



APPLIX  TM1™



Wege zur Datenanalyse

White Paper

APPLIX GmbH
Boschetsrieder Str. 67
D-81379 München
Telefon: 089-748589-0
Telefax: 089-748589-20
Internet: <http://www.applix.de>
Email: info@applix.de



APPLIX iTM1
Wege zur Datenanalyse

Inhalt

Einführung

- Überblick
- Zusammenfassung

Hintergründe

- Heutige Unternehmens *umgebungen*
- Der Wert von multidimensionalen Daten

Multidimensionale Architekturen

- Philosophie der Multidimensionalen Datenhaltung
- Multidimensionale Architekturen
- Physikalische Multidimensionale Datenbanken
- Datenexplosion
- Physikalische Multidimensionale Datenbanken (MOLAP)
- Real-time Analytical Processing (RAP)

Andere Gesichtspunkte

- Single Hypercube vgl. Multi-Cube

Zusammenfassungen

Anhang A – Datenexplosion

INTRODUCTION

• Überblick

Die heutigen Unternehmen sehen die Notwendigkeit DSS Systeme immer wieder neu zu definieren. OLAP (OnLineAnalyticalProcessing) war als ein Kernelement der Datawarehouse Architekturen gedacht, ähnlich der Basistechnologie der Decision Support Systeme (DSS). Bei den rasenden Entwicklungen der heutigen Zeit machen es die OLAP Anbieter den Kunden nicht unbedingt leicht, die geeignete Architektur zu finden.

Diese Broschüre behandelt einige der Architekturen, die für die Implementierung von OLAP Systemen eingesetzt werden und fasst deren Vorteile zusammen.

Es gibt in erster Linie drei Architekturen die eingesetzt werden, um schnell und dynamisch anspruchsvolle Analysen der multidimensionalen Daten zu erstellen.



APPLIX iTM1 *Wege zur Datenanalyse*

ROLAP (Relational OLAP): nutzt eine Standard Relationale DB, um die Daten physikalisch zu speichern.

Physical multidimensional Databases (MOLAP): nutzt einen Speichermechanismus, der speziell für die Vorberechnung, Speicherung und Abfrage von multidimensionalen Daten optimiert wurde.

Real-time Analytical Processing (RAP): alle multidimensionale Werte werden in den Speicher geladen und abgeleitete Werte werden nach Bedarf aktuell berechnet.

ROLAP's sind am ehesten für große, transaktionsintensive Anwendungen wie z.B. umfangreiche Einzelposten Verkaufsanalysen geeignet. Die Vorteile liegen vor allem im verarbeiten von extrem großen Datensätzen. Außerdem liegt dieselbe Technologie zugrunde, wie bei den bestehenden RDBMS-basierten Systemen (obwohl eine andere Technik und Optimierung erforderlich ist). Doch ihre Komplexität, die Speicherorientierung im Vergleich zur Berechnungsphilosophie, Kosten und Performance führen zwangsläufig zu einer Begrenzung der Einsatzfähigkeit. Deshalb, werden sie meist nicht für Planungs- oder Finanzapplikationen eingesetzt.

MOLAP's sind am besten für statische Anwendungen mittlerer Größe geeignet. Anwendungen dieser Art sind z.B. die Analysen von historischen Umsatz- und Rechnungsinformationen. Obwohl die Batch-Vorberechnungen eine lange Zeit in Anspruch nehmen können, sind sie nicht optimal für dynamische Anwendungen geeignet, bei denen die Ergebnisse von neuen oder geänderten Daten benötigt werden. Die Notwendigkeit der Vorberechnung macht diese Architektur ungeeignet, um große speicherintensive Anwendungen mit mehr als 5 Dimensionen zu verarbeiten, da die Datenexplosion nicht vorhersehbar ist und die Verwaltung der Anwendung damit unter Umständen nicht mehr möglich ist.

RAP ist die Architektur für dynamische Anwendungen, für Umgebungen die bewegliche Arbeitsgruppen unterstützen müssen und die über eine umfassende Skalierungsmöglichkeit verfügen (von kleinen Desktopsystemen bis hin zu sehr großen Anwendungen mit mehr als 5 Dimensionen). Die Möglichkeit Berechnungen in Echtzeit auszuführen (damit erübrigt sich die Verzögerung durch Batchverarbeitung) machen RAP zur besten Architektur für dynamische Anwendungen wie z.B. Budgetierung, Rechnungswesen, und Planung und Management in den Bereichen Marketing, Verkauf und Auftragsabwicklung. Außerdem wird hierbei die Datenexplosion vermieden, die zwangsläufig bei vorberechneten abgeleiteten Werten entstehen. Damit ist RAP am besten für große Anwendungen (viele Dimensionen und viele Werte) geeignet, in denen auch der Bedarf an mobilen Arbeitsplätzen besteht. Aufgrund der Berechnung während der Ausführung ergeben sich natürlich auch etwas längere Antwortzeiten. Deshalb ist RAP eher ungeeignet für sehr große statische Anwendungen.

