

MySQL Engine Infobright: Speicherplatz sparen und schnellere Anfragen

Olaf Herden

Duale Hochschule Baden-Württemberg Campus Horb

Schlüsselworte

MySQL, Engine, Infobright, Data Warehouse, Spaltenorientierte Speicherung, Kompression.

Abstract

Starkes Wachstum von Daten im Data-Warehouse-Umfeld verlangt nach neuen Datenbankkonzepten. Eines hiervon ist die spaltenorientierte Speicherung in Verbindung mit Kompression von Daten, was von der MySQL-Engine 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. Typische Anwendungsgebiete sind hierbei Verkaufsanalysen oder andere Daten, die aus operativen Vorsystemen ins DWH transportiert werden. In der jüngeren Vergangenheit spielen maschinell generierte Daten eine wichtige Rolle. Diese werden z.B. von RFID(Radio Frequency Identification)-Sensoren, Netzwerkkomponenten, Web-Servern oder Terminals erzeugt. Weil diese Daten häufig ein Volumen erreichen, das das klassischer DWHs um ein Vielfaches übersteigt, spricht man in diesem Falle von Big Data.

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 oder Materialisierungen verlangsamen sich auch die Ladezeiten wegen deren Pflege.

Die populärsten, klassischen Gegenmaßnahmen sind Tuning der Datenbanken und Investition in mehr Hardware. Als Tuningmaßnahmen stellen Datenbanken Indexstrukturen, Materialisierungen und Partitionierungen zur Verfügung. Diese müssen jedoch geplant und ausgeführt werden, was administrativen Overhead bedeutet. Ebenso führt z.B. die Aktualisierung von Indexen zu Zeitverlusten bei Ladeoperationen. Als Hardwareerweiterungen kommen mehr Hauptspeicher und Prozessorleistung für den Server ebenso in Frage wie performantere Speichersysteme oder eine verbesserte Netzwerk-Infrastruktur.

Neuere Entwicklungen (z.B. [FJ+09]) sind In-Memory-Lösungen, Kompression und spaltenorientierte Speicherung. Im Bereich spalten-orientierter 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 werden einige wichtige Prinzipien von Infobright vorgestellt, Abschnitt 3 skizziert das Konzept der spaltenorientierten Speicherung und legt dar, warum diese Speicherungsform hohe Kompressionsraten erwarten lässt. Abschnitt 4 geht dann speziell auf Infobright ein, beschreibt über die spaltenorientierte Speicherung hinaus gehende Konzepte des Systems. In Abschnitt 5 werden die Resultate verschiedener Untersuchungen von Infobright vorgestellt [Ha11, HeHa11]. Abschließend erfolgt eine Zusammenfassung und ein Ausblick.

2 Infobright: Prinzipien

In diesem Abschnitt sollen die wesentlichen Aspekte von Infobright genannt werden:

Integration in MySQL

Das System ist als Engine für MySQL konzipiert, dadurch können wesentliche Komponenten von MySQL (z.B. Caches) verwendet werden und hierfür ist keine aufwändige Re-Implementierung notwendig. Ebenso können Zusatzwerkzeuge wie die MySQL-Workbench genutzt werden. Die Architektur ist in Abbildung 1 skizziert, wobei die hinterlegten Teile im rechten Bereich Infobright-spezifisch sind, die restlichen dem MySQL-Standard entstammen.

