

# Information Management in the Cloud

Edited by

Anastassia Ailamaki<sup>1</sup>, Michael J. Carey<sup>2</sup>, Donald Kossmann<sup>3</sup>,  
Steve Loughran<sup>4</sup>, and Volker Markl<sup>5</sup>

- 1 EPFL – Lausanne, CH, [anastasia.ailamaki@epfl.ch](mailto:anastasia.ailamaki@epfl.ch)
- 2 University of California – Irvine, US, [mjcarey@ics.uci.edu](mailto:mjcarey@ics.uci.edu)
- 3 ETH Zürich, CH, [donald.kossmann@inf.ethz.ch](mailto:donald.kossmann@inf.ethz.ch)
- 4 HP Lab – Bristol, GB, [steve.loughran@hp.com](mailto:steve.loughran@hp.com)
- 5 TU Berlin, DE, [volker.markl@tu-berlin.de](mailto:volker.markl@tu-berlin.de)

---

## Abstract

Cloud computing is emerging as a new paradigm for highly scalable, fault-tolerant, and adaptable computing on large clusters of off-the-shelf computers. Cloud architectures strive to massively parallelize complex processing tasks through a computational model motivated by functional programming. They provide highly available storage and compute capacity through distribution and redundancy. Most importantly, Cloud architectures adapt to changing requirements by dynamically provisioning new (virtualized) compute or storage nodes. Economies of scale enable cloud providers to provide compute and storage powers to a multitude of users. On the infrastructure side, such a model has been pioneered by Amazon with EC2, whereas software as a service on cloud infrastructures with multi-tenancy has been pioneered by Salesforce.com.

The Dagstuhl Seminar 11321 “Information Management in the Cloud” brought together a diverse set of researchers and practitioners with a broad range of expertise. The purpose of this seminar was to consider and to discuss causes, opportunities, and solutions for technologies, and architectures that enable cloud information management. The scope ranged from web-scale log file analysis using cluster computing techniques to dynamic provisioning of resources in data centers, covering topics from the areas of analytical and transactional processing, parallelization of large scale data and compute intensive operations as well as implementation techniques for fault tolerance.

**Seminar** 07.–12. August, 2011 – [www.dagstuhl.de/11321](http://www.dagstuhl.de/11321)

**1998 ACM Subject Classification** H.0 [Information Systems] General

**Keywords and phrases** Cloud Technologies, Information Management, Distributed Systems, Parallel Databases

**Digital Object Identifier** 10.4230/DagRep.1.8.1



Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license

Information Management in the Cloud, *Dagstuhl Reports*, Vol. 1, Issue 8, pp. 1–28

Editors: Anastassia Ailamaki, Michael J. Carey, Donald Kossmann, Steve Loughran, and Volker Markl



DAGSTUHL  
REPORTS

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Executive Summary


*Anastassia Ailamaki*

*Michael J. Carey*

*Donald Kossmann*

*Steve Loughran*

*Volker Markl*

**License**  Creative Commons BY-NC-ND 3.0 Unported license  
 © Anastassia Ailamaki, Michael J. Carey, Donald Kossmann, Steve Loughran,  
 Volker Markl

Cloud computing is emerging as a new paradigm for highly scalable, fault-tolerant, and adaptable computing on large clusters of off-the-shelf computers. Cloud architectures strive to massively parallelize complex processing tasks through a computational model motivated by functional programming. They provide highly available storage and compute capacity through distribution and redundancy. Most importantly, Cloud architectures adapt to changing requirements by dynamically provisioning new (virtualized) compute or storage nodes. Economies of scale enable cloud providers to provide compute and storage powers to a multitude of users. On the infrastructure side, such a model has been pioneered by Amazon with EC2, whereas software as a service on cloud infrastructures with multi-tenancy has been pioneered by Salesforce.com.

The Dagstuhl seminar on Information Management in the Cloud brought together a diverse set of researchers and practitioners with a broad range of expertise. The purpose of this seminar was to consider and to discuss causes, opportunities, and solutions for technologies, and architectures that enable cloud information management. The scope ranged from web-scale log file analysis using cluster computing techniques to dynamic provisioning of resources in data centers, covering topics from the areas of analytical and transactional processing, parallelization of large scale data and compute intensive operations as well as implementation techniques for fault tolerance.

The seminar consisted of keynotes, participant presentations, demos and working groups. The first two seminar days consisted of a keynote by Helmut Krcmar on “Business Aspects of Cloud Computing” as well as 33 short presentations on various aspects of cloud computing. On the evening of the second day, the participants formed working groups on economic aspects, programming models, benchmarking. The third day of the seminar consisted of two keynotes, by Dirk Riehle on “Open Source and Cloud Computing” and by Donald Kossmann on “Benchmarking”. After these keynotes, working groups discussed their respective topics. In the evening, an industrial panel with Miron Livny, Steve Loughran, Sergey Melnik, Russell Sears, and Dean Jacobs discussed research challenges in Cloud Computing from an industrial point of view. On the fourth day, a keynote by Miron Livny discussed Cloud Computing from a distributed systems and high-performance computing point of way. After the keynote, a demo session presented the following systems:

- HyPer: A Cloud-scale Main Memory Database System (Team from TUM)
- Asterix and Hyrax (Team from UCI)
- Stratosphere (Team from TU Berlin, HU Berlin and HPI)
- Myriad Parallel Data Generator (Team from TU Berlin)

After these demos, working groups continued during the day and presented their results in the evening. The last day of the seminar, participants continued in working groups and discussed further collaborations with respect to papers and project proposals. During this

day, several abstracts for papers have been prepared, and discussions about several joint research project proposals have started.

The organizers hope that the seminar has helped to organize the research space in cloud computing and identified new research challenges. We look forward towards research collaborations and papers that were bootstrapped during this intensive week.

## 2 Table of Contents

### Executive Summary

<i>Anastassia Ailamaki, Michael J. Carey, Donald Kossmann, Steve Loughran, Volker Markl</i> . . . . .	2
---	---

### Overview of Talks

Facilitating Scientific Analytics in the Cloud <i>Magdalena Balazinska</i> . . . . .	7
Web Data Cleaning <i>Felix Naumann</i> . . . . .	7
Cloud Computing Support for Massively Social Gaming <i>Alexandru Iosup</i> . . . . .	8
Genome Data Preprocessing with MapReduce <i>Keijo Heljanko</i> . . . . .	9
HyPer: Hybrid OLTP & OLAP High-Performance Database System <i>Alfons Kemper</i> . . . . .	9
Making Sense at Scale with Algorithms, Machines & People <i>Tim Kraska</i> . . . . .	11
Optimization of PACT Programs <i>Fabian Hueske</i> . . . . .	11
The ASTERIX Project: Cloudy DB Research at UC Irvine <i>Mike Carey</i> . . . . .	12
Algebricks + Hyracks: An efficient Data-Centric Virtual Machine for the Cloud <i>Vinayak Borkar</i> . . . . .	12
Extending Map-Reduce for Efficient Predicate-Based Sampling <i>Raman Grover</i> . . . . .	13
Challenges for Cloud Benchmarking <i>Enno Folkerts</i> . . . . .	13
Trade-Offs in Cloud Application Architecture <i>Stephan Tai</i> . . . . .	14
Benchmarking Large-Scale Parallel Processing Systems <i>Alexander Alexandrov</i> . . . . .	14
To Cloud or Not To. Musings on Cloud Deployment Viability and Cost Models <i>Radu Sion</i> . . . . .	14
Building Large XML Stores in the Amazon Cloud <i>Jesus Camacho-Rodriguez</i> . . . . .	15
Storage for End-user Programming <i>Dirk Riehle</i> . . . . .	15
Adaptive Query Processing in Stratosphere <i>Johann-Christoph Freytag</i> . . . . .	15
Nephele: (Cost) Efficient Parallel Data Flows in the Cloud <i>Daniel Warneke</i> . . . . .	16

