



SyncSort's Exploitation of System Facilities in the z/OS Environment

© Syncsort Incorporated, 2009

All rights reserved. This publication may not be reproduced in whole or in part, in any form, except with written permission from Syncsort Incorporated. All other company and product names used herein may be the trademarks of their respective companies.

SyncSort's Exploitation of System Facilities in the z/OS Environment

Introduction

SyncSort for z/OS is a high-performance sort/merge/copy utility designed to exploit the advanced facilities of the z/OS operating system and zSeries computers. Three facilities recently introduced by IBM are Parallel Access Volume (PAV) technology, the Modified Indirect Address Word (MIDAW) facility and the System z9 Integrated Information Processor (zIIP). This paper examines the creative means by which SyncSort for z/OS exploits these facilities to reduce CPU time and elapsed time.

PAV

What Is It?

Parallel Access Volume technology was introduced to relieve the problem of UCB I/O serialization. Before the advent of PAV, jobs issuing I/O requests to the same DASD (mapped by a single UCB) could experience delays caused by the system limitation which allows only one active I/O request to transfer data between a DASD and the CPU.

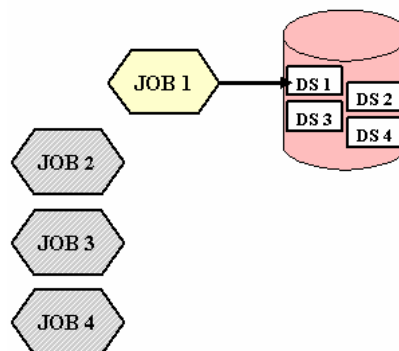


Figure 1: System activity on a system without PAV active

PAV technology creates separate logical images of a DASD as alias UCBs. This technology directs I/O requests to alias UCBs instead of queuing the I/Os if the primary UCB is in use. Thus, multiple jobs can perform I/O concurrently to data sets residing on the same volume. This increases system throughput by decreasing I/O wait time. PAV technology has been described as the most important advance in I/O performance since caching.

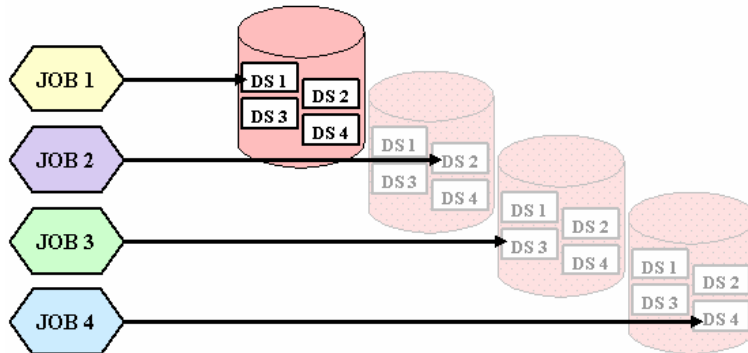


Figure 2: System activity on a system with PAV active

How Does SyncSort for z/OS Exploit PAV Technology?

While all jobs on a system can benefit from PAV technology, SyncSort for z/OS goes a step further. SyncSort developers created new I/O techniques to take advantage of the ability to issue multiple I/O requests to the same device simultaneously. These new techniques are used to automatically drive multiple concurrent I/Os to any SORTIN or SORTOUT data set which is allocated on a PAV-capable DASD volume. These concurrent I/O operations complete faster than would otherwise be possible using more conventional serial I/O. This parallel processing can reduce the elapsed time of a SyncSort application up to 30%.

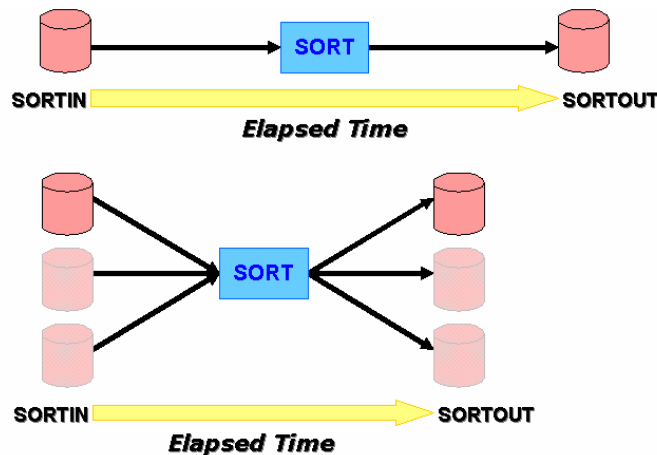


Figure 3: SyncSort's parallel I/O reduces elapsed time up to 30%

SyncSort handles the control and coordination necessary to manage these multiple concurrent I/Os. No user intervention is required. Users are typically not even aware that this processing is occurring. All they see is that their jobs with sorts are completing in a much shorter amount of time.

