中国传媒大学
COMMUNICATION UNIVERSITY OF CHINA

数据科学导论第 14 讲—— 并行、分布式和云计算

王小宁

中国传媒大学数据科学与智能媒体学院

2023 年 06 月 19 日

目录

大数据时代/人工智能时代

背景

- 最早提出” 大数据” 时代到来的是全球知名咨询公司麦肯锡，麦肯锡称:“数据，已经渗透到当今每一个行业和业务职能领域，成为重要的生产因素。人们对于海量数据的挖掘和运用，预示着新一波生产率增长和消费者盈余浪潮的到来。”
- “大数据” 在物理学、生物学、环境生态学等领域以及军事、金融、通讯等行业存在已有时日，却因为近年来**互联网和信息行业**的发展而引起人们关注。
- 简单来说：大量数据 + 云计算 = 大数据时代

大数据特点

1. 数据量大 (Volume)：大数据的起始计量单位至少是 P(1000 个 T)、E(100 万个 T) 或 Z(10 亿个 T)。
2. 类型繁多 (Variety)，包括网络日志、音频、视频、图片、地理位置信息等等，多类型的数据对数据的处理能力提出了更高的要求。
3. 价值密度低 (Value)，如随着物联网的广泛应用，信息感知无处不在，信息海量，但价值密度较低，如何通过强大的机器算法更迅速地完成数据的价值" 提纯"，是大数据时代亟待解决的难题。
4. 速度快、时效高 (Velocity)，这是大数据区分于传统数据挖掘最显著的特征。

人工智能

- 人工智能的研究目的就是让机器具有人的智能。
- 人工智能其实是起源于国外的概念，英文为“Artificial Intelligence”(AI)，意思是“人造的智能”。
- 这里的“智能”，指的是人类大脑所具有的一些诸如意识、思维、感知、学习等高级功能，人工智能就是指应用计算机的软硬件来模拟或实现人类的某些智能行为的技术。

未来展望

- 根据 IBM 的估计，到 2020 年，数据专业人员的职位数量将增加 270 万。有能力通过 AI 实施处理大量数据的候选人的需求量很大。
- IBM 本身认为，数据专业人员的作用将很高，而像高级分析师和数据科学家这样的人将增加 28%。
- 医疗：AI 使医学研究人员得以实现许多我们认为无法实现的事情。未来几年，医疗领域对 AI 聊天机器人开发服务的需求将会激增。
- 语音：智能家居，小度，小爱，天猫精灵与智能穿戴设备。
- 自然语言：自动翻译，情感分析，问答系统，查重，自动摘要，自动写稿、拼写检查等。
- 图像：智能锁，人脸识别，场景渲染等。
- 视频：视频交互、视频修复、目标检测和图像分割等。

并行计算

并行计算

- 并行计算，是指同时使用多种计算资源解决计算问题的过程，是提高计算机系统计算速度和处理能力的一种有效手段，R 的并行计算主要通过它内置的并行任务包来实现。
- 它的基本思想是用多个处理器来协同求解同一问题，即将被求解的问题分解成若干个部分，各部分均由一个独立的处理机来并行计算。
- 并行计算又分为数据并行和任务并行两种，R 的并行计算主要通过它内置的任务并行包来实现，包括 parallel 包和 foreach 包。

并行计算的基本思路

1. 建立 M 个工作节点，对每个进程做基本的初始化
2. 将所需数据传输到每个工作节点
3. 将任务分割成 M 份传输到各个工作节点进行计算
4. 合并整理 M 个工作节点的结果
5. 重复上述过程进行进一步的计算
6. 关闭工作节点

parallel 包

- 在进行并行计算前，首先需要了解计算机或者服务器的性能。
- R 提供了 detectCores() 来查看计算机的逻辑核心数，detectCores (logical = FALSE) 来查看计算机的物理核心数。

```
library(parallel)
detectCores();detectCores(logical = FALSE)
```

```
## [1] 12
```

```
## [1] 6
```

- 逻辑核心一般会大于其物理核心，是采用多线程方式对物理核心处理器的性能提升。该函数只是显示的计算机的可用核心，并不代表执行当前任务时可用核心数。
- parallel 包的思路和 lapply() 函数类似，都是将输入数据分割成几份，然后计算并把结果整合在一起，只不过，parallel 并行计算是用不同的 CPU 来运算。

