

# RTU klastera lietotāja rokasgrāmata

*v.1.2*

## Satura rādītājs

1.	Kā saņemt piekļuvi klasterim? .....	3
2.	Vispārīgas norādes klastera lietošanai .....	3
3.	Klastera tehniskais apraksts .....	3
3.1.	Parametri īsumā .....	3
3.2.	Shēma .....	4
3.3.	Aparatūra .....	4
3.4.	Mezglu saraksts ( <i>hostnames</i> ) .....	6
4.	Klasterī instalētā lietojumprogrammatūra .....	8
5.	Klastera lietotāju saskarne (interfeiss) .....	8
5.1.	Komandrinda .....	9
5.2.	Lietojumprogrammatūras grafiskā saskarne .....	9
5.3.	Web saskarne .....	9
6.	Pieslēgšanās klasterim .....	10
6.1.	RTU klastera piekļuves parametri .....	10
6.2.	Rīki attālai piekļuvei .....	10
6.2.1.	Komandrindas piekļuve (SSH) .....	10
7.	Darbs ar klasteri .....	11
7.1.	Lietotāja darba apgabals .....	11
7.2.	Vides sagatavošana (moduļi) .....	12
7.3.	Kur izpildās uzdevums? .....	12
7.4.	Uzdevuma ievietošana rindā .....	13
7.4.1.	Vienkāršs uzdevums .....	13
7.4.2.	Interaktīvs uzdevums .....	13
7.4.3.	Uzdevuma parametri un prasības .....	14
7.4.4.	Paralēls (MPI) uzdevums .....	15
7.4.5.	Uzdevuma mainīgie .....	15
7.4.6.	Uzdevuma pārtraukšana .....	16
7.5.	Rindas, mezglu un uzdevumu monitorings .....	16
7.6.	Uzdevuma izpildes efektivitāte .....	16
8.	Uzdevumu rindas .....	17
9.	Failu kopēšana .....	18
10.	HPC lietojuma uzskaitē .....	19
11.	Grafiskie rīki .....	19
11.1.	Piemērs: izmantot <i>MATLAB</i> pakotni grafiskā režīmā .....	20
12.	Noderīgas saites .....	20

## 1. Kā saņemt piekļuvi klasterim?

Lai saņemtu piekļuvi klasterim (superdatoram), jāaizpilda [pieteikums](#). Parasti ar jauniem lietotājiem tiekamies klātienē, lai pārrunātu lietotāja vajadzības un veiktu īsu instruktāžu darbam ar klasteri.

## 2. Vispārīgas norādes klastera lietošanai

HPC klasteris lietotājam pieejams nepārtraukti, bet, tā kā vienlaicīgi var strādāt vairāki lietotāji, darbojas uzdevumu rindas princips. Tas nozīmē – ja uzdevumam pieprasīto resursu apjoms pārsniedz klastera brīvo resursu (procesoru kodolu vai atmiņas) apjomu, uzdevums tiek ielikts rindā, un gaidīšanas laiks ir atkarīgs no uzdevumu rindas garuma un lietotāja prasīto resursu apjoma.

Lai sabalansētu HPC klastera noslodzi starp lietotājiem, administratori var mainīt lietotājam pieejamo klastera resursu (piemēram, procesoru kodolu) apjomu vai vienlaicīgi izpildāmo uzdevumu skaitu.

Paralēliem (MPI) uzdevumiem novērtējiet ātrdarbību, izmantojot dažādu mezglu/kodolu skaitu, lai noteiktu piemērotāko jūsu uzdevumam. Ne vienmēr lielāks kodolu skaits nodrošinās ātrāku uzdevuma izpildi.

Centieties norādīt uzdevumam nepieciešamo laiku (*walltime*), kas ļaus arī citiem lietotājiem prognozēt rindā gaidīšanas laiku.

Ja uzdevuma izpilde prasa intensīvu darbu ar disku (I/O), iespēju robežās neizmantojot tīkla direktoriju (/home/\*), bet lokālu SSD disku. Lokāls disks ir piemontēts direktorijai /scratch. Pēc uzdevuma izpildes, lūdzu, iztīrīt šajā direktorijā izveidotos failus.

Iespēju robežās neizpildīt uzdevumus lokāli piekļuves servera (*ui-1.hpc.rtu.lv*). To izmantot tikai uzdevumu ievietošanai rindā, programmu kompilēšanai un īsu uzdevumu testēšanai.

Lietotājs nepieļauj darbības, kas var traucēt citu lietotāju darbu (uzdevumu rindas apiešana, rezervētā resursu apjoma pārsniegšana u. tml.).

## 3. Klastera tehniskais apraksts

Klasteris sastāv no viena galvenā mezgla, kas pilda uzdevumu pārvaldības funkciju, datu glabātuves, piekļuves servera un 56 skaitļošanas mezgliem, kas nodrošina lietotāju uzdevumu izpildi.

### 3.1. Parametri īsumā

Kodolu skaits: 1200

Kopējā RAM: 8 TB

RAM uz vienu procesu: līdz 1.5 TB

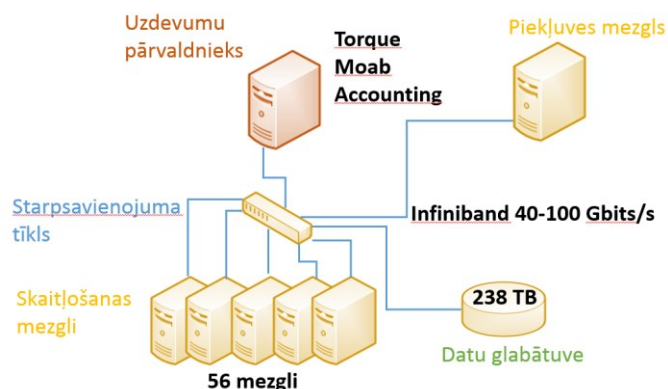
GPU paātrinātāju skaits: 22

CUDA kodolu skaits: 66752

Datu glabāšanas apjoms: 238 TB

Kopējā veiktspēja: 139 Tflops

## 3.2. Shēma



## 3.3. Aparatūra

### Galvenais mezgls – uzdevumu pārvaldnieks

- Dell PowerEdge R630
- CPU: 2 x Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz (kopā 12 kodoli)
- RAM: 32 GB DDR4 2133MHz ECC
- 2.4 TB SAS 10000 rpm HDD

### Piekļuves (login) mezgls

- Lenovo x3650 M5
- CPU: 2 x Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz (kopā 20 kodoli)
- RAM: 64 GB DDR4 1600MHz ECC

