GridGain

apache Ignite

Comparing Apache Ignite and Cassandra For HTAP Applications

Denis Magda
Apache Ignite PMC Chair
GridGain Product Management

© 2018 GridGain Systems, Inc.

Agenda

- Hybrid Transactional/Analytical Processing
- Ignite vs. Cassandra
  - Denormalized Architecture or Collocated Processing?
  - Row-level Isolation or Distributed Transactions?
  - Caching or In-Memory Store?
- Q & A

© 2018 GridGain Systems, Inc.

Hybrid Transaction/Analytical Processing

© 2018 GridGain Systems, Inc.

“HTAP is an emerging application architecture that "breaks the wall" between transaction processing and analytics. It enables more informed and "in business real time" decision making.” (Gartner Inc., 2014)

© 2018 GridGain Systems, Inc.

Ignite vs. Cassandra:

Denormalized Architecture or Collocated Processing?

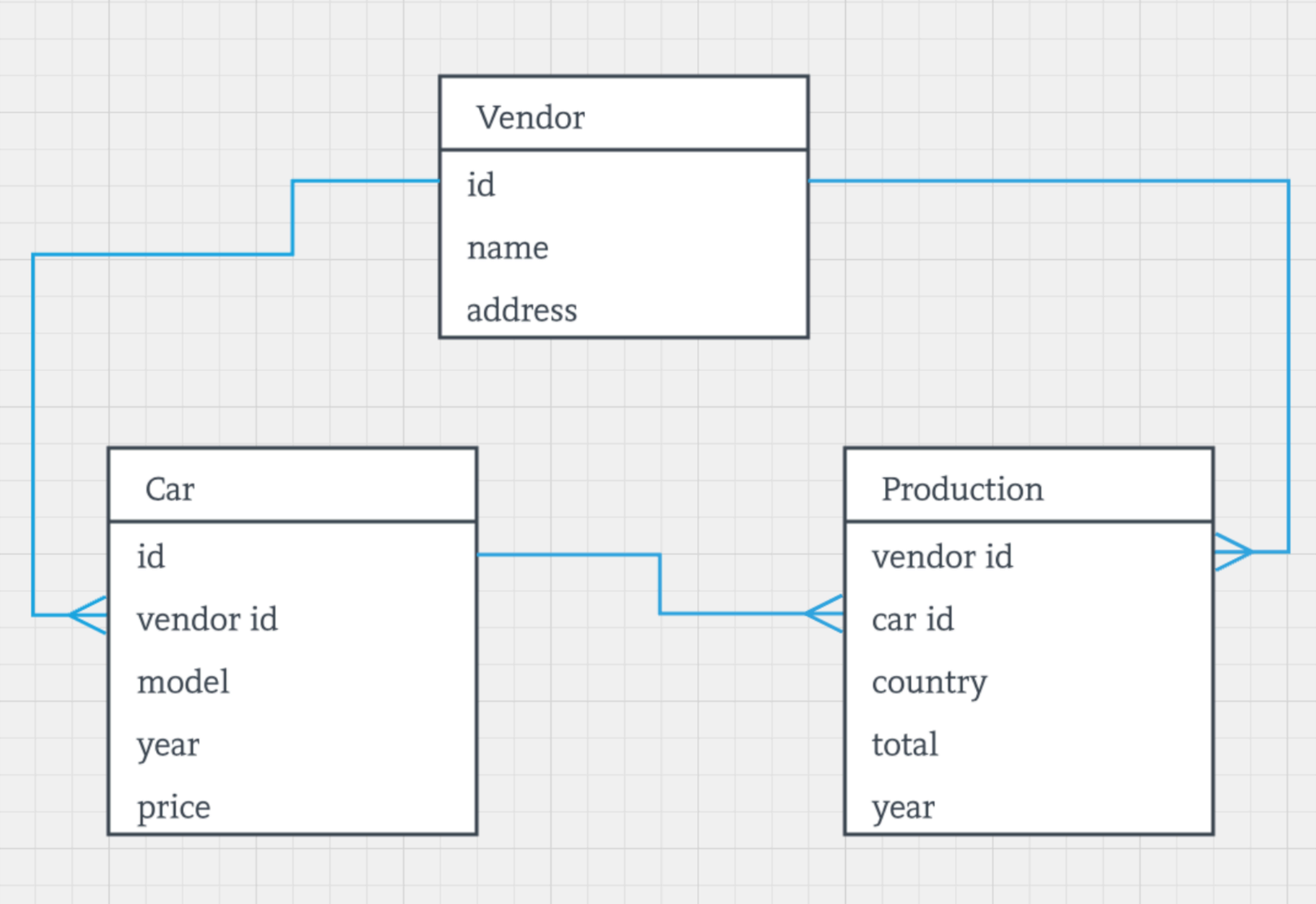
© 2018 GridGain Systems, Inc.

