Datenbank Spektrum (2013) 13:231–236
DOI 10.1007/s13222-013-0136-8

DATENBANKGRUPPEN VORGESTELLT

Vorstellung des Lehrstuhls für Datenbanksysteme der TUM

Alfons Kemper · Thomas Neumann

Online publiziert: 18. September 2013
© Springer-Verlag Berlin Heidelberg 2013

Zusammenfassung Der Lehrstuhl für Datenbanksysteme der Technischen Universität München beschäftigt sich in der Forschung mit der Entwicklung, Optimierung und Anwendung „moderner" Datenbanktechnologie. Die Schwerpunkte der letzten Jahre lagen in der Entwicklung von Multi-Tenancy-fähigen Datenbanken, eScience-Datenbankanwendungen, Workload-Management für heterogene Anwendungen, RDF-Datenbanken und, insbesondere, Hauptspeicher-Datenbanken für hybride OLTP&OLAP-Anwendungen.

Keywords Lehrstuhl Datenbanksysteme TUM

1 Die Forschungsplattform: HyPer

Fast alle neueren Forschungsarbeiten wurden in das am Lehrstuhl entwickelte Datenbanksystem HyPer (High Performance Hybrid OLTP&OLAP Main-Memory Database System) integriert, das somit unseren Doktoranden als gemeinsame Forschungsplattform dient. Die Konzentration auf dieses System gibt den Doktoranden die Möglichkeit, ihre Ideen prototypisch in einem realen System zu erproben und hat zum anderen den Vorteil, das HyPer-System durch die vielfältigen Forschungsarbeiten für ein breiteres Anwedugsszenario weiterzuentwickeln.

Die Systemarchitektur wird auf der neuen Webseite www.hyper-db.de dargestellt, deren Einstiegsseite in Abb. 1 als „screen shot" gezeigt ist. Um die Leistungsfähigkeit sowie den Funktionsumfang des HyPer-Systems zu demonstrieren, stellen wir online eine TPC-H-Datenbank zur Verfügung. Die 22 TPC-H-Anfragen können dort (als vordefinierte Bausteine) ausgeführt werden. Weiterhin kann man sich die optimierten Anfragepläne als Baumstruktur inklusive annotierter Metainformation, wie z.B. Kardinalitätenabschätzung, anzeigen lassen. Natürlich kann man auch selbst formulierte Anfragen optimieren und ausführen lassen, da HyPer den vollständigen SQL92-Standard abdeckt. Für Lehrzwecke wurde neben der TPC-H-Datenbank auch die aus dem Lehrbuch [13] bekannte Universitäts-Datenbank installiert, so dass die Schnittstelle auch für SQL-Übungen für unsere (und andernorts ansässige) Studierende geeignet ist. Abbildung 2 zeigt einen „screen shot" dieser Web-Schnittstelle, bei dem Query 5 (also eine der 22 vorgegebenen TPC-H-Anfragen) mitsamt des optimierten Anfrageplans gezeigt ist.

Es gibt schon seit langer Zeit Hauptspeicher-Datenbanksysteme (in-memory DBMS). Darunter versteht man Datenbanksysteme, die den gesamten Datenbestand im Hauptspeicher halten, also keine Seiten zwischen Hauptspeicher-Puffer und Plattenspeicher hin und her tauschen (swappen) – wie dies bei herkömmlichen relationalen Datenbanksystemen der Fall ist. Die Hauptspeicher-Datenbanksysteme aus den Achziger Jahren waren allerdings eher Nischenprodukte, die für spezielle Anwendungen verwendet wurden. Dies hat sich durch den rasanten Fortschritt im Hardwarebereich geändert: Man hat heute Datenbank-Server mit Hauptspeicherkapazitäten von mehreren Terabytes, die Multi-Core-Server haben viele (bald hundert/e) Rechenkerne und es wurden neue Algorithmen und Datenstrukturen für Hauptspeicher-effiziente Datenverarbeitung entwickelt (Cache-effiziente Datensatzstrukturen wie PAX

