大数据处理技术：Hadoop与Spark实战

01
Hadoop概述及安装配置

Hadoop的基本概念和原理

- **Hadoop**是一种基于**分布式**架构的**大数据处理**框架
  - 通过对**海量数据**进行**分布式存储**和**并行处理**，实现对**大规模数据集**的高效处理
  - **高可用性**、**可扩展性**和**容错性**是Hadoop的核心特性
- **Hadoop**的核心组件包括**Hadoop Common**、**Hadoop MapReduce**和**Hadoop YARN**
  - **Hadoop Common**提供了一系列基础组件，如**文件系统**、**RPC框架**和**Hadoop分布式文件系统(HDFS)**
  - **Hadoop MapReduce**提供了一套编程模型，用于处理大规模数据集
  - **Hadoop YARN**是一种**资源管理系统**，负责为Hadoop应用分配和管理资源

Hadoop的核心组件介绍

Hadoop Common

- **Hadoop分布式文件系统(HDFS)**：用于**存储**和**访问**大量**数据**的分布式文件系统
- **Hadoop YARN**：负责**资源管理**和**任务调度**的框架
- **Hadoop MapReduce**：提供了一套并行处理数据集的编程模型

Hadoop MapReduce

- **MapReduce**是一种编程模型，用于**处理**大规模**数据**集
- **MapReduce**程序由**Map**和**Reduce**两个阶段组成
- **MapReduce**通过**分片**和**并行化**的方式，实现对**大规模数据集**的高效处理

Hadoop YARN

- **YARN**是一种**资源管理系统**，负责为Hadoop应用分配和管理**资源**
- **YARN**通过**资源调度**和**作业管理**的方式，实现对**Hadoop集群**的高效管理

Hadoop的安装配置及环境搭建

- **硬件**和**软件**要求
  - **硬件**要求：较高的计算能力和**存储**容量
  - **软件**要求：支持Java的**操作系统**，如**CentOS**或**Ubuntu**
- **安装包下载**
  - 从**官方网站**或**第三方镜像站点**下载**Hadoop安装包**
  - 选择合适的**版本**和**配置**进行安装
- **环境配置**
  - 配置**Java环境**和**SSH**
  - 配置**Hadoop环境变量**，如HADOOP_HOME和JAVA_HOME
  - 修改conf/hadoop-env.sh配置文件，设置YARN_HOME和JAVA_HOME
- **集群安装**
  - 配置**Hadoop集群**，包括**主节点**和**从节点**
  - 配置**Hadoop集群**的**HDFS**和**YARN**服务
  - 测试**Hadoop集群**的功能和性能

03 Hadoop分布式文件系统(HDFS)

HDFS的基本概念和功能

HDFS的主要功能

- **数据存储**：将**大数据**划分为多个**数据块**，分布式存储在多个计算节点上
- **数据访问**：提供**简单的文件操作接口**，如put、get和rm
- **数据复制**：实现**数据冗余**，提高数据**可靠性**和**可用性**
- **数据平衡**：自动调整数据分布，平衡计算节点的负载

HDFS是一种分布式文件系统，用于存储和管理大规模数据集

- **高可用性**、**可扩展性**和**容错性**是HDFS的核心特性
- HDFS适用于**批量数据处理**和**离线分析**场景

HDFS的数据存储和访问机制

数据分片：将大数据划分为多个较小的数据块，存储在多个计算节点上

- 减少**单点故障**和**数据丢失**的风险
- 提高**数据访问**和**处理**的速度

数据副本：为每个数据块创建多个副本，存储在不同的计算节点上

- 提高**数据可靠性**和**可用性**
- 平衡计算节点的负载，提高**集群性能**

数据存储和访问策略

- 数据块大小：默认值为**64MB**，可根据实际需求进行调整
- 复制因子：默认值为**3**，可根据实际需求进行调整
- 数据访问路径：通过**namenode**和**datanode**进行数据的访问和操作

HDFS的操作命令和文件管理

命令行工具：使用hadoop命令行工具进行HDFS的操作和管理

- hadoop fs：文件系统操作命令，如ls、mkdir、put和get
- hadoop dfsadmin：集群管理命令，如status、report和balance
- hadoop jar：运行Hadoop MapReduce程序

文件管理

- 文件创建：使用hadoop fs -put命令将本地文件上传到HDFS
- 文件删除：使用hadoop fs -rm命令删除HDFS上的文件
- 文件查看：使用hadoop fs -ls命令查看HDFS上的文件和目录信息

03 Hadoop MapReduce编程模型

MapReduce的基本概念和任务划分

MapReduce是一种编程模型，用于处理大规模数据集

- 通过**MapReduce**程序，实现了**分布式计算**和**并行处理**的功能
- 适用于**批量数据处理**、**数据分析**和**机器学习**等场景

MapReduce的任务划分

- **Map任务**：处理**输入数据**，生成**中间数据**
- **Reduce任务**：对**中间数据**进行**聚合**和**汇总**，生成**最终结果**

MapReduce的输入和输出

- 输入：可以是**本地文件**、**HDFS文件**或其他**数据源**
- 输出：通常存储在**HDFS**上，以便于**后续处理**和**分析**

MapReduce的工作流程和编程实例

MapReduce的工作流程

- **Mapper阶段**：处理**输入数据**，生成**中间数据**
- **Shuffle阶段**：对**中间数据**进行**分区**、**排序**和**分组**
- **Reducer阶段**：对**中间数据**进行**聚合**和**汇总**，生成**最终结果**

编程实例：计算单词频率

- 使用Hadoop Streaming或Hadoop Java API编写MapReduce程序
- 使用hadoop jar命令运行程序，产生**单词频率统计表**

MapReduce的优化策略和性能调优

- **优化策略**
  - **合理划分数据**：根据计算节点的**计算能力和负载**，合理划分数据
  - **选择合适的输入格式**：使用**SequenceFile**或**Avro**等输入格式，提高数据读取速度
  - **避免磁盘I/O瓶颈**：将数据存储在**内存**或**SSD**等高性能存储设备上
  - **使用合理的Reducer数量**：根据数据量和计算需求，选择合适的Reducer数量
- **性能调优**
  - **调整内存分配**：使用mapreduce.map.memory.mb和mapreduce.reduce.memory.mb配置项调整内存分配
  - **调整并行度**：使用mapreduce.map.parallelism和mapreduce.reduce.parallelism配置项调整并行度
  - **使用Combiner**：使用Combiner减少数据传输量和Reducer处理时间
  - **禁用不必要的操作**：禁用不必要的Shuffle阶段操作，如secondary sort和map output compaction

04
Spark概述及安装配置

Spark的基本概念和原理

Spark的核心组件

- **Spark Core**：提供基础功能，如**任务调度**、**内存管理**和**通信协议**
- **Spark SQL**：支持**结构化数据**的查询和分析
- **Spark Streaming**：支持**实时数据流**的处理和分析
- **MLlib**：提供**机器学习**算法的库和接口

Spark是一种内存计算框架，用于处理大规模数据集

- 通过对数据进行**缓存**和**迭代计算**，实现**快速处理**和**实时分析**的功能
- 适用于**实时数据处理**、**流计算**、**机器学习和图计算**等场景