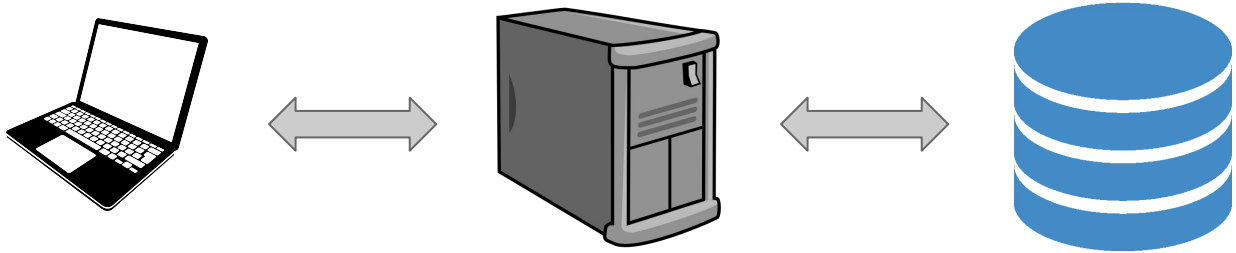
Demystify Database Management Systems (DBMS) in Modern Era

By Shichao

VidarDB
Simplify Your Backend

a little bit about me

- 8 years experience in DBMS development
- Ex-Redshift software engineer in AWS
- Ex-PhD student at UWaterloo in Database
- Top peer-reviewed papers in DBMS
- Read hundreds of DBMS papers
- Internals of data storage systems
- Big fan of PostgreSQL

