VoltDB Overview

DFW BigData Meetup

September 13, 2011

Bobbi Heath

VP of Engineering, VoltDB

VoltDB Background

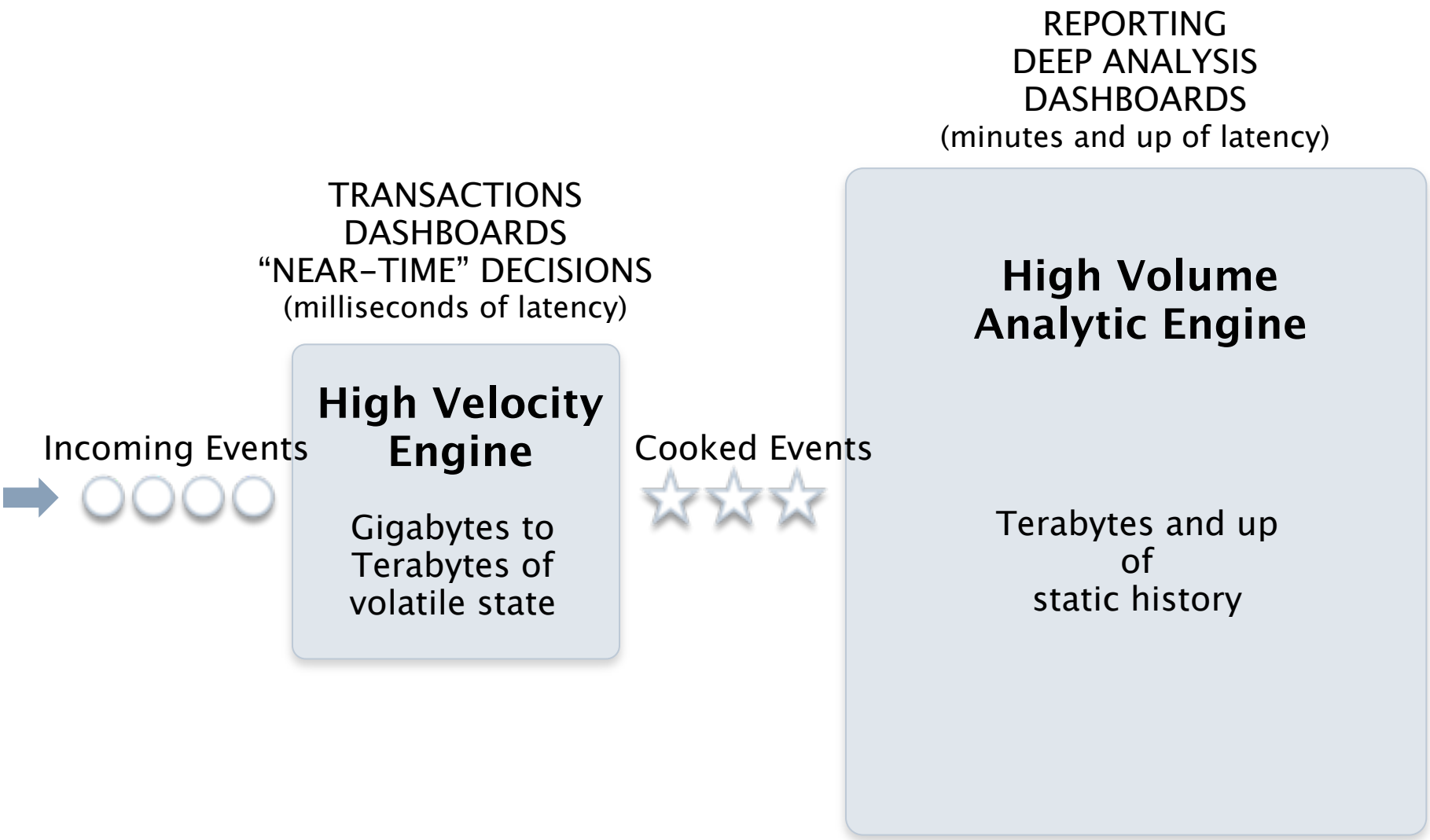
- At Scale everything changes
  - + "One-size-fits-all" does not work
- Database Specialization – transactional workloads
  - + H-store academic prototype led by Mike Stonebraker
  - + Keep functionality of RDBMS
  - + Leverage modern architectures (Memory, CPU, network, etc.),
  - + Design for scale and performance
- Targeted at a new class of data problems where:
  - + Data is arriving at a very fast rate

