Paper 303

Get Hiper About Hiperspaces

Fred Forst - Eli Lilly & Co.

Abstract

SAS® offers the option of using hiperspaces as the standard WORK file under MVS/ESA® as a performance enhancement. This paper introduces hiperspaces under MVS, and the applicable SAS options to invoke hiperspace use. Also, attention is given to tuning the use of hiperspaces within MVS (i.e. MVS's SRM parameters). Performance gains can be astonishing with little or no work on the part of the user. Finally, specific case study results will be given.

Introduction

All SAS jobs (either batch or interactive) use a WORK file to store temporary data for the life of the job. The WORK file is normally placed on DASD, or occasionally VIO (MVS's virtual input/output). A SAS job could incur a great deal of time reading and writing data to this DASD workfile if the job manipulates a lot of temporary data. MVS tuning, in general, attempts to reduce or eliminate physical I/O to DASD since it usually constitutes the largest portion of a job's runtime. Hiperspaces(H/S) offers an alternative to the usual DASD WORK file with reduced runtimes for virtually all jobs, and substantially reduced runtimes for I/O intensive jobs.

MVS/ESA Hiperspaces

MVS's processor storage consists of Central Storage (CS), Expanded Storage (ES), and Auxiliary Storage (AUX). CS & ES are solid-state type memory and AUX is DASD. A job's

virtual storage is "backed" by processor storage (i.e. a job's data buffers as well as the programs the job executes are physically stored in processor storage-CS,ES, or AUX). A H/S exists in either ES or AUX, or both. Normal
MVS hierarchy of storage dictates that AUX is an overflow area for ES. When ES is strained, the data (in the case of a H/S) or program pages (in 4K segments) are stored in AUX.

By default, the H/S will first be written to ES. Then, when ES becomes overloaded by H/S's as well as non-H/S frames (e.g. Working Set frames from non-SAS jobs/users), the H/S pages then migrate to AUX to make room for the demand for ES space. This migration isn't limited to H/S's: any frame in ES could be a candidate for migration to AUX. However, H/S's are the first candidates to bypass ES and go directly to AUX. Therefore, even frames owned by on-line systems *could* migrate to AUX, which could degrade response time. It's possible (through the IEAIPSxx & IEAOPTxx members of SYS1.PARMLIB) to make the H/S pages bypass ES and go directly to AUX. One might do this for several reasons:

1. ES isn't installed on your system
2. You don't want to degrade on-line systems by forcing their ES frames to AUX when a SAS job places it's H/S in ES.
3. You want to avoid the thrashing of constantly moving pages from ES to AUX.

To accomplish this, code the H/S **criteria age** table in IEAOPTxx as follows (example shown is for domain #9):

(in IEAOPTxx)
ESCTBDS(9)=32767
(in IEAIPSxx)
DMN=9,ESCRTABX=9

This way, domain9's H/S frames will exist

only in AUX. Note that, however, if constraint of ES is not a problem, the above coding of IEAOPTxx is not needed. The default value for the H/S criteria age table is 1500 seconds, which means if the migration age is less than 1500 pages/sec, then a H/S page out should bypass ES and go directly to AUX. Figure 1 below shows this configuration.

