Git at the PI

Computing at the PI

HTCondor and the D0 cluster

Blake Leverington, with notes from Alexey Zhelezov

The PI Git Repository

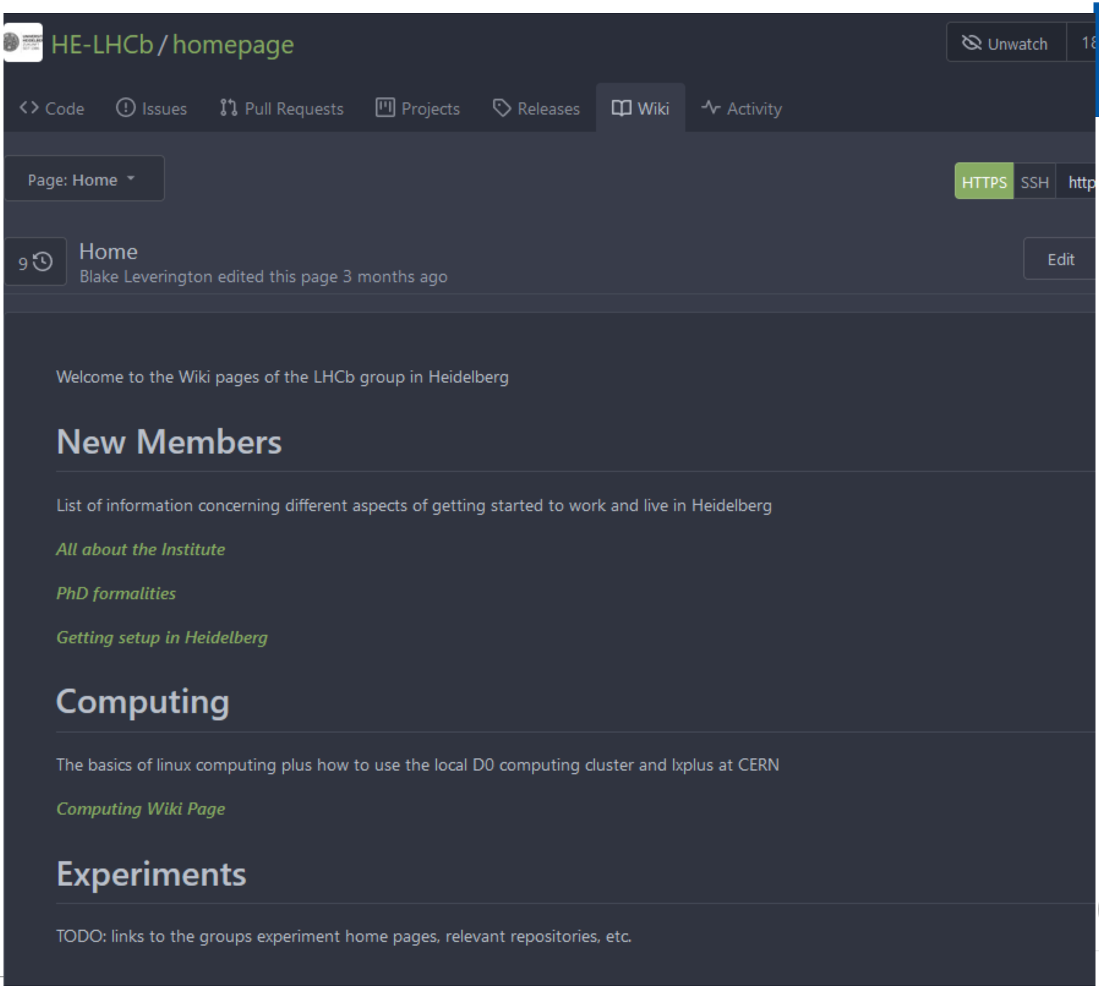
- Login with your PI credentials.
- Not everyone has, or will maintain CERN credentials
- Not everyone is an LHCb member
- Thesis repository and archive: https://git.physi.uni-heidelberg.de/lhcb_theses
  - ~~Not publicly accessible. Only internal.~~
  - Archive your thesis pdf and relevant project work here, for future use by the group.
  - Please upload the figures in a folder, and/or figure data and macros to make them (.C).
  - If you keep some files private (i.e. not archived here), that's fine.
  - How many bachelor, masters and phd projects have disappeared?

- https://git.physi.uni-heidelberg.de/HIT-PAT
- https://git.physi.uni-heidelberg.de/HE-LHCb
- Etc..

The PI Wiki

- https://git.physi.uni-heidelberg.de/HE-LHCB/homepage/wiki

- Easy to edit and create useful pages of information
- Computing How-to’s,
- new joiner information
- Best when used and maintained by everyone
    - Contribute your knowledge
    - What did you wish you knew when you started?



Home

Blake Leverington edited this page 3 months ago

Welcome to the Wiki pages of the LHCb group in Heidelberg

New Members

List of information concerning different aspects of getting started to work and live in Heidelberg

All about the Institute

PhD formalities

Getting setup in Heidelberg

Computing

The basics of linux computing plus how to use the local D0 computing cluster and lxplus at CERN

Computing Wiki Page

Experiments

TODO: links to the groups experiment home pages, relevant repositories, etc.

The computing page:

- https://git.physi.uni-heidelberg.de/HE-LHCb/homepage/wiki/computing

- Some links with useful computing information for beginners in HEP
  - HEP Essentials: Linux, bash scripting, python, jupyter, git, etc.
  - LHCb Starter kits
  - A lot of the LHCb Software Twiki pages are old and outdated.

- How to use the local computing cluster "D0"
- Building the LHCb Software stack

- Please add more…
  - Like Git how-to's, etc

Available systems

- 4 Interactive nodes with Ubuntu environment:
  - Lhcba2: Dual AMD EPYC 7713 64-Core Processors (256 CPU threads), 3.7 GHz, 1 TB RAM
  - Lhcba1: Dual AMD EPYC 7352 24-Core Processors (96 CPU threads), 3.2 GHz, 264 GB RAM
  - d0new: Dual AMD Opteron(tm) 6344 6-core Processor (24 CPU threads), 2.6 GHz, 132 GB RAM
  - delta, Dual Quad-Core AMD Opteron(tm) Processor 2360 SE (8 CPU threads), 2.5 GHz, 16 GB RAM
  - Lhcbi1 (special purpose): Dual socket Intel(R) Xeon(R) Silver 4214 CPU (48 CPU threads), 2.20GHz , 96 GB RAM –
    - Don’t use unless you know you need to use this specific machine
- singularity support under Ubuntu (lhcba2, lhcba1, d0new, delta, batch system)
- batch system (190 slots HTCondor)

```
$ ssh -X -p32  leverington@lhcba2.physi.uni-heidelberg.de
```

Ports:
24 - host Ubuntu
30 - CentOS7
32 – ALMA Linux 9 (RHEL9)

- Storage
  - /scratch on some local machines (eg. build stacks here)
  - /work
    - **Mounted on sigma0 (network connected via NFS)**
    - Relatively high performance 22 TB.
    - daily backups. Please do not use this space for data sets greater then O(100 GB).
  - /auto/data
    - **Mounted on d0new (network connected via GlusterFS)**
    - Slow distributed 260TB storage **for big files.**
    - No backups.
  - /home
    - **Mounted on rho0 (network connected) via NFS**
    - Do not use for anything more than documents.
    - Hosts your e-mail folder, etc.
    - Backed up

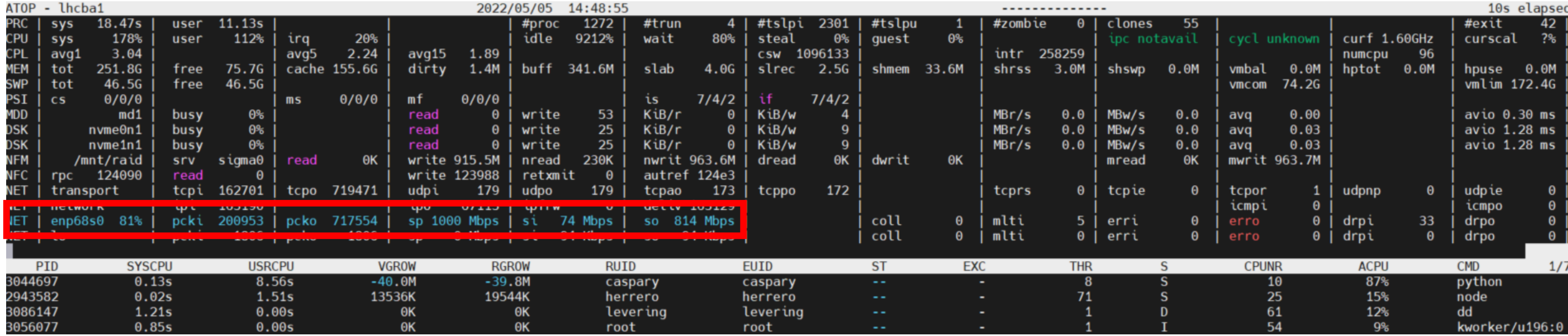
```
$ mkdir -p /work/<yourusername>   #make your own work directory if it doesn’t exist
```

Play nice.

