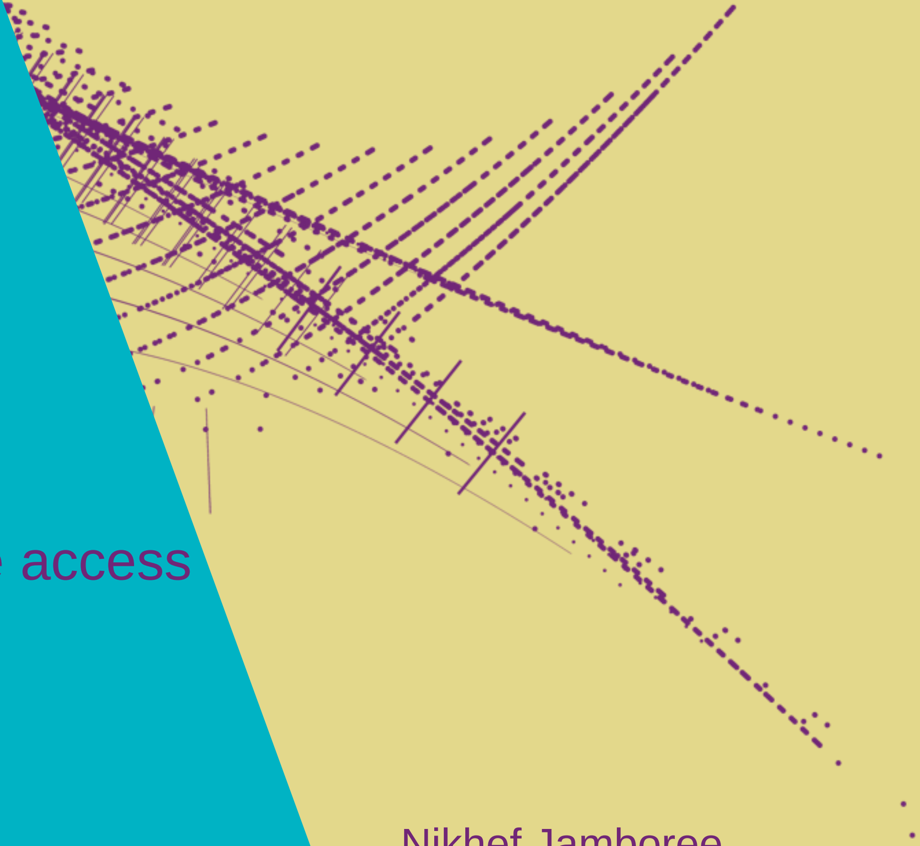
Nikhef

Maastricht University



PDP
infrastructure, algorithms, collaborative access

And the winner is …

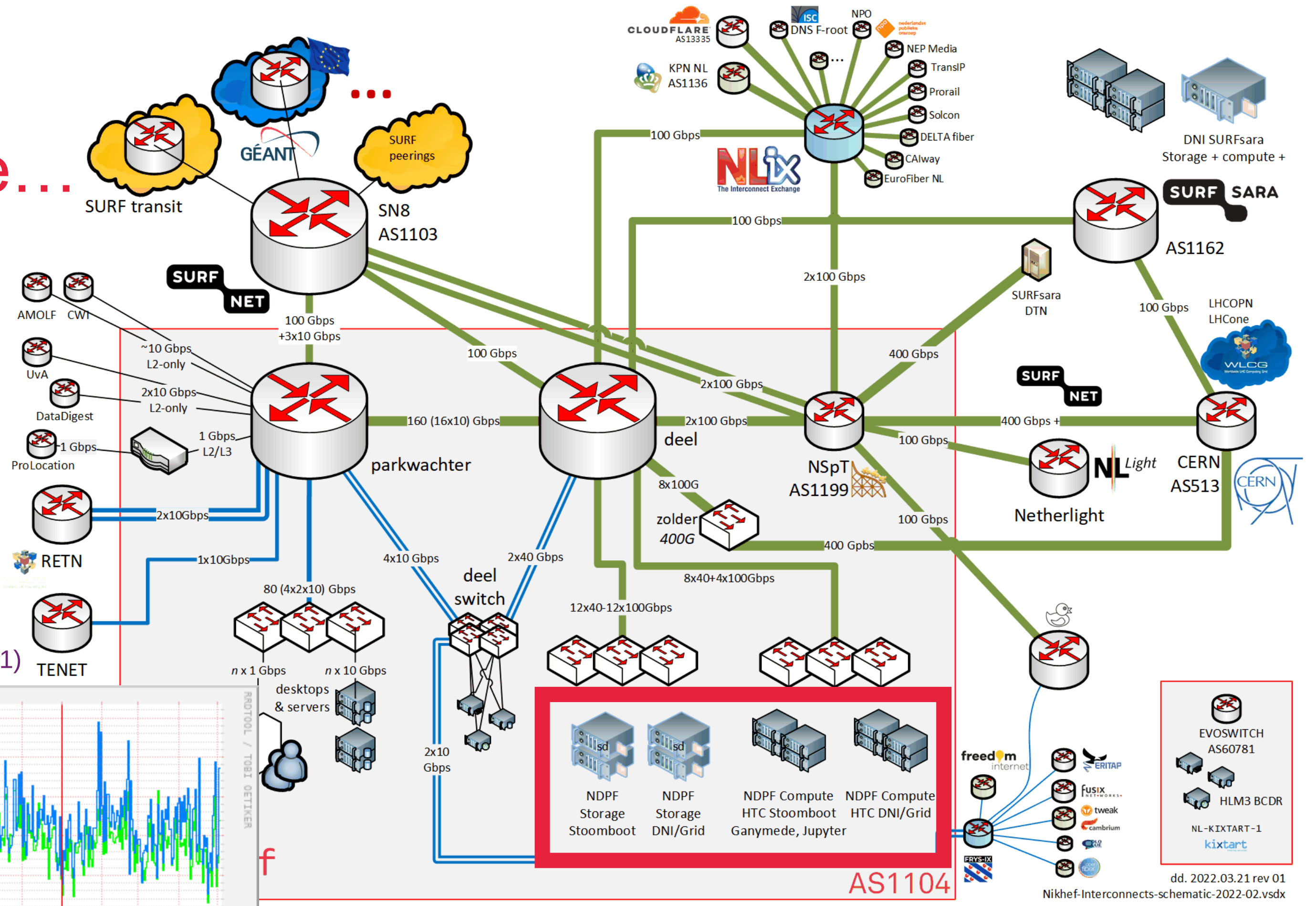
Nikhef Jamboree
May 2022

For a box full of disks … (and a fistful of euro’s)

