Spark架构原理

www.huawei.com

HUAWEI

前言

- 本章主要对**Spark**组件的应用场景，功能和架构以及在**FusionInsight**平台中的使用等进行简单介绍

目标

- 学完本课程后，您将能够：
  - 理解**Spark**应用场景，了解**Spark**特点
  - 了解**Spark**计算能力及其技术架构
  - 了解**Spark**组件在**FusionInsight** 平台中的使用

目录

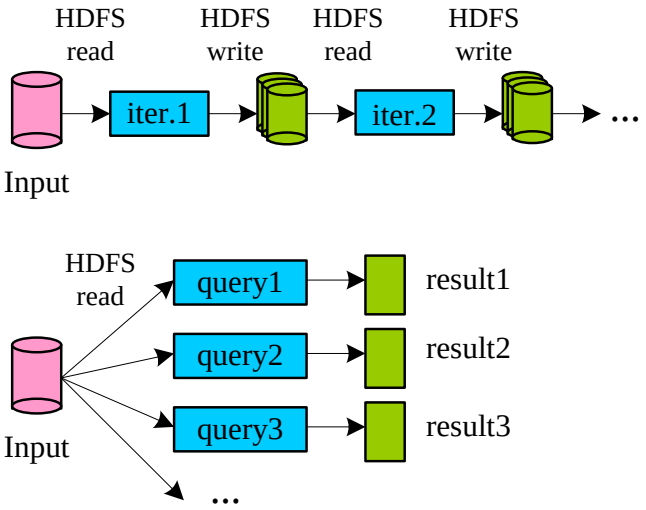
HUAWEI

Spark适用场景

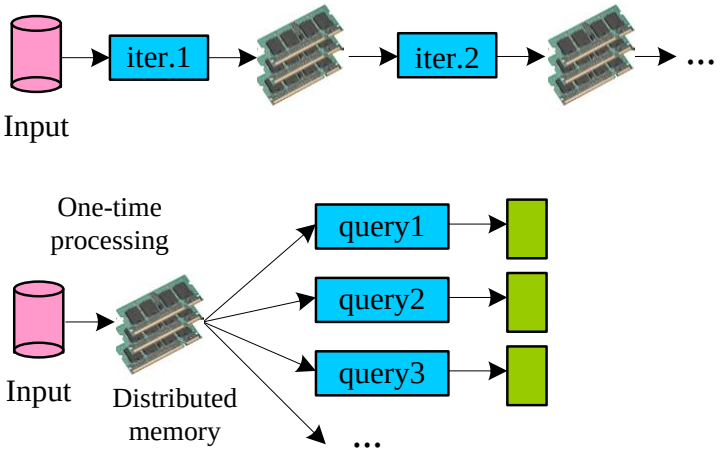
- 是什么
  - **Spark**系统是**分布式批处理系统**和**分析挖掘引擎**
  - **Spark** 是**AMP LAB**贡献到**Apache**社区的开源项目，是**AMP**大数据栈的基础组件
- 做什么
  - 数据处理**(Data Processing)**：可以用来快速处理数据，兼具容错性和可扩展性
  - 迭代计算**(Iterative Computation)**：支持迭代计算，有效应对多步的数据处理逻辑
  - 数据挖掘**(Data Mining)**：在海量数据基础上进行复杂的挖掘分析，可支持各种数据挖掘和机器学习算法

Spark适用场景

- 大多数现有集群计算框架如**Hadoop**等基于从稳定存储（文件系统）到稳定存储的非循环数据流---应对数据集重用型应用时低效，与传统的**MR**任务的频繁读写磁盘数据相比，基于内存计算的Spark则更适合应用与迭代计算，交互式分析等场景



Data Sharing in MapReduce



Data Sharing in Spark

HUAWEI

Spark特点

- 轻：**Spark**核心代码有3万行。
  - **Scala**语言的简洁和丰富表达力
  - 巧妙利用了**Hadoop**和**Mesos**的基础设施
- 快：**Spark**对小数据集可达到亚秒级的延迟
  - 对大数据集的迭代机器学习即席查询、图计算等应用，**Spark** 版本比基于**MapReduce**、**Hive**和**Pregel**的实现快
  - 内存计算、数据本地性和传输优化、调度优化

