

Hauptspeicherdatenbanken für Unternehmensanwendungen

Erschienen in: A. Johannsen; R. Franz; H. Heinrich (Hrsg.):
In-memory Computing – Die Rückkehr der Unternehmenssteuerung
in Echtzeit?

Tagungsband zum 11. Berlin-Brandenburger SAP-Forum der
Fachhochschule Brandenburg am 25.10.2012, S. 37-50,
Shaker Verlag, 2012

Impressum

Schriften des Instituts für Betriebliche Anwendungssysteme (IBAW)

Herausgeber: Prof. Dr. Andreas Johannsen, Fachhochschule Brandenburg

Die Schriftenreihe des IBAW enthält vornehmlich Vorab-Veröffentlichungen, spezialisierte Einzelergebnisse und ergänzende Materialien. Im Interesse einer späteren Veröffentlichung wird gebeten, die Schriften nicht weiter zu vervielfältigen. Die Autoren sind für kritische Hinweise dankbar.

Autoren: Jens Krüger, Andreas Johannsen

Kontakt: Fachhochschule Brandenburg
University of Applied Sciences
Magdeburger Str. 50
14770 Brandenburg an der Havel
T +49 3381 355 - 256
F +49 3381 355 - 199
E johannse@fh-brandenburg.de
www.fh-brandenburg.de

ISSN 2198-090X

Bezug: Die Schriften der Schriftenreihe des IBAW erscheinen unregelmäßig und sind kostenfrei

© Fachhochschule Brandenburg

Hauptspeicherdatenbanken für Unternehmensanwendungen

Jens Krüger, Hasso-Plattner-Institut an der Universität Potsdam

Andreas Johannsen, Fachhochschule Brandenburg

Einleitung

Die Datenhaltung in Unternehmensanwendungen hat sich in den letzten Jahrzehnten kaum geändert. Zum Einsatz kommen im Allgemeinen relationale Datenbanken, deren Architektur anhand der vor rund 20 Jahren definierten Charakteristiken für transaktionale Datenverarbeitung entworfen wurde. Im Laufe der Zeit haben sich die Anforderungen an die Unternehmensanwendungen geändert. Einerseits sind kürzere Prozesslaufzeiten in immer komplexeren Anwendungen notwendig, andererseits sollen möglichst aktuelle Daten für analytische Anfragen zur Entscheidungsunterstützung verwendet werden. Daneben sorgen in heutigen Unternehmensanwendungen automatisierte Prozesse für ein größeres Datenaufkommen, welches effizient – transaktional wie auch analytisch – verarbeitet werden muss.

Heutige Unternehmensanwendungen sind in ihrer Architektur mit der Zeit komplexer geworden, um den steigenden Anforderungen gerecht zu werden und die fehlenden Eigenschaften des Datenmanagementsystems auszugleichen. Beispiele hierfür sind die redundante Speicherung von Aggregaten, die Materialisierung vorberechneter Ergebnismengen in dedizierten Tabellen oder auch die Auslagerung von Prozessen in spezialisierte Systeme oder asynchrone Programme.

Während die meisten dieser Varianten redundante Ablage von Daten als Lösung nutzen, um die erforderlichen geringen Antwortzeiten für analytische Anfragen zu erreichen, wird dabei gleichzeitig durch die Notwendigkeit einer vordefinierten Materialisierungsstrategie die Fle-

xibilität des Systems verringert. Daneben führt diese höhere Komplexität zu einer Steigerung der Gesamtkosten des Systems.

In diesem Artikel wird untersucht, inwiefern Technologien in der Datenverarbeitung die Anforderungen und Datencharakteristiken moderner Unternehmensanwendungen optimal unterstützen können.

Fortschritte der Hardware

In den letzten Jahren wurde ein bis dato üblicher Trend unterbrochen. Der bis dahin kontinuierliche Anstieg der Taktrate von Prozessoren erreichte aufgrund der auftretenden Abwärme eine physikalische Grenze. Gleichzeitig konnte ein starker Anstieg der zur Verfügung stehenden Menge an Hauptspeicher beobachtet werden. Da die Rechenkapazität nicht länger durch eine Anhebung der Taktrate gesteigert werden konnte, wurde der Weg der Parallelisierung eingeschlagen. Mehrkernprozessoren sind heutzutage in Notebooks und sogar Mobiltelefonen zu finden. Die in Unternehmenshardware eingesetzten Server beinhalten inzwischen bis zu 96 Kerne und bis zu 2 TB Hauptspeicher. Weiterhin ist auch für die Zukunft davon auszugehen, dass das Kosten-pro-Performance-Verhältnis seinen rapiden Sinkflug beibehält. Diese veränderten Grundvoraussetzungen schaffen einerseits neue Potenziale für Unternehmensanwendungen, erfordern aber andererseits auch angepasste Herangehensweisen und Umsetzungen für die gestellten Aufgaben, um den Anforderungen der veränderten Hardwarearchitekturen gerecht zu werden. Im Folgenden wird zunächst die Entwicklung hinsichtlich der Datenverarbeitung in Unternehmen näher betrachtet.

