Prof. Dr.-Ing. Thomas Kudraß
HTWK Leipzig, F IMN

Themen im OS "Datenbanksysteme - Aktuelle Trends" SS2015

Nachfolgend finden Sie einige Hinweise und Fragestellungen zu den ausgegebenen Themen. Die mit * gekennzeichneten Materialien sind leihweise bei Prof. Kudraß erhältlich.

1. Big Data

In dem Überblicksvortrag ist das aktuelle Schlagwort "Big Data" aus verschiedenen Perspektiven näher zu beleuchten, also die Verarbeitung großer Datenmengen. Was kennzeichnet diese gegenüber herkömmlichen Datenbanktechnologien? Was treibt die Entwicklung voran? Folgende Aspekte sollte der Vortrag behandeln:

- Die vier V's, die drei F's
- Datenquellen und Kategorien der Verarbeitung von Big Data
- Anwendungsbereiche (einer davon etwas mehr im Detail, z.B. Big Data in Natuwissenschaften)
- Überblick über Big-Data-Technologien
- Komponenten und Aufgaben einer Big-Data-Plattform
- Gesellschaftliche Aspekte von Big Data, z.B. Datenschutz

Quellen:

- Informatik-Spektrum, Bd. 37, Heft 2, April 2014, Themenheft "Big Data" .
- Pavlo Baron: Big Data für IT-Entscheider: Riesige Datenmengen und moderne Technologien gewinnbringend nutzen. Carl Hanser Verlag 2013.
- www.datanami.com

2. Big-Data-Technologien: Apache Hadoop

In diesem Vortrag soll als typisches Beispiel einer Big-Data-Technologie das frei verfügbare Framework Apache Hadoop mit seinen Komponenten vorgestellt werden. Folgende Schwerpunkte sollten im Vortrag enthalten sein:

- MapReduce-Ansatz und Anwendungshintergrund (Web 2.0) für NoSQL-Datenbanken
- Hadoop Distributed File System (HDFS)
- HBase-Datenbank
- Erweiterungen um Data-Warehouse-Funktionalität mittels Hive / HiveQL
- Entwicklung von MapReduce-Programmen mittels Pig / PigLatin
- Kommerzielle Systeme auf Basis von Apache Hadoop

Quellen:

- http://hadoop.apache.org

- Ramon Wartala: Hadoop: Zuverlässige, verteilte und skalierbare Big-Data-Anwendungen. Open Source Press, München 2012.
- Tom White: Hadoop: The Definitive Guide. O'Reilly & Associates, 3. Auflage, 2012.

3. Anfragesprachen für Big Data

Mit dem Aufkommen von NoSQL-Datenbanksystemen entstand eine Vielzahl von APIs und Sprachschnittstellen zum Zugriff auf die Daten. Derzeit laufen zahlreiche Bemühungen, um eine standardisierte und einheitliche Sprache zu entwickeln, so etwas wie das „SQL für NoSQL". Im Vortrag sollen die dafür vorhandenen Ansätze überblicksartig vorgestellt werden.

- UnQL – erste Ansätze für eine Unstructured Data Query Language
- JSONiq – die JSON Query Language für NoSQL-Datenbanken auf der Basis des JSON-Formats
- Oracle Big Data SQL – eine einheitliche SQL-Schnittstelle für Hadoop, No-SQL und herkömmliche relationale Datenbanken in Oracle

Quellen:

- Erik Meijer, Gavin Bierman: A Co-Relational Model of Data for Large Shared Data Banks. Communications of the ACM, Vol. 54 No. 4, Pages 49-58
- http://www.jsoniq.org
- http://www.oracle.com/us/products/database/big-data-sql/overview/index.html

4. Graphdatenbanken

Eine Graphdatenbank ist eine Datenbank, um mittels Graphen stark vernetzte Informationen darzustellen bzw. abzuspeichern. Die Graphen werden dabei durch Knoten und Kanten repräsentiert, denen Eigenschaften (Properties) zugeordnet werden können (Property Graph Model). Graphdatenbanken bieten eine Reihe spezialisierter Graph-Algorithmen, um z.B. Graphen zu traversieren, kürzeste Phade zu finden oder Hotspots in einem Netzwerk zu identifizieren. Sie eignen sich für Anwendungen in Sozialen Netzwerken oder für das Management großer Rechnernetze. Der Vortrag sollte folgende Punkte adressieren.

