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
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:

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.

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.

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

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

<table>
<thead>
<tr>
<th>Qname</th>
<th>Rname</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSVSAM</td>
<td>dsncatnameL1L2L3*</td>
<td>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</td>
</tr>
<tr>
<td>SYSVTOC</td>
<td>volser</td>
<td>volser</td>
</tr>
<tr>
<td>SYSDSN</td>
<td>Data set name</td>
<td>Data sets. Note: Normally issued under initiator TCB</td>
</tr>
</tbody>
</table>
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:
    - [ENQMAXA(nnnnnnnnn)]
    - [ENQMAXU(nnnnnnnnn)]
  - Sets for all address spaces. Not recommended unless required.
  - Maximum value: 99,999,999
- System Command
  - SETGRS {ENQMAXA|ENQMAXU=nnnnnnnnn}
    - 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 its 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

1. Loop for each resource
   - ISGQXMTR and ISGQXIT
   - RNL Processing
   - ISGQXITPREBATCH

2. Process locals, queue globals

3. Wait for globals to be processed

4. Wait for contention and return to caller

5. (Enq/obtain only) ISGQXITQUEUED1
6. (Enq/obtain only) ISGQXITQUEUED2
7. ISGQXITSYSPLex
8. ISGQXITSYSTEM
9. ISGQXITBATCHCND
10. ISGQXITBATCH

ENQ, DEQ, RESERVE, ISQGENQ
Contestation… 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

ENQUEUE ACTIVITY

<table>
<thead>
<tr>
<th>Sysname</th>
<th>Name</th>
<th>Contention Time</th>
<th>Jobs at Maximum Contention</th>
<th>%QLEN Distribution</th>
<th>Avg Q Length</th>
<th>Request Type</th>
<th>Total</th>
<th>Name</th>
<th>Total</th>
<th>Name</th>
<th>Sysname</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASMLJ096</td>
<td>1.641</td>
<td>IAM1C038(E)</td>
<td>1.42</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>1157</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PY3A186</td>
<td>180.52</td>
<td>PY2A184(E)</td>
<td>6.73</td>
<td>1</td>
<td>72</td>
<td>0</td>
<td>6557</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PY4B262</td>
<td>221.55</td>
<td>PY2B265(E)</td>
<td>4.60</td>
<td>1</td>
<td>51</td>
<td>0</td>
<td>5223</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sysname**: SYSIEFSD

**Sysname**: SYSTOC

**Sysname**: PUB1

**Sysname**: PUB3
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}
  - ECDQ 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

