CS 686: Special Topics in Big Data

Networking Fundamentals

Lecture 4

Today's Agenda

- Discussion: group work from previous class
- Designing a simple distributed system
- Reviewing the paper evaluation form
- Project 1

Today's Agenda

- **Discussion: group work from previous class**
- Designing a simple distributed system
- Reviewing the paper evaluation form
- Project 1

Today's Agenda

- Discussion: group work from previous class
- **Designing a simple distributed system**
- Reviewing the paper evaluation form
- Project 1

Designing a Distributed Execution Framework

- Let's say we are going to make a simple system that executes programs on multiple machines in parallel
- The user gives us a list of *tasks*, or jobs to run on the systems:
  - cd /home/me/somewhere; ./my_prog –n 1 > output.txt
  - cd /home/me/somewhere; ./my_prog –n 2 > output.txt
  - …
- What components/messages do we need?

Components

- Coordinator
  - Holds the list of tasks to run
  - Assigns tasks to machines
- Task Server
  - Retrieves tasks from the coordinator
  - Executes tasks
  - Tells the coordinator when finished
- Client
  - Provides tasks to the controller

Messages/Events

- Task specification
  - Coordinator → Task Server
- Task request
  - Task Server → Coordinator
- Task completion notification
  - Task Server → Coordinator
- New task import request
  - Client → Coordinator
- New task import response
  - Coordinator → Client

Things to Think About

- What about fault tolerance?
- Should the Task Servers be continuously connected to the Controller?
  - Why or why not?
- Could we add administrative messages that tell us how many tasks are complete, number of errors, etc?

Today's Agenda

- Discussion: group work from previous class
- Designing a simple distributed system
- **Reviewing the paper evaluation form**
- Project 1

Paper Evaluations

- By now hopefully everyone has seen the link on the schedule for Paper 1
  - K. Shvachko et al.,*The Hadoop Distributed File System*
- Due tomorrow at 6pm
  - This will give us a chance to get used to submitting through canvas
- Use the evaluation template to guide your write-ups

Evaluation Template (1/2)

- Tweet
  - Short, sweet
- Summary
  - Remember your audience: folks who have already read the paper
- Critical Analysis
  - Evaluate design choices
- Questions
  - If the author was here, what would you ask them?

Evaluation Template (2/2)

- Aim for being thoughtful, not verbose
  - 1-1.5 pages
  - I'm not in the business of checking margins, number of words, etc
- Cite sources (if any). For example:
  - Finding a follow-up paper that went more in depth
  - A paper that refutes the points made in the paper under review

Today's Agenda

- Discussion: group work from previous class
- Designing a simple distributed system
- Reviewing the paper evaluation form
- **Project 1**

Project 1

- For our first project, we'll design our own version of HDFS
  - Also similar to GFS
- Design document is available online
  - Includes a link to set up your GitHub repository
  - Implement in **Java**
- The system will allow you to store and distribute files over a set of machines

Functionality (1/3)

- Each file will be broken up into multiple **chunks**
- Chunks are distributed to the **Storage Nodes**
- When you want to retrieve a file, you'll:
  - Contact the **Controller** to find out where the chunks for the file are
  - Retrieve the chunks in parallel from the storage nodes
  - Reconstruct the file on the client side

Functionality (2/3)

- The system will also be fault tolerant
- Each chunk is replicated twice for a total of three chunks
- If a storage node goes down, you can locate and use other chunks in the system seamlessly
  - You will also maintain the replication level by creating more copies

Functionality (3/3)

- To detect failures, storage nodes will periodically send **heartbeat** messages
- Heartbeats also include metadata about what chunks the storage nodes are currently maintaining
- Finally, to communicate we'll be using Google **Protocol Buffers**
  - Popular data interchange/serialization format
  - We will have an in-class lab session on using protobufs

Logistics (1/2)

- I will continue to update the spec with clarifications/hints as things come up
  - But I will not modify the requirements mid-project
- There is also a list of **milestones**
  - Suggested times for completing functionality
- We'll have in-class lab sessions to discuss and work on the projects
  - You can turn in milestones early during the lab if you'd like to get feedback

Logistics (2/2)

- This is a large project:
  - 20% of your grade
  - Lots of coding
- Start early and ask questions!
  - We'll use Piazza for online discussions/help
  - Come by office hours or email

Questions/Concerns

Also see the project description online:
https://www.cs.usfca.edu/~mmalensek/courses/cs686/projects/project-1.html