So You Think You Know JPA?

Leverage JPA for NoSQL, &
Big Complex Data Applications

Dirk Bartels, VP Strategy
Matthew Barker, Director Prof. Services
Versant Corporation

VERSANT

Agenda

- Big Complex Data challenges
- When JPA OR Mappers fail
- Live benchmarks
- The results are in
- Conclusions and Q&A

Big Complex Data Challenges

Business depends on intelligent data

- Data stored, transacted, transformed, analyzed

Modern world data challenges

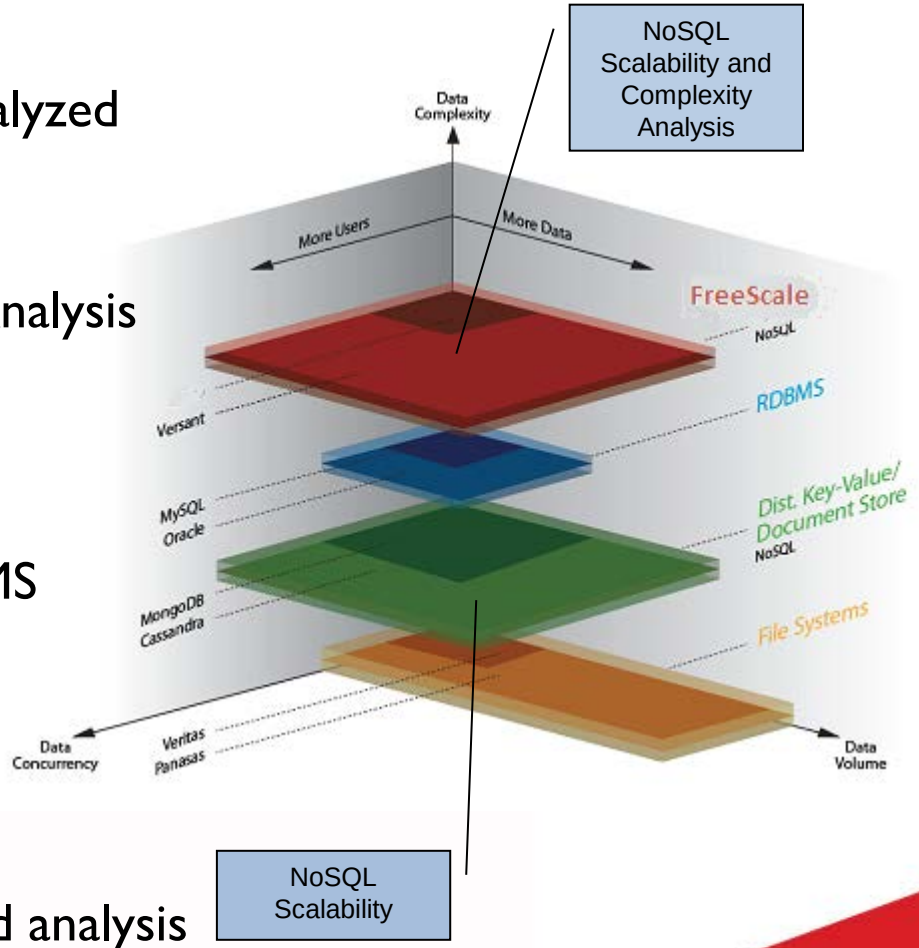
- Data is Big :: Complexity, Concurrency, Analysis

Big Data technologies

- Data too big to work with a RDBMS
- Data too complex to work with a RDBMS

NoSQL technologies

- Partition tolerant
- easy to scale out
- Enterprise "linked" data management and analysis

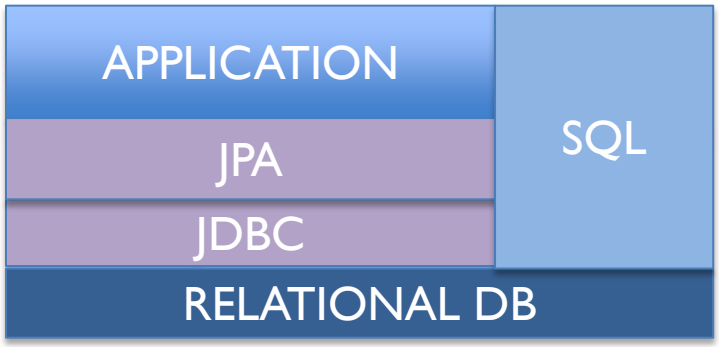


VERSANT

The "de-facto" Java Persistence

JPA OR mapping & RDBMS

- Rapid development
- Time-to-Market
- Managing objects
- Works for low to medium data size/complexity



BUT: JPA ORM and Big Data?

