Grundlagen der Anfrageoptimierung in virtuellen Data Warehouses

Oliver Dunemann

Zusammenfassung

Ein virtuelles Data Warehouse entspricht einer integrierten Sicht auf mehrere Data Marts einer Organisation. Es dient zum Einen dazu, alle steuerungsrelevanten Informationen an einer zentralen Stelle für Analysen zur Verfügung zu stellen. Zum Anderen wird es durch die Kombination der Informationsquellen möglich, zusätzliche Informationen zu generieren, die aus der isolierten Betrachtung einzelner Data Marts separat nicht abzuleiten sind. Da eine der Kernanforderungen an entscheidungsunterstützende Systeme die Gewährleistung konstant schneller Antwortzeiten ist, kommt der Optimierung multidimensionaler Anfragen an ein virtuelles Data Warehouse entscheidende Bedeutung zu. Analog zur regel- sowie kostenbasierten Anfrageoptimierung in zentralen relationalen Datenbankmanagementsystemen sind Modelle und Verfahren notwendig, die die Bestimmung einer möglichst optimalen Auswahl und Reihenfolge der Operationen ermöglichen, die zur Beantwortung einer Anfrage durchzuführen sind.

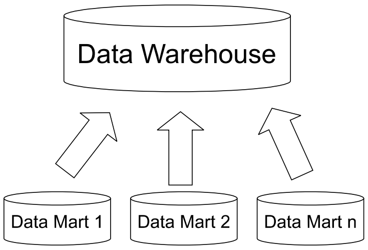
1 Das virtuelle Data Warehouse

Qualitativ hochwertige, schnell verfügbare Informationen zum Zweck der Steuerung und Kontrolle von Unternehmen und anderen Organisationen gewinnen eine zunehmende strategische Bedeutung. Aus diesem Grund hat sich die Integration verteilter, heterogener Informationen mit dem Ziel, aus diesen neue Informationen einer höheren Qualität zu generieren, zu einer wichtigen Aufgabe entwickelt. Um diese erfüllen zu können, muss ein analyseorientiertes System die Möglichkeit bieten, Daten in Beziehung zu mehreren Dimensionen zu aggregieren, konsolidieren, weiterzuberechnen und letztendlich zu visualisieren [CCS93].

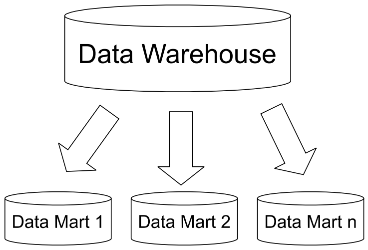
Ein Ansatz zur Lösung dieser Aufgabe ist das Data Warehouse-Konzept. Unter einem Data Warehouse versteht man nach Inmon ([Inm96]) eine themenorientierte, integrierte, zeitbezogene und dauerhafte Sammlung von entscheidungsunterstützenden Informationen. Während in einem Data Warehouse alle in einer Organisation vorliegenden Informationen berücksichtigt werden, handelt es sich bei Data Marts um 'kleine' Data Warehouses, die nur Teilbereiche wie beispielsweise fachliche oder regionale Abteilungen abdecken. Beiden gemeinsam ist, dass sie getrennt von den Daten der operativen Systeme ausschließlich zu Analysezwecken vorgehalten werden. Durch diese Entkopplung wird es möglich, den speziellen Anforderungen der jeweiligen Anwendungsszenarien beispielsweise durch angepasste Modellierung Rechnung zu tragen.

Grundsätzlich kann die Implementierung eines Data Warehouses mit zwei unterschiedlichen Vorgehensweisen erfolgen, die in Abbildung 1 miteinander verglichen werden. Bei Anwendung des Top Down-Ansatzes wird zunächst ein unternehmensweites Data Warehouse aufgebaut, dessen Inhalte anschließend in abteilungsspezifische Data Marts aufgeteilt und den Abteilungen zugänglich gemacht werden. Im anderen Fall werden anfangs mehrere Data Marts implementiert, die im Anschluss daran zu einem Data Warehouse zusammengefasst werden. Die zweite Vorgehensweise bietet die Vorteile, dass schneller erste Lösungen umgesetzt und dass die Anforderungen aus den Abteilungen besser berücksichtigt werden können. Nachteilig wirkt sich aus,

*Diese Arbeit wird von der DFG gefördert (FOR 345/1).



