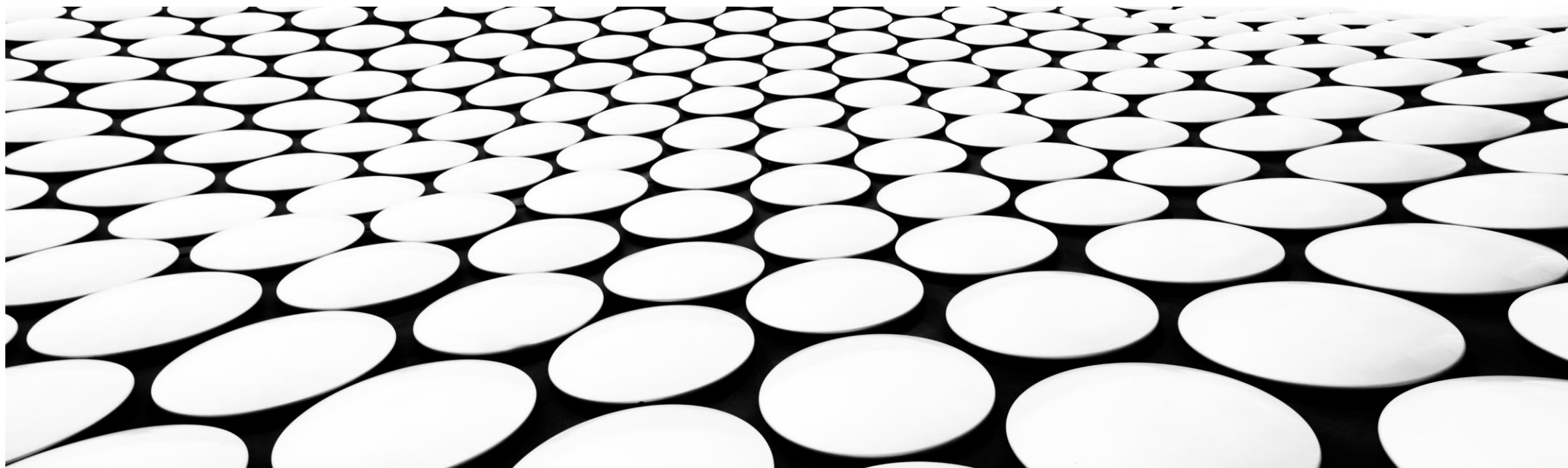


分布式计算

邱怡轩



今天的主题

- 并行计算基本概念
- 并行计算与分布式计算
- Apache Spark简介与安装



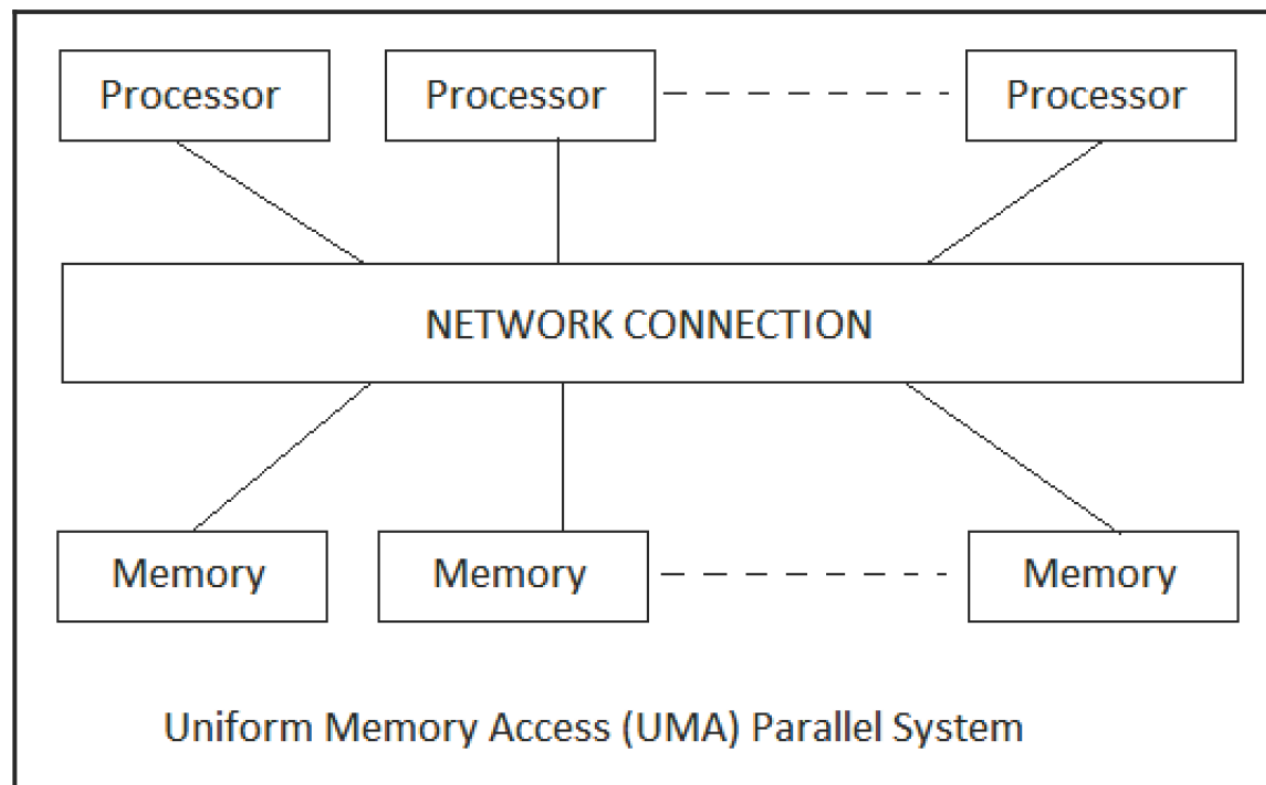
并行计算

概念

- 一种计算模型
- 将总任务分成若干子任务
- 每个子任务同时执行
- 主要目的是提升计算效率

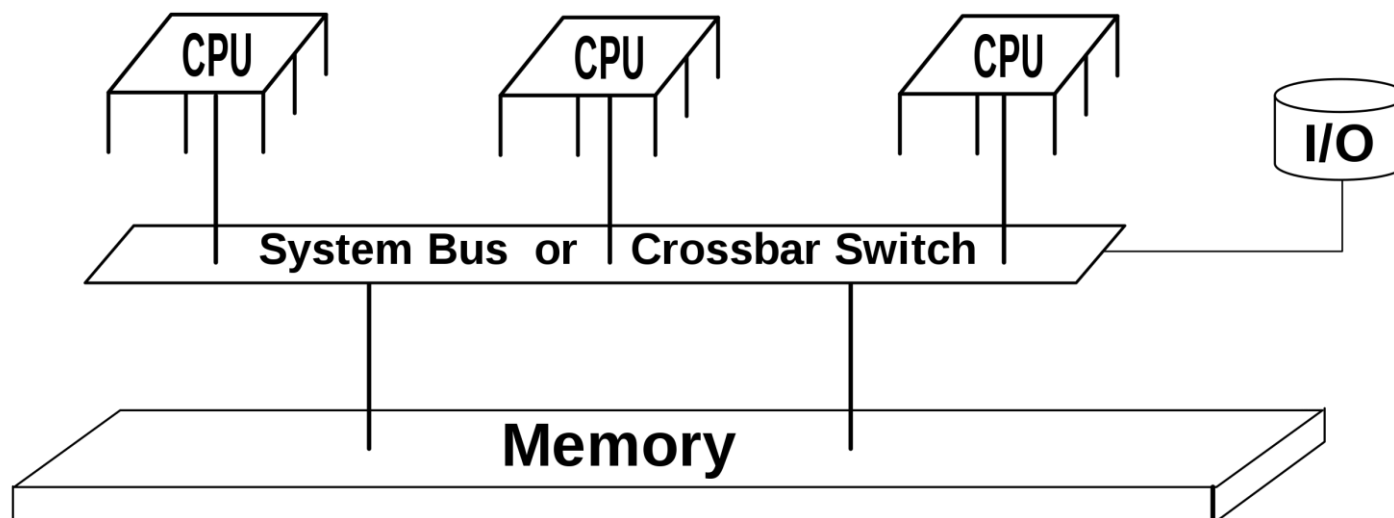
架构

- 一类重要的用来实现并行计算的架构叫做统一内存访问架构
- (Uniform Memory Access, UMA)



架构

- 例如我们日常使用的多核个人电脑或多CPU服务器



架构

- 狭义的并行计算通常指代这类由共享内存机制实现的计算

Amdahl定律

- 用于预测并行系统的加速比

$$S_{\text{latency}}(s) = \frac{1}{(1 - p) + \frac{p}{s}}$$

