

BTW 2007 - Studierendenprogramm

Self-Tuning-Konzepte zur Verwaltung von Bitmap-Index-Konfigurationen

Andreas Lübcke

Dr.-Ing. Schallehn / Prof. Dr. Saake

ITI - Fakultät für Informatik

Otto-von-Guericke-Universität Magdeburg

andreas.luebcke@iti.cs.uni-magdeburg.de



Gliederung

1. Motivation
2. Zielsetzung
3. Bitmap-Indexe und B-Bäume
4. Self-Tuning für Index-Konfigurationen
5. Herangehensweise und Kostenmodelle
6. Anfrageauswertung
7. Evaluierung
8. Zusammenfassung/Future Work
9. Literatur

Motivation

- Immer wichtiger werdende Bedeutung von Datenbanksystemen in der Wirtschaft
- Stetig steigende Anforderungen an Datenbanksysteme
 - Unterstützung für Optimierung bestehender Systeme
 - Weiterentwicklung für neue Systeme
- Optimierung von zeitkritischen Operationen, speziell für große Datenmengen im Bereich Data-Warehouse
- autonome dynamische Datenbanksysteme
- Verbesserung bisheriger Ansätze durch Spezialisierung, bisher nur von Oracle Unterstützung für Such-Bitmap-Indexe

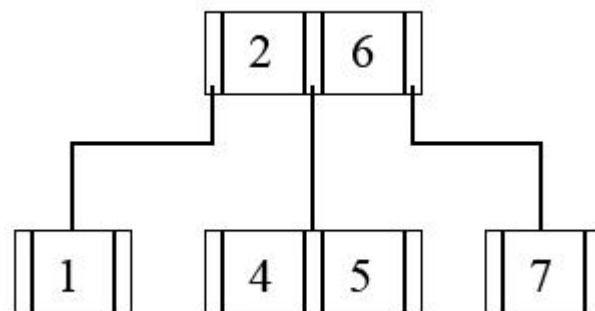
Zielsetzung

- Erweiterung bisheriger Ansätze im Bereich des Self-Tunings
- Unterstützung von Bitmap-Index-Konfigurationen
 - Neue Optimierungsmöglichkeiten
 - Spezialisierung auf Data-Warehouses
- Kostenmodell für Bitmap-Indexe
- Anfrageanalyse für Bitmap-Indexe
- Anwendungsgebiete abstecken sowie Definition von Anforderungen und Grenzen der neuen Ansätze

Bitmap-Indexe und B-Bäume

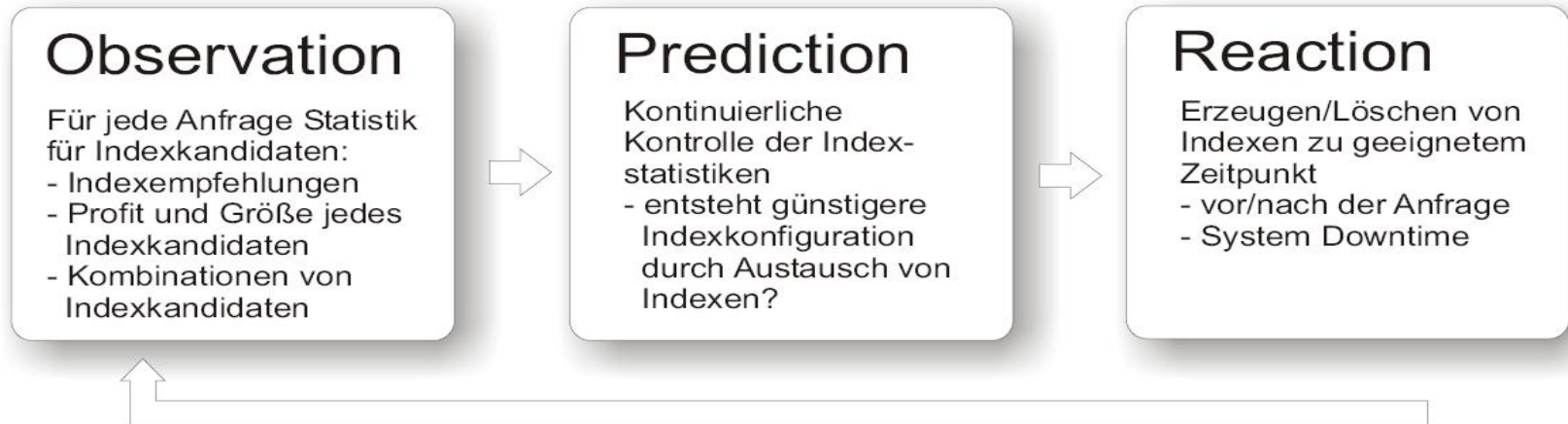
- Bitmap-Index:
 - TID's als Bitvektoren
 - Geringer Platzbedarf
 - Sehr schnell bei geringer Domäne der Attribute
 - Logisch miteinander verknüpfbar
- B-Baum:
 - Suchbaum
 - Operationen mit $O(n \log n)$
 - Anfällig gegen hohe Anzahl von Dimensionen
 - Speicherplatznutzung ca. 50%

