
Computing cluster
sunrise
Zdenek Hubacek
Bily Potok, WEJČF 2022

Computing resources

- What to do when you realize your laptop is not enough
- Resources available at KF – cluster sunrise
- Basics of batch tools – PBSPro

- What to do next

Typical HEP use cases

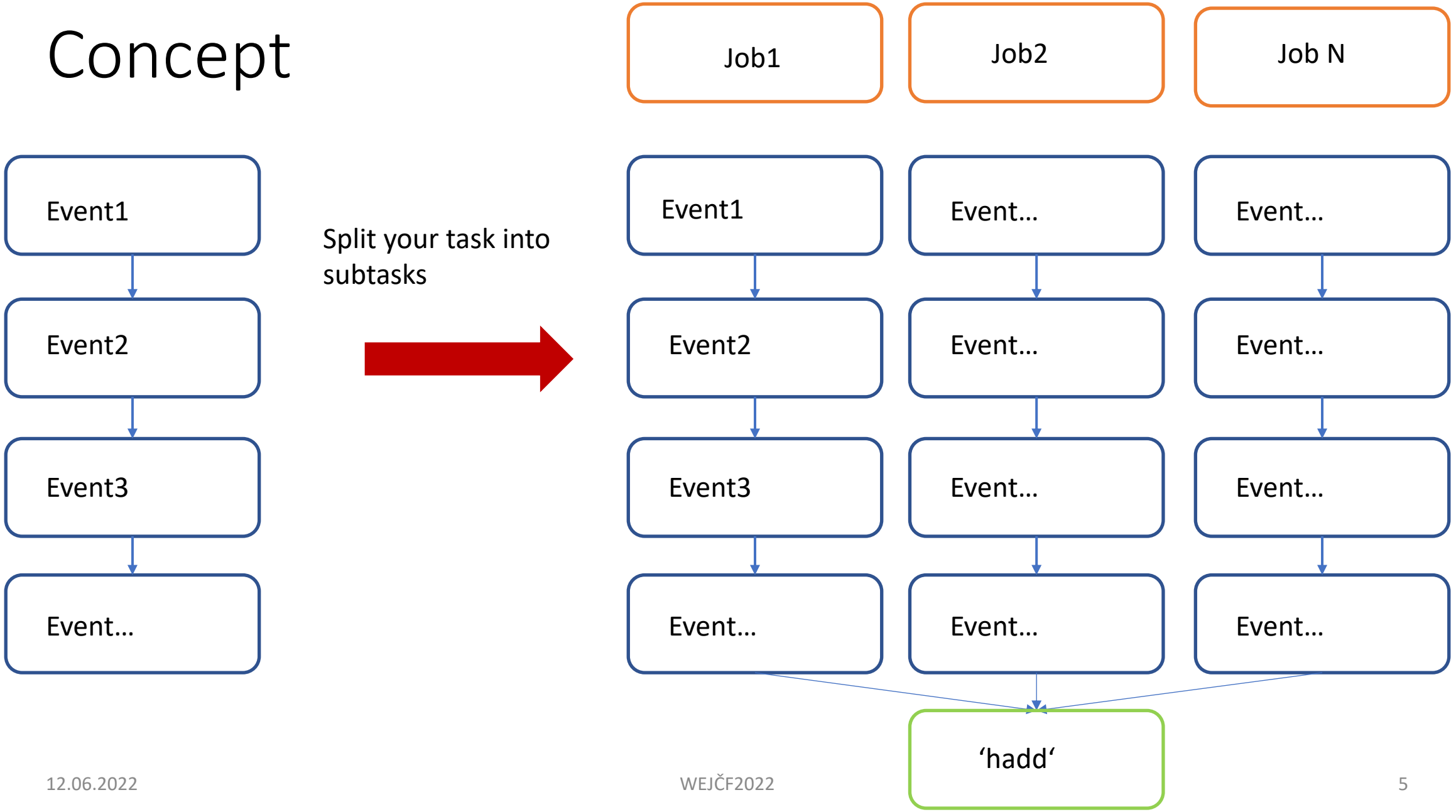
- Generate huge number of MC events
- Reconstruct event(s) from my detector
- Analyze large number of events
- Train my multivariate model

