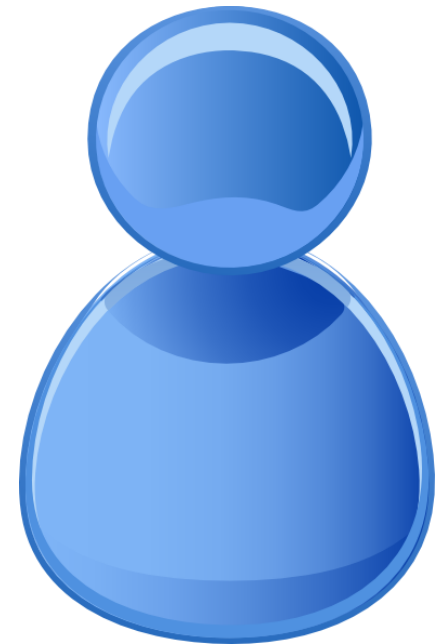


NoSQL

06.06.2013

About me

- Markus Deutschl
- BSc (FH Joanneum – ITM09)
- currently reaching for MSc
- <http://movlib.org>



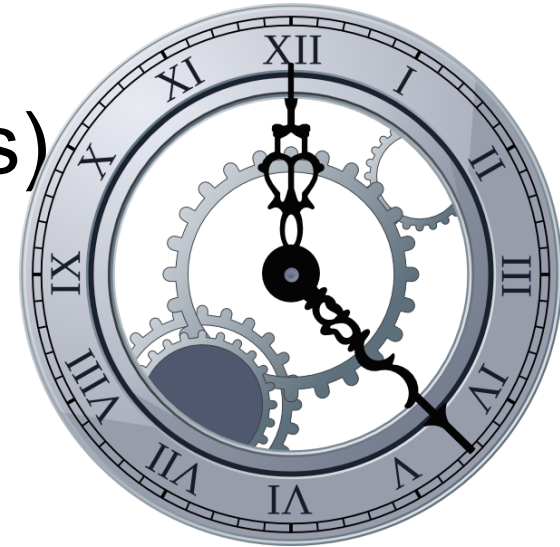
Agenda

- Overview & Definition
- Consistency
- Scaling & MapReduce
- Types of NoSQL
- Demo?



Historical overview

- DBM (1979)
- Lotus Notes, BerkeleyDB (80's)
- NoSQL (1998)
- Web 2.0 – BigTable (2004)
- The movement (2009)



Definition

“Next Generation Databases mostly addressing some of the points: being **non-relational, distributed, open-source** and **horizontally scalable**.

The original intention has been **modern web-scale databases**. The movement began early 2009 and is growing rapidly. Often more characteristics apply such as: **schema-free, easy replication support, simple API, eventually consistent / BASE** (not ACID), a **huge amount of data** 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.”

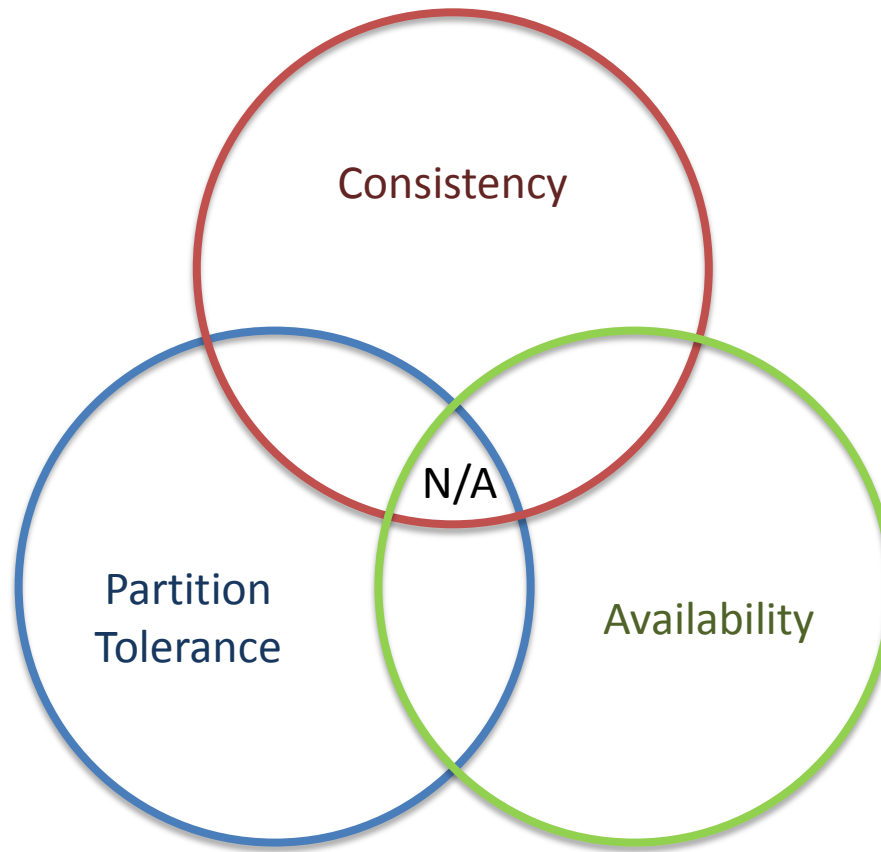
- nosql-database.org

Clarification

- Web 2.0 needs
- Scalability
- Flexible data models
- Different storage
- Consistency?

