IBM CICS Tools:
Unrealized Productivity Gains and True Cost Savings

June 2009

Branham
BRANHAM GROUP INC.

100 Constellation Crescent, Suite 915 • Ottawa, ON • Canada • K2G 7E6
Tel: 613.745.2282 • Fax: 613.745.4990 • www.branhamgroup.com

Executive Summary

The resurging popularity of the mainframe or IBM System z platform has seen a significant consolidation of new and varied workloads on the platform. Customers need to take every opportunity to reduce processing requirements including the tuning of existing applications. This optimization can help postpone processor upgrades and with it, the expense of additional software licensing costs.

Of the same vein, the increase in new workloads has put tremendous demand and strain on existing system and application programmer staff. Supporting this staff with the right tools can help them complete their tasks more quickly and efficiently, ultimately decreasing the need for additional, increasingly expensive and difficult to find resources.

The challenges expressed by organizations experiencing these growing pains have prompted Branham to investigate some of the possible solutions to address these challenges. One of the most ubiquitous applications for System z is the IBM CICS Transaction Server, being deployed in almost every mainframe environment. This ubiquity allowed Branham to focus on solutions that would provide the widest possible value for System z customers.

With the ability to provide hardware, software, and services through a single vendor solution, Branham chose to look towards the CICS tools available from IBM. In co-operation with IBM, Branham was able to interview various existing IBM CICS Tools customers, focusing in particular on IBM CICS Interdependency Analyzer, CICS Performance Analyzer, and CICS Configuration Manager.

Of the customers interviewed, IT environments and experiences were largely unique. However, there were common value propositions for System z customers that emerged. The following provides just a snapshot of the cost savings these customers indicated were achievable.

- 75% time savings for the identification, coding, and testing of CPSM rules, in some cases equating to the recovery of a whole month in time savings for a major affinity.
- Upwards of 90% time savings to identify and validate typical performance issue related changes.
- An average of 66% less time to administer CICS Service Definition changes, which happen on a daily basis.
- The recovery of 2%-15% of CPU cycles through the designation of CICS applications as threadsafe.

Whether replacing existing vendor tools or home grown solutions, these customers continue to find additional value and savings that “alone justify the cost of the IBM CICS tools.” This report includes a summary of some of the potential cost savings. Given the ubiquity of CICS, this report is designed to address a varied audience, without being overly technical. With this information, readers will be empowered with the knowledge to extend the value and impact of these findings, helping cut costs associated with hardware, software, and personnel.

Introduction

The mainframe, or IBM System z platform, has been an integral part of the business fabric for many years and today is seeing a resurgence in its use for cost control and reliability, among additional factors. This renewed interest is in stark contrast to the practice of migrating from the mainframe to distributed systems for a “perceived” cost delta. Over time, as distributed systems have grown in size and complexity, users have started to see an escalation in personnel hours, maintenance requirements, and environmental costs. The movement back to the mainframe has helped to significantly reduce these associated costs. While the mainframe is not void of maintenance related costs, studies are showing that the number of personnel hours is often significantly less.

Similarly, the mindset towards application development and systems management for the System z platform can be quite different from that of its distributed counterpart. Specifically, in the distributed world, inefficient programming that results in slow performing applications will often simply have additional hardware thrown at it to improve response times. While this additional hardware may be inexpensive, it again increases costs around personnel, maintenance and management.

In contrast, the frame of mind towards mainframe application development is to tune applications for optimal performance, thus reducing hardware requirements and application response times. Similarly, the advancement of management tools has also helped to relieve the amount of time associated with application and system management, allowing application developers and system managers to focus on new business initiatives, even with shrinking budgets.

In conjunction with mainframe implementations, many large organizations have been using the IBM Customer Information Control System (CICS) since the early 1970s. In fact, CICS is nearly ubiquitous, being installed at almost every System z site. Used for developing, running and managing transaction applications on the mainframe, these systems have grown and evolved with the business over time.

The purpose of this report is to provide customers with additional input on some of the intricacies of the mainframe, while identifying areas where applications can be tuned and productivity benefits realized. This discussion will focus on the ubiquitous CICS platform in particular, and look towards areas where customers can further reduce costs through system optimization.

Mainframe Cost Benefit Factors

It's no surprise that businesses today, continue to push the "do more with less" mentality. With the number of mergers and acquisitions in which the largest organizations partake, this way of thinking becomes increasingly difficult to support. Add the growing demands of regulatory compliance while maximizing system availability and performance, and the demands placed on the IT staff are significant.

