

# OS/390 and z/OS Workload Manager Goal Mode: Advanced Capabilities

**Stephen L. Samson**  
**Senior Technical Staff Member**  
**Candle Corporation**

*steve\_samson@candle.com*  
*steve.samson@attglobal.net*

**New York**  
**Computer Measurement Group**

**November 9, 2001**  
**New York, New York**

Trademarks and registered trademarks used in this presentation are the property of their respective owners and are to be regarded as appearing with the appropriate ™ or ® symbols at their first mention. Predictions and evaluative statements are the opinion of the author only and do not necessarily represent Candle Corporation positions.

© Copyright 2000-2001 Candle Corporation. Permission is granted to NYCMG to distribute images of the slides and notes of this presentation to attendees. All other rights reserved.

SLS 23OCT2001

# Topics

- Introduction
- CICS/IMS Server Management
- WLM-managed initiators
- Scheduling environments
- Application environments
- Advanced Server Address Space Management
- WLM-managed hardware resources
- A brief survey of enclaves
- Summary

# Introduction

- **Workload Manager has evolved to deal with some objections**
  - **new functions for OS/390 R10 and z/OS—not covered here**
- **WLM now controls many aspects of performance management**
  - **helping to balance some kinds of work across images in a sysplex**
  - **helping to achieve response time goals**
  - **responding to levels of business importance**
  - **protecting work from known sources of delay**
  - **starting and stopping server address spaces**
- **There's much more it can manage**
  - **hardware resources**
  - **new kinds of work unit organization**
  - **system affinities**
  - **batch job initiation**



---

# WLM-managed Software Resources

# Server Management in CICS and IMS

- There are two approaches to managing CICS or IMS “regions”:
  1. set region velocity goals and ignore transaction performance
    - ▲ this is *not* server management—regions are not seen as servers
    - ▲ velocity goals may be inferior to fixed DP and storage isolation
  2. set transaction goals and ignore region goals (which must still be set)
    - ▲ region will be on its velocity goal until it becomes a server
    - ▲ region resources will be managed so that transaction goals are met
    - ▲ this *is* server management
    - ▲ region reverts to velocity goal after ~20 minutes of inactivity
- A transaction service class affects regions more or less depending on:
  - whether it meets its goal—if so, not at all
  - residency—if it uses the region more of the time it will have more effect
  - importance—if goals are not met, the most important class is helped
  - degree of missing goal—at equal importance, the most pain is helped
- Therefore:
  - give more important classes difficult but not impossible goals
  - give less important work low importance and very easy goals

# CICS & IMS Server Management (continued)

- Getting to server management has been regarded as difficult
  - confusion about identifying transactions for classification
  - impression that transactions must be rearranged across regions
- It's not that hard!
  - start with a default service class and report classes
  - move rapidly to a more important service class for the *Loved Ones*
  - add more service classes as needed
  - key considerations: which goals are important to meet?
  - OS/390 R10 changes allow special case "opt-out" (e.g. test regions)
- Example:
  - CICS default: 70% complete in 1.0 seconds, importance 4
  - most-loved: 85% complete in 0.6 seconds, importance 2
  - loved: 80% complete in 0.8 seconds, importance 3
  - unloved: 10% complete in 10 hours, importance 5

(importance 1 is reserved for emergency or recovery work)

