Understanding GRS ENQ and Latch Usage and Contention

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

Date of Presentation: August 4, 2010

Session: 7409

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.

Agenda

- What's ENQ and Latch
- ENQ, Latch exploiters
- ENQ scalability
- Interfacing with alternate serialization
- Performance of ENQs, Latch, QScans, and SVC Dumps
- Differentiating IPCS GRSTRACE from GRSDATA
- Monitoring ENQ contention, what's available?
- Monitoring ENQ activity overall
- Differentiating Latches from ENQs
- Latch Identity crisis
- Latch deadlock detection
- Displaying ENQ & Latch contention

Contrasting ENQ with Latches

Latch Sets

1	
2	
x	

	ENQ	Latch
Scope	Job Step, System, GRS Complex	System via Primary space = creator's space
Resource ID	Qname,Rname, Scope	Latch Set + Latch number
Disposition	Share/Exclusive	Share/Exclusive
Authorization	Auth/Unauth	Auth
Usage	Widely used	Widely used
Performance	Reasonable	Very fast (orders of magnitude)
External Controls	RNLs,Exits, SRM ERV	SRM ERV
Backing Storage	GRS space	Creator's space + min GRS
Coding ease	Pretty easy, ASM only	Harder, but high level language
Work unit	Task only, xmem	Task/SRB, xmem

Who uses *the most* ENQs?

- The most *concurrently held* ENQs
  - Probably DB2 but shops vary
    - 100k data set support via DSMAX specification
    - Multiple ENQs per data set via Allocation, VSAM, etc. in DB2 address space
    - Triggering several discussion points…
      - Concurrent ENQ maximums per address space
      - GRS Virtual Storage Constraint Relief (VSCR)
      - Interfacing with alternate serialization
- The highest *frequency* of ENQs
  - Catalog
    - Qname SYSZVVDS, scope SYSTEMS
  - Allocation
    - Qname SYSZTIOT, scope SYSTEM, but serialized an address space TIOT
    - Qname SYSDSN, scope SYSTEM usually promoted to SYSTEMS

Many ENQs out there!

- The GRS EQDQ Monitor can be used to obtain the ENQ/Reserves issued within a period of time
- z/OS MVS Diagnosis Reference GA22-7588 ENQ/DEQ Summary table has QNAME/RNAME usage for many but not complete

Qname	Rname	Resource
SYSVSAM	dsncatnameL1L2L3*	VSAM data sets (dsn = data set name, catname = catalog name, L1 = RNAME length, L2 = data set name length, L3 = catalog name length, * = ENQ/DEQ control indicator). IDA0200T, IDA0231T, iDA0557A, IGG0CLBG
SYSVTOC	volser	volser
SYSDSN	Data set name	Data sets. Note: Normally issued under initiator TCB

Understanding ENQMAX

- What is ENQMAXA?
  - The maximum number of concurrent authorized ENQ requests that can be made within a single address space
  - Default is 250,000
- What is ENQMAXU?
  - The corresponding maximum for unauthorized ENQ requests with an address space
  - Default: 16,384
- What is the purpose of these maximums?
  - Protect against inadvertent looping of ENQ requests
  - ABEND538 results when a maximum is exceeded
  - Message ISG368E warns when maximum is being approached
  - This is NOT directly related to a GRS Virtual Storage Constraint

Adjusting ENQMAX

- Parmlib
  - GRSCNFxx: GRSDEF
    [ENQMAXA(*nnnnnnnn*)]
    [ENQMAXU(*nnnnnnnn*)]
  - Sets for all address spaces. Not recommended unless required.
  - Maximum value: 99,999,999
- System Command
  - SETGRS {ENQMAXA|ENQMAXU=*nnnnnnnn*}
    - Maximum value: 99,999,999
    - Sets for all address spaces. Not recommended unless required.
    - Update GRSCNFxx appropriately for next IPL
- System Service for those programs who know they are going to go higher than defaults
  - ISGADMIN REQUEST=SETENQMAX,
    MAXTYPE={AUTHORIZED|UNAUTHORIZED},MAXVALUE=maxvalue
  - Address-space-specific value (e.g. for use by DB2)
    - Greater of the address-space and system values used
    - Maintains ABEND538 ENQ loop protection for the rest of the system
  - Must be authorized (APF, system key, or supervisor state)

DB2 exploits ISGADMIN service to tailor its ENQMAX

- DB2 exploits ISGADMIN REQUEST=SETENQMAX
  - High concurrent maximum for DB2 related to DB2 DSMAX
  - Lower default system maximum for other address spaces
- Delivery date mismatch
  - ISGADMIN support delivered in z/OS R7 via OA11382
  - DB2 exploitation *to be enabled* with PM00068 (DB2 V8)
    - Temporary workaround is to adjust ENQMAXA for the system
  - DB2 customers have experienced ABEND538
    - Believed it to be Virtual Storage Constraint
    - While VSC no longer true for GRS, several years ago it was…

Who uses *the most* Latches?

- Unix System Services claims to be the winner with the most latches but who really knows!
- There are many latch exploiters:
  - z/OS
    - logger, RRS, APPC, SMB, IOS, WLM, GRS, VSAM, RACF, Consoles, you name it!
    - zFS and probably others use their own latching service
  - ISVs
  - Subsystems
    - DB2 uses it own latching service

What about latch restrictions?

