GridGain

How to boost and scale Postgres – from sharding to in-memory data grids

Denis Mekhanikov

Apache Ignite Contributor

GridGain Client Service Lead

Agenda

- Tapping into RAM with caching techniques
- Sharding and replication solutions
- Cache and scale out with in-memory data grids
- Q&A

Caching Techniques

Ultimate Purpose of Caching

Speed up operations by reducing
disk access and **computation** (i.e. CPU)

Computer Latency at Human Scale

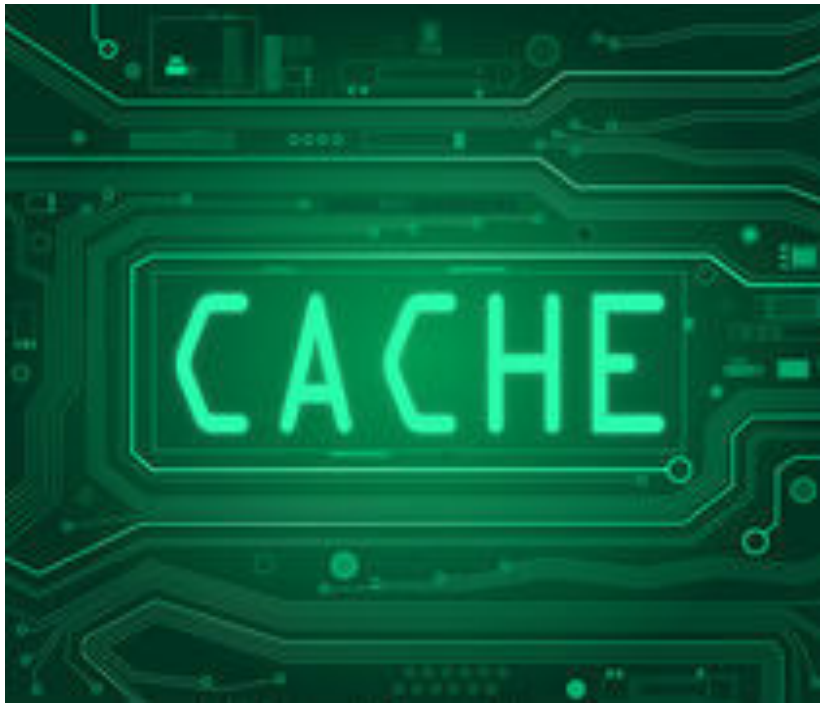
System Event	Actual Latency	Scaled Latency
One CPU cycle	0.4 ns	1 s
Level 1 cache access	0.9 ns	2 s
Level 2 cache access	2.8 ns	7 s
Level 3 cache access	28 ns	1 min
Main memory access (DDR DIMM)	~100 ns	4 min
Intel Optane DC persistent memory access	~350 ns	15 min
Intel Optane DC SSD I/O	< 10 µs	7 hrs
NVMe SSD I/O	~25 µs	17 hrs
SSD I/O	50-150 µs	1.5 - 4 days
Rotational disk I/O	1 – 10ms	1 – 9 months
Internet: SF to NY	65 ms	5 years

Computer Latency at Human Scale

System Event	Actual Latency	Scaled Latency
One CPU cycle	0.4 ns	1 s
Level 1 cache access	0.9 ns	2 s
Level 2 cache access	2.8 ns	7 s
Level 3 cache access	28 ns	1 min
Main memory access (DDR DIMM)	**~100 ns**	**4 min**
Intel Optane DC persistent memory access	**~350 ns**	**15 min**
Intel Optane DC SSD I/O	< 10 µs	7 hrs
NVMe SSD I/O	~25 µs	17 hrs
SSD I/O	50-150 µs	1.5 - 4 days
Rotational disk I/O	1 – 10ms	1 – 9 months
Internet: SF to NY	65 ms	5 years

