CS341: Project in Mining Massive Datasets

Michele Catasta, Jure Leskovec, Jeffrey Ullman

Agenda

- Intro by Michele
- Logistics & Class Overview
- Intro to Google Cloud

Projects in Spring 2019

- Discovering Driver Signatures in Automotive Data (x 2)
- Subgraph Pattern Matching on Graphs with Deep Representations
- RecSys Challenge 2019
- Recommender System for Publisher of Technical News
- Diagnosing TMJ Arthritis
- Anomaly Detection of Computer Health
- Wildlife detection
- Longitudinal analysis of the Web Graph

Class Logistics

- Please register on Piazza if you haven't: https://piazza.com/stanford/spring2019/cs341
- Onboard: 4/3, we will meet every Wed on April, then on a per-need basis
- Checkpoint presentations: Checkpoint 1 on 4/24, Checkpoint 2 on 5/15
- Checkpoint reports: Checkpoint 1 on 4/28, Checkpoint 2 on 5/19
    - 1. What problem are you working on?
    - 2. What data are you using?
    - 3. What methods for solution have you tried?
    - 4. What are your results so far?
    - 5. What are your plans to complete the project?
- Final Presentation: 6/5 (Wed), more info to be given by then
- Final Report: 6/9

Expectations / Advice

- Self-motivation, how much you learn from the course totally depends on you
- Good to set up a regular meeting with mentors every week to keep track of progress
- Don’t wait for mentors to tell you what to do
- Please use Office Hours as much as possible! See scheduling information at http://cs341.stanford.edu/
    - Possible issues: build bugs, cloud setup, interpersonal issues, need ideas, etc.
- Use Piazza as a StackOverflow for TAs/mentors

Advice on conducting research

- **Make sure you put in the time required (or more :-), work hard, consistently, independently, but also as a team player!**
- **Don’t be afraid to be innovative and creative in your thoughts**
  - Don’t be afraid to modify/shift the project direction

Advice on conducting research

- **Do supplemental reading**
- **Don’t be afraid to make a mistake or take a risk**
  - Some of the best innovations occur from people taking risks, making errors, and learning from them
- **Take your work seriously!**

How to prepare for a meeting

How to prepare for a research meeting:

- **Update on your progress** (max 10 minutes)
  - Prepare a printout or slides with your past progress
    - **Send these out before your meeting**
  - Cover the essential results and findings. Be precise!
  - Results of failed experiments are especially useful
  - Don’t try to cover every little thing you did, just focus on important results

How to prepare for a meeting

How to prepare for a research meeting:

- **Prepare questions/ideas for further directions**
  - Bring a written list of questions or issues to each meeting
    - Mentors cannot fully answer questions that are not asked!
  - Think about what you plan to do next
- **Take notes!**
  - Keep precise research progress and meeting notes

Grading

- **The grade for the course is composed of the following parts**
  - Checkpoint 1 presentation: **10%**
  - Checkpoint 2 presentation: **20%**
  - Final project presentation: **20%**
  - Final project writeup: **50%**

Google Cloud Platform

CS341

- Founded a company in 2014 (Denizen)
- Product Manager for a distributed systems company (Mesosphere)
- Research Fellow for Microsoft Research. Research topics: deep reinforcement Learning, curriculum learning, HCI
- Experienced with production deployment of distributed systems, e.g. Docker, Kubernetes, Mesos, Spark, Cassandra, Kafka, Akka etc.
- Come to me for help setting up data pipelines and infrastructure!

Abhay Agarwal (MS Design ‘19)

Agenda

- Account/Billing/Alerts
- Launching VMs
- Clusters
- Containers
- Tips

What is Google Cloud Platform?

