Работа с большими данными

SELEZNEV ARTEM
HEAD OF CVM ANALYTICS @ MAGNIT

tg: @SeleznevArtem

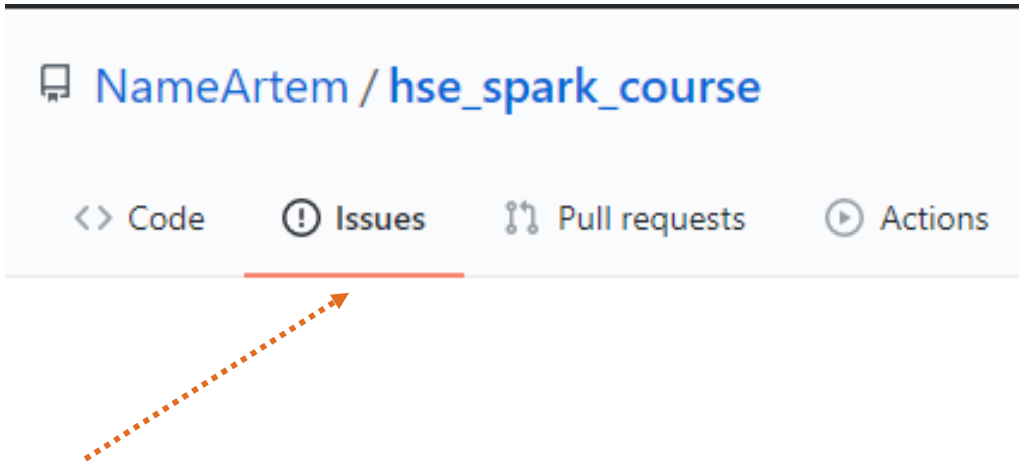
/NameArtem

/seleznev-artem

/seleznev.artem.info

https://github.com/NameArtem/hse_spark_course

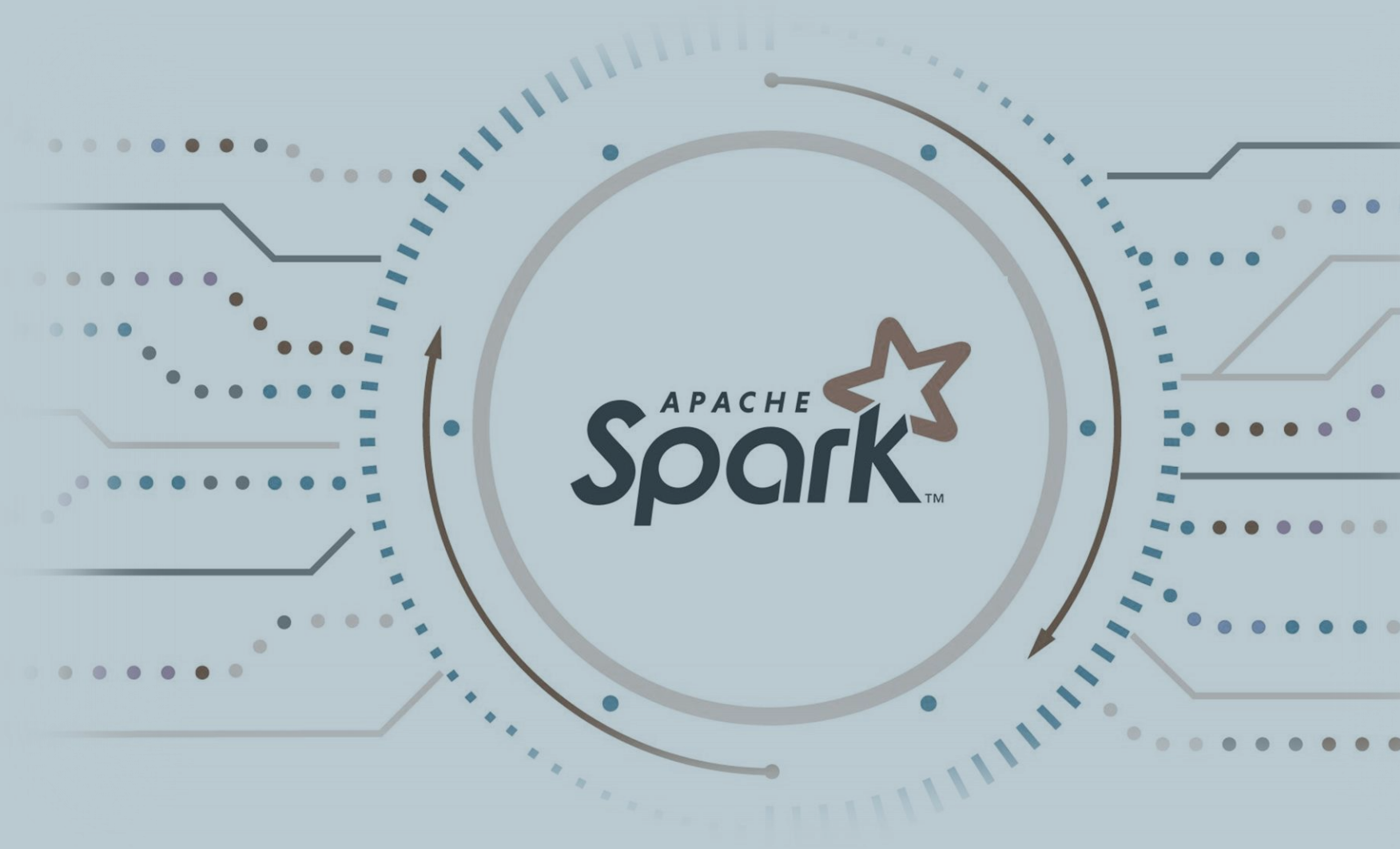
https://github.com/NameArtem/hse_spark_course



https://github.com/NameArtem/hse_spark_course

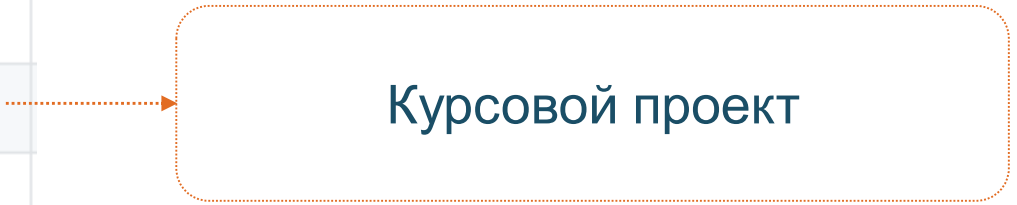
https://cutt.ly/2lSpyZs

О КУРСЕ



№	Тема занятия
1	Введение в распределенные вычисления
2	Apache Spark (RDD) (+ FuncProg на Python)
3	Spark SQL. Анализ больших данных
4	Подробнее о модели вычислений Spark. Знакомство со Scala
5	Spark ML
6	Рекомендательные системы на Spark
7	Ещё о системах рекомендаций. О Spark UDF. Spark Structure Streaming (+ интеграция со Spark ML)
8	Модели в прод. Управленеи кластеровм

№	Тема занятия
1	Введение в распределенные вычисления
2	Apache Spark (RDD) (+ FuncProg на Python)
3	Spark SQL. Анализ больших данных
4	Подробнее о модели вычислений Spark. Знакомство со Scala
5	Spark ML
6	Рекомендательные системы на Spark
7	Ещё о системах рекомендаций. О Spark UDF. Spark Structure Streaming (+ интеграция со Spark ML)
8	Модели в прод. Управленеи кластеровм



Курсовой проект

Проверка качества данных (в таблице / файле):

- Разработать процесс качества данных (таблицы)
- Предикты на линейные и бинарные данные
- Отбор репрезентативного сэмпла, который показывает такое же распределение, как есть в каждой колонке
- авто определение join
  (найти колонки самостоятельно и сделать join (правильно))

UNIQUENESS
- Existence of unique values for a specific data attribute within a table
- **Example:** Data attribute which has duplicated values will not have the highest score on uniqueness dimension

CONSISTENCY
- Logical coherence within data of a system that free them from contradiction
- **Example:** 'Order Fulfilment Date' should be after the 'Order Creation Date'

INTEGRITY
- Existence of data values in reference table(s) from different system(s)
- **Example:** 'Product ID' values should exist in the Product reference table

COMPLETENESS
- Existence of values in a specific data attribute (data field)
- **Example:** Data attribute with missing values is not complete

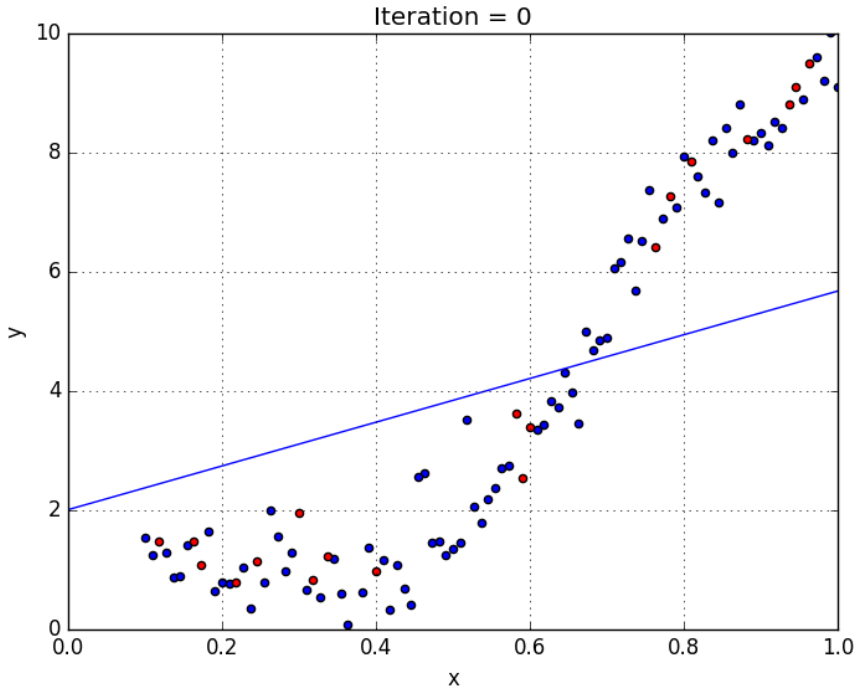
TIMELINESS
- Degree to which data is representative of current business conditions (updated and available)
- **Example:** A plan price change not updated on the day it was issued creates a breach of timeliness

CONFORMITY
- Data are valid if it conforms to the syntax (format, type, range) of its definition
- **Example:** 'Landline Number' should be numeric with 8 digits

Курсовой проект

Проверка качества данных (в таблице / файле):

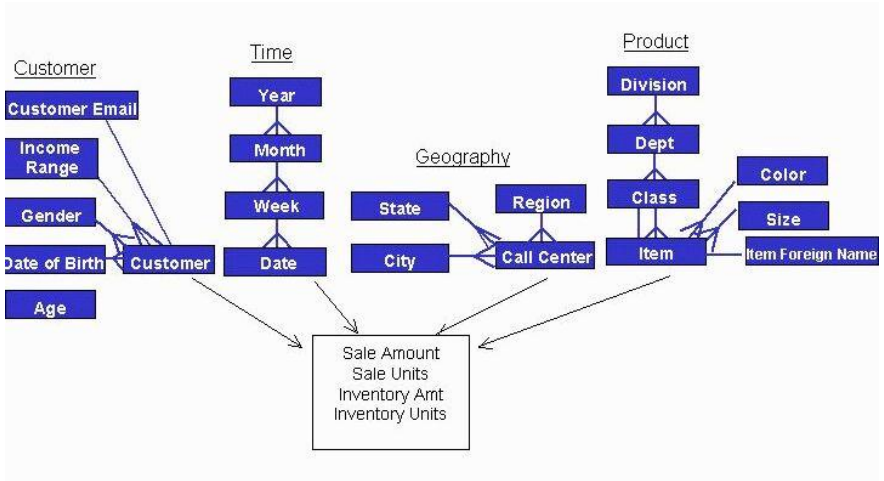
- Разработать процесс качества данных (таблицы)
- Предикты на линейные и бинарные данные
- Отбор репрезентативного сэмпла, который показывает такое же распределение, как есть в каждой колонке
- авто определение join
  (найти колонки самостоятельно и сделать join (правильно))



Курсовой проект

Проверка качества данных (в таблице / файле):

- Разработать процесс качества данных (таблицы)
- Предикты на линейные и бинарные данные
- Отбор репрезентативного сэмпла данных, который показывает такое же распределение, как есть в каждой колонке
- авто определение join
  (найти колонки самостоятельно и сделать join (правильно))



Курсовой проект

Проверка качества данных (в таблице / файле):

- Разработать процесс качества данных (таблицы)
- Предикты на линейные и бинарные данные
- Отбор репрезентативного сэмпла данных, который показывает такое же распределение, как есть в каждой колонке
- авто определение join (найти колонки самостоятельно и сделать join (правильно))

???

ИНСТРУМЕНТЫ

Python

Linux

Git

Spark

ИНСТРУМЕНТЫ

Python

Linux

Git

Spark

РАБОЧАЯ СРЕДА КУРСА

Инфраструктура курса

- Локальный кластре на Docker
- DataBricks Community

или установить самостоятельно

CLUSTER DEEPER…

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
    - Apache Hadoop3

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
  - Apache Hadoop3
    - Apache Spark 3

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
  - Apache Hadoop3
    - Apache Spark 3
      - Apache Drill

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
  - Apache Hadoop3
    - Apache Spark 3
      - Apache Drill
        - JupyterHub + Kernel

ЖИЗНЬ НА КЛАСТЕРЕ

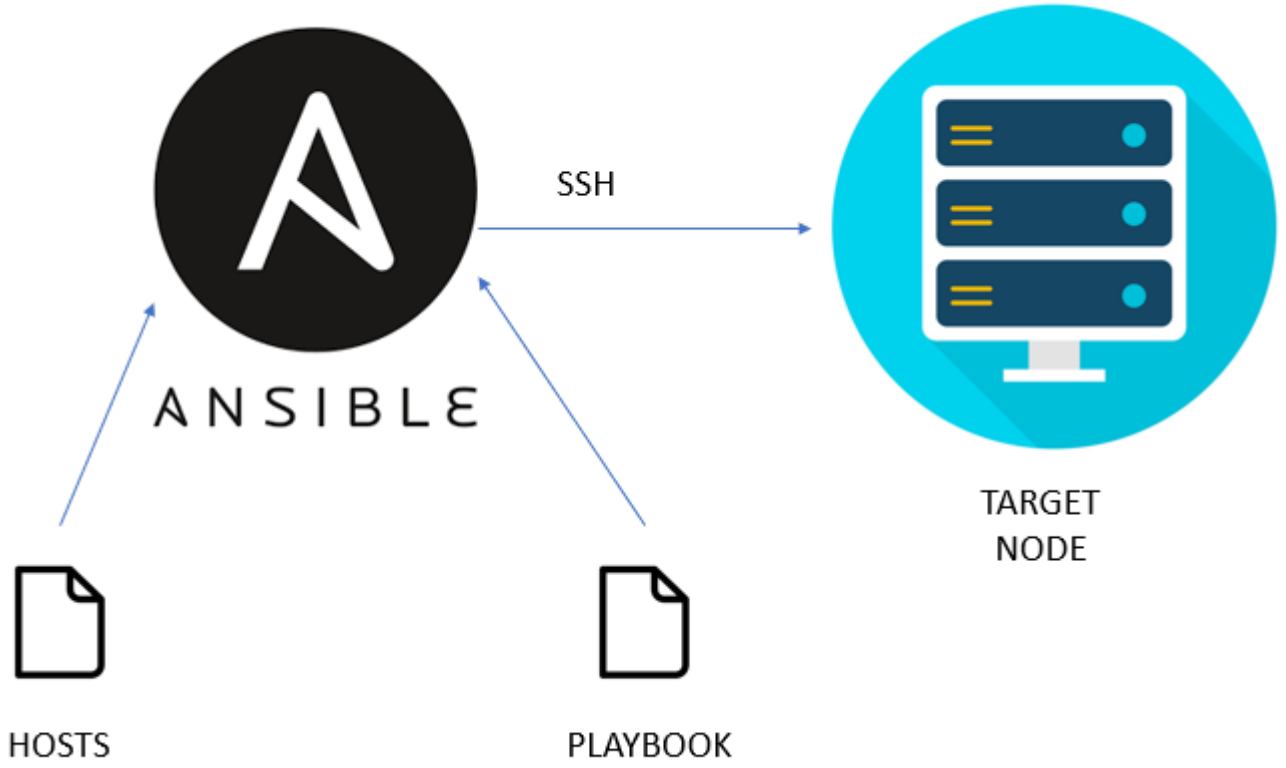
- Ansible
  - Apache Hadoop3
    - Apache Spark 3
      - Apache Drill
        - JupyterHub + Kernel
          - Feature Store

ЖИЗНЬ НА КЛАСТЕРЕ

- Linux Centos 7
- Java 8
- Scala 12
- Python 3
- Публичные IP / Внешние IP

IP адрес	Имя узла	Роли
10.0.0.2	cnt-cls-m1	NameNode, ResourceManager
10.0.0.3	cnt-cls-s1	SecondaryNameNode, DataNode, NodeManager
10.0.0.4	cnt-cls-s2	DataNode
10.0.0.5	cnt-cls-s3	DataNode

ANSIBLE



https://tproger.ru/translations/ansible-how-to-use/

КЛАСТЕР

```
yum install -y net-tools openssh-server
yum install ntp ntpdate ntp-doc -y
yum install openssl
yum install -y zookeeper
yum install -y zookeeper-server
```

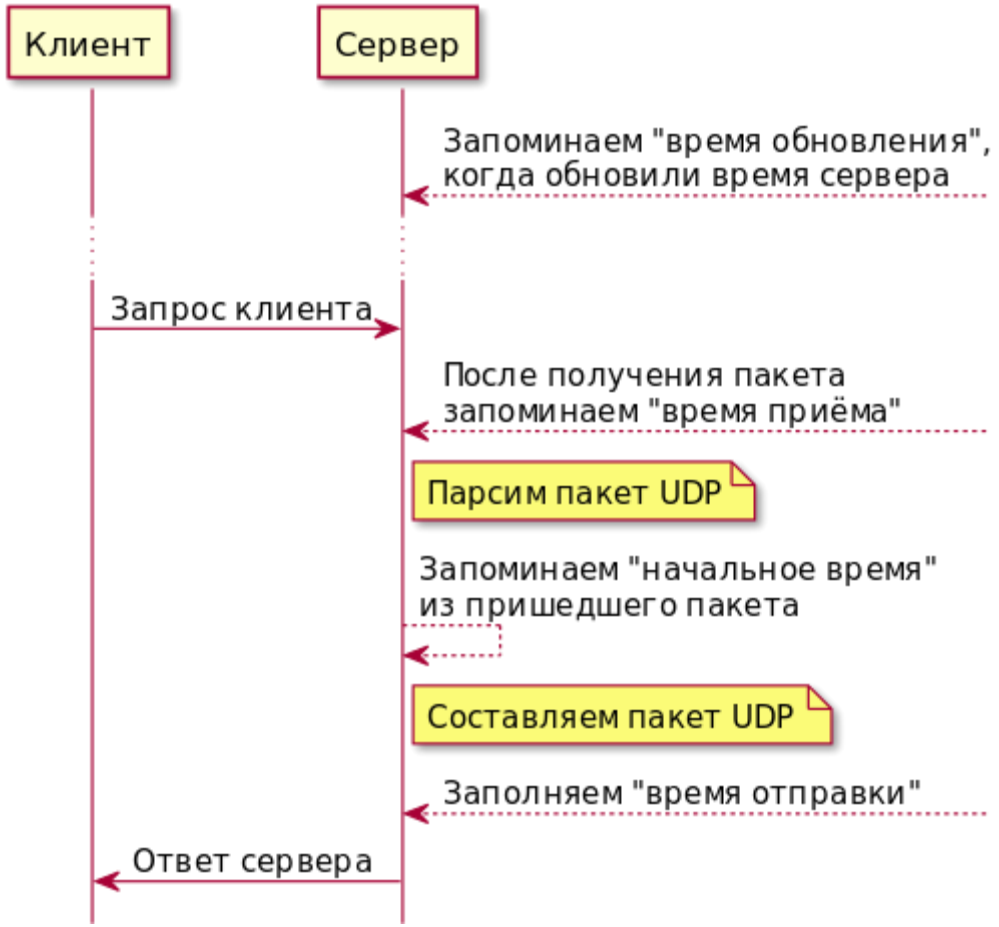
КЛАСТЕР

```
yum install -y net-tools openssh-server
yum install ntp ntpdate ntp-doc -y
yum install openssl
yum install -y zookeeper
yum install -y zookeeper-server
```

Clock Drift
Clock Skew

КЛАСТЕР

```
yum install -y net-tools openssh-server
yum install ntp ntpdate ntp-doc -y
yum install openssl
yum install -y zookeeper
yum install -y zookeeper-server
```



https://clck.ru/R87Kx

КЛАСТЕР

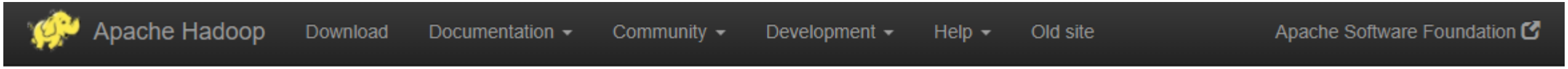
SSH KEY

```
ssh-keygen -t rsa -b 4096
ssh-copy-id *имя узла*
```

ADD USER

```
sudo groupadd hadoop
sudo useradd -d /home/hadoop -g hadoop hadoop
sudo passwd hadoop
```

HADOOP



Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
3.1.4	2020 Aug 3	source (checksum signature)	binary (checksum signature)	Announcement
3.3.0	2020 Jul 14	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
2.10.0	2019 Oct 29	source (checksum signature)	binary (checksum signature)	Announcement
3.2.1	2019 Sep 22	source (checksum signature)	binary (checksum signature)	Announcement
2.9.2	2018 Nov 19	source (checksum signature)	binary (checksum signature)	Announcement

HADOOP

```
Sep  2 15:50 capacity-scheduler.xml
Sep  2 15:49 configuration.xsl
Sep  2 15:50 container-executor.cfg
Sep  2 15:49 core-site.xml
Sep  2 15:50 hadoop-env.cmd
Sep  2 15:50 hadoop-env.sh
Sep  2 15:50 hadoop-metrics2.properties
Sep  2 15:50 hadoop-metrics.properties
Sep  2 15:50 hadoop-policy.xml
Sep  2 15:50 hdfs-site.xml
Sep  2 15:50 httpfs-env.sh
Sep  2 15:50 httpfs-log4j.properties
Sep  2 15:50 httpfs-signature.secret
Sep  2 15:50 httpfs-site.xml
Sep  2 15:50 kms-acls.xml
Sep  2 15:50 kms-env.sh
Sep  2 15:50 kms-log4j.properties
Sep  2 15:50 kms-site.xml
Sep  2 15:50 log4j.properties
Sep  2 15:50 mapred-env.cmd
Sep  2 15:49 mapred-env.sh
Sep  2 15:50 mapred-queues.xml.template
Sep  2 15:49 mapred-site.xml
Sep  2 15:50 mapred-site.xml.template
Sep  2 15:49 masters
Sep  4 20:12 slaves
Sep  2 15:50 ssl-client.xml.example
Sep  2 15:50 ssl-server.xml.example
Sep  2 15:49 yarn-env.cmd
Sep  2 15:50 yarn-env.sh
Sep  2 15:50 yarn-site.xml
```

- core-site.xml
- hdfs-site.xml
- mapred-site.xml
- yarn-site.xml

HADOOP

ADD DIRS

```
mkdir -p $HADOOP_HOME/tmp
mkdir -p $HADOOP_HOME/hdfs/name
mkdir -p $HADOOP_HOME/hdfs/data
```

COPY SETTINGS

```
scp ~/.bashrc cnt-cls-m2:~/ #(для всех 2, 3, 4)
scp -r /opt/hadoop3/etc/hadoop/ cnt-cls-m2:/opt/hadoop3/etc/ #(для всех 2, 3, 4)
```

HADOOP

ADD HOSTS

```
cnt-cls-m1> $HADOOP_HOME/etc/hadoop/workers
cnt-cls-m2> $HADOOP_HOME/etc/hadoop/workers
cnt-cls-m3> $HADOOP_HOME/etc/hadoop/workers
cnt-cls-m4> $HADOOP_HOME/etc/hadoop/workers
```

HADOOP

```
export HADOOP_HOME=/opt/hadoop3
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_ROOT_LOGGERi=INFO,console
export HADOOP_SECURITY_LOGGER=INFO,NullAppender
export HADOOP_INSTALL=$HADOOP_HOME
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_PREFIX=$HADOOP_HOME
export HADOOP_LIBEXEC_DIR=$HADOOP_HOME/libexec
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
export HADOOP_YARN_HOME=$HADOOP_HOME
```

HADOOP

ПЕРВЫЙ ЗАПУСК

```
hdfs namenode -format
```

```
start-dfs.sh
```

```
start-yarn.sh
```

SPARK

Lightning-fast unified analytics engine

Download **Libraries** **Documentation** **Examples** **Community** **Developers**

Download Apache Spark™

1. Choose a Spark release: 3.0.0 (Jun 18 2020)
2. Choose a package type: Pre-built for Apache Hadoop 3.2 and later
   - Pre-built for Apache Hadoop 2.7
   - Pre-built for Apache Hadoop 3.2 and later
   - Pre-built with user-provided Apache Hadoop
   - Source Code
3. Download Spark: spark-3
4. Verify this release using t ase KEYS.

