Lezione 1 – Servizi Web REST

Architetture orientate ai servizi

Workflow Engines

Ernesto Damiani

Università di Milano

Apache Oozie

https://oozie.apache.org/

http://hortonworks.com/hadoop/oozie/

Oozie is a scalable, reliable and extensible server-based workflow scheduler system to manage Apache Hadoop jobs coming from Yahoo.

Oozie Workflow jobs are Directed Acyclical Graphs (DAGs) which specify a sequence of actions to execute. Oozie also provides Coordinator jobs, which are Workflow jobs which can be scheduled to recur based on time or data availability.

Oozie can schedule various types of Hadoop jobs (map-reduce, Pig, Hive and others) as well as other types of jobs such as Java programs and shell scripts.

Apache ODE™

The Orchestration Director Engine ODE http://ode.apache.org/ executes business processes written following the WS-BPEL standard. This is "Web Service approach to orchestration/workflow" and stresses control flow whereas usually dataflow more natural

Side-by-side support for both the WS-BPEL 2.0 OASIS standard http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html and the legacy BPEL4WS 1.1 vendor specification. (BPEL = Business Process Execution Language)

Supports 2 communication layers: one based on Axis2 (Web Services http transport) and another one based on the JBI http://en.wikipedia.org/wiki/Java_Business_Integration standard (http://jcp.org/en/jsr/detail?id=208).

Support for the HTTP WSDL binding (initial SOAP-based Web Service protocol), allowing invocation of REST-style (this won) web services.

Possibility to map process variables externally to a database table of your choice.

High level API to the engine that allows you to integrate the core with virtually any communication layer.

Hot-deployment of your processes.

Compiled approach to BPEL that provides detailed analysis and validation at the command line or at deployment.

Management interface for processes, instances and messages.

There were other open source BPEL projects but maybe this is only remaining one?

http://en.wikipedia.org/wiki/List_of_BPEL_engines

ActiveBPEL

This was top open source BPEL implementation from "Active Endpoints" which was acquired by Informatica

The last release was 5.0.2 in 2008 http://activebpel502.sourceforge.net/

See http://www.activevos.com/cp/329/active-endpoints-releases-milestone-1-of-activebpel-community-edition-with-bpel4people

Replaced by ActiveVOS (Visual Orchestration System) http://www.activevos.com/; proprietary coding of open BPEL standard and supports people in the loop (BPEL4People)

- Implements BPMN visual interface (Business Process Model and Notation)

Apache Airavata

https://airavata.apache.org/

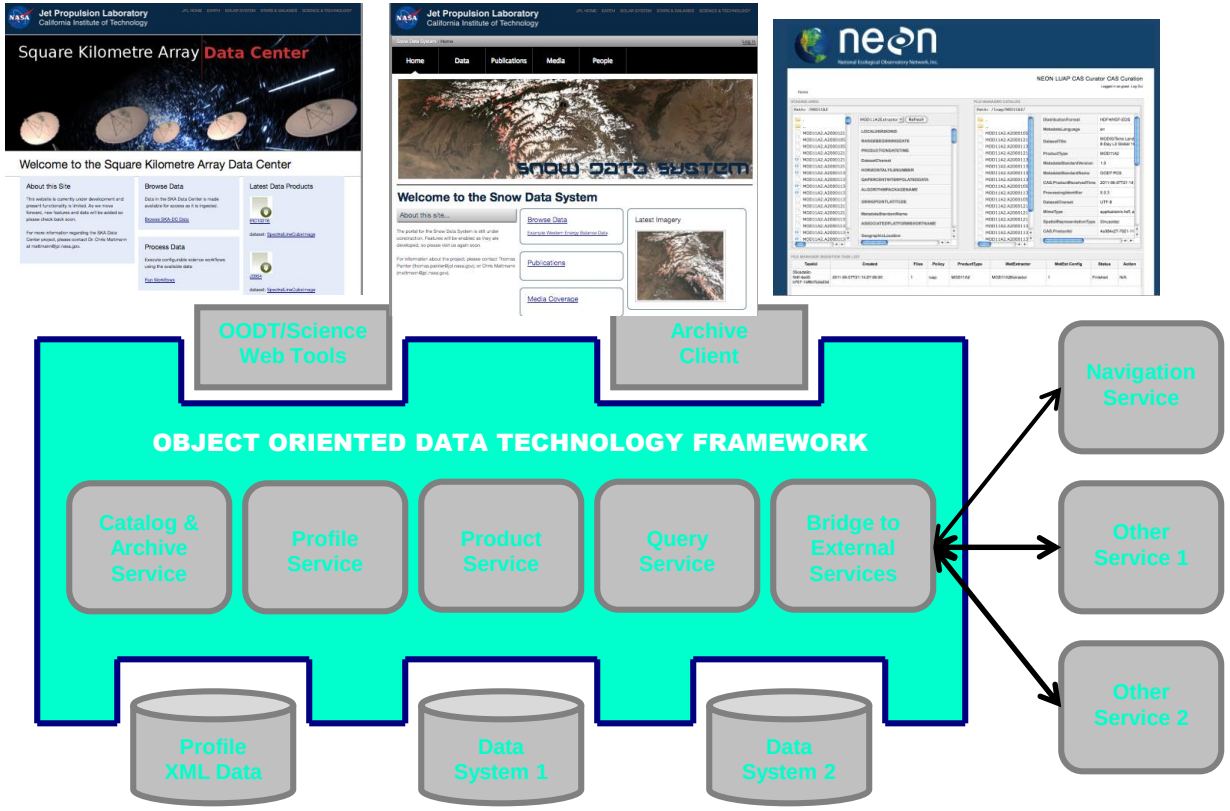
Apache Airavata is a workflow framework to enable the scheduling and execution of jobs and workflows on distributed computing resources, including local resources, national computing resources, and computing clouds.

