THE HIDDEN COST OF TCP/IP ON Z/OS

Prepared by Bill Yeager, Chief Technical Officer, Alebra Technologies Inc.

Mainframe systems have traditionally used channel protocols (CTC) to exchange data, thereby realizing performance and efficiency benefits. Most enterprises use communications networks (TCP/IP) to exchange data between homogenous and heterogeneous systems. This paper will contrast the performance and processor impacts on z/OS™ with the use of TCP/IP for high speed data transfer, provide empirical data and a measurement methodology, as well as describe alternative mechanisms to dramatically reduce processing while increasing performance for high volume data movement.

Introduction

This paper will present and discuss:
1. The assumptions which spawned this project.
2. Background information regarding the z/OS™ TCP/IP stack.
3. The methodology used to validate the assumptions.
4. The results achieved.
5. Why data networks are better equipped to transport large amounts of data between systems.
7. A mechanism whereby others can benefit by what has been learned.

The Challenge

Alebra Technologies was involved in projects with several different customers who expressed concern that (i) significant z/OS processor resources were devoted to file transfer over TCP/IP and that (ii) a lot of the resulting processing expenses were not being charged back to end users. Most of these customers were using both FTP (client and server) and Connect:Direct™ from Sterling Commerce. In one case, the additional load on the z/OS processor resources was so substantial that the customer shut down file transfer activity during peak times in order to devote maximum processor capacity to online applications.

Alebra was asked to provide an analysis tool that would help quantify the magnitude of the problem in order to justify architecture and process change as well as a methodology for charge-backs that was fair to all of their users.

The Approach

In order to address this challenge, we first reviewed all available IBM™ documentation. Using these results from IBM’s studies, we were able to estimate the impact of file transfer over TCP/IP on z/OS processors. We found that, although performance enhancements have been made from release to release, there is still significant resource consumption for file transfer over TCP/IP.

On the basis of our resource impact estimates, we contacted Dr. H. Pat Artis of Performance Associates to establish a methodology for quantifying z/OS resource consumption. Key to the effort was the need to develop a
simple method of presenting the results as well as a method for predicting and modeling across all enterprises which use z/OS.

Later in this paper we will present the methodology that was developed; however, it is necessary to precede the methodology with some background information.

TCP/IP Socket Processing

When a z/OS application is reading from or writing to the TCP/IP stack over a socket, there are activities that are performed by the stack that can be directly associated with and therefore recorded as CPU time in the callers Task Control Block (TCB). There are also asynchronous activities that run under a Service Request Block (SRB) that can accrue to the caller's Address Space total SRB time. Both of these would appear under job accounting and become available for charge-back.

There is additional processing that the stack must perform to move data to and from stack buffers and adapters as well as to manage the state of each connection. This processor time is accumulated under the TCP/IP address space and is not charged back to the socket owner's address space.

The z/OS FTP client and server are examples of such socket applications. The z/OS FTP client and server both log SMF records (type 118 and 119); however, these records do not contain processor usage. They are logged for each transfer operation and contain (among other things), the total byte count of the transfer. Because it executes as a z/OS job, the FTP client does charge some of the CPU time to the z/OS user; however, when the FTP server is executing, it is doing so on behalf of a client running in the network, so there is no time charged to a z/OS job.

It becomes apparent that the easiest way to accurately account for this time is to come up with a formula which ties the total processor usage by the stack to the amount of data that each user moves.

Measurement Methodology

All benchmark runs used standalone processor environments with no other workloads present. To achieve the objective of measuring all CPU utilization associated with file transfer, both from the transfer task and any operating system overhead, Resource Management Facility (RMF) data that recorded the total CPU seconds consumed by all work (type 70) was collected. The CPU Data Section of this record contains the wait time accumulated per processor in the SMF70WAT field.

For systems running in an LPAR mode, the wait time must be adjusted by data found in the PR/SM Partition Data section to account for time the processor was not dispatched for this system. Subtracting the PR/SM adjusted wait time from the total available CPU time (Interval Duration X Number of Processors) yields total CPU time consumed.

A z/OS system will consume some processor resources even when “idle”. This CPU usage is obtained by measuring the CPU consumed (see Type 70 record above) for a number of “idle” intervals. Once calculated, the “idle” consumption is subtracted from the Total CPU time of active intervals to remove normal Operating System activity from the calculations. CPU time is then converted to millions of instructions by multiplying the CPU time in seconds by the single processor MIPS rating of the machine. After the total number of instructions is calculated, this value can be expressed as instructions per unit of work (path lengths) by dividing the number of instructions by the units desired. The unit of work we selected was megabytes transferred. The value for total megabytes transferred was obtained from the Type 42 subtype 6 SMF records.