Trends in der Datenverarbeitung

Verschiedene Entwicklungen im Bereich der Datenverarbeitung, die einzeln betrachtet nicht neu sind, ermöglichen durch ihre Kombination neue Anwendungsbereiche. Die vertikale Partitionierung (Spaltenorientierung) wurde beispielsweise schon Ende der 70er-Jahre betrachtet, ihre tatsächlichen Potenziale lassen sich jedoch erst mit der Zusammenführung aktueller Hardwaretrends effizient nutzen. In den letzten Jahren wurde deutlich, dass spezialisierte Datenbanken in einigen Anwendungsfeldern gegenüber so genannten „allgemeinen Datenbanken“ (engl. General Purpose Databases) im Vorteil sind. Damit wird implizit ein auf Unternehmensanwendungen spezialisiertes Datenmanagement motiviert, wobei das Verarbeiten von Daten rein im Hauptspeicher eine zentrale Rolle spielt. Zu erwähnen ist, dass dedizierte Datenbanken für transaktionale sowie analytische Verarbeitung das Konzept der Spezialisierung hinsichtlich der Anforderungen der konsumierenden Anwendung generell berücksichtigen. Dem entgegen steht der Fakt, dass moderne Unternehmensanwendungen einen großen Teil ihres Mehrwertes aus der latenzfreien Integration von Daten

und Prozessen beider Bereiche ziehen. Dem entsprechend ist ein auf Unternehmensanwendungen spezialisiertes Datenmanagement notwendig, um diese Anforderung effizient umzusetzen. Ausgehend von der konventionellen Trennung wurde, z. B. in [KRU10] gezeigt, dass eine Neuaufnahme der Anwendungscharakteristiken unter Berücksichtigung aktueller Trends notwendig ist, um genau dieses zu erreichen.

Zum einen hat die Entwicklung der Hardware in den letzten Jahren dazu geführt, dass Hauptspeicherdatenbanken, allein schon von der zur Verfügung stehenden Größe des Hauptspeichers, realisierbar wurden und somit in den Fokus der Forschung gelangten. Zum anderen bieten Hauptspeicherdatenbanken, welche die Daten spaltenorientiert speichern, die Möglichkeit, bestehende transaktionale und analytische Datenbanken wieder zu vereinen, da sie eine wesentlich höhere Geschwindigkeit im Datenzugriff bieten und somit analytische Anfragen auf Basis transaktionaler Daten im Sekundenbereich beantwortet werden können. Umgesetzt wird dies durch die Kombination verschiedener Technologien im Management von Daten bei gleichzeitiger Berücksichtigung unternehmensspezifischer Datencharakteristiken.

Neben Datenstrukturen, die für den Zugriff im Hauptspeicher optimiert sind, kommen verschiedene Kompressionsverfahren zum Einsatz, welche im Folgenden ebenfalls kurz erläutert werden.

Zeilen und Spaltenorientierung

Durch die steigende Nachfrage nach analytischen Funktionen wie Ad-hoc Anfragen auf transaktionalen Daten rückten im letzten Jahrzehnt spaltenorientierte Datenbanken – vor allem im Umfeld der OLAP-Anwendungen – in den Fokus der Wissenschaft.

Spaltenorientierte Datenbanken basieren auf einer vertikalen Partitionierung nach Spalten, welche Ende der 70er-Jahre zuerst betrachtet wurde und von Copeland und Khoshafian 1985 [COP85] weiterentwickelt wurde. Jede Spalte wird hierbei separat gespeichert, während das logische Tabellenschema durch die Einführung von eindeutigen Positionsschlüsseln beibehalten wird. Diese Verwendung der Positionsschlüsseln führt zu zusätzlichem Speicherbedarf, kann jedoch implizit mit Hilfe von Positionsinformationen der Attribute in den Spalten vermieden werden. Letzteres verhindert eine Sortierung der einzelnen Attribute, bietet jedoch effizienteres Lesen im Falle des Zugriffs weiterer Attribute, da der implizite Positionsschlüssel direkt zum Adressieren benutzt werden kann.

Die spaltenweise Organisation der Daten bietet insbesondere Vorteile im Lesen weniger Attribute, da für die Anfrage unnötige Spalten nicht gelesen werden müssen, wie es z. B. bei

der zeilenorientierten Struktur auf Festplatten der Fall ist. Demnach wird allein schon durch die Reduzierung der zu lesenden Datenmenge für attribut-fokussierte Anfragen, wie sie im analytischen Umfeld häufig Fall vorkommen, ein Geschwindigkeitsvorteil erreicht.

Dem entgegen steht der Nachteil, dass die vertikale Partitionierung, im Vergleich zur konventionellen zeilenorientierten Struktur beim Wiederherstellen vollständiger Zeilen aus Relationen zu Mehrkosten führt. Im Falle eines festplattenbasierten Datenbanksystems muss hierfür auf sämtliche Dateien zugegriffen werden, so dass das sequenzielle Lesen eines Tupels durch einen zufälligen und damit langsamen Zugriff ersetzt wird. Gleiches gilt für den schreibenden Zugriff, da hier die komplette Relation auf die Dateien verteilt werden muss. Abbildung 1 verdeutlicht beide Zugriffsmuster im Zusammenhang mit den jeweilig optimalen Speichervarianten.

