Intro to the Hadoop Tool Chain

HDFS, Sqoop, Pig, Hive, Map Reduce, & Oozie By Example

Joe McTee

About Me

- Principal Engineer at Tendril Networks
- Experience with embedded programming, server-side Java, big data, very basic web dev and sql
- JEKLsoft is my sandbox for experiment and play
- Contact me at
  - mcjoe@jeklsoft.com / jmctee@tendrilinc.com
  - @jmctee on twitter
  - https://github.com/jmctee for teh codez

About Tendril

OPEN for business.
We believe in the power of an **open platform** to deliver applications that **ENGAGE consumers** with energy providers and empower them to **take charge** of their **energy.**

- We develop products that bring consumers, utilities, and consumer product manufacturers together in a partnership to save energy while maintaining quality of life.
- http://tendrilinc.com

What is Hadoop

- A distributed file system
  - Defaults to HDFS but others supported
- An environment to run Map-Reduce jobs
  - Typically batch mode
- Other
  - NOSQL Database (HBase)
  - Real-time query engine (Impala)
  - New uses are cropping up all the time

What is Map-Reduce

- The "assembly language" of Hadoop
- An algorithm for efficiently processing large amounts of data in parallel
  - Rigorous mathematical underpinnings that we will ignore
- See http://en.wikipedia.org/wiki/MapReduce
- Optimized for data locality
  - Don't move data around, move code around, it is smaller

Map (Wikipedia)

- "Map" step: The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node.

Reduce (Wikipedia)

- "Reduce" step: The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

Shuffle-Sort

- After the map phase, data is "shuffled" to combine similar data from different mappers on a single reducer
- Sometimes, to optimize, a pre-reduce is run on the data prior to shuffle
  - This can decrease the amount of data transferred to other nodes

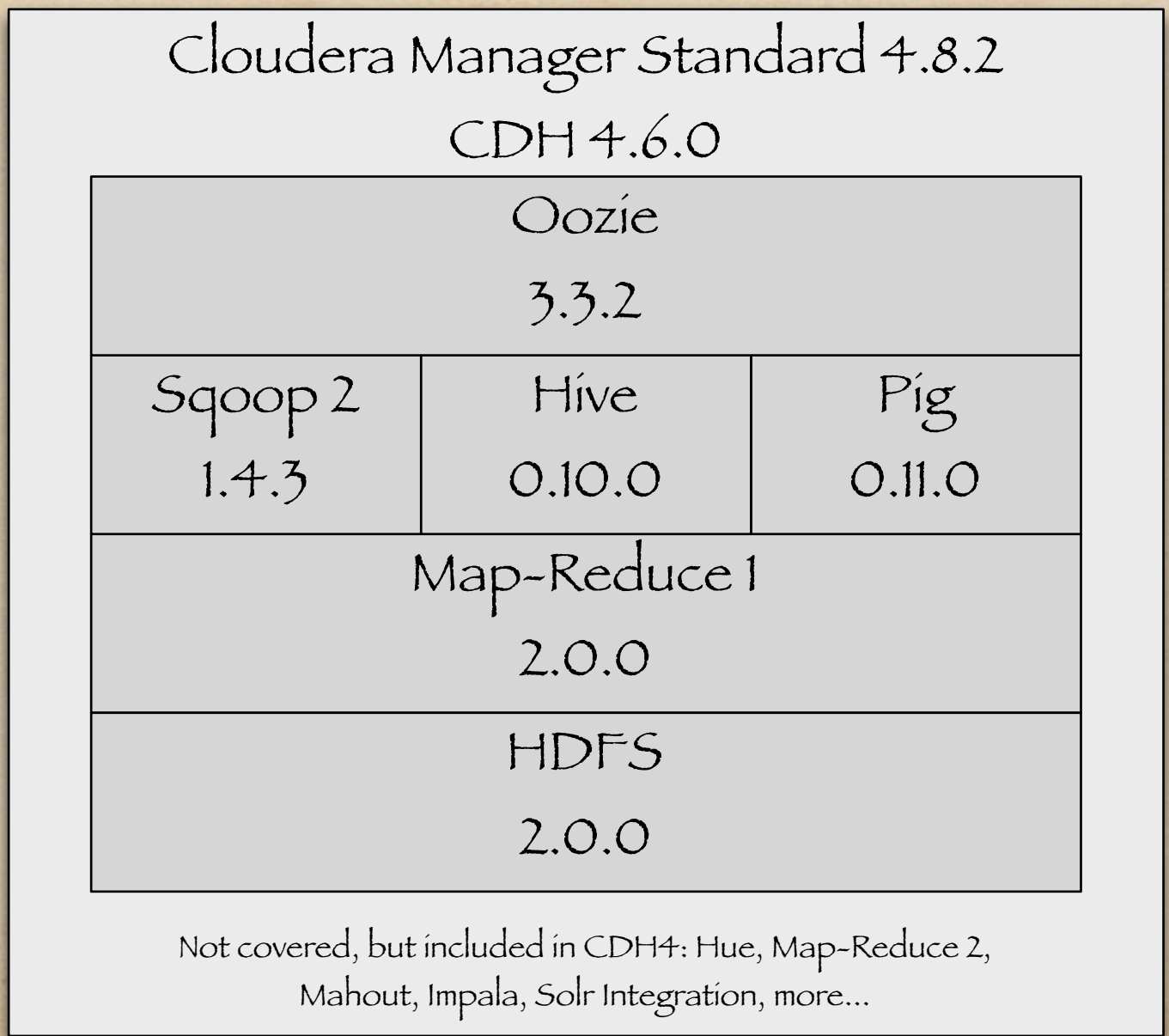
Pig & Hive

- It can be tricky to set up a problem as a pure map-reduce job
- Pig & Hive are tools that abstract the problem definition
  - Underneath, they transform work into a set of map-reduce jobs