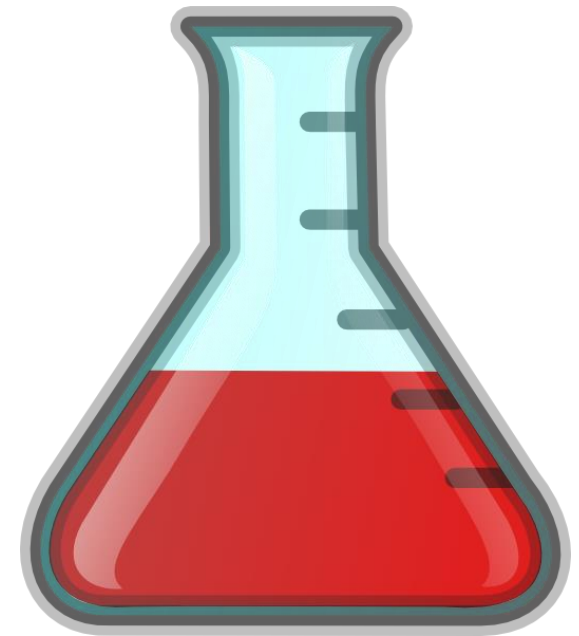


Implications



Strong consistency

- **Atomicity**
- **Consistency**
- **Isolation**
- **Durability**



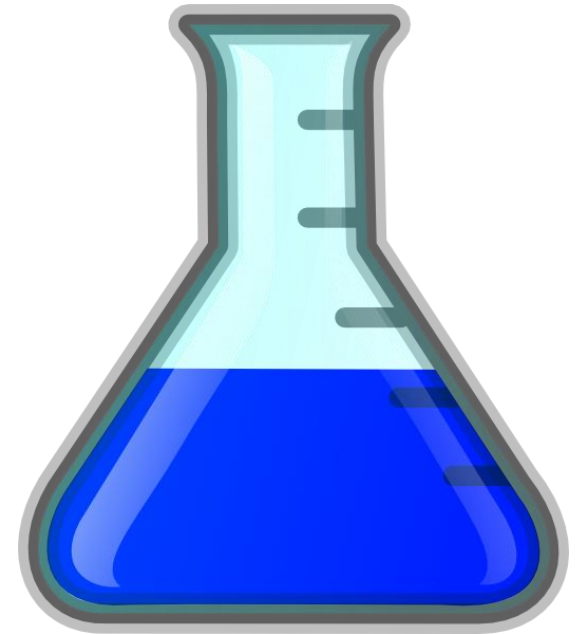
Eventual consistency

- Casual
- Read-your-writes
- Session
- Monotonic read
- Monotonic write



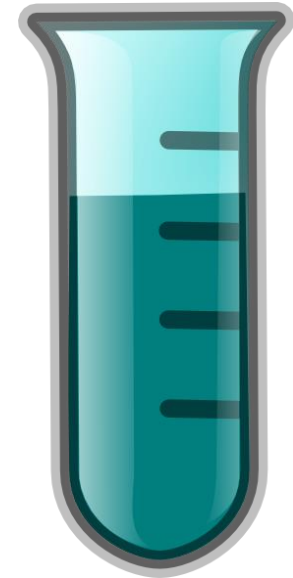
BASE

- **B**asically **A**vailable
- **S**oft state
- **E**ventually consistent



Scaling

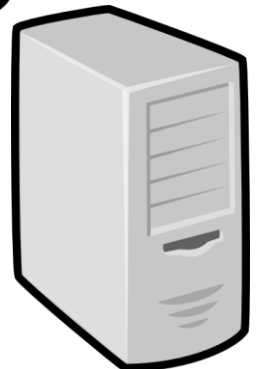
- Requests
- Data capacity
- Performance