Insbesondere im Kontext von transaktionalen bzw. kombinierten Workloads haben die gerade genannten Nachteile den Einsatz des spaltenorientierten Schemas bisher verhindert. Erst die Kombination dieses Ablagemusters mit einer Hauptspeicherdatenbank ermöglicht den Einsatz in solchen Szenarien. Hinzu kommt, dass sich die Annahmen hinsichtlich transaktionaler Verarbeitung über die Jahre geändert haben. Wie im späteren Abschnitt „Anfrageverteilung“ gezeigt wird, sind in der transaktionalen Welt die meisten Anfragen lesender Natur. Auch wenn diese zu einem großen Teil aus Schlüsselzugriffen bestehen, kann eine spaltenorientierte Hauptspeicherdatenbank Vorteile ausspielen, da das Zusammensetzen kompletter Relationen durch die schnelle „Random-access“-Funktionalität des Hauptspeichers im Vergleich zur Festplatte den Nachteil der vertikalen Dekomposition ausgleichen kann. Dies

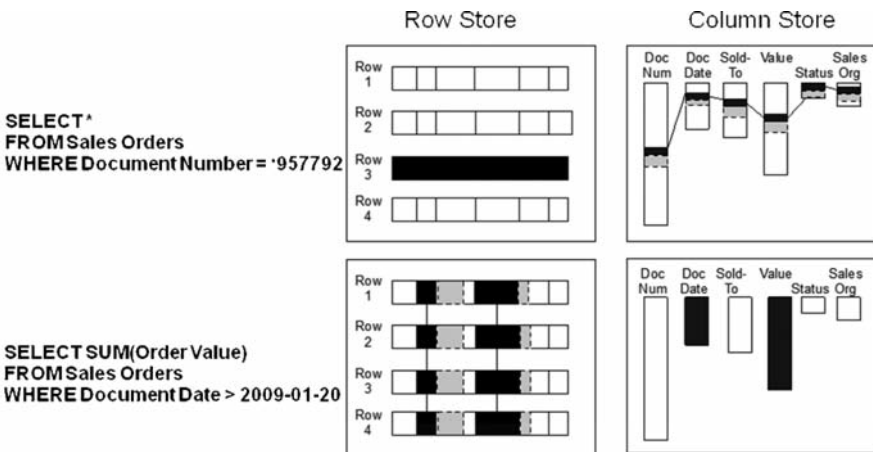


Abbildung 1: Datenzugriff: Zeilen und Spaltenorientiert. Oben: Zugriff auf einen kompletten Datensatz, typisch bei OLTP-Anfragen. Unten: Vollständige Scans über bestimmte Spalten, typisch bei OLAP-Anfragen

trifft vor allem dann zu, wenn Anfragen zumeist auf wenige Attribute zugreifen und dabei eine große Anzahl an Zeilen lesen. Hier kommt die Möglichkeit des sequenziellen Lesens auf dedizierten Spalten zum Tragen, was um Faktoren schneller ist als der wahlfreie Zugriff, was sowohl für hauptspeicher- als auch festplattenbasierte Systeme gilt.

Kompression

Das Moore'sche Gesetz besagt, dass sich die Rechenleistung von Prozessoren alle 18 Monate verdoppelt und gilt nun mehr seit über 35 Jahren. Dies gilt hingegen nicht für die Entwicklung von Festplatten- oder Hauptspeicherzugriffsgeschwindigkeiten, so dass für die Mehrzahl der Anfragen trotz cache-optimierter Datenstrukturen zunehmend I/O-Verzögerungen den größten Engpass darstellen. Die wachsende Kluft zwischen den Geschwindigkeitszuwächsen der CPUs auf der einen Seite und des Speichers auf der anderen Seite ist ein bekanntes Problem und wird bspw. in [BON99] beschrieben. Leichtgewichtige und verlustfreie Kompressionstechniken helfen diesen Flaschenhals zu verringern, indem sie es eine effizientere Nutzung der zur Verfügung stehenden Bandbreite erlauben. Mögliche Mehrkosten zum Entpacken der komprimierten Daten können durch die im Vergleich zur Speicherbandbreite stärker angewachsene CPU-Leistung im Allgemeinen problemlos aufgefangen werden.

Verschiedene Verfahren erfordern einen teils höheren Dekompressionsaufwand, jedoch ermöglicht die Komprimierung – aufgrund der weiterhin stark wachsenden CPU-Leistung – eine effiziente Überbrückung von I/O-Wartezyklen, insbesondere bei Verwendung von paralleler Dekompression.

Kompressionsverfahren nutzen Redundanz innerhalb der Daten und Wissen über die jeweilige Datendomäne, um möglichst effektiv zu komprimieren. Aufgrund mehrerer Eigenschaften von spaltenorientierten Datenstrukturen ist Komprimierung bei diesen besonders effizient. Da alle Daten innerhalb einer Spalte den selben Datentypen aufweisen und in der Regel eine ähnliche Semantik besitzen, ergibt sich ein niedriger Informationsgehalt (Entropie), d. h. es gibt in vielen Fällen nur wenige unterschiedliche Werte. Insbesondere Unternehmensanwendungen, deren Aufgabe es ist, sich wiederholende Prozesse und Events abzuarbeiten bzw. festzuhalten, schöpfen so gut wie nie den Wertebereich des zur Verfügung stehenden Datentyps aus. Sehr oft ist es sogar der Fall, dass nur sehr wenige Werte verwendet werden, da z. B. der Betrieb nur einige verschiedene Materialien und Produkte verwendet.

Bei der **Laufängenkodierung** (engl. Run Length Encoding, RLE) wird die Wiederholung eines Wertes durch ein Wertepaar (Wert und Wertanzahl) gespeichert. Zum Beispiel wird die Sequenz „aaaa“ auf „a[4]“ komprimiert. Dieser Ansatz eignet sich besonders für sortierte Spalten mit wenig Varianz in den Attributwerten. Bei unsortierten Spalten sinkt die Effekti-

viät, so dass ein Sortieren obligatorisch erscheint. In einem spaltenorientierten Szenario wird dadurch eine ID für die Zeile notwendig, die zu dem oben genannten Mehraufwand führt.

