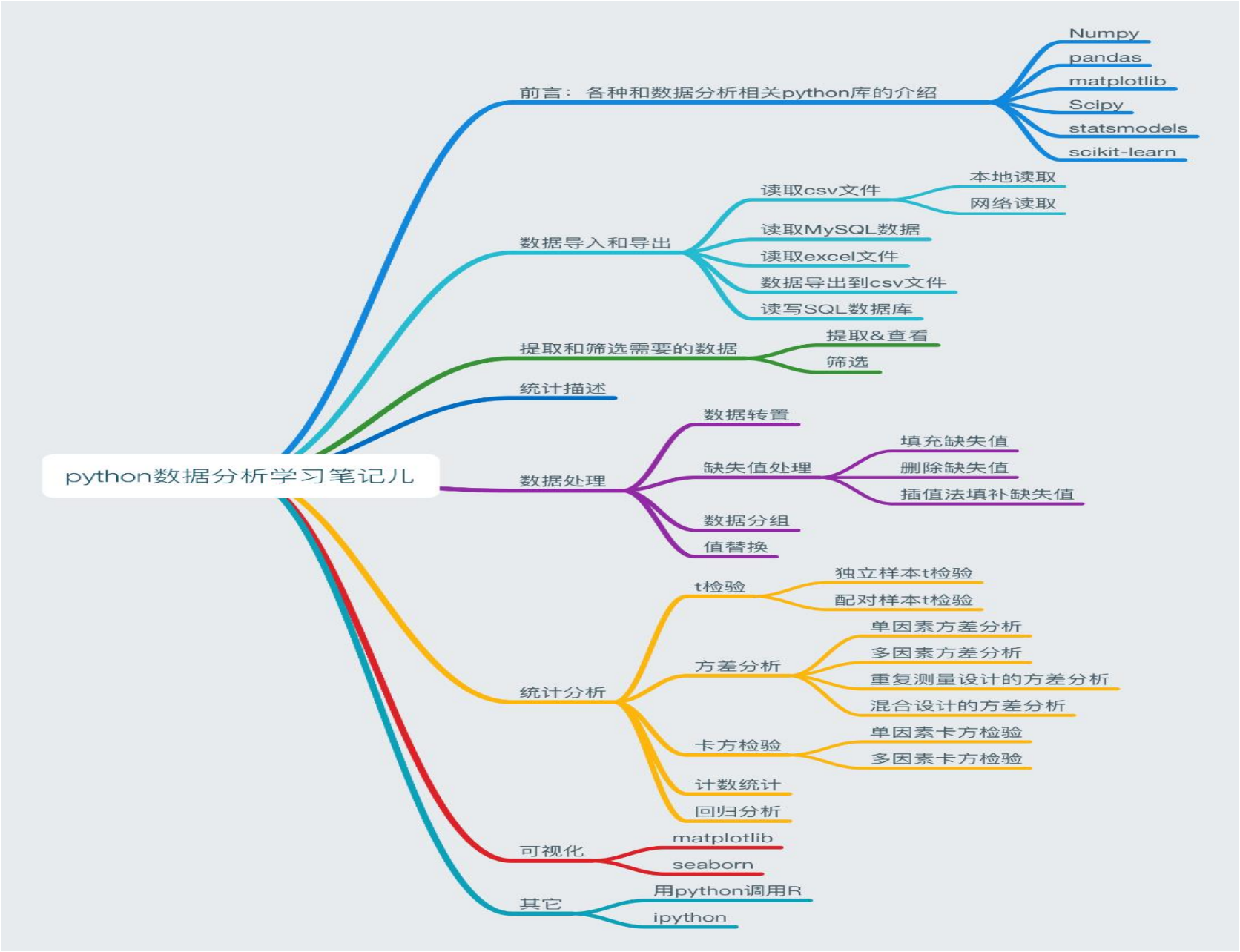
大数据分析入门


python数据分析学习笔记儿
前言： 各种和数据分析相关python库的介绍
Numpy
pandas
matplotlib
Scipy
statsmodels
scikit-learn
数据导入和导出
读取csv文件
本地读取
网络读取
读取MySQL数据
读取excel文件
数据导出到csv文件
读写SQL数据库
提取和筛选需要的数据
提取&查看
筛选
统计描述
数据处理
数据转置
缺失值处理
填充缺失值
删除缺失值
插值法填补缺失值
数据分组
值替换
统计分析
t检验
独立样本t检验
配对样本t检验
方差分析
单因素方差分析
多因素方差分析
重复测量设计的方差分析
混合设计的方差分析
卡方检验
单因素卡方检验
多因素卡方检验
计数统计
回归分析
可视化
matplotlib
seaborn
其它
用python调用R
ipython

python 大数据挖掘系列之基础知识入门

preface

Python 在大数据行业非常火爆近两年，as a pythonic，所以也得涉足下大数据分析，下面就聊聊它们。

Python 数据分析与挖掘技术概述

所谓数据分析，即对已知的数据进行分析，然后提取出一些有价值的信息，比如统计平均数，标准差等信息，数据分析的数据量可能不会太大，而数据挖掘，是指对大量的数据进行分析与挖倔，得到一些未知的，有价值的信息等，比如从网站的用户和用户行为中挖掘出用户的潜在需求信息，从而对网站进行改善等。

数据分析与数据挖掘密不可分，数据挖掘是对数据分析的提升。数据挖掘技术可以帮助我们更好的发现事物之间的规律。所以我们可以利用数据挖掘技术可以帮助我们更好的发现事物之间的规律。比如发掘用户潜在需求，实现信息的个性化推送，发现疾病与病状甚至病与药物之间的规律等。

预先善其事必先利其器

我们首先聊聊数据分析的模块有哪些：

1. numpy 高效处理数据，提供数组支持，很多模块都依赖它，比如 pandas，scipy，matplotlib 都依赖他，所以这个模块都是基础。所以必须先安装 numpy。
2. pandas 主要用于进行数据的采集与分析
3. scipy 主要进行数值计算。同时支持矩阵运算，并提供了很多高等数据处理功能，比如积分，微分方程求样等。
4. matplotlib 作图模块，结合其他数据分析模块，解决可视化问题
5. statsmodels 这个模块主要用于统计分析
6. Gensim 这个模块主要用于文本挖掘
7. sklearn，keras 前者机器学习，后者深度学习。

python 大数据招聘

[招聘要求]:

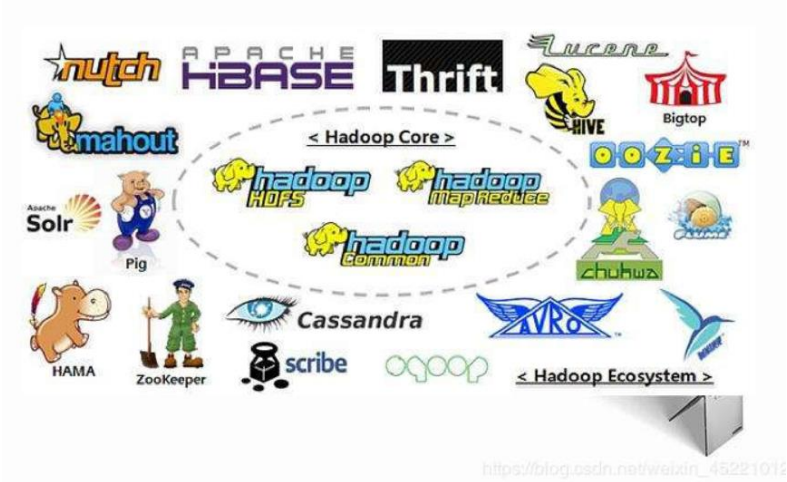
1、 大数据：有 hadoop、spark、flink 等至少一种大数据平台的使用经验（、熟悉 Hadoop 大数据生态圈及各组件的原理，并有实际部署维护经验以及调优经验；包括但不限于 HDFS、YARN、Spark、Flink、Druid、HBase、Kerberos、Hive、Zookeeper、Greenplum 等；
）；
2、有用户行为分析、客户画像、智能推荐、数据仓库建设、商业数据分析、增长项目经验者优先；
3、具备较强的表达能力和抽象总结能力，具备极强的逻辑思维能力；
4、有自然语言处理、机器学习经验者优先；
5、熟悉数据结构和算法，Python 或 Java 基础扎实，熟悉 Python 或 Java 常见框架，至少熟悉 Ansible、SaltStack 中的一种自动化运维工具，具有 Django，Flask 等框架开发经验；
6、熟悉 Linux 编程环境和常用的编程，调试工具，具有 DevOps 系统搭建及开发经验(必须)，包括但不限于自动化运维平台、CMDB、自动发布平台，有基于 Python 的全栈开发经验尤佳；

Python 开发架构师

岗位要求：
1、具有团队协作工作经验，能熟练使用 svn,git 等分布式代码控制管理仓库工具；
2、深入了解 Python 和 Golang 语言，至少了解一种 framework（tornado,flask,Django 等）；
3、熟悉或了解一种或多种高级编程语言（除 python 外）,如 nodejs,go,ruby,等优先考虑；
4、熟悉 PostgreSQL/Redis 等常用的开源存储工具；
5、熟练掌握 OO 的编程思想，掌握多种常用设计模式；
6、了解 mirco service，了解 soa 架构这优先考虑;
7、了解异步处理，消息处理模式，具备 HTTP，TCP 等网络服务端开发经验，并能针对具体业务场景做并发异步服务的开发和优化。

Hadoop 入门——初识 Hadoop

生态化反——hadhoop 生态圈
hadhoop 动物园
spark



hadhoop 动物园

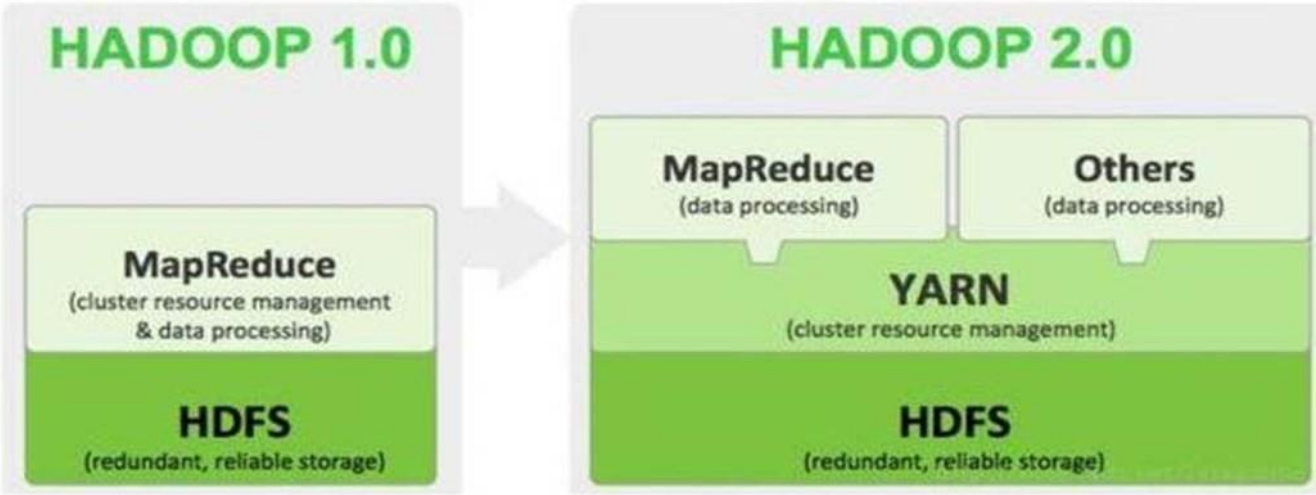
apache 开源的分布式计算框架(一系列产品)。
HDFS(hadhoop distribute file system)，很平滑，不够就加普通 PC，冗余备份，(参考 raid0 1 2 3 4 5) ,pd.read_hdf()，pandas 也可以读
MapReduce(YARN2.0)，分布式计算框架，求和、字频；不能分布计算，序列式(圆周率、斐波那契数列)；将不能分布式改成分布式。
HIVE(模仿 SQL，进行 SQL 查询的工具)
HBase(NoSQL 数据库)
ZooKeeper
Kafka(类似消息队列)
lucene(全文检索)
mahout(java 实现的类似于 sklearn 的机器学习库)

spark

不包含 HDFS 文件系统，像 pandas 自己不带数据库，支持含 hadhoop 等各种文件系统。
RDD
spark-streaming(消息队列)