Cassandra Path: Denormalized Architecture

- Denormalization Strategy
  - Increasing Performance in Favor of Data Redundancy
- Pros
  - Performance
- Cons
  - Query-Driven Architecture
  - Easy to Start and Hard to Evolve

© 2018 GridGain Systems, Inc.

Let’s Take an Example



© 2018 GridGain Systems, Inc.

Start With a Question!

Q1: What are the car models produced by a vendor within a particular time frame (newest first)?

```
CREATE TABLE cars_by_vendor_year_model (
 vendor_name text,
 production_year int,
 car_model text,
 total int,
 PRIMARY KEY ((vendor_name), production_year, car_model)
) WITH CLUSTERING ORDER BY (production_year DESC, car_model ASC);
```

© 2018 GridGain Systems, Inc.

Run Queries

Q1: What are the car models produced by a vendor within a particular time frame (newest first)?

```
SELECT car_model, production_year, total
FROM cars_by_vendor_year_model
WHERE vendor_name = 'Ford Motors' and production_year >= 2017;
```

Extra Queries Supported Out of the Box

```
SELECT * FROM cars_by_vendor_year_model
WHERE vendor_name = 'Ford Motors';
```

© 2018 GridGain Systems, Inc.

Somewhere in Production…

Q2: What is the number of cars of a specific model produced by a vendor?

© 2018 GridGain Systems, Inc.

Let’s reuse existing table!

Q2: What is the number of cars of a specific model produced by a vendor?

```
SELECT production_year, total
FROM cars_by_vendor_year_model
WHERE vendor_name = 'Ford Motors'
and car_model = 'Edge';
```

© 2018 GridGain Systems, Inc.

Let’s reuse existing table!

Q2: What is the number of cars of a specific model produced by a vendor?

InvalidRequest: code=2200 [Invalid query] message="PRIMARY KEY column "car_model" cannot be restricted (preceding column "production_year" is not restricted)"

© 2018 GridGain Systems, Inc.

Solution: Create New Table

Q2:

```
CREATE TABLE cars_by_vendor_model (
 vendor_name text,
 car_model text,
 production_year int,
 total int,
 PRIMARY KEY ((vendor_name), car_model, production_year)
);
```

Q1:

```
CREATE TABLE cars_by_vendor_year_model (
 vendor_name text,
 production_year int,
 car_model text,
 total int,
 PRIMARY KEY ((vendor_name), production_year, car_model)
) WITH CLUSTERING ORDER BY (production_year DESC, car_model ASC);
```

© 2018 GridGain Systems, Inc.

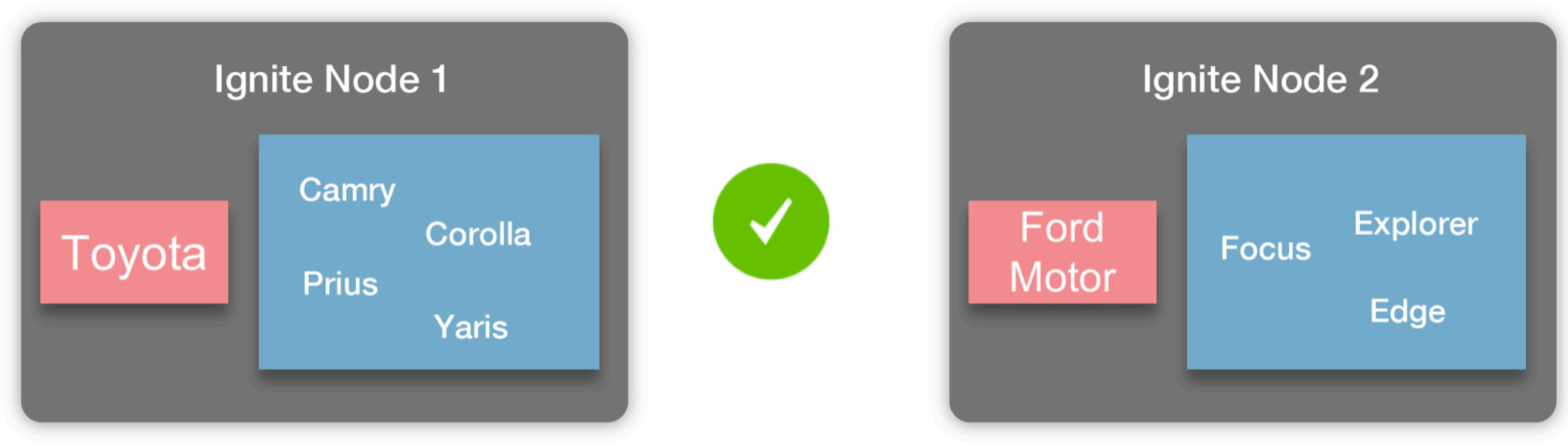
Ignite Path: Affinity Collocation

- Related Data Collocation
  - Countries and Cities, Vendors and Cars
  - On a Single Cluster Node

