Paper SAS1715-2015

More Data, Less Chatter: Improving Performance on z/OS via IBM zHPF

Lewis King and Fred Forst, SAS Institute Inc., Cary, NC

ABSTRACT

This paper describes how we reduced elapsed time for the third maintenance release for SAS® 9.4 by as much as 22% by using the High Performance FICON for IBM System z (zHPF) facility to perform I/O for SAS® files on IBM mainframe systems. The paper details the performance improvements, internal testing to quantify improvements, and the customer actions needed to enable zHPF on their system. The benefits of zHPF are discussed within the larger context of other techniques that a customer can use to accelerate processing of SAS files.

INTRODUCTION

To achieve optimum input/output (I/O) performance, the SAS product on the IBM mainframe operating system z/OS dynamically generates low-level I/O instructions (in the form of channel programs) to process SAS direct access bound libraries. For the types of I/O operations required by SAS processing, the channel programs generated by SAS provide a higher rate of throughput (amount of data per unit of elapsed time) than what is possible with standard IBM DFSMS access methods. In October 2008, IBM introduced High Performance FICON for System z (zHPF), which is a set of I/O architecture improvements that enable higher rates of I/O throughput (IBM 2015a). Over the next several years, zHPF capabilities were extended (IBM 2015a), and in October 2012, IBM announced functionality that enables programs like SAS to execute channel programs that take advantage of the zHPF feature (IBM 2015b). The second maintenance release for SAS 9.4, which became available in August 2014, can use zHPF channel programs for Read operations against direct access bound libraries. The third maintenance release for SAS 9.4 will use zHPF for both Read and Write operations to these SAS libraries.

Use of zHPF channel programs eliminates delays imposed by the channel command word (CCW) channel programs, a facility first provided in the System/360 architecture from which modern IBM z/Architecture processors descend. The zHPF channel programs reduce the amount of time that the I/O channel is connected to the I/O device, thus reducing the elapsed time required for I/O processing. In a battery of tests conducted at SAS Institute, the authors demonstrated that zHPF accelerated SAS jobs for which I/O operations are the rate-limiting factor. When using zHPF channel programs, SAS ran 22% faster overall than it ran with CCW channel programs. These improvements were observed using the default block size of 27K. The reduction in elapsed time was even greater for similar jobs using libraries with a block size of 6K.

This paper describes zHPF, the mechanism by which it improves performance for SAS I/O, and the IBM hardware and software levels required for SAS to use zHPF. The authors also discuss various techniques for analyzing the performance of I/O to direct access bound libraries and provide recommendations about how to obtain the best performance with zHPF and current I/O hardware.

I/O TECHNOLOGY

Accessing SAS direct access bound libraries in disk data sets on z/OS requires several supporting technologies, some that are part of the SAS product and some that are provided by IBM as part of z/Architecture or by the direct access storage device (DASD) vendor. This section surveys the SAS technologies as well as zHPF and related technologies developed by IBM.

FORMAT OF SAS DIRECT ACCESS BOUND LIBRARIES

The format of direct access bound libraries is designed to achieve the following design goals, some of which are detailed in Bowman (1989):

- Fully support all the types of members and features required by the BASE engine.
- Not impose architectural limits on the number of members, size of members, or the total size of the library (subject to any limits imposed by the physical media or operating system).
- Access member data directly, by member page number, without requiring preceding member pages to be read in sequence.
- Enable the library data set to be copied, backed up, or restored using ordinary system utilities without regard to the type of device.
- Reuse library space when a member is deleted or superseded by a new version.

The term *direct access* means that SAS can access blocks of the library directly by block number. In other words, locating a particular library block does not require sequentially reading preceding library blocks. The internal structure of a direct access bound library addresses blocks via a 31-bit block number that indicates the position of the block relative to the beginning of the library, regardless of physical boundaries such as tracks or volumes. Consequently, a library can be copied to a device with a different track size or to a different set of volumes (with different size DASD extents) without affecting the internal structure of the library (Bowman 1989).

The term *bound* means that the collection of members that constitutes the library is encapsulated in a single MVS data set. This design "binds" together a set of related SAS members and prevents them from being separated. The same system authorization facility (SAF) rules automatically apply to all members. The concept is similar to that of MVS partitioned data set (PDS), with the important distinction that z/OS operating system features such job control language (JCL) cannot be used to access individual members by name. In other words, the internal structure of the library is hidden from the operating system; therefore, specific members within the library can be accessed only by SAS software.

Note: MVS is the name of IBM mainframe operating system from which z/OS evolved, and it is the most widely recognized descriptive adjective for many z/OS operating system facilities.

The blocks of a direct access bound library can theoretically reside in any medium that allows direct access to fixed length blocks of data. For example, the hiperspace library implementation, supported by SAS on z/OS, is an in-memory form of direct access bound library in which the individual library blocks are 4K in size and reside in hiperspace pages for the duration of a SAS session. However, the subject of this paper is direct access bound libraries that reside on DASD in MVS physical sequential data sets.

The only architectural limitation on the size of direct access bound library is that imposed by the use of a 31-bit block number. However, assuming that the library is created with the recommend block size of 27K, 31 bits provides enough addressability for the largest library (about 49TB) that can be created on z/OS, subject to current operating system and hardware limits. To achieve this maximum, the following requirements must be met:

- The library data set must span 59 volumes, which is the z/OS maximum for DASD devices.
- DSNTYPE=LARGE must be specified when allocating the library data set. This allows up to 2^{24}-1 tracks per volume to be used.
- Extended Address Volumes (EAVs) must be used, and the EATTR=OPT parameter must be specified when allocating the library data set. This allows the library data set to occupy more than 64K cylinders per volume.

As detailed in Table 1, information about the disk space used by a direct access bound library might be obtained by submitting the `LIBNAME` *`libref`* `LIST` statement. The Directory statistics produced by PROC CONTENTS also include the same information.

