

Schnell, schneller, SPDE

Raimund Storb
UCB Biosciences
Alfred Nobel Straße 10
40789 Monheim am Rhein
raimund.storb@ucb.com

Zusammenfassung

Die Scalable Performance Data Engine (SPDE), ursprünglich für die SAS Scalable Performance Data Server (SPDS) entwickelt gehören seit SAS Version 9.2 zu SAS Base dazu. Niemals umworben steht sie zwar zur Verfügung, wird aber im Allgemeinen ignoriert. Richtig eingesetzt vermag diese Engine jedoch Laufzeiten drastisch zu verkürzen. Zu beachten ist hier, dass die Allokation der SPDE Engine Bibliotheken sich von denen der Base Engine Bibliotheken grundsätzlich unterscheidet. Man kann getrennt den Speicherort für Dateiheader, Datenpartitionen und Indexpartitionen angeben.

Maximalen Performance Gewinn erzielt man, wenn man die Datenpartitionen und Indexpartitionen auf verschiedenen Devices ablegt. In diesem Falle kann zeitgleich auf alle Devices zugegriffen werden. Damit einher gehen einige Optionen um die I/O zu optimieren. Des Weiteren ermöglicht diese Engine einige Programme, die dem geneigten alten SAS Benutzer anormal erscheinen mögen. Zum Beispiel gruppiertes einlesen (By Statement) im Datastep ohne vorheriges sortieren und ohne Index. Es sei angemerkt, dass dieser Datastep durchaus schneller abgearbeitet werden kann als die entsprechende Sort Prozedur mit Daten in der Base Engine.

Schlüsselwörter: SPDE, Scalable Performance Data Engine

1 Einleitung

Seit 2005, mit dem Siegeszug der Mehrkernprozessoren, vervielfältigen sich die Möglichkeiten Programme, Prozesse parallel abzuarbeiten. Damit stellt sich die Frage ob und wie man mit einem I/O Orientiertem System wie SAS davon profitieren kann. Die Scalable Performance Data Engine implementiert eine Möglichkeit parallel auf (SAS) Dateien zuzugreifen.

In den Kapiteln "2. SPDE – Was ist das?" und "3. SPDE für Dummies" wird erklärt wie dies erreicht wird, worauf zu achten ist und wie sich das Ergebnis in den Verzeichnissen darstellt.

In den Kapiteln "4. SPDE – Die Stellschräubchen" und "5. Limitierungen" werden weitere Vorteile aber auch Limitierungen dieser Engine dargestellt.

In Kapitel "6. Performance im Vergleich" werden einige Vergleichszahlen tabelliert. Diese Zahlen sind zum Teil auf Testdaten und zum Teil auf US-Amerikanischen Krankenkassenabrechnungsdaten (Claims Data) ermittelt.

2 SPDE – Was ist das?

Zunächst einmal ist SPDE ein zusätzliches Akronym zu den vielen anderen, die es in der Industrie gibt. Es steht für Scalable Performance Data Engine. Auf Deutsch: Skalierbare Leistungs-Daten Maschine. Wem sich die Bedeutung nicht sofort erschließt dem sei versichert, das ist normal. Letztlich ist auch die Übersetzung von „Data Engine“ hier fragwürdig, da diese Worte hier eher der „SAS Sprache“ entspringen und nicht wirklich dem englischen. Mit anderen Worten, es handelt sich hier um „SAS-Fachchinesisch“. Füllen wir also die Worte sukzessive mit Sinn.

2.1 Skalierbare Leistung

Dies beschreibt die Fähigkeit der SPDE Ressourcen „skalierbar“ einzusetzen, d.h. eine Verdopplung der Ressourcen sollte auch in etwa eine Verdopplung der Verarbeitungsgeschwindigkeit bedeuten. Jedoch natürlich nur wenn es keine anderen Faktoren des Systems gibt, die dem entgegenstehen. Dies geschieht dadurch, dass die Arbeitslast auf verschiedene Ressourcen des gleichen Typs verteilt wird.

Z.B. sollte die Nutzung eines zweiten Laufwerkes die Verarbeitungsgeschwindigkeit verdoppeln. Dies natürlich nur unter der Prämisse, dass die CPU Leistung und die IO Kanäle des Systems dies erlauben.

