National Grid Infrastructure (NGI)
for scientific computations, collaborative research & its support services

Tomáš Rebok
CERIT-SC, Institute of Computer Science MU
MetaCentrum, CESNET z.s.p.o.
(rebok@ics.muni.cz)
National Grid Infrastructure (NGI)

- operated by MetaCentrum NGI (CESNET) since 1996
- MetaCentrum responsible for management and coordination

Distributed infrastructure

NGI integrates medium/large HW centers (clusters, powerful servers, storages) of several universities/institutions

- further integrated into the European Grid Infrastructure (EGI.eu)
Computing clusters

a set of interconnected („common“) computers

(oldschool)
Computing clusters

a set of interconnected („common“) computers (nowadays)
MetaCentrum NGI & Resource integration I.

- **MetaCentrum and CERIT-SC**
  - MetaCentrum provides its own HW resources (CESNET) and integrates resources of external providers
    - CERIT-SC/MUNI is one of them
    - others are CEITEC/NCBR, FZU, ČVUT, JČU, ZČU, UPOL, MU, TUL, etc.
      - as well as global projects like ELIXIR CZ

+ shared storages and shared SW apps
resource owners (usually) have priority access to their HW resources

- under agreed conditions
- technically accomplished using **specific scheduler queues**
  - more details later
MetaCentrum NGI

Available to all academic users from Czech universities, Academy of Science, research institutes, etc.
– commercial bodies just for public research

Offers:
– computing resources
– storage resources
– application programs

After registration, all the resources/services are available free of charge
– without any administrative burden
– users “pay” via publications with acknowledgements → results in user priorities in cases of high load

http://metavo.metacentrum.cz
Meta NGI – basic characteristics

• after registration, all the resources are available without any administrative burden
  – → ~ immediately (depending on actual usage)
  – no applications for resources

• user accounts extensions every year
  – validates users‘ relationship to an academic institution
    – federated infrastructure eduID.cz used for minimalization of users‘ burden
  – reports of user publications with acknowledgements to MetaCentrum/CERIT-SC
    – used as a proof of infrastructure benefits for Czech research area

• best-effort service
Meta NGI – computing resources available

Computing resources: ca 14900 cores (x86_64)

- nodes with lower number of computing cores: 2x4-8 cores
- nodes with medium number of comp. cores (SMP nodes): 32-80 cores
- memory (RAM) up to 1 TB per node
- nodes with high number of computing cores: SGI UV2000
  - 384 cores (x86_64), 6 TB of RAM
  - 288 cores (x86_64), 6 TB of RAM
- other „exotic“ hardware:
  - nodes with GPU cards, SSD discs, Xeon Phi, etc.

http://metavo.metacentrum.cz/cs/state/hardware.html
Meta NGI – storage resources available

ca 3 PB for operational data
- centralized storage arrays distributed through various cities in the CR
- user quota 1-3 TB on each storage array

ca 22 PB for archival data
- (HSM – MAID, tapes)
- “unlimited” user quota

http://metavo.metacentrum.cz/cs/state/nodes
Meta NGI – software available

~ 300 different applications (commercial & free/open s.)
  − see http://meta.cesnet.cz/wiki/Kategorie:Aplikace

• development tools
  − GNU, Intel, and PGI compilers, profiling and debugging tools (TotalView, Allinea), …

• mathematical software
  − Matlab, Maple, Mathematica, gridMathematica, …

• application chemistry
  − Gaussian 09, Gaussian-Linda, Gamess, Gromacs, …

• material simulations
  − Wien2k, ANSYS Fluent CFD, Ansys Mechanical, Ansys HPC…

• structural biology, bioinformatics
  − CLC Genomics Workbench, Geneious, Turbomole, Molpro, …
Meta NGI – grid environment

• *batch jobs*
  – the computations described by script files

• *interactive jobs*
  – text & graphical environment

• *cloud computing*
  – instead of running jobs with computations, users run the whole virtual machines
    focused on research computations again (not for webhosting)
    Windows & Linux images provided, user-uploaded images also supported
    more info later…
Meta VO in numbers…

- ca 14914 cores, ca 600 server nodes
- year 2017:
  - ca 4,76 mil. of running jobs
    - ca 13000 jobs per day
    - ca 2500 jobs per user
  - CPU time
    - ca 10,5 thousands of CPUyears
... and graphs
... and graphs

Users @ MetaCentrum

<table>
<thead>
<tr>
<th>Year</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>420</td>
</tr>
<tr>
<td>2011</td>
<td>491</td>
</tr>
<tr>
<td>2012</td>
<td>613</td>
</tr>
<tr>
<td>2013</td>
<td>761</td>
</tr>
<tr>
<td>2014</td>
<td>1087</td>
</tr>
<tr>
<td>2015</td>
<td>1338</td>
</tr>
<tr>
<td>2016</td>
<td>1611</td>
</tr>
<tr>
<td>2017</td>
<td>1908</td>
</tr>
</tbody>
</table>
... and graphs
Meta VO – how to become our user?

• register
  – EduID.cz => proves your academic identity using your home institution services (and credentials)

• make yourself familiar with basics of OS Linux

• compute
Research Infrastructures in CR I.

- **IT4innovations (Ostrava)**
  - 3344 computing cores ("small" supercomputer/cluster)
  - 24192 computing cores ("big" supercomputer/cluster)
  - attributes:
    - computing time assigned to research projects
    - **formal application** is necessary (evaluation of research and technical readiness + financial contributions)
    - **public competitions** 2x per year
    - if accepted, easier resource access (low number of competitive users)
  - purpose:
    - large (proven) computations using homogeneous infrastructure
Research Infrastructures in CR II.

• **National Grid Infrastructure (NGI) MetaCentrum**
  - ca 15000 computing cores (including CERIT-SC resources)
  - attributes:
    - computing time available free of charge, without formal applications
    - heterogeneous resources available (including „exotic“ ones)
    - resources shared with competitive users (sometimes hard to access)
  - purpose:
    - common smaller to middle –sized computations (larger computations after agreement)
    - preparation of computations/projects for computations at IT4innovations (~ technical readiness)

• **CERIT-SC @ ICS MU**
  - HW and SW provider (resources available through NGI)
  - main emphasis on services supporting user research
Centre CERIT-SC

A computing and research centre operating at Masaryk University

- long-term history (→ long-term experience in ICT science)
  - CERIT-SC evolved from Supercomputing Center Brno (established in 1994), and
  - participates on the operation of National Grid Infrastructure

Our mission:

- production services for computational science
  - high-performance computing clusters
  - large data storage, back-ups and data archives
  - web portals & projects’ back-office

- an application of top-level ICT in the science
  - own research in e-infrastructures (know-how)
  - novel forms of infrastructure utilization (experimental usage support)
  - research collaborations with other science areas

http://www.cerit-sc.cz
Centre CERIT-SC & NGI

CERIT-SC is an important NGI partner

- HW & SW resources provider
  - SMP nodes (2592 cores)
  - HD nodes (2624 cores)
  - SGI UV node (288 cores, 6 TB RAM)
  - SGI UV node (384 cores, 6 TB RAM)
  - storage capacity (~ 3.5 PB)

- significant personal overlaps with NGI exist

CERIT-SC (SCB) established MetaCentrum NGI

- → much research/work is performed in collaboration

http://www.cerit-sc.cz
Research support by CERIT-SC

Fact I. Common HW centers provide just a “dumb” power without any support how to effectively use it.

Fact II. Common HW centers do not participate on the users’ research aiming to help them with ICT problems.

CERIT-SC collaborates with its users:

- to help them effectively use the provided resources
- to help them to cope with their ICT research problems focusing on an application of top-level ICT in the science smaller as well as bigger (= funded) projects.
What’s the idea?

