


IBM

z/OS Design and Development

**GRS 101:
Non-Sysplex Ring, Sysplex Ring, and Star**




Nicholas C. Matsakis
z/OS GRS Development
Core Technology Design, IBM Corporation
matsakis@us.ibm.com

NY Metro NaSPA®

© 2005 IBM Corporation

z/OS Design and Development



Permission is granted to NY Metro NaSPA® to copy, reproduce or republish this document in whole or in part for their activities only.

NY Metro NaSPA®

© 2005 IBM Corporation



The following are trademarks or registered trademarks of the International Business Machines Corporation:



- Parallel Sysplex®**
- OS/390®**
- z/OS®**
- z/VM®**
- Websphere®**

Topics



- What's GRS?
- Programming interfaces
- Getting started
- Operational interfaces
- Tuning
- Questions

z/OS Design and Development





NY Metro NaSPA®

© 2005 IBM Corporation

z/OS Design and Development

GRS – ENQ and Latch Services

1	
2	
x	

- APIs: ENQ,DEQ,Reserve,ISGENQ,GQSCAN,ISGQUERY
- 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 systems/subsystems
- Very fast
- No installation controls

NY Metro NaSPA © © 2005 IBM Corporation

GRS provides two sets of critical system serialization services. When GRS services are not working well you know it because something usually fails or there is poor performance.

- The GRS ENQ services provide the ability to serialize an abstract resource within the scope of a JOB STEP, SYSTEM or multi-system complex (GRS Complex). The GRS complex is usually equal to the sysplex but it does not have to be. Via the HW reserve function, DASD Volumes can be shared between different systems that are not in the same GRS complex or even the same operating system. For example, between z/VM, ..., and z/OS. ENQ/Reserve services can be used by authorized or unauthorized users. Almost every component, subsystem, and many applications use ENQ in some shape or form.


- The GRS latch services provide a high speed serialization service for authorized callers. Latch services know nothing of the intended scope. Scoping is completely control by the user. It uses user provided storage to manage a lock/latch table that is indexed by a user defined lock/latch number. GRS latch is also widely used. Very big users are USS, Logger, RRS, MVS, etc... The user is required to be in the latch set creator's space when using the latch set.

- GRS latch non-contention path is on the order of 10s of instructions while ENQ is on the order of a thousand for a local (single system) ENQ. GRS latch requires more recovery type coding on behalf of the user i.e. resource manager cleanup at task, address space, etc. termination.

- Other serialization means include system locks (i.e. LOCAL Lock), home grown CS type of latches, and XES IXLLOCK exploitation for cross system sharing i.e. IRLM for DB2/IMS.

z/OS Design and Development

ENQs – Resource Identity




- ENQ “resource identity” is determined from the final:
 - QNAME (Queue or Major name),
 - RNAME (Resource or Minor name) ,
 - SCOPE (STEP, SYSTEM or SYSTEMS=SYSPLEX).
- UCB on RESERVE specifies the device to reserve after the ENQ is obtained

NY Metro NaSPA © © 2005 IBM Corporation

- GRS uses the “Resource identity” to determine what “abstract resource” will be serialized. It is made up of the “final” QNAME, RNAME, and SCOPE. This identity can be changed by various means from what was originally specified on the API.
- Note
 1. that the disposition (exclusive/shared) is not part of the resource identity.
 2. In GRS=NONE environments or multi-system configurations where there is only 1 system in the GRS complex, the SCOPE is still part of the resource identity. Thus, two ENQs with the same QNAME/RNAME but different scopes are always considered to be different resources.

z/OS Design and Development

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 – make system systems
- Exclusion – make systems system
- Conversion – convert reserve to ENQ only

•GRS Installation exits:

- ISGNQXIT/ISGNQXITFAST
 - See APAR OW56028 z/OS V1R2-V1R4
- OEM oriented exits:
 - ISGNQXITPREBATCH, ISGNQXITBATCHCND, ISGNQXITBATCH, ISGNQXITQUEUEU1, ISGENDOFQCB
 - See white paper: “GRS/MIM Installation Exit Performance”

NY Metro NaSPA © © 2005 IBM Corporation

- Applications specify a specific QNAME, RNAME, and scope on the GRS ENQ/DEQ APIs to uniquely identify the resource that is to be serialized. The installation can use the GRS RNLs to change the SCOPE or to convert a RESERVE request to a local or global ENQ. IBM recommends converting all possible reserves to GLOBAL ENQs in GRS STAR mode. See the GRS planning guide for more details. Applications can prevent alterations of scope that they specify by coding the RNL=NO keyword on the API.
- All the exits mentioned on this chart are dynamic exit points and can be used by the installation or third party automation or serialization products. For example, CA MIM uses the “OEM” oriented exits mentioned on this chart.
- The ISGNQXIT or ISGNQXITFAST installation exit can be used to change the QNAME, RNAME, UCB, SCOPE, prior to RNL processing. It can also request that the RNLs are not searched. The exit gets control on every ENQ,DEQ,RESERVE. The RNLs are searched on every ENQ where the RNL=NO keyword was NOT specified or an exit did want the RNLs to be searched.
- ISGNQXITFAST offers better performance than isgnqxit (may not be applicable for use in all environments, please see doc in apar). The customer needs to take action to use it and should use either ISGNQXIT or ISGNQXITFAST.
- ISGNQXIT was introduced along with GRS Wildcard RNLs in z/OS R2 (retrofitted to OS/390 and z/OS R1 via OW49779), GRS replaced the RNL Search exit interface, known as ISGGREX0, with the new dynamic exit point called ISGNQXIT.
- The installation OEM oriented exits were provided for CA MIM (and possibly other products) to allow them to coexist with LINKAGE=SYSTEM (PCENQ) support that was added to GRS for Automatic Tape Switching (ATS) via the service stream. See the white paper “GRS/MIM Installation Exit Performance” for more details.
- The GRS/MIM Installation Exit white paper can be found at: <http://www-1.ibm.com/servers/eserver/zseries/library/literature/papers.html>

