

Instalacja środowiska MPI w systemie operacyjnym Linux

Zakres ćwiczenia

W tym ćwiczeniu dowiesz się, w jaki sposób dokonać instalacji środowiska MPI w systemie operacyjnym Linux. Do instalacji wybrano pakiet MPICH2, będący implementacją standardu MPI-2, stworzoną w Argonne National Laboratory. Pakiet ten jest stanowczo zalecany, jeżeli nie istnieje potrzeba komunikacji danych heterogenicznych, np. o innym porządku bajtów, albo innej długości dla liczb całkowitych.

Etapy instalacji środowiska MPI w systemie operacyjnym Linux

W celu zainstalowania środowiska MPI niezbędne jest podjęcie następujących kroków:

A. Pobranie pliku `mpich2-1.0.3.tar.gz`. Plik ten zawiera kod źródłowy, stąd może być instalowany na dowolnej platformie, w tym pod systemem operacyjnym Linux.

1. Wejdź na stronę: <http://www-unix.mcs.anl.gov/mpi/mpich2/>.
2. Wejdź w zakładkę **Download**. Pobierz plik `mpich2-1.0.3.tar.gz` dla dowolnej platformy.
3. Zapisz plik do katalogu `/home/you/install/mpich2` gdzie `you` jest Twoją nazwą użytkownika

B. Instalacja pliku `mpich2-1.0.3.tar.gz`.

1. Przed instalacją powinny być spełnione następujące wymagania:

- pobrany jest plik `mpich2-1.0.3.tar.gz`.

- zainstalowano kompilator języka C, np. `gcc`.

- zainstalowano kompilator języka Fortran, np. `g77`, kompilator języka C++, np. `g++`, i kompilator języka Fortran 90, np. `g95`, `gfortran` lub `F`, jeżeli przewiduje się korzystanie z programów w tych językach.

- zainstalowano pakiet Python 2.2 lub późniejszy, do budowy systemu zarządzania procesami MPD.

2. Rozpakuj plik `mpich2-1.0.3.tar.gz` poleceniem:

```
gunzip mpich2-1.0.3.tar.gz
```

utwórz strukturę katalogu poleceniem:

```
tar xf mpich2-1.0.3.tar
```

i przejdź do katalogu `mpich2-1.0.3` poleceniem:

```
cd mpich2-1.0.3
```

3. Utwórz katalog instalacyjny (domyślnym jest katalog `/usr/local/bin`):

```
mkdir /home/you/mpich-install
```

Najwygodniej byłoby, gdyby ten katalog był współdzielony przez wszystkie komputery, na których chcesz uruchamiać procesy. Jeżeli nie, musisz powielić go na każdym komputerze po zakończeniu instalacji.

4. Dokonaj konfiguracji pakietu MPICH2, wskazując katalog instalacyjny poleceniem:

```
./configure --prefix=/home/you/mpich2-install |& tee  
configure.log
```

a w powłoce `sh` poleceniem:

```
./configure --prefix=/home/you/mpich2-install 2>&1 | tee  
configure.log
```

Sprawdź zawartość pliku `configure.log`, aby upewnić się, że wszystko idzie dobrze.

5. Skompiluj MPICH2 poleceniem:

```
make |& tee make.log
```

Sprawdź zawartość pliku `make.log`, czy proces ten przebiega poprawnie.

6. Przenieś pliki MPICH2 do właściwego katalogu poleceniem:

```
make install |& tee install.log
```

W tym kroku umieszcza się wszystkie wymagane pliki wykonywalne i skrypty w podkatalogu `bin` katalogu określonego argumentem `prefix` podczas konfiguracji pakietu, czyli katalogu `/home/you/mpich2-install`.