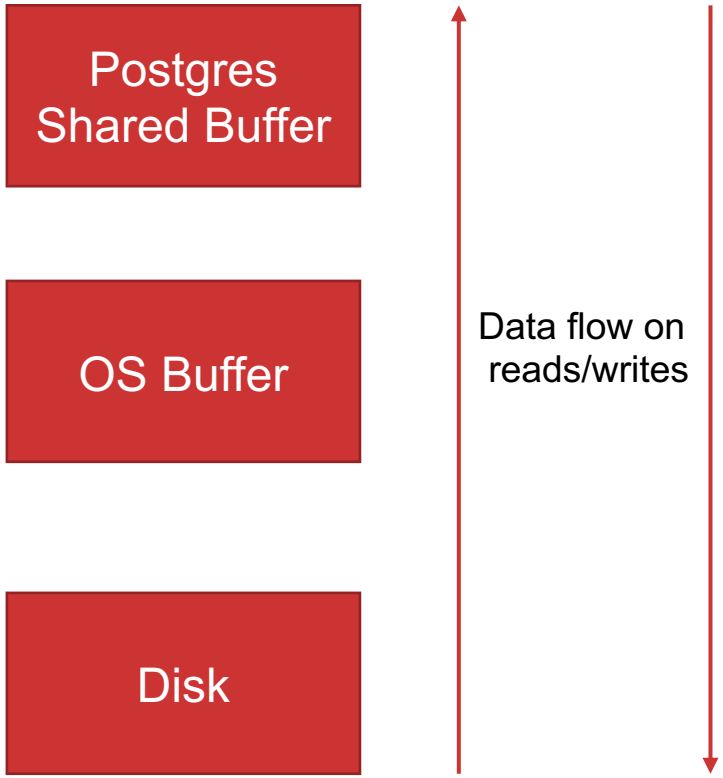
Basic Types of Caching in Postgres

- Query result caching
- Query plan caching
- Relation caching
  - Data and indexes



Relation caching: Shared Buffer and OS Buffer

- Postgres Shared Buffer Cache
  - Allocated and managed by Postgres
- OS Buffer (aka. Page Cache)
  - Caches chunks (pages) of files
- Suggestions/considerations:
  - No silver bullet – select and tune
  - Possible duplication between shared and OS caches
  - Limited by local RAM capacity



