Das spaltenorientierte MySQL-Plugin Infobright als Kern einer Open Source basierten Data-Warehouse-Infrastruktur

Olaf Herden, Tobias Haller

Fakultät Technik
Studiengang Informatik
Duale Hochschule Baden-Württemberg Campus Horb
Florianstr. 15
72160 Horb
o.herden@hb.dhbw-stuttgart.de
t.haller@hb.dhbw-stuttgart.de

Abstract: In den letzten Jahren sind zwei Trends im Data-Warehouse-Umfeld zu beobachten: Zum einen haben sich Open-Source-Komponenten zu ernst zu nehmenden Lösungen entwickelt, zum anderen ist ein starkes Wachstum der Daten zu verzeichnen, welches neue Datenbankkonzepte verlangt. Eines hiervon ist die spaltenorientierte Speicherung und Kompression von Daten, was durch das MySQL-Plugin Infobright angeboten wird. Der Beitrag beschreibt die in Infobright realisierten Konzepte und zeigt anhand von Evaluationen Vorteile und Grenzen gegenüber der herkömmlichen, zeilenorientierten Speicherung auf.

1 Einleitung

Seit nunmehr 15 Jahren haben Data-Warehouse-Systeme (DWH) eine starke Verbreitung in der Praxis gefunden [BG08, Ki96]. Während die ersten Jahre durch komplexe kommerzielle Closed-Source-Systeme gekennzeichnet waren, haben sich seit ein paar Jahren Open-Source-Lösungen am Markt etabliert. Während erste Ansätze noch Prototyp-Status besaßen und für den praktischen Einsatz nur bedingt zu verwenden waren [HP05], haben sich diese inzwischen zu kompletten Produktsuiten weiterentwickelt [HT+10]. Diese bilden eine Alternative für Unternehmen, die bisher kommerzielle Closed-Source-Lösungen einsetzen und eröffnen für kleine und mittelständische Organisationen eine Option für Data-Warehouse-Umgebungen.

Das Datenvolumen in DWHs wächst ständig an. Gründe hierfür sind z.B. natürliches Wachstum von Daten im Geschäftsumfeld, der Wunsch nach differenzierteren Analysen und das Erschließen neuer Anwendungsbereiche für analytische Informationssysteme. Ein weiterer Aspekt in der jüngeren Vergangenheit sind automatisch generierte Daten, z.B. von RFID(Radio Frequency Identification)-Sensoren, Netzwerkkomponenten oder Terminals.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Bis vor wenigen Jahren ausschließlich und in der Gegenwart hauptsächlich werden klassische relationale Datenbanksysteme wie z.B. Oracle, Microsoft SQL Server oder IBM DB2 als DWHs eingesetzt. All diese Systeme speichern ihre Datensätze zeilenorientiert ab. Diese Technologie stößt mit wachsendem Datenvolumen und komplexeren Applikationen jedoch an ihre Grenzen. Wesentlicher Faktor ist dabei der Festplatten-I/O, da die Zugriffszeiten auf Platten in den letzten Jahren nahezu konstant geblieben sind. So werden mit zunehmender Tabellengröße Anfragen immer langsamer, existieren zusätzliche Indizes verlangsamen sich auch die Ladezeiten wegen deren Pflege.
Die populärsten, klassischen Gegenmaßnahmen sind Tuning der Datenbanken und Investition in mehr Hardware. Neuere Entwicklungen (z.B. [FJ+09]) sind In-Memory-Lösungen, Kompression und spaltenorientierte Speicherung. Im Bereich spaltenorientierter Datenbanken existieren Forschungsprototypen, z.B. C-Store [St05] oder MonetDB [BKM08], kommerzielle Systeme, z.B. SybaseIQ [Sy11] oder Exasol [Ex11] und mit Infobright auch eine Open-Source-Lösung in Form einer Engine für MySQL [DO08, CM09]. Die Firma Infobright [In11a,In11b] wurde 2005 gegründet und hat ursprünglich eine rein kommerzielle Lösung angeboten, 2008 wurde die erste Open-Source-Version auf den Markt gebracht. Seitdem existieren sowohl eine kommerzielle (Infobright Enterprise Edition) als auch eine Open Source Variante (Infobright Community Edition).
Der Rest des Beitrags ist folgendermaßen gegliedert: Im folgenden Abschnitt wird das Prinzip der spaltenorientierten Speicherung skizziert und dargelegt, warum diese Speicherungsform hohe Kompressionsraten erwarten lässt. Abschnitt 3 geht dann speziell auf Infobright ein, beschreibt über die spaltenorientierte Speicherung hinaus gehende Konzepte des Systems. In Abschnitt 4 stellen wir unsere Vorgehensweise zur Evaluation spaltenorientierter Systeme vor und nennen mit Infobright erzielte Resultate. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick.