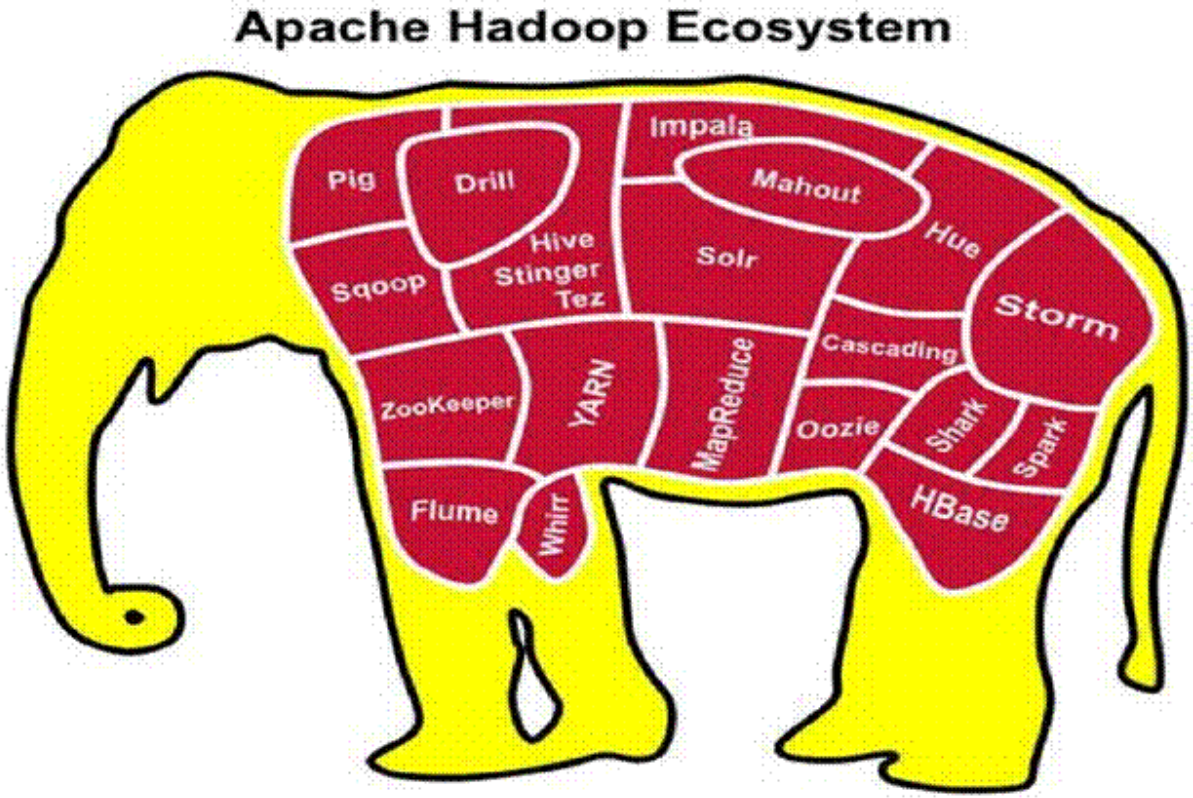
1． Hadoop 介绍

Hadoop 是 Apache 旗下的一个用 java 语言实现开源软件框架，是一个开发和运行处理大规模数据的软件平台。允许使用简单的编程模型在大量计算机集群上对大型数据集进行分布式处理。



狭义上说，Hadoop 指 Apache 这款开源框架，它的核心组件有：

- HDFS（分布式文件系统）：解决海量数据存储
- YARN（作业调度和集群资源管理的框架）：解决资源任务调度
- MAPREDUCE（分布式运算编程框架）：解决海量数据计算 广义上来说，Hadoop 通常是指一个更广泛的概念——Hadoop 生态圈。



当下的 Hadoop 已经成长为一个庞大的体系，随着生态系统的成长，新出现的项目越来越多，其中不乏一些非 Apache 主管的项目，这些项目对 HADOOP 是很好的补充或者更高层的抽象。比如：

- HDFS：分布式文件系统
- MAPREDUCE：分布式运算程序开发框架
- HIVE：基于 HADOOP 的分布式数据仓库，提供基于 SQL 的查询数据操作
- HBASE：基于 HADOOP 的分布式海量数据库
- ZOOKEEPER：分布式协调服务基础组件
- Mahout：基于 mapreduce/spark/flink 等分布式运算框架的机器学习算法库
- Oozie：工作流调度框架
- Sqoop：数据导入导出工具（比如用于 mysql 和 HDFS 之间）
- Flume：日志数据采集框架
- Impala：基于 Hadoop 的实时分析

2． Hadoop 发展简史

Hadoop 是 Apache Lucene 创始人 Doug Cutting 创建的。最早起源于 Nutch，它是 Lucene 的子项目。Nutch 的设计目标是构建一个大型的全网搜索引擎，包括网页抓取、索引、查询等功能，但随着抓取网页数量的增加，遇到了严重的可扩展性问题：如何解决数十亿网页的存储和索引问题。

2003 年 Google 发表了一篇论文为该问题提供了可行的解决方案。论文中描述的是谷歌的产品架构，该架构称为：谷歌分布式文件系统（GFS）,可以解决他们在网页爬取和索引过程中产生的超大文件的存储需求。

2004 年 Google 发表论文向全世界介绍了谷歌版的 MapReduce 系统。 同时期，Nutch 的开发人员完成了相应的开源实现 HDFS 和 MAPREDUCE，并从 Nutch 中剥离成为独立项目 HADOOP，到 2008 年 1 月，HADOOP 成为 Apache 顶级项目，迎来了它的快速发展期。

2006 年 Google 发表了论文是关于 BigTable 的，这促使了后来的 Hbase 的发展。 因此，Hadoop 及其生态圈的发展离不开 Google 的贡献。

3． Hadoop 特性优点

扩容能力（Scalable）：Hadoop 是在可用的计算机集群间分配数据并完成计算任务的，这些集群可用方便的扩展到数以千计的节点中。

成本低（Economical）：Hadoop 通过普通廉价的机器组成服务器集群来分发以及处理数据，以至于成本很低。

高效率（Efficient）：通过并发数据，Hadoop 可以在节点之间动态并行的移动数据，使得速度非常快。 可靠性（Rellable）：能自动维护数据的多份复制，并且在任务失败后能自动地重新部署（redeploy）计算任务。所以 Hadoop 的按位存储和处理数据的能力值得人们信赖。

4． Hadoop 国内外应用

不管是国内还是国外，Hadoop 最受青睐的行业是互联网领域，可以说互联网公司是 hadoop 的主要使用力量。 国外来说，Yahoo、Facebook、IBM 等公司都大量使用 hadoop 集群来支撑业务。比如：

Yahoo 的 Hadoop 应用在支持广告系统、用户行为分析、支持 Web 搜索等。

Facebook 主要使用 Hadoop 存储内部日志与多维数据，并以此作为报告、分析和机器学习的数据源。 国内来说，BAT 领头的互联网公司是当仁不让的 Hadoop 使用者、维护者。比如 Ali 云梯（14 年国内最大 Hadoop 集群）、百度的日志分析平台、推荐引擎系统等。

国内其他非互联网领域也有不少 hadoop 的应用，比如： 金融行业： 个人征信分析 证券行业： 投资模型分析 交通行业： 车辆、路况监控分析 电信行业： 用户上网行为分析 总之：hadoop 并不会跟某种具体的行业或者某个具体的业务挂钩，它只是一种用来做海量数据分析处理的工具。

Hadoop

Hadoop 最早起源于 **Nutch**。

Nutch 的设计目标是构建一个大型的**全网搜索引擎**，包括网页**抓取、索引、查询**等功能，但随着抓取网页数量的增加，遇到了严重的可扩展性问题——**如何解决数十亿网页的存储和索引问题**。

2003 年、2004 年谷歌发表的两篇论文为该问题提供了可行的解决方案。

——**分布式文件系统**（GFS），可用于处理海量网页的存储

——**分布式计算框架 MAPREDUCE**，可用于处理海量网页的索引计算问题。

狭义上来说，hadoop 就是单独指代 hadoop 这个软件，

广义上来说，hadoop 指代大数据的一个生态圈，包括很多其他的软件

hadoop 的四大特点:

