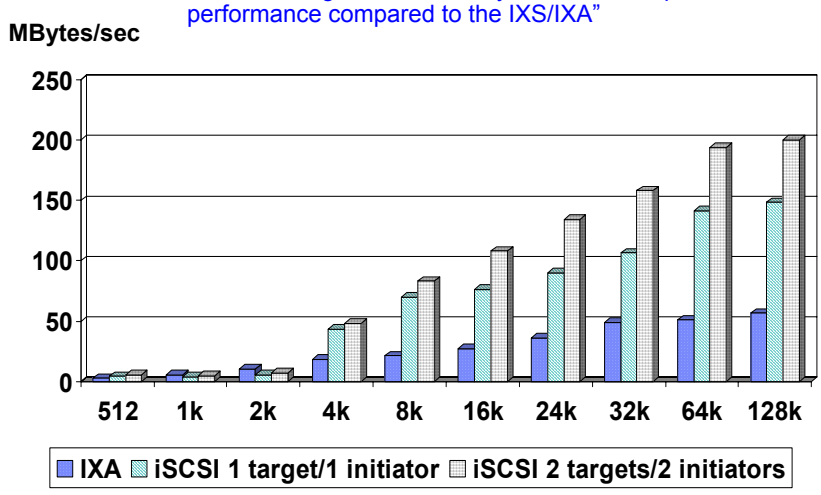
Performance Comparison



note: 4k block sizes and higher are representative of more I/O server scenarios.

Performance Scalability Heavy I/O

"the iSCSI solution offers greater scalability and overall improved performance compared to the IXS/IXA"



System i CPW Sizing

- V5R2 CPW - IXS/IXA
  - 10 CPW per 100 Disk Ops/sec
- V5R3 CPW - IXS/IXA
  - 12 CPW per 100 Disk Ops/sec
- V5R4 CPW
  - 13 CPW per 100 Disk Ops/sec - IXS/IXA
  - 19 CPW per 100 Disk Ops/sec - iSCSI

iSCSI example: Windows server doing 300 Ops/sec
- CPW = 300 * 19/100
- CPW = 57

Note: These rules of thumb are estimated from the results of performing file serving or application types of loads with various packet sizes. For more specific details see System i Capabilities Reference Guide

System i Memory Sizing

Memory Pool Type	For Each Active Target HBA	For each Active NWSD
Machine Pool:	21 MBytes	1 MByte
Base Pool:**	1 MBytes	0.5 MByte
QFPHIS Private Pool	0.5 MByte	1 MByte *
Total	22.5 MBytes	2.5 MBytes

Example: 6 System x doing typical workload
- 2 Target HBAs
- 6 NWSDs
- Total Additional Memory = 500** + (2 X 22.5) + (6 X 2.5) = 560 MBytes

* Private pool assigned to QFPHIS must still be a 4 MB minimum size
** IBM Director Server requires an additional 500MBytes in the base pool

System i iSCSI Target HBA Sizing

Each System i iSCSI Target HBA can do approximately 90 Mbytes/sec

- Based on approx 8K packet sizes
- Mixed workload of reads/writes

Collect data via Windows Performance monitor

- Physical Disk performance object
- Add up Disk Bytes/sec for all Windows server

Server	MB/sec
Server A	18.5
Server B	5
Server C	56.5
Server D	90
Server E	2
Total	172

- # targets = 172/85 = 2.02
- # targets = 2 + 1 for redundancy

General Rules of Thumb

- **3 System x or Blade servers per System i iSCSI Target HBA (recommend a 1 extra for redundancy)**
- **70 CPW per System x or Blade server**
- **500 MB + 20 MB per System x or Blade**

What's out there - Revisited?