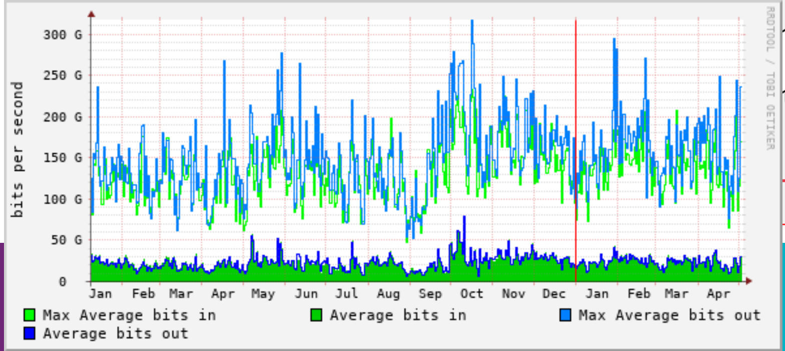
84x16TByte gross capacity, so ~ 1PByte net after redundancy overhead

For a few boxes more…

connected to the world …



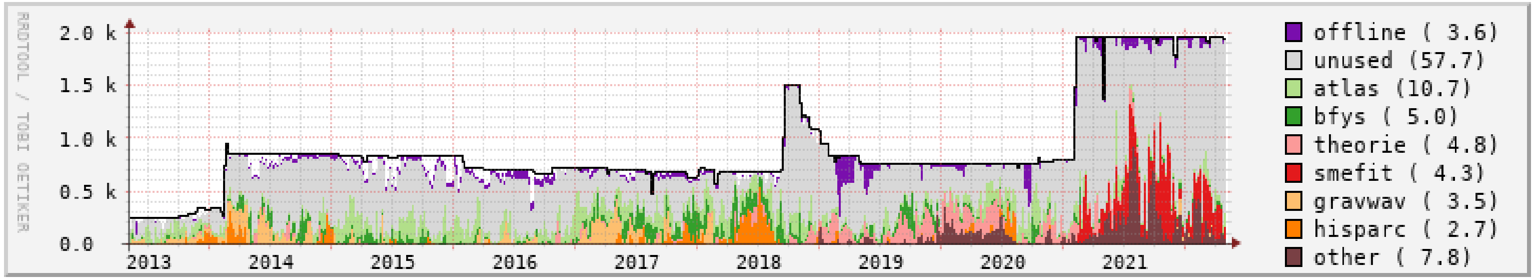
total traffic through data processing core
(using harbour measurement method, 2021)



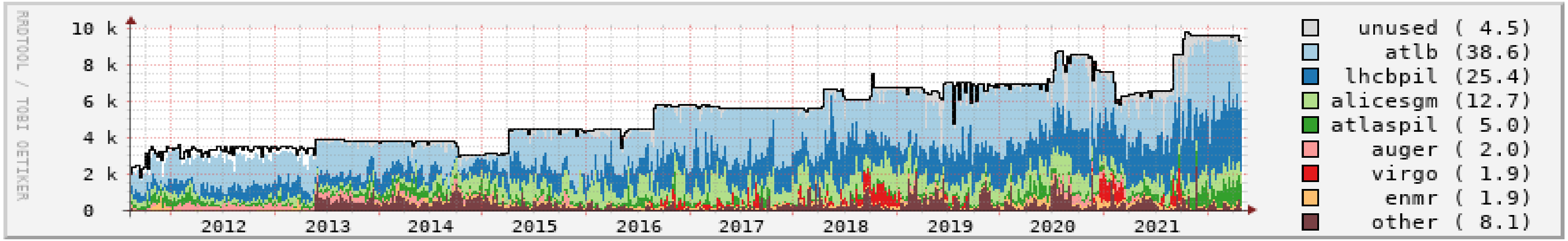
https://www.nikhef.nl/pdp/doc/facility

But effective usage differs …

Stoomboot

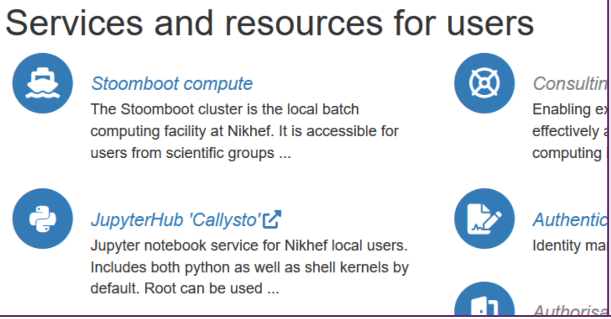


Grid (NL-T1, SURF National e-Infra @Nikhef AMS)



https://www.nikhef.nl/pdp/stats/

Of course you analyze small things in an executable Jupyter paper



Services and resources for users

Stoomboot compute

The Stoomboot cluster is the local batch computing facility at Nikhef. It is accessible for users from scientific groups ...