- Abgrenzung gegenüber relationalen Datenbanken und RDF Triple Stores auf der Basis von RDF
- Typische Graph-Algorithmen
- Systembeispiele: Neo4j, Oracle 12c Spatial and Graph/ NDM Features (Network Data Model)
- Cypher für Neo4j als Beispiel einer Graph-DB-Anfragesprache
- Interessante Anwendungen für Graphdatenbanken

Quellen:

- Ian Robinson, Jim Webber, Emil Eifrem: Graph Databases, O'Reilly, 2013 (auch als E-Book erhältlich)
- http://neo4j.com/
- http://www.oracle.com/technetwork/database/options/spatialandgraph/overview/index.html

5. Cassandra als Beispiel eines Wide Column Store

NoSQL (zumeist interpretiert als "not only SQL") beschreibt ein breites Spektrum von Datenbankmanagementsystemen, die dadurch charakterisiert sind, dass sie nicht dem weitverbreiteten relationalen Datenmodell folgen. NoSQL Datenbanken operieren daher nicht primär auf Tabellen und nutzen im Allgemeinen nicht SQL für den Datenzugriff. NoSQL-Datenbanksysteme sind oft optimiert für Anwendungen mit gleichzeitig hohen Datenanforderungen und häufigen Datenänderungen, wie sie im Web 2.0 auftreten. Sie zeichnen sich durch eine verbesserte (horizontale) Skalierbarkeit und Performance für bestimmte (nichtrelationale) Datenmodelle aus. Der Vortrag sollte auf folgende Aspekte eingehen:

- Motivation und Anwendungshintergrund (Facebook & Co.)
- Datenmodell von Wide Column Stores
- Anfrageschnittstellen, u.a. CQL
- Praktische Vorführung einer Beispiel-Datenbank

Quellen:

- S. Edlich, A. Friedland, J. Hampe, B. Brauer, M. Brückner: NoSQL : Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken. 2., aktualisierte und erweiterte Auflage. Hanser Verlag, München 2011.
- E. W. Redmond: Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. The Pragmatic Bookshelf, 2012.
- A. Lakshman, P. Malik: Cassandra - A Decentralized Structured Storage System

6. Spaltenorientierte Datenbanken (Column Stores)

Traditionell werden Datenbankanwendungen in einem Unternehmen in OLTP (Online Trasactional Processing) und OLAP (Online Analytical Processing) unterteilt. OLTP- und OLAP-Systeme wurden in der Vergangenheit bereits sehr stark optimiert, die Leistung in entsprechenden Benchmarks bewertet. Dabei haben sich sowohl Hardware als auch Datenbanken weiterentwickelt. Mittlerweile gibt es DBMS, die Daten spaltenorientiert organisieren (Column Stores) und dabei ideal das Anforderungsprofil analytischer Anfragen abdecken. Der Vortrag sollte auf folgende Aspekte eingehen:

- Prinzip der spaltenorientierten Speicherung (Row Store vs. Column Store)
- Decomposition Storage Model zur vertikalen Speicherug von Daten
- Kompressionstechniken
- System-Implementierungen
- Bewertung / Performance

Quellen:

- O. Herden: Spaltenorientierte Speicherung. in: T. Kudraß (Hrsg): Taschenbuch Datenbanken, Carl Hanser Verlag, 2. Auflage, 2015. *
- K.-U. Sattler: Column Stores, Datenbank-Spektrum 30/2009. *
- G. P. Copeland, S. N. Khoshafian: A decomposition storage model. SIGMOD ’85, 1985
- S. Harizopoulos, D. Abadi, P. Boncz: Column-Oriented Database Systems, Tutorial. VLDB 2009.

7. Hauptspeicherdatenbanken (In Memory Databases)

Heutzutage steht deutlich mehr Hauptspeicher zur Verfügung, der in Kombination mit der ebenfalls wesentlich gesteigerten Rechenleistung es erlaubt, komplette Datenbanken von Unternehmen komprimiert im Speicher vorzuhalten. Beide Entwicklungen ermöglichen die Bearbeitung komplexer analytischer Anfragen in Sekundenbruchteilen und ermöglichen so völlig neue Geschäftsanwendungen (z.B. im Bereich Decision Support). Der am Hasso-Plattner-Institut entwickelte Prototyp SanssouciDB vereinigt beide Konzepte und wurde bei SAP mittlerweile zur Produktreife unter dem Namen HANA geführt. Der Vortrag sollte auf folgende Aspekte eingehen:

- Echtzeit-Anforderungen von Geschäftsanwendungen (z.B. Mahnungen, Available-to-Promise): Workload, Charakteristika von OLTP- und OLAP-Anwendungen
- DBMS-Architektur am Beispiel von SanssouciDB
- DB-Basisoperationen in Hauptspeicher-DB: INSERT, UPDATE, DELETE, SELECT
- Anfrageverarbeitung (Aggregation, Joins)
- Insert-Only-Strategien
- Transaktionsmanagement, Logging, Recovery
- Partitionierung und Replikation

Quellen:

- J. Krueger, M. Grund, C. Tinnefeld, B. Eckart, A. Zeier, H. Plattner: Hauptspeicherdatenbanken für Unternehmensanwendungen - Datenmanagement für Unternehmensanwendungen im Kontext heutiger Anforderungen und Trends, in: Datenbank-Spektrum Bd. 10 Heft 3/Dez. 2010, Springer-Verlag. *
- H. Plattner: SanssouciDB: An In-Memory Database for Processing Enterprise Workloads, in: 14. Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW), Kaiserslautern, 2011. *
- H. Plattner: Lehrbuch In-Memory Data Management: Grundlagen der In-Memory-Technologie, Springer Gabler, 2013.

8. Cloud-Datenbanken

Cloud Computing besitzt ein großes Potential für Unternehmen zur Reduktion ihrer Kosten sowie einer Verkürzung der Entwicklungszeiten für marktreife Produkte (Time-to-Market) durch Verschlankung notwendiger Hardware-Infrastruktur. Besonders betrachtet werden sollen Speicher- und Datenbank-Service, die von einer Cloud zur Verfügung gestellt werden können. Der Vortrag sollte auf folgende Aspekte eingehen:

- Einführung in das Cloud Computing bzw. Cloud Data Management: Klassifikation, Prinzipien und Vorteile
- Allgemeine Probleme von Cloud Data Management: Partitionierung der Daten, Konsistenzkontrolle (CAP, PACELC), Skalierbarkeit, Performance, Migration
- Speicherkategorien in der Cloud: Blob, Table, Datenbank
- Database-as-a-Service

- Anbieter von Cloud-Datenbanken: z.B. Amazon, Google, Microsoft
- APIs, Datenmodelle und Speichermedien für Cloud-Datenbanken, Multi-Tenant
- Kriterien für Cloud-Datenbanken: Elastizität hinsichtlich Datenvolumen, Ausfallsicherheit/Hochverfügbarkeit, Kosteneinsparung durch Elastizität, Administrationsaufwand

Quellen:

- D. Kossmann, T. Kraska: Data Management in the Cloud: Promises, State-of-the-art, and Open Questions, in: Datenbank-Spektrum Bd. 10, Heft 3/Dezember 2010, Springer.*
- M.C. Jaeger, U. Hohenstein: Cloud Storage: Wieviel Cloud Computing steckt dahinter?, in: 14. Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW), Kaiserslautern, 2011. *
- N. Ritter: Cloud-Datenbanken, in: T. Kudraß (Hrsg): Taschenbuch Datenbanken, 2.Auflage, Carl Hanser, 2015. *

9. Data Streams & Complex Event Processing

Der technologische Fortschritt im Bereich der Mikroelektronik und Kommunikationstechnik führt zunehmend zu stark vernetzten, mit Sensoren ausgestatteten verteilten Informationssystemen (Internet of Things). Die damit einhergehende steigende Anzahl an Sensorinformationen, deren Daten in Form von Datenströmen bereitgestellt werden, ermöglichen neue Anwendungsszenarien und treiben neue Verarbeitungstechniken. Smart Data können zur Verbesserung von Steuerungs- und Entscheidungsprozessen eingesetzt werden. Wesentliche Anwendungen hierfür sind Smart Cities oder das Smart Home.
Schwerpunktmäßig soll der Vortrag die Verarbeitung von Datenströmen (data streams) betrachten, bei dem kontinuierlich Anfragen an einen Strom von eingehenden Daten gestellt werden. Hierfür existiert auch der Begriff Complex Event Processing (CEP). Dabei werden Ereignisse aus unterschiedlichen Quellen kombiniert, um daraus bestimmte Muster abzuleiten, die auf ein relevantes Ereignis hindeuten, z.B. eine Bedrohungssituation, auf das umgehend reagiert werden muss.