- Plot, present, publish, book travel to Stockholm…

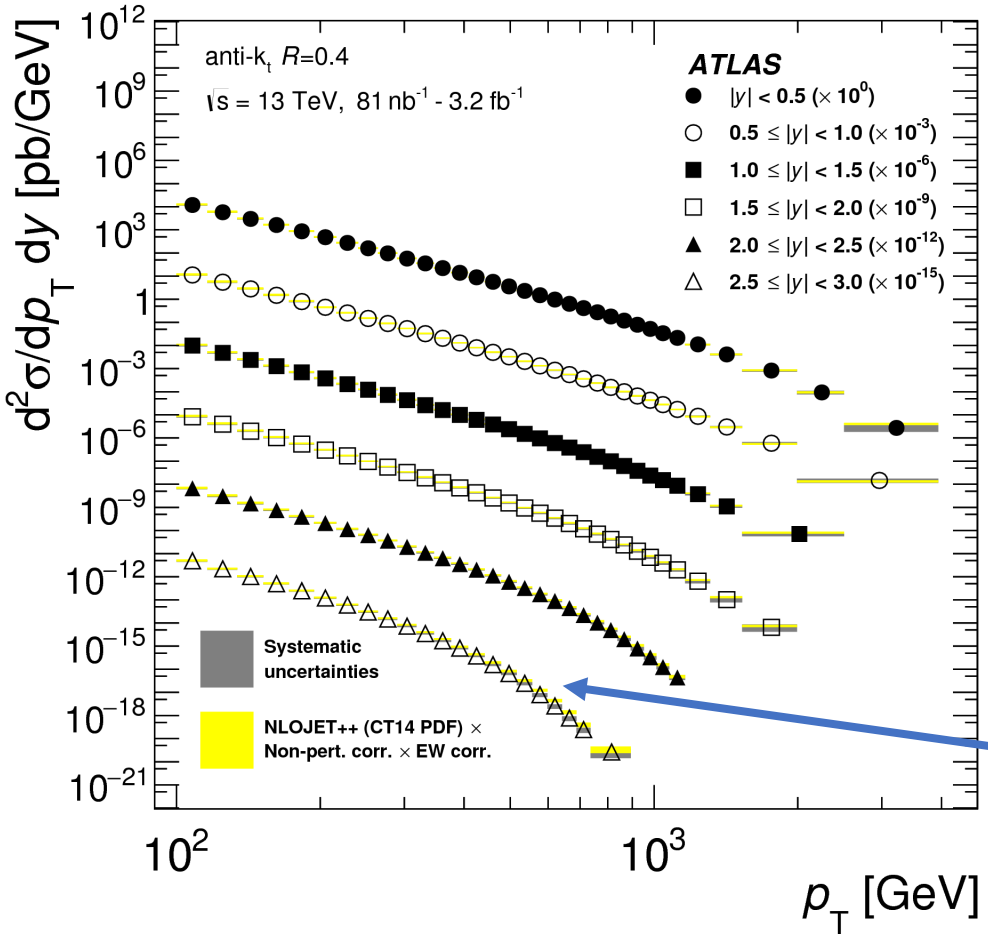
Task: Generate 100B = 10^{11} events

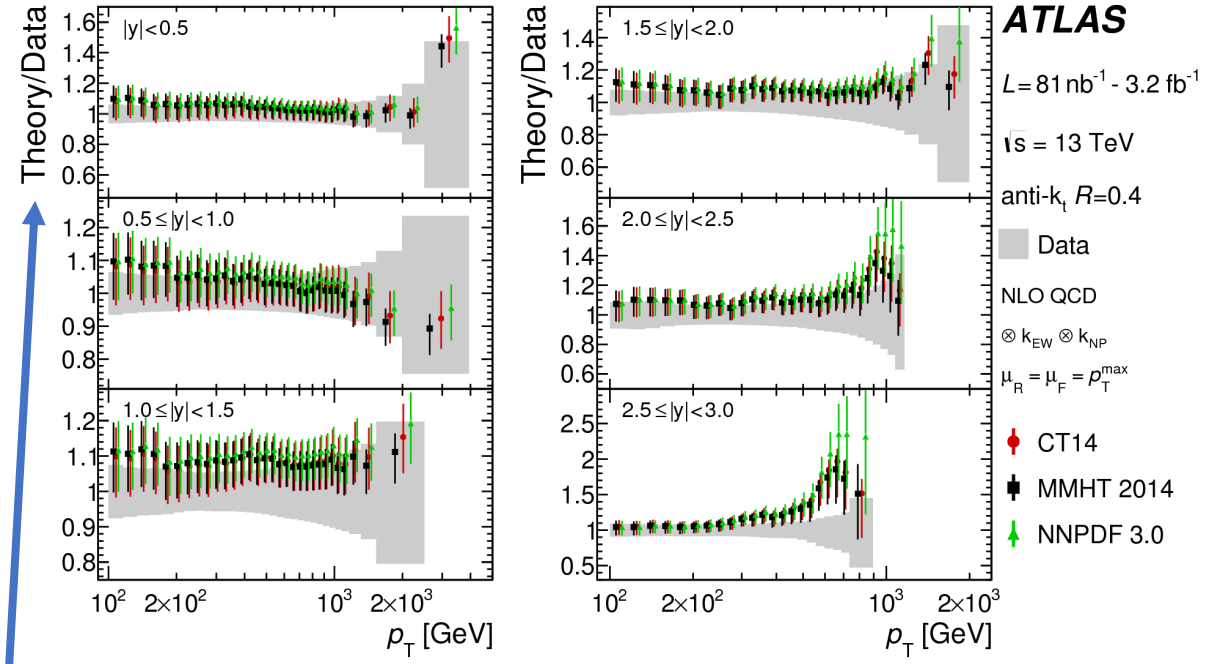
- Say, typical time is 1000=10^3events/s
- 10^8s => year has $\pi \cdot 10^7$s = 3+ years
- Will your laptop survive running 3 years (uninterrupted)?
- Do you need it for other tasks?
- When do you want to finish your study? What does your advisor say?
- If it is 1 computer for 40months, would 40 computers for 1 month work?

Concept


Job1
Job2
Job N
Event1
Event2
Event3
Event…
Split your task into subtasks
Event1
Event2
Event3
Event…
Event…
Event…
Event…
Event…
Event…
Event…
Event…
Event…
‘hadd‘

