

# Pakiet Deino MPI jako środowiska programowania równoległego – Instalacja i konfiguracja pakietu

Jędrzej Ułasiewicz

Wrocław 2010

# 1. PAKIET DEINO MPI JAKO ŚRODOWISKA PROGRAMOWANIA RÓWNOLEGŁEGO

Deino MPI jest oprogramowaniem warstwy pośredniej (ang. *middleware*) umożliwiającym wykonywanie programów równoległych na klastrze składającym się z komputerów połączonych siecią. Pakiet ten jest implementacją standardu MPI-2 dla komputerów pracujących pod kontrolą systemu operacyjnego Microsoft Windows XP, Vista, 7. W dalszej części została opisana instalacja i konfiguracja pakietu DeinoMPI w wersji 2.0.1 dla przykładowego klastra złożonego z 2 komputerów 4 rdzeniowych pracujących pod kontrolą systemu operacyjnego Windows XP. W przypadku innych konfiguracji sprzętowych i innych wersji systemu Windows postępowanie będzie podobne. Proces instalacji i konfiguracji klastra opisany jest w **Błąd! Nie można odnaleźć źródła odsyłacza.**

## 1.1 Instalacja i konfiguracja pakietu

### 1.1.1 DZIAŁANIA WSTĘPNE

Przed przystąpieniem do instalacji klastra należy zadbać aby spełnione były podane niżej warunki wstępne. Pozwoli to uniknąć sytuacji gdy proces instalacji utknie gdzieś w połowie na skutek braku spełnienia wymaganych warunków.

Warunkiem instalacji pakietu DeinoMPI jest:

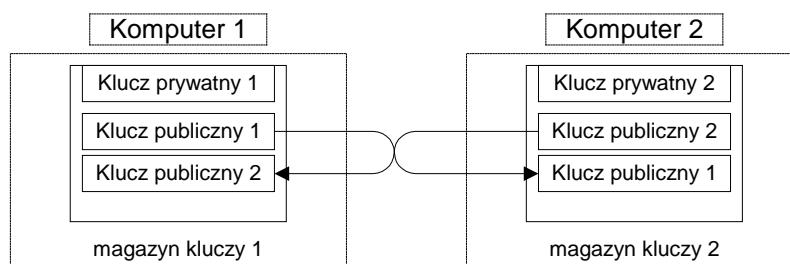
1. Posiadanie komputerów wyposażonych w system operacyjny Windows XP/Server/ Vista/7 z zainstalowanym uaktualnieniem Service Pack3.
2. Komputery wchodzące w skład klastra powinny posiadać sprawnie działające połączenie sieciowe o możliwie dużej przepustowości.
3. W komputerach powinien być zainstalowany pakiet .NET Framework 2.0 (albo późniejszy). Autor zainstalował pakiet Microsoft .NET Framework 3.5
4. W komputerach powinien być zainstalowany pakiet Visual C++ 2005 Service Pack 1 Redistributable Package ATL Security Update.

Przed przystąpieniem do instalacji należy:

1. Sprawdzić czy komputery mają prawidłowo ustawioną nazwę i jednakową grupę roboczą. Można to zrobić wybierając w Panelu Sterowania opcję **System** i zakładkę **Nazwa komputera**. W instalacji przykładowej użyto komputerów o nazwach KOREK i ROMA z grupy roboczej LAB019A.
2. Na każdym z komputerów utworzyć użytkownika posiadającego tę samą nazwę i przywileje administratora. W instalacji przykładowej był to użytkownik `juka`. Użytkownik musi mieć niepuste hasło.
3. Wyinstalować z komputera poprzednie wersje pakietu o ile takie były zainstalowane.
4. Sprowadzić z strony <http://mpi.deino.net> odpowiednią wersję pakietu ( tutaj użyto DeinoMPI.2.0.1.msi dla systemu 32 bitowego lub DeinoMPI.x64.2.0.1.msi dla systemu 64 bitowego) i zapisać go w folderze roboczym.

Istotną funkcją systemu jest możliwość zdalnego uruchamianie programów na poszczególnych węzłach klastra. Z punktu widzenia lokalnego systemu operacyjnego, pozwolenie aby zdalny komputer uruchamiał lokalne procesy, jest bardzo niebezpieczną funkcjonalnością. Dla zapewnienia bezpieczeństwa system DeinoMPI (podobnie jak inne implementacje MPI) używa mechanizmu szyfrowania niesymetrycznego - kluczy prywatnych i publicznych. Dla każdego z komputerów wchodzących w skład klastra generowany jest klucz prywatny i publiczny. Następnie klucze publiczne przenoszone są dowolną metodą na pozostałe komputery klastra i dołączane do systemu szyfrującego. Przykładowe postępowanie dla klastra składającego się z dwóch komputerów (komputer 1 i komputer 2) pokazuje Rys. 1-1. Każdy z komputerów posiada klucz prywatny i publiczny. Klucz publiczny komputera 1 kopiowany jest na komputer 2 i dołączany do magazynu

kluczy 2 a klucz publiczny komputera 2 kopiowany jest na komputer 1 i dołączany do magazynu kluczy 1. W przypadku większej liczby komputerów postępowanie jest analogiczne.