z/OS Design and Development

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 excluded to SYSTEM
 - Not done due to
 - conversion via RNLs OR
 - unshared device (IODF)

NY Metro NaSPA®

© 2005 IBM Corporation

ENQs are used for many things. “Local ENQs” are referred to ones that only serialize across unit of work within the same system or address space. Their resource identity would have a SCOPE = to STEP or SYSTEM. Where STEP is within the same address space and SYSTEM is within the same z/OS image.

Note that RESERVEs always have two pieces, there is the ENQ and the actual HW reserve of the device. The RESERVE ENQ always starts out as SCOPE=SYSTEMS but can be excluded to SYSTEM in cases where global serialization is not needed in a multi-system GRS complex. So, the IODF SHARED(YES) attribute of the device determines if the HW reserve is done and the SCOPE of the ENQs if the ENQ is global or local.

z/OS Design and Development

Global Resource Sharing

- Concurrent/uncontrolled resource allocation
 - Data consistency errors
 - Data integrity errors

Shared DASD

MVS Images

NY Metro NaSPA®

© 2005 IBM Corporation

Resources that are shared among many units of work need to be serialized in order to maintain data integrity. It needs to be noted that in some cases the serialization of the resources are controlled by job scheduling rather than using a programmatic serialization protocol.

z/OS Design and Development

Global Resource Sharing: HW 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

NY Metro NaSPA®

© 2005 IBM Corporation

HW Reserve can be used to serialize DASD resources between systems. It has many drawbacks. Note that in the past RESERVE was faster than global (SYSTEMS) ENQs but GRS STAR and fast CFs have made global ENQs faster than RESERVES.

When sharing outside of the GRS complex (or extended complex via third party integrity products), HW RESERVE may be your only option. For example, sharing between LINUX or VM and Z/OS or between test/production (usually not a good idea).

z/OS Design and Development

Global Resource Sharing:
RESERVE - cons

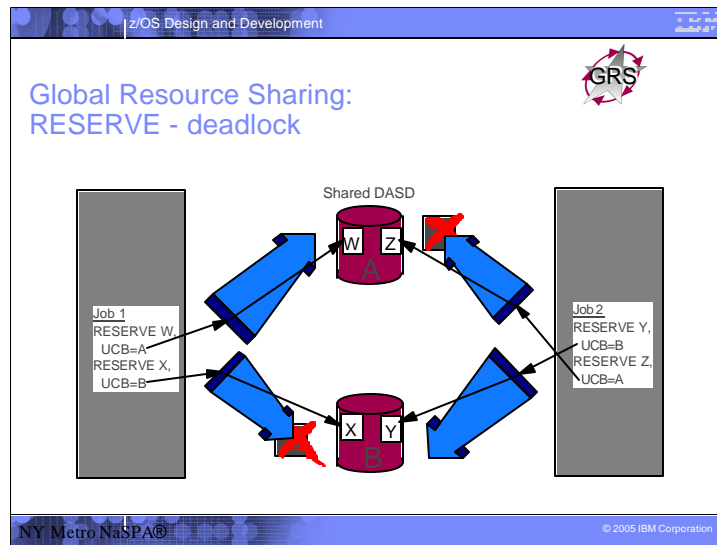
- Locks whole device
 - ▶ can cause performance/deadlock issues
- Does not distinguish between read and write access
- Starvation – no fair sharing!
- Deadlock possible
 - ▶ aka "Deadly Embrace" due to
 - configuration errors – hard to manage
 - due to non-synchronous reserves – hidden got ya..

NY Metro NaSPA®

© 2005 IBM Corporation

In general, if you can use a global ENQ and the ENQ is managed by GRS STAR or a third party integrity product performances well enough, you should consider converting the RESERVE to a global ENQ.

Devices can have many things on them and from what is on them it is hard to determine what might be required by who. As such, locking an entire device when only a single dataset is required to be serialized usually leads to performance or deadlock issues.



As the granularity of a HW RESERVE is the entire volume which may contain many different resources, RESERVEs lend themselves to cause deadlock more often than ENQs. In this example, the RESOURCES should be considered to be the two devices, A and B. As they are obtained in different orders the two jobs get deadlocked. This is true even though the real resources W,X,Y, and Z have no relation to one another. Installation configuration is usually required to eliminate such cases. However, it is hard to determine what the relationships are as code running under a job may call a service which requires a resource that was completely hidden. If the RESERVEs were converted to global ENQs, then none of the resources would even be in contention.



