GeekBrains

HDFS

Принцип работы распределенной файловой системы,
репликация данных

РАБОТА С HDFS

HDFS FS SHELL

Синтаксис:

hadoop fs -<command> -<option> <URI>

hdfs dfs -<command> -<option> <URI>

URI:

hdfs://<NameNode-Host>:<Port>**/user/home**

scheme *authority* *hdfs path*

HDFS FS SHELL

Просмотр локальной файловой системы:

```
hdfs dfs -ls file:///target/path/
```

Просмотр распределенной файловой системы:

```
hdfs dfs -ls hdfs://namenode/user/sasha/
```

fs.default.name=hdfs://namenode:

```
hdfs dfs -ls /user/sasha/
```

Wildcards:

```
hdfs dfs -ls /user/sa*
```

HDFS FS SHELL

GeekBrains

Список часто используемых команд:

- cat — вывод содержимого файла
- count — посчитать количество директорий, файлов и их размер
- du — отображает размер файла или директории
- get — скопировать файл из hdfs в локальную файловую систему
- put — скопировать файл из локальной файловой системы в hdfs
- ls — отобразить список файлов
- mkdir — создать директорию
- mv — переместить файл/директорию
- rm — удалить файл

Полный список с описанием опций:

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

ПОМЕЩАЕМ ДАННЫЕ В HDFS

1. Скачаем датасет ppkm_sentiment, размеченный эмоциональной окраской отзывов индонезийской компании PPKM
2. Поместим его в контейнер:

```
# docker cp archive.zip gbhdp:/home/hduser
```

3. Распакуем:

```
$ unzip archive.zip -d ppkm
$ rm archive.zip
```

4. Копируем директорию в hdfs:

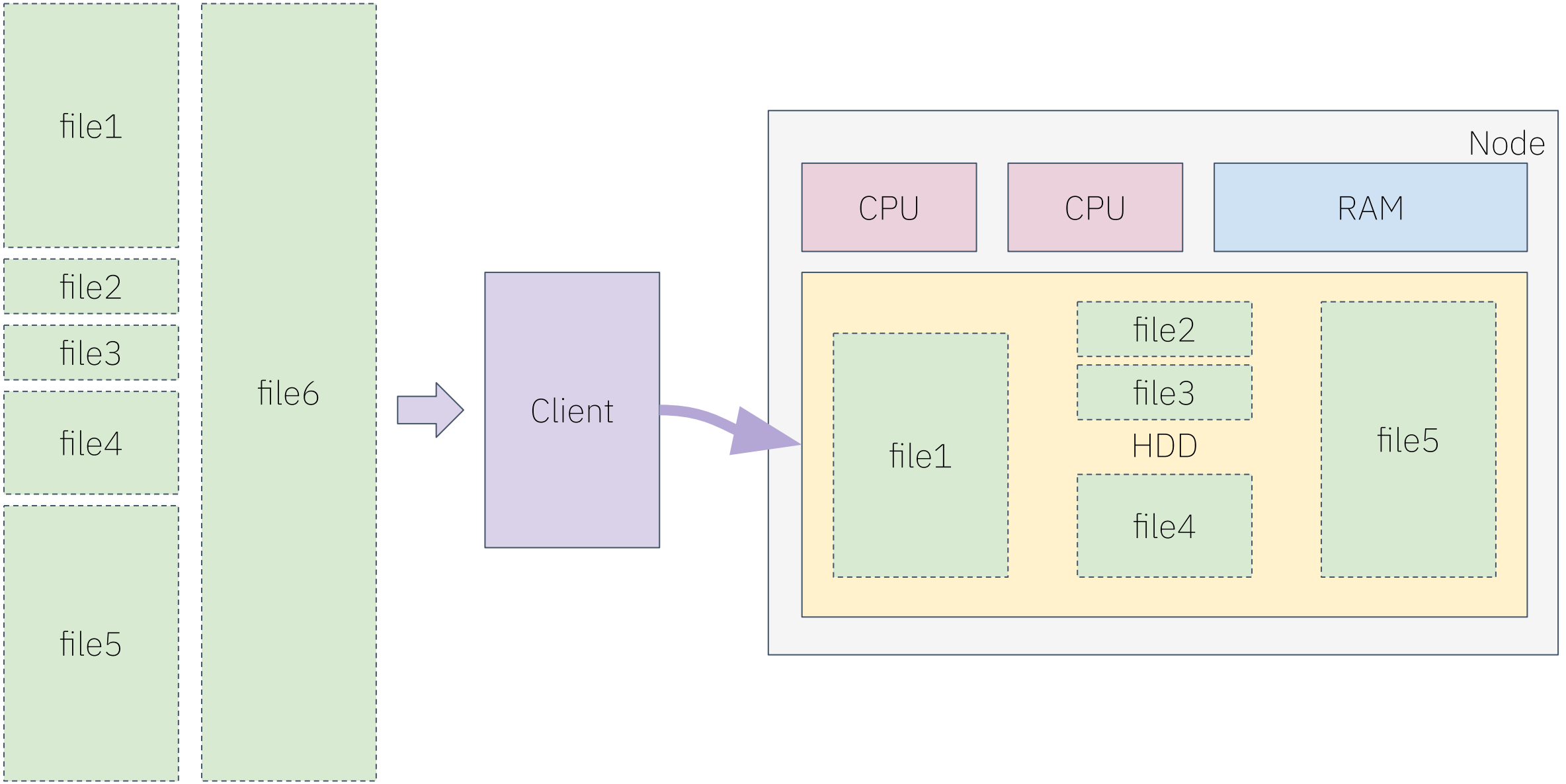
```
$ hdfs dfs -put ppkm /user/hduser/
```

