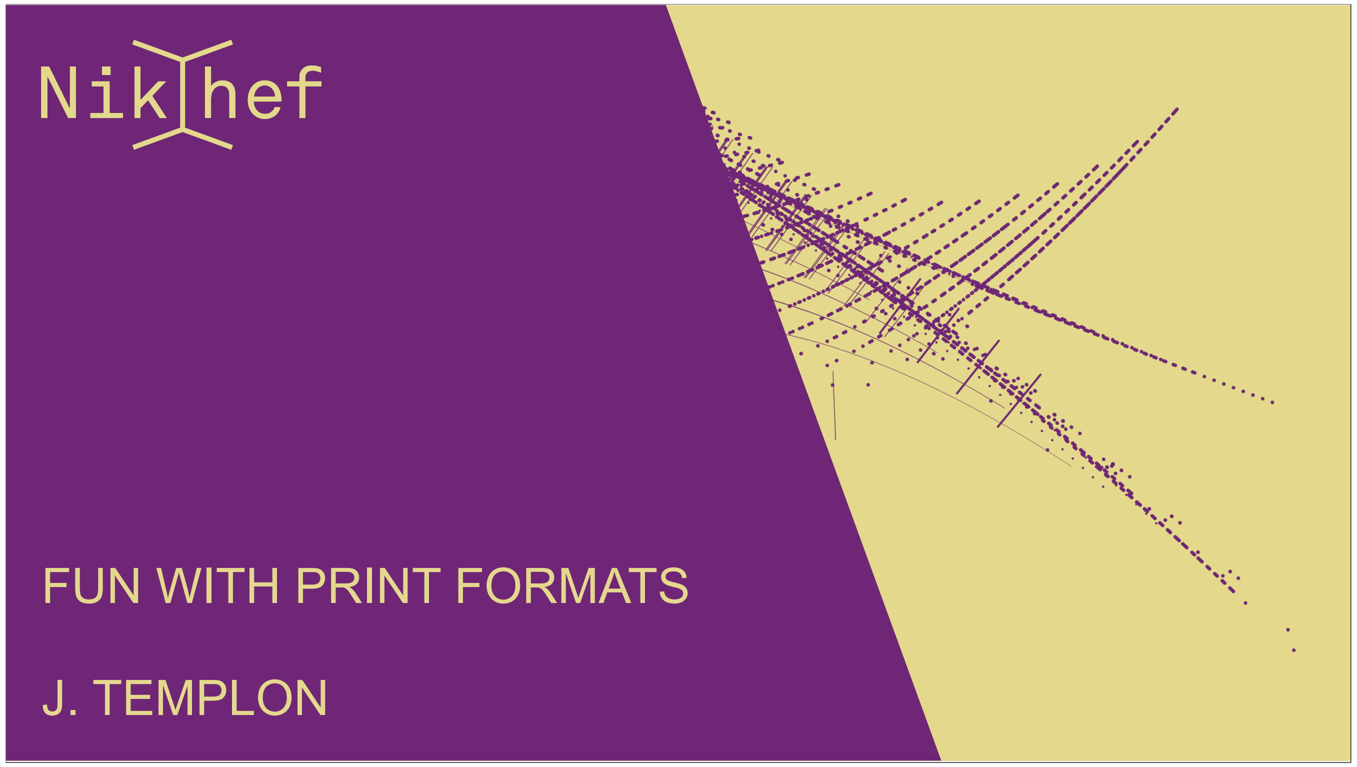


Nikhef

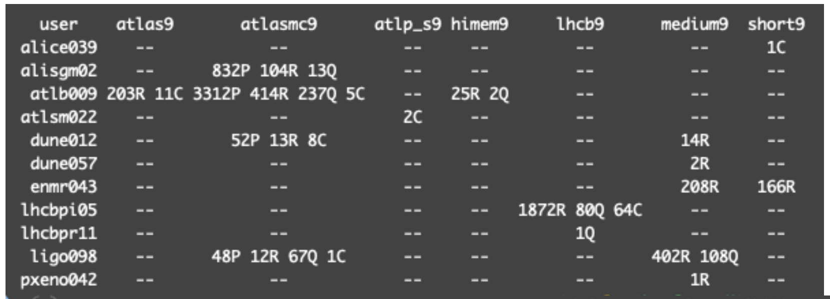
FUN WITH PRINT FORMATS

J. TEMPLON

Three goals :

- give a talk again
- Expose print formats, don’t hear much about them
- Learn about better condor-native ways to do what I want

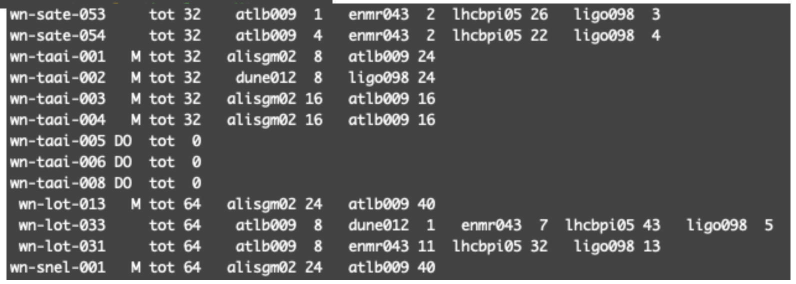
BACKGROUND : OPERATIONS TOOLBOX



```
  user    atlas9        atlasmc9        atlp_s9 himem9     lhcb9        medium9  short9
alice039    --             --             --     --         --            --       1C
alisgm02    --       832P 104R 13Q        --     --         --            --       --
 atlb009 203R 11C 3312P 414R 237Q 5C      --    25R 2Q      --            --       --
atlsm022    --             --             2C     --         --            --       --
 dune012    --         52P 13R 8C         --     --         --           14R       --
 dune057    --             --             --     --         --            2R       --
 enmr043    --             --             --     --         --           208R     166R
lhcbpi05    --             --             --     --    1872R 80Q 64C      --       --
lhcbpr11    --             --             --     --         1Q            --       --
 ligo098    --       48P 12R 67Q 1C       --     --         --         402R 108Q   --
pxeno042    --             --             --     --         --            1R       --
```