Die folgenden Ressourcen werden unterstützt:

- CPU's
- CPU Kerne
- IO Kanäle
- Laufwerke
- Prozesse

2.2 Data Engine

Was ist nun aber die Bedeutung von „Data Engine“? Begeben wir uns also an die Übersetzung von SAS ins Deutsche. Eine Engine regelt die Kommunikation des SAS Systems mit anderen Systemen. Dies schließt die folgenden Systeme ein: Dateisysteme, andere Datenbanken, andere SAS Systeme (egal ob auf dem gleichen oder einem anderen Computer), named und unnamed pipes, Ftp Server etc. Benutzt werden diese Engines mittels Filename oder Libname Statements.

Jede dieser Engines hat ein eigenes Set von Optionen. SPDE ist eine Engine für (SAS) Bibliotheken, dem zufolge wird SPDE mit einem Libname Statement deklariert.

SAS Engines:

Filename: DISK, FTP, PIPE ...

Libname: BASE, EXCEL, ACCESS, REMOTE, MYSQL, DB2, XML SPDE

2.3 (SAS) Daten unter SPDE

SPDE nutzt 3 verschiedenen Typen von Dateien:

- Nutzdaten/Observationen. Die Dateiendung beginnt mit DPF.
- Metadaten. Die Dateiendung beginnt mit MDF
- Indexdateien. Die Dateiendungen beginnen mit IDX oder HBX

2.3.1 Nutzdaten

SPDE legt die Nutzdaten in verschiedene Segmente ab. Dies geschieht damit man auf diese Segmente parallel zugreifen kann. Dies erlaubt es auch, eine einzelne (SAS) Datei auf verschiedene Laufwerke zu verteilen und über verschiedene IO Kanäle mit verschiedenen Prozessen anzusprechen. Diese Prozesse dürfen gerne auf verschiedenen CPU's oder CPU Kernen ausgeführt werden. Genau dies macht die Leistung skalierbar. Im Folgenden ein Bild, wie sich diese Dateien im Windows Explorer darstellen. Die Endungen entschlüsseln sich wie folgt: Am Anfang steht dpf als Akronym für „Data Partition file“ gefolgt von einer eindeutigen Dataset ID, laufende Nummer des Segmentes, Version Nummer der Datei und dem Akronym spds9 für „Scalable Performance Data Server 9“. Der letzte Teil der Dateiendung zeugt von der Herkunft dieser Engine. Sie wurde ursprünglich Ende der 90'er Jahre für die Unix basierten „Scalable Performance Data Server“ entwickelt.

<input type="checkbox"/>	mdcrd092.dpf.00012ecd.2.3.spds9	11-Mar-15 17:43	SPDS9 File	947,037 KB
<input type="checkbox"/>	mdcrd092.dpf.00012ecd.5.3.spds9	11-Mar-15 17:44	SPDS9 File	951,320 KB
<input type="checkbox"/>	mdcrd092.dpf.00012ecd.8.3.spds9	11-Mar-15 17:44	SPDS9 File	950,645 KB
<input type="checkbox"/>	mdcrd092.dpf.00012ecd.11.3.spds9	11-Mar-15 17:46	SPDS9 File	944,991 KB

Abbildung 19: Beispiel für eine Liste von Nutzdateien

Es sei angemerkt, dass die Dataset ID eine hexadezimale Darstellung des Pfadnamens der Metadaten ist. Jedem der sich jetzt fragt, was denn damit passiert wenn man diese Dateien mit einem Dateibrowser verschiebt, wird das Kapitel 5, Limitierungen, wärmstens ans Herz gelegt.

Dem aufmerksamen Leser wird nicht entgangen sein das hier Dateien fehlen, denn zwischen der ersten Datei mit laufender Nummer 2 und der Zweiten mit laufender Nummer 5 fehlen zwei Dateien. Es sei hier angemerkt, dass diese Dateien in diesem Fall in zwei anderen Verzeichnissen gespeichert sind.

2.3.2 Metadaten

Die Metadaten, in „SAS“ auch „Meta Data Component File“ genannt, ist, mal wieder, ein Akronym. Hier für Meta Data File. Diese Datei enthält Informationen über Struktur und Lage der Nutzdaten (DPF Dateien, partitioned data component file ...)

Die Namenskonventionen ähneln denen der „Data Partition Files“. Man beachte das die Datei ID immer 0 ist, ebenso die laufende Nummer der Datei. Und die Endung beginnt mit dem Akronym „.mdf“.

2.3.3 Index Dateien

Jeder Index unter der SPDE Engine legt 2 verschiedene Typen von Dateien ab. Zum einen Dateien die einen globalen, Index enthalten und Dateien die den Index für die Segmente enthalten. Die Namenskonvention entspricht der der Nutzdaten bis auf den ersten Teil der Dateiendung. Diese beginnt mit den Literalen „HBX“ oder „IDX“ concateniert mit dem Variablennamen der indizierten Variable.

