Shared Memory Layer for Spark Applications

Fast Data Meets Open Source

DMITRIY SETRAKYAN

Founder, PPMC

http://www.ignite.incubator.apache.org

@apacheignite

@dsetrakyan

GridGain

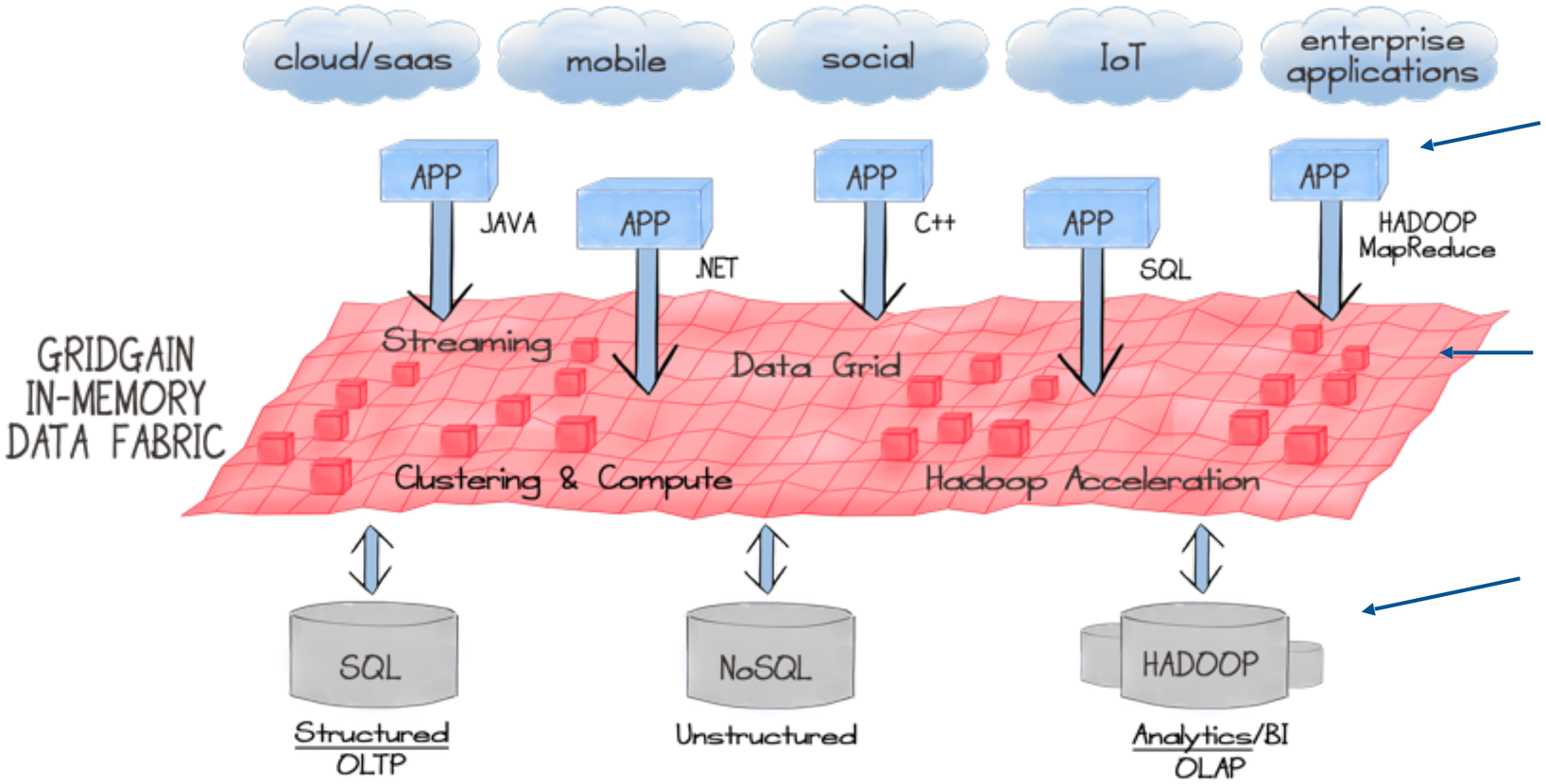
Agenda

- About In-Memory Computing
- Advanced Clustering
- Data Grid
- Streaming & CEP
- Share State Across Spark Jobs
- In-Memory MapReduce
- Interactive SQL
- DevOps: Yarn and Mesos
- Q & A