qsummnew

wnu



```
wn-sate-053     tot 32    atlb009  1   enmr043  2  lhcbpi05 26   ligo098  3
wn-sate-054     tot 32    atlb009  4   enmr043  2  lhcbpi05 22   ligo098  4
wn-taai-001   M tot 32   alisgm02  8   atlb009 24
wn-taai-002   M tot 32    dune012  8   ligo098 24
wn-taai-003   M tot 32   alisgm02 16   atlb009 16
wn-taai-004   M tot 32   alisgm02 16   atlb009 16
wn-taai-005 DO  tot  0
wn-taai-006 DO  tot  0
wn-taai-008 DO  tot  0
 wn-lot-013   M tot 64   alisgm02 24   atlb009 40
 wn-lot-033     tot 64    atlb009  8   dune012  1   enmr043  7  lhcbpi05 43   ligo098  5
 wn-lot-031     tot 64    atlb009  8   enmr043 11  lhcbpi05 32   ligo098 13
wn-snel-001   M tot 64   alisgm02 24   atlb009 40
```

Main two commands used on Torque cluster - who’s running what, and node status
Qsummnew is very rich - can see which users are running what classes of jobs,
How many queued, waiting, recently completed, what fraction is multicore
A large “C” count is a primary red flag for Torque - that’s usually that one user

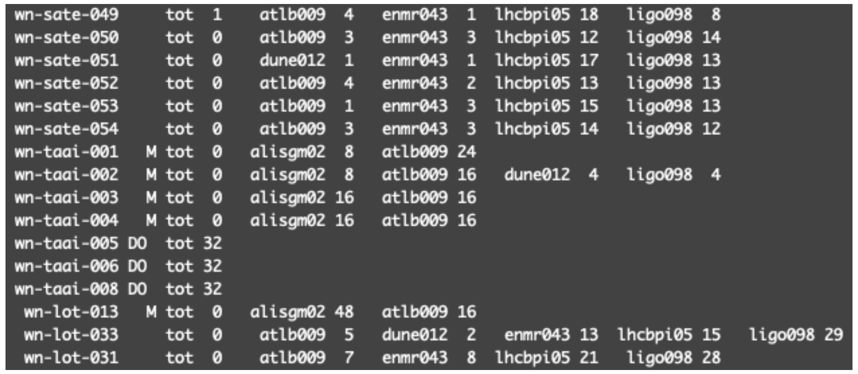
cqsummnew : a H flag for held, a U for unmaterialized, etc.

BACKGROUND : OPERATIONS TOOLBOX

```
wnu -f | grep -v "[OM]" | \
   awk '/wn-/ && $3>6 {print $1 ".farm.nikhef.nl"}' | \
   xargs mcnodeman -a
```

wnu -f

-f flag to wnu: 3rd column
ls # free slots

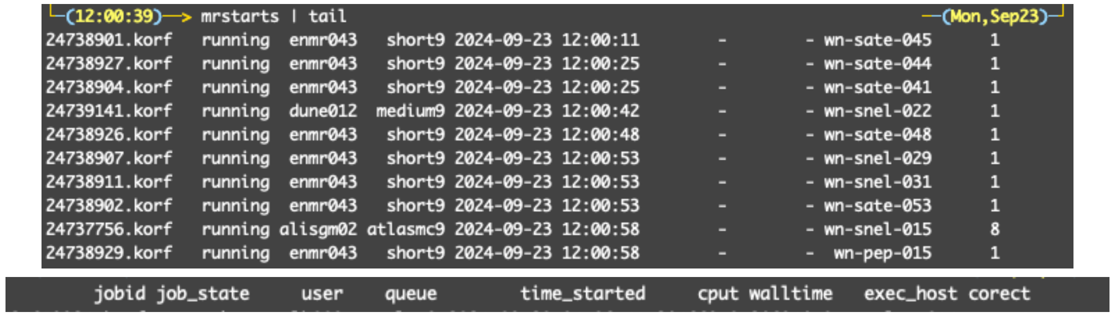


```
wn-sate-049     tot  1    atlb009  4   enmr043  1  lhcbpi05 18   ligo098  8
wn-sate-050     tot  0    atlb009  3   enmr043  3  lhcbpi05 12   ligo098 14
wn-sate-051     tot  0    dune012  1   enmr043  1  lhcbpi05 17   ligo098 13
wn-sate-052     tot  0    atlb009  4   enmr043  2  lhcbpi05 13   ligo098 13
wn-sate-053     tot  0    atlb009  1   enmr043  3  lhcbpi05 15   ligo098 13
wn-sate-054     tot  0    atlb009  3   enmr043  3  lhcbpi05 14   ligo098 12
wn-taai-001   M tot  0   alisgm02  8   atlb009 24
wn-taai-002   M tot  0   alisgm02  8   atlb009 16   dune012  4   ligo098  4
wn-taai-003   M tot  0   alisgm02 16   atlb009 16
wn-taai-004   M tot  0   alisgm02 16   atlb009 16
wn-taai-005 DO  tot 32
wn-taai-006 DO  tot 32
wn-taai-008 DO  tot 32
 wn-lot-013   M tot  0   alisgm02 48   atlb009 16
 wn-lot-033     tot  0    atlb009  5   dune012  2   enmr043 13  lhcbpi05 15   ligo098 29
 wn-lot-031     tot  0    atlb009  7   enmr043  8  lhcbpi05 21   ligo098 28
```

Command line tools - in unix philosophy, can pipe stuff together
I like the command line … in general not a big fan of dashboards.
There you have to know what you want in advance
Or else learn the tools equivalent of awk

BACKGROUND : OPERATIONS TOOLBOX

`mrstarts`



```
└─(12:00:39)──> mrstarts | tail                                                    ─(Mon,Sep23)─┘
24738901.korf   running  enmr043   short9 2024-09-23 12:00:11        -         - wn-sate-045      1
24738927.korf   running  enmr043   short9 2024-09-23 12:00:25        -         - wn-sate-044      1
24738904.korf   running  enmr043   short9 2024-09-23 12:00:25        -         - wn-sate-041      1
24739141.korf   running  dune012  medium9 2024-09-23 12:00:42        -         - wn-snel-022      1
24738926.korf   running  enmr043   short9 2024-09-23 12:00:48        -         - wn-sate-048      1
24738907.korf   running  enmr043   short9 2024-09-23 12:00:53        -         - wn-snel-029      1
24738911.korf   running  enmr043   short9 2024-09-23 12:00:53        -         - wn-snel-031      1
24738902.korf   running  enmr043   short9 2024-09-23 12:00:53        -         - wn-sate-053      1
24737756.korf   running alisgm02 atlasmc9 2024-09-23 12:00:58        -         - wn-snel-015      8
24738929.korf   running  enmr043   short9 2024-09-23 12:00:58        -         -  wn-pep-015      1
```

