

C2115

Praktický úvod do superpočítání

14. lekce

Petr Kulhánek
kulhanek@chemi.muni.cz

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

➤ Infinity

úloha, přehled příkazů

➤ Spouštíme aplikace

gaussian, pmemd paralelní spouštění

➤ Cvičení

efektivita paralelního spouštění aplikaci pmemd

Infinity

<https://lcc.ncbr.muni.cz/whitezone/development/infinity/>

Přehled příkazů

Správa software:

- site aktivace logických výpočetních zdrojů
- module aktivace/deaktivace software

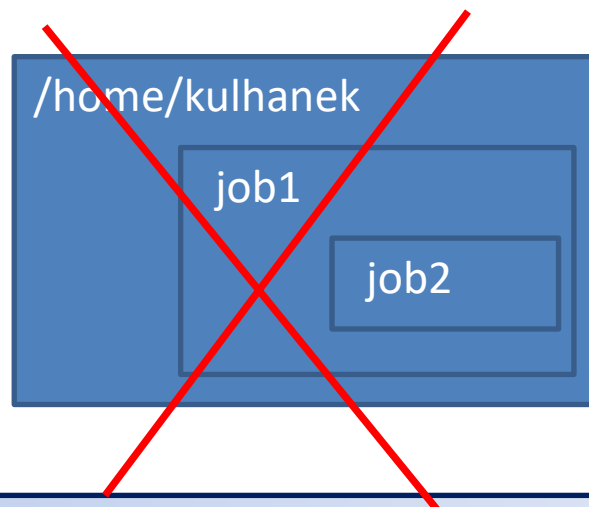
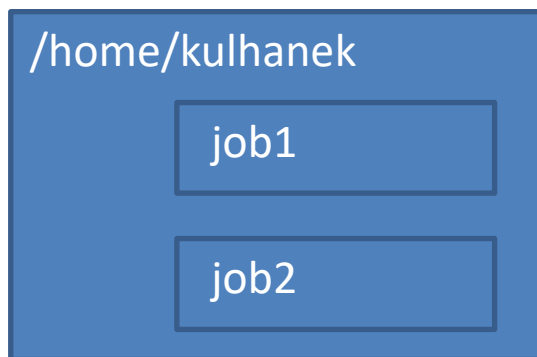
Správa úloh:

- pqueues přehled front z dávkového systému dostupných uživateli
- pnodes přehled výpočetních uzlů dostupných uživateli
- pqstat přehled všech úloh zadaných do dávkového systému
- pjobs přehled úloh uživatele zadaných do dávkového systému
- psubmit zadání úlohy do dávkového systému
- pinfo informace o úloze
- pgo přihlásí uživatele na výpočetní uzel, kde se úloha vykonává
- psync manuální synchronizace dat

Úloha

Úloha **musí splňovat** následující podmínky:

- každá úloha se spouští v samostatném adresáři
- všechny vstupní data úlohy musí být v adresáři úlohy
- adresáře úloh nesmí být do sebe zanořené
- průběh úlohy je řízen skriptem nebo vstupním souborem (u automaticky detekovaných úloh)
- skript úlohy musí být v bashi
- ve skriptu úlohy se nesmí používat absolutní cesty, všechny cesty musí být uvedeny relativně k adresáři úlohy



Skript úlohy

Skript úlohy může být uvozen standardním interpretrem pro **bash** nebo speciálním interpretrem **infinity-env**, který nedovolí spuštění úlohy mimo výpočetní uzel. Druhý přístup zabráňuje případnému poškození/přepsání/smazání již vypočtených dat nechtěným opětovným spuštěním skriptu.

```
#!/bin/bash
```

```
# vlastní skript
```

```
#!/usr/bin/env infinity-env
```

```
# vlastní skript
```

Spuštění úlohy

Úlohu spouštíme **v adresáři úlohy** příkazem **psubmit**.

```
psubmit destination job [resources]
```

destination (kam) je buď:

- název_fronty
- alias

job je buď:

- název skriptu úlohy
- název vstupního souboru pro automaticky rozpoznávané úlohy

resources jsou požadované zdroje pro úlohu, pokud není uvedeno, požaduje se běh na 1 CPU

