Stratosphere: Informationsmanagement "above the Clouds"

Johann-Christoph Freytag[*] Odej Kao[+] Ulf Leser[*] Volker Markl[+] Felix Naumann[§]

[*]*Humboldt-Universität zu Berlin*
[+]*Technische Universität Berlin*
[§]*Hasso-Plattner-Institut, Potsdam*

{freytag, leser}@informatik.hu-berlin.de,
{odej.kao, volker.markl}@tu-berlin.de,
naumann@hpi.uni-potsdam.de

http://stratosphere.tu-berlin.de

Zusammenfassung

Dieser Artikel beschreibt die Vision und Ziele von Stratosphere, einer von der DFG geförderten Forschergruppe, in der fünf Fachgebiete an drei Universitäten in Berlin und Potsdam das Thema „Informationsmanagement in massiv-parallelen, virtualisierten Rechner-Infrastrukturen (*Cloud*)" untersuchen. Neben der Entwicklung eines Forschungsprototypen zur fehlertoleranten, verteilten, adaptiven Anfrageverarbeitung werden dabei auch Anwendungsfälle aus den Bereichen Lebenswissenschaften, Datenreinigung für Linked-Open-Daten und wissenschaftliches Rechnen untersucht. Stratosphere wird von der DFG von Oktober 2010 an für zunächst 3 Jahre gefördert.

1. Einführung

Cloud Computing entwickelt sich zum populärsten Paradigma für hochskalierbare, fehlertolerante und anpassungsfähige Rechner-Infrastrukturen. Es basiert auf großen Clustern, die aus „off-the-shelf" Rechnern bestehen und ihre Rechenleistung als Dienst (engl. *Service*) anbieten. Um eine solche Infrastruktur für die Bearbeitung großer Datenmengen effektiv und effizient nutzen zu können, ist es notwendig, existierende Technologien aus der Datenbankforschung an die neuen Gegebenheiten anzupassen, zu erweitern bzw. neue Konzepte und Techniken zu entwickeln.

Die DFG-geförderte Forschergruppe Stratosphere, in der fünf Forschungsgruppen dreier Berlin/Brandenburger Universitäten aus den Bereichen Datenbanksysteme und verteilte Systeme zusammenarbeiten, hat es sich zum Ziel gesetzt, die neuen Möglichkeiten des Cloud Computing – zusammen mit dem Map/Reduce-Programmiermodell – zum Management und zur Analyse sehr großer Datenmengen zu untersuchen. Dazu verfolgt Stratosphere einen datenbankinspirierten Ansatz, d.h. die durchzuführenden Analysen sollen in Form von deklarativen Anfragen formuliert und dann automatisch auf einer Cloud ausgeführt und parallelisiert werden. In Stratosphere sollen nicht nur neuartige Ansätze zur Bearbeitung komplexer Anfragen entwickelt werden, sondern diese sollen auch anhand einer prototypischen Realisierung auf ihre Tauglichkeit in einer Reihe ausgewählter Anwendungsfälle hin untersucht werden. Dieser Beitrag gibt im Folgenden einen Überblick über das Projekt, welches im Herbst 2010 startet.

2. Vision von Stratosphere

In der von der DFG als FOR 1306 geförderten Forschungsgruppe Stratosphere werden die beteiligten Fachgebiete DIMA und CIT der TU Berlin, DBIS und WBI der HU Berlin und IS des HPI Potsdam neue Techniken und Architekturen zur virtualisierten, parallelen und adaptiven Analyse sehr großer Datenbestände erforschen. Speziell wird sich Stratosphere – benannt nach dem Bereich der Atmosphäre oberhalb der Wolken – auf die

Einsatzmöglichkeiten von Cloud Computing zur Analyse von großen Datenmengen fokussieren. Dabei soll ein neuer, von Datenbanktechnologien inspirierter Ansatz zur Analyse, Aggregation und Verknüpfung sehr großer Mengen von textuellen und/oder (semi-) strukturierten Daten verfolgt werden. Die Entwicklung von Stratosphere wird dazu auf den folgenden Prinzipien basieren:

- Analyseprozesse werden als komplexe Datenflussprogramme formuliert. Stratosphere verfolgt einen deklarativen Ansatz zur Formulierung, Optimierung und Ausführung derartiger Programme. Diese werden u.a. mit Hilfe Funktionen höherer Ordnung beschrieben, die – eingebettet in einen an der funktionalen Programmierung orientierten Framework – auf einer Cloud verteilt, optimiert und ausgeführt werden können. Damit erweitert Stratosphere das im Cloud-Kontext populäre Map/Reduce-Programmiermodell.
- Stratosphere zielt auf komplexe und rechenintensive Analyseaufgaben auf großen Datenmengen. Um solche Prozesse in einer deklarativen Sprache formulieren zu können, wird Stratosphere neben den aus der relationalen Algebra bekannten Basisoperationen von Selektion, Projektion, Kartesisches Produkt/Join, Vereinigung und Differenz in einem Erweiterungsmodell auch komplexe Operationen zur Datenbereinigung, Datenintegration, Informationsextraktion, oder zum Data Mining entwickeln.
- Programme werden von Stratosphere auf einer dynamisch adaptierbaren Umgebung ausgeführt, die sowohl voraussichtliche Kosten als auch der geschätzte Nutzen einzelner Verarbeitungsschritte und ihrer Verkettung berücksichtigt. Veränderungen in der Ausführungsumgebung (Ressourcen, Daten, Parallelität) können jederzeit stattfinden, sogar während der Verarbeitung von Datenanalyseprogrammen (dynamische Optimierung).
- Stratosphere macht keine prinzipiellen Annahmen über die Art der zu analysierenden Daten. Die Analyse textueller und semi-strukturierter (XML) Daten wird von Stratosphere direkt durch entsprechende Operationen unterstützt.

3. Anwendungsfälle

Moderne Datenverarbeitungssysteme müssen oftmals sehr große Mengen strukturierter und unstrukturierter Daten analysieren und integrieren. Diese Daten können vielfältiger Natur sein; Beispiele reichen im strukturierten Bereich von RFID Daten für das *supply chain management* [GHL+06] über *data warehouses* [JLV+02] bis hin zu Simulationsergebnissen z.B. in den Naturwissenschaften [HJS+00] und Lebenswissenschaften [Har08]. Beispiele unstrukturierter Daten sind große Dokumentenbestände; dazu zählen sowohl öffentlich zugängliche Bestände (Publikationssammlungen, das Web etc.) als auch solche, die innerhalb von Unternehmen anfallen, wie z.B. Protokolle, Dokumentationen, E-Mails und Webseiten [RT07]. Nach [RT07] werden künftig weltweit und täglich mehrere TB an textuellen Daten produziert und in elektronischer Form gespeichert. Viele der genannten Daten sind in Dateien abgelegt und sind weder strukturiert noch bereinigt oder konsistent. In Stratosphere sollen vor allem drei Anwendungsfälle untersucht werden, die für Stratosphere als exemplarisch in ihrer Art und Größe angesehen werden:

Anwendungsfall 1: Scientific Data Management.

Wissenschaftliche Daten aus physikalischen Experimenten oder Simulationen erreicht in ihrer Größe oft den Tera-Byte- oder sogar Peta-Byte-Bereich [GLN+05]. Zudem sollen

oft mehrere solche Datenbestände aus verschiedenen Instituten integriert und gemeinsam analysiert werden. Typische Beispiele sind Klimasimulationen, wie sie beispielsweise am Potsdam Institut für Klimafolgenforschung (PIK) erfolgen. Ein einziger Simulationslauf, der sich über mehrere Monate über eine Teilregion des Planeten erstreckt und bestimmte Annahmen prüft, erzeugt bereits mehrere Tera-Byte an Daten. Für die meisten Forschungsfragen müssen Forscher am PIK für ihre Beantwortung Daten aus mehreren Simulationsläufen vergleichen, verknüpfen und aggregieren, die zur Bearbeitung von hunderten von Tera-Byte führt. Beispielsweise können so Vorhersagen über das Auftreten des Monsuns und ein Vergleich mit anderen Klimamodellen erfolgen.

Anwendungsfall 2: Analyse textueller Daten in den Lebenswissenschaften.

