z/OS Performance: New algorithm for synchronous to asynchronous conversion of CF requests with z/OS 1.2

z/OS 1.2 introduces a new algorithm for determining whether or not it is more efficient to issue a command to the coupling facility synchronously or asynchronously. Based on the configuration and workload, some parallel sysplexes may experience a significant change in the reporting of activity to some structures in the Coupling Facility RMF report. This flash discusses the value of the new algorithm and the type of changes in the reports one may experience.

XES (the component of OS/390 and z/OS that communicates with the coupling facility) has the ability to issue requests to the coupling facility in a synchronous (processor waits or "dwells" until the operation is complete) or asynchronous (processor can run another task rather than waiting) manner. Asynchronous operations will take longer than synchronous operations and require more software processing to complete (as well as impacting the hardware efficiency due to context switching between tasks). However, from a total capacity impact standpoint, a long running synchronous operation can cost more than an asynchronous operation due to the cost of the processor dwelling time. Based on the speed of the processor, as the synchronous service time for a request increases, there will come a point where issuing the operation asynchronously will cost less than the dwelling time of a synchronous operation.

In releases prior to z/OS 1.2, XES tried to compensate for these differences by converting some list or cache synchronous requests to asynchronous requests based on preset rules of thumb.

z/OS 1.2 contains a new algorithm to recognize this crossover point for all configurations, link technologies and workloads and makes the appropriate decision. Also, with z/OS 1.2, essentially all operations to the coupling facility are eligible to be issued asynchronously. Of particular importance is the fact, prior to z/OS 1.2, lock structure operations had to be issued synchronously.

Performance analysis of coupling facility data has focused on service time as one measure of goodness - the lower the service time the better. This is mostly an outgrowth of a "synchronous" view of coupling. As illustrated in the examples below, a migration to z/OS 1.2 may cause relatively long synchronous operations to be converted to even longer asynchronous operations - on the surface, moving things in the "wrong" direction.

However, what is really happening is a tradeoff favoring improved host CPU capacity over a generally unnoticeable elongation of response time (as viewed by the end user). Consider the following example. Due to a long distance to the coupling facility or due to slower coupling technology, a simple lock operation issued by a z900 processor was observed to take 100 microseconds of synchronous service time. After migrating to z/OS 1.2, this operation was converted to asynchronous with a service time of 250 microseconds. If a typical transaction does 10 lock operations, this means the transaction response time would elongate by 10 x .000150=.0015 seconds - not likely to be noticed by the end user! However, if the z900 processor running such a transaction processing workload was executing a total of 10,000 lock operations per second, this conversion may have just saved 100 MIPS of host CPU capacity! Thus, the new synch/asynch conversion algorithm in z/OS 1.2 is recognizing when it can provide a host CPU capacity benefit with no noticeable impact on end user response time.

 © IBM Corporation, 2002

Let's look in more detail at what changed with the algorithm and some examples of RMF reports which illustrate the differences. Note many systems will see only a minor increase in asynchronous processing, while other systems, particularly those with a long distance to a coupling facility (several kilometers or more), will see a significant increase in asynchronous activity.

The conversion of "expensive" synchronous requests is not a new phenomena. In releases prior to z/OS 1.2, XES converted some list or cache synchronous (SYNC) requests to asynchronous (ASYNC) based on preset rules of thumb. Three conditions triggered a conversion:

1. Request type - Certain types of CF requests (for example, a request with a list of operations to perform to a structure) were known to be generally long running requests, so all of these requests were converted.
2. Sender and receiver processor speed - If the sending processor was significantly faster than the CF processor, many more host MIPS would be spent waiting for a slower CF to process the request. Tables of IBM sending processors and CFs were used to determine if requests were to be converted.
3. Amount of data being sent - All requests which contained more than 4K bytes of data were converted.

While these hard-coded rules largely accomplished their intended purpose, they did not handle some of the conditions which were specific to certain configurations or workloads. An example of such a configuration would be a geographically dispersed parallel sysplex (GDPS), where one CF is located a considerable distance from the sending processor. A certain type of request might complete in a reasonable time on a local CF but would take much longer if sent to the distant CF. In this case, it would be more efficient to convert the request to the distant CF but not convert the request to the local CF. Another example of such a configuration is one in which fluctuations in the CF workload cause fluctuations in the synchronous CF service time, regardless of the "inherent" processor speed of the CF. Yet another example might be one where the CF is executing in a low-weighted CF partition with shared CPUs, or with Dynamic CF Dispatching active, both of which tend to elongate synchronous CF service times regardless of the inherent processor speed of the CF.

To address these situations, a new function, known as the heuristic synch/asynch conversion algorithm was introduced in z/OS 1.2. This function monitors CF service times for all (LIST, LOCK, and CACHE) synchronous request types to a specific CF, also taking into account the amount of data transfer requested on the operation, and other request-specific operands that significantly influence the service time for the request. It compares these observed service times to thresholds to determine which operations would be more efficiently executed asynchronously. Different thresholds are used for simplex and duplex requests and for lock and non-lock requests. All thresholds are normalized based on the processor speed of the sending processor. The algorithm and thresholds are not externally adjustable. z/OS APAR OW51813 contains the latest threshold and algorithm adjustments.

Let's look at a few examples.