2 Spaltenorientierung und Kompression

Während Datenbanken traditionell Datensätze zeilenweise speichern [Sc04], haben sich in den letzten Jahren einige spaltenorientierte Systeme am Markt etabliert (z.B.[Sa09]). Zeilen- und spaltenorientierte Speicherung sind in Abbildung 1 gegenübergestellt: Tabelle FTable ist dabei ein charakteristischer Vertreter einer Fakttabelle in einem DWH. Sie besitzt einen zusammengesetzten Primärschlüssel, drei ganzzahlige Attribute und drei Attribute mit Aufzählungsdatentyp.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

ID1	ID2	ID3	IntA_1	IntA_2	IntA_3	EnumA_1	EnumA_2	EnumA_3
1000	1001	1001	1	1	5	Rot	Ja	Foto
1000	1001	1002	1	2	5	Rot	Ja	Foto
1000	1001	1003	2	3	5	Gelb	Ja	Kleidung
1000	1002	1001	0	1	5	Grün	Ja	Kleidung
1000	1002	1002	0	2	5	Rot	Nein	Frucht
1000	1002	1003	2	3	5	Rot	Ja	Frucht
1000	1003	1001	0	1	5	Rot	Ja	Tiefkühlwaren
1000	1003	1002	0	2	5	Grün	Ja	Foto

Zeilenorientierte Speicherung

Block 1

```
1000|1001|1001|1|1|5|Rot|Foto#
1000|1001|1002|1|2|5|Rot|Foto#
1000|1001|1003|2|3|5|Gelb|Kleidung#
1000|1002|1001|0|1|5|Grün|Kleidung#
```

Block 2

```
1000|1002|1002|0|2|5|Rot|Frucht#
1000|1002|1003|2|3|5|Rot|Frucht#
1000|1003|1001|0|1|5|Rot|Tiefkühlwaren#
1000|1003|1002|0|2|5|Grün|Foto#
```

Spaltenorientierte Speicherung

Block 1

```
1000|1000|1000|1000|1000|1000|1000|100
0#1001|1001|1001|1002|1002|1002|1003|1
003#1001|1002|1003|1001|1002|1003|1001
|1002#1|1|2|0|0|2|0|0#
```

Block 2

```
1|2|3|1|2|3|1|2#5|5|5|5|5|5|5|5#Rot|Ro
t|Gelb|Grün|Rot|Rot|Rot|Grün#Foto|Foto
|Kleidung|Kleidung|Frucht|Frucht|Tiefk
ühlwaren|Foto#
```

Abbildung 1: Zeilen- und spaltenorientierte Speicherung

Spaltenorientierung alleine spart zunächst keinen Speicherplatz, aber zum einen kann sich die Anzahl zu lesender Seiten bei breiten Tabellen reduzieren, weil nur die Seiten mit den tatsächlich angefragten Daten geladen werden müssen. Zum anderen lassen sich die spaltenbasierten Speicherungen sehr gut komprimieren, da stets Daten gleichen Typs nebeneinander stehen.

Zentraler Kompressionsalgorithmus ist dabei die in Abbildung 2 skizzierte Wörterbuchkompression: In einem Wörterbuch wird für jede tatsächliche Ausprägung ein Eintrag und eine Kodierung vorgenommen. Anstelle des tatsächlichen Wertes wird dann die Kodierung gespeichert.