### Skaitļošanas mezgli (vasara)

- Kopējie parametri
  - 13 mezgli
  - 494 CPU kodoli
  - 8 GPU (40960 CUDA kodoli)
  - 44 Tflops (x86) + 62 Tflops (GPU) = 106 Tflops
- 10 mezgli Dell EMC PowerEdge R640
  - CPU: 2 x Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz (kopā 36 kodoli)
  - RAM: 384 GB DDR4 2666 MHz ECC
  - 240 GB SSD
- 2 mezgli Dell EMC PowerEdge C4140
  - CPU: 2 x Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz (kopā 32 kodoli)
  - RAM: 192 GB DDR4 2666 MHz ECC
  - GPU: 4 x NVIDIA Tesla V100, NVLink, 16 GB HBM2, 5120 CUDA kodoli
  - 240 GB SSD
- 1 mezgls Dell EMC PowerEdge R940
  - CPU: 4 x Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz (kopā 72 kodoli)
  - RAM: 1.5 TB DDR4 2666 MHz ECC
  - 400 GB SSD
- Mezglu starpsavienojums
  - Infiniband EDR 100 Gbit/s
  - 1 GigE

## Skaitļošanas mezgli (rudens)

- Kopējie parametri
  - 16 mezgli
  - 384 CPU kodoli
  - 8 GPU (23104 CUDA kodoli)
  - 15.4 Tflops (x86) + 11.4 Tflops (GPU) = 26.8 Tflops
- 12 mezgli Dell PowerEdge R630
  - CPU: 2 x Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (kopā 24 kodoli)
  - RAM: 128 GB DDR4 2133MHz ECC
  - 200 GB SSD
- 4 mezgli Dell PowerEdge R730
  - CPU: 2 x Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (kopā 24 kodoli)
  - RAM: 128 GB DDR4 2133MHz ECC
  - GPU: 2 x NVIDIA Tesla K40, 12GB GDDR5, 2888 CUDA kodoli
  - 200 GB SSD
- Mezglu starpsavienojums
  - Infiniband FDR 56 Gbit/s
  - 1 GigE

## Skaitļošanas mezgli (tb2)

- Kopējie parametri
  - 27 mezgli
  - 312 CPU kodoli
  - 6 GPU (2688 CUDA kodoli)
  - 3 Tflops (x86) + 3.1 Tflops (GPU) = 6.1 Tflops
- 24 mezgli *T-Platforms T-Blade2*
  - CPU: 2 x Intel(R) Xeon(R) CPU X5670 @ 2.93GHz (kopā 12 kodoli)
  - RAM: 12 GB DDR3 1333 MHz ECC
- 3 mezgli *T-Platforms T-Blade2 GPU*
  - CPU: 2 x Intel(R) Xeon(R) CPU E5630 @ 2.53GHz (kopā 8 kodoli)
  - RAM: 24 GB DDR3 1066 MHz ECC
  - GPU: 2 x NVIDIA Tesla M2070, 6 GB GDDR5, 448 CUDA kodoli
- Mezglu starpsavienojums
  - Infiniband QDR 40 Gbit/s
  - 1 GigE

## Datu glabātuve

- 8 mezgli EMC Isilon x200
- Distributīva failu sist, klientu pieslēgums ar NFS protokolu
- 40 Gbit/s InfiniBand tīkla savienojums ar klasteri
- Lietotājiem pieejamā vieta: 238 TB

## Klastera pārvaldības programmatūra

- Operētājsistēma: Centos 6.9/Centos 7.5
- Operētājsistēmu attēlu (*image*) nodrošinājums: xCAT
- Uzdevumu pārvaldība: Torque 6.1.1.1/ Moab 9.1.1
- Lietojuma uzskaitē: Moab Accounting Manager

### 3.4. Mezglu saraksts (*hostnames*)

Servisa mezgli
Mezgli – standarta
Mezgli – 4 CPU, ar palielinātu atmiņu
Mezgli – ar GPU

Mezglā vārds	Tips	OS	Fiziskā atmiņa	CPU modelis	Kodolu skaits	GPU skaits	Lokāls disks	Iezīmes ( <i>node features</i> )
<b>rudens</b>	cluser head	Centos 6	32 GB	E5-2620 v3				
<b>ui-1</b>	login	Centos 7	64 GB	E5-2650 v3	20	0	/scratch	
<b>wn01</b>	compute-smp	Centos 7	1.48 TB	Gold 6140	72	0	/scratch	dell vasara smp highmem centos7
<b>wn02</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn03</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn04</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn05</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn06</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn07</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn08</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn09</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn10</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn11</b>	compute	Centos 7	384 GB	Gold 6154	36	0	/scratch	dell vasara centos7
<b>wn12</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn13</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn14</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn15</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn16</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn17</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn18</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn19</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn20</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn21</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn22</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn23</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn24</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn25</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
<b>wn26</b>	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6

wn27	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn28	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn29	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn30	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn31	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn32	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn33	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn34	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn35	compute-gpu	Centos 7	24 GB	E5630	8	2	-	tb2-8 gpu centos7
wn37	compute-gpu	Centos 7	24 GB	E5630	8	2	-	tb2-8 gpu centos7
wn39	compute-gpu	Centos 7	24 GB	E5630	8	2	-	tb2-8 gpu centos7
wn42	compute	Centos 6	12 GB	X5670	12	0	-	tb2 inf centos6
wn43	compute-gpu	Centos 7	128 GB	E5-2680 v3	24	2	/scratch	dell rudens gpu k40 centos7
wn44	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn45	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn46	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn47	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn48	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn49	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn50	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn51	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn52	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn53	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn54	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn55	compute	Centos 7	128 GB	E5-2680 v3	24	0	/scratch	dell rudens centos7
wn56	compute-gpu	Centos 7	128 GB	E5-2680 v3	24	2	/scratch	dell rudens gpu k40 centos7
wn57	compute-gpu	Centos 7	128 GB	E5-2680 v3	24	2	/scratch	dell rudens gpu k40 centos7
wn58	compute-gpu	Centos 7	128 GB	E5-2680 v3	24	2	/scratch	dell rudens gpu k40 centos7
wn59	compute-gpu	Centos 7	192 GB	Gold 6130	32	4	/scratch	dell vasara gpu v100 centos7
wn60	compute-gpu	Centos 7	192 GB	Gold 6130	32	4	/scratch	dell vasara gpu v100 centos7
			<b>8228 GB</b>		<b>1212</b>	<b>22</b>		

## 4. Klasterī instalētā lietojumprogrammatūra

Klasterī instalētas šāda atvērtā koda lietojumprogrammatūra:

- *bcl2fastq (ver. 1.8.4)*
- *Blender (ver. 2.7)*
- *GROMACS (ver. 2018-2)*
- *Hashcat Network-simulator NS-2 (ver. 2.34)*
- *OpenFoam (ver. 2.4, 5)*
- *Peridigm (ver. 1.5)*
- *R (ver. 3.4.0)*
- *SalomeMeca (ver. 2015)*
- *Scilab (ver. 5.4)*

