Command and Control Subsystems Report

Jake Vossen: OREPACKAGERS

April 1, 2019

1 Subsystem Description

The Command and Control subsystem is the subsystem responsible for converting the requests that have been collected into downloaded data to be distributed to users. It starts by receiving a list of `request` objects - a structure for contating information about each request. To prevent confusion, the mono-spaced `request` will refer to they Python object itself, wheras plain "request" refers to the concept of a user request.

With this list of requests, the first thing it does is use Pythons `multiprocessing`[1] library to split work up between the different threads on the computer. While this software is designed for low end machines to be more accessable to developing areas, most computers[2] in recent times will have more than 1 CPU core (including the Raspberry Pi[3]). This allows for the processor to split up all the requests, and execute them in paralel, instead of waiting for each one to finish individually, which can provide a large preformance boost.

When downloading a `request`, it determens the type of request. The types are URL, search, youtube, and ipfs. The steps for each type of request is outlined below.

1.1 URLs

URLs are your basic websites, such as `https://en.wikipedia.org/wiki/Monty_Python_and_the_Holy_Grail`, or `https://www.nytimes.com/2019/03/27/technology/turing-award-ai.html`. This is for users who already know the content they want. In the backend, the Python program is going to use the `wget`[4] utility. Specifically, `wget -E -H -k -K -p -P path url robots=off` where `path` is the output directory and `url` is the url that has been requested. To break it down:

- `-E` tells `wget` to change the file extention if the url isn't a .html file. This allows for the downloading of PDF files as well as HTML files
- `-H` Tells `wget` that it is okay to download material from hosts that aren't from the specified URL. While this seems backwards at first, many websites host their fonts or pictures in a place that isn't the same as the document that is being request. This allows the page to appear just as it would when visited in a web browser
- `-k` This stands for "convert links", which means that when the download is complete, it converts the links on the page so they are sudible for browsing on the local machine. For example, if a blog has `otherwebsite.com/picture` on it, it will replace that with just `picture` to ensure that the browser will use the local versions of that picture
- `-K` This means that `wget` will make a backup of the HTML file when converting links with the `-k` option.
- `-p` is the most important option, as it tells `wget` to download all the requirements as well as the url. So if the site links to an outside source (such as `otherwebsite.com/picture`) also gets downloaded if it is linked in the requested url.

All of those options ensures that downloading the URL requested get's the website exactly as it appears in a browser, including linked images. Additionally, it works with PDF and ZIP files, which is really important to ensure all possible media can be obtained. This method is also used by other parts of the program.

1.2 Search

Sometimes the user will not know exactly what they want, so we added an option to get the first page of Google results (top 10 results). The `googlesearch`[5] library was very helpful for this. This library provides a list of URLs, and then we use the URL method to download those results (or the youtube download option if it is a youtube link). The results are each in thier own folder nammed based on the google search rank (1 is first result, 2 is second result, etc).

1.3 YouTube

It is well known that lot of quality educational and entiertainment content is in video format, and the majority of that content is on YouTube. That is why we are adding functionallity to request YouTube videos (through a link, or a result from the search function). In this case, the `youtube-dl` program allows for content retrevial.

1.4 IPFS

IPFS stands for "InterPlanetary File System", which is a "A peer-to-peer hypermedia protocol to make the web faster, safer, and more open"[7]. The internet that is familiar to most people is the client-server model[8], but IPFS changes that so everyone is both a client and a server. Media is distribtued based of their crytographic hash, a unique ID for each object instead of a URL. Anybody can add objects, and when requesting an object, it can be downloaded from any number of servers, not just the origional person hosting the server. The `ipfs` command line utility[9] is used to retreive objects.