2.4 Das ist SPDE

SPDE erlaubt einen skalierbar performanten Datenzugriff. Dies wird erreicht durch parallelisieren des Zugriffs. Die Erfahrung lehrt dass der Zugriff auf Daten den Flaschenhals bei der Datenverarbeitung mit SAS darstellt. Folglich ist es ein probates Mittel diesen Zugriff effizienter zu gestalten, um eine insgesamt höhere Performanz zu erzielen.

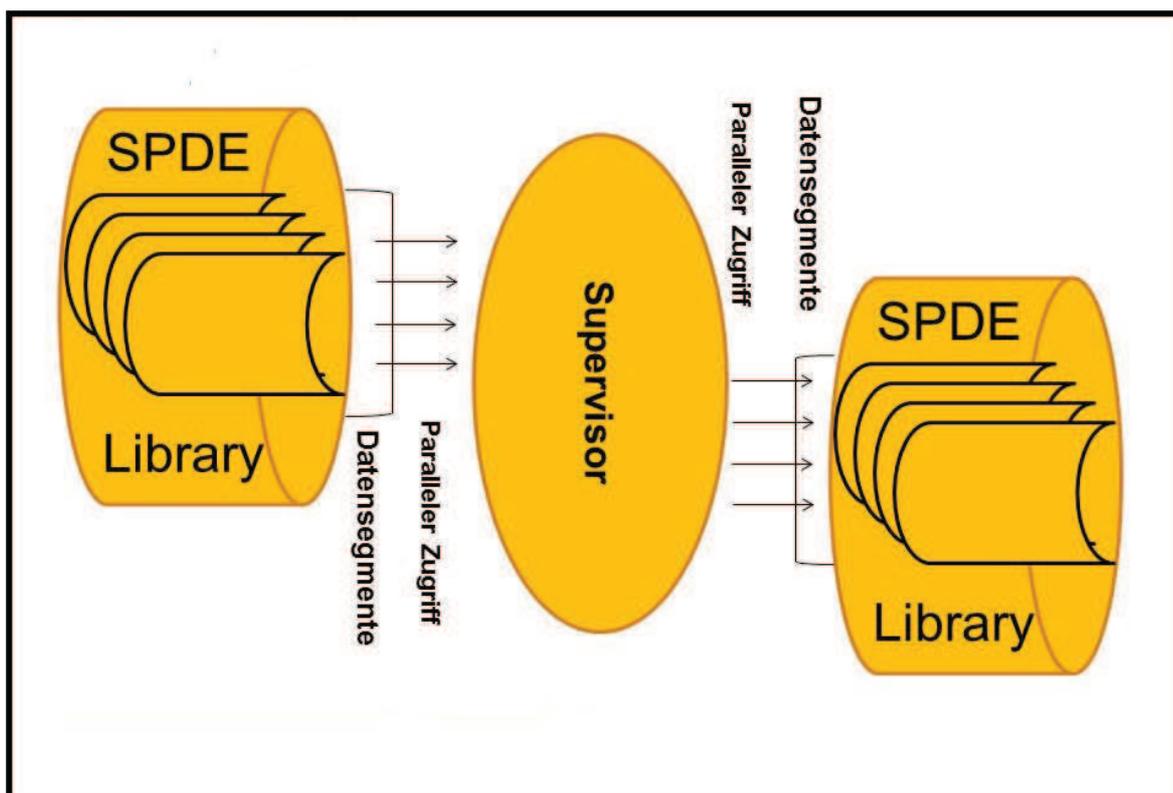


Abbildung 2: SPDE im grafischen Überblick

Es sei angemerkt das die eigentliche Verarbeitung der Daten in z.B. einem Data Step nach wie vor sequenziell erfolgt. Dazu übergeben die lesenden Prozesse der SPDE Engine die Daten an Prozess, auch Supervisor genannt. Und umgekehrt übergibt der Supervisor Daten an die schreibenden SPDE Prozesse. Alles natürlich unter der Prämisse

dass sowohl die zu lesende Datei unter dieser Engine angelegt wurde als die zu schreibende Datei. Schematisch ist dies in Abbildung 2 dargestellt. Es mag dem Leser helfen sich das Zusammenspiel besser vorzustellen.

3 SPDE für Dummies

Die meisten Neuerungen entspringen weniger einem Ausbruch an Genialität sondern der Neuinterpretation alter Ideen in neuem Kontext oder aber der Änderung der Wahrnehmung, d.h. es war immer schon da, wurde aber nicht wahr genommen. Für SPDE trifft beides zu. Diese Engine gibt es schon seit Ende der 90'er des letzten Jahrhunderts. Aber erst seit diese Engine mit SAS Base ausgeliefert wird und nicht mehr den SPDS Servern vorbehalten ist findet SPDE breitere Beachtung. Es gibt SPDE also schon länger. Doch in welchem Kontext ist eine derartige Arbeitsweise bekannt?

