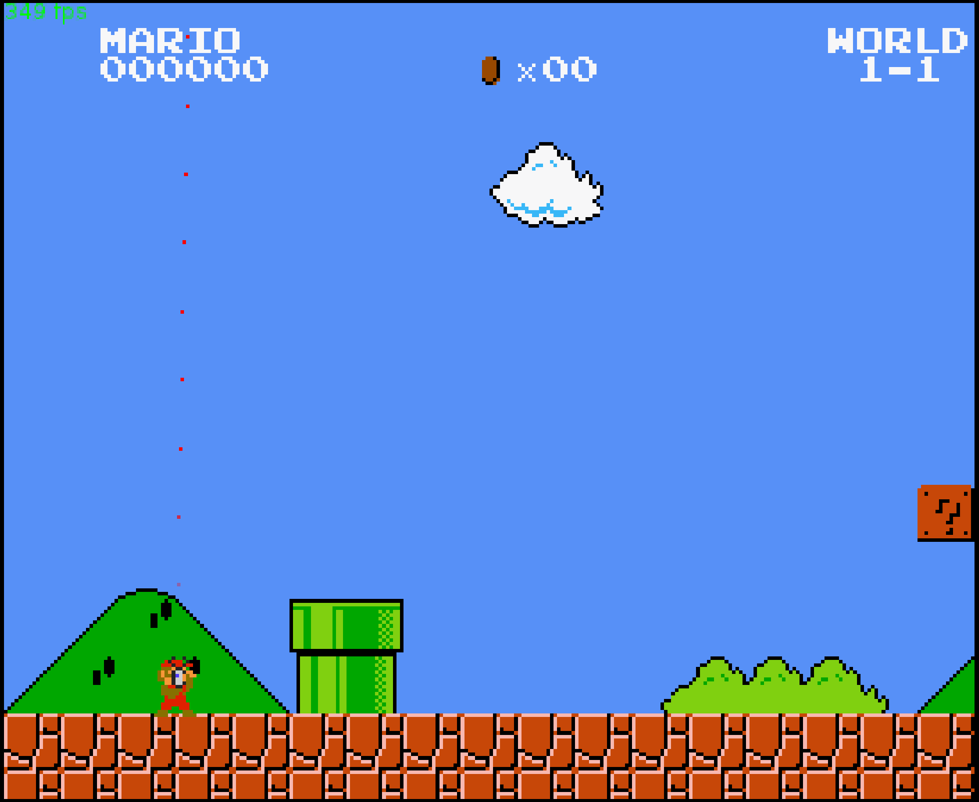
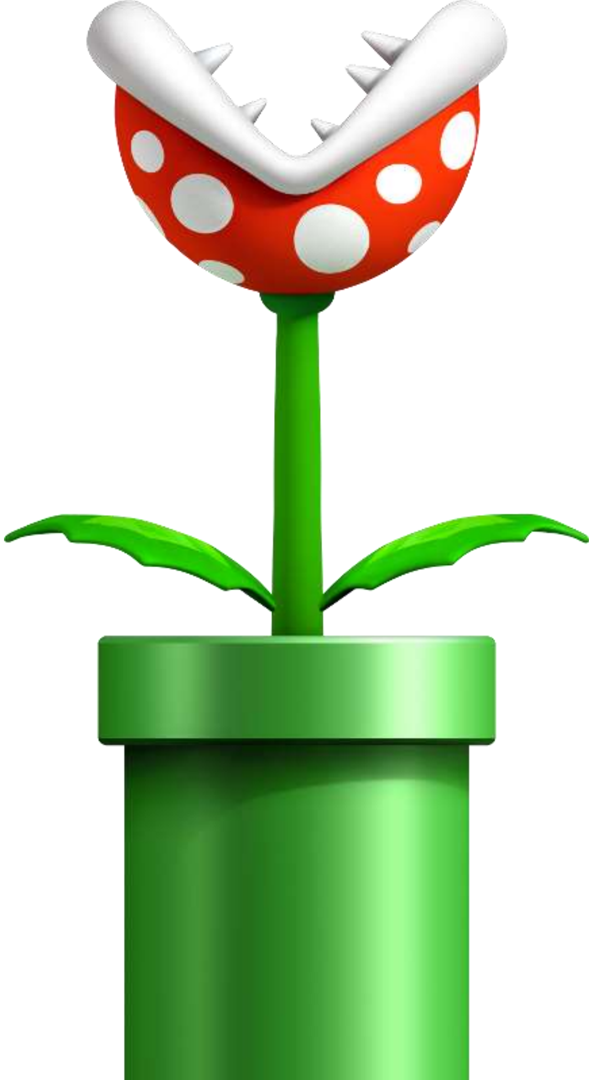
Building Data Pipelines



Jonathan Holloway @jph98

Pipeline Concepts



What I’ll Cover Tonight...

Basic intro and history of pipelines

Some examples of pipelines

An overview of big data pipelines

Some AWS technologies for building pipelines

History of pipelines

Invented by Douglas McIlroy

Pipes added to UNIX in 1973

```
ps -ef | grep java
```

Processes chained together by their standard streams

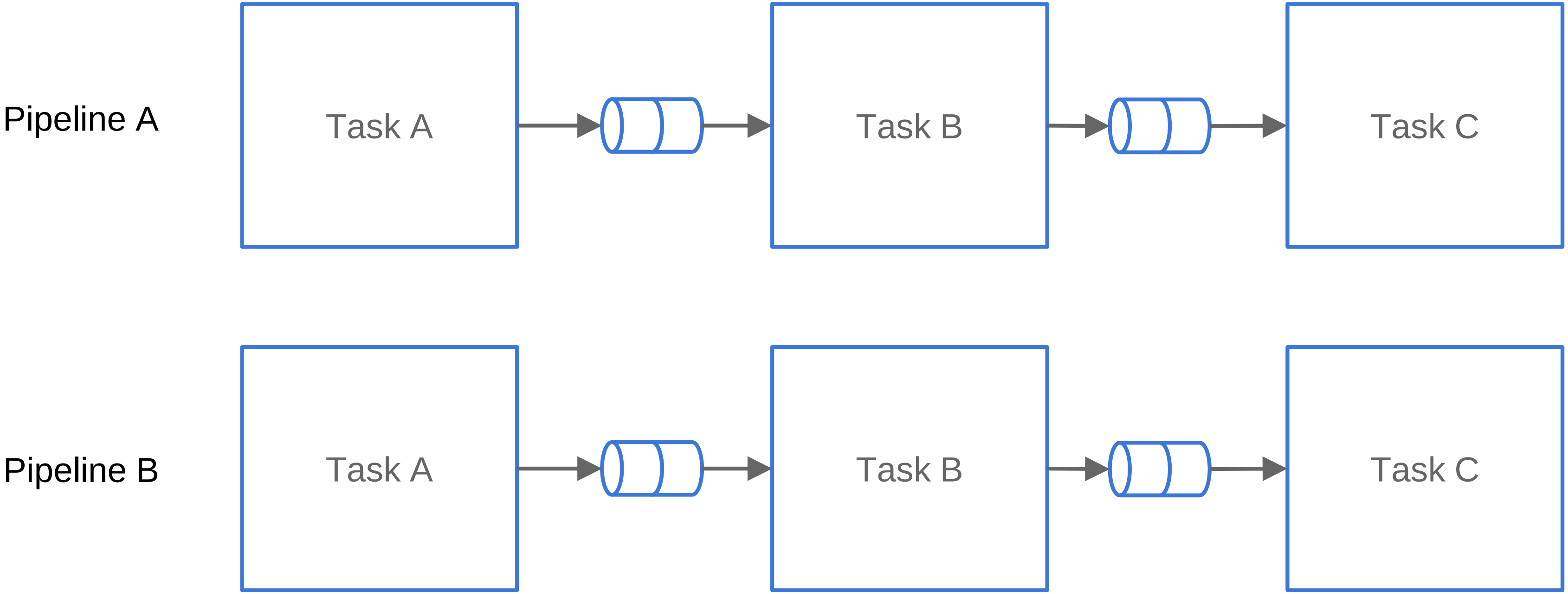
Pipes and Filters Architecture

Pipeline concept brought into software development mainstream

First used this for a message extractor, analyser and indexing solution circa 2009

Enterprise integration patterns went further

Pipes and Filters Architecture



Why? What do we achieve?

Decoupling of tasks

Encapsulation of processing within a task

Reuse of tasks in different workflows possibly

Some Considerations

How do we specify a task?

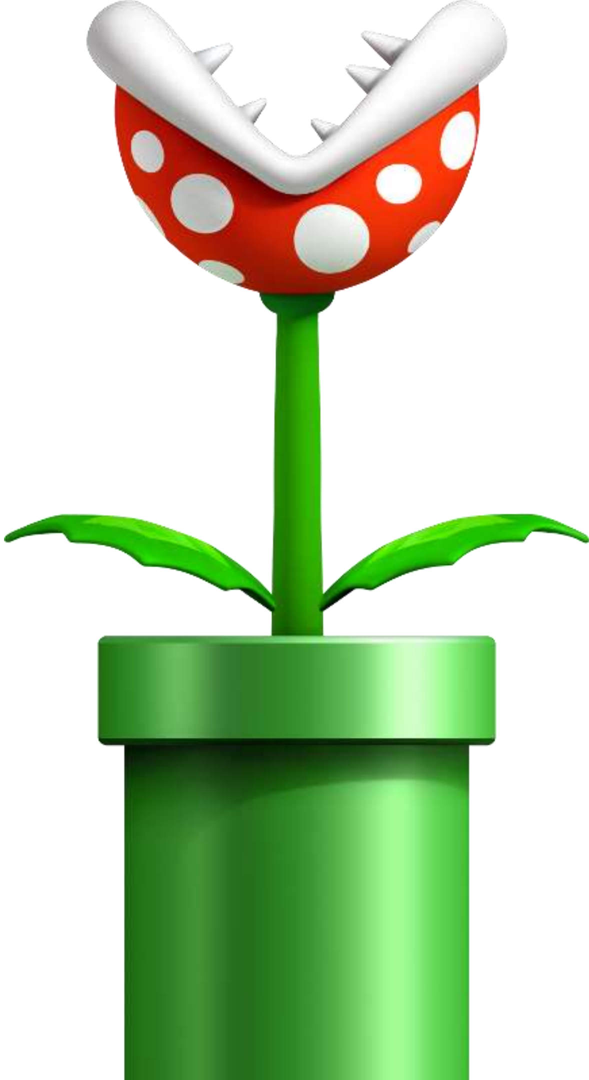
How do we feed data between the tasks?

When do they run, how often?

Do they run in serial or parallel?

What happens in terms of a step failure?

