Spark概述

Spark Overview

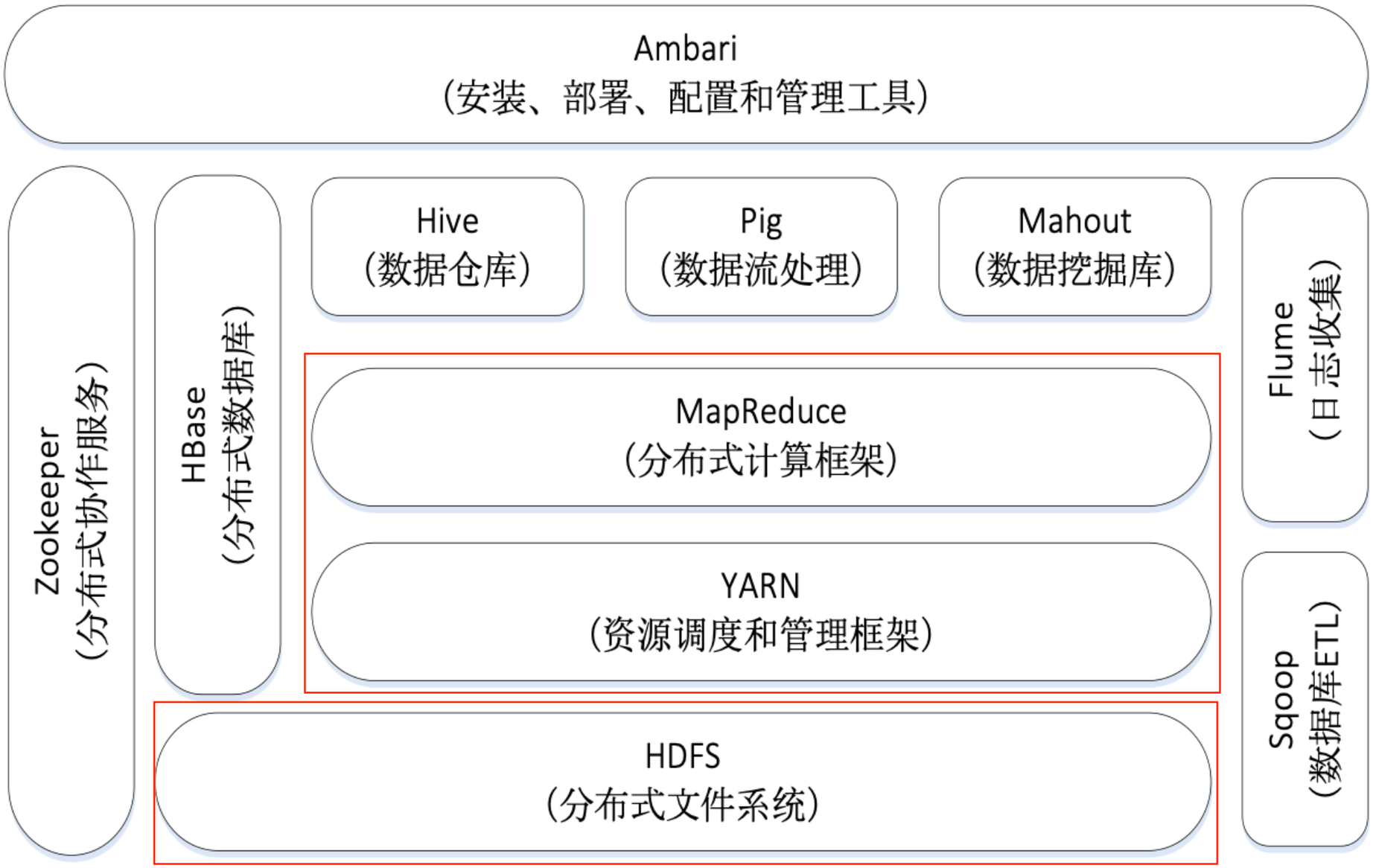
大数据时代

- 存储空间越来越廉价
- 网络带宽越来越大
- 几乎每个人每个人每天都会产生数据

关键技术

- 如何存储
  - 分布式存储（HDFS、HBase…）
- 如何高性能的计算
  - 分布式计算（MapReduce）

代表性技术-Hadoop



Hadoop-MapReduce

- MapReduce将复杂的、运行于大规模集群上的并行计算过程高度地抽象到了两个函数：Map和Reduce

- 不需要掌握分布式并行编程细节，也可以很容易把自己的程序运行在分布式系统上，完成海量数据的计算。但是，即使很简单的任务也需要编写MapReduce。

- MapReduce采用“分而治之”策略，一个存储在分布式文件系统中的大规模数据集，会被切分成许多独立的分片（split），这些分片可以被多个Map任务并行处理