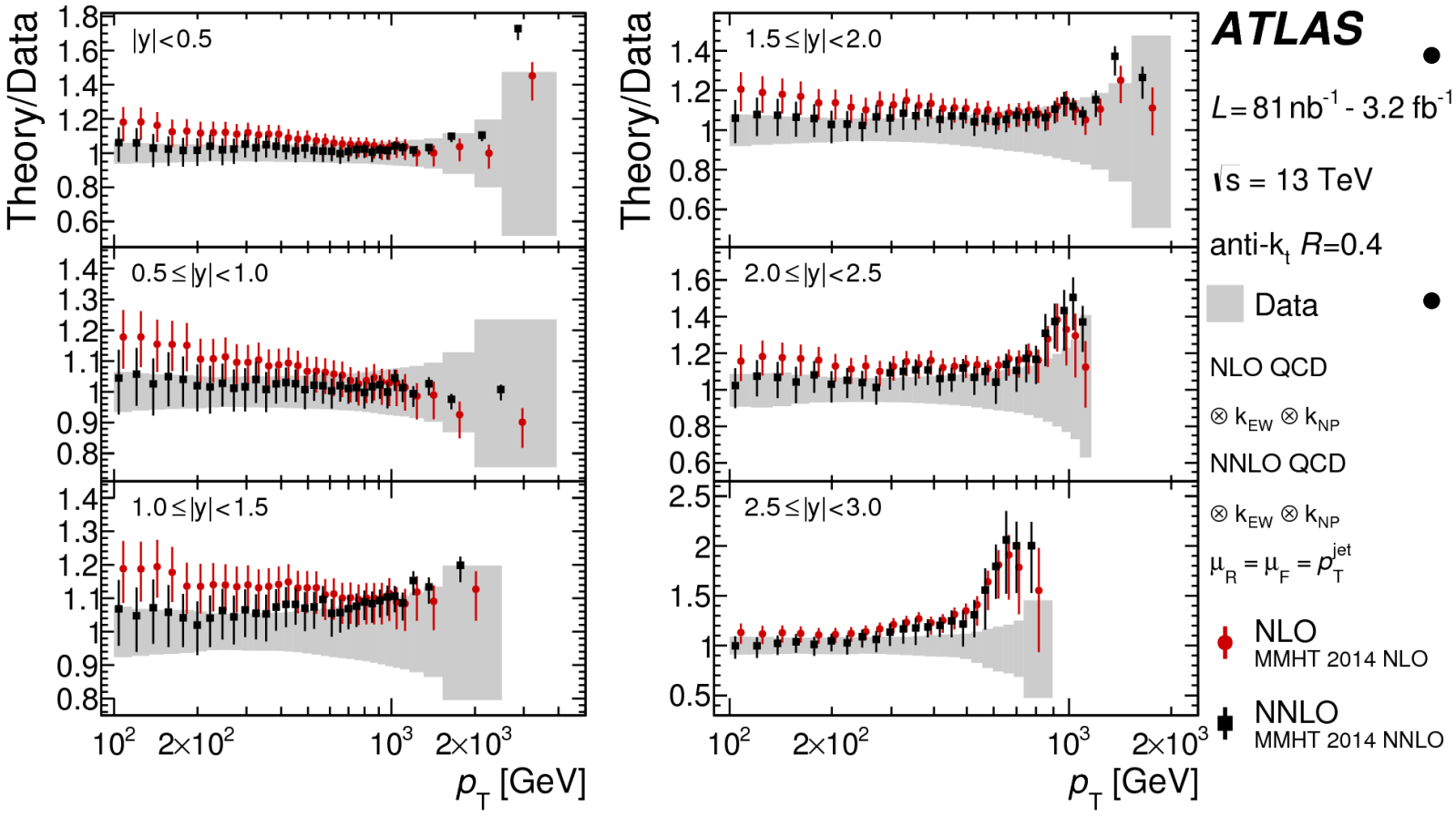
ATLAS Inclusive jet cross section measurement *JHEP 05 (2018) 195*





NLO pQCD prediction needs O(10B) generated events requires O(1000 CPUs) for 1-2 weeks

