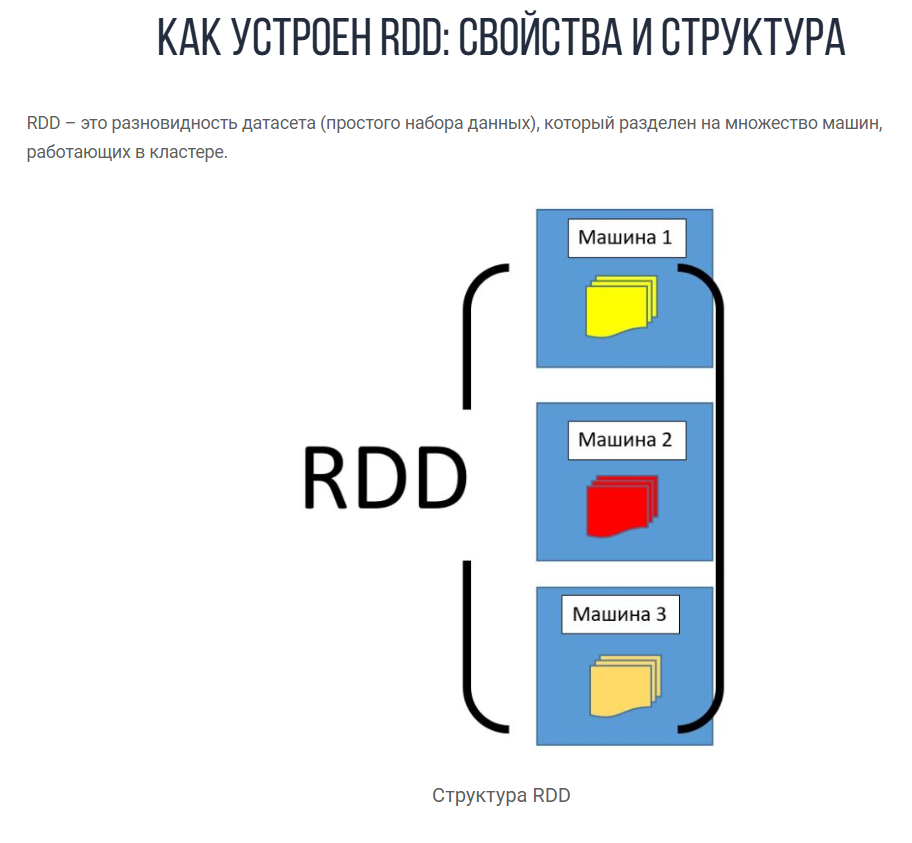
Так что же такое RDD?

RDD - это «отказоустойчивая коллекция элементов, разделенных по узлам кластера, с которыми можно работать параллельно»

**Отказоустойчивость**: данные копируются и могут быть восстановлены в случае сбоя.  
**Разбиение или распределение**: наборы данных распределяются между узлами кластера  
**Параллельная работа**: задачи выполняются одновременно для всех фрагментов данных.



RDD имеет следующие свойства:

1. Неизменяемость и секционирование – РДД состоит из набора записей, которые делятся на разделы. Раздел — это единица параллелизма в РДД. Каждый раздел является логическим подразделением данных, которое является неизменяемым (immutable) и хранится на отдельном узле в кластере.
2. Применение общих операций (coarse-grained operations), которые способны манипулировать всеми данными одновременно (например, фильтр или группировка).
3. Отказоустойчивость: все преобразования над наборами РДД ведутся в распределенной среде с поддержкой репликации (копированием данных между узлами), и каждое преобразование регистрируется каждым отдельным узлом в кластере. Следовательно, при выходе из строя одного узла, данные можно будет восстановить с помощью любого другого рабочего узла.
4. Ленивые вычисления: Apache Spark проводит необходимые преобразования над РДД только один раз в момент их создания. Это значительно сокращает общее время выполнения всех операций и ускоряет работу над данными.
5. Сохраняемость: пользователи могут выбирать удобный для себя формат хранения РДД (например, в памяти или на диске в файле).

RDD можно создавать вручную, а можно загружать из внешних источников. Источниками хранения РДД могут служить следующие источники:

* текстовый файл;
* CSV-файл;
* файл со структурой JSON-документа;
* база данных (через драйвер JDBC) **[1]**.

Преобразования (Transformations) — это операции, которые вы применяете к данным в RDD (Resilient Distributed Dataset), чтобы создать новый набор данных. Они "ленивые", что означает, что изменения не происходят сразу. Вместо этого они просто запоминают, что вы хотите сделать, и создают новый RDD, который можно использовать позже. Примеры:

* map: изменяет каждое значение в наборе данных.
* filter: оставляет только те значения, которые соответствуют определенному условию. map, filter, flatMap, groupBy, sortBy

Действия (Actions) — это операции, которые фактически выполняют вычисления на RDD и возвращают результаты. Когда вы вызываете действия, все преобразования, которые были запомнены, выполняются. Примеры:

* collect: получает все данные из RDD и возвращает их в программу.
* count: считает количество элементов в RDD.
* saveAsTextFile: сохраняет данные RDD в файл.collect, count, reduce, saveAsTextFile, foreach

В общем, преобразования — это подготовка данных, а действия — это их использование

* **Преобразования (Transformations)** считаются ленивыми вычислениями. Они создают новый набор данных на основе старого, но вычисления не происходят сразу.
* **Действия (Actions)** не являются ленивыми вычислениями. Когда вы вызываете действие, Spark выполняет все запомненные преобразования и возвращает результат.

Таким образом, преобразования ожидают выполнения, а действия инициируют его.