Лекция 8

Распределённые системы хранения больших данных и технология HDFS

6B06201 - Радиотехника, электроника и телекоммуникации

А.А. Хохлов • Алматы, 2026

Цель лекции

- Сформировать знания о принципах построения распределённых файловых систем и их применении для хранения и обработки больших массивов цифровой информации.
- Изучить архитектуру Hadoop Distributed File System (HDFS), назначение компонентов NameNode и DataNode, особенности управления метаданными и физического размещения данных на узлах кластера.
- Рассмотреть механизмы разбиения файлов на блоки, репликации, последовательности выполнения операций чтения и записи, а также восстановления доступности информации при отказах оборудования.
- Изучить принципы горизонтального масштабирования, распределения ресурсов хранения и вычислений, сетевого взаимодействия компонентов и эксплуатационного мониторинга кластера.
- Сформировать понимание возможностей и ограничений HDFS при организации платформ Big Data.

Результаты обучения

По завершении лекции обучающийся сможет:

- Объяснять архитектуру HDFS, функции NameNode, DataNode и клиента, а также принципы распределённого размещения файлов.
- Описывать процессы разбиения данных на блоки, репликации, чтения и записи информации в распределённом кластере.
- Характеризовать механизмы отказоустойчивости и масштабирования HDFS, определять области применения и ограничения технологии.

Зачем нужны распределённые файловые системы?

Множество серверов образуют единое логическое файловое пространство

Единое пространство

Приложение работает с файлами без ручного выбора физического сервера.

Масштабирование

Ёмкость можно наращивать добавлением узлов хранения в кластер.

Устойчивость

Копии блоков на разных узлах помогают переносить отдельные отказы.

Примеры данных: журналы событий, телеметрия, архивы измерений, Big Data и научные наборы.

HDFS в экосистеме Hadoop

Hadoop Distributed File System: крупные файлы и потоковая обработка

Сильная сторона

Высокая суммарная пропускная способность при последовательном чтении и пакетной аналитике.

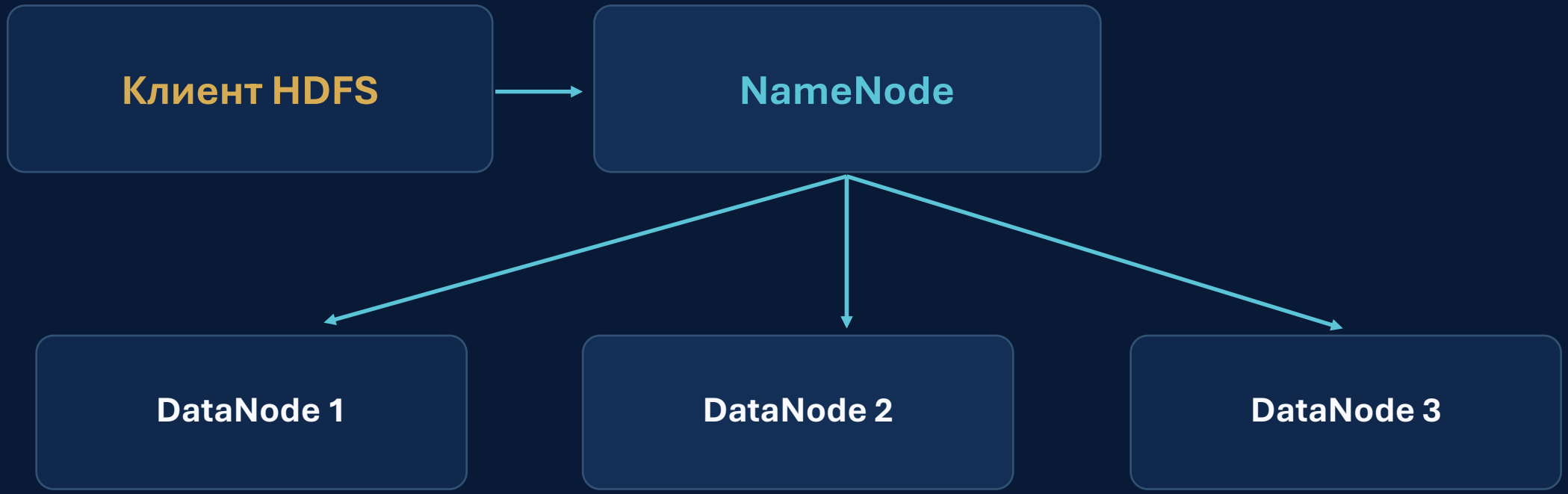
Ограничения

Множество мелких файлов и частые произвольные изменения требуют особого подхода.

HDFS - распределённая файловая система, а не база данных и не универсальная замена сетевому диску.

Три роли в архитектуре HDFS

Метаданные отделены от пользовательских блоков данных



NameNode управляет именами файлов и блоками; DataNode хранят содержимое.

