An ATLAS Researchers' Experience with HTCondor on STBC

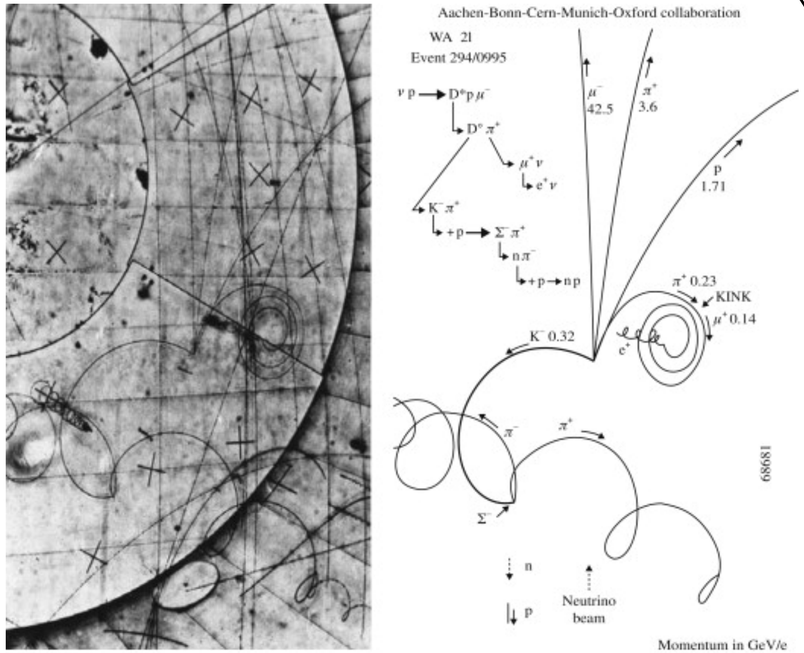
Nikhef, HTCondor EU Meeting, 24 September 2024

Zef Wolffs

Nikhef

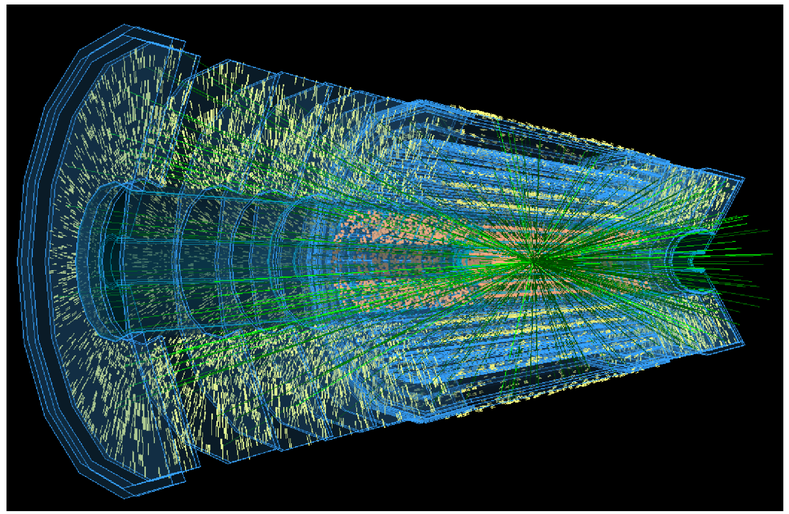
Background - Computational challenges for the LHC

Big European bubble chamber at CERN



Identifying events by meticulously following visible tracks

1970s



every 25ns

2020s

The explosion of data exerts pressure on the whole physics analysis infrastructure

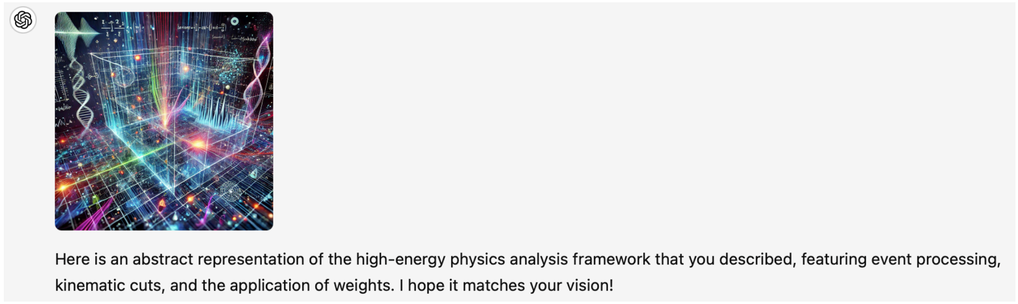
Cluster systems are a foundational building block to nearly all of the computing that we do

