Sahale, The Good Parts

Agenda

1. Eli Reisman
2. Basics
3. Oozie for Adhoc Jobs
4. Walkthrough
5. Improvements
6. Live Debugging

Eli Reisman

1. https://codeascraft.com/2015/02/11/sahale-visualizing-cascading-workflows-at-etsy/
2. Apache Committer
3. Labor of Love (spent many months on this project)
4. No photo, so quote: “delayed trolling is not good trolling. timing is everything you have step up your game”
5. Project is open-sourced: https://github.com/etsy/Sahale

Basics

1. Custom UI for tracking jobs because default UI sucks
2. Good for:
    a. Debugging failed/completed jobs
    b. Optimizing Jobs using historical runs

Oozie for adhoc jobs preamble

1. run_scalding creates an adhoc workflow
2. First check in oozie UI for failures
3. Sahale link from oozie and vice verse

Walkthrough

1. Elephants
    a. data transferred (kB could mean trouble)
    b. Isolated elephants. Spit graphs.
2. Time taken at each step (watch for very long steps)
3. Stats top-level info: tuples read and written
4. Taps: identify sources and sinks (60% science 40% art?)
5. Input and output sources

Walkthrough

1. Counters:
   a. More detailed stats than Stats tab
   b. You can add your own custom counters for debugging
   c. You can use GraphiteCountersJob; Hadoop counters sunk to graphite.
   d. Tricky bit is figuring out which step counter attaches to.
2. Links: application master
3. Mappers and Reducers: too many mappers are tricky on a busy cluster
4. Sahale to oozie link; first place to look for errors

Room for improvement

1. UI needs some designer love
    a. Disappearing Elephants!?
    b. Discoverability Unintuitive
2. Maintainer? Improvements?
3. Custom counters displayed in central location
4. Ability to override auto-refreshes

Live Debugging

1. Failed Job
    a. Client logs
    b. Application Master
2. Job Optimization
    a. Application Master
    b. Look at average durations of reducers/ hot keys