

# Overview of all HTCondor submit commands



*Le Tour  
de Submit*



**PATH** PARTNERSHIP to ADVANCE  
THROUGHPUT  
COMPUTING

# de Le Tour Submit



# Overview

- › A (nearly) comprehensive tour of the HTCondor submit file commands and macros.

Legend:

**Green** - submit file command keywords and macros

**fixed pitch** - submit file examples

`fixed pitch` - console command fragments

# Qualification



# What are we talking about?

```
# This is comment in our submit file
```

```
U  
E  
#  
E  
E
```

```
Whole 30 minute talk  
"Submitting multiple  
https://www.youtube.com/watch
```

"Name = Value"  
We call "commands"  
subject of this talk

```
My.Test_tube_number = $(tube_number)  
Queue tube_number, cell from in.txt
```

# Required commands

- ❖ One (or two) required commands
  1. **Universe** (if not Vanilla)
    - Grid, VM, Parallel, Java, Local, Vanilla
    - Docker and Container are Vanilla+
  2. **Executable** or **docker\_image** or **container\_image**
    - Can use both Executable and one image
  3. **Arguments** or **Args**
    - Not very High Throughput without arguments...

# Resources needed

- ❖ `request_cpus`, `request_memory`, `request_disk`
  - How many cpus, memory, disk needed to run your job
  - Note "G", "M", "K" suffix for units
- ❖ `request_gpus`
  - How many GPUs needed
- ❖ `require_gpus`
  - What sort of GPUs needed

```
require_gpus = Capability >= 7.5 && GlobalMemoryMB > 4000
```
- ❖ `request_*`
  - Other resource type names defined by EP admin



# Arguments and Environment

- ❖ **Args / Arguments / Arguments2**
  - Use Arguments2 or "" around value for args with spaces
- ❖ **Env / Environment / Environment2**
  - Use Environment2 or "" around value for env with spaces or  
**Env = |key=value|key=value**
- ❖ **getenv = <pattern1> <pattern2>**
  - keys that match a pattern get a value from the submit process
  - Please don't use getenv = true

# Minimal submit file

```
executable = recalculate
arguments = run_a_bit 107 47

Request_cpus = 1
Request_memory = 1G
Request_Disk = 10G

queue
```

# Grands Départs



# Grands Départs



# Minimal submit file

```
executable = recalculate
```

```
arguments = 1 1 1
```

These are all

```
My_submit_var = test_tube7
```

```
Request_cpus = 1
```

```
Request_memory = 1G
```

Unless they are not...

```
Request_Disk = 10G
```

```
Environment = Var=$(My_submit_var)
```

```
queue
```

# Submit variables / macros

- ❖  $\$(variable:default)$  is text replacement at submit time
  - Built-in variables
    - **Row**, **Item**, **Step** - use with complex Queue statement
    - **ClusterId** or **Cluster** - unique id per submit
    - **Procid** or **Process** - unique id for each job in submit
    - **JobId** (new!) is  $\$(ClusterId).\$(Procid)$
    - **SUBMIT\_TIME**, **Day**, **Month**, **Year**
    - **SUBMIT\_FILE** - use with  $\$BASENAME()$  or  $\$F()$
    - **IsLinux**, **IsWindows**, **Arch**, **Opsys**, **OpSysAndVer**
    - **Node** (parallel universe only)

# Execution point attributes

- ❖ `$$(attr:default)` or `$$([<expr>])` is text replacement just before execution
    - Use any slot attribute, like
      - `$(Arch)`
      - `$(AssignedGpus)`
      - `$(CondorScratchDir)`
        - Expands in `Environment` to execution directory
- ```
transfer_input_files = stuff/  
Environment = STUFF_DIR=$$(CondorScratchDir)/stuff
```

Useful for admin – defined EP Attrs with STARTD\_ATTR

# Log of job progress

## ❖ Log

- Log of job state changes a.k.a. job event log
- Share a log between jobs for use with `condor_watch_q`

## ❖ `uexec_attrs`

- Additional slot attributes to print in the execute event

## ❖ `job_machine_attrs` - copy slot attrs into the job ad

## ❖ `job_machine_attrs_history_length` - keep previous attrs

## ❖ `job_ad_information_attrs` - write an attrs event into Log



# Containerization

- ❖ **Universe = Docker or Container**
  - Actual universe is **Vanilla** with container runtime matching
- ❖ **Container\_Image** - implies Container universe
  - `docker://` or `oras://` sets `WantDockerImage`
  - `*.sif` sets `WantSIF`
  - `*/` sets `WantSandboxImage`
- ❖ **Docker\_Image** - implies Docker universe
  - sets `WantDocker` - accept no substitutes

