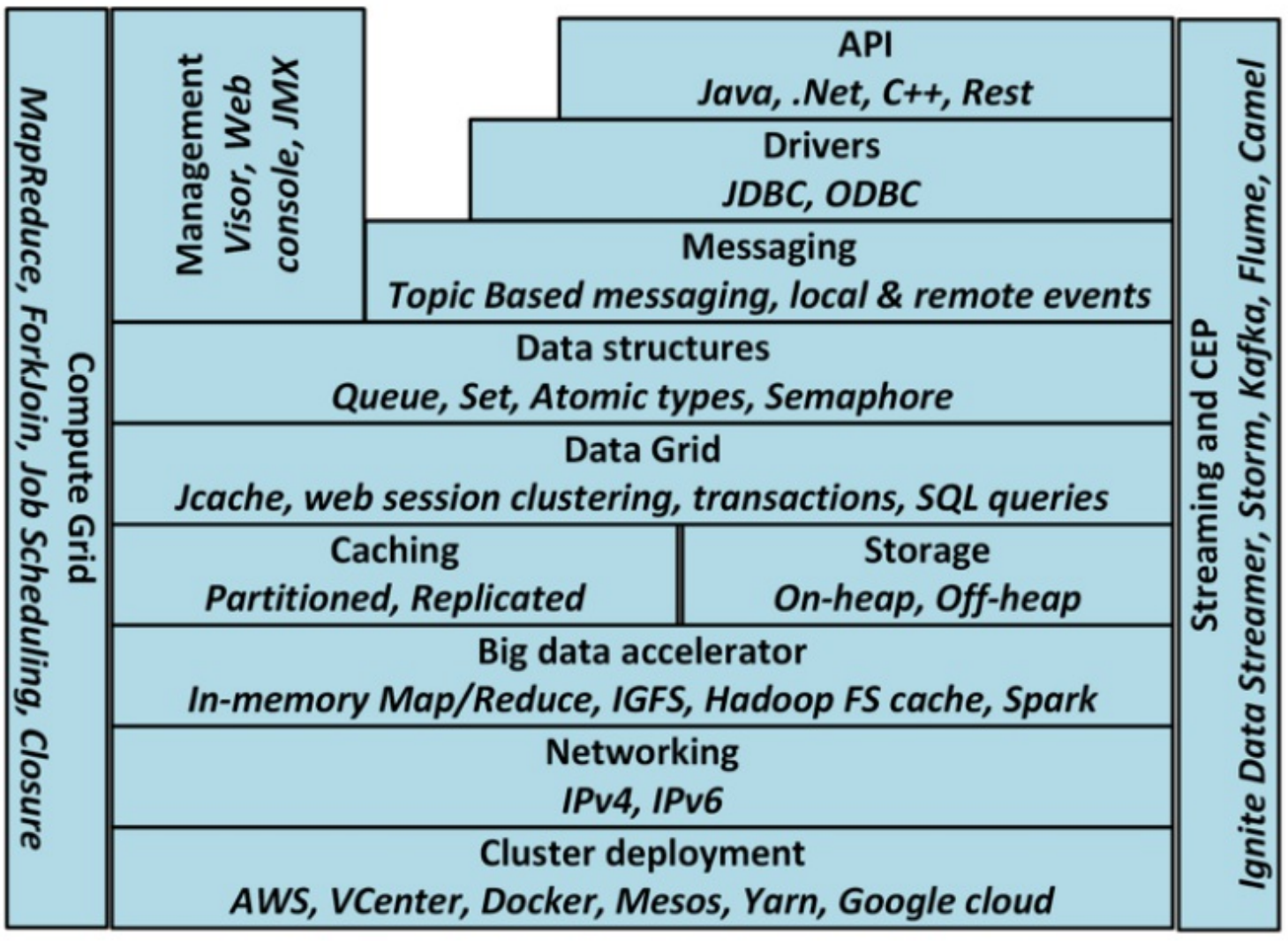
Структура ignite

Chapter two: Architecture overview



- все ноды равнозначны, нет никаких мастер-нод

Node	Description
Server	Contains Data, participates in caching, computations, streaming and can be part of the in-memory Map-Reduce tasks.
Client	Provides the ability to connect to the servers remotely to put/get elements into the cache. It can also store portions of data (near cache), which is a smaller local cache that stores most recently and most frequently accessed data.

Топология кластера

- Клиентские ноды могут выполнять рассчеты, более того можно создать клстер в котором серверные ноды будут только хранить данные, а клинентские ноды их обрабатывать.