5. Проверим, что файлы в hdfs:

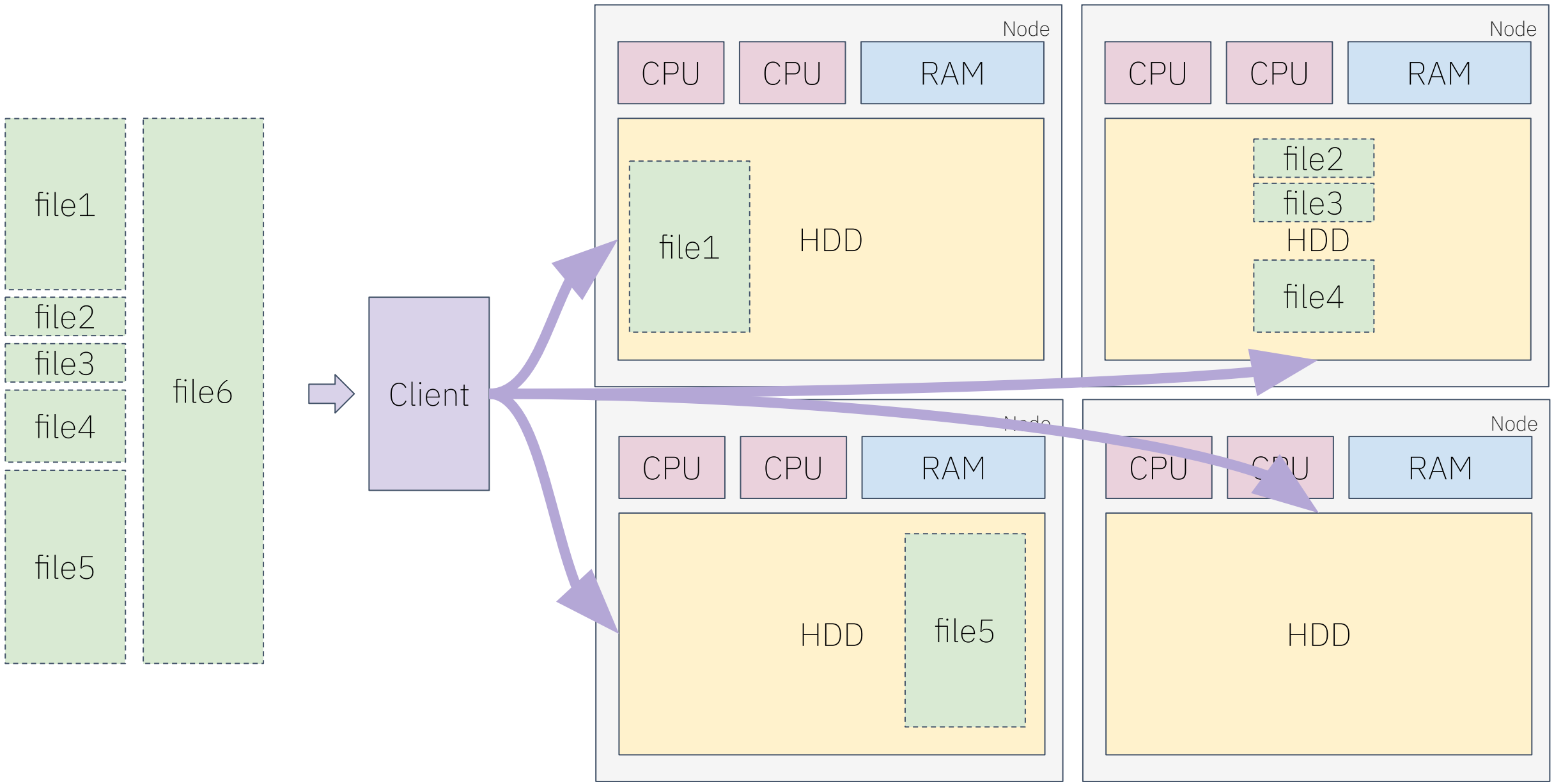
```
$ hdfs dfs -ls /user/hduser/ppkm
```

