# BristolHadoopWorkshopSpring2010

## Bristol Hadoop Workshop Spring 2010

This was a one-day event hosted by HP Laboratories, Bristol, and co-organised by HPLabs and Bristol University. It was a followup to the 2009 workshop, again a meeting of locals to discuss what they were up to and look at Hadoop in physics, among other things.

### Julien Nioche: Behemoth

Slides

Julien Nioche at digitalPebble has been working on Natural Language Processing at scale.

- Started with Apache UIMA: fairly simple
- Now working on Behemoth, "Hadoop's evil twin":not a nice elephant at all

The goal is large scale document analysis based on Hadoop; to let you deploy GATE or UIMA applications on Hadoop clusters. It was driven by the need to implement this for more than one client client, opened it up to avoid writing from scratch every time.

Workflow: load to HDFS, import to Behemoth Doc format (PDF, HTML, WARC, Nutch segments, etc. uses Apache Tika to extract text and metadata). Output (key==URI, value=BehemothDocument)

Features

- Common ground between UIMA and GATE. (GATE is open source; Cloud GATE is not)
- Supports different (non-Java) annotators
- Easy to configure using the Hadoop config file format and Behemoth/UIMA rules in JARs
- Works on Hadoop the ecosystem

Demo: shows that the JobTracker JSP page had been extended with GATE metrics.

Future work: cascading support and Avro for cross-language code, SOLR and Mahout. It needs to be tested at scale. Run @200K documents so far, Julien would be interested in anyone with a datacentre and an NLP problem.

### James Jackson: Hadoop and High Energy Physics

James is from CERN and the CMS experiment -he spoke about ongoing work exploring using Hadoop for HEP event mining.

The LHC experiments ~~Atlas, CMS, etc~~ generate event data, most of which is uninteresting. Physics events can be split into

- Uninteresting and known physics
- Unknown and uninteresting. We don't have the theory ready for these events yet
- Unknown and interesting: stuff people are looking for that matches (somewhat) the current theories, gives you Nobel prizes and the like.

To make life complicated there is a lot of noise on the detectors, timing problems can have stuff come in out of order. You need to do a lot of filtering and look for signals a long way off random noise before you can declare that you've found something interesting.

Most physicists not only code as if they were writing FORTRAN, they never wrote good FORTRAN either. (this is a complaint by Greg Wilson in Toronto - the computing departments never teach software engineering to all the scientists who are expected to code as part of their day to day science).

HDFS has been used as a filestore in some of the US CMS Tier-2 sites, the new work that James discussed was that of actually treating physics problems as MapReduce jobs. They are bringing up a cluster of machines with storage for this, but would also like to use idle CPU time on other machines in the datacentre -there was some discussion on how to do this MAPREDUCE-1603 is now a feature request asking for a way to make the assessing of availability a feature that supported plugins. This would allow someone to write something that looked at non-Hadoop workload of machines and reduced the number Hadoop slots to report as being available when busy with other work.

### Leo Simons: The BBC

Leo spoke about their CouchDB back end for the BBC web site

- Codeswarm: live graphics of their repository work.
- There's a new BBC homepage http://www.live.bbc.co.uk
- The web page is integrated with iplayer.
- Friday afternoons are busy iPlayer times. People either skive off work or watch TV from their desk.
- Lets you change your prefs -no need to login, the preferences are just bound to cookies
- Uses a hash of JSON to drive CouchDB lookup, this lets them stay with 4M docs rather than 60M docs.
- They reach consistency in 40mS or so, no need for microsecond consistency as the rate of change of homepage is below that.
- Compaction reduced the status display to "blue", rather than green, had everyone panicing but no visible change in behaviour. Moral: use light green instead.

Lots of fun with incomplete resharding causing intermittent replication failures. When an app saw a 404, it created a new doc as it expected this and kept going, created extra load and resulted in a 7h replication.

## Steve Loughran, HP: New Roles in the cloud

Slides

Steve argued that with machine allocation/release being an API call away, you can avoid some of the problems of classic applications (needing large capital investment based on demand estimations), but there is a price: everything needs to be agile. There is no way to hard code hostnames into JSP, PHP or ASP pages; no way to offload High Availability problems to the hardware vendors. Your architects need to think about how to include load measurement in their design, how to make the application adapt to machines coming or going. Hadoop was cited as an example of an application designed to be un-agile: it does have hardcoded and cached hostnames in the configuration files; the workers' reaction to any NameNode or JobTracker failure is to spin waiting for it to come back, not to look up the hostnames in case they have moved. Similarly, the blacklisting process, while ideal for physical machines, is not the right way to deal with failures in virtual infrastructure, where the moment a machine starts playing up you ask for a new one.

