Final Project: Proposal

Objective: Solve a problem that requires the scale and power of the cloud. Work through course content to devise a technical solution to solve the problem.

The final project is a substantial portion of a student's grade, and thus we have broken the final project into many steps (proposal, two checkpoints, final presentations and paper). The first part of the final project is to find an interesting problem and to outline the high-level solution.

Updates:

- Dec 11: Some clarifications from Piazza added to report instructions
- Dec 6: updated turn-in instructions for class presentations
- Dec 4: Final project report instructions added at bottom of doc
- Dec 4: Final project presentation instructions added at bottom of doc
- Nov 27: Checkpoint 2 instructions added at bottom of document
- Nov 11: Checkpoint 1 instructions added at bottom of document
- Oct 23: Turn-in instructions updated
- Oct 23: Due date extended to Tuesday, Oct 30
- Please reference this document frequently for updates.

Details:

Final projects can be performed in **groups up to 4 people in size**. Unlike the first three projects, the **students can choose to form the groups themselves.** Any group size less than 4 is allowed, but students are encouraged to work in large groups.

In general, students have a lot of flexibility to define their own projects. Projects can range from optimizing some low-level infrastructure within the datacenter (say someone wants to create a new datacenter-based TCP congestion control algorithm) to stitching together the technologies that we learned about in the class to build some sort of solution, service, or application.

The first thing each group should ask is: What problem do I want to solve? Solving a problem requires lots of components, so at the least students should define and outline what sort of datasets they'll use and also what type of software solutions (things we learned about in class) will be used to build the solution. The architecture of the system should be defined. In other words, how will all of the solutions (maybe a database that connects to Spark that utilizes a machine learning library) fit together?

One of the biggest challenges will be finding interesting and cool datasets to work with. I've listed a few potential datasets below, but students are encouraged to follow their own interests and see where that leads. If anyone else has any other interesting lists, please post below.

Tools:

- https://blog.gdeltproject.org/peering-into-the-visual-landscape-of-half-a-billion-news-images-with-googles-cloud-inference-api/
- Data repo sources:
  - Open data blog:
  - https://www.nextgov.com/topic/ng-open-data/
- Some stackoverflow data:
  - https://github.com/seahrh/stackoverflow-spark
  - http://jmcauley.ucsd.edu/data/amazon/
- https://cseweb.ucsd.edu/~jmcauley/datasets.html
- http://bytequest.net/index.php/2017/01/03/freely-available-large-datasets-to-try-out-hadoop/
- https://www.yelp.com/dataset
- https://www.reddit.com/r/datasets
- Click events
  - http://recsys.yoochoose.net/challenge.html
- Kaggle hosts lots of datasets
- MSR MyLifeBits project (picture every 1sec)
- https://toolbox.google.com/datasetsearch
- Amazon Public Data sets
  - https://registry.opendata.aws
- Data lakes:
  - https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/
- http://infochimps.org
- https://www.quora.com/Data/Where-can-I-find-large-datasets-open-to-the-public
- https://www.kdnuggets.com/datasets/index.html
- https://www.google.com/publicdata/directory
- ImageNet (http://image-net.org)
- http://snap.stanford.edu/data/index.html
- https://en.wikipedia.org/wiki/Wikipedia:Database_download
- https://planet.openstreetmap.org
- https://stackoverflow.blog/tags/cc-wiki-dump/
- Metalist:
  - https://hadoopilluminated.com/hadoop_illuminated/Public_Bigdata_Sets.html
- Developer links often have data access:

    - https://developer.twitter.com/content/developer-twitter/en.html
    - https://developers.google.com/maps/documentation/
    - https://www.instagram.com/developer/
    - https://developers.facebook.com
- Build your own dataset with IoT sensors
    - https://www.instructables.com/id/Esay-IoT-Weather-Station-With-Multiple-Sensors/

Turn in:

Students should turn in a **pdf file** with their proposal. The proposal should have **at least** the following sections:

- Problem description (at least half a page)
    - Clearly outlining the problem each group is trying to solve.
    - Motivating why the problem is an important problem to solve.
- High-level solution architecture (about one page)
    - Students are encouraged to draw a graph that pictorially shows how system components will interact
    - Students should provide a description, in text, of how the components of their solution will work together. If necessary, outline configurations or parameters that may need to be set or tuned.
- Dataset (at least ¾'s of page)
    - What datasets will be used for the project?
    - Describe how the data looks:
        - What format is it in
        - Does the data need significant preprocessing?
        - Describe dataset: is it streaming, static, or otherwise?
        - How will the data be accessed? Is a developer account needed.
        - Where will the data be stored and how?
- Challenges (at least half a page)
    - Why is the problem hard to solve?
    - What challenges will students have in defining the solution?
- Timeline (at least half a page)
    - Rough time estimates of what will be done, and by when
    - What part of the project each person will be responsible for, and each person's timeline
- Concerns (optional)
    - Concerns or issues students have or anticipate with their final project

Please note: I may update the outline as needed. Please refer to this document (and check piazza) frequently.