A. Kemper (✉) · T. Neumann
Fakultät für Informatik, TU München, München, Deutschland
e-mail: alfons.kemper@in.tum.de

T. Neumann
e-mail: thomas.neumann@in.tum.de

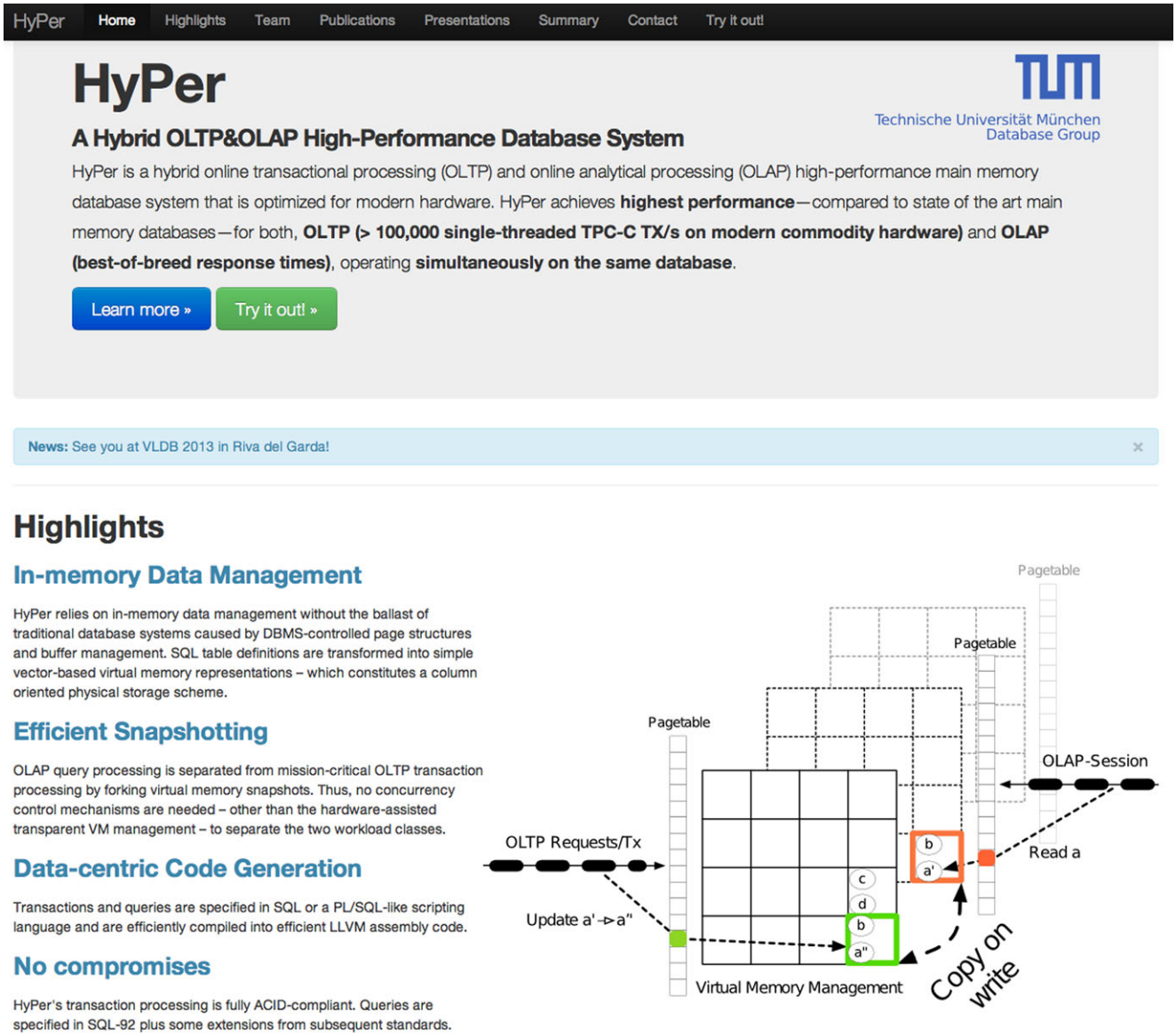

Abb. 1 Die HyPer-Website: www.hyper-db.de

oder Column-Stores, Kompression, Cache-effiziente Indexstrukturen, etc.).

