GridGain

Powered by Apache Ignite

In-Memory Computing Platform:

Data Grid Deep Dive

Rachel Pedreschi
Director of Solutions Architecture
GridGain Systems
rachel@gridgain.com
@rachelpedreschi

Matt Sarrel
Director of Technical Marketing
GridGain Systems
matt.sarrel@gridgain.com
@msarrel

Agenda

- Introduction
- In-Memory Computing
- GridGain / Apache Ignite Overview
- Survey Results
- Data Grid Deep Dive
- Customer Case Studies

Why In-Memory Now?

Digital Transformation is Driving Companies Closer to Their Customers

- Driving a need for real-time interactions

Internet Traffic, Data, and Connected Devices Continue to Grow

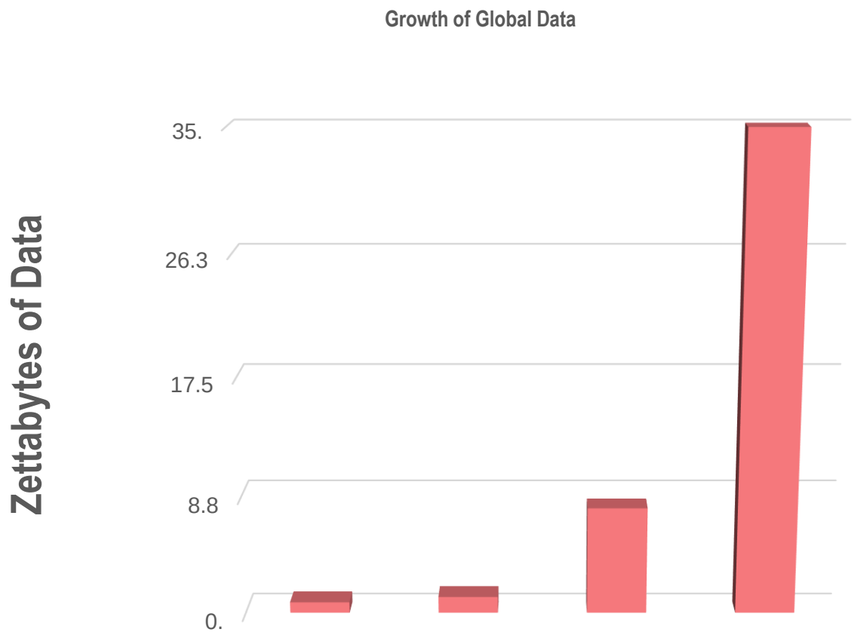
- Web-scale applications and massive datasets require in-memory computing to scale out and speed up to keep pace
- The Internet of Things generates huge amounts of data which require real-time analysis for real world uses

The Cost of RAM Continues to Fall

- In-memory solutions are increasingly cost effective versus disk-based storage for many use cases

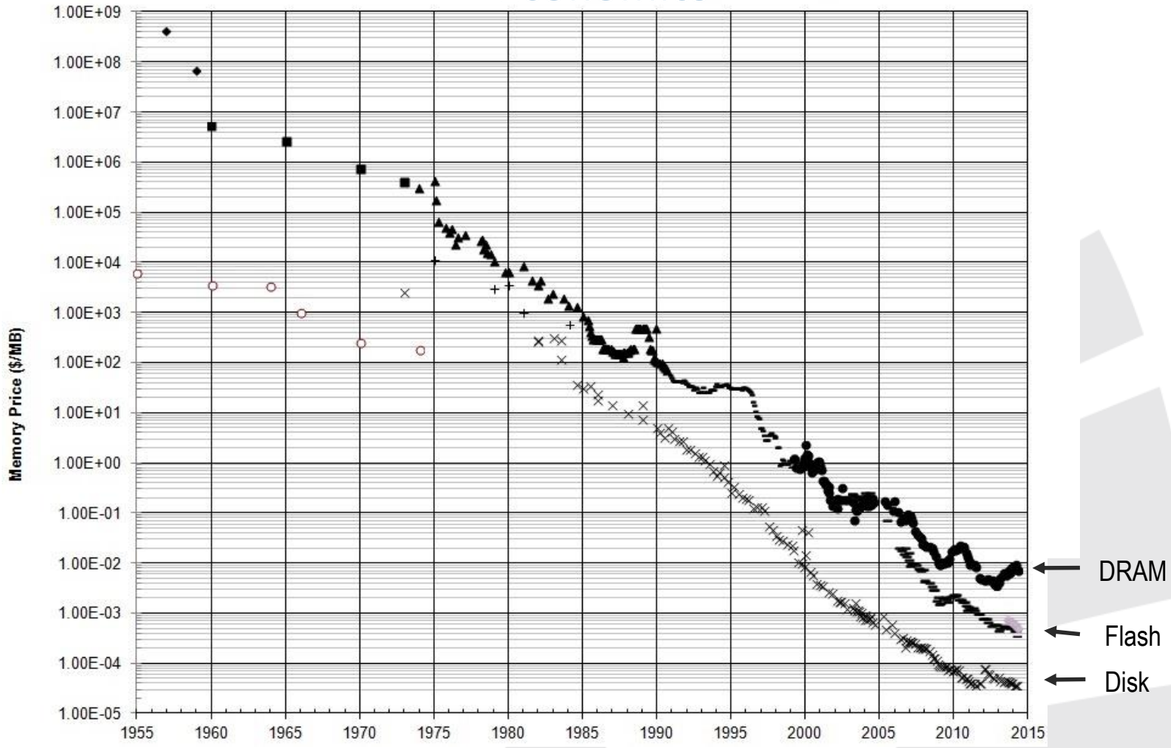
Why Now?

Data Growth and Internet Scale Driving Demand



8 zettabytes in 2015 growing to 35 in 2020

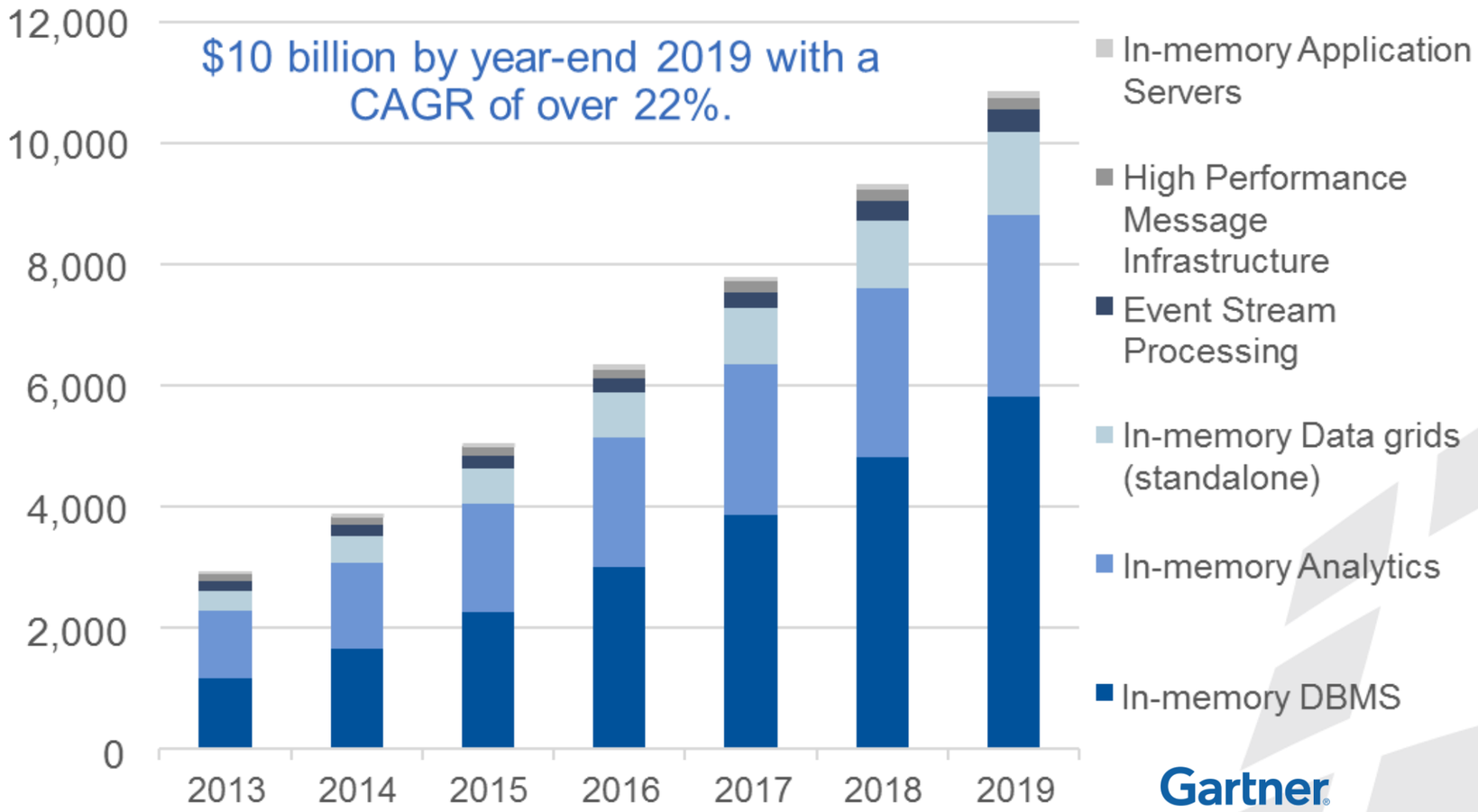
Declining DRAM Cost Driving Attractive Economics



Cost drops 30% every 12 months

The In-Memory Computing Technology Market Is Big — And Growing Rapidly

IMC-Enabling Application Infrastructure ($M)



Apache Ignite GridGain

What is an In-Memory Computing Platform?

