

C2115

Praktický úvod do superpočítání

XI. lekce

Petr Kulhánek, Tomáš Bouchal

kulhanek@chemi.muni.cz

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

Obsah

➤ Infinity

úloha, přehled příkazů, aliasy

➤ Spouštíme aplikace

sander, pmemd, gaussian, paralelní spouštění

➤ Cvičení

efektivita paralelního spouštění aplikaci sander, pmemd, gaussian

Infinity

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

Přehled příkazů

Správa software:

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

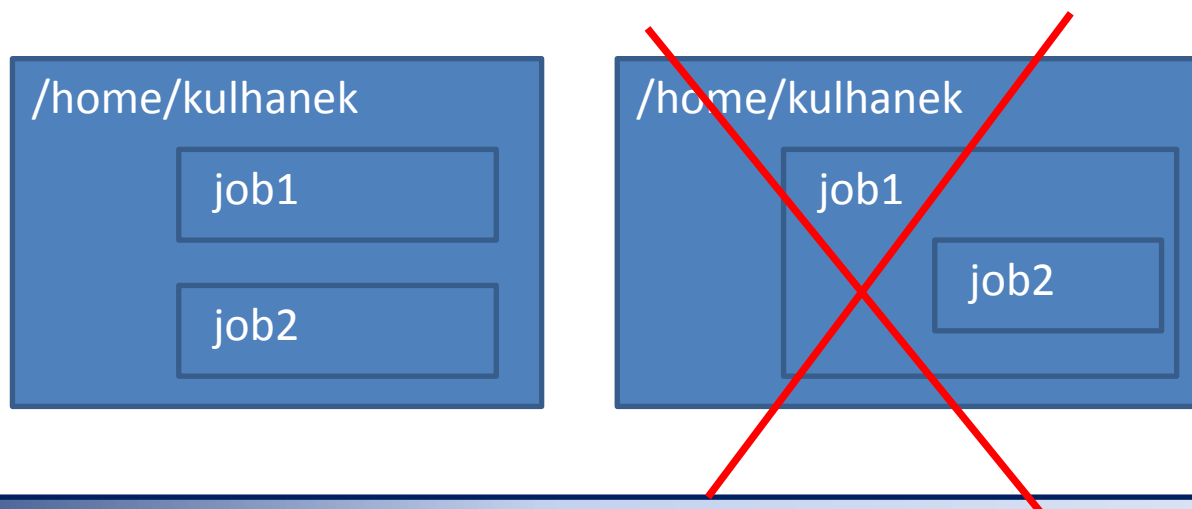
Správa úloh:

- 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
- paliases definování aliasů

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] [syncmode]
```

destination (kam) je buď:

- název_fronty
- název_uzlu@název_fronty

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

syncmode určuje způsob kopírování dat mezi adresářem úlohy a výpočetním uzlem, výchozím módem je "sync"

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 se naloguje na výpočetní uzel a změní aktuální adresář do pracovního adresáře úlohy.

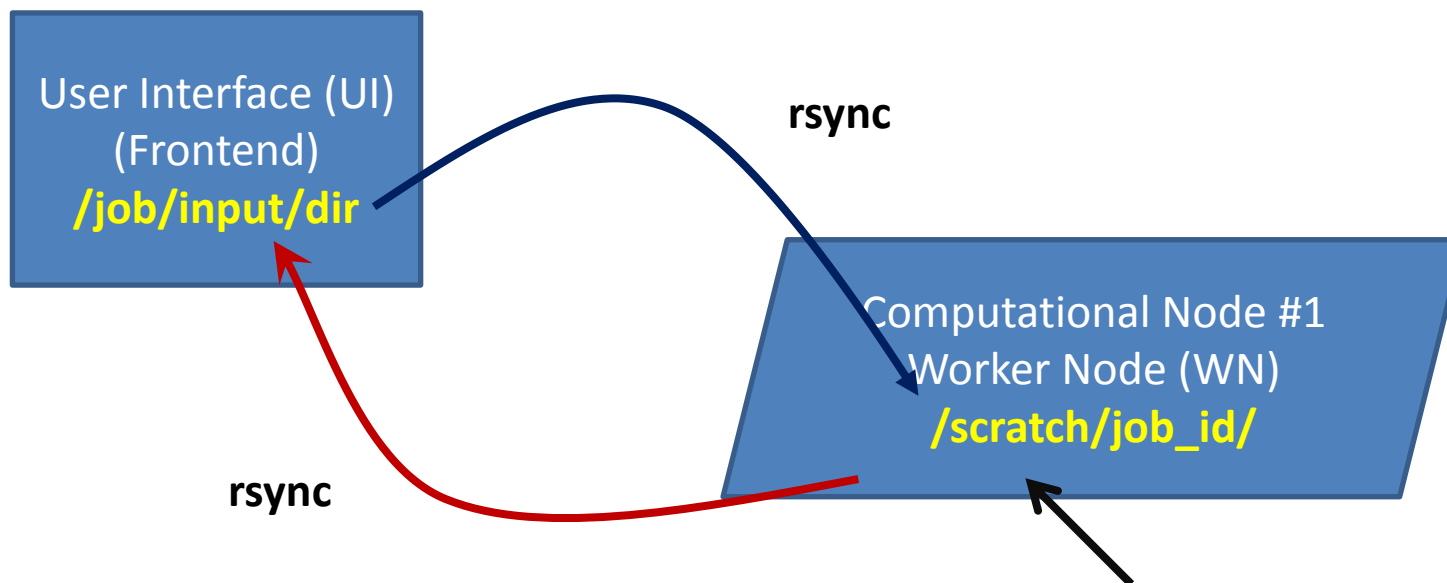
Servisní soubory

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
- *.gpus seznam GPU karet vyhrazených pro úlohu
- *.key unikátní identifikátor úlohy
- *.stdout **standardní výstup z běhu skriptu úlohy**

Synchronization modes, sync

Mode	Meaning
sync	Data are copied from the job input directory to the working directory on the computational node. The working directory is created on the scratch of the computational node. After the job is finished, all data from the working directory are copied back to the job input directory. Finally, the working directory is removed if the data transfer was successful.



Note: default synchronization mode

determined by **scratch_type** resource token

Spouštíme aplikace

sander

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