Four main components:

- workflow suite
- application wrapper service
- registry service
- message broking service

Airavata evolved out of work at Indiana University's Extreme Computing Lab, originally supporting the Linked Environments for Atmospheric Discovery (LEAD) project. The Open Gateway Computing Environments (OGCE) project abstracted and generalized the workflow developed for LEAD, making it applicable to other science gateways. Airavata became an incubator project in 2011 and a top level project in 2012.

Apache OODT (Tools)

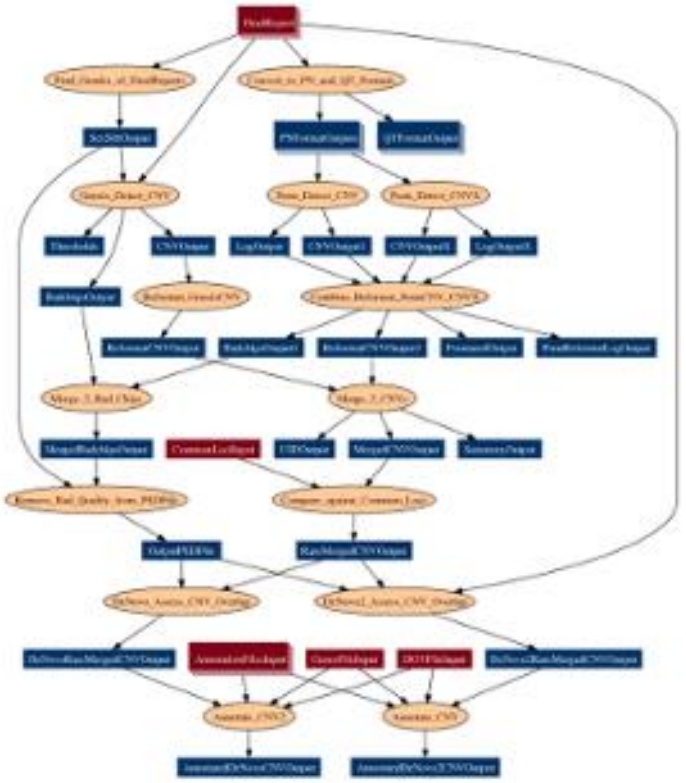


Object Oriented Data Technology, or OODT, http://oodt.apache.org/ develops and promotes science data management and archiving systems that span scientific disciplines and enable interoperability among data agnostic systems in the fields of astrophysics, planetary, space science data systems, open source web analytics, etc. It was based on NASA JPL software

http://www.slideshare.net/chrismattmann/a-look-into-the-apache-oodt-ecosystem

Pegasus

https://pegasus.isi.edu/



"Grid" workflow engine capable of executing tasks connected by data flow graphs as shown

The Pegasus project encompasses a set of technologies that help workflow-based applications execute in a number of different environments including desktops, campus clusters, grids, and clouds.

Pegasus bridges the scientific domain and the execution environment by automatically mapping high-level workflow descriptions onto distributed resources.

It automatically locates the necessary input data and computational resources necessary for workflow execution.

Pegasus enables scientists to construct workflows in abstract terms without worrying about the details of the underlying execution environment or the particulars of the low-level specifications required by the middleware (Condor, Globus, or Amazon EC2).

Pegasus also bridges the current cyber-infrastructure by effectively coordinating multiple distributed resources.

Kepler, BioKepler

The Kepler Project (UC Davis, UC Santa Barbara, and UC San Diego) https://kepler-project.org/ is dedicated to furthering and supporting the capabilities, use, and awareness of the free and open source, scientific workflow application, Kepler.

Kepler is designed to help scientists, analysts, and computer programmers create, execute, and share models and analyses across a broad range of scientific and engineering disciplines.

Kepler can operate on data stored in a variety of formats, locally and over the internet, and is an effective environment for integrating disparate software components, such as merging "R" scripts with compiled "C" code, or facilitating remote, distributed execution of models.

Using Kepler's graphical user interface, users simply select and then connect pertinent analytical components and data sources to create a "scientific workflow"—an executable representation of the steps required to generate results.