RDBMS performance/scalability degrades when

- Scaling "UP" → larger data sets
- Scale "OUT" → more connections, distributed data
- Scale "UP" and ""OUT"

Question: "Where lies the problem here:?"

Is this a JPA problem

OR

Is this a database backend problem?

Using JPA ORM with Big Data*

*Big Data App development even with an ORM mapping tool gets difficult

Programming	Issues
Deep class structures	Normalization versus efficiency
Mapping of Many-to-M any relationships	Necessitates "intersection tables", more on this later
Social networking (see previous)	Difficulty modeling hyper-graphs
Operations	**Issues**
Indices for JOINS, mapping subclasses, etc.	Database size explosion
Server CPU usage explosion (e.g. JOINS)	Total Cost of Ownership (TCO)

Live Comparison Benchmark

VERSANT JPA & JPA ORM

It's Never Easy to Introduce New Advanced Technologies

Benchmark Technical Details

Hardware

- Dell Latitude E6520, Intel i7-2720QM CPU at 2.2 GHZ, Quad Core, 8.0 GB RAM
- Windows 7 Professional Edition, 64-bit

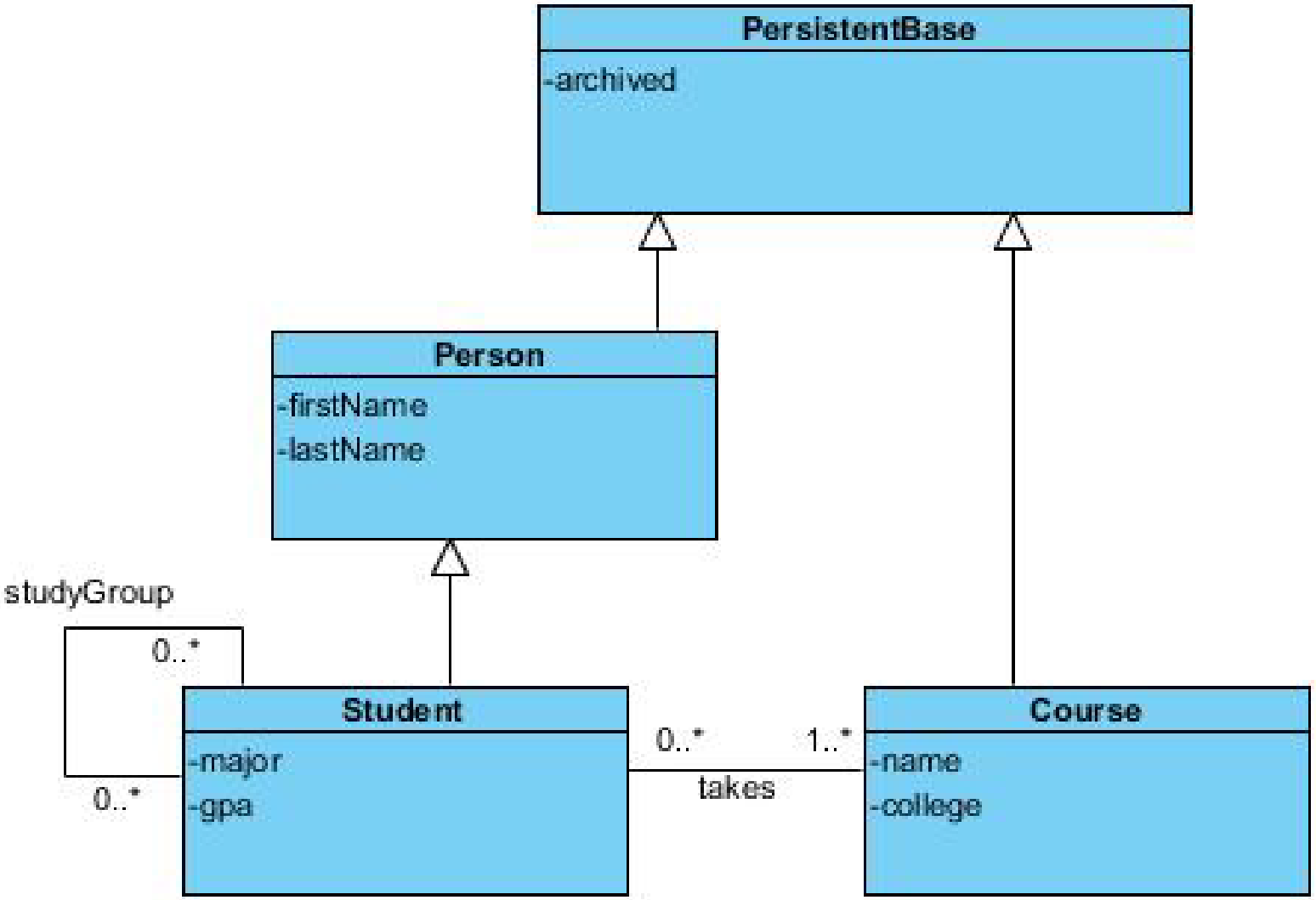
Versant JPA

- Versant JPA Technical Preview 8.0.2.x

JPA ORM

- Hibernate 4.1.0 (best of breed ORM tool)
- MySQL Server 5.1

Benchmark Object Model



Why a Simple Object Model?

A "digestible" model needed for live demo

- Exposes typical ORM issues
- Complex object model will expose these issues further

Many-to-many relationship from Student to Course

- Performs poorly in JPA ORM
- Necessitates "mappedBy" and "intersection table class" *

Many-to-many relationship from Student to Student

- Social networking produces a "hyper-graph" which performs poorly in JPA ORM
- "same entity" many-to-many presents unique issues

* See J. Marques article on "Hibernate Many-To-Many Revisited":
http://josephmarques.wordpress.com/2010/02/22/many-to-many-revisited/

Benchmarks I

Simple CRUD Test

- Create courses and N students, enroll each student in 5-7 courses, add study groups
- Update all students in a chosen college including study groups
- Retrieve all students and their associated courses and study group partners (depth of 2)
- Non-parallel

Scale Up

- Repeat “Simple CRUD Test” several times with larger data sets each time
- Which JPA has linear performance?
- Moderate complexity, non-parallel

Benchmarks II

Scale Out

- In parallel, each thread with it's own connection to db
- Start a write thread and an update thread
- Start N retrieve threads
- Each retrieve thread retrieves courses and associated students including study partners to depth of 2
- Repeat several times using increasing values for N

Scale Up and Out

- Use large number of parallel threads performing CRUD operations using large data sets

JPA Comparison Live Demo

Using Eclipse IDE, we

- See how easy to add "Versant JPA" nature to your existing project
- Quick look at the code, highlighting differences between the two JPA's
- Execute one run for each use case for each JPA, examining performance results
  - After the live demo we will examine complete results with graphs

