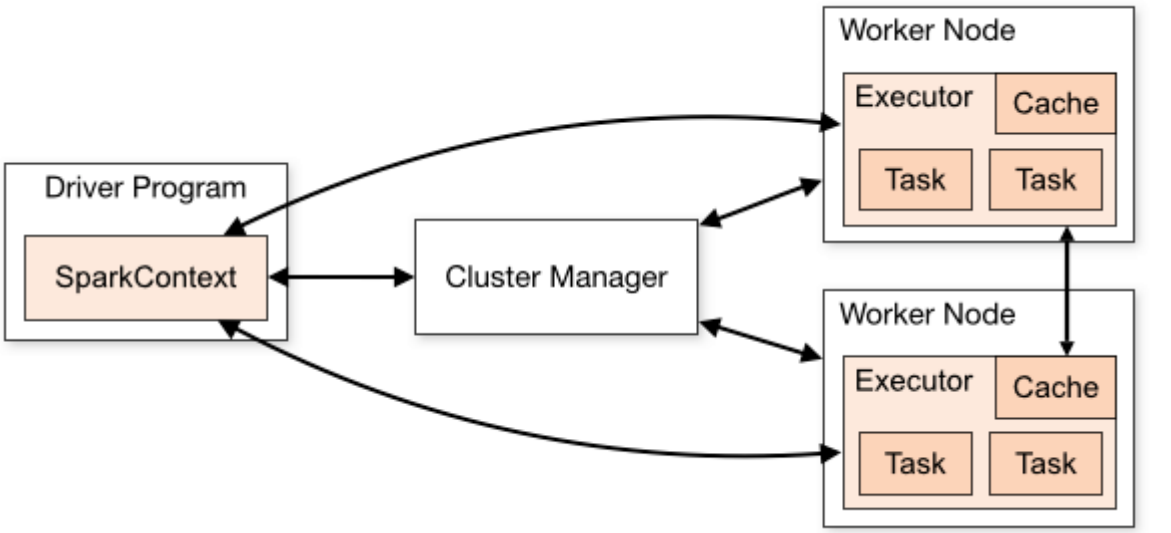
ChubaoFS Spark Shuffle

Spark 简介

Spark 是一种是基于内存计算的大数据并行计算框架，主要分为 Driver、Worker 两个组件，可通过 yarn，mesos、k8s 进行调度。其主要架构如下：

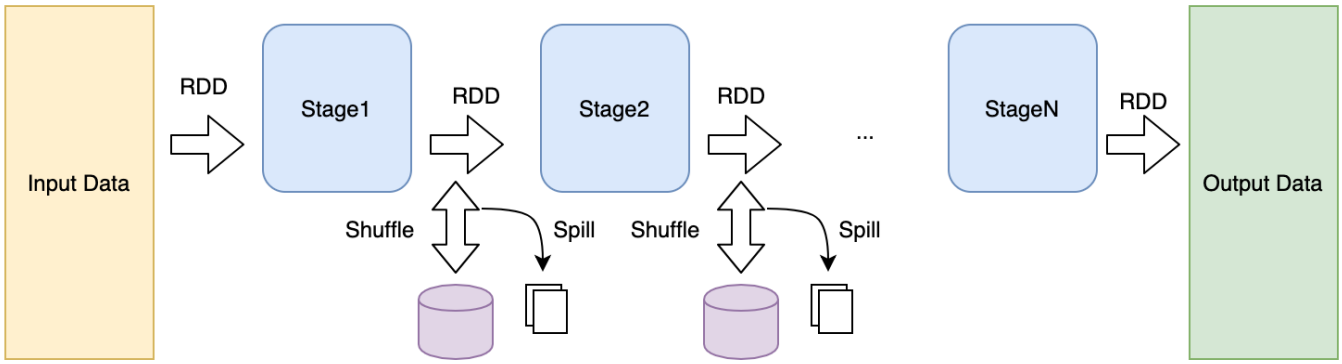


主要组件

- client：负责提交task；
- Driver Node：驱动节点，负责接受client任务提交，并分派任务，监控任务执行；
- Cluster Manager：集群管理器，yarn、mesos、k8s；
- Worker Node：工作节点，负责task的具体执行；