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME

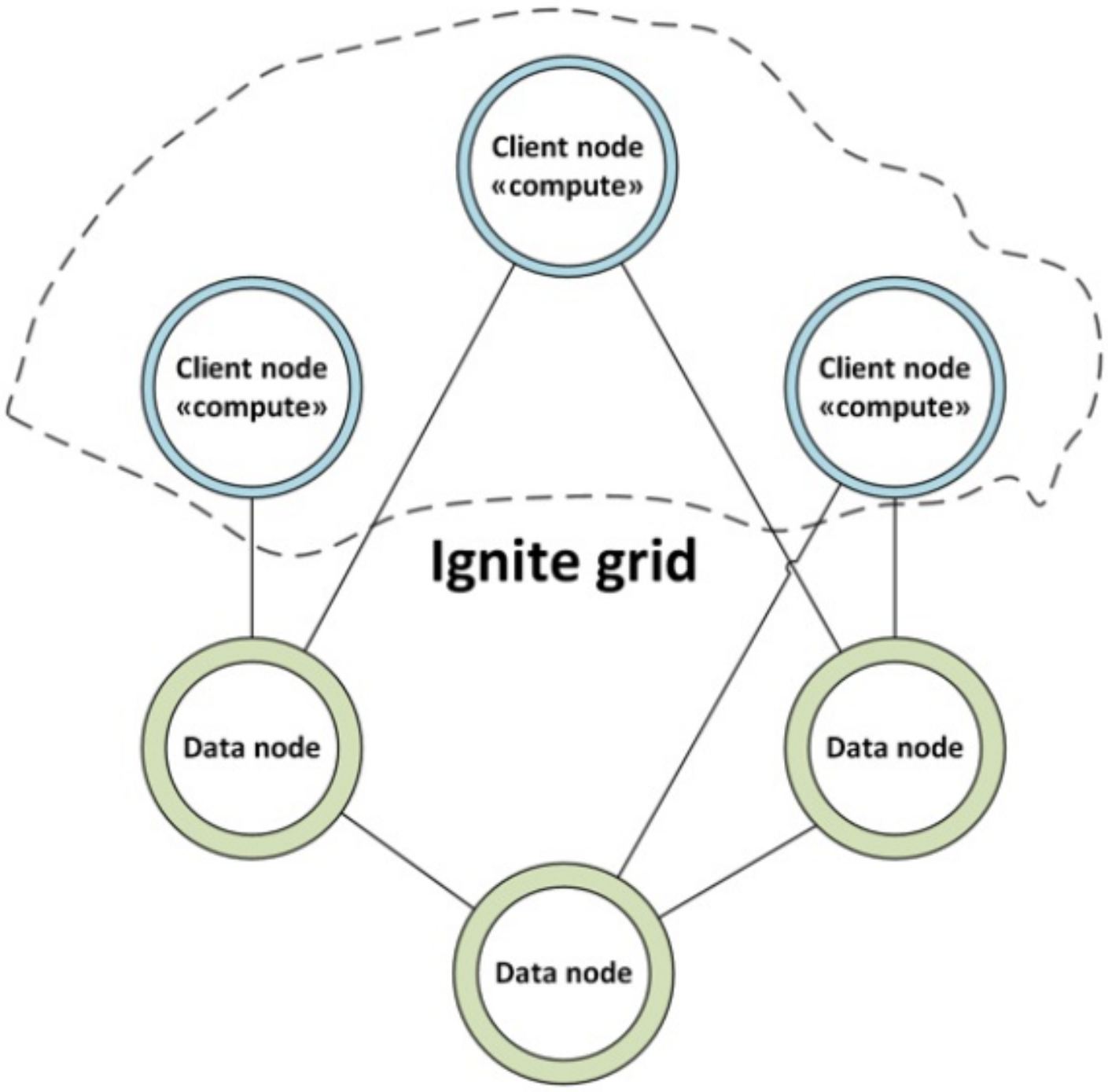


Figure 2.3. Client cluster group

- Ноды ignite могут быть

1. embedded

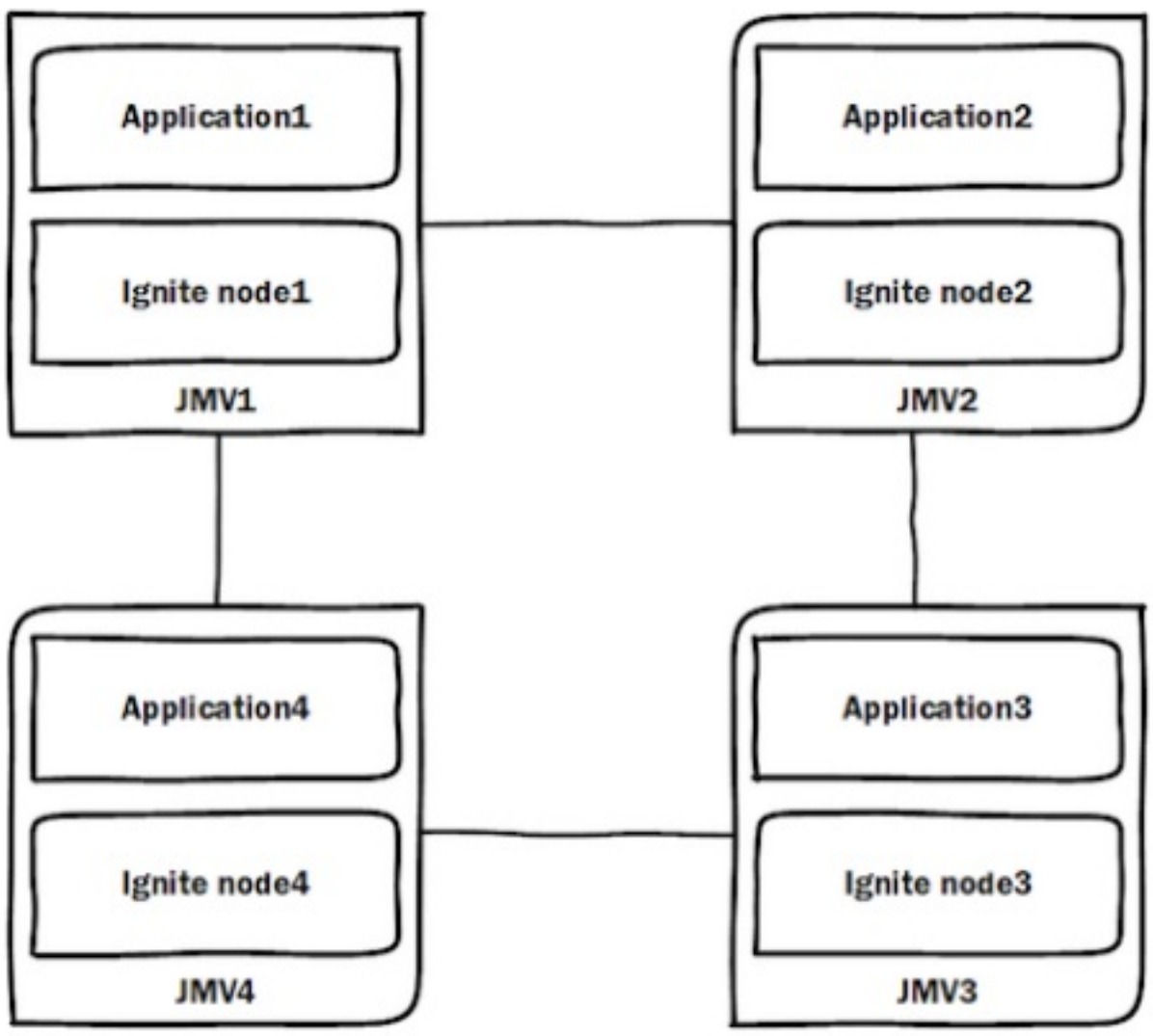


Figure 2.4. Embedded with the application