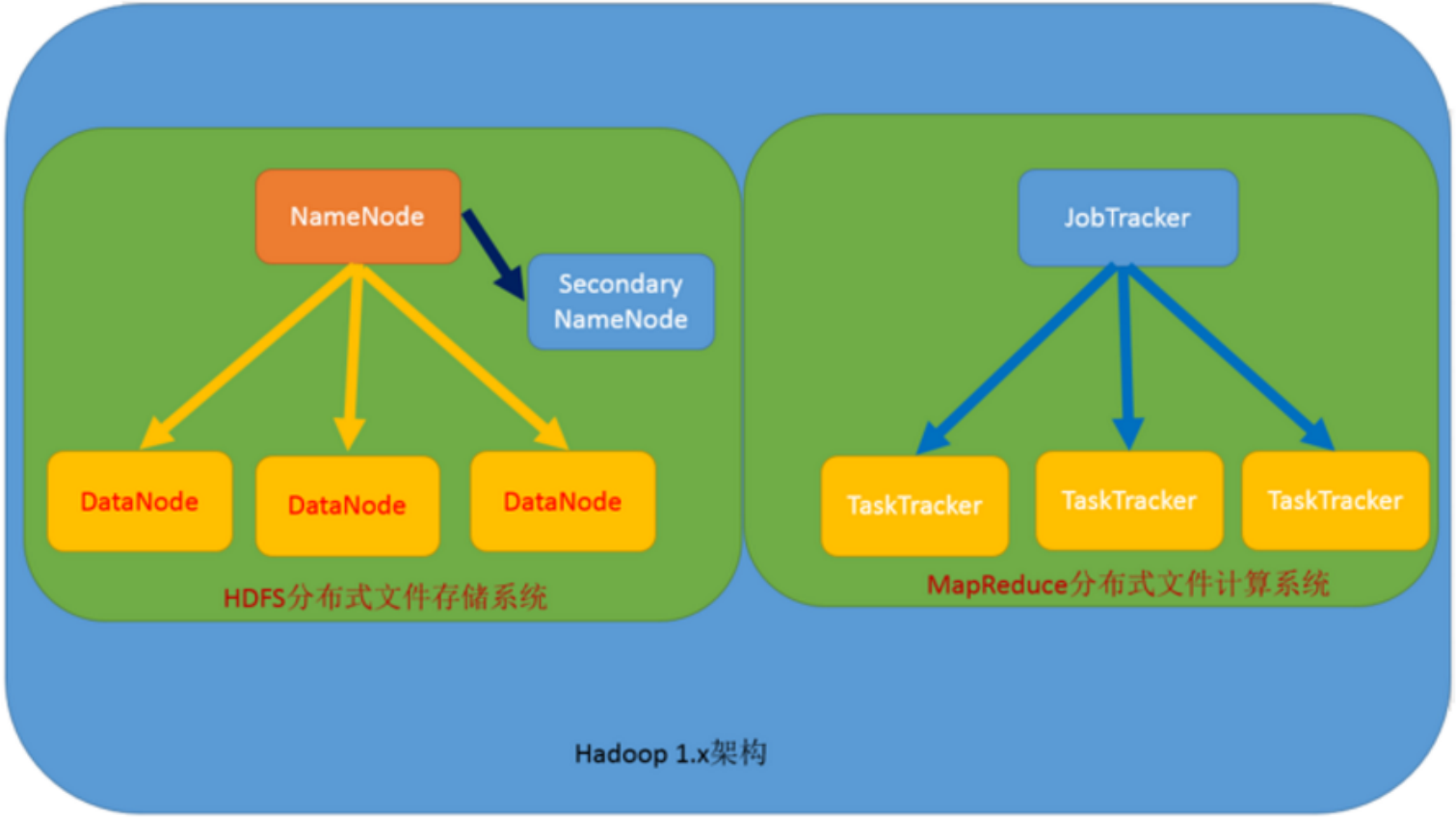
1.扩展能力:Hadoop 在可用的计算机集群分配数据并完成计算任务的,可以扩展到数以千计节点之中.

2.成本低:hadoop 通过廉价的机器组成服务器集群来分发以及处理数据,以至于成本很低.

3.高效率:通过并发数据,Hadoop 课以在节点之间动态并行的移动数据.是的速度非常快

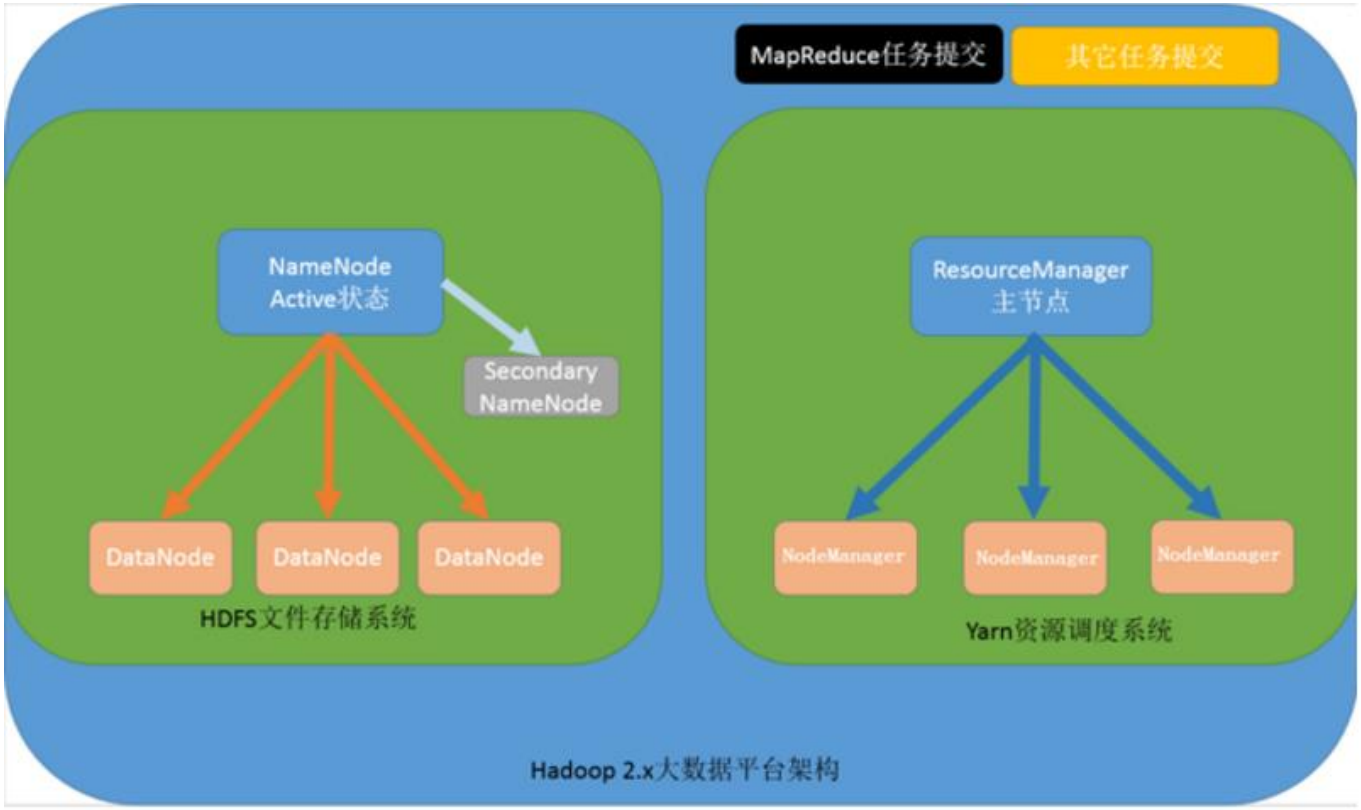
4.可靠性:能自动维护数据的多酚复制,并在任务失败后能自动的重新部署计算任务,所以 hadoop 的按位存储和处理数据的能力值得人们信赖.

(hadoop 有好多版本,但高版本不一定包含低版本的特性)
1.x 版本架构模型:

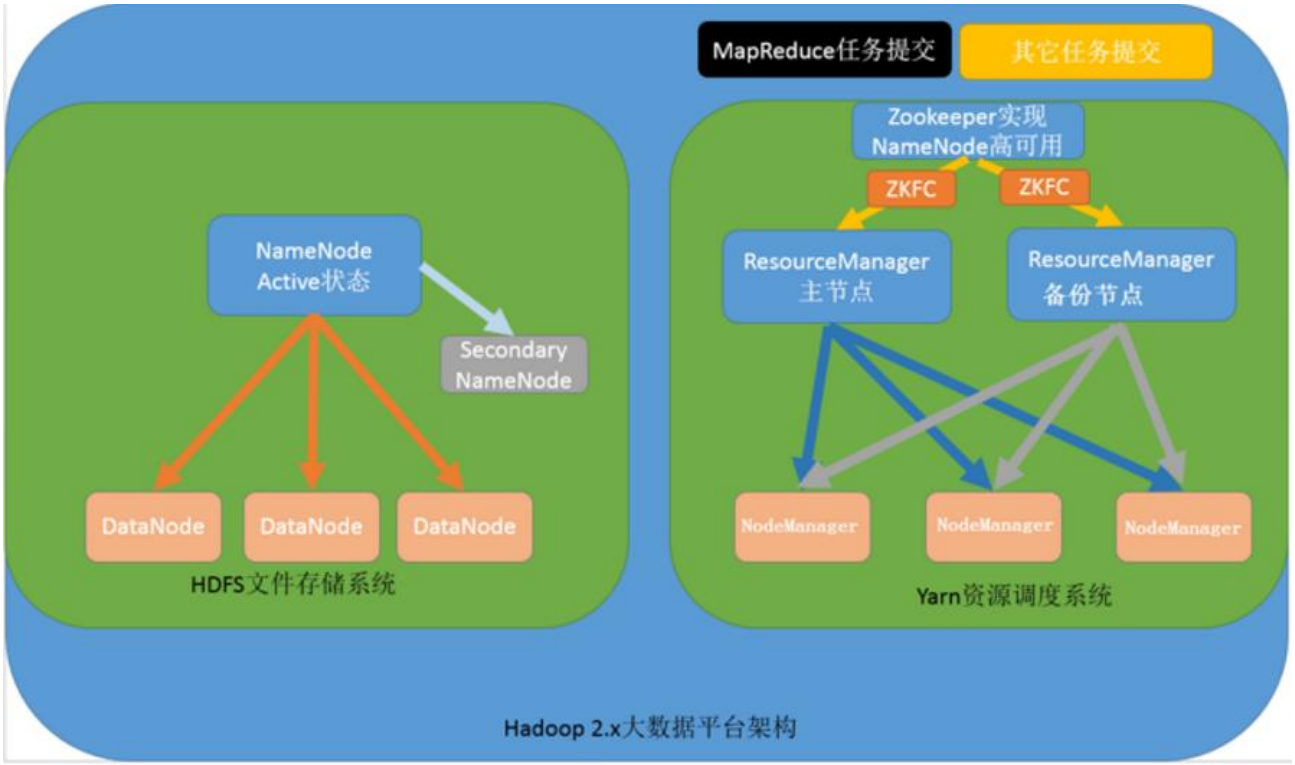


文件系统核心模块:
NameNode:集群当中的主节点,主要用于管理集群当中的各种数据
secondaryNameNode:主要能用于 hadoop 当中元数据信息的辅助管理
DataNode:集群当中的从节点,主要用于存储集群当中的各种数据
数据计算核心模块:
JobTracker:接收用户的计算请求任务,并分配任务给从节点
TaskTracker:负责执行主节点 JobTracker 分配的任务
2.x 版本架构模型介绍:

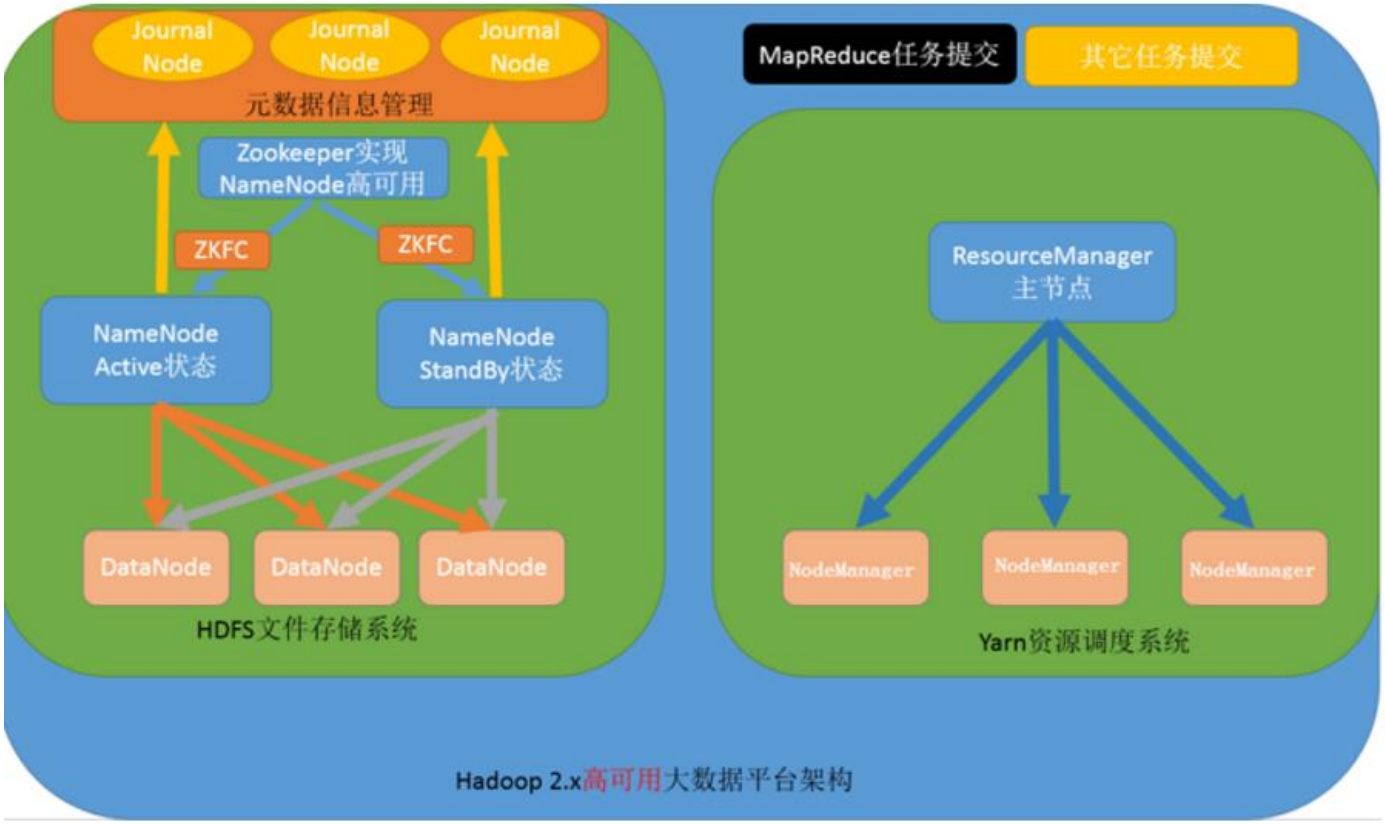
第一种 namenode 与 resourceManager 单点架构模型:



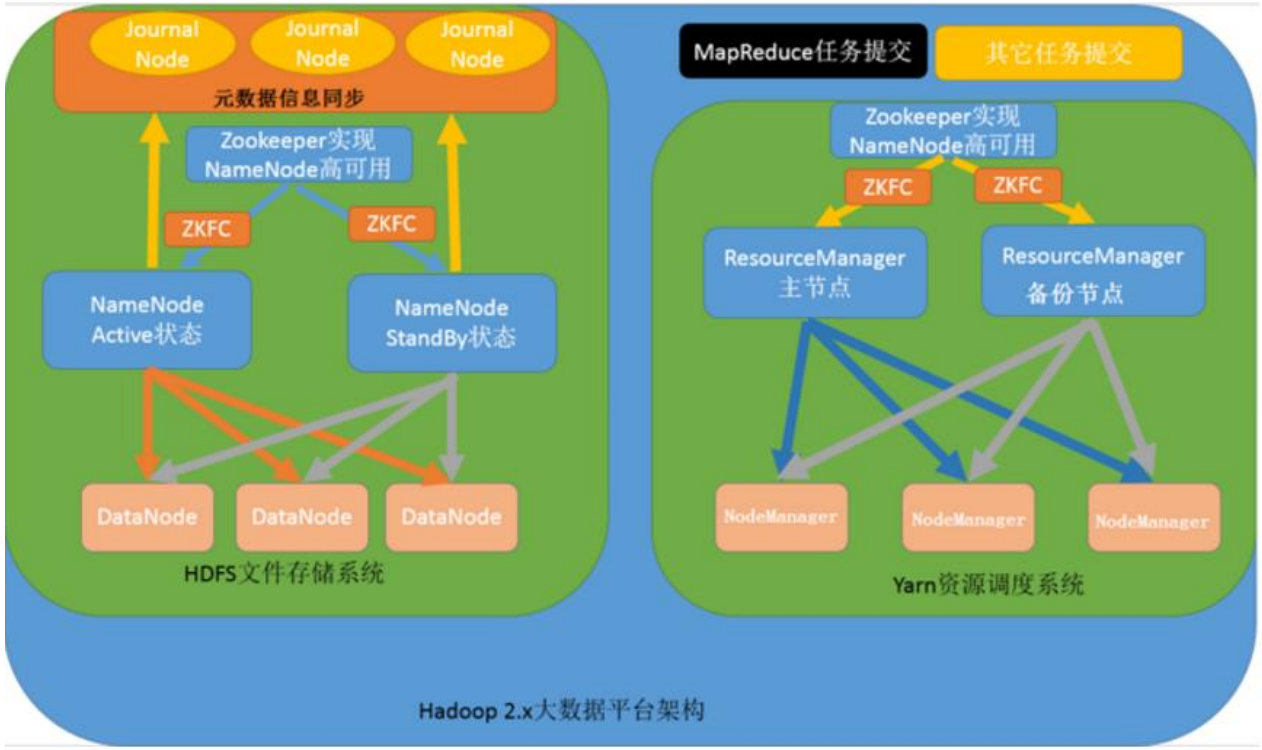
文件系统核心模块：
NameNode：集群当中的主节点，主要用于管理集群当中的各种数据
secondaryNameNode：主要能用于 hadoop 当中元数据信息的辅助管理
DataNode：集群当中的从节点，主要用于存储集群当中的各种数据
数据计算核心模块：
ResourceManager：接收用户的计算请求任务，并负责集群的资源分配，以及计算任务的划分
NodeManager：负责执行主节点 ResourceManager 分配的任务
第二种：NameNode 单节点与 ResourceManager 高可用架构模型



文件系统核心模块：
NameNode：集群当中的主节点，主要用于管理集群当中的各种数据
secondaryNameNode：主要能用于 hadoop 当中元数据信息的辅助管理
DataNode：集群当中的从节点，主要用于存储集群当中的各种数据
数据计算核心模块：
ResourceManager：接收用户的计算请求任务，并负责集群的资源分配，以及计算任务的划分，通过 zookeeper 实现 ResourceManager 的高可用
NodeManager：负责执行主节点 ResourceManager 分配的任务
第三种：NameNode 高可用与 ResourceManager 单节点架构模型



文件系统核心模块：
NameNode：集群当中的主节点，主要用于管理集群当中的各种数据，其中 nameNode 可以有两个，形成高可用状态
DataNode：集群当中的从节点，主要用于存储集群当中的各种数据
JournalNode：文件系统元数据信息管理
数据计算核心模块：
ResourceManager：接收用户的计算请求任务，并负责集群的资源分配，以及计算任务的划分
NodeManager：负责执行主节点 ResourceManager 分配的任务
第四种：NameNode 与 ResourceManager 高可用架构模型



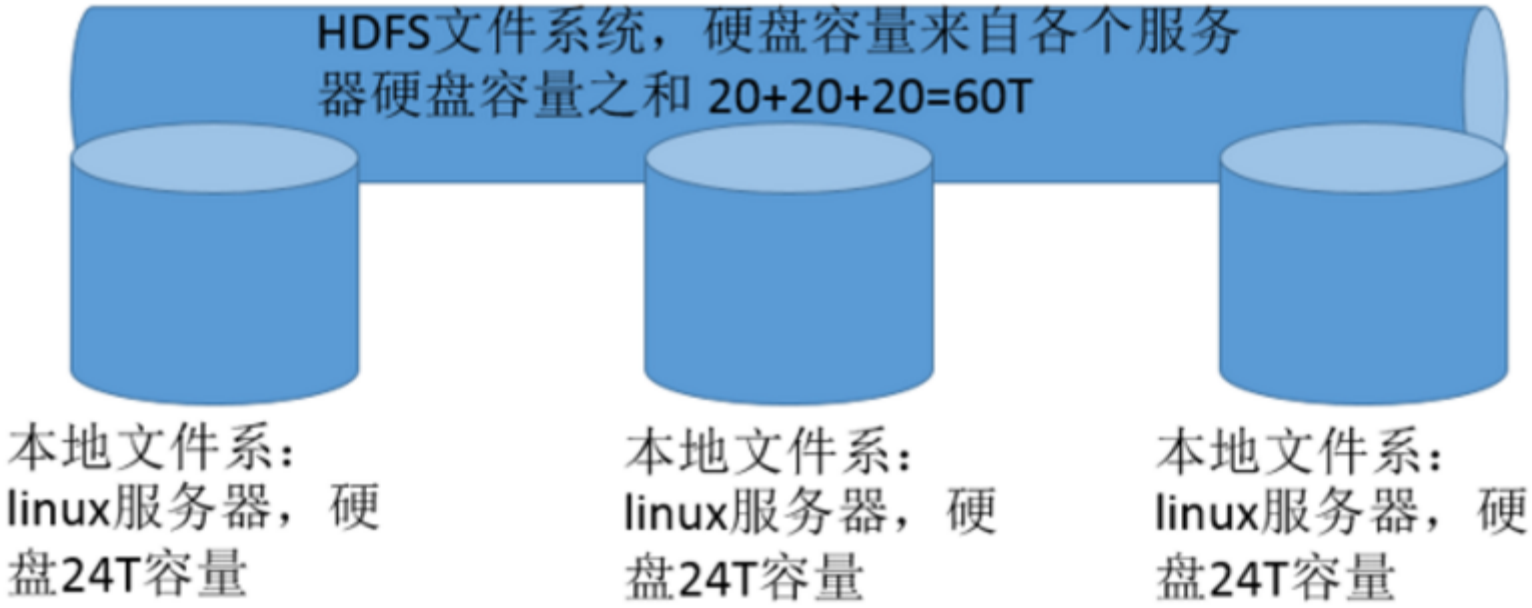
文件系统核心模块：
NameNode：集群当中的主节点，主要用于管理集群当中的各种数据，一般都是使用两个，实现 HA 高可用
JournalNode：元数据信息管理进程，一般都是奇数个
DataNode：从节点，用于数据的存储

数据计算核心模块：
ResourceManager：Yarn 平台的主节点，主要用于接收各种任务，通过两个，构建成高可用
NodeManager：Yarn 平台的从节点，主要用于处理 ResourceManager 分配的任务

hdfs 入门介绍

hdfs 是 hadoop 的分布式文件存储系统,是 hadoop 的核心组件之一,作为最底层的分布式存储服务而存在.

分布式文件系统解决的问题就是大数据存储.它们是横跨多个计算机存储系统的存储服务.分布式文件系统在大数据有着广泛的应用前景,它们为存储和处理超大规模数据提供所需的扩展能力.



hdfs 的特性:

用于存储文件,通过统一的命名空间目录树来定位文件.