Horizontal Scalability

GridGain

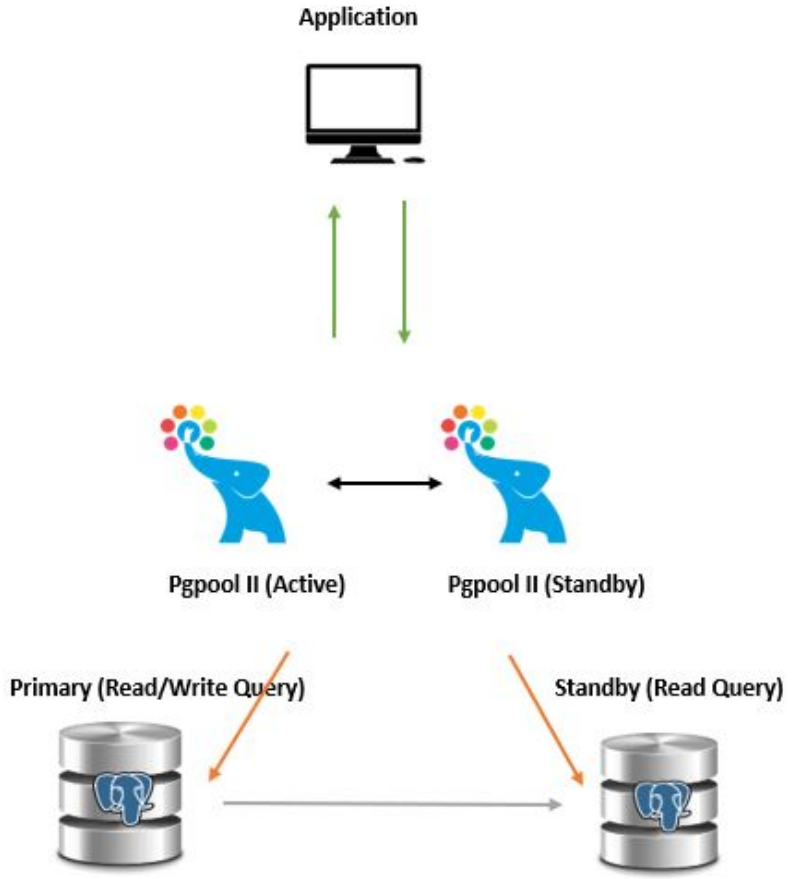
Defining Requirements for Solution

- Strong Consistency (ACID)
- Load Balancing
- Failover

Pgpool 2 for Read-Heavy Workloads

- Pgpool coordinator
- Primary for reads and writes
- Hot replicas for reads

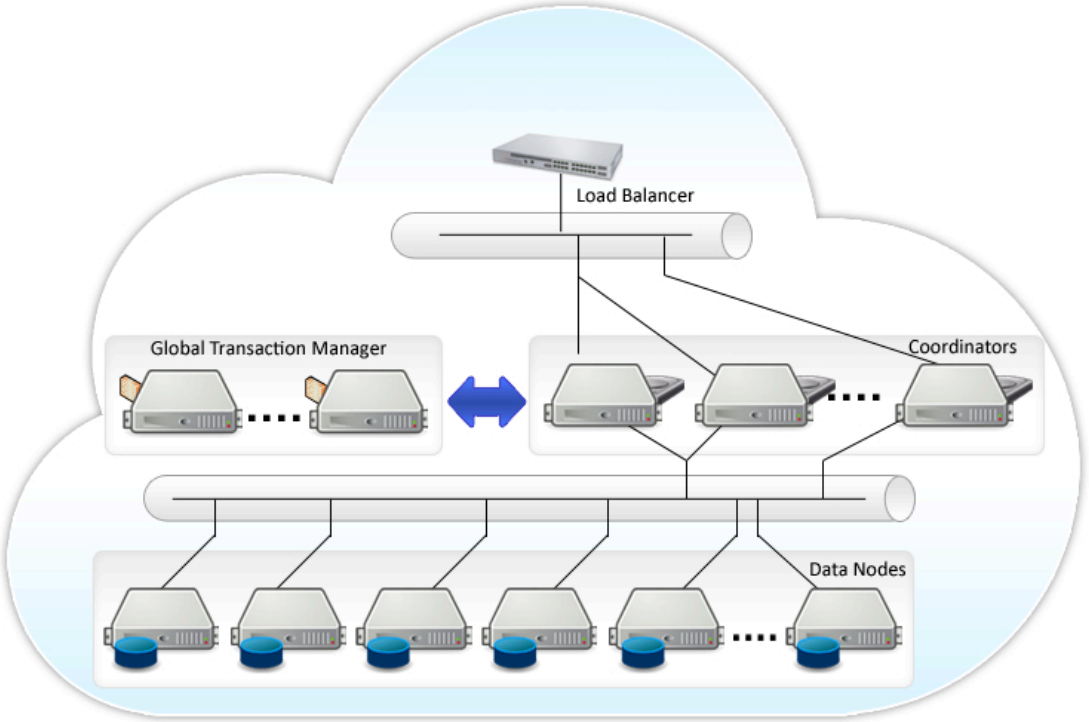
- Suggestions/considerations:
  - Good for load balancing of read-heavy workloads
  - ACID enforces sync replication and limits a number of replicas
  - Primary machine capacity defines your total cluster capacity



Sharding With PostgreSQL-XL or CitusData

- Coordinator keeps metadata and distributes queries
- Data nodes store shards/partitions
- Supports data co-location and JOINs

