Top 7 Apache Ignite FAQs From Around the World!

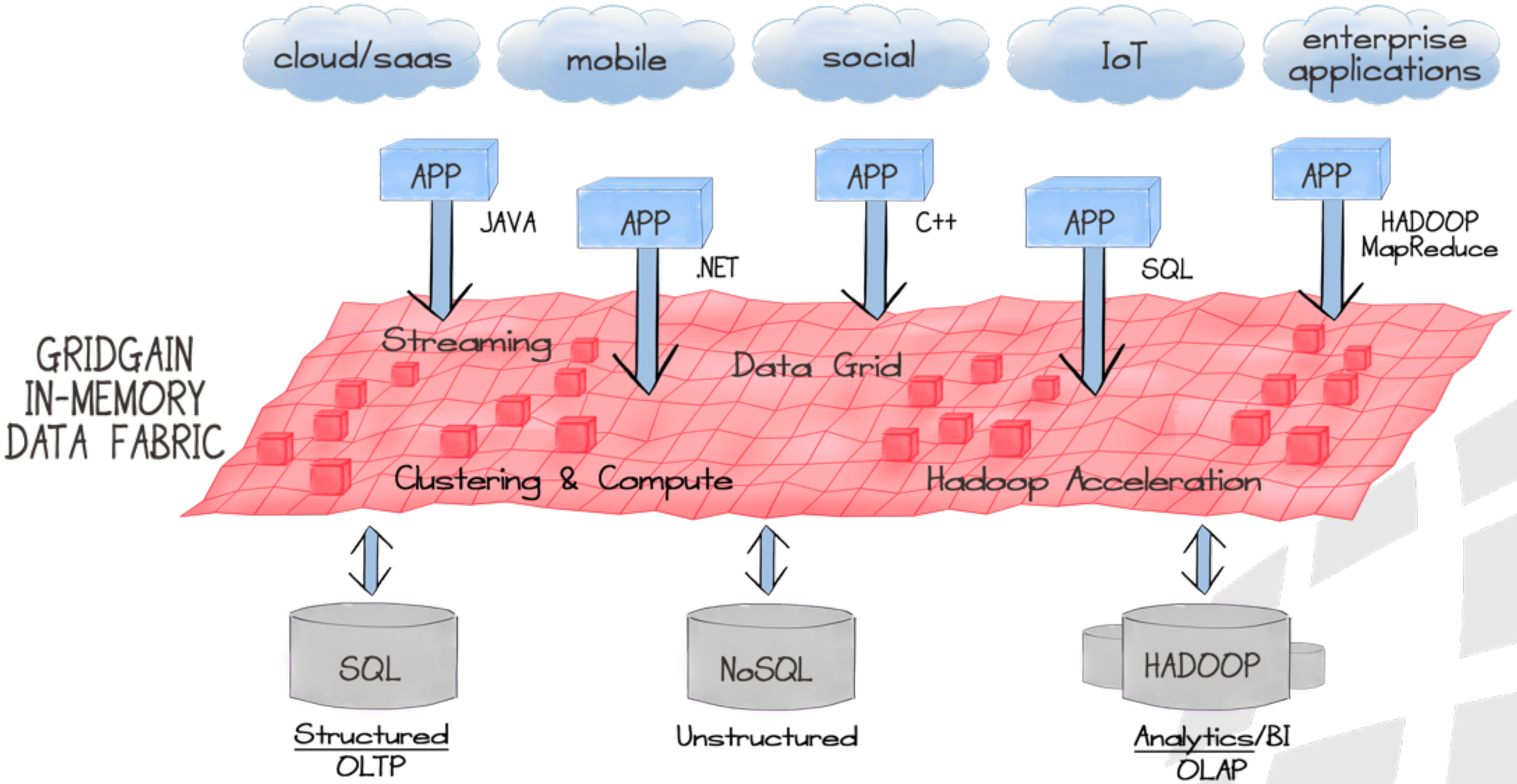
Christos Erotocritou
Managing Solutions Architect - EMEA

