Semantische Indexierung von Oracle Siebel

Dr. Sebastian Leuoth, dimensio informatics GmbH

Ein Kundenbeispiel aus einer bekannten Schweizer Versicherung zeigt, wie sich mittels semantischer mehrdimensionaler Indexierung eine bestehende Oracle-Siebel-Installation extrem beschleunigen lässt. Dies erhöht die Akzeptanz der Nutzer und verringert gleichzeitig die Last der Datenbank.

„Oracle Siebel CRM ist das weltweit umfassendste CRM." Diese Marketingaussage lässt sich direkt auf die Anforderungen an die Datenbank übertragen. Da ein CRM von den zu verwaltenden Daten lebt und diese wiederum zum Teil stark nutzerspezifisch modelliert sind, muss das CRM ein extrem dynamisches Datenmodell bereitstellen.

Jeder kennt die Herausforderungen der Normalisierung, die sich im Zuge einer Datenmodellierung ergeben. Je höher der Grad der Normalisierung vorangetrieben wird, desto mehr Vorteile ergeben sich zwar hinsichtlich der Reduzierung der Datenredundanz, jedoch steigt in gleichem Maße der Aufwand für die Lieferung der Daten durch die Datenbank an die zu verarbeitende Applikation enorm. In der Praxis kann dieser Umstand die Datenbank und den Nutzer an ihre zeitlichen Belastungsgrenzen bringen.

Vergleichbares lässt sich bei der Siebel-Installation des Kunden feststellen. Die von ihm gewünschte Flexibilität der Anwendung resultiert in komplexen Join-Anfragen, bei denen dreißig und mehr Tabellen keine Seltenheit sind. Endnutzer wissen lediglich um die Auswirkungen dieser Implementierung: lange Antwortzeiten und nicht wiederkehrende Oberflächendialoge.

Um diese Herausforderungen zu lösen, wurde versucht, die vorhandenen Defizite durch klassische Tuning-Maßnahmen zu begrenzen. Nach weiteren Analysen kam der Kunde zu dem Ergebnis, dass nicht alle von Siebel bereitgestellten Funktionalitäten verwendet werden können. Natürlich wurde auch der klassische KIWI-Ansatz („kill it with iron) angewandt, um den Performance-Problemen also mit schnellerer Hardware zu begegnen. *Tabelle 1* gibt einen Einblick in die Abfrageprozesse sowie deren Antwortzeiten innerhalb der Siebel-basierten Vertriebsplattform einer Versicherungsgesellschaft. Das Resultat der langen Wartezeiten führte zu unzufriedenen Mitarbeiter sowie zu Fehlern, Fehlmanipulationen, prozessbezogenen Ausweichmanövern und schlussendlich zur Ablehnung des Systems durch die Mitarbeiter.

Die Herausforderung durch mehrdimensionale semantische Indexierung meistern

Gängige Verfahren zum schnelleren Auffinden von Datensätzen unterteilen den eindimensionalen Suchraum in Intervalle und restrukturieren die indizierten Daten entweder sequentiell oder baumartig. Der bekannteste Vertreter solcher Strukturen ist der „B*"-Baum, von dem eine Vielzahl von Spezialisierungen existiert.

Die sich aufdrängende Analogie zu Inhaltsverzeichnissen und Indizes in Büchern ist in diesem Fall durchaus verständnisfördernd. Hierbei stellt ein Inhaltsverzeichnis einerseits Textintervalle in Form von Kapiteln und Abschnitten dar. Andererseits repräsentiert es deren Anordnung beziehungsweise Reihenfolge innerhalb des Buchs. Im Gegensatz zur inhaltlich orientierten, also semantischen Intervallbildung in Büchern, teilen Datenbank-Indizes den Suchraum lediglich nach formal-mathematischen Kriterien auf (etwa gleich große Intervalle, Metriken etc.) und berücksichtigen die Semantik der Datensätze nicht.