Pipeline Solutions

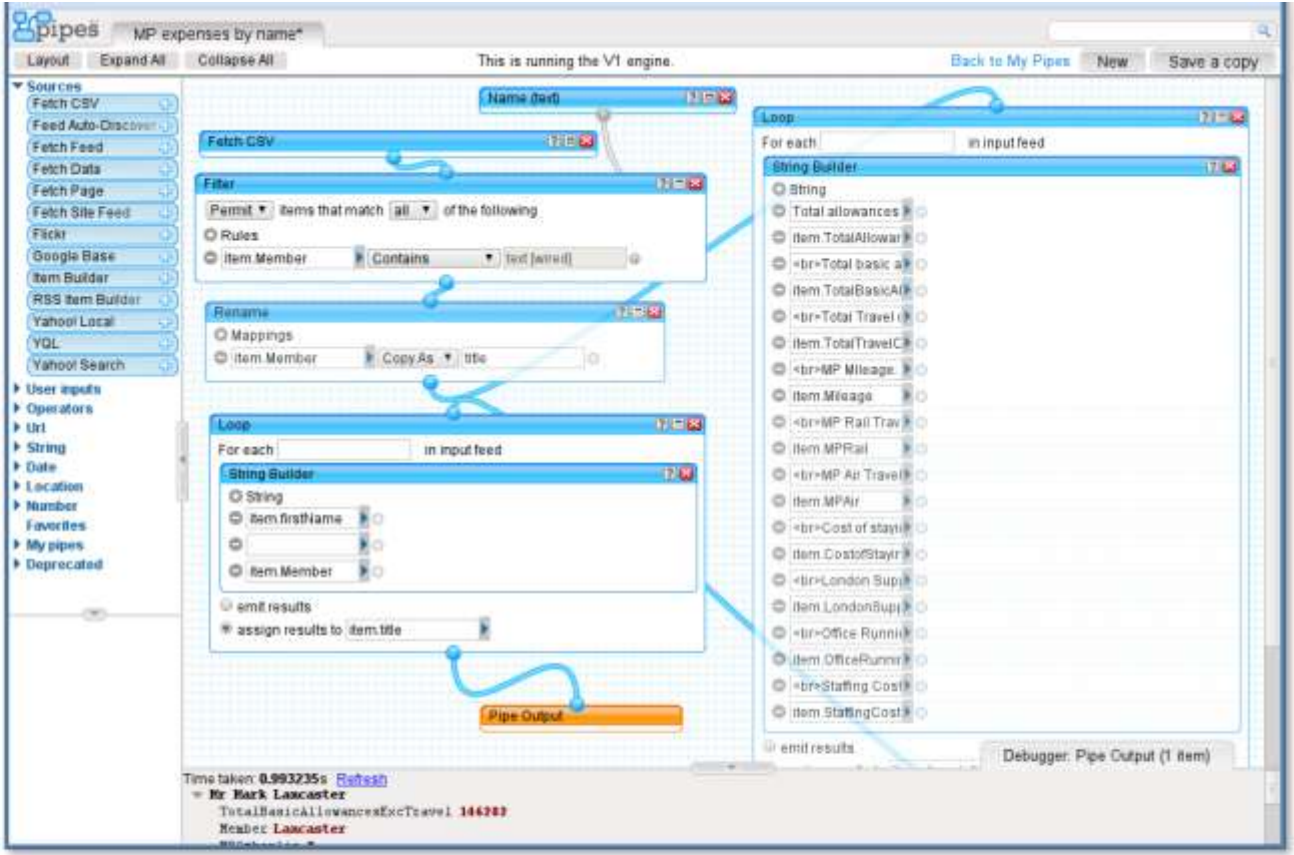


Graphical Pipelines

Your point and click, drag to connect
Specify inputs and outputs
Quite laborious IMHO

Lets take a look at some...

Yahoo Pipes



Permit items that match all of the following

Rules: item.Member Contains text [wired]

Mappings: item.Member Copy As title

For each in input feed

String: item.firstName; ; item.Member

○ emit results
● assign results to item.title

For each in input feed

String: Total allowances; item.TotalAllowar;
Total basic a; item.TotalBasicA;
Total Travel; item.TotalTravelC;
MP Mileage; item.Mileage;
MP Rail Trav; item.MPRail;
MP Air Travel; item.MPAir;
Cost of stayi; item.CostofStayir;
London Supp; item.LondonSupp;
Office Runni; item.OfficeRunni;
Staffing Cost; item.StaffingCost

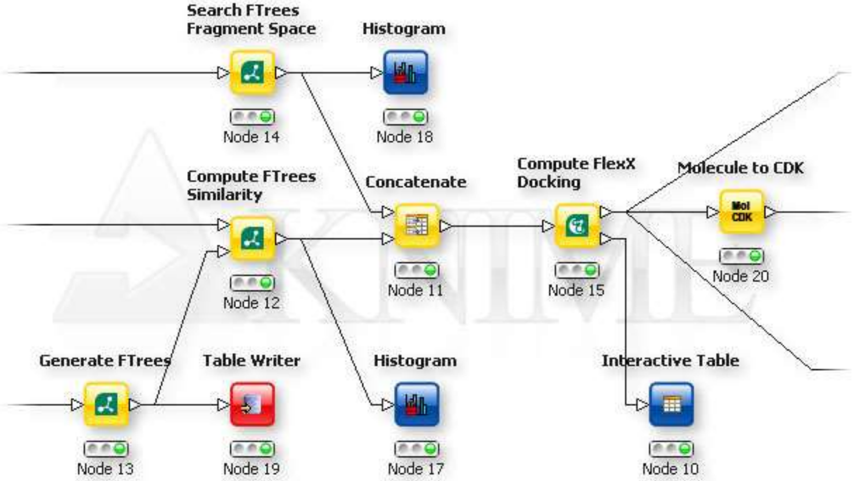
○ emit results

```
Time taken: 0.993235s Refresh
▾ Mr Mark Lancaster
    TotalBasicAllowancesExcTravel 146283
    Member Lancaster
```

Scientific Pipelines

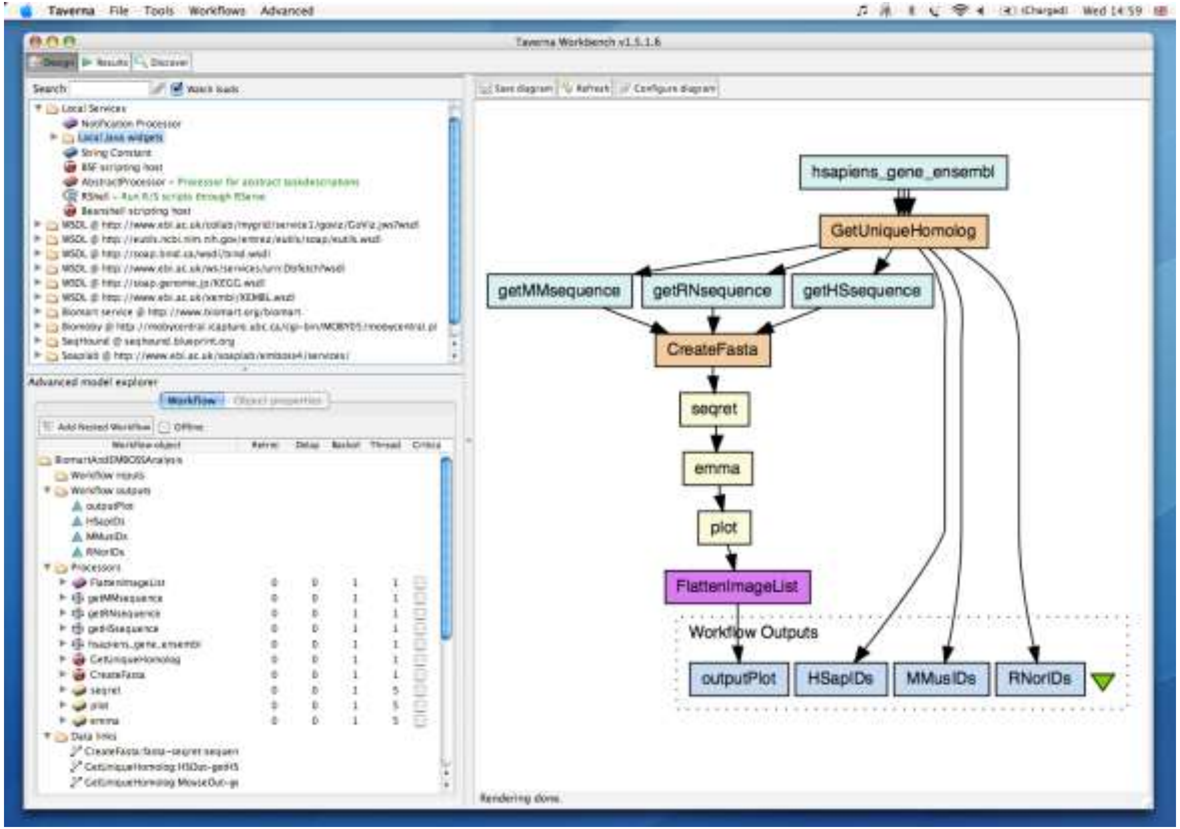
Saw graphical pipelines applied quite a lot in scientific workflows previously...

Bioinformatics
Geonomics



Graphical Pipeline Solutions

Knime
Taverna
Galaxy
Pipeline Pilot
Kepler



- Local Services
  - Notification Processor
  - Local Java widgets
  - String Constant
  - BSF scripting host
  - AbstractProcessor - Processor for abstract taskdescriptions
  - RShell - Run R/S scripts through RServe
  - Beanshell scripting host
- WSDL @ http://www.ebi.ac.uk/collab/mygrid/service1/goviz/GoViz.jws?wsdl
- WSDL @ http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl
- WSDL @ http://soap.bind.ca/wsdl/bind.wsdl
- WSDL @ http://www.ebi.ac.uk/ws/services/urn:Dbfetch?wsdl
- WSDL @ http://soap.genome.jp/KEGG.wsdl
- WSDL @ http://www.ebi.ac.uk/xembl/XEMBL.wsdl
- Biomart service @ http://www.biomart.org/biomart
- Biomoby @ http://mobycentral.icapture.ubc.ca/cgi-bin/MOBY05/mobycentral.pl
- SeqHound @ seqhound.blueprint.org
- Soaplab @ http://www.ebi.ac.uk/soaplab/emboss4/services/

Workflow object	Retries	Delay	Backoff	Threads	Critical
BiomartAndEMBOSSAnalysis					
Workflow inputs					
Workflow outputs					
outputPlot					
HSapIDs					
MMusIDs					
RNorIDs					
Processors					
FlattenImageList	0	0	1	1	□
getMMsequence	0	0	1	1	□
getRNsequence	0	0	1	1	□
getHSsequence	0	0	1	1	□
hsapiens_gene_ensembl	0	0	1	1	□
GetUniqueHomolog	0	0	1	1	□
CreateFasta	0	0	1	1	□
seqret	0	0	1	5	□
plot	0	0	1	5	□
emma	0	0	1	5	□
Data links					
CreateFasta:fasta→seqret:sequen					
GetUniqueHomolog:HSOut→getHS					
GetUniqueHomolog:MouseOut→ge					

Graphical Pipeline Summary

Nice, but generally found that:

- People don’t like/want to work graphically with pipelines, especially programmers

- Still suitable for non-programmers who just want to reuse past work though

Lightweight Pipeline Solutions

There’s some great lightweight solutions for building pipelines in various languages:

Luigi (Python)

Piecepipe (Ruby)

Spring Integration and Batch (Java)

Java Data Pipelines

Seeing as this is Bristol Java Meetup… there's a number of (non hadoop) Java data pipeline frameworks including:

NorthConcepts Data Pipeline

Java Data Processing Framework

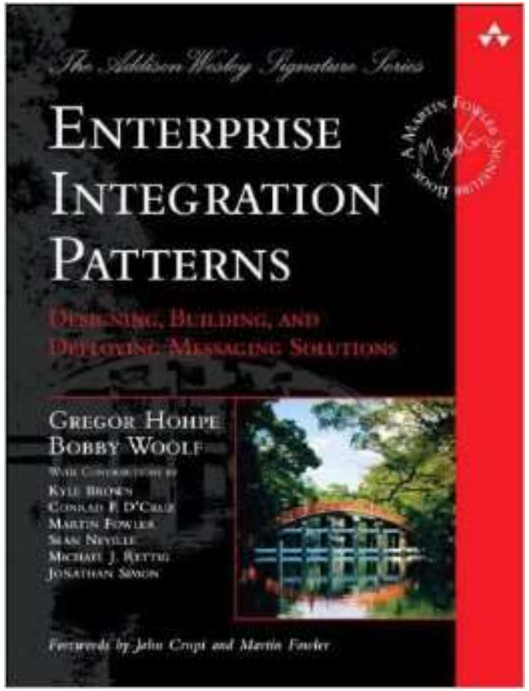
Spring Batch (and Spring Integration)

Enterprise Integration Patterns

Set of patterns describing generic integration patterns

Book By Gregor Hophe

Generally configure using XML
or a DSL of sorts


The Addison-Wesley Signature Series
ENTERPRISE INTEGRATION PATTERNS
DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS
A MARTIN FOWLER SIGNATURE BOOK
GREGOR HOHPE
BOBBY WOOLF
WITH CONTRIBUTIONS BY
KYLE BROWN
CONRAD F. D'CRUZ
MARTIN FOWLER
SEAN NEVILLE
MICHAEL J. RETTIG
JONATHAN SIMON
Forewords by John Crupi and Martin Fowler

EIP Implementations

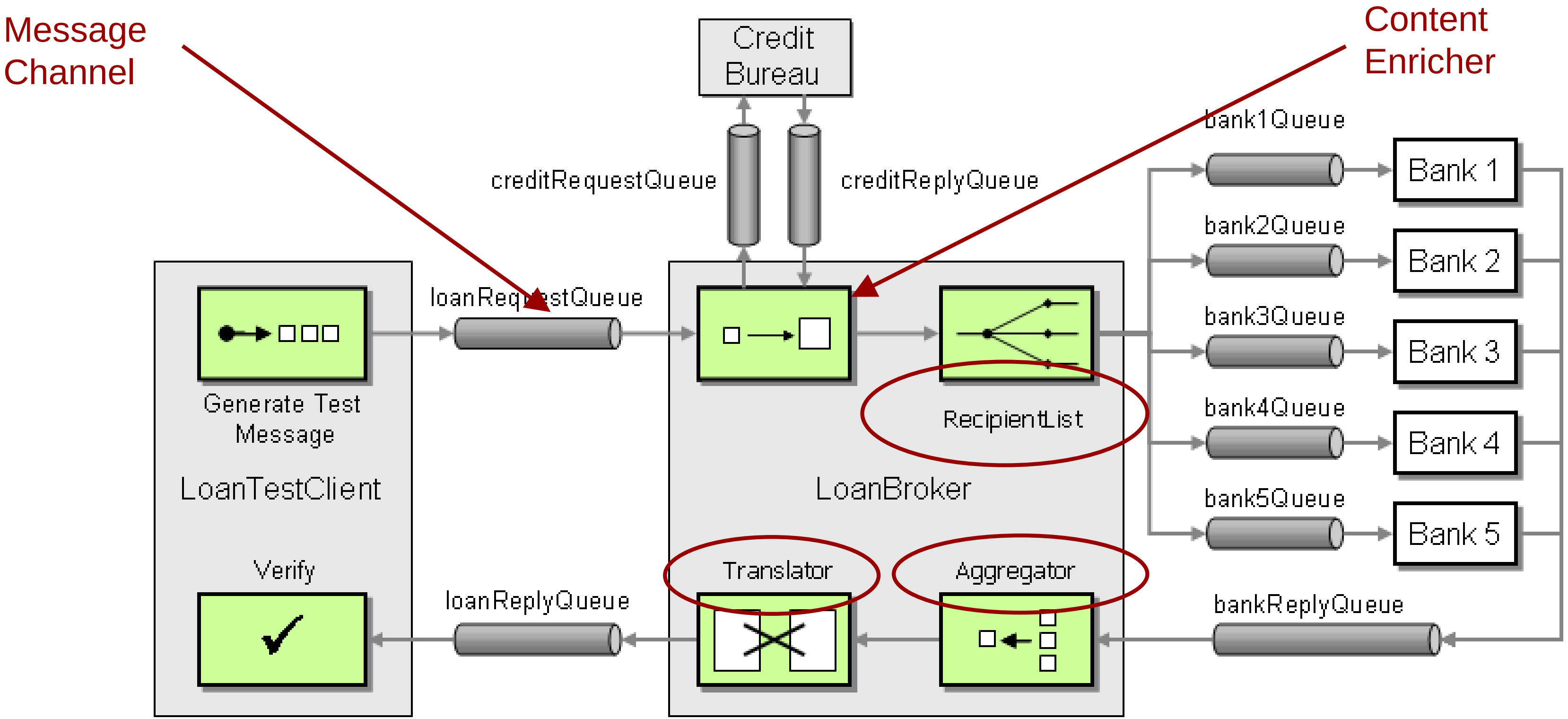
Implementations exist:

- Spring Integration
- Camel

Nice way of abstracting out components

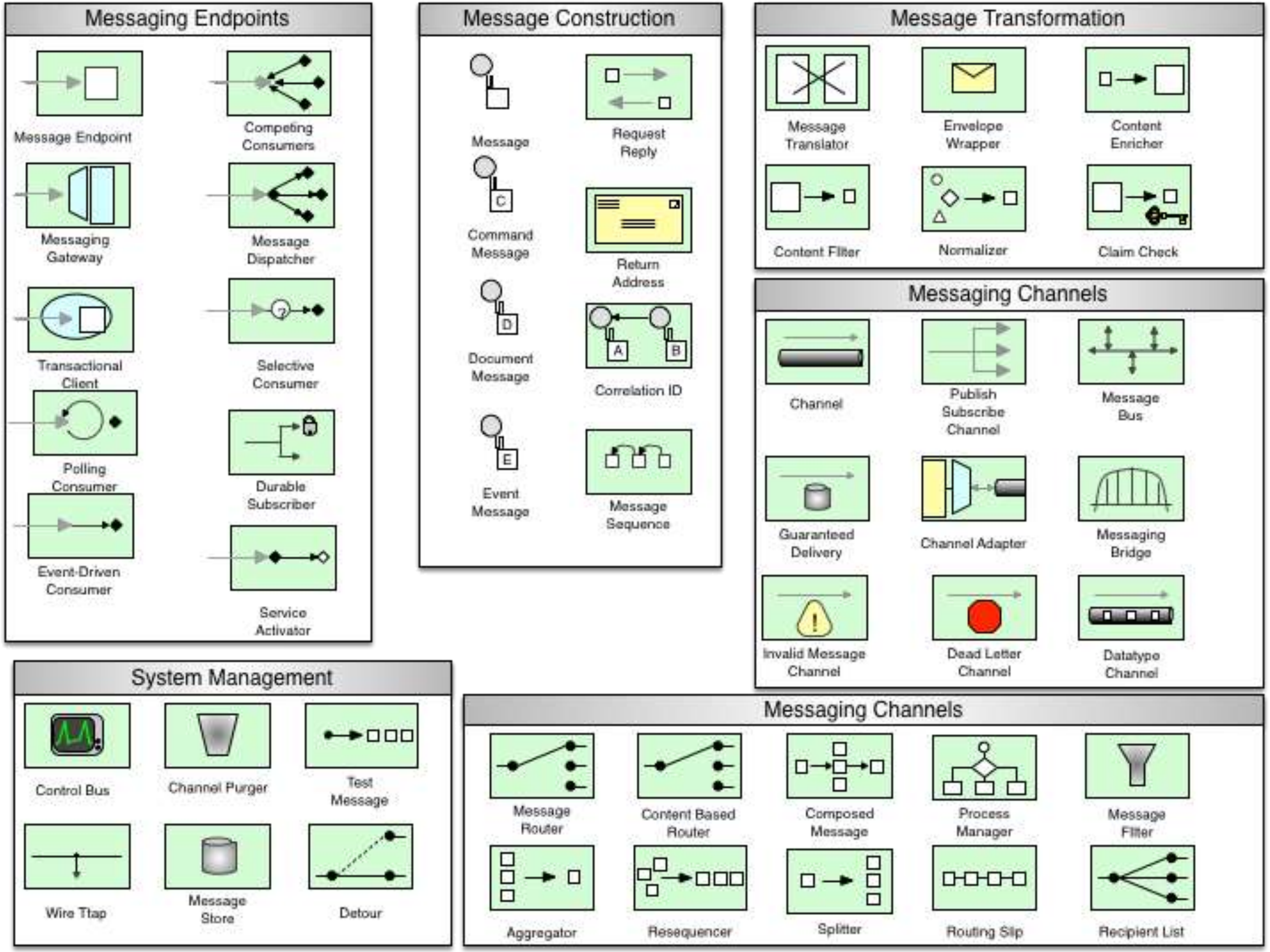
Somewhat leaky… control of fine grained threading can be problematic

Overview of EIP Patterns (Brief)



Message Channel

Content Enricher


Messaging Endpoints
Message Endpoint
Competing Consumers
Messaging Gateway
Message Dispatcher
Transactional Client
Selective Consumer
Polling Consumer
Durable Subscriber
Event-Driven Consumer
Service Activator
Message Construction
Message
Request Reply
C
Command Message
Return Address
D
Document Message
A
B
Correlation ID
E
Event Message
Message Sequence
Message Transformation
Message Translator
Envelope Wrapper
Content Enricher
Content Filter
Normalizer
Claim Check
Messaging Channels
Channel
Publish Subscribe Channel
Message Bus
Guaranteed Delivery
Channel Adapter
Messaging Bridge
!
Invalid Message Channel
Dead Letter Channel
Datatype Channel
System Management
Control Bus
Channel Purger
Test Message
Wire Tap
Message Store
Detour
Messaging Channels
Message Router
Content Based Router
Composed Message
Process Manager
Message Filter
Aggregator
Resequencer
Splitter
Routing Slip
Recipient List