并行计算步骤

❶ 设置需要调用的工作节点，如调用两个节点进行并行计算：

```
cl <- makeCluster(getOption("cl.cores",2))
```

❷ 采用不同的函数进行并行计算

3. 关闭开辟的节点：

```
stopCluster(cl)
```

举例

```
cl <- makeCluster(getOption("cl.cores",2))
xx <- 1 : 2
yy <- 3 : 4
clusterExport(cl , c ( "xx" , "yy"))
clusterCall(cl , function ( z ) xx + yy + sin(z), pi)

## [[1]]
## [1] 4 6
##
## [[2]]
## [1] 4 6

stopCluster(cl)
```

- clusterApply(cl = NULL , x , fun , ..)，这个函数表示将向量 x 的每一个值带入函数 fun 进行计算。
- 例如 clusterApply (cl , 1 : 2 , get (“+”) , 3)，其中 cl 为集群数，1:2 为输入变量值，“+” 为即将进行的计算，3 为 “+” 函数的另一参数

```
# fxy <- function(x,y){
#       x^2 + sin(y) - 1
# }
#
# Z <- clusterApply(cl,1:10,fxy,y = 2)
# Z[[1]]
```

定义函数

```
M <- matrix ( rnorm (50000000),100,500000)
Mysort <- function(x){
          return(sort(x)[1:10])
  }
do_apply <- function ( M )
{
  return(apply(M,2, Mysort))
}

do_parallel <- function(M,ncl)
{
      cl <- makeCluster(getOption("cl.cores" ,ncl))
      ans <- parApply ( cl,M,2,Mysort)
      stopCluster (cl)
      return(ans)
}
```

结果

```
system.time(ans <- do_apply(M))
```

```
##    user  system elapsed
##  30.914   0.523  31.739
```

```
system.time(ans2 <- do_parallel(M,2))
```

```
##    user  system elapsed
##   3.334   1.135  19.425
```

foreach 包与 doParallel 包

- doParalle 多与 foreach 包一同使用，doParallel 包使 foreach 并行计算成为可能。
- foreach 的功能类似于 for 或者 lapply 函数，其最大的好处在于代码简单而且容易采用并行方式进行计算。
- foreach 可进行并行或串行计算，为了提示计算机 foreach 将采用并行计算的方式，在开辟工作节点的基础上，需声明并行计算将要用到的节点数。声明方式如下：

```
library ( foreach )
library ( doParallel )

## Loading required package: iterators

cl <- makeCluster ( getOption ( "cl.cores" , 2 ) )
registerDoParallel ( cl )
#.....
stopCluster ( cl )
```

HPC 多线程并行计算

- HPC 是高性能计算 (High Performance Computing) 机群的简称。
- 指能够执行一般个人电脑无法处理的大资料量与高速运算的电脑，其基本组成组件与个人电脑的概念无太大差异，但规格与性能则强大许多。现有的超级计算机运算速度大都可以达到每秒一兆（万亿，非百万）次以上。
- 高性能计算集群依赖于并行处理系统，所以高性能计算集群信息需要快速的传入与传出内存。高性能计算集群系统往往是 I/O 密集型的，因此高性能计算集群选择正确的内存配置，可以显著提升高性能计算集群应用程序性能。

分布式计算

分布式计算

- 分布式计算是一种计算方法，和集中式计算是相对的。
- 随着计算技术的发展，有些应用需要非常巨大的计算能力才能完成，如果采用集中式计算，需要耗费相当长的时间来完成。
- 分布式计算将该应用分解成许多小的部分，分配给多台计算机进行处理。这样可以节约整体计算时间，大大提高计算效率。

定义-中科院

- 分布式计算比起其它算法具有以下几个优点：

1. 稀有资源可以共享
2. 通过分布式计算可以在多台计算机上平衡计算负载
3. 可以把程序放在最适合运行它的计算机上

- 其中，共享稀有资源和平衡负载是计算机分布式计算的核心思想之一。

网格计算和分布式计算架构

- 网格计算就是分布式计算的一种。
- 如果某项工作是分布式的，那么，参与这项工作的一定不只是一台计算机，而是一个计算机网络，这种“蚂蚁搬山”的方式将具有很强的数据处理能力。
- 网格计算的实质就是组合与共享资源并确保系统安全。
- 分布式计算是利用网络把成千上万台计算机连接起来，组成一台虚拟的超级计算机，完成单台计算机无法完成的超大规模的问题求解。
- 开放分布式计算架构是指以分布式计算技术为基础，用于解决大规模的问题开放式软件架构。开放分布式计算架构具有较好的可移植性和可裁剪性。