分布式的,有很多服务器联合来实现功能,集群中的服务器都有各自的角色

master/slave 架构

hdfs 采用 master/slave 架构,一般一个 hdfs 集群是有一个 namenode 和一定数目的 datanode 组成,namenode 是主节点,datanode 是从节点,两种角色各司其职,共同协调完成分布式的文件存储服务.

分块存储

hdfs 中的文件在物理上是分块存储的,快的大小可以通过配置参数来规定,默认大在 hadoop 中是 128M

命名空间

hdfs 支持传统的层次型文件组织结构,用户或者应用用程序创建目录,然后将文件保存在这些目录里,文件系统的名字空间的层次结构和大多数现有的文件系统类似:用户可以创建,删除,移动,或者重命名文件.

namenode 负责维护文件系统的名字空间,任何对文件系统的名字空间或者属性修改都被 namenode 记录下来.

namenode 元数据管理:

我们把目录结构和文件分块位置信息叫做元数据,namenode 负责维护整个 hdfs 文件系统的目录树结构,以及每个节点的 block 信息(block 的 id,及所在的 datanode 服务器)

datanode 数据存储

文件的各个 block 的具体存储管理由 datanode 节点承担,每个 block 都可以在多个 datanode 上,datanode 需要定向 namenode 汇报自己持有的 block 信息,存储多个副本(副本数量通过参数设置,默认是三个)

副本机制

为了容错,文件的所有 block 都会有副本,每个文件的 block 大小和副本系数都是可以配置的,应用程序可以指定文件的副本数量,副本系数可以在文件创建的时候指定,也可以在之后改变

一次写入,多次读出

hdfs 设计成适应一次写入多次读出的场景,且不支持文件的 修改

hdfs 适合用来做大数据分析的底层存储服务,并不适合做网盘等应用,因为修改不方便,延迟大,网络开销大,成本太高.

Hadoop 入门——初识 Hadoop

一.hadoop 是什么

Hadoop 被公认是一套行业大数据标准开源软件，在分布式环境下提供了海量数据的处理能力。几乎所有主流厂商都围绕 Hadoop 开发工具、开源软件、商业化工具和技术服务。今年大型 IT 公司，如 EMC、Microsoft、Intel、Teradata、Cisco 都明显增加了 Hadoop 方面的投入。

二 .hadoop 能干什么

hadoop 擅长日志分析，facebook 就用 Hive 来进行日志分析，2009 年时 facebook 就有非编程人员的 30%的人使用 HiveQL 进行数据分析；淘宝搜索中的自定义筛选也使用的 Hive；利用 Pig 还可以做高级的数据处理，包括 Twitter、LinkedIn 上用于发现您可能认识的人，可以实现类似 Amazon.com 的协同过滤的推荐效果。淘宝的商品推荐也是！在 Yahoo！的 40%的 Hadoop 作业是用 pig 运行的，包括垃圾邮件的识别和过滤，还有用户特征建模。（2012 年 8 月 25 新更新，天猫的推荐系统是 hive，少量尝试 mahout！）

三.hadoop 的核心

1.HDFS: Hadoop Distributed File System 分布式文件系统

2.YARN: Yet Another Resource Negotiator 资源管理调度系统

3.Mapreduce：分布式运算框架

四.HDFS 的架构

主从结构

- 主节点， namenode
- 从节点，有很多个: datanode

namenode 负责：

- 接收用户操作请求
- 维护文件系统的目录结构
- 管理文件与 block 之间关系，block 与 datanode 之间关系

datanode 负责：

- 存储文件
- 文件被分成 block 存储在磁盘上
- 为保证数据安全，文件会有多个副本

Secondary NameNode 负责：

合并 fsimage 和 edits 文件来更新 NameNode 的 metedata

五.Hadoop 的特点

扩容能力（Scalable）：能可靠地（reliably）存储和处理千兆字节（PB）数据。

成本低（Economical）：可以通过普通机器组成的服务器群来分发以及处理数据。这些服务器群总计可达数千个节点。

高效率（Efficient）：通过分发数据，hadoop 可以在数据所在的节点上并行地（parallel）处理它们，这使得处理非常的快速。

可靠性（Reliable）：hadoop 能自动地维护数据的多份副本，并且在任务失败后能自动地重新部署（redeploy）计算任务。

六.NameNode

1.简介

namenode 是整个文件系统的管理节点。他维护着整个文件系统的文件目录树，文件/目录的元信息和每个文件对应的数据块列表。接收用户的操作请求。

文件包括：

fsimage:元数据镜像文件。存储某一时段 NameNode 内存元数据信息。

edits:操作日志文件。

fstime:保存最近一次 checkpoint 的时间。

2.NameNode 的工作特点

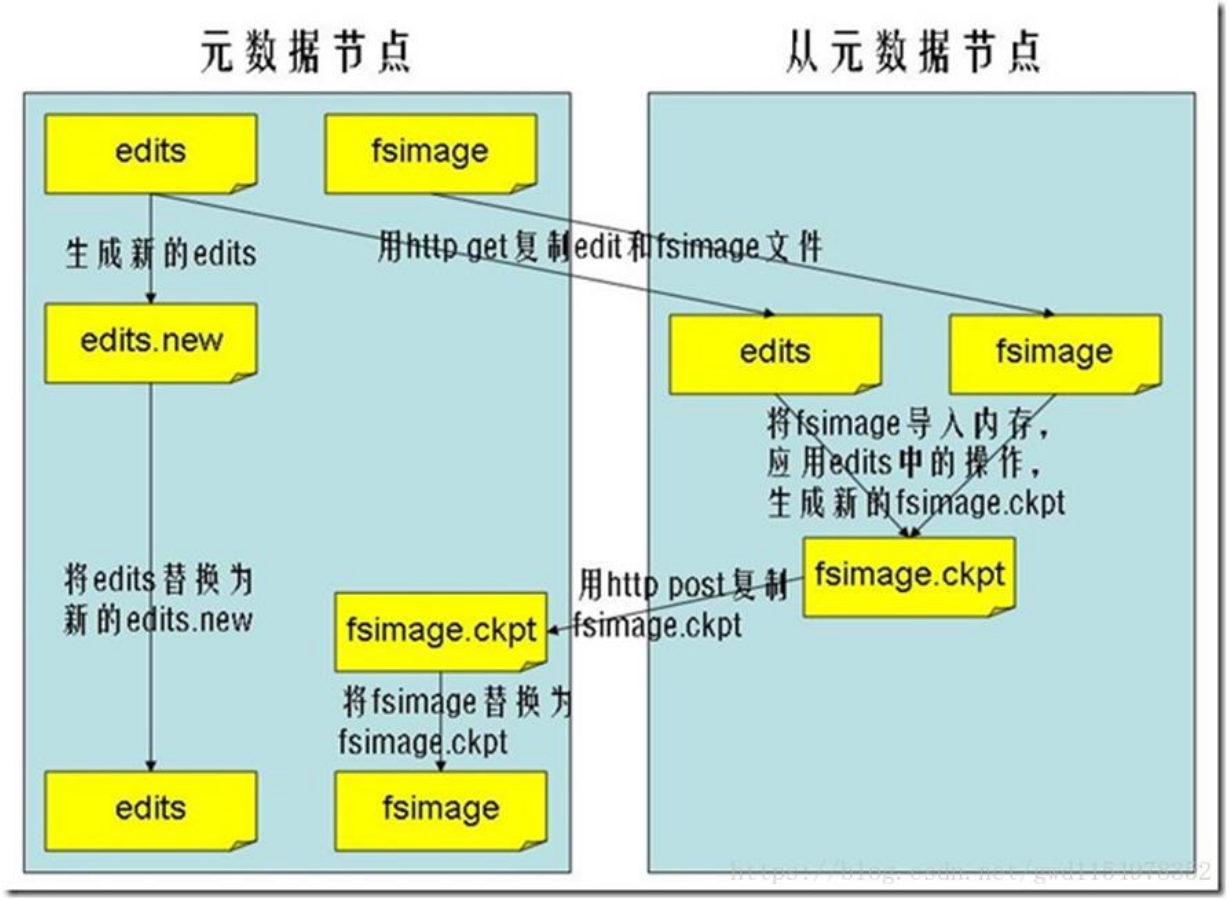
NameNode 始终在内存中保存 metedata，用于处理“读请求”，到有“写请求”到来时，NameNode 首先会写 editlog 到磁盘，即向 edits 文件中写日志，成功返回后，才会修改内存，并且向客户端返回。

Hadoop 会维护一个人 fsimage 文件，也就是 NameNode 中 metedata 的镜像，但是 fsimage 不会随时与 NameNode 内存中的 metedata 保持一致，而是每隔一段时间通过合并 edits 文件来更新内容。Secondary NameNode 就是用来合并 fsimage 和 edits 文件来更新 NameNode 的 metedata 的。

3.什么时候 checkpoint

fs.checkpoint.period 指定两次 checkpoint 的最大时间间隔，默认 3600 秒。

fs.checkpoint.size 规定 edits 文件的最大值，一旦超过这个值则强制 checkpoint，不管是否到达最大时间间隔。默认大小是 64M。



七.SecondaryNameNode

1.简介

HA 的一个解决方案。但不支持热备。配置即可。

执行过程：从 NameNode 上下载元数据信息（fsimage,edits），然后把二者合并，生成新的 fsimage，在本地保存，并将其推送到 NameNode，替换旧的 fsimage.

默认在安装在 NameNode 节点上，但这样...不安全！

2.工作流程

（1）secondary 通知 namenode 切换 edits 文件；

（2）secondary 从 namenode 获得 fsimage 和 edits(通过 http)；

（3）secondary 将 fsimage 载入内存，然后开始合并 edits；

（4）secondary 将新的 fsimage 发回给 namenode；

（5）namenode 用新的 fsimage 替换旧的 fsimage；

八.DataNode

提供真实文件数据的存储服务。

文件块（block）：最基本的存储单位。对于文件内容而言，一个文件的长度大小是 size，那么从文件的 0 偏移开始，按照固定的大小，顺序对文件进行划分并编号，划分好的每一个块称一个 Block。HDFS 默认 Block 大小是 128MB，以一个 256MB 文件，共有 256/128=2 个 Block.

dfs.block.size

不同于普通文件系统的是，HDFS 中，如果一个文件小于一个数据块的大小，并不占用整个数据块存储空间;