- Very large number of latch sets can be supported
- CBs required to manage Latches are contained below the bar in the creator's space
- A latch set is not limited beyond the creator's storage limits
  - Latch set is fixed in size.. an array (big or small) base
    - Documented lset storage usage (header, 32 bytes per latch, etc.)
  - CBs representing waiters/holders are allocated as needed
  - The LowStorage create option reduces dynamic usage at the cost of performance in large MP environments
  - If private storage is constrained, GRS will automatically kick in the LowStgUsage but it might be too late i.e. spike.

ENQ scalability

- GRS has its own storage management subcomponent
  - Historically one huge block allocated during GRS initialization
  - Sub-divided into lots of cells of GRS-specific types
  - Various benefits to ENQ performance and GRS RAS
- What would happen if the storage ran out?
  - ABEND09A RC8108 for every subsequent ENQ request
  - Eventually the system would waitstate
    - Depending on the needs of the ENQ requester
- z/OS V1R9 most GRS blocks moved above the bar
- z/OS V1R11 QSCAN backing blocks moved from dataspace to above the bar

Misconceptions of ENQ alternate serialization

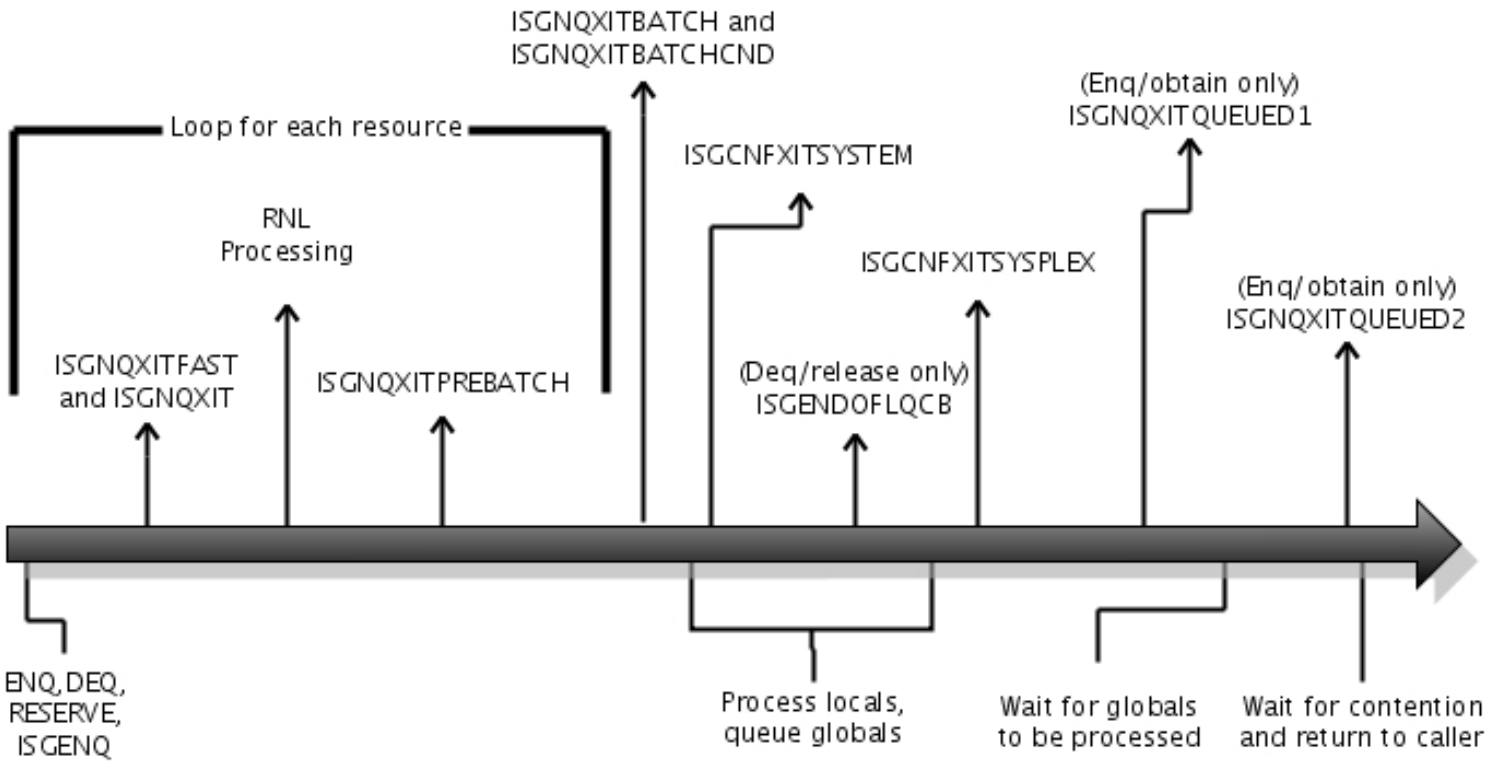
- GRS still active when alternate serialization used for ENQ
  - Same ENQ, DEQ, RESERVE, ISGENQ services used
  - Alternate serialization modifies results of GRS processing
    - GRS considers them local, alt. serialization considers them global
    - Partially configured via installation's Resource Name Lists (RNLs)
  - RNL=NO ENQ/ISGENQ requests still global to GRS
    - RNL and alternate serialization processing skipped
    - Designed primarily for sysplex global processing
  - GRS Star still best practice for RAS & performance

ENQ Installation Exits (Latch has none)