Pay attention to resource consumption! Use `atop` (only Ubuntu –p24) and/or `top`

1000 Mbps network connection → read/write data is the first bottleneck seen usually

No network capacity left means no good connection for your colleagues



```
ATOP - lhcba1                           2022/05/05  14:48:55                          --------------                           10s elapsed
PRC | sys    18.47s | user  11.13s |              |              | #proc   1272 | #trun      4 | #tslpi  2301 | #tslpu     1 | #zombie    0 | clones    55 |               |              | #exit     42
CPU | sys      178% | user    112% | irq      20% |              | idle   9212% | wait     80% | steal     0% | guest     0% |              | ipc notavail | cycl unknown  | curf 1.60GHz | curscal   ?%
CPL | avg1     3.04 |              | avg5    2.24 | avg15   1.89 |              |              | csw  1096133 |              | intr  258259 |              |               | numcpu    96 |
MEM | tot    251.8G | free   75.7G | cache 155.6G | dirty   1.4M | buff  341.6M | slab    4.0G | slrec   2.5G | shmem  33.6M | shrss   3.0M | shswp   0.0M | vmbal   0.0M  | hptot   0.0M | hpuse   0.0M
SWP | tot     46.5G | free   46.5G |              |              |              |              |              |              |              |              | vmcom  74.2G  |              | vmlim 172.4G
PSI | cs      0/0/0 |              | ms     0/0/0 | mf     0/0/0 |              | is     7/4/2 | if     7/4/2 |              |              |              |               |              |
MDD |           md1 | busy      0% |              | read       0 | write     53 | KiB/r      0 | KiB/w      4 |              | MBr/s    0.0 | MBw/s    0.0 | avq      0.00 |              | avio 0.30 ms
DSK |       nvme0n1 | busy      0% |              | read       0 | write     25 | KiB/r      0 | KiB/w      9 |              | MBr/s    0.0 | MBw/s    0.0 | avq      0.03 |              | avio 1.28 ms
DSK |       nvme1n1 | busy      0% |              | read       0 | write     25 | KiB/r      0 | KiB/w      9 |              | MBr/s    0.0 | MBw/s    0.0 | avq      0.03 |              | avio 1.28 ms
NFM |     /mnt/raid | srv   sigma0 | read      0K | write 915.5M | nread   230K | nwrit 963.6M | dread     0K | dwrit     0K |              | mread     0K | mwrit 963.7M  |              |
NFC | rpc    124090 | read       0 |              | write 123988 | retxmit    0 | autref 124e3 |              |              |              |              |               |              |
NET | transport     | tcpi  162701 | tcpo  719471 | udpi     179 | udpo     179 | tcpao    173 | tcppo    172 |              | tcprs      0 | tcpie      0 | tcpor       1 | udpnp      0 | udpie      0
NET | network       | ipi   163196 |              | ipo    67113 | ipfrw      0 | deliv 163129 |              |              |              |              | icmpi       0 |              | icmpo      0
NET | enp68s0   81% | pcki  200953 | pcko  717554 | sp 1000 Mbps | si   74 Mbps | so  814 Mbps |              | coll       0 | mlti       5 | erri       0 | erro        0 | drpi      33 | drpo       0
NET | lo            | pcki    1806 | pcko    1806 | sp    0 Mbps | si   94 Kbps | so   94 Kbps |              | coll       0 | mlti       0 | erri       0 | erro        0 | drpi       0 | drpo       0

    PID     SYSCPU     USRCPU     VGROW     RGROW    RUID       EUID       ST    EXC    THR    S    CPUNR    ACPU    CMD          1/7
3044697      0.13s      8.56s    -40.0M    -39.8M    caspary    caspary    --      -      8    S       10     87%    python
2943582      0.02s      1.51s    13536K    19544K    herrero    herrero    --      -     71    S       25     15%    node
3086147      1.21s      0.00s        0K        0K    levering   levering   --      -      1    D       61     12%    dd
3056077      0.85s      0.00s        0K        0K    root       root       --      -      1    I       54      9%    kworker/u196:0
```

Disk Space

(this seems to be our bottle-kneck most times)

- There are 30-40 people using the resources and only 22 TB on /work
- Easy to fill these days with 500 GB of data sets per person

```
$ du -h --summarize /work/clangenb/ #check the size of your directory
```

```
$ /home/bleverin/software/gdu_linux_amd64 -m 8 /work #check the size of all
directories
```

Compression

Fast and efficient lossless compression with **zstd:**

```
$ zstd -8 –T8 –rm –r --output-dir-mirror -o /work/compressedfolder/*.zst
/work/sourcefolder/*.mdf
```

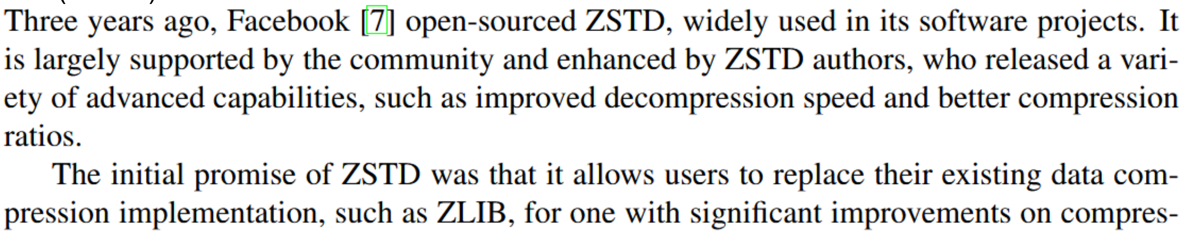
.root files already have some significant compression.

- https://root.cern/manual/io/#compression-and-performance

ROOT offers several options, such as LZMA with very high compression ratio, or LZ4 with very high decompression throughput, or ZSTD with a good compromise in performance. The default compression for RNTuple is ZSTD level 5; for everything else it’s zlib with compression level 1. Algorithm and compression level can be selected using TFile::SetCompressionAlgorithm() and TFile::SetCompressionLevel(), respectively, at the time data is written.

Compression

(2017)

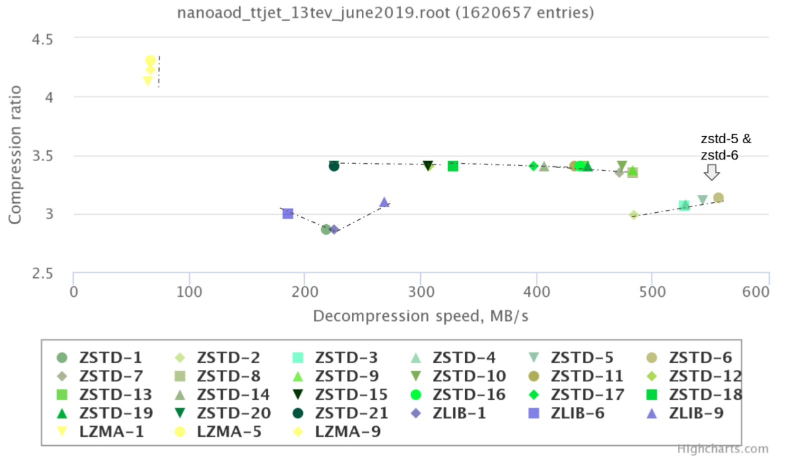


Three years ago, Facebook [7] open-sourced ZSTD, widely used in its software projects. It is largely supported by the community and enhanced by ZSTD authors, who released a variety of advanced capabilities, such as improved decompression speed and better compression ratios.

The initial promise of ZSTD was that it allows users to replace their existing data compression implementation, such as ZLIB, for one with significant improvements on compression speed, compression ratio, and decompression speed. [6]

The ratio here is for a particular standard file.

In LHCb, global .mdf can compress by 5 times!
SciFi NZS by 20 times.



igure 1. Comparison of compression ratio and decompression speed for ZLIB, LZMA and ZSTD
lgorithms for NanoAOD 2019 file

https://www.epj-conferences.org/articles/epjconf/pdf/2020/21/epjconf_chep2020_02017.pdf

Best Practices at the PI (to be discussed)

- use lhcba2 as your interactive login machine.
- Compile your development code and store histogram files and results on /work. Consider using /scratch to build standard stacks, lbconda, etc.
  - /work for things that need to be backed up. Weeks of work lost if something happens.
- Put large data sets on /auto/data
  - things that can be downloaded from elsewhere or rerun in a couple days
- Try not to stream data from cern and run multi-core process on lhcba1/2 for extended periods (>2 minutes)
  - Fine for 1 person, doesn't scale beyond a couple of people on our system.
  - Consider using the/a batch system and /auto/data, keep results on /work, if you are running data/computationally intensive process.
  - Can also be run interactively on the batch system
- Full LHCb analyses should be sent to the GRID via Ganga (lhcba1/2 or lxplus)
- If in doubt, ask Alexey how to best run your data.

Environment Setup

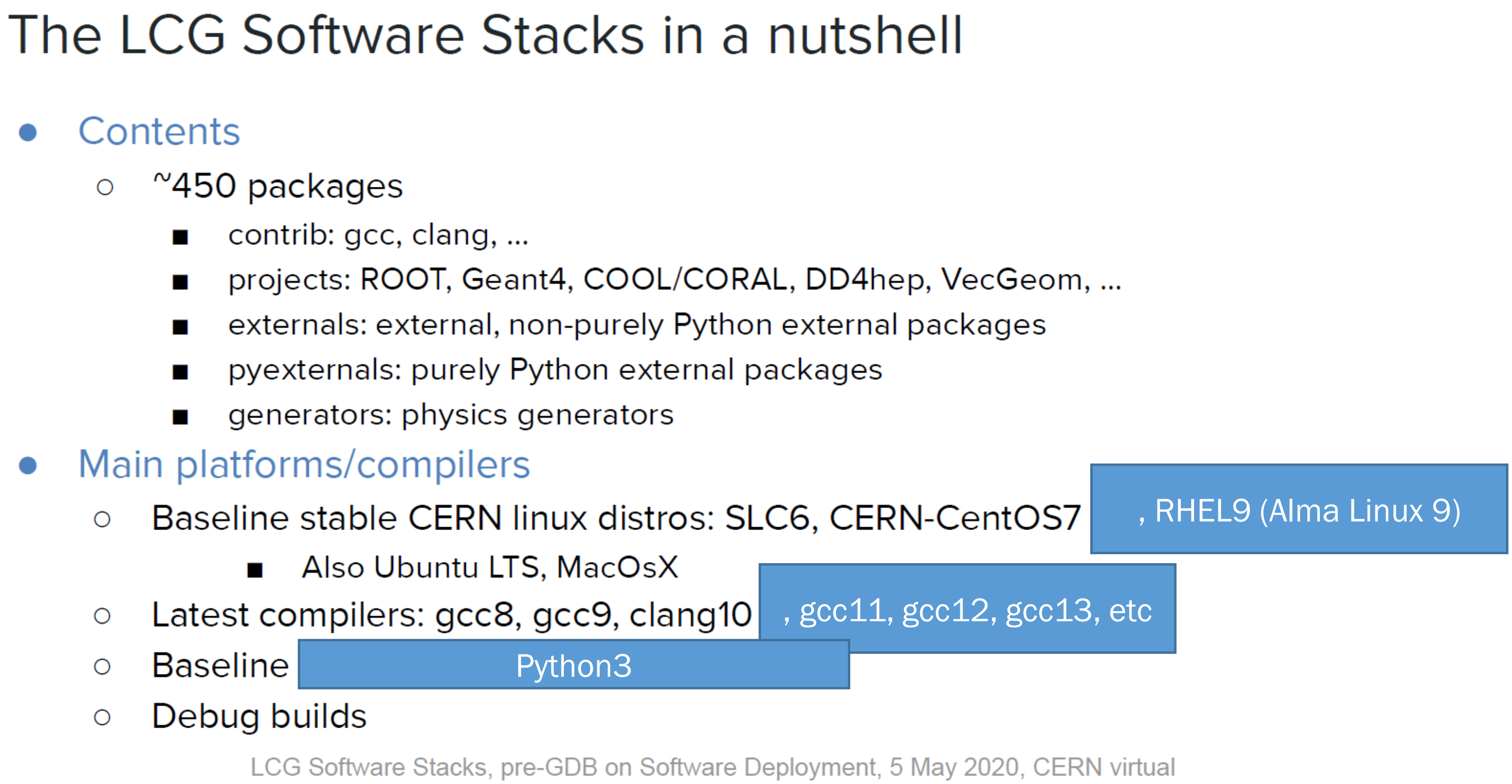
```
#See what is already defined in your environment:
leverington@lhcba2:/work/leverington/lhcb
$ env

#Basic environment which will give you access to the python-based lb-XXXX env-tools:
leverington@lhcba2:/work/leverington/lhcb
$ /cvmfs/lhcb.cern.ch/lib/LbEnv

#Check which platforms are available on the server you logged in to:
leverington@lhcba2:/work/leverington/lhcb
$ lb-describe-platform

#set the environment to point to an LCG software stack
leverington@lhcba2:/work/leverington/lhcb
$ source /cvmfs/sft.cern.ch/lcg/views/LCG_105a/x86_64-el9-gcc12-opt/setup.sh
```