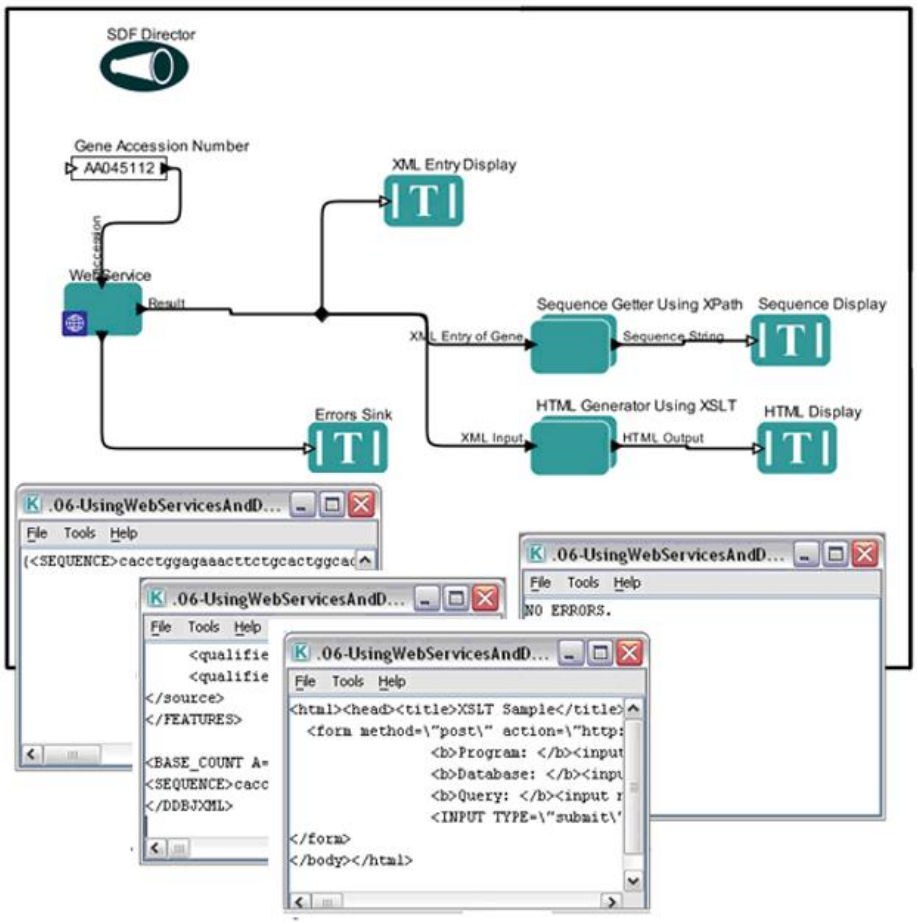
The Kepler software helps users share and reuse data, workflows, and components developed by the scientific community to address common needs.

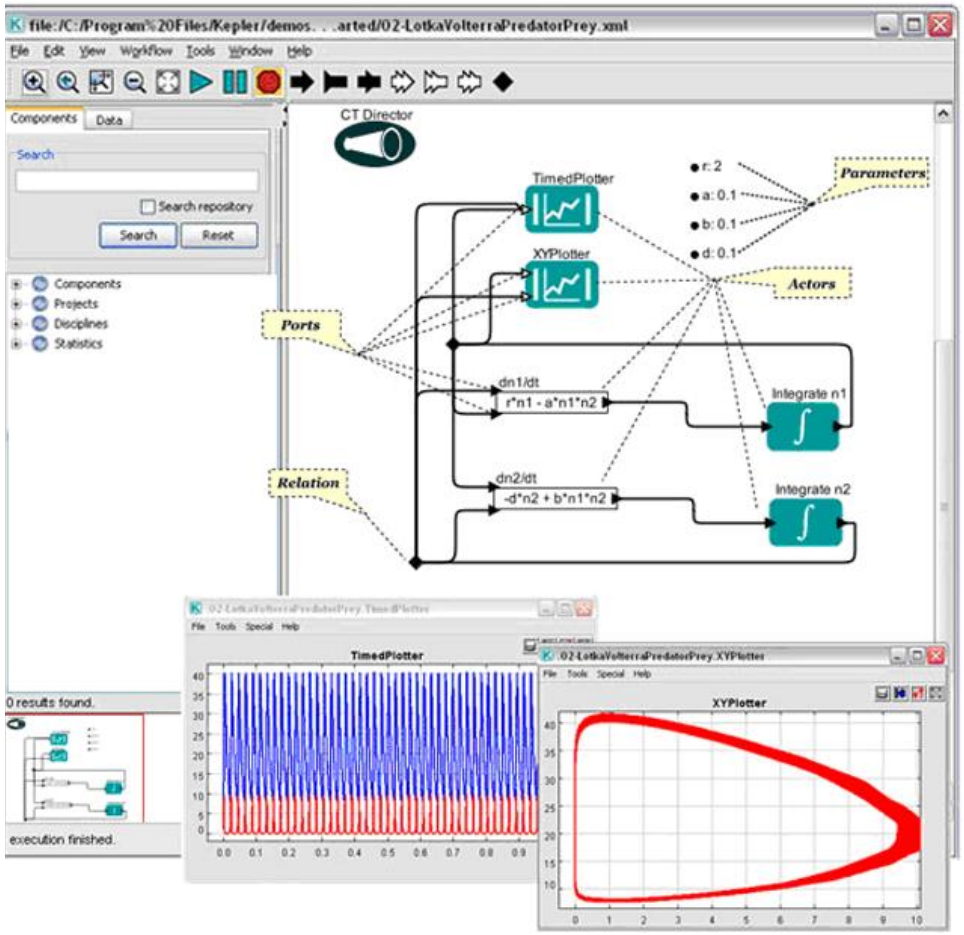
Kepler is a java-based application that is maintained for the Windows, OSX, and Linux operating systems.

http://www.biokepler.org/ bioKepler is a Kepler module of scientific workflow components to execute a set of bioinformatics tools

Kepler Example Workflows

https://kepler-project.org/users/sample-workflows





Galaxy

http://galaxyproject.org/Galaxy
http://en.wikipedia.org/wiki/Galaxy_(computational_biology) is an open source scientific workflow system written in Python.