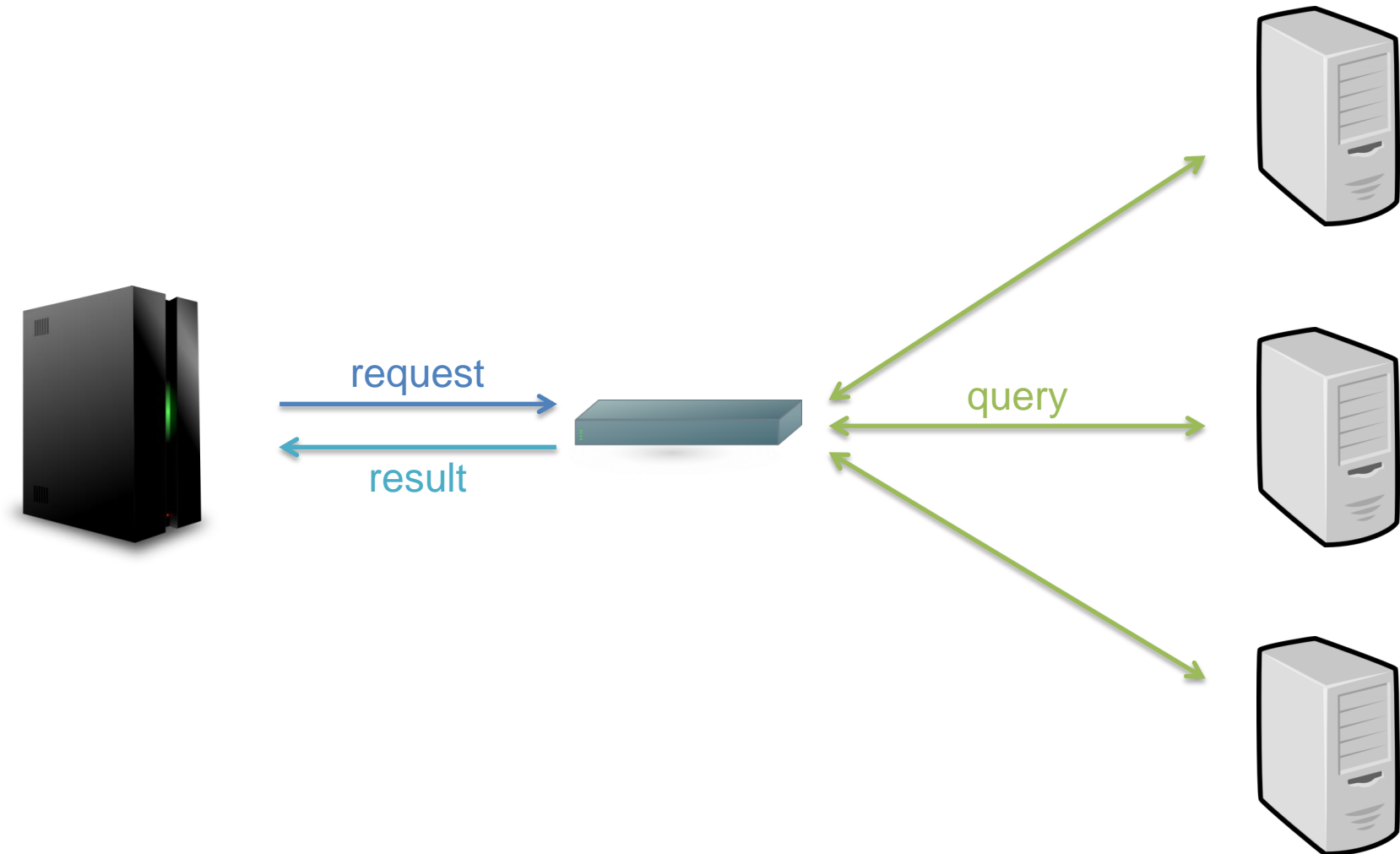
Scaling approaches



VS.