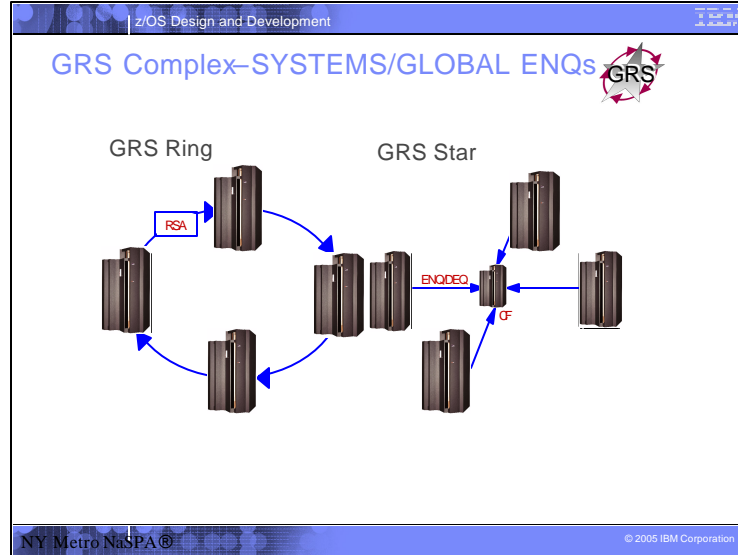
Global Resource Serialization Advantages:

- 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 z/OS sysplex)
 - ▶ Shared or Exclusive access
- Fairness (FIFO)
 - ▶ Ensures no starvation



Global Resource Serialization Disadvantages:


- Introduces MVS overhead
- Requires sysplex
or
- Requires additional hardware and significant additional setup
- Requires additional setup
 - GRSCNFxx
 - GRSRNLxx



For GLOBAL or multi-system ENQs, GRS uses either a RING or STAR mode configuration in order to communicate with other instances within the GRS complex. IBM recommends GRS STAR for performance as well as availability reasons. RING uses its own CTCs or XCF messaging if in a sysplex to send an RSA (contains all Global Resources) around the “RING” of systems in the GRS complex. Each ENQ cannot be granted until all other systems in the RING have seen the request. STAR uses a Coupling Facility lock structure so each system can go directly to the coupling facility to get an ENQ. No XCF messages are sent in non-contention cases.

z/OS Design and Development

Global Resource Serialization




<p>GRS Ring</p> <ul style="list-style-type: none"> ▪ Peer Coupling ▪ Connected via XCF and/or GRS managed CTCs ▪ Time slicing via RSA ▪ Global data view ▪ Can bridge between <ul style="list-style-type: none"> ▪ a single multi-system sysplex and Monplexes or XCF local system ▪ Monplex /XCF local systems ▪ Performance proportional to number of ENQs and number of systems <ul style="list-style-type: none"> ▪ Does not scale well ▪ Functionally stabilized 	<p>GRS Star</p> <ul style="list-style-type: none"> ▪ Peer Coupling ▪ Connected via CF Lock Structure <ul style="list-style-type: none"> ▪ XCF signaling for contention management ▪ No time slicing ▪ Local data view <ul style="list-style-type: none"> ▪ Global view is provided on request ▪ All systems must be part of the same sysplex ▪ Significant performance benefits <ul style="list-style-type: none"> ▪ Scales very well ▪ Target for global sharing enhancements
--	--

NY Metro NaSPA®

© 2005 IBM Corporation

GRS STAR is the best performer by far. However, due to SW pricing issues, other costs, and/or configurations GRS RING provides value to some customers. Also, remember that GRS is always there regardless of third party integrity software being installed. For global ENQs, GRS will always field any that specify RNL=NO (they know that their ENQ should never be greater than the GRS complex's scope) or are not handled by the third party software (potentially not needed to be handled). As such, STAR's performance and RAS (usability) can still be beneficial when third party software is being used.

z/OS Design and Development




Programming Interfaces

NY Metro NaSPA®

© 2005 IBM Corporation

z/OS Design and Development

GRS ENQ – API History



- **ENQ, DEQ, RESERVE**
 - ENQ/DEQ: Obtain an abstract resource (QNAME, RNAME, SCOPE, and DISPOSITION)
 - RESERVE: Obtain an abstract resource and also do HW reserve on a volume (UCB)
 - Originally SVC only interfaces which limits their usage
- **z/OS V1R2 SPE** to support a PC interface (LINKAGE=SYSTEM) for cross memory mode callers. Required by Alternate Tape Serialization (ATS Star)
 - Introduced new OEM GRS exits for third party serialization products
- **z/OS V1R6** New ISGENQ and ISGQUERY interfaces to support AMODE 64, AR mode, ...