- Collocated Processing
  - Efficient SQL JOINS
  - Map-Reduce for Fast Data
  - Machine Learning w/o ELT
  - Reduced Network Traffic -> Better Performance

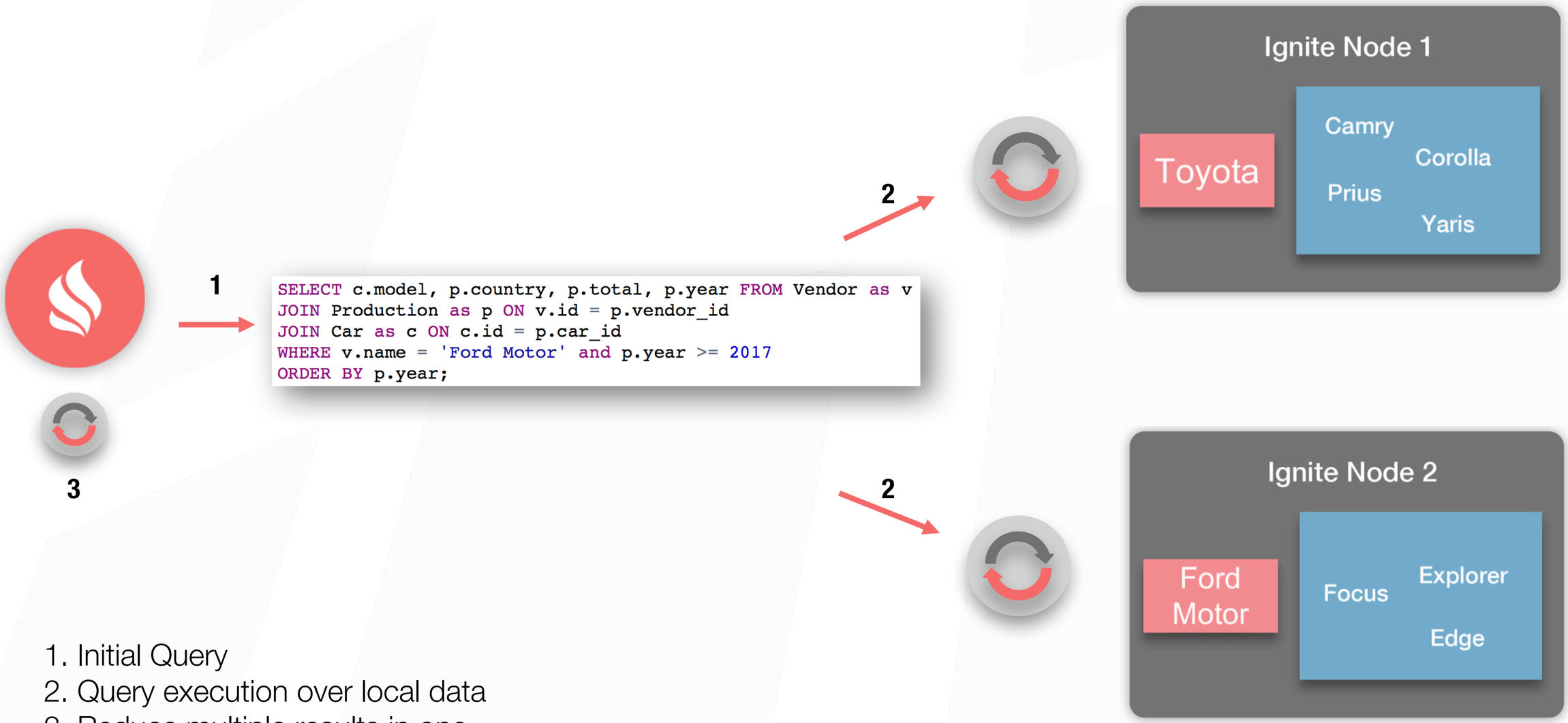
© 2018 GridGain Systems, Inc.

Collocated Data Distribution



© 2018 GridGain Systems, Inc.