Playing card example

- For an excellent explanation, checkout:
  - http://www.youtube.com/watch?v=bcjSe0xCHbE
  - One nit, not crazy about his use of the term magic
    - Makes something simple seem complex

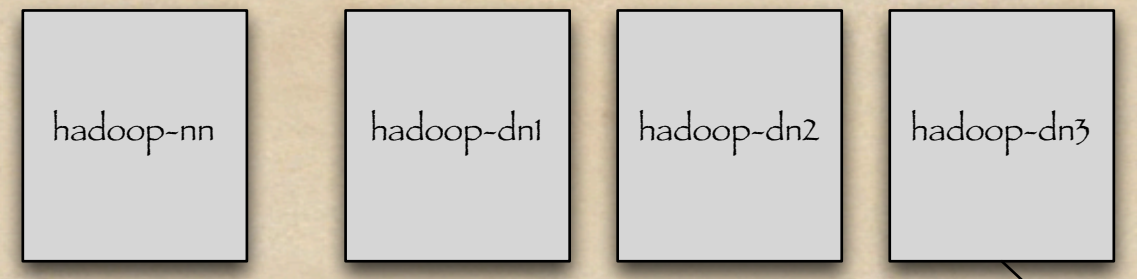
The Cloudera CDH4 Distro



The Cluster

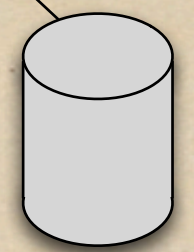
- Using Digital Ocean Cloud Hosting for test cluster
  - https://www.digitalocean.com/?refcode=12768ce5992e
    - Shameless plug: Use this link to sign up and I get credit
- Hadoop is resource hungry, particularly for RAM
  - Was unable to get install to work with < 4 GB
  - $40/month per instance and I have 4 (OUCH!)
  - So I only spin them up when I am testing
    - Currently, requires reinstall of Hadoop on spin up. Have it down to ~60 minutes.
    - Reinstall required because Hadoop stores IP addressed instead of host names (grrr!).

The Cluster (cont)



Name Node
Job Tracker
Hive MetaStore
Sqoop MetaStore
Oozie Server

Data Node
Task Tracker
Hive Gateway



Test DB
(postgreSQL

All Remote Nodes:

CentOS 6
Dual Core
4GB Ram
60GB SSD Disk
Hosted in New York
$0.06/hr

The Code

Tendril Home Energy Report

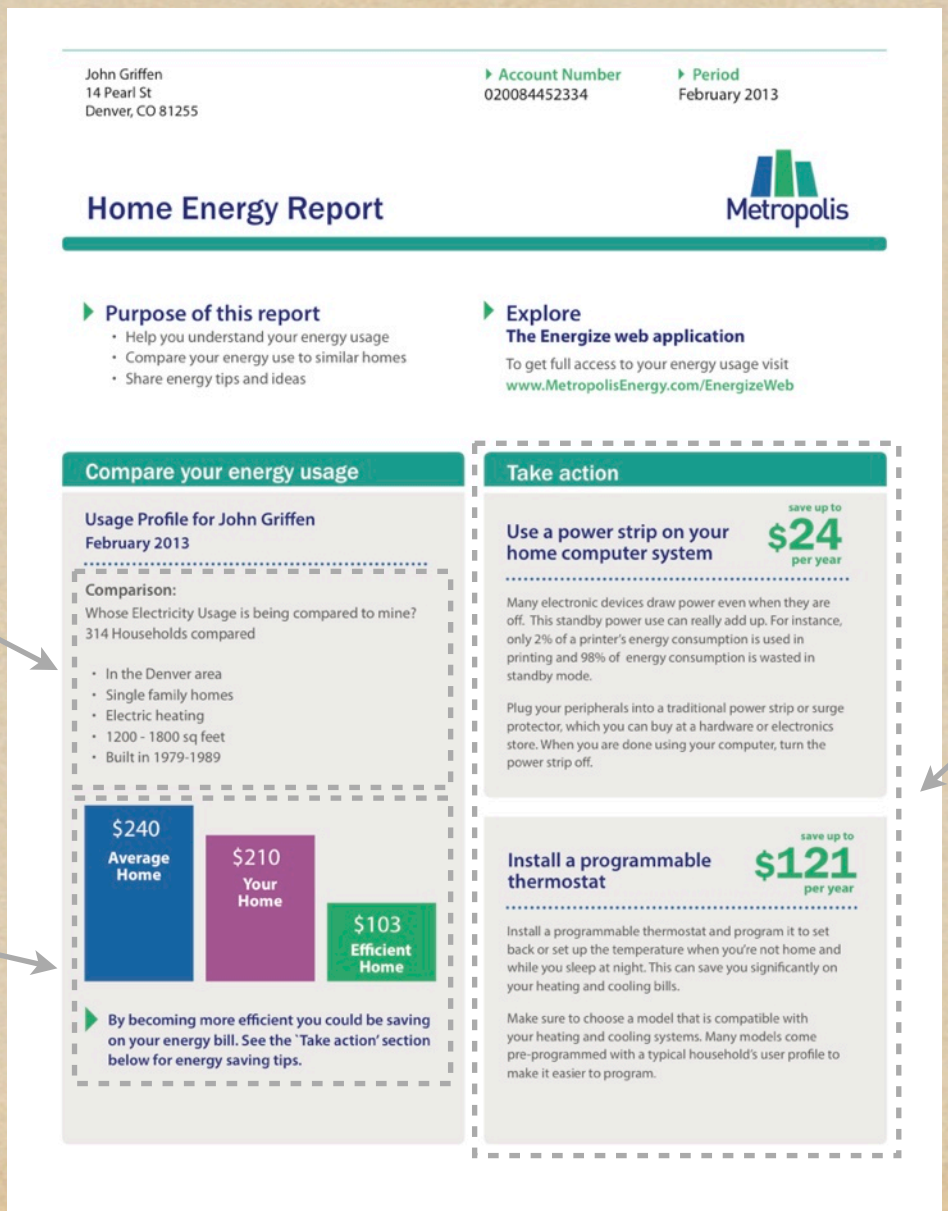
- A real world product produced with Hadoop

