Part 2 / 2

Part 2

Sorting in Hadoop MapReduce — it really doesn't sort!

Continued analysis of forensicswiki.org log files to determine:

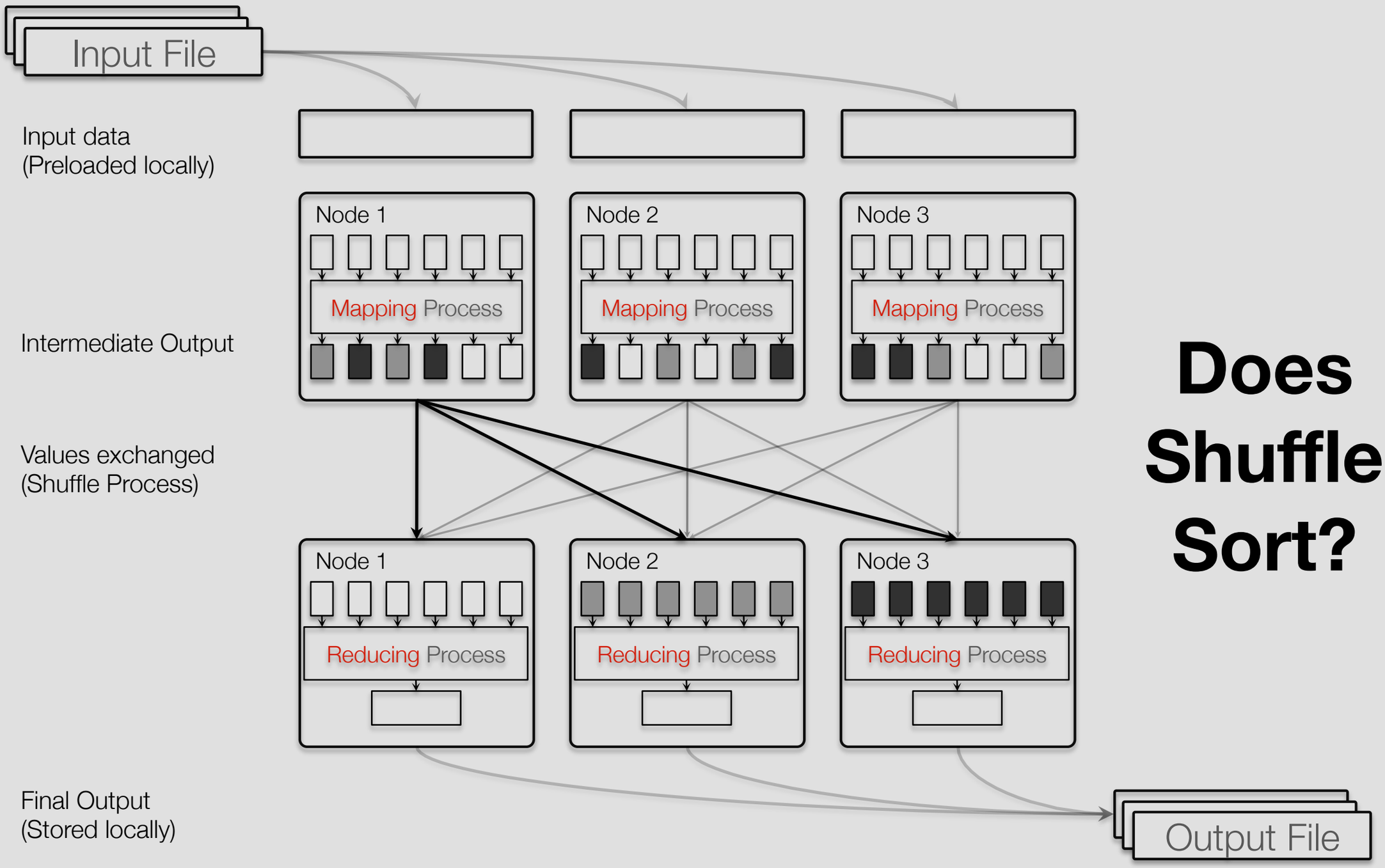
—*How many web pages could be cached by a Content Distribution Network (CDN)?*

Data Wrangling with MTR

- Compute the distribution of (*page name*, *page length*) tuples.
- See how a 1-day cache policy impacts savings.

Sorting and MapReduce

MapReduce



Does Shuffle Sort?

http://megasthenis.github.io/repository/Hadoop-WNCG-Intro.pdf

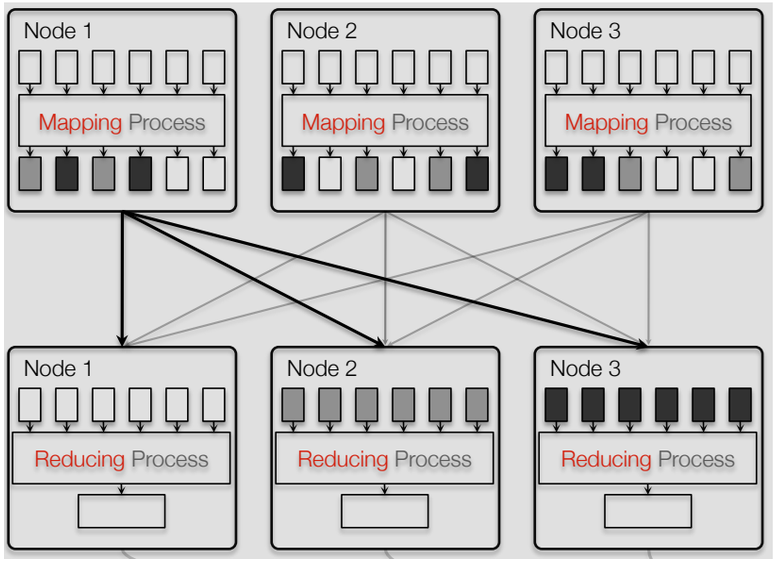
To find out of Shuffle Sorts… conduct an experiment.