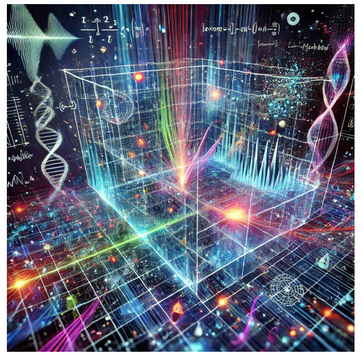
Often taken for granted

Nikhef

Background - What do I use HTCondor for?

- Analysing ATLAS' data in order to study the properties of the Higgs
  - 20 Tb of event by event processing: kinematic cuts, application of weights…

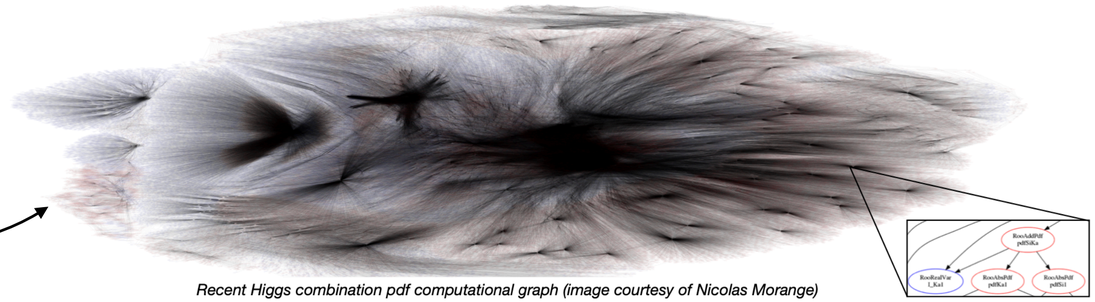




???

- Fitting models to data (maximum likelihood estimation), many times for many different hypotheses!

Computational graph of one of these models

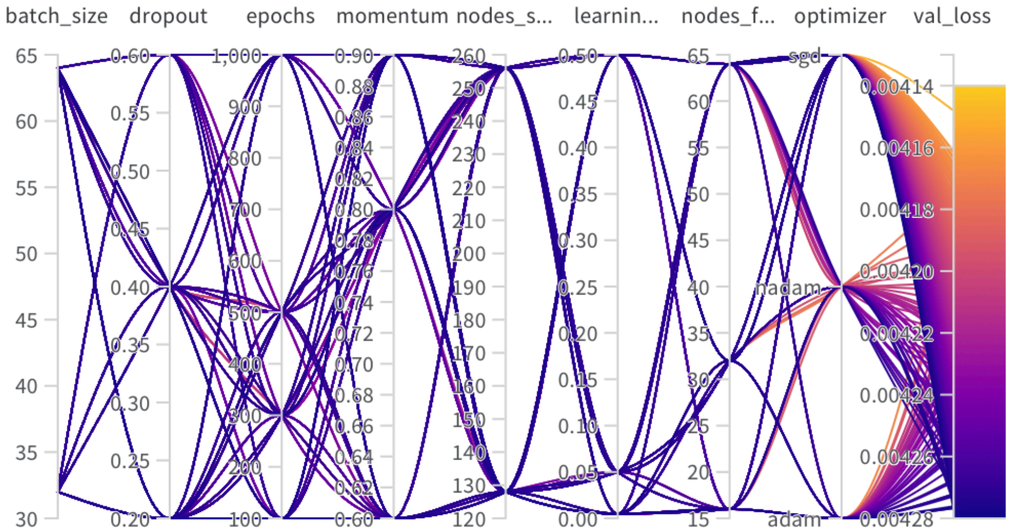


Recent Higgs combination pdf computational graph (image courtesy of Nicolas Morange)

Nikhef

Background - What do I use HTCondor for?

