

# JuMP-dev 2025

## JuMP on demand: Creating your own compute cluster for solving optimisation problems

James Foster

- Abstract
  - I will give a brief overview of the why, what and how of solving JuMP models via an on-demand cloud-based compute cluster. Inspired by the NEOS Server for solving optimisation models as a service, we also make use of the HTCondor software system through the Google cluster toolkit, but with resource limits that can be specified by the user based on their needs and budget.
- About me:
  - Research scientist in Energy with Australia's national science agency, the Commonwealth Scientific and Industrial Research Organisation (CSIRO).
  - Research interests include energy system models, modelling languages and data, power flow formulations and convex relaxations, linear algebra, quadratically constrained and mixed-integer optimisation.

james.foster@csiro.au @jd-foster



# Background

- This talk is about:
  - challenges and techniques for scaling workflows
  - guiding those working through the same process
  - discussing the potential of a flexible JuMP-focused modelling workflow.





## **NEOS Server: State-of-the-Art Solvers for Numerical Optimization**

The **NEOS Server** is a free internet-based service for solving numerical optimization problems. Hosted by the [Wisconsin Institute for Discovery at the University of Wisconsin in Madison](#), the NEOS Server provides access to more than 60 state-of-the-art solvers in more than a dozen optimization categories. Solvers hosted by the University of Wisconsin in Madison run on distributed high-performance machines enabled by the [HTCondor software](#); remote solvers run on machines at [Arizona State University](#), the [University of Klagenfurt](#) in Austria, and the [University of Minho](#) in Portugal.

- NEOS has been a long-standing and useful service to the optimisation community.
- Web API/web submission front-end site
  - Other 3<sup>rd</sup>-party interfaces
  - see <https://github.com/jump-dev/NEOSServer.jl> !
- HTCondor: for managing and distributing the 'job queue' (problems to solve)
  - "HTC" stands for High-Throughput Computing - more later
- **Solvers** (> 60\*)
- Built and hosted on different servers at U. Wisconsin-Madison, Arizona State, U. Klagenfurt & U. Minho.
  - \*including commercial solvers that may otherwise be hard to access or obtain.

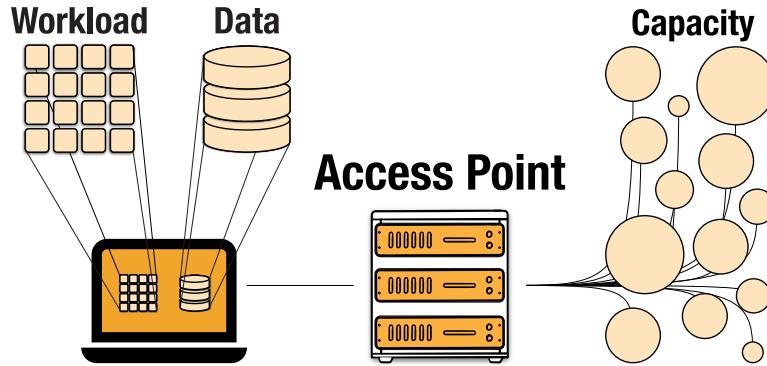
# Limitations of NEOS

- 10-15 jobs at a time
- Not all input formats are available to every solver
- Best case available memory: 3 GB of RAM
- Maximum run time: 8 hours
- Maximum of 4 threads per job
- Maximum output produced by solver: 100MB
  - "For security reasons, jobs are not allowed to read additional input files after they have been received by NEOS."
  - "In general, jobs are not allowed to write additional output files."