- Dynamic GRS installation exits of interest for the ENQ path:
  - ISGNQXIT — Adjusts Qname, Rname, Scope, etc
  - ISGNQXITFAST — Also adjusts ENQ (better performance)
  - ISGNQXITBATCH ENQ — ENQ info prior to queuing
- Dynamic GRS exits designed for alternate serialization:
  - ISGNQXITPREBATCH — Selects candidates for BATCHCND
  - ISGNQXITBATCHCND — ENQ info prior to queuing
  - ISGNQXITQUEUED1 — ENQ info after local queuing
  - ISGNQXITQUEUED2* — ENQ info after global queuing
  - ISGENQOFLQCB ENQ — Info when local resource gone

* New as of z/OS R11

ENQ Exit Flow

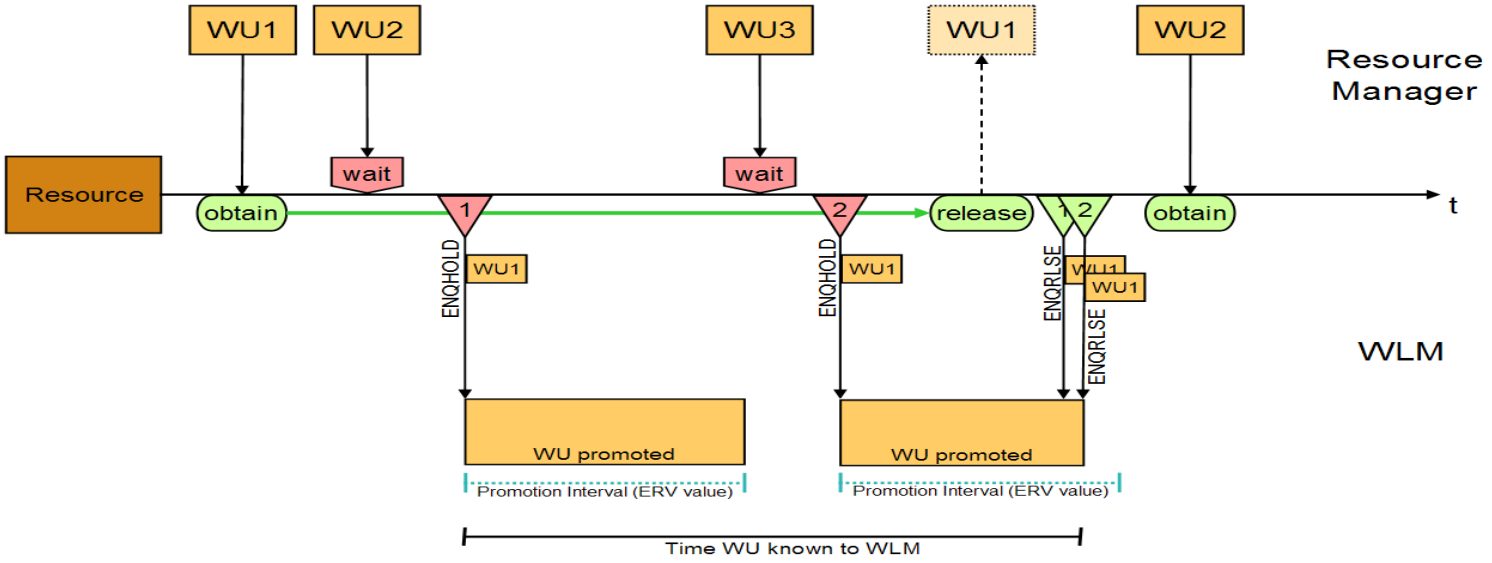


Contention… what happens?

1. Suspend the waiter and grant ownership when the blocker(s) release the resource
2. Promote the blocker, increase his dispatch priority, in order to get him to release the resource eventually
3. Tell others about the contention
   - For ENQ,
     - ENF 51 signals are issued IDing holder(s)/waiter(s)
     - Polling
       - *via QSCAN is also used by some monitors*
       - *D GRS,Analyze and D GRS,Contention by customers*
   - For Latch,
     - Signals are not generated
     - Polling via D GRS,Analyze and D GRS,Contention

WLM/SRM Enqueue Promotion for Latches/ENQs

- GRS informs WLM about holder of resources via Sysevent ENQHOLD
  - Every invocation of ENQHOLD restarts the promotion interval for the Work Unit (WU)
- At the end resource manager tells WLM that the resource is released (ENQRLSE)
  - ENQRLSEs must match ENQHOLDs
- Promotion Interval can be specified via ERV parameter (service units) in OPT parmlib member
- Purpose
  - Promotion works well
    - if the frequency of the requests is not too high
    - If the time the request needs is not too short and not too long



ENQ Contention Exits

- Dynamic GRS exits used in ENF 51 contention processing:
  - ISGCNFXITSYSTEM
  - ISGCNFXITSYSPLEX
- Originally designed to filter “contention noise”
  - Certain resources constantly in contention muddied reporting
- Updated in z/OS R11
  - Exit parameter lists now provide more contention data
  - Alternative to ENF 51 signals to monitor overall contention

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/Detailed for specific resource over 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