Item of Information	Definition
Total Library Blocks	The maximum number of blocks that can potentially reside in all the DASD space currently allocated to the library data set on all volumes, up to and including the volume on which the last formatted block of the library exists. This does not include space that might be allocated on any subsequent volumes on which no library blocks have been written, a distinction that might be important when using pre-allocated library data sets.
Total Used Blocks	The total number of blocks that currently contain member data or information used by SAS to manage the internal structure of the library.
Percent Used Blocks	(Total Used Blocks / Total Library Blocks), expressed as a percentage.
Total Free Blocks	Total Library Blocks - Total Used Blocks. This statistic counts blocks that have not yet been formatted but potentially could be formatted in the allocated but unused space beyond the highest formatted block.
Highest Used Block	The library relative block number of the last block in the library that is used to contain member data or internal structures.
Highest Formatted Block	The number of the last library block that has been written. Additional DASD space beyond this block might be allocated to the library data set, but no blocks have ever been written beyond this point. Consequently, this statistic, when measured at the end of a job, can be used to determine the maximum number of blocks in the library used at any time in the past.
DSNTYPE	The type of MVS data set in which the library resides. Two values, BASIC and LARGE, are possible for a direct access bound library. BASIC data sets are limited to 64K tracks on each volume. DSNTYPE=LARGE are not subject to that limit, but DSNTYPE=LARGE must be specified when the library is originally created.

Table 1 - Library Space Information Items

The library space concepts described in Table 1 are illustrated in Figure 1. The particular library shown in this example has two regions of free space (rendered in light gray) that were previously occupied by member data. Since those free space regions once contained data, SAS must have formatted that space with blocks corresponding to the library block size. In contrast, SAS has not yet needed to use the space beyond the highest formatted block since the library has been created; therefore, SAS must first format that space with blocks of the appropriate size before that space can be used for SAS data.

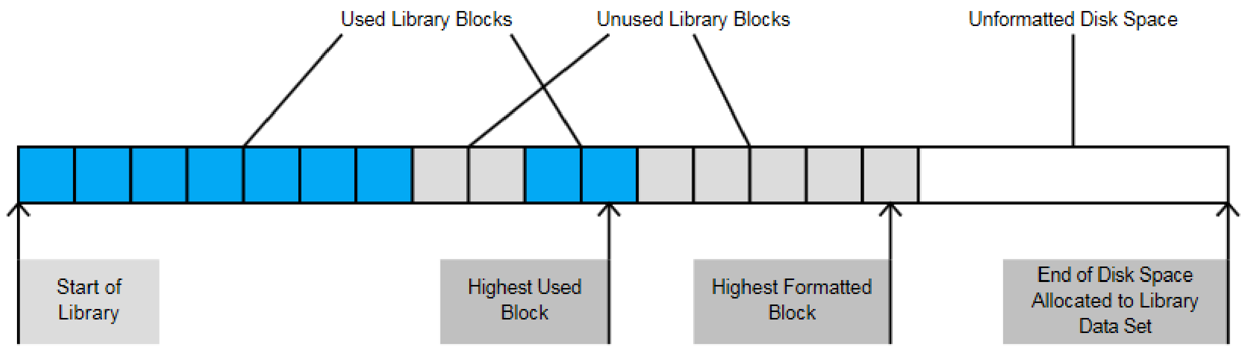

Figure 1 - Library Space Map

SAS CHANNEL PROGRAM GENERATION

Design Objectives Achieved by SAS Channel Programs

The method by which SAS stores data in a direct access bound library and subsequently retrieves the data is designed to achieve certain objectives. Table 2 summarizes these objectives and the design features that help achieve each objective.

Objective	Feature
Avoid the need for assistance from an MVS systems programmer.	SAS direct access bound libraries are stored in an ordinary MVS physical sequential data set, which can be created without special authorization or assistance from an MVS systems programmer, in most cases.
Minimize CPU overhead.	SAS generates channel programs that are specifically customized to the requirements of SAS processing. Channel programs execute within the channel subsystem, which is responsible for the task of transferring data to or from main memory locations in the processor. Central processor cycles are not used for transferring data.
Ensure high throughput and low latency for both random and sequential access patterns.	A single channel program generated by SAS can transfer data for multiple library blocks, even if the blocks reside in discontiguous locations on disk.
Extend the library data set automatically when additional space is required.	When insufficient free space exists within the library data set to write member data, SAS invokes the system End-of-Volume (EOV) service to request an additional DASD extent (range of contiguous disk tracks). This allows the library to grow seamlessly, provided that a secondary space allocation is specified. SAS can also extend the library data set to additional volumes.
Allow a server environment, such as SAS/SHARE, to interleave I/O and processing for separate client requests.	If one task within SAS, such as responding to a client request, requires an I/O operation, SAS can transfer control to another task and suspend the first task until the I/O operation completes. For each library member, SAS maintains a single shared pool of buffers regardless of the number of tasks are currently processing the member.
Permit multiple MVS jobs to share the library for Read access.	Update access to a direct access bound library requires exclusive access, specified by the allocation parameter DISP=OLD.

Table 2 - Access Method Design Considerations

The advantages of some design features listed above are best understood in comparison with alternative system facilities.

Basic Sequential Access Method (BSAM), included with z/OS, satisfies many of the preceding objectives for the limited requirements of simple application programs. However, it has the following limitations for the types of I/O operations performed by SAS:

- BSAM does have the capability to reposition its access to another point in the data set. However, the program must first wait for all outstanding I/O to complete (IBM 2013a, p. 311). Consequently, it is not possible for I/O to multiple discontiguous blocks to be scheduled at the same time. For example, the blocks that comprise a page of a SAS library member might be discontiguous. Closing a member after it has been written requires a number of updates to be the library structure to be saved to the library. These structures typically reside in discontiguous blocks.

- When opening a library or an individual member of a library, SAS typically loads information about the library structure into memory. If the structure changes, SAS replaces the contents of the blocks affected. BSAM requires that a block be read first before the block is updated. Channel programs avoid this requirement.

The z/OS UNIX file system and databases also achieve high throughput and low latency. However, these components, which can also be used to store SAS library data, do not share some of the other benefits of direct access bound libraries. For example, a database requires an administrator to configure and manage operations. The z/OS operating system natively supports a UNIX file system with robust caching, but transferring the data in storage between the file system address space and the user address space requires central processor cycles, which adds significant CPU cost.

Channel Architecture Overview

The *channel subsystem* is a component of an IBM System z processor that coordinates the transfer of data between I/O devices such as a DASD controller and the processor memory locations in the manner specified by a *channel program*. As described in Chapter 13, the channel subsystem "relieves CPUs [central processing units] of the task of communicating directly with I/O devices and permits data processing to proceed concurrently with I/O processing" (IBM 2012c). Figure 2 illustrates the relationships between these components. Since the channel subsystem has its own independent access to main storage, it can transfer data without the involvement of the central processors, designated in the figure by "CP".