(a) Bottom Up



(b) Top Down

Abbildung 1: Vorgehensweisen zur Implementierung eines Data Warehouses

dass einzelne Integrationsschritte unter Umständen mehrfach vollzogen werden müssen, wenn die selben Datenquellen für mehrere Data Marts benötigt werden. Bei beiden Ansätzen werden sowohl die Data Marts als auch das Data Warehouse materialisiert und liegen somit redundant vor. Im hier vorgestellten Ansatz des 'virtuellen Data Warehouses' wird aus den oben angeführten Gründen der Bottom Up-Ansatz verfolgt, wobei nicht zwangsläufig eine Materialisierung der integrierten Daten stattfindet. Stattdessen werden die Data Marts virtuell zu einer organisationsweiten Datenbasis integriert. Dazu werden keine Annahmen bezüglich der physikalischen Modellierung der Datenwürfel in den Komponentensystemen getroffen. Es sollen dabei in den einzelnen Data Marts vorliegende Strukturen und Informationen nicht nur zusammengestellt werden. Vielmehr sollen durch deren Kombination neue Informationen generiert werden, die durch getrennte Analysen in einzelnen Datenquellen nicht zu entdecken gewesen wären.

Da die Analyse von Daten oft einen interaktiven Prozess darstellt, sind die Antwortzeiten ein wesentliches Kriterium für die Einsetzbarkeit eines OLAP-Systems. Eine gute Optimierung von Anfragen an ein virtuelles Data Warehouse kann hier einen entsprechenden Beitrag leisten.

2 Phasen der Anfrageoptimierung

Die Verarbeitung von Anfragen in Datenbanksystemen wird in mehreren Schritten vollzogen [GMUW00]. Zunächst wird im Prozess der sogenannten *Query Compilation* anhand der Anfrage der Ausführungsplan generiert, der die geringste Laufzeit erwarten lässt. Der zweite Schritt ist das Ausführen der einzelnen Elemente des Plans, die eigentliche *Query Execution*. Da der zweite Schritt von den Komponentensystemen, den Data Marts, durchzuführen ist, liegt der Fokus im Folgenden auf der Generierung des Ausführungsplans. Die Phase der Query Compilation lässt sich in die folgenden Teilphasen untergliedern:

1. *Parsen der Anfrage:*
   Die in der Datenbankanfragesprache formulierte Anfrage, muss in ein Format transformiert werden, das für die weiteren Bearbeitungsschritte geeignet ist. Grundlage bildet hier im Fall von relationalen Systemen die relationale Algebra.

2. *Anfrageoptimierung:*

   (a) *Auswahl eines logischen Ausführungsplans:*
   Die Anfrage kann als Baum interpretiert werden, dessen Wurzelknoten das Endergebnis der Anfrage generiert. Da mehrere Bäume inhaltlich äquivalent sind, aber unterschiedliche zu erwartende Aufwände aufweisen, wird mit Hilfe verschiedener Regeln ein Baum generiert, der einen möglichst geringen Aufwand erwarten lässt.

   (b) *Auswahl eines physikalischen Ausführungsplans:*
   Die meisten logischen Operationen können mittels verschiedener Algorithmen imple-

mentiert werden. Den letzten Schritt der Query Compilation bildet die Auswahl der im vorliegenden Kontext optimalen Verfahren.

2.1 Parsen der Anfrage

Wie oben angesprochen, wird in RDBMS eine Anfrage in ein auf der relationalen Algebra basierendes Format transformiert. Um ähnliches auch für den multidimensionalen Fall zu ermöglichen, wird eine entsprechende multidimensionale Algebra benötigt, die im Folgenden kurz vorgestellt wird. Dazu werden zunächst die Schemabestandteile kurz beschrieben. Anschließend wird beispielhaft als ein Vertreter der Operationen die Bildung der Vereinigungsmenge skizziert.

