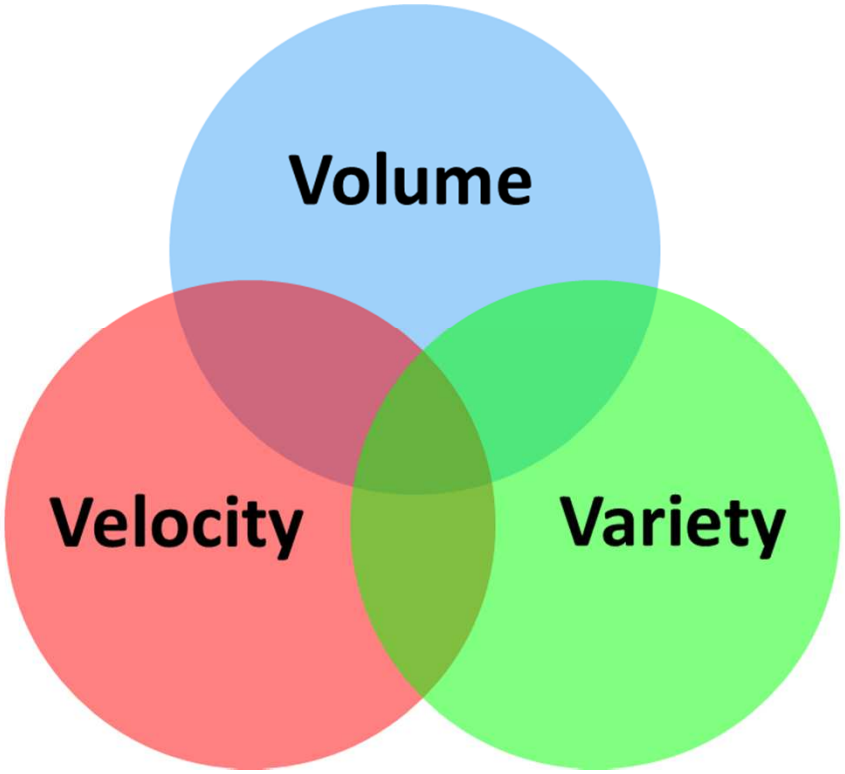
大数据系统与大规模数据分析

大作业



陈世敏
（中科院计算所）
孙翼
（国科大计算机学院）

课程相关

- 成绩分配
  - 闭卷考试：50%
  - 作业1+作业2+作业3：30%
  - 大作业：20%
  - 课堂表现：+5%

大作业安排

- 成绩： 占总成绩20%
- 时间
  - 发布：2017/4/12 (Wed)
  - Mid-term Report：**2017/5/14(Sun)，北京时间 11:59pm**
    - 报告：1页，A4，pdf
    - 内容：成员，选题，分工，初步设计
  - Final Report：**2017/6/6 (Tue)，北京时间 11:59pm**
    - 报告：至少6页，A4，pdf
    - 程序（包括README安装和运行指令）
  - Final Presentation: **2017/6/7 (Wed)**
    - 每组10分钟
    - 目的、文献/现有系统分析、设计、实现、性能/演示等
    - 对成员分工执行情况进行自我总结，特别突出的加分

上机安排(1)

- 地点
  - 计算机学院，4层
  - 网络安全教学实验室（447室）：50台
  - 云计算教学实验室（432室）：20台
- 机器：联想PC机M6400t，Windows 7/32bit
  - 环境：每台机器安装了一个虚拟机，运行Ubuntu Linux 14.04.2, Hadoop 2.6.0, HBase 0.98等
- **云计算实验室有10台刀片服务器**
  - **和助教联系，分配账号，可以用于实验**
- 注：可以在自己的计算机上完成作业

上机安排(2)

- 时间
  - 周五上午，8:30-11:50am
  - 周五下午，1:00-4:20pm
- 上机期间助教的职责
  - **管理上机秩序：**上机前找助教签到，分配机器；使用完毕，找助教签出；助教负责监督机房秩序(不得喧哗、打闹等)。
  - **解答机器使用的问题：**包括如何开机、如何登录、如何使用编辑器、如何编译和运行程序
  - **不包括：其它关于作业内容的问题**

分组和选题

- 自愿组合，每组不多于5人
  - 共239人，大致48组
- 每个组
  - 起一个组名（中文或英文），不要太长
  - 选一名组长，组长负责召集组员完成作业
  - 确定成员分工
- 大致确定了作业的选题后，联系助教
  - 希望选不同的题目
  - 同一个题目，至多4个组可以选
  - 助教协调，会告知题目是否已经被选了
- 报名：杨若雪 yangruoxue@ict.ac.cn

作业内容

- 目的
  - 在课程学习的基础上，通过一定的开放性探索和编程实践，加深理解，锻炼实践和学习能力

- 选题
  - 我们会提供一些题目
  - 鼓励自由选题：有一定的新意
  - *每个题目最多4组可以同时选*

作业成绩20分

- 5分：新颖性
  - 如果是特别具有创新性，可以额外加分
- 10分：探索深度和实现效果
  - 根据工作量，工作深度，实现程度和效率/演示质量等
- 5分：表达
  - Report的文字是否通顺、易懂
  - PPT讲述是否清楚
  - 注意：需要把新颖性和工作深度有效地表达出来
- +2分：贡献
  - 根据成员的具体贡献，对于特别突出
    - 如果1人特别突出，加2分
    - 如果2人特别突出，每人加1分

