PBSPro

Zdeněk Hubáček

Bílý Potok, WEJČF 2020

PBSPro

Zdeněk Hubáček

Bílý Potok, WEJČF 2020

V životě každého člověka jednou nastane okamžik…

PBSPro

Zdeněk Hubáček

Bílý Potok, WEJČF 2020

V životě každého člověka jednou nastane okamžik, kdy mu jeden počítač přestane stačit

Zadání – vygenerovat XXX eventů / zpracovat YYY eventů z ntuplů

Zadání – vygenerovat/zpracovat 10M eventů

- Zjistíte, že jste schopní udělat/zpracovat řekněme 100 eventů/s = 360kevent/h = 8.64Mevents/d

Zadání – vygenerovat/zpracovat 10M eventů

- Zjistíte, že jste schopní udělat/zpracovat řekněme 100 eventů/s = 360kevent/h = 8.64Mevents/d
- Stačí vám to?

Zadání – vygenerovat/zpracovat 10M eventů

- Zjistíte, že jste schopní udělat/zpracovat řekněme 100 eventů/s = 360kevent/h = 8.64Mevents/d
- Stačí vám to?
- Stačí to vašemu školiteli? Deadline?

Zadání – vygenerovat/zpracovat 10M eventů

- Zjistíte, že jste schopní udělat/zpracovat řekněme 100 eventů/s = 360kevent/h = 8.64Mevents/d
- Stačí vám to?

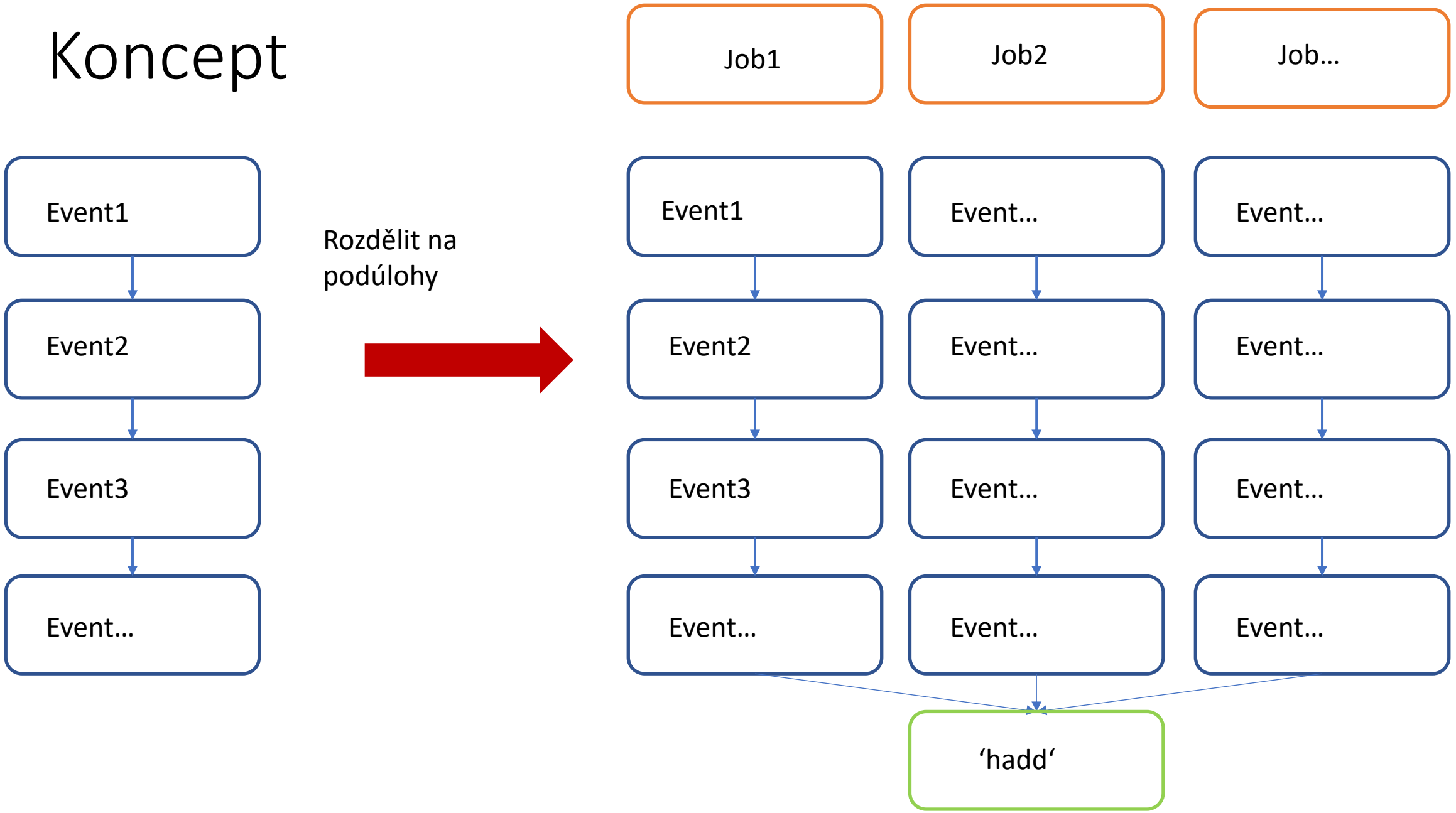
- 1x24h nebo 24x1h?

