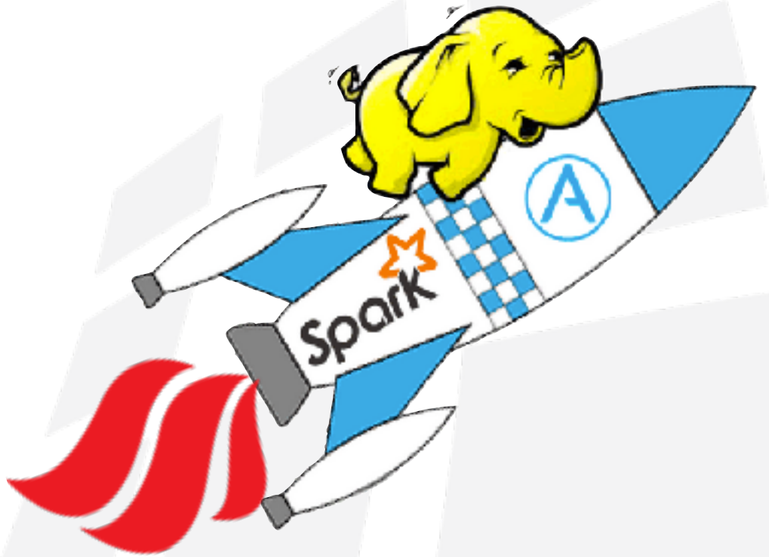
Better Together: Fast Data with Ignite & Spark

Christos Erotocritou - Spark Summit EU 2016

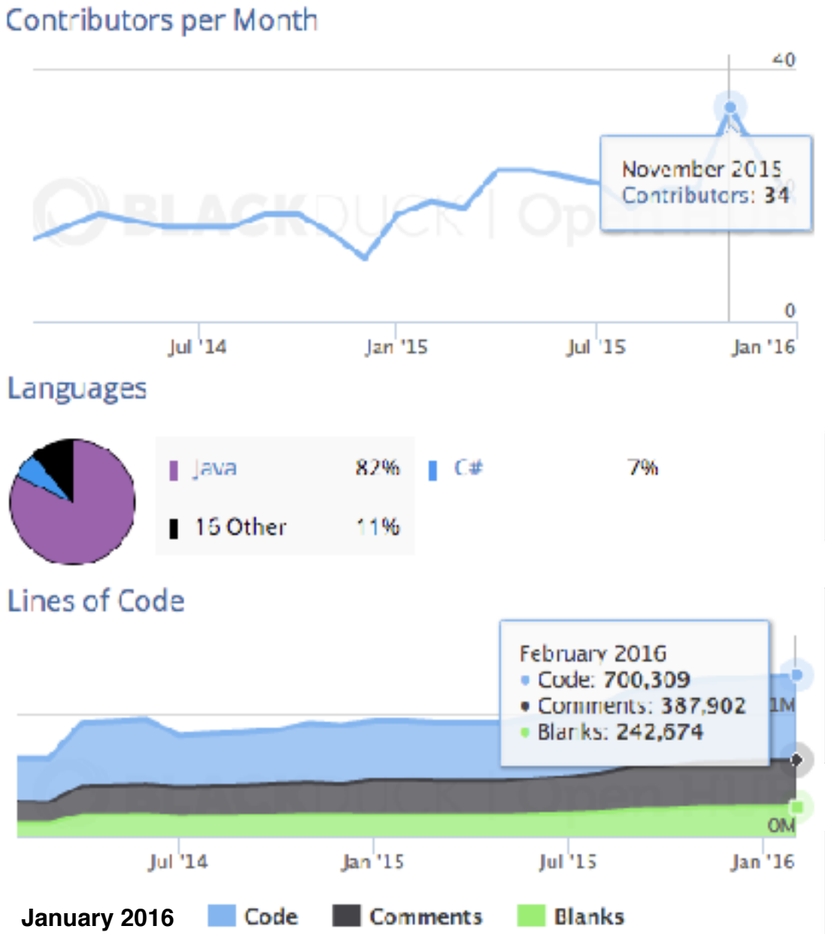
Agenda

- GridGain & Apache Ignite Project
- Ignite In-Memory Data Fabric
- Apache Ignite vs. Apache Spark
- Hadoop & Spark Integration
- Q & A



Apache Ignite Project

- **2007:** First version of GridGain
- **Oct. 2014:** GridGain contributes Ignite to ASF
- **Aug. 2015:** Ignite is the second fastest project to graduate after Spark
- **Today:**
  - 82+ contributors and growing rapidly
  - Huge development momentum - Estimated 233 years of effort since the first commit in February, 2014 [Openhub]
  - Mature codebase: 840k+ SLOC & more than 17k commits



- GridGain Enterprise Edition
  - Is a binary build of Apache Ignite™ created by GridGain
  - Added enterprise features for enterprise deployments
  - Earlier features and bug fixes by a few weeks
  - Heavily tested

Customer Use Cases

Automated Trading Systems

Real time analysis of trading positions & market risk. High volume transactions, ultra low latencies.

Financial Services

Fraud Detection, Risk Analysis, Insurance rating and modelling.

Online & Mobile Advertising

Real time decisions, geo-targeting & retail traffic information.

Big Data Analytics

Customer 360 view, real-time analysis of KPIs, up-to-the-second operational BI.

Online Gaming

Real-time back-ends for mobile and massively parallel games.

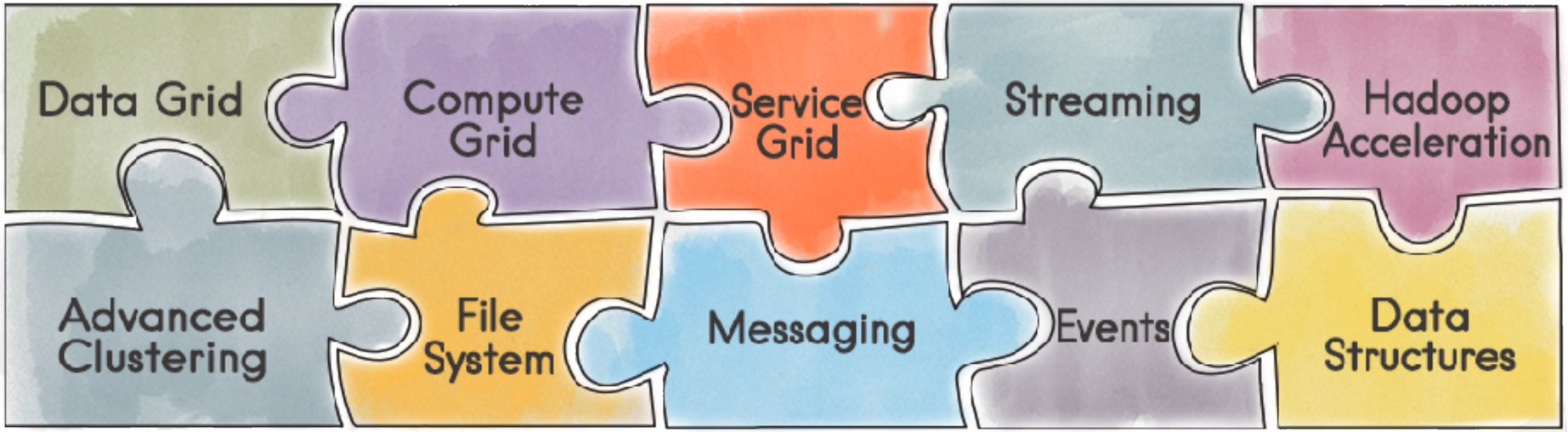
SaaS Platforms & Apps

High performance next-generation architectures for Software as a Service Application vendors.

