common search:

The original vision of Nutch, 14 years later: Building an open source search engine

Apache Big Data Europe 2016

sylvain@sylvainzimmer.com
@sylvinus

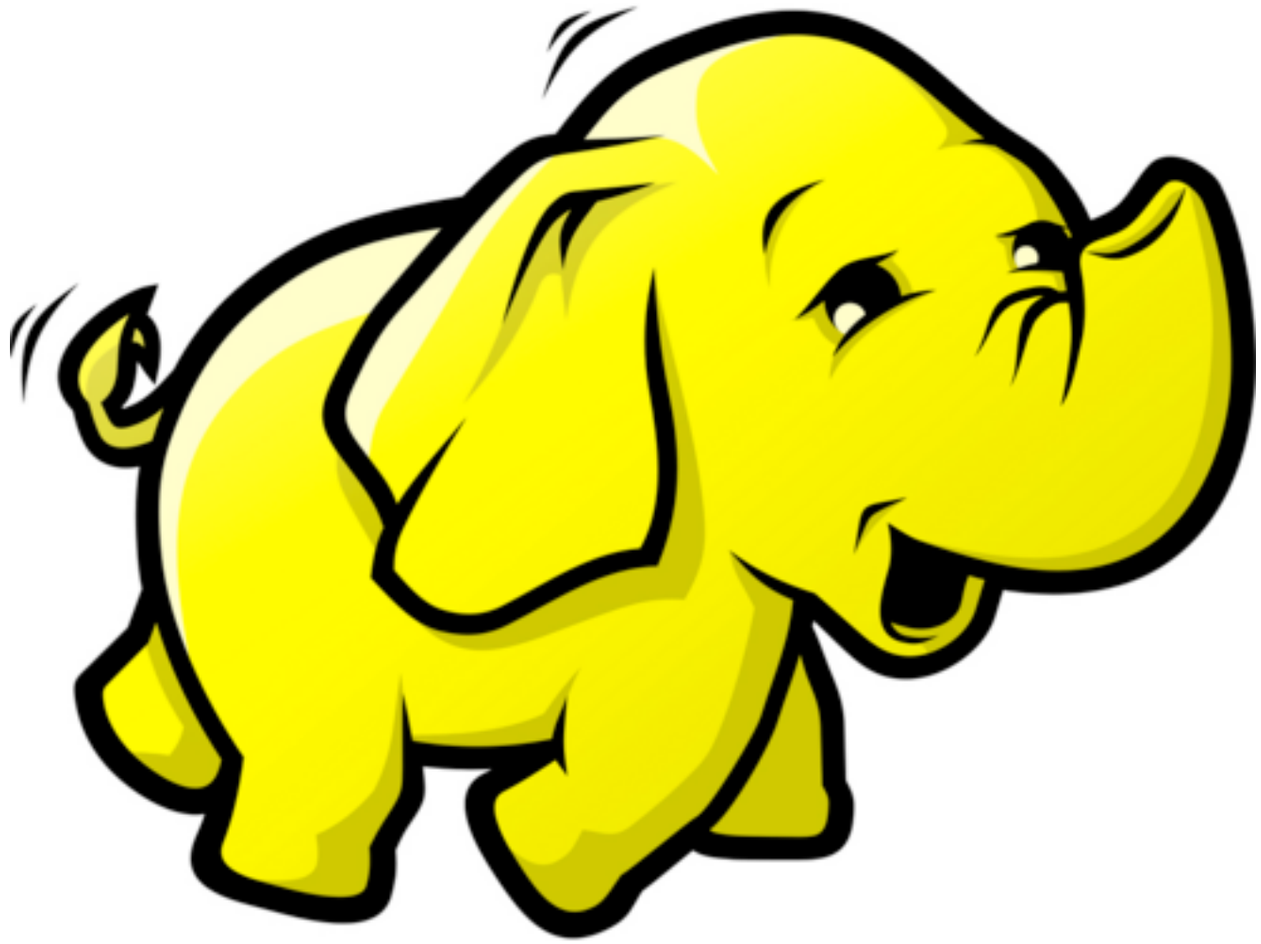
/usr/bin/whoami

- Jamendo (Founder & CTO, 2004-2011)
- TEDxParis (Co-founder, 2009-2012)
- dotConferences (Founder, 2012-)
- Pricing Assistant (Co-founder & CTO, 2012-)

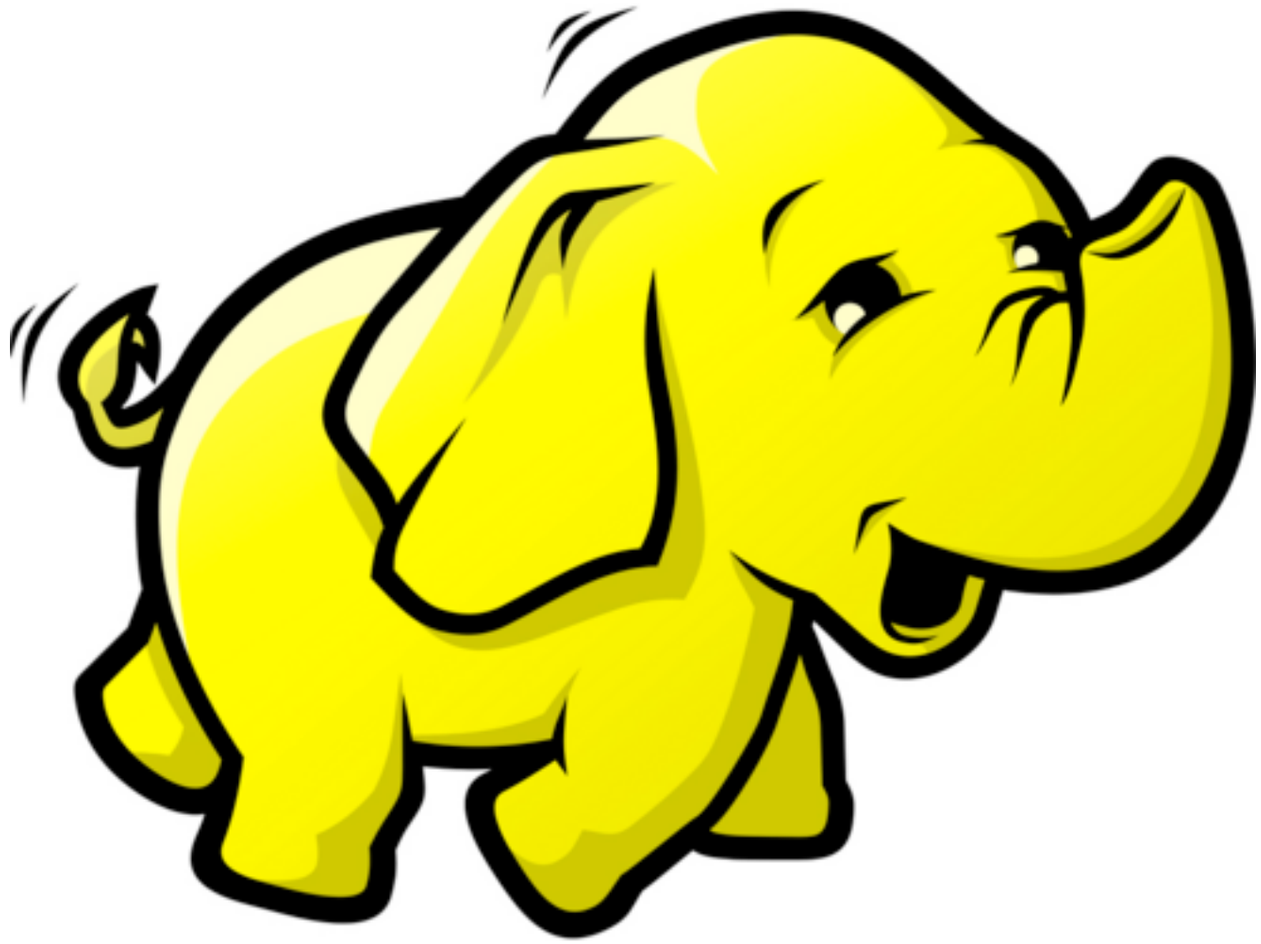
"The original motivation for the Nutch project was to provide a transparent alternative to the growing power of a handful of private search services over most users' view of the Web.