Use github classroom to turn in the proposal and any accompanying documents:
https://classroom.github.com/g/K1dyn6Zm

Project proposals are due on ~~Oct 25,~~ Tuesday, Oct 30 11:59:59 PM.

Checkpoint 1

Checkpoint 1 is due on November 16. Use the same github repo as you used for the final project proposal. You should name your file "checkpoint1.pdf". The file turned in should be PDF format!

Some notes:

- Please make any changes to your proposal document that I may have requested with the proposal feedback. Highlight these changes in a different color so they are easy for me to find.
- Make a new section called Checkpoint 1 and ***append*** your previous proposal document with the new checkpoint information.
- Checkpoint 1 documents should have one section that clearly states any changes to the proposal. An example here might be if you had to change the scope of your project based on either my feedback, or based on some of your initial work. If there are no changes, then you can simply state that.
- You should have one section in the checkpoint that details a new timeline. This section should **very clearly** document the work that you have done so far. In addition, the description should **very clearly** describe the work to be done.
    - For the work done so far, **this must be done for each project member.** In other words, each project member must document their progress.
    - You should include a screenshot of your Github check-ins. Students are expected to be actively coding throughout the project timeline. Include a brief description about the check-ins.
- You should include one section on costs. Detail any costs that you have used thus far and estimate costs for future project development. Remember, I still have Google Cloud credits for the class.
- You should include one section on dataset management. Describe any issues you may have with your dataset. What sort of scrubbing, cleaning, etc you had to perform on the dataset. Describe any outstanding work to be done in this area.

Checkpoint 2

Checkpoint 1 is due on Dec 4. Use the same github repo as you used for the final project proposal. You should name your file “checkpoint2.pdf”. The file turned in should be PDF format!

Some notes:

- **Make a new section** called Checkpoint 2 and *append* your previous proposal document (and checkpoint) with the new checkpoint information. The checkpoint2.pdf file should contain your proposal, checkpoint 1, and now checkpoint 2.
- *New instructions for this time:* **Include a section** that contains a description of any meetings your team may have had (time, dates, lengths, participants, and topics covered). You don’t need to get detailed, just a rough set of notes is fine. I want to see how often teams are working together. A video chat or email correspondence is also ok.
- Checkpoint 2 documents should have **one section that clearly states** any changes to the proposal. An example here might be if you had to change the scope of your project based on either my feedback, or based on some of your initial work. If there are no changes, then you can simply state that.
    - If you are making any changes to your proposal, then highlight these changes in a different color so they are easy for me to find. Also mention
- You should have **one section in the checkpoint that details a new timeline**. This section should *very clearly* document the work that you have done so far. In addition, the description should *very clearly* describe the work to be done.
    - For the work done so far, **this must be done for each project member.** In other words, each project member must document their progress.
    - You should **include a screenshot of your Github check-ins.** Students are expected to be actively coding throughout the project timeline. Include a brief description about the check-ins.
- You should include **one section on costs**. Detail any costs that you have used thus far and estimate costs for future project development. Remember, I still have Google Cloud credits for the class.
- You should include **one section on dataset management**. Describe any issues you may have with your dataset. What sort of scrubbing, cleaning, etc you had to perform on the dataset. Describe any outstanding work to be done in this area.
- You should include **one section on any challenges** your team is facing. This can be brief if things are going smoothly. Remember, you can post to Piazza if you are having issues-- utilize the knowledge of the class!
- Remember, the point of the checkpoints is for me to assess the amount of work being done. The checkpoints should be at least 2-3 pages in length. Many teams’ checkpoints 1 were over this length. *If I can’t tell what you’ve done because the document is not descriptive enough, I will deduct points.*

In-class Project Presentations

In class presentations are set for the week of Dec 11 and Dec 13. Due to the large number of groups, we will have very short presentations. **Please carefully read the instructions for the presentations below:**