Komerčiālo izstrādātāju lietojumprogrammatūra:

- *Agisoft PhotoScan*
- *Comsol Multiphysics (ver. 5.3a)*
- *MathWorks MATLAB (ver. 2017b)*

Programmēšanas rīki un kompilatori:

- *Conda/Anaconda (ver. 4.4)*
- *CUDA Toolkit (ver. 8.0, 9.2)*
- *GCC (ver. 4.8.5, 7.3.0)*
- *Intel Parallel Studio (ver. 2015)/ IntelMPI (ver. 4.0.3)*
- *OpenMPI (ver. 3.1.1)*
- *Perl (ver. 5)*
- *Python (ver. 2.7, 3.6)*
- *R (ver. 3.4.0)*

Pēc lietotāja lūguma var tikt instalēta papildus lietojumprogrammatūra vai esošās cita versija. Lietotāji drīkst arī paši instalēt/kompilēt programmatūru savā lietotāja apgabalā, ja instalācijas process neprasa administratora (*root*) tiesības.

Lietojumprogrammu izmantošanas piemērus meklējiet klastera direktorijā: [/opt/exp\\_soft/user\\_info](/opt/exp_soft/user_info)

Atsevišķām programmām instrukcijas arī HPC centra mājaslapā: <http://hpc.rtu.lv/hpc-programmatura/>

## 5. Klastera lietotāju saskarne (interfeiss)

Lietotāju mijiedarbe ar klasteri var notikt vairākos veidos:

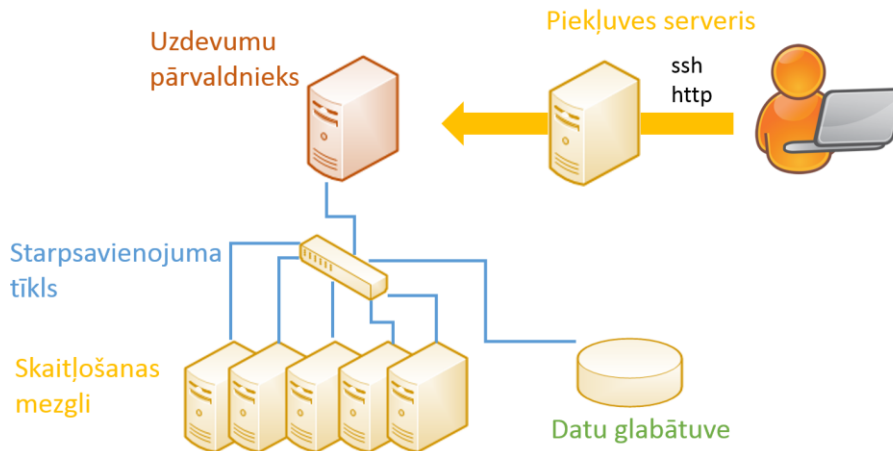
- no komandrindas (attāls terminālis)
- no lietojumprogrammas grafiskā saskarne (GUI)
- no *web* saskarnes





## 6. Pieslēgšanās klasterim

Darbs ar klasteri notiek, izmantojot piekļuves serveri (*login node*), kurā ir instalēta *Linux* vide ar speciāliem klastera klienta rīkiem.



### 6.1. RTU klastera piekļuves parametri

Autentifikācijai tiek izmantots **lietotājvārds un parole**, kuru saņēmāt reģistrējoties klastera lietošanai.

Servera adrese: 85.254.226.76 (ui-1.hpc.rtu.lv)

Protokols: SSH

Ports:22

Piekļuve tiek nodrošināta no jebkuras IP adreses.

### 6.2. Rīki attālai piekļuvei

Klasterim var pieslēgties šādos veidos:

- ar SSH protokolu - attāls terminālis/komandrinda
- 

#### 6.2.1. Komandrindas piekļuve (SSH)

Piekļuves servera komandrindai pieslēdzas attāli no lietotāja datora, izmantojot *SSH* protokolu. Tam var izmantot šādu rīkus:

**Linux un MacOS** vidē.

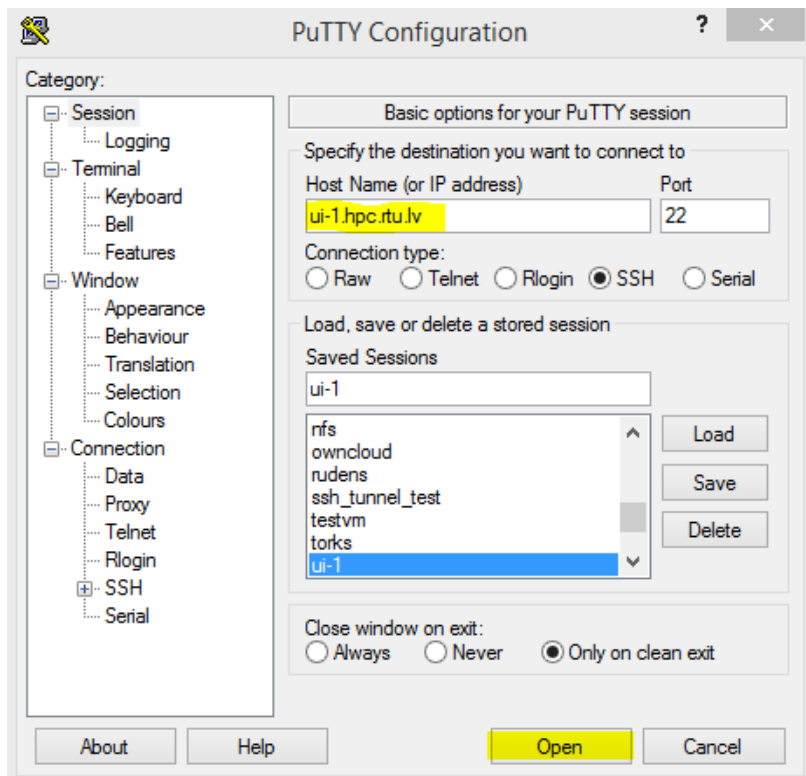
1. Atveriet komandrindu (*terminal*)
2. Izmantojiet SSH klienta programmu, izpildot:

```
ssh -p [port] [username]@[hostname]
```

**Piemērs:**

```
ssh -p 22 username@ui-1.hpc.rtu.lv
```

**MS Windows** vidē izmantojiet SSH klienta programmu *PuTTY*. Lejupielādējiet no: <http://www.putty.org/>



**Komentārs.** No komandrindas var izsaukt arī grafiskus rīkus/logus (vairāk skat. 11. sadaļā).

## 7. Darbs ar klasteri

Darbs ar klasteri iedalāms 4 etapos.