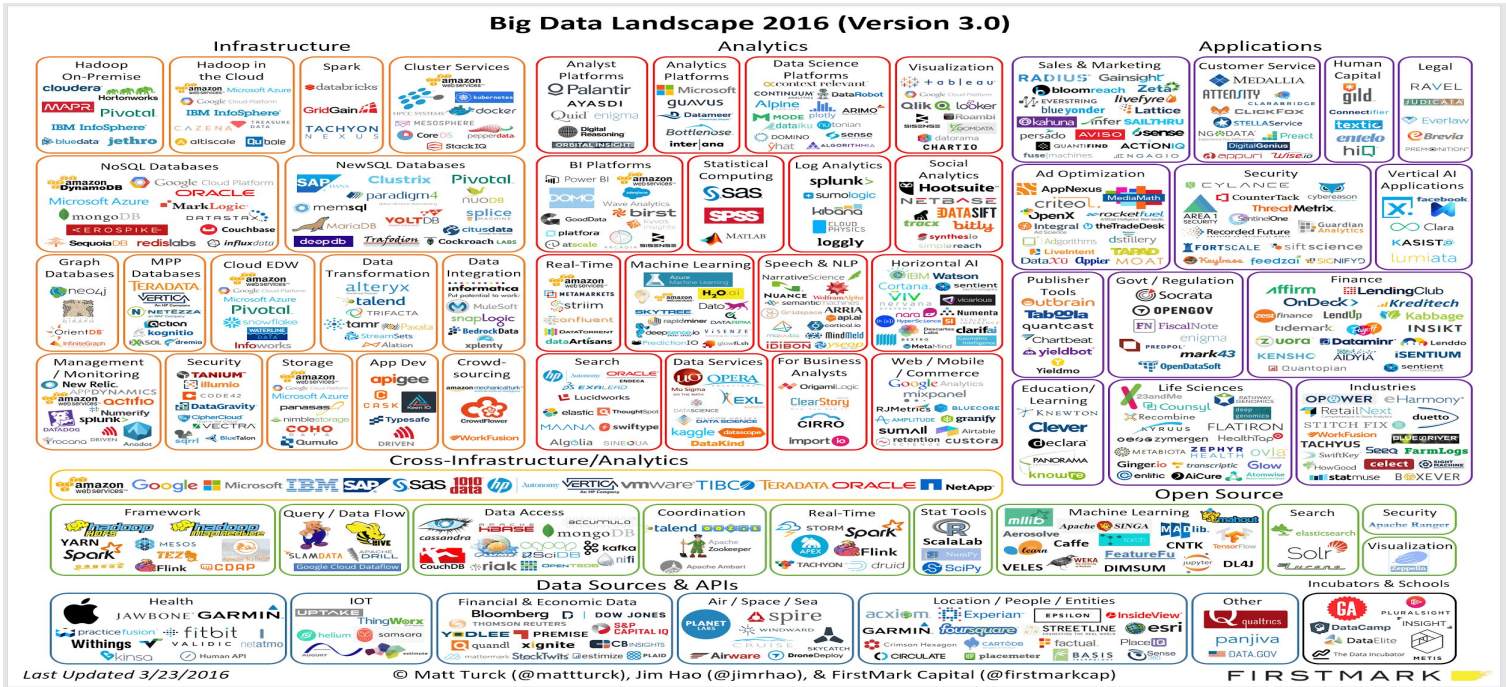


Presentation Layer

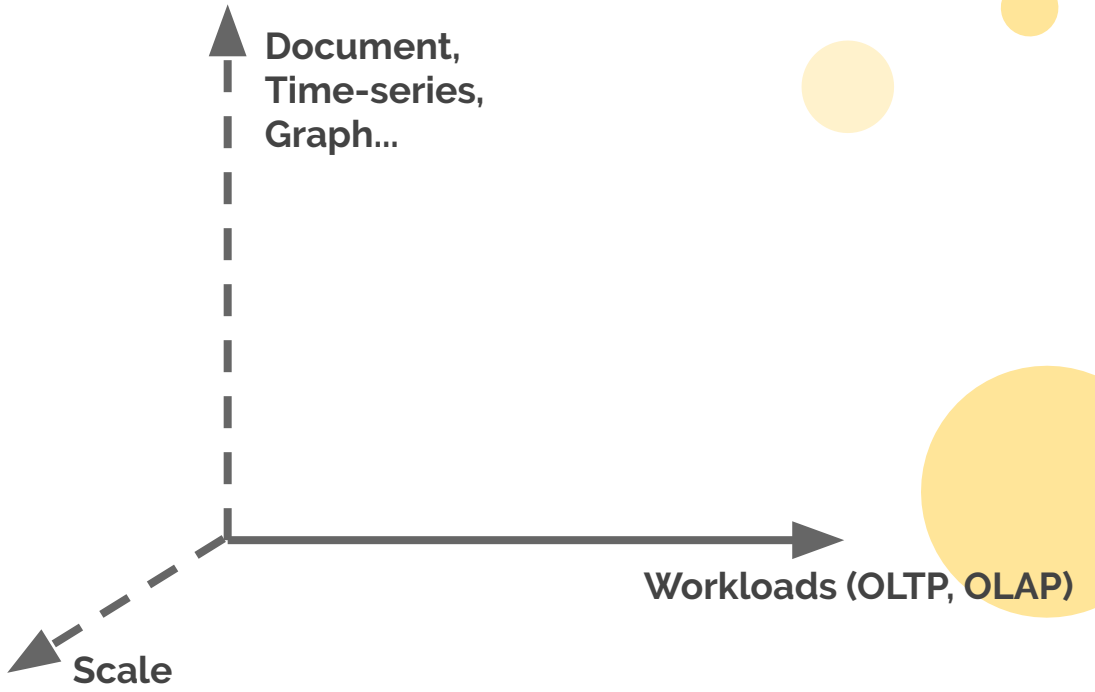
Business Logic Layer

Data Access Layer

Motivation


Big Data Landscape 2016 (Version 3.0)
Infrastructure
Analytics
Applications
Hadoop On-Premise
Hadoop in the Cloud
Spark
Cluster Services
Analyst Platforms
Analytics Platforms
Data Science Platforms
Visualization
Sales & Marketing
Customer Service
Human Capital
Legal
NoSQL Databases
NewSQL Databases
BI Platforms
Statistical Computing
Log Analytics
Social Analytics
Ad Optimization
Security
Vertical AI Applications
Graph Databases
MPP Databases
Cloud EDW
Data Transformation
Data Integration
Real-Time
Machine Learning
Speech & NLP
Horizontal AI
Publisher Tools
Govt / Regulation
Finance
Management / Monitoring
Security
Storage
App Dev
Crowd-sourcing
Search
Data Services
For Business Analysts
Web / Mobile / Commerce
Education / Learning
Life Sciences
Industries
Cross-Infrastructure/Analytics
Open Source
Framework
Query / Data Flow
Data Access
Coordination
Real-Time
Stat Tools
Machine Learning
Search
Security
Visualization
Data Sources & APIs
Health
IOT
Financial & Economic Data
Air / Space / Sea
Location / People / Entities
Other
Incubators & Schools
Last Updated 3/23/2016
© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)
FIRSTMARK

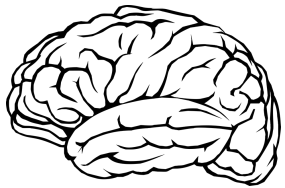
Landscape



Architecture

Query Processing

- Parser
- Planner / Optimizer
- Executor



Storage Engine

Query Processing

Parser: insert into table X values(x1, x2...) ⇒ Put(x1, x2...)