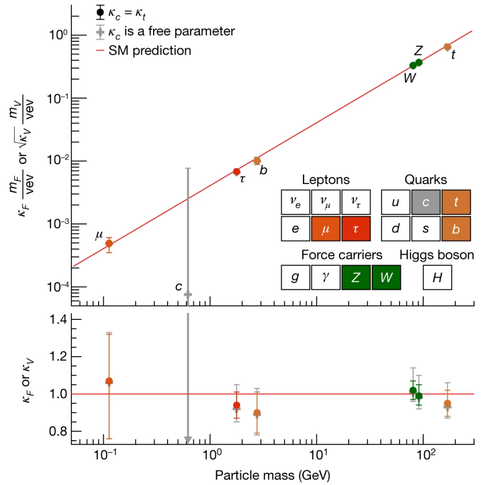
- Training and hyperparameter-optimizing machine learning models
  - Hyperparameter-optimization is the task of finding those **non-trainable** model parameters that result in an optimal model
  - A black box problem, often a random grid search is the best solution, where each iteration requires a full training -> high computational load



Background - What do I use HTCondor for?

- All these things are computationally intensive, but all are conveniently embarassingly parallelisable -> ideal for cluster systems!
- Most importantly they lead to nice (and important) results!!

Testing the standard model at the Highest energies possible, mapping out in detail a particle discovered only just 10 years ago



ATLAS finds evidence of off-shell Higgs boson production and measures Higgs boson's total width

15 November 2022 I By ATLAS Collaboration

ATLAS measures Higgs boson mass with unprecedented precision

21 July 2023 I By ATLAS Collaboration

ATLAS dives deeper into di-Higgs

By combining multiple Higgs boson pair studies, physicists are closer to finding out how the particle interacts with itself, providing clues to the stability of the Universe

18 JUNE, 2024 | By ATLAS collaboration

Nikhef

HTCondor User Experience

Background

- Have worked with some cluster systems
  - **PBS at Imperial**
  - (Old version of) **PBS/Torque at Nikhef**
  - **HTCondor at CERN**
  - **SLURM at SURF** (Dutch national supercomputer)
  - **HTCondor at Nikhef**

- Main conclusion: All are quite similar from a user perspective
  - This is good! Often code needs to be ported from one to the other
  - HTCondor feels the most different -> requires extra file with job options
    - Slightly less user friendly for beginners, better once used to it

Nikhef

Improvements with respect to previous PBS system

- Great feature: `+SingularityImage`
  - Using local centos7 (deprecated) in analysis workflow

- Great feature: “held jobs”, option to see “hold reason”

- Good documentation: https://htcondor.readthedocs.io
  - Much more functionality than I (or most of my colleagues) use

- Good python API

- Very convenient to have the same cluster submission system as at CERN, much of the work already done in analysis codes

Some point(s) of criticism

- Learning curve tends to be a bit steeper than for other systems
  - Having an extra file just for job options is not very intuitive at first
  - A very simple demo seems to not be easy to find

Nikhef

Helper Tools

Convenience tool: “condorsub”

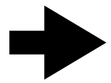
First stage of development workflow is often local prototyping ($N = 1$) on interactive node

- From this to full submitted workload should not cause much overhead
- Ideally do not want to write job configuration file

Script that allows user to run any command from the terminal as a job without the need for any extra code

Example:

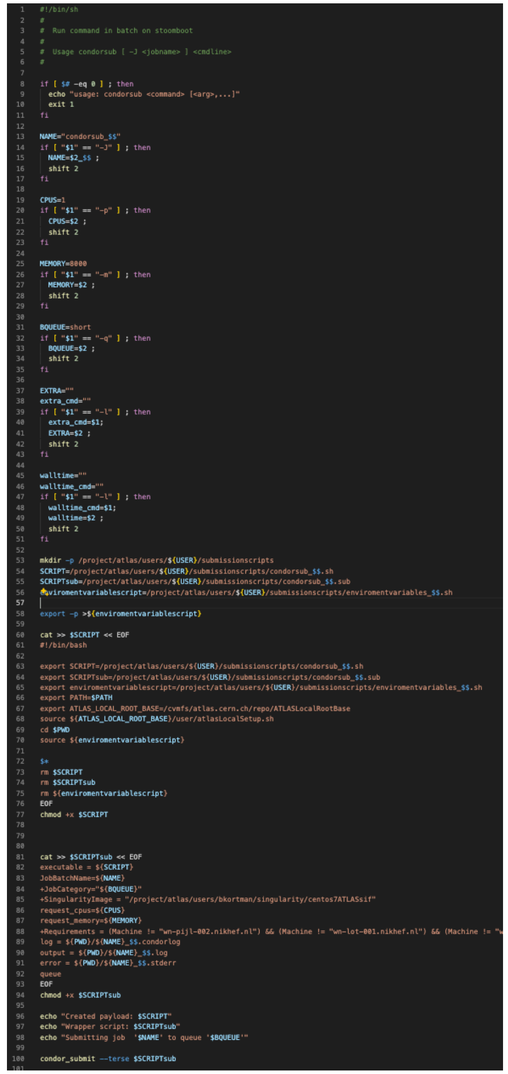
```
[zwolffs@stbc-i1 HWWOffshel-EFT]$ condorsub -q short python -c 'print("Hi")'
Created payload: /project/atlas/users/zwolffs/submissionscripts/condorsub_2065545.sh
Wrapper script: /project/atlas/users/zwolffs/submissionscripts/condorsub_2065545.sub
Submitting job  'condorsub_2065545' to queue 'short'
569491.0 - 569491.0
```



```
condorsub_2065545_2065545.condorlog
condorsub_2065545_2065545.log
condorsub_2065545_2065545.stderr
```