Big Data Defined – The 3 Vs

- Velocity
  - + Moves at very high rates (think sensor-driven systems)
  - + Valuable in its temporal, high velocity state
- Volume
  - + Fast-moving data creates massive historical archives
  - + Valuable for mining patterns, trends and relationships
- Variety
  - + Structured (logs, business transactions)
  - + Semi-structured and unstructured

James Taylor on Everything Decision Management

About Velocity and Volume



High Velocity Database Requirements

- Handle lots of independent events at a very high frequency
  - + Update state, decisioning, transactions, enrichment, etc…
- Stay up in the face of failures
  - + Make handling failures and recovery as automatic as possible
- Support complex manipulations of state per event
  - + Support a range of real-time (or "near-time") analytics
- Integrate easily with high volume analytic datastores
  - + Raw, enriched or sampled data is migrated to companion stores

What is VoltDB?

- An ultra high-performance RDBMS
- In-memory operation
- Automatic scale-out on commodity servers
- Built-in fault tolerance
- Relational structures, ACID and SQL

A scalable, shared-nothing database architecture

Benchmarked over 3 million operations per

How Fast is VoltDB

Per 8-core node:

> 1 million SQL statements per second

> 50,000 multi-statement procedures per second

> 100,000 simpler procedures per second

How Scalable is VoltDB?

VoltDB is very scalable; it should scale to 120 partitions, 39 servers, and 1.6 million complex transactions per second at over 300 CPU cores.

Baron Schwartz
Chief Performance Architect
Percona

Scalable

SGI Announces Support and Record Benchmarks for VoltDB Database

High Performance SGI® Rackable™ Cluster Delivers Over Three Million Transactions per Second

Speed

Example Case Studies

	Data Source	High-frequency operations	Lower-frequency operations
Financial trade monitoring	Capital markets	Write/index all trades, store tick data	Show consolidated risk across traders
Telco call data record management	Call initiation request	Real-time authorization	Fraud detection/ analysis
Website analytics, fraud detection	Inbound HTTP requests	Visitor logging, analysis, alerting	Traffic pattern analytics
Online gaming micro transactions	Online game	Rank scores: •Defined intervals •Player "bests"	Leaderboard lookups
Digital ad exchange services	Real-time ad trading systems	Match form factor, placement criteria, bid/ask	Report ad performance from exhaust stream
Wireless location-based services	Mobile device location sensor	Location updates, QoS, transactions	Analytics on transactions

Relational, ACID and SQL Matter

- Relational SQL
  - + Data management “heavy lifting”: Joins, data independence, etc…
  - + Analytic queries & Materialized Views
  - + Attempting to do this in application code is very painful
- Data Consistency
  - + Update multiple records with guaranteed consistency
  - + Consistent view of cluster-wide data
  - + Makes developing and iterating complex applications easier
  - + Attempting to do this in application code is VERY painful
- The application may look simple today, but

