College of Computing

CS 8803
Big Data Systems and Analytics

Ling Liu

Professor
School of Computer Science
College of Computing, Georgia Tech

Theme of this Course

Big Data Systems* + *Big Data Analytics

Big Data Research Areas: CS Perspective

- Big Data Collections
- Big Data Feature Extractions
- Data Representation and Organization
- Big Data Storage and I/O Performance
  - network IO and Storage IO
- Processing and Analysis of Big Data
  - Deep Learning Models and Algorithms
  - Scalable and Fast Implementation Engineering and Optimizations
- Big Data Cloud / Data Center Efficiency
- Security and Privacy of Big Data Analytics

Collection of Big Data

- Collection/acquisition of big data is challenging
  - Difficult to get access to valuable data
  - It stresses the computer system's ability to acquire a lot of data efficiently
It is also difficult to collect useful information from a sea of irrelevant data
  - Data collection should not negatively impact the target system's operation
- An example – Web data collection
  - How to crawl the web efficiently, on the right topic, without affecting the normal uses of the web?

Data Representation and Organization

- Relational databases and SQL
  - Hard to scale for big data, no mainframe is big enough for big data
- Key-value, NoSQL stores
  - Bigtable, Write-optimized
- Specialized indexes
  - Inverted indexes for web search
  - Multi-dimensional data organization & indexing

Storage and I/O

- Storage and I/O are critical for big data performance and reliability
- Hardware: disks, Flash, SSD, non-volatile memory, 3D memory
- Parallelism:
  - RAID, parallel data storage, Distributed file system
- Data durability and consistency

Processing and Analysis of Big Data

- Processing large datasets is time consuming
  - Parallel data processing is necessary
  - But parallel data processing is challenging
- Map/Reduce (Spark or Hadoop)
  - Parallel data processing with easy programming and automatic support of data movement, load balancing, and fault-tolerance
  - Originated in web data processing (counting words); suitable for easily parallelizable workloads
  - But limited semantics
- Threaded and networked data processing in parallel

Cloud / Data Center Efficiency

- Data center today
  - Containing racks of machines and storage
  - Size of warehouses
  - Sometimes built next to rivers because
    - Cheap energy from nearby dams
    - Good corporate image for using renewable energy
- Geographically distributed Data Centers
- Cloud brokering
- High Performance Cluster Computing

Energy and Sustainability

- Energy efficiency in data collection and data centers
- Data center construction from low-power computers
  - Think of a stack of tablets
  - Low joules per unit of work compared to conventional data center
- Data centers on renewable energy
  - Hydro-power, wind, solar, ...
- Minimize environmental harm in field data collection
  - Use renewable energy; no batteries

Data privacy & protection

- Misuse of big data is a big concern
  - Aperson's online activities can reveal all aspects of the person's life
- Systems need to provide clear guidelines on data privacy and protection
  - Sensitive clinical information
- Understand how the big data world operates
  - as an user
  - as a developer

Big Data Collection

Challenges of Big Data Collection

- Challenging to acquire a lot of data quickly
  - Need of large processing, networking, and storage throughput
  - Parallelism can help
- Challenging to acquire useful information from a sea of irrelevant data
  - What data is more important than others
  - Identify redundancy efficiently
  - Collect topic-specific data
- Challenging to collect from distributed, remote data sources