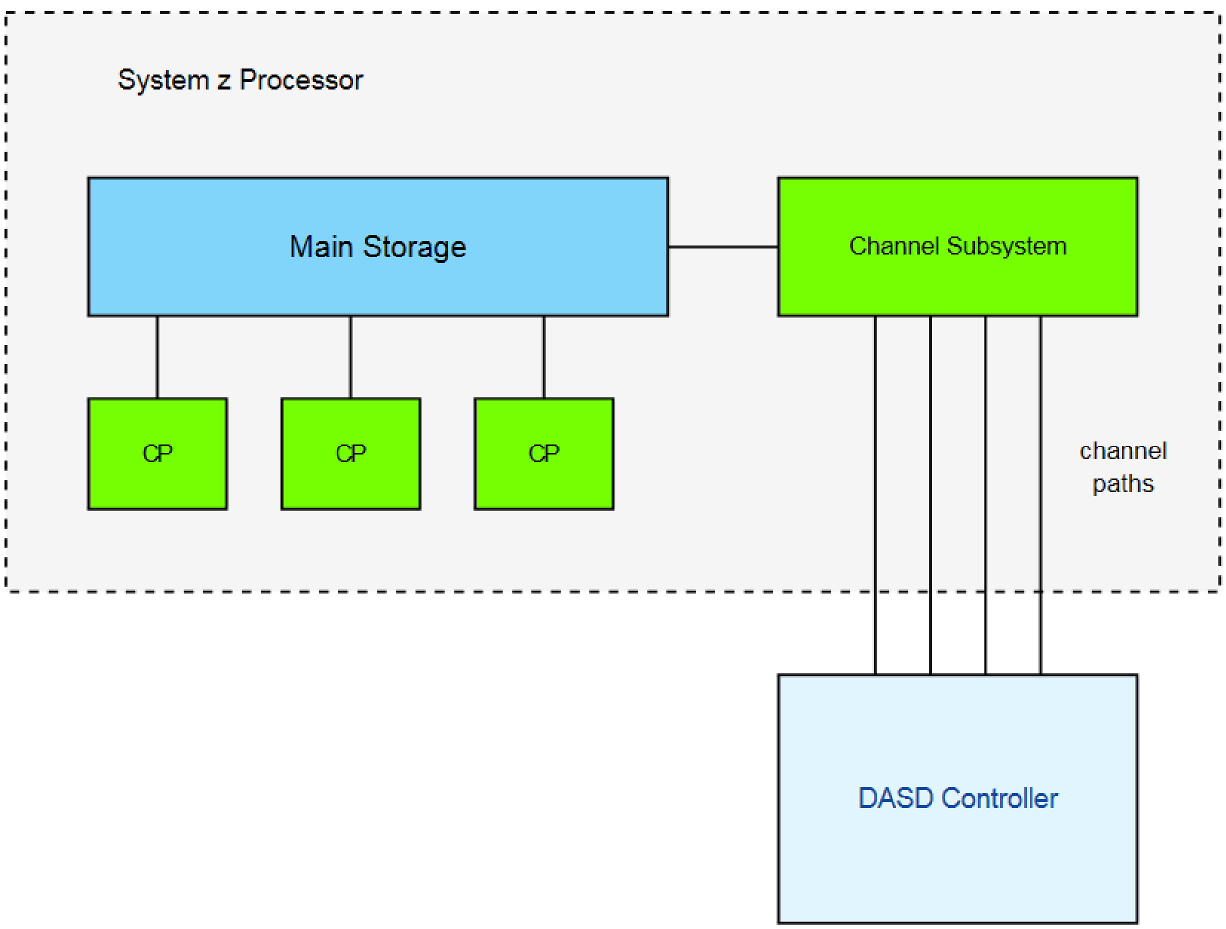

Figure 2 - Relationship of Channel Subsystem to Other Processor Components

As detailed in Chapter 11, "Introduction to Channel Programming" in Cannatello (1999), channel programs for DASD I/O specify the absolute disk addresses of the blocks to be read, updated, or formatted, and the locations in memory (main storage) to which data will be read or from which data will be written. SAS schedules its channel programs for execution by invoking the Execute Channel Program EXCP service described in Chapter 4, "Executing Your Own Channel Programs" (IBM 2013c). This sequence of events is illustrated in Figure 3.

