The other Apache Technologies your Big Data solution needs!

Nick Burch

Alfresco

The Apache Software Foundation

- Apache Technologies as in the ASF
- 91 Top Level Projects
- 59 Incubating Projects (74 past ones)
- Y is the only letter we lack
- C and S are favourites, at 10 projects
- Meritocratic, Community driven Open Source

What we're not covering

Projects not being covered

- Cassandra
- CouchDB
- Hadoop
- HBase
- Lucene and SOLR
- Mahout
- Nutch

What we are looking at

Talk Structure

- Loading and querying Big Data
- Building your MapReduce Jobs
- Deploying and Building for the Cloud
- Servers for Big Data
- Building out your solution

- Many projects – only an overview!

Loading and Querying

Pig – pig.apache.org

- Originally from Yahoo, entered the Incubator in 2007, graduated 2008
- Provides an easy way to query data, which is compiled into Hadoop M/R
- Typically 1/20th of the lines of code, and 1/15th of the development time
- Optimising compiler – often only slightly slower, occasionally faster!

Pig – pig.apache.org

- Shell, scripting and embedded Java
- Local mode for development
- Built-ins for loading, filtering, joining, processing, sorting and saving
- User Defined Functions too
- Similar range of operations as SQL, but quicker and easier to learn
- Allows non coders to easily query

Pig – pig.apache.org

```
$ pig -x local
grunt>

grunt> A = LOAD 'student' USING PigStorage() AS (name:chararray, age:int,
gpa:float);
grunt> B = FOREACH A GENERATE name;
grunt> DUMP B;
(John)
(Mary)
(Bill)
(Joe)

grunt> C = LOAD 'votertab10k' AS (name: chararray, age: int, registration:
chararray, donation: float);
grunt> D = COGROUP A BY name, C BY name;
grunt> E = FOREACH D GENERATE FLATTEN((IsEmpty(A) ? null : A)),
FLATTEN((IsEmpty(C) ? null : C));

grunt> DUMP E;
(John, 21, 2.1, ABCDE, 21.1)
(Mary, 19, 3.4, null, null)
(Bill, 21, 2.4, ABCDE, 0.0)
(Joe, 22, 4.9, null, null)

grunt> DESCRIBE A;
A: {name: chararray,age: int,gpa: float}
```

Hive – hive.apache.org

- Data Warehouse tool on Hadoop
- Originally from Facebook, Netflix now a big user (amongst many others!)
- Query with HiveQL, a SQL like language that runs map/reduce query
- You can drop in your own mappers and reducers for custom bits too

Hive – hive.apache.org

- Define table structure
- Optionally load your data in, either from Local, S3 or HDFS
- Control internal format if needed
- Query (from table or raw data)
- Query can Group, Join, Filter etc

Hive – hive.apache.org

```
add jar ../build/contrib/hive_contrib.jar;
CREATE TABLE apachelog (
  host STRING,
  identity STRING,
  user STRING,
  time STRING,
  request STRING,
  status STRING,
  size STRING,
  referer STRING,
  agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  "input.regex" = "([^ ]*) ([^ ]*) ([^ ]*) (-|\\[[^\\]]*\\])
([^ \"]*|\"[^\"]*\") (-|[0-9]*) (-|[0-9]*)(?: ([^ \"]*|\".*\")
([^ \"]*|\".*\"))?",
  "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s"
)
STORED AS TEXTFILE;

SELECT COUNT(*) FROM apachelog;

SELECT agent, COUNT(*) FROM apachelog
WHERE status = 200 AND time > '2011-01-01' GROUP BY agent;
```

Gora (Incubating)

- ORM Framework for Column Stores
- Grew out of the Nutch project
- Supports HBase and Cassandra
- Hypertable, Redis etc planned
- Data is stored using Avro (more later)
- Query with Pig, Lucene, Hive, Hadoop Map/Reduce, or native Store code

Gora (Incubating)

- Example: Web Server Log

```
{
  "type": "record",
  "name": "Pageview",
  "namespace": "org.apache.gora.tutorial.log.generated",
  "fields" : [
    {"name": "url", "type": "string"},
    {"name": "timestamp", "type": "long"},
    {"name": "ip", "type": "string"},
    {"name": "httpMethod", "type": "string"},
    {"name": "httpStatusCode", "type": "int"},
    {"name": "responseSize", "type": "int"},
    {"name": "referrer", "type": "string"},
    {"name": "userAgent", "type": "string"}
  ]
}
```