Quintessenz

Es gibt keine allumfassende Lösung. Zukünftige OLAP Benutzer müssen intensiv ihre Anforderungen mit den Stärken und Schwächen der Möglichkeiten der Architekturen abgleichen und sich dann für die geeignetste Lösung entscheiden.



HINTERGRÜNDE

Heutige Geschäftsfelder

Die heutigen Wettbewerbsumgebungen in Verbindung mit der einmaligen Leistungsfähigkeit der Computer haben unsere Informationssysteme erheblich beeinflusst. Die Wettbewerbssituation erfordert immer komplexere Analysen der wachsenden Datenmengen. Die anspruchsvollen Informationssysteme bieten eine große Menge an Basisdaten, um diese Analysen zu unterstützen. Aus dem Bedarf aus diesen riesigen Datenmengen wertvolle Informationen ziehen zu können, entstand die Technologie des Datawarehousing und OLAP.

Einige Skeptiker behaupten zwar, das Datawarehousing lediglich eine weitere List ist, uns dazubringen die Daten zu duplizieren, mehr Geld für neue Hardware auszugeben und letztlich noch teure Software für die Benutzer zu kaufen die auf die Daten zugreifen. Doch einige erfolgreiche Datawarehouse Implementierungen zeigen bereits die Vorteile dieser Vorgehensweise. Das Konzept der optimierten Datenhaltung für die Entscheidungsunterstützung und die Trennung von der operationalen Datenhaltung bietet viele Vorteile. Daten können schnell mit Hilfe multi-dimensionaler Abfragen verwaltet werden. Abgeleitete Werte können schnell und effektiv berechnet werden und die Datenintegrität wird als Teil des Bereinigungsprozesses während des Ladens in das Datawarehouse gewährleistet.

Einer der Bereiche der bei der Definition eines Datawarehouses meist vernachlässigt wird, ist die Datenselektion aus dem Warehouse. Glücklicherweise wird OLAP nun als Schlüsseltechnologie betrachtet, um den Anwendern intuitiven Zugriff auf die unternehmensrelevanten Informationen zu geben. OLAP kann damit auch als Technologie der ‚Know-how‘-Träger im Unternehmen betrachtet werden.

Der Begriff OLAP wurde zuerst von Dr. E.F. Codd, dem Erfinder des relationalen Modells, gebraucht. Damit wird eine Art Software beschrieben die benutzt wird, um Unternehmensdaten hierarchisch (TOP-DOWN) zu analysieren. Bemerkenswert ist auf jeden Fall, daß Dr. Codd in seiner OLAP Studie (White Paper) im Besonderen darauf hinweist, daß relationale Datenbanken nie dafür gedacht waren Funktionen für die Datensynthese, Analyse und Konsolidierung, also multi-dimensionale Analysen, bereitzustellen. OLAP Werkzeuge stellen die Daten nicht-technisch orientierten Benutzern auf eine intuitive Art und Weise zur Verfügung. Damit sind sie in der Lage die Unternehmensdaten so zu betrachten, wie es früher, im Zeitalter der ausgedruckten Listen und Berichts-/Abfragewerkzeuge, niemals möglich gewesen wäre.

Obwohl OLAP ein recht neuer Begriff ist, basiert es auf Grundregeln die vor über 20 Jahren bei der Definition der relationalen Datenbanken entstanden sind. Viele Leute setzen heute die relationale Technologie gleich mit dem Begriff Datenbanken. APL war die erste Maschinensprache, die die Idee der multidimensionalen Daten zum Leben erweckte und die bereits vor ungefähr 30 Jahren zum ersten Mal erwähnt wird.



APPLIX iTM1 *Wege zur Datenanalyse*

Der Wert der multidimensionalen Daten

Um die Bedeutung von OLAP messen zu können, ist es wichtig die multidimensionale Arten unserer heutigen Unternehmensdaten zu verstehen. Nehmen Sie als Beispiel den Verkaufsleiter einer Einzelhandelskette. Er möchte gerne wissen, wie sich die einzelnen Produkte in den Zweigstellen verkaufen. Er möchte Umsatztrends herausheben, erfolgreiche und weniger erfolgreiche Verkaufsstrategien herausfinden und einen allgemeinen Überblick über die Verkaufszahlen der Produkte bekommen bzw. was er tun kann um die Umsatzzahlen und damit den Gewinn zu steigern.

