# Overview

## DB-Types

<http://www.enterprisestorageforum.com/storage-management/the-top-of-the-big-data-stack-database-applications.html>



* Wide Column Store/Column Families
* Document Store
* Key Value/Tuple Store
* Graph Databases
* Multimodel Databases
* Object Databases
* Multivalue databases
* RDF databases

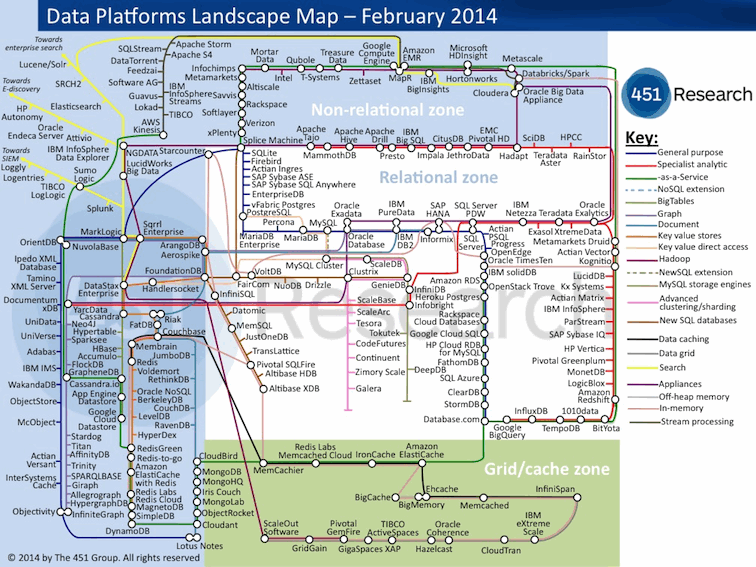
### noSQL Databases

<http://en.wikipedia.org/wiki/NoSQL#Examples>

### Database Map (451 Group)

<http://blogs.the451group.com/information_management/2014/03/18/updated-data-platforms-landscape-map-february-2014/>

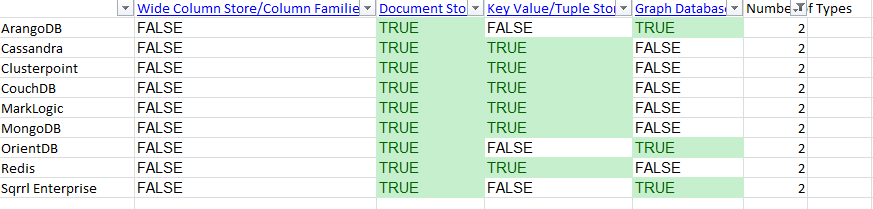
<https://blogs.the451group.com/information_management/files/2014/03/data_map.gif>