Apache Ignite for BI and Analytics

GridGain

Apache Ignite™ In-Memory Data Fabric: Strategic Approach to IMC



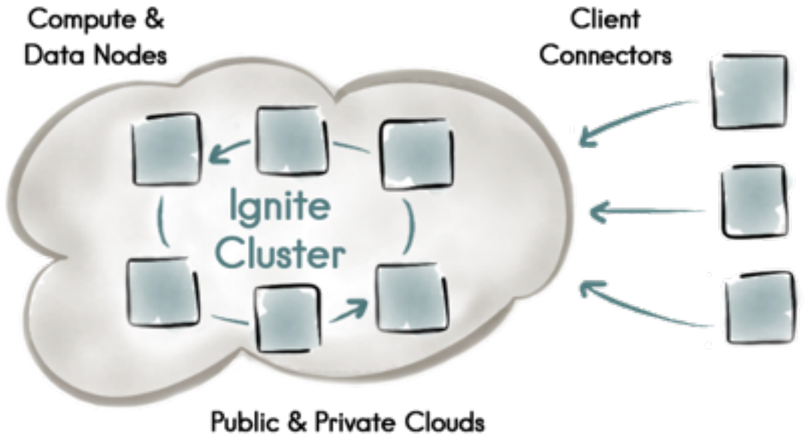
- Supports Applications of various types and languages

- Open Source – Apache 2.0
- Simple Java APIs
- 1 JAR Dependency
- High Performance & Scale
- Automatic Fault Tolerance
- Management/Monitoring
- Runs on Commodity Hardware

- Supports existing & new data sources
- No need to rip & replace

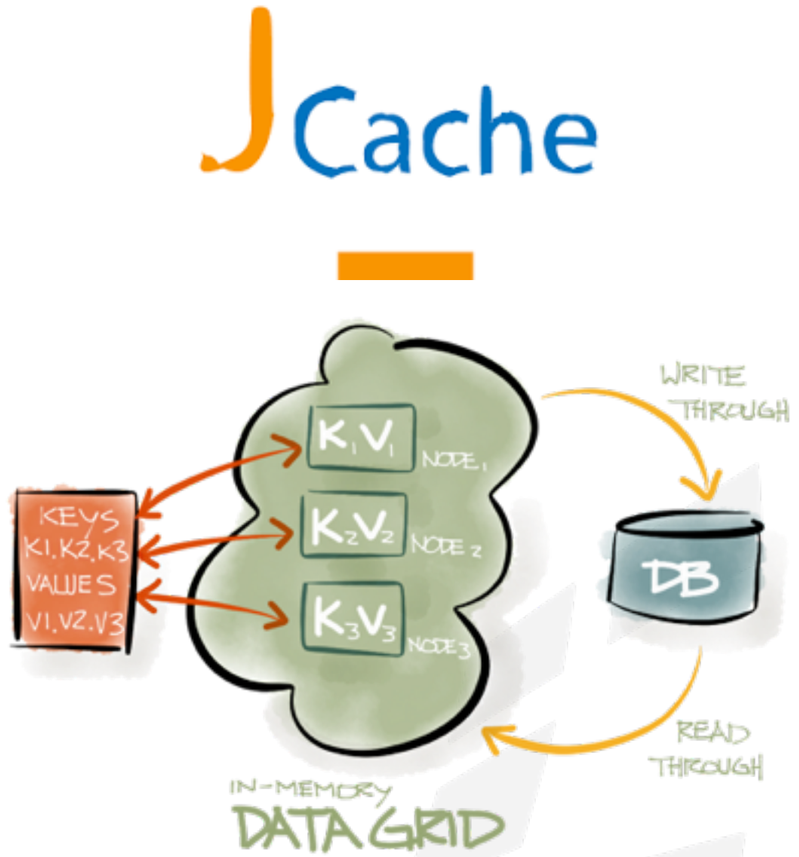
Apache Ignite: Better Cloud Support

- Automatic Cloud Discovery
  - Simple Configuration
  - AWS/EC2/S3
  - Google Compute Engine (NEW)
  - Other Clouds with JClouds (NEW)
- Docker Support
  - Automatically Build and Deploy