CVMFS

- *CernVM File System (CernVM-FS)*
  - global delivery of scientific software, containers, and auxiliary data
  - ***read-only*** filesystem for those who access it
  - designed to be ***scalable and reliable***, with known deployments involving hundreds of millions of files and many tens of thousands of clients.
  - https://cvmfs-contrib.github.io/cvmfs-tutorial-2021/01_introduction/

- Exactly the same software version and compiled the same way at CERN and here at the PI

LCG Software Stack in CVMFS



The LCG Software Stacks in a nutshell

- Contents
  - ~450 packages
    - contrib: gcc, clang, ...
    - projects: ROOT, Geant4, COOL/CORAL, DD4hep, VecGeom, ...
    - externals: external, non-purely Python external packages
    - pyexternals: purely Python external packages
    - generators: physics generators
- Main platforms/compilers
  - Baseline stable CERN linux distros: SLC6, CERN-CentOS7 , RHEL9 (Alma Linux 9)
    - Also Ubuntu LTS, MacOsX
  - Latest compilers: gcc8, gcc9, clang10 , gcc11, gcc12, gcc13, etc
  - Baseline Python3
  - Debug builds

LCG Software Stacks, pre-GDB on Software Deployment, 5 May 2020, CERN virtual

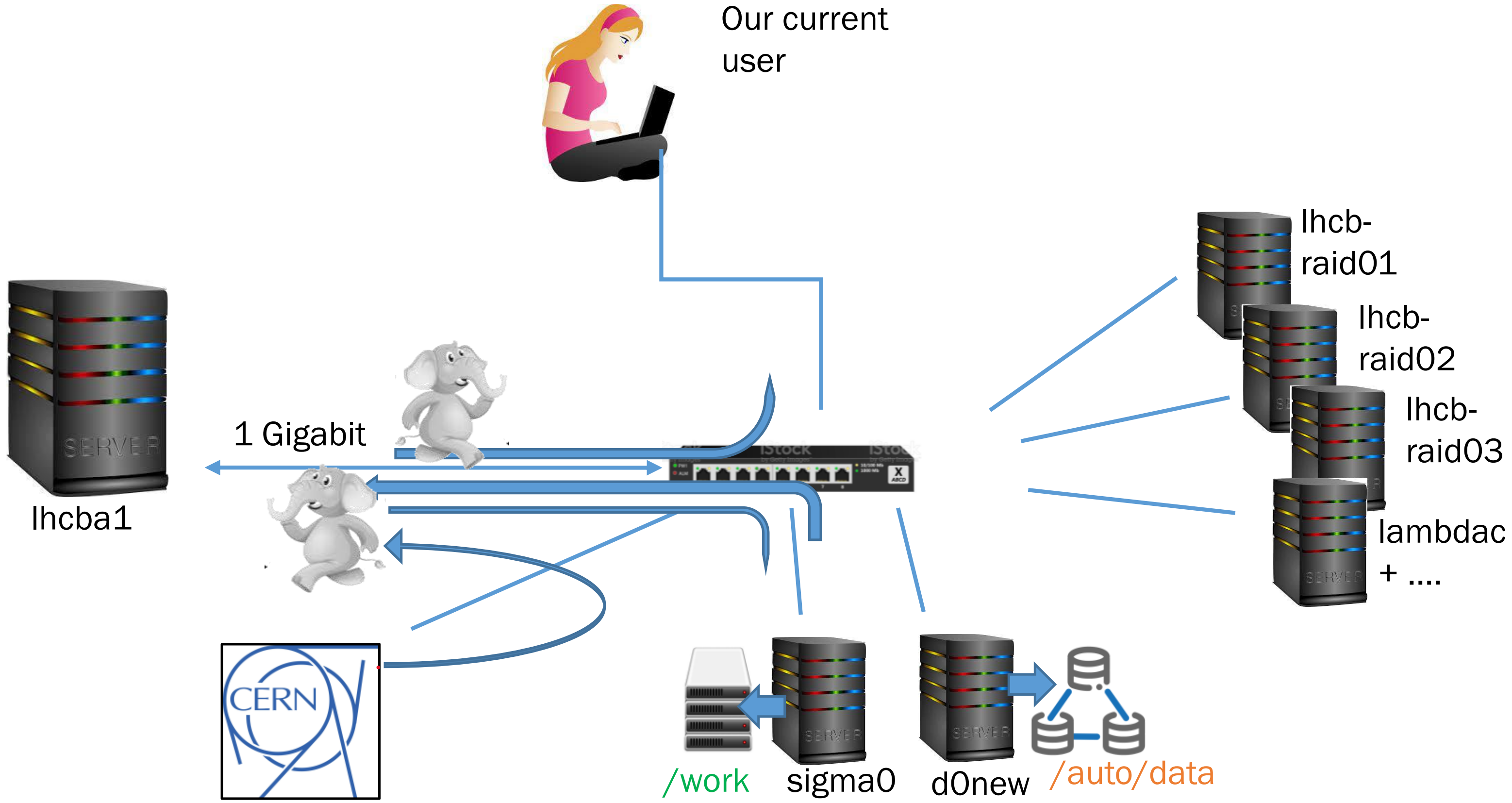
```
leverington@lhcba1:/work/leverington/lhcb
$ source /cvmfs/sft.cern.ch/lcg/views/LCG_101/x86_64-centos7-gcc8-opt/setup.sh
leverington@lhcba1:/work/leverington/lhcb
$ which python
/cvmfs/sft.cern.ch/lcg/views/LCG_101/x86_64-centos7-gcc8-opt/bin/python
leverington@lhcba1:/work/leverington/lhcb
$  python --version
Python 3.9.6
leverington@lhcba1:/work/leverington/lhcb
$ gcc –v
gcc version 8.3.0 (GCC)
leverington@lhcba1:/work/leverington/lhcb
$ which root
$ which root
alias root='root -l'
        /cvmfs/sft.cern.ch/lcg/views/LCG_101/x86_64-centos7-gcc8-opt/bin/root
leverington@lhcba1:/work/leverington/lhcb
$ root --version
ROOT Version: 6.24/06
```

Personal experience: not all LCG versions will work with your code. Depends on compilation options a lot.

LCG_105a seems to be bug free and works with current LHCb software stack.