Travel & E-Commerce

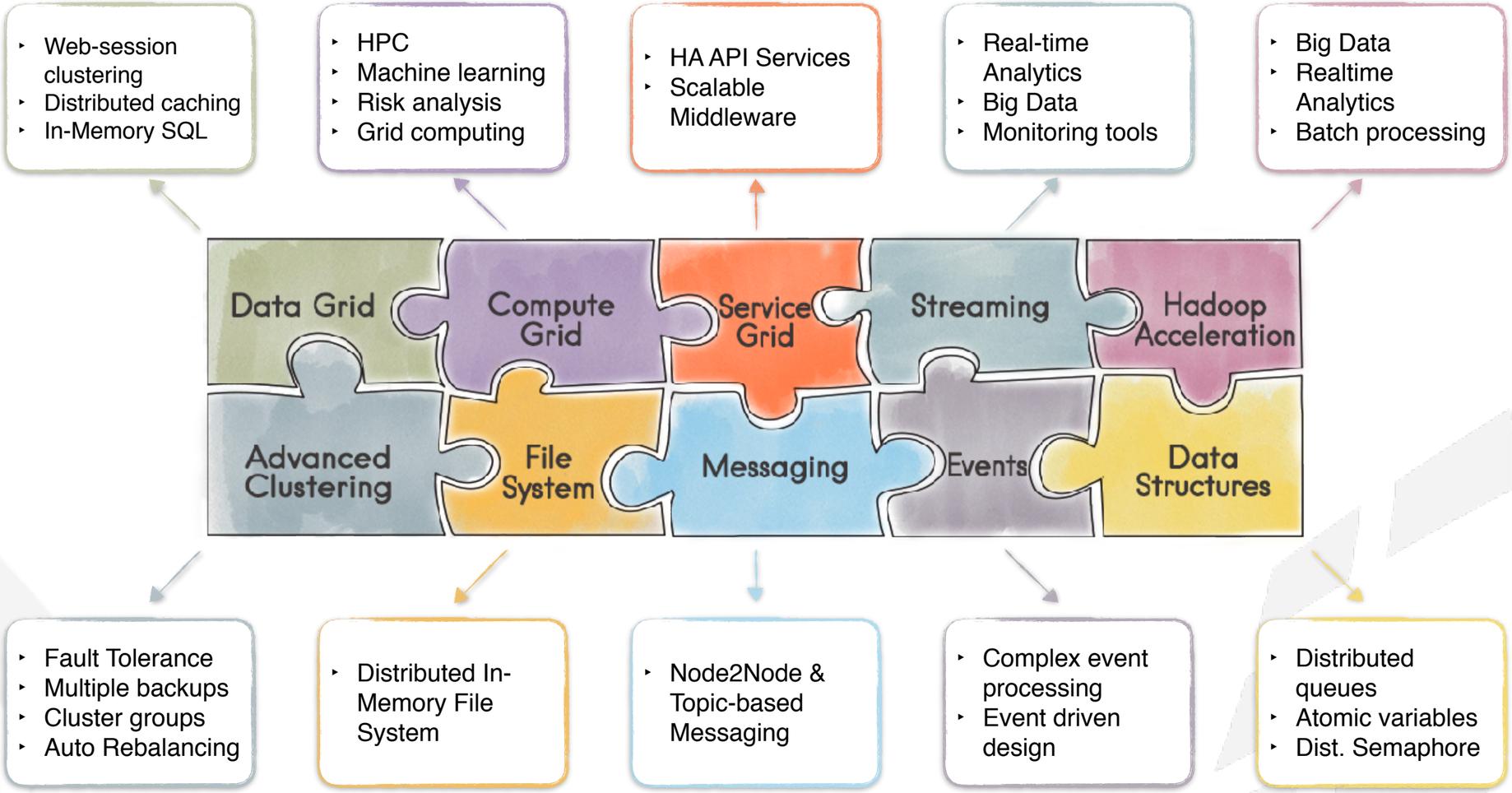
High performance next-generation architectures for online hotel booking.

What is an IMDF?