Rachel Pedreschi
Managing Solutions Architect - Americas

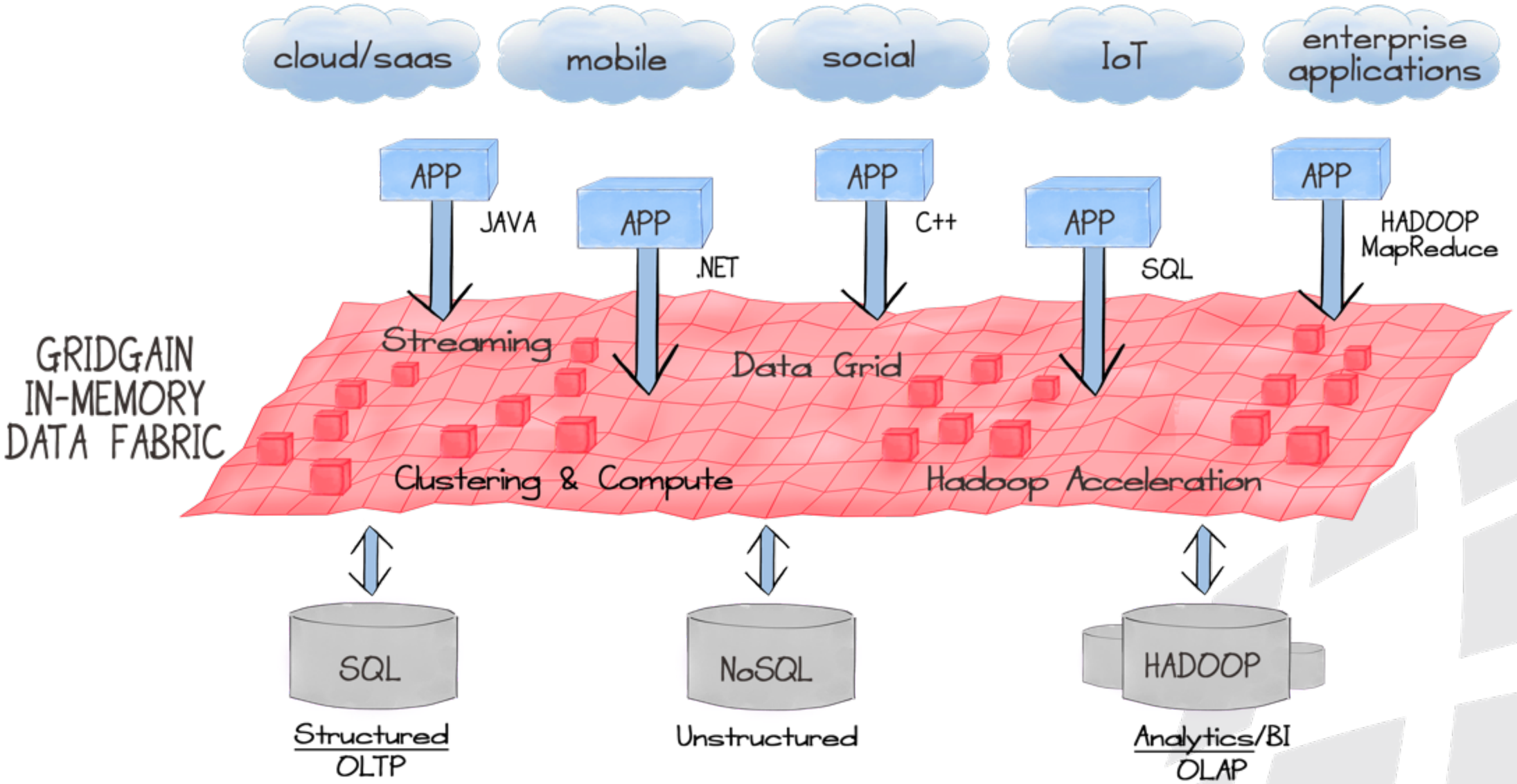
GridGain

Agenda

- Brief architecture overview of Apache Ignite
- Which APIs can I use with Apache Ignite?
- How can I get faster data manipulation using the compute grid?
- How can my transactions be ACID compliant and also highly available?
- How many servers / nodes are needed for my use case?
- Which persistent data stores work with Ignite?
- What are the architectural best practices for both large and small deployments?
- What is the best choice for my deployment: Apache Ignite, GridGain Professional or GridGain Enterprise Edition?
- Q & A