Replication:多复本。默认是三个。

九.HDFS

（1）读过程

1.初始化 FileSystem，然后客户端(client)用 FileSystem 的 open()函数打开文件

2.FileSystem 用 RPC 调用元数据节点，得到文件的数据块信息，对于每一个数据块，元数据节点返回保存数据块的数据节点的地址。

3.FileSystem 返回 FSDataInputStream 给客户端，用来读取数据，客户端调用 stream 的 read()函数开始读取数据。

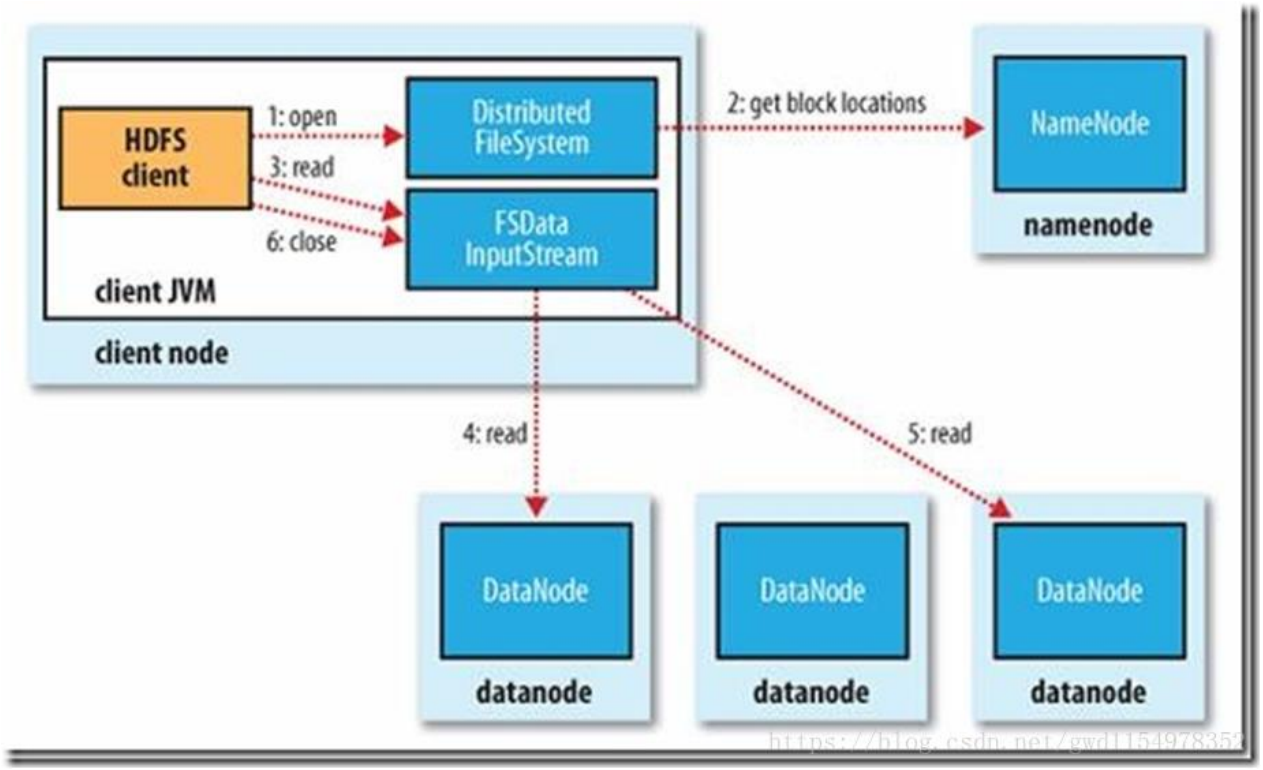
4.DFSInputStream 连接保存此文件第一个数据块的最近的数据节点，data 从数据节点读到客户端(client)

5.当此数据块读取完毕时，DFSInputStream 关闭和此数据节点的连接，然后连接此文件下一个数据块的最近的数据节点。

6.当客户端读取完毕数据的时候，调用 FSDataInputStream 的 close 函数。

7.在读取数据的过程中，如果客户端在与数据节点通信出现错误，则尝试连接包含此数据块的下一个数据节点。

8.失败的数据节点将被记录，以后不再连接。



（2）写过程

1.初始化 FileSystem，客户端调用 create()来创建文件

2.FileSystem 用 RPC 调用元数据节点，在文件系统的命名空间中创建一个新的文件，元数据节点首先确定文件原来不存在，并且客户端有创建文件的权限，然后创建新文件。

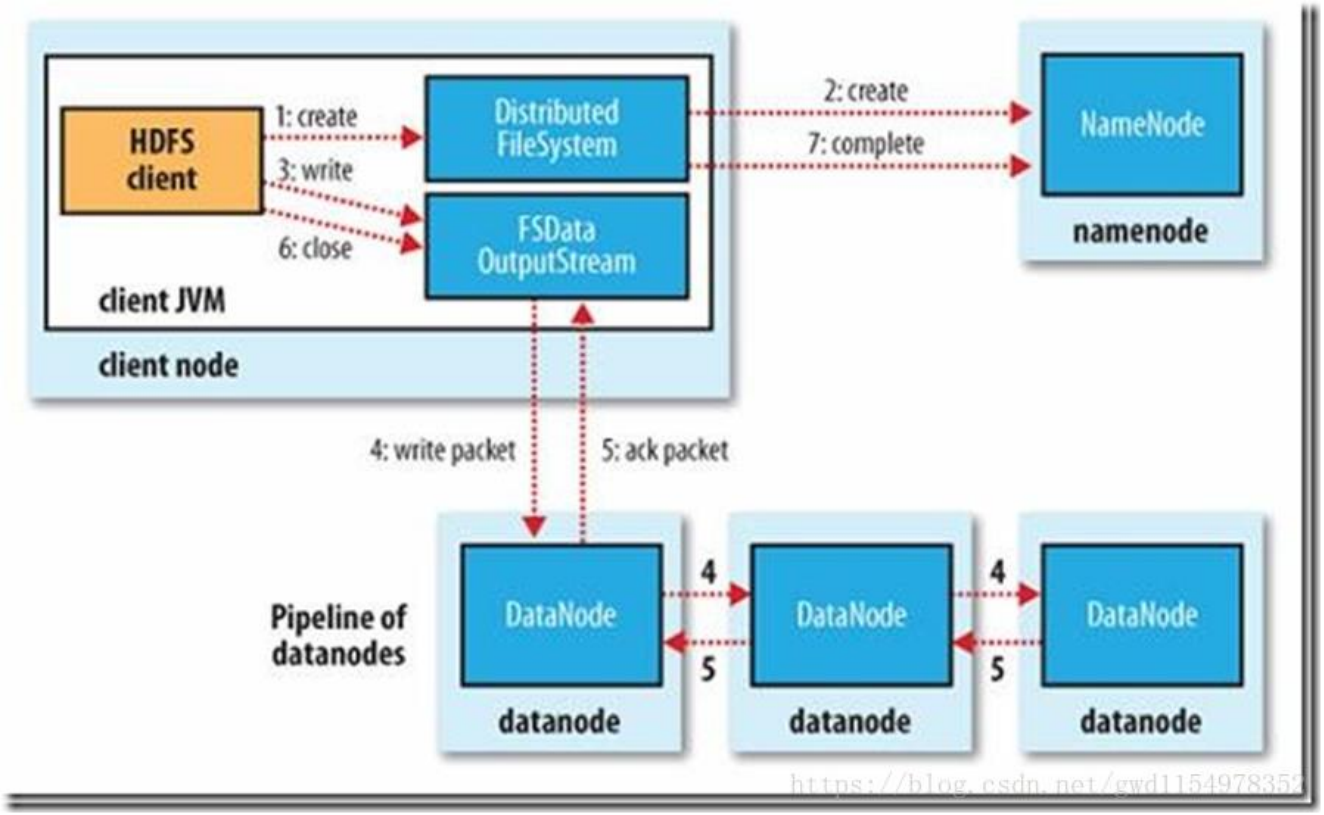
3.FileSystem 返回 DFSOutputStream，客户端用于写数据，客户端开始写入数据。

4.DFSOutputStream 将数据分成块，写入 data queue。data queue 由 Data Streamer 读取，并通知元数据节点分配数据节点，用来存储数据块(每块默认复制 3 块)。分配的数据节点放在一个 pipeline 里。Data Streamer 将数据块写入 pipeline 中的第一个数据节点。第一个数据节点将数据块发送给第二个数据节点。第二个数据节点将数据发送给第三个数据节点。

5.DFSOutputStream 为发出去的数据块保存了 ack queue，等待 pipeline 中的数据节点告知数据已经写入成功。

6.当客户端结束写入数据，则调用 stream 的 close 函数。此操作将所有的数据块写入 pipeline 中的数据节点，并等待 ack queue 返回成功。最后通知元数据节点写入完毕。

7.如果数据节点在写入的过程中失败，关闭 pipeline，将 ack queue 中的数据块放入 data queue 的开始，当前的数据块在已经写入的数据节点中被元数据节点赋予新的标示，则错误节点重启后能够察觉其数据块是过时的，会被删除。失败的数据节点从 pipeline 中移除，另外的数据块则写入 pipeline 中的另外两个数据节点。元数据节点则被通知此数据块是复制块数不足，将来会再创建第三份备份。



Hadoop 核心之分布式文件系统 HDFS

参考：https://www.cnblogs.com/maybe2030/p/4593190.html

Hadoop 分布式文件系统（Hadoop Distributed File System，简称 HDFS）是 Hadoop 的核心模块之一，它主要解决 Hadoop 的大数据存储问题，其思想来源与 Google 的文件系统 GFS。HDFS 的主要特点：

保存多个副本，且提供容错机制，副本丢失或宕机自动恢复。默认存 3 份。运行在廉价的机器上。

适合大数据的处理。**HDFS 默认会将文件分割成 block，64M 为 1 个 block。然后将 block 按键值对存储在 HDFS 上，并将键值对的映射存到内存中。**如果小文件太多，那内存的负担会很重。

HDFS 中的两个重要角色：

[Namenode]

1）管理文件系统的命名空间。

2）记录每个文件数据快在各个 Datanode 上的位置和副本信息。

3）协调客户端对文件的访问。

4）记录命名空间内的改动或者空间本省属性的改动。

5）Namenode 使用事务日志记录 HDFS 元数据的变化。使用映像文件存储文件系统的命名空间，包括文件映射，文件属性等。

从社会学来看，Namenode 是 HDFS 里面的管理者，发挥者管理、协调、操控的作用。

[Datanode]

1）负责所在物理节点的存储管理。

2）一次写入，多次读取（不修改）。

3）文件由数据库组成，一般情况下，数据块的大小为 64MB。

4）数据尽量散步到各个节点。

从社会学的角度来看，Datanode 是 HDFS 的工作者，发挥按着 Namenode 的命令干活，并且把干活的进展和问题反馈到 Namenode 的作用。

客户端如何访问 HDFS 中一个文件呢？具体流程如下:

1）首先从 Namenode 获得组成这个文件的数据块位置列表。

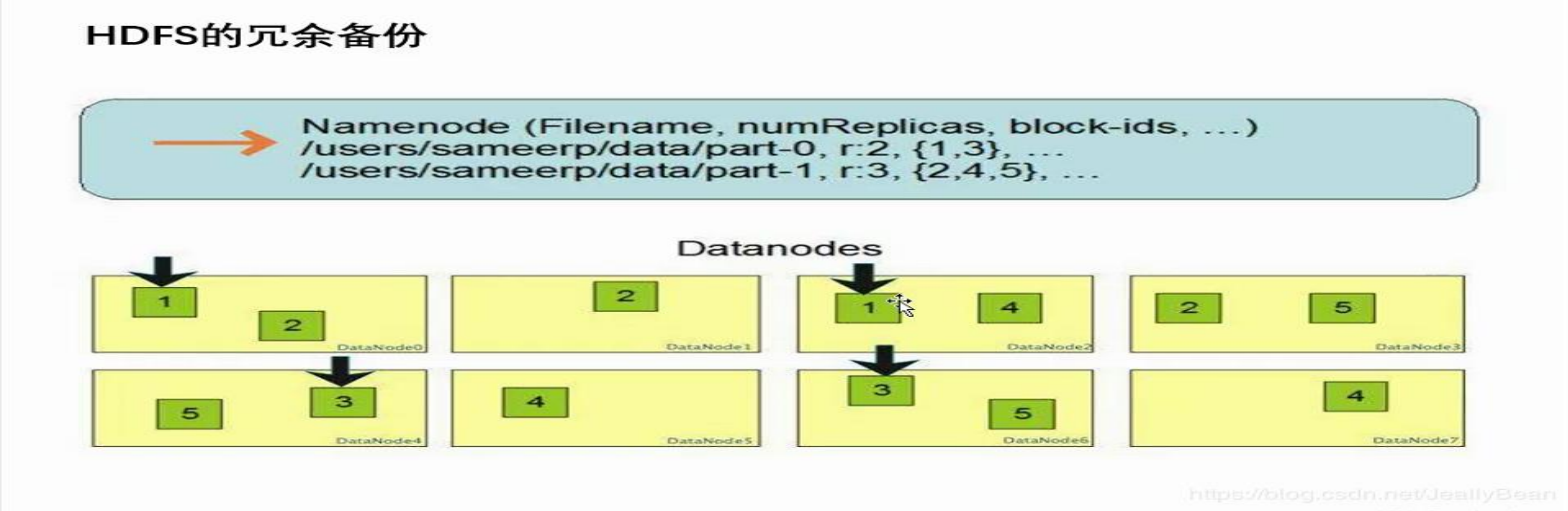
2）接下来根据位置列表知道存储数据块的 Datanode。

3）最后访问 Datanode 获取数据。

注意：Namenode 并不参与数据实际传输。

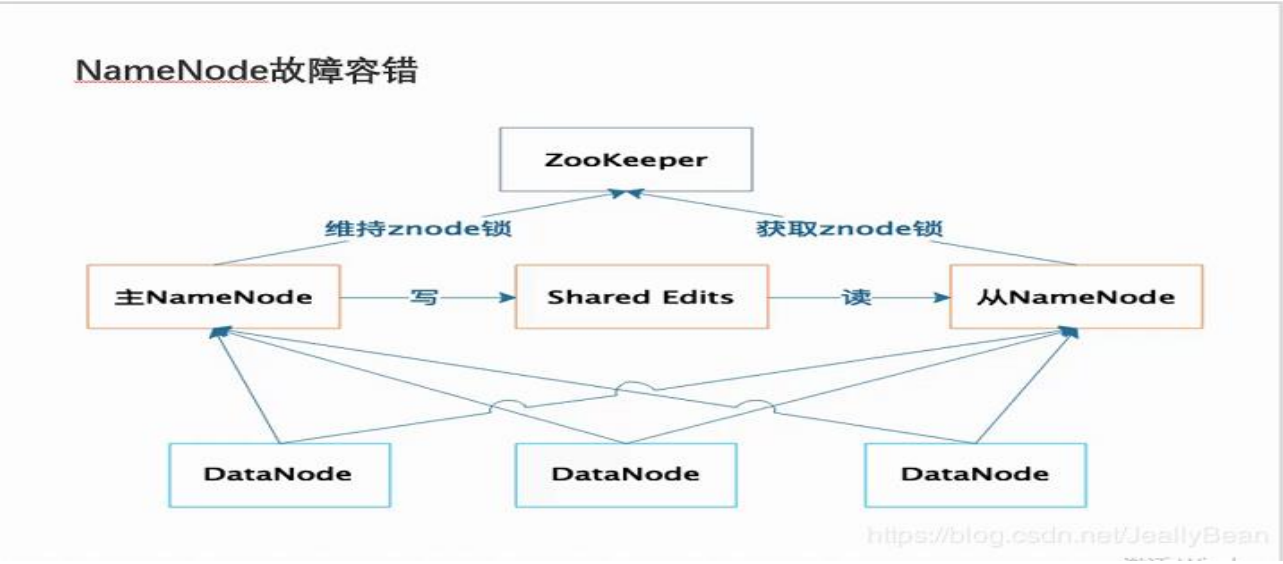
数据存储系统，数据存储的可靠性至关重要。HDFS 是如何保证其可靠性呢？它主要采用如下机理:

1）冗余副本策略，即所有数据都有副本，副本的数目可以在 hdfs-site.xml 中设置相应的复制因子。



2）机架策略，即 HDFS 的“机架感知”，一般在本机架存放一个副本，在其它机架再存放别的副本，这样可以防止机架失效时丢失数据，也可以提供带宽利用率。

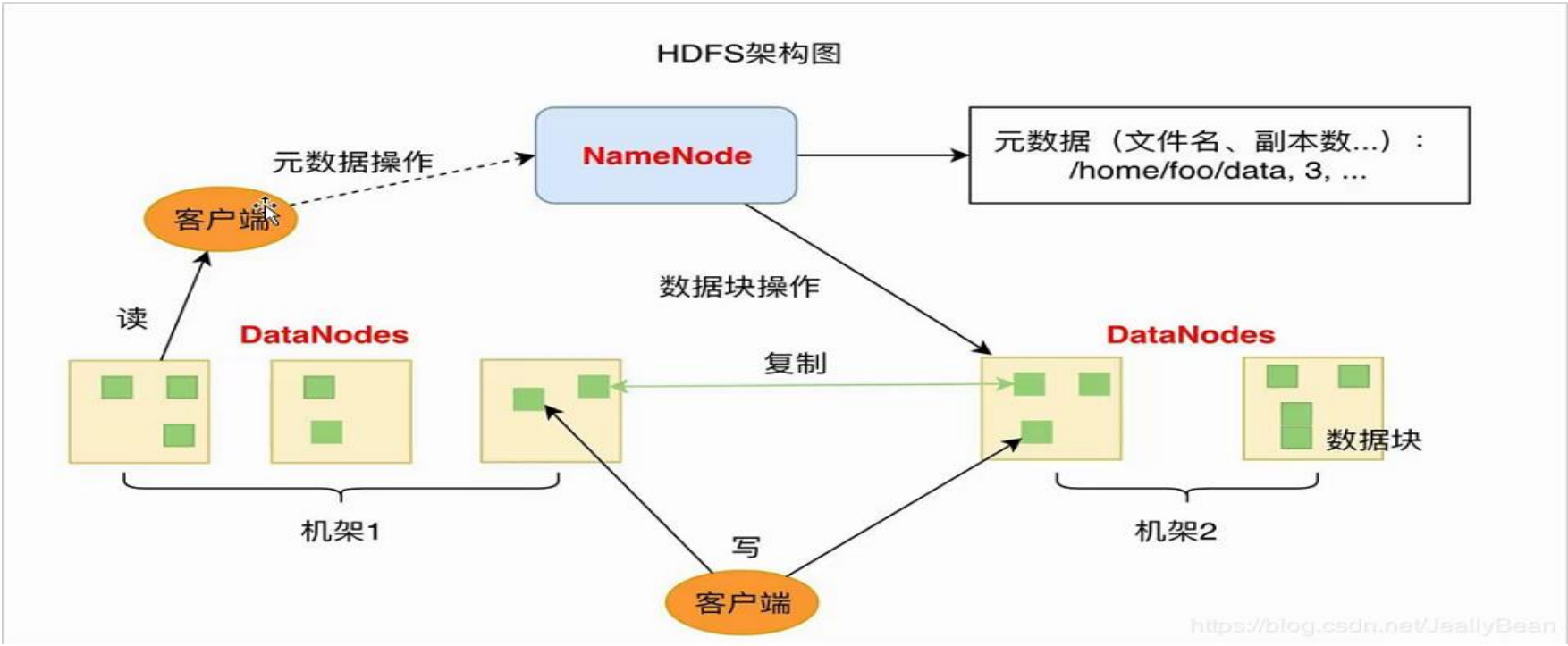
3）心跳机制，即 Namenode 周期性从 Datanode 接受心跳信号和快报告，没有按时发送心跳的 Datanode 会被标记为宕机，不会再给任何 I/O 请求，若是 Datanode 失效造成副本数量下降，并且低于预先设置的阈值，Namenode 会检测出这些数据块，并在合适的时机进行重新复制。

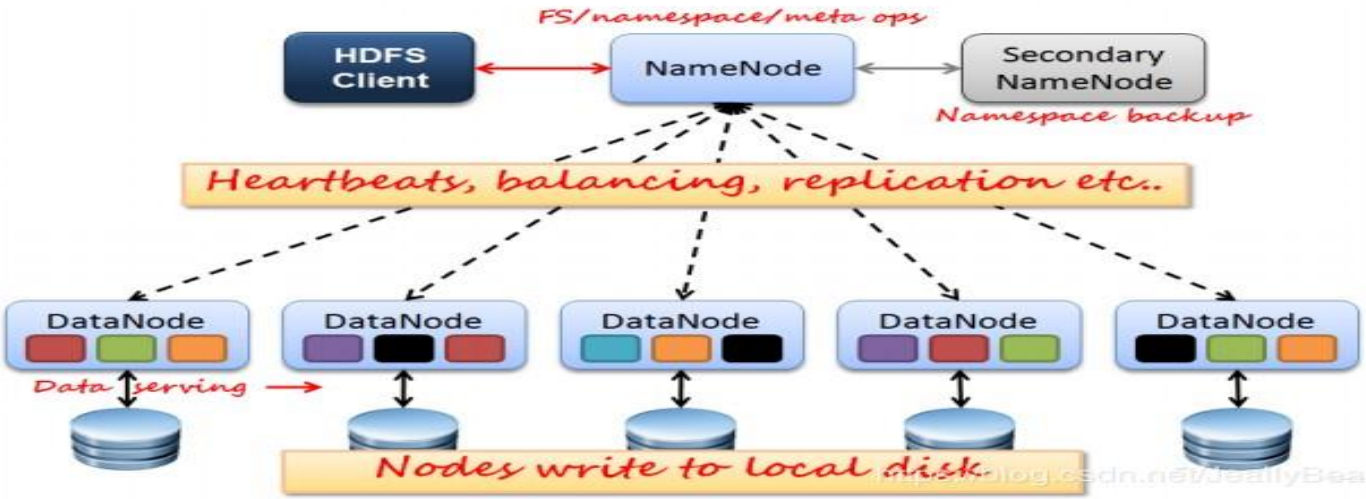


4）安全模式，Namenode 启动时会先经过一个“安全模式”阶段。

5）校验和，客户端获取数据通过检查校验和，发现数据块是否损坏，从而确定是否要读取副本。
6）回收站，删除文件，会先到回收站/trash，其里面文件可以快速回复。
7）元数据保护，映像文件和事务日志是 Namenode 的核心数据，可以配置为拥有多个副本。