Zadání – vygenerovat/zpracovat 10M eventů

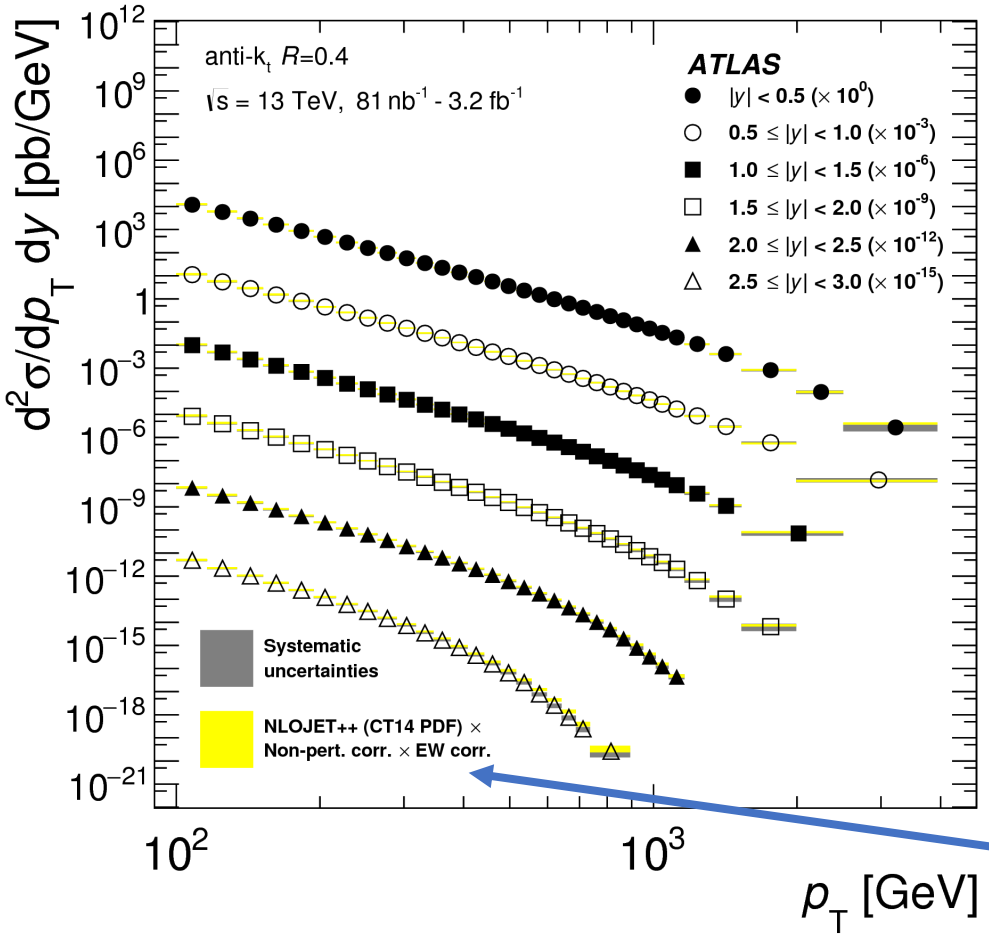
- Zjistíte, že jste schopní udělat/zpracovat řekněme 100 eventů/s = 360kevent/h = 8.64Mevents/d
- Stačí vám to?