Bei der Entwicklung neuer Medikamente ist es äußerst hilfreich, so viele Informationen wie möglich zu dem neuen Wirkstoff und zu strukturell ähnlichen Wirkstoffen zu sammeln [AS08]. Zu diesem Zweck stehen große Sammlungen wissenschaftlicher Veröffentlichungen zur Verfügung (etwa PubMed mit ca. 20 Million Referenzen, oder die Public Library of Science mit über 1.5 Millionen frei verfügbaren Volltext-Artikeln[i]); wichtig sind darüber hinaus Daten aus anderen Webquellen wie Patentdatenbanken (die Volltext- und Bilddatenbank des *United States Patent and Trademark Office* enthalten ca. 7 Million Patente). Eine typische Aufgabe wäre die folgende: Ausgehend von einem aktuell untersuchten aussichtsreichen Wirkstoff sollen alle Erwähnungen des Wirkstoffes oder eines chemisch ähnlichen Stoffes in Verbindung mit medizinischen oder physiologischen (negativen) Effekten gesucht und in eine tabellarische Form gebracht werden. Darüber hinaus sollen Interaktionen mit anderen biochemischen Stoffen oder Eigenschaften der chemischen Struktur des Wirkstoffes gefunden und extrahiert werden [NTL10]. Die so gesammelten Informationen sollen aggregiert und nach Wirkstoff, Art des Assays (Zellkultur, in-vivo, in-silico, etc.), und Spezies/Gewebe sortiert werden. Da neue Wirkstoffe in der Regel noch nicht detailliert untersucht wurden, soll die Suche optional noch erweitert werden können, indem auch Texte analysiert werden, die solche Dokumente referenzieren, die den Wirkstoff direkt erwähnen. Extrahierte Daten müssen anschließend standardisiert, gesäubert und mit lokalen Datenbanken eines Pharmazieherstellers, die die eigenen experimentellen Ergebnisse enthalten, integriert werden [ZDF+08].

Anwendungsfall 3: Linked Open Data-Bereinigung.

Mehr und mehr Daten auch allgemeiner Natur werden im Web veröffentlicht. Jedoch bleibt es überaus schwierig, diese Daten zu integrieren und so einen Mehrwert zu erzeugen. Berners-Lee führte 2009 das Konzept des Linked Open Data (LOD) ein [BHB09]: Jedes Objekt erhält eine http-URI, deren Dereferenzierung mehr Informationen zu dem Objekt liefert. Zudem sollen Objekte möglichst viel miteinander verlinkt werden. Ein Geburtsort ist dann nicht mehr “Berlin”, sondern beispielsweise „http://dbpedia.org/resource/Berlin“. Daten, die diesem Konzept folgen, werden in der Regel als RDF Tripel öffentlich verfügbar gemacht. So entsteht zurzeit ein stetig wachsendes Netzwerk aus RDF-Daten über viele Datenquellen hinweg. Im Mai 2009 berichtete das W3C von mehr als 70 Datensammlungen, die über 4.7 Milliarden RDF-Tripel und 142 Million RDF links enthalten [W3C09]. Die Daten stammen aus vielen verschiedenen Domänen von öffentlichem, kommerziellem und wissenschaftlichem Interesse. Trotz des großen Potenzials ist die produktive Nutzung der Daten noch sehr stark durch die extreme strukturelle und semantische Heterogenität eingeschränkt. Die Heterogenität, sogar innerhalb einzelner Datenquellen, rührt von der oft manuellen Erzeugung von Struktur und

Daten her; ein prominentes Beispiel ist der DBpedia Datensatz, der aus Wikipedia extrahiert wird [BLK+09]. Anfragen über mehrere Datenquellen hinweg müssen zusätzliche Heterogenität überwinden, zum Beispiel in der Benennung von Prädikaten oder bei der Eindeutigkeit von Objekten. Ein weiteres Problem ist die Qualität der Daten selbst, die oft unsicher, unvollständig, unformatiert, inkonsistent, usw. sind. Die Erstellung einer effizienten Anfrageschnittstelle, die bei lesendem Zugriff (Browsen) bereinigte und konsistente Antwortmengen liefert, ist von großem Interesse.