ПРИНЦИПЫ РАБОТЫ DFS

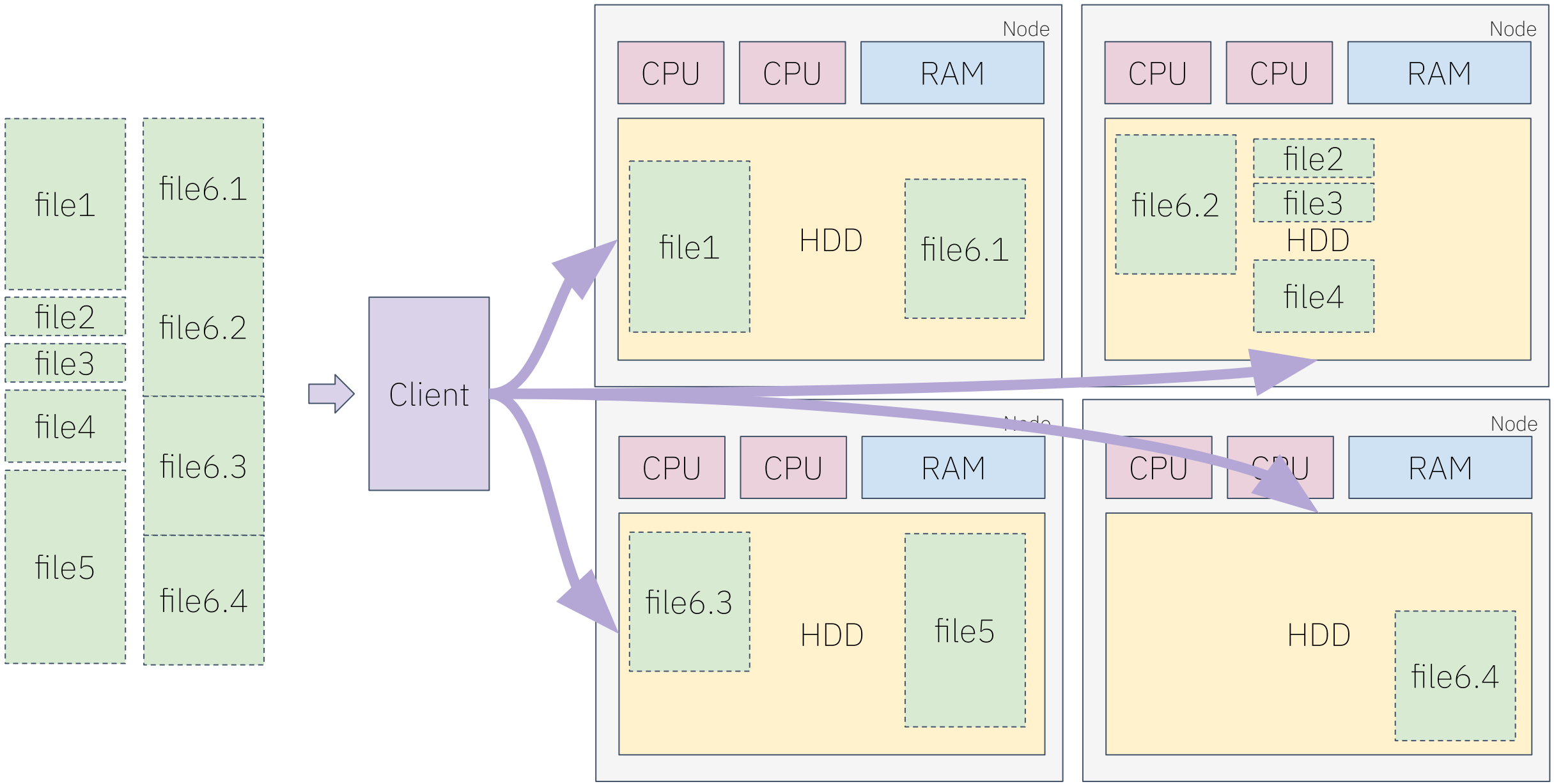
ЗАПИСЬ В HDFS



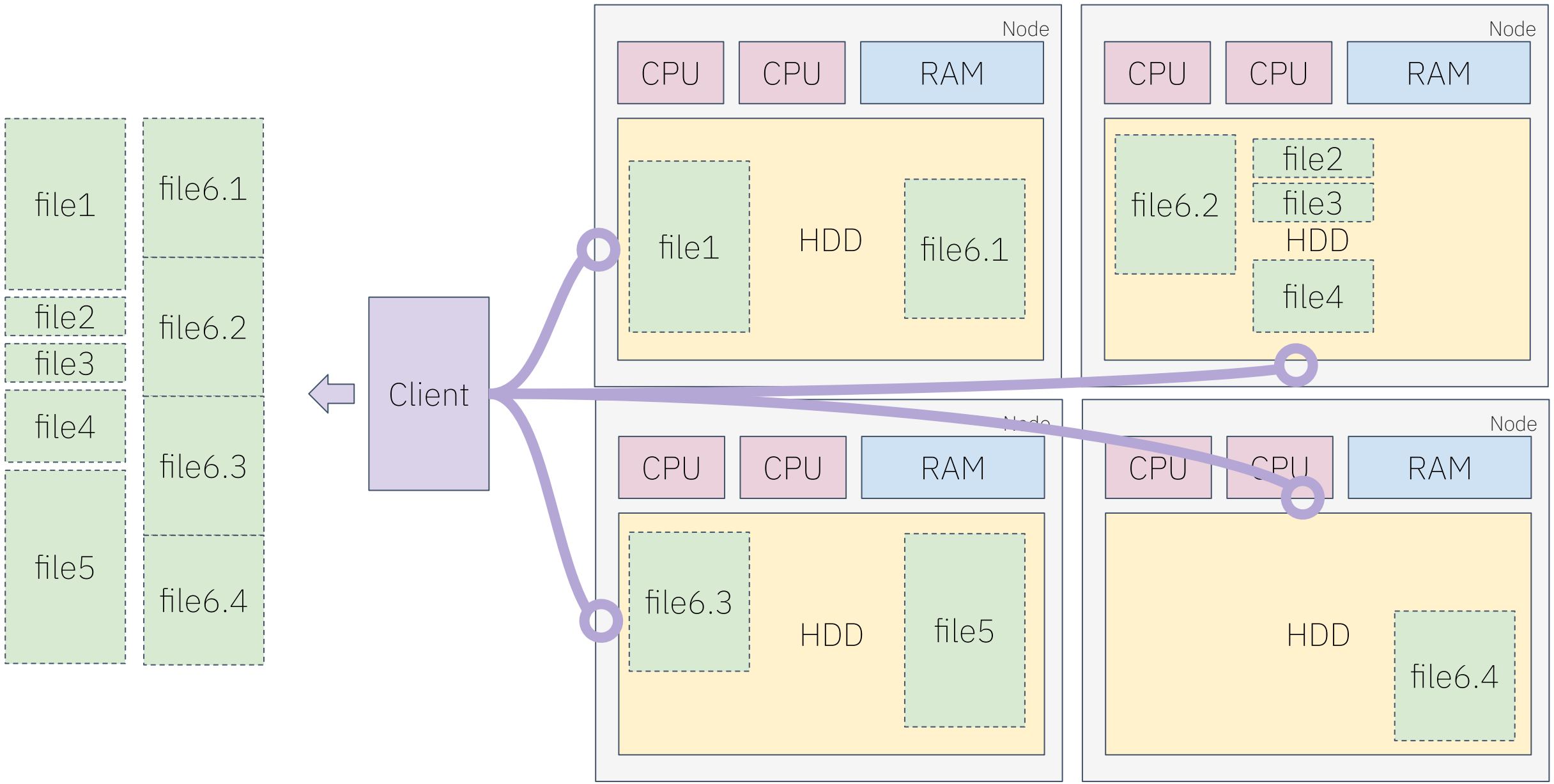
ЗАПИСЬ В HDFS



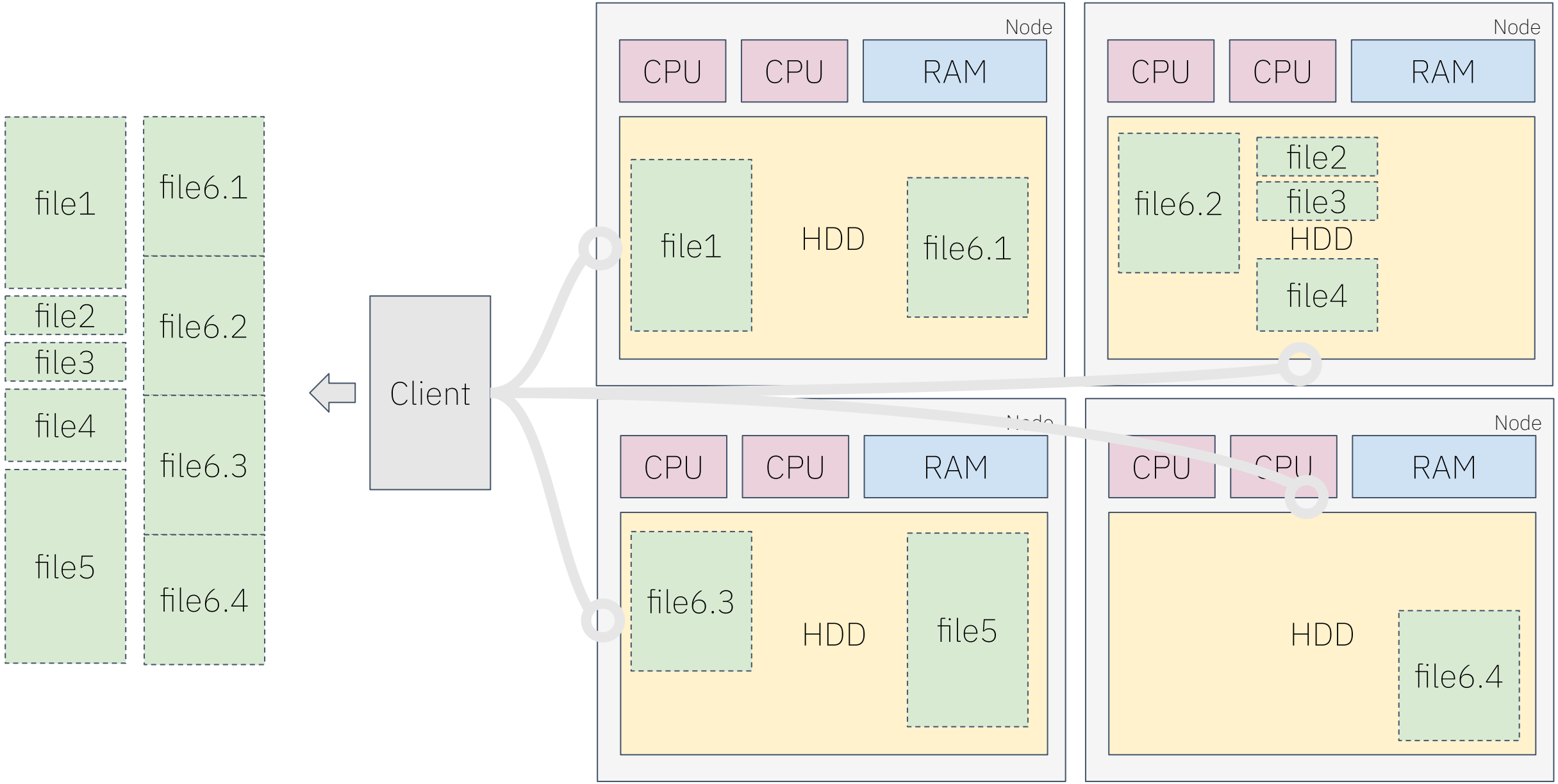
ЗАПИСЬ В HDFS



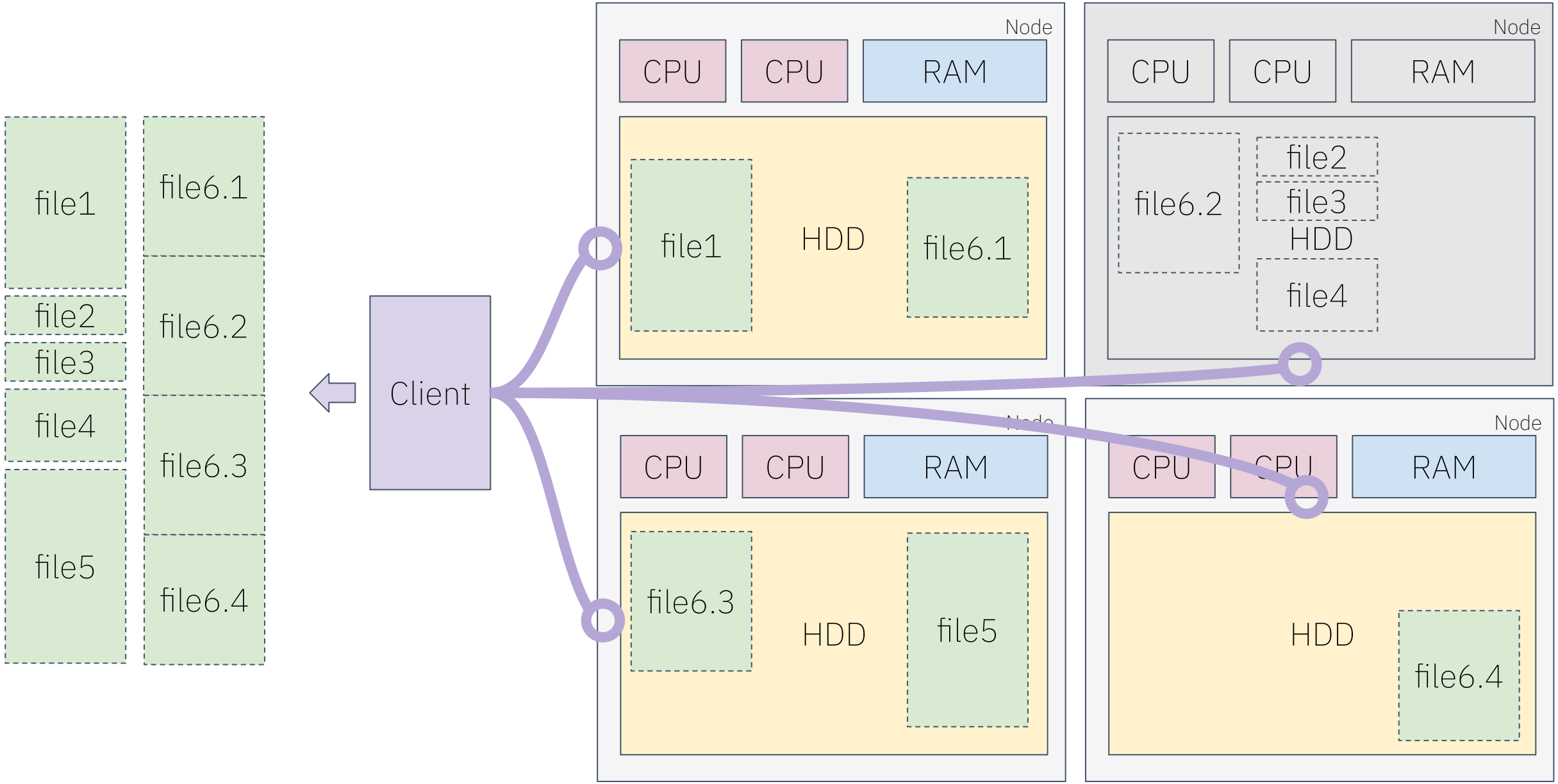
ЧТЕНИЕ ИЗ HDFS



ЧТЕНИЕ ИЗ HDFS: ПРОБЛЕМА ДОСТУПА



ЧТЕНИЕ ИЗ HDFS: ПРОБЛЕМА КОНСИСТЕНТНОСТИ



КОМПОНЕНТЫ HDFS

DATA NODE

- Хранит **блоки данных**. Размер блока по умолчанию равен 128 МБ.
- Блок данных образуется путем прозрачного разбиения файла на куски, равные размеру блока, при его попадании в HDFS
    - *Если файл меньше блока — логически ему будет выделен полный блок, физически он будет занимать свой размер*

DATA NODE

- Хранит **блоки данных**. Размер блока по умолчанию равен 128 МБ.
- Блок данных образуется путем прозрачного разбиения файла на куски, равные размеру блока, при его попадании в HDFS