Unkomprimierte Speicherung

Daten

```
Kleidung|Kleidung|Frucht|Foto|
Foto|Kleidung|Frucht|Tiefkühlwaren|
Tiefkühlwaren|Frucht|Frucht|Frucht|
Kleidung|Foto|Tiefkühlwaren|Tiefküh
lwaren|Tiefkühlwaren|
```

Wörterbuchkomprimierte Speicherung

Wörterbuch

Kleidung	00
Frucht	01
Tiefkühlwaren	10
Foto	11

Komprimierte Daten

```
00|00|01|11|11|00|01|10|10|01|01|01
|00|11|10|10|10|
```

Abbildung 2: Wörterbuchkompression

ISBN 978-3-88579-286-4

Unter der Annahme jede Zeichenkette habe eine Länge von 10 Byte und es liege eine bitgenaue Kompression wie in Abbildung 2 vor, ließen sich damit bei bis zu 16 unterschiedlichen Werten für das Attribut Produktgruppe ein Kompressionsfaktor von 20 erreichen (4 Bit statt durchschnittlich 10 Byte pro gespeichertem Wert).

3 Architektur von Infobright

In diesem Abschnitt wird die Architektur von Infobright dargestellt und beschrieben welche weiteren Konzepte neben der spaltenorientierten Speicherung im System realisiert sind.

Zur Speicherung unterteilt Infobright die Daten innerhalb einer Spalte in Einheiten zu 65536 Werten, die als Datenpakete bezeichnet werden. Für jedes einzelne Datenpaket wird abhängig von Datentyp und –verteilung individuell entschieden, welcher Kompressionsalgorithmus verwendet wird. Damit lassen sich u.U. (bei hoher Lokalität der Daten) bessere Kompressionsraten erzielen und auch bei lesenden Anfragen (siehe weiter unten) wird ein Vorteil erzielt. Neben der komprimierten Speicherung der Daten besitzt Infobright ein sog. Knowledge Grid, in dem verschiedene Metainformationen über die Daten abgelegt werden. Die einzelnen Komponenten des Knowledge Grid heißen Knoten. Diese können statischen Charakter haben (d.h. sie werden beim Ladevorgang einmal erstellt und sind dann stabil) oder einer gewissen Dynamik unterliegen, z.B. sich aufgrund von Anfrageprofilen im laufenden Betrieb ändern. Ein spezieller Typ statischer Knoten sind die sog. Datenpaketknoten (DPN, Data Pack Node). Sie enthalten Metainformationen, die sich auf ein einzelnes Datenpaket beziehen, z.B. werden für numerische Daten minimaler und maximaler Wert, Summe, Anzahl der von NULL verschiedenen Werte und die Gesamtzahl von Werten gespeichert. Abbildung 3 zeigt die Speicherung zweier Spalten mit ganzzahligen Attributwerten aus dem Intervall [1,10] mit 1.000.000 Einträgen.

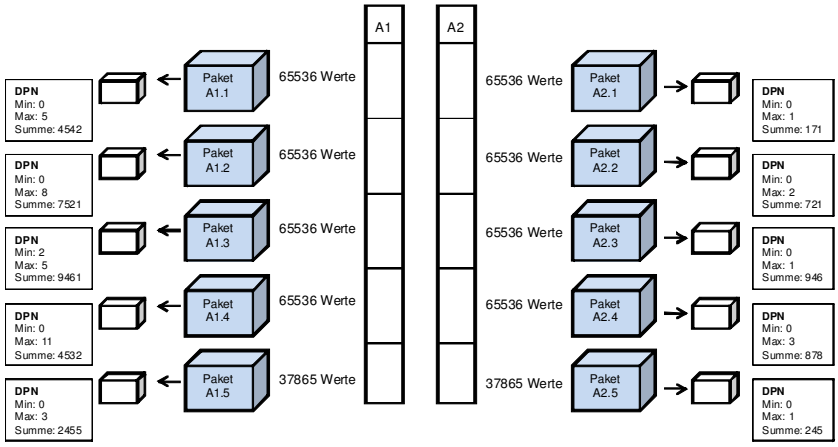

Abbildung 3: Spalten-Speicherung in Infobright

ISBN 978-3-88579-286-4

Genutzt werden die im Knowledge Grid abgelegten Informationen vom Optimierer, der für eine Anfrage die minimale Anzahl zu lesender (und zu dekomprimierender) Datenpakete ermittelt. Beispielsweise kann der Optimierer zu einfachen Anfragen, wie z.B.

```
SELECT MAX(T.A1) FROM T;
```

oder

```
SELECT SUM(T.A1) FROM T;
```

das Resultat ausschließlich aus den in den DPNs gespeicherten Metainformationen berechnen.