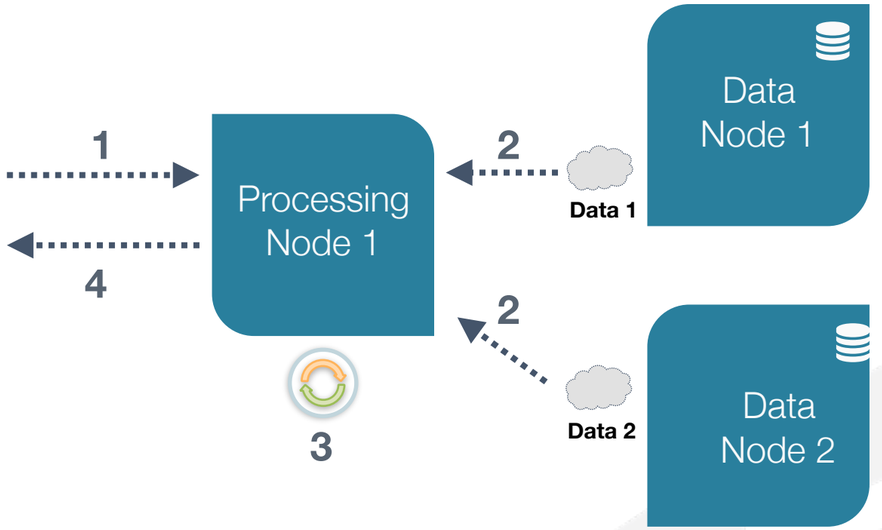
Which APIs can I use with Apache Ignite?



How can I get faster data manipulation using the compute grid?

Client / Server Pattern

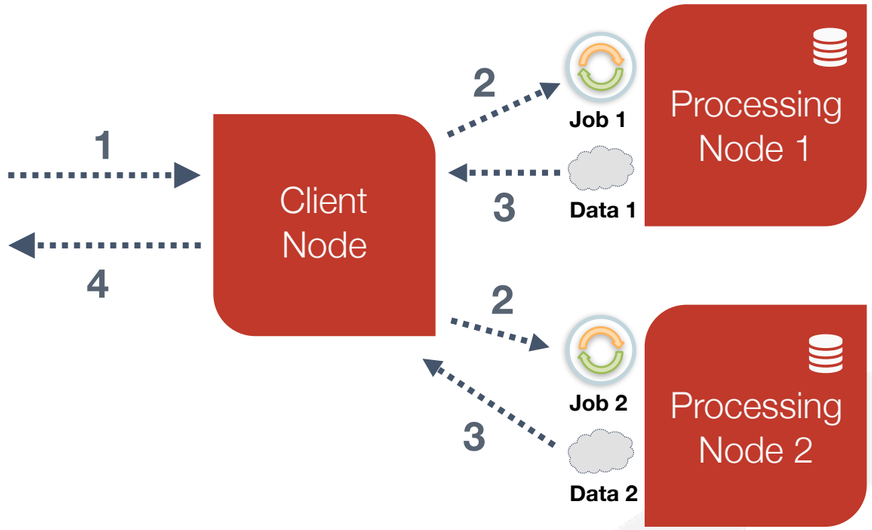
- Moving data around is costly
  - High network utilisation
  - Higher latency
  - More error prone