Spark 计算模型

Spark 将数据抽象为 `RDD` (弹性数据集)，并根据数据的依赖关系将RDD计算过程划分为一个个 `Stage`，RDD随着计算过程在各个Stage间流动。



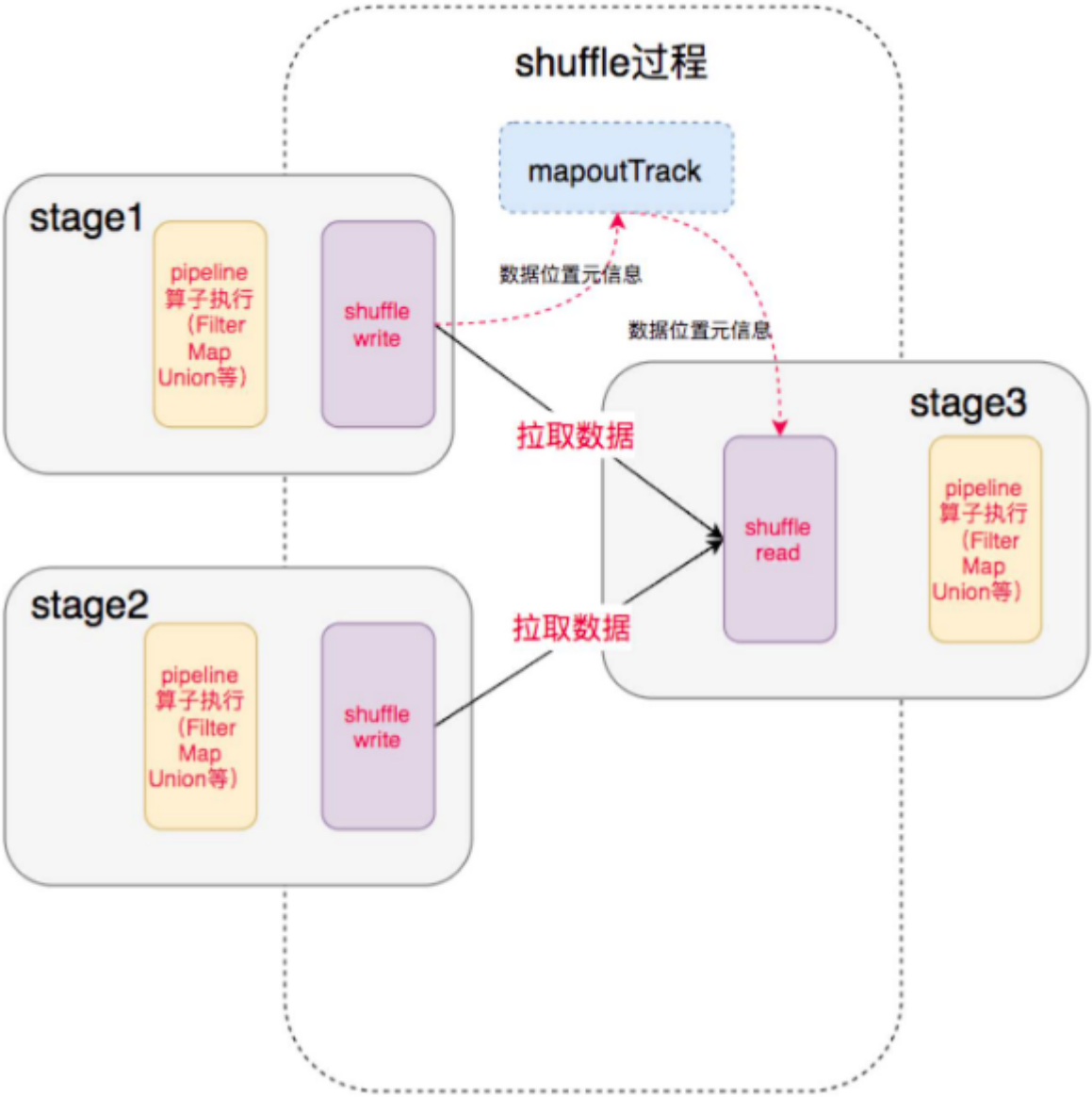
在计算过程中需要处理大量数据，其涉及的 IO 主要包括以下几个：

- **输入**：任务开始从外部数据源读取以构建输入 RDD，支持多种实现接口：hdfs, file, s3 等；
- **输出**：计算任务结束后的数据输出，写入到外部存储；
- **Shuffle**：两个 stage 之间的数据操作，包括可分为 shuffle write，shuffle read 两个阶段；
- **Spill**：Shuffle 过程中有些操作需要大量的内存，为避免 jvm 的 oom，需要将缓存数据临时存入磁盘中，这个过程称为 spill；