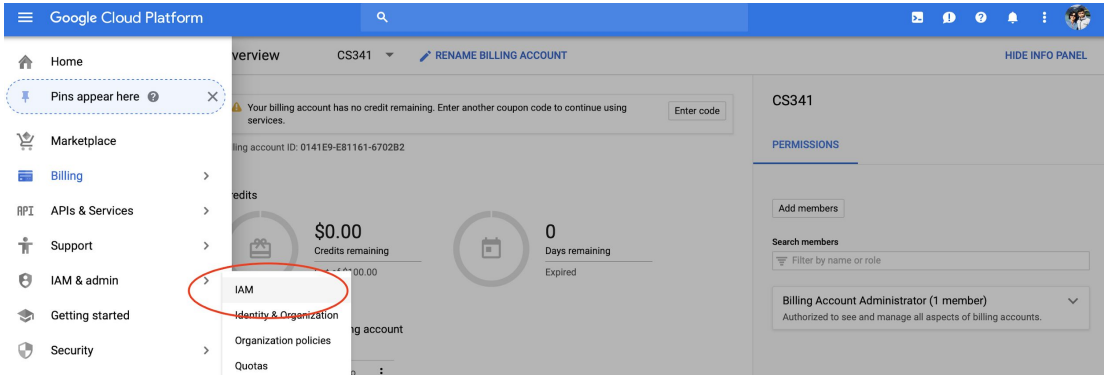
Google's cloud computing service (using same infrastructure used by Google for products like search). Relevant for this class:

Compute Engine	Virtual Machines
Storage Services	Relational and NoSQL cloud storage
Data Services	Hadoop/Spark clusters, cloud ML service, APIs for natural language, vision, speech

Full list of products: https://cloud.google.com/products/

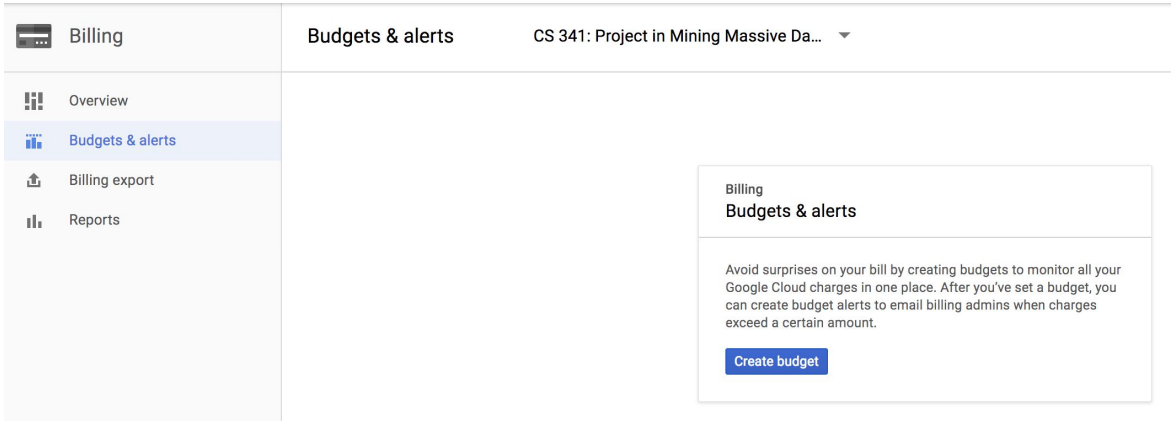
Setup: Create a project

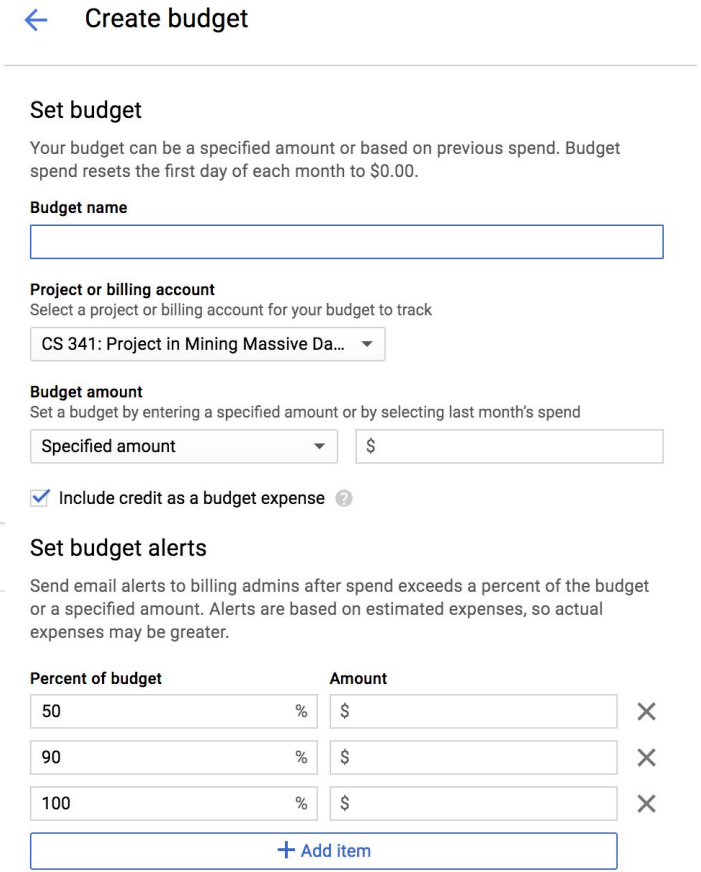
1. Visit https://console.cloud.google.com
2. Click on “Create a Project” and complete the flow. Billing should be set up automatically to use the EDU credits
3. Go to “IAM” from main menu, add rest of team members (using Google accounts, **NOT** stanford.edu account)
4. Go to Piazza for info about adding your Google Cloud credits (1 per team!)