Note that, Spark 2.x is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12. Spark 3.0+ is pre-built with Scala 2.12.

```
pip3 install pyspark
pip3 install py4j
```

SPARK

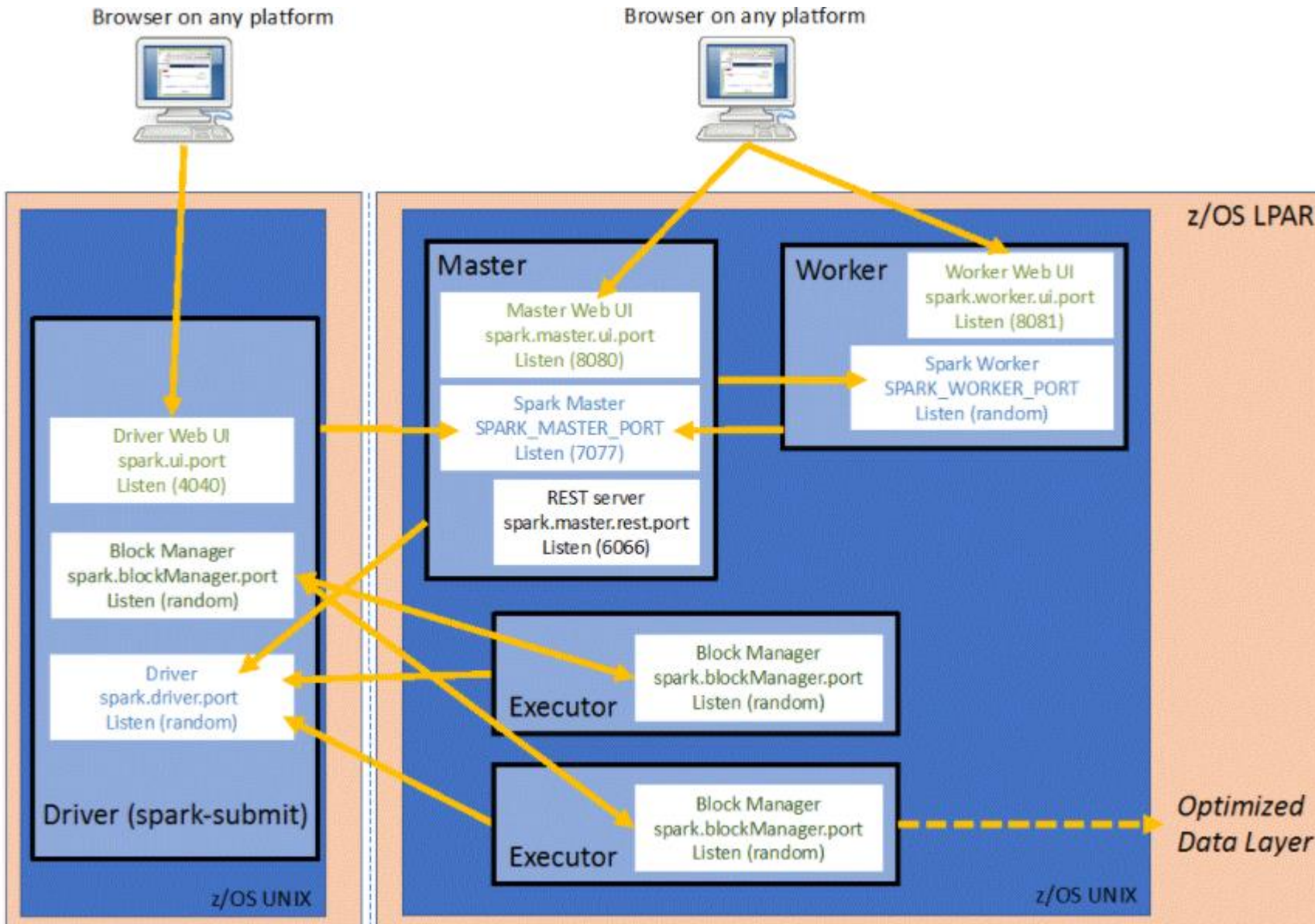
/opt/spark3/conf/spark-env.sh

```
SPARK_LOCAL_IP=cnt-cls-m1
SPARK_MASTER_IP=pub-cnt-cls-m1
SPARK_MASTER_HOST=cnt-cls-m1
SPARK_MASTER_PORT=7070
PYSPARK_PYTHON=/usr/bin/python3
PYSPARK_DRIVER_PYTHON=/usr/bin/python3
```

SPARK

/opt/spark3/conf/spark-env.sh

```
SPARK_LOCAL_IP=cnt-cls-m1
SPARK_MASTER_IP=pub-cnt-cls-m1
SPARK_MASTER_HOST=cnt-cls-m1
SPARK_MASTER_PORT=7070
PYSPARK_PYTHON=/usr/bin/python3
PYSPARK_DRIVER_PYTHON=/usr/bin/python3
```



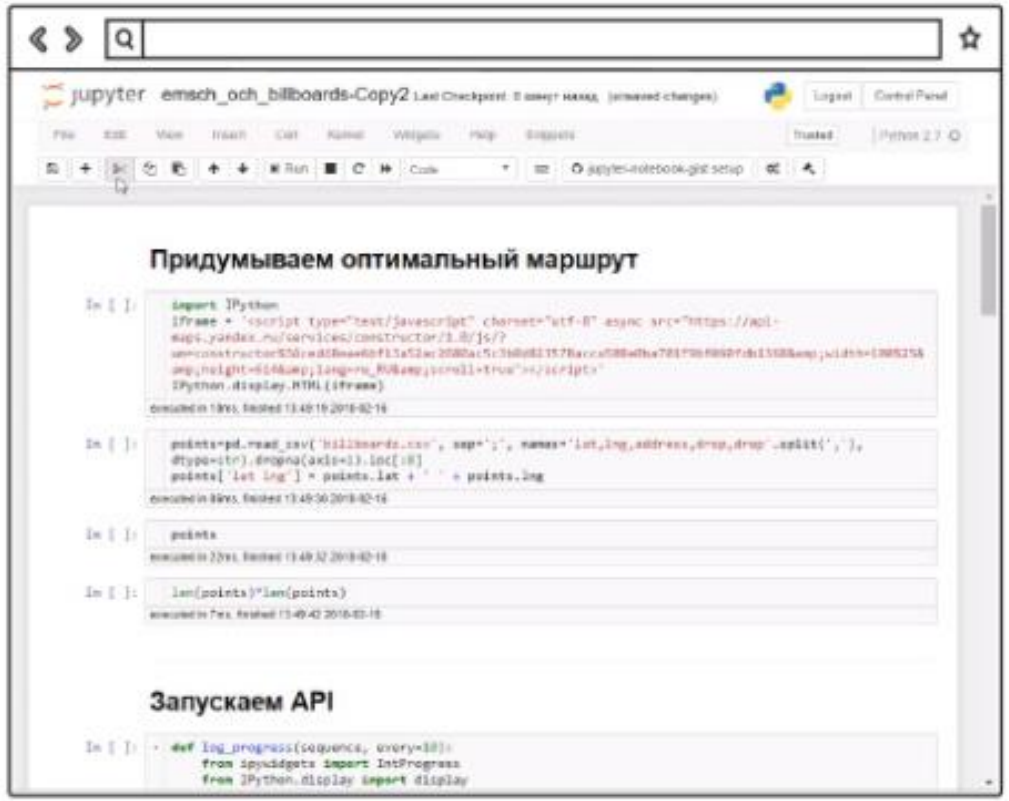
SPARK

- `start-master.sh`
- `start-slave.sh spark://cnt-cls-m1:7070` (выполнить на каждой ноде)

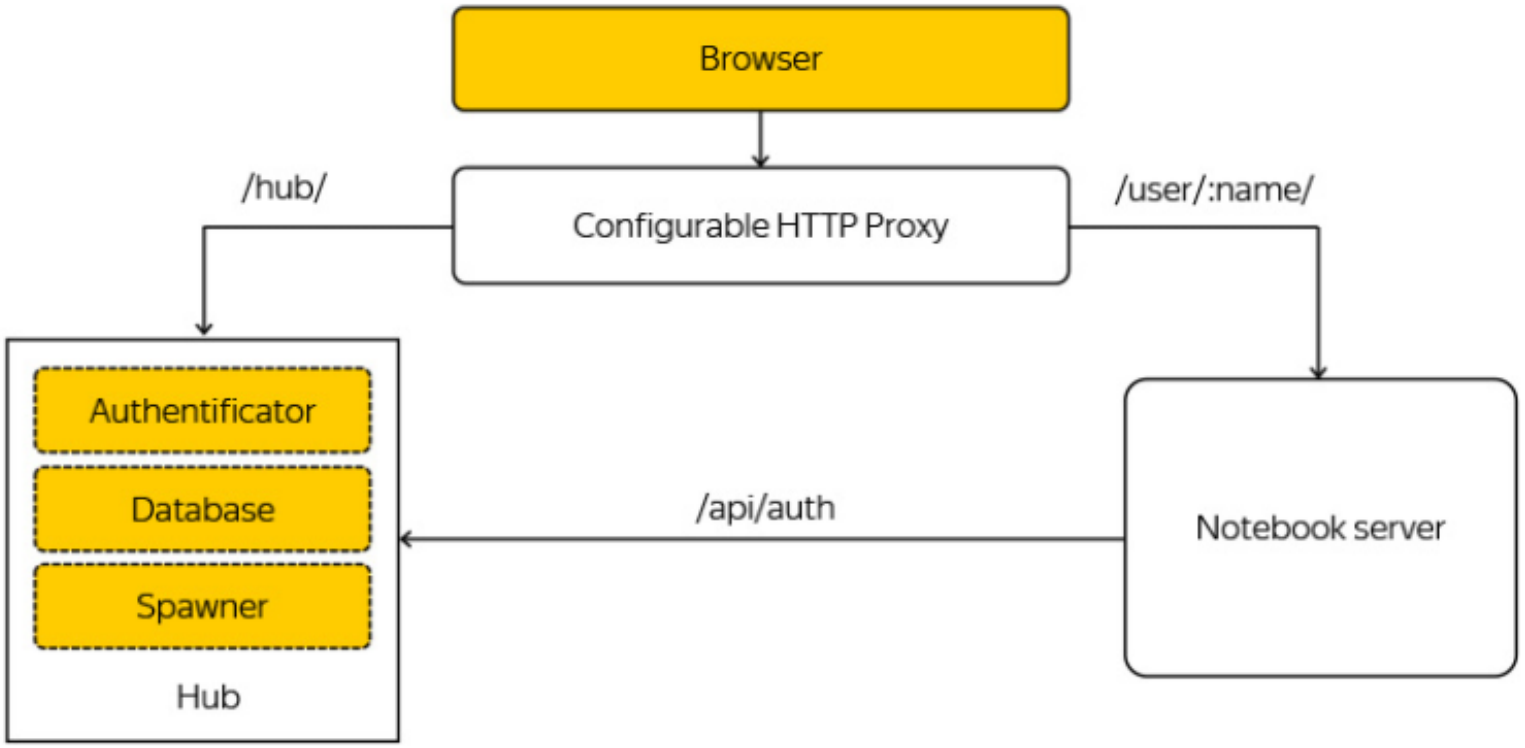
JUPYTERHUB

```
yum install install npm nodejs-legacy
pip3 install jupyterhub
npm install -g configurable-http-proxy
```

- Классический «ноутбук»
- Различные языки программирования
- Интерактивный код, легко менять на лету
- Визуализации, произвольный output



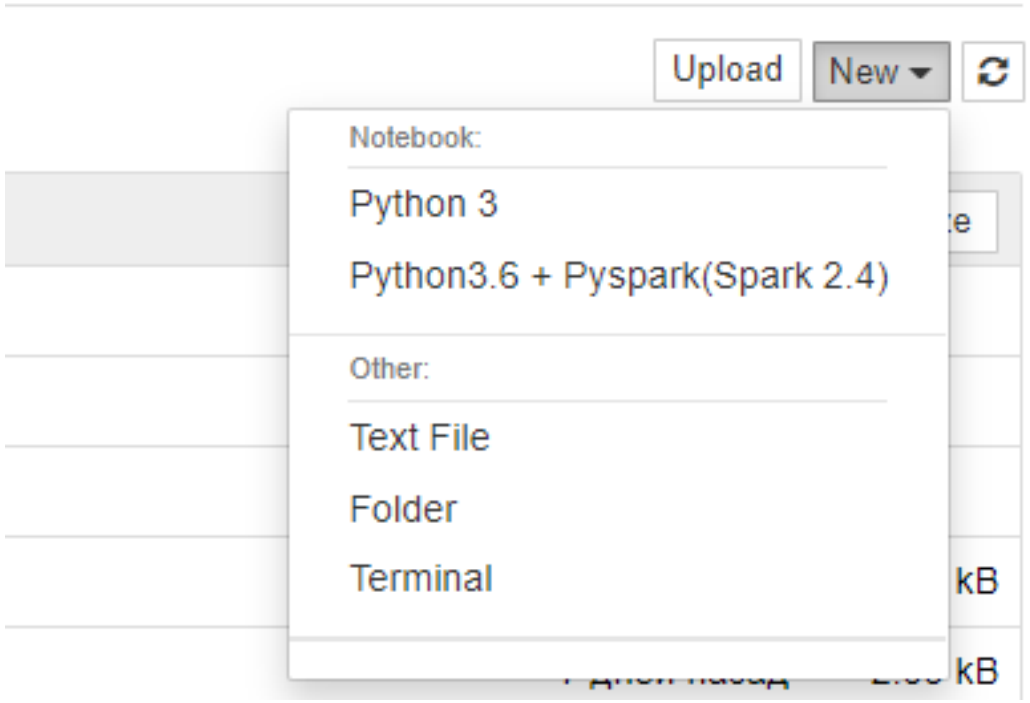
JUPYTERHUB


Browser
/hub/
Configurable HTTP Proxy
/user/:name/
Authentificator
Database
Spawner
Hub
/api/auth
Notebook server

JUPYTERHUB

```
# базовый путь и публичный IP адрес для хаба
c.JupyterHub.base_url = '/'
c.JupyterHub.bind_url = 'http://pub-cnt-cls-m1:8765'
# если планируется использование более чем для 1 пользователя
c.JupyterHub.spawner_class = 'jupyterhub.spawner.SimpleLocalProcessSpawner'
c.Spawner.args = ['--allow-root', '--debug', '--profile=PHYS131']
# пользователь в linux- это пользователь в jupyterhub
c.Authenticator.admin_users = {'добавляем админов кластера',}
c.Authenticator.whitelist = {список пользователей Linux, которые будут заходить на jupyterhub}
# так как у нас кластер на внутренней сети, то добавляем параметр прокси
# localhost (127.0.0.1) меняем на внутрению сеть
c.ConfigurableHTTPProxy.api_url='http://10.0.0.2:8108'
c.JupyterHub.proxy_api_ip = '10.0.0.2'
c.JupyterHub.proxy_api_port = 5678
c.JupyterHub.hub_ip = '10.0.0.2'
c.JupyterHub.hub_port = 5678
# переменные серды для spark окружения в jupyterhub
c.YarnSpawner.environment = {
    'PYTHONPATH': 'opt/spark3/python',
    'SPARK_CONF_DIR': '/opt/spark3/conf'
       }
```

JUPYTERHUB KERNEL



JUPYTERHUB KERNEL

/usr/share/jupyter/kernels/

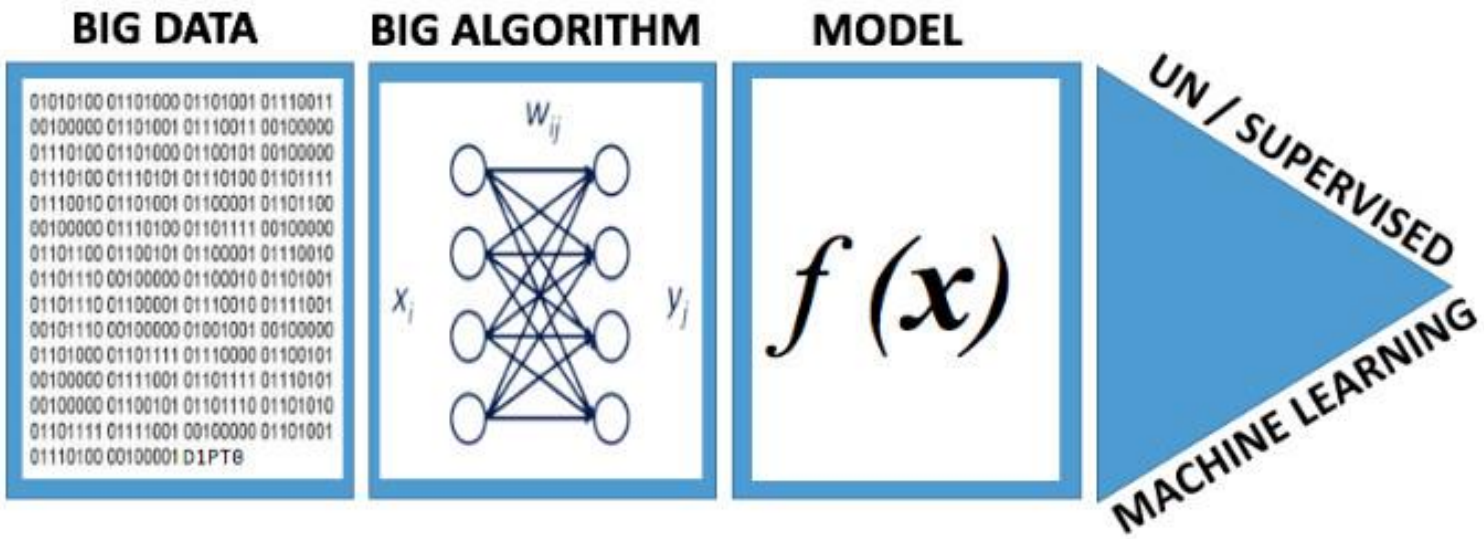
```
{
  "argv": [
    "python3.6",
    "-m",
    "ipykernel_launcher",
    "-f",
    "{connection_file}"
  ],
  "display_name": "Python3.6 + Pyspark(Spark 3.0)",
  "language": "python",
  "env": {
    "PYSPARK_PYTHON": "/usr/bin/python3.6",
    "SPARK_HOME": "/opt/spark3",
    "HADOOP_CONF_DIR": "/etc/spark3/conf/yarn-conf",
    "HADOOP_CLIENT_OPTS": "-Xmx2147483648 -XX:MaxPermSize=512M -Djava.net.preferIPv4Stack=true",
    "PYTHONPATH": "/opt/spark3/python/lib/py4j-0.10.4-src.zip:/opt/spark3/python/",
    "PYTHONSTARTUP": "/opt/spark3/python/pyspark/shell.py",
    "PYSPARK_SUBMIT_ARGS": " --master yarn --deploy-mode client pyspark-shell"
  }
}
```

SPARK | HADOOP CLUSTER

…. RETURN TO

BIG DATA?

ИЗ БОЛЬШОГО В МАЛОЕ



ИЗ БОЛЬШОГО В МАЛОЕ

Агрегат: data, user, goods_id

Детализация заказа в магазине?

Портфель акций одного инвестора?

Все заказы по всем магазинам?

Все транзакции по всем акциям?

Детализация заказа в магазине?

Портфель акций одного инвестора?

Все заказы по всем магазинам?

Все транзакции по всем акциям?

Детализация заказа в магазине?

Портфель акций одного инвестора?

Все заказы по всем магазинам?

Все транзакции по всем акциям?

Агрегат:
shop_id, cust_id,

Агрегат:
date, stock_id,

DATA БРОСАЕТ ВЫЗОВ

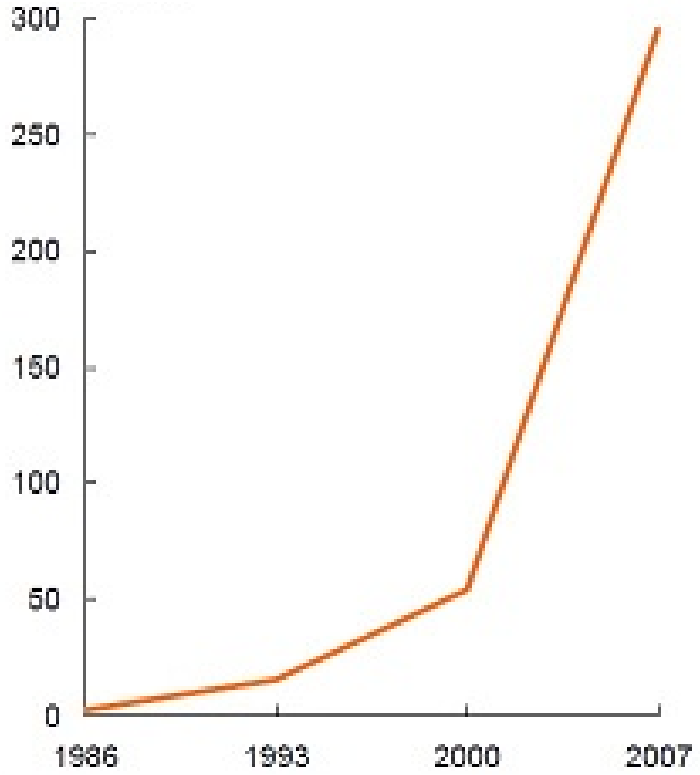
Данные создаются
очень быстро

Данные из разных
источников и в
разных форматах

DATA БРОСАЕТ ВЫЗОВ

Данные создаются очень быстро

Данные из разных источников и в разных форматах



DATA БРОСАЕТ ВЫЗОВ

Данные создаются очень быстро

Данные из разных источников и в разных форматах

3V

VOLUME

VARIETY

VELOCITY

3V

VOLUME

VARIETY

VELOCITY

$

3V

VOLUME

VARIETY

VELOCITY

3V

VOLUME

VARIETY

VELOCITY


```
[mpm_winnt:notice] [pid 5776:tid 740] AH00456: Apache Lounge VC15 Server built:
[core:notice] [pid 5776:tid 740] AH00094: Command line: 'C:\\Server\\bin\\Apach
[mpm_winnt:notice] [pid 5776:tid 740] AH00418: Parent: Created child process 87
[mpm_winnt:notice] [pid 8752:tid 712] AH00354: Child: Starting 64 worker thread
[mpm_winnt:notice] [pid 5776:tid 740] AH00422: Parent: Received shutdown signal
[mpm_winnt:notice] [pid 8752:tid 712] AH00364: Child: All worker threads have e
[mpm_winnt:notice] [pid 5776:tid 740] AH00430: Parent: Child process 8752 exite
[mpm_winnt:notice] [pid 3584:tid 740] AH00455: Apache/2.4.39 (Win64) PHP/7.3.2
[mpm_winnt:notice] [pid 3584:tid 740] AH00456: Apache Lounge VC15 Server built:
[core:notice] [pid 3584:tid 740] AH00094: Command line: 'C:\\Server\\bin\\Apach
[mpm_winnt:notice] [pid 3584:tid 740] AH00418: Parent: Created child process 11
[mpm_winnt:notice] [pid 1140:tid 716] AH00354: Child: Starting 64 worker thread
ing.The 'Apache2.4' service has restarted.winnt:notice] [pid 3584:tid 740] AH00
[ssl:warn] [pid 3584:tid 740] AH01873: Init: Session Cache is not configured [h
```

3V

VOLUME

VARIETY

VELOCITY

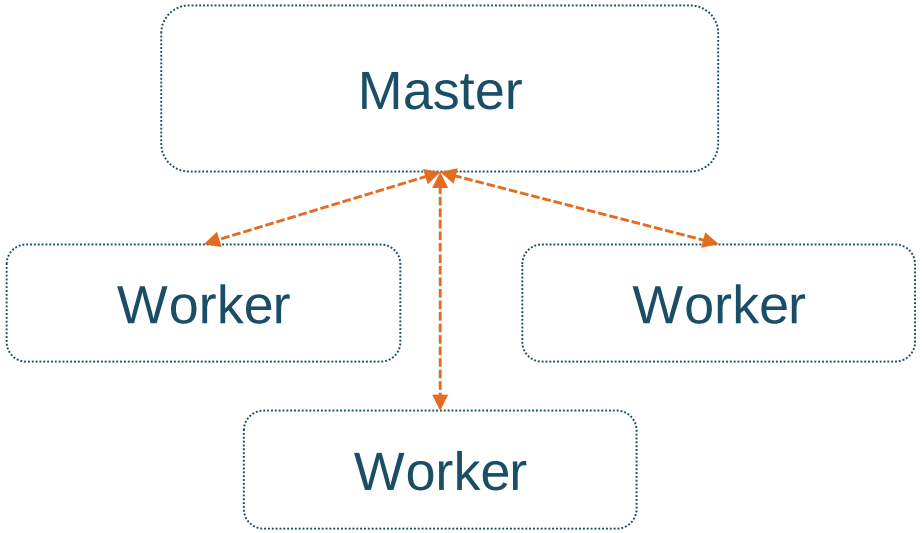
КЛАСТЕР

ПОЯВИЛСЯ КЛАСТЕР

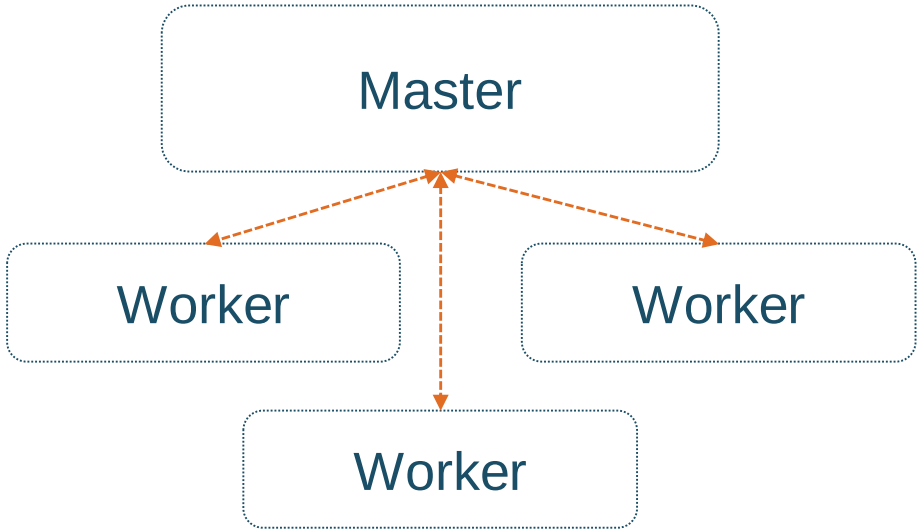
Один сервер !