Home Energy Report (pg 1)

Compared to similar homes

Current month's usage comparison

Customized savings tips and tricks



John Griffen
14 Pearl St
Denver, CO 81255

Account Number
020084452334

Period
February 2013

Home Energy Report

Metropolis

Purpose of this report

- Help you understand your energy usage
- Compare your energy use to similar homes
- Share energy tips and ideas

Explore

The Energize web application

To get full access to your energy usage visit
www.MetropolisEnergy.com/EnergizeWeb

Compare your energy usage

Usage Profile for John Griffen
February 2013

Comparison:
Whose Electricity Usage is being compared to mine?
314 Households compared

- In the Denver area
- Single family homes
- Electric heating
- 1200 - 1800 sq feet
- Built in 1979-1989

By becoming more efficient you could be saving on your energy bill. See the 'Take action' section below for energy saving tips.

Take action

Use a power strip on your home computer system

save up to **$24** per year

Many electronic devices draw power even when they are off. This standby power use can really add up. For instance, only 2% of a printer's energy consumption is used in printing and 98% of energy consumption is wasted in standby mode.

Plug your peripherals into a traditional power strip or surge protector, which you can buy at a hardware or electronics store. When you are done using your computer, turn the power strip off.

Install a programmable thermostat

save up to **$121** per year

Install a programmable thermostat and program it to set back or set up the temperature when you're not home and while you sleep at night. This can save you significantly on your heating and cooling bills.

Make sure to choose a model that is compatible with your heating and cooling systems. Many models come pre-programmed with a typical household's user profile to make it easier to program.

Home Energy Report (pg 2)

Historical usage comparison

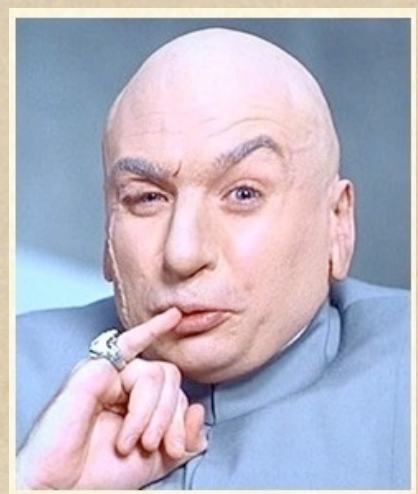
Customizable utility marketing messages
- Service notices
- Program announcements

Not Just about the Petabytes

- 1.7 million total accounts
- Eligibility, e.g., voluntary opt out, control group, and VIP
  - 500K "control" group accounts
- 13 months of billing data for most accounts
- 9 prioritized demographics sources
- Accounts clustered into "like" neighborhoods
  - Dynamic clustering use Mahout (Canopy & K-Means)
- Business Rules, e.g., no rentals
- Customized savings tips
  - Proprietary physics-based building model
- Customizable utility marketing messages
- Filters, e.g., not enough bills, small neighborhood, address hygiene
- End of run analytics

Resulting in...

- 1.2 Million reports per month (and growing)!