ENQ Contention–RMF Enqueue Activity Report example

```
                                    E N Q U E U E   A C T I V I T Y
z/OS V1R11                SYSTEM ID P30              DATE 04/07/2010            INTERVAL 10.21.780
                                  RPT VERSION V1R11 RMF      TIME 01.10.02              CYCLE 1.000 SECONDS

ENQUEUE DETAIL ACTIVITY                GRS MODE: STAR
-NAME-  ----- CONTENTION TIME -----  -- JOBS AT MAXIMUM CONTENTION--  -%QLEN DISTRIBUTION- AVG Q -REQUEST TYPE - TOTAL
MAJOR    MIN    MAX    TOT    AVG   ----- OWN ----- ----- WAIT ----    1    2    3    4+  LNGTH -EXCL-  -SHARE- EVENT
 MINOR                               TOT     NAME    TOT     NAME                                MIN MAX MIN MAX
                                          SYSNAME         SYSNAME
SYSIEFSD
 Q10
        0.000  0.015  1.641  0.001    1 ASMLJ096(E)   5 IAM1C038(E)  71.2 20.1  5.8  2.9   1.42   1   9   0   0  1157
                                          P30             P30
                                                         GNRC3088(E)
                                                          P30
SYSVTOC
 30PUB1
        0.000 10.983 180.52  0.027    1 PY3A186 (E)  65 PY2A184 (E)  32.9 16.2  9.8 41.2   6.73   1  72   0   0  6557
                                          P30             P30
                                                         PY3A185 (E)
                                                          P30
 30PUB3
        0.000 15.683 221.55  0.042    1 PY4B262 (E)  20 PY2B265 (E)  30.8 17.2 11.4 40.6   4.60   1  51   0   0  5223
                                          P30             P30
                                                         PY2B275 (E)
                                                          P30
```

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 Monitor II (SENQR)

- Monitor II takes snap shots
- System Enqueue Reserve Report (SENQR)
  - *SENQR {ALLVSER} | | {volser }*
  - EQDQ Monitor is better for monitoring Reserve activity

```
          RMF - SENQR System Enqueue Reserve Line 1 of 4
Command ===>                                  Scroll ===> HALF
          CPU= 37/ 35 UIC=2540 PR= 0    System= SYS1 Total
14:52:57 SYSTEM ENQUEUE RESERVE REPORT
JOBNAME ASID SYSTEM REQ VOLUME DEV  RSV MAJOR MINOR
CATALOG   34 AQTS   SO  TSO020 0AF1 CNV SYSIGGV2 CATALOG.VTSO020
BMORRISP  70 AQTS   EO  RES84Z 0D4E ON  SYSVTOC  RES84Z
JES2      31 AQTS   EO  JES2PK 0702 OFF SYSZJES2 JES2PKSYS1.BKUPCKPT
CATALOG   34 AQTS   SO  CAT212 04B1 OFF SYSZVVDS CAT212
```

ENQ Contention–RMF ENQR example

```
                    RMF V1R11  ENQ Resource Delays                Line 1 of 23
Command ===> _                                                 Scroll ===> CSR

Samples: 200     System: AQFT  Date: 06/09/10  Time: 14.00.00  Range: 200   Sec

----------- Resource Name -----------    ---- Delayed ----    ---- Holding ----
Major/Minor      (Scope)                  %  Name     STAT     %  Name/SYS STAT

SYSDSN           (SYSS)                   1 WOEM61T   EW       1 TPCR41T   EO
 MVSBUILD.PRDZLA.CSI                      1 ZOSMFBT   EW       1 XML110T   EO
                                                               /AQTS
                                          1 ZOSMFCT   EW       1 XML110T   EO
                                                               /AQTS
                                          1 XML110T   EW       1 WOEM70T   EO
SYSDSN           (SYSS)                 100 ZART16CM  EW     100 JGENTIL   SO
 D46GRS.A2.ART.CMDLOG                   100 ZART20CM  EW     100 JGENTIL   SO
                                        100 ZSART2CM  EW     100 JGENTIL   SO
CLRSHARE         (SYSS)                   1 ISGNAINT  EW       1 ISGQPC    EO
 B390.HBB7780.BROOKER5.UNIMODC                                 /AQTS
                                          1 ISGNASIM  EW       1 ISGGTRM1  EO
                                                               /AQTS
IGWLHANZ         (SYS)                    9 SMSPDSE   EW       9 SMSPDSE1  EO
 SERIALIZEPDSEANALYSIS
SYSZCBR          (SYSS)                   1 OAM       EW       1 OAM       EO
 CARTRIDGE_ENTRY_P710TATL                                      /AQTS
CLRSHARE         (SYSS)                   1 ISGNAINT  EW       1 ISGQPC    EO
 WHRUSED.B390.HBB7780.BROOKER5                                 /AQTS
SYSDSN           (SYSS)                  19 ISGNAINT  EW      19 ISGQPC    EO
 B390.HBB7780.BROOKER5.BKPTADAT                                /AQTS
KLVGLOCK         (SYSS)                  13 CANSDSST  EW      13 CANSDSST  EO
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=TOGGLE
 F7=UP        F8=DOWN      F9=SWAP     F10=BREF     F11=FREF     F12=RETRIEVE
```