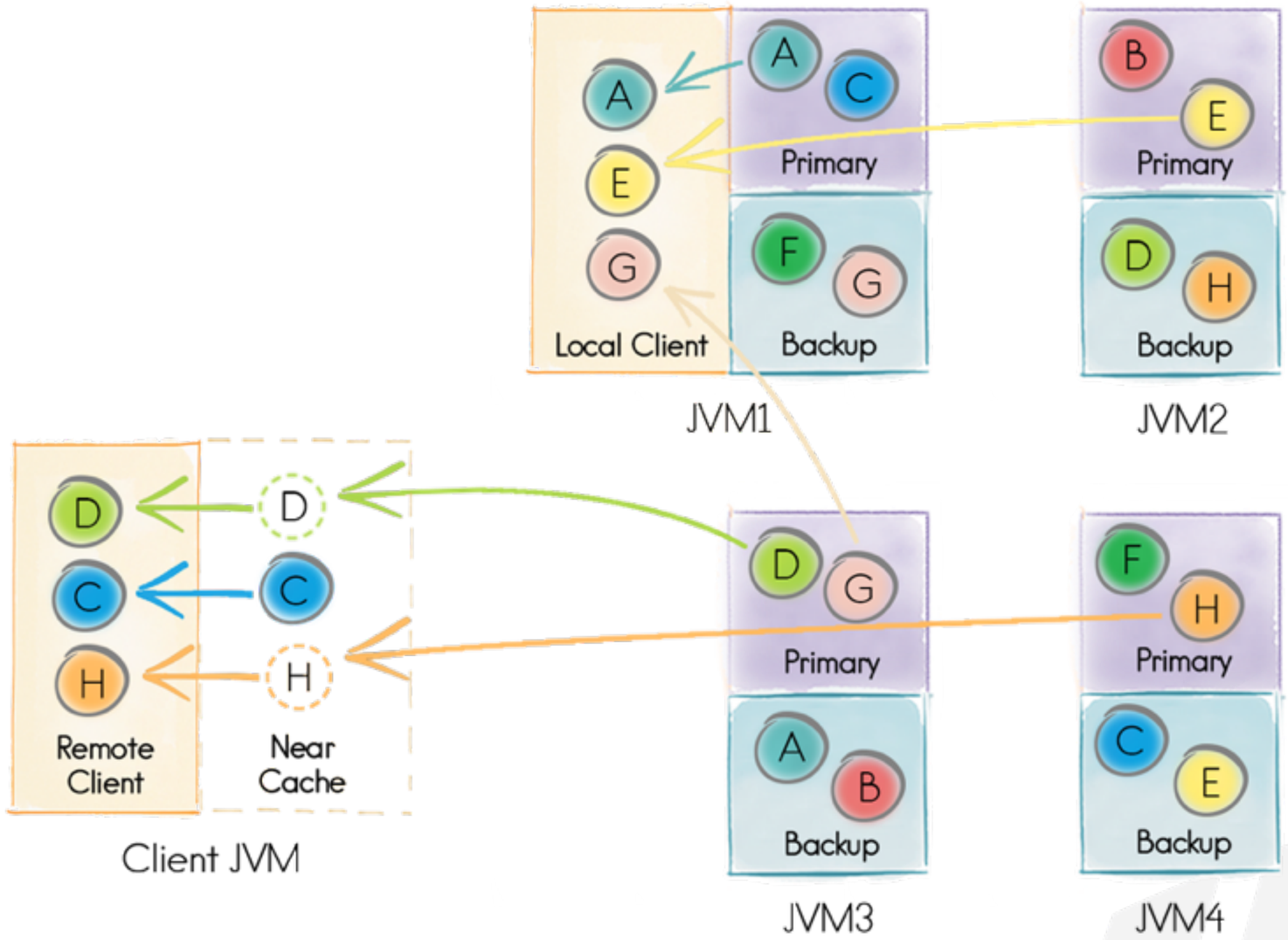


Data Grid: JCache (JSR 107)