All these are well-founded design choices for the service.  
*But what if we want a little more?*

# High-Throughput Computing (HTC) and HTCondor

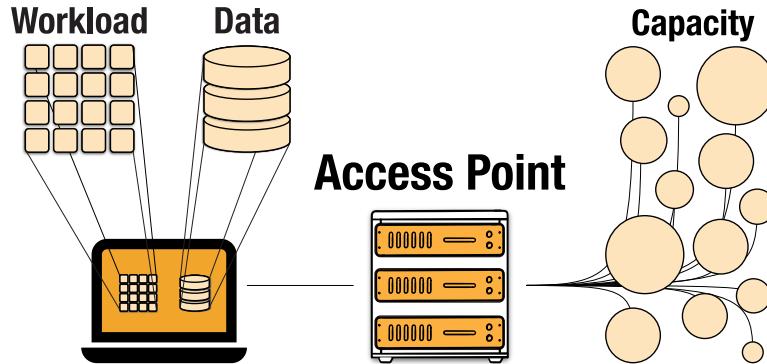
- $\text{HTC} \neq \text{HPC}$
- Designed to manage effectively up to hundreds of thousands of jobs (rather than a small number of large compute task).
- The HTCondor 'Formula':
  - "describe your computational tasks as a series of **independent, asynchronous "jobs."** [an atomic unit of work]
  - " access computational resources by submitting (or "placing") job descriptions at an HTCondor **"access point"** (AP), also known as a **"submit node."** "
  - "HTCondor locates an **appropriate machine** for each job, packages up the job and ships it off to ... **execution points (EP)**.



<https://osg-htc.org/services/access-point>

executable  
arguments  
log  
output  
error  
request\_memory  
request\_disk  
request\_cpus  
queue

= /bin/sleep  
= 4  
= simple.log  
= simple.out  
= simple.error  
= 1GB  
= 1GB  
= 4



executable	= compare_states
arguments	= wi.dat us.dat wi.dat.out
should_transfer_files	= YES
transfer_input_files	= us.dat, wi.dat
when_to_transfer_output	= ON_EXIT
log	= job.log
output	= job.out
error	= job.err
request_cpus	= 1
request_memory	= 20MB
request_disk	= 20MB
queue 5	

# 'Features' of HTCondor



- Great community supported by the Center for High Throughput Computing (CHTC) at UW-Madison.
- Training materials and high-quality documentation
  - Throughput Community Week
  - OSG Summer School



HTCondor Manual

Search docs

QUICK START GUIDES

- Users' Quick Start Guide
- Downloading and Installing Overview

REFERENCE MANUALS

- Users' Manual
- Administrators' Manual
- ClassAds
- DAGMan Workflows
- Python Bindings

DOCS

- Using
- Submitting
- Specific Information
- Topics, and Other Answers
- Survey and Release Notes

GLOSSARY AND INDEX

- Reference from past
- Topics
- Other Needed Values

Environment and services for a running job

Services for Running Jobs

HTCondor provides an environment and certain services for running jobs. Jobs can use these services to provide more reliable runs, to give logging and monitoring data for users, and to synchronize with other jobs. Note that different HTCondor job universes may provide different services. The functionality below is available in the vanilla universe, unless otherwise stated.

Environment Variables

An HTCondor job running on a specific host does not, by default, inherit its environment variables from the machine it runs on or the machine it was submitted from. If it did, the environment might change from run to run, unrelated to machine, and create non-reproducible, difficult-to-debug problems. Rather, HTCondor is deliberate about what environment variables a job sees, and allows the user to set them in the job description file.

The user may define environment variables for the job with the `environment` submit command.

Instead of defining environment variables individually, the entire set of environment variables in the `condor_submit`'s environment can be copied into the job. The `getenv` command does this.

In general, it is preferable to *per-job* to the minimum set of needed environment variables with the `environment` command, as that clearly specifies the needed environment variables. If the needed set is not known, the `getenv` command is useful. If the environment is set with both the `environment` command and `getenv` is also set to true, values specified with `environment` override values in the submitter's environment, regardless of the order of the `environment` and `getenv` commands in the submit file.

Commands within the submit description file may reference the environment variables of the submitter. Submit description file commands use `$ENV{EnvironmentVariableName}` to reference the value of an environment variable.

Extra Environment Variables HTCondor sets for Jobs

HTCondor sets several additional environment variables for each executing job that may be useful.

- `_CONDOR_SCRATCH_DIR` names the directory where the job may place temporary data files. This directory is unique for every job that is run, and its contents are deleted by HTCondor when the job stops running on a machine. When file transfer is enabled, the job is started in this directory.