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME

2. in separate jvm

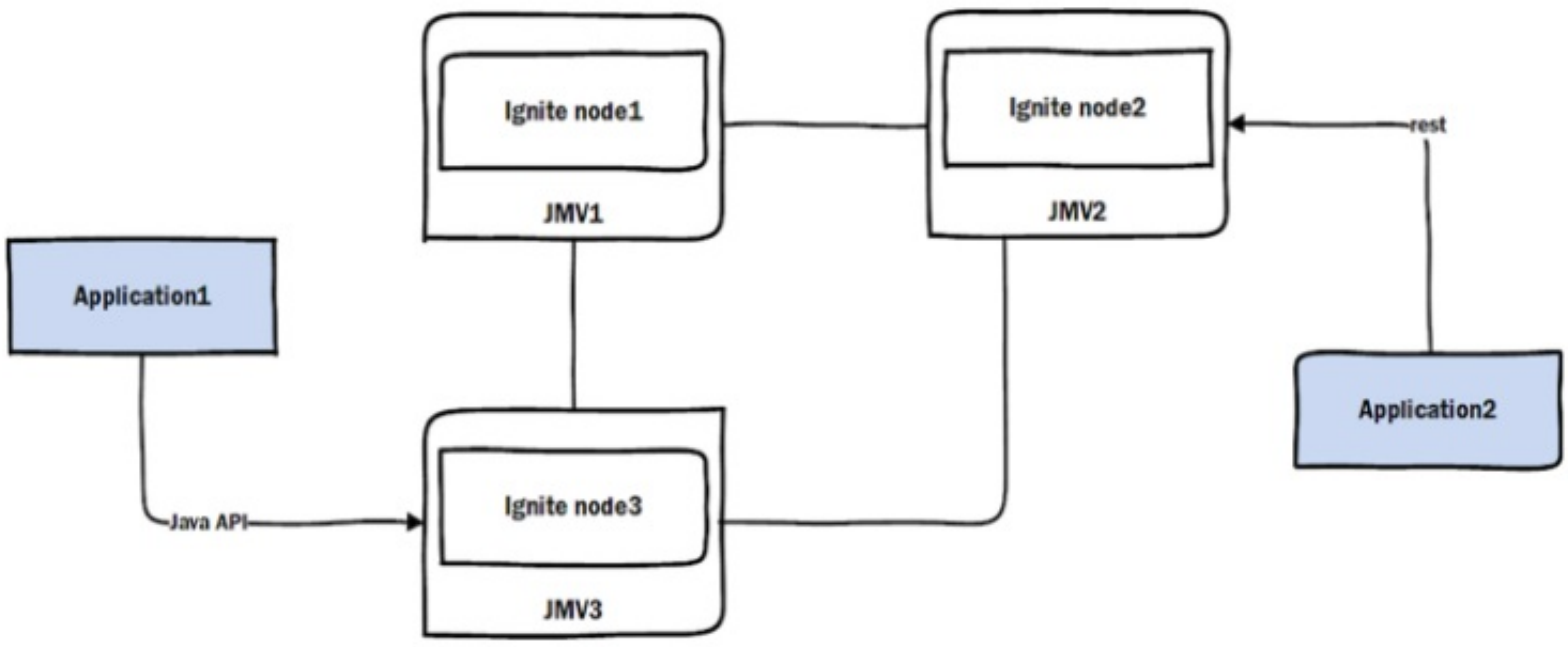


Figure 2.6. Real cluster topology

Топология кеша

- Типы топологий
    1. Partitioned (топология по умолчанию) - данные кеша разбиты с избыточностью по нодам кластера.