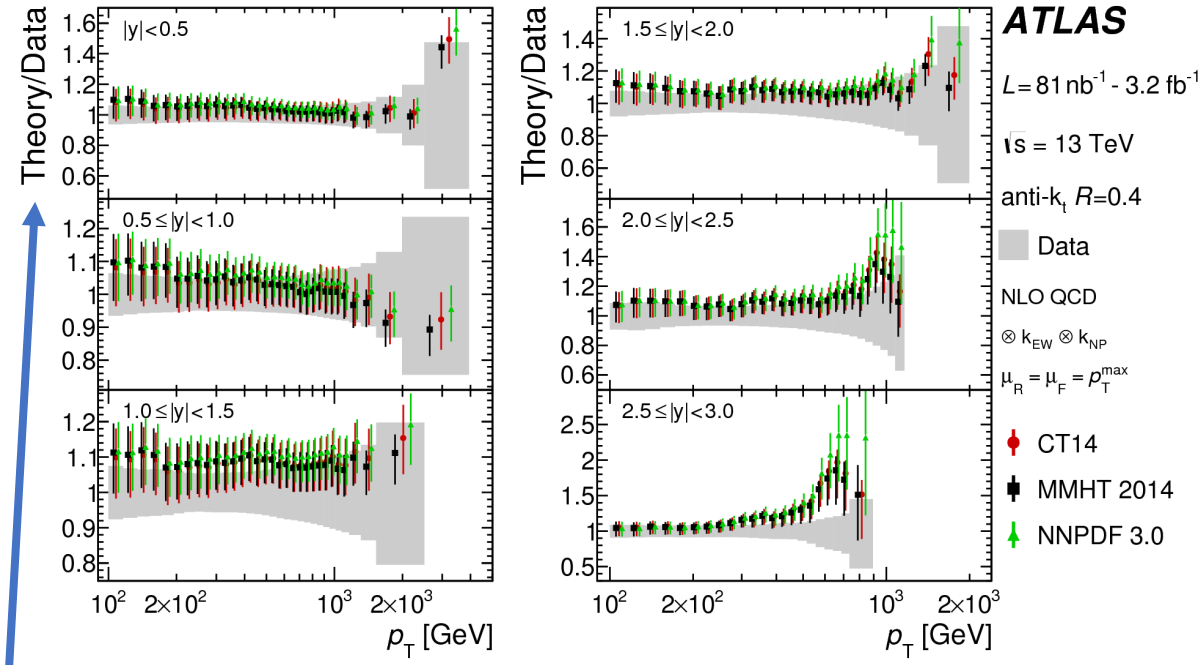
- 1x24h nebo 24x1h?
- Chtělo by to více počítačů naráz

Koncept


Event1
Event2
Event3
Event…
Rozdělit na podúlohy
Job1
Job2
Job…
Event1
Event2
Event3
Event…
Event…
Event…
Event…
Event…
Event…
Event…
Event…
Event…
‘hadd‘

ATLAS Inkluzivní účinný průřez produkce jetů

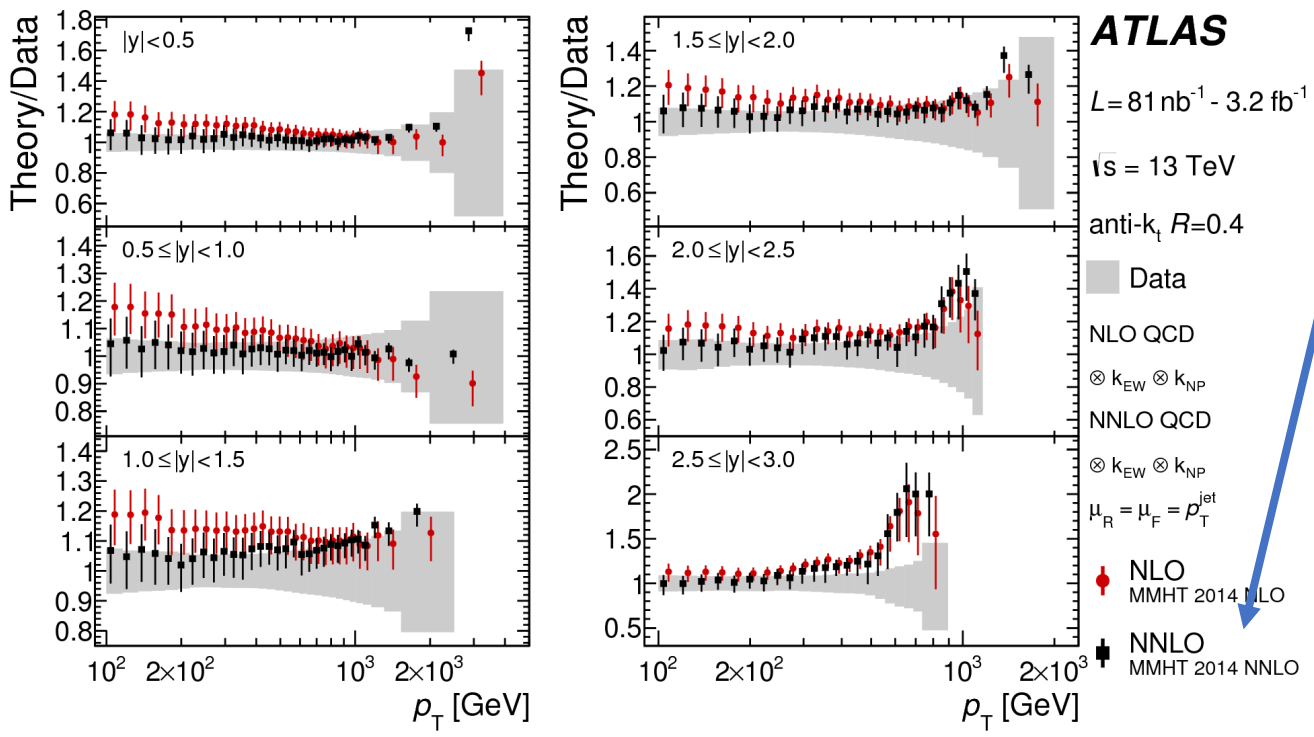




Teoretická předpověď v NLO řádu – O(10G) událostí vygenerovaných pomocí O(1000 CPU) za ~1 týden