Workflow Engines

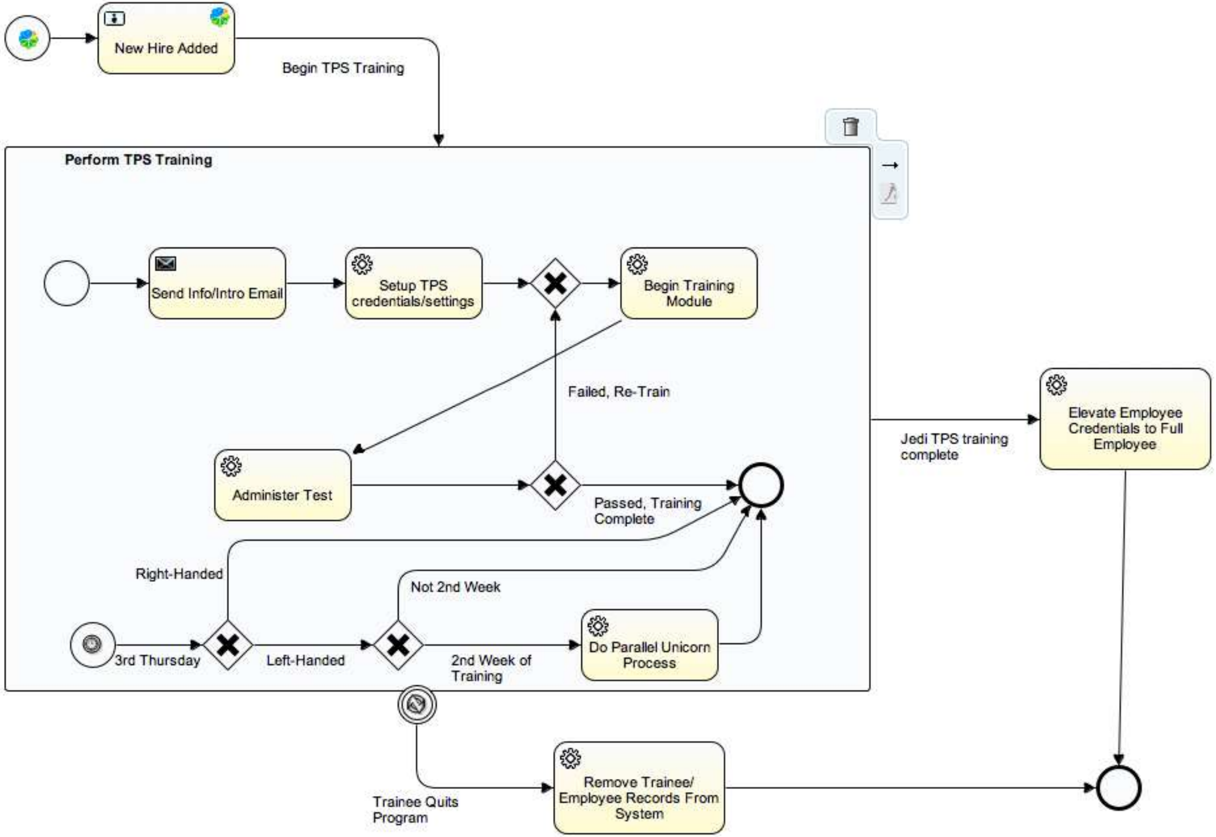
Another way to define a pipeline is as a workflow with several connected steps

Store intermediate state in a database

- JBPM
- Activiti

Execute processes written in BPEL or BPMN

Activiti Example



Luigi (Python)

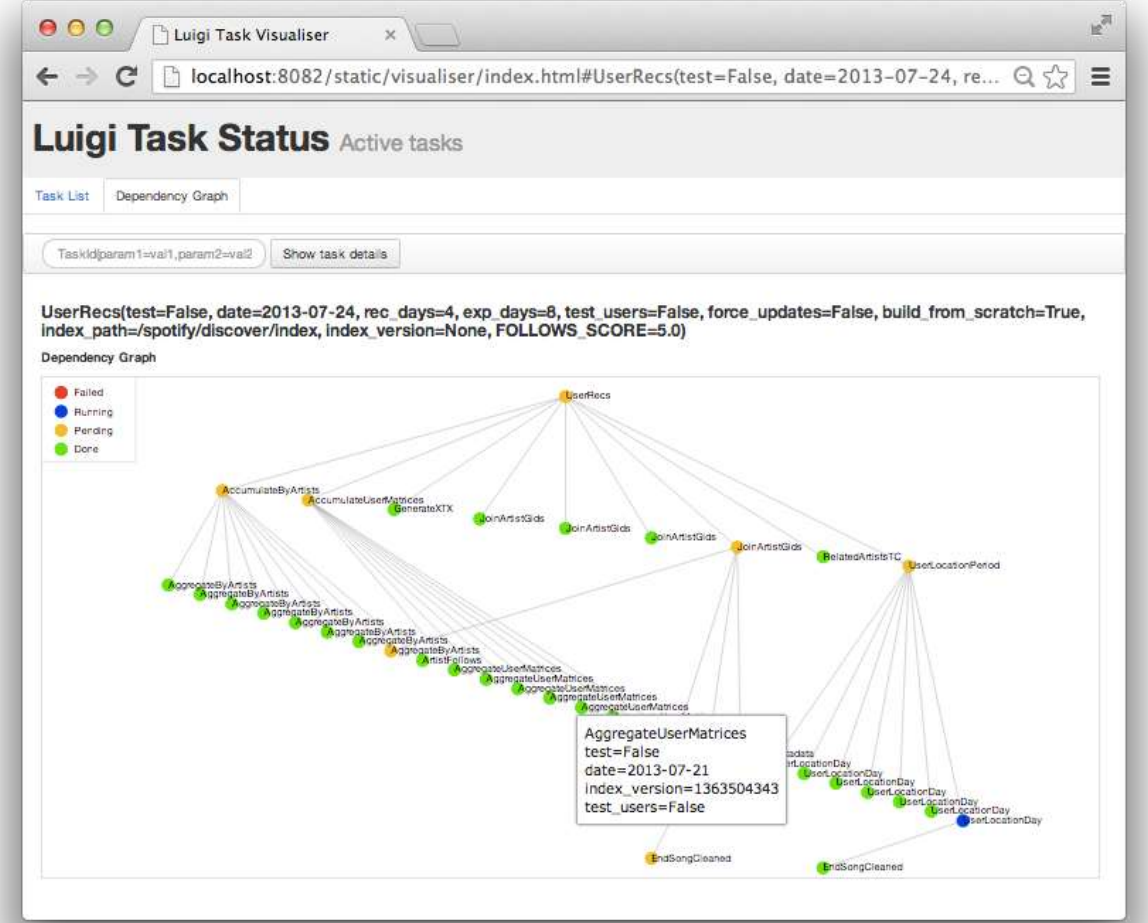
Luigi is used internally at Spotify to build complex data pipelines

Supports Hadoop, Spark for data processing

Supports Hive, HDFS and a range of other data sinks

Programmatic Example

```
class AggregateArtists(luigi.Task):
    date_interval = luigi.DateIntervalParameter()
    def run(self):
        artist_count = defaultdict(int)
        for input in self.input():
            with input.open('r') as in_file:
                for line in in_file:
                    timestamp, artist, track = line.strip().split()
                    artist_count[artist] += 1
        with self.output().open('w') as out_file:
            for artist, count in artist_count.iteritems():
                print >> out_file, artist, count
```



Luigi Task Visualiser
localhost:8082/static/visualiser/index.html#UserRecs(test=False, date=2013-07-24, re...

Luigi Task Status Active tasks

UserRecs(test=False, date=2013-07-24, rec_days=4, exp_days=8, test_users=False, force_updates=False, build_from_scratch=True, index_path=/spotify/discover/index, index_version=None, FOLLOWS_SCORE=5.0)

Dependency Graph

AggregateUserMatrices
test=False
date=2013-07-21
index_version=1363504343
test_users=False

Piecepipe (Ruby)

Made up of a set of steps (contractive, iterating, transforming)

Assembly Steps use hashes as their inputs

Can use for partial processing

Piecepipe (Ruby)

```ruby
PiecePipe::Pipeline.new.
      source([{region: region}]).
      step(FetchPowerPlantsByRegion).
      step(FindWorstReactor).
      step(DetermineStatusClass).
      step(BuildPlantHealthSummary).
      step(SortByRadiationLevelsDescending).
      collect(:plant_health_summary).
      to_enum
```

Spring Integration - XML Hell

```
<si:transformer id="t1" input-channel="flow1.inputChannel"
output-channel="sa1.inputChannel" expression="'Hello,' +
payload"/>

<si:service-activator id="sa1" input-
channel="sa.inputChannel" expression =
"T(java.lang.System).out.println(payload)"/>
```

Spring Integration - Groovy DSL

```
messageFlow {
    transform {"hello, $it"}
    handle {println it}
}
```

Spring Batch

More suited to batch processing (as per the name)

Specify the pipeline in XML or a DSL:

- Job
- Step, tasklets and chunks
- Chunks refer to Spring beans

Spring Batch

```
<batch:job id="reportJob">
   <batch:step id="step1">
     <batch:tasklet>
       <batch:chunk reader="cvsFileItemReader"
   writer="mysqlItemWriter" commit-interval="2">
       </batch:chunk>
     </batch:tasklet>
   </batch:step>
</batch:job>
```

Enterprise Service Bus

And it goes on and on...

I’m not covering this…

Camel, Mule yada yada

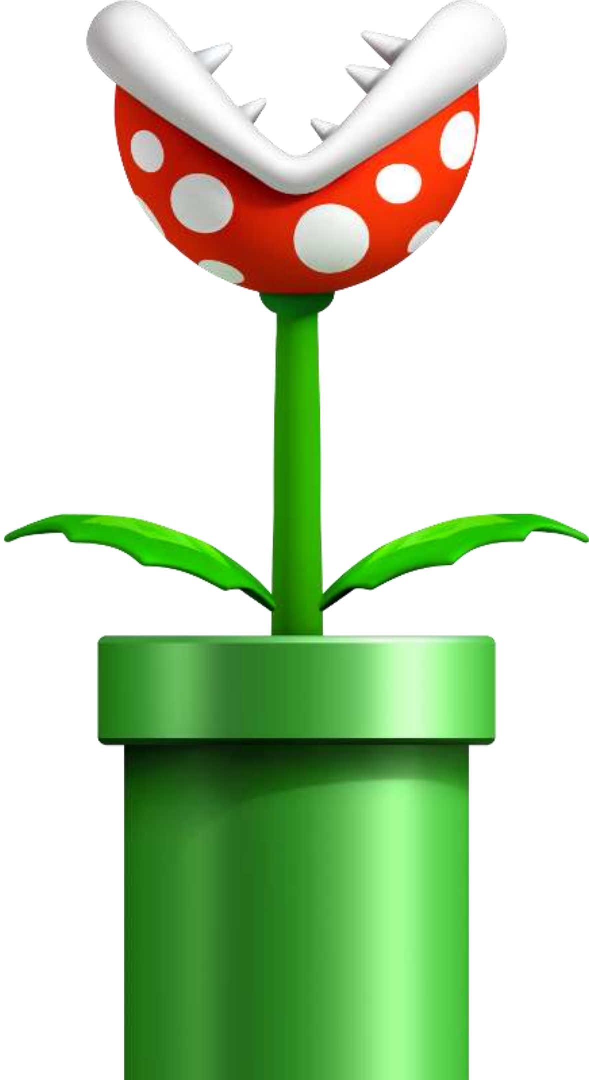
Write your Own?