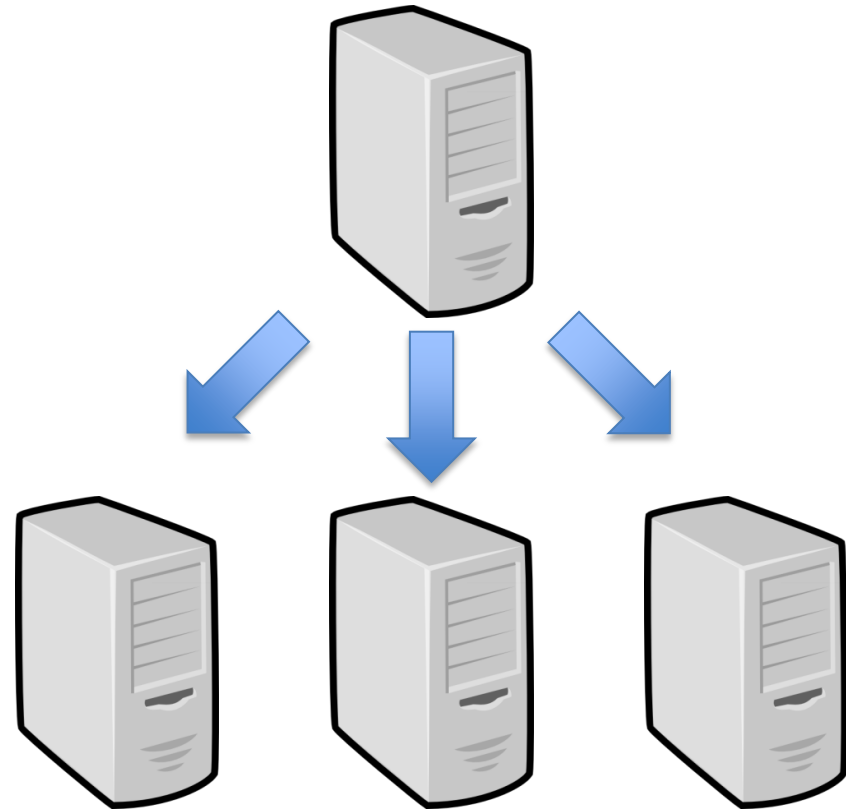


Sharding

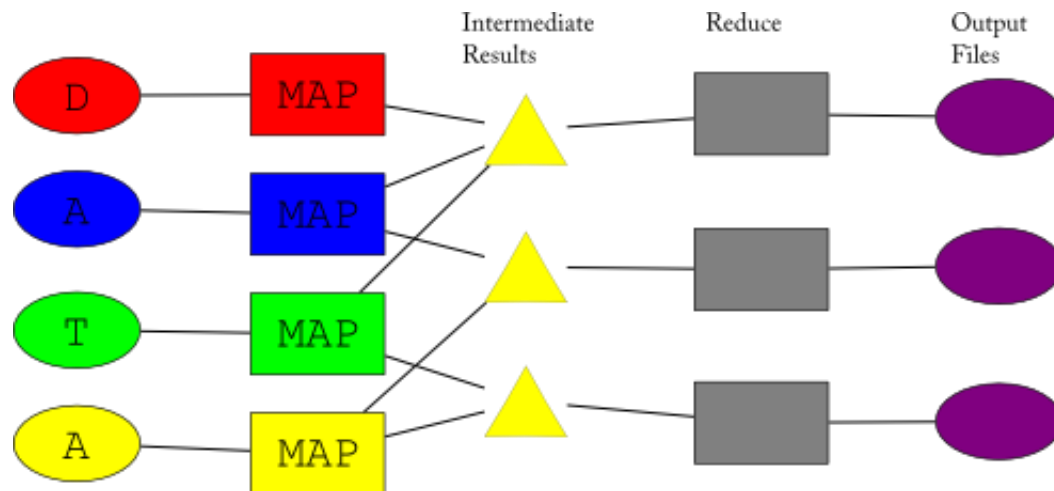


MapReduce

- Huge amounts of data
- Parallel computing
- Distribution
- Efficiency



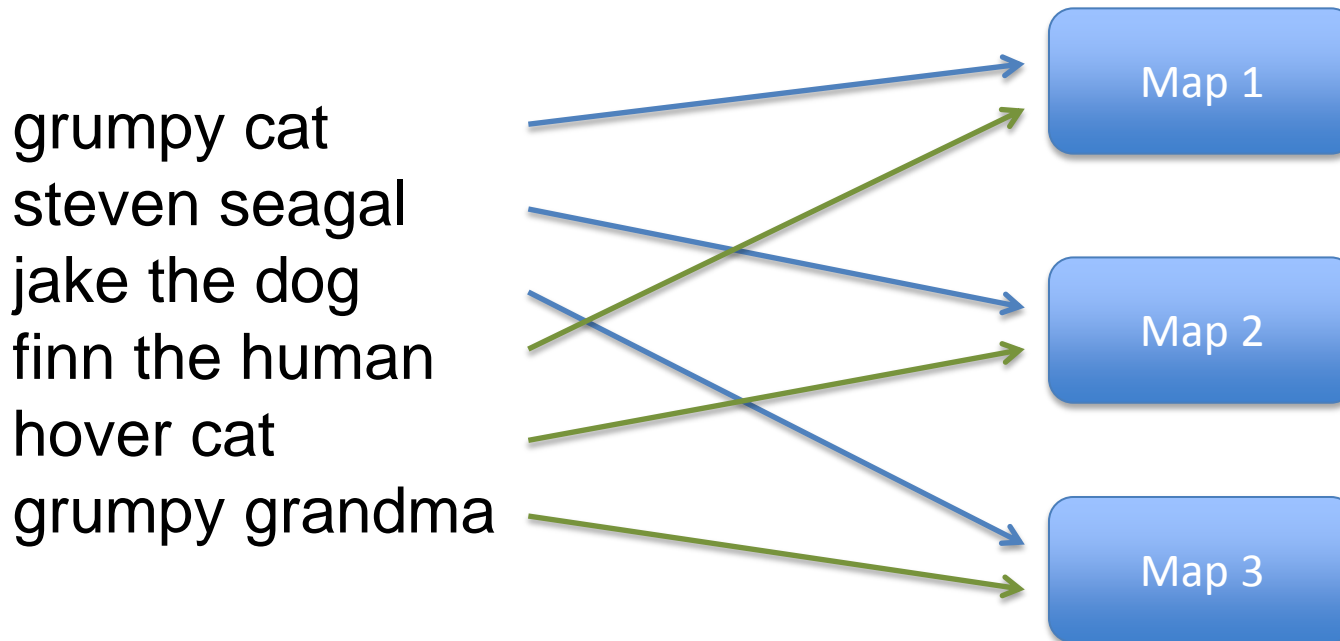
MapReduce



Example – word count

grumpy cat
steven seagal
jake the dog
finn the human
hover cat
grumpy grandma

Map phase



Map function

```
function map(line) {  
    line.split(" ").forEach(function (word) {  
        emit(word, 1);  
    });  
}
```

Map output

```
grumpy, 1  
cat, 1  
steven, 1  
seagal, 1  
jake, 1  
the, 1  
dog, 1  
finn, 1  
the, 1  
human, 1  
hover, 1  
cat, 1  
grumpy, 1  
grandma, 1
```



shuffle / group

```
grumpy  
1, 1
```

```
cat  
1, 1
```

```
steven  
1
```

```
seagal  
1
```

```
jake  
1
```

```
the  
1, 1
```

```
dog  
1
```

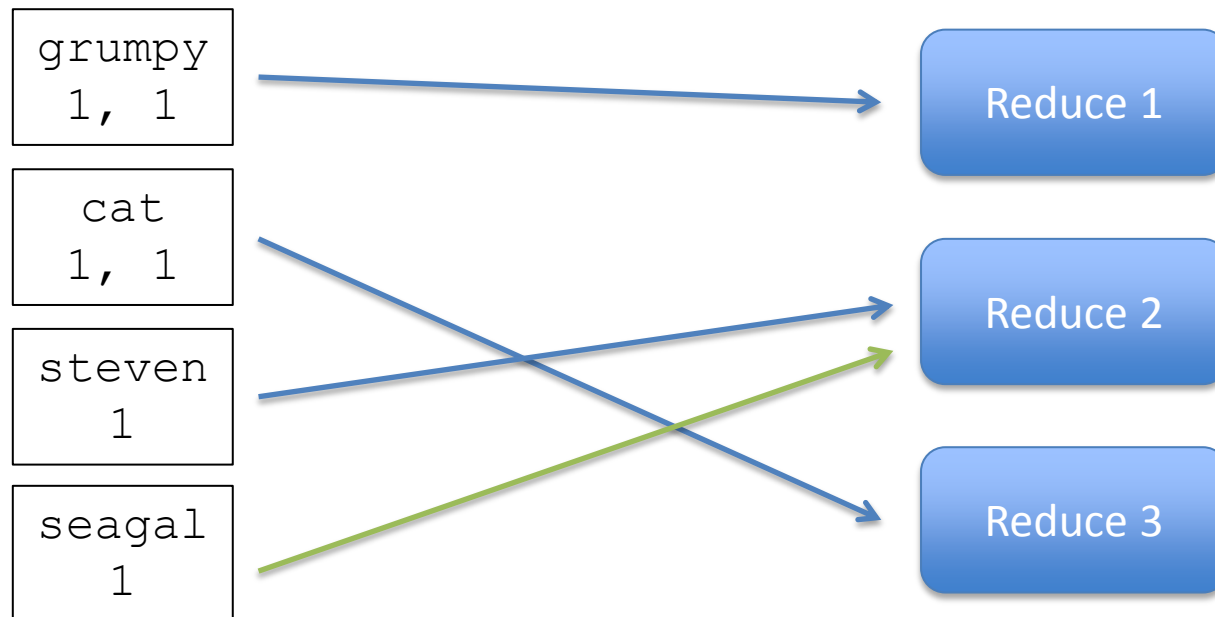
```
finn  
1
```

```
human  
1
```

```
hover  
1
```

```
grandma  
1
```

Reduce phase



Reduce function

```
function reduce(key, values) {  
    var sum = 0;  
    values.forEach(function (value) {  
        sum += value;  
    });  
    return sum;  
}
```