Organizations need to know that they have taken every opportunity to optimize their applications in efforts of controlling the costs associated with hardware upgrades and third party software. The following outlines a variety of areas that should be considered as part of a cost mitigation strategy for not only a CICS environment, but also the broader mainframe.

Cost Considerations

» **CPU Time:** Some organizations may already have a value associated with CPU time in their environment. This is often simply the cost of the CPU lease per month. The ability to tune applications to help decrease CPU usage during peak times can have significant value, particularly if it helps postpone pending new processor purchases, which themselves would also have added software licensing costs. Important to note is that CPU savings are only realized if the saved processing time could be used for something else. Specifically, saving 10 minutes of CPU time from a job that runs at 3 am in a site where the nightly batch cycle completes an hour before needed is not really saving anything. However, saving this time from peak period transactions in a CICS region that is constrained at 11am can provide significant savings.

  Further, it should also be recognized that even the smallest of changes can still produce significant savings. For example, a large US banking institution stated that "one CICS transaction that was tuned only saved fifteen seconds from the original, except that 15-20 tellers in 500 branches executed this transaction every hour. So 20 times 500 equals 10,000 transactions per hour, times 15 seconds equals 150,000 seconds every hour. Over an 8 hour day, that's 5.5 hours in CPU savings!"

» **Software:** Many organizations are uncomfortable with the thought of the increased costs that are associated to third party software when CPU upgrades are required. While these costs may be fairly difficult to determine without a lot of effort, it is very important for organizations to do so. Software costs may be one of the most significant savings that can be achieved. In some sites, this amount is equal to the CPU costs, while in others it may be up to double the CPU costs. Even as a percentage of the overall CPU costs, these costs should not be ignored, as the savings can often easily justify the costs associated with the tools used to postpone the perceived necessary processor upgrades. Delaying a processor upgrade can save organizations thousands of dollars in delayed software costs.

» **Personnel:** While hardware costs continue to decrease, personnel costs continue to rise. Providing application and system programmers with the tools necessary to complete their tasks more productively not only helps to reduce the demands on these personnel, but also helps them fulfill various requests, such as system changes, more rapidly. Similarly, users waiting for these changes to be implemented can get back to work sooner. While the intention may not be to eliminate personnel, providing them with more efficient tools could allow them to put more focus on new business initiatives, thus leading to increased competitive advantage.

» **Outages:** Making changes to applications without understanding their relationships with, and effects on, related applications is a recipe for failure, which can subsequently lead to application outages. Downtime for large organizations can be quite expensive, and the ability to eliminate or reduce their duration is significant. Many may not understand the magnitude of such outages. Taking into consideration consequences such as lost user productivity, lost IT staff productivity, lost revenue, overtime payments, wasted goods and material, and imposed fines or penalties, average hourly impacts due to outages can be upwards of millions of dollars depending on the industry. Tools that help automate system changes and quickly back out of erroneous changes can significantly help reduce unplanned outages based on human error.

» **Customer Value:** Customer satisfaction and user time as further cost consideration, while more of an esoteric savings, is difficult to measure. In many cases, organizations will be able to find enough savings in the hard resources to justify most optimization efforts and may not need to investigate the requirement to put a dollar value on customer savings. However, depending on the type of business and the nature of the competition, this may be a requirement. Stories abound about customers leaving their brokerage firms, or banks, or Internet providers, etc., because of poor response times. Customers today have more options and less patience. In short, the primary goal of optimization may not be the delaying of a processor upgrade, but rather an improvement related to response times during peak hours to keep and attract new customers.

Licensing Considerations

IBM has continued to innovate and maintain the relevancy of the System z platform, while trying to help reduce costs for its customers through advancements in not only the technology itself, but also how it is licensed. From the hardware perspective in particular, each generation of the System z platform continues to see a technology dividend reduction of approximately 10% MSU (million CPU service units). Specifically, new hardware is cheaper for the same number of MIPS. In other words, customers are receiving more MIPS per MSU. For example, on the z900, 1 MSU was equivalent to about 5.9 MIPS; the z990 provided approximately 6.6 MIPS per MSU; and the System z9 provides about 7.3 MIPS per MSU. Therefore, software for a 580 MIPS machine will be charged at a rate of 81 MSUs. Ironically enough, purchasing the latest hardware can actually help decrease overall software costs.