Auch bei komplexeren Anfragen kann der Optimierer das Knowledge Grid nutzen, um die Anzahl zu lesender (und zu dekomprimierender) Datenpakete gering zu halten, wie das Beispiel in Abbildung 4 zeigt.

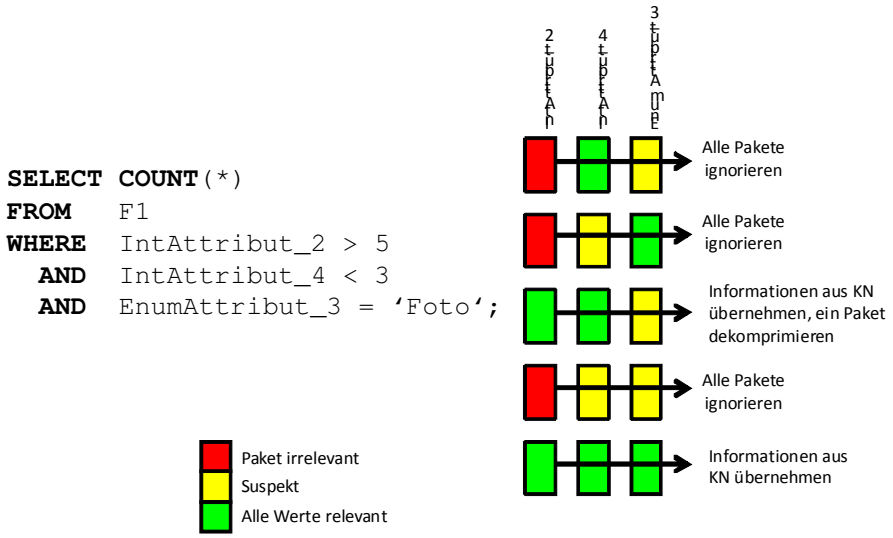

Abbildung 4: Komplexe Anfrage

Es wird zunächst jedem Paket durch Nachschauen in den DPN einer der drei Status „Irrelevant“, „Suspekt“ oder „Alle Werte relevant“ zugeordnet. Dabei kann aufgrund von AND-Verknüpfungen in der WHERE-Klausel die Anzahl zu lesender Datenpakete reduziert werden. Das Beispiel in Abbildung 4 zeigt eine Anfrage, in der drei Attribute logisch AND verknüpft werden. Zu jeder Spalte existieren fünf DPs mit zugehörigen DPNs, d.h. potenziell wären 15 Pakete zu lesen. Sobald jedoch in einer Zeile ein Paket

aufgrund der Informationen im zugehörigen DPN als irrelevant erkannt wird, können auch die zugehörigen Pakete der anderen Attribute ignoriert werden (Zeile 1,2 und 4 in der Abbildung). Sind alle korrespondierenden Pakete relevant (Zeile 5 in der Abbildung), können die entsprechenden Informationen (im Beispiel die Summe) aus den DPNs übernommen werden. Nur bei suspekten Paketen (im Beispiel in Zeile 3) ist ein Lesen und Dekomprimieren der Daten notwendig. Insgesamt wird somit im Beispiel die Anzahl zu lesender und dekomprimierender Pakete damit von 15 auf 1 reduziert. Dieser Effekt reduziert den I/O und verbessert damit die Lesegeschwindigkeit.

4 Evaluation

In diesem Abschnitt soll anhand von Messungen untersucht werden, inwieweit die in den vorangegangenen Abschnitten beschriebenen Konzepte einen Vorteil bieten. Relevante Kriterien sind hierbei Ladezeiten, Speicherplatzverbrauch und Antwortzeitverhalten. Es wird jeweils der relative Unterschied zwischen der MySQL-Standard-Engine InnoDB und Infobright betrachtet.