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

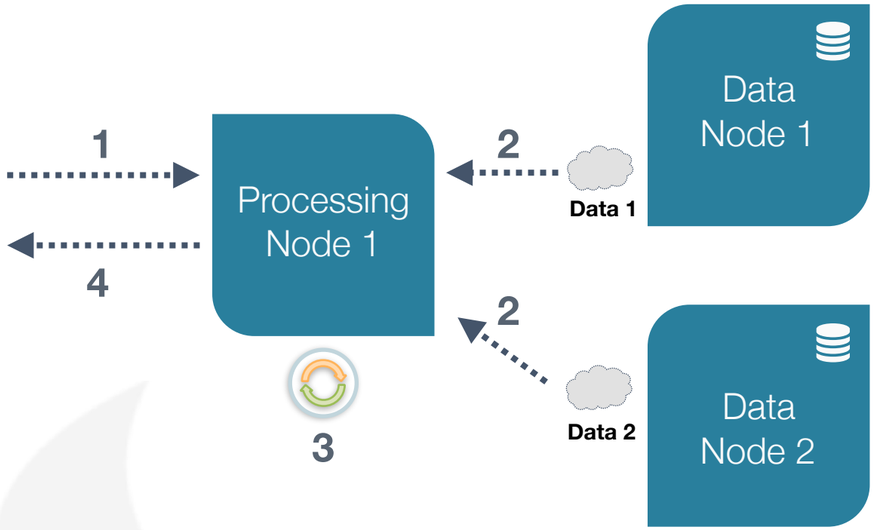
Map/Reduce & Affinity Runs

- Sending the processing to the data is faster and more efficient:
  - Entry processors
  - Compute tasks
    - Map / Reduce
    - Affinity Runs

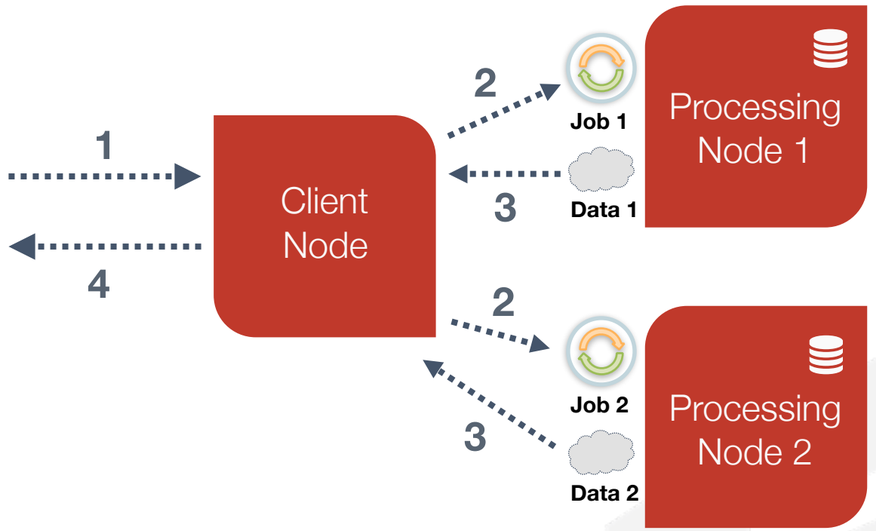


1. Initial Request
2. Co-locating processing with data
3. Return partial result
4. Reduce & return to client

Client-Server vs. Map/Reduce



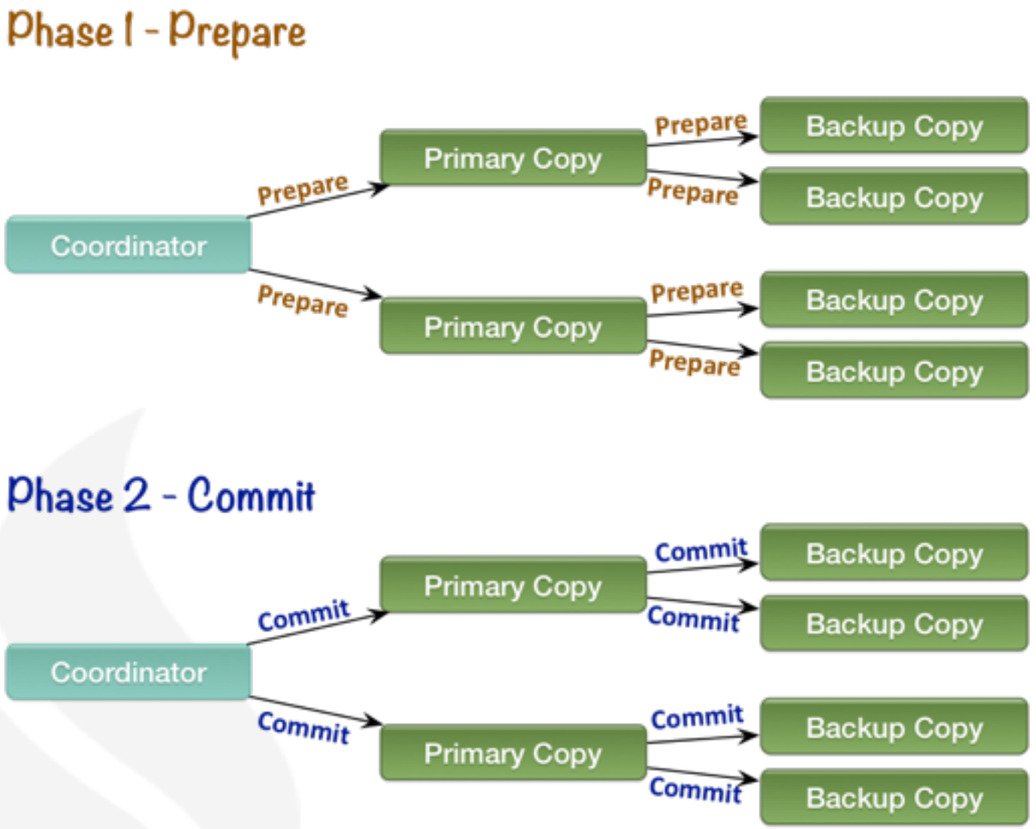
1. Initial Request
2. Fetch data from remote nodes
3. Process entire data-set
4. Return to client