Workflow systems provide a means to build multi-step computational analyses akin to a recipe. They typically provide a graphical user interface for specifying what data to operate on, what steps to take, and what order to do them in.

Galaxy is also a data integration platform for biological data. It supports data uploads from the user's computer, by URL, and from many online resources (such as the UCSC Genome Browser, BioMart and InterMine).

Galaxy supports a range of widely used biological data formats, and translation between those formats. Galaxy provides a web interface to many text manipulation utilities, enabling researchers to do their own custom reformatting and manipulation without having to do any programming.

Galaxy Objects include Histories are computational analyses (recipes) run with specified input datasets, computational steps and parameters. Histories include all intermediate and output datasets as well. Datasets, workflows and pages are other objects

Swift

Apache license http://swift-lang.org/main/ is a scripting language approach to workflow as in

```
foreach protein in proteinList {
        runBLAST(protein);
}
```

This is called Many task computing

Swift is easy: Short, simple scripts can do large-scale work. The same script runs on multicore computers, clusters, grids, clouds, and supercomputers.

Swift is parallel: it runs multiple programs concurrently as soon as their inputs are available, reducing the need for complex parallel programming.

Taverna

Taverna http://www.taverna.org.uk/ http://en.wikipedia.org/wiki/Taverna_workbench is an LGPL open source and domain-independent Workflow Management System – a suite of tools used to design and execute scientific workflows and aid in silico experimentation.

Well developed and robust system linking to tools like Bioconductor

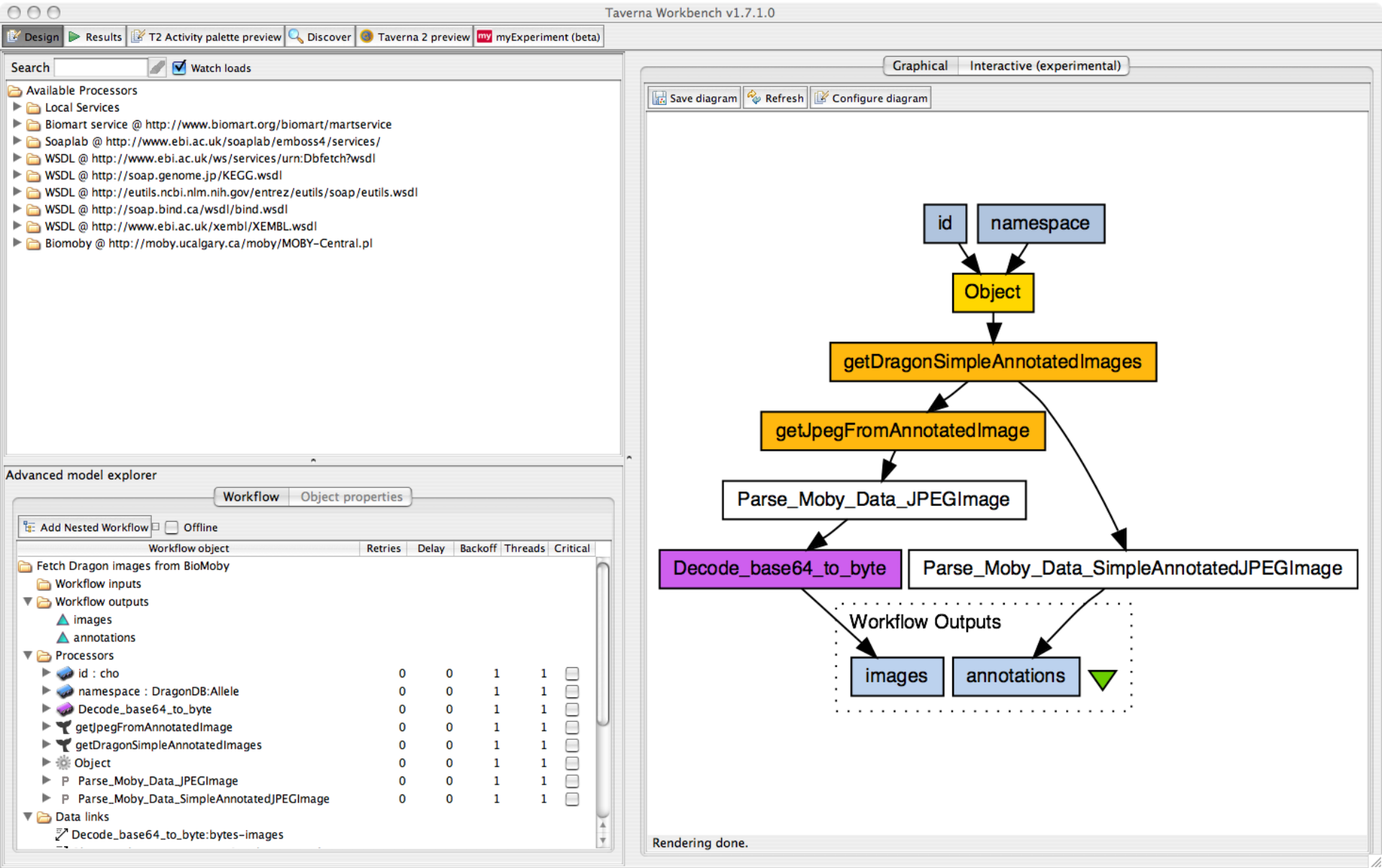
Various service types available: WSDL-style and RESTful Web services, BioMart, BioMoby, SoapLab, R, Beanshell, Excel and csv spreadsheets

Taverna has been created by the myGrid team and is currently funded though FP7 projects BioVeL, SCAPE and Wf4Ever.