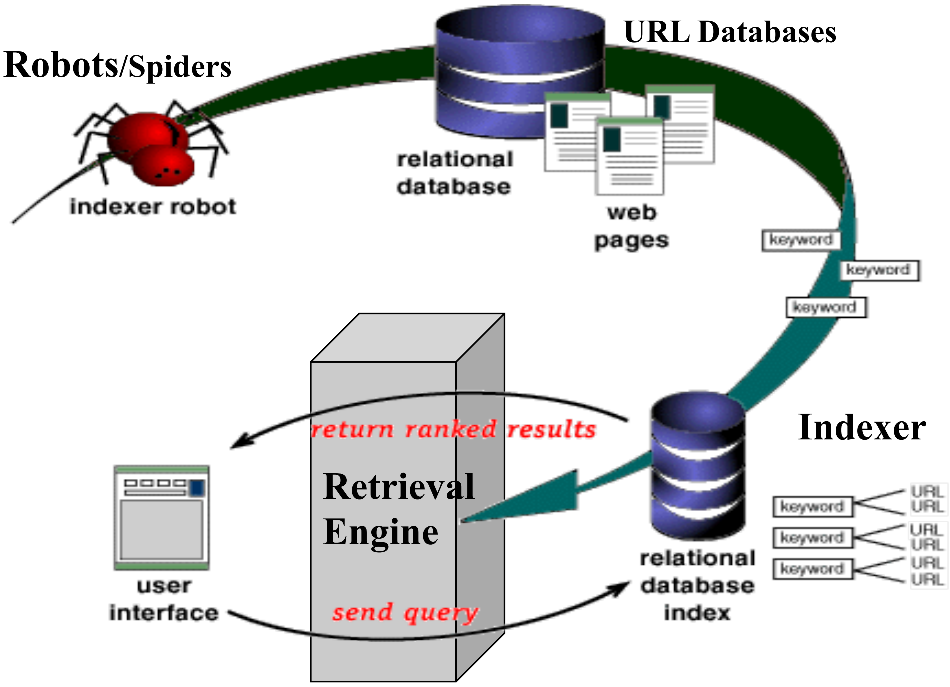
Web Crawling

- Collect published web content – crawling
  - First retrieve some root/seed pages;
  - Parse their content and follow hyperlinks to retrieve more pages;
  - Repeat the last step.
- How is it useful?
  - Web search engines
  - Dive deep (go beyond what is provided by search engines) on a specific topic
  - Businesses/advertisers to find potential customers

Goal

- Collect good (high-quality) pages
- Collect web pages on a certain topic
- Crawl efficiently
- Crawl without annoying others

How does a SE work



Functionality of A Crawler

- Crawlers are programs that
  - go out to the Web to gather information about the content of pages from sites and
  - feed that information to the search engine’s indexing program
- Two main tasks:
  - identify new sites that are to be added to the search engine, and
  - identify sites already covered that have been changed since the last visit, detect changes if any.
- Is Web Browser a crawler?

How does a Crawler work?

- Different crawlers use different strategies for document retrieval
- Standard Process
  - starting from a historical list of URLs (a known set of documents), e.g., server lists, what-is-new pages, and the most popular sites on the web
  - examining the outbound links from them
  - following one of the links that leads to a new document
  - then recursively repeating the whole process

Robot: An example

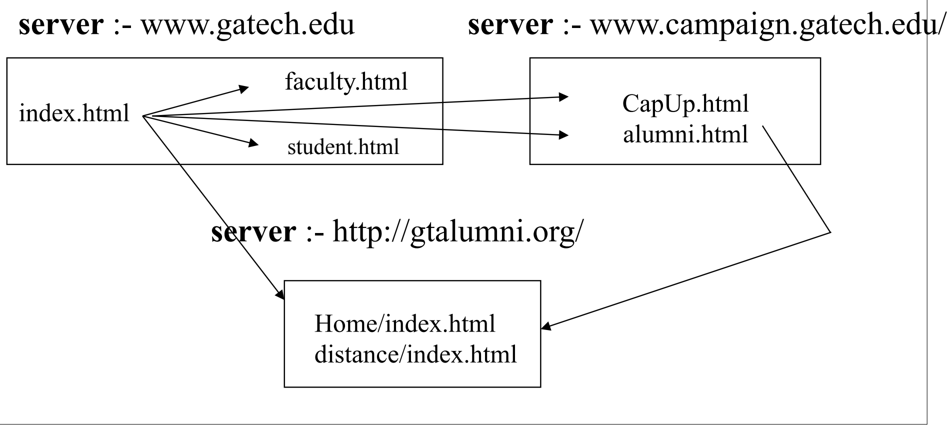
- Starts with a known list of URLs (seed list)
- Example :

Initial URL List :-

www.gatech.edu/
www.cnn.com/news.html
www.cnet.com/cars.html
www.intel.com/
www.microsoft.com/
:
:

A Crawler’s View

- Visualization of Web as a graph



server :- www.gatech.edu

index.html

faculty.html

student.html

server :- www.campaign.gatech.edu/

CapUp.html
alumni.html

server :- http://gtalumni.org/

Home/index.html
distance/index.html

How does a Robot Crawl the Web?

- Breadth-First Navigation strategy
- Depth-First Navigation strategy
- Hybrid strategy
  - use both interchangeably

Crawler (Cont.)

- BFS (Breadth First Traversal)

URLs to be visited

www.conte.gatech.edu
www.gatech.edu/techhome/subpgs/faculty.html
www.gatech.edu/techhome/subpgs/student.html
www.gatech.edu/techhome/subpgs/research.html
www.campaign.gatech.edu/
…...
www.cnn.com/news.html
www.cnet.com/cars.html
www.microsoft.com/products.html
www.intel.com/content.html
www.intel.com/intel/product.html
:

Breadth-First

- Pros
  - when used with an initial seed list like the official registry of servers, yields excellent results at first, because it reaches many different servers.

- Cons
  - In general, less efficient at penetrating the web deeply, going far beyond the starting points.
  - may lead to periods of putting heavy load on a single server

Robot (Cont.)