Also, the advent of the Workload License Charge provides sub-capacity software pricing where fees are linked to the actual capacity used by software with the LPAR where it runs. This ensures customers pay only for the software they actually use. Unfortunately, customers may still be paying too much, and optimizing applications can still have a dramatic effect on overall software costs even when paying through this sub-capacity pricing model. To help provide a better understanding of this important note, a clear understanding of Variable Workload License Charges (VWLC) is required and how it compares to the licensing of the entire capacity of a Central Processing Complex (CPC). Specifically, it's important to understand what "CPU usage" means.

For VWLC, CPU usage is calculated based on the MSU measurement unit. Every 5 minutes the Workload Manager (WLM) inside each LPAR calculates a 4-hour rolling MSU average. At the beginning of each month, the Sub-Capacity Report Tool (SCT) provides the LPAR utilization, being the highest sum of the measured 4-hour rolling MSU average for the LPARs in the CPC. Charges are not based on the product CPU utilization in the LPARs, but rather on the sum of the utilization of all the LPARs where the products were running.

Table 1 shows an example of how the SCRT calculates the LPAR utilization capacity for different products and LPAR combinations.

Table 1: LPAR Utilization

4-hour rolling average			z/OS			CICS			IMS		
Hour	LPAR1	LPAR2	LPAR1	LPAR2	SUM	LPAR1	LPAR2	SUM	LPAR1	LPAR2	SUM
1	100	333	100	333	433	0	333	333	0	333	333
2	120	321	120	321	441	0	321	321	0	321	321
3	112	345	112	345	457	0	345	345	0	345	345
4	118	402	118	402	**520**	0	402	402	0	402	**402**
.	.	.	.	.	.	.	.	.	.	.	.
718	127	348	127	348	475	127	348	**475**	0	0	0
719	133	299	133	299	432	133	299	432	0	0	0
720	122	300	122	300	422	122	300	422	0	0	0
	MAX				**520**			**475**			**402**

IBM CICS Tools

IBM CICS Interdependency Analyzer (IA) for z/OS is a runtime tool that automatically builds a comprehensive database of CICS resource relationships. It enables development teams to understand quickly and easily which transactions, programs, maps, and other resources will be affected by planned changes. Inexperienced users can quickly understand what a CICS region contains, what resources a transaction needs in order to run; which programs use which resources; and which resources are no longer used.

IBM CICS Performance Analyzer (PA) is an offline reporting tool that complements real-time monitoring solutions, analyzing a wide range of System Management Facilities (SMF) records to produce comprehensive reports on all aspects of CICS system performance. Whether planning, building, managing, or deploying complex mainframe CICS applications, CICS Performance Analyzer makes it easier to improve CICS related system performance.

IBM CICS Configuration Manager (CM) wraps a software solution around CICS resource definitions and the migration of these definitions so that they can be managed in a controlled fashion. Working equally well with CICS CSDs and Data Repositories under CICSPlex System Management (CPSM), it can migrate changes seamlessly between these environments. It also integrates with change management software that may be in place by using a batch interface to migrate the resource definitions.

The MAX value is the highest utilization determined from the sum of the utilization for all LPARs in which a particular product ran in a given hour. It is not the sum of the highest utilization for individual LPARs in which a particular product ran during different hours.

In this example, the peak value of the month for z/OS is in hour 4. It is the sum of the z/OS utilizations across all the LPARs during hour 4, or 520 MSUs. The peak value for CICS is in hour 718. IMS ran only in the first part of the month in LPAR2 so its utilization value is 402 MSU.

In the above example, note that running IMS in the same partition as CICS also increases the IMS usage in that LPAR, making it more expensive because IMS will be paid for on its own consumption plus the MSU already used by CICS. Reducing CICS related overhead therefore, will also help decrease those consumption costs related to IMS helping reduce costs not only for CICS, but also other VWLC software running in the same LPARs which might include DB2, WebSphere MQ, IMS, Lotus Domino, and z/OS itself.

With a clear understanding of the Variable Workload License Charge, organizations will further understand how application tuning and optimization can have dramatic cost savings for organizations whether paying for full capacity and delaying processor upgrades, or if paying for only what they use through VWLC. Similarly, helping application and system programmers be more productive can also help organizations invest less in the maintenance of existing assets and spend more time on new innovative initiatives. While there are many areas these optimizations might be applicable, one of the most ubiquitous areas is around CICS based applications.