JHEP 05 (2018) 195

Až po NNLO výpočty



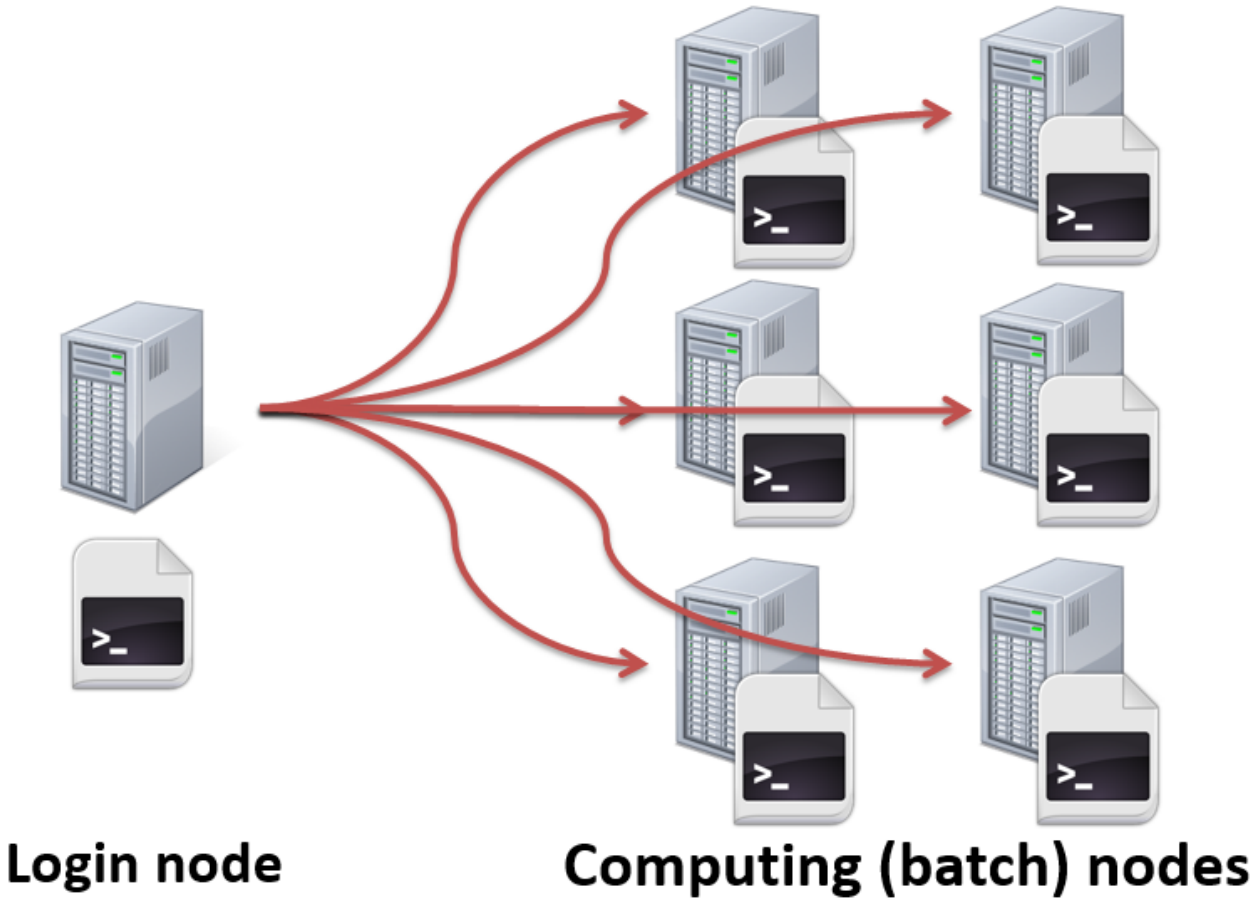
- NNLO výpočet – O(1000CPU) zhruba 1 měsíc
- Bez PDF neurčitostí, to by nám dnes trvalo zhruba to samé ještě 50x…