Information Extraction in Stratosphere <i>Astrid Rheinlaender</i> . . . . .	16
Cloud Computing and Next Generation Sequencing <i>Ulf Leser</i> . . . . .	16
Cost-aware data management in the cloud <i>Verena Kantere</i> . . . . .	17
Building high performance indexes for key value storage <i>Russell Sears</i> . . . . .	17
MuTeDB - A dbms that shows quiet on multi-tenancy <i>Bernhard Mitschang</i> . . . . .	17
Yes, but does it work? <i>Steve Loughran</i> . . . . .	18
ScalOps: Cloud Computing in a High-Level Programming Language <i>Tyson Condie</i> . . . . .	18
Cloud-based Web data management (it's all about how you view it) <i>Ioana Manolescu</i> . . . . .	19
<b>Overview of Demos</b>	
Asterix <i>Vinayak Borkar, Raman Grover</i> . . . . .	20
Stratosphere <i>Daniel Warneke, Fabian Hueske</i> . . . . .	20
Parallel Data Generation with Myriad <i>Alexander Alexandrov</i> . . . . .	20
<b>Break-Out Group Reports</b>	
Cloud Benchmarking <i>Alexandru Iosup, Alexander Alexandrov, Enno Folkerts, Donald Kossmann, Seif Haridi, Volker Markl, Tim Kraska, Radu Sion, Anastasia Ailamaki, Dean Jacobs</i> . . . . .	21
Biomedical Analytics in the Cloud <i>Jim Dowling, Johann-Christoph Freytag, Keijo Heljanko, Ulf Leser, Felix Naumann, Astrid Rheinländer</i> . . . . .	22
Data and Programming Models <i>Vinayak Borkar, Jesus Camacho-Rodriguez, Mike Carey, Tyson Condie, Raman Grover, Arvid Heise, Fabian Hueske, Dean Jacobs, Steve Loughran, Ioana Manolescu-Goujot, Sergey Melnik, Bernhard Mitschang, Daniel Warneke</i> . . . . .	23
Transactions in the Cloud <i>A. Ailamaki, S. Haridi, A. Kemper, T. Kraska, S. Loesing, S. Melnik, R. Sears</i> . . . . .	26
Cloud Economics <i>Verena Kantere, Magdalena Balazinska, Athanasios Papaioannou, Helmut Kremer, Miron Livny</i> . . . . .	26
Cloud Storage <i>Anastasia Ailamaki, Russell Sears</i> . . . . .	26

**6 11321 – Information Management in the Cloud**





Participants . . . . . 28

### 3 Overview of Talks

This section lists the talks and abstracts of all seminar participants. The titles and abstracts were taken from the seminar's material web site whenever available.

#### 3.1 Facilitating Scientific Analytics in the Cloud

*Magdalena Balazinska (University of Washington – Seattle, US)*

License     Creative Commons BY-NC-ND 3.0 Unported license  
© Magdalena Balazinska

Sciences are becoming increasingly data rich and data analysis is becoming the bottleneck to discovery. The cloud holds the promise to facilitate large-scale data analysis because it provides easy access to compute resources and data management software with a flexible pay-as-you-go charging mechanism. There are, however, several challenges in leveraging the cloud for scientific analytics. We discuss three challenges in this talk. First, it is extremely challenging to get high-performance from today's data management systems out-of-the box. Second, data management systems can be hard to use even after the cloud takes away the installation and operations tasks. Finally, the interplay between data management and cloud economics raise several interesting new challenges and opportunities. In this talk, we will explain these three challenges and present recent research results from the database group at the University of Washington related to addressing them.

#### 3.2 Web Data Cleaning


*Felix Naumann (Hasso Plattner Institut – Potsdam, DE)*

License     Creative Commons BY-NC-ND 3.0 Unported license  
© Felix Naumann

The wealth of freely available, structured information on the Web is constantly growing. Driving domains are public data from and about governments and administrations, scientific data, and data about media, such as articles, books and albums. In addition, general-purpose datasets, such as DBpedia and Freebase from the linked open data community, serve as a focal point for many data sets. Thus, it is possible to query or integrate data from multiple sources and create new, integrated data sets with added value. Yet integration is far from simple: It happens at technical level by ingesting data in various formats, at structural level by providing a common ontology and mapping the data source structures to it, and at semantic level by linking multiple records about same real world entities and fusing these representations into a clean and consistent record. The talk highlights the extreme heterogeneity of web data and points to three research directions: (i) Domain-specific Integration Projects, such as govwild.org, (ii) ad-hoc and declarative data cleansing, such as in the Stratosphere project, and (iii) dynamic provisioning of Linked Data in a Data as a Service (DaaS) fashion.

### 3.3 Cloud Computing Support for Massively Social Gaming

*Alexandru Iosup (TU Delft, NL)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Alexandru Iosup

Cloud computing is an emerging commercial infrastructure paradigm that promises to eliminate the need for maintaining expensive computing hardware. Through the use of virtualization and resource time-sharing, Infrastructure as a Service (IaaS) clouds address with a single set of physical resources a large user base with different needs. Similarly, Platform as a Service (PaaS) clouds focus on providing platforms that address each the needs of a large and varied community. Thus, clouds promise to enable for their owners the benefits of an economy of scale and, at the same time, reduce the operating costs for many applications. For example, clouds may become for scientists an alternative to clusters, grids, and parallel production environments. In this presentation we focus on three main research questions related to cloud computing:

1. What is the performance of virtualized cloud resources, as perceived by their users? Many production clouds, including some of the largest publicly-accessible commercial clouds such as the Amazon Web Services and the Google App Engine, use virtualized resources to address diverse user requirements with the same set of physical resources. Virtualization can introduce performance penalties, either due of the additional middleware layer or to the interaction of workloads belonging to different virtual machines. Do virtualized resources deliver the same performance regardless of the application? In particular, are applications affected by execution on virtualized resources? We present here our findings from a large-scale performance evaluation study that focuses on four commercial IaaS clouds.
2. What guarantees do we have about the good performability of clouds over long periods of time? A major impediment to cloud adoption at large is their perceived instability, due, in lack of hard evidence, to novelty ("clouds are a technology too immature to be reliable"). Even if a cloud is available and works well today, it may well happen that it will not tomorrow. Does performance change over time (for the worse)? Are clouds really available all the time? We present here our findings from a long-term performance evaluation study that focuses on two commercial clouds, one IaaS and one PaaS.
3. Which new applications can make use of clouds? (By new applications we understand applications with a workload different from the applications of the past, including the workloads typical for grid computing.) Commercial clouds are new to the public. What applications that we could not previously afford to run are now enabled by clouds? What applications can function well under the availability and performance profiles of the current production cloud services? We focus in this presentation on Massively Multiplayer Online Games (MMOGs) and Massively Social Games (MSGs), which have recently emerged as a novel Internet-based entertainment application. Hundreds of MMOGs and MSGs already serve over a quarter of a billion paying customers world-wide, with virtual worlds such as World of Warcraft, FarmVille, and Runescape hosting daily several millions of players. These players want fast-paced entertainment delivered through the Internet, which raises important content and resource requirements; when these are not met in full and on time, players are likely to quit. However, the current industry approach in addressing these requirements, of building and maintaining large data centers, has high cost and limited scalability. The high cost makes the market inaccessible for amateur and small game developers. The limited scalability means that even the largest game






developers cannot support this rapidly growing community. We present our early results in understanding if IaaS and PaaS clouds can provide a scalable, dependable, yet low-cost computational technology for MMOGs and MSGs.