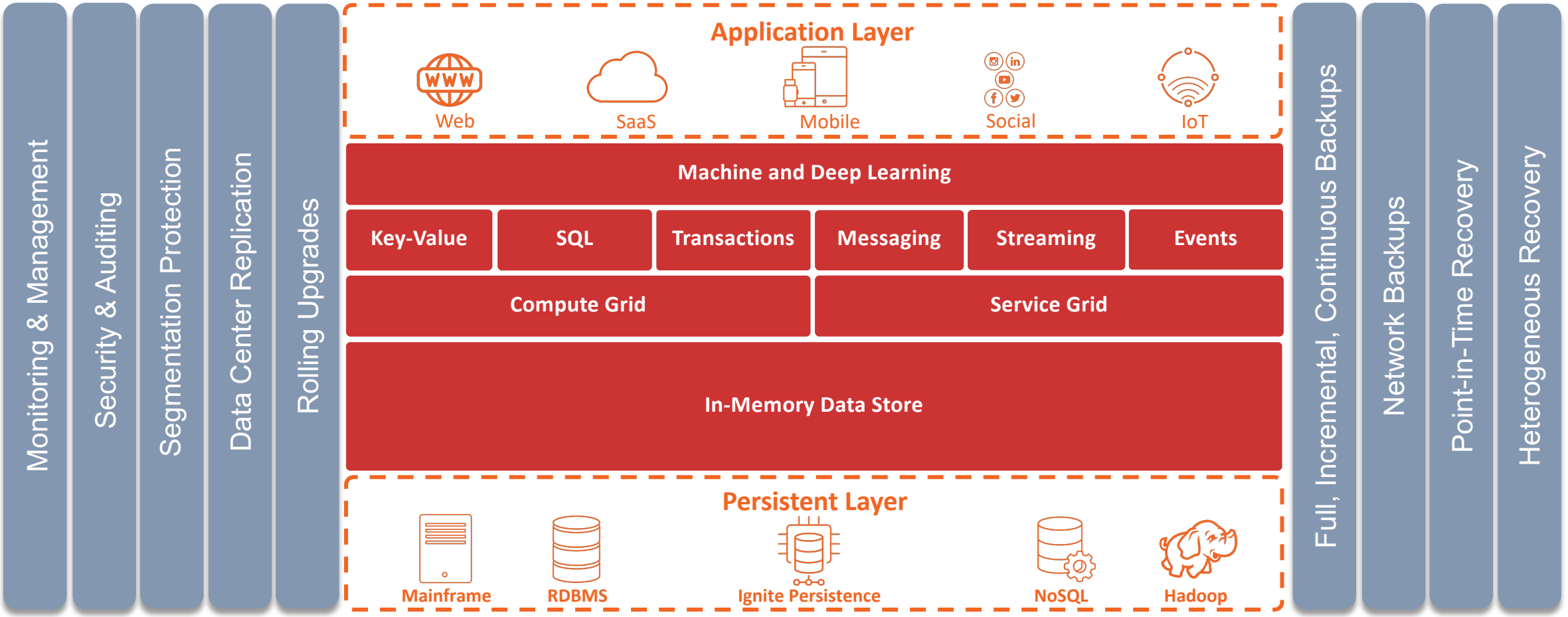
- Suggestions/considerations:
  - Suited for mixed workloads
  - Total capacity is your cluster capacity
  - Scaling and failover is not trivial
  - Disk-based solution