Multi-Featured Solution

- Supports data caching, massive parallel processing, in-memory SQL, streaming and much more

Does Not Replace Existing Databases

- Slides in between the existing application and data layers

Supports OLTP and OLAP Use Cases

- Offers ACID compliant transactions as well as analytics support

Multi-Platform Integration

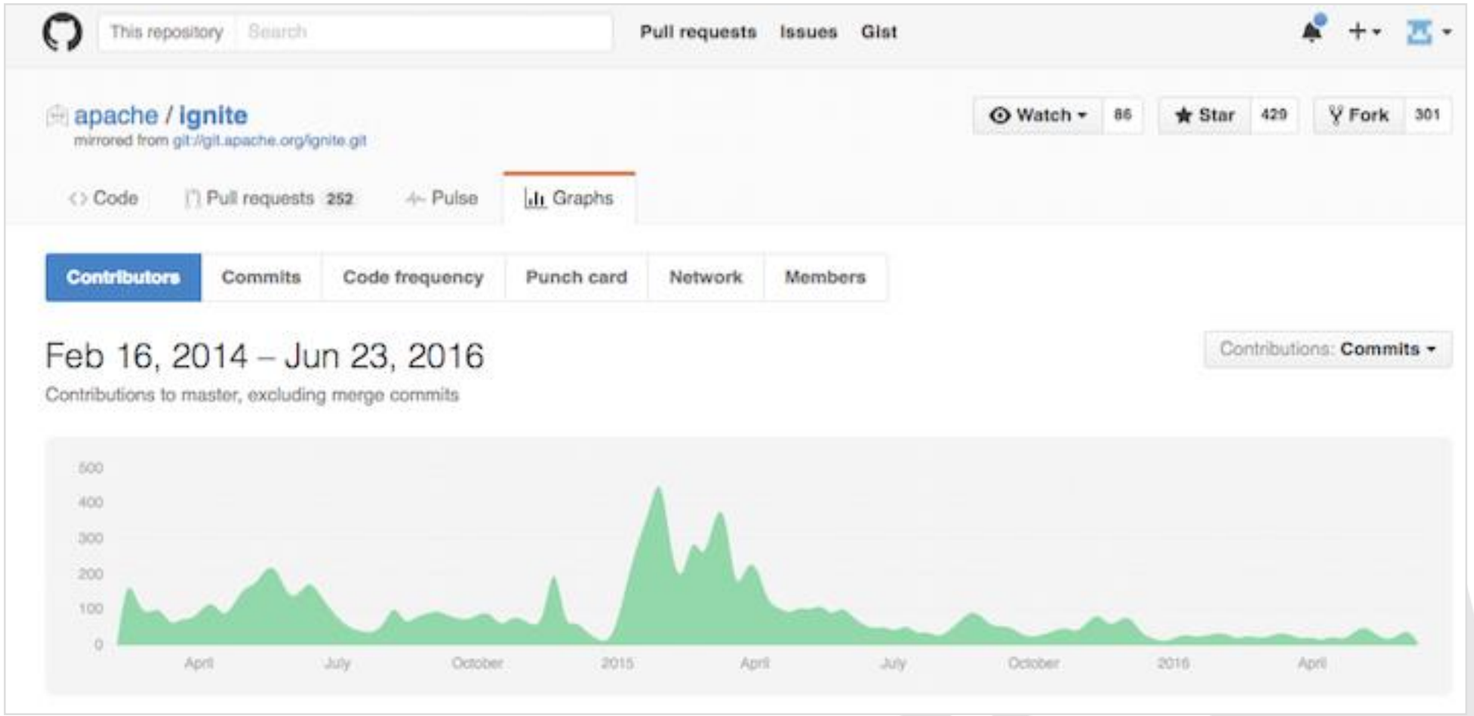
- Works with all popular RDBMS, NoSQL and Hadoop databases and offers a Unified API with support for a wide range of languages

