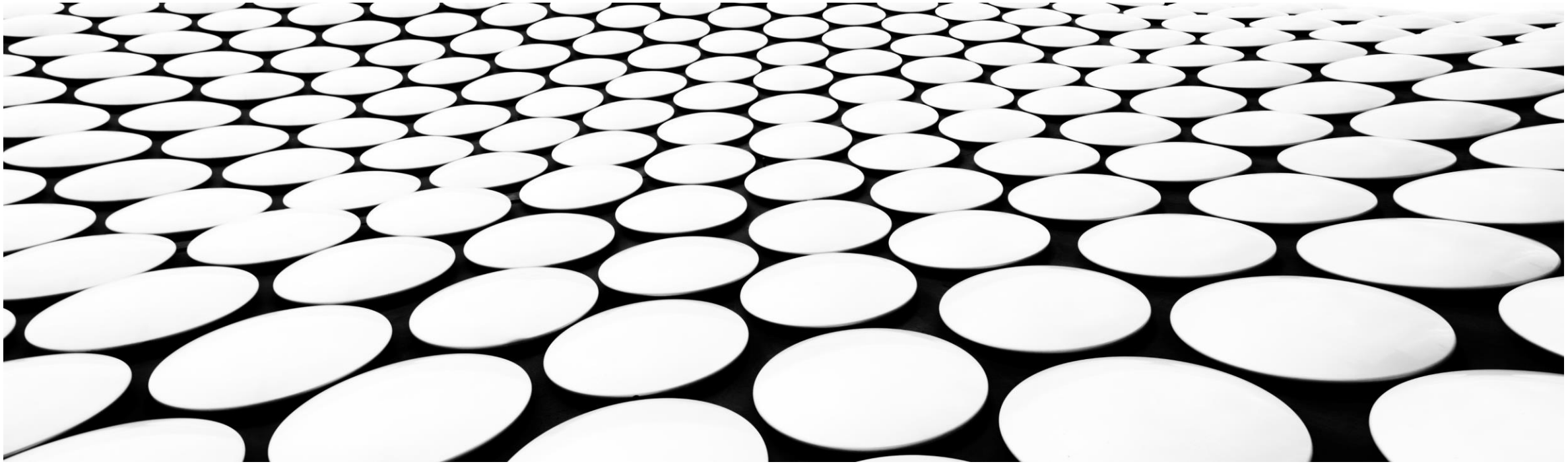
分布式计算

邱怡轩



今天的主题

- 谁在教这门课
- 谁应该上这门课
- 这门课讲什么
- 分布式计算是什么
- 用什么工具软件

教学团队

- 教师：
- 邱怡轩，副教授
- 邮箱 qiuyixuan@sufe.edu.cn

- 助教：
- 刘杨漾，2018级硕博
- 邮箱 liuyy.sufedc@outlook.com

关于我

- 博士毕业于美国普渡大学统计系
- 普渡大学不是佛学院

关于我

- 博士后工作于卡内基梅隆大学
- My heart is in the work

— Andrew Carnegie

关于我

- 普渡大学钟楼

关于我

- 普渡大学统计系

关于我

- 普渡大学统计系

关于我

- 卡内基梅隆大学校园

关于我

- 卡内基梅隆大学校园茅以升先生铜像

- 思考并挑选4+个问题回答（第一次作业）：

- 你的未来规划（近期或长期）是什么?
- 凭第一印象，你觉得分布式计算是做什么的?
- 你期待从这门课中学到什么?
- 你还有哪些喜欢的获取知识的渠道（例如B站，知乎，技术博客，公众号等）？如果有请向大家推荐一些，具体到账号或网址，不需要与本课程相关。
- 你对编程的喜欢/厌恶程度是多少？1为很厌恶，10为很喜欢。
- 你喜欢和不喜欢的上课方式是什么?

