

# Matlab i HPC

Bartłomiej Frydrych, Erwin Kus

Akademia Górniczo-Hutnicza w Krakowie

Grudzień 2008

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia



# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Spis treści

- 1 Wstęp
- 2 HPC
  - Definicja
  - Zastosowanie
- 3 Matlab
- 4 HPC w Matlabie
  - Ogólne spojrzenie na problem
  - Konfiguracja MATLABA
  - Przykłady zastosowania
- 5 Bibliografia

# Definicja

**HPC:** High-Performance Computing, czyli używanie superkomputerów i klastrów do rozwiązywania skomplikowanych problemów obliczeniowych. Dziś systemy komputerowe zbliżają się do okolic teraflop-ów i są sklasyfikowane jako komputery HPC.

# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
  - Superkomputery
  - Obliczenia równoległe
  - Magazyny danych
  - W operacjach biznesowych
  - Obliczenia kwantowe
  - W klastrach

# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
- Superkomputery
  - Obliczenia równoległe
  - Magazyny danych
  - W operacjach biznesowych
  - Obliczenia kwantowe
  - W klastrach

# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
- Superkomputery
- Obliczenia równoległe
- Magazyny danych
- W operacjach biznesowych
- Obliczenia kwantowe
- W klastrach

# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
- Superkomputery
- Obliczenia równoległe
- Magazyny danych
- W operacjach biznesowych
- Obliczenia kwantowe
- W klastrach



# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
- Superkomputery
- Obliczenia równoległe
- Magazyny danych
- W operacjach biznesowych
- Obliczenia kwantowe
- W klastrach

# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
- Superkomputery
- Obliczenia równoległe
- Magazyny danych
- W operacjach biznesowych
- Obliczenia kwantowe
- W klastrach

# HPC - Zastosowanie

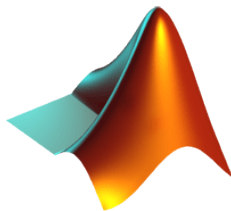
- Zaawansowane problemy obliczeniowe
- Superkomputery
- Obliczenia równoległe
- Magazyny danych
- W operacjach biznesowych
- Obliczenia kwantowe
- W klastrach

# HPC - Zastosowanie

- Zaawansowane problemy obliczeniowe
- Superkomputery
- Obliczenia równoległe
- Magazyny danych
- W operacjach biznesowych
- Obliczenia kwantowe
- W klastrach

# Matlab

MATLAB - program komputerowy będący interaktywnym środowiskiem do wykonywania obliczeń naukowych i inżynierskich, oraz do tworzenia symulacji komputerowych.



# Rozwiązywanie dużych problemów technicznych



# Pętle a równoległość

```
parfor i=1 : n  
//działamy na i  
end
```

- Wyrażenie '//działamy na i' jest rozdzielane pomiędzy użytkowników MATLABA (matlabpool)
  - Jeżeli nie ma żadnych użytkowników, zachowuje się jak zwykła pętla for
- Można używać dla regularnych sesji MATLAB-a
- Analiza "M-Lint" pomaga zlokalizować istniejące pętle for, które mogą zostać zamienione na parfor

# Pętle a równoległość

```
parfor i=1 : n
//działamy na i
end
```

- Wyrażenie '//działamy na i' jest rozdzielane pomiędzy użytkowników MATLABA (matlabpool)
  - Jeżeli nie ma żadnych użytkowników, zachowuje się jak zwykła pętla for
- Można używać dla regularnych sesji MATLAB-a
- Analiza "M-Lint" pomaga zlokalizować istniejące pętle for, które mogą zostać zamienione na parfor



# Pętle a równoległość

```
parfor i=1 : n
//działamy na i
end
```

- Wyrażenie '//działamy na i' jest rozdzielane pomiędzy użytkowników MATLABA (matlabpool)
  - Jeżeli nie ma żadnych użytkowników, zachowuje się jak zwykła pętla for
- Można używać dla regularnych sesji MATLAB-a
- Analiza "M-Lint" pomaga zlokalizować istniejące pętle for, które mogą zostać zamienione na parfor

# Pętle a równoległość

```
parfor i=1 : n  
//działamy na i  
end
```

- Wyrażenie '//działamy na i' jest rozdzielane pomiędzy użytkowników MATLABA (matlabpool)
  - Jeżeli nie ma żadnych użytkowników, zachowuje się jak zwykła pętla for
- Można używać dla regularnych sesji MATLAB-a
- Analiza "M-Lint" pomaga zlokalizować istniejące pętle for, które mogą zostać zamienione na parfor

# Pętle a równoległość

```
parfor i=1 : n  
//działamy na i  
end
```

- Wyrażenie '//działamy na i' jest rozdzielane pomiędzy użytkowników MATLABA (matlabpool)
  - Jeżeli nie ma żadnych użytkowników, zachowuje się jak zwykła pętla for
- Można używać dla regularnych sesji MATLAB-a
- Analiza "M-Lint" pomaga zlokalizować istniejące pętle for, które mogą zostać zamienione na parfor