However, as Nutch has been adopted with greater enthusiasm by smaller organizations, the Nutch Organization has de-emphasized operating a multi-billion-page index in the public interest."

CommerceNet Labs Technical Report, Nov 2004



again?




FUN
ZONE


MONOPOLY
GOOGLE
DON'T BE EVIL
GOOGLE CARD
CHANCE
CHANCE
GO
JUST
VISITING
Yahoo!
Google
CARD
BROADBAND PROVIDER
FREE
PARKING
CHANCE
BROADBAND PROVIDER
LAWYER FEES
GO TO
COURT
ebay
TESCO
asos
BT
kelkoo
Google
CARD
Wikipedia
Google
CARD
amazon
YouTube
Google
CARD
facebook
twitter
sky
BROADBAND PROVIDER
CHANCE
CONVENIENCE
TAX
twitter
facebook

Why did ProtonMail vanish from Google search results for months?

Posted Oct 27, 2016 by Natasha Lomas (@riptari)

If you're the maker of a popular, zero access encrypted webmail product and suddenly discover your product is no longer featuring in Google search results for queries such as "secure email" and "encrypted email," what do you conclude?

That something is amiss, for sure.

But the rather more pertinent question is whether your product's disappearance is accidental or intentional — given that Google also offers a popular webmail product, Gmail, albeit one that does not offer zero access because users "pay" the company with their personal data, which feeds into Alphabet's user profiling and ad targeting engines.

transparency

reproducibility

common search:

common search: python EN

About 41380 results

Welcome to **Python**.org
www.python.org
The official home of the **Python** Programming Language

Dive Into **Python**
www.diveintopython.net
This book lives at . If you're reading it somewhere else, you may not have the latest version.

The Eric **Python** IDE
eric-ide.python-projects.org
Eric is a full featured **Python** editor and IDE, written in **Python**. It is based on the cross platform Qt gui toolkit, integrating the highly flexible Scintilla...

Starship
www.python.net
The home of pythonistas

Tutorials, **Python** Courses: Online and On Site
python-course.eu
Free comprehensive online tutorials suitable for self-study and high-quality on site **Python** courses in Europe, Canada (Toronto) and the US

learning **python** | one man's journey into **python**…
www.learningpython.com
one man's journey into **python**...

https://uidemo.commonsearch.org

Common Search Explainer | Search test | URL test | About

python | OK | EN

Results (50)

Welcome to Python.org

[debug] https://www.python.org/

The official home of the Python Programming Language

docid -4478921722574158000

static rank 0.7923434

ES score 87.821815

ES explain

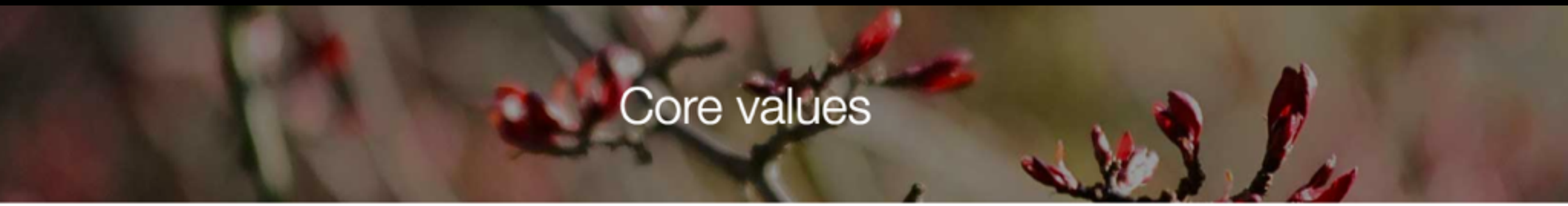
```
47.75143 | sum of:
    47.75143 | function score, product of:
        60.87483 | max plus 0.5 times others of:
            60.768433 | weight(domain_words:python in 77874) [PerFieldSimilarity], result of:
                60.768433 | score(doc=77874,freq=1.0 = termFreq=1.0
), product of:
                    8.0 | boost
                    5.97652 | idf(docFreq=9023, maxDocs=3555860)
                    1.2709827 | tfNorm, computed from:
                        1.0 | termFreq=1.0
                        1.0 | parameter k1
                        0.75 | parameter b
                        2.31778 | avgFieldLength
                        1.0 | fieldLength
            0.21279304 | weight(body:python in 77874) [PerFieldSimilarity], result of:
                0.21279304 | score(doc=77874,freq=21.0), product of:
                    0.14198984 | queryWeight, product of:
                        6.976686 | idf(docFreq=9021, maxDocs=3555860)
                        0.020352047 | queryNorm
                    1.4986497 | fieldWeight in 77874, product of:
                        4.582576 | tf(freq=21.0), with freq of:
                            21.0 | termFreq=21.0
                        6.976686 | idf(docFreq=9021, maxDocs=3555860)
                        0.046875 | fieldNorm(doc=77874)
        0.78441995 | min of:
            0.78441995 | function score, score mode [multiply]
                0.7923434 | function score, product of:
                    1.0 | match filter: *:*
```

https://explain.commonsearch.org/?q=python&g=en

Agenda

- Values & tech choices
- Search engine components
- Challenges
- Opportunities

Values & tech choices



Core values

Mission
Values
Governance
People
Roadmap
F.A.Q.
Technology
Privacy
Data sources
Partners
Credits