ПОЯВИЛСЯ КЛАСТЕР

Один сервер !

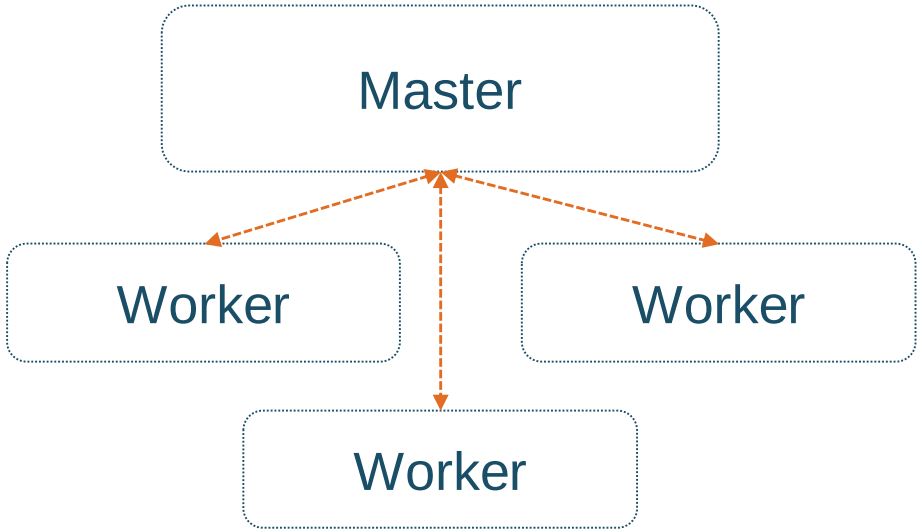


ПОЯВИЛСЯ КЛАСТЕР и добавил проблем



Проблемы координации

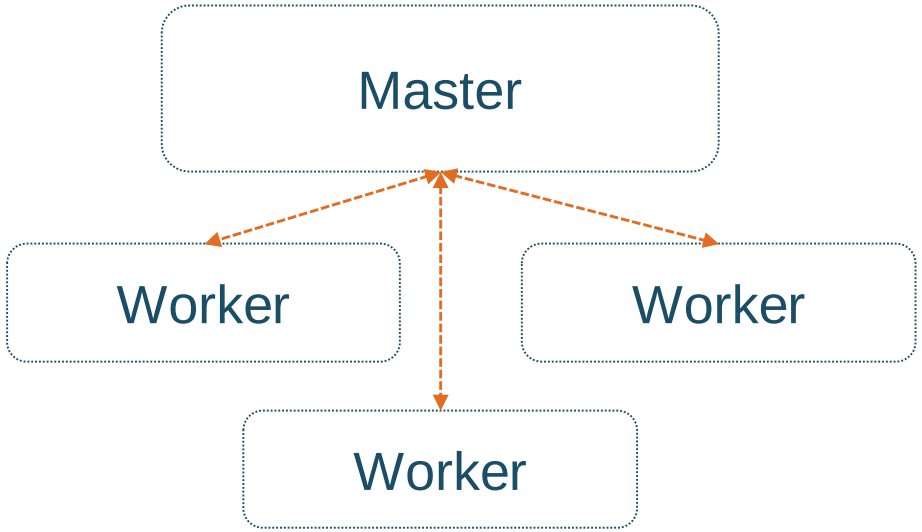
ПОЯВИЛСЯ КЛАСТЕР и добавил проблем



Проблемы координации

Проблемы коммуникации

ПОЯВИЛСЯ КЛАСТЕР и добавил проблем

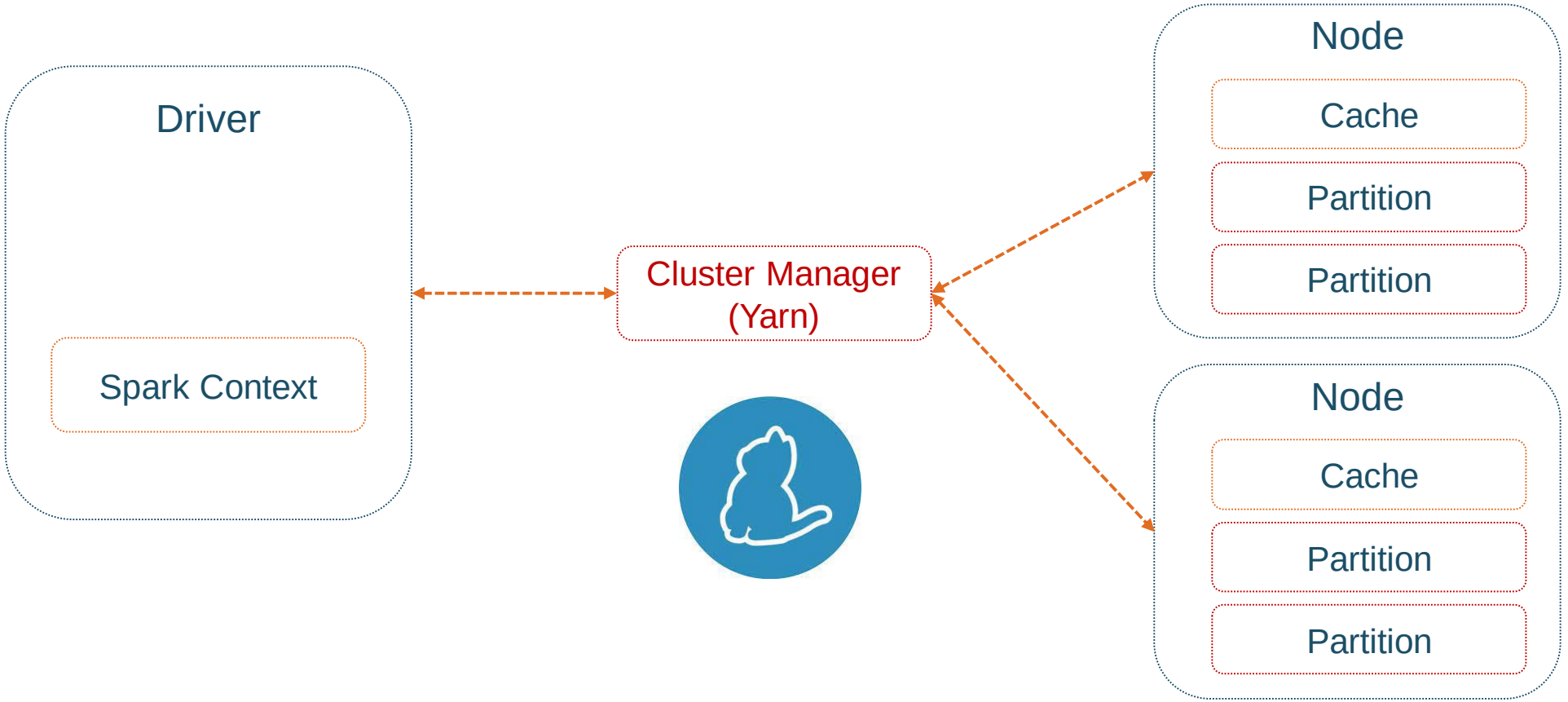


Проблемы координации

Проблемы коммуникации

Проблемы стабильности

НА КЛАСТЕРЕ ДРУГОЕ УРАВЛЕНИЕ

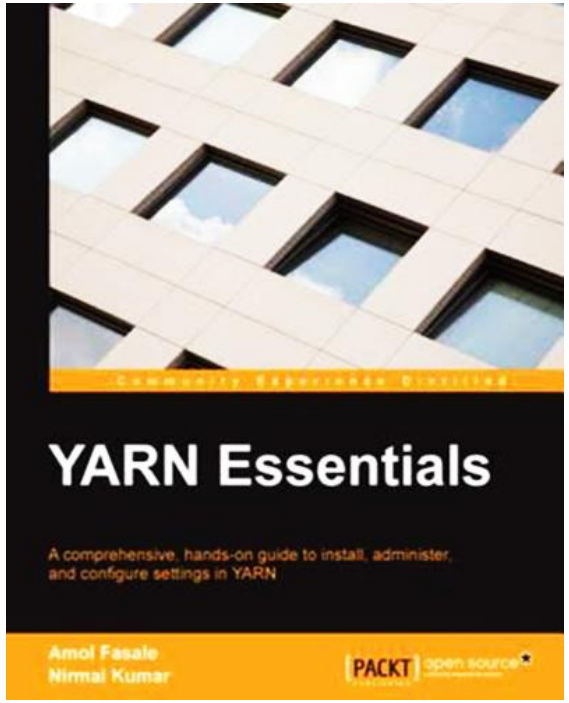


НА КЛАСТЕРЕ ДРУГОЕ УРАВЛЕНИЕ

- yarn app –list
- yarn app -status
- yarn app –appStates
- yarn app –destroy appId
- yarn app –kill appId

https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html

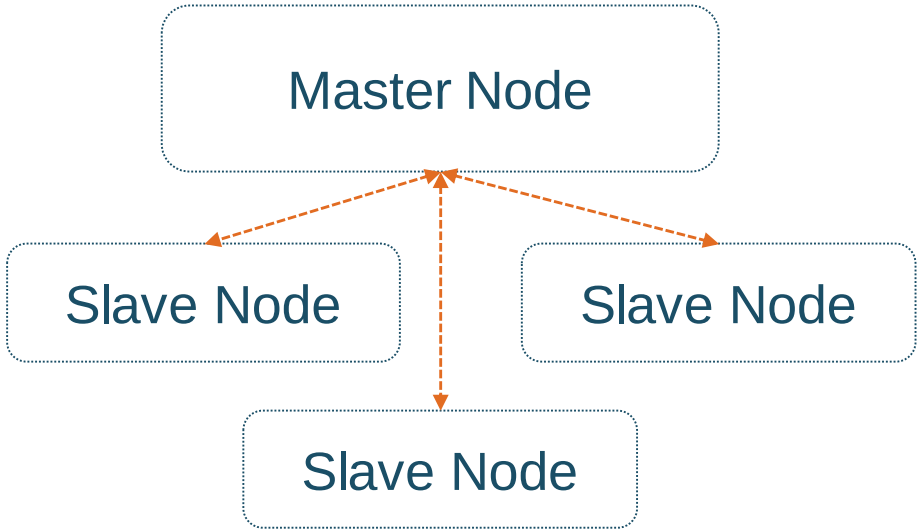
НА КЛАСТЕРЕ ДРУГОЕ УРАВЛЕНИЕ



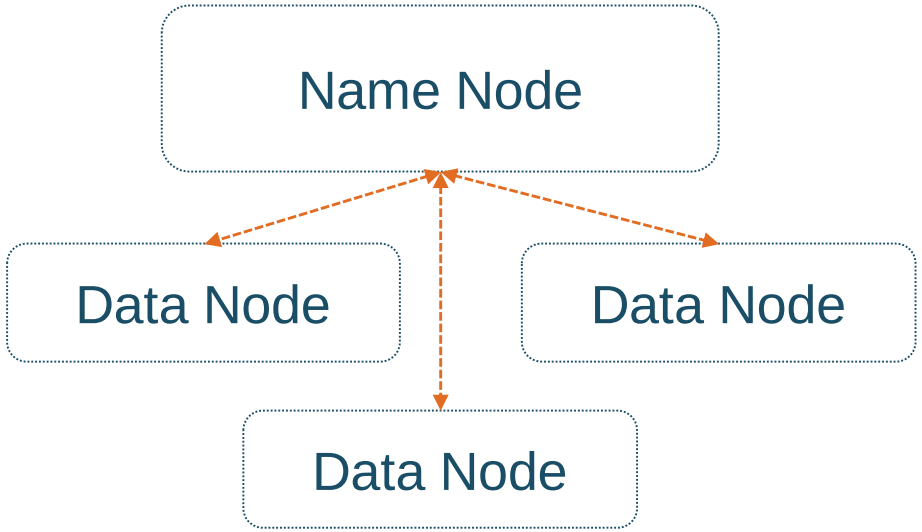
https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html

SPARK | HADOOP CLUSTER

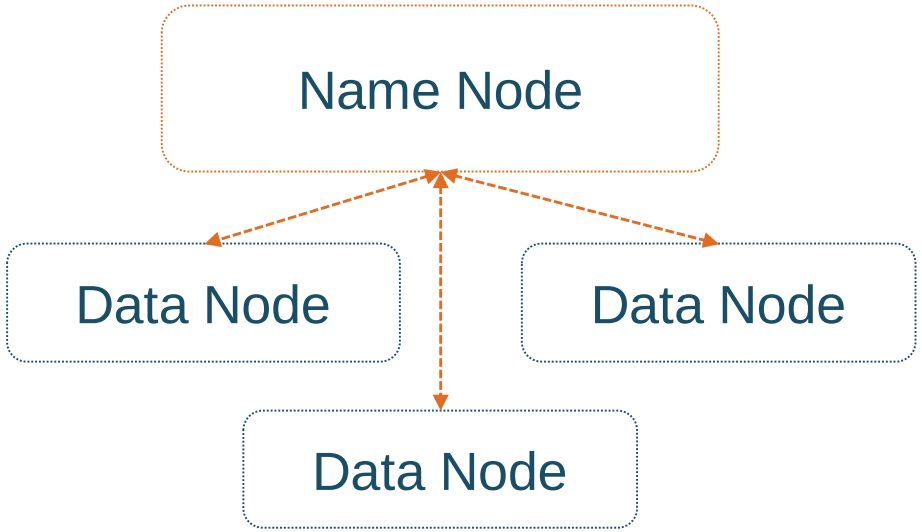
HDFS



HDFS



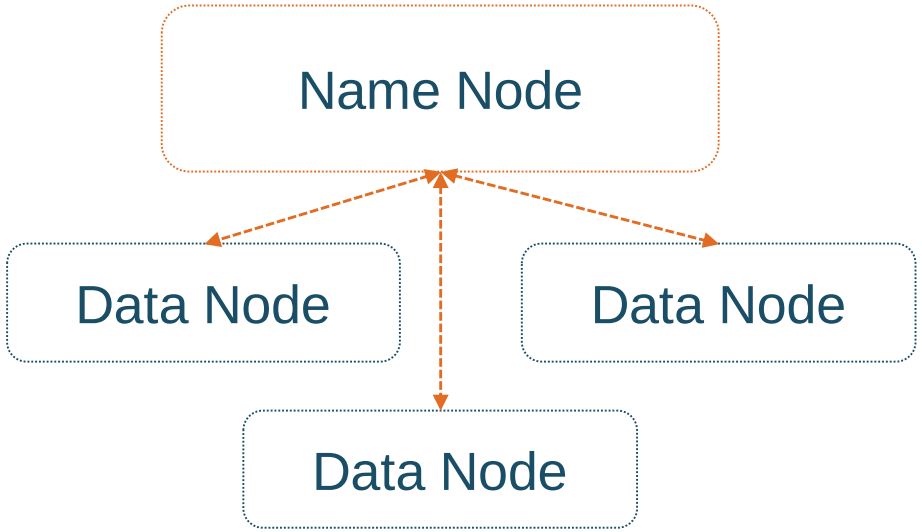
HDFS



Name Node

- Предоставляет и контролирует доступ
- Координирует задачи
- Содержит пространство имен и управляет: (open, close, rename)