Mainframe Optimization with CICS Tools

In an effort to further understand some of the cost considerations and current challenges that customers may face in a mainframe environment, Branham Group interviewed a variety of IBM CICS customers to assess their ongoing use of the tool sets and what steps they had taken in efforts of realizing productivity improvements and ultimate cost savings. In specific, input was taken from over a dozen users of IBM CICS tools including CICS Interdependency Analyzer, CICS Performance Analyzer, and CICS Configuration Manager.

Throughout these interviews, two major themes presented themselves: reduced CPU cycles through conversion to threadsafe applications, and increased productivity through added time savings. The conversion of CICS applications to threadsafe was noted to provide the most significant cost benefit for these customers, while increased productivity was also experienced, incorporating a number of different perspectives and esoteric benefits not as easily calculable.

Reduced CPU Cycles through Conversion to Threadsafe Applications

This particular topic may be one of the simplest yet significant changes that organizations can make, providing some of the largest benefits when optimizing the performance of their CICS applications. Although some will be familiar with this topic, many may not understand the tremendous value that the movement of CICS applications toward threadsafe may have. To really portray this value, a little parallel education is required.

A Task Control Block (TCB) is a z/OS control block which represents a dispatchable unit of work. In short, TCBs represent tasks, such as an executing user program. CICS has only one TCB referred to as the quasi-reentrant (QR) TCB that is shared by all CICS tasks. This facilitates “free” serialization to share resources such as files, data sources, etc. This means that although the same program can be executed by multiple CICS tasks, only one of those CICS tasks is active at any one point in time. Processing these applications in parallel could cause issues.

To protect against these issues, application code was “automatically” serialized because it had to be dispatched on the single QR TCB. Since there could not be multiple application programs running simultaneously within CICS, the QR TCB provided the protection mechanism for shared resources. However, this limited the multi-processing capacity of CICS to specialized functions and in many installations, the QR TCB reflected 80 to 95% of the total CPU used by the CICS address space.

Through the advent of open TCBs, CICS applications can now be dispatched to run in parallel (simultaneously) with the QR TCB providing a possible reduction in CPU usage and increased transaction throughput. This parallelism now presents the situation where two different tasks running simultaneously on separate TCBs could require access to the same resources. If there is no synchronization, the result could produce invalid results or compromise data integrity.

A threadsafe program requires built in mechanisms (e.g., ENQ/DEQ) to ensure serialized access to a shared resource to avoid data integrity problems. It is important to understand that a single program that operates without the agreed-upon serialization technique(s) can affect the predictability and integrity of an entire system of otherwise threadsafe programs. Therefore, an application system cannot be considered threadsafe until all programs that are sharing common resources implement appropriate standards.

If not defined as threadsafe, then a TCB switch from the open TCB to the QR TCB must occur in order for the application to continue, and there is a CPU price to pay for this switch. Specifically, a TCB switch costs around 2,000 instructions. CICS will automatically switch to the QR TCB for a non-threadsafe command. With a high percentage of threadsafe commands within a threadsafe application, the thread of execution can remain on an open TCB for longer and so improve the performance of the system.

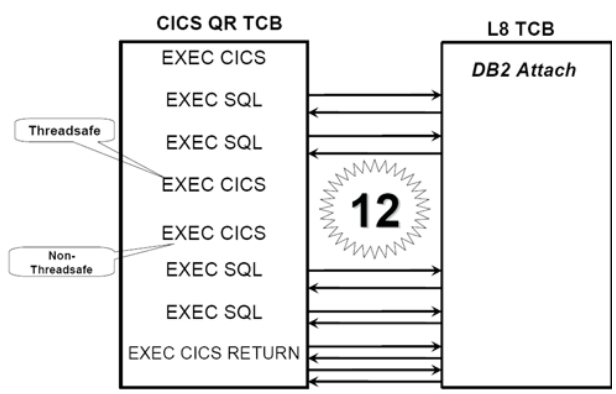

Figure 1: TCB Switches for CICS Apps Not Yet Defined as Threadsafe

To better illustrate the impact of this cost, consider that if a task was executing on the QR TCB when a DB2 (or MQ as of CICS TS 3.2) request was made, then a switch to an open TCB is made at a cost of 2,000 instructions. Once the request has completed, the program returns to the QR TCB when the program issues a further CICS API command, incurring a further cost in CPU instructions. The reason that the request is assigned to the open TCB is so that it can handle the request and communication with DB2 (or WebSphere MQ), and the QR TCB can continue to dispatch other work.

A round trip from the QR TCB to the open TCB and back would represent around 4,000 instructions. The same could apply to many exits in use by the CICS system. Therefore, it is imperative that all exits be converted to threadsafe or the CPU utilization may increase.