HyPer ist ein *modernes* „ballastfreies“ Hauptspeicher-Datenbanksystem, das die Hardware-unterstützte virtuelle Speicherverwaltung des Betriebssystems für die Datenverwaltung und die Synchronisation zwischen OLTP-Transaktionen und OLAP-Anfragen effektiv ausnutzt. In Bezug auf die „in-core“-Datenverwaltung werden die relationalen Daten direkt, also ohne zusätzliche Indirektion durch eine DBMS-kontrollierte Puffer- und Seitenverwaltung auf den virtuellen Adressraum des OLTP-Prozesses abgebildet. Dieser Prozess kann transaktionskonsistente Snapshots der Datenbank anlegen, indem ein neuer OLAP-Prozess abgespaltet wird (in Linux mit dem Systembefehl **fork**). Der *copy on write*-Mechanismus des Betriebssystems/Prozessors sorgt für die Konsistenzerhaltung dieses Snapshots, indem Seiten mit sich ändernden Datenobjekten repliziert werden. Dieser Snapshot-Mechanismus entspricht dem alt-bekannten Schattenspeicherkonzept, das Lorie 1977 [21] erfunden hat – mit dem Unterschied, dass die virtuellen Speicher-Snapshots keine der damaligen Nachteile haben. Die Speicherfragmentierung ist im Hauptspeicher kein Problem und die ursprünglich aufwendige Verwaltung der Schattenkopien ist im HyPer-Ansatz durch die eingebaute Prozessorunterstützung hoch-effizient. Weiterhin erlaubt die virtuelle Speicherverwaltung den Unterhalt beliebig vieler (zeitlich versetzter) Schattenkopien. Dadurch war es möglich, mit HyPer ein Datenbanksystem zu realisieren, das die Vorteile beider „Welten“, der OLTP- und der OLAP-Datenbanken, vereint: Im Transaktionsdurchsatz

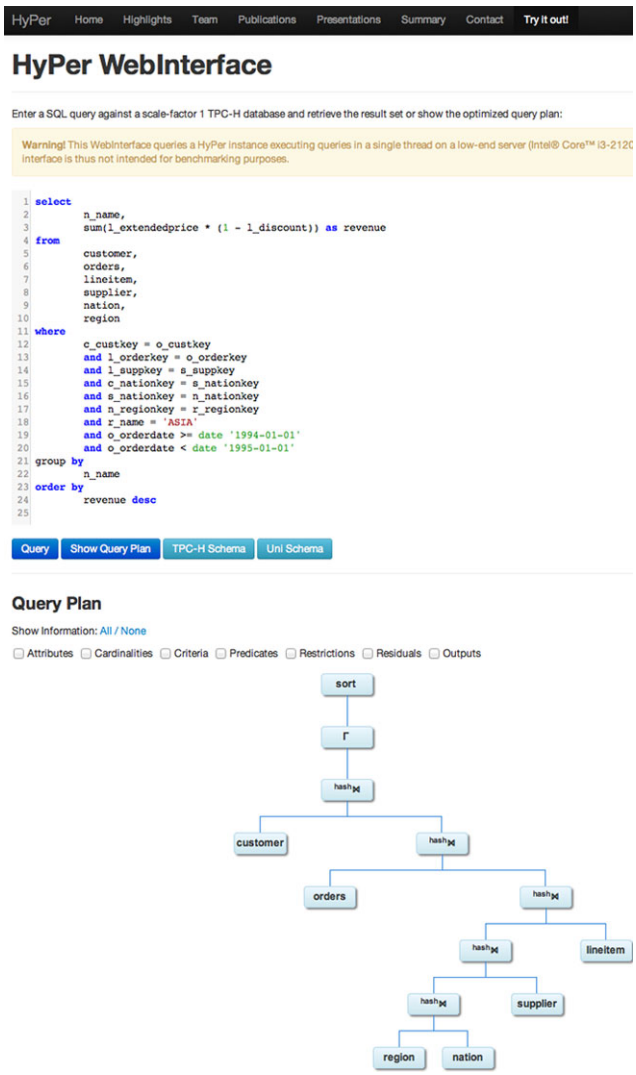

Abb. 2 Die vorinstallierten und via Internet flexibel nutzbaren Datenbanken: TPC-H und Uni