Consultin

Enabling ex
effectively a
computing

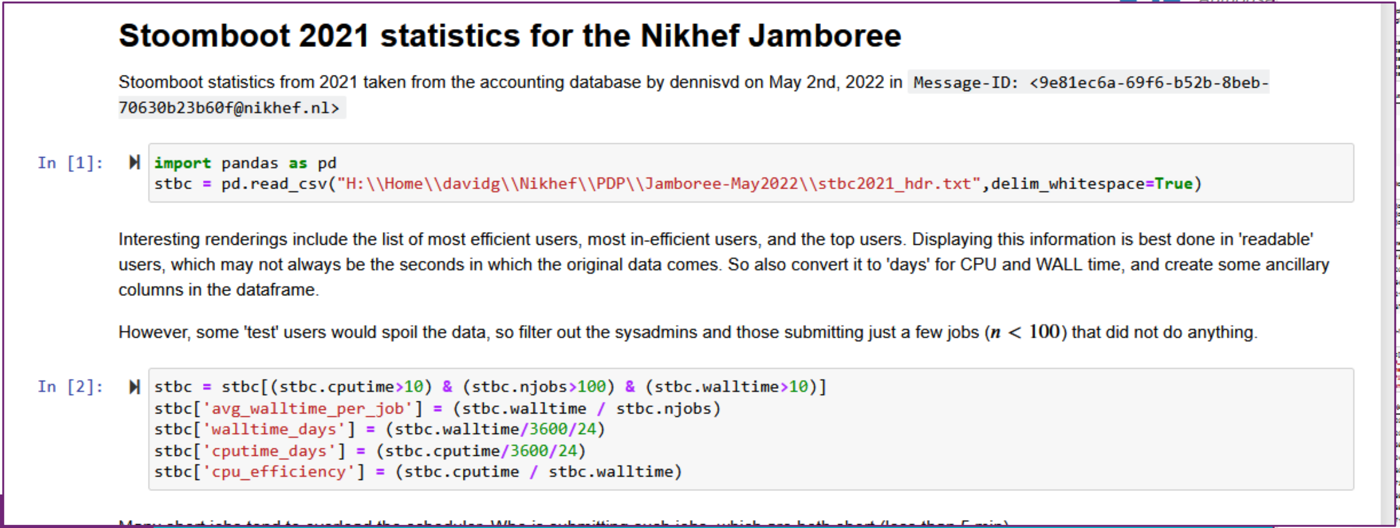
JupyterHub 'Callysto'

Jupyter notebook service for Nikhef local users. Includes both python as well as shell kernels by default. Root can be used ...

Authentic

Identity ma

Authorisa



Stoomboot 2021 statistics for the Nikhef Jamboree

Stoomboot statistics from 2021 taken from the accounting database by dennisvd on May 2nd, 2022 in `Message-ID: <9e81ec6a-69f6-b52b-8beb-70630b23b60f@nikhef.nl>`

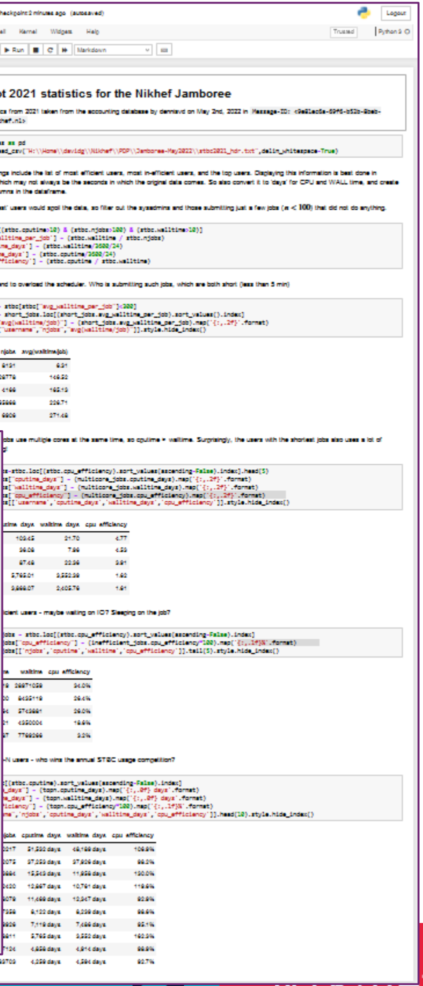
In [1]:
```
import pandas as pd
stbc = pd.read_csv("H:\\Home\\davidg\\Nikhef\\PDP\\Jamboree-May2022\\stbc2021_hdr.txt",delim_whitespace=True)
```

Interesting renderings include the list of most efficient users, most in-efficient users, and the top users. Displaying this information is best done in 'readable' users, which may not always be the seconds in which the original data comes. So also convert it to 'days' for CPU and WALL time, and create some ancillary columns in the dataframe.

However, some 'test' users would spoil the data, so filter out the sysadmins and those submitting just a few jobs ($n < 100$) that did not do anything.

In [2]:
```
stbc = stbc[(stbc.cputime>10) & (stbc.njobs>100) & (stbc.walltime>10)]
stbc['avg_walltime_per_job'] = (stbc.walltime / stbc.njobs)
stbc['walltime_days'] = (stbc.walltime/3600/24)
stbc['cputime_days'] = (stbc.cputime/3600/24)
stbc['cpu_efficiency'] = (stbc.cputime / stbc.walltime)
```

Many short jobs tend to overload the scheduler. Who is submitting such jobs, which are both short (less than 5 min)



Stoomboot 2021 statistics for the Nikhef Jamboree

Stoomboot statistics from 2021 taken from the accounting database by dennisvd on May 2nd, 2022 in Message-ID: <9e81ec6a-69f6-b52b-8beb-70630b23b60f@nikhef.nl>

In [1]: import pandas as pd
stbc = pd.read_csv("H:\\Home\\davidg\\Nikhef\\PDP\\Jamboree-May2022\\stbc2021_hdr.txt",delim_whitespace=True)

Interesting renderings include the list of most efficient users, most in-efficient users, and the top users. Displaying this information is best done in 'readable' users, which may not always be the seconds in which the original data comes. So also convert it to 'days' for CPU and WALL time, and create some ancillary columns in the dataframe.

However, some 'test' users would spoil the data, so filter out the sysadmins and those submitting just a few jobs (n < 100) that did not do anything.