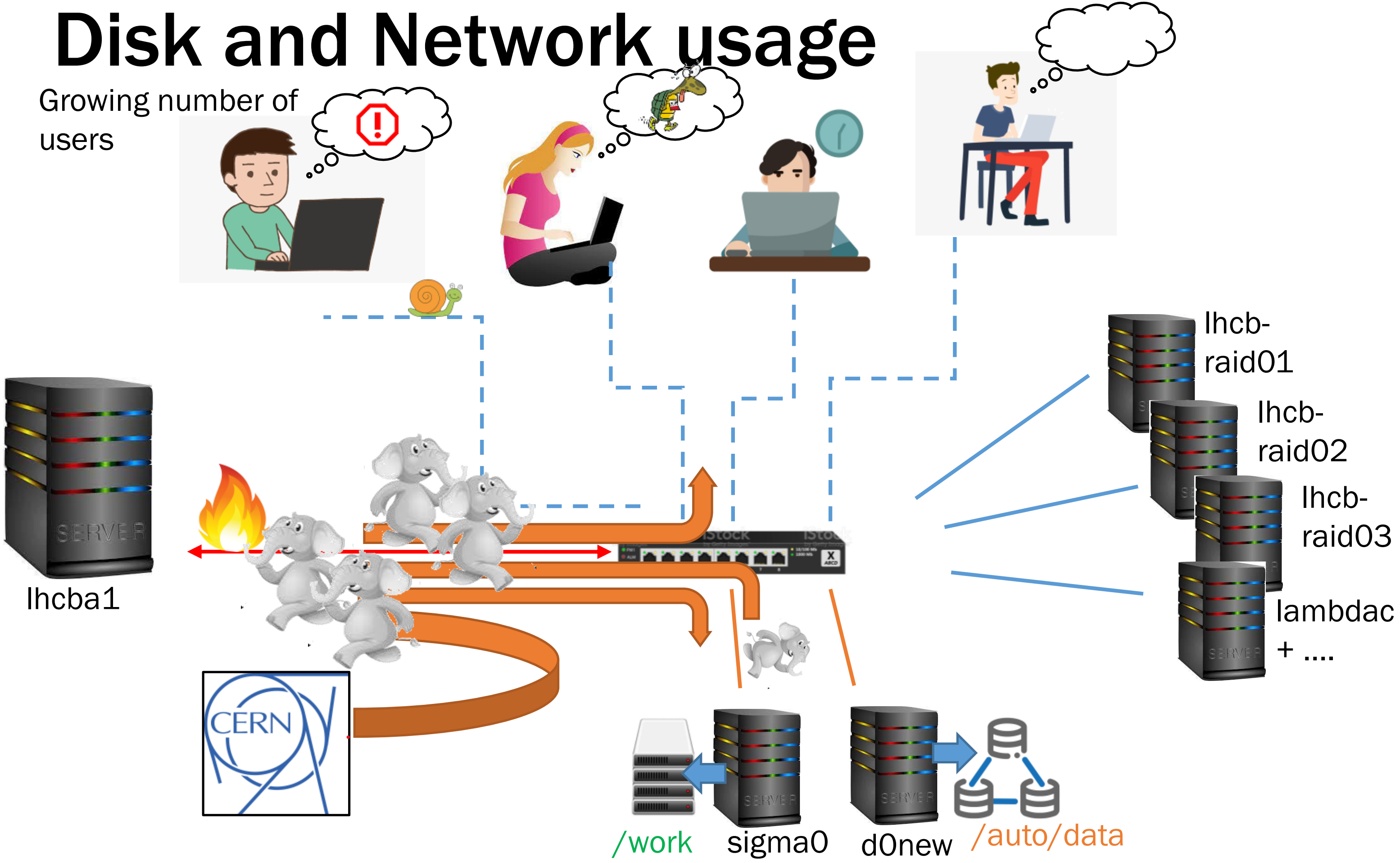
HTCondor and Batch Jobs

Disk and Network usage



Disk and Network usage

Growing number of users

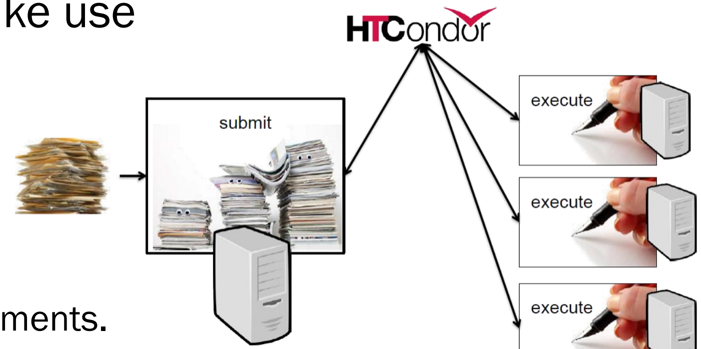


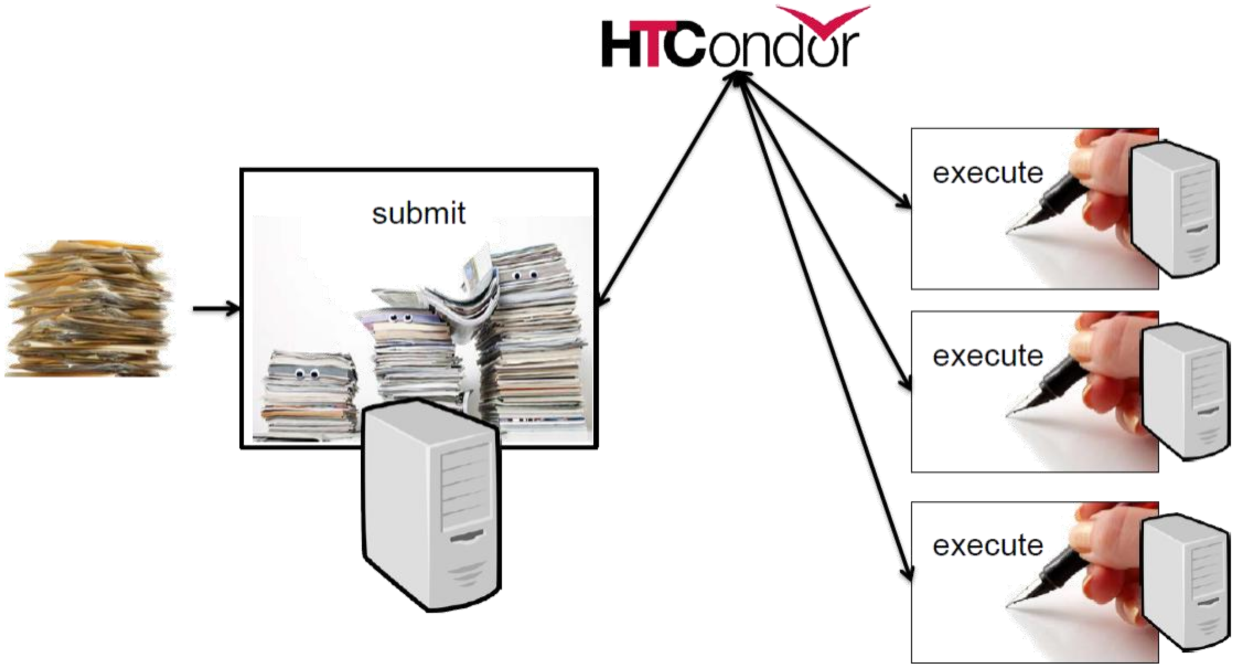
Batch submission via HTCondor

- Split a computational job into multiple jobs to make use of computing resources and your time.

- The job (program) must be executable from the command line (not interactive).
  - You can specify input files, and output files, and pass arguments.
  - We used a shared filesystem in Heidelberg (no need to transfer)

- Will run in a "sandbox". Doesn't know your environment variables unless you pass them or set them up explicitly.

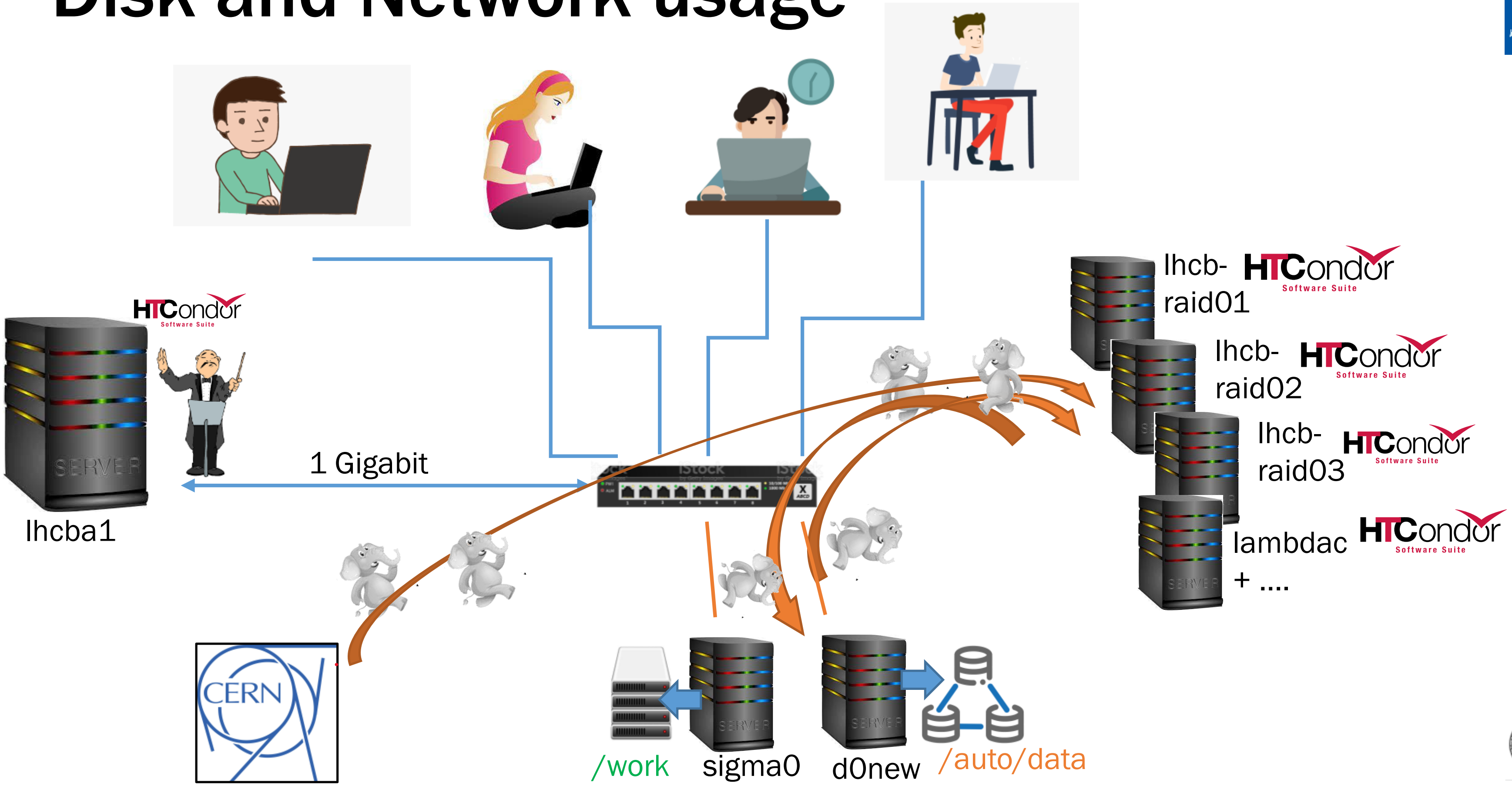




Use case example:

- 1000 events LHCb detector simulation (Gauss) takes ~150 hours on 1 CPU.
  - split it into 20 different 50 event jobs and run on 20 CPUs (each finishes in 8 hours)
  - I can specify when each job is to start (spread them out in time, or run at night to be nice.)
- Many configurable options. https://htcondor.readthedocs.io/en/latest/users-manual/

Disk and Network usage



Preparing a job submission

- Two files to be prepared:

1. The submission file: (.sub)
    - Contains all the parameters that HTCondor needs to run your executable on the available slots of the computing pool

2. The executable: typically a bash script
    - Sets up your environment, folder structure
    - (maybe) takes the arguments passed from the submission file to execute the program in a different way
        - i.e. random number seed, input files, etc.