- Applications and How to Make Them Behave
- Tools and Where They Are

Java using 32-bit or 64-bit Virtual Machines

- I5/OS V5R4 delivers two JVM and SDK environments:
  - Java TM 2 Software Development Kit, Standard Edition
    - "Classic", running in 64-bit mode
    - "32-bit" IBM Technology for JVM runs in 32-bit mode
- "32-bit" implements a standard IBM JVM 1.5 runtime environment and SDK:
  - Specifically focused for applications on smaller systems, with less impact on memory utilization
  - Simple implementation, mostly without code changes for the majority of the applications
  - Affects management of storage references
- Garbage Collection is similar to most other platforms for the 32-bit JVM, while it remains unchanged for the 64-bit JVM
- Requires less storage and heap size than the 64-bit version

Java Performance: JIT versus DE

- Starting with V5R3, JIT is always recommended for best performance of 32-bit (no DE) or 64-bit
- Direct Execution (DE) is the default, unless:
  - ADDENVVAR ENVVAR (JAVA_COMPILER) VALUE(jitc) JAVA CLASS(Test)
  - OR Create your Java programs under:
    - Either Set java.compiler system property on the i5/OS command line -> JAVA CLASS(Test) PROP((java.compiler jitc)) or
    - Set the java.compiler system property on the Qshell Interpreter command line. For example, enter java -Djava.compiler=jitc Test
- Even if running JIT, the JVM will have to create a Java program object (with optimization level *INTERPRET) the first time a particular Java class is used on the system, if one does not already exist.
  - Creation of this program object is much faster than creating a full DE program, but it may still take some time
  - Running CRTJVAPGM with OPTIMIZE(*INTERPRET) will create this program ahead of time, making the first startup faster

Analyzing JVM – Memory Needs

- To determine the approximate amount of memory required for a single JVM:
  1. Start the JVM.
  2. Ramp up the workload on the JVM.
  3. Perform the i5/OS Dump JVM (DMPJVM) command on the JVM and look at the output section labeled Garbage Collection.
  4. The general rule is that this JVM will require the amount of memory that is indicated by the following formula (where JIT stands for just in time): heap size + ((JIT heap size + JVM heap size)/2) ➡ 1887964 + ((272352 + 598204)/2) = **2.278.242** KB

```
.........................
. Garbage Collection
.........................
Garbage collector parameters
   Initial size: 786432 K
   Max size: 240000000 K
Current values
   Heap size: 1887964 K
   Garbage collections: 42
Additional values
   JIT heap size: 272352 K
   JVM heap size: 508204 K
```

Database Optimization

- Index structure:
  - Organize indexes to be created from *most complex* to *least complex*
  - Create indexes with access path size set to 1 TB
  - Consider the use of encoded vector indexes for fast retrieval if low cardinality of the key
  - Create the Perfect Index (become eligible for a Grand Prize!)
- Enable the Optimizer to make wise decisions:
  - Check if Statistics are being collected correctly
- Perform pro-active query tuning
- Consider Materialized Query Tables
- Use Symmetric Processing where possible
  - Fine-grained control (use properties file QAQQINI) versus coarse-grained control (using system values)
- Re-use deleted record space

What's out there - Revisited?

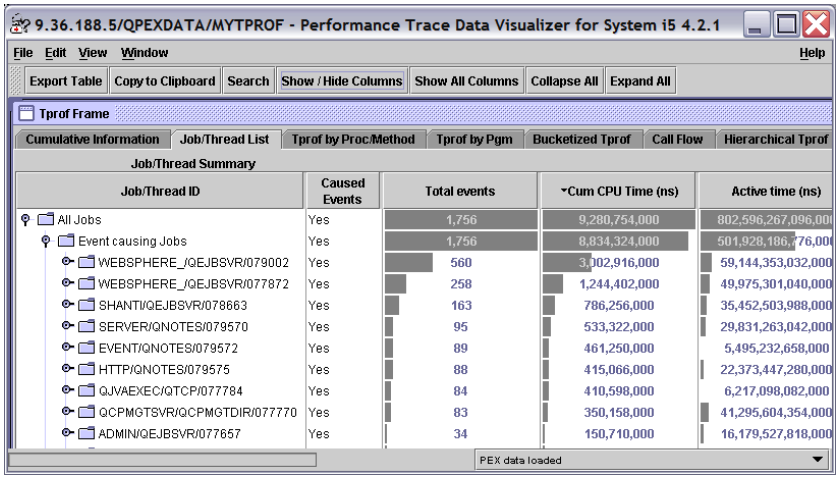
- Applications and How to Make Them Behave
- Tools and Where They Are