- DFS (Depth First Traversal)

URLs to be visited

www.gatech.edu/research.html
www.cc.gatech.edu/research.html
www.cc.gatech.edu/systems/
www.cc.gatech.edu/systems/people
www.cc.gatech.edu/~lingliu
...
www.intel.com/content.html
www.intel.com/intel/product.html
...
www.cnn.com/news.html
...
www.cnet.com/cars.html
...
www.microsoft.com/products.html
:

Depth-First Navigation

- Pros
  - gives the best overall distribution of URLs over the Web site (seed URL), which is important only when a small part of the Web can be retrieved
- Cons
  - infinite loops:
    - generates the danger of leading a robot into infinitely recursive document trees on servers that generate documents dynamically on the fly.

How to find the seed list for Robot

In addition to a server list of URLs,

- Most of indexed services also allow users to submit URLs manually
    - (e.g., www.submit-it.com), which are then visited by the robot
- Other sources for obtaining a list of seed URLs
    - e.g., scanners which scan through USENET postings and published mailing list archives
- Main issue:
    - How to penetrate the Web faster
    - how to detect and prevent “spam” cases ...

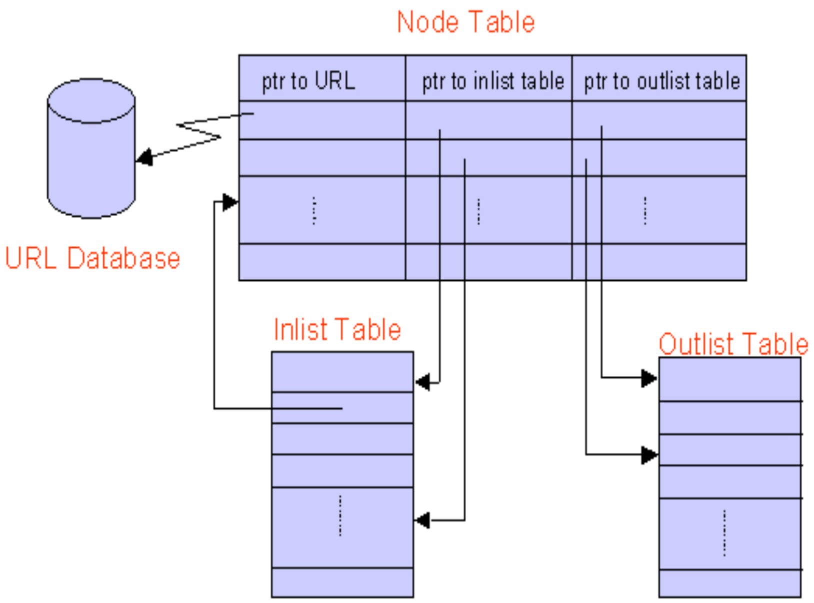
Redundancy Removal

- Avoid parsing and following the same page more than once
    - Record all URLs that have been parsed and followed
    - Same page may have different URLs
- URL normalization
    - Host portion is case-insensitive
    - Decode percent-encoded octets of unreserved characters (‘%30’ is ‘0’)
    - ‘/./’ or ‘/’
    - … …
- URLs that look totally irrelevant may also be the same page
    - Compute and match page content checksum

Link Selection

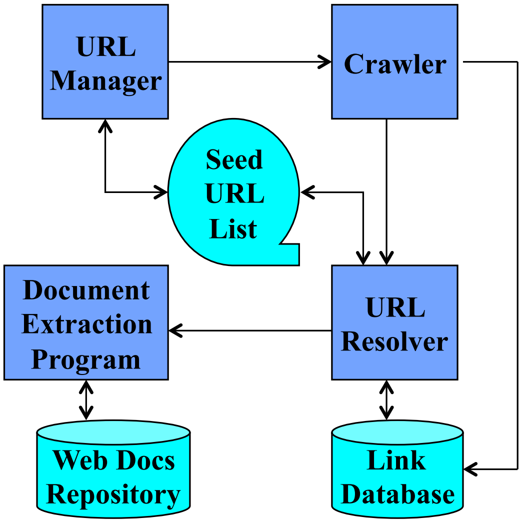
- There are several choices at each step of crawling
  - Depth-first search vs. breadth-first search?
  - Seed list selection
  - ……
- Hyperlink with high likelihood pointing to a high-quality page?
  - High in-links
  - Pointed to from high-quality pages
  - Spam identification and avoidance
  - ……

URL Database: Graph Data Structure



General Architecture for Web Robots

- URL Manager/ Server
  - Seed URL list
- Crawler
  - BFN or DFN
- Indexable Text Extraction Server
  - Extracted Web page repository
- URL Resolver
  - Link database