We focus on **intelligent & novel usage forms of the provided infrastructure**

- the provided HW/SW resources serve just as a tool for research and development
  → highly-flexible infrastructure (convenient to experiments)
  in comparison with NGI resources, the production computations are at the second-level of interest
- the centre aims to be equipped with cutting-edge technologies in order to allow top-level research (both internal & collaborative)
- real research collaboration with our partners
  the collaborations generate **new questions/problems for IT**
  the collaborations generate **novel opportunities for the science**
  (we DON’T want to be a common service organization)
How do we fulfill the idea?

How are the research collaborations performed?
- The work is carried via a doctoral/diploma thesis of a FI MU student
- The CERIT-SC staff supervises/consults the student and regularly meets with the research partners
  the partners provide the expert knowledge from the particular area

Collaborations through (international) projects
- CERIT-SC participates on several projects, usually developing IT infrastructure supporting the particular research area
  ELIXIR-CZ, BBMRI, Thalamoss, SDI4Apps, Onco-Steer, CzeCOS/ICOS, …
  KYPO, 3M SmartMeters in cloud, MeteoPredictions, …

Strong ICT expert knowledge available:
- Long-term collaboration with Faculty of Informatics MU
- Long-term collaboration with CESNET
  → Consultations with experts in particular areas
Selected research collaborations
3D tree reconstructions from terrestrial LiDAR scans

- partner: Global Change Research Centre - Academy of Sciences of the Czech Republic (CzechGlobe)
- the goal: to propose an algorithm able to perform fully-automated reconstruction of tree skeletons (main focus on Norway spruce trees)
  - from a 3D point cloud
    - scanned by a LiDAR scanner
    - the points provide information about XYZ coordinates + reflection intensity
  - the expected output: 3D tree skeleton
- the main issue: overlaps (→ gaps in the input data)
3D tree reconstructions from terrestrial LiDAR scans – cont’d

- the diploma thesis proposed a novel innovative approach to the reconstructions of 3D tree models
- the reconstructed models used in subsequent research
  - determining a statistical information about the amount of wood biomass and about basic tree structure
  - parametric supplementation of green biomass (young branches + needles) – a part of the PhD work
  - importing the 3D models into tools performing various analysis (e.g., DART radiative transfer model)
Selected (ongoing) collaborations II

3D reconstruction of tree forests from full-wave LiDAR scans

- subsequent PhD thesis, a preparation of joint project
- the goal: an accurate 3D reconstruction of tree forests scanned by aerial full-waveform LiDAR scans
  - possibly supplemented by hyperspectral or thermal scans, in-situ measurements,
An algorithm for determination of problematic closures in a road network

- partner: Transport Research Centre, Olomouc
- the goal: to find a robust algorithm able to identify all the road network break-ups and evaluate their impacts
- main issue: computation demands
  - the brute-force algorithms fail because of large state space
  - 2 algorithms proposed able to cope with multiple road closures
Selected (ongoing) collaborations IV.

- An application of neural networks for filling in the gaps in eddy-covariance measurements
  - partner: CzechGlobe

- Biobanking research infrastructure (BBMRI_CZ)
  - partner: Masaryk Memorial Cancer Institute, Recamo

- Propagation models of epilepsy and other processes in the brain
  - partner: MED MU, ÚPT AV, CEITEC

- Photometric archive of astronomical images

- Extraction of photometric data on the objects of astronomical images
  - 2x partner: Institute of theoretical physics and astrophysics SCI MU

- Bioinformatic analysis of data from the mass spectrometer
  - partner: Institute of experimental biology SCI MU

- Synchronizing timestamps in aerial landscape scans
  - partner: CzechGlobe

- Optimization of Ansys computation for flow determination around a large two-shaft gas turbine
  - partner: SVS FEM

- 3.5 Million smartmeters in the cloud
  - partner: CEZ group, MycroftMind

- ...
Additional services available to academic research community
Hierarchical data storages
- 22+ PB of physical capacity
- useful for data archivals, backups, etc.
- various access protocols available

Further end-user services
- FileSender
- OwnCloud

http://du.cesnet.cz
Data Services for end-users

- **FileSender** – file sharing/transfering service
  - web service intended for sending big data files
    - big = current limit is 500 GB
    - [http://filesender.cesnet.cz](http://filesender.cesnet.cz)
  - at least one user has to be an authorized infrastructure user
    - federated authentication through eduID.cz
  - authorized user is allowed to **upload a file** (and **send a notification to the receiver**)
  - if an authorized user needs to **receive data from a non-authorized user**, she sends him **an invitation link** (so he is allowed to use it for uploading the file)
FileSender – example I.

Vítejte na FileSender

FileSender je bezpečná cesta pro sdílení velkých souborů mezi všemi! Přihlaš se a nahraj své soubory nebo pozví ostatní, ať soubory nahrají oni.

Přihlásit
FileSender – example II.

Zvolte svou domovskou organizaci

Přístup ke zdroji na serveru 'filesender.cesnet.cz' vyžaduje autentizaci.

CESNET, z.s.p.o. ▼ Zvolit

Uložit tuto volbu do ukončení relace prohlížeče.
Uložit tuto volbu nastálo.

Operátorem federace eduID.cz je CESNET, z.s.p.o.

Uživatelské jméno
tkosnar

Heslo

Přihlásit
FileSender – example III.
OwnCloud

- cloud storage „like Dropbox“
  - quota: 100 GB / user
  - available through web interface
    - [https://owncloud.cesnet.cz/](https://owncloud.cesnet.cz/)
  - clients for Windows, Linux, OS X
  - clients for smartphones and tablets
  - allows sharing among a group of users
  - data backups every day
  - document versioning
  - calendars and contacts sharing
  - etc.
OwnCloud – example I.
OwnCloud – example II.
OwnCloud – example III.
OwnCloud – example IV.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>documents</td>
<td>22.8 kB</td>
<td>4 months ago</td>
</tr>
<tr>
<td>music</td>
<td>3.5 MB</td>
<td>4 months ago</td>
</tr>
<tr>
<td>ownCloudUserManual.pdf</td>
<td>1.5 MB</td>
<td>4 months ago</td>
</tr>
</tbody>
</table>
Remote collaboration support

Support for interactive collaborative work in real-time

- **videoconferences**
  HD videoconferencing support via H.323 HW/SW equipment

- **webconferences**
  SD videoconferencing support via Adobe Connect (Adobe Flash)
  [http://meetings.cesnet.cz](http://meetings.cesnet.cz)

- **special transmissions**
  HD, UHD, 2K, 4K, 8K with compressed/uncompressed video transmission (UltraGrid tool)

- **IP telephony**

Support for offline content access

- **streaming**

- **video archive**
  [http://vidcon.cesnet.cz](http://vidcon.cesnet.cz)
Security services

Security incidents handling
- detailed monitoring of possible security incidents
- the users/administrators are informed about security incidents, and
- helped to resolve the incident
- additional services:
  seminars, workshops, etc.