Simplistic set of database tables, abstract Step, Job and Workflow concepts

Link to scripts in Groovy/Clojure/JRuby for dynamic

Store state in the filesystem, database or pass hashmaps between the steps

Simples



Big Data Pipelines

Big Data Pipelines

Everybody loves Big Data right?

There he is the smiley bastard...

Pipelines we’ve seen so far

Generally serial - no good when we need to shift large amounts of data around.

Better option is to parallelise the processing

In some of the cases before (e.g. Luigi) you can shift processing to Hadoop, Storm etc...

Parallelising the data

Java works well here with a great concurrency API, executor services, locks etc…

Cumbersome to write, we have to partition the data correctly to begin with

Spring integration/batch helps with abstractions

Hadoop - map reduce

Hadoop is a good solution for shifting big data

Batch processing though - job startup time can take minutes

Also it’s pretty cumbersome to write Hadoop mappers/reducers in Java

Pig and Oozie

Fortunately we have some help

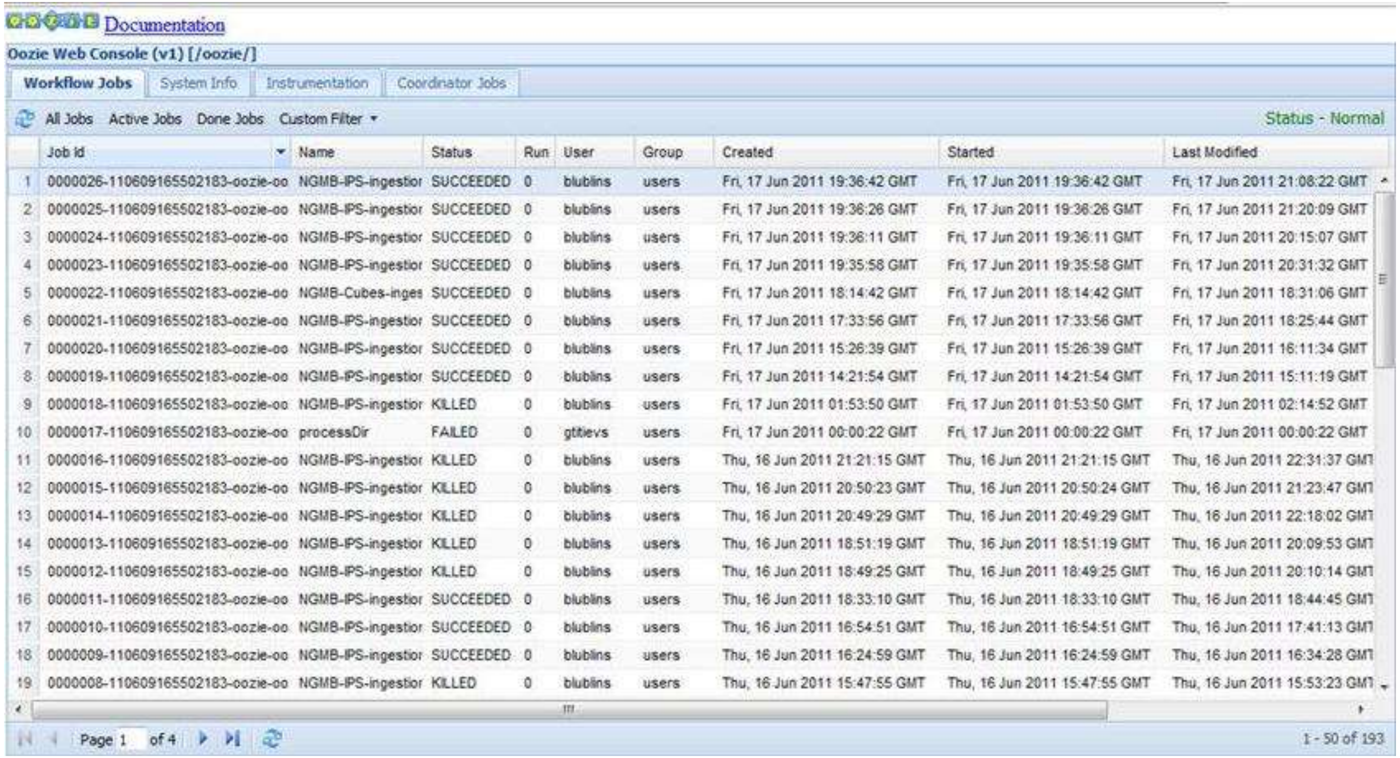
Pig is a SQL like language aimed at making MapReduce jobs easier to write

Oozie provides a way of building graphical pipelines using Hadoop

Word Count - yada yada

```
a = load 'words.txt';

b = foreach a generate flatten(TOKENIZE((chararray)$0)) as
word;

c = group b by word;

d = foreach c generate COUNT(b), group;

store d into '/user/jon';
```

Oozie Workflow



Status - Normal

	Job id	Name	Status	Run	User	Group	Created	Started	Last Modified
1	0000026-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 19:36:42 GMT	Fri, 17 Jun 2011 19:36:42 GMT	Fri, 17 Jun 2011 21:08:22 GMT
2	0000025-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 19:36:26 GMT	Fri, 17 Jun 2011 19:36:26 GMT	Fri, 17 Jun 2011 21:20:09 GMT
3	0000024-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 19:36:11 GMT	Fri, 17 Jun 2011 19:36:11 GMT	Fri, 17 Jun 2011 20:15:07 GMT
4	0000023-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 19:35:58 GMT	Fri, 17 Jun 2011 19:35:58 GMT	Fri, 17 Jun 2011 20:31:32 GMT
5	0000022-110609165502183-oozie-oo	NGMB-Cubes-inges	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 18:14:42 GMT	Fri, 17 Jun 2011 18:14:42 GMT	Fri, 17 Jun 2011 18:31:06 GMT
6	0000021-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 17:33:56 GMT	Fri, 17 Jun 2011 17:33:56 GMT	Fri, 17 Jun 2011 18:25:44 GMT
7	0000020-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 15:26:39 GMT	Fri, 17 Jun 2011 15:26:39 GMT	Fri, 17 Jun 2011 16:11:34 GMT
8	0000019-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Fri, 17 Jun 2011 14:21:54 GMT	Fri, 17 Jun 2011 14:21:54 GMT	Fri, 17 Jun 2011 15:11:19 GMT
9	0000018-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Fri, 17 Jun 2011 01:53:50 GMT	Fri, 17 Jun 2011 01:53:50 GMT	Fri, 17 Jun 2011 02:14:52 GMT
10	0000017-110609165502183-oozie-oo	processDir	FAILED	0	gtitievs	users	Fri, 17 Jun 2011 00:00:22 GMT	Fri, 17 Jun 2011 00:00:22 GMT	Fri, 17 Jun 2011 00:00:22 GMT
11	0000016-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Thu, 16 Jun 2011 21:21:15 GMT	Thu, 16 Jun 2011 21:21:15 GMT	Thu, 16 Jun 2011 22:31:37 GMT
12	0000015-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Thu, 16 Jun 2011 20:50:23 GMT	Thu, 16 Jun 2011 20:50:24 GMT	Thu, 16 Jun 2011 21:23:47 GMT
13	0000014-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Thu, 16 Jun 2011 20:49:29 GMT	Thu, 16 Jun 2011 20:49:29 GMT	Thu, 16 Jun 2011 22:18:02 GMT
14	0000013-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Thu, 16 Jun 2011 18:51:19 GMT	Thu, 16 Jun 2011 18:51:19 GMT	Thu, 16 Jun 2011 20:09:53 GMT
15	0000012-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Thu, 16 Jun 2011 18:49:25 GMT	Thu, 16 Jun 2011 18:49:25 GMT	Thu, 16 Jun 2011 20:10:14 GMT
16	0000011-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Thu, 16 Jun 2011 18:33:10 GMT	Thu, 16 Jun 2011 18:33:10 GMT	Thu, 16 Jun 2011 18:44:45 GMT
17	0000010-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Thu, 16 Jun 2011 16:54:51 GMT	Thu, 16 Jun 2011 16:54:51 GMT	Thu, 16 Jun 2011 17:41:13 GMT
18	0000009-110609165502183-oozie-oo	NGMB-IPS-ingestior	SUCCEEDED	0	blublins	users	Thu, 16 Jun 2011 16:24:59 GMT	Thu, 16 Jun 2011 16:24:59 GMT	Thu, 16 Jun 2011 16:34:28 GMT
19	0000008-110609165502183-oozie-oo	NGMB-IPS-ingestior	KILLED	0	blublins	users	Thu, 16 Jun 2011 15:47:55 GMT	Thu, 16 Jun 2011 15:47:55 GMT	Thu, 16 Jun 2011 15:53:23 GMT

Page 1 of 4

1 - 50 of 193

Hadoop Workflow

There’s lots more...