- Avro data bean, JSON

Gora (Incubating)

```
// ID is a long, Pageview is compiled Avro bean
dataStore = DataStoreFactory.getDataStore(Long.class, Pageview.class);

// Parse the log file, and store
while(going) {
   Pageview page = parseLine(reader.readLine());
   dataStore.put(logFileId, page);
}
DataStore.close();


private Pageview parseLine(String line) throws ParseException {
    StringTokenizer matcher = new StringTokenizer(line);
    //parse the log line
    String ip = matcher.nextToken();
    ...

    //construct and return pageview object
    Pageview pageview = new Pageview();
    pageview.setIp(new Utf8(ip));
    pageview.setTimestamp(timestamp);
    ...

    return pageview;
}
```

Accumulo (Entering Incubator)

- Distributed Key/Value store, built on top of Hadoop, Zookeeper and Thrift
- Inspired by BigTable, with some improvements to the design
- Cell level permissioning (access labels) and server side hooks to tweak data as it's read/written
- Just entered the Incubator, still getting set up there.
- Initial work mostly done by the NSA!

Giraph (Incubating)

- Graph processing platform built on top of Hadoop
- Bulk-Synchronous parallel model
- Verticies send messages to each other, process messages, send next
- Uses ZooKeeper for co-ordination and fault tolerance
- Similar to things like Pregal

Sqoop (Incubating)

- Bulk data transfer tool
- Hadoop (HDFS), HBase and Hive on one side
- SQL Databases on the other
- Can be used to import data into your big data cluster
- Or, export the results of a big data job out to your data wharehouse

Chukwa (Incubating)

- Log collection and analysis framework based on Hadoop
- Incubating since 2010
- Collects and aggregates logs from many different machines
- Stores data in HDFS, in chunks that are both HDFS and Hadoop friendly
- Lets you dump, query and analyze

Chukwa (Incubating)

- Chukwa agent runs on source nodes
- Collects from Log4j, Syslog, plain text log files etc
- Agent sends to a Collector on the Hadoop cluster
- Collector can transform if needed
- Data written to HDFS, and optionally to HBase (needed for visualiser)

Chukwa (Incubating)

- Map/Reduce and Pig query the HDFS files, and/or the HBase store
- Can do M/R anomaly detection
- Can integrate with Hive
- eg Netflix collect weblogs with Chukwa, transform with Thrift, and store in HDFS ready for Hive queries

Flume (Incubating)

- Another Log collection framework
- Concentrates on rapidly getting data to a variety of sources
- Typically write to HDFS + Hive + FTS
- Joint Agent+Collector model
- Data and Control planes independent
- More OOTB, less scope to alter

Building MapReduce Jobs

Avro – avro.apache.org

- Language neutral data serialization
- Rich data structures (JSON based)
- Compact and fast binary data format
- Code generation optional for dynamic languages
- Supports RPC
- Data includes schema details

Avro – avro.apache.org

- Schema is always present – allows dynamic typing and smaller sizes
- Java, C, C++, C#, Python, Ruby, PHP
- Different languages can transparently talk to each other, and make RPC calls to each other
- Often faster than Thrift and ProtoBuf
- No streaming support though

Thrift – thrift.apache.org

- Java, C++, Python, PHP, Ruby, Erland, Perl, Haskell, C#, JS and more
- From Facebook, at Apache since 2008
- Rich data structure, compiled down into suitable code
- RPC support too
- Streaming is available
- Worth reading the White Paper!

HCatalog (Incubating)

- Provides a table like structure on top of HDFS files, with friendly addressing
- Allows Pig, Hadoop MR jobs etc to easily read/write data structured data
- Simpler, lighter weight than Avro or Thrift based serialisation
- Based on Hive's metastore format
- Doesn't require an additional datastore

MRUnit (Incubating)

- New to the Incubator, started in 2009
- Built on top of JUnit
- Checks Map, Reduce, then combined
- Provides test drivers for hadoop
- Avoids you needing lots of boiler plate code to start/stop hadoop
- Avoids brittle mock objects

MRUnit (Incubating)

- IdentityMapper – same input/output

```
public class TestExample extends TestCase {
  private Mapper mapper;
  private MapDriver driver;

  @Before
  public void setUp() {
    mapper = new IdentityMapper();
    driver = new MapDriver(mapper);
  }

  @Test
  public void testIdentityMapper() {
    // Pass in { “foo”, “bar” }, ensure it comes back again
    driver.withInput(new Text("foo"), new Text("bar"))
            .withOutput(new Text("foo"), new Text("bar"))
            .runTest();
    assertEquals(1, driver.getCounters().findCounter(“foo”,”bar”));
  }
}
```

Oozie (Incubating)

- Workflow, scheduler and dependency manager for Hadoop jobs (inc Pig etc)
- Define a workflow to describe the data flow for your desired output
- Oozie handles running dependencies as needed, and scheduled execution of steps as requested
- Builds up a data pipe, then executes as required on a cloud scale

BigTop (Incubating)