NY Metro NaSPA © © 2005 IBM Corporation

GRS's main interfaces were always SVC entered. Via a z/OS V1R2 SPE, GRS provided the LINKAGE=SYSTEM keyword on ENQ,DEQ,RESERVE to allow these services to be issued in cross memory mode. A cross memory environment is typical for server type environments. This was done for ATS STAR. However, the interface has been gaining other users as they too require the support. JES2 is an example. In z/OS V1R6 we introduced a new ISGENQ service which provides ENQ,DEQ,RESERVE support in one service. The goal was to provide AMODE 64 support and to provide better RAS (Reliability Availability Serviceability) for GRS's users. Some of these RAS items had to be done at this time as it would be harder to do in the future.

z/OS Design and Development

ENQ/ISGENQ
Request Control of a Serially Reusable Resource

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

NY Metro NaSPA®

© 2005 IBM Corporation

ENQ and ISGENQ are the API for obtaining an abstract resource either shared or exclusive. ISGENQ was introduced in z/OS V1R6.

ISGENQ OBTAIN returns an ENQTOKEN which can be used on a subsequent ISGENQ RELEASE.

z/OS Design and Development

GRS

RESERVE/ISGENQ

Reserve a Device (Shared DASD)

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

NY Metro NaSPA®

© 2005 IBM Corporation

As stated previously, for each RESERVE there is always an associated ENQ. This is required because the RESERVE only serialized the device between the issues host images and not a particular unit of work. The ENQ, should be local (SYSTEM) in scope, serializes between the sharing units of work.

z/OS Design and Development

Release a Serially Reusable Resource

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

NY Metro NaSPA®

© 2005 IBM Corporation

For the DEQ API, the same resource identity that was specified on the ENQ must be provided on the DEQ. Any installation exit alterations that are made must also result in the same values. Otherwise, the ENQ will not be found correctly. This could result in an ABEND (DEQ with no ENQ) or an integrity problem... the wrong ENQ being released.

For an ISGENQ all that is required to be provided for a release is the ENQTOKEN that was returned on ISGENQ. This insures that the DEQ matches the ENQ that was performed. It also prevents installation exits for attempting to alter the target of the DEQ in error.

Note that UCB is defined as part DEQ of the API but is not required and not recommended to be specified.

z/OS Design and Development

GRS

GQSCAN/ISGQUERY

Extract Information from GRS


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

NY Metro NaSPA®

© 2005 IBM Corporation

ISGQUERY was introduced in z/OS 1.6. It is now the recommended GRS query service.

z/OS Design and Development



Getting Started

NY Metro NaSPA®

© 2005 IBM Corporation



Create GRS complex

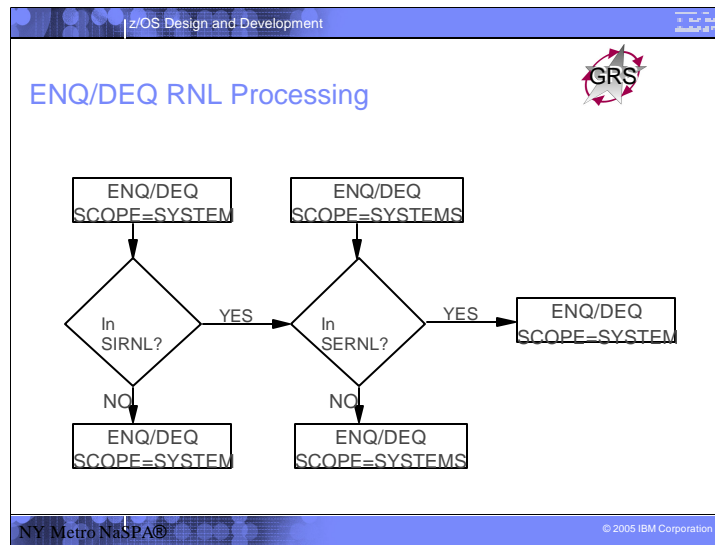
- Read
 - ▶ z/OS MVS: Setting up a Sysplex (GC28-1779)
 - ▶ z/OS MVS Planning: Global Resource Serialization (GC28-1759)
 - ▶ Sysplex Migration Guide (Redbook) (GG24-4368)
 - ▶ Merging Systems into a Sysplex (Redbook) (SG24-6818-00)
- Choose complex type
IBM recommends GRS Star for new complexes
- Understand and define RNLs (Resource Name Lists)
- Use the GRS monitor to help determine how to change your RNLs
- Update PARMLIB
 - ▶ IEASYSxx
 - ▶ GRSCNFxx
 - ▶ GRSRNLxx



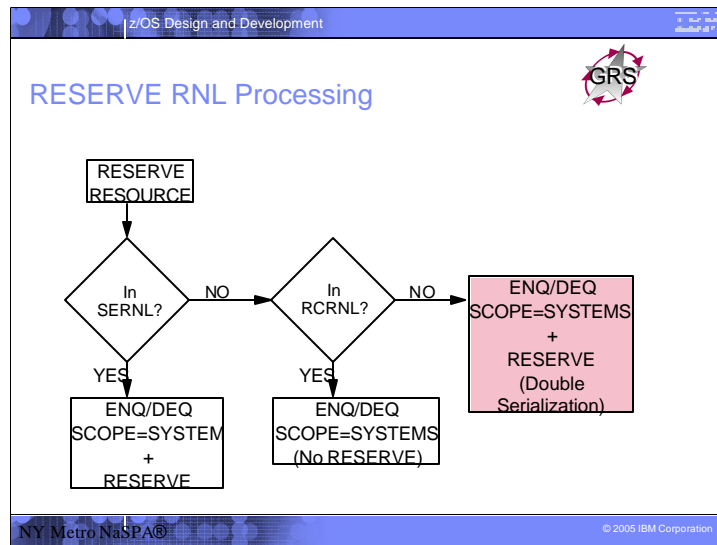
Resource Name Lists

- Influence the scope of ENQ and DEQ processing
- Influence if RESERVE occurs
- Three lists
 - ▶ SYSTEM Inclusion (SIRNL)
Promotes local ENQ to global ENQ
 - ▶ SYSTEMS Exclusion (SERNL)
Demotes global ENQ to local ENQ
 - ▶ RESERVE Conversion (RCRNL)
Suppresses RESERVE to be global ENQ only