In erster Linie interessiert ihn jedoch der Umsatz je Produkt. Für jedes Produkt möchte er die Stückzahlen, Umsatz, Rabattierung und andere Statistik sehen. Diese Informationen möchte er täglich, für jeder Region, für jeden Vertriebsrepräsentanten und jeden Vertriebskanal sehen. Damit haben wir ein 6-Dimensionales Modell definiert. Die Dimensionen sind Wertarten (oft auch Variablen oder Einheiten genannt), Produkte, Zeit, Kanäle, Regionen und Vertriebsrepräsentanten.

Eine der Schlüsselfunktionen der OLAP Technologie ist die Möglichkeit für den Benutzer die Daten in einer Sichtweise zu betrachten, die für ihn Sinn macht, ohne im vornherein zu wissen wie er zu diesem Ergebnis kommt. Nehmen Sie noch einmal den Fall unseres Verkaufsleiters. Zunächst betrachtet er den Gesamtumsatz der Regionen für diesen Monat verglichen mit den selben Monat im Vorjahr. Dabei bemerkt er, daß die Umsätze trotz der Eröffnung von 3 neuen Zweigstellen in den vergangenen 3 Monaten nicht gestiegen ist. Eine Drilldown Funktion ermöglicht ihm dann zuerst die Umsätze pro Zweigstelle zu betrachten, weiterhin im Vergleich aktueller Monat zum selben Monat im Vorjahr. Er erkennt schnell, daß die meisten Zweigstellen einen beachtlichen Umsatzanstieg von rund 7% im Vergleich zum Vorjahr zu verzeichnen haben. Mit Hilfe des OLAP Werkzeuges ist er in der Lage die abgeleiteten Werte diesen Jahres als Prozentwert im Vergleich zum Vorjahr darzustellen und die Zweigstellen basierend auf diesen Prozentwerten zu sortieren. Am Ende dieser Liste findet er die Zweigstellen die offensichtlich nachgelassen haben. Durch einen Drilldown auf die Region erkennt er, daß die neuen Zweigstellen Kunden von den bestehenden Zweigstellen abwerben. Gleichzeitig haben seine Nachforschungen ergeben, daß es bei bestimmten Produktlinien einen Umsatzeinbruch gab. Um zu diesem Ergebnis zu kommen, musste er allerdings die Daten in einer völlig anderen Darstellungsweise betrachten als zuvor.

Ein anderes Beispiel ist die Finanzplanung. Die meisten Unternehmen planen entweder nach der TOP-DOWN- oder der BOTTOM-UP-Methode. Bei der TOP-DOWN Methode muß die Auswirkung des Plans bis auf die unterste Ebene der Verantwortlichen Manager betrachtet werden. Wenn die Planung nach der BOTTOM-UP Methode ausgeführt wird, müssen die Planzahlen konsolidiert werden, so daß die Führungskräfte dem Gesamtplan zustimmen können. In der Praxis sieht es jedoch meist so aus, daß die Führungsebene Richtlinien definiert, auf deren Basis die Planungsmitarbeiter ihre detaillierten Pläne erstellen. Der konsolidierte Plan wird dann nochmal vom Top-Management überarbeitet, um die definierten Richtlinien zu verfeinern. Daraufhin werden die Detail-Planzahlen nochmals überarbeitet. Dieser Vorgang kann sich dann mehrmals wiederholen. Und auch hier kann man eine Multidimensionalität erkennen. Die Dimensionen enthalten Kosten, Verantwortlichkeitsbereiche, Zeit, Versionen (Ist, Plan, Überarbeiteter Plan etc.). Zusätzliche Dimensionen enthalten normalerweise Produkt- und Kundeninformationen. Diese Modelle können auch komplizierte



APPLIX iTM1 Wege zur Datenanalyse

Geschäftsvorfälle beinhalten. Idealerweise sollte man in der Lage sein ein Schlüsselfeld zu ändern und sofort die Auswirkung auf das Modell zu erkennen. Z.B., Gehaltskosten sind von der Anzahl der Mitarbeiter, dem Wachstum der Mitarbeiterzahl, des Durchschnittsgehalts je Abteilung, Durchschnittsgehalt je Position etc. abhängig. Umsatz- und Produktprofitabilität hängen von der industriellen Wachstumsrate, dem Marktanteil, der Produktzusammensetzung und Herstellungskosten ab. Die heutigen OLAP Werkzeuge sollten alle in der Lage sein, diese Geschäftsvorfälle in einem multi-dimensionalen Modell darzustellen und Änderungen in Real-time durchführen zu können.