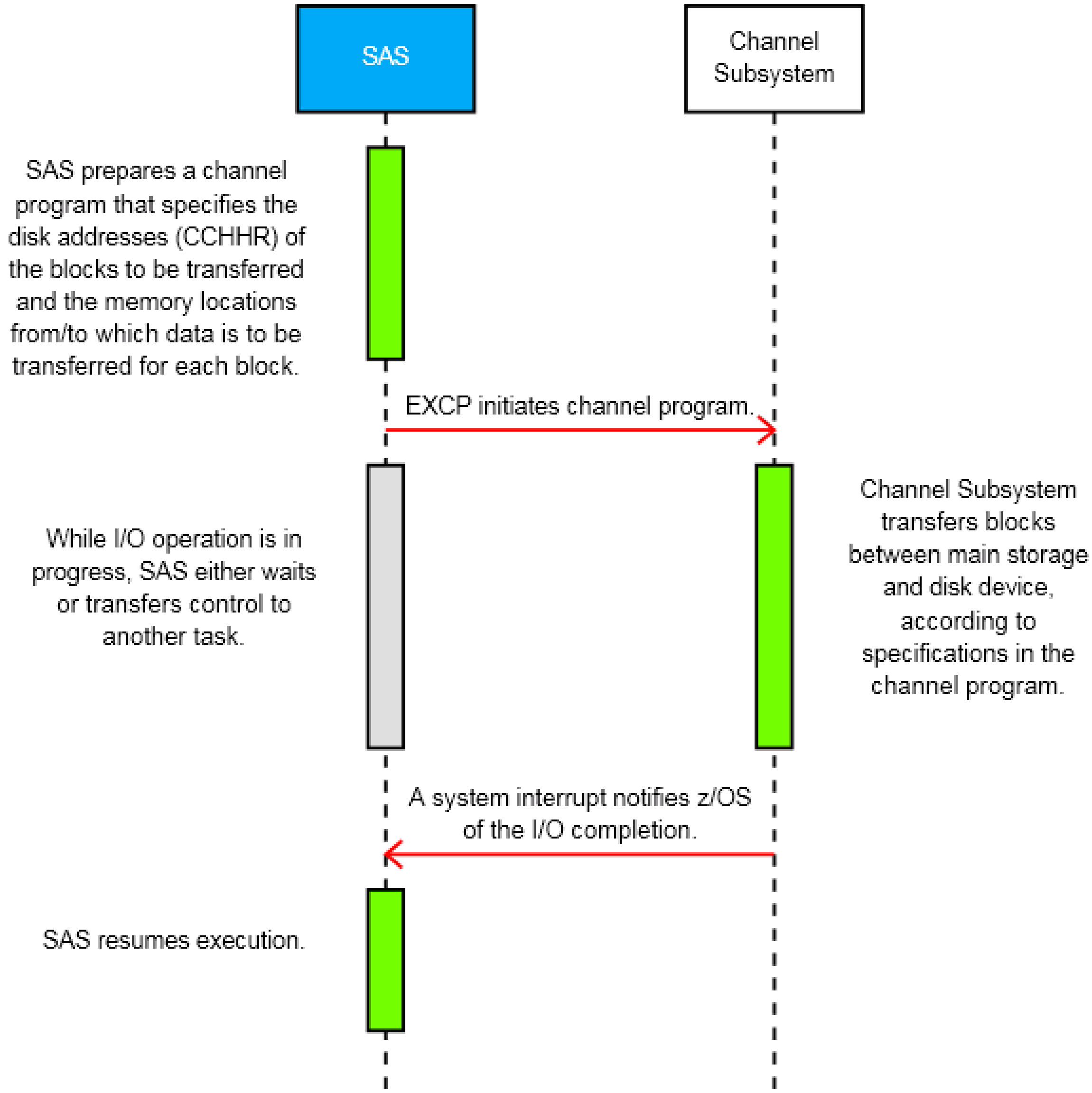

Figure 3 - I/O Operation

Disk Drive Track Format

The smallest unit of space allocation in an MVS DASD volume is a track. In the early days of disk hardware, a track corresponded to one of many concentric circles on the spinning platter of the recording medium. However, modern control units use virtual representations of a physical track, although the track concept maintains consistency with the large base of channel program generation logic (Houtekamer and Artis 1993).

A block is the smallest unit of data that can be accessed in an I/O operation. MVS DASD volumes allow each track to be formatted in a custom manner to contain blocks of arbitrary length, limited only by the track size. However, IBM standard access methods support a maximum block length of 32760, and the data libraries supported by SAS follow this restriction as well. To support this arrangement, each block, or "physical record" is further subdivided into one or more "areas" as described in Chapter 2, "Elements of data access" (IBM 2000):

- Count Area: This is an 8-byte area that describes the entire physical record. It contains the physical address of the physical record, which consists of the tuple (cylinder number, head number, and record number). The head number denotes the track within a cylinder, and it describes a physical representation that no longer exists on modern DASD, all of which have 15 tracks per cylinder. The record number enumerates the physical record, relative to the other blocks on the track. The count area also identifies the length of the key and data areas that follow. All physical records have a count area.
- Key Area: This optional area provides information by which a channel program can search for a physical record. This feature provided a useful way to offload processing from the host system in the early days of DASD technology, but it has been made obsolete by DASD advances. SAS does not use a key area in direct access bound libraries, so the key length is zero.
- Data Area: This area contains the actual data of the block. It is always present except for the EOF record, which the operating system automatically places as a marker after the last physical record in the data set.

Therefore, MVS DASD are called Count-Key-Data (CKD) devices, based on the names of these areas. This term distinguishes them from Fixed-Block Architecture (FBA) devices commonly used on other platforms.

When the operating system allocates space for an MVS data set, an application program such as SAS does not have any way of knowing how the physical records or blocks are formatted on the disk space. Consequently, SAS must issue special types of I/O instructions to format each track the first time the track is written. The contents of any individual block can subsequently be updated without having to issue special commands to format the track. SAS formats tracks in the following manner:

- Each block has the same length.
- Each track, except perhaps for the final track, is formatted with the maximum number of blocks of that size that will fit on the track.
- In most cases, all the blocks on the track are formatted in the same I/O operation.
- Tracks are formatted in a strictly ascending sequence, without skipping any tracks.

SAS maintains a record of the highest formatted block as it is extending the library data set. SAS also stores, in a system control block, the absolute address of the last formatted block so that the operating system can write an EOF record and record the relative track address of the last block in the data set label when the system closes the data set.

This method of formatting blocks complies with the "Standard Format" requirements of the MVS allocation parameter RECFM=FBS as described in Chapter 20, "Selecting Record Formats for Non-VSAM Data Sets" (IBM 2014). Since each track contains the same number of blocks, a simple algorithm can be used to convert a library relative block number to an absolute disk address. This allows SAS to directly access a specified block without requiring a sequential search of the library data set.

Channel Command Word Channel Program

When the System/360 architecture was designed, disk drive controllers lacked on-board processors (Houtekamer and Artis 1993). In this design, the sequence of events required to access the desired track, wait for the desired block to rotate into position, and to initiate the transfer of data are orchestrated, one action at a time, by a channel program in a form that is still supported at the current time. Each step in the I/O operation is specified by an 8-byte channel command word (CCW). Execution of a channel program involves an exchange of messages between the channel and the I/O device for each CCW. The channel transmits a CCW and transfers data (if required). When the device has completed the operation, it sends a channel end and device end (CE/DE) signal to the channel (IBM 2012c, p. 13-10). With IBM ESCON channels, the next CCW cannot be transmitted to the device until the CE/DE signal is received from the device (IBM 2012b, p. 28). The conversational nature of this protocol introduces delays because a full round-trip exchange (command and response) must take place for each command.

As described in Section 5.5 “ECKD Command Set for Disk Drives” in Houtekamer and Artis (1993), the addition of data caches to DASD controllers enabled the introduction of a new set of channel commands termed Extended Count-Key-Data (ECKD). This new command set exploited the greater processing power of the new cached controllers by specifying the I/O operation at a higher level of abstraction. For example, the steps to establish orientation to a particular block could be accomplished by a single CCW that specified the Locate Record command (IBM 2000).

