Quadrupling your Elephants

RDF and the Hadoop Ecosystem

Rob Vesse
Twitter: @RobVesse
Email: rvesse@apache.org

About Me

- **Software Engineer at YarcData (part of Cray Inc)**
  - Working on big data analytics products
- **Active open source contributor primarily to RDF & SPARQL related projects**
  - Apache Jena Committer and PMC Member
  - dotNetRDF Lead Developer
- **Primarily interested in RDF, SPARQL and Big Data Analytics technologies**

Overview

- What's missing in the Hadoop ecosystem?
- What's needed to fill the gap?
- What's already available?
  - Jena Hadoop RDF Tools
  - GraphBuilder
  - Other Projects
- Getting Involved
- Questions

What's missing in the Hadoop ecosystem?

Not a Lot!

POWERED BY APACHE
accumulo
AVRO
Cassandra
chukwa
FLUME
APACHE GIRAPH
GORA
APACHE HBASE
HIVE
KNOX
mahout
MESOS
Apache MRUnit
nutch
OOZIE
Spark
TAJO

YarcData
A DIVISION OF CRAY INC.

Apache, the projects and their logo shown here are registered trademarks or trademarks of The Apache Software Foundation in the U.S. and/or other countries

Where's RDF?

- **No first class projects**
- **Some limited support in other projects**
  - E.g. Giraph supports RDF by bridging through the Tinkerpop stack
- **Some existing external projects**
  - Lots of academic proof of concepts
  - Some open source efforts but mostly task specific
  - E.g. Infovore targeted at creating curated Freebase and DBPedia datasets

What's needed to fill the gap?

Writable Types

- Need to efficiently represent RDF concepts as Writable types
- Nodes, Triples, Quads, Graphs, Datasets, Query Results etc
- What's the minimum viable subset?

Input and Output Formats

- Need to be able to get data in and out of RDF formats
- Without this we can't use the power of the Hadoop ecosystem to do useful work
- Lots of serializations out there:
  - RDF/XML
  - Turtle
  - NTriples
  - NQuads
  - JSON-LD
  - etc
- Also would like to be able to produce end results as RDF

Translation to RDF

- **Map/Reduce building blocks**
  - Common operations e.g. splitting
  - Enable developers to focus on their applications
- **User Friendly tooling**
  - i.e. non-programmer tools

What's already available?

Apache Jena - RDF Tools for Hadoop

- **Set of modules part of the Apache Jena project**
  - Originally developed at Cray and donated to the project earlier this year
- **Experimental modules on the hadoop-rdf branch of our**
- **Currently only available as development SNAPSHOT releases**
  - **Group ID:** org.apache.jena
  - **Artifact IDs:**
    - jena-hadoop-rdf-common
    - jena-hadoop-rdf-io
    - jena-hadoop-rdf-mapreduce
  - **Latest Version:** 0.9.0-SNAPSHOT
- **Aims to fulfill all the basic requirements for enabling RDF on Hadoop**
- **Built against Hadoop Map/Reduce 2.x APIs**

Common API

- **Provides the Writable types for RDF primitives**
  - NodeWritable
  - TripleWritable
  - QuadWritable
  - NodeTupleWritable
- **All backed by RDF Thrift**
  - A compact binary serialization for RDF using Apache Thrift
  - See http://afs.github.io/rdf-thrift/
  - Extremely efficient to serialize and deserialize
  - Allows for efficient WritableComparator implementations that perform binary comparisons

IO API

- Provides InputFormat and OutputFormat implementations
- Supports most formats that Jena supports
  - Designed to be extendable with new formats
- Will split and parallelize inputs where the RDF serialization is amenable to this
- Also transparently handles compressed inputs and outputs
  - Note that compression blocks splitting
  - i.e. trade off between IO and parallelism

Map/Reduce API

- Various reusable building block Mapper and Reducer implementations:
  - Counting
  - Filtering
  - Grouping
  - Splitting
  - Transforming
- Can be used as-is to do some basic Hadoop tasks or used as building blocks for more complex tasks

Node Count Demo

Node Count Demo - Performance Notes

- For NTriples inputs compared performance of a Text based node count versus RDF based node count
- Performance as good (within 10%) and sometimes significantly better
  - Heavily dataset dependent
  - Varies considerably with cluster setup
  - Also depends on how the input is processed
  - YMMV!
- For other RDF formats you would struggle to implement this at all

Intel Graph Builder

- **Originally developed by Intel**
  - Some contributions by Cray - awaiting merging at time of writing
- **Open source under Apache License**
  - https://github.com/01org/graphbuilder/tree/2.0.alpha
  - 2.0.alpha is the Pig based branch
- **Allows data to be transformed into graphs using Pig scripts**
  - Provides set of Pig UDFs for translating data to graph formats
  - Supports both property graphs and RDF graphs