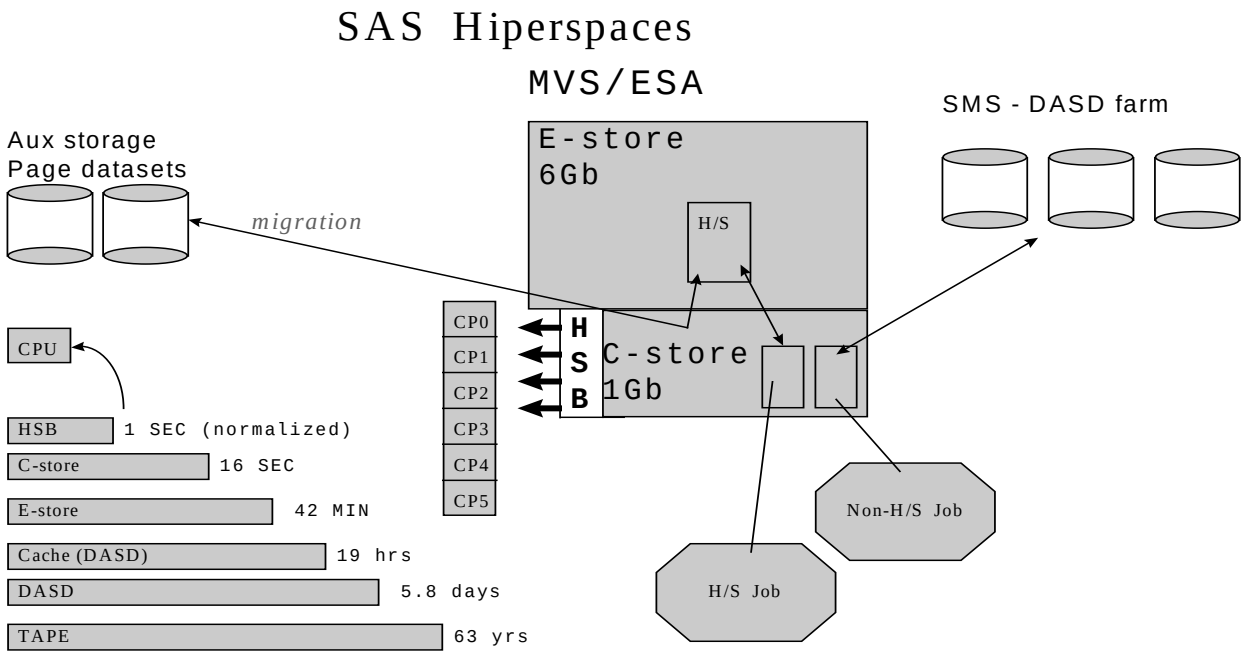

Figure 1

Basic H/S vs Non-H/S I/O

Figure 2 below shows the results of running two hourly jobs. Each job is identical except that one uses H/S's and the other does not. The jobs consist of only a DATA step. Notice the improved run times for the H/S job.