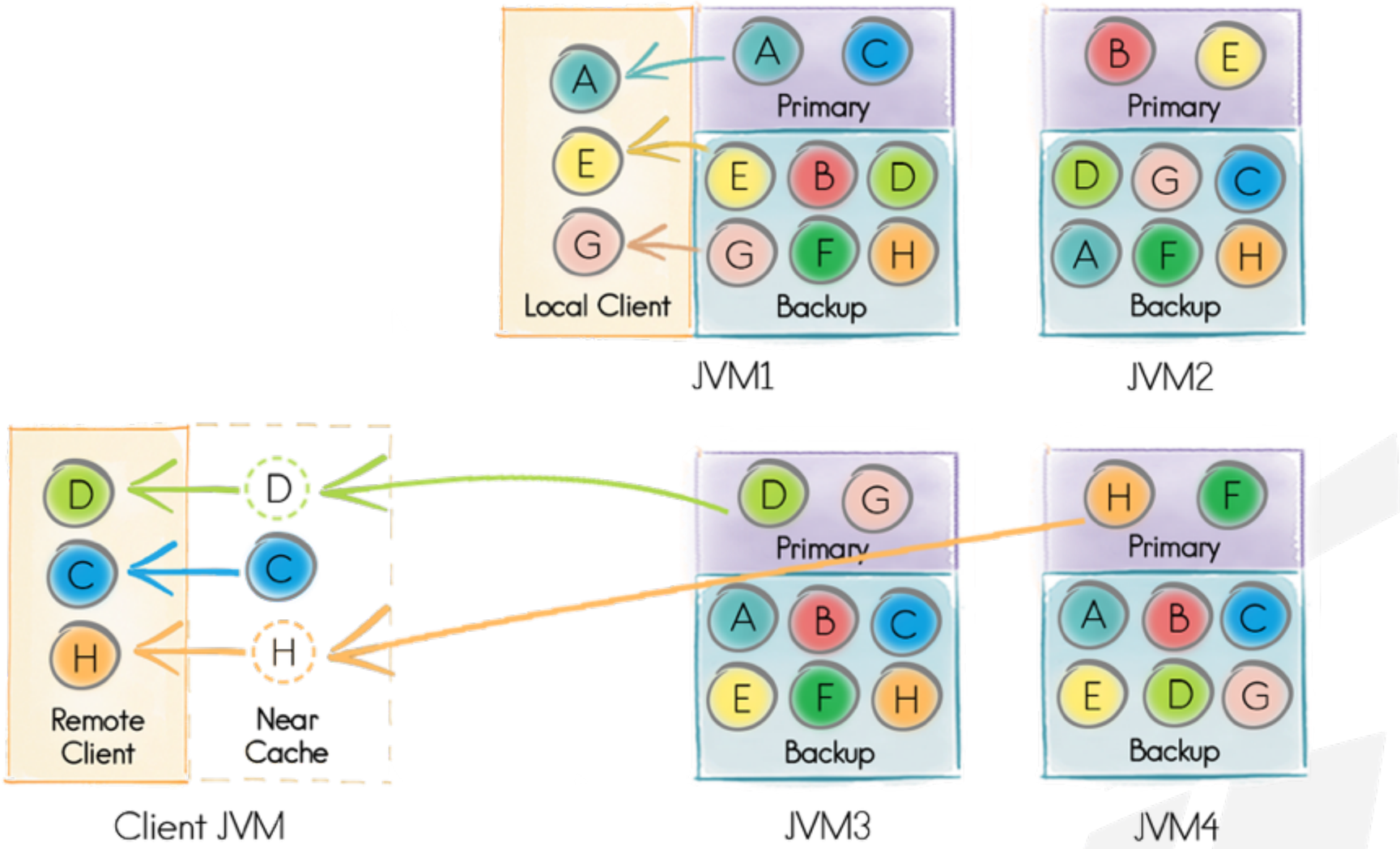
- JCache (JSR 107)
  - Basic Cache Operations
  - ConcurrentMap APIs
  - Collocated Processing (EntryProcessor)
  - Events and Metrics
  - Pluggable Persistence
- Ignite Data Grid
  - ACID Transactions
  - SQL Queries (ANSI 99)
  - In-Memory Indexes
  - Automatic RDBMS Integration