Rys. 1-1 Konfiguracja kluczy w komputerach klastra

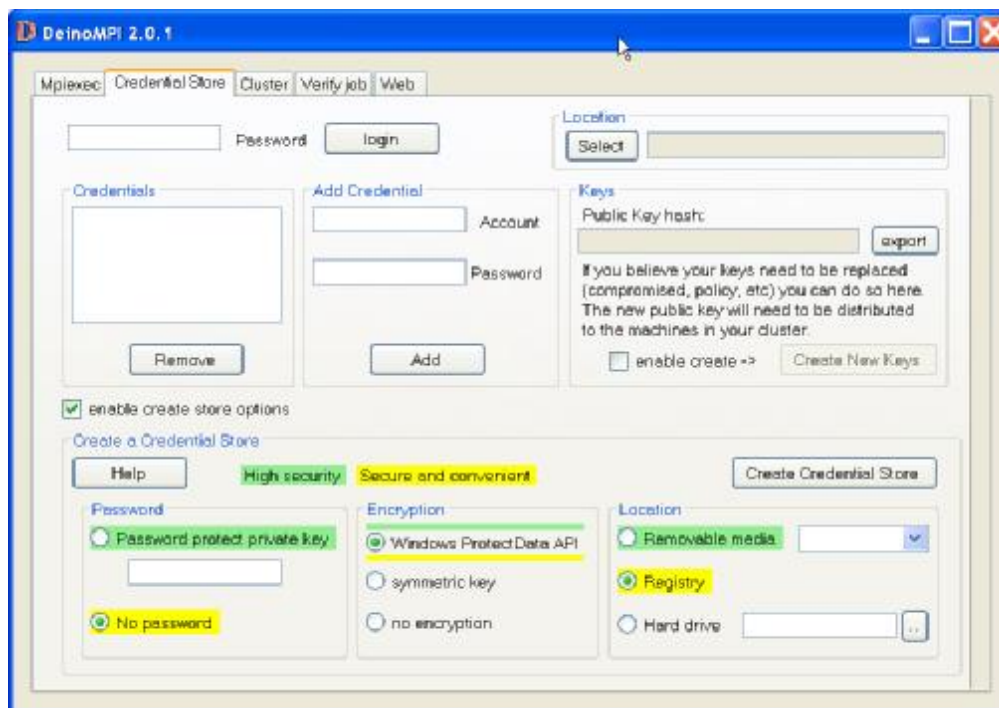
### 1.1.2 INSTALACJA I KONFIGURACJA KLASTRA

Gdy spełnione są warunki wstępne można przystąpić do procesu instalacji oprogramowania klastra. Przebieg procesu instalacji składa się z podanych poniżej kroków.

1. Należy uruchomić proces instalacji klikając na plik instalacyjny DeinoMPI.2.0.1.msi
2. Następnie należy wcisnąć klawisz Next, zaakceptować umowę licencyjną i kontynuować instalację.
3. Z zaakceptować domyślną lokalizację pakietu (C:\Program Files\DeinoMPI) lub też wybrać inną.
4. Zakończyć instalację wciskając przycisk Finish.
5. Uruchomić interfejs graficzny DeinoMPIwin z belki programów lub bezpośrednio. Powinno pojawić się okno pokazane na Rys. 1-2.
6. Następnie należy skonfigurować mechanizm zabezpieczeń (klucze prywatne i publiczne). Można to zrobić z korzystając z interfejsu graficznego (zakładka Credential Store). Jednak Autor wykonał tę pracę wsadowo korzystając z programów pokazanych w Tab. 1-1. Aby kontynuować konfigurowanie należy uruchomić interpreter poleceń (Start / uruchom / cmd) i przejść do foldera C:\Program Files\DeinoMPI\bin. W folderze tym znajdują się programy do zarządzania kluczami pokazane w Tab. 1-1.

create_credential_store	Tworzenie magazynu kluczy
manage_credentials	Zarządzanie magazynem kluczy
manage_public_keys	Zarządzanie kluczami publicznymi

Tab. 1-1 Programy do zarządzania kluczami pakietu DeinoMPI



Rys. 1-2 Widok interfejsu graficznego (konsoli) pakietu DeinoMPI

7. Utworzyć nowy magazyn kluczy. Magazyn kluczy tworzy się pisząc na konsoli:  
`create_credentials_store`

Program wypisze zapytanie:

Do you want to create a new store <yes, no> ? [yes]

Należy wcisnąć Enter lub napisać yes.

Następnie program wypisze komunikat:

Please enter a passphrase to protect the private key:

Należy nacisnąć Enter. Wtedy program, jakby niedowierzając nam wypisze ponowne zapytanie:

Are you sure you want no password for the private key  
 <yes,no>? [no]

Należy wpisać yes. Następnie program wyprowadzi zapytanie czy szyfrować klucz prywatny.

Please enter the method to encrypt the private key:  
 [protect]

Należy wcisnąć Enter decydując się na szyfrowanie klucza. W końcu program spyta o miejsce przechowywania magazynu kluczy dając wybór:

(1) Removable media

(2) Windows Registry

(3) Hard drive

Należy wybrać opcję 2. po czym program wyprowadzi komunikat:

Credential store written to Windows Registry.