The IBM System z FICON channel technology improves the performance of CCW channel programs relative to its predecessor, ESCON. “With a FICON channel, CCWs are transferred to the control unit without waiting for the first command response (CMR) from the control unit or for a CE/DE after each CCW execution” (IBM 2012b, p. 28). In addition, FICON allows multiple I/O operations to take place concurrently on a channel, with the data from separate I/O operations being interleaved in separate frames. This capability is not shared by ESCON (IBM 2012a, p.407).

zHPF Channel Program

As the preceding summary illustrates, the SAS product has continued to benefit from improvements in I/O hardware technology, despite using a programming interface that is decades old. This testifies to commitment IBM has to protecting the software investment of its customers and partners. Nonetheless, the structure of CCW channel programs limits the benefits that can be obtained from FICON technology. In particular, a separate FICON frame is still required for each CCW (IBM 2012b, p. 28).

In response to this limitation, IBM introduced zHPF, an architectural extension to FICON that includes a new type of channel program format. zHPF channel programs package all the commands for a given channel program into a single Transport Command Control Block (TCCB) (IBM 2012c, p. 13-7). This allows multiple I/O commands to be “… sent in a single frame to the control unit” (IBM 2012b, p.28). zHPF “…reduces the overhead on the channel processors, control unit ports, switch ports, and links by improving the way channel programs are written and processed” (Cronin, Entwistle, and Murphy 2011, p. 14).

PERFORMANCE IMPROVEMENTS

Begninning with the second maintenance release for SAS 9.4, the SAS product must identify the level of zHPF support that is available on the system. When assigning a direct access bound library, SAS interrogates the system to determine whether the processor, as well as the I/O hardware associated with each volume allocated to the library data set, support the set of zHPF functionality required by the release of SAS. In the second maintenance release for SAS 9.4, SAS uses zHPF channel programs for Read operations involving direct access bound libraries. In the third maintenance release for SAS 9.4, SAS extends zHPF support to update and formatting operations as well.

The zHPF channel programs generated by the third maintenance release for SAS 9.4 to format library blocks also include a separate innovation besides the use of zHPF itself. The logic in SAS to generate CCW channel programs for Write operations attempts to write all the blocks that the engine needs to write at a given time. The number of blocks is closely related to the setting of the BUFNO option, and the set of blocks the engine needs to write might not correspond with track boundaries. To avoid the inefficiency of generating a channel program that formats less than a full track, the logic specific to CCW channel programs buffers the blocks destined for the last track if possible until data for all of the blocks on the track is available to be written. The third maintenance release for SAS 9.4 uses this optimization technique to a greater extent. The logic for generating zHPF channel programs avoids executing a formatting write until 10 tracks are ready to be formatted. This technique further accelerates formatting operations beyond the benefit provided by zHPF alone.

INCREASED I/O THROUGHPUT

To measure the effect of the zHPF channel programs, the authors developed a variety of tests that represent typical types of I/O processing. These tests performed little or no computational processing so that only the effect of I/O delays would be significant. We chose the library block size of 27K as the most representative of the processing done by SAS customers. The 27K block size requires the minimum number of inter-block gaps and thus provides the best track utilization (in bytes per track) possible for any block size that SAS supports for direct access bound libraries. Consequently, 27K is both the recommended and the default block size for these libraries.

To explore the effect of different page sizes, we ran the tests against two separate members, TALL and WIDE, the former having a smaller observation size. The sizes of these members are listed in Table 3.

Member	Page Size	Number of Pages	Size in Mbytes
tall	27648	240176	6333
wide	55296	147061	7755

Table 3 - Member Sizes

As detailed in Table 4, we compared the median (50th percentile) values for each set of runs for a particular test and channel program type. Overall, when using zHPF, SAS ran in 22% less elapsed time with zHPF channel programs than it ran with CCW channel programs. For individual types of tests, the improvements ranged from 17% to 32%. These tests compared the performance of the third maintenance release for SAS 9.4 (using zHPF) with SAS 9.4 (which can generate only CCW channel programs). The second maintenance release for SAS 9.4 (which uses zHPF for Read operations) demonstrated similar performance improvements for the Random Read and Sequential Read tests.

Type of Test	Member	Elapsed Seconds CCW	Elapsed Seconds zHPF	Ratio
Random Read	tall	32.7	22.2	0.68
Random Read	wide	39.3	29.1	0.74
Random Write	tall	129.5	89.7	0.69
Random Write	wide	124.2	89.0	0.72
Sequential Read	tall	57.7	48.0	0.83
Sequential Read	wide	35.8	25.4	0.71
DATA Step - Merge	tall	475.5	412.3	0.87
DATA Step - Merge	wide	132.0	86.3	0.65
PROC APPEND	tall	243.8	193.2	0.79
PROC APPEND	wide	115.4	85.7	0.74
DATA Step - Copy and Modify	tall	137.4	113.9	0.83
DATA Step - Copy and Modify	wide	84.7	53.9	0.64
Combined		**1608.0**	**1248.7**	**0.78**

Table 4 - Median Elapsed Time for CCW versus zHPF Channel Programs - Library BLKSIZE=27K

We also repeated a subset of these tests using a library block size of 6K, since that block size was recommended in the past for libraries that contain SAS catalogs (Squillace and Tharpe 1995). These results, detailed in Table 5, show that the ratio of zHPF elapsed time to CCW elapsed time is significantly higher for a library block size of 6K than it is for a block size of 27K. This result is not surprising considering the fact that a 6K library has more blocks per unit of data than a 27K library. Since zHPF improves performance by reducing the per-block overhead (Cronin, Entwistle, and Murphy 2011), the smaller block size provides more opportunity for benefit.

Type of Test	Member	Elapsed Seconds CCW	Elapsed Seconds zHPF	Ratio
Random Read	tall	41.82	25.5	0.61
Random Read	wide	48.06	32.41	0.67
Random Write	tall	145.97	103.01	0.71
Random Write	wide	171.1	102.04	0.60
Sequential Read	tall	78.93	48.97	0.62
Sequential Read	wide	61.79	30.65	0.50
Combined		**547.67**	**342.58**	**0.63**

Table 5 - Median Elapsed Time for CCW versus zHPF Channel Programs - Library BLKSIZE=6K

Finally, the authors examined the performance of a suite of weekly benchmark I/O tests to see the overall effect of adding zHPF write support in the third maintenance release for SAS 9.4. After eliminating all tests that did not involve any I/O to a bound library, we used the GLM procedure to test the hypothesis that there is a difference in elapsed time before and after the version of the third maintenance release for SAS 9.4 under development was modified to use zHPF for Write operations. There is a 96% probability that the overall performance of SAS was improved by this change.