Planner / Optimizer: join A & B & C (A&B&C or A&C&B or B&C&A)

Executor: interact with the storage engine

Storage Engine

Data structure

- Row Store: B-tree
- Column Store
- Hash
- Inverted Index

+ Durability

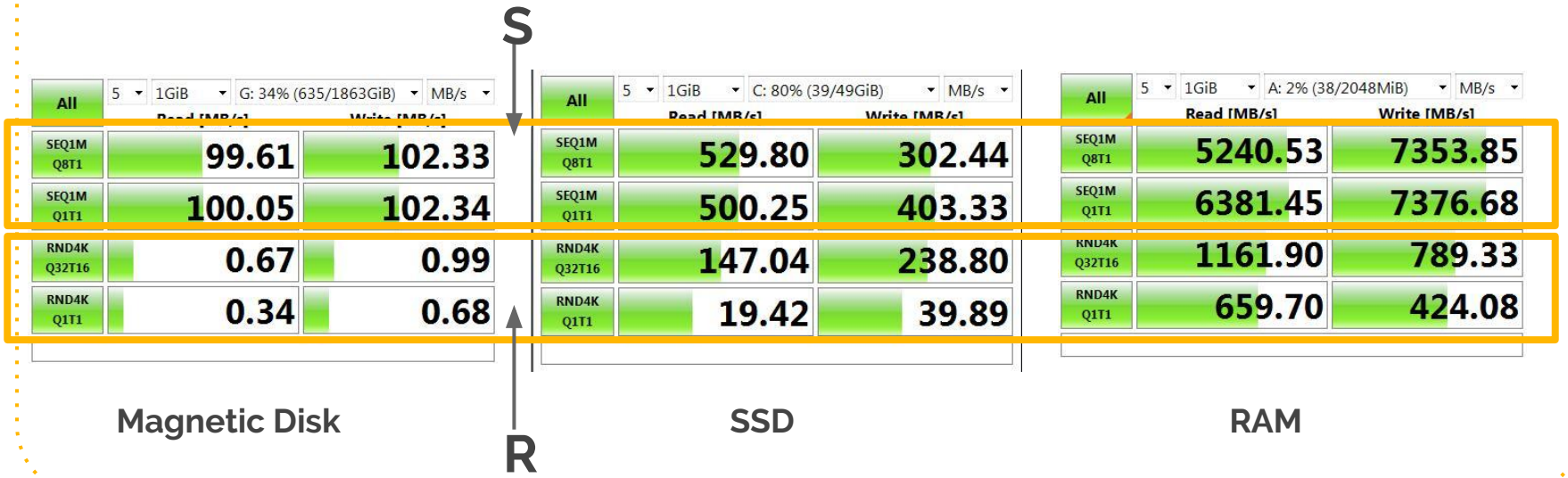
+ Concurrency

+ Log

+ ...