The Taverna tools include the Workbench (desktop client application), the Command Line Tool (for a quick execution of workflows from a terminal), the Server (for remote execution of workflows) and the Player (Web interface plugin for submitting workflows for remote execution). Taverna Online lets you create Taverna workflows from a Web browser.

Also featured in http://www.myexperiment.org/ workflow sharing environment

Taverna


Taverna Workbench v1.7.1.0
Design
Results
T2 Activity palette preview
Discover
Taverna 2 preview
myExperiment (beta)
Search
Watch loads
Available Processors
Local Services
Biomart service @ http://www.biomart.org/biomart/martservice
Soaplab @ http://www.ebi.ac.uk/soaplab/emboss4/services/
WSDL @ http://www.ebi.ac.uk/ws/services/urn:Dbfetch?wsdl
WSDL @ http://soap.genome.jp/KEGG.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/xembl/XEMBL.wsdl
Biomoby @ http://moby.ucalgary.ca/moby/MOBY-Central.pl
Advanced model explorer
Workflow
Object properties
Add Nested Workflow
Offline
Workflow object
Retries
Delay
Backoff
Threads
Critical
Fetch Dragon images from BioMoby
Workflow inputs
Workflow outputs
images
annotations
Processors
id : cho
namespace : DragonDB:Allele
Decode_base64_to_byte
getJpegFromAnnotatedImage
getDragonSimpleAnnotatedImages
Object
Parse_Moby_Data_JPEGImage
Parse_Moby_Data_SimpleAnnotatedJPEGImage
Data links
Decode_base64_to_byte:bytes-images
Graphical
Interactive (experimental)
Save diagram
Refresh
Configure diagram
id
namespace
Object
getDragonSimpleAnnotatedImages
getJpegFromAnnotatedImage
Parse_Moby_Data_JPEGImage
Decode_base64_to_byte
Parse_Moby_Data_SimpleAnnotatedJPEGImage
Workflow Outputs
images
annotations
Rendering done.

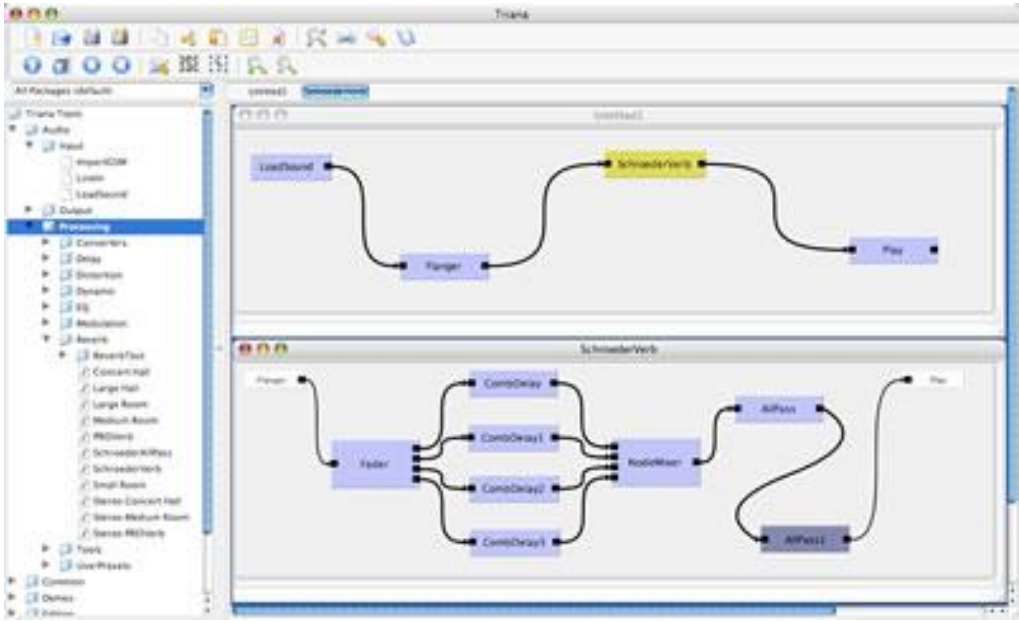
Triana

Triana Apache license http://www.trianacode.org/ from Cardiff termed a problem solving environment – an "old fashioned name"

It was one of two testbed applications for GridLab, a large EU funded project. The aim of GridLab was to develop a simple and robust grid application toolkit (GAT) enabling applications to exploit the power of the Grid.

TrianaCloud uses RabbitMQ to link modules together

- Triana is a Java system with one implementation based on the Sun's JXTA protocols and distributed across many mobile devices
- Generally built around distributed modules
- Triana has a traditional GUI interface for linking components together
- It can link to Pegasus



Trident

Apache license https://www.microsoft.com/mscorp/tc/trident.mspx
http://tridentworkflow.codeplex.com/ Scientific Workflow Workbench is no longer active

Trident is a set of tools based on the Windows Workflow Foundation that addresses scientists' need for a flexible, powerful way to analyze large, diverse datasets. It includes graphical tools for creating, running, managing, and sharing workflows. This central control present in many workflow systems but not obviously good.