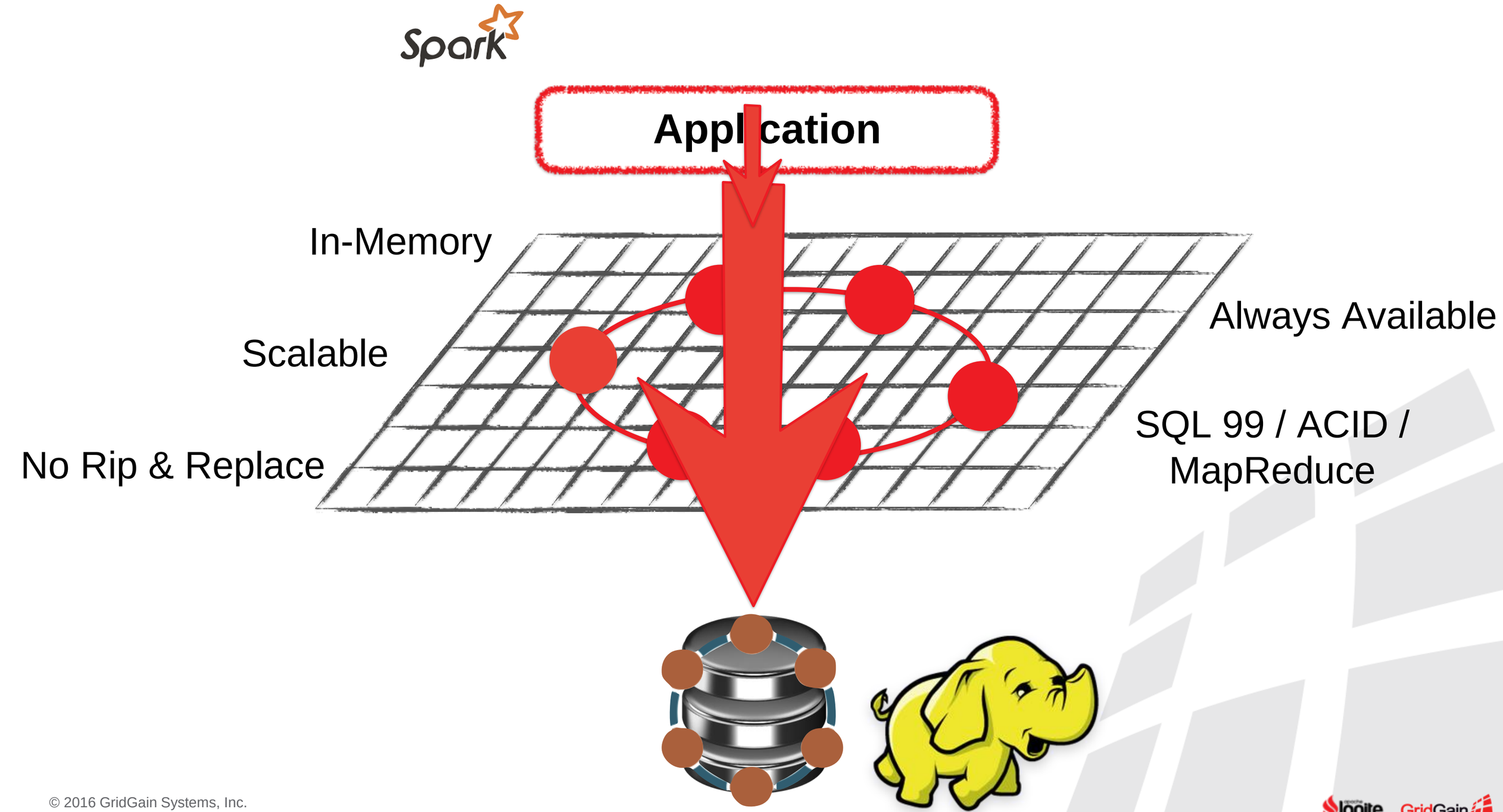
Deployable Anywhere

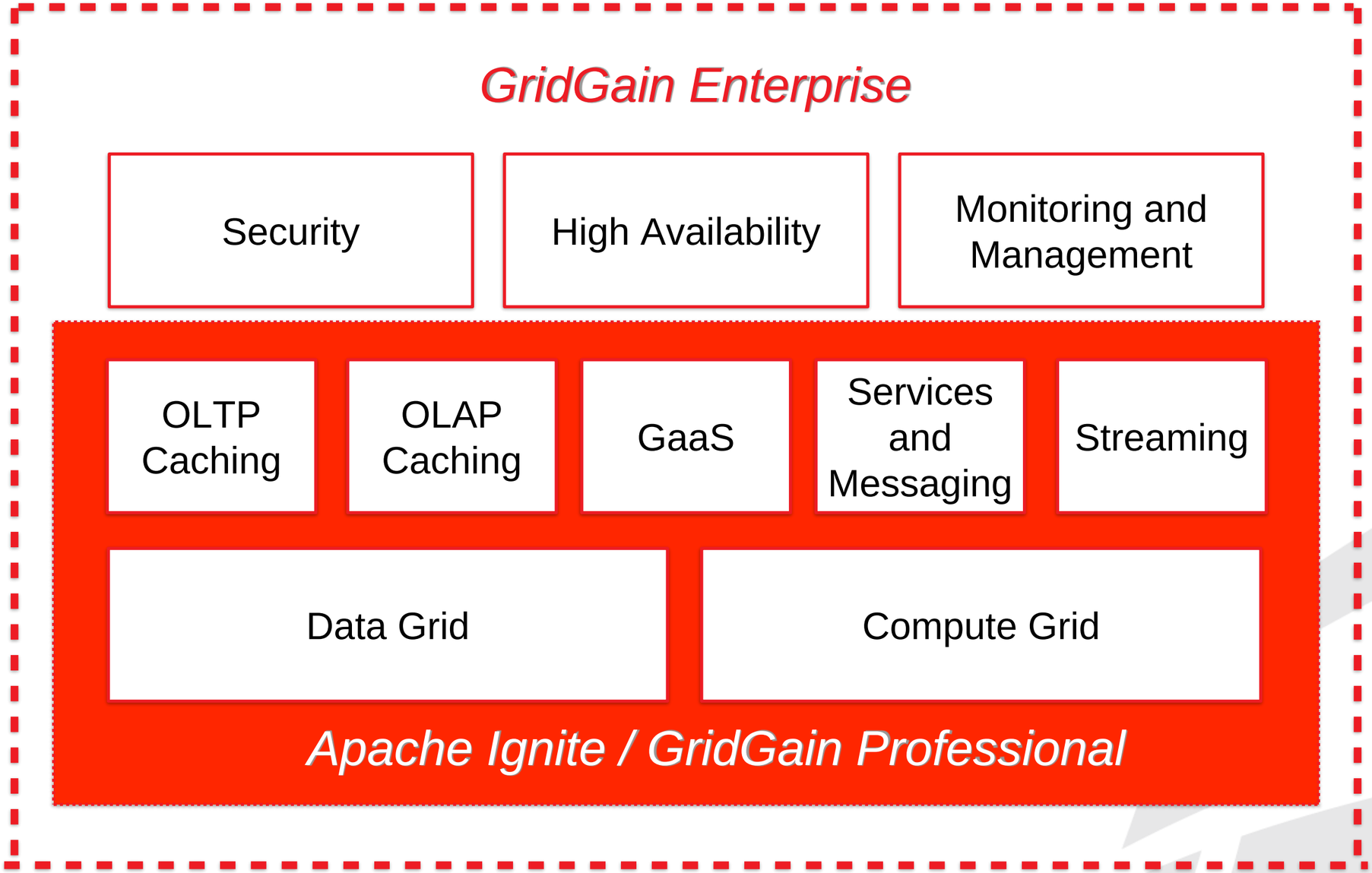
- Can be deployed on premise, in the cloud, or in hybrid environments

Apache Ignite Project

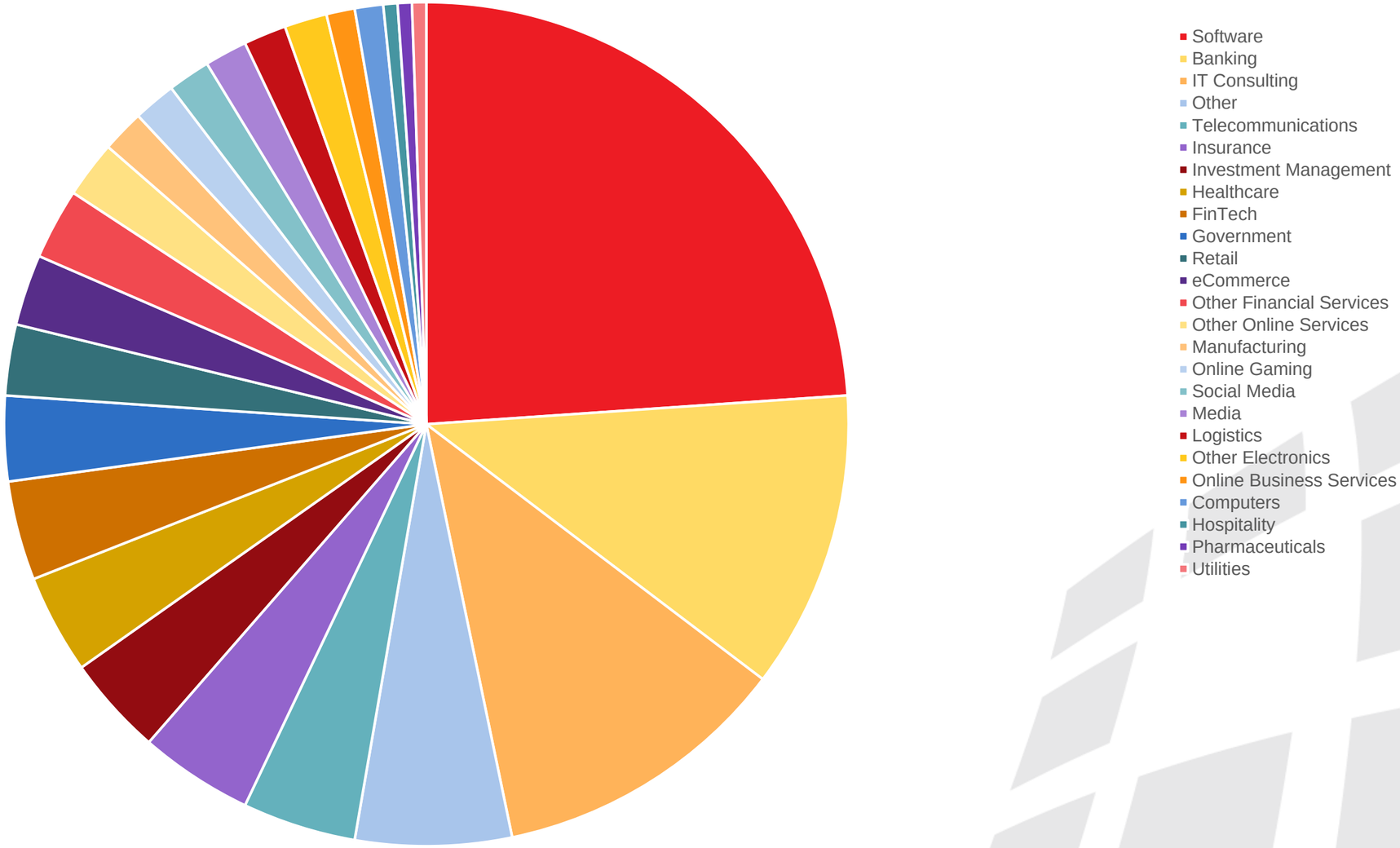
- 2007: First version of GridGain
- Oct. 2014: GridGain contributes Ignite to ASF
- Aug. 2015: Ignite is the second fastest project to | graduate after Spark
- Today:
  - 60+ contributors and rapidly growing
  - Huge development momentum - Estimated 192 years of effort since the first commit in February, 2014 [Openhub]
  - Mature codebase: 1M+ lines of code



