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

Themen im OS "Datenbanksysteme - Aktuelle Trends" SS2018

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 Naturwissenschaften)
- Beziehung zum Internet of Things (IoT)
- 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".
- Min Chen, Shiwen Mao, Yunhao Liu: Big Data – A Survey. Mobile Netw Appl (2014) 19:171–209, Springer-Verlag.
- Pavlo Baron: Big Data für IT-Entscheider: Riesige Datenmengen und moderne Technologien gewinnbringend nutzen. Carl Hanser Verlag 2013.
- www.datanami.com

2. Data Lakes

Der Data Lake al sein Repository speichert strukturierte und unstrukturierte Rohdaten in der Form, in der sie von der Datenquelle bereitgestellt werden. Bei Data Lakes muss vorher nicht bekannt sein, welche Analysen voraussichtlich durchgeführt werden sollen. Der Data Lake kann dabei helfen, organisatorische Grenzen zu überwinden und die Systemkomplexität zu reduzieren. Hierfür werden jedoch Ansätze benötigt, die die Datenintegration und andere Anforderungen bewältigen. Folgende Aspekte sollte der Vortrag behandeln:

- Motivation und Idee von Data Lakes
- Themenlandkarte zu Data Lakes
- Data-Lake-Technologien: Datenspeicherung, Datenaufnahme (Ingestion), Data Profiling und Integration, Verarbeitung, Datenbank-Anbindung

- Beziehung zu Data Streams: Lambda- und Kappa-Architektur
- Data Governance
- Beispiele
- Kritikpunkte (Data Lakes = Fake News?)

Quellen:

- Christian Mathis: Data Lakes. In: Datenbank-Spektrum, Bd. 17, Heft 3, November 2017, S. 289-293 *
- James Dixon: Pentaho, Hadoop and data lakes. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/
- Data Lake. https://en.wikipedia.org/wiki/Data_lake

3. 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, 4. Auflage, 2015.

4. Apache Spark – Ein Framework zur Echtzeitdatenanalyse

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/
- Bill Chambers, Mathei Zaria: Spark: The Definitive Guide. Big data processing made simple. O'Reilly, 2018.
- Holden Karau, Andy Kowinski u.a.: Learning Spark: Ligthning-Fast Data Analysis. O'Reilly and Associates, 2015.

5. Apache Flink – Ein Framework zur Verarbeitung von Datenströmen

Apache Flink ist ein Framework zur Verarbeitung von Datenströmen (Streams), welches im Kern eine verteilte Datenfluss-Engine beinhaltet. Flink führt beliebige Datenfluss-Programme in einem parallelen oder einem Pipeline-Modus aus. Das Laufzeitsystem von Apache Flink erlaubt die Verarbeitung von Datenströmen und von Stapelprogrammen. Programme könen in Java, Scala, Python und SQL geschrieben werden und werden automatisch kompiliert und optimiert in Datenflussprogramme, die in einem Cluster oder einer Cloud-Umgebung ausgeführt werden.
Im Jahre 2010 wurde das Forschungsprojekt Stratosphere in Berlin gestartet, aus dessen Laufzeitumgebung Flink hervorgegangen war. Seit 2014 ist Flink ein Top Level-Projekt bei Apache.
Flink verfügt über keine eigene Datenhaltung, stellt aber Datenverbindungen zu anderen Systemen, wie z.B. HDFS, Apache Cassandra oder ElasticSearch, als Quelle oder Senke bereit.
Apache Flink beinhaltet zwei Core APIs: eine DataStream API für begrenzte und unbegrenzte Datenströme und eine DataSet API für begrenzte Datenmengen. Flink bietet auch eine Table API, eine SQL-artige Sprache für relationale Stream- oder Batch-Verarbeitung, die leicht in die APIs von Flink eingebettet werden kann. Der Vortrag sollte auf folgende Aspekte eingehen:

- Programmiermodell & verteilte Laufzeitumgebung
- Systemarchitektur
- Fehlertoleranz: Checkpoints und Savepoints – Machine Learning
- APIs: DataStream API, DataSet API. Table API & SQL
- Beispiel-Anwendungen

Quellen:

- https://flink.apache.org/
- Alexander Alexandrov u.a.: The Stratosphere platform for big data analytics. The VLDB Journal 23, 6 (Dezember 2014), 939-964

6. 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". Mittlerweile ist auch ein Standardisierungsvorschlag für JSON und SQL veröffent-

licht worden. Im Vortrag sollen die dafür vorhandenen Ansätze überblicksartig vorgestellt werden.

