

Multidimensionales Datenmanagement in mobilen Umgebungen

Ilias Michalarias

Institut für Informatik
Freie Universität Berlin
Ilias.Michalarias@fu-berlin.de

Abstract: Immerwachsende Datenmengen und ständige Datenänderungen verändern die Rolle von Data-Warehousing für ein konkurrenzfähiges Unternehmen. Dabei wird Data-Warehousing zunehmend nicht nur für strategisches sondern auch für operatives Management eingesetzt. Hierbei stehen die Unternehmen der Aufgabe gegenüber ein Real-Time Data-Warehousing zu realisieren. Eine wesentliche Anforderung dabei ist die Teilnahme tragbarer, mobiler Geräte in solchen Systemverbunden. Das Design mobiler Informationssysteme muss die inhärenten Grenzen der Handheld-Geräte und die Unzuverlässigkeit von drahtlosen Netzwerken berücksichtigen.

1 Einführung

Fortschritte in der drahtlosen Technologie ermöglichen eine umfangreiche Einführung mobiler Informationssysteme. Darüber hinaus werden mobile Geräte ständig kleiner, billiger und leistungsfähiger. Laptops können heutzutage problemlos große Datenbanken speichern und verarbeiten, was einst nur auf Mainframes möglich war. Die Leistung und die Kapazität von PDAs steigen kontinuierlich, so dass sie bereits heute in der Lage sind, eine Reihe von Aufgaben eines Datenbankmanagementsystems sehr effizient zu erledigen.

Unternehmen setzen mobile Anwendungen ein, da dadurch erhebliche geschäftlichen Vorteile realisiert werden können. Heute sind allgegenwärtige (ubiquitous) Daten in einem wettbewerbsorientierten Umfeld von entscheidender Bedeutung für den Unternehmenserfolg. Daten werden in einem sehr schnellen Tempo erzeugt, und folglich muss auf deren Basis eine Entscheidung zeitnah getroffen werden.

Deshalb ist Mobilität, sowohl von Personen als auch von Geräten, eine unentbehrliche Komponente für hochqualifizierte Arbeitskräfte. Ein weitverbreiteter Ansatz ist dabei jeden Ort zu jeder Zeit zu einem Büro zu verwandeln. So kann ein Manager, während einer Reise im Auto oder im Flugzeug, oder einfach am Strand sitzend, ein Job-Büro mit Zugang zu den notwendigen Informationen ins Leben rufen, um seine Aufgaben zu erledigen. Allerdings kann die Netzwerkverbindung unter diesen Bedingungen nicht immer garantiert werden, deswegen sind zusätzliche Offline-Lösungen erforderlich.

Data-Warehouse (DW) ist ein traditionelles Instrument zur Entscheidungsunterstützung. Basierend auf multidimensionalen Modellen verarbeiten DWs große Volumen von ag-

gregierten, hierarchischen Daten. Immerwachsende Datenmengen und ständige Datenänderungen verändern die Rolle von DW-Systemen.

Das Forschungsfeld *mobil OLAP* (mOLAP) kombiniert die beiden genannten Forschungs- und Anwendungsgebiete. Der Ausdruck mOLAP führt alle notwendigen Technologien für mobile Informationssysteme, welche multidimensionale Datenzugriffe ermöglichen, zusammen.

2 Problem Beschreibung

Ein Daten-Broadcast ist in drei verschiedenen Modi möglich: *Pull-basiert*, *Push-basiert* und *hybrid*. In Pull-basierten Systemen senden Clients explizite Abfragen an den Server. Wenn mehrere Clients die gleichen Daten zur etwa gleichen Zeit anfordern, ist der Server in der Lage diese Abfragen mit nur einem Broadcast zu beantworten. In Push-basierten Systemen, setzt der Server eine Punkt-zu-Mehrpunkt Kommunikation ein. Daten werden vom Server an die Clients ohne explizite Client-Abfragen gesendet. Der Server folgt einem Sendeplan, der sowohl die Reihenfolge als auch die Häufigkeit der Sendungen bestimmt. Hybrid-Systeme kombinieren Push- und Pull, indem sie häufig angefragte Daten einem Push-Kanal und selten benötigte Daten einem Pull-Kanal zuordnen.