Spark特点

- 灵：**Spark**提供了不同层面的灵活性
  - **Scala**语言**trait**动态混入策略(如可更换的集群调度器、序列化库)
  - 允许扩展新的数据算子、新的数据源、新的**language bindings**
  - **Spark**支持内存计算、多迭代批量处理、即席查询、流处理和图计算等多种范式
- 巧：巧妙借力现有大数据组件
  - **Spark**借**Hadoop**之势，与**Hadoop**无缝结合
  - 图计算借用**Pregel**和**PowerGraph**的**API**以及**PowerGraph**的点分割思想

本节总结

- 本章主要对**Spark**的产生背景和应用场景给予简单介绍，同时介绍了**spark**的特点。

目录

HUAWEI

Spark技术架构

- **Spark**架构采用了分布式计算中的**Master-Slave**模型。**Master**是对应集群中的含有**Master**进程的节点（**ClusterManager**），**Slave**是集群中含有**Worker**进程的节点。**Master**作为整个集群的控制器，负责整个集群的正常运行；**Worker**相当于是计算节点，接收主节点命令与进行状态汇报；**Executor**负责任务的执行，运行在**Worker**节点（在**FI**集群中，**Master**节点即为**Resourcemanager**节点，**Slave**节点即为**NodeManager**节点）；
- **Spark**的任务流程：**Client**作为客户端提交应用，**Master**找到一个**Worker**启动**Driver(**或者本地启动**Driver)**，**Driver**向**Master**申请资源，之后将应用转化为**RDD Graph**，再由**DAGScheduler**将**RDD Graph**转化为**Stage**后提交给**TaskScheduler**，由**TaskScheduler**提交给**Executor**执行。

Spark Application基本概念

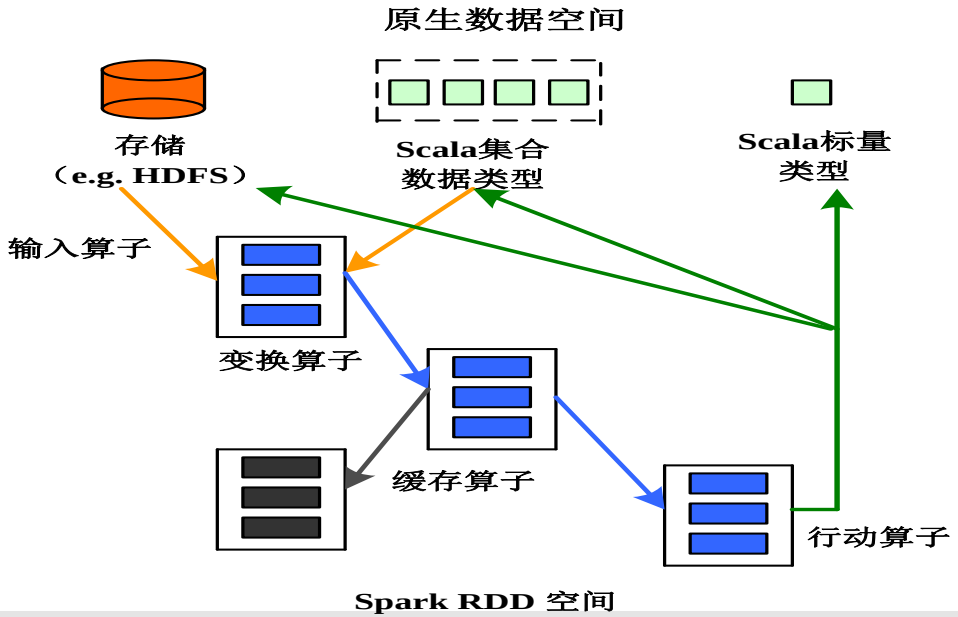
- **Application**： **Spark**用户程序，提交一次应用为一个**App**，一个App会启动一个**SparkContext**，也就是**app**的**driver**，驱动整个**App**的运行
- **Job**： 一个**App**可能包含多个**Job**，每个**action**算子对应一个**Job**；**action**算子有**collect**，**count**等。
- **Stage**： 每个**Job**可能包含多层**Stage**，划分标记为**shuffle**过程；**Stage**按照依赖关系依次执行。
- **Task**： 具体执行任务的基本单位，被发到**executor**上执行。

