OWL

Francesco Pierfederici

History

- Need to have a Workflow Management System for JWST
- Completed a trade study to identify promising technology
- OWL born as prototype to inform the trade study

OWL = Condor + Blackboard + Dynamic Workflows

OWL = Condor + Blackboard + Dynamic Workflows

"Condor is an open source high-throughput computing software framework for coarse-grained distributed parallelization of computationally intensive tasks".

Wikipedia

History

- Established in 1985
- ~35 faculty/staff/students + external collaborators
- Open source
- Used in production across disciplines, research, industry...

Condor World Map

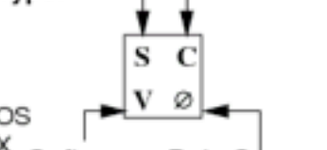
http://www.cs.wisc.edu/condor/map

Tue Aug 9 19:00:29 CDT 2011

SWEDEN
FINLAND
NORWAY
LATVIA
GERMANY
RUSSIA
AUSTRIA
HUNGARY
ITALY
SLOVENIA
BULGARIA
RUMANIA
GREECE
POLAND
ISRAEL
TURKEY
IRAN
JORDAN
CZECH REPUBLIC
CHINA
SOUTH KOREA
JAPAN
INDIA
OMAN
HONG KONG
MALAYSIA
SINGAPORE
INDONESIA
CONGO
PERU
BRAZIL
CHILE
ARGENTINA

System Type:

A	AIX
D	DUX
M	OSX
S	SUNOS
H	HPUX
I	IRIX
L	Linux
W	Windows



CPU Type:

I	Intel x86
6	Intel IA64
A	Alpha
P	Power PC
S	SPARC
H	HPPA
M	MIPS

Size of Pool:

- > 1000 hosts
- 100–999 hosts
- 10–99 hosts
- 1–9 hosts

Overview

- Flexible scheduling
- Workflow management (DAGMan)
- Several job types (serial, parallel, VM etc.)
- Fault-tolerance
- No shared file-system required
- APIs (SOAP, DRMAA, GAHP, command-line)
- Cross-platform (Linux, Mac OS X, Windows etc.)
- Able to federate different GRID/Cloud technologies
- Dynamic resource provisioning (e.g. via VMs)
- Scavenge idle cycles

OWL = Condor + Blackboard + Dynamic Workflows

- Written in Python (2 dependencies)
- Thin, portable layer on top of Condor

OWL = Condor + Blackboard + Dynamic Workflows

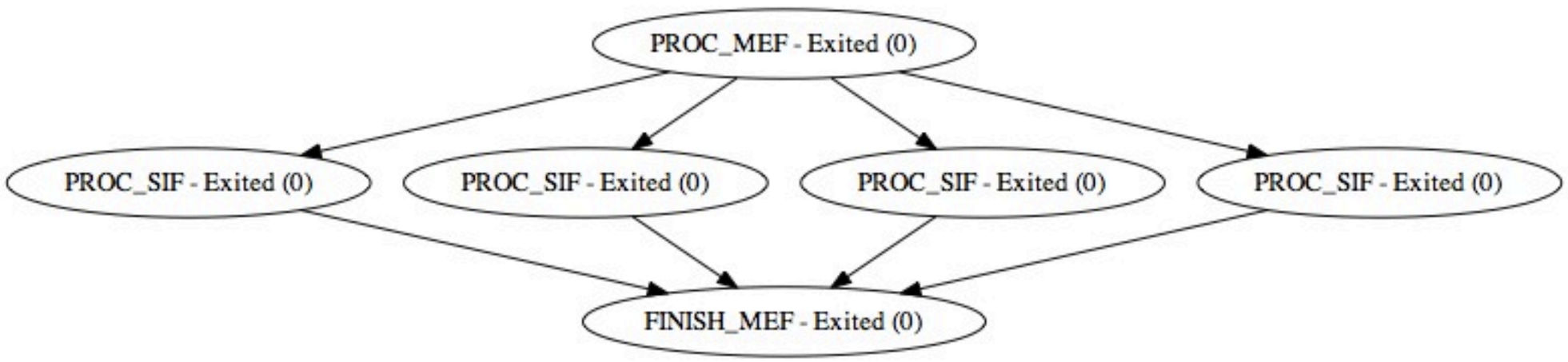
- Persisted in SQL Server
- Process view
- Dataset view

