Definicja

Not Only SQL

Baza danych NoSQL to program zapewniający szybki dostęp do danych różniący się w jakiś sposób od stadardowych baz RDBMS.

Baza NoSQL to szereg różnych rozwiązań nazwanych jednym określeniem.

NoSQL często rezygnuje z pośrednictwa SQL przy manipulacji danymi i zapewnia alternatywny sposób.

Mikołaj Sochacki

Kto korzysta?

MongoDB: Foursquare, Disney, SourceForge
CauchDB: Ubuntu One, BBC, CERN
Hbase: Adobe, Facebook, Twitter, Yahoo
Cassandra: Rackspace, Twitter, Reddit
Redis: Flickr, GitHub, StackOverflow
Riak: Mozilla Fundation, AOL

Wydajne sieciowe aplikacje z olbrzymią ilością danych

Dlaczego nie SQL?

- Skalowalność
- Replikacja danych
- Rozporszenie
- Ekstremalna wydajność
- Olbrzymia ilość danych
- Nieustruktualizowane dane
- Dla ORM SQL jest tylko obciążeniem

Mikołaj Sochacki

Not Only SQL — Typy baz NoSql

- Klucz–wartość – Redis
- Kolumnowe – Cassandra, HBase (Big Table)
- Objektowe – Db4o, Versant
- Dokumentowe – MongoDB, CrouchDB
- Grafowe – Neo4J
- XML - Apache Xindice, MarkLogic Server

Mikołaj Sochacki

NotOnlySQL redis

- prosta baza typu klucz – wartość
- klucz string (odzielanie ":" tylko konwecja)
- wartość: string, zbiory, listy, hashmap
- licencja BSD
- redis-cli – wiersz poleceń
- prosta i wydajna replikacja
- dobra dla wielu prostych danych
- bardzo wydajna, wyszukiwanie O(log(N) + M)
- cała baza działa tylko w pamięci

Mikołaj Sochacki

Redis: Api - przyklad

```
import java.util._
import redis.clients.jedis._
object RedisExample {
def main(args:Array[String]) {
val localhost_pool = new JedisPool(new JedisPoolConfig(),
“localhost”)
val  localhost_redis = localhost_pool.getResource
try {
localhost_redis.set(“user:1:name”, “Miki”)
localhost_redis.zadd(“user:1:phone”, 1,  “0700000100”)
localhost_redis.zadd(“user:1:phone”,  3, “0599888444”)
val  phones = localhost_redis.zrange(“user:1:phone”, 0, -1)
println(“User 1 phones: “ + phones)
localhost_redis.set(“user:2:name”, “Cyryl”)
localhost_redis.lpush(“user:2:phone”, “0599999555”)
localhost_redis.lpush(“user:2:phone”, “0456043235”)
localhost_redis.rpush(“user:2:phone”, “0600700800”)
val phone2 = localhost_redis.lrange(“user:2:phone”, 0, -1)
println(“User 2 phones: “ + phone2)
} finally { localhost_pool.returnResource(localhost_redis)
  localhost_pool.destroy(); }
```

Mikołaj Sochacki

Not Only SQL

mongoDB

- oparta o JSON (zapis w BSON)
- napisana w C++
- GridSF – przechowywanie plików o dowolnych rozmiarach
- replikacja
- duża wydajność
- Query Expression Objects – wiersz poleceń opraty o JavaScript

Mikołaj Sochacki

NotOnlySQL MongoDB: Api - przyklad

```
import net.liftweb._
import mongodb._
import org.bson.types.ObjectId
import java.util.Date

case class Address(street: String, city: String)
case class Child(name: String, age: Int, birth: Option[Date])
object Person extends MongoDocumentMeta[Person] {
  override def collectionName = "mypersons"
  override def formats = super.formats + new
ObjectIdSerializer + new DateSerializer }
case class Person(_id: ObjectId, name: String, age: Int,
address: Address, children: List[Child])
  extends  MongoDocument[Person] {
  def meta = Person }

val p = Person(ObjectId.get,"Joe",27,Address("Bulevard",
"Helsinki"),List(Child("Mary", 5, Some(date("2004-09-04"))),
Child("Mazy", 3, None)))
    p.save
```