Output

```
grumpy, 2  
cat, 2  
steven, 1  
seagal, 1  
jake, 1  
the, 2  
dog, 1  
finn, 1  
human, 1  
hover, 1  
grandma, 1
```



Types of NoSQL

- Core NoSQL
- Soft NoSQL
 - Object
 - Grid & Cloud
 - XML
 - Multidimensional
 - Multivalued



Key/Value stores

- Simple model
- Access by key
- Limited queries
- Low latency
- Record independence



KVS – Data types

- Primitives
- Lists
- Sets
- Objects
- Dictionaries

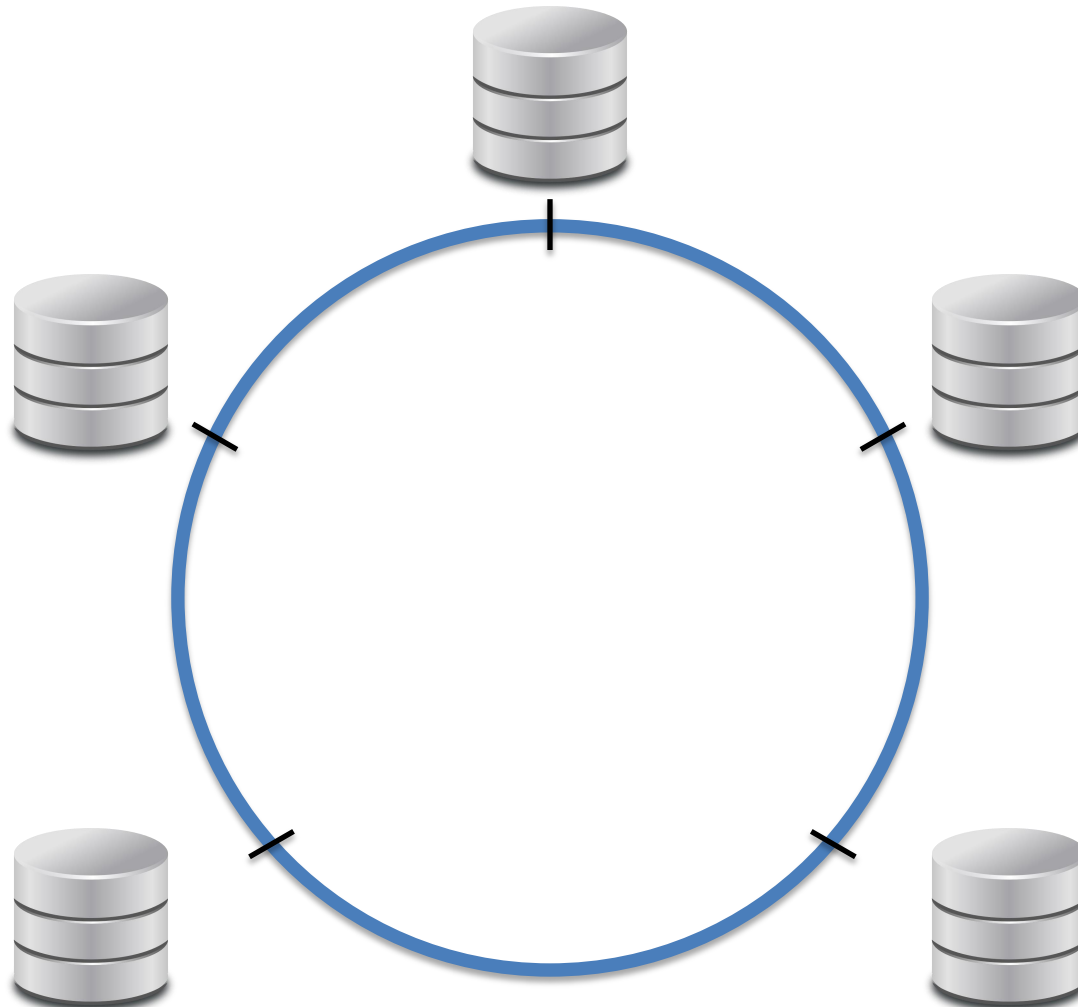


KVS – Operations

- GET
- SET
- PUT
- DELETE
- ...



KVS – Scaling



KVS – Use

- Caching
- Independent data
- Unique keys
- Scalability



Document stores

- Semi-structured format
- Mostly JSON
- Complex queries
- Flexible schema
- Validation



DS – Data format

id	name	tel	fax
1	Chuck Norris	01234	null

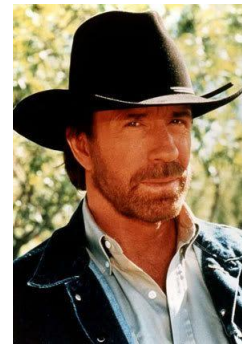
```
{  
  _id: 1,  
  name: "Chuck Norris",  
  tel: 01234  
}
```



Chuck Norris

Tel.: 01234

Fax:



Chuck Norris

Tel.: 01234

DS – Features

- Replication
 - Master-Slave
 - Master-Master
- Sharding
- Rapid development



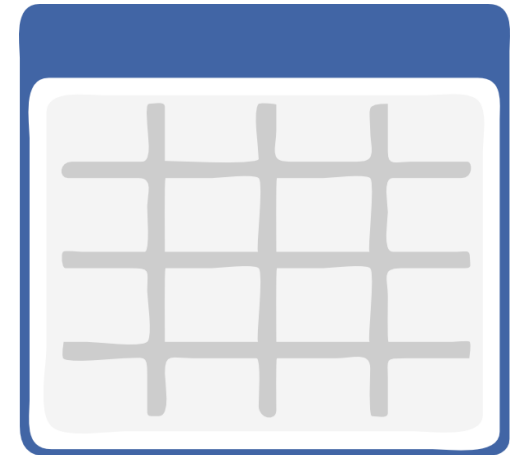
DS – Use

- Semi-structured data
- Flexibility
- Querying possibilities
- App-shaped DB
- Integration