Spark基本概念

- **Cluster Manager**： 集群资源管理服务，通常包含主节点（主备）和多个运行节点；支持运行模式有**Standalone**模式、**on Mesos**模式、**on Yarn**模式（**FI**环境中使用**Yarn**作为**spark**任务调度的资源管理器）。
- **Driver**：运行**App**的大脑，负责**job**的初始化，将**job**转换成**task**并提交执行
- **DAGScheduler**：是一个面向**Stage**层面的调度器，把**Job**分解成**Stage** ，按照**Stage**提交**TaskSet**给**TaskScheduler**。
- **TaskScheduler**：提交**Task**给**Executor**运行，并管理执行结果。
- **BlockManager**：管理**App**运行周期的中间数据，比如存在内存、本地。
- **Executor**： 是**App**运行在work 节点上的一个进程，该进程负责运行**task** ，生命周期和**App**相同。

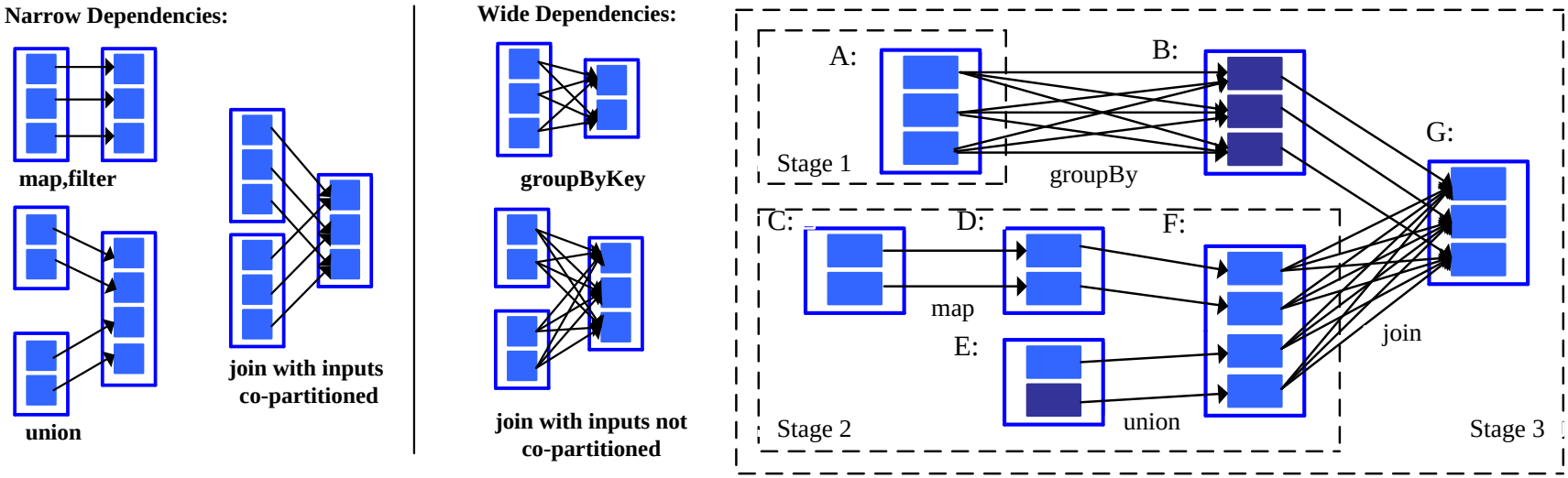
Spark核心概念 - RDD（Resilient Distributed Datasets）

- 定义：只读的，可分区的分布式数据集；数据集可全部或部分缓存在内存中，在一个**App**多次计算间重用，**RDD**是**Spark**的核心。
- 血统容错：根据血统（父子间依赖关系）重计算恢复丢失数据
- **RDD**操作： **Transformation**算子和**Action**算子。



Spark核心概念 - 宽依赖和窄依赖

RDD父子依赖关系：窄（Narrow）依赖和宽（Wide）依赖。窄依赖指父RDD的每一个分区最多被一个子RDD的分区所用。宽依赖指子RDD的分区依赖于父RDD的所有分区。