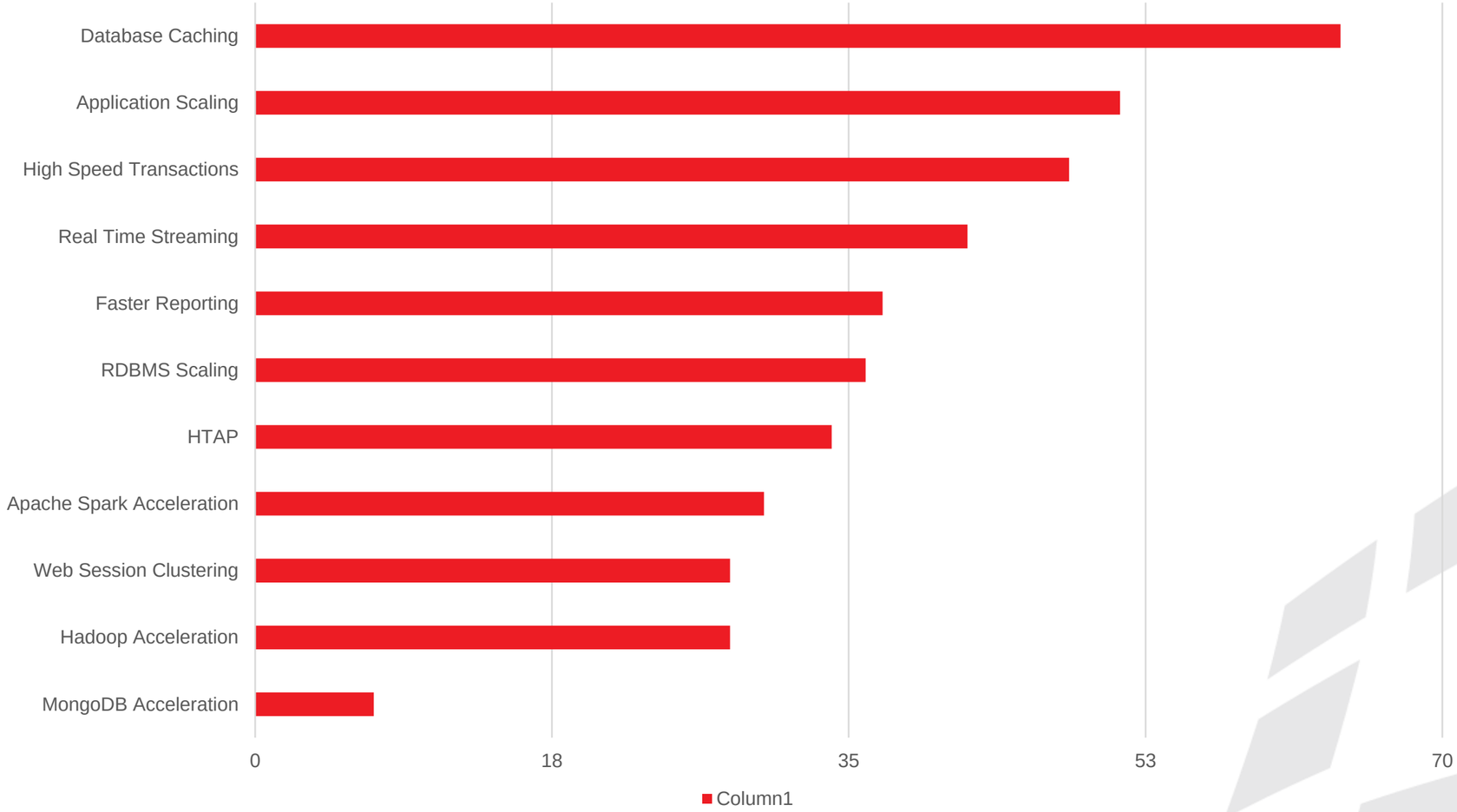




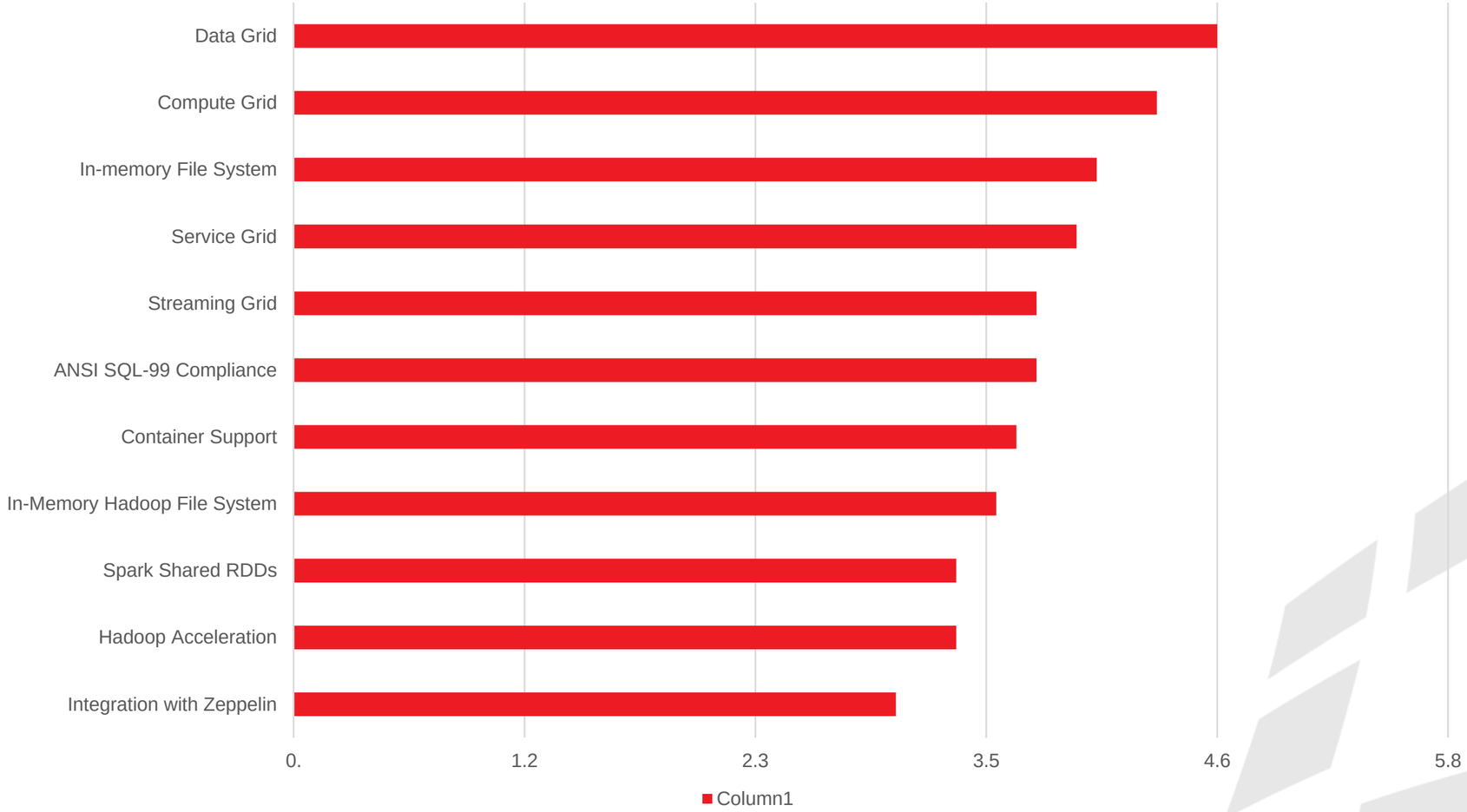
Survey Results: Primary Industries



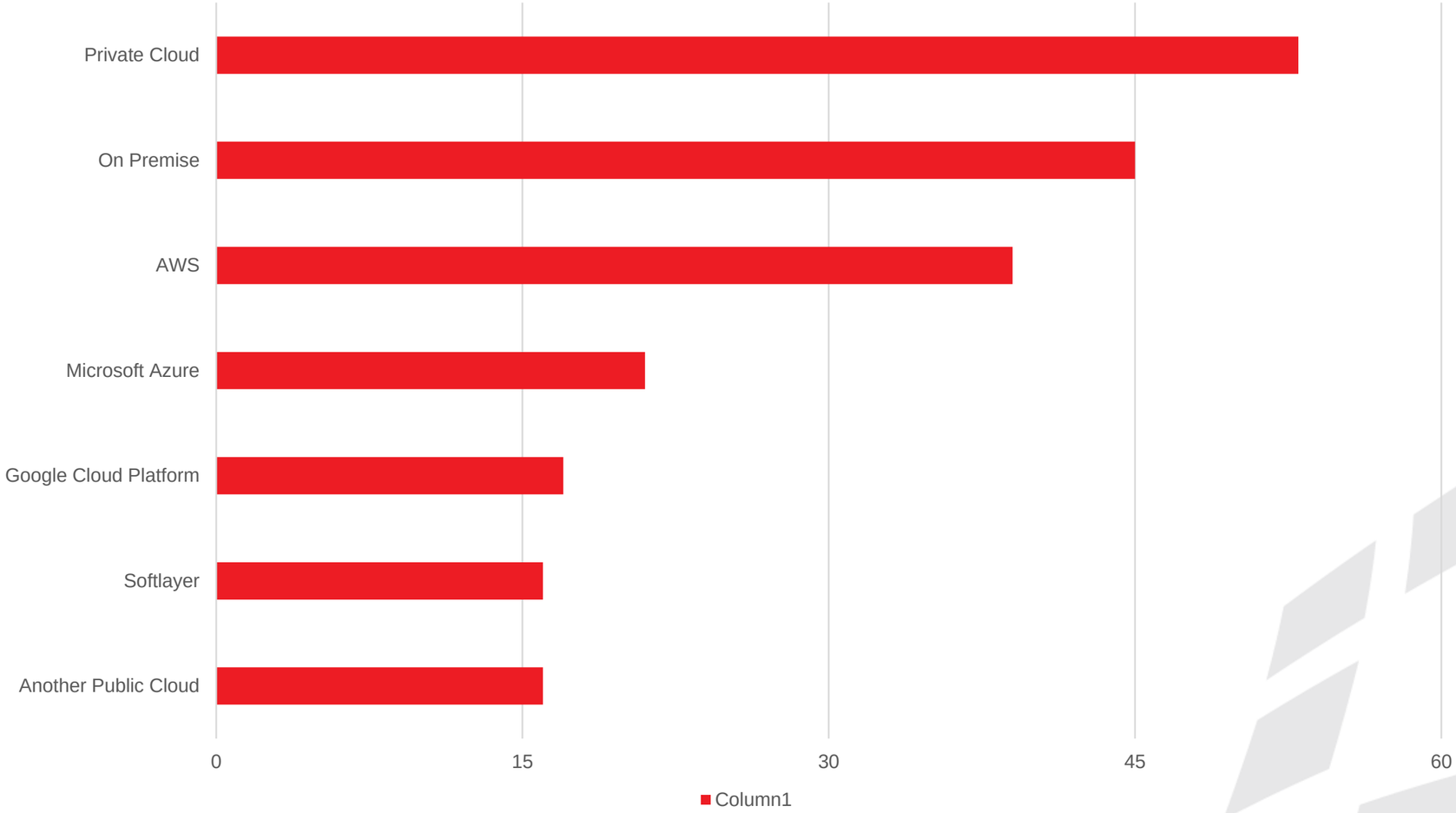
Survey Results: What uses were you considering for in-memory computing