Effizientes und robustes mOLAP ist mit konventionellem Daten-Broadcast nicht möglich, unabhängig vom Modus. Dies hat die folgenden Ursachen:

1. *Semantik*: Der Inhalt von Broadcast-Daten wird nicht berücksichtigt. Allerdings sind mehrdimensionale Daten semantisch miteinander verbunden.
2. *Thin/Fat Clients*: Bei Thin Clients sind die übermittelten Daten für eine direkte Nutzung aufbereitet. Da mobile Geräte immer leistungsfähiger werden, ist eine Lastverteilung eine effizientere Möglichkeit. Lokale Verarbeitung kann sowohl die Performance des gesamten Systems verbessern, als auch die Offline-Funktionalität erweitern.
3. *OLAP-Endbenutzer Verhalten*: OLAP-Endbenutzer navigieren in der Regel durch die angefragten Daten und führen typische OLAP-Operationen aus, z.B. Roll-up oder Drill-Down. Dies muss beim Design des Systems berücksichtigt werden, um unnötige Sendungen zu vermeiden.
4. *Offline-Funktionalität*: Aufgrund der Annahme eines Thin-Clients fehlt der Offline-Aspekt. Das gesamte Konzept von mOLAP Systemen basiert allerdings auf der Anforderung der Offline-Funktionalität. Dies ist von entscheidender Bedeutung, weil mobile Geräte nicht ständig mit einem Netzwerk verbunden sind.

Abgesehen von den obigen Ausführungen, weisen die verschiedenen Broadcast-Modi zusätzliche Mängel auf. Pull-basierte Systeme sind bekanntlich nicht skalierbar, da sie nicht mit wachsender Anzahl von Clients und somit wachsender Anzahl der eingehenden

Abfragen mithalten können. Obwohl sie eine begrenzte Anzahl von Endnutzern effizient verarbeiten können, sind sie ungeeignet, wenn diese Anzahl steigt.

Skalierbarkeit wird durch Push-basiertes Broadcast erreicht. Dennoch, ist ein herkömmliches Push-basiertes Broadcast auch ungeeignet für mOLAP. Die Gründe hierfür sind:

1. *Anzahl der Datenobjekte*: Die Anzahl der in mOLAP zu sendenden Daten ist nicht begrenzt, dies ist ein Widerspruch zur fundamentalen Annahme von bestehenden Push-basierten Systemen. mOLAP Daten sind Antworten auf Abfragen. Da die Anzahl der möglichen Abfragen praktisch unendlich ist, kann ein typischer Sendeplan nicht eingesetzt werden.
2. *Grösse der Datenobjekte*: Datenobjekte in bestehenden Push-basierten Systemen benötigen relativ wenig Speicherplatz, z.B. Web-Seiten. Multidimensionale Würfel (Cubes) sind jedoch Datenobjekte, die um einen erheblichen Faktor größer sind.

Außerdem verbrauchen Push-basierte Systeme unnötig viel Bandbreite, wenn die Zahl der eingehenden Abfragen relativ gering ist.

Hybrid-Systeme leiden mehr oder weniger an den oben genannten Ausführungen. Obwohl sie theoretisch selbst-adaptiv sind, indem sie dynamisch "heiße" Daten in Push-Kanäle und "kalte" Daten in Pull-Kanäle unterteilen, bedeutet dies in der Praxis eine zunehmende Komplexität und ein erhöhter Wartungsaufwand.

Offensichtlich ist effizientes und robustes mOLAP nicht mit bestehenden Broadcast-Architekturen zu realisieren. mOLAP erfordert eine völlig andere Architektur, die die Eigenschaften der multidimensionalen Daten und das Verhalten der OLAP-Endbenutzer berücksichtigt.