Data Grid: Partitioned Cache



Data Grid: Replicated Cache

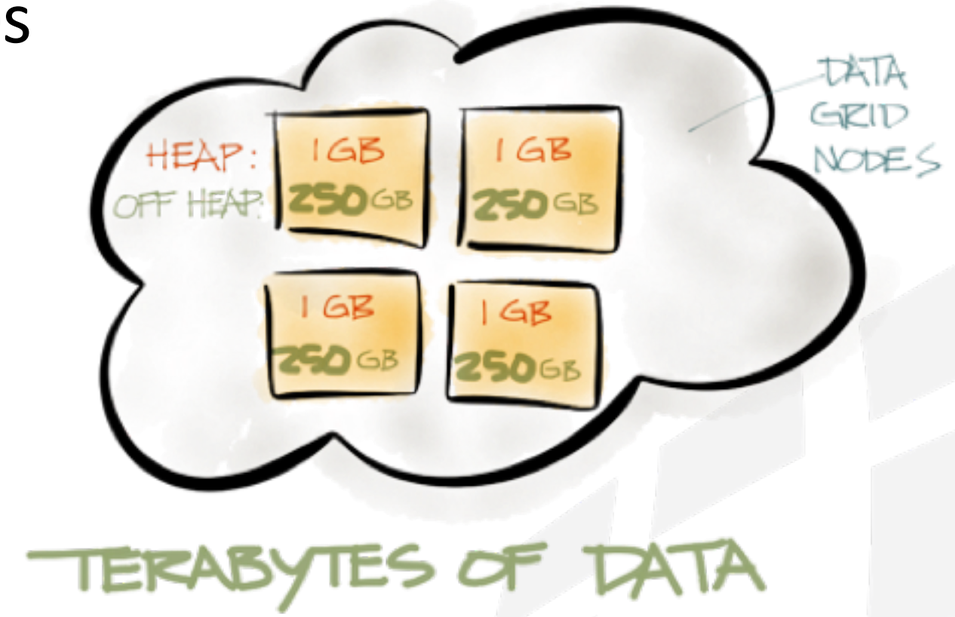


DevOps: Integration with Yarn and Mesos

- Automatic Resource Management
- Easy Data Center Installation
- Easy Data Center Configuration
- On-Demand Elasticity