Evidence that shuffle sorts:

- Output from forensicswiki.org log file analysis was sorted.

Evidence against shuffle sorting:

- Hadoop MapReduce supports TotalOrderPartitioner



- forensicswiki.org log file was only 4.2GB in size:

```
$ aws s3 ls s3://gu-anly502/logs/
2017-01-08 18:56:48 4,268,793,922 forensicswiki.2012.txt
$
```

- Default node type: m3.xlarge

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Hypotheses:

- If the work mostly fits in memory, Hadoop sorts the keys.

Test: Give Hadoop a really big problem!

Conducting a big data experiment

Our goal is to ***test the infrastructure.***

- Minimize confounding variables.
- Small example.
- Use code that is believed to work.

```
#!/usr/bin/env python34
#
from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+")

class MRWordFreqCount(MRJob):

    def mapper(self, _, line):
        for word in WORD_RE.findall(line):
            yield word.lower(), 1

    def reducer(self, word, counts):
        yield word, sum(counts)

if __name__ == '__main__':
    MRWordFreqCount.run()
```

Approach:

- Use L03/wordcount.py — simple code that we've used
- Use "big data"
  - *This is what se used before:*

```
$ aws s3 ls s3://gu-anly502/logs/
2017-01-08 18:56:48 4,268,793,922 forensicswiki.2012.txt
```

  - *This is what we'll use:*

```
2017-01-21 15:36:36            676 A2/sonnet18.txt
2017-01-08 19:16:19 39,489,364,082 A3/quazyilx3.txt
```

  - *(Commas added!)*
  - *Sonnet 18 provides words through the alphabet, while quazyilx3 provides lots of data.*
- Use a lot of nodes
  - *Reduces opportunity for sorting in memory*

Create a big cluster

1 master, 5 core, 5 task

m1.medium — *whoops!*

Bid price $0.10 — spot price was $0.01 (except for us-east-1b)

dvanced Options Go to quick options

Hardware Configuration

If you need more than 20 EC2 instances, complete this form.

Network vpc-f3401097 (172.31.0.0/16) (default) Create a VPC

EC2 Subnet subnet-056b0d3b | Default in us-east-1e

Node type	EC2 instance type	Instance count	Storage per instance	Request spot	Bid price	Auto Scaling
Master Master - 1	m1.medium	1	410 GiB Add EBS volumes	✔	.1	
Core Core - 2	m1.medium	5	410 GiB Add EBS volumes	✔	.1	
Task ✖ Task - 3	m1.medium	5	410 GiB Add EBS volumes	✔	.1	Not enabled

Availability zone	Price
us-east-1a	$0.010
us-east-1b	$0.870
us-east-1c	$0.010
us-east-1d	$0.010
us-east-1e	$0.010

+ Add task instance group

Cancel Previous Next

Using M1 was a mistake…

I tried to save money (and I did), but performance was horrible.

- https://aws.amazon.com/ec2/previous-generation/

Upgrade Paths

We encourage you to use the latest generation of instances to get the best performance, but we will continue to support Previous Generation Instances after new instances launch If you are currently using a Previous Generation Instance and would like to see which one would be a suitable upgrade, see the table below and learn how the latest generation of instances could benefit you.

T1/M1 to T2

M1 to M3

M3 instances provide better, more consistent performance than M1 instances for most use-cases. M3 instances also offer SSD-backed instance storage that delivers higher I/O performance. M3 instances are also less expensive than M1 instances. Due to these reasons, we recommend M3 for applications that require general purpose instances with a balance of compute, memory, and network resources.

M1		M3
No	SSD Storage	Yes
No	Latest Intel Xeon Processor	Yes
Good	I/O Performance	Better

- No SSD, job took ≈ 17 hours

Running the job

Command line:

```
$ python3.4 wordcount.py -r hadoop -o hdfs:///tmp/output1 \
  s3://gu-anly502/A1/quazyilx3.txt s3://gu-anly502/A2/sonnet18.txt
```

…

```
Using configs in /home/hadoop/.mrjob.conf
Using Hadoop version 2.7.3
Looking for Hadoop streaming jar in /usr/lib/hadoop...
Found Hadoop streaming jar: /usr/lib/hadoop/hadoop-streaming.jar
Creating temp directory /tmp/wordcount.hadoop.20170212.035119.087363
Copying local files to hdfs:///user/hadoop/tmp/mrjob/wordcount.hadoop.20170212.035119.087363/
files/...
Running step 1 of 1...
  packageJobJar: [] [/usr/lib/hadoop/hadoop-streaming-2.7.3-amzn-1.jar] /tmp/
streamjob8238248022899493747.jar tmpDir=null
  Connecting to ResourceManager at ip-172-31-48-188.ec2.internal/172.31.48.188:8032
  Loaded native gpl library
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev
d94115f47e58e29d8113a887a1f5c9960c61ab83]
  Total input paths to process : 2
  number of splits:590
  Submitting tokens for job: job_1486864426796_0001
  Submitted application application_1486864426796_0001
  The url to track the job: http://ip-172-31-48-188.ec2.internal:20888/proxy/
application_1486864426796_0001/
  Running job: job_1486864426796_0001
  Job job_1486864426796_0001 running in uber mode : false
   map 0% reduce 0%
   map 1% reduce 0%
   map 2% reduce 0%
   map 3% reduce 0%
   map 4% reduce 0%
   map 5% reduce 0%
```