HDFS

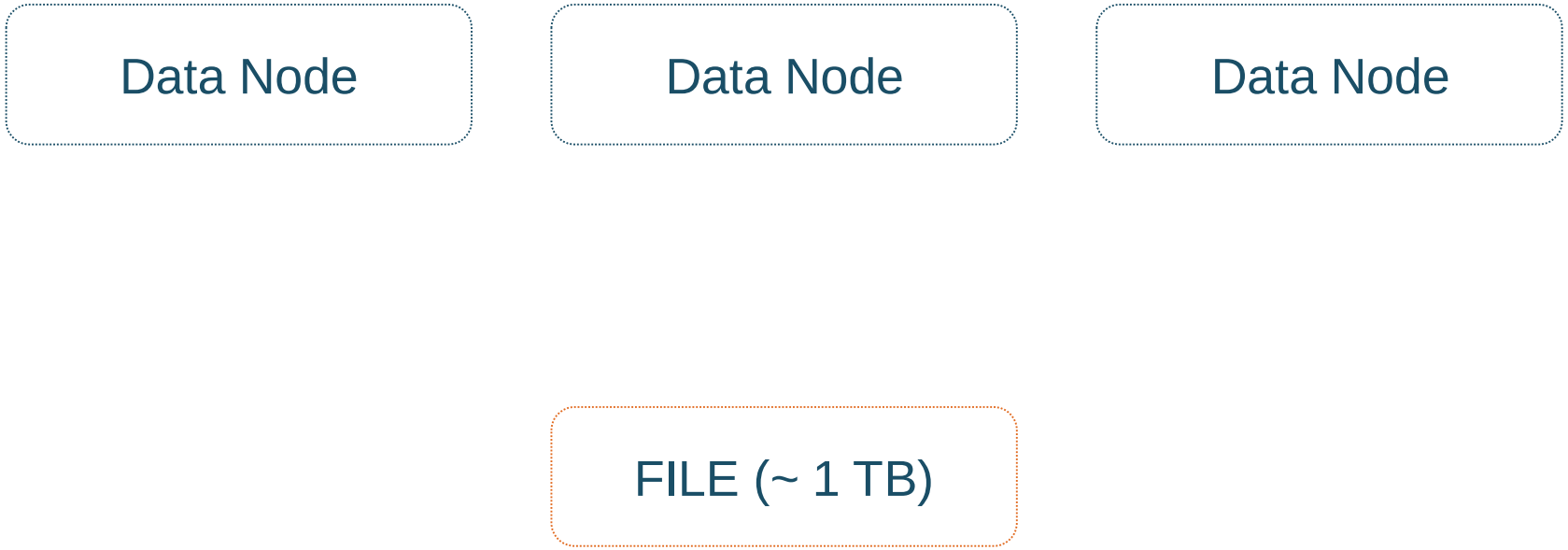


Data Node

- Хранят и обрабатываю данные

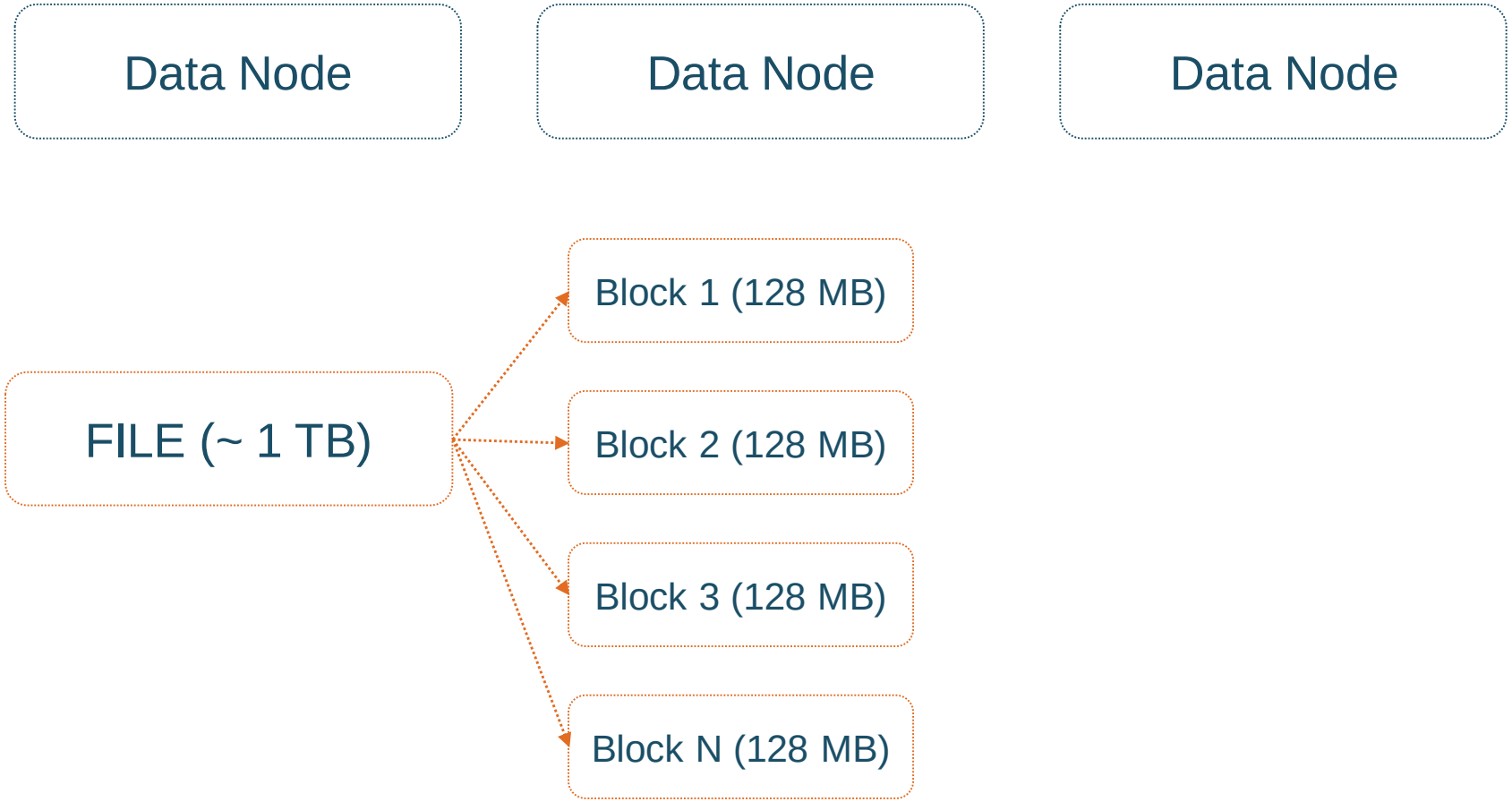
HDFS

ОСОБЕННОСТЬ ХРАНЕНИЯ



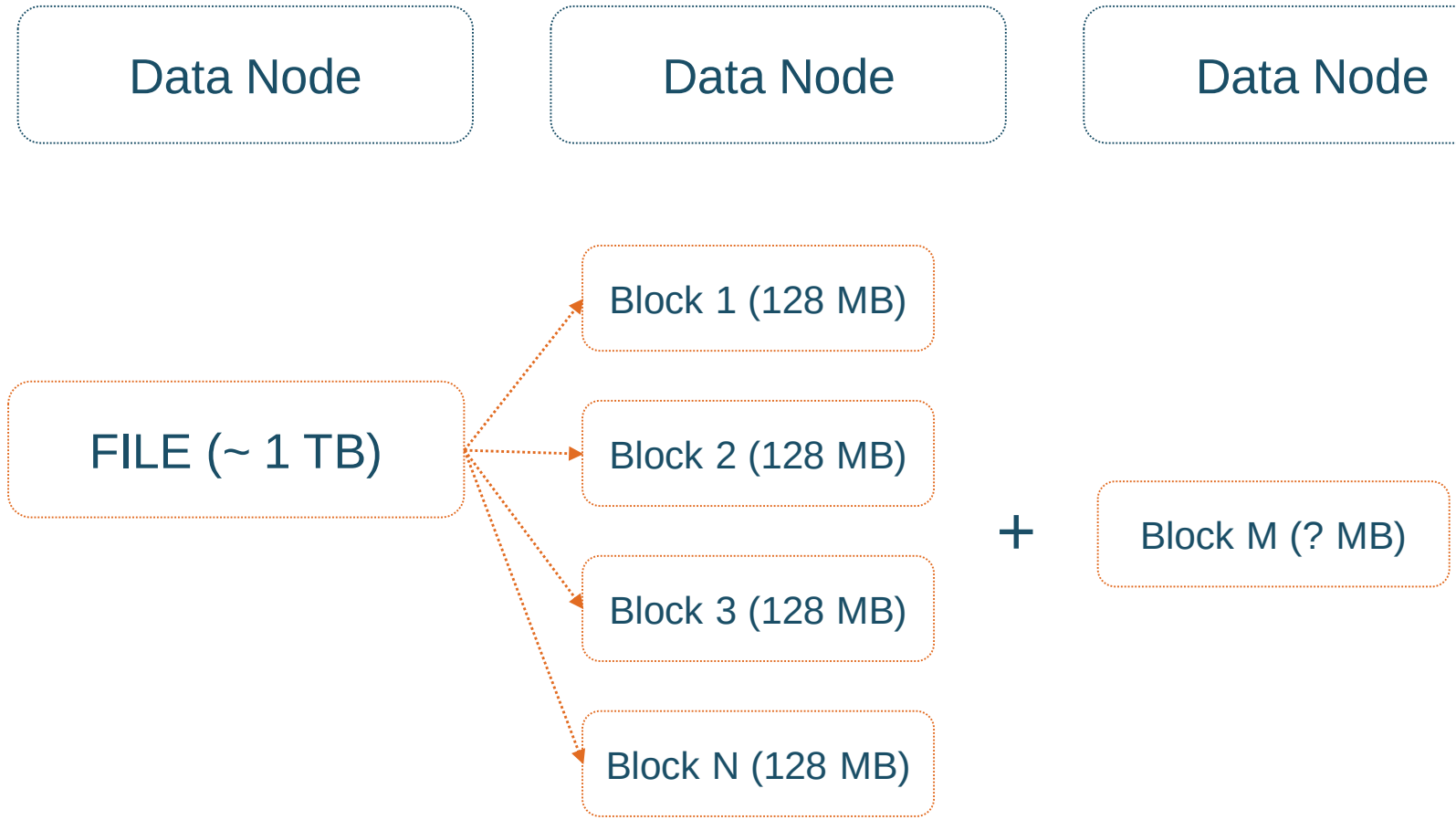
HDFS

ОСОБЕННОСТЬ ХРАНЕНИЯ



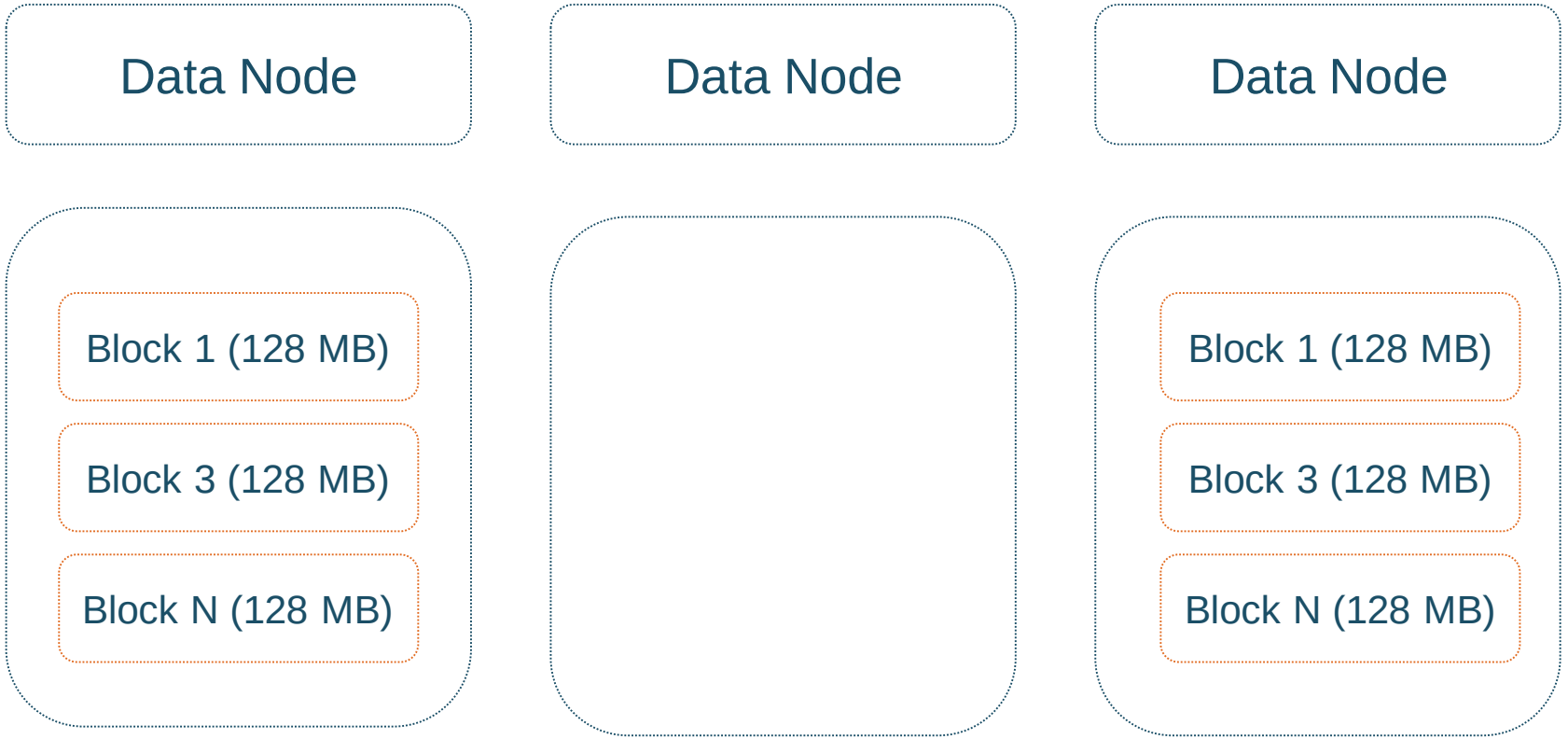
HDFS

ОСОБЕННОСТЬ ХРАНЕНИЯ



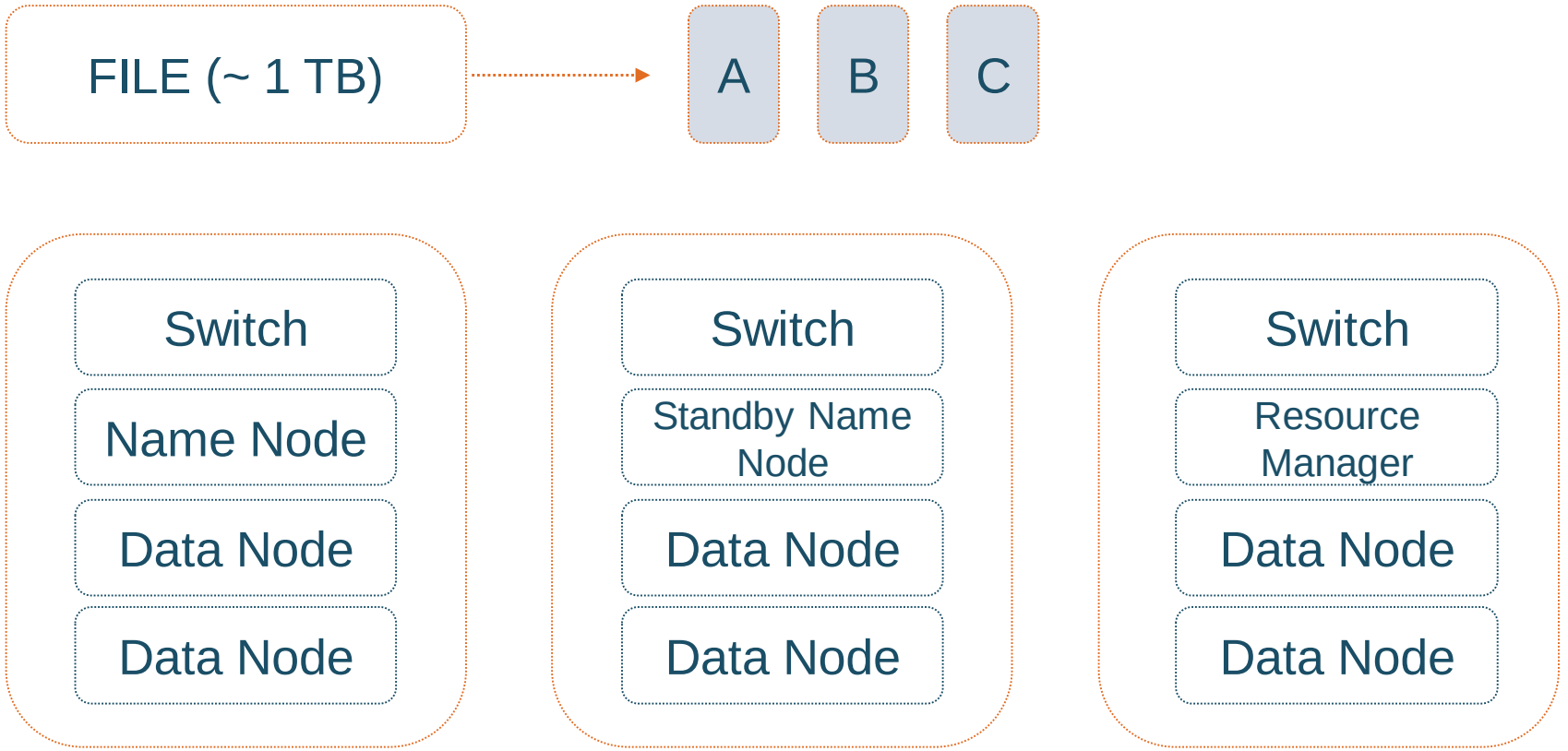
HDFS

ОСОБЕННОСТЬ ХРАНЕНИЯ



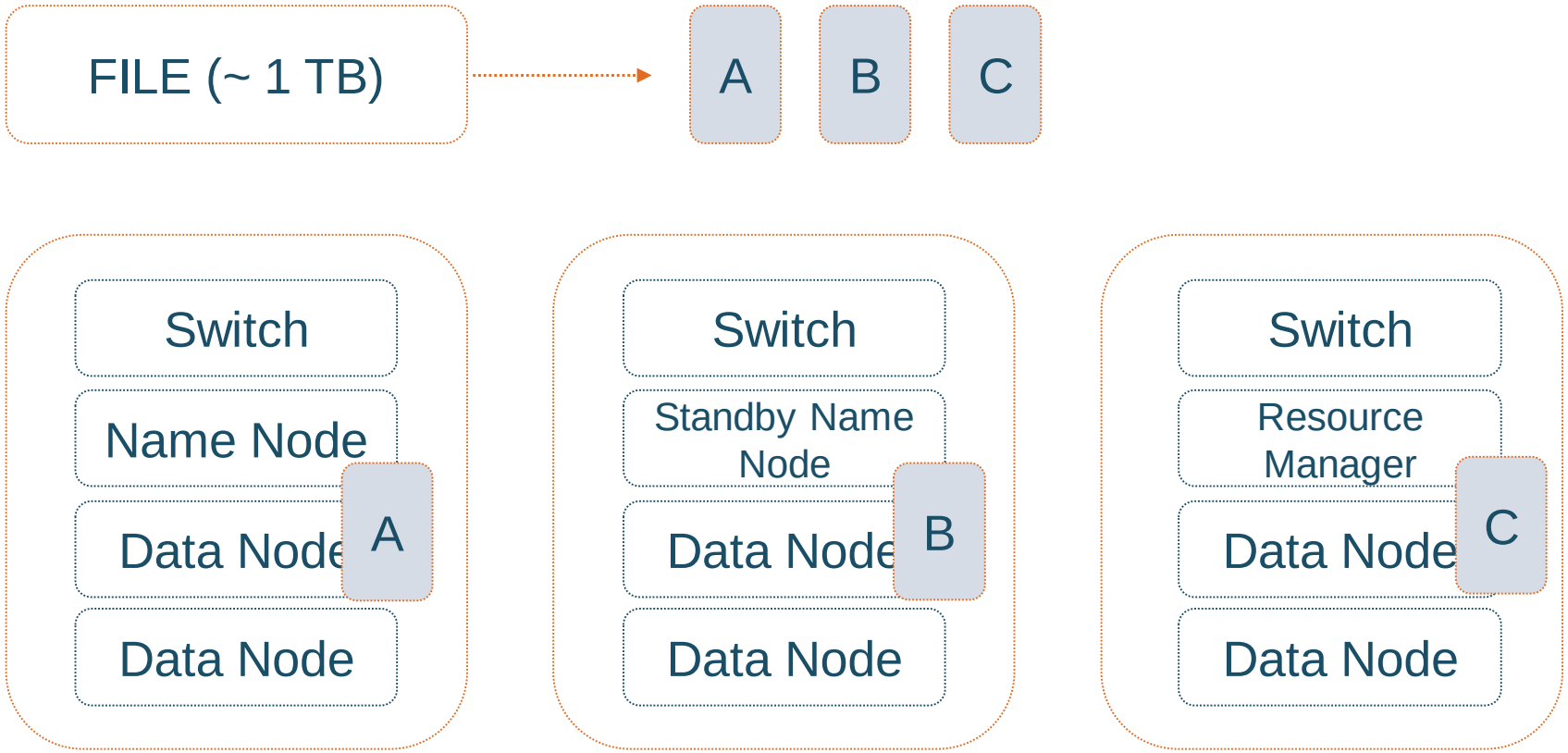
HDFS – RACK AWARENESS

ОСОБЕННОСТЬ ХРАНЕНИЯ



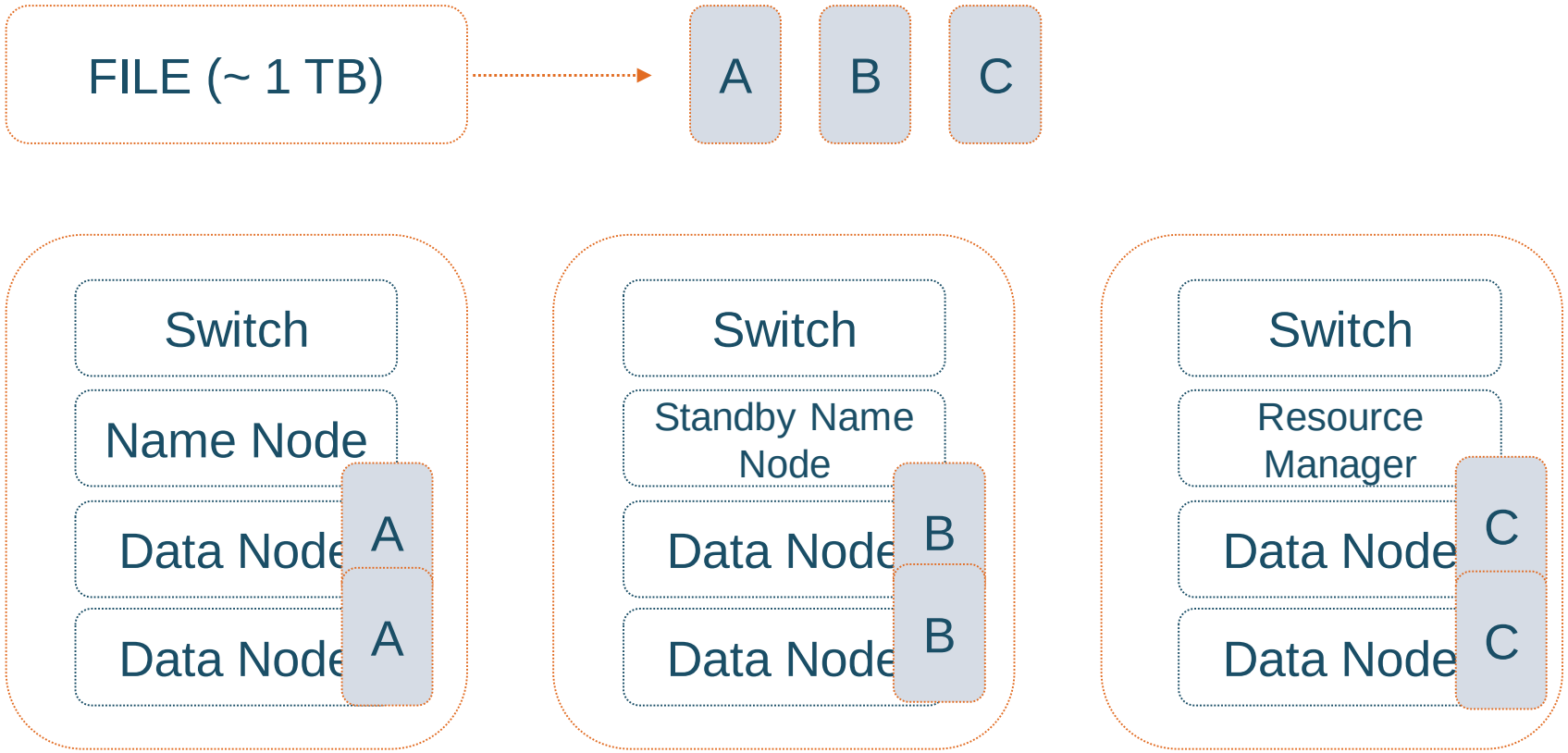
HDFS – RACK AWARENESS

Rep.1
–
на разных узлах



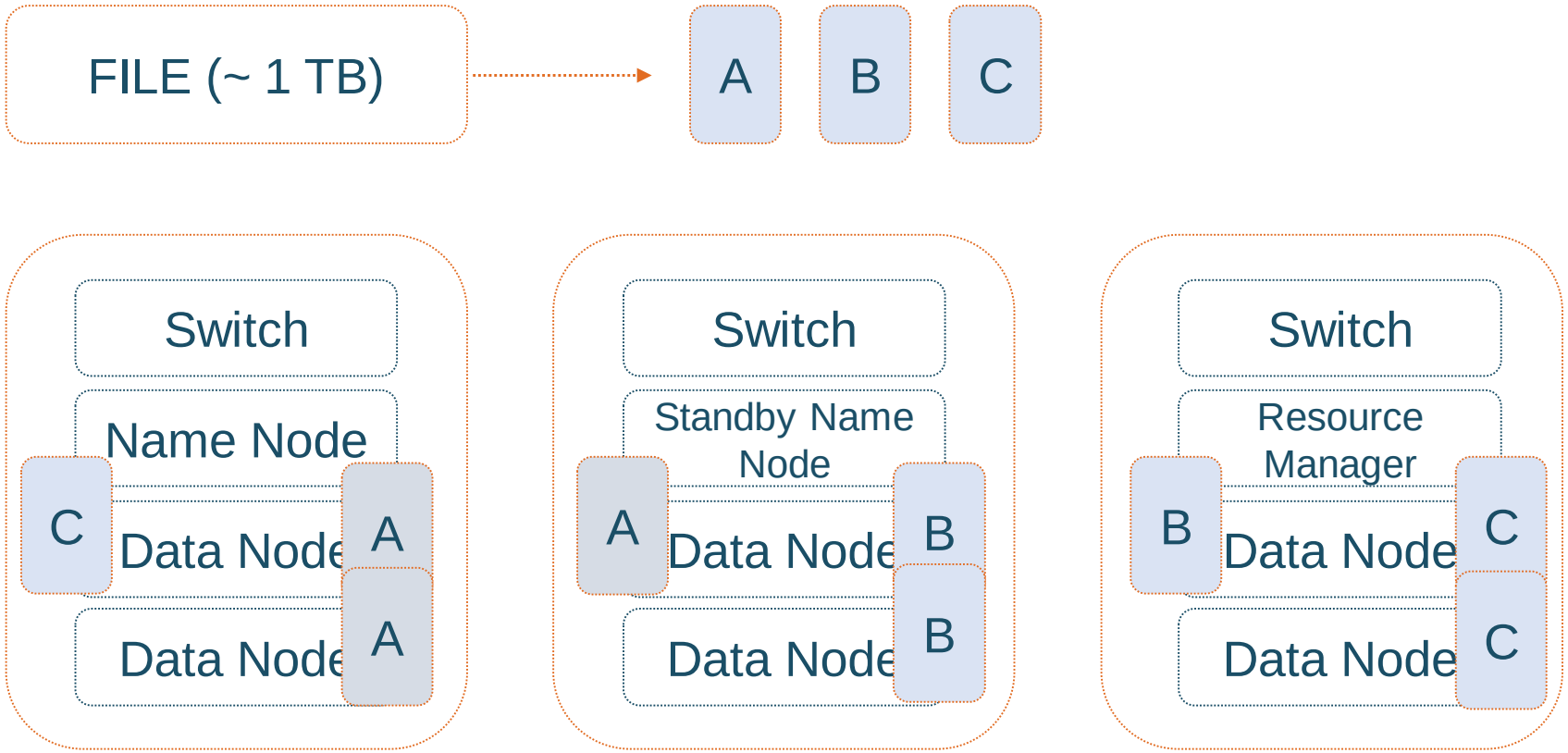
HDFS – RACK AWARENESS

Rep.2
–
на разных нодах внутри



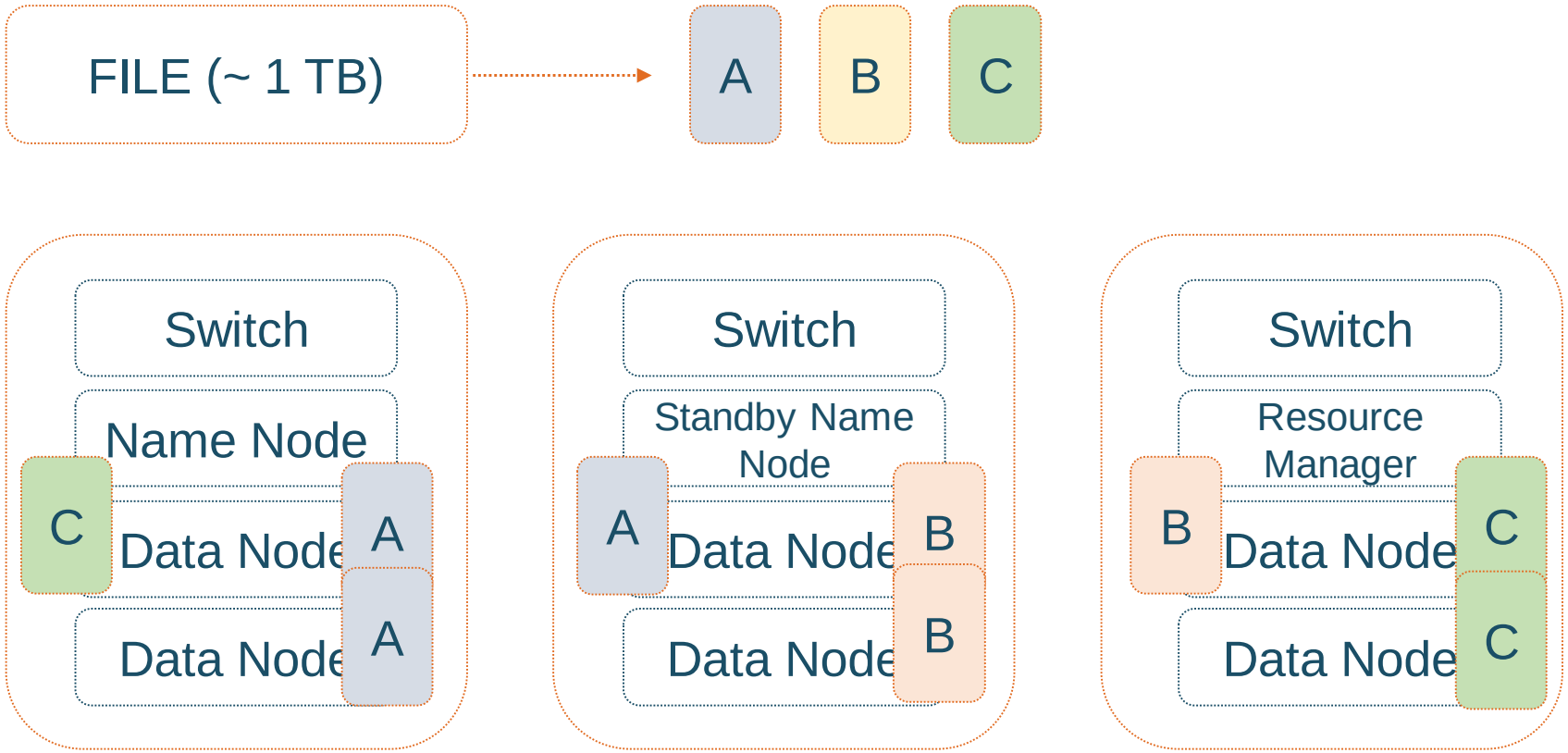
HDFS – RACK AWARENESS

Rep.3
–
на разных стойках



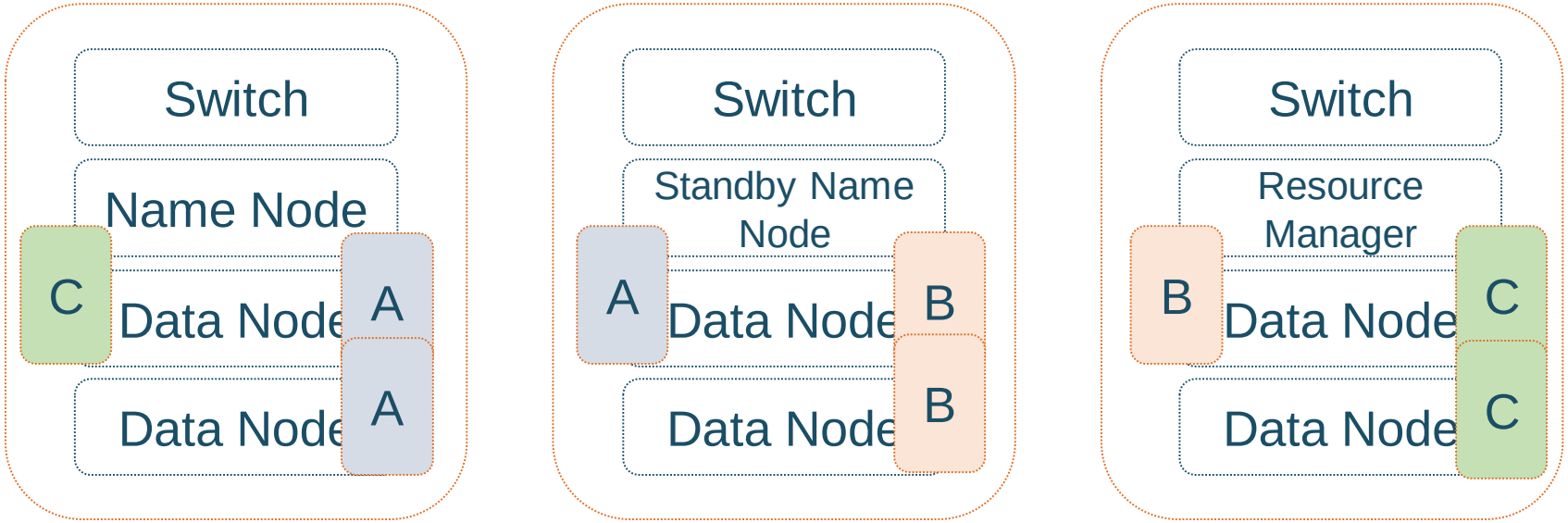
HDFS – RACK AWARENESS

Rep.3
–
на разных стойках



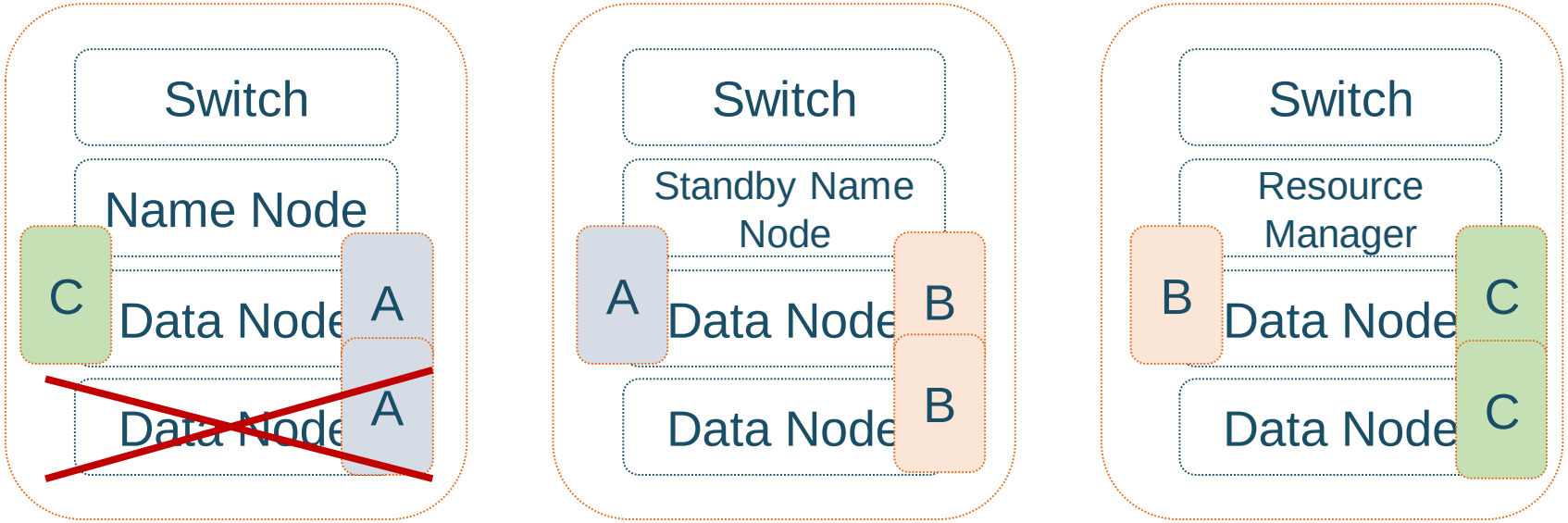
HDFS – FAULT TOLERANCE

Данные доступны всегда



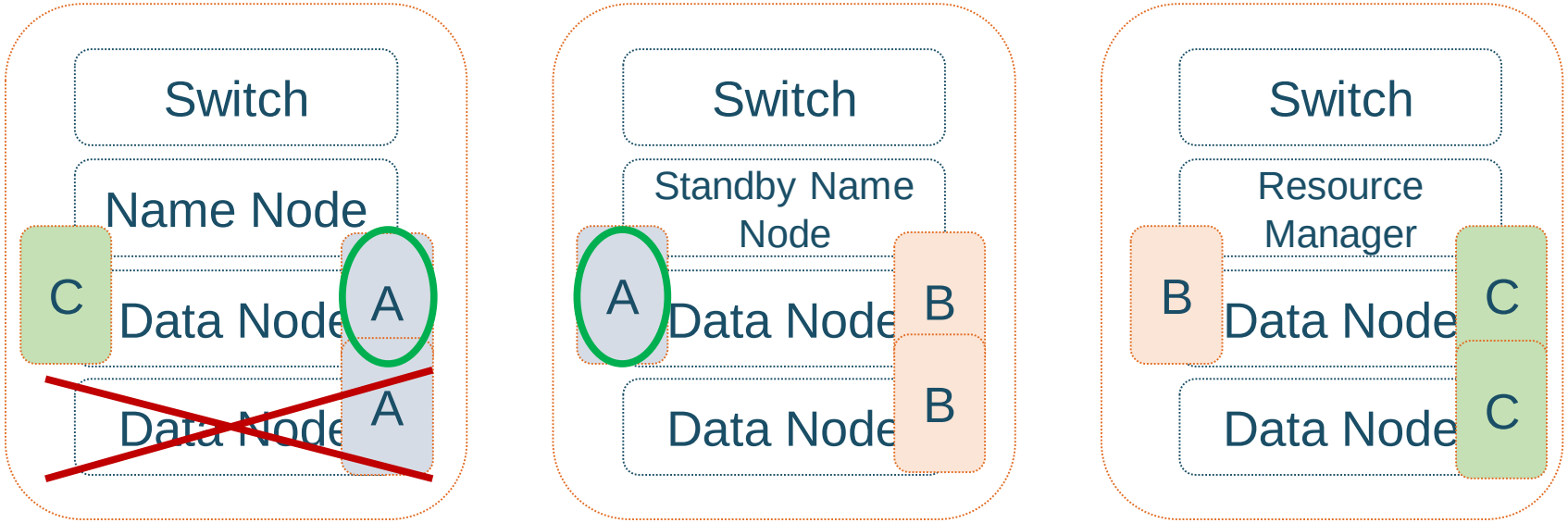
HDFS – FAULT TOLERANCE

Данные доступны всегда



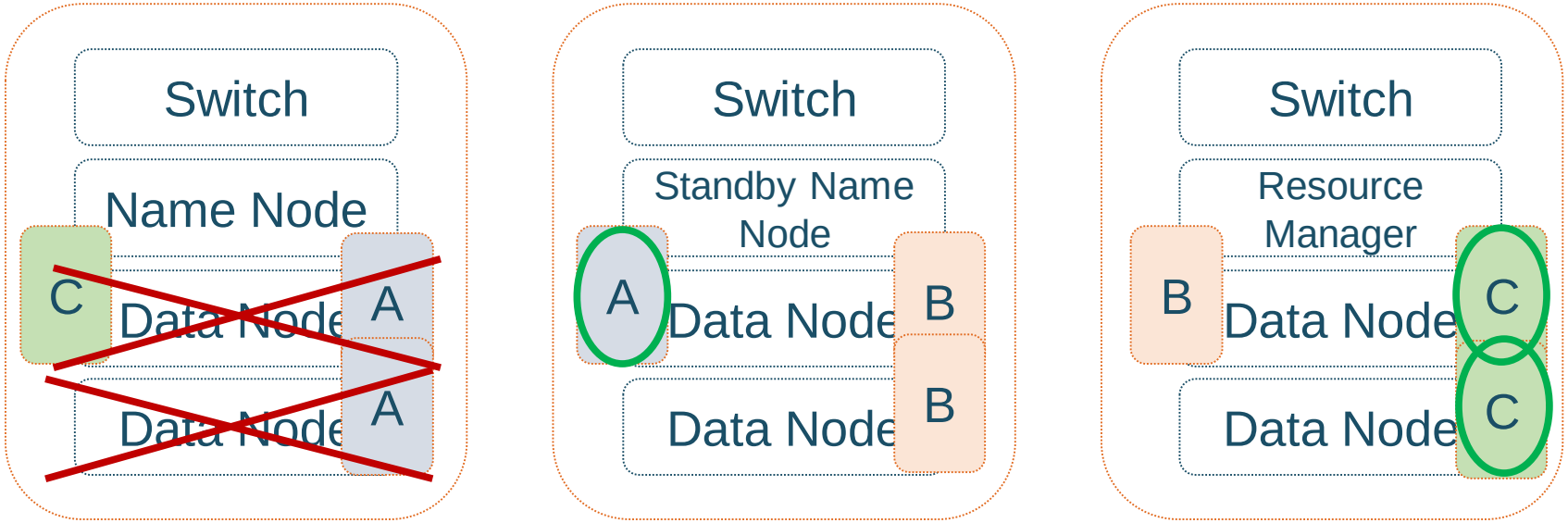
HDFS – FAULT TOLERANCE

Данные доступны всегда



HDFS – FAULT TOLERANCE

Данные доступны всегда



HDFS | БАЛАНСИРОВКА ДИСКОВ

Расчет баланса
и утилизации диска

	Disk1	Disk2	Disk3	Disk4		Total	Ideal