Der Vortrag sollte auf folgende Aspekte eingehen:

- Anwendungsszenarien für CEP / Data Streams insbesondere im Netzwerk- und Systemmanagement, Geschäftsprozessmanagement, Smart-City-Anwendungen und in der Finanzwirtschaft (Trading, Fraud Detection)
- Event Query Languages: Eventbegriff, Eventalgebra, Data Stream Query Language(CQL), Integration von Event Processing in SQL
- Beziehung von CEP zu Zeitreihen-Datenbanken
- CEP / Data Stream - Systeme, z.B. Apache Storm, Oracle Event Processing

Quellen:

- K.P. Eckert, R. Popescu-Zeletin: Smart Data als Motor für Smart Cities, in: Informatik-Spektrum, Bd. 37, Heft 2, April 2014
- D. Luckham: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Professional, 2002.
- M. Eckert, F. Bry: Complex Event Processing, in: Informatik-Spektrum: Bd. 32, Heft 2, 2009.
- http://www.complexevents.com/

10. Apache Spark – Ein Framework zur Echtezeitdatenanalyse

Apache Spark ist eine Open-Source-Engine zur schnellen Verarbeitung großer Datenmengen mit großer Geschwindigkeit, einfacher Nutzbarkeit und komplexen Analysen. Die Spark-Engine läuft in einer Vielzahl von Umgebungen, z.B. Cloud-Services oder Hadoop. Es kann genutzt werden, um ETL-Prozesse auszuführen, interaktive Anfragen (SQL), fortgeschrittene Analytik (z.B. Machine Learning) und Streaming über großen Datenmengen in einer Vielzahl von Datenquellen (z.B. HDFS, Cassandra, HBase, S3). Spark unterstützt verschiedene populäre Sprachen wie Java, Python und Scala.

- Überblick über das Sparc Ecosystem
- SparkSQL:
- MLIib – Machine Learning
- Spark Streaming:
- GraphX
- Ausblick: Apache Flink vs. Apache Spark

Quellen:

- https://spark.apache.org/
- https://flink.apache.org

11. Geodatenbanken

Geodatenbanken sind ein wesentlicher Bestand von Geoinformationssystemen (GIS) und anderen Anwendungen, die räumliche Daten (Geodaten) verarbeiten. Sie dienen der Modellierung, der Speicherung und der Anfrage von Geodaten.

In einem Überblicksvortrag sollten folgende Aspekte behandelt werden:

- Einordung und Abgrenzung: Geo-Informationssysteme (GIS)
- Geodaten: Eigenschaften, Metadaten
- Standardisierung von Geodatenmodellen: Datenschemata
- Funktionalität von Geodatenbanksystemen
- Räumliche Datenbankanfragen
- Räumliche Indexe
- Geocoding
- Produkte (Oracle Spatial oder PostGIS)

Quellen:

- T. Brinkhoff: Geodatenbanksysteme in Theorie und Praxis, Wichmann Verlag, 2005.
- T. Brinkhoff: Geodatenbanken, in: T. Kudraß (Hrsg.): Taschenbuch Datenbanken, Hanser-Verlag. 2. Auflage 2015.
- R. Kothuri, A. Godfrind, E. Beinat: Pro Oracle Spatial, Dokumentation Oracle Spatial Reference and User's Guide, Apress, 2004.*

12. Temporale Datenbanken

Temporale Datenbanksysteme unterstützen die Verarbeitung und Speicherung von zeitbezogenen (temporalen) Daten über die zeitbezogene Datentypen hinaus. Derzeit existiert kein kommerzielles DBMS, das die Anforderungen der temporalen Datenhaltung vollständig abbildet. Allerdings gibt es Komponenten für bestimmte Arten von temporalen Daten, z.B. Oracle Time Series für die Verarbeitung von Zeitreihen auf der Basis von spezifischen Time-Series-Datentypen.

Der Vortrag sollte auf folgende Aspekte eingehen:

- Basiskonzepte temporaler Datenbanken: Gültigkeitszeit/Aufzeichnungszeit, temporale Datentypen, Historisierung, Kalender, Zeitstempel
- Integrität in temporalen Datenbanken
- Abbildung auf herkömmliche relationale Datenbanken
- Aktueller Stand der Standardisierung in SQL:2011, SQL-Erweiterungen
- Unterstützung in DBMS-Produkten, insbesondere IBM DB2 V10

Quellen:

- D. Petkovic: Was lange währt, wird endlich gut: Temporale Daten in SQL-Standard, in: Datenbank-Spektrum, Bd. 13, Heft 2, 2013.
- R.T. Snodgrass: The TSQL2 temporal query language, Springer-Verlag, Berlin 1995.
- R.T. Snodgrass, Michael Böhlen, Christian S. Jensen, Andreas Steiner: Transitioning Temporal Support in TSQL2 to SQL3, 1997, TIMECENTER Technical Report TR-8
- K. Kulkarni, J. Michels: Temporal Features in SQL:2011, SIGMOD Record Vol. 41, No. 3, 2012, S. 34-43.