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>ASID</th>
<th>SYSTEM</th>
<th>REQ</th>
<th>VOLUME</th>
<th>DEV</th>
<th>RSV</th>
<th>MAJOR</th>
<th>MINOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG</td>
<td>34</td>
<td>AQTS</td>
<td>SO</td>
<td>TSO020</td>
<td>0AF1</td>
<td>CNV</td>
<td>SYSIGGV2</td>
<td>CATALOG.VTSO020</td>
</tr>
<tr>
<td>BMORRISP</td>
<td>70</td>
<td>AQTS</td>
<td>EO</td>
<td>RES84Z</td>
<td>0D4E</td>
<td>ON</td>
<td>SYSVTOC</td>
<td>RES84Z</td>
</tr>
<tr>
<td>JES2</td>
<td>31</td>
<td>AQTS</td>
<td>EO</td>
<td>JES2PK</td>
<td>0702</td>
<td>OFF</td>
<td>SYSZJES2</td>
<td>JES2PKSYS1.BKUPCKPT</td>
</tr>
<tr>
<td>CATALOG</td>
<td>34</td>
<td>AQTS</td>
<td>SO</td>
<td>CAT212</td>
<td>04B1</td>
<td>OFF</td>
<td>SYSZVVDS</td>
<td>CAT212</td>
</tr>
</tbody>
</table>
```
## ENQ Contention–RMF ENQR example

### RMF V1R11 ENQ Resource Delays

**Command ==> _**

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

<table>
<thead>
<tr>
<th>Major/Minor (Scope)</th>
<th>% Delayed</th>
<th>Name</th>
<th>STAT</th>
<th>% Holding</th>
<th>Name/SYS</th>
<th>STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSDSN (SYSS)</td>
<td>1</td>
<td>WOEM61T</td>
<td>EW</td>
<td>1</td>
<td>TPCR41T</td>
<td>EO</td>
</tr>
<tr>
<td>MVSBUILD.PRDZLA.CSI</td>
<td></td>
<td>ZOSMFBT</td>
<td>EW</td>
<td></td>
<td>/AQT5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ZOSMFCT</td>
<td>EW</td>
<td></td>
<td>/AQT5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XML110T</td>
<td>EW</td>
<td>1</td>
<td>XML110T</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>XML110T</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WXV80T</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>XML110T</td>
<td>EW</td>
<td></td>
<td>WXV80T</td>
<td></td>
</tr>
<tr>
<td>SYSDSN (SYSS)</td>
<td>100</td>
<td>ZART16CM</td>
<td>EW</td>
<td>100</td>
<td>JGENTIL</td>
<td>SQ</td>
</tr>
<tr>
<td>D46GRS.A2.ART.CMDLOG</td>
<td></td>
<td>ZART20CM</td>
<td>EW</td>
<td>100</td>
<td>JGENTIL</td>
<td>SQ</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>ZSART2CM</td>
<td>EW</td>
<td>100</td>
<td>JGENTIL</td>
<td>SQ</td>
</tr>
<tr>
<td>CLRSHARE (SYSS)</td>
<td>1</td>
<td>ISGNAINT</td>
<td>EW</td>
<td>1</td>
<td>ISGQPC</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B390.HBB7780.BROOKER5.UNIMODC</td>
<td>1</td>
<td>ISGNASIM</td>
<td>EW</td>
<td>1</td>
<td>ISGGTRM1</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>SMSPDSE</td>
<td>EW</td>
<td>9</td>
<td>SMSPDSE1</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>OAM</td>
<td>EW</td>
<td>1</td>
<td>OAM</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ISGNAINT</td>
<td>EW</td>
<td>1</td>
<td>ISGQPC</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>ISGNAINT</td>
<td>EW</td>
<td>19</td>
<td>ISGQPC</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>/AQT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>CANSDSST</td>
<td>EW</td>
<td>13</td>
<td>CANSDSST</td>
<td>EO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**F1=HELP  F2=SPLIT  F3=END  F4=RETURN  F5=RFIND  F6=TOGGLE  F7=UP  F8=DOWN  F9=SWAP  F10=BRER  F11=FREF  F12=RETRIEVE**
## ENQ Contention–RMF SYSENQ example

<table>
<thead>
<tr>
<th>Major/Minor</th>
<th>Resource Name</th>
<th>Delayed</th>
<th>Holding</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSDSN</td>
<td>ZSART2CM AQFT</td>
<td>EW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td>DYERGRS.A2.ART.CMDLOG</td>
<td>ZSART2CM AQFT</td>
<td>EW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td></td>
<td>ZSART16CM AQFT</td>
<td>EW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td></td>
<td>ZART23CM AQTS</td>
<td>SW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td></td>
<td>ZART9CM AQTS</td>
<td>SW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td></td>
<td>ZART22CM AQTS</td>
<td>SW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td></td>
<td>ZART21CM AQTS</td>
<td>SW</td>
<td>JGENTIL AQFT</td>
</tr>
<tr>
<td>SYSDSN</td>
<td>XML110T AQFT</td>
<td>EW</td>
<td>WOEM70T AQFT</td>
</tr>
<tr>
<td>MVSBUILD.PRDZLA.CSI</td>
<td>XML110T AQTS</td>
<td>SW</td>
<td>WOEM70T AQTS</td>
</tr>
<tr>
<td></td>
<td>ZOSMF8T AQFT</td>
<td>EW</td>
<td>XML110T AQTS</td>
</tr>
<tr>
<td></td>
<td>ZOSMFC GFT AQFT</td>
<td>EW</td>
<td>XML110T AQTS</td>
</tr>
<tr>
<td></td>
<td>WOEM61T AQFT</td>
<td>EW</td>
<td>TPCR41T AQFT</td>
</tr>
<tr>
<td></td>
<td>TROYDCF AQTS</td>
<td>SW</td>
<td>WOEM70T AQTS</td>
</tr>
<tr>
<td></td>
<td>TPCR41T AQTS</td>
<td>EW</td>
<td>WOEM70T AQFT</td>
</tr>
<tr>
<td>CLRSHARE</td>
<td>ISGGRM1 AQTS</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td>B390.HBB7780.BROOKER5.UNIM</td>
<td>ISGQPC AQTS</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td></td>
<td>ISGQPC AQTS</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td></td>
<td>ISGNASIM AQFT</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td></td>
<td>ISGNASIM AQFT</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td></td>
<td>ISGNASIM AQFT</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td></td>
<td>ISGNASIM AQFT</td>
<td>EW</td>
<td>ISGNASIM AQFT</td>
</tr>
<tr>
<td></td>
<td>CANSDSS1 AQFT</td>
<td>EW</td>
