

The Basics of GRS: An overview of GRS ENQ processing

Speaker Name: Nick Matsakis Matsakis@us.ibm.com
(presentation by GRS Development Team)
Speaker Company: IBM Corporation

Date of Presentation: August 4, 2010

Session: 7552



SHARE in Boston

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business (logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Permission is granted to SHARE Inc. to copy, reproduce or republish this document in whole or in part for SHARE activities only.

Topics

- GRS concepts/configurations
- GRS programming interfaces
- Getting started
- Tuning
- Operational interfaces
- Questions

Lots of material to cover... STAR mode will be given preference over RING if time gets tight

Terminology

- **Serialize** - v. To keep something from changing by others, to make things happen in an order
- **Serially reusable resource** – n. a resource which can be used multiple times but not necessarily simultaneously
- **Enqueue** - v. To add something to a queue
- **ENQ** - v. To request control of a serially reusable resource, being added to a wait queue if the resource is not immediately available. The request is made to GRS using the ENQ or ISGENQ macros.
- **ENQ** - n. (1) Act of ENQing; (2) Control of a resource gotten by means of an ENQ(1); (3) The part of GRS which services ENQ requests.
- **ENQ** - adj. Pertaining to or related to ENQing or ENQs

Terminology

- **Reserve** – v. Set aside for use by a particular entity
- **RESERVE** – n. an I/O channel command that requests exclusive access to a device
- **RESERVE** – v. To issue a RESERVE channel command to request exclusive use of a device for the purpose of serializing access to the data held on the device
- **RESERVE** – n. The exclusive reservation of a device obtained by the use of a RESERVE channel command. In z/OS it also involves an ENQ on the requesting system.
- **RESERVE** – n. The z/OS macro for requesting a RESERVE. It is a form of ENQ which also issues a RESERVE channel command to the target device.

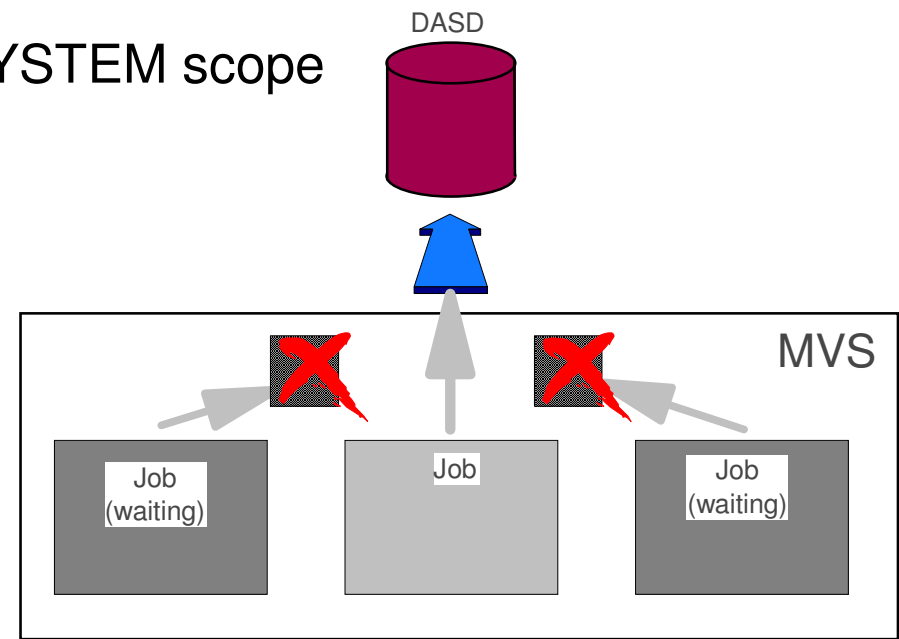
GRS – RESERVE and ENQ Services



- GRS is always there and is used regardless of the configuration
- APIs: ENQ,DEQ,Reserve,ISGENQ,GQSCAN, ISGQUERY,ISGADMIN
- Resource Identity: QNAME, RNAME, SCOPE
- Scope: JOB STEP,SYSTEM, GRS Complex
- Shared/Exclusive ownership
- Authorized/Unauthorized
- Widely used
- Reasonable performance
- Installation controls – RNLs and Exits

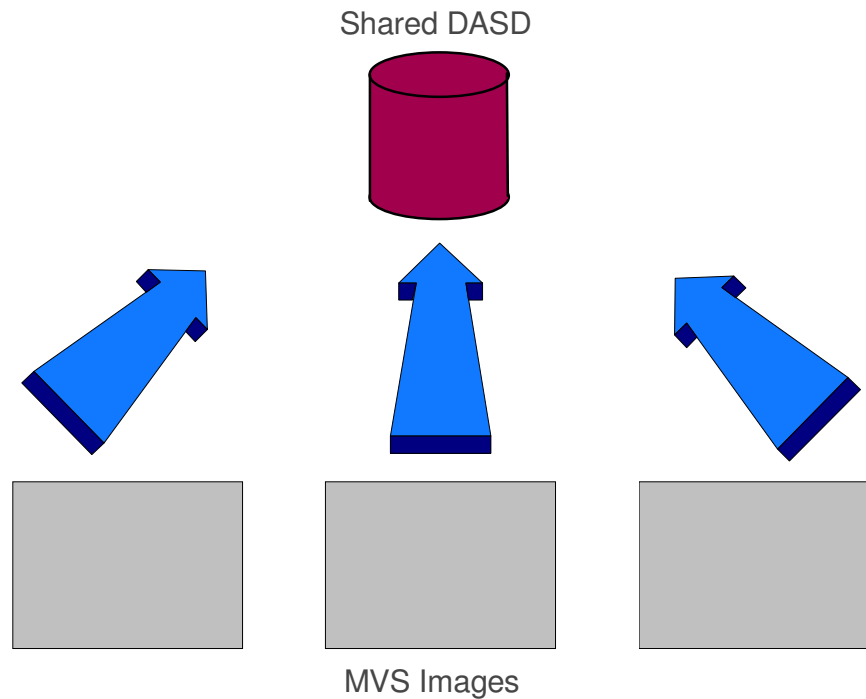
Local Resource Sharing

- Resource allocation within a single system
- z/OS APIs are:
 - ENQ/DEQ/ISGENQ
 - Local resources are Scope=STEP or SYSTEM
 - RESERVE
 - The ENQ was demoted to SYSTEM scope