4. Forschungsfragen

Stratosphere betreibt Grundlagenforschung in der Fragestellung, wie eine Cloud Computing-Architektur den Entwurf und die Anwendungen moderner Informationsmanagementsysteme beeinflusst. Dabei betrachtet die Forschergruppe Cloud Computing sowohl aus dem Blickwinkel der Datenbankforschung als auch dem Blickwinkel der verteilten Systeme.

In diesem Umfeld stehen folgende Forschungsfragen im Fokus von Stratosphere:

1. Welche Anforderungen muss ein deklaratives Daten- und Verarbeitungsmodell erfüllen, um komplexe, rechenintensive Operationen auf großen Datenmengen auszuführen? Wie können Daten und Metadaten effizient verwaltet werden? Wie kann Basisoperatoren dieses Modells in einer geschlossenen Algebra formalisiert werden?
2. Wie kann das Modell effizient auf eine Ausführungsumgebung in einer Cloud übersetzt werden? Inwiefern können bestehenden Cloud-Programmiermodellen wie das Map/Reduce-Programmiermodell und dem Key/Value-Datenmodell erweitert werden? Wie werden Operatoren implementiert und deren Ausführung parallelisiert, optimiert und dynamisch an Änderungen in der Ausführungsumgebung adaptiert?
3. Welche Konzepte sind nötig, um die Verarbeitung strukturierter und textueller Daten auf einer Cloud-Architektur zu ermöglichen? Wie werden Informationsextraktionsoperatoren parallelisiert und verarbeitet?
4. Welche Extraktions- und Datenreinigungsoperatoren sind für die genannten Anwendungsfälle nötig? Wie können diese effizient in einer virtualisierten Infrastruktur realisiert werden? Wie können Crawling und Extraktion kombiniert werden, um diese Operatoren auf demselben Prozessor einer Cloud auszuführen?
5. Welche potentiell komplexen, rechenintensiven Analyseoperatoren (z.B., Zeitreihenanalyse, Clustering) sollte ein Cloud basiertes Informationsverarbeitungssystem anbieten? Wie kann ein derartiges System flexibel mit weiteren Reinigungs- und Analyseoperatoren erweitert werden?
6. Wie betreibt man ein Cloud basiertes Informationsmanagementsystem fehlertolerant mit hoher Skalierbarkeit bei geringen Betriebskosten?
7. Aufgrund der komplexen, rechenintensiven Operationen auf großen Datenmengen verbietet sich der Ansatz von klassischen, zweistufigen Anfrageoptimierungsverfahren, die heutzutage Stand der Technik in Datenbanksystemen sind. Stattdessen ist eine robust initiale Planungskomponente erforderlich, die Ausführungsstrategien mit guter „worst-case“ Charakteristik erstellt. Diese Strategien können dann während der Anfrageverarbeitung kontinuierlich *adaptiert* werden, um bei Veränderungen der Ausführungsumgebung bessere Skalierbarkeit und Fehlertoleranz zu erzielen. Der Prototype von Stratosphere, bestehend aus einem Programmiermodell (PACT) und dynami-

schem Ausführungssubsystem (Nephele) wird eine derartige Verzahnung von Anfrageplanung und Ausführung realisieren.

5. Systemarchitektur

Die Grundlage für die Realisierung der genannten Anforderungen sind Hochleistungsressourcen, welche sich dynamisch an den aktuellen Bedarf einer Anwendung oder einer Anfrage anpassen können. Die in Abbildung 1 skizzierte Stratosphere-Architektur erlaubt zusammen mit der darunter liegenden Cloud Computing Software – derzeit auf der Basis von Eucalyptus [Euc10] – eine elastische Zuweisung von Rechenkapazität und Speicherplatz an den aktuellen Task.