Our core values are the DNA of Common Search. They clearly state what we stand for, what makes us different and provide guidance when making hard decisions.

Starting with the most essential:

Radical transparency. Our search results must be explainable and reproducible. All our code is open source and results are generated only using publicly available data. Transparency also extends to our governance, finances and day-to-day operations.

Independence. No single person, company or special interest must be able to influence the order of our search results to their benefit. Our board of trustees is the watchdog of that independence.

Public service. We want to build and operate a free service targeted at a large, mainstream audience. Our impact and ultimate contribution to the Web grows with our size so we should make our service accessible and useful to as many users as we can.

Pragmatism. Recognizing the immensity of the task at hand, we should be willing to accept short-term compromises when necessary, as long as they don't go against the values above.

Privacy. Users should be informed and in full control of the personal information they share with us and with any third parties.

Focus. We are building a search engine. Not a browser, not an operating system, not an encyclopedia. Collaboration with other organizations sharing our core values should be encouraged instead of replicating their efforts.

Frugalism. Lowering costs is easier than increasing revenue. Having a low burn rate reduces the influence of money and guarantees our long-term sustainability.

Participation. Contributing to Common Search should be easy for everyone, developer or otherwise. We must embrace and nurture an open community that will surely lead, in time, to a great search engine.

Radical transparency

- Open source (Apache License v2)
- Open data
- (Governance)

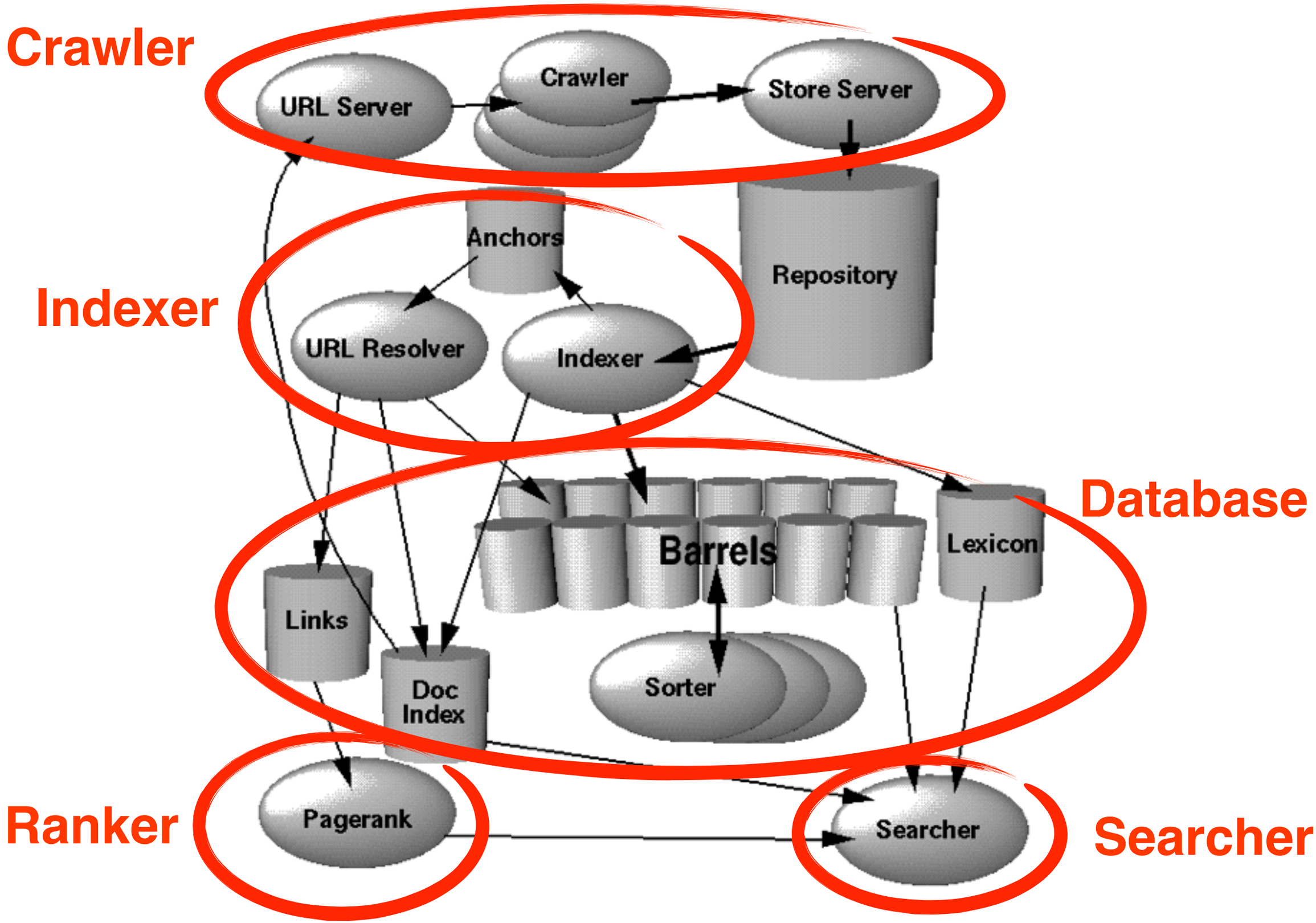
Privacy

- Results can be tailored by language/country, but NOT by user/cookie/sessionid
- \o/ Cache everything!
- Tor service: http://comsearchl2zlnre.onion

Participation & Pragmatism

- Use high-level languages as much as possible (Python, Go)
- Embrace active communities (Spark, Elasticsearch)
- Use mainstream participation platforms, even if they are nonfree (GitHub, Slack)

Search engines



The Anatomy of a Large-Scale Hypertextual Web Search Engine (1998)

http://infolab.stanford.edu/~backrub/google.html

Crawler

October 2016 Crawl Archive Now Available

November 7, 2016 **Sebastian Nagel**

The crawl archive for October 2016 is now available! The archive is located in the **commoncrawl** bucket at **crawl-data/CC-MAIN-2016-44/.** It contains more than 3.25 billion web pages.