  - *Если файл меньше блока — логически ему будет выделен полный блок, физически он будет занимать свой размер*
- **Copy-on-write**: блоки нельзя модифицировать, только перезаписать
  - *Нельзя писать многопоточно*
  - *Но можно писать в конец*

DATA NODE

- Хранит **блоки данных**. Размер блока по умолчанию равен 128 МБ.
- Блок данных образуется путем прозрачного разбиения файла на куски, равные размеру блока, при его попадании в HDFS
  - *Если файл меньше блока — логически ему будет выделен полный блок, физически он будет занимать свой размер*
- **Copy-on-write**: блоки нельзя модифицировать, только перезаписать
  - *Нельзя писать многопоточно*
  - *Но можно писать в конец*
- Надежность, доступность и производительность достигается **репликацией** блоков
  - *По умолчанию фактор репликации равен 3*
  - *Начиная с HDFS 3.x поддерживается EC кодирование данных*

DATA NODE

- Хранит **блоки данных**. Размер блока по умолчанию равен 128 МБ.
- Блок данных образуется путем прозрачного разбиения файла на куски, равные размеру блока, при его попадании в HDFS
  - *Если файл меньше блока — логически ему будет выделен полный блок, физически он будет занимать свой размер*
- **Copy-on-write**: блоки нельзя модифицировать, только перезаписать
  - *Нельзя писать многопоточно*
  - *Но можно писать в конец*
- Надежность, доступность и производительность достигается **репликацией** блоков
  - *По умолчанию фактор репликации равен 3*
  - *Начиная с HDFS 3.x поддерживается EC кодирование данных*
- **Rack awareness**: блоки распределяются с учетом стоек/дата центров в которых они расположены

DATA NODE

- Хранит **блоки данных**. Размер блока по умолчанию равен 128 МБ.
- Блок данных образуется путем прозрачного разбиения файла на куски, равные размеру блока, при его попадании в HDFS
  - *Если файл меньше блока — логически ему будет выделен полный блок, физически он будет занимать свой размер*
- **Copy-on-write**: блоки нельзя модифицировать, только перезаписать
  - *Нельзя писать многопоточно*
  - *Но можно писать в конец*
- Надежность, доступность и производительность достигается **репликацией** блоков
  - *По умолчанию фактор репликации равен 3*
  - *Начиная с HDFS 3.x поддерживается EC кодирование данных*
- **Rack awareness**: блоки распределяются с учетом стоек/дата центров в которых они расположены
- Отчитывается перед активной NameNode

NAME NODE

- Хранит **индекс файловой системы**: соответствие расположения блоков в DataNode с каждым файлом в HDFS и **журнал изменений** на диске
- Индекс целиком располагается в **оперативной памяти** и ограничен ею же
  - *Small Files Problem: решается HAR архивами или федерализацией*
  - *На NameNode ложится повышенная сетевая нагрузка*
  - *При потере кластер превращается в тыкву*