Eine mOLAP Architektur sollte Effizienz, Skalierbarkeit, Anpassungsfähigkeit bei sich ändernden Belastungen, Client-Server Lastverteilung, Online- und Offline-Funktionalität und Unabhängigkeit von der physikalischen Struktur des Daten-Würfels aufweisen.

3 Beitrag

In Anbetracht der oben genannten Probleme, ist der wesentliche Beitrag dieser Arbeit die Präsentation und Evaluierung von *FCLOS* (*Force CLustering OLAP Scheduler*). *FCLOS* ist ein mobiles Informationssystem, welches explizit für mOLAP konzipiert ist. Die folgenden Features besitzt *FCLOS*:

- *Offline-Funktionalität*: *FCLOS* Clients erhalten in der Regel Sub-Cubes, die Daten auf die untersten Ebene aller beteiligten Dimensionen enthalten. Dies geschieht, weil der Server die eingehenden Abfragen dem *DCL* (*Data Cube Lattice*) [BPT97] zuordnet, wodurch weniger detaillierte (aggregierte) Ebenen (*hDCL* (*Hierarchical Data Cube Lattice*)) nicht gesendet werden. Im Falle von begrenzter oder gar keiner

Konnektivität, können Endnutzer weiterhin erweiterte Offline-Funktionalität benutzen.

- *Skalierbarkeit*: Die Performanz von *FCLOS* verschlechtert sich nicht, wenn die Anzahl der Clients steigt. Im Gegenteil, aufgrund der Nutzung von Subsumption (ein Sub-Cube kann von einem anderen Sub-Cube abgeleitet werden, wenn es einen Pfad zwischen den entsprechenden Knoten in der *DCL/hDCL* gibt) verbessert sie sich sogar.
- *Anpassungsfähigkeit*: *FCLOS* reagiert ausdrücklich auf Client Abfragen und beinhaltet keinen festen Sendeplan. In diesem Sinne kann *FCLOS* als ein Pull-basiertes System betrachtet werden. Allerdings ähnelt die Subsumption-basierte Bearbeitung der eingehenden Abfragen und die umfassende Broadcast-Nutzung, einem Push-basierten System. Einerseits, wenn die Belastung relativ gering ist, zeigt *FCLOS* die Vorteile von Pull-basierten Systemen, hierbei verschwendet kein fester Sendeplan die verfügbare Bandbreite. Andererseits, aufgrund der Nutzung von Subsumption, weist *FCLOS* bei steigender Belastung das Hauptmerkmal der Push-basierten Systeme auf: Skalierbarkeit.
- *Lastverteilung*: *FCLOS* bietet verschiedene Möglichkeiten der Client-Server Lastverteilung an. Die grundlegende Idee in *FCLOS* ist, einen Sub-Cube an den Client zu senden, der entweder genau der angefragte oder alternativ einen Sub-Cube-Vorfahren darstellt. Dieser Sub-Cube-Vorfahre kann lokal verarbeitet werden um die benötigte Information zu liefern. Da es mehr als einen Vorfahren geben kann, hat die Wahl des übermittelten Vorfahren einen direkten Einfluss auf die Last, die auf dem Client entsteht. Darüber hinaus, unabhängig vom logischen Sub-Cube, verursacht die Wahl der physikalischen Struktur auch eine (indirekte) Lastverteilung. Der Grund dafür ist, dass die Antwortzeit auf lokalen Daten maßgeblich von der Kompression und der Indizierung der physikalischen Struktur abhängig ist. Stärker komprimierte physikalische Strukturen erzeugen zwar weniger Verkehr, aber verursachen eine erhöhte lokale Verarbeitung. Schließlich ist es möglich eine Lastverteilung mit verschiedenen Zuordnungen der eingehenden Abfragen zu den Aggregation Lattices zu erreichen.
- *Effizienz*: Umfangreiche experimentelle Evaluationen zeigen dass, *FCLOS* alle bisherigen Systeme in den wichtigsten Kriterien übertrifft: Antwortzeit, Energieverbrauch und Netzwerkverkehr. Der Grad der Optimierung ist natürlich maßgeblich vom verwendeten Datensatz, der Abfrageverteilung und dem Netzwerk abhängig. Im Standard-Szenario, ein realer Data Mart, weist *FCLOS* eine Optimierung von mehr als 65 % in den relevanten Kriterien auf.
- *Abfrage-Verteilung Unabhängigkeit*: Um die Anzahl der sendenden Datenobjekte zu reduzieren, werden eingehende Abfragen den entsprechenden Sub-Cubes des *DCL* zugeordnet. Dies führt zu einem grobkörnigen Abfrage-Modus, und somit bleibt *FCLOS* robust bei unterschiedlichen Abfrage-Verteilungen.
- *Physikalische Struktur Unabhängigkeit*: Scheduling Entscheidungen sind von der Größe der Datenobjekten völlig unabhängig. *FCLOS* berücksichtigt Dimensiona-