# Container options

- ❖ `container_target_dir`
  - Working directory of the job inside the container
- ❖ `transfer_container = true | false`
- ❖ `docker_pull_policy = always`
  - Never use a cached docker image, always pull
- ❖ `container_service_names`
  - Request docker port forwarding
- ❖ `docker_network_type = host | none | <custom>`

# The Transfer



# Inputs

- ❖ `transfer_input_files`
  - Files and/or directories with or without keeping the directory
- ❖ `should_transfer_files` = YES | NO | IF\_NEEDED
- ❖ `transfer_executable` = True | False

# "Std" Outputs and/or logging

- ❖ Stdout or Output
  - write stdout of job to a file
- ❖ Stderr or Error
  - write stderr of job to a file
  - merged stdout/stderr if same filename as Stdout
- ❖ transfer\_output or stream\_output
  - Refers to Stdout or Output only
- ❖ transfer\_error or stream\_error
  - Refers to Stderr or Error only

# Output Files

## ❖ `transfer_output_files`

- List files and directories to transfer with or without preserving the directory name. Default is to transfer all changed files

## ❖ `when_to_transfer_output`

- `ON_SUCCESS`, `ON_EXIT`, `ON_EXIT_OR_EVICT`, `ALWAYS`, `NONE`

## ❖ `output_destination`

- Send output files to the given directory, or URL/plugin

## ❖ `transfer_output_remaps`, `preserve_relative_paths`

- Rename files and/or change destination during transfer

# File transfer plugins

- ❖ Use a URL prefix for any input or output transfer
    - one of: file,ftp,https,osdf,davs,box,gdrive,http,data,dav,s3,gs
    - EP admin can extend the list of prefix
  - ❖ **transfer\_plugins** = <list-of-transfer-plugins>
    - Transfer a transfer plugin, then use it as a transfer plugin
    - Runs in the job context before the job
- ```
transfer_plugins = unzip=myunzipper.sh  
transfer_input_files = foo.zip, unzip://foo.zip
```

# Alpe d'Huez





# Verifying job correctness

- ❖ `allowed_job_duration` - hold if over cumulative time
- ❖ `allowed_execute_duration` - hold if current run over time
- ❖ `manifest` - save environment and a list of file checksums
- ❖ `manifest_dir` - where to put the manifest files
- ❖ `max_transfer_input_mb` - don't start if input is large
- ❖ `max_transfer_output_mb` - don't transfer if output is large
- ❖ `periodic_hold`, `periodic_release`, `periodic_remove`

# Job retries

- ❖ **success\_exit\_code**
  - Retry until this exit code, required if success is not 0
- ❖ **max\_retries**
  - Rerun job until success exit code or max retries
- ❖ **retry\_until** - futility exit code or success expression
- ❖ **on\_exit\_remove** - completion expression success or fail
- ❖ **on\_exit\_hold** - job recoverable failure expression

# Exit and Restart Checkpointing

- ❖ `checkpoint_exit_code = <exit-code>`
  - Checkpoint when job exits with this code
- ❖ `transfer_checkpoint_files`
  - Override `transfer_output_files` for checkpoints
- ❖ `erase_output_and_error_on_restart = false`
  - Start each execution with fresh stdout and stderr files

# Who am I ?

- ❖ `run_as_owner` = true | false
  - run job on EP as submitting user
- ❖ `load_profile` = true
  - on Windows, load a Registry before starting the job
- ❖ `accounting_group`, `accounting_group_user`, `nice_user`
  - set (or influence) usage accounting for the job
- ❖ `use_oauth_services`, `use_scitokens`, `x509userproxy`
  - send access tokens along with the job

# Tag your jobs

- ❖ **batch\_name** = <your own tagging schema>
  - condor\_q groups jobs by batch\_name
  - also `condor_submit -batch-name`
  - suggestion: **batch\_name** = `$BASENAME(SUBMIT_FILE)`
- ❖ **batch\_id** = <id>.<anything>
  - associate a job with another job for `condor_q <id>`
  - also `condor_submit -batch-id`
- ❖ **description** = <text>
  - `condor_q -nobatch` shows the <text>