ENQ Contention–RMF SYSENQ example

```
                    RMF V1R11  Sysplex ENQ Delays - MCLXCF01        Line 1 of 27
Command ===> _                                                  Scroll ===> CSR

Samples: 200     Systems: 2    Date: 06/09/10  Time: 14.00.00  Range: 200   Sec

------ Resource Name ------ ------- Delayed --------  ------- Holding --------
Major/Minor                   %  Jobname  Sys-Name ST   %  Jobname  Sys-Name ST

SYSDSN                      100 ZSART2CM AQFT     EW  100 JGENTIL  AQFT     SO
 D46GRS.A2.ART.CMDLOG       100 ZART20CM AQFT     EW  100 JGENTIL  AQFT     SO
                            100 ZART16CM AQFT     EW  100 JGENTIL  AQFT     SO
                            100 ZART23CM AQTS     EW  100 JGENTIL  AQFT     SO
                            100 ZART19CM AQTS     EW  100 JGENTIL  AQFT     SO
                            100 ZART22CM AQTS     EW  100 JGENTIL  AQFT     SO
                            100 ZART21CM AQTS     EW  100 JGENTIL  AQFT     SO
SYSDSN                        1 XML110T  AQFT     EW    1 WOEM70T  AQFT     EO
 MVSBUILD.PRDZLA.CSI          1 ZOSMFBT  AQFT     EW    1 XML110T  AQTS     EO
                              1 ZOSMFCT  AQFT     EW    1 XML110T  AQTS     EO
                              1 WOEM61T  AQFT     EW    1 TPCR41T  AQFT     EO
                              1 TROYDCF  AQTS     SW    1 WOEM70T  AQTS     EO
                              1 TPCR41T  AQTS     EW    1 PUT110T  AQFT     EO
                              1 ZOSMFBT  AQTS     EW    1 WOEM70T  AQFT     EO
CLRSHARE                      2 ISGGTRM1 AQTS     EW    2 ISGNASIM AQFT     EO
 B390.HBB7780.BROOKER5.UNIM   2 ISGQPC   AQTS     EW    1 ISGNAINT AQFT     SO
                                                      1 ISGNAINT AQFT     EO
                              1 ISGNASIM AQFT     EW    1 ISGGTRM1 AQTS     EO
                              1 ISGNAINT AQFT     EW    1 ISGQPC   AQTS     EO
KLVGLOCK                     13 CANSDSST AQFT     EW   13 CANSDSST AQTS     EO
 XPRODPLE
CLRSHARE                      1 ISGNAINT AQFT     EW    1 ISGQPC   AQTS     EO
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=TOGGLE
 F7=UP        F8=DOWN      F9=SWAP     F10=BREF     F11=FREF     F12=RETRIEVE
```

Holder Notifications to TSO (ISGECMON) and Operator for Batch (ISGECJES)

- ISGECMON (Samplib) / ISGECJES (Advanced Technical Support)
  - **This program runs as a never ending task that checks dataset contention at periodic intervals, and sends messages to TSO users and/or system operators for batch jobs causing contention.**

1 - Issue notification messages either to the operator or
the TSO user for the following conditions for a resource:

OWN	WAIT	
TSO/E	BATCH	
BATCH	BATCH	(ISGECJES Only)

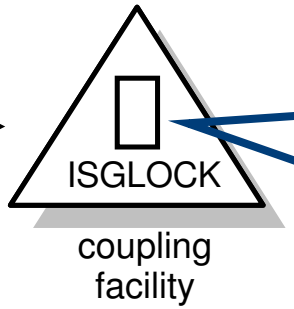
The following contention situation can be handled
by ISPF. Enter the PF1 help key (twice) for enqueue
contention involving only TSO/E users:

OWN	WAIT
TSO/E	TSO/E

False contention – a STAR Issue!

No contention

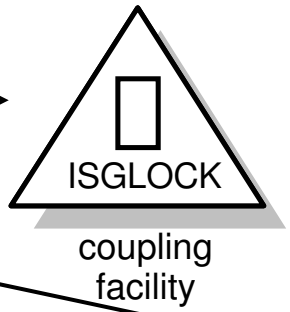
1) Quick synch CF trip



Hash of QNAME/RNAME using structure size determines **lock structure entry,** more entries, less false contention

Contention

1) Quick synch CF trip
2) Wait for long trip to/from Global Manager as has sysplex view
3) If true contention, wait for grant



XCF msg via XES

IXLLOxxx group

XES Global management

False cont?

-No, Call Global Manager

-Yes, let requester go

False contention is wasteful, check for >2% and increase the ISGLOCK appropriately