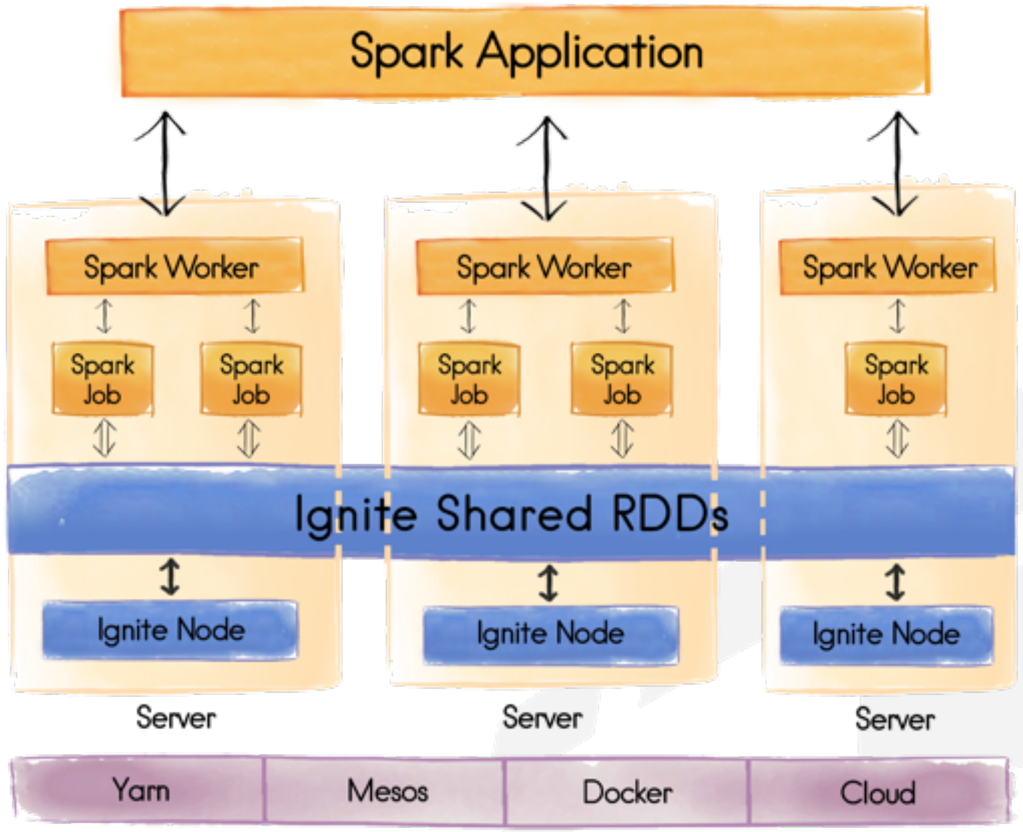
Data Grid: Off-Heap Memory

- Unlimited Vertical Scale
- Avoid Java Garbage Collection Pauses
- Small On-Heap Footprint
- Large Off-Heap Footprint
- Off-Heap Indexes
- Full RAM Utilization
- Simple Configuration



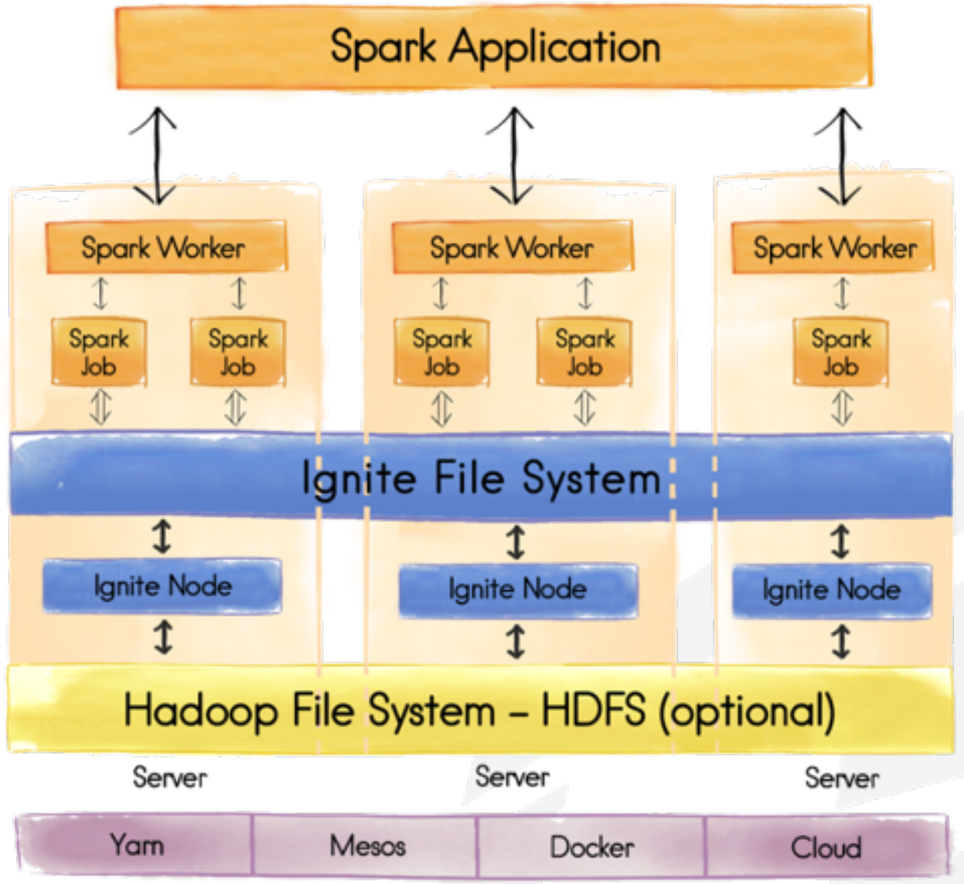
Share RDDs Across Spark Jobs

- IgniteRDD
  - Share RDD across jobs on the host
  - Share RDD across jobs in the application
  - Share RDD globally
- Faster SQL
  - In-Memory Indexes
  - SQL on top of Shared RDD