**Rule of thumb:
If the resource is shared, it should be
considered for RNL processing.**



It is important to understand this flow chart. SIRNL=Systems Inclusion List and SERNL = Systems exclusion List. The original scope of the ENQ request determine where you enter the flow chart. For example, if SYSTEMS request then the SIRNL is not searched as it already started out as a SYSTEMS request. A SYSTEM request which is promoted via the SIRNL can be demoted/excluded via the SERNL. Many times, for example SYSDSN QNAME, customers will specify a generic QNAME in the SIRNL and then exclude by a specific datasets which do not need to be globally shared via a more specific entry in the exclusion list SERNL.



Reserves always start out as SYSTEMS level ENQs because it is the only scope that can be provided on the APIs. The general rule is that a RESERVE should either:

1. be performed and its associated ENQ should be excluded (in SERNL) to a local (SYSTEM) ENQ or
2. not be performed (converted via the RCRNL) and a global (SYSTEMS) level ENQ should be performed.

Not converting and not excluding can lead to a case where both the ENQ and RESERVE are done! This can cause deadlocks and is not recommended.

Note that the RCRNL is not converted if the ENQ is excluded via the SERNL. Some customers have not understood this which resulted in problems because they thought they were converting a RESERVE. In a future release, GRS will be adding a health checking checker to identify cases where a RCRNL entry would not be used because all associated RESERVEs would be excluded via the SERNL anyway.

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 GRSRNL=EXCLUDE



- Used to defeat global GRS processing
- Useful when:
 - Using alternative serialization
- GRS only processes specific ENQs globally
 - RNL=NO
- Can not be changed without a re-ipl!
 - Using an exclusion RNL with a wildcard to exclude all SYSTEMS may be a better choice.






PARMLIB: IEASYSxx

NONE
JOIN
GRS= 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

z/OS Design and Development

PARMLIB: GRSCNF xx



SYNCHRES{YES|NO}

- GRS Ring, Star or None
- 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
- Default: YES starting with z/OS R1V6
 - ▶ GRSCNFxx SYNCHRES{YES} for prior releases
 - ▶ SETGRS SYNCHRES=YES/NO

NY Metro NaSPA®

© 2005 IBM Corporation

SYNCHRES=YES is recommended as a RESERVE deadlock prevention feature.

The default is SYNCHRES=YES in z/OS 1.6 and up.

Note that for GRS=NONE, GRSCNFxx was not parsed. This is being fixed in a future release.

PARMLIB: GRSCNF *xx* RING - RESMIL



RESMIL(number|OFF)

- GRS Ring complex
- RESidency time in MILiseconds
 - How long GRS holds the RSA
- Default: 10
 - 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
- SETGRS RESMIL=


PARMLIB: GRSCNF *xx* RING - TOLINT



TOLINT(number)

- GRS Ring complex
- TOLeration time INTerval in seconds
 - How long GRS will wait for the RSA to arrive before triggering an error condition
- Default: 180
- SETGRS TOLINT=

z/OS Design and Development

PARMLIB: GRSCNF *xx* – RING ACCELSYS 

ACCELSYS(number)

- GRS Ring complex
- 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


NY Metro NaSPA® © 2005 IBM Corporation

Ring acceleration can significantly reduce the amount of time tasks spend waiting for global resources, especially in a large complex. Ring acceleration also requires alternate links, except between systems in a sysplex, and IBM recommends that the complex be a fully-connected complex. An installation where the complex is the same as the sysplex does not need to perform any additional setup to use ring acceleration. Using ring acceleration changes the processing of a global resource request such that all systems in the RING do not need to see an ENQ/DEQ before the request is considered complete.

Recovery: Ring acceleration provides significant performance improvement, but it does introduce recovery considerations. The ACCELSYS value specifies the number of systems that must see the RSA-message before the originating system can grant a request. It also specifies the number of consecutive systems that can fail before ring acceleration introduces a possible data integrity exposure. See the GRS Planning Guide for more details.

z/OS Design and Development

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

NY Metro NaSPA®

© 2005 IBM Corporation

A non-sysplex RING is not recommended... As it provides the worst performance and RAS. You're using pre 1990 GRS technology! Long term saving may be worth more than what you think you are actually saving.