Crawling Courtesy

- Crawling is not always welcome (in fact, often unwelcome) by web sites
  - Some of my content is good to show up on a search engine, others (processing scripts etc.) has no use (to me) to be crawled
  - Crawlers/bots consume my web server resources that may affect the experience of human users
- Robot Exclusion Standard
  - Do not crawl with Robot Exclusion turned on
- Limit the crawler bandwidth use
  - Page download rate; downloading bandwidth
  - Limit per site

Example Web Crawler

- Steve Proell
  - Spring 1998
  - PULSE

    A Web Crawler for Intel's Intranet

- Todd Miller
  - Fall 2000
  - HyperBee: A P2P crawler for the Web

- Apoidea: A decentralized P2P crawler
  - Aameek Singh and Mudhakar Srivatsa, Spring 2002
- PeerCrawl
  - VaibhavJ.Padliya (2005)
  - Mahesh Palekar and Joseph Patrao (2006 Spring)
  - Tushar Bansal (2006 Fall), Suyang Li (2006 Summer and Fall)

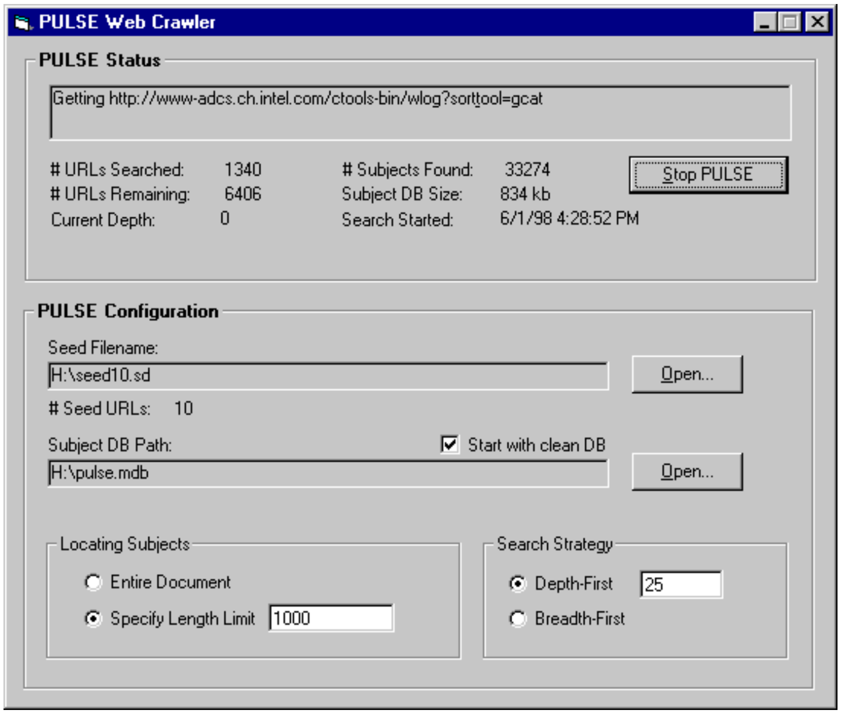
Motivation & Objectives

- Intrigued by the discussions on web crawlers in class
  - How big of a problem are we talking about here?
- Objectives:
  - To quantify the amount of work performed in building a web crawler or a search indexer.
    - How many URLs are in Intel's intranet?
  - To understand the implementation requirements for a web crawler.

Project Components

- A semi-configurable web crawler
  - Visual Basic 5.0
- A search database to contain URLs searched and subjects found
  - MS Access
- A search engine interface
  - HTML/ASP

Project Components (2)



PULSE Web Crawler

PULSE Status

Getting http://www-adcs.ch.intel.com/ctools-bin/wlog?sorttool=gcat

URLs Searched: 1340
URLs Remaining: 6406
Current Depth: 0
Subjects Found: 33274
Subject DB Size: 834 kb
Search Started: 6/1/98 4:28:52 PM

PULSE Configuration

Seed Filename: H:\seed10.sd

Seed URLs: 10

Subject DB Path: H:\pulse.mdb

☑ Start with clean DB

Locating Subjects
○ Entire Document
● Specify Length Limit 1000

Search Strategy
● Depth-First 25
○ Breadth-First

Technical Aspects

- Pseudo Depth-First Search

```
Add seed URLs to list, set 1st URL as NextURL
While search list not empty
  Retrieve HTML page specified by NextURL
  Scan page and add subjects to DB
  If page has links
         Add the 2nd-nth links to the list
         Set NextURL to 1st URL
  Else
         Set NextURL to first on list
  End If
End While
```

- Sounds easy, huh?