Caching and Scaling With In-Memory Data Grids

GridGain

Apache Ignite/GridGain In-Memory Computing Platform

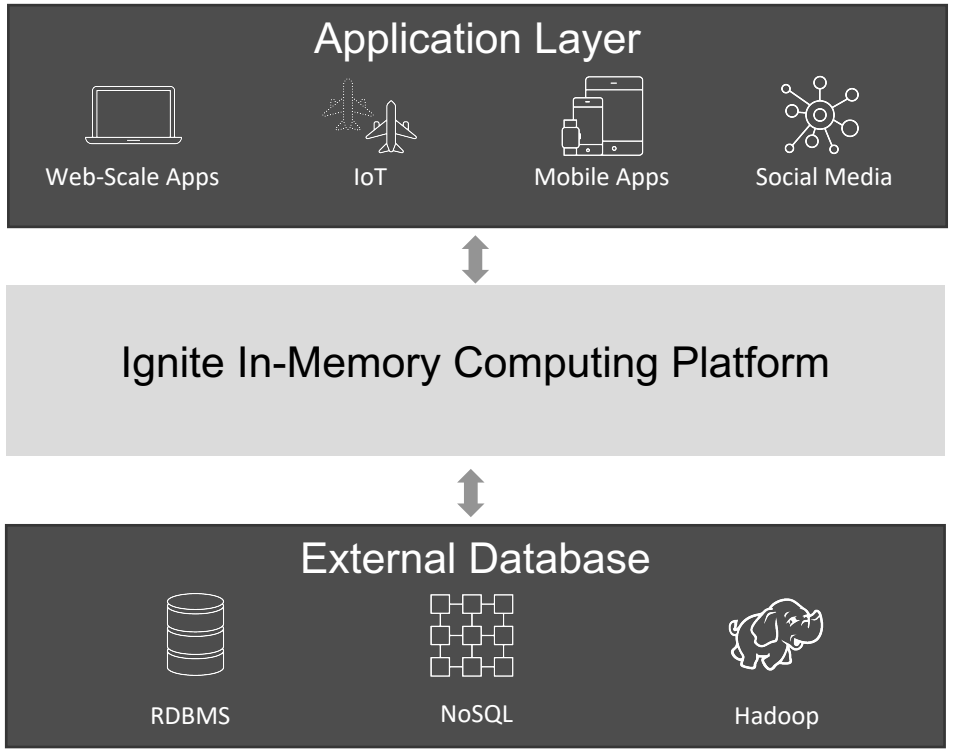


Apache Ignite Features
GridGain Enterprise Features

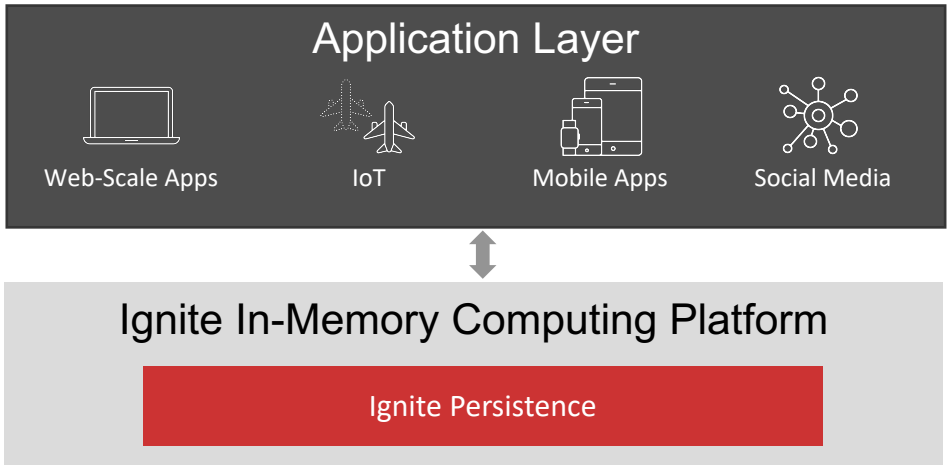