- JSONiq – die JSON Query Language für NoSQL-Datenbanken auf der Basis des JSON-Formats
- SQL/JSON – ANSI SQL/JSON Part 1 & 2, Proposal von 2014
- SQL++ - SQL-inspirierte Anfragefrage für das Asterix Datenmodell (angereichertes JSON)
- Oracle Big Data SQL – eine einheitliche SQL-Schnittstelle für Hadoop, No-SQL und herkömmliche relationale Datenbanken in Oracle

Quellen:

- Dusan Petkovic: SQL/JSON Standard: Properties and Deficiencies, in: Datenbank-Spektrum, Bd. 17 (2017), S. 277-287. *
- Asterix DB: The SQL++ Query Language, https://ci.apache.org/projects/asterixdb/sqlpp/manual.html
- http://www.jsoniq.org
- https://www.oracle.com/database/big-data-sql/index.html

7. Grundlagen der Graphdatenverarbeitung

Viele Big-Data-Anwendungen in Wirtschaft und Wissenschaft erfordern die Verwaltung und Analyse von großen Mengen von Graphdaten. Geeignete Systeme zur Verwaltung und Auswertung von Graphdaten müssen Anforderungen entsprechen wie die Unterstützung ausdrucksstarker Graphdatenmodelle mit heterogenen Knoten und Kanten, mächtige Anfrage- und Graph-Mining-Möglichkeiten, einfache Benutzbarkeit, gute Performance und Skalierbarkeit. Im Vortrag sollen die dafür vorhandenen Ansätze überblicksartig vorgestellt werden.

- Anforderungen an Graphdatenbanksysteme
- Graphdatenmodelle: Graphstrukturen (Gerichteter Graph u.a.), RDF Graph, (Extended) Property Graph
- Query Interfaces: vertex-centric, edge-centric, graph-centric
- Speicherung von Graphen: Datenstrukturen (z.B. Adjazenzmatrix, Adjazenzliste, CSR, Triple Table …)
- Indexierung von Graphen
- Engines zur Graphenverarbeitung (Graph Processing System), z.B. Apache Giraph, Pregel

Quellen:

- Martin Junghanns, Andre Petermann, Martin Neumann, Erhard Rahm: Management and Analysis of Big Graph Data: Current Systems and Open Challenges, Handbook of Big Data Technologies, 2017, S. 457-505. *
- Marcus Paradies, Hannes Voigt: Big Graph Data Analytics on Single Machines – An Overview, in: Datenbank-Spektrum Bd. 17, Heft 2, Juli 2017. *

8. Graphdatenbanken und Graph-Frameworks

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. 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. Verteilte Graph Dataflow Systeme bieten allgemeine Operatoren (z.B. Map, Reduce, Filter, Join) zum Laden und Transformieren unstrukturierter und strukturierter Daten sowie spezialisierte Operatoren und Bibliotheken für iterative Algorithmen (z.B. Machine Learning, Graphenanalyse). Mit solchen Systemen als Framework lässt sich die Gesamtkomplexität und die Performance für den Anwender verbessern. Der Vortrag sollte folgende Punkte adressieren.

- Abgrenzung gegenüber relationalen Datenbanken und RDF Triple Stores
- Typische Graph-Algorithmen
- Graph Dataflow Systems, zum Beispiel Gelly, Gradoop
- Graph-Datenbanksysteme, zum Beispiel Neo4j, Oracle 12c Spatial and Graph/NDM Features
- Graph-DB-Anfragesprachen, zum Beispiel Cypher für Neo4j
- Interessante Anwendungen für Graphdatenbanken

Quellen:

- Ian Robinson, Jim Webber, Emil Eifrem: Graph Databases, O'Reilly, 2013 (auch als E-Book erhältlich)
- Martin Junghanns, Andre Petermann, Martin Neumann, Erhard Rahm: Management and Analysis of Big Graph Data: Current Systems and Open Challenges, Handbook of Big Data Technologies, 2017, S. 457-505.
- Marcus Paradies, Hannes Voigt: Big Graph Data Analytics on Single Machines – An Overview, in: Datenbank-Spektrum Bd. 17, Heft 2, Juli 2017.
- http://neo4j.com/
- http://www.oracle.com/technetwork/database/options/spatialandgraph/overview/index.html