GRS Star – Tuning/False contention

```
STRUCTURE NAME = ISGLOCK           TYPE = LOCK
          # REQ        ---------- REQUESTS -------------   -------------- DELAYED REQUESTS -------------
SYSTEM    TOTAL              #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----  EXTERNAL REQUEST
NAME      AVG/SEC            REQ    ALL     AVG   STD_DEV            REQ    REQ   /DEL    STD_DEV   /ALL   CONTENTIONS

XCFD       1439K   SYNC   1439K   100%    92.0     22.6                                                      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                                                      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 (61/1410 = 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.

GRS latch contention stats

- IPCS GRSDATA Latch Stats are buried at the bottom:

Latch Statistics

Latch Set Name: LS1.XMITDAT.LATCH.SET

Creator Jobname: MYJOB1

Creator ASID: 0024

Latch Number	Fast Obtains	Slow Obtains	Ratio (slow/total)
0	320	1	00.31%
1	209,989	33	00.02%
2	5,530,998	3,294,036	37.33%
3	3 611,721	24,967	03.92%
4	211,574	11,987	05.36% *
Summary:	6,564,602	3,341,024	33.73% *

Total number of latches in above latch set: 5

Number of latches with non-zero statistics: 5

z/OS MVS Diagnosis Reference Guide – Latch usage & Contention info

- Latch usage and contention handling procedures:
  - Unix System Services –
    - Related command
    - Mount
    - Physical file system (PFS)
    - other latches
  - Logger –
    - Miscellaneous latches
    - Structure and task latches
    - Log stream latches
    - Uses latch idendity to provide logger specific info about the latch

ENQ & QScan (GQSCAN/ISGQUERY) performance relationship

- **General rules of thumb regarding GRS modes**
  - In Star, ENQs are “very fast” and QScans are “slower”
    - CF lock structure centralizes global contention management
    - Each system on a need-to-know basis for global resources
      - *Therefore GQScan processing needs to gather data remotely*
  - In Ring, ENQs are “much slower” and QScans are “faster”
    - RSA propagated around every system in the Ring
    - Every system knows everything about global resources
      - *Therefore GQScan processing always has the data locally*
    - Ring runs best in a base sysplex
      - *Non-Sysplex relies on GRS to do system monitoring.*
      - *Many RAS operational hardships*
      - *1980s technology!*

- **IBM recommends GRS Star**
  - Improved global ENQ performance
    - Contention is not that frequent
    - Slow QSCANs are not that common
  - Improved scalability with regard to additional systems
  - Better RAS and recovery times – no “ring disruptions”
  - Even when 3rd party serialization product is present

QScan performance affects SVC Dumps in STAR mode

- SVC Dumps with SDATA=GRSQ internally initiate a QScan
  - This data is the basis for IPCS GRSDATA subcommand
  - Defaulted to gathering all global data across Sysplex
    - In Star mode, need to query all remote systems
  - Larger system images mean more ENQs, longer dump times
- GRSQ setting in GRSCNFxx
  - GRSQ adjustable through SETGRS system command
  - GRSQ(ALL) requests all ENQ data
  - GRSQ(LOCAL) requests just local ENQ data
  - GRSQ(CONTENTION) requests data for ENQs in contention
    - This is the default
    - In certain error scenarios, GRSQ(ALL) may still be necessary

SVC Dumps and SDATA=GRSQ

- z/OS V1R12 – GRS lets SDUMP/RSM capture data
  - Big performance improvement
  - Data moved direct to dump capture area rather than staging area then sdump area, then dump capture area
  - Paged out data paged into dump capture area rather than GRS first
    - Reduces demand on real storage
- z/OS V1R11 - Compression of free queues assists storage constrained systems
  - Paging in of free elements impacted performance
  - Rolled down via APAR OA29329 to V1R9
  - APAR OA32993 to suppression compression msgs

Contention monitoring: Understanding ENF 51 signals

- Event code 51 is specific to GRS
  - ISGE51CN maps signals for contention
  - ISGE51RN maps signals for RNL suspension
- ENF Qualifier's 4 bytes distinguish the event
  - 1st byte: 01x for contention, 02x for RNLs, 03x for mode change
  - 2nd byte: currently unused
  - 3rd byte: 00x for traditional contention, 01x for "waitless"*
  - 4th byte: 01x local contention, 02x global, 03x for recovery
- ENF 51 is input to RMF Monitoring
- ENF 51 is also input to NFS for data set delegation*

* New as of z/OS R11

Contention monitoring: Another internal QScan

- ENF 51 information results from an internal QScan
  - Variable data portions are mapped by ISGRIB
  - QScan is asynchronous from when contention was detected
    - Contention may resolve before QScan completes
- QScan polling is discouraged
  - Contention monitoring through separate QScans is an inefficient use of system resources
    - QScan polling is a common reason for high GRS cpu utilization

Setting the CNS

- For GRS Star, there is a Contention Notification System
  - All global resource contention events are routed to it
  - All ENF 51 signals for contention are broadcast from it
- The CNS is arbitrarily chosen by GRS
  - There may be rare instances where the CNS task fails
    - An ENF 51 recovery-related signal is issued
    - The task will be re-established but contention data may be lost
  - Installations may override the choice of CNS
    - Often the largest system is chosen to best handle the workload
    - SETGRS CNS=systemname
    - COMMNDxx can be used for this SETGRS at IPL-time
    - DISPLAY GRS will output the CNS with other system information

Monitoring ENQ activity

- GRS EQDQ Monitor
  - Monitoring activity is different than monitoring contention
  - Used for planning and confirming RNL updates
  - Measuring ENQ & RESERVE activity
  - More information available in the GRS Planning Guide
- Monitor input from the ISGNQXITBATCH exit point
  - Unlike ISGNQXITBATCHCND used by alternate serialization
    - ISGNQXITBATCH gets control for EVERY ENQ/DEQ request
  - Timed just prior to queuing, local or global
  - Provides lots of data regarding the ENQ request

Monitoring unwanted Reserves

- Filter REQTYPE=NCRESERVE
  - Outputs non-converted Reserve requests
  - Useful in planning RNL conversion entries
- Potential problems with RESERVE
  - Lack of granularity
  - Starvation
  - Volume hierarchy deadlock
  - Asynchronous I/O deadlock (alleviated via Synchres)
  - Double serialization deadlock
  - Data integrity exposure upon system reset
  - Compounding problems for more than one of the above

EQDQ Monitor output example