13. Objektdatenbanken am Beispiel db4o

Objektdatenbanken waren in den 1990-er Jahren ein großer Trend und beeinflussten die Weiterentwicklung relationaler Datenbanksysteme hin zu objektrelationalen Systemen. Heutzutage haben Objektdatenbanken als embedded Databases ein neues Anwendungsgebiet mit Wachstumspotential gefunden. Der Vortrag sollte auf folgende Aspekte eingehen:

- Basiskonzepte objektorientierter Datenbanken (auch in Abgrenzung zu objektrelationalen Datenbanken), insbesondere Persistenz
- Modellierung von Beziehungen in Objektdatenbanken
- API für einen Objektlebenszyklus (CRUD-Operationen) am Beispiel von db4o
- Anfrageschnittstellen: QBE (Query By Example), S.O.D.A. / Criteria Queries, Native Abfragen
- Transaktionen in db4o
- Client/Server-Modes in db4o
- weitere interessante Eigenschaften (Replikation, Callbacks, Ladeverhalten)
- Alternativen für Small Databases, z.B. Apache Derby

Quellen:

- I. Brenner: Datenbankentwicklung mit db4o - Einführung in eine objektorientierte Datenbank, online unter www.inabrenner.de
- http://odbms.org (Portal rund um das Thema Objektorientierte Datenbanken)

14. R & SQL

R ist eine Programmiersprache für statistisches Rechnen und statistische Graphiken für eine Vielzahl wissenschaftlicher und kommerzieller Anwendungen. Die Kombination mit Datenbanken erhöht deren Funktionsmächtigkeit. Dies wird durch weitere zusätzliche Pakete zur SQL- und Datenbank-Anbindung unterstützt. Der Vortrag sollte folgende Schwerpunkte setzen.

- Basiskonzepte und Schnittstellen der Sprache R
- SQLDF-Package zum Ausführung von SQL-Befehlen auf R Data Frames
- Weitere Pakete mit Anbindung von Datenbanken und XML
- Ansätze für Data Mining
- Vorführung eines kleinen Beispiels

Quellen:

- http://de.wikipedia.org/wiki/R_(Programmiersprache), mit vielen Hinweisen auf weiterführende Literatur
- G. Grothendieck sqldf: Perform SQL Selects on R Data Frames, http://cran.r-project.org/web/packages/sqldf/index.html

15. Information Extraction

Information Extraction (IE) bezeichnet den Ansatz, strukturiertes Wissen aus unstrukturierten oder bestenfalls semi-strukturierten Daten (z.B. HTML- oder XML-Dokumente) zu gewinnen. Intelligente Informationsextraktionstechniken sind dabei die wichtigsten Bestandteile bei der Generierung und Repräsentation von Wissen für eine Vielzahl von Anwendungen, insbesondere bei der Auswertung des World Wide Web als weltgrößtem Informationsbestand.

Der Vortrag sollte folgende Schwerpunkte umfassen:

- Einordnung und Abgrenzung von IE gegenüber anderen Teilgebieten der Informatik: Natural Language Processing (NLP), Machine Learning, Text Mining, Information Retrieval
- Historie: Message Understanding Conferences (MUC)
- Anwendungen
- Extraktion von (named) Entities und Beziehungen, Attribute und Klassen von Entities
- Extraktionstechniken: Klassifikatoren, Sequenz-Modelle (Hidden Markov Modelle)
- hybride Techniken unter Einbeziehung von menschlicher Interaktion
- semantische Aspekte der Informationsextraktion
- Bewertungskriterien bei der Informationsextraktion

- Werkzeuge zur Informationsextraktion (z.B. Open-Source-Tool GATE, System T von IBM)

Quellen:

- W.-T. Balke: Introduction to Information Extraction: Basic Notions and Current Tremds, in: Datenbank-Spektrum Bd. 12 Heft 2, 2012. *
- P. Klügl, M. Toepfer: Informationsextraktion, in Informatik-Spektrum Bd. 37 Heft 2, 2014
- J. Piskorski, R. Yangarber: Information Extraction: past, Present and Future, in: Poibeau et. el. (eds.): Multisource, Multilingual Information Extraction and Summarization, Springer-Verlag, 2013, S. 23-48.