NAME NODE

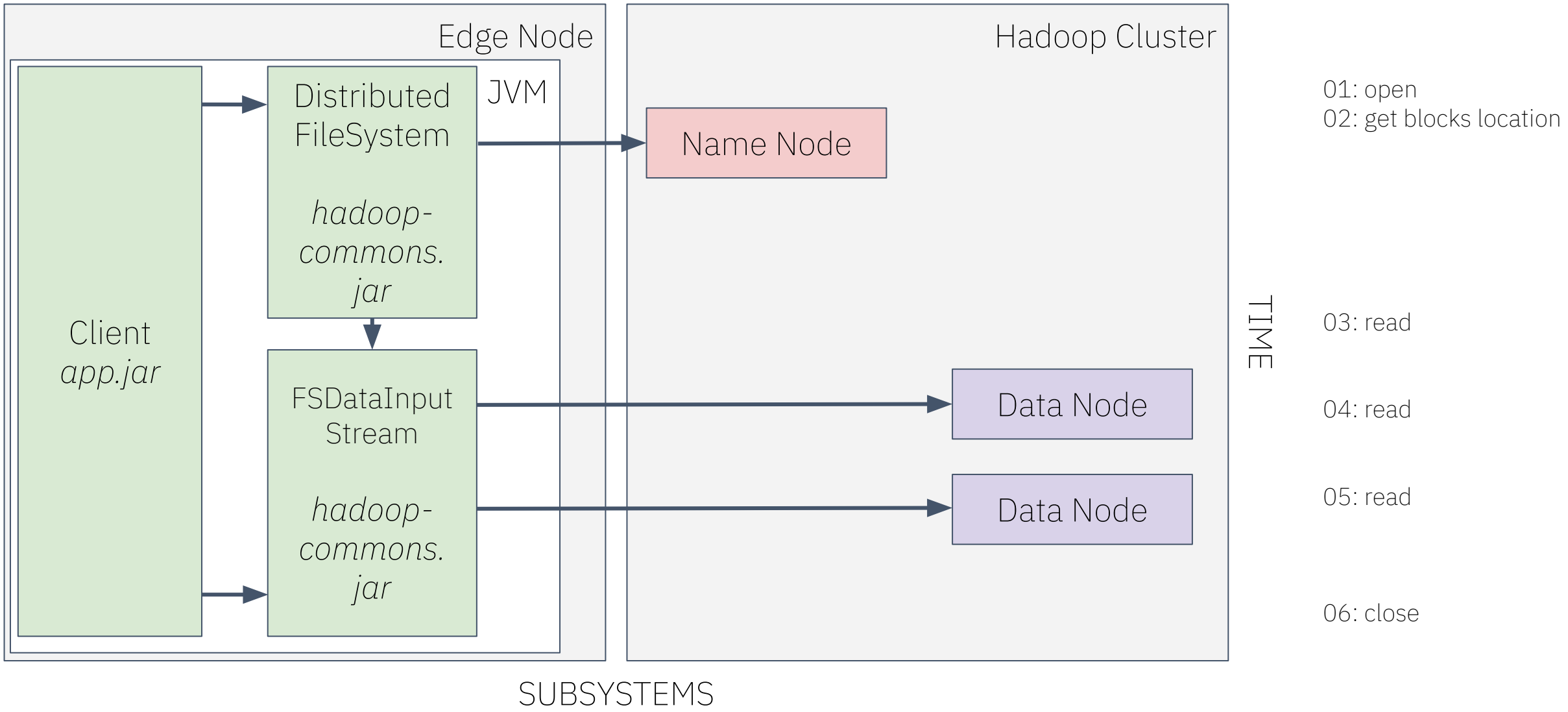
- Хранит **индекс файловой системы**: соответствие расположения блоков в DataNode с каждым файлом в HDFS и **журнал изменений** на диске
- Индекс целиком располагается в **оперативной памяти** и ограничен ею же
  - *Small Files Problem: решается HAR архивами или федерализацией*
  - *На NameNode ложится повышенная сетевая нагрузка*
  - *При потере кластер превращается в тыкву*
- **Secondary NameNode** находится в состоянии standby, в котором копирует изменения с журнала действий.