### Database Map (cbu)

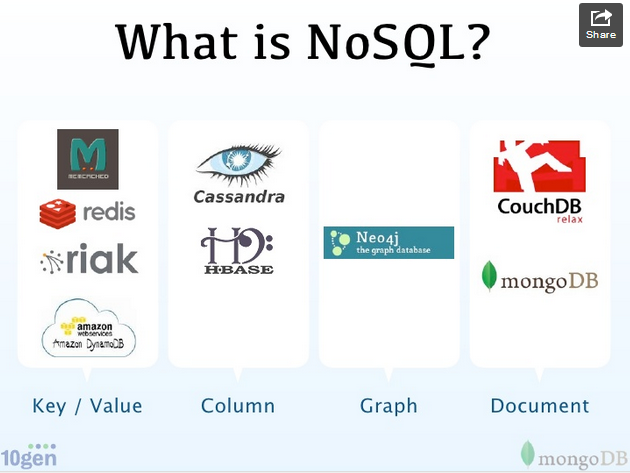
DB-Overview.xlsx

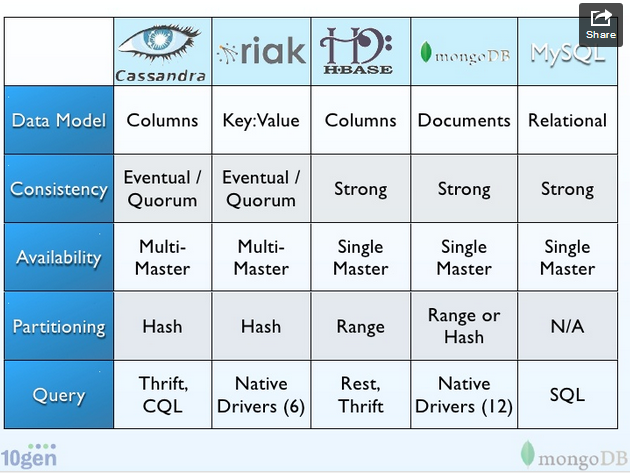
DBs, die mehr als 1 Typ implementieren:



### Selected DBs

<http://de.slideshare.net/spf13/nosql-databases-and-managing-big-data>



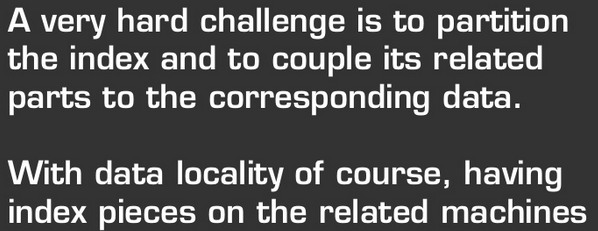


## Slides Pavlo Baron

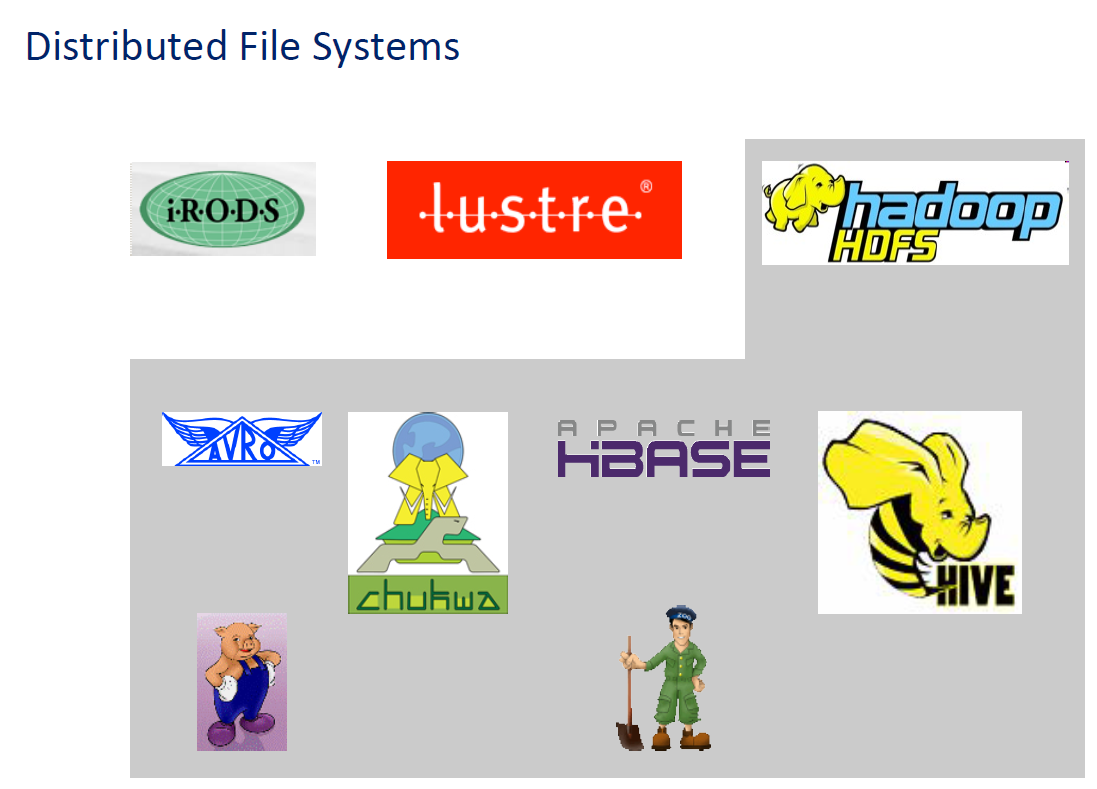
<http://de.slideshare.net/pavlobaron/big-data-nosql-efs11-pavlo-baron>