TID	B	F	O
rowid ₁	0	0	1
rowid ₂	0	1	0
rowid ₃	0	0	1
rowid ₄	1	0	0
rowid ₅	1	0	0
rowid ₆	0	1	0
rowid ₇	0	0	1
⋮	⋮	⋮	⋮



Self-Tuning für Index-Konfigurationen

- Indexe entscheidender Bestandteil für Anfrageverarbeitung
- Unterstützung und Weiterentwicklung von Design-Tools
- Stetige und dynamische Optimierung von Datenbanksystemen
- Selbstentwicklung der Datenbanksysteme ohne DBA
- Weiterentwicklung für Bitmap-Index-Konfigurationen



Herangehensweise und Kostenmodelle

- Lösung des Index Selection Problems (ISP)

- Bestimmung des Profit eines Indexes:

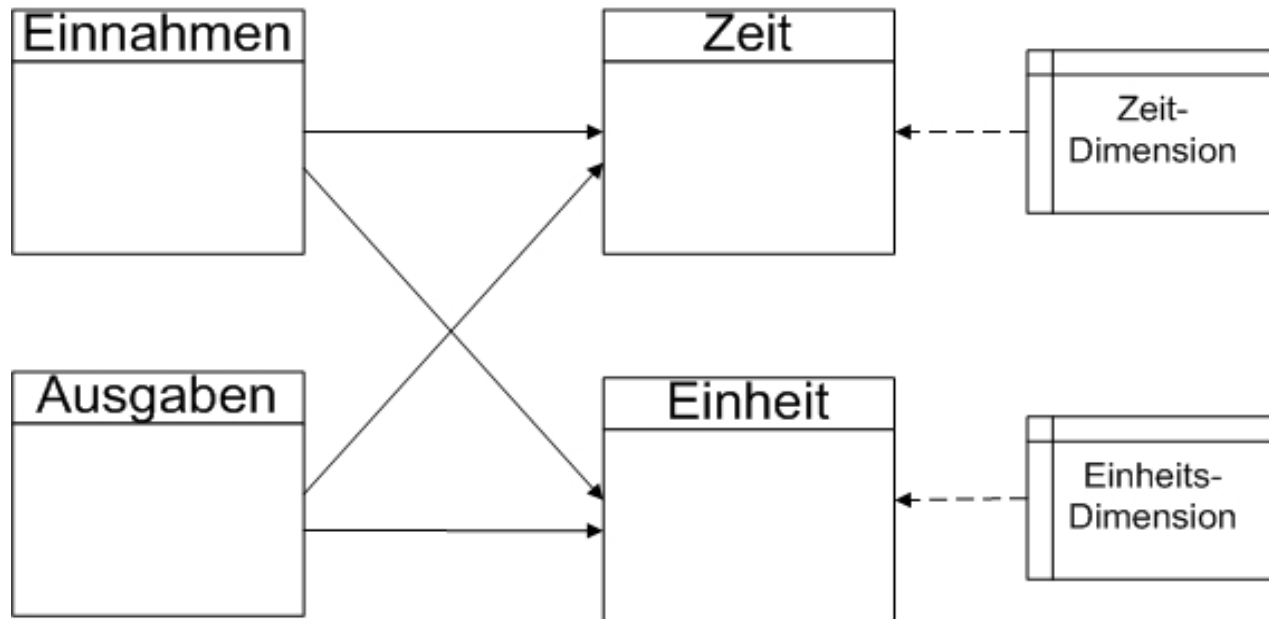
$$profit(Q_k, I_i) = \max\{0, cost(Q_k) - cost(Q_k, I_i)\}$$

- Gesucht ist Index-Konfiguration C (Menge von Indexen I_j), die gegebene Anfragemenge Q_j bestmöglich erfüllt
- Dabei muß gelten: $\sum_{I_j \in C} size(I_j) \leq S$
- Erhöhung des Profits wegen logische Verknüpfbarkeit durch Verteilungsfunktion
- Einbindung hoher Reorganisationskosten durch Wahrscheinlichkeitsabschätzung