Security teams CSIRT-MU and CESNET-CERTS
- several successes:
  e.g., Chuck Norris botnet discovery

http://www.muni.cz/ics/services/csirt
http://csirt.cesnet.cz
Federated identity management

Czech academic identity federation eduID.cz

- provides means for inter-organizational identity management and access control to network services, while respecting the privacy of the users
- users may access multiple applications using just a single password
- service provider administrators do not have to preserve user's credentials and implement authentication
- user authentication is always performed at the home organization, user credentials are not revealed to the service providers

http://www.eduid.cz
CESNET CA certification authority

- provides the users with TERENA (Trans-European Research and Education Networking Association) certificates
  - usable for electronic signatures as well as for encryption

- CESNET CA services:
  - issues personal certificates
  - issues certificates for servers and services
  - certificates registration offices
  - certificates certification offices

http://pki.cesnet.cz
Mobility and roaming support

Eduroam.cz

– idea to enable transparent usage of (especially wireless) networks of partner (Czech as well as abroad) institutions
Communication infrastructure and its monitoring

The basis of all the services: high-speed computer network

- 100 Gbps, called CESNET2
- interconnected with pan-european network GÉANT

and its monitoring

- detailed network monitoring (quality issues as well as individual nodes behaviour) available
- automatic detection of various events, anomalies, etc.
Conclusions
Conclusions

• CESNET infrastructure:
  - *computing services* (MetaCentrum NGI & MetaVO)
  - *data services* (archivals, backups, data sharing and transfers, …)
  - *remote collaborations support services* (videoconferences, webconferences, streaming, …)
  - further supporting services (…)

• Centrum CERIT-SC:
  - *computing services* (flexible infrastructure for production and research)
  - *services supporting collaborative research*
  - user identities/accounts shared with the CESNET infrastructure

• *The message*: „If you cannot find a solution to your specific needs in the provided services, let us know - we will try to find the solution together with you…“
The CERIT Scientific Cloud project (reg. no. CZ.1.05/3.2.00/08.0144) is supported by the Operational Program Research and Development for Innovations, priority axis 3, subarea 2.3 Information Infrastructure for Research and Development.

Hands-on training
for MetaCentrum/CERIT-SC users

Tomáš Rebok
MetaCentrum, CESNET
CERIT-SC, Masaryk University
rebok@ics.muni.cz
Overview

- Introduction
- **MetaCentrum / CERIT-SC infrastructure overview**
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
Infrastructure overview
Infrastructure Access

**Frontends**
- skirt
- nympha
- hermes
- alfrid
- zuphux

**Schedulers**
- arien-pro
- wagap-pro

**Computing Nodes**
- Non-virtualised nodes:
  - mandos1
  - mandos2
- Virtualised nodes:
  - tarkil1
  - tarkil1-1
  - tarkil1-2
  - tarkil2
  - tarkil2-1
  - tarkil2-2

**Access Methods**
- ssh (Linux)
- putty (Windows)

**Domain Access**
https://wiki.metacentrum.cz/wiki/Frontend

**Description**
- All the nodes available under the domain metacentrum.cz

**Portal Access**
- PBSMon on web portal
Infrastructure System Specifics

**frontends, computing nodes**

CPUs (Brno node)
RAM
local disk /scratch/user
/home
network disk /storage/brno1/home/user
network disk /storage/brno2/home/user
network disk /storage/plzen1/home/user
network disk /software, /packages/run

**storage frontends**

storage-brno1.metacentrum.cz
NFSv4 file server with 100TB disk array

storage-brno2.metacentrum.cz
NFSv4 file server with 120TB disk array

storage-plzen1.metacentrum.cz
NFSv4 file server with 120TB disk array

replicated AFS servers for software installations
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
How to ... specify requested resources I.

- before running a job, one needs to know **what resources the job requires**
  - and how much/many of them
- for example:
  - number of **nodes**
  - number of **CPUs/cores per node**
  - an **upper estimation** of job’s runtime
  - amount of **free memory**
  - amount of **scratch space** for temporal data
  - number of requested **software licenses**
  - etc.
- the resource requirements are then **provided to the qsub utility** (when submitting a job)
  - the requested resources are **reserved for the job** by the infrastructure scheduler
    - the computation is allowed to use them

- **details about resources’ specification:**
How to ... specify requested resources II.

Graphical way:


- allows to:
  - graphically specify the requested resources
  - check, whether such resources are available
  - generate command line options for `qsub`
  - check the usage of MetaVO resources

Textual way:

- *more powerful* and (once being experienced user) *more convenient*

- see the following slides/examples →
PBS Professional – the infrastructure scheduler

- a novel scheduling system used in MetaCentrum NGI
  - see advanced information at [https://wiki.metacentrum.cz/wiki/Prostředí_PBS_Professional](https://wiki.metacentrum.cz/wiki/Prostředí_PBS_Professional)

New term – CHUNK:

- chunk = further indivisible set of resources allocated to a job on a physical node
- contains resources, which could be asked from the infrastructure nodes

- for simplicity reasons: chunk = node
  - later, we will generalize…
How to … specify requested resources III.

Chunk(s) specification:

- **general format**: `-l select=...`

**Examples:**

- 2 chunks/nodes:
  - `-l select=2`

- 5 chunks/nodes:
  - `-l select=5`

- by default, allocates just a single core in each chunk
  - → should be used together with **number of CPUs (NCPUs)** specification

- if “`-l select=...`” is not provided, just a single chunk with a single CPU/core is allocated
How to ... specify requested resources IV.

Number of CPUs (NCPUs) specification (in each chunk):

- **general format:** `-l select=...:ncpus=...`

- 1 chunk with 4 cores:
  - `-l select=1:ncpus=4`

- 5 chunks, each of them with 2 cores:
  - `-l select=5:ncpus=2`

(Advanced chunks specification:)

- **general format:** `-l select=[chunk_1][+chunk_2]...[+chunk_n]`

- 1 chunk with 4 cores and 2 chunks with 3 cores and 10 chunks with 1 core:
  - `-l select=1:ncpus=4+2:ncpus=3+10:ncpus=1`
How to … specify requested resources V.

Other useful features:

- chunks from just a **single (specified) cluster** (suitable e.g. for MPI jobs):
  - general format: `-l select=...:cl_<cluster_name>=true`
  - e.g., `-l select=3:ncpus=1:cl_doom=true`

- chunks located in a **specific location** (suitable when accessing storage in the location)
  - general format: `-l select=...:<brno|plzen|praha|...>=true`
  - e.g., `-l select=1:ncpus=4:brno=true`

- **exclusive node(s) assignment** (useful for testing purposes, all resources available):
  - general format: `-l select=... -l place=exclhost`
  - e.g., `-l select=1 -l place=exclhost`

- **negative** specification:
  - general format: `-l select=...:<feature>=false`
  - e.g., `-l select=1:ncpus=4:hyperthreading=false`

- ...

A list of nodes’ features can be found here: [http://metavo.metacentrum.cz/pbsmon2/props](http://metavo.metacentrum.cz/pbsmon2/props)
How to ... specify requested resources VI.

Specifying memory resources (default = 400mb):
- **general format**: `-l select=...:mem=...<suffix>`
  - e.g., `-l select=...:mem=100mb`
  - e.g., `-l select=...:mem=2gb`

Specifying job’s maximum runtime (default = 24 hours):
- it is necessary to specify an upper limit on job’s runtime:
- **general format**: `-l walltime=[[hh:]mm:]ss`
  - e.g., `-l walltime=13:00`
  - e.g., `-l walltime=2:14:30`
How to ... specify requested resources VII.

Specifying requested scratch space:

- useful, when the application performs I/O intensive operations OR for long-term computations (reduces the impact of network failures)
- requesting scratch is mandatory (no defaults)
- **scratch space specification** : `-l select=...:scratch_type=...<suffix>`
  - e.g., `-l select=...:scratch_local=500mb`

**Types of scratches:**

- `scratch_local`
- `scratch_ssd`
- `scratch_shared`
Why to use scratches?

Data processing using central storage
- low computing performance (I/O operations)
- dependency on (functional) network connection
- high load on the central storage

Data processing using scratches
+ highest computing performance
+ resilience to network connection failures
+ minimal load on the central storage
How to use scratches?

- **private scratch directory for particular job**
  - `/scratch/$USER/job_$PBS_JOBID` directory for (local) job’s scratch
  - `/scratch.ssd/$USER/job_$PBS_JOBID` for job’s scratch on SSD
  - `/scratch.shared/$USER/job_$PBS_JOBID` for shared job’s scratch
- **the master directory** `/scratch*/$USER` is not available for writing
- **to make things easier**, there is a `SCRATCHDIR` environment variable available in the system
  - (within a job) points to the assigned scratch space/location

Please, clean scratches after your jobs

- there is a “clean_scratch” utility to perform safe scratch cleanup
  - also reports scratch garbage from your previous jobs
  - usage example will be provided later
How to ... specify requested resources VIII.