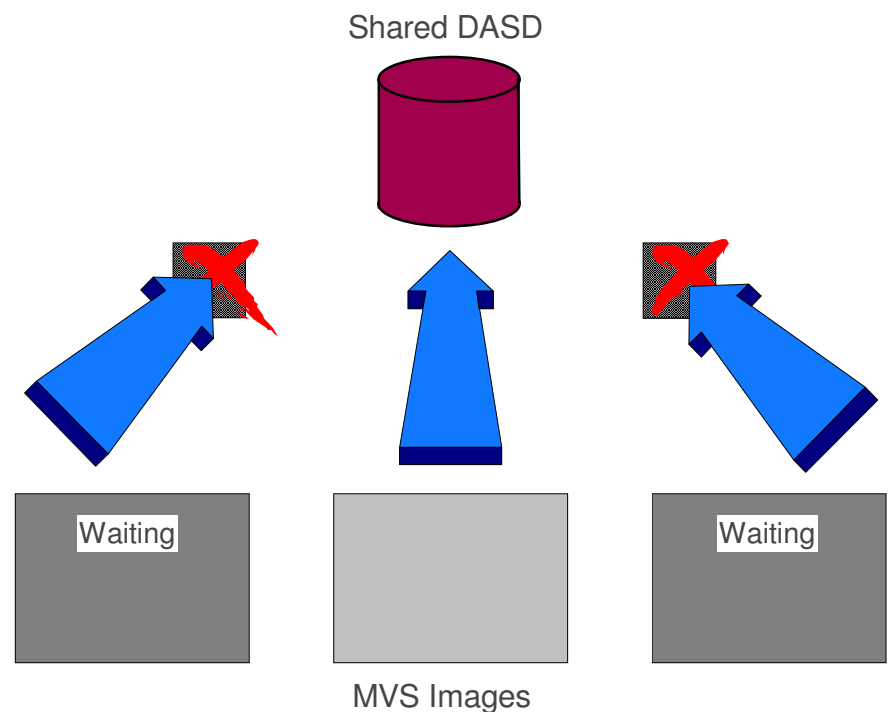
Global Resource Sharing

- Concurrent/uncontrolled resource allocation
 - Data consistency errors
 - Data integrity errors
- Means of serializing:
 - HW/Device RESERVE
 - Abstract software “lock”
 - GRS ENQ
 - XES lock
 - Distributed protocol
 - RYO communications locking protocol etc..



Global Resource Sharing: Hardware RESERVE

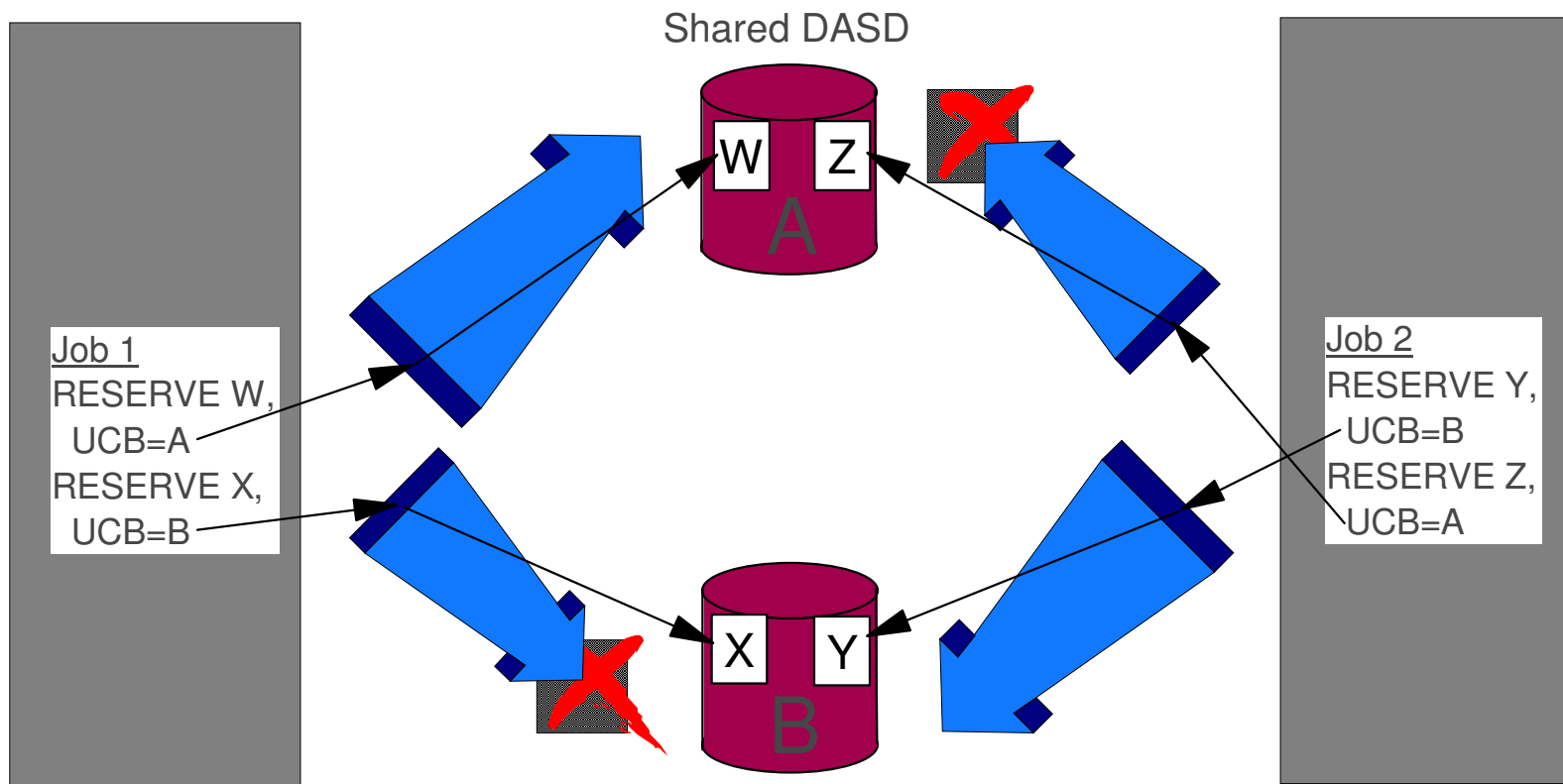
- Resource allocation controlled by device (via control unit)
 - Device only communicates with "owning" system
 - Serialization on the owning system is managed via a SYSTEM level ENQ.
 - Other systems wait, even for read only access
 - RESERVE ends when last job on owning system releases the resource



Global Resource Sharing: Hardware RESERVE - disadvantages

- Locks whole device
 - can cause performance/deadlock issues
- Does not distinguish between read and write access
- Starvation – no fair sharing!
- Deadlock possible
 - aka "Deadly Embrace" due to
 - configuration errors – hard to manage
 - due to non-synchronous reserves – hidden gotcha..