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone

- Przechowują dane w segmentach pracowników biorących udział w projekcie
- Tworzone są, przez każdą wbudowaną klasę MATLABA
  - Przykład: podwajanie, tablice rzadkie, komórki tablic

- Algorytmy równoległe dla tablic rozproszonych

- Operacje macierzowe
  - Przykład: indeksowanie, konwersja typów danych, transpozycja
- Równoległe rozwiązywanie układów równań liniowych (svd, lu)
- Ponad 150 dostępnych funkcji MATLABA
- Dystrybucja danych
  - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas



# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

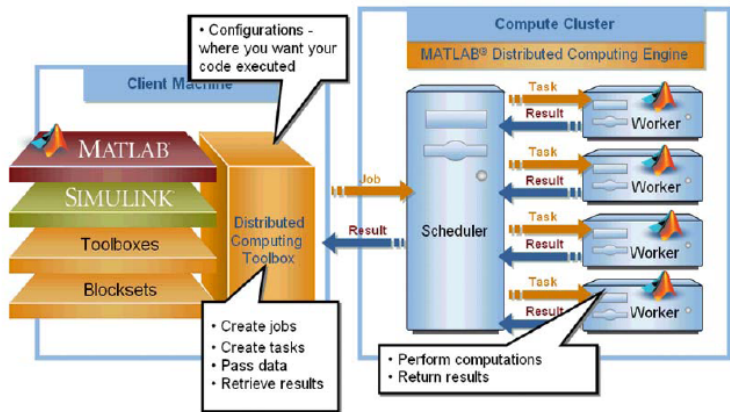
- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Tablice rozproszone i algorytmy równoległe

- Tablice rozproszone
  - Przechowują dane w segmentach pracowników biorących udział w projekcie
  - Tworzone są, przez każdą wbudowaną klasę MATLABA
    - Przykład: podwajanie, tablice rzadkie, komórki tablic
- Algorytmy równoległe dla tablic rozproszonych
  - Operacje macierzowe
    - Przykład: indeksowanie, konwersja typów danych, transpozycja
  - Równoległe rozwiązywanie układów równań liniowych (svd, lu)
  - Ponad 150 dostępnych funkcji MATLABA
  - Dystrybucja danych
    - Automatyczna, zdefiniowana przez użytkownika, zmienna cały czas

# Schemat działania

## Batch Execution



# Pisanie kodu równoległego

No code changes

- **Other toolboxes:**
  - Optimization Toolbox™
  - Genetic Algorithm and Direct Search Toolbox™
  - SystemTest™

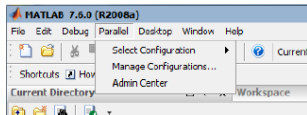
Trivial changes

- `parfor`
- **distributed arrays**
- **jobs and tasks**
  
- **MATLAB® message passing**

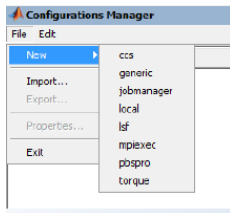


# Konfiguracja MATLABA dla systemu wsadowego - 1/3

- Otwieramy zakładkę Parallel a następnie Manage Configurations...



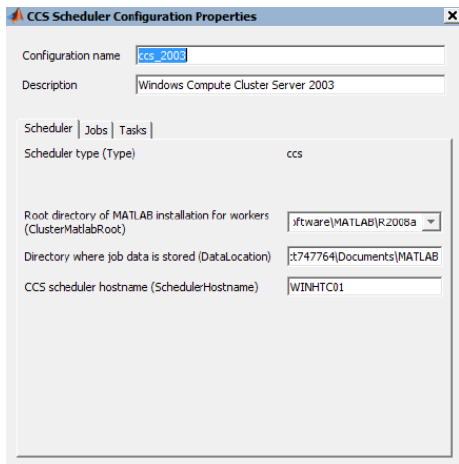
- W Configurations Manager wybieramy Select File - New - ccs



- CCS-Compute Cluster Server

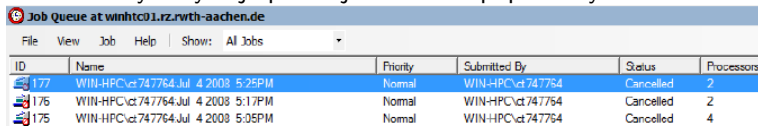
# Konfiguracja MATLABA dla systemu wsadowego - 2/3

- Wybieramy dowolną nazwę konfiguracyjną oraz opis, np. ccs 2003
- Ustawiamy katalog główny - `//cifs/cluster/software/MATLAB/R2008a`
- Ustawiamy katalog z danymi - `//cifs/cluster/profile/...` lub `/cifs/cluster/home/...`
- Ustawiamy CCS scheduler: WINHTC01



