z/JOURNAL

CICS Top Performance & Tuning Issues

By Trevor Eddolls

This article introduces some basic performance concepts so readers interested in tuning, specifically new analysts, have a starting point. We'll examine the three main areas to look for performance issues—I/O, virtual storage and real storage—and address some additional considerations.

Before looking at performance issues in detail, let's review some background that provides a foundation for understanding CICS.

CICS Overview

More than 90 percent of *Fortune* 500 companies use CICS, which, since its debut in 1969, has grown substantially in >

the amount of work it can process reliably. CICS lets applications read and update data quickly and securely. Each activity is called a transaction, so CICS is referred to as a transaction manager or server.

CICS supports applications written in Assembler, COBOL, PL/I C, C++, REXX, Java, and Enterprise Java Beans (EJB); Version 4.2 provides access to the latest, multi-threaded, 64-bit Java Virtual Machine (JVM) server.

Traditionally, CICS users worked on "forms" on local terminals, but, today, a CICS user could be running a transaction from a remote browser. Application programs access CICS facilities via the Application Programming Interface (API). It's also possible to run background transactions. The advantage of using CICS is that it takes care of recovering data in the event problems occur. More recent advances in CICS have allowed it to provide and consume Web services and provide a Service-Oriented Architecture (SOA). The current release for z/OS is CICS TS Version 4.2, which was announced in April and became generally available in late June.

Typically, CICS activity occurs inside a CICS region, and there's typically more than one region. The CICS regions aren't limited to a single z/OS system. Inside each region is the all-important Quasi-Re-entrant Task Control Block (QR TCB). Accessing other services may require using other Task Control Blocks (TCBs). In the world of CICS acronyms, there are Terminal-Owning Regions (TORs), Application-Owning Regions (AORs), and perhaps File-Owning Regions (FORs). The final complication is that some applications can be marked as threadsafe; these behave slightly differently from non-threadsafe applications in that they don't return to the QR TCB when they finish using a service.

Monitoring

How do you know if there's a problem with CICS? Several different tools within CICS provide insight into CICS performance:

- CICS Monitoring Facility (CMF) collects data about the performance of all user- and CICS-supplied transactions; the data can be analyzed offline at a later time.
- CICS trace facility will record the progress of a CICS transaction through the CICS management modules, providing a history of events.
- CICS statistics collect information on the CICS system as a whole. Five types of statistics are written to System Management Facility (SMF) data sets: interval, end-of-day, requested, requested reset, and unsolicited.
- The IBM CICS Performance Analyzer extends CICS Explorer support to help users develop, tune, and manage their CICS systems. (CICS Explorer is an Eclipse-based graphical tooling interface for CICS.)

Third-party monitor products are also available that display information about CICS system activity, active tasks, dynamic storage areas, transaction logs, and other CICS performance information.

You can use several other z/OS tools, including the familiar Service Level Reporter (SLR), Generalized Trace Facility (GTF), and Resource Measurement Facility (RMF).

I/O Issues

Accessing files and databases to create/update/delete records is so common that it can be taken for granted, but it is perhaps the most important area for tuning—one that can have a huge impact on CICS performance. So, for a transaction that involves lots of I/O, the bulk of the task response time is spent doing I/O; that's because four primary operations might be required when a record is retrieved: IOSQ time, PEND time, DISC time, and CONN time.

IOSQ time is the time an I/O waits for a Unit Control Block (UCB) to become available. The wait is usually due to another I/O currently being under way. PEND time is the time spent waiting for a physical path (channel, control unit, DASD) to become available. DISC time is the time it takes the DASD hardware to access the correct part of the disk to access the data. CONN time is the useful connection time when I/O activity actually occurs. You can reduce IOSQ time by using Parallel Access Volumes (PAV) that allow for multiple accesses to the same volume under certain conditions.

The perfect way to reduce file I/O is to move the data into real storage ahead of time. However, this isn't always a realistic option because real storage space is limited. An alternative is to move data into the cache of storage controllers.

Using CICS data tables means files are preloaded into CICS storage. If real storage is sufficient, Virtual Storage Access Method (VSAM) data can be stored in real storage and I/O speeds are almost immediate. Ideal candidates for this are medium-size or smaller files that are mainly browsed (rather than regularly updated). The settings used when defining a file in Resource Definition Online (RDO) are TABLE and MAXNUMRECS.