Collocated JOINs



1. Initial Query
2. Query execution over local data
3. Reduce multiple results in one

© 2018 GridGain Systems, Inc.

GridGain Company Confidential

GridGain

Machine Learning: Genetic Algorithms

Biological Evolution
Simulation

F = F1 + F2

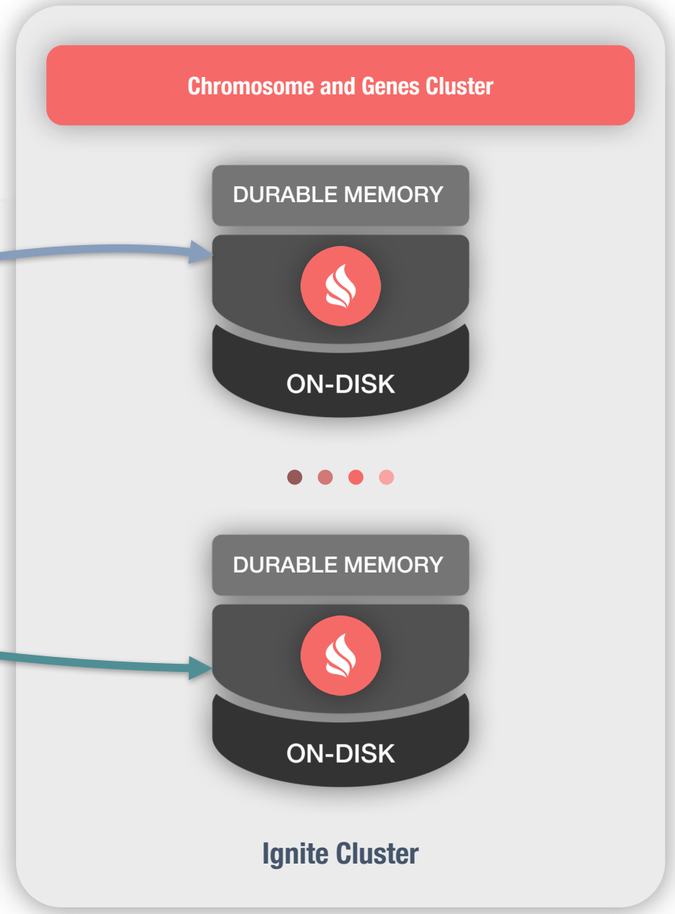
C = C1 + C2

M = M1 + M2

Collocated Computation

F1, C1, M1

F2, C2, M2



F = Fitness Calculation
C = Crossover
M = Mutation

© 2018 GridGain Systems, Inc.

Ignite vs. Cassandra:

Row-level Isolation or Distributed Transactions?

© 2018 GridGain Systems, Inc.

Cassandra Path: Lightweight Transactions

- Cassandra is Eventually Consistent DB
  - With Tunable Consistency

- Lightweight Transactions
  - Aka. Compare and Set Transactions
  - Row-Level Isolation

- Applicability
  - Atomic and linearizable updates of a record
  - Prevent duplications on inserts

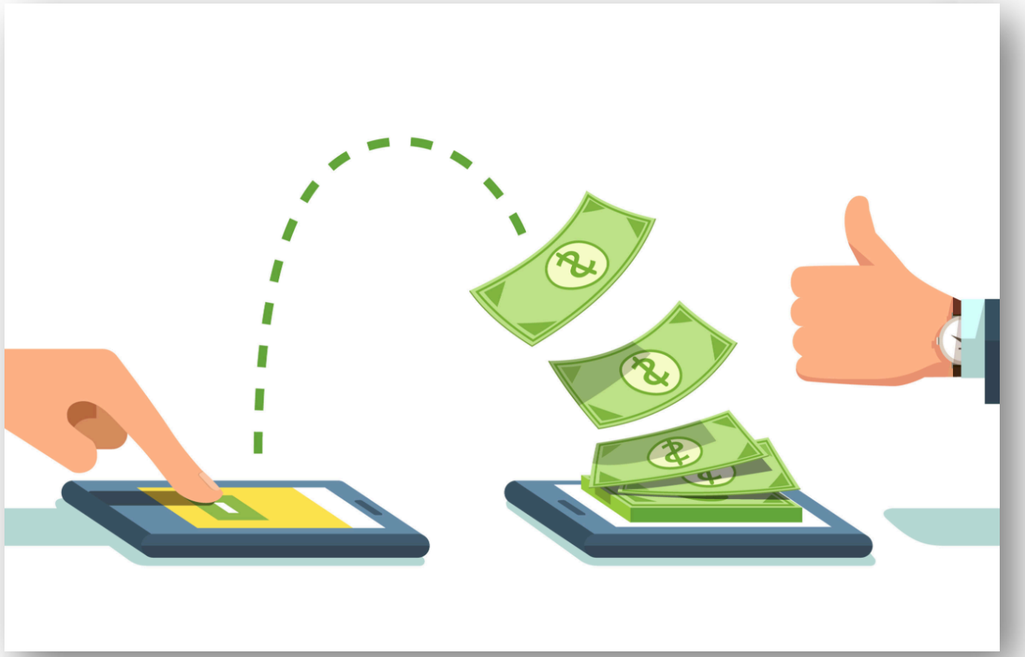
```
UPDATE user_account
SET    user_age=35
IF     user_id='Bob Smith';
```