DECREASED CHANNEL UTILIZATION

Not only do zHPF channel programs generated by SAS enable a higher rate of I/O, but they also use significantly less channel capacity, avoiding delays for other jobs and processes on the system when channel capacity is a limiting factor for the overall system I/O workload. IBM testing with the FICON Express8S Channel demonstrated that the admixture of even a small amount of CCW traffic (FICON) reduced the maximum amount of zHPF traffic that the channel could carry. See Figure 3 on page 7 in Cronin, Entwistle, and Murphy (2011) for details.

To measure this effect for the channel programs generated by SAS, the authors conducted two different types of tests to place a high load on the channel, measured the results via the IBM RMF (Resource Measurement Facility) product, and compared the results for SAS CCW channel programs with the results for SAS zHPF channel programs. To maximize the channel utilization, we reduced the number of channels online to the logical partition (LPAR) in which the test jobs were being run, leaving only two channels online to the DASD controller on which the test data sets resided. The test jobs were the only significant workload being run on the LPAR at the time of the tests. The information in the following tables was extracted from the following two RMF reports described in IBM (2013b):

- Channel Path Activity Report
- Device Activity Report

Two of the channel activity statistics require additional explanation:

- Channel Path Utilization Percentage – LPAR: This statistic indicates the percentage of time that the microprocessor for the channel was busy (IBM 2013b, p. 381). The zHPF channel programs require significantly less microprocessor utilization than equivalent CCW channel programs (Cronin, Entwistle, and Murphy 2011, p. 12). Our testing confirmed that this benefit also exists for the zHPF channel programs generated by SAS.
- Percentage of Bus Cycles Busy: This statistic indicates the ratio of the channel traffic produced by the LPAR on the channel link to the maximum rated capacity of the channel link, *in both directions* (Riedy 2015-03-13). For example, the channels used in the tests summarized in Table 6 and Table 7 were operating with a link speed of 8G bits per second. This corresponds to 800M bytes per second. Since the channel is capable of transmitting in both directions simultaneously, the divisor in this ratio

is 1600M bytes per second. Consequently, RMF indicates that a channel being used to the maximum capacity for read operations has 50% of its bus cycles busy.

Sequentially Reading a Single SAS Library with Eight Jobs Running in Parallel

The results listed in Table 6 were obtained from a test in which we simultaneously submitted eight SAS jobs in batch. Each job repeatedly read the same member of a SAS library without performing any processing on the data. The library had a block size of 27K, and the member had a page size of 27K as well. Since the SAS system option BUFNO had a value of 3, SAS read three pages in each I/O operation. This resulted in each channel program reading three library blocks. Note the following:

- The channel-specific statistics (shaded rows) contain the average of the values for the two channels that were online during the test.
- The device statistics contain the values for the volume containing the single test data set. The times are in milliseconds (ms).

In this test, the zHPF channel programs used less than one fourth as much channel processor capacity (Channel Path Utilization) as the CCW channel programs while sustaining a higher I/O rate. They also used about one half as much connect time, which was a major component of the entire response time.

RMF Statistic	zHPF	CCW
Channel Path Utilization Percentage – LPAR	16	67
Percentage of Bus Cycles Busy	44	28
Data Transfer Rate for Reads – LPAR (Mbytes/Second)	706	440
Number of FICON (CCW) Operations Per Second	--	5150
Avg. Number of FICON Operations Concurrently Active	--	4
Number of zHPF Operations Per Second	8735	--
Avg. Number of zHPF Operations Concurrently Active	5	--
Device Activity Rate (Start Subchannel Operations Per Second)	16773	10253
Avg. Response Time (ms)	0.437	0.704
Avg. Command Response (CMR) Delay	0.128	0.135
Avg. Pending Time (ms)	0.213	0.255
Avg. Disconnect Time (ms)	0.002	0.000
Avg. Connect Time (ms)	0.222	0.450
Pct. of Time Device Connected	51	61
Pct. of Time Device In Use	52	61
Avg. Number of Allocations	8	8

Table 6 - Channel and Device Utilization for Sequential Read (8-Way)

Randomly Reading Eight SAS Libraries with Eight Jobs Running in Parallel

In attempt to use the full capacity of the channels, we constructed a different test that submitted smaller I/O operations more frequently. As in the previous test, we submitted eight SAS batch jobs in parallel. Each job was set up to simulate the behavior of a SAS/SHARE server, asynchronously reading a narrow range of observations in parallel from eight identical libraries residing on eight separate volumes. This resulted in a total of 64 independent streams of data within the set of eight jobs. Each library had a block size of 27K, and each member had a page size of 27K. Since the access pattern was random, SAS read a single page per I/O operation, resulting in channel programs that each read a single block.

The zHPF channel program traffic generated by SAS sustained an aggregate data rate almost twice that of the CCW test while at the same time using half as much of the channel capacity. The device statistics (averages across eight volumes) show that the connect time for zHPF was 1/8 that of CCW, resulting in an overall response time that was twice as fast.

RMF Statistic	zHPF	CCW
Channel Path Utilization Percentage – LPAR	41	90
Percentage of Bus Cycles Busy	50	25
Data Transfer Rate for Reads – LPAR (Mbytes/Second)	797	401
Number of FICON (CCW) Operations Per Second	--	14517
Avg. Number of FICON Operations Concurrently Active	--	30
Number of zHPF Operations Per Second	28983	--
Avg. Number of zHPF Operations Concurrently Active	31	--
Device Activity Rate (Start Subchannel Operations Per Second)	7209	3626
Avg. Response Time (ms)	1.030	2.071
Avg. Command Response (CMR) Delay	0.793	0.729
Avg. Pending Time (ms)	0.895	0.951
Avg. Disconnect Time (ms)	0.001	0.000
Avg. Connect Time (ms)	0.135	1.121
Pct. of Time Device Connected	13	54
Pct. of Time Device In Use	13	54
Avg. Number of Allocations	8	8

Table 7 - Channel and Device Utilization for Random Read (64-Way)

TEST ENVIRONMENT

Obviously, the amount of performance improvement possible with zHPF depends on characteristics of the workload and hardware configuration, not to mention the metrics of interest. However, the preceding results show that the zHPF channel programs generated by SAS out-perform CCW channel programs generated by SAS to a degree that is consistent with the performance results published by IBM (Cronin, Entwistle, and Murphy 2011).

We invite you share your performance experience with us. To help you draw more useful comparisons, Table 8 describes the hardware of our test environment.

