

Using a RAM Disk to Speed up your Windows 2000 SAS Session

Erik H. Dilts, Rho Inc.

Abstract

Old ideas that fall out of favor due to the advent of new technologies rarely make a comeback. However, it is worth examining “the old ways” every so often to see if any benefits can be gleaned from their usage or methodology. The RAM disk is an example of this: a computer optimization method that was considered necessary as recently as SAS version 5 but has since fallen out of general use. Why? For many years, computer memory has been prohibitively expensive in relation to the benefits received by using a RAM disk on a standard workstation, but that is no longer the case. Now one can set up a large (512 MB) RAM drive for roughly \$125. This paper will demonstrate that for many (but not all) SAS users, the speed gains realized are easily worth the cost. It is written toward any skill level and is applicable to any version of SAS run on Microsoft Windows 2000.

Introduction

The concept of a RAM disk is simple: set aside an area of the computer’s memory (RAM) to serve as a simulated hard disk. The reason for use is also straightforward. While hard drive speeds have increased tremendously since RAM disks fell out of use, they are still nowhere near as fast as RAM, either in access time or read-write speed. For a quick comparison, a three year-old disk drive has an average input/output (I/O) rate of around 2700 Kb/second. The newest generation workstation disk drives have increased this to near 30000 Kb/second. A RAM drive set up will have an I/O rate in excess of 100000 Kb/second. The access time differences are even more impressive. Typical IDE hard disk access times range from 11.1 to 17.5 milliseconds, whereas memory access times are 6-10 nanoseconds – greater than a million-fold increase! For procedures that read from and write to the disk thousands of times, these speed improvements add up very quickly to provide a very substantial decrease in total running time.

RAM Disk Setup

There are several choices involved in setting up a RAM disk. The decision is based on how much memory one has to spare, the size of the data sets being processed, and coding style.

Regardless of other setup considerations, overall, setting up the SAS WORK directory on a RAM disk generates the best performance gains. The first level of usage is limited by its maximum size, but has the benefit of being free. With the right data and judicious programming, even a small RAM disk can be very effective. Under these circumstances, the Microsoft “Ramdisk.sys Sample Driver for Windows 2000”¹ can be used to set up a 15 megabyte (MB) RAM disk that performs well enough for small jobs. With a RAM disk this small, a serious limitation to note is that no one-level (equivalent to WORK.aaa) data set names may be used. With such a small disk, the WORK directory will quickly be filled with the intermediate files that SAS creates during most PROCs and DATA steps, so if whole data sets are written to it in addition to these files the RAM disk will run out of space and the program will abend.

Most users will find the Microsoft Windows 2000 RAM disk too small to be helpful in production use, but it does serve as a helpful test case to see if a given set of data manipulations will benefit. A 3rd-party program is required to create RAM disks that are larger and more useful. Several programs are available ranging from \$35 to \$129 - incidentally, the program found to be the most useful during testing for this paper was also the cheapest. All the RAM disk programs tested while researching this paper had very similar feature sets, so cost and stability become the deciding factors. Using a 3rd-party solution will likely make the probable limit on RAM disk size be the amount of memory your computer’s motherboard will allow, rather than the 15 MB maximum of the sample driver. Most motherboards designed in the last 3-4 years will hold at least 512 MB of RAM, and many allow up to 3072 MB (3 GB). Windows 2000 needs 128 MB of its own on most systems, so an amount greater than that is needed to set up a RAM disk. Over the past few years, many business workstations have come equipped with 256 MB of RAM, which

¹ <http://support.microsoft.com/directory/article.asp?ID=KB;EN-US;Q257405>

will allow a 128 MB RAM disk to create without adding extra memory. This size will serve the needs of many SAS users. However, in order to fully harness the potential benefits of having the SAS WORK directory on a RAM disk, 512 MB is the minimum recommended size. Many Windows 2000 workstation SAS users work with SAS files small enough that a 512 MB disk will rarely fill up over the course of any one program run, which gives the immediate benefit of being able to run previously written code with no modifications. A smaller RAM disk may require the addition of a TEMP library in the code to replace one-level data set names and forced clearing of the WORK directory to existing programs, as shown in Figure 1.

Figure 1:

<pre>/* Original program */ data foo; set test1 test2; proc sort data=foo; by var_name; run;</pre>	<pre>/ Modified program */ libname TEMP "C:\TEMP" ACCESS=TEMP; data temp.foo; set temp.test1 temp.test2; proc sort data=temp.foo; by var_name; run;</pre>
--	---

Speed gains are realized in this scenario as SAS writes its intermediate records and data sets to and from the RAM disk. Even more performance improvement is realized when the RAM disk is large enough to fit the temporary data sets created in the original version of the program.

SAS Setup

The greatest performance gains from using a RAM disk with SAS for Windows are seen when the WORK directory is set up on the RAM disk. SAS stores macros in the work library in addition to intermediate data files and one-level name data sets, so even macro calls increase in speed when this setup is used. The CONFIG file must be changed to alter the location of the WORK directory – it cannot be done once SAS has started. The default setup (CONFIG.SAS in V6 or xxxxCFG.SAS in V8) has the line `–WORK C:\SASWORK` to set the location of the WORK directory on the local hard disk, so if the location of the RAM disk has been designated as Z:, this line would be changed to `–WORK Z:\` to set the location of the WORK library on the RAM disk. Another option is to create a TEMP library on the RAM disk. This option takes away the advantage of rapid access and I/O of the intermediate files, but it adds the ability to control precisely how much space on the disk is used, which can be helpful with RAM disk setups less than 256 MB in size. Referring back to the modified program in Figure 1, one would set the library TEMP to “Z:\” rather than “C:\TEMP” which allows the programmer to only put the data sets that (a) fit on the disk, and (b) benefit the most from the rapid I/O, into the RAM disk library.

Usage

If a large enough RAM disk is available, once the disk is set up and SAS is configured to for it, usage will be transparent. No programming changes are needed and the only noticeable difference from a standard setup will be the performance increase. This setup has been tested with a 512 MB RAM disk using data sets up to 61 MB in size with no disk space problems occurring. Using the sample program (shown below in figure 2) in this setup yields over a 250% performance increase over the default setup. Using a 256 MB RAM disk will work in very many situations, and frequent use of the `PROC DATASETS library=WORK kill;` statement to clear the contents of the WORK library often prevent “out of disk space” errors. This setup shows an average 81% speed improvement over the default. If the RAM disk is less than 128 MB in size, only either the WORK library or the created TEMP library may be placed on the RAM disk but not both. Testing showed that programs heavy in sorting benefit the most from a RAM disk work directory, and DATA step I/O-heavy programs to benefit near equally from either arrangement. Even this setup can result in 40% gains, although more program adjustment is necessary than with the previous two arrangements.

Conclusion

RAM disks were considered nearly essential for time-efficient database processing as recently as the 1980’s but technology outgrew them. Now, due to the extreme drop in the price of large memory modules, they have regained their potential as a very easy and inexpensive way to more than halve the program run time for many disk-intensive

procedures. For the test setup used for researching this paper, a RAM disk program costing \$35 and a 512 MB PC133 memory module costing \$89 were purchased for a total cost of \$124. The 250% potential performance gain offered from this arrangement is far greater than that offered by buying a whole new computer setup with twice the raw processing power of one's current system. Installation of both the memory module and the program can be done in less than one hour.

Author Contact

Erik H. Dilts

Rho, Inc.

100 Eastowne Drive

Chapel Hill, NC 27514

Dilts@rhoworld.com