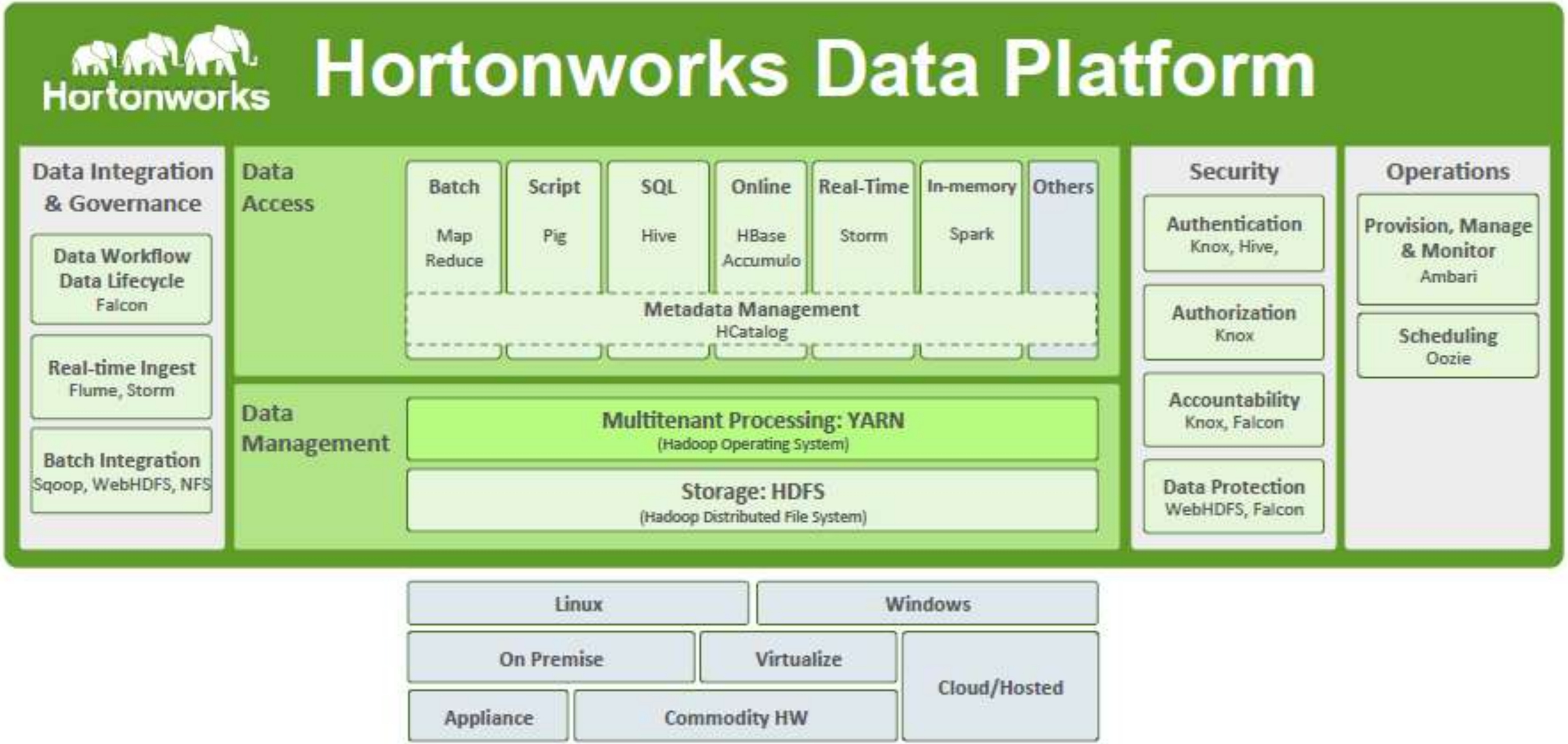
Netflix Linkedin Cascading Luigi

Big Data Vendors

Lots of solutions for generalising data processing into a platform solution

Hadoop - batch processing
Storm - real-time interactive processing

Similar to the graphical pipelines we saw earlier on



Hortonworks

Hortonworks Data Platform

Data Integration & Governance

Data Workflow
Data Lifecycle
Falcon

Real-time Ingest
Flume, Storm

Batch Integration
Sqoop, WebHDFS, NFS

Data Access

Batch
Map Reduce

Script
Pig

SQL
Hive

Online
HBase Accumulo

Real-Time
Storm

In-memory
Spark

Others

Metadata Management
HCatalog

Data Management

Multitenant Processing: YARN
(Hadoop Operating System)

Storage: HDFS
(Hadoop Distributed File System)

Security

Authentication
Knox, Hive,

Authorization
Knox

Accountability
Knox, Falcon

Data Protection
WebHDFS, Falcon

Operations

Provision, Manage & Monitor
Ambari

Scheduling
Oozie

Linux

Windows

On Premise

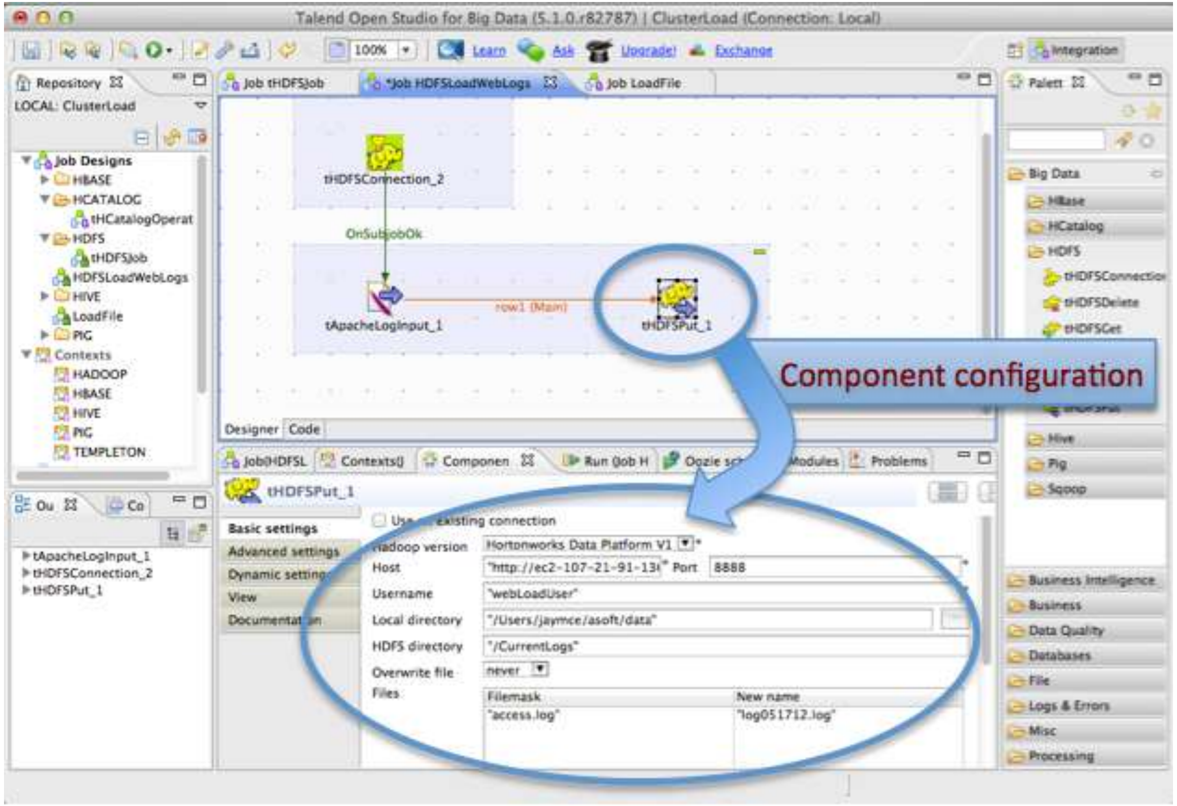
Virtualize

Cloud/Hosted

Appliance

Commodity HW

Talend Open Studio



Talend Open Studio for Big Data (5.1.0.r82787) | ClusterLoad (Connection: Local)

Component configuration

tHDFSPut_1

Basic settings

□ Use an existing connection

Hadoop version: Hortonworks Data Platform V1

Host: "http://ec2-107-21-91-13(Port: 8888

Username: "webLoadUser"

Local directory: "/Users/jaymce/asoft/data"

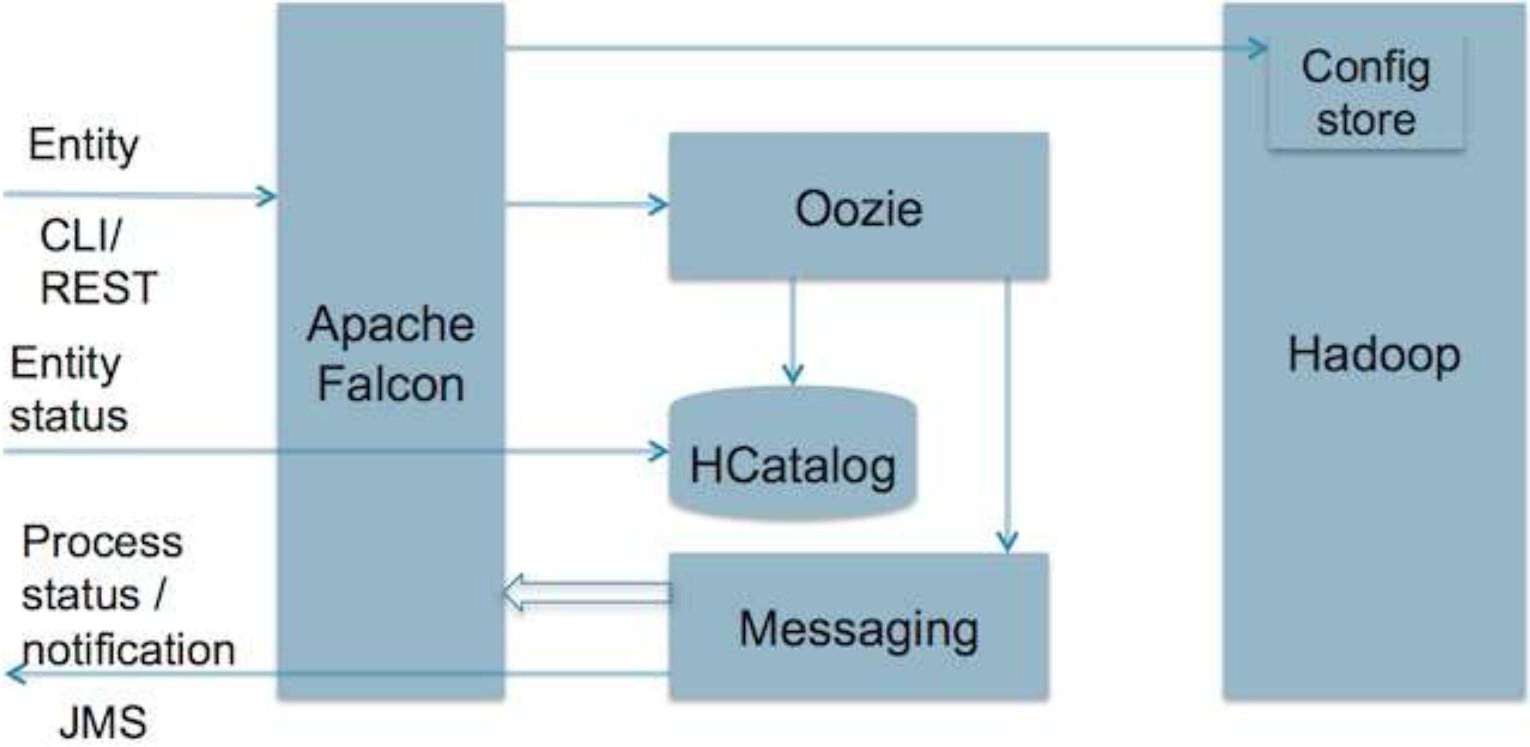
HDFS directory: "/CurrentLogs"

Overwrite file: never

Files

Filemask	New name
"access.log"	"log051712.log"

Apache Falcon



Apache Falcon - Data Processing

Apache Incubator project, 0.6 currently

Think of it as an open-source data management platform

Late data handling, retry policies etc...