Opisane wyżej postępowanie pokazuje Rys. 1-3.

```

c:\Program Files\DeinoMPI\bin>create_credential_store

This program creates a Credential Store for the current user.
The credential store contains a public and private security key
and zero or more secure user credentials.
Deino MPI uses this store to establish secure connections between machines
and store sensitive information like user credentials.

Do you want to create a new store (yes, no)? [yes] yes
Please enter a passphrase to protect the private key:
Are you sure you want no password for the private key (yes,no)? [no]
yes

Options for encrypting the private key:
protect - Use the ProtectedData.Protect/Unprotect methods to encrypt.
none - Do not encrypt.
Please enter a method to encrypt the private key: [protect]

Available locations to create the store:
(1) Removable media
(2) Windows Registry
(3) Hard drive
Where would you like to create the store (1,2,3): [1] 2

Credential store written to the Windows Registry.

c:\Program Files\DeinoMPI\bin>

```

Rys. 1-3 Utworzenie magazynu kluczy

8. Dalej należy powiązać magazyn kluczy z użytkownikiem. Dokonuje się tego pisząc na konsoli:

```
manage_credentials /add
```

Program zapyta o nazwę komputera:

```
Enter the account name: [KOREK\juka]
```

Należy zatwierdzić propozycję lub podać właściwą nazwę użytkownika. Następnie program zapyta o podanie hasła:

```
Enter the account password:
```

```
*****
```

```
Enter the password again to verify
```

```
*****
```

Należy podać właściwe hasło (takie jak przy logowaniu).

Po podaniu hasła i jego powtórzeniu program zakończy działanie.

9. Zapisać klucz publiczny w zewnętrznym pliku

Klucz publiczny komputera KOREK musi być udostępniony innym komputerom (w tym komputerowi ROMA) w tym celu należy zapisać go w postaci pliku tekstowego ( w tym przypadku pliku korek\_pub.txt). Dokonujemy tego za pomocą programu `manage_public_keys` wywołanego z opcją `/export`. Jako drugi parametr należy podać nazwę pliku do którego zapisany ma być klucz publiczny. Tak więc na konsoli należy napisać:

```
manage_public_keys /export korek_pub.txt
```

Program odpowie komunikatem:

```
Public key written to file:
```

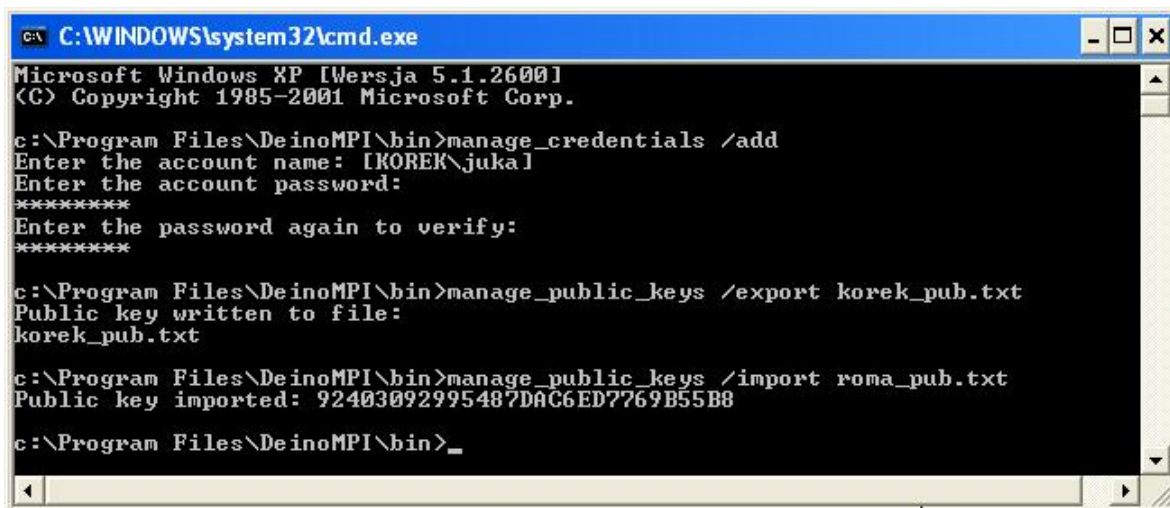
```
korek_pub.txt
```

Uzyskany plik z kluczem publicznym należy dostarczyć do pozostałych komputerów i zapisać w ich magazynach kluczy. W naszym przykładzie plik należy skopiować do katalogu `C:\Program Files\DeinoMPI\bin` komputera ROMA..

10. Następnie należy dokonać importu kluczy publicznych z innych komputerów klastra. System bezpieczeństwa komputera lokalnego musi znać klucze publiczne innych komputerów klastra. Przekazanie takiej informacji następuje poprzez import kluczy prywatnych innych komputerów. Importu dokonuje się poprzez uruchomienie programu `manage_public_keys` z kluczem `/import` podając jako parametr nazwę pliku zawierającego klucz publiczny innego komputera. W rozpatrywanym przykładzie jest to plik `roma_pub.txt` zawierający klucz publiczny komputera ROMA.

```
manage_public_keys /import roma_pub.txt
```

Program odpowie odpowiednim komunikatem co pokazuje Rys. 1-4.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Wersja 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\Program Files\DeinoMPI\bin>manage_credentials /add
Enter the account name: [KOREK\juka]
Enter the account password:
*****
Enter the password again to verify:
*****

c:\Program Files\DeinoMPI\bin>manage_public_keys /export korek_pub.txt
Public key written to file:
korek_pub.txt

c:\Program Files\DeinoMPI\bin>manage_public_keys /import roma_pub.txt
Public key imported: 92403092995487DAC6ED7769B55B8