Stellt euch ein Büro oder eine Behörde vor.

Da gibt es verschiedene Sachbearbeiter die gleichzeitig das Gleiche tun. Anzeigen des zuständigen Sachbearbeiter erfolgt über Buchstaben oder Steuernummer oder oder oder. Außerdem sollte der Zugang zu den Sachbearbeitern so geregelt sein das die Kunden ohne Verzug ihren Sachbearbeiter erreichen.

Und wie findet der Sachbearbeiter seine Akten? Vermutlich benutzt er kleine Reiter am oberen Rand der Registratur mit Buchstaben darauf. Bert Brecht wird unser Sachbearbeiter dann unter „B“ suchen und keineswegs unter „Z“. Dort, unter „Z“ sucht er eher Konrad Zuse. Das unser Sachbearbeiter Konrad Zuse nicht unter „K“ sucht ist primär der Tatsache geschuldet das der erste Buchstabe des Nachname angezeigt wird. Kulturellen Gegebenheiten bedingen dies nur sekundär. Falls unserem Sachbearbeiter die Kompetenz/Information/Berechtigung fehlt den Vorgang zu bearbeiten dann reicht er es zu einem Experten/Chef etc. weiter.

Das Ganze stößt an seine Grenzen wenn die folgenden Situationen auftreten:

- Es sind mehr Sachbearbeiter als Kunden vorhanden.
- Nicht genug Platz für die Sachbearbeiter.
- Ungleichverteilung der Kunden auf die Sachbearbeiter.
- Unzureichende Berechtigung/Kompetenz der Sachbearbeiter. (Wenn der Chef zu viel selbst entscheiden will oder muss geht's halt schief.)

Um diese Vorgehensweise vorzubereiten bedarf es der folgenden Schritte:

- Indizieren der Akten.
- Aufteilen der Akten und Kunden.
- Bereitstellen der Sachbearbeiter.

Mit dem folgenden Schritt kann man garantiert diesen Prozess nachhaltig stören:

- Falsche Zuordnung von Kunden zu Sachbearbeitern/Akten.

Das „Übersetzen wir jetzt in SPDE“ in Tabelle 1:

Tabelle 4: Analogien zur Erklärung der SPDE Engine

„Das Leben“	„Die SPDE Engine“
Verschiedene Sachbearbeiter.	Parallele Prozesse
Anzeigen des zuständigen Sachbearbeiter.	Globaler Index (HBX Dateien)
Die Buchstaben an der Registratur.	Lokaler Index (IBX Dateien)
Der Experte.	Supervisor Prozess
Mehr Sachbearbeiter als Kunden.	Mehr Prozesse als SPDE „partitioned data component files“ (dpf Dateien)
Platz für die Sachbearbeiter.	Anzahl der CPU's, CPU Kerne etc. Alles was die Anzahl der parallelen Prozesse beeinflusst. Faustregel für SAS: Nicht mehr als die doppelte Anzahl der Kerne.
Zugang zu den Sachbearbeitern	IO Channels.
Ungleichverteilung der Kunden	Ungleiche Aufteilung der dpf Dateien.
Unzureichende Berechtigung der Sachbearbeiter	Gebrauch von Funktionalitäten die SPDE nicht unterstützt.
Indizieren der Akten.	Erzeugen von Index Dateien
Aufteilen der Akten und Kunden.	Anlegen der Datenpartitionen.
Bereitstellen der Sachbearbeiter.	Einbinden der SPDE Engine durch ein Libname Statement.
Falsche Zuordnung von Kunden zu Sachbearbeitern/Akten.	Speicherort der Metadaten, Datenpartitionen oder Indexdateien entsprechen nicht dem in den Metadaten enthaltenen Informationen. Dies ist leicht zu erreichen durch unterschiedliche Libname Statements zum Schreiben und lesen der Datei oder durch kopieren der Daten mit Betriebssystem eigenen Mitteln.

Und wie sieht das jetzt in der Praxis aus?