关于这门课

学习内容

- 分布式计算的基本概念和原理
- Apache Spark 计算框架的使用
- 统计与机器学习模型的分布式算法
- 动手实践、编程

必备技能

- 较熟练地使用 Python 编程
- 基本的线性代数和矩阵知识
- 了解常见的统计模型
- 赶紧去复习极大似然估计，说三遍
- 好奇心

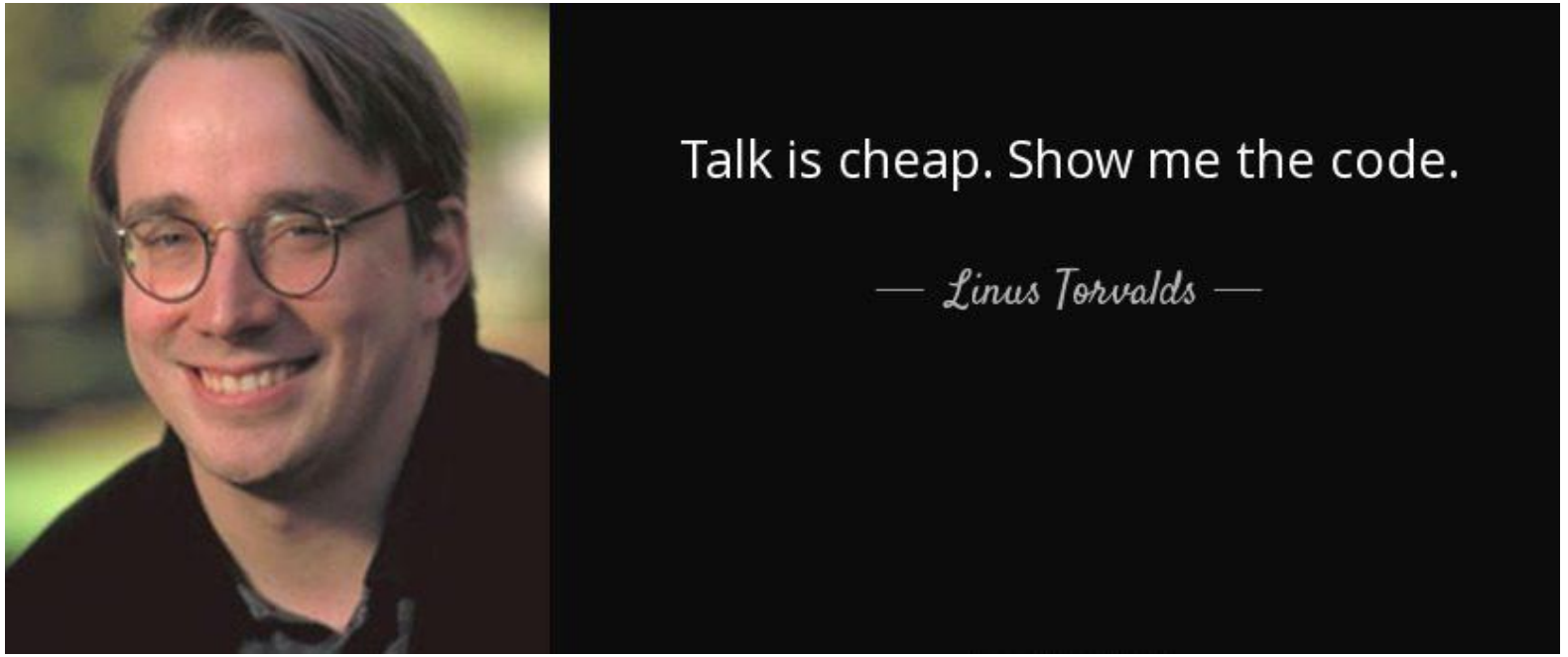
我的期望

- 上完这门课以后，我希望你们：

- 对“计算”这件事产生兴趣
- 有信心说“我也是玩过大数据的人了”
- 收获一门实用的技能

为什么一定要强调动手

- 君子动口不动手
- 编程动手不动口



为什么一定要强调动手

- 编程实现是验证有没有学懂的最佳方法
- 很多技能是数据科学的标配
- 平时的作业可能是将来的面试题

课堂形式

- 课堂教学
- 提问、讨论
- 上机、编程

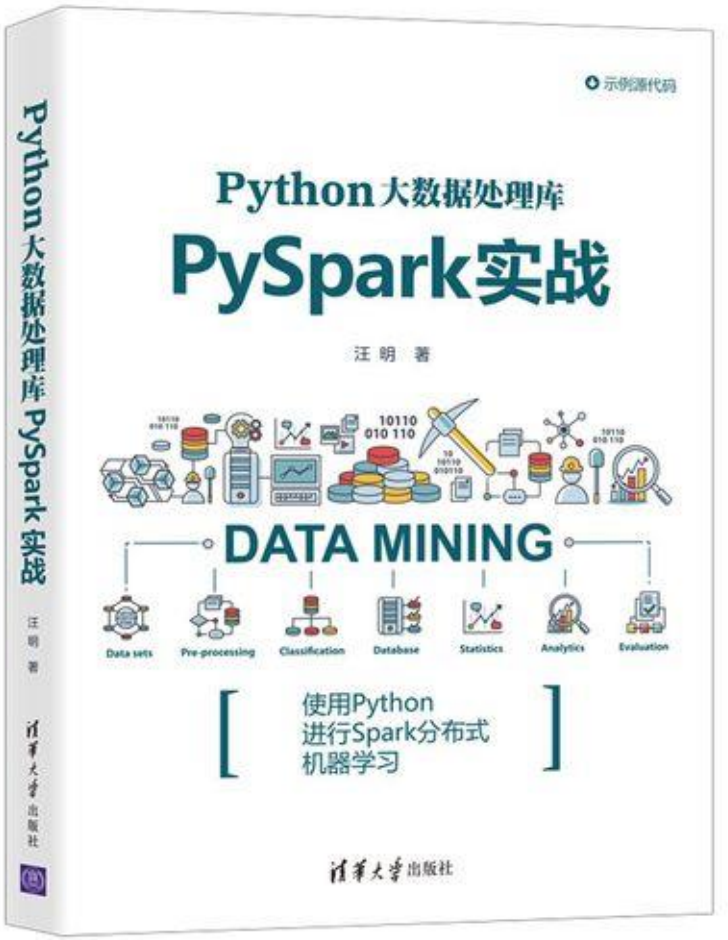
