<https://github.com/cbdscolin/CS5700-P4-CDN.git>

<https://github.com/medhavim/CS-5700---Network-Fundamentals>

# On the CDN Project Nodes

Hi All,

Following up on [@233](233), some ground rules on using the cdn testing nodes:

* These machines are shared between all the groups. Don’t pin them at super high load. If you do that several times, I will disable your access.
  + If you are using a compiled language for this project, do not run compilation jobs on these nodes. Use the dedicated build node at cs5700cdnproject.khoury.northeastern.edu using your **khoury credentials**.
  + Running interpreters on the nodes is permitted.
* You may not install software to these nodes. We've provided scamper and Python’s dnslib. Ask in a private thread if you have another requirement you would like to request.
* Treat these machines as ephemeral. You need to consider both the runtime and storage elements of this statement.
  + While your jobs can run for a long time they cannot run continuously. To keep the nodes running generally well, I have configured them to reboot nightly. Additionally, they may have to go down unannounced for breif periods to, eg, tame a run-away job. Don’t setup jobs to automatically restart your infrastructure.
    - In the event of something catastrophic, I may have to completely replace any given node. ***Keep backups of all code/datasets/artifacts you deploy.*** We consider you all responsible for your data.
* We need some way for each group to provide their infrastructure. We do so using port numbers. Your code must provide DNS on this port from the DNS node and HTTP on the same port from the HTTP node (later nodes). Bindings are below.

# Project 4 posted

Almost everything is set up, and while you wait you can read about the project here: <https://david.choffnes.com/classes/cs5700f22/project4.php>

The replica and DNS servers (where you will run your HTTP server and DNS server, respectively) are at:

* proj4-dns.5700.network (97.107.140.73)
* proj4-repl1.5700.network (139.144.30.25)
* proj4-repl2.5700.network (173.255.210.124)
* proj4-repl3.5700.network (139.144.69.56)
* proj4-repl4.5700.network(185.3.95.25)
* proj4-repl5.5700.network(139.162.83.107)
* proj4-repl6.5700.network(192.46.211.228)
* proj4-repl7.5700.network(170.187.240.5)

Update:

Here is the popularity of each page, according to how many views it has: [pageviews.csv](file:////redirect/s3%3fbucket=uploads&prefix=paste%252Fho06biocf8jp0%252Feb3ed044f8c9a064760175e30aee3f0afcc1538ea0bc89cfd55bae71f487c06a%252Fpageviews.csv). This will be used for the request distribution.

# CDN Beacon Page

Hello again all!

The Beacon site is now available at <https://cdn-beacon.5700.network/> with one node pinging your CDN Ports as listed in [@234](234). This beacon performs, in order: a dns query, the health check, and a single GET request. You need to have implemented the /grading/beacon endpoint as described in the assignment before the beacon will test a GET request. More nodes will come after the Milestone to begin giving you insight into how your code is mapping clients to replicas.

Tariq S.

# Why not just pre-cache the pages according to the distribution?

The project description states: I will test your code using simple clients that 1) ask your DNS server for the IP address of *cs5700cdn.example.com*and 2) uses an HTTP get to fetch a file from that address. These clients will run from servers located all over the world. The request frequency for each piece of content will follow a [Zipf distribution](http://en.wikipedia.org/wiki/Zipf's_law" \t "_blank), and the size of content at the origin server is much larger than the size of your cache on each replica server.

Following this, why would we need to implement a “live” caching strategy which adapts to the traffic the replica is seeing; I.e. isn’t the optimal caching strategy simply to just to cache the first 20/40MB of content by frequency in the distribution?

The only reason I could think of why this would not be optimal would be if, for a given test run, only a subset of pages from the distribution are tested.

project4

Editgood question

# Imp Piazza List:

<https://piazza.com/class/l781ljflhl536s/post/302>

<https://piazza.com/class/l781ljflhl536s/post/294>

<https://piazza.com/class/l781ljflhl536s/post/287>

# Scamper Usage example:

<https://manpages.ubuntu.com/manpages/trusty/man1/scamper.1.html>