Nutch:
Open-Source
Web Search Software

Doug Cutting
<doug@nutch.org>

November 26th 2004
University of Pisa

Lucene is...

- A mature Apache open-source project;
- Java library for text indexing and search;
  - Not an application;
- A large community of contributors;
- The search technology behind a lot of web sites & applications (ZOË, JIRA, Lookout, Furl, etc.)
- http://jakarta.apache.org/lucene/
- A Lucene book is out next week!

Nutch is...

- A young open-source project;
- Web search application software;
- Small but growing group of users and developers;
- Behind a few sites.
- Soon to be an Apache project.
- Built on Lucene

Nutch isn't...

- A business;
  - Currently a non-profit legal entity to own copyright
    - Only until it joins Apache
  - No employees.
- A search site;
  - But want to power lots of search sites;
  - From domain-specific, to whole-web.
- A research project.
  - But want to be platform for research.

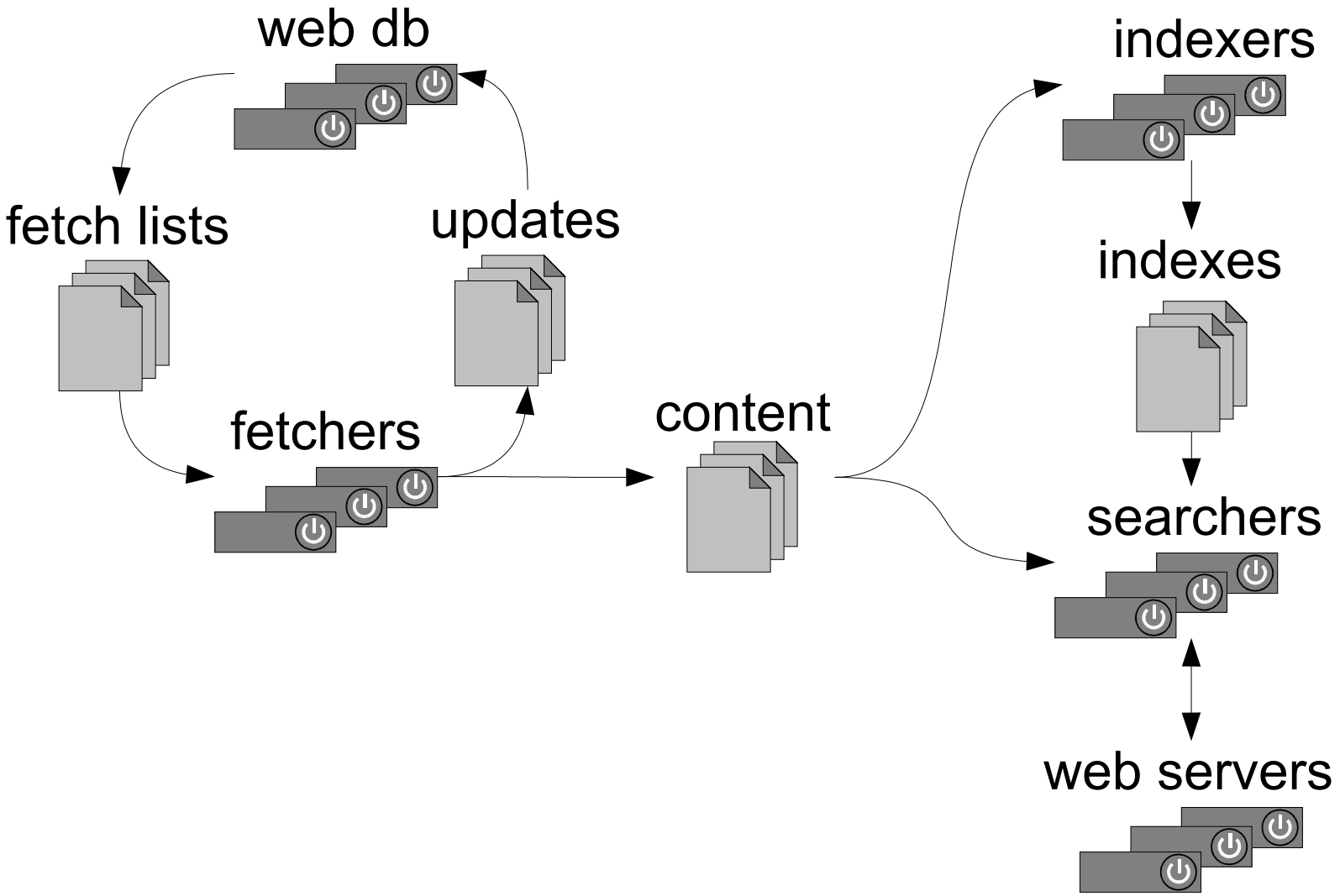
Nutch's Civil Goals

- Increase availability of web-search technology.
  - standard civil goal of open source projects
- Increase transparency of web search.
  - search is essential to internet navigation
  - yet algorithms are secret
  - free, open-source implementation should help.

Nutch Technical Goals

- Scale to entire web
  - pages on millions of different servers
  - billions of pages
  - complete crawl takes weeks
  - very noisy
- Support high traffic
  - thousands of searches per second
- State-of-the-art search quality

Nutch Architecture



Scalability

- To meet scalability goals:
  - multiple simultaneous fetches
    (100+ pages/second / CPU, ~10M / day)
  - parallel, distributed db update
    (100M pages @ 100 pages/second / CPU)
  - distributed search
    (2-20M pages, 1-40 searches/second / CPU)

Web Database

- Page Database
  - Used for fetch scheduling.
- Link Database
  - Represents full link graph.
  - Stores anchor text associated with each link.
  - Used for:
    - Link analysis;
    - Anchor text indexing.

