CS 33001 Data-Intensive Computing Systems Project Assignment
Chien -- Spring 2012
April 8, 2012

Project Mechanics

I. Crawl Assignment (1-2 pages), Due April 13
- Identify the systems question
- Identify the data-intensive computing systems infrastructure
- Identify the data set and workload/application drivers you'll use
- Initial feasibility experiments: installed, software-architecture feasible, performance rational (what you did to ensure, data if relevant)

II. Walk Assignment (5 pages), Due April 23
- Crawl Assignment with improvements + new items below
- Full Project Plan (major tasks, schedule – implementation, debug/test, experiments)
- Committed choice of infrastructure, data set, workload/application. You should have all of these in hand, on disk, etc.
- Committed hardware resources for execution of experiments
- Justify soundness for significant choices

III. Run: Project Status Report (check-in, demos, learnings, in-class presentation+discuss)
- Partial demonstrations
- Final adjustments to project plan
- 10-slide presentation, including 1-slide of learnings (4-5 bullets)

IV. Final Project Presentations and Demos Week 10
- Full presentation and demo

V. Final Project Report

Motivation and Focus

Data-intensive computing systems are the foundation of modern internet and cloud systems that underly nearly every aspect of society. These systems not only support large-scale communication – email, text messaging, "tweeting", blogs, wikis, status, "wall"s, lifelines, interests, personal video channels, and a host of other forms – they also underly commerce, government, and nearly every operational aspect of our modern societal infrastructure, supporting monitoring, intelligent management, planning, and operation of air traffic, to package delivery, intercontinental logistics, research, and even multi-government NGO humanitarian aid. These uses give rise to ever increasing challenges in scalability, reliability, capability, and of course cost-effectiveness which demand new technologies and creative new research ideas.

In the context of the course objective to expose students to the technical challenges of data-intensive computing systems, including canonical driving problems, research systems, and emerging technologies. And to provide hands-on experience with a range of systems to provide a solid preparation for research in the area, the following are focusing criteria for your project.

1. All projects should explore a data-intensive computing **systems** issue. Typical examples listed below:
    a. Organizational questions (how to organize the data? Computation? Metadata? System extension? User/application programming?)
        i. Ex: unified or distributed tables (scalable vs. shared nothing relational DB)
        ii. Ex: HDFS/GFS centralized metadata vs. Cassandra's DHT based storage
    b. Architecture and Interface (what are right interfaces for this type system? What guarantees and control at each interface? What do they enable in terms of extension, configuration flexibility, optimization, capability?, what do they prevent?)
        i. Ex: sharding architecture in MongoDB or MemcacheD
        ii. Ex: reliability architecture in PVFS vs. HDFS
    c. Algorithmic and Implementation questions (Given the system organization and function, what are the canonical efficiency and scaling questions?)
        i. Ex: query optimization in SQL databases; query and disk scheduling, scalable locking protocols, non-blocking primitives
        ii. Ex: block-size and read-ahead size in parallel filesystems
    d. Environment and hardware questions (What environments does the system perform well – mobile, server, low power; reliable networking, unreliable; high vs. low bandwidth and latency, etc. What hardware architecture, configuration, ratios, etc. affect performance; can insensitivity to these be improved)
        i. Ex. NVRAM for write caching in Autoraid
        ii. Ex. Local disk for HDFS/GFS
2. All projects will involve an **empirical evaluation** of systems question – and therefore a comparison of a set of alternatives.
    a. Evaluation metrics might include capability (what can be done), performance (scalability, latency, availability), flexibility/portability (what range of systems, heterogeneity, distribution), simplicity, cost-efficiency, etc.
3. All projects should include the use of **a realistic data-intensive computing data set (or sophisticated model thereof), and application computation** across it. These can be workloads designed and implemented by others.

Your project can be motivated by a particular data-intensive computing problem or domain, but should be focused around a data-intensive computing systems question. Your project write-ups should explicitly and clearly identify the question and how the study will provide insight into the answer to the question.