Spark Shuffle

Spark 2 个 Stage间需要对所有中间数据进行重排，这个过程称为`Shuffle`。Shuffle 过程需要操作大量的数据，无法全部在内存中完成，因此数据需要进行存储到磁盘中。Shuffle过程分为 Shuffle Write 和 Shuffle Read两个阶段。

Shuffle Write将上一个 stage 的 输出数据写入磁盘中，并且把数据位置元信息上报到 driver 的中， Shuffle Read在下一个 stage 开始，根据数据位置元信息，拉取对应的数据作为该stage的输入。

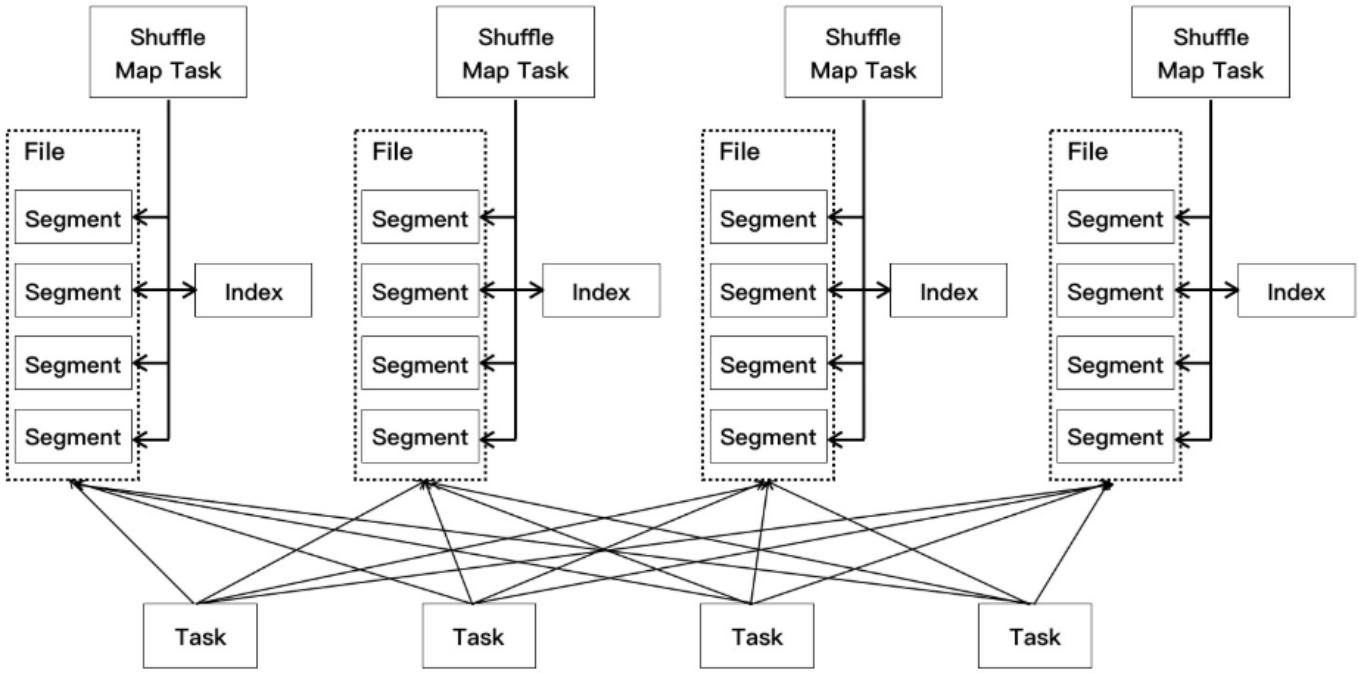


Shuffle Writer

spark 2.x中 shuffle write 有 3 种具体的实现：

- `BypassMergeSortShuffleWriter`：不需要进行Map端合并，并且RDD对应的依赖中的Partition的数量<=配置参数（默认200）。这种writer会直接将数据通过hash将数据分区写入一个个临时文件中，最后合并处理，最终形成一个dataFile和indexFile；
- `SortShuffleWriter`：Serializer支持relocation操作且允许对将要输出数据对象进行排序。sortshufflewriter在将数据写入磁盘前，会根据key对的数据进行外部排序，最终形成的文件；