Similar to the September crawl, we used sitemaps to improve the crawl seed list, including sitemaps named in the robots.txt file of the top-million domains from Alexa, and sitemaps from the top 150,000 hosts in Common Search's host-level page ranks. The maximum number of URL's extracted per domain was 200,000. The resulting crawl included 2 billion new URLs, not contained in previous crawls.

We are grateful to webxtrakt for donating a list of 14 million verified, DNS-resolvable domain names of European country-code TLDs (eu, .fr, .be, .de, .ch, .nl, .pl). We included these domains into the October crawl and we hope for a ongoing partnership with webxtract to improve the coverage of the crawls.

To assist with exploring and using the dataset, we provide gzipped files that list:

- all segments (CC-MAIN-2016-44/segment.paths.gz)
- all WARC files (CC-MAIN-2016-44/warc.paths.gz)
- all WAT files (CC-MAIN-2016-44/wat.paths.gz)
- all WET files (CC-MAIN-2016-44/wet.paths.gz)
- robots.txt files (CC-MAIN-2016-44/robotstxt.paths.gz)
- non-200 HTTP status code responses (CC-MAIN-2016-44/non200responses.paths.gz)

Recent Posts

October 2016 Crawl Archive Now Available

September 2016 Crawl Archive Now Available

News Dataset Available

August 2016 Crawl Archive Now Available

Data Sets Containing Robots.txt Files and Non-200 Responses

http://commoncrawl.org

Apache Nutch

Today at **3:30pm**!

HERITRIX

Scrapy

An open source and collaborative framework
for extracting the data you need from websites.
In a fast, simple, yet extensible way.

pypi v1.1.3 | wheel yes | Python 3 Porting Status | coverage 83%

Install the latest version of Scrapy

Scrapy 1.1

```
$ pip install scrapy
```

PyPI | Conda | APT | Source

Build and run your **web spiders**

Terminal

```
$ pip install scrapy
$ cat > myspider.py <<EOF
import scrapy

class BlogSpider(scrapy.Spider):
    name = 'blogspider'
    start_urls = ['https://blog.scrapinghub.com']

    def parse(self, response):
        for title in response.css('h2.entry-title'):
            yield {'title': title.css('a ::text').extract_first()}

        next_page = response.css('div.prev-post > a ::attr(href)').extract_first()
        if next_page:
            yield scrapy.Request(response.urljoin(next_page), callback=self.parse)
EOF
$ scrapy runspider myspider.py
```

http://scrapy.org

cocrawler / cocrawler

Watch 3 | Unstar 6 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs

CoCrawler is a versatile web crawler built using modern tools and concurrency.

193 commits | 1 branch | 0 releases | 1 contributor | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

wumpus process affinity is a nice win — Latest commit c87b825 5 days ago

cocrawler	process affinity is a nice win	5 days ago
examples	bump versions; measure decode cpu burn; tweaks	2 months ago
tests	process affinity is a nice win	5 days ago
.gitignore	shinier	3 months ago
.travis.yml	ok 3.5.0 is gone	29 days ago
LICENSE	initial import	3 months ago
README.md	update blather	6 days ago
TODO	move dns to separate file; first pytest-asyncio tests	12 days ago
requirements.txt	add histograms	7 days ago

http://github.com/cocrawler/cocrawler

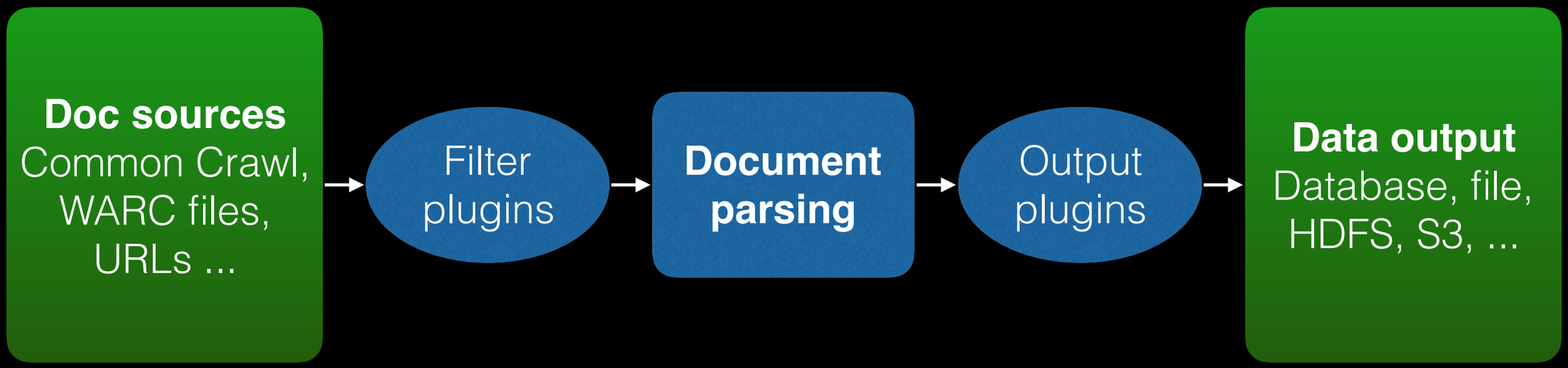
Indexer

Specs

- HTML parsing & analysis
- Tokenization / NLP
- Static rankings
- Language detection
- I/O from crawls to databases

APACHE Spark™

Common Search Pipeline




```
spark-submit [spark_options] \
    /cosr/back/spark/jobs/pipeline.py \
    --source [source_options] \
    --plugin [plugin_options] \
    [other_pipeline_options]
```

HTML parsers

- BeautifulSoup & friends
- lxml
- html5lib
- Gumbo!