Der erste Schritt ist, die richtige Engine für eine Bibliothek zu wählen:

```
libname mybib SPDE /* Die Engine */
    "Pfad zu den Metadaten"
    datapath = ("Pfad 1" "Pfad 2" "Pfad 3" ...)
    indexpath = ("Pfad 1 zu den Indices" ...)
    partsize = <nnn>M TEMP=<NO|YES>;
```

Hierbei bitte folgendes beachten:

- Partsize Option gibt die Größe der Datenpartitionen vor. Bitte so wählen das auch wirklich mehrere Partitionen erstellt werde, und auch so dass die Anzahl der Partitionen die das Prozesse höchstens moderat übersteigt.

- Die Datapath und Indexpath Option ermöglichen es die Daten über die hier aufgeführten Verzeichnisse zu verteilen.
- Die Temp Option ermöglicht eine Temporäre Bibliothek der Work Library entsprechend. Nur benutzen wenn das auch gewünscht ist.

Und Wie kommen jetzt die Daten in diese Bibliothek?

Damit kommen wir zum 2. Schritt der zu tun ist.

Hier gibt es zwei Möglichkeiten:

- Die Daten werden in der Bibliothek unter SPDE mit einem Data Step oder einer Prozedur angelegt.
- Die Daten werden in die SPDE Bibliothek kopiert. Dazu empfiehlt sich die Append Prozedur.

Daten anlegen, hier einmal mit einem Data Step:

```
data SPDELIB.yourdata
  (index=(<Liste der zu indizierenden Variablen >) asyncindex=yes)
  Partsize = <Geschätztes Volumen in MB / 2 * Anzahl der Kerne>;
  set yourlib.yourdata;
  <Ganz viele tolle SAS-Kommandos>
run;
```

Die Option Asyncindex=Yes sorgt dafür das die Indices in parallelen Threads erzeugt werden. Partsize wird so gesetzt das es genug Datenpartitionen gibt um die Daten parallel zu schreiben und zu lesen. „2 * Anzahl der Kerne“ ist die Standard Anzahl an Prozessen mit denen SAS arbeitet. Je nach System kann es auch geboten sein die Anzahl der Prozesse mit der Option Threadnum zu verringern.

Daten kopieren mit proc append: bitte darauf achten das die Zieldatei nicht existiert!

```
proc append
  base = <Zu erstellende Daten>
  (partsize = <Datenvolumen / anzahl Prozesse>)
  data = <Geschätztes Volumen in MB / 2 * Anzahl der Kerne>
  asyncindex= yes;
run;
```

Die Option Asyncindex=Yes sorgt dafür das existierende Indices in parallelen Threads neu erzeugt werden. Da die SPD Engine andere Index Dateien benutzt als die Base Engine müssen die Index Dateien neu erzeugt werden.

Zum Schluss sei noch auf die folgenden zwei potentiellen Irrtümer hingewiesen:

- Die Standard Prozedur zum Kopieren ist die Copy Prozedur. Leder wird diese sogar an einigen Stellen der SAS eigenen Dokumentation als eine der Möglichkeiten neben der Append Prozedur genannt, an anderen Stellen jedoch darauf hingewiesen das man die Append Prozedur benutzen soll. Hintergrund ist hier die Tatsache das die Copy Prozedur allen gewählten Optionen zum Trotz immer Datenpartitionen von 64 MB realisiert. Im weniger ungünstigen Falle, bei kleineren Datensätzen konterkariert dies die Idee der parallelen Verarbeitung. Im weniger

günstigen Falle erzeugt dies so viele kleine Dateien das das Dateisystem nicht mehr performant arbeitet. Beide Effekte können durch sinnvolle Wahl der Partsize option mit Proc Append vermieden werden.

- Benutzen der SPD Engine mit kleinen Datensätzen führt zu keinerlei Performanz Steigerung. Dies ist der Tatsache geschuldet das keine Datenpartitionen unterhalb von 16 MB möglich sind.

4 SPDE – Die Stellschraubchen

Ziel dieses Kapitels ist es, eine Übersicht über die wichtigsten Optionen und Erwägungen im Zusammenhang mit SPDE zu geben. Dieses Kapitel widmet sich dem benötigten Data Management um den größtmöglichen Nutzen zu ziehen.

Diese Engine entwickelt ihre Stärke durch parallelen Zugriff auf Ressourcen des Computers. Also empfiehlt es sich hier einen Überblick über diese im Folgenden aufgelisteten Ressourcen zu haben.