Next, specify high values for buffers in the Local Shared Resources (LSR) pool. This way, as many buffers as possible are allocated in real and virtual storage. This is particularly effective for highly active files because the next record required is likely already in the buffer.

With Key Sequenced Data Sets (KSDS) VSAM files, it's better to increase the number of index buffers than data buffers because index buffers are read repeatedly for different data record accesses. This improves I/O performance for several different transactions or multiple accesses of data for a single transaction.

VSAM strings are logical entities that represent requests for VSAM services involving data set positioning. Depending on a file's access activity, you'll probably have more strings assigned. File strings are specified in individual file definitions. Pool strings can be defined for up to the eight pools possible with LSR. By monitoring the wait on strings, it's possible to identify where changes can have the maximum impact on performance.

DFHRPL libraries contain all the user-written programs and CICS programs. Contents of these libraries are loaded asynchronously into CICS dynamic storage. CICS performance can be improved by reducing DFHRPL I/O. This performance gain can be achieved by modifying Library Look-Aside (LLA) ratios and the Virtual Look-Aside Facility (VLF).

Virtual Storage Issues

Virtual storage issues with CICS can arise simply because CICS is being used for long periods. The consequence is a Short on Storage (SOS) condition. The appearance of storage cushion release messages is a clue that storage is running out. The problem may occur because main temporary storage, in-storage tables, and reference matrices may use virtual storage, but not release it when they've finished.

When CICS is started, the REGION parameter determines the amount of virtual storage initially available. A good region size is 0 M, which reserves the entire 2GB address space for CICS. The DSALIM and EDSALIM parameters in the System Initialization Table (SIT) determine how much of the space reserved by the REGION Job Control Language (JCL) parameter will be used for the Dynamic Storage Area (DSA) and Extended Dynamic Storage Area (EDSA). The values specified should be in excess of the requirements (by 200 percent or more) to allow for any "dead" space. Because CICS allocates storage as it's needed, the extra allocation doesn't affect performance.

Other parameters, such as CDSASZE, ECDSASZE, RDSASZE, ERDSASZE, SDSASZE, ESDSASZE, UDSASZE, and EUDSASZE can be submitted as overrides. These were used to resolve a problem that occurred when users could get SOS conditions below the line when using transaction isolation because the UDSA had to be allocated in 1MB increments on a 1MB boundary. Their use is no longer recommended because they eliminate the CICS storage manager's dynamic capacity to handle virtual storage.

The CICS maximum number of tasks (MXT) parameter can help influence virtual storage use by controlling the maximum number of tasks you can execute at any one time in CICS. You can also use MAXACTIVE and TRANCLASS to limit a particular class of transaction. By controlling the number of transactions, it's possible to allow high-priority tasks to run while limiting execution of lower-priority or high virtual storage consumers.

CICS systems are often split into TORs, FORs and AORs, as it provides a way of overcoming virtual storage constraints. There are two ways of sharing resources: Multi-Region Operation (MRO) and InterSystem Communication (ISC). MRO is used when the splits are within a single z/OS image. ISC or cross-system coupling facility/multiple-region option (XCF/MRO) is used with system splits across multiple z/OS images. XCF/MRO must be used in the same parallel sysplex.

Using CICS data tables lets you move buffers to a data space (a z/OS data-only address space). This reduces the demand on virtual storage for buffers in the CICS address space.

Real Storage Issues

Real storage constraints can hamper CICS significantly. A paging operation makes the whole CICS region pause until it completes. This has a domino effect, resulting in CICS performance across the board going from bad to worse. The good news for most sites is that modern processors have a large amount of real storage available and the cost is significantly lower than in the past. If real storage constraints are occurring, you should:

- Make CICS non-swappable. This prevents the address space from being swapped out by z/OS and reduces the paging overhead. This can be achieved by using the PPTNSWP option in the MVS Program Properties Table (PPT). **Note:** CICS performs swapping for MRO and cross-memory environments.
- Look at the file and database buffer allocations. Problems can occur if there isn't enough real storage to support the number of buffers allocated. Gradually reducing the number of buffers used might be the first strategy. You can use cache controllers to improve I/O performance. With less real storage being used, more is available for CICS to use for other tasks.
- Implement VSAM LSRs. This means that buffers can be shared between files instead of having them dedicated to individual files. Again, this reduces real storage usage.
- Place all eligible modules in the Link Pack Area (LPA) and Extended Link Pack Area (ELPA). This means only one copy of the modules will be used for all CICS regions rather than each region having its own modules. This can have a huge impact on real storage usage.
- Split CICS across different physical processors. This makes extra real storage space available for CICS to use. CICS ISC and XCF/MRO can be used to allow communication between different processors or z/OS systems.

Other Issues

While I/O, virtual storage, and real storage are the main places to look for performance issues, sites can also benefit from ensuring VSAM files are tuned optimally. Here, buffers and LSR, sub-tasking, data tables, and string settings are all important.

Terminal control is an area that benefits from tuning, including Terminal Input/Output Area (TIOA), the VTAM high-performance option, and compression of output terminal data streams.

Data Language Interface (DL/I) can be tuned to improve performance in areas such as DBCT maximum and minimum threads. RDO is used to define the CICS DB2 connection, which uses three different objects: the DB2 connection definition (DB2CONN), the DB2 entry definition (DB2ENTRY), and the DB2 transaction definition (DB2TRAN). The values assigned can tune DB2 connection thread performance. Activity keypoint frequency and journaling options can help with journaling performance. MRO and ISC both benefit from tuning.

With closer examination, a wide variety of CICS services can improve performance. For example, look at how you specify temporary storage (main temporary storage, auxiliary temporary storage, and secondary extents for temporary storage) and transient data.

Outside the Box

The interaction between different components running on the mainframe makes solving performance issues more complex. Users may be aware that CICS is running slowly and your site has a performance issue, but the cause of the issue might be IMS, DB2, WebSphere MQ, or even z/OS itself.

From a business perspective, it may seem as if a single transaction goes out, accesses some data, and displays it. From a technical perspective, a single CICS transaction could perhaps involve an IMS transaction, a DB2 intervention, and WebSphere MQ before the results are displayed on the user's screen. You can use products such as the IBM CICS Performance Analyzer to identify the cause of a problem in CICS. However, if the symptom appears to be in CICS, but is really in WebSphere MQ, then CICS Performance Analyzer won't identify the problem. In this case, IBM's Transactional Analysis Workbench can help. It automates the collection of data needed and provides a session manager to manage problem analysis through its lifecycle. It also provides a way for less-experienced staff to identify the source of the problem. The next step is for the appropriate system (IMS, WebSphere MQ, DB2, z/OS) experts to take a detailed look to determine the cause of the problem.

Transactional Analysis Workbench links closely with other tools, providing a window into other subsystems that impact CICS performance. To find the origin of the problem, consider what's changed and use information from SMF, OPERLOG, and other data sources such as:

- CICS-DBCTL transaction performance
- IMS address space resource consumption
- WebSphere address space performance
- WebSphere MQ and DB2 external subsystem (ESAF) performance
- Advanced Program-to-Program Communication (APPC) transaction performance
- IMS Resource LockManager (IRLM) long-lock activity.

Analyzing these sources can help you identify and resolve CICS performance issues that are caused outside the CICS "box."

Conclusion

CICS has been around for more than 40 years and has evolved to encompass a massive number of parameters that a systems programmer can set—almost any of which can impact performance and response times. Here we've identified the primary areas to examine in order to resolve performance problems that occur within CICS as well as originate elsewhere.

Trevor Eddolls is an IBM Information Champion and CEO of iTech-Ed Ltd., a company specializing in mainframe consultancy and writing. He has more than 30 years of experience in all aspects of IT. For many years, he edited Xephon's *CICS Update* journal and now maintains the Virtual CICS user group (www.fundi.com/virtualcics) Website. You can read his weekly blog at mainframeupdate.blogspot.com, follow his tweets at twitter.com/t_eddolls, or find him on Facebook at www.facebook.com/pages/iTech-Ed-Ltd/201736748040. Email: trevor@itech-ed.com; Website: www.itech-ed.com