Choosing the right NoSQL database

Juozas *"Joe"* Kaziukėnas

http://juokaz.com / juozas@juokaz.com / @juokaz

Who is this guy?

- Juozas Kaziukėnas, Lithuanian
- You can call me *Joe*
- ~4 years in Edinburgh, UK
- CEO of Web Species Ltd
- *Occasional* open source developer
- Conferences speaker
- More info in **http://juokaz.com**
- Tweet me @juokaz

Do not throw away MySQL

Don’t throw away MySQL

- Reliability
- Relational model
- Transactions
- SQL
- Integration

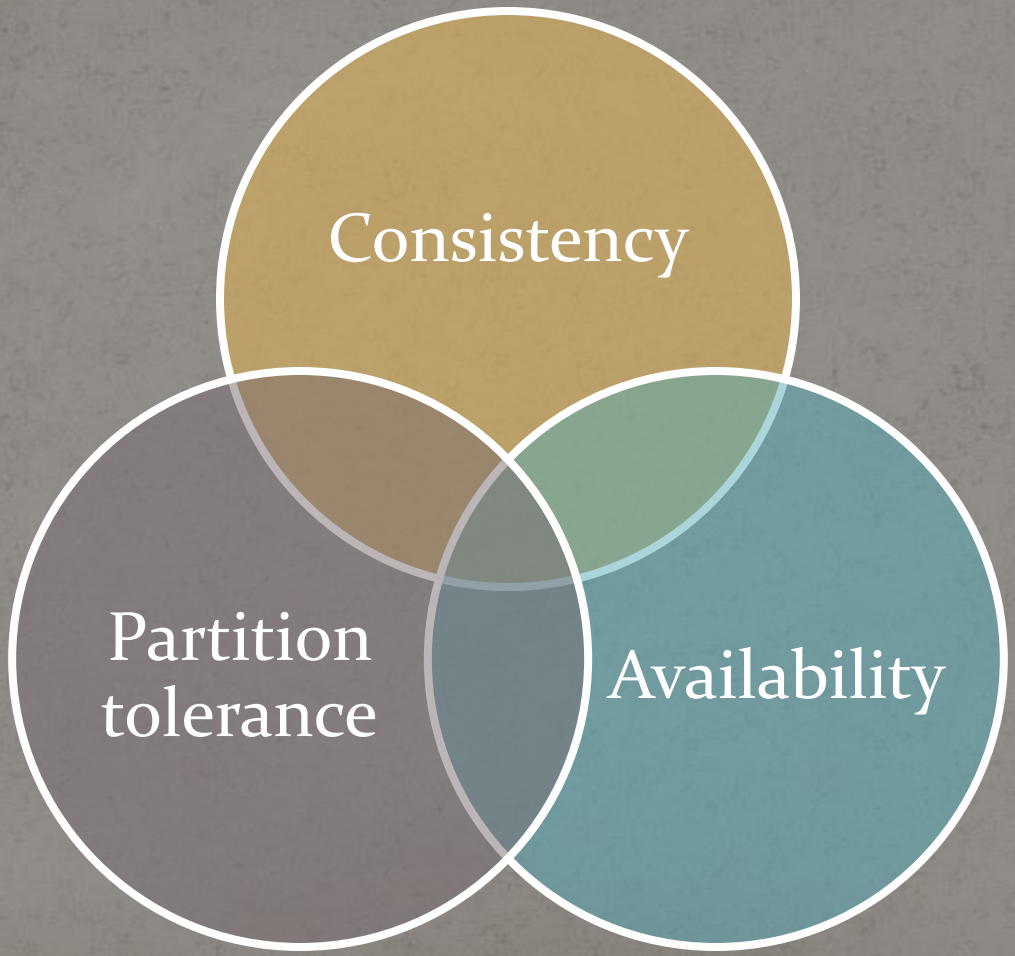
ACID

- Atomicity
- Consistency
- Isolation
- Durability

Problems with RDBMS

- Vertical scalability
  - Hardware (memory) limits
- Horizontal scalability
  - Joins
  - Transactions
- Consistency is a major bottleneck

