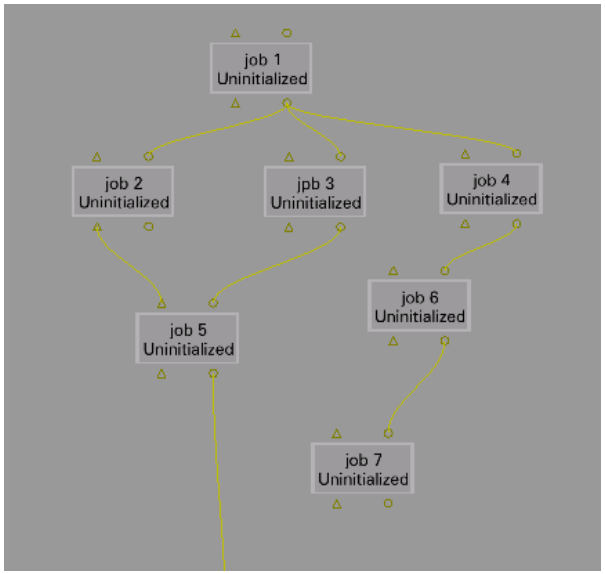
pomsets: Workflow management for your cloud

Michael J Pan
Nephosity

20 April, 2010

Workflow management is ...

the design, specification, coordination of the execution of tasks and task dependencies.



Motivation

We have lots of data and compute nodes to process that data. To minimize execution time, we need a tool to

- design and specify of task parallelism and task dependency ordering
- coordinate execution of the tasks over large compute resources
- enable workflow reuse over multiple datasets

Why workflow management + cloud computing?

- Cloud computing provides the ability to scale compute resources with the work that needs to be done
- Better than what is available today, i.e. WFM+grid computing
- WFM is critical to a successful long-term cloud computing strategy
  - A critical component of the cloud computing software stack
  - Significant cloud computing community desire for WFM functionalities
  - Identification by scientific community as essential technologies

Workflow management + grid computing

Large computing resources historically available as grid computing.
Issues with WFM on grids

- Jobs submitted to grids are often queued up behind jobs of other users, reduces the effectivity of workflow management optimizations
- Hetereogeneous compute environments may result in different task results and/or make the workflow specification unnecessarily complex
- Grids are not easily federated, limiting burst computing
- Available only to institutions with the resources to deploy their own grid, as well as implement their own WFM
- WFM systems developed specifically for particular grids, subject to their idiosyncrasies and not generalizable

Components of a cloud computing software stack

- virtual machines (VMWare, Xen, Virtuzzo, KVM)
- dynamic provisioning (Amazon EC2, Eucalyptus, GoGrid, Rackspace, Dell/Joyent)
- task partitions (MapReduce, Hadoop, Disco, Sphere)
- data distribution (GFS, HDFS, Ceph, Sector, Voldemort, MongoDB, CouchDB)
- unified messaging (Qpid, RabbitMQ, Amazon SNS)
- workflow management (Azkaban, Kepler, Oozie, Pipeline, Pegasus, Taverna, Triana, pomsets)
- monitoring & reporting (RightScale, Nagios, Ganglia, Graphite)

Significant community demand

(screencap 2009-12-04, currently 57 watchers)

Home About Community Training Support Distribution Blog

Hadoop Development Status

Report generated on 2009-12-04 19:43:21 by cron

cloudera

JIRA Issues | Mailing List | Contributors

Most Watched Issues

Hadoop uses JIRA to track bugs and tasks. Anyone can sign up for a JIRA account and watch issues; sign up here to get involved in the community! The following table shows the most watched open tickets in Hadoop core.

Download CSV

Name	Type	Status	Component	Watchers
[HADOOP-5303] Oozie, Hadoop Workflow System	New Feature	Patch Available	None	47
[HADOOP-5670] Hadoop configurations should be read from a distributed system	New Feature	Open	conf	25
[HADOOP-3421] Requirements for a Resource Manager for Hadoop	New Feature	Open	None	25
[HADOOP-4487] Security features for Hadoop	New Feature	Open	security	24
[HADOOP-2560] Processing multiple input splits per mapper task	Bug	Open	None	23
[HADOOP-6332] Large-scale Automated Test Framework	New Feature	Open	test	22
[HADOOP-5071] Hadoop 1.0 Compatibility Requirements	Sub-task	Open	None	21
[HADOOP-4998] Implement a native OS runtime for Hadoop	New Feature	Open	native	19

Identification by the scientific community

"Beyond the Data Deluge"
(Science, Vol. 323. no. 5919, pp. 1297-1298, 2009)

> *In the future, the rapidity with which any given discipline advances is likely to depend on how well the community acquires the necessary expertise in database,* **workflow management**, *visualization, and* **cloud computing** *technologies.*

Challenges with workflow management

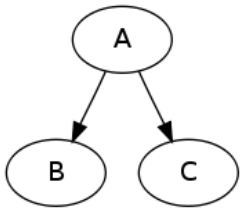
- Ability to handle the various workflow structures
- Ease of use
- Others that we will not cover, including, but not limited to
  - data management and distribution
  - validation of data (both inputs and outputs)
  - data provenance
  - command versioning

Workflow structures

- Fan out
- Fan in
- Diamond
- Intermediary
- N
- Task partitioning (Parameter sweep, MapReduce)

What do they look like, in a dependency graph, and when linearized (coded into a script)? What issues do they present?