объем	256	512	1024	2048		3840	0,45
использовано	100	176	950	520		1746	
% использования	0,39	0,34	0,93	0,25			
% плотности	0,06	0,11	-0,48	0,2			

- Total объем = сумма объемов всех дисков == sum(Disk (1->N))
- Total использ. = сумма использования всех дисков == sum(Disk (1->N)
- Ideal = Total использ. / Total объем

HDFS | БАЛАНСИРОВКА ДИСКОВ

Расчет баланса
и утилизации диска

	Disk1	Disk2	Disk3	Disk4		Total	Ideal
объем	256	512	1024	2048		3840	0,45
использовано	100	176	950	520		1746	
% использования	0,39	0,34	0,93	0,25			
% плотности	0,06	0,11	-0,48	0,2			

- % плотности = Ideal - % использования

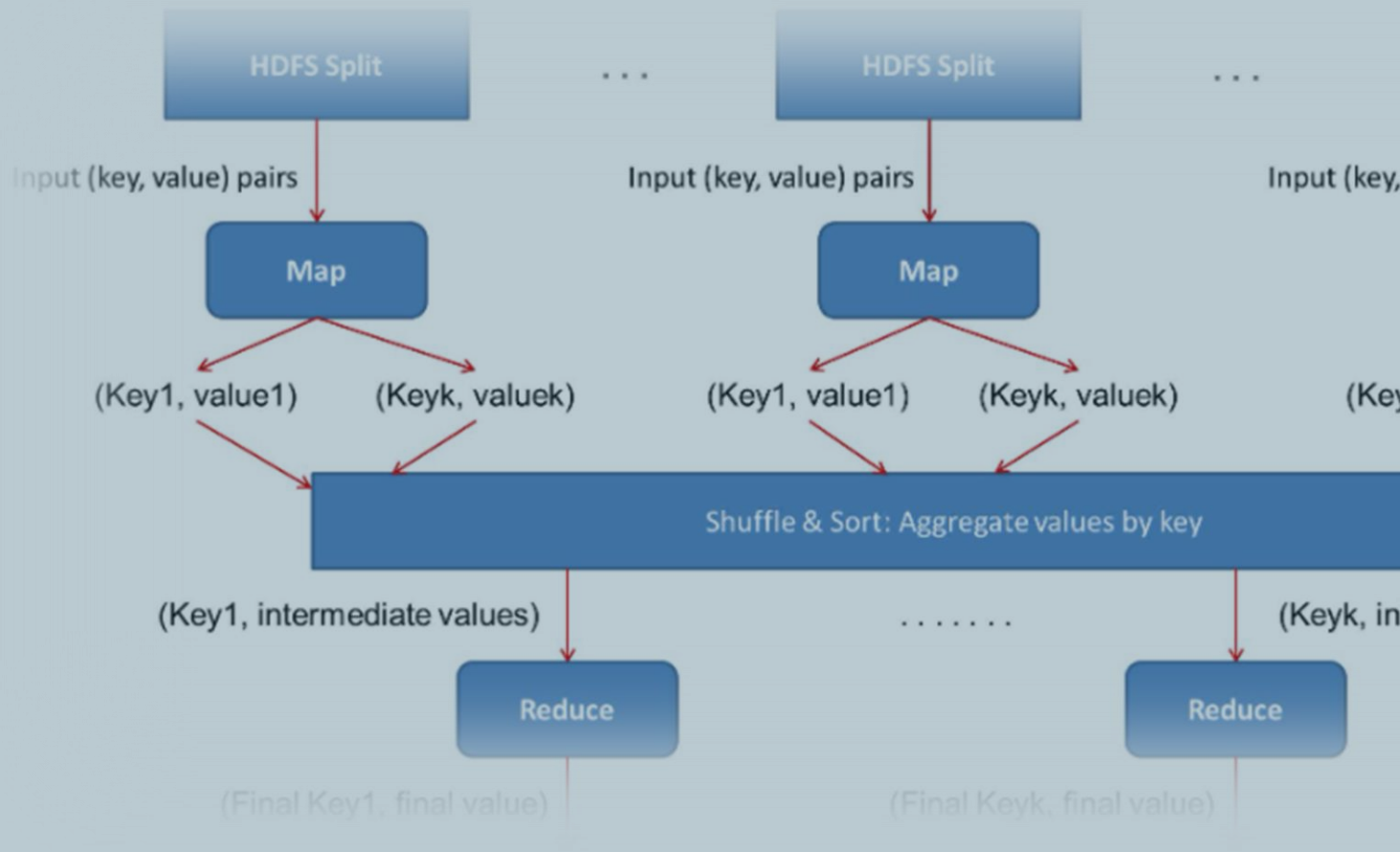
HDFS | БАЛАНСИРОВКА ДИСКОВ

Нормализуем

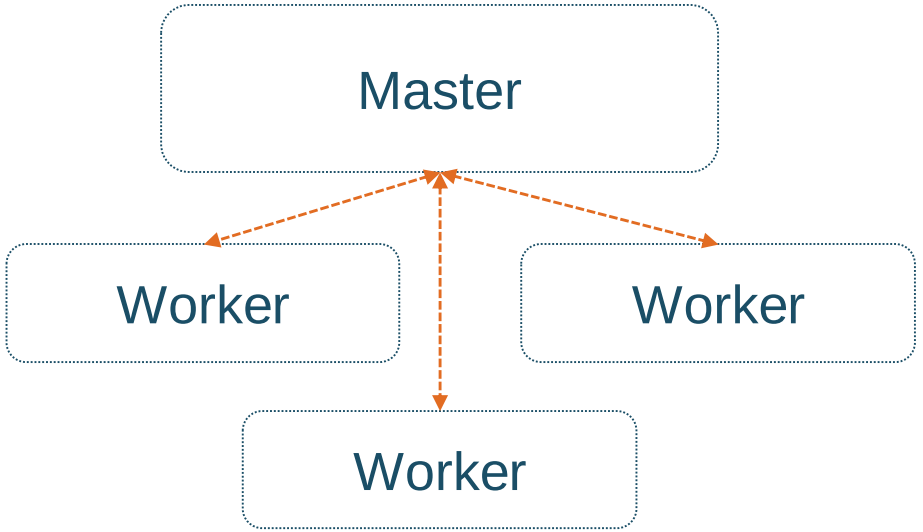
- hdfs diskbalancer – plan <datanode>
- настраиваем ednabled, out, thresholdPercentage, maxerror
- проверяем отчеты:

  hdfs diskbalancer –fs namenode.url –report file_path//

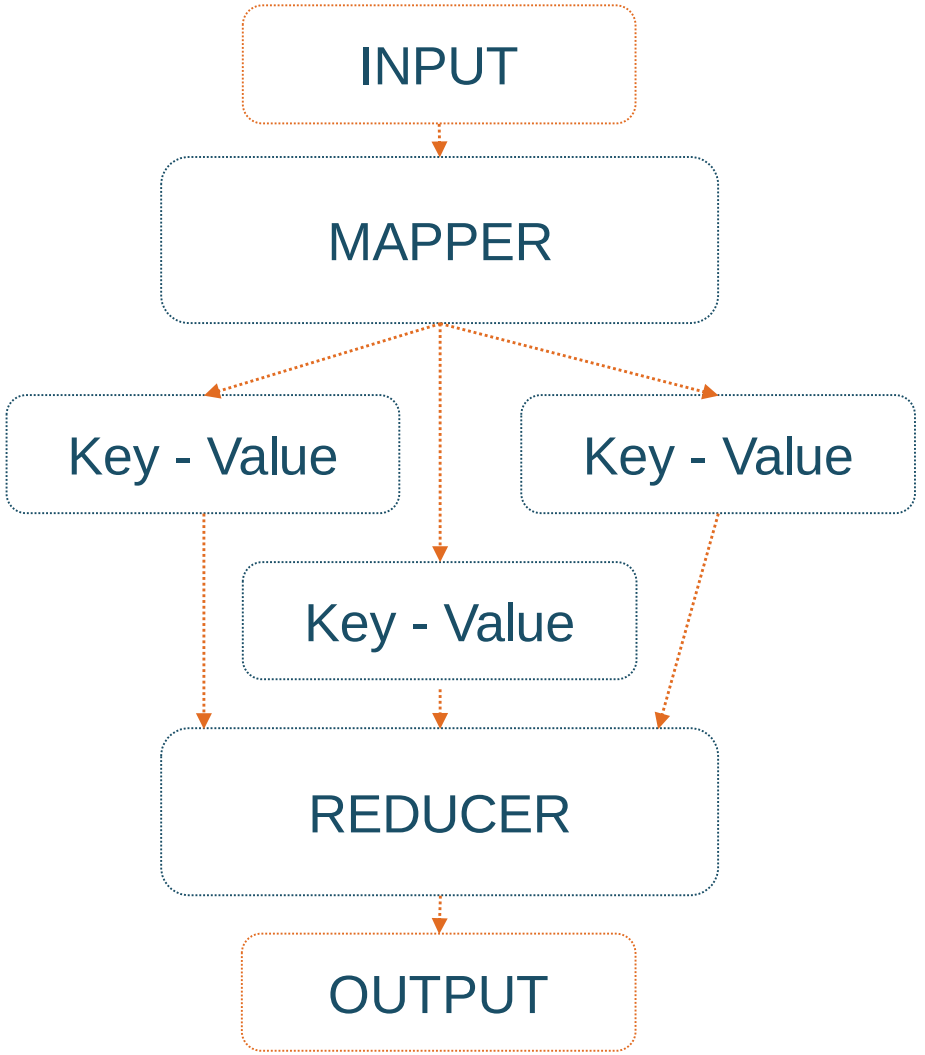
MAP - REDUCE


HDFS Split
HDFS Split
Input (key, value) pairs
Input (key, value) pairs
Map
Map
(Key1, value1)
(Keyk, valuek)
(Key1, value1)
(Keyk, valuek)
Shuffle & Sort: Aggregate values by key
(Key1, intermediate values)
Reduce
Reduce
(Final Key1, final value)
(Final Keyk, final value)

