Der DataStep 2 – Features, Fakten, Fragezeichen

Andreas Menrath
HMS Analytical Software GmbH
Rohrbacher Str. 26
69115 Heidelberg
Andreas.Menrath@analytical-software.de

Zusamenfassung

Der vorliegende Beitrag befasst sich mit der Datenverarbeitung mittels PROC DS2. Der Fokus liegt jedoch nicht darauf, eine Einführung in die Syntax und Programmierung zu geben, sondern vielmehr darauf, die technischen Konzepte hinter dem DS2 zu vermitteln und die Mehrwerte sowie Grenzen des DS2 aufzuzeigen.
Nach der Lektüre ist der Leser im Stande eine Bewertung durchzuführen, ob und in welchen Fällen sich der Einsatz von DS2 im persönlichen Programmieralltag anbietet.

Schlüsselwörter: PROC DS2, FedSQL, ANSI SQL, Threading

1 Einleitung

Mit der Version 9.4 hat SAS eine neue Möglichkeit der Datenverarbeitung mit PROC DS2 eingeführt. Viele Entwickler stehen nun vor der Frage, was sich mit dem Datastep 2 in ihrem Alltag ändern wird. Welche Konzepte sind gleich? Worin bestehen die Unterschiede? Warum sollte man sich mit dem DS2 beschäftigen? Muss Bestandscode nun auf DS2 migriert werden?

Diese und ähnliche Fragen stellt sich vermutlich jeder SAS Programmierer, der zum ersten Mal von DS2 hört. Dieser Artikel versucht, die wichtigsten Fragen zu beantworten und auf abstrakter Ebene die Notwendigkeit für DS2 aufzuzeigen. Außerdem sollen die dahinter liegenden Konzepte und Einsatzszenarien vermittelt werden. Durch eine Gegenüberstellung der Vor- und Nachteile sollte der Leser in die Lage versetzt werden, selbst zu entscheiden, ob die DS2-Programmierung für die eigenen Projekte einen Mehrwert liefern kann oder nicht.

2 Welche Probleme löst DS2?

Um zu verstehen, warum SAS den DS2 überhaupt ins Leben gerufen hat, sollte man zunächst die Historie der Datenverarbeitung mit SAS erläutern.

SAS hat seine Wurzeln auf dem Großrechner in den 80er Jahren. Trotz mehrerer grundlegender Überarbeitungen der SAS-Software findet die Datenverarbeitung über den klassischen DataStep noch wie vor 40 Jahren statt: rein sequentiell auf nur einem einzigen Prozessorkern.

Mittlerweile erweist sich diese Vorgehensweise nicht mehr als zeitgemäß. Moderne Rechner verfügen ausnahmslos über mehrere Prozessorkerne und eine Beschleunigung der Datenverarbeitung lässt sich heute nur noch über Parallelisierung erzielen.

Bei Prozeduren hat SAS mit den High-Performance-Prozeduren bereits eine automatische Parallelisierung der Berechnungen implementiert. Verwendet man beispielsweise statt der PROC SUMMARY eine PROC HPSUMMARY wird die Datenverarbeitung innerhalb der Prozedur automatisch auf mehrere Prozessoren (oder falls ein SAS GRID zur Verfügung steht sogar auf mehrere Rechner) verteilt, um dem Anwender schneller das Ergebnis ausgeben zu können.

Die nicht parallelisierbare Datenaufbereitung per DataStep wird daher zunehmend zum Performance-Flaschenhals in der Datenverarbeitungskette.

Doch damit nicht genug. Die moderne Datenverarbeitung stellt auch noch weitere Anforderungen an die Software. Denn: Big Data ist längst in der Realität angekommen. Firmen bauen bereits heute Data Lakes auf – mit Datenmengen, die vor wenigen Jahren noch unvorstellbar (bzw. unbezahlbar) gewesen wären.