The job submission file:

```
$ condor_submit examplejob.sub
```

```
#examplejob.sub
# Generic job which will run under local CentOS7 container, on modern servers only  What to run and arguments, can (and probably always
should be) a script
executable            = ./job.sh
arguments             = $(ClusterId) $(ProcId)
# $(ClusterId) is the number assigned to the group of sub-jobs
# $(ProcId)  is the number assigned to each sub-job
# Safe option, so files are not transfer when not required (our servers have access # to the same storage, /auto/work, /auto/data, /home).
should_transfer_files = IF_NEEDED

# Without variable part, output files will be the same for all jobs. # That can be confusing.
output                = output/$(ClusterId)/job.$(ProcId).out
error                 = output/$(ClusterId)/job.$(ProcId).err
log                   = output/$(ClusterId)/job.log

# It is possible apply current environment variables in jobs, with the following # flag. But in general when particular environment is
required, safe option # is directly call the script which setup correct environment (lb-run, etc). In LHCb case # that can also set
optimal platform.
#getenv                = True
# The following flag by itself set 2 following options. It is set by default # when submiting from CentOS7 container, but setting it
explicitly produce
# desired result from Ubuntu as well (when submited from hosts supporting it).
+FromVIRTC = "CentOS7“
# the number of jobs
queue 10
```

https://htcondor.readthedocs.io/en/latest/users-manual/submitting-a-job.html

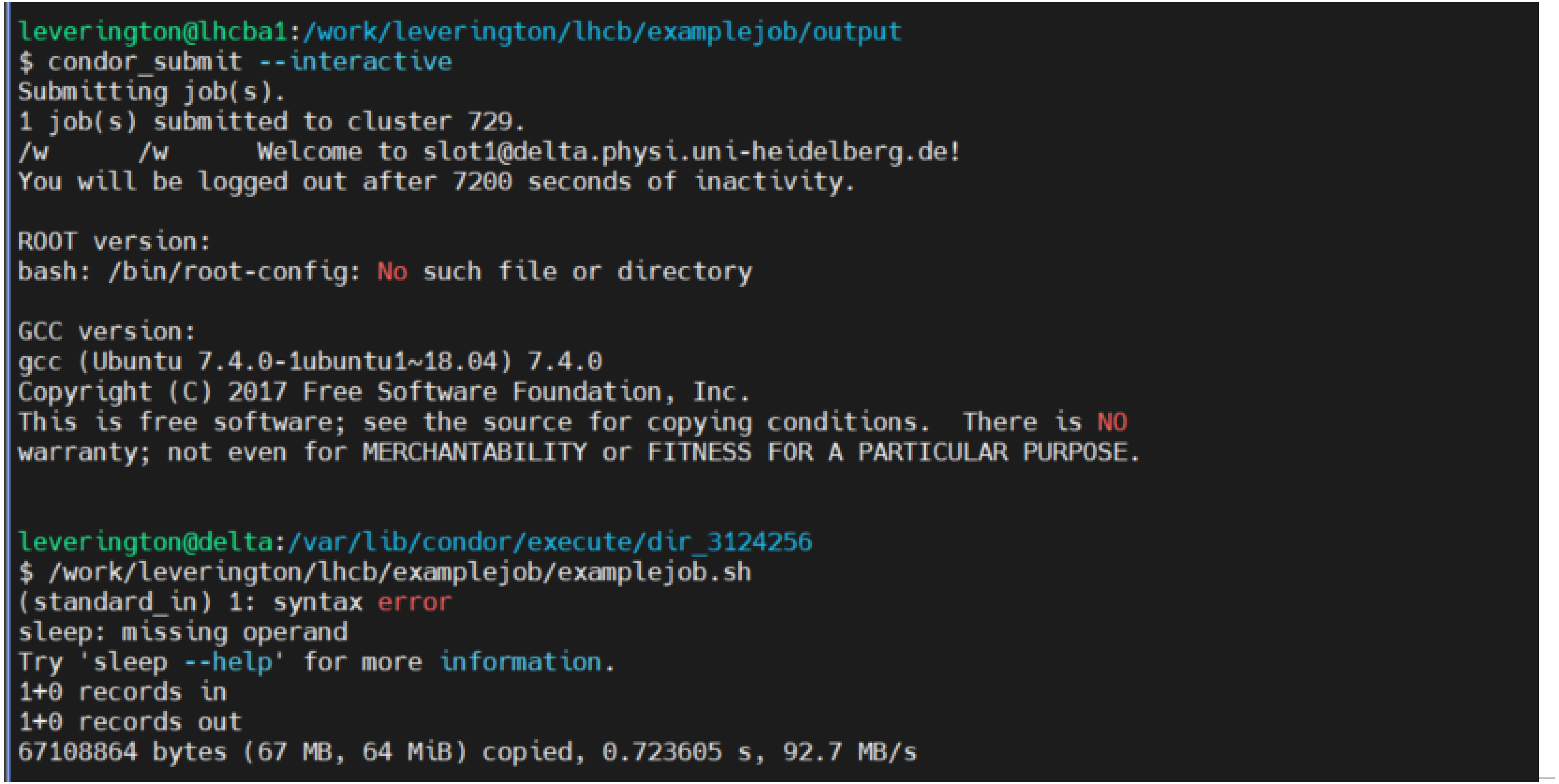
The executable:

- Typically a bash script, but not necessarily. Should be executable on its own.
- https://git.physi.uni-heidelberg.de/HE-LHCb/homepage/wiki/computing_shell_scripts

```
#!/bin/bash
#examplejob.sh

#setup the job folders
HOMEFOLDER=/work/leverington/lhcb/examplejob
#\$1 is the first argument passed to a bash script (job ClusterID in this case)
mkdir -p $HOMEFOLDER/output/${1}/${2}
cd $HOMEFOLDER/output/${1}/${2}

#setup a simple LCG environment
#source /cvmfs/sft.cern.ch/lcg/views/LCG_101/x86_64-centos7-gcc8-opt/setup.sh


#delay each executable so that they do not all start and stop at the same instant. Be nice to the network.
sleeptime=$(echo "$2 * 30" | bc)
sleep $sleeptime


#our executable that we really wanted to perform (something simple to try, write a 1 GB file). Use full paths.
dd if=/dev/zero of=test.file bs=64M count=16 oflag=dsync
```

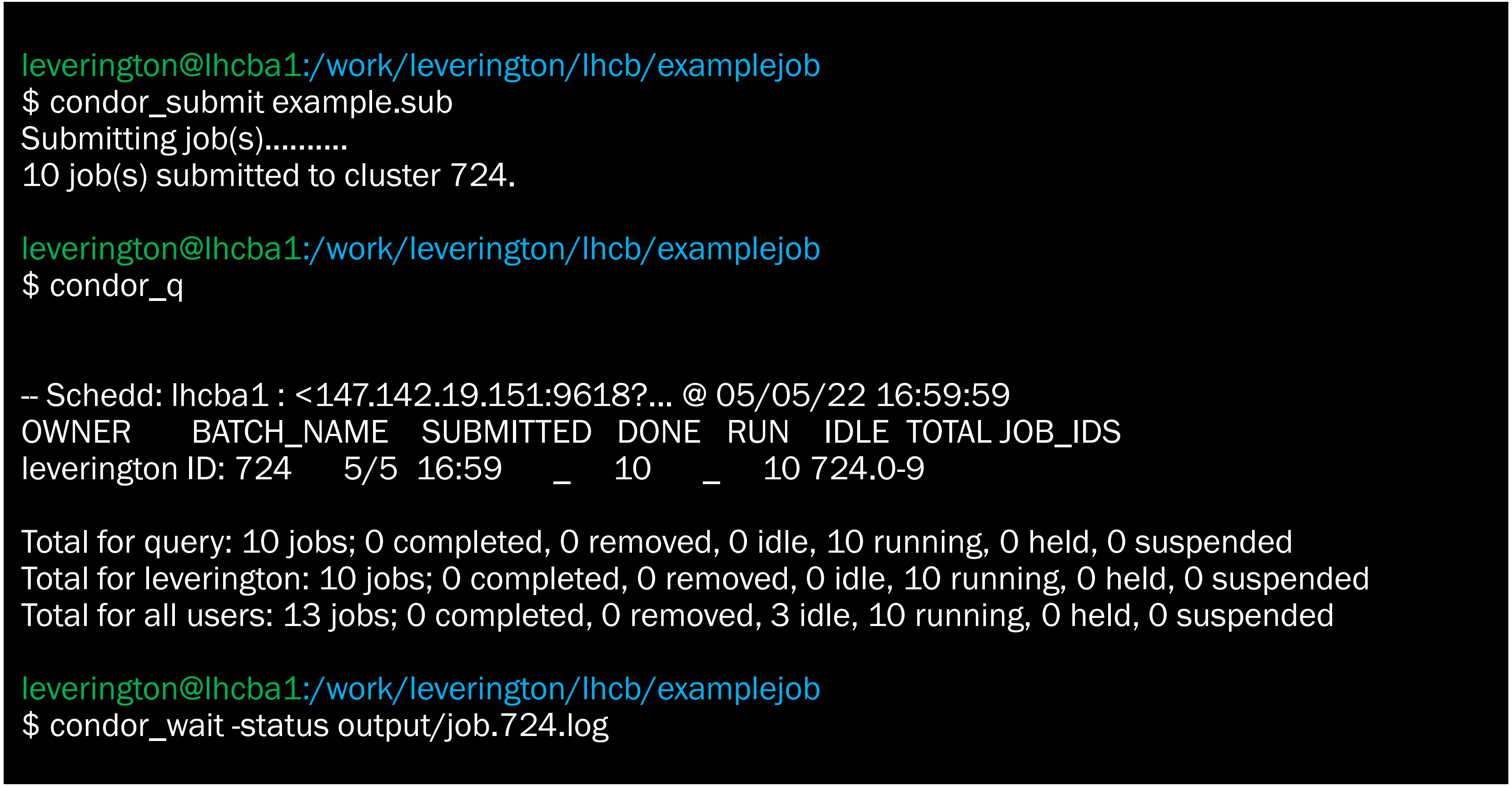
Run htcondor in interactive mode

Run your executable from a condor node. Check for errors.