<http://de.slideshare.net/pavlobaron/a-tech-guys-take-on-big-data-business-cases-pavlobaron>

<http://de.slideshare.net/pavlobaron/the-big-data-developer-pavlobaron>



## Distributed File Systems



### Hadoop (HDFS)

<http://en.wikipedia.org/wiki/Hadoop>

Beyond HDFS, YARN and MapReduce, the entire Apache Hadoop "platform" is now commonly considered to consist of a number of related projects as well – [Apache Pig](http://en.wikipedia.org/wiki/Pig_(programming_tool)), [Apache Hive](http://en.wikipedia.org/wiki/Apache_Hive), [Apache HBase](http://en.wikipedia.org/wiki/Apache_HBase), [Apache Spark](http://en.wikipedia.org/wiki/Apache_Spark), and others.[[3]](http://en.wikipedia.org/wiki/Hadoop_distributed_file_system#cite_note-3)

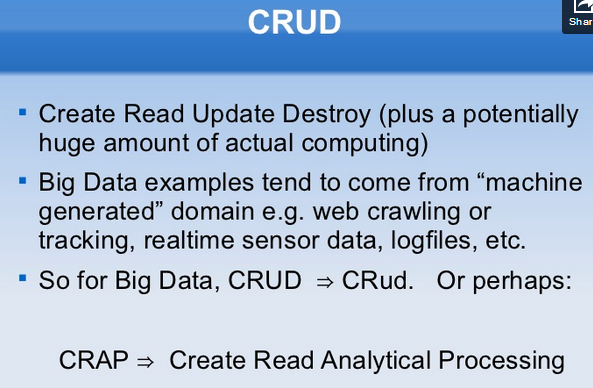
**Hadoop on Amazon EC2/S3 services[[edit](http://en.wikipedia.org/w/index.php?title=Apache_Hadoop&action=edit&section=15" \o "Edit section: Hadoop on Amazon EC2/S3 services)]**

It is possible to run Hadoop on [Amazon Elastic Compute Cloud](http://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud) (EC2) and [Amazon Simple Storage Service](http://en.wikipedia.org/wiki/Amazon_Simple_Storage_Service) (S3).[[36]](http://en.wikipedia.org/wiki/Hadoop#cite_note-36) As an example [The New York Times](http://en.wikipedia.org/wiki/The_New_York_Times) used 100 Amazon EC2 instances and a Hadoop application to process 4 TB of raw image [TIFF](http://en.wikipedia.org/wiki/TIFF) data (stored in S3) into 11 million finished [PDFs](http://en.wikipedia.org/wiki/PDF) in the space of 24 hours at a computation cost of about $240 (not including bandwidth).[[37]](http://en.wikipedia.org/wiki/Hadoop#cite_note-37)

## Concepts

http://de.slideshare.net/hcr02/big-data-overview-13366836

CRUD 🡪 CRAP



# Details

## MongoDB + Hadoop

<http://de.slideshare.net/spf13/mongodb-and-hadoop>

## Map-Reduce

<http://en.wikipedia.org/wiki/MapReduce>

For processes that complete fast, and where the data fits into main memory of a single machine or a small cluster, using a MapReduce framework usually is not effective: since these frameworks are designed to recover from the loss of whole nodes during the computation, they write interim results to distributed storage. This crash recovery is expensive, and only pays off when the computation involves many computers and a long runtime of the computation - a task that completes in seconds can just be restarted in the case of an error; and the likelihood of at least one machine failing grows quickly with the cluster size. On such problems, implementations keeping all data in memory and simply restarting a computation on node failures, or - when the data is small enough - non-distributed solutions will often be faster than a MapReduce system.