<https://htcondor.readthedocs.io/en/latest/users-manual/env-of-job.html>

<https://www.ctc.cs.wisc.edu/events/2025/12/10/Bioinformatics-Cafe>

- Consistency with researcher experience through the OSG Consortium and use of the Open Science Pool



# One Researcher's Leap into Throughput Computing: Bringing Machine Learning to Dairy Farm Management

Sophie Dorros

Nov 17, 2025

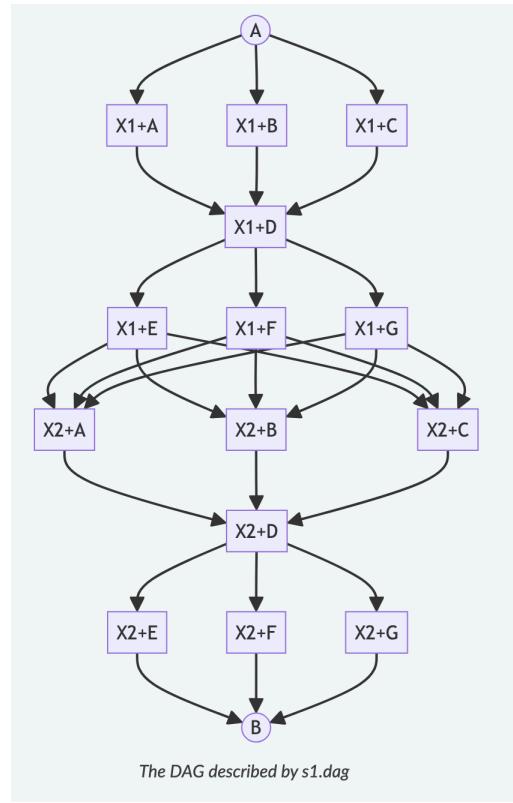
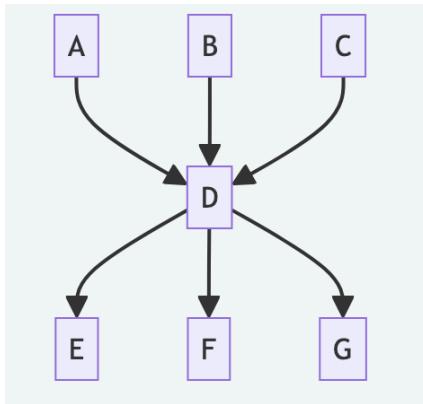
Ariana Negreiro, Ph.D, in the [UW-Madison Digital Livestock Lab](#), discusses her work using images of cows to develop a machine learning application to monitor their health.



Pictured: Ariana Negreiro credits: Ariana Negreiro

The goals for Ariana Negreiro's work were clear: healthier dairy cows with an improved quality of life. Working under the direction of her lab director [Dr. Joao Dorea](#), Negreiro's research utilizes machine learning and deep learning models to develop tools for dairy cattle management. She used an on-site processing system to produce top down, infrared and depth images of the cows, which are then processed and used to monitor the heifers' health and milk production potential. One strong advantage of the computer vision approach is that it eliminates the need to manually weigh and monitor cattle, which is

# DAGMan Workflows



# Running your own cluster

Google Cloud    Technology areas ▾    Cross-product tools ▾    Search /    Console    E

## Cluster Toolkit

Overview    **Guides**    Resources

Filter

Discover

- Product overview**
- Supported modules
- Cluster blueprint
- Cluster blueprint catalog

Quickstarts

- Deploy an HPC cluster with Slurm
- Create an AI-optimized Slurm cluster with an A4 machine type
- Create an RDMA-enabled HPC Slurm

Home > Documentation > Compute > Cluster Toolkit > Guides

Was this helpful?

## Cluster Toolkit

Send feedback

Cluster Toolkit, formerly known as *Cloud HPC Toolkit*, is open-source software offered by Google Cloud which simplifies the process for you to deploy high performance computing (HPC), artificial intelligence (AI), and machine learning (ML) workloads on Google Cloud. It is designed to be highly customizable and extensible, and intends to address the deployment needs of a broad range of use cases.

Benefits

## Cluster Toolkit HTCondor Tutorial

Cluster Toolkit is an open-source software offered by Google Cloud which makes it easy for customers to deploy HPC environments on Google Cloud.