Trident Workflow Composer provides graphical tools for creating workflows.

For large data sets, Trident can run multiple workflows in parallel on a Windows HPC Server 2008 cluster. It had "classic mistake" of a centralized controller

Trident Workflow Application - available in Windows Presentation Foundation and Silverlight versions - provides a simple way to run Trident workflows. The Microsoft Silverlight version of the Trident workflow application enables users to run workflows remotely using a Silverlight-enabled browser via Web services.

Trident is integrated with myExperiment - a workflow collaboration portal - so scientists can easily share their Trident workflows with colleagues.

Trident security model supports users and roles that allows scientists to control access rights to their workflows.

The Trident Registry maintains libraries of workflows and activities, workflow inputs and outputs, workflow versions, and associated data products; each user can have a personal workflow library and can share workflows with other users.

Trident Management Studio manages the Trident Registry and workflow execution, schedules workflows, and monitors workflow executions locally or remotely.

IPython

- Scripting as in Swift is an important approach to workflow/orchestration
- Several projects have used Python as the scripting language as have related (in nature of problem) System specification languages like Ansible
- https://wiki.openstack.org/wiki/NovaOrchestration/WorkflowEngines
- IPython notebook preserves provenance of activity
- Taverna can be executed from inside Ipython
  http://www.pro-ibiosphere.eu/news/4654_the%20running%20of%20taverna%20workflows%20within%20an%20ipython%20notebook/

Previous workflow systems come from Grid community although they have been adapted to clouds

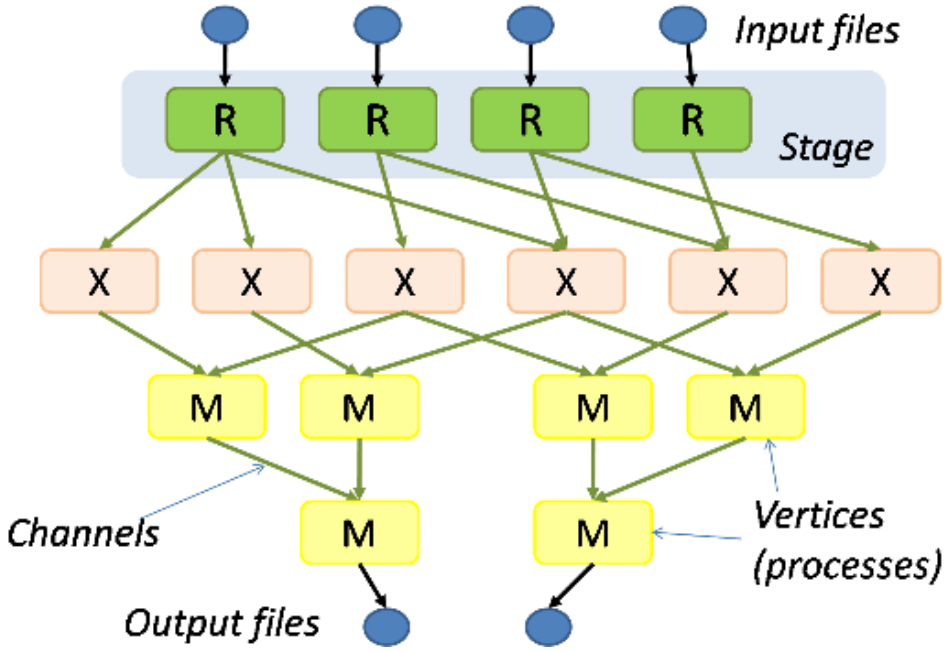
Following systems are from more recent cloud specific goals

Microsoft Dryad

http://research.microsoft.com/en-us/projects/dryad/

A Dryad programmer writes several sequential programs and connects them using one-way channels. The computation is structured as a directed graph: programs are graph vertices, while the channels are graph edges. A Dryad job is a graph generator which can synthesize any directed acyclic graph. These graphs can even change during execution, in response to important events in the computation.

Dryad is quite expressive. It completely subsumes other computation frameworks, such as Google's map-reduce, or the relational algebra. Moreover, Dryad handles job creation and management, resource management, job monitoring and visualization, fault tolerance, re-execution, scheduling, and accounting.



Microsoft Naiad

Open Source http://microsoftresearch.github.io/Naiad/ http://research.microsoft.com/en-us/projects/naiad/

http://research.microsoft.com/apps/pubs/?id=201100

A new computational model, timely dataflow, underlies Naiad and captures opportunities for parallelism across a wide class of algorithms. This model enriches dataflow computation with timestamps that represent logical points in the computation and provide the basis for an efficient, lightweight coordination mechanism.

Many powerful high-level programming models can be built on Naiad's low-level primitives, enabling such diverse tasks as streaming data analysis, iterative machine learning, and interactive graph mining. Naiad outperforms specialized systems in their target application domains, and its unique features enable the development of new high-performance applications.