© 2018 GridGain Systems, Inc.

Ignite Path: ACID Transactions

- Distributed ACID Transactions
  - Pessimistic/Optimistic
- 2 Phase Commit
  - From RAM to disk
- Deadlock-free Transactions

- Applicability
  - No limitations. Classic Transactions.



© 2018 GridGain Systems, Inc.

GridGain Company Confidential

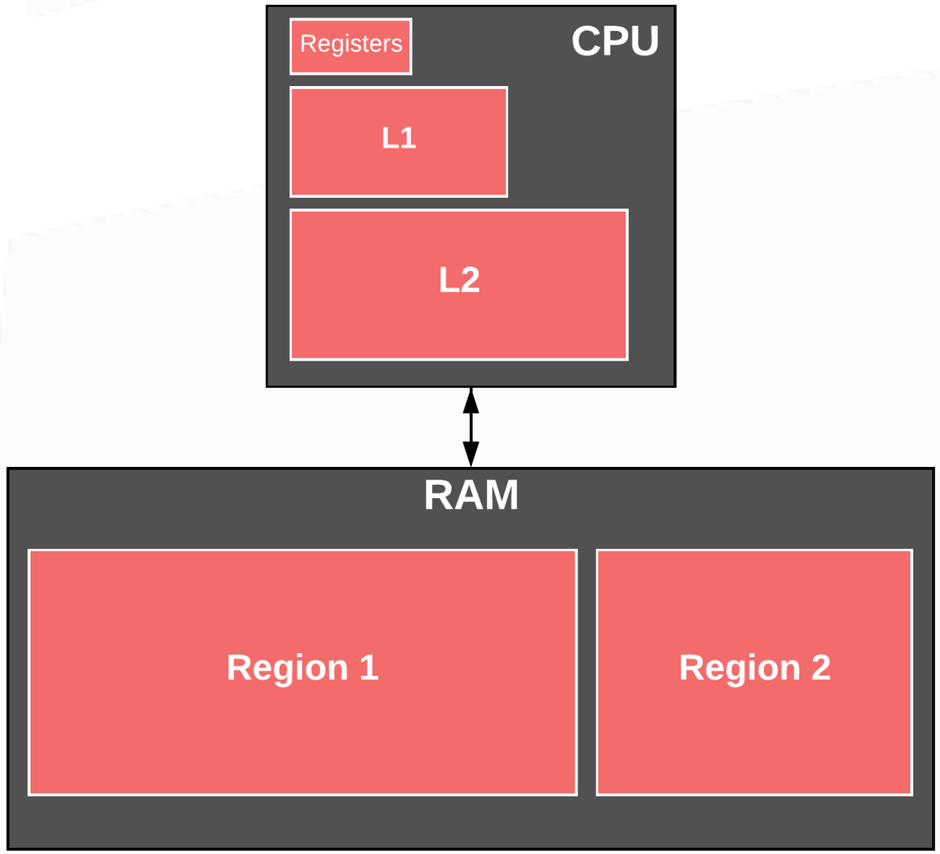
Ignite vs. Cassandra:

Caching or In-Memory Store?

© 2018 GridGain Systems, Inc.