State of the art - NNLO pQCD

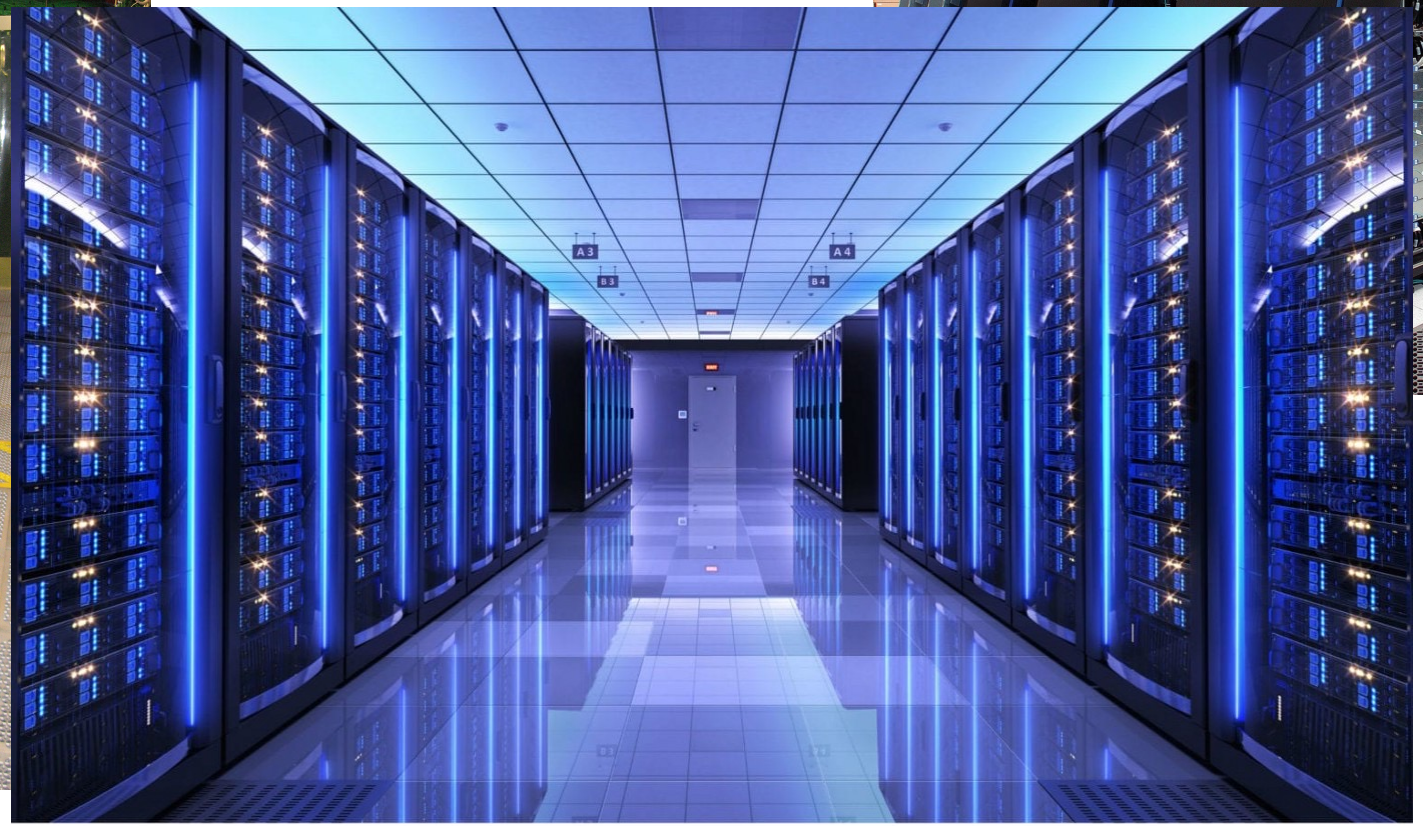


- NNLO calculation took – O(1000CPUs) for 1-2 months
- That was **WITHOUT** PDF uncertainties – normally O(50) variations for each PDF set

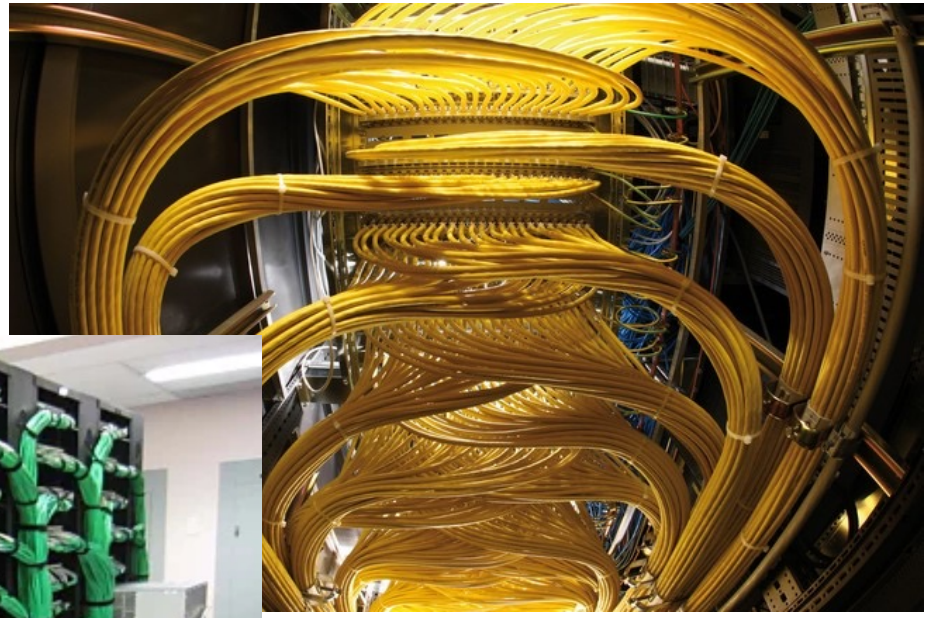
OK, where do I get 1000 CPUs

There ought to be a better way…

HPC, Computing farm, Supercomputer…



Servers and cables



HPC in general

- Not easy to describe – multiple architectures depending on the task solved
- CPUs, GPUs, architecture, memory, disk space…

- For simplicity – imagine one or more racks with one (more) computing units (servers) with one or more CPUs (multicore/multithreaded now) with other features

- So can I get 1000 CPUs now?

Not so fast…

- Do we have something like that at the department?
- Can I use it?
- How can I use it?
- What can I and can‘t I do?

Not so fast…

- Do we have something like that at the department?
- Can I use it?
- How can I use it?
- What can I and can‘t I do?

