Karlsruhe Institute of Technology

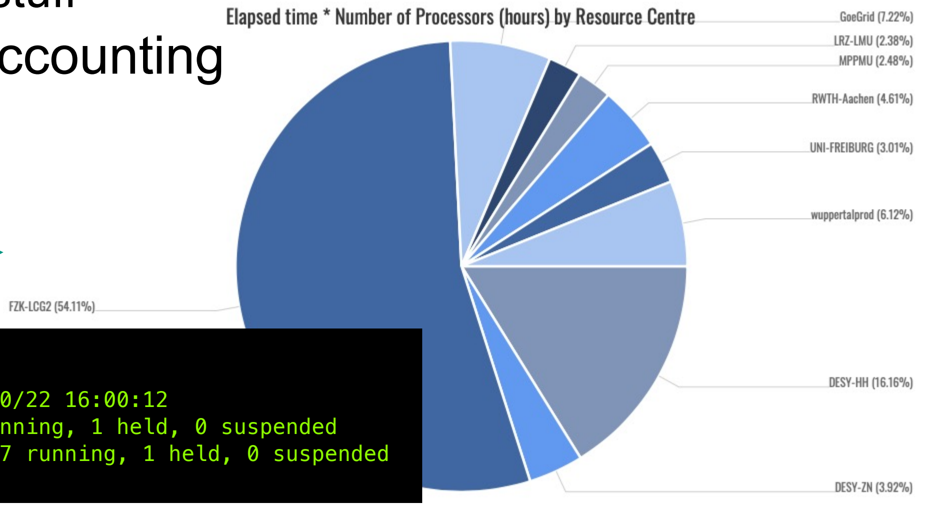
HTCondor-CE: APEL Accounting and HEPScore

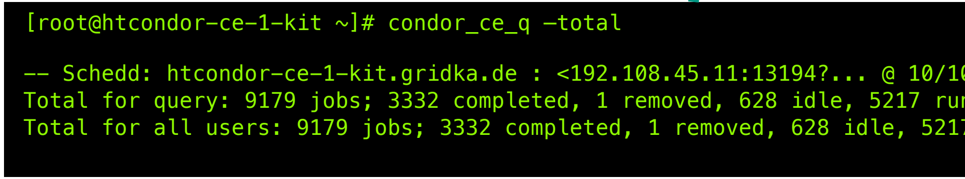
Max Fischer

(keeping HTCondor-CE APEL up to date and running, not officially affiliated with HTCondor-CE nor APEL)

What this talk is about

- What you have to do if you have to do APEL accounting
  - Read: Unless you are part of OSG (lucky you!)
- What you might want to do with APEL accounting
  - From starting simple to the complex stuff
- What you can learn about (APEL) accounting
  - In case you need/want your own
  - [Source code on GitHub]()



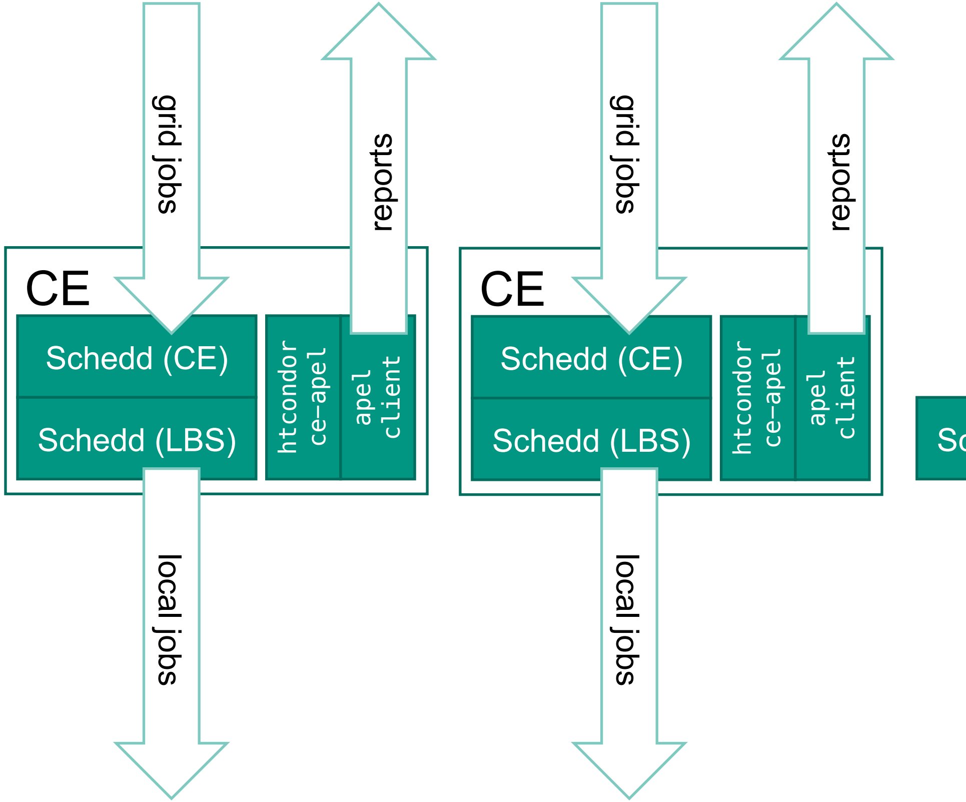


```
[root@htcondor-ce-1-kit ~]# condor_ce_q -total

-- Schedd: htcondor-ce-1-kit.gridka.de : <192.108.45.11:13194?... @ 10/10/22 16:00:12
Total for query: 9179 jobs; 3332 completed, 1 removed, 628 idle, 5217 running, 1 held, 0 suspended
Total for all users: 9179 jobs; 3332 completed, 1 removed, 628 idle, 5217 running, 1 held, 0 suspended
```