- M&V (Energy savings of treatment vs control groups) is ongoing
  - Initial value is redacted kWh per household

The "Contrived" Problem

- An energy monitoring company has the following data:
  - List of accounts, some customers have solar, some don't
  - Hourly residential solar production readings
  - Weather data for all customer locations
  - Sunrise/Sunset data for all customer locations
  - A list of "Do not call" accounts not to market to
    - These include solar and non-solar customers

The "Contrived" Problem (cont)

- Need to create daily sunrise-sunset graphs (png's?) for each solar customer
  - Should integrate weather data in the graphs
    - Temperature and cloud cover
  - Will be added to newsletters sent to the customers
- Need to create a report for marketing containing all accounts without solar who can be contacted as part of the current marketing campaign
- Other contrivey things...

How Do We Get Data Into the Cluster?

The HDFS CLI

- For some data types, this is the easiest way to import data
  - Typically used for static files that don't get updated a lot

The HDFS CLI (cont)

- Command format:
  - hadoop fs -<command> args
  - File specification: From network
    - Use hdfs://<namenode>:8020 prefix
      - e.g. hdfs://hadoop-nn:8020/tmp
  - From name node, prefix not needed
- Commands are mostly familiar linux file commands
  - ls, cat, mkdir, rm, cp, mv, etc.
- A few hdfs specific
  - put, getmerge, copyfromlocal, copytolocal, etc.

The HDFS CLI (cont)

- Some hdfs specifics
  - Files are stored in blocked chunks
    - Default is 128 MB
    - So not optimized for small files
  - Blocks for a given file not necessarily stored on the same data node
    - But will be replicated on more than one node
    - Name Node keeps track of what's where
  - hdfs won't overwrite a file
    - Pattern to use is delete, then write
    - Or write to new file, then delete old file and move
  - Directories sometimes act as files

A note on examples

- Will use the scriptRunner.sh tool from project (mostly)
  - Example slides will list the scripts to run
- Scripts are run on name-node (hadoop-nn)
  - For convenience, could run elsewhere
- Assumes cluster is configured and ssh keys have been configured to avoid passwords

HDFS Example

- We have some bulk files on our filesystem that we need to import
  - prepareForTalk.script (1)
  - bulkCopy.script (2)
  - Review /solar in HDFS browser
- We'll come back to some other HDFS commands later

Let's Sqoop some data!

- As cool as NOSQL is, there are still a "couple" of relational DB's out there
- Sqoop is the tool to import (and export) from SQL DB's
- To avoid typing passwords...
  - .pgpass in my home dir
  - /home/root/pg_password in HDFS

Sqoop Example 1

- Populate the DB with 3 accounts
  - populateWith3.script (3)
- Now sqoop data
  - simpleSqoop.script (4)
- Note that data is now in /solar/accounts
- Also note that sqoop has helpfully created a file, accounts.java, for your use, if you choose.

A Brief Digression

- Whats up with this file?
  - The data was processed by 3 mappers
  - Still one file, but written by three processes
- Let's use getmerge to view this file

Contents of directory /solar/accounts

Goto : /solar/accounts go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 B	3	128 MB	2013-09-09 00:37	rw-r--r--	root	supergroup
_logs	dir				2013-09-09 00:37	rwxr-xr-x	root	supergroup
part-m-00000	file	81 B	3	128 MB	2013-09-09 00:37	rw-r--r--	root	supergroup
part-m-00001	file	81 B	3	128 MB	2013-09-09 00:37	rw-r--r--	root	supergroup
part-m-00002	file	82 B	3	128 MB	2013-09-09 00:37	rw-r--r--	root	supergroup

Example - getMerge

- getMergeAccounts.script (5)
- Note that all "part" files have been concatenated (order is not guaranteed, don't count on it).
- Note the crc file. If you intend to modify this file and re-upload, make sure and delete it, or the upload won't work.

```
running: hadoop fs -getmerge hdfs://hadoop-nn:8020/solar/accounts accounts
13/09/08 18:50:03 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
running: ls -a .
.               ..              .accounts.crc   accounts
running: cat accounts
1,11-111-1,First1 Last1,1 111 Street,,City1,CO,80011,1.0,2013-09-08 18:30:00.592
2,22-222-2,First2 Last2,2 222 Street,,City2,CO,80012,2.0,2013-09-08 18:30:00.592
3,33-333-3,First3 Last3,3 333 Street,,City3,CO,80013,7.56,2013-09-08 18:30:00.592
```

Sqoop Example 2

- Populate the DB with 3 more accounts
  - populateWith3More.script (6)
- And sqoop data
  - simpleSqoop.script (7)
- Oops...
  - org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory /solar/accounts already exists

Let's Fix This

- Could delete file and sqoop all data each time
  - Should we sqoop 1.7 million accounts when 10K changed?
- Let's add some new options to our sqoop
  - --incremental append
  - --check-column last_updated
  - --last-value '2000-01-01'
- Requires relational table to have an appropriate column

Sqoop Example 3

- removeAccounts.script (8)
- populateWith3.script (9)
- And sqoop data
  - advancedSqoop.script (10)
- populateWith3More.script (11)
- Now edit script (yuk!), then
  - advancedSqoop.script (12)

Enter Sqoop Saved Job

- Job details are saved in sqoop metastore
  - By default, an HSQL DB
- After job runs, last value is auto-updated
- Had to make some changes to support Oozie
  - Start a server (on name node):
    - sqoop metastore --meta-connect jdbc:hsqldb:hsql://hadoop-nn:16000/sqoop &
  - Add argument to sqoop commands
    - --meta-connect jdbc:hsqldb:hsql://hadoop-nn:16000/
  - Not needed for command line use