lität statt Größe, die unabhängig von der physikalischen Implementierung ist. Dies erlaubt nicht nur die Auswahl zwischen den verschiedenen physikalischen Strukturen des Daten-Cubes, sondern macht die gesamte Architektur flexibel. Jede künftige physikalische Struktur eines Daten-Würfels kann nahtlos integriert werden.

Im Rahmen der mOLAP Architektur, wurde der *m-Dwarf* entwickelt, eine physikalische Struktur des Daten-Würfels, die explizit für mOLAP konzipiert ist und Daten komprimiert. *m-Dwarfs* werden zwar von *FCLOS* eingesetzt, können allerdings auch in jedem mOLAP System integriert werden.

Ein weiterer wichtiger Beitrag der Dissertation ist das *QDGV* (*Query and Disseminate under the Global View*) Protokoll [MB07], das Szenarien, in denen Clients ausschließlich einen Server anfragen, mit zusätzlichen Ad-hoc-Verbindungen erweitert. Leider, lassen sich die fundamentalen Ideen aus dem mOLAP Client-Server Bereich nicht direkt in mobilen Ad-hoc-Netzwerken umsetzen. Dies ist geschuldet den folgenden Gründen: ein gemeinsames drahtloses Medium (keine exklusiven drahtlosen Kanäle); die Abfrage-Verbreitung muss entworfen werden (keine zentralen Einrichtungen); intelligentes Scheduling kann keine erhebliche Optimierung liefern (keine kumulierten Abfragen in einem Knoten). *QDGV* ist ein einfaches und adaptives Protokoll für die *Abfrage-Verbreitung* und *Datenübermittlung*. *QDGV* ist jedoch kein allgemeines Protokoll für die Daten-Verbreitung, weder in Bezug auf die laufende Anwendung, noch in Bezug auf die Netzwerk-Topologie. Es ist explizit für mOLAP Ad-hoc-Abfragen, unter realistischen Annahmen konzipiert. *QDGV* beschäftigt sich nicht mit Cross-Layer-Techniken und ist völlig unabhängig von dem MAC- und Ad-hoc-Routing-Protokoll.

Abbildung 1 zeigt eine mOLAP-Architektur basierend auf *FCLOS* für Client-Server Sendungen von mehrdimensionalen Würfeln und auf *QDGV* für den Ad-hoc-Austausch von Würfeln zwischen mobilen Clients. *FCLOS* und *QDGV* können sowohl getrennt als auch gleichzeitig eingesetzt werden. Wegen Platzmangels befasst sich der Rest dieses Dokuments nicht mit *QDGV*.

4 Experimentelle Ergebnisse

Dieser Abschnitt enthält einen Teil der experimentellen Ergebnisse für die Bewertung von *FCLOS*. Das System wird mit anderen mOLAP-Systemen mit Hilfe eines C/C++-Simulators evaluiert. Eine gründliche Beschreibung des verwendeten Simulationsmodells, sowie umfangreiche experimentelle und analytische Ergebnisse befinden sich in [MOL09].