ПЕРЕРЫВ

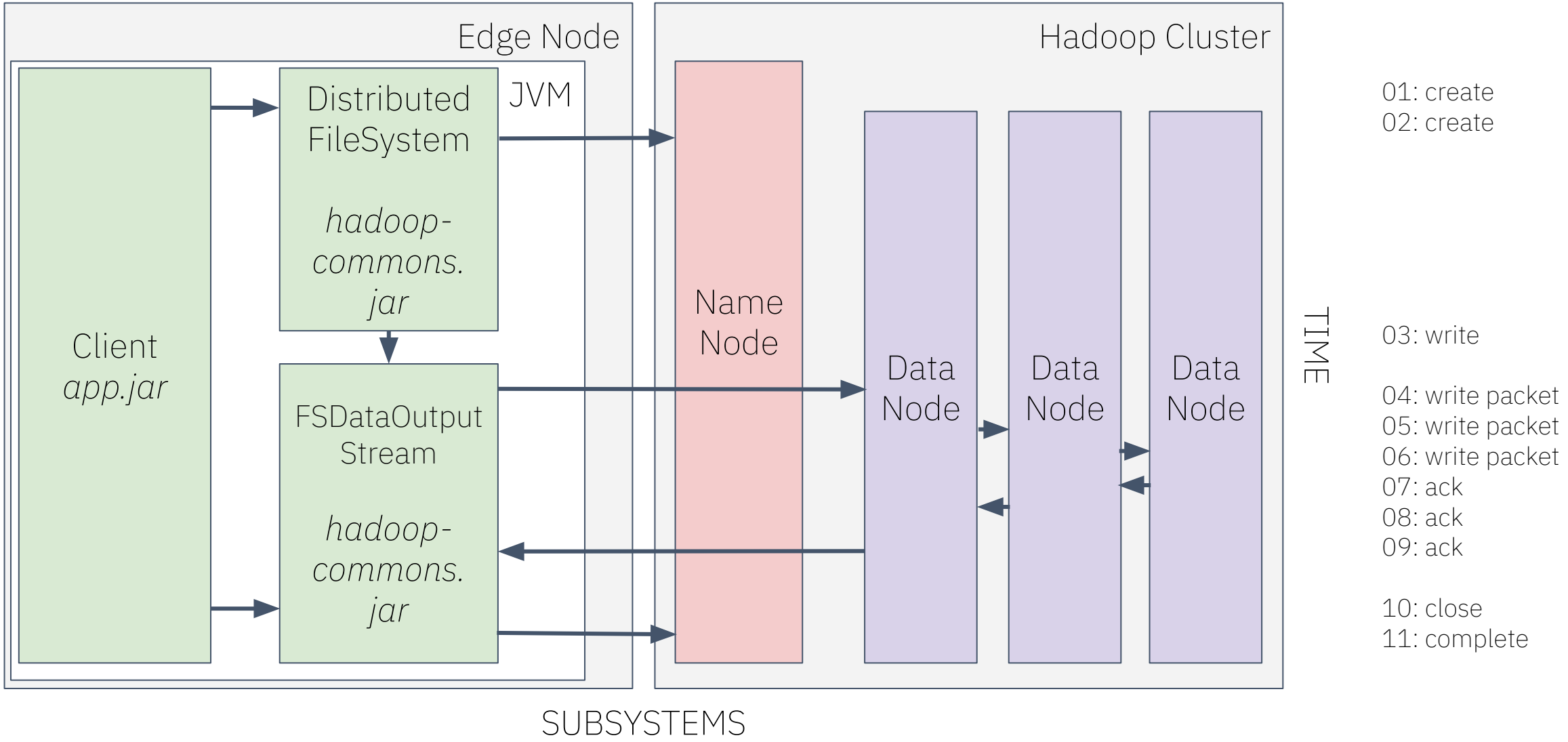

10:00

РАБОТА NAMENODE

ЧТЕНИЕ ИЗ HDFS



ЗАПИСЬ В HDFS



HDFS ADMIN

HDFS SHELL

GeekBrains

Помимо команд для работы с файловой системой (dfs) также доступны другие подсистемы:

- fsck — проверка целостности блоков
- dfsadmin — административные настройки файловой системы (сбор меты, настройка квот...)
- distcp — распределенное копирование директории Map only задачей
- ec — настройка политики EC для указанной директории

Полный список с описанием опций:

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html

ДОСТУП К HDFS

GeekBrains

- Консольные утилиты (`hadoop fs`, `hdfs dfs`)
- REST и WebHDFS
- Монтирование файловой системы через NFS Gateway или HDFS Fuse
- API

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Поместите датасет ppkm_sentiment у себя в HDFS и дайте всем пользователям на них полные права
2. Определите расположение блоков
3. У вас 20 файлов, каждый размером 130 мб. Сколько блоков будет аллоцировано в NameNode, при условии, что размер блока по умолчанию у вас 128 мб, а фактор репликации равен 3?
4. У вас 1 файл, размером 1.56 Тб. Сколько блоков будет аллоцировано в NameNode, при условии, что размер блока по умолчанию у вас 128 мб, а фактор репликации равен 3?
5. В вашей компании развернут Hadoop кластер из 400 нод. Фактор репликации равен 3. Сколько единовременно может быть выведено машин из строя, чтобы не было потери данных?

GeekBrains

Что могут спросить на собеседовании

- то такое HDFS и для чего на нужна ?
- Каковы преимущества HDFS перед другими файловыми системами ?
- HDFS vs. S3

Материалы для самостоятельного изучения

- Общая архитектура
  https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/
  https://www.edureka.co/blog/hadoop-yarn-tutorial/

- О Posix-совместимости HDFS
  https://www.quora.com/Is-HDFS-compliant-with-POSIX-Why

Спасибо!

Каждый день вы становитесь лучше :)