1. Pieslēgšanās klasterim
2. Uzdevuma (skripta) ievietošana rindā
3. Uzdevums izpilde un sekošanai tai.
4. Rezultātu saņemšana

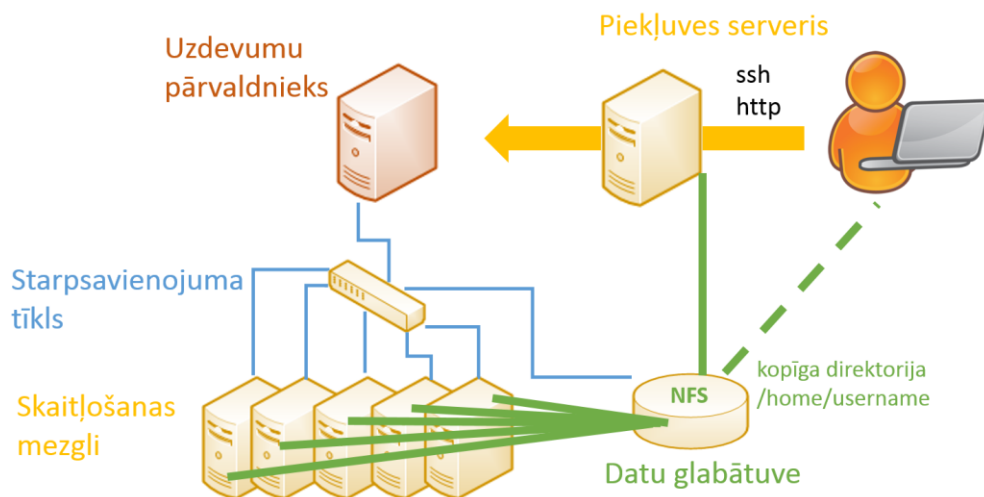
Uzdevums – no angļu *job/task*. Simulācija vai programmas kods, kuru lietotājs izpilda klasterī.

### 7.1. Lietotāja darba apgabals

Katram lietotājam ir izveidots darba apgabals, kur var glabāt ar uzdevumiem saistītos failus. Pieslēdzoties piekļuves servera komandrindai, automātiski nonāksiet savā darba direktorijā:

```
/home/username
```

Failu sistēma starp klastera mezgliem un piekļuves serveri ir koplietota (NFS – *Network File System*), tāpēc nav nepieciešama failu kopēšana uz un no izpildes mezgla un starp mezgliem. Tas nozīmē - ja iekopēsiet failu piekļuves serverī, tas būs pieejams arī visos citos mezglos. Tāpat – visas izmaiņas, ko veiksiet failiem piekļuves serverī, atainosies arī mezglos.



Savā darba apgabalā lietotājs var arī instalēt/kompilēt klasterī izmantojamu lietojumprogrammatūru, ja instalācijas process neprasa administratora (*root*) tiesības.

## 7.2. Vides sagatavošana (moduļi)

Lai sagatavotu klastera vidi dažādu lietojumprogrammu, kompilatoru, bibliotēku izmantošanai, lietotāji var izmantot moduļus (*Environment Modules*). Moduli ielādē ar komandu:

```
module load module_name
```

Komanda uzstāda ceļu (PATH) uz attiecīgās programmatūras direktoriju un citus nepieciešamos vides mainīgos. Lai iegūtu visu moduļu (lietojumprogrammatūru un rīku) sarakstu, izpildiet:

```
module avail
```

```
[ciko@ui-1 ~]$ module avail
----- /usr/share/Modules/modulefiles -----
dot      module-git  module-info  modules      null         use.own
----- /opt/exp_soft/env_modules/ -----
R/R-3.4   conda         intel/intel-2013  ns/ns-2.34   openmpi
bcl2fastq/bcl2fastq-1.8.4  cuda/cuda-7.5    matlab/R2015a  openfoam/openfoam-2.4-centos6  peridigm/1.4.1
blender/blender-2.70      cuda/cuda-8.0    matlab/R2015b  openfoam/openfoam-2.4-centos7  peridigm/1.5
boost/boost-1.6.1         cuda/cuda-9.2    matlab/R2016a  openfoam/openfoam-5.0          salomemeca/salomemeca-2015
comsol/5.1                 glibc-2.17-centos6  matlab/R2017b  openfoam/openfoam-5.0-intelmpi  scilab/scilab-5.4-centos6
comsol/5.2a                gromacs/gromacs-2016.1  mpi/intelmpi   openfoam/openfoam-5.0-sysmpi
comsol/5.3a                hashcat/hashcat-cuda-2.01  mpi/openmpi-1.10.5a1  openfoam/openfoam-default
----- /opt/exp_soft/ohpc/pub/modulefiles -----
EasyBuild/3.6.1_autotools  gnu7/7.3.0       hwloc/1.11.10  prun/1.2
```

*Linux* distributīvā iekļautajiem rīkiem parasti moduļu ielādē nav nepieciešama.

Modulim jābūt ielādētam uz tā mezgla, kur tiek startēta programma. Ja uzdevums tiek izpildīts uz skaitļošanas mezgla/iem, tad modulis jāielādē arī katrā no tiem.

Lai atiestatītu vidi:

```
module unload module_name
```

Vairāk par moduļu lietošanu: <http://modules.sourceforge.net/>

## 7.3. Kur izpildās uzdevums?

Darbā ar klasteri parasti iesaistīti 3 komponenti:

- lietotāja dators
- klastera piekļuves serveris (ui-1.hpc.rtu.lv)
- skaitļošanas mezgls (piem., wn02)

Pieslēdzoties klastera piekļuves serverim un izpildot komandu termināli (arī atverot lietojumprogrammas GUI), vēl nenodrošina, ka uzdevums automātiski izpildīsies uz skaitļošanas mezgliem. Visdrīzāk tas noslogos piekļuves serveri.

## 7.4. Uzdevuma ievietošana rindā

Pirms uzdevums nonāk skaitļošanas mezglā un sāk izpildīties, tas tiek ievietots virtuālā rindā. Rinda organizē resursu sadali daudzlietotāju sistēmā, kur uzdevumu skaits un prasības var pārsniegt brīvo resursu (CPU, atmiņas) apjomu. Atbrīvojoties resursiem, parasti nākamais izpildīsies tas uzdevums, kas rindā gaidījis visilgāk. Lietotājam nav jāseko, kad resursi atbrīvosies – uzdevuma pārvietošanās rindā un izpildes sākšana notiek automātiski. Ja rindas nav (ir brīvi resursi), tad uzdevums sāk izpildīties uzreiz. RTU klasterī ir vairākas rindas, kas atšķiras pēc uzdevumu izpildes ilguma un pieejamā resursu apjoma:

- *batch*
- *fast*
- *long*

Detalizēts rindu apraksts 8. sadaļā.