Request #	Dataset	User	Start Time	Process Name	Parent Names	State	Duration (s)	Exit Code	Detail	Hold/Release
405	raw-000002	apache	2011-08-10 13:08:53.0000000	FINISH_MEF	PROC_SIF	Exited	5.04511499405	0	More Info	
405	raw-000002	apache	2011-08-10 13:07:53.0000000	PROC_SIF	PROC_MEF	Exited	30.0510444641	0	More Info	
405	raw-000002	apache	2011-08-10 13:07:53.0000000	PROC_SIF	PROC_MEF	Exited	30.0509109497	0	More Info	
405	raw-000002	apache	2011-08-10 13:07:53.0000000	PROC_SIF	PROC_MEF	Exited	30.0447692871	0	More Info	
405	raw-000002	apache	2011-08-10 13:07:53.0000000	PROC_SIF	PROC_MEF	Exited	30.0438938141	0	More Info	
405	raw-000002	apache	2011-08-10 13:07:13.0000000	PROC_MEF		Exited	10.0446805954	0	More Info	
401	raw-000002	apache	2011-08-09 20:16:26.0000000	FINISH_MEF	PROC_SIF	Exited	5.04631710052	0	More Info	
401	raw-000002	apache	2011-08-09 20:15:46.0000000	PROC_SIF	PROC_MEF	Exited	30.0435886383	0	More Info	
401	raw-000002	apache	2011-08-09 20:14:46.0000000	PROC_SIF	PROC_MEF	Exited	30.0539417267	0	More Info	
401	raw-000002	apache	2011-08-09 20:14:46.0000000	PROC_SIF	PROC_MEF	Exited	30.0502624512	0	More Info	
401	raw-000002	apache	2011-08-09 20:14:46.0000000	PROC_SIF	PROC_MEF	Exited	30.0501461029	0	More Info	
401	raw-000002	apache	2011-08-09 20:14:07.0000000	PROC_MEF		Exited	10.0454416275	0	More Info	
399	raw-000002	apache	2011-08-09 20:12:25.0000000	PROC_MEF		Exited	10.047867775	1	More Info	
327	raw-000004	apache	2011-08-04 20:53:09.0000000	FINISH_MEF	PROC_SIF	Exited	5.0455827713	0	More Info	
327	raw-000004	apache	2011-08-04 20:52:09.0000000	PROC_SIF	PROC_MEF	Exited	30.041852951	0	More Info	
327	raw-000004	apache	2011-08-04 20:39:09.0000000	PROC_SIF	PROC_MEF	Exited	30.0452003479	0	More Info	
327	raw-000004	apache	2011-08-04 20:39:09.0000000	PROC_SIF	PROC_MEF	Exited	30.0434589386	0	More Info	
327	raw-000004	apache	2011-08-04 20:39:09.0000000	PROC_SIF	PROC_MEF	Exited	30.0415306091	0	More Info	
327	raw-000004	apache	2011-08-04 20:38:49.0000000	PROC_MEF		Exited	10.0458850861	0	More Info	
323	raw-000001	apache	2011-08-04 20:34:08.0000000	FINISH_MEF	PROC_SIF	Exited	5.04317903519	0	More Info	
323	raw-000001	apache	2011-08-04 20:33:09.0000000	PROC_SIF	PROC_MEF	Exited	30.0421085358	0	More Info	
323	raw-000001	apache	2011-08-04 20:28:29.0000000	PROC_SIF	PROC_MEF	Exited	30.0420684814	0	More Info	
323	raw-000001	apache	2011-08-04 20:28:28.0000000	PROC_SIF	PROC_MEF	Exited	30.041185379	0	More Info	
323	raw-000001	apache	2011-08-04 20:28:28.0000000	PROC_SIF	PROC_MEF	Exited	30.0420207977	0	More Info	
323	raw-000001	apache	2011-08-04 20:27:48.0000000	PROC_MEF		Exited	10.044462204	0	More Info	
319	raw-000001	apache	2011-08-04 20:22:49.0000000	FINISH_MEF	PROC_SIF	Exited	5.04511213303	0	More Info	
319	raw-000001	apache	2011-08-04 20:21:49.0000000	PROC_SIF	PROC_MEF	Exited	30.0414428711	0	More Info	
319	raw-000001	apache	2011-08-04 20:21:49.0000000	PROC_SIF	PROC_MEF	Exited	30.0434951782	0	More Info	
319	raw-000001	apache	2011-08-04 20:21:48.0000000	PROC_SIF	PROC_MEF	Exited	30.0416488647	0	More Info	
319	raw-000001	apache	2011-08-04 20:21:48.0000000	PROC_SIF	PROC_MEF	Exited	30.0441589355	0	More Info	
319	raw-000001	apache	2011-08-04 20:21:08.0000000	PROC_MEF		Exited	10.045668602	0	More Info	

