

MATLAB @ MetaVO

aneb

„Jak u nás v Matlabu počítati...“

Tomáš Rebok

MetaCentrum, CESNET z.s.p.o.

CERIT-SC, Masarykova univerzita

(rebok@ics.muni.cz)

SW Matlab & MetaVO I.

Dostupné verze:

- Matlab verze 8.4
 - `module add matlab-8.4`
 - `module add matlab` **# vždy nejnovější verze**
- + starší verze 8.3, 8.2, 8.1, 8.0

Dostupné množství licencí (nezávisle na verzi):

- *Matlab* ... **450** licencí
- *Simulink* ... **150** licencí
- + *mnoho toolboxů*
- *jen pro veřejný výzkum/akademické využití!*

SW Matlab & MetaVO II.

Bioinformatics_Toolbox 15
Communication_Toolbox 25
Video_and_Image_Blockset 1
Control_Toolbox 50
Curve_Fitting_Toolbox 52
Signal_Blocks 50
Data_Acq_Toolbox 2
Database_Toolbox 9
Datafeed_Toolbox 1
Econometrics_Toolbox 1
RTW_Embedded_Coder 1
Fin_Instruments_Toolbox 2
Financial_Toolbox 2
Fixed_Point_Toolbox 1
Fuzzy_Toolbox 51
GADS_Toolbox 1
Image_Acquisition_Toolbox 2
Image_Toolbox 31
Instr_Control_Toolbox 1
MATLAB_Builder_for_Java 5
MATLAB_Coder 7
Compiler 7
Neural_Network_Toolbox 150

Optimization_Toolbox 150
Distrib_Computing_Toolbox 15
PDE_Toolbox 50
Real-Time_Win_Target 51
Signal_Toolbox 50
SimBiology 5
SimHydraulics 1
SimMechanics 4
Power_System_Blocks 1
Simscape 5
Virtual_Reality_Toolbox 3
Real-Time_Workshop 2
Simulink_Control_Design 50
Excel_Link 1
Statistics_Toolbox 50
Symbolic_Toolbox 150
Identification_Toolbox 51
Vehicle_Network_Toolbox 1
Wavelet_Toolbox 8
Simulink_HDL_Coder 1
Embedded_IDE_Link 1
Target_Support_Package 1
MATLAB_Distrib_Comp_Engine 320

SW Matlab & MetaVO III.

Jak zajistit dostupnost licence při spuštění úlohy?

- tj. zamezit „pádu“ výpočtu pro nedostupnou licenci
- → oznámit plánovači požadavek na konkrétní licenci
 - -l matlab=1
 - -l simulink=3
 - -l matlab_Identification_Toolbox=3
 - pro **toolboxy obecně**: -l matlab_*jméno_toolboxu*
 - **skutečný příklad:**

```
$ qsub -l nodes=1:ppn=4 -l matlab=1 -l  
matlab_Distrib_Computing_Toolbox=1 -l mem=5g skript.sh
```
- využitelné jak pro **dávkové**, tak **interaktivní úlohy**

SW Matlab & MetaVO IV.

- **upozornění: nejedná se o rezervaci licence!**
 - dostupnost licence je zajištěna **pouze při startu úlohy**
 - pokud bude vyžádána později, nemusí být dostupná
 - na licenci **lze počkat** nebo si ji **převzít hned na začátku** výpočtu (návody viz naše dokumentace k aplikaci Matlab)

Jak zjistit aktuální využití licencí?

- `$ /software/matlab-8.4/etc/lmstat -s skiritf.ics.muni.cz -a | grep "in use"`

Použití v MetaVO – interaktivní úlohy I.

