Multithreading Applications in Version 9.1

Prepared by

destiny CORPORATION

International SAS® Training and Consulting

Destiny Corporation
100 Great Meadow Rd Suite 601
Wethersfield, CT 06109-2379
Phone: (860) 721-1684 1-800-7TRAINING Fax: (860) 721-9784
Email: info@destinycorp.com
Web: www.destinycorp.com

Multi-Threaded Architecture

One of the most helpful enhancements with SAS software in Version 9 is its ability to support multi-threaded access to files in the Data Step and certain procedures. The concept is simple: instead of using the traditional 'serial' approach to either sorting or summarizing data, SAS breaks up the data into chunks, performs the operation on each piece with a different processor and then puts the result back together.

Sorting Analogy

Consider the following analogy for sorting. There are four decks of playing cards. The goal is to order them. One person can combine all four decks together to produce an ordered result. Another way is to have four people each order one deck at the same time. The ordered four decks are then combined for a final result.

Summarizing Analogy

The goal is to sum a total of 100 numbers. One person can total the 100 numbers. Another way is to have four people each take 25 numbers and produce totals. The four totals are then added to produce the final result.

The division of labor or tasks in these examples demonstrates why a multi-threaded architecture provides faster results when available.

Multi-threading is supported for:

1. PROC SORT
2. PROC SUMMARY
3. PROC MEANS
4. PROC REPORT
5. PROC TABULATE
6. PROC SQL
7. PROC REG
8. PROC GLM
9. PROC ROBUSTREG

By default, multi-threading is turned on in Version 9 for all of these procedures. Therefore, there is a new option available with each procedure (THREADS/NOTHREADS) to turn this feature off.

CPUCOUNT Option

Multi-threading is much more effective in a multiple processor environment. For example, if SAS is running on a four processor server, it will attempt to utilize all of the CPUs available (up to four). For large SAS jobs, this may impact performance of other applications or programs running on that server.

The default value of CPUCOUNT is 0. This indicates that SAS use as many processors as it can access. In some situations, it may be wise to limit the number of CPUs accessed by SAS by setting this value to a maximum CPU number. CPUCOUNT is the way to throttle back the number of processors used in multiple processor environments.

Benchmarking

In threading environments, it is possible that the CPU time may be larger than real time/wall time. This may be considered when scheduling large production jobs.

Scalable Performance Data Architecture

Since Version 6 of SAS, the Scalable Performance Data Server has been available as a separate product. It allows for breaking apart data storage to speed up I/O and avoids the traditional 'serial based' I/O architecture. SPDS Software also supported very secure access to this data. The ideal storage of a data source is split across several locations that are usually separate disks with separate disk controllers.

In Version 9 this entire architecture, except for the security model, is included in Base SAS software.

Scalable Performance Data Engine SPDE

This is a new engine that is part of Base SAS. It is specified on a libname statement for much quicker access to data stored on disk. The ideal environment is the same as listed above with the addition of a metadata repository and index.

To summarize:

- Each data location is stored on a separate disk
- Each disk has a separate disk controller
- Each metadata repository is stored in a separate location
- Each index is stored in a separate location

Scalable Analogy

Consider a two lane highway with two toll booths. Traffic can proceed through those toll booths at a particular average speed.

Now consider a two lane highway with 10 toll booths. Cars move to other lanes to pay the tolls and then merge back together into a two lane highway. Would this be better than the prior analogy?

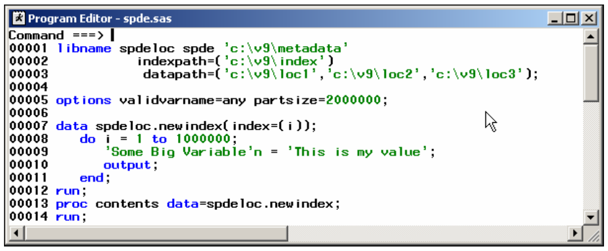
Suppose a 10 lane highway has 10 toll booths.

Now consider a 10 lane highway with no toll booths.

Which one would be fastest?

SPDE performs best when all functionality is separated into specific tasks with no interrupting processes to reduce speed. A shared disk controller would impede application speed. Performance is best on a 10 lane highway with no toll booths.