sunrise

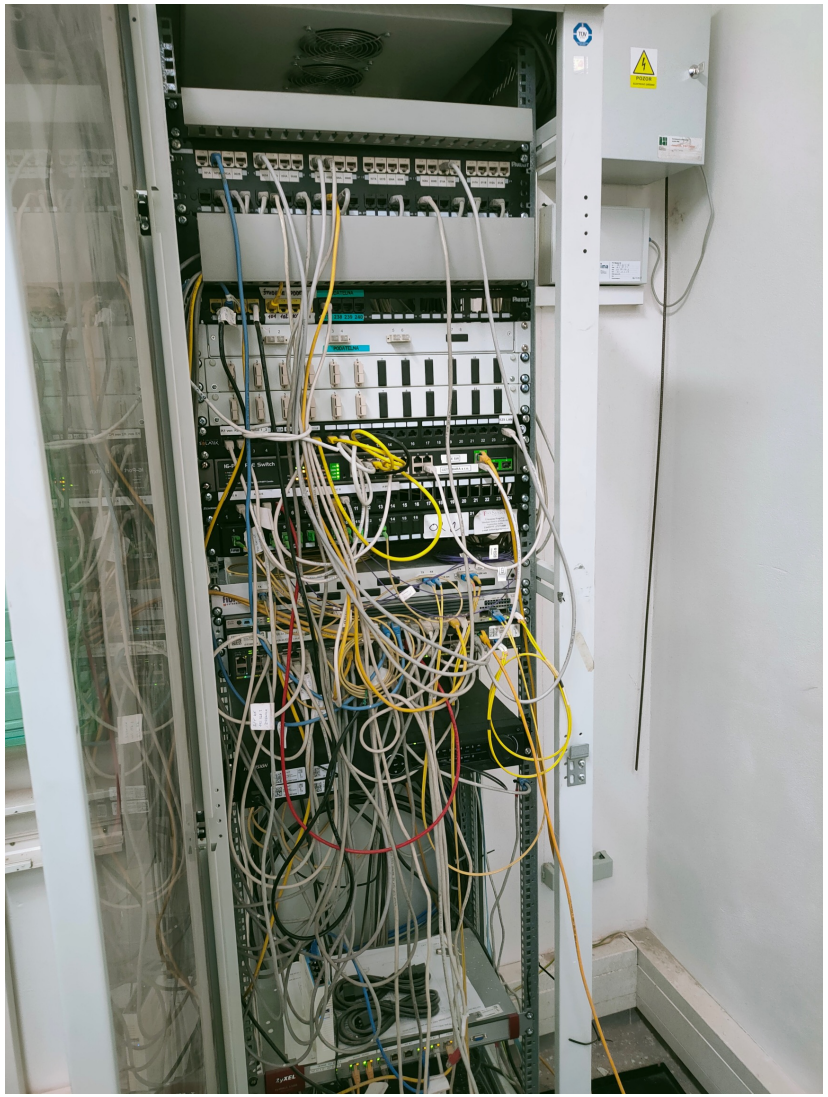
sunrise.fjfi.cvut.cz (2022)

- Small computing cluster of KF
- Located in Brehova
- ~650 CPUs at the moment, multiple configurations
- Disk storage – 200TB

NEW Sunrise 2022

- Sunset 29-34
- 1x AMD EPYC3 7543@2.8GHz (64 HT cores, 256GB RAM each) = 384 new cores
- New (and some older nodes) connected with 10Gbit/s to the disk storage
- New SSD disk storage available soon (use case?)

Bottleneck…

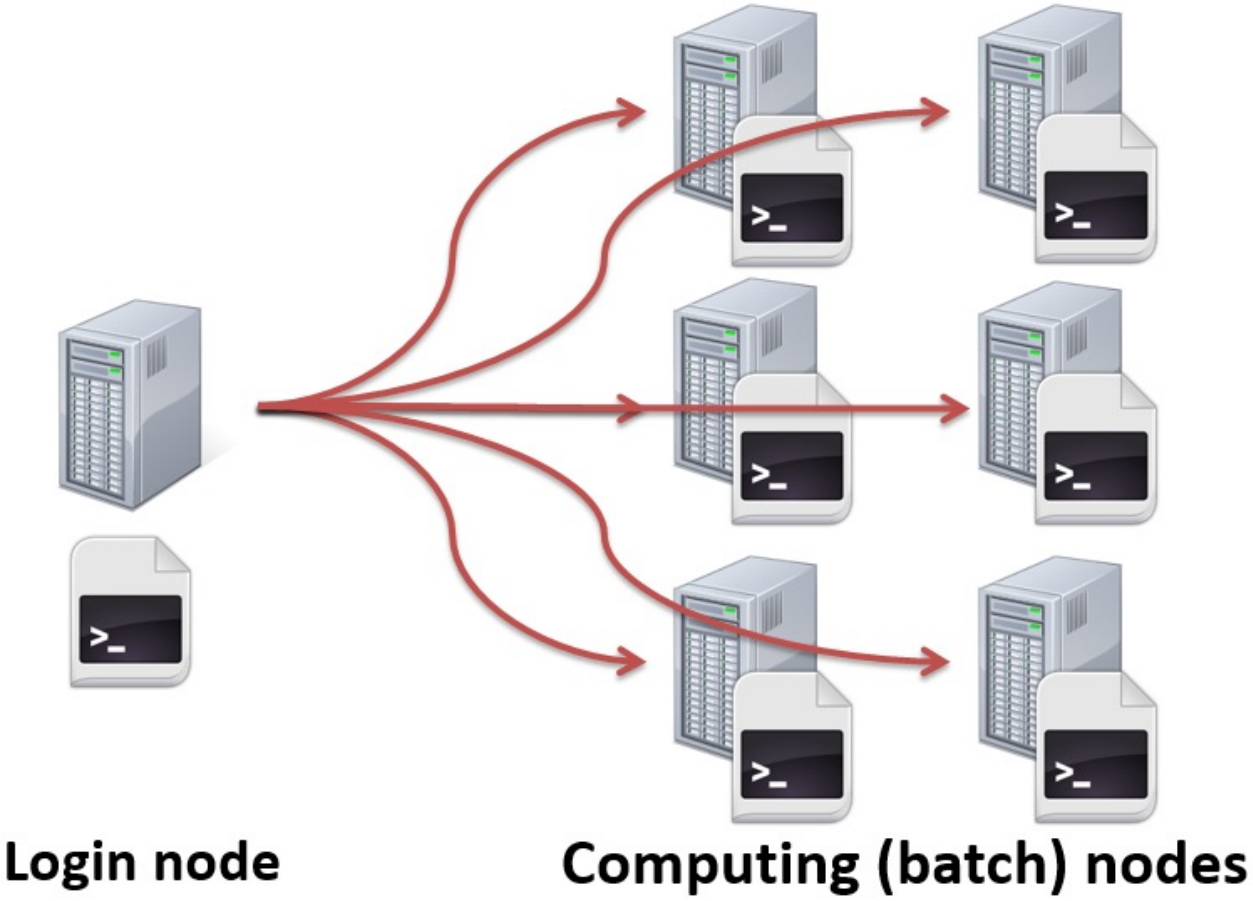


- This is not KF ☺
- Limits our connection outside of Brehova (copying data to/from CERN)…
- Working on solution