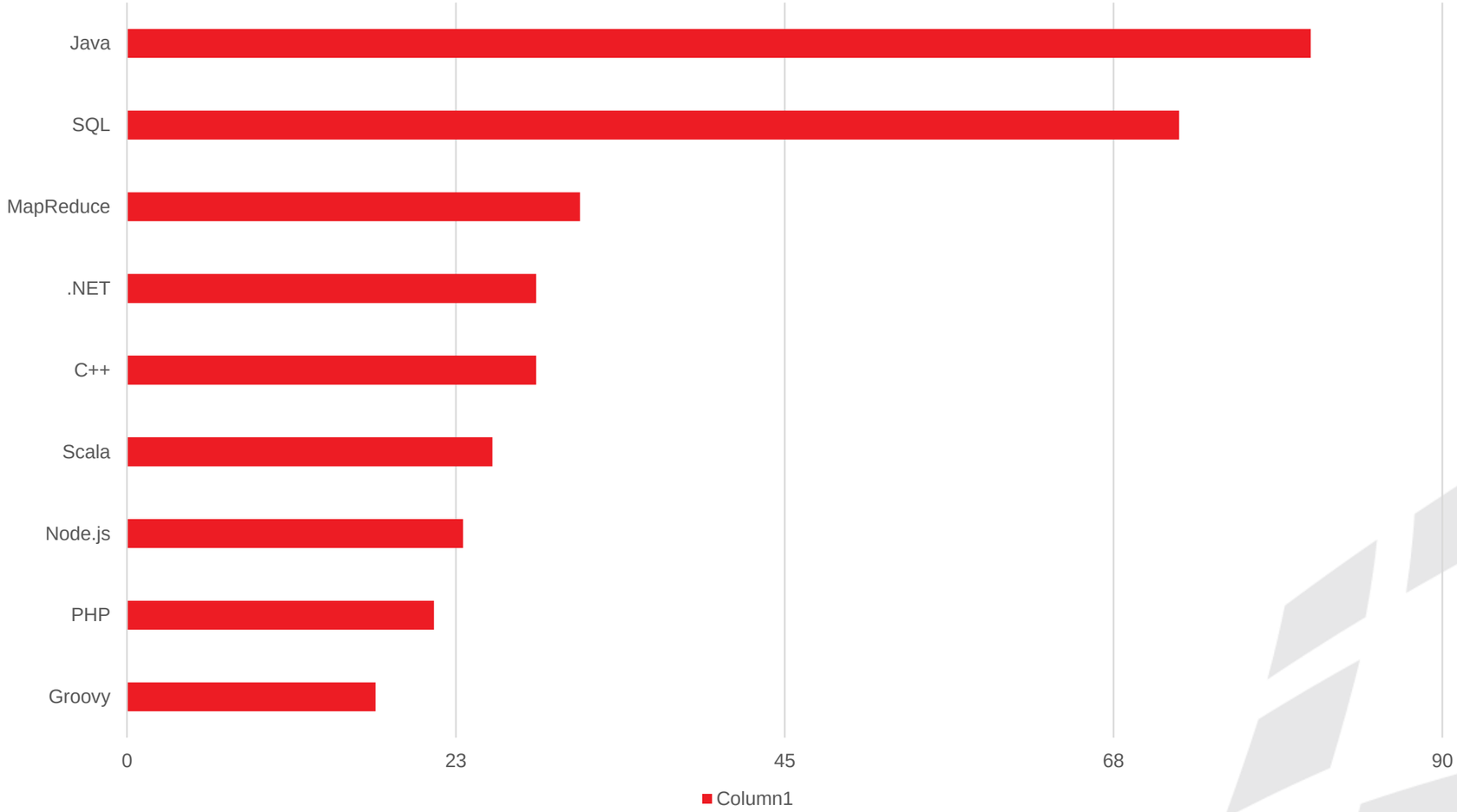
Survey Results: How important are each of the following product features to your organization?



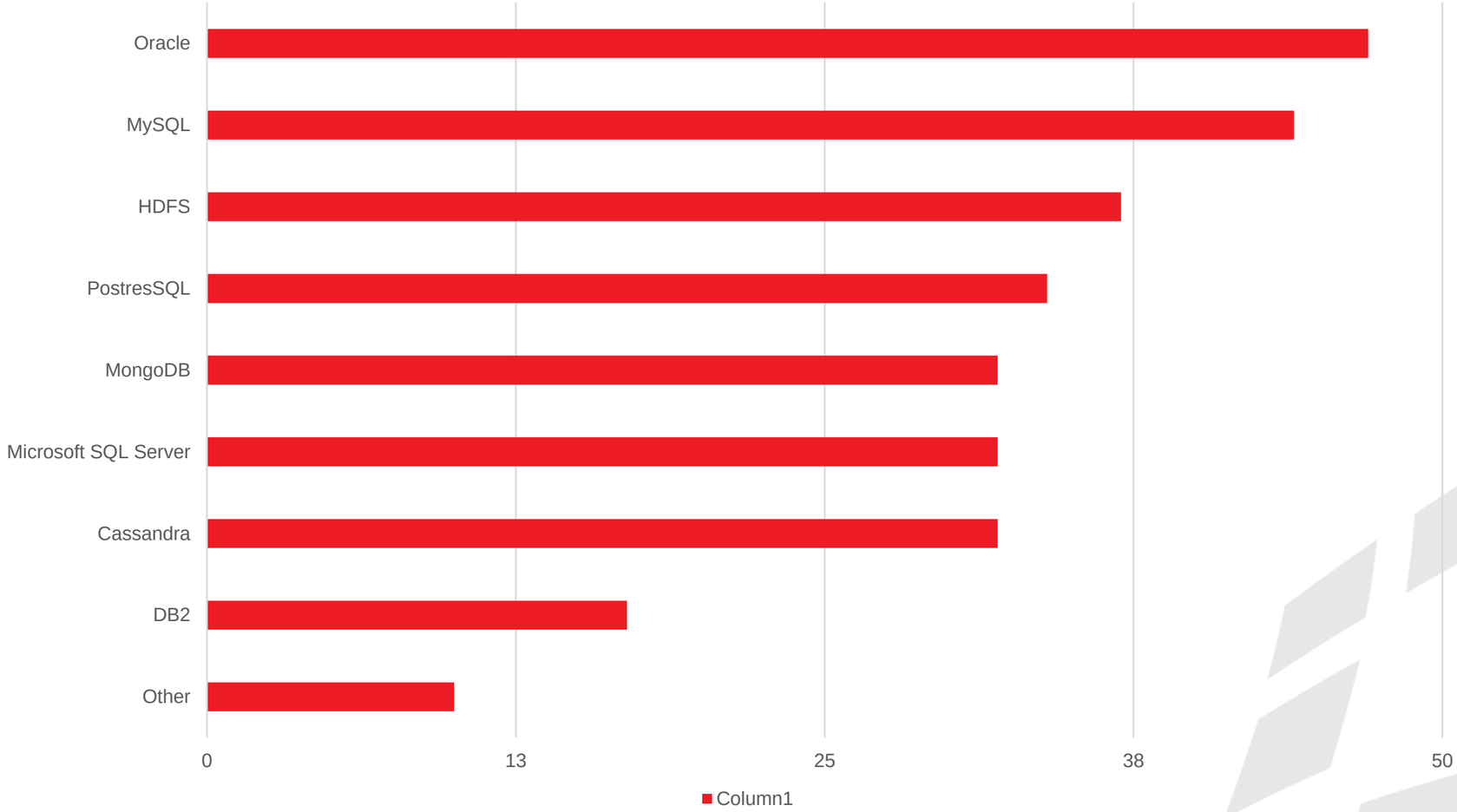
Survey Results: Where do you run GridGain and/or Apache Ignite?



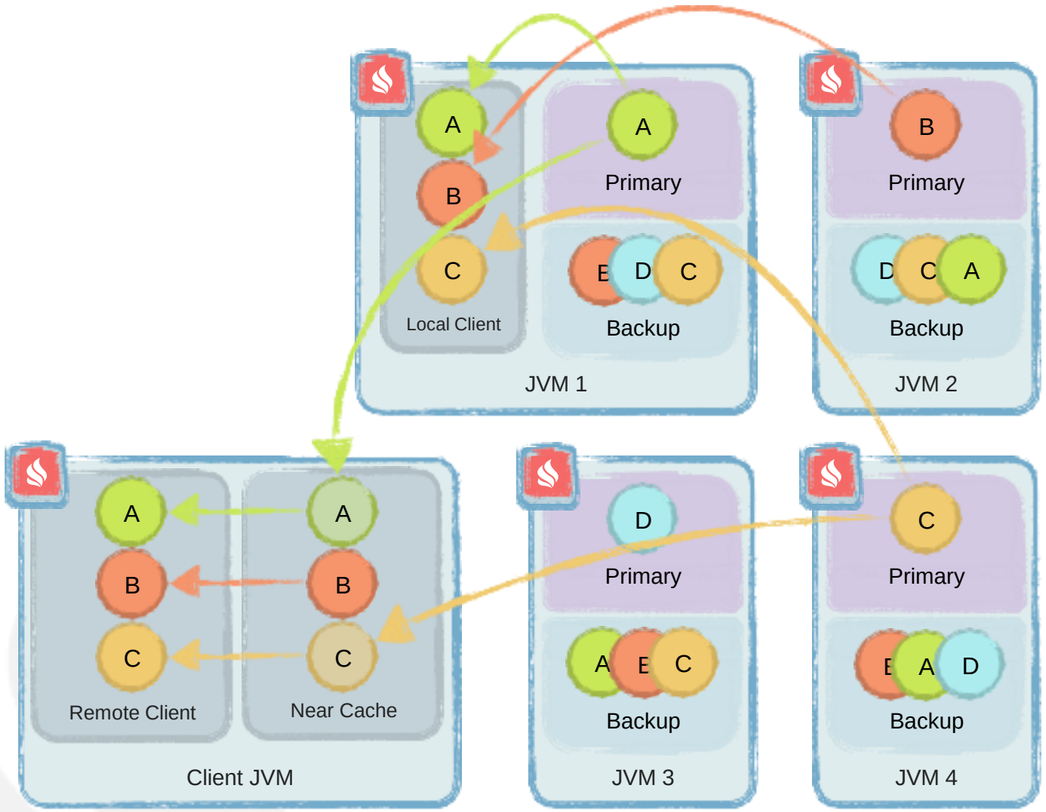
Survey Results: Which of the following protocols do you use to access your data?