Diese Beispiel sollen Ihnen zeigen, wie nützlich es für den Analysten sein kann, mit seinen Daten in Form einer multi-dimensionalen Struktur zu arbeiten.

Multidimensionale Architekturen

Verständnis für multidimensionale Daten

Um die OLAP Architektur und deren Nutzen verstehen zu können, muß man zunächst die Art und Herkunft der multidimensionalen Daten verstehen. Multidimensionale Daten sind haben meistens keine 100% Dichte. Das ergibt sich aus der Tatsache, daß von allen theoretisch möglichen Zellen in der Datenbank, normalerweise nur ein kleiner Prozentsatz befüllt sind. Betrachten Sie das folgende reale Beispiel des Vergleichs von theoretischen mit aktuellen Hypercube Größen.

Kostenstellen	Einträge	Zeiträume	Versionen	Theoretische Hypercube Größe	Aktuelle Anzahl befüllter Zellen	Dichte
1.000	200	40	4	32.000.000	800.000	98 %

Obwohl die Tabelle theoretisch bis zu 32.000.000 Zellen enthalten kann, sind nur 800.000 wirklich befüllt. Das hört sich zwar nach einer relativ geringen Anzahl an, ist das ein reales Beispiel und typisch für das Vorkommen von Finanzdaten.

Je mehr Dimensionen dem Würfel hinzugefügt werden, desto größer wird die Dichte. Einfach ausgedrückt heißt das, mit jeder Dimension die wir einfügen, wird nicht unbedingt gleichzeitig ein Wert für jedes Element der neuen Dimension eingefügt. Angenommen wir fügen eine Kundendimension mit 10.000 Kunden in einen Umsatzwürfel ein dann bedeutet das, daß das theoretische Volumen des Würfels um ein 10.000-faches steigt. Doch in Wirklichkeit steigt das Volumen um maximal das 10-fache, wobei 10 die durchschnittliche Zahl der Kunden ist, die jeden Monat, in jeder Region ein Produkt erwerben.

Aufgabe eines OLAP Servers ist es diese Dichte, die übrigens in allen multi-dimensionalen Daten auftritt, zu verwalten und die bestmögliche Performance bereitzustellen.



Multidimensionale Architekturen

Eines der Designziele eines multi-dimensionalen Servers ist es, schnellen linearen Zugriff auf die Daten zu geben, ohne die Abfrage formulieren zu müssen. Die einfachste Abfrage ist ein zweidimensionaler Schnitt aus einem n-dimensionalen Würfel. Ziel ist es die Daten immer gleich schnell zu erhalten, egal wieviel Dimensionen abgefragt werden. In der Realität sind solch einfachen Schnitte ziemlich selten; häufiger sind gemischte Schnitte gefragt, wobei 2 oder mehr Dimensionen als Zeilen oder Spalten verknüpft sind.

Ein weitere Aufgabe des Servers ist es, Zellen zu berechnen. Die beliebteste Berechnung ist die Aggregation, aber auch komplexere Berechnung wie Durchschnittsberechnung und Quotenrechnung werden benötigt. Letztlich ist es das Ziel eine mathematische Gesamtlösung anzubieten, wobei jede Zelle des Würfels von einer anderen abgeleitet werden kann. Dazu werden sämtliche mathematische und statistische Funktionen, einschliesslich der logischen Bedingungen, verwendet.

Die meisten OLAP Server erreichen die schnellen Antwortzeiten in dem die abgeleiteten Werte vorberechnet werden. Diese Technologie kann zwar sehr effektiv sein, macht aber keinen Sinn, wenn der vorberechnete Würfel tausendmal größer ist, als die Datenbasis. Das mag unglaublich klingen, kann aber sehr schnell zur Realität werden, speziell wenn es um eine große Anzahl an Dimensionen geht und die Hierarchien in jeder Dimension sehr tief gehen.

Diese Vorberechnung ist ebenfalls ungeeignet, wenn die Daten real-time geändert werden, wie im Falle von interaktiven Planungs- und Vertriebsanwendungen. Die Analysten wollen nämlich die Auswirkungen der Änderungen sofort sehen. Dabei ist es störend, wenn erst ein Batchlauf zur Neuberechnung ausgeführt werden muss.