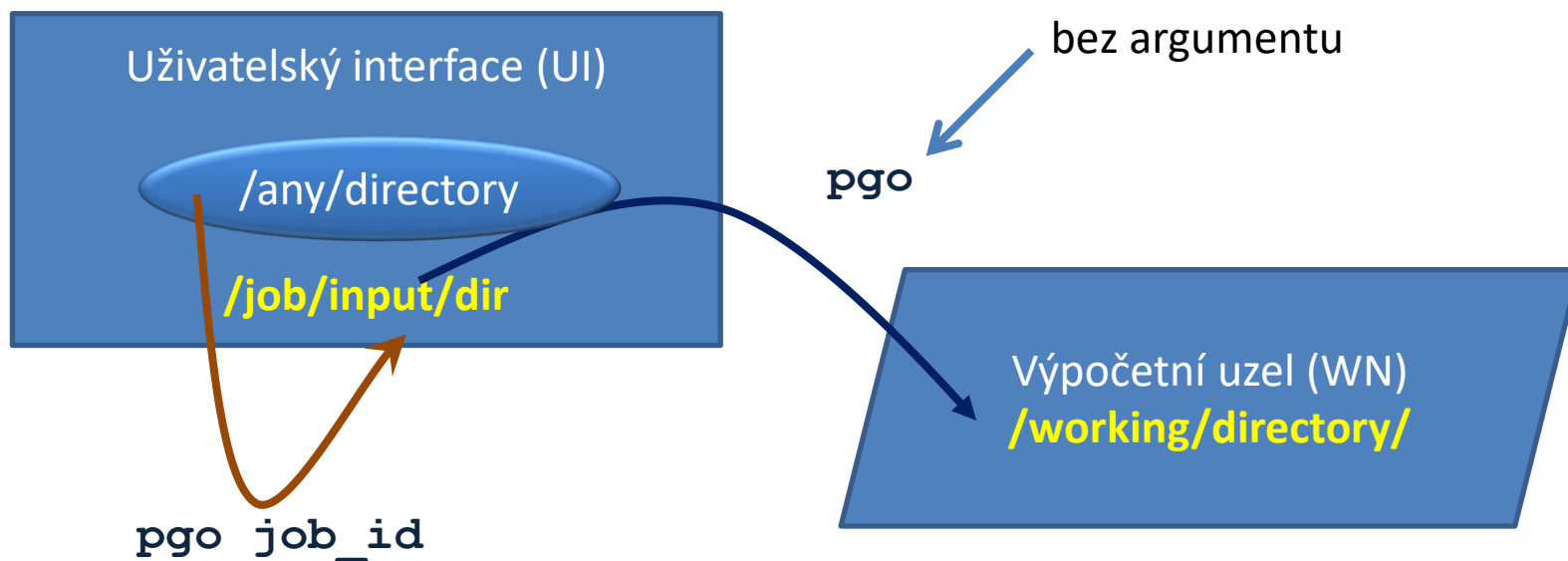
Specifikace zdrojů (nejdůležitější)

Zdroj	Popis
ncpus	celkový počet požadovaných CPU
ngpus	celkový počet požadovaných GPU
nnodes	počet výpočetních uzlů (WN)
mem	celková velikost požadované paměti (CPU), jednotky mb, gb
walltime	maximální doba běhu úlohy
workdir	typ pracovního adresáře na WN
place	způsob obsazování výpočetních uzlů
props	požadované vlastnosti výpočetních uzlů

Monitorování běhu úlohy

K monitorování průběhu úlohy lze použít příkaz **pinfo**, který se spouští buď v adresáři úlohy nebo v pracovním adresáři na výpočetním uzlu. Dalšími možnostmi jsou příkazy **pjobs** a **pqstat**.

Pokud je úloha spuštěna na výpočetním uzlu, je možné použít příkaz **pgo**, který přihlásí uživatele na výpočetní uzel a změní aktuální adresář do pracovního adresáře úlohy.



Monitoring úlohy v terminálu.