Specifying requested software licenses:

- necessary when an application requires a SW licence
  - the job becomes started once the requested licences are available
  - the information about a licence necessity is **provided within the application description** (see later)

- **general format:** `-l <lic_name>=<amount>`
  - e.g., `-l matlab=2`
  - e.g., `-l gridmath8=20`

(advanced) Dependencies among jobs

- allows to create a workflow
  - e.g., to start a job once another one successfully finishes, breaks, etc.
- see qsub’s “–W” option (man qsub)
  - e.g., `$ qsub ... –W depend=afterok:12345.arien-pro.ics.muni.cz`
Resource chunks vs. nodes

How do chunks correspond to nodes?

- **chunks arrangement** – option „-l place=...“
  - `place=free`: chunks are **free to spread** over available nodes
    - default behaviour
  - `place=pack`: all chunks will be allocated **on the same node**
    - the node has to have enough resources available
  - `place=scatter`: each chunk will be allocated **on a different node**

```plaintext
select=3:ncpus=2:mem=5g:scratch_local=10g
```

![Diagram showing resource allocation with different chunk arrangements]
Chunks arrangement

- free vs. pack vs. scatter

arrangement (free/pack/scatter)
Chunks arrangement

- **free** vs. pack vs. scatter

arrangement = free
Chunks arrangement

- free vs. **pack** vs. scatter

\[ \text{arrangement} = \text{pack} \]
Chunks arrangement

- free vs. pack vs. **scatter**

arrangement = scatter
Chunks grouping

- useful for distributed jobs
- `-l place=group=infiniband`

```
Infiniband=A
  Vnode1
  Vnode2
  Vnode3
  Vnode4

Infiniband=B
  Vnode5
  Vnode6
  Vnode7
  Vnode8
  Vnode9
  Vnode10

Infiniband=C
```

```
Vnode11
Vnode12
Vnode13
Vnode14
Vnode15
Vnode16
Vnode17
Vnode18
Vnode19
Vnode20
Vnode21
Vnode22
Vnode23
Vnode24
```
Chunks grouping

- useful for distributed jobs
- -l place=group=infiniband
Chunks grouping

- useful for distributed jobs
- `-l place=group=infiniband`

![Diagram showing Chunks grouping]
Chunks grouping

- useful for distributed jobs
- `-l place=group=infiniband`
How to ... specify requested resources IX.

Questions and Answers:

- **Why is it necessary to specify the resources in a proper number/amount?**
  - because when a job consumes more resources than announced, it will be **killed** by us (you’ll be informed)
  - otherwise it may influence other processes running on the node

- **Why is it necessary not to ask for excessive number/amount of resources?**
  - the jobs having smaller resource requirements are started (i.e., get the time slot) **faster**

- **Any other questions?**
Questions and Answers:

- **Why is it necessary to specify the resources in a proper number/amount?**
  - because when a job consumes more resources than announced, it will be killed by us (you’ll be informed)
  - otherwise it may influence other processes running on the node

- **Why is it necessary not to ask for excessive number/amount of resources?**
  - the jobs having smaller resource requirements are started

**See more details about PBSpro scheduler:**

Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- **How to … run an interactive job**
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
How to … run an interactive job I.

Interactive jobs:
- result in getting a prompt on a single (master) node
  - one may perform interactive computations
  - the other nodes, if requested, remain allocated and accessible (see later)

How to ask for an interactive job?
- add the option “-I” to the qsub command
- e.g., qsub -I -l select=1:ncpus=4

Example (valid for this demo session):
- qsub -I -q MetaSeminar  # ( -l select=1:ncpus=1)
How to ... run an interactive job II.

Textual mode: simple

Graphical mode:

- (preffered) remote desktops based on VNC servers (pilot run):
  - available from frontends as well as computing nodes (interactive jobs)
    - module add gui
    - gui start [-s] [-w] [-g GEOMETRY] [-c COLORS]
      - uses one-time passwords
      - allows to access the VNC via a supported TigerVNC client or WWW browser
      - allows SSH tunnels to be able to connect with a wide-range of clients
      - allows to specify several parameters (e.g., desktop resolution, color depth)
    - gui info [-p] ... displays active sessions (optionally with login password)
    - gui stop [sessionID] ... allows to stop/kill an active session

- see more info at
How to ... run an interactive job II.
How to ... run an interactive job II.

Graphical mode (further options):

- **(fallback) tunnelling a display through ssh** (Windows/Linux):
  - connect to the frontend node having SSH forwarding/tunneling enabled:
    - Linux: `ssh -X skirit.metacentrum.cz`
    - Windows:
      - install an XServer (e.g., Xming)
      - set Putty appropriately to enable X11 forwarding when connecting to the frontend node
        - Connection → SSH → X11 → Enable X11 forwarding
  - ask for an interactive job, **adding “-x” option** to the qsub command
    - e.g., `qsub -I -x -l select=` ...

- **(tech. gurus) exporting a display** from the master node to a Linux box:
  - export `DISPLAY=mycomputer.mydomain.cz:0.0`
  - on a Linux box, run “xhost +” to allow all the remote clients to connect
    - be sure that your display manager allows remote connections
Questions and Answers:

- **How to get an information about the other nodes/chunks allocated (if requested)?**
  - `master_node$ cat $PBS_NODEFILE`
  - works for batch jobs as well

- **How to use the other nodes/chunks?** *(holds for batch jobs as well)*
  - MPI jobs use them automatically
  - otherwise, use the `pbsdsh` utility *(see "man pbsdsh" for details)* to run a remote command
  - if the `pbsdsh` does not work for you, use the `ssh` to run the remote command

- **Any other questions?**
How to ... run an interactive job III.

Questions and Answers:

- **How to get an information about the other nodes/chunks allocated?**

  Hint:
  
  - there are several useful environment variables one may use
    - $ set | grep PBS
    - e.g.:
      - PBS_JOBID ... job’s identificator
      - PBS_NUM_NODES, PBS_NUM_PPN ... allocated number of nodes/processors
      - PBS_O_WORKDIR ... submit directory
      - ...

- **How to use the other nodes/chunks?** (holds for batch jobs as well)
  
  - MPI jobs use them automatically
  - otherwise, use the `pbsdsh` utility (see "man pbsdsh" for details)
  - if the `pbsdsh` does not work for you, use the `ssh` to run a remote command
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
How to … use application modules I.

Application modules:

- the **modular subsystem** provides a user interface to modifications of user environment, which are necessary for running the requested applications
- allows to “add” an application to a user environment

**getting a list** of available application modules:

- $ module avail
- $ module avail matl
  - provides the documentation about modules’ usage
  - besides others, includes:
    - information whether it is necessary to ask the scheduler for an available licence
    - information whether it is necessary to express consent with their licence agreement
How to … use application modules II.

Application modules:

- **loading** an application into the environment:
  - `$ module add <modulename>`
  - e.g., `module add maple`

- **listing** the already loaded modules:
  - `$ module list`

- **unloading** an application from the environment:
  - `$ module del <modulename>`
  - e.g., `module del openmpi`

**Note:** An application may require to express consent with its licence agreement before it may be used (see the application’s description). To provide the agreement, visit the following webpage: [https://metavo.metacentrum.cz/cs/myaccount/licence.html](https://metavo.metacentrum.cz/cs/myaccount/licence.html)

- for more information about application modules, see [https://wiki.metacentrum.cz/wiki/Application_modules](https://wiki.metacentrum.cz/wiki/Application_modules)
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- **How to … run a batch job**
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
How to ... run a batch job I.

