CHAPTER

1

Overview: The SPD Engine

Introduction to the SPD Engine

The SPD Engine is designed for high-performance data delivery. It enables rapid access to SAS data for processing by the application. The SPD Engine delivers data to applications rapidly because it organizes the data into a streamlined file format that takes advantage of multiple CPUs to perform parallel input/output functions.

The SPD Engine uses *threads* to read blocks of data very rapidly and in parallel. The software tasks are performed in conjunction with an operating system that enables threads to execute on any of the computer's available CPUs. Although threaded I/O is an important part of the SPD Engine functionality, the real power of the SPD Engine comes from the way that the software structures SAS data. The SPD Engine organizes data into a new file format that includes partitioning of the data. This data structure permits threads, running in parallel, to perform I/O tasks efficiently.

Although it is not intended to replace the default default Base SAS engine, the SPD Engine is a high-speed alternative for processing very large data sets. It reads and writes data sets that contain millions of observations. For example, this includes data sets that expand beyond the 2-gigabyte size limit imposed by some operating systems and data sets that SAS analytic software and procedures must process faster.

The SPD Engine performance is boosted in the following ways:

□ support for gigabytes of data

□ scalability on symmetric multiprocessor (SMP) computers

□ parallel WHERE selections

□ parallel loads

□ parallel index creation

□ parallel I/O data delivery to applications

□ automatic sorting on BY statements

The SPD Engine runs on UNIX, Windows, z/OS (zFS file system only), and OpenVMS on HP Integrity Servers (ODS-5 file systems only).

Note: Be sure to visit the Scalability and Performance focus area at **`http://support.sas.com/rnd/scalability`** for more information about scalability in SAS 9.2. For system requirements, visit the Install Center at **`http://support.sas.com/documentation/installcenter`**. △

Using the SMP Computer

The SPD Engine exploits a hardware and software architecture known as symmetric multiprocessing. An SMP computer has multiple central processing units (CPUs) and an operating system that supports threads. An SMP computer is usually configured with multiple controllers and multiple disk drives per controller. When the SPD Engine reads a data file, it launches one or more threads for each CPU; these threads then read data in parallel from multiple disk drives, driven by one or more controllers per CPU. The SPD Engine running on an SMP computer provides the capability to read and deliver much more data to an application in a given elapsed time.

Reading a data set with an SMP computer that has 5 CPUs and 10 disk drives could be as much as 5 times faster than I/O on a single-CPU computer. In addition to threaded I/O, an SMP computer enables threading of application processes (for example, threaded sorting in the SORT procedure in SAS 9.1 or later).

The exact number of CPUs on an SMP computer varies by manufacturer and model. The operating system of the computer is also specialized; it must be capable of scheduling code segments so that they execute in parallel. If the operating system kernel is threaded, performance is further enhanced because it prevents contention between the executing threads.

As threads run on the SMP computer, managed by a threaded operating system, the available CPUs work together. The synergy between the CPUs and threads enables the software to scale processing performance. The scalability, in turn, significantly increases overall processing speed for tasks such as creating data sets, appending data, and querying the data by using WHERE statements.

Organizing SAS Data Using the SPD Engine

How the SPD Engine Organizes SAS Data

Because the SPD Engine organizes data for high-performance processing, an SPD Engine data set is physically different from a default Base SAS engine data set. The default Base SAS engine stores data in a single data file that contains both data and data descriptors for the file (metadata). The SPD Engine creates separate files for the data and data descriptors. In addition, if the data set is indexed, two index files are created for each index. Each of these four types of files is called an SPD Engine *component file* and each has an identifying file extension.

In addition, each of these components can consist of one or more physical files so that the component can span volumes but can be referenced as one logical file. For example, the SPD Engine can create many physical files containing data, but reference the files containing data as a single data component in an SPD Engine data set. The *metadata* and index components differ from the data component in two ways:

1. You can specify a fixed-length partition size for data component files using the PARTSIZE= option. However, you have little or no control over the size of the metadata or index partitions.
2. The data component files are created in a cyclical fashion across all defined paths. The metadata and index components are created in a single path until that path is full, and then the next path is used.