# Classifying CICS Address Spaces

Subsystem Type . . . . . : STC

Description . . . . . All started tasks

-----Qualifier-----		-----Class-----			Storage	Manage Regions
Type	Name	Start	Service	Report	Critical	to Goals of
DEFAULTS:						
1	TN	%MASTER%	DISC____ SYSTEM__	_____ MASTER__	_____ ____	_____ _____
... ..						
1	<b>SY</b>	HOTPROD	STCMED__	GENSTC__	<b>NO</b> __	_____ _____
2	TNG	HOTCICS	VEL60__	CICSREGS	<b>NO</b>	<b>TRANSACTION</b>
2	TN	CICSP03	TOPCICS	CICSREGS	<b>YES</b>	<b>REGION</b>
1	TNG	HOTCICS	CICS	CICSREGS	<b>NO</b>	<b>TRANSACTION</b>
1	TN	CICST*	TESTCICS	CICSREGS	<b>NO</b>	<b>REGION</b>
1	TN	%%%IRLM	SYSSTC__	IRLMS__	<b>YES</b>	_____ _____
1	TNG	DB2_____	VEL60__	DB2S__	<b>NO</b> __	_____ _____

**RED** denotes function added post-V2R10

# CICS Address Space Service Class

Service Class Name . . . . . : ORDCICS  
Description . . . . . CICS region  
Workload Name . . . . . CICS (name or ?)  
Base Resource Group . . . . . \_\_\_\_\_ (name or ?)  
**CPU Critical . . . . . NO**

---Period---	-----Goal-----		
#	Duration	Imp.	Description
1		2	Velocity=40%

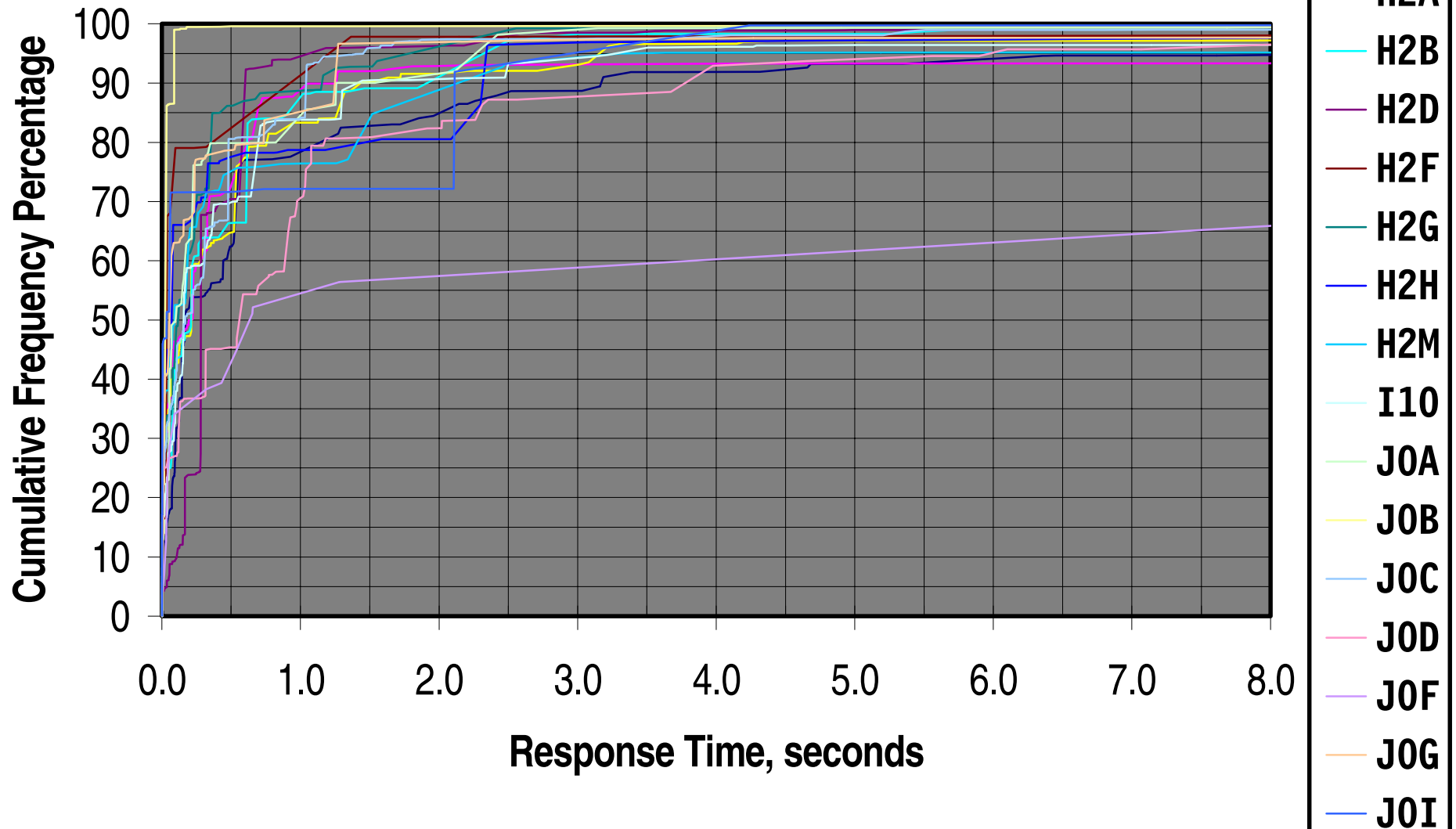
# CICS Address Space Service Class (HOT!)