erzielt HyPer Werte vergleichbar oder besser als die dedizierten OLTP-Systeme (z.B. VoltDB) und in der OLAP-Anfragebearbeitung ist HyPer vergleichbar mit dedizierten Column-Stores (z.B. MonetDB, VectorWise, oder ähnliche). Diese Leistungsfähigkeit wurde anhand des neuen „Mixed Workload CH-BenCHmark“ [8] nachgewiesen, der die Transaktionsverarbeitung des TPC-C-Benchmarks und die Anfragen des TPC-H-Benchmarks in einem gemischten, parallel auf demselben Datenbestand auszuführenden Workload vereint. Die Leistungsfähigkeit eines derartigen hybriden Datenbanksystems kann für die effektive Unterstützung sogenannter „real-time business intelligence“ ausgenutzt werden, wie dies von namhaften Industrievertretern wie Hasso Plattner von SAP [32] gefordert wurde.

Die Leistungsfähigkeit von HyPer resultiert, neben gutem Software Engineering, aus folgenden Architekturentscheidungen:

1. Direkte Datenverwaltung im virtuellen Adressraum ohne zusätzliche Indirektion in der Form eines Datenbankpuffers oder einer Datenbank-kontrollierten Seitenverwaltung. Im Wesentlichen werden die relationalen Daten auf Vektoren abgebildet.
2. Minimal-invasive Synchronisation, indem Transaktionen Zulassungssperren für eine ganze Partition oder die gesamte Datenbank anfordern. Während der Verarbeitung werden keine Sperren mehr angefordert oder kontrolliert.
3. Die virtuellen Speicher-Snapshots erlauben die parallele Bearbeitung von Transaktionen und OLAP-Anfragen ohne jegliche Synchronisation seitens des DBMS wie in der ersten HyPer-Architekturbeschreibung [14] dargestellt.
4. Sowohl die Transaktionen als auch die Anfragen werden in nativen Maschinencode kompiliert, um den Interpretations-Overhead klassischer Datenbank-Systeme auszuschließen. Hierzu werden moderne Compilertechniken ausgenutzt, um hochoptimierenden Code auch noch sehr schnell kompilieren zu können wie in [28] beschrieben.

Diese „ballastfreie“ Datenbankarchitektur führt dazu, dass die Auswertung deklarativer SQL-Anfragen oder Transaktionen (stored procedurs) in HyPer dieselbe Performanz erzielt wie ein manuell geschriebenes Programm in einer maschinennahen Sprache wie C oder C++.

2 Forschungsarbeiten

Fast alle jüngeren Forschungsarbeiten wurden im Kontext des am Lehrstuhl entwickelten Hauptspeicher-Datenbanksystem HyPer [14] konzipiert und auch experimentell validiert.

In neueren Ausgaben des Data Engineering Bulletin finden sich Überblicksartikel über unsere Forschungsarbeiten. Das von P. Boncz [5] herausgegebene Sonderheft des Data Engineering Bulletin behandelt Column-Stores, u.a. HYRISE, HANA, HyPer, VectorWise, MonetDB und die Column-Store-Erweiterung des MS SQL Server. Ein weiteres Sonderheft über Hauptspeicher-Datenbanken wurde von P. Larson [18] editiert und enthält Beschreibungen über TimesTen, SolidDB, Hekaton, VoltDB, Calvin, und auch wieder HANA und HyPer.

Nachfolgend werden einige der Forschungsschwerpunkte unserer Gruppe aufgeführt.

Snapshotting für die Isolation von OLAP-Anfragen auf der transaktionalen Datenbank: In [23] haben wir verschiedene Snapshotting-Verfahren untersucht. Die Leistungsfähigkeit der virtuellen Speicher-Snapshots wurde anhand unseres Hauptspeicher-DBMS *HyPer* [14] experimentell gezeigt.

Langlaufende Transaktionen: Langlaufende Transaktionen verursachen nicht nur, aber besonders, in Hauptspeicher-Datenbanken erhebliche Probleme, da sie Ressourcen über lange Zeit blockieren. In [25] haben wir deshalb das Konzept der tentativen Ausführung langer Transaktionen auf einem Snapshot der operationalen Datenbasis entwickelt.