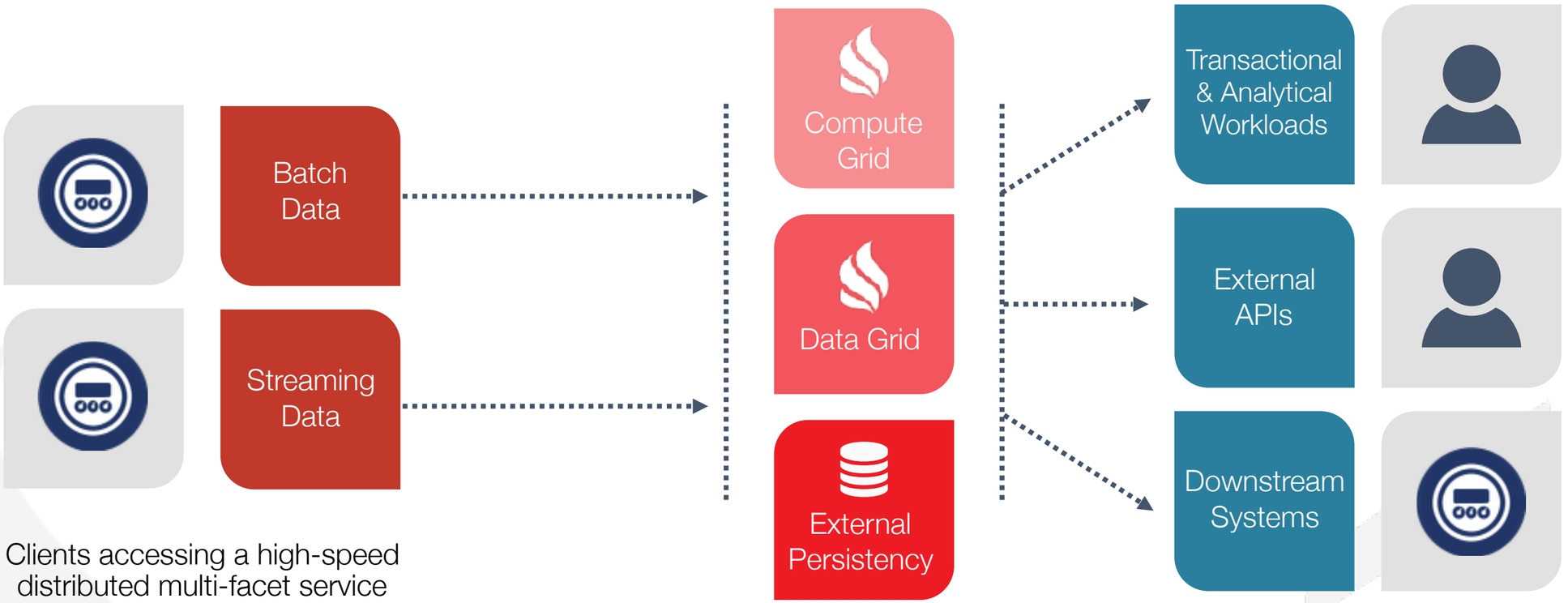


High-performance distributed in-memory platform for computing and transacting on large-scale data sets in near real-time.

What is an IMDF?