```
#!/bin/bash

# aktivovat modul amber obsahující aplikace
# sander a pmemd
module add amber

# spuštění aplikace
sander -O -i prod.in -p topology.parm7 \
      -c input.rst7
```

pmemd

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

```
#!/bin/bash

# aktivovat modul amber obsahující aplikace
# sander a pmemd
module add amber

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

Délka simulace:

Délka simulace (výpočtu) je určena klíčovým slovem (**nstim**) uvedeným v souboru prod.in, který určuje počet integračních kroků.

Výsledkem simulace jsou soubory:

mdout

mdinfo

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

mdcrd

restrt

sander/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 short test_sander ncpus=1
```



může se vynechat

*.stdout

```
.....  
Module build: amber:12.0:x86_64:single  
.....
```

Výpočetní uzel:

S	%CPU	%MEM	TIME+	COMMAND
R	100	0.6	1:13.37	sander
R	0	0.0	0:00.01	top

```
$ psubmit short test_sander ncpus=2
```

*.stdout

```
.....  
Module build: amber:12.0:x86_64:para  
.....
```

Výpočetní uzel:

%CPU	%MEM	TIME+	COMMAND
100	1.6	0:40.41	sander.MPI
99	1.7	0:40.60	sander.MPI
0	1.2	0:52.86	unity.greet

Vstupní data:

/home/kulhanek/Data/C2115/data/sander/small

1. Spusťte úlohu na 1CPU na klastru wolf.
2. Spusťte úlohu na 2CPU na klastru wolf.

gaussian

gaussian je program určen pro kvantově-chemické výpočty. Podrobnější informace lze nalézt zde: <http://www.gaussian.com>

```
#!/bin/bash

# aktivovat modul gaussian
module add gaussian

# spusteni aplikace
g09 input
```

vstupni soubor **input.com** bez zakončení

gaussian

Délka výpočtu:

Délku výpočtu lze ovlivnit maximálním počtem optimalizačních kroků (**MaxCycle** ve vstupním souboru).

Výsledkem výpočtu je soubor:

input.log

gaussian – paralelní běh

Při paralelním spuštění se mění zadání zdrojů u příkazu psubmit a vstupní soubor pro program gaussian (input.com).

input.com (první řádek)

```
%NProcShared=1
```

```
$ psubmit short test_gaussian ncpus=1
```

↗
může se vynechat

Výpočetní uzel:

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

input.com (první řádek)

```
%NProcShared=4
```

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

identické číslo!!!

Výpočetní uzel:

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

gaussian – paralelní běh, II

Při spouštění výpočtů v gaussianu lze využít autodetekci.

Bez autodetekce:

input.com (první řádek)

```
%NProcShared=4
```

identické číslo!!!



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

S autodetekcí:

input.com (nemusi obsahovat %NProcShared=4)

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

Cvičení

Vstupní data:

/home/kulhanek/Data/C2115/data/gaussian

1. Spusťte úlohu na 1CPU na klastru wolf.
2. Spusťte úlohu na 2CPU na klastru wolf.

Cvičení

Cvičení LIX.1

Cílem cvičení je určit jak dobře škálují aplikace sander, pmemd a gaussian na stroji pip v rozsahu počtu CPU 1, 6, 12, 24, 48, 96 a 192. Určete skutečnou a teoretickou délku výpočtu, reálné urychlení a reálné využití CPU v procentech. Do grafu vyneste reálné urychlení jako funkci počtu CPU. Nalezenou křivku porovnejte s křivkou pro ideální škálování.

Testovací výpočty můžete provádět na klastru WOLF. Finální výpočty pak budete provádět na stroji pip.

K výpočtům používejte frontu long.

Vstupní data jsou na klastru WOLF v adresářích:

Lesson09/pmemd/small/

Lesson09/pmemd/big/

Lesson09/sander/small/

Lesson09/sander/big/

Lesson09/gaussian/