Anfrageauswertung

- Für Bitmap-Index als Suchindex:
 - Genügend kleine Domäne des Attributs (ca. 1% von Anzahl der Tupel der Relation)
 - Geringe Wahrscheinlichkeit für Änderung von Tupeln (bei nicht temporären Indexen)
 - Erstellungskosten geringer als Nutzen des Indexes
- Für Bitmap-Index als Joinindex:
 - Einhaltung der Join-Bedingung für beide Join-Partner
 - Erfüllung des Bitmap-Kriteriums für die Attribute, damit auch für das Join-Ergebnis
 - Einbeziehung bestehender Bitmap-Indexe in Entscheidung
 - Größenverhältnisse der Join-Partner

Evaluierung (I)



- Anfrage über alle Relationen (gejoined)
- Gruppierung zum multidimensionalen Cube
- Dimensionen des Cubes und Aggregation als Anfrageergebnis

Evaluierung (II)

- Berechnung von aggregierter Anfrage auf Cube-Operator:
 - Ohne weitere Indexe außer PK-Indexe CPU Cost 57687 (Einheitsmaß von Oracle)
 - Mit B-Tree-Indexen über benötigte Relationen/Attribute CPU Cost 26321; Erstellung zusätzlicher Systemindex vom Optimizer bei erstmaliger Ausführung (mit Sort-Operation drin)
 - Bitmap-Indexe keine Angabe CPU Cost für Index-Scan, aber CPU Cost von 17929 für Umberechnung der Indexe
- Joins für Betrachtungen nicht relevant in diesem Bezug
- Erweiterung der Joinindexe auf Bitmap-Basis

Zusammenfassung/Future Work

- Verfeinerung des gewonnenen Ansatzes durch weitere Fallstudien
- Empfehlungen zur Bestimmung von Verteilungsfunktionen für das Kostenmodell von Bitmap-Index-Konfigurationen
- Verknüpfung mit bisherigen Ansätzen
- Einbindung in bestehende Index- oder Design Tools
- Entwicklungsvorschlag für kommerzielle Datenbankmanagementsysteme
- Möglichkeit zur Mischung der verschiedenen Ansätze denkbar?

Literatur (I)

- [BM72] Bayer, R.; McCreight, E. M.: Organization and maintenance of large ordered indices. Acta Inf., Band 1, S. 173–189, 1972.
- [CDF+01] Chong, E. I.; Das, S.; Freiwald, C.; Srinivasan, J.; Yalamanchi, A.; Jagannath, M.; Tran, A.-T.; Krishnan, R.: B+-tree indexes with hybrid row identifiers in oracle8i. In ICDE '01: Proceedings of the 17th International Conference on Data Engineering, S. 341. IEEE Computer Society, Washington, DC, USA, 2001.
- [CI98] Chan, C.-Y.; Ioannidis, Y. E.: Bitmap index design and evaluation. In SIGMOD'98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, S. 355–366. ACM Press, New York, NY, USA, 1998.
- [Leh03] Lehner, W.: Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden, dpunkt-Verlag/Heidelberg, 1. Auflage, 2003.
- [LSS07] Lühring, M.; Sattler, K.-U.; Schallehn, E.; Schmidt, K.: Autonomes Index Tuning - DBMS-integrierte Verwaltung von Soft Indexen. 2007.
- [SHS00] Saake, G.; Heuer, A.; Sattler, K.-U.: Datenbanken: Konzepte und Sprachen, mitp-Verlag/Bonn, 2. Auflage, 2000.

Literatur (II)

- [SHS05] Saake, G.; Heuer, A.; Sattler, K.-U.: Datenbanken: Implementierungstechniken, mitp-Verlag/Bonn, 2. Auflage, 2005.
- [SSG04] Sattler, K.; Schallehn, E.; Geist, I.: Autonomous Query-driven Index Tuning. In Proc. Int. Database Engineering and Applications Symposium (IDEAS 2004), Coimbra, Portugal, S. 439–448, Juli 2004.
- [Val87] Valduriez, P.: Join indices. ACM Trans. Database Syst., Band 12, Nr. 2, 1987.
- [WHMZ94] Weikum, G.; Hasse, C.; Moenkeberg, A.; Zabback, P.: The COMFORT Automatic Tuning Project, Invited Project Review. Information Systems, Band 19, Nr. 5, 1994.
- [ZRL+04] Zilio, D. C.; Rao, J.; Lightstone, S.; Lohman, G. M.; Storm, A.; Garcia-Arellano, C.; Fadden, S.: DB2 Design Advisor: Integrated Automatic Physical Database Design. In VLDB '04, 2004.



Das war's

Vielen Dank für Ihre Aufmerksamkeit!
Haben Sie Fragen?