

# C2115

# Praktický úvod do superpočítání

VII. lekce

Petr Kulhánek, Jakub Štěpán

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta,  
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

## ➤ **Dávkové systémy**

definice, přehled

## ➤ **Instalace Ubuntu Server 12.04.3**

VirtualBox, přístup přes ssh

## ➤ **Instalace a konfigurace dávkového systému Torque**

baličky, konfigurace výpočetního uzlu, konfigurace front, spuštění úlohy

# Dávkové systémy

# Dávkové zpracování

**Dávkové zpracování** je vykonávání série programů (tzv. dávek) na počítači bez účasti uživatele. Dávky jsou připraveny předem, takže mohou být zpracovány předem bez účasti uživatele. Všechna vstupní data jsou předem připravena v souborech (skriptech) nebo zadána pomocí parametrů na příkazovém řádku. Dávkové zpracování je opakem interaktivního zpracování, kdy uživatel až teprve za běhu programu poskytuje požadované vstupy.

## Výhody dávkového zpracování

- sdílení zdrojů počítače mezi mnoha uživateli a programy
- odložení zpracování dávek do doby, kdy je počítač méně vytížen
- odstranění prodlev způsobeným čekáním na vstup od uživatele
- maximalizace využití počítače zlepšuje využití investic (zejména u dražších počítačů)

zdroj: [www.wikipedia.cz](http://www.wikipedia.cz), upraveno

# Nástroje pro dávkové spouštění

➤ **OpenPBS**

<http://www.mcs.anl.gov/research/projects/openpbs/>

➤ **PBSPro**

<http://www.pbsworks.com>

➤ **Oracle Grid Engine**

<http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html>

➤ **Open Grid Scheduler**

<http://gridscheduler.sourceforge.net/>

➤ **Torque**

<http://www.adaptivecomputing.com/products/open-source/torque/>

# Nástroje pro dávkové spouštění

➤ **OpenPBS**

<http://www.mcs.anl.gov/research/projects/openpbs/>

➤ **PBSPro**

<http://www.pbsworks.com>

➤ **Oracle Grid Engine**

<http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html>

➤ **Open Grid Scheduler**

<http://gridscheduler.sourceforge.net/>

**open source**

➤ **Torque**

<http://www.adaptivecomputing.com/products/open-source/torque/>

# Nástroje pro dávkové spouštění

➤ **OpenPBS**

<http://www.mcs.anl.gov/research/projects/openpbs/>

➤ **PBSPro**

<http://www.pbsworks.com>

➤ **Oracle Grid Engine**

<http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html>

➤ **Open Grid Scheduler**

<http://gridscheduler.sourceforge.net/>

**open source**

➤ **Torque**

<http://www.adaptivecomputing.com/products/open-source/torque/>

je použit jako dávkový systém v MetaCentrum VO, na klastrech SOKAR a WOLF

# Ubuntu 12.04.3



# Cvičení LVII.1

1. Nainstalujte Ubuntu Server 12.04.3 do virtuálního prostředí VirtualBox. Při instalaci nainstalujte ssh server.
  1. Nastavení VirtualBoxu (File -> Preferences)
    1. Default Machine Folder: změňte na podadresář (dle vaší volby) ve vašem scratch adresáři
  2. Nastavení virtuálního stroje
    1. Network -> Attached to: NAT
    2. Network -> Advanced -> Port Forwarding
      1. Host Port: 2222
      2. Guest Port: 22
      3. Zbytek ponechat nezměněn
2. Přihlaste se do běžící instance serveru přes grafické rozhraní virtualizačního prostředí.
3. Přihlaste se do běžící instance serveru pomocí programu ssh z hostitelského počítače.  
`ssh -p 2222 server_login@localhost`