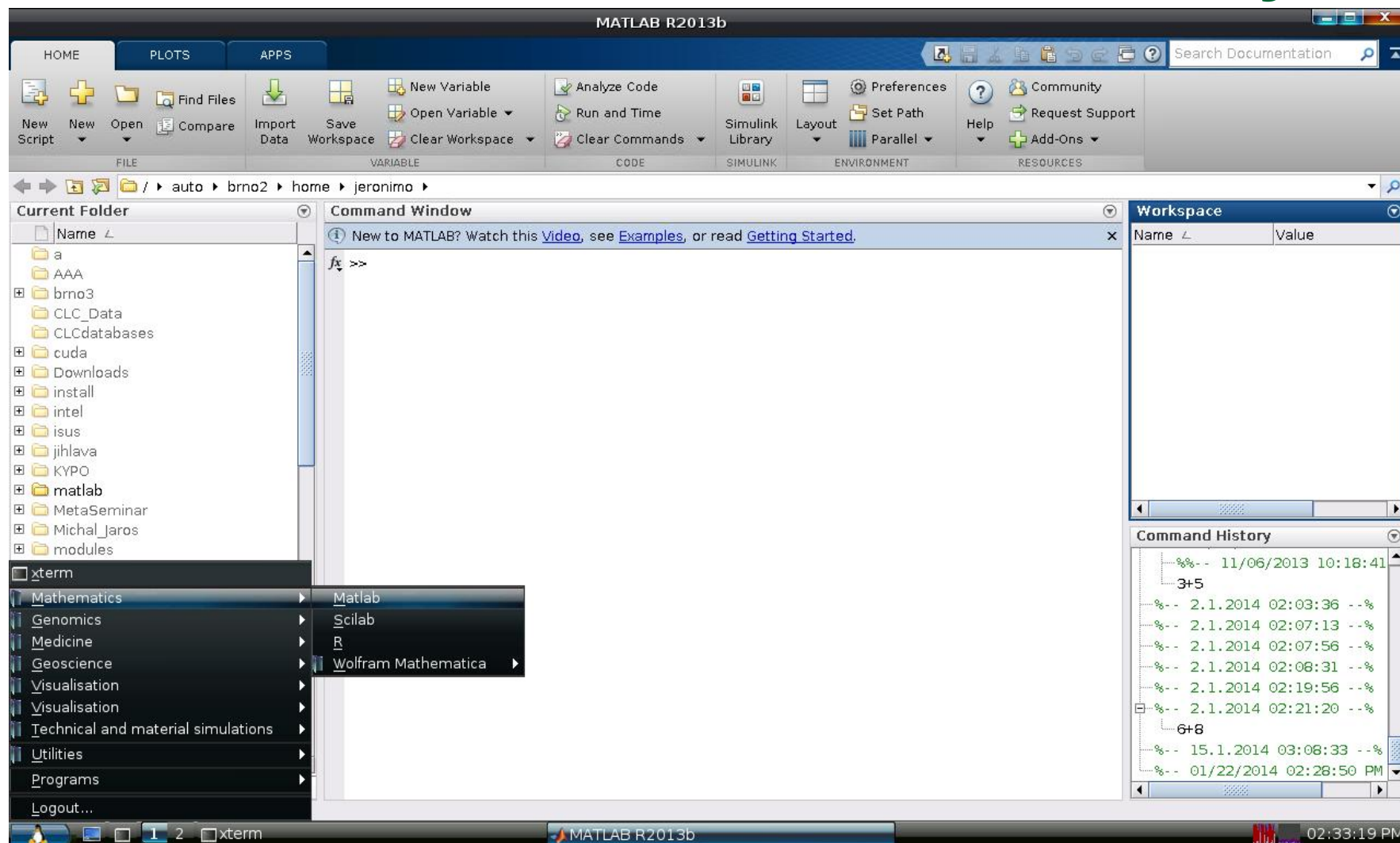
Textový režim:

- jednoduché
- ```
$ module add matlab
$ matlab
```

## Grafický režim:

- využijte prostředí VNC serverů
- ```
$ module add gui
$ gui start
*****
Your VNC session has been started.
The connection details are as follows:
  Host & port : doom20.metacentrum.cz:10316
  Password   : YsXkkGN0
  Display    : :23
*****
```
- viz https://wiki.metacentrum.cz/wiki/Vzdálený_desktop

Použití v MetaVO – interaktivní úlohy II.