In [2]: stbc = stbc[(stbc.cputime>10) & (stbc.njobs>100) & (stbc.walltime>10)]
stbc['avg_walltime_per_job'] = (stbc.walltime / stbc.njobs)
stbc['walltime_days'] = (stbc.walltime/3600/24)
stbc['cputime_days'] = (stbc.cputime/3600/24)
stbc['cpu_efficiency'] = (stbc.cputime / stbc.walltime)

Many short jobs tend to overload the scheduler. Who is submitting such jobs, which are both short (less than 5 min)

In [3]:
```
short_jobs = stbc[stbc["avg_walltime_per_job"]<300]
short_jobs = short_jobs.loc[(short_jobs.avg_walltime_per_job).sort_values().index]
short_jobs["avg(walltime/job)"] = (short_jobs.avg_walltime_per_job).map('{:,.2f}'.format)
short_jobs[["username","njobs","avg(walltime/job)"]].style.hide_index()
```

Out[3]:

username	njobs	avg(walltime/job)
raquel	6131	6.31
mmlaslo	26776	149.52
gmagni	4199	165.13
ofearrab	165666	226.71
jmanczak	6909	271.48

obs use multiple cores at the same time, so cputime > walltime. Surprisingly, the users with the shortest jobs also uses a lot of
g!

```
s=stbc.loc[(stbc.cpu_efficiency).sort_values(ascending=False).index].head(5)
s["cputime_days"] = (multicore_jobs.cputime_days).map('{:,.2f}'.format)
s["walltime_days"] = (multicore_jobs.walltime_days).map('{:,.2f}'.format)
s["cpu_efficiency"] = (multicore_jobs.cpu_efficiency).map('{:,.2f}'.format)
s[['username','cputime_days','walltime_days','cpu_efficiency']].style.hide_index()
```

time days	walltime days	cpu efficiency
103.45	21.70	4.77
36.09	7.99	4.53
87.48	22.36	3.91
5,765.01	3,552.56	1.62
3,888.07	2,405.79	1.61

cient users - maybe waiting on IO? Sleeping on the job?

```
jobs = stbc.loc[(stbc.cpu_efficiency).sort_values(ascending=False).index]
jobs["cpu_efficiency"] = (inefficient_jobs.cpu_efficiency*100).map('{:,.1f}%'.format)
jobs[['njobs','cputime','walltime','cpu_efficiency']].tail(5).style.hide_index()
```

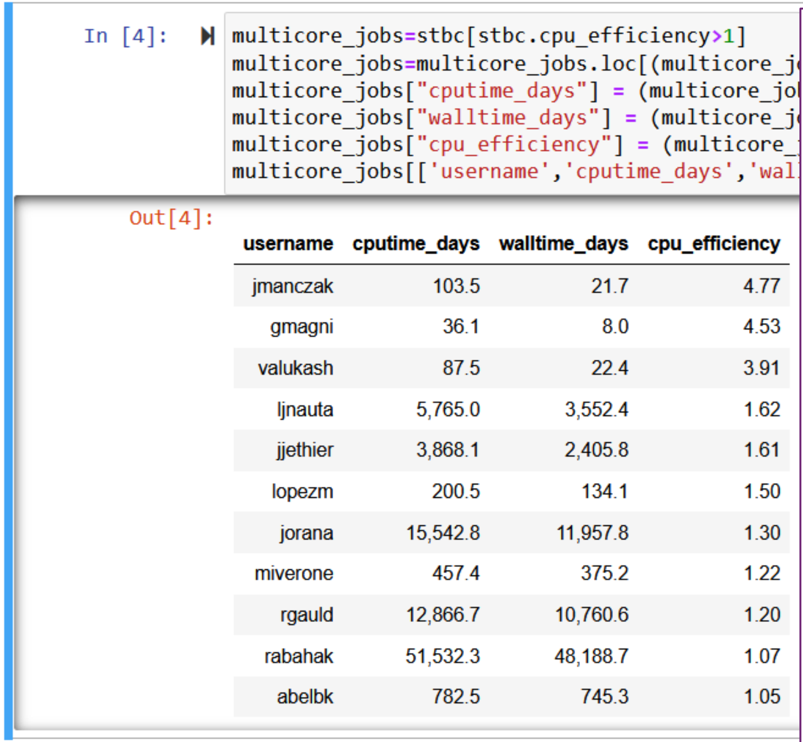
ne	walltime	cpu efficiency
19	28871038	34.0%
00	8435119	26.4%
94	5743881	26.0%
21	4350004	16.6%
67	7768266	3.2%

-N users - who wins the annual STBC usage competition?

```
c[(stbc.cputime).sort_values(ascending=False).index]
_days"] = (topn.cputime_days).map('{:,.0f} days'.format)
e_days"] = (topn.walltime_days).map('{:,.0f} days'.format)
ficiency"] = (topn.cpu_efficiency*100).map('{:,.1f}%'.format)
me",'njobs','cputime_days','walltime_days','cpu_efficiency']].head(10).style.hide_index()
```

	njobs	cputime days	walltime days	cpu efficiency
	0217	51,530 days	48,198 days	106.9%
	0075	37,253 days	37,926 days	98.2%
	9984	15,543 days	11,959 days	130.0%
	0420	12,667 days	10,791 days	119.6%
	6079	11,469 days	12,347 days	92.9%
	7358	8,133 days	8,239 days	98.6%
	6926	7,119 days	7,486 days	95.1%
	6811	5,765 days	3,552 days	162.3%
	7134	4,658 days	4,814 days	96.8%
kasgervd	63703	4,258 days	4,584 days	92.7%

The most efficient users use more than one core!

