CS280 Proposal for Independent Study

Shawn Tice

February 9, 2012

I propose to modify the Yioop search engine to be capable of performing a distributed crawl of a large web archive file on the local network. Web archive files are used to insert new data into the engine without having to actually send requests to production servers. For example, Wikipedia provides a single archive of all of their pages, as well as smaller archives for all of the articles written in a particular language; rather than sending millions of HTTP requests to Wikipedia's live servers while performing an online crawl, the Yioop administrator can download the archive and instruct Yioop to crawl it in a manner very similar to the way that it would an actual website. This procedure avoids the overhead of millions of individual HTTP requests, and allows most of the work to be done offline.

The trade-off for the benefits gained by crawling an archive is that we've moved the work from a remote server to our own network. Now all of the data is on a single server in our own network, and if we want to take advantage of the extra machines that we have available for online fetching, then we need to split up the archive and send it to our fetchers. This problem is one of the motivations behind distributed file systems like Apache Hadoop, Amazon S3, CloudStore, and several others. Adopting a naive approach to this task could result in spending more time transferring the split-up archive (and the results of indexing) over the local network than we gain by performing the indexing in parallel. The alternative is simply to do all of the indexing on the machine where the archive resides.

My plan for the semester is to first research distributed file systems and how they're used in the architecture of large search engines, then to apply that research to modifying Yioop to be able to perform distributed indexing of a single large archive, and finally to run experiments to tune the parameters of the process. I will read research papers on several large distributed file systems, implement the changes to Yioop in PHP, and write a report detailing my experimental findings. In order, my deliverables will be:

1. A short report outlining Yioop's current architecture (where relevant to offline indexing) and my proposed changes.
2. A patch to Yioop that implements my proposed changes.
3. A report on my experimental results, which provides recommendations for tuning the system parameters.

4. A final patch to Yioop that implements my recommendations.

I will be meeting with Dr. Pollett each Tuesday at 3 P.M. to discuss my progress and findings, as well as any problems that I run into.