Hadoop-Yarn

- 一套系统当中同时存在各种不同的业务应用场景，需要采用不同的计算框架
    - MapReduce实现离线批处理
    - 使用Impala实现实时交互式查询分析
    - 使用Storm实现流式数据实时分析
    - 使用Spark实现迭代计算
- 这些产品通常来自不同的开发团队，具有各自的资源调度管理机制，为了避免不同类型应用之间互相干扰，需要把内部的服务器拆分成多个集群，分别安装运行不同的计算框架，即“一个框架一个集群”
- 导致问题
    - 集群资源利用率低
    - 学习成本高
    - 维护代价高

Hadoop-Yarn

- YARN的目标就是实现“一个集群多个框架”，即在一个集群上部署一个统一的资源调度管理框架YARN，在YARN之上可以部署其他各种计算框架

- 由YARN为这些计算框架提供统一的资源调度管理服务，并且能够根据各种计算框架的负载需求，调整各自占用的资源，实现集群资源共享和资源弹性收缩

- 可以实现一个集群上的不同应用负载混搭，有效提高了集群的利用率

- 不同计算框架可以共享底层存储，避免了数据集跨集群移动

Hadoop-HDFS

- HDFS （Hadoop Distributed File System）

- Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。

- HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。

代表性技术-Spark

Lightning-fast unified analytics engine

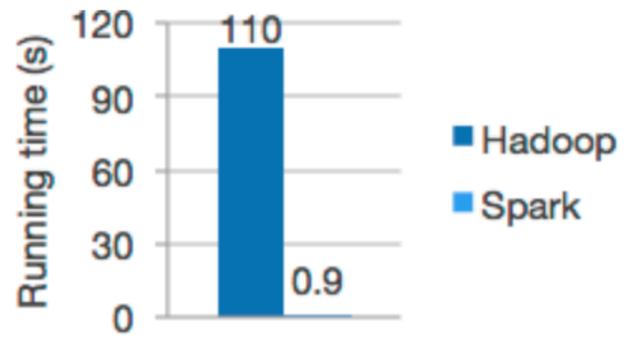
Download Libraries Documentation Examples Community Developers

Apache Spark™ is a unified analytics engine for large-scale data processing.

Speed

Run workloads 100x faster.

Apache Spark achieves high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.



Logistic regression in Hadoop and Spark

Spark

- Spark最初由美国加州伯克利大学（UCBerkeley）的AMP（Algorithms, Machines and People）实验室于2009年开发，是基于内存计算的大数据并行计算框架，可用于构建大型的、低延迟的数据分析应用程序。

- 2014年打破了Hadoop保持的基准排序（Sort Benchmark）纪录，使用206个节点在23分钟的时间里完成了100TB数据的排序，而Hadoop则是使用2000个节点在72分钟的时间里完成同样数据的排序。也就是说，Spark仅使用了十分之一的计算资源，获得了比Hadoop快3倍的速度。

Spark特点

- 运行速度快
    - Spark使用先进的DAG（Directed Acyclic Graph，有向无环图）执行引擎，以支持循环数据流与内存计算，基于内存的执行速度可比Hadoop MapReduce快上百倍，基于磁盘的执行速度也能快十倍。
- 容易使用
    - Spark支持使用Scala、Java、Python和R语言进行编程，简洁的API设计有助于用户轻松构建并行程序，并且可以通过Spark Shell进行交互式编程。
- 通用性
    - Spark提供了完整而强大的技术栈，包括SQL查询、流式计算、机器学习和图算法组件，这些组件可以无缝整合在同一个应用中，足以应对复杂的计算。
- 运行模式多样
    - Spark可运行于独立的集群模式中，或者运行于Hadoop中，也可运行于Amazon EC2等云环境中，并且可以访问HDFS、Cassandra、HBase、Hive等多种数据源。
- Spark源码托管在Github中，截至2019年6月，共有超过1392名来自不同公司的开发人员贡献了24586次代码提交，可见Spark的受欢迎程度。

Spark相比于Hadoop

- Hadoop虽然已成为大数据技术的事实标准，但其本身还存在诸多缺陷，最主要的缺陷是其MapReduce计算模型延迟过高，无法胜任实时、快速计算的需求，因而只适用于离线批处理的应用场景。

- Hadoop的缺陷
  - 表达能力有限。计算都必须要转化成Map和Reduce两个操作，但这并不适合所有的情况，难以描述复杂的数据处理过程通用性
  - 磁盘IO开销大。每次执行时都需要从磁盘读取数据，并且在计算完成后需要将中间结果写入到磁盘中，IO开销较大。
  - 延迟高。一次计算可能需要分解成一系列按顺序执行的MapReduce任务，任务之间的衔接由于涉及到IO开销，会产生较高延迟。而且，在前一个任务执行完成之前，其他任务无法开始，难以胜任复杂、多阶段的计算任务。