# Cvičení LVII.2

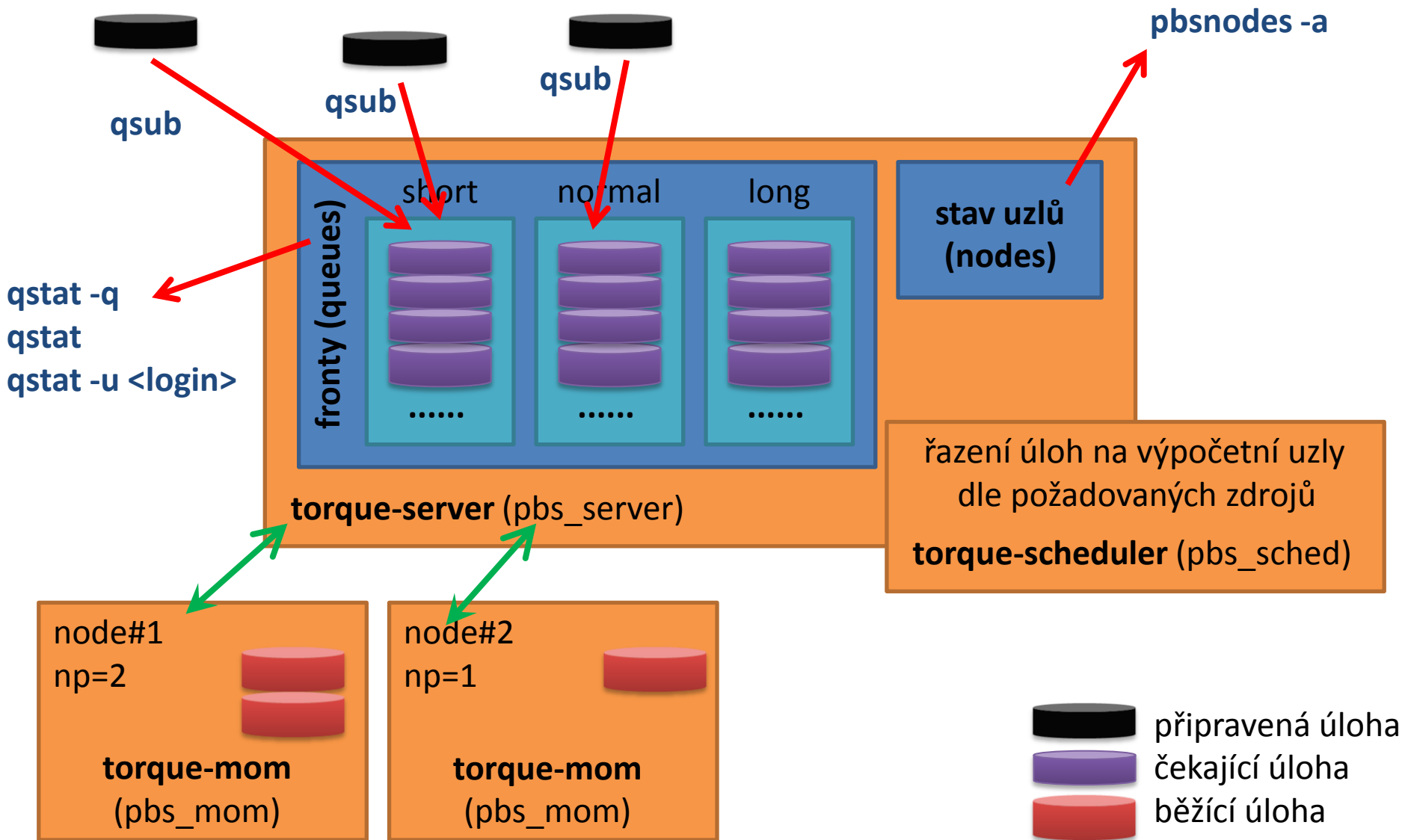
1. Nainstalujte program mc :
2. Vypněte server:
3. Zapněte server.

```
$ sudo apt-get install mc
```

```
$ sudo poweroff
```

# Torque

# Architektura



# Instalace

1. Instalace balíčků pro klienta, server a výpočetní uzel:

```
# apt-get install torque-client torque-client-x11
# apt-get install torque-server
# apt-get install torque-mom
```