The following formula calculates the CPU path length, expressed as millions of instructions per megabyte:

\[
\frac{(TCPU) - (ICPU)}{MIPS \times TRATE} = \text{ Millions of Instructions per Megabyte}
\]

Where:

- TCPU = Total CPU seconds recorded during the period of file transfer
- ICPU = Measured CPU seconds when machine is idle for the equivalent period
MIPS = Machine performance rating in Millions of Instructions per Second
TRATE = Transfer rate in megabytes per second

Our testing consisted of fourteen runs. Seven runs pushed data from z/OS to a target system. Another seven runs pulled data from a source system to a z/OS target system. The seven runs varied logical record length (LRECL). The file block size (BLKSIZE) was set to half-track blocking of 3390 disk format, the most common blocking factor used in z/OS environments. The following table shows the combinations of LRECL and BLKSIZE used.

<table>
<thead>
<tr>
<th>LRECL</th>
<th>BLKSIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>27920</td>
</tr>
<tr>
<td>133</td>
<td>27930</td>
</tr>
<tr>
<td>2048</td>
<td>18432</td>
</tr>
<tr>
<td>4096</td>
<td>24576</td>
</tr>
<tr>
<td>8192</td>
<td>24576</td>
</tr>
<tr>
<td>16384</td>
<td>16384</td>
</tr>
<tr>
<td>27998</td>
<td>27998</td>
</tr>
</tbody>
</table>

The z/OS FTP client was used in batch mode for these tests.

**Measurement Results and Analysis**

The seven runs did not produce results that were significantly different, so the average is presented here:

- FTP from z/OS (push) – 1.02 Million Instructions/MB
- FTP to z/OS (pull) – 1.25 Million Instructions/MB

These numbers compare very closely to what IBM achieved in their tests (see link above).

By consulting the job accounting for the FTP clients, we find that only 50% of the CPU cycles are captured. A methodology to derive CPU cycles per transfer would therefore involve capturing the type 118 or 119 SMF records, using the byte count, direction of transfer and the MIP rating per engine as follows:

- **Push Operation:** CPU Cycles = \( \frac{1.02 \times \text{Megabytes}}{\text{Engine MIPS}} \)
- **Pull Operation:** CPU Cycles = \( \frac{1.25 \times \text{Megabytes}}{\text{Engine MIPS}} \)

Note that these results only reflect a single implementation of a socket file transfer solution (FTP). We believe FTP to be a very efficient implementation.

**Alternative Data Movement Techniques**

Using the above numbers, one sees that to drive a single GbE adapter at capacity (70-100MB/second) requires around 100 MIPS. This is significant, especially considering the explosive growth in data volumes and the effect that faster and multiple adapters might cause on the processing capacity of a z/OS system. In essence, it was found that what was suspected at the "Challenge" phase of the project was validated.

There are alternatives; however, they are based on a different approach to data movement that does not use TCP/IP. These use data networks and I/O protocols, in lieu of sockets. The following comparison of network approaches serves as background to exploring these alternatives.

**Network Comparison**

Communications networks (CNs) consist of NICs (Network Interface Cards), hubs, routers, switches, etc. and are for the purpose of providing communications mechanisms between servers in the enterprise and client applications. These networks use TCP/IP as the underlying transport protocol.

Data networks (DNs) consist of HBAs (Host Bus Adapters), directors, switches, etc. and are for the purpose of providing access to data on storage media from servers. On mainframes, the terms ESCON™ (Enterprise Systems Connectivity) and FICON™ (Fibre Connectivity) are used to refer to these networks. On UNIX™, the
terms SAN (Storage Area Network) and fibre channel are used. Within this paper, we use the DN acronym to refer to all of these.

CNs tend to be serial in nature and have a small number of connections per server while DNs tend to be massively parallel and have a great number of connections per server. The latest zSeries systems from IBM have up to 1024 FICON connections. The performance of CNs is unpredictable and dependent on workload and the type of traffic; whereas, given a specific configuration, DNs deliver consistent, predictable performance.

For performance reasons, most companies spend significantly more (orders of magnitude) on their DNs than their CNs. Getting the most out of a DN means getting the most out of the processor, since the data access rate is directly related to response time. For that reason, these systems are very sophisticated and involve caching within the processors as well as outboard in disk arrays. SLAs (Service Level Agreements) rely very heavily on consistent DN performance.