Intel Graph Builder - Pig Script Example

```
-- Declare our mappings
x = FOREACH propertyGraph GENERATE (*,
  [ 'idBase' # 'http://example.org/instances/',
    'base' # 'http://example.org/ontology/',
    'namespaces' # [ 'foaf' # 'http://xmlns.com/foaf/0.1/' ],
    'propertyMap' # [ 'type' # 'a',
                      'name' # 'foaf:name',
                      'age' # 'foaf:age' ],
    'idProperty' # 'id' ]);

-- Convert to NTriples
rdf_triples = FOREACH propertyGraphWithMappings GENERATE FLATTEN(RDF(*));

-- Write out NTriples
STORE rdf_triples INTO '/tmp/rdf_triples' USING PigStorage();
```

Mappings

- Uses a declarative mapping based on Pig primitives
    - Maps and Tuples
- Have to be explicitly joined to the data because Pig UDFs can only be called with String arguments
    - Has some benefits e.g. conditional mappings
- RDF Mappings operate on Property Graphs
    - Requires original data to be mapped to a property graph first
    - Direct mapping to RDF is a future enhancement that has yet to be implemented

Graph Builder Demo

Other Projects

- **Infovore - Paul Houle**
  - https://github.com/paulhoule/infovore/wiki
  - Cleaned and curated Freebase datasets processed with Hadoop
- **CumulusRDF - Institute of Applied Informatics and Formal Description Methods**
  - https://code.google.com/p/cumulusrdf/
  - RDF store backed by Apache Cassandra

Getting Involved

- **Please start playing with these projects**
- **Please interact with the community:**
  - dev@jena.apache.org
  - What works?
  - What is broken?
  - What is missing?
- **Contribute**
  - Apache projects are ultimately driven by the community
  - If there's a feature you want please suggest it
  - Or better still contribute it yourself!

Questions?

Personal Email: rvesse@apache.org
Jena Mailing List: dev@jena.apache.org

Demo Scripts

RDF Stats Demo

```
> bin/hadoop jar jena-hadoop-rdf-stats-0.9.0-SNAPSHOT-hadoop-job.jar
org.apache.jena.hadoop.rdf.stats.RdfStats --node-count --output /user/
output --input-type triples /user/input
```

- **--node-count requests the Node Count statistics be calculated**
- **Assumes mixed quads and triples input if no --input-type specified**
  - Using this for triples only data can skew statistics
  - e.g. can result in high node counts for default graph node
  - Hence we explicitly specify input as triples

Demo Code

Node Count Demo - Mapper

```
public abstract class AbstractNodeTupleNodeCountMapper<TKey, TValue, T extends AbstractNodeTupleWritable<TValue>>
extends Mapper<TKey, T, NodeWritable, LongWritable> {
    private LongWritable initialCount = new LongWritable(1);

    @Override
    protected void map(TKey key, T value, Context context) throws IOException, InterruptedException {
        NodeWritable[] ns = this.getNodes(value);
        for (NodeWritable n : ns) {
            context.write(n, this.initialCount);
        }
    }

    protected abstract NodeWritable[] getNodes(T tuple);
}

public class TripleNodeCountMapper<TKey> extends AbstractNodeTupleNodeCountMapper<TKey, Triple, TripleWritable> {
    @Override
    protected NodeWritable[] getNodes(TripleWritable tuple) {
        Triple t = tuple.get();
        return new NodeWritable[] { new NodeWritable(t.getSubject()), new NodeWritable(t.getPredicate()),
                new NodeWritable(t.getObject()) };
    }
}
```

Node Count Demo - Reducer

```
public class NodeCountReducer extends Reducer<NodeWritable, LongWritable, NodeWritable,
LongWritable> {

    @Override
    protected void reduce(NodeWritable key, Iterable<LongWritable> values, Context context)
throws IOException, InterruptedException {
        long count = 0;
        Iterator<LongWritable> iter = values.iterator();
        while (iter.hasNext()) {
            count += iter.next().get();
        }
        context.write(key, new LongWritable(count));
    }
}
```

Node Count Demo - Job Configuration

```
Job job = Job.getInstance(config);
job.setJarByClass(JobFactory.class);
job.setJobName("RDF Triples Node Usage Count");

// Map/Reduce classes
job.setMapperClass(TripleNodeCountMapper.class);
job.setMapOutputKeyClass(NodeWritable.class);
job.setMapOutputValueClass(LongWritable.class);
job.setReducerClass(NodeCountReducer.class);

// Input and Output
job.setInputFormatClass(TriplesInputFormat.class);
job.setOutputFormatClass(NTriplesNodeOutputFormat.class);
FileInputFormat.setInputPaths(job, StringUtils.arrayToString(inputPaths));
FileOutputFormat.setOutputPath(job, new Path(outputPath));

return job;
```