4.1 Simulationsumfeld

Mobile Clients, zufällig in einem quadratischen Raum platziert, fragen ein DW ab. Abfragen werden in regelmäßigen Abständen über ein 802.11 drahtloses Netzwerk gesendet. Wenn eine passende Antwort empfangen wird, sendet der Client nach einem gleichmäßigen

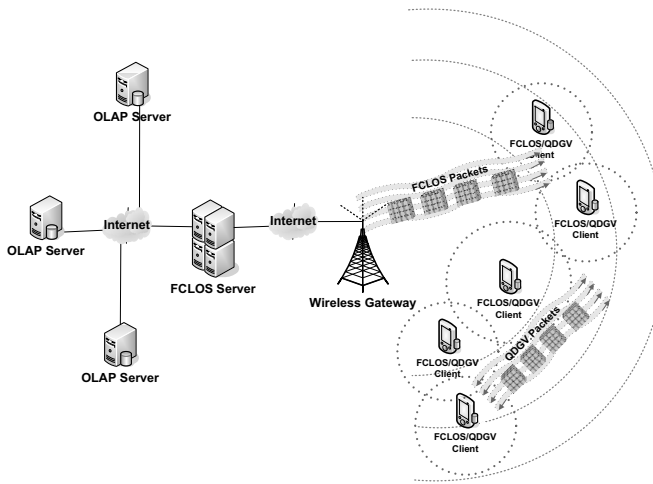


Abbildung 1: Eine mOLAP Architektur basierend auf FCLOS und QDGV

verteilten Zeit-Intervall von $[\frac{t}{2}, t]$ s eine neue Abfrage.

4.2 Speicherplatz-Einsparungen des m-Dwarf

Zwei Datenbanken, eine halb-synthetische und eine reale, werden benutzt um die *m-Dwarfs* zu bewerten. Ein Vergleich zwischen *m-Dwarfs* und STs (summary tables) und (*fcd* (fully coarse-grained Dwarfs)) wird durchgeführt [SRDK02].

Abbildung 2 zeigt die Einsparungen der *m-Dwarfs* gegen STs und *fcd*, anhand der realen Datenbank. Hierbei wird offensichtlich, dass die Datenstruktur *m-Dwarf* weniger Speicherplatz für jeden Sub-Cube im Vergleich zu STs benötigt. Jedoch für 10 % der Sub-Cubes benötigt der entsprechende *fcd* weniger Platz als der *m-Dwarf*. Die Einsparung ist abhängig von der Dimensionalität des Sub-Cubes: die Reduktion steigt mit zunehmender Dimensionalität.

Es ist jedoch wichtig zu betonen, dass die Kompression des *m-Dwarf* nicht nur auf Kosten von fehlenden aggregierten Werten, sondern auch von fehlender Indizierung erfolgt. Während in traditionellen Desktop-DW-Systemen, Kompression ohne Indizierung praktisch nutzlos ist, kann argumentiert werden, dass für die Zwecke der Datenübermittlung über ein drahtloses Medium, Indexierungsanforderungen gelockert werden können. Zusätzliche Index-Strukturen können lokal, nach Eingang der Daten, erzeugt werden. In der Tat, erfordert die Konvertierung eines *m-Dwarf* in den entsprechenden *Dwarf* oder *fcd* nur ein einfaches Scannen der *m-Dwarf*-Struktur.