GridGain

Primary Ignite Deployment Modes

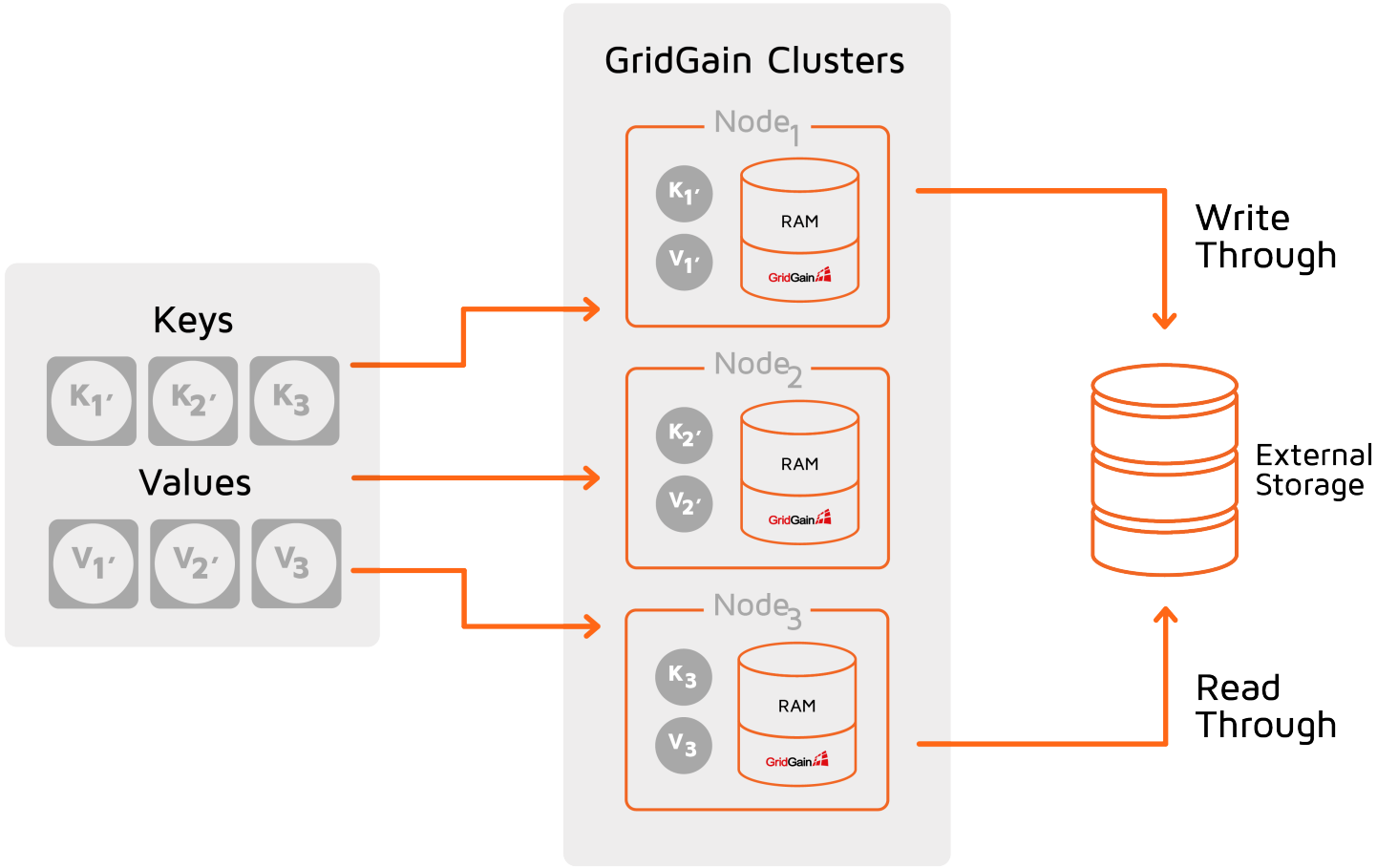
Enhance Legacy Architecture - IMDG



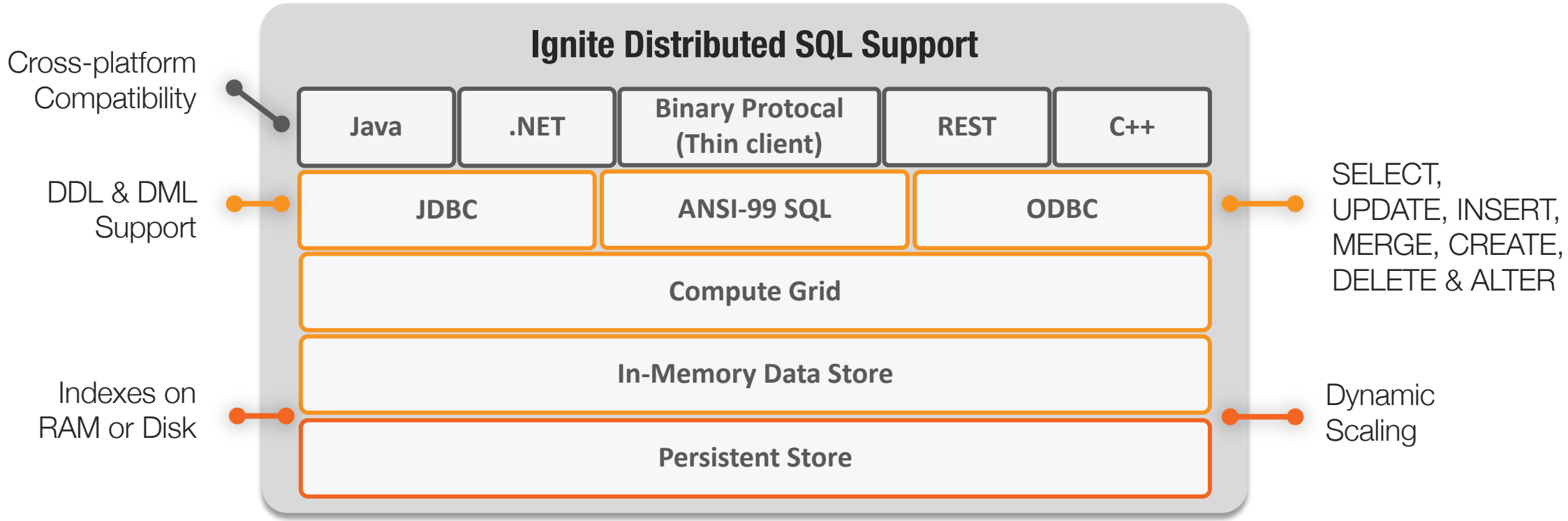
Simplified Modern Architecture - IMDB



How Postgres is Accelerated?



