SSMB Benchmark für SAP HANA

Datenbanken

JAN HOFMEIER, MARIUS JOCHHEIM, LION SCHERER,
KRISTINA ALBRECHT

2018-03-06

Tabellenverzeichnis

Abbildungsverzeichnis

List of Listings

1 SAP HANA

2 Generell HANA als in-memory Datenbank

SAP Hana (Die High Performance Analytic Appliance) ist eine Entwicklungsplattform und bestehtim Kern aus einer „in-memory“ Datenbank.

Transaktionen und Analysen werden aufeiner einzigen, singulären Datenkopie im Hauptspeicher verarbeitet, anstatt dieFestplatte als Datenspeicher zu benutzen. Dadurch ist es möglich sehr komplexeAbfragen und Datenbankoperationen mit sehr hohen Durchsatz auszuführen.

Hana verbindet OLTP, durch die SQL undACID (Atomicity, Consistency, Isolation andDurability) Kompatibilität, und OLAP durch das „in-memory“ feature.Durch das ACID Prinzips ist die Datenbank geeignet um Unternehmensinterne Datenzu speichern. Es ist nicht nötig Datenanalysen über einen ETL Prozess an einDatawarehouse weiterzuleiten. Komplexe Echtzeit Analysen [1]könnennun direkt durch SAP Hana durchgeführt werden. Das erspart die erheblichenKosten und vor allem Zeit.

Beim der „in-memory“ Technologie werdendie Daten im Hauptspeicher anstatt auf elektromagnetischen Festplattengespeichert. Antwortzeiten und Auswertungen können dadurch schneller als beigewöhnlichen Festplatten durch den Prozessor vorgenommen werden. Dadurch, dassder Zugriff auf die Festplatte nun wegfällt, verkürzt sich dieDatenzugriffszeit bis auf das Fünffache.

img

https://intellipaat.com/blog/what-is-sap-hana/

Um nun aber dem „D“ des ACID Prinzips gerechtzu werden reicht eine Speicherung im füchtigen Hauptspeicher nicht. Für die Datensicherungmüssen deshalb traditionelle Festplatten benutzt werden. Diese werden bei derreinen Analyse von Daten nicht berücksichtigt. Wenn Transaktionen getätigtwerden, müssen die regelmäßig an das nicht flüchtige Speichermedium übergebenwerden. Außerdem wird dort zu jeder Transaktion ein Protokolleintraghinterlegt.

---

[1]https://intellipaat.com/interview-question/sap-hana-interview-questions/

2https://link.springer.com.ezproxy.dhbw-mannheim.de/book/10.1007%2F978-3-658-18603-6

3https://www.sap.com/germany/products/hana.html#pdf-asset=2caaec36-847c-0010-82c7-eda71af511fa&page=3

- In-Memory Datenbank
- Column-Based Architektur
- Komprimierung

- Memory Zugrigge

3 Star Schema Benchmark (SSBM)

Der Star Schema Benchmark (SSB) wurde von Pat O'Neil, Betty O'Neil und Quedong Chen entwickelt, um die Performance von Datenbanksystemen, welche mit Data-Marts nach dem Star Schema arbeiten, zu ermitteln und Vergleichbar zu machen [Star Schema Benchmark Quelle]. Dabei nutzen sie das bekannte TPC-H Benchmark [TPCH Quelle] als Grundlage für ihr Star Schema Benchmark, modifizieren es jedoch vielfach zugunsten eines guten Star Schemas.

TPC-H zu SSB-Transformation

Die von Chen, O'Neil und O'Neil durchgeführten Transformationen von TPC-H zu SSB wurden an die von Kimball und Ross erläuterten Prinzipien zur Dimensionalen Modellierung [**The Data Warehouse Toolkit Second Edition - Quelle einfügen**] angelehnt.

