ifis

Institut für Informationssysteme
Technische Universität Braunschweig

Seminar
NoSQL and NewSQL Databases

Wolf-Tilo Balke
Christoph Lofi
Institut für Informationssysteme
Technische Universität Braunschweig
http://www.ifis.cs.tu-bs.de

Towards NoSQL & NewSQL

- Traditional databases are usually **all-purpose systems**
  - e.g. DB2, Oracle, MySQL, …
  - Theoretically, general purpose DB provide all features to develop any data driven application
  - **Powerful query languages**
    - SQL, can be used to **update** and **query** data; even **very complex analytical queries** possible
  - **Expressive data model**
    - Most data modeling needs can be served by the **relational model**

Towards NoSQL & NewSQL

- **Full transaction support**
  - Transactions are guaranteed to be "safe"
    - i.e. ACID transaction properties
- **System durability and security**
  - Database servers are **resilient to failures**
    - **Log files** are continuously written
      - » Transactions running during a failure can **recovered**
    - Most databases have support for constant **backup**
      - » Even severe failures can be recovered from backups
    - Most databases support "**hot-standby**"
      - » 2nd database system running simultaneously which can take over in case of severe failure of the primary system
  - Most databases offer basic **access control**
    - i.e. **authentication** and **authorization**

Towards NoSQL & NewSQL

- In short, databases could be used as storage solutions in all kinds of applications
- Higher scalability can be achieved with **distributed databases**, showing all features known from classical **all-purpose** databases
  - In order to be distributed, additional mechanisms were needed
    - partitioning, **fragmentation**, allocation, distributed transactions, distributed query processor,….

Towards NoSQL & NewSQL

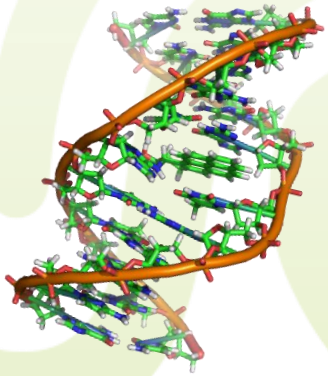
- However, classical **all-purpose databases** may lead to problems in extreme conditions
  - Problems when being faced with **massively** high query loads
    - i.e. millions of transactions per second
    - Load to high for a single machine or even a traditional distrusted database
      - Limited scaling
  - Problems with fully **global applications**
    - Transactions originate from all over the globe
    - **Latency matters**!
      - Data should be geographically close to users
    - Claims:
      - Amazon: increasing the latency by 10% will decrease the sales by 1%
      - Google: increasing the latency by 500ms will decrease traffic by 20%

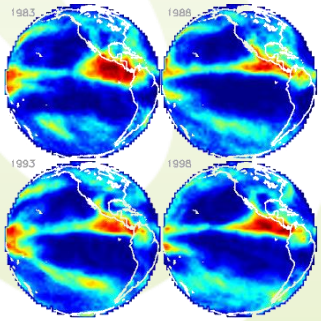
Towards NoSQL & NewSQL

- Problems with extremely high **availability** constraints
  - Traditionally, databases can be recovered using logs or backups
  - Hot-Standbys may help during repair time
  - But for some applications, this is not enough: **Extreme Availability** (Amazon)
    - "… must be available even if disks are failing, network routes are flapping, and several data centers are destroyed by massive tornados"
    - Additional availability and durability concepts needed!

Towards NoSQL & NewSQL

- Problems with **emerging applications** requiring **new data models**
  - Traditional databases rely on the **relational model** which is not optimal for many new applications
    - e.g. scientific data management like genome databases, geo-information databases, etc.
    - e.g. for handling data streams and massive volumes of sensor data
    - e.g. for handling knowledge networks and reasoning





Towards NoSQL & NewSQL

- In extreme cases, specialized database-like systems may be beneficial
  - Specialize on certain query types
  - **Focus on a certain characteristic**
    - i.e. availability, scalability, expressiveness, etc…
  - Allow weaknesses and limited features for other characteristics

Towards NoSQL & NewSQL

- In the recent years, discussing "**NoSQL**" databases have become very popular
  - Careful: big misnomer!
    - Does not necessarily mean that no SQL is used
      - There are SQL-supporting NoSQL systems…
    - NoSQL often refers to "non-standard" architectures for database or database-like systems
      - i.e. system not implemented as shown in RDB2
      - Sometimes, the label **NewSQL** is also used
    - Not formally defined, more used as a "hype" word
  - Popular base dogma: **K**eep **I**t **S**tupid **S**imple!