Das Abwägen zwischen dem Support von komplexen Berechnungen, schnellem Datenzugriff, minimieren der Datenexplosion und dem Support von real-time updates führte zur Definition der drei OLAP Architekturen: ROLAP, MOLAP und RAP. Diese werden im folgenden erklärt.

Multidimensionale Sichten auf relationale Daten

Einige Hersteller vertreten die Ansicht, daß alle Daten in relationalen Datenbank abgelegt werden sollten. Sie bieten eine multi-dimensionale Sicht auf diese Daten an. Um diese Sicht zu bekommen, ist es erforderlich die Daten in einem sogenannten Star- oder Snowflake-schema. Die am häufigsten verwendete Form speichert die Werte in einer denormalisierten Tabelle, Fact-Tabelle genannt. Eine Dimension wird als Fact-Dimension bestimmt und diese Dimension benennt die Spalten der Fact-Tabelle. Die anderen Dimensionen werden in zusätzlichen Tabellen gespeichert, wobei die Hierarchie mit Hilfe von Child-Parent-Spalten definiert wird. Die Dimensionstabellen werden dann relational mit der Fact-Tabelle gejoint. Damit wird es ermöglicht multi-dimensionale Abfragen zu erstellen.

Die Daten werden mit Hilfe von SQL Abfragen aus der relationalen Datenbank in das Client Werkzeug kopiert. Doch SQL wurde als Access Language für relationale Datenbanken entwickelt, deshalb ist es auch kaum verwunderlich, daß sie recht ungeeignet ist, um



APPLIX iTM1 **Wege zur Datenanalyse**

optimale multi-dimensionale Abfragen zu erstellen. So kann SQL z.B. komplexere Berechnungen über Zeilen ausführen als über Spalten. Das ist weniger ein Defizit von SQL als viel mehr die Tatsache das das relationale Modell erfunden wurde, um Datenbankmanagementprobleme zu lösen. Eines der vordringlichsten Probleme war die Datenbankintegrität und das Absichern von konsistenten Datenupdates. Beim Speichern der Daten in relationalen Tabellen wird ein Datensatz an einem bestimmten Platz und zwar nur da gespeichert. Damit ist die Datenkonsistenz gesichert und die Transaktionen können schneller und effizienter ausgeführt werden. Die Vertreter dieser Architektur argumentieren folgendermaßen:

- Die Daten werden in einer offenen Umgebung gespeichert und sind deshalb einfacher zu selektieren.
- Nachdem die Daten schonmal in der relationalen Datenbank gespeichert sind – Warum soll man sie nicht dort lassen ? Warum duplizieren ?

Doch die Realität ist leider etwas anders:

Obwohl die Fact-Daten in einer relationalen Tabelle gespeichert werden und durch die RDBMS auch selektiert werden können, kann so manche ungewollte Überraschung eintreten. Um die multidimensionale View nutzen zu können, müssen die Daten auf jeden Fall im Star- oder Snowflakeschema abgelegt werden. Das bedeutet aber, das die Daten doch dupliziert werden müssen. Es gibt noch ein paar andere gute Gründe (Performance, Summierung und Organisation der Daten in bestimmte Zeiträume etc.) warum die Daten auf jeden Fall dupliziert werden müssen. Deshalb ist dieser Rückschluss nicht ganz korrekt. Die Mehrzahl der ROLAP Anwendungen liegt deshalb bei einfache Analysen auf große Datenmengen. Einzelhandelsumsatzanalysen ist ein beliebtes Einsatzgebiet. Die Komplexität bei der Definition und Wartung bringt es mit sich, dass nur relativ wenige ROLAP Anwendungen in den Bereichen Datawarehouse und Finanzberichtswesen oder Planung zu finden sind.

Physikalische Multidimensionale Datenbanken

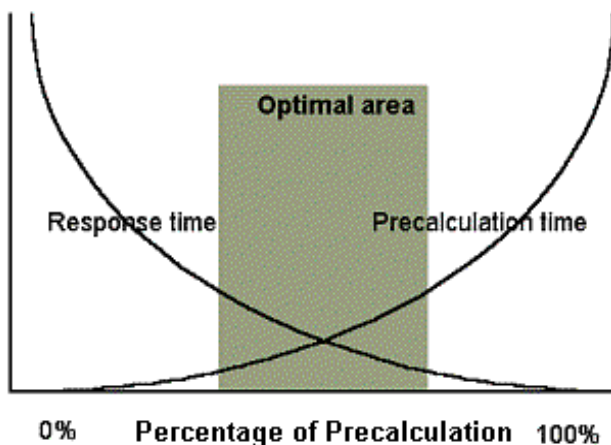
Die beiden nächsten bekannten OLAP Architekturen, MOLAP und RAP, enthalten ihre eigene physikalische multidimensionale Datenbank. Diese Lösungsansätze setzen ihre eigenen Architekturen voraus. Einige dieser Architekturen basieren sogar auf dem relationalen Modell. Alle Hersteller sind der Meinung, daß ihre Architektur einzigartig effizient ist (einige Hersteller, haben sogar ihre Algorithmen patentieren lassen), doch letztlich sind es einige wenige Technologien die von vielen Herstellern genutzt werden. Diese Technologien enthalten:

- Eliminieren der Nullwerte durch Datenkompression
- Indizierung von Pointern auf komprimierte Datenreihen
- Anspruchsvolle Caching Algorithmen

Bevor diese beiden Architekturen besprochen werden, ist es wichtig noch mehr über Datendichte und Datenexplosion und deren Auswirkungen zu erfahren.

Datenexplosion

Es ist nicht immer offensichtlich, daß ein vollständig berechneter Würfel gewöhnlich eintausendmal, und in vielen Fällen sogar tausendmal, größer ist als die Ausgangsbasis (siehe Anhang A). Einige bringen hier dann vielleicht das Argument, daß Speicherplatz zur Zeit schließlich billig zu erwerben ist. Obwohl Plattenplatz zwar relativ preisgünstig ist, doch stellen Sie sich vor was passiert wenn eine 200 MB große Datenquelle auf 10 GB aufgeblasen wird. Die Datenbank hat auf keinem Laptop mehr Platz. Stellen Sie sich vor was geschieht wenn eine Datenquelle von 1 GB auf 50 GB anwächst. Es ist nahezu unmöglich die Datenbank auf einem normalen Desktop Server zu speichern. In beiden Fällen, würde die Ladezeit für die Vorberechnung des Modells Stunden dauern. Das bringt uns zu folgender Erkenntnis: Plattenplatz ist zwar billig, doch die Gesamtkosten für die Vorberechnung können unerwartet hoch werden.



Wie Sie aus der obigen Grafik ersehen können, besteht ein Zusammenhang zwischen Abweichung zwischen der Antwortzeit und der Vorberechnungszeit. Viele OLAP Hersteller setzen voraus das entweder alles oder nichts vorberechnet wird. Wie die obige Grafik zeigt, erreicht man das optimale Ergebnis wenn ein Teil der Daten vorberechnet wird und nur die weniger gefragten oder kleineren Berechnungen während der Laufzeit ausführt.

Mit diesen Kenntnissen kommen wir nun zu den Architekturen MOLAP und RAP

Physikalische Multidimensionale Datenbanken (MOLAP)

MOLAP gründet sich in einer Datenbank, die für die Eigenschaften multi-dimensionaler Daten entwickelt wurde, um dann alle abgeleiteten Werte vorzuberechnen. Die Begründung für die Vorberechnung liegt in den schnellen Antwortzeiten die daraus resultieren und das Datenexplosionen unbedeutend sind, da Plattenspeicher relativ preisgünstig zu erwerben ist. Antwortzeiten sind natürlich meist das wichtigste Kriterium bei der Entscheidungsfindung und damit eben auch einer der Gründe für die große Akzeptanz von MOLAP am Markt.



APPLIX iTM1 Wege zur Datenanalyse

Da abgeleitete Werte vorberechnet und gespeichert werden, muss der Benutzer bei Änderungen in der Datenbasis jedesmal die Datenbank neu berechnen. Das bedeutet, daß der Benutzer warten muß und das können Wartezeiten von Stunden sein, bis die Berechnung beendet ist. Manche MOLAP Engines bieten die Möglichkeit, nur die abgeleiteten Werte neu zu berechnen, die von den Änderungen betroffen sind. Das reduziert die Berechnungszeit zwar um 50 %, ist aber trotzdem nicht für Anwendungen geeignet, in denen sich Daten häufig ändern.

RAP wurde so entwickelt, daß die abgeleiteten Werte auf Anforderung berechnet werden, ohne Vorbereitung. Das vermeidet beides, die lange Berechnungszeit und die Datenexplosion. Um die Berechnung schnell genug durchführen zu können um performante Antwortzeiten zu erhalten, müssen die Daten in den Hauptspeicher geladen werden. Dies beschleunigt die Berechnung erheblich und zeigt sich in sehr schnellen Antwortzeiten, bei einem Großteil der Abfragen.

