FernUniversität in Hagen

Fakultät für Mathematik und Informatik

Sommersemester 2013

Seminar
Big Data Management

Prof. Dr. Ralf Hartmut Güting

Fabio Valdés

Einführung

Angesichts der rasant anwachsenden weltweiten Informationsmenge, verursacht u.a. durch Standort- und Verbindungsdaten mobiler Geräte, Logdaten von IT-Systemen, Social-Media-Daten, Videoaufzeichnungen oder Finanztransaktionen – im Jahr 2012 wurden laut IBM täglich 2,5 Trillionen Bytes produziert; über Facebook werden monatlich 135 Mrd. Nachrichten geschrieben; Twitter speichert täglich 7 Terabyte Daten – müssen Datenbanksysteme große Datenmengen schnell für komplexe Anfragen von immer mehr Nutzern bereitstellen. SQL-Datenbanken sind dafür nur eingeschränkt geeignet.

Big Data Management ist der Oberbegriff für neue Methoden und Technologien zur Erfassung, Speicherung und Analyse von Daten verschiedener Strukturen in nichtrelationalen verteilten Datenbanken. Zum Einstieg in das Thema seien die Werke [LLC+11] und [Cat10] empfohlen.

Zugang zu Quellen

Die im Literaturverzeichnis erwähnten Paper finden Sie unter dem Link

`http://dna.fernuni-hagen.de/Lehre-offen/Seminare/1912-SS13/`.

Der Zugriff auf die Seite ist aus urheberrechtlichen Gründen auf Seminarteilnehmer beschränkt.

Themenauswahl

Bitte senden Sie bis zum 11.03.2013 eine Liste mit Ihren Prioritäten für die einzelnen Themen an **fabio.valdes@fernuni-hagen.de**. Die Liste soll 16 Einträge enthalten. Vergeben Sie Priorität 1 für das Thema, welches Sie am liebsten bearbeiten möchten. Markieren Sie das Thema, welches Sie auf keinen Fall bearbeiten möchten, mit Priorität 16. Dazwischenliegende Zahlen werden entsprechend gewertet. Alle Zahlen von 1 bis 16 sollen genau einmal vergeben werden.

Ich werde die Themen so verteilen, dass Ihre Prioritäten weitestgehend Berücksichtigung finden.

Themen

Die Seminarthemen gliedern sich in zwei Grundlagenvorträge, sechs Vorträge über MapReduce sowie acht Vorträge über Key-Value-Stores. Jeder Absatz beschreibt dabei genau ein Vortragsthema.

1 Grundlagen

1.1 Parallele Datenbanken

Der erste Vortrag bietet einen Überblick über grundlegende Ziele und Techniken der parallelen Datenverarbeitung [DG92, Kos00]. Im Vortrag sollen im Wesentlichen die Inhalte von [DG92] dargestellt werden.

1.2 Spaltenorientierte Datenbanken

Eine wichtige Grundlage für die nachfolgenden Vortragsthemen ist das Konzept der spaltenorientierten Datenbanken [SAB+05, AMF06]. Während beim traditionellen (zeilenorientierten) Ansatz alle Attribute eines Tupels nacheinander im Speicher abgelegt werden, kann bei einer spaltenorientierten Datenbank effizient auf einzelne Spalten einer Relation zugegriffen werden.

2 MapReduce – Skalierbare Datenauswertung

2.1 MapReduce

Das von Google entwickelte MapReduce Framework [DG04, DG08] dient dazu, die Verarbeitung sehr großer Datenmengen effizient auf zahlreiche Rechner zu verteilen. Dieser Vortrag beleuchtet die technischen Hintergründe des MapReduce-Verfahrens. MapReduce ist äußerst populär geworden, vor allem auch durch die frei verfügbare Implementierung Hadoop.

2.2 Google File System

Beim Google File System [GGL03] handelt es sich um ein verteiltes Dateisystem für datenintensive Anwendungen, welches die technische Grundlage für MapReduce liefert (analog zum Hadoop File System, welches grundlegend für Hadoop ist) und vor allem für Googles Websuche optimiert ist und verwendet wird. Die Aspekte Fehlertoleranz und Korrektheit sollten in diesem Vortrag eine wichtige Rolle spielen.