In-Memory Computing Platform

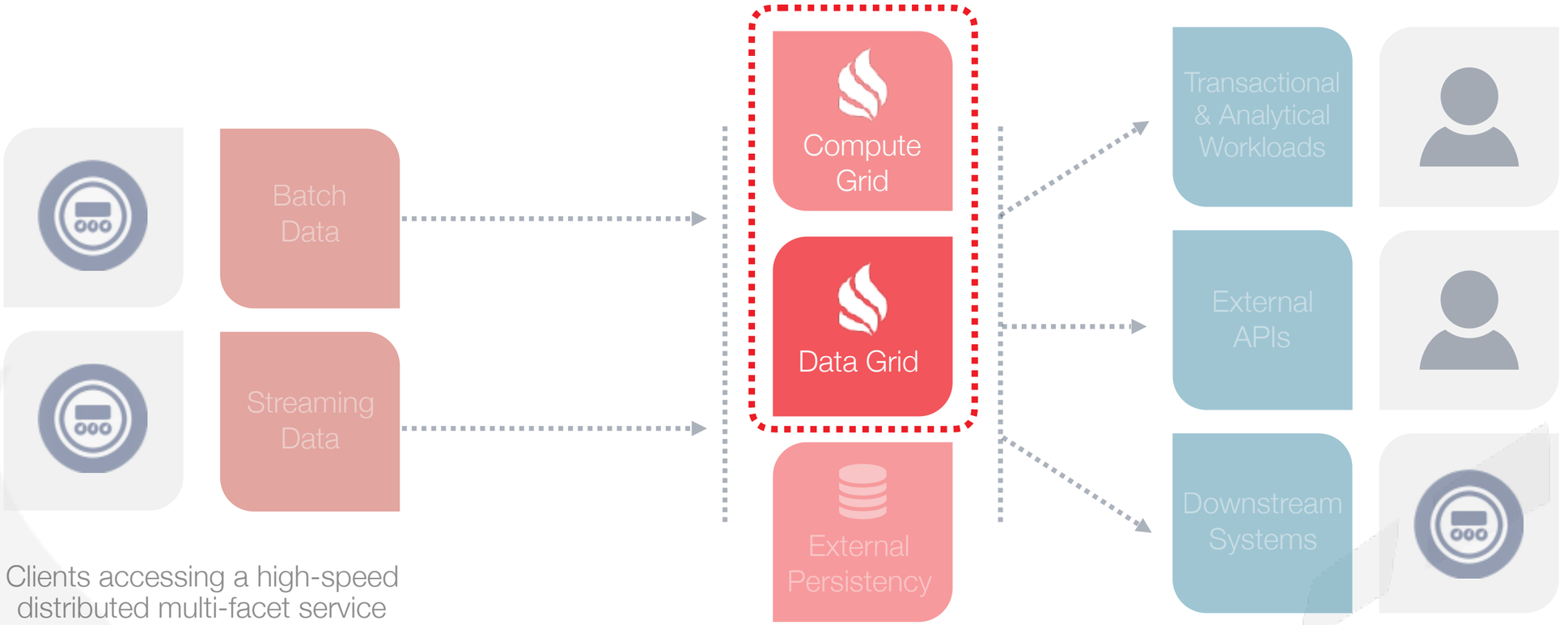


Clients accessing a high-speed distributed multi-facet service

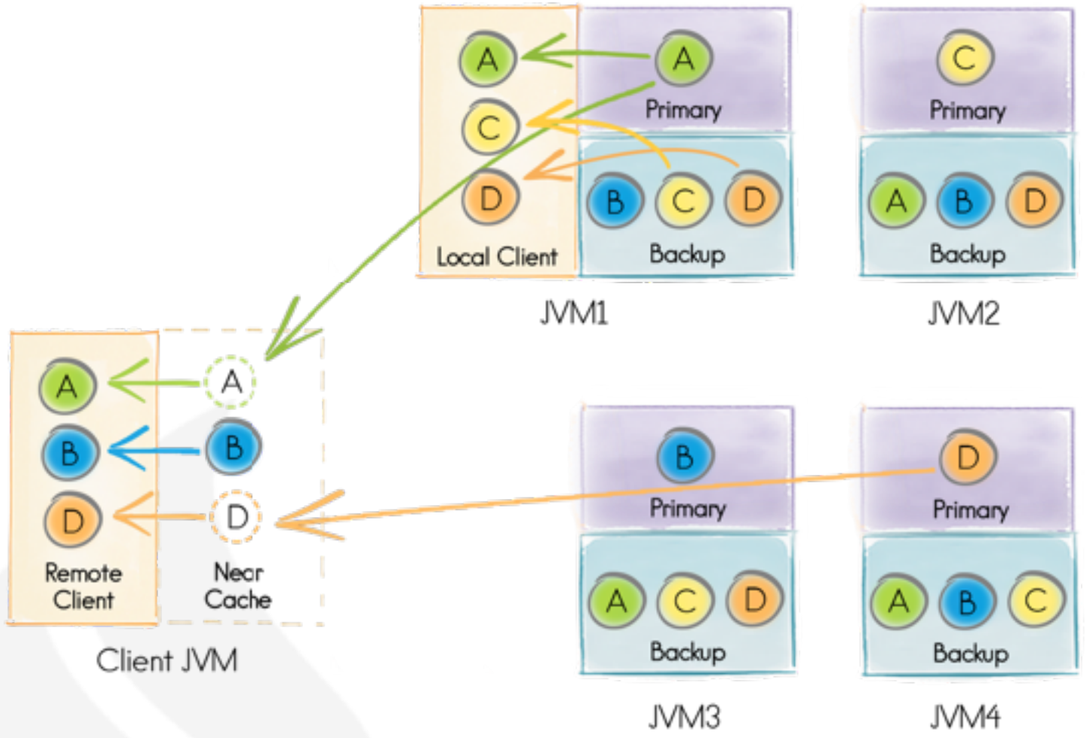
Transactional & Analytical workloads

Back-end users, third-party clients and downstream systems

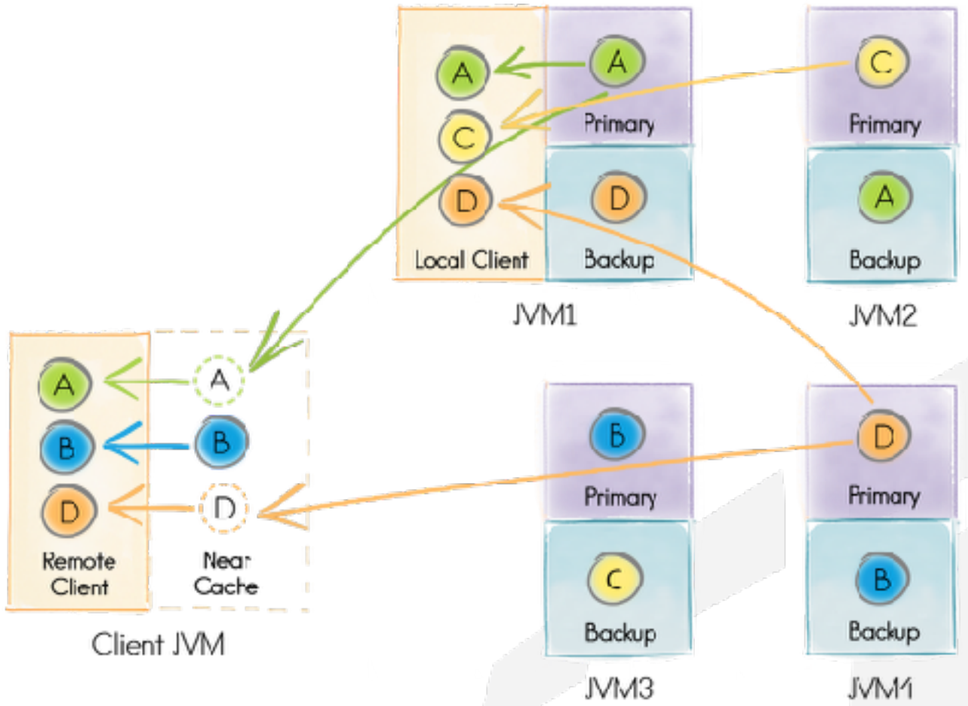
Scalability & Resilience with Ignite