Ein weiteres prominentes Beispiel für leichtgewichtige Kompressionstechniken ist **Dictionary-Encoding**, bei der häufig erscheinende Muster durch kürzere Symbole ersetzt werden. Im Falle eines Attributes einer Tabelle wird jeder eindeutige Wert in einem Wörterbuch abgelegt und der Wert in der Spalte durch einen Schlüssel ersetzt, der auf den Wert im Wörterbuch zeigt. Eine weitere Optimierung ist die bitweise Komprimierung dieser Schlüssel um die Länge so weit zu reduzieren, dass nur die notwendige Anzahl im Wörterbuch damit abgebildet werden kann, was die Effektivität dieses Verfahrens weiter erhöht, insbesondere wenn nur wenige verschiedene Werte vorliegen. Ein sortiertes Wörterbuch hingegen erlaubt zusätzliche Optimierungen auf der Anfrageseite, da sich von den Schlüsseln die originale Sortierung ohne Abgleich mit dem Wörterbuch ableiten lässt.

Durch die Komprimierung innerhalb der Spalten wird die Informationsdichte in Bezug auf den verbrauchten Speicherplatz erhöht. Dadurch können relevantere Informationen zur gleichzeitigen Verarbeitung in den Cache geladen werden, was zu einer Erhöhung der Bandbreitennutzung führt. Zu erwähnen ist, dass diese Techniken auch für zeilenorientierte Speicherschemata gelten, jedoch erst in der Kombination mit spaltenorientierter Ablage und der Datenverarbeitung im Hauptspeicher Vorteile im betrachteten Rahmen erzeugen, da die sich aus der zeilenbasierten Ablage ergebenden Cache-Misses den Nutzen der Kompression zunichtemachen.

Ein weiterer wichtiger Aspekt der Komprimierung ist der Nachteil, dass Datenmodifikationen eine erneute Komprimierung des Datenbestandes herbeiführen. Um diesen Aufwand zu umgehen und dennoch schnelle Einfüge-, Änderungs- und Löschoperationen zu erlauben, können Änderungen in einem Puffer gesammelt werden, der Datensätze unkomprimiert ablegt. Das Ergebnis jeder Anfrage besteht aus dem komprimierten Hauptspeicherbereich und dem Puffer, so dass auch noch nicht komprimierte Änderungen berücksichtigt werden. Um die Änderungen in den komprimierten Hauptspeicherbereich zu übertragen, werden in definierten Intervallen die gesammelten Änderungen des Puffers asynchron mit dem bestehenden komprimierten Datenbestand zusammengeführt.

Während frühere Arbeiten, wie [COR85], den Fokus auf die Verbesserung der I/O-Leistung durch Reduzierung der zu ladenden Daten behandelten, konzentrieren sich aktuelle Arbeiten [ABAO6], [WESoo] auf die Auswirkungen der Komprimierung hinsichtlich der Anfrageausführung, z. B. durch Verwendung von „Late-materialization“-Strategien. Hierbei stehen

leichte Kompressionsverfahren im Fokus, die einen direkten Zugriff auf einzelnen Werte erlauben, ohne dass der komplette Datenbestand entpackt werden muss.

Darüber hinaus kann die Leistung dadurch verbessert werden, dass die spaltenorientierte Anfrageausführung – wie in [ABA07] beschrieben – Kenntnis über das verwendete Komprimierungsverfahren hat, so dass die CPU während des Einlesens der Daten diese parallel dekomprimieren kann. Ein wichtiger Aspekt bei der Komprimierung ist dabei die Abwägung zwischen Komprimierungsverhältnis und den Kosten für die Dekomprimierung. Generell ist es das Ziel die Dekomprimierung, so lange wie möglich hinauszuzögern, um einen möglichst großen Nutzen aus den komprimierten Daten zu ziehen und nur wirklich benötigte Daten zu dekomprimieren.

Vor allem im Bereich der Unternehmensanwendungen kann Komprimierung für analytische Anfragen signifikante Vorteile bringen. Wenn z.B. diese nicht mehr auf voraggregierten Ergebnissen beruhen, sondern in Echtzeit berechnet werden, können Abfragen wesentlich schneller ausgeführt werden, wenn Kompressionstechniken eingesetzt werden, bei denen eine Aggregation ohne Dekompression möglich ist.

Betrachtet man Aggregationen, welche nun durch Echtzeitberechnungen an Stelle von voraggregierten Ergebnissen in Anfragen weitaus häufiger vorkommen, so hat die Mehrzahl der Operationen in Unternehmensanwendungen typisch analytische Query-Eigenschaften. Vor allem Kompressionstechniken, welche direkt ohne Dekompression aggregieren können, bringen hier wesentliche Vorteile.

Typischerweise wird bei Unternehmensdaten ein Kompressionsfaktor von etwa 10 erreicht, bei Finanz-Anwendungen steigt der Faktor sogar auf bis zu 50.