Column families

- Data in columns
- Flexible “Rows”
- Column families
- Super columns
- Access by key



CF – Data model

Column Family: Persons	
1	2
"name": "Chuck Norris"	"name": "Steven Seagal"
"tel": 01234	"tel": 0815
	"fax": 4711

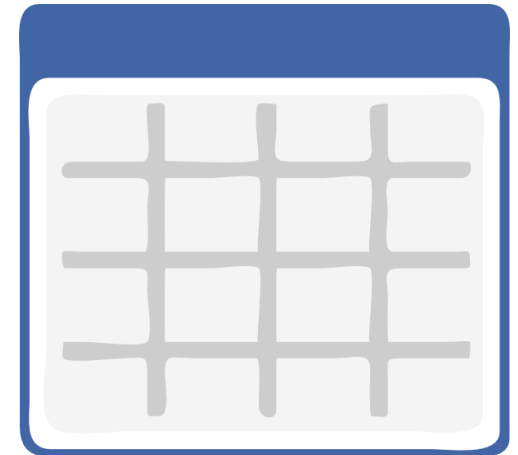
Column Family: Persons		
	1	
		"name": "Chuck Norris"
		"tel": 01234
	2	
		"name": "Steven Seagal"
		"tel": 0815
		"fax": 4711

CF – Super columns

Column Family: Movies		
"Nor_Movies"		
	"The Delta Force"	
		"year": 1986
		"country": "USA"
	"The Way of the Dragon"	
		"year": 1972
		"country": "HK"
"Sea_Movies"		
	"Machete"	
		"year": 2010
		"country": "USA"

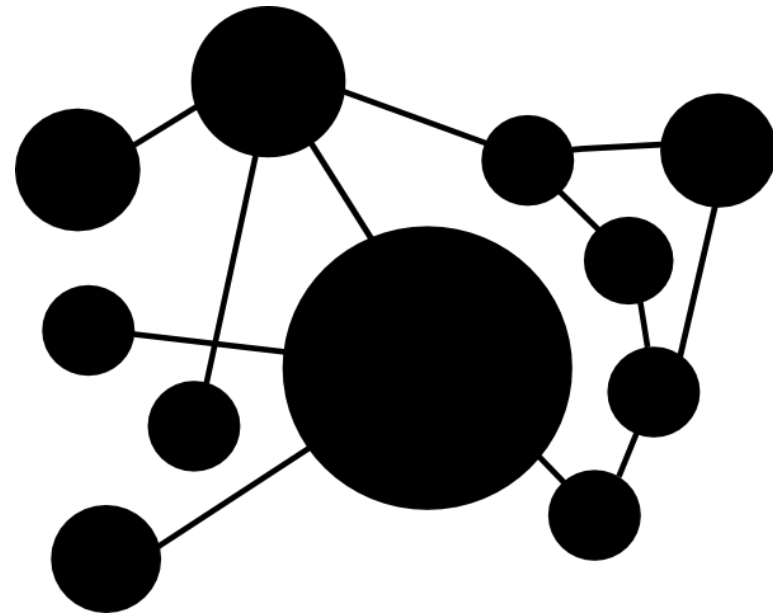
CF – Use

- Flexible data model
- Huge data amount
- Scalability
- Efficiency
- Data analysis