Feature	Specifications
Processor	IBM z196 Model 2817-706 – 6250 mips – 766 MSU 48GB allotted to test LPAR. z/OS 2.1 at maintenance level RSU1501 (recommended service upgrade)
Channel	FICON Express8S operating at 8 Gbps link speed.
DASD Controller	DS8800 IBM 2423-951 Fully loaded with Raid-5 600GB 10Krpm disk – 76TB physical; 58TB usable 64GB memory PAV and HyperPAV Arrays Across Loops Feature zHPF Running at the most recent (code) bundle -- 86.31.152.0

Table 8 - Hardware Specifications

CUSTOMER SITE IMPLEMENTATION

In order to experience the benefits of zHPF when accessing direct access bound libraries, you must implement the appropriate level of zHPF support in your System z hardware and software environment, as listed in detail in the following section. In addition, you must also upgrade to the second maintenance release for SAS 9.4 (which uses zHPF for Read access) or to the third maintenance release for SAS 9.4 (which uses zHPF for all access). However, even earlier releases of SAS can benefit in other ways from the presence of zHPF support on your system. SAS uses IBM DFSMSdfp standard access methods for reading and writing external files (that is, files whose format is determined by the operating environment rather than by SAS software). Since these IBM access methods can generate zHPF channel programs at certain release levels (Burgess 2014), implementing zHPF on your system can improve the performance of I/O to external files in all releases of SAS.

SYSTEM REQUIREMENTS

In order for zHPF channel programs to execute, several hardware requirements must be met. The processor, channel hardware, and DASD controller (both hardware and microcode) must all support zHPF, and the support must be enabled (IBM 2012b, Table 1-1, p. 10). In addition, the zHPF channel programs generated by SAS require certain minimum levels of hardware support because they exploit zHPF capabilities that are not available prior to certain microcode levels. Finally, the capability to submit a zHPF channel program via the EXCP service did not exist prior to z/OS V1R12. Table 9 combines these requirements into a single list (Riedy 2015-01-22).

Component	Requirements and Recommendations
z/OS	One of the following levels of z/OS: • V1R12 (with APAR OA38185) • V1R13 (with APAR OA38185) • V2R1 ZHPF=YES must be set in the IECIOSxx member in SYS1.PARMLIB (default for this parameter is 'NO'). z/OS must not be running as a guest under z/VM. We recommend that the fix be applied for APAR OA45589 - An Incorrect zHPF Channel Program Might Be Generated In Certain Retry Situations.
Processor	See topic "Which servers support zHPF?" in Burgess (2014).
DASD Controller	One of the following models in the IBM DS8000 series: • DS8700 with maintenance bundle 76.20.90.0 or higher • DS8800 with maintenance bundle 86.31.110.00 or higher • DS8870 As a general practice, we recommend that customers keep their DASD controller close to the current microcode levels, which can be found at the following link: http://www-01.ibm.com/support/docview.wss?uid=ssg1S1004456.

Table 9 - zHPF System Requirements

EVALUATING PERFORMANCE

Once your installation satisfies the requirements for SAS to generate zHPF channel programs, it is a worthwhile effort to evaluate the performance improvement.

First, verify that SAS has chosen to generate zHPF channel programs. In the third maintenance release for SAS 9.4, this can easily be determined by issuing a LIBNAME LIST statement for the library in question and examining the value of the statistic "Channel Program Type", which will be either "CCW" or "zHPF". In the second maintenance release for SAS 9.4, if zHPF support is *not* available, a message will be issued at library assign time stating that command mode (CCW) channel programs will be generated if the SAS system option DLMSGLEVEL is set to a value of WARN, as shown in Output 1.

```
NOTE: Command mode channel programs will be used to process library ...
1     options dlmsglevel=warn;
2     libname x '&temp' unit=vio;
NOTE: Libref X was successfully assigned as follows:
      Engine:        V9
      Physical Name: ...
```

Output 1 - Determining zHPF Support at 9.4m2

Note that zHPF cannot be used for VIO data sets, as illustrated in Output 1. In addition, the channel and device-specific requirements for zHPF must be satisfied for all volumes of a multi-volume library data set in order for SAS to use zHPF for processing the library. If SAS has chosen not to use zHPF on a system that seems to meet the requirements, set the option DLMSGLEVEL to DIAG, allocate the ddname SASSNAP to the appropriate destination, and send the resulting output to SAS Technical Support for diagnosis.

Second, determine the I/O throughput associated with the library. This can be achieved via the IBM RMF product, as shown in the preceding example. Note that the EXCP count for a ddname might not a useful measurement of the I/O for SAS direct access bound libraries because it reflects only the count of the number of channel programs executed, not necessarily the amount of data. SAS processes multiple blocks with each channel program, and the number of blocks per channel program might vary. (For IBM DFSMSdfp access methods, the EXCP statistic does reflect the number of blocks.)

To gain further insight into the amount of I/O performed for a direct access bound library, it is possible to request channel program generation statistics for the library by specifying the option DLDEBUG=1X in the LIBNAME statement by which the library is assigned. When the last libref to the library is cleared, SAS will write a set of statistics to SASLOG. An example of the statistics produced by the third maintenance release for SAS 9.4 is included in Output 2. Similar statistics are produced by the second maintenance release for SAS 9.4.

```
Channel Program Generation Statistics for Library ...:
 NumOperTotal              401550
 Method                      zHPF
 NumOperErrSchd                 0
 NumOperErrComp                 0
 NumOperRead               401550
 NumOperWrite                   0
 NumOperFlush                   0
 NumOperUnSort                  0
 NumSemS                   401547
 NumSemE                        0
 NumEXCPRead               401550
 NumEXCPUpdate                  0
 NumEXCPFormat                  0
 NumEXCPWrite                   0
 NumLRRead                 401550
 NumLRUpdate                    0
 NumLRFormat                    0
 NumBlkRead                401552
 NumBlkUpdate                   0
 NumBlkFormat                   0
 NumDefReads                    0
 NumDefUpdates                  0
 NumDefNull                     0
 MaxIOcount                     1
 MaxNumChanPgm                  1
 NumOverflowTCA                 0
 NumOverflowTIDAL               0
 BLKSIZE                0000006C00
 Flags                  00000004B0
```

Output 2 - Channel Program Generation Statistics for the Third Maintenance Release for SAS 9.4

The most important statistics for performance analysis are defined in Table 10.