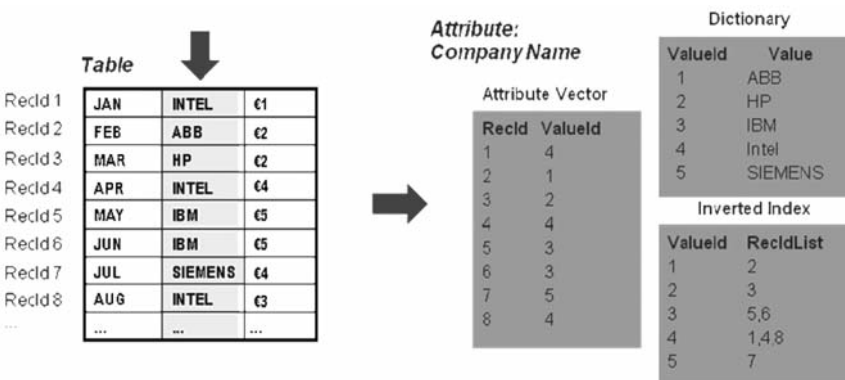


Abbildung 2: Dictionary Encoding an einer Beispieldatenbank

Heutige Unternehmensanwendungen

Im Folgenden wird auf die Charakteristiken und Anforderungen von heutigen Geschäftsanwendungen eingegangen und erörtert, wie diese von spaltenorientierten Hauptspeicherdatenbanken profitieren können.

Aufgrund steigender Datenvolumen, neuen Anforderungen an Geschäftsprozesse und steigender Nachfrage nach aktuellen Geschäftsdaten, werden Unternehmensanwendungen zunehmend komplizierter, da Mängel in der Datenmanagement-Infrastruktur ausgeglichen werden müssen. Dies schließt bspw. die redundante Speicherung von transaktionalen und analytischen Daten ein. Der Grund dafür ist, dass heutige Datenbanksysteme die Anforderungen moderner Unternehmen nicht erfüllen können, da diese zumeist für lediglich einen Anwendungsfall optimiert sind: entweder für OLTP- oder OLAP-Verarbeitung. Dieser Fakt führt zur einer Diskrepanz von Unternehmensanwendungen in Bezug auf das zugrundeliegende Datenmanagement, da mit konventionellen Datenbanken vor allem komplexe Operationen nicht in annehmbarer Zeit auszuführen sind. Während dieses Problem für analytische Anwendungen weithin anerkannt ist und dafür in der Regel separate Data Warehouses genutzt werden, trifft es ebenso auf transaktionale Anwendungen zu, die in modernen Systemen ähnlich komplex sind. Um diesen Nachteil der genutzten Datenmanagement-Infrastruktur auszugleichen, werden komplexe Operationen und länger laufende Prozesse zumeist in Batch-Jobs ausgelagert. Diese Lösung verlangsamt die Geschwindigkeit, mit der Geschäftsprozesse abgeschlossen werden können, wodurch möglicherweise externe Anforderungen nicht mehr erfüllt werden können. Ein weiterer und oft zum Einsatz kommender Lösungsansatz ist die Materialisierung vorberechneter Ergebnisse. Materialisierte Sichten in Data Warehouses für analytische Anwendungen sind hierfür ein Beispiel. Dies führt jedoch zu einer geringeren Flexibilität und erschwert die Wartung in solchen OLAP-Systemen. Ein analoges Verfahren findet in OLTP-Systemen Verwendung, wobei die Materialisierung von den jeweiligen Anwendungen verwaltet wird, da meist Anwendungslogik mit berücksichtigt werden muss. Durch die redundante Speicherung von Daten durch vordefinierte Charakteristiken wird den Performanzproblemen entgegengetreten, während gleichzeitig eine Steigerung der Komplexität und eine Abnahme der Flexibilität hingenommen wird. Neben der steigenden Programmkomplexität, die notwendig ist, um die Sichten konsistent zu halten, stört insbesondere die fehlende Flexibilität in heutigen Anwendungen, da Anforderungen programmseitig nur noch schwer umgesetzt werden können.

Zusammenfassend lässt sich sagen, dass der aktuelle Ansatz heutiger Unternehmensanwendungen zu einer steigenden Komplexität bei der Datenmodifizierung und -speicherung führt. Folgend wird gezeigt, welche Charakteristiken Unternehmensdaten und Anwendungen innehaben und wie sich diese auf aktuelle Software-Technologien, wie z. B. spalten-

orientierte Hauptspeicherdatenbanken, anwenden lassen und welche Verbesserungen und Vereinfachungen dadurch ermöglicht werden.

Anfrageverteilung

Datenbanken im Bereich des Unternehmensdatenmanagements werden allgemein abhängig von Ihrer Optimierung – entweder für OLTP oder OLAP – klassifiziert. Wie zuvor gezeigt, umfassen Unternehmensanwendungen zumeist OLTP-Charakteristiken. Bislang wurde oftmals angenommen, dass der Workload von Unternehmensanwendungen neben den Select-Anweisungen weiterhin zu einem großen Anteil aus Insert- und Update(Modification)- und Delete-Anfragen besteht. Diese Annahme lässt sich an der Zusammensetzung des TPC-C Benchmarks in Abbildung 3 ablesen, der immerhin zu 46 % aus Schreibanfragen besteht.

Um zu bestimmen, ob in der Realität lese- oder schreiboptimierte Datenbanken besser für tatsächliche Workloads geeignet sind, wurden 65 Installationen eines Enterprise-Resource-Planning-Systems (ERP) in Hinsicht auf die durchgeführten Datenbankabfragen untersucht. Das Ergebnis dieser Untersuchung ist ebenfalls in Abbildung 3 zu sehen. Diese zeigt deutlich, dass die Verteilung der Anfragen äußerst leselastig ist. Mindestens 80 % aller Anfragen sind Select-Anfragen. Dieser Fakt spricht für die Verwendung einer leseoptimierten Datenbank, gegebenenfalls mit einem schreiboptimierten Puffer. Spaltenorientierte Hauptspeicherdatenbanken mit leichtgewichtiger Kompression bieten den besten Durchsatz bei komplexen Leseanfragen.