Service Class Name . . . . . : TOPCICS  
 Description . . . . . Super-hot CICS region  
 Workload Name . . . . . CICS (name or ?)  
 Base Resource Group . . . . . \_\_\_\_\_ (name or ?)  
**CPU Critical . . . . . YES**

---Period---		-----Goal-----	
#	Duration	Imp.	Description
1		1	Velocity=60%

# Some Real Data

## CFDs of 95th Percentiles by ApplID



# CICS rules in a "Real" Service Definition

Default service class is CICSDFLT (P85S10I3)

Qualifier #	Qualifier type	Qualifier name	Starting position	Service Class
1	TNG	STARS		CICSLOVD (P85S10I2)
1	TNG	WORST		CICSUGLY (P10H2I5)
1	SIG	FASTEST		P90S07I3
2	TNG	EXCEPT1		P80S40I3
1	SIG	SLOWER		P85S15I3
1	SIG	SLOWEST		P80S40I3
1	SIG	WEIRD1		P75S20I3
1	SIG	WEIRD2		P55S30I3

# CICS Service Class (Red-hot)

Service Class Name . . . . . : TOPTRAN  
Description . . . . . Ultra-Time-critical CICS  
Workload Name . . . . . CICS (name or ?)  
Base Resource Group . . . . . \_\_\_\_\_ (name or ?)  
**CPU Critical . . . . . YES**

---Period---	-----Goal-----
# Duration	Imp. Description
1	1 85% complete within 00:00:01.000

# Service Class for a Bottom Feeder

Service Class Name . . . . . : UGLYSLUG  
Description . . . . . Conversational CICS  
Workload Name . . . . . CICS (name or ?)  
Base Resource Group . . . . . \_\_\_\_\_ (name or ?)  
**CPU Critical . . . . . NO**