- 鼓励互动

课程资料

- Canvas
- https://canvas.sufe.edu.cn

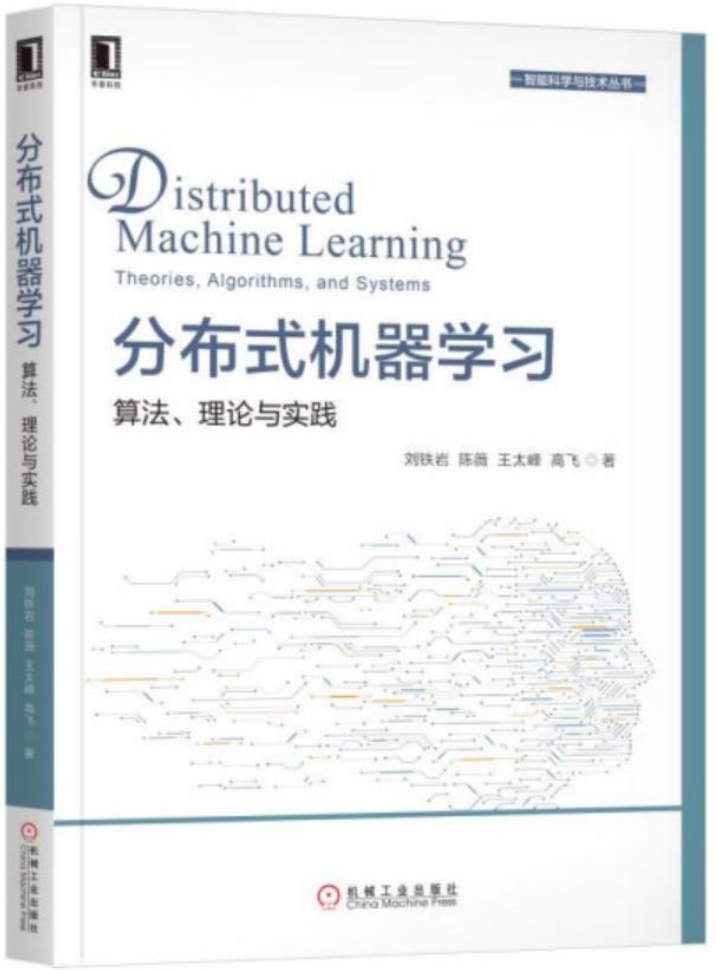
参考书目

- 《PySpark实战》



参考书目

- 《分布式机器学习》



考核方式

- 考勤——10%
- 作业——30%
- 期末——60%

- 期末考试暂定为上机答题和编程
- 请随时关注信息更新

纪律

- 千万不要抄袭
- 千万不要抄袭
- 千万不要抄袭

- 鼓励交流想法，主体需自己完成
- 参考或引用了别处代码时需注明（包括人类和非人类）
- 适用于作业与期末考试

关于分布式计算

三个问题

- 什么是分布式计算（What）
- 为什么需要分布式计算（Why）
- 怎么进行分布式计算（How）

三个问题

- 我们这门课全程都将围绕这三个问题展开
- 下面将用两个实验启发大家的思考

实验1

- 你真的充分利用了你的计算资源吗?