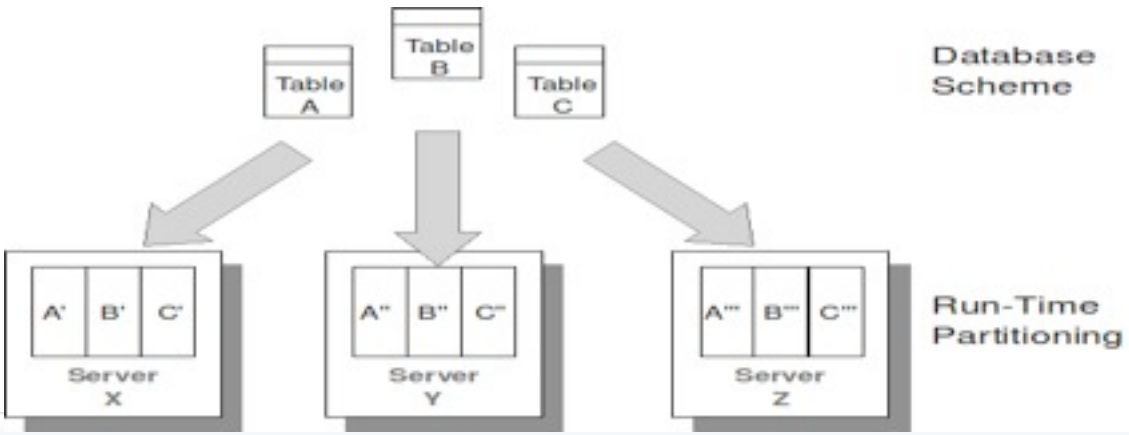
Comparison

	OldSQL	NoSQL	VoltDB
Scale-out growth	w/ Sharding	•	•
Clustered high availability		•	•
Integrated replication		•	•
ACID compliant	Not w/ sharding		•
SQL interface	•		•
Relational	Yes	No	Yes
Cross-partition joins	In app code	In app code	Automatic
Cost at web scale	$$$$	$$	$

Architectural Overview

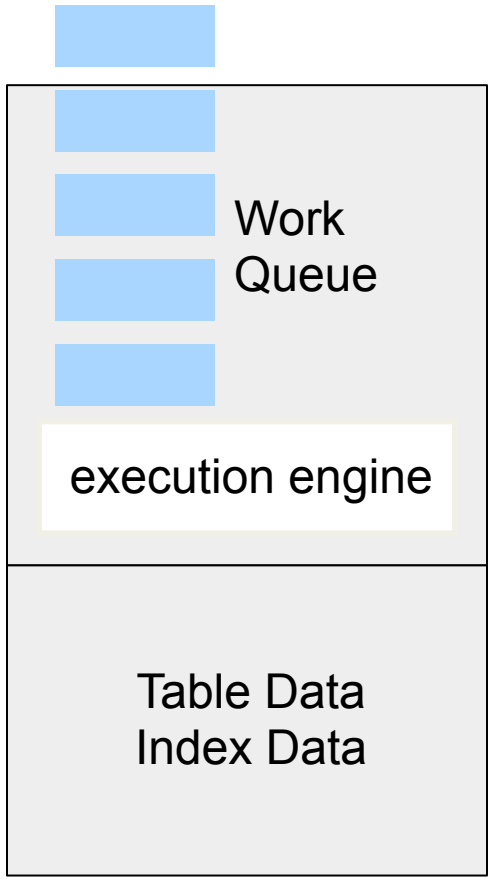
VoltDB Partitions

- A VoltDB partition contains data and code
- Each partition has its own CPU core and RAM
- Data – Two types of tables
  - + Replicated
    - — Rows exist within all VoltDB partitions
    - — Low frequency of modification (lookup tables: city, state, …)
  - + Partitioned
    - — Rows exists within a single VoltDB partition
    - — Single table column serves as partitioning key
    - — High frequency of modification (transactional data)