Servisní soubory

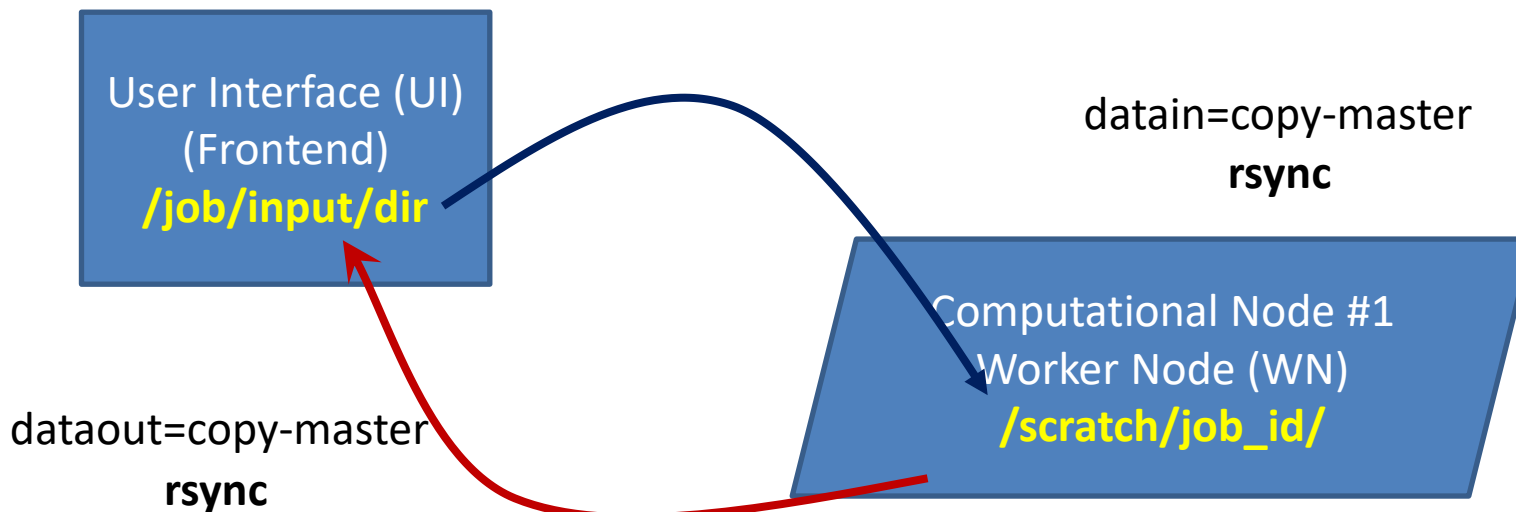
V adresáři úlohy vznikají při zadání úlohy do dávkového systému a dále v průběhu života úlohy a po jejím ukončení servisní soubory. Jejich význam je následující:

- *.info kontrolní soubor s informacemi o průběhu úlohy
- *.infex vlastní skript (wrapper), který se spouští dávkovým systémem
- *.infout standardní výstup z běhu *.infex skriptu, **nutno analyzovat při nestandardním ukončení úlohy**
- *.nodes seznam uzlů vyhrazených pro úlohu
- *.mpinodes seznam uzlů vyhrazených pro úlohu ve formátu pro OpenMP
- *.gpus seznam GPU karet vyhrazených pro úlohu
- *.key unikátní identifikátor úlohy
- *.stdout **standardní výstup z běhu skriptu úlohy**