- CPU's
- CPU Kerne
- IO Kanäle (und deren möglicher Datendurchsatz)
- Laufwerke (und deren verfügbare Größe)
- Prozesse (und wie viele davon günstiger weise wirklich in Anspruch genommen werden)

Die Anzahl der CPU's und CPU Kerne beschränken auf natürliche Weise die Anzahl Prozesse, die man sinnvollerweise benutzt. Die Laufwerke, ob nun physikalische Laufwerke, RAID Systeme, Netzwerklaufwerke sollte man entsprechend ihrer Leistungsfähigkeit benutzen. Bitte hierbei beachten das die Performanz der Laufwerke auch mit den IO Kanälen übereinstimmen muss. Eine Faustregel mag es hier geben, ich kenne keine. Lange Rede, kurzer Sinn, am Ende braucht es Abschätzungen für die folgenden Punkte:

- Wie viele Prozesse sollen es denn circa sein?
- Abschätzen welche Datasets oft benutzt werden und welche nicht.
- Welches Laufwerk bietet die höchste Performanz?
- In welchem Verhältnis soll der Platz auf den Laufwerken genutzt werden.
- In welchem Verhältnis steht die Zugriffsgeschwindigkeit der einzelnen Laufwerke.
- Welche Größe haben die Daten (unkomprimiert) auf den Laufwerken.

Der Rest der Überlegungen ist zielgerichtet.

1. Die Anzahl der Prozesse die SPDE benutzt wird eingestellt über:
 - a. Spdemaxthreads Option. Da dies beim Start einer SAS Session festgelegt werden muss lohnt es sich mit der Administration der Systeme zu sprechen.
 - b. Threadnum Dataset Option. Dies kann jeder Anwender für sich selbst erledigen, muss aber bei jedem Dataset aufs Neue gemacht werden.

- c. Abschätzen der Prozessanzahl entweder:
 - i. An einem System ohne Last. Ein passendes Programm findet man im Quick Guide to the SPD Engine Disk-I/O Set-Up auf der SAS Homepage.
 - ii. An einem System unter normaler Last. Schließlich soll es ja später auch auf einem ausgelasteten System benutzt werden. Ein passendes Programm befindet sich im Anhang.
2. Anlegen der Verzeichnisse für die Metadaten und der Indexdaten auf dem schnellsten Laufwerk.
3. Anlegen der Verzeichnisse für die meistgenutzten Daten, Anzahl der Verzeichnisse gemäß der Zugriffsgeschwindigkeit.
4. Kopieren/erzeugen der Daten aus 3). Dabei wird die Partsize Option pro Datei so gesetzt das so viele Datenpartitionen erzeugt werden wie es Prozesse gibt. Bitte beachten: Die Partitionen werden auf Basis der unkomprimierten Daten angelegt. (Das Volumen errechnet sich wie folgt: Länge aller Variablen multipliziert mit der Anzahl der Observationen.)
5. Verteilen der weniger genutzten Daten gemäß den Zielvorgaben auf die Laufwerke.

Und nun das Ganze in einem Beispiel.

Ein Zwei Prozessor System mit 6 Cores pro CPU, zwei RAID irgendwo in der SAN und einem lokalen Laufwerk. Der Zugriff zu den RAID ist ungefähr doppelt so schnell. Der Platz auf dem lokalen Laufwerk ist limitiert auf ungefähr 300 GB. Zwei Mount Points für die RAID Laufwerke seien /sasdata/tr/ und /sasdata/tr/analysis/ und /slow/ für die lokale Platte. Erste Erfahrungswerte indizieren den Gebrauch von 12 Prozessen.

Es sollen Krankenkassendaten abgelegt werden um sie gemäß Vorgaben der Epidemiologen der Drug Safety Abteilung, der Analysten des Marketings und anderer Abteilungen auszuwerten. Die Daten liegen für 5 Jahre vor. Die folgende Tabelle 2 gibt einen Überblick über die Dateien.

Tabelle 5: Überblick Claims Daten und ihre Größe

Größe der Dateien in GB	2009	2010	2011	2012	2013	Summe
I: Inpatient Admissions	0,8	0,8	0,8	0,8	0,8	3,9
F: Facility Header	48	47	54	55	47	2,1
S: Inpatient Services	17	16	15	15	13	76
O: Outpatient Services	167	154	166	174	152	813
D: Outpatient Drug Claims	54	49	53	53	46	255
P: Populations	0,2	0,3	0,3	0,3	0,3	1,4
A: Annual Summary Enrollment	12	11	13	13	11	60
T: Detail Enrollment	33	31	36	38	32	170
Summe	332	309,1	338,1	349,1	302	1630,3

Die anzulegenden Indices werden in Tabelle 3 dargestellt:

Tabelle 6: Überblick über die gewünschten Indices

Datei	Name	Index
I: Inpatient Admissions	CCA EI	Enrolid DX1-DX15 PROC1-PROC15
F: Facility Header	CCA EF	Enrolid DX1-DX3 PROC1-PROC6
S: Inpatient Services	CCA ES	Enrolid DX1 PROC1
O: Outpatient Services	CCA EO	Enrolid DX1-DX4 PROC1-PROC6
D: Outpatient Drug Claims	CCA ED	Enrolid NDCNUM
P: Populations	CCA EP	Enrolid
A: Annual Summary Enrollment	CCA EA	Enrolid
T: Detail Enrollment	CCA ET	Enrolid

Die meistgenutzten Daten sind hier die „Outpatient Service“ Dateien. Diese sollen also nun über alle 3 Laufwerke verteilt werden. Die Gruppierung der Daten in 5 Jahre soll beibehalten werden. Die Verzeichnisse werden gemäß Tabelle 4 angelegt. Dabei wird die last gemäß dem Verhältnis 2:2:1 auf die Raid Systeme und dem lokalen Laufwerk verteilt. Indices werden in der Planung berücksichtigt.

Tabelle 7: Mögliche Verteilung der Partitionen

Laufwerk	Verzeichnis	Typ
Raid 1	/sasdata/tr/<Jahr, 4 stellig>	Metadaten Partition
	/sasdata/tr/index/<Jahr, 4 stellig>	Index Partition
	/sasdata/tr/part1/<Jahr, 4 stellig>	Datenpartition
Raid 2	/sasdata/tr/analysis/index/<Jahr, 4 stellig>	Index Partition
	/sasdata/tr/analysis/part2/<Jahr, 4 stellig>	Datenpartition
	/sasdata/tr/analysis/part3/<Jahr, 4 stellig>	Datenpartition
Lokal	/slow/tr/part4/<Jahr, 4 stellig>	Datenpartition

Um den Speicherplatz auf der lokalen Platte nicht zu sprengen werden nur Teile der „Outpatient Service“ Tabellen dort abgelegt. Dazu wird die Bibliothek doppelt angelegt, einmal mit und einmal ohne die Partition auf der lokalen Platte. Die Bibliotheken werden also zuerst wie folgt allokiert:

```
Libname TR SPDE "/sasdata/tr/<Jahr, 4 stellig>"
    datapath = ("/sasdata/tr/part1/<Jahr, 4 stellig>"
                "/sasdata/tr/analysis/part2/<Jahr, 4 stellig>"
                "/sasdata/tr/analysis/part3/<Jahr, 4 stellig>")
    indexpath = ("/sasdata/tr/index/<Jahr, 4 stellig>"
                 "/sasdata/tr/analysis/index/<Jahr, 4 stellig>");
```

Im Anschluß werden die Dateien mit Ausnahme der „Outpatient Service“ Tabellen kopiert:

```
Proc append base = tr.CCAE<Datei typ kürzel>
    (partsize = 2^^floor(log(<Datenvolumen / 12>)) )
    data = old.CCAE<Datei typ kürzel>
    asyncindex= yes;
run;
```

Datei wird die Größe der Partitionen so bestimmt dass es auch mindestens so viele Datenpartitionen gibt wie Prozesse benutzt werden sollen. Die Option Asyncindex wird gesetzt damit vorhandene Indices parallel neu erzeugt werden. Sollten diese nicht vorhanden sein können diese mit Proc Dataset oder Proc SQL angelegt werden. Zum Kopieren wird nicht die Copy Prozedur benutzt da diese die Datenpartitionen immer in einer Größe von 64 MB anlegt.

5 Limitierungen und Unterschiede

Einige Features des SAS Systems sind eng mit der Engine verbunden. Hier kann es dann zu unerwarteten Verhalten kommen falls man sich vorher nicht damit auseinandergesetzt hat. Das betrifft vor allem die folgenden geänderten Eigenschaften.

- Es ist nicht möglich Dateien gleichzeitig zu verschlüsseln und zu komprimieren
- SPDE unterstützt den Remote Library Service nicht.
- Es ist nicht möglich der Copy Prozedur vorzugeben wie groß die Datenpartitionen werden dürfen. Es wird ein Default von 64 MB genommen. Dies kann bei großen Datenvolumen zu Problemen des Betriebssystems führen. Man sollte diese Prozedur vermeiden wenn man mit SPDE arbeitet.
- Je nach gewählten Optionen kann es sein das beim sortierten einlesen einer unsortierten Datei (set / by) kein Fehler generiert wird sondern die Datei beim Einlesen sortiert wird.
- Es gibt dokumentierte Fälle in denen die Transpose Prozedur nicht mehr performant ist.
- SPDE kann nicht mit SAS Formaten umgehen. Selektion von Daten mittels Formate erfolgt nicht parallel. Genauer gesagt erst nachdem die Observationen eingelesen sind.