Sqoop Example 4

- createSavedJob.script (13)
- removeAccounts.script (14)
- populateWith3.script (15)
- runSqoopJob.script (16)
- populateWith3More.script (17)
- runSqoopJob.script (18)

More on saved jobs

- sqoop job --list
- sqoop job --show sqoopAccounts
- sqoop job --delete sqoopAccounts

Now let's process the data

- Pig
  - A DSL for creating data processing pipelines
  - DSL is converted into series of map-reduce jobs
  - DSL is called pig latin
  - REPL-like shell called Grunt
  - Easy to add User Defined Functions (UDFs)
    - For our system, written in Java
    - Stored in the piggybank
  - Who says devs don't have a sense of humor?

Step 1 - Date Normalization

- Dates in the raw data are stored in different formats
- Later processing will be easier if we convert to a common format
- Will use a UDF for the actual conversion, pig will make sure processing is parallelized
  - This UDF is an evalFunc
  - Will use a filterFunc in next example
  - Many others, these seem to be the most common
- normalize.script (19)
  - While it is running, we'll look at the code

Step 2 - Aggregate

- Want to merge account info, solar, and weather
- Remove night time readings (a filterFunc UDF)
- Remove dupes (this was a reported defect)
- Calculate day of reading (yup, an evalFunc UDF)
- Group data by account and day of reading
- Sort by time of individual readings
- mergeAndFilter.script (20)
  - While it is running, we'll look at the code

Contrived Map-Reduce

- Group all solar readings
  - by account/date (day)/system size
  - This example is trivial, we already solved similar problem in Pig, but...
  - Long term, we want this data to be turned into a graph (probably a png) and stored in HDFS for other processes to use (stamped onto pdfs or emails).
  - M-R is a good tool for this (in conjunction with other libraries)

Map Reduce Example 1 a & b

- Run scripts/runBasicAggregator.sh
  - Note: run from ~, not scripts directory

```
3,33-333-3,20130701,7.56    ((3,33-333-3,7.56,20130701,1372669200000,3.626,17.8,0.5),(3,33-3
3,7.56,20130701,1372680000000,4.729,22.2,0.5),(3,33-333-3,7.56,20130701,1372683600000,3.4
333-3,7.56,20130701,1372665600000,2.548,17.2,0.5),(3,33-333-3,7.56,20130701,137269080000
(3,33-333-3,7.56,20130701,1372658400000,0.158,13.9,0.5),(3,33-333-3,7.56,20130701,1372701
3,7.56,20130701,1372708800000,0.035,21.7,0.9))
3,33-333-3,20130702,7.56    ((3,33-333-3,7.56,20130702,1372770000000,5.99,25.0,0.5),(3,33-33
3,7.56,20130702,1372752000000,2.521,16.6,0.0),(3,33-333-3,7.56,20130702,1372755600000,3.7
333-3,7.56,20130702,1372766400000,5.047,22.8,0.5),(3,33-333-3,7.56,20130702,137277360000
(3,33-333-3,7.56,20130702,1372784400000,2.685,27.2,0.5),(3,33-333-3,7.56,20130702,1372788
3,7.56,20130702,1372795200000,0.023,23.9,0.9))
```

```
3,33-333-3,20130701,7.56,(3,33-333-3,7.56,20130701,1372669200000,3.626,17.8,0.5),(3,33-333-3,7.56,
3,7.56,20130701,1372680000000,4.729,22.2,0.5),(3,33-333-3,7.56,20130701,1372683600000,3.45,23.3,0.
333-3,7.56,20130701,1372665600000,2.548,17.2,0.5),(3,33-333-3,7.56,20130701,1372690800000,4.534,
(3,33-333-3,7.56,20130701,1372658400000,0.158,13.9,0.5),(3,33-333-3,7.56,20130701,1372701600000,
3,7.56,20130701,1372708800000,0.035,21.7,0.9)
3,33-333-3,20130702,7.56,(3,33-333-3,7.56,20130702,1372770000000,5.99,25.0,0.5),(3,33-333-3,7.56,
3,7.56,20130702,1372752000000,2.521,16.6,0.0),(3,33-333-3,7.56,20130702,1372755600000,3.782,18.3,
333-3,7.56,20130702,1372766400000,5.047,22.8,0.5),(3,33-333-3,7.56,20130702,1372773600000,5.501,
(3,33-333-3,7.56,20130702,1372784400000,2.685,27.2,0.5),(3,33-333-3,7.56,20130702,1372788000000,
3,7.56,20130702,1372795200000,0.023,23.9,0.9)
```

