Reutlingen University

1. Panel DBKDA 2010, Les Menuires, 14.04.2010

NoSQL (not only SQL) – Expert Panel Discussion

Moderation: Fritz Laux, Reutlingen University, Germany

Expert Panelists:

Lena Strömbäck, Linköpings Universitet, Sweden

Yasuhiko Morimoto, Hiroshima University, Japan

Christian Kop, Alpen-Adria-Universitaet Klagenfurt, Austria

Jean-Daniel Cryans, HBase project / Ecole de Technologie Superieure-Montreal, Canada

Reutlingen University

NoSQL "Definition" from nosql-databases.org

- *Next Generation Databases mostly address some of the points: being non-relational, distributed, open-source and horizontal scalable. The original intention has been modern web-scale databases. The movement began early 2009 and is growing rapidly. Often more characteristics apply as: schema-free, replication support, easy API, eventually consistency / BASE (not ACID), and more. So the misleading term "nosql" (the community now translates it mostly with "not only sql") should be seen as an alias to something like the definition above*

- *Unhandy, fuzzy advertising style (discriminating non open source products)*

- *Can we produce a formal definition for NoSQL?*

Reutlingen University

NoSQL Examples and Classification from nosql-databases.org

- *Core NoSQL Systems:*
  - Wide Column Store / Column Families
    - ⇨ Hadoop / HBase, Cassandra, Hypertable
  - Document Store
    - ⇨ CouchDB, MongoDB, Riak
  - Key Value / Tuple Store
    - ⇨ Amazon SimpleDB, Chordless, Redis, Scalaris, BerkeleyDB
  - Eventually Consistent Key Value Store
    - ⇨ Dynamo/KAI/Voldemont
  - Graph Databases
    - ⇨ Neo4J, Sones, InfoGrid
- *Soft NoSQL Systems:*
  - Object Databases
    - ⇨ db4o, Versant
  - XML Databases
    - ⇨ Mark Logic Server, eXist, Xindice, Tamino
- *Is there a tailored solution for every need?*
  Ad-hoc query? Data integrity?

Reutlingen University

What's behind the scene

- *Déjà vu concepts*
  - Hashing is faster [O(1)] than indexing [O(log(n))] or sequential access [O(n)]
  - Navigational access (follow pointer) is faster than lookup (read and compare)
  - Hash-maps do not support approximate access (e.g. Key > Value, Key like M*)
  - Queries orthogonal to the access path are awesome slow no matter of the storage model
  - Shared nothing architecture
- *Anything really new?*

Reutlingen University

Translation Table

Old name	*New name*
Hash file	*Key-Value store*
Hierarchical file (HSAM, HDAM)	*BigTable*
Parent node	*Column family*
Local autonomy	*Partition tolerant*
Horizontal partition	*Sharding*
non-ACID (atomic, consistent, isolated, durable)	*BASE (basically available, soft state, eventually consistent)*

↳ *Any more fashion terms?*

5 /7

Reutlingen University

Pros and Cons of NoSQL

↳ *Pros*

- ☞ Scalability, big data
- ☞ Fast
- ☞ Availability

↳ *Cons*

- ☞ Not ACID (atomic, consistent, isolated, durable)
- ☞ No security (authentication, authorization) ← Dynamo
- ☞ No SQL (No ad-hoc query!)
- ☞ data independence missing
  (No distinction between conceptual and storage model)

↳ *Use it, if you need speed and availability and accept inconsistency including lost transactions!*

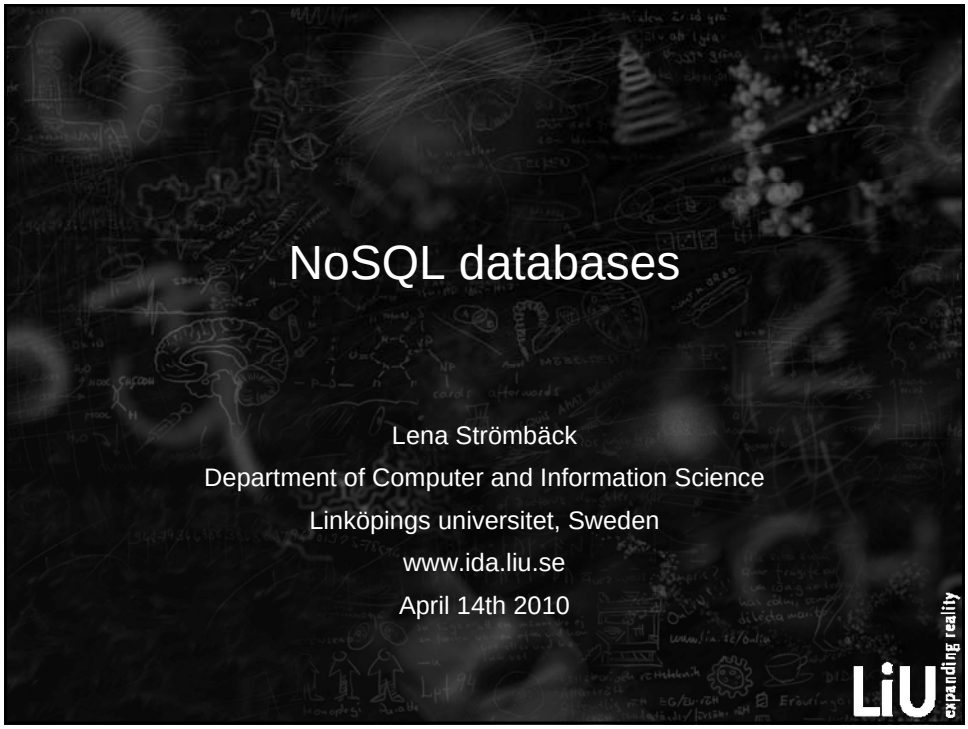
↳ *May we talk about a "Database" if it does not support transactions?*

6 /7

Reutlingen University

Headline Statements

- *Lena Strömbäck: Semantics first*
- *Yasuhiko Morimoto: one size doesn't fit all*
- *Christian Kop: Data quality is important, flexibility endangers quality*
- *Jean-Daniel Cryans: Prepare yourself for the architectures of the future*
- *Fritz Laux: Core NoSQL = fast, scalable file structure*

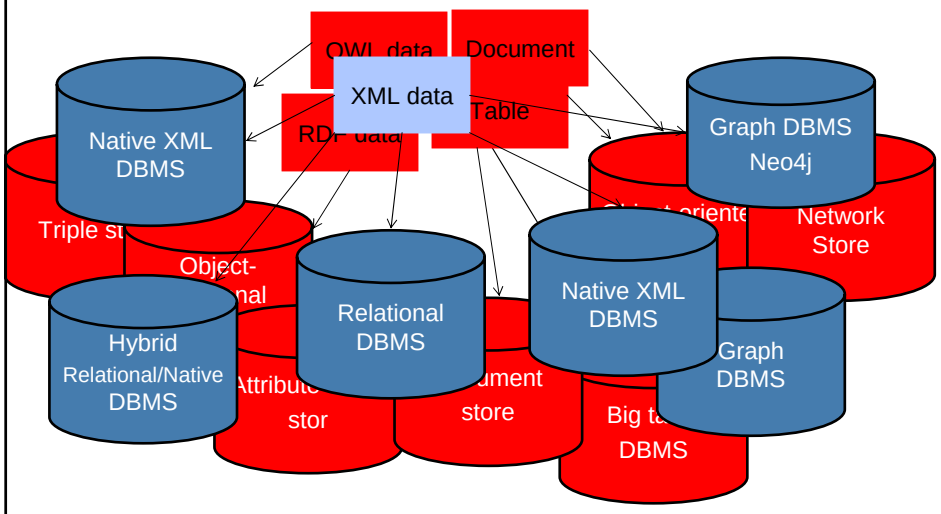


The Web brings new challenges

- Many publishers of data
  - Enterprises – everyday users
- Less well defined than traditional applications
  - Text – semi-structured - structured
- New representation formats
  - XML – RDF – OWL – Standards - text
- Parallel and mixed data models
  - Every user and use for their own needs

LiU

Data storage – work so far…



LiU

Important issues

- What are important properties of data?
  - Data collections
  - Features – semantic concepts
  - Automatic analysis
- How to choose storage solution?
  - Structured comparisons
  - Benchmarking
- How to query the data?
  - Conceptual model
  - Independent of underlying storage model
  - What features important

LiU


Linköping University
expanding reality
www.liu.se

NoSQL: Prepare Yourself for the Architectures of the Future

Jean-Daniel Cryans

Apache HBase committer, part time master student @ ETS Montreal

The Second International Conference on Advances in Databases, Knowledge, and Data Application

April 14th

- NoSQL isn't really about the language, it's more about new ways of storing and accessing data that are better adapted to today's problems.
- The first major issue is scaling, typically you see stories about MySQL. It usually goes like this:
  - Start with a single machine, with a backup to another one, a SAN, etc.
  - Setup Master/Slave replication to spread the **read** load, start getting rid of some indexes because they are too expensive to maintain.
  - Vertically shard the original database by topics because the Master is becoming a bottle neck for the **write** load. Relations are broken between tables, *JOINs* are impossible when two tables are on different shards.
  - Get rid of most the remaining indexes because you can't maintain them since your main bottle neck is, again, inserting data. Tables are now accessed only by **primary key**.

- The second major problem is reliability. Modern RDBMS were designed with a single machine in mind, which becomes the **Single Point Of Failure**. Even with replication, losing a read slave means a significant loss of throughput unless the cluster is over-provisioned… but those servers usually don't come cheap (think *big iron*).
- Finally, as Dr. Stonebraker [1] said, the "One Size Fits All" mentality of current DBMS architecture no longer applies to the database market.

1. Michael Stonebraker, Uğur Çetintemel, "One Size Fits All": An Idea Whose Time Has Come and Gone, pp. 2-11, 21st International Conference on Data Engineering, IEEE Computer Society Press, Tokyo, Japan, April 2005, 0-7695-2285-8.

- Enters NoSQL (Not only SQL).
- Those databases almost all share the quality of being scalable from the beginning, usually not relying on a SPOF.
- They can be used for specific use cases: wide tabular, key-value, document and graph stores, to name a few.
- In the end, we can see that not supporting SQL wasn't a goal, but a necessity since it doesn't fit the supporting data models.
- Personal experience: HBase was initially supporting a SQL-like interactive shell and new users increasingly began asking "How can I do joins?" or "How do I declare an index?". We finally decided to completely change the syntax in order to reduce confusion… and possibly false expectations.

- In conclusion, I'm not arguing that everybody should jump on the NoSQL wagon… It's all about *your* requirements:
  - Do you have Big Data? A few hundred gigabytes isn't considered that much these days and it's easily manageable using well known techniques. Unless you are planning to scale to 10x, 100x?
  - Do you need all the relational features of RDBMS? Would simple lookups be sufficient?
  - Do you have millions of dollars to spend on licenses? If yes, then you probably already have geo-distributed Oracle clusters and you should stop worrying. If not, it's worth researching these solutions.

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT

Application Engineering

Panel
Not only SQL Databases

Christian Kop
Alpen-Adria Universität Klagenfurt

Overview

- The need for other paradigms is not new
- Also at the end of the old century (1990s) databases with other paradigms were also already developed
  - OO Databases
  - Deductive Databases
  - XML Databases
  - …
- Nevertheless, people work with Relational Databases for 30 years – WHY?

Advantages of relational model

- Lean model based on the mathematical concept of a relation
- The operators (selection, projection, join, …) are a basis for a (more) declarative retrieval language.
- Normal forms as well as entity- referential integrity defined for relations help to identify the quality of data.
- In it's core principle also understandable for Non IT Specialist (relation ~ table).
- The relational model is supported by reliable database management systems.

The (organizational) environment

- Seems to be sufficient for general purpose applications
  - Applications in insurance companies, banks, offices in general
- The life time of a technical system is much longer in the IT Practice than in universities (-> budget restrictions).
- The curriculum in databases is still based to a large extend on relational database systems - it's the basis.
- Nearly Everyone gets a data store based on the relational model with his MS Office (MS Access)
- Web ←→ MySQL

The role of RDBMS providers

- Database Management systems providers are clever enough to integrate other “best practices” into their systems.
- ORACLE provides
  - Object types (OO features)
  - XML Documents and Schemas
  - Grid databases
  - …
- What comes next?

Advantages of NoSQL (Why NoSQL)

- Support for a specific problem / situation
- No need to think in terms of relations but in terms given in a situation (e.g. documents, nodes, …)
- In most cases freely available
- In most cases open source

However

- I am afraid that data quality will get lost if the possibilities of NoSQL databases are used too extensively or not in the right domain.

Personal Conclusions

- For special needs, special No SQL databases will be used
- For general purpose application, relational databases will be used

Open Questions / Final Statements

- What can/shall be said about the quality of data in NoSQL databases?
- How much flexibility is acceptable in which situation? (Flexibility and Ad-Hoc Processes NEED Control)

DBKDA 2010 Panel: NoSQL

Yasuhiko Morimoto
Hiroshima University, JAPAN

NoSQL movement:

- Nothing new to me.
- No Single Software (or API) can fit everything.
-SQL cannot fit everything
  though it have been extending (BLOB, Object relational, etc.).
- So called “Key-Value-Stores (KVSs)” have been utilized for more than 5 years
  in emerging data management systems such as e-Commerce, SNS, etc.

RDBMS — KVS (as examples of NoSQL) — File System

ACID guarantee — scalability

trade-off

Data Management software that fits in niche
but important area (infrastructure, application, hardware, etc.)
will increase its importance.

DBKDA 2010 Panel: NoSQL

Yasuhiko Morimoto
Hiroshima University, JAPAN

NoSQL movement:

As a “knowledge discovery and data mining” researcher,

I welcome this movement because ...

if huge data is stored in NoSQL, there is a demand of data mining technology

there must be new challenging open problems in NoSQL Era.

such as “how to find knowledge”
under NoSQL conditions
(ACID guarantee is very weak
difficult find unexpected linkage between one key entry and another
database size is too large to handle etc.)