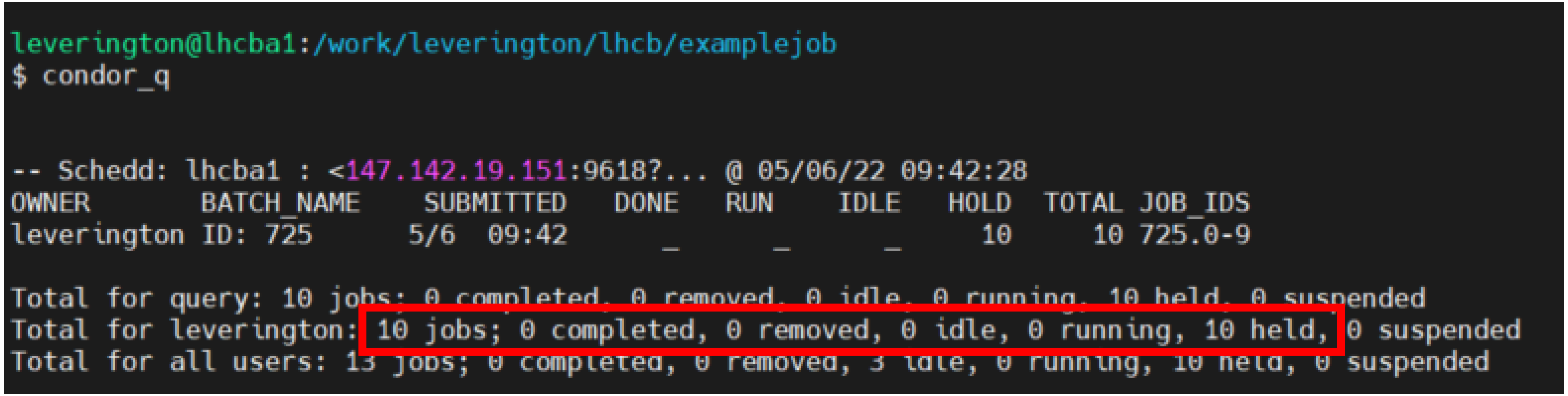
```
leverington@lhcba1:/work/leverington/lhcb/examplejob/output
$ condor_submit --interactive
Submitting job(s).
1 job(s) submitted to cluster 729.
/w      /w      Welcome to slot1@delta.physi.uni-heidelberg.de!
You will be logged out after 7200 seconds of inactivity.

ROOT version:
bash: /bin/root-config: No such file or directory

GCC version:
gcc (Ubuntu 7.4.0-1ubuntu1~18.04) 7.4.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


leverington@delta:/var/lib/condor/execute/dir_3124256
$ /work/leverington/lhcb/examplejob/examplejob.sh
(standard_in) 1: syntax error
sleep: missing operand
Try 'sleep --help' for more information.
1+0 records in
1+0 records out
67108864 bytes (67 MB, 64 MiB) copied, 0.723605 s, 92.7 MB/s
```

Run and Monitor your jobs



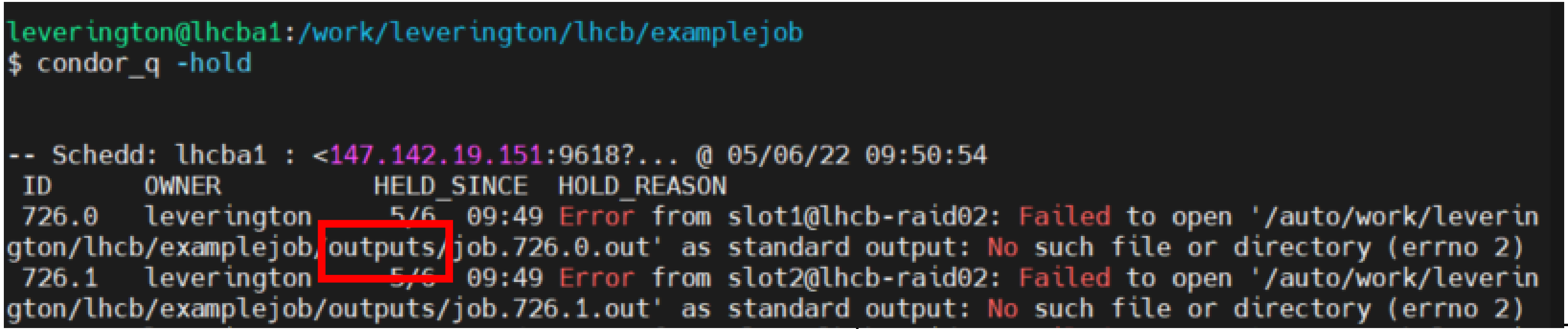
```
leverington@lhcba1:/work/leverington/lhcb/examplejob
$ condor_submit example.sub
Submitting job(s)..........
10 job(s) submitted to cluster 724.

leverington@lhcba1:/work/leverington/lhcb/examplejob
$ condor_q


-- Schedd: lhcba1 : <147.142.19.151:9618?... @ 05/05/22 16:59:59
OWNER       BATCH_NAME    SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
leverington ID: 724      5/5  16:59     _     10      _     10 724.0-9

Total for query: 10 jobs; 0 completed, 0 removed, 0 idle, 10 running, 0 held, 0 suspended
Total for leverington: 10 jobs; 0 completed, 0 removed, 0 idle, 10 running, 0 held, 0 suspended
Total for all users: 13 jobs; 0 completed, 0 removed, 3 idle, 10 running, 0 held, 0 suspended

leverington@lhcba1:/work/leverington/lhcb/examplejob
$ condor_wait -status output/job.724.log
```

Common errors

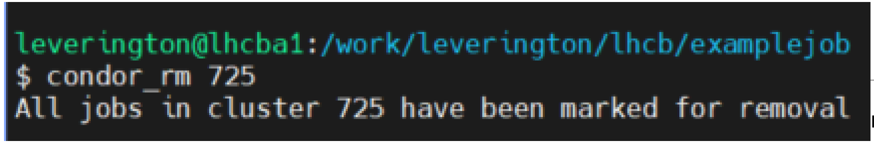


```
leverington@lhcba1:/work/leverington/lhcb/examplejob
$ condor_q


-- Schedd: lhcba1 : <147.142.19.151:9618?... @ 05/06/22 09:42:28
OWNER       BATCH_NAME    SUBMITTED   DONE   RUN    IDLE   HOLD  TOTAL JOB_IDS
leverington ID: 725      5/6  09:42      _      _      _     10     10 725.0-9

Total for query: 10 jobs; 0 completed, 0 removed, 0 idle, 0 running, 10 held, 0 suspended
Total for leverington: 10 jobs; 0 completed, 0 removed, 0 idle, 0 running, 10 held, 0 suspended
Total for all users: 13 jobs; 0 completed, 0 removed, 3 idle, 0 running, 10 held, 0 suspended
```

Cause: my error, stdout/stderr output and logfiles were to be written to a folder that doesn’t exist



```
leverington@lhcba1:/work/leverington/lhcb/examplejob
$ condor_q -hold


-- Schedd: lhcba1 : <147.142.19.151:9618?... @ 05/06/22 09:50:54
 ID       OWNER           HELD_SINCE  HOLD_REASON
 726.0    leverington      5/6  09:49 Error from slot1@lhcb-raid02: Failed to open '/auto/work/leverin
gton/lhcb/examplejob/outputs/job.726.0.out' as standard output: No such file or directory (errno 2)
 726.1    leverington      5/6  09:49 Error from slot2@lhcb-raid02: Failed to open '/auto/work/leverin
gton/lhcb/examplejob/outputs/job.726.1.out' as standard output: No such file or directory (errno 2)
```



```
leverington@lhcba1:/work/leverington/lhcb/examplejob
$ condor_rm 725
All jobs in cluster 725 have been marked for removal
```

Check that one or two small jobs run before submitting anything large and time consuming.

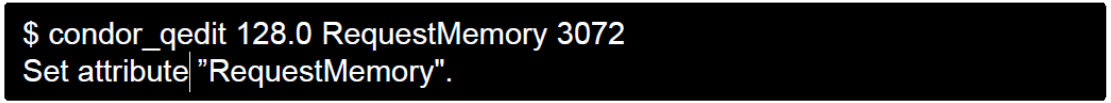
Held Jobs

- HTCondor will put your job on hold if there’s something YOU need to fix.
- A job that goes on hold is interrupted (all progress is lost) and kept from running again, but remains in the queue in the “H” state.

Common Hold Reasons

- Job has used more memory than requested
- Incorrect path to files that need to be transferred
- Badly formatted bash scripts (have Windows instead of Unix line endings)
- Submit directory is over quota
- The admin has put your job on hold

Job attributes can be edited while jobs are in the queue using:
condor_qedit [U/C/J] Attribute Value

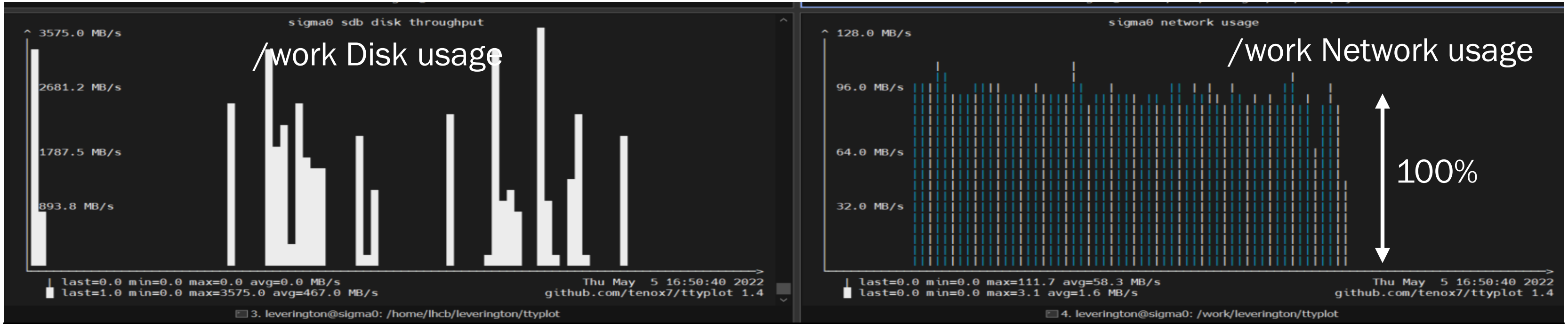


```
$ condor_qedit 128.0 RequestMemory 3072
Set attribute ”RequestMemory”.
```

If a job has been fixed and can run again, release it with:
condor_release [U/C/J]

Disk and Network usage.

A series of jobs all started and finished at once.