CAP theorem



* http://en.wikipedia.org/wiki/CAP_theorem

NoSQL birth

NoSQL

- Data driven projects
- A lot of data
- Real-time analysis
- Google BigTable and Amazon Dynamo

Relaxed ACID

- Acid:
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- Hard to implement in distributed systems
- Eventual consistency

Schema

- Schema-less
- Types
  - Key Value
    - Dynamo, Membase, Riak, Redis
  - Document
    - MongoDB and CouchDB
  - Graph
    - Neo4j, FlockDB
  - Column
    - Big Table, Cassandra, Hbase

Business defines architecture

Business defines architecture

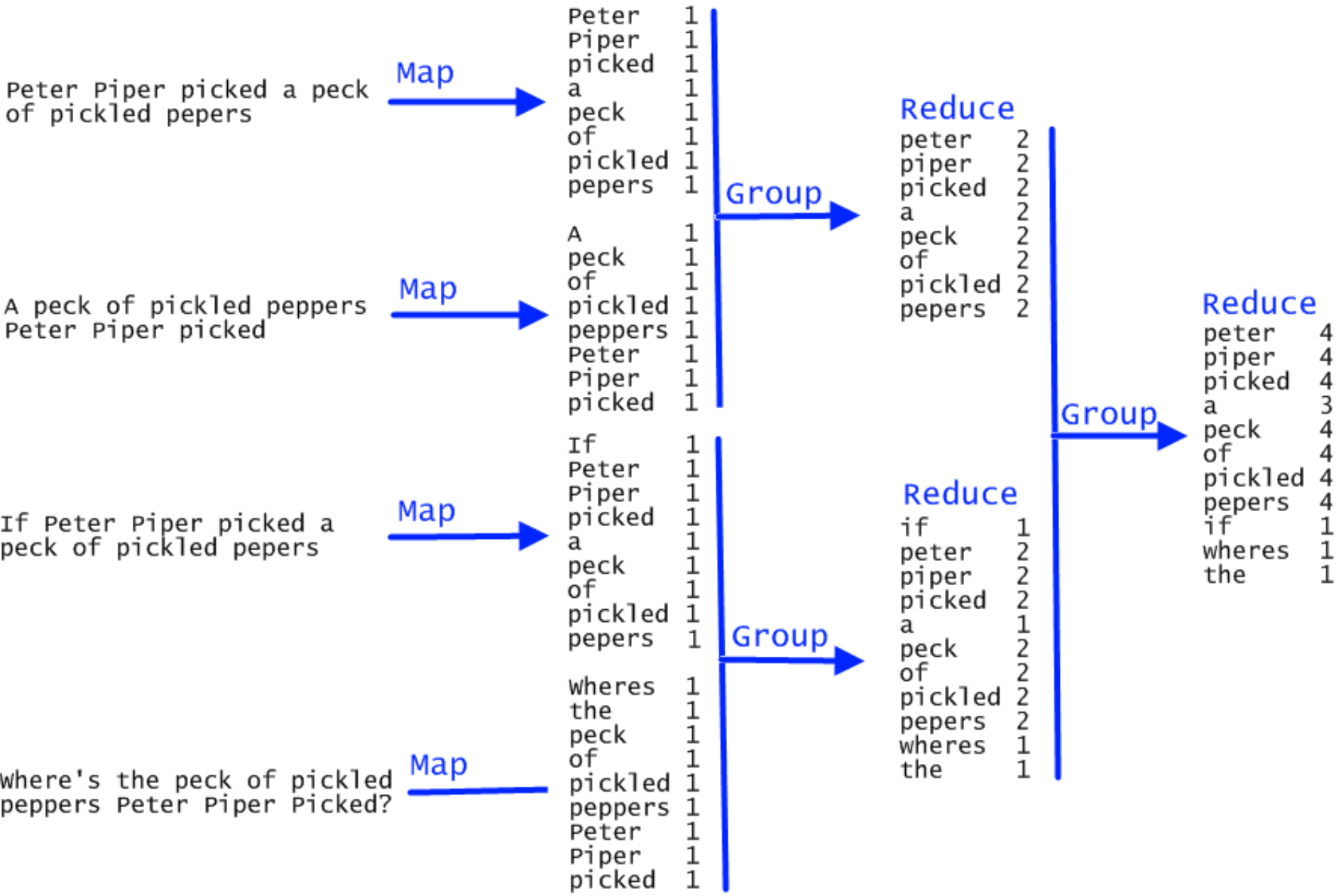
- Understand
  - Business model
  - Use cases
  - Size
  - Requirements
- Do not over-engineer, it will fail anyway
- Do not lock-in