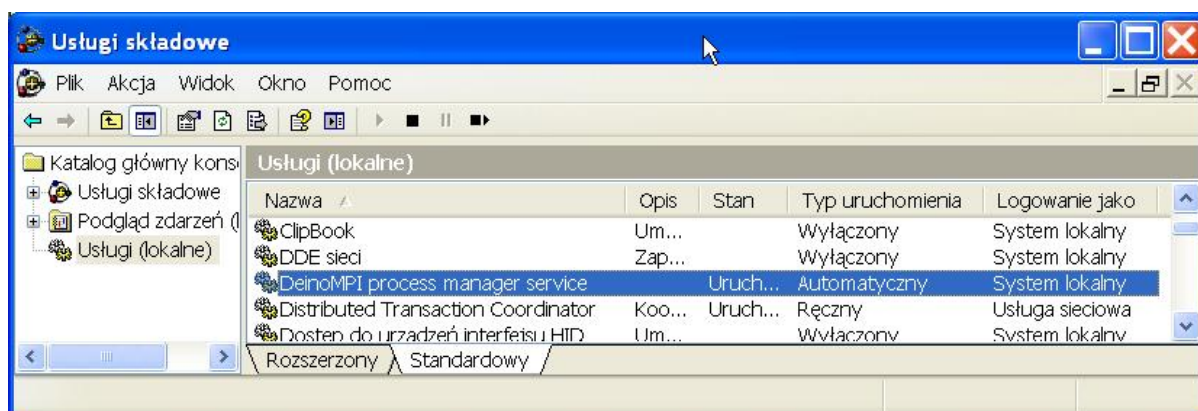
c:\Program Files\DeinoMPI\bin>_

```

Rys. 1-4 Eksport i import kluczy publicznych

11 W kolejności należy powtórzyć powyższe postępowanie opisane w punktach 1 – 10 na wszystkich węzłach klastra.

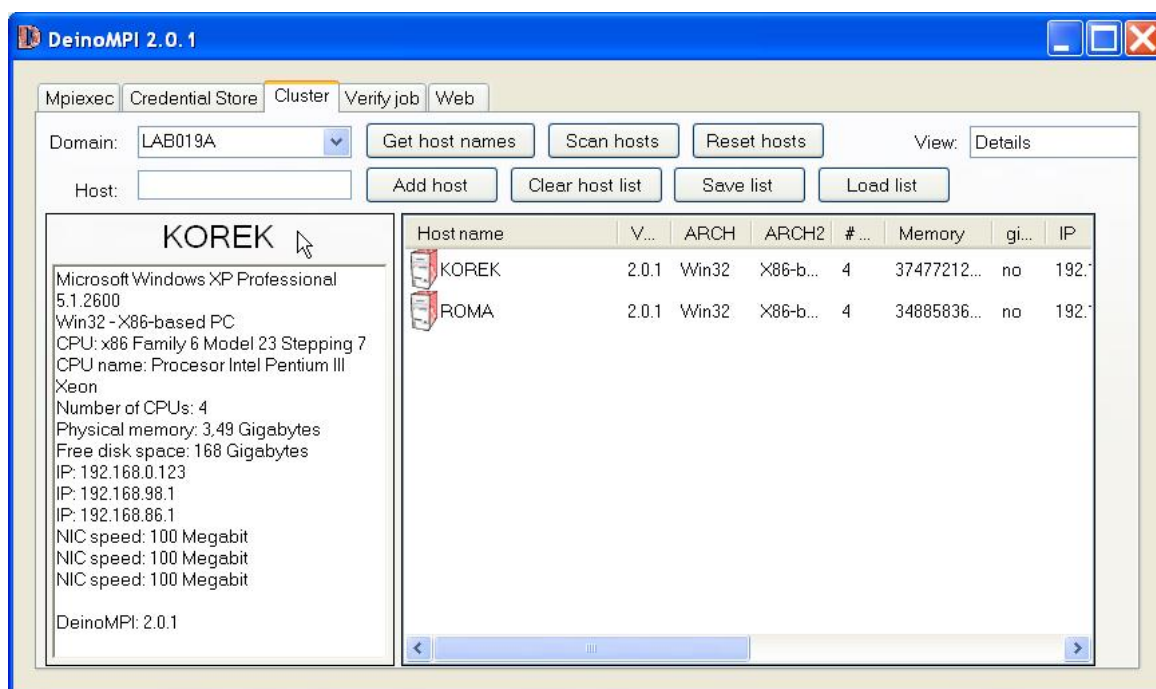
Po zainstalowaniu oprogramowania program `DeinoPM.exe` będzie uruchomiony jako usługa systemu Windows. Program ten pełni funkcję zarządzania klastem i umożliwia uruchamianie procesów ze zdalnego węzła. Można dokonać sprawdzenia prawidłowości jego uruchomienia wybierając okno Panel sterowania / Narzędzia administracyjne / Usługi składowe. Program `DeinoPM` będzie widoczny jako `Deino MPI process manager service` co pokazuje Rys. 1-5.



Rys. 1-5 Proces zarządzania klastrem DeinoMPI jako usługą systemu Windows.

### 1.1.3 SPRAWDZENIE KONFIGURACJI KLASTRA

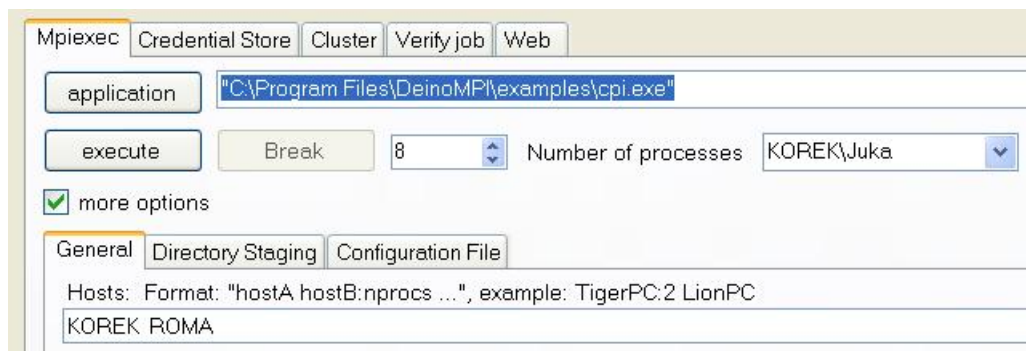
Aby sprawdzić czy węzły klastra są widoczne należy na jednym z komputerów uruchomić interfejs graficzny klastra (konsolę) i przejść do zakładki `Cluster` tak jak pokazano na Rys. 1-6. Następnie w polu `Domain` należy ustawić nazwę grupy roboczej do której należy dany komputer i pozostałe komputery klastra. W naszym przykładzie jest to `LAB019A`. Dalej należy wcisnąć przycisk `Get host names` po naciśnięciu którego powinny się pojawić nazwy komputerów wchodzących w skład grupy roboczej klastra (odpowiadające im ikony będą w szarym kolorze). Następnie należy wcisnąć przycisk `Scan hosts` w skutek czego powinny się pojawić ikony węzłów klastra w czerwonym kolorze tak jak pokazuje to Rys. 1-6. Jak widać przykładowy klasterek składa się z dwóch komputerów połączonych siecią Fast Ethernet 100 i wyposażonych w 4 rdzeniowe procesory Xeon.



Rys. 1-6 Konsola klastra

### 1.1.4 URUCHOMIENIE PROGRAMU PRZYKŁADOWEGO.

Kolejnym krokiem mającym na celu sprawdzenie działania klastra jest uruchomienie przykładowego programu testowego dostarczanego wraz z pakietem. Będzie to program `cpi.exe` umieszczony w folderze `C:\Program Files\DeinoMPI\examples`. Program ten w sposób równoległy oblicza liczbę  $\pi$ . Aby uruchomić program należy w konsoli wybrać zakładkę `Mpiexec`, w polu `application` wybrać program `C:\Program Files\DeinoMPI\examples\cpi`, w polu `Number of processes` wybrać liczbę procesów (powinna ona odpowiadać sumarycznej liczbie procesorów w klastrze) a w polu `Hosts` wpisać komputery wchodzące w skład klastra. W naszym przykładzie będą to komputery `KOREK` i `ROMA`. Następnie należy kliknąć w klawisz `Execute`. W odpowiedzi w dolnym oknie powinny się pojawić komunikaty procesów składowych aplikacji tak jak pokazano na Rys. 1-7.



Rys. 1-7 Przykład obliczania liczby  $\pi$  na klastrze.

Prawidłowy wynik działania programu pokazuje Przykład 1-1. Jak widać procesy zostały uruchomione na wszystkich węzłach klastra.