Colo and global aggregations

Cloud Pipelines

Amazon Web Service Components

We’re big users of AWS and components:

- AWS RDS - main datastore
- AWS Datapipe - basic ETL processes (log processing)
- AWS DynamoDB - dead code analysis
- AWS S3 - Log file source

Also use BigQuery to store large amounts of parsed access logs

Amazon Web Service Components

AWS provides a number of tools for building data pipelines

Some are packaged open-source tools

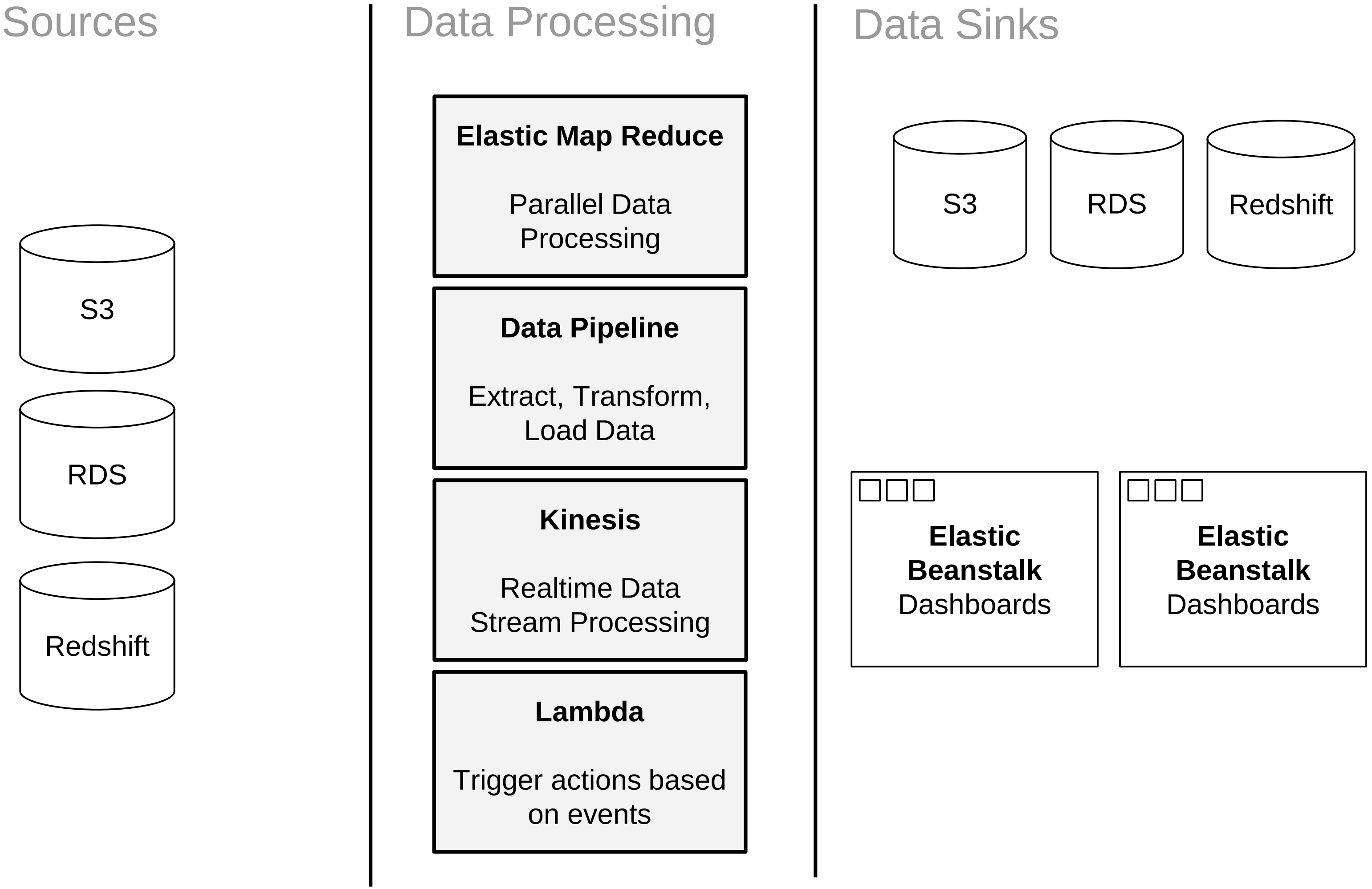
Useful when you don’t want to setup your own infrastructure.

Not necessarily cheaper though

Google Pipeline Components

Google also has a set of components for building data pipelines:

- Cloud Dataflow - (AWS Data Pipeline)
- Google Cloud Storage (AWS S3)
- BigQuery - (AWS Redshift)
- App Engine Pipeline API



Data Sources

S3

RDS

Redshift

Data Processing

Elastic Map Reduce

Parallel Data Processing

Data Pipeline

Extract, Transform, Load Data

Kinesis

Realtime Data Stream Processing

Lambda

Trigger actions based on events

Data Sinks

S3

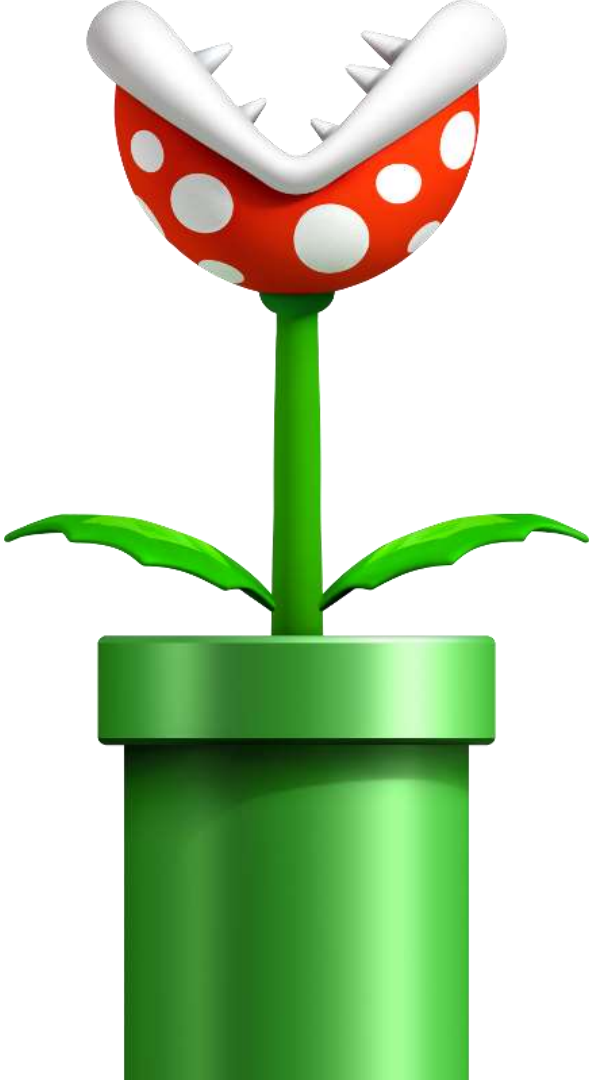
RDS

Redshift

Elastic Beanstalk Dashboards

Elastic Beanstalk Dashboards

Elastic Map Reduce



Elastic Map Reduce

Hadoopken...

- Hadoop distro is Amazon’s own
- Allows you to create a cluster with n nodes
- Logs directly to S3
- Security/Access through standard IAM

Elastic Map Reduce

Choice of applications to install in addition:

- Various databases (Hive, HBase or Impala)
- Pig - SQL like data processing
- Hunk - Splunk analytics for Hadoop
- Ganglia - for monitoring the cluster

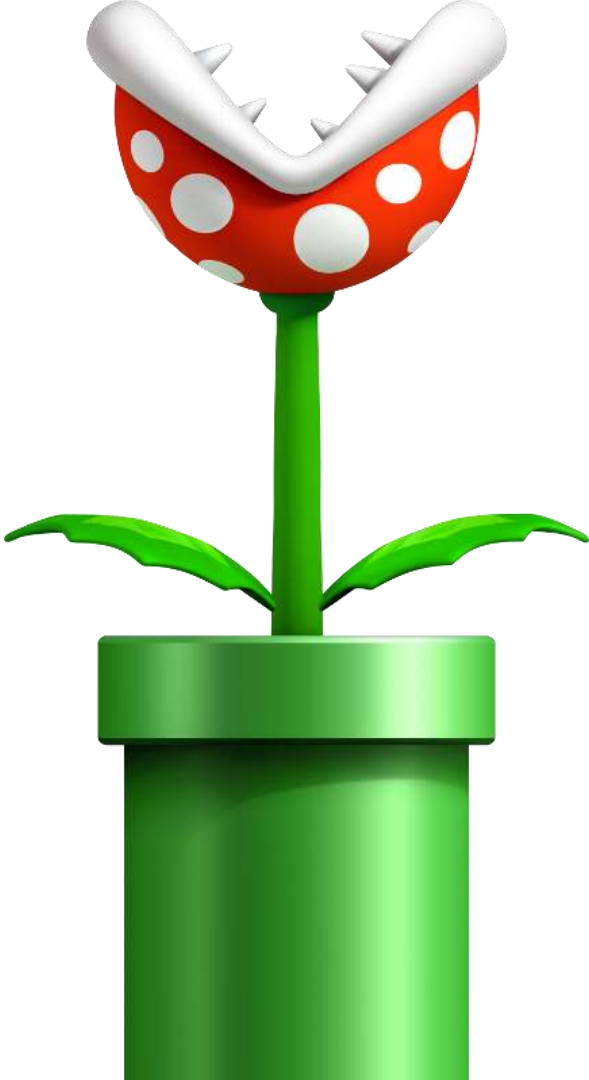
Elastic Map Reduce

Bootstrap actions before Hadoop starts

Submit unit of work to the cluster:

- Streaming program
- Hive
- Pig - can do streaming with this as well...
- Impala
- Custom JAR

AWS Data Pipeline



Data Pipeline

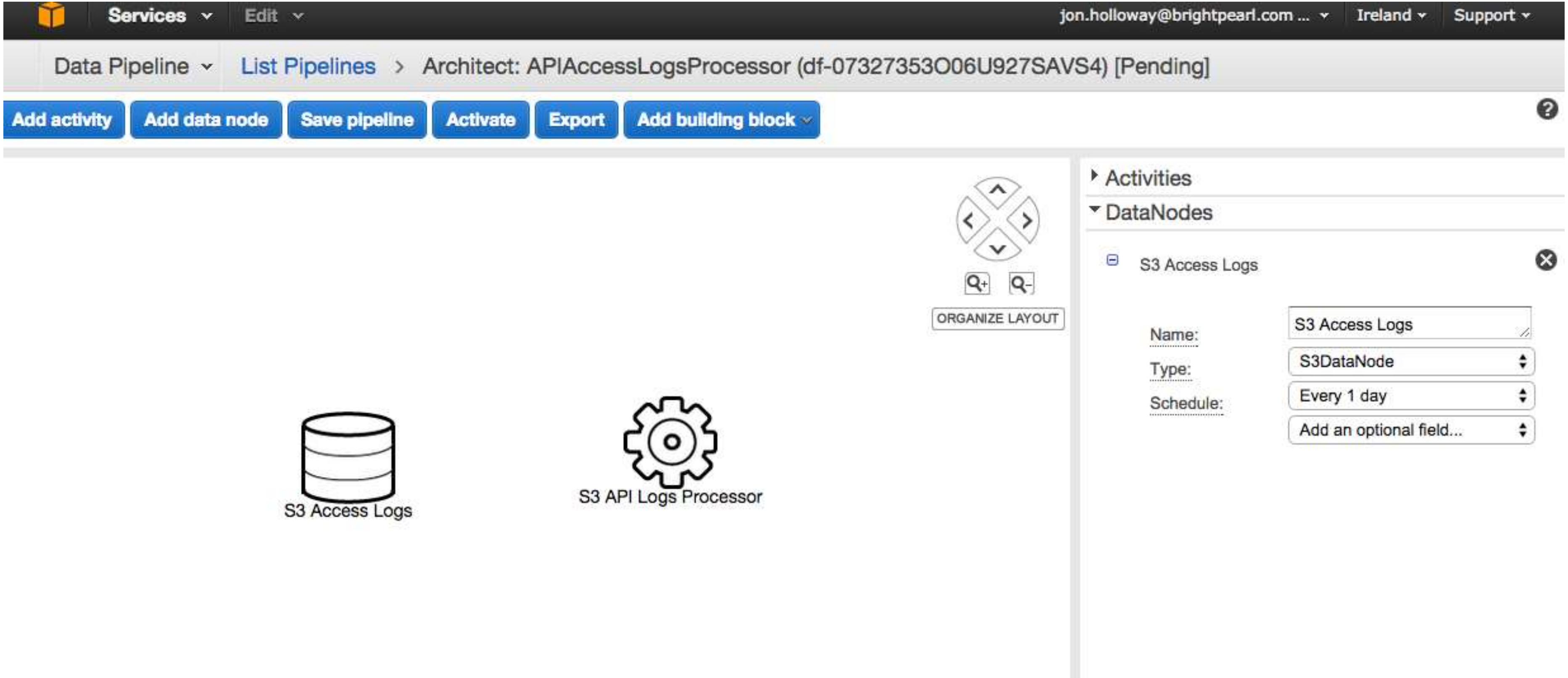
Data Node - the source of our data

- Amazon S3 Filesystem
- MySQL, Redshift, Dynamo, SQL Data Node

Activity - processor

- Shell Commands
- Copy Command
- EMR, Pig, Hive
- SQL Command

Data Pipeline



Data Pipeline > List Pipelines > Architect: APIAccessLogsProcessor (df-07327353O06U927SAVS4) [Pending]