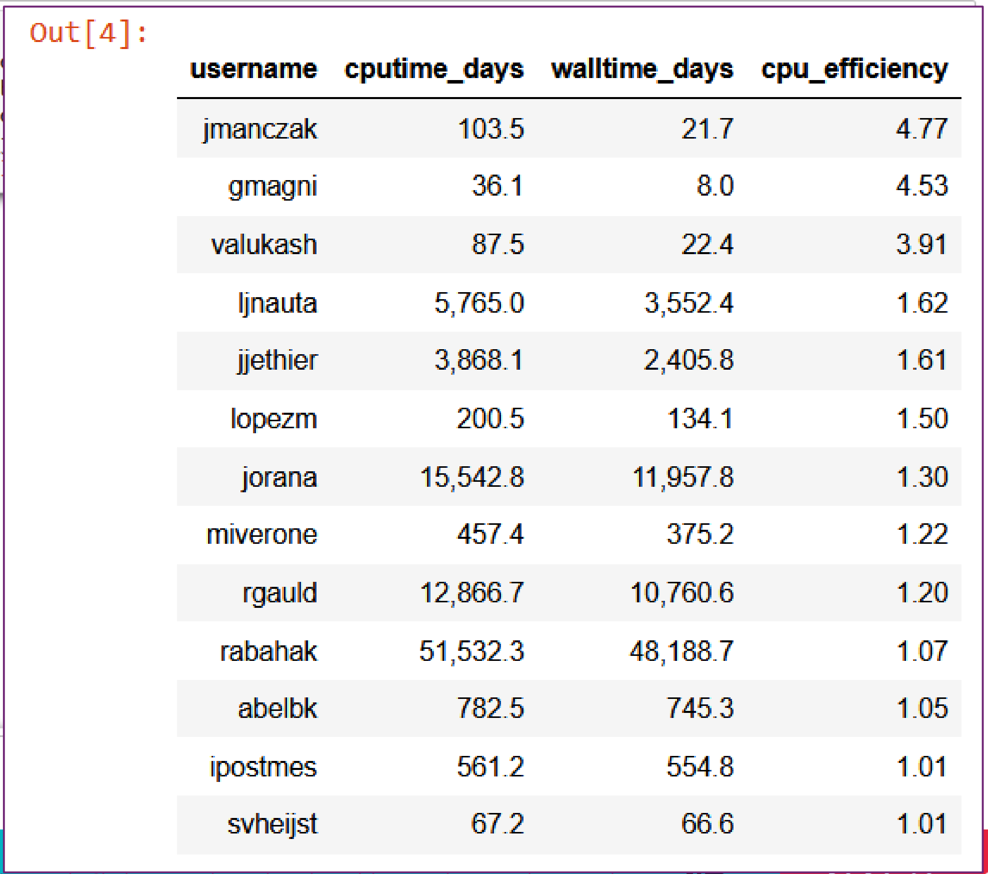


In [4]:

```
multicore_jobs=stbc[stbc.cpu_efficiency>1]
multicore_jobs=multicore_jobs.loc[(multicore_j
multicore_jobs["cputime_days"] = (multicore_jo
multicore_jobs["walltime_days"] = (multicore_j
multicore_jobs["cpu_efficiency"] = (multicore_
multicore_jobs[['username','cputime_days','wal
```

Out[4]:

username	cputime_days	walltime_days	cpu_efficiency
jmanczak	103.5	21.7	4.77
gmagni	36.1	8.0	4.53
valukash	87.5	22.4	3.91
ljnauta	5,765.0	3,552.4	1.62
jjethier	3,868.1	2,405.8	1.61
lopezm	200.5	134.1	1.50
jorana	15,542.8	11,957.8	1.30
miverone	457.4	375.2	1.22
rgauld	12,866.7	10,760.6	1.20
rabahak	51,532.3	48,188.7	1.07
abelbk	782.5	745.3	1.05



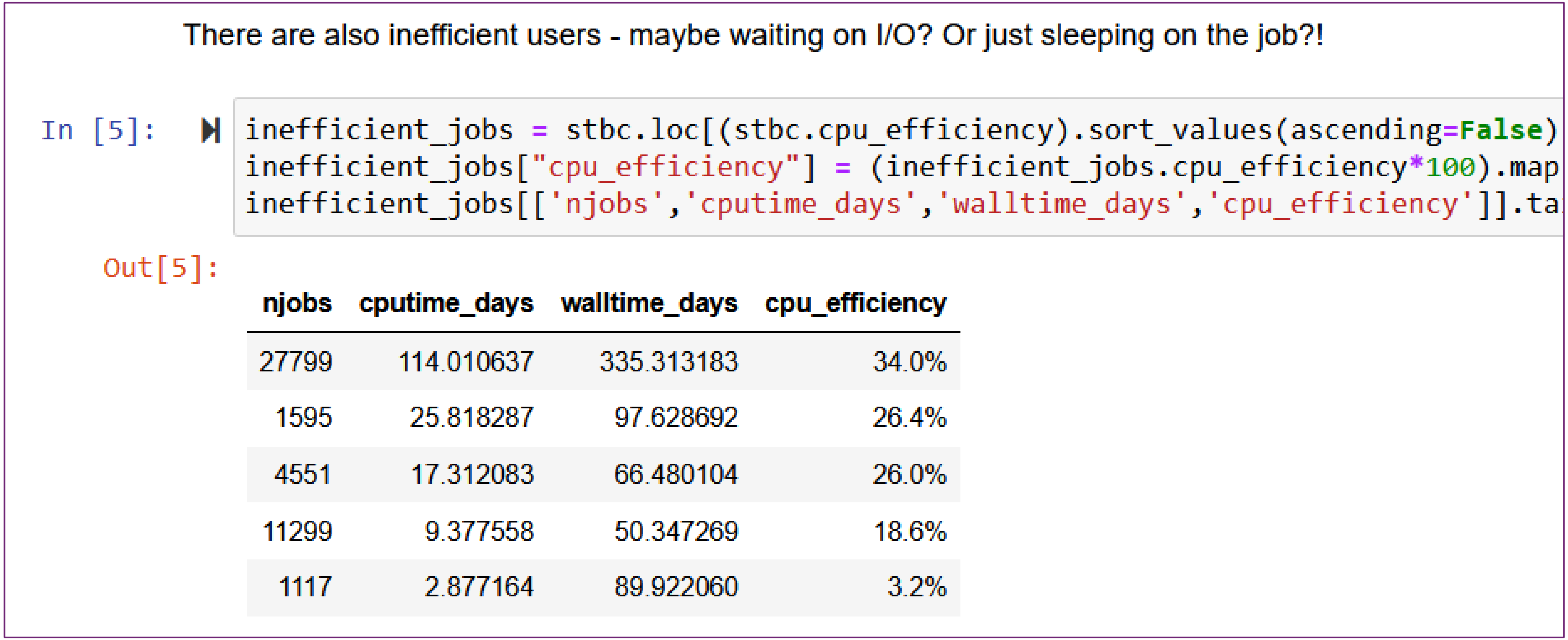
Out[4]:

username	cputime_days	walltime_days	cpu_efficiency
jmanczak	103.5	21.7	4.77
gmagni	36.1	8.0	4.53
valukash	87.5	22.4	3.91
ljnauta	5,765.0	3,552.4	1.62
jjethier	3,868.1	2,405.8	1.61
lopezm	200.5	134.1	1.50
jorana	15,542.8	11,957.8	1.30
miverone	457.4	375.2	1.22
rgauld	12,866.7	10,760.6	1.20
rabahak	51,532.3	48,188.7	1.07
abelbk	782.5	745.3	1.05
ipostmes	561.2	554.8	1.01
svheijst	67.2	66.6	1.01

but there are just 15 of them …

The STBC analysis facility is a collaborative environment where users learn from their peers by example. Sharing community good practice and demonstrating appropriate use to the global physics community is beneficial for Nikhef as an organization and for its collaborative mission. Sharing sufficient user information such that experts can be found by their peers who are already in the community, by username, we therefore consider as legitimate interest of Nikhef as an organization. If you object to processing your pseudonymous personal data in this way, please contact grid.sysadmin@nikhef.nl to get it removed from this presentation. Further information can be found in the Nikhef privacy notice at https://www.nikhef.nl/privacy/

But some other statistics shall remain anonymous



There are also inefficient users - maybe waiting on I/O? Or just sleeping on the job?!

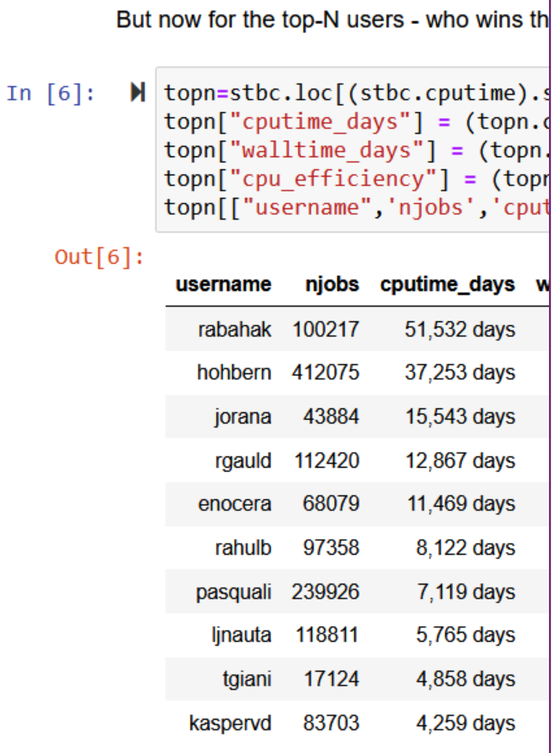
In [5]:

```
inefficient_jobs = stbc.loc[(stbc.cpu_efficiency).sort_values(ascending=False)
inefficient_jobs["cpu_efficiency"] = (inefficient_jobs.cpu_efficiency*100).map
inefficient_jobs[['njobs','cputime_days','walltime_days','cpu_efficiency']].ta
```

Out[5]:

njobs	cputime_days	walltime_days	cpu_efficiency
27799	114.010637	335.313183	34.0%
1595	25.818287	97.628692	26.4%
4551	17.312083	66.480104	26.0%
11299	9.377558	50.347269	18.6%
1117	2.877164	89.922060	3.2%

What you’ve all been waiting for … our top users in 2021!



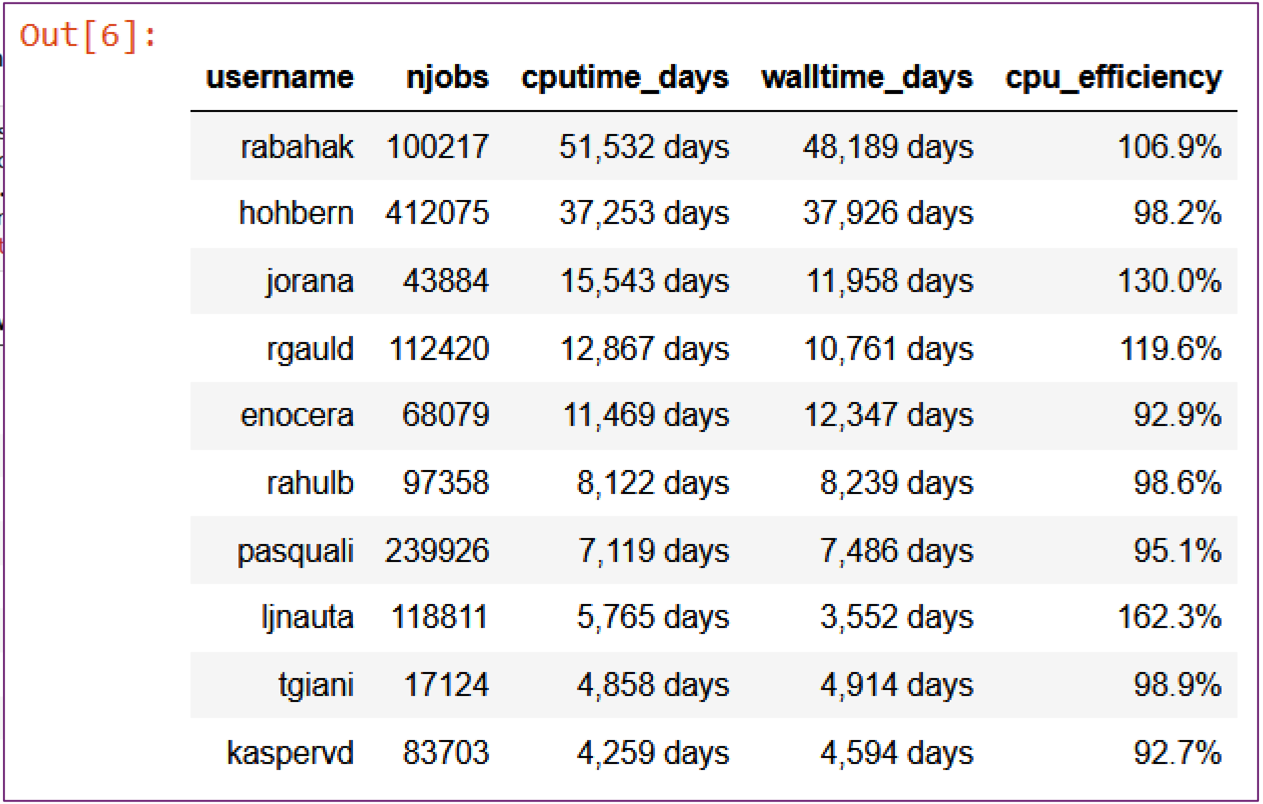
But now for the top-N users - who wins th