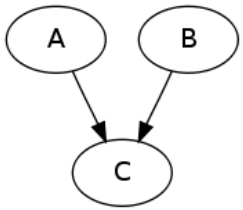
Fan out



- A; B; C
- A; C; B

Completion of a task needs to notify all its successors.

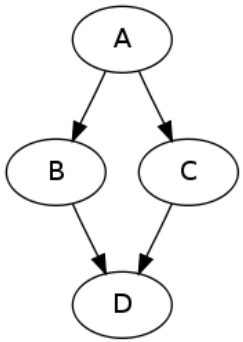
Fan in



- A; B; C
- B; A; C

A task cannot start until all its predecessors have completed.

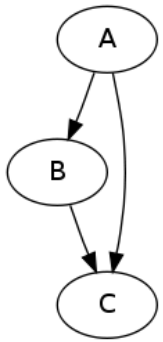
Diamond



- A; B; C; D
- A; C; B; D

Combination of "fan in" and "fan out". Need to ensure that D is not run twice.

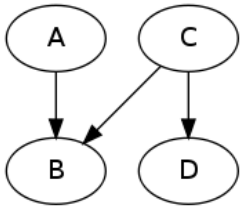
Intermediary



- A; B; C

Another variation of combination "fan in" and "fan out". Need to ensure that C is not run twice.

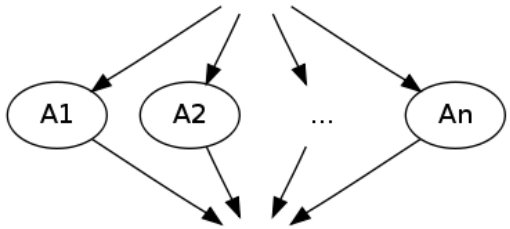
N



- A; C; B; D
- A; C; D; B
- C; A; B; D
- C; A; D; B
- C; D; A; B

Another variation of combination "fan in" and "fan out". Computational linguistics theory: N structures in a pomset grammer increases the degree of context sensitivity of the grammar.

Task partitioning



- A_1; A_2; ...; A_n

Issues

- Dynamic generation of task partitions
- Combination of task partitioning with previous structure types

Pig

IMHO, not very user friendly.

```
-- Query 1
A = load 'myfile' as (t, u, v);
B = load 'myotherfile' as (x, y, z);
C = filter A by t == 1;
D = join C by t, B by x;
E = group D by u;
F = foreach E generate group, COUNT($1);

-- Query 2
A = load 'myfile' as (t, u, v);
B = load 'myotherfile' as (x, y, z);
C = join A by t, B by x;
D = group C by u;
E = foreach D generate group, COUNT($1);
F = filter E by C.t == 1;
```

Oozie

IMHO, not very user friendly.

```
<action name="myPigjob" type="pig">
  <task name="pig" >
   <inputs>
     <dataset url="s3:/traffic/oystercard/london/2009" />
     <dataset url="s3:/traffic/anpr/london/2009/" />
     <file path="/users/steve/pig/car-vs-tube.pig" />
   </inputs>
   <option name="limit" value="40000" />
   <option name="pigfile" value="/users/steve/pig/car-vs-tube.pig" />
   <option name="startdate" value="2009-01-01" />
   <option name="enddate" value="2009-02-28" />
  <outputs>
    <dataset url="hdfs://traffic/anpr/london" />
  </outputs>
  </task>
</action>
```

Usability Hypothesis

All things being equal (i.e. functionality), the product easiest to use becomes dominant

- Search and mail: Google
- Phone and tablet: Apple
- Social networking: Facebook

Usability goals

- Visual: no user coding
- Simple: easy enough for non-programmers to design their workflows and to execute workflows on existing clouds
- Powerful: capable of specifying dependencies, task partitions, etc. if desired by user, but not overwhelm user by default

pomsets is ...

- a mathematical model– first used in 1985 by Vaughn Pratt to describe concurrent processes
- an application that implements the mathematical model as the data structures that represent workflow components, facilitates the design and specification of workflows, and coordinates the execution of the workflows on cloud deployments.

The mathematical model

A labelled partial order is a 4 tuple (V, Σ, $\preceq$, μ) where

- V is a set of vertices
- Σ is the alphabet
- $\preceq$ is the partial order on the vertices
- μ is the labelling function μ: V $\rightarrow \Sigma$

A pomset (partially ordered multiset) is a LPO up to isomorphism.

The workflow management system

Two main components

- the core is the backend and provides an API
- the GUI is the front end and interacts with the user

Features

- Parallel computing
- Data flow
- Flow control
- Workflow reusability
- Compute cloud agnosticism
- Execute environment agnosticism
- MapReduce
- Intuitive GUI
- Simple API

Demo

Target users

- end users who have workflows that they run repetitively over different datasets
- subject matter experts who design workflows to share with their colleagues/collaborators
- developers who develop programs to be executed as workflow tasks
- developers who explicitly define workflows that their application executes

Future work

Apply workflow management to applications in various domains; make improvements as necessary

- rendering, animation, special effects
- medical imaging
- scientific computing
- text processing

Getting to know pomsets

http://pomsets.org

- Current release is 1.0.6
- Download source
- Download Mac OS X application bundle
- Prepackage binaries for other platforms soon
- Sign up on the user and/or announcement groups