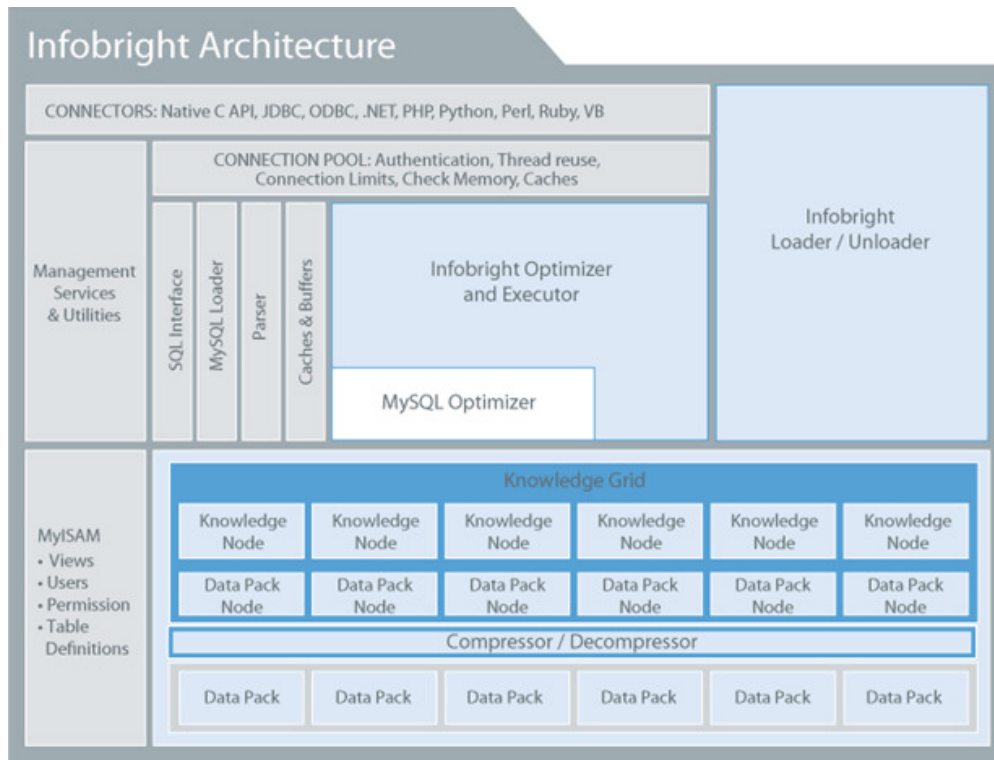


Abb. 1: Infobright Architektur [Info10]

Breite Verfügbarkeit

Infobright ist für verschiedene Linux- (u.a. Red Hat, Debian und Ubuntu) und Windows-Derivate (Windows XP und Windows Server 2003) verfügbar, nach Herstellerangaben liegt eine Optimierung für die Linux-Versionen vor, so dass diese für High-End-Lösungen vorzuziehen sind.

Für beide Plattformen existiert jeweils eine freie Open Source Community Edition und (ICE, Infobright Community Edition) und eine kostenpflichtige Closed Source Variante (IEE, Infobright Enterprise Edition). Die ICE steht unter der GPL (Gnu Public Licence). In den (auch im weiter Verlauf dieses Beitrags beschriebenen) Kernkonzepten der Spaltenorientierung, Kompression und der Knowledge Komponenten bestehen zwischen ICE und IEE keine Unterschiede, wesentliche Ergänzungen der IEE sind neben organisatorischen Aspekten Support und Garantie auf der technischen Seite z.B. die Möglichkeit konkurrierenden Ladens und Abfragens sowie schnellere Ladeoptionen. Eine detaillierte Auflistung der Unterschiede beider Editionen ist unter [In11a] zu finden.

Einfache Hardware

Infobright ist dafür konzipiert auf Standard-Hardware zu laufen. So wird für die Evaluations- und Entwicklungszwecke als Mindestvoraussetzungen ein 2 GHz-Prozessor 2 bis 4 GB Hauptspeicher genannt, für Produktivumgebungen mit Mehrbenutzerbetrieb werden mindestens 16 GB Hauptspeicher empfohlen.

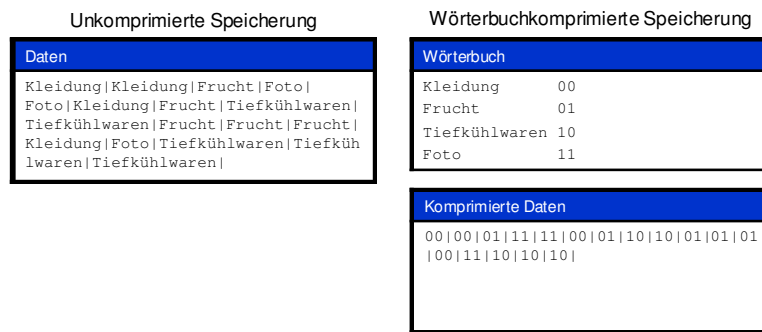


Abb. 3: Wörterbuchkompression

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).