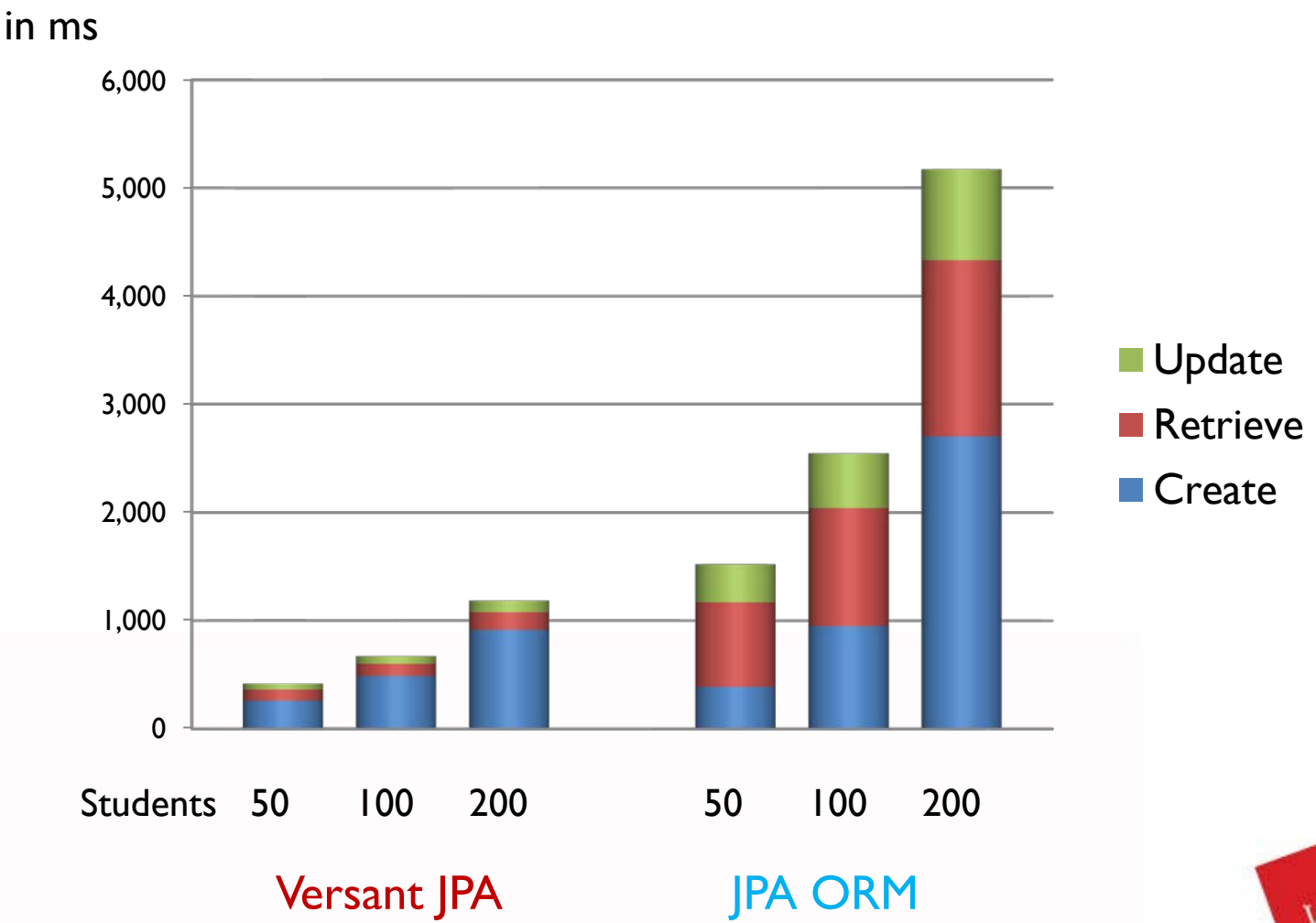
The Benchmark

THE RESULTS ARE IN

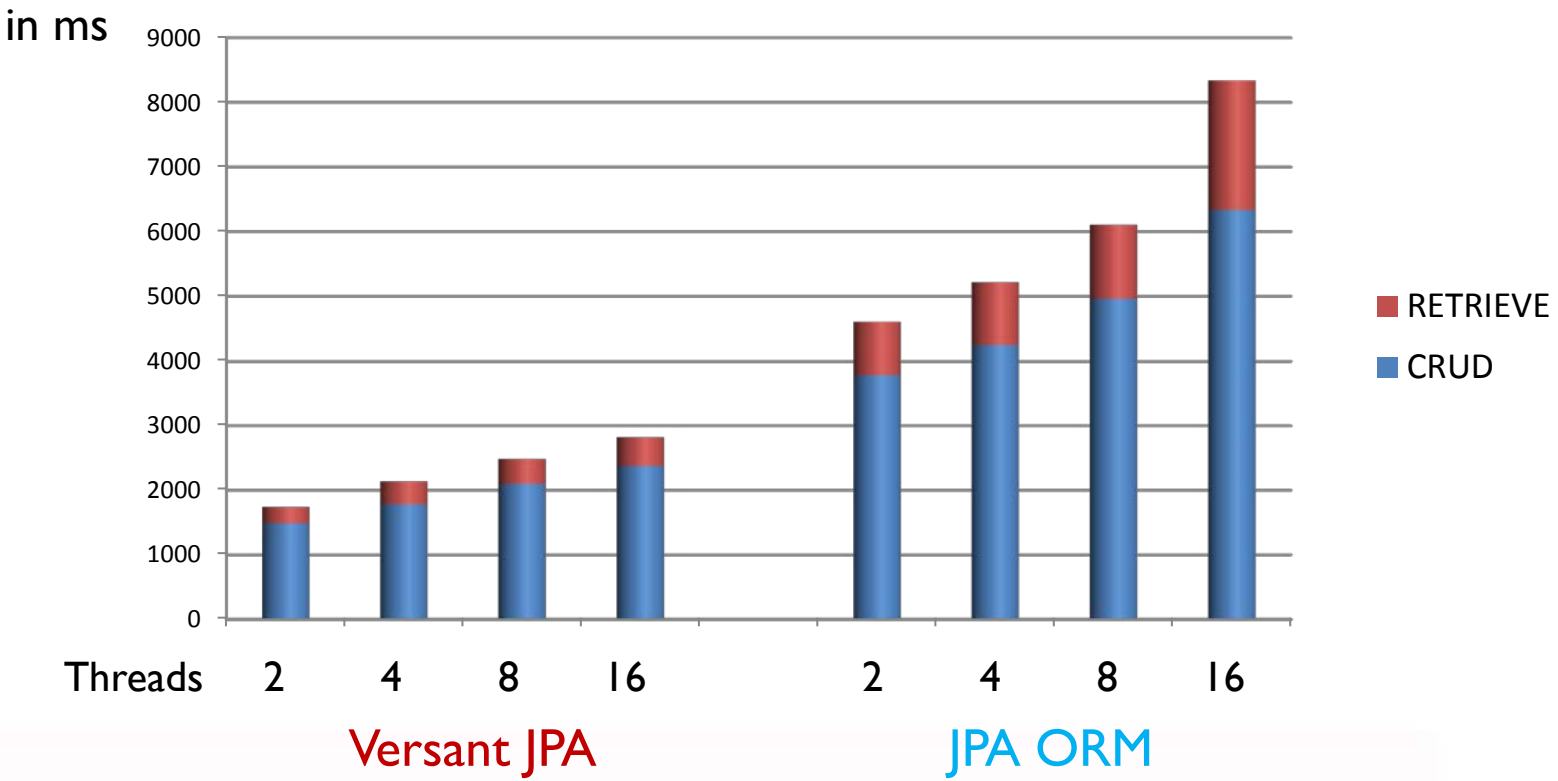
VERSANT

Benchmark 1
Simple CRUD and Scale Up

(create times lowered by 10X for scaling)