Neben den typischen strukturierten Daten (Tabellen, CSV-Format) tauchen aber auch zunehmend unstrukturierte und semistrukturierte Daten (z.B. in Form von JSON-Dokumenten) auf. Daten sind außerdem nicht länger nur statisch, sondern werden „on demand“ (z.B. über Webservices) um weitere Informationen angereichert oder gleich in Realtime verarbeitet und gar nicht mehr gespeichert (z.B. bei Telemetriedaten von Maschinen). Auch Trends wie „In Database Processing“ versprechen deutliche Performancevorteile, da die Analytic direkt in der Datenbank gerechnet wird und die Daten nicht mehr langwierig von der Datenbank zur Analytic Engine übertragen werden müssen.

SAS adressiert diese Herausforderungen durch eine Vielzahl von spezialisierten, neuen Produkten, die in den letzten Jahren entstanden sind: SAS Visual Analytics, SAS GRID, SAS Event Stream Processing, SAS Viya, SAS In-Database Code Accelerator, u.v.m. . Im Kern einiger dieser Produkte spielt der DS2 eine tragende Rolle und adressiert die zuvor aufgezeigten Probleme der Moderne.

3 Konzepte hinter DS2

Hinter DS2 verbirgt sich nicht einfach nur eine weitere SAS-Prozedur. DS2 ist eine eigenständige Laufzeitumgebung, ähnlich wie beispielsweise SAS/IML für In-Memory Matrizenrechnung. Die DS2-Runtime ist relativ klein und kann auch auf Systemen installiert werden, auf denen keine weiteren SAS Komponenten installiert sind, z.B. auf einem Datenbankserver oder den Knoten eines Hadoop-Clusters. Auch in einer Vielzahl von SAS Produkten ist die DS2 Laufzeitumgebung verfügbar. Hierzu zählen: SAS BASE, der SAS Federation Server, der SAS LASR Analytic Server (Visual Analytics/Visual Statistics), SAS Event Stream Processing und einige mehr.

Der DataStep 2 hat daher mit dem klassischen DataStep nicht mehr viel gemeinsam, da es sich um eine komplette Neuentwicklung handelt. Trotzdem haben die Entwickler darauf geachtet, dass sich der DataStep 2 so weit wie möglich wie ein klassischer DataStep anfühlt und die Syntax ähnlich ist. Die meisten Funktionen sind auch im DS2 verfügbar und verhalten sich gleich. Andererseits fehlen eine Reihe von bekannten DataStep Funktionalitäten: So gibt es z.B. keine Statements zum Einlesen von Rohdaten (Input Statement, Infile Statement, Datalines Statement, usw.). Konzeptionell gibt es keine CALL Routinen mehr, sondern Funktionen ohne Rückgabewert. Auch einige spezialisierte Funktionen wie z.B. DOSUBL werden nicht unterstützt. Nach und nach wird der Funktionsumfang von SAS jedoch erweitert: So kam beispielsweise im Maintenance Release 4 die lange vermisste LAG-Funktion hinzu.

Die Neuentwicklung des DS2 ermöglicht es aber auch, neue Konzepte einzuführen. So kann der DS2 nun sämtliche ANSI-SQL-Datentypen verarbeiten. Während in der klassischen Welt nur Fließkommazahlen und Texte mit fester Breite existieren gibt es im DS2 nun also auch BOOLEAN, VARCHAR, INTEGER, DECIMAL, DATETIME, usw.