Map Reduce Example 2

- Reformat the data, don't include the key
  - Mapper is same as previous example
  - Added a combiner, same code as reducer in previous example
  - New reducer
- Run scripts/runAdvancedAggregator.sh

```
3,33-333-3,20130701,7.56,(1372669200000,3.626,17.8,0.5),(1372672800000,4.474,19.4,0.5),(1372676400000,5.829
(1372662000000,0.693,16.1,0.5),(1372665600000,2.548,17.2,0.5),(1372690800000,4.534,21.7,0.9),(1372694400000,
(1372705200000,0.274,22.2,0.5),(1372708800000,0.035,21.7,0.9)
3,33-333-3,20130702,7.56,(1372770000000,5.99,25.0,0.5),(1372744800000,0.141,13.4,0.0),(1372748400000,0.669,
(1372762800000,5.349,22.2,0.5),(1372766400000,5.047,22.8,0.5),(1372773600000,5.501,26.1,0.5),(1372777200000,
(1372791600000,0.265,25.0,0.5),(1372795200000,0.023,23.9,0.9)
```

Map Reduce Example 3

- Save each reading in separate file (long term, this will be the png, not a csv)
- Demonstrates custom output format class
- Mapper and Combiner are still the same, new Reducer
- Run scripts/runMultiOutput.sh
  - Results stored in hierarchical structure
  - /solar/daily_multi/account/date
    - See example /solar/daily_multi/3/20130701

```
3,33-333-3,20130701,7.56    3,33-333-3,20130701,7.56,(1372669200000,3.626,17.8,0.5),(1372672800000
(1372687200000,4.095,23.9,0.5),(1372662000000,0.693,16.1,0.5),(1372665600000,2.548,17.2,0.5),(137269(
(1372701600000,1.125,22.8,0.9),(1372705200000,0.274,22.2,0.5),(1372708800000,0.035,21.7,0.9)
```

Hive

- A SQL-like tool for bulk data query
  - Often referred to as data warehousing
- Most things you can do in Pig can be done in Hive and vice versa
  - Hive is designed for SQL devs (I'm not one)

Hive Example 1

- Simple example
  - Find which accounts are solar and which are not
- getSolarAndNonSolarAccounts.script (21)

Hive Example 2

- Need a list of all non-solar customers who are not on do not call list
  - Would like to do the following (but can't):
    - SELECT accounts.id
      - FROM accounts
      - LEFT JOIN do_not_call ON accounts.id <> do_not_call.id
      - WHERE accounts.capacity == 0;
  - Hive's syntax is "almost" SQL
    - Biggest incompatibility is that only equi-joins are supported (will likely be fixed in a future release)
    - There is a hacky work around using an outer join (which can be expensive!)
- getAccountsToContact.script (22)

Hive Example 2 (cont)

- Why is the outer join expense?
  - Creates more intermediate rows that we need, which then have to be filtered

Account ID	DNC ID
4	NULL
5	5
6	NULL
7	7
8	NULL
9	NULL

Oozie example

- Will take a while to run, so will start, then discuss
- Workflow bundle must be deployed on HDFS
  - TODO: incorporate into gradle script
  - deployLibs.script (only when jar dependencies change)
  - deployWorkflow.script - when app changes

Oozie example (cont.)

- removeAccounts.script (23)
- populateWith3.script (24)
- populateWith3More.script (25)
- oozie.script (26)
- Review progress in Oozie Console

Oozie - Putting it all together

- Oozie is Hadoop's workflow scheduler
- Oozie Workflow jobs are Directed Acyclical Graphs (DAGs) of actions.
- Oozie Coordinator jobs are recurrent Oozie Workflow jobs triggered by time (frequency) and data availability.
- Oozie is integrated in Hadoop stack supporting several types of jobs out of the box (map-reduce, Pig, Hive, Sqoop, java, etc.)

But...

- It puts the hell in XML-hell!
- Anti-DRY
- The web-based tool is kind of lame
  - LOTS of clicking to find what you are after
  - Somewhat non-intuitive interface

workflow.xml

- Describes your processing pipeline
- Each job (pig, MR, hive, java, etc.) is enclosed in an action block
  - Prepare - setup steps
  - Configure - define properties
  - OK - next step after success
  - Error - next step after a failure

