

FIRST INDICO WORKSHOP

27-29 MAY 2013 CERN

Database Technology

Pedro Ferreira



THE FUTURE

Let's start from the end

A more «social» Indico

User-oriented approach

Personal home page, user-tailored information

More data in less time

Need for an adequate DB

AVOIDING DISASTER



ZODB

Why not?

It's object-oriented

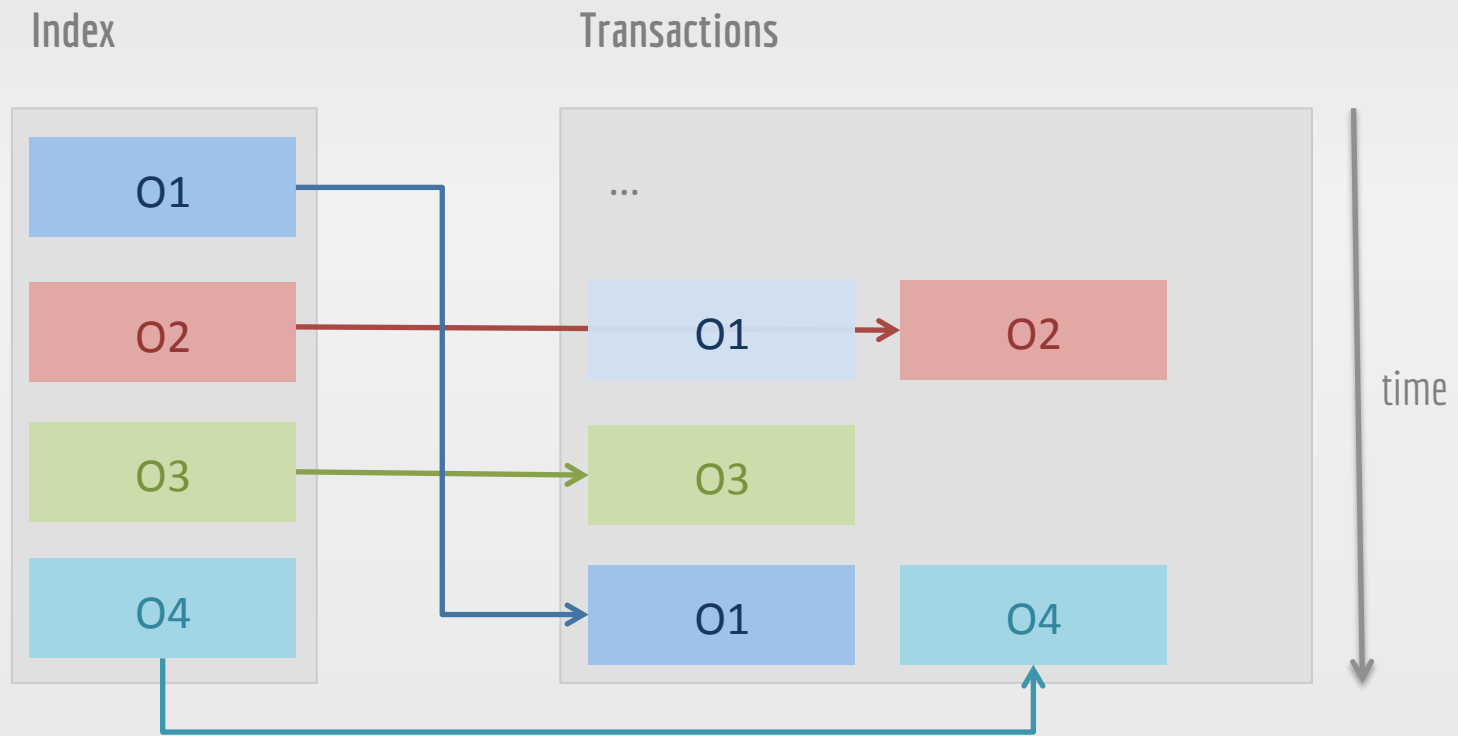
It's been around for a long time

We know how to use it

We are already using it for some of these things

ZODB

An object-oriented database



ZODB

The good parts

It's simple

No need for ORM or mapping layers

Tightly integrated with Python

It is ACID – things work as expected

ZODB

The bad parts

No server-side queries

No built-in indexing – application level

Data recovery is slow (latency, unpickling, setting object state)

No way to fetch more than an object at once (pre-load)

ZODB

Yes, there's more...

Has to be packed regularly

No caching on the server side (besides OS cache)

FileStorage Replication is not Open Source - money

RelStorage could work, but requires migration

It's a niche product

DON'T PANIC!

We're working on it!



NG DB PROJECT

The Next Generation DB for Indico

<http://indico-software.org/wiki/Dev/FutureDB>

Aims to find DB infrastructure that can support growth

6 month initial phase (tech preview/boilerplate – end 2013)

Tech Survey, Evaluation, Prototyping...