Job finally finished...

```
 map 100% reduce 99%
 map 100% reduce 100%
Job job_1486864426796_0001 completed successfully
Output directory: hdfs:///tmp/output1

Counters: 56
        File Input Format Counters
                Bytes Read=39,497,563,548
        File Output Format Counters
                Bytes Written=2668
        File System Counters
                FILE: Number of bytes read=10,672,675,982
                FILE: Number of bytes written=14,636,772,618
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=49559
                HDFS: Number of bytes written=2668
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=1237
                HDFS: Number of write operations=38
                S3: Number of bytes read=39,497,563,548
                S3: Number of bytes written=0
                S3: Number of large read operations=0
                S3: Number of read operations=0
                S3: Number of write operations=0
```

commas added

commas added

commas added

job counters

```
Job Counters
  Data-local map tasks=592
  Killed map tasks=2
  Killed reduce tasks=3
  Launched map tasks=592
  Launched reduce tasks=22
  Total megabyte-milliseconds taken by all map tasks=320882654976
  Total megabyte-milliseconds taken by all reduce tasks=558950498304
  Total time spent by all map tasks (ms)=417815957
  Total time spent by all maps in occupied slots (ms)=10027582968
  Total time spent by all reduce tasks (ms)=545850096
  Total time spent by all reduces in occupied slots (ms)=17467203072
  Total vcore-milliseconds taken by all map tasks=417815957
  Total vcore-milliseconds taken by all reduce tasks=545850096
```

2 maps killed!
3 reduces killed!
22 reducers

```
Map-Reduce Framework
        CPU time spent (ms)=435433960
        Combine input records=0
        Combine output records=0
        Failed Shuffles=0
        GC time elapsed (ms)=963966
        Input split bytes=49559
        Map input records=752981151
        Map output bytes=81539162911
        Map output materialized bytes=4852231744
        Map output records=10541736000
        Merged Map outputs=11210
        Physical memory (bytes) snapshot=240022339584
        Reduce input groups=210
        Reduce input records=10541736000
        Reduce output records=210
        Reduce shuffle bytes=4852231744
        Shuffled Maps =11210
        Spilled Records=31625207876
        Total committed heap usage (bytes)=173340581888
        Virtual memory (bytes) snapshot=798113402880
```

=7257 min
=120 hours

Final results...

```
Streaming final output from
hdfs:///tmp/output1...
"04"    143432129
"2"     117180434
"2012"  15811932
"2029"  15770780
"2044"  15813545
"36"    39587021
"51"    25102037
"sonnet"        1
"too"   2
"10"    261348976
"2003"  15768565
"2035"  15769086
"27"    64291151
"42"    39580631
"59"    25099265
"day"   1
"nature"        1
"of"    3
"s"     3
"so"    2
"sometime"      2
"william"       1
"01"    145504579
"18"    198356262
"2026"  15766842
"2041"  15768110
"33"    39582683
"cark"  752981134
"course"        1
"eyes"  1
"shade" 1
"09"    143009149
"2000"  15808402
"2017"  15768824
"2032"  15815697
"24"    64289064
"56"    25103542
"7"     117165904
"as"    1
"can"   2
"shake" 1
"shall" 3
"15"    198397271
"2008"  15811072
"2023"  15765107
"30"    62212400
"47"    39583032
...
```

The output went into *many* HDFS files

```
[hadoop@ip-172-31-48-188 L03]$ hdfs dfs -ls  /tmp/output1
Found 20 items
-rw-r--r--   2 hadoop hadoop          0 2017-02-12 22:04 /tmp/output1/_SUCCESS
-rw-r--r--   2 hadoop hadoop        124 2017-02-12 16:14 /tmp/output1/part-00000
-rw-r--r--   2 hadoop hadoop        153 2017-02-12 16:34 /tmp/output1/part-00001
-rw-r--r--   2 hadoop hadoop        123 2017-02-12 20:37 /tmp/output1/part-00002
-rw-r--r--   2 hadoop hadoop        140 2017-02-12 16:23 /tmp/output1/part-00003
-rw-r--r--   2 hadoop hadoop        125 2017-02-12 15:56 /tmp/output1/part-00004
-rw-r--r--   2 hadoop hadoop        170 2017-02-12 16:33 /tmp/output1/part-00005
-rw-r--r--   2 hadoop hadoop        140 2017-02-12 22:04 /tmp/output1/part-00006
-rw-r--r--   2 hadoop hadoop        134 2017-02-12 17:35 /tmp/output1/part-00007
-rw-r--r--   2 hadoop hadoop        131 2017-02-12 19:18 /tmp/output1/part-00008
-rw-r--r--   2 hadoop hadoop        132 2017-02-12 16:17 /tmp/output1/part-00009
-rw-r--r--   2 hadoop hadoop        132 2017-02-12 14:42 /tmp/output1/part-00010
-rw-r--r--   2 hadoop hadoop        113 2017-02-12 15:25 /tmp/output1/part-00011
-rw-r--r--   2 hadoop hadoop        157 2017-02-12 17:35 /tmp/output1/part-00012
-rw-r--r--   2 hadoop hadoop        107 2017-02-12 16:26 /tmp/output1/part-00013
-rw-r--r--   2 hadoop hadoop        175 2017-02-12 18:08 /tmp/output1/part-00014
-rw-r--r--   2 hadoop hadoop        168 2017-02-12 19:28 /tmp/output1/part-00015
-rw-r--r--   2 hadoop hadoop        174 2017-02-12 15:58 /tmp/output1/part-00016
-rw-r--r--   2 hadoop hadoop        137 2017-02-12 17:11 /tmp/output1/part-00017
-rw-r--r--   2 hadoop hadoop        133 2017-02-12 15:31 /tmp/output1/part-00018
[hadoop@ip-172-31-48-188 L03]$
```