* Can a Map-Reduce Framework be configured with “crash recovery off”?

## On-Top-Of Hadoop

### PIG

<http://en.wikipedia.org/wiki/Pig_(programming_tool)>

**Pig** [[1]](http://en.wikipedia.org/wiki/Pig_(programming_tool)#cite_note-mainpage-1) is a high-level platform for creating [MapReduce](http://en.wikipedia.org/wiki/MapReduce" \o "MapReduce) programs used with [Hadoop](http://en.wikipedia.org/wiki/Hadoop).

### Hive

<http://en.wikipedia.org/wiki/Apache_Hive>

**Apache Hive** is a [data warehouse](http://en.wikipedia.org/wiki/Data_warehouse) infrastructure built on top of [Hadoop](http://en.wikipedia.org/wiki/Hadoop) for providing data summarization, query, and analysis.[[1]](http://en.wikipedia.org/wiki/Apache_Hive#cite_note-1) While initially developed by [Facebook](http://en.wikipedia.org/wiki/Facebook), Apache Hive is now used and developed by other companies such as [Netflix](http://en.wikipedia.org/wiki/Netflix).[[2]](http://en.wikipedia.org/wiki/Apache_Hive#cite_note-2)[[3]](http://en.wikipedia.org/wiki/Apache_Hive#cite_note-3) Amazon maintains a software fork of Apache Hive that is included in *Amazon Elastic MapReduce* on [Amazon Web Services](http://en.wikipedia.org/wiki/Amazon_Web_Services).[[4]](http://en.wikipedia.org/wiki/Apache_Hive#cite_note-4)

### Spark

<http://en.wikipedia.org/wiki/Apache_Spark>

**Apache Spark** is an open-source data analytics cluster computing framework originally developed in the AMPLab at [UC Berkeley](http://en.wikipedia.org/wiki/UC_Berkeley).[[1]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-wired-1) Spark fits into the Hadoop open-source community, building on top of the [Hadoop Distributed File System](http://en.wikipedia.org/wiki/Apache_Hadoop#Hadoop_distributed_file_system) (HDFS).[[2]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-2) However, Spark is not tied to the two-stage [MapReduce](http://en.wikipedia.org/wiki/MapReduce" \o "MapReduce) paradigm, and promises performance up to 100 times faster than Hadoop MapReduce, for certain applications.[[3]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-3) Spark provides primitives for in-memory cluster computing that allows user programs to load data into a cluster's memory and query it repeatedly, making it well suited to machine learning algorithms.[[4]](http://en.wikipedia.org/wiki/Apache_Spark#cite_note-4)

### HBase

<http://en.wikipedia.org/wiki/Apache_HBase>

**HBase** is an [open source](http://en.wikipedia.org/wiki/Open_source), [non-relational](http://en.wikipedia.org/wiki/Non-relational_database), [distributed database](http://en.wikipedia.org/wiki/Distributed_database) modeled after [Google's](http://en.wikipedia.org/wiki/Google) [BigTable](http://en.wikipedia.org/wiki/BigTable" \o "BigTable) and written in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)). It is developed as part of [Apache Software Foundation](http://en.wikipedia.org/wiki/Apache_Software_Foundation)'s [Apache Hadoop](http://en.wikipedia.org/wiki/Hadoop) project and runs on top of[HDFS (Hadoop Distributed Filesystem)](http://en.wikipedia.org/wiki/Hadoop_Distributed_Filesystem), providing BigTable-like capabilities for Hadoop. That is, it provides a [fault-tolerant](http://en.wikipedia.org/wiki/Fault-tolerant) way of storing large quantities of [sparse](http://en.wikipedia.org/wiki/Sparse_file) data (small amounts of information caught within a large collection of empty or unimportant data, such as finding the 50 largest items in a group of 2 billion records, or finding the non-zero items representing less than 0.1% of a huge collection).