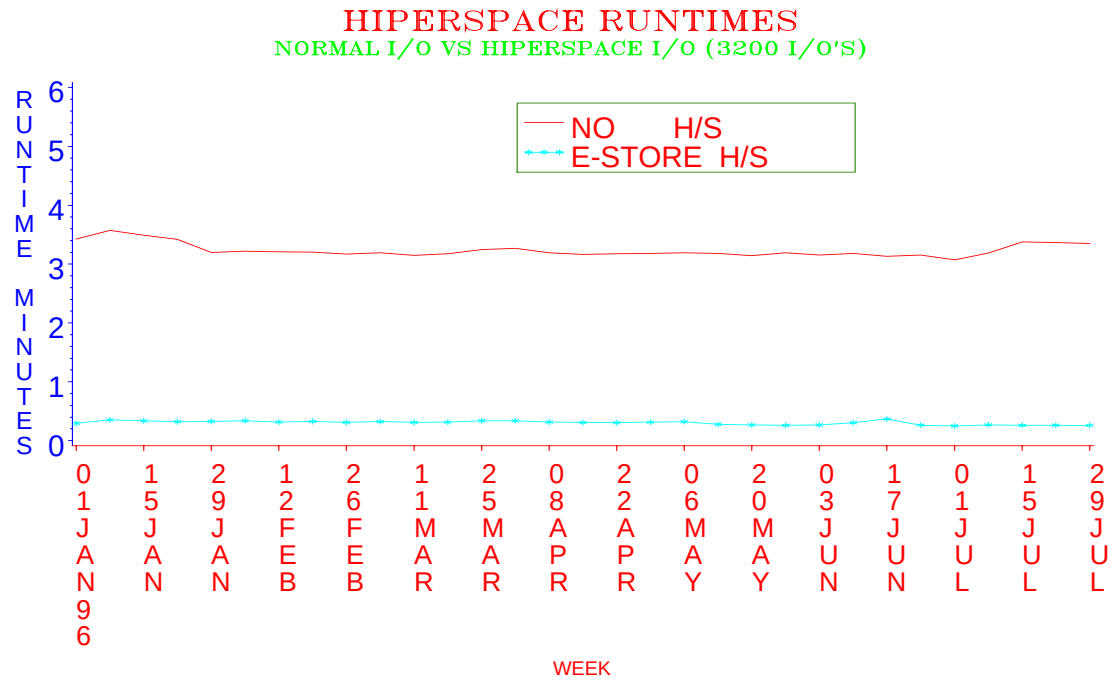

Figure 2

Implementation

Analyzing resource needs

Before converting all jobs to H/S, an analysis should be done to determine the amount of WORK I/O the SAS workload does as it uses standard DASD space for it's WORKfile.This can be accomplished by reading SMF30 records and examining the field SMF30BLK when it is associated with the ddname WORK. This value is the number of blocks read from the WORKfile. Also needed is the size of the WORKfile of the SAS jobs. This can be obtained from SMF14 records and examining the field SMF14NTA. This is the number of tracks allocated. But there is a small problem here: SMF14NTA will tell you the amount of tracks allocated for a specific dataset and job. You do not easily know the dataset name for the SAS WORKfiles (i.e. they will be named just like any other temporary dataset). You need to know the size of the WORK datasets only. This can be accomplished by coding DSN=&&SASWORK in the SAS PROC (or some other eye-catcher) that will identify the dataset as a SAS WORK dataset and then it can be easily identified in the SMF14 records. This will have to be done long before you implement H/S's in order to get a good historical look at the WORK space allocations. Figure 3 below shows the WORK space allocated, both before and after H/S implementation.

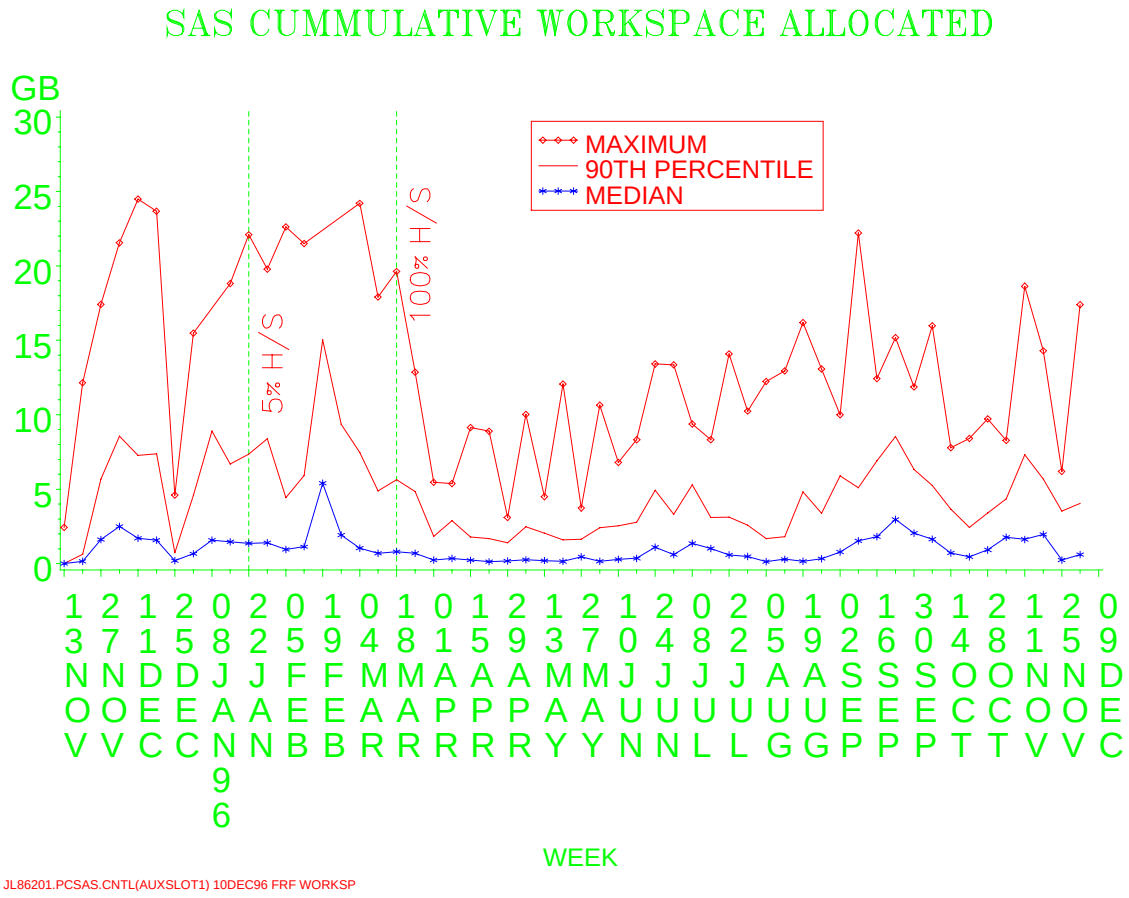

Figure 3

After you have the above information, you can find peaks to your resource needs and plan the H/S configuration accordingly. Notice the drastic reduction in WORK space allocations after H/S was 100% implemented. This is caused by three things:

1. People over-allocate onDASD
2. H/S's only allocate what the job needs.
3. When all jobs are using H/S's, they will run faster and therefore at any given time, there will be fewer running concurrently.