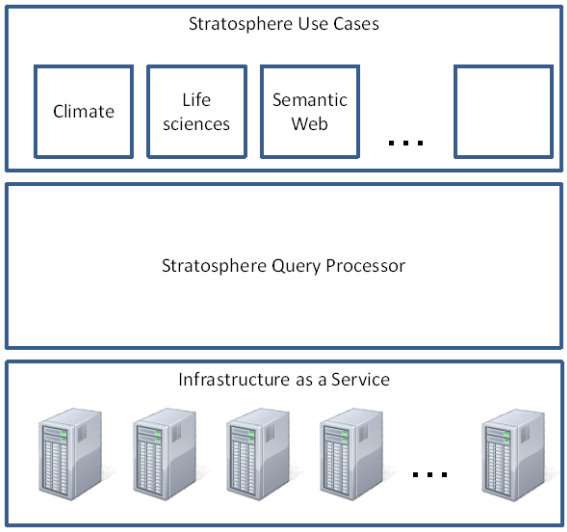

Abbildung 1: Überblick der Stratosphere-Architektur

Die unterste Ebene der Architektur von Stratosphere folgt der Idee der "*Infrastructure as a Service (IaaS)*" und setzt auf Eucalyptus auf, da diese Software kompatibel zu Amazons Elastic Compute Cloud (EC2), Simple Storage Service (S3) und Elastic Block Store (EBS) ist. Den Anfragen werden virtuelle Maschinen zugewiesen, deren Anzahl und Qualität sich an die aktuelle Systemlast anpassen können. Die darauf liegende Entwicklung *Nephele* führt die DAG-basierten Flussprogramme aus. Dazu werden die benötigten Ressourcen allokiert und zugewiesen sowie die erforderlichen Kommunikationskanäle aufgebaut. Damit sollen Skalierbarkeitseigenschaften für massiv-parallele Architekturen sowie eine kurzfristige Vergrößerung bzw. Verkleinerung der belegten Ressourcen erreicht werden. Schließlich sind eine kontinuierliche Überwachung des Ressourcenstatus und kurzlebige Checkpoints zum Erreichen von Fehlertoleranzeigenschaften integriert.

Auf dieser Basis wird der Stratosphere Anfrageprozessor aufgebaut, welcher ein geeignetes Programmiermodell für die effiziente, skalierbare und fehlertolerante Ausführung von Anfrageoperatoren zur Verfügung stellt. Neben der Ausführungsplattform Nephele [WK09] wird eine Speicherschicht implementiert, die zunächst aus Kompatibilitätsgründen auf dem Hadoop-Dateisystem HDFS aufsetzt. Oberhalb von *Nephele* und der Speicherschicht werden parallelisierbare Ausführungspläne in einem Programmiermodell mit Funktionen höherer Ordnung für Key/Value Werte realisiert, dem sogenannten. PACT Programmiermodell, einer Verallgemeinerung des Map/Reduce-Programmiermodells [ABE+10, BEH+10]. Darüber befindet sich die Datenmodell – Ebene, welche ein semantisch angereichertes Programmiermodell umsetzt. Insbesondere werden hier Konzepte für

den Umgang mit Unsicherheiten und Operatoren für Informationsextraktion und Informationsintegration realisiert.

Die einzelnen Schritte zur Verarbeitung der Daten werden mittels der deklarativen Programmiersprache spezifiziert, die von JAQL, HIVE und Pig inspiriert ist. Programme werden dann in die interne Darstellung des darunterliegenden Datenflussmodells überführt. Anschließend wird ein logischer Optimierer diese Darstellung unter Verwendung algebraischer Regeln und Metriken für Leistungsfähigkeit und Robustheit umschreiben und optimieren. Schließlich wird das gewählte Datenflussprogramm in einen parallelen Ausführungsplan aus Funktionen zweiter Ordnung (sog. PACTs) überführt, der den Kontroll- und Datenfluss beschreibt.

Dieser Plan wird auf eine Menge zulässiger *Nephele*-Schedules abgebildet. Die Auswahl des besten Schedules erfolgt mit einer kostenbasierten Zielfunktion unter Berücksichtigung der von *Nephele* ermittelten Informationen zur Topologie und Auslastung. *Nephele* ergänzt den Plan weiterhin um Funktionen zur Sicherung der Fehlertoleranz und führt ihn dann aus. Die Ausführung wird, basierend auf kontinuierlich erhobenen Monitoringinformationen, dynamisch, auch um auf Änderungen in der Verfügbarkeit von Ausführungsplattformen in der Cloud zu reagieren. Die Adaption kann den *Nephele* Scheduler zur die Re-Optimierung auf verschiedenen Granularitätsebenen veranlassen, etwa auf Ebene der PEPs oder des SOPREMO-Programms.