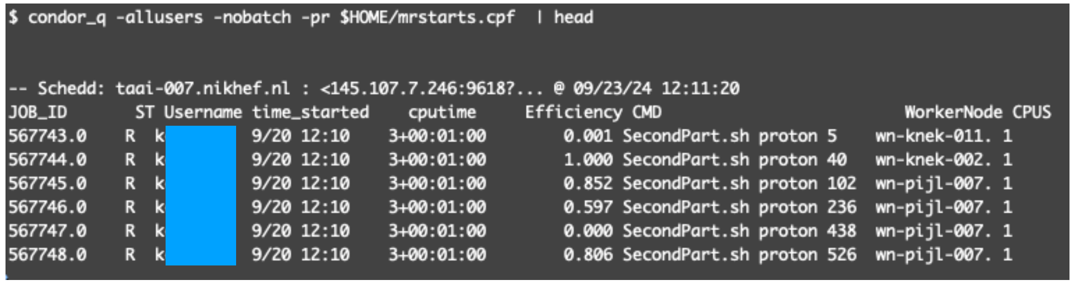
```
        jobid job_state      user     queue         time_started      cput walltime    exec_host corect
```

HTCondor gotcha - have to query both condor_q and condor_history
(Most recent start may have already finished!)

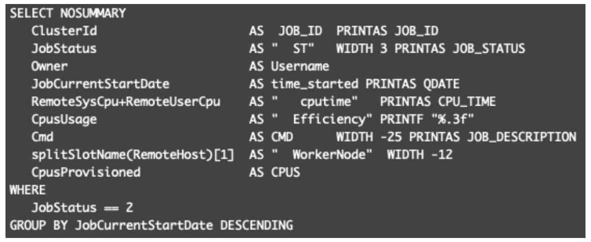
Note : yellow timestamp is time at which prompt appeared :)
Torque doesn’t like lots of starts and stops, one use case
Another is that user complains, no jobs starting - take a look, who is starting and how long ago was the last start?

CONDOR MRSTARTS USING PRINT FORMAT

cmrstarts

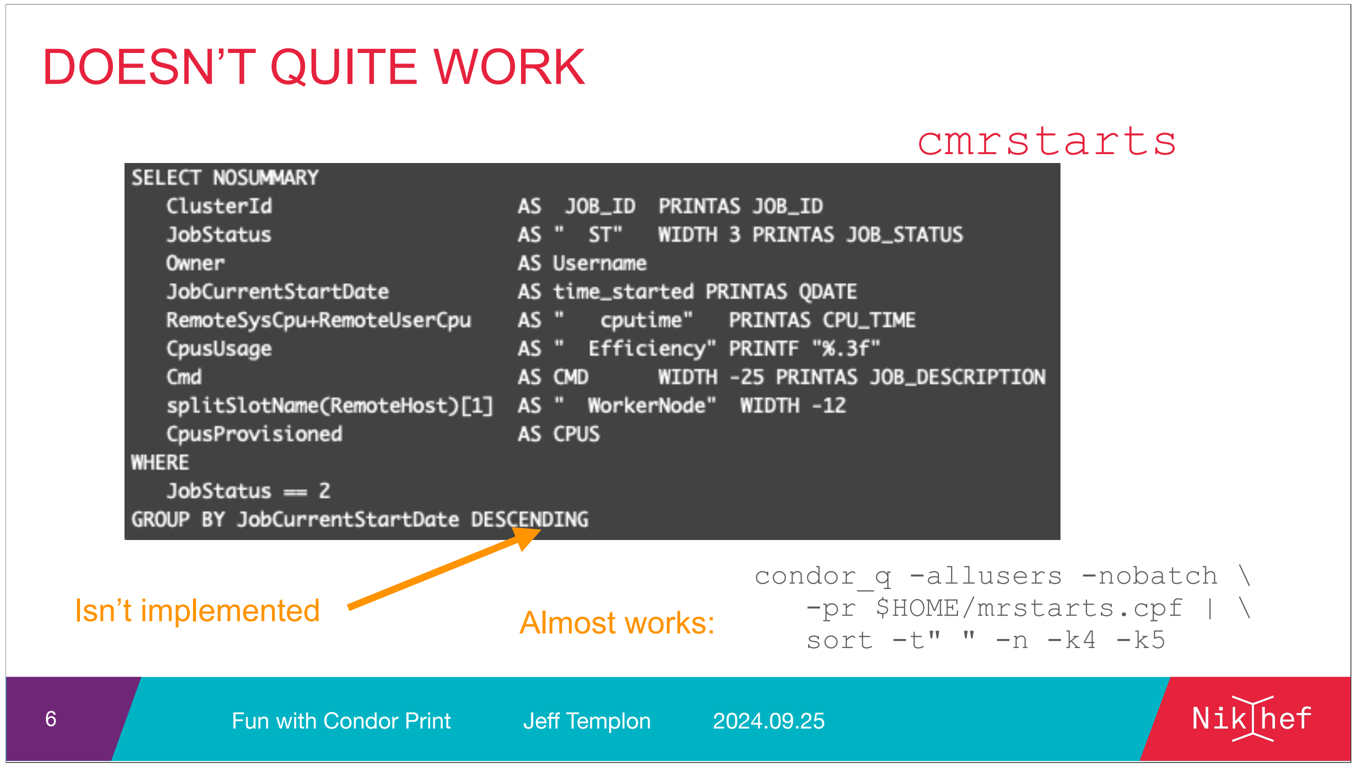


```
$ condor_q -allusers -nobatch -pr $HOME/mrstarts.cpf  | head


-- Schedd: taai-007.nikhef.nl : <145.107.7.246:9618?... @ 09/23/24 12:11:20
JOB_ID       ST Username time_started    cputime     Efficiency CMD                      WorkerNode CPUS
567743.0     R  k[redacted] 9/20 12:10    3+00:01:00       0.001 SecondPart.sh proton 5    wn-knek-011. 1
567744.0     R  k        9/20 12:10    3+00:01:00       1.000 SecondPart.sh proton 40   wn-knek-002. 1
567745.0     R  k        9/20 12:10    3+00:01:00       0.852 SecondPart.sh proton 102  wn-pijl-007. 1
567746.0     R  k        9/20 12:10    3+00:01:00       0.597 SecondPart.sh proton 236  wn-pijl-007. 1
567747.0     R  k        9/20 12:10    3+00:01:00       0.000 SecondPart.sh proton 438  wn-pijl-007. 1
567748.0     R  k        9/20 12:10    3+00:01:00       0.806 SecondPart.sh proton 526  wn-pijl-007. 1
```



```
SELECT NOSUMMARY
   ClusterId                       AS  JOB_ID  PRINTAS JOB_ID
   JobStatus                       AS "  ST"   WIDTH 3 PRINTAS JOB_STATUS
   Owner                           AS Username
   JobCurrentStartDate             AS time_started PRINTAS QDATE
   RemoteSysCpu+RemoteUserCpu      AS "   cputime"   PRINTAS CPU_TIME
   CpusUsage                       AS "  Efficiency" PRINTF "%.3f"
   Cmd                             AS CMD      WIDTH -25 PRINTAS JOB_DESCRIPTION
   splitSlotName(RemoteHost)[1]    AS "  WorkerNode"  WIDTH -12
   CpusProvisioned                 AS CPUS
WHERE
   JobStatus == 2
GROUP BY JobCurrentStartDate DESCENDING
```