The Tools for Analysis

- Included in i5/OS:
  - Work with Active Jobs (WRKACTJOB) - Work with System Status (WRKSYSSTS) - Work with Disk Status (WRKDSKSTS)
  - Collection Services (STRPFRCOL)
  - Performance Trace (STRPFRTRC)
  - SQL Performance Monitors (STRDBMON)
  - iSeries Navigator equivalents and Monitors
- Included in the Performance Tools Licensed Product:
  - Work with System Activity (WRKSYSACT)
  - Performance Explorer
  - Reports for Collection and Trace functions
  - Performance Advisor
- Advanced Tooling:
  - iDoctor, Job Watcher, PEX Trace Data Visualizer
  - Web Performance Monitor and Advisor

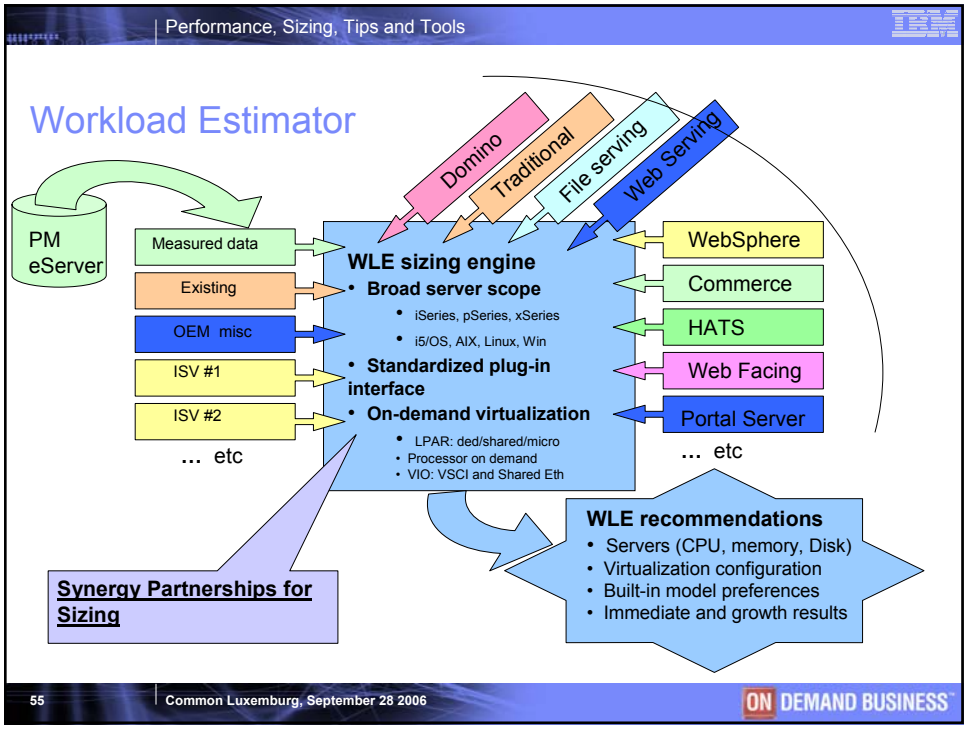
Job Watcher Sample

PEX Trace Data Visualizer



The Tools for Capacity Planning and Sizing

- Workload Estimator
  - See: http://www-912.ibm.com/estimator/index.html
- Performance Navigator
  - See: http://www.mpginc.com/
- BMC PATROL for iSeries – Predict
  - See: http://www.bmc.com/products/proddocview/0,2832,19052_19429_4091705_9687,00.html


Performance, Sizing, Tips and Tools
Workload Estimator
Domino
Traditional
File serving
Web Serving
PM eServer
Measured data
Existing
OEM misc
ISV #1
ISV #2
... etc
WLE sizing engine
• Broad server scope
• iSeries, pSeries, xSeries
• i5/OS, AIX, Linux, Win
• Standardized plug-in interface
• On-demand virtualization
• LPAR: ded/shared/micro
• Processor on demand
• VIO: VSCI and Shared Eth
WebSphere
Commerce
HATS
Web Facing
Portal Server
... etc
WLE recommendations
• Servers (CPU, memory, Disk)
• Virtualization configuration
• Built-in model preferences
• Immediate and growth results
Synergy Partnerships for Sizing
55
Common Luxemburg, September 28 2006
ON DEMAND BUSINESS


Performance, Sizing, Tips and Tools
Performance Navigator
56
Common Luxemburg, September 28 2006
ON DEMAND BUSINESS

BMC PATROL for iSeries - Predict