David Muir Sharnoff

resume addendum: Searchme details

http://dave.sharnoff.org/resume

Details of my 13 months at Searchme

Docstore Aggregation: Dump the docstore, bucketize, sort, aggregate into mysql according to a configuration file. Ran on 150 nodes, dumping data for 4B+ documents each run and producing data for 50M+ domain-path pairs. Perl. Roles: sole designer, sole implemntor, implemented configuration suggestions from my team.

Aggregation table tools: dump, url-merge, combine, sort, etc. We needed to use the aggregation data to build indexes and mysql could not easily handle that so the data was dumped into TSVs. Perl. Roles: determined need, implmented, took design suggestions from my team.

Log processing: Built a cascading map-reduce framework to process SearchMe's web logs. My goal was to calculate ROI, judge search result quality, feed into the advertising billing system, and provide platform for research into our web logs. I designed and built a multi-step MapReduce-like framework for SearchMe. This included dependencies that changed durations (eg: combining days into weeks); code generation for cross-product and nested aggregations of data with both prebuilt and custom reducers (see next section); and all of the plumbing to manage the starting of tasks on a network of systems. I could have used Hadoop to provide the MapReduce framework but I chose not to do so because using Hadoop would have pushed us into a Java-centric world and that would have slowed me down for the 3/4ths of the log processing task that wasn't starting tasks in parallel. My framework worked and I generated many of the numbers we needed. Perl, mysql. Roles: determined features, sole designer, sole implementer, configuration writer.

Streaming Aggregator: that can do nested aggregations and cross product aggregations. Nested: median number of clicks per url, per path component, per host and per domain. Cross product: count of clicks for each combination of ad in-link, country of IP address, interface version. Used for log processing, but not Docstore Aggregation since it was not written in time. Also used by other team members for various projects. Perl. Roles: determined need, sole designer, sole implementer.

Alcatraz: Another project at SearchMe involved overrides for our page categorization system. We had a system called Alcatraz that stored overrides in a database and applied the overrides to pages in our document store. Alcatraz ran out of steam as the number overrides increased. I proposed and then implemented a solution: redesign its database to do less and change the search index build pipeline to include a step to apply overrides from Alcatraz. These changes required coordination with several other teams and ultimately solved the performance issues completely. Perl, mysql. Roles: implmentation team member for the first version, design and implementation team leader for the rewrite.

Special Features Processing: Load web pages into XML using libXML and then apply rules (either in perl or Xpath) to extract features to be used by various machine learning systems. It turns out that libXML will sometimes hang and sometimes coredump so making this a reliable process required running the parsing in a subprocess and limiting retries. Rules for this system were written by several different researchers. Perl, XML. Roles: only peripheral involvement for the initial version; sole designer, sole implementer for the rewrites; team member for writing rules.

Media SFP: Apply the Special Features Processing to extract media-specific features for relevance and generating the tags needed for the front-end. Perl. Roles: sole designer of the structure. I did the extractions of youtube, flickr, and metacafe. With my guidance, researcher did the extraction for ehow.

data/research: A collection of binaries and data files useful for all the researchers so that only one of us needed to collect such things. Roles: determined need, solved problem.

Paid Inclusion: Processing scripts to push customer-specified URLs through all processing steps and into our index. Perl, mysql. Roles: team member.

Unified Command Interface: A standard command line interface for all research group connected tools so that they are easy to run and their behavior is flexible. Perl. Roles: determined need, sole designer, sole implementer.

SearchMe.pm: A perl module to fit us into the Searchme deployment situation: our finished (internal) products needed to be deployed as stand-alone directories with no additional items (eg: CPAN modules) installed. My solution allowed easy development (leveraging /data/research) and then easy installation though a canned install script. Perl. Role: determined need, sole designer, sole implementer.

Machine Learning Feature Experiments: Experiments to find features to help our machine-learned relevance. I tried: aggregation by domain/path (see above, successful); aggregation by features of in-linking URLs (no gain); aggregation by features of outlink destination ULRs (no gain); aggregation of features derived from user behavior in our search logs (no gain). Perl. Role: primary designer, sole implementer.

Torgo scapers: parse the search results from google, ask, yahoo, msn, bing, cuil for DCG comparisons. Role: inherited broken code; sole re-designer, sole re-implementer.

do.hosts: A command for running commands on a cluster. Perl. Roles: determined need, sole designer, sole implementer.

tsv: a simple tool for manipulating TSV files (cut by name; grep by column, rotate, etc). Perl. Roles: determined need, sole designer, sole implementer.