Mikołaj Sochacki

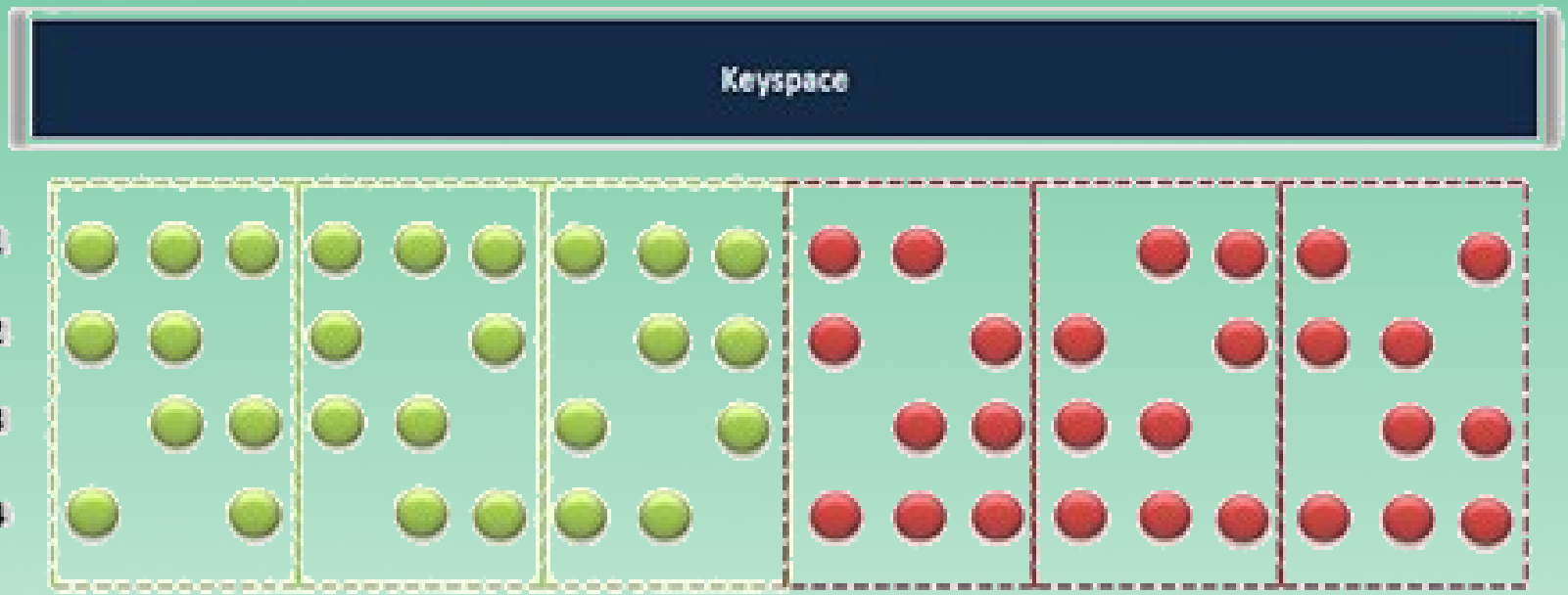
NoSQL Cassandra

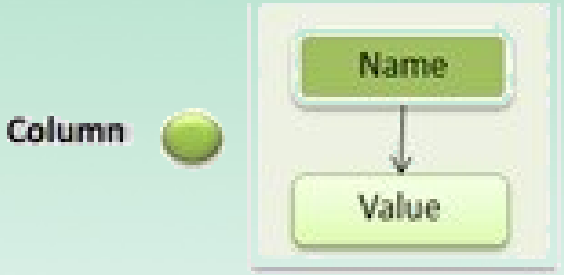
- zainspirowana przez Google Big Table
- klucz – wartość powiązane z column family
- jeden proces na węźle (inaczej niż Hbase)
- wysoka wydajność
- replikacja zainspirowana Amazon Dynamo
- korzysta z Apache Hadoop
- Cassandra Query Language – zamiast SQL