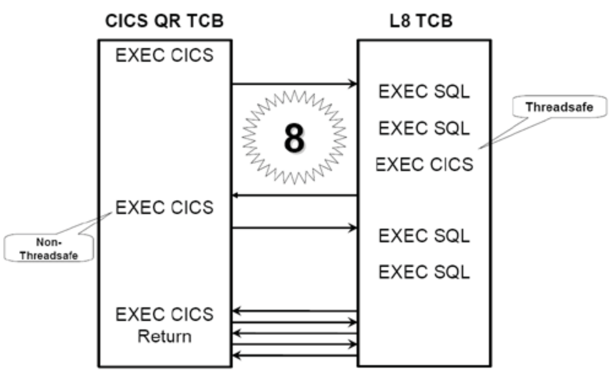

Figure 2: Reduced TCB Switches for CICS Apps Defined as Threadsafe

More importantly, further performance gains to CICS response time and throughput without processor upgrades can be achieved through the use of threadsafe programming, as more and more API and SPI commands are made threadsafe with each new release. Processing as many tasks as possible on the open TCBs will remove constraints on the QR TCB and reduce the response times of both threadsafe and non-threadsafe transactions. Please note, that it is highly recommended that threadsafe enablement should be completed with proper analysis to avoid any potential program conflicts and/or outages.

How important is this? Based on the z9 redbook value, one (1) million TCB switches waste 6.96 CPU seconds, and for one (1) billion switches, 116 CPU minutes are wasted. Similarly, one billion TCB switches on a z10 processor will waste 46 CPU minutes. In one example, a customer had ½ billion TCB switches in their environment every day, which provided significant return on investment when designating their CICS applications as threadsafe. An average projection is a savings between 2 – 15%. In another instance, a major US bank saved 700 MIPS by making only one major application threadsafe.

Table 2 provides an example of a leading brokerage firm and its documented savings, which were realized when it converted four of its modules to threadsafe.

To help organizations convert their CICS applications to threadsafe, IBM CICS Interdependency Analyzer provides insight into existing programs and their relationships with other transactions and resources. It provides the facilities to assist in the movement to a threadsafe environment by analyzing the application to uncover data integrity exposures, show which programs can be mode threadsafe immediately, determine which exits are threadsafe, and to ensure application changes do not regress an already defined threadsafe environment. Similarly, IBM CICS Performance Analyzer can provide answers to questions such as: Which TCBs a transaction used? How many TCB switches occurred? What was the switch delay time? How much Dispatch and CPU time did they use? Why did my transaction take so long?, etc.

Calculating Wasted CPU

Interested in calculating your wasted CPU due to TCB switching? The average number of TCB switches per transaction is available through SMF 110 (CICS CMF) records, obtainable through any tool that reports on SMF data. The TCB switch information is obtained from group DFHTASK, class 248. The field name for this class is CHMODECT before CICS TS 3.1 and DSCHMDLY for CICS TS 3.1 and higher.

Assuming a z9 and z10 process is .58 and .23 nanoseconds per cycle respectively, and the average instruction required 6 cycles, each TCB switch requires 2,000 instructions. The only two variables required are the number of TCB switches per transaction and the number of times that transaction is executed in a single day. The calculation is as follows:

$$\frac{\text{2000 Instructions}}{\text{1 switch}} \times \frac{\text{6 cycles}}{\text{1 instruction}} \times \frac{\text{.58ns (or .23ns)}}{\text{1 cycle}} \times \frac{\text{1 second}}{\text{1000000000ns}} \times \frac{\text{1 minute}}{\text{60 seconds}} \times \frac{\text{\# TCB switches}}{\text{1 transaction}} \times \frac{\text{\# transactions}}{\text{1 day}} = \frac{\text{wasted CPU minutes}}{\text{day}}$$