6. Verwandte Arbeiten

Die Vision für Stratosphere fußt auf einer Reihe prominenter Vorarbeiten. Eine wichtige Wurzel ist das *Cloud Computing*-Paradigma, welches seit einigen Jahren viele Diskussionen über zukünftige IT Architekturen beherrscht. Es wird dabei aber eher unter wirtschaftsinformatischen als informatischen Gesichtspunkten erörtert [BYV+08]. Viele Eigenschaften unseres Ansatzes, wie die Flexibilität in der Ressourcenzuweisung, die hochparallele Verarbeitung von Daten, oder die einfachen Schnittstellen für komplexe Dienste, sind eng mit dem Begriff des *Cloud Computing* verbunden. Unsere Vision geht aber über den heutigen Stand der Technik weit hinaus, da wir deklarativ beschriebene Analyseprozesse automatisch in einer Cloud verteilen wollen, also auch eine adaptive Lastbalancierung anstreben.

In den letzten Jahren wurde eine Reihe von Ansätzen entwickelt, die auf eine datenflussorientierte Beschreibung von Analyseprozessen zielen. Besonders populär ist der Map/Reduce-Ansatz [DG04] sowie seine frei verfügbare Implementierung in Hadoop [Had10], der auf dem Grundsatz basiert, derartige Analysen als Sequenzen von Operationen zu beschreiben, die als Funktionen für die generischen Operationen Map und Reduce implementiert werden. Das Framework verteilt die Ausführung der Prozesse dann über einen Rechencluster und übernimmt automatisch wichtige Funktionalitäten wie Fehlererholung, Datenverteilung, oder Logging. Aufbauend auf dem Map/Reduce-Paradigma gibt es eine Reihe von Erweiterungen, wie MapReduceMerge, mit dem auch Verknüpfungen heterogener Eingabeströme möglich werden [YDH+07]. Im Unterschied zu diesen Arbeiten, in der die Analysefunktionen und deren Datenfluss immer noch in einer Host-Sprache programmiert und im Kern statisch verteilt werden müssen, verfolgt Stratosphere die Idee einer Analysesprache, die eine reiche Menge von Grundoperatoren zur Verfügung stellt und deren Ausführung automatisch optimiert wird. Projekte, die in eine ähnliche Richtung zielen, sind zum Beispiel DryadLinQ [YIF+08]. Im Unterschied zu diesen Ansätzen adressiert Stratosphere neben strukturierten auch unstrukturierte und semi-

strukturierte Daten und erlaubt die parallele Ausführung von Anfragen auf einer virtualisierten Infrastruktur mit Robustheit, die durch Adaption an die Umweltbedingungen der Ausführungsumgebung erreicht wird. In diesem Bereich gibt es Ähnlichkeiten mit jüngsten Vorschlägen zur Beschreibung von Informationsextraktionsprozessen als deklarative Anfragen [RRK+08, SDN+07]. die aber den Aspekt der adaptiven Parallelität noch vollkommen außer Acht lassen.

7. Zusammenfassung und Ausblick

Die von der DFG von 2010 bis 2013 geförderte Forschergruppe Stratosphere zielt auf die grundlegende Untersuchung von Methoden zur Analyse von riesigen Datenmengen auf einer massiv-parallelen, virtualisierten, adaptiven Infrastruktur ab. Dabei werden für Anwendungen der Naturwissenschaften, der Textanalyse und im Bereich des Data Cleansing Programmiermodelle sowie fehlertolerante und robuste Anfrageverarbeitungsalgorithmen untersucht. Mittelfristig ist es geplant, in diesem Kontext und auf dieser Platform betriebswirtschaftliche Fragen, Fragen des Datenschutzes und der Datensicherheit, insbesondere im Hinblick auf Multi-Tenancy, zu betrachten. Weitere offene Fragen stellen sich hinsichtlich Data Streaming und der damit verbundenen Echtzeitanforderungen.