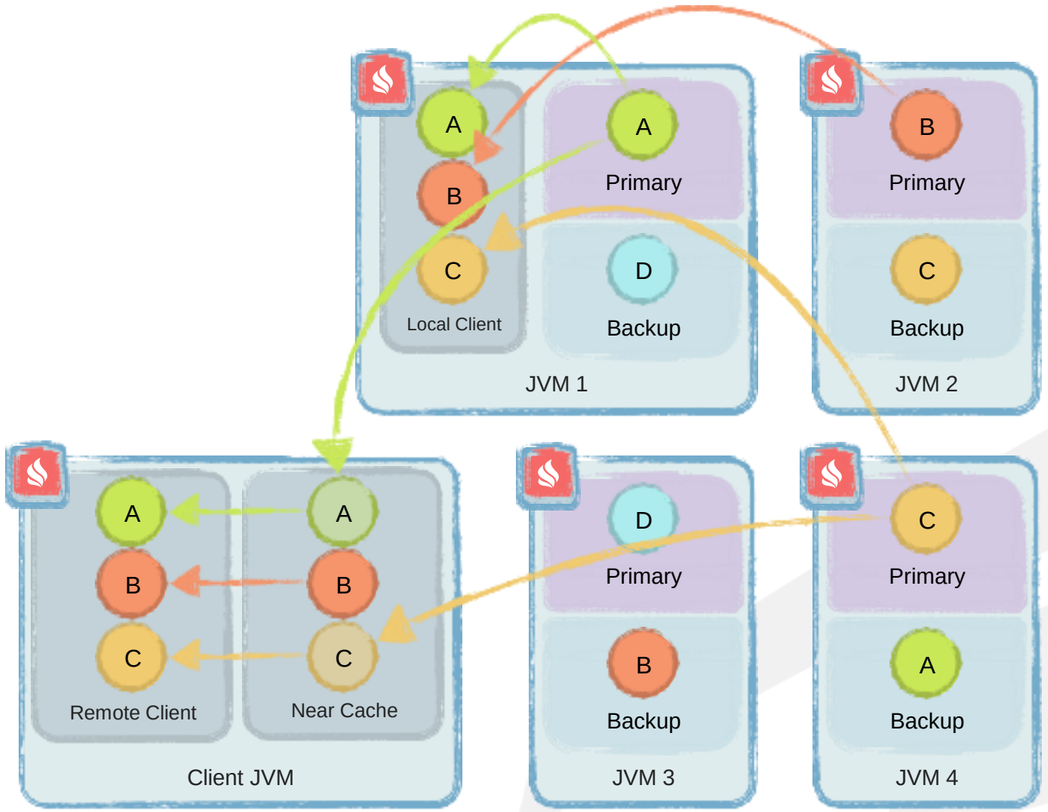
Survey Results: Which data stores are you/would you likely use with GridGain/Apache Ignite?