Table 2: Brokerage Industry Threadsafe Conversion Savings								
Modules	**Non-Threadsafe**				**Threadsafe**			
	Response Time	CPU	TCB Switch Count	Wait Time	Response Time	CPU	TCB Switch Count	Wait Time
Balances by Account	0.0477	0.0185	56	0.0045	0.0507	0.0173	8	0.0020
Balances by FC Core #	0.0826	0.0319	88	0.0056	0.0478	0.0286	12	0.0013
Holdings By Account	0.2922	0.0946	622	0.0762	0.2298	0.0801	378	0.0284
Holdings by Group #	0.3409	0.1009	650	0.1065	0.2564	0.0908	380	0.0358
Modules	**Savings**				**Percentiles of Reduction**			
	Response Time	CPU	TCB Switch Count	Wait Time	Response Time	CPU	TCB Switch Count	Wait Time
Balances by Account	0.0031	0.0012	48	0.0025	6.42%	6.59%	85.7%	56.3%
Balances by FC Core #	0.0348	0.0033	76	0.0043	42.11%	10.42%	86.4%	76.2%
Holdings by Account	0.0624	0.0145	244	0.0477	21.36%	15.35%	39.2%	62.7%
Holdings by Group #	0.0845	0.0101	270	0.0707	24.78%	10.04%	41.5%	66.4%

For many customers, the financial cost of running their applications is related directly to the amount of CPU consumed. Under these circumstances, the CPU savings gained by migrating appropriate applications to a threadsafe environment can equate to a significant financial saving.

Increased Productivity through Added Time Savings

While the conversion of CICS applications to threadsafe can provide a calculable benefit to customers, Branham's interviews also revealed a number of additional benefits that provide cost savings in other ways, including productivity benefits, the virtual elimination of errors, help with meeting new governance requirements, and more. Some of the more commonly noted examples included the following:

» **Affinities and CPSM Rules:** CICS transactions use many different techniques to pass data from one to another. Some techniques require that the transactions that are exchanging data must execute in the same CICS region and therefore impose restrictions on the dynamic routing of transactions. If transactions exchange data in ways that impose such restrictions, there is said to be an affinity between them. IBM CICS Interdependency Analyzer will identify affinities that would prevent the combining of CICS regions and/or the movement of resources to a different CICS region. CICS Interdependency Analyzer also provides the facilities to generate the necessary CICSPlex System Manager (CPSM) rules required to activate transaction routing while maintaining the existing transaction affinities. For a site moving to dynamic transaction routing, this feature can reduce the time to production while eliminating the risk of transaction abends. In fact, during the Branham interviews, customers noted a savings upward of 75% for the time required to identify, code, and test CPSM rules. For a process that can take several business days for major affinities, this provided customers with significant time savings.

Major Affinity Example Savings

The average time required for a system programmer to identify, code, and test CPSM rules for a major affinity can take upwards of 200 hours. For reference, major affinity rules apply to transactions with resources that exist for a unit of work, plus system or transaction affinities for persistent resources such as a user login state. A 75% savings based on this 200 hours is a 150 hour reduction. That's approximately a month or 20 work days that a system programmer now has available to focus on other initiatives. For a paid consultant, at a rate of $65/hr, this one example could provide a savings of almost $10,000.

- **Performance Tuning and Analysis**: IBM CICS Performance Analyzer provides the facilities for performance bench marking before and after changes to validate improved optimization. On average, this tool provided 50% time savings when system programmers were tasked with the identification and validation of performance related changes. For example, simple performance analysis could be reduced from 2 hours to 1. Similarly, complex performance related issues were reduced from 48 hours to 24 hours. Some of the most significant benefits seen were around the more typical performance related issues which in some cases saw upwards of a 90% decrease in time required to identify and validate performance changes. Using IBM CICS Performance Analyzer, customers were "better able to improve performance" and "better able to diagnose the root cause" of performance issues after a recovery had been achieved.

- **Automated Subsystems Management:** Every CICS region refers to a set of resource definitions. Typically, organizations maintain each of their CICS regions in at least three separate environments: development, test, and production. Changes to resource definitions are migrated from development to test, and then from test to production. As shown in Figure 3, even if each environment contains only one CICS region, this means three sets of resource definitions, and two migrations are needed to move each change into production. IBM CICS Configuration Manager is designed specifically to help reduce the time required to migrate changes to CICS System Definition (CSD) files, and provides the ability to track and back out changes if the results are not satisfactory.

Several of the customers that were interviewed had implemented IBM CICS Configuration Manager. Interesting to note was that many of these customers already had home grown solutions to help ease the complexity of migrating CSDs. While the average IBM CICS Configuration Manager customer saw a time savings of about 3:1, those customers with home grown solutions or existing vendor products did not see this time savings. So why the purchase?