Chtělo by to více počítačů, pak by to šlo…

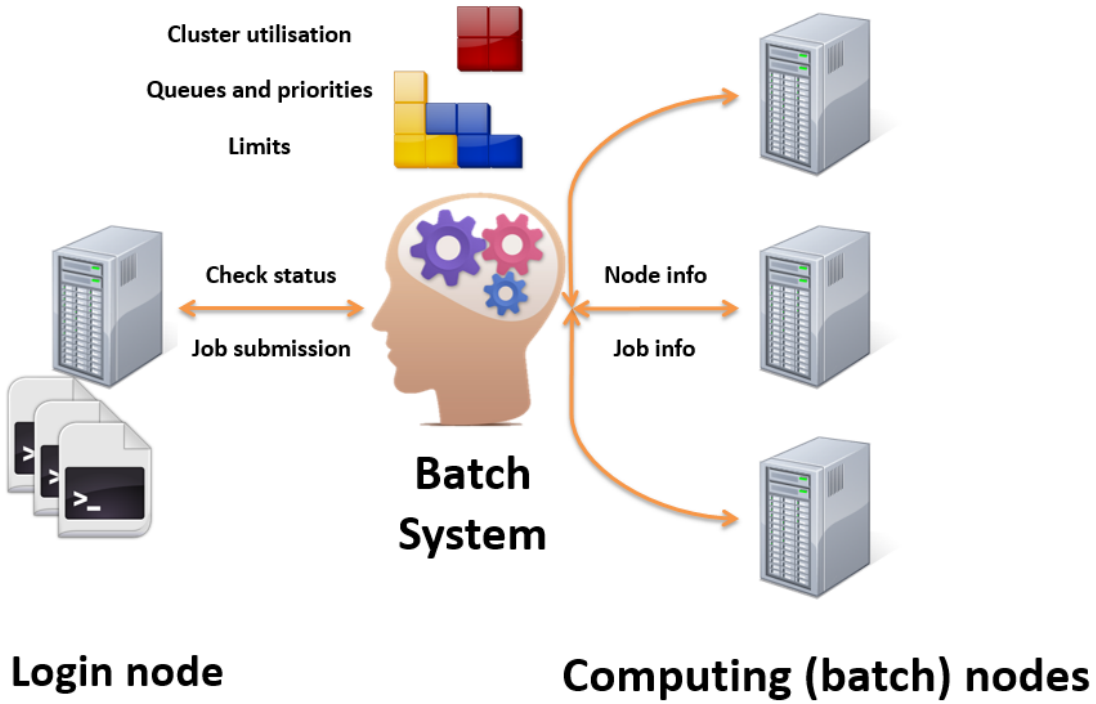
Malá část Fermilab výpočetního clusteru cca 2010

A chtělo by to nějaký rozumný systém, jak ty tisíce počítačů ovládat…

Co je to batch systém



Co je to batch systém



- Job scheduler/batch system/distributed resource management system (DRMS) kontroluje (automaticky) spouštění výpočetních úloh
- 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

A máme něco takového k dispozici?

A máme něco takového k dispozici?

- ANO! sunrise.fjfi.cvut.cz
- Ve sklepě v Břehové

A máme něco takového k dispozici?

- ANO! sunrise.fjfi.cvut.cz
- O něco skromnější ☺

Sunrise (2020)

- CPU:
  - SLC6 - 256 jader (historický hardware, Intel Xeon 14x E5420 & 9x E5620, 8 resp. 16 jader, 32 resp. 48GB RAM)
  - CentOS7 - 256 jader (nový hardware, AMD EPYC 7281, 64jader & 256GB RAM)
- Diskové pole - /data2/user_data (~200TB)
- Správci: Michal Broz, Jan Čepila, Radek Novotný
- Batch systém: PBSPro

A máme něco takového k dispozici?

- ANO! sunrise.fjfi.cvut.cz
- (dnes v podstatě každá katedra má k dispozici různě veliký výpočetní cluster, otázka je, jestli by vám tam někdo zřídil účet…)
  - Pro zajímavost: KM 1000+ jader, 100Gb/s mezi nody, nVidia Tesla Volta V100
  - KIPL: 8x Intel Xeon Platinum 8180, 2.5GHz, 28core, 6TB RAM
  - …
- Farma goliáš (FzU): https://www.farm.particle.cz/twiki/bin/view/VS/VsGolias
- Metacentrum (https://www.metacentrum.cz/cs/)
- (LCG) GRID – skrze virtuální organizaci (VO) daného experimentu (ATLAS, ALICE, STAR, PHENIX,…)

Batch systémy

- Torque, LSF, PBS, Condor, … Grid
  - Stejný princip, rozdílné příkazy/možnosti
- Na sunrise PBSPro
  - http://www.pbsworks.com/pdfs/PBSUserGuide14.2.pdf
  - Základní příkazy: **qsub, qstat, qdel**
  - Job – jedna výpočetní úloha
  - Node – jedno CPU