# Cassandra

<http://en.wikipedia.org/wiki/Apache_Cassandra>

**Apache Cassandra** is an [open source](http://en.wikipedia.org/wiki/Open_source_software) [distributed](http://en.wikipedia.org/wiki/Distributed_database) [database management system](http://en.wikipedia.org/wiki/Database_management_system) designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple datacenters,[[1]](http://en.wikipedia.org/wiki/Apache_Cassandra#cite_note-1) with asynchronous masterless replication allowing low latency operations for all clients.

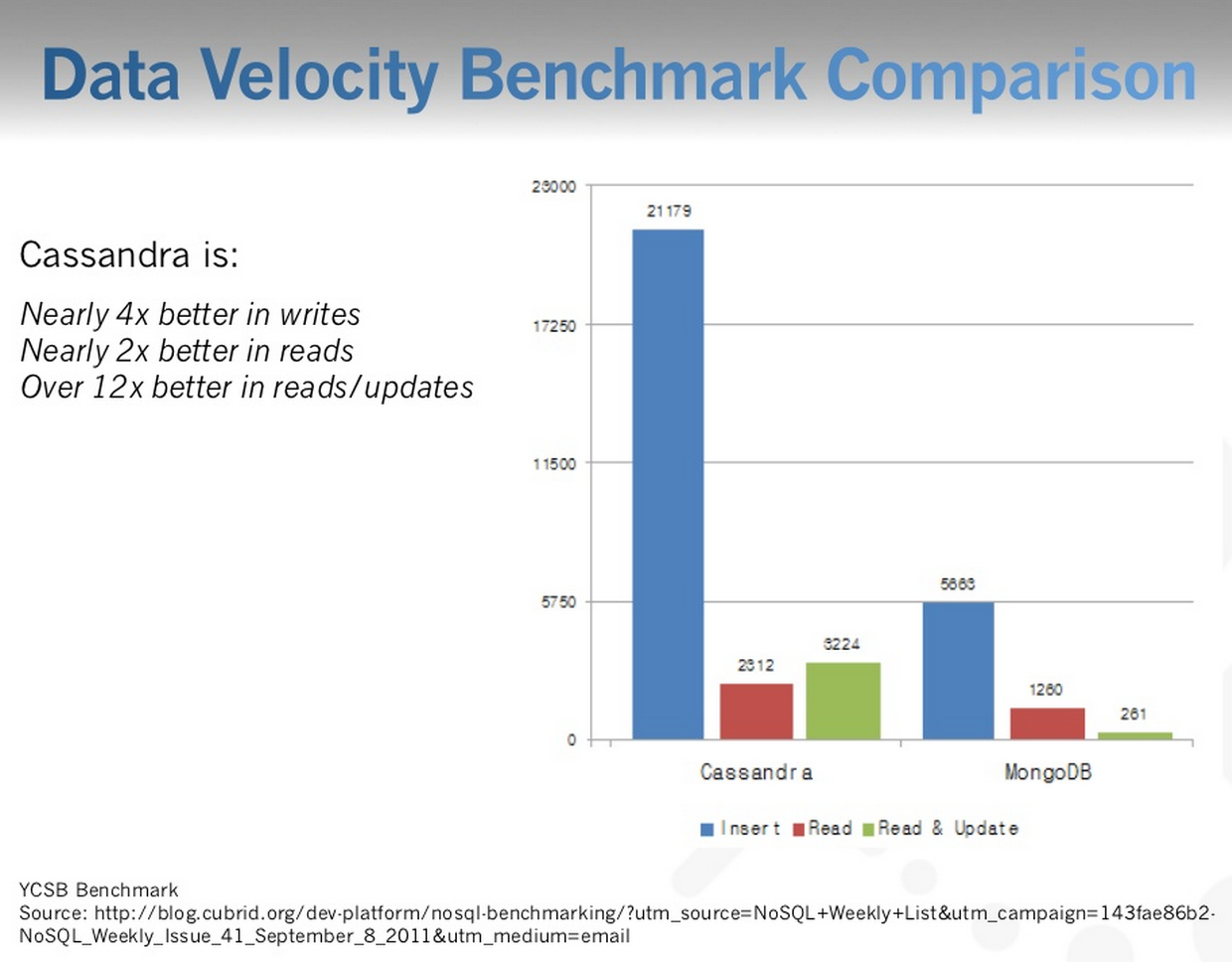
Cassandra also places a high value on performance. In 2012, University of Toronto researchers studying [NoSQL](http://en.wikipedia.org/wiki/NoSQL) systems concluded that "In terms of scalability, there is a clear winner throughout our experiments. Cassandra achieves the highest throughput for the maximum number of nodes in all experiments."[[2]](http://en.wikipedia.org/wiki/Apache_Cassandra#cite_note-2)

Cassandra's data model is a partitioned row store with tunable consistency.[[3]](http://en.wikipedia.org/wiki/Apache_Cassandra#cite_note-tunable_consistency-3) Rows are organized into [tables](http://en.wikipedia.org/wiki/Table_(database)); the first component of a table's primary key is the partition key; within a partition, rows are [clustered](http://en.wikipedia.org/wiki/Clustered_index) by the remaining columns of the key.[[4]](http://en.wikipedia.org/wiki/Apache_Cassandra#cite_note-4) Other columns may be indexed separately from the primary key.[[5]](http://en.wikipedia.org/wiki/Apache_Cassandra#cite_note-5)

Tables may be created, dropped, and altered at runtime without blocking updates and queries.[[6]](http://en.wikipedia.org/wiki/Apache_Cassandra#cite_note-6)

<http://de.slideshare.net/DataStax/top-5-considerations-for-a-big-data-solution>

Vs. Mongo DB



# Provisioning

## Ambari

<http://ambari.apache.org/>

The Apache Ambari project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.

Ambari enables System Administrators to:

* Provision a Hadoop Cluster
  + Ambari provides a step-by-step wizard for installing Hadoop services across any number of hosts.
  + Ambari handles configuration of Hadoop services for the cluster.
* Manage a Hadoop Cluster
  + Ambari provides central management for starting, stopping, and reconfiguring Hadoop services across the entire cluster.
* Monitor a Hadoop Cluster
  + Ambari provides a dashboard for monitoring health and status of the Hadoop cluster.
  + Ambari leverages [Ganglia](http://ganglia.sourceforge.net/) for metrics collection.
  + Ambari leverages [Nagios](http://www.nagios.org/) for system alerting and will send emails when your attention is needed (e.g., a node goes down, remaining disk space is low, etc).

## Vagrant + Hadoop + Ambari

<https://blog.codecentric.de/en/2014/04/hadoop-cluster-automation/>

Installed:

* <http://rubyinstaller.org/> Ruby 2.0.0-p481 (x64)
* VirtualBox 4.3.12
* Vagrant 1.6.3

Angepasst:

* NAT-IP-Adressen (192.168.0.xxx 🡪 192.168.66.xxx + NAT-Setting in VBox)
* URL für ambari  
  alt: <http://public-repo-1.hortonworks.com/ambari/centos6/1.x/GA/ambari.repo>   
  neu: <http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.2.3.7/ambari.repo>

### Not (yet) used:

<http://java.dzone.com/articles/setting-hadoop-virtual-cluster>

<http://hortonworks.com/blog/building-hadoop-vm-quickly-ambari-vagrant/>

# Tutorials

## Hadoop + MongoDB

<http://docs.mongodb.org/ecosystem/tutorial/getting-started-with-hadoop/>

# Others

<http://de.slideshare.net/pavlobaron/20-reasons-why-we-dont-need-architects-pavlobaron>