- S: 整体提升倍数
- s: 可并行部分的加速比
- p: 可被并行部分所占时间的比例

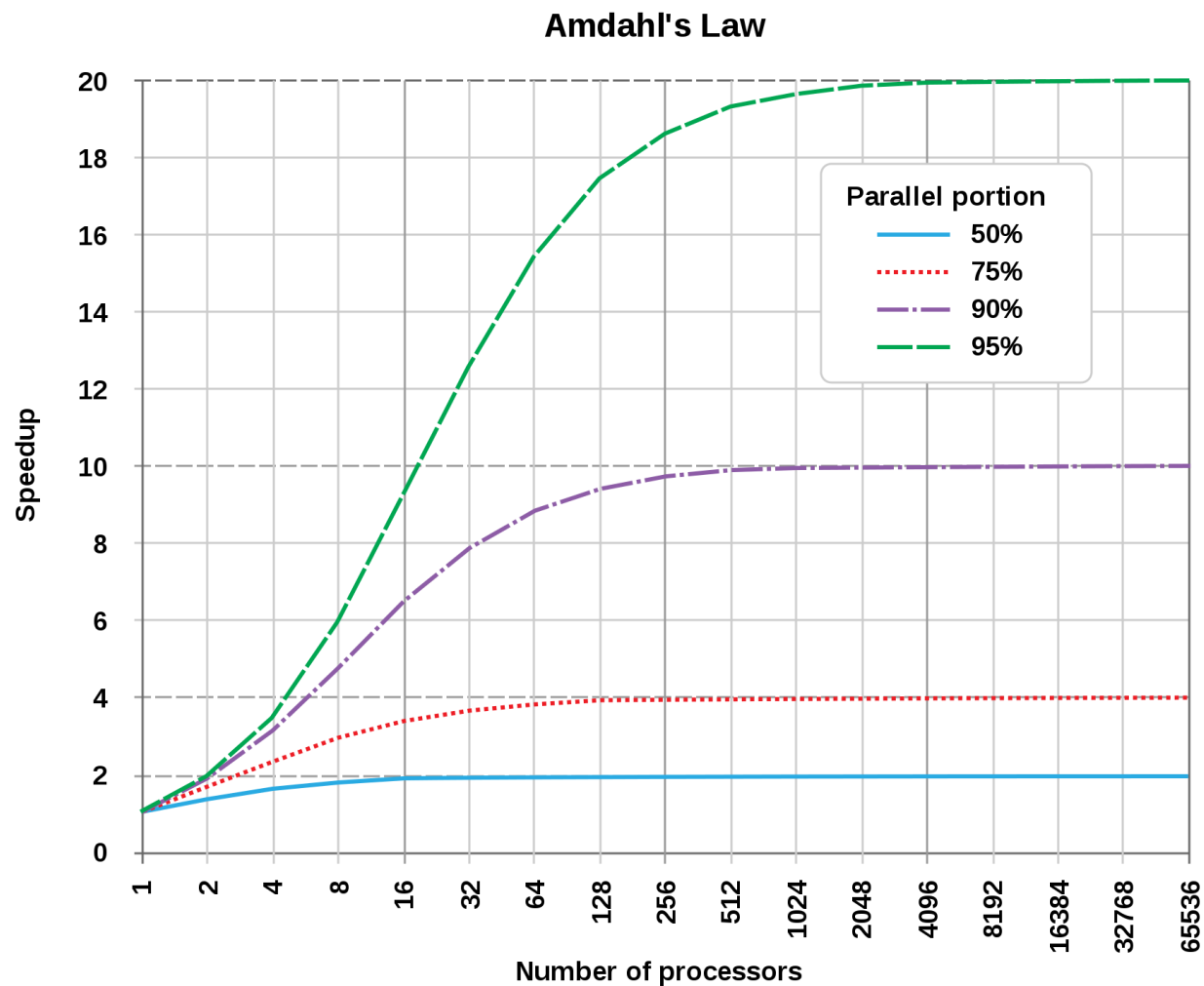
Amdahl定律

- 一个简单例子
- $p_1 = 0.11, p_2 = 0.18, p_3 = 0.23, p_4 = 0.48$
- $s_1 = 1, s_2 = 5, s_3 = 20, s_4 = 1.6$

$$\begin{aligned} S_{\text{latency}} &= \frac{1}{\frac{p_1}{s_1} + \frac{p_2}{s_2} + \frac{p_3}{s_3} + \frac{p_4}{s_4}} \\ &= \frac{1}{\frac{0.11}{1} + \frac{0.18}{5} + \frac{0.23}{20} + \frac{0.48}{1.6}} = 2.19. \end{aligned}$$