Global Resource Sharing: Hardware RESERVE - deadlock



ENQ Advantages Over RESERVE

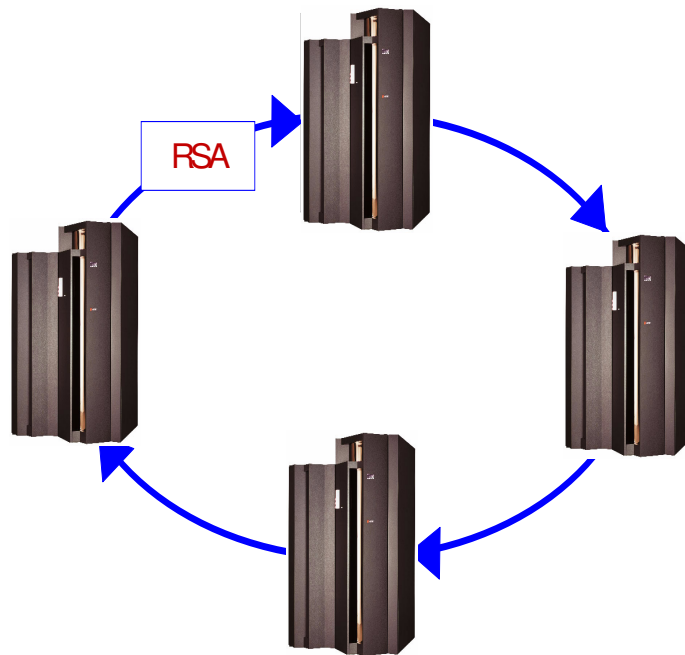
- Flexibility
 - Can serialize more than just DASD
- Granularity
 - Serialize by name over multiple scopes:
 - STEP (local address space)
 - SYSTEM (across one z/OS image)
 - SYSTEMS (across GRS complex)
 - Shared or Exclusive access
- Fairness (FIFO)
 - Ensures no starvation

ENQ Disadvantages compared to RESERVE

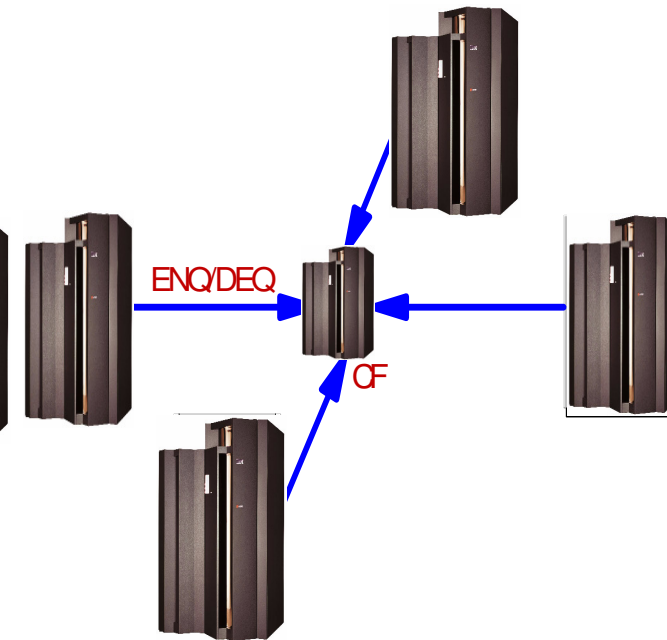
- Introduces MVS overhead
 - GRS STAR provides better performance than RESERVE
 - We recommend converting all possible RESERVEs in STAR mode
- Requires additional hardware... one of the following
 - Parallel sysplex (GRS STAR)
 - Requires sysplex (Best Ring Environment)
 - GRS managed CTCs (Basic Ring Environment)
- Requires additional setup
 - GRSCNFxx, GRSRNLxx

GRS Complex–SYSTEMS/GLOBAL ENQs

GRS Ring



GRS Star



Global Resource Serialization

GRS Ring

- Peer Coupling
- Connected via XCF and/or GRS managed CTCs
- Time slicing via RSA
- Global data view
- Can bridge between
 - a single multi-system sysplex and Monoplexes or XCF local system
 - Monoplex /XCF local systems
- Performance proportional to number of ENQs and number of systems
 - Works fine in many shops
- Functionally stabilized

GRS Star

- Peer Coupling
- Connected via CF Lock Structure
 - XCF signaling for contention management
- No time slicing
- Local data view
 - Global view is provided on request
- All systems must be part of the same sysplex
- Significant performance benefits
 - Best performer .. scales very well
- Target for global sharing enhancements

ENQs – Resource Identity

- ENQ “resource identity” is determined from the:
 - SCOPE (STEP, SYSTEM or SYSTEMS)
 - QNAME (Queue or Major name)
 - RNAME (Resource or Minor name)
- The attributes might be changed by an exit or by RNL processing
- The UCB on RESERVE specifies the device to reserve after the ENQ has been obtained

Programming Interfaces

GRS ENQ – API History

- **ENQ, DEQ, RESERVE**

- ENQ/DEQ: Obtain an abstract resource (SCOPE, QNAME, RNAME, and DISPOSITION)
- RESERVE: Obtain an abstract resource and also do HW reserve on a volume (UCB)
- Originally only an SVC interfaces which limits their usage
- z/OS V1R2 introduced support for cross-memory mode requesters

- **ISGENQ and ISGQUERY**

- Introduced in z/OS V1R6.
- These interfaces support AMODE 64, AR mode, and more...

ENQ/ISGENQ

Request Control of a Serially Reusable Resource



- Scope
 - STEP (address space uniqueness)
 - SYSTEM (MVS system uniqueness)
 - SYSTEMS (GRS complex uniqueness)
- Resource name
 - QNAME (or Major Name)
 - RNAME (or Minor Name)
- Access
 - Shared
 - Exclusive
- ISGENQ returns an ENQTOKEN

RESERVE/ISGENQ

Reserve a Device (Shared DASD)

- Scope
 - SYSTEMS (ENQ has complex uniqueness)
- Resource name
 - QNAME (or Major Name)
 - RNAME (or Minor Name)
- Access (associated ENQ)
 - Shared
 - Exclusive
- UCB
 - Volume to be RESERVEed
- ISGENQ returns an ENQTOKEN

Release a Serially Reusable Resource

- Releases an ENQ or RESERVE
 - ISGENQ REQUEST=RELEASE
 - ENQTOKEN returned on ISGENQ REQUEST=OBTAIN
 - DEQ
 - Scope
 - *STEP* (address space uniqueness)
 - *SYSTEM* (MVS system uniqueness)
 - *SYSTEMS* (sysplex uniqueness)
 - Resource name
 - *QNAME* (or Major Name)
 - *RNAME* (or Minor Name)

GQSCAN/ISGQUERY

Extract Information from GRS

- Returns resource allocation information
- Query by
 - Scope
 - Resource Name
 - Number of
 - Requesters
 - Owners
 - Waiters
 - RESERVE/ENQ
 - More...