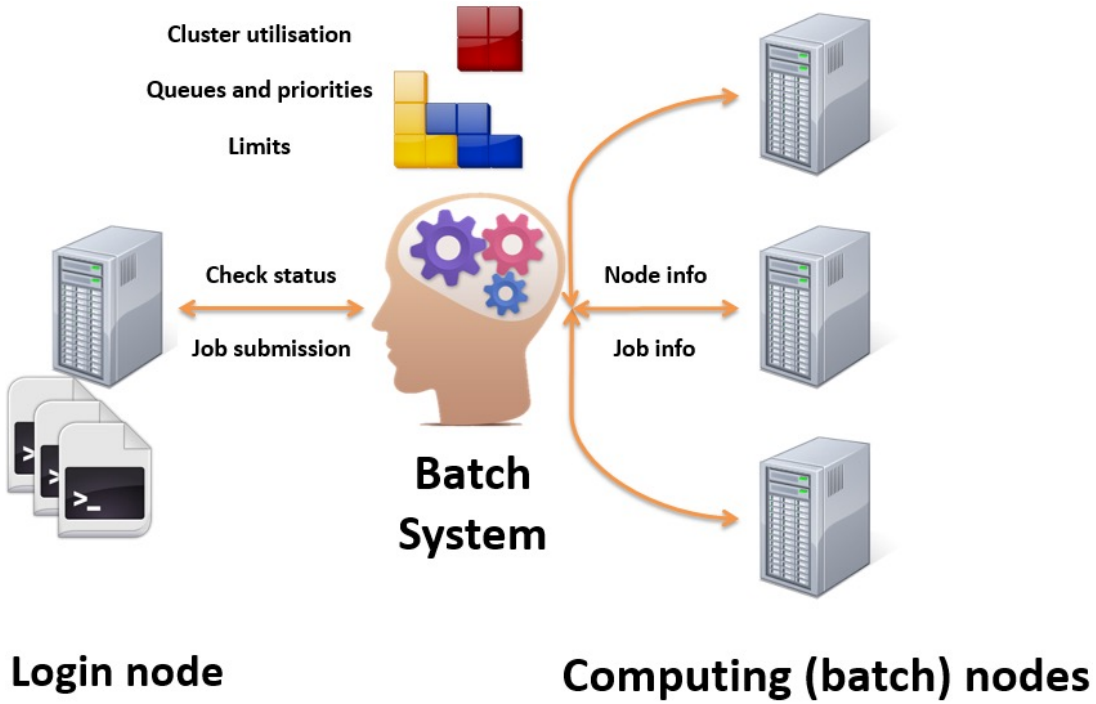
Sunrise info

- Web: https://nms.fjfi.cvut.cz/wiki/Sunrise.fjfi.cvut.cz
- Admins: Michal Broz – accounts…
- (Jan Cepila, Petr Vokac)
- (don‘t be afraid to ask for help)

Now - how to use it?



Batch system, job scheduler, …



- Job scheduler/batch system/distributed resource management system (DRMS) controls launching, removing individual tasks
- Basic features expected of job scheduler software include:
  - interfaces which help to define workflows and/or job dependencies
  - automatic submission of executions
  - interfaces to monitor the executions
  - priorities and/or queues to control the execution order of unrelated jobs

Batch system

- Torque, LSF, PBS, Condor, … Grid
  - Same principles, but different commands, switches etc…
- sunrise uses PBSPro
  - http://www.pbsworks.com/pdfs/PBSUserGuide14.2.pdf
  - Basic commands: **qsub, qstat, qdel**
  - Job – one task to be computed
  - Node – one CPU

Main glossary

- **Main node** – sunrise.fjfi.cvut.cz
  - There could be multiple (lxplus v CERNu), serves for submission of your jobs
- **Worker nodes** (sunsetXX.fjfi.cvut.cz)
  - In general, you won‘t have access rights to log to worker nodes (**possible on sunrise**)
  - In general, $HOME and other disks might not be available on worker nodes (**available on sunrise**) – how to submit your configuration, data, etc to worker nodes
  - **Don‘t log to individual worker nodes** to run jobs there directly!

Instead – prepare your job and submit it to batch queue

- Each batch system can have multiple queus and you need to choose where to submit your jobs
  - Easiest – time requirement – short, normal, long
  - In general – according to requirement of your job (CPUs, memory, other) – batch system will (try to) find a computer for you with the right parameters
- How to find out which queus are available?
- PBS: **qstat –q**
- **qstat –Q –f** (or **qstat –Q short –f**)

Sunrise queues

```
[hubaczde@sunrise ~]$ qstat -q

server: sunrise

Queue            Memory CPU Time Walltime Node   Run   Que   Lm  State
---------------- ------ -------- -------- ---- ----- ----- ----  -----
default            --      --    720:00:0  --      0     0   --   E R
short              --      --    02:00:00  --      0     1   --   E R
hiprio             --      --    48:00:00  --      0     0   --   E R
normal             --      --    24:00:00  --      0     0   --   E R
long               --      --    720:00:0  --      0     0   --   E R
backfill           --      --    24:00:00    1     0     0   --   E R
infinite           --      --       --     --      0     0   --   E R
                                                ----- -----
                                                    0     1

[hubaczde@sunrise ~]$
```