Amdahl定律

- 不能被并行的部分决定了并行效果的上限



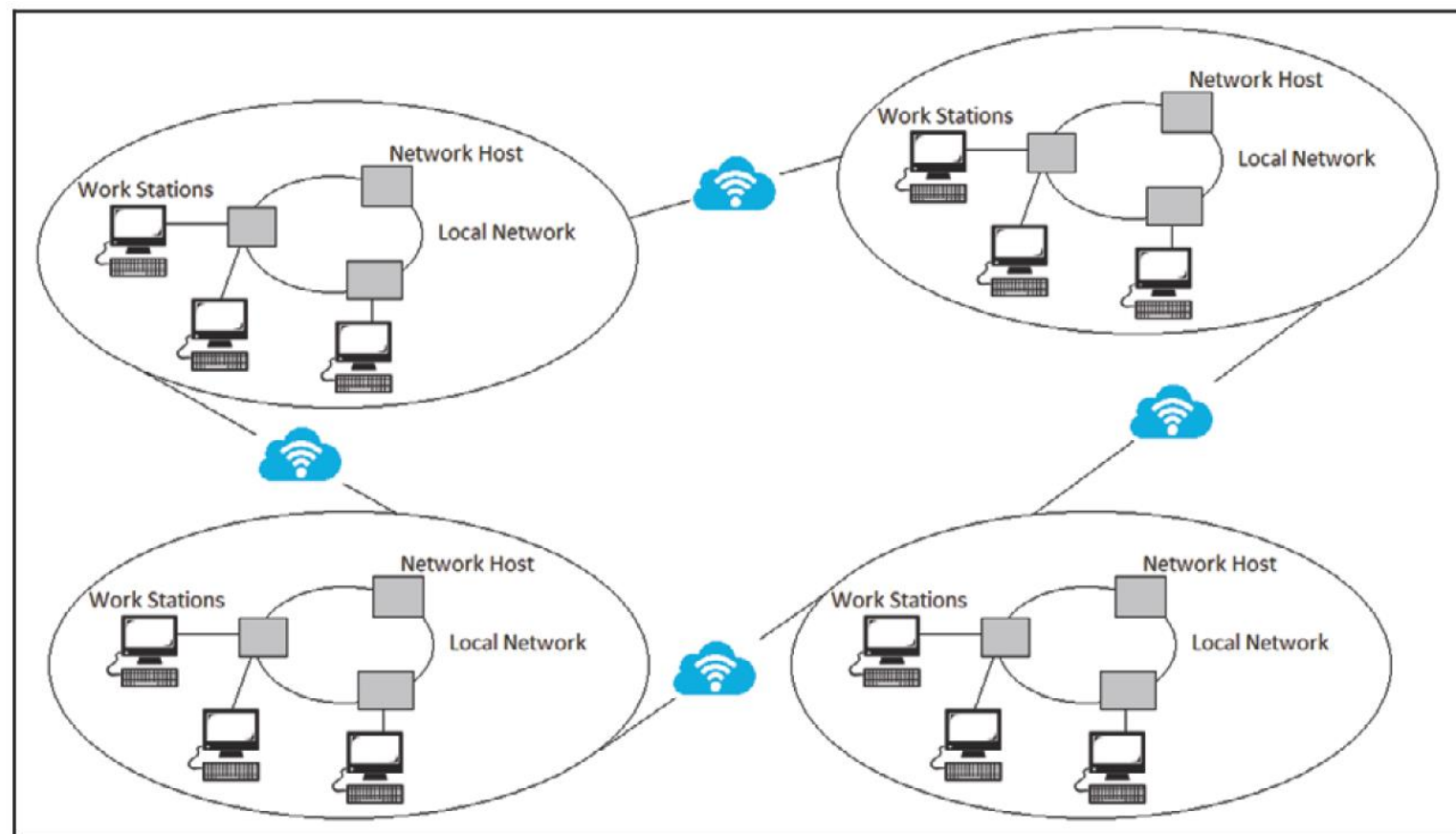


分布式计算

概念

- 利用多台通过网络连接在一起的计算机共同完成某项计算任务
- 与并行计算有许多相通之处
- 但分布式系统中每台机器有独立的内存
- 除了提升计算效率之外，另一重要目的是扩展计算的规模