Measuring system overhead

When using H/S's for WORK files, the I/O to/from the H/S WORK file is done in one of three ways:

1. If the H/S page to be read/written exists in AUX, then MVS's ASM (auxiliary storage manager) does the I/O.
2. If the H/S page to be read/written exists in ES *and it's the first time the page has been referenced,* then MVS's RASP (real storage manager) moves the page between CS & ES.
3. If the H/S page to be read/written exists in ES *and it has been referenced before,* the address space itself will move the pages between CS & ES by using the MOVPG instruction.

It is important to realize the difference between #2 and #3 above. Moving pages with the MOVPG instruction is much more efficient because MVS doesn't need to get involved as it does when RASP is needed to move pages. Also, the effects of traffic between CS & ES are not considered "paging" as is the case when RASP moves the pages. If you are monitoring paging rates between CS & ES with RMF (or equivalent), the pages moved via MOVPG will not be reflected in the paging rates. Only the pages moved by RASP. Therefore, most paging reports will underestimate this traffic. Figure 4 below shows our paging rate (between CS & AUX).

As a check to ensure the paging rates were not adversely affecting other critical applications, (e.g. TSO response time), TSO response time is plotted in figure 4 above along with the page fault rate. Notice that, even when the page fault rate increases, there is *not* a degrade to TSO response time.

Also shown on the graph is when ES was upgraded from 1Gb to 4Gb which caused the page fault rate to decline, as expected. With more ES, fewer H/S pages are stored in AUX, hence less I/O. The horizontal line labeled "pain level" is the approximate page fault threshold level at which the jobs using H/S's start to degrade to a point they would have ran faster by not using H/S's. It can be seen this point is usually not reached. Remember that this is the rate of pages/second between CS & AUX.

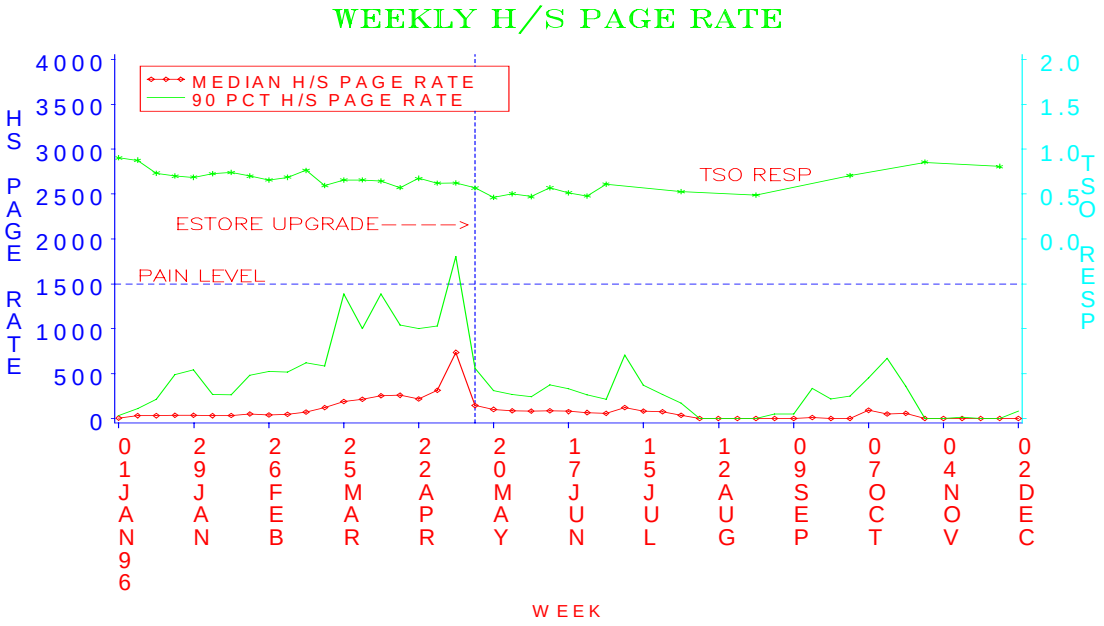

Figure 4

Also important to track is the capacity and allocation of the page datasets, which constitute AUX storage. Figure 5 below shows the AUX capacity, the warning level and the median and 90th percentile of slots in use. If the slots in use gets above the warning level, MVS prohibits certain things(LOGONS, jobs starting, etc).