HUAWEI

Spark核心概念 - Transformation和Action

- **Transformation**是懒操作，用于定义新的 **RDD**；而**Action**启动计算操作，并向用户程序返回值或向外部存储写数据。

Transformation有如下几种：

map(f: T => U): RDD[T] => RDD [U]

filter(f: T => Bool): RDD[T] => RDD[T]

flatMap(f: T => Seq[U]): RDD[T] => RDD[U]

groupByKey(): RDD[(K, V)] => RDD[(K, Seq[V])]

reduceByKey(f: (V,V) => V): RDD[(K, V)] => RDD[(K, V)]

union(): (RDD[T],RDD[T]) => RDD[T]

join(): (RDD[(K, V)],RDD[(K, W)]) => RDD[(K, (V, W))]

mapValues(f: V => W): RDD[(K, V)] => RDD[(K, W)]

partitionBy(p: Partitioner[K]): RDD[(K, V)] => RDD[(K, V)]

Action有如下几种：

count(): RDD[T] => Long

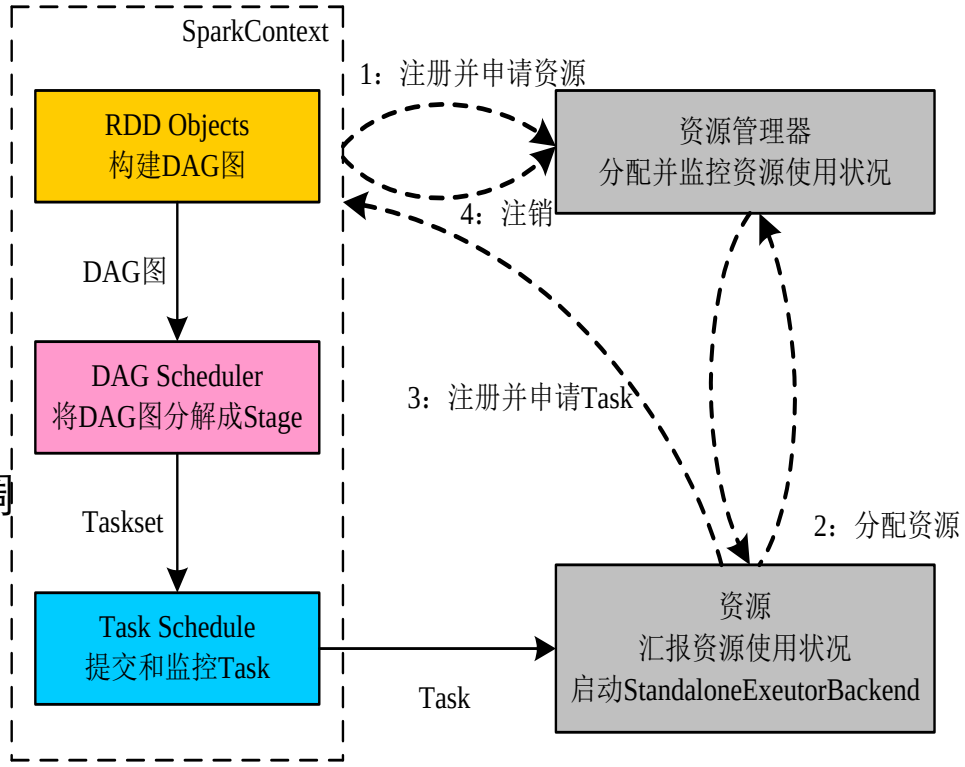
collect(): RDD[T] => Seq[T]

reduce(f: (T, T) => T): RDD[T] => T

lookup(k: K): RDD[(K, V)] => Seq[V]