4 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 4 zeigt die Speicherung zweier Spalten mit ganzzahligen Attributwerten aus dem Intervall [1,10] mit 1.000.000 Einträgen.

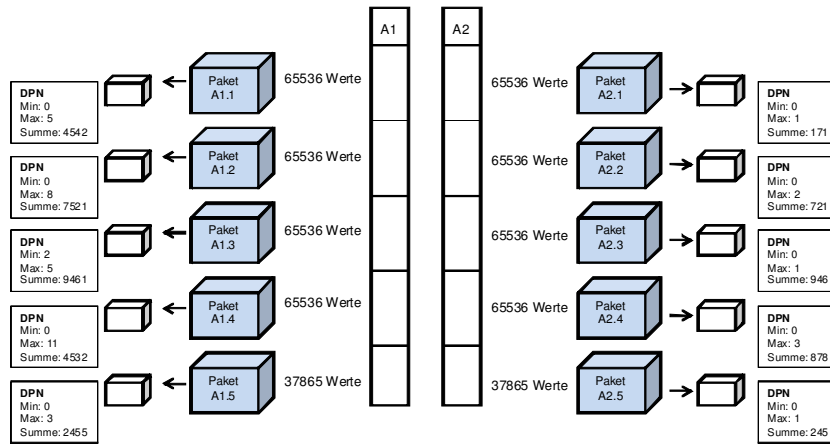


Abb. 4: Spalten-Speicherung in Infobright

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 5 zeigt.

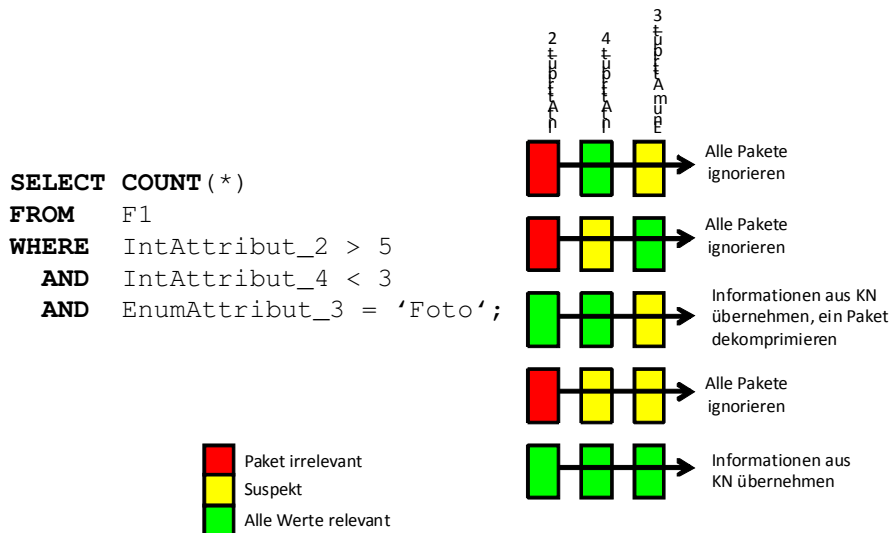


Abb. 5: Komplexe Anfrage

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 5 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.

5 Untersuchungen

Für den ersten Evaluationsblock dient die in Abbildung 6 skizzierte Tabelle F_Karte, die Datensätze beinhaltet, die von Terminals mit EC-Kartenzahlung geliefert werden.