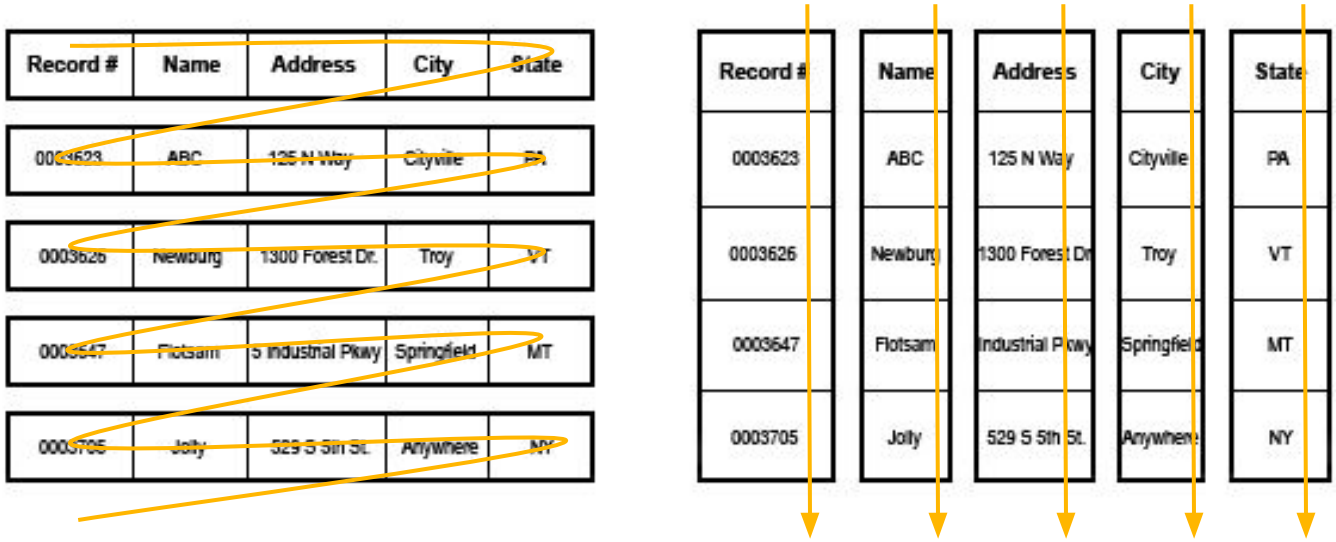
Storage Engine

Sequential vs. Random

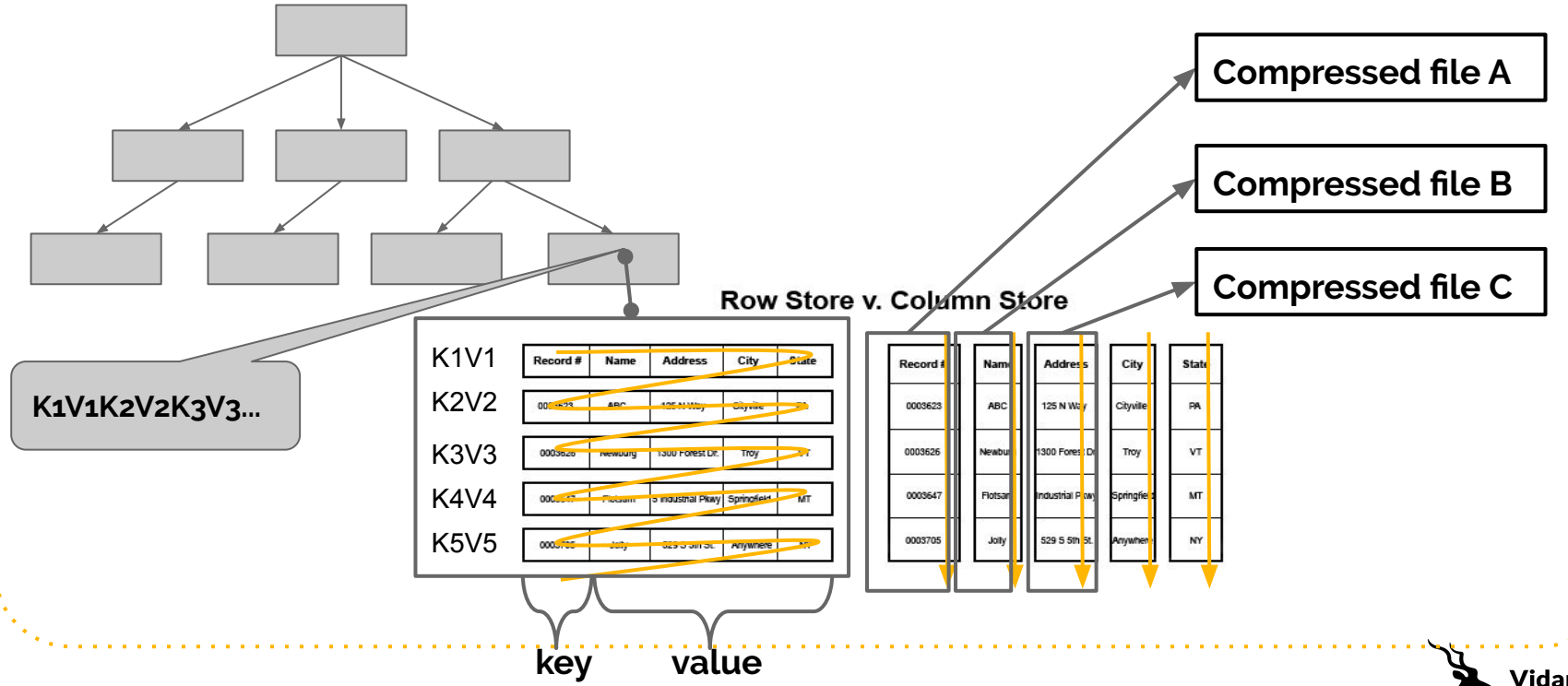


Storage Engine

Row Store v. Column Store



Storage Engine



Storage Engine

OLTP
(Online Transactional Processing)