Batch jobs:
- perform the computation as described in their **startup script**
  - the submission results in getting a **job identifier**, which further serves for getting more information about the job (see later)

- **How to submit a batch job?**
  - add the reference to the startup script to the `qsub` command
  - e.g., `qsub -l select=3:ncpus=4 <myscript.sh>`

- **Example** (valid for this demo session):
  - `qsub -q MetaSeminar -l select=1:ncpus=1 myscript.sh`
  - results in getting something like “12345.arien-pro.ics.muni.cz”
How to ... run a batch job I.

### Hint:

- create the file `myscript.sh` with the following content:
  
  ```bash
  #!/bin/bash
  
  # my first batch job
  uname -a
  ```

- see the standard output file (`myscript.sh.o<JOBID>`) 
  
  ```bash
  $ cat myscript.sh.o<JOBID>
  ```

- `qsub -q MetaSeminar -l select=1:ncpus=1 myscript.sh`
- results in getting something like “12345.arien-pro.ics.muni.cz”
How to ... run a batch job II.

Startup script skelet: (non IO-intensive computations)
- use just when you know, what you are doing...

```bash
#!/bin/bash

DATADIR="/storage/brno2/home/$USER/"  # shared via NFSv4

cd $DATADIR

# ... load modules & perform the computation ...
```

- further details – see
How to ... run a batch job III.

**Recommended startup script skelet: (IO-intensive computations or long-term jobs)**

```
#!/bin/bash

# set a handler to clean the SCRATCHDIR once finished
trap 'clean_scratch' TERM EXIT

# if temporal results are important/useful
trap 'cp -r $SCRATCHDIR/neuplna.data $DATADIR && clean_scratch' TERM

# set the location of input/output data
DATADIR="/storage/brno2/home/$USER/"
DATADIR="$PBS_O_WORKDIR"

# prepare the input data
cp $DATADIR/input.txt $SCRATCHDIR || exit 1

# go to the working directory and perform the computation
cd $SCRATCHDIR

# ... load modules & perform the computation ...

# copy out the output data
# if the copying fails, let the data in SCRATCHDIR and inform the user
cp $SCRATCHDIR/output.txt $DATADIR || export CLEAN_SCRATCH=false
```
How to ... run a batch job IV.

Using the application modules within the batch script:

- include the initialization line ("source ...") if necessary:
  - if you experience problems like "module: command not found"

    ```
source /software/modules/init
...
module add maple
```

Getting the job’s standard output and standard error output:

- once finished, there appear two files in the directory, which the job has been started from:
  - `<job_name>.o<jobID>` ... standard output
  - `<job_name>.e<jobID>` ... standard error output
  - the `<job_name>` can be modified via the "-N" qsub option
How to ... run a batch job V.

**Job attributes specification:**

in the case of batch jobs, the requested resources and further job information (*job attributes* in short) may be specified either on the command line (see “`man qsub`”) or directly within the script:

- by adding the “`#PBS`” directives (see “`man qsub`”):
  ```
  #PBS -N Job_name
  #PBS -l select=2:ncpus=1:mem=320kb:scratch_local=100m
  #PBS -m abe
  #
  < ... commands ... >
  ```

- the submission may be then simply performed by:
  - `$ qsub myscript.sh`

  if options are provided both in the script and on the command-line, the **command-line arguments override the script ones**
How to ... run a batch job VI. (complex example)

#!/bin/bash
#PBS -l select=1:ncpus=2:mem=500mb:scratch_local=100m
#PBS -m abe

# set a handler to clean the SCRATCHDIR once finished
trap "clean_scratch" TERM EXIT

# set the location of input/output data
DATADIR="$PBS_O_WORKDIR"

# prepare the input data
cp $DATADIR/input.mpl $SCRATCHDIR || exit 1

# go to the working directory and perform the computation
cd $SCRATCHDIR

# load the appropriate module
module add maple

# run the computation
maple input.mpl

# copy out the output data (if it fails, let the data in SCRATCHDIR and inform the user)
cp $SCRATCHDIR/output.gif $DATADIR || export CLEAN_SCRATCH=false
How to ... run a batch job VII.

Questions and Answers:

- *Should you prefer batch or interactive jobs?*
  - definitely the **batch ones** – they use the computing resources more effectively
  - use the interactive ones just for testing your startup script, GUI apps, or data preparation

- *Any other questions?*
Example:
- Create and submit a batch script, which performs a simple Maple computation, described in a file:

```
plotsetup(gif, plotoutput=`myplot.gif`,
plotoptions=`height=1024,width=768`);
plot3d( x*y, x=-1..1, y=-1..1, axes = BOXED, style = PATCH);
```

- process the file using Maple (from a batch script):
  - hint: $ maple <filename>
How to … run a batch job VIII.

Example:

- Create and submit a batch script, which performs a simple Maple computation, described in a file:

  ```maple
  plotsetup(gif, plotoutput=`myplot.gif`,
             plotoptions=`height=1024,width=768`);
  plot3d( x*y, x=-1..1, y=-1..1, axes = BOXED, style = PATCH);
  ```

- process the file using Maple (from a batch script):
  - hint: `$ maple <filename>`

**Hint:**

- see the solution at
  `/storage/brno2/home/jeronimo/MetaSeminar/latest/Maple`
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- **How to ... determine a job state**
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
How to … determine a job state I.

Job identifiers

- every job (no matter whether interactive or batch) is **uniquely identified** by its identifier (JOBID)
  - e.g., 12345.arien-pro.ics.muni.cz
- to obtain any information about a job, the **knowledge of its identifier is necessary**
  - how to list all the recent jobs?
    - frontend$ qstat  (run on any frontend)
      - to include finished ones, run $ qstat -x
  - how to list all the recent jobs of a specific user?
    - frontend$ qstat -u <username>  (again, any frontend)
      - to include finished ones, run $ qstat -x -u <username>
How to determine a job state?

- **graphical way** – see PBSMON
  - list all your jobs and click on the particular job’s identifier

- **textual way** – `qstat` command (see `man qstat`)
  - brief information about a job: `$ qstat JOBID`
    - informs about: job’s state (\texttt{Q=queued}, \texttt{R=running}, \texttt{E=exiting}, \texttt{F=finished}, …), job’s runtime, …
  - complex information about a job: `$ qstat -f JOBID`
    - shows all the available information about a job
    - useful properties:
      - \texttt{exec\_host} -- the nodes, where the job did really run
      - \texttt{resources\_used}, \texttt{start/completion time}, \texttt{exit status}, …
  - necessary to add „\texttt{-x}“ \textbf{option} when examining already finished job(s)
How to … determine a job state III.

Hell, when my jobs will really start?

- nobody can tell you 😊
  - the God/scheduler decides (based on the other job’s finish)
  - we’re working on an estimation method to inform you about its probable startup

- check the queues’ fulfilment:
  - the higher fairshare (queue’s AND job’s) is, the earlier the job will be started

- stay informed about job’s startup / finish / abort (via email)
  - by default, just an information about job’s abortion is sent
  - → when submitting a job, add “-m abe” option to the qsub command to be informed about all the job’s states
    - or “#PBS -m abe” directive to the startup script
How to … determine a job state IV.

Monitoring running job’s stdout, stderr, working/temporal files

1. via ssh, log in directly to the execution node(s)
   - how to get the job’s execution node(s)?
   - to examine the working/temporal files, navigate directly to them
     - logging to the execution node(s) is necessary -- even though the files are on a shared storage, their content propagation takes some time
   - to examine the stdout/stderr of a running job:
     - navigate to the /var/spool/pbs/spool/ directory and examine the files:
       - $PBS_JOBID.OU for standard output (stdout – e.g., “1234.arien-pro.ics.muni.cz.OU”)
       - $PBS_JOBID.ER for standard error output (stderr – e.g., “1234.arien-pro.ics.muni.cz.ER”)

Job’s forcible termination

- $ qdel JOBID  (the job may be terminated in any previous state)
- during termination, the job turns to E (exiting) and finally to F (finished) state
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?
- Real-world examples
- Appendices
Another mini-HowTos …

- **How to use privileged resources?**
  - If your institution/project integrates HW resources, a defined group of users may have priority access to them
  - Technically accomplished using scheduler queues
  - A job has to be **submitted to the particular queue**
    - qsub -l select=… -l walltime=… -q PRIORITY_QUEUE script.sh
    - E.g., ELIXIR CZ project integrates a set of resources
    - Priority queue „elixir_2w“ available for ELIXIR CZ users

- Moving jobs between scheduler queues
  - From priority queue **to default queue**
    - qmove default JOBID
  - From default queue(s) **to a priority queue**
    - qmove elixir_2w JOBID
Another mini-HowTos ...

- how to make your SW tool available within MetaVO?

  - **commercial apps:**
    - **assumption:** you own a license, and the license allows the application to be run on our infrastructure (nodes not owned by you, located elsewhere, etc.)
    - once installed, we can restrict its usage just for you (or for your group)

  - **open-source/freeware apps:**
    - you can compile/install the app in your HOME directory
    - OR you can install/compile the app on your own and ask us to make it available in the software repository
      - compile the application in your HOME directory
      - prepare a **modulefile** setting the application environment
        - inspire yourself by modules located at `/packages/run/modules-2.0/modulefiles`
      - test the app/modulefile
        - `$ export MODULEPATH=$MODULEPATH:$HOME/myapps`

  - OR you can ask us for preparing the application for you
Another mini-HowTos ...

- how to ask for nodes equipped by GPU cards?
  - determine, **how many GPUs** your application will need (\(-l \ ngpus=X\))
    - consult the HW information page: [http://metavo.metacentrum.cz/cs/state/hardware.html](http://metavo.metacentrum.cz/cs/state/hardware.html)
  - determine, **how long** the application will run (if you need more, let us know)
    - gpu queue … maximum runtime 1 day
    - qpu_long queue … maximum runtime 1 week
  - make the submission:
    - \$ qsub -l select=1:ncpus=4:mem=10g:ngpus=1 -q gpu_long –l walltime=4d …
    - specific GPU cards by restricting the cluster:
      - qsub -l select=...:cl_doom=true ...
  - **do not change** the CUDA_VISIBLE_DEVICES environment variable
    - it’s automatically set in order to determine the GPU card(s) that has/have been reserved for your application
  - details about GPU cards performance within MetaVO:
Another mini-HowTos …

How to ask for nodes equipped with Xeon Phi?

**phi[1-6].cerit-sc.cz**

- new cluster purchased by CERIT-SC
  - available through “phi” queue (PBS Pro) on zuphux.cerit-sc.cz frontend
    
    ```bash
    zuphux$ qsub -q phi -l select=...
    ```

- the newest generation of Xeon Phi (7210 Knights Landing)
  - currently, the only installation in the CR

- see more details at
  

**Installation specifics:**

- `/storage/brno3-cerit` is the only central storage **directly mounted**
  - all the rest storages available through SCP only
Central storages of phi.cerit-sc.cz cluster

Central storages not available through NFS
i.e. /storage/XXX/home/<username>

– technical reasons
– data storages available through SCP
  • besides brno3-cerit, which is mounted directly
  • in most cases, just a minor change in your scripts

<table>
<thead>
<tr>
<th>NFS sharing (most clusters)</th>
<th>SCP sharing (phi[1-6].cerit-sc.cz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATADIR=&quot;/storage/brno2/home/&lt;username&gt;/example&quot;</td>
<td>DATADIR=&quot;storage-brno2.metacentrum.cz:~/example&quot;</td>
</tr>
<tr>
<td><code>cp</code> –R $DATADIR/mydata $SCRATCHDIR</td>
<td><code>scp</code> –R $DATADIR/mydata $SCRATCHDIR</td>
</tr>
</tbody>
</table>
Another mini-HowTos …

- how to transfer large amount of data to computing nodes?
  - copying through the frontends/computing nodes may not be efficient (hostnames are storage-XXX.metacentrum.cz)
    - XXX = brno2, brno3-cerit, plzen1, budejovice1, praha1, ...
  - → connect directly to the storage frontends (via SCP or SFTP)
    - $ sftp storage-brno2.metacentrum.cz
    - $ scp <files> storage-plzen1.metacentrum.cz:<dir>
    - etc.
  - use FTP only together with the Kerberos authentication
    - otherwise insecure
Another mini-HowTos …

- how to get information about your quotas?
  - by default, all the users have quotas on the storage arrays (per array)
    - may be different on every array
  - to get an information about your quotas and/or free space on the storage arrays
    - **textual way:** log-in to a MetaCentrum frontend and see the “motd” (information displayed when logged-in)
    - **graphical way:**

- how to restore accidentally erased data
  - the storage arrays (⇒ including homes) are regularly backed-up
    - several times a week
  - → write an email to [meta@cesnet.cz](mailto:meta@cesnet.cz) specifying what to restore
Another mini-HowTos …

- how to secure private data?
  - by default, all the data are readable by everyone
  - → use common Linux/Unix mechanisms/tools to make the data private
    - r,w,x rights for user, group, other
    - e.g., chmod go= $filename$
      - see man chmod
      - use “–R” option for recursive traversal (applicable to directories)

- how to share data among working group?
  - ask us for creating a common unix user group
    - user administration will be up to you (GUI frontend is provided)
  - use common unix mechanisms for sharing data among a group
    - see “man chmod” and “man chgrp”