This tutorial will walk you through deploying a simple HTCondor pool on Google Cloud using the Cluster Toolkit.

### Select a Project

Select a project in which to deploy an HPC cluster on Google.

Once you have selected a project, click START.

### Add Credits to the Project

Talk with your tutorial leader to see if Google Cloud credits are available.



# Batteries included\*

GoogleCloudPlatform / cluster-toolkit · 296

HTCondor tutorial: stalled launch of htcondor-pool-ap-mig #1722

Answered by jd-foster

I hope this is the last question (and really appreciate your help so far).

The error is below, where it appears that htcondor-pool-ap-mig is expected to be available but isn't: when I inspect the Instance Groups console I can see it listed but the status is listed as Up/Down, with the hover tip reading "Instance group is resizing: 0 instances, scaling to 1" but the process has completely stalled (for 2.5 hours and counting). However, htcondor-pool-ap-mig is successfully created and running.

Are there any tips for diagnosing what has gone wrong and how to remedy this? Thanks again.

```
...  
module(htcondor,_access,_exec,_module,htcondor_ap_google_comPUTE_region_instance_group_manager,mig): Creation complete after 1s  
module(htcondor,_access,_exec,_module,htcondor_ap_google_comPUTE_region_instance_group_manager,mig): Still creating... ([@s elapsed)  
...  
module(htcondor,_access,_exec,_module,htcondor_ap_google_comPUTE_region_instance_group,mig): Creation complete after 2m6s [id:2023-08-29T01:39:53Z]  
module(htcondor,_access,_exec,_module,htcondor_ap_google_comPUTE_region_instance_group,mig): Reading...  
module(htcondor,_access,_exec,_module,htcondor_ap_google_comPUTE_region_instance_group,mig): Read complete after 1s [id:projects/<PROJECT_ID>/regions/us-central1/instances/htcondor-pool-ap-mig]  
Error: Resource postcondition failed
```

GoogleCloudPlatform / cluster-toolkit · 296

Fix htcondor config

Stable IP for HTCondor MIGs

Configure Renovate

Name conflict in HTCondor config object name

Release 1.42.0

Fix SSH issues with integration tests

update HTCondor modules to be compatible with TPG v6

Make CloudSQL secret replication configurable

Update internal usage of toolkit modules on develop to v1.36.0

Add allow automatic updates flag

GoogleCloudPlatform / cluster-toolkit · 296

HTCondor tutorial: Creating image error & Artifact Registry? #1716

Answered by tpdownes

Thanks for your previous help.

It appears that in the HTCondor tutorial build script, from here:

```
building image using packer module at htcondor-pool/packer/custom-image
```

It's trying to create an image in the Container Registry (gcr)?

The issue is this specific error:

```
==> htcondor-pool.googlecompute.toolkit_image: Startup script, if any, has finished running.  
==> htcondor-pool.googlecompute.toolkit_image: Deleting instance...  
htcondor-pool.googlecompute.toolkit_image: Instance has been deleted!  
==> htcondor-pool.googlecompute.toolkit_image: Creating image...  
==> htcondor-pool.googlecompute.toolkit_image: Error waiting for image: googleapis: Error 412: Location asia violates constraints/gcp/resourceLocations property)
```

(My organisation policy is to restrict projects to resources in the Australian region, so this image creation fails by the constraints/gcp/resourceLocations property.)

Is there perhaps a way to use the Artifacts Registry to circumvent this issue, or some other means to get past this?

Thanks again for any help.

tpdownes on Aug 25, 2023



# Potential applications

- Programmatically running different variations on the same problem
- Sub-problem solving on medium-sized instances
- Testing solvers
  - benchmarking updates and features
  - exploring problem-specific settings and customisations
    - linear algebra libraries
    - new features in dependencies



# Final thoughts



*Condor calls for aid!*

# Final thoughts

- I believe this is an accessible route for general users to access additional computing resources.
  - This is particularly something that could assist general JuMP modellers.
- I hope to have interested you (enough) to try this, and please reach out to chat.

Thank you.

@jd-foster



# CSIRO