While DNs have been improved substantially in both speed and technology, CN improvements have been limited to increases in available bandwidth and improved routing techniques, error handling, and new standards relating to the internet.

The handling of CN traffic within processors has always been CPU-intensive. While the use of TCP/IP has become the standard, most of the work needed to implement the protocol is still done within processors by software in a single stack.

By contrast, from the earliest days of computing, most of the work done to implement DNs occurred within the components of the DN itself in hardware and not within the general purpose processors. From the first IBM System/360™ mainframes through today's zSeries™, channel processors manage I/O. Once an application initiates an I/O operation, the channel takes over and handles direct memory transfer to and from the device. In this way the processor continues to perform other tasks until the I/O completes. zSeries systems from IBM also have System Assist Processors to offload I/O completely from the main processors. UNIX systems, as well, have on-board processors within HBAs to handle DMA (Direct Memory Access) to and from the fibre channel network.

Data Movement

No business of any substantial size has a single server. Generally, the availability of applications or cost dictates that multiple platforms are used. This results in the need to move data between processors; because, typically, the processor that generates and manages a set of data is not the processor that exclusively uses this data.

In the early days of mainframes, IBM implemented a way to move data between DNs without crossing the boundary into CNs. This was done via a Channel-to-Channel Adapter (CTCA) which allowed for direct I/O links between processors so that data could be transmitted efficiently using all of the power of the DN. Today, these are implemented within FICON as FCTCs (FICON CTCs) that still provide a very efficient mechanism for connecting two or more z/OS processors.

Conversely, in the UNIX world, FTP and managed file transfer solutions are used to move data over CNs. The world struggled for many years (and still does) to solve the problem of co-existence of interactive traffic with file transfer traffic. In essence, the presence of DN traffic on the CN is corrupting the original objective of having a CN. Configuring networks for both data and interactive traffic is therefore problematic.

As mainframes running z/OS and high-end UNIX servers collide, most businesses choose the common denominator (TCP/IP) as the means to move data between them, because they believe it is the only way.

As data volumes increase and transfer windows decrease, many businesses look at any and all alternatives to solve data movement issues. They add multiple CN connections to elevate bandwidth but find that the cycle utilization necessary to drive these networks at capacity is excessive. They may turn to very expensive software solutions that reduce the amount of data that needs to be moved but create significant complexity in their daily processing and recovery procedures.

Some disk vendors have delivered alternatives that involve moving data through a disk drive, thereby keeping the data within the DN; however, these solutions are vendor-specific. The solution discussed here is disk vendor-
neutral and also provides connections between z/OS systems and UNIX systems which allows data to move over the DN.

**Analysis of CN vs. DN**

In order to evaluate the effect of using a DN, identical runs were performed comparing FTP with Alebra’s Parallel Data Mover™ (PDM) combined with the z/OpenGate™ FICON-FC gateway. Two environments were added to the test conducted above: A FICON CTC connection between two z/OS images and a FICON-FC connection between a z/OS image and a UNIX image.

Following are the results:

- CTC from z/OS (push) – .20 Million Instructions/MB
- CTC to z/OS (pull) – .20 Million Instructions/MB
- z/OpenGate from z/OS (push) - .26 Million Instructions/MB
- z/OpenGate to z/OS (pull) - .24 Million Instructions/MB

The following chart summarizes all results:

![Chart](image)

**Figure 1, Measurement results**

**Summary**

Enterprises moving data between z/OS mainframes and other servers are probably not aware that there is significant associated cost. A single GbE connection running at capacity requires almost 100MIPs of z/OS processor. This is significant enough that either or both of the following should be considered:

- A charge-back system should be put in place to account for this time.
- Alternative methods should be considered.

Networks are getting faster and z/OS network adapters will keep up with the growth in speed. These changes are taking place in order-of-magnitude leaps: 10Mbit to 100 Mbit to 1Gbit to 10 Gbit. Improvements in TCP/IP stack processing are not taking place at the same rate. As enterprises deploy z/OS into these networks, the movement of data becomes a chronic and increasingly resource consuming challenge.

**Trademarks**

Connect:Direct is a registered trademark of Sterling Commerce, Inc. IBM, z/OS, ESCON, FICON, zSeries, and System/360 are trademarks or registered trademarks of International Business Machines Corporation. UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group. Parallel Data Mover and z/OpenGate, are trademarks of Alebra Technologies Inc. All other products, trade names, and service marks are trademarks, registered trademarks, or service marks of their respective owners.