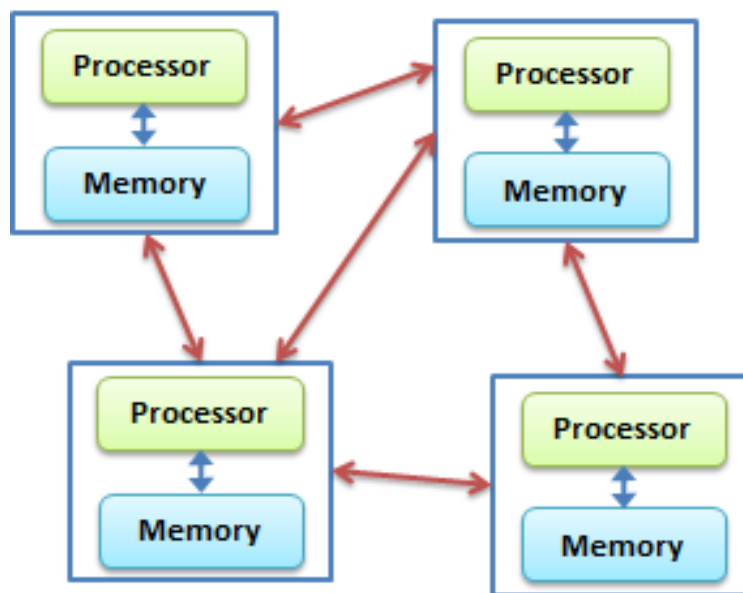
概念



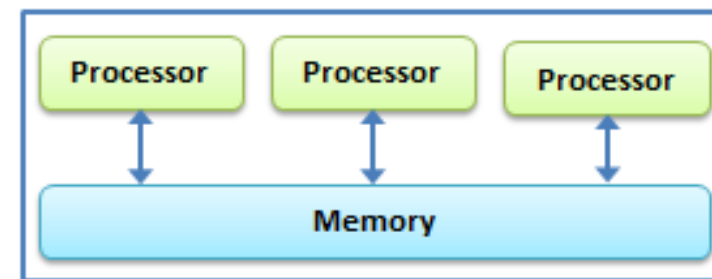
对比

■ 并行计算 vs 分布式计算

Distributed Computing



Parallel Computing



对比

- 从可扩展性的角度来看
- 并行计算：受单机性能制约，如内存大小
- 分布式计算：理论上可以无限进行扩展

实现

- 分布式计算的实现要考虑以下两个问题
- 数据如何存储？
- 计算任务如何执行？



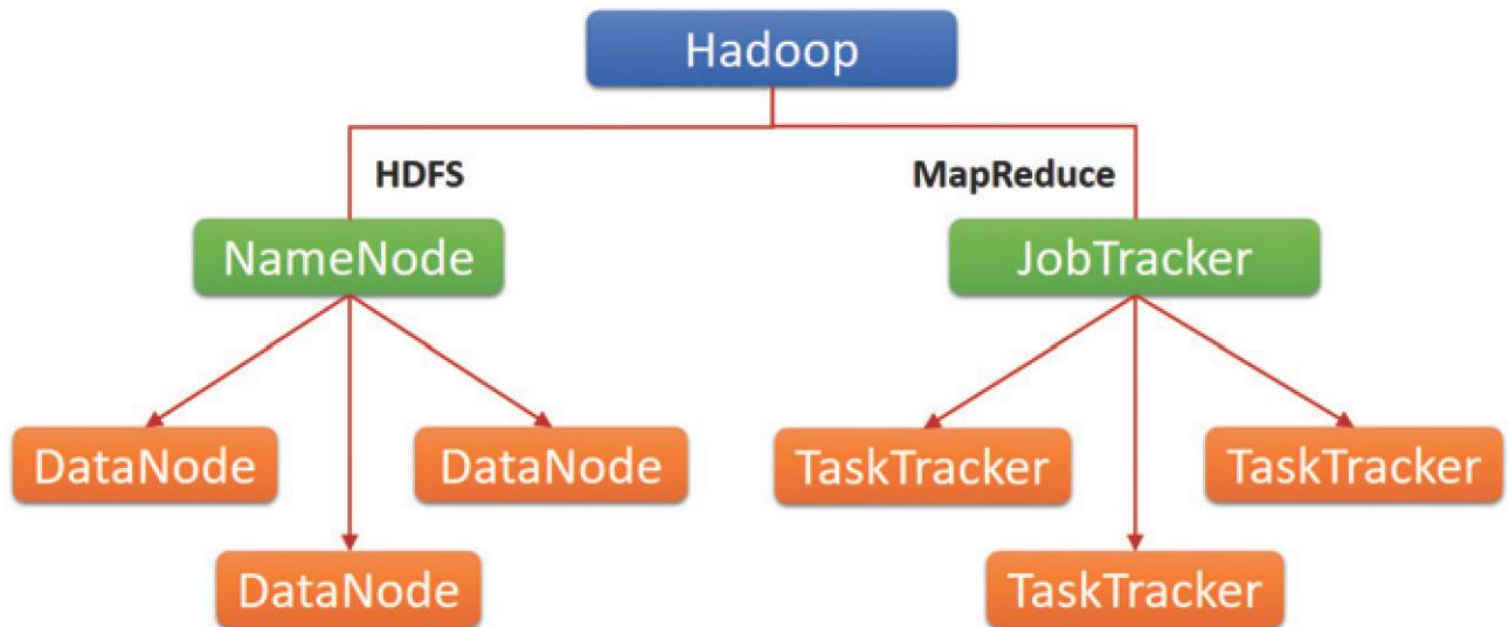
Hadoop

简介

- Hadoop 的官方名称是 Apache Hadoop
- 是由 Apache 软件基金会开发和维护的一套开源分布式计算框架
- 可以扩展到上千台机器组成的集群

