Hortonworks

Hadoop: Beyond MapReduce

Steve Loughran, Hortonworks
stevel@hortonworks.com
@steveloughran

Big Data workshop, June 2013

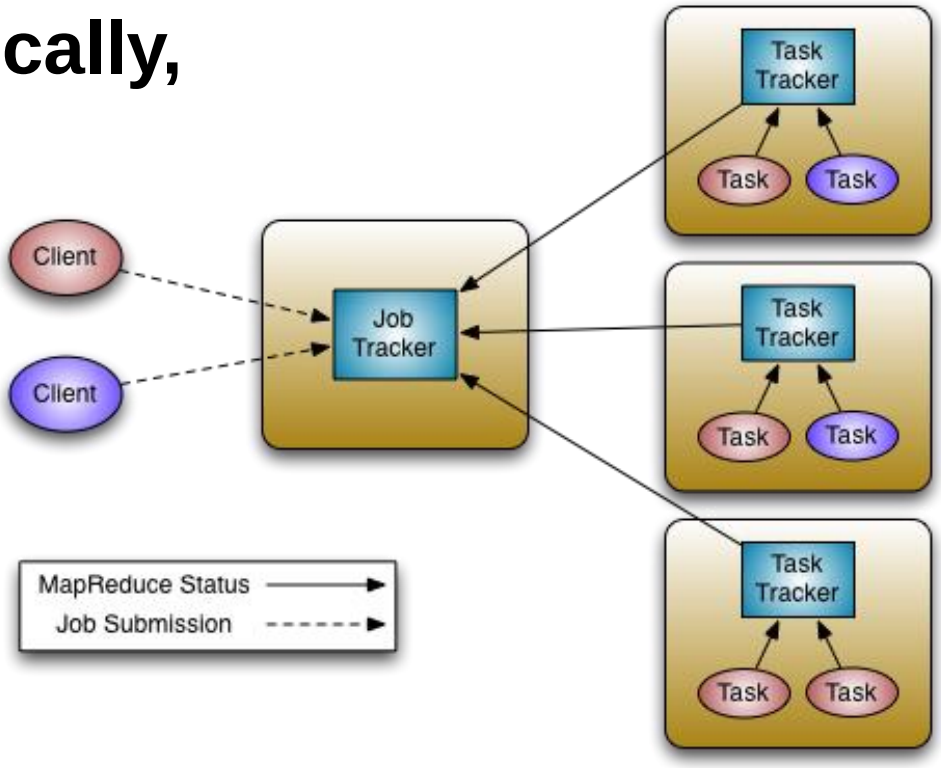
Hadoop MapReduce

1. Map: events ➔ <k,v>* pairs
2. Reduce: <k,[v_1, v_2,.. v_n]> ➔ <k,v'>

- **Map trivially parallelisable on blocks in a file**
- **Reduce parallelise on keys**
- **MapReduce engine can execute Map and Reduce sequences against data**
- **HDFS provides data location for work placement**

MapReduce democratised big data

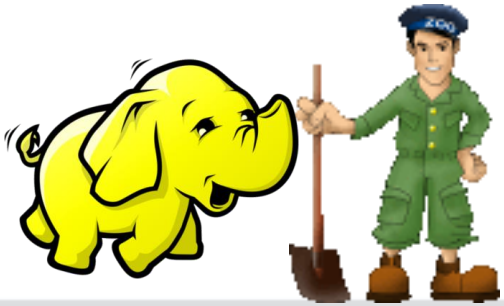
- **Conceptual model easy to grasp**
- **Can write and test locally, superlinear scaleup**
- **Tools and stack**



You don't need to understand parallel coding to run apps across 1000 machines

The stack is key to use

Kafka

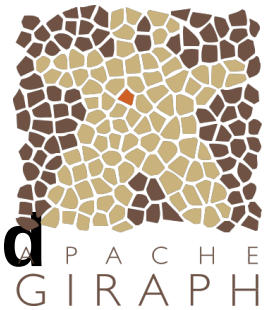


Example: Pig

```
generated = LOAD '$src/$srcfile'
    USING PigStorage(',' , '-noschema')
    AS (line: int, gaussian: double, b:
boolean, c:chararray );
sorted = ORDER generated BY c ASC;
result = FILTER sorted BY gaussian >= 0;
```

Example: Apache Giraph

- **Graph nodes in RAM**
- **exchange data with peers at barriers**
- **use cases: PageRank, Friend-of-Friend**