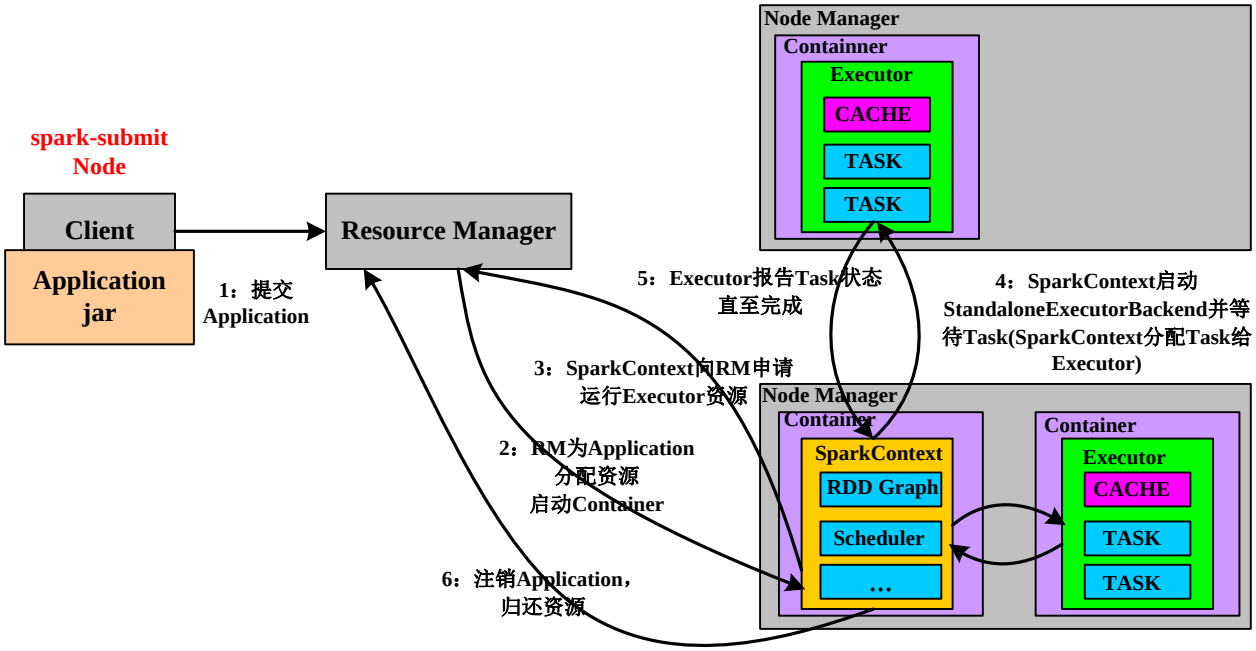
运行过程

- 构建**Spark Application**的运行环境（启动**SparkContext**）
- **SparkContext**向资源管理器**(Standalone**、**Mesos**、**Yarn)**申请运行**Executor**资源，并启动**StandaloneExecutorBackend**
- **Executor**向**SparkContext**注册
- **SparkContext**启动应用程序**DAG**调度、**Stage**划分，**TaskSet**生成
- **Task Scheduler**调度**Taskset**，将**task**发放给**Executor**运行。
- **Task**在**Executor**上运行，运行完毕释放所有资源。