### 7.4.1. Vienkāršs uzdevums

Uzdevuma ievietošana rindā notiek ar speciāliem *Torque/Moab* klastera klienta rīkiem (izstrādātāja dokumentācija <http://docs.adaptivecomputing.com/torque/6-1-2/adminGuide/torque.htm>).

Komanda vienkārša uzdevuma (skripta) ievietošanai rindā:

```
qsub test.sh
```

`test.sh` ir *bash* valodas (*Linux* komandrindas) skripts, kurā lietotājs ieraksta secīgi izpildāmās komandas, uzdevumam nonākot skaitļošanas mezglā. Tas nodrošina pakešuzdevuma (*batch*) izpildi bez lietotāja līdzdalības. Skripts var saturēt, piemēram, šādu komandu:

```
#!/bin/bash
echo "Hello world from node `./bin/hostname`"
```

Komanda izdrukā skaitļošanas mezgla vārdu. Varat izpildīt to arī lokāli.

Skripta paraugs, lai skaitļošanas mezglā palaistu jūsu programmu ar parametriem:

```
#!/bin/bash
./myprogram 12 3
```

Ja nepieciešams, pirms programmas izpildes ielādējiet attiecīgo moduli (skat. 7.2. sadaļā).

Lietojumprogrammu palaišanas skriptu piemērus un citu noderīgu informāciju atradīsiet direktorijā:

[/opt/exp\\_soft/user\\_info](/opt/exp_soft/user_info)

### 7.4.2. Interaktīvs uzdevums

Alternatīvi pakešuzdevumam var izmantot interaktīvu uzdevumu veidu. Interaktīvais režīms ērts uzdevumu testēšanai un atklūdošanai, kā arī, ja izmantojiet grafiskus rīkus. Sākt interaktīvu uzdevumu:

```
qsub -I
```

Automātiski tiks atvērts attālināts terminālis uz skaitļošanas mezgla, kur varēsiet izpildīt nepieciešamās komandas, rakstot tās komandrindā. Komanda līdzvērtīga "`ssh wn [xx]`" ar atšķirību, ka resursi tiks rezervēti un neradīsies konflikti ar citiem lietotājiem.

Ja interaktīvā režīmā nepieciešams atvērtu grafisku logu, pievienojiet `-X` parametru:

```
qsub -X -I
```

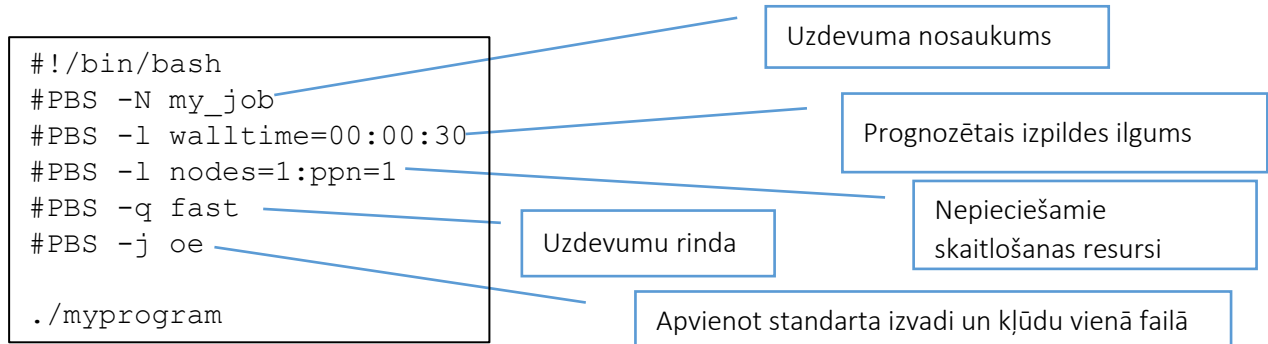
Par grafiskās sesijas pārsūtīšanu vairāk lasiet 11. sadaļā.

### 7.4.3. Uzdevuma parametri un prasības

Jūs variet norādīt uzdevuma parametrus un prasības, piemērām, nosaukumu, kuru rindu izmantot vai cik laika būs nepieciešams uzdevuma izpildei. Šo informāciju rindu sistēma izmantos, lai atrastu uzdevumam piemērotākos resursus.

```
qsub -N my_job -q fast -l walltime=00:00:30 test.sh
```

Uzdevuma prasības var pievienot palaišanas skripta sākumā:



Komentārs. Parametrus var norādīt vienlaicīgi gan komandrindā, gan skriptā, tomēr, dublēšanās gadījumā, komandrindas parametri tiks ņemti vērā ar augstāku prioritāti.

**Kā pieprasīt konkrētus skaitļošanas resursus?** Definējiet prasības ar “`qsub -l`”, kā parametrus norādot:

- kodolu skaitu  
`-l procs=12`
- mezglu un kodolu skaitu katrā mezglā (lietojiet šo pierakstu MPI uzdevumiem!)  
`-l nodes=2:ppn=12`
- konkrētu mezglu/s (var nākties ilgāk gaidīt rindā!)  
`-l nodes=wn44:ppn=12`  
`-l nodes=wn44:ppn=12+wn46:ppn=12`
- grafisko procesoru (GPU) skaitu  
`-l nodes=1:ppn=12:gpus=2`
- nepieciešamo atmiņas apjomu  
`-l nodes=1:ppn=12:pmem=1g` ← atmiņas apjoms uz katru kodolu (procesu)  
`-l nodes=1:ppn=12:mem=12g` ← kopējais uzdevuma atmiņas apjoms
- pieprasīt skaitļošanas mezglus ar noteiktām iezīmēm (*features*). Iezīmes parasti tiek izmantotas klasteros ar dažādiem mezgļiem (nehomogēnos)  
`-l feature=centos7`

Mezglu nosaukumus (*hostnames*), parametrus un iezīmes variet noskaidrot 3.3. sadaļā.

Vairāk par `qsub` komandu un parametriem varat iegūt informāciju, izpildot:

```
man qsub
```

#### 7.4.4. Paralēls (MPI) uzdevums

Uzdevums tiek sadalīts uz vairākiem kodoliem vai klastera mezgliem, izmantojot *Message Passing Interface* (MPI) protokolu.