part-00000

```
$ hdfs dfs -cat  /tmp/output1/part-00000
"04"      143432129
"2"       117180434
"2012"    15811932
"2029"    15770780
"2044"    15813545
"36"      39587021
"51"      25102037
"sonnet"  1
"too"     2
$
```

Note:

- Sorted within the part, but not between parts
- 1.4 billion "04" words. — this might not have fit in memory!
  - *That's why the reducer gets an iterator of values, rather than an array.*
- The same program that worked a toy example handled this.

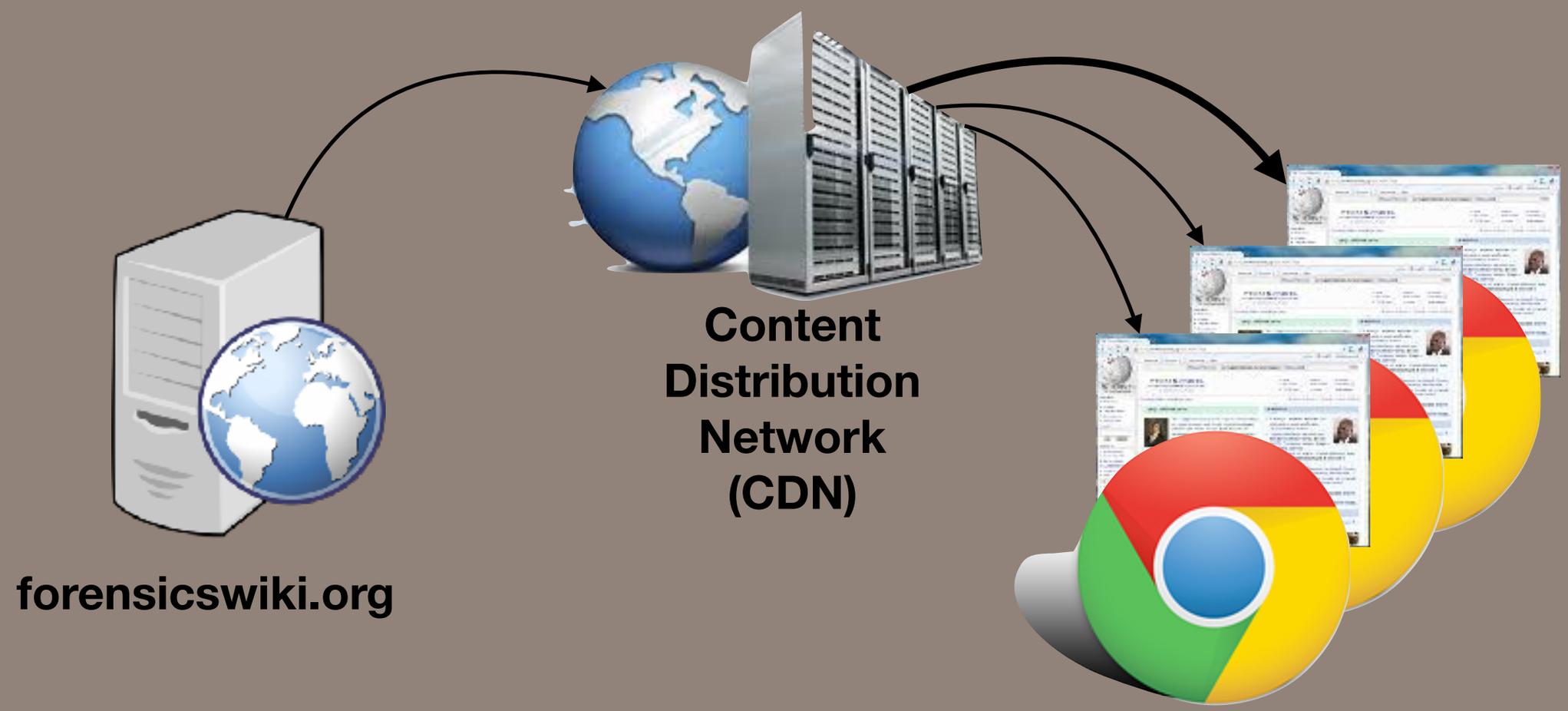
Further reading

"What is secondary sort in Hadoop, and how does it work?"