Разбиение файла на блоки

Файл представляется как последовательность блоков, распределённых по узлам



Метаданные задают порядок блоков; физическое размещение скрыто от приложения.

Репликация блоков и распределение по стойкам

Пример: три копии блока на разных серверах; число реплик задаётся политикой



Репликация снижает риск потери доступа при отказе узла, но не заменяет независимый backup.

Запись файла в HDFS: поток данных

NameNode организует размещение; содержимое передаётся по цепочке DataNode

1

Клиент

Создание файла

→

2

NameNode

Разрешение и адреса

→

3

DataNode A

Первый блок

→

4

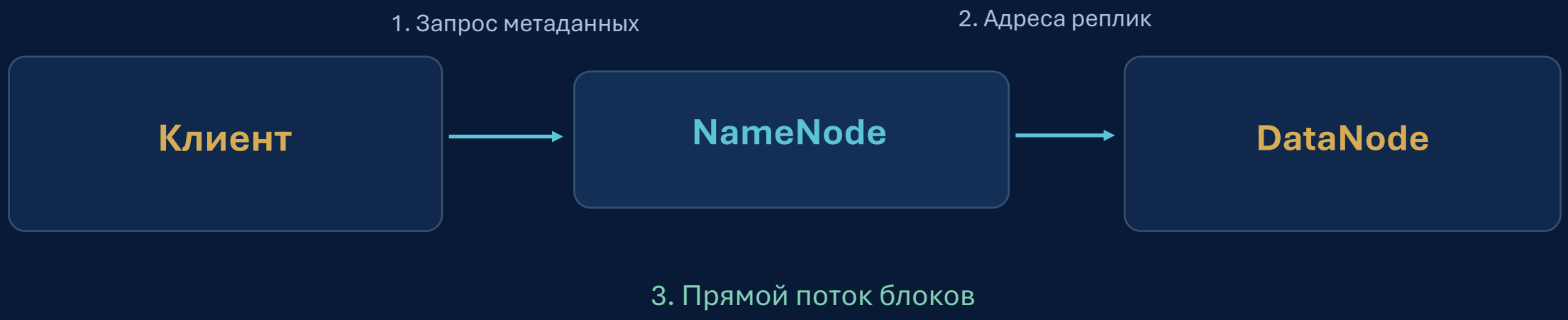
DataNode B / C

Реплики и ACK

После подтверждения реплик и завершения записи файл закрывается, метаданные обновляются.

Чтение файла в HDFS: путь запроса

NameNode не передаёт полезные блоки клиенту



Если один узел недоступен, клиент может обратиться к другой копии блока.

Сеть и служебные сообщения HDFS

Heartbeat

Регулярная информация о доступности DataNode для управляющего сервиса.

Отчёт о блоках

Сведения о том, какие блоки и реплики фактически хранятся на узлах.

TCP/IP и RPC

Связь клиента, NameNode и DataNode; важны DNS, адресация и доступность портов.

Восстановление копий

Дополнительная нагрузка на диски и сеть после отказа узла.

Отказоустойчивость: узлы и метаданные

Потеря DataNode

Оцениваются доступные реплики, затем создаются дополнительные копии.

Защита NameNode

Метаданные требуют сохранности; в HA применяют активный и резервный узлы.

Контроль кластера

Недореплицированные блоки, ошибки носителей, заполнение и состояние сети.

Отдельный backup

Реплики не защищают автоматически от логического повреждения и удаления.

Вычисления рядом с данными или отдельно?

Два способа построения инфраструктуры больших данных

Совмещённые узлы

Хранение и вычислительные процессы на одном сервере: локальность данных, связанное масштабирование.

Разделённые ресурсы

Вычислительные узлы и самостоятельный кластер хранения: независимое расширение при достаточной сети.

Выбор зависит от нагрузки, затрат на сеть, задержек и частоты доступа к данным.

Применение HDFS в университетской среде

Сбор данных, распределённое хранение и пакетный анализ



Для научной воспроизводимости важно хранить происхождение, версии и контроль целостности данных.

Администрирование и расширение HDFS

Перед запуском

Проверить сеть, диски, время, безопасность, доступность NameNode и регистрацию DataNode.

Во время работы

Следить за заполнением, числом узлов, здоровьем блоков и числом недостающих реплик.

При расширении

Добавить DataNode и оценить необходимость перераспределения блоков.

При выводе узла

Переместить нужные реплики до отключения; проверить доступность данных.

Итоги и контрольный вопрос

- HDFS хранит большие файлы в виде блоков на нескольких узлах DataNode.
- NameNode управляет пространством имён, правами и размещением блоков.
- Чтение и запись данных идут напрямую между клиентом и узлами хранения.
- Репликация, контроль метаданных и сеть определяют доступность кластера.

Вопрос: почему NameNode обычно не передаёт клиенту содержимое файлов?