Hadoop

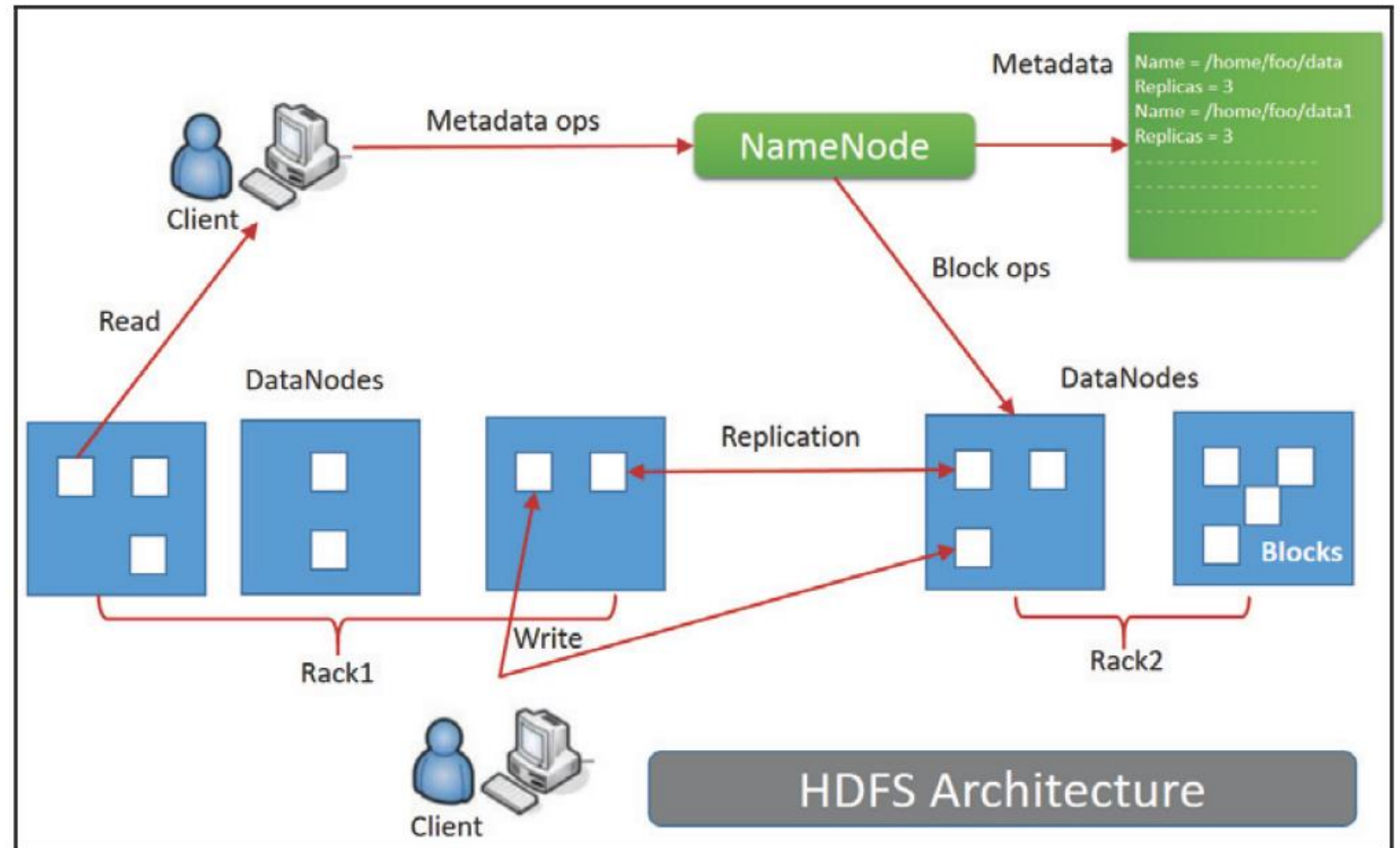
- Hadoop \approx HDFS + MapReduce



HDFS

- HDFS 是 Hadoop 提供的分布式文件系统
- 解决了大规模数据的存储问题
- 实现高容错、低成本的数据存储方案

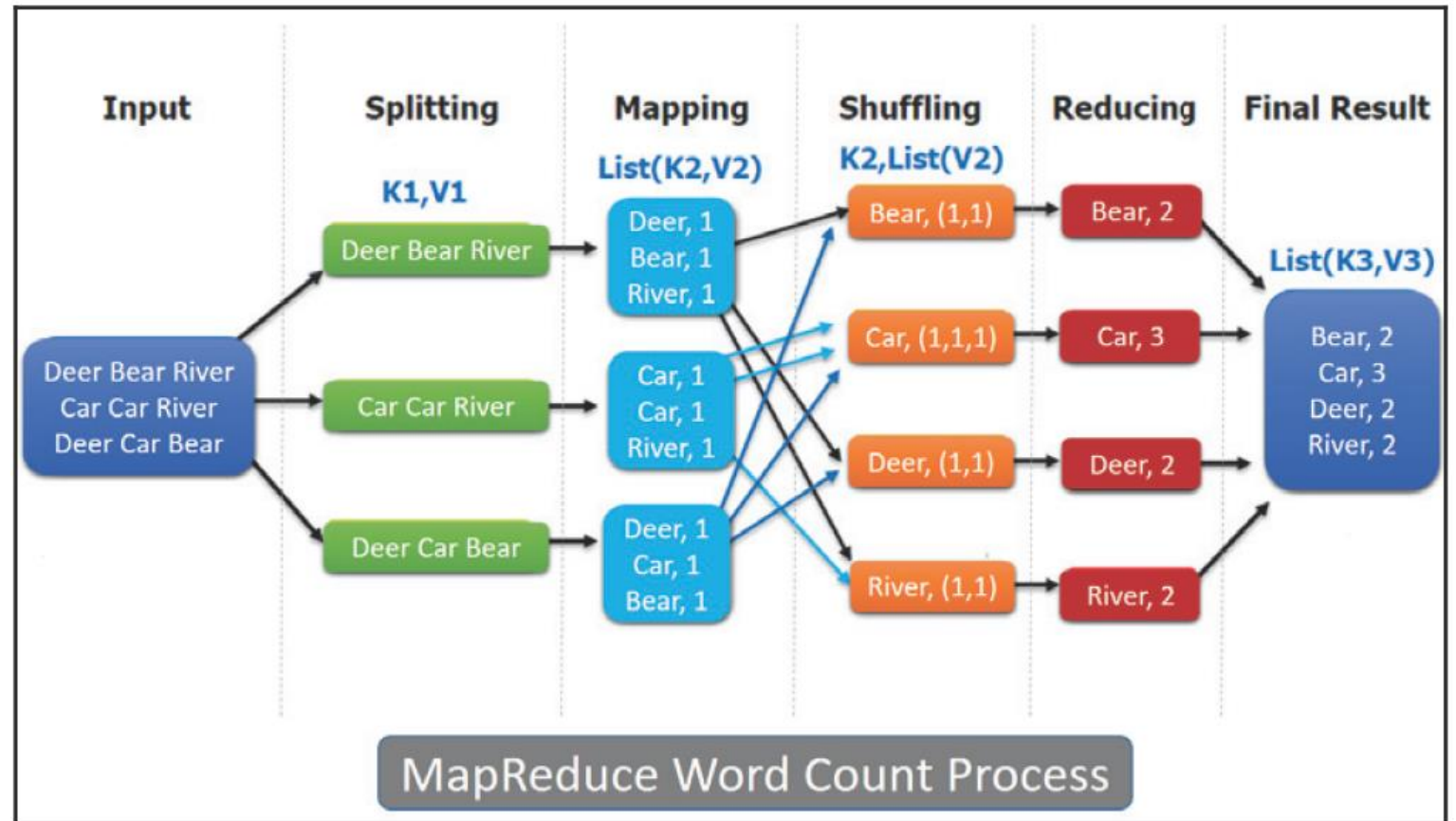
HDFS



MapReduce

- MapReduce 是 Hadoop 采用的计算框架
- 把数据处理过程分成两个阶段：Map 和 Reduce
- Map 用来对输入数据进行变换
- Reduce 用来汇总结果

MapReduce



不足

- Hadoop 对于分布式计算有着重大意义
- 但也存在一些不足
- 表达能力较弱（一些操作难以仅用 map 和 reduce 完成）
- 不容易实现迭代计算
- 基于磁盘进行数据传递，效率较低
- 交互性不强



Spark

简介

- 与 Hadoop 类似, Spark的“全名”叫 Apache Spark
- 是由 Apache 软件基金会开发和维护的一套开源大数据分析平台
- 针对 Hadoop 的缺陷进行了多项改进

工具集

Structured
Streaming

Advanced
Analytics

Libraries &
Ecosystem

Structured APIs

Datasets

DataFrames

SQL

Low-level APIs

RDDs

Distributed Variables

编程语言

- Spark 本身利用 Scala 编程语言编写
- 提供了多种语言的接口 (Python、R、Java等)
- Scala 版本功能最全
- Python 和 R 在数据科学中更常见
- 后面的课程中将主要使用 Python 接口, 即 PySpark



安装 Spark/PySpark



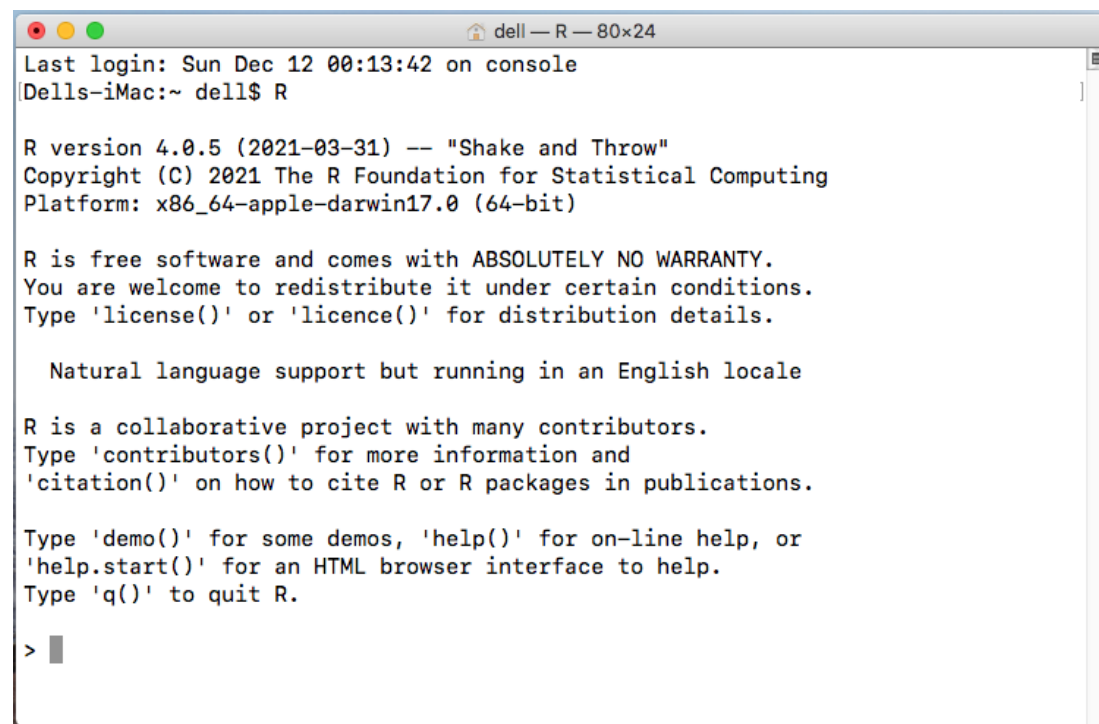
补充知识：环境变量

环境变量

- 环境变量指的是操作系统中用来指定系统运行环境的一些参数
- 例如临时文件夹的位置，系统文件夹的位置等
- 可以被应用程序读取
- 其中决定应用程序搜索位置的变量叫做PATH