Memcached/Redis + HBase/Cassandra

- 目标
  - Hbase/Cassandra等KV-store是面向硬盘的
  - Memcached/Redis等是面向内存的

- 探索
  - 是否可以两者相结合
    - 自动把热点key放入Memcached/Redis
    - 放置策略需要考虑key的访问频率和读写情况
    - 容错的支持
  - 通过实验，观察优点和缺陷

Transactional Key-Value Store

- 目标
  - KV store只支持在单个key上的一致性
  - 以一个KV store为基础，实现一个事务处理系统
- 功能
  - 客户端提交的一个批量操作，认为是一个transaction
  - 需要保证 分布式ACID

KV-Store vs. MongoDB

- Key-Value数据可以很容易的表达为JSON
  - {key: "…", value: "…"}
  - 所以可以很容易地存储在MongoDB中
- 反之，JSON也可以分解为KV数据存储在KV-Store中
- 设计一组Benchmark比较的性能
  - KV-Store vs. Key-Value in MongoDB
  - MongoDB vs. JSON in KV-store
- 考虑数据小于和数据大于内存的情况

Graph Store on KV-Store

- Neo4j实现对于图的访问十分低效
- 希望用KV-Store实现基本的图存储
  - 把图的顶点，边，属性的信息存储在KV-Store中
  - 自行设计存储方案
- 进行性能比较
  - 随机顶点读操作（读所有属性）
  - 随机边的读操作（读所有属性）
  - 找邻居
  - 找邻居的邻居
  - 高级目标：子图匹配

Graph Store on MongoDB

- Neo4j实现对于图的访问十分低效
- 希望用MongoDB实现基本的图存储
  - 把图的顶点，边，属性的信息存储在MongoDB中
  - 自行设计存储方案
- 进行性能比较
  - 随机顶点读操作（读所有属性）
  - 随机边的读操作（读所有属性）
  - 找邻居
  - 找邻居的邻居
  - 高级目标：子图匹配

GraphLite基本图算法的实现与测试

“LDBC Graphalytics: A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Platforms”, VLDB 2016.
http://www.vldb.org/pvldb/vol9/p1317-iosup.pdf

- Benchmark提供了真实图数据和图产生器
- 规定了6中算法
  - Bread-first search
  - PageRank
  - Weakly connected components
  - Community detection using label propagation
  - Local clustering coefficient
  - Single-source shortest path
- 要求：实现上述算法，并进行性能测试
  - 可以与Spark GraphX进行比较

GraphLite上实现新的图算法

- Stochastic gradient descent
- Belief propagation
- Gibbs sampling
- MRF parameter learning
- CoEM
- Compressed Sensing
- Lasso
- Conductance
- Lanczos algorithm
- Singular value decomposition
- Support vector machine
- Node2vec
- 思考：如何实现？是否可以优化？是否可以通过改进GraphLite优化？

Online Aggregation on Spark

- Spark 可以支持Sample功能
  - Sample在处理大量数据时非常重要
- Online Aggregation
  - Sample数据量逐渐增加，从少到多，直至全部数据
  - 估计Aggregation的结果，越来越精确
- 基于现有的Spark Sample功能，实现Online Aggregation

数据流系统：文献综述

- 写一篇综述，调研现有的数据流系统
  - Apache Storm
  - Yahoo S4
  - Spark streaming
  - Apach Flink
  - 等
- 特征比较
  - 系统结构、编程模型、运行方式、容错处理等
- 性能比较
  - 设计一个benchmark，比较不同系统的性能

基于Storm的SQL处理

- 要求在Storm的基础上实现简单的SQL操作
  - Selection
  - Projection
  - Join (time window based)
  - Group-by
  - Aggregation
- 简化的维度
  - Selection条件的个数
  - 数据类型：例如只支持整数类型

数据分析部分：文献阅读

- 学习机器学习/深度学习相关知识（根据自己兴趣选择）,比如说CNN，LSTM，GAN等。
- 实现算法实例

数据分析：个性化推荐模块开发

- 在实际的学习过程中,我们希望基于学生的答题情况了解学生对于各个知识点的掌握情况,从而更有针对性地进行题目推荐
- 本次作业，同学们自己收集题库（比如四六级的英语考题/数学考题/编程语言测试题等），利用推荐算法进行个性化的试题推荐的简易系统

数据分析：记忆增强软件原型开发

- 记忆作为知识学习中极其关键的一个环节,不仅与知识概念本身的组织有关,更与记忆者本人
- 的学习方法密切关联,我们希望通过合理的调度算法确保学生可以在合适的时间重复特定的知识概念,从而实现更加高效的记忆效率,这是我们记忆增强软件的初衷。
- 基于对记忆过程的研究,可以实现针对不同应用场景下的 Scheduling Algorithms 测试，本次作业以背单词为例，实现论文中的记忆调度算法或者构建自己的记忆调度算法，并能够把这些方法以网页/APP等形式进行交互展现。

数据分析部分

- 对大作业感兴趣的同学可以联系王浩博助教或孙老师索要前期资料

相关会议

- SIGMOD/VLDB/ICDE
- SOSP/OSDI
- NSDI
- SOCC
- KDD/EDM/LAK/AIED/SDM
- IJCAI/IAAA/NIPS