Example 1

Example 1 shows two MVS images, JA0 and JE0, on a z900 processor. The CF structure resides on a z900 CF. A few requests on JE0 were converted to ASYNC. These requests were probably long running DB2 batch unlock commands.

```
 STRUCTURE NAME = DSNDB1G_LOCK1     TYPE = LOCK   STATUS = ACTIVE

          # REQ    ------------ REQUESTS -------------    ------- DELAYED REQUESTS ------
SYSTEM   TOTAL          #     % OF  -SERV TIME(MIC)-    REASON    #  % OF   AVG TIME(MIC)
NAME     AVG/SEC        REQ    ALL     AVG   STD_DEV            REQ   REQ   /DEL   STD_DEV


JA0       641K   SYNC   641K   20.8    33.0    132.7     NO SCH    4   0.0   21.8     4.9
       355.9     ASYNC    0     0.0     0.0      0.0     PR WT     0   0.0    0.0     0.0
                 CHNGD    0     0.0  INCLUDED IN ASYNC   PR CMP    0   0.0    0.0     0.0

JE0      1073K   SYNC  1072K   34.8    34.5    134.0     NO SCH  114   0.0  104.2   241.9
       596.1     ASYNC  502     0.0   128.2    224.9     PR WT     0   0.0    0.0     0.0
                 CHNGD    0     0.0  INCLUDED IN ASYNC   PR CMP    0   0.0    0.0     0.0
```

Notice the requests which were converted by XES are reported as ASYNC. The CHNGD field reports only the non-immediate requests which were changed because of a subchannel busy condition. The CHNGD count thus continues to be useful as an indicator of a shortage of subchannel resources which may need to be corrected, as today.

There is another change you may notice on the RMF 1.2 reports. Prior to z/OS 1.2, subchannel delays (NO SCH) on the CF Structure Activity report were only reported for ASYNC requests. With z/OS 1.2, subchannel delays for both ASYNC and SYNC requests are combined and reported on a structure basis.

Example 2

Example 2 shows a list structure residing on a G5 CF. JA0 is an MVS image on a z900 processor and J80 is an MVS image on a G6 processor. On JA0, the faster processor, most of the requests have been converted to ASYNC. (Note: XES periodically overrides its decision and sends a request as SYNC to maintain a current measure of synch service time). J80, the slower processor, is more closely matched with the speed of the G5 CF, so most of these requests remain synchronous.

```
 STRUCTURE NAME = RRSLOG_DELAYED    TYPE = LIST   STATUS = ACTIVE
          # REQ    -------------- REQUESTS -------------   --------------
 SYSTEM   TOTAL             #     % OF  -SERV TIME(MIC)-    REASON    #
 NAME     AVG/SEC           REQ    ALL     AVG   STD_DEV            REQ

 JA0        107    SYNC      2     0.2    57.0     26.9     NO SCH    0
           0.12    ASYNC   105    10.0   346.3    960.3     PR WT     0
                   CHNGD     0     0.0  INCLUDED IN ASYNC   PR CMP    0
                                                            DUMP      0

 J80         95    SYNC     74     7.1    80.2      6.6     NO SCH    0
           0.11    ASYNC    21     2.0   466.5     59.4     PR WT     0
                   CHNGD     0     0.0  INCLUDED IN ASYNC   PR CMP    0
```

Example 3

Example 3 is drawn from a configuration with distance involved. Image L2 is on the same z900 processor as the CF containing this SCA structure, while image L1 is on a z900 5km distant from the CF. Notice all the requests from L2 are synchronous, but the vast majority of the requests from L1 are asynchronous (due to the elongation of service time due to distance). The few synchronous requests from L1 are those forced to remain synchronous to provide a current synchronous service time for comparison.

```
STRUCTURE NAME = DSNDB71_SCA        TYPE = LIST   STATUS = ACTIVE
           # REQ     -------------- REQUESTS -------------    --------------
SYSTEM     TOTAL              #      % OF  -SERV TIME(MIC)-    REASON     #
 NAME       AVG/SEC            REQ     ALL      AVG   STD_DEV             REQ

 L1          2688     SYNC      33      0.6     72.6      30.4     NO SCH     2
             2.24     ASYNC   2653     49.8    165.1      96.5     PR WT      0
                      CHNGD      2      0.0  INCLUDED IN ASYNC    PR CMP     0
                                                                  DUMP       0

 L2          2638     SYNC    2638     49.5     25.6      29.3     NO SCH     0
            2.20      ASYNC      0      0.0      0.0       0.0     PR WT      0
                      CHNGD      0      0.0  INCLUDED IN ASYNC    PR CMP     0
                                                                  DUMP       0
```

Summary

In summary, z/OS 1.2 introduces a new algorithm for synchronous to asynchronous conversion of CF requests. It is capable of recognizing any situation where host processor capacity is being impacted by poor synchronous service times (relative to the speed of the sending processor), and taking the appropriate action of converting those requests to asynchronous to limit the impact. The resulting higher service time for the CF request (due to it now being asynchronous) should have no noticeable effect on end user response time for a transaction processing workload.

Special Notices

This publication is intended to help the customer manage the performance of a parallel sysplex in a z/OS 1.2 or later environment. The information in this publication is not intended as the specification of any programming interfaces provided by z/OS. See the publication section of the IBM programming announcement for the appropriate z/OS release for more information about what publications are considered to be product documentation. Where possible it is recommended to follow-up with product related publications to understand the specific impact of the information documented in this publication.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Performance data contained in this document was determined in a controlled environment; therefore the results which may be obtained in other operating environments may vary significantly. No commitment as to your ability to obtain comparable results is any way intended or made by this release of information.