Kompilieren von Anfrageplänen: In [28] wurden effiziente Kompilierungstechniken für moderne Hardware entwickelt. In HyPer werden nicht nur die SQL-Anfragen, sondern auch die benutzerdefinierten Transaktionen (stored procedures) in LLVM-Code kompiliert.

Hauptspeicher-optimierte Indexstrukturen: Die Indexstruktur ART (Adaptive Radix Tree) wurde in [19] entwickelt. Experimentell konnte gezeigt werden, dass diese voll-dynamische Indexstruktur ART sogar effizienter ist als die in [16] beschriebene Cache-optimierte Suchbaumstruktur namens *FAST*, die allerdings nicht änderbar ist; sich also nur für eher statische Datenbestände eignet. Bezogen auf den TPC-C-Benchmark kann man mit ART eine Verdoppelung des Durchsatzes erzielen – im Vergleich zur Nutzung von Suchbäumen und Hashtabellen für die Indexierung.

Kompaktifizierung der Datenbank: Die Kompaktifizierung von Hauptspeicher-Datenbanken durch die Reorganisation in heiße und kalte Bereiche wurde in [11] entwickelt. Dadurch ist es möglich, den heißen Teil der Datenbank unkomprimiert im schnellstmöglichen Zugriff zu halten, wohingegen der kalte Teil komprimiert werden kann. Die Ideen wurden im Microsoft Hekaton-Projekt von [20] sowie von [34] für VoltDB aufgegriffen, um den gefrorenen Teil der Datenbank sogar aus dem Haupspeicher auszulagern. Dies würde aber nur bei OLTP-Anwendungen funktionieren; bei den Scan-basierten OLAP-Anwendungen müsste man immer wieder die ausgelagerten Teile der Datenbank in den Hauptspeicher transferieren.

Hybride Speichermodelle – adaptiv zwischen Column- und Row-Store: Die HPI/MIT-Autoren von [12] beschreiben HYRISE, ein Hauptspeicher-Datenbanksystem mit hybriden Speicherstrukturen. Die Relationen werden dabei vertikal fragmentiert, so dass man einen Trade-off zwischen Cache-Lokalität und Clustering der für eine Anwendung relevanten Attributwerte erzielen kann. Die dadurch erzielbare Leistungssteigerung wurde auch für das HyPer-Datenbanksystem in [31] experimentell untersucht.

Skalierbarkeit von Hauptspeicher-Datenbanken: Der Einsatz von Hauptspeicher-Datenbanksystemen für das Multi-Tenancy wurde in [24] demonstriert. Dabei wird der äußerst kleine „foot print“ des Hauptspeichersystems HyPer ausgenutzt, um sehr viele Instanzen parallel auf einem Multi-Core-Server laufen zu lassen.

In [26] haben wir das Scale-out eines Hauptspeicher-Datenbanksystems für skalierbare OLAP-Anwendungen auf den transaktionalen Daten demonstriert. Hierbei kommen mehrere Sekundär-Server zum Einsatz, die durch das Einbringen des Logs des Primär-Servers aktuell gehalten werden. Diese Methode der Aktualisierung über das Log des Primär-Servers bezeichnet man auch als „Log Sniffing“. Durch das Snapshotting kann man auf den Sekundärservern OLAP-Anfragen auswerten, währenddessen parallel auch die Log-Einträge in die Datenbank eingebracht werden.

Multi-Core- und NUMA-optimierte Anfrageauswertungstechniken: In [1] haben wir den für Multi-Core NUMA-Architekturen optimierten massiv-parallelen Sort/Merge-Join MPSM konzipiert. In einer weitergehenden Arbeit haben wir den Hash-Join auf NUMA-Architekturen optimiert und in [17] gezeigt, dass man damit auf einer 32-Core Maschine bis zu 3/4 Mrd Datensätze pro Sekunde joinen kann.

Hierarchische Datenstrukturen in relationalen Datenbanken: In [9, 10] haben wir in Kooperation mit der HANA-Gruppe von SAP Unterstützung für versionierte hierarchische Datenstrukturen im relationalen Modell konzipiert und in die Hauptspeicher-Datenbanksysteme HyPer und HANA integriert.