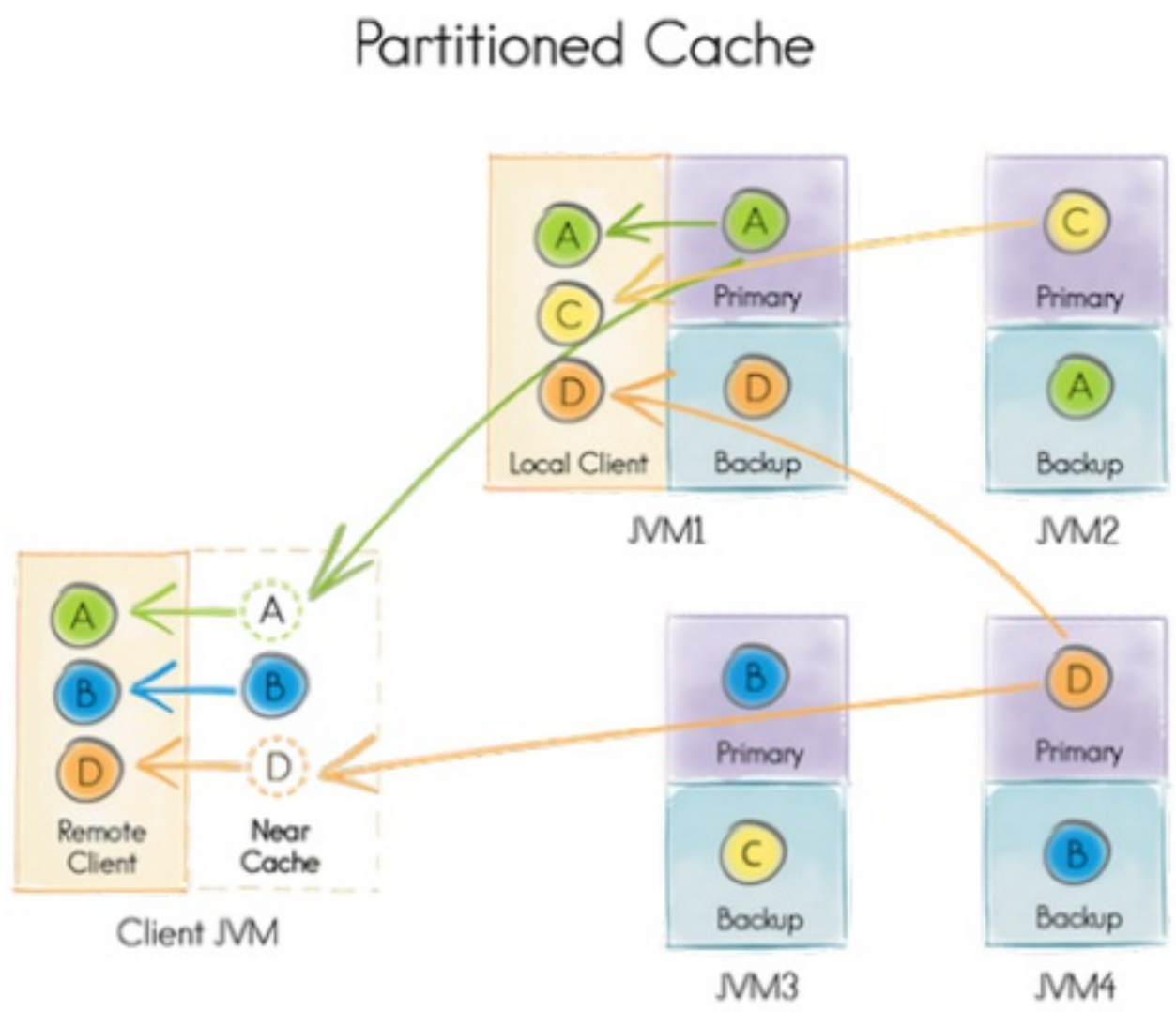


Figure 2.8. Partitioned caching topology

2. Replicated - все данные хранятся на всех нодах - очень быстрый доступ к данным. Недостаток - каждая нова порция данных должна быть скопирована на все ноды - медленные апдейты. Подходит для быстрого доступа к небольшим редко изменяемым данным.