In [6]:

```
topn=stbc.loc[(stbc.cputime).s
topn["cputime_days"] = (topn.c
topn["walltime_days"] = (topn.
topn["cpu_efficiency"] = (topn
topn[["username",'njobs','cput
```

Out[6]:

username	njobs	cputime_days
rabahak	100217	51,532 days
hohbern	412075	37,253 days
jorana	43884	15,543 days
rgauld	112420	12,867 days
enocera	68079	11,469 days
rahulb	97358	8,122 days
pasquali	239926	7,119 days
ljnauta	118811	5,765 days
tgiani	17124	4,858 days
kaspervd	83703	4,259 days



Out[6]:

username	njobs	cputime_days	walltime_days	cpu_efficiency
rabahak	100217	51,532 days	48,189 days	106.9%
hohbern	412075	37,253 days	37,926 days	98.2%
jorana	43884	15,543 days	11,958 days	130.0%
rgauld	112420	12,867 days	10,761 days	119.6%
enocera	68079	11,469 days	12,347 days	92.9%
rahulb	97358	8,122 days	8,239 days	98.6%
pasquali	239926	7,119 days	7,486 days	95.1%
ljnauta	118811	5,765 days	3,552 days	162.3%
tgiani	17124	4,858 days	4,914 days	98.9%
kaspervd	83703	4,259 days	4,594 days	92.7%

The STBC analysis facility is a collaborative environment where users learn from their peers by example. Sharing community good practice and demonstrating appropriate use to the global physics community is beneficial for Nikhef as an organization and for its collaborative mission. Sharing sufficient user information such that experts can be found by their peers who are already in the community, by username, we therefore consider as legitimate interest of Nikhef as an organization. If you object to processing your pseudonymous personal data in this way, please contact grid.sysadmin@nikhef.nl to get it removed from this presentation. Further information can be found in the Nikhef privacy notice at https://www.nikhef.nl/privacy/

But there many roads that lead to … where exactly??!

‘My_Pythia_generation.log’

```
...
WARNING: four vector not initialized; Seed for event 1 : 345321456345
WARNING: four vector not initialized; Seed for event 2 : 143919132860
...
```

and you’re happy you got a few events OK!

So now, immediately:

1. login to stbc-i1
2. *qsub* your array-jobs of 1 million events each
3. write the log results to /project/*
4. go on holiday!

and feel rather surprised once you find out … that you killed the system for all your colleagues ☹

Photo by Maarten van den Heuvel on Unsplash

OK, ssh agents help keep your credentials secure, but …

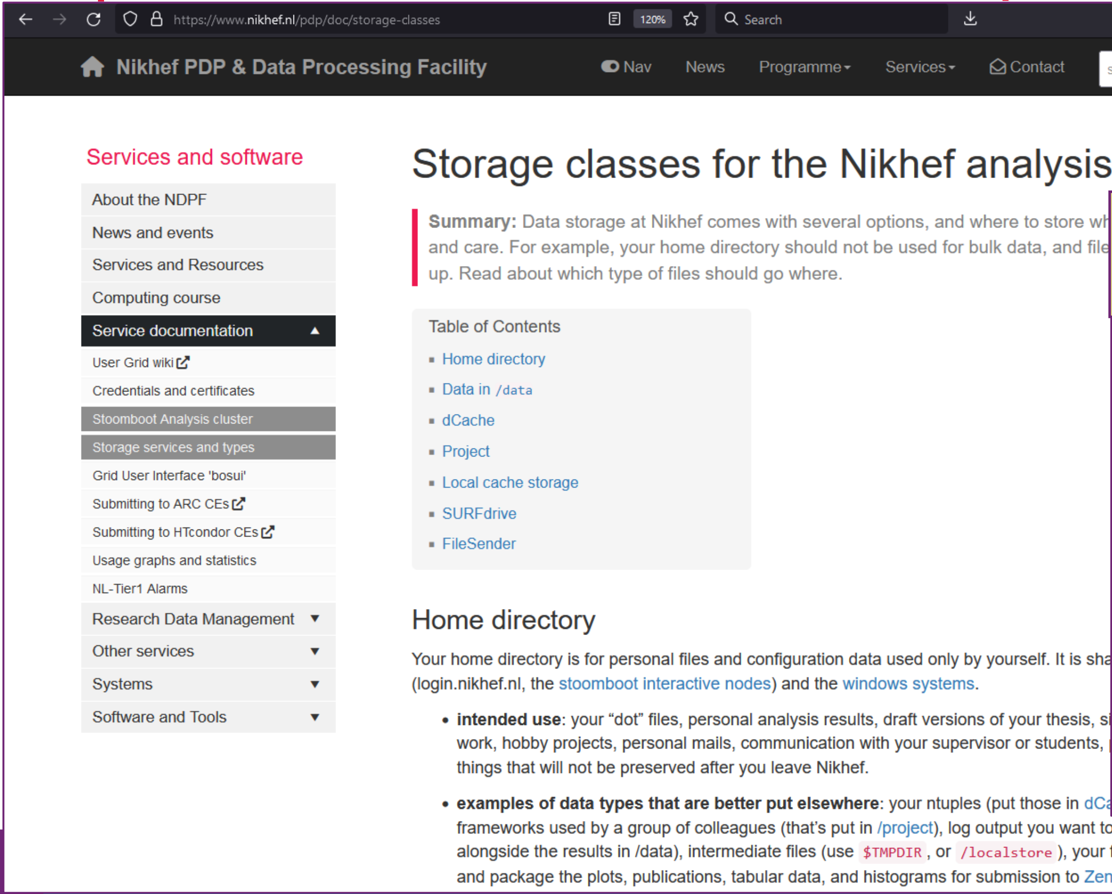
> 200 ssh-agents on the same system is not 200 times more secure …

```
[root@stbc-i1 ~]# ps -o pid,user,comm -C ssh-agent|head -20
   PID USER      COMMAND
   661 h******   ssh-agent
   808 h******   ssh-agent
  1123 h******   ssh-agent
  1169 h******   ssh-agent
  1460 h******   ssh-agent
  3007 h******   ssh-agent
  3290 h******   ssh-agent
  3308 h******   ssh-agent
  3417 h******   ssh-agent
  ...
```