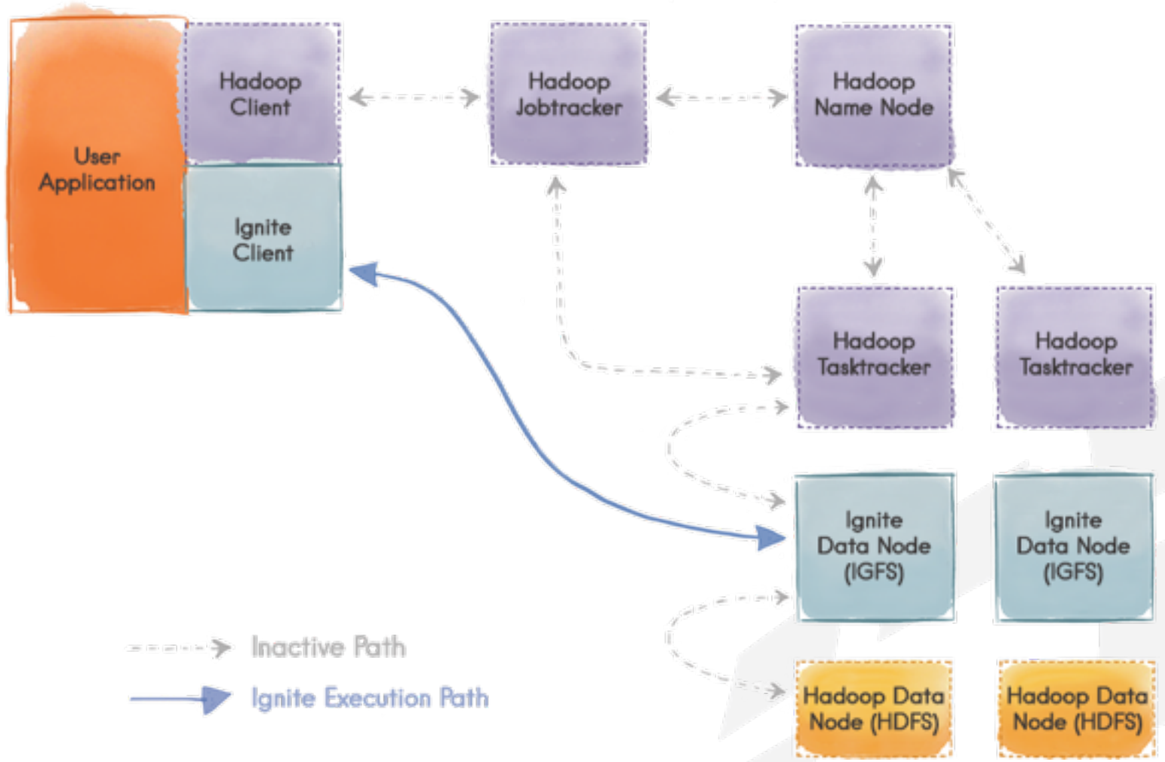
Ignite In-Memory File System

- Ignite In-Memory File System (IGFS)
  - Hadoop-compliant
  - Easy to Install
  - On-Heap and Off-Heap
  - Caching Layer for HDFS
  - Write-through and Read-through HDFS
  - Performance Boost



Ignite In-Memory Map Reduce

- In-Memory Native Performance
- Zero Code Change
- Use existing MR code
- Use existing Hive queries
- No Name Node
- No Network Noise
- In-Process Data Colocation
- Eager Push Scheduling



Data Grid: Ad-Hoc SQL (ANSI 99)

- ANSI-99 SQL
- Always Consistent
- Fault Tolerant
- In-Memory Indexes (On-Heap and Off-Heap)
- Automatic Group By, Aggregations, Sorting
- Cross-Cache Joins, Unions, etc.
- Ad-Hoc SQL Support

SQL Cross-Cache JOIN Example

```
IgniteCache<AffinityKey<UUID>, Person> cache = ignite.cache("persons");

// Execute query to get names of all employees.
SqlFieldsQuery qry = new SqlFieldsQuery(
    "select concat(firstName, ' ', lastName), org.name " +
    "from Person, \"Organizations\".Organization as org " +
    "where Person.orgId = org.id");

QueryCursor<List<?>> cursor = cache.query(qry);

for (List<?> row : cursor)
    print(row);
```