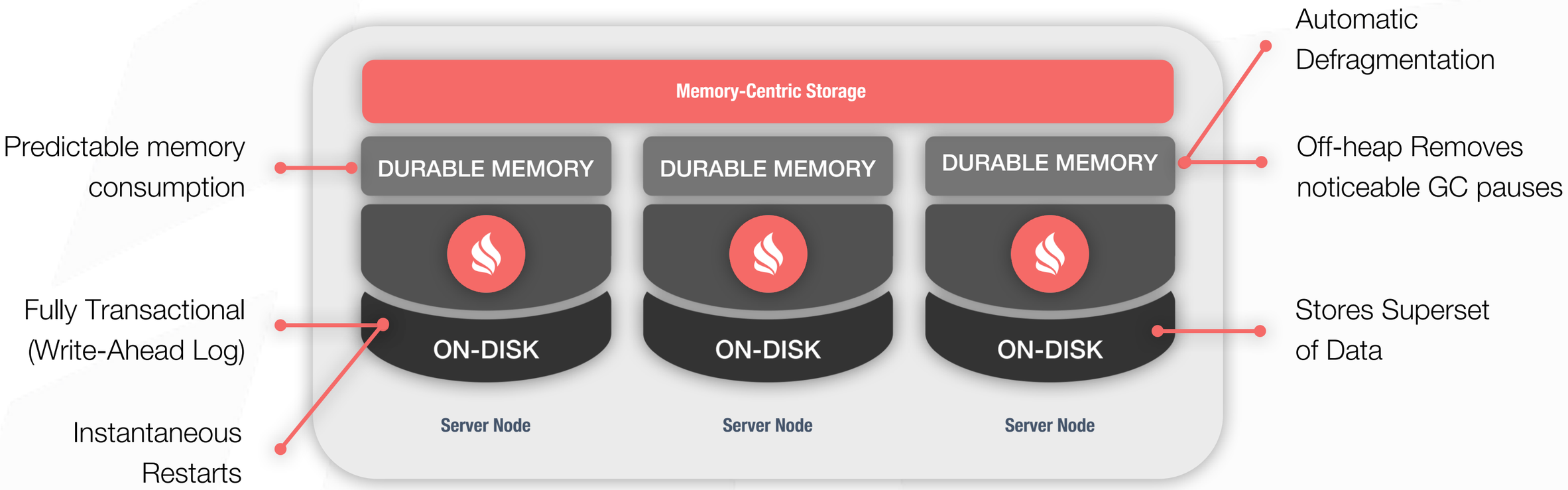
Caching Cassandra

- Off-heap Options
  - Page Cache
  - Row Cache
  - Bloom Filters
- Java Heap Options
  - Key Cache
  - DSE In-Memory



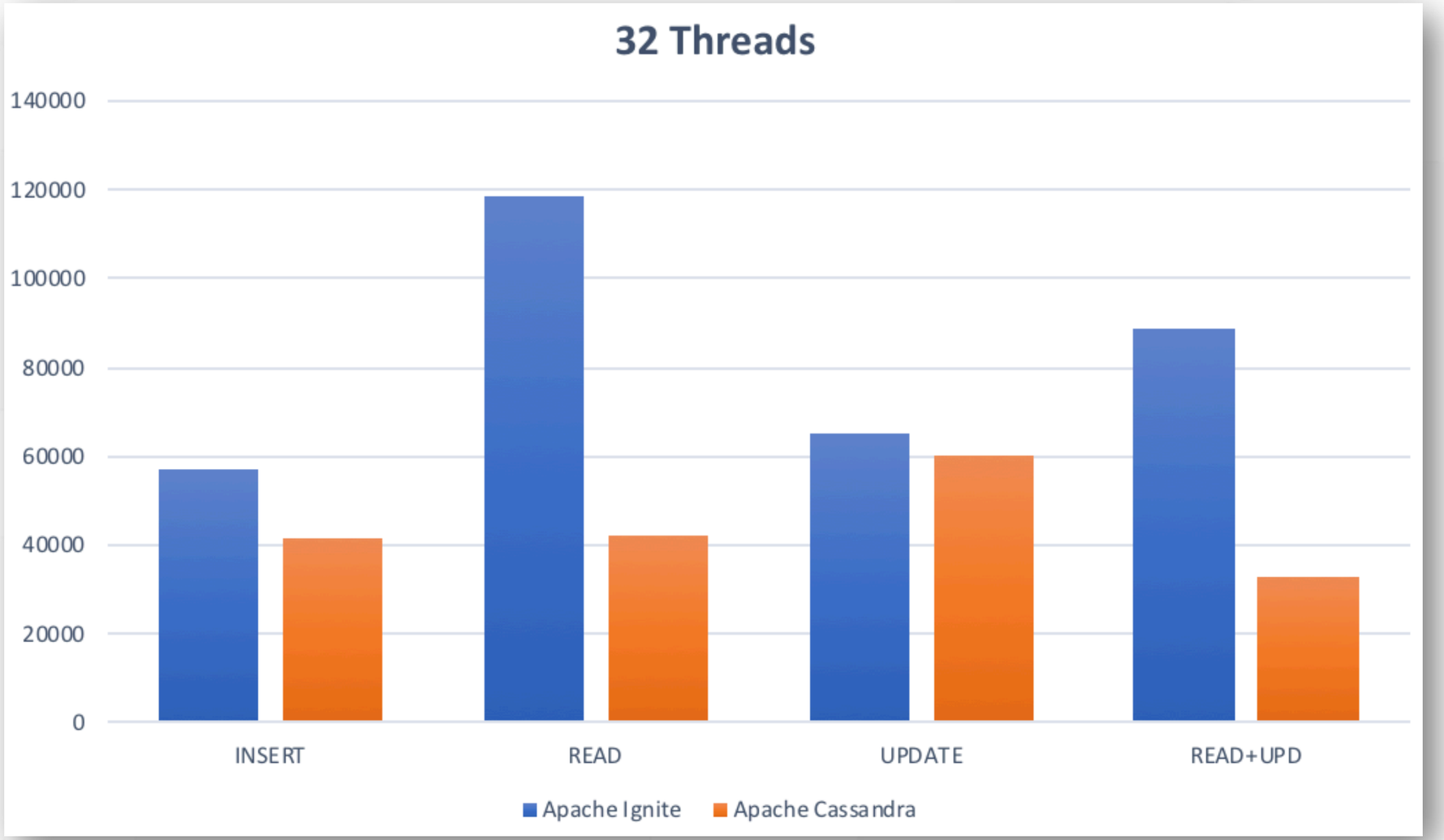
© 2018 GridGain Systems, Inc.