2. Konfigurace sítě. IP adresu rozhraní eth0 je nutné přiřadit jménu hosta v souboru /etc/hosts místo výchozí hodnoty 127.0.1.1

```
# ifconfig
    inet addr:10.0.2.25 . . . .
```

```
# vi /etc/hosts
127.0.0.1        localhost
10.0.2.15      ubuntu
```

3. Změna výchozího jména pro Torque server. Editujeme soubor /etc/torque/server\_name nebo /var/spool/torque/server\_name

```
# vi /etc/torque/server_name
ubuntu
```

# Restart server a prvotní připojení

## 1. Restart Torque serveru a plánovače:

```
# service torque-server stop
# service torque-server start
# service torque-scheduler stop
# service torque-scheduler start
```

## 2. Výpis seznamu výpočetních uzlů:

```
# pbsnodes -a
pbsnodes: Server has no node list MSG=node list is empty
- check 'server_priv/nodes' file
```

# Konfigurace - uzly

Konfigurace se provádí příkazem **qmgr**. V našem případě bude virtuální stroj vystupovat v roli výpočetního uzlu.

Konfiguraci lze provést editací souboru **/var/spool/torque/server\_priv/nodes**. Každá jeho změna však vyžaduje restart Torque serveru. Vhodnější alternativou je použití příkazu **qmgr**

```
# qmgr
Qmgr: create node ubuntu
Qmgr: set node ubuntu np = 1
Qmgr: set node ubuntu properties = node
Qmgr: set node ubuntu properties += master
Qmgr: list node @ubuntu
Node ubuntu
    state = free
    np = 1
    properties = master,node
    ntype = cluster
    status = .....
```

jméno uzlu

počet dostupných CPU pro výpočty

volitelné nastavení

jméno serveru  
bez jména uzlu vypíše všechny uzly  
(ekvivalent pbsnodes -a)

# Konfigurace - fronty

Konfigurace se provádí příkazem **qmgr**.

```
# qmgr
Qmgr: create queue normal
Qmgr: set queue normal queue_type = execution
Qmgr: set queue normal enabled = true
Qmgr: set queue normal started = true
Qmgr: print queue @ubuntu
....
....
```

jméno fronty



minimální konfigurace



jméno serveru



bez jména fronty vypíše všechny fronty  
(ekvivalent qstat -q)

```
$ qstat -q
server: ubuntu
Queue           Memory CPU Time Walltime Node  Run Que Lm  State
-----
normal          --    --    --    --    --    0  0  --  E R
-----
                                0    0
```



# Konfigurace - serveru

Konfigurace se provádí příkazem **qmgr**. Výchozí nastavení je dostačující pro běžné použití.

```
# qmgr
```

```
Qmgr: set server scheduling = True
```

aktivuje plánování úloh

```
Qmgr: print server
```

vypíše konfiguraci serveru

```
#
```

```
create queue normal
```

```
set queue normal queue_type = Execution
```

```
set queue normal enabled = True
```

```
set queue normal started = True
```

```
#
```

```
set server scheduling = True
```

```
set server acl_hosts = ubuntu
```

```
set server log_events = 511
```

```
set server mail_from = adm
```

```
set server scheduler_iteration = 600
```

```
set server node_check_rate = 150
```

```
set server tcp_timeout = 6
```

```
set server next_job_number = 0
```

# Přihlašování bez hesla

Mezi výpočetními uzly a serverem (a naopak) je nutné nastavit přihlašování pomocí ssh bez explicitního zadávání hesla. Postup viz prerekvizity.

# Zadáváme úlohy

K zadávání úloh do dávkového systému se používá příkaz **qsub**.

```
$ qsub -q normal uloha.sh  
1.ubuntu
```

jméno fronty, do které  
chceme úlohu zařadit

skript úlohy, např.

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

příkaz vypíše ID úlohy, pokud  
zařazení proběhne v pořádku

```
$ ls  
uloha.sh uloha.sh.o1 uloha.sh.e1
```

soubory jsou  
dostupné až po  
skončení úlohy

standardní výstup  
úlohy

standardní chybový  
výstup úlohy