2.3 Kontroverse: Parallele Datenbanken vs. MapReduce

Der Einsatz von MapReduce im Datenbankumfeld ist nicht unumstritten. Einerseits werden die Vorteile der MapReduce-Techniken betont und der umfassende Einsatz von MapReduce gefordert, andererseits berufen sich die Verfechter herkömmlicher paralleler Datenbanksysteme auf den Einsatz von Indexen und andere Stärken, die von MapReduce nicht unterstützt werden. Diskussionen über Stärken und Schwächen sowie Experimente zu beiden Systeme bezüglich der Verarbeitung großer Datenmengen sind u.a. in [PPR+09, SAD+10, DG10] zu finden.

2.4 HadoopDB & SQL/MapReduce

HadoopDB [ABPA+09] ist ein hybrides System, welches die Vorteile traditioneller paralleler Datenbanksysteme mit denen des MapReduce-Konzepts vereinigt. Es bietet eine komplette Open-Source-Lösung für die verteilte Speicherung und parallele Auswertung großer Datenmengen durch MapReduce-Techniken in Rechnerclustern. In [FPC09] wird außerdem ein Ansatz für die Implementierung benutzerdefinierter Funktionen mit Hilfe von MapReduce vorgestellt.

2.5 Hive

Basierend auf Hadoop wurde von Facebook das System Hive [TSJ+09, TSJ+10] entwickelt. Hive bietet u.a. zusätzlich eine deklarative SQL-ähnliche Abfragesprache namens HiveQL, deren Befehle in MapReduce-Kommandos übersetzt und vom zugrundeliegenden Hadoop-System ausgeführt werden.

2.6 Pig

Das von Yahoo! entwickelte System Pig [ORS+08, GNC+09] basiert ebenfalls auf Hadoop und ist frei verfügbar. In der höheren Sprache Pig Latin lassen sich Programme entwickeln, die MapReduce-Befehle in Hadoop ausführen.

3 Key-Value-Stores und Verwandte – Skalierbare dynamische Datenbanken

3.1 Überblick: NoSQL-Datenbanken & Key-Value Stores

Dieser Vortrag führt in das Thema NoSQL-Datenbanken und Key-Value Stores ein. Einen ersten Überblick dazu bietet [BLS+11]. Auf das dort kurz erwähnte CAP-Theorem [Bre00, GL02] soll genau eingegangen werden.

3.2 Dynamo

Dynamo [DHJ+07] ist ein von Amazon eingeführtes Datenbanksystem auf Basis von Key-Value-Stores, das für die Zuverlässigkeit der Amazon-Internetplattform verantwortlich ist. Um höchstmögliche Verfügbarkeit zu gewährleisten, die für Amazon maßgeblich ist, verzichtet Dynamo in bestimmten Fehlerszenarien auf die Aufrechterhaltung der Datenkonsistenz.

3.3 Bigtable

Das verteilte Datenbanksystem Bigtable [CDG+08] wurde von Google entwickelt, um riesige Datenmengen auf tausende Standardrechner zu verteilen. Eine Bigtable ist eine verteilte multidimensionale Tabelle, in der jeder Wert durch einen Zeilenschlüssel, einen Spaltenschlüssel sowie einen Zeitstempel eindeutig identifiziert wird. Diese Technologie wird von Google selbst u.a. für die Suchmaschine, Google Earth und Google Finance verwendet.

3.4 CouchDB

Bei CouchDB (Cluster of unreliable commodity hardware Data Base) [Fou13, ALS10] handelt es sich um ein frei verfügbares dokumentenorientiertes Datenbanksystem von Apache. Es kombiniert ein einfaches Datenmodell mit dem MapReduce-Ansatz von Bigtable.

3.5 MongoDB

Das am meisten verwendete NoSQL-Datenbanksystem ist MongoDB [CD10]. MongoDB ist dokumentenorientiert und frei erhältlich, besitzt kein festes Schema und skaliert horizontal. Das System unterstützt die Indexierung beliebiger Attribute und verwendet das Format JSON.

