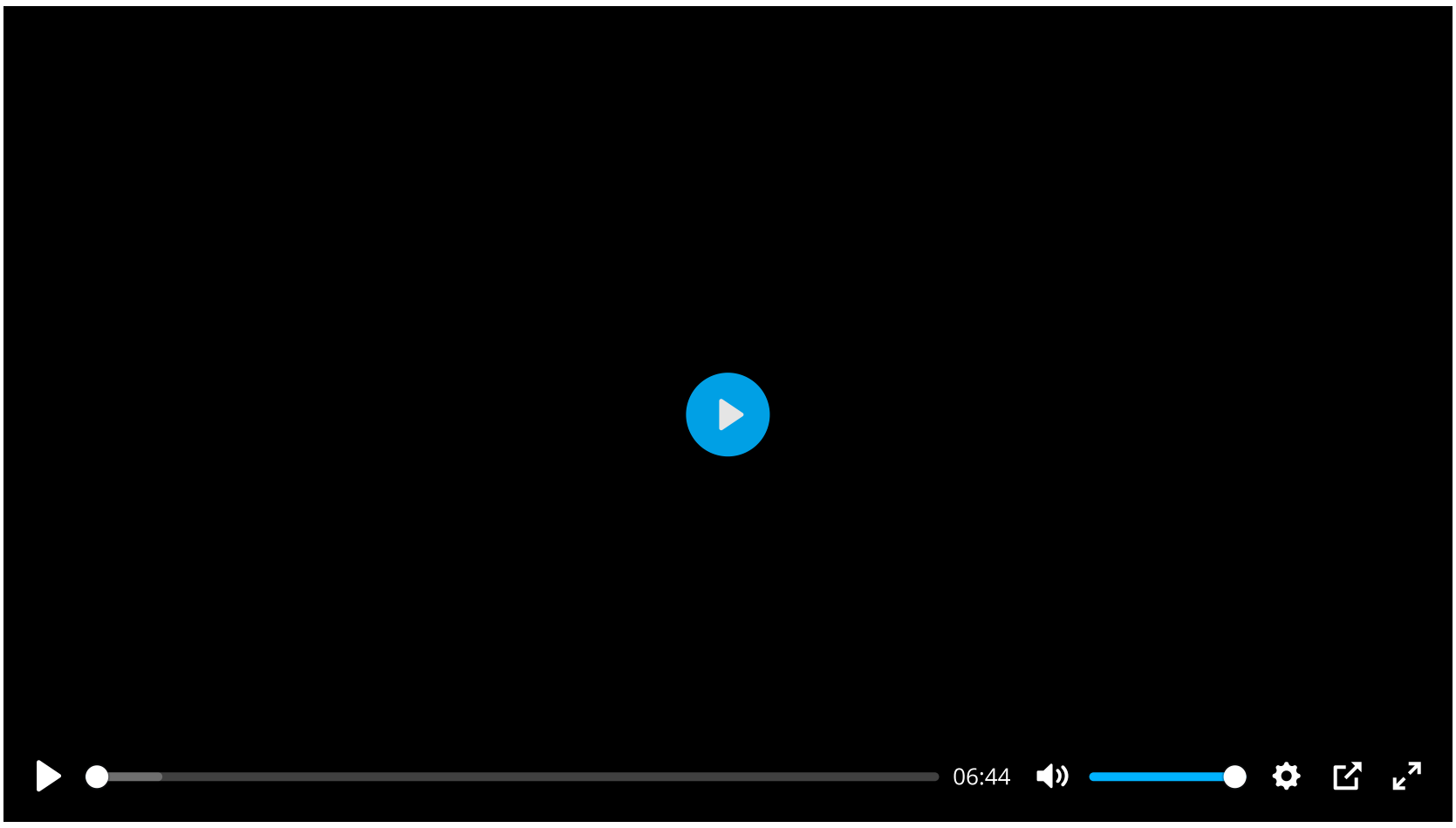
4.2 Что такое Spark и зачем он нужен DE. Введение в Resilient Distributed Datasets



Текстовая расшифровка видео:

КАК УСТРОЕН SPARK

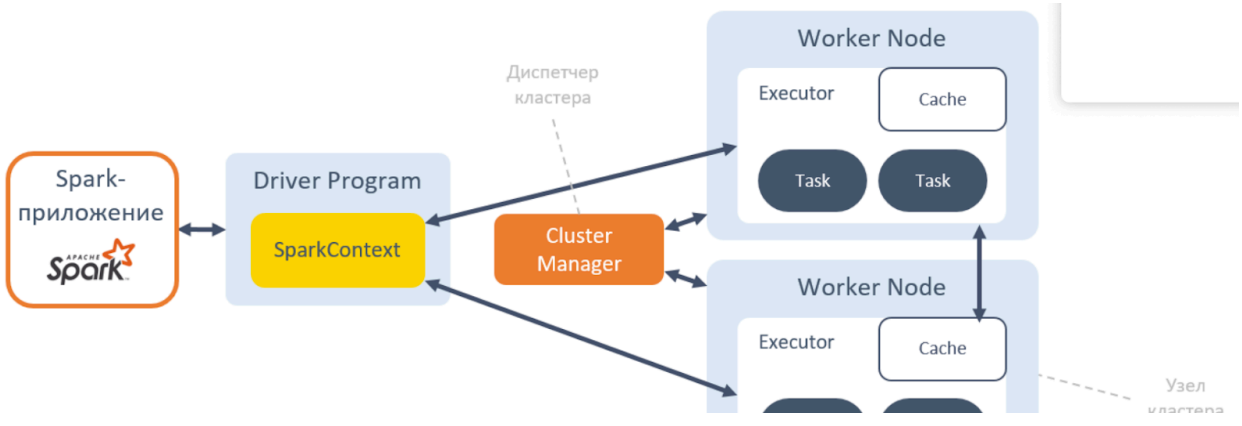
План:

- Упрощенная схема работы;
- Основные компоненты;
- Логистический граф;
- Менеджер кластера – YARN;
- Итоги.

Упрощенная схема работы

Задача данного фреймворка – максимально абстрагировать сложность.

С клиентской стороны Spark предоставляет driver, от которого мы получаем API. Драйвер выдает нам специальный объект, который можем импортировать и через него получить доступ к кластеру. С этим объектом мы можем по-разному работать, вызывать у него разные методы и запускать распределенные вычисления. Для управления задачами на кластере используется cluster manager (диспетчер кластера), который все «размазывает». Также есть какое-то количество рабочих машин (worker nodes), на которых запускаются процессы и происходят вычисления:



Запустить стенд

Дедлайн 07 июля, 23:59 Мск

Основные компоненты

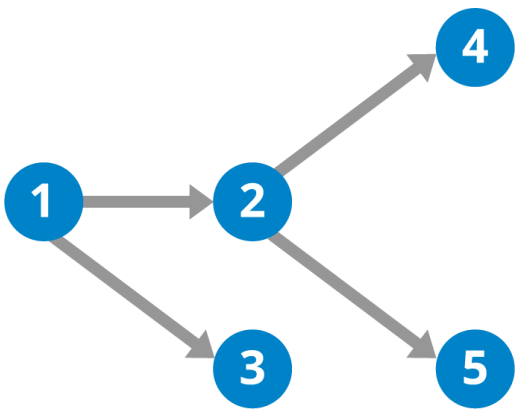
Driver выступает основной точкой входа, он:

- Управляет информацией о приложении;
- Откликается на команды пользователя;
- Анализирует и распределяет работу для воркеров.

Worker’ы (Executors):

- Обрабатывают данные;
- Получают задачи не напрямую, а через driver;
- Работают с одним или несколькими Partition’ами.

Логистический граф



На основе кода нашей программы, по тому, как мы взаимодействуем с API Spark’a, драйвер строит направленный ациклический граф (Directed Acyclic Graph – DAG), который состоит из нод и связывающих их ребер. Это порядок действий, который мы должны выполнить, чтобы получить результат в конце.

Driver строит направленный ациклический граф (DAG) из задач на основе кода пользователя. Однако проблема не только в параллелизме, но и в балансировке задач на Executor’ах. При равном объеме работы на всех исполнителях кластер будет работать эффективнее всего. Это касается и кода, и данных.

Проблема скорости выполнения на любой распределенной задаче описывается законом Амдала. Когда мы параллелизовали алгоритм по максимому, дальнейшее добавление машин или параллельных обработчиков в процесс не ускорит процесс, а только замедлит его.

Менеджер кластера – YARN

Для управления ресурсами кластера Spark использует **YARN (Yet Another Resource Negotiator)** – стандартный механизм в Hadoop-инсталляциях для управления задачами.

У Spark’a есть собственный встроенный менеджер задач StandAlone scheduler, однако чаще всего в продуктовых инсталляциях его настраивают с поддержкой YARN.

Менеджер кластера – YARN:

- Отвечает за поиск ресурсов для запуска и Driver’a и Worker’ов;
- Квотирует ресурсы между пользователями;
- Отслеживает статус выполнения.