PATH

- 假设系统里已经安装了 R
- 为什么在 Mac 的终端里键入 R 就会打开 R 程序
- 而在 Windows 下就不行?



```
dell — R — 80x24
Last login: Sun Dec 12 00:13:42 on console
Dells-iMac:~ dell$ R

R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

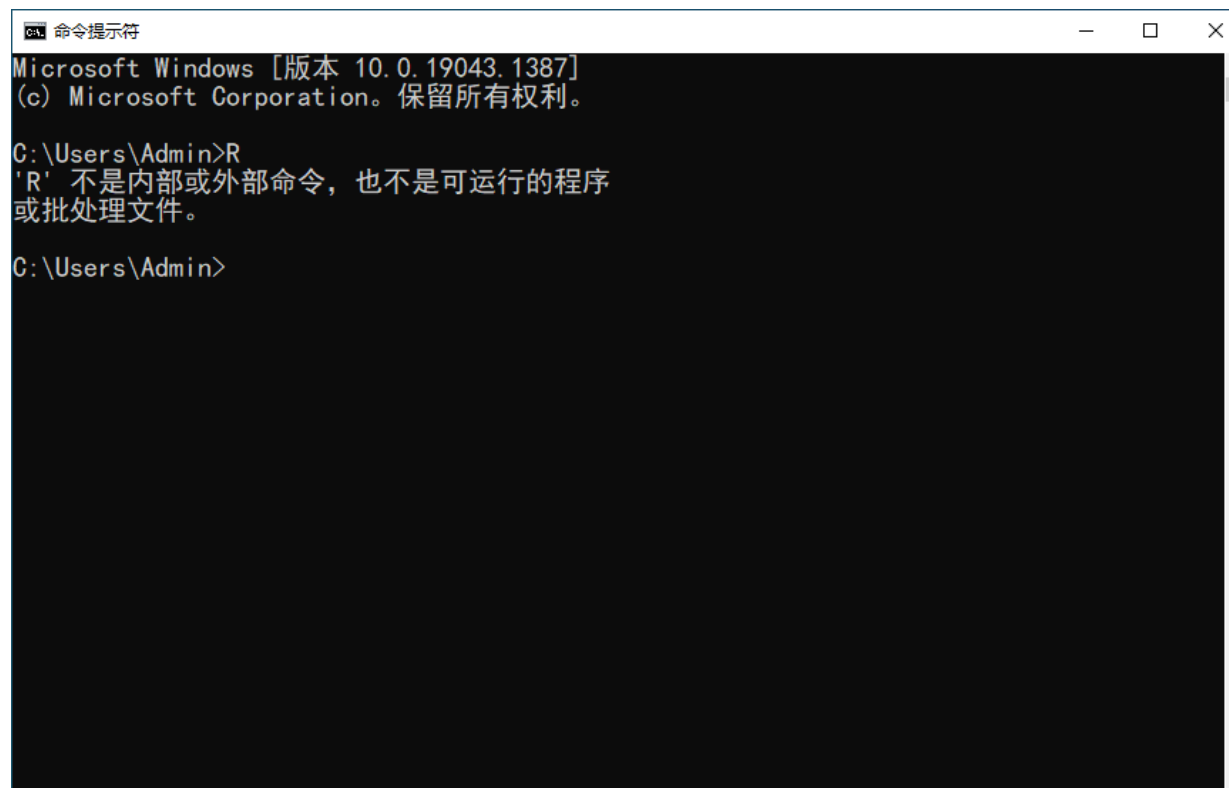
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █
```


PATH

- 假设系统里已经安装了 R
- 为什么在 Mac 的终端里键入 R 就会打开 R 程序
- 而在 Windows 下就不行？



```
命令提示符
Microsoft Windows [版本 10.0.19043.1387]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Admin>R
'R' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

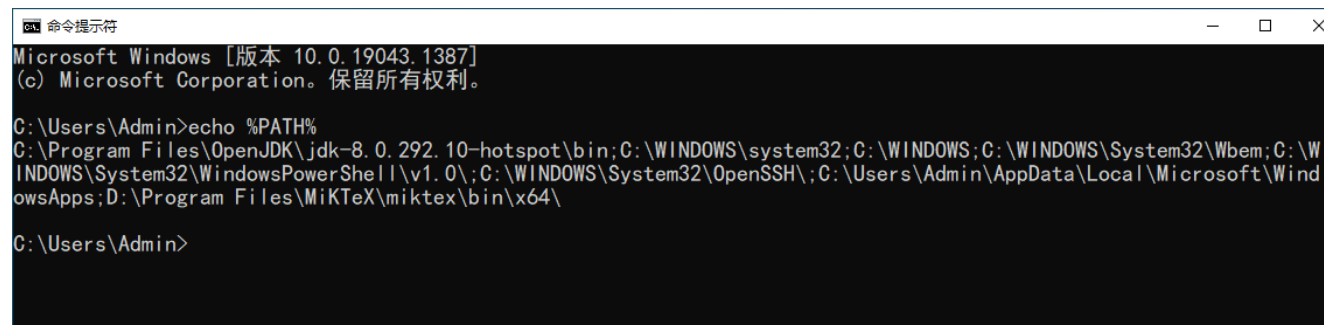
C:\Users\Admin>
```

PATH

- 在终端中输入应用程序名称（如 pyspark）时
- 终端会检查它是否是完整路径
- 如果不是，就会在一些预定义好的位置中搜索这个程序
- 如果找到匹配的程序，就会继续执行
- 否则将提示找不到程序
- 这些预定义好的位置就是 PATH 变量

PATH

- Windows 下可以在终端输入 `echo %PATH%` 查看 PATH 变量

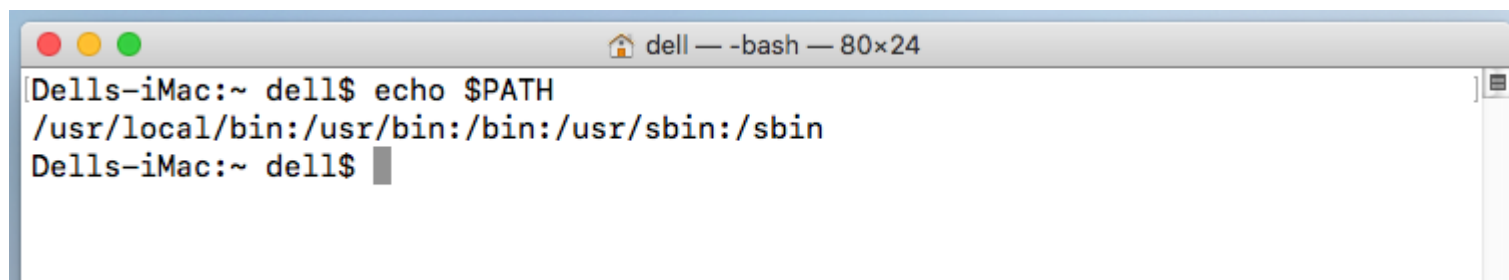


```
命令提示符
Microsoft Windows [版本 10.0.19043.1387]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Admin>echo %PATH%
C:\Program Files\OpenJDK\jdk-8.0.292.10-hotspot\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\W
INDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Users\Admin\AppData\Local\Microsoft\Wind
owsApps;D:\Program Files\MiKTeX\miktex\bin\x64\

C:\Users\Admin>
```