Metadata Component Files

The SPD Engine data set stores the descriptive metadata in a file with the file extension .mdf. Usually an SPD Engine data set has only one .mdf file.

Index Component Files

If the file is indexed, the SPD Engine creates two index component files for each index. Each of these files contains a particular view of the index, so both exist for each data set.

- The index file with the .hbx file extension contains the global index.
- The index file with the .idx file extension contains the segment index.

Data Component Files

The data component of an SPD Engine data set can be several files (partitions) per path or device, rather than just one. Each of these partitions is a fixed length, specified by you when you create the SPD Engine data set.

Specifying a partition size for the data component files enables you to tune the performance of your applications. The partitions are the threadable units, that is, each partition (file) is read in one thread. Chapter 2, “Creating and Loading SPD Engine Files,” on page 11 provides details on how the SPD Engine stores data, metadata, and indexes.

Comparing the Default Base SAS Engine and the SPD Engine

Overview of Comparisons

Default Base SAS engine data sets and SPD Engine data sets have many similarities. They both store data in a SAS library, which is a collection of files that reside in one or more directories. However, because the SPD Engine data libraries can span devices and file systems, the SPD Engine is ideal for use with very large data sets. Also, the SPD Engine enables you to specify separate directories, or devices, for each component in the LIBNAME statement. Chapter 2, “Creating and Loading SPD Engine Files,” on page 11 provides details on designing and setting up the SPD Engine data libraries.

The SPD Engine Libraries and File Systems

An SPD Engine library can contain data files, metadata files, and index files. The SPD Engine does not support catalogs, SAS views, MDDBs, or other utility (byte) files.

The SPD Engine uses the zFS file system for z/OS and the ODS-5 file system for OpenVMS on HP Integrity Servers. This means that some functionality might be slightly different on these platforms. For example, for z/OS, the user must have a home directory on zFS.

Utility File Workspace

Utility files are generated during the SPD Engine operations that need extra space (for example, when creating parallel indexes or when sorting very large files). Default locations exist for all platforms but, if you have large amounts of data to process, the default location might not be large enough. The SPD Engine system option SPDEUTILLOC= lets you specify a set of file locations in which to store utility scratch files. See “SPDEUTILLOC= System Option” on page 83 for details.

Temporary Storage of Interim Data Sets

To create a library to store interim data sets, specify the SPD Engine option TEMP= in the LIBNAME statement. If you want current applications to refer to these interim files using one-level names, specify the library on the USER= system option.

The following example code creates a user libref for interim data sets. It is deleted at the end of the session.

```
libname user spde '/mydata' temp=yes;
data a; x=1;
run;
proc print data=a;
```

The USER= option can be set in the configuration file so that applications that reference interim data sets with one-level names can run in the SPD Engine.

Differences between the Default Base SAS Engine Data Sets and the SPD Engine Data Sets

The following chart compares the SPD Engine capabilities to default Base SAS engine capabilities.

Table 1.1 Comparing the Default Base SAS engine Data Sets and the SPD Engine Data Sets

Feature	SPD Engine	Default Base SAS Engine
Partitioned data sets	yes	no
Parallel WHERE optimization	yes	no
Lowest locking level	member	record
Concurrent access from multiple SAS sessions on a given data set	READ (INPUT open mode)	READ and WRITE (all open modes)
Remote computing via SAS/CONNECT	no	yes
Data transfer via SAS/CONNECT	no	yes
RLS (Remote Library Services) via SAS/CONNECT	no	yes
Available via SAS/CONNECT	no	yes
Support in SAS/SHARE	no	yes
Automatic sort for SAS BY processing (sort a temporary copy of the data to support BY processing)	yes	no
User-defined formats and informats	yes, except in WHERE[1]	yes
Catalogs	no	yes
Views	no	yes
MDDBs	no	yes
Integrity constraints	no	yes
Data set generations	no	yes
CEDA	no	yes
Audit trail	no	yes
NLS transcoding	no	yes
Number of observations that can be counted	2^{63}-1 (on all hosts)	2^{31}-1 (on 32-bit hosts) 2^{63}-1 (on 64-bit hosts)
COMPRESS=	YES|NO|CHAR|BINARY (only if the file is not encrypted)	YES|NO|CHAR|BINARY
ENCRYPT=	cannot be used with COMPRESS=	can be used with COMPRESS=
Encryption	data files only	yes (all files)
FIRSTOBS= system option and data set option	no	yes
OBS= system option and data set option	yes, if used without ENDOBS= or STARTOBS= SPD Engine options	yes