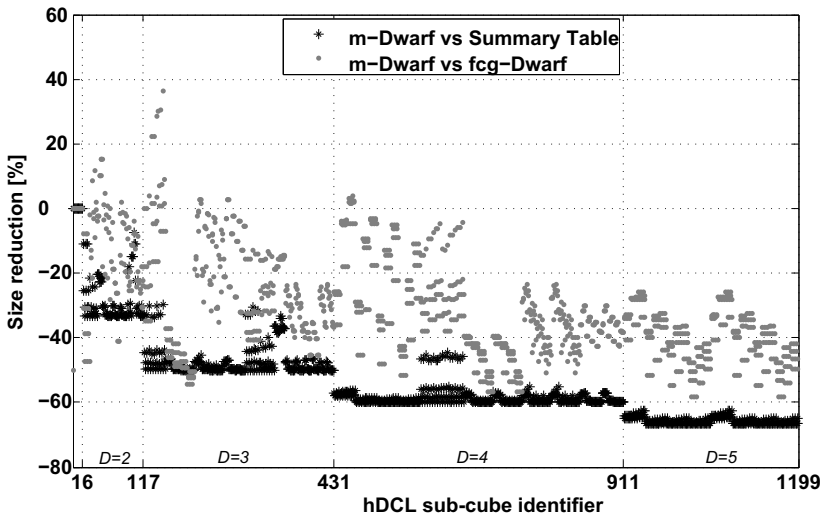


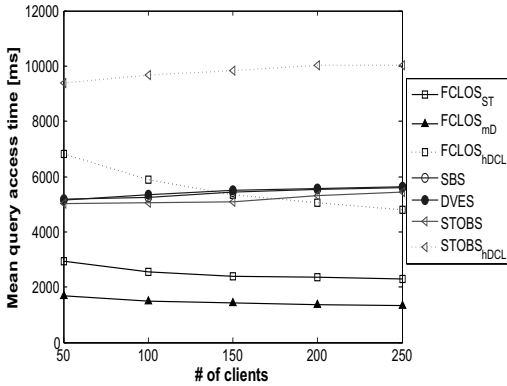
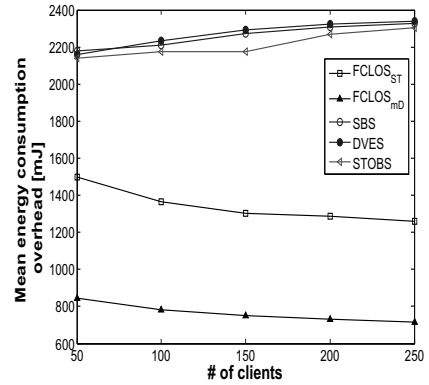
Abbildung 2: m-Dwarf: Speicherplatz-Einsparungen anhand einer realen Datenbank

Tabelle 1: Überblick der implementierten mOLAP-Systeme

System	Scheduler	Physikalische Struktur	Abfrage-Zuordnung
$FCLOS_{ST}$	$FCLOS$	ST	DCL
$FCLOS_{mD}$	$FCLOS$	$m\text{-Dwarf}$	DCL
$FCLOS_{hDCL}$	$FCLOS$	ST	$hDCL$
SBS	$LTSF$	ST	DCL
$DV - ES$	$LTSF$	ST/Dwarf	DCL
$STOBS$	$\frac{R \times W}{S}$	ST	DCL
$STOBS_{hDCL}$	$\frac{R \times W}{S}$	ST	$hDCL$

4.3 Experimentelle Evaluation von FCLOS

Dieser Abschnitt bietet eine experimentelle Bewertung der mOLAP-Systeme. Um die Wirkung des Scheduling-Algorithmus zu isolieren, wurde $FCLOS_{ST}$ implementiert, der STs sendet, um einen direkten Vergleich mit $STOBS$ [SC04], und SBS [SC02] zu ermöglichen. $FCLOS_{mD}$ sendet $m\text{-Dwarfs}$. In Bezug auf die Abfrage-Zuordnung, wurden die Erweiterungen von $FCLOS$ und $STOBS$ (DCL Zuordnung), $FCLOS_{hDCL}$ und $STOBS_{hDCL}$ implementiert, die Abfragen an die jeweilige $hDCL$ zuordnen. Tabelle 1 gibt einen Überblick der evaluierten mOLAP-Systeme.

Abbildung 3: Durchschnittliche Antwortzeit (T_{all})Abbildung 4: Durchschnittlicher Energieverbrauch (E_{all})

4.3.1 Antwortzeit

Abbildung 3 zeigt die Überlegenheit von $FCLOS_{ST}$ gegen seine Konkurrenten. Erwartungsgemäß zeigen $STOBS$ und SBS praktisch die gleiche Leistung aufgrund ihrer Ähnlichkeit. Das neuartige Konzept von $FCLOS_{ST}$ erreicht nicht nur eine Reduzierung von mehr als 50%, sondern stellt auch seine Skalierbarkeit unter Beweis. Wenn die Anzahl der Clients wächst, nimmt auch die Zahl der eingehenden Anfragen zu. $FCLOS_{ST}$ kann dann größere Broadcast Clusters ($BCLs$) bilden und somit mehr Clients pro Sendung bedienen. Im Gegenteil, nutzen seine Vergleichspartner, die traditionellen Pull-Basierte Systemen ähneln, die Subsumptions nicht optimal aus, und zeigen somit eine erhöhte Antwortzeit.