# batch\_name and description example

*submit file fragment*

```
...  
batch_name = transcode  
Description = $Fn(song)  
Queue song from division/*.mp3
```

```
> condor_q  
-- Schedd: example.cs.wisc.edu  
OWNER  BATCH_NAME  SUBMITTED  DONE  RUN  IDLE  TOTAL  JOB_IDS  
johnkn transcode  7/6  15:04  _    8    4    16  20.0 ... 21.11  
  
> condor_q -nobatch  
-- Schedd: example.cs.wisc.edu  
ID      OWNER      SUBMITTED  RUN_TIME  ST  PRI  SIZE  CMD  
20.0    johnkn     7/6  15:04    0+00:00:13  R  0    0.0  (Wearing the Inside Out)  
20.1    johnkn     7/6  15:04    0+00:00:13  R  0    0.0  (Marooned)  
20.2    johnkn     7/6  15:04    0+00:00:12  R  0    0.0  (Coming Back To Life)  
20.3    johnkn     7/6  15:04    0+00:00:12  R  0    0.0  (Cluster One)  
20.4    johnkn     7/6  15:04    0+00:00:12  R  0    0.0  (Poles Apart)  
20.5    johnkn     7/6  15:04    0+00:00:12  R  0    0.0  (Lost for Words)  
20.6    johnkn     7/6  15:04    0+00:00:13  R  0    0.0  (A Great Day For Freedom)  
20.7    johnkn     7/6  15:04    0+00:00:12  R  0    0.0  (Keep Talking)  
...  
...
```

# Time Trial



# Cron (run job at a specific time)

- ❖ cron\_minute, cron\_hour, cron\_month, cron\_day\_of\_week, cron\_day\_of\_month
  - Job runs at specific day/time
- ❖ deferral\_time
  - time need to get ready to run job
- ❖ cron\_window
  - How far off of requested time is ok to run



# Materialize jobs in the AP

## ❖ `max_idle`

- Limit number of non-running materialized jobs

## ❖ `max_materialize`

- Limit total number of materialized jobs

## ❖ `condor_submit -factory`

- Vastly reduce submit time, and use AP configuration to limit number of materialized jobs

# Champs-Élysées



# Raw Job ClassAd attributes

- ❖ `My.<attr>` or `+<attr> = <value>`
  - ❖ Insert `<attr>=<value>` directly into the job ClassAd
  - ❖ Value must be a ClassAd expression, strings must be quoted!
  - ❖ Define your own attributes
  - ❖ Can override the value set by a submit command (caution!)
  - ❖ Reference using `$(My.<attr>)`
  - ❖ Use `$F(My.<attr>)` to remove quotes

# EXTENDED\_SUBMIT\_COMMANDS

- › AP defined submit commands for simple things, mix with JOB\_TRANSFORMS to do complex things

```
EXTENDED_SUBMIT_COMMANDS @=end
  WantGlidein = true
  LongJob = false
  RetryIfTransferFails = "string"
  ProjectName = "string"
  accounting_group_user = error
@end
```

## *submit file*

```
# use just like normal submit keywords, the value will be converted into the correct type of data
LongJob = true
RetryIfTransferFails = Syracuse
```

# EXTENDED\_SUBMIT\_HELPFILE

- › AP defined file or URL to inform the user

```
# return the contents of this file to the user
EXTENDED_SUBMIT_HELPFILE = $(LOCAL_DIR)/submit_help.txt
# or return the URL to the user
EXTENDED_SUBMIT_HELPFILE = http://example.com/submit_help
```

```
> condor_submit -capabilities
Schedd ap0.chtc.wisc.edu
Has Late Materialization enabled
Has Extended submit commands:
    accounting_group_user value is forbidden
    LongJob                value is Boolean true/false
    ProjectName            value is string
    RetryIfTransferFails  value is string
    WantFlocking           value is boolean true/false
    WantGlidein            value is boolean true/false
Has Extended help:
    http://example.com/submit_help
```

# SUBMIT\_TEMPLATE\_<name>

- › Submit language templates defined in config of submit

*config file*

```
SUBMIT_TEMPLATE_NAMES = $(SUBMIT_TEMPLATE_NAMES) TensorFlow
SUBMIT_TEMPLATE_TensorFlow @=end
  if ! $(1?)
    error : Template:TensorFlow requires at least 1 argument - TensorFlow(ver, target_dir)
  endif
  Universe = container
  container_image = TensorFlow$(1).sif
  container_target_dir = $(2:/workspace/dir)
@end
```

*submit file*

```
use Template : TensorFlow(95)
```



# Thank You!



**Follow us on Twitter!**  
<https://twitter.com/HTCondor>



**PATH** PARTNERSHIP to ADVANCE  
**THROUGHPUT**  
**COMPUTING**

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATH Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.