- Build, Package and Test code built on top of Hadoop related projects
- Allows you to check that a given mix of say HDFS, Hadoop Core and ZooKeeper work well together
- Then package a tested bundle
- Integration testing of your stack
- Test upgrades, generate packages

Ambari (Incubating)

- Monitoring, Admin and LifeCycle management for Hadoop clusters
- eg HBase, HDFS, Hive, Pig, ZooKeeper
- Deploy+Configure stack to a cluster of machines
- Update software stack versions
- Monitoring and Service Admin
- REST APIs for cluster management

stdc++ / APR

- Cross platform C and C++ libraries
- stdc++ delivers portable, consistent algorithms, containers, iterators, thread safe implementations etc
- APR delivers predictable (if sometimes OS specific) code for reading and writing files, sockets, strings, tables, hashes etc in pure C

For the Cloud

Provider Independent Cloud APIs

- Lets you provision, manage and query Cloud services, without vendor lock-in
- Translates general calls to the specific (often proprietary) ones for a given cloud provider
- Work with remote and local cloud providers (almost) transparently

Provider Independent Cloud APIs

- Create, stop, start, reboot and destroy instances
- Control what's run on new instances
- List active instances
- Fetch available and active profiles
- EC2, Eycalyptos, Rackspace, RHEV, vSphere, Linode, OpenStack

LibCloud – libcloud.apache.org

- Python library (limited Java support)
- Very wide range of providers
- Script your cloud services

```
from libcloud.compute.types import Provider
from libcloud.compute.providers import get_driver

EC2_ACCESS_ID = 'your access id'
EC2_SECRET_KEY = 'your secret key'

Driver = get_driver(Provider.EC2)
conn = Driver(EC2_ACCESS_ID, EC2_SECRET_KEY)

nodes = conn.list_nodes()
# [<Node: uuid=..., state=3, public_ip=['1.1.1.1'], provider=EC2
...>, ...]
```

DeltaCloud (Incubating)

- REST API (xml) + web portal
- Bigger providers only, so far!

```
<instances>
  <instance href="http://fancycloudprovider.com/api/instances/inst1" id='inst1'>
    <owner_id>larry</owner_id>
    <name>Production JBoss Instance</name>
    <image href="http://fancycloudprovider.com/api/images/img3"/>
    <hardware_profile href="http://fancycloudprovider.com/api/hardware_profiles/m1-small"/>
    <realm href="http://fancycloudprovider.com/api/realms/us"/>
    <state>RUNNING</state>
    <actions>
      <link rel="reboot" href="http://fancycloudprovider.com/api/instances/inst1/reboot"/>
      <link rel="stop" href="http://fancycloudprovider.com/api/instances/inst1/stop"/>
    </actions>
    <public_addresses>
      <address>inst1.larry.fancycloudprovider.com</address>
    </public_addresses>
    <private_addresses>
      <address>inst1.larry.internal</address>
    </private_addresses>
  </instance>
</instances>
```

Whirr - whirr.apache.org

- Grew out of Hadoop
- Aimed at running Hadoop, Cassandra, HBase and ZooKeeper
- Higher level – running services not running machines
- Java (jclouds) and Python versions
- Can be run on the command line

Airavata (Incubating)

- Toolkit to build, manage, execute and monitor large scale applications
- Workflow driven processing
- Historically aimed at Scientific Processing, but now expanding
- Facilities for developer to deploy
- End user launches, workflow manages, then able to monitor eg via widgets

VCL (Incubating)

- Virtual Computing Lab
- Provision and broker a compute environment, on bare machines, virtual machines or spare machines
- Background and Interactive uses
- Web interface to request & provision
- Private clouds through to HPC setups

Serving Big Data

TrafficServer – trafficserver.apache.org

- Caching proxy web server
- Inktomi → Yahoo → Apache
- Fast and scalable – 150,000 requests per second possible on an i7-920!
- Yahoo served 400TB/day off 150 commodity servers running TS in 2009
- “Highway to the Cloud”
- Serve static, and proxy+cache dynamic

ZooKeeper – zookeeper.apache.org

- Centralised service for configuration, naming and synchronisation
- Provides Consensus, Group Mangement and Presence tracking
- Single co-ordination service across all the different components
- ZooKeeper is distributed and highly reliable (avoid config being SPOF)

ZooKeeper – zookeeper.apache.org

- “A central nervous system for distributed applications and services”
- Bindings for Java, C, Perl, Python, Scala, .Net (C#), Node.js, Erlang
- Applications can read nodes, send events, and watch for them
- eg fetch config, come up, perform leader election, share active list

Kitty (Incubating)

- Lightweight command line JMX client
- Not just Tomcat, now all Java Apps
- Query, discover and change JMX
- JVM has JMX properties
- All Hadoop parts expose information
- Memory, threads, jobs, capacity etc
- Must have for SysOps!