Web Database Implementation

- Not an RDBMS application!
  - 100M pages @ 100 pages/second
  - Requires 1000 link updates/second in 1B entry table
  - B-tree's become fragmented, forcing seek/update
  - Seeks require ~10ms – order of magnitude too slow!
- Instead, sort updates and merge w/ entire DB
  - 100M links/day, 100B/link, ~10 passes = ~100GB xfr
  - @10MB/s:10ks sort + 100GB merge in 10ks = 6hrs

Nutch Documents

field	*stored*	*indexed*	*analyzed*
`url`	Yes	Yes	Yes
`anchor`	No	Yes	Yes
`content`	No	Yes	Yes
`site`	Yes	Yes	No
`lang`	Yes	Yes	No
…			

Nutch Analysis

- Defined w/ JavaCC
- Words are (<letter> | [0-9_&])+
  - or acronyms: <letter> [.] (<letter> [.])+
  - or CJK character
- First word in each anchor gets big position increment, to inhibit cross-anchor matches.
- No stop list or stemmer.
- URLs, email, etc. tokenized same as other text.

Nutch Queries

- By default:
  - require all query terms
  - search url, anchors and content
  - reward for proximity
- E.g., search for “search engine” is expanded to:
  - +(url:search^*x* anchor:search^*y* content:search^*z*)
  - +(url:engine^*x* anchor:engine^*y* content:engine^*z*)
  - url:“search engine”~*p*^*a*
  - anchor:“search engine”~*q*^*b*
  - content:“search engine”~*r*^*c*

Query Parsing

- Certain characters cause implicit phrases:
  - dash, plus, colon, slash, dot, apostrophe and atsign
  - URL & email are thus phrase searches
  - e.g., http://www.nutch.org/ = “http www nutch org”, doug@nutch.org = “doug nutch org”, etc.
- Stop words removed
  - unless in phrase or required.
  - can use N-grams if in phrase
- Plugins can extend for new fields

Nutch N-Grams

- Very common terms are indexed with neighbors.
  - E.g., w/ “the”, “http”, “www”, “http-www” & “org”:
  - “Buffy the Vampire” is indexed as
    buffy, buffy-the+0, the, the-vampire+0, vampire,
  - “http://www.nutch.org/” is indexed as:
    http, http-www+0, http-www-nutch+0,
    www, www-nutch+0, nutch, nutch-org+0, org.
  - terms are field specific
  - improves performance of phrase searches

Nutch N-Gram Query

- For explicit phrase query: “Buffy the Vampire”:
  - for content field, translated to:
    content:“buffy-the the-vampire”
  - much faster b/c we don't have to search for “the”
- For query: http://www.nutch.org/:
  - implicit phrase: “http www nutch org”
  - for URL field, translated to:
    url:“http-www-nutch www-nutch nutch-org”

Intranets & Verticals
Part 1: Scale

- Fetch, DB & search can all run on one box.
- Complete crawl takes only hours.
- Handful of servers on LAN—easy to overload!
- Lessons:
  - may need to throttle fetcher
  - need simple operation—single command
  - can crawl deeper

Intranets & Verticals
Part 2: Control

- cleaner content
- knowledge about structure of sites (cgi's, etc)
- lessons:
  - can index more dynamic content (cgi's, etc.)
  - can better customize crawler to sites

Intranets & Verticals
Part 3: Quality

- only ~1M pages
- lesson:
  - not great for link analysis
  - but plenty for anchor text

Intranets & Verticals
How To Step 1: Install

- Nutch requires only Java & JSP.
- Download & unpack.
- No admin GUI (yet)
  - command line
  - config files

How To Step 2: Configure

- Specify root URLs.
- Specify URL filters.
  - a separate config file, containing regexps
  - each either includes or excludes URLs
  - first matching pattern determines fate of each URL
- Optionally, add a config file specifying:
  - delay between fetches
  - num fetcher threads
  - levels to crawl

URL Filter Example

```
# skip image and other suffixes
-\.(gif|jpg|pdf|doc|sit|rtf|exe)$

# skip URLs w/ certain characters
-[?*!@=]

# accept hosts in nutch.org
+^http://([a-z0-9]*\.)*nutch.org/

# skip everything else
-.
```

How To Step 3: Test Run

- Crawl just a few levels deep

```
bin/nutch crawl urls \
 -dir crawl.test -depth 3 \
 >& crawl.log &
```

- Examine output log for:
  - warnings
    - exclude some file types?
  - sites hit too hard (e.g., infinite sites)
    - exclude some hosts or paths
  - sites not hit?
    - add more root urls, or crawl deeper

How To Step 4: Finish up

- customize the look and feel
  - by default, uses XSLT template
  - or can roll your own.
- perform a full crawl (depth = ~10)
- tell folks about it!

Advantages

- Free!
- Scalability & quality.
- Open source easier to:
  - Customize
    - e.g., ranking, operators, look & feel, plugins
  - Debug
    - You've got the full source!
  - Extend
    - Non-HTTP, non-HTML content, metadata, etc.

Demonstrations

- Intranet: http://search.oregonstate.edu/
- Intranet: http://campusgw.library.cornell.edu/
- Vertical: http://www.playfuls.com/
- Vertical: http://search.creativecommons.org/
- Web: http://www.objectssearch.com/

Preliminary Evaluation at OSU: Nutch versus a Google Appliance

- For OSU's top-25 queries:
  - 9 queries nutch and google were both perfect: 10/10
  - 2 queries nutch was slightly better
  - 2 queries google was slightly better than nutch
  - 1 query google was much better: 10 to 6
  - 1 query google was much better: 10 to 6
  - 1 query both scored 5
  - Google Appliance had a slight overall advantage.

Thanks!

http://www.nutch.org/

doug@nutch.org