Setup: Create Billing Alerts

1. **Very important**! You do not want to accidentally spend all of your money.
2. Go to Billing and select your project.
3. Set up many alerts based on monthly spend, percentage spend, etc.





Interacting with Google Cloud Platform

Broadly you can interact with GCP in three ways:

1. Graphical UI (https://console.cloud.google.com/): Useful to create VMs, set up clusters, provision resources, manage teams etc

2. Command line (gcloud sdk tools): Useful for using the resources once provisioned. E.g. ssh into instances, submit jobs, copy files etc

3. Cloud Shell (**recommended**): Same as command line, but web-based and pre-installed with SDK and tools, and a persistent home directory (!). More info here: https://cloud.google.com/shell/docs/quickstart

Setup: Command line tools

1. Make sure you have Python 2.7.9 or higher
2. Download SDK: https://cloud.google.com/sdk/docs/
3. Install: run `./install.sh` and follow the installation steps
4. Authorize using your credentials: Run `./bin/gcloud init`
5. Test: `gcloud components list, gcloud auth list`

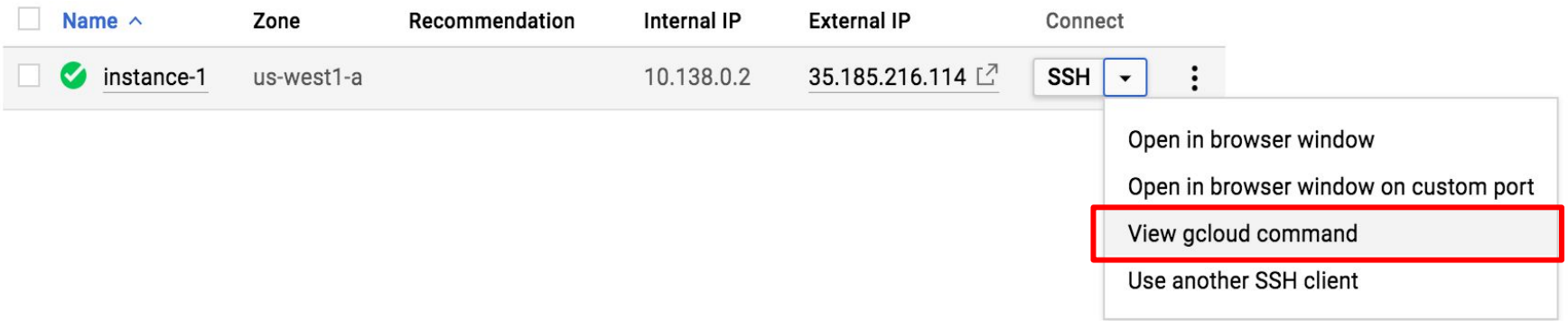
Setup: Command line tools

```
nihit@nihit-lp1:~/Documents$ gcloud components list

Your current Cloud SDK version is: 149.0.0
The latest available version is: 149.0.0

                                                Components
┌──────────────┬──────────────────────────────────────────────────────┬──────────────────────────┬───────────┐
│    Status    │                         Name                         │            ID            │    Size   │
├──────────────┼──────────────────────────────────────────────────────┼──────────────────────────┼───────────┤
│ Not Installed │ App Engine Go Extensions                             │ app-engine-go            │  47.7 MiB │
│ Not Installed │ Cloud Datastore Emulator                             │ cloud-datastore-emulator │  15.4 MiB │
│ Not Installed │ Cloud Datastore Emulator (Legacy)                    │ gcd-emulator             │  38.1 MiB │
│ Not Installed │ Cloud Pub/Sub Emulator                               │ pubsub-emulator          │  21.0 MiB │
│ Not Installed │ Emulator Reverse Proxy                               │ emulator-reverse-proxy   │  56.8 MiB │
│ Not Installed │ Google Container Registry's Docker credential helper │ docker-credential-gcr    │   3.4 MiB │
│ Not Installed │ gcloud app Java Extensions                           │ app-engine-java          │ 128.6 MiB │
│ Not Installed │ gcloud app PHP Extensions (Mac OS X)                 │ app-engine-php-darwin    │  21.9 MiB │
│ Not Installed │ gcloud app Python Extensions                         │ app-engine-python        │   6.1 MiB │