The original intent of PAV was to improve system throughput by reducing I/O queuing time. SyncSort's innovative use of PAV improves elapsed time in a way that is unique among sorting products.

MIDAW

What Is It?

The Modified Indirect Address Word facility is a modification to a channel programming technique that has existed for decades. An Indirect Address Word (IDAW) contains I/O addresses which are used by channel programs. The use of IDAWs is tailored to movement of records to or from contiguous buffer areas. With more complex channel programs, such as those which use data chaining to read or write physical blocks of data to or from non-contiguous buffer areas, CCW/IDAW structures are not as efficient, especially in today's I/O environments. Complex channel programs can generate significant amounts of CCW and control information traffic on the channels, thus reducing the amount of bandwidth available for transmitting data.

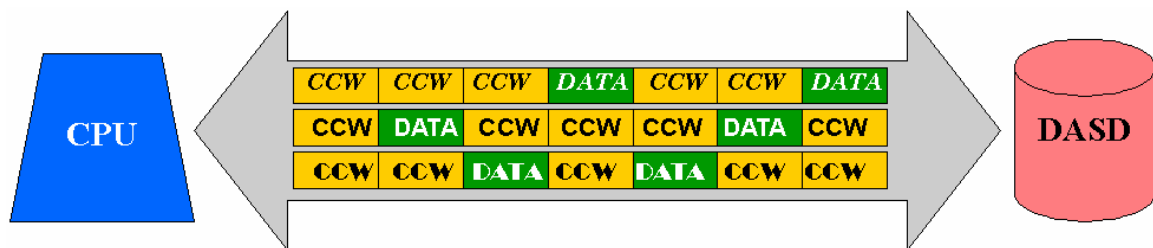


Figure 4: Channel activity without MIDAW

The Modified Indirect Address Word was created to address these inefficiencies. When MIDAWs are used, the amount of CCW and control information traffic can be significantly reduced, allowing more data to be transferred. This enhances the performance of a job's I/O processing, leaving bandwidth for other applications to use.

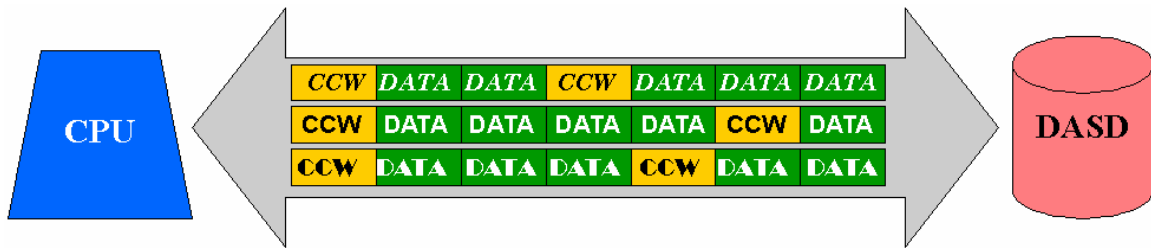


Figure 5: Channel activity with MIDAW

The use of MIDAWs makes FICON channels substantially more efficient at processing certain types of channel programs. DB2, Media Manager and PDS index processing are some of the IBM programs that benefit from the MIDAW facility.

How Does SyncSort for z/OS Exploit MIDAW Technology?

One of the ways SyncSort for z/OS derives its superior performance is from the use of complex and highly sophisticated channel programs. SyncSort has used the data chaining I/O technique to read and write with non-contiguous storage areas for buffers to provide substantial CPU time benefits. When the MIDAW facility became available, SyncSort developers immediately took advantage of the improvements this facility had to offer. Modifications to SyncSort's channel programs to incorporate MIDAWs result in additional performance savings of up to 10% in CPU time and 25% in elapsed time. SyncSort applications automatically use the MIDAW facility when possible, without any user intervention.

The savings achieved by SyncSort's use of MIDAWs extend beyond individual sort applications. SyncSort's use of MIDAW reduces the load of the channel subsystem. As with PAV exploitation, SyncSort's use of MIDAWs is unique among sorting products.

zIIP

What Is It?