Data Grid: Cache modes & Horizontal Scaling

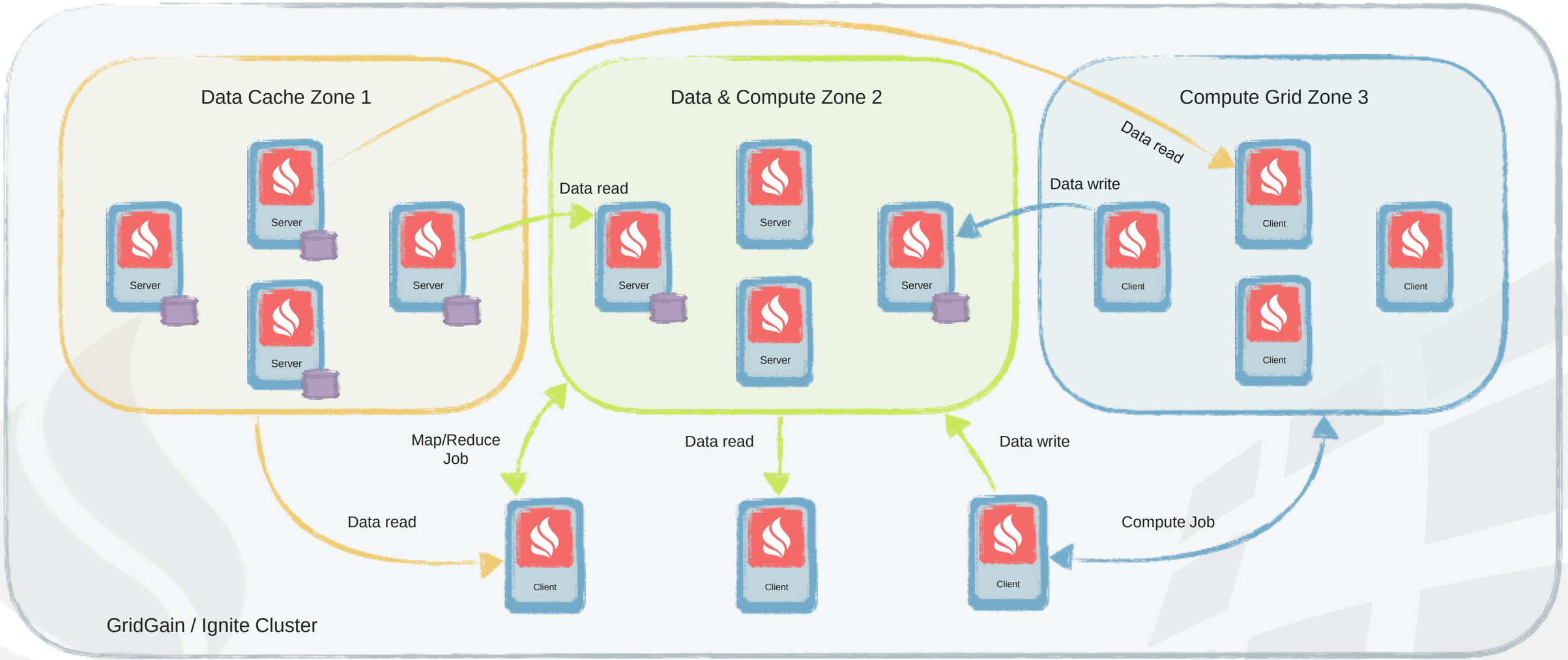


Replicated Cache



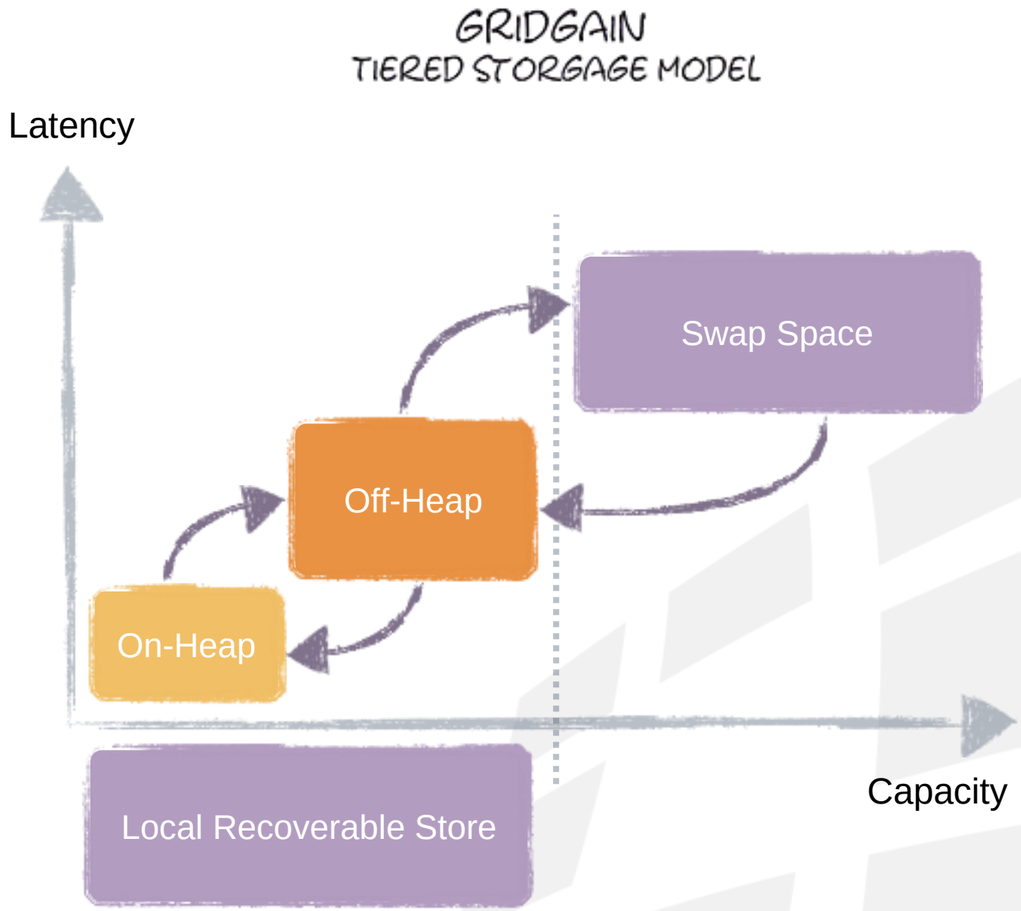
Partitioned Cache

Cluster Groups



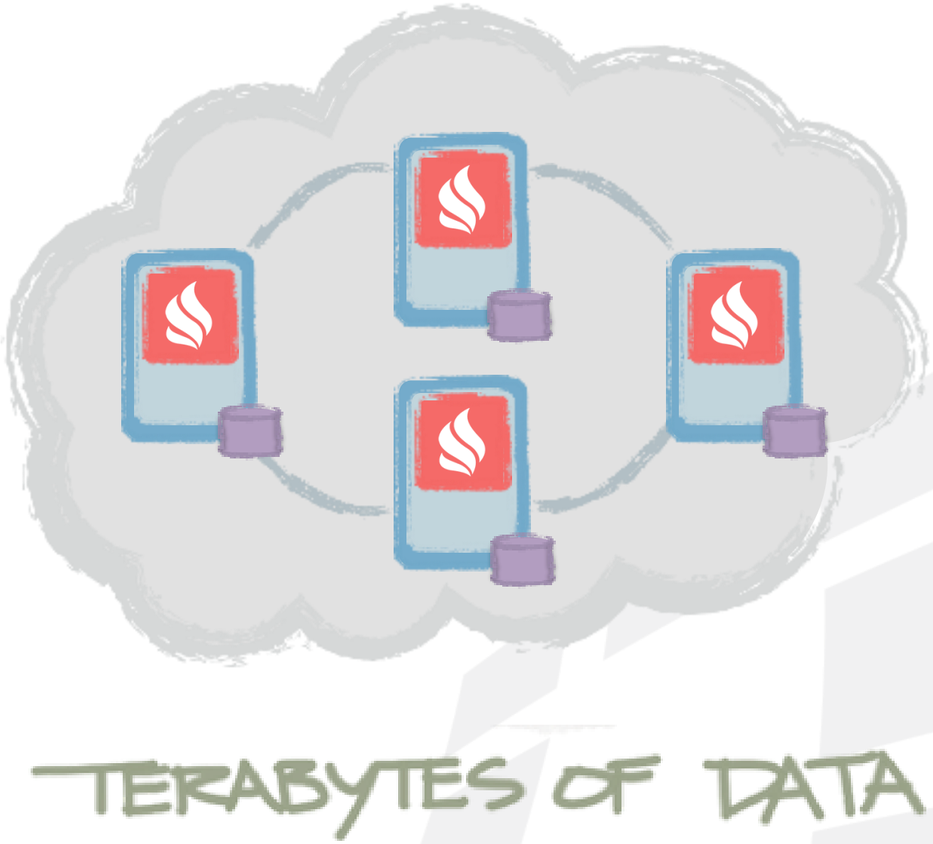
Data Grid: Tiered Memory & Local Store

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



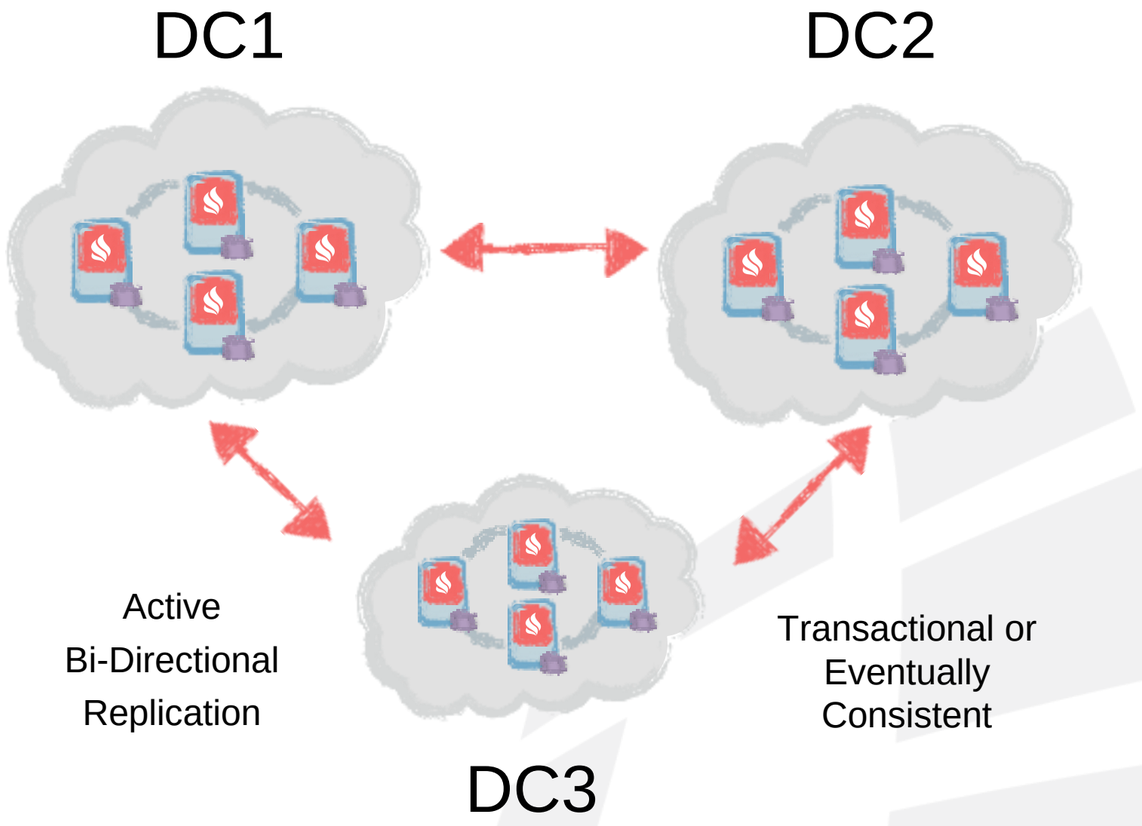
Data Grid: Off-Heap Memory

- Unlimited Vertical Scale
- Avoid Java Garbage Collection Pauses
- Small On-Heap Footprint
- Configurable eviction policies
- Off-Heap Indexes
- Full RAM Utilisation
- Simple Configuration



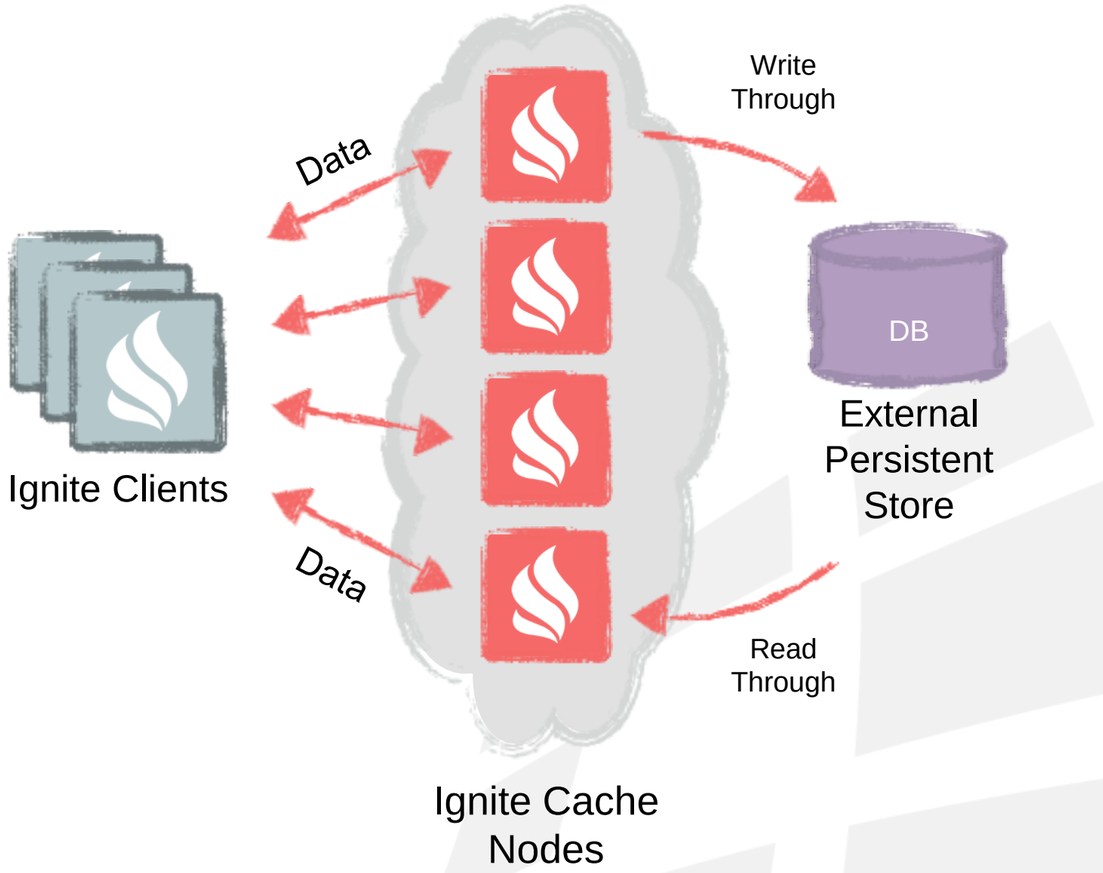
Data Grid: DC Replication

- Multiple (up to 32) Data Centres
- Complex Replication Technologies
- Active-Active & Active-Passive
- Smart Conflict Resolution
- Durable Persistent Queues
- Automatic Throttling
- **GridGain Enterprise**



Data Grid: External Persistence

- Read-through & Write-through
- Support for Write-behind
- Configurable eviction policies
- DB schema mapping wizard:
  - Generates all the XML configuration and Java POJOs