<td>CANSDSS1 AQFT</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>OWN</th>
<th>WAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO/E</td>
<td>BATCH</td>
</tr>
<tr>
<td>BATCH</td>
<td>BATCH</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>OWN</th>
<th>WAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO/E</td>
<td>TSO/E</td>
</tr>
</tbody>
</table>
False contention – a STAR Issue!

No contention

1) Quick synch CF trip

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

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

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

XES Global management

False cont?
- No, Call Global Manager
- Yes, let requester go
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 identity 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
  • 1\textsuperscript{st} byte: 01x for contention, 02x for RNLs, 03x for mode change
  • 2\textsuperscript{nd} byte: currently unused
  • 3\textsuperscript{rd} byte: 00x for traditional contention, 01x for “waitless”\textsuperscript{*}
  • 4\textsuperscript{th} 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\textsuperscript{*}

\textsuperscript{*} 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

<table>
<thead>
<tr>
<th>Sel.</th>
<th>Major Name</th>
<th>Scope</th>
<th>Exit</th>
<th>RNL</th>
<th>Counter</th>
<th>msec</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>SYSZVVDS</td>
<td>*RES</td>
<td>EXCL</td>
<td>720406</td>
<td>16</td>
<td>11952</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>SYSVTOC</td>
<td>*RES</td>
<td>EXCL</td>
<td>59982</td>
<td>54</td>
<td>3344</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>SYSIGGV2</td>
<td>*RES</td>
<td>EXCL</td>
<td>208156</td>
<td>2</td>
<td>577</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>STKALSQN</td>
<td>*RES</td>
<td>EXCL</td>
<td>10477</td>
<td>15</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>GFSZTQS</td>
<td>*RES</td>
<td></td>
<td>156</td>
<td>833</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>NMUAMS</td>
<td>*RES</td>
<td></td>
<td>804</td>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCBACK</td>
<td>*RES</td>
<td></td>
<td>608</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCAUDIT</td>
<td>*RES</td>
<td></td>
<td>615</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCUPDT</td>
<td>*RES</td>
<td></td>
<td>610</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCGPAL</td>
<td>*RES</td>
<td></td>
<td>613</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>STROBEHA</td>
<td>*RES</td>
<td></td>
<td>59</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>STROBECV</td>
<td>*RES</td>
<td></td>
<td>53</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCDSN</td>
<td>SYSS</td>
<td></td>
<td>2137</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCGPA</td>
<td>*SYSS</td>
<td>CONV</td>
<td>11269</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>ARCENQG</td>
<td>SYSS</td>
<td></td>
<td>26799</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>BBIUSER</td>
<td>SYSS</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>NDMUPDNM</td>
<td>SYSS</td>
<td></td>
<td>338</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 (ISGLCRTC)
  • 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
DGRS,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)

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.

<table>
<thead>
<tr>
<th>S</th>
<th>Name</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALCWAIT</td>
<td>Allocation wait summary</td>
</tr>
<tr>
<td></td>
<td>AOMDATA</td>
<td>AOM analysis</td>
</tr>
<tr>
<td></td>
<td>APPCDATA</td>
<td>APPC/MVS Data Analysis</td>
</tr>
<tr>
<td></td>
<td>ASCHDATA</td>
<td>APPC/MVS Scheduler Data Analysis</td>
</tr>
<tr>
<td></td>
<td>ASMCHECK</td>
<td>Auxiliary storage paging activity</td>
</tr>
<tr>
<td></td>
<td>ASMDATA</td>
<td>ASM control block analysis</td>
</tr>
<tr>
<td></td>
<td>AVMDATA</td>
<td>AVM control block analysis</td>
</tr>
<tr>
<td></td>
<td>COMCHECK</td>
<td>Operator communications data</td>
</tr>
<tr>
<td></td>
<td>COUPLE</td>
<td>XCF Coupling analysis</td>
</tr>
<tr>
<td></td>
<td>CSFDATA</td>
<td>ICSF control block analysis</td>
</tr>
<tr>
<td></td>
<td>CTRACE</td>
<td>Component trace summary</td>
</tr>
<tr>
<td></td>
<td>DAEADATA</td>
<td>DAE header data</td>
</tr>
<tr>
<td></td>
<td>DB2DATA</td>
<td>DB2 analysis</td>
</tr>
<tr>
<td></td>
<td>DIVDATA</td>
<td>Data in virtual storage</td>
</tr>
<tr>
<td></td>
<td>DLFDATA</td>
<td>Data Lockaside Facility data</td>
</tr>
<tr>
<td></td>
<td>DLFTRACE</td>
<td>Data Lockaside Facility trace</td>
</tr>
<tr>
<td></td>
<td>GRSDATA</td>
<td>GRS managed resources</td>
</tr>
<tr>
<td></td>
<td>IMSDUMP</td>
<td>IMS analysis</td>
</tr>
<tr>
<td></td>
<td>IOCHECK</td>
<td>Active input/output requests</td>
</tr>
<tr>
<td></td>
<td>NTERMINATE</td>
<td></td>
</tr>
</tbody>
</table>
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 '___' TCB '___' _______

QNAME _ MYQNAME_

RNAME _

SCOPE: _ STEP _ SYSTEM _ SYSTEMS

  _ CONTENTION _ RESERVE


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: xrnlaactions
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?