The Big Picture

- Pure HTCondor grid setup
  - HTC-CE accepts grid jobs
  - Local HTC manages resources
- Package: `htcondor-ce-apel`
  - Dumps and reads history ClassAds
  - Extracts generic summary per job
- Package: `apel-client`
  - Collects generic job summaries
  - Reports usage information



Getting Started [1/2]: Averaged Accounting

- Do: Install `htcondor-ce-apel` on each CE

```
yum install htcondor-ce-apel
```

  - This creates history dumps via `PER_JOB_HISTORY_DIR`
- Do: Enable `condor-ce-apel.timer` on each CE

```
systemctl enable condor-ce-apel.timer
```

  - This processes history dumps regularly via a script
  - This runs apel-client to upload reports by default
- Do: Configure apel client and parser

```
Let’s look at that on the next slide!
```

  - Parser needs to know what to parse
  - Client needs average cluster performance

Getting Started [2/2]: What APEL needs

- Parser needs per-job summaries
  - This is always the same!

```
vim /etc/apel/parser.cfg

[blah]
enabled = true
dir = /var/lib/condor-ce/apel/
filename_prefix = blah
subdirs = false


[batch]
enabled = true
reparse = false
type = HTCondor
parallel = false
dir = /var/lib/condor-ce/apel/
filename_prefix = batch
subdirs = false

…
```

- Client needs cluster information
  - This depends on your site!

```
vim /etc/apel/client.cfg

[spec_updater]
enabled = false
# Site as known in GOCDB
site_name = FZK-LCG2
# Average performance of the Cluster
# Format: <ce name>:9619/<ce name>,<spec type>,<spec value>
manual_spec1 = htcondor-ce-1-kit.gridka.de:9619/htcondor-ce-1-
kit.gridka.de-condor,HEPSPEC,13.45

…
```

  - Note: Must match data in GOCDB
    see Appendix

Interlude: Walltime, CPUTime and Performance

- Accounting goal: Report resources allocated and consumed by users
  - Allocated: RemoteWallClockTime x RequestCPU, RequestMemory, …
  - Consumed: RemoteUserCpu + RemoteSysCpu, RSS/PSS, …
  - Efficiently available via condor_history, PER_JOB_HISTORY_DIR, …

```
condor_history -af 'RemoteWallClockTime*RequestCPUs' 'RemoteUserCpu+RemoteSysCpu'
```

- Accounting challenge: Compare reports across machines/clusters
  - Approach: Weight simple metrics by performance of resources
  - APEL weights defined by performance benchmarks, currently HEPSPEC

```
v- Compute Entrypoint address -v                 benchmark specification -v
htcondor-ce-1-kit.gridka.de:9619/htcondor-ce-1-kit.gridka.de-condor,HEPSPEC,13.45
                                 ^-- Schedd/LRMS address --^   benchmark value -^
```

Advanced Steps: Per-Machine Performance

- Define performance per startd
  - `htcondor-ce-apel` adds CPU scale factor to each job summary
  - Useful for heterogeneous clusters
- Improved in HTCondor-CE v5.1.6
  - Optional: Start with a simple setup, add scaling/specs as needed
  - Absolute Specs: scaling computed automatically w.r.t. cluster average
  - Multiple Specs: future proof support for several performance metrics

```
# In the StartD condor config:
# define a performance factor compared to the average
ApelScaling = 1.414
STARTD_ATTRS = $(STARTD_ATTRS) ApelScaling
```