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME

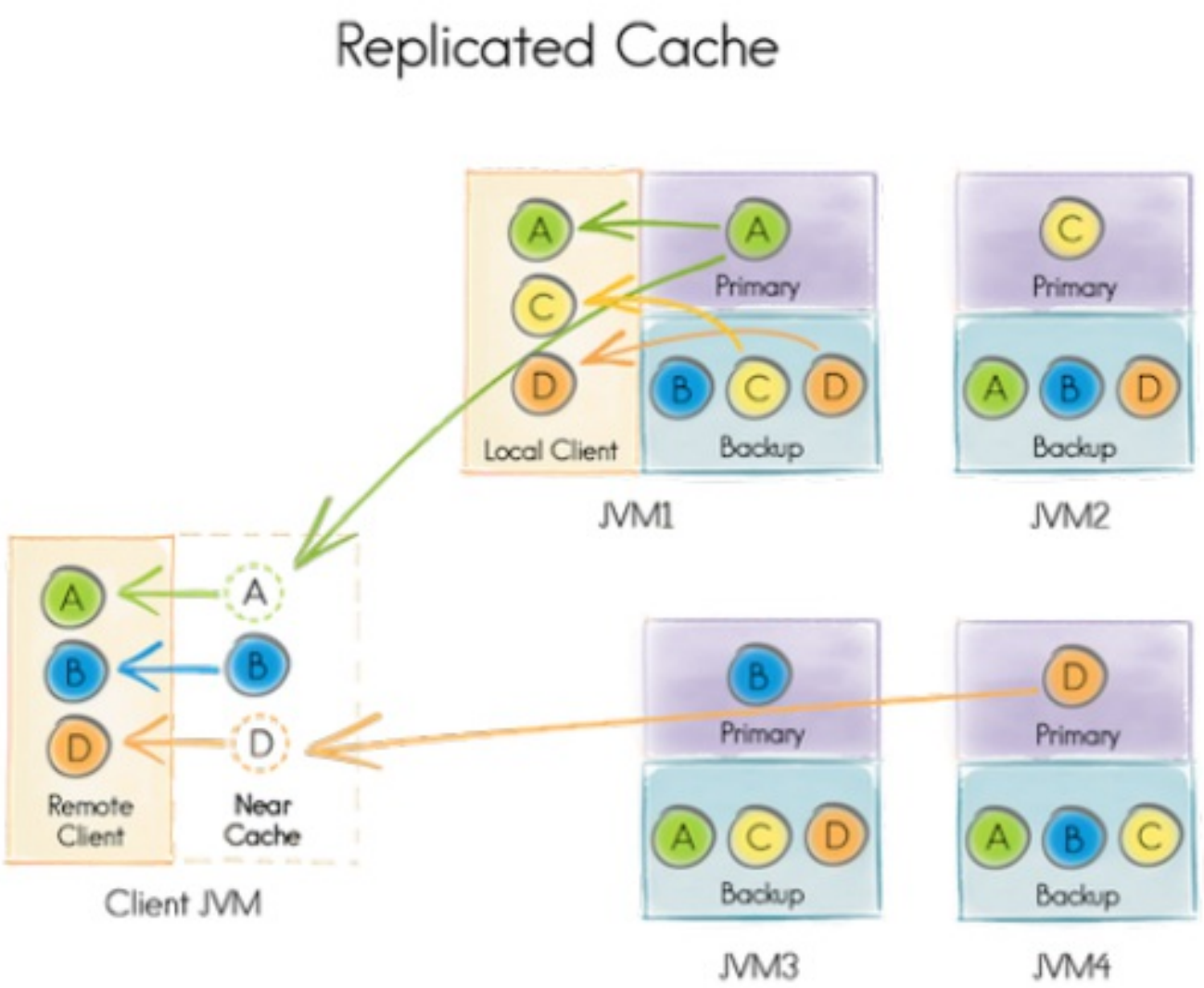


Figure 2.9. Replicated caching topology

3. Local - очень быстрый и хорошо подходит для read\wrtite-through кеша.

Стратегии использования кеша

1. cache-aside - приложение само пишет данные в БД и обновлять кеш. Например если в кеше нет данных то они читаются приложением и записываются в кеш. При обновлении данных в БД приложение само обновляет кеш.
2. Read-through and Write-through - приложение пишет и читает данные только из кеша. Кеш сам взаимодействует с БД.