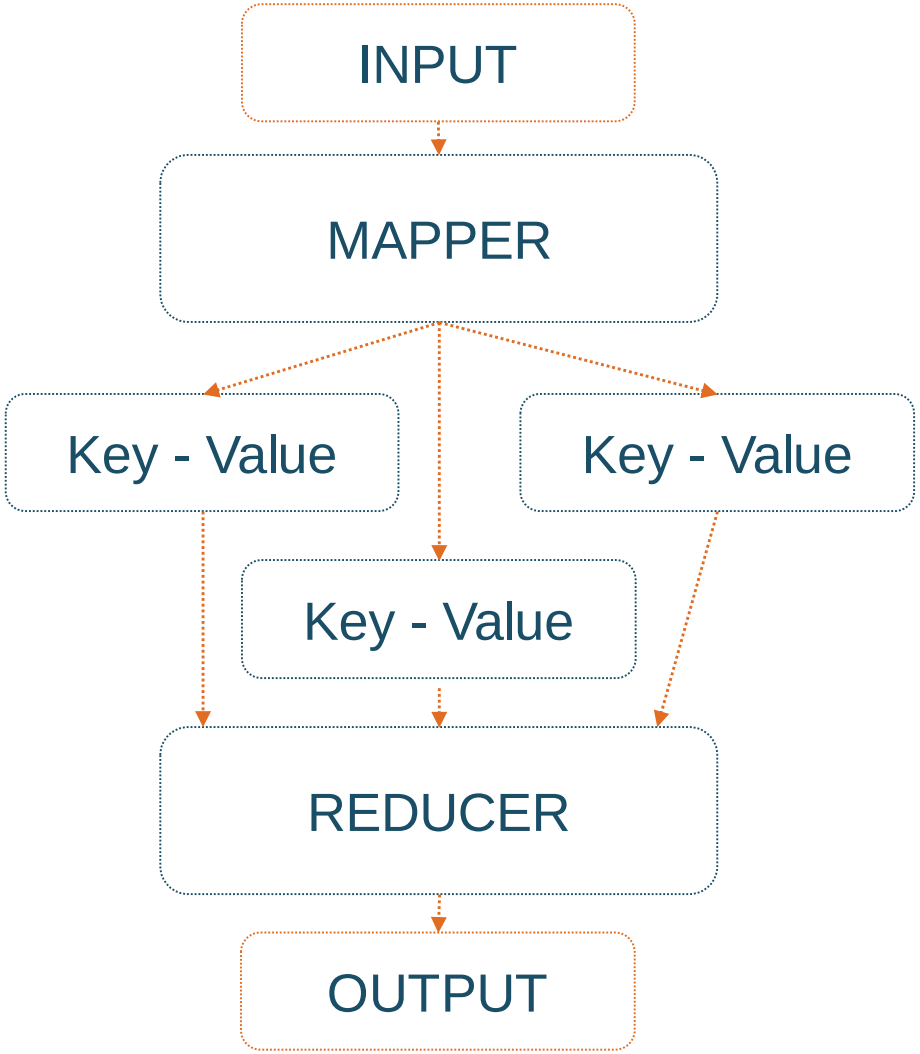
MAP – REDUCE



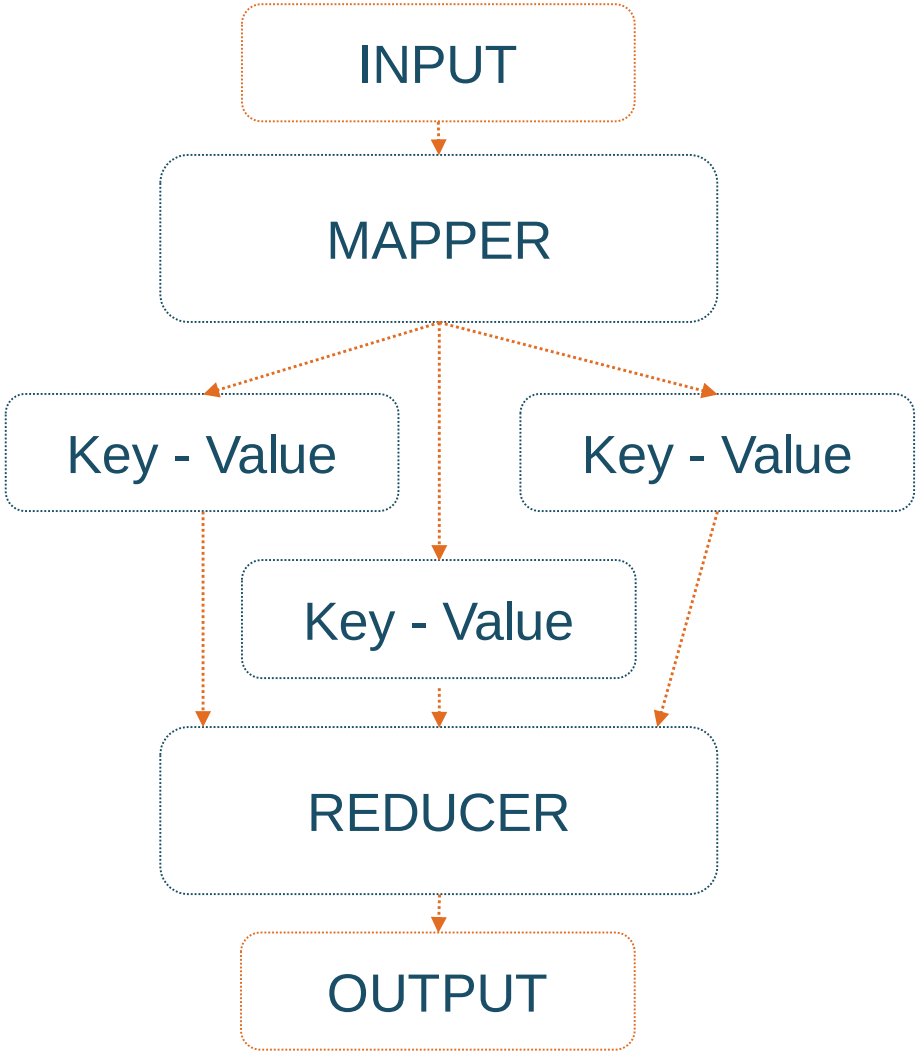
MAP – REDUCE



MAP – REDUCE



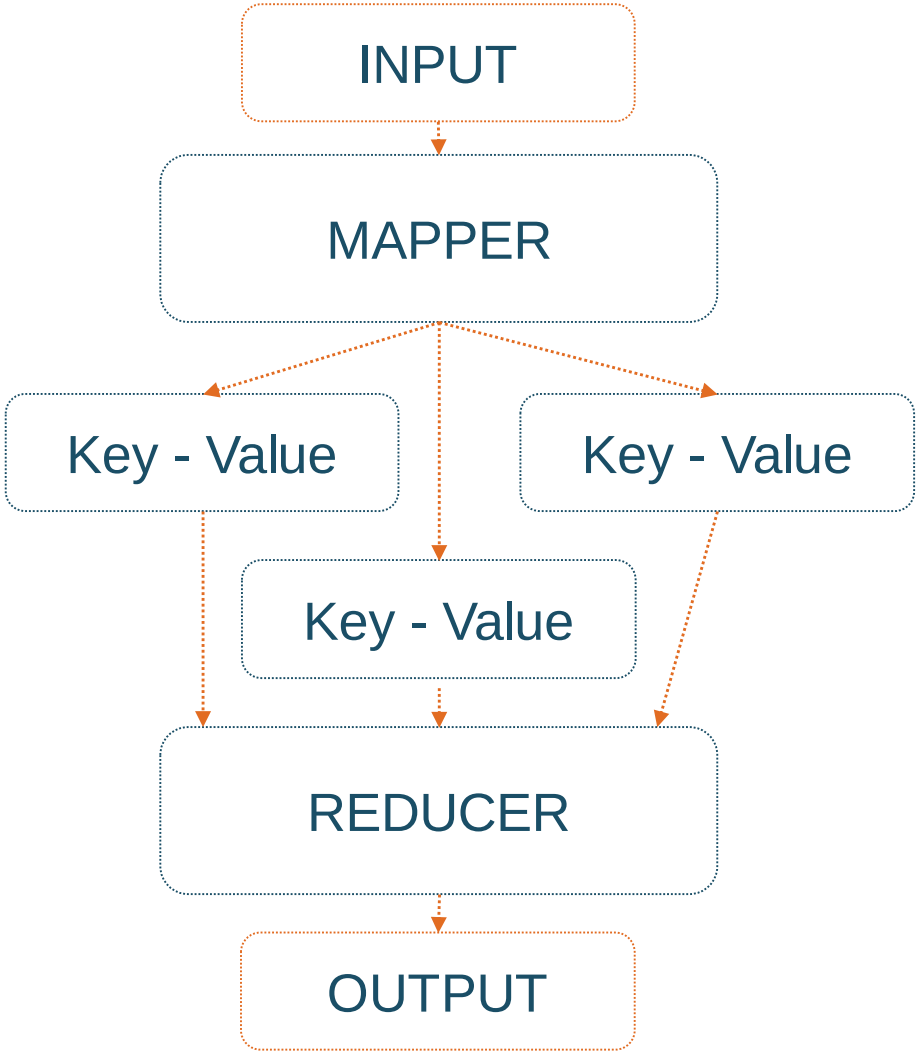
MAP – REDUCE



INPUT

обычный файл\tc вашими\tданными

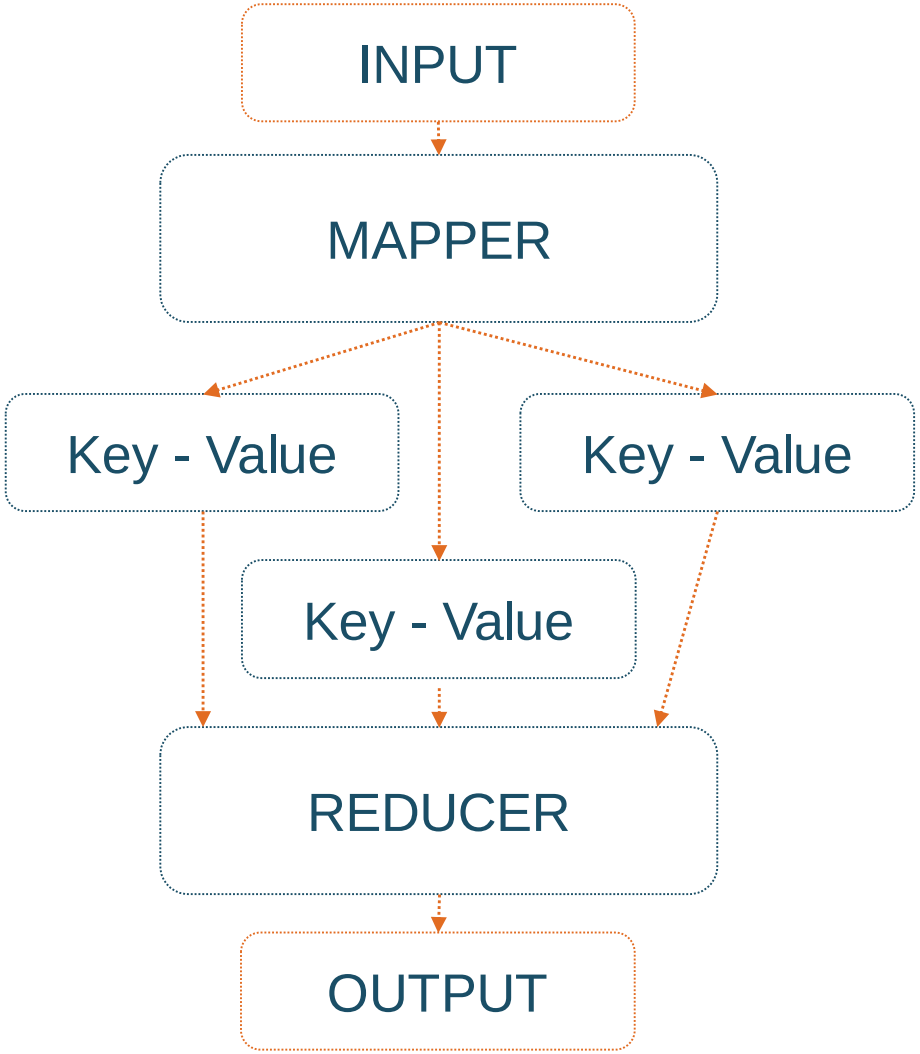
MAP – REDUCE



INPUT

обычный файл\tс вашими\tданными

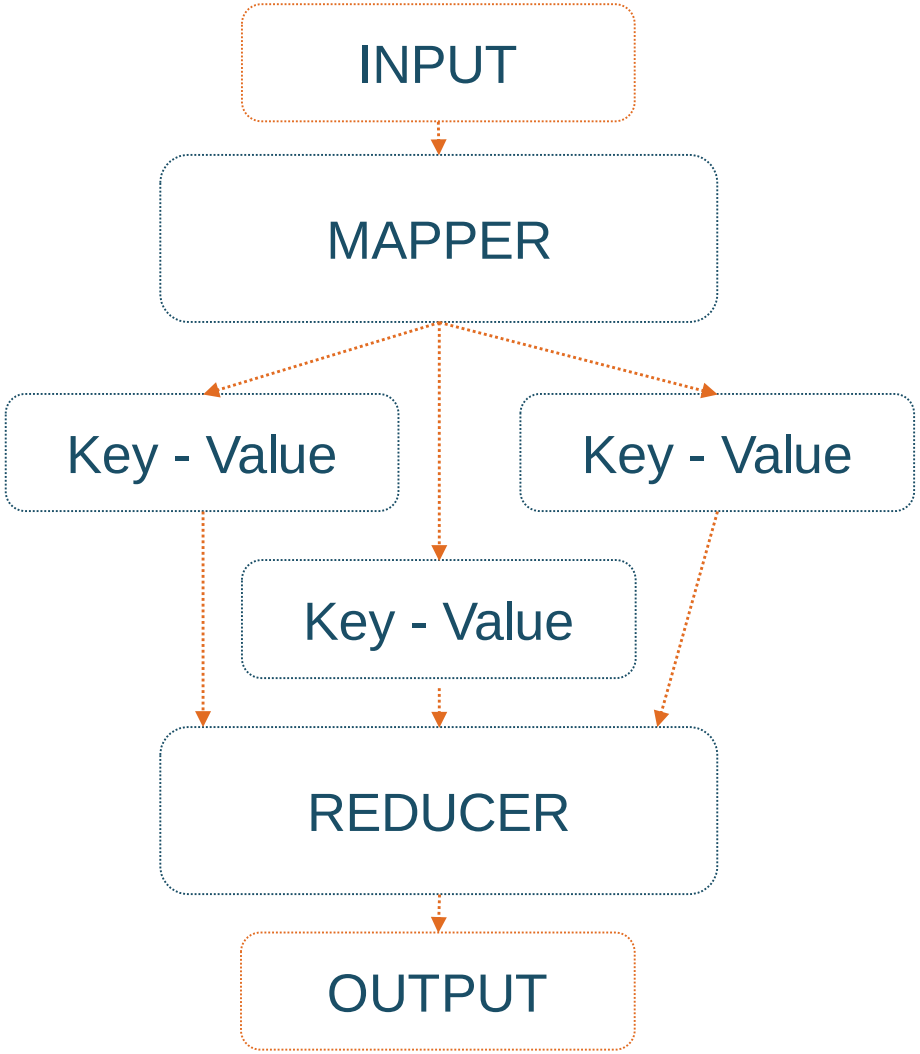
MAP – REDUCE



MAPPER

- str(обычный файл\tс вашими\tданными)
- list(list(str(обычный файл),
  str(с вашими),
  str(данными)
  ))
- function(object) <- list(str)
- return: key – value

MAP – REDUCE



MAPPER

- опция процесса:

  mapred.max.split.size

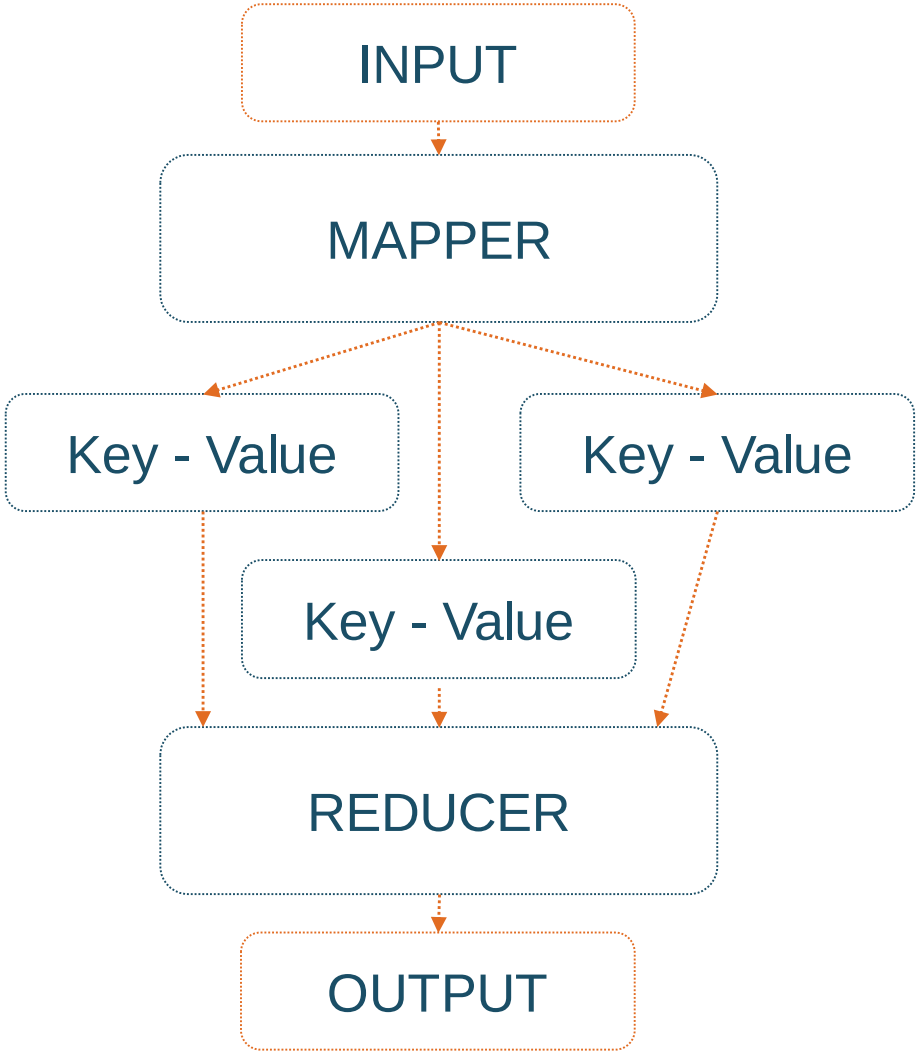
- формула расчета мапперов:

  общий размер данных / mapred.max.split.size

  Пример: 1TB данных, 100MB split size:

  (1000*1000) / 100 = 10000 мапперов

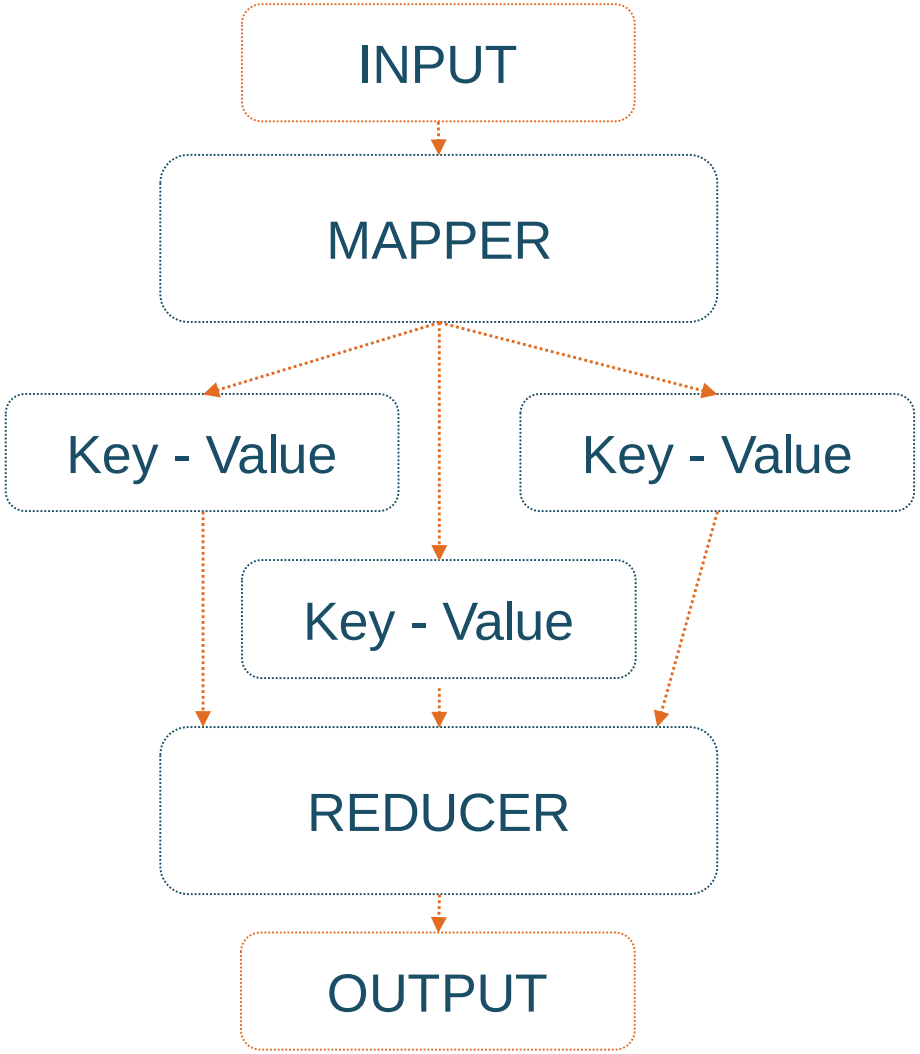
MAP – REDUCE



REDUCER

- list(key - value)
- function(object) <- key - value
- return: result

MAP – REDUCE



REDUCER

- опция процесса:

  job.setNumreduceTasks(int)

- формула расчета редьюсеров:

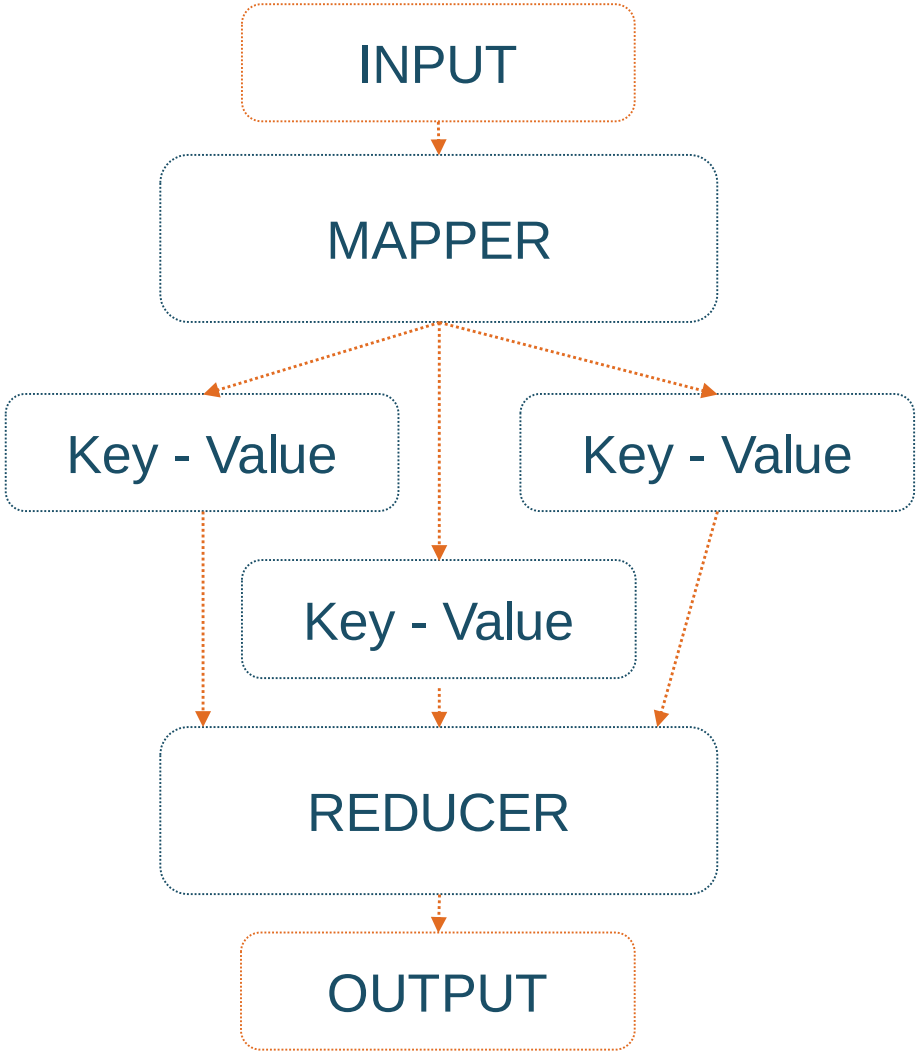
  const * (кол-во нод * макс. контейнеров на ноде)

  Пример: const = 0.95 или 1.75 всегд, 3 ноды,

  8 контейнеров на ноде

  math.ceil(0.95 * (3 * 8))

MAP – REDUCE



REDUCER

- опция процесса:

  job.setNumreduceTasks(int)

  если int = 0, то reducers не выполнятся
- формула расчета редьюсеров:

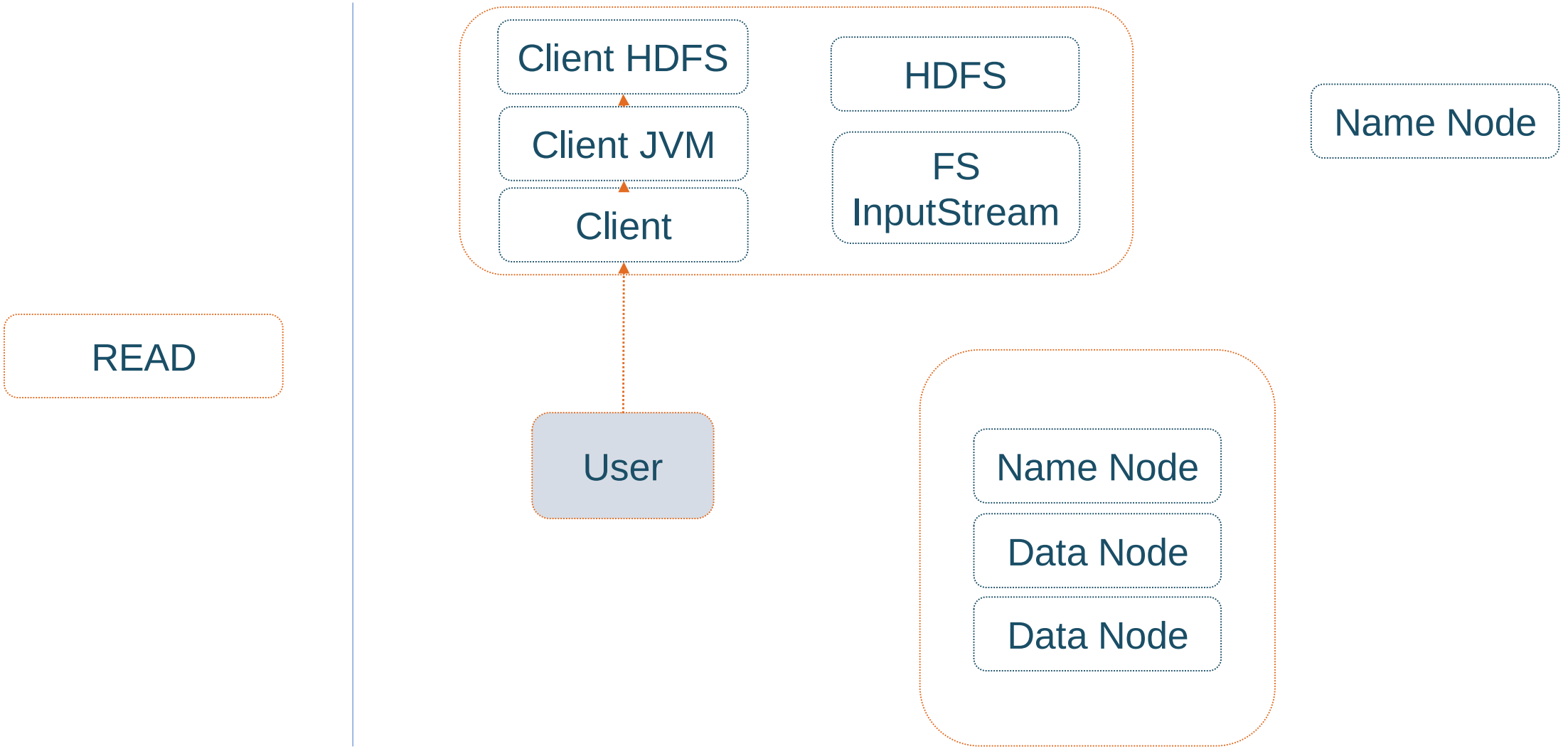
  const * (кол-во нод * макс. контейнеров на ноде)

  Пример: const = 0.95 или 1.75 всегд, 3 ноды,

  8 контейнеров на ноде

  math.ceil(0.95 * (3 * 8))