Print format was a complete surprise, brilliant idea. Lots to explore there.



DOESN'T QUITE WORK

cmrstarts

```
SELECT NOSUMMARY
   ClusterId                     AS  JOB_ID  PRINTAS JOB_ID
   JobStatus                     AS "  ST"   WIDTH 3 PRINTAS JOB_STATUS
   Owner                         AS Username
   JobCurrentStartDate           AS time_started PRINTAS QDATE
   RemoteSysCpu+RemoteUserCpu    AS "   cputime"   PRINTAS CPU_TIME
   CpusUsage                     AS "  Efficiency" PRINTF "%.3f"
   Cmd                           AS CMD      WIDTH -25 PRINTAS JOB_DESCRIPTION
   splitSlotName(RemoteHost)[1]  AS "  WorkerNode"  WIDTH -12
   CpusProvisioned               AS CPUS
WHERE
   JobStatus == 2
GROUP BY JobCurrentStartDate DESCENDING
```

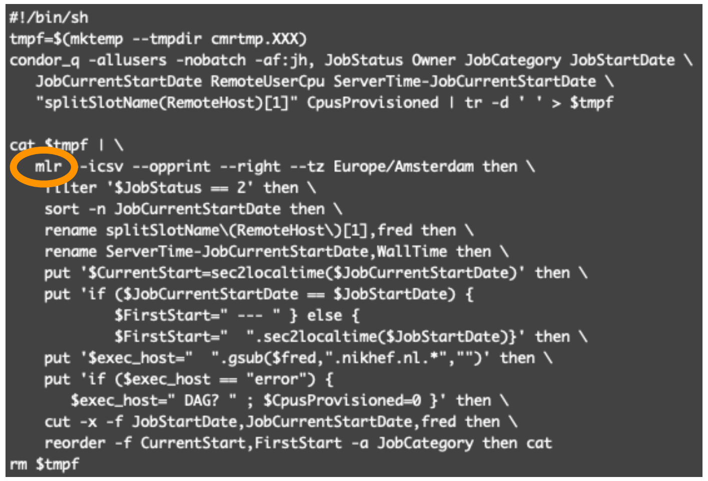
Isn't implemented

Almost works:

```
condor_q -allusers -nobatch \
    -pr $HOME/mrstarts.cpf | \
    sort -t" " -n -k4 -k5
```

Almost works except close to new years (1 < 12) and the header becomes a footer
Please fix the sort!