So ~25% of the user processes on stbc-i1 are *one single user's* ssh-agents:

```
[root@stbc-i1 ~]# ps -o pid,user,comm -C ssh-agent|grep -c hXXXXXX
202
[root@stbc-i1 ~]# ps auwwx|grep -vc root
852
```

Get up there - read the docs! … https://www.nikhef.nl/pdp/


https://www.nikhef.nl/pdp/doc/storage-classes
120%
Search
Nikhef PDP & Data Processing Facility
Nav
News
Programme
Services
Contact
search...

Services and software

- About the NDPF
- News and events
- Services and Resources
- Computing course
- Service documentation ▲
  - User Grid wiki
  - Credentials and certificates
  - Stoomboot Analysis cluster
  - Storage services and types
  - Grid User Interface 'bosui'
  - Submitting to ARC CEs
  - Submitting to HTcondor CEs
  - Usage graphs and statistics
  - NL-Tier1 Alarms
- Research Data Management ▼
- Other services ▼
- Systems ▼
- Software and Tools ▼

Storage classes for the Nikhef analysis facilities

Summary: Data storage at Nikhef comes with several options, and where to store wh and care. For example, your home directory should not be used for bulk data, and file up. Read about which type of files should go where.

Table of Contents

Home directory

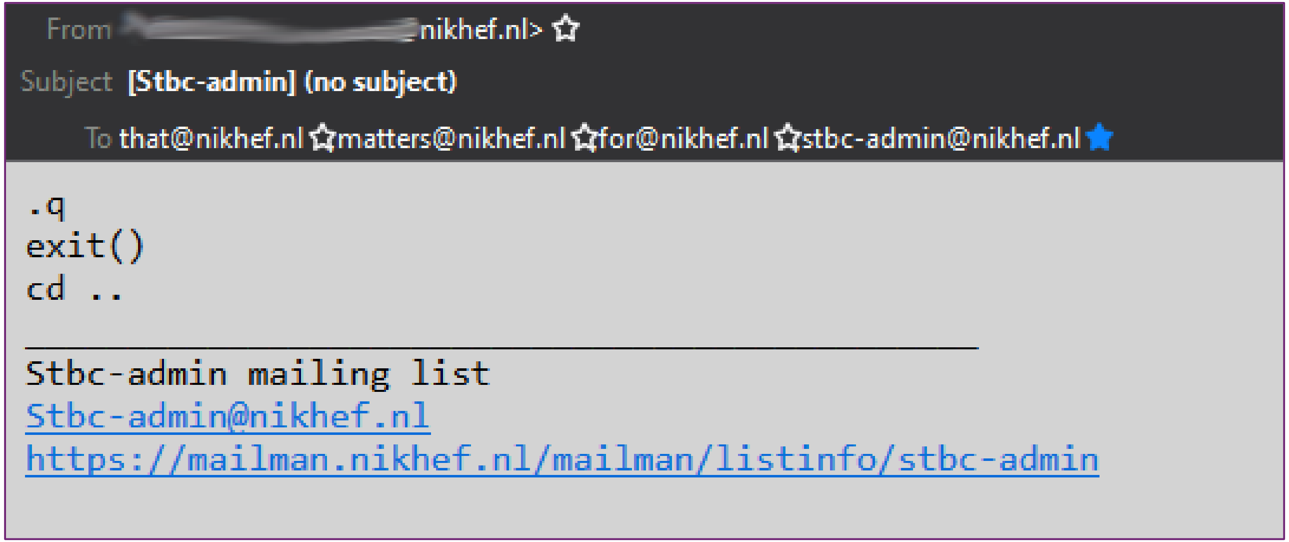
Your home directory is for personal files and configuration data used only by yourself. It is sha (login.nikhef.nl, the stoomboot interactive nodes) and the windows systems.

- **intended use**: your "dot" files, personal analysis results, draft versions of your thesis, si work, hobby projects, personal mails, communication with your supervisor or students, things that will not be preserved after you leave Nikhef.
- **examples of data types that are better put elsewhere**: your ntuples (put those in dCa frameworks used by a group of colleagues (that's put in /project), log output you want to look at later (best put this alongside the results in /data), intermediate files (use `$TMPDIR`, or `/localstore`), your final thesis (submit it to the library, and package the plots, publications, tabular data, and histograms for submission to Zenodo)

or take the computing course (coming soon again!)

And do ask!
Mary will tell you about all subsidiary benefits shortly :)

but it’s quite hard to answer this one …



From [redacted]nikhef.nl>

Subject [Stbc-admin] (no subject)

To that@nikhef.nl matters@nikhef.nl for@nikhef.nl stbc-admin@nikhef.nl

```
.q
exit()
cd ..

_______________________________________________
Stbc-admin mailing list
Stbc-admin@nikhef.nl
https://mailman.nikhef.nl/mailman/listinfo/stbc-admin
```

Next up:

Roel Aaij *accelerating you*

Mary Hester *on tokens, cake, and office hours*

David Groep
davidg@nikhef.nl
https://www.nikhef.nl/~davidg/presentations/
https://orcid.org/0000-0003-1026-6606

Maastricht University

Nikhef

CC BY