Data Grid: Cache APIs

- Predicate-based **Scan Queries**
- **Text Queries** based on Lucene indexing
- Query configuration using annotations, Spring XML or simple Java code
- **SQL Queries:** Automatic Group By, Aggregations, Sorting, Cross-Cache Joins, Unions
- **Memcached (PHP, Java, Python, Ruby)**
- HTTP REST API
- **JDBC & ODBC**

```java
IgniteCache<Long, Person> cache = ignite.cache("mycache");

// Find only persons earning more than 1,000.
try (QueryCursor cursor = cache.query(new ScanQuery((k, p) -> p.getSalary() > 1000)) {
  for (Person p : cursor)
    System.out.println(p.toString());
}
```

```java
// Query for all people with "Master Degree" in their resumes.
TextQuery txt = new TextQuery(Person.class, "Master Degree");

try (QueryCursor<Entry<Long, Person>> masters = cache.query(txt)) {
  for (Entry<Long, Person> e : cursor)
    System.out.println(e.getValue().toString());
}
```

```java
// Listing indexes.
Collection<QueryIndex> indexes = new ArrayList<>(3);

indexes.add(new QueryIndex("id"));
indexes.add(new QueryIndex("orgId"));
indexes.add(new QueryIndex("salary"));

queryEntity.setIndexes(indexes);
```

```java
IgniteCache<Long, Person> cache = ignite.cache("mycache");

// SQL join on Person and Organization.
SqlQuery sql = new SqlQuery(Person.class,
  "from Person, Organization "
  + "where Person.orgId = Organization.id "
  + "and lower(Organization.name) = lower(?)");

// Find all persons working for Ignite organization.
try (QueryCursor<Entry<Long, Person>> cursor = cache.query
(sql.setArgs("Ignite"))) {
  for (Entry<Long, Person> e : cursor)
    System.out.println(e.getValue().toString());
}
```

Data Grid: SQL Support (ANSI 99)

- ANSI-99 SQL
- In-Memory Indexes (On and Off-Heap)
- Automatic Group By, Aggregations, Sorting
- Cross-Cache Joins, Unions
- Use local H2 engine

```
IgniteCache<Long, Person> cache = ignite.cache("mycache");

// SQL join on Person and Organization.
SqlQuery sql = new SqlQuery(Person.class,
  "from Person, Organization "
  + "where Person.orgId = Organization.id "
  + "and lower(Organization.name) = lower(?)");

// Find all persons working for Ignite organization.
try (QueryCursor<Entry<Long, Person>> cursor = cache.query
(sql.setArgs("Ignite"))) {
  for (Entry<Long, Person> e : cursor)
    System.out.println(e.getValue().toString());
}
```

Data Grid: Transactions

- Fully ACID
- Support for Transactional & Atomic
- Cross-cache transactions
- Optimistic and Pessimistic concurrency modes with multiple isolation levels
- Deadlock protection
- JTA Integration

```
try (Transaction tx = transactions.txStart()) {
    Integer hello = cache.get("Hello");

    if (hello == 1)
        cache.put("Hello", 11);

    cache.put("World", 22);

    tx.commit();
}
```

```
IgniteTransactions txs = ignite.transactions();

// Start transaction in optimistic mode with repeatable read is
olation level.
Transaction tx = txs.txStart(TransactionConcurrency.OPTIMISTIC,
TransactionIsolation.REPEATABLE_READ);
```

1000’s of Deployments

Automated Trading Systems

- *Real time analysis of trading positions*
- *Real time market risk assessment*
- *High volume transactions*
- *Ultra low latencies trading*

Financial Services

- *Fraud Detection*
- *Risk Analysis*
- *Insurance rating and modeling*

Big Data Analytics

- *Real time analysis of inventory*
- *Operational up-to-the-second BI*

Mobile & IoT

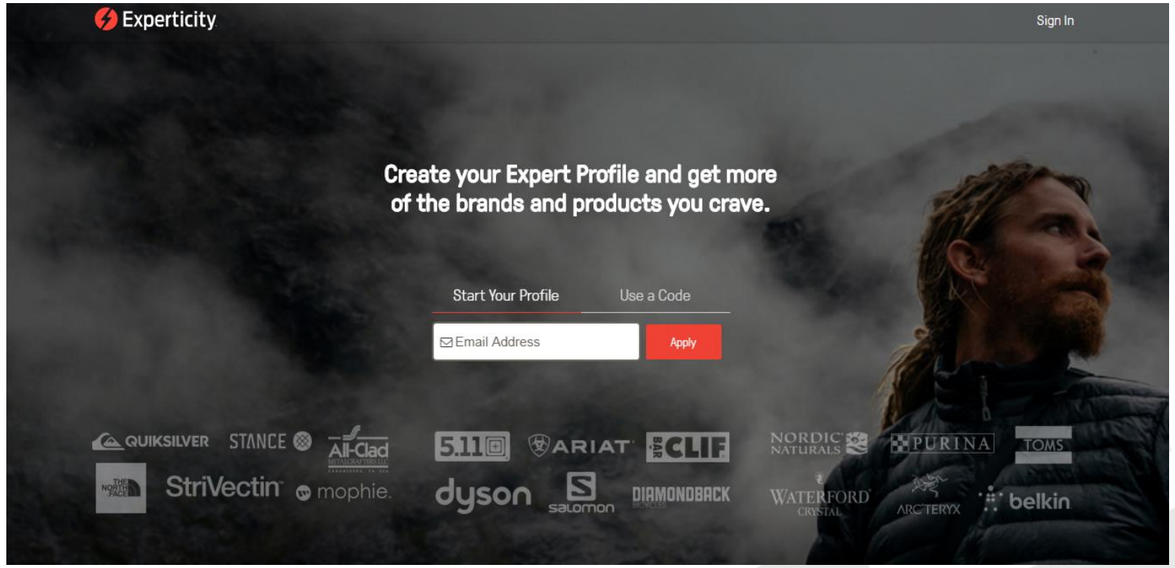
- *Real-time streaming processing*
- *Complex event processing*

Biotech

- *High performance genome data matching*
- *Drug discovery*

Case Study: Experticity

- Connects over 650 lifestyle and consumer brands (more than 4.5 million products) with brand advocates
- Brands use the network to identify experts and engage them in their sales channel
- Member profiles (interests, experience, content viewed, purchases)
- Real Time Analytics on thousands of active user sessions for personalized content



"We need it to work fast and work at scale, and GridGain does. We couldn't do what we do without GridGain, and we're confident we can scale to meet the growth we anticipate."
-Jeremy Knudsen, CTO

Case Study: intelligentpipe

- Background:
  - Intelligentpipe is a big data software company serving the global telecommunications industry by developing solutions for mobile operators to improve their business and operational processes
- The Challenge
  - Collect and analyze massive amounts of mobile user traffic data in real time
  - Tens of millions of users
  - Consumption of network resources
  - Type of network traffic (voice or data)

Case Study: intelligentpipe

- GridGain Professional Edition used to build a high performance low latency analysis platform
- “GridGain ensures responsiveness regardless of how much information we need to search through.” Sakari Paloviita, CTO, Intelligentpipe
- Collect and analyze multiple terabytes per day

Case Study: intelligentpipe

- Real-time analytics provides fast insight
- Easy integration with existing systems due to GridGain's Unified API and ANSI SQL-99 support
- Linear scaling across deployed server to keep up seamlessly as the business grows

> We'll want to use technology GridGain offers so we can focus on our core business ourselves."
> - Jari Kuusela, Director of Product Management

Case Study: CYBER DUST

- Background:
  - Cyber Dust is a platform for text messages: “A safer place to text.”
    - Untraceable
    - Encrypted
    - Disappearing
    - Screenshot blocking
  - Available for Andoid and IoS
  - Mark Cuban funded

- The Challenge
  - To build a real-time, reliable and highly available server infrastructure to support a mobile messaging platform
  - More than 500K users
  - Millions of messages a day
  - Avoid writing messages to disk

Case Study: 

- GridGain Professional Edition used to build a messaging platform
- Runs completely on Amazon EC2
- All user account data, configurations, and messages held in memory
- Messages deleted without a trace because they were never written to disk
- Extensive use of Unified API

Case Study: CYBER DUST

- “Blast” feature performance: capable of broadcasting disappearing messages to all of a user’s contacts
- Real-world performance of 300,000 messages sent and disappeared in 30 seconds

I was pleasantly surprised by the GridGain solution and performance.
-Igor Shpitalnik, CTO

I keep learning about additional capabilities GridGain offers. It’s what I expected and more.
-Igor Shpitalnik, CTO

Case Study:

- Financial services software
  - Retail and corporate banking
  - Lending
  - Treasury
  - Capital markets
  - Investment management
  - Enterprise risk
- More than 2,000 customers in 130 countries
- Used by 48 of the world's 50 largest banks

- The Challenge: Eliminating Data Processing Bottlenecks
  - Huge amounts of trade and accounting data
  - Customers need
    - High-speed transactions
    - Real-time reporting
  - New Java-based IT stack with data lake support
  - Global regulatory compliance

Case Study:

- Commodity servers (256GB RAM)
- Data stored in memory
  - Transactions
  - Market data
- Parallel processing across cluster
  - Calculation heavy reporting for regulatory compliance

Case Study:

FusionFabric.cloud

- Integrates trading systems with cloud-based components
  - OTC derivatives
  - Exchange traded derivatives
  - Inflation
  - Fixed income
  - FX/MM
  - Hybrids
  - Developing additional modules

"With GridGain, we have achieved real-time processing of massive amounts of trade and transaction data, eliminating bottlenecks and enabling us to offer next-generation financial services to our customers."
-Felix Grevy, Director of Product Management for FusionFabric.cloud at Misys

ANY QUESTIONS?

Thank you for joining us. Follow the conversation.

www.gridgain.com
www.gridgain.com/resources/blog

@gridgain
#gridgain #inmemorycomputing
@msarrel
matt.sarrel@gridgain.com