Detailed information about queues

```
[hubaczde@sunrise ~]$ qstat -Q short -f
Queue: short
    queue_type = Execution
    Priority = 100
    total_jobs = 1
    state_count = Transit:0 Queued:1 Held:0 Waiting:0 Running:0 Exiting:0 Begun
        :0
    max_queued = [u:PBS_GENERIC=5000]
    resources_max.walltime = 02:00:00
    resources_default.walltime = 01:00:00
    resources_assigned.mem = 0kb
    resources_assigned.mpiprocs = 0
    resources_assigned.ncpus = 0
    resources_assigned.nodect = 0
    max_run_res.ncpus = [u:PBS_GENERIC=800]
    backfill_depth = 10
    enabled = True
    started = True

[hubaczde@sunrise ~]$
```

- sunrise – short (<2h), normal (2-24h), long (24-720h) walltime (NB: some batches distinguish between wall time and CPU time)
- fairshare

Test job

- Job submitting – **qsub (qsub –parameters submitscript.sh)**
  - Simplest: **qsub –q short skript.sh**
  - Where **skript.sh** does something like

```
[hubaczde@sunrise ~]$ cat skript.sh
#! /bin/bash
echo "Ahoj svete"
sleep 60s
[hubaczde@sunrise ~]$ qsub skript.sh
47093.sunrise
[hubaczde@sunrise ~]$
```

Hello world example

- 47093 is the job number

Monitoring

- **qstat**
- **qstat –u hubaczde**
- **qstat –u hubaczde –n** (shows the node name where the job runs)
- **qstat 47093** (information about this job)
- **qstat 47093 –f** (full job information)
- **qstat –q short**

```
[hubaczde@sunrise ~]$ qsub skript.sh
47094.sunrise
[hubaczde@sunrise ~]$ qstat
Job id            Name             User              Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
45077.sunrise     test.sh          vokacpet                 0 Q short
47094.sunrise     skript.sh        hubaczde          00:00:00 R short
[hubaczde@sunrise ~]$
```

- Status (S) – job could be waiting – queue is full, there isn‘t a computer satisfying your requirements, other user has higher priority (fairshare), running, (failing), exiting

Monitoring2

- NEW:
- http://sunrise.fjfi.cvut.cz/grafana/d/batch/batch?orgId=1

- Individual job monitoring – each job writes stdout a stderr (what you would normally have in a terminal) to files saved on worker nodes **/var/spool/pbs/spool/XXXXX.sunrise.{OU,ER}**
- You get the logs back when the job finishes (check them for errors, etc.)

Removing jobs from queue

- Running or waiting job – **qdel JOBNUMBER**

Results

- Each job runs individually
- Job (**usually**) runs in a temporary directory, unless you specify something else (but HOME **IS NOT** a good idea!)
- Batch system sets up multiple variables which could be used:
- PBS_O_WORKDIR, PBS_JOBID,PBS_JOBDIR,TMPDIR
- TMPDIR should be unique (/var/tmp/pbs.47096.sunrise) but you can also use /tmp directory - **mkdir $TMPDIR/$PBS_JOBID; cd $TMPDIR/$PBS_JOBID** … (or use mktemp)
- In general it‘s up to you to specify in your script which results to copy back and where

Setting up job environment

- Worker nodes have no idea what setup do you need for your program – the submission script **must** include all your setups (root, compilation)
- You can use variables and pass them to job:
  - **qsub … -v OUTPUT=$IWANTTHERESULTHERE … script.sh**
  - script.sh can then use $OUTPUT variable
- You can name individual jobs
  - **qsub … -N jobname**

Additional settings

- Sending emails when the job status changes (start/error/end):
  - **qsub –m e –M zhubacek@cern.ch** (e = exit, a = abort, b = beginning, n = no mail)
- Additional computing resources
  - **qsub … -l nodes=1:ppn=4 -l mem=4G**
- Batch job array (submitting multiple jobs, for example with different starting seeds:
  - **qsub –J 0-9 skript.sh**
  - Submits 10 jobs 47098[].sunrise, each job can use ${PBS_ARRAY_ID} as a main name and ${PBS_ARRAY_INDEX} for seeding individual seeds
  - Use qstat -t 47098[] for monitoring
  - In general I don’t recommend this (you can submit multiple jobs in shell for loop)

Settings - LCG tools

- Unless you really need your experiment setup, you can use ROOT (and other tools) from LCG releases:

```
export LCGENV_PATH=/cvmfs/sft.cern.ch/lcg/releases
export PATH=/cvmfs/sft.cern.ch/lcg/releases/lcgenv/latest:${PATH}
eval "`lcgenv x86_64-centos7-gcc11-opt all`"

# try LCG_101
eval "`lcgenv -p LCG_101 x86_64-centos7-gcc11-opt ROOT`"
```

- https://nms.fjfi.cvut.cz/wiki/Sunrise.fjfi.cvut.cz#LCG_Software_Elements
- Note that in general you don‘t need to recompile your code in each job because worker nodes are the same operating systém as sunrise

Summary

- 600+ CPUs available on sunrise.fjfi.cvut.cz
- PBSPro batch system (**qsub, qstat, qdel**,…)
- Use it wisely

Other options

- Don‘t think that this is unique to KF – some (2020) - KM 1000+ jader, 100Gb/s between nodes, nVidia Tesla Volta V100 KIPL: 8x Intel Xeon Platinum 8180, 2.5GHz, 28core, 6TB RAM,…
- Particle physics (CERN, Astro) – computing cluster Golias (Institute of Physics, CAS), O(10000) cores – LHC Grid Tier 2 center
- Metacentrum.cz – collection of clusters of Czech universities
- IT4I – Czech Supercomputing center (https://www.it4i.cz/infrastruktura/nase-superpocitace)
- Distributed computing resources – grid (through VO of your experiment)