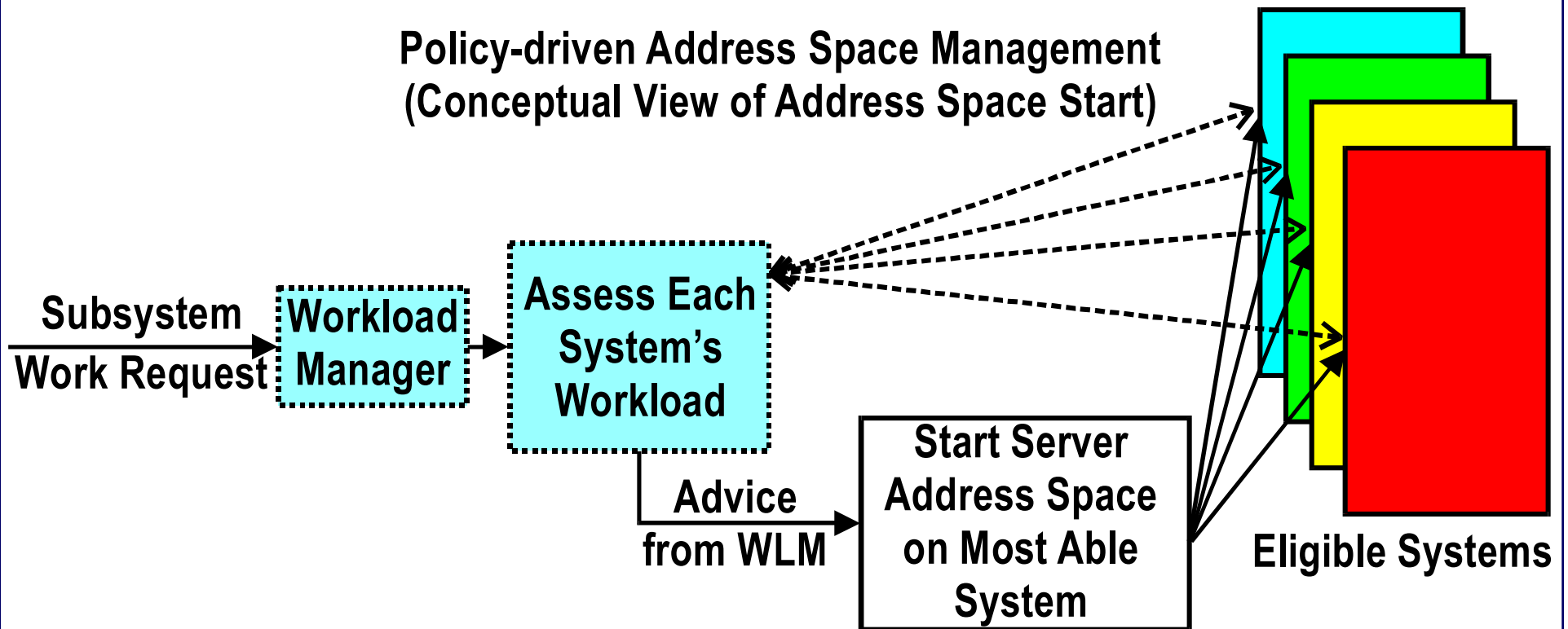
---Period---	-----Goal-----
# Duration	Imp. Description
1	5 10% complete within 10:00:00.000

# Enhanced Server Management

- In some subsystems WLM can start and stop server address spaces
- WLM and the work manager monitor the work in the servers
  - if work is not meeting its goal because it's waiting for a server...
  - and a system in the sysplex can afford the price of another server...
  - and the waiting unit of work is free to move (or there are more like it)...
  - and there is nothing stopping WLM from starting a new server—
  - then a new server is started on the most capable system
- When the servers “dry up” WLM will stop them
  - except that the last one left will linger longer
- Benefits—
  - reduced need for manual starting and stopping of servers
  - optimum number of active servers at any given time
  - minimum overhead when demand decreases
  - less tuning effort
  - performance management is reduced to setting goal and importance

# Adaptive Resource Management

Policy-driven Address Space Management  
(Conceptual View of Address Space Start)





---

**Now Imagine That We're  
Speaking of... Batch!**

# WLM-Managed Initiators

- Why it's needed
  - JES initiators are more or less statically defined
  - responding to workload changes requires operators to start, stop, or reset initiators based on perceived demand
  - batch service levels are at the mercy of these processes
  - managing non-production batch can't deal with time on job queue
    - ▲ but scheduled or production batch is usually closely managed
- How it works
  - individual job classes are designated in JES as WLM-managed
    - ▲ dynamic redesignation is possible—applies sysplex-wide
  - each such job class should be classified to a service class
    - ▲ each service class **must** serve only JES, or only WLM classes
      - ❖ not a syntax error but “unpredictable results” if mixed
  - job queue delay is added as a manageable wait reason
  - WLM starts and stops one-class initiators depending on need
    - ▲ initiators can be started on any system in the sysplex (uh-oh!)
    - ▲ but scheduling environments can restrict job placement