how to use SGI UV2000 nodes? (ungu, urga . cerit-sc.cz)

- because of their nature, these nodes are not – by default – used by common jobs
  - to be available for jobs that really need them
- to use these nodes, one has to submit the job to a specific queue called “uv”
  - $ qsub -l select=1:ncpus=X:mem=Yg -q uv
    -l walltime=Zd ...
  - to use a specific UV node, submit e.g. with
    $ qsub -q uv -l select=1:ncpus=X:cl_urga=true ...
- for convenience, submit from zuphux.cerit-sc.cz frontend
Another mini-HowTos ...

- how to run a set of (managed) jobs?
  - some computations consist of a set of (managed) sub-computations
  - optional cases:
    - the computing workflow is known when submitting
      - specify dependencies among jobs
        - qsub’s “-W” option (man qsub)
      - in case of many parallel subjobs, use „job arrays“ (qsub‘s „-J“ option)
    - the computing workflow depends on result(s) of subcomputations
      - run a master job, which analyzes results of subjobs and submits new ones
        - the master job should be submitted to a node dedicated for low-performance (controlling/re-submitting) tasks
          - available through the „oven“ queue
          - qsub -q oven -l select=1:ncpus=... control_script.sh
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- Another mini-HowTos ...
- What to do if something goes wrong?

- Real-world examples
- Appendices
What to do if something goes wrong?

1. check the MetaVO/CERIT-SC documentation, application module documentation
   - whether you use the things correctly

2. check, whether there haven’t been any infrastructure updates performed
     - one may stay informed via an RSS feed

3. write an email to meta@cesnet.cz, resp. support@cerit-sc.cz
   - your email will create a ticket in our Request Tracking system
     - identified by a unique number → one can easily monitor the problem solving process
   - please, include as good problem description as possible
     - problematic job’s JOBID, startup script, problem symptoms, etc.
Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
Real-world examples

Examples:
- Maple
- Gaussian + Gaussian Linda
- Gromacs (CPU + GPU)
- Matlab (parallel & GPU)
- Ansys CFX
- OpenFoam
- Echo
- R – Rmpi

Demo sources:
/storage/brno2/home/jeronimo/MetaSeminar/latest

Command: `cp -r /storage/brno2/home/jeronimo/MetaSeminar/latest $HOME`
Projekt CERIT Scientific Cloud (reg. no. CZ.1.05/3.2.00/08.0144) byl podporován operačním programem Výzkum a vývoj pro inovace, 3 prioritní osy, podoblasti 2.3 Informační infrastruktura pro výzkum a vývoj.

Overview

- Introduction
- MetaCentrum / CERIT-SC infrastructure overview
- How to … specify requested resources
- How to … run an interactive job
- How to … use application modules
- How to … run a batch job
- How to … determine a job state
- Another mini-HowTos …
- What to do if something goes wrong?

- Real-world examples
- Appendices
Appendices

- Common mistakes in computations
- How to deal with parallel/distributed computations?
- Other computing possibilities
  - MetaCloud
  - Hadoop (MapReduce)
  - Specialized frontends – Galaxy, Chipster, …
Common mistakes in computations
Common mistakes in computations

Feel free to use the infrastructure – if something crashes, it’s our fault. 😊
Big data transfers

Do not copy higher amounts of data through frontends

- slower transfer
- frontends load
Big data transfers

Do not copy higher amounts of data through frontends
– slower transfer
– frontends load

Data could be copied directly through storage frontends
– SCP, WinSCP
  /storage/brno2 -> storage-brno2.metacentrum.cz
  /storage/brno3-cerit -> storage-brno3-cerit.metacentrum.cz
  ...
  • https://wiki.metacentrum.cz/wiki/Working_with_data/Direct_access_to_data_storages
Big data transfers

Do not copy higher amounts of data through frontends
  – slower transfer
  – frontends load

Data could be copied directly through storage frontends
  – SCP, WinSCP
    
    /storage/brno2 -> storage-brno2.metacentrum.cz
    /storage/brno3-cerit -> storage-brno3-cerit.metacentrum.cz
    
    …
    
Computations and central storages

Do not run computations that compute over data located at central storages

- especially the I/O-intensive ones
  - increases central storage load and makes the computation slower

Compute over local copies in scratch directories

- benefits:
  - faster computations
  - computations do not rely on the availability of the central storage

- usage:
  - $ qsub -l select=1:ncpus=4:scratch_local=1gb ...
  - cp /storage/.../home/<username>/mydata $SCRATCHDIR/mydata
  - cd $SCRATCHDIR
  - <compute>
  - cp $SCRATCHDIR/results /storage/.../home/<username>/results
  - ....:scratch_shared=Xgb ... shared scratch (distributed computations)
  - ....:scratch_ssd=Xgb ... local scratch – SSD disks
Data in scratches

Clean the scratches once computations finish

– scratch data could be seen similarly as data in RAM memory
  • once a computation finishes, the data should be cleaned
– scratches are automatically cleaned by us
  • usually after 2 weeks a computation finishes

Clean scratch after your computations

– "clean_scratch" utility

– usage (in script file):
  trap ‘clean_scratch’ TERM EXIT

  ...
  cp results /storage/… || export CLEAN_SCRATCH=false
  • if the central storage is not available (the results could not be copied out), the data will remain in the particular scratch directory
    ◦ user is informed about (non-)useful scratch cleanup
    ◦ user is informed about scratches not correctly cleaned after previous computations
Central storages overused

Central (working) storages are not infinitely large 😞

/storage/<CITY>

Clean/move currently unnecessary data

- *possibilities*:
  - delete unnecessary data
  - move actually unnecessary data to archival storages

see [https://wiki.metacentrum.cz/wiki/Archival_data_handling](https://wiki.metacentrum.cz/wiki/Archival_data_handling)
Huge jobs' outputs and data in `/tmp`

Computing nodes restrict the amount of data a user is able to store to local disks (outside the scratch space) = 1 GB quota

- influences the `/tmp` directory (temporal files)
- influences the huge job's outputs (stdout, stderr)

**Store bigger amounts of data to scratches**

- forwarding the temporal directory
  
  many applications follow the system variable `TMPDIR`
  
  • **usage:** `export TMPDIR=$SCRATCHDIR`

- forwarding the stdout/stderr of an application
  
  • `myapp ... 1>$SCRATCHDIR/stdout 2>$SCRATCHDIR/stderr`

- checking the status of your local quota and a list of files occupying it (once being informed by email)
  
  • **utility** `$ check-local-quota` has to be run on the particular node (with exhausted local quota)
Non-effective computations

Be aware of the resource usage effectivity of your jobs

– a request for multiple CPUs/cores will not make a single-processor (single-thread) computation parallel (= it won’t be faster)
  • just a single CPU will be used
– many applications significantly vary between the number of CPUs used throughout a computation
  • higher number of CPUs might be used just for a short time of the computation

Observing the computation usage of (not only) CPUs:

– during a computation:
  • log-in to the computation node (SSH) and use standard Linux tools (top, htop, …)
– after a computation:
  • see the list of jobs at the MetaCentrum portal (https://metavo.metacentrum.cz/cs/myaccount/myjobs.html)
    the non-effective jobs have red background color
Infiniband

Distributed jobs might run ineffectively because of slow communication channel

– the inter-process communication using standard network services (Ethernet) is slow
– Infiniband – specialized low-latency interconnect for fast inter-process communication in distributed computations

Most of our clusters are equipped with Infiniband

– considerably accelerates the performance of distributed (MPI) computations
  • the Infiniband availability is automatically detected
  computations started always in the same way: mpirun myapp
  • if the Infiniband is not available, Ethernet is used as a fallback

– request:
  • $ qsub –l select=… –l place=group=infiniband script.sh
Many short-term jobs processing

Group/gather short-term jobs

- e.g., the ones running less than few minutes
  • startup overhead may be a significant part of the whole processing time
  results in wasting resources

Run more computations within a single job

- possibilities:
  • serial computations run inside a single job
    process data1
    process data2
    ...
  • parallel computations run inside a single job (necessary to ask for enough CPUs)
    o pbsdsh
    o parallel
Computations on frontends

Do not run computations on frontends

– neither for computations nor for complex results analyses
  • increased frontend load results in limitation of its services (and usually frontend crash)
– frontend’s primary job is jobs’ preparation and very simple and short-term computations

Use interactive jobs

– request:
  • $ qsub -I -l select=...
– usage possibilities:
  • textual mode
  • graphical mode – VNC access