BONUS

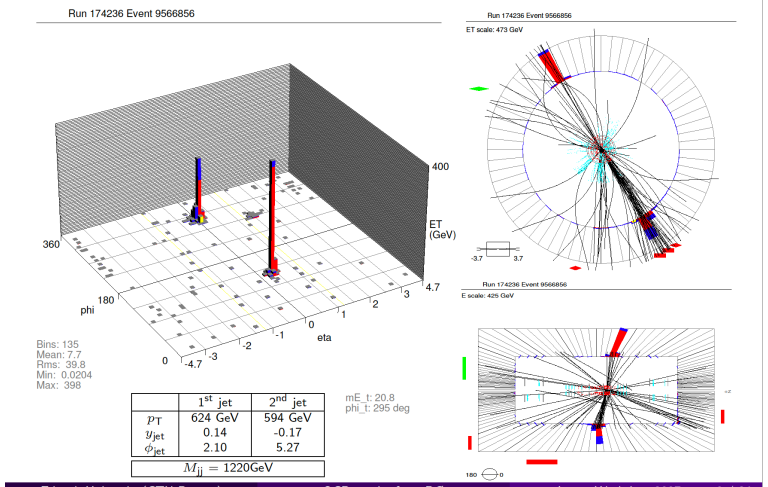
QCD results from DØ

Zdenek Hubacek

Czech Technical University in Prague
(on behalf of DØ Collaboration)

**Workshop on Low-x Physics,
Lisbon, Portugal
June 28 - July 1, 2006**

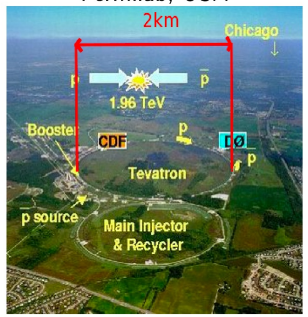
Run IIa Highest Dijet Mass Event



	1st jet	2nd jet
p_T	624 GeV	594 GeV
y_{jet}	0.14	-0.17
ϕ_{jet}	2.10	5.27
M_{jj} = 1220GeV		

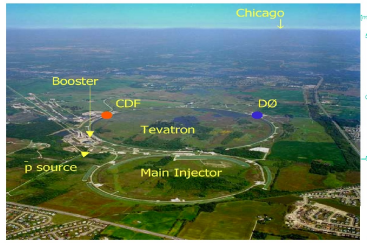
Tevatron vs LHC

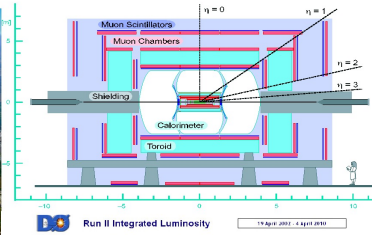
Fermilab, USA



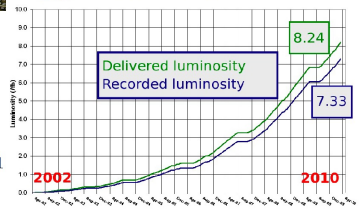
CERN, Švýcarsko

Tevatron and DØ Experiment Overview



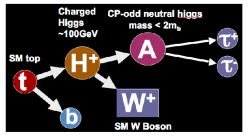


- $\sqrt{s} = 1.96\text{TeV}$
- Peak luminosity $4.0 \cdot 10^{32}\text{cm}^{-2}\text{s}^{-1}$
- Integrated luminosity $> 7\text{fb}^{-1}$

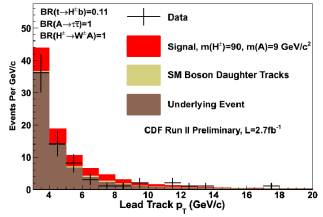


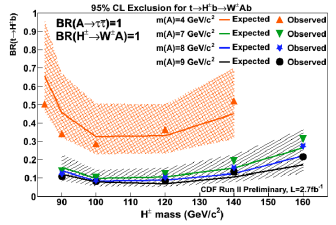
NMSSM

Includes additional CP-even and CP-odd neutral Higgs bosons and an additional neutralino - search H^+ if $M_A < 2M_b$



The τs from the A boson typically have low p_T, bad for efficient τ identification → search instead for isolated low p_T track in lepton+ 3+ jets sample with b-tag and missing E_T.

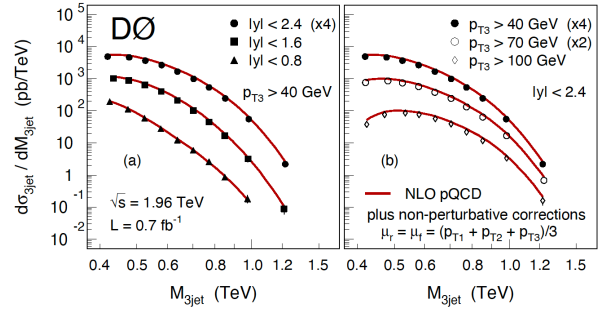




CDF Note 10104

Three-jet Invariant Mass Cross Section

Submitted to Phys. Lett. B!



- First study of the three-jet invariant mass dependence
- NLO pQCD = MSTW2008

AND YOU THOUGHT THAT COMIC SANS WAS FUNNY

ZDENĚK HUBÁČEK

JUNE 12, 2022

Death by Powerpoint

Zdenek Hubacek

Nov 12, 2021

Death by Powerpoint

Zdenek Hubacek

Nov 12, 2021

Beamer demo

- Let's finish here