Getting Started

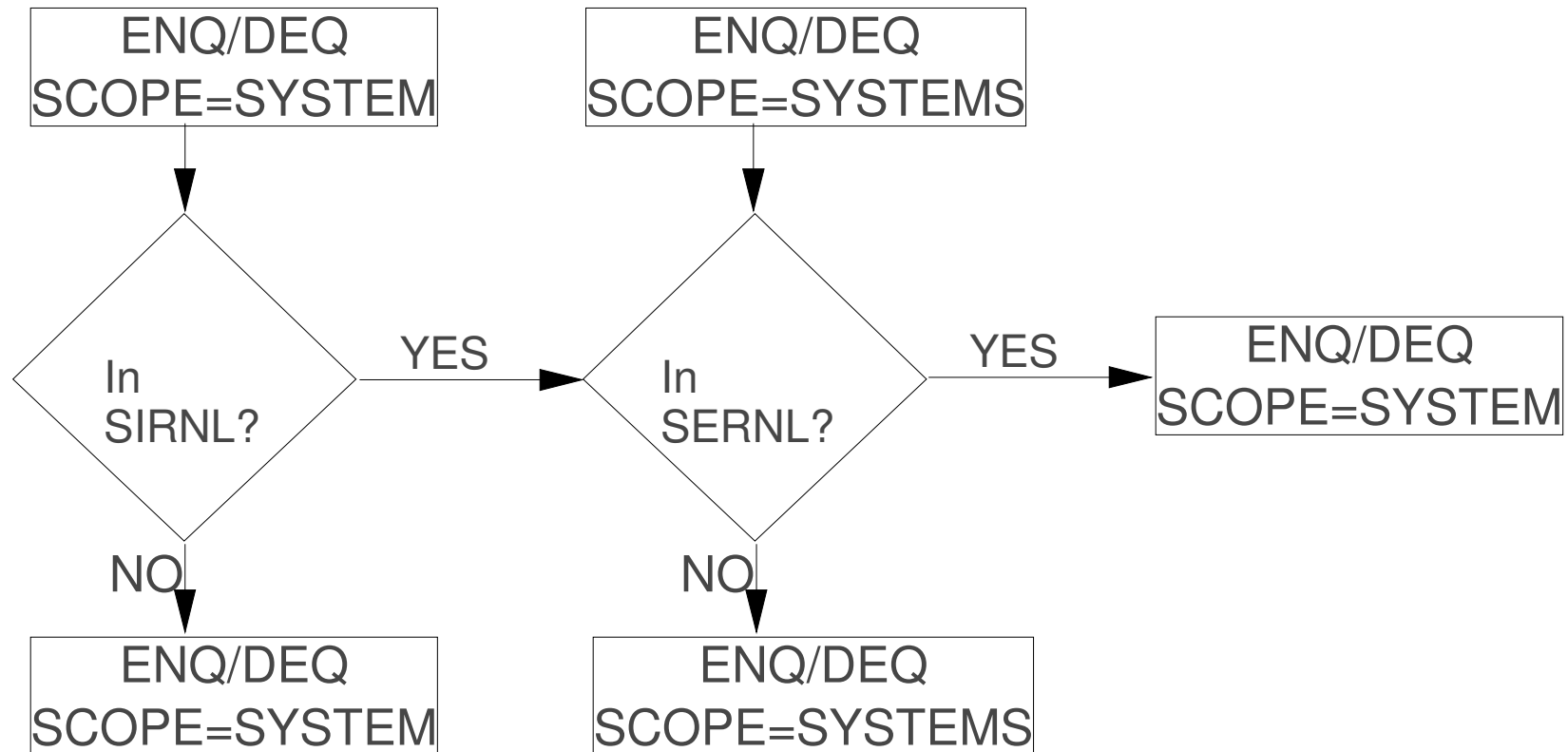
Create GRS complex

- Read
 - z/OS MVS: Setting up a Sysplex (SA22-7625)
 - z/OS MVS Planning: Global Resource Serialization (SA22-7600)
 - z/OS Planned Outage Avoidance Checklist (SG24-7328-00)
 - Merging Systems into a Sysplex (SG24-6818-00)
- Choose complex type (IBM recommends GRS Star)
- Understand and define RNLs (Resource Name Lists)
- Use the GRS ENQ/DEQ monitor to help determine your RNLs
- Update PARMLIB
 - IEASYSxx, GRSCNFxx, GRSRNLxx