# Konfiguracja MATLABA dla systemu wsadowego - 3/3

- Możliwość dwóch konfiguracji
  - local: używa wielordzeniowej możliwości naszej maszyny
  - ccs 2003: wykorzystuje potencjał klastrów poprzez System



The screenshot shows a window titled "Job Queue at winhpc01.rz.rwth-aachen.de". It has a menu bar with "File", "View", "Job", and "Help", and a "Show: All Jobs" dropdown. Below is a table with the following data:

ID	Name	Priority	Submitted By	Status	Processors
177	WIN-HPC\ct747764.Jul 4 2008 5:25PM	Normal	WIN-HPC\ct747764	Cancelled	2
176	WIN-HPC\ct747764.Jul 4 2008 5:17PM	Normal	WIN-HPC\ct747764	Cancelled	2
175	WIN-HPC\ct747764.Jul 4 2008 5:05PM	Normal	WIN-HPC\ct747764	Cancelled	4

- Użycie równoległych możliwości MATLABA
  - matlabpool, open, configuration, size
  - pmode, open, configuration, size
- Ograniczenia
  - local: maksymalnie 4 procesy (silniki przetwarzania)
  - klaster (ccs 2003): maksymalnie 8 procesów (silników przetwarzania), ze względu na ograniczenia licencji

## Przykład - pętla parfor

Prosty kod, który może być wykonywany na  $n$  pracownikach.

```
N=60;  
a=zeros(N, 1);  
tic  
parfor (i = 1:N)  
    a(i) = max(abs(eig(rand(300))))  
end  
toc
```

## Przykład nr.2

`%funkcja gradientów sprzężonych`

`function CG`

```
n = 1400; nonzer = 7; lambda = 20; niter = 15;  
nz = n * (nonzer + 1) * (nonzer + 1) + n * (nonzer + 2);
```

`%tworzymy losową macierz rzadką`

```
A=sprand(n, n, 0.5 * nz/n2, distributor());
```

```
A = 0.5 * (A + A');
```

```
I = speye(n, distributor())
```

```
A = A - lambda * I; %przesunięcie lambda, przybliżenie wartości własnej
```

```
x = ones(n, 1); % inicjacja 1-kami
```

```
for iter = 1:niter
```

```
[z, rnorm] = cgit(A, x);
```

```
zeta = lambda + 1/(x' * z);
```

```
x = z/norm(z);
```

```
fprintf('%5d %13.4e %20.13f', iter, rnorm, zeta);
```

```
end
```

## Powiązanie zadań wsadowych - 1/3

- Ustawienie konfiguracji dla naszego Klastra

```
sched = findResource('scheduler', 'configuration', 'ccs 2003')
```

- Stworzenie i uruchomienie rozproszonego zadania

```
job = createJob(sched);  
createTask(job, @sum, 1, [1 1]);  
createTask(job, @sum, 1, [2 2]);  
createTask(job, @sum, 1, [3 3]);  
submit(job);  
waitForState(job, 'finished', 60);
```

- Zapisywanie wyników zadania

```
results = getAllOutputArguments(job);
```

- Wynik powinien wyglądać tak:

```
results=
```

```
[2]  
[4]  
[6]
```

## Powiązanie zadań wsadowych - 2/3

- Ustawienie konfiguracji dla naszego Klastra

```
sched = findResource('scheduler', 'configuration', 'ccs 2003')
```

- Stworzenie i uruchomienie zadania równoległego

```
job = createParallelJob(sched);  
createTask(job, @labindex, 1, );  
set(job, 'MinimumNumberOfWorkers', 3);  
set(job, 'MaximumNumberOfWorkers', 3);  
submit(job);  
waitForState(job, 'finished', 60);
```

- Zapisywanie wyników zadania

```
results = getAllOutputArguments(job);
```

- Wynik powinien wyglądać tak:

```
results=
```

```
[1]  
[2]  
[3]
```

## Powiązanie zadań wsadowych - 3/3

- Stworzenie i uruchomienie Pool Job oraz ustawienie konfiguracji

```
job = createMatlabPoolJob(,configuration', ,ccs 2003');  
createTask(job, @sum, 1, [1 1]);  
set(job, ,MinimumNumberOfWorkers', 3);  
set(job, ,MaximumNumberOfWorkers', 3);  
submit(job);  
waitForState(job, ,finished', 60);
```

- Zapisywanie wyników zadania  
results = getAllOutputArguments(job);

- Wynik powinien wyglądać tak:

```
results=  
    [2]  
    []  
    []
```



# Bibliografia

- [http://en.wikipedia.org/wiki/High-performance\\_computing](http://en.wikipedia.org/wiki/High-performance_computing)
- <http://www.mathworks.com/>
- <http://www.rz.rwth-aachen.de>