# WLM-Managed Initiators: Indications and Contraindications

- **WLM-managed is good for:**
  - randomly arriving batch
  - short-running jobs
  - high volume
  - well-defined service targets
  - minimal system affinities
- **WLM-managed initiators are *not* good for:**
  - time-released scheduling of many jobs
  - heavy system or inter-job dependencies/affinities
    - ▲ but see scheduling environments to get around this problem
  - long-running jobs and jobs with highly variable run times
  - small volume or sporadic arrival rates
  - lack of service targets
- **However—**  
many have made all initiators WLM-managed with good results



---

# Scheduling Environments

# Scheduling Environments

- **Current affinity mechanisms are difficult and inflexible to use**
- **WLM-managed initiators need a complementary WLM-managed affinity scheduling facility to reflect today's environment**
- **Facility is limited to batch only—today**
  - **other subsystems might eventually use the scheduling environments or underlying resources as well**
  - **WLM provides internal and external interfaces to resources and environments**

# Scheduling Environments— How they Work

- The WLM ISPF application is used to define SCHENVs and resources
- Resources and scheduling environments are defined sysplex-wide
- Resources are set *on each system* to ON, OFF, or RESET state
  - RESET condition is set at IPL or policy activation if resource is new
  - operator commands or WLM internal service requests may be used to set resource states
- A scheduling environment may require a resource to be ON or OFF
- A scheduling environment is *available* or *not available* on each system
  - the state on each system is determined by the states of its required resources on that system
- Automation is key tool in managing resource states and SCHENVs
  - monitor the underlying conditions to trigger state changes
  - issue the commands or system calls to set the states
  - check for anomalies and correct them

# Scheduling Environments—Example

- Suppose you want to control the ability of IMS BMPs to run on systems on which a particular IMS control region is active
- In the JOB statement for each BMP, specify SCHENV=IMSCRUP
  - the job may run on any system on which IMSCRUP is *available*
- Define IMSCRUP as:

Scheduling Environment Name IMSCRUP \_\_\_\_\_

Description . . . . . SCHENV to allow BMP Scheduling

Required

Resource Name	State	Resource Description
CREGUP	ON _____	IMS Control Region is up

- The resource CREGUP comes up as RESET at IPL
- Use automation to set CREGUP to ON when IMS issues an “Initialization Complete” message from a named CREG, using the internal interface or:

**F WLM, RESOURCE=CREGUP, ON**

# Other Users of Server Management

- **WLM-managed initiators are a very basic user of server management**
- **There are other, more sophisticated users—**
  - **DB2 stored procedures**
    - ▲ **can be very effective performance enhancement for standard queries**
    - ▲ **enclaves waiting for service can trigger server address space start**
  - **Scalable Web Server (IHS or as part of Websphere)**
    - ▲ **extra address spaces can be started when transactions are queued**
    - ▲ **transactions are typically HTTP requests**
    - ▲ **when peak is past, extra servers are stopped after some inactivity**
- **This model applies well to any volatile workload**
- **The WLM functions are open and documented—anyone can use them**

# Application Environments

An application environment is a template that allows WLM to start server address spaces based on subsystem-specified criteria

Application - Environment	Notes	Options	Help
---------------------------	-------	---------	------

-----  
Modify and Application Environment

Command===> \_\_\_\_\_

Application Environment . . .	WEBHTML_____	(Required)
Description . . . . .	HTTP test environment_____	
Subsystem Type. . . . .	IWEB (Required)	
Procedure Name. . . . .	IMWIWM	
Start Parameter . . . . .	IMWSN=&IWMSSNM, IWMAE=WEBHTML	

\_\_\_\_\_  
\_\_\_\_\_

Limit on starting server address spaces for a subsystem instance:

1. No limit
2. Single address space per system
3. Single address space per sysplex

# Uses of Application Environments

- **Work managers that support application environments**
  - **DB2 (for stored procedures)**
    - ▲ mapped in DB2 SYSIBM.SYSROUTINES catalog table
  - **Component Broker (as part of Websphere Application Server V4)**
    - ▲ mapped in server group name
  - **IBM HTTP Server**
    - ▲ mapped in Web Configuration File
  - **MQ Series Workflow**
    - ▲ mapped in MQ process definition for WLM-managed queue
- **Address spaces are started when enclaves' service classes miss goals**
  - **WLM stops idle address spaces after a period of time**
  - **WLM stops the last idle address space in a subsystem after longer interval of time**



---

# A Brief Survey of Enclaves

# What is an Enclave?

- An enclave is an independent dispatchable unit of work
  - has an external name and a place on the dispatch queue
- Represents a “business unit of work”
- Is managed separately from the address space in which it resides
- Can include multiple SRBs and TCBs
  - SRBs are preemptible type
  - all enclave CPU is counted as if TCB
- Can span multiple address spaces
- Can have many enclaves in a single address space
- Exists in both goal and compatibility mode
  - assigned to service class or PGN

# How are Enclaves Managed?

- Enclaves are created using system service requests
- They are classified to service classes by the WLM service definition
  - *dependent* enclaves inherit from the originating address space
    - ▲ e.g. DB2 stored procedures and query parallelism
  - *independent* enclaves are classified on their own
    - ▲ e.g. DDF and HTTP transactions
- Management differs in goal and compatibility mode:
  - if in goal mode, service class is managed to goal
  - in compat mode, service class provides a tag to be picked up in ICS to assign to a performance group
- Consequences of not classifying differ too:
  - in goal mode, enclave goes to SYSOTHER
  - in compat mode, enclave inherits PGN attributes from address space
- Accounting hits several SMF records
  - type 72 for service class and report class (txn counts, resource use)
  - type 30 of the owning address space (txn counts, resource use)
  - type 89 (usage-based pricing) of home address space

# How do Subsystems Use Enclaves?

- CB: for all Component Broker Object Method requests (CB is now part of Websphere Application Server)
  - **CICS: not at all. Pity.**
  - DB2: for transactions split for query parallelism, for stored procedures
  - DDF: for all DDF transactions
  - IWEB: for requests handled by:
    - ICSS, Domino Go, or IBM HTTP Server (IHS)
    - Secure Sockets Layer
    - Fast Response Cache Accelerator
  - LSFM: for all LAN Server requests
  - MQ: for MQ Workflow requests including new client-server requests, activity executions, activity responses, and sub-process requests
  - **SMS: nobody home—another pity, especially HSM**
  - SOM: All SOM client object class binding requests (SOM out in z/OS 1.2)
  - Batch, OMVS, TSO, and STCs: applications may create enclaves
    - these are classified in the originating subsystem
- More uses of enclaves are very likely in the future

# What is the Significance of Enclaves?

- Much of e-business arrives from DDF, IWEB, and MQ subsystems
  - These three subsystems are major users of enclaves
- DB2 Query Parallelism (another user of enclaves) is a principal means of implementing data mining and other business intelligence and decision support applications
- Benefits:
  - a subsystem's work units can be managed according to business goals and priorities—no need to settle for an overall compromise
  - high volume work can use multiple server address spaces as needed with no program changes
  - low volume work can reside in a single address space
- Therefore:
  - good enclave performance is required to support e-business and other advanced OS/390 applications
  - measurement and management of enclaves is an essential part of OS/390 performance management