Apache Tez

http://hortonworks.com/hadoop/tez/

Related to Llama (Yarn to Impala) http://cloudera.github.io/llama/

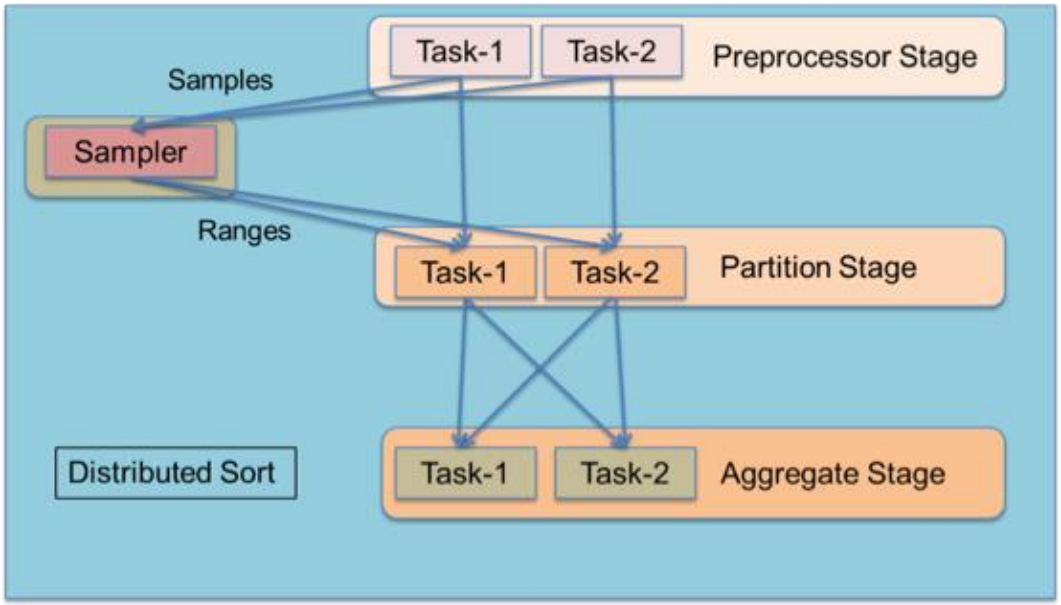
Tez from Hortonworks adds general workflow capabilities to Hadoop as seen earlier in MIcrosoft Dryad.

Tez models data processing as a dataflow graph with vertices in the graph representing application logic and edges representing movement of data.

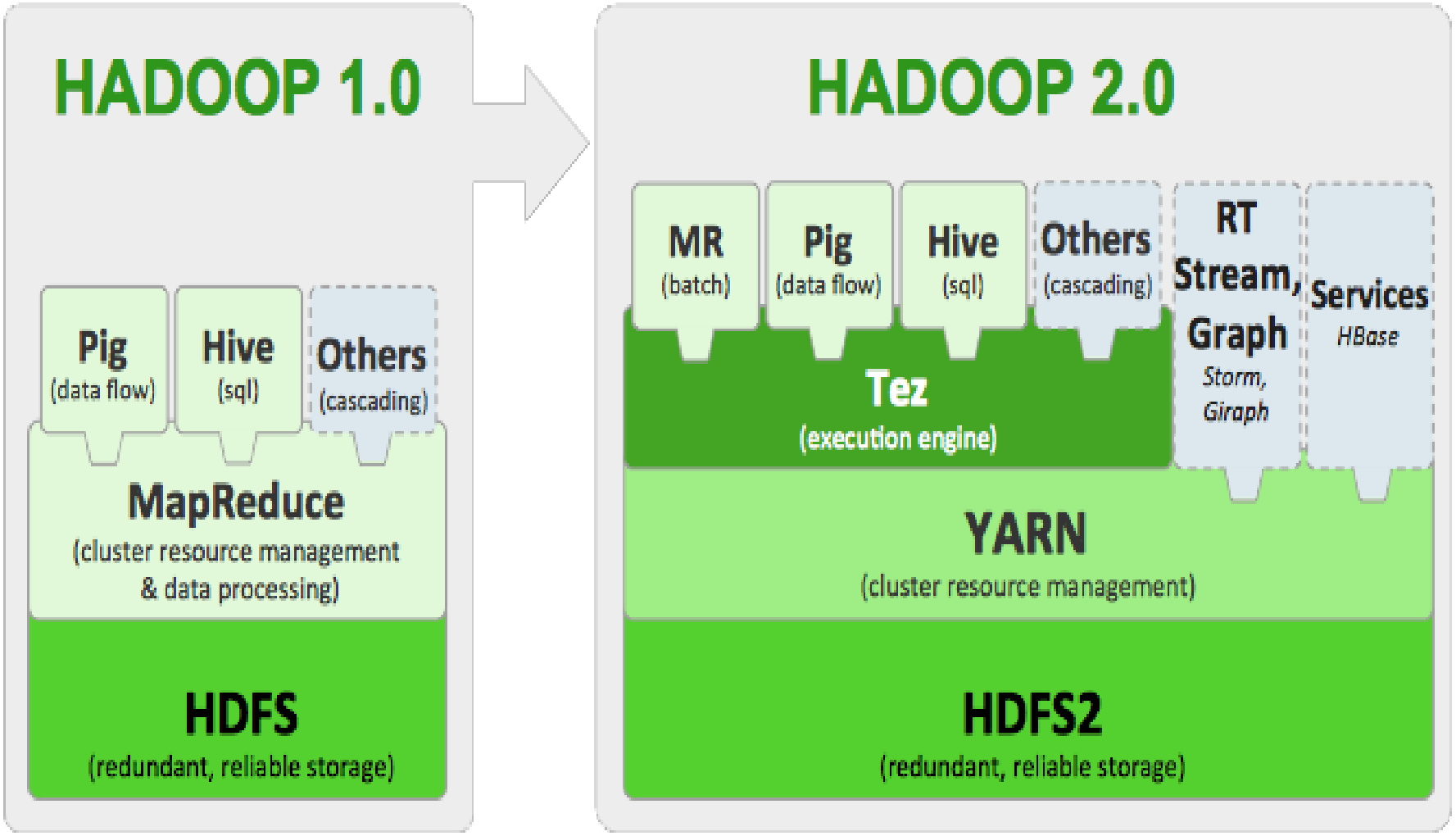
Built to work with Yarn in mixed workload clusters