8）快照，支持存储某个时间点的映像，需要时可以使数据重返这个时间点的状态。





如上图所示，HDFS 也是按照 Master 和 Slave 的结构。分 NameNode、SecondaryNameNode、DataNode 这几个角色。

- NameNode：是 Master 节点，是大领导。管理数据块映射；处理客户端的读写请求；配置副本策略；管理 HDFS 的名称空间；
- SecondaryNameNode：是一个小弟，分担大哥 namenode 的工作量；是 NameNode 的冷备份；合并 fsimage 和 fsedits 然后再发给 namenode。
- DataNode：Slave 节点，奴隶，干活的。负责存储 client 发来的数据块 block；执行数据块的读写操作。
- 热备份：b 是 a 的热备份，如果 a 坏掉。那么 b 马上运行代替 a 的工作。
- 冷备份：b 是 a 的冷备份，如果 a 坏掉。那么 b 不能马上代替 a 工作。但是 b 上存储 a 的一些信息，减少 a 坏掉之后的损失。
- fsimage:元数据镜像文件（文件系统的目录树。）
- edits：元数据的操作日志（针对文件系统做的修改操作记录）
- namenode 内存中存储的是=fsimage+edits。
- SecondaryNameNode 负责定时默认 1 小时，从 namenode 上，获取 fsimage 和 edits 来进行合并，然后再发送给 namenode。减少 namenode 的工作量。

Hadoop 核心之 MapReduce

上部分提到 Hadoop 存储大数据的核心模块 HDFS，这一部分介绍 Hadoop 处理大数据部分的核心模块 MapReduce。
Apache Foundation 对 MapReduce 的介绍： “Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.”
由此可知，Hadoop 核心之 MapReduce 是一个软件框架，基于该框架能够容易地编写应用程序，这些应用程序能够运行在由上千个商用机器组成的大集群上，并以一种可靠的，具有容错能力的方式并行地处理上 TB 级别的海量数据集。这个定义里面有着这些关键词，一是软件框架，二是并行处

理，三是可靠且容错，四是大规模集群，五是海量数据集。因此，对于 MapReduce，可以简洁地认为，它是一个软件框架，海量数据是它的“菜”，它在大规模集群上以一种可靠且容错的方式并行地“烹饪这道菜”。

MapReduce 主要是用于解决 Hadoop 大数据处理的。所谓大数据处理，即以价值为导向，对大数据加工、挖掘和优化等各种处理。MapReduce 擅长处理大数据，它为什么具有这种能力呢？这可由 MapReduce 的设计思想发觉。MapReduce 的思想就是“分而治之”。Mapper 负责“分”，即把复杂的任务分解为若干个“简单的任务”来处理。“简单的任务”包含三层含义：一是数据或计算的规模相对原任务要大大缩小；二是就近计算原则，即任务会分配到存放着所需数据的节点上进行计算；三是这些小任务可以并行计算，彼此间几乎没有依赖关系。Reducer 负责对 map 阶段的结果进行汇总。至于需要多少个 Reducer，用户可以根据具体问题，通过在 mapred-site.xml 配置文件里设置参数 mapred.reduce.tasks 的值，缺省值为 1。

MapReduce 的工作机制如图所示：

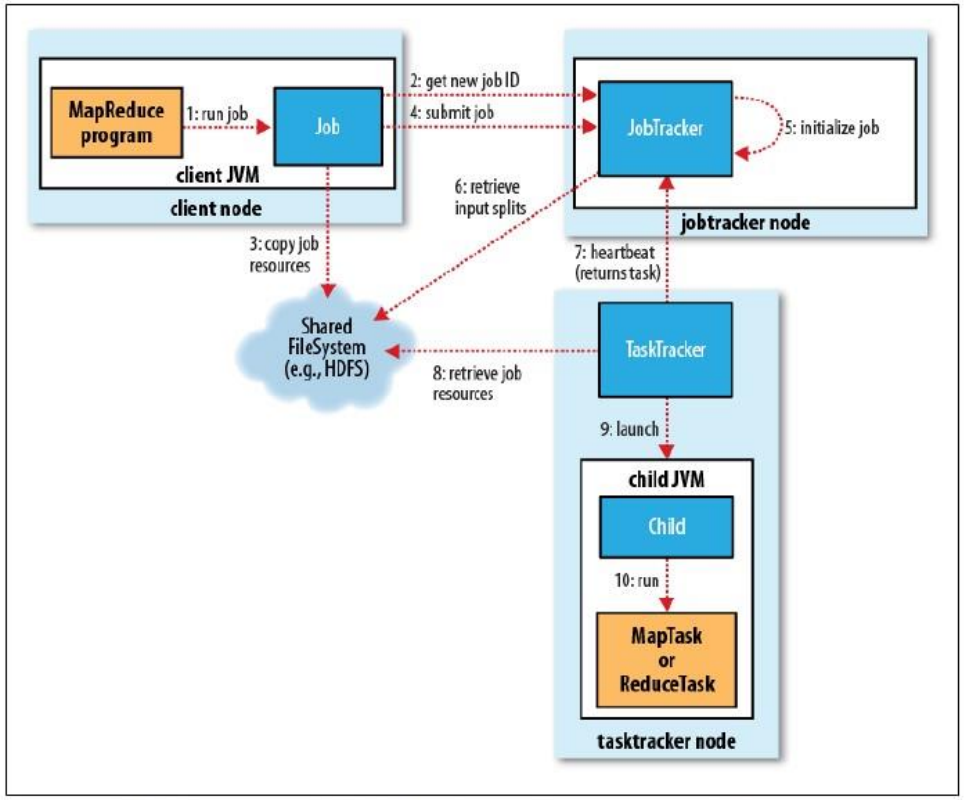


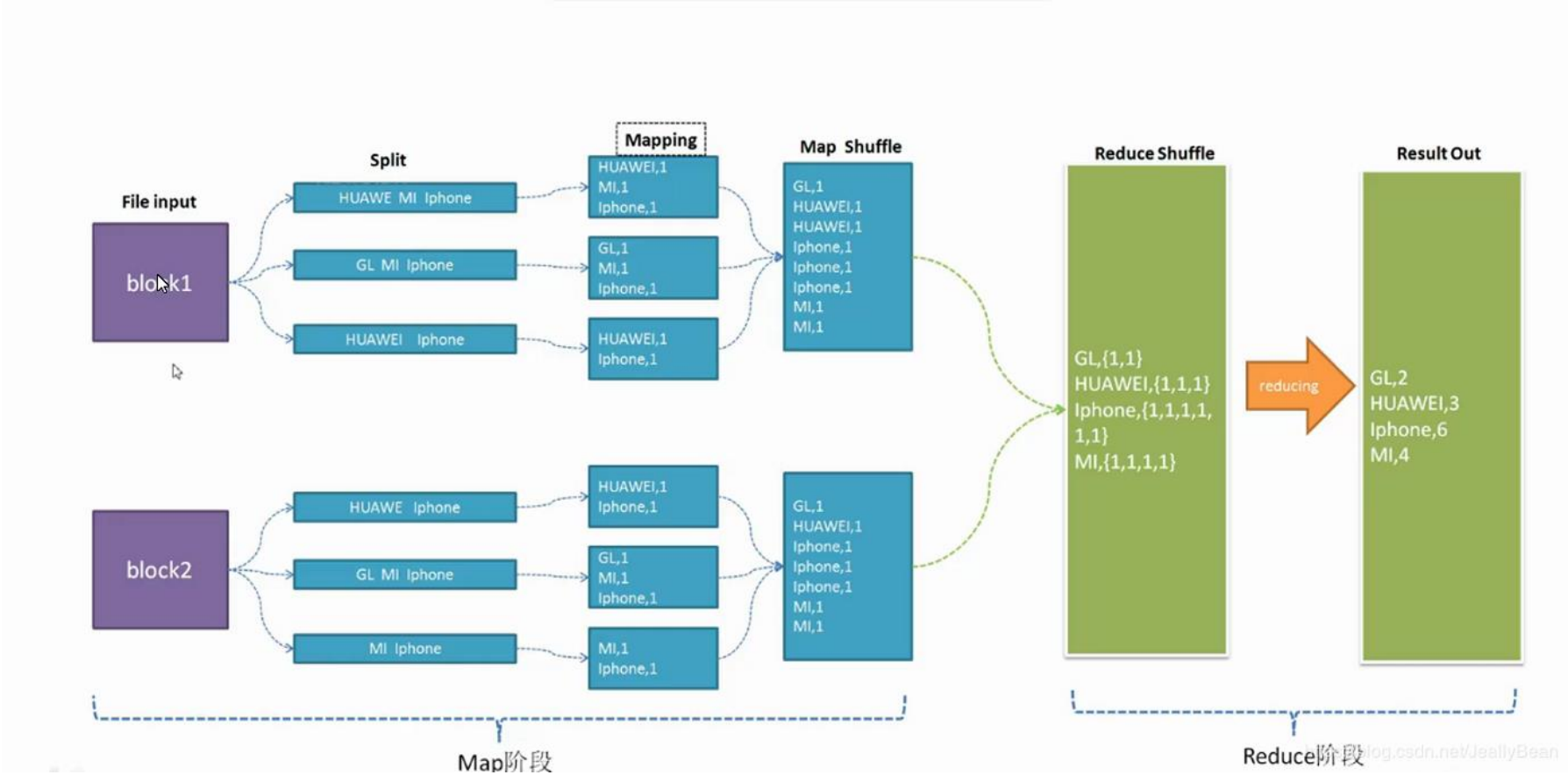
Figure 6-1. How Hadoop runs a MapReduce job using the classic framework

MapReduce 的整个工作过程如上图所示，它包含如下 4 个独立的实体：

1）客户端，用来提交 MapReduce 作业（程序员编写的运行在 MapReduce 上的应用程序称为作业（job））。

2）jobtracker，用来协调作业的运行。
3）tasktracker，用来处理作业划分后的任务。
4）HDFS，用来在其它实体间共享作业文件。
MapReduce 整个工作过程有序地包含如下工作环节：
1）作业的提交
2）作业的初始化
3）任务的分配
4）任务的执行
5）进程和状态的更新
6）作业的完成

- 分阶段展示



Hadoop 核心之 Yarn

YARN 通过两种长期运行的守护进程提供其核心服务：一个资源管理器(resource manager, one per cluster) 管理集群资源的使用; 运行于集群内所有节点上的
节点管理器(node managers) 用于启动和监视容器(container). 一个容器使用一系列受限的资源(内存，CPU，等)执行应用程序特定的进程。依赖于 YARN 是如何
配置的，一个容器可能是一个 Unix 进程，或者一个 Linux cgroup.
在 YARN 上运行一个应用程序，客户端联系 resource manager 并请求它运行一个 application master 进程。 resource manager 找到一个能够在一个容器内运行
application master 的 node manager. 一旦 application master 运行起来它到底能做什么完全取决于应用程序。它可以简单地在它运行的容器内运行一个计算并
将结果返回给客户端。或者，它从 resource manager 请求更多的容器，并使用它们运行一个分布式计算。后面就是 MapReduce YARN 应用程序做的事情了。
YARN 本身并不为应用程序各个部分(client, master, process) 之间提供任何彼此通信的方法。很多重要的 YARN 应用程序使用一些远程通信机制(例如 Hadoop 的 RPC 层)来将状态更新和结果传递回给客户端，但这些是应用程序特定的。
原文：https://blog.csdn.net/devalone/article/details/80679872