- https://www.quora.com/What-is-secondary-sort-in-Hadoop-and-how-does-it-work

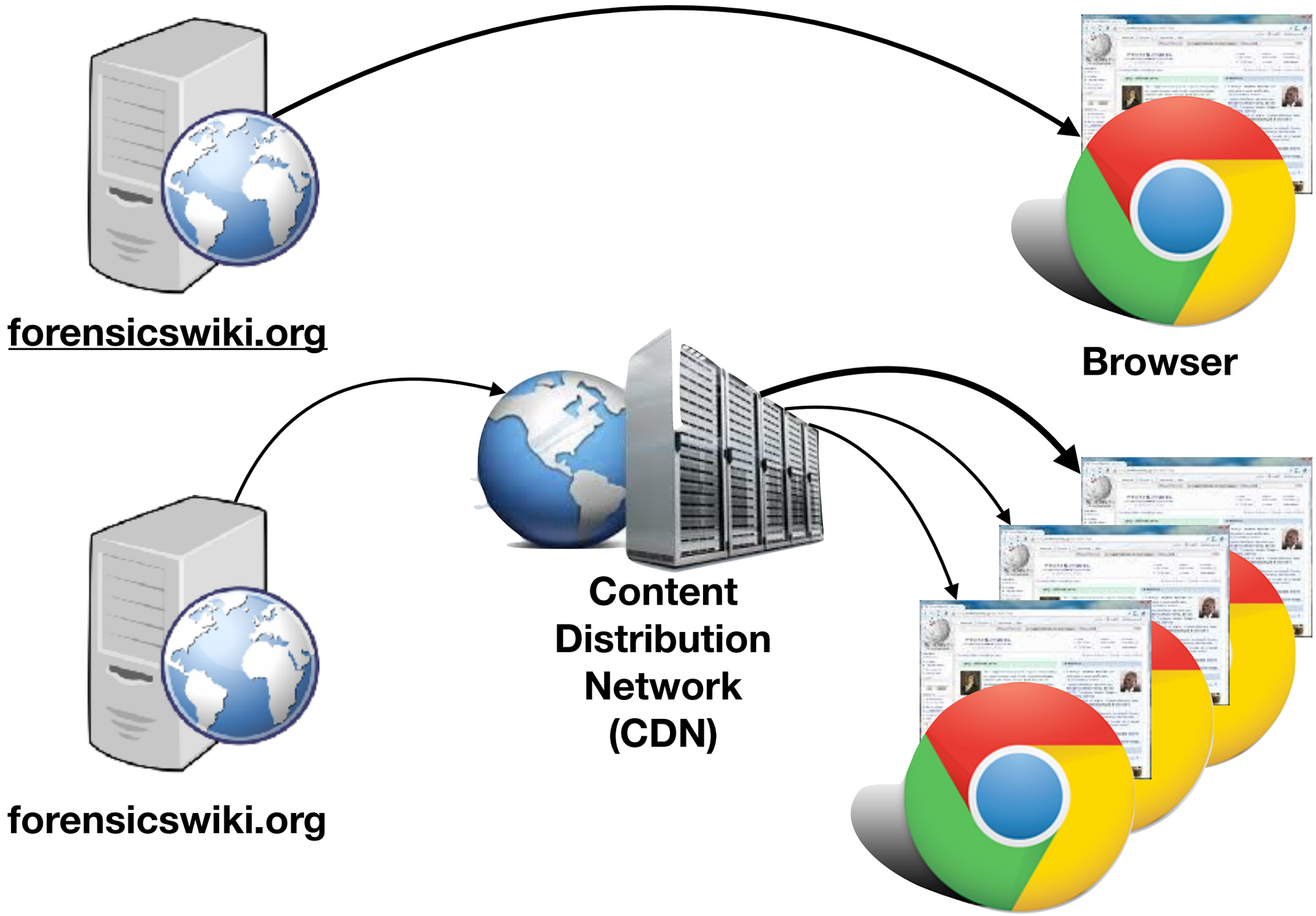
"What is the purpose of "uber mode" in hadooop?"

- http://stackoverflow.com/questions/30284237/what-is-the-purpose-of-uber-mode-in-hadoop