Desweiteren stehen die berechneten Werte nach einer Anforderung weiterhin für Berechnungen zur Verfügung (solange sich noch gültig sind). Das hat zwei zwingende Vorteile. Erstens, nur die Aggregationen die benötigt werden, werden durchgeführt. Zweitens, in einer dynamischen, interaktiven Umgebung, z.B. im Bereich Planung, sind die Berechnungen immer aktuell. Es entstehen keine Wartezeiten auf notwendige Vorberechnungen nach Änderungen der Datenbasis.

Fraglich ist, ob eine multidimensionale Anwendung jeder Größenordnung in den Speicher geladen werden kann. Die Antwort ist ja, aus mehreren Gründen. Erstens, alle multidimensionalen Datenbanken speichern jede Zahl sehr effizient, normalerweise 10 bis 15 bytes pro Zahl. Wie auch die folgende Grafik über bestehende Anwendungen zeigt, kann ein Server mit 500 MB Speicher über 45 Millionen Werte speichern.

Größe des RAM (MB)	Befüllte Zellen
64	5.592.405
128	11.184.811
256	22.369.621
512	44.739.243

Da RAP nicht vorberechnet, haben RAP Datenbanken typischerweise 10% bis 25% der Größe der Quelldaten. Die Datenquellen benötigen normalerweise ungefähr 50 bis 100 bytes pro Datensatz. Im Regelfall speichert die Datenquelle eine Zahl pro Datensatz der in die multidimensionale Datenbank eingehen soll. Da RAP eine Zahl (plus Indexes) in ungefähr 12 bytes speichert, erreicht die durchschnittliche Größe zwischen RAP und der Datenquelle zwischen $12/100 = 12\%$ und $12/50 = 24\%$.

Der zweite Grund warum Applikationen im allgemeinen in den Speicher geladen werden können, resultiert auf der enorm hohen Sparsity, die zuvor beschrieben wurde. Sparsity ermöglicht es ein Modell mit 5 oder mehr Dimensionen und 45 Millionen aktuelle Werte auf einem 5 GB Server zu halten, das entspricht einem theoretischen Volumen von mehr als 4 Milliarden Zellen. Es gibt wenige Finanzmodelle die diese Datenmengen erreichen. Ein großes Finanzmodell enthält meist nicht mehr als ein paar Millionen Zellen.



APPLIX iTM1 *Wege zur Datenanalyse*

Deshalb vermeidet RAP lange Berechnungszeiten und Datenexplosion. Das ist besonders wichtig für dynamische Anwendungen bei denen sofortiger Zugriff auf Ergebnisse (basierend auf Daten die sich häufig ändern) erforderlich ist.

ANDERE ÜBERLEGUNGEN

Einzelner Hypercube vgl. Multicube

Wie in einer relationalen Datenbank, wobei die Daten typischerweise in zweidimensionalen Tabellen gespeichert werden, bestehen die Daten in einer OLAP Datenbank aus unterschiedlichen Dimensionen. Ein Beispiel: Eine Bank möchte eine Profitabilitätsrechnung pro Kunde und Produkt erstellen. Bestimmte Umsätze müssen nach Kunden, Produkt, Zeit, Ort und Szenario sortiert werden. Aber Kosten wie z.B. Gehälter müssen nach Kostenart, Kostencenter und Zeit sortiert werden. Deshalb muß eine reele multidimensionale Datenbank in der Lage sein Daten mit unterschiedlichen Dimensionalitäten in einer logischen Datenbank zu speichern.

Es ist nicht akzeptabel anzunehmen, daß alle Daten im Modell nach denselben Dimensionen dimensioniert werden. Daten in ein Element zu fassen, das „Kein Produkt“ genannt wird, weil die Daten vielleicht nicht nach Produkt dimensioniert werden, verwirren den Benutzer unnötig. Produkte die keine multicube Architektur, mit der Möglichkeit Würfel logisch zu verbinden, unterstützen, erfordern unnötige Kompromisse der Entwickler, die anspruchsvolle Modelle erstellen wollen. Es macht zwar nicht immer den Datenzugriff unlogisch, aber es kann ernstzunehmende Auswirkungen auf die Datenbankgröße und Antwortzeiten haben.

ZUSAMMENFASSUNG

ROLAP Produkte sind komplex und deshalb sehr kostspielig zu implementieren und zu pflegen. Sie bieten nicht dieselbe Performance, wie andere OLAP Architekturen. Deshalb sind die meisten ROLAP Anwendungen eher für Analysen großer Datenmengen geeignet, bei denen die Integration mit der RDBMS von Vorteil ist.