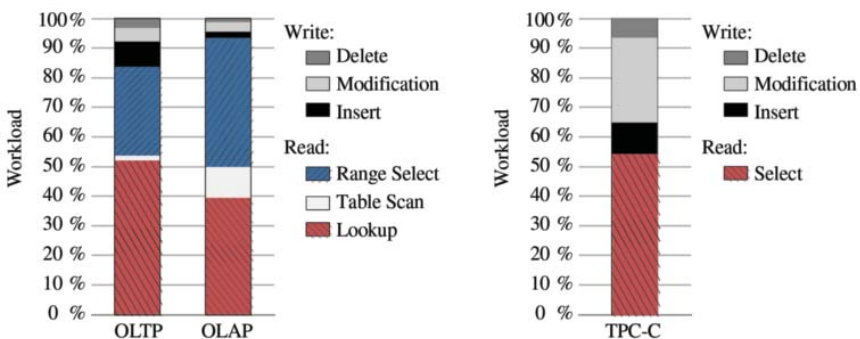


Abbildung 3: Tatsächliche Workloads nach Anwendungsfall im Vergleich zum TPC-C Benchmark

Werteverteilung

Neben der Verteilung von Datenbankabfragen sind auch die Werteverteilung und Eigenschaften der Daten selbst wichtig für spaltenorientierte Datenbanken. Eine weitgehende

Annahme für Daten von Geschäftsanwendungen ist, dass diese eine hohe Werteverteilung bzw. Entropie aufweisen, also sehr viele verschiedene Werte für die einzelnen Attribute existieren. Untersuchungen haben gezeigt, dass Werte einzelner Spalten in der Realität jedoch kaum verteilt sind, also nur wenige unterschiedliche Werte existieren. Zudem wird oftmals nur ein sehr kleiner Teil der verfügbaren Attribute überhaupt verwendet.

Um die Werteverteilung zu validieren, wurde die Häufigkeit der unterschiedlichen Werte pro Attribut gemessen. Diese Analyse geschah anhand der Haupttabellen der Finanzverwaltung sowie Verkaufs- und Vertriebsdaten mehrerer Kundensysteme. In der untersuchten Unternehmensanwendung besteht bspw. ein Buchhaltungsbeleg aus rund 100 Attributen, während sich die dazugehörigen Positionen jeweils aus 300 Attributen zusammensetzen. Die Tabelle für Materialbewegungen besteht aus 180 Attributen.

Abbildung 4 zeigt die prozentuale Häufigkeit gruppiert nach der Werteverteilung pro Spalte über alle untersuchten Tabellen der analysierten Kundensysteme. Der Abbildung ist zu entnehmen, dass die meisten Attribute der Tabellen der ersten Gruppe zuzuordnen sind, welche lediglich 1 - 32 unterschiedliche Werte besitzen. Der Großteil dieser Spalten beinhaltet darüber hinaus lediglich einen „unterschiedlichen“ Wert, welcher entweder der Null-Wert oder ein Standardeintrag ist. Folglich werden nicht alle Attribute von der Anwendung genutzt, was von großem Interesse für die Anwendungsoptimierung ist. Zu erwähnen ist, dass die ermittelten Werte Durchschnittswerte aller betrachteten Kundensysteme sind und die Ausprägung der einzelnen Attribute vom Unternehmen und dem jeweiligen Industriezweig abhängig ist.

Bei genauerer Analyse der Tabelle der Materialbewegungen wurde festgestellt, dass lediglich 43 von 180 Attributen eine signifikante Anzahl (>1023) an unterschiedlichen Werten aufweisen. Folglich haben 76% aller Spalten eine relative geringe Werteverteilung. Das Attribut mit der höchsten Anzahl an unterschiedlichen Werten ist das Attribut „Bewegungsnummer“, welches gleichzeitig auch primärer Schlüssel der Tabelle ist. Weiterhin haben auch die „Transportanfragenummer“, „Anzahl“, „Datum“ und „Zeit“ eine hohe Werteverteilung. Bemerkenswert ist weiterhin, dass 35% aller Spalten nur einen Wert besitzen.

Folglich lassen sich sehr wichtige unternehmensspezifische Charakteristiken bestimmen. Abhängig von der Anwendung und dem jeweiligen Industriezweig, wird in vielen Spalten nur jeweils ein Wert hinterlegt. Diese Anwendungen profitieren daher signifikant von einer leichtgewichtigen Komprimierung. Hinzukommt dass vor allem Spalten für analytische Anfragen von Bedeutung sind, die eine gewisse Kardinalität von verschiedenen Werten aufweisen. Dieser Fakt spricht für die Verwendung von spaltenorientierten Datenbanken, in der nur die projizierten Attribute gelesen werden müssen.

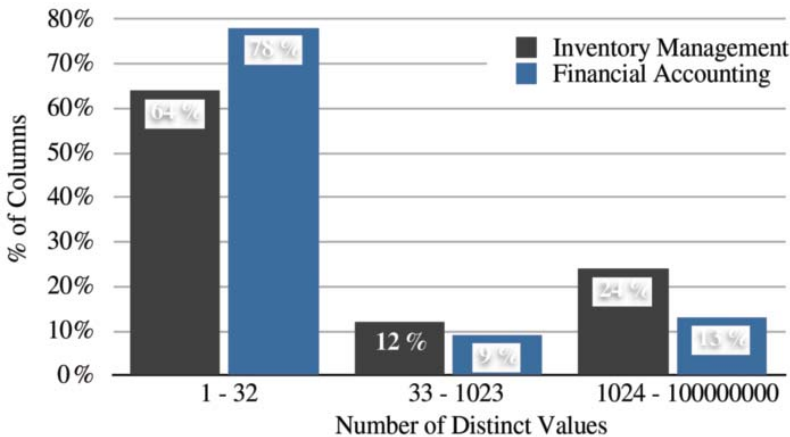


Abbildung 4: Werteverteilung bzw. Entropie der Daten in unterschiedlichen Unternehmensbereichen