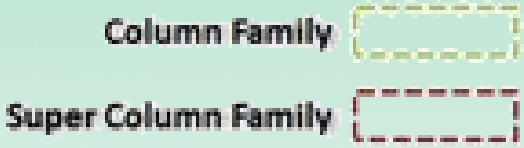
Mikołaj Sochacki

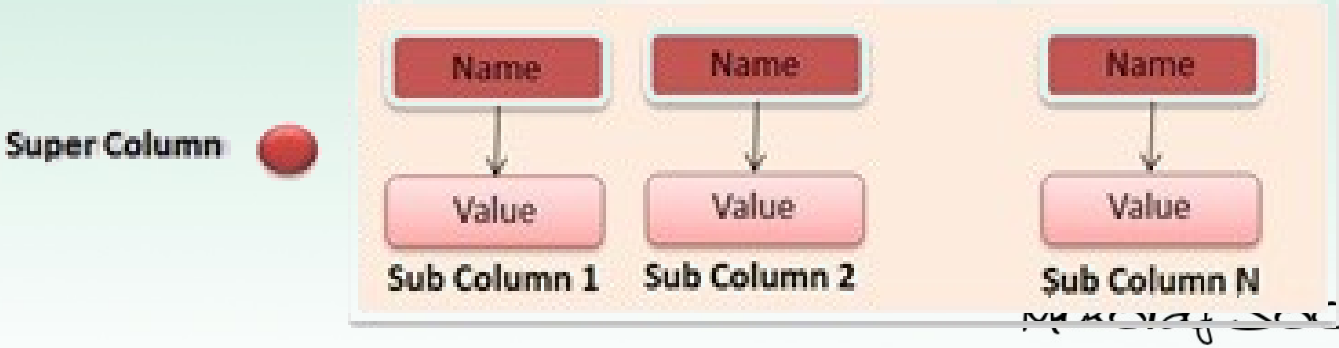
NoSQL

Cassandra - schemat

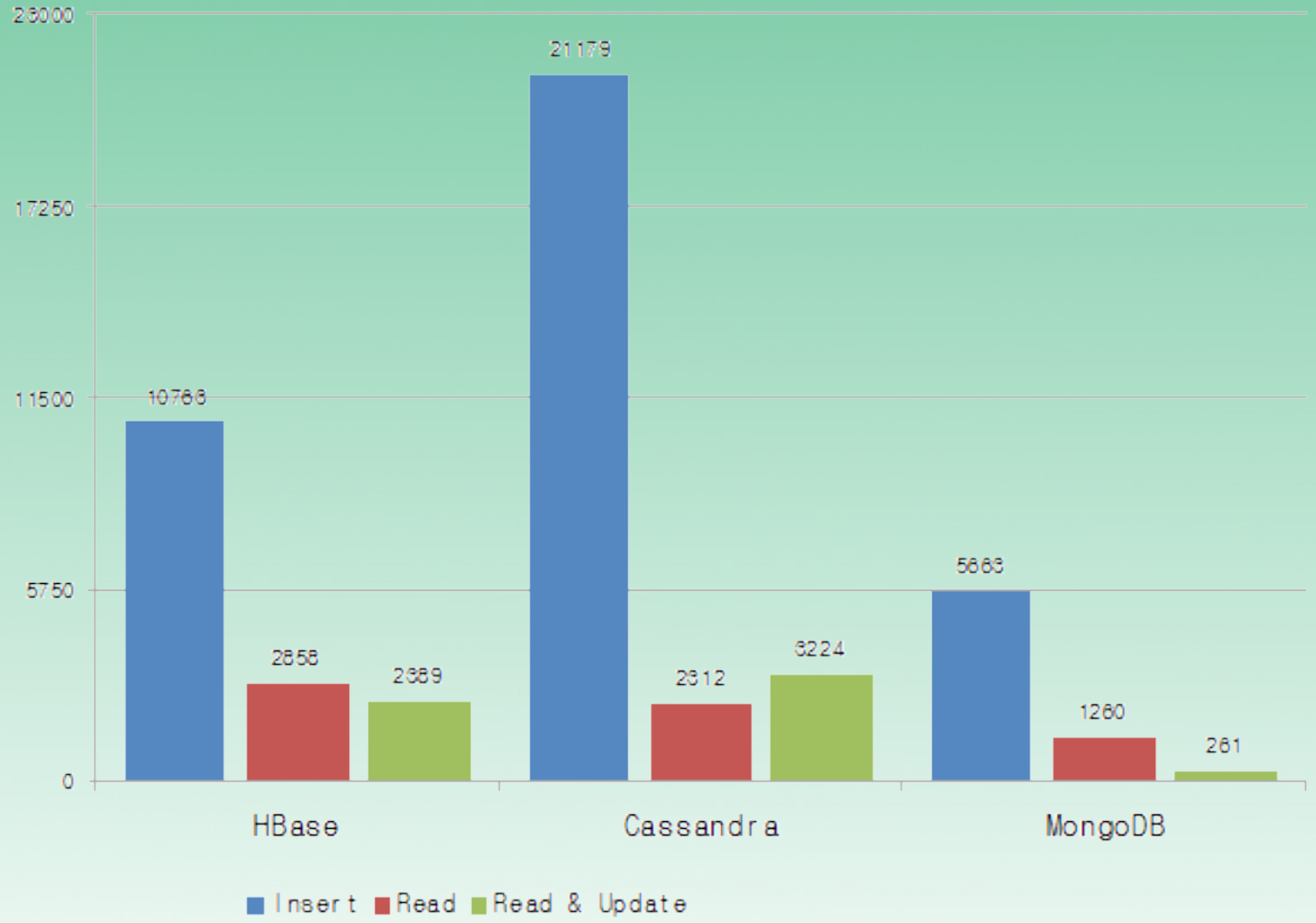








Not Only SQL Wydajnosc - porownanie

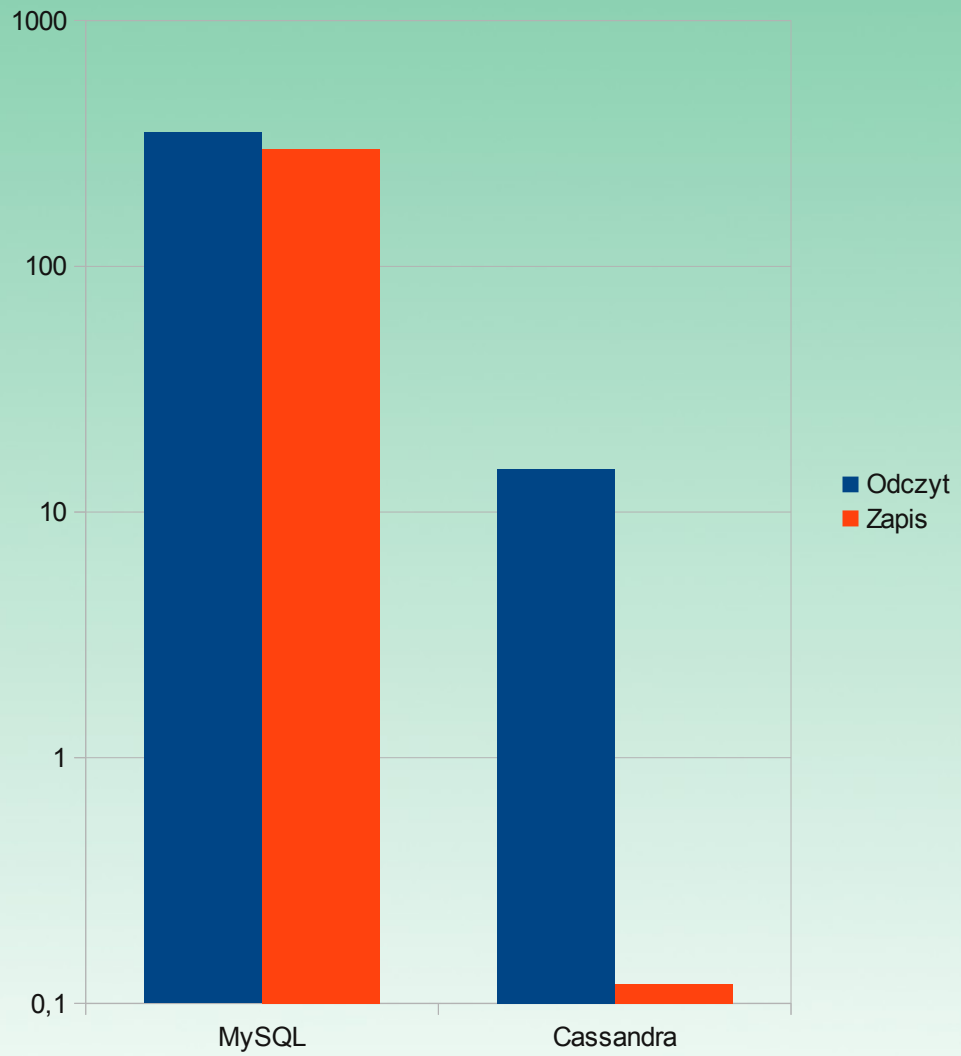


Mikołaj Sochacki

Porownanie MySQL

MySQL –> 50GB
Odczyt: ~350ms
Zapis: ~300ms

Cassandra –> 50GB
Odczyt: ~15ms
Zapis: ~0,12ms



Mikołaj Sochacki

Cassandra: Api - przyklad

CQL – cassandra query language:

```
CREATE KEYSPACE test with strategy_class='SimpleStrategy'
and strategy_options:replication_factor=1;
```

Api w większości popularnych języków:

```
val tr:TTransport = new TSocket(HOST, PORT)
val tf = new TFramedTransport(tr)
val proto:TProtocol = new TBinaryProtocol(tf)
val client = new Cassandra.Client(proto)
tf.open()
val id = “1“.toBytes
val CL = ConsistencyLevel.ONE
val clock = new Clock(System.currentTimeMillis())
client.insert(id, cp, new Column("name".getBytes(UTF8),
"George Clinton".getBytes(), clock), CL)
val col = client.get(id, “name“.getBytes(UTF8), CL).getColumn
println(“Name: “ + col.name + “ Value: “ + col.value +
        “ Timestamp: “ + col.clock.timestamp)
```