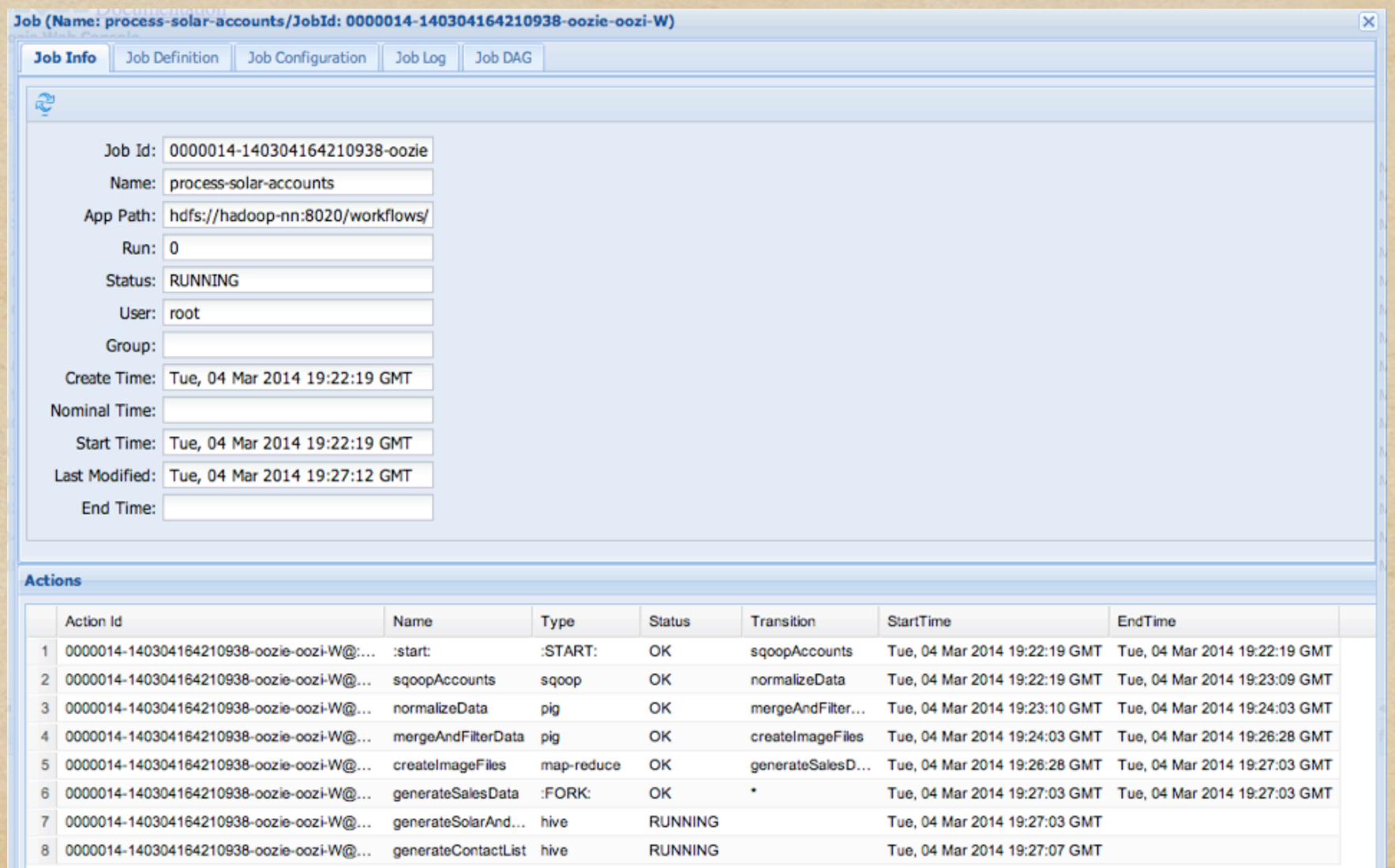
Fork/Join

- Easier than scripting the same
- Allows independent jobs to run in parallel
- Makes optimal use of your cluster resources