Building out your Solution

UIMA – uima.apache.org

- Unstructured Information analysis
- Lets you build a tool to extract information from unstructured data
- Components in C++ and Java
- Network enabled – can spread work out across a cluster
- Helped IBM to win Jeopardy!

Tika – tika.apache.org

- Text and Metadata extraction
- Identify file type, language, encoding
- Extracts text as structured XHTML
- Consistent Metadata across formats
- Java library, CLI and Network Server
- SOLR integration
- Handles format differences for you

OpenNLP (Incubating)

- Natural Language Processing
- Various tools for sentence detection, tokenization, tagging, chunking, entity detection etc
- UIMA likely to be better if you want a whole-solution
- OpenNLP good when integrating NLP into your own solution

MINA – mina.apache.org

- Framework for writing scalable, high performance network apps in Java
- TCP and UDP, Client and Server
- Build non blocking, event driven networking code in Java
- MINA also provides pure Java SSH, XMPP, Web and FTP servers

Deft (Incubating)

- High performance non-blocking webserver / webapp server, written in pure Java
- Backed by NIO, very similar design to that given by MINA
- Very quick to get started with for hosting non-blocking web application
- MINA is better if you need full control

ActiveMQ / Qpid / Synapse

- Messaging, Queueing and Brokerage solutions across most languages
- Decide on your chosen message format, endpoint languages and messaging needs, one of these three will likely fit your needs!
- Queues, Message Brokers, Enterprise Service Buses, high performance and yet also buzzword compliant!

Kafka (Incubating)

- Distributed push-subscribe (pub-sub) messaging system
- Allows high throughput on one system, partitionable for many
- More general than log based systems such as Flume
- Supports persisted messages, clients can catch up
- Distributed, remote sources and sinks

S4 (Incubating)

- Simple Scalable Streaming System
- Platform for building tools to work on continuous, unbounded data streams
- Stream is broken into events, which are routed to PEs and processed
- Uses Actors model, highly concurrent
- PEs are Java based, but event sources and sinks can be in any language

Logging – logging.apache.org

- Java, C++, .Net and PHP
- Configurable logging levels, formats, output sinks etc
- Fits nicely with Chukwa – have your Java log4j logging collated and stored into HDFS, or locally logged in dev
- Well known, easy to use framework

Commons – commons.apache.org

- Collection of libraries for Java projects
- Some historic, many still useful!

Attributes, BeanUtils, Betwixt, Chain, CLI, Codec, Collections, Compress, Configuration, Daemon, DHCP, DbUtils, Digester, Discovery, EL, Email, Exec, FileUpload, IO, JCI, Jelly, Jexl, JXPath, Lang, Launcher, Logging, Math, Modeler, Net, Pool, Primitivies, Proxy, Sanselan, SXML, Transaction, Validator, VFS

BeanValidation, CLI2, Convert, CSV, Digester3, Finder, Flatfile, Functor, I18N, Id, Javaflow, Jnet, Monitoring, Nabla, OpenPGP, Performance, Pipeline, Runtime

Directory – directory.apache.org

- Pure Java LDAP solutions
- If you've loads of machines, you need to be using something like LDAP!
- ApacheDS server worth considering if your SysAdmins prefer Java
- Directory Studio is an Eclipse RCP App for managing and querying LDAP
- Cross platform LDAP administration!

JMeter – jakarta.apache.org/jmeter/

- Load testing tool
- Performance test network services
- Define a series of tasks, execute them in parallel
- Talks to web, SOAP, LDAP, JMS, JDBC
- Handy for checking how external resources will hold up when a big data system starts heavily using them!

Chemistry – chemistry.apache.org

- Java, Python, .Net and PHP interface to Content Management Systems
- Implements the OASIS CMIS spec
- Browse, read and write data in your content repositories
- Rich information and structure
- Supported by Alfresco, Microsoft, SAP, Adobe, EMC, OpenText and more

ManifoldCF (Connectors) (Incubating)

- Framework for content (mostly text) extraction from content repositories
- Aimed at indexing solutions, eg SOLR
- Connectors for reading and writing
- Simpler than Chemistry, but also works for CIFS, file systems, RSS etc
- Extract from SharePoint, FileNet, Documentum, LiveLink etc

OpenOffice (Incubating)

- You'll probably need to read, write and share some documents while building your solution
- Apache licensed way to do that!
- Our first big “Consumer Focused” project
- Needs new contributors too, if anyone wants to get involved :)

Questions?

Thanks!

- Twitter - @Gagravarr
- Email – nick.burch@alfresco.com

- The Apache Software Foundation: http://www.apache.org/
- Apache projects list: http://projects.apache.org/