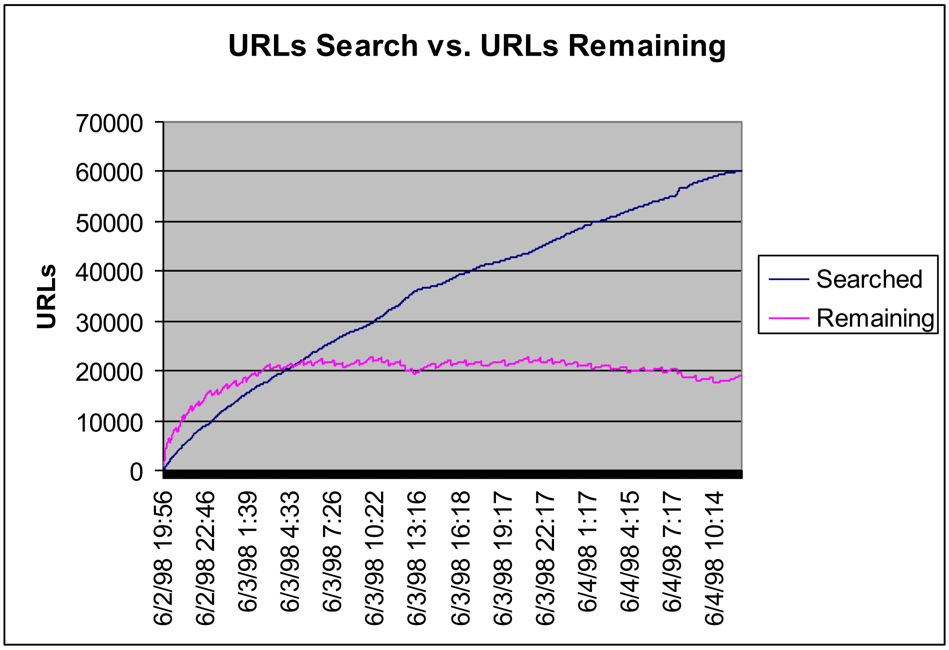
Technical Aspects (2)

- It’s *not* easy to:
  - Scan for URLs
    - How do you handle relative links?
    - What about a frame set?
    - a MAILTO tag?
    - etc.
  - Scan for subjects
    - What words should we choose?
- What do you do if the page contains incorrect HTML, e.g., a missing tag?

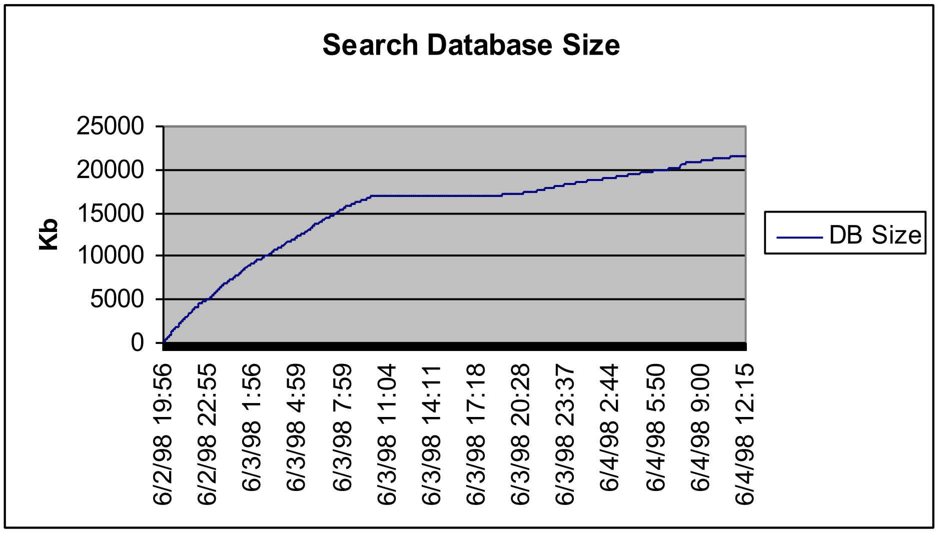
Results

- The Internet is BIG.
- Example:
  - Searching from 1 seed URL: www.intel.com
  - Only scan first 1000 characters per page
  - Started June 2 at 8pm. Still running as of June 4 at 12:30pm.
  - Searched ~60,000 pages, ~20,000 remaining on the search list.