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

How can my transactions be ACID compliant and also highly available?

2 Phase Commit



How many servers / nodes are needed for my use case?

Memory Capacity Planning

- Calculate primary data size: multiply the size of one entry in bytes by the total number of entries
- If you have backups, multiply by their number
- Indexes also require memory. Basic use cases will add a 30% increase
- Add around 20MB per cache
- Add around 200-300MB per node for internal memory and reasonable amount of memory for JVM and GC to operate efficiently

Example Specification

- 2,000,000 objects
- 1,024 bytes per object (1 KB)
- 1 backup
- 4 nodes

Memory Capacity Planning

- **I have 300GB of data in DB will this be the same in Ignite?**

No, data size on disk is not a direct 1-to-1 mapping in memory. As a very rough estimate it can be about 2.5/3 times size on disk excluding indexes and any other overhead. To get a more accurate estimation you need to figure out the average object size by importing a record into Ignite and multiplying by the number of objects expected.

Processing Capacity Planning

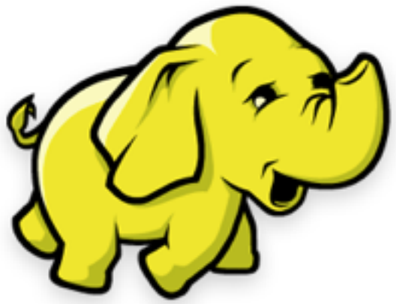
- Understand the cost of a given operation that your application will be performing and multiply this by the number of operations expected at various times
- A good starting point for this would be the Ignite benchmarks which detail the results of standard operations and give a rough estimate of the capacity required to deliver such performance

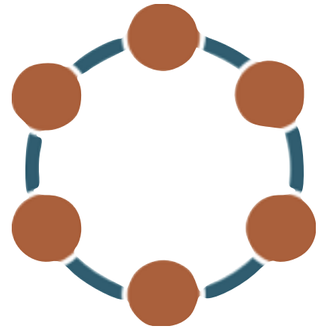
With 32 cores over 4 large AWS instances the following benchmarks were recorded:

- PUT/GET: 26k/sec
- PUT (TRANSACTIONAL): 68k/sec
- PUT (TRANSACTIONAL - PESSIMISTIC): 20k/sec
- PUT (TRANSACTIONAL - OPTIMISTIC): 44k/sec
- SQL Query: 72k/sec

More results here

Which persistent data stores work with Ignite?