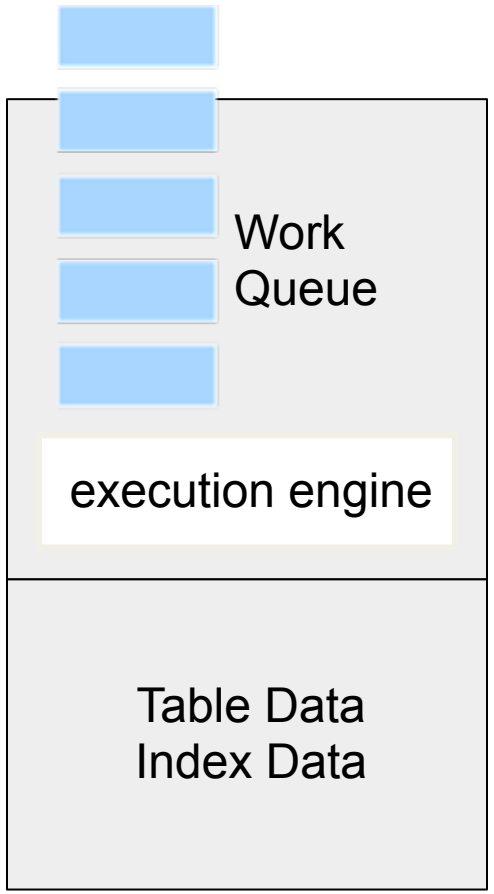
Inside a VoltDB Partition

- Each partition contains data and an execution engine
- The execution engine contains a queue for transaction requests
- Requests run to completion, serially, at each partition