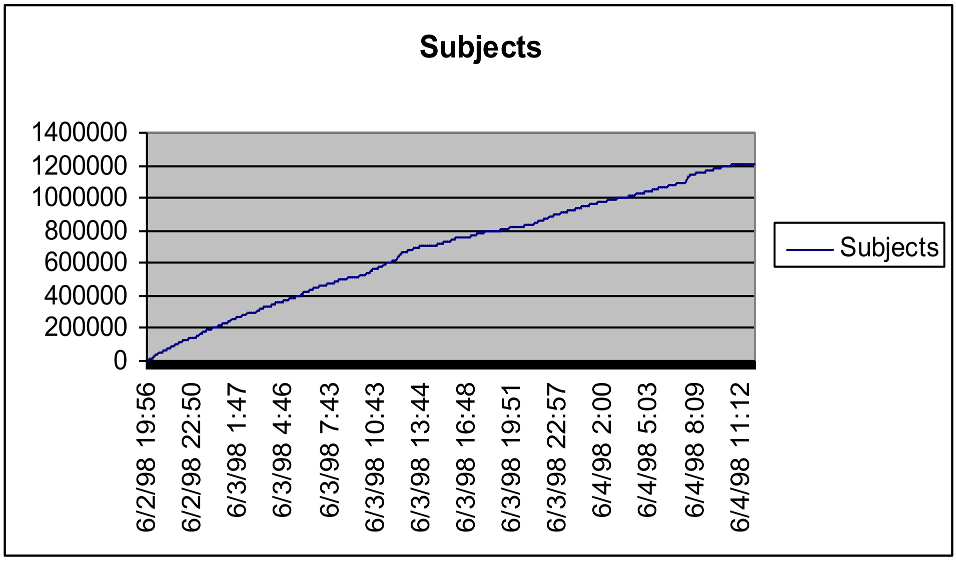
Results (2)



Results (3)



Results (4)



Results (5)

- Think about this:
  - **Scenario 1:**
    - To crawl 200,000,000 pages in the Internet
  - **Scenario 2:**
    - how long to crawl 1 billion pages?
- Using PULSE web crawler for scenario 1:
    - It would take 15.5 years to search all 200 millions of URLs
    - It would scan 4,028,292,765 subjects
    - The database would require 71GB of storage.

Key Learnings

- A web crawler must be robust. It must:
  - be restartable
    - What happens if the web crawler is fine but DB crashes?
  - be able to detect loops in the search list
    - Imagine a site with a navigation bar at the top of every page
  - be able to handle many different file types
    - doc, ppt, xls, ps, gif, pdf, exe, zip, *etc.*

Key Learnings (2)

- How should the subject database be structured?
  - How much of a given page do you store?
  - Consider a page that invokes a CGI script with a series of parameters. Imagine there are several pages that link to this page with different parameters.
    - Do you store the result page once or every time?

Focused Crawling

- Maximize the page downloads on a certain target, minimize the resources spent on irrelevant downloads
- Crawl all pages from a domain (e.g., rochester.edu)
- Topic-specific crawling:
  - A page's topic is inferred from the URL text, anchor text, and surrounding text of the hyperlink
  - Look for specific keywords
  - Reinforcement learning [McCallum, Rennie, et.al 1999]
    - Train a Q function: mapping from "bag of words" to a value (future rewards – chance of hitting on-topic pages if following the link)

Scalable Web Crawling

- What are the resources consumed?
  - CPU processing for network operations and the parsing of page content
  - writing to disk storage
  - network bandwidth to remote web sites
- Parallel web crawling
  - Use multiple CPUs
  - Use multiple storage devices
- Challenges:
  - Synchronization on already visited URLs
  - Downloading network bandwidth remains the bottleneck

Example Assignment

- Focused web crawling → A strawman version
  - Within a domain
  - Topic-specific
- Follow the robot exclusion standard
- No parallelism
- Optional
  - Not to visit the same URL twice, URL normalization [Optional]
  - Crawl at a determined slow rate (e.g., sleep a second between consecutive page downloads)

Example Assignment (cont.)

- Parsing a web page for hyperlinks and anchor texts?
  - HTML pages are notoriously error-pronebrowsers go to great length to tolerate imperfect HTML pages; you may find it not easy to do
- Maintaining the list of visited URLs → what data structure will be used?
- Programming languages:
  - Java, C, Perl, Python, …
- Deliverable: Crawl Statistics + ScreenShots

Big Data Collection: Sensors, IoT

- Example 1: USArrayseismicdatacollection
  - http://www.usarray.org/

- Challenges of data collection at remote locations
  - No power infrastructuresolar panels + batteries
  - Reliability (batteries died; GPS malfunctioning)
  - GUI

Big Data Collection: Sensors, IoT

- Example 2: High-speed cameras on roads and highways

- Challenges
  - A lot of data produced by high-speed cameras
  - Wireless deployment is economically efficient, but wireless networks are not very good (low bandwidth, intermittent)
  - Processing on site to relieve the burden of data transfer