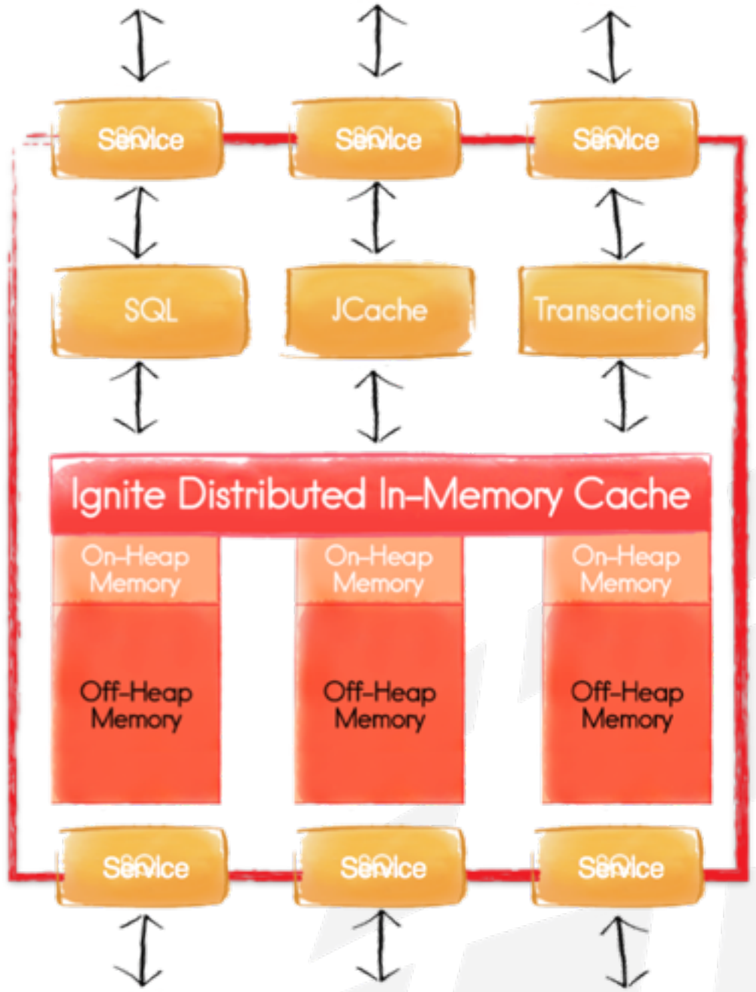




What are architectural best practices for both large and small deployments?

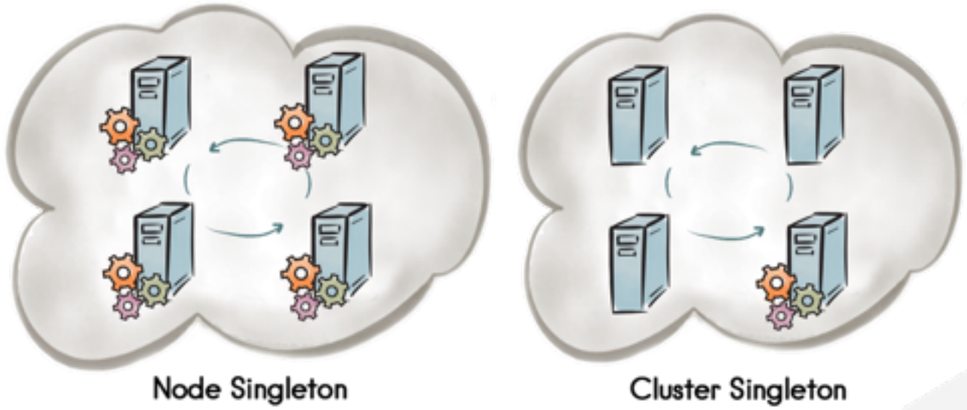
In-Memory Service Grid

- Resilience - Build an in-memory resilient service layer between your client application and the grid
- Only expose application APIs and not direct grid APIs
- Service Chaining - Call services internally via compute tasks to create service chains



In-Memory Service Grid

- Singletons on the Cluster
  - Cluster Singleton
  - Node Singleton
  - Key Singleton
- Guaranteed Availability
  - Auto Redeployment in Case of Failures