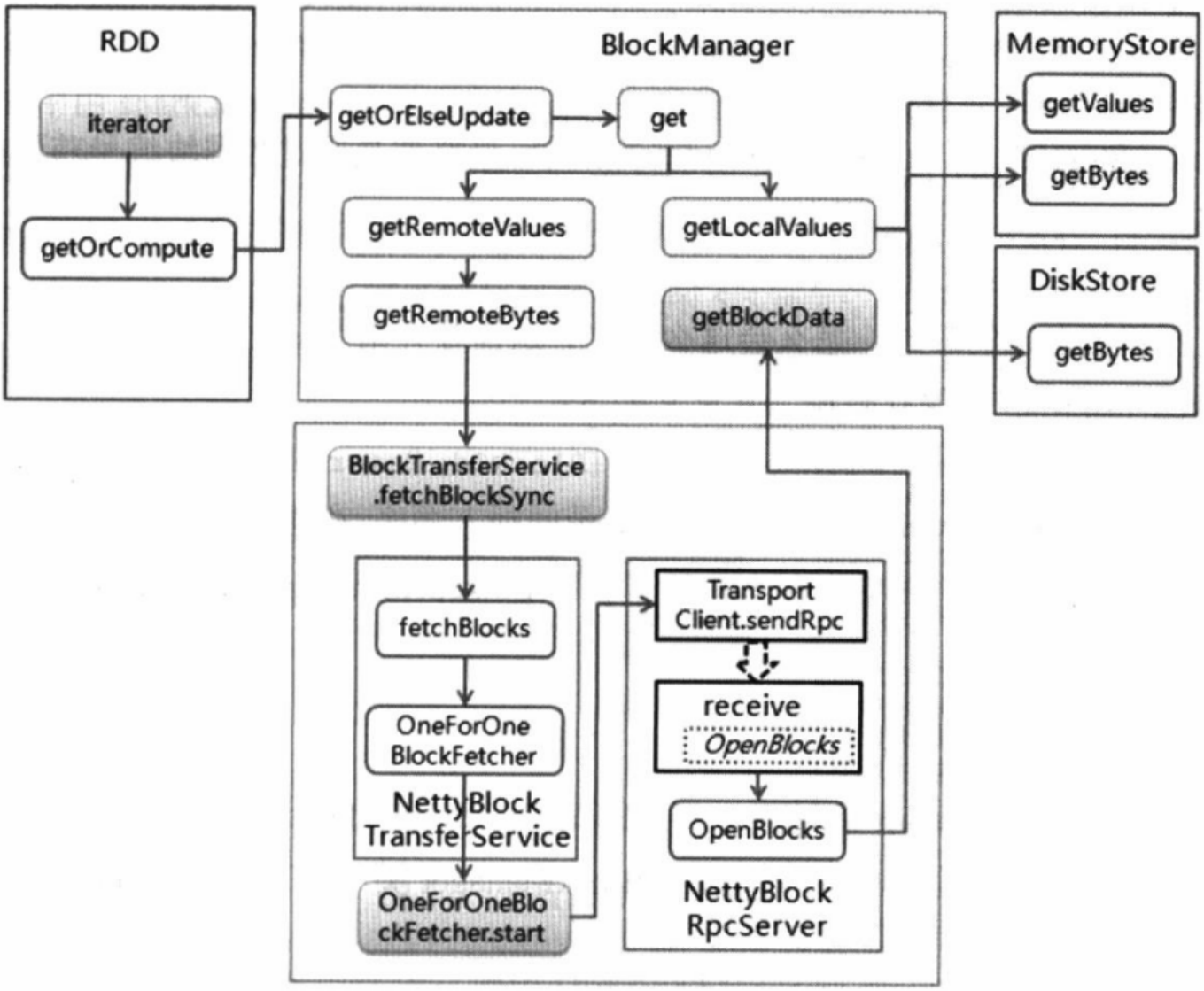
- `UnsafeShuffleWriter`：UnsafeShuffleWriter 将数据序列化后插入sorter，然后对已经序列化的record进行排序，并在排序完成后写入磁盘文件作为spill file，再将多个spill file合并成一个输出文件。



