Distributed HTTP

Athula Balachandran

Today's Agenda

- Overview of the project
- Overview of Checkpoint 1
- Python Resources Needed
- Routing Daemon Design

Logistics

- Checkpoint 1 – Due on Sunday (30th Oct)
    - Sunday better? More time after recitation?
- Finding partners
    - Bboards
    - Else email me
- Email me your group details
    - Partners in the team
    - What is your repository name?
        - <andrewid>-<andrewid2>-15-441-project2
        - eg. abalacha-wolf-15-441-project2

Checkpoint 1 Details (Website)

Useful Python Resources for the Checkpoint

- Flask webpage: http://flask.pocoo.org/
- Flask Tutorial: http://flask.pocoo.org/docs/tutorial/
- Check the following functions
  - request.args.get() - to parse arguments from the GET request
  - request.form.get() - to parse form arguments in the POST request
  - request.files() - to get the file data
- Uploading a File: http://flask.pocoo.org/docs/patterns/fileuploads/
- Reading and Writing from files: http://docs.python.org/tutorial/inputoutput.html
- Hash: http://docs.python.org/library/hashlib.html
- Socket Programming: http://docs.python.org/howto/sockets.html
- Split: http://docs.python.org/library/stdtypes.html

Handling Large Files

- Get the file
  - file = request.files['uploadFile']
- Store it in a temporary file
  - tmpname = tempfile.mktemp(prefix='.../static/')
  - file.save(tmpname)
- Read 4096 bytes at a time and keep updating the hash
  - hash = hashlib.sha256()
  - with open(tmpname, 'r') as f:

        hash.update(f.read(4096)) # loop
- Move the file to the new location computed using the hash digest
  - shutil.move(tmpname, '.../static/' + hash.hexdigest())

Routing Daemon

- Read the config file and the file list file
- From the config file and the node-name, figure out the peername, server port, local port and routing port.
- Start a TCP server at localport and wait for requests from the Flask application.

Routing Daemon Requests

- GETRD <object-name>
  - eg. GETRD athula
  - For checkpoint 1, go through the local file-list.
  - Suppose there is an entry
    - athula /static/e627abdec8
    - Return: “OK http://”+peername+”:”+serverport+”/static/e627abdec8”
    - OK http://127.0.0.1:5000/static/e627abdec8
    - (Note that the e627abdec8 is not a valid sha256sum, it is just a placeholder. Sha256sums are much longer than this)
    - If you didn't know, you can do sha256sum <filename> on your commandline and compare the sha256sum against the one generated by the python script.
  - Suppose there is no entry
    - Return “404”

Routing Daemon Request

- ADDFILE <object> <local-relative-path>
  - (Happens at the front end node)
  - ADDFILE wolf /static/172356adcc107f8
  - Add this to the object, path mapping that is being stored
  - Response: OK
  - Response: 404 (say, object does not match the criteria or some other error.)
  - You may be replacing an already existing object to path mapping

Routing Daemon Request

- Any other request or not well-formed requests
  - Send “501”

Project 1 General Overview (Blackboard based Discussion)