Ryan Simeon

- *I am a 3rd year graduate student at UW-Madison*
- *Mentors: Sridhara Dasu (UW-Madison), Nick Smith (FNAL)*

My TAC-HEP R&D project:

I am developing an Analysis Facility (AF) at the University of Wisconsin. The AF gives users a place to run analysis code that is user friendly, is pre-configured for columnar HEP analysis, and provides access to large-scale computing resources (in our case, the Wisconsin Tier-2). In particular, analysts should be able to go from testing on a local machine to running production-level code on thousands of cores without needing to know details about the computing infrastructure. Altogether, the goal is to let analysts not worry about computing, and instead focus on physics.

Accomplishments:

- Created the cowtools package, letting users connect to the Tier-2 in one line of code
- Created wisc-af-examples, a collection of examples and instructions to introduce new users to the Wisconsin Analysis Facility
- Tested Wisconsin's XCache system, finding a bottleneck that was severely impacting its performance
    - An XCache allows analysts to read frequently-accessed data much more quickly and reliably by caching it on-site
    - This work contributed to a talk at The 8th Asian Tier Center Forum

Next steps:

- Remove the XCache bottleneck and put it to production-level use
- Implement our systems in Kubernetes
- Grow our user base!
- Participate in benchmarking with other US Analysis Facilities
    - Ex: AGC

Tests in the Benchmark

The Test	Run the physics analysis above
What We Measure	• How long does it take to run the analysis? • How reliably can we read input files without failure? • What kinds of error messages do we get from reading files? • Where do our files come from?

Versions of the Test	• Where are the files?
	  • Read files remotely, from hosts around the world, OR
	  • Read files with Wisconsin Xcache
	• How many files to read?
	  • Read all files (1.5 TB in total), OR
	  • Read only first 5 files in each sample

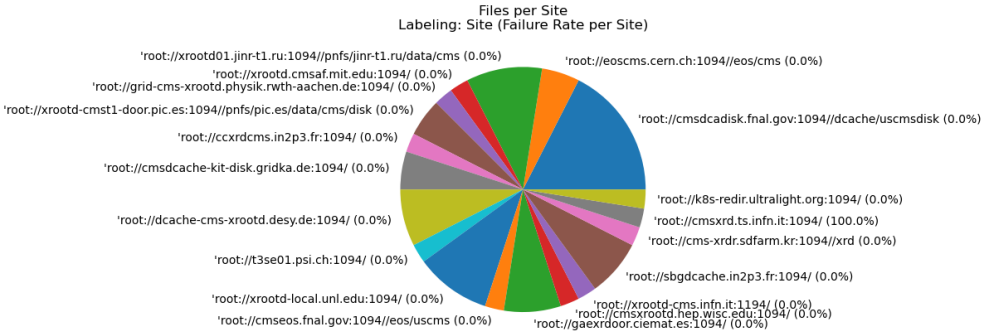
Configurations

- Running on up to 250 Condor cluster nodes
  - In practice, there are typically ~180 jobs concurrently running
  - The limit appears to be the taskgraph, ie: more nodes are available, the task just does not call for more than ~180 at a time
- Worker nodes run jobs in a custom Singularity container, based on central coffea v2024 image
- Clusters use nodes from Wisconsin Tier-2 and Center for High Throughput Computing (CHTC)

Remote Reading: Where are the Files From?

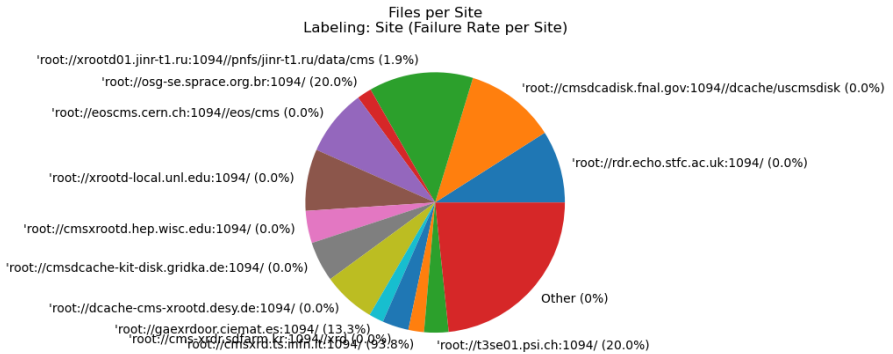
Max 5 Files per Sample

- 35 total files
- Pie wedges proportional to number of files from that site
- Number in parentheses is file read failure rate from that site



All Files (823 Files)

- 823 total files
- Pie wedges proportional to number of files from that site
- Number in parentheses is file read failure rate from that site
- Sites grouped in “other” host < 3% of total files and have zero failures



Analysis Facility Demo Images

```
os.environ["CONDOR_CONFIG"] = "/etc/condor/condor_config"

try:
    _x509_localpath = (
        [
            line
            for line in os.popen("voms-proxy-info").read().split("\n")
            if line.startswith("path")
        ][0]
        .split(":")[-1]
        .strip()
    )
except Exception as err:
    raise RuntimeError(
        "x509 proxy could not be parsed, try creating it with 'voms-proxy-init'"
    ) from err
_x509_path = f'/scratch/{os.environ["USER"]}/{_x509_localpath.split("/")[-1]}'
os.system(f"cp {_x509_localpath} {_x509_path}")
_x509_path = os.path.basename(_x509_localpath)

PWD = os.getcwd()
print(f"Transferring contents of {PWD} to workers")
INITIAL_DIR = f'/scratch/{os.environ["USER"]}'
cluster = HTCondorCluster(
    cores=1,
    memory="2 GB",
    disk="1 GB",
    death_timeout = '60',
    job_extra_directives={
        #"Requirements": 'Microarch >= "x86_64-v2"',
        "+JobFlavour": "tomorrow",
        "log": "dask_job_output.$(PROCESS).$(CLUSTER).log",
        "output": "dask_job_output.$(PROCESS).$(CLUSTER).out",
        "error": "dask_job_output.$(PROCESS).$(CLUSTER).err",
        "should_transfer_files": "yes",
        "when_to_transfer_output": "ON_EXIT_OR_EVICT",
        "container_image": "/cvmfs/unpacked.cern.ch/registry.hub.docker.com/coffeateam/coffea-dask-almalinux8:2024.11.0-py3.11",
        "InitialDir": INITIAL_DIR,
        "transfer_input_files": f'{_x509_path}'
    },
    job_script_prologue=[
        "export XRD_RUNFORKHANDLER=1",
        f"export X509_USER_PROXY={_x509_path}",
    ]
)
print('Condor logs, output files, error files in {}'.format(INITIAL_DIR))
cluster.adapt(minimum=1, maximum=MAX_WORKERS)
```

With cowtools

```
client = cowtools.GetCondorClient(max_workers=MAX_WORKERS)
```

Running on the Wisconsin Tier-2 via the Analysis Facility