Shuffle Read

Shuffle Read 主要分为 fetch 和 aggreation 两个步骤：

- fetch：将shuffle write的各个分散数据拉取到本地；
- aggregate：将拉取的数据根据分区ID聚合，最终得到该分区相关输入RDD；



现有Shuffle痛点

1. shuffle数据有各个节点自行管理，计算和存储紧密结合，不便动态调度；
2. shuffle过程数据膨胀厉害（平均膨胀4倍）；
3. shuffle数据量巨大，io瓶颈；

Shuffle 改进方案

- [SPARK-1529]([SPARK-1529] Support DFS based shuffle in addition to Netty shuffle - ASF JIRA)：一种提出使用hdfs进行shuffle落盘的提案。未实现（有人测试过，性能约为15%）。

- SPARK-25299: 讨论了现有 external shuffle service 的不足，提出了几种使用远程存储的改进方案，未实现；
- 搜狗Alluxio for Shuffle： 搜狗使用 alluxio 来存储 shuffle 数据，性能提升非常明显(没有具体的数据)；
- Facebook的SOS： 将大量小的 shuffle 读请求转换成少并且大的顺序 I/O 请求。作业整体的 I/O 提升了两倍，计算效率提高10%。
- 阿里Smart shuffle：shuffle数据在map端累积到一定数量发送到reduce端, shuffle数据的生成和shuffle数据的发送可以并行执行。
- crail-spark-io：使用RDMA网络来改进shuffle过程中的网络IO瓶颈；

Chubao Spark Shuffle

ChubaoFS Spark Shuffle 是基于ChubaoFS的spark shuffle插件，依托ChubaoFS的分布式、高可靠、高性能存储为Spark Shuffle 提供可靠的共享存储接口，以实现spark的存储与计算分离。

目标：

- 依靠chubaofs，实现spark的存储与计算分离；
- 插件化，兼容spark原有架构，方便升级更新；
- 优化shuffle 存储层读写流程，提升shuffle性能；

基本方案


Driver
Spark Client
MapOutputTracerMaster
BlockManagerMaster
ChubaoFS Shuffle
worker1
worker2
MapOutputTracerWorker
BlockManagerWorker
executor i
ChubaoFS Shuffle
map
reduce
MapOutputTracerWorker
BlockManagerWorker
executor k
ChubaoFS Shuffle
write
read
ChubaoFS-Server

相关类

主要接口

- `ChubaoShuffleManager`：继承自`ShuffleManger`，实现ShuffleManager插件管理相关接口；
    - registerShuffle/unregisterShuffle: 根据参数注册不同类型的shuffle handle，以
    - getWriter：获取具体的ShuffleWriter；
    - getReader: 获取具体的ShuffleReader；
    - stop: shuffle模块停止时的收尾处理；

- `ChubaoBypassSortShuffleHandle` / `ChubaoSerializedShuffleHandle`: 分别对应不同的情形下的shuffle操作handle；
- `ChubaoShuffleWriter`： shuffle writer的具体实现
    - write：根据需要对shuffle map out 数据进行排序分组，写入到chubaofs中，并将数据块路径及索引上报给`MapOutTracerMaster`；
    - stop：writer结束后的处理流程；
- `ChubaoShuffleReader`:
    - read：通过`MapOutTracer`获取对应executor的数据块迭代器；
- `ChubaoShuffleBlockResolve`： chubao shuffle 数据块组织
- `ChubaoSorter` / `UnSafeSorter`: 不同情形下的排序操作；
    - spill：排序过程中的数据溢出操作；
    - insertAll: 具体插入排序实现；
- `ChubaoBufferedOutputStream`: shuffle操作过程缓冲数据流实现；
- `ChubaoSpillInfo`： spill数据组织信息；