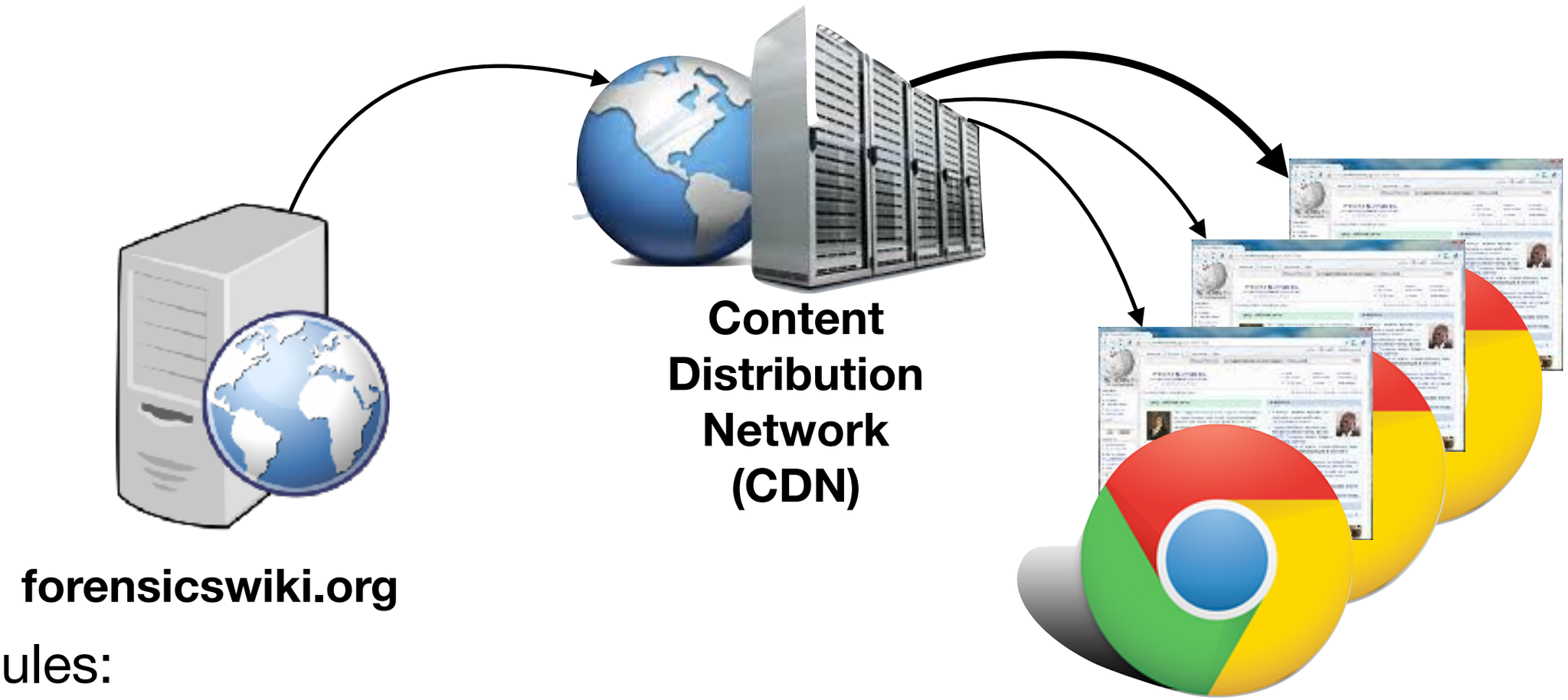


A worked example... continued

Basic Web Architecture



CDN — Key points



CDN Rules:

1. Page moves from web server to CDN *with first request*.
2. Subsequent requests satisfied by CDN.
3. After time T, the page is removed from the CDN.

To answer:

1. How many page fetches would benefit from a CDN?

The Forensics Wiki logfile:

On S3:

```
$ aws s3 ls s3://gu-anly502/
                                   PRE A1/
                                   PRE A2/
                                   PRE A3/
                                   PRE L04/
                                   PRE gutenberg/
                                   PRE logs/
                                   PRE maxmind/
                                   PRE new folder/
                                   PRE ps03/
                                   PRE ps04/
```

.

```
$ aws s3 ls s3://gu-anly502/logs/
2017-01-08 23:56:48 4268793922 forensicswiki.2012.txt
```

.

```
$ aws s3 cp s3://gu-anly502/logs/forensicswiki.2012.txt - | head -2
77.21.0.59 - - [01/Jan/2012:00:35:03 -0800] "GET /wiki/Write_Blockers
HTTP/1.1" 200 5742 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
77.21.0.59 - - [01/Jan/2012:00:35:04 -0800] "GET /w/skins/common/
wikibits.js?270 HTTP/1.1" 200 31165 "http://www.forensicswiki.org/wiki/
Write_Blockers" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/534.52.7 (KHTML, like Gecko) Version/5.1.2 Safari/534.52.7"
```

Outline of our solution

For every forensicswiki.org URL:

- Generate a list of each time (URL,size) pair was accessed.
- Report savings.
- Implement cache flush strategy (flush after 1 day)
- Report savings

How to develop code:

- Develop with a small excerpt of the wiki code & MRJOB in Local Mode

The new log file parser

LogLine() is a class that takes a log file line and returns a LogLine() object:

```
CLR_RE = re.compile(r'^(\S+) (\S+) (\S+) \[([^\]]+)\] "(\S+) (\S+) \S+" (\S+)
(\S+) "([^"]*)" "([^"]*)"')

class LogLine(object):
    __slots__ = ['ipaddr','datetime','verb','path','code','bytes','refer','user_agent']

    def __init__(self,line):
        import datetime
        m = CLR_RE.search(line)
        if m:
            self.ipaddr     = m.group(1)
            self.datetime   = datetime.datetime.strptime(m.group(4),"%d/%b/%Y:%H:%M:%S %z")
            self.verb       = m.group(5)
            self.path       = m.group(6)
            self.code       = int(m.group(7))
            self.bytes      = int(m.group(8))
            self.refer      = m.group(9)
            self.user_agent = m.group(10)
        else:
            self.ipaddr     = None
            self.datetime   = None
            self.verb       = None
            self.path       = None
            self.code       = None
            self.bytes      = None
            self.refer      = None
            self.user_agent = None
```

The new FWikiAnalyzer:

```
# This job accepts the basic log files and outputs
class FWikiAnalyzer(MRJob):
    SORT_VALUES = True
    def mapper(self, _, line):
        w = LogLine(line)
        try:
            if w.ipaddr:
                yield ( w.path, w.bytes ), w.datetime.isoformat()
            else:
                self.increment_counter("Info","CLR_RE Unmatched Lines",1)
        except RuntimeError as e:
            self.increment_counter("Error","RuntimeError in FWikiAnalyzer.mapper",1)
            pass

    def reducer(self, url_count, ones):
        yield url_count, sum((1 for x in ones))


if __name__ == '__main__':
    FWikiAnalyzer.run()
```

Convert dates to isoformat()

Notice the generator

Run on small scale:

$ python3.4 fwiki_cdn1.py -r hadoop forensicswiki_10k.txt

```
["/", 388]        11
["/", 433]        5
["/", 469]        3
["//Forensics_Wiki:General_disclaimer", 537]    1
["//Talk:Main_Page", 518]5
["//favicon.ico", 275]   1
["//www.forensicswiki.org/wiki/Talk:Abbreviated_Dialing_Numbers",
599]      1
["/Talk:BitLocker_Disk_Encryption", 534] 1
["/\\\"", 546]   1
["/apple-touch-icon.png", 500]   3
["/favicon.ico", 220]    43
["/images/7/79/Maxtor_Disk_Geometry.pdf", 322039]       1
["/images/e/ea/?C=M;O=A", 531]   1
["/images/f/f9/Mcdaniel01.pdf", 176]    1
["/images/forensics_logo.jpg", 37041]   1
```

How can we improve this?