Ignite Memory-Centric Storage



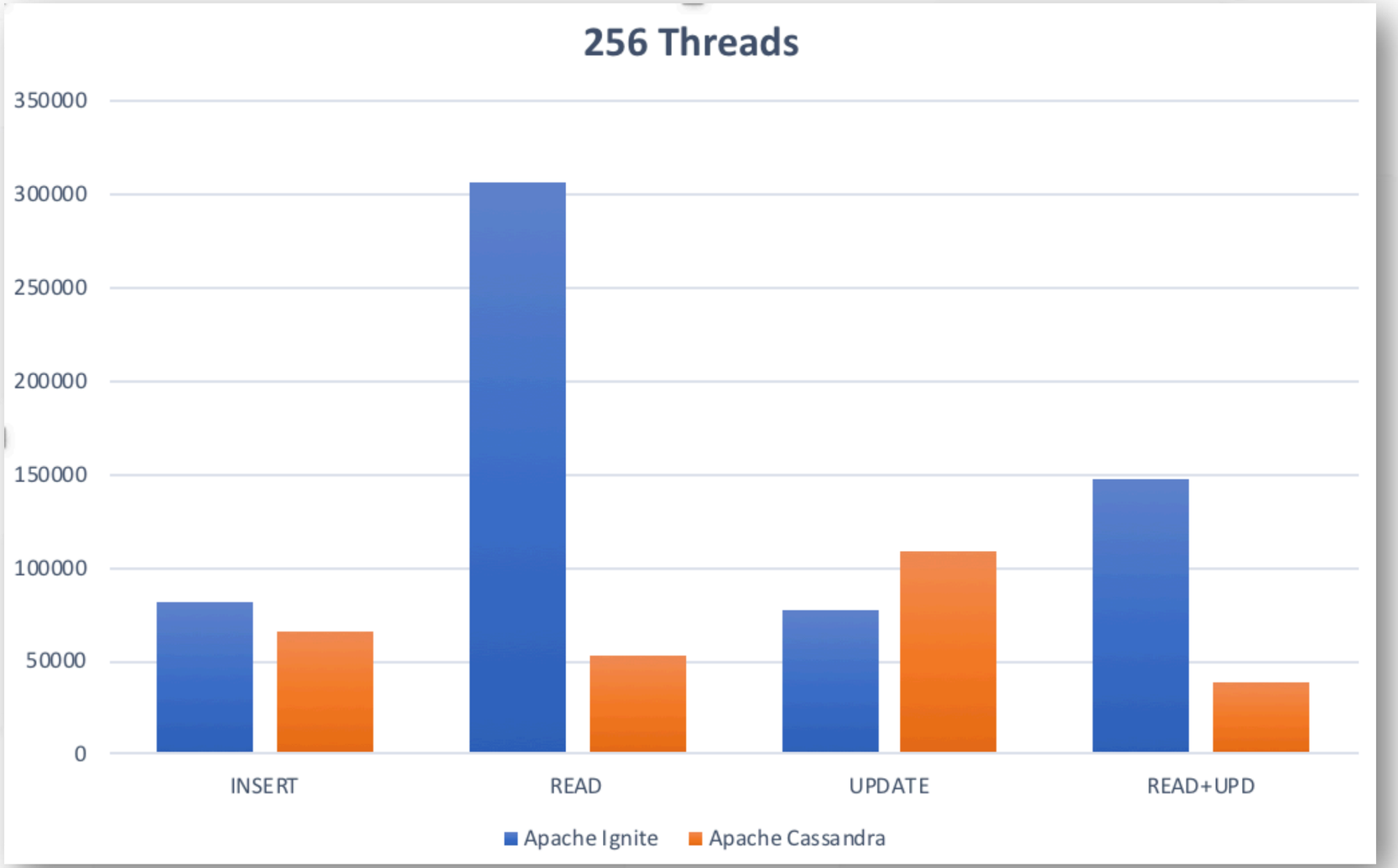
© 2018 GridGain Systems, Inc.