MOLAP Produkte nutzen die Vorberechnung, um beeindruckende Antwortzeiten zu erreichen. Dies funktioniert wunderbar mit statischen und nahezu statischen Anwendungen, aber die daraus resultierende lange Berechnungszeit machen es nahezu unmöglich dynamische Anwendungen zu definieren. Außerdem macht die Datenexplosion, verursacht durch die Vorberechnung, MOLAP ungeeignet für große Anwendungen mit mehr als 5 Dimensionen.

RAP ist die optimale Architektur für dynamische Anwendungen, die häufige Datenänderungen oder Analysen von Was-wäre-Wenn Szenarien verwalten müssen. Diese Anwendungen schließen das Finanzberichtswesen und Konsolidierungen, Planung, Budgetierung und Produktprofitabilität ein. RAP ist folglich die optimale Architektur, um mobile oder verteilte Umgebungen zu unterstützen. Da RAP nicht vorberechnet, ist es durch die geringen Datenmengen bestens geeignet für den Einsatz auf Laptops und sogar das Verteilen von Anwendungen via E-Mail. Letztlich ermöglicht es die extreme Skalierbarkeit den Einsatz von Anwendungen mit mehr als 5 Dimensionen, wenn MOLAPs wegen der großen Datenmengen nicht mehr anwendbar sind.

ANHANG A – Datenexplosion

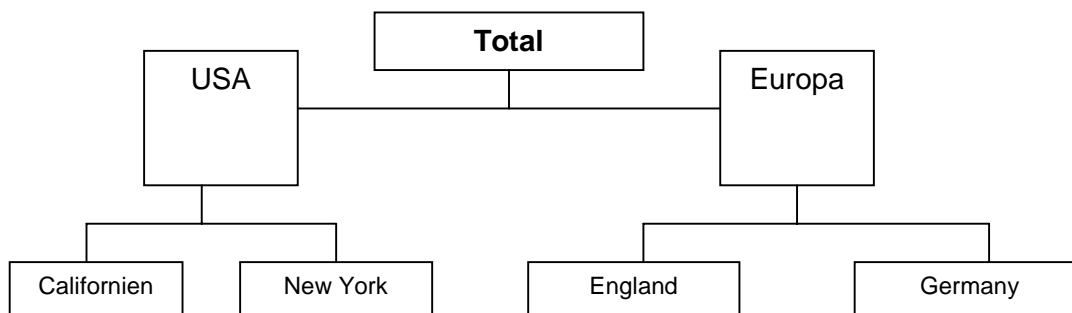
Einführung

Die meisten ROLAP und MOLAP Datenbanken berechnen die abgeleiteten Werte vor. Das bietet den Vorteil schneller Antwortzeiten bei statischen Daten, führt aber zwangsläufig zum Abspeichern hunderter und tausender abgeleiteter Werte für jeden möglichen Wert. Das wird weitläufig als Datenexplosion bezeichnet.

Es kann zwar mathematisch erklärt werden, aber ist meist einfacher an realen Beispielen zu erläutern.

Das Extrem

Je weniger dicht die Daten sind, desto größer die Datenexplosion. Das wird deutlich wenn man sieht was geschieht wenn es im folgenden nur einen Eingabewert gibt. Z.B.



Die Tabelle könnte so aussehen.

Das zeigt auch, dass eine Eingabezahl in 3 insgesamt 3 Werten resultiert.

	Sales
Californien	
New York	1
England	
Deutschland	
USA	1
Europa	
Total	1

Die Zahl die dieses Wachstum ausdrückt, nennt man Compound Growth Faktor (CGF). Der CGF in diesem Beispiel errechnet sich folgendermassen:

$$CGF = \frac{\text{Total no of elements}}{\text{No of. inputelements}} = \frac{3}{1} = 3$$

Wenn Sie Dimensionen verbinden, um eine multidimensionale Datenbank zu definieren, ist der CGF für die Datenbank die Multiplikation der CGFs für jede Dimension.



APPLIX iTM1
Wege zur Datenanalyse

	Limousine	Coupes	Vans	LKWs	Total Autos	Total Zubehör	Total Fahrzeuge
Californien							
New York		1			1		1
England							
Deutschland							
USA		1			1		1
Europa							
Total		1			1		1

Ein einziger Wert resultiert in 9 gespeicherten Werten ! Der Zusammenhang wird in der Formel so deutlich:

$$CGF_{total} = CGF1 * CGF2 * \dots * CGFn$$

Wobei n die Anzahl der Dimensionen ist. In diesem Beispiel sieht das so aus:

$$CGF_{total} = 3 * 3 = 9$$