Access patterns

- Dynamic queries
  - Index data
- Map/Reduce
- Key lookups

Map/Reduce

- Created by Google
- Process data using mappers and reducers
- Can be distributed on any amount of machines
- Popular to use with Hadoop


Peter Piper picked a peck of pickled pepers
Map
Peter 1
Piper 1
picked 1
a 1
peck 1
of 1
pickled 1
pepers 1
A peck of pickled peppers Peter Piper picked
Map
A 1
peck 1
of 1
pickled 1
peppers 1
Peter 1
Piper 1
picked 1
If Peter Piper picked a peck of pickled pepers
Map
If 1
Peter 1
Piper 1
picked 1
a 1
peck 1
of 1
pickled 1
pepers 1
Where's the peck of pickled peppers Peter Piper Picked?
Map
Wheres 1
the 1
peck 1
of 1
pickled 1
peppers 1
Peter 1
Piper 1
picked 1
Group
Group
Reduce
peter 2
piper 2
picked 2
a 2
peck 2
of 2
pickled 2
pepers 2
Reduce
if 1
peter 2
piper 2
picked 2
a 1
peck 2
of 2
pickled 2
pepers 2
wheres 1
the 1
Group
Reduce
peter 4
piper 4
picked 4
a 3
peck 4
of 4
pickled 4
pepers 4
if 1
wheres 1
the 1

If you don't need secondary-indices anything will work

You will fail

- Distributed systems are tricky
- Databases are buggy…
- Foursquare, Tumblr, Twitter and more publicly failed
- Outage
- Data loss
- Consistency problem

Performance is not your goal*

Choosing

NoSQL vs RDBMS

NoSQL > SQL

- Horizontal scalability
- High write OR read throughput
- Stores any data

NoSQL < SQL

- No partial reads
- No security
- No relational model
- Only stores data, no reporting, aggregating

Pick

- Distribution model
  - Dynamo like
  - Master-Master
  - Master-Slave
- Query model
  - Map/Reduce
  - Dynamic queries
- Disk structure
  - How database is persisted on a disk

Redis

- In-memory database (needs to fit in memory*)
  - Eventual consistency in disk
- Master-slave replication
- Key-value, but also sets, lists and hashes
- Supports transactions
- Good for expiring and/or rapidly changing data

MongoDB

- Master/Slave replication
- Sharding
- Dynamic queries
  - Using JavaScript expressions
- Update-in-place with atomic operations
- Can store files
- For anything MySQL would be used for, but schema-less is required
- Used to be unreliable on a single machine

CouchDB

- Bi-directional replication. Master-master
- Versioning and conflict detection
- Always consistent
- Needs compacting, not good for rapid changing data
- Map/Reduce as query mechanism
- Real-time data updates feed (_changes)
- Document validation
- Best for offline systems. Great for content stores

Cassandra

- Faster writes than reads
- Query by column
- Secondary indices
- Map/reduce possible with Hadoop
- Complex, Java system
- Used to store a lot of data

HBase

- Map/Reduce with Hadoop
- Random access
- Real-time read/write access

Neo4j

- Graph database
- Master-slave
- Path finding
- Optimized for reads
- For complicated interconnected data

Database will fail

Database will fail

- Measure
  - Memory
  - Disk I/O
  - CPU utilization
- Don't try to make a database do things it wasn't designed for
- Create a non-relational model
- Denormalize

Thank you!

Keep in touch

- http://juokaz.com
- juozas@juokaz.com
- twitter: @juokaz