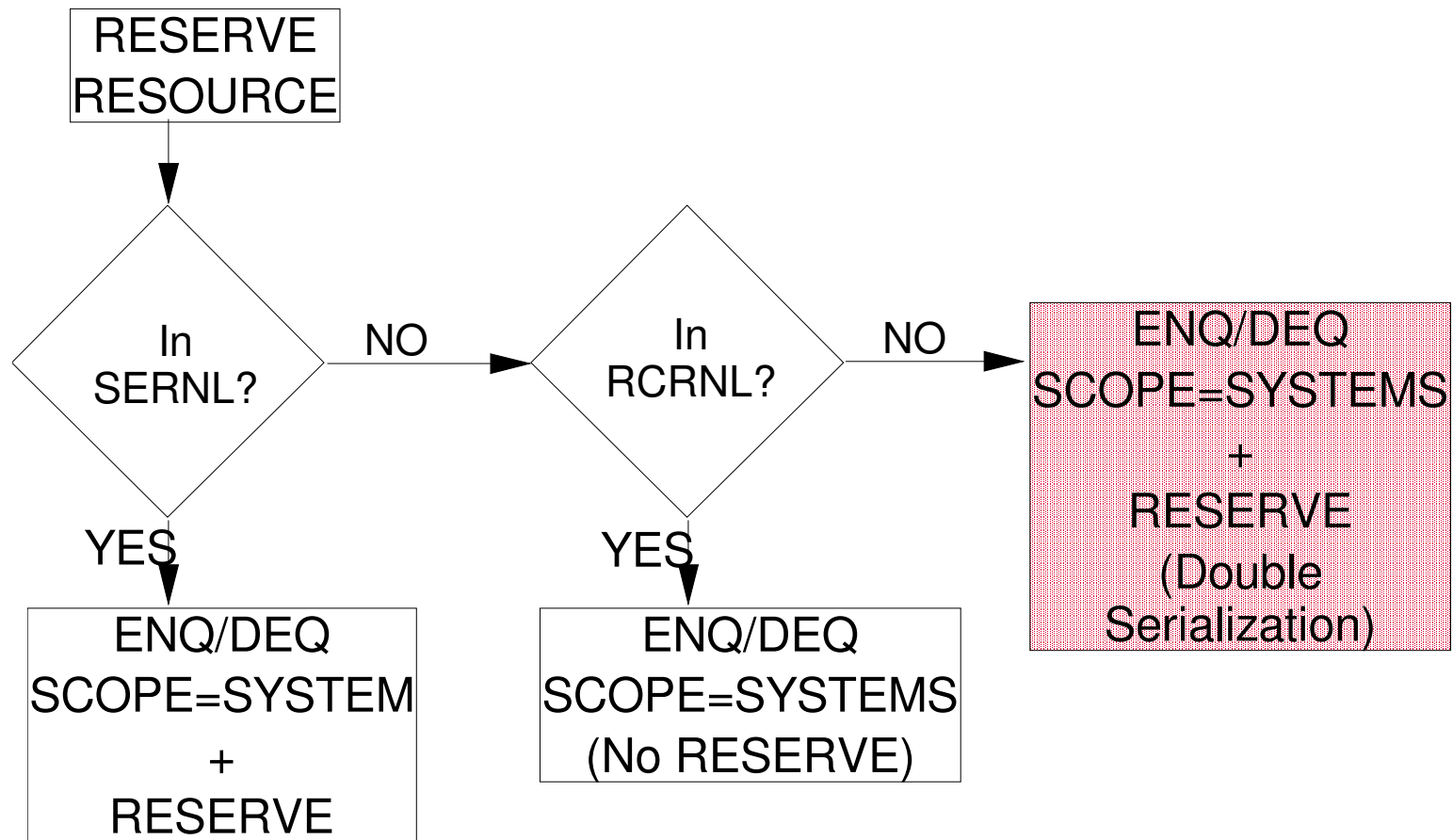
ENQs – Changing what they get

- The original “resource identity” can be changed by the installation or a third party:
- **GRS Resource Name Lists (RNLs)**
 - Inclusion – promote SYSTEM to SYSTEMS
 - Exclusion – demote SYSTEMS to SYSTEM
 - Conversion – convert reserve to ENQ only
- **GRS Installation exits:**
 - ISGNQXIT/ISGNQXITFAST
 - OEM oriented exits:
 - ISGNQXITPREBATCH, ISGNQXITBATCHCND, ISGNQXITBATCH, ISGNQXITQUEUED1, ISGNQXITQUEUED2, ISGENDOFLQCB
 - See white paper: “GRS/MIM Installation Exit Performance”

ENQ/DEQ RNL Processing



RESERVE RNL Processing



RESERVE RNL Processing

- A RESERVE that does not appear in either the SERNL or RCRNL will be serialized by both methods.
 - ENQ SCOPE=SYSTEMS
 - Hardware RESERVE
- This may cause deadlocks that are difficult to debug!



RNL Processing : GR SRNL=EXCLUDE

- Used to defeat global GRS processing
 - Many use it when using alternative serialization
 - GRS only processes specific ENQs globally
 - RNL=NO
 - Using an exclusion RNL to exclude all SYSTEMS ENQs may be a better choice
- Dynamic change via SET GR SRNL requires only 1 system left in the sysplex or sysplex wide re-IPL
 - Other restrictions apply, see GRS Planning Guide



Getting Started

Parms Setup

PARMLIB: IEASYSxx

GRS= **NONE**
 JOIN
 TRYJOIN
 START
 STAR

- GRS=NONE to run without GRS global processing capability
- GRS=TRYJOIN recommended for GRS Ring when complex=sysplex (basic sysplex)
- JOIN, TRYJOIN, START are for a Ring complex
- STAR is for a Star complex (parallel sysplex)
- Use SETGRS MODE=STAR to migrate from RING mode to STAR mode, when ready

Synchronous RESERVE

SYNCHRES(YES|NO)

- SYNCHronous REServe:
 - When SYNCHRES=NO, the RESERVE I/O is 'pre-pended' to the first I/O to the device
 - Can be a delay between obtaining the RESERVE and doing the I/O
 - The API user needs to know this! Some don't and get burned.
 - When SYNCHRES=YES, the RESERVE I/O is generated immediately upon owning the ENQ resource. The requestor is granted control after the device is RESERVED to the system
- GRS Ring, Star or None
- Default: YES starting with z/OS R1V6
 - GRSCNFxx SYNCHRES(YES) for prior releases
 - SETGRS SYNCHRES=YES/NO

ENQ Limits

ENQMAXA(value)

ENQMAXU(value)

- These set the system-wide authorized and unauthorized concurrent request limits for an address space.
 - Note GQSCAN and ISGQUERY requests that need to be resumed count against the unauthorized limit.
- GRS Ring, Star or None
- System Parameter in GRSCNFxx
- Can modify with SETGRS command
 - SETGRS ENQMAXA=
 - SETGRS ENQMAXU=

Specifying GRS data to be include in dumps



GRSQ(ALL|CONTENTION|LOCAL)

- GRS Star complex only
- Indicates the amount of data to gather in the event of a SYSMDUMP or SVC dump that specifies the SDATA=GRSQ option
- ALL – Gathers all local and global resources.
- **CONTENTION** – Gathers all local resources, and all global resources that are in contention.
- LOCAL – Gathers only resources on the local system.

Setting up a Star Complex

Setting up a Star Complex

- Create lock structure in CF policy
 - ISGLOCK
 - Backup/Rebuild considerations
 - No CF duplexing
 - CFSizer to calculate appropriate size
 - Fail over considerations
 - Monitor GRS lock structure usage to fine tune
- Create GRS record on Sysplex Couple Dataset
 - IXCL1DSU utility
 - DEFINEDS
 - DATA TYPE(SYSPLEX)*
 - ITEM NAME (GRS) NUMBER(1)*
 - Insure GRSCNFxx specifies GRSQ(CONTENTION)
 - Consider converting RESERVEs for ENQ processing

Setting up a Ring Complex

Ring RSA Residency Time

RESMIL(number|OFF)

- GRS Ring complex only
- RESidency time in MILLiseconds
 - How long GRS holds the RSA before passing it forward
- Default: 10 milliseconds
 - IBM recommends starting with a lower value (e.g. 1 - 5)
- GRS tunes RESMIL based on complex-wide GRS utilization
- OFF indicates that no tuning is to occur
 - Residency always limited to zero
- System Parameter in GRSCNFxx
- Can modify with SETGRS command
 - SETGRS RESMIL=

Ring Delay Toleration Interval

TOLINT(number)

