



IIPC Metadata Workshop

Brad Tofel
Vinay Goel
Aaron Binns
Internet Archive



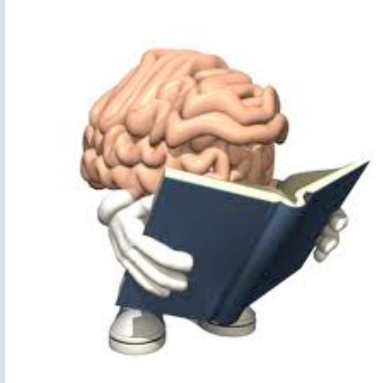


How can I implement analysis?





Researchers



Engineers



Tools



Hardware



DATA!!! : D



Researchers



Folks who want to do analysis of some kind on your data.

Some will have technical expertise, and know exactly what they want, some will have a vague notion..





Engineers

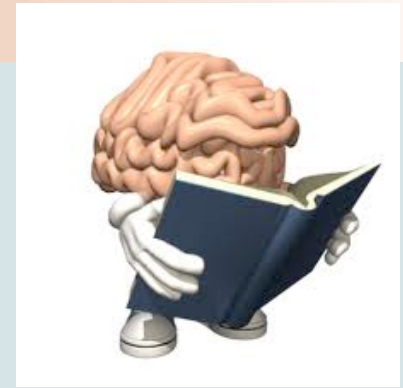
Individuals familiar with:

The data

The tools

The infrastructure

Translate researcher questions into code.





Tools

This is the major component we'll be focusing on today:

Hadoop DFS

Hadoop Map Reduce

IgPay AtinLay

Web Archive specific tools



Hardware



How much you need depends on what you will use it for, and how much data you need to process

Can grow over time

IA configuration:

40 x 2 Core, 4 GB, 4x700GB disks

40 x 4 Core, 16 GB, 4x1TB disks





Data



ARCs and WARCs are “heavy”

WAT – Web Archive Transformation

Uses WARC format as a generic meta data container

Extract everything you're likely to want from
ARCs/WARCs once

Transferable?!?

Feedback requested!

Store into HDFS?

Part of standard ingest process?



WAT

```
{  
  "Container": {  
    "Filename": "TENN-000001.warc.gz",  
    "Offset": "677"  
  },  
  "Envelope": {  
    "Format": "WARC",  
    "Payload-Metadata": {  
      "Actual-Content-Type": "application/http; msgtype=response",  
      "HTTP-Response-Metadata": {  
        "Headers": {  
          "Accept-Ranges": "bytes",  
          "Connection": "close",  
          "Content-Length": "22",  
          "Content-Type": "text/plain",
```



Tools: Redux

HDFS

Map Reduce

Pig Latin

Web archive code

Other extraction layers: Tika, Jhove(2), etc



HDFS

Distributed

Durable

Reliable

Scalable

Data stored in blocks on the same nodes that run processes

Single “head” node stores filename, permissions, and knows where a files blocks are located – right now.



Map Reduce

2 parts:

Programming model, “map()” and “reduce()”

Execution framework which distributes jobs and manages the 10s, 100s, or 1000s of nodes where those jobs run

Fault tolerant

Robust

Scalable



Map Reduce programming



client

Submit job



Job tracker

Run Job



Hadoop cluster

Client sends map-reduce job to Job Tracker, inside a “Jar” file
Job Tracker distributes Jar file to cluster, assigns tasks, monitors
Hadoop cluster runs job, stores result
Job Tracker informs client of success



Pig Latin

Map-Reduce programs are not too hard to write, but can get verbose for multi-stage jobs, and have low code reuse

Many interesting operations require more than one job

Pig Latin:

- High level data flow language

- Allows concise expressing of complex transformations

- Sends multiple Map-Reduce jobs to a cluster

- Manages intermediate data, cleanup

- Handles messy Joins!



Example: Count MIME Types in CDX

--Load CDX lines

```
CDXLines = LOAD '/tmp/example.cdx' USING PigStorage(' ') AS (url, timestamp, orig_url,mime,response_code,checksum,redirect_url,offset,filename);
```

--Group CDX lines by MIME Type

```
GrpdMimes = GROUP CDXLines BY mime;
```

--Count number of occurrences of each MIME type

```
CountMimes = FOREACH GrpdMimes GENERATE group, COUNT(CDXLines) AS cnt;
```

--Order the counts in descending order

```
SortedCountMimes = ORDER CountMimes BY cnt DESC;
```

--Dump the output to screen

```
DUMP SortedCountMimes;
```



Web archive code

Currently just a bit of glue code around an ARC/WARC reader

Does HTML metadata extraction

Includes example “UDF” code

simplifies and speeds up injecting external libraries into Pig Latin, to expand Pig Latin's capabilities

Will integrate with Jhove(2), Tiki, etc



Trivial Example: URL + Title

-- Load IA magic sauce:

```
REGISTER /home/brad/archive-meta-extractor-20110413.jar;
```

-- load data from INPUT_DIR:

```
Orig = LOAD '/tmp/sample-wats/' USING  
org.archive.hadoop.ArchiveJSONViewLoader('Envelope.WARC-Header-  
Metadata.WARC-Target-URI','Envelope.Payload-Metadata.HTTP-Response-  
Metadata.HTML-Metadata.Head.Title') AS (src,title);
```

-- discard lines without titles

```
PagesWithTitles = FILTER Orig BY title != "";
```

-- remove duplicates

```
Result = DISTINCT PagesWithTitles;
```

```
STORE Result INTO '/tmp/sample-output/' USING PigStorage();
```



Interoperability, Sharing UDFs

Common formats enables sharing of data, code, experience across institutions

As various institutions develop more UDFs and Pig Latin scripts, the whole community grows more capable