google / gumbo-parser

Watch 321 | Star 3,818 | Fork 502

Code | Issues 30 | Pull requests 6 | Projects 0 | Wiki | Pulse | Graphs

An HTML5 parsing library in pure C99

403 commits | 4 branches | 7 releases | 27 contributors | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

nostrademons committed on GitHub Merge pull request #367 from mominul/patch-1 ... Latest commit aa91b27 on Jun 29

benchmarks	Add baidu benchmark which has been left out of the git repository all...	2 years ago
examples	Recognize templates in serialize and prettyprint	2 years ago
python/gumbo	Update gen_tags.py to exempt generated files from clang-format, and r...	a year ago
src	Fix error mesage use of return value form vnsprintf	10 months ago
testdata @ e633ddf	Move html5lib-tests submodule ref up to include the ruby fix.	2 years ago
tests	Added a test for fragments with multiple nodes.	a year ago
third_party	Integrate gumbo_parser with gtest.	3 years ago
visualc	Update strings.h	2 years ago
.clang-format	Reformat the source code with clang-format, and add a config file for...	a year ago

https://github.com/google/gumbo-parser

Gumbocy

- Use Cython instead of ctypes
- Smaller API
- Tree traversal on the Cython side with basic boilerplate/visibility support

https://github.com/commonsearch/gumbocy

urlparse4

`urlparse4` is a performance-focused replacement for Python's `urlparse` module, using C++ code from Chromium's own URL parser.

It is not production-ready yet.

Many credits go to gurl-cython for inspiration.

Differences with Python's `urlparse`

`urlparse4` should be a transparent, drop-in replacement in almost all cases. Still, there are a few differences to be aware of:

- `urlparse4` is 2-7x faster for most operations (see benchmarks below)
- `urlparse4` currently doesn't pass CPython's `test_urlparse.py` suite due to edge cases that Chromium's parser manages differently (usually in accordance to the RFCs, which `urlparse` doesn't follow entirely).
- `urlparse4` only supports Python 2.7 for now

How to install

```
pip install urlparse4
```

How to use

The most straightforward way to use `urlparse4` is to replace your imports of `urlparse` with this:

```
import urlparse4 as urlparse
```

https://github.com/commonsearch/urlparse4

Database(s)

Document 1

The bright blue butterfly hangs on the breeze.

Document 2

It's best to forget the great sky and to retire from every wind.

Document 3

Under blue sky, in bright sunlight, one need not search around.

Stopword list

a
and
around
every
for
from
in
is
it
not
on
one
the
to
under

Inverted index

ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

Search with Apache So @ select provider

Core (Java) | Solr | PyLucene

Ultra-fast Search Library and Server

Apache Lucene and Solr set the standard for search and indexing performance

Welcome to Apache Lucene

The Apache Lucene™ project develops open-source search software, including:

- Lucene Core, our flagship sub-project, provides Java-based indexing and search technology, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities.
- Solr™ is a high performance search server built using Lucene Core, with XML/HTTP and JSON/Python/Ruby APIs, hit highlighting, faceted search, caching, replication, and a web admin interface.
- PyLucene is a Python port of the Core project.