Mit dem Aufkommen sogenannter Nicht-Standard-Datenbankanwendungen, gegen Ende der 1970er-Jahre, trat das Problem mehrdimensionaler Suchräume auf, da für diese Art der Daten oftmals ein linearer Wertebereich für den Primärschlüssel nicht mehr ausreichte. Beispiele hierfür sind Bilddaten, bei denen die Bildpunkte innerhalb des Bilds durch einen X/Y-Koordinatenwert, also durch ein Wertepaar und nicht mehr nur durch einen Einzelwert, eindeutig identifiziert werden.

Auch für diese Art der Daten wurden Indizes erforscht und implementiert: die mehrdimensionalen Datenbank-Indizes. Grundsätzlich folgen diese auch dem Prinzip der Intervallbildung, nun jedoch für mehrere Dimensionen, und der möglichst geschickten Anordnung dieser Intervalle in Datenstrukturen. Hierzu zählen Mehrwegbäume und mehrdimensionale Gitterstrukturen.

Der Vorteil des schnellen Datenzugriffs ist durch diese Art des Vorgehens ersichtlich. Allerdings sorgt das auch hier beibehaltene Kriterium der formal-mathematischen Bildung von Intervallen dafür, dass diese Verfahren keinerlei Bezug zu den inhaltlichen Beziehungen zwischen den Datensätzen

SQL-Statement	Typ	Laufzeit (t)
SQL A	Kunden-Segmentierung	03 h 26 min 06 sek – 04 h 53 min 13 sek
SQL B	Markt-Segmentierung	01 h 58 min 56 sek – 02 h 05 min 10 sek
SQL C	Vertragsanfrage	02 min 47 sek – 08 min 12 sek

Tabelle 1

herstellen. Dies führt dazu, dass diese konventionellen Indizes nur für wenige Dimensionen (< 15) nutzbar sind. Als Beispiel sei hier Oracle Spatial genannt, das zumindest vier Dimensionen unterstützt.

Es stellt sich die Frage, ob es möglich ist, die Beschränkung der Dimensionen zu überwinden. Die Antwort lautet: „Ja, mittels Wissen". Hinter diesem Wissen stecken die zusätzlichen Informationen, die durch die Semantik der Daten gegeben sind. Somit beinhalten die individuellen Daten, deren Indexierung die eigentliche Herausforderung darstellt, gleichzeitig die Lösung hinsichtlich des notwendigen Indexierungsvorgangs, der auch als „Lernvorgang" bezeichnet wird. Ist die Semantik der Daten bekannt, lässt sich dieses Wissen zum Bau einer mehrdimensionalen Index-Struktur nutzen.

Im ersten Schritt des Lernens werden die für die Indexierung relevanten Attribute ausgewählt. Man ermittelt sie durch Analyse der Suchanfragen. Dabei werden die in den „WHERE"-Bedingungen enthaltenen Attribute aus dem vorkommenden Tabellen-Verbund extrahiert. Somit ist der mehrdimensionale Suchraum, in dem jedes Attribut mit seinem (tatsächlich in der Datenbank enthaltenen, nicht dem theoretisch möglichen) Wertebereich eine Dimension aufspannt und in dem die Datensätze später gefunden werden sollen, ausreichend definiert.

Entsprechend dem Aufbau eines klassischen Index werden daraufhin die relevanten Tupel an ein spezialisiertes, KI-basiertes Lernverfahren weitergegeben. Dieses ist in der Lage, eine Art „taxonomisches Wissen" zu erzeugen. Vergleichbar mit der aus der Biologie bekannten Taxonomie des Tierreichs oder der Pflanzenwelt, entstehen Gruppen beziehungsweise Klassen von ähnlichen Objekten. Der wichtigste Unterschied zu den bekannten Taxonomien liegt darin, den Klassen keine Begriffe oder Kategorien zuzuordnen, da es sich um ein vom Menschen nicht überwachtes Lernverfahren handelt. Die Benennung von Objekten oder Objektgruppen stellt somit einen sprachlichen, vom Menschen gesteuerten Vorgang dar. Dagegen erfolgt beim KI-basierten Lernvorgang das Erkennen eines Objekts oder einer Objekt-Gruppierung völlig unabhängig von einer solchen menschlichen Gruppierung.