PARMLIB: GRSCNF *xx RING CTCs***CTC(unitaddr)**

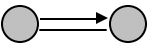
- GRS Ring complex
- Indicates a device to be used for GRS communications
- Must be defined as a BCTC (SCTC will not work)
I/O Definition:
IODEVICEADDRESS=(F81,004) , CUNUMBR=(01A1) , STADET=Y, UNIT=BCTC
GRSDEF:
CTC (F81)
- If complex=sysplex, remove all definitions as handled by XCF
 - ▶ GRS CTCs will not be used

z/OS Design and Development

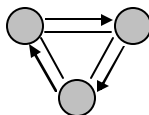
GRS

CTC considerations

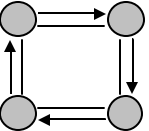

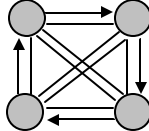
What happens if one system or one link fails?



Could survive with 1 CTC
(used bidirectionally)



Additional links required for ACCELSYS
and IPL/failure processing


NY Metro NaSPA®

© 2005 IBM Corporation

Make sure that you do not have single points of failure! Multiple CTC across multiple control units. If the RSA can not make it around the ring in time, perhaps due to a CTC failure, GRS will cause a RING disruption in order to “rebuild the ring”. This can be painful and may require that a system that does not have connectivity be quiesced and removed from the GRS complex. When the COMLPEX=SYSPLEX, these concerns are removed as XCF will insure multiple paths and will allow paths to be shared between users. In addition, it handles monitoring of systems and also will automatically remove an ill system.

z/OS Design and Development

Star Complex Setup



- 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)
 - ▶ Function enabled via APAR OA07975
- Consider converting RESERVEs for ENQ processing


NY Metro NaSPA®

© 2005 IBM Corporation

Setting GRSCNFxx to indicate GRSQ(CONTENTION) dramatically reduces the amount of the time GRS takes to collect SYSPLEX wide ENQ information for SDUMP which specify SDATA=GRSQ. In a future release, IBM plans to provide an operator command to be able to set the GRSQ value without a required IPL. So for now, get it right when you start.

IBM recommends that conversion of RESERVEs to global ENQs when in a STAR environment. Care must be taken to determine which can and can not be converted. See the documentation for more information.

z/OS Design and Development



Operational Interfaces

NY Metro NaSPA®

© 2005 IBM Corporation

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
```

- Ring Mode Display
 - Shows state of the systems in the GRS complex
 - Link information added for non-sysplex configuration

z/OS Design and Development

GRS

DISPLAY GRS,SYSTEM

```

ISG343I 18.04.38 GRS STATUS          FRAME LAST  F    E  SYS=DOIT1
SYSTEM   STATE          SYSTEM   STATE
DOIT1    CONNECTED      DOIT2    CONNECTING
  
```

GRS STAR MODE INFORMATION
 LOCK STRUCTURE (ISGLOCK) CONTAINS 1048576 LOCKS.
 THE CONTENTION NOTIFYING SYSTEM IS DOIT1
 SYNCHRES: YES

- Star Mode Display
 - ▶ Shows state of the systems in the GRS complex
 - ▶ Shows lock structure information

NY Metro NaSPA® © 2005 IBM Corporation

Note that the GRS displays do not show any stats on the lock structure/CF. You'll need to use the D CF, D XCF,CF=, D XCF,STRNAME=ISGLOCK commands for that and also RMF types of reports to see false contention rates.

DISPLAY GRS,RNL=



```
ISG343I 00.48.02 GRS STATUS          FRAME LAST  F    E  SYS=FAGEN1
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
```

z/OS Design and Development

DISPLAY GRS,CONTENTION

ISG343I 19.11.49 GRS STATUS FRAME LAST F E SYS=FAGEN1

S=SYSTEMS SYSDSN SYS1.LINKLIB

SYSNAME	JOBNAME	ASID	TCBADDR	EXC/SHR	STATUS
FAGEN1	XCFAS	0006	005FFD90	SHARE	OWN
FAGEN1	LLA	0016	005FFD90	SHARE	OWN
FAGEN1	GRSTOOL	001D	005E6A68	EXCLUSIVE	WAIT

NO REQUESTS PENDING FOR ISGLOCK STRUCTURE

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

NY Metro NaSPA® © 2005 IBM Corporation

D GRS,Contention display contention for both ENQs and GRS latch structures. However, for ENQ analysis, D GRS,ANALYZE is the recommended command. There is currently no ANALYZE function for GRS latches.

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,RES=



```
ISG343I 19.13.25 GRS STATUS          FRAME LAST  F    E  SYS=FAGEN1
S=SYSTEMS SYSDSN  SYS1.LINKLIB
SYSNAME      JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
FAGEN1      XCFAS          0006      005FFD90  SHARE    OWN
FAGEN1      LLA              0016      005FFD90  SHARE    OWN
```

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

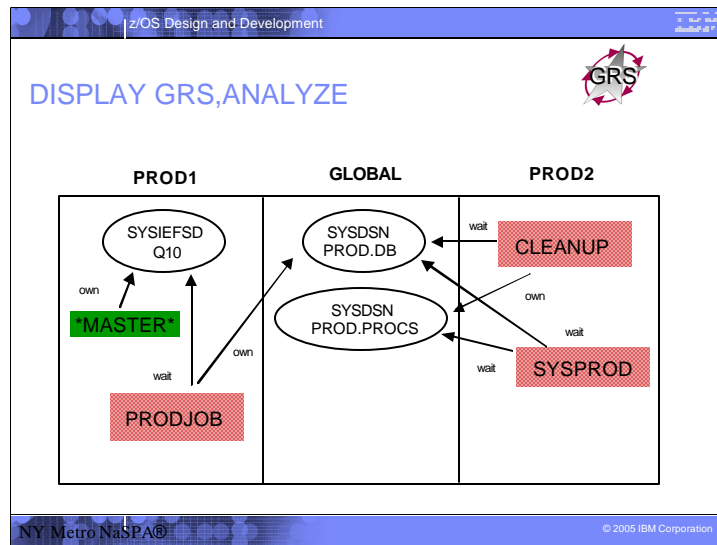
DISPLAY GRS,DEV=