```
Process 5 of 8 is on roma
Process 2 of 8 is on roma
Process 3 of 8 is on korek
Process 1 of 8 is on korek
Process 6 of 8 is on korek
Process 0 of 8 is on korek
pi is approximately 3.1415926544231247, Error is 0.000000008333316
wall clock time = 0.007503
Process 4 of 8 is on roma
Process 7 of 8 is on roma
```

Przykład 1-1 Wynik działania programu cpi

## 1.2 Kompilacja programów w środowisku Microsoft Visual Studio 2008 i ich uruchamianie w środowisku MPI

Pakiet DeinoMPI umożliwia tworzenie programów równoległych za pomocą różnych kompilatorów. Wymienić tutaj można Microsoft Visual Studio 2008, Dev C++, Borland C++ Builder i wiele innych podanych w [2].

W ogólności można powiedzieć że aby skompilować program MPI należy przekazać środowisku programistycznemu informacje o:

- Położeniu plików nagłówkowych (standardowo C:\Program Files\DeinoMPI\include)
- Położeniu bibliotek (standardowo C:\Program Files\DeinoMPI\lib)

Jako że nie ma tu możliwości opisania wszystkich tych narzędzi skoncentrujemy się na platformie Microsoft Visual Studio 2008 gdyż jest to najbardziej zaawansowane środowisko programistyczne dla systemu Windows. Jako że prawidłowa konfiguracja wielu opcji projektu jest trudna, to najpierw wykorzystamy umieszczony w folderze C:\Program Files\DeinoMPI\examples uruchamiany już uprzednio przykład cpi dotyczący obliczania liczby  $\pi$ .

### 1.2.1 KOMPILACJA PRZYKŁADU

Pierwszym krokiem jest uruchomienie środowiska Visual Studio 2008 i wybranie projektu dostarczonego z pakietem programu cpi. Jak wspomniano pakiet ten służy do obliczania liczby  $\pi$  i jest umieszczony w folderze C:\Program Files\DeinoMPI\examples. Wczytujemy projekt do środowiska za pomocą kolejnych wyborów: File / Open / Project/Solution i wybór pliku cpi.vcproj. System zaproponuje przekształcenie projektu do aktualnej postaci, na co należy się zgodzić. Następnie należy



skompilować program wybierając opcje: Build / Build Solution. Program wynikowy `mpi.exe` umieszczony zostanie w katalogu `C:\Program Files\DeinoMPI\examples\Debug`.

### 1.2.2 URUCHOMIENIE PRZYKŁADU

System DeinoMPI podobnie jak inne implementacje MPI, zasadniczo wymaga aby programy wykonywalne umieszczone były we wszystkich węzłach klastra w folderze o tej samej nazwie. Toteż można utworzyć oddzielny folder (powiedzmy `C:\Wspólny`) udostępnić go innym węzłom klastra i za jego pośrednictwem skopiować program wykonywalny do katalogów `C:\Wspólny` na poszczególnych węzłach. Następnie należy uruchomić program `mpi.exe` za pomocą konsoli MPI tak jak pokazano na (należy podać właściwą ścieżkę). Gdy wszystko pójdzie dobrze powinniśmy uzyskać wynik jak podaje Przykład 1-1.

### 1.2.3 KOMPILACJA PROGRAMU HELLO

Przed przystąpieniem do bardziej złożonych przykładów pokażemy jak samodzielnie skompilować i uruchomić w systemie MPI program `hello.c` przy pomocy środowiska Visual Studio 2008. Kod programu podaje Przykład 1-2. Aby uniknąć konieczności ustawiania dużej liczby opcji i parametrów środowiska wykorzystamy przykładowy projekt `mpi` i dokonamy w nim niezbędnych modyfikacji.

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>

int main(int argc, char *argv[]) {
    int    myid, numprocs;
    int    namelen;
    char   maszyna[MPI_MAX_PROCESSOR_NAME];
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    MPI_Get_processor_name(maszyna, &namelen);
    fprintf(stdout, "Proces 2 %d z %d wykonywany na komp
    %s\n", myid, numprocs, maszyna);
    fflush(stdout);
    MPI_Finalize();
    return 0;
}
```