Untersuchungsobjekt ist die in Abbildung 5 skizzierte Fakttabelle, die auch ín [He10] zur Untersuchung der Kompressionsfähigkeit der Oracle-Datenbank verwendet wurde.

Tabelle F1		
Attribut	**Typ**	**Wertebereich**
Dim1_ID	**NUMBER(10)**	**Zusammengesetzter Primärschlüssel**
Dim2_ID	**NUMBER(10)**	
Dim3_ID	**NUMBER(10)**	
IntAttribut_1	NUMBER(10)	0-2
IntAttribut_2	NUMBER(10)	1-3
IntAttribut_3	NUMBER(10)	1-10
EnumAttribut_1	VARCHAR2	red, yellow, green
EnumAttribut_2	VARCHAR2	true, false
EnumAttribut_3	VARCHAR2	cheese, cloth, toys, sale, foto, cosmetics, furniture, fruit, meat, frozen
RealAttribut_1	NUMBER(10,2)	0,0 – 2,0
RealAttribut_2	NUMBER(10,2)	0,0 – 2,0
RealAttribut_3	NUMBER(10,2)	0,0 – 2,0
RealAttribut_4	NUMBER(10,2)	1,0 – 10,0
RealAttribut_5	NUMBER(10,2)	1,0 – 10,0

Abbildung 5: Fakttabellen

Fakttabellen in einem Data Warehouse bieten sich für Kompression besonders an, denn sie sind zum einen „groß“, so dass sich Kompression lohnt und zum anderen besitzen sie häufig die Eigenschaft wiederkehrender Werte innerhalb einer Spalte. Fakttabellen besitzen typischerweise wie die Tabelle F1 einen zusammengesetzten Primärschlüssel und eine Reihe von Attributen der Typen Integer, Real und Aufzählungstypen mit kleinem Wertebereich. Für die Tabelle F1 und Varianten, die in unterschiedlicher Anzahl der Faktattribute der verschiedenen Datentypen bestanden, haben wir Routinen zur synthetischen Generierung von Daten geschrieben. Auch für diese Testdaten wurden Varianten gewählt, die z.B. in unterschiedlichen Verteilungen der Attributwerte über die Wertebereiche oder einer Vorsortierung der Daten nach einem bestimmten Wert bestanden.

Die generierten Daten wurden in Form von CSV-Dateien als Bulk Load in das DBMS geladen. Bei diesem Vorgang wurden die Zeiten protokolliert, die von MySQL während des Ladens gemessen wurden. Alle Messungen wurden jeweils für verschiedene Mengengerüste zwischen 100.000 und 1.000.000 Datensätzen durchgeführt, jede einzelne Messung mehrfach wiederholt und aus diesen Messungen der Durchschnittswert gebildet.

Die Messungen in [Ha11] werden vergleichend zwischen der MySQL-Standard-Engine InnoDB und Infobright durchgeführt. Dabei wurden zusammenfassend folgende Resultate erzielt:

- Die Ladezeit ließ sich ca. um den Faktor 2 senken.
- Der verbrauchte Speicherplatz ließ sich in den meisten Fällen um den Faktor 10 bis 20 senken, extreme Fälle wiesen kleinere oder größere Faktoren auf. Diese Schwankungen lassen sich durch Eigenschaften der Daten (Verteilung, Distinktheit, Sortierung) erklären.
- Bei Bereichsanfragen, in denen das Knowledge Grid nicht genutzt werden kann, sind die Antwortzeiten von Infobright deutlich schlechter als die von InnoDB.
- Bei komplexen, analytischen Anfragen erzielte Infobright etwas bessere Antwortzeiten als InnoDB, allerdings nicht in dem von Infobright angegebenen Maße [In11a].
- Auch wurden die Grenzen des Einsatzes von Infobright deutlich: Zum einen wurden bei Anfragen mit kleinem Resultat schlechtere Ergebnisse erzielt, zum anderen sind bei Infobright keine Aktualisierungsoperationen möglich.