S3 Access Logs

Name: S3 Access Logs

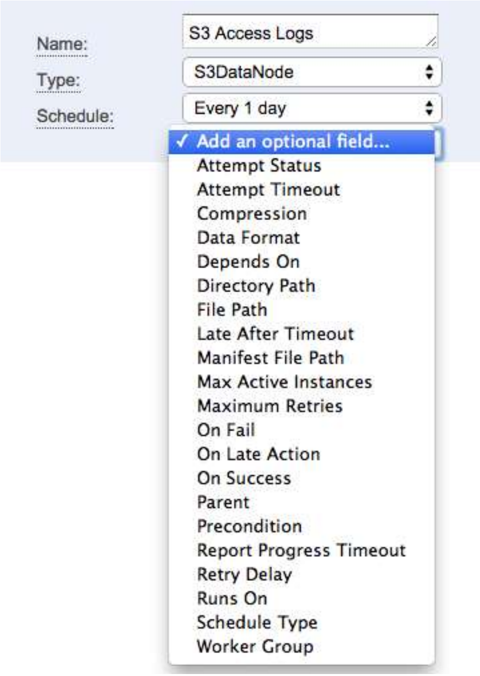
Type: S3DataNode

Schedule: Every 1 day

Add an optional field...

Data Pipeline - Source

- Define a datasource - S3 Logs
  - Runs every day
  - Can specify:
    - Attempts
    - Failure Actions
    - Success Actions/Follow up jobs



Name: S3 Access Logs

Type: S3DataNode

Schedule: Every 1 day

Data Pipeline - Source

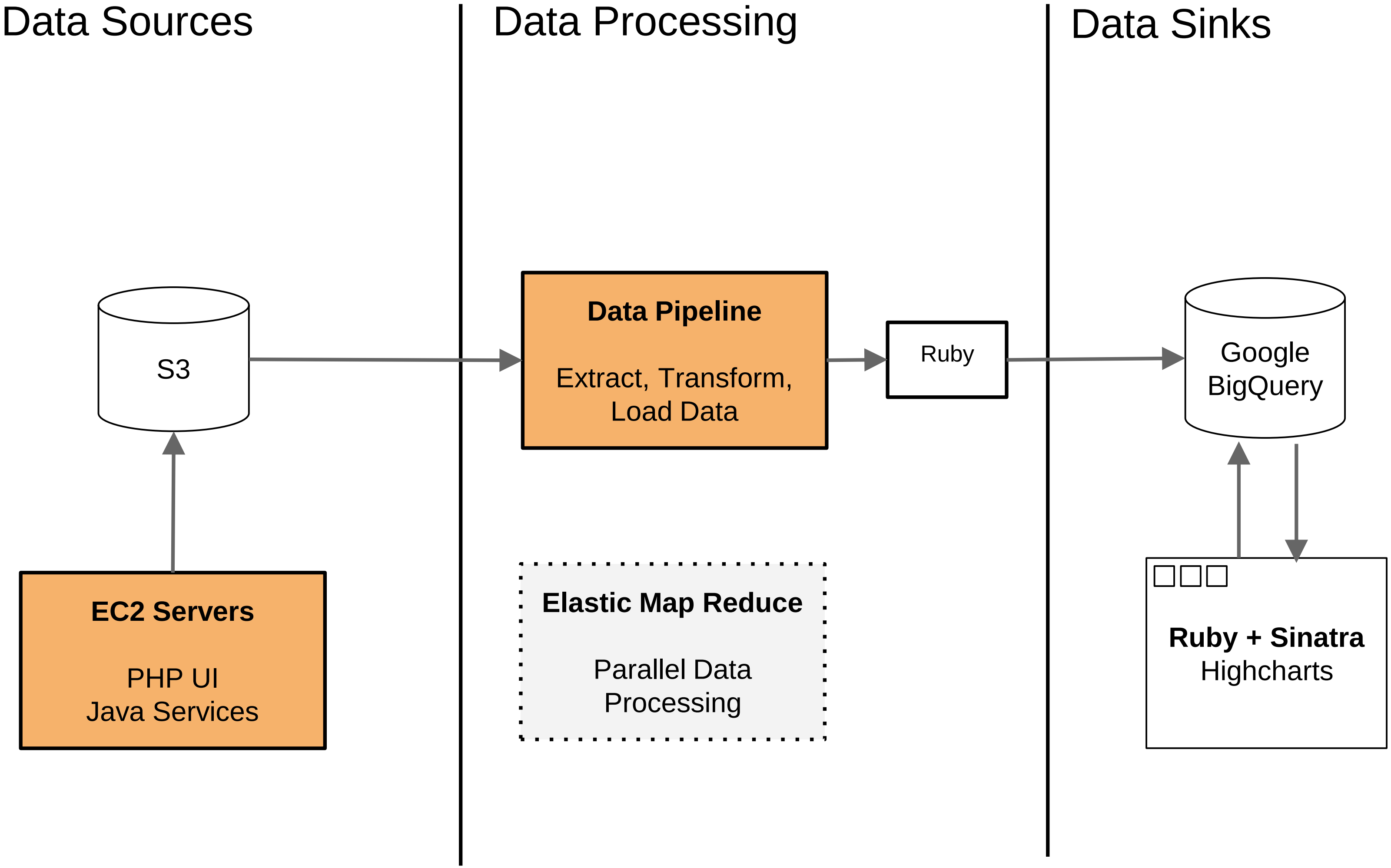
Define a datasource - S3 Logs

- Runs every day
- Can specify:
  - Attempts
  - Failure Actions
  - Success Actions/Follow up jobs

Data Pipeline - Processor

Simple processor that specifies a Ruby script to run to:

- Read in files from S3
- Parse the access log lines with universal-access-log-parser (awesome)
- Insert the results into a BigQuery table



Data Sources

Data Processing

Data Sinks

S3

EC2 Servers

PHP UI
Java Services

Data Pipeline

Extract, Transform,
Load Data

Ruby

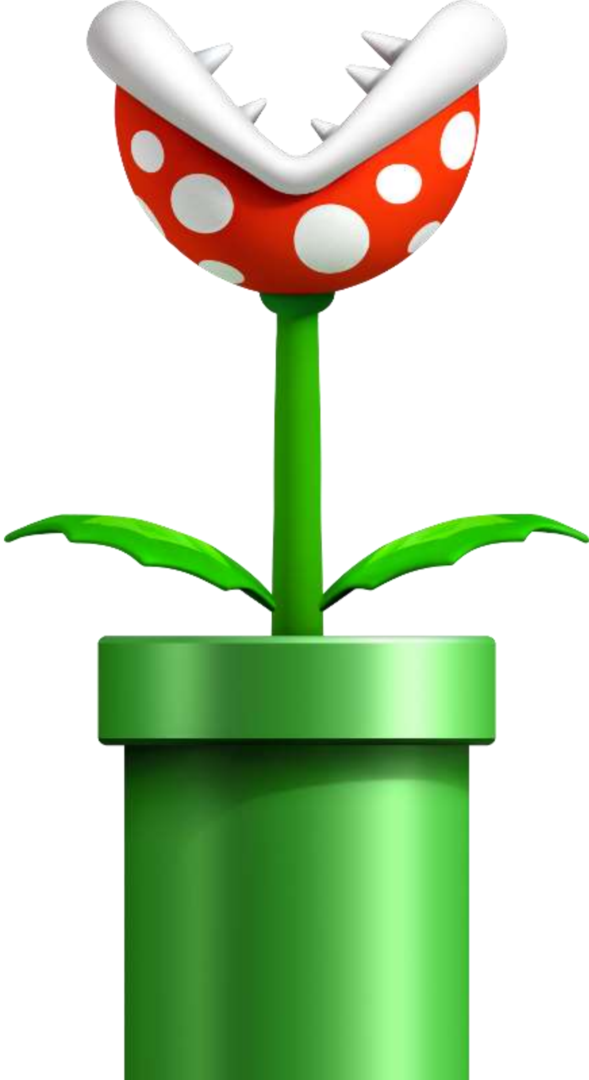
Elastic Map Reduce

Parallel Data
Processing

Google
BigQuery

Ruby + Sinatra
Highcharts

AWS Kinesis



AWS Kinesis (Real Time Streaming)

Simple set of steps:

Ingestion of data (Producer)
Durable storage of data
Consumers do parallel processing
Only keeps 24 hours of data

Sending Data (PutRecordsRequest)

```
PutRecordsRequest req = new PutRecordsRequest();
req.setStreamName(myStreamName);
List<PutRecordsRequestEntry> entryList = new ArrayList<>();

for (int j = 0; j < 100; j++) {
    PutRecordsRequestEntry entry = new PutRecordsRequestEntry();
    entry.setData(ByteBuffer.wrap(String.valueOf(i).getBytes()));
    entry.setPartitionKey(String.format("partitionKey-%d", i));
    entryList.add(entry);
}
PutRecordsResult res = kinesis.putRecords(req);
```

Consuming Data (GetRecords)

Need to use the Kinesis Client Library

Implement IRecordProcessor class and implement init, processRecords and shutdown

Call checkpoint when done

https://github.com/awslabs/amazon-kinesis-client

Scaling this out

Number of ways:

- Assign more shards to a stream
- Increase the EC2 instance size
- Increase EC2 instances up to max no. shards

Shards are stored in DynamoDB

Can also use auto scaling to modify the number of shards

Utilising Kinesis

- Alternative to parsing your logs for various metrics - these might be a day old

- Real-time metrics information from various sources:
  - Rate of orders being processed
  - Web application click stream data

AWS Lambda (Preview currently)

Runs code in response to events

- Data Events
- Scheduled Events
- External Events (sensors, alarms etc…)

Provision EC2 instances or AWS Containers

Auto scaling

Questions? Btw… we’re recruiting

Sysadmin/Devops

AWS (S3, EC2, Dynamo, Data Pipeline) Ruby and Chef hackery

Senior/Mid Java Developers

Elasticsearch, RabbitMQ, Hazelcast, Spring

Solutions Architect

eCommerce experience

www.brightpearl.com/careers