Going Sysplex With CICS: The Affinity Problem

BY KEVIN COGLEY

The CMOS hardware evolution has caused significant problems for applications such as CICS which were built to perform optimally in the “big iron” environment.

In order to improve the cost/performance of large-scale computing environments, IBM has moved away from the large, monolithic bipolar water-cooled technology (80 to 120 MIPS) to the much less expensive, air-cooled CMOS (10 to 20 MIPS) technology. In order to deliver this new technology, it was necessary to repackage the processing power in smaller increments. So while the aggregate MIPS available have increased greatly, the number of MIPS available to a single TCB (task control block) or SRB (service request block) is less. The end result is that the processing power is spread over more MVS regions and more physical CPUs. This means that applications such as CICS, which are essentially limited to the speed of a single processor, will suffer throughput degradation in a CMOS hardware environment unless their architecture is restructured.

The hardware evolution causes significant problems for applications and systems, such as CICS, which were built to perform optimally in the “big iron” environment.

CICS operates primarily as a single TCB, so most work is performed at the speed of a single processor. With the ever-growing workload, CICS was enhanced with Multi-Region Operation (MRO) to have terminal owning regions (TORs), application owning regions (AORs), and file owning regions (FORs). This helped CICS do some parallel processing but applications were primarily grouped by function.

Like IMS, CICS could run well in a Parallel Sysplex, since it comes with a dynamic transaction routing (DTR) program which is capable of distributing the CICS workload over multiple regions in multiple MVSes. If each CICS was a clone, this would work perfectly. However, the CICS architecture includes something which prevents this ideal situation from occurring: affinities.

CICS AFFINITIES

Affinities arise because CICS transactions are allowed to leave behind “footprints” in the particular CICS region where the transaction executed. These footprints are expected to be found by subsequent transactions, and thereby affect the execution of subsequent transactions. Examples of these footprints include updates to the CWA (common work area), Temporary Storage queues, REQID processes, etc. Sometimes the problems arise because a transaction issues an ENQ/DEQ sequence to serialize some process. Since the scope of the ENQ is just the one CICS system, and is not known in the cloned CICS systems, this leads to integrity exposures.

Collectively, these and other parts of the CICS API cause CICS transactions to develop “affinities” for a particular CICS region. Determining the duration of these affinities is difficult; they can last for the entire VTAM session, the life of a particular CICS, or other periods of time.

Affinities essentially mean that transaction routing for workload balancing becomes a secondary objective, as the DTR program is forced to deal with routing transactions to the CICS with the proper affinities. Workload tends to be skewed badly as this occurs, and little or no benefit is seen. Also, since the CICSes are essentially no longer clones because of the footprints, failure of a CICS causes an outage. This occurs because all transactions that had an affinity for the failed CICS must wait for emergency restart of that CICS to complete in order to back out all recoverable resources of the failed in-flight transactions. By removing affinities, failure of a CICS or MVS system would hardly be noticed because another CICS would take over for the failed system. See Figure 1.

IDENTIFYING AFFINITIES AND THEIR SCOPE

IBM provides the CICS Affinities Utility (CAU), which helps identify affinities and their scope. This utility has four major parts:

- scanner;
- detector;
- reporter; and
- builder.

For more information on the CAU, see the CICS Transaction Affinities Utility User’s Guide CAUAA101.

The Scanner

The scanner runs as a batch utility against the application load libraries and searches for affinities. It does this by examining the individual object programs, looking for patterns that match the argument zero format for the commands in question. This shows the module names that invoke EXEC CICS commands that could cause affinities. Some of the affinities it shows are ENQ/DEQ, Temporary Storage, CWA, LOAD, RELEASE, DELAY, POST, START, and CANCEL. There are some instances of these commands that do not cause affinities. The scanner will also not show any affinities for programs that start with CAU or DFH. After the scanner has executed and its output has been reviewed, the detector can be used to determine affinity lifetimes.

Determining Affinity Lifetimes With the Detector

The detector is run against an active CICS system and searches for affinities and their scope. It consists of a control transaction (CAFF), an autosave transaction (CAF), some global user exit programs, and a task-related user exit program. All the user exits...
are designed to co-exist with other user exits your shop may be running. With these programs, the detector intercepts all EXEC CICS commands and other events that are needed to determine the affinities, their relations and lifetimes. The data is saved to VSAM files for later use by the reporter and builder. The performance of the detector is the same as running a third-party CICS monitor product. After applying maintenance or a new application, you should rerun the detector to make sure you catch all the affinities. Now comes the time to review the output of the detector with the reporter.