- Linux/Mac 下可以输入 `echo $PATH`

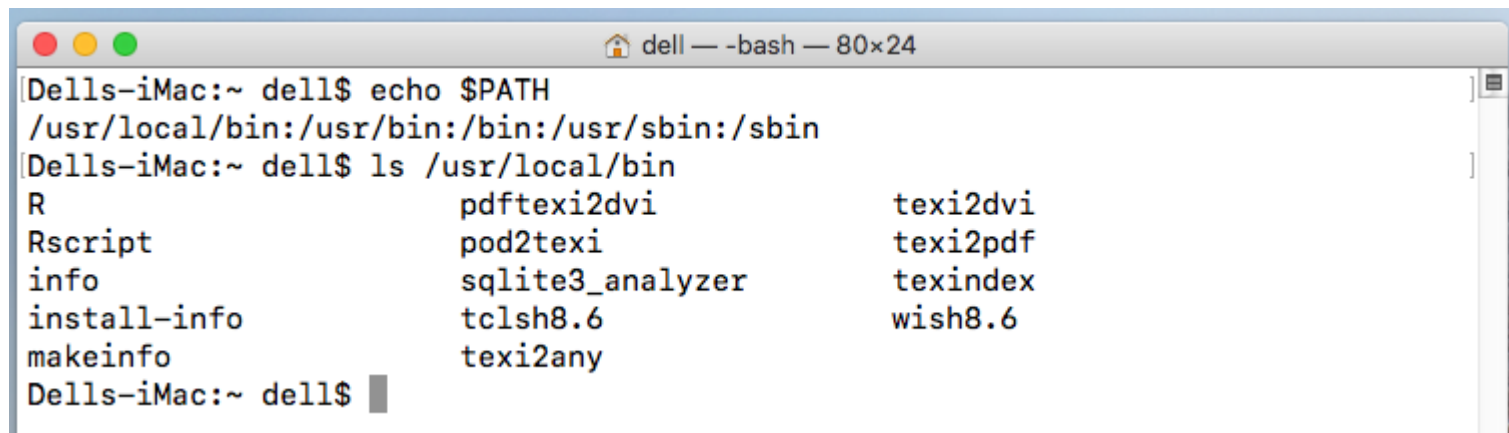


```
dell — -bash — 80x24

[Dells-iMac:~ dell$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
Dells-iMac:~ dell$
```

PATH

- Mac 会将程序安装至 PATH 下的某个目录
- 因此通常输入程序名就可以执行

A screenshot of a macOS terminal window titled 'dell — -bash — 80x24'. The terminal shows the output of two commands. The first command, 'echo \$PATH', displays the system's PATH as '/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin'. The second command, 'ls /usr/local/bin', lists the contents of that directory in three columns: R, Rscript, info, install-info, makeinfo, pdftexi2dvi, pod2texi, sqlite3_analyzer, tclsh8.6, texi2any, texi2dvi, texi2pdf, texindex, and wish8.6.

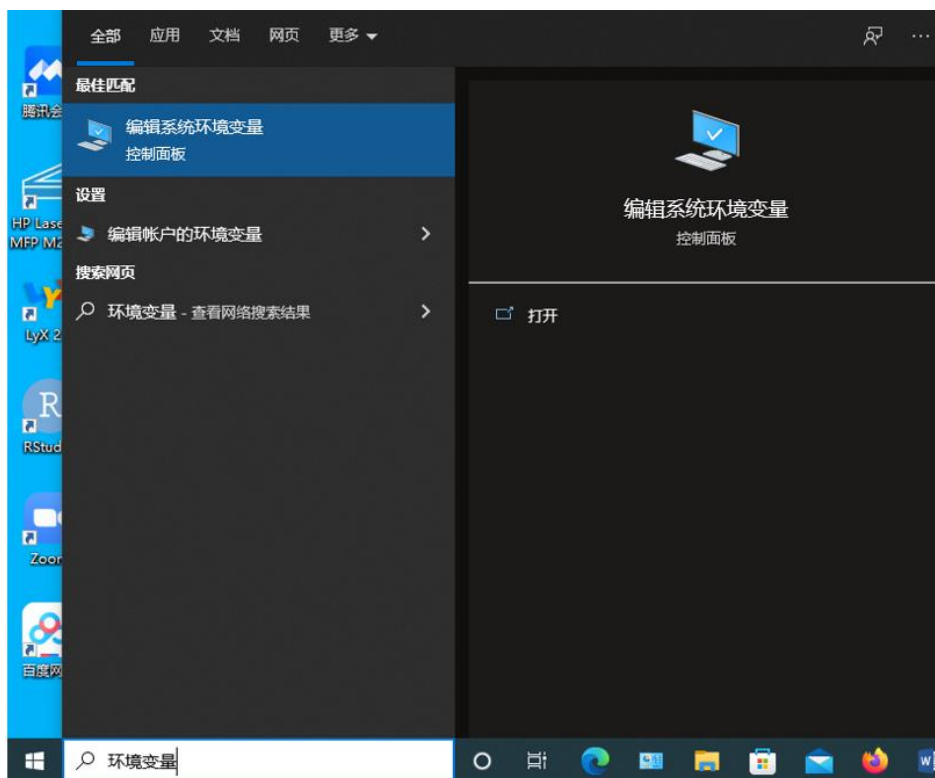
```
[Dells-iMac:~ dell$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
[Dells-iMac:~ dell$ ls /usr/local/bin
R                               pdftexi2dvi                texi2dvi
Rscript                        pod2texi                   texi2pdf
info                           sqlite3_analyzer           texindex
install-info                  tclsh8.6                  wish8.6
makeinfo                      texi2any
```

设置环境变量

- 安装 Spark ≈ 解压缩文件+设置环境变量
- HADOOP_HOME 指定 Hadoop 的位置
- SPARK_HOME 指定 Spark 的位置
- PATH 中加入 pyspark 等程序的位置

