AIM3 – Scalable Data Mining and Data Analysis

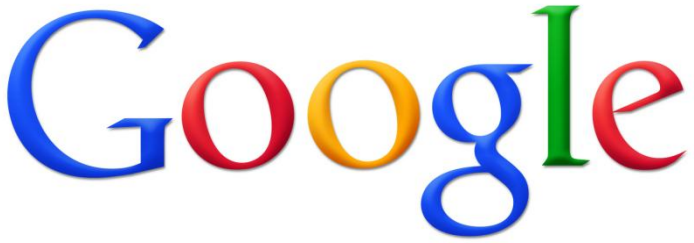
01 – Motivation

Sebastian Schelter, Christoph Boden, Volker Markl

Fachgebiet Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

http://www.dima.tu-berlin.de/

A web-scale search engine



- maintains a copy of the World Wide Web
  - estimated to have 7.3 billion pages, March 2012
- Challenges
  - search this copy in sub-seconds
  - identify duplicate content
  - compute the ‚importance' of pages (PageRank)
  - display content-related ads

The largest social network

- maintains the world's largest social network
  - 721 million active users (May 2011), more than 1/10 of the world's population!
  - 68.7 billion friendship links
  - 2.7 billion likes per day

- Challenges
  - provide realtime updates of friends activities
  - suggest new friends (link prediction)
  - display content-related ads
  - compute statistics about the social graph

The world's fastest news medium

- realtime communication via short messages
  - 2009: 2 million tweets per day
  - 2010: 65 million tweets per day
  - 2011: 200 million tweets per day

- Challenges
  - allow search in (near) realtime
  - recommend interesting people (link prediction)
  - find topics in the messages

What happens at such a scale?

What happens at such a scale?

A technological challenge

- Do existing approaches suffice to solve this challenge?
  - Can we put the data in a relational database?
  - Can we find an appropriate schema for the data?
  - Can we process the data on a single machine?
  - Can we process the data in Matlab?

Probably not.

- Solution: run computations in parallel on dozens, hundreds, thousands of machines, **but**:
  - for economic reasons, we wish to use commodity hardware, such machines will fail and break regularly
  - software that is designed to run on a single machine cannot 'magically' run on a cluster
  - scheduling tasks, handling concurrency and failure as well as transferring intermediate results in a distributed system are extremely difficult engineering tasks

A new set of tools

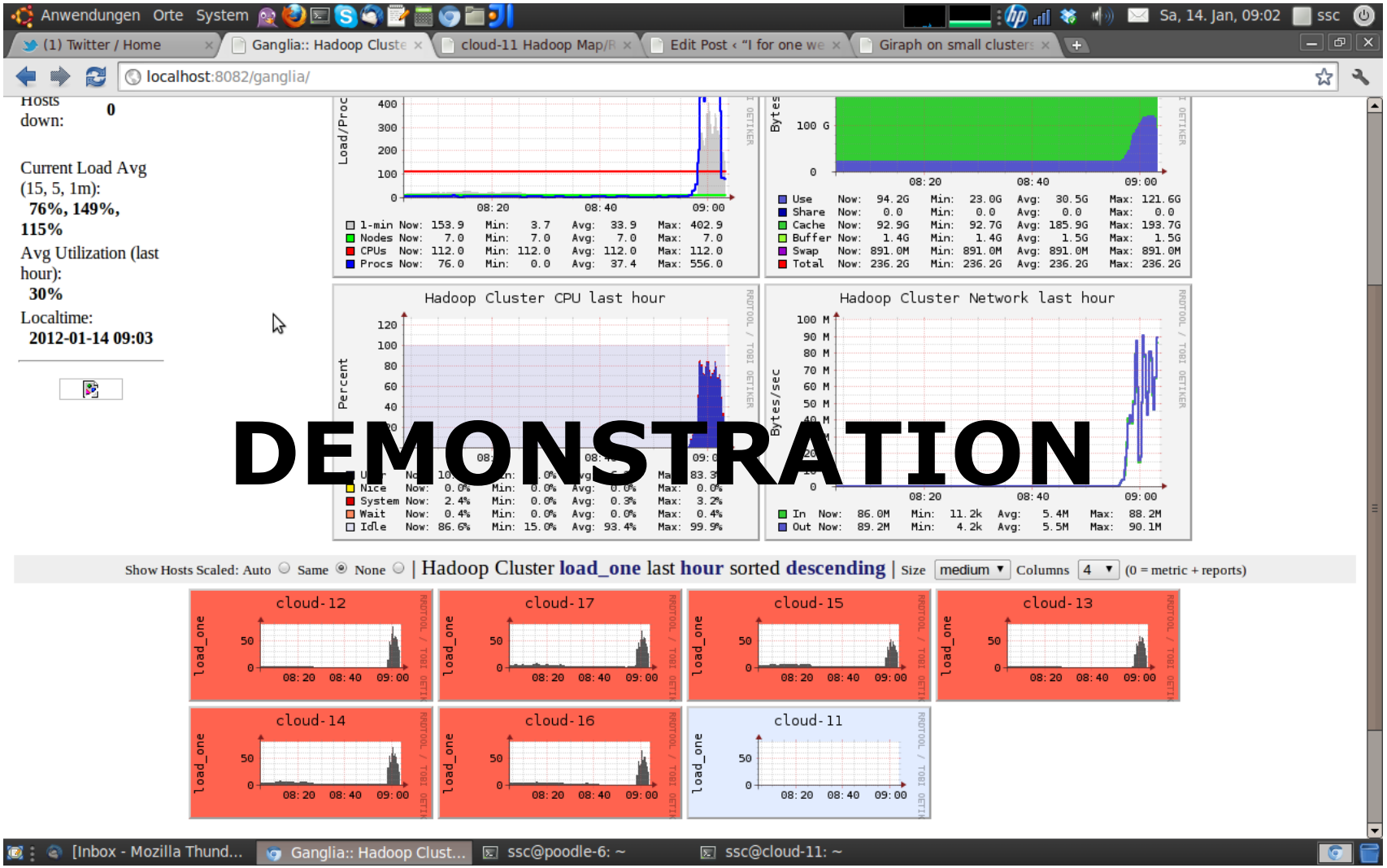
- Distributed filesystems
  - store petabytes of data in the cluster
  - transparently handle reads, writes and replication