Das Ergebnis ist eine weltweit einzigartige Technologie zur mehrdimensionalen semantischen Indexierung von Datenbanken. Das bei der Indexierung erlangte und angewendete semantische Wissen wird in einer hochspezialisierten Speicherstruktur innerhalb eines beliebigen Dateisystems abgelegt. Ferner ist das beschriebene KI-Verfahren in der Lage, sich an veränderte Datenbestände anzupassen. Kommen neue Datensätze hinzu oder ändern sich bestehende, wird das Wissen übernommen, ohne dass die bestehenden Datensätze erneut bearbeitet werden müssen.

Neben der Verwaltung des Wissens über die Datensätze speichert das neuartige Lernverfahren die Identifikatoren der in der Datenbank enthaltenen Tupel. Diese werden, wie bei einem Index üblich, als Ergebnis einer Suchanfrage zurückgeliefert und der Datenbank übergeben. Diese zusätzlichen Informationen zum Auffinden der betreffenden Tupel erlauben es dem Optimierer der Datenbank, einen anderen, effizienten Ausführungsplan aufzustellen. Die komplexe Anfrage über mehrere Tabellen, bei denen die unterschiedlichen Filterbedingungen lediglich eine geringe Selektivität pro Tabelle aufweisen, wird durch die Bekanntgabe der Schlüssel der Ziel-Tupel hochselektiv.

Vom Optimierer wird zunächst die Basismenge (aus der Basis-Tabelle) bestimmt und mit der durch den KI-Index bekanntgegebenen Zielmenge verknüpft. Daher reduziert sich der Aufwand zur Bearbeitung der Suchanfrage auf ein Minimum. In der Anfrage enthaltene „WHERE"-Bedingungen bleiben unverändert, da – wie die Praxis zeigt – die hieraus resultierenden Filterschritte die Verarbeitungsgeschwindigkeit kaum beeinflussen.

Integration in die bestehende Infrastruktur des Kunden

Als System-Umgebung wird vom Kunden ein auf AIX basierender Siebel-Applikationsserver betrieben und als Integrationspunkt bereitgestellt, dessen Daten in einer Oracle-Datenbank 11g R2 abgelegt sind. Der Kunde verfügt über ein großes, hochqualifiziertes Team zum Betrieb seiner IT-Infrastruktur und investiert kontinuierlich in dessen Optimierung und Ausbau. Er verfügt über neueste Storage-Lösungen, sodass davon ausgegangen werden kann, dass alle konventionellen Möglichkeiten zu Optimierung der Nutzung bereits im Einsatz sind oder erprobt und verworfen wurden.

Im ersten Schritt erfolgt die Integration in die vorhandene System-Landschaft. Hierzu wurde ein virtueller 64bit-Linux-Rechner (4 Kerne, 16 GB RAM und 500 GB Storage) für den Betrieb der Indexierungskomponente bereitgestellt. Auf diesem laufen die einzelnen Indexierungsinstanzen, sodass mehrere Indizes auf einem System betrieben werden können.

Im zweiten Schritt erfolgt die minimal-invasive Integration in die bestehende Architektur zwischen dem Siebel-Applikationsserver und der Datenbank. Dies ermöglicht einen Zugriff auf der Ebene der SQL-Kommunikation – ohne die Anwendung oder die Datenbank anpassen zu müssen. Eine Beeinflussung von kommunikativen Funktionen wie beispielsweise dem verschlüsselten Datentransfer oder dem redundanten Betrieb der Datenbank ist damit ausgeschlossen.

Das Ergebnis ist die Integration der Indexierungs-Technologie in die bestehende Interaktion und eine Möglichkeit zur Analyse der hochparallelen und hochfrequenten Kommunikation. In der Regel geht die Analyse einer Nutzung voraus, da die zu bildenden Indexkerne definiert werden müssen.