Not Only SQL db4o

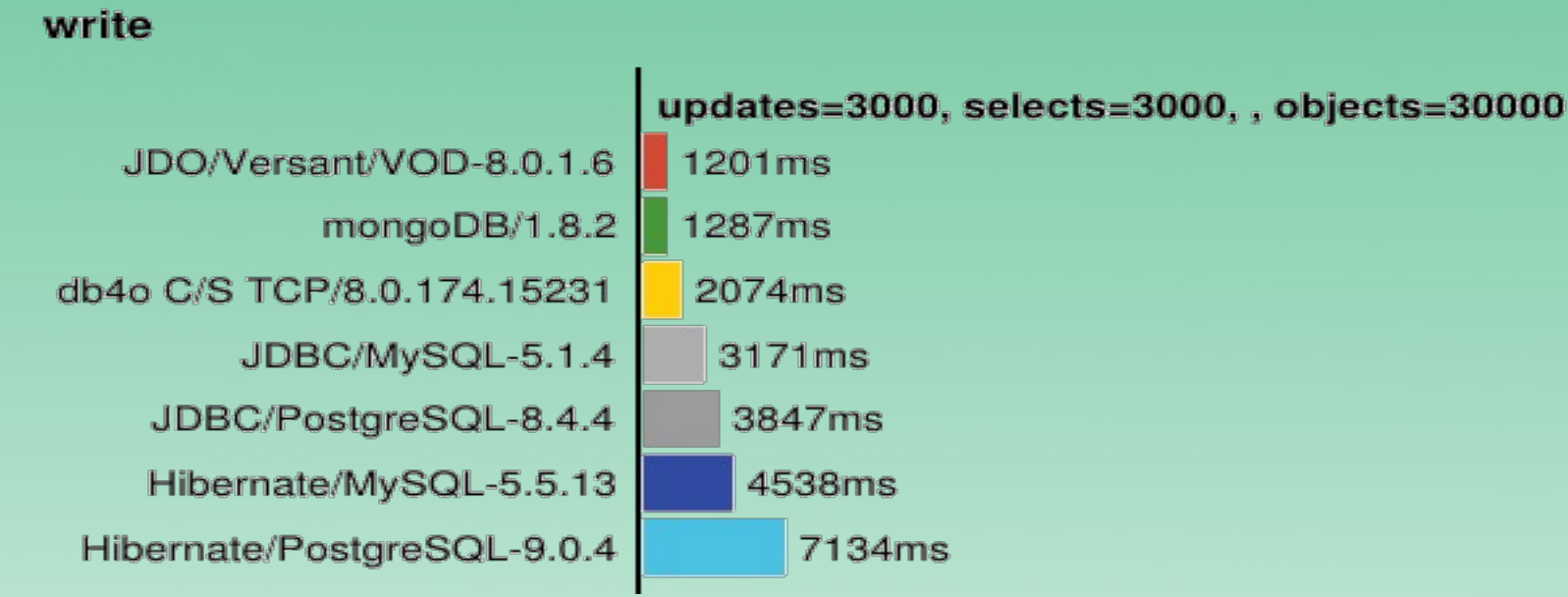
- licencja GPL lub komercyjna
- tylko Java i .NET
- Native Query
- zapisywanie każdego obiektu
- cała baza w pliku
- wersja wbudowana – plik jar
- wersja client – serwer
- szybki odczyt złożonych danych