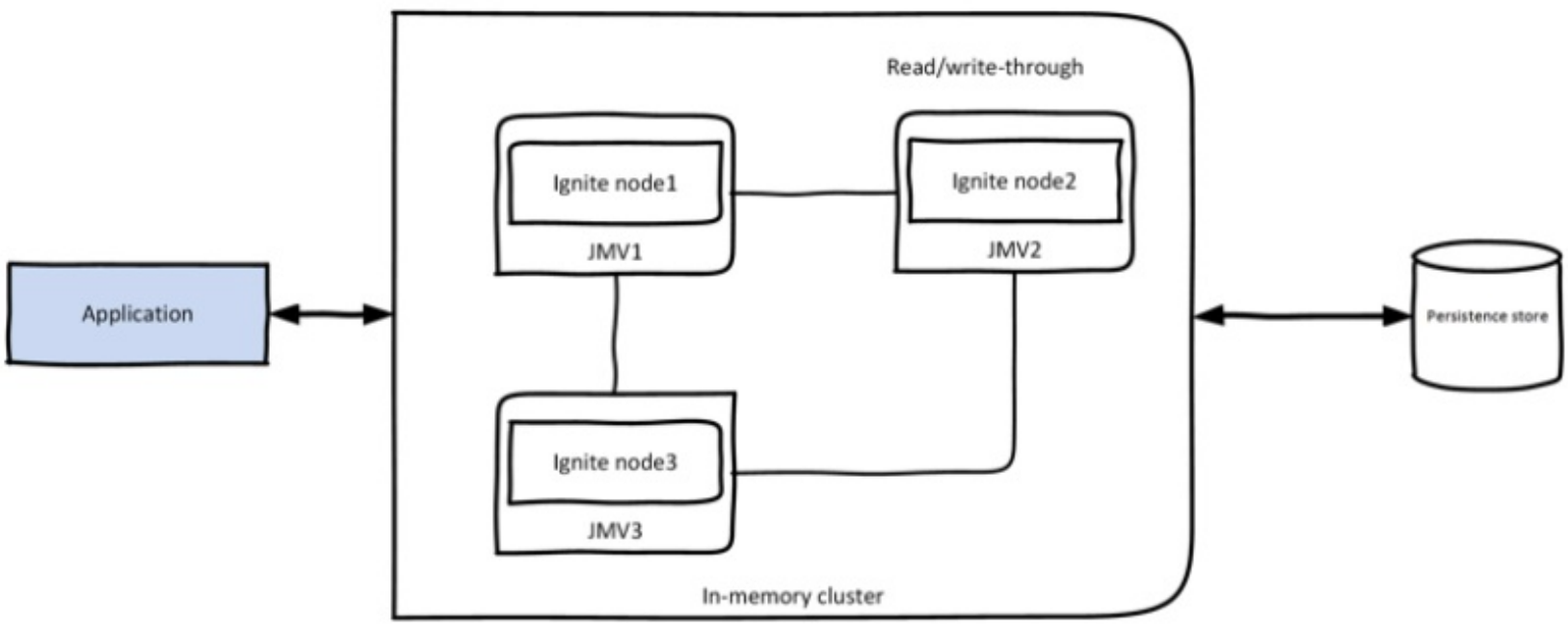


Figure 2.12. Read/Write-through

3. Write behind - операции записи аггрегируются и асинхронно отправляются в дисковое хранилище.