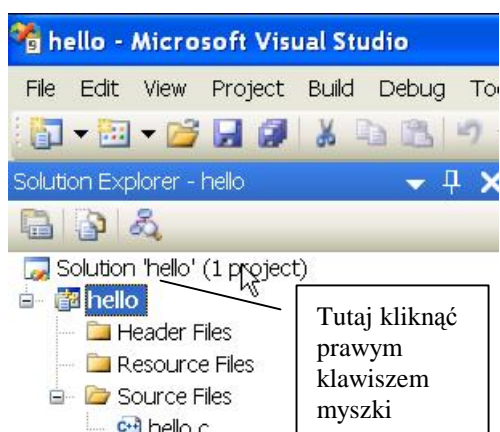
#### Przykład 1-2 Program `hello.c` dla środowiska MPI

Aby skompilować program `hello.c` w środowisku Visual Studio 2008 należy:

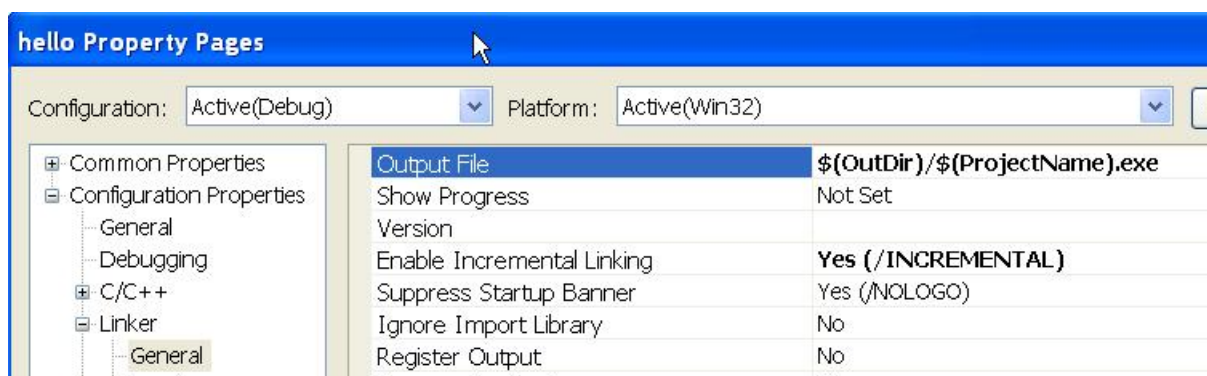
1. Z katalogu `C:\Program Files\DeinoMPI\examples` do nowego katalogu, powiedzmy `C:\wspolny\hello`, skopiować pliki `mpi.c` i `mpi.vcproj`. Zmienić nazwę pliku `mpi.vcproj` na `hello.vcproj`.
2. Za pomocą dowolnego edytora otworzyć plik `hello.vcproj` i zmienić linię `Name="mpi"` na linię `Name="hello"`. Zachować zmieniony plik.
3. Otworzyć nowy projekt w środowisku Visual Studio za pomocą opcji `File / Open / Project/Solution` i wybór pliku `hello.vcproj`.
4. Zmienić nazwę pliku `mpi.c` na `hello.c`. Dokonać tego można poprzez kliknięcie prawym klawiszem myszy na ikonę pliku `mpi.c` i wybór opcji `Rename`.
5. Zmienić nazwę pliku wykonywalnego. Dokonać tego można klikając prawym klawiszem myszy na ikonę nazwy projektu (w tym przypadku `hello`) tak jak pokazuje Rys. 1-8. W skutek tego pojawi się lista wyboru z której wybrać należy

opcję Property. Pojawi się okno Property Pages w którym należy zmodyfikować ustawienie Configuration Properties / Linker / General ustawiając ją na wartość  $\$(OutDir)/\$(ProjectName).exe$  tak jak pokazuje to Rys. 1-9.

