

# The NoSQL-Movement

Sergey Serebryakov, 545

based on “NoSQL Databases” by Christof Strauch

# Motives I

- Avoidance of unneeded complexity
- High throughput
- Horizontal scalability and running on commodity hardware
- Avoidance of expensive object-relational mapping
- Compromising reliability for better performance

# Motives II

- The current “One size fits all” databases thinking is wrong
- The myth of effortless distribution and partitioning of centralized data models
- Movements in programming languages and development frameworks
- Requirements of cloud computing
- The RDBMS plus caching layer vs. Systems built from scratch with scalability in mind

# Drivers

- Google (BigTable)
- Facebook (Cassandra)
- LinkedIn (Project Voldemort)
- Amazon (Dynamo)
- others: adopted and modified

# Criticism

- Scepticism on the business side
- NoSQL as a hype
- NoSQL as being nothing new
- NoSQL meant as a total “No to SQL”  
(NOSQL / LessSQL / post-relational)
- Requirements of administrators and operators

# Misconceptions of critics

- “NoSQL is just about scalability and/or performance”
- “NoSQL is just document databases, or key-value stores, or...”
- “I can do NoSQL just as well in a relational database”
- “NoSQL is a wholesale rejection of relational databases”

# Classification by Data Model

<b>Datastore</b>	<b>Data Model</b>
Cassandra	Columnfamily
CouchDB	Document
HBase	Columnfamily
MongoDB	Document
Neo4j	Graph
Redis	Collection
Scalaris	Key/value
Tokyo Cabinet	Key/value
Voldemort	Key/value

# Classification by Persistence Design

<b>Datastore</b>	<b>Persistence Design</b>
Cassandra	Memtable / SSTable
CouchDB	Append-only B-tree
HBase	Memtable / SSTable on HDFS
MongoDB	B-tree
Neo4j	On-disk linked lists
Redis	In-memory with background snapshots
Scalaris	In-memory only
Tokyo Cabinet	Hash or B-tree
Voldemort	Pluggable (primarily BDB MySQL)