- **But also: modelling cells in a heart**

Bulk-Synchronous-Parallel -read Pregel paper

Hortonworks

But there is a lot more we can do

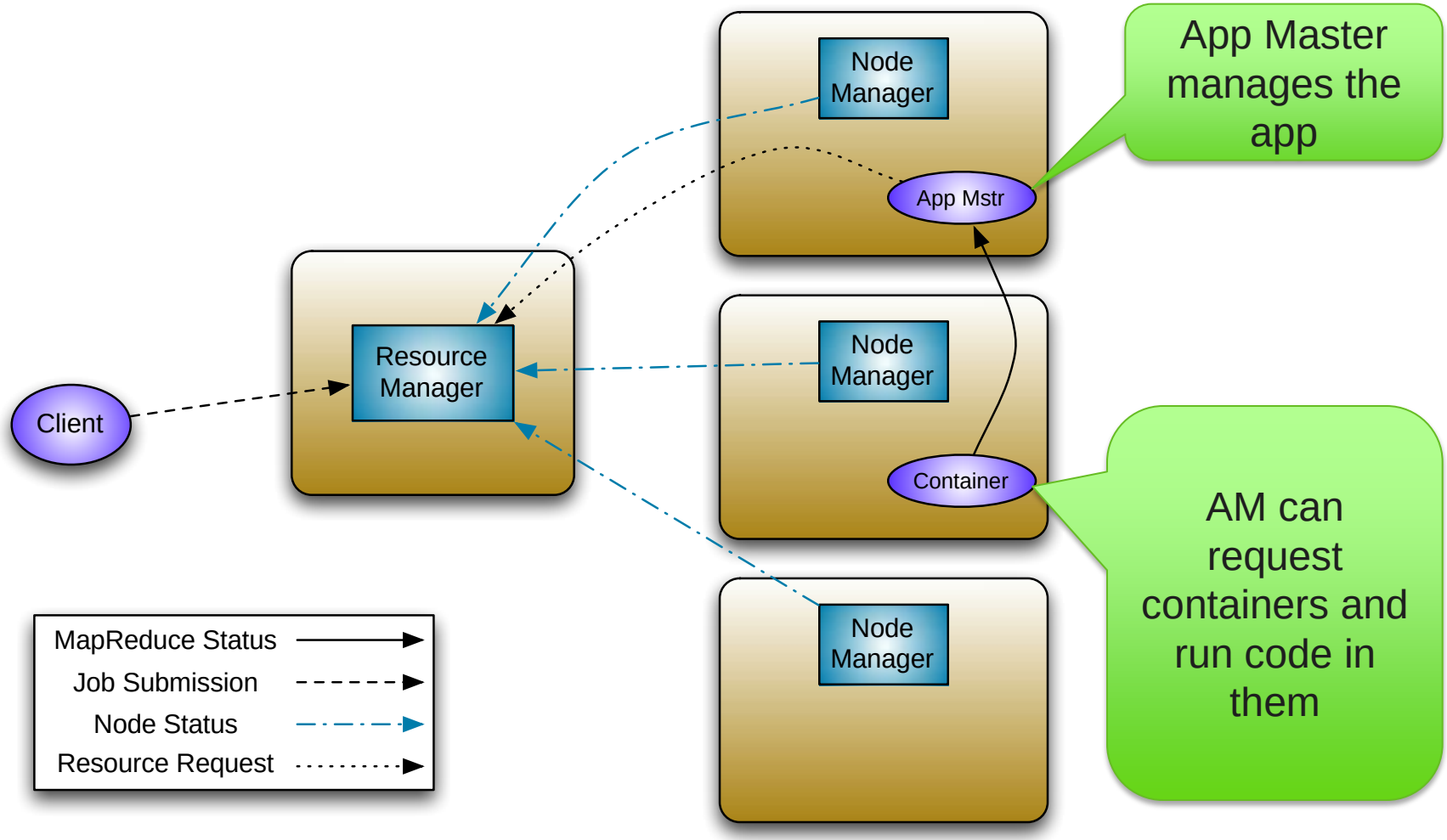
New Algorithms and runtimes

- **Giraph for graph work**
- **Stream processing: Storm**
- **Iterative and chained processing: Dryad-style**
- **Long-lived processes**

Production-side issues

- **Scale to 10K nodes**
- **Eliminate SPOFs & Bottlenecks**
- **Improve versioning by moving MR engine user-side**
- **Avoid having dedicated servers for other roles**