9. Blockchains

Blockchains, die „Magie hinter Bitcoin", werden in den nächsten Jahren vieles grundlegend verändern und erscheinen manchen als ein neues Betriebssystem für unsere Gesellschaft. Somit hat die Blockchain-Bewegung mehr Potential als eine einfache Netzwerk-Technologie. Dabei besteht der Paradigmenwechsel für die Gesellschaft darin, dass zur Abwicklung von Geschäften kein Vermittler mehr zwischen den beteiligten Menschen benötigt wird, z.B. eine Bank oder eine staatliche Institution. Diese Vermittler garantieren viele Voraussetzungen für Business-Transaktionen, z.B. bei Geldüberweisungen. Hierzu gehören die Authentisierung der Geschäftspartner, die Aufzeichnung von Transaktionen und die Validierung der eigentlichen Transaktion.
Die Blockchain ist eine verteilte Softwarearchitektur, die die Aufgaben eines Vermittlers übernimmt. Sie sichert das Vertrauen der Teilnehmer untereinander, ohne dass es eine zentrale Instanz oder einen besonderen Knoten dafür gibt.

Der Vortrag sollte auf folgende Aspekte eingehen:

- Verteilte Peer-to-Peer-Systeme (P2P) als Spezialfall dezentraler Systeme, Besonderheiten von Blockchains gegenüber herkömmlichen P2P-Systemen
- Bestandteile der Blockchain-Technologie: Accountverwaltung, Transaktionen und Ledger (Kontobuch), Hash-Puzzles als Proof-of-Work, Miner
- Integritätssicherung: Datenstrukturen von Transaktionen und Blockchains (Merkle Tree), Operationen (Hinzufügen von neuen Blöcken), Umgang mit Manipulationen
- Dezentrale Verwaltung der Blockchain: Bestimmung des gültigen Heads eines Ledgers / Validierung von Blöcken, "Mining" von neuen Blöcken, Kommunikation mit den Peers / Weiterleiten von neuen Blöcken
- Grenzen der Blockchain-Technologie
- Anwendungen: Bitcoin, Ethereum, Hyperleder Fabric, IOTA

Quellen:

- Daniel Drescher: Blockchain Basics, Apress, 1. Auflage, 2017. *

  Alternativ: Daniel Drescher: Blockchain Grundlagen. Eine Einführung in die elementaren Konzepte in 25 Schritten, mitp Business, 2017.
- Tim Menapace: Clustering Blockchain Protocools with Regards to Security Testing, Masterarbeit HTWK Leipzig, 2017. *

10. Privacy Preserving Record Linkage (PPRL)

Die Analyse von personenbezogenen Daten in Big-Data-Anwendungen muss mit dem Zielkonflikt zwischen dem Auffinden nützlicher Ergebnisse und der Einhaltung eines hohen Datenschutz-Standards (Privacy) umgehen, insbesondere dann, wenn Personendaten aus unterschiedlichen Quellen integriert und analysiert werden sollen. Privacy Preserving Record Linkage (PPRL) behandelt dieses Problem durch Verschlüsselung sensibler Attributwerte, so dass die Identifizierung der Personen verhindert wird, korrespondierende Datensätze aber trotzdem zugeordnet (gematcht) werden können, wenn sie das gleiche Realwelt-Objekt repräsentieren. Der Vortrag sollte auf folgende Aspekte eingehen:

- Record-Linkage / Entity Resolution Ansätze
- Verschlüsselung durch Bloom-Filter
- PPRL für metrische Räume: Dreiecksungleichung, Ähnlichkeits- und Abstandsmaße
- PPRL mit M-Tree
- PPRL mit Pivotelementen, Ansätze zur Verteilung und Parallelisierung
- Implementierung von PPRL mittels Apache Flink
- Performancebetrachtungen

Quellen:

- Ziad Sehili, Erhard Rahm: Speeding up Privacy Preserving Record Linkage for Metric Space Similarity Measures, in: Datenbank-Spektrum, Bd. 16, Nr. 3, S. 227-236. *
- Marcel Gladbach: Verteilte Verfahren für Privacy Preserving Record Linkage unter Verwendung von metrischen Räumen. Masterarbeit HTWK Leipzig, 2017. *

11. 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:

- http://cassandra.apache.org
- 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.
- J. Carpenter, E. Hewitt: Cassandra: The Definitive Guide, O'Reilly, 2nd edition, 2016.
- A. Lakshman, P. Malik: Cassandra - A Decentralized Structured Storage System

12. 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.

