

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 implementor, 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 an alternative to Hadoop. Log processing was configured as a series of steps. Each step parsed its input, filtered the records, grouped the records, transformed the records into new records, bucketized the output to multiple hosts, and sorted the output. The framework understood time spans and the dependency relationship between the steps and ran the steps in parallel across a cluster. I built the configurations for this system to generate the daily search statistics and to answer specific questions from the executive team, eg: what is the click-through-rate per result position and type? Perl, mysql. Roles: sole designer, sole implementer, sole 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: Domain/path scanning to apply categorization corrections and then later a rewrite to store the completed corrections in the Docstore instead of mysql. 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. 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 ehov.

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 URLs (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.