CS 744 -- Big Data Systems

Welcome!

The “lite” version

CS 744 - Big Data Systems

http://www.cs.wisc.edu/~akella/CS744/S19

Check course web page for: office hours, readings, schedule, homeworks, announcements,...

Aditya Akella
Professor
Email: akella@cs.wisc.edu

Interests:
Networking, Big Data Systems
Formal methods applied to systems,
Data centers

Derek Hancock
(Full time TA)

Arjun Singhvi
(Unofficial help)

Email: cs744spring2019@gmail.com

My background

Networking: Data centers; SDN; network functions; Network verification, synthesis, and repair.

Systems: Serverless compute, isolation, interface between OS and network hardware.

Big data systems: scheduling (Tetris, Graphene, Carbyne), query optimization (Clarinet), serverless systems (F2), time series analysis, ML systems (MLFabric), analytics for spot markets/dynamic settings (QOOP).

Big Data Systems - 10000ft view

Large cluster || Lots of data || Analyze data || Multiple analyses in parallel

The big data stack:

Many applications, detailed logic

Computation, dependencies

Machines, failures/contention

Data, rich, multi-dimensional

10000ft view, again

Large cluster || Lots of data || Analyze data || Multiple analyses in parallel

- How to express computation? Programming model
- How to represent data?
- How to run computation at scale? Data parallelism, Execution framework
- How to deal with run time issues, e.g., slowdowns, failures?
- Key properties: Correctness? Consistency?

This class

We will cover:

- Cluster architecture
- Stacks that support many apps: Hadoop, Spark, Tez
- Schedulers
- Apps:
  a. Batch and stream analytics
  b. Graph processing
  c. Machine learning
- Trends:
  a. New hardware
  b. Serverless platforms
  c. Approximation

Theory, practice and research

10000ft view, again

Large cluster || Lots of data || Analyze data || Multiple analyses in parallel

- How to express computation? Programming model
- How to represent data?
- How to run computation at scale? Data parallelism, Execution framework
- How to deal with run time issues, e.g., slowdowns, failures?
- Key properties: Correctness? Consistency?

Questions common, but diff. types of data may benefit from custom answers.

- When are custom approaches needed? Why?
- Can a generic system provide the basis to answer above questions for all applications/types of data? At what trade-off?

Breezy Overview -- I

- Infrastructure
  - Networking, storage, compute; their common attributes, implications for design
  - Cluster management: scheduling, multi-tenancy, placement
- Frameworks for big data processing
  - General frameworks: programming model
    - MR, Tez, Spark
  - Execution
    - Intra-job scheduling, stragglers, fault tolerance

Breezy Overview -- II

- Applications -- how they are designed
  - Built atop general frameworks
    - Hive, SparkSQL, Clarinet, Spark Streaming, GraphX,
  - Custom applications with custom programming model
    - Storm, StreamScope, Pregel, TensorFlow, GraphLab
- Applications -- what problem they solve
  - Batch analytics -- SQL-like queries atop on-disk data; query completion time, makespan
  - Stream analytics -- queries on continuously arriving data; per-record latency, consistency
  - Graph processing -- processing graph data; iterative; vertex centric; how to partition, combine with other data; expressiveness
  - Machine learning -- building large models on large data sets; iterative; approximate; data and/or model parallelism; asynchronous/synchronous; programming model

My group

- Adding multi-tenancy support (e.g., to ML, Kafka)
- Geo-distributed systems for {batch, stream, graph, ML, …}
- OS/network support: NIC scheduling; NIC-OS interface; RDMA
- Network-assisted acceleration: using smart NICs or programmable switches
- Using big data systems ideas in other domains (e.g., rethinking network functions, network routing)
- Distributed HD video analytics and video query processing
- Verification and privacy in ML
- CloudLab++

Logistics and details

Lectures

Typically, one **assigned reading** per lecture

Will be posted on Piazza ~2 days prior

Every student must submit a summary on Piazza.

Due 10am on class day.

Use piazza to discuss/pose outstanding questions.

Each lecture will cover:

Background (where applicable)
Assigned reading in depth
Potential for future directions (time permitting)

Grading and Timeline

Two assignments 30% total

1. First two weeks in Feb
2. Mid April

Project (30%)

- List out early Feb
- Proposal (one page write up): ~Feb 15
- Regular meetings/checkpoints

2 Midterms

- Midterm 1 - Around Feb 27 (15%)
- Midterm 2 - Last class day (15%)
- Will each cover subset of papers

10% grades for class & Piazza participation

Assignments

Assignment 0 - getting used to CloudLab

Assignment 1

- HDFS, Spark
- Simple Spark applications

Assignment 2

- Machine Learning

Welcome to do additional work from prior years if interested

Academic Misconduct

We take any evidence of plagiarism seriously

Includes everything: paper summaries, assignments, and exams

We use a variety of tools to detect plagiarism

Once detected and confirmed, we **will** report you (both plagiarists and enablers) to the Dean of Students → Misconduct will go on your transcript

Getting Started

1. Decide you will **definitely** be taking the class -- drop now if not 100% sure
2. Form **groups of 3** to work on assignments and projects. Note: mid-semester group changes are tough to accommodate, but we will help out where possible
3. Enter your group details on the group signup spreadsheet - will be shared today on Piazza
4. Start working on Assignment #0 (releasing in the next day or two - check Piazza)

Getting Answers

1. Ask your peers using Piazza
2. Email or setup an appointment with the TA(s)
3. Last resort - Prof :)

What's next...

We won't meet until Friday 2/1:

1. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, L.A. Barroso, U. Holzle, Synthesis Lectures on Computer Architecture, 2009. Chapter 1 and 2.
2. *VL2: A Scalable and Flexible Data Center Network*, Greenberg et al., SIGCOMM 2009.
3. *The Hadoop Distributed File System*, Schvachko et al, MSST, 2010.

Paper review/summary only for HDFS

Action Items/things to remember

Make sure you can access CS744-S19 Piazza

Find your group-mates, NOW (use Piazza)

Tell us about your group (we will share a spreadsheet you can edit)

Keep an eye for the project list (to be released next 2 weeks)