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME

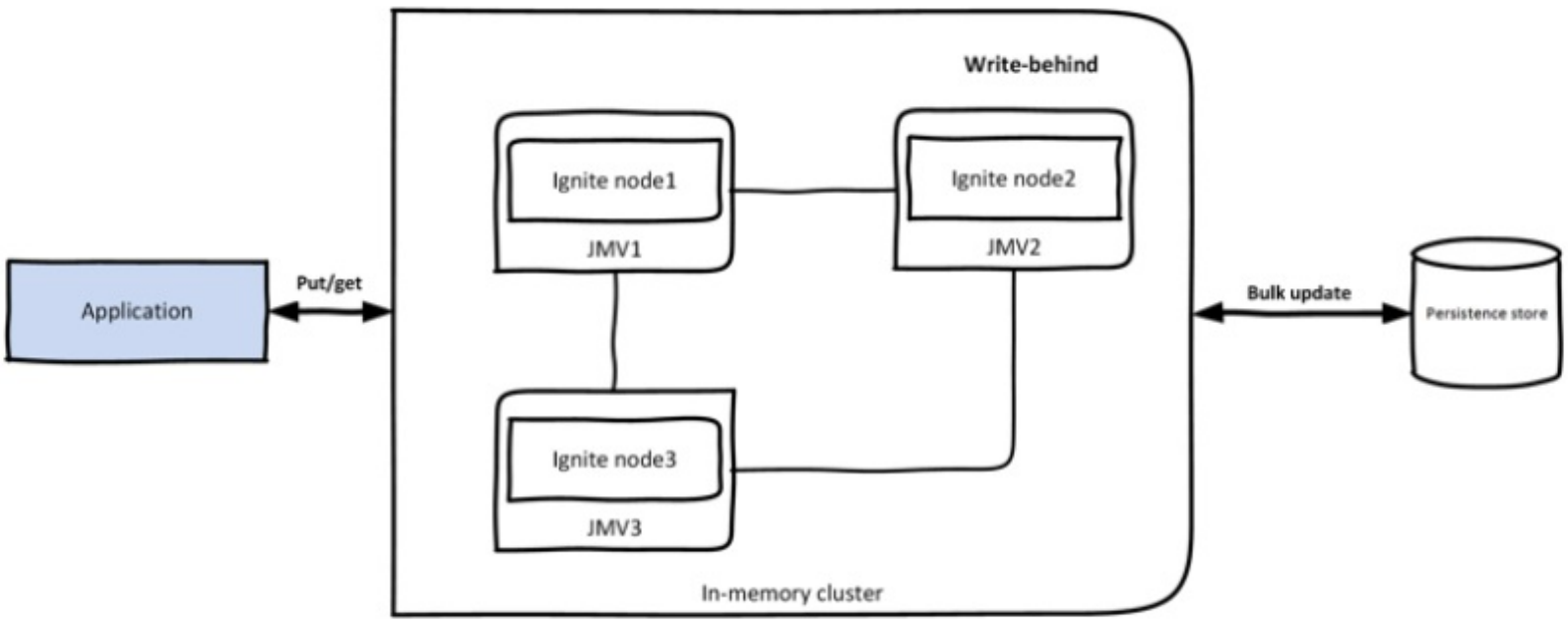
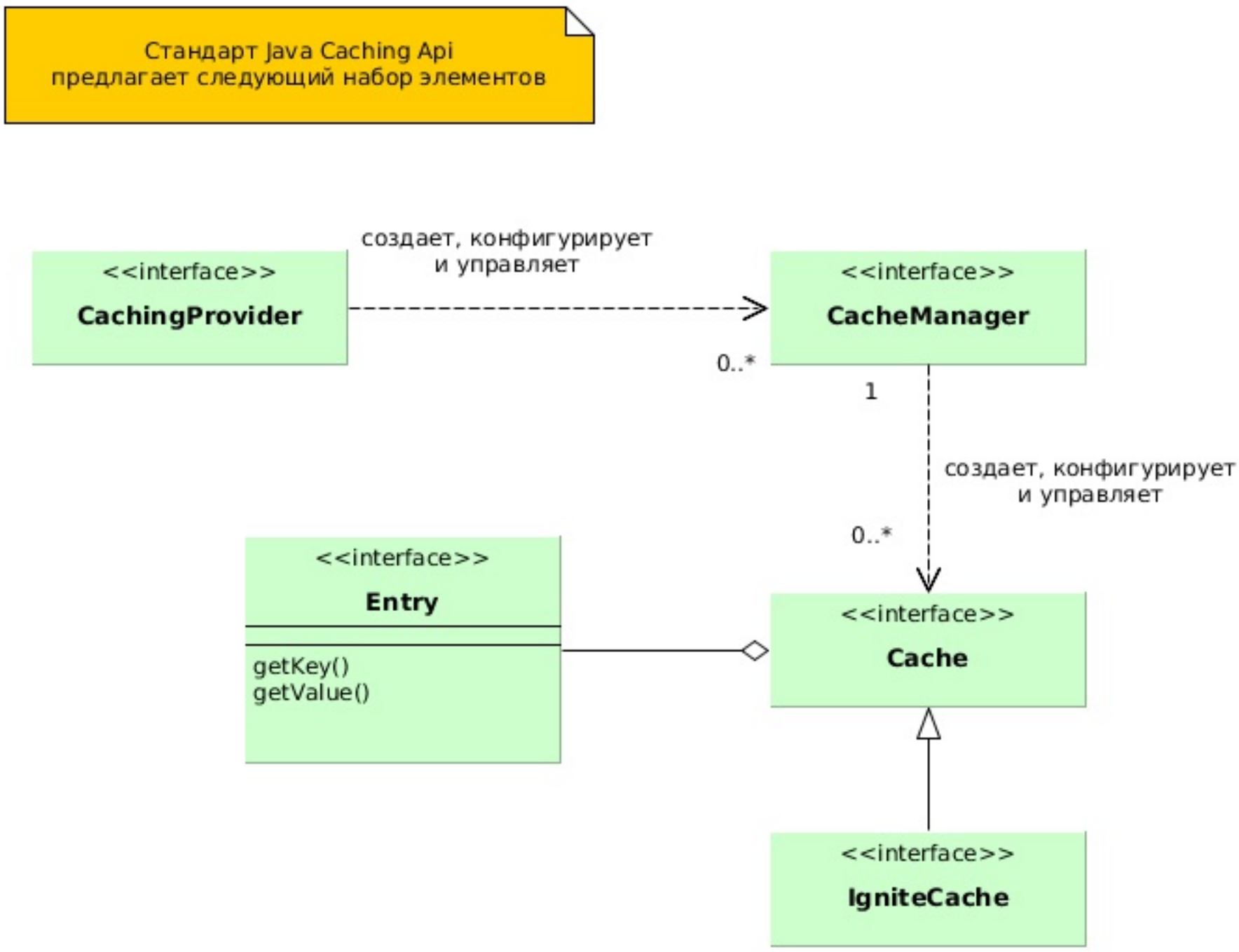