The loosely coupled team who has done the work presented here is: Undergraduate Students at TU Delft: Martin Biczak, Arnoud Bakker, Nassos Antoniou, Thomas de Ruiter, etc. Graduate students at TU Delft: Siqi Shen, Nezhir Yigitbasi, Ozan Sonmez. Staff at TU Delft: Henk Sips, Dick Epema, Alexandru Iosup. Collaborators Ion Stoica and the Mesos team (UC Berkeley), Vlad Nae, Thomas Fahringer, Radu Prodan (U. Innsbruck), Nicolae Tapus, Mihaela Balint, Ad. Lascateu, Vlad Posea (UPB), Derrick Kondo, Emmanuel Jeannot (INRIA), etc.

### 3.4 Genome Data Preprocessing with MapReduce




*Keijo Heljanko (Helsinki University of Technology, FI)*

License    Creative Commons BY-NC-ND 3.0 Unported license  
© Keijo Heljanko

We describe joint work between Aalto University and CSC done to visualize genomic data using our new preprocessing tool based on the MapReduce programming framework. The work uses the Apache Hadoop system to build a tool for preprocessing sequence alignment map in BAM file format resulting in an opensource tool Hadoop-BAM. We also describe future directions on research in the area.

### 3.5 HyPer: Hybrid OLTP & OLAP High-Performance Database System

*Alfons Kemper (TU München, DE)*

License    Creative Commons BY-NC-ND 3.0 Unported license  
© Alfons Kemper

The HyPer prototype demonstrates that it is indeed possible to build a main-memory database system that achieves world-record transaction processing throughput and best-of-breed OLAP query response times in one system in parallel on the same database state. The two workloads of online transaction processing (OLTP) and online analytical processing (OLAP) present different challenges for database architectures. Currently, users with high rates of mission-critical transactions have split their data into two separate systems, one database for OLTP and one so-called data warehouse for OLAP. While allowing for decent transaction rates, this separation has many disadvantages including data freshness issues due to the delay caused by only periodically initiating the Extract Transform Load-data staging and excessive resource consumption due to maintaining two separate information systems. We present an efficient hybrid system, called HyPer, that can handle both OLTP and OLAP simultaneously by using hardware-assisted replication mechanisms to maintain consistent snapshots of the transactional data (see the figure on the right). HyPer is a main-memory database system that guarantees the full ACID properties for OLTP transactions and executes OLAP query sessions (multiple queries) on arbitrarily current and consistent snapshots. The utilization of the processor-inherent support for virtual memory management (address translation, caching, copy-on-write) yields both at the same time: unprecedentedly high transaction rates as high

as 100000 per second and very fast OLAP query response times on a single system executing both workloads in parallel. The performance analysis is based on a combined TPC-C and TPC-H benchmark.

We have developed the novel hybrid OLTP & OLAP database system HyPer that is based on snapshotting transactional data via the virtual memory management of the operating system. In this architecture the OLTP process owns the database and periodically (e.g., in the order of seconds or minutes) forks an OLAP process. This OLAP process constitutes a fresh transaction consistent snapshot of the database. Thereby, we exploit operating systems functionality to create virtual memory snapshots for new, cloned processes. In Unix, for example, this is done by creating a child process of the OLTP process via the fork system call.


The forked child process obtains an exact copy of the parent processes address space. This virtual memory snapshot that is created by the fork-operation will be used for executing a session of OLAP queries. These queries can be executed in parallel threads or serially, depending on the system resources or client requirements. In essence, the virtual memory snapshot mechanism constitutes a OS/hardware supported shadow paging mechanism as proposed decades ago for disk-based database systems. However, the original proposal incurred severe costs as it had to be software-controlled and it destroyed the clustering on disk. Neither of these drawbacks occurs in the virtual memory snapshotting as clustering across RAM pages is not an issue. Furthermore, the sharing of pages and the necessary copy-on-update/write is managed by the operating system with effective hardware support of the MMU (memory management unit) via the page table that translates VM addresses to physical pages and traps necessary replication (copy-on-write) actions. Therefore, the page replication is extremely efficiently done in  $2\mu\text{s}$  as we measured in a micro-benchmark.

HyPer's OLTP throughput is better than VoltDB's published TPC-C performance and HyPer's OLAP query response times are superior to MonetDB's query response times. It should be emphasized that HyPer can match (or beat) these two best- of-breed transaction (VoltDB) and query (MonetDB) processing engines at the same time by performing both workloads in parallel on the same database state. HyPer's performance is due to the following design:

- HyPer relies on in-memory data management without the ballast of traditional database systems caused by DBMS-controlled page structures and buffer management. The SQL table definitions are transformed into simple vector-based virtual memory representations – which constitutes a column oriented physical storage scheme.
- The OLAP processing is separated from the mission-critical OLTP transaction processing by fork-ing virtual memory snapshots. Thus, no concurrency control mechanisms are needed – other than the hardware-assisted VM management – to separate the two workload classes.
- Transactions and queries are specified in SQL and are efficiently compiled into efficient LLVM assembly code.
- As in VoltDB, the parallel transactions are separated via lock-free admission control that allows only non-conflicting transactions at the same time.
- HyPer relies on logical logging where, in essence, the invocation parameters of the stored (transaction) procedures are logged via a high-speed network.

### 3.6 Making Sense at Scale with Algorithms, Machines & People

*Tim Kraska (University of California – Berkeley, US)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Tim Kraska

The creation, analysis, and dissemination of data have become profoundly democratized. Social networks spanning 100s of millions of users enable instantaneous discussion, debate, and information sharing. Streams of tweets, blogs, photos, and videos identify breaking events faster and in more detail than ever before. Deep, on-line datasets enable analysis of previously unreachable information. This sea change is the result of a confluence of Information Technology advances such as: intensively networked systems, cloud computing, social computing, and pervasive devices and communication.

The key challenge is that the massive scale and diversity of this continuous flood of information breaks our existing technologies. State-of-the-art Machine Learning algorithms do not scale to massive data sets. Existing data analytics frameworks cope poorly with incomplete and dirty data and cannot process heterogeneous multi-format information. Current large-scale processing architectures struggle with diversity of programming models and job types and do not support the rapid marshalling and unmarshalling of resources to solve specific problems. All of these limitations lead to a Scalability Dilemma: beyond a point, our current systems tend to perform worse as they are given more data, more processing resources, and involve more people, exactly the opposite of what should happen.

To address these issues, we are starting a new five-year, multi-faculty research effort called the AMPLab, where AMP stands for "Algorithms, Machines, and People". AMPLab envisions a world where massive data, computing, communication and people resources can be continually, flexibly and dynamically be brought to bear on a range of hard problems by huge numbers of people connected to the cloud via mobile and other client devices of increasing power and sophistication. In this talk, I will give an overview of the AMPLab motivation and research agenda and discuss several of our initial projects. One such project, PIQL, is a declarative query language that also provides scale-independence in addition to data-independence by calculating an upper bound on the number of key/value store operations that will be performed for any query. Coupled with a service level objective (SLO) compliance prediction model and PIQL's scalable database architecture, these bounds make it easy for developers to write applications that support an arbitrarily large number of users while still providing acceptable and predictable performance.

### 3.7 Optimization of PACT Programs

*Fabian Hueske (TU Berlin, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Fabian Hueske


The PACT Programming Model is a generalization and extension of the well-known MapReduce Programming Model. Both models have a common ground: they use a key-value pair data model and are based on parallelizable second-order functions which call first-order user functions. While MapReduce offers only two of such second-order functions Map and Reduce, PACT has an extended set of parallelization primitives that also handle multiple inputs. Furthermore, PACT supports so-called Output Contracts which are annotations that reveal

certain characteristics of the black-box user code. Finally, PACT programs are composed as arbitrary acyclic graphs. In contrast, MapReduce jobs have a static structure.