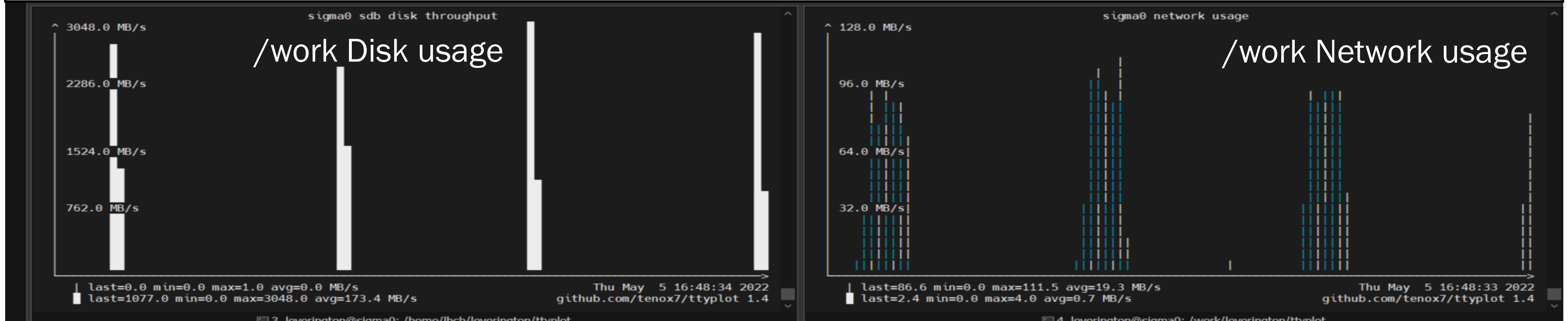
/work Disk usage

/work Network usage

100%

```
| last=0.0 min=0.0 max=0.0 avg=0.0 MB/s                 Thu May  5 16:50:40 2022
█ last=1.0 min=0.0 max=3575.0 avg=467.0 MB/s     github.com/tenox7/ttyplot 1.4
3. leverington@sigma0: /home/lhcb/leverington/ttyplot

| last=0.0 min=0.0 max=111.7 avg=58.3 MB/s              Thu May  5 16:50:40 2022
█ last=0.0 min=0.0 max=3.1 avg=1.6 MB/s          github.com/tenox7/ttyplot 1.4
4. leverington@sigma0: /work/leverington/ttyplot
```

...spaced out by 30 seconds



/work Disk usage

/work Network usage

```
| last=0.0 min=0.0 max=1.0 avg=0.0 MB/s                 Thu May  5 16:48:34 2022
█ last=1077.0 min=0.0 max=3048.0 avg=173.4 MB/s  github.com/tenox7/ttyplot 1.4
3. leverington@sigma0: /home/lhcb/leverington/ttyplot

| last=86.6 min=0.0 max=111.5 avg=19.3 MB/s             Thu May  5 16:48:33 2022
█ last=2.4 min=0.0 max=4.0 avg=0.7 MB/s          github.com/tenox7/ttyplot 1.4
4. leverington@sigma0: /work/leverington/ttyplot
```

Summary

- The resources are there to be used, so use them.
  - But be aware of what you are using and be nice.
  - Share them or make a plan for the best time.

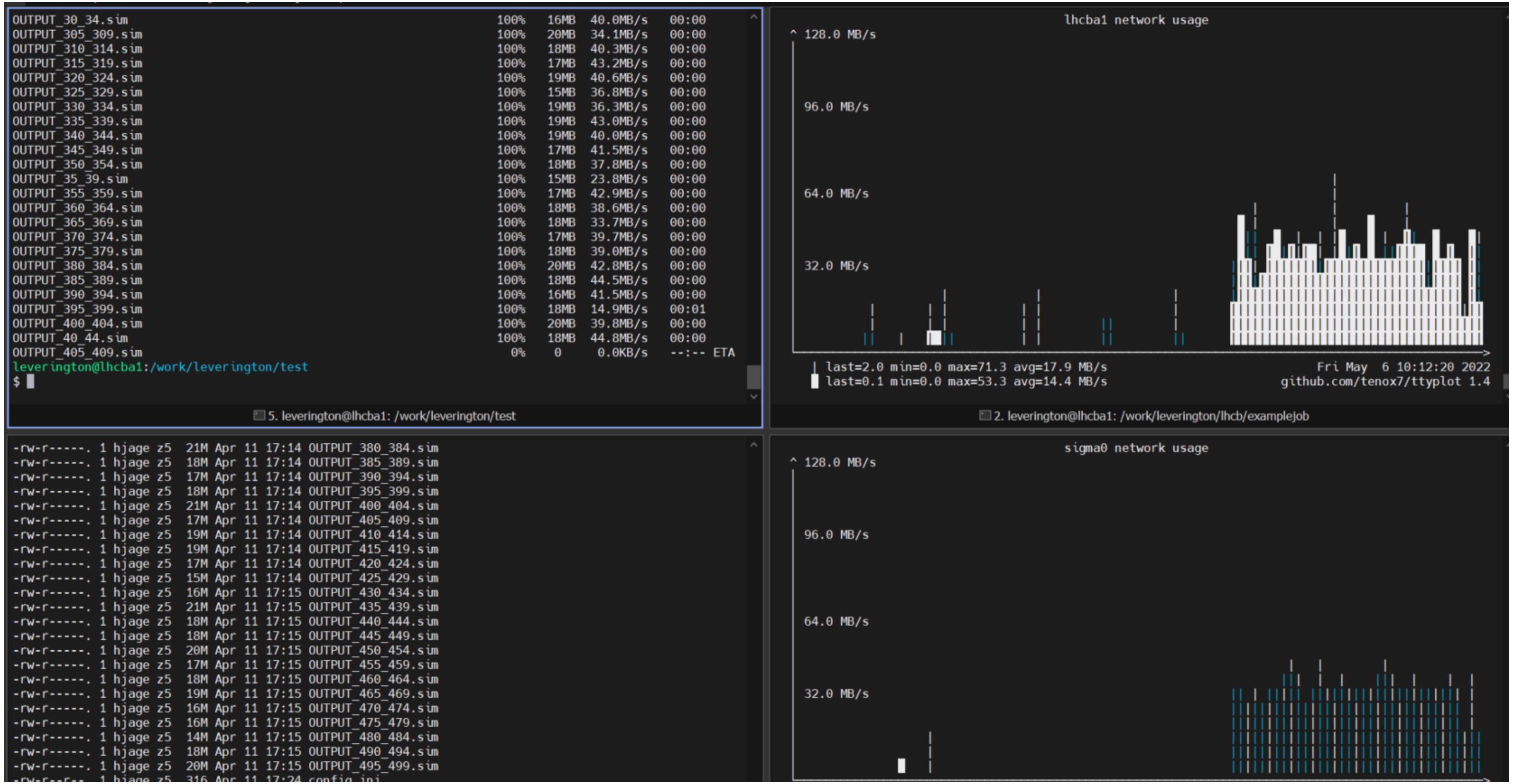
- Mattermost channel: sign up with PI email or CERN email.
  - Probably ask any questions here first.
  - https://mattermost.web.cern.ch/heidelberglhcb/channels/lhcbi1-comrades
- Alexey Zhelezov is the person to contact when you have problems or requests.
  - He is maintaining the servers.

- Network and disk monitoring from the terminal (previous plots) https://github.com/tenox7/ttyplot

- Backup

Disk and Network usage

scp many files from CERN (~40 MBytes/s)



```
OUTPUT_30_34.sim                                              100%   16MB  40.0MB/s   00:00
OUTPUT_305_309.sim                                            100%   20MB  34.1MB/s   00:00
OUTPUT_310_314.sim                                            100%   18MB  40.3MB/s   00:00
OUTPUT_315_319.sim                                            100%   17MB  43.2MB/s   00:00
OUTPUT_320_324.sim                                            100%   19MB  40.6MB/s   00:00
OUTPUT_325_329.sim                                            100%   15MB  36.8MB/s   00:00
OUTPUT_330_334.sim                                            100%   19MB  36.3MB/s   00:00
OUTPUT_335_339.sim                                            100%   19MB  43.0MB/s   00:00
OUTPUT_340_344.sim                                            100%   19MB  40.0MB/s   00:00
OUTPUT_345_349.sim                                            100%   17MB  41.5MB/s   00:00
OUTPUT_350_354.sim                                            100%   18MB  37.8MB/s   00:00
OUTPUT_35_39.sim                                              100%   15MB  23.8MB/s   00:00
OUTPUT_355_359.sim                                            100%   17MB  42.9MB/s   00:00
OUTPUT_360_364.sim                                            100%   18MB  38.6MB/s   00:00
OUTPUT_365_369.sim                                            100%   18MB  33.7MB/s   00:00
OUTPUT_370_374.sim                                            100%   17MB  39.7MB/s   00:00
OUTPUT_375_379.sim                                            100%   18MB  39.0MB/s   00:00
OUTPUT_380_384.sim                                            100%   20MB  42.8MB/s   00:00
OUTPUT_385_389.sim                                            100%   18MB  44.5MB/s   00:00
OUTPUT_390_394.sim                                            100%   16MB  41.5MB/s   00:00
OUTPUT_395_399.sim                                            100%   18MB  14.9MB/s   00:01
OUTPUT_400_404.sim                                            100%   20MB  39.8MB/s   00:00
OUTPUT_40_44.sim                                              100%   18MB  44.8MB/s   00:00
OUTPUT_405_409.sim                                              0%    0     0.0KB/s   --:-- ETA
leverington@lhcba1:/work/leverington/test
$
```