Fault Tolerance & Horizontal Scalability



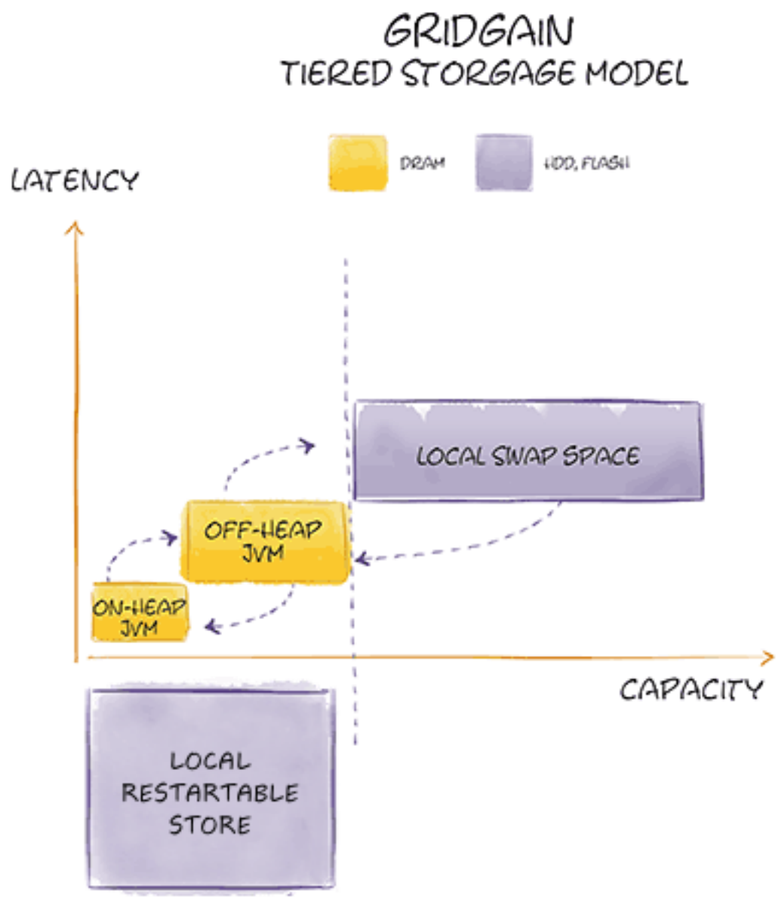
Replicated Cache



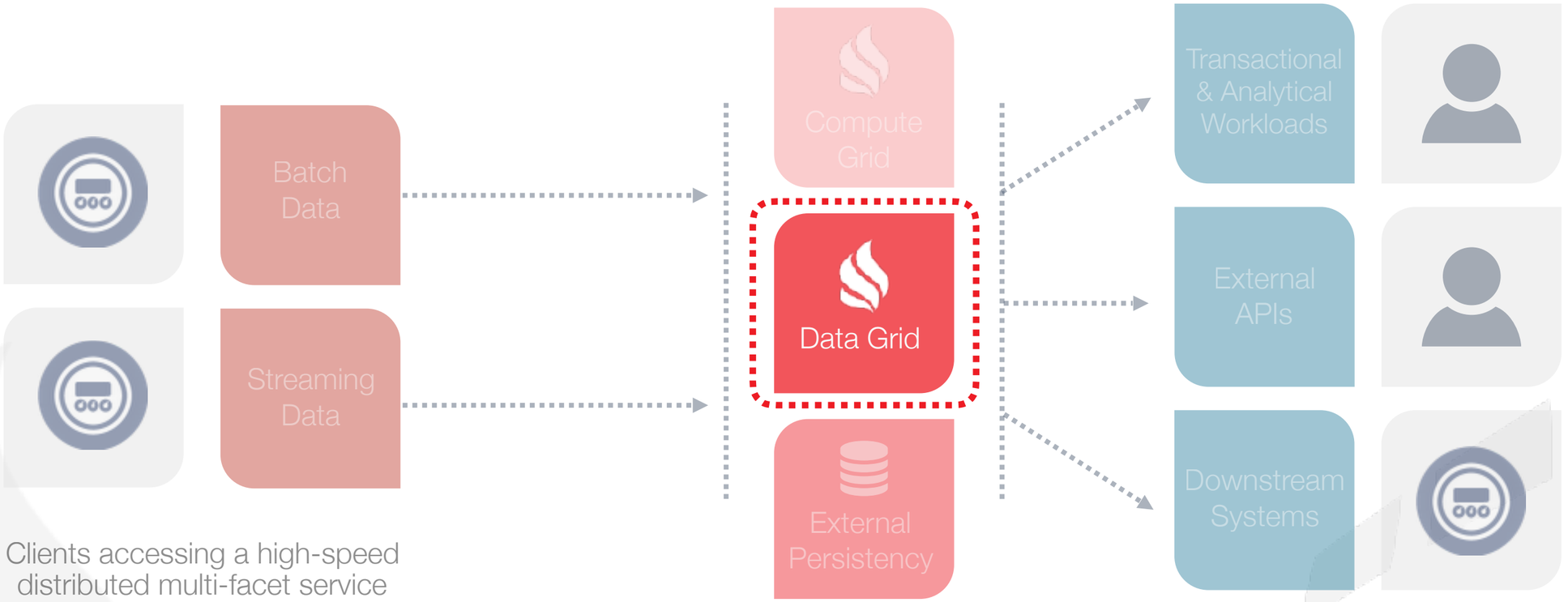
Partitioned Cache

Local Store & Vertical Scale

- Tiered Memory
  - On-Heap -> Off-Heap -> Disk
- Persistent On-Disk Store
- Fast Recovery
- Local Data Reload
  - Eliminate Network and Db impacts when reloading in-memory store