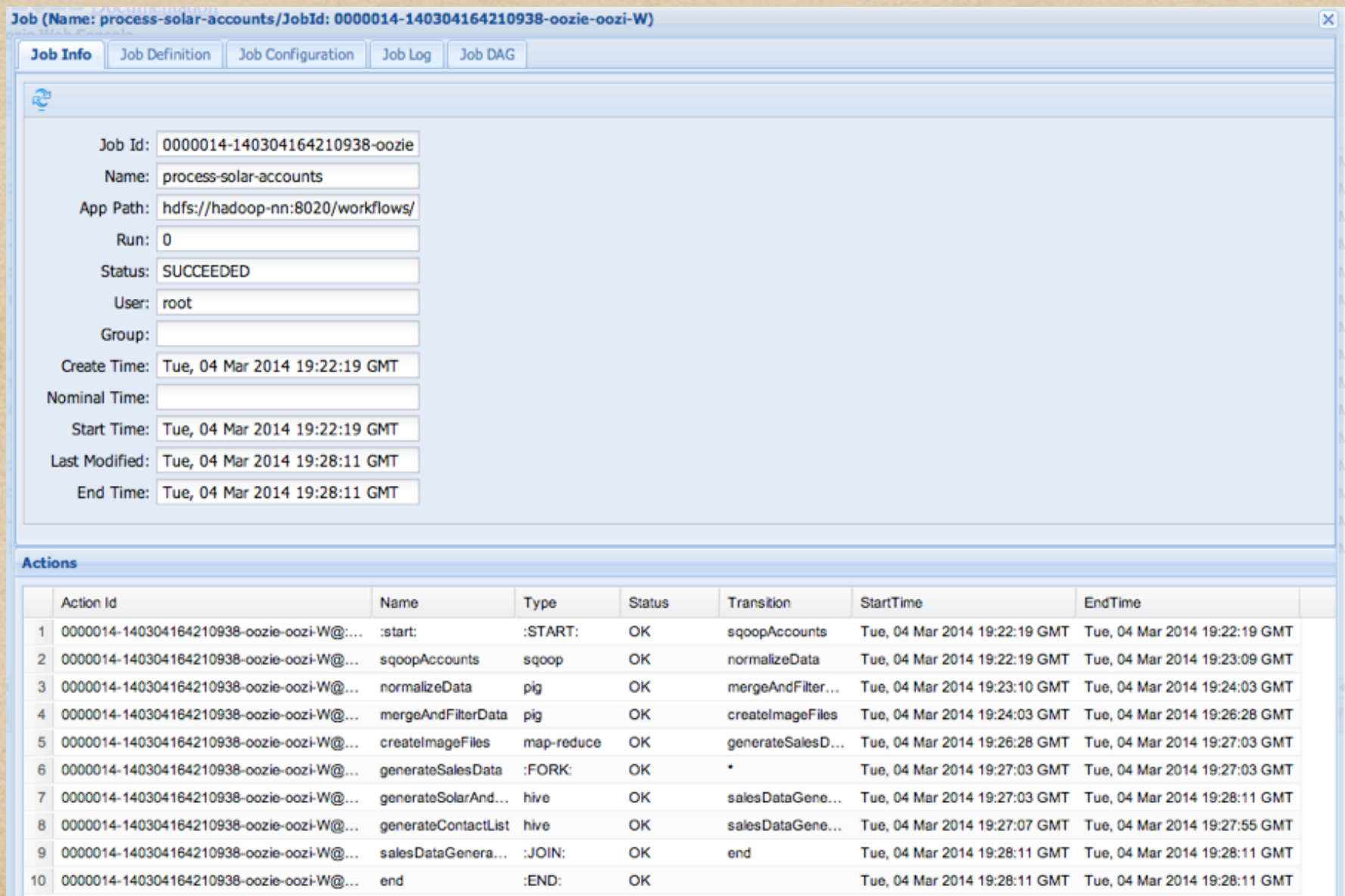
Sqoop workaround

- Evidently, when using a password file, there is a known issue in Sqoop where it closes filesystem prematurely
  - Workaround
    - Set fs.hdfs.impl.disable.cache to true
  - http://stackoverflow.com/questions/19735672

Running in parallel



A successful run...



	Action Id	Name	Type	Status	Transition	StartTime	EndTime
1	0000014-140304164210938-oozie-oozi-W@:...	:start:	:START:	OK	sqoopAccounts	Tue, 04 Mar 2014 19:22:19 GMT	Tue, 04 Mar 2014 19:22:19 GMT
2	0000014-140304164210938-oozie-oozi-W@...	sqoopAccounts	sqoop	OK	normalizeData	Tue, 04 Mar 2014 19:22:19 GMT	Tue, 04 Mar 2014 19:23:09 GMT
3	0000014-140304164210938-oozie-oozi-W@...	normalizeData	pig	OK	mergeAndFilter...	Tue, 04 Mar 2014 19:23:10 GMT	Tue, 04 Mar 2014 19:24:03 GMT
4	0000014-140304164210938-oozie-oozi-W@...	mergeAndFilterData	pig	OK	createImageFiles	Tue, 04 Mar 2014 19:24:03 GMT	Tue, 04 Mar 2014 19:26:28 GMT
5	0000014-140304164210938-oozie-oozi-W@...	createImageFiles	map-reduce	OK	generateSalesD...	Tue, 04 Mar 2014 19:26:28 GMT	Tue, 04 Mar 2014 19:27:03 GMT
6	0000014-140304164210938-oozie-oozi-W@...	generateSalesData	:FORK:	OK	*	Tue, 04 Mar 2014 19:27:03 GMT	Tue, 04 Mar 2014 19:27:03 GMT
7	0000014-140304164210938-oozie-oozi-W@...	generateSolarAnd...	hive	OK	salesDataGene...	Tue, 04 Mar 2014 19:27:03 GMT	Tue, 04 Mar 2014 19:28:11 GMT
8	0000014-140304164210938-oozie-oozi-W@...	generateContactList	hive	OK	salesDataGene...	Tue, 04 Mar 2014 19:27:07 GMT	Tue, 04 Mar 2014 19:27:55 GMT
9	0000014-140304164210938-oozie-oozi-W@...	salesDataGenera...	:JOIN:	OK	end	Tue, 04 Mar 2014 19:28:11 GMT	Tue, 04 Mar 2014 19:28:11 GMT
10	0000014-140304164210938-oozie-oozi-W@...	end	:END:	OK		Tue, 04 Mar 2014 19:28:11 GMT	Tue, 04 Mar 2014 19:28:11 GMT

References

- This project, https://github.com/jmctee/hadoopTools
- Hadoop: The Definitive Guide, 3rd Edition - Tom White
  - http://shop.oreilly.com/product/0636920021773.do
- Programming Pig - Alan Gates
  - http://shop.oreilly.com/product/0636920018087.do
- Programming Hive - Edward Capriolo, Dean Wampler, & Jason Rutherglen
  - http://shop.oreilly.com/product/0636920023555.do
- Apache Sqoop Cookbook - By Kathleen Ting & Jarek Jarcec Cecho
  - http://shop.oreilly.com/product/0636920029519.do
- Oozie
  - http://oozie.apache.org/docs/3.3.2/
- Cloudera CDH4 Documentation
  - http://www.cloudera.com/content/support/en/documentation/cdh4-documentation/cdh4-documentation-v4-latest.html
- CDH Users Forum
  - https://groups.google.com/a/cloudera.org/forum/#!forum/cdh-user

Thank You! (and go solar!)