Consider the following example which demonstrates the SPDE syntax.



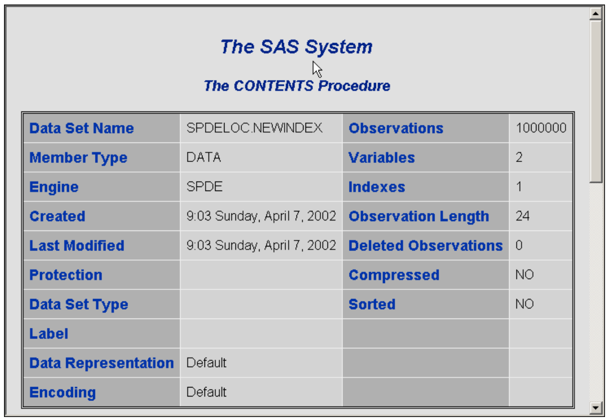
```
Program Editor - spde.sas
Command ===>
libname spdeloc spde 'c:\v9\metadata'
              indexpath=('c:\v9\index')
               datapath=('c:\v9\loc1','c:\v9\loc2','c:\v9\loc3');

options validvarname=any partsize=2000000;

data spdeloc.newindex(index=(i));
   do i = 1 to 1000000;
      'Some Big Variable'n = 'This is my value';
      output;
   end;
run;
proc contents data=spdeloc.newindex;
run;
```

This form of the libname statement uses the SPDE engine.

There is a separate location for:

1. Metadata
2. Datapath1
3. Datapath2
4. Datapath3
5. Indexes

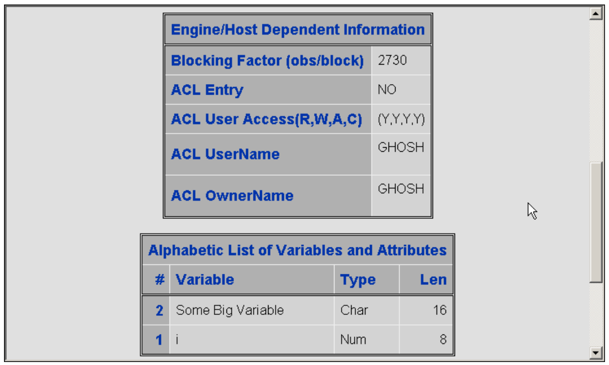
This example is for syntax only. In an ideal situation, the locations would all reside on separate disks with different disk controllers. Note that datapath 3 is defined but not used in this case.



The SAS System

The CONTENTS Procedure

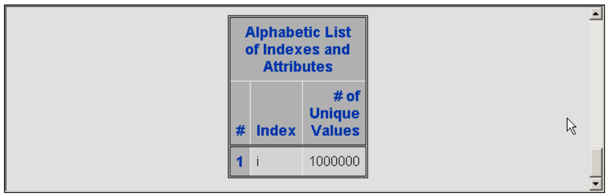
Data Set Name	SPDELOC.NEWINDEX	**Observations**	1000000
Member Type	DATA	**Variables**	2
Engine	SPDE	**Indexes**	1
Created	9:03 Sunday, April 7, 2002	**Observation Length**	24
Last Modified	9:03 Sunday, April 7, 2002	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	Default		
Encoding	Default		



Engine/Host Dependent Information	
Blocking Factor (obs/block)	2730
ACL Entry	NO
ACL User Access(R,W,A,C)	(Y,Y,Y,Y)
ACL UserName	GHOSH
ACL OwnerName	GHOSH

Alphabetic List of Variables and Attributes

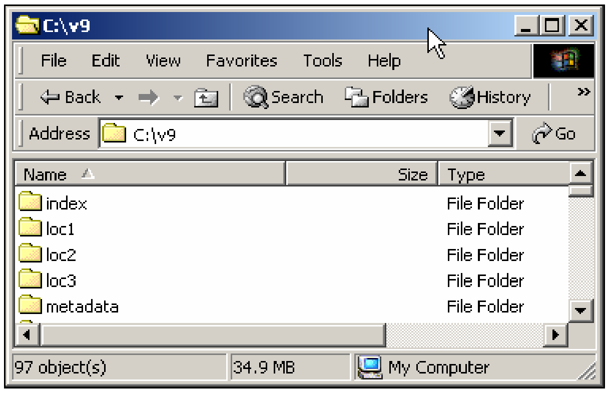
#	Variable	Type	Len
2	Some Big Variable	Char	16
1	i	Num	8