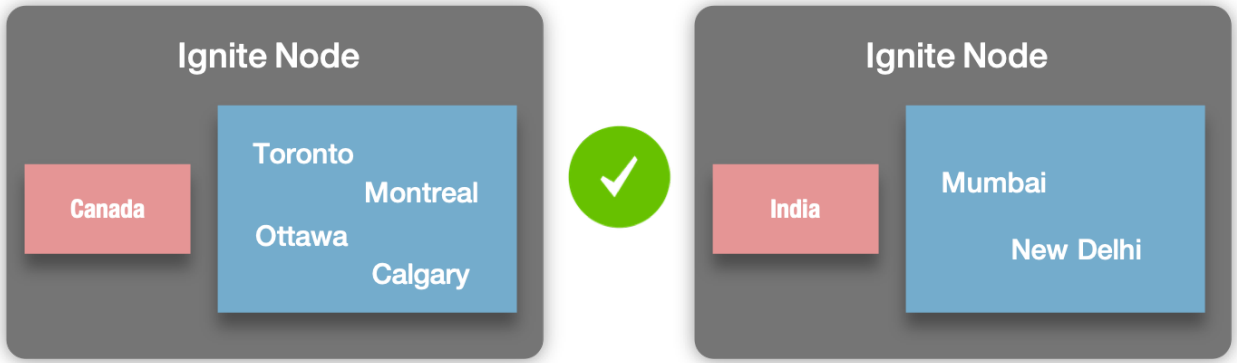
Distributed SQL



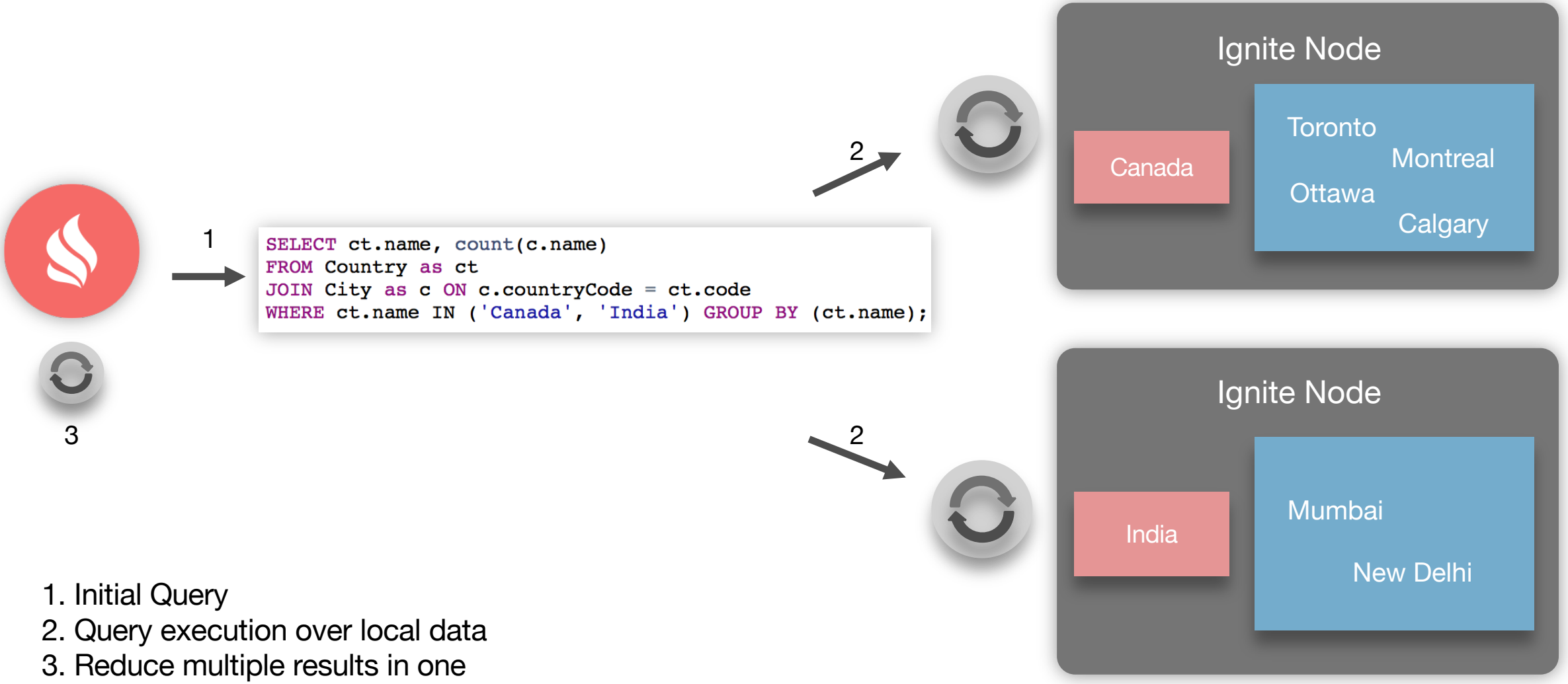
GridGain

Holy Grail of Distributed World: Affinity Collocation

- Related data is on the same node
  - Countries and Cities
  - Departments and Employees
- Collocated Processing
  - Efficient execution planning
  - Reduced network traffic
  - Performance boost!