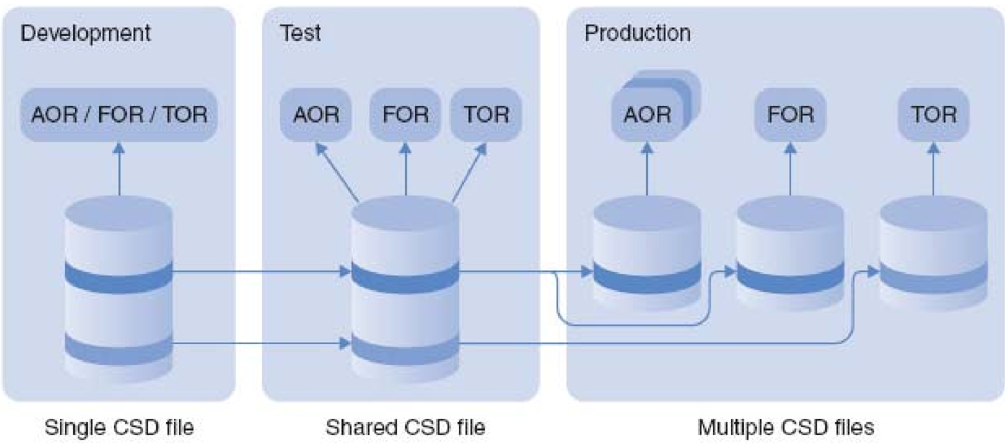

Figure 3: Migration of CSDs

Beyond the initial time savings provided by CICS Configuration Manager, or its cost savings when used to replace an existing vendor implementation, many of these customers were after, or found value in, the additional facilities. Specifically, a number of customers interviewed looked specifically at the auditing capabilities. A number of customers noted that this provided "the biggest bang for the buck" with regards to purchasing CICS Configuration Manager. This provided value on a number of fronts including, but not limited to, compliance related requirements, tight and transparent control over changes, the ability to quickly compare proposed changes for accepting/declining self-service requests, and the reduction in errors and downtime by rolling back erroneous modifications. Some customers admittedly have found tremendous value in the rollback feature "more than once." Based on the already discussed costs associated with downtimes, this can provide significant value in and of itself, regardless of the time savings associated with definition migrations.

CICS Configuration Manager Time Savings

CICS Configuration Manager customers, on average, saw a time savings of 3:1. Since changes can range from 15 minutes to a few hours, a couple of scenarios may help to provide some additional perspective to this value. On average, customers were required to make 6 CSD changes per day, with over 12% of them being complex (25+ resource definition changes). The following provides two example scenarios.

CICS Resource Definition Scenario 1: RENAME 50 FILE definitions

CEDA Process = ~2 hrs

» VIEW FILE(F1*) GROUP(NG), ALTER FILE(F1*)] x 50
» Create change management record
» Re-enter description of all resource definitions in change management record
» Forward change record for approval

CICS CM Process = ~10 min

» Create PACKAGE
» VIEW FILE(F1*) using filter
» Select filtered entries, use REN * and overtype new names
» ADD selected FILEs to PACKAGE
» Forward PACKAGE for approval

CICS Resource Definition Scenario 2: New Application

CEDA Process (30-45 min)

Assuming PROGRAM and MAPSET AUTOINSTALL is being used:

» DEFINE GROUP(NG)
» DEFINE TRANSACTION(NT) GROUP(NG)
» DEFINE FILE(NF) GROUP(NG)
» Create change management record
» Re-enter description of all resource definitions in change management record
» Forward change record for approval.

CICS CM Process (10-15 min)

» Create PACKAGE
» DEFINE GROUP(NG)
» DEFINE new TRANSACTION(NT) using template model, change name
» DEFINE new FILE(NT) using template model, change name
» ADD GROUP, TRANSACTION and FILE to PACKAGE
» Forward PACKAGE for approval

Even recovering a conservative 15 minutes per task, 6 times a day, could provide an additional one and a half hours a day in savings (i.e., one entire work day/week or over one month/year). Further automation through self-service capabilities returns even more time to existing staff, eliminating the potential need for new personnel.

- **Eased Learning Curves:** Customers noted that the IBM tools help them with respect to reducing learning curves in a couple of ways. For example, CICS Interdependency Analyzer helps "unravel the complexity" allowing new or inexperienced developers to quickly learn existing applications, including the relationships between the different components. Second, CICS Performance Analyzer comes with more than 150 reports designed to meet a customers reporting and analysis objectives. Customers can easily tailor these reports to their specific analysis requirements or create their own reports using the easy to follow ISPF dialogs. These prebuilt reports and the ability to easily create custom reports were reported across the board as being easier than alternatives such as SAS, MXG, etc. Specifically, there was a significantly shorter learning curve. A large financial institution stated that through CICS Performance Analyzer, they were able to "become more capable in things like tracking transactions, that were otherwise frustrating" with other third party tools.