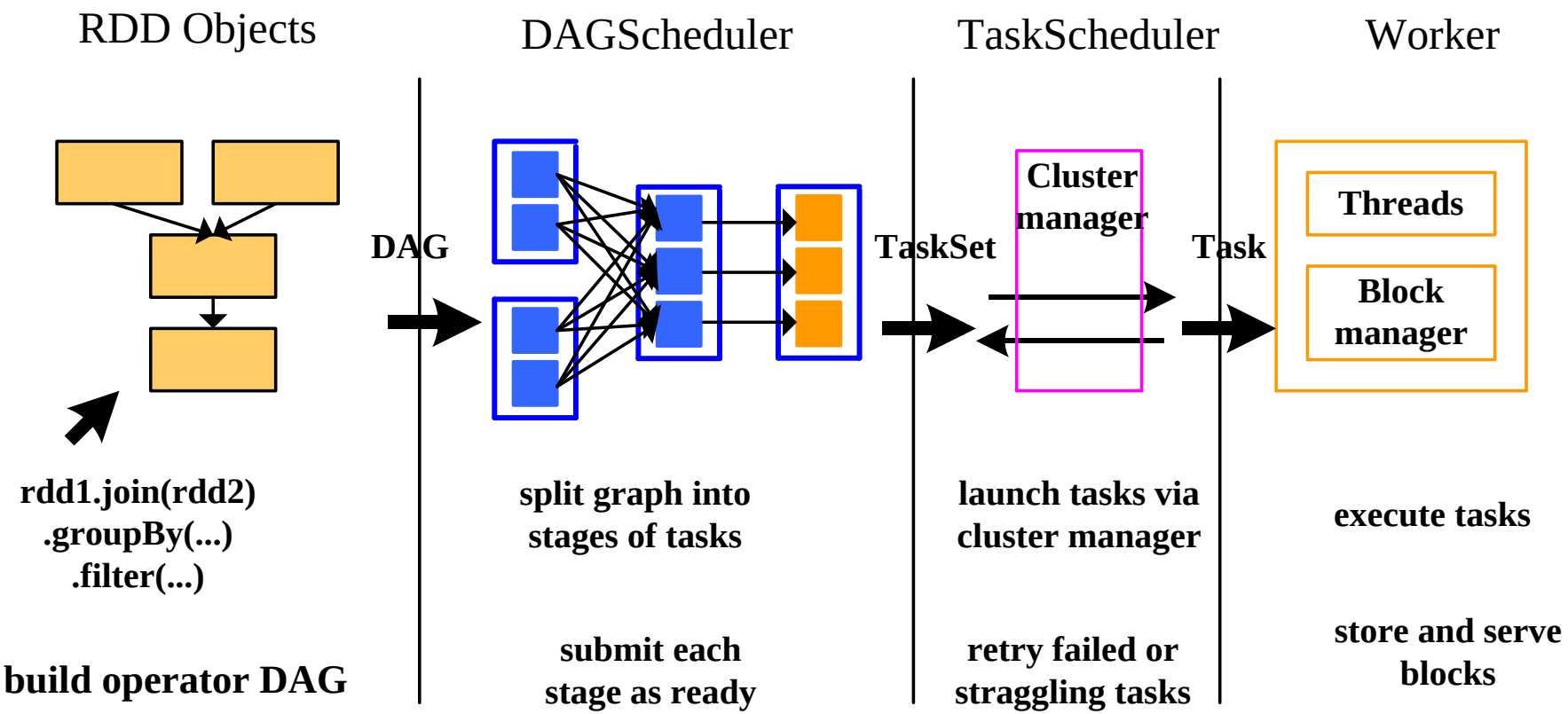


Spark On Yarn模式App运行过程

- **Cluster模式：Spark Driver**启动在集群中，**ResourceManager**在集群中分配一个**container**来启动**Spark Driver**驱动整个应用。



Spark应用调度



HUAWEI

本章总结

- 本章主要介绍了**Spark**的基本概念，技术架构，着重介绍了**Spark**任务的进程运行，**On Yarn**模式，以及**Spark** 的应用调度。