Anwendungsverteilung

Weiteres Potenzial offenbart sich bei der Betrachtung der Anwendungsarchitektur, die im Laufe der Jahre bei den meisten Unternehmen natürlich gewachsen ist. Die Vorgehensweise, auftretende Engpässe durch die Auslagerung von Daten in dedizierte Spezialsysteme zu beheben, führte zu einer Grundarchitektur ähnlich Abbildung 5. Die Trennung der Datenbasis in unterschiedliche, in sich abgeschlossene Silos hat einerseits zu einer gewissen Sicherheit geführt, brachte allerdings auch einige Nachteile mit sich, deren Auswirkungen erst mit fortschreitender Entwicklung wahrhaft sichtbar wurden. Die Teilung der Daten auf getrennte Datenbanken für einerseits das Tagesgeschäft bzw. die laufende Auftragsbearbeitung und andererseits die Analyse von historischen Daten für Reportinganwendungen garantierte, dass zeitkritische Manipulationen und Abfragen von einzelnen Tupeln für das Tagesgeschäft nicht von lang laufenden analytischen Abfragen gestört wurden. Jedoch wurde mit steigendem Datenaufkommen und steigenden Anforderungen an die Reportinganwendungen deutlich, dass diese Separierung nicht länger aufrecht erhalten werden sollte. Mit einer getrennten Datenbasis ist es systemimmanent nicht möglich, OLAP-Anfragen auf den aktuellen Daten auszuführen, da das dedizierte OLAP-System zwingend auf einer älteren Datenbasis als den aktuellen OLTP-Daten arbeiten muss, da zur Transformation der Daten erst ein rechenintensiver ETL-Prozess stattfinden muss. Weiterhin können Reportinganfragen nur auf denjenigen voraggregierten Daten ausgeführt werden, die im ETL-Prozess definiert wurden. Sogenannte Ad-hoc-Queries sind somit nur in begrenztem Maße möglich. Eine Erweiterung der automatisch generierten Aggregationen ist prinzipiell möglich, erhöht jedoch wiederum den Rechenaufwand des ETL-Prozesses, wodurch die zuvor angesprochene Verzögerung zu den aktuellen Daten erhöht wird. Die Redundanz der Daten resultiert in ge-

steigerten Kosten für die physische Datenhaltung, sowie in einem erhöhtem Aufwand zur Pflege dieser Daten, welcher im Allgemeinen den höheren Kostenfaktor ausmacht. Insbesondere bei der Entwicklung neuer oder der Anpassung bestehender Anwendungen resultiert die gesteigerte Komplexität, mehrere Datenquellen kombinieren zu müssen, allgemein in einem erheblichen Mehraufwand. Ein Umstieg auf eine kombinierte Datenbank für OLTP- und OLAP-Anwendungen ist demzufolge auch aus diesen Gründen sinnvoll.

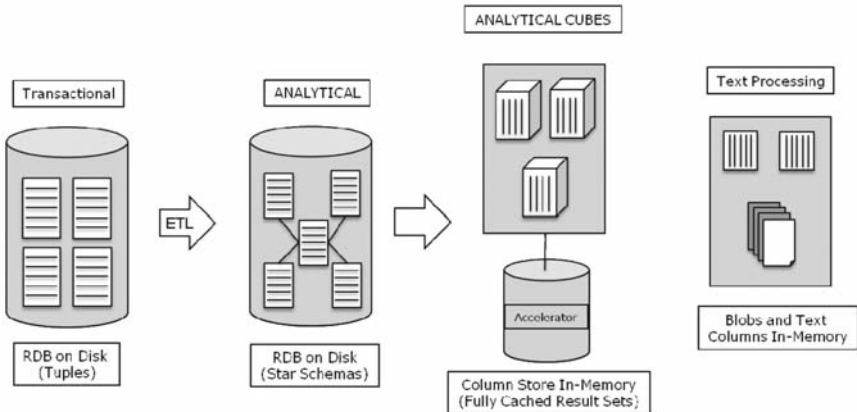


Abbildung 5: Standard-Datenaufteilung in Unternehmen für transaktionale und analytische Anwendungen

Hauptspeicherbasiertes Datenmanagement für Unternehmensanwendungen

Kombiniert man die Erkenntnisse der vorherigen Absätze, so wird ersichtlich, dass es sowohl aus Gesichtspunkten der heutigen Anfrageverteilung, der Werteverteilung und der Anwendungsverteilung von Unternehmen geboten ist, ein spaltenorientiertes Hauptspeicherdatenmanagement zu realisieren. Die anfänglich genannten Trends und Entwicklungen der Hardware stellen die Grundlagen für solche eine Umsetzung bereit. Hinsichtlich der Organisation dieses Datenmanagements bietet ein Aufbau wie in Abbildung 6 gezeigt Vorteile.