Feature	SPD Engine	Default Base SAS Engine
Functions and call routines	yes, with some exceptions[2]	yes
Move table via OS utilities to a different directory or folder	no	yes
Observations returned in physical order	no, if BY or WHERE is present	yes

[1] In WHERE processing, user-defined formats and informats are passed to the supervisor for handling; therefore, they are not processed in parallel.

[2] In WHERE processing, functions and call routines introduced in SAS 9 or later are passed to the supervisor for handling; therefore, they are not processed in parallel.

Interoperability of the Default Base SAS Engine and the SPD Engine Data Sets

Default Base SAS engine data sets must be converted to the SPD Engine format so that the SPD Engine to access them. You can convert the default Base SAS engine data sets easily using the COPY procedure, the APPEND procedure, or a DATA step. (PROC MIGRATE cannot be used.) In addition, most of your existing SAS programs can run on the SPD Engine files with little modification other than to the LIBNAME statement. Chapter 2, “Creating and Loading SPD Engine Files,” on page 11 provides details on converting default Base SAS engine data sets to the SPD Engine format.

Sharing the SPD Engine Files

The SPD Engine supports member-level locking, which means that multiple users can have the same SPD Engine data set open for INPUT (read-only). However, if an SPD Engine data set has been opened for update, then only that user can access it.

Features That Enhance I/O Performance

Overview of I/O Performance Enhancements

The SPD Engine has several new features that enhance I/O performance. These features can dramatically increase the performance of I/O bound applications, in which large amounts of data must be delivered to the application for processing.

Multiple Directory Paths

You can specify multiple directory paths and devices for each component type, because the SPD Engine can reference multiple physical files across volumes as a single logical file. For very large data sets, this feature circumvents any file size limits that the operating system might impose.

Physical Separation of the Data File and the Associated Indexes

Because each component file type can be stored in a different location, file dependencies are not a concern when deciding where to store the component files. Only cost, performance, and availability of disk space need to be considered.

WHERE Optimization

The SPD Engine automatically determines the optimal process to use to evaluate observations for qualifying criteria specified in a WHERE statement. WHERE statement efficiency depends on such factors as whether the variables in the expression are indexed. A WHERE evaluation planner is included in the SPD Engine which can choose the best method to use to evaluate WHERE expressions that use indexes to optimize evaluation.

Features That Boost Processing Performance

Automatic Sort Capabilities

The SPD Engine's automatic sort capabilities save time and resources for SAS applications that process large data sets. With the SPD Engine, you do not need to invoke the SORT procedure before you submit a SAS statement with a BY clause. When the SPD Engine encounters a BY clause, the data is not already sorted or indexed on the BY variable. The SPD Engine automatically sorts the data without affecting the permanent data set or producing a new output data set.

Queries Using Indexes

Large data sets can be indexed to maximize performance. Indexes permit rapid WHERE expression evaluations for indexed variables. The SPD Engine takes advantage of multiple CPUs to search the index component file efficiently.

Parallel Index Creation

In addition, the SPD Engine supports parallel index creation so that indexing large data sets is not time-consuming. The SPD Engine decomposes data set append or insert operations into a set of steps that can be performed in parallel. The level of parallelism depends on the number of indexes present in the data set. The more indexes you have, the greater the exploitation of parallelism during index creation. However, index creation requires utility file space and memory resources.

The SPD Engine Options

The SPD Engine works with many default Base SAS engine options. In addition, there are options that are used only with the SPD Engine that enable you to further manage the SPD Engine libraries and processing.

See:

□ Chapter 3, "SPD Engine LIBNAME Statement Options," on page 25

□ Chapter 4, "SPD Engine Data Set Options," on page 41

□ Chapter 5, "SPD Engine System Options," on page 77