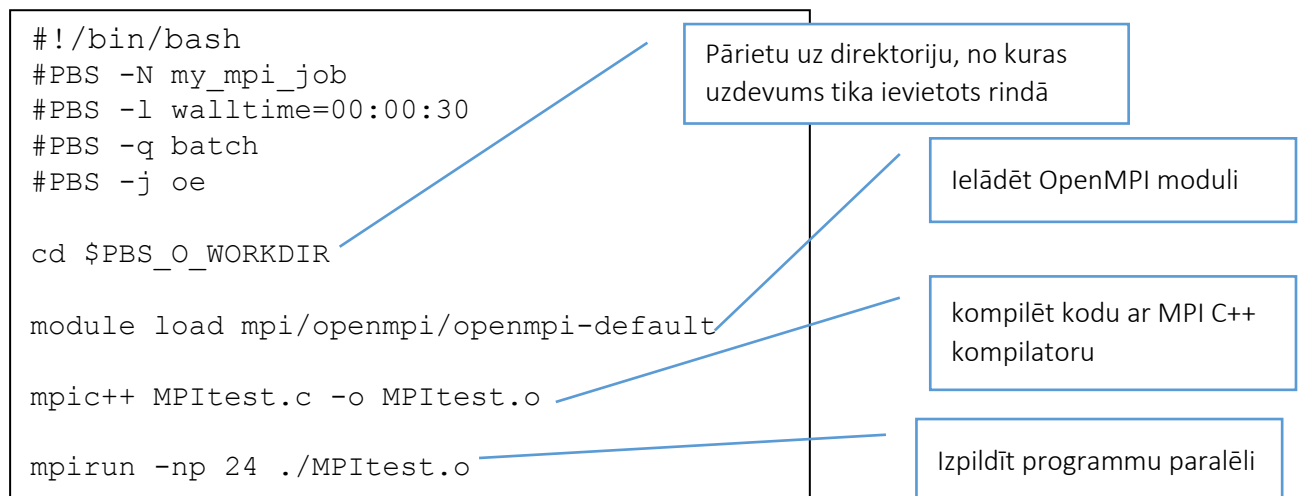
Ievietot rindā paralēlu uzdevumu, kam izpildei nepieciešami 24 kodoli (2 mezgli × 12 kodoli katrā mezglā):

```
qsub -l nodes=2:ppn=12 run_mpi.sh
```

vai nenorādot konkrētu mezglu skaitu

```
qsub -l procs=24 run_mpi.sh
```

run\_mpi.sh skripta piemērs:



MPI piemērus atradīsiet klastera direktorijā: /opt/exp\_soft/users\_info/mpi

#### 7.4.5. Uzdevuma mainīgie

Uzdevuma skriptā parocīgi izmantot mainīgos, kuri uzstādās automātiski uzdevumam nonākot uz skaitļošanas mezgla.

- \$PBS\_O\_WORKDIR
- \$PBS\_NODEFILE
- \$PBS\_GPUFILE
- \$PBS\_NP
- \$PBS\_JOBID

Piemēram, lai pārietu uz direktoriju, no kuras uzdevums tika ievietots rindā:

```
cd $PBS_O_WORKDIR
```

Lai iegūtu sarakstu ar visiem mainīgajiem, sāciet interaktīvu uzdevumu (`qsub -I`) un izpildiet komandu:

```
env | grep PBS
```

## 7.4.6. Uzdevuma pārtraukšana

Izpildē esošu vai rindā gaidošu uzdevumu patrauc ar komandu:

```
qdel [job_id]
```

vai

```
canceljob [job_id]
```

job\_id – uzdevuma identifikators.

Pārtraukt visus lietotāja uzdevumus

```
qdel `all`
```

## 7.5. Rindas, mezglu un uzdevumu monitorings

Komanda, lai pārbaudītu rindā ievietoto uzdevumu izpildes gaitu:

```
qstat
```

R – running, C – completed, Q – queued

Lai redzētu kopējo uzdevumu rindu (par visiem lietotājiem):

```
showq
```

Komanda, lai noskaidrotu pieejamos skaitļošanas resursus:

```
showbf
```

vai detalizētākai informācijai:

```
pbsnodes
```

Lai iegūtu detalizētu informāciju par uzdevuma izpildi, kā arī iemesliem uzdevuma aizturēšanai rindā:

```
checkjob job_id -vvv
```

## 7.6. Uzdevuma izpildes efektivitāte

CPU izmantošanas efektivitāte izpildē esošiem uzdevumiem.

```
showq -r
```

skat. kolonu EFFIC

Pārbaudīt uzdevuma radīto sistēmas noslodzi un efektivitāti skaitļošanas mezglā.

1. Noskaidrot, kurā mezglā uzdevums izpildās

```
qstat -n job_id
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	S	Elap Time
95941.rudens wn01/0	ciko	batch	test.sh	9237	1	1	1gb	00:30:00	R	00:00:05

2. Attālināti pieslēgties attiecīgajam mezglam, piemēram, node10

```
ssh wn01
```



### 3. Izmantot Linux rīkus monitoringam:

```
htop
nvidia-smi
iostat
nfsstat
```

- CPU izmantošanas efektivitātei ar komandu htop

```

1  [|||||100.0%] 4  [|||||100.0%] 7  [|||||99.3%] 10 [|||||99.3%]
2  [|||||100.0%] 5  [|||||99.3%] 8  [|||||100.0%] 11 [|||||100.0%]
3  [|||||100.0%] 6  [|||||100.0%] 9  [|||||99.3%] 12 [|||||100.0%]
Mem|          | 6.64G/62.9G| Tasks: 21, 36 thr: 3 running
Swp|          | 0K/46.6G| Load average: 11.83 6.90 2.92
                                         Uptime: 17:37:15

PID USER      PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command
6812 ciko       23   3 85.5G 407M 169M R 1198 0.6 51:41.36 gmx mdrun -deffnm md_0_1
6835 ciko       20   0 85.5G 407M 169M S 100. 0.6 4:18.16 gmx mdrun -deffnm md_0_1
6834 ciko       20   0 85.5G 407M 169M S 100. 0.6 4:18.24 gmx mdrun -deffnm md_0_1
6831 ciko       20   0 85.5G 407M 169M S 100. 0.6 4:18.14 gmx mdrun -deffnm md_0_1
6836 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.26 gmx mdrun -deffnm md_0_1
6830 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.34 gmx mdrun -deffnm md_0_1
6832 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.28 gmx mdrun -deffnm md_0_1
6833 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.25 gmx mdrun -deffnm md_0_1
6837 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.23 gmx mdrun -deffnm md_0_1
6828 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.28 gmx mdrun -deffnm md_0_1
6829 ciko       20   0 85.5G 407M 169M S 99.7 0.6 4:18.22 gmx mdrun -deffnm md_0_1
6827 ciko       20   0 85.5G 407M 169M R 99.0 0.6 4:19.46 gmx mdrun -deffnm md_0_1

```

- Sekot GPU izmantošanas efektivitātei

```

[ciko@node10 2016.1]$ nvidia-smi
Tue Jul 24 10:24:10 2018

+-----+
| NVIDIA-SMI 396.26                Driver Version: 396.26          |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|    0   Tesla K20m        On       | 00000000:04:00:0 Off  |      67%   E. Process |
| N/A   38C    P0      89W / 225W |  81MiB /  5062MiB |           |
+-----+-----+
|    1   Tesla K20m        On       | 00000000:42:00:0 Off  |      68%   E. Process |
| N/A   38C    P0      92W / 225W |  81MiB /  5062MiB |           |
+-----+-----+