- GRS Ring complex only
- TOLeration time INTerval in seconds
 - How long GRS will wait for the RSA to arrive before triggering an error condition
- Default: 180 seconds
- System Parameter in GRSCNFxx
- Can modify with SETGRS command
 - SETGRS TOLINT=

Ring Acceleration

ACCELSYS(number)

- GRS Ring complex only
- Ring ACCELeration number of SYStems
 - How many systems must see an ENQ before it can be granted
 - How many systems must fail before there is a possible integrity exposure
- Default: (none)
- Range: 2-99
 - If ACCELSYS greater than number of systems, there is no ring acceleration
 - If the nth system can not communicate back to the originating system then there is no ring acceleration for that request.
 - The ACCELSYS used by the complex is the largest of all the values specified in the complex

Non-XCF GRS RING Setup

- **IBM recommends at least a basic sysplex**
- Many functions available with sysplex that are not available in non-sysplex GRS Ring environments:
 - Fully automatic restart/rejoin
 - Dynamic RNL changes
 - Enhanced Contention Analysis
 - Many z/OS functions...
- But if you must ...
 - Must establish enough CTC links between the systems in the complex to establish a ring
 - Steady state is not enough!
 - More required for ACCELSYS (performance!)
 - GRS will use CTCs 'bi-directionally'
 - XCF use is PATHIN or PATHOUT

Ring - Defining GRS CTCs for non-Sysplex

CTC(unitaddr)

- GRS Ring complex only
- Indicates a device to be used for GRS communications
- Must be defined as a BCTC (SCTC will not work)

I/O Definition:

```
IODEVICEADDRESS=(F81,004),CUNUMBR=(01A1),STADET=Y,UNIT=BCTC
```

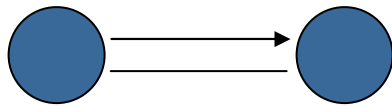
```
GRSDEF:
```

```
CTC (F81)
```

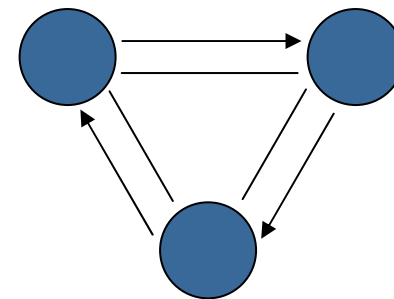
- If complex=sysplex, remove all definitions as handled by XCF
 - GRS CTCs will not be used

Ring - CTC considerations

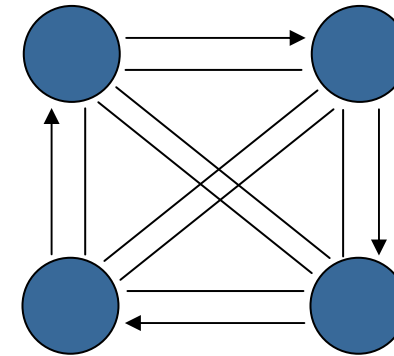
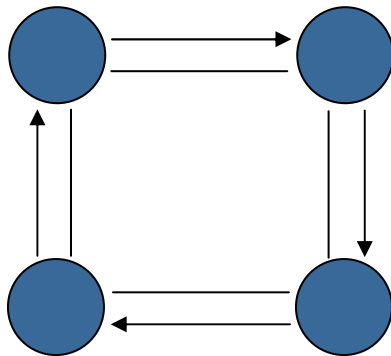
What happens if one system or one link fails?



Could survive with 1 CTC
(used bidirectionally)



Additional links required for ACCELSYS
and IPL/failure processing



Tuning

GRS does 3 things!

1. Processes ENQs

- Local (STEP,SYSTEM) are done under the client's thread
- Global (SYSTEMS) are done under GRS task(s) + SRB exists
 - High GRS CPU? Check RMF CF reports for false contention!

2. QSCANS

- Local (STEP,SYSTEM) collection is done under the client's thread
- Global (SYSTEMS)
 - RING – under client's thread on requesting system
 - STAR – under ISGQDR (task) and SRB exits on all systems. See follow on charts

3. Contention Notification processing

- RING – under client's thread except for global under GRS Task
- STAR – separate GRS task (ISGGCN) with communication to CNS system etc.. See follow on charts

GRS does 3 things! STAR Tuning

High GRS CPU? Bad elapse times?

1. Processes Global ENQs
 - Check RMF CF ISGLOCK Reports for false contention and asynch request %
2. QSCANs
 - Check for high contention rates
 - QSCAN abusers?
3. Contention Notification processing
 - Check RMF SYSTEMS level contention reports
 - CNS Drives QSCANs too!
 - Some ENF 51 receivers run under GRS signaling task!

GRS Star - Tuning

STRUCTURE NAME = ISGLOCK		TYPE = LOCK											
SYSTEM NAME	# REQ TOTAL AVG/SEC		REQUETS # REQ	% OF ALL	-SERV TIME(MIC)- AVG STD_DEV	REASON	# REQ	% OF REQ	DELATED REQUESTS /DEL	AVG TIME(MIC) STD_DEV /ALL	EXTERNAL REQUEST CONTENTIONS		
XCFD	1439K 797.4	SYNC	1439K	100%	92.0 22.6								
		ASYNC	0	0.0%	0.0 0.0	NO SCH	0	0.0%	0.0	0.0	0.0		
		CHNGD	0	0.0%	INCLUDED IN ASYNC								

TOTAL	1439K 797.4	SYNC	1439K	100%	92.0 22.6								
		ASYNC	0	0.0%	0.0 0.0	NO SCH	0	0.0%	0.0	0.0	0.0		
		CHNGD	0	0.0%									

REQ TOTAL	1410K
REQ DEFERRED	61K
-CONT	61K
-FALSE CONT	61K

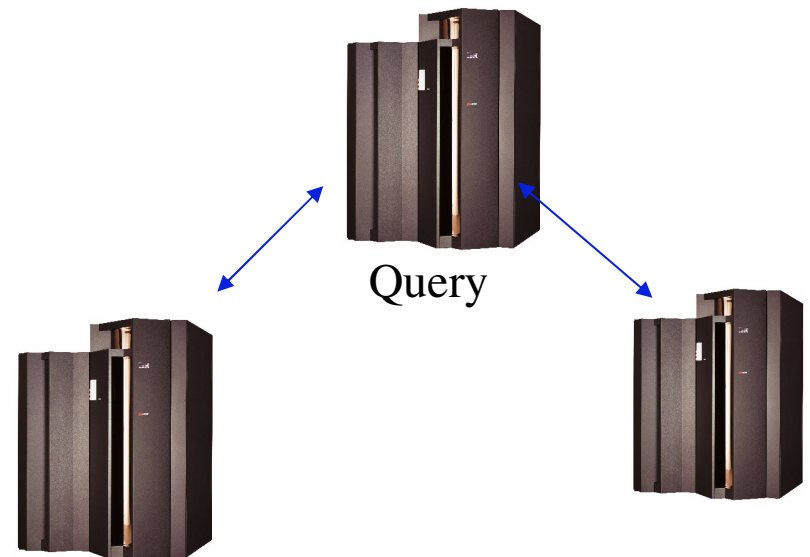
REQ TOTAL	1410K
REQ DEFERRED	61K
-CONT	61K