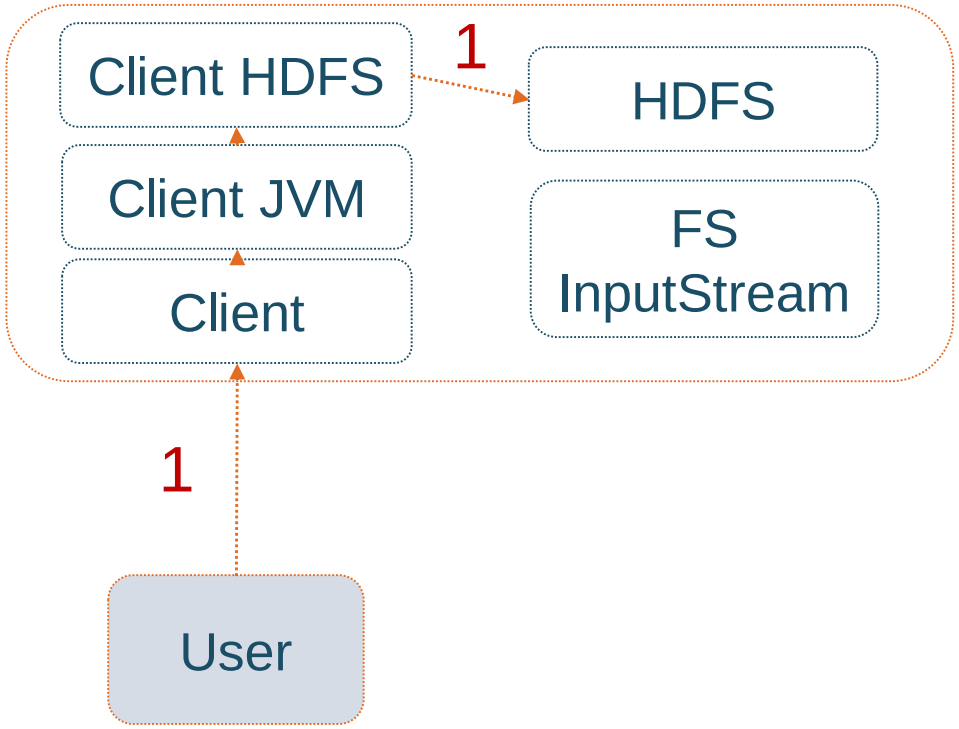
MAP – REDUCE | READ HDFS



MAP – REDUCE | READ HDFS

READ

1
Обращение клиента.
Открытие HDFS



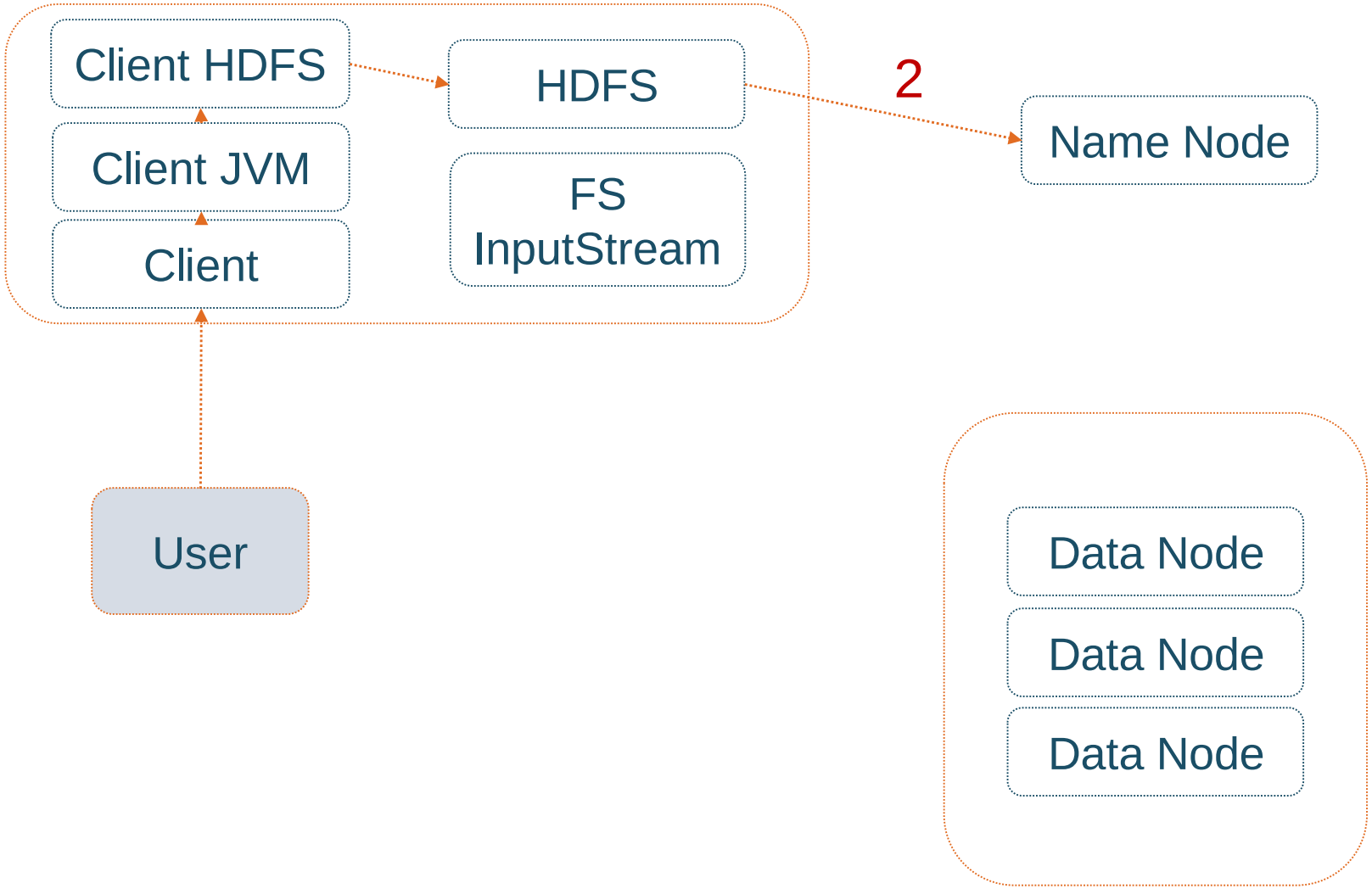
Name Node

Data Node
Data Node
Data Node

MAP – REDUCE | READ HDFS

READ

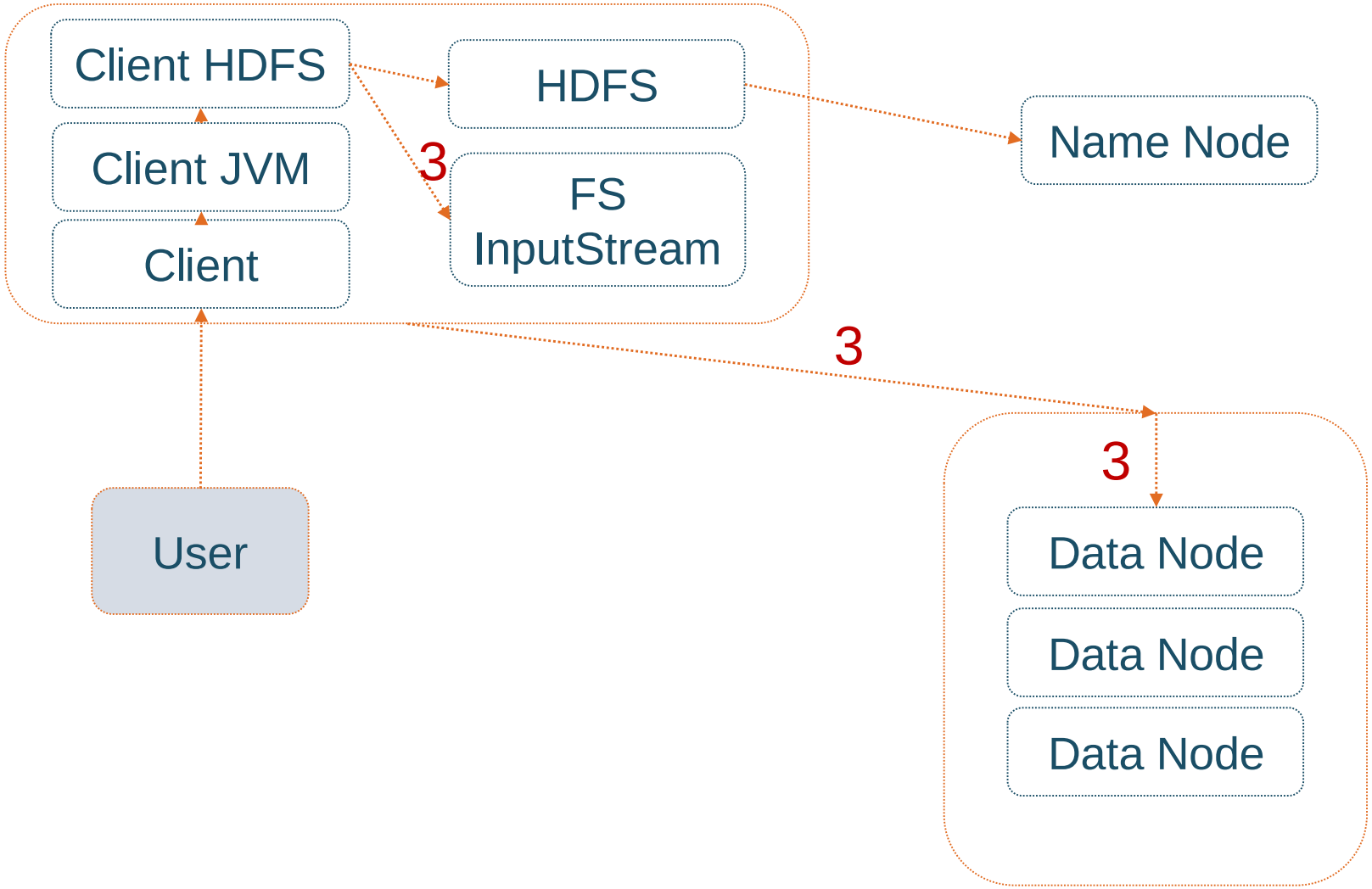
2
Получение мест расположения объектов



MAP – REDUCE | READ HDFS

READ

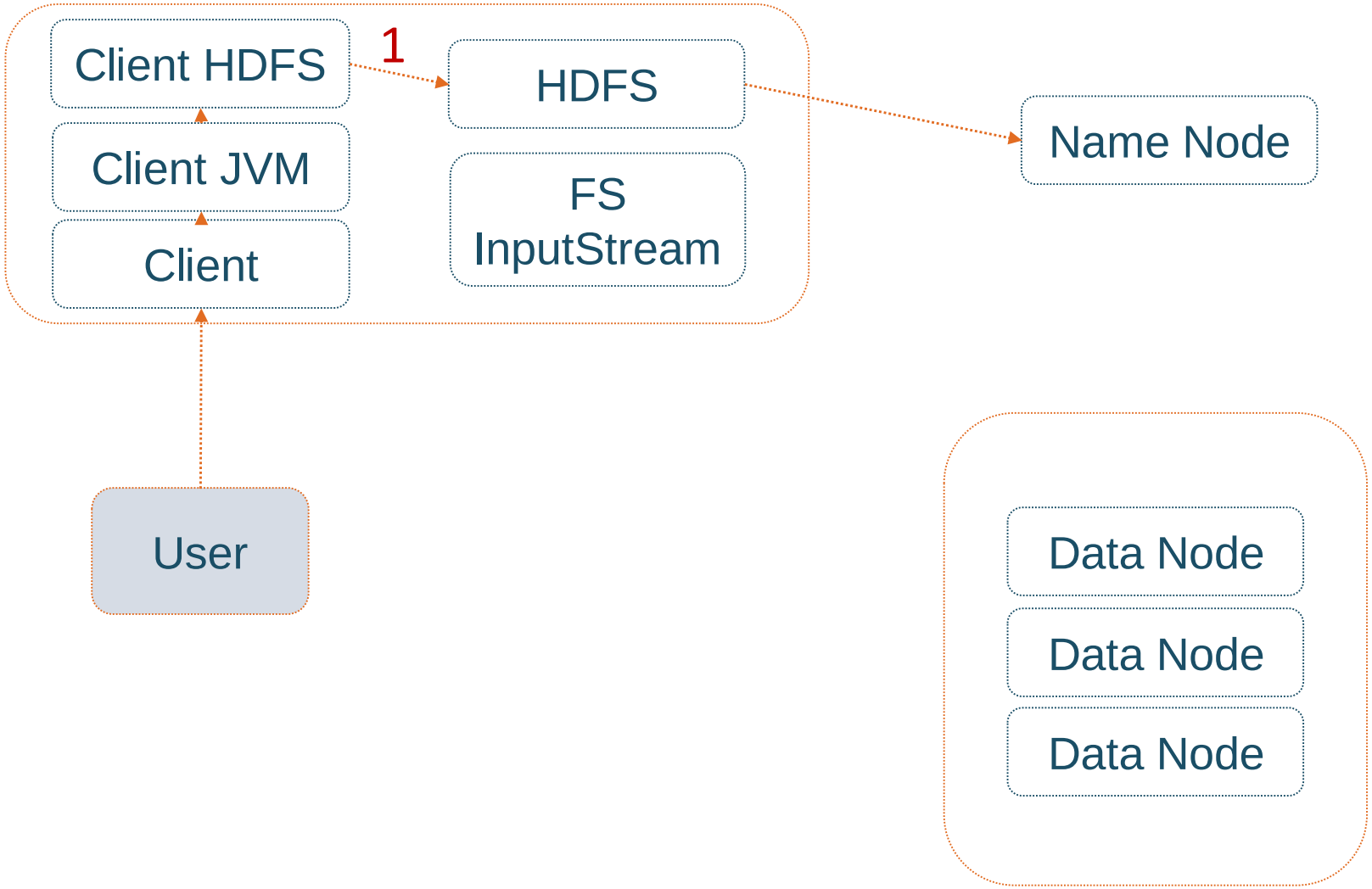
3
Чтение блоков



MAP – REDUCE | WRITE HDFS

READ

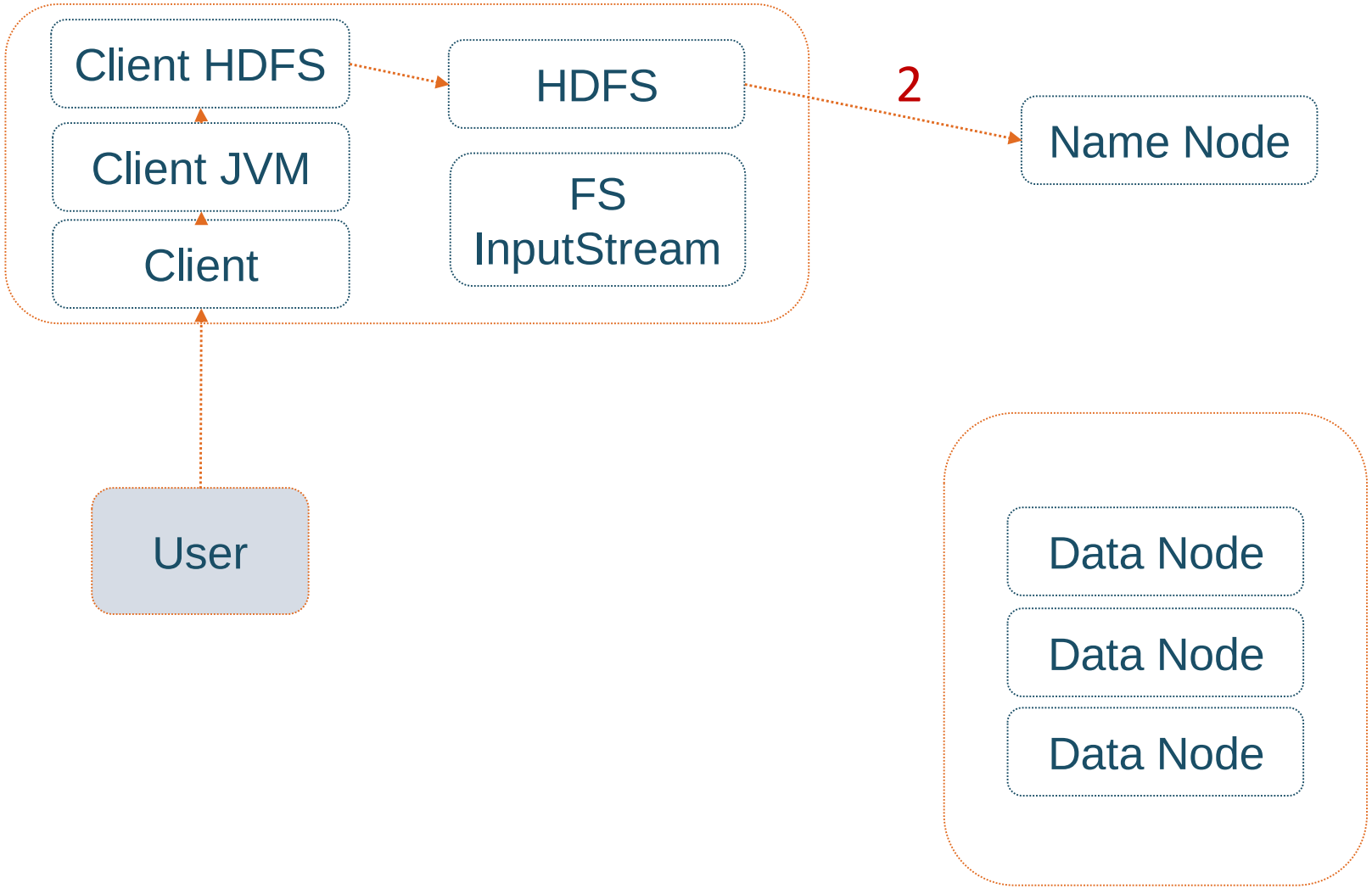
1
Создание файла



MAP – REDUCE | WRITE HDFS

READ

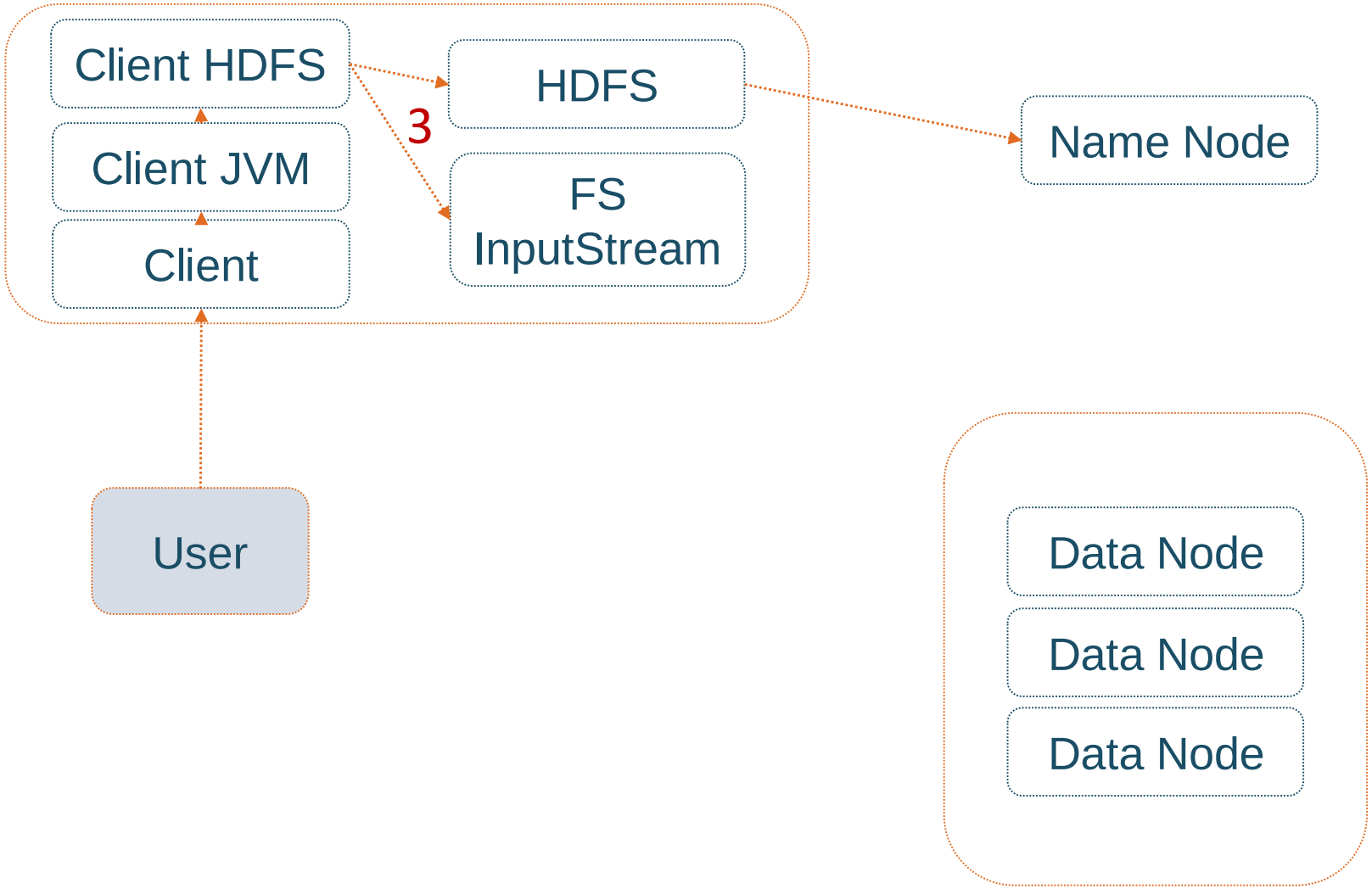
2
Получение файла
в файловой системе
(без блоков)