The talk concluded with a demo of the CloudFarmer prototype UI, which is a simple front end on a model-driven infrastructure. In CloudFarmer, one person specifies the machine roles with disk image options, VM requirements, a list of (protocol, port, path) strings for URLS, and some other values. The web and RESTy interfaces then let callers create instances of each role; the URL lists are turned into absolute values for the web UI to work with.

Hadoop deployment with CloudFarmer was shown, and while HDFS came up, the JobTracker wasn't so happy. This led to a discussion on another problem in this world: debugging from log files in a world where the VMs can go away without much warning.

## Tim @last.fm: Hive

Slides of Hive @ last.fm

- last.fm have been using Hive for 6 months
- Their cluster receivsd 600 events/second. On a par withTtwitter right now, but twitter "tweets" are growing faster and they have to do notifications
- Cluster: 44 nodes, each with 8 cores, 16 GB RAM and 4x1TB 7200 RPM storage (=704 GB RAM, 176 TB of storage, 352 cores).
- Charts: what's being played?
- Reporting: what they owe record companies?
- Corrections: cleaning up user supplied data. Most user data is pretty messy.
- Neighbours: finding similar users.
- Lots of queries about stuff -effectively a form of data warehousing.
- Hive tables can be one or more HDFS files;
- Hive also lets them import tables without bothering to pull into the HDFS filestore
- Some patches for various formats, Twitter did one for protocol buffers.
- No support for EchoIO; last.fm tried to do one and gave up, used Dumbo to import it into HDFS instead.
- External data is trusted more than anything else. Hive is not a database, just a query tool.
- Some queries can take minutes if they have to schedule MR jobs

Why Hive?

1. Developers have some familiarity with SQL, especially the web team that live off python. Make queries like how many people hit a page. Business people don't do MySQL, like the advertising team; they bring the questions to the developers who use the console.
2. Liked the ability to import from different sources
3. It worked, at the time they looked, Pig didn't.

Example: Rage against the machine vs Joe from X-factor.

- Query: how many users listen to music on the radio after they've scrobbled from their own collection?
- The #of users that scrobble is << #of users that use the radio, but scrobbling users generate lots more data.
- Hive "explain" provides the execution plan.

Example: "reach": how many people have listened to an artist? Example: "popularity": how often it is listened to Workflow: scrobbles -> hive -> solr

### Weaknesses

- No recordio record support
- Big joins used to OOM, but this seems to have gone away. It used to join in RAM, fast but would OOM.
- Pig? Better for exploding data -cross products. Could be better for user functions
- After Last.fm upgraded to Cloudera 0.20 Hadoop, Hive would start jobs but not finish them. They didn't upgrade Hive at first. Recompiled Hive and eventually it went away. Cloudera's Hive release fixed this.
- The thrift server for hive stopped working as the Cloudera version pointed to a different lib which led to conflicts.

Question: Has anyone tried to do any Object-Relational mappings? no.

## Sanders van der Waal: Community Engagement

Slides

Sanders van der Waal from OSSWatch gave a talk on Community Engagement. OSSWatch provide consultation and some support on Open Source in UK Higher Education and Universities, and have been getting involved in Hadoop in the past year as it makes a good platform for some scientific research, as well as a place for CS people to explore scheduling problems.

Sanders emphasised that there is no Open Source community other than that which the users choose to make themselves; he also looked at the benefits of local groups ~~face to face discussion~~ with the risks -you are restricted in your contacts, and the discussions tend not to be archived/searchable as per mailing lists and bug tracker issues.

There is a workshop in Oxford on June 24 & 25 on technology transfer, followed by a BarCamp on June 26; all are welcome.

Sanders talk triggered an interesting discussion on whether the Grid model had delivered on what it had promised, or not. The answer: some stuff got addressed, but some things (storage) had been ignored, and turned out to be rather important.

## Thomas Sandholm: Economic Scheduling of Hadoop Jobs

Slides

Thomas Sandholm joined us from via videoconference to talk about the scheduler that he and Kevin Lai wrote.

- The two main schedulers are optimised for the Yahoo! and Facebook workloads. Although converging they are tuned differently; both teams are nervous about changes that would reduce their throughputs, as the cost would be significant.
- Hadoop 0.21 adds a plugin API to add your own scheduler more easily.
- The DynamicPriorityScheduler is designed for multiple users competing for time on a shared cluster.
- You bid for time; the scheduler gives priority to those who bid the most.
- You can bid $0, you will still get time if nobody else bids more than you.
- Running Map or Reduce jobs will get killed if higher priority work comes in. The scheduler tries to be clever here and leave stuff that has been running a while alone (on the expectation that it will finish soon). The benefits of killing processes comes in if people can schedule long running jobs.
- It avoids any kind of history to make it scalable, no need to worry about persistence.
- If your bid doesn't get through, you don't get billed.
- To use: give every user/team their own queue

The scheduler is in the contrib directory for Hadoop 0.21; it's not easy to backport as it uses the scheduler plugin API.