- Lock structure entries are determined by hashing on QNAME/RNAME.
 - Contention that results in the same entry being used for different resources is false contention which increases elapse time and system cpu + msg resources
- Never more than 1 to 2% false contention = $(\text{False Cont}/\text{ReqTotal}) * 100$
- Iterative process of increasing Lock Structure Size to see results

GRS Star – Tuning GQSCAN/ISGQUERY

SYSPLEX wide GQSCAN/ISGQUERYs can

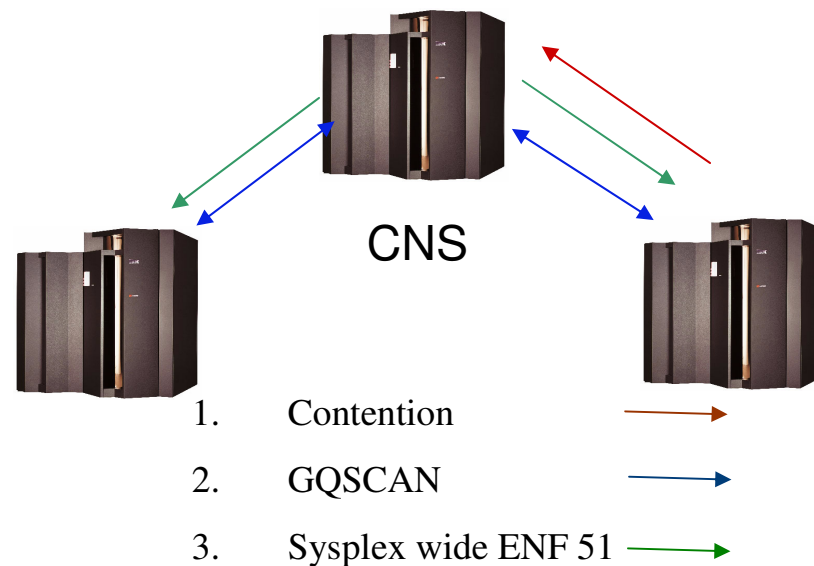
- be CPU intensive
- increase times
- Because it
 - Requires an XCF message with response that is sent to all systems in the sysplex in order to gather a sysplex view
 - Needs to merge the results
- Keywords
 - GQSCAN XSYS=YES (default)
 - ISGQUERY
 - GATHERFROM=SYSPLEX



GRS Star – Tuning ENF 51/CNS

ENF 51 is used to communicate contention to monitors

- As STAR mode systems only know about ENQs issued from their system, contention notification processing for SYSTEMS level ENQs requires sysplex wide coordination.
- At any given time, a single system is designated by the system to be the **Contention Notification System (CNS)**.
 - Automatically assigned by GRS
 - SETGRS CNS=sysname can be used to change CNS
 - No parmlib member support
 - D GRS shows what the contention notification system is
- ENF 51s can be filtered for specific resources via installation exits ISGCNFXITSYSTEM and ISGCNFXITSYSPLEX



GRS Ring - Tuning

- Really only one knob: RESMIL
 - The tradeoff:
 - Shorter RESMIL => faster response time
but
Shorter RESMIL => greater GRS CPU consumption
 - RESMIL=resmil will tune to resmil+5, if ring activity is low
 - RESMIL=OFF never tunes (always 0)
- For vastly improved performance use GRS Star
 - Microsecond vs. Millisecond response

Operational Interfaces

Star - DISPLAY GRS,SYSTEM

```
ISG343I 18.04.38 GRS STATUS          FRAME LAST  F      E  SYS=DOIT1
SYSTEM    STATE          SYSTEM    STATE
GRS1     CONNECTED      GRS2     CONNECTING
```

GRS STAR MODE INFORMATION

LOCK STRUCTURE (ISGLOCK) CONTAINS 1048576 LOCKS.
THE CONTENTION NOTIFYING SYSTEM IS GRS1

SYNCHRES: YES

ENQMAXU: 16384

ENQMAXA: 250000

GRSQ: CONTENTION

- Star Mode Display
 - Shows state of the systems in the GRS complex
 - Shows lock structure information
 - Shows the location of the CNS, and the GRSQ setting

Ring - DISPLAY GRS,SYSTEM

```
ISG343I 18.04.38 GRS STATUS          FRAME LAST  F      E  SYS=DOIT1
SYSTEM   STATE          SYSTEM   STATE
DOIT1   ACTIVE          DOIT2   QUIESCED
```

GRS RING MODE INFORMATION

```
RESMIL:      10
TOLINT:      180
SYNCHRES:    YES
ENQMAXU:     16384
ENQMAXA:     250000
```

- Ring Mode Display
 - Shows state of the systems in the GRS complex
 - Link information added for non-sysplex configuration
- All modes
 - SYNCHRES = the default synchronous reserve action the system should take
 - ENQMAXU/ENQMAXA = system wide default max unauthorized and authorized GRS ENQ and Query service per address space.
 - GRSQ=dump action GRS should take for SDATA(GRSQ)

DISPLAY GRS,RNL=

```

ISG343I 00.48.02 GRS STATUS          FRAME LAST  F      E      SYS=DOIT1
LIST TYPE  QNAME      RNAME
INCL GEN   SYSDSN
EXCL SPEC  SYSDSN      PASSWORD
EXCL SPEC  SYSDSN      SYS1.BROADCAST
EXCL SPEC  SYSDSN      SYS1.DAE
EXCL SPEC  SYSDSN      SYS1.DCMLIB
EXCL GEN   SYSDSN      SYS1.DUMP
EXCL SPEC  SYSDSN      SYS1.LOGREC
EXCL GEN   SYSDSN      SYS1.MAN
EXCL SPEC  SYSDSN      SYS1.NUCLEUS
EXCL GEN   SYSDSN      SYS1.PAGE
EXCL SPEC  SYSDSN      SYS1.STGINDEX
EXCL SPEC  SYSDSN      SYS1.SVCLIB
EXCL SPEC  SYSDSN      SYS1.UADS
EXCL GEN   SYSZJES2 SPOOL1SYS1.
NO ENTRIES EXIST IN THE RESERVE CONVERSION RNL
  