Tabelle F_Karte	
Attribut	Typ
Version	INTEGER
TerminalID	INTEGER
BelegKopfzeile1	VARCHAR(50)
BelegKopfzeile2	VARCHAR(50)
BelegKopfzeile3	VARCHAR(50)
BelegKopfzeile4	VARCHAR(50)
BelegKopfzeile5	VARCHAR(50)
Fehlercode	VARCHAR(5)
Fehlertext	VARCHAR(50)
Online	VARCHAR(5)
TraceNr	INTEGER
BelegNr	INTEGER
Verfall	VARCHAR(6)
Uhrzeit	DATE
Datum	DATE
Betrag	INTEGER
Valuta	INTEGER
Kartenart	VARCHAR(20)
Verwendung	VARCHAR(20)
KontoNr	VARCHAR(10)
BLZ	VARCHAR(8)
Folgenummer	INTEGER

Abb.6: Tabelle F_Karte

Dabei wurden Testdatensätze mithilfe eines selbstgeschriebenen Java-Programms generiert. Für verschiedene Mengengerüste wurden zunächst die Ladezeiten und der verbrauchte Speicherplatz gemessen, sowohl für die Infobright-Engine als auch für die MyISAM-Engine. Die Resultate sind in Abbildung 7 dargestellt: Infobright erreicht hier einen Kompressionsfaktor von ca. 8. Der erreichte Kompressionsfaktor ist in Abbildung 7 einmal im direkten Vergleich zur MyISAM-Engine dargestellt und einmal ist der von Infobright angegebene Wert (berechnet im Verhältnis zum RAW-Volumen) angegeben. Bezahlt wird dieses mit einer erhöhten Ladegeschwindigkeit. Diese erklärt sich durch die

für das Komprimieren und Aufbauen des DPN benötigten Zeit. Dabei ist zu beobachten, dass die Differenz in der Ladegeschwindigkeit mit steigendem Volumen geringer wird, was vor allem daran liegt, dass MyISAM in diesem Falle nicht mehr linear skaliert.

Datensätze	Ladezeit			Speicherplatz			Tabelle F4	
	Infobright	MyISAM	Faktor	Infobright	MyISAM	Faktor	RAW [MB]	Faktor
100.000	4,8	0,5	9,6	1.928.007	16.362.296	8,5	15	7,966
200.000	4,8	1,0	4,8	3.932.454	32.726.920	8,3	30	8,030
500.000	11,0	2,4	4,6	9.824.702	81.817.912	8,3	75	8,035
1.000.000	22,7	4,9	4,6	19.638.063	163.640.252	8,3	150	8,040
2.000.000	43,5	9,5	4,6	39.266.595	327.276.472	8,3	301	8,042
5.000.000	110,8	56,0	2,0	98.161.773	818.192.564	8,3	752	8,042
10.000.000	227,5	63,6	3,6	196.322.049	1.636.387.144	8,3	1505	8,042
20.000.000	477,1	192,2	2,5	392.634.251	3.272.392.812	8,3	3011	8,042
50.000.000	1158,2	515,4	2,2	981.587.675	8.181.607.408	8,3	7528	8,042

Abb.7: Infobright vs. MyISAM: Ladezeiten und Kompression

Abbildung 8 zeigt eine Reihe exemplarischer Abfragen an die Tabelle F_Karte und die Antwortzeiten in Infobright und MyISAM. Dabei wird bei vielen Anfragen ein sehr deutlicher Geschwindigkeitsvorteil von Infobright deutlich. Ursache ist jeweils die Tatsache, dass die meisten Abfragen mithilfe des Knowledge Grid beantwortet werden können und die eigentlichen Daten physisch nicht gelesen werden müssen.

Nummer	SQL-Anweisung
Q1	SELECT AVG(betrag) FROM f_karte;
Q2	SELECT verfall, COUNT(*) FROM f_karte GROUP BY verfall;
Q3	SELECT COUNT(*) FROM f_karte WHERE betrag>5000;
Q4	SELECT COUNT(*) FROM f_karte WHERE betrag<1000;
Q5	SELECT * FROM f_karte WHERE TerminalID = '99999509';
Q6	SELECT * FROM f_karte WHERE uhrzeit = '14:15:30' AND datum = '2011-10-01';
Q7	SELECT * FROM f_karte;

