HOW IT ALL STARTED

This was the diagnosis route that we took after discovering a performance problem within GRS1 and MIM2 during the migration from OS/390 2.10 to z/OS 1.2. As the old saying goes: “All roads lead to Rome.”

We had already installed and tested our z/OS 1.2 systems for several weeks in a test and acceptance Sysplex environment consisting of 3 LPARs. The LPARs were running on 2064 model 113s with varying levels of central storage ranging between 768MB and 6GB, in 64-bit mode. At this point, there were no obvious problems and the rollout began on our production Sysplex.

The first system was implemented successfully and, after several weeks, we proceeded gingerly to upgrade the next LPAR. This is when it all went a bit pear shaped. The expected throughput on the system was not being attained. An initial problem indicator was that the IMS transaction rate for this system dropped appreciably after the migration. There was a very I/O intensive hot batch workload on the system in question, which was processed in an unusually slow manner. Verifying that there were no I/O problems by checking the DASD subsystem, I/O queuing, cache, channel paths and that system paging was normal, we noticed that LPAR CPU usage was unusually high. Service Level Objectives were being exceeded and response times were unacceptably high.

RECOGNIZING THE PROBLEM

To get a view from 100,000 feet we used the RMF postprocessor PLOT(CPU) and REPORTS(CPU) reports from the SMF type 70 records. This was compared to RMF reports taken with a similar load under OS/390 2.10.

JCL AND OUTPUT FROM RMF POSTPROCESSOR

This indicated that CPU usage after the migration was much greater than normal.

IDENTIFYING THE MAJOR CAUSE OF THE PROBLEM

The RMF Monitor III delay reports were used to identify a potential candidate for further analysis, the CATALOG address space.

Using Omegamon’s drill-down menu on the Catalog Address space, we were able to narrow the problem down to the code within a couple of components:

IEAVESLK and IEAVESLL
-Suspend Lock Manager code

MVS uses a locking mechanism to serially use the set of sensitive reusable resources among
processors in a multiprocessor system. This method is also used to serialize reusable resources among programs executing within the same processor. Global Locks protect serially reusable resources related to more than one address space. Local Locks protect serially reusable resources assigned to a particular address space.

**ISGLNQDQ**

- **ENQ/DEQ code**

  ISGLNQDQ is the first GRS module to receive control for an ENQ or DEQ request.

We were sure that the Catalog Address was the guilty party. Using the Catalog REPORT,PERFORMANCE command we were able to collect more information about the problem.

- We reset the counters and waited about two minutes before issuing the command.

  To provide a valid comparison we produced this report on an OS/390 2.10 production system and on our 2 Sysprog test systems, running OS/390 2.10 and z/OS 1.2 respectively. Immediately apparent was the large count of ENQ/DEQ entries on the z/OS 1.2 system.

  To confirm our suspicions we carried out the following:

1. Increased size of System Trace Table using system command
   "TRACE ST 999K" (expect a proportionate increase in Real Storage used by the TRACE address space)
2. Dumped my own TSO address space with the MVS DUMP command, preventing locking of the Catalog address space during the dump phase
3. Examined System Trace Table for activity. We found an unusually high number of ENQ/DEQ SVC entries:

   Of all the SVC events recorded in 1.75 seconds of System Trace table, the Catalog address space had 43% SVC 38 (Enqueue/Reserve) and 54% SVC 30 (Dequeue).

**GRS ENQ/DEQ MONITOR**

IBM provides a free GRS monitor with z/OS, which has been available since OS/390. This tool was used to observe the ENQ/DEQ activity in the system. It requires a Job/STC to be active. JCL is provided in SYS1.SAMPLIB(ISGRUNAU). The monitor is activated by issuing the following command in a TSO environment:

```
EX ‘SYS1.SBLSCLI0(ISGACL5O)’
```

**OPENING THE PMRS**

We opened a problem record with IBM, via DIAL IBM (IBMLINK), and provided all the available information.

IBM has written an Information APAR II10752 to document what information is required to report a Catalog Performance problem when running a supported version of DFSMS supplied with OS/390 or z/OS. A problem record was also opened with Computer Associates via StarTCC http://esupport.cai.com/.

Additional information regarding problems between GRS and MIM can be obtained from APAR Q138222. This APAR provides a list of IBM APARs that address problems of MIM in collaboration with MVS.

Tobias wrote an assembler program that issued 999999 Enqueue/Dequeue SVCs to simulate the problem. This gave us a benchmark against which we could measure the results of fixes from IBM and CA. CPU and elapsed time were used to evaluate any improvements.

The results of these tests are fully documented in the paper by Scott Fagen from IBM.

**SOLUTION**

After applying and thoroughly testing a series of fixes provided by IBM and CA, this problem was successfully resolved.
SUMMARY

- Get an overview to understand the problem better.
- Identify the component/address space causing the delay or disruption of service.
- Supply detailed information to enable the software/hardware company to provide a solution quickly.
- Test the fix provided by your software/hardware company, remembering to compare this with the results received during the disruption.
- Document the outage for reference, in case of recurring problems and for employee education. Make this available to all your support personnel.

Tobias Dörkes, born in Germany, has been an MVS Systems Programmer since 1996. He is currently employed by Sparkassen-Informatik in Cologne, Germany. Reach him at tobias@world-of-os390.de

Charles MacNiven, MVS Systems Programmer since 1986, is always looking for new challenges. charles@world-of-os390.de is his e-mail address.

1. GRS

In a multi-tasking, multi-processing environment, resource serialization is the technique used to coordinate access to resources that are used by more than one program. When multiple users share data, a way to control access to that data is needed. Users who update data, for example, need exclusive access to that data; if several users tried to update the same data at the same time, the result would be a data integrity exposure (the possibility of incorrect or damaged data). In contrast, users who only read data can safely access the same data at the same time.

Global resource serialization offers the control needed to ensure the integrity of resources in a multi-system environment. Combining the systems that access shared resources into a global resource serialization complex enables you to serialize resources across multiple systems.

2. MIM Integrity

MIM is a software offering from Computer Associates that works in tandem with GRS. CA-Multi-image Integrity (MII) protects z/OS data integrity and resolves resource conflicts in shared DASD environments, adding additional integrity at a local system level. When an ENQ on a resource is issued, the following occurs:

1. SVC 56 (X’38’) is called. This SVC has been front-ended by MIM MII code, which intercepts every enqueue.
2. This code checks to see if the resource should be managed by MIM MII or passed to the original IBM SVC 56 code.

ISGLNQDQ is the first GRS module to receive control for an ENQ or DEQ request. When an ENQ on a resource is issued, the following occurs:

- ISGLNQDQ (for example, SYSZTIOT) are typically processed very quickly (about 20-25 microseconds per ENQ/DEQ pair at one installation). With the new GRS exits active, however, ISGLNQDQ passes all requests to another GRS module, ISGGNQDQ, which has a much longer code path since it processes requests that are more complex. Analysis of performance data obtained by one customer suggests this change to GRS logic increases the elapsed time of an ENQ/DEQ pair, which was previously satisfied by ISGLNQDQ alone, by approximately 70 microseconds, or 0.000070 seconds. Consequently, clients with IBM APAR OW 51103 applied (z/OS 1.2 and higher) who run Unicenter CA-MII 4.5 will be using these exits, and therefore may observe a performance degradation for local ENQ and DEQ requests.