THIS DOES WORK



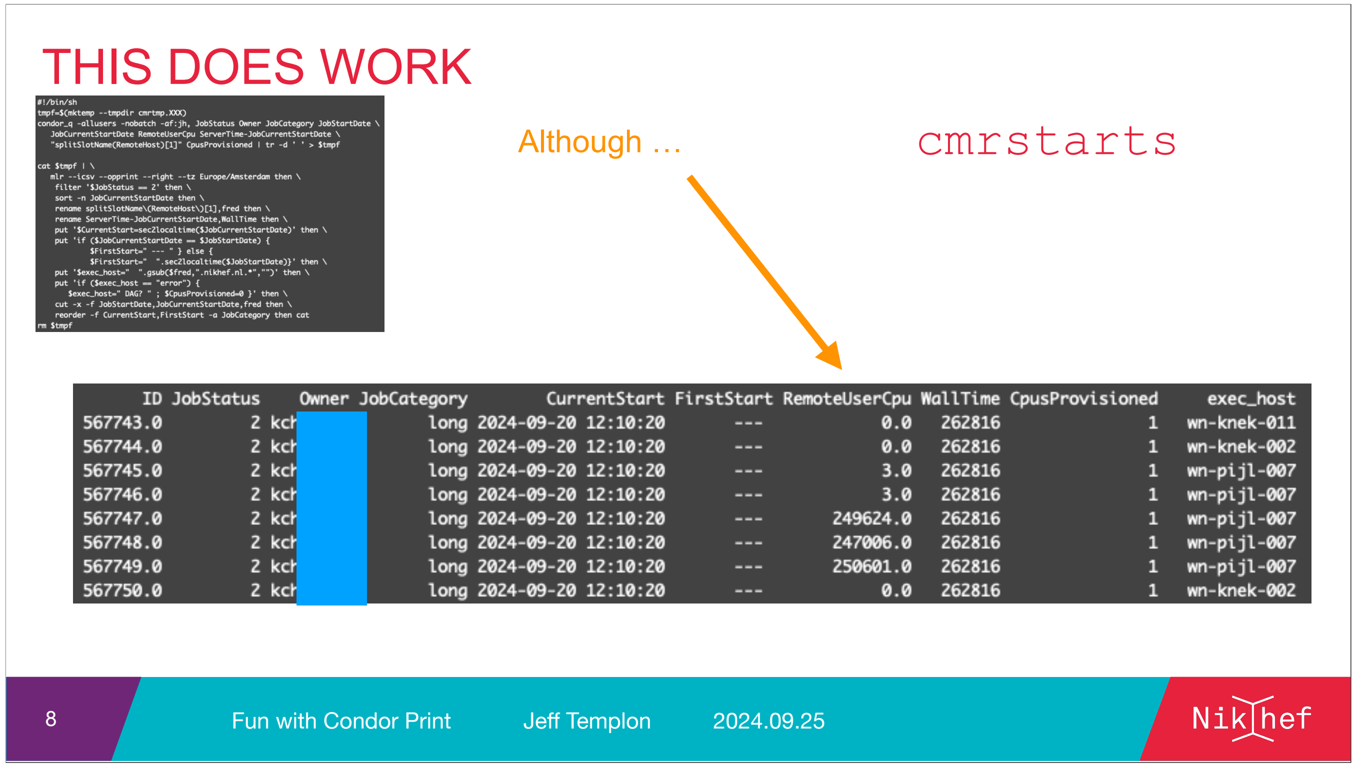
```
#!/bin/sh
tmpf=$(mktemp --tmpdir cmrtmp.XXX)
condor_q -allusers -nobatch -af:jh, JobStatus Owner JobCategory JobStartDate \
   JobCurrentStartDate RemoteUserCpu ServerTime-JobCurrentStartDate \
   "splitSlotName(RemoteHost)[1]" CpusProvisioned | tr -d ' ' > $tmpf

cat $tmpf | \
   mlr -icsv --opprint --right --tz Europe/Amsterdam then \
    filter '$JobStatus == 2' then \
    sort -n JobCurrentStartDate then \
    rename splitSlotName\(RemoteHost\)[1],fred then \
    rename ServerTime-JobCurrentStartDate,WallTime then \
    put '$CurrentStart=sec2localtime($JobCurrentStartDate)' then \
    put 'if ($JobCurrentStartDate == $JobStartDate) {
            $FirstStart=" --- " } else {
            $FirstStart="  ".sec2localtime($JobStartDate)}' then \
    put '$exec_host="  ".gsub($fred,".nikhef.nl.*",""))' then \
    put 'if ($exec_host == "error") {
       $exec_host=" DAG? " ; $CpusProvisioned=0 }' then \
    cut -x -f JobStartDate,JobCurrentStartDate,fred then \
    reorder -f CurrentStart,FirstStart -a JobCategory then cat
rm $tmpf
```

cmrstarts

mlr - awk for csv files

Almost works except close to new years (1 < 12) and the header becomes a footer
Please fix the sort!

This looks complicated, but the Torque equivalent was about 650 lines of Python!



THIS DOES WORK

```
#!/bin/sh
tmpf=$(mktemp --tmpdir cmrtmp.XXX)
condor_q -allusers -nobatch -af:jh, JobStatus Owner JobCategory JobStartDate \
   JobCurrentStartDate RemoteUserCpu ServerTime-JobCurrentStartDate \
   "splitSlotName(RemoteHost)[1]" CpusProvisioned | tr -d ' ' > $tmpf

cat $tmpf | \
   mlr --icsv --opprint --right --tz Europe/Amsterdam then \
    filter '$JobStatus == 2' then \
    sort -n JobCurrentStartDate then \
    rename splitSlotName\(RemoteHost\)[1],fred then \
    rename ServerTime-JobCurrentStartDate,WallTime then \
    put '$CurrentStart=sec2localtime($JobCurrentStartDate)' then \
    put 'if ($JobCurrentStartDate == $JobStartDate) {
            $FirstStart=" --- " } else {
            $FirstStart="  ".sec2localtime($JobStartDate)}' then \
    put '$exec_host="  ".gsub($fred,".nikhef.nl.*","")' then \
    put 'if ($exec_host == "error") {
       $exec_host=" DAG? " ; $CpusProvisioned=0 }' then \
    cut -x -f JobStartDate,JobCurrentStartDate,fred then \
    reorder -f CurrentStart,FirstStart -a JobCategory then cat
rm $tmpf
```