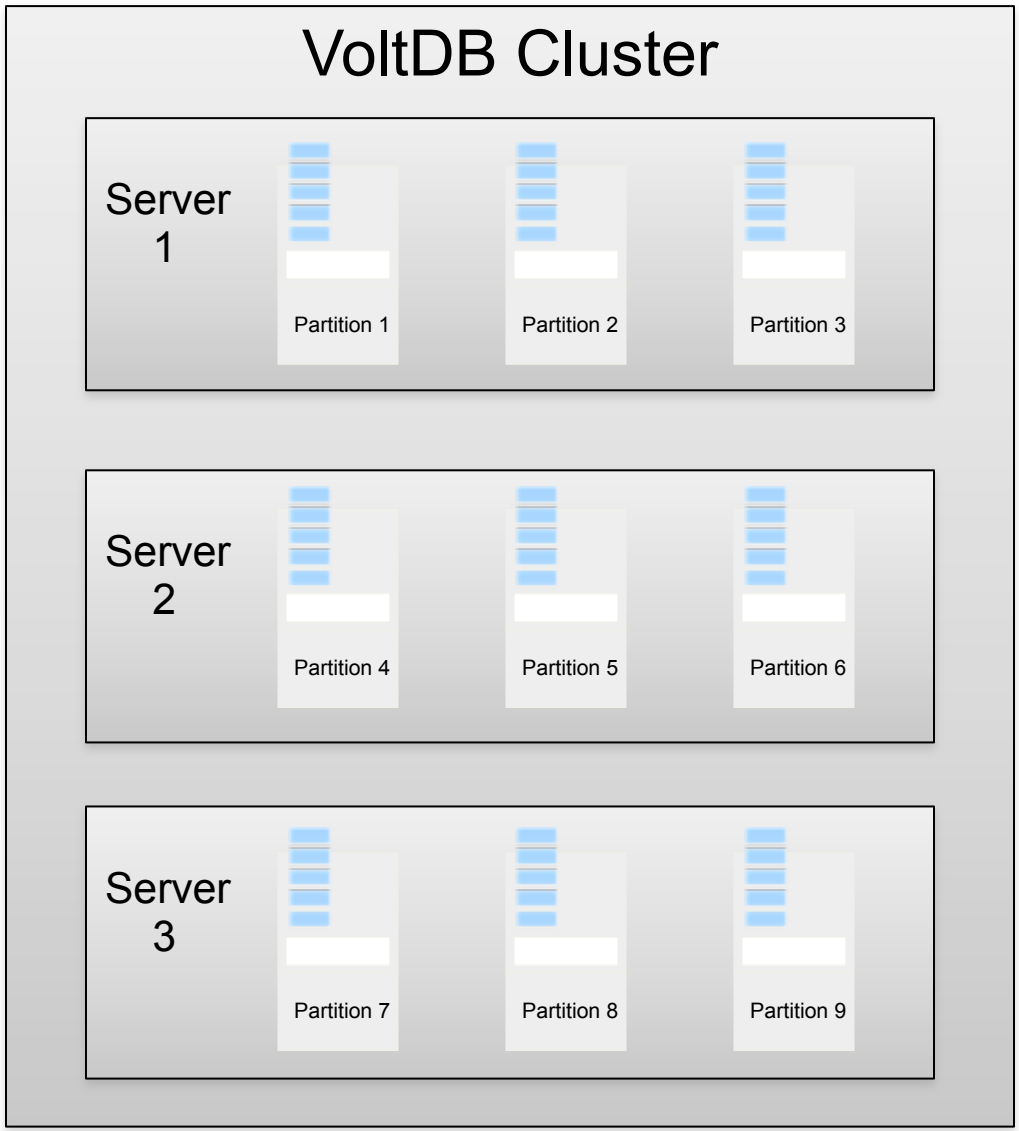
Inside a VoltDB Partition

- Each partition contains data and an execution engine
- The execution engine contains a queue for transaction requests
- Requests run to completion, serially, at each partition