6. Następnie należy wykonać wymianę zawartości pliku `hello.c` na nową (pokazaną w Przykład 1-2) poprzez opcję kopiuj/wklej lub poprzez skopiowanie pliku .
7. Dokonać kompilacji nowego projektu poprzez wybór opcji Build / Build Solution. System zaproponuje utworzenie nowego pliku `hello.sln` na co należy się zgodzić. Nowy plik wykonywalny `hello.exe` zostanie umieszczony w folderze Debug. Uruchomienie programu opisane zostanie dalej.



Rys. 1-8 Zmiana nazwy pliku wykonywalnego etap I



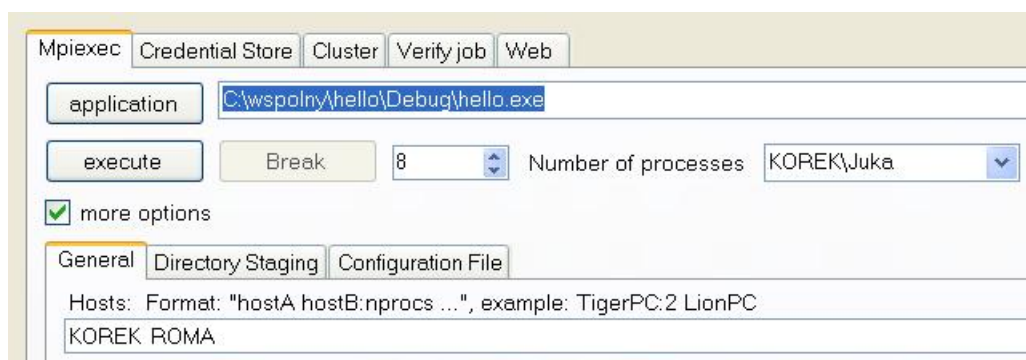
Rys. 1-9 Zmiana nazwy pliku wykonywalnego etap II

#### 1.2.4 URUCHAMIANIE PROGRAMÓW Z KONSOLI GRAFICZNEJ

Programy MPI uruchamiane mogą być za pomocą interfejsu graficznego systemu tak jak pokazano na lub z konsoli tekstowej. Aby programy dały się uruchomić na wszystkich węzłach klastra, powinny być one skopiowane do folderów o tych samych nazwach umieszczonych na wszystkich węzłach klastra. W naszym przykładzie program `hello` znajduje się w folderze `C:\wspolny\hello\Debug`. Aby skopiować go na pozostałe węzły klastra należy:

1. Na każdym z węzłów utworzyć folder `C:\wspolny`.
2. Na każdym z węzłów klastra udostępnić ten folder w sieci. Odbywa się to poprzez kliknięcie prawym klawiszem myszy w ikonę tego folderu i wybór opcji

- Udostępnianie i zabezpieczenia. Należy zezwolić użytkownikom sieciowym na zapis do tego folderu i określić nazwę udziału jako `wspolny`.
- Na komputerze głównym wykonać mapowanie udostępnionych folderów jako dyski sieciowe. Można to wykonać włączając Eksplorator Windows następnie przeszukać sieć, odnaleźć węzły klastra i mapować ich udostępnione katalogi na dyski sieciowe. W tym celu należy kliknąć prawym klawiszem myszy na ikonę udostępnionego folderu `wspolny` i wybrać opcję **Mapuj dysk sieciowy**. Należy następnie wybrać nazwę dysku sieciowego (w przykładzie jest to dysk `Z:`)
  - Do zmapowanych dysków sieciowych, do odpowiedniego folderu, skopiować program wykonywalny z folderu `C:\wspolny` komputera głównego. W przykładzie folderem docelowym jest folder `Z:\hello\Debug`.
  - W interfejsie graficznym wybrać nazwę aplikacji jako `C:\wspolny\Debug\hello`, wpisać żadaną liczbę procesów (w przykładzie 8) a w oknie **Hosts** wpisać nazwy komputerów klastra (w przykładzie `KOREK ROMA`) co pokazuje Rys. 1-10.
  - Naciśnięcie przycisku **Execute** spowoduje wykonanie programu `hello`. Wyniki pokazuje Przykład 1-3.



