

July 2003



GRS/MIM Installation Exit Performance,

Or, how I learned to stop worrying and love cross-memory mode.

by Scott Fagen
z/OS Core Technology Design

Contents

2 Abstract
2 Introduction
**4 How PC-ENQ was enabled for
CA-MIM support**
7 The Problems
10 Results
11 Acknowledgements

Abstract

This article is intended to describe the changes made to the z/OS® operating system intended to improve the interoperability between the Unicenter CA-MII Data Sharing (formerly named Multi-Image Integrity) component of the Unicenter CA-Multi-image Manager for Resource Sharing for z/OS and OS/390® product (CA-MIM) and the GRS (Global Resource Serialization) component. These changes were precipitated by the introduction of PC (program call) linkage for the ENQ, RESERVE, and DEQ services provided by GRS. Without these changes, customers using CA-MIM would not be able to migrate to z/OS 1.2 (or subsequent releases), as the product would not be able to correctly process service requests that use the new linkage.

The original changes had significant deleterious performance effects on a number of customers, and, in particular, customers that used the CA-MIM product. This article will also describe the subsequent improvements made to both GRS and CA-MIM to resolve the performance issue.

Introduction

With the introduction of GRS Wildcard RNLs in z/OS R2 (retrofitted to OS/390 and z/OS R1 via OW49779), GRS replaced the RNL Search exit interface, known as ISGGREX0, with a new, dynamic exit point called ISGNQXIT. The intent in changing this exit was two-fold:

- 1. Now that the RNL searching algorithm was much more flexible and robust, installations no longer needed to provide their own modifications for searching the list. Many customers could remove their installation specific code and rely strictly on the search capability provided by the operating system.*

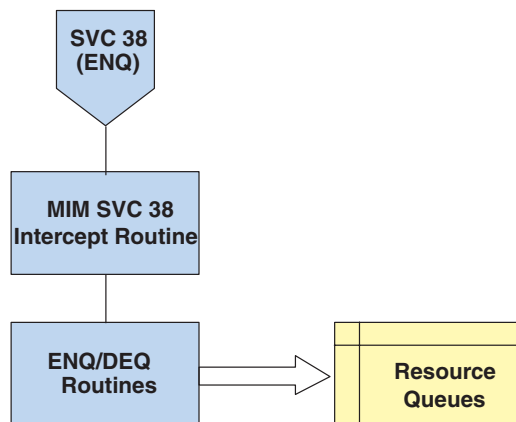
2. Many installations used the exit to alter (albeit illegally) attributes of the ENQ request. The ISGGREX0 interface required the installation code be able to run in cross-memory mode while locked, which can be a difficult environment to code to, especially for recovery. A failure in this code could cause system integrity exposures or system failures. The new exit point provides installations the ability to modify most of the attributes of the request (in addition to just the scope) in a more code-friendly environment, with the robust recovery provided by the dynamic exits interface.

Following GRS Wildcard RNLs, the development team turned to answering a long standing requirement against GRS, providing the ability to call the ENQ suite of services in cross-memory mode, which is not supported by SVC (supervisor call) entry (ENQ and RESERVE share SVC 38, DEQ uses SVC 30). By adding a PC (program call, commonly called LINKAGE=SYSTEM on macro invocations) mechanism to invoke ENQ, programs could obtain serialization in environments, where it was not previously possible. This support, introduced via OW51103 on z/OS V1.2, was first exploited by ATS Star, but is now gaining acceptance and popularity within the operating system, subsystems, and other applications.

Before the advent of PC-entered ENQ, CA-MIM only needed to be concerned with requests coming in through the SVC interface. CA-MIM operated by replacing system defined SVC entry point for ENQ, replacing it with an entry into their product. It would interrogate each request, looking for resources that it would serialize, process them and then branch to the system supplied ENQ routines. Mainline ENQ process was unaware that any of this activity was occurring. Now, requests submitted through the PC interface will not be seen by the CA-MIM SVC intercept routine, creating the potential for the same resource name to not be correctly serialized with a request that was submitted through the SVC intercept.