Improvements 1

1 - Suppress output that can't be cached

```
["/images/f/f9/Mcdaniel01.pdf", 176]    1
```

```
def reducer(self, url_count, ones):
    hits = sum((1 for x in ones))
    if hits > 1:
        yield url_count, hits
```

Improvements 2

2 - Keep track of stats with counters

```
def mapper(self, _, line):
  w = LogLine(line)
  try:
      if w.ipaddr:
          yield ( w.path, w.bytes ), w.datetime.isoformat()
          self.increment_counter("Info","Total hits",1)
          self.increment_counter("Info","Total bytes",w.bytes)
      else:
          self.increment_counter("Info","CLR_RE Unmatched Lines",1)
  except RuntimeError as e:
      self.increment_counter("Error","RuntimeError in FWikiAnalyzer.mapper",1)
      pass
```

Need to change mapper test

Old:

```
def test_mapper():
    for (key, value) in FWikiAnalyzer.mapper(None,"",log):
        assert key[0] == "/w/skins/common/wikibits.js?270"
        assert key[1] == 31165
        assert value  == '2012-01-01T00:35:04-08:00'
```

New

```
def test_mapper():
    fw = FWikiAnalyzer()
    for (key, value) in fw.mapper("",log):
        assert key[0] == "/w/skins/common/wikibits.js?270"
        assert key[1] == 31165
        assert value  == '2012-01-01T00:35:04-08:00'
```

Reason:

```
     self.increment_counter("Info","CLR_RE Unmatched Lines",1)
```

self must be an object that has an .increment_counter() method!

output 2 (on reduced set)

```
Counters: 53
  File Input Format Counters
          Bytes Read=2611461
  File Output Format Counters
          Bytes Written=21800
  File System Counters
          FILE: Number of bytes read=123352
          FILE: Number of bytes written=1709682
          FILE: Number of large read operations=0
          FILE: Number of read operations=0
          FILE: Number of write operations=0
          HDFS: Number of bytes read=2612941
          HDFS: Number of bytes written=21800
          HDFS: Number of large read operations=0
          HDFS: Number of read operations=33
          HDFS: Number of write operations=6
  Info
          CLR_RE Unmatched Lines=2
          Total bytes=73779488
          Total hits=9998
```

Improvement #3

3 - Have the reducer track the cache hits:

```
def reducer(self, url_count, dates):
    import dateutil.parser, datetime
    # Look for a gap longer than a day.
    prev = None
    cached = 0
    hits  = 0
    for day in dates:
        d = dateutil.parser.parse(day)
        if prev:
            # We have a previous hit. Is the current one more than a day?
            # If not, it is cached

            if d - prev < datetime.timedelta(0,24*60*60):
                cached += 1      # note that one was cached!
        prev = d
        hits += 1
    # We have finished the run. Print hits and saves for this one
    if cached>0:
        yield url_count,{"hits":hits,"cached":cached}

    # And remember the total cached
    self.increment_counter("Info","Cached hits",cached)
    self.increment_counter("Info","Cached bytes",cached * url_count[1])
```

Example output (limited input)

```
Counters: 5
   Info
              CLR_RE Unmatched Lines=2
              Cached bytes=61150815
              Cached hits=8764
              Total bytes=73779488
              Total hits=9998


["/wiki/Write_Blockers", 223]       {"hits": 1, "cached": 0}
["/wiki/Write_Blockers", 260]       {"hits": 3, "cached": 2}
["/wiki/Write_Blockers", 302]       {"hits": 1, "cached": 0}
["/wiki/Write_Blockers", 5741]      {"hits": 12, "cached": 11}
["/wiki/Write_Blockers", 5742]      {"hits": 5, "cached": 4}
["/wiki/Write_Blockers/", 9997]     {"hits": 1, "cached": 0}
["/wiki/Xmount", 4937]   {"hits": 1, "cached": 0}
["/wiki/Xplico", 5096]   {"hits": 1, "cached": 0}
["/wiki/YAFFS2", 11712]  {"hits": 1, "cached": 0}
["/wiki/YAFFS2", 4354]   {"hits": 1, "cached": 0}
["/wiki/admin/banner_manager.php/login.php", 413]         {"hits": 6, "cached": 5}
["/wiki/admin/categories.php/login.php", 409]  {"hits": 8, "cached": 7}
["/wiki/admin/file_manager.php/login.php", 411] {"hits": 7, "cached": 6}
["/wiki/admin/sqlpatch.php/password_forgotten.php?action=execute", 532]         {"hits": 5,
"cached": 4}
["/wiki/digital_forensic_research_workshop", 495]         {"hits": 1, "cached": 0}
["/wiki/email_headers/index.php?user-register", 3681]     {"hits": 1, "cached": 0}
["/wiki/forensics_wiki:about", 481]  {"hits": 1, "cached": 0}
["/wiki/helix", 384]     {"hits": 1, "cached": 0}
["/wiki/index.php?title=Special:UserLogin&type=signup", 3625]        {"hits": 2, "cached": 1}
["/wiki/index.php?title=Special:UserLogin&type=signup", 3626]        {"hits": 4, "cached": 3}
["/wiki/mobiledit!", 471] {"hits": 1, "cached": 0}
["/wiki/recovering_deleted_data", 484]         {"hits": 1, "cached": 0}
["/wiki/sim_cards", 414]  {"hits": 1, "cached": 0}
["/wiki/special:recentchanges", 381] {"hits": 1, "cached": 0}
["/wiki/user:simsong", 473]         {"hits": 1, "cached": 0}
["http://www.forensicswiki.org/wiki/Open_Research_Topics/", 10093]   {"hits": 3, "cached": 2}
```