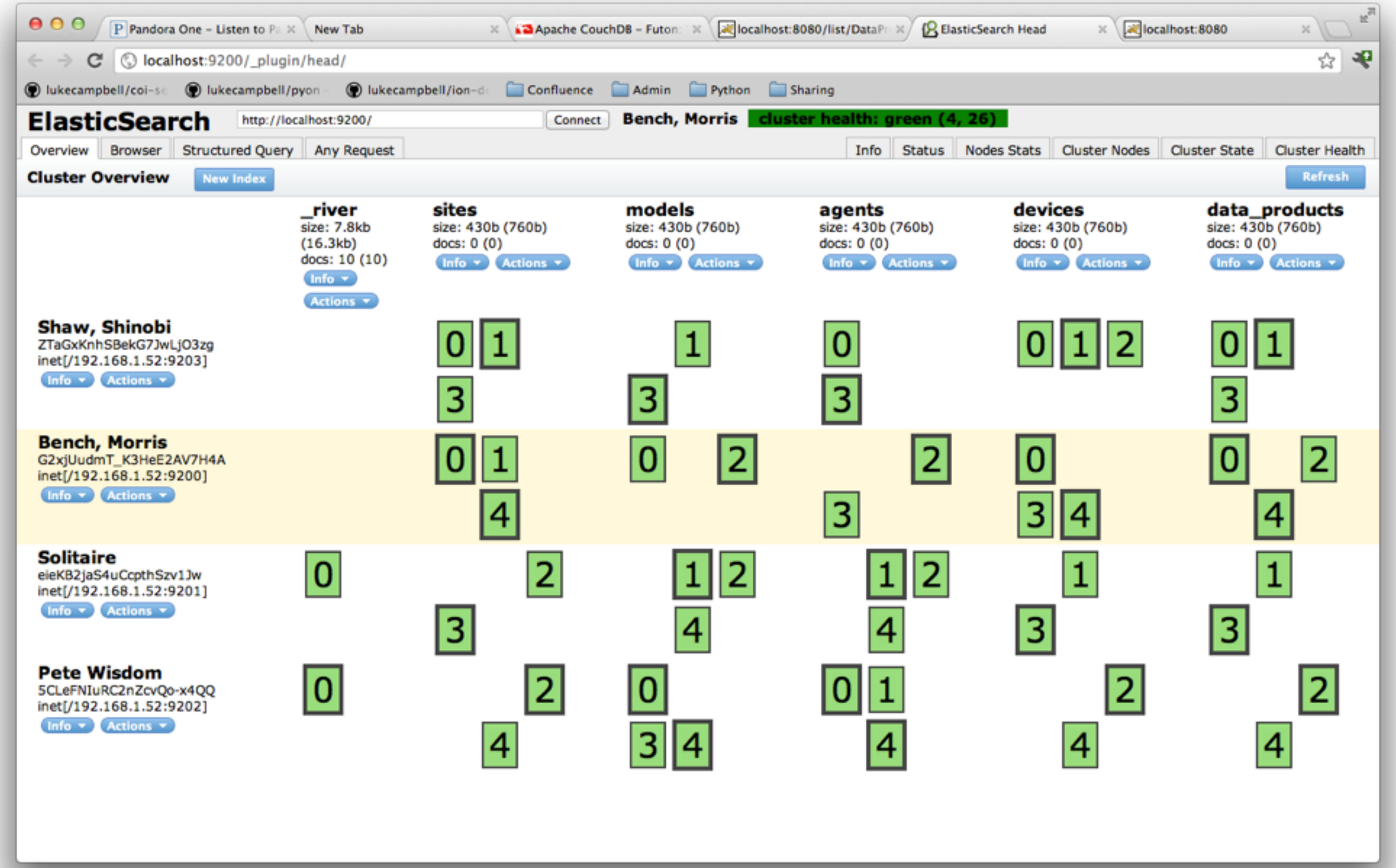
DOWNLOAD

Apache Lucene 6.2.1

DOWNLOAD

Apache Solr 6.2.1

http://lucene.apache.org/


Pandora One - Listen to Pa
New Tab
Apache CouchDB - Futon:
localhost:8080/list/DataPr
ElasticSearch Head
localhost:8080
localhost:9200/_plugin/head/
lukecampbell/coi-se
lukecampbell/pyon
lukecampbell/ion-d
Confluence
Admin
Python
Sharing
ElasticSearch
http://localhost:9200/
Connect
Bench, Morris
cluster health: green (4, 26)
Overview
Browser
Structured Query
Any Request
Info
Status
Nodes Stats
Cluster Nodes
Cluster State
Cluster Health
Cluster Overview
New Index
Refresh
_river
size: 7.8kb (16.3kb)
docs: 10 (10)
sites
size: 430b (760b)
docs: 0 (0)
models
size: 430b (760b)
docs: 0 (0)
agents
size: 430b (760b)
docs: 0 (0)
devices
size: 430b (760b)
docs: 0 (0)
data_products
size: 430b (760b)
docs: 0 (0)
Info
Actions
Shaw, Shinobi
ZTaGxKnhSBekG7JwLjO3zg
inet[/192.168.1.52:9203]
Bench, Morris
G2xjUudmT_K3HeE2AV7H4A
inet[/192.168.1.52:9200]
Solitaire
eieKB2jaS4uCcpthSzv1Jw
inet[/192.168.1.52:9201]
Pete Wisdom
5CLeFNIuRC2nZcvQo-x4QQ
inet[/192.168.1.52:9202]

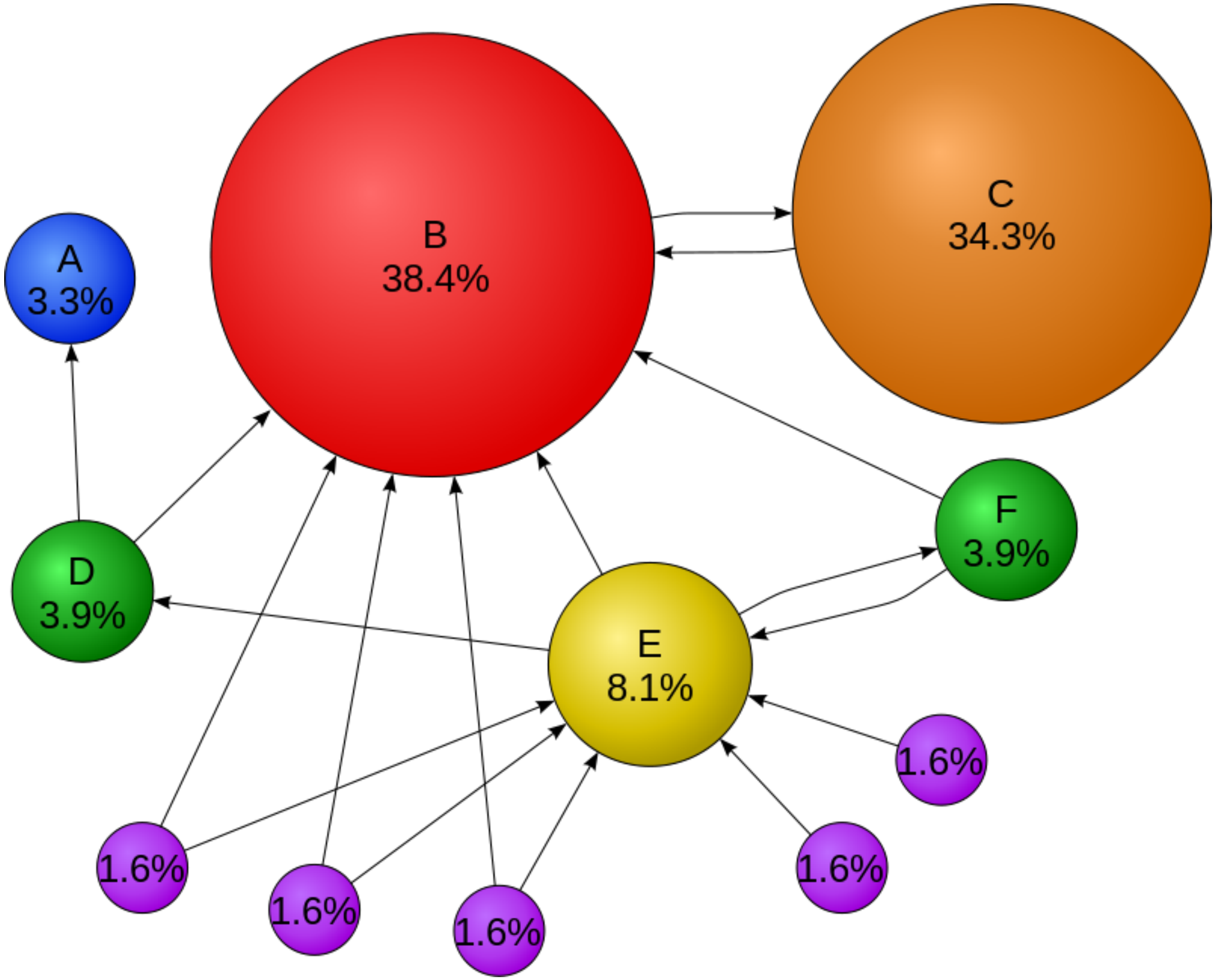
Ranker

Ranking formula

rank = f(static_score , dynamic_score(query))

Alexa
DMOZ
Blacklists
PageRank
...

ElasticSearch & Lucene
TF-IDF
BM25
...