Berechnungen profitieren davon, da nun auch eine höhere Genauigkeit als Double-Werte erreicht werden können. Auch im ETL-Umfeld ist es nun möglich, Daten aus einer Datenbank zu lesen und direkt in eine zweite Datenbank zu schreiben – ohne dass die Datenfelder umständlich konvertiert werden müssen.
Der DS2 bringt auch eine Reihe sogenannter PACKAGEs mit. Hierbei handelt es sich um abgegrenzte, wiederverwendbare Module wie man es von Klassen in der Objektorientierten Programmierung kennt. Zu den wichtigsten Packages zählen:

- HTTP (Abfrage von Webseiten, Interaktion mit WebServices)
- JSON (Funktionalität um JSON Dokumente zu verarbeiten)
- SQLSTMT (dynamische SQL Abfrage an eine Datenbank senden)
- SQLEXEC (SQL Statements innerhalb einer Datenbank ausführen)
- MATRIX (Funktionalität für Matrizenrechnung, ähnlich SAS/IML)
- HASH (Hash-Tables)

Mit jedem neuen Maintenance Release führt SAS auch neue PACKAGES hinzu bzw. erweitert deren Funktionsumfang. Darüber hinaus ist es aber auch möglich eigene Packages zu programmieren und somit wiederkehrende Fachlogik in einem wiederverwendbaren Modul zu kapseln.

Zu den größeren Neuerungen des DS2 zählt FedSQL. Der DS2 ist darauf ausgelegt, Daten in strukturierter Form aus einem SAS Dataset oder von einer Datenbank zu beziehen. Etwas ungewohnt ist es, dass man innerhalb eines DS2 beispielsweise kein MERGE Statement findet, um Daten aus zwei Tabellen zusammenzuführen.

Hier kommt FedSQL ins Spiel: Datenabfragen im DS2 können nun auch direkt als FedSQL Abfrage definiert werden. Innerhalb dieser Abfrage können Daten über SQL JOINs miteinander verknüpft werden, selbst wenn die Tabellen aus unterschiedlichen Datenbanken oder SAS Libraries kommen. Im Hintergrund kümmert sich die FedSQL Engine darum, dass die Daten möglichst effizient aus den einzelnen Quellen abgefragt und die Daten bereits aufbereitet an den DS2 übergeben werden. Weiterer großer Vorteil der FedSQL Engine im Vergleich zur klassischen Datenverarbeitung: FedSQL arbeitet multithreaded und kann somit Daten parallel auf mehreren CPU-Kernen verarbeiten.

Einen großen Mehrwert liefert der DS2 beim sogenannten In-Database Processing. In diesem Fall wird vom DS2-Programm ein Teil der Datenverarbeitung an die Datenbank ausgelagert. Der Quellcode des THREAD Blocks im DS2-Programm wird als Code direkt an einen oder mehrere Datenbankserver geschickt und dort ausgeführt. Anstelle von SQL führt die Datenbank selbst den DS2-Code aus und kann so zum Beispiel komplexe Scoring-Modelle (die in klassischem SQL nicht abzubilden sind) direkt auf die Daten anwenden.

Hierzu muss auf allen Datenbankrechnern die DS2-Laufzeitumgebung in Form des SAS Code Accelerator installiert sein. SAS unterstützt aktuell Hadoop, Greenplum und Teradata für diese Form der Datenverarbeitung. Nachdem der DS2 Code in der Datenbank ausgeführt wurde, werden die berechneten Daten zurück an das DS2-Hauptprogramm geschickt und können dort weiterverarbeitet werden, z.B. um die Teilergebnisse der einzelnen Server zusammenzuführen.

Die Nachteile des DS2 sollen aber auch nicht verschwiegen werden. Zunächst wäre da der hohe Einarbeitungsaufwand zu nennen. Neben den neuen Konzepten muss der SAS Programmierer sich an neue Syntaxelemente gewöhnen. Er muss lernen, die z.T. kryptischen Fehlermeldungen richtig zu interpretieren, sich mit FedSQL befassen und an der ein oder anderen Stelle mit Überraschungen rechnen. So ist das Einfügen des Inhalts einer Makrovariablen in den DS2 über das %TSLIT Makro höchst ungewohnt. Datasets werden standardmäßig nicht überschrieben und ersetzt; versucht es der Entwickler dennoch, so kommt es zu einem Fehler, wenn nicht explizit die Dataset-Option *(overwrite=yes)* gesetzt wird. Auch globale Optionen, wie z.B. COMPRESS, wirken

sich nicht direkt auf die Ausgabetabellen eines DS2 aus. Auch hier muss die Kompression explizit im Programmcode angegeben werden.