Bulk Loading für Big-Data-Analysen: Die Effizienz der Hauptspeicher-Datenbanksysteme in Bezug auf die Anfragebearbeitung wird in Zukunft sicherlich eine große Rolle im Bereich der Analyse von „Big Data“ spielen. Dazu ist es essentiell, dass man in textueller Form vorliegende Daten sehr schnell in die Hauptspeicher-Datenbank laden kann, um dort das interessierende „Datenfenster“ detailliert analysieren zu können. Hierzu wurde in [27] für HyPer ein hochparalleles Ladeverfahren entwickelt, das die Daten genauso schnell in das Datenbanksystem laden kann, wie sie über ein breitbandiges LAN-Netzwerk empfangen werden können.

Multi-Tenancy: Verschiedene Ansätze für Multi-Tenancy-Architekturen wurden in [2] und [3] entwickelt. In der neueren Arbeit [4] haben wir die Schema- und Datenevolution für Multi-Tenancy-DBMS konzipiert. In [33] wurde ein Verfahren für die Einhaltung von Servicelevel-Garantien für Multi-Tenancy-Datenbanken vorgestellt.

Anfrageoptimierung: Eines der Schwerpunktthemen, das in großer Breite behandelt wird, stellt die Anfrageoptimierung dar. Diese Ergebnisse sind zum großen Teil in das HyPer-Projekt eingeflossen, um hoch-effiziente Anfragepläne generieren zu können. Von den vielen Arbeiten in diesem Bereich seien nur die Arbeit über den sogenannten *GroupJoin* [22] sowie die jüngst mit dem „Best Paper Award“ der BTW-Tagung ausgezeichnete Arbeit zur

Abb. 3 Die Inflation des Lehrumfangs der Datenbank-Ausbildung

Selektivitätsabschätzungsfehler-toleranten Anfrageauswertung [29] genannt, die in Kooperation mit der SQL-Server-Gruppe von Microsoft entstand.

3 Lehraufgaben des Lehrstuhls

Der Lehrstuhl übernimmt die Datenbank-Grundlagenausbildung in den Bachelor-Studiengängen Informatik, Wirtschaftsinformatik, Bioinformatik sowie Informatik: Games Engineering. Die Vorlesung „Grundlagen Datenbanken“ wird dementsprechend von über 600 Studierenden belegt.

Darauf aufbauend wird jedes Jahr die Vorlesung „Einsatz und Realisierung von Datenbanken“ gehalten, die als Wahlpflichtveranstaltung des Bachelor/Master-Studiengangs Informatik sowie verpflichtend im Master-Studiengang Wirtschaftsinformatik gehört wird.

Als Grundlage für diese beiden (zahlenmäßig) großen Vorlesungen dient das mittlerweile in neunter Auflage erschienene Lehrbuch [13] sowie das Übungsbuch-Kompendium [15]. Abbildung 3 zeigt eine Ausstellung am Lehrstuhl, die Studierende zum zügigen Studium animieren soll - um dem inflationär wachsenden Lehrumfang zu entkommen.

Als Spezialvorlesungen werden regelmäßig gehalten: Implementierung von Datenbanken, Anfrageoptimierung, Transaktionssysteme sowie Verteilte Informationssysteme. Darüber hinaus können Studierende ein eher anwendungs- oder ein implementierungs-orientiertes Datenbankpraktikum belegen.

Im Elite-Masterstudiengang Softwaretechnik, der gemeinsam mit der Universität Augsburg sowie der LMU durchgeführt wird, übernimmt der Lehrstuhl den Schwerpunkt Datenbanken mit insgesamt drei Vorlesungen, davon eine Pflichtvorlesung.

Weiterhin übernimmt der Lehrstuhl die Datenbankvorlesungen für andere Fakultäten, wie z.B., der Geodäsie sowie der Munich School of Engineering im Bachelorstudiengang „Allgemeine Ingenieurwissenschaften“.