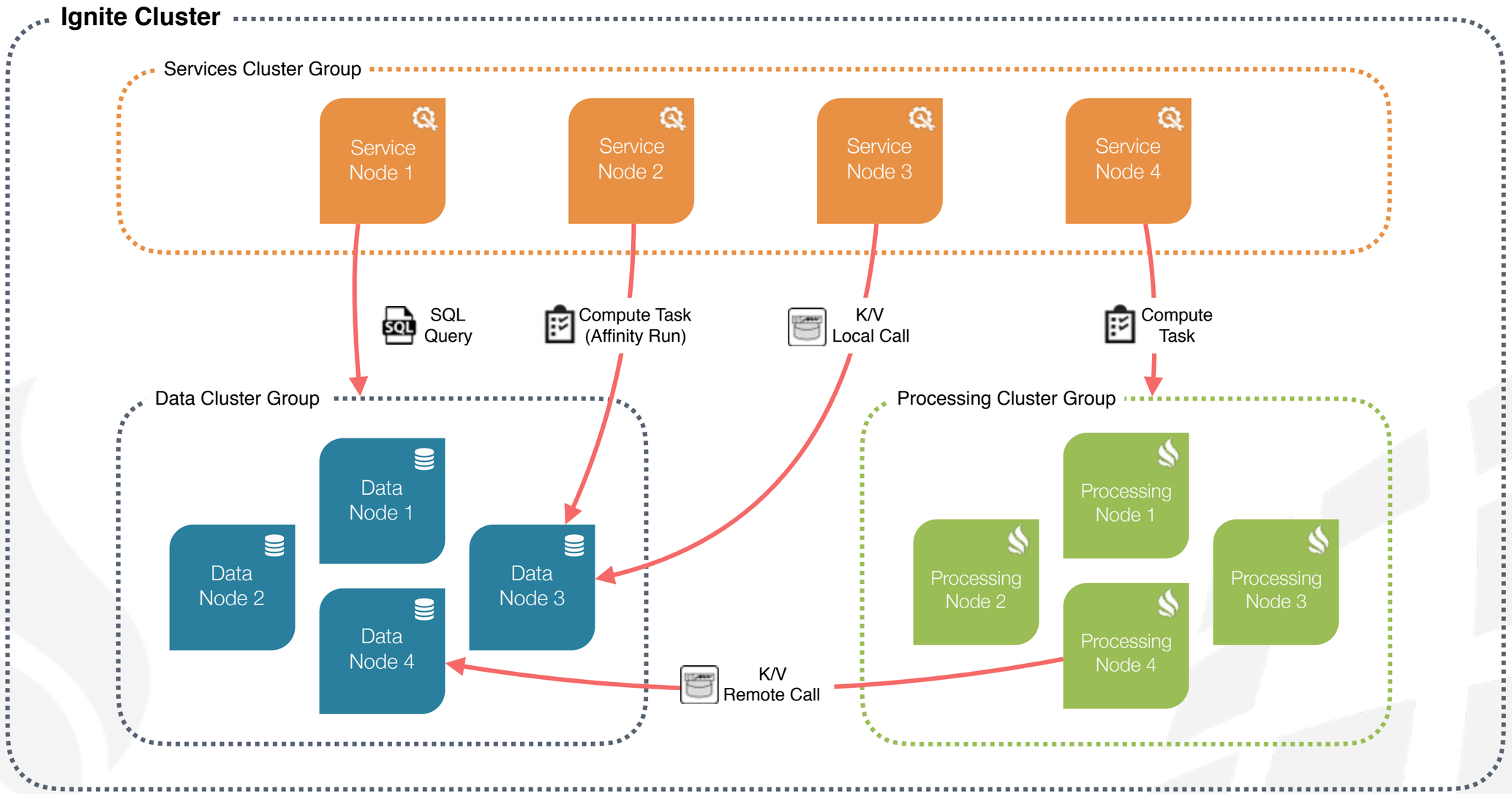



```
svcs.deployNodeSingleton("myNodeSingleton", new MyService());
```