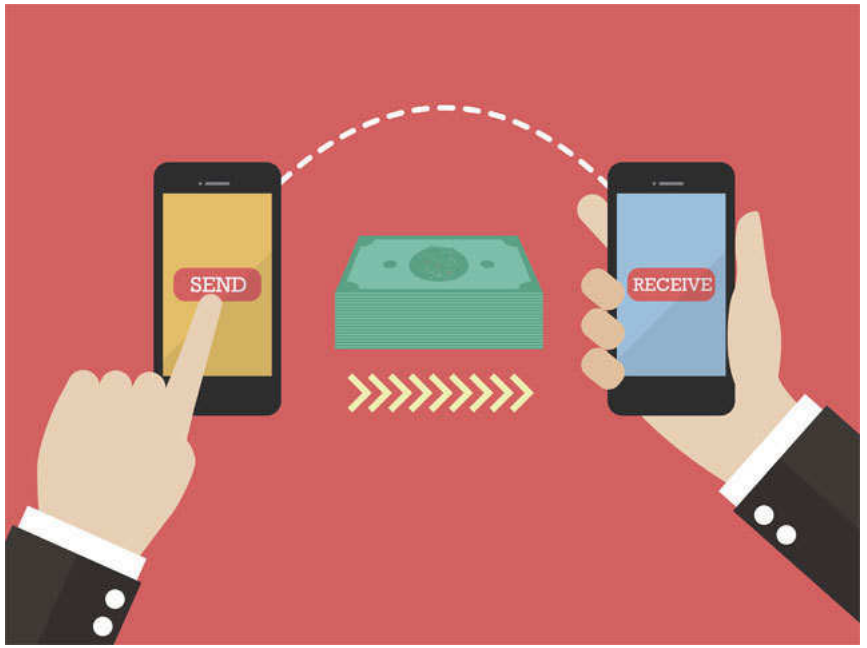
Ignite SQL Queries



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

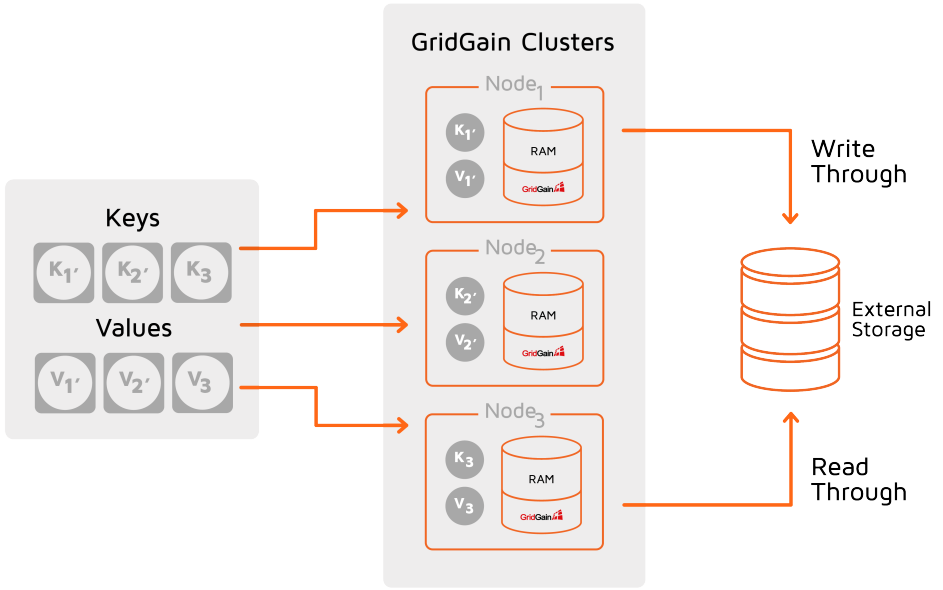
Transactions and Consistency

- Distributed Key-Value Transactions
  - 2 phase commit protocol
  - Spans to Postgres
- Transactional SQL (Beta)
  - MVCC
- Strong or relaxed consistency
  - Atomic and transactional tables
  - Tunable Write-ahead-log settings



Consistency Across Postgres and Ignite/GridGain

- Coordinator writes to the database first
- Commits in the cluster afterwards
- The database must be transactional
  - Postgres!



GridGain

Demo

Q&A

@apacheignite
@gridgain

dmekhanikov@gmail.com
https://github.com/dmekhanikov

GridGain