The screenshot displays the MATLAB R2013b environment. The top ribbon includes tabs for HOME, PLOTS, and APPS. The ribbon contains various toolbars for file operations (New Script, Open, Find Files), workspace management (New Variable, Save, Clear Workspace), code execution (Run and Time, Clear Commands), and environment settings (Simulink Library, Layout, Set Path). The main workspace area is divided into several panes:

- Current Folder:** Shows a directory tree with folders like 'a', 'AAA', 'brno3', 'CLC_Data', 'CLCdatabases', 'cuda', 'Downloads', 'install', 'intel', 'isus', 'jihlava', 'KYPO', 'matlab', 'MetaSeminar', 'Michal_Jaros', 'modules', and 'xterm'.
- Command Window:** Contains the prompt `>>` and a message: "New to MATLAB? Watch this [video](#), see [Examples](#), or read [Getting Started](#)." Below the prompt, the text `3+5` is visible.
- Workspace:** A table with columns 'Name' and 'Value'.
- Command History:** Lists previous commands and their execution times, including `3+5` and `6+8`.

A context menu is open over the 'xterm' folder, listing various applications: Mathematics (Matlab, Scilab, R, Wolfram Mathematica), Genomics, Medicine, Geoscience, Visualisation, Technical and material simulations, Utilities, Programs, and Logout...

The taskbar at the bottom shows the MATLAB R2013b window and a terminal window titled 'xterm'. The system clock indicates the time is 02:33:19 PM.

Použití v MetaVO – dávkové úlohy

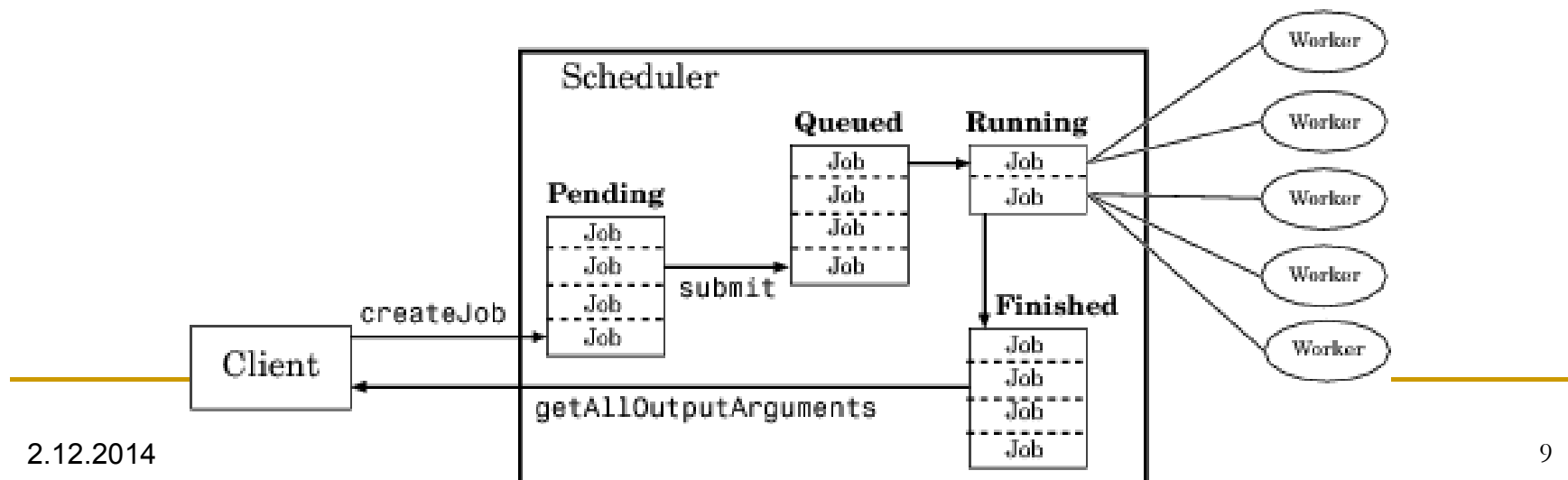
- **obsah popisného skriptu:**

```
...  
module add matlab  
matlab -nodisplay -nosplash <${INPUT_FILE}  
...
```


Matlab & paralelizace/distribuce výpočtů I.