```
ENQ/DEQ Monitor - Major Name List     Row 1 to 23 of 132

Enter S to select a Major Name for details     .
      L major on command line to locate a Major.   Elapsed seconds:   39801
```

Sel. Field	Major Name	Scope	Exit	RNL	**Counter**	-Average- msec	-Reserved- seconds
_	SYSZVVDS	*RES		EXCL	720406	16	11952
_	SYSVTOC	*RES		EXCL	59982	54	3344
_	SYSIGGV2	*RES		EXCL	208156	2	577
_	STKALSQN	*RES		EXCL	10477	15	165
_	GFSZTQS	*RES			156	833	130
_	NMUAMS	*RES			804	6	4
_	ARCBACK	*RES			608	0	0
_	ARCAUDIT	*RES			615	0	0
_	ARCUPDT	*RES			610	0	0
_	ARCGPAL	*RES			613	0	0
_	STROBEHA	*RES			59	5	0
_	STROBECV	*RES			53	5	0
_	ARCDSN	SYSS			2137		
_	ARCGPA	*SYSS		CONV	11269		
_	ARCENQG	SYSS			26799		
_	BBIUSER	SYSS			2		
_	NDMUPDNM	SYSS			338		

The small list of reasons to keep a RESERVE

- Resource shared outside the GRS complex
- Inconsistent Qname/Rname used on the ENQ
  - Reserve could be masking a bad programming practice
- Reserve serializes many resources
  - For example, all ENQs on the same volume
- Ring performance sensitivity
  - Reserve can be faster than a global ENQ in Ring mode
  - Reaffirms the recommendation to use GRS Star

Displaying contention: A couple options

- D GRS,CONTENTION
  - Simple display of ENQ & Latch resources in contention
- D GRS,ANALYZE
  - Provides a more comprehensive contention picture, by…
    - Blocker
    - Waiter
    - Dependency
  - As of z/OS R11, includes Latch support!
  - IBM recommends this method

Latch Identity crisis

- Latches are organized by numbers, not names
  - Latch Set created specifying the number of latches
  - Latch request made by number
  - Can be difficult for serviceability
- As of z/OS R11, latches can have identities!
  - ISGLID callable service
    - ISGLID64 callable service for 64-bit callers
    - Latch Identity pointer array for the Latch Set
  - Early exploiters: Logger, RRS, Unix System Services (R12)

Latch self deadlock detection

- Minimal user error checking theme
  - Authorized users expected to “know what they’re doing”
  - Requesting a latch that results in contention with one that was already owned by your unit of work causes deadlock
- There are now several levels of deadlock detection
  - Optionally specified on Latch Set creation (ISGLCRT)
  - Quickest option only checks against exclusive ownership case
  - Second option runs the chain for shared ownership cases
  - ABEND9C6 RSN xxxx002E, xxxx2F, or xxxx36
- But there was a bug in the detection!
  - OA32000 fixes valid latch request was erroneously ABENDed by this function

Another look at D GRS,ANALYZE

- D GRS,ANALYZE now for both ENQ and LATCH
  - Does NOT analyze both ENQs and Latches together
  - Supports Blocker, Waiter, & Dependency for either type
  - Supports Summary & Detail for either type
  - Latch Identities displayed for LATCH if available

Sample Syntax

DISPLAY GRS,ANALYZE,LATCH,DEPENDENCY

{ [,ASID=*asid*] | [,JOBNAME=*jobname*] |
  [,XJOBNAME=(*xjobnamelist*)]
  [,TCB|WEB=*workunitaddr*]
 [,XLSETNM=(*xlsetnamelist*)]
 [,CASID=*cr-asid*] | [,CJOBNAME=*cr-jobname*]}
 [,LAT=(*lsetname,latchnum*)]
 [,COUNT|CNT=nn]
 [,DETAIL|DET]
}

DISPLAY GRS,ANALYZE,LATCH, DEPENDENCY output format

```
DEPENDENCY ANALYSIS:  request specification desc.
  WAITTIME  JOBNAME  E/S  CASID LSETNAME/LATCHID
  hh:mm:ss  jobname  *r*  casid lsetname
                                latchId
  BLOCKER   jobname2  r2
  ANALYSIS ENDED: analysis outcome
```

```
D GRS,AN,LATCH,DEPENDENCY,DETAIL
ISG374I 09.55.25 GRS ANALYSIS 134
DEPENDENCY ANALYSIS: ENTIRE SYSTEM
----- LONG WAITER #1
         JOBNAME: SVRASID2  (ASID=002A, TCB=004E68E0)
         REQUEST: EXCLUSIVE                        LT:7F52B07800
WAITING  00:00:03 FOR RESOURCE (CREATOR ASID=0025)
A1T1L1N4                                          LST:7F52FC0000
0:(ID NOT SPECIFIED)
         JOBNAME: SVRASID1  (ASID=0025, TCB=004E6B70)
         REQUEST: SHARED                           LT:7F52B01000
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
----- LONG WAITER #2
         JOBNAME: SVRASID3  (ASID=0029, TCB=004E6650)
         REQUEST: EXCLUSIVE                        LT:7F52B07800
WAITING  00:00:01 FOR RESOURCE (CREATOR ASID=002A)
A2T2L2N4                                          LST:7F52E80000
1:(ID NOT SPECIFIED)
         JOBNAME: SVRASID2  (ASID=002A, TCB=004E68E0)
         REQUEST: SHARED                           LT:7F52B01000
WAITING  00:00:03 FOR RESOURCE (CREATOR ASID=0025)
A1T1L1N4                                          LST:7F52FC0000
0:(ID NOT SPECIFIED)
         JOBNAME: SVRASID1  (ASID=0025, TCB=004E6B70)
         REQUEST: SHARED                           LT:7F52B01000
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
```

Differentiating IPCS GRSTRACE from GRSDATA

- GRSDATA subcommand based on QScan output
  - Affected by GRSQ setting of ALL, LOCAL, or CONTENTION
- VERBEXIT GRSTRACE based on in-storage control blocks
  - Affected by the GRS mode
    - GRS Ring will have all global data
    - GRS Star will only have global data relevant to the current system

Filtering both GRSTRACE and GRSDATA

- Larger systems, more data
  - IPCS formatting of all GRS ENQ data can take hours
  - Reviewing irrelevant data increases diagnosis time
- Filters added in z/OS R10
  - ISPF panel support as well
- Filters include:
  - Time range
  - Qname
  - Rname
  - Jobname
  - ASID
  - Scope
  - Contention
  - Reserve
  - TCB@

IPCS Component Data Analysis panel

```
------------------ IPCS MVS DUMP COMPONENT DATA ANALYSIS ---------------

To display information, specify "S option name" or enter S to the left
of the option desired.  Enter ? to the left of an option to display
help regarding the component support.