Data processing tasks implemented in PACT are compiled into parallel data flows. During this step, some degrees of freedom enable the compiler to perform physical optimization. These opportunities come from the declarative character of the parallelization primitives and knowledge that is derived from user code annotations. The compiler performs cost-based optimization and aims to reduce network and disk I/O. It chooses shipping (broadcast vs. repartition) and local strategies (sort-merge join vs. hash join) and reuses of existing physical data properties. In this regard the compiler is very similar to the physical optimizer of a traditional PDBMS. However, in contrast to well-defined SQL queries, PACT programs are arbitrary data flows solely consisting of UDFs. The talk concludes by giving a short overview of upcoming features of the PACT programming model and motivates the need for robust optimization in the context of massively parallel analytics in cloud environments.

### 3.8 The ASTERIX Project: Cloudy DB Research at UC Irvine


*Mike Carey (University of California – Irvine, US)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Mike Carey

The ASTERIX project is developing new technologies for ingesting, storing, managing, indexing, querying, analyzing, and subscribing to vast quantities of semi-structured information. The project is combining ideas from three distinct areas - semi-structured data, parallel databases, and data-intensive computing - to create a next-generation, open source software platform that scales by running on large, shared-nothing commodity computing clusters. ASTERIX targets a wide range of semi-structured information, ranging from "data" use cases, where information is well-tagged and highly regular, to "content" use cases, where data is irregular and much of each datum is textual. ASTERIX is taking an open stance on data formats and addressing research issues including highly scalable data storage and indexing, semi-structured query processing on very large clusters, and merging parallel database techniques with today's data-intensive computing techniques to support performant yet declarative solutions to the problem of analyzing semi-structured information. This presentation will provide a whirlwind overview of the project, including its three-layer architecture - the ASTERIX parallel information system (with its ADM data model and AQL query language), the Algebricks query processing layer (which aims to support other implementors of data-intensive computing languages as well), and the Hyracks data-intensive computing platform (an alternative to such platforms as Hadoop and Dryad).

### 3.9 Algebricks + Hyracks: An efficient Data-Centric Virtual Machine for the Cloud

*Vinayak Borkar (University of California – Irvine, US)*





License  Creative Commons BY-NC-ND 3.0 Unported license  
© Vinayak Borkar

In order to harness the power of the cloud for data-intensive tasks, we need a higher level of abstraction that eases the specification of jobs. In this talk we present two systems: Hyracks,

a low-level runtime infrastructure that provides APIs to implement data-parallel operators. In addition, the Hyracks platform includes some commonly useful operators along with a Hadoop compatibility layer to transparently run Hadoop jobs on Hyracks. The second layer, Algebricks, is a higher level of abstraction that provides logical operators which get optimized and compiled down into Hyracks jobs. Algebricks provides a rewriting framework that allows users to implement new rewrite rules.

### 3.10 Extending Map-Reduce for Efficient Predicate-Based Sampling





*Raman Grover (University of California – Irvine, US)*

License     Creative Commons BY-NC-ND 3.0 Unported license  
© Raman Grover

Data analysts today want to grab every bit of data and extract useful information from it. The collected data may scale tera or even petabytes. Sampling has been established as an effective tool in avoiding the subsequent processing cost. A fixed size random sample may not suffice as the sampled data is often required to satisfy additional predicates in order for the collected sample to be useful. We refer this kind of sampling as "Predicate-Based" sampling and is a widely occurring pattern at Facebook. We desire to be able to produce such samples from large scale data in a manner such that the response time is independent of the size of the input dataset. This allows to produce desired samples from increasingly large sizes of input data. Predicate-based sampling can be expressed as a Map-Reduce task. Hadoop as a Map-Reduce implementation provides inefficient execution as it assumes that all input must be processed for a job to produce the required result. Predicate-Based sampling belongs to a class of jobs that can potentially produce the required result by processing partial input. We present an extension of Map-Reduce execution model ( as implemented in Hadoop ) that allows incremental processing wherein input is added dynamical to a running job in accordance with the need and the load on the cluster. The extended model allows us to produce predicate-based samples from increasingly large quantities of data with response time being independent of the size of the input.

### 3.11 Challenges for Cloud Benchmarking


*Enno Folkerts (SAP AG – Walldorf, DE)*

License     Creative Commons BY-NC-ND 3.0 Unported license  
© Enno Folkerts

We develop guidelines for designing and running cloud benchmarks. A cloud benchmark is a benchmark, which makes it possible to compare cloud services of a certain domain. We will not define a benchmark. We will state, what cloud benchmarks may have in common and what differentiates cloud benchmarks from traditional benchmarks. We will also check which traditional benchmarking principles are still valid for the cloud and which principles may have to be altered. We will argue, that it is not sufficient to run well established benchmarks in the cloud, but that the cloud calls for a new generation of benchmarks. We will also see, that there may be different challenges for consumers and providers in the domain of cloud benchmarks.

### 3.12 Trade-Offs in Cloud Application Architecture

*Stephan Tai (KIT – Karlsruhe Institute of Technology, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Stephan Tai

There are diverse objectives in cloud computing – however, not always can all of these objectives be met at the same time. This includes, for example, the traditional question of data consistency versus high availability in distributed data storage. Other potentially conflicting (classes of) objectives include cost efficiency, dependability, performance, or security. We study cloud application architectures from a service-oriented computing perspective and discuss the problem of trade-offs between conflicting objectives. We argue for a novel service engineering model that incorporates trade-offs as first-class abstractions in application architecture design, and call for additional runtime features ("tuning knobs") to flexibly manage trade-offs at runtime.

### 3.13 Benchmarking Large-Scale Parallel Processing Systems


*Alexander Alexandrov (TU Berlin, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Alexander Alexandrov

This presentation captures an overview of recent work done in the areas of cloud benchmarking and parallel data generation. We first present Myriad – a toolkit for massively parallel generation of synthetic datasets. We show how the parallelization approach implemented by the toolkit relies on horizontal partitioning of the generated data sequences and is alleviated by the use of efficient SeedSkip operations on the underlying PRNG streams. In addition, we also explain how the toolkit fits into our general-purpose benchmark for high-level analytics languages running on top of Hadoop or similar parallelization frameworks.

### 3.14 To Cloud or Not To. Musings on Cloud Deployment Viability and Cost Models


*Radu Sion (Stony Brook University, US)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Radu Sion

In this talk we explore the economics of technology outsourcing in general and cloud computing in particular. We identify cost trade-offs and postulate the key principles of outsourcing that define when cloud deployment is appropriate and why.

### 3.15 Building Large XML Stores in the Amazon Cloud

*Jesus Camacho-Rodriguez (INRIA Saclay – Orsay, FR)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Jesus Camacho-Rodriguez

It has been by now widely accepted that an increasing part of the world's interesting data is either shared through the Web or directly produced through and for Web platforms using formats like XML (structured documents). At the same time, cloud storage and computing platforms such as Amazon Web Services (AWS) have gained traction and attracted interest for their elastic scalability. In particular, AWS provides a set of basic sub-systems (such as storage for bulk, respectively, small-grained data, queue systems etc.) on top of which one can build more complex applications.

We present our ongoing work on designing an architecture and associated algorithms for efficiently managing large corpora of XML documents based on the AWS components. We consider different indexing strategies to use in order to facilitate the access to a collection of XML documents stored within AWS and efficiently support query processing on these documents. Work is ongoing, in particular on enabling our indexing algorithms to scale through the boundaries of AWS structures, and to experimentally evaluate the trade-offs brought by each strategy.

### 3.16 Storage for End-user Programming


*Dirk Riehle (Universität Erlangen-Nürnberg, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Dirk Riehle

This talk illustrates our vision for end-user programming taking a wiki-style approach. We show some of the challenges that arise for a backing database.

### 3.17 Adaptive Query Processing in Stratosphere


*Johann-Christoph Freytag (HU Berlin, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Johann-Christoph Freytag

The talks presents the first results on adaptive query processing in Stratosphere. Our approach is based on the SCORE operator, and extension of Hellerstein's Eddy operator, and on a competition model which is motivated by the work of G. Antoshekov (1992). We show that the two approaches together improve the overall response time when considering join processing.

### 3.18 Nephele: (Cost) Efficient Parallel Data Flows in the Cloud


*Daniel Warneke (TU Berlin, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Daniel Warneke

The world of parallel computing faces data sets which are increasing rapidly in complexity and size. While many different higher-level programming abstractions have recently been introduced to facilitate domain-specific application development on these data sets, the underlying execution engines are still heavily tailored towards cluster-centric long-running batch jobs. This talk highlights the different directions for future research in the field of data-intensive execution engines and sketches our ongoing efforts in the scope of the Stratosphere project.

### 3.19 Information Extraction in Stratosphere

*Astrid Rheinlaender (HU Berlin, DE)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Astrid Rheinlaender

Large scale analytical text processing is important for many real-world scenarios. In drug development, for instance, it is extremely helpful to gather as much information as possible on the drug itself and on other, structurally similar drugs. Such information is contained in various large text collections like patent or scientific publication databases. As a part of the StratoSphere project, we therefore investigate query-based analysis of large quantities of unstructured text. Such a query is parsed, optimized, parallelized, and executed on a cloud infrastructure. Our extraction operators are configurable to embrace different IE strategies, either geared towards high throughput, high precision, or high recall. On the other hand, we also develop optimization strategies such as rewrite rules or cost estimates that allow an efficient execution of IE queries.

### 3.20 Cloud Computing and Next Generation Sequencing

*Ulf Leser (HU Berlin, DE)*


License  Creative Commons BY-NC-ND 3.0 Unported license  
© Ulf Leser

The Life Sciences, and in particular the recent advances in DNA sequencing (Next Generation Sequencing, NGS) create an ever growing amount of data. Interestingly, the rate at which data production is increasing is much higher than Moore's law predicts for the increase in computational power - while sequencing throughput doubles roughly every 6-9 months, CPU power is doubling only every 18 months. This poses considerable challenges to the analysis of sequencing data sets. The talk explains the problem, presents the state-of-the-art, and discusses the opportunities Cloud Computing might offer for sequence analysis and well as the problems that have to be tackled.



### 3.21 Cost-aware data management in the cloud


*Verena Kantere (Cyprus University of Technology – Lemesos, CY)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Verena Kantere

The success of offering data services in the cloud is achieving to perform both cost-efficient and traditionally time-efficient data management. We have proposed a novel economy model for a cloud provider, where users pay on-the-go for the data services they receive and user payments can be used for service provision, infrastructure operation and profit. The economy employs a cost model that takes into account all the available resources in a cloud, such as disk space and I/O operations, CPU time and network bandwidth. In order to ensure the economic viability of the cloud, the cost of offering new services has to be amortized to prospective users that will use them. We have proposed a novel cost amortization model that predicts the extent of amortization in time and number of users. The economy is completed with a dynamic pricing scheme that achieves optimal cloud profit while ensuring user satisfaction with service prices. We envision a cloud data service provider with three conceptual layers that should interact closely; namely, the cloud DBMS, the service and the economy layer. There are many open research issues on all the layers. Coarsely, it is necessary to provide techniques for offering and pricing groups and workflows of services that are customizable for various user needs and cloud environments taking into account risk factors.

### 3.22 Building high performance indexes for key value storage


*Russell Sears (Yahoo! Research – Santa Clara, US)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Russell Sears

This talk provides an overview of Yahoo!’s distributed key-value store, PNUTS. We are in the process of implementing a new log structured index for PNUTS, and discuss its implementation, and a number of issues that arise when benchmarking of log structured storage systems.

### 3.23 MuTeDB - A dbms that shows quiet on multi-tenancy

*Bernhard Mitschang (Universität Stuttgart, DE)*


License  Creative Commons BY-NC-ND 3.0 Unported license  
© Bernhard Mitschang

Software as a Service (SaaS) facilitates acquiring a huge number of small tenants by providing low service fees. To achieve low service fees, it is essential to reduce costs per tenant. For this, consolidating multiple tenants onto a single relational schema instance turned out beneficial because of low overheads per tenant and scalable manageability. We contribute first features of an extended RDBMS to support tenant-aware data management natively. We introduce tenants as first-class database objects and propose the concept of a tenant context to isolate

a tenant from other tenants. We present a schema inheritance concept that allows sharing a core application schema among tenants while enabling schema extensions per tenant.

### 3.24 Yes, but does it work?


*Steve Loughran (HP Lab – Bristol, GB)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Steve Loughran

Coverage of the testing issues related to Cloud infrastructures and how applications deployed in such a world don't work the way they should, because they contain assumptions about their environment that are no longer valid.

### 3.25 ScalOps: Cloud Computing in a High-Level Programming Language

*Tyson Condie (Yahoo! Inc., US)*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Tyson Condie

Machine learning systems take part in billions of page views every day at Yahoo!. Examples include recommended reading on Yahoo! News, personalized advertisements and, possibly most well known, the Yahoo! Mail spam filter and the personalized assembly of Yahoo! Frontpage. Building the underlying models includes several distinct phases, currently accomplished using different tools: 1. Feature Extraction / Data preparation: A ETL-style feature extraction and data joining phase that is typically accomplished by Apache Pig. 2. Modeling: Yahoo! uses any number of different machine learning techniques and algorithms to model the data. They all share one key characteristic that makes them unsuitable for DAG-based systems such as Hadoop or Dryad: They perform multiple passes over the data, changing state along the way. It has been demonstrated numerous times in the machine learning community that speedups of at least 10x can be achieved by custom MPI-style implementations when compared to Hadoop MapReduce. 3. Evaluation: This, again, typically performs ETL-style computations and can be accomplished using the large scale data processing tools widely available today.

Scalops is a new machine learning toolkit currently under development. It provides an API and a runtime that can natively express and execute iterations and recursion over Big Data. This in turn allows us to unify all three steps outlined above in a single, concise and easily approachable programming interface in the form of an internal domain specific language hosted by the Scala programming language. We expect the latter to be of great benefit to machine learning practitioners at Yahoo! and beyond. We also envision unifying several now disparate computational paradigms under a single runtime.

### 3.26 Cloud-based Web data management (it's all about how you view it)

*Ioana Manolescu (Université Paris Sud – Orsay, FR)*

**Joint work of** Dario Colazzo, Francois Goasdoue, Jesus Camacho-Rodriguez, Andres Aranda Andujar, and Zoi Kaoudi

**License** © © ⊖ Creative Commons BY-NC-ND 3.0 Unported license  
© Ioana Manolescu

The development of the Web led to a strong increase in the volumes of Web-style data being produced, exchanged, analyzed and consumed daily; thus, it is estimated that we now produce every two days the same amount of data that was produced from the beginning of humanity until 2003 . Moreover, most of this continuously produced data does not reside in databases but in Web content such as Web pages, social networking sites, blogs, user videos etc. This wealth of data leads to great interest in efficiently and reliably storing, querying, analyzing and transforming such data. By "Web data", we designate document data, in the style of Web pages, and which we views as XML documents, as well as Semantic Web style data, represented by RDF triples, possibly endowed with RDF Schemas.

In this context, cloud platforms provide a distributed framework, providing at least some lower (file-) level storage of the data. Typical cloud infrastructure provide several levels of storage, one dedicated to very large (unstructured) data objects, and another one built for storing numerous small items, typically structured as sets of attribute-value pairs. Also within the context of cloud computing, frameworks and programming languages have emerged, typically with an emphasis on parallel processing, distributing parallel computations and gathering back results, along the lines of “and typically generalizing or extending” the Map/Reduce paradigm. The starting point of our research agenda in this context is the observation that the complex, heterogeneous or missing structure of Web data raises many challenges for large-scale efficient data management platforms. Indeed, in a large distributed setting, it is not clear how one user’ or application’s data should be organized on the available storage layers, in order to support efficiently the required query/transactions mix. The complex shape of Web data formats is typically not a good format for storing the data, thus various segmentations or fragmentation strategies are often applied to re-organize the content in smaller, more manageable fragments. In turn, these fragments may be replicated on several sites and adaptively placed across the distributed storage units. When available, schema information as well as information about the workload of each user can also be used to this purpose. The design of such indexes and views should be made with parallelism in mind, so that no single point of contention is introduced when searching for the data structures suited for a given query.


We plan to investigate the design, algorithmic and performance properties of efficient storage structures in the context of cloud-based data management, in particular distributed indexes and distributed materialized views. This work is to be performed in particular within the ICT Labs Europa activity.

## 4 Overview of Demos

During the demo session three system demos were given. Each demo is shortly described in the remainder of this section.

## 4.1 Asterix

*Vinayak Borkar, Raman Grover*


**URL** <http://asterix.ics.uci.edu>  
**License**  Creative Commons BY-NC-ND 3.0 Unported license  
© Vinayak Borkar, Raman Grover

The Asterix system is developed at UC Irvine, UC Riverside, and UC San Diego. It is a platform to execute queries on semi-structured data in a massively parallel fashion. The basic system consists of three components, an execution engine called Hyracks, an algebraic optimization layer named Algebrix, and the Asterix query language (AQL). The Hyracks engine is published as open source. The demo showed how data schemas and analytical (OLAP-style) queries are specified by AQL, how they are optimized and executed on Hyracks.

A second demo showed a use case that computes a geo-spatial frequency aggregation of twitter feeds which contain a certain keyword. The result was visualized as heat-map using a web-based map service.

## 4.2 Stratosphere

*Daniel Warneke, Fabian Hueske*

**URL** <http://www.stratosphere.eu>  
**License**  Creative Commons BY-NC-ND 3.0 Unported license  
© Daniel Warneke, Fabian Hueske

Stratosphere is a joint research project by TU Berlin, HU Berlin, and HPI Potsdam. The project researches data management in the cloud and builds a prototype that is publicly available as open source. The Stratosphere system consists of the parallel PACT programming model, an database-inspired optimizer, and a flexible execution engine called Nephele. PACT is a generalization of the MapReduce programming model. Nephele can request computing nodes on demand from Infrastructure-as-a-service (IaaS) providers. The demo showed how an analytical query is defined a PACT program, how it is optimized, and how it is executed on the Amazon EC2 IaaS environment.

A second use case demonstrated a biomedical information extraction pipeline that was defined as a PACT program.

## 4.3 Parallel Data Generation with Myriad

*Alexander Alexandrov*

**URL** <http://www.myriad-toolkit.com>  
**License**  Creative Commons BY-NC-ND 3.0 Unported license  
© Alexander Alexandrov

Myriad is a development framework for parallel data generators. Myriad relies on an efficient skip-ahead pseudo-random number generator (PRNG) sequence to create virtual pseudo-random sequences for user-defined data types that can be partitioned and randomly accessed at constant computational cost. Myriad-based generators can therefore generate skewed, correlated, and referencing data without any communication between generator instances

running in parallel. This feature makes Myriad a viable framework to define workloads and benchmarks for massively parallel systems such as Hadoop, Asterix, or Stratosphere.


The demo session showed how Myriad can be extended to generate graphically structured data in parallel. The statistical constraints implemented by the data generator make the produced datasets a good fit for testing certain types of analytical queries in a large-scale environment.

## 5 Break-Out Group Reports

This section lists the abstracts of the break-out sessions. The abstracts and figures were taken from the seminar's material web site.

### 5.1 Cloud Benchmarking

*Alexandru Iosup, Alexander Alexandrov, Enno Folkerts, Donald Kossmann, Seif Haridi, Volker Markl, Tim Kraska, Radu Sion, Anastasia Ailamaki, Dean Jacobs*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Alexandru Iosup, Alexander Alexandrov, Enno Folkerts, Donald Kossmann, Seif Haridi, Volker Markl, Tim Kraska, Radu Sion, Anastasia Ailamaki, Dean Jacobs


The goal of this breakout session was to begin work on providing a procedure for rating cloud Infrastructures and Platforms. An important target was to consider ways through which clouds could receive ratings that IT consumers, especially small companies, can use to guide their IT provisioning processes. The need for new benchmarks and benchmarking practices derives from the need of these IT consumers to understand the performance-, the availability-, the reliability-, the scalability-, the elasticity-related, etc. characteristics of clouds. Without a standard benchmarking suite, cloud operators are unable to demonstrate their claims; conversely, potential buyers are not persuaded to buy.

Our group has focused on three main tasks:

1. Defining a framework for the process of benchmarking, which can guide the creation, use, and reporting based on a suite (family) of benchmarks.
2. Understanding the main cloud characteristics that may require new approaches to benchmarking. For example, due to the performance variability exhibited by many clouds, benchmarking metrics have to focus on both expectation and variability. Similarly, elasticity, which encompasses the behavior of the system under varying load, needs possibly new metrics. Other notions discussed were: scalability (including the time needed to reach the desired scale), reliability, availability, robustness (against a "TNT" test), information availability (knowing partially the status of the system), the data management lifecycle (including backups under load and archival), the ability to benchmark data consistency, etc.
3. Asking the questions that can guide the creation of new cloud-related benchmarks. What are good Key Performance Indicators and how to build a Single-Value Rating? Should we test under (external) load? Should we use test drivers located in the cloud? How to benchmark for different provisioning and allocation models/policies? How to benchmark for interactive workloads? (the distance to customer is now part of cloud location) How to build scalable benchmarks and tools for them? etc.
4. Creating a plan for continuation. We have agreed on a plan for continuation.

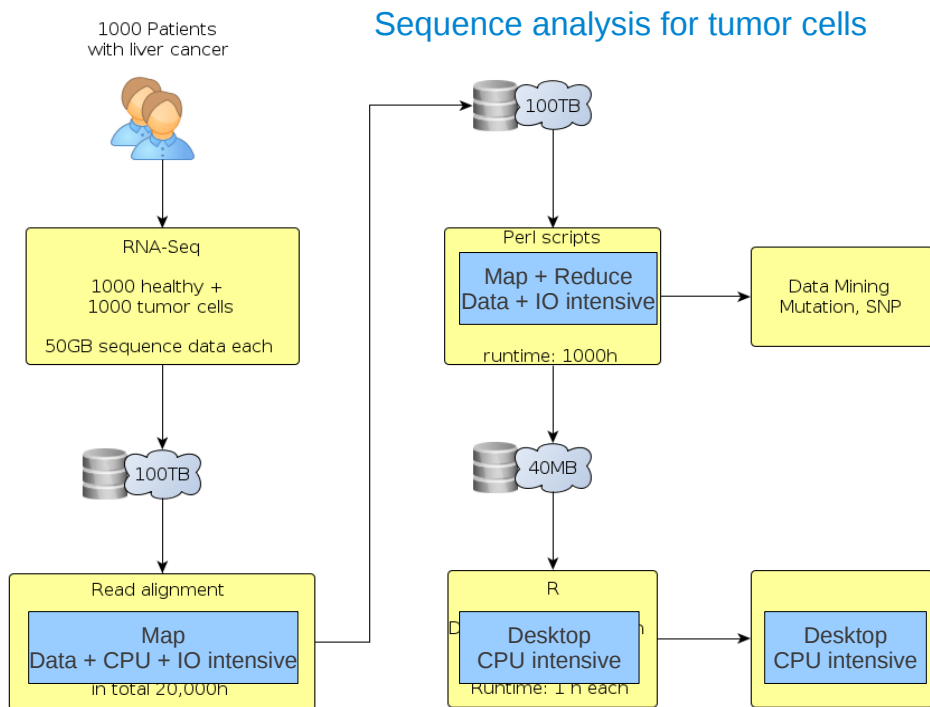
## 5.2 Biomedical Analytics in the Cloud

*Jim Dowling, Johann-Christoph Freytag, Keijo Heljanko, Ulf Leser, Felix Naumann, Astrid Rheinländer*

License  Creative Commons BY-NC-ND 3.0 Unported license  
 © Jim Dowling, Johann-Christoph Freytag, Keijo Heljanko, Ulf Leser, Felix Naumann, Astrid Rheinländer

Recent improvements in both the cost and throughput of sequencing machines has caused a mismatch between the increasing rate at which they can generate data and the ability of our existing tools and computational infrastructure to both store and analyse this data. Currently, organizations are investing significant amounts of resources in sequencing machines before they either have the necessary storage infrastructure or analysis tools that can archive and process the resultant data.

The goal of this breakout-group is to propose both a cloud-computing infrastructure and parallel-programming support that will enable the secure storage and parallel analysis of the coming flood of sequence data. We anticipate that an infrastructure of only 100 machines should cost at most the same as an existing sequencing machine and, with the help of recent cloud computing technologies, it should have minimal administration costs. Such an infrastructure will enable organizations to support the long-term archival of sequence data and reduce the time required to process sequence data by a factor of around one hundred. A sample workflow on top of our parallel infrastructure is depicted on Figure 1.



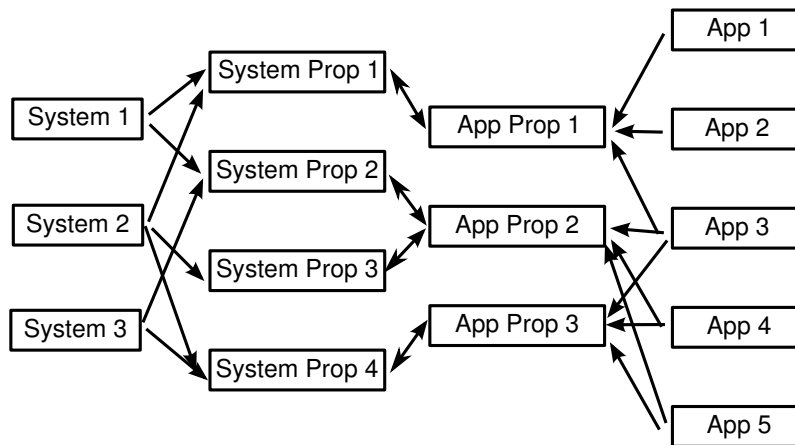
■ **Figure 1** Sequence analysis workflow

### 5.3 Data and Programming Models

*Vinayak Borkar, Jesus Camacho-Rodriguez, Mike Carey, Tyson Condie, Raman Grover, Arvid Heise, Fabian Hueske, Dean Jacobs, Steve Loughran, Ioana Manolescu-Goujot, Sergey Melnik, Bernhard Mitschang, Daniel Warneke*

License © ⓘ ⊖ Creative Commons BY-NC-ND 3.0 Unported license  
 © Vinayak Borkar, Jesus Camacho-Rodriguez, Mike Carey, Tyson Condie, Raman Grover, Arvid Heise, Fabian Hueske, Dean Jacobs, Steve Loughran, Ioana Manolescu-Goujot, Sergey Melnik, Bernhard Mitschang, Daniel Warneke

The Data and Programming Model breakout session tried to come up with characterizations of large-scale data applications and platforms. These characterizations should be used to describe and specify the requirements of applications and features of platforms in order to find matches between both. Figure 2 shows how to derive application platform matches based on their characteristics. After a set of characteristics had been derived, they were applied to a couple of example applications and platforms. In addition a 'map' of software stacks of selected parallel data processing platforms was created.



■ **Figure 2** Matching of Application and Platform Characteristics

#### Abstract

The era of the mainframe and the cluster may seem over, but their concepts are being applied to large-scale datacentres, offering massively-parallel, data-intensive computing and storage services. The challenge in this world is what algorithms can scale up to this environment, tolerate the frequent failures, and support the complex analysis and computational needs of the latest generation of applications.

The goal of this breakout session is to characterise the algorithms and the programming models that have been built to work in this environment. For some popular problems, we show their characteristics, and therefore how their needs match the feature set of these programming models. The characteristics show gaps in the feature set of today’s technologies; features that future systems could address.

#### Application Characteristics

1. Application types: **A**nalyze vs. **T**ransform vs. **E**xtract

■ **Table 1** Application Characteristics

<b>System</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<i>WordCount</i>	A	B	M	D	S	
<i>PageRank</i>	A	B	M	D	I	
<i>TPC-H</i>	A	O	S+M	D	S	
<i>K-Means</i>	A	B	M	D	I	
<i>Tile Rendering</i>	T	B	M	C	S	
<i>ETL</i>	T	B	M	D	S	
<i>Recommendation (SGD/LDA)</i>	A	B	M	C	I	
<i>TrendAnalysis (Twitter)</i>	A	B	S	D+C	S	

2. Operation response mode: **Online** (sync) vs. **Batch** (async)
3. Request types: **Selective** vs. **Massive**
4. Operation step types: **Data Intensive** vs. **Compute Intensive**
5. Operation processing mode: **Single flow** vs. **Iterative / recursive**
6. Data access modes: **Get** vs. **Filter** vs. **Query**

### Platform Characteristics

1. Application types: **Analyze** vs. **Transform** vs. **Extract**
2. Data types: **Static Typed** vs. **Dynamic Typed** vs. **Untyped** (to be updated in Table 2)
3. Operation response mode: **Online** (sync) vs. **Batch** (async)
4. Operation semantics: **Transparent** vs. **Opaque**
5. Request types: **Selective** vs. **Massive**
6. Operation step types: **Data Intensive** vs. **Compute Intensive**
7. Operation processing mode: **Single flow** vs. **Iterative / recursive**
8. Data access modes: **Get** vs. **Filter** vs. **Query** (to be updated in table 2)

### Characterization of Selected Applications

List of potential example applications:

- Genome Alignment
- Data Cleansing
- Enterprise OLAP
- E-Health Record Management
- Twitter Analysis
- (Ad) Recommendation Systems
- Indexing
- Auditing / Sensor Networks
- (Realtime) Log & Click Analysis
- Tile Rendering
- PageRank
- Social Network Analysis
- Shortest Path

Table 1 shows the characterization of selected applications.



■ **Table 2** System Characteristics

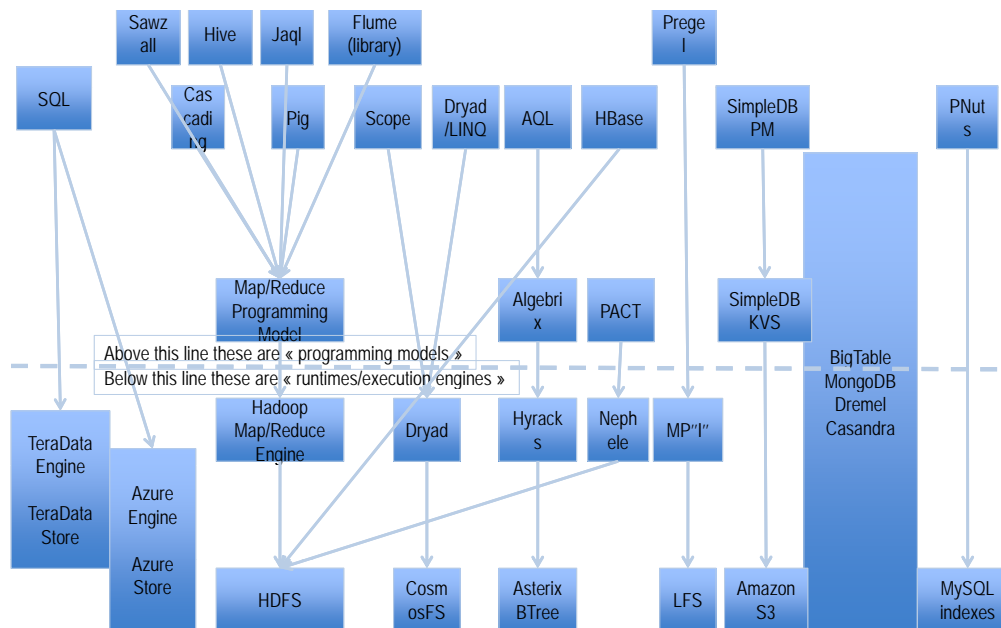
System	1	2	3	4	5	6	7	8
<i>Pig</i>	T	U	B	O <sup>-</sup>	S+M	D+C	S	
<i>Hive</i>	A	T	B	T	S+M	D	S	
<i>AQL</i>	A+E	T	O+B	T	S+M	D	I	
<i>PACT</i>	A+T	U	B	O <sup>-</sup>	M	D+C	S	
<i>MR PM</i>	A+T	U	B	O	M	D+C	S	
<i>SQL</i>	A+E	T	O	T	S	D	I <sup>-</sup>	
<i>Pregel</i>	A	T <sup>-</sup>	B	O	M	D+C	I	
<i>Nephele</i>	A+T	U	B	O	M	D+C	S	
<i>MPI</i>	-	U	B	O	M	C	I	
<i>Dremel</i>	A	T	O	T	S	D	S	
<i>SimpleDB</i>	E	T	O	T	S+M	D	S	
<i>HBase</i>	E	U	O	T <sup>-</sup>	S	D	S	

**Characterization of Existing Platforms**

Table 2 shows the characterization of selected platforms.

**Selected Parallel Data Processing Stacks**


Figure 3 shows the processing stacks of selected parallel data processing platforms.



■ **Figure 3** Parallel Data Processing Stacks

## 5.4 Transactions in the Cloud


*Anastisia Ailamaki, Seif Haridi, Alfons Kemper, Tim Kraska, Simon Loesing, Sergey Melnik, Russell Sears*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© A. Ailamaki, S. Haridi, A. Kemper, T. Kraska, S. Loesing, S. Melnik, R. Sears

The increasing scale of data management and high-availability requirements have led large Internet services to deploy scalable storage systems that span many data-centers. Such systems relax transactional semantics, such as atomicity and consistency, for scalability. Based on experiences with these systems, we want to explore the fundamental trade-offs these systems face. This includes defining the design requirements that systems of this scale have to fulfill. Some of these requirements are universal, such as manageability and fault-tolerance, while others vary with the application. In particular, these application-dependent differences lead to different data models, consistency properties, data placement and programming models which directly impact the approaches applications can use to transactionally modify the data.

## 5.5 Cloud Economics

*Verena Kantere, Magdalena Balazinska, Athanasios Papaioannou, Helmut Krcmar, Miron Livny*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Verena Kantere, Magdalena Balazinska, Athanasios Papaioannou, Helmut Krcmar, Miron Livny

Cloud-computing has recently emerged as a new paradigm for delivering compute infrastructures and software in an "elastic" (i.e. flexible, scalable, and pay-as-you-go) manner. While the service elasticity offers significant advantages, such as reducing the time-to-market and allowing adaptive capacity planning, it also creates important challenges for the platform and software design. For instance, software must effectively take advantage of the possibility to grow and shrink resources as needed. In this work, we study the case of data-management-as-a-service. Today, cloud providers offer data management solutions but they are either feature limited (e.g., Amazon SimpleDB) or lack scalability (e.g., SQL Azure). We identify key design challenges (e.g. system adaptivity to agile resource planning, setup and data migration costs, etc.) and sketch possible architectures.

## 5.6 Cloud Storage

*Anastisia Ailamaki, Russell Sears*

License  Creative Commons BY-NC-ND 3.0 Unported license  
© Anastisia Ailamaki, Russell Sears

Gaps in solid state disk, network and magnetic disk performance are growing exponentially. In the long term, solid state storage performance will outpace networking, while networking will outpace magnetic media. Given these trends, and the need for both magnetic and solid state media, it is unclear whether it will continue to be possible to build general purpose clouds in the future, or how many types of specialized clouds will make sense.

It is also unclear whether the underlying hardware architecture should be homogeneous, so that each machine contains multiple types of storage devices, or heterogeneous, with many classes of machines provisioned for distinct workloads. In a homogeneous system, interference from different types of applications may severely impact long-tail latencies and overall throughput. However, heterogeneous designs statically partition applications into silos, preventing capacity sharing. Also, different classes of applications lead to different bottlenecks; the heterogeneous approach amplifies these bottlenecks.

We intend to benchmark a number of configurations and systems based on both approaches, and to see which of the above problems are most serious. We will use these results to inform the design of new cloud-based storage hardware and software stacks.

## Participants

- Alexander Alexandrov  
TU Berlin, DE
- Alexandru Iosup  
TU Delft, DE
- Alfons Kemper  
TU Munich, DE
- Anastassia Ailamaki, EPFL  
Lausanne, CH
- Arvid Heise  
Hasso Plattner Institute  
Potsdam, DE
- Astrid Rheinlaender  
HU Berlin, DE
- Athanasios Papaioannou  
EPFL Lausanne, CH
- Bernhard Mitschang  
University Stuttgart, DE
- Daniel Warneke  
TU Berlin, DE
- Dean Jacobs  
SAP AG, Walldorf, DE
- Dirk Riehle  
Univ. Erlangen-Nuernberg, DE
- Donald Kossmann, ETH  
Zurich, CH
- Enno Folkerts  
SAP AG, Walldorf, DE
- Fabian Hueske  
TU Berlin, DE
- Felix Naumann  
Hasso Plattner Institute  
Potsdam, DE
- Helmut Krcmar  
TU Munich, DE
- Ioana Manolescu-Goujot  
Universite Paris Sud-Orsay, FR
- Jesus Camacho-Rodriguez  
INRIA Saclay-Orsay, FR
- Jim Dowling  
Swedish Institute of Computer  
Science, Kista, SE
- Johann-Christoph Freytag  
HU Berlin, DE
- Keijo Heljanko  
Helsinki Univ. of Technology, FI
- Magdalena Balazinska  
University of Washington,  
Seattle, US
- Mike Carey,  
UC Irvine, US
- Miron Livny  
Univ. of Wisconsin-Madison, US
- Radu Sion  
Stony Brook University, US
- Raman Grover  
UC Irvine, US
- Russell Sears  
Yahoo! Research, US
- Seif Haridi  
Swedish Institute of Computer  
Science, Kista, SE
- Sergey Melnik  
Google, US
- Simon Loesing  
ETH Zürich, CH
- Stefan Tai  
Karlsruhe Institute of  
Technology, DE
- Steve Loughran, HP Labs  
Bristol, UK
- Tim Kraska  
UC Berkeley, US
- Tyson Condie  
Yahoo! Inc., US
- Ulf Leser  
HU Berlin, DE
- Verena Kantere  
Cyprus University of Technology,  
Lemesos, CY
- Vinayak Borkar  
UC Irvine, US
- Volker Markl, TU Berlin, DE