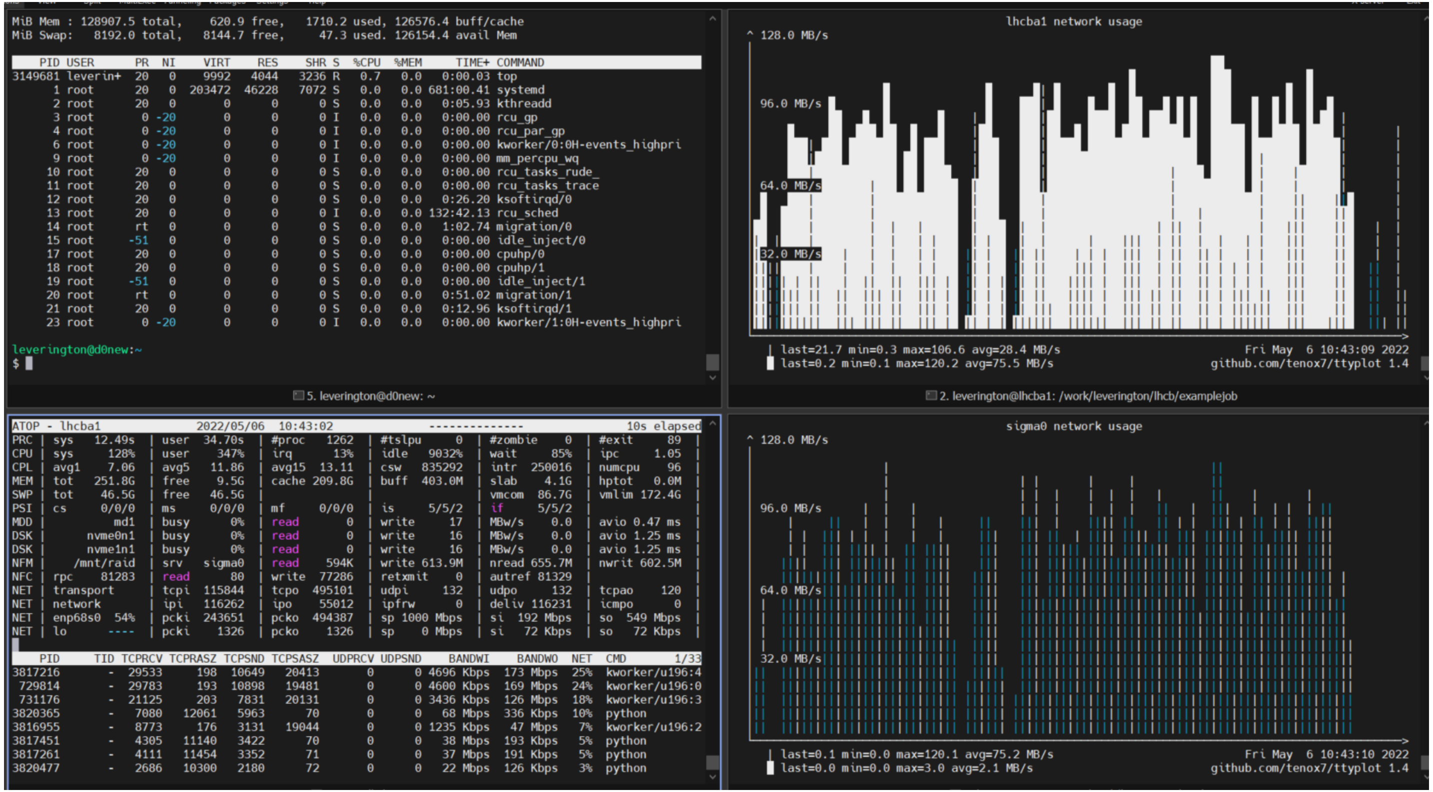
5. leverington@lhcba1: /work/leverington/test

```
lhcba1 network usage

 last=2.0 min=0.0 max=71.3 avg=17.9 MB/s                Fri May  6 10:12:20 2022
 last=0.1 min=0.0 max=53.3 avg=14.4 MB/s             github.com/tenox7/ttyplot 1.4
```

2. leverington@lhcba1: /work/leverington/lhcb/examplejob

```
-rw-r-----. 1 hjage z5  21M Apr 11 17:14 OUTPUT_380_384.sim
-rw-r-----. 1 hjage z5  18M Apr 11 17:14 OUTPUT_385_389.sim
-rw-r-----. 1 hjage z5  17M Apr 11 17:14 OUTPUT_390_394.sim
-rw-r-----. 1 hjage z5  18M Apr 11 17:14 OUTPUT_395_399.sim
-rw-r-----. 1 hjage z5  21M Apr 11 17:14 OUTPUT_400_404.sim
-rw-r-----. 1 hjage z5  17M Apr 11 17:14 OUTPUT_405_409.sim
-rw-r-----. 1 hjage z5  19M Apr 11 17:14 OUTPUT_410_414.sim
-rw-r-----. 1 hjage z5  19M Apr 11 17:14 OUTPUT_415_419.sim
-rw-r-----. 1 hjage z5  17M Apr 11 17:14 OUTPUT_420_424.sim
-rw-r-----. 1 hjage z5  15M Apr 11 17:14 OUTPUT_425_429.sim
-rw-r-----. 1 hjage z5  16M Apr 11 17:15 OUTPUT_430_434.sim
-rw-r-----. 1 hjage z5  21M Apr 11 17:15 OUTPUT_435_439.sim
-rw-r-----. 1 hjage z5  18M Apr 11 17:15 OUTPUT_440_444.sim
-rw-r-----. 1 hjage z5  18M Apr 11 17:15 OUTPUT_445_449.sim
-rw-r-----. 1 hjage z5  20M Apr 11 17:15 OUTPUT_450_454.sim
-rw-r-----. 1 hjage z5  17M Apr 11 17:15 OUTPUT_455_459.sim
-rw-r-----. 1 hjage z5  18M Apr 11 17:15 OUTPUT_460_464.sim
-rw-r-----. 1 hjage z5  19M Apr 11 17:15 OUTPUT_465_469.sim
-rw-r-----. 1 hjage z5  16M Apr 11 17:15 OUTPUT_470_474.sim
-rw-r-----. 1 hjage z5  16M Apr 11 17:15 OUTPUT_475_479.sim
-rw-r-----. 1 hjage z5  14M Apr 11 17:15 OUTPUT_480_484.sim
-rw-r-----. 1 hjage z5  18M Apr 11 17:15 OUTPUT_490_494.sim
-rw-r-----. 1 hjage z5  20M Apr 11 17:15 OUTPUT_495_499.sim
-rw-r--r--. 1 hjage z5  316 Apr 11 17:24 config.ini
```

```
sigma0 network usage
```



```
MiB Mem : 128907.5 total,    620.9 free,   1710.2 used, 126576.4 buff/cache
MiB Swap:   8192.0 total,   8144.7 free,     47.3 used. 126154.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
3149681 leverin+  20   0    9992   4044   3236 R   0.7   0.0   0:00.03 top
      1 root      20   0  203472  46228   7072 S   0.0   0.0 681:00.41 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:05.93 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     12 root      20   0       0      0      0 S   0.0   0.0   0:26.20 ksoftirqd/0
     13 root      20   0       0      0      0 I   0.0   0.0 132:42.13 rcu_sched
     14 root      rt   0       0      0      0 S   0.0   0.0   1:02.74 migration/0
     15 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     19 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     20 root      rt   0       0      0      0 S   0.0   0.0   0:51.02 migration/1
     21 root      20   0       0      0      0 S   0.0   0.0   0:12.96 ksoftirqd/1
     23 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri

leverington@d0new:~
$
```

5. leverington@d0new: ~

```
                       lhcba1 network usage
^ 128.0 MB/s

  96.0 MB/s

  64.0 MB/s

  32.0 MB/s

 | last=21.7 min=0.3 max=106.6 avg=28.4 MB/s          Fri May  6 10:43:09 2022
 █ last=0.2 min=0.1 max=120.2 avg=75.5 MB/s      github.com/tenox7/ttyplot 1.4
```

2. leverington@lhcba1: /work/leverington/lhcb/examplejob

```
ATOP - lhcba1             2022/05/06  10:43:02             --------------            10s elapsed
PRC | sys    12.49s | user  34.70s | #proc   1262 | #tslpu      0 | #zombie    0 | #exit     89 |
CPU | sys      128% | user   347%  | irq      13% | idle    9032% | wait     85% | ipc     1.05 |
CPL | avg1     7.06 | avg5   11.86 | avg15  13.11 | csw   835292  | intr  250016 | numcpu    96 |
MEM | tot    251.8G | free    9.5G | cache 209.8G | buff  403.0M  | slab    4.1G | hptot   0.0M |
SWP | tot     46.5G | free   46.5G |              |               | vmcom  86.7G | vmlim 172.4G |
PSI | cs      0/0/0 | ms     0/0/0 | mf     0/0/0 | is      5/5/2 | if     5/5/2 |              |
MDD |           md1 | busy      0% | read       0 | write      17 | MBw/s    0.0 | avio 0.47 ms |
DSK |       nvme0n1 | busy      0% | read       0 | write      16 | MBw/s    0.0 | avio 1.25 ms |
DSK |       nvme1n1 | busy      0% | read       0 | write      16 | MBw/s    0.0 | avio 1.25 ms |
NFM |     /mnt/raid | srv   sigma0 | read    594K | write 613.9M  | nread 655.7M | nwrit 602.5M |
NFC | rpc     81283 | read      80 | write  77286 | retxmit     0 | autref 81329 |              |
NET | transport     | tcpi  115844 | tcpo  495101 | udpi      132 | udpo     132 | tcpao    120 |
NET | network       | ipi   116262 | ipo    55012 | ipfrw       0 | deliv 116231 | icmpo      0 |
NET | enp68s0   54% | pcki  243651 | pcko  494387 | sp 1000 Mbps  | si  192 Mbps | so  549 Mbps |
NET | lo       ---- | pcki    1326 | pcko    1326 | sp     0 Mbps | si   72 Kbps | so   72 Kbps |

    PID      TID TCPRCV TCPRASZ TCPSND TCPSASZ  UDPRCV UDPSND      BANDWI      BANDWO  NET  CMD         1/33
3817216        -  29533     198  10649   20413       0      0 4696 Kbps   173 Mbps  25%  kworker/u196:4
 729814        -  29783     193  10898   19481       0      0 4600 Kbps   169 Mbps  24%  kworker/u196:0
 731176        -  21125     203   7831   20131       0      0 3436 Kbps   126 Mbps  18%  kworker/u196:3
3820365        -   7080   12061   5963      70       0      0   68 Mbps   336 Kbps  10%  python
3816955        -   8773     176   3131   19044       0      0 1235 Kbps    47 Mbps   7%  kworker/u196:2
3817451        -   4305   11140   3422      70       0      0   38 Mbps   193 Kbps   5%  python
3817261        -   4111   11454   3352      71       0      0   37 Mbps   191 Kbps   5%  python
3820477        -   2686   10300   2180      72       0      0   22 Mbps   126 Kbps   3%  python
```

```
                       sigma0 network usage
^ 128.0 MB/s

  96.0 MB/s

  64.0 MB/s

  32.0 MB/s

 | last=0.1 min=0.0 max=120.1 avg=75.2 MB/s           Fri May  6 10:43:10 2022
 █ last=0.0 min=0.0 max=3.0 avg=2.1 MB/s         github.com/tenox7/ttyplot 1.4
```