— Hier SSB-M Schema Grafik einfügen —

Im Folgenden sind die wichtigsten Änderungen kurz zusammengefasst:

1. Die beiden Tabellen LINEITEM und ORDER aus dem TPC-H Schema werden in SSB zu einer gemeinsamen Tabelle LINEORDER zusammengefasst, was als Denormalisierung bezeichnet wird [**The Data Warehouse Toolkit Seite 121 - Check**]. Dadurch werden für gängige Abfragen weniger Joins benötigt. Die Kardinalität der Tabelle entspricht der ursprünglichen LINEITEM Tabelle und beinhaltet einen replizierten ORDERKEY zur Verknüpfung der Tabellen.
2. Die Tabelle PARTSUPP aus dem TPC-H Schema wird nicht in das SSB übernommen, da die Granularität zwischen PARTSUPP und LINEORDER nicht übereinstimmt. Dies kommt daher, dass LINEORDER bei jeder Transaktion vergrößert wird, die PARTSUPP Tabelle jedoch nicht. Sie hat lediglich die Granularität Periodic Snapshot, da es keinen Transaction Key für sie gibt. Auch im TPC-H Schema gibt es keine Aktualisierungen über den Verlauf. Damit bleibt sie im Gegensatz zur LINEORDER Tabelle über den Zeitverlauf unverändert.

Dies würde kein Problem darstellen, wenn PARTSUPP und LINEORDER durchgehend als getrennte Faktentabellen behandelt würden, welche nur getrennt abgefragt und nie zusammengefügt werden. Jedoch zeigt Abfrage Q9 aus dem TPC-H Schema, dass LINEITEM, ORDERS und PARTSUPP kombiniert werden, womit Konflikte entstehen.

Die Autoren des SSB-M argumentieren, dass die PARTSUPP Tabelle im Kontext eines Data Marts unnötig ist, woraus die Löschung der Tabelle erfolgt. Stattdessen wird eine Spalte SUPPLYCOST aus der Tabelle zu jeder LINEORDER Zeile im neuen Schema hinzugefügt. Dadurch wird die Korrektheit der Information in Bezug zur Bestellzeit sicher gestellt.

TODO: Für andere Transformationsdetails von TPC-H zu SSB verweisen wir den Leser auf [Star Schema Benchmark]. Beispielsweise werden die Spalten TPC-H SHIPDATE, RECEIPTDATE und RETURNFLAG gelöscht, da die Bestellinformationen vor dem Versand abgefragt werden müssen, und wir wollten uns nicht mit einer Folge von Faktentabellen befassen, wie in [Kimball, Ross], pg. 94. Außerdem hat TPC-H keine Spalten mit relativ kleinem Filterfaktor, daher fügen wir eine Anzahl von Rollup-Spalten hinzu, wie P_BRAND1 (mit 1000 Werten), S_CITY und C_CITY und so weiter.

- Warum SSBM? Für Dimensionale Modellierung, interessant für OLAP
- Unterschiede zu TPC-H ausarbeiten anhand SSB-M Schema, Quellen, Bilder
- Generierung von SSBM-Tabellen
- Tabellen in HANA laden

4 SQL-Abfragen für SSBM

- Anpassung der TPC-H Queries auf SSBM
- Generierung von Abfragen mit Qgen

5 Durchführung von Benchmarks

Aufsetzen von HANA: Installation, Beschreibung vom System (Prozessoren, RAM, OS, Festplattenspeicher etc.)

Durchführung von Performance Tests und Auswertung der Query Execution Plans

- Row vs. Column Store
- Wie kann man durch Indizes oder Hints beschleunigen?
- Parallele Zugriffe (Concurrency, unter Umständen)

6 Fragen

- Out of Memory Problem
- Zugriff auf HANA an der DH
- Gibt es Standard-Queries für SSBM?
- Wie viele Queries? Skalierung?
- Welche Metriken sind bei dem Benchmark wichtig?
  Laufzeit, (Systembelastung)