+-----+-----+
| Processes:                        GPU Memory Usage          |
| GPU   PID     Type    Process name      Usage          |
+-----+-----+
|    0   6812    C       gmx                70MiB          |
|    1   6812    C       gmx                70MiB          |
+-----+-----+

```

## 8. Uzdevumu rindas

Klastera lietotājiem pieejams vairākas virtuālas rindas, kas atšķiras pēc uzdevumu izpildes ilguma un pieejamā resursu apjoma. Rindu izvēlas ar “qsub -q” parametru. Nenorādot rindu, uzdevumi tiks ievietoti rindā “batch”.

Pārsniedzot uzdevuma maksimālo pieļaujamo izpildes ilgumu, uzdevums tiks automātiski pārtraukts. Iesakām regulāri saglabāt starprezultātus.

**batch** – noklusētā rinda; piemērota uzdevumiem bez īpašām prasībām (līdz 96 h)

Skaitļošanas mezgli:	wn[02-35,37,39,42-60]
Uzdevuma ilgums (walltime):	12h (maks. 96 h)*
Fiziskā atmiņa uz vienu izmantoto kodolu:	1 GB (maks. 5 GB)
Kodolu skaits/ uzdevums:	1 (maks. bez limita)
Maksimālais izpildošos uzdevumu skaits/lietotājs (running):	bez limita
Maksimālais rindā ievietojamo uzdevumu skaits/lietotājs (queued + running):	400

**fast** – īsiem uzdevumiem (līdz 12 h); ar augstāku uzdevumu izpildes prioritāti; piemērota interaktīviem un testa uzdevumiem

Skaitļošanas mezgli:	wn[02-35,37,39,42-60]
Uzdevuma ilgums (walltime):	1h (maks. 12 h)
Fiziskā atmiņa uz vienu izmantoto kodolu:	2 GB (maks. 10 GB)

Kodolu skaits/ uzdevums:	1 (maks. 48)
Maksimālais izpildošos uzdevumu skaits/lietotājs (running):	1
Maksimālais rindā ievietojamo uzdevumu skaits/lietotājs (queued + running):	2

**long** – rinda laikielipīgiem uzdevumiem (līdz 2 nedēļas)

Skaitļošanas mezgli:	wn[02-35,37,39,42-60]
Uzdevuma ilgums (walltime):	48h (maks. 336 h)
Fiziskā atmiņa uz vienu izmantoto kodolu:	2 GB (maks. 5 GB)
Kodolu skaits/ uzdevums:	1 (maks. 48)
Maksimālais izpildošos uzdevumu skaits/lietotājs (running):	30
Maksimālais rindā ievietojamo uzdevumu skaits/lietotājs (queued + running):	30

**highmem** – piekļuve lielas atmiņas (1.5 TB) mezglam

Skaitļošanas mezgli:	wn01
Uzdevuma ilgums (walltime):	12h (maks. 96 h)
Fiziskā atmiņa uz vienu izmantoto kodolu:	6 GB (maks. 1.5 TB)
Kodolu skaits/ uzdevums:	1 (maks. 72)
Maksimālais izpildošos uzdevumu skaits/lietotājs (running):	bez limita
Maksimālais rindā ievietojamo uzdevumu skaits/lietotājs (queued + running):	20

**NB. highmem rinda pieejama tikai pēc saskaņošanas ar HPC centru!**

**rudens** – tikai mezgli Dell, paralēlu uzdevumu izpilde

Skaitļošanas mezgli:	wn[43-58]
Uzdevuma ilgums (walltime):	24h (maks. 144 h)
Fiziskā atmiņa uz vienu izmantoto kodolu:	2 GB (maks. 10 GB)
Kodolu skaits/ uzdevums:	1 (maks. 384)
Maksimālais izpildošos uzdevumu skaits/lietotājs (running):	bez limita
Maksimālais rindā ievietojamo uzdevumu skaits/lietotājs (queued + running):	200

**inf** – tikai mezgli T-blade (12 kodolu), paralēlu uzdevumu izpilde

Skaitļošanas mezgli:	wn[12-35, 37, 39, 42]
Uzdevuma ilgums (walltime):	24h (maks. 144 h)
Fiziskā atmiņa uz vienu izmantoto kodolu:	1 GB (maks. 2 GB)
Kodolu skaits/ uzdevums:	1 (maks. 256)
Maksimālais izpildošos uzdevumu skaits/lietotājs (running):	bez limita
Maksimālais rindā ievietojamo uzdevumu skaits/lietotājs (queued + running):	200

\* 12h (maks. 96 h) nozīmē, ka 12 h ir noklusētā vērtība, kas tiks iekļauta uzdevuma prasībās, ja lietotājs nenorāda citu; maks. 96 h – maksimālā pieļaujamā vērtība, kuru pārsniedzot, uzdevums dotajā rindā darboties nevarēs.

## 9. Failu kopēšana

Failu kopēšana starp savu datoru un piekļuves serveri.

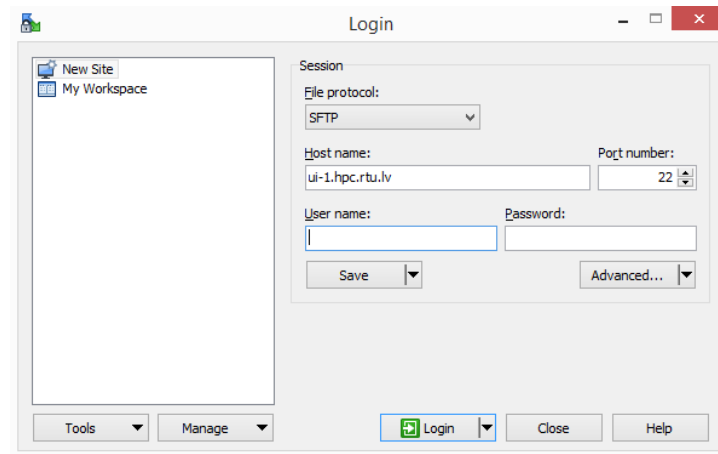
Unix vidē (MacOS, Linux) izmantojiet komandrindas SCP vai kādu grafisko rīku.