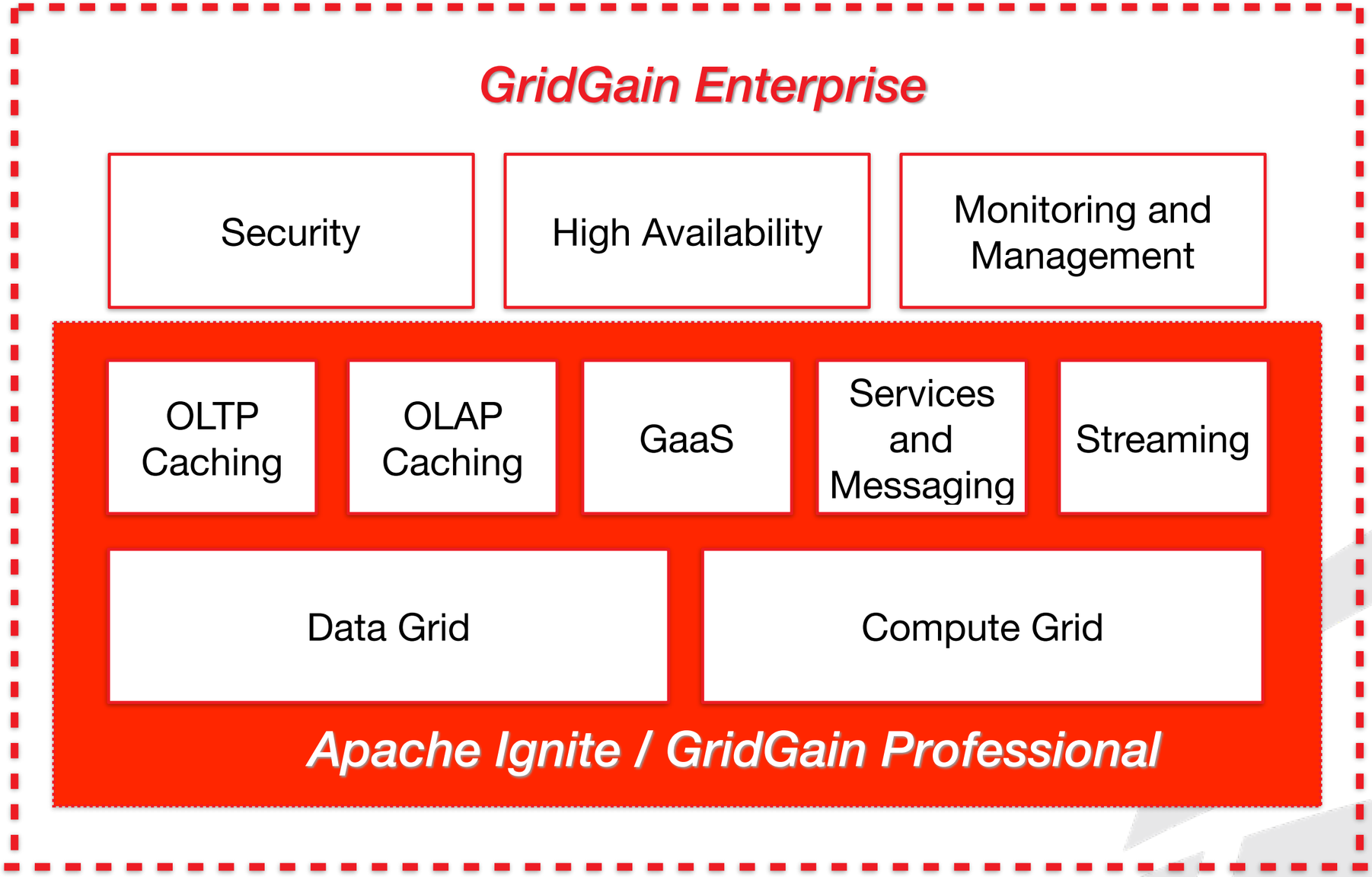
```
svcs.deployClusterSingleton("myClusterSingleton", new MyService());
```

```
svcs.deployKeyAffinitySingleton("myKeySingleton", new MyService(), "myCache", new MyCacheKey());
```

Logical Cluster Groups



What is the best choice for my deployment: Apache Ignite, GridGain Professional or GridGain Enterprise Edition?



Thank You!

Authors: Christos Erotocritou and Rachel Pedreschi

Thank you for joining us. Follow the conversation.

www.gridgain.com

@gridgain
#gridgain