Hadoop

- Hadoop 是一款支持数据密集型分布式应用程序并以 Apache 2.0 许可协议发布的开源软件框架。它支持在商品硬件构建的大型集群上运行的应用程序。
- Hadoop 是根据谷歌公司发表的 MapReduce 和 Google 文件系统的论文自行实现而成。
- 所有的 Hadoop 模块都有一个基本假设，即硬件故障是常见情况，应该由框架自动处理。
- Hadoop 框架透明地为应用提供可靠性和数据移动。它实现了名为 MapReduce 的编程范式：应用程序被分区成许多小部分，而每个部分都能在集群中的任意节点上运行或重新运行。

基本组成

- Hadoop 还提供了分布式文件系统 (HDFS)，用以存储所有计算节点的数据，这为整个集群带来了非常高的带宽。
- MapReduce 和分布式文件系统的设计，使得整个框架能够自动处理节点故障。它使应用程序与成千上万的独立计算的电脑和 PB 级的数据。
- 现在普遍认为整个 Apache Hadoop“平台” 包括 Hadoop 内核、MapReduce、Hadoop 分布式文件系统（HDFS）以及一些相关项目，有 Apache Hive 和 Apache HBase 等等。

Spark

- Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。Spark 是 UC Berkeley AMP lab (加州大学伯克利分校的 AMP 实验室) 所开源的类 Hadoop MapReduce 的通用并行框架，Spark，拥有 Hadoop MapReduce 所具有的优点；但不同于 MapReduce 的是 Job 中间输出结果可以保存在内存中，从而不再需要读写 HDFS，因此 Spark 能更好地适用于数据挖掘与机器学习等需要迭代的 MapReduce 的算法。
- Spark 是在 Scala 语言中实现的，它将 Scala 用作其应用程序框架。与 Hadoop 不同，Spark 和 Scala 能够紧密集成，其中的 Scala 可以像操作本地集合对象一样轻松地操作分布式数据集。

HDFS 架构概述

- 在 “经典”HDFS 有三个守护进程

1. NameNode(主)
2. Secondary NameNode(主)
3. DataNode(从)

MapReduce

- MapReduce 是一个编程模型

1. 既不是平台也不基于特定于语言
2. 面向记录的数据处理 (键和值)

3. 多节点共同处理一个任务

- 在可能的情况下，每个节点处理存储在各自节点上的数据，包括两个阶段：

1. Map
2. Reduce
3. 在 Map 和 Reduce 之间是 shuffle 和 sort 阶段：从 Mapper 向 Reducer 发送数据

云计算

云计算

- 云计算（Cloud Computing）：是分布式处理 (Distributed Computing)、并行处理 (Parallel Computing) 和网格计算 (Grid Computing) 的发展，或者说是这些计算机科学概念的商业实现。
- 云计算是指基于互联网的超级计算模式–即把存储于个人电脑、移动电话和其他设备上的大量信息和处理器资源集中在一起，协同工作。在极大规模上可扩展的信息技术能力向外部客户作为服务来提供的一种计算方式。

特点

- 数据在云端：不怕丢失，不必备份，可以任意点的恢复；
- 软件在云端：不必下载自动升级；
- 无所不在的计算：在任何时间，任意地点，任何设备登录后就可以进行计算服务；
- 无限强大的计算：具有无限空间的，无限速度。

服务方式

- SAAS（Software as a Service ）- 各类的网盘 (Dropbox、百度网盘等)，JIRA，GitLab 等
- PAAS（Platform as a Service ）- 数据库服务、web 应用以及容器服务
- IAAS（Infrastructure as a Service ）- 虚拟机、虚拟网络
- 云存- OSS

体系结构

- 云计算的基本原理是通过使计算分布在大量的分布式计算机上，而非本地计算机或远程服务器中，企业数据中心的运行将更与互联网相似。这使得企业能够将资源切换到需要的应用上，根据需求访问计算机和存储系统。
- 通过 Internet 接入
- 不需要自身具有 IT 技术来实施
- 第三方提供
- 资源共享
- 无多余功能开发
- 无多余费用
- 系统延续性好

云计算厂商

- AWS
- Azure
- 阿里云
- 腾讯云

谢　谢!