NG DB PROJECT

The criteria

Availability (OSS)

Scalability/Replication

Ease of use/development

Transactions/Consistency

Community/Momentum

Costs

THE CONTESTANTS

The NoSQL crowd

Key-value stores

Riak, Redis, Voldemort

Document-oriented

MongoDB, CouchDB

Column-oriented

Cassandra, Hbase

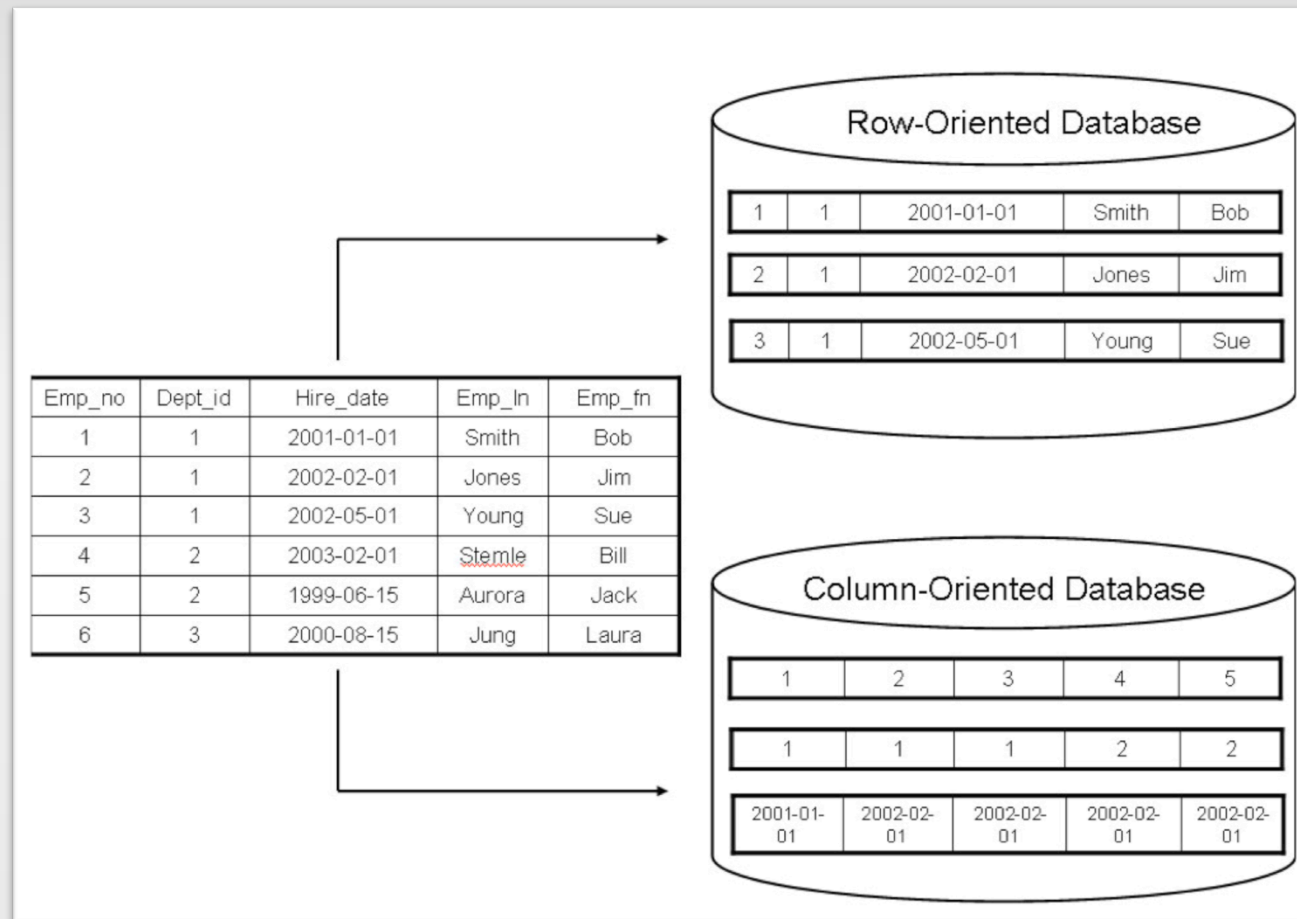
Neo4J

KEY-VALUE

Just a mapping structure

user1	Pedro Ferreira
user2	Alberto Resco
user3	{name: {first: "Jose Benito", last: "Gonzalez"}}

COLUMN-ORIENTED



DOCUMENT-ORIENTED

Closer to the OO philosophy

user1	name	Pedro Ferreira
	city	Geneva
user2	name	Alberto Resco
	hobbies	Running
		Cycling

THE CONTESTANTS

The usual (relational) suspects

~~MySQL~~

MariaDB

Drizzle

Percona

PostgreSQL

RELATIONAL VS. NOSQL

A *very* coarse comparison

	Relational	NoSQL
ACID	Yes	Not usually
Philosophy	General-purpose Table-oriented	Problem-specific Normally closer to OO
Maturity	Decades	Pretty recent
Consistency	Normally strong	Normally eventual

RELATIONAL VS. NOSQL

The problems

Relational	NoSQL
Requires ORM Different philosophy	Lack of transactions Eventual consistency Too simplistic

TYPICAL QUERY

«All events in a user's favourite categories»

Relational

Simple join between two tables

MongoDB

Either replicate data or use DB refs (slow!)

HYBRID APPROACH

The best of two worlds?

ZODB - excellent storage for business objects

SELECT style queries...

ZODB as primary storage?

Already kind of doing that (Redis)

Need for transition period



HYBRID APPROACH

No such thing as a free lunch...

Keeping data consistent

Multiple DB calls per request

Yet another thing to install

CONCLUSION

Research project

Still a lot of ground to cover

Hard to evaluate the hype

Hybrid could be a good option



PEDRO FERREIRA

[@pferreir](http://github.com/pferreir)