YARN: Yet Another Resource Negotiator

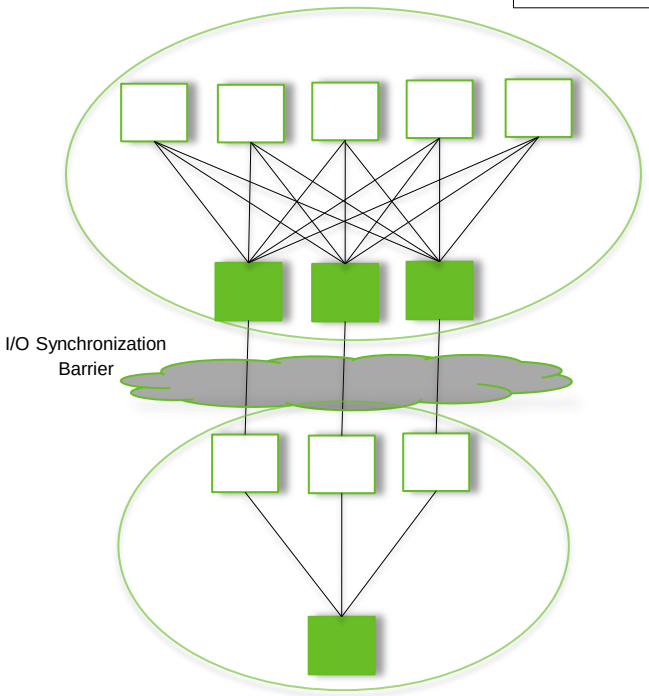


YARN vs Other Resource Negotiators

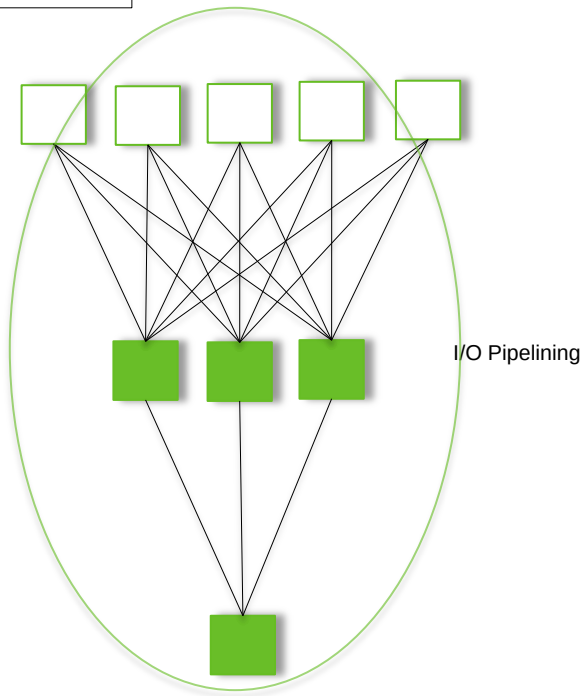
- **MapReduce #1 initial use case**
- ***Failures:***
  AM handles worker failures, YARN handles AM failures
- ***Scheduling Locality:*** **sources of data, destinations. AM gets provides location requests along with (CPU, RAM**

Pig/Hive-MR versus Pig/Hive-Tez

```
SELECT a.state, COUNT(*)
FROM a JOIN b ON (a.id = b.id)
GROUP BY a.state
```

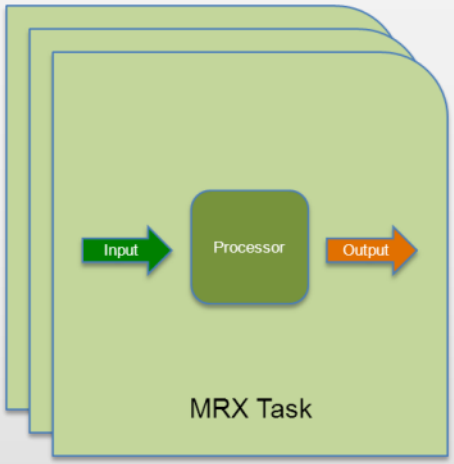


Pig/Hive - Tez



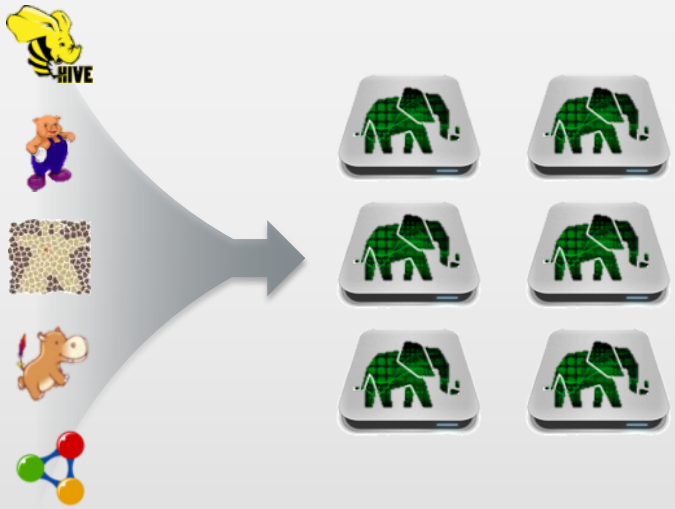
Pig/Hive - Tez

FastQuery: Beyond Batch with YARN



Tez Generalizes Map-Reduce

Simplified execution plans process data more efficiently



Always-On Tez Service

Low latency processing for all Hadoop data processing

Hortonworks

You too can write a distributed execution framework -if you need to

Start the work in progress

- **Hamster: MPI**
- **Storm-YARN from Yahoo!**
- **Hoya: HBase on YARN ← me**

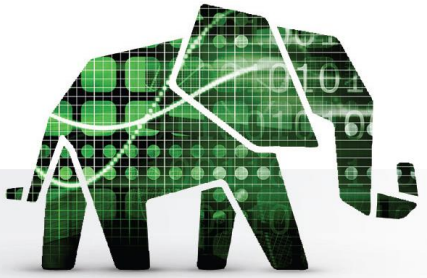
And start with other people's code

- **Continuuity Weave -looks best place to start**

Hortonworks

What are the services and algorithms we are going to need?

P.S: we are hiring



http://hortonworks.com/careers/