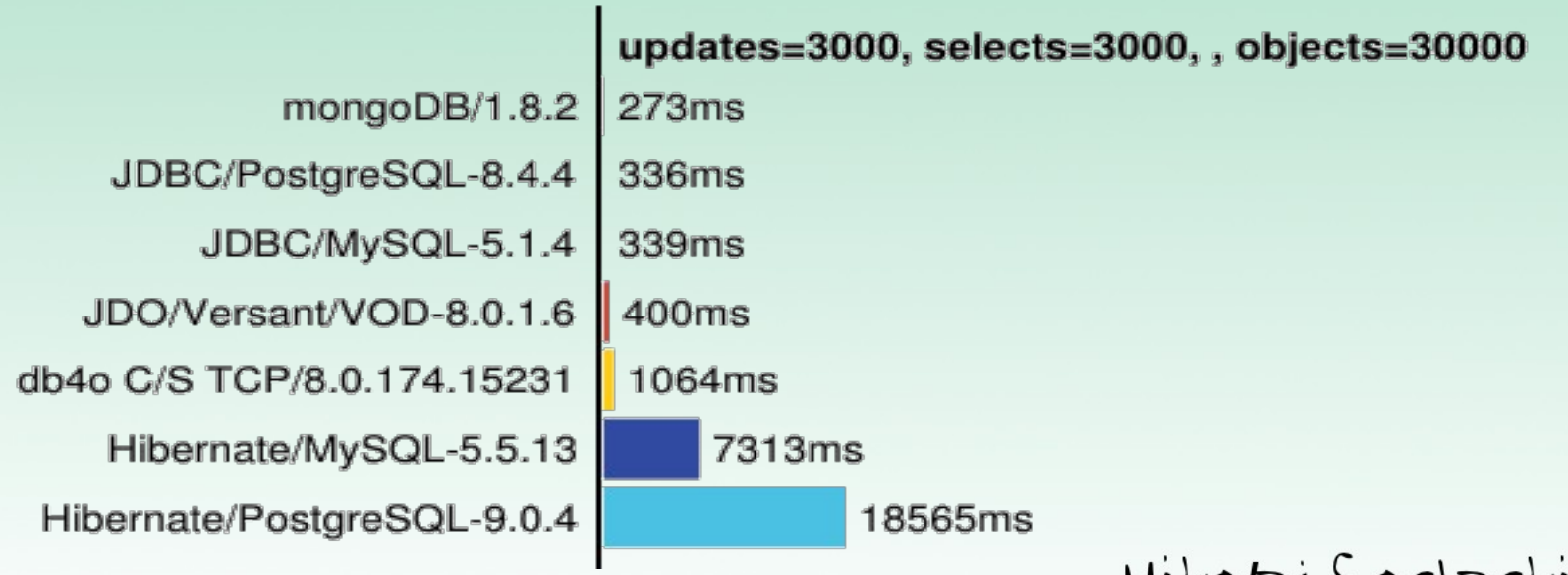
Mikołaj Sochacki

NOSQL

DB4O - WYDAJNOSC

Płaskie obiekty





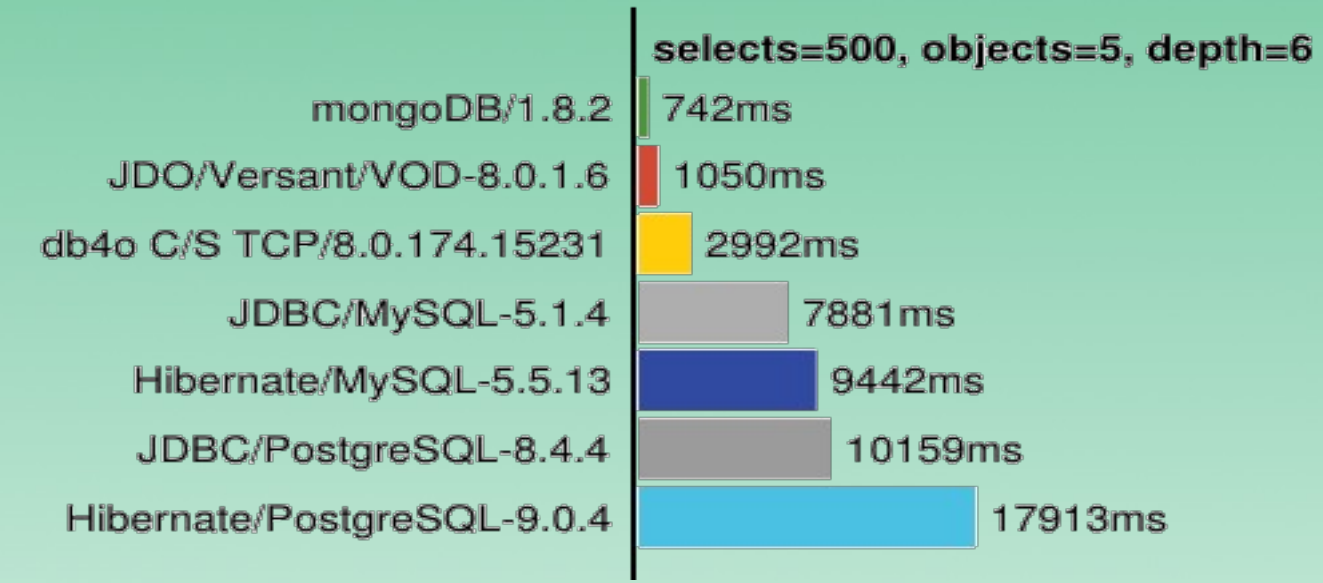
Mikołaj Sochacki

NoSQL

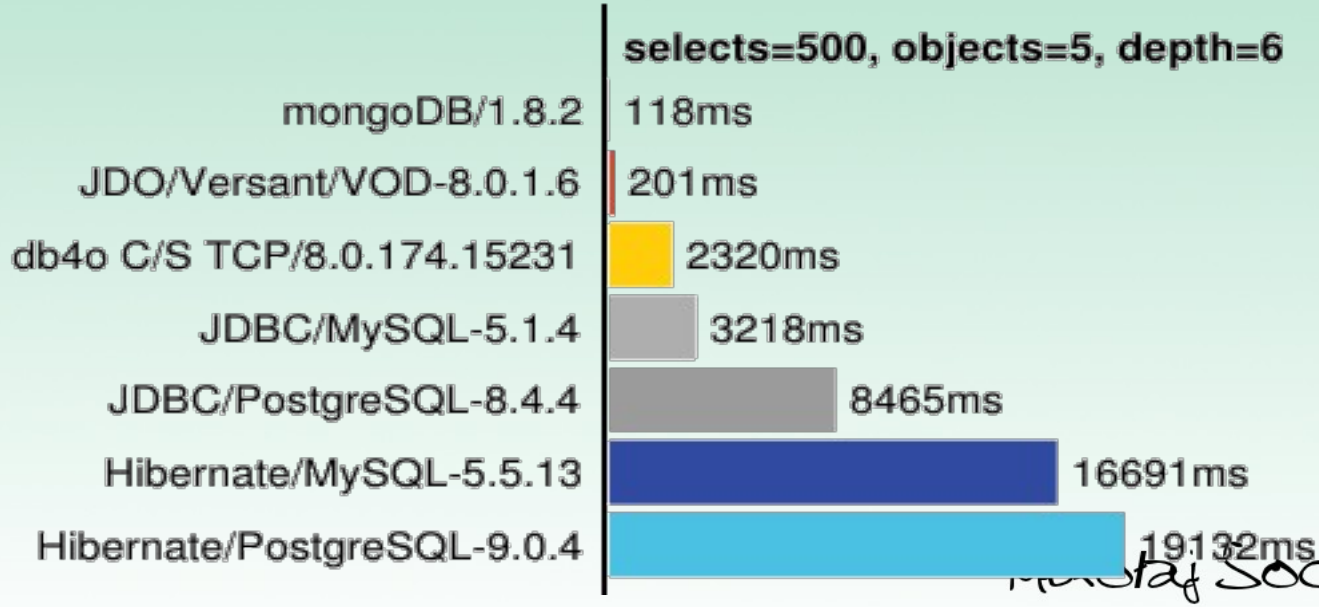
DB4O - WYDAJNOSC

Obiekty złożone

write



query



Mikołaj Sochacki

NotOnlySQL DB4O: API - PRZYKLAD

```
import com.db4o._
import com.db4o.query.Predicate

case class Pupil(nr:Int, lastName:String, firstName:String)

object Main {
  def main(argc:Array[String]){
    val db = Db4o.openFile("database.db")
    try {
      db.store(Pupil(1,"Kowalski","Jan"))
      db.store(Pupil(2,"Kowalski","Piotr"))
      db.store(Pupil(3,"Nowak","Anna"))
      val r = db.queryByExample(Pupil(0,"Kowalski",null))
      println("Query By Example")
      while (r.hasNext) println(r.next)
      val pupils = db.query(new Predicate[Pupil](){
              def `match`(p:Pupil) = { p.nr == 1}
            })
      println("Native Query")
      while (pupils.hasNext) println(pupils.next)
    }
    finally { db.close() }
    println("koniec")
}}
```

Mikołaj Sochacki

Podsumowanie

Kiedy warto przemyśleć użycie NoSQL:

- duża ilość danych
- konieczna replikacja
- potrzebna duża wydajność
- nie wszystkie dane nadają się do „tabelek“
- dane są głębokie (dużo złączeń SQL)
- do większości prostych stron nadaje się MongoDB

Nie warto:

- mała aplikacja
- system bankowy

Dziękuję za uwagę!

Pytania?