MATLAB Job Scheduler (MJS) – přímá práce s workery:

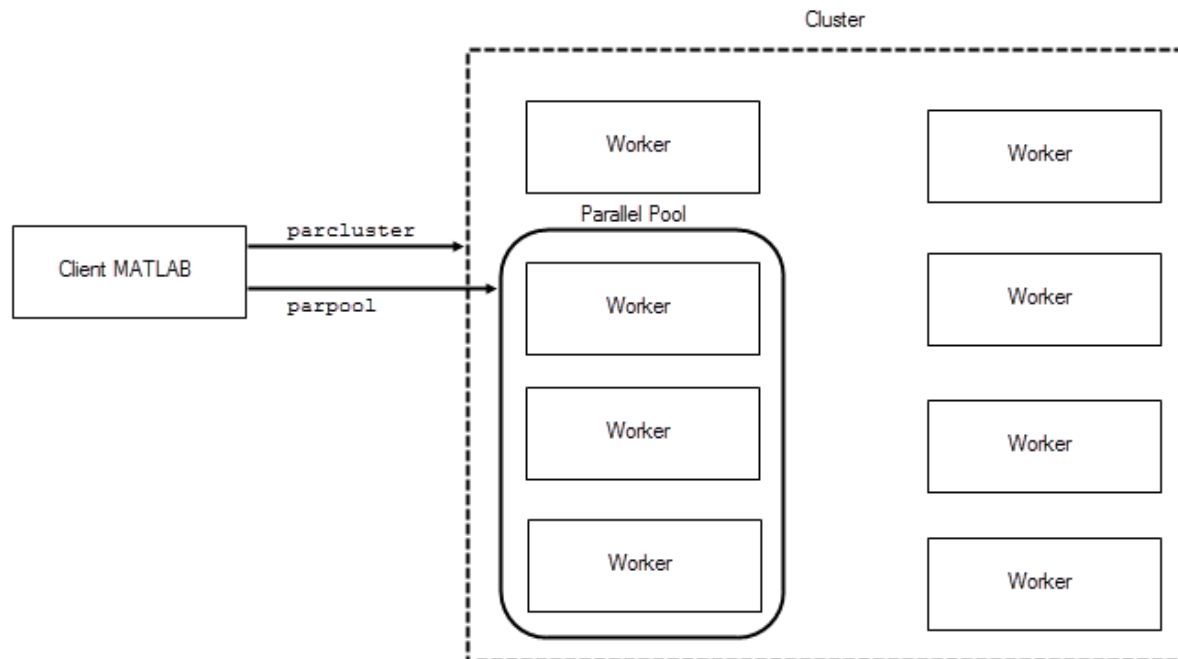
- vytváření **interních úloh Matlabu**, které jsou poté distribuovány na workery
- uživatel pracuje s **MJS**
 - kterému **posílá úlohy** (`createJob` & `submit`)
 - **složené z konkrétních výpočtů/tasků** (`createTask`)
 - a **čeká/odebírá výsledky** (`wait` & `getAllOutputArguments`)



Matlab & paralelizace/distribuce výpočtů II.

Parpool – paralelní pool vytvořený nad workery:

- zpřístupnění plné funkcionality paralelního jazyka systému Matlab (funkce `parfor`, `spmd`, `distributed`, `Composite`, `parfeval`, ...)
- vytváří speciální úlohu nad workery a tuto propojí s klientem
- **jednoduché na použití**, mnoho funkcí umí využít „transparentně“



Matlab & paralelní/distribuované výpočty v MetaVO

MetaParPool() & MetaGridPool()

- připravené MATLAB funkce pro **usnadnění inicializace/konfigurace** paralelního poolu / MJS
- **doplnění** standardní funkcionality MATLABu

Cíl:

- maximální **jednoduchost**
- výpočty **v rámci zdrojů dané úlohy**
 - tj. bez vytváření nových podúloh a čekání hlavní úlohy na jejich spuštění/ukončení
- všechny výpočty **uvnitř sandboxu úlohy**
 - *korektní accounting* využitých zdrojů
 - *korektní ukončení procesů* při násilném ukončení úlohy
- **efektivní využití licencí**

Matlab Job Scheduler (MJS) & MetaVO I.