YCSB: Average Load



© 2018 GridGain Systems, Inc.

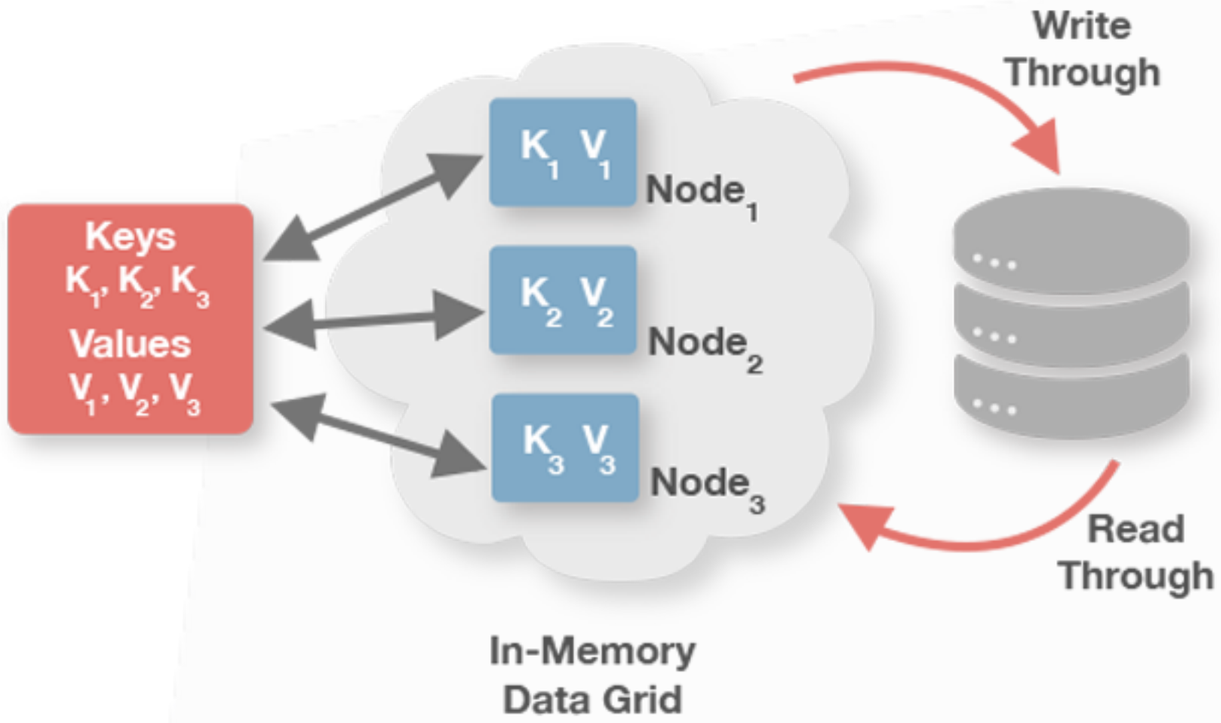
YCSB: Ramping Up Load



© 2018 GridGain Systems, Inc.

Caching Cassandra With Ignite

- No rip-and-replace
  - Keep Cassandra
- Automatic Read/Write-Through
  - Key-Value Only
- Distributed SQL
  - Over Ignite Data
- ACID Transactions
  - Ignite layer



© 2018 GridGain Systems, Inc.

Summary: Ignite or Cassandra?

- Simplified Architecture
  - Denormalization vs Affinity Collocation
- Data Consistency and Transactions
  - Ignite: ACID Transactions
  - Cassandra: Row-level Isolation
- In-Memory Store and Performance
  - Ignite: read intensive and mixed workloads
  - Cassandra: write intensive workloads (big load)

© 2018 GridGain Systems, Inc.

Any Questions?

Thank you for joining us. Follow the conversation.
https://ignite.apache.org

https://www.gridgain.com

#apacheignite
#gridgain
#dmagda

© 2018 GridGain Systems, Inc.