Spark相比于Hadoop

- Spark在借鉴Hadoop MapReduce优点的同时，很好地解决了MapReduce所面临的问题。相比于MapReduce，Spark主要具有如下优点

  - Spark的计算模式也属于MapReduce，但不局限于Map和Reduce操作，还提供了多种数据集操作类型，编程模型比MapReduce更灵活

  - Spark提供了内存计算，中间结果直接放到内存中，带来了更高的迭代运算效率

  - Spark基于DAG的任务调度执行机制，要优于MapReduce的迭代执行机制

Spark相比于Hadoop

- Spark最大的特点就是将计算数据、中间结果都存储在内存中，大大减少了IO开销。
    - Spark更适合于迭代运算比较多的机器学习运算。
        - 使用Hadoop进行迭代计算非常耗资源，因为每次迭代都需要从磁盘中写入、读取中间数据，IO开销大。而Spark将数据载入内存后，之后的迭代计算都可以直接使用内存中的中间结果作运算，避免了从磁盘中频繁读取数据。
- 使用Hadoop需要编写不少相对底层的代码，而Spark提供了多种高层次、简洁的API。通常情况下，Spark的代码量要比Hadoop少2-5倍。更重要的是，Spark提供了实时交互式编程反馈，可以方便地验证、调整算法。

Spark相比于Hadoop

- Spark 并不能取代Hadoop，只是替代了Hadoop中的MapReduce。Spark已经很好的融入到了Hadoop的生态系统。

- 可以使用Yarn管理Spark集群

- 可以读写HDFS、HBase中的数据

Spark生态系统

- 一套大数据系统中一般主要包括以下三个类型：

  - 复杂的批量数据处理：时间跨度通常在数十分钟到数小时之间；

  - 基于历史数据的交互式查询：时间跨度通常在数十秒到数分钟之间；

  - 基于实时数据流的数据处理：时间跨度通常在数百毫秒到数秒之间。

Spark生态系统

- 一套大数据系统中一般主要包括以下三个类型：
  - 复杂的批量数据处理：时间跨度通常在数十分钟到数小时之间。
    - Hadoop MapReduce
  - 基于历史数据的交互式查询：时间跨度通常在数十秒到数分钟之间
    - Hive、Druid、Impala等
  - 基于实时数据流的数据处理：时间跨度通常在数百毫秒到数秒之间
    - Storm

Spark生态系统

- 对于企业来说，可能只需要解决上述的一个或者两个问题
- 但对于互联网来说，往往是需要解决上述的所有问题
  - 不同场景之间输入输出数据无法做到无缝共享，通常需要进行数据格式的转换
  - 不同的软件需要不同的开发和维护团队，带来了较高的使用成本
  - 比较难以对同一个集群中的各个系统进行统一的资源协调和分配。

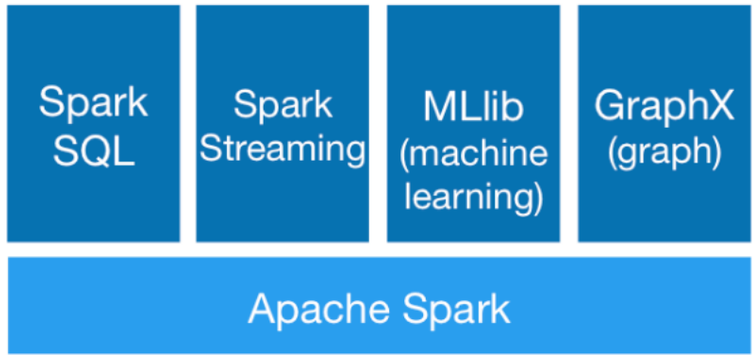
Spark生态系统

- Spark的设计遵循“一个软件栈满足不同应用场景”的理念，逐渐形成了一套完整的生态系统，既能够提供内存计算框架，也可以支持SQL即席查询、实时流式计算、机器学习和图计算等。

Generality

Combine SQL, streaming, and complex analytics.

Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming. You can combine these libraries seamlessly in the same application.



Spark生态系统

- Spark Core
    - Spark Core包含Spark的基本功能，如内存计算、任务调度、部署模式、故障恢复、存储管理等。Spark建立在统一的抽象RDD之上，使其可以以基本一致的方式应对不同的大数据处理场景
- Spark SQL
    - Spark SQL允许开发人员直接处理RDD，同时也可查询Hive、HBase等外部数据源。Spark SQL的一个重要特点是其能够统一处理关系表和RDD，使得开发人员可以轻松地使用SQL命令进行查询，并进行更复杂的数据分析
- Spark Streaming
    - Spark Streaming支持高吞吐量、可容错处理的实时流数据处理，其核心思路是将流式计算分解成一系列短小的批处理作业。Spark Streaming支持多种数据输入源，如Kafka、Flume和TCP套接字等
- MLlib（机器学习）
    - MLlib提供了常用机器学习算法的实现
- GraphX（图计算）
    - GraphX是Spark中用于图计算的API，可认为是Pregel在Spark上的重写及优化，Graphx性能良好，拥有丰富的功能和运算符，能在海量数据上自如地运行复杂的图算法。