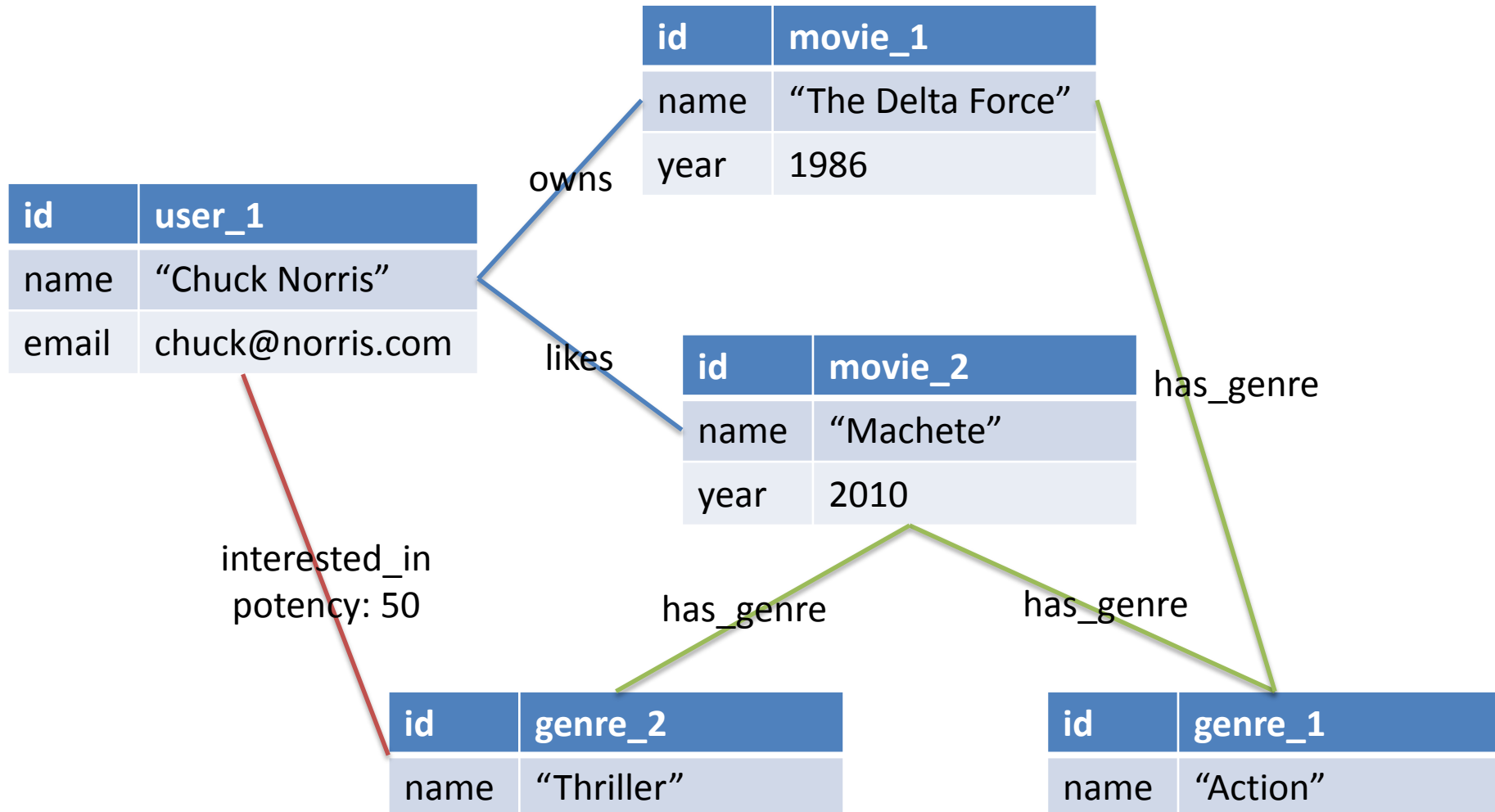


Graph databases

- Interconnected data
- Flexible nodes/edges
- Graph as data model
- Traversing

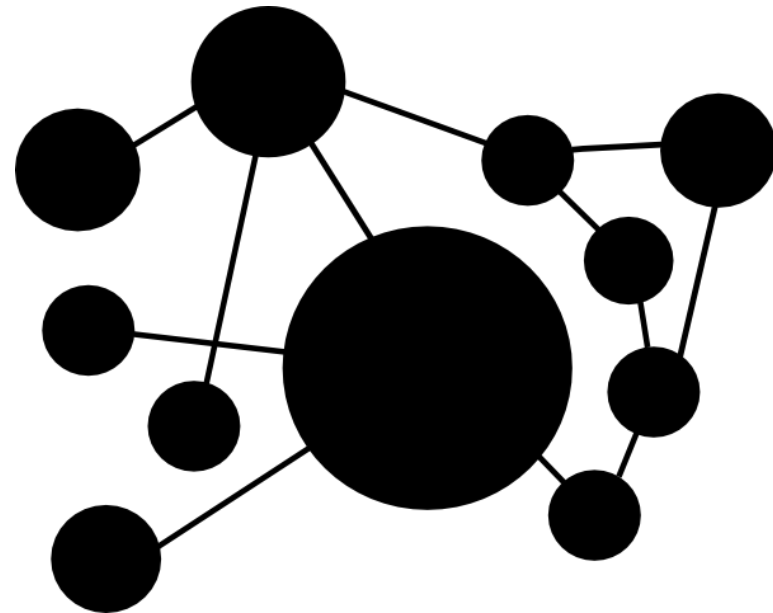


GD – Data model



GD – Use

- Interconnected data
- Graphs and trees
- Scientific data
- Traversing desirable



RDMS – doomed to die?

- Maturity
- Transactions
- Querying capabilities
- Knowledge
- Jack-of-all-trades schemas
- Performance



Drawbacks of NoSQL

- Querying capabilities
- Limited features
- Maturity
- Many different systems
- Interchangeability



Reading

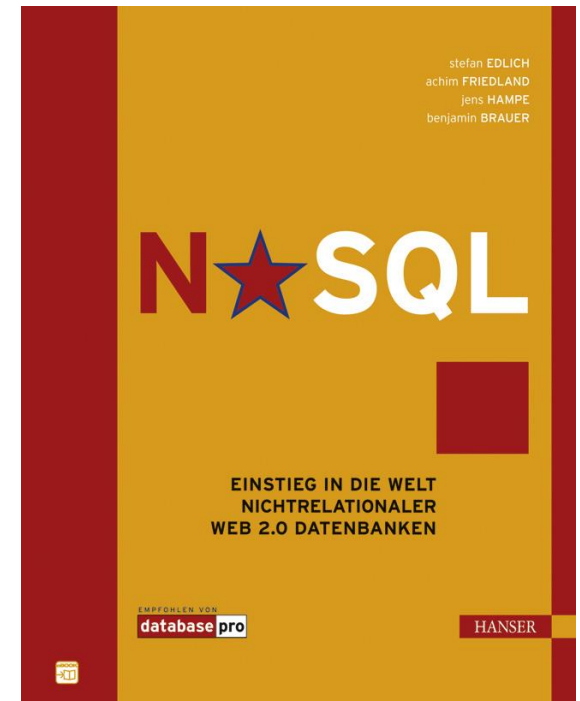
NoSQL –

Einstieg in die Welt nichtrelationaler
Web 2.0 Datenbanken

S. Edlich, A. Friedland, J. Hampe, B. Brauer
2010

2nd Edition (2011)