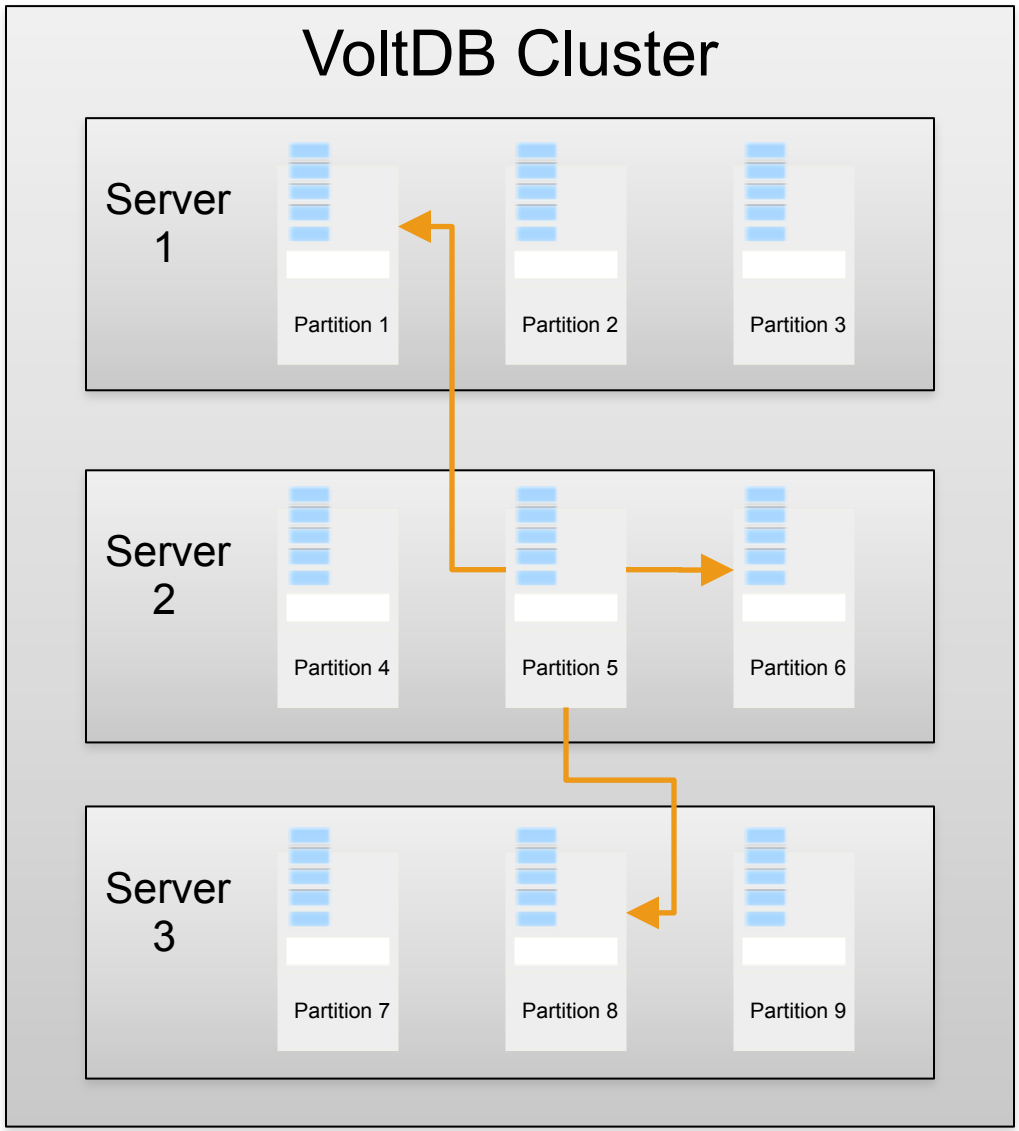
Transactions – ACID

- Single partition transactions
  - + All data is in one partition
  - + Each partition operates autonomously
- Multi-partition transactions
  - + one partition distributes and coordinates work plans



Transactions – ACID

- Single partition transactions
  - + All data is in one partition
  - + Each partition operates autonomously
- Multi-partition transactions
  - + one partition distributes and coordinates work plans



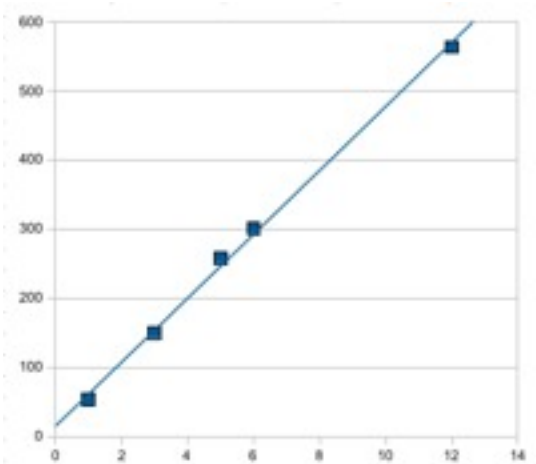
Transactions

- Transaction == Single SQL Statement or Stored Procedure Invocation
  - + Committed on Success
- Access VoltDB via Java stored procedures
  - + Combine the power of Java with SQL
  - + Efficiently process SQL at the server
  - + Move the code to the data, not the other way around

Clustering

Linear scale *(tested to 240 cores)*

Tps
,000



VoltDB Servers

Clustering

- Scalability
  - + 85+% on additional servers
  - + Add servers to increase performance and capacity

Linear scale *(tested to 240 cores)*

- Cluster management
  - + Replace failed nodes
  - + Change schema, stored procedures, security
  - + Enterprise Monitor provides web based management/monitoring

Client Interaction

- Client Options
  - + Libraries for Java, C++, C#, JDBC and other popular languages
  - + JSON via HTTP
  - + Asynchronous wire protocol and synchronous
- Client connects to the cluster
  - + Data location is transparent
  - + Topology is transparent
  - + Cluster manages routing, data movement and consistency

Availability and Data Protection

Availability and Data Protection

- High Availability
  - + Data stored on multiple machines (user configurable)
  - + Failover data redundancy
  - + No single point of failure