Die technische Umsetzung

Ein Kern enthält alle Tabellen, die relevante Selektionskriterien beinhalten. Im Zentrum ist eine Schlüsselbasis enthalten. Von dieser Tabelle werden die Primärschlüssel im Index abgelegt und dann der Datenbank als Selektionsbasis übergeben.

Nach der Definition des Kerns erfolgt der automatische Aufbau der Index-Instanz. Hierbei werden die Instanz-Informationen dem minimal-invasiven Integrationspunkt bereitgestellt. Dieser ist dann in der Lage, einen Anfragevergleich auf SQL-Basis durchzuführen. Sobald eine SQL-Anfrage von einem verfügbaren Index bedient werden kann, die definierten Verbünde also übereinstimmen, wird die Anfrage als relevant klassifiziert. Nun erfolgt die Extraktion der „WHERE"-Bedingungen. Dabei wird eine Vielzahl unterschiedlicher Operationen pro Attribut unterstützt:

- >, >=, =, <>, >=, >, BETWEEN, NOT ...
- IN, NOT IN, NULL, NOT NULL
- Definierte Funktionen, wie beispielsweise SDE.INTERSECTS()

In Abhängigkeit von den jeweiligen Filterkriterien kann ein Attribut mehrfach mit unter-

schiedlichen Operationen auftreten. All diese formulierten Bedingungen werden extrahiert und an die Indexinstanz weitergeleitet. Im Unterschied zu konventionellen Indizes muss bei diesem Index nicht jede Dimension spezifiziert sein, der Index ist also voll nutzbar; dies gilt selbst dann, wenn nur wenige Spalten Selektionswerte besitzen. Zudem ist die Reihenfolge unerheblich. Die Attribute, die nicht angegeben sind, werden als nicht relevant betrachtet und bei der Anfrage ignoriert.

Sollte die SQL-Anfrage zusätzliche „WHERE"-Bedingungen besitzen, die nicht mit dem Indexkern übereinstimmen beziehungsweise darin nicht enthalten sind, ist der Index dennoch nutzbar. Da die „WHERE"-Bedingungen in der SQL-Anfrage verbleiben, übernimmt der bereits beschriebene Filterschritt in der Datenbank-Verarbeitungspipeline diese Aufgabe und berücksichtigt diese Bedingung. Somit ergibt sich ein äußerst flexibler Indexkern, dessen Konsistenz der Ergebnisse stets gewährleistet ist.

Erfolgskontrolle durch Messungen

Zur Dokumentation der Ergebnisse wurde eine Messanlage entwickelt, die den Benutzer im Fokus hat und die realen Zeiten ermitteln soll, die dieser auch an seinem Arbeitsplatz hätte. Bei dem Aufbau einer solchen Messanlage gilt es einige Herausforderungen zu meistern. Es soll eine repräsentative Messanlage sein, die statistisch signifikante Werte ermittelt, ohne dabei den Betrieb zu beeinträchtigen. Sie ist so zu konzipieren, dass sowohl manuelle als auch automatisierte Tests durchgeführt werden können. Hinsichtlich dieser Bedingungen sollen natürlich keine Verfälschungen entstehen.

Als Ausgangsbasis für die Messungen kommt somit lediglich die Browser-basierte GUI in Frage. Diese ist zwar nicht in der Lage, die Anfrage-/Transferzeiten der Daten sowie die Anzeigezeiten direkt zu ermitteln, sie entspricht aber dem typischen Nutzer-Interface. Eine Bildschirm-Aufzeichnung kann allerdings manuell durchgeführte Interaktionen sehr gut dokumentieren. In Kombination mit den Laufzeit-Informationen aus der Datenbank lassen sich belastbare Werte sammeln. Darüber hinaus verfügte der Kunde über verschiedene Regressionstests, die definierte Aktionen durchführen und die Laufzeiten pro Aufgabe dokumentieren. Somit ergibt sich der beschriebene zweiteilige Testaufbau.