│ Not Installed │ kubectl                                              │ kubectl                  │  11.4 MiB │
│ Installed     │ BigQuery Command Line Tool                           │ bq                       │   < 1 MiB │
│ Installed     │ Bigtable Command Line Tool                           │ cbt                      │   3.9 MiB │
│ Installed     │ Cloud Datalab Command Line Tool                      │ datalab                  │   < 1 MiB │
│ Installed     │ Cloud SDK Core Libraries                             │ core                     │   5.8 MiB │
│ Installed     │ Cloud Storage Command Line Tool                      │ gsutil                   │   2.8 MiB │
│ Installed     │ Default set of gcloud commands                       │ gcloud                   │           │
│ Installed     │ gcloud Alpha Commands                                │ alpha                    │   < 1 MiB │
│ Installed     │ gcloud Beta Commands                                 │ beta                     │   < 1 MiB │
└──────────────┴──────────────────────────────────────────────────────┴──────────────────────────┴───────────┘
```

Configure and use a VM

1. Visit https://console.cloud.google.com/compute/instances.
2. Click on the “Create Instance” button.
3. Configure instance name, zone, machine type, network traffic, etc.
4. Congrats, your VM has been created! Use “View gcloud command” and copy the message in the pop-up dialog to your bash shell.

   (something like: gcloud compute --project "yourProjectID" ssh --zone "yourInstanceZone" "yourInstanceName"*)*



Configure and use a VM (Cont'd)

5. Stop your machine when not in use to avoid unexpected charges.

6. For more details, see https://cloud.google.com/compute/docs/quickstart-linux.

FAQ: My bash shell is complaining gcloud command not found. :(Reload your bash_profile using the "source" command, OR simply restart your bash shell.

Attach a Disk to Your VM

1. **Create your blank disk.**
   (1) VM instances -> click on your instance -> “Edit” button at the top -> additional disks -> “Add item” button.
   (2) Select “Name” dropdown -> Create disk -> Source type: select “blank disk” -> configure whatever nickname and size to your disk.
2. **Format and mount your disk**
3. **Every time you reboot, you need to mount your disk again**:
   `sudo mount -o discard,defaults /dev/[DEVICE_ID] /mnt/disks/[MNT_DIR]`
4. For more details, see
   https://cloud.google.com/compute/docs/disks/add-persistent-disk

Create a Cluster

1. **Two ways to create a cluster**:
   Use command line (easier): `gcloud dataproc clusters create <cluster-name>`
   OR Use GUI: visit https://console.cloud.google.com/dataproc/clusters.
2. **View your clusters:** https://console.cloud.google.com/dataproc/clusters.