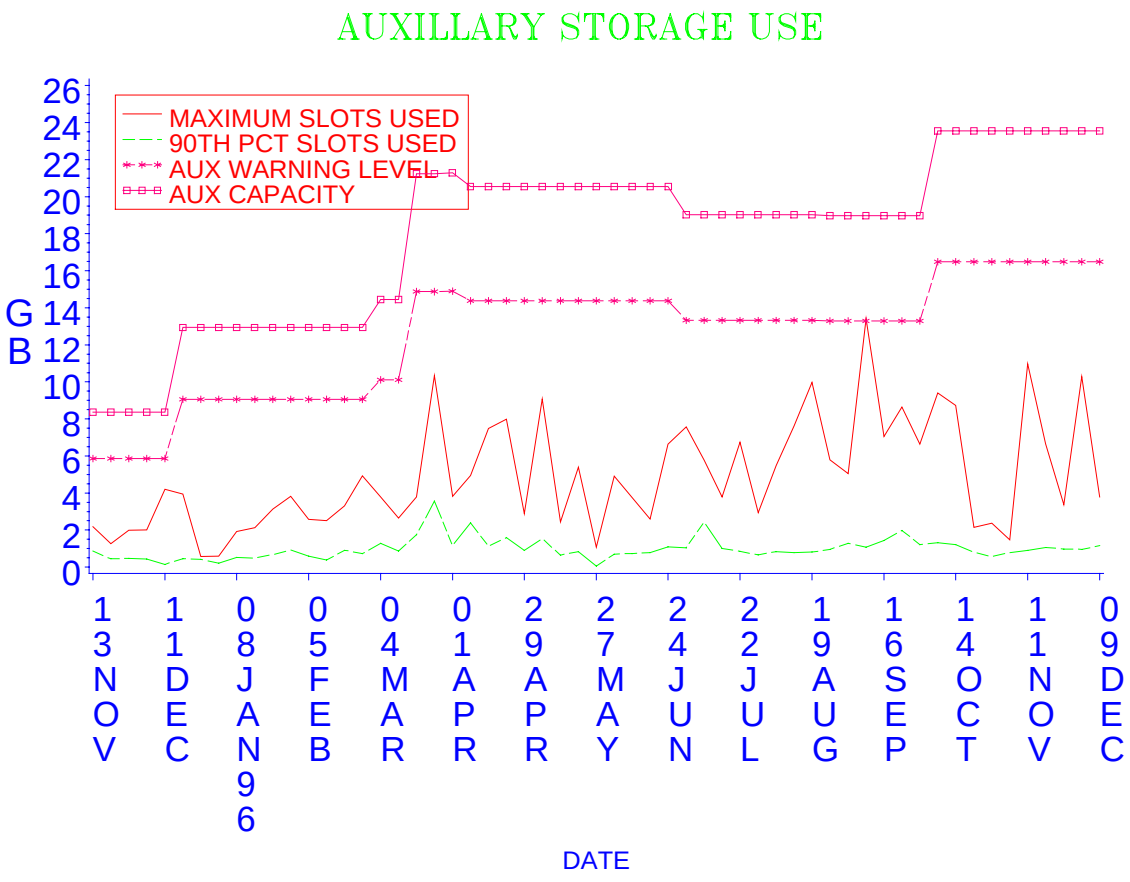

Figure 5

Results

Figure 6 below shows the effects of using H/S's on the median execution time of our SAS workload. Values given are the median weekly run time of all SAS jobs (left hand Y-axis) and weekly total SAS jobs executed (right hand Y-axis). Also shown are the dates the conversion project to H/S's started (November 1995) and ended (May 1996). The median runtime before H/S's were implemented was 29 seconds and after they were implemented, it was 7 seconds for an improvement of over 300%. Run times were reduced dramatically while at the same time, the number of weekly SAS jobs were increasing. Since these execution time shown are the *median*, one can feel confident that more I/O bound jobs would show more improvement. Jobs improving by a factor of 6-8 are not uncommon.