SQL Cross-Cache GROUP BY Example

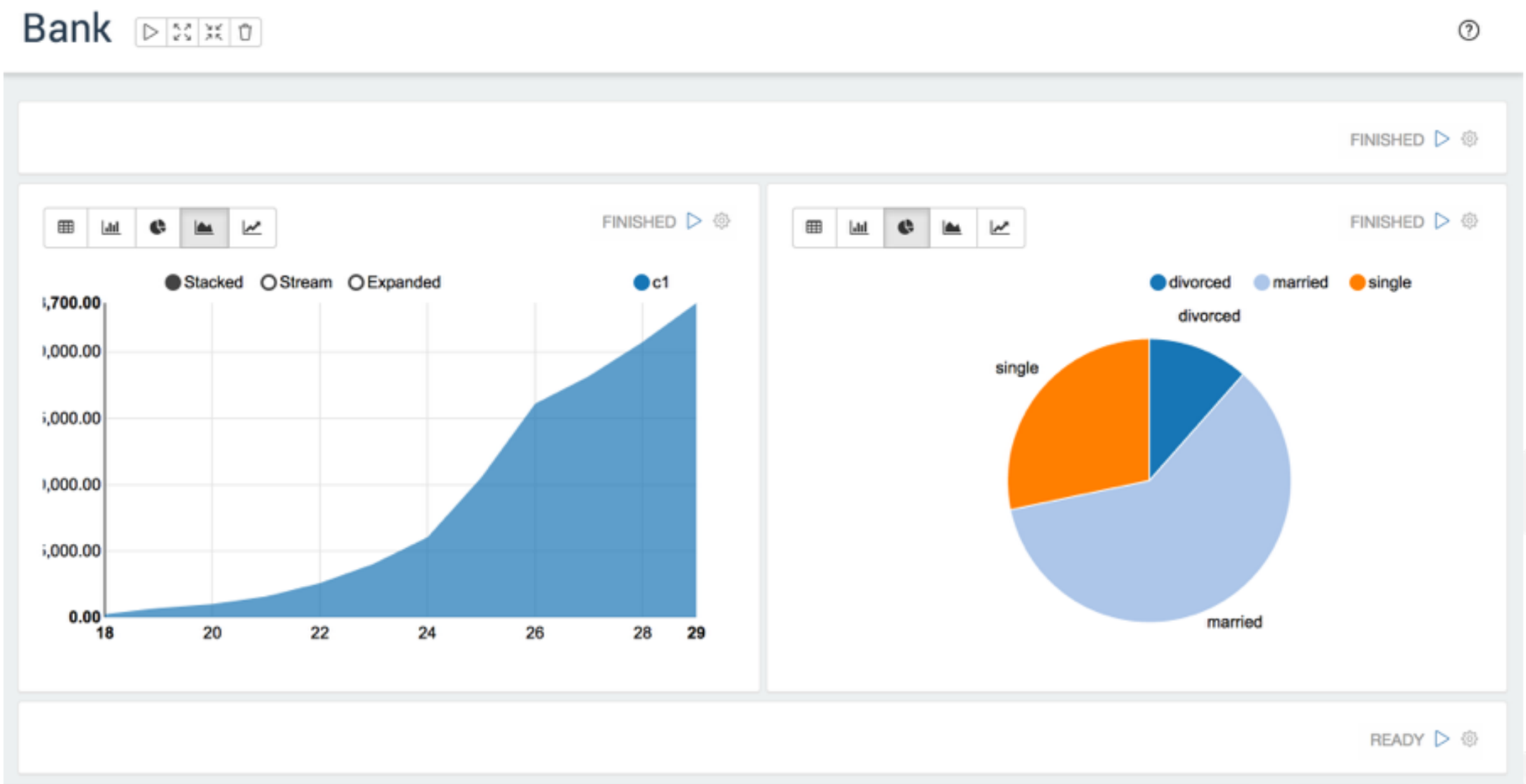
```
IgniteCache<AffinityKey<UUID>, Person> cache = ignite.cache("persons");

// Query to get salaries grouped by organization.
SqlFieldsQuery qry = new SqlFieldsQuery(
    "select org.name, avg(salary), max(salary), min(salary) " +
        "from Person, \"Organizations\".Organization as org " +
        "where Person.orgId = org.id " +
        "group by org.name " +
        "order by org.name");

QueryCursor<List<?>> cursor = cache.query(qry);

List<List<?>> res = cursor.getAll();
```

Interactive SQL with Apache Zeppelin



ANY QUESTIONS?

Thank you for joining us. Follow the conversation.

http://www.ignite.incubator.apache.org

 @apacheignite  @dsetrakyan