A
3.3%
B
38.4%
C
34.3%
D
3.9%
E
8.1%
F
3.9%
1.6%
1.6%
1.6%
1.6%
1.6%
1.6%

Tutorial: Running PageRank on the Web

Get Started
Architecture
Backend
Frontend
Operations
Result Quality
Tutorial: 1st Frontend patch
Tutorial: Analyzing the web with Spark
Tutorial: Running PageRank on the web

This tutorial get you through all the steps required to run PageRank on billions of pages using Common Search's codebase and tools such as Apache Spark and AWS.

1. Prerequisites

You should go through our Analyzing the web with Spark on EC2 first, to install the required software, understand the basic concepts of our pipeline, and run a simpler job first, at least on your local machine.

You should also be familiar with basic Graph theory.

2. Dumping the Web Graph

Before computing PageRank, we need to parse all the link in our corpus and save them as a directed graph.

(In some cases, you can actually skip this step by using one of the dumps we publish directly.)

To dump the web graph, we are doing to use the `webgraph` plugin. Here is how you would dump it for the first 400 URLs from Common Crawl, at the host level:

```
spark-submit --verbose \
    /cosr/back/spark/jobs/pipeline.py \
    --source commoncrawl:limit=4,maxdocs=100 \
    --plugin plugins.webgraph.DomainToDomainParquet:path=out/webgraph/ \
    --stop_delay 600
```

This will actually create 2 subdirectories in `out/webgraph/`: one for the vertices and one for the edges. Both dumps will be stored as Apache Parquet format, so that we can easily reuse them in the next step.

You might notice this command will go over the source documents multiple times. This shouldn't be a big issue with so few

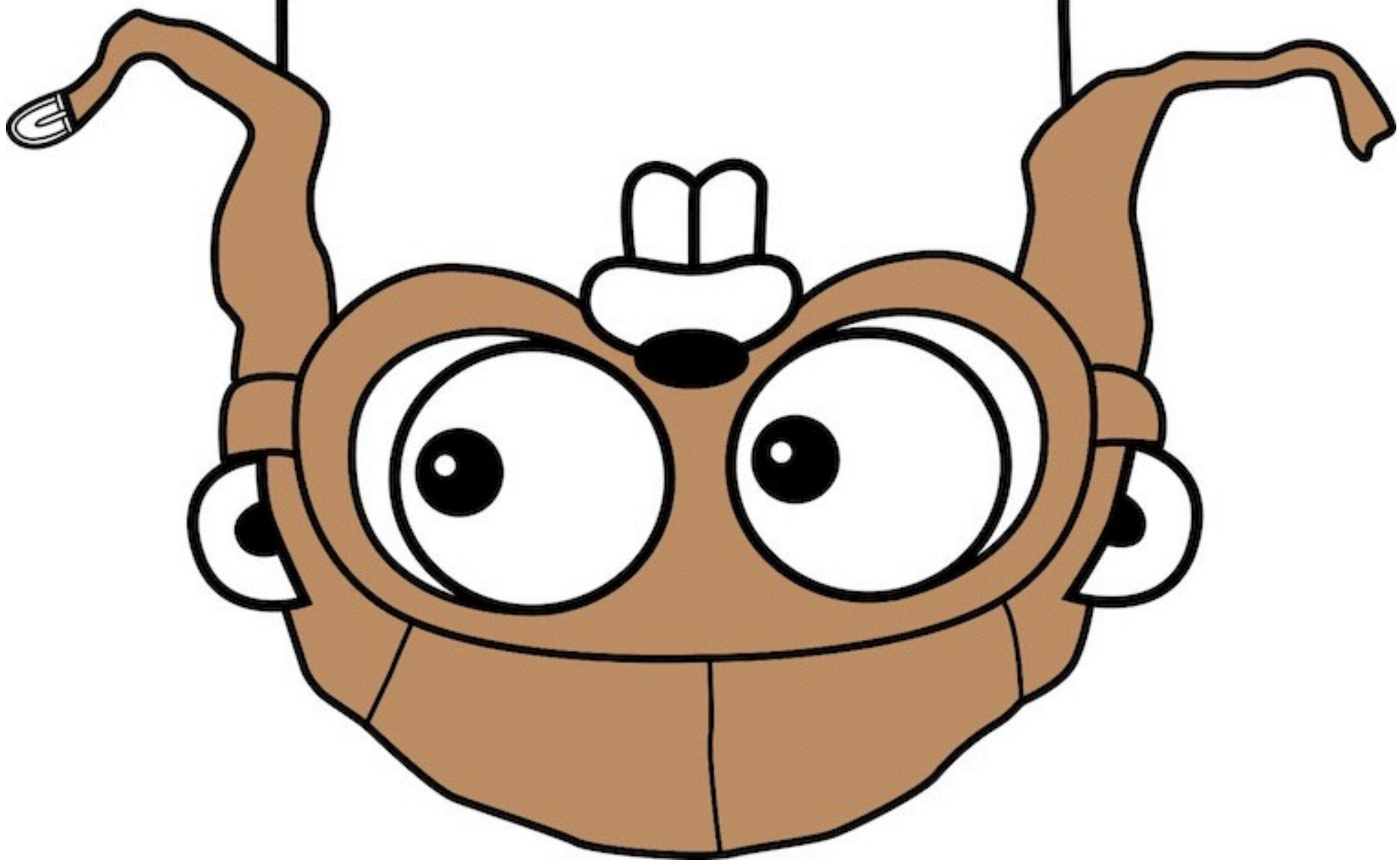
https://about.commonsearch.org/developer/get-started

Today @ 4:30pm ;-)

Searcher / Frontend

Specs

- Send user query to databases
- Search-as-you-type
- HTML & JSON endpoints
- High performance



common search: python EN

About 41380 results

Welcome to **Python**.org
www.python.org
The official home of the **Python** Programming Language

Dive Into **Python**
www.diveintopython.net
This book lives at . If you're reading it somewhere else, you may not have the latest version.