7. Dodaj podkatalog `bin` do katalogu instalacyjnego poleceniem `setenv` albo `export`, w zależności od powłoki.

Dla powłoki `csh` i `tcsh`:

```
setenv PATH /home/you/mpich2-install/bin:$PATH
```

a dla powłoki `bash` i `sh`:

```
export PATH=/home/you/mpich2-install/bin:$PATH
```

Sprawdź, czy proces instalacji do tego momentu przebiega normalnie, wydając następujące trzy polecenia:

```
which mpd
```

```
which mpiexec
```

```
which mpirun
```

Odnoszą się one do trzech poleceń MPICH2: `mpd`, `mpiexec` i `mpirun` umieszczonych wcześniej, w punkcie 6, w podkatalogu `bin` Twojego katalogu instalacyjnego `/home/you/mpich2-install`. Jeżeli ten katalog, tzn. `/home/you/mpich2-install`, nie jest współdzielony przez inne komputery, np. poprzez współdzielony system plików NFS, to właśnie teraz powinieneś skopiować go na pozostałe komputery.

C. Uruchomienie i przetestowanie menedżera procesów MPD.

1. MPICH2 stosuje zewnętrzny menedżer procesów do dającego się skalować uruchomienia dużych zadań MPI. Domyślnym menedżerem zadań jest MPD, alternatywnymi SMPD i gforker.
2. MPD jest pierścieniem demonów na komputerach, na których masz zamiar uruchomić programy MPI. W kilku następnych krokach dowiesz się, w jaki sposób utworzyć taki pierścień, jak go uaktywnić, i jak go przetestować. Więcej szczegółów znajdziesz w pliku `README` w katalogu `mpich2-1.0.3/src/pm/mpd`. Szczegóły te dotyczą wyświetlania listy uruchomionych zadań, zawieszania i usuwania zadań, przekazywania sygnałów do zadań, oraz stosowania debuggera `mpigdb`.
3. Rozpocznij od stworzenia w Twoim katalogu domowym pliku `.mpd.conf`, plik ten zawiera jeden wiersz:

```
secretword=<secretword>
```

gdzie `<secretword>` jest unikalnym łańcuchem znaków, znanym tylko Tobie, odmiennym od Twojego hasła w systemie Linux. Nadaj temu plikowi prawa do odczytu i zapisu tylko przez Ciebie poleceniem:

```
chmod 600 .mpd.conf
```

4. Pierwszym sprawdzianem jest aktywacja pierścienia z jednym menedżerem na lokalnym komputerze, przetestowanie jednego polecenia `mpd` i deaktywacja pierścienia:

```
mpd&
```

```
mpdtrace
```

```
mpdallexit
```

Wyjściem polecenia `mpdtrace` jest nazwa komputera, na której pracujesz. Polecenie `mpdallexit` powoduje wyjście z demona `mpd`.

5. Teraz dokonasz aktywacji pierścienia demonów `mpd` na pewnym zbiorze komputerów. W tym celu utwórz plik nazwie `mpd.hosts`. W pliku tym wpisz, wiersz po wierszu, nazwy wszystkich komputerów w pierścieniu. Sprawdź, po kolei, czy możesz osiągnąć każdy z tych komputerów poprzez `ssh` albo `rsh`, bez wpisywania hasła. Przetestuj ten fakt poleceniem:

```
ssh othermachine date
```

albo

```
rsh othermachine date
```

Jeżeli nie możesz tego zrobić bez wpisywania hasła, zmień konfigurację `ssh` albo `rsh`. Jeżeli nadal się to nie udaje, postępuj jak opisano w kolejnym punkcie.

6. Uruchom demony na pewnej liczbie komputerów z listy w pliku `mpd.hosts`.

```
mpdboot -n <number to start>
```

gdzie `<number to start>` może być mniejszy niż 1 + liczba komputerów w pliku `mpd.hosts`, ale nie może być większy niż 1 + liczba komputerów w pliku `mpd.hosts`. Jeden demon `mpd` jest uruchomiony na komputerze, na którym uruchomiono polecenie `mpdboot`, i jest już wliczony do liczby `<number to start>`, niezależnie czy występuje w pliku `mpd.hosts`.

Jeżeli nie można wykonać polecenia `mpdboot`, ponieważ jest problem z konfiguracją `ssh` i `rsh`, możesz prowizorycznie rozwiązać ten problem, uruchamiając ręcznie demony, poleceniem:

```
mpd &
```

które uruchamia lokalnego demona, oraz poleceniem:