Darüber hinaus, wird bei $FCLOS_{mD}$ die Wirkung von m -Dwarf Sendungen ersichtlich. $FCLOS_{mD}$ weist eine weitere Reduzierung von etwa 40% im Vergleich zu $FCLOS_{ST}$ auf. $DV-ES$ zeigt trotz der Sendungen komprimierter Strukturen, eine etwas schlechtere Antwortzeit als SBS . Dies kann durch die Tatsache gerechtfertigt werden, dass die gemeinsame Scheduling-Komponente eine Metrik nutzt, die direkt von der Größe des Sub-Cubes beeinflusst wird. Daher verursacht die Integration einer neuen physikalischen Struktur unterschiedliche Scheduling-Entscheidungen und die Wirkung der übermittelten Struktur kann nicht isoliert betrachtet werden. Im Gegenteil, stützt sich die Scheduling-Komponente von $FCLOS$ auf die Dimensionalität des Sub-Cubes, die unabhängig von der physikalischen Implementierung ist.

Schließlich zeigt Abb. 3, dass mobile Clients von DCL -Zuordnung profitieren, unabhängig von der eingesetzten Scheduling-Komponente. Die $hDCL$ -Zuordnung verdoppelt fast die durchschnittliche Antwortzeit T_{all} , und zwar sowohl für $FCLOS$ als auch für $STOBS$.

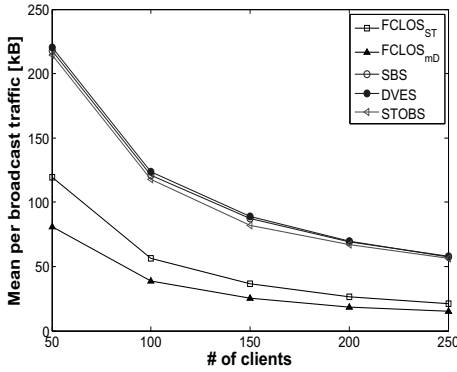


Abbildung 5: Durchschnittlicher pro Abfrage generierter Netzwerkverkehr (Tr_q)

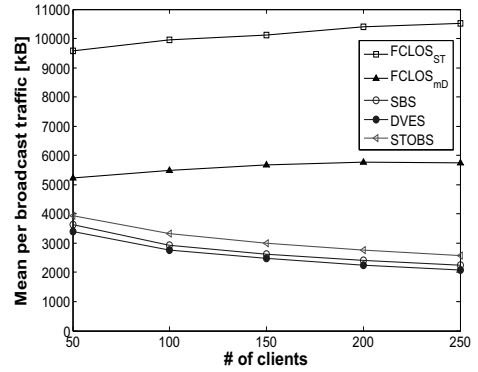


Abbildung 6: Durchschnittlicher pro Broadcast generierter Netzwerkverkehr (Tr_b)

4.3.2 Energieverbrauch

Der Energieverbrauch ist eine entscheidende Metrik, weil die Maximierung der Betriebszeit von mobilen Geräten eine fundamentale Anforderung ist. Die Ergebnisse der Abb. 4 sind ähnlich zu denen der Antwortzeit. Der Grund dafür ist, dass die beiden Metriken leistungsbezogen sind. Je mehr ein Client auf die Antwort wartet, desto mehr Energie wird verbraucht.