Synchronizace dat

Výchozí pracovní režim

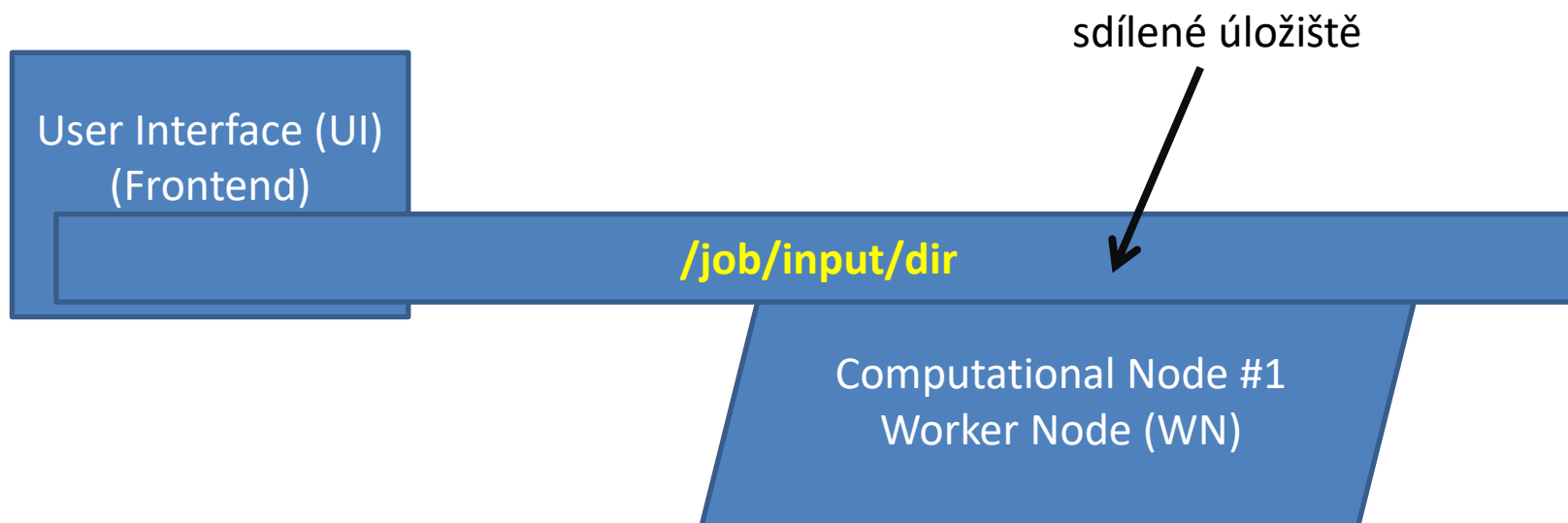
Zdroj	Význam
<code>workdir=scratch-local</code>	Data se zkopírují ze vstupního adresáře úlohy do pracovního adresáře ve výpočetním uzlu. Pracovní adresář je vytvořen na začátku úlohy dávkovým systémem. Po dokončení úlohy se všechna data z pracovního adresáře zkopírují zpět do vstupního adresáře úlohy. Nakonec bude pracovní adresář odstraněn, pokud byl přenos dat úspěšný.



Synchronizace dat, pokr.

Vhodné pro analýzy

Zdroj	Význam
workdir= jobdir	Data úlohy jsou na sdíleném úložišti.



Spouštíme aplikace

Požadavek/využití zdrojů

Dávkový systém

Úloha

Nativní dávkový systém (PBSPro)

- **uživatel určuje** požadované výpočetní zdroje
- **uživatel musí zajistit**, aby úloha přiřazené výpočetní zdroje využila

Infinity

- **uživatel určuje** požadované výpočetní zdroje
- **prostředí Infinity zajistí** správné spuštění úlohy (pouze vybrané aplikace)
- (ostatní úlohy) **uživatel musí zajistit**, aby úloha přiřazené výpočetní zdroje využila

pmemd

pmemd je program určen pro molekulovou dynamiku. Podrobnější informace lze nalézt zde: <http://ambermd.org>

Skript pro běh aplikace na CPU:

```
#!/bin/bash

# aktivovat modul amber obsahující aplikaci
# pmemd
module add amber

# spuštění aplikace
pmemd -O -i prod.in -p 6000.parm7 \
      -c 6000.rst7
```

pmemd – paralelní běh

Při paralelním spouštění se mění jen zadání zdrojů u příkazu psubmit. **Ostatní se nemění!** (zůstávají stejná vstupní data a skript úlohy).

```
$ psubmit default run.sh ncpus=1
```



může se vynechat

*.stdout

```
.....  
Module build: amber:16.0:x86_64:single  
.....
```

Výpočetní uzel:

S	%CPU	%MEM	TIME+	COMMAND
R	99.7	0.2	0:06.41	pmemd
S	0.3	0.0	0:00.01	sshd
R	0.3	0.0	0:00.09	top

```
$ psubmit default run.sh ncpus=2
```

*.stdout

```
.....  
Module build: amber:16.0:x86_64:para  
.....
```

Výpočetní uzel:

S	%CPU	%MEM	TIME+	COMMAND
R	100.0	0.2	0:03.64	pmemd.MPI
R	100.0	0.2	0:03.64	pmemd.MPI
R	0.3	0.0	0:00.06	top


gaussian, manual script preparation

The **gaussian** package contains tools for quantum chemical calculations. Detailed description can be found on <http://www.gaussian.com>

```
#!/bin/bash

# activate gaussian module
module add gaussian:09.C1

# execute g09
g09 input
```



input file **input.com** must contain specification for number of CPUs requested for parallel execution (this number MUST be consistent with resource specification via **psubmit** command).

```
%NProcShared=4
```

```
$ psubmit short test_gaussian ncpus=4
```



gaussian, manual script preparation

The **gaussian** package contains tools for quantum chemical calculations. Detailed description can be found on <http://www.gaussian.com>

```
#!/bin/bash

# activate gaussian module
module add gaussian:09.C1

# execute g09
g09 input
```

NOT RECOMMENDED

input file **input.com** must contain specification for number of CPUs requested for parallel execution (this number **MUST** be consistent with resource specification via **psubmit** command).

```
%NProcShared=4
```

```
$ psubmit short test_gaussian ncpus=4
```

gaussian, autodetection

The ABS subsystem is able to recognize the gaussian job type. The job script is automatically created and the input file is automatically updated according to requested resources.

```
$ module add gaussian
$ psubmit short input.com ncpus=4
```



gaussian input file (must have .com extension), **this is NOT job script!**

Autodetection:

- job script is created automatically with correct gaussian binary name (g98, g03, g09)
- %NProcShared is added or updated in the input file
- check if only single node is requested (parallel execution is limited to a single node)

```
[kulhanek@perian test]$ psubmit short input.com
```

```
Job name      : input
Job title     : input (Job type: gaussian)
Job directory : perian.ncbr.muni.cz:/home/kulhanek/Tests/test
Job project   : -none- (Collection: -none-)
Site name     : metacentrum (Torque server: arien.ics.muni.cz)
Job key      : 384e3be5-9dac-405e-b235-74609ae4c486
```

```
=====
```

gaussian – single/parallel execution

The only difference between sequential and parallel execution is in the resource specification during psubmit. **The input data are the same!**

```
$ psubmit short input.com ncpus=1
```

↑
it can be omitted

Computational node:

S	%CPU	%MEM	TIME+	COMMAND
R	100	1.2	1:01.25	l502.exe
S	0	0.1	1:38.57	pbs_mom

```
$ psubmit short input.com ncpus=4
```

Computational node:

S	%CPU	%MEM	TIME+	COMMAND
R	399	1.1	0:49.38	l502.exe
S	0	0.1	0:00.90	init

Cvičení

Cvičení 1

1. Vypočítejte molekulární vibrace molekuly fulleren v programu gaussian. Úlohu zadejte na klastr WOLF pomocí prostředí infinity. Využijte autodetekce typu úlohy. Zdroje (velikost paměti a diskového prostoru) nastavte na hodnoty nalezené při spuštění úlohy v MetaCentru.

Cvičení 2

Vstupní data úlohy jsou na klastru WOLF v adresáři:
`/home/kulhanek/Documents/C2115/data/chitin/cpu`

1. Cílem cvičení je určit jak dobře škáluje aplikace `pmemd` v rozsahu počtu CPU, které jsou násobky dvou. Určete skutečnou a teoretickou délku výpočtu, reálné urychlení a reálné využití CPU v procentech. Do grafu vynesete reálné urychlení jako funkci počtu CPU. Nalezenou křivku porovnejte s křivkou pro ideální škálování.
2. Úlohy zadávejte pomocí prostředí Infinity s proměnným množstvím `ncpus`. Každý test spouštějte v samostatném adresáři. Bez ohledu na počet `ncpus` vždy požadujte celý uzel (`place=excl`) a používejte stejný výpočetní uzel (`vnode=wolf30`).

Způsob zadání úlohy:

```
$ psubmit default run.sh ncpus=8 place=excl vnode=wolf30
```

Viz následující stránka s poznámkami

Délka simulace:

Délka simulace (výpočtu) je určena klíčovým slovem (**nstlim**) uvedeným v souboru prod.in, který určuje počet integračních kroků. Velikost nstlim zvolte tak, aby doba běhu úlohy byla cca 60 minut na 1 CPU.

Výsledkem simulace jsou soubory:

mdout

mdinfo

<-- obsahuje statistické informace, např. kolik ns za den je program schopen nasimulovat

mdcrd

restrt