S Name     Abstract
_ ALCWAIT  Allocation wait summary
_ AOMDATA  AOM analysis
_ APPCDATA APPC/MVS Data Analysis
_ ASCHDATA APPC/MVS Scheduler Data Analysis
_ ASMCHECK Auxiliary storage paging activity
_ ASMDATA  ASM control block analysis
_ AVMDATA  AVM control block analysis
_ COMCHECK Operator communications data
_ COUPLE   XCF Coupling analysis
_ CSFDATA  ICSF control block analysis
_ CTRACE   Component trace summary
_ DAEDATA  DAE header data
_ DB2DATA  DB2 analysis
_ DIVDATA  Data in virtual storage
_ DLFDATA  Data Lookaside Facility data
_ DLFTRACE Data Lookaside Facility trace
S GRSDATA  GRS managed resources
_ IMSDUMP  IMS analysis
_ IOSCHECK Active input/output requests
```

IPCS GRSDATA panel

```
------------------------- IPCS - GRSDATA SUBCOMMAND -------------- Enter option
SELECT OPTION ===>
 Select a report type. The default is the GRSDATA report type.
   * GRSDATA          _ GRSTRACE
 Select a level of detail. The default is SUMMARY reporting.
   * SUMMARY          _ DETAIL (GRSTRACE only)
 Select the time format to use for the GRSTRACE report. The default is LOCAL.
   _ LOCAL            _ GMT          _ UTC
 Select zero or more filtering options. The default is NO filtering.
 Filters that do not apply to a given report will be ignored.
   SYSNAME  SY1_____   JOBNAME  ________   ASID x' ____ '   TCB x' ________ '
   QNAME    MYQNAME_
   RNAME    ________________________________________________
   SCOPE:    _ STEP     _ SYSTEM     _ SYSTEMS
   _ CONTENTION        _ RESERVE
   START TIME MM/DD/YY,HH:MM:SS.DDDDDD  STOP TIME MM/DD/YY,HH:MM:SS.DDDDDD

 GRSDATA SUMMARY SYSNAME('SY1') QNAME('MYQNAME')

  S = START  selected report.
  R = Reset all panel variables.
  END = Exit GRSDATA panel.
```

Sample GRSTRACE summary

```
MAJOR NAME: TESTENQ

*   MINOR NAME: DUMMYENQ
      SCOPE: SYSTEMS   SYSNAME: S1        STATUS: *SHARED*    /OWN
         ASID: 0000002C   TCB: 006FF020    JOBNAME: GRSTOOL
         Critical ENQ Time(s):
            Request:     06/04/2007 15:30:05.804018
            Grant:       06/04/2007 15:30:05.834250
      SCOPE: SYSTEMS   SYSNAME: S1        STATUS: *SHARED*    /OWN
         ASID: 00000028   TCB: 006FF020    JOBNAME: GRSTOOL
        Critical ENQ Time(s):
           Request:     06/04/2007 15:32:18.460284
           Contention:  06/04/2007 15:32:18.484524
           Grant:       06/04/2007 15:32:34.846436
           Delta Time Waiting: 00:00:16.361911
      SCOPE: SYSTEMS   SYSNAME: S2        STATUS: *EXCLUSIVE* /WAIT
        ASID: 0000002F   TCB: 006FF020    JOBNAME: GRSTOOL
        Critical ENQ Time(s):
          Request:     06/04/2007 15:33:18.738913
        Some ENQ information is unavailable for this remote request
```

```
MAJOR NAME: xmajorname
    MINOR NAME: xminorname
         Resource Creation Time: xdate xtime
         Last Movewaiter Time:   xdate xtime
      SCOPE: xscope    SYSNAME: xsysname  STATUS: xstatus
         ASID: xasid      TCB: xtcb        JOBNAME: xjobname
         MASID: xmasid    MTCB: xmtcb
         Reserve Device: xdevice Volser: xvolser
         Critical ENQ Time(s):
           Request:     xdate xtime
           Contention:  xdate xtime
           Grant:       xdate xtime
           Delta Time Waiting: xdeltatime
           Movewaiter:  xdate xtime
        Caller PSW: xpsw              Caller TCB: xcallertcb
        Request Type: xrequesttype
        RNL Processing Actions: xrnlactions
        Affected by ISGNQXIT/FAST
        Affected by ISGNQXITBATCH/CND
        Managed by an Alternate Serialization Product
        ISGENQ Userdata:
          xuserdata
        QEL: xqeladdr                QXB: xqxbaddr
        QCB: xqcbaddr               SVRB: xsvrbaddr
```

Detail information is in green

Appendix of z/OS V1R12 Publications

- MVS Authorized Assembler Services Guide
- MVS Authorized Assembler Services Reference
- MVS Callable Services
- MVS Diagnosis: Reference
- MVS Initialization and Tuning Reference
- MVS Installation Exits
- MVS Planning: Global Resource Serialization
- MVS System Commands

Questions?