13. 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.*

14. 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.

15. Wissenschaftliche Datenbanken (Scientific Data Management)

Herkömmliche Datenbanken konzentrieren sich auf Verwaltung und Analyse von geschäftsorientierten Daten. Dem steht aber eine noch größere Menge an wissenschaftlichen Daten gegenüber, die bisher bei DB-Forschung und Entwicklung nicht ausreichend berücksichtigt wurden. Diese warden auch als Forschungsdaten bezeichnet. Die effiziente Verwaltung, Speicherung, Suche und Analyse wissenschaftlicher Daten stellt eine immense Herausforderung an diese und verwandte Bereiche der Naturwissenschaften dar. Wie kann man effektiv neues Wissen aus den Daten ableiten? Wo stoßen aktuelle Systeme an ihre Grenzen?
Durch den immensen Fortschritt bei der Instrumentierung von Experimenten, Simulation und Beobachtungen in allen Bereichen der Naturwissenschaften entstehen neue Herausforderungen für Datenbanktechnologien. Die bei Experimenten anfallenden Daten entstehen dabei oft schneller als sie verarbeitet werden können, was zu Bottlenecks führen kann. Wissenschaftliche Daten sind typischerweise sehr heterogen und komplex, erfordern neue Datenstrukturen und Zugriffsmuster. Dies bewirkt neue Aspekte der Zugriffsoptimierung und Datenintegration. Um rechenintensive und datenintensive Abläufe bei der Verarbeitung wissenschaftlicher Daten zu beschreiben, sind Scientific-Workflow-Technologien zu entwickeln, die sich von herkömmlichen business-orientierten Workflows unterscheiden. Dazu zählt insbesondere das Problem der Datenherkunft (Data Lineage). Der Vortrag sollte auf folgende Aspekte eingehen:

- Beispiele für naturwissenschaftliche Anwendungen: Biologie (Genetik, Moleekularbiologie), Astronomie, Meterologie,
- Wissenschaftliche Datenformate
- Bedeutung von Metadaten
- Data Provenance: Taxonomie und Techniken

Quellen:

- Mario Valle: Scientific Data Management. http://mariovalle.name/sdm/scientific-data-management.html

- J. Gray, D. Liu, M. Nieto-Santisteban, A. S. Szalay, D. DeWitt, G. Heber: Scientific Data Management in the Coming Decade, SIGMOD Record, Vol. 34 No. 4, 2005.
- Yogesh L. Simmhan, Beth Plale, Dennis Gannon: A Survey of Data Provenance Techniques. Technical Report IUB-CS-TR618.
- V. Cuevas-Vicenttin, S. Dey, S. Köhler, S. Riddle, B. Ludäscher: Scientific Workflows and Provenance: Introduction and Research Opportunities, Datenbank-Spektrum, Bd. 12, Heft 3, 2012. *

16. Data Management on New Hardware

Die Beschleunigung von Anwendungssystemen durch Anwendung von Moore's Law ist inzwischen an ihre Grenzen angelangt. Signifikante Leistungssteigerungen sind nur noch durch einen hohen Grad an Hardware-Spezialisierung möglich. Neuerungen in der Computersystem-Architektur ermöglichen Innovationen in der Architektur eines DBMS

Der Vortrag sollte folgende Schwerpunkte umfassen:

- Parallelität und Multi-Core-Prozessoren
- Grafikprozessoren (GPU) und programmierbare Logikbausteine (FPGAs)
- Neue Speichermedien (insb. SSD)
- Cache-Hierarchien
- NUMA Support in Datenbanken
- Energieeffizienz in Main-Memory-Datenbanken

Quellen:

- Jens Teubner u.a.: New Hardware Architectures for Data Management. Tutorial. BTW 2013.
- Annett Ungethüm, u.a.: Overview on Hardware Optimizations for Database Engines. Proceedings of the BTW 2017, http://btw2017.informatik.uni-stuttgart.de/
- Stefan Noll, Henning Funke, Jens Teubner: Energy Efficiency in Main-Memory-Databases. In: Datenbank-Spektrum Bd. 17, Heft 3, Nov. 2017, S. 223-232. *