ApacheConUs2009MeetUp

We had a "Web Crawler Developer" MeetUp at this year's ApacheCon US in Oakland.

It wound up being an UnMeetUp (MeetDown?) on Wednesday, November 4th from 11am - 1pm.

Attendees

- Andrzej Bialecki - Apache Nutch
- Thorsten Sherler - Apache Droids
- Michael Stack - Formerly with Heritrix, now HBase
- Ken Krugler - Bixo

Topics

Roadmaps

- Nutch - become more component based.
- Droids - get more people involved.

Sharable Components

- robots.txt parsing
- URL normalization
- URL filtering
- Page cleansing
    - General purpose
    - Specialized
- Sub-page parsing (portlets)
- AJAX-ish page interactions
- Document parsing (via Tika)
- HttpClient (configuration)
- Text similarity
- Mime/charset/language detection

Tika

- Needs help to become really usable
- Would benefit from large test corpus
- Could do comparison with Nutch parser
- Needs option for direct DOM querying (screen scraping tasks)
- Handles mime & charset detection now (some issues)
- Could be extended to include language detection (wrap other impl)

URL Normalization

- Includes both domain (www.x.com == x.com), path, and query portions of URL
- Often site-specific rules
    - Option to derive rules using URLs to similar documents.

AJAX-ish Page Interaction

- Not applicable for broad/general crawling
- Can be very important for specific web sites
- Use Selenium or headless Mozilla

Component API Issues

- Want to avoid using an API that's tied too closely to any implementation.
- One option is to have simple (e.g. URL param) API that takes meta-data.
    - Similar to Tika passing in of meta-data.

Hosting Options

- As part of Nutch - but easy to get lost in Nutch codebase, and can be associated too closely with Nutch.
- As part of Droids - but Droids is both a framework (queue-based) and set of components.
- New sub-project under Lucene TLP - but overhead to set up/maintain, and then confusion between it and Droids.
- Google code - seems like a good short-term solution, to judge level of interest and help shake out issues.

Next Steps

- Get input from Gordon re Heritrix. Stack to follow up with him. Ideally he'd add his comments to this page.
- Get input from Thorsten on Google code option. If OK as starting point, then Andrzej to set up.
- Make decision about build system (and then move on to code formatting debate 😃
    - I'm going to propose ant + maven ant tasks for dependency management. I'm using this with Bixo, and so far it's been pretty good.
- Start contributing code
    - Ken will put in robots.txt parser.

Original Discussion Topic List

Below are some potential topics for discussion - feel free to add/comment.

- Potential synergies between crawler projects - e.g. sharing robots.txt processing code.
- How to avoid end-user abuse - webmasters sometimes block crawlers because users configure it to be impolite.
- Politeness vs. efficiency - various options for how to be considered polite, while still crawling quickly.
- robots.txt processing - current problems with existing implementations
- Avoiding crawler traps - link farms, honeypots, etc.
- Parsing content - home grown, Neko/TagSoup, Tika, screen scraping
- Search infrastructure - options for serving up crawl results (Nutch, Solr, Katta, others?)
- Testing challenges - is it possible to unit test a crawler?
- Fuzzy classification - mime-type, charset, language.
- The future of Nutch, Droids, Heritrix, Bixo, etc.
- Optimizing for types of crawling - intranet, focused, whole web.