Nikhef

Convenience tool: “condorsub”



```
#!/bin/sh
#
#  Run command in batch on stoomboot
#
#  Usage condorsub [ -J <jobname> ] <cmdline>
#

if [ $# -eq 0 ] ; then
  echo "usage: condorsub <command> [<arg>,...]"
  exit 1
fi

NAME="condorsub_$$"
if [ "$1" == "-J" ] ; then
  NAME=$2_$$ ;
  shift 2
fi

CPUS=1
if [ "$1" == "-p" ] ; then
  CPUS=$2 ;
  shift 2
fi

MEMORY=8000
if [ "$1" == "-m" ] ; then
  MEMORY=$2 ;
  shift 2
fi

BQUEUE=short
if [ "$1" == "-q" ] ; then
  BQUEUE=$2 ;
  shift 2
fi

EXTRA=""
extra_cmd=""
if [ "$1" == "-l" ] ; then
  extra_cmd=$1;
  EXTRA=$2 ;
  shift 2
fi

walltime=""
walltime_cmd=""
if [ "$1" == "-l" ] ; then
  walltime_cmd=$1;
  walltime=$2 ;
  shift 2
fi

mkdir -p /project/atlas/users/${USER}/submissionscripts
SCRIPT=/project/atlas/users/${USER}/submissionscripts/condorsub_$$.sh
SCRIPTsub=/project/atlas/users/${USER}/submissionscripts/condorsub_$$.sub
enviromentvariablescript=/project/atlas/users/${USER}/submissionscripts/enviromentvariables_$$.sh

export -p >${enviromentvariablescript}

cat >> $SCRIPT << EOF
#!/bin/bash

export SCRIPT=/project/atlas/users/${USER}/submissionscripts/condorsub_$$.sh
export SCRIPTsub=/project/atlas/users/${USER}/submissionscripts/condorsub_$$.sub
export enviromentvariablescript=/project/atlas/users/${USER}/submissionscripts/enviromentvariables_$$.sh
export PATH=$PATH
export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh
cd $PWD
source ${enviromentvariablescript}

$*
rm $SCRIPT
rm $SCRIPTsub
rm ${enviromentvariablescript}
EOF
chmod +x $SCRIPT



cat >> $SCRIPTsub << EOF
executable = ${SCRIPT}
JobBatchName=${NAME}
+JobCategory="${BQUEUE}"
+SingularityImage = "/project/atlas/users/bkortman/singularity/centos7ATLASsif"
request_cpus=${CPUS}
request_memory=${MEMORY}
+Requirements = (Machine != "wn-pijl-002.nikhef.nl") && (Machine != "wn-lot-001.nikhef.nl") && (Machine != "
log = ${PWD}/${NAME}_$$.condorlog
output = ${PWD}/${NAME}_$$.log
error = ${PWD}/${NAME}_$$.stderr
queue
EOF
chmod +x $SCRIPTsub

echo "Created payload: $SCRIPT"
echo "Wrapper script: $SCRIPTsub"
echo "Submitting job  '$NAME' to queue '$BQUEUE'"

condor_submit --terse $SCRIPTsub
```

Supply command line arguments ‘`-p 16 -m 16000 …`’