Abbildung 1 zeigt die Ergebnisse in Form eines Diagramms. Der durch den semantischen Index erzeugte Nutzen ist trotz Skalierung deutlich zu erkennen. Die Reduzierung der Wartezeiten ist enorm. Die technische Begründung findet sich direkt in der Darstellung wieder. Die vom Datenbanksystem zu bearbeitenden Datenmengen haben sich gravierend verändert. Während die Oracle-Datenbank in der Ausgangssituation fast 26 GB an Daten gelesen hat, werden nur noch 5 MB benötigt, was dem beschriebenen Minimum an zu verarbeitenden Daten für diese Anfrage entspricht. *Tabelle 2* zeigt die korrespondierenden Ergebnisse zur eingangs aufgeführten *Tabelle 1*.

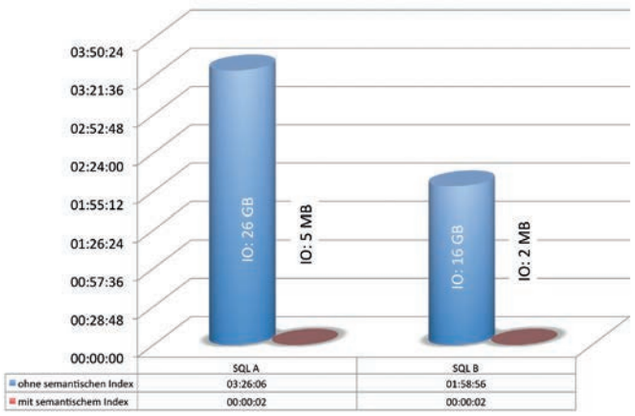

Abbildung 1: Vergleich der Antwortzeiten an zwei Beispielen

SQL-Statement	Typ	Laufzeit (t)
SQL A	Kunden-Segmentierung	03 h 26 min 06 sek – 04 h 53 min 13 sek
SQL B	Markt-Segmentierung	01 h 58 min 56 sek – 02 h 05 min 10 sek
SQL C	Vertragsanfrage	02 min 47 sek – 08 min 12 sek

Tabelle 2

Fazit und Ausblick

Sind Technik und Technologie Synonyme? Im Sprachgebrauch oftmals ja, tatsächlich nein. Technik ist immer durch ihre Funktion definiert, aber es bleibt die Kombinatorik des funktionell Vorhandenen.

Für den gezeigten Ansatz ist keine In-Memory-Technik erforderlich, es kann aber mit Systemen, die In-Memory-Technik verwenden, sehr gut zusammengearbeitet werden. Gemeinsam erzielen beide Systeme noch einmal einen deutlichen Geschwindigkeitsgewinn, denn die zugehörigen Datensätze und der im Index gruppierte Primärschlüssel liegen nicht auf einer Festplatte, sondern werden in komprimierter, eventuell auch spaltenbasierter Form im Hauptspeicher vorgehalten. Es entfallen also auch noch die letzten Plattenzugriffe.

Um die Technologie zur mehrdimensionalen semantischen Suche breiter einzusetzen, arbeitet das Unternehmen des Autors seit Neuestem mit der Firma OdiSys GmbH zusammen, die ein erfahrenes Team von Mitarbeitern hat. Einen der Schwerpunkte bilden Business-Intelligence-Lösungen, wobei hier auf das IBM-Produkt Kongos gesetzt wird. Die Vorteile, die in diesem Artikel an einem Kundenbeispiel wiedergegeben wurden, lassen sich auf eine Vielzahl von BI-Herausforderungen direkt übertragen. Angestrebt ist die Integration der vorgestellten Lösung unmittelbar in die Applikation, um so deren Vorteil mit dem Report-Generator zu kombinieren.

Dr. Sebastian Leuoth
lese@dimensio-informatics.com