Statistic	**Definition**
Method	Type of channel programs generated, either CCW or zHPF.
NumBlkRead	Number of library blocks read during the library assignment.
NumBlkUpdate	Number of existing library blocks updated during the library assignment.
NumBlkFormat	Number of new library blocks formatted during the library assignment.
BLKSIZE	Library block size in hexadecimal representation.

Table 10 - Definitions of Key Channel Program Generation Statistics

This information enables you to evaluate the performance based on the types of operations being performed. This is important because read, update, and formatting operations have different performance characteristics. However, observe the following precautions:

- Do not specify any value for DLDEBUG other than 1X unless advised by SAS Technical Support.
- The primary purpose of these statistics is for problem determination by SAS personnel. The statistics might be altered or removed in a future release.

Finally, remember that I/O throughput is only one aspect of the overall performance of a job on z/OS. Increasing the I/O rate can provide a benefit only for jobs that spend most of their time waiting for I/O to complete. Jobs that perform a large number of computations might be delayed for access to the CPU, and different tuning approaches will be required to accelerate those workloads. A complete discussion of optimization techniques can be found in Appendix 1, "Optimizing Performance" (SAS Institute Inc. 2014).

CONCLUSION

SAS is designed to provide high-performance access to direct access bound libraries. Some of the performance characteristics of this library implementation are unmatched by any other SAS library implementation. The introduction of support for the zHPF channel protocol has further improved the I/O throughput of this library implementation. This support also reduces competition for channel microprocessor resources, which might improve overall I/O throughput for the entire system environment.

REFERENCES

Bowman, Brian. 1989. "The Version 6 MVS SAS® Data Library Architecture." *Proceedings of the Fourteenth Annual SAS Users Group International Conference*, 696-703. San Francisco, California: SAS Institute Inc. Available http://www.sascommunity.org/sugi/SUGI89/Sugi-89-117%20Bowman.pdf.

Burgess, Jasmine. 2014. *System zHPF: Commonly Asked Questions.* Armonk, NY: International Business Machines Corporation. Available http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102468.

Cannatello, Carmine A. 1999. *Advanced Assembler Language and MVS Interfaces for IBM Systems and Application Programmers.* 2d ed. New York, NY: John Willey & Sons, Inc.

Cronin Cathy, Jon Entwistle, and Brian Murphy. 2011. *IBM zEnterprise 196 and IBM zEnterprise 114 I/O and FICON Express8S Channel Performance, Version 2.* Somers, New York: IBM Systems and Technology Group. Available http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&htmlfid=ZSW03196USEN#loaded.

Houtekamer, Gilbert E. and H. Pat Artis. 1993. *MVS I/O Subsystems: Configuration Management and Performance Analysis (3rd Printing, June 2006)*. Pagosa Springs, Colorado: Performance Associates, Inc.

IBM 2000. *IBM Enterprise Storage Server™ System/390 Command Reference 2105 Models E10, E20, F10, and F20*. Armonk, NY: International Business Machines Corporation.

IBM 2012a. *ABCs of z/OS System Programming Volume 10, Fifth Edition (March 2012)*. Armonk, NY: International Business Machines Corporation.

IBM 2012b. *FICON Planning and Implementation Guide, Fourth Edition (March 2012).* Armonk, NY: International Business Machines Corporation.

IBM 2012c. *z/Architecture Principles of Operation, Tenth Edition (September, 2012).* Armonk, NY: International Business Machines Corporation.

IBM 2013a. *DFSMS Macro Instructions for Data Sets, Version 2 Release 1.* Armonk, NY: International Business Machines Corporation.

IBM 2013b. *z/OS Resource Measurement Facility Report Analysis, Version 2 Release 1.* Armonk, NY: International Business Machines Corporation.

IBM 2013c. *z/OS V2R1.0 DFSMSdfp Advanced Services.* Armonk, NY: International Business Machines Corporation.

IBM 2014. *z/OS DFSMS Using Data Sets Version 2 Release 1.* Armonk, NY: International Business Machines Corporation.

IBM 2015a. I/O Connectivity: FICON/zHPF/CTC, Fibre Channel Protocol (FCP). Available http://www-03.ibm.com/systems/z/hardware/connectivity/ficon_performance.html. Accessed on March 25, 2015.

IBM 2015b. OA38185: NEW FUNCTION APAR - EXCP VIRTUAL SUPPORT FOR ZHPF. Available http://www-01.ibm.com/support/docview.wss?uid=isg1OA38185. Accessed on March 25, 2015.

Riedy, Dale (z/OS IOS Design and Development). "Re: zHPF Hardware & Software Requirements for SAS" Email to Lewis King (Lewis.King@sas.com). January 22, 2015.

Riedy, Dale (z/OS IOS Design and Development). "Re: Paper on zHPF for SAS Global Forum" Email to Lewis King (Lewis.King@sas.com). March 13, 2015.

SAS Institute Inc. 2014. *SAS® 9.4 Companion for z/OS, Third Edition*. Cary, NC: SAS Institute Inc.

Squillace, Daniel J. and Annette R. Tharpe. 1995. "Tuning the SAS® System Under MVS: A Release 6.08 Update." *Proceedings of the Twentieth Annual SAS Users Group International Conference*, 1353-1359. Orlando, Florida: SAS Institute Inc. Available at http://www.sascommunity.org/sugi/SUGI95/Sugi-95-225%20Squillace%20Tharpe.pdf.

ACKNOWLEDGMENTS

Dale Riedy of IBM z/OS IOS Design and Development provided invaluable advice for SAS R&D at many points during the process of designing the SAS zHPF channel program support and preparing this paper.

Dan Squillace was a strong proponent of zHPF support in SAS during the entire project, and his technical insight greatly aided the authors in measuring the performance improvements of SAS zHPF channel programs.

The authors would like to thank our SAS colleagues John Hamlet, Ron Burt, and Ed Webb for their expertise in z/OS and the stable test environment they helped maintain. We are also indebted to Lisa Cook, David Fahey, and Ron Burt for their helpful review comments and to Annette Tharpe for her role in validating SAS zHPF support and reviewing this paper.

RECOMMENDED READING

SAS® 9.4 Companion for z/OS, Third Edition (See "Using SAS Libraries" and "Optimizing Performance.")

Raithel, Michael A. 2003. *Tuning SAS® Applications in the OS/390 and z/OS Environments, Second Edition.* Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Lewis King
SAS Institute Inc.
919-531-3351
Lewis.King@sas.com

Fred Forst
SAS Institute Inc.
919-531-4793
Fred.Forst@sas.com