The System z9 Integrated Information Processor is a specialty engine that is available on System z9 mainframes. It is intended to lower the overall cost of mainframe ownership. The machine cycles on these processors do not factor into the MSU calculation for the capacity of a mainframe. This can reduce system-wide software costs.

The Workload Manager (WLM) component of z/OS is responsible for directing work to the general purpose processors and the zIIP engines. The unique nature of the zIIP places restrictions on the type of work that this processor can handle.

Only a preemptible enclave Service Request Block (SRB) will be dispatched on a zIIP. This is a very specific and restrictive type of workload, but for software products that can conform to these limitations, the zIIP offers a way to lower CPU time costs.

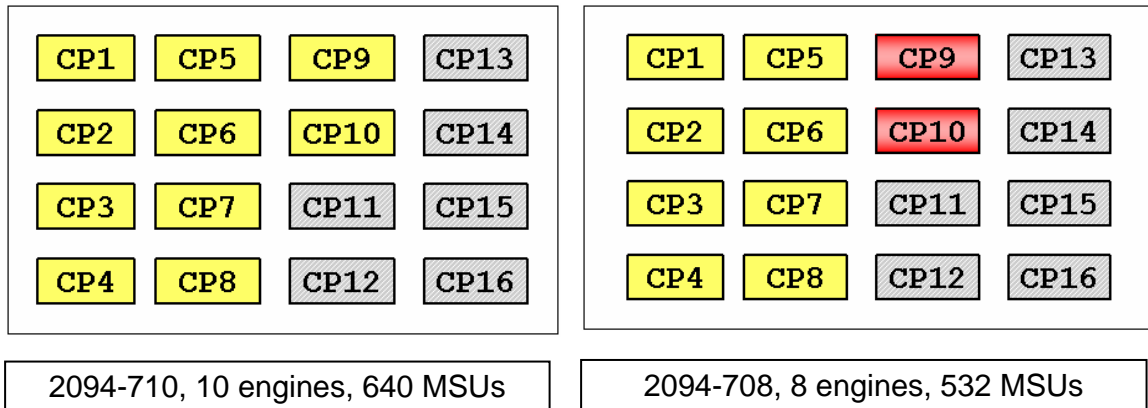


Figure 6: Configuration differences possible with zIIP

To demonstrate the power of the zIIP, suppose the workload of a particular company required a mainframe capable of 640 MSUs. This is represented by the configuration above on the left. If the work on that particular mainframe was such that it could be accomplished with only 8 general purpose processors and 2 zIIPs, the company would be able to license the configuration above on the right. The difference in MSU capacity between these 2 configurations is 108 or 16.9%. Since all software pricing on this machine is based on the reduced number of MSUs, savings can be realized on software products on this machine even if they are not able to exploit the zIIP.

DB2 was the initial IBM subsystem that exploited the zIIP. This exploitation significantly improves the performance of DB2 while lowering software costs across the board. It has proven to be very popular.

How Does SyncSort for z/OS Exploit zIIP Technology?

There are many processes that occur during a sort. During the input phase, the input files need to be read, records need to be de-blocked and manipulated, data utility functions need to be performed, the sorting process must begin and records need to be moved to SORTWK files. Similar processing occurs during the output phase. SyncSort achieves performance benefits by the seamless integration and overlap of these different processes. Some of these processes are CPU-intensive, some are I/O-intensive, and some are dependent on system services through SVC calls. The I/O and SVC processes are not eligible for zIIP processing.

In order to exploit the zIIP technology, SyncSort developers isolated and restructured some of SyncSort's more intensive CPU processes. By reorganizing code to separate functions that could be managed independently, the developers were able to create stand-alone units of work and repackage them as SRBs. These zIIP-eligible SRBs can be passed to WLM in a way that will allow them to be dispatched to a zIIP. All of this work is accomplished without the awareness or involvement of the user. SyncSort's use of the zIIP is available in Release 1.3.

Offloading these sort processes to a zIIP lowers the traditional CPU time cost associated with sorting. This decreases the MSU demand on the mainframe which can lead to substantial software savings.

Conclusion

Syncsort Inc. is committed to providing the best performance possible on evolving computers and operating systems. The exploitation of PAV, MIDAW and zIIP demonstrates the high level of technical sophistication built into SyncSort for z/OS.

Significantly, the performance improvements are automatic. SyncSort determines if a given application can benefit from using these features without requiring any modification to individual jobs.

If you would like further information on these or other issues related to SyncSort performance, please contact us at zos_tech@syncsort.com or call us at 201-930-8260.