Clusters:

Name	Zone	Total worker nodes	Cloud Storage staging bucket	Created	Status
cluster-1	us-central1-a	2	dataproc-a60e0265-f815-44e1-83e2-8b7284431f9e-us	Apr 4, 2017, 11:34:03 PM	Running

Instances: 1 master node and 2 worker nodes have been created

Name	Zone	Recommendation	Internal IP	External IP	Connect
cluster-1-m	us-central1-a		10.128.0.2	104.198.52.60	SSH
cluster-1-w-0	us-central1-a		10.128.0.4	35.184.84.218	SSH
cluster-1-w-1	us-central1-a		10.128.0.3	104.154.182.220	SSH

Submit a Job

1. **Create your job.**

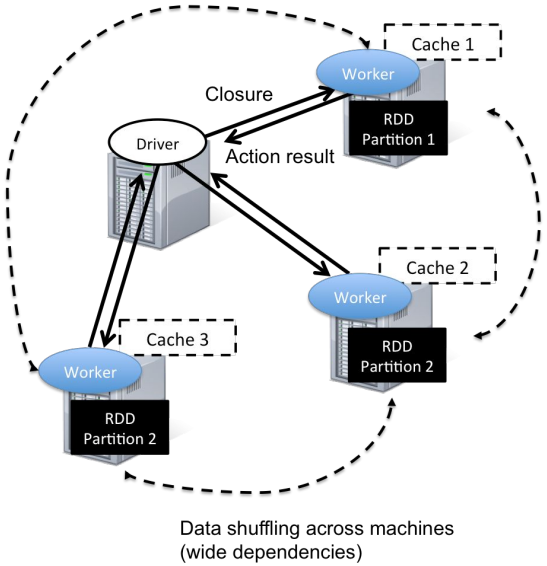
   Simple example: add one to every element in an array.

```
import pyspark
sc = pyspark.SparkContext()
original_array_rdd = sc.parallelize([3,2,5,1,4])
new_array_rdd = original_array_rdd.map(lambda x: x+1)
new_array = sorted(new_array_rdd.collect())
print new_array
```

2. **Submit your job:**

```
gcloud dataproc jobs submit pyspark --cluster
<my-dataproc-cluster> my-first-job.py
```

3. **View your jobs:** https://console.cloud.google.com/dataproc/jobs.



Data shuffling across machines
(wide dependencies)

Storage Solutions for Clusters

1. You can choose to use
   (1) cloud storage
   (2) share a persistent disk among your cluster
   (3) Other solutions depending on your needs
   This page offers detailed explanation
   https://cloud.google.com/solutions/filers-on-compute-engine#cloud-storage.
2. To set up **cloud storage**, see tutorial on
   https://cloud.google.com/compute/docs/disks/gcs-buckets.
3. To **share a persistent disk** among all machines in your cluster, see tutorial on
   https://cloud.google.com/compute/docs/disks/add-persistent-disk#use_multi_instances.

Google Kubernetes Engine (GKE)

1. Containers are lightweight, isolated VM-like objects for running code in a consistent, repeatable environment (e.g. packaging your code with needed libraries)
2. Visit https://cloud.google.com/kubernetes-engine/
3. Create a cluster
4. Launch a distributed application
5. Congrats, you are running a distributed system with isolation, scalability, repeatability.

Create a Cluster & Deploy your app

1. Use command line: `gcloud container clusters create [CLUSTER_NAME]`
2. Deploy an application: `kubectl run hello-server --image [my-app]`
3. Your application can run code, expose a web UI, scrape from the web, add data to a table, etc. If your process dies, it is restarted automatically.
4. Find more info in the quickstart guide: https://cloud.google.com/kubernetes-engine/docs/quickstart

Other services that might be useful

- Natural Language: https://cloud.google.com/natural-language/
- BigQuery: https://cloud.google.com/bigquery
- DataPrep: https://cloud.google.com/dataprep/
- DataProc: https://cloud.google.com/dataproc/
- Cloud ML Engine: https://cloud.google.com/ml-engine/

Suggested Developer Patterns

- Create a Continuous Integration Pipeline: create a git repo with your code, add a build manifest that compiles/packages/tests your code, add a dockerfile that runs the build tool, and create a build trigger to auto-build a container for every code push. https://docs.docker.com/docker-hub/builds/
- Delete your dataproc clusters automatically after your jobs complete! Saves tons of money. Create a bash script for your job: https://cloud.google.com/dataproc/docs/guides/manage-cluster#delete_a_cluster
- Create versioned models and host them in your data store!