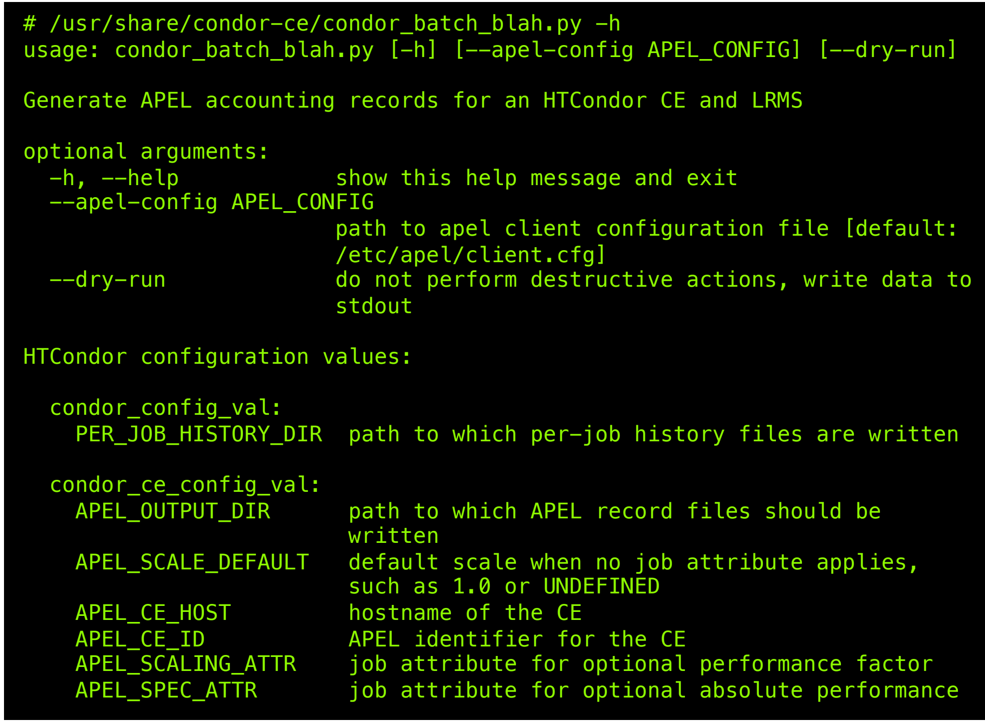
```
# In the StartD condor config:
# just do nothing and the cluster average is used
```

```
# In the StartD condor config:
# define absolute performance for specific benchmarks
ApelSpecs = [HEPSPEC=14.37; SI2K=2793]
STARTD_ATTRS = $(STARTD_ATTRS) ApelSpecs
```

New style ClassAd
(think JSON)

Knobs! Knobs! Knobs?

- Uses HTCondor(CE) config
  - Defaults should “just work”
- CE: APEL_SEND_RECORDS
  - Set to `False` if you have a central APEL database+client+sender
- CE: APEL_SCALE_DEFAULT
  - Set to UNDEFINED to force per-machine scaling/specs
- SYSTEM_JOB_MACHINE_ATTRS +CE:APEL_SPEC_ATTR+CE:APEL_SCALING_ATTR
  - Set the StartD attribute names storing scaling/specs



```
# /usr/share/condor-ce/condor_batch_blah.py -h
usage: condor_batch_blah.py [-h] [--apel-config APEL_CONFIG] [--dry-run]

Generate APEL accounting records for an HTCondor CE and LRMS

optional arguments:
  -h, --help            show this help message and exit
  --apel-config APEL_CONFIG
                        path to apel client configuration file [default:
                        /etc/apel/client.cfg]
  --dry-run             do not perform destructive actions, write data to
                        stdout

HTCondor configuration values:

  condor_config_val:
    PER_JOB_HISTORY_DIR  path to which per-job history files are written

  condor_ce_config_val:
    APEL_OUTPUT_DIR      path to which APEL record files should be
                         written
    APEL_SCALE_DEFAULT   default scale when no job attribute applies,
                         such as 1.0 or UNDEFINED
    APEL_CE_HOST         hostname of the CE
    APEL_CE_ID           APEL identifier for the CE
    APEL_SCALING_ATTR    job attribute for optional performance factor
    APEL_SPEC_ATTR       job attribute for optional absolute performance
```

Looking back, looking further…

- HTCondor-CE with APEL
  - Seamlessly adds APEL accounting to standard HTCondor-CE setup
  - Powered by community contribs
- Further improved in v5.1.6
  - Gradual per-machine specs
  - Absolute per-machine specs for automatically computed scaling
  - Multiple specs per machine

- Upcoming HEPSCORE
  - …with multiple scores in addition to HEPSPEC
  - Feel free to experiment adding HEPSCORE into ApelSpecs
  - ApelSpecs format stays the same

Want to try HTC-CE with APEL?
Cheatsheet and Examples
attached to this talk