	Infobright						
	Q1	Q2	Q3	Q4	Q5	Q6	Q7
2.000.000	0,0	1,0	0,2	0,1	0,1	0,0	0,4
5.000.000	0,0	2,6	0,6	0,3	0,3	0,0	0,3
10.000.000	0,0	5,0	1,2	0,6	0,5	0,0	0,3
20.000.000	0,0	11,3	3,3	1,1	62,9	0,0	0,2
50.000.000	0,0	36,7	11,3	2,7	78,0	0,0	0,2

	MyISAM						
	Q1	Q2	Q3	Q4	Q5	Q6	Q7
2.000.000	1,3	1,9	1,2	1,2	0,2	1,2	0,0
5.000.000	3,4	5,0	3,0	2,9	0,2	3,0	0,0
10.000.000	33,3	9,9	6,0	5,9	0,2	5,9	0,0
20.000.000	63,1	70,1	75,7	63,6	0,9	42,1	0,0
50.000.000	173,9	165,0	167,9	169,8	0,9	140,1	0,1

Abb.8: Infobright vs. MyISAM: Abfragezeiten

Für die weiteren Untersuchungen fand die in Abbildung 9 skizzierte Tabelle F4 (mit entsprechend beschriebenen Variationen) Anwendung.

Tabelle F4		
Attribut	Typ	Werte in F4
Dim1_ID	NUMBER(10)	Zusammengesetzter Primärschlüssel
Dim2_ID	NUMBER(10)	
Dim3_ID	NUMBER(10)	
IntAttribut_1	NUMBER(10)	100001-100010
IntAttribut_2	NUMBER(10)	100001-100010
IntAttribut_3	NUMBER(10)	100001-100010
IntAttribut_4	NUMBER(10)	100001-100010
IntAttribut_5	NUMBER(10)	100001-100010
IntAttribut_6	NUMBER(10)	100001-100010
IntAttribut_7	NUMBER(10)	100001-100010
IntAttribut_8	NUMBER(10)	100001-100010
IntAttribut_9	NUMBER(10)	100001-100010
IntAttribut_10	NUMBER(10)	100001-100010

Abb. 9: Tabelle F4

Zunächst wurde die Anzahl unterschiedlicher Werte innerhalb einer Spalte untersucht. Dabei wurden für die Spalten Int_A_1 bis Int_A_10 jeweils n unterschiedliche Werte generiert. Das in Abbildung 10 dargestellte Resultat zeigt einen starken Einfluss der Distinktheit der Werte auf die Kompressionsrate. Bei nur zwei unterschiedlichen Werten wird ein Kompressionsfaktor von 36 erreicht, dieser fällt dann mit steigender Anzahl verschiedener Attribute, zunächst stark (so ist bei 5 verschiedenen Werten der KF schon nur noch 17 und bei 10 unterschiedlichen Werten 12), dann mit steigender Distinktheit weniger.

Anzahl unterschiedlicher Wert pro Spalte	Tabelle F4	
	Volumen	Faktor
2	14.585.963	36,764
3	23.781.491	22,549
4	28.962.876	18,515
5	31.226.782	17,173
7	39.232.581	13,668
10	43.644.058	12,287
15	52.910.878	10,135
20	56.146.307	9,551
50	72.675.515	7,379
100	87.024.080	6,162
200	99.512.089	5,388
500	116.046.409	4,621

Jeweils 10.000.000 Datensätze, Rohvolumen 511MB.

Abb.10: Distinktheit und Kompressionsfaktor

Es ist an dieser Stelle anzumerken, dass die Distinktheit nur bei Werten innerhalb einer Spalte Bedeutung hat. Messungen mit gleichen und unterschiedlichen Wertebereichen verschiedener Attribute haben ergeben, dass diese keinen Einfluss auf den erreichten Kompressionsfaktor haben.

Es gilt der Zusammenhang, dass mit steigender Anzahl NULL-Werte der Kompressionsfaktor steigt. Allerdings muss der prozentuale Anteil an NULL-Werten schon 50% übersteigen, damit das Ansteigen des Kompressionsfaktors signifikant ist. Abbildung 11 zeigt die Resultate einer Untersuchung mit Tabelle F4. Es wurden jeweils 10.000.000 Datensätze generiert, dabei der prozentuale Anteil an den Faktattributen Int_A_1 bis Int_A_10 zwischen 0 und 100% variiert.