Select * From X Where id=123;

Operational Database

OLAP
(Online Analytical Processing)

Select avg(salary) From Y;

Data Warehouse

Practice

MapReduce: filesystem

Key-Value Store

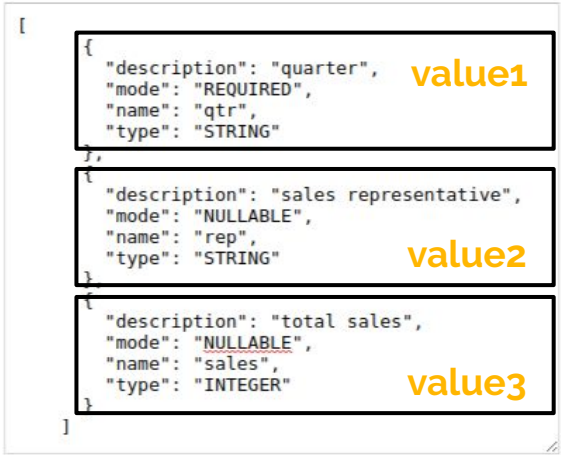
Document

Time-series

Graph: adjacency list

Cache: in-memory hash

Message Queue: in-memory array



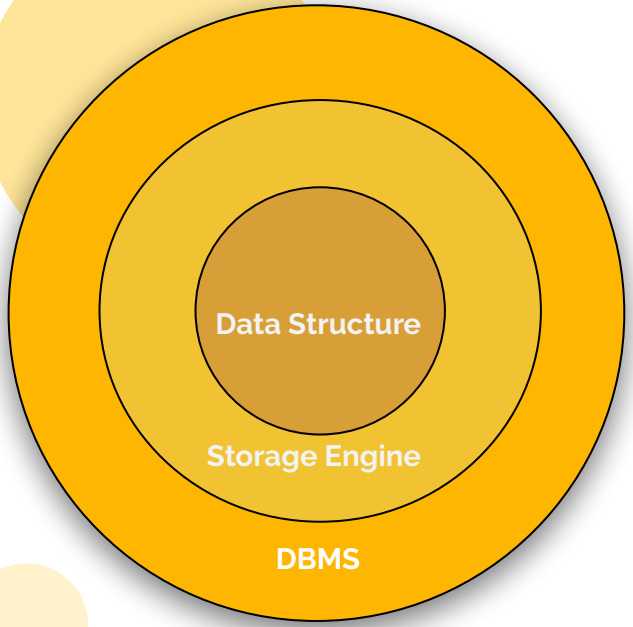
Conclusion

Speed:

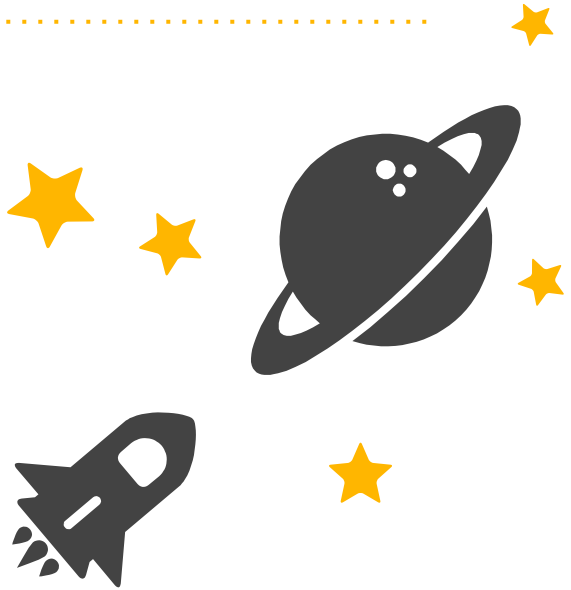
1 millisec = 2 million instrus

Cost:

100 Magnetic Disk = 1 RAM



Thanks!



Shichao@vidardb.com

Versatile Storage Engine:

github.com/vidardb/vidardb

github.com/vidardb/PostgresForeignDataWrapper