Request #405

Dataset: raw-000002

User: apache



OWL = Condor + Blackboard + Dynamic Workflows

Templates make workflows dynamic

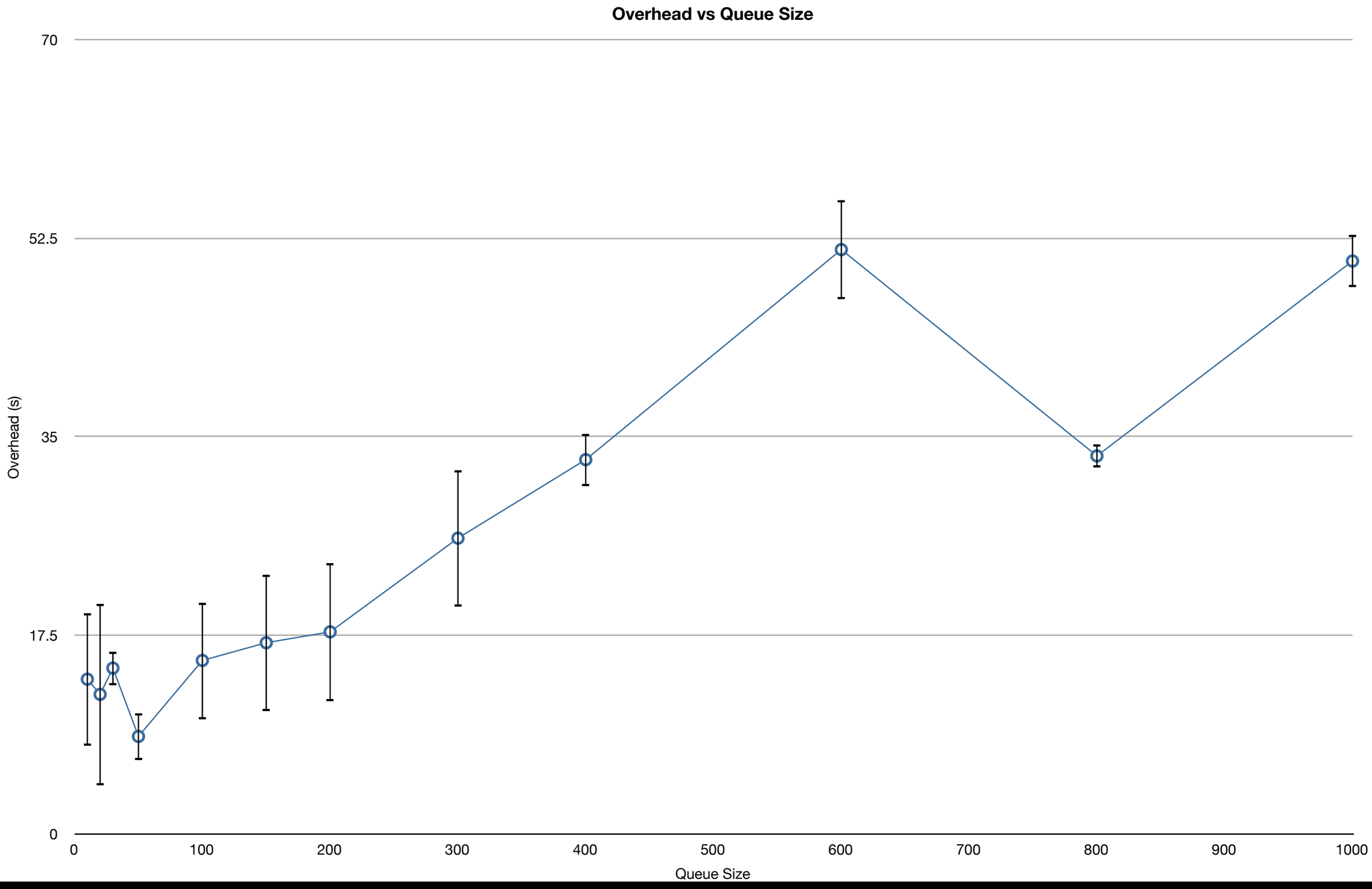
- Python template library (Jinja2)
- Support for complex Python objects
- Can embed limited logic (if...then...else, for loops)
- Can be instrument/mode/filter... specific
- Stored in the file system and/or database
- Template inheritance

```
#
# Reassemble SIFs into MEF template
#
Executable              = {{ code_root }}/finishMef.py
Arguments               = -i {{ dataset }}_calib_%(ccdId)s.fits \
                          -o {{ dataset }}_calib.fits \
                          -n {{ num_ccds }}
Output                  = finishMef_{{ dataset }}.out
Error                   = finishMef_{{ dataset }}.err
Universe                = vanilla
Log                     = {{ dataset }}.log
+InputDataset           = "{{ dataset }}"

Should_Transfer_Files   = IF_NEEDED
When_to_Transfer_Output = ON_EXIT
{% set comma = joiner(",") %}
transfer_input_files    = {% for i in range(num_ccds) %}
    {{ comma() }}{{ dataset }}_calib_{{ i }}.fits
  {% endfor %}

+HookKeyword            = "FOO"
Notification            = Never
Queue
```

Performance (Vanilla Condor)


Overhead vs Queue Size
70
52.5
35
17.5
0
Overhead (s)
0
100
200
300
400
500
600
700
800
900
1000
Queue Size