8. Referenzen

[ABE+10] A. Alexandrov, D. Battré, S. Ewen, M. Heimel, F. Hueske, O. Kao, V. Markl, E. Nijkamp, D. Warneke: Massively Parallel Data Analysis with PACTs on Nephele. PVLDB 3(2): 1625-1628 (2010)

[AS08] P. Agarwal, D.B. Searls: Literature mining in support of drug discovery. Brief Bioinform 9(6): 479-92 (2008)

[BYV+08] R.Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I.Brandic: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation Computer Systems 25(6): 599-616 (2009)

[BEH+10] D. Battré, S. Ewen, F. Hueske, O. Kao, V. Markl, D. Warneke: Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. SoCC 2010: 119-130

[BHB09] C. Bizer, T. Heath, T. Berners-Lee: "Linked Data - The Story So Far", International Journal on Semantic Web and Information Systems, Vol. 5(3), 2009: 1-22

[BLK+09] C. Bizer, J. Lehmann, S. A. Georgi Kobilarov, C. Becker, R. Cyganiak, S. Hellmann: “DBpedia - a crystallization point for the web of data”, Journal of Web Semantics (JWS) 7(3), 2009

[DG04] J. Dean, S.Ghemawat: MapReduce: Simplified Data Processing on Large Clusters, 6th Symposium on Operating System Design and Implementation, San Francisco, USA, 137-150 (2004).

[Euc10] http://open.eucalyptus.com/ (last access: Oct. 4th, 2010)

[Had10] http://hadoop.apache.org/ (last access: Oct. 4th, 2010)

[Har08] G. Hardiman: Ultra-high-throughput sequencing, microarray-based genomic selection and pharmacogenomics, Pharmacogenomics and Future Medicine 9(1) (2008)

[HJS+00] W. Hotschek, F. Jaen-Martinez, A. Samar, H. Stockinger, K. Stockinger: Data Management in an International Data Grid Project, LNCS Vol. 1971, 2000: 77-90 (1971)

[GHL+06] H. Gonzalez, J. Han, X. Li, D. Klabjan: Warehousing and analyzing massive RFID Data Sets ICDE 2006: 83 (2006)

[GLN+05] J. Gray, D. T. Liu, M. A. Nieto-Santisteban, A. S. Szalay, D. J. DeWitt, G. Heber: Scientific data management in the coming decade, SIGMOD Record Vol. 34(4), 2005: 34-41 (2005)

[JLV+02] M. Jarke, M. Lenzerini, Y. Vassilou, P. Vassiliadis: "Fundamentals of Data Warehouses", Springer, Berlin, (2002)

[NTL10] Ngyuen, L. Q., Tikk, D. and Leser, U. (2010). "Simple Tricks for Improving Pattern-Based Information Extraction from the Biomedical Literature." Journal of Biomedical Semantics 2010, 1:9.

[RRK+08] Reiss, F., Raghavan, S., Krishnamurthy, R., Zhu, H. and Vaithyanathan, S. (2008). "An Algebraic Approach to Rule-Based Information Extraction". 24th International Conference on Data Engineering, Cancun, Mexico. pp 933-942.

[RT07] R. Ramakrishnan, A. Tomkins: "Towards a PeopleWeb", IEEE Computer Vol. 40(8), 2007: 63-72

[SDN+07] Shen, W., Doan, A., Naughton, J. F. and Ramakrishnan, R. (2007). "Declarative Information Extraction Using Datalog with Embedded Extraction Predicates". Int Conf. on Very Large Databases, Vienna, Austria. pp 1033-1044.

[YDH+07] Yang, H.-C., Dasdan, A., Hsiao, R.-L. and Parker, D. S. (2007). "Map-reduce-merge: simplified relational data processing on large clusters". SIGMOD Conference, Beijing, CHina. pp 1029-1040.

[YIF+08] Yu, Y., Isard, M., Fetterly, D., Budiu, M., Erlingsson, Ú., Gunda, P. K. and Currey, J. (2008). "DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language". Symposium on Operating Systems Design and Implementation.

[W3C09] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData (last access: Nov. 30th, 2009)

[WK09] Daniel Warneke, Odej Kao: Nephele: efficient parallel data processing in the cloud. SC-MTAGS 2009

[ZDF+08] Zweigenbaum, P., Demner-Fushman, D., Yu, H. and Cohen, K. B. (2007). "Frontiers of biomedical text mining: current progress." Brief Bioinform 8(5): 358-75.

[i] http://www.ncbi.nlm.nih.gov/pubmed/; *http://www.plos.org/*