http://hortonworks.com/blog/apache-tez-a-new-chapter-in-hadoop-data-processing/

A rich dataflow definition API allows users to express complex query logic in an intuitive manner and it is a natural fit for query plans produced by higher-level declarative applications like Hive and Pig. As an example, the diagram shows how to model an ordered distributed sort using range partitioning.



Apache Tez


HADOOP 1.0
Pig
(data flow)
Hive
(sql)
Others
(cascading)
MapReduce
(cluster resource management
& data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MR
(batch)
Pig
(data flow)
Hive
(sql)
Others
(cascading)
Tez
(execution engine)
RT
Stream,
Graph
Storm,
Giraph
Services
HBase
YARN
(cluster resource management)
HDFS2
(redundant, reliable storage)

Google FlumeJava

http://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/FlumeJava.pdf

FlumeJava is a Java library developed at Google to develop, test and run large scale data parallel pipelines in an efficient manner.

The data pipelines are specified using set of parallel operations available in the library. The library abstracts how the data is presented as in-memory or as file.

The data pipeline and the processing logic is written in Java. The library abstracts how the data processing happens, i.e weather local loop or map reduce job.

At the runtime these parallel operations are run as Map tasks, Reduce tasks, streaming computations etc.

FlumeJava uses defered evaluation to optimize the data flow between the parallel operations.

Google claims that they no longer uses direct Map Reduce implementation and instead they use FlumeJava to run their data parallel tasks

The project is not open source and is planned to be available to general public through Google Cloud platform as a SaaS

Part of Google Cloud Dataflow that also has Google Pub-Sub and Google MillWheel

Apache Crunch

https://crunch.apache.org/ runs on Hadoop or Spark

The Apache Crunch project develops and supports Java APIs that simplify the process of creating data pipelines on top of Apache Hadoop. The Crunch APIs are modeled after Google FlumeJava

One can compare with Apache Pig, Apache Hive, and Cascading.

Developer focused. Apache Hive and Apache Pig were built to make MapReduce accessible to data analysts with limited experience in Java programming. Crunch was designed for developers who understand Java and want to use MapReduce effectively in order to write fast, reliable applications that need to meet tight SLAs. Crunch is often used in conjunction with Hive and Pig; a Crunch pipeline written by the development can be processed by a diverse collection of Pig scripts and Hive queries written by analysts.

Minimal abstractions. Crunch pipelines provide a thin veneer on top of MapReduce. Developers have access to low-level MapReduce APIs whenever they need them. This mimimalism also means that Crunch is extremely fast, only slightly slower than a hand-tuned pipeline developed with the MapReduce APIs, and the community is working on making it faster all the time.

- One of the goals of the project is portability, and the abstractions that Crunch provides are designed to ease the transition from Hadoop 1.0 to Hadoop 2.0 and to provide transparent support for future data processing frameworks that run on Hadoop, including Apache Spark and Apache Tez.

Flexible Data Model. Hive, Pig, and Cascading all use a tuple-centric data model that works best when your input data can be represented using a named collection of scalar values, much like the rows of a database table.

- Crunch allows developers considerable flexibility in how they represent their data, which makes Crunch the best pipeline platform for developers working with complex structures like Apache Avro records or protocol buffers, geospatial and time series data, and data stored in Apache HBase tables.

Cascading - Twitter Scalding

http://www.cascading.org/ open source with many subsidiary projects such as Twitter PyCascading (Python) and Scalding (Scala)

Java tuple data model

From Concurrent http://www.concurrentinc.com/ that offers commercial support

Supports Hadoop, Hive with Storm, Spark, Tez to come

It follows a 'source-pipe-sink' paradigm, where data is captured from sources, follows reusable 'pipes' that perform data analysis processes, where the results are stored in output files or 'sinks'.

- Pipes are created independent from the data they will process. Once tied to data sources and sinks, it is called a 'flow'.
- These flows can be grouped into a 'cascade', and the process scheduler will ensure a given flow does not execute until all its dependencies are satisfied.
- Pipes and flows can be reused and reordered to support different business needs.

e-Science Central

http://www.esciencecentral.co.uk/

e-Science Central is a Science-as-a-Service platform that combines three emerging technologies — Software as a Service (so you only need a web browser to do your science), Social Networking (to encourage you to interact and create your communities) and Cloud Computing (to provide you with storage and computational power).

Using only a browser, you can upload your data, share it in a controlled way with your colleagues, and analyse the data using either a set of pre-defined services, or your own, which you can upload for execution and sharing. You can also record your progress in notebooks and publish your work on-line or conventionally.

- Moreover, e-Science Central gives you a workflow editing and enactment tool to allow you automation of analysis through the browser.
- Azure is typically backend cloud