- **CICS Explorer:** During Branham's interviews with customers, many noted the desire to implement and use the new CICS Explorer, which was recently released for production environments (in June 2009); customers were already noting the perceived benefits of this new CICS feature. As such, it is important to introduce this tool here for completeness of vision. IBM CICS Explorer is an Eclipse-based graphical tooling interface for CICS. In short, it is a new Eclipse-based systems management environment for application programmers, system programmers, and other operations staff. It provides a modern interface to CICS that is highly customizable and can be integrated with information from additional third party tools, providing users with a single Master Terminal. Existing and skilled mature staff can be augmented with less-experienced staff thanks to this youthful and dynamic interface that can be run from anywhere through a browser. As an Eclipse-based solution, no individual seat charges apply, there is no Microsoft Windows installer, or any requirement for Windows administrative authority.

  One particular customer noted that it saves "time jumping from one tool to another in a single view." Customers can use "Performance Analyzer to uncover a performance problem. From the CICS Explorer graphical display, one click can link to Interdependency Analyzer data to show what resources are used in the poor performing transaction. Selecting a resource can link to Configuration Manager to view the resource definitions," which can then be updated with Configuration Manager.

Through the conversion of CICS applications to threadsafe and increased productivity through added time savings, organizations can witness significant cost savings overall. Pursuant to this, there are also other benefits that come with the latest in CICS tooling. For example, while the companies interviewed by Branham were not yet to this point, IBM CICS Interdependency Analyzer can also help identify application code that is no longer used. Decommissioning this code can contribute to cost savings through alleviated maintenance costs long after these tools have paid for themselves.

Summary

Owing to its ubiquitous nature, CICS provides mainframe customers with significant opportunity for cost restraint and increased productivity gains. Tuning these mainframe applications can provide dramatic savings, when these modifications help to postpone CPU upgrades. The savings associated with charges for third party software can be tremendous. Even those organizations licensing software based on Variable Workload License Charges still have a significant opportunity to reduce associated software costs.

One of the most cost effective techniques with respect to optimization and cost control can be the conversion of CICS applications to threadsafe. This allows the use of parallel processing to decrease CPU consumption, simultaneously helping satisfy impatient users and/or customers through reduced response times.

CICS Tools such as IBM CICS Interdependency Analyzer, CICS Performance Analyzer, and CICS Configuration Manager also help to increase overall productivity. Some examples articulated in this report include the automation of processes such as the generation of CICSPlex System Manager rules, or the automated migration of CSD alterations between CICS environments. Similarly, organizations can offload actual CSD change requests to application programmers, alleviating load from System Programmers, while still maintaining control of changes through an approval process, all while supporting compliancy auditing.

Finally, new interfaces such as those provided by IBM CICS Explorer provide a younger inexperienced generation of operational personnel with more familiar tools helping to ease learning curves. New users are provided with the facilities to quickly get up to speed with existing applications, and/or deal with resource changes in a timely manner; all through a single interface. This intuitive visualization helps developers modify or extend CICS applications with confidence.

Organizations need to know that they have done everything they can to optimize their existing applications to help postpone added costs that come with new hardware purchases. This tuning can have significant ramifications with respect to additional software purchases. Organizations also need to know that customers are being provided with the best response times possible to help prevent defection to competitive alternatives.

In short, organizations should consider the latest in IBM's CICS tooling as part of any investment plans for the future. There is significant value that can be achieved through these types of tools, with an immediate return on the investment. Based on real world examples with broad relevance, it is clear that there are several benefits to working with the latest in IBM's CICS tooling.

About Branham Group Inc.

Branham Group is a leading industry analyst and strategic consulting firm servicing the global information technology marketplace. Branham Group assists information technology companies and related institutions in achieving market success through its custom consulting services (Planning, Marketing and Partnering), and through its multi-client research subscription programs (eHealth, Outsourcing, Green IT and Wireless). Branham also produces an annual listing of the top information technology companies in Canada (www.branham300.com), tracks the Canadian Outsourcing industry (www.branhamoutsourcing.com) and monitors over 450 eHealth vendors.

For more information regarding Branham Group, please visit www.branhamgroup.com.

Branham Group Inc.
100 Constellation Crescent, Suite 915
Ottawa, ON
Canada K2G 7E6
Tel: 613.745.2282
Fax: 613.745.4990
www.branhamgroup.com