Základní slovníček a vlastnosti

- Základní/hlavní node – sunrise.fjfi.cvut.cz (ashley)
  - (klidně jich může být více – lxplus v CERNu)
- Výpočetní/pracovní nody (sunset01-sunset24, sunset25-sunset28)
  - Obecně na ně jednotlivý uživatel nemusí mít práva k přihlášení (**na sunrise ano**)
  - Obecně na nich nemusí mít jednotlivý uživatel práva k přístupu k $HOME adresáři (**na sunrise ano**) – potřeba řešit uživatelské nastavení/konfiguraci
  - Obecně na nich nemusí být přístup ke všem sdíleným diskům (**na sunrise ano**) – potřeba řešit přístup k uživatelským datům
  - **Nepřihlašujte** se přímo na jednotlivé nody a nespouštějte tam úlohy!! Pokud si toho admin všimne, tak vám je sestřelí/můžete přijít o práva

Přehled jednotlivých front

- Každý batch systém může mít nadefinované různé typy front – je potřeba se rozhodnout/ si vybrat jaká fronta je vhodná pro danou úlohu
  - Nejjednodušší volba – podle časové náročnosti – short, normal, long
  - Podle typu operačního systému – např. SLC6 vs centos7 na sunrise
  - Obecně možnost přímého výběru hardwaru – typu potřebuju X výpočetních jader, Y GB operační paměti a Z GB diskového místa – batch se postará o nalezení správného počítače (pokud existuje)
- V PBS: **qstat –q**
- **qstat –Q –f** (nebo **qstat –Q short –f**)