3.6 Cassandra

Bei Cassandra [LM09, LM10] handelt es sich um ein verteiles NoSQL-Datenbanksystem, das von Facebook eingeführt wurde und vor allem hohe Skalierbarkeit und Ausfallsicherheit bietet. Heute wird Cassandra zwar nicht mehr von Facebook aber u.a. von Twitter verwendet.

3.7 H-Store & VoltDB

Um den rasanten technischen Fortschritt der letzten 30 Jahre bezüglich Prozessor-, Speicher- und Netzwerkkapazitäten für OLTP (Online Transaction Processing)-Systeme [HAMS08] auszunutzen, wurde das System H-Store [KKN+08] entwickelt. Es bietet typische Vorzüge eines NoSQL-Systems und erfüllt gleichzeitig das ACID-Prinzip. Ein Vergleich von H-Store mit einem relationalen Datenbankmanagementsystem ist in [SMA+07] zu finden. Das kommerzielle System VoltDB [Inc13] ist eine Implementierung von H-Store.

3.8 Benchmarks

Mit der Anzahl verschiedener Datenbanksysteme wächst auch der Bedarf, deren Leistungen zu vergleichen. Zu diesem Zweck wurde das erweiterbare Benchmarksystem Yahoo! Cloud Serving Benchmark [CST+10] eingeführt. Weiterhin analysieren die Autoren von [FTD+12] die Performanz traditioneller relationaler Datenbankmanagementsysteme im Vergleich zu NoSQL-Systemen.

Verwendete Technologien & theoretische Grundlagen

Der Vollständigkeit halber sei auf folgende Werke hingewiesen, die als Grundlage für einige der vorgestellten Themen dienen.

Vector Clocks

Das Konzept zeitlicher (Teil-)Ordnung von Ereignissen in verteilten Systemen wird in [Lam78] vorgestellt. Außerdem wird ein verteilter Algorithmus präsentiert, der ein System logischer Uhren synchronisiert.

Consistent Hashing

Mit Hilfe von konsistenten Hashfunktionen wird in [KLL+97] das Problem überlasteter Netzwerkknoten behandelt.

Multiversion Concurrency Control

Die Autoren von [BG83] entwickeln eine theoretische Grundlage, um die Korrektheit von Synchronisationsalgorithmen für konkurrierende Datenbankzugriffe zu analysieren.

Literatur

[ABPA+09] Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel J. Abadi, Alexander Rasin, and Avi Silberschatz. Hadoopdb: An architectural hybrid of mapreduce and dbms technologies for analytical workloads. *PVLDB*, 2(1):922–933, 2009.

[ALS10] J. Chris Anderson, Jan Lehnardt, and Noah Slater. *CouchDB - The Definitive Guide: Time to Relax*. O'Reilly, 2010.

[AMF06] Daniel J. Abadi, Samuel Madden, and Miguel Ferreira. Integrating compression and execution in column-oriented database systems. In *SIGMOD Conference*, pages 671–682, 2006.

[BG83] Philip A. Bernstein and Nathan Goodman. Multiversion concurrency control - theory and algorithms. *ACM Trans. Database Syst.*, 8(4):465–483, 1983.

[BLS+11] Laurent Bonnet, Anne Laurent, Michel Sala, Benedicte Laurent, and Nicolas Sicard. Reduce, you say: What nosql can do for data aggregation and bi in large repositories. *2012 23rd International Workshop on Database and Expert Systems Applications*, 0:483–488, 2011.

[Bre00] Eric A. Brewer. Towards robust distributed systems (abstract). In *PODC*, page 7, 2000.

[Cat10] Rick Cattell. Scalable sql and nosql data stores. *SIGMOD Record*, 39(4):12–27, 2010.

[CD10] Kristina Chodorow and Michael Dirolf. *MongoDB - The Definitive Guide: Powerful and Scalable Data Storage*. O'Reilly, 2010.

[CDG+08] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, June 2008.

[CST+10] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. Benchmarking cloud serving systems with ycsb. In *SoCC*, pages 143–154, 2010.