Although …

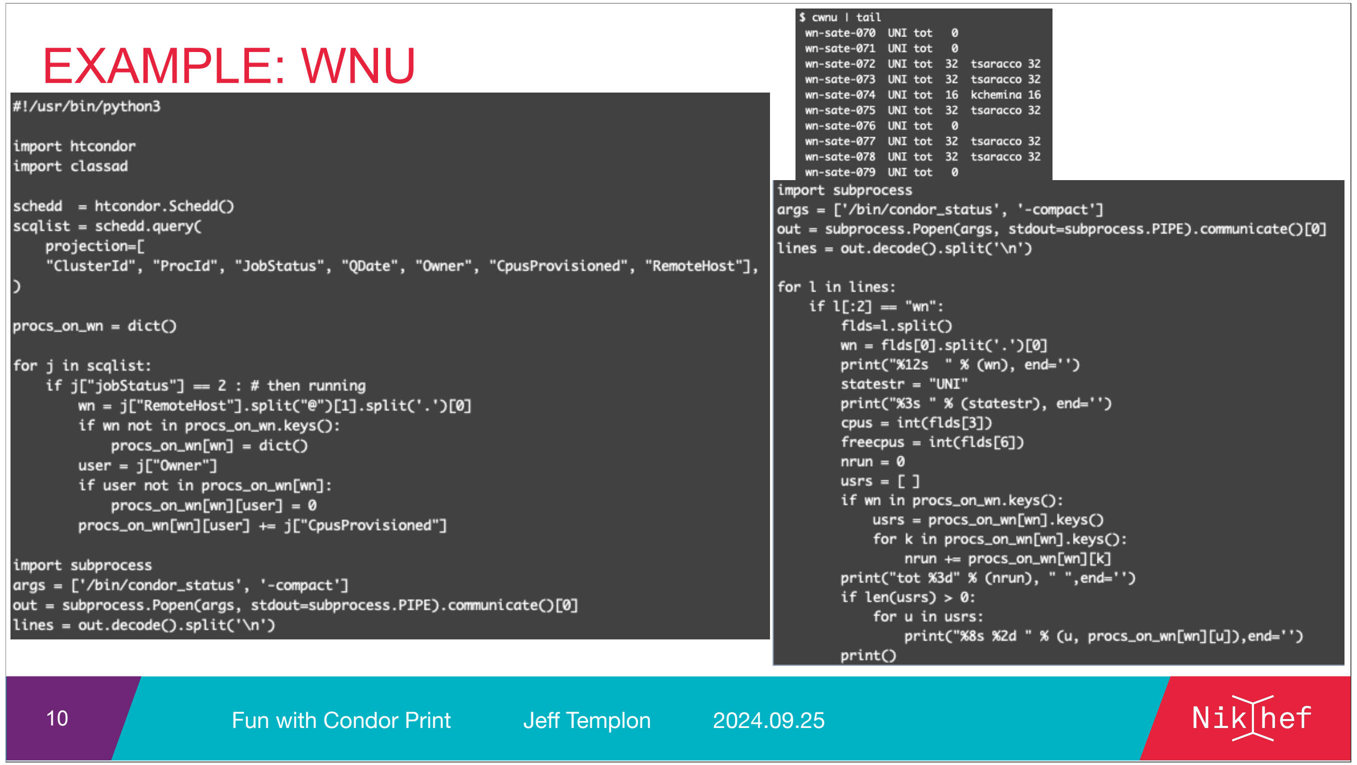
cmrstarts

```
       ID JobStatus    Owner JobCategory        CurrentStart FirstStart RemoteUserCpu WallTime CpusProvisioned    exec_host
567743.0          2 [redacted]        long 2024-09-20 12:10:20       ---           0.0   262816               1  wn-knek-011
567744.0          2 [redacted]        long 2024-09-20 12:10:20       ---           0.0   262816               1  wn-knek-002
567745.0          2 [redacted]        long 2024-09-20 12:10:20       ---           3.0   262816               1  wn-pijl-007
567746.0          2 [redacted]        long 2024-09-20 12:10:20       ---           3.0   262816               1  wn-pijl-007
567747.0          2 [redacted]        long 2024-09-20 12:10:20       ---      249624.0   262816               1  wn-pijl-007
567748.0          2 [redacted]        long 2024-09-20 12:10:20       ---      247006.0   262816               1  wn-pijl-007
567749.0          2 [redacted]        long 2024-09-20 12:10:20       ---      250601.0   262816               1  wn-pijl-007
567750.0          2 [redacted]        long 2024-09-20 12:10:20       ---           0.0   262816               1  wn-knek-002
```

Cross checking, the jobs really are using CPU … for some reason, some jobs report back and others do not.

REMARKS

- The “-af” options to the commands are a vast improvement over Torque (all or nothing, plus an irritating-to-parse format)
- Print format makes things even better (please fix the sort!)
- We need to adapt to split between condor_q and condor_history for “recent” jobs
- Still learning about what info is available



EXAMPLE: WNU

```
#!/usr/bin/python3

import htcondor
import classad

schedd  = htcondor.Schedd()
scqlist = schedd.query(
    projection=[
    "ClusterId", "ProcId", "JobStatus", "QDate", "Owner", "CpusProvisioned", "RemoteHost"],
)

procs_on_wn = dict()

for j in scqlist:
    if j["jobStatus"] == 2 : # then running
        wn = j["RemoteHost"].split("@")[1].split('.')[0]
        if wn not in procs_on_wn.keys():
            procs_on_wn[wn] = dict()
        user = j["Owner"]
        if user not in procs_on_wn[wn]:
            procs_on_wn[wn][user] = 0
        procs_on_wn[wn][user] += j["CpusProvisioned"]

import subprocess
args = ['/bin/condor_status', '-compact']
out = subprocess.Popen(args, stdout=subprocess.PIPE).communicate()[0]
lines = out.decode().split('\n')
```

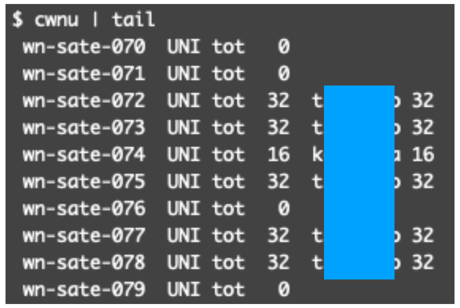
```
$ cwnu | tail
 wn-sate-070  UNI tot   0
 wn-sate-071  UNI tot   0
 wn-sate-072  UNI tot  32  tsaracco 32
 wn-sate-073  UNI tot  32  tsaracco 32
 wn-sate-074  UNI tot  16  kchemina 16
 wn-sate-075  UNI tot  32  tsaracco 32
 wn-sate-076  UNI tot   0
 wn-sate-077  UNI tot  32  tsaracco 32
 wn-sate-078  UNI tot  32  tsaracco 32
 wn-sate-079  UNI tot   0
```

```
import subprocess
args = ['/bin/condor_status', '-compact']
out = subprocess.Popen(args, stdout=subprocess.PIPE).communicate()[0]
lines = out.decode().split('\n')
```

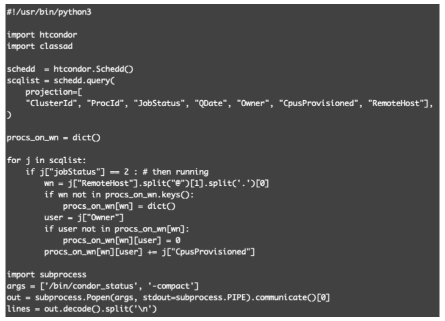
```
for l in lines:
    if l[:2] == "wn":
        flds=l.split()
        wn = flds[0].split('.')[0]
        print("%12s  " % (wn), end='')
        statestr = "UNI"
        print("%3s " % (statestr), end='')
        cpus = int(flds[3])
        freecpus = int(flds[6])
        nrun = 0
        usrs = [ ]
        if wn in procs_on_wn.keys():
            usrs = procs_on_wn[wn].keys()
            for k in procs_on_wn[wn].keys():
                nrun += procs_on_wn[wn][k]
        print("tot %3d" % (nrun), " ",end='')
        if len(usrs) > 0:
            for u in usrs:
                print("%8s %2d " % (u, procs_on_wn[wn][u]),end='')
        print()
```