```

- Shows contents of current RNLs

DISPLAY GRS,CONTENTION

```
ISG343I 19.11.49 GRS STATUS          FRAME LAST   F       E   SYS=DOIT1
S=SYSTEMS SYSDSN   SYS1.LINKLIB
SYSNAME           JOBNAME        ASID        TCBADDR      EXC/SHR      STATUS
DOIT1             XCFAS          0006        005FFD90     SHARE        OWN
DOIT1             LLA            0016        005FFD90     SHARE        OWN
DOIT1             GRSTOOL        001D        005E6A68     EXCLUSIVE    WAIT
NO REQUESTS PENDING FOR ISGLOCK STRUCTURE
```

- Shows list of resources in contention
 - SYSNAME
 - JOBNAME
 - ASID
 - TCB

ENQ Contention– RMF Monitor III

- **Enqueues Delay Report (ENQ)**
 - A **single system report** that reports the data for one system that reports jobs that are waiting
- **Enqueue Resource Delays Report (ENQR)**
 - **Similar to ENQ** report but the report shows the ENQ delays from the a GRS resource view.. Order:
 - Resources in descending delay percentage by
 - *Jobs waiting in descending delay percentage by*
 - Job holding the resource in descending holding percentage
- **Sysplex Enqueue Delays Report (SYSENQ)**
 - A **sysplex report** which is **similar to ENQR** report but reports sysplex-wide delays only (SCOPE = SYSTEMS):

ENQ Contention–RMF SMF 77 Monitor I ENQ Activity Report



- **Enqueue Activity Report**
 - Post processor using SMF 77 queue build up information
 - Summary
 - By resource (Qname/Rname/Scope) for interval
 - Contention time
 - Queue length distribution/avg
 - Counts
 - *Exclusive/shared requests made*
 - *Number of contention events*
 - Detailed or Detailed for specific resource over an interval
 - System on which the job is running + job name
 - Counts
 - *Number of owning jobs and job names of 2 owners*
 - *Number of waiting jobs and job names of 2 waiter*
 - Indication if Exclusive or Shared request

DISPLAY GRS,ANALYZE

- Requests 'enhanced contention analysis':
 - List the waiting units of work, by length of time
 - The resource name and top blocker are also returned
 - List the blocking units of work, by length of time
 - The resource name and number of waiters are also returned
 - Analyze resource request dependencies
 - Identifies the "top blocker" in a string of dependent requests
 - Identifies resource request deadlock (deadly embrace)

DISPLAY GRS,ANALYZE...

D GRS,ANALYZE,BLOCKER from any system in the sysplex:

ISG349I 15.03.09 GRS ANALYSIS 984

LONG BLOCKER ANALYSIS: ENTIRE SYSPLEX

BLOCKTIME	SYSTEM	JOBNAME	E/S	SCOPE	QNAME	RNAME
00:01:33	PROD1	*MASTER**E*		SYS	SYSIEFSD	Q10
					OTHER BLOCKERS:	0 WAITERS: 1
00:00:57	PROD1	PRODJOB	*E*	SYSS	SYSDSN	PROD.DB
					OTHER BLOCKERS:	0 WAITERS: 2
00:00:44	PROD2	CLEANUP	*E*	SYSS	SYSDSN	PROD.PROCS
					OTHER BLOCKERS:	0 WAITERS: 1

DISPLAY GRS,ANALYZE...

D GRS,ANALYZE,WAITER from any system in the sysplex:

```
ISG349I 15.03.31 GRS ANALYSIS 987
LONG WAITER ANALYSIS:  ENTIRE SYSPLEX
WAITTIME  SYSTEM  JOBNAME E/S  SCOPE QNAME  RNAME
00:01:53  PROD1    PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER   PROD1    *MASTER* E
00:01:17  PROD2    CLEANUP *S*  SYSS SYSDSN  PROD.DB
BLOCKER   PROD1    PRODJOB  E  OTHER BLOCKERS: 0  WAITERS: 1
00:01:04  PROD2    SYSPROG *S*  SYSS SYSDSN  PROD.DB
BLOCKER   PROD1    PRODJOB  E  OTHER BLOCKERS: 0  WAITERS: 1
00:01:04  PROD2    SYSPROG *E*  SYSS SYSDSN  PROD.PROCS
BLOCKER   PROD2    CLEANUP  E
```

DISPLAY GRS,RES=

```
ISG343I 19.13.25 GRS STATUS          FRAME LAST  F      E  SYS=DOIT1
S=SYSTEMS SYSDSN  SYS1.LINKLIB
SYSNAME          JOBNAME          ASID      TCBADDR    EXC/SHR    STATUS
SYS1             XCFAS             0006      005FFD90   SHARE     OWN
SYS1             LLA               0016      005FFD90   SHARE     OWN
```

- Shows list of resources by resource name
 - SYSNAME
 - JOBNAME
 - ASID
 - TCB

DISPLAY GRS,DEV=

```
ISG343I 12.57.26 GRS STATUS          FRAME LAST   F       E   SYS=DOIT2
DEVICE:027E VOLUME:TMPPAK NOT RESERVED BY SYSTEM DOIT2
NO RESERVE RESOURCE REQUEST EXISTS
```

- Shows whether a specified device is RESERVED by the system
- Does not show if devices are RESERVED by other systems

The SETGRS Command

RESMIL=

TOLINT=

SYNCHRES=

GRSQ=

ENQMAXA=

ENQMAXU=

CNS=

- SETGRS changes values for the local system only (except for CNS=)
 - Use ROUTE *ALL to affect a sysplex-wide change
- CNS= affects the whole sysplex

Contrasting ENQ/DEQ with Latches



Latch Sets

1	
2	
x	

- APIs: ENQ,DEQ,Reserve,ISGENQ,GQSCAN, ISGQUERY,ISGADMIN
- Resource Identity: QNAME, RNAME, SCOPE
- Scope: JOB STEP,SYSTEM, GRS Complex
- Shared/Exclusive ownership
- Authorized/Unauthorized
- Widely used
- Reasonable performance
- Installation controls – RNLs and Exits
- APIs: Create, Obtain, Release and Purge
- Resource Identity: latch#
- Scope: single system
- Shared /Exclusive ownership
- Authorized only
- Widely used by the system/subsystems
- Very fast
- No installation controls

Appendix

- References:
 - [SA22-7600] z/OS Planning: Global Resource Serialization
 - [SA22-7592] z/OS Initialization and Tuning Reference
 - [SA22-7265] z/OS Setting Up a Sysplex
 - [SA22-7267] z/OS System Commands
 - [SA22-7593] z/OS Installation Exits
 - [SA22-7639] z/OS System Messages Vol. 9
 - [SC33-7991] z/OS Resource Measurement Facility Report Analysis