Figure 2.13. Write behind

Модель данных

- Ignite использует хранилище типа key-value.



CAP

```
1. Consistency - все ноды имеют одни и теже данные.
2. Availability - клиент может читать и писать
3. Partitioning tolerance - система поддерживает партицирование, при выходе одной ноды з строя
система все еще работоспособна
```

CA - базы данных, AP - cassandra и тп.

- Ignite modes:
  1. Transactional - можно группировать коммманды (DML) в транзакцию и коммтить ее, будут использоваться пессиместические

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME

блокировки

2. Atomic - каждая комманда (DML) может выполниться успешно либо не успешно, блокировок нет.

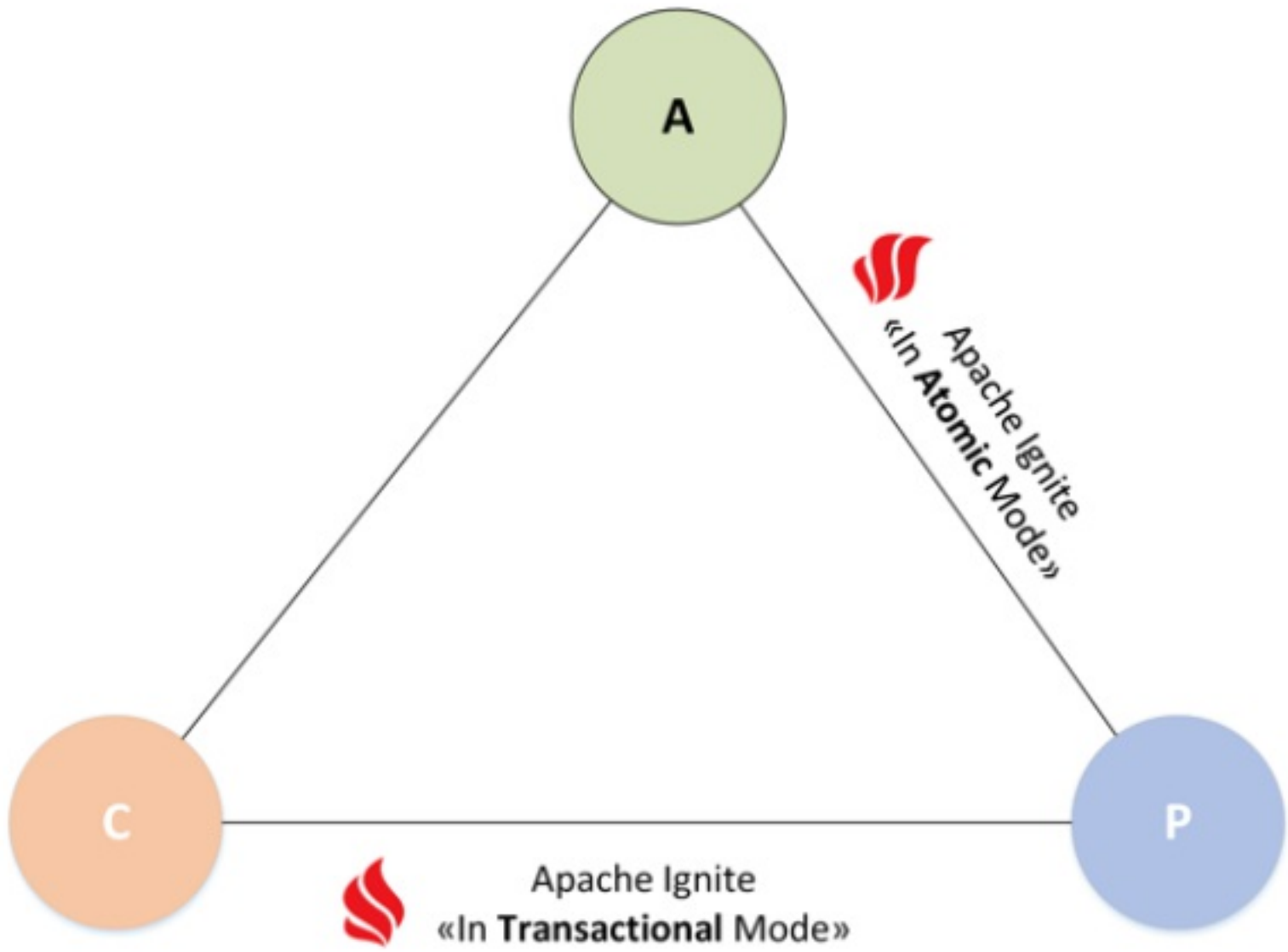


Figure 2.17. Ignite position in CAP theorem

Cluster groups

- В ignite все ноды равноценны но можно создать логическую группу. Например сгруппируем ноды обслуживающие myCache:

```
IgniteCluster cluster = ignite.cluster();
// All the data nodes responsible for caching data for "myCache".
ClusterGroup dataGroup = cluster.forDataNodes("myCache");
```

- Ноды в Ignite можно также сгруппировать по аттрибутам - master,worker,data nodes

```
//запуск мастер-ноды
IgniteConfiguration cfg = new IgniteConfiguration();
Map<String, String> attrs = Collections.singletonMap("ROLE", "master");
cfg.setUserAttributes(attrs);
// Start Ignite node.
Ignite ignite = Ignition.start(cfg);
//далее можно получить все мастер ноды
IgniteCluster cluster = ignite.cluster();
ClusterGroup workerGroup = cluster.forAttribute("ROLE", "master");
Collection<GridNode> workerNodes = workerGroup.nodes();
```

Выполнение SQL-запросов

- В партицированном кластере запросы выполняются в виде map-reduce работ
- В реплицированном кластере запросы выполняются используя H2 бд

Асинхронные методы

- Асинхронне методы возвращают IgniteFuture который имеет метод get - это тн promise - обещание вернуть результат. Метод IgniteFuture.listen(FunctionalInterface) неблокирующий и выполняется по готовности. Примеры кода в project-installation модуле.

Resilence

- Термин означает скорейшее восстановление после сбоя, например нода будет

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME

API

автоподключаться к кластеру в случае сбоя.

Name	Description
org.apache.ignite.IgniteCache	The main cache interface, the entry point for all Data Grid API's. The interface extends javax.cache.Cache interface.
org.apache.ignite.cache.store.CacheStore	Interface for cache persistence storage for read-through and write-through behavior.
org.apache.ignite.cache.store.CacheStoreAdapter	The cache storage convenience adapter, implements interface CacheStore. It's provides default implements for bulk operations, such as writeAll and readAll.
org.apache.ignite.cache.query.ScanQuery	Scan query over cache entries.
org.apache.ignite.cache.query.TextQuery	Query for Lucene based fulltext search.
org.apache.ignite.cache.query.SqlFieldsQuery	SQL Fields query. This query can return specific fields of data based on SQL clause.
org.apache.ignite.transactions.Transaction	Ignite cache transaction interface, have a default 2PC behavior and supports different isolation levels.
org.apache.ignite.IgniteFileSystem	Ignite file system API, provides a typical file system view on the particular cache. Very similar to HDFS, but only on in-memory.
org.apache.ignite.IgniteDataStreamer	Data streamer is responsible for streaming external data into the cache. This streamer will stream data concurrently by multiple internal threads.
org.apache.ignite.IgniteCompute	Defines compute grid functionality for executing tasks and closures over nodes ClusterGroup.
org.apache.ignite.services.Service	An instance of the grid managed service. Whenever a service is deployed, Ignite will automatically calculate how many instances of this service should be deployed on each node within the cluster.
org.apache.ignite.IgniteMessaging	An interface that provides functionality for topic-based message exchange among nodes defined by ClusterGroup.

© Copyright Tuesday, Aug 1, 2017, 8:13 PM by COMPANYNAME