Elemente, die Punkte oder Bereiche der Realwelt beschreiben, werden im Data Warehouse-Umfeld in Dimensionen eingeteilt. In den Dimensionen werden jeweils inhaltlich zusammengehörige Elemente gruppiert, die eine Auswertungsrichtung in verschiedenen Granularitäten beschreiben. Eine Dimension $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ ist eine Gruppierung logisch zusammengehöriger, diskreter Objekte, den Dimensionselementen. Jede Dimension enthält ein Element, das die gesamte Dimension einschließt. Eine Menge von Dimensionen sei mit $\mathcal{D}_{set}$ bezeichnet.

Weitere Elemente sind die Kennzahlen *(measures)*, die dem Anwender in Anfragen für Analysen zur Verfügung stehen. Die Kennzahlen stellen eine eigene Dimension dar, da es sich um zusammengehörige Objekte handelt. Eine Anforderung an OLAP-Systeme ist, dass die Kennzahlendimension genauso wie alle anderen Dimensionen behandelt wird [AGS97]. Dieses rührt daher, dass Abfragen, in denen nach Kennzahlen selektiert oder gruppiert wird, diesen ebenfalls den Charakter von Dimensionen verleihen. Daher werden Kennzahlen analog zu den Dimensionen behandelt. Zusätzlich stellt ihre Zusammenfassung eine eigenständige Dimension dar. Die Kennzahlendimension $[M = \{m_1, m_2, \ldots, m_n\}$ ist die Menge der Kennzahlen m_i. Eine Kennzahl ist dabei als als Erweiterung Dimensionsbegriffes zu verstehen. So können ihre Domänen stetig sein. Zusätzlich muss zum Einen eine Aggregationsfunktion $agg(m_i)$ definiert sein, die das Verhalten der Kennzahl bei der Zusammenfassung mehrerer Elemente anderer Dimensionen beschreibt. Zum Anderen kann eine Liste von Dimensionen angegeben werden, die zur korrekten Bestimmung der Kennzahl vorliegen muss. Folglich stellen Dimensionen lediglich einen Spezialfall von Kennzahlen dar.[1]

Eine Dimension kann Elemente enthalten, die unterschiedliche Granularitäten aufweisen. Daher können zu jeder Dimension eine oder mehrere Hierarchien angegeben werden. Eine Hierarchie $\mathcal{H}(\mathcal{D}) : \mathcal{D} \rightarrow \mathcal{D} \cup \{\perp\}$ zu einer Dimension $\mathcal{D}$ ist eine eindeutige Abbildung innerhalb von Elementen einer Dimension. Sie stellt die inhaltliche Komponentenbeziehung dar. Die Eindeutigkeit wird gefordert, damit jedes Dimensionselement auf genau ein anderes oder auf `NIL` ($\perp$) abgebildet wird. Die Abbildung darf keine Zyklen enthalten. Dadurch wird ebenfalls ausgeschlossen, dass ein Element auf sich selbst abgebildet wird. Es gibt jeweils genau ein Element, das auf $\perp$ abgebildet wird, wodurch angezeigt wird, dass das Element die kleinste Granularität der Hierarchie darstellt. Eine Menge von Hierarchien sei mit $\mathcal{H}_{set}$ bezeichnet.

Neben den Kennzahlen, die anhand einer beliebigen Kombination aus Dimensionselementen spezifiziert werden, können den Dimensionen Attribute, sogenannte Dimensionsattribute, zugeordnet werden. Ein Dimensionsattribut $\mathcal{DA}$ ist ein Attribut, das Elementen genau einer Dimension zugeordnet ist. Eine Menge von Dimensionsattributen sei mit $\mathcal{DA}_{set}$ bezeichnet. Es existiert somit eine Abbildung $f_{dimattr} : \mathcal{DA}_{set} \rightarrow d_i$, die jedem Dimensionsattribut diejenigen Dimensionselemente zuweist, für die es definiert ist.