4.3.3 Netzwerkverkehr

Abb. 5 zeigt den pro Abfrage generierten Verkehr. Jede Sendung (Broadcast) dient mehreren Anfragen. Die Überlegenheit von $FCLOS_{ST}$ ist offensichtlich: der Verkehr wird um mehr als 50% im Vergleich zu SBS verringert. $FCLOS_{mD}$ reduziert den Verkehr weiter. Bemerkenswert ist es auch, dass sämtliche Ansätze gut skalieren. Dies rechtfertigt die Subsumption-basierte Scheduling-Komponente aller Ansätze.

Im Gegensatz zu Abb. 5 wird in Abb. 6 der Verkehr pro Broadcast dargestellt. Hier kann ein völlig anderes Verhalten beobachtet werden. Die $FCLOS$ Ansätze senden deutlich mehr Daten pro Sendung. Trotzdem gibt es keinen Widerspruch zu den Ergebnissen der Abb. 5. Die Intuition hinter $FCLOS$ ist genau die: größere Daten-Würfel werden gesendet und somit werden mehr Clients bedient und insgesamt werden weniger Sendungen benötigt. Tatsächlich, hat $FCLOS_{ST}$ an diesem Experiment fast dreimal weniger Sendungen (Broadcasts) als seine Konkurrenten gebraucht.

5 Zusammenfassung und Ausblick

In dieser Arbeit wird argumentiert, dass effizientes und robustes mOLAP nicht durch herkömmliches Daten-Broadcast möglich ist. In diesem Zusammenhang wird *FCLOS* vorgestellt, eine speziell für mOLAP gedachte Broadcast-Architektur. Die Leistung von *FCLOS* wurde sowohl analytisch als auch experimentell geprüft und übertrifft existierende Ansätze in allen relevanten Kriterien, mit einer durchschnittlichen Optimierung von mehr als 65%. Außerdem, weist *FCLOS* Skalierbarkeit und Anpassungsfähigkeit auf: eine Reaktion auf explizite Client-Abfragen ist möglich, wie bei einem Pull-basiertem System. Allerdings ähnelt die Subsumption-basierte Bearbeitung der eingehenden Anfragen und die umfassende Nutzung von Broadcasts eher einem Push-basierten System.

Literatur

- [BPT97] Elena Baralis, Stefano Paraboschi und Ernest Teniente. Materialized Views Selection in a Multidimensional Database. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, Seiten 156–165. Morgan Kaufmann Publishers Inc., 1997.
- [MB07] Ilias Michalarias und Christian Becker. Multidimensional querying in wireless ad hoc networks. In *Proceedings of the ACM symposium on Applied computing, Special Track on Database Theory, Technology, and Applications (SAC-DTTA)*, Seiten 529–530, 2007.
- [MOL09] Ilias Michalarias, Arkadiy Omelchenko und Hans-Joachim Lenz. FCLOS: A client-server architecture for mobile OLAP. *Data Knowl. Eng.*, 68(2):192–220, 2009.
- [SC02] Mohamed A. Sharaf und Panos K. Chrysanthis. Semantic-based delivery of OLAP summary tables in wireless environments. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, Seiten 84–92, New York, NY, USA, 2002. ACM Press.
- [SC04] M. Sharaf und P. Chrysanthis. On-Demand Data Broadcasting for Mobile Decision Making. *Mobile Networks and Applications*, 9:703–714, 2004.
- [SRDK02] Yannis Sismanis, Nick Roussopoulos, Antonios Deligianannakis und Yannis Kotidis. Dwarf: Shrinking the Petacube. *ACM SIGMOD International Conference on Management of Data*, Seiten 464–475, 2002.



Ilias Michalarias wurde am 22. Dezember 1981 in Athen, Griechenland geboren. Ilias erhielt seinen M.Sc. in Elektro- und Computertechnik von der Nationalen Technischen Universität Athen im Jahr 2004 und 2008 erhielt er den Ph.D. in Informatik an der Freien Universität Berlin. Seine Forschungsinteressen umfassen Data Warehousing, ETL, OLAP- und Daten-Management in mobilen und verteilten Umgebungen. Seit seinem Abschluss ist er als Consultant in Data Warehouse und Business Intelligence Projekten tätig.