Rys. 1-10 Wykonanie programu hello na klastrze

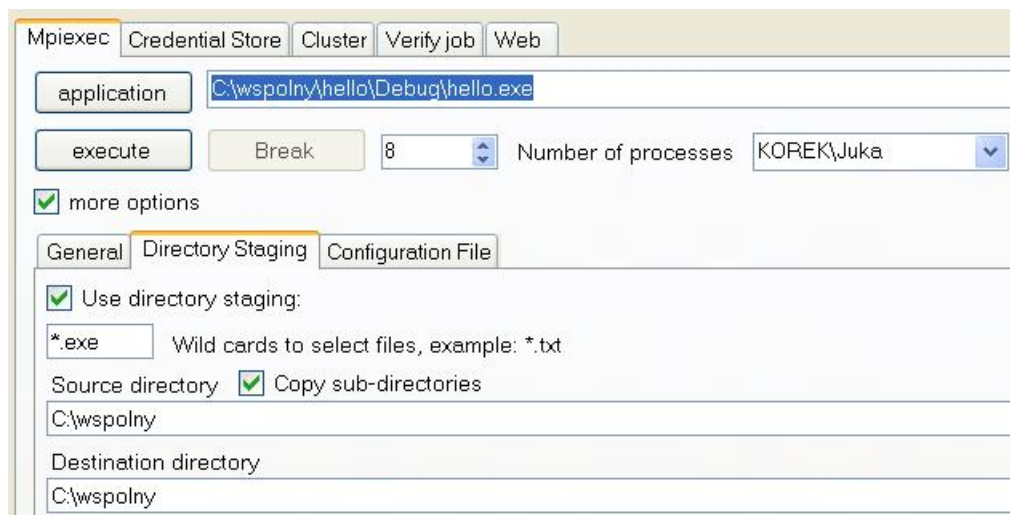
```

Proces 2 2 z 8 wykonywany na komp roma
Proces 2 5 z 8 wykonywany na komp roma
Proces 2 6 z 8 wykonywany na komp roma
Proces 2 7 z 8 wykonywany na komp roma
Proces 2 4 z 8 wykonywany na komp korek
Proces 2 1 z 8 wykonywany na komp korek
Proces 2 3 z 8 wykonywany na komp korek
Proces 2 0 z 8 wykonywany na komp korek

```

Przykład 1-3 Wyniki działania programu hello dla środowiska MPI

Opisany wyżej sposób dystrybucji programów na węzły klastra jest niewygodny. Środowisko DeinoMPI umożliwia automatyczne kopiowanie programów na poszczególne węzły klastra. Można to osiągnąć wybierając w konsoli graficznej zakładkę **Directory Staging** tak jak pokazano na Rys. 1-11 Automatyczne kopiowanie programów wykonywalnych na węzły klastra. Następnie należy oznaczyć pole wyboru **Use directory staging**, w polu **Source directory** wpisać nazwę folderu źródłowego (w przykładzie `C:\wspolny`) a w polu **Destination directory** (w przykładzie `C:\wspolny`). Należy także odznaczyć pole **Copy sub-directories**. Następnie należy wybrać aplikację i nacisnąć przycisk **Execute**. System automatycznie skopiuje potrzebne pliki z foldera źródłowego komputera głównego do folderów docelowych pozostałych węzłów klastra i uruchomi te programy.



Rys. 1-11 Automatyczne kopiowanie programów wykonywalnych na węzły klastra

### 1.2.5 URUCHAMIANIE PROGRAMÓW Z KONSOLI TEKSTOWEJ

Procesy aplikacyjne mogą być również uruchamiane z konsoli tekstowej przy użyciu programu mpiexec. Aby wygodniej było program ten uruchamiać należy dodać ścieżkę do folderu C:\Program Files\DeinoMPI\bin do ścieżki systemowej Path. Można to zrobić uruchamiając Panel Sterowania i wybierając opcje System / Zaawansowane / Zmienne środowiskowe i edytując zmienną środowiska Path. Do istniejącej ścieżki należy dopisać ;C:\Program Files\DeinoMPI\bin. Program mpiexec posiada szereg opcji opisanych w **Błąd! Nie można odnaleźć źródła odsyłacza.** Jednak najistotniejsza jest możliwość uruchomienia procesów na zadanych węzłach klastra. Wykonuje się to podając opcję hosts. Polecenie to ma wtedy postać:

```
mpiexec -hosts liczba_wezlow n1 p1 n2 p2 ... nk pk nazwa_prog
```

Gdzie:  $n_i$  – nazwa węzła  $i$ ,

$p_i$  – liczba procesów uruchomionych na węźle  $i$

nazwa\_prog – nazwa programu aplikacyjnego

Przykładowe uruchomienie 8 procesów hello – 4 na węźle KOREK i 4 na węźle ROMA pokazuje Przykład 1-4.

```
mpiexec -hosts 2 korek 4 roma 4 c:\wspolny\hello\Debug\hello
Proces 4 z 8 wykonywany na komp roma
Proces 5 z 8 wykonywany na komp roma
Proces 6 z 8 wykonywany na komp roma
Proces 1 z 8 wykonywany na komp korek
Proces 0 z 8 wykonywany na komp korek
Proces 3 z 8 wykonywany na komp korek
Proces 2 z 8 wykonywany na komp korek
Proces 7 z 8 wykonywany na komp roma
```

Przykład 1-4 Wykonanie programu przykładowego za pomocą programu mpiexec z linii poleceń

## Literatura

- [1] DeinoMPI, Deino Software © 2009 , <http://mpi.deino.net/DeinoMPI.pdf>.
- [2] DeinoMPI Compiler Support, Deino Software © 2009 ,
- [3] MPI: A Message-Passing Interface Standard Version 2.1 Message Passing Interface Forum, June 23, 2008
- [4] Wikipedia the Free Encyclopedia, <http://en.wikipedia.org/>
- [5] Wikipedia wolna encyklopedia, <http://pl.wikipedia.org/>