Datenwürfel oder Hypercubes sind das Konzept, das die im oben definierten Bestandteile integriert. Ein Datenwürfel-Schema ist ein 4-Tupel $CUBE = \langle \mathcal{D}_{set}, \mathcal{H}_{set}, \mathcal{DA}_{set}, f_{dimattr} \rangle$. Zu jedem Schema können mehrere Instanzen existieren. Die Instanzen, oder Datenwürfel, sind Daten,

[1] An dieser Stelle ist zu beachten, dass noch keine Annahmen über die logische Modellierung der Daten getroffen werden. Die Aussage, dass Dimensionen spezielle Kennzahlen sind, wird hier nur für die konzeptionelle Ebene getroffen.

die dem Schema entsprechend strukturiert sind. In Anlehnung an [DT97] wird also die Instanz über eine Erweiterung der Schemadefinition beschrieben. Eine Instanz eines Datenwürfels ist ein 3-Tupel $cube = \langle CUBE, g_{dimattr}, g_{measures} \rangle$ mit folgenden Eigenschaften:

- $g_{dimattr} : \mathcal{DA}_{set} \times d_i \rightarrow \mathbf{D}(\mathcal{DA}) \cup \{\perp\}$ ordnet jedem Dimensionsattribut für jede Ausprägung eines Dimensionselements genau einen Wert zu. Ist das Dimensionsattribut für das als Argument verwendete Dimensionselement nicht definiert, so wird NIL als Funktionsergebnis geliefert.

- $g_{measures} : \mathcal{D}_1 \times \mathcal{D}_2 \times \ldots \mathcal{D}_n \rightarrow \{0; 1\}$ ist diejenige Abbildung, die einer Ausprägung von Dimensionselementen eine 1 zuordnet, wenn diese existiert und den Wert 0 sonst.

Ein Beispiel für eine Operation auf Datenwürfeln ist die Vereinigungsmenge. Sie dient dazu, zwei (oder mehr) schematisch gleichartige Objekte zu einem zusammenzufassen. Eine Vereinigung ist nur dann möglich, wenn die Mengen der Dimensionen und Dimensionsattribute identisch sind. Diese Anforderung muss für die Hierarchien nicht erfüllt sein.

Seien $CUBE_1 = \langle \mathcal{D}_{set}, \mathcal{H}_{set_1}, \mathcal{DA}_{set}, f_{dimattr} \rangle$ und $CUBE_2 = \langle \mathcal{D}_{set}, \mathcal{H}_{set_2}, \mathcal{DA}_{set}, f_{dimattr} \rangle$ zwei Datenwürfel-Schemata. Dann ist $CUBE_1 \cup CUBE_2 = \langle \mathcal{D}_{set}, \mathcal{H}_{set_1} \cup \mathcal{H}_{set_2}, \mathcal{DA}, f_{dimattr} \rangle$ das Schema der Vereinigung der beiden Datenwürfel. Die Vereinigung der Instanzen ergibt sich dann aus:

$$cube_1 \cup cube_2 = \langle CUBE_1 \cup CUBE_2, g_{dimattr_1} \cup g_{dimattr_2}, g_{measures_1} \wedge g_{measures_2} \rangle .$$

Zu beachten ist hier, dass die Abbildung $g_{dimattr}$ definitionsgemäß eindeutig ist. Eine Vereinigungsmenge der Instanzen kann folglich nur dann gebildet werden, wenn die Vereinigung dieser Abbildungen nicht zu einem Widerspruch führt. Abbildung 2 zeigt schematisch die Vereinigungsmengen-Operation.