*Přímá práce s workery – funkce **MetaGridPool()***

- plně automatická inicializace Matlab jobscheduleru a workerů
- **automatická detekce počtu jader** přidělených úloze
 - inicializováno **N-1 workerů**
- využitelné **v rámci jednoho i více uzlů** úlohy
 - *jeden uzel*: transparentní vybudování MJS nad tzv. „local“ profilem
 - *více uzlů*: transparentní vytvoření MJS
- **použití:**
 - `jm = MetaGridPool('open');` % návratová hodnota = MJS objekt
 - `size = MetaGridPool('size');`
 - `MetaGridPool('info');` % vypíše info o využití, uzlech, ..., vrátí MJS
 - `MetaGridPool('close');`

Matlab Job Scheduler (MJS) & MetaVO I.

*Přímá práce s workery – funkce **MetaGridPool()***

- plně automatické inicializace Matlab jobscheduleru a workerů

Příklad použití:

```
jm = MetaGridPool('open');
```

```
...
```

```
% Váš výpočet
```

```
job = createJob(jm);
```

```
createTask(job, ...);
```

```
submit(job);
```

```
wait(job);
```

```
...
```

```
MetaGridPool('close');
```

ekt

MJS

Matlab Job Scheduler (MJS) & MetaVO II.

Požadavky na licence:

- **MetaGridPool v rámci jediného uzlu**

- tj. `$ qsub -l nodes=1:ppn=X ...`
- **využívané licence:** 1x MATLAB + 1x Distrib_Computing_Toolbox
- požadavek:
`$ qsub ... -l matlab=1,matlab_Distrib_Computing_Toolbox=1`

- **MetaGridPool v rámci více uzlů**

- tj. `$ qsub -l nodes=X:ppn=Y ...`
- **využívané licence:** 1x MATLAB + 1x Distrib_Computing_Toolbox + (N-1)x MATLAB_Distrib_Comp_Engine
 - $N = X*Y$
- požadavek:
`$ qsub ... -l matlab=1,matlab_Distrib_Computing_Toolbox=1,
matlab_MATLAB_Distrib_Comp_Engine=(N-1)`

Matlab `parpool` & MetaVO I.

Výpočty využívající `parpool` – funkce `MetaParPool()`

- plně automatická inicializace Matlab `parpoolu` v rámci úlohy
- **automatická detekce počtu jader** přidělených úloze
 - inicializováno **N-1 workerů**
- využitelné **v rámci jednoho i více uzlů** úlohy
 - *jeden uzel*: transparentní využití tzv. „local“ profilu
 - *více uzlů*: transparentní využití workerů nad MJS
- **použití:**
 - `MetaParPool('open');` *% návratová hodnota = velikost poolu*
 - `size = MetaParPool('size');`
 - `MetaParPool('close');`

Matlab parpool & MetaVO I.

Vý

Příklad využití (Matlab skript):

- `MetaParPool('open');`
 - `...`
 - `% Váš paralelní výpočet`
 - `% (např. s využitím parfor/spmd)`
 - `...`
 - `MetaParPool('close');`
-
- **použití:**
 - `MetaParPool('open');` *% návratová hodnota = velikost poolu*
 - `size = MetaParPool('size');`
 - `MetaParPool('close');`

Matlab parpool & MetaVO II.