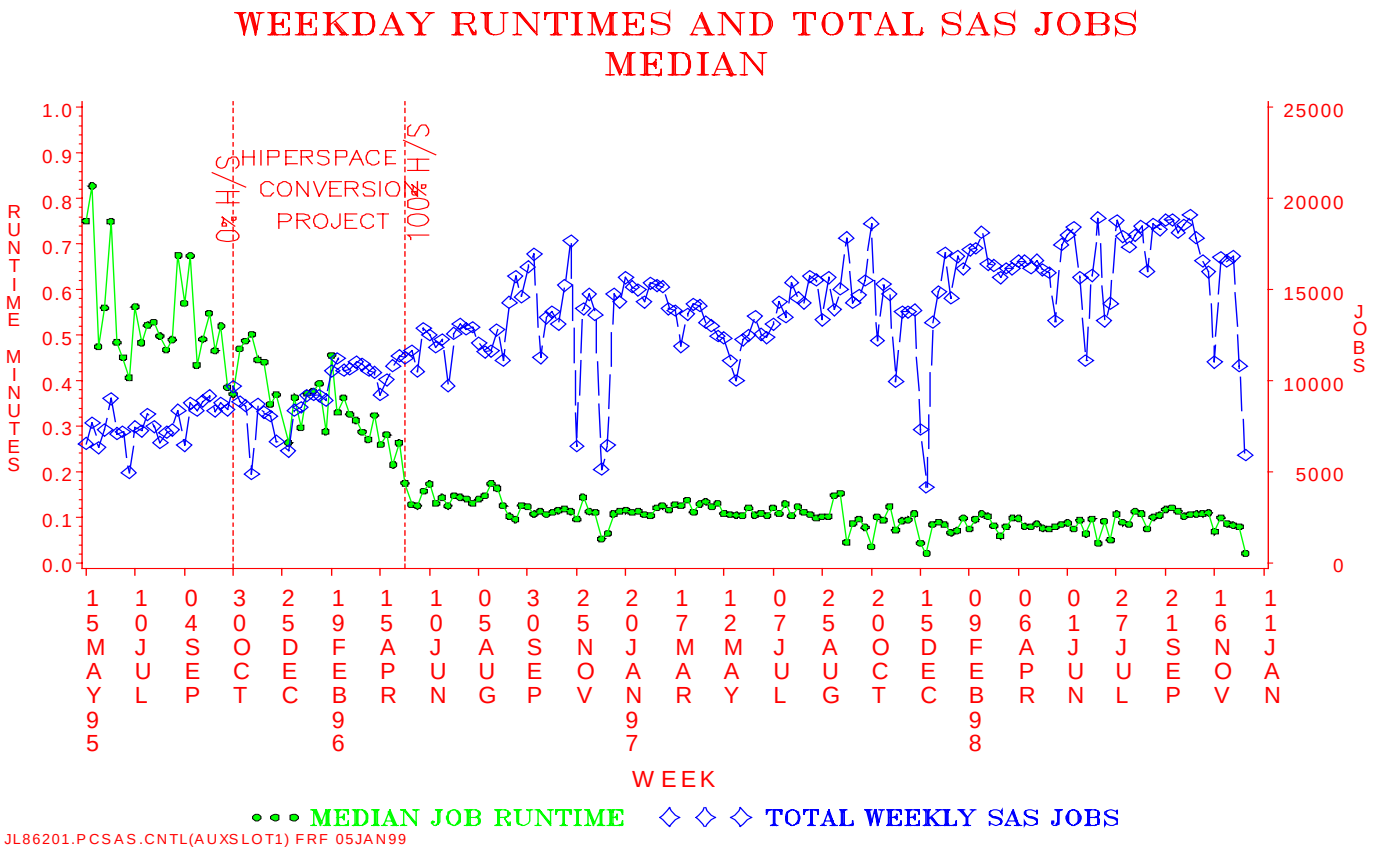

Figure 6

Conclusion

Hiperspace WORK files have been proven to be a performance improvement for SAS jobs. In this case study, median runtimes improved by a factor of 4, while suffering no noticeable degradation in other applications. Those jobs which are more I/O bound than normal will run as much as 6-8 times as fast.

References

IBM (1993), *Initialization and Tuning Reference,Fifth Edition,*IBM Corp, Poughkeepsie, NY

IBM (1993), *Initialization and Tuning Guide,Sixth Edition,*IBM Corp, Poughkeepsie, NY

Fred R. Forst
FORST_FRED_R@LILLY.COM
(317) 276-0842