目录

1. Spark应用场景
2. Spark 基本功能和技术架构
3. **Spark组件介绍**
   - **Spark在FI界面呈现**
   - **Spark常用进程**
   - **Spark维护关键点**

Spark的WebUI呈现

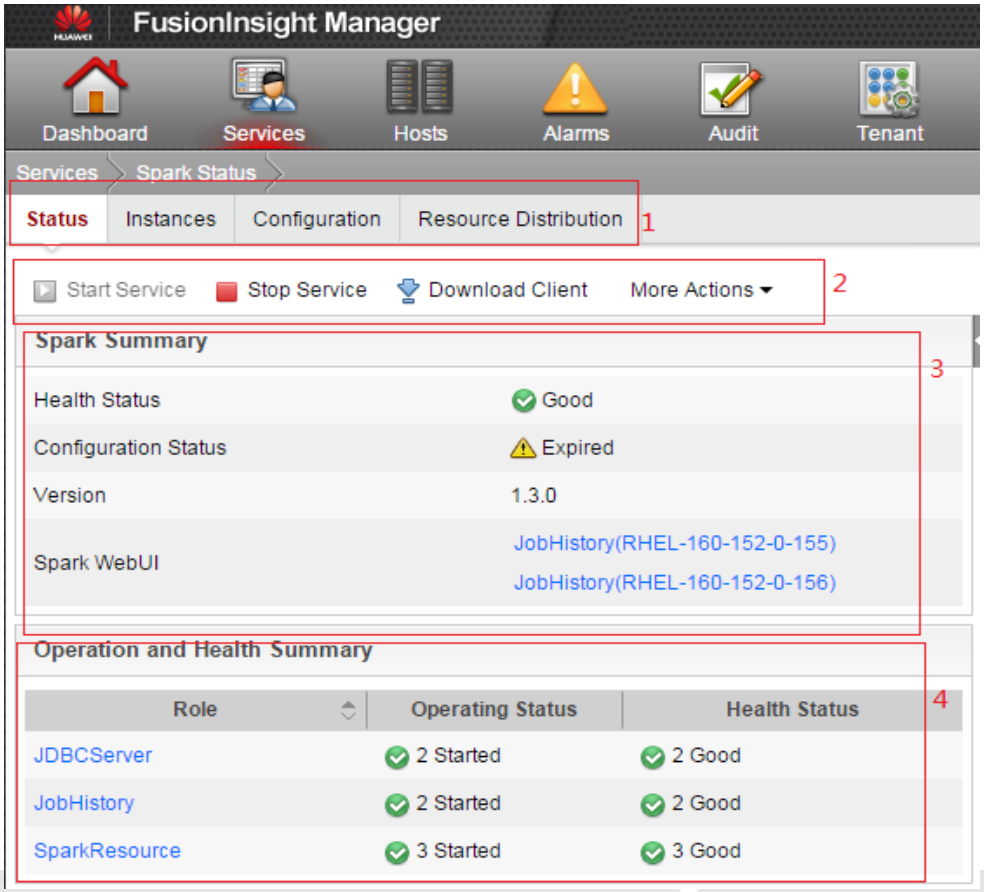
FusionInsight平台为**Spark**服务提供了管理监控的可视化界面，通过**Web UI**界面，可完成以下动作：

1. 服务状态信息、角色信息以及开放的配置项

2. 管理操作：启停spark、下载spark客户端、同步配置

3. 服务总体概况

4. 角色的显示和健康状况，点击相应角色可查看角色下的实例



HUAWEI

Spark与其他组件交互

在**FI**集群中，**Spark**主要与以下组件进行交互：

1）HDFS：Spark在**HDFS**文件系统中读写数据**(**必选**)**

2）YARN：Spark任务的运行以来**Yarn**来进行资源的调度管理**(**必选**)**

3）DBService：Spark-sql的表存储在**Dbservice**的数据库中**(**必选**)**

4）Zookeeper，JDBCServer的**HA**的实现依赖于**Zookeeper**的协调**(**必选**)**

5）Kafka：Spark可以接收**Kafka**发送的数据流**(**可选**)**

6）Hbase：Spark可以操作**Hbase**的表**(**可选**)**

Spark常用进程

- **JDBCServer**
  - 实际上是一个长驻的**spark**应用，通过**shell_start-thriftserver.sh**脚本启动。
  - 用户可以通过执行**beeline**脚本，连接**JDBCServer**，执行**sql**语句
- **JobHistory**
  - 是一个单节点进程，通过**shell_start-history-server.sh**脚本启动
  - 该进程用于提供**HistoryServer**页面，展示历史应用的执行信息

服务维护关键点

- **Spark**在**FusionInsight**平台中主要有三个角色，其中**SparkResource**主要为**Spark**任务执行提供必要的资源，**JobHistory**和**JDBCServer**这两个角色提供服务。
  - 当**spark**角色异常时，可到对应节点的**/var/log/Bigdata/spark/**路径下查看相关日志；
  - 当**Spark**任务运行失败时，可通过**FI**的链接在**yarn**原生界面查看对应任务的日志信息或者到**yarn.nodemanager.log-dirs**的位置查看相应的**container**日志；如果任务已经运行完毕，且**yarn**上开启日志归集功能(通过**yarn.log-aggregation-enable**配置)，则日志应到**hdfs**文件系统中查看。

本章总结

- 本章主要介绍了**Spark**组件再**FusionInsight**平台中的使用和日志的记录的查询读取。

学习推荐

- 华为**Learning**网站
  - **http://support.huawei.com/learning/Index!toTrainIndex**
- 华为**Support**案例库
  - **http://support.huawei.com/enterprise/servicecenter?lang=zh**

HUAWEI

谢谢

www.huawei.com