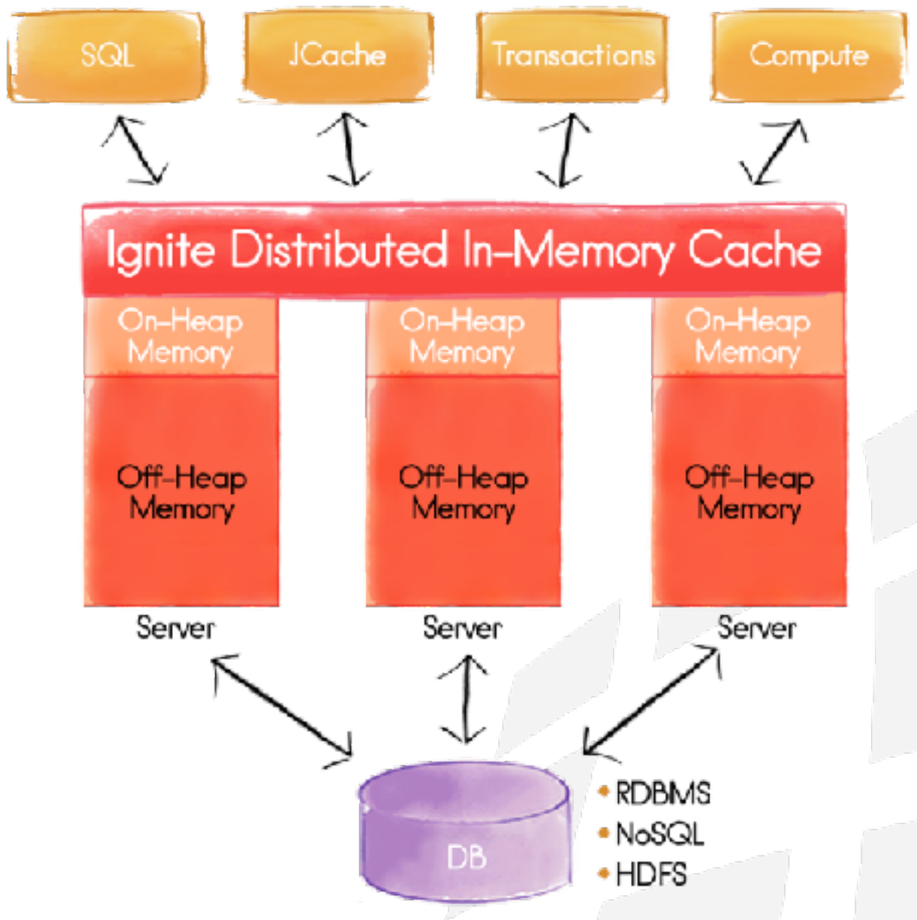


Storage and Caching using Ignite

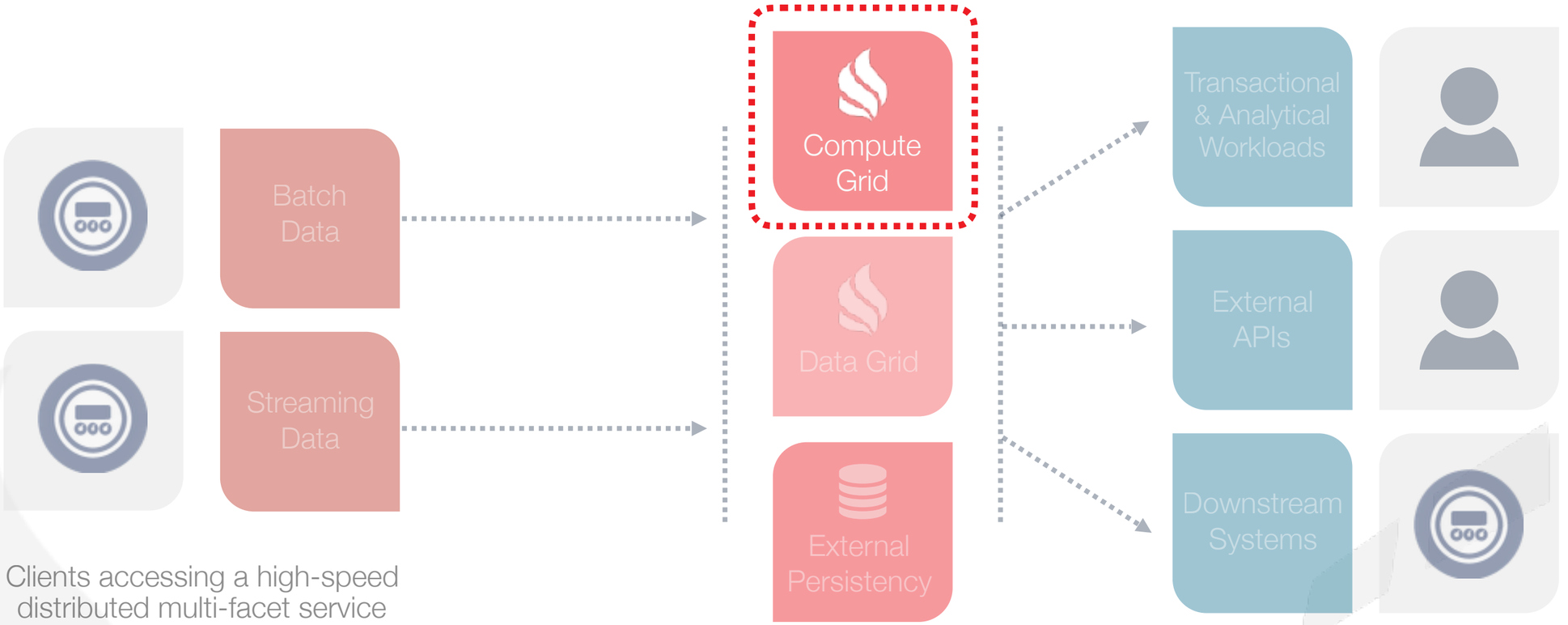


In-Memory Data Grid

- 100% JCache Compliant (JSR 107)
  - Basic Cache Operations
  - Concurrent Map APIs
  - Collocated Processing (EntryProcessor)
  - Events and Metrics
  - Pluggable Persistence
- Ignite Data Grid
  - Fault Tolerance and Scalability
  - Distributed Key-Value Store
  - SQL Queries (ANSI 99)
  - ACID Transactions
  - In-Memory Indexes
  - RDBMS / NoSQL Integration



Distributed Computing with Ignite

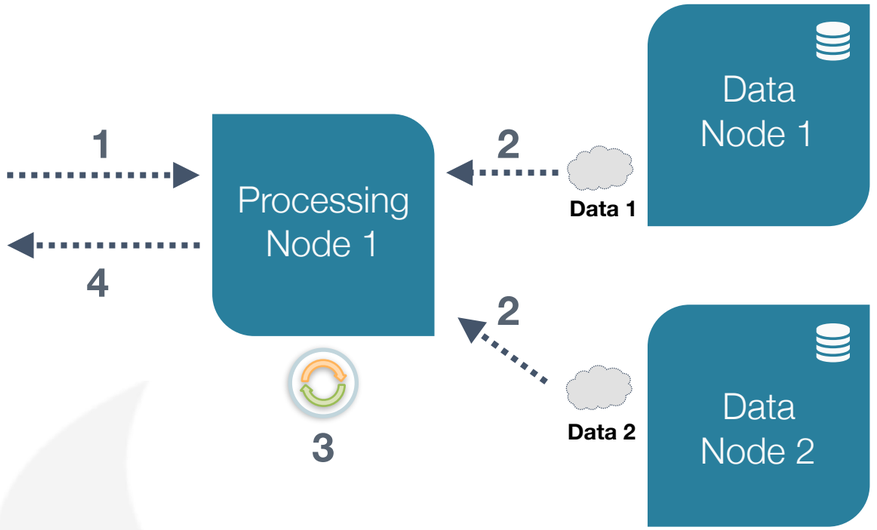


Clients accessing a high-speed distributed multi-facet service

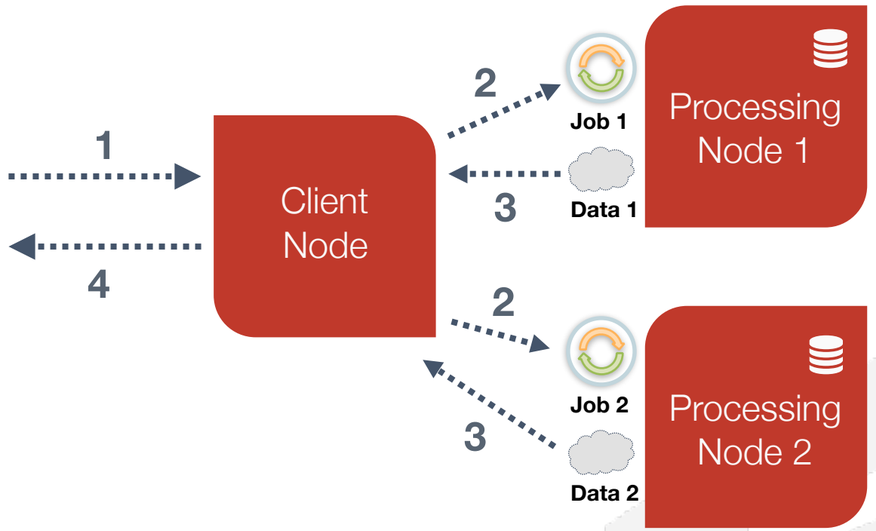
Transactional & Analytical workloads

Back-end users, third-party clients and downstream systems

Client-Server vs. Affinity Colocation



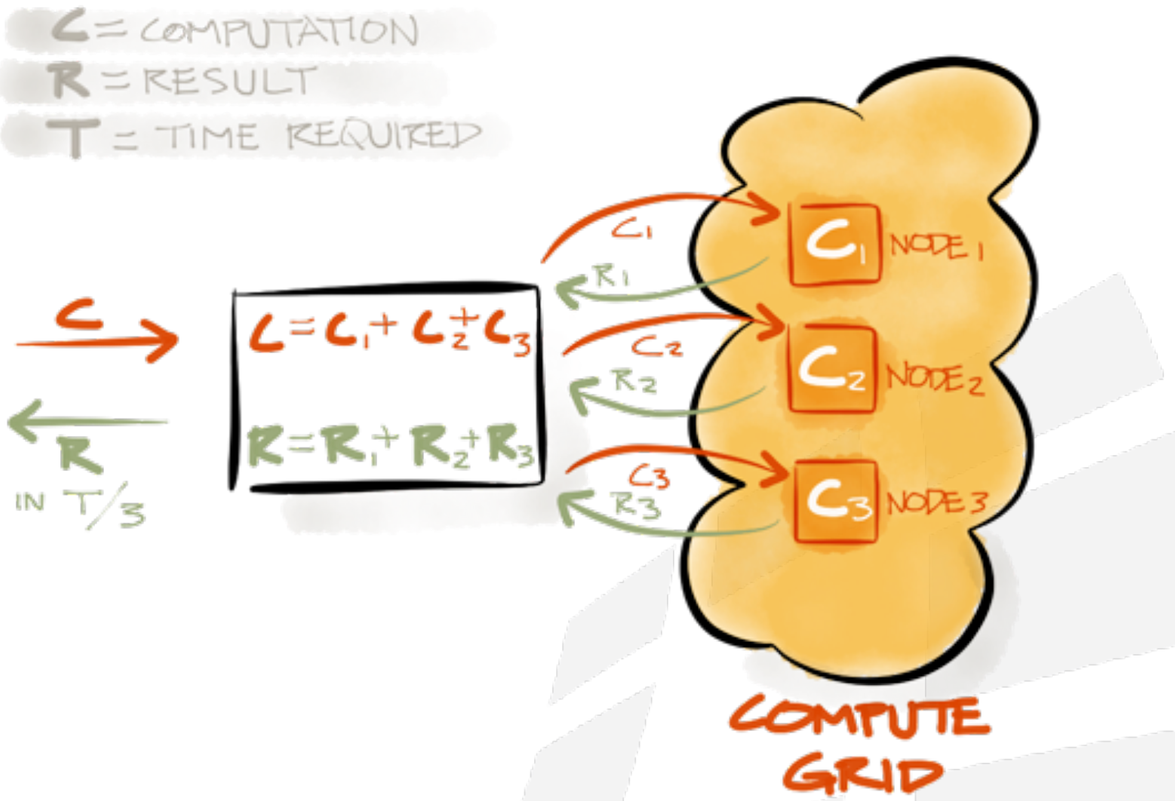
1. Initial Request
2. Fetch data from remote nodes
3. Process entire data-set
4. Return to client