$ module add gui
$ gui start
  ○ see https://wiki.metacentrum.cz/wiki/Remote_desktop
Interactive jobs

Minimize the time lags in interactive jobs

– especially the time between job startup and your work (starting computations)
  • -> in the time lag, the resources are wasted

Stay informed about your job’s startup

– request:
  • $ qsub -m ab -I -l select=...
    will send you an email once the job begins
    o ("-m abe" also in the case of job’s finish)

– these options could be also used in batch jobs
  but be aware of running too many jobs with this option set!
  • overloads your mailbox
  • may blacklist our mailservers at external mail providers 😊
Cloud nodes

Be aware about your VMs running

– even the unused VMs/nodes (but running) consume infrastructure resources
  • -> results in wasting resources, which somebody could use

Terminate/Suspend unused VMs

– we‘ll regularly inform you about your VMs running in case of no response (= time extension), the VMs are terminated
How to deal with parallel/distributed computations?
Parallel jobs (OpenMP):

- if your application is able to use multiple threads via a shared memory, ask for a single node with multiple processors
  $ qsub -l select=1:ncpus=...

- make sure, that before running your application, the OMP_NUM_THREADS environment variable is appropriately set
  - otherwise, your application will use all the cores available on the node
    - → and influence other jobs…
  - usually, setting it to NCPUs is OK
  $ export OMP_NUM_THREADS=$PBS_NUM_PPN
How to ... run a parallel/distributed computation II.

Distributed jobs (MPI):

- if your application consists of multiple processes communicating via a message passing interface, **ask for a set of nodes** (with arbitrary number of processors)
  
  ```
  $ qsub -l select=...:ncpus=...
  ```

- **make sure**, that before running your application, the appropriate **openmpi/mpich2/mpich3/lam** module is loaded into the environment
  
  ```
  $ module add openmpi
  ```

  then, you can use the **mpirun/mpiexec** routines
  
  ```
  $ mpirun myMPIapp
  ```

- it’s **not necessary** to provide these routines neither with the number of nodes to use (“-np” option) nor with the nodes itself (“--hostfile” option)

  - the computing nodes are **automatically detected** by the openmpi/mpich/lam
How to … run a parallel/distributed computation III.

Distributed jobs (MPI): accelerating their speed I.

- to accelerate the speed of MPI computations, ask just for the nodes interconnected by a **low-latency Infiniband interconnection**
  - all the nodes of a cluster are interconnected by Infiniband
  - there are several clusters having an Infiniband interconnection
    - mandos, minos, hildor, skirit, tarkil, nympha, gram, luna, manwe (MetaCentrum)
    - zewura, zegox, zigur, zapat (CERIT-SC)

- **submission example:**
  
  $ qsub -l select=4:ncpus=2 -l place=group=infiniband MPIscript.sh

- **starting an MPI computation using an Infiniband interconnection:**
  - in a common way: $ mpirun myMPIapp
    - the Infiniband will be automatically detected
  - is the Infiniband available for a job? **check using** $ check-IB
Questions and Answers:

- *Is it possible to simultaneously use both OpenMP and MPI?*
  - Yes, it is. But be sure, how many processors your job is using

  - Appropriately set the “-np” option (MPI) and the OMP_NUM_THREADS variable (OpenMP)

  - **OpenMPI**: a single process on each machine (`mpirun -pernode ...`) being threaded based on the number of processors (`export OMP_NUM_THREADS=$PBS_NUM_PPN`)

- Any other questions?
Other computing possibilities

Cloud computing – MetaCloud
Grid vs. Cloud computing

**Grid computing** suitable for:
- long-term and/or large-scale computations
  - (primarily batch processing)
- applications not requiring special OSs (features)
  - pre-installed or users’ ones

**Cloud computing** suitable for:
- applications requiring special environment (OS) and/or features
  - one can run various operating systems (incl. Windows OS) and/or application equipment
  - administrator/root access is provided
Cloud computing

How to compute?

• **additional registration to MetaCloud** group required
  • your SSH key is needed to access the VMs
  • [https://perun.metacentrum.cz/fed/registrar/?vo=meta&group=metacloud](https://perun.metacentrum.cz/fed/registrar/?vo=meta&group=metacloud)

• OpenNebula GUI for deployed nodes management
  • [https://cloud.metacentrum.cz/](https://cloud.metacentrum.cz/)

• interact via:
  • cloud/VM console
  • internal OS services (SSH, VNC, Rdesktop, …)
Cloud computing

Virtual Machines

<table>
<thead>
<tr>
<th>ID</th>
<th>Owner</th>
<th>Group</th>
<th>Name</th>
<th>Status</th>
<th>Host</th>
<th>IPs</th>
<th>VNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>21254</td>
<td>xvelan</td>
<td>kypo</td>
<td>PB173-2015</td>
<td>RUNNING</td>
<td>hda46.privl.certi-sc.cz</td>
<td>147.251.253.168</td>
<td></td>
</tr>
<tr>
<td>21241</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO-player</td>
<td>RUNNING</td>
<td>hda12.privl.certi-sc.cz</td>
<td>10.19.3.66</td>
<td></td>
</tr>
<tr>
<td>21240</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO-eurovis-DB-server</td>
<td>RUNNING</td>
<td>hda12.privl.certi-sc.cz</td>
<td>10.19.3.110</td>
<td></td>
</tr>
<tr>
<td>21239</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO-general-user</td>
<td>RUNNING</td>
<td>hda12.privl.certi-sc.cz</td>
<td>10.19.3.192</td>
<td></td>
</tr>
<tr>
<td>21238</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO-eurovis-web-server</td>
<td>RUNNING</td>
<td>hda12.privl.certi-sc.cz</td>
<td>10.19.3.104</td>
<td></td>
</tr>
<tr>
<td>21237</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO-general-mobile-device</td>
<td>RUNNING</td>
<td>hda12.privl.certi-sc.cz</td>
<td>10.19.3.48</td>
<td></td>
</tr>
<tr>
<td>21236</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO-general-web-server</td>
<td>RUNNING</td>
<td>hda12.privl.certi-sc.cz</td>
<td>10.19.3.84</td>
<td></td>
</tr>
<tr>
<td>21234</td>
<td>kypo-on</td>
<td>kypo</td>
<td>EUROVIS-SCENARIO</td>
<td>RUNNING</td>
<td>hda46.privl.certi-sc.cz</td>
<td>10.18.2.183</td>
<td></td>
</tr>
<tr>
<td>21190</td>
<td>kypo-on</td>
<td>kypo</td>
<td>lvo-analysis-test2</td>
<td>RUNNING</td>
<td>hda41.privl.certi-sc.cz</td>
<td>10.19.3.96</td>
<td></td>
</tr>
<tr>
<td>21189</td>
<td>kypo-on</td>
<td>kypo</td>
<td>lvo-analysis-test1</td>
<td>RUNNING</td>
<td>hda41.privl.certi-sc.cz</td>
<td>10.19.3.60</td>
<td></td>
</tr>
</tbody>
</table>
Basic terminology

- **template** – a description of the node ("computer") you are asking from the cloud
  - specificies requested CPUs, memory, disk storage, network interfaces, etc.
  - predefined ones & own ones

- **image** – a storage space ("HDD")
  - equipped with an OS or not
  - *persistent & non-persistent* (default; data are lost when destroying the VM)

- **network security group** – a set of firewall rules
  - restricts the access to the VM

- **instance** – a running node/computer created based on some template receipt

- **VM console** – a VNC connection to the VM ("computer screen")
Common usage

Common operations with MetaCloud:

- see video tutorial
- for advanced use, see MetaCloud documentation
  - e.g., creating your own template (duplicate existing one) or disk image

**Warning:** Please, be aware of the VMs you are running. And if not used, suspend or terminate them...

- every 3 months, we’ll recommend you your running VMs
  - if not explicitly renewed/extended in the defined time period, **the VMs will be terminated**
Other computing possibilities

Hadoop computing
Hadoop: an open-source framework for distributed storage and distributed processing of large volumes of data
- large data blocks splitted and distributed amongst nodes
- a MapReduce-based algorithm (= data processing code) is distributed over the distributed blocks and processed in parallel

Suitable for:
- huge datasets to be processed
- but NOT suitable for arbitrary data processing one can imagine
- just for the processing meeting the MapReduce programming model
  - e.g., counting the number of times words occur in a corpus

Other computing possibilities

Specialized frontends – Galaxy, Chipster, …
Specialized frontends/environments

Suitable for:
• user communities with well-defined processing needs
  • workload & computing pipeline orchestrators available via GUI
  • usually adapted to user needs and/or because of interoperability with our infrastructure
    • in background, the proper computing method is used (grid, cloud, etc.)

How to compute?
• **Galaxy** (docs [https://wiki.metacentrum.cz/wiki/Galaxy](https://wiki.metacentrum.cz/wiki/Galaxy))
  • *ELIXIR RepeatExplorer Galaxy* – [https://repeatexplorer-elixir.cerit-sc.cz](https://repeatexplorer-elixir.cerit-sc.cz)
  • *MetaCentrum Galaxy* – [https://galaxy.metacentrum.cz](https://galaxy.metacentrum.cz)
• **Chipster** (docs [https://wiki.metacentrum.cz/wiki/Chipster](https://wiki.metacentrum.cz/wiki/Chipster))
• basic details are provided in the leaflet