4 Kooperationen

Dem Lehrstuhl gehören derzeit auch zwei kooperierende Forscher an:

- Dr. Claudia Plant, die eine Nachwuchsgruppe des Helmholtz-Zentrums leitet und als Nachwuchsgruppenleiterin in die TUM eingebunden ist.
- Dr. Peter Boncz von der Universität Amsterdam bzw. dem CWI hat den renommierten Humboldt-Forschungspreis erhalten, um als Gastwissenschaftler am Lehrstuhl zu arbeiten.

Der Lehrstuhl für Datenbanksysteme organisierte das Herbsttreffen 2012 der GI-Fachgruppe Datenbanksysteme (die derzeit von A. Kemper geleitet wird) mit dem Thema „Scalable Analytics“, das mit ca. 90 Teilnehmern sehr gut besucht war. Das attraktive Programm mit vielen „hochkarätigen“ Vortragenden, insbesondere auch aus der Industrie, findet sich unter www-db.in.tum.de/scalableanalytics.

Der Lehrstuhl ist an einer europäischen Initiative zur Erstellung eines Linked Data Benchmarks www.ldbc.eu beteiligt und bringt dort die Expertise aus dem RDF-3X-Projekt [30], dem effizientesten RDF-Datenbanksystem, ein. Dieses Konsortium wurde auf Initiative „unseres“ Humboldt Forschungspreisträgers Peter Boncz [6] etabliert.

Neben den akademischen Partnern kooperiert der Lehrstuhl (in unterschiedlichsten Formen) mit industriellen Forschungslaboren, um den Technologietransfer zu ermöglichen. Hierzu zählen das HANA-Team der SAP, Siemens Corporate Technology, Oracle Labs, Google, HP Labs, Software AG, IBM und Microsoft.

Literatur

1. Albutiu M-C, Kemper A, Neumann T (2012) Massively parallel sort-merge joins in main memory multi-core database systems. PVLDB 5(10):1064–1075
2. Aulbach S, Grust T, Jacobs D, Kemper A, Rittinger J (2008) Multi-tenant databases for software as a service: schema-mapping techniques. In: Proc of the ACM SIGMOD conf on management of data
3. Aulbach S, Jacobs D, Kemper A, Seibold M (2009) A comparison of flexible schemas for software as a service. In: Çetintemel et al. [7], S 881–888
4. Aulbach S, Seibold M, Jacobs D, Kemper A (2011) Extesibility and data sharing in evolving multi-tenancy databases. In: Proceedings of the international conference on data engineering (ICDE)
5. Boncz PA (2012) Letter from the special issue editor on column stores. IEEE Data Eng Bull 35(1):2
6. Boncz PA, Fundulaki I, Gubichev A, Larriba-Pey J-L, Neumann T (2013) The linked data benchmark council project. Datenbank Spektrum 13(2):121–129