```
scp -r my.file username@ui-1.hpc.rtu.lv:
```

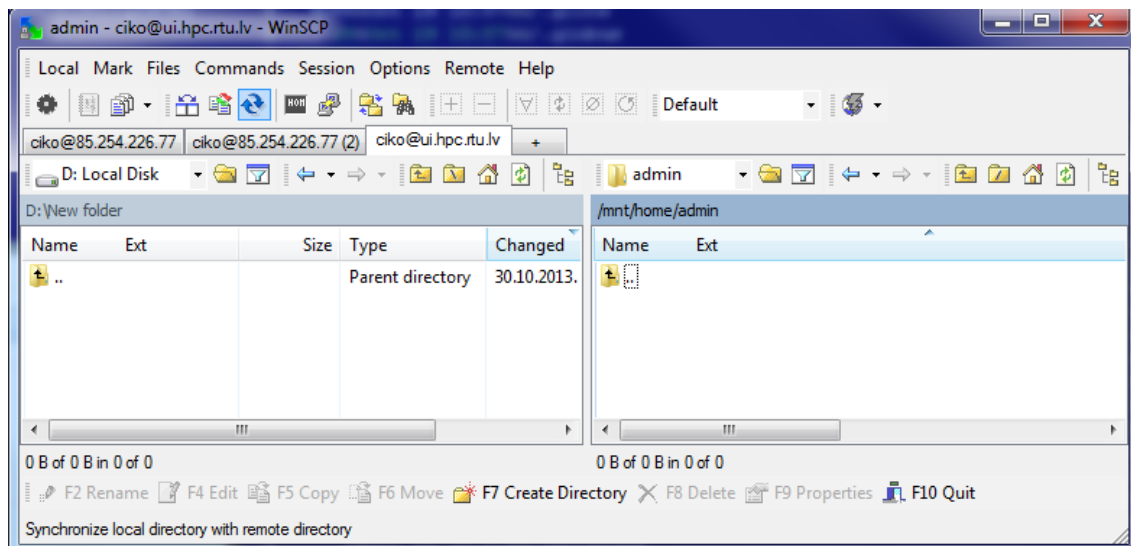
MS Windows lietotāji var izmantot, piemēram, WinScp vai Far failu menedžeri.

Lejupielādējiet WinScp: <https://winscp.net/eng/download.php>.

Pieslēgšana līdzīgi kā ar *PutTY*.



Kreisajā pusē būs redzami Jūsu datora faili, labajā – darba apgabals klastera pieejas serverī.



## 10. HPC lietojuma uzskaitē

Bilanci var apskatīt arī ar komandu:

```
mam-balance
```

```
mam-list-funds
```

legūt detalizētu atskaiti par katra uzdevuma patērētajiem klastera resursiem, piemēram, par 2018. gada jūliju:

```
mam-list-usagerecords -s 2018-07-01 -e 2018-08-01
```

```
[ciko@master ~]$ mam-list-usagerecords -s 2018-07-01 -e 2018-08-01
User Instance Nodes Processors GPUs Duration Memory SubmitTime      StartTime      EndTime
-----
ciko 349694      1          1          31    2018-07-01 07:39:18 2018-07-01 07:39:19 2018-07-01 07:39:50
ciko 349695      1          1          1    2018-07-01 07:39:27 2018-07-01 07:39:50 2018-07-01 07:39:51
ciko 353029      1          1         124    2018-07-23 12:10:47 2018-07-23 12:10:52 2018-07-23 12:12:56
```

## 11. Grafiskie rīki

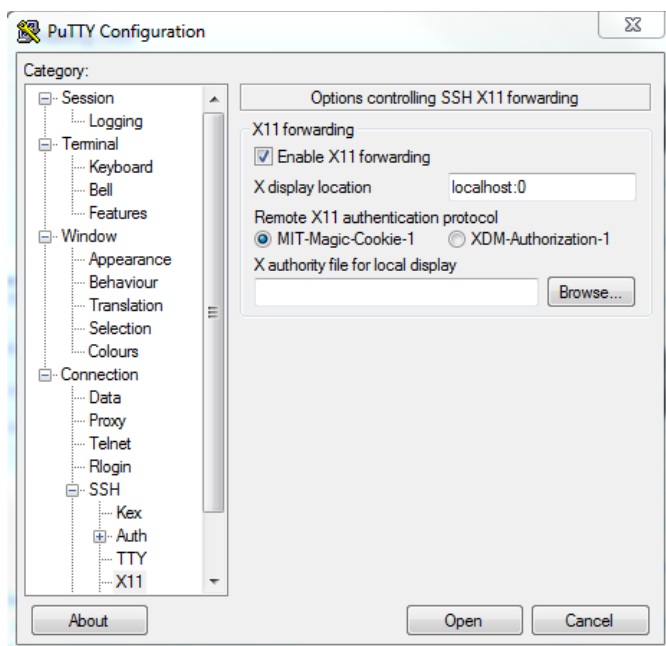
Pieslēdzoties klasterim ar *SSH* klientu (piemēram, *PutTY*), var lietot ne tikai termināla komandrindu, bet arī grafiskos rīkus. *SSH* nodrošina grafiskās sesijas pārsūtīšanu (*X11 forwarding*) no klastera uz lietotāja datoru.

**Linux vidē**, pieslēdzoties klasterim, izmantojiet `-X` parametru

```
ssh -p 22 user@ui-1.hpc.rtu.lv
```

Lai šo funkciju izmantotu MS **Windows vidē**, lietotāja personālajā datorā jāveic papildu sagatavošanās soļi.

1. Instalējiet rīku *Xming X Server for Windows*, kuru var lejupielādēt: <http://sourceforge.net/projects/xming/>.
2. Palaidiet to kā servisu fonā. Pirmajā startēšanas reizē var prasīt mainīt uzstādījumus, atstājiet noklusētos.
3. Papildus nepieciešama *PuTTY* konfigurācija darbam ar *Xming*. Izmainiet saglabāto konfigurāciju priekš klastera piekļuves servera, veicot turpmāk parādītās izmaiņas:



Saglabājiet *PuTTY* konfigurāciju!

### 11.1. Piemērs: izmantot *MATLAB* pakotni grafiskā režīmā

Lai uz klastera piekļuves servera (*ui-1.hpc.rtu.lv*) startētu *Matlab* grafisko saskarni, pieslēdzieties klastera komandrindai, izmantojot iepriekš aprakstīto grafiskās sesijas pārsūtīšanu.

Piekļuves servera komandrindā izpildiet:

```
module load matlab/R2017b
matlab
```

Jūsu datorā tiks atvērts programmas logs, bet visas darbības tiks izpildītas attāli uz klastera. *MATLAB* saskarne nodrošina vajadzīgos rīkus uzdevuma ievietošanai rindā un izpildei uz skaitļošanas mezgliem.

## 12. Noderīgas saites

RTU HPC centra mājaslapa: <http://hpc.rtu.lv>

Uzdevumu pārvaldības sistēmas (*Torque/Moab*) dokumentācija:

<http://www.adaptivecomputing.com/support/documentation-index/>

Noderīgas pamācības: <https://www.hpc.science.unsw.edu.au/about/hpc-basics>