Požadavky na licence:

- **MetaParPool v rámci jediného uzlu**

- tj. `$ qsub -l nodes=1:ppn=X ...`
- **využívané licence:** 1x MATLAB + 1x Distrib_Computing_Toolbox
- požadavek:
`$ qsub ... -l matlab=1,matlab_Distrib_Computing_Toolbox=1`

- **MetaParPool v rámci více uzlů**

- tj. `$ qsub -l nodes=X:ppn=Y ...`
- **využívané licence:** 1x MATLAB + 1x Distrib_Computing_Toolbox + (N-1)x MATLAB_Distrib_Comp_Engine
 - $N = X*Y$
- požadavek:
`$ qsub ... -l matlab=1,matlab_Distrib_Computing_Toolbox=1,matlab_MATLAB_Distrib_Comp_Engine=(N-1)`

Distribuované výpočty ze vzdálených klientů I.

Inicializace MJS workerů na MetaVO infrastruktuře a jejich vzdálené využití pro výpočty (z vlastního klienta)

(Využitelné jak pro práci s MJS, tak parpoolem)

1. požadavek na interaktivní úlohu

- tj. `$ qsub -I -l nodes=X:ppn=Y ...`

2. inicializace MJS a workerů – příkaz `matlab-gridpool`

- `$ module add matlab`
`$ matlab-gridpool open`
`... (informace o průběhu inicializace) ...`
`% To connect the MJS, copy&paste into your Matlab client`
`cluster = parallel.cluster.MJS('Host', 'HOST:PORT',`
`'Name', 'MJS_ID');`
`% if you want to establish a parpool over MJS, uncomment`
`% parpool(cluster);`

Distribuované výpočty ze vzdálených klientů I.

Inicializace MJS workerů na MetaVO infrastruktuře a jejich vzdálené využití pro výpočty (z vlastního klienta)

(Vy) **Příklad použití (na straně vzdáleného klienta) - MJS:**

```
1. cluster = parallel.cluster.MJS('Host',  
2.   'doom14.metacentrum.cz:27490', 'Name',  
   '7304045.arien.ics.muni.cz_MJS');
```

```
...
```

```
% Váš výpočet
```

```
job = createJob(cluster);
```

```
createTask(job, ...);
```

```
submit(job);
```

```
wait(job);
```

```
...
```

Distribuované výpočty ze vzdálených klientů I.

Inicializace MJS workerů na MetaVO infrastruktuře a jejich vzdálené využití pro výpočty (z vlastního klienta)

(**Příklad použití (na straně vzdáleného klienta) - parpool:**

```
1 cluster = parallel.cluster.MJS('Host',  
2   'doom14.metacentrum.cz:27490', 'Name',  
   '7304045.arien.ics.muni.cz_MJS');
```

```
parpool(cluster);
```

```
...
```

```
% Váš paralelní výpočet
```

```
% (např. s využitím parfor/spmd)
```

```
...
```

```
...
```

Distribuované výpočty ze vzdálených klientů II.

Požadavky na licence:

- *(pro spouštěnou interaktivní úlohu):*
 - tj. `$ qsub -I -l nodes=X:ppn=Y ...`
 - **využívané licence:** 1x Distrib_Computing_Toolbox + (N-1)x MATLAB_Distrib_Comp_Engine
 - $N = X \cdot Y$
 - požadavek:
`$ qsub ... -l matlab_Distrib_Computing_Toolbox=1,
matlab_MATLAB_Distrib_Comp_Engine=(N-1)`

Závěrem ...

- Matlab lze v rámci MetaVO infrastruktury využívat **interaktivně i dávkově**
 - interaktivní využití je možné textově i skrze grafické rozhraní
- paralelizace/distribuce výpočtů zjednodušena **inicializačními funkcemi (zatím experimentální využití!)**:
 - `MetaParPool()` – příprava parpoolu
 - `MetaGridPool()` – příprava MJS
 - **ukázkové příklady:** `/software/matlab-meta_ext/examples/`
 - *(standardní přístup inicializace rovněž podporován)*
- možnost **připojení vzdáleného klienta** k poolu workerů:
 - `matlab-metagrid [open|close|info|size]`
 - v případě zájmu lze připravit i profil pro vzdálené zadávání výpočtů

Děkuji Vám za pozornost!

`rebok@ics.muni.cz`



metavo.metacentrum.cz

www.cerit-sc.cz