```
ISG343I 12.51.06 GRS STATUS      FRAME 1   F   E   SYS=FAGEN1
DEVICE:027E VOLUME:TMPPAK RESERVED BY SYSTEM FAGEN1
```

```
ISG343I 12.57.26 GRS STATUS      FRAME LAST F   E   SYS=FAGEN2
DEVICE:027E VOLUME:TMPPAK NOT RESERVED BY SYSTEM FAGEN2
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



Lockout scenario:

MASTER on PROD1 gets the SYSIEFSD.Q10 (command) resource and a program error occurs, causing the resource to not be freed.

PRODJOB is running, owning the SYSDSN.PROD.DB resource.

PRODJOB issues a command and gets blocked on the SYSIEFSD.Q10 resource.

The CLEANUP job is submitted, obtains the SYSDSN.PROD.PROCS resource, but gets blocked on the SYSDSN.PROD.DB resource.

The system programmer tries to run some analysis program and gets blocked on both the SYSDSN.PROD.DB and PROD.PROCS resources.

When a D GRS,C command is routed to PROD1, it cannot run (command processing cannot obtain the SYSIEFSD.Q10 resource). A D GRS,C command on PROD2 does not show the relationship between *MASTER* and PRODJOB.

DISPLAY GRS,ANALYZE...



D GRS,C from PROD1:

ISG343I 15.02.58 GRS STATUS 981

S=SYSTEMS SYSDSN PROD.DB

SYSNAME	JOBNAME	ASID	TCBADDR	EXC/SHR	STATUS
PROD1	PRODJOB	001A	007E7B68	EXCLUSIVE	OWN
PROD2	CLEANUP	0029	007E7B68	SHARE	WAIT
PROD2	SYSPROG	0027	007E7B68	SHARE	WAIT

S=SYSTEMS SYSDSN PROD.PROCS

SYSNAME	JOBNAME	ASID	TCBADDR	EXC/SHR	STATUS
PROD2	CLEANUP	0029	007E7B68	EXCLUSIVE	OWN
PROD2	SYSPROG	0027	007E7B68	EXCLUSIVE	WAIT

S=SYSTEM SYSDSN Q10

SYSNAME	JOBNAME	ASID	TCBADDR	EXC/SHR	STATUS
PROD1	*MASTER*	0001	007E6B40	EXCLUSIVE	OWN
PROD1	PRODJOB	001A	007E7B68	EXCLUSIVE	WAIT

This command can't run because no commands can run on PROD1!

DISPLAY GRS,ANALYZE...



D GRS,C from PROD2:

ISG343I 15.05.24 GRS STATUS 539

S=SYSTEMS SYSDSN PROD.DB

SYSNAME	JOBNAME	ASID	TCBADDR	EXC/SHR	
PROD1	PRODJOB	001A	007E7B68	EXCLUSIVE	OWN
PROD2	CLEANUP	0029	007E7B68	SHARE	WAIT
PROD2	SYSPROG	0027	007E7B68	SHARE	WAIT

S=SYSTEMS SYSDSN PROD.PROCS

SYSNAME	JOBNAME	ASID	TCBADDR	EXC/SHR	
PROD2	CLEANUP	0029	007E7B68	EXCLUSIVE	OWN
PROD2	SYSPROG	0027	007E7B68	EXCLUSIVE	WAIT

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,ANALYZE...

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

```
ISG349I 15.03.54 GRS ANALYSIS 990
DEPENDENCY ANALYSIS:  ENTIRE SYSPLEX
----- LONG WAITER #1
WAITTIME  SYSTEM  JOBNAME E/S  SCOPE QNAME  RNAME
00:02:16  PROD1   PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER   PROD1   *MASTER* E
--:--:--  PROD1   *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
----- LONG WAITER #2
WAITTIME  SYSTEM  JOBNAME E/S  SCOPE QNAME  RNAME
00:01:40  PROD2   CLEANUP *S*  SYSS SYSDSN  PROD.DB
BLOCKER   PROD1   PRODJOB E
00:02:16  PROD1   PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER   PROD1   *MASTER* E
--:--:--  PROD1   *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING

[Output continued on next page]
```