How PC-ENQ was enabled for CA-MIM support

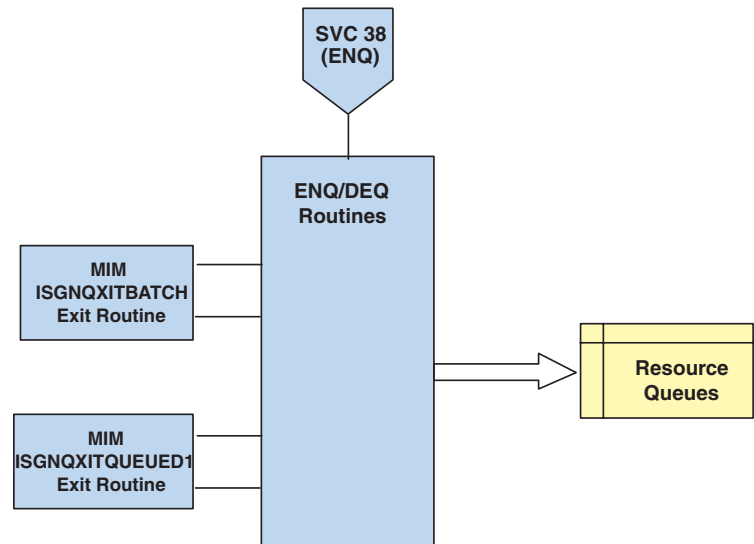
For LINKAGE=SVC requests, CA-MIM receives control by ‘front-ending’ the ENQ SVC. When CA-MIM initializes, it removes the z/OS supplied entry point from the system SVC table and replaces it with the address of a CA-MIM routine. In this routine, CA-MIM is able to determine which resource requests it is interested in, perform its version of global queuing, and pass the requests on to the z/OS ENQ routine.



To enable CA-MIM to manage ENQ and DEQ requests received via the LINKAGE=SYSTEM entry, the IBM and CA development teams architected a new set of exits points which would allow CA-MIM to have the same influence on ENQ processing as their SVC intercept. These exit points are provided on both the LINKAGE=SVC path and the LINKAGE=SYSTEM path, so that a product would be able to see and modify all ENQ and DEQ requests, regardless of the linkage. It was also our long-term goal to have CA-MIM to perform its processing via these exits for SVC entered requests, but as of this writing, this has not yet been realized.

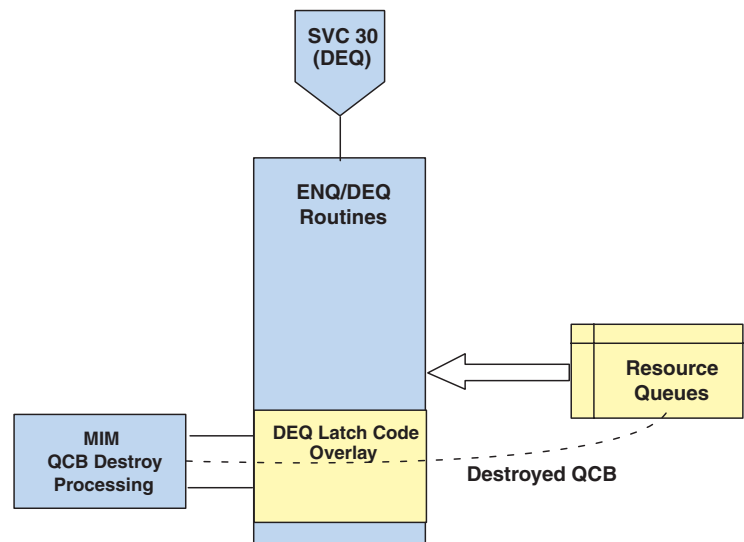
The new exit points are:

- 1. ISGNQXITBATCH:** *This exit point is invoked after all parameter list validation has been completed by the ENQ/DEQ routines. The exit routine can alter some of the processing associated with a request (i.e. convert the request from a RESERVE to an ENQ) and set a return code (perhaps, for a conditional request). The exit receives control enabled, unlocked and in the requester's home address space.*
- 2. ISGNQXITQUEUED1:** *This exit point is invoked after all of the local requests have been processed and the global requests have been queued to the GRS global processor. The results of all the local requests (return/reason codes, ABEND code) are presented to the exit routine.*

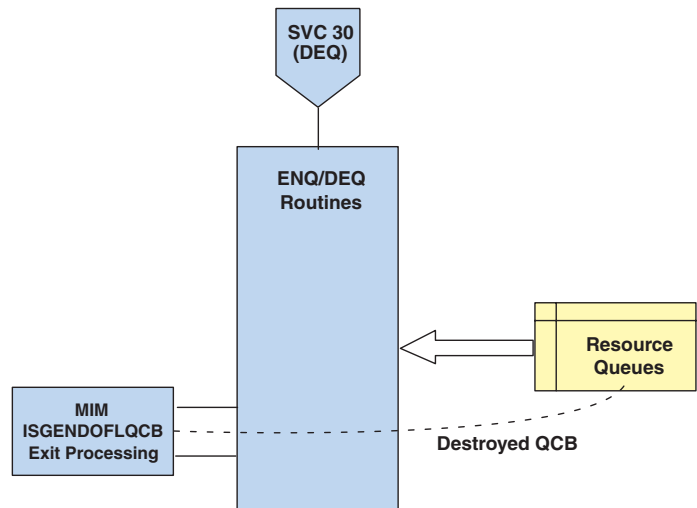


There are a number of advantages to this structure. First, it provides the operating system the freedom to provide new format interfaces into ENQ and DEQ without breaking the CA-MIM product. Second, it removes the need for CA-MIM to maintain its own version of parameter list checking, which must be kept in synch with that provided by the operating system. Third, it paves the way for further streamlining of the processing performed by future versions of CA-MIM, and, finally, it provides well-defined interfaces for serialization request processing with good reliability, availability and serviceability characteristics.

To manage DEQ requests, CA-MIM needed to understand when a specific event (GRS destroying a QCB) occurred during DEQ processing. This was handled by looking for a specific series of instructions within the DEQ process (called 'the DEQ latch'), and replacing it with a branch to CA-MIM module that processed the DEQ.



Again, to ensure that changes to GRS would not affect CA-MIM, the DEQ latch code was replaced with an exit point, called ISGENDOFLQCB.



So, on paper, this structure looks great. In addition to the functional changes, IBM and CA shipped code that ensured compatibility between levels of z/OS and CA-MIM so that installations would be able to smoothly transition to using OW51103 with CA-MIM.

The Problems

Two major performance problems were brought to light soon after this code hit the field. The first problem was really introduced with the exit support provided with the GRS Wildcard support. This problem was due to the inefficiency in the code setting up to call the exits. In the SVC code, the system is running in a rather onerous environment: holding system locks and executing in cross-memory

mode. To clear this hurdle, the exit setup routine would save status (serialized by the system lock) into newly obtained storage, clear the locks and recovery, establish a new recovery environment and invoke the exit. An additional side effect of obtaining the storage and copying the data was a large number of page faults incurred during the copy.

This problem was partly addressed by re-implementing the storage management, pooling the storage used to save the status, rather than obtaining them anew for each ENQ or DEQ request. For ISGNQXIT users, additional benefit was derived from providing another exit point, called ISGNQXITFAST, which was invoked in the onerous environment, eliminating the overhead of the storage management and recovery manipulation.

The second problem turned out to be an excessive number of calls to the ISGNQXITBATCH exit. Requests for global ENQs represent a small portion of the overall ENQ traffic generated by the system. Therefore, CA-MIM would only need to be invoked for this minority of requests. Additionally, CA-MIM operates without having to directly process any of the DEQ requests (remember, CA-MIM manages DEQs from the ISGENDOFLQCB exit). Since the ISGNQXITBATCH exit has a fairly complex interface, the cost to construct and destruct the parameter list, plus the overhead of the locking and recovery changes caused significant system overhead when driven for every ENQ and DEQ in the system, particularly when injected into the relatively short code path for non-global ENQ and all DEQ requests.