- All presentations must be **4 minutes** in length. We do not have time for longer presentations, unfortunately. I will set a timer and cut off any presentation that is longer than 4 minutes. Presentations can be followed by a quick Q&A.
- Due to short presentation time, all presentations must be done in **Google slides**.
- **At 4pm, the day before the lecture, presentations are due.** I need time to put all presentations into one large document.
    - Presentations turned in after 4pm will be deemed late.
    - Dec 11 lectures: slides due to instructor (~~via email~~) at 4pm on Dec 10
    - Dec 13 lectures: slides due to instructor (~~via email~~) at 4pm on Dec 12
    - Please turn in your slides to the following folder:
        - https://drive.google.com/drive/folders/1c7B5mJPCnHFTZG6la6-dWgYaHRwOgiVD?usp=sharing
        - There are folders for Tuesday, Thursday, and Video lectures
        - **Name your project XX-lastnames, where XX is the order in which you will present in lecture (see link directly below) and lastnames are your last names, hyphenated.**
            - For examle, the first presentation on Tuesday would be named "01-Mason". **Please use the two-digit numeral prefix.** (ie, use "01" instead of just "1").
- Presentation ordering and dates/times are on the following Google doc:
    - https://docs.google.com/spreadsheets/d/1pw9skJhgkQh9C-FExkOoIIrfxxtIZS-PqAM0InIZ6EY/edit?usp=sharing
- Presentations should roughly cover:
    - What is the problem you are trying to solve
    - How are you trying to solve it? Discuss architecture of your system and how it fits into datacenter-scale computing.
    - Any results, demos, GUI, or findings you'd like to show.
        - Remember, you must use Google slides. It is okay to embed a small video on the slides, but assume you'll be using my laptop. Apologies we don't have more time to let everyone show off their systems with really cool demos.
- Grading will be done by me and the TAs. We'll grade on:
    - Clarity: did we understand your problem and your solution.
    - Handling of Q&A: did you do a good job answering questions?
    - Time: did we have to cut off your presentation? If so, the group may be docked.
    - Due to the short time scale, **not all group members need to present**. We will give the same grade to every member in the group.
    - Final presentations count towards 10% of the overall Final Project (which is 40% of course grade). You can check moodle for updates.

Final Project Report

Final project reports are due by the end of the day on Thursday, Dec 13. The final project report is the largest part of the final project grade (50% of the final project grade), and students are expected to do thorough job. The report is meant to show all of the hard work that has been done over the last part of the semester, and therefore should be several pages long (at least 6 pages, but probably a bit more). Please see the details below:

- Name your final “final-project.pdf”. The format should be PDF.
- A rough grading rubric can be found here:
  - https://docs.google.com/document/d/1nLoGkPX5m2DC6dZvubZyzcpBwcV9zODpEkWQsdOTMp0/edit?usp=sharing
- Each of the following **sections must be included** (see grading rubric for more details and post to Piazza if any questions):
  - Introduction
  - Related Work
  - System Design
  - Evaluation/Findings
  - Review of Team Member Work
  - Conclusion
- The “Introduction” section should define the problem and motivate the problem. It should discuss why the problem needs to be solved via a datacenter-scale solution.
- See rubric document for “Related Work” section notes
- For the “System Design” section, remember to clearly outline the system. **Add a diagram of how the components of the system work together.** Describe *in detail* all of the components, and discuss any false starts or other issues that may have arose throughout the project.
  - This section should also include information about the datasets used, and any problems using the datasets, or amount of sanitization needed to get the datasets in working order.
- See rubric for “Evaluation/Findings” section notes
  - You can include a video demo of your work and then put a link to the video in the document. Videos cannot supplement text alone, however.
- The “Review of Team Member Work” section should clearly describe what each student did in relation to the final project. Add what components of the system each student completed, and discuss timelines each of the system components were completed in.
- See rubric for “Conclusion” section notes
- See rubric for formatting, writing, and other document notes
- Using text/diagrams/data/etc from previous proposals and checkpoints is fine. I would recommend you still add more content, as most groups didn't have very detailed descriptions in their earlier reports. I'm expecting the final report to be more thorough.
- *Clarifications from Piazza (12/11) below:*

- Remember, for the “Evaluation/Findings” section, you can include a video demo of your work and then put a link to the video in the document. Videos cannot supplement text alone, however. This is especially relevant for projects that do not have any graphs or findings.
- For font size: no larger than 12 size font
- Remember, you can copy/paste liberally from your proposal and checkpoints. But the final project document needs to be self-contained. In other words, you should assume we can just give this document to a TA (that hasn't read any of the previous proposals/checkpoints), and the TA will understand the project and its contributions.