Standardmäßig verarbeitet aber auch der DS2 die Daten nicht parallel, sondern sequenziell. Möchte man mit dem DS2 Daten tatsächlich parallel verarbeiten, so muss man die Programmteile, die parallel ausgeführt werden sollen, explizit ausprogrammieren. Hierdurch wächst die Menge an Programmcode auch relativ schnell an und die Übersichtlichkeit leidet darunter.

Der Einarbeitungsaufwand für DS2 ist also nicht zu unterschätzen. Durch die veränderte Form der Datenverarbeitung ergeben sich aber auch noch weitere Nachteile. Durch die parallele Datenverarbeitung entsteht ein nicht zu unterschätzender Overhead, da die Daten zunächst auf mehrere Threads verteilt und anschließend auch wieder aus mehreren Threads mit Teilergebnissen zusammengeführt werden müssen. Auch die nun häufiger auftretenden Datentypkonvertierungen zwischen den ANSI-Datentypen und SAS-Datentypen zehren an der Performance.

Sollen nur Tabellen aus einer Datenquelle in eine andere übertragen werden, so ist der DS2 sogar spürbar langsamer als die Verarbeitung im klassischen Datastep. Insbesondere die SAS BASE Libname Engine arbeitet weiterhin mit nur einem einzigen Thread und verhindert somit, dass DS2 seine Stärken ausspielen kann. DS2 und parallele Datenverarbeitung machen also erst im Zusammenspiel mit Datenbanken und größeren Datenmengen wirklich Sinn.

Zuletzt soll auch noch ein weiterer schwerwiegender Nachteil der parallelen Datenverarbeitung genannt werden: Die Reihenfolge der Datensätze in der Ausgabetabelle ist nicht mehr deterministisch. Wird der gleiche DS2-Step mehrere Male hintereinander ausgeführt, so zeigt sich, dass die Daten meist in unterschiedlicher Reihenfolge geschrieben werden. Insbesondere für Unittests ist dies ein echtes Problem. Ebenso tritt ein Problem auf, wenn Sie sortierte Daten einlesen und nach der DS2-Verabeitung implizit davon ausgehen, dass sich die Sortierreihenfolge nicht geändert hat.

Spielt die Reihenfolge der Datensätze eine Rolle, so müssen die Daten nach der Verarbeitung noch einmal mit einem PROC SORT „nachbearbeitet“ werden, was sich wiederum negativ auf die Gesamtlaufzeit des Programms auswirkt.

4 Fazit

Als persönliches Fazit zieht der Autor den Schluss, dass für die typische Datenverarbeitung in SAS-Programmen der klassische DataStep weiterhin die erste Wahl bleibt. Erst wenn tatsächlich die Performance der SAS Programme zum Problem wird und sich nicht mehr durch Codeoptimierungen verbessern lässt, sollte man überlegen, ob die Verarbeitung durch Parallelisierung oder durch Verlagerung der Datenverarbeitung auf die Datenbank durch DS2 beschleunigt werden kann.

Darüber hinaus bietet sich der DS2 an, wenn eine seiner Stärken einen deutlichen Mehrwert bringt: Beispielsweise in Form von genaueren Berechnungen (ANSI Datentypen). Aber auch, wenn komplexe Datenverarbeitungsschritte in wiederverwendbaren PACKAGE Modulen gekapselt oder Webservices mit semistrukturierte Daten konsumiert werden.

Für den SAS-Anwender und Entwickler hingegen wird es zukünftig wichtiger werden, DS2-Code lesen und verstehen zu können, da zunehmend mehr SAS Produkte DS2-Code generieren.