MAP – REDUCE | WRITE HDFS

READ

3
Запись файла

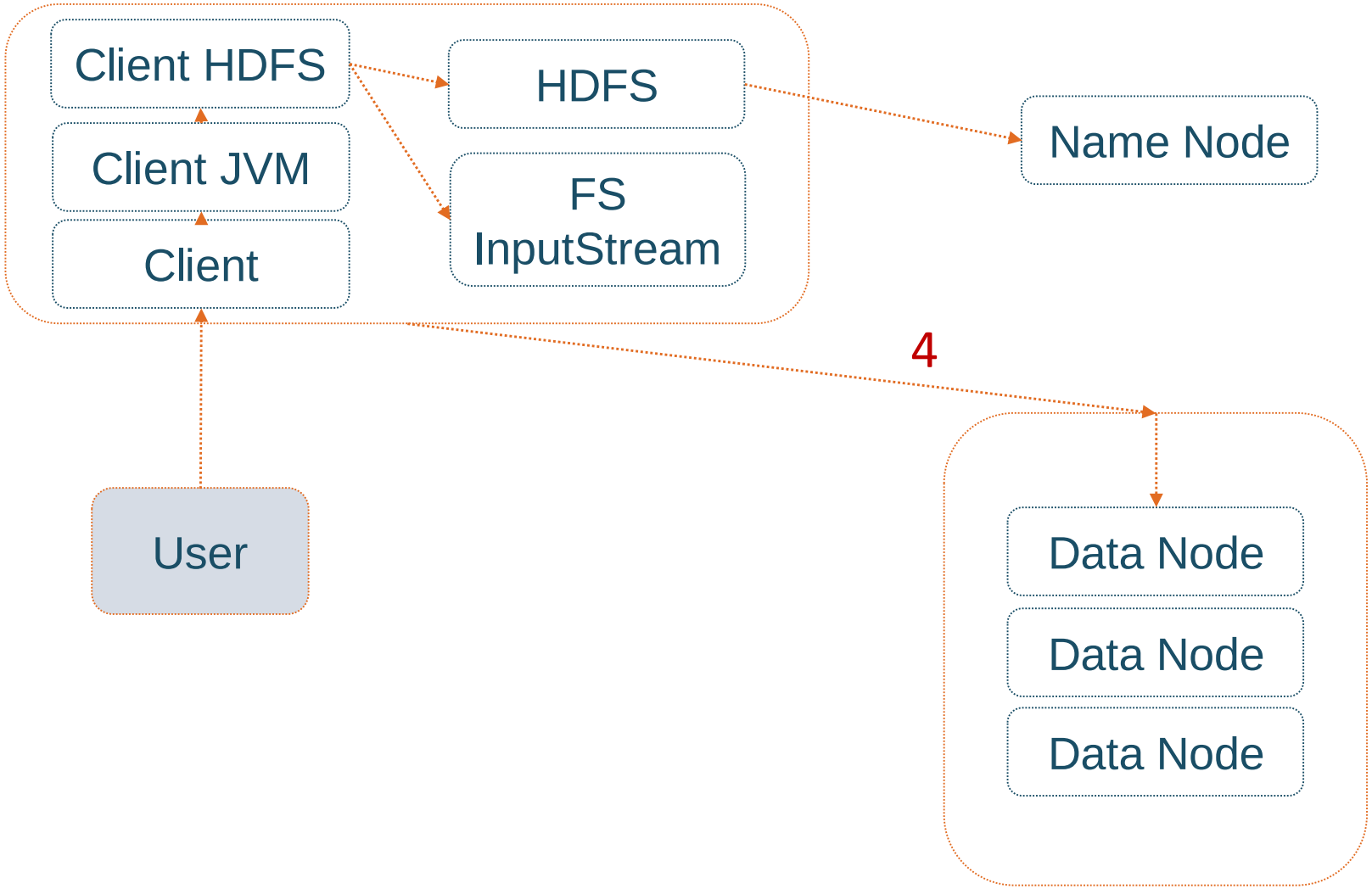


MAP – REDUCE | WRITE HDFS

READ

4
Запись файла:
- разделение на блоки
- разделениена ноды

DataNode Pipeline

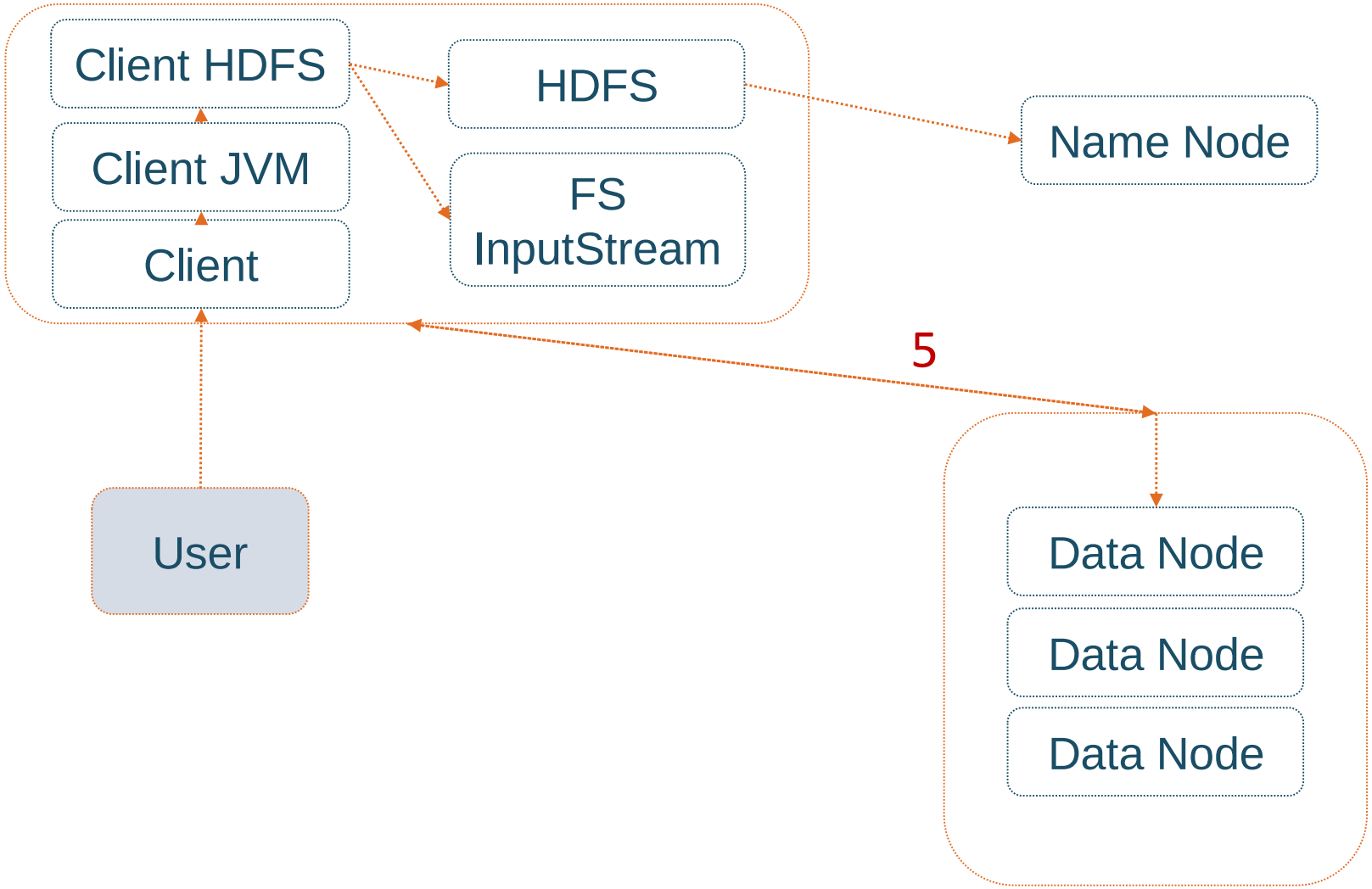


MAP – REDUCE | WRITE HDFS

READ

5
Знания о успехе

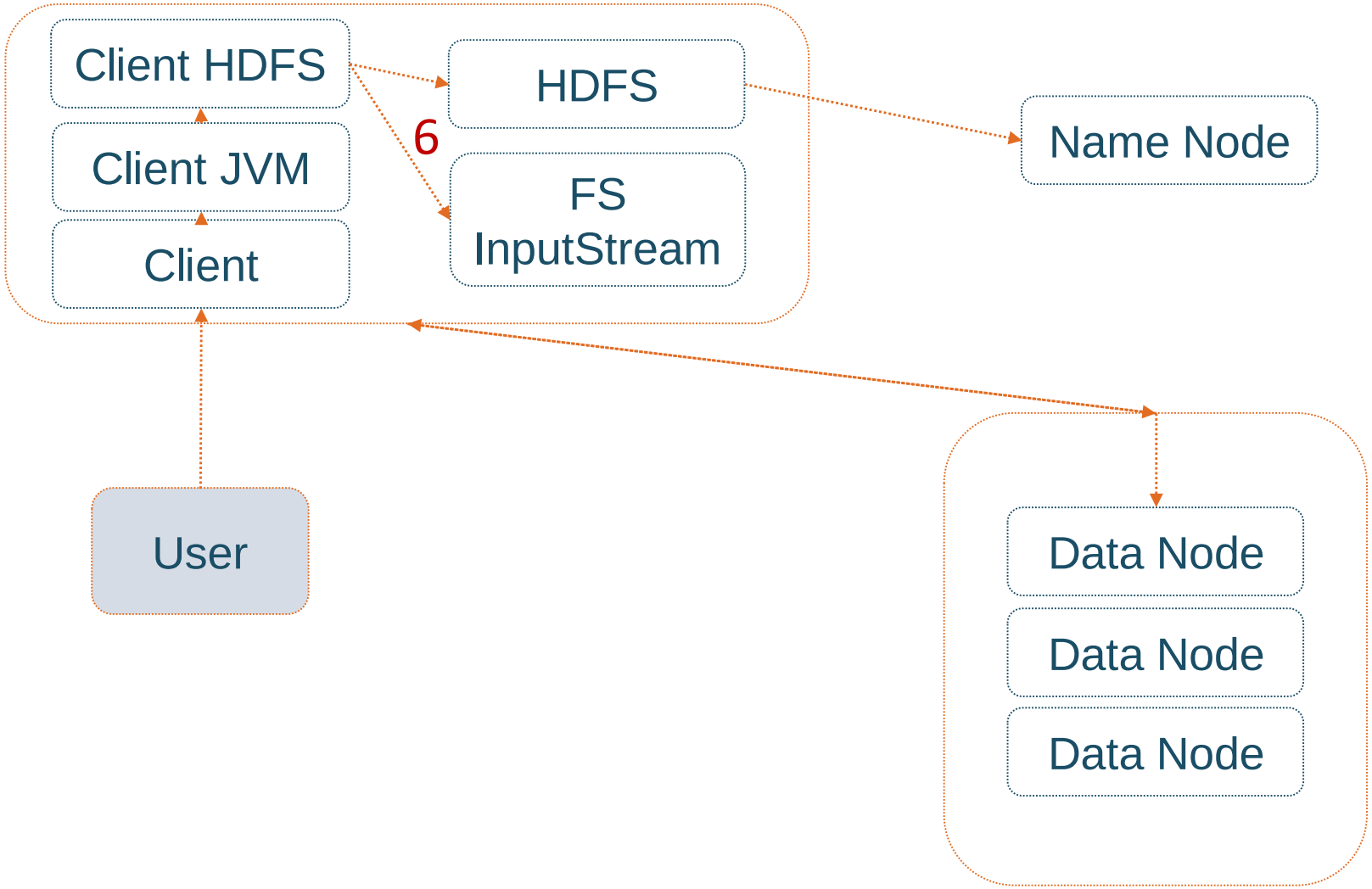
__success



MAP – REDUCE | WRITE HDFS

READ

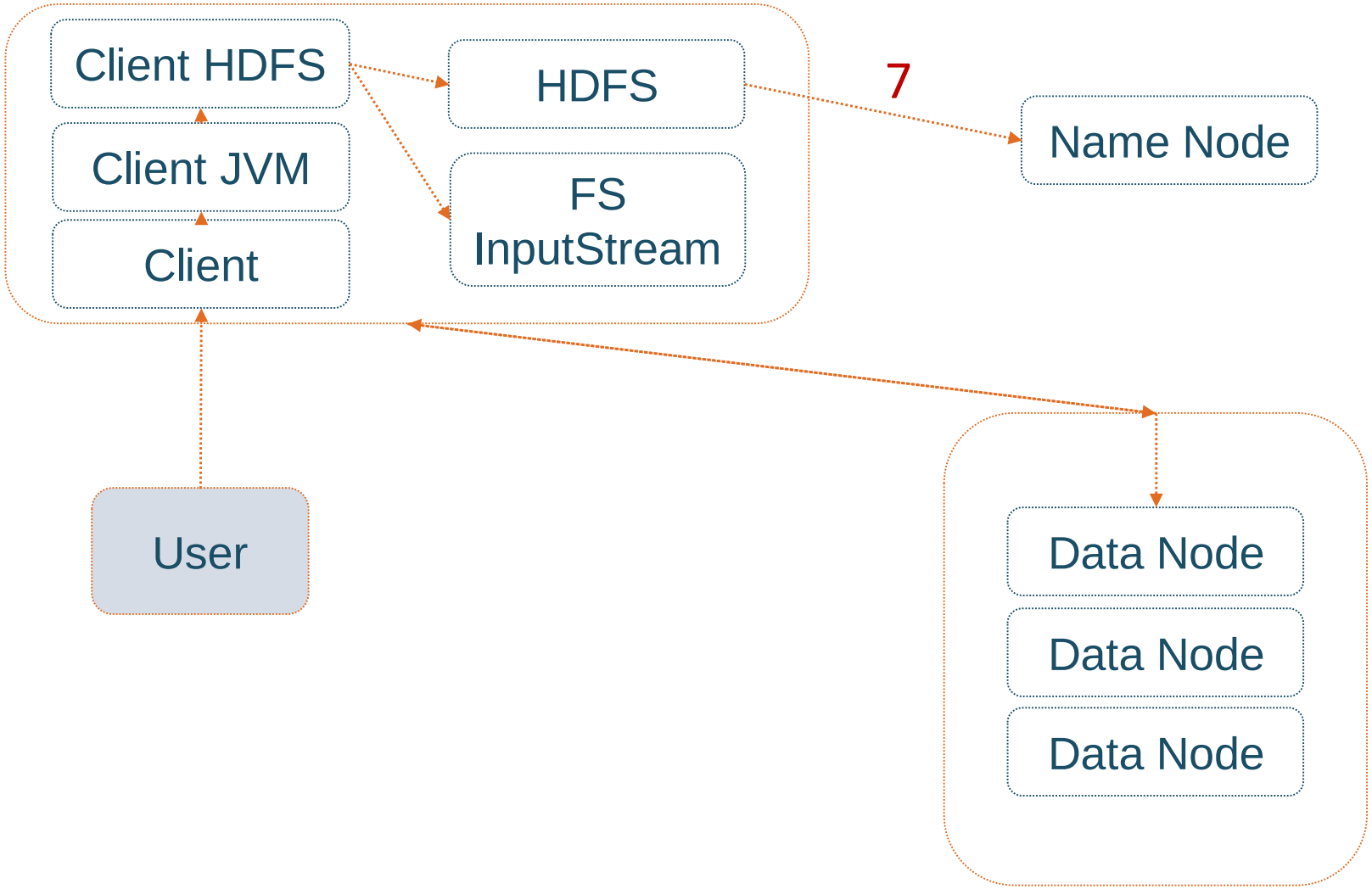
6
Закрытие операций



MAP – REDUCE | WRITE HDFS

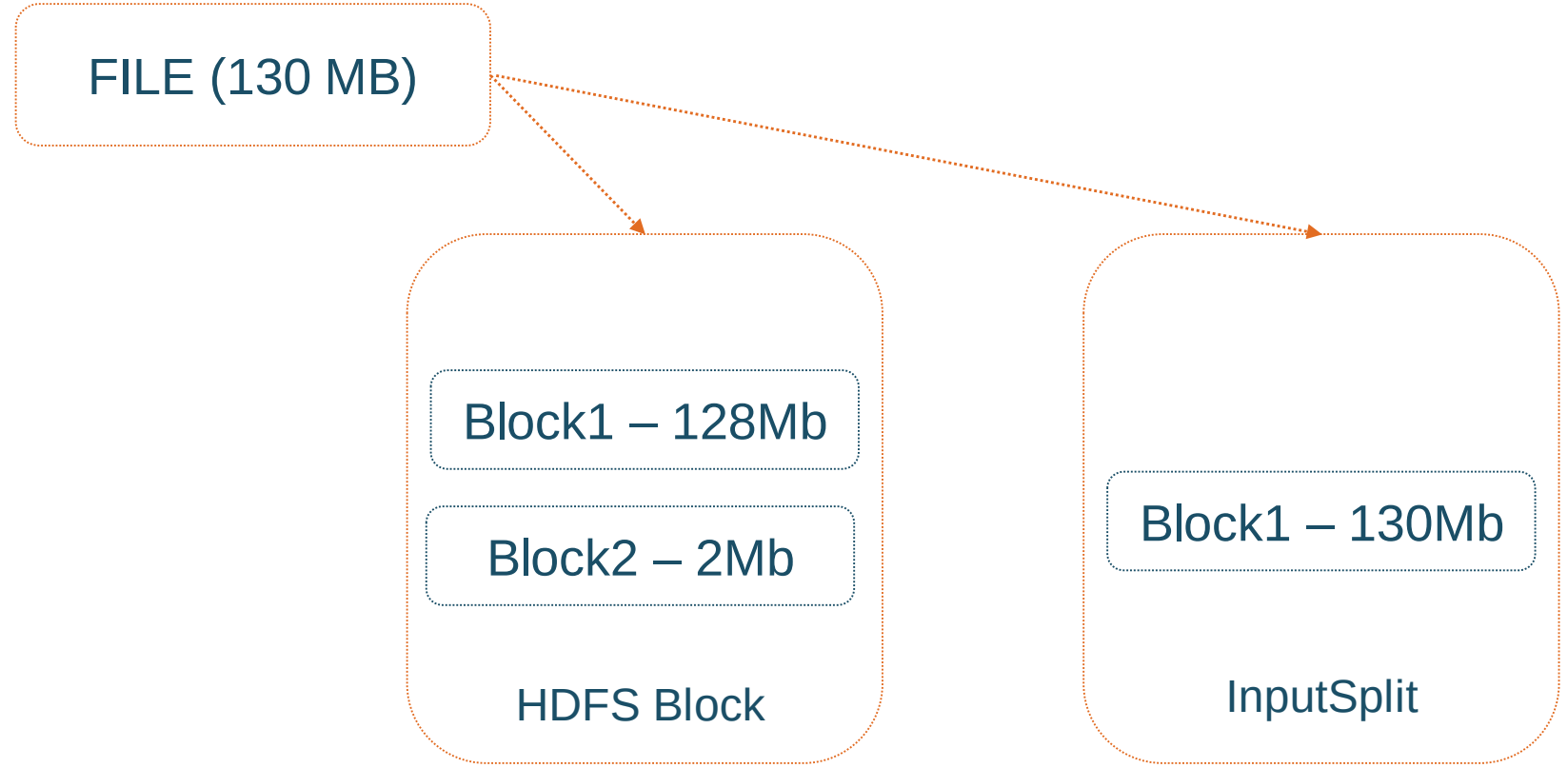
READ

7
Запись знаний
о новых данных



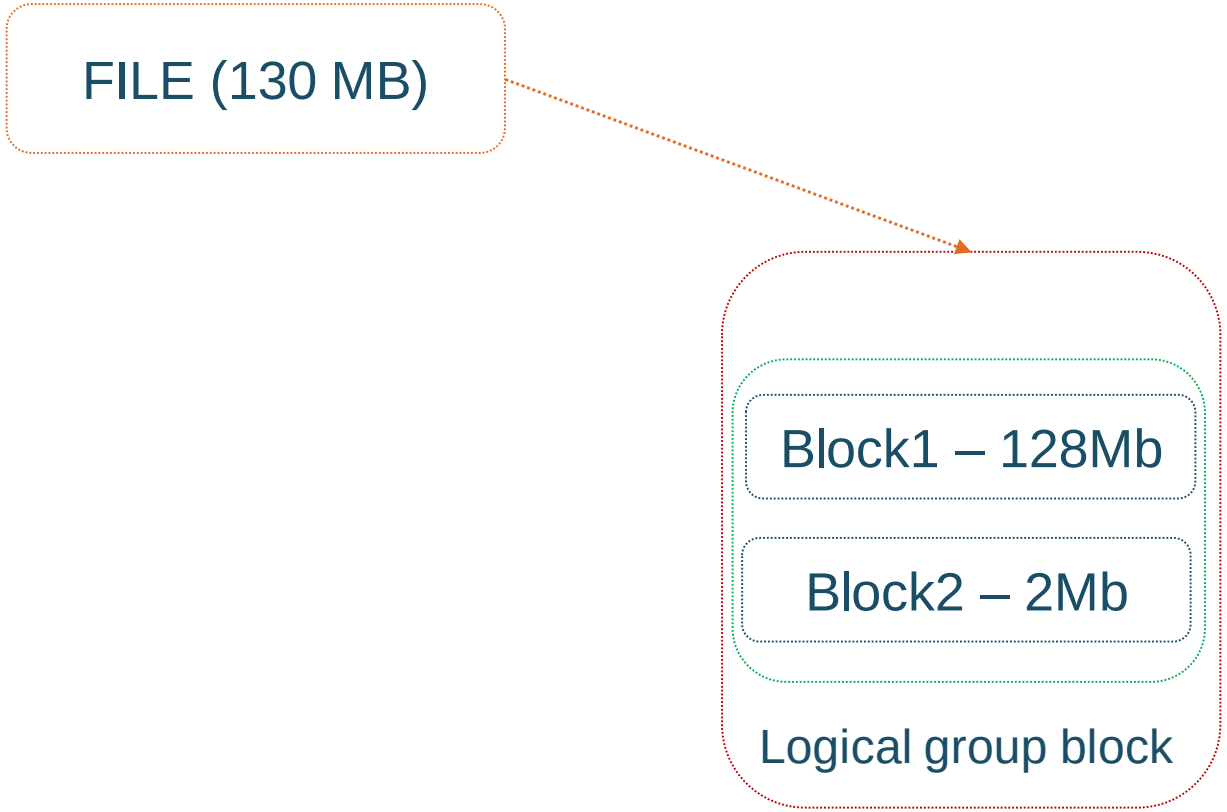
MAP – REDUCE | HDFS BLOCKS

InputSplit != HDFS Block



MAP – REDUCE | HDFS BLOCKS

InputSplit != HDFS Block



ПОПРОБУЕМ САМОСТОЯТЕЛЬНО

ДОМАШНЯЯ РАБОТА

MAR-REDUCE

- **t1_mapr. Используя библиотеку MRJOB:**

  - Исследуйте данные и определите ключ для join двух наборов данных

  - Сделайте join двух наборов

  - Определите самый популярный почтовый домен у пользователей

  - Определите куда больше всего транзачат

  - Определите популярность (топ 3):

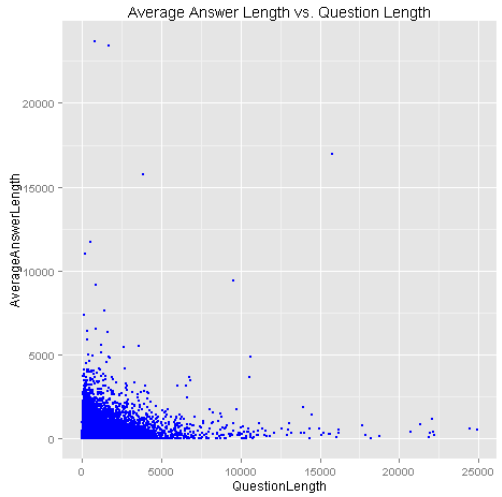
  по стране отправителя,

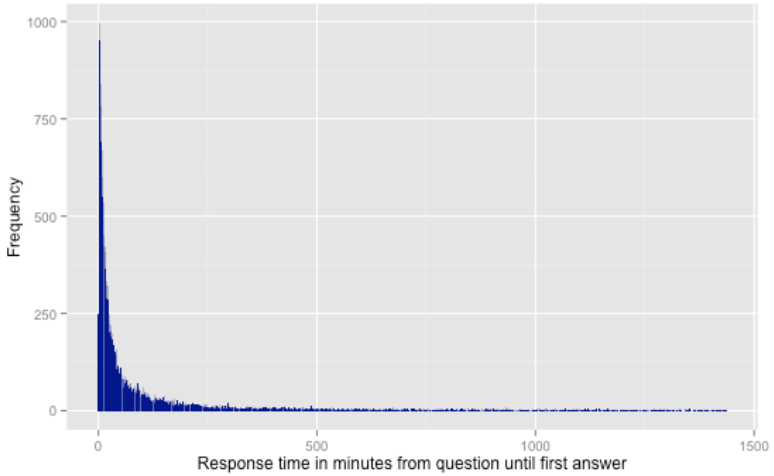
  по связке страна-домен,

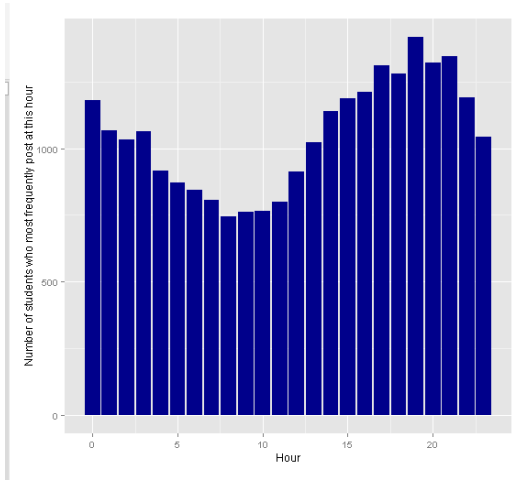
  по связке страна-транзакция

- **t2_mapreduce_viz. Используя Hadoop map-reduce извлеките данные и визуализируйте результат вычислений (пример далее)**

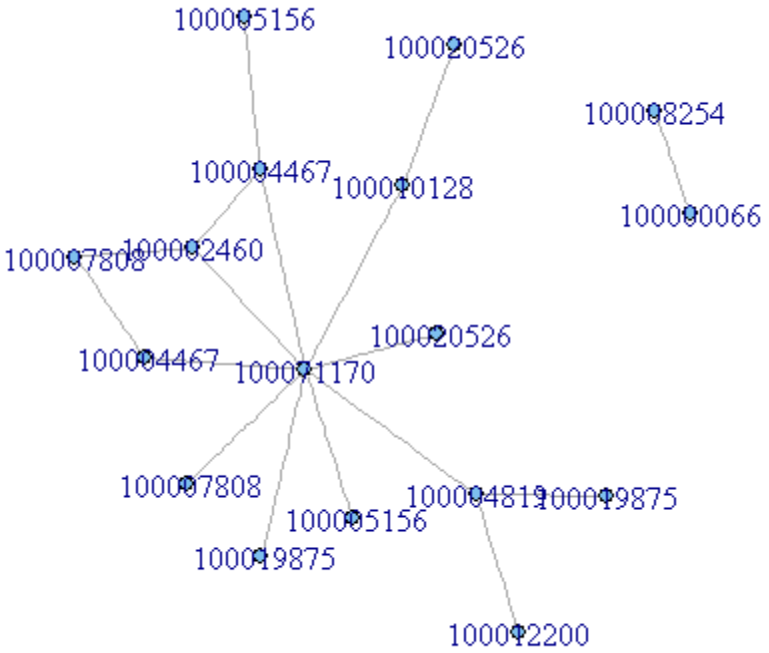
MAR-REDUCE


Average Answer Length vs. Question Length
AverageAnswerLength
QuestionLength
0
5000
10000
15000
20000
25000


Frequency
Response time in minutes from question until first answer
0
250
500
750
1000
1500


Number of students who most frequently post at this hour
Hour
0
500
1000
5
10
15
20


cs101
cs253
st101
bug
cs212
homework
meta
cs262
cs373
discussion


100005156
100020526
100008254
100004467
100010128
100000066
100007808
100002460
100020526
100004467
100001170
100007808
100005156
100019875
100012200