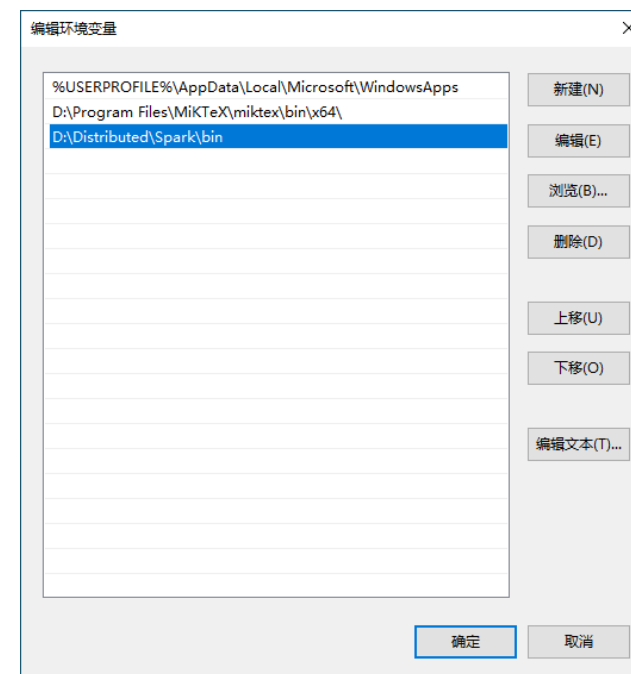
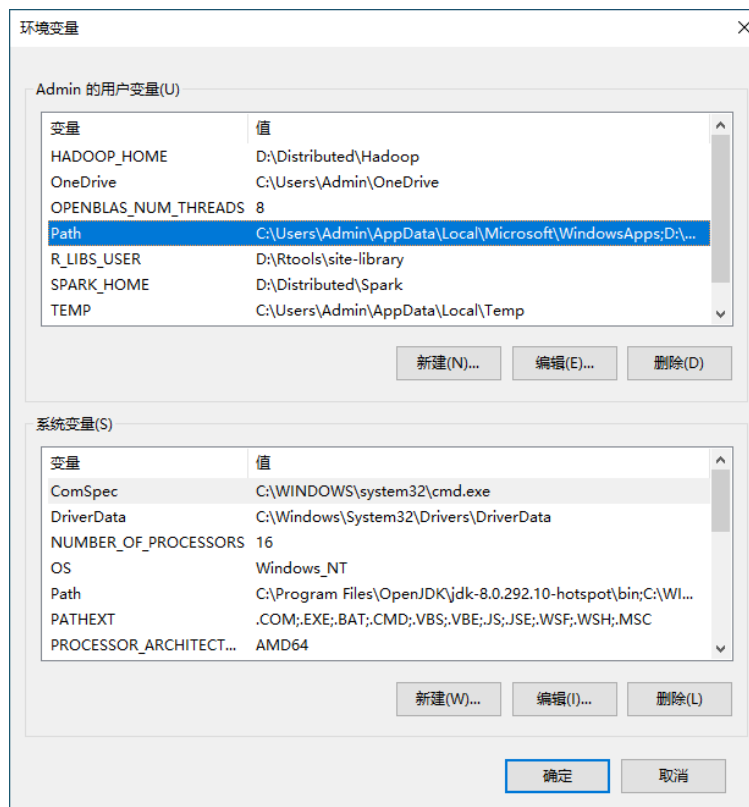
设置环境变量

- Windows 下在开始菜单搜索栏中输入“环境变量”，打开编辑器



设置环境变量

- 双击用户变量中的 Path 栏，加入 Spark 的 bin 文件夹
- 新建 HADOOP_HOME 和 SPARK_HOME 环境变量



设置环境变量

- Linux/Mac 下环境变量可以通过终端命令设置
- `export PATH=<new path>:$PATH`
- 后面的 “:\$PATH” 指的是添加到原有路径，而不是覆盖
- 假设要添加的路径是 `/Users/dell/spark/bin`



```
dell — -bash — 80x24
[Dells-iMac:~ dell$ export PATH=/Users/dell/spark/bin:$PATH
[Dells-iMac:~ dell$ echo $PATH
/Users/dell/spark/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
Dells-iMac:~ dell$
```


设置环境变量

- 但这样只对当前的终端有效，新打开的终端需要重新设置，不方便使用
- 在 Linux/Mac 下终端会读取用户目录下的一个隐藏文件 `.bashrc`，可以在其中加入 `export` 命令
- 保存文件后重新打开终端，检查 `echo $PATH` 是否包含了新的路径



The image shows two overlapping windows from a macOS desktop. The background window is a terminal titled 'dell - bash - 80x24'. It contains the following commands and output:

```
Dells-iMac:~ dell$ touch ~/.bashrc
Dells-iMac:~ dell$ open -e ~/.bashrc
Dells-iMac:~ dell$
```

The foreground window is a text editor titled '.bashrc'. It contains the following line of code:

```
export PATH=/Users/dell/spark/bin:$PATH
```

安装 Spark

- 具体安装流程参见 [PySpark安装.docx](#)

PySpark 示例： 分析联合国决议文件

数据源

- 1990年至2014年间联合国的公开文件
- 包含六种官方语言版本
- 六种语言间的相互翻译，按句子对齐
- <https://conferences.unite.un.org/UNCORPUS/>

示例

- 参见 [lec2-text-file.ipynb](#)



Spark 运行模式

运行模式

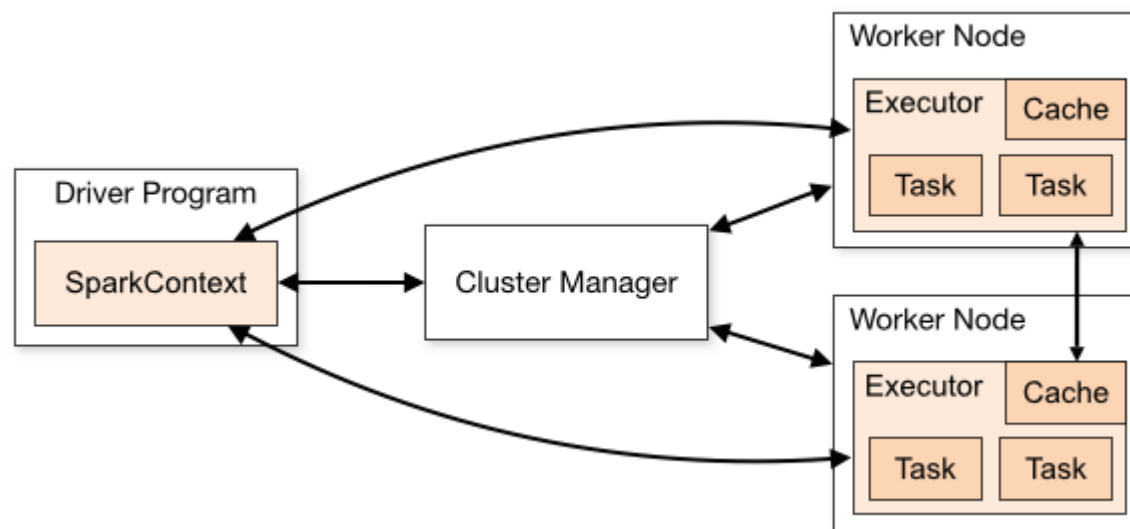
- Spark/PySpark 既可以运行在单机上，也可以部署在计算机集群上
- 通常单机模式用于展示、开发和调试
- 可以指定本地机器并行处理数

单机模式

- `spark = SparkSession.builder.\nmaster("local[*]").\nappName("Reading Text").\ngetOrCreate()`
- `local[*]` 指使用所有 CPU 核心
- 也可以显式指定数目, 如 `local[8]`

集群模式

- 集群模式下 Spark 通过一个集群管理器来连接不同的机器
- 支持的集群管理器包括 Apache Mesos, Hadoop YARN 等
- Spark 也自带了一个集群管理器



集群模式

- Spark 内置的集群管理器使用很方便
- 建立主机: `spark-class`
`org.apache.spark.deploy.master.Master`
- 建立工作机: `spark-class`
`org.apache.spark.deploy.worker.Worker` `spark://<ip>:<port>`

集群模式

- 连接 Spark 内置集群管理器
- ```
spark = SparkSession.builder.\n master("spark://<ip>:<port>").\n appName("Reading Text").\n getOrCreate()
```

# 运行模式

- Spark 的一大优势
- 大部分代码可以一次编写
- 在不同的环境下运行