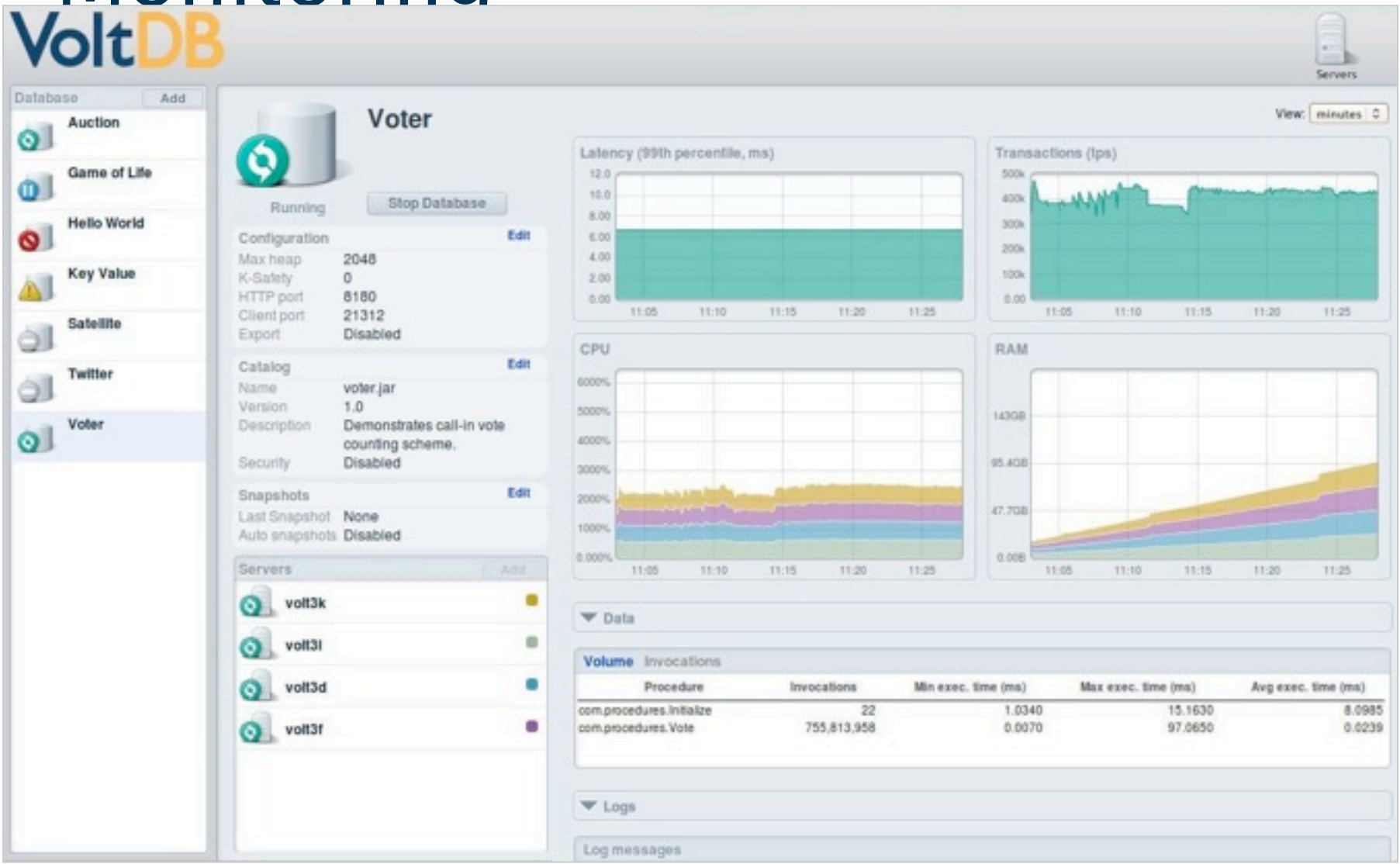
Availability and Data Protection

- High Availability
  - + Data stored on multiple machines (user configurable)
  - + Failover data redundancy
  - + No single point of failure
- Disk Based Persistence
  - + Cluster-wide consistent copy of all data
  - + Simplifies backup/restore
  - + Database Snapshots
  - + Command Logging - transactionally durable to disk
    - — Available September 2011

VoltDB OLAP Database Integration

- VoltDB high-throughput export feature
  - + Export of real-time and "near-time" data to target data stores
  - + Enrich data prior to export
    - — Pre-join, de-duplicate, aggregate
- VoltDB Export key features
  - + Loosely-coupled integration
  - + Buffer for impedance mismatches
  - + Auto-discovery of cluster configurations with retry
- Direct Hadoop integration

Database Management & Monitoring



Using VoltDB

VoltDB gives us both the advantages of a relational database and the speed and linear scalability of NoSQL solutions

JasperLabs

Using VoltDB, we can significantly increase the performance of ad campaigns at a fraction of the cost

Jingit

I would recommend VoltDB to anyone looking to build high scaling database applications

In a variety of financial applications, analytics are as important as transaction processing. VoltDB provides the ability to do both at very high speed and scale

With VoltDB, we don't have to trade consistency for significantly better performance and scale

End