- Parallel processing platforms
  - offer a parallel programming model to allow developers to write distributed applications
  - move computation to data, not data to computation
  - relieve the developer from handling concurrency, network communication and machine failures

Our algorithms have to change

- Each machine will only see a small portion of the data
  - we cannot use random access anymore, we must always work on partitioned data
  - joining data become very costly as lots of machines will be involved
- Communication via network and disk becomes the bottleneck
  - our algorithms must try to locally aggregate as much as possible
  - minimizing network traffic becomes the key to scaling out algorithms
- Concurrency and recovery must be hidden from the developer
  - algorithms must fit into a simple, parallelizable programming model
  - the system (not the developer) handles concurrency and recovery

Demo


Anwendungen Orte System
Sa, 14. Jan, 09:02 ssc
(1) Twitter / Home
Ganglia:: Hadoop Cluste
cloud-11 Hadoop Map/R
Edit Post ‹ "I for one we
Giraph on small clusters
localhost:8082/ganglia/
Hosts down: 0
Current Load Avg (15, 5, 1m): 76%, 149%, 115%
Avg Utilization (last hour): 30%
Localtime: 2012-01-14 09:03
Load/Proc
Bytes
Hadoop Cluster CPU last hour
Percent
Hadoop Cluster Network last hour
Bytes/sec
DEMONSTRATION
Show Hosts Scaled: Auto Same None | Hadoop Cluster load_one last hour sorted descending | Size medium Columns 4 (0 = metric + reports)
cloud-12
cloud-17
cloud-15
cloud-13
cloud-14
cloud-16
cloud-11
load_one
[Inbox - Mozilla Thund...
Ganglia:: Hadoop Clust...
ssc@poodle-6: ~
ssc@cloud-11: ~

Agenda

- Topics of the course
  - Motivation, Overview
  - MapReduce & Distributed filesystems
  - MapReduce: Joins, Patterns & Extensions
  - Stratosphere
  - Clustering
  - Dimensionality Reduction
  - Data Stream Mining
  - Graph Processing & Social Network Analysis
  - Graph Processing: Google Pregel
  - Collaborative Filtering: Neighborhood Methods
  - Collaborative Filtering: Latent Factor Models
  - Classification
  - Textmining
  - Specialized Machine Learning approaches

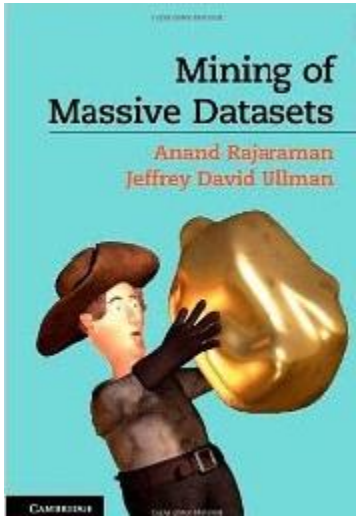
Tasks in the course

- 3 two week homework assignments
  - available as Java project on github
  - implement your solution and send us a patch
  - present your solution in the course
- six week project (in groups of 2-3 students)
  - implement a data mining algorithm on a parallel processing platform
  - demonstrate your solution on a real world dataset
  - 3 ten minute presentations: problem and planned solution, prototypical implementation, final presentation with results on real world data
- oral exam

Literature

- Mining of Massive Datasets (Rajaraman, Ullman)

  free PDF version available at:
  http://infolab.stanford.edu/~ullman/mmds.html



- Hadoop: The definitive guide (White)