DISPLAY GRS,ANALYZE...

```
----- LONG WAITER #3
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME  RNAME
00:01:27  PROD2   SYSPROG *S* SYSS SYSDSN  PROD.DB
BLOCKER   PROD1   PRODJOB  E
00:02:16  PROD1   PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER   PROD1   *MASTER* E
--:--:--  PROD1   *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
----- LONG WAITER #4
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME  RNAME
00:01:27  PROD2   SYSPROG *E*  SYSS SYSDSN  PROD.PROCS
BLOCKER   PROD2   CLEANUP  E
00:01:40  PROD2   CLEANUP *S*  SYSS SYSDSN  PROD.DB
BLOCKER   PROD1   PRODJOB  E
00:02:16  PROD1   PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER   PROD1   *MASTER* E
--:--:--  PROD1   *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
```

z/OS Design and Development


SETGRS

**RESMIL=
TOLINT=
SYNCHRES=**

- Changes value for the local system only
- Use ROUTE *ALL to effect a sysplex-wide change


NY Metro NaSPA®

© 2005 IBM Corporation



RESMIL and TOLINT are RING mode only parms.

z/OS Design and Development




Tuning

NY Metro NaSPA®

© 2005 IBM Corporation

z/OS Design and Development

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


NY Metro NaSPA®

© 2005 IBM Corporation

If the ring is lightly loaded, GRS will tune the RESMIL value up one millisecond each time an empty RSA makes a trip around the sysplex until RESMIL reaches the specified value plus 5 (RESMIL=1 will tune between 1 and 6 milliseconds). When the ring becomes loaded, RESMIL returns to the specified value. When an installation specifies RESMIL=OFF, the RSA will be sent immediately after receipt and processing by each system, without tuning. This might adversely impact CPU performance and should be used with care.

z/OS Design and Development

GRS Star - Tuning



```

STRUCTURE NAME = ISGLOCK          TYPE = LOCK
-----
SYSTEM # REQ  REQUESTS  DELAYED REQUESTS  EXTERNAL REQUEST
NAME   TOTAL  #    % OF  -SERV TIME(MIC)-  REASON #    % OF  --- AVG TIME(MIC) ---  CONTENTIONS
      AVG/SEC  REQ  ALL  AVG  STD_DEV  REQ  REQ  /DEL  STD_DEV  /ALL
-----
XCFD  1439K  SYNC  1439K  100%  92.0  22.6  NO SCH  0  0.0%  0.0  0.0  0.0  REQ TOTAL  1410K
      797.4  ASYNC  0  0.0%  0.0  0.0  NO SCH  0  0.0%  0.0  0.0  0.0  REQ DEFERRED  61K
      CHNGD  0  0.0%  INCLUDED IN ASYNC  -CONT  61K
      -FALSE CONT  61K
-----
TOTAL  1439K  SYNC  1439K  100%  92.0  22.6  NO SCH  0  0.0%  0.0  0.0  0.0  REQ TOTAL  1410K
      797.4  ASYNC  0  0.0%  0.0  0.0  NO SCH  0  0.0%  0.0  0.0  0.0  REQ DEFERRED  61K
      CHNGD  0  0.0%  -CONT  61K

```

In the above excerpt from the RMF report, it is clear that there is a significant amount (4.3%) of false contention occurring during the reporting period. This is most likely due to the size of the structure, which, for this test was only 10Mb. This could be tuned (to improve response time) by increasing the size (hence the number of locks) in the lock structure.

NY Metro NaSPA®

© 2005 IBM Corporation

False contention is the main thing to keep your eye on. You can increase the size of the ISGLOCK lock structure, and its rebuild CF locations (more memory). You can do a dynamic operator initiated rebuild to increase the lock structure size.

z/OS Design and Development

GRS Star – Tuning GQSCAN/ISGQUERY

GRS

SYSPLEX wide GQSCAN/ISGQUERYs can

- be CPU intensive
- increase times

•Because it

- Requires an XCF message with response is sent to all systems in the sysplex in order to gather a sysplex view
- Needs to merge the results

•Keywords

- GQSCAN XSYS=YES
- ISGQUERY GATHERFROM=SYSPLEX

NY Metro NaSPA®

© 2005 IBM Corporation

STAR mode makes GQSCAN/ISGQUERY sysplex wide requests more cpu intensive and as a result can increase response times.

Setting GRSCNFxx to indicate GRSQ(CONTENTION) dramatically reduces the amount of the time GRS takes to collect SYSPLEX wide ENQ information for SDUMP which specify SDATA=GRSQ

z/OS Design and Development

GRS

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).
 - GRS shows who the contention notification system is
 - In a future release, IBM intends to provide the ability for the operator to move the CNS to a system of his choice.
- ENF 51s can be filtered for specific resources via installation exits ISGCNFXITSYSTEM and ISGCNFXITSYSPLEX

NY Metro NaSPA®

© 2005 IBM Corporation

Contention notification processing for SYSPLEX/SYSTEMS wide ENQs is required to be coordinated by a single system in the sysplex. This system designated system is called the contention notification system (CNS). Lots of contention can cause lots of processing as for each contention the following occurs:

1. An XCF message is sent to the CNS system from the system that detects contention.
2. The CNS issues a sysplex GQSCAN which results in XCF messages to and from every system in the sysplex.
3. The CNS then issues a sysplex wide ENF 51 which results in an XCF message being issue to every system in the sysplex.

Via installation exits ISGCNFXITSYSTEM and ISGCNFXITSYSPLEX, the installation can suppress ENF51s for specific resources. These do not show up on RMF reports but by filtering it is known to not be interesting. See APAR OW53323 or z/OS 1.4 installation exits details