Multiple targets

Hadoop python errors — Stored in the file system on every node. It is a lot faster to look at the file system than use S3 (5 min polling)

MRJob errors are written to:

```
/mnt/log/hadoop-yarn/containers
```

Example:

```
[root@ip-172-31-39-163 containers]# cd /mnt/log/hadoop-yarn/containers
[root@ip-172-31-39-163 containers]# ls -l
total 28
drwxr-x--- 22 yarn yarn 4096 Feb 13 04:44 application_1486954432295_0001
drwxr-x--- 14 yarn yarn 4096 Feb 13 04:50 application_1486954432295_0002
drwxr-x--- 14 yarn yarn 4096 Feb 13 05:02 application_1486954432295_0003
drwxr-x--- 14 yarn yarn 4096 Feb 13 05:33 application_1486954432295_0004
drwxr-x--- 22 yarn yarn 4096 Feb 13 05:37 application_1486954432295_0005
drwxr-x--- 17 yarn yarn 4096 Feb 13 05:39 application_1486954432295_0006
drwxr-x--- 18 yarn yarn 4096 Feb 13 05:42 application_1486954432295_0007
[root@ip-172-31-39-163 containers]#

[root@ip-172-31-39-163 containers]# cd application_1486954432295_0007
[root@ip-172-31-39-163 application_1486954432295_0007]# ls -l
total 0
drwxr-x--- 2 yarn yarn 45 Feb 13 05:40 container_1486954432295_0007_01_000001
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000002
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000003
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000004
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000005
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000006
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000007
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000008
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000009
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000010
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000011
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000012
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000013
drwxr-x--- 2 yarn yarn 45 Feb 13 05:41 container_1486954432295_0007_01_000014
drwxr-x--- 2 yarn yarn 45 Feb 13 05:42 container_1486954432295_0007_01_000015
drwxr-x--- 2 yarn yarn 45 Feb 13 05:42 container_1486954432295_0007_01_000016
[root@ip-172-31-39-163 application_1486954432295_0007]#
```

You can get the actual python errors!

```
[root@ip-172-31-39-163 application_1486954432295_0007]# cat */*stderr

...
Traceback (most recent call last):
  File "fwiki_cdn2.py", line 80, in <module>
    FWikiAnalyzer.run()
  File "/mnt/yarn/usercache/hadoop/appcache/application_1486954432295_0007/
container_1486954432295_0007_01_000003/mrjob.zip/mrjob/job.py", line 439, in run
  File "/mnt/yarn/usercache/hadoop/appcache/application_1486954432295_0007/
container_1486954432295_0007_01_000003/mrjob.zip/mrjob/job.py", line 448, in execute
  File "/mnt/yarn/usercache/hadoop/appcache/application_1486954432295_0007/
container_1486954432295_0007_01_000003/mrjob.zip/mrjob/job.py", line 526, in run_mapper
  File "fwiki_cdn2.py", line 43, in mapper
    w = LogLine(line)
  File "fwiki_cdn2.py", line 20, in __init__
    self.datetime   = datetime.datetime.strptime(m.group(4),"%m/%b/%Y:%H:%M:%S %z")
  File "/usr/lib64/python3.4/_strptime.py", line 500, in _strptime_datetime
    tt, fraction = _strptime(data_string, format)
  File "/usr/lib64/python3.4/_strptime.py", line 337, in _strptime
    (data_string, format))
ValueError: time data '13/Jan/2012:00:00:11 -0800' does not match format '%m/%b/%Y:%H:%M:%S %z'
...
```

Should be %d
Test case of Jan 1 did not catch this!

additional

Here are some lines that didn't match:

```
unmatched= 72.35.92.170 - - [01/Jan/2012:02:08:41 -0800] "GET /\" +
gaJsHost + \"google-analytics.com/ga.js HTTP/1.1" 301 517 "-" "Java/
1.5.0_09"

unmatched= 92.82.225.48 - - [01/Jan/2012:08:07:16 -0800] "GET /\" +
gaJsHost + \"google-analytics.com/ga.js HTTP/1.1" 301 517 "-" "Java/
1.6.0_04"
```

Finished run

```
...
  File System Counters
          FILE: Number of bytes read=215180794
          FILE: Number of bytes written=446268771
          FILE: Number of large read operations=0
          FILE: Number of read operations=0
          FILE: Number of write operations=0
          HDFS: Number of bytes read=4270829250
          HDFS: Number of bytes written=0
          HDFS: Number of large read operations=0
          HDFS: Number of read operations=105
          HDFS: Number of write operations=6
  Info
          CLR_RE Unmatched Lines=586
          Cached bytes=161801664426
          Cached hits=13044268
          Total bytes=190677677466
          Total hits=15948968
...
```

Total savings: 161,801,664,426 / 190,677,677,466 = 84%