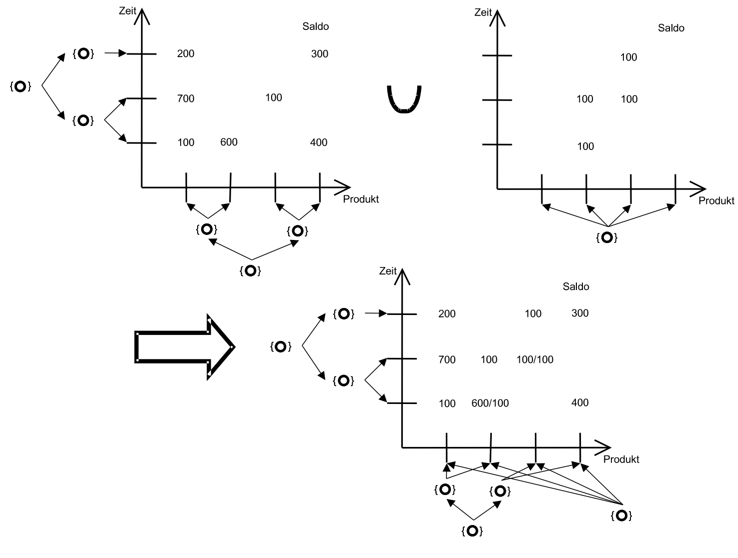


Abbildung 2: Vereinigung von Datenwürfeln

2.2 Anfrageoptimierung

Liegt die Anfrage in einer formalen Form wie der im vorigen Abschnitt skizzierten multidimensionalen Algebra vor, so können durch Umformungen äquivalente Ausdrücke erzeugt werden. Ein Beispiel für ein relationales System, das Anfragen umformuliert und an die entsprechenden Komponentensysteme verteilt, ist Garlic [HKWY97]. Die Suche nach dem kostenminimalen Plan findet hier dem Prinzip des Branch and Bound-Verfahrens statt: Zuerst werden ausgehend von der Struktur der Anfrage Wurzelpläne generiert. Diese werden dann mit Produktionsregeln einer Grammatik weiterentwickelt. Der jeweils gefundene Plan, der das Anfrageergebnis mit den geringsten Kosten generiert, markiert die obere Grenze für die Suche in weiteren Ästen. Mit Hilfe von sogenannten *STrategy Alternative Rules* (STARs), werden weitere Pläne erzeugt, solange noch Äste existieren, deren Kosten unterhalb der durch den oben gefundenen Plan determinierten Schranke liegen. STARs entsprechen also den gültigen Umformungen der zugrunde liegenden Algebra. Zusätzlich zu Grammatiken in zentralen Datenbankmanagementsystemen wird ein Operator benötigt, der dazu dient, eine Unteranfrage an ein Komponentensystem weiterzuleiten.

Einen Schritt weiter geht die dynamische Optimierung. Hier wird im Anschluss an den obigen Schritt während der Ausführung der Teilanfragen auf den Komponentensystemen stets überwacht, welche Teilantworten bereits vorliegen. Basierend auf Schätzungen der Laufzeiten aller Teilschritte wird in bestimmten Situationen während der Laufzeit der Anfrage der Anfragebaum modifiziert [ONK+97].

3 Ausblick

Im Anschluss an eine vollständige Definition der multidimensionalen Algebra kann anhand der definierten Umformungen eine Ableitung der STARs für den multidimensionalen Fall erfolgen. Es ist die Qualität dieser Pläne zu prüfen und weiterhin zu evaluieren, in wieweit die Qualität der so erhaltenen Anfragepläne durch geeignete Kostenmodelle und die dynamische Optimierung weiter gesteigert werden kann.

Literatur

[AGS97] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In Alex Gray and Per-Åke Larson, editors, *Proc. 13th Int. Conf. Data Engineering, ICDE*, pages 232–243. IEEE Press, 1997.

[CCS93] E.F. Codd, S.B. Codd, and C.T. Salley. Beyond Decision Support. *Computerworld*, 27(30), July 1993.

[DT97] A. Datta and H. Thomas. A Conceptual Model and an Algebra for On-Line Analytical Processing in Data Warehouses. In *Proc. Workshop on Information technologies and Systems 1997 (WITS'97)*, Atlanta, December 1997.

[GMUW00] H. Garcia-Molina, J.D. Ullman, and J. Widom. *Database System Implementation.* Prentice Hall, New Jersey, USA, 2000.

[HKWY97] L. Haas, D. Kossman, E. Wimmers, and J. Yang. Optimizing Queries across Diverse Data Sources. In *Proceedings of the 23th VLDB Conference*, San Jose, USA, Februar 1997.

[Inm96] W. H. Inmon. *Building the Data Warehouse.* John Wiley & Sons, 2. edition, March 1996.

[ONK+97] F. Ozcan, S. Nural, P. Koksal, C. Evrendilek, and A. Dogac. Dynamic Query Optimization in Multidadabases. *Bulletin of the IEEE Computer Society Technical Committee of Data Engineering*, 1997.