ISBN 978-3-88579-286-4

5 Zusammenfassung und Ausblick

In diesem Beitrag haben wird zunächst das Problemfeld wachsender Datenmengen in Data-Warehouse-Umgebungen charakterisiert. Weiterhin ist in den letzten Jahren ein Trend zum Einsatz von Open-Source-Komponenten im Data-Warehouse-Umfeld zu beobachten. Dabei kommt als Datenbank häufig MySQL zum Einsatz, die hierbei mitgelieferte Standard Engine InnoDB ist auf klassische, zeilenorientierte Speicherung und transaktionale Verarbeitung ausgerichtet. Die Engine Infobright bietet eine Alternative für analytische Umgebungen, weil sie spaltenorientiert speichert und Kompressionstechniken einsetzt.

Wir haben unsere Vorgehensweise zum Evaluieren spaltenorientierter Datenbanken vorgestellt. Dabei werden verschiedene Aspekte, wie z.B. Ladezeiten, verbrauchter Speicherplatz und Antwortzeiten für ausgewählte Anfragen berücksichtigt. Ein Generator erzeugt für typische Fakttabellen Daten, die als erster Test beim Evaluieren eines neuen Systems verwendet werden können. Anschließend ist jedoch immer noch ein Test mit eigenen Daten notwendig, weil die Struktur von Daten und Anfragen erheblichen Einfluss auf Kompressionsfaktor und Antwortzeiten haben.

In der Zukunft sollen weitere Untersuchungen spaltenorientierter Datenbanken durchgeführt werden, dabei sollen als Ziel Richtlinien erstellt werden, um aufgrund der Beschaffenheit der Daten Vorhersagen über den zu erwartenden Vorteil zu erhalten.

Literaturverzeichnis

[BG08] Bauer, A.; Günzel, H.: Data-Warehouse-Systeme: Architektur, Entwicklung, AnwendDaten ung. dpunkt-Verlag, 2008.

[BKM08] Boncz, Peter A.; Kersten, Martin L.; Manegold, Stefan: Breaking the memory wall in MonetDB. CACM 51(12):77-85, 2008.

[CM09] Cabral, S.K.; Murphy, K.: MySQL Administrator's Bible. John Wiley & Sons, 2009.

[DO08] Dyer, Russell; Oram, Andy: MySQL in a Nutshell. O'Reilly Media, 2008.

[Ex11] http://www.exasol.com/, Letzter Abruf am 3.5.2011.

[FJ+09] Färber, F.; Jäcksch, B.; Lemke, C.; Große, P.; Lehner, W.: Hybride Datenbankarchitekturen am Beispiel der neuen SAP In-Memory-Technologie. Datenbank-Spektrum, 10(2): 81-92, 2010.

[Ha11] Haller, T.: Ein Verfahren zur Evaluation spaltenorientierter Datenbanken. Studienarbeit, DHBW Campus Horb, 2011.

[HP05] Herden, O.; Pfrommer, B.: Aufbau eines Data Warehouse mit den Open-Source-Komponenten JPivot, Mondrian und MySQL - Alternative zu kommerziellen Produkten? Proceedings GI-Jahrestagung, Bonn, 2005; S. 96-100.

[He10] Herden, O.: Hybride Datenbankarchitekturen am Beispiel der neuen SAP In-Memory-Technologie. Proceedings DOAG-Konferenz 2010, Nürnberg, 2010.

[HT+10] Haneke, U.; Trahasch, S.; Hagen, T.; Lauer, T.: Open Source Business Intelligence (OSBI): Möglichkeiten, Chancen und Risiken quelloffener BI-Lösungen, Hanser-Verlag, 2010.

[In11a] http://www.infobright.org/, Letzter Abruf am 3.5.2011.

[In11b] http://www.infobright.com/, Letzter Abruf am 3.5.2011.

[Ki96] Kimball, R.: The Data Warehouse Toolkit. John Wiley & Sons, Inc., 1996.

[Sa09] Sattler, K.-U.: Column Stores. Datenbank-Spektrum, 30/2009, S. 39-40.

[Sc04] Schneider, M.: Implementierungskonzepte für Datenbanksysteme, Springer-Verlag, 2004.

[St05] Stonebraker, M. u.a.: C-Store: A Column-oriented DBMS. Proceedings of the 31st VLDB Conference, Trondheim (Norwegen), 2005; S. 553-564.

[Sy11] http://www.sybase.de/, Letzter Abruf am 3.5.2011.