7. Çetintemel U, Zdonik SB, Kossmann D, Tatbul N (2009) In: Proceedings of the ACM SIGMOD international conference on management of data, SIGMOD 2009. June 29–July 2, 2009, Providence, Rhode Island, USA, ACM, New York
8. Cole R, Funke F, Giakoumakis L, Guy W, Kemper A, Krompass S, Kuno HA, Othayoth Nambiar R, Neumann T, Poess M, Sattler K-U, Seibold M, Simon E, Waas F (2011) The mixed workload CH-benCHmark. In: Graefe G, Salem K (Hrsg) DBTest. ACM, New York, S 8
9. Finis J, Brunel R, Kemper A, Neumann T, Färber F, May N (2013) DeltaNI: an efficient labeling scheme for versioned hierarchical data. In: Proc ACM SIGMOD conference on management of data, S 905–916
10. Finis J, Raiber M, Augsten N, Brunel R, Kemper A, Färber F (2013) Rws-diff: flexible and efficient change detection in hierarchical data. In: Proc of the ACM international conference on information and knowledge management (CIKM 2013), San Francisco, October 2013
11. Funke F, Kemper A, Neumann T (2012) Compacting transactional data in hybrid OLTP & OLAP databases. PVLDB 5(11):1424–1435
12. Grund M, Cudré-Mauroux P, Krüger J, Madden S, Plattner H (2012) An overview of HYRISE – a main memory hybrid storage engine. IEEE Data Eng Bull 35(1):52–57
13. Kemper A, Eickler A (2013) Datenbanksysteme – eine Einführung, 9. Aufl. Oldenbourg, München
14. Kemper A, Neumann T (2011) HyPer: a hybrid OLTP&OLAP main memory database system based on virtual memory snapshots. In: Abiteboul S, Böhm K, Koch C, Tan K-L (Hrsg) ICDE. IEEE Comput Soc, Los Alamitos, S 195–206
15. Kemper A, Wimmer M (2012) Übungsbuch Datenbanksysteme, 3. Aufl. Oldenbourg, München
16. Kim C, Chhugani J, Satish N, Sedlar E, Nguyen AD, Kaldewey T, Lee VW, Brandt SA, Dubey P (2010) FAST: fast architecture sensitive tree search on modern CPUs and GPUs. In: Proc of the ACM SIGMOD conf on management of data, S 339–350
17. Lang H, Leis V, Albutiu M-C, Neumann T, Kemper A (2013) Massively parallel NUMA-aware hash joins. In: 1st international workshop on in-memory data management and analytics (a VLDB 2013 workshop)
18. Larson PA (2013) Letter from the special issue editor on main-memory databases. IEEE Data Eng Bull 36(2):5
19. Leis V, Kemper A, Neumann T (2013) The adaptive radix tree: ARTful indexing for main-memory databases. In: Proceedings of the international conference on data engineering (ICDE)
20. Levandoski J, Larson P-A, Stoica R (2013) Identifying hot and cold data in main-memory databases. In: Proceedings of the international conference on data engineering (ICDE)
21. Lorie RA (1977) Physical integrity in a large segmented database. ACM Trans Database Syst 2(1):91–104
22. Moerkotte G, Neumann T (2011) Accelerating queries with group-by and join by groupjoin. PVLDB 4(11):843–851
23. Mühe H, Kemper A, Neumann T (2011) How to efficiently snapshot transactional data: hardware or software controlled? In: Proceedings of the international workshop on data management on new hardware, DaMoN, S 17–26
24. Mühe H, Kemper A, Neumann T (2012) The mainframe strikes back: elastic multi-tenancy using main memory database systems on a many-core server. In: Proc international conference on extending database technolog (EDBT), S 578–581
25. Mühe H, Kemper A, Neumann T (2013) Executing long-running transactions in synchronization-free main memory database systems. In: Proc of the conference on innovative data systems research (CIDR)
26. Mühlbauer T, Rödiger W, Reiser A, Kemper A, Neumann T (2013) ScyPer: elastic OLAP throughput on transactional data. In: Proc of the ACM SIGMOD workshop on data analytics in the cloud 2013 (DanaC 2013).
27. Mühlbauer T, Rödiger W, Seilbeck R, Reiser A, Kemper A, Neumann T (2013) Instant loading for main memory databases. In: Proc of the conf on very large data bases (VLDB), China
28. Neumann T (2011) Efficiently compiling efficient query plans for modern hardware. PVLDB 4(9):539–550
29. Neumann T, Galindo-Legaria CA (2013) Taking the edge off cardinality estimation errors using incremental execution. In: Markl V, Saake G, Sattler K-U, Hackenbroich G, Mitschang B, Härder T, Köppen V (Hrsg) BTW. LNI, Bd 214. S 73–92, GI
30. Neumann T, Weikum G (2008) RDF-3X: a RISC-style engine for RDF. PVLDB 1(1):647–659
31. Pirk H, Funke F, Grund M, Neumann T, Leser U, Manegold S, Kemper A, Kersten M (2013) CPU and cache efficient management of memory-resident databases. In: Proceedings of the international conference on data engineering (ICDE)
32. Plattner H (2009) A common database approach for OLTP and OLAP using an in-memory column database. In: Çetintemel et al. [7] S 1-2
33. Seibold M, Kemper A, Jacobs D (2011) Strict SLAs for operational business intelligence. In: IEEE international conference on cloud computing, IEEE CLOUD, S 25–32
34. Stoica R, Ailamaki A (2013) Enabling efficient OS paging for main-memory OLTP databases. In: Proceedings of the ninth international workshop on data management on new hardware, DaMoN