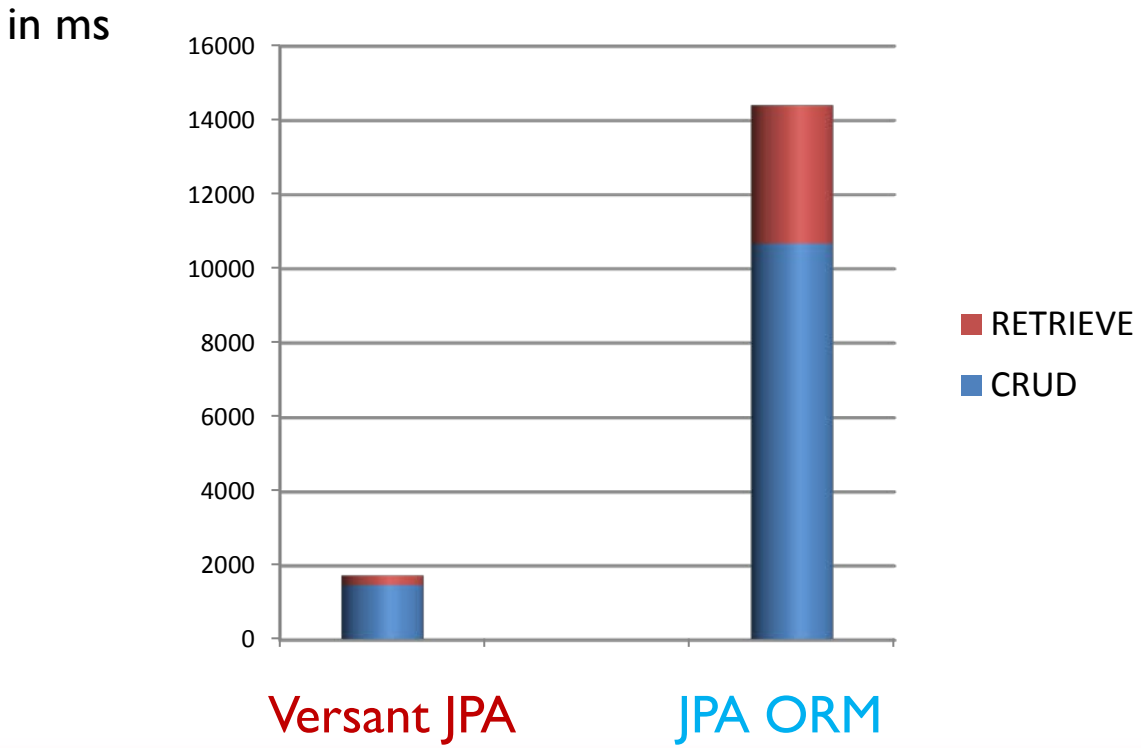
Benchmark II
Scale Out



600 Students, CRUD and Retrieve Test

VERSANT

Benchmark II
Scale Up and Out



1200 Students, 1 Create Thread, 1 Update Thread, 16 Retrieve Threads

Benchmark Conclusions*

* With a Medium Complexity Object Model

SIMPLE CRUD operations

- Versant JPA 4-10X faster than JPA ORM

SCALE UP

- Versant JPA scales up almost linear, JPA ORM does not

SCALE OUT

- Versant JPA scales out (execution times are stable), JPA ORM does not