Prozentualer Anzahl NULL-Werte Wert pro Spalte	Tabelle F4	
	Volumen	Faktor
0	47.197.126	11,362
10	47.391.786	11,315
20	46.380.447	11,562
30	44.260.698	12,116
40	41.182.184	13,021
50	37.295.852	14,378
60	32.819.788	16,339
70	27.536.022	19,474
80	21.358.955	25,107
90	13.949.670	38,442
100	3.906.678	137,265

Jeweils 10.000.000 Datensätze, pro Spalte 10 unterschiedliche Werte, Rohvolumen 511MB.

Abb. 11: NULL-Werte und Kompressionsfaktor

Abschließend soll untersucht werden, welchen Einfluss unterschiedliche Datentypen auf die Kompressionsrate haben. Dazu wurde in Tabelle F4 anstelle der zehn verschiedenen Integer-Werte, einmal zehn verschiedene Datumswerte und einmal zehn verschiedene Zeichenketten (im Durchschnitt 7 Zeichen lang) verwendet. Das Ergebnis ist in Abbildung 12 dargestellt: Date-Attribute werden am besten komprimiert, gefolgt von Integer und String.

Datentyp	Tabelle F4		
	Volumen	RAW	Faktor
Integer	47.197.126	511	11,362
Date	45.634.258	1083	24,899
String	72.561.053	663	9,595

Jeweils 10.000.000 Datensätze, pro Spalte 10 unterschiedliche Werte.

Abb. 12: Datentyp und Kompressionsfaktor

6 Zusammenfassung und Ausblick

Durch das wachsende Volumen in DWHs kommen relationale Datenbanken zunehmend an ihre Grenzen. Ein Konzept diesem entgegen zu wirken, ist die spaltenorientierte Speicherung in Verbindung mit Kompression. Dieses Konzept wird u.a. von der MySQL- Engine Infobright angeboten. In diesem Beitrag wurden wesentliche Prinzipien, Konzepte und die Architektur von Infobright beschrieben. Anschließend erfolgten einige Untersuchungen, die Infobright evaluiert und mit der Engine MyISAM verglichen haben. Dabei wurden die Vorteile und Grenzen von Infobright herausgestellt.

In der Zukunft sollen weitere Untersuchungen mit anderen Daten und auch der Vergleich mit anderen spaltenorientierten Datenbanken durchgeführt werden. Ziel dieser Untersuchungen sollen Richtlinien sein, die aufgrund der Beschaffenheit der Daten Vorhersagen über den zu erwartenden Vorteil erlauben.

Kontaktadresse

Prof. Dr.-Ing. Olaf Herden
Duale Hochschule BW Campus Horb
Studiengang Informatik
Florianstr. 15
D-72160 Horb

Telefon: +49 (0) 7451-521 146
Fax: +49 (0) 7451-521 101
E-Mail: o.herden@hb.dhbw-stuttgart.de
Internet: www.dhbw-stuttgart.de/horb

Referenzen

- [BG08] Bauer, A.; Günzel, H.: Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung. 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. DB-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.
- [HeHa11] Herden, O.; Haller, T.: Das spaltenorientierte MySQL-Plugin Infobright als Kern einer Open Source basierten Data-Warehouse-Infrastruktur. Proceedings GI-Jahrestagung, Berlin, 2011.
- [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.
- [Salo06] Salomon, D. Data Compression: The Complete Reference. Springer, Berlin (2006)
- [Sa09] Sattler, K.-U.: Column Stores. Datenbank-Spektrum, 30/2009, S. 39-40.
- [Sayo05] Sayood, K. Introduction to Data Compression. Morgan Kaufmann, San Francisco (2005)
- [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.
- [ZiLe77] Ziv, J., Lempel, A.: A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory 23(3): 337-343 (1977)