Split between condor_q and condor_status
Only after building it, realised that all the information is in condor_status
condor_status has a print format as well, may be possible with that

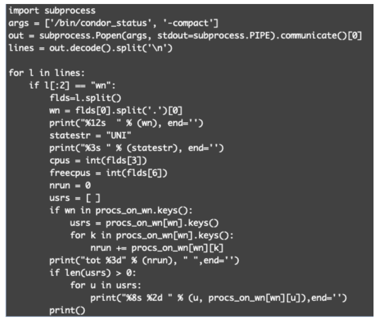
EXAMPLE: WNU



```
$ cwnu | tail
 wn-sate-070  UNI tot   0
 wn-sate-071  UNI tot   0
 wn-sate-072  UNI tot  32  [redacted] 32
 wn-sate-073  UNI tot  32  [redacted] 32
 wn-sate-074  UNI tot  16  [redacted] 16
 wn-sate-075  UNI tot  32  [redacted] 32
 wn-sate-076  UNI tot   0
 wn-sate-077  UNI tot  32  [redacted] 32
 wn-sate-078  UNI tot  32  [redacted] 32
 wn-sate-079  UNI tot   0
```



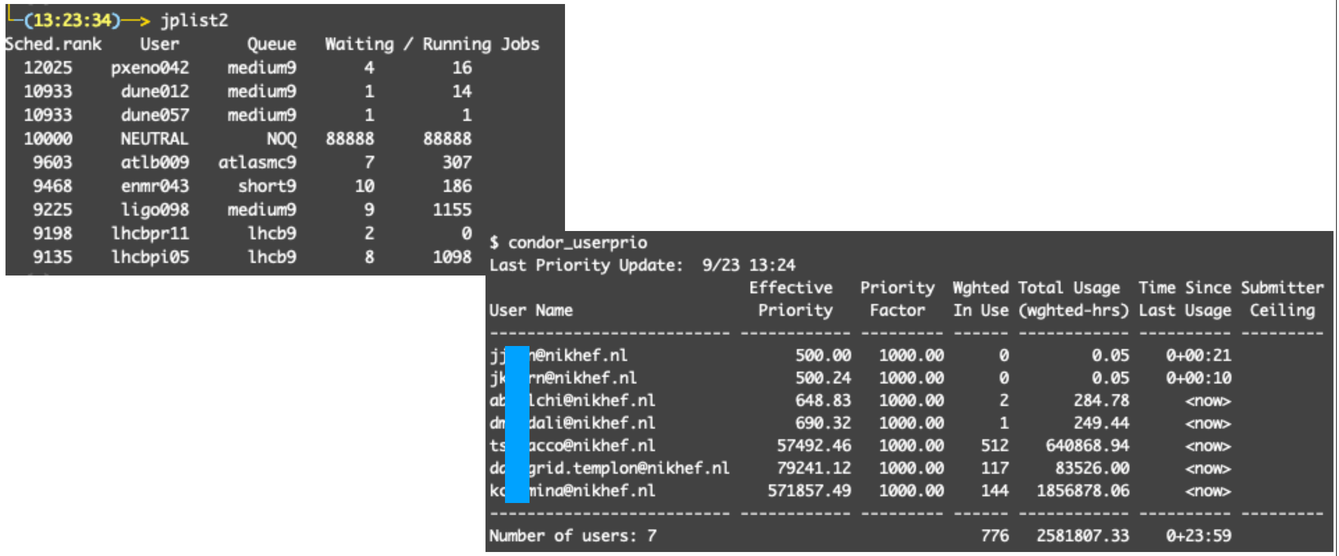
```
#!/usr/bin/python3

import htcondor
import classad

schedd  = htcondor.Schedd()
scqlist = schedd.query(
    projection=[
    "ClusterId", "ProcId", "JobStatus", "QDate", "Owner", "CpusProvisioned", "RemoteHost"],
)

procs_on_wn = dict()

for j in scqlist:
    if j["jobStatus"] == 2 : # then running
        wn = j["RemoteHost"].split("@")[1].split('.')[0]
        if wn not in procs_on_wn.keys():
            procs_on_wn[wn] = dict()
        user = j["Owner"]
        if user not in procs_on_wn[wn]:
            procs_on_wn[wn][user] = 0
        procs_on_wn[wn][user] += j["CpusProvisioned"]

import subprocess
args = ['/bin/condor_status', '-compact']
out = subprocess.Popen(args, stdout=subprocess.PIPE).communicate()[0]
lines = out.decode().split('\n')
```



```
import subprocess
args = ['/bin/condor_status', '-compact']
out = subprocess.Popen(args, stdout=subprocess.PIPE).communicate()[0]
lines = out.decode().split('\n')

for l in lines:
    if l[:2] == "wn":
        flds=l.split()
        wn = flds[0].split('.')[0]
        print("%12s  " % (wn), end='')
        statestr = "UNI"
        print("%3s " % (statestr), end='')
        cpus = int(flds[3])
        freecpus = int(flds[6])
        nrun = 0
        usrs = [ ]
        if wn in procs_on_wn.keys():
            usrs = procs_on_wn[wn].keys()
            for k in procs_on_wn[wn].keys():
                nrun += procs_on_wn[wn][k]
        print("tot %3d" % (nrun), " ",end='')
        if len(usrs) > 0:
            for u in usrs:
                print("%8s %2d " % (u, procs_on_wn[wn][u]),end='')
        print()
```

Split between condor_q and condor_status
Only after building it, realised that all the information is in condor_status
condor_status has a print format as well, may be possible with that