SCALE UP AND OUT

- Versant JPA >10X faster at large scale up and out

Rapid Development Comparison

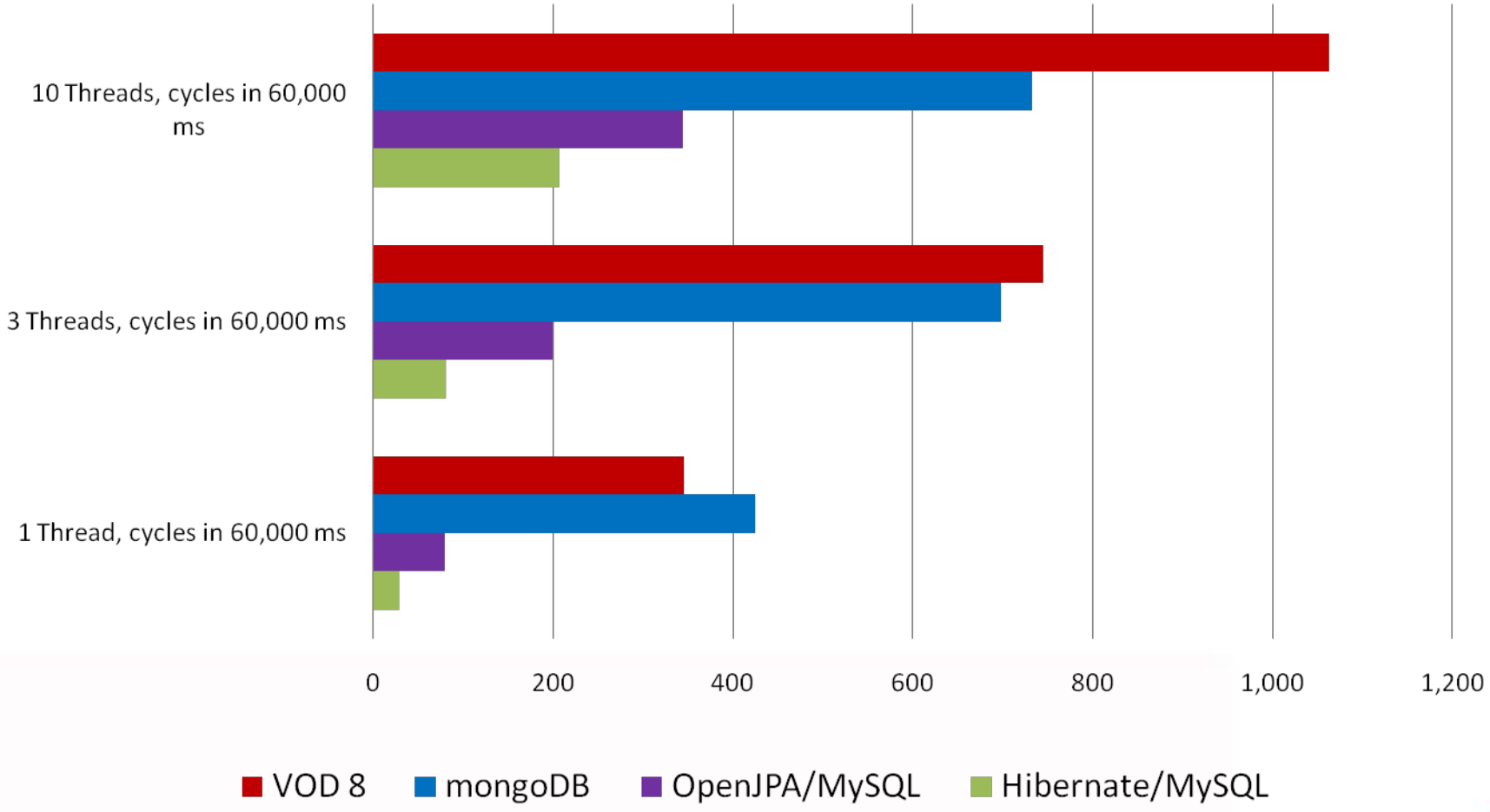
Feature	Versant JPA	JPA ORM
Supports JPA 2.0 Standard	YES	YES
Easy to install/configure	EASY	HARD
Annotations	EASY	MEDIUM
Efficiently implementing inheritance	EASY	HARD
Implementing many-to-many relationships	EASY	MEDIUM
Single vendor solution	YES	no
Connect to distributed databases	EASY	HARD
Supports rapid development cycles	YES	no