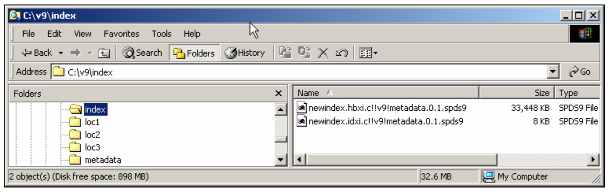
Alphabetic List of Indexes and Attributes

#	Index	# of Unique Values
1	i	1000000

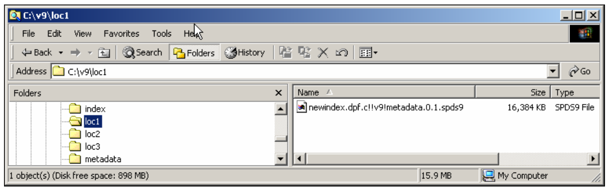
This is a list of the resulting files in these directories.



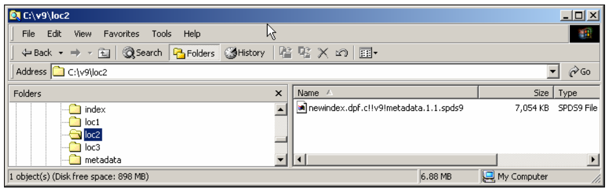
```
C:\v9
Address  C:\v9
Name          Size  Type
index               File Folder
loc1                File Folder
loc2                File Folder
loc3                File Folder
metadata            File Folder
97 object(s)   34.9 MB   My Computer
```



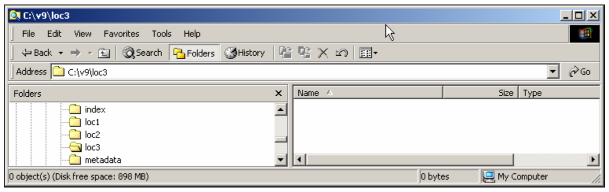
```
C:\v9\index
Address  C:\v9\index
Name                                        Size       Type
newindex.hbxi.c!!v9!metadata.0.1.spds9      33,448 KB  SPDS9 File
newindex.idxi.c!!v9!metadata.0.1.spds9           8 KB  SPDS9 File
2 object(s) (Disk free space: 898 MB)   32.6 MB   My Computer
```



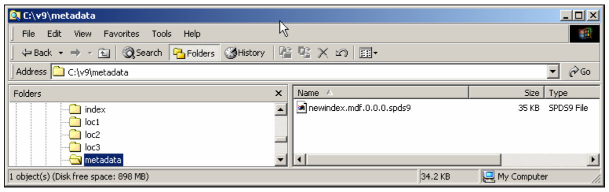
```
C:\v9\loc1
Address  C:\v9\loc1
Name                                        Size       Type
newindex.dpf.c!!v9!metadata.0.1.spds9       16,384 KB  SPDS9 File
1 object(s) (Disk free space: 898 MB)   15.9 MB   My Computer
```



```
C:\v9\loc2
Address  C:\v9\loc2
Name                                        Size       Type
newindex.dpf.c!!v9!metadata.1.1.spds9        7,054 KB  SPDS9 File
1 object(s) (Disk free space: 898 MB)   6.88 MB   My Computer
```



```
C:\v9\loc3
Address  C:\v9\loc3
Name                                        Size       Type
0 object(s) (Disk free space: 898 MB)   0 bytes   My Computer
```



```
C:\v9\metadata
Address  C:\v9\metadata
Name                                        Size       Type
newindex.mdf.0.0.0.spds9                       35 KB  SPDS9 File
1 object(s) (Disk free space: 898 MB)   34.2 KB   My Computer
```

In this example, there was not enough data to utilize loc3.

Conclusion

Large amounts of data can be processed effectively with this architecture. The location of data is now split and must be maintained. This is an important consideration when moving data. However, the performance efficiency gained may be worth the additional effort.

For additional information on SPDE, go to the following link:

http://www.destinycorp.com/documentation/spde/index.html