1. Initial Request
2. Co-locating processing with data
3. Return partial result
4. Reduce & return to client

In-Memory Compute Grid

- Direct API for MapReduce
- Cron-like Task Scheduling
- State Checkpoints
- Load Balancing
  - Round-robin
  - Random & weighted
- Automatic Failover
- Per-node Shared State
- Zero Deployment
  - Distributed class loading



Hadoop & Spark Integration

Apache Ignite

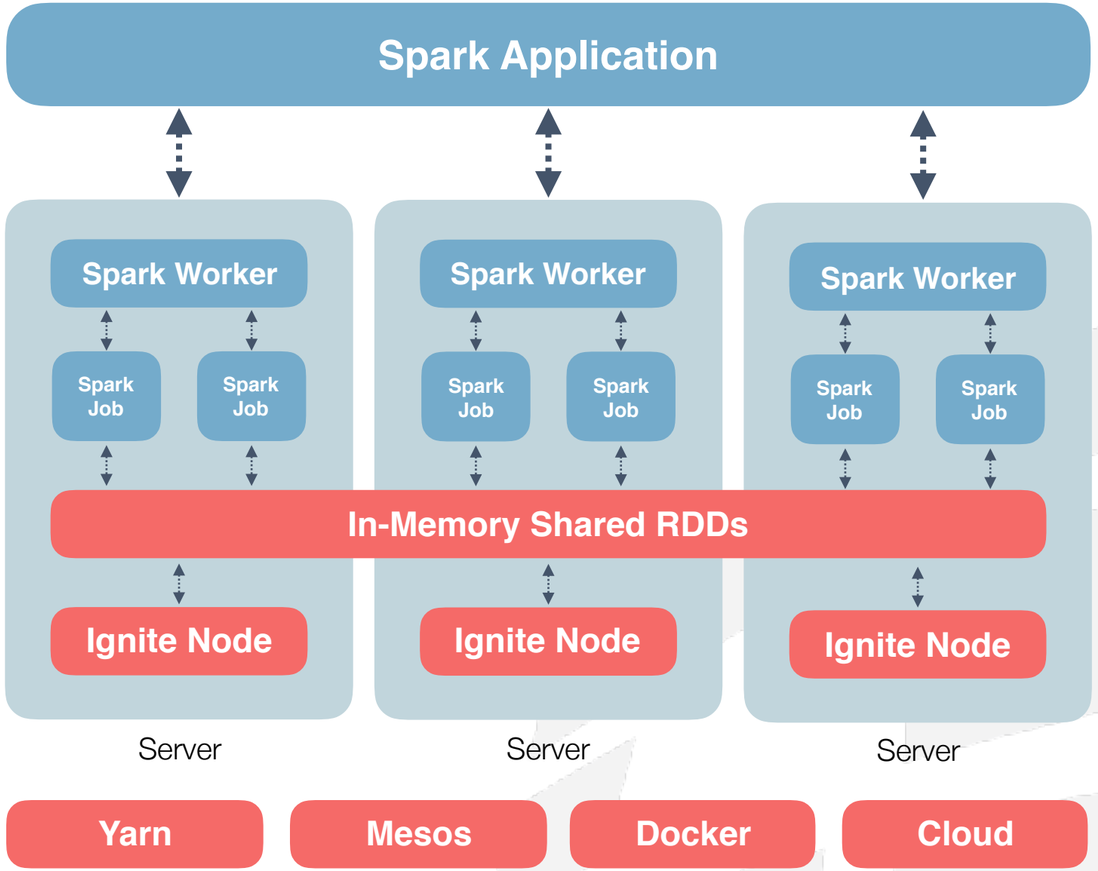
- Data source agnostic
- Fully fledged compute engine and resilient data storage in-memory for OLAP & OLTP
- Zero-deployment
- In-Memory SQL support
- Fully ACID transactions across memory and disk
- Broader in-memory system that is less focused on Hadoop
- Off-heap memory to avoid GC pauses
- In production since 2007

Apache Spark

- Ingests data from HDFS or another distributed file system
- Inclined towards analytics (OLAP) and focused on MR-specific payloads
- Requires the creation of RDD and data and processing operations are governed by it
- Basic disk-based SQL support
- Strong ML libraries
- Big community

Spark & Ignite Integration

- 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



Spark & Ignite Integration: IgniteRDD

- IgniteContext is the main entry point to Spark-Ignite integration:

```
val igniteContext = new IgniteContext[Integer, Integer]
   (sparkContext, () => new IgniteConfiguration())
```

- Reading values from Ignite:

```
val cache = igniteContext.fromCache("myRdd")
val result = cache.filter(_._2.contains("Ignite")).collect()
```

- Saving values to Ignite:

```
val cacheRdd = igniteContext.fromCache("myRdd")
  cacheRdd.savePairs(sparkContext.parallelize(1 to 10000, 10).map(i => (i, i)))
```

- Running SQL queries against Ignite Cache:

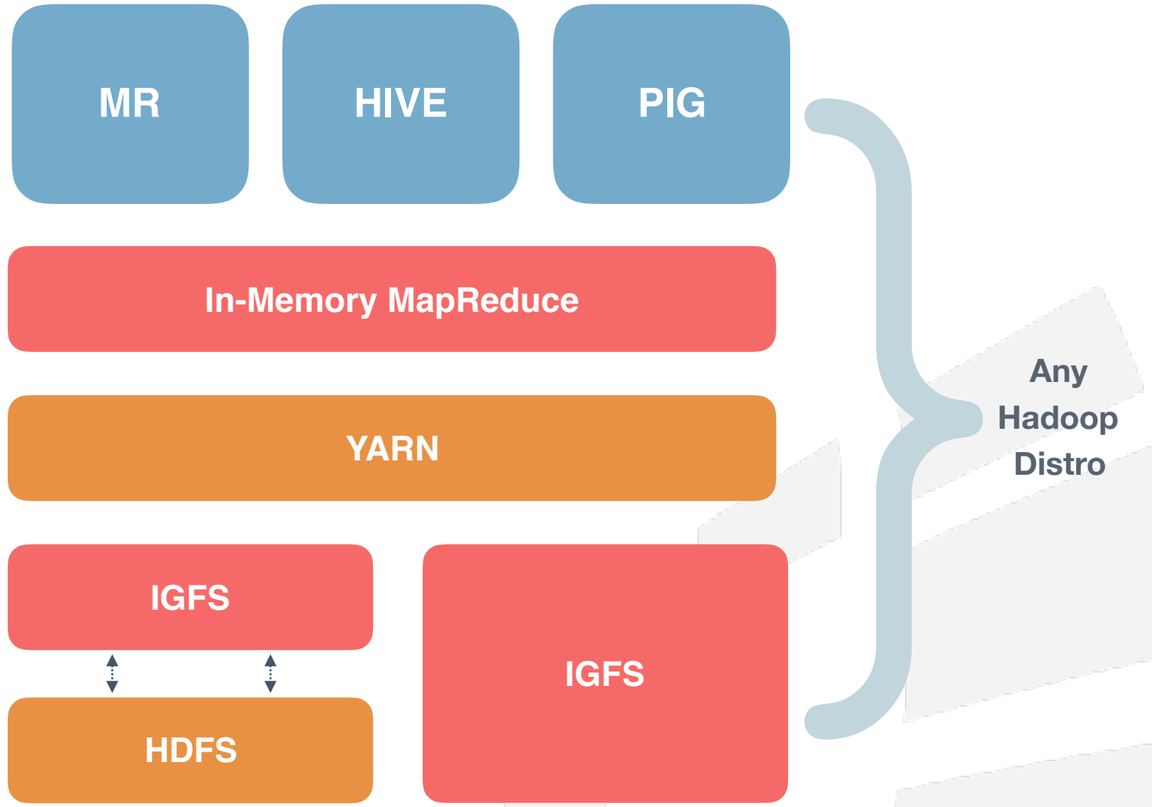
```
val cacheRdd = igniteContext.fromCache("myRdd")
val result = cacheRdd.sql
   ("select _val from Integer where val > ? and val < ?", 10, 100)
```

Spark Integration: Using Dataframes from IgniteRDDs

```
// Create an IgniteRDD

val companyCacheIgnite = new IgniteContext[Int, String](sc, () =>
new IgniteConfiguration()).fromCache("CompanyCache")

// Create company DataFrame

val dfCompany = sqlContext.createDataFrame(companyCacheIgnite.map(p =>
Company(p._1, p._2)))

// Register DataFrame as a table

dfCompany.registerTempTable("company")
```

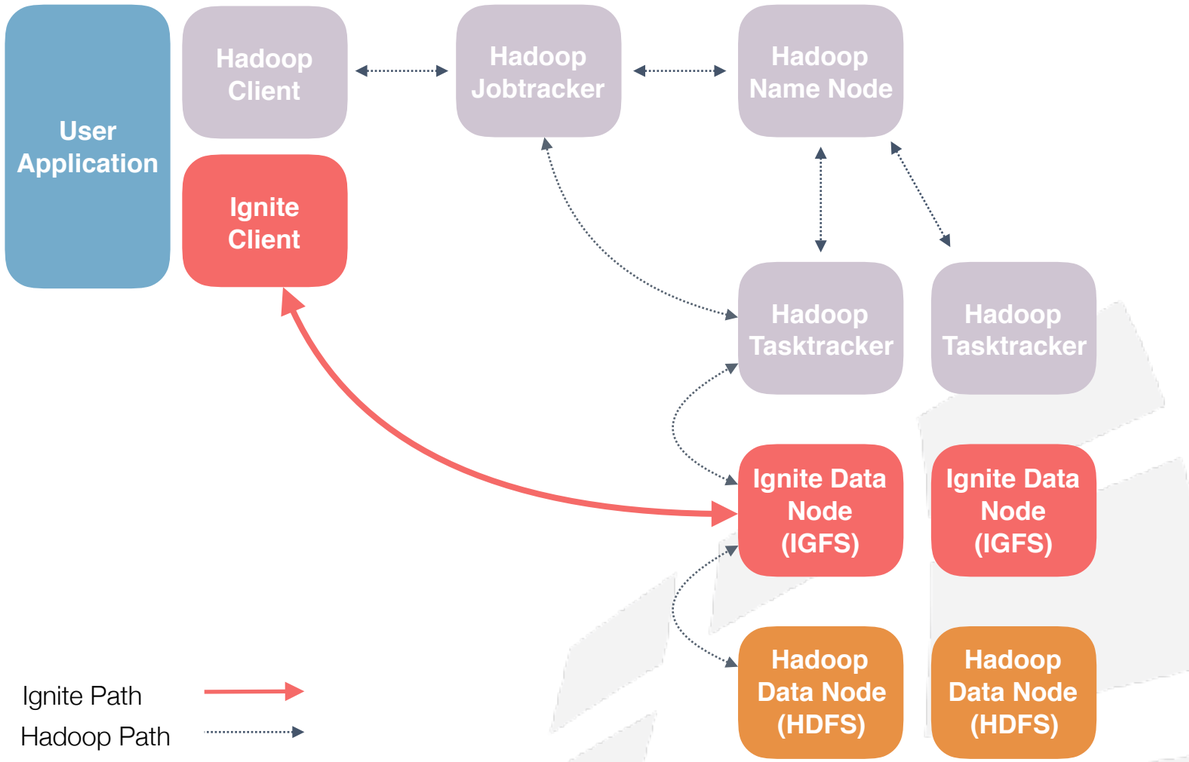
IGFS: In-Memory File System

- 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



Hadoop Accelerator: Map Reduce

- In-Memory 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



Deployment

- Docker
- Amazon AWS
- Azure Marketplace
- Google Cloud
- Apache JClouds
- Mesos
- YARN
- Apache Karaf (OSGi)

Thank You!

Author: Christos Erotocritou

github.com/kemiz/SparkIgniteSimpleExample

Thank you for joining us. Follow the conversation.

www.gridgain.com

@gridgain @ApacheIgnite
#gridgain #ApacheIgnite