Versant JPA vs. Other NoSQL Technologies

Requirements	Versant JPA	Other NoSQL
Persist complex data models	EASY	VERY HARD
Scale out	YES	YES
ACID transactions	YES	no
High availability features	YES	MAYBE
Replication options	YES	MAYBE
Proven	YES, 150k deployments	No track record

Complex Object Graph Workload – Concurrency

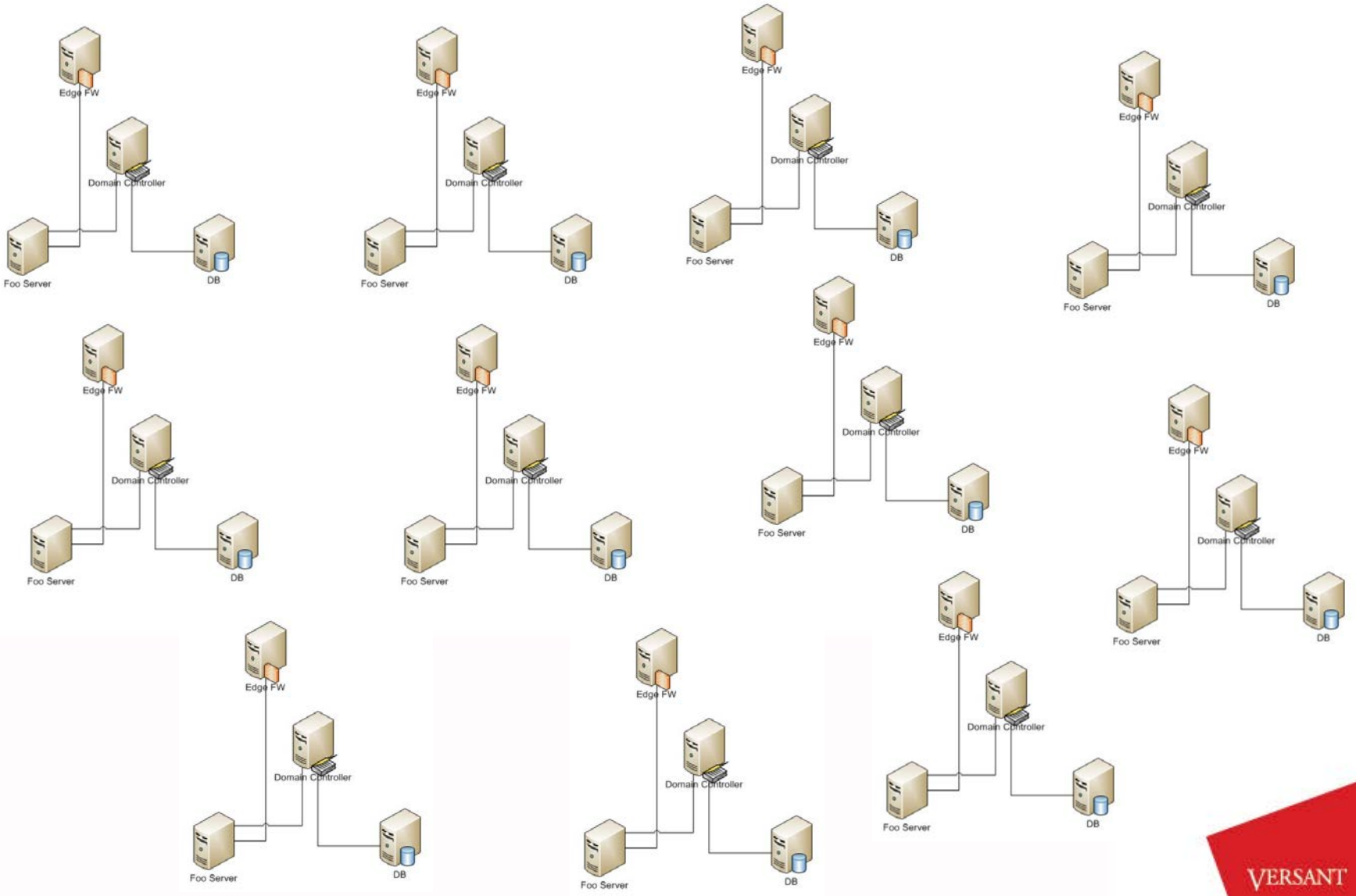
(cycles completed in set time=60,000ms)