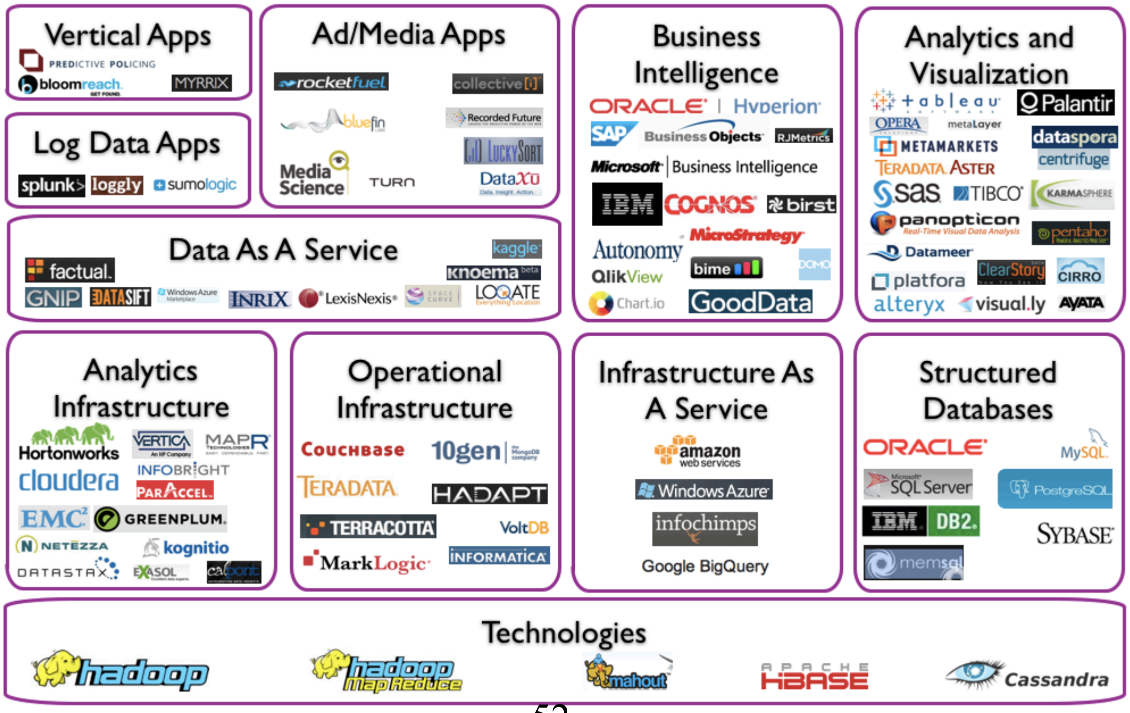
Public Eavesdropping

- Listen to open WiFi signals
  - Lots of communications are unencrypted
  - WEP encryption is weak

- Insert a Gnutella relay node to collect information on Gnutella traffic
  - Know what are being searched
  - Know the group of searches made by one person

What Technology Do We Have For Big Data ??

Big Data Landscape



Vertical Apps

Log Data Apps

Ad/Media Apps

Data As A Service

Business Intelligence

Analytics and Visualization

Analytics Infrastructure

Operational Infrastructure

Infrastructure As A Service

Structured Databases

Technologies

dave@vcdave.com blogs.forbes.com/davefeinleib

Course Structure

- Lectures
  - by Instructor + invited guest speakers
  - Discussions and interaction in the class
- Your Part
  - Homework (4)
    - Reading + writing critiques
    - Programming
  - Project (team):
    - proposal, implementation, demo, workshop presentation
  - Technology Review (final exam)
  - Participation in the class

Administravia

- Office Hours
  - Thursdays 11am -12noon, or by appointment
- Grading

Class Participation	15%
Homework Assignments (4)	20%
Project	50%
Final (Technology Review)	15 %

- Announcements
  - In class and on T-Square
- Course Materials
  - No textbooks Required
  - Course Notes + A collection of papers, available on Tsquare
- Useful and Related Links
  - TSquare postings under resource

Homework Assignments

- 4 homework assignments total (starting in the 3rd week)
  - Posted on Monday of Week 3, 6,8, 10
  - Due on Friday of Week 4, 7, 9, 11
- All homework assignments are individual and should be completed independently
- Two Types of Homework Assignments (your choice)
  - Reading Assignment
  - Programming Assignment

Reading Assignment

- For each reading assignment, you choose two papers from the list of recommended readings.
- Two papers should be related to one subject.
- You are asked to write two reading critiques, one for each of the two papers you read.
- Submit your reading summary on TSquare by the due date.

Suggestions and Tips for Reading Summary

- Read each paper at least twice
- Comments on the following questions:
  - ✦ What is the subject area?
  - ✦ What are the general problems and specific problems this paper intends to address
  - ✦ Describe the main ideas and techniques of the paper
  - ✦ Summarize the strong and weak points
  - ✦ Does the paper solve the problems promised?
  - ✦ Is there any other solutions?
  - ✦ What are your suggestions for revision and/or what are your insights for the problem addressed?