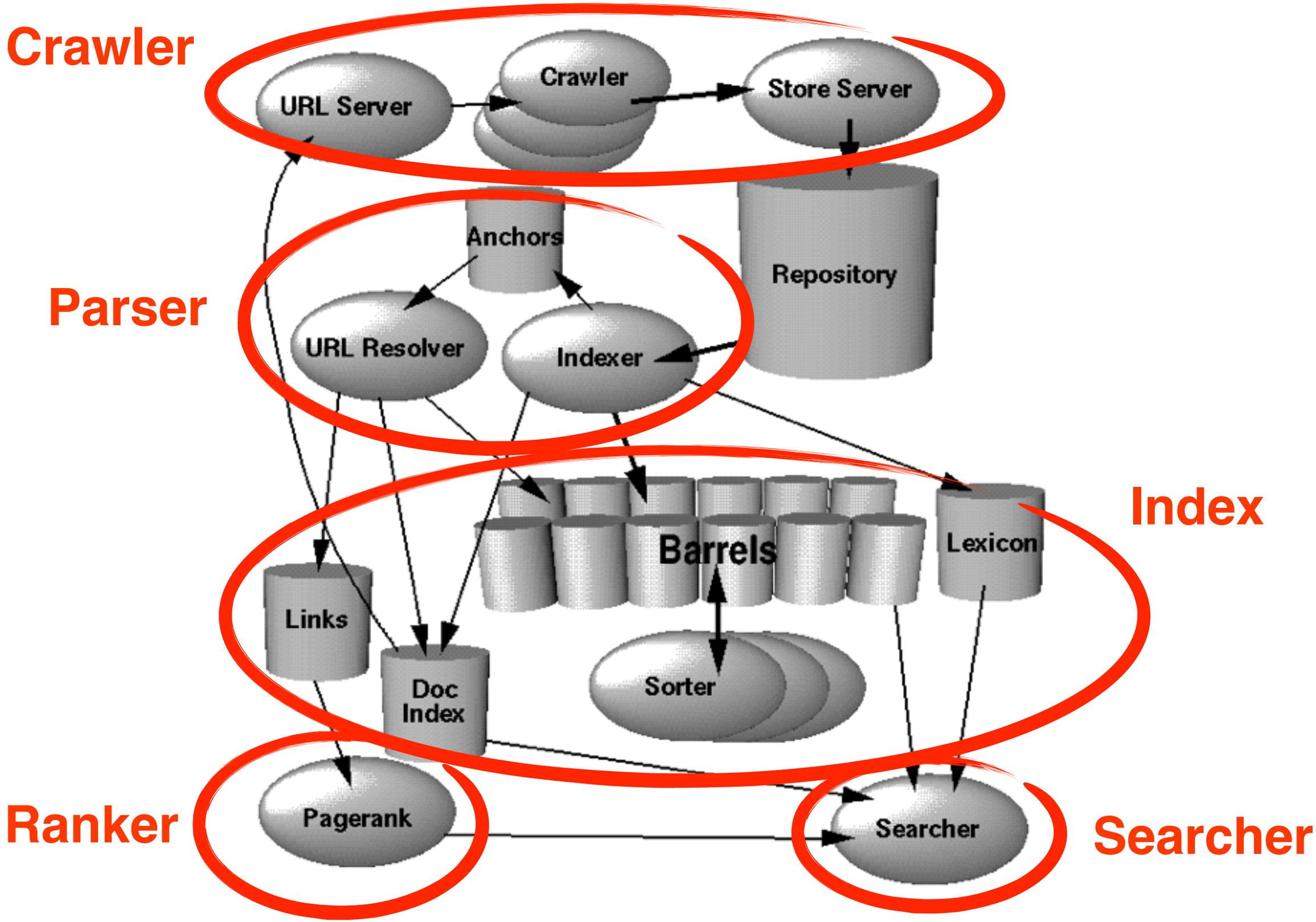
The Eric **Python** IDE
eric-ide.python-projects.org
Eric is a full featured **Python** editor and IDE, written in **Python**. It is based on the cross platform Qt gui toolkit, integrating the highly flexible Scintilla...

Starship
www.python.net
The home of pythonistas

Tutorials, **Python** Courses: Online and On Site
python-course.eu
Free comprehensive online tutorials suitable for self-study and high-quality on site **Python** courses in Europe, Canada (Toronto) and the US

learning **python** | one man's journey into **python**…
www.learningpython.com
one man's journey into **python**...

https://github.com/commonsearch/cosr-front



The Anatomy of a Large-Scale Hypertextual Web Search Engine (1998)

http://infolab.stanford.edu/~backrub/google.html

Challenges

Funding / Scale

- Frugalism
- Caching
- In-kind services
- Individual donations / Foundation grants
- General economic incentives

Spam

- Email spam
- Wikipedia vandalism
- Algorithm complexity & scale
- *Given enough eyeballs, all spam is shallow?*

Relevance

- Exhaustivity
- Rescoring
- Evaluation
- More at 4:30pm ;-)

More search dimensions

- Realtime search
- Local search
- Universal search

Semantic search

- Wikidata
- YAGO
- Conversational / Voice search

Outreach

- Easy onboarding & docs
- Making people ~~care~~ believe

Opportunities

Decentralization

- YaCy
- Extremely high technical & social cost!
- Transparency?

Research

- More people should know how to build search engines
- Spam, Relevance, Large-scale data processing
- We need more open datasets!

common search: About the project | UI Demo | How to contribute | Developers | Blog | Discuss | Contact

Our first public datasets: Host-level WebGraph and PageRank!

Jul 31, 2016 Back to blog

Common Search is building an open source search engine with transparent rankings, and analyzing the hyperlinks on the web is a major part of this effort.

To make that possible, we are going to publish datasets that will let contributors, students and researchers reproduce the rankings, submit improvements and hopefully use the underlying data for their own work.

The first two we are happy to announce today are a **host-level WebGraph** and a list of **host-level PageRanks**.

We want to give credit to both Common Crawl for their amazing work and to the Web Data Commons project who published similar dumps in 2012 and 2014.

Our datasets are released under a Creative Commons Attribution 4.0 license.

Host-level WebGraph

This dataset is based on the June 2016 Common Crawl. It represents the directed graph of all hyperlinks aggregated at the hostname level (e.g. "about.commonsearch.org").

- vertices.txt.gz (575M lines, 2.3 GB). Format: `[int64 id] [hostname]`
- edges.txt.gz (112M lines, 4.7 GB). Format: `[int64 src_id] [int64 dst_id]`

The Python code used to generate these files is available on GitHub!

Host-level PageRank

This dataset was generated directly from the Host-level WebGraph above and contains a PageRank for

https://about.commonsearch.org/blog/

Make the Web a better place!

- SEO
- Transparency
- Influence of money
- Public service

Questions?

https://about.commonsearch.org/contributing

https://github.com/commonsearch

contact@commonsearch.org

slack.commonsearch.org