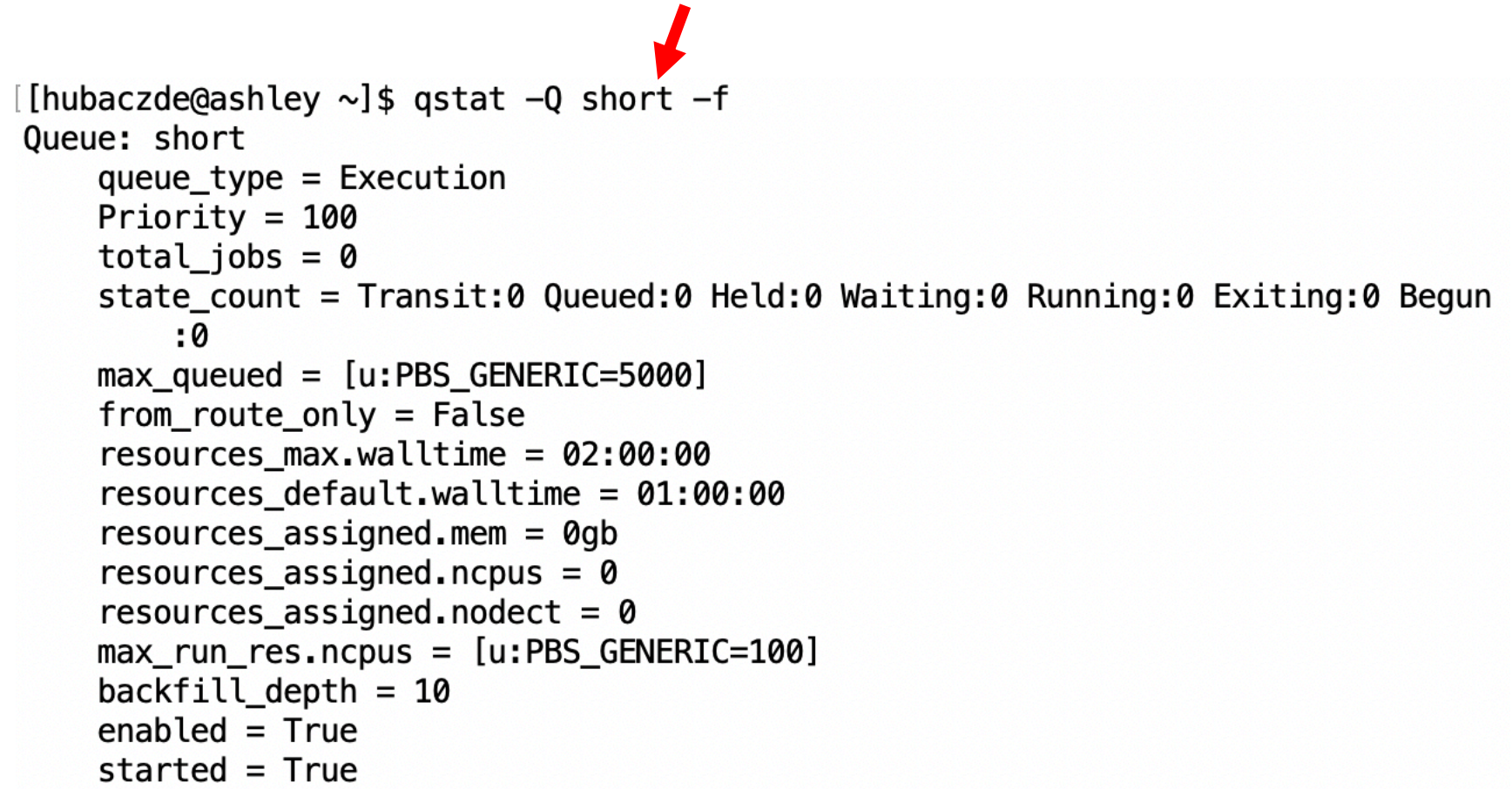
Přehled jednotlivých front

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

server: ashley.fjfi.cvut.cz

Queue            Memory CPU Time Walltime Node   Run   Que   Lm  State
---------------- ------ -------- -------- ---- ----- ----- ----  -----
backfill           --      --    24:00:00    1     0     0   --   E R
infinite           --      --       --      --     0     0   --   E R
short              --      --    02:00:00   --     0     0   --   E R
normal             --      --    24:00:00   --     0     2   --   E R
hiprio             --      --    48:00:00   --     0     0   --   E R
default            --      --    720:00:0   --     0     0   --   E R
long               --      --    720:00:0   --     1     0   --   E R
                                                ----- -----
                                                    1     2
```

Informace o detailech dané fronty



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

- Na sunrise – short (<2h), normal (2-24h), long (24-720h) walltime (pozn. některé batche rozlišují walltime a cputime)
- Každá fronta má jiné parametry např. na počet běžících úloh, atd…

Testovací job

- Příkaz na poslání jobu – **qsub (qsub –parametry submitovaciskript.sh)**
  - Nejjednodušší: **qsub –q short skript.sh**
  - Kde skript.sh bude jen jednoduchá věc jako sleep 10s

```
[hubaczde@ashley ~]$ cat skript.sh
#! /bin/bash
sleep 10s

[hubaczde@ashley ~]$ qsub skript.sh
80357.ashley.fjfi.cvut.cz
```

- 80357 je číslo jobu v batch systému

Monitoring

- **qstat**
- **qstat –u hubaczde**
- **qstat –u hubaczde –n** (vypíše i jména nodu, kde joby běží)
- **qstat 80357** (informace o jednom jobu)
- **qstat 80357 –f** (plna informace o danem jobu)
- **qstat –q short**

```
[hubaczde@ashley ~]$ qstat -u hubaczde

ashley.fjfi.cvut.cz:

                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
80357.ashley.fj hubaczde short    skript.sh   32568   1   1    2gb 01:00 R 00:00
```

- Status (S) – job čeká na spuštění (waiting – buď může být fronta plná – není žádný volný počítač nebo máte nižší priority než ostatní uživatelé (fairshare) nebo neexistuje počítač, který by splnil vaše nároky na potřebné prostředky), job běží, job je v chybovém stavu, job skončil)

Monitoring2

- http://sunrise.fjfi.cvut.cz/ganglia/

- Monitoring jednotlivých jobů – na daném nodu jsou stdout a stderr uloženy v **/var/spool/pbs/spool/XXXXX.ashley.fjfi.cvut.cz.{OU,ER}** (za předpokladu, ze tam máte přístup)

Odstranění

- Běžícího nebo čekajícího jobu – **qdel ČÍSLO**

Výsledky

- Jednotlivé výpočty běží na nodech nezávisle
- Výpočty (**obvykle**) běží v /tmp adresáři, pokud si uživatel nenastaví něco jiného (HOME adresář **NENÍ** dobrý nápad)
- Podle nastavení už může/nemusí být nastavený adresář pro daný job
- Standardně existující proměnné, které můžete využít: PBS_O_WORKDIR, PBS_JOBID,PBS_JOBDIR,TMPDIR
- Pokud TMPDIR neukazuje na jednoznačný adresář, tak je na vás, abyste si někde udělali něco jako **mkdir $TMPDIR/$PBS_JOBID; cd $TMPDIR/$PBS_JOBID** … (pozn. mktemp)
- Některé batch systémy automaticky posílají tento adresář zpátky do adresáře odkud byl job spuštěn (**$PBS_O_WORKDIR**), ale obecně je to na uživateli, aby si do skript.sh sám napsal, co se má s výsledky udělat (zkopírovat (část) zpět a podobně)

Nastavení

- Jednotlivé nody neví nic o vašem nastavení! – submitovací skript **musí** obsahovat všechny vaše setupy, které váš výpočet potřebuje (root, kompilace programu, …)
- (Většinou se musíte i sami postarat o přesun výsledků zpět)
- Možnost předat vlastní proměnnou z vašeho prostředí na pracovní node:
  - **qsub … -v OUTPUT=$TADYCHCIVYSTUP … script.sh**
  - Pak v script.sh jde použít proměnná $OUTPUT
- Možnost pojmenovat jednotlivé joby:
  - **qsub … -N jobname**

Nastavení 2

- Možnost posílání emailu o stavu jobu (na začátku/při chybě/na konci):
  - **qsub –m e –M zhubacek@cern.ch** (e = exit, a = abort, b = beginning, n = no mail)
- Extra požadavky na výpočetní výkon:
  - **qsub … -l nodes=1:ppn=4 -l mem=4G**
- Batch job array (vice stejných jobů, například s různým starting seed):
  - **qsub –J 0-10 skript.sh**
  - Pošle 10 jobů 80360[].ashley…. Uvnitř skript.sh se dá použít ${PBS_ARRAY_ID} jako hlavní jméno a ${PBS_ARRAY_INDEX} jako číslo pro volbu seed

Nastavení - LCG nástroje

- Možnost nastavení překladače, ROOT, dalších nástrojů z LCG