[DG92] David J. Dewitt and Jim Gray. Parallel database systems: the future of high performance database systems. *Communications of the ACM*, 35:85–98, 1992.

[DG04] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[DG10] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: a flexible data processing tool. *Commun. ACM*, 53(1):72–77, 2010.

[DHJ+07] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev.*, 41(6):205–220, October 2007.

[Fou13] Apache Software Foundation. Couchdb. Website, 2013. http://guide.apache.org/.

[FPC09] Eric Friedman, Peter M. Pawlowski, and John Cieslewicz. Sql/mapreduce: A practical approach to self-describing, polymorphic, and parallelizable user-defined functions. *PVLDB*, 2(2):1402–1413, 2009.

[FTD+12] Avrilia Floratou, Nikhil Teletia, David J. DeWitt, Jignesh M. Patel, and Donghui Zhang. Can the elephants handle the nosql onslaught? *PVLDB*, 5(12):1712–1723, 2012.

[GGL03] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, October 2003.

[GL02] Seth Gilbert and Nancy A. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, 2002.

[GNC+09] Alan Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan Narayanam, Christopher Olston, Benjamin Reed, Santhosh Srinivasan, and Utkarsh Srivastava. Building a highlevel dataflow system on top of mapreduce: The pig experience. *PVLDB*, 2(2):1414–1425, 2009.

[HAMS08] Stavros Harizopoulos, Daniel J. Abadi, Samuel Madden, and Michael Stonebraker. Oltp through the looking glass, and what we found there. In *SIGMOD Conference*, pages 981–992, 2008.

[Inc13] VoltDB Inc. Voltdb technical overview. Website, 2013. http://voltdb.com/downloads/datasheets_collateral/technical_overview.pdf.

[KKN+08] Robert Kallman, Hideaki Kimura, Jonathan Natkins, Andrew Pavlo, Alex Rasin, Stanley B. Zdonik, Evan P. C. Jones, Samuel Madden, Michael Stonebraker, Yang Zhang, John Hugg, and Daniel J. Abadi. H-store: a high-performance, distributed main memory transaction processing system. *PVLDB*, 1(2):1496–1499, 2008.

[KLL+97] David R. Karger, Eric Lehman, Frank Thomson Leighton, Rina Panigrahy, Matthew S. Levine, and Daniel Lewin. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *STOC*, pages 654–663, 1997.

[Kos00] Donald Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.

[Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.

[LLC+11] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. Parallel data processing with mapreduce: a survey. *SIGMOD Record*, 40(4):11–20, 2011.

[LM09] Avinash Lakshman and Prashant Malik. Cassandra: a structured storage system on a p2p network. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, SPAA '09, pages 47–47, New York, NY, USA, 2009. ACM.

[LM10] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system. *Operating Systems Review*, 44(2):35–40, 2010.

[ORS+08] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *SIGMOD Conference*, pages 1099–1110, 2008.

[PPR+09] Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, and Michael Stonebraker. A comparison of approaches to large-scale data analysis. In *SIGMOD '09: Proceedings of the 35th SIGMOD international conference on Management of data*, pages 165–178, New York, NY, USA, 2009. ACM.

[SAB+05] Michael Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Samuel Madden, Elizabeth J. O'Neil, Patrick E. O'Neil, Alex Rasin, Nga Tran, and Stanley B. Zdonik. C-store: A column-oriented dbms. In *VLDB*, pages 553–564, 2005.

[SAD+10] Michael Stonebraker, Daniel Abadi, David J. DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, and Alexander Rasin. Mapreduce and parallel dbmss: friends or foes? *Commun. ACM*, 53(1):64–71, January 2010.

[SMA+07] Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos, Nabil Hachem, and Pat Helland. The end of an architectural era (it's time for a complete rewrite). In *VLDB*, pages 1150–1160, 2007.

[TSJ+09] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive- a warehousing solution over a map-reduce framework. In *IN VLDB '09: PROCEEDINGS OF THE VLDB ENDOWMENT*, pages 1626–1629, 2009.

[TSJ+10] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Anthony, Hao Liu, and Raghotham Murthy. Hive - a petabyte scale data warehouse using hadoop. In *ICDE*, pages 996–1005, 2010.