Reviewing the Output
With the Reporter

The reporter is a batch utility that converts the affinity data collected by the detector into two output formats. The first is a report presenting the affinity data in a readable form. This assists the systems programmer and application designer in determining what affinities are real. This report should be used with the scanner report to assist you in learning your transaction workload and determining what affinities really exist and their lifetimes.

The second output format is a file containing a set of basic affinity transaction group definitions in a syntax approximating to the batch API of CICSPlex/SM. This is input into the builder. Input from the reporter is provided for one CICS region at a time, and may not show all the possible affinities. This occurs because workload patterns such as month end, year end, and quarter end may cause different execution paths in the monitored transactions. These all produce different transactions that could affect the affinities. All situations must be studied closely using the reporter and the scanner output. Now that the workloads are known, the builder can be run.

The Builder

The builder is a batch utility that is run against a set of files containing the basic affinity transaction group definitions as created by the reporter. The builder outputs a file containing combined affinity transaction group definitions suitable for input to CICSPlex/SM. This file will then become the basis for Dynamic Transaction Routing based on affinities, with workload balancing a secondary priority.

Whether you utilize CICSPlex/SM or your own program, now you have the necessary information about your transactions and affinities to build your DTR to take advantage of parallel processing. However, this could become very complicated if you have some cloned AORs. You must keep track of what is running where in order to send another dependent transaction to the correct CICS. Another way to make your DTR simpler is to use a Temporary Storage Queue Owning Region (QOR). This region can be used like a FOR for temporary storage only. This removes Temp Store affinities from your DTR.

Another solution would be to utilize a third-party product. One example is Affinities Server for CICSPlex from NEON Systems, Sugarland, Texas, which removes affinities for all Temporary Storage, ENQ/DEQ, REQID, and CWA processing. It does this by utilizing the MVS Sysplex coupling facility. The coupling facility consists of both hardware and software. The hardware is comprised of non-pageable memory, processors, a Sysplex timer, and two or more coupling facility links, which are high-bandwidth fiber optic connections to each MVS. The software consists of an assembler-based macro interface which allows applications to interface with one or more types of the three structures available in the coupling facility:

- lock structure;
- list structure; and
- cache structure.

Each structure allows a specific set of operations to be used for Sysplex operation. These structures allow sharing of data in a Sysplex with integrity and high-performance. Up to eight coupling facilities may be connected to each MVS. Many structures can be built inside each coupling facility. In addition, multiple CMOS processors can be configured for each coupling facility and additional links can be installed, providing for a truly high-performance system. Up to 32 MVSes may be connected to a coupling facility. Many MVS components (i.e., VTAM, JES2, RACF, IMS, DB2, etc.) already utilize the coupling facility.

Utilizing the coupling facility Affinities Server gets around affinities such as ENQ/DEQ by placing a lock in the coupling facility on the ENQ name. See Figure 2. This “globalizes” the resource so any other ENQ will wait as if the ENQ was issued on... applications such as CICS, which are essentially limited to the speed of a single processor, will suffer throughput degradation in a CMOS hardware environment unless their architecture is restructured.
the same CICS. Utilizing a product like this allows you to clone your CICS regions and use your dynamic transaction routing program to route mostly on availability and workload requirements with very little consideration for affinities. See Figure 3.

IBM has offered a glimpse of the future by stating that the next release of CICS/ESA will support Temporary Storage sharing for non-recoverable user queues and DFSMS VSAM record level sharing. Shared Temporary Storage processing will be a significant overhead improvement over using the QOR, and VSAM RLS will help VSAM accomplish parallel processing as IMS and DB2 do today. To date there are no other announcements related to CICS affinity problems.

**THE FUTURE**

Other vendors are also working on the Parallel CICS problems. In the next year or so you will see third-party DTR programs, products that remove affinities, and monitors that help manage CICS in the Sysplex world. Along with new knowledge of our applications, these products will allow us to uncomplicate our DTR, allowing it to balance on workload and availability. When this is accomplished, we will start to see the real advantages of highly parallel Sysplex processing: availability, performance and scaleability.

---

**Kevin Cogley** is a product developer for NEODN Systems Affinities Server product. He has 10 years of commercial MVS software experience working with Ultraopt/VTAM at BMC Software and Multsess with Westinghouse/C&K Software. He is CICS/ESA project manager for GUIDE, and speaks at user groups and conferences around the world.

©1996 Technical Enterprises, Inc. Reprinted with permission of Technical Support magazine. For subscription information, email mbrship@naspa.net or call 414-768-8000, Ext. 116.