Maybe Performance Is Not a Critical Factor?

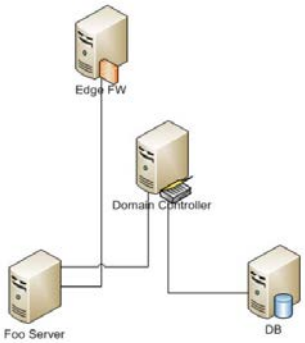
Consider a medium complexity Big Data Application with a 10X performance improvement with Versant JPA over JPA ORM.

JPA ORM Server Footprint



VERSANT

Versant JPA Server Footprint



VERSANT

Versant JPA – More Information

http://community.versant.com

- Free download
- blog articles
- forum entries

jpa@versant.com

- asking questions

VERSANT

Conclusions

Run your own benchmarks – know your workloads

- Understand your domain model
- Understand your concurrency model
- Understand your hardware (e.g., cores, memory, IO)

Build an efficient application model

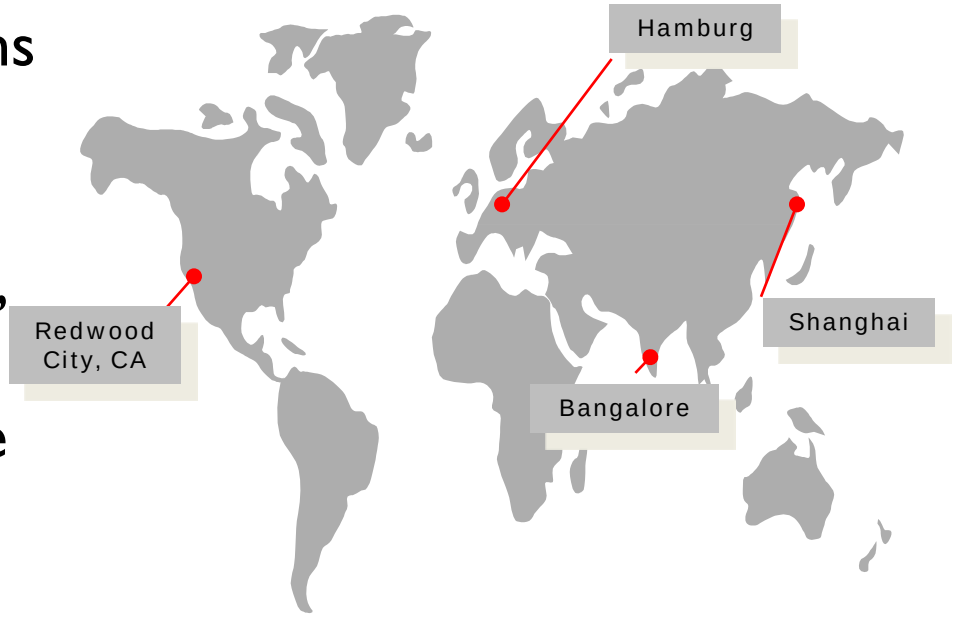
- JOINs are not needed - unless ad-hoc queries are needed
- Consider data distribution and partitioning
- Look for a database that supports your application model

Take care of business, dump a load of code

- Serialization is bad
- Data is only useful when structured and semantically organized
- Mapping is problematic for Big Data (even when it's not an ORM)

About Versant

- Helping innovators to build high performance database applications
- NASDAQ:VSNT since 1996
- Telecommunications, Defense, Financial Services, Transportation, Manufacturing, other.
- Mission critical, high performance databases
- >150k installations world wide



Dirk Bartels, dbartels@versant.com
Matthew Barker, mbarker@versant.com

VERSANT