实验1

- 打开 R 和任务管理器，运行下面的程序

```
set.seed(123)
x = matrix(rnorm(2000^2), 2000)
system.time(for(i in 1:5) x %*% x)
```

- 观察任务管理器中的 CPU 使用率

实验1

- 找到 R 的安装路径
- 如 C:\Program Files\R\R-4.2.2\bin\x64
- 用 Rblas_s.dll 替换 Rblas.dll

- `set.seed(123)`
- `x = matrix(rnorm(2000^2), 2000)`
- `system.time(for(i in 1:20) x %*% x)`

实验1

- 用 `Rblas_p.dll` 替换 `Rblas.dll`
- 再次运行代码，观察 CPU 使用率

实验1

- 如果矩阵是50000x50000呢?

实验2

- 怎样把机房的电脑组合成一台超级计算机?

实验2

- Spark主机
- `spark-class org.apache.spark.deploy.master.Master`

实验2

- Spark工作机
- spark-class org.apache.spark.deploy.worker.Worker spark://<ip>:<port>

实验2

- Spark主机
- `lec1-intro-pyspark.py`

思考

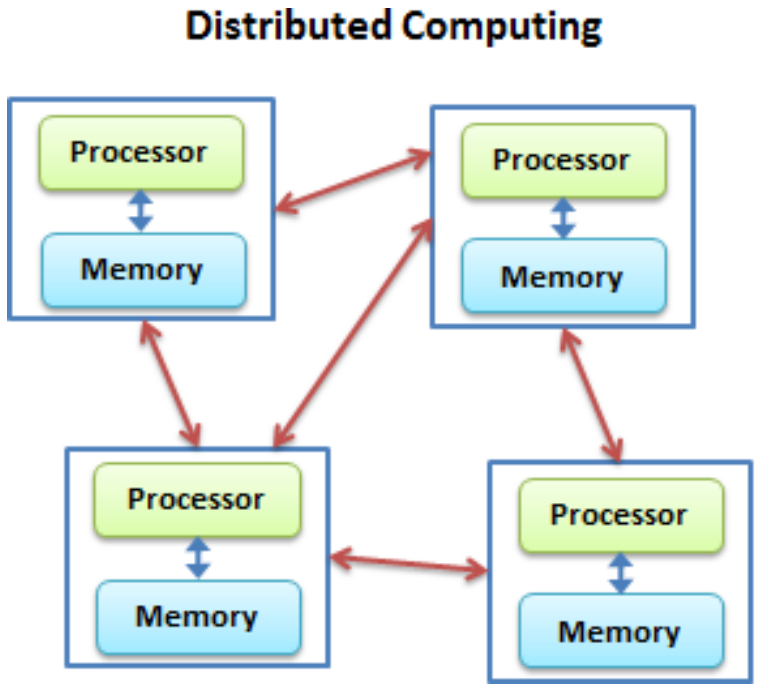
- 刚才的例子透露出了哪些信息?

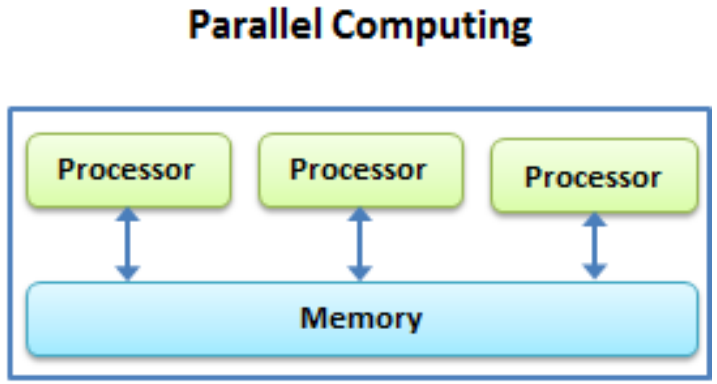
思考

- 这门课的终极目的是最大限度地利用现有的计算资源完成数据分析的任务

思考

- 并行计算 vs 分布式计算





思考

- 并行计算 vs 分布式计算

- CPU 指令并行
- 多核/多 CPU 并行
- 多机并行

思考

- 系统+算法+实现

思考

- 哪些模型和方法容易并行化?
- 如何为常用的模型设计分布式算法?

- 线性回归
- Logistic 回归
- 决策树、随机森林
- 神经网络、深度学习
- ……

思考

- 分布式计算一定比单机快吗?
- 影响计算速度的因素有哪些?

我们将在后续课程逐渐展开

课程工具软件

作业流程

- 本节课将使用Git来提交作业
- 参见《Git快速入门》
- 附加功能：利用Git制作简历（https://gitee.com/cool-resume）