Hanser, ISBN: 978-3-446-42753-2



Reading

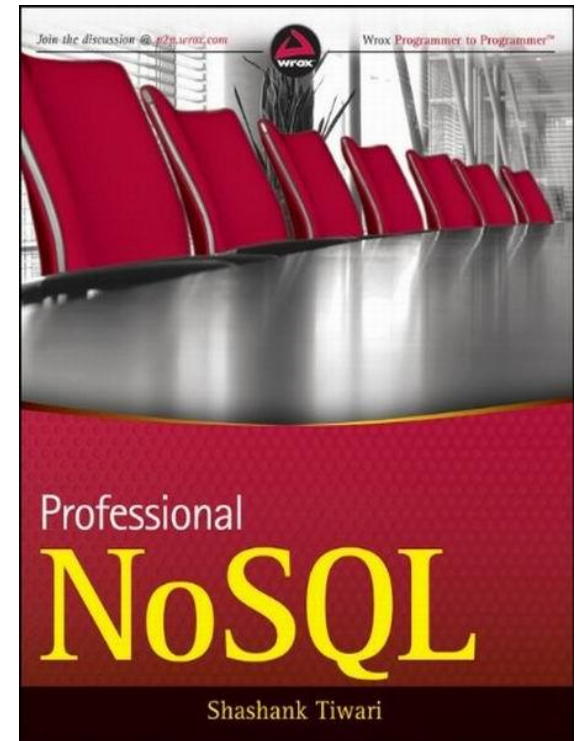
Professional NoSQL

S. Tiwari

2011

Wrox, ISBN: 978-1-4571-0685-9

<http://it-ebooks.info/book/812/>



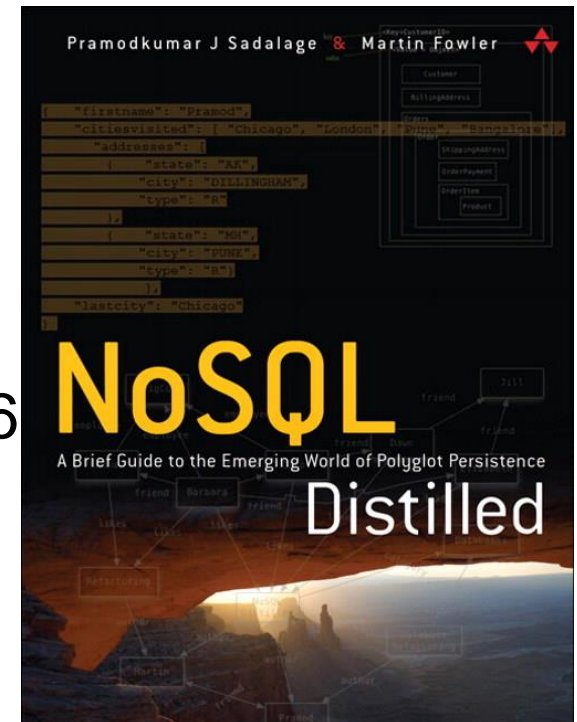
Reading

NoSQL Distilled

P. J. Sadalage, M. Fowler

2012

Addison-Wesley, ISBN: 978-0321826626



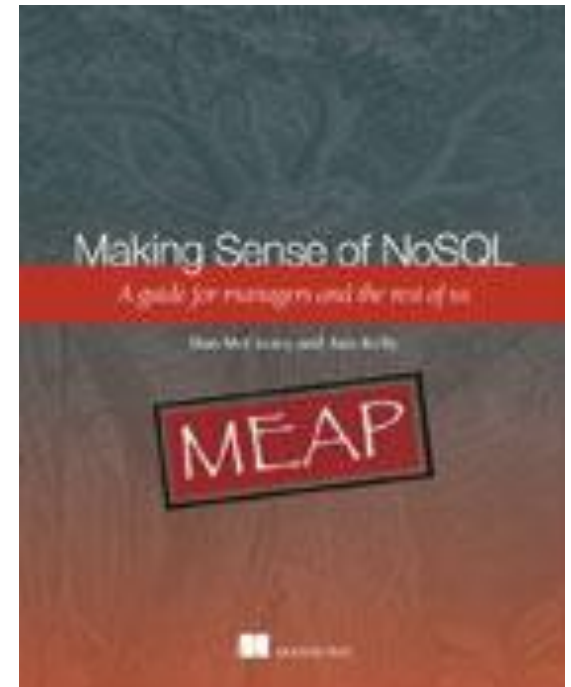
Reading

Making sense of NoSQL

D. McCreary, A. Kelly

August 2013 (est.)

Manning, ISBN: 9781617291074

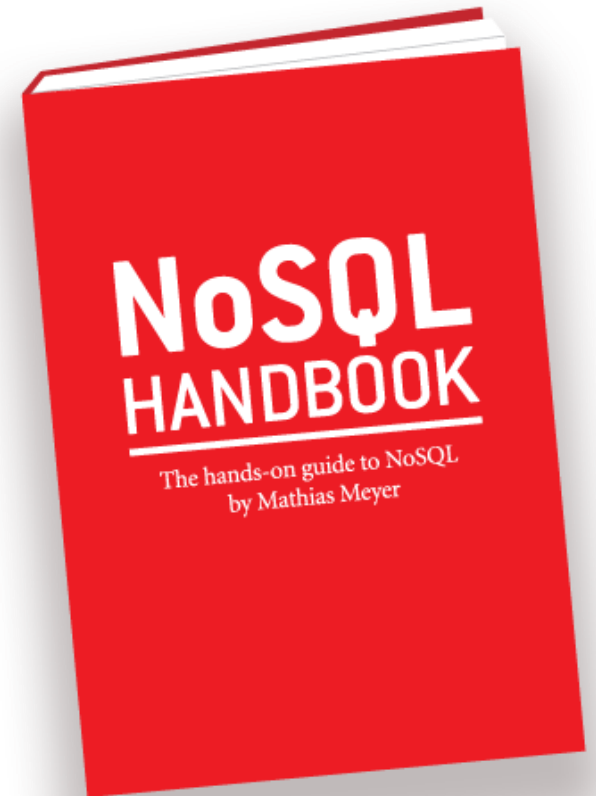


Reading

NoSQL Handbook

M. Mayer
(not fixed yet)

<http://nosqlhandbook.com/>



Questions



MongoDB

- Multi-Platform
- Databases
- Collections
- Relationships
- File store
- Drivers



MongoDB – Specs

- Doc size 16MB
- Master-Slave
- Autosharding
- Indexes
- Queries on contents



MongoDB – Data types

- BSON
- String, Array, Bool, Number
- Date / Timestamp
- RegEx
- Code
- Document