```
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-gcc8-opt all`"

# try LCG_96
eval "`lcgenv -p LCG_96 x86_64-centos7-gcc8-opt ROOT`"
```

- https://nms.fjfi.cvut.cz/wiki/Sunrise.fjfi.cvut.cz#LCG_Software_Elements

Shrnutí

- Máme k dispozici cca 500 výpočetních CPU na sunrise.fjfi.cvut.cz
- Batch systém, který tam běží je PBSPro (**qsub, qstat, qdel**,…)
- Respektujte základní pravidla (500CPU na měsíc pro jednoho člověka bychom řešili individuálně ☺)

Backup – složitější příklad

Starsi priklad – script, ktery poslal treba 1000 jobu pro NNLO vypocet na stare farme ve FzU

```
....
today=`date +%Y%m%d_%H%M`
for i in $(seq 0 $((NJOB-1)))
do
cp empty_job.sh new_template_${i}.sh
cp $CONFIG new_card_${i}.run
seed=$((500 + $i))
#update the seed in the runcard
perl -p -i -e "s/XXXX/${seed}/g" new_card_${i}.run
#update the config file for each job
perl -p -i -e "s/CONFIGFILE/new_card_${i}.run/g" new_template_${i}.sh
perl -p -i -e "s/SEED/${seed}/g" new_template_${i}.sh
log=log_${today}_${i}.txt
OUTPUTDIR=${PWD}/output_${today}_${i}
SUBMITDIR=${PWD}
qsub -V -q gridatlas -m e -M zhubacek@cern.ch -N nnlo_${CONFIG}_${i} \
-v OUTPUTDIR=${OUTPUTDIR},SUBMITDIR=${SUBMITDIR} -l nodes=1:ppn=4 -l mem=4G\
 -o ${log} new_template_${i}.sh
done
```

Vytvorit NJOB vypoctu

Pro kazdy kopiruju prazdnou sablonu, ktera je shodna pro vsechny

Kazdy job ma jiny seed a potecialne jiny runcard soubor na vstupu

Jednoduche prepsani retezce z prazdne sablony za spravnou hodnotu (**Perl Pie: If you only learn how to do one thing with Perl, this is it.)**

Priklad, kam kopirovat vysledky

Submit!

empty_job.sh

```
#!/bin/bash
#LCG/gcc setup
export LCGENV_PATH=/cvmfs/sft.cern.ch/lcg/releases
source /cvmfs/sft.cern.ch/lcg/releases/LCG_88/gcc/6.2.0/x86_64-slc6/setup.sh;

# path to my LHAPDF library
export PATH=/mnt/nfs06/zhubacek/hepforge4_ownlhapdf/install/bin:${PATH}
export
LD_LIBRARY_PATH=/mnt/nfs06/zhubacek/hepforge4_ownlhapdf/install/lib:${LD_LIBRA
RY_PATH}

# path to LHAPDF grids in CERN
export LHAPDF_DATA_PATH=/cvmfs/sft.cern.ch/lcg/external/lhapdfsets/current/

export WARMUPDIR=WWWW

# create a temp directory for the job:
temp=`mktemp -d`
# go to submit area
cd ${SUBMITDIR}

# create output directory
mkdir -p ${OUTPUTDIR}

#copy the executable
cp NNLOJET ${temp}/.
#copy the config
cp CONFIGFILE ${temp}/.
#copy the warmup result
cp $WARMUPDIR/* ${temp}/.

#check that everything seems fine:
cd ${temp}

echo "Checking run directory: "
ls

./NNLOJET -run CONFIGFILE

echo "Checking run directory for results: "
ls

#copy results
cp 1jet*SEED* ${OUTPUTDIR}

#return back
cd ${START}
rm -rf $temp
```