PRIORITIES



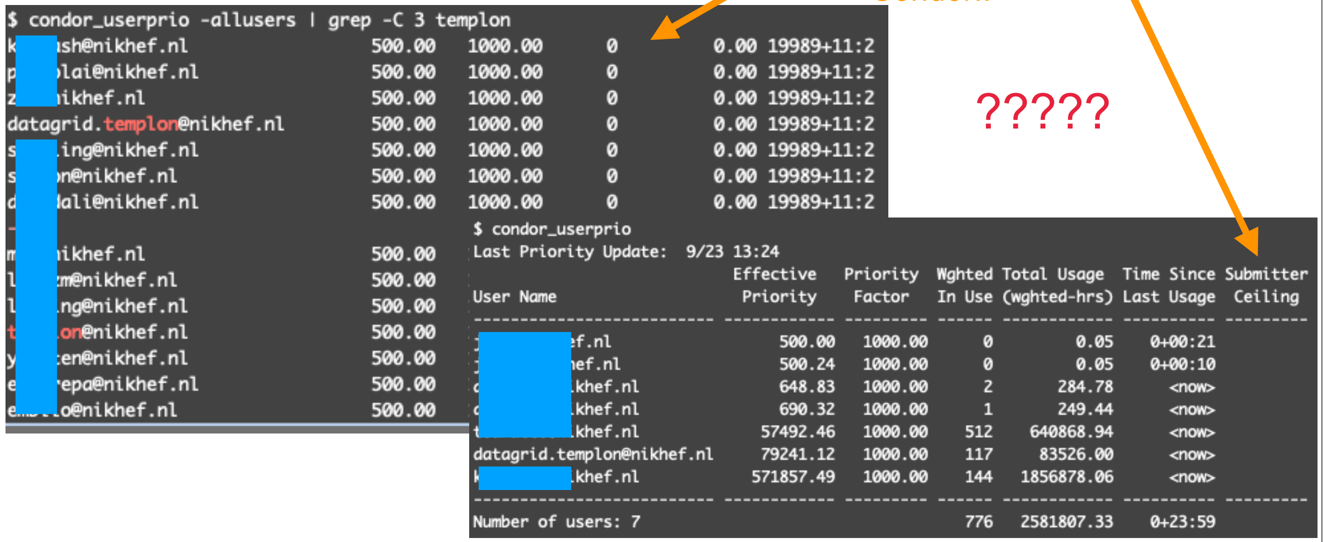
```
└─(13:23:34)──> jplist2
Sched.rank     User       Queue   Waiting / Running Jobs
  12025    pxeno042    medium9       4         16
  10933     dune012    medium9       1         14
  10933     dune057    medium9       1          1
  10000     NEUTRAL        NOQ   88888      88888
   9603     atlb009   atlasmc9       7        307
   9468     enmr043     short9      10        186
   9225     ligo098    medium9       9       1155
   9198    lhcbpr11      lhcb9       2          0
   9135    lhcbpi05      lhcb9       8       1098
```

```
$ condor_userprio
Last Priority Update:  9/23 13:24
                             Effective   Priority  Wghted Total Usage  Time Since Submitter
User Name                     Priority    Factor   In Use (wghted-hrs) Last Usage  Ceiling
---------------------------- ----------- --------- ------ ------------ ---------- ---------
jj[redacted]n@nikhef.nl                 500.00   1000.00      0         0.05    0+00:21
jk[redacted]rn@nikhef.nl                500.24   1000.00      0         0.05    0+00:10
ab[redacted]lchi@nikhef.nl              648.83   1000.00      2       284.78      <now>
dm[redacted]dali@nikhef.nl              690.32   1000.00      1       249.44      <now>
ts[redacted]acco@nikhef.nl            57492.46   1000.00    512    640868.94      <now>
do[redacted]grid.templon@nikhef.nl    79241.12   1000.00    117     83526.00      <now>
ko[redacted]mina@nikhef.nl           571857.49   1000.00    144   1856878.06      <now>
---------------------------- ----------- --------- ------ ------------ ---------- ---------
Number of users: 7                                          776   2581807.33    0+23:59
```

Pretty much what you want, although

PRIORITY MYSTERIES

Mary's Hack from Yesterday …
Most people in list never used Condor!!



```
$ condor_userprio -allusers | grep -C 3 templon
[redacted]ish@nikhef.nl                    500.00   1000.00      0          0.00 19989+11:2
[redacted]lai@nikhef.nl                    500.00   1000.00      0          0.00 19989+11:2
[redacted]ikhef.nl                         500.00   1000.00      0          0.00 19989+11:2
datagrid.templon@nikhef.nl                 500.00   1000.00      0          0.00 19989+11:2
[redacted]ing@nikhef.nl                    500.00   1000.00      0          0.00 19989+11:2
[redacted]n@nikhef.nl                      500.00   1000.00      0          0.00 19989+11:2
[redacted]ali@nikhef.nl                    500.00   1000.00      0          0.00 19989+11:2
-[redacted]
[redacted]ikhef.nl                         500.00
[redacted]m@nikhef.nl                      500.00
[redacted]ng@nikhef.nl                     500.00
[redacted]on@nikhef.nl                     500.00
[redacted]en@nikhef.nl                     500.00
[redacted]epa@nikhef.nl                    500.00
[redacted]o@nikhef.nl                      500.00
```

```
$ condor_userprio
Last Priority Update:  9/23 13:24
                            Effective   Priority  Wghted Total Usage  Time Since Submitter
User Name                    Priority    Factor   In Use (wghted-hrs) Last Usage  Ceiling
--------------------------- ------------ --------- ------ ------------ ---------- ---------
[redacted]ef.nl                 500.00   1000.00      0         0.05    0+00:21
[redacted]ef.nl                 500.24   1000.00      0         0.05    0+00:10
[redacted]khef.nl               648.83   1000.00      2       284.78      <now>
[redacted]khef.nl               690.32   1000.00      1       249.44      <now>
[redacted]khef.nl             57492.46   1000.00    512    640868.94      <now>
datagrid.templon@nikhef.nl    79241.12   1000.00    117     83526.00      <now>
[redacted]khef.nl            571857.49   1000.00    144   1856878.06      <now>
--------------------------- ------------ --------- ------ ------------ ---------- ---------
Number of users: 7                                  776   2581807.33    0+23:59
```

?????

CONCLUSIONS

- Wealth of information from the **`COMMAND LINE`** tools
- Change of paradigm to be expected
- Ability to select fields +++
- Print format capability +++++
- Tools can probably be simpler than now implemented
- Hope for answers to the mysteries at Office Hours
- Please fix the sorting