6 Performance im Vergleich

Um die ersten Vergleiche anzustellen wurde ein kleines Programm benutzt, das die folgenden Schritte enthält. Anlegen einer Testdatei, lesen und schreiben der Testdaten, Selektion auf den Testdaten, sortieren (nur Base Engine), gruppiert verarbeiten (in der in Base neu angelegten Ordnung), sortieren in alter Ordnung. Ausgeführt auf einem Rechner mit einem 6 Core Prozessor, Daten auf einem SD Data Array. Es sei angemerkt dass die erzeugten Testdaten mit 2002 Bytes recht große Observationen besaßen.

Die folgende Tabelle 5 enthält die resultierenden Laufzeiten.

Tabelle 8: Schritte des Testprogramms und die Laufzeiten im Vergleich

		Base	SPDE	Ratio
1	Erzeugen der Testdaten	5.59	4.91	0.88
2	Primär lesen und schreiben	10.35	8.17	0.79
3	Selektieren Testdaten.	4.18	0.57	0.14
4.a	Sortieren (Nur Base)	40.78	Implizit	0.42*
4.b	Verarbeiten mit „By Group“	18.82	25.26	1.34
5	Sortieren in vorherige Ordnung	39.81	17.88	0.45

*0,42=Ratio für Gesamt 4, Laufzeiten in Sekunden.

Im Folgenden einige Vergleichszahlen ermittelt auf Krankenkassenabrechnungsdaten. Diese Tests entsprechen den Anforderungen nach dem was in der Epidemiologie u.U. wirklich gebraucht wird. Auch wenn die hier verwendeten Datensätze nicht die Größten sind die Verwendung finden. Getestet wurde auf einem SAS Connect Server mit konventionellen (RAID 5) Laufwerken.

Tabelle 9: SPDE im Vergleich

Abfrage	Datei	Base	SPDE	Base mit Index	SPDE mit Index	Selektierte Obs.
Abfrage mit Format auf eine Liste von Medikamenten	Verschreibungen	1:43.37	1:15.87	1:37.64	1:10.87	40.600.802
Abfrage mit Format auf eine Liste von Behandlungen	Praxis Abrechnungen	8:09.37	6:34.88	3:41.56	5:34.01	152.890.673
Abfrage mit Format auf eine Liste von Diagnosen	Praxis Abrechnungen	6:13.75	4:01.59	7:50.16	8:51.12	173.045.679
Abfrage mit Format auf eine Liste von Patienten	Praxis Abrechnungen	54.87	1:52.56	55.63	1:49.59	457.413
Einfache Abfrage auf 4 verschiedene Diagnosen, oder Verknüpfung	Praxis Abrechnungen	20.60	1:20.91	20.93	0.79	67.178
Einfache Abfrage auf 4 Patienten, „in Abfrage“	Praxis Abrechnungen	1:10.70	1:47.10	0.01	0.01	24

Generell zeigt sich, dass bei steigendem Volumen der Selektion die Vorteile von SPDE überwiegen. Das vermag dann auch den Nachteil ausgleichen das SPDE nicht mit SAS Formaten arbeitet. Natürlich zeigt sich auch dass ein geeigneter Index viel bewirken kann. Je schlanker ein Datensatz ist, d.h. je kürzer die einzelne Beobachtung ausfällt, desto ausgeprägter zeigen sich die Vorzüge der SPDE. Es sei noch darauf hingewiesen dass die benutzte Hardware nicht für SPDE ausgelegt war.

Literatur

- [1] Daniel Sargent Clinton Rickards (1998). The Scalable Performance Data Server: Real World Experiences, NESUG 1998
<http://www.lexjansen.com/nesug/nesug98/dbas/p006.pdf>
- [2] Haibin Shu (2007) Unleashing the Power of SPDE Under Windows NESUG (2007) <http://www.lexjansen.com/nesug/nesug07/ap/ap18.pdf>
- [3] SAS® 9.3 Scalable Performance Data Engine: Reference
<http://support.sas.com/documentation/cdl/en/engspde/62981/PDF/default/engspde.pdf>
- [4] SAS 9.4 Scalable Performance Data Engine, ReferenceSecond:
<http://support.sas.com/documentation/cdl/en/engspde/67454/PDF/default/engspde.pdf>

Ein Beispielprogramm zur Abschätzung einer guten Einstellung der Prozessanzahl sowie Beispiele einiger Plots zur Visualisierung der Performance bei Benutzung verschiedener Anzahlen von Prozessen können vom Autor angefordert werden.