To further streamline exit processing, GRS implemented a “lighter weight” (fast) set of exits that required even less overhead to call. These exits are invoked in the locked, cross-memory environment, restricting the available function, but still providing some value. The ISGNQXITFAST exit point was provided to

improve performance over the ISGNQXIT exit point. To further streamline the ISGNQXITBATCH process, another new fast exit point, called ISGNQXITPREBATCH, was created to determine whether or not another new (slow) exit point ISGNQXITBATCHCND (Batch Conditional Exit) gets called. In the ISGNQXITPREBATCH exit, CA-MIM indicates when the ISGNQXITBATCHCND exit gets called:

- 1. When the request is an ENQ*
- 2. When it is interested in processing the resource.*

The ISGNQXITBATCHCND exit point has the same interface definition as the ISGNQXITBATCH exit.

Using the ISGNQXITPREBATCH exit to ‘short-circuit’ calls to the ISGNQXITBATCHCND exit turned out to be an excellent mechanism to drastically reduce the performance impact. Calls to the more complicated ISGNQXITBATCHCND exit interface now only occur when CA-MIM needs to perform global processing. Note that there is still a degradation compared to just using the SVC intercept. However, this overhead is only a small part of the time it takes to process a global request. This cost is far outweighed by having a firm set of interfaces with strong reliability, availability, and serviceability characteristics.

The GRS development team made one final tweak to speed up the ‘fast’ exits a little bit more. Rather than obtaining and releasing storage from the pools for the parameter lists associated with the ‘fast’ exits, the code now uses a single parameter list area, which is serialized by the system locks held by ENQ and DEQ processing.

Results

The results shown here are from a real customer’s system programmer environment, so the usual ‘your mileage may vary’ caveats apply. The test issued 999999 non-global ENQ and DEQ requests for the same resource, serially.

Test Environment	Test Execution Time (Wall clock)
OS/390 2.10 CA-MIM 4.5.0	19 sec
z/OS 1.2 + OW51103 CA-MIM 4.5.0	354 sec
z/OS 1.2+OW51103 + OW56028 CA-MIM 4.5.0 + TD45048	45 sec
z/OS 1.4* + OW56028 CA-MIM 4.5.1 + TD45048	33 sec
z/OS 1.4* + OA01861 CA-MIM 4.5.1 + TD45048	20 sec

*-indicates that OW51103 functionality is part of the base z/OS 1.3 support

Notes:

1. OW51103 introduced ENQ/DEQ LINKAGE=SYSTEM system support and the ISGNQXITBATCH, ISGNQXITQUEUED1, and ISGENDOFLQCB exits. CA-MIM 4.5.0 provided exploitation of these exits.
2. OW56028 introduced the new exits to GRS (ISGNQXITFAST, ISGNQXITPRE-BATCH, ISGNQXITBATCHCND). TD45048 provides the CA-MIM exploitation of the exits. TD45048 is included in CA-MIM 4.5.2.
3. OA01861 changed the ‘fast’ exits to use the single, serialized area, rather than obtaining storage from the pool.

Acknowledgements

I'd like to acknowledge the following people (and the ones who supported them) in our getting these fixes out and tested:

Heather Dennis (IBM) GRS Development

Bryan Childs (IBM) GRS Development

Susan Lamberton (IBM) GRS Function Test

Tobias Doerkes (SIS West)

Dieter Baltschus (SIS West)

Kathy Milligan (SSAN)

The IBM Catalog Level 2 and Level 3 teams

And, of course, the CA-MIM product development team.



© Copyright IBM Corporation 2003

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
07-03
All Rights Reserved

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

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

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

Intel is a trademark of Intel Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

Information concerning non-IBM products was obtained from the suppliers of their products or their published announcements. Questions on the capabilities of the non-IBM products should be addressed with the suppliers.

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

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.

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.



GM13-0328-00