Saving current environment’s variables, and names of scripts

Creating the executable payload, picking up environment’s variables,
Setting up some ATLAS specific stuff too that’s usually in ~/.bashrc

Wrapper script, specifying job configuration, setting supplied arguments

Run

Nikhef

Convenience tool: “condornode”

- Sometimes want to run interactively, even larger jobs not suitable for interactive nodes
- Use `condornode`, tiny wrapper around `condor_submit`

bin > condornode

```
1  condor_submit -interactive /user/zwolffs/bin/condor_interactive_medium.job
```

bin > condor_interactive_medium.job

```
1  executable=/user/zwolffs/bin/hi.sh
2  +JobCategory="medium"
3  +SingularityImage = "/project/atlas/users/bkortman/singularity/centos7ATLASsif"
4  request_cpus=4
5  request_memory = 16000
6  +Requirements = (Machine != "wn-pijl-002.nikhef.nl") && (Machine != "wn-lot-001.nikhef.nl")
7  queue
```

- Different options, for different interactive node requests

```
condor_interactive_long.job
condor_interactive_longbig.job
condor_interactive_longcpu.job
condor_interactive_longcpucpu.job
condor_interactive_medium.job
condor_interactive_mediumcpu.job
condor_interactive_short.job
condornode
condornodecpu
condornodelong
condornodelongbig
condornodelongcpu
condornodelongcpucpu
condornodeshort
```

- My most common workflow: `screen -> condornode` (to come back later)

Nikhef

Convenience tool: removing jobs by name

- Sometimes want to remove jobs by name, “`BATCH_NAME`” field with a wildcard, e.g. `condorsub*`

```
[zwolffs@stbc-i1 ~]$ condor_q


-- Schedd: taai-007.nikhef.nl : <145.107.7.246:9618?... @ 09/21/24 12:30:26
OWNER   BATCH_NAME          SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
zwolffs condorsub_2845009   9/20 23:24      _      1       _      1 569551.0
```

- Currently have ~50 line python script for this, could be convenient if it were easier

```python
#! /bin/python

import htcondor
import fnmatch
import argparse

def delete_jobs_by_name(wildcard_name):
    # Connect to the HTCondor schedd (scheduler)
    schedd = htcondor.Schedd()

    # Define a query to get all jobs
    query = schedd.query()

    if not query:
        print("No jobs found in the queue.")
        return

    # Filter jobs by matching the wildcard pattern in the job name
    matching_jobs = [job for job in query if fnmatch.fnmatch(job.get("JobBatchName", ""), wildcard_name)]

    if not matching_jobs:
        print(f"No jobs found matching the wildcard name: {wildcard_name}")
        return

    print(f"Found {len(matching_jobs)} job(s) matching the wildcard name: {wildcard_name}")

    # Create a list of job IDs to remove
    job_ids = [f'{job["ClusterId"]}.{job["ProcId"]}' for job in matching_jobs]

    try:
        # Remove the jobs
        schedd.act(htcondor.JobAction.Remove, job_ids)
        print(f"Successfully removed {len(job_ids)} job(s).")
    except Exception as e:
        print(f"An error occurred while removing jobs: {e}")

if __name__ == "__main__":
    # parse input argument
    parser = argparse.ArgumentParser(description="Remove jobs by name matching a wildcard substring.")
    parser.add_argument("wildcard_name", type=str, help="Wildcard substring to match job names.")
    args = parser.parse_args()

    wildcard_name = args.wildcard_name
    delete_jobs_by_name(wildcard_name)
```

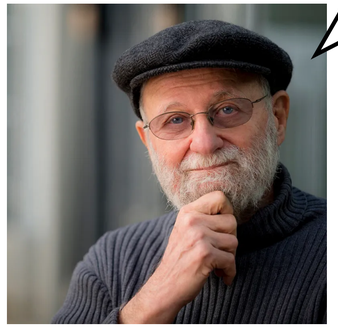
Nikhef

Conclusion

Conclusion

- Often we take the underlying computing systems that we do our physics on top of for granted
  - This actually means that they work serve our needs well

- I would say that, from the user side, HTCondor and the stoomboot cluster are often taken for granted
  - Transition to HTCondor at Nikhef was smooth
  - HTCondor is already proving to be much more easier to work with than previous (PBS/ torque) system



Prof. Donald Norman

Bill Gates