# For More Information on Enclaves

- **MVS Planning: Workload Management (GC28-1761)**—a “must have”
- **MVS Programming: Workload Management Services (GC28-1773)**
  - chapter on Enclaves includes a chart showing how accounting is done
  - chapter on Queueing Manager Services shows DB2 model
- **DB2 for OS/390 V5 Administration Guide (SC26-8957)**—DB2 use of enclaves
- **WLM home page: *<http://www.s390.ibm.com/wlm>***
  - For latest papers, presentations, and frequently asked questions



---

# WLM-managed Hardware Resources

# WLM-managed I/O Resources

- I/O Priority Queuing has been in all systems since OS/360
  - could only sort the IOS queue in the past—not much
- New hardware creates new options
  - enhanced DEFINE EXTENT provides a “priority” field in its data
    - ▲ ESS (Shark) implements this enhancement (others too, now or soon)
    - ▲ WLM can set the priority on each I/O driven by goal attainment (PI) and the extent of I/O delay for the service class, and importance
    - ▲ the priority is used to reorder device’s pending backstore requests
  - Shark can also support Parallel Access Volumes
    - ▲ each defined device can have static and dynamic alias UCBs
    - ▲ WLM manages dynamic aliases
    - ▲ dynamic aliases are moved among devices, also depending on goal attainment, extent of I/O delay for the service class, and importance
  - in z/OS on zSeries 900, WLM also manages Dynamic Channel Paths and Channel Subsystem Priority Queuing
    - 👉 The I/O subsystem becomes [more] self-tuning with WLM help

# WLM-Managed CPU Resources

- **LPAR Clusters and Intelligent Resource Direction in z/OS on z/Arch**
  - requires multiple LPARs on a single zSeries CPC in a parallel sysplex
  - all CPUs are shared
  - LPARs are explicitly defined as WLM-managed
  - all systems must be in goal mode (no problem as of z/OS 1.3!)
- **WLM will act to reduce CPU delay across the sysplex if goals are missed**
  - existing WLM actions within each LPAR are tried first
  - if an LPAR has excess CPU capacity and another suffers CPU delay
    - ▲ WLM can alter LPAR weights to adjust resources
    - ▲ if necessary, WLM can move a CPU from one LPAR to another
- **Implication**
  - paradigm shift: the resources can move to the work within a CPC
    - ▲ the work still moves to the resources between CPCs
  - WLM interactions with License Manager may be... interesting
- **Future possibility**
  - CPUs today, storage tomorrow!
- **In combination with I/O resource management, a single big z-box can become a self-tuning, self-optimized sysplex server**

# Information Sources

- Read the manual!
  - SA22-7602 (-02 for z/OS V1R2) *MVS Planning: Workload Management*
  - download as a PDF from:  
<http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/r2pdf/mvs.html>
  - read the Redbooks!
  - check the website (<http://www.redbooks.ibm.com/>) frequently
- Go to conferences! (OS/390 EXPO, SHARE, CMG)
- Browse the WLM website:  
<http://www.s390.ibm.com/products/wlm/>
  - if you use the OS/390 Web Server, see the FAQ link to a discussion of classifying IWEB enclaves
- Check Cheryl Watson at <http://www.watsonwalker.com>
- Get on the discussion groups:
  - IBM-MAIN (free newsgroup *bit.listserv.ibm-main* or mailing list)
  - [search390.com](http://search390.com)
  - MXG-L (free mailing list)

# Summary

- Goal mode opens the way to advanced WLM features
- Existing capabilities like enclaves are better controlled in goal mode
- Automatic server address space management—only in goal mode
- WLM-managed initiators automate operator-intensive activities
- Scheduling Environments can run in compatibility or goal mode
  - but WLM-managed initiators require goal mode
- WLM can manage dynamic properties of new hardware (z/OS on Z900)
  - CPU delay in a one-box sysplex (LPAR CPU management)
  - Dynamic Channel Path Management
  - Channel Subsystem Priority Queuing
- There's much more to come...
  - more delays to be managed by WLM
  - more hardware to be managed dynamically
- ... but it will all be dependent on running in goal mode