Reading Summary Template

CS8803/CS4365 Homework Reading Summaries

Student ID:

Family Name:

Given Name:

Paper Title:

Summary/Critique

(1) Problems

(2) New Idea and Strengths

(3) Weaknesses and Extensions

Reading Summary - Grading Scheme

- Grading scale: 0-100 points
- Pass (>=60)
  - a good summary to the assigned reading
- Pass – (<60)
  - in limited circumstances in which either the summary is incomplete or it is clear from the summary that the student did not read the material or the summary contains poorly worded criticism.
- Pass + (>90)
  - a summary in which the student goes beyond a basic review and presents insight that is thoughtful and well expressed.

Programming Assignment

- For each programming assignment, you are required to choose one task from the list of recommended programming tasks.
- The Programming Assignment should be submitted to Tsquare by the due date
- Deliverable includes the following 4 items
  - Source code + Read Me (or open source URL)
  - Executable code
  - Inputs and outputs of your program (e.g., datasets used)
  - Flow Chart of your program

    https://en.wikipedia.org/wiki/Flowchart

    http://users.evtek.fi/~jaanah/IntroC/DBeech/3gl_flow.htm

Programming Assignment: Grading Scheme

- Grading scale: 0-100
- Pass (>=60)
  - a complete package of deliverable with quality documentation
- Pass + (>= 90)
  - a complete package of deliverable with quality documentation
  - The program presents some novelty in design or implementation or the documentation is superb and GUI is elegant.
- Pass – (<60)
  - The program did not provide the full functionality as required
  - The documentation is incomplete or poorly written

Project

- Interesting, Novel, and value-added ideas/Concepts
- Innovative engineering/executions
- Four Components
  - Proposal (20%)
  - Workshop Project Presentation (20%)
  - Final Project Demo + Deliverables (60%)

Two Types of Projects

- Big Data Systems Projects
  - Novel in System Design, Optimization and Engineering
  - Novel in Problem Identification and/or Solution Approach
- Data Analytics Projects
  - Novel in Analytic Problems to be solved
  - Novel in Algorithm design and implementation methods (performance and analytic quality/accuracy)
  - Novel in Applications being deployed

Project

- Project team: 3-4 persons per team
- Count 50% of your total grade
- Four Types of Deliverables
  - Project Proposal
    - proposal submission, meeting with instructor, proposal revision
  - Project presentation at the workshop
    - Every team member must be present at the project presentation
  - Project Demo
    - Every team member must be present at the demo
  - Final project deliverable

Project (cont.)

- Project Requirement
  - Must be topics related to Big Data Systems or Application-specific Analytics
  - Make sure that your project exhibits innovative ideas or novel applications.

- Proposal Due on Friday of Week 5 (Sept 23)
  - 2-3 pages (in pdf or word)
  - covering problems to be addressed, concepts, techniques, system architecture, and the hardware/software environment to be used.
  - A statement with adequate elaboration on why the proposed project is innovative and useful

Project (cont.)

- Project team meeting with Professor
  - Tuesday of Week 6 (Sept 27)
  - Sign-up on TSquare course Wiki
  - Location: in the small conf. room across my office KACB 3340

- Selected project proposal presentation
  - Invitation in Week 6
  - Presentation on Tuesday of Week 7

- Proposal revision if required due by Friday of Week 8

Project (cont.)

- Final Project Presentation - Workshop
  - 15 ~ 20 minutes, incl. 5 min questions
  - emphasize on new ideas and the most interesting components of your project
  - Present evaluation methods and lessons learned
  - Every team member must present your contribution as a integral part of your team project.

Project (cont.)

- Final Project Demo
  - 15~20 minutes in the small conf. room across my office KACB 3340
  - What to hand in?
    - In-class project presentation (ppt), submit to TSquare
    - Project Deliverable, submit to TSquare
      - Final Report (ppt or word, highlights, figures, Screen shots)
      - Source Code with ReadMe
      - Executable (using .tar file, compressed)
      - Datasets used for measurement and demo (Input)
      - Data output + Performance measurements

Technology Review

- Topic Selection
  - Topics covered or highly related to the topics covered in the courses, incl. lectures and homework reading and programming assignments.
  - can be combined with the theme of your course project (optional).

- Structure
  - Should cover the state of art in the selected technology area.
  - should contain a discussion section describing your thought and your prediction on the deployment and adoption of the technology being reviewed.
  - Should provide a list of major references that you have used in preparing your review.
  - The expected length of the technology review is about 10~15 pages, single column and single spacing (1.2 pt). Figures are welcome.
  - Due date: Final exam day: **Dec 15 (Thu) 3pm**

Questionnaire

- Fill in the course questionnaire
  - Accessible on Tsquare for those who have registered the course
  - Upload your answer to Tsquare https://t-square.gatech.edu/

Questions