Jegliche Daten verbleiben zur Laufzeit des Systems vollständig im Hauptspeicher. Der Hauptspeicher stellt damit die primäre „Persistenz“ dar. Die Daten werden zusätzlich zu Sicherungszwecken auf einen nicht-flüchtigen Datenträger (Festplatte oder SSD) geschrieben. Von diesem Datenträger wird sofern notwendig die Datenbasis nach einem Neustart bzw. einem Recovery-Vorfall wieder geladen. Der Flaschenhals ist bei solch einem Aufbau nicht länger die Anbindung zu einem Datenträger, sondern die Anbindung zum Hauptspeicher. Obwohl die Zugriffszeiten um ein Vielfaches geringer sind, sollte demzufolge trotzdem nach dem Prinzip „Locality is King“ auf eine cachesensitive Implementierung von Algorithmen und Datenstrukturen geachtet werden.

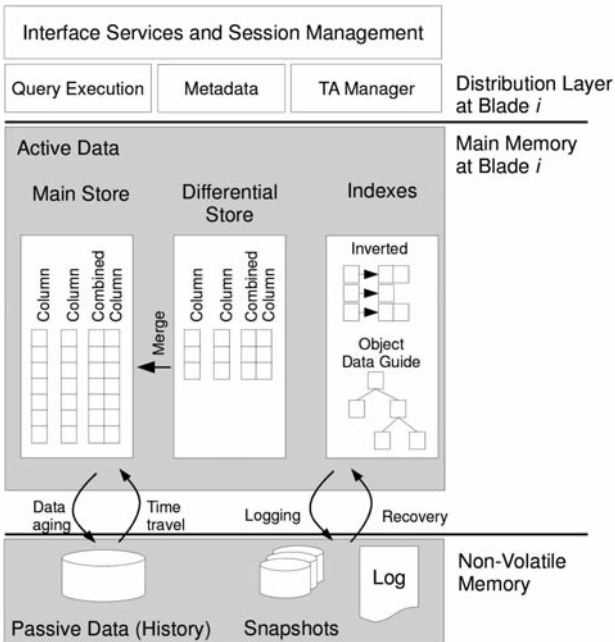


Abbildung 6: Architektur eines kombinierten OLTP- und OLAP-Systems auf Hauptspeicherbasis

Auswirkungen auf Unternehmensanwendungen

Der Einsatz des beschriebenen Datenmanagements hat abseits der trivialen Auswirkung der stark höheren Geschwindigkeit einige weitere direkte Auswirkungen, die hier beispielhaft für die Anwendungsentwicklung kurz beschrieben werden.

Auswirkungen auf die Anwendungsentwicklung

Die Entwicklung von Anwendungen wird in vielfältiger Weise vereinfacht. Am offensichtlichsten ist die Verbesserung des logischen Zugriffs, der durch das sogenannte „Single-source-of-truth“-Prinzip erreicht wird. Bei komplexen Anwendungen, die bisher Zugriff auf mehrere der bisher getrennten Silos erforderten, entfällt damit jegliche Notwendigkeit zur Herstellung von mehreren Datenbankverbindungen und zur Abbildung der unterschiedlich repräsentierten Daten auf ein gemeinsames Modell (Mapping). Weiterhin müssen Programmierer anwendungsseitig weniger Caches vorsehen, um Latenzen beim Zugriff auf die Datenbank zu kompensieren. Da die Verwaltung und Synchronisation von Caches aufwändig und fehleranfällig ist, sinkt damit einerseits der Anteil an störendem technischem Code, der hinsichtlich der eigentlichen Geschäftslogik keinen Mehrwert bietet, und andererseits wird die Fehleranfälligkeit der gesamten Anwendung reduziert. Auch die Bewegung von Daten inner-

halb eines Programms kann insofern vereinfacht werden, dass eine Weitergabe von Daten je nach Anwendungsfall einfach ausbleibt und die jeweils aktuell benötigten Daten neu von der Datenbank angefordert werden.

Datenbankprogrammierer profitieren neben der offensichtlichen Tatsache, dass lediglich eine Datenbank statt viele verschiedene Silos existiert, weiterhin von einer verminderten Anzahl an Indizes, da diese zum Großteil durch die gute Scanperformance unnötig werden. Auch die Pflege einer Vielzahl von vordefinierten Aggregationen und materialisierten Sichten, welche bei genauerer Betrachtung lediglich aus der Not erzeugte technische Kopien der Ursprungsdaten sind, entfällt ersatzlos. Dies erlaubt eine Fokussierung der Tätigkeit auf den eigentlichen Hauptbestandteil der Datenbasis, nämlich die ursprünglichen Unternehmensdaten und ihre Repräsentation in der Datenbank.

Literatur

- [ABAo6] Abadi, D.J., Madden, S., Ferreira, M. „Integrating compression and execution in column-oriented database systems.“ SIGMOD 2006
- [ABAo7] Abadi, D.J., Myers, D.S., DeWitt, D.J., Madden, S. „Materialization strategies in a column-oriented dbms.“ ICDE 2007
- [BON99] Boncz, P.A., Manegold, S., Kersten, M.L. „Database Architecture Optimized for the New Bottleneck: Memory Access.“, VLDB 1999
- [COP85] Copeland, G.P., Khoshafian, S. „A decomposition storage model.“ SIGMOD Conference, pp. 268– 279. ACM Press 1985
- [COR85] Cormack, G.V.: Data compression on a database system. Commun. ACM 28(12), 1336–1342 1985
- [KRU10] Krueger, J., Tinnefeld, C., Grund, M., Zeier, A., Plattner, H. „A case for online mixed workload processing.“, DBTest. ACM 2010
- [WESoo] Westmann, T., Kossmann, D., Helmer, S., Moerkotte, G. „The implementation and performance of compressed databases.“ SIGMOD 2000