Towards NoSQL & NewSQL

- The NoSQL movement popularized the development of **special purpose databases**
  - In contrast to **general purpose systems** like e.g. DB2
- NoSQL usually means one or more of the following
  - Being massively **scalable**
    - Usually, the goal is unlimited linear scalability
  - Being massively **distributed**
  - Being extremely **available**
  - Showing extremely high **OLTP performance**
    - Usually, not suited for OLAP queries
  - Not being "**all-purpose**"
    - Application-specific storage solutions showing some database characteristics

Towards NoSQL & NewSQL

- Not using the **relational model**
  - Usually, much simpler data models are used
  - Some, much more complex data models are used (XML, Logic-based, objects, etc.)
- Not using strict **ACID** transactions
  - No transactions at all or weaker transaction models
- Not using **SQL**
  - But using simpler query paradigms
- Especially, not supporting "typical" **query** interfaces
  - i.e. **JDBC**
  - Offering direct access from application to storage system
- System is **cloud-based,** i.e. not installed on a local server
  - System managed by a 3rd party

Towards NoSQL & NewSQL

- In short:
  - Many NoSQL & NewSQL focus on building specialized **high-performance** data storage systems!

Towards NoSQL & NewSQL

- NoSQL and special databases have been popularized by different **communities** and a driven by different design motivations
- Base motivations
  - **Extreme Requirements**
    - Extremely high availability, extremely high performance, guaranteed low latency, etc.
      - e.g. global web platforms
  - **Alternative data models**
    - Less complex data model suffices
    - (More complex) non-relational data model necessary
      - e.g. multi-media or scientific data
  - **Alternative database implementation techniques**
    - Try to maintain most database features but lessen the drawbacks
      - e.g. "traditional" database applications, e.g. VoltDB

Towards NoSQL & NewSQL

Classification	System
KV Cache	Coherence, eXtreme Scale, GigaSpaces, Hazelcast, Infinispan, JBoss Cache, Memcached, Repcached, Terracotta, Velocity
KV Store	Flare, Keyspace, RAMCloud, SchemaFree
KV Store - Eventually consistent	DovetailDB, Dynamo, Dynomite, MotionDb, Voldemort, SubRecord
KV Store - Ordered	Actord, Lightcloud, Luxio, MemcacheDB, NMDB, Scalaris, TokyoTyrant
Tuple Store	Apache River, Coord, GigaSpaces
Object Database	DB4O, Perst, Shoal, ZopeDB,
Document Store	Clusterpoint, CouchDB, MarkLogic, MongoDB, Riak, XML-databases
Wide Columnar Store	BigTable, Cassandra, HBase, Hypertable, KAI, KDI, OpenNeptune, Qbase
Array Databases	SciDB, PostGIS, Oracle GeoRaster, Rasdaman
Stream Databases	StreamSQL, STREAM, AURORA
Analytical Column Stores	Vertica, SybaseIQ
High Throughput OLTP	VoltDB, Hana

Seminar Topics

1. **Highly Scalable Document-Based Databases**
   - **Rebecca**
   - Examples: CouchDB and MongoDB
2. **Highly Scalable Wide Column Stores**
   - **Shafig**
   - Examples: Hbase, Cassandra, BigTable
3. **Array Databases**
   - Nobody ☹
   - Examples: SciDB & Rasdaman
4. **TripleStores & GraphDBs**
   - **Felix**
   - Examples: AllegroGraph, Virtuoso, SparqlDB, Jena

Seminar Topics

5. **Stream Databases**
   - Nobody ☹
   - Examples: StreamSQL, STREAM, AURORA
6. **NewSQL In-Memory Systems**
   - **Lars**
   - Examples: VoltDB, SAP Hana
7. **Analytic Column Stores**
   - **Yevgen**
   - Examples: Vertica, Sybase IQ

Seminar Schedule

	Date	Topic	Who?
0	06.03.2014	Kick-Off	
1	16.04.2014	Introduction, Assignment of topics	
2	23.04.2014	Analyzing talks 1	
3	30.04.2014	Analyzing talks 2	
4	07.05.2014	Developing guidelines for good talks	
5	14.05.2014	Preliminary Results (private meetings)	
6	21.05.2014	Talk 1	
7	28.05.2014	Talk 2	
8	04.06.2014	Talk 3	
	11.06.2014	Exkursionswoche (no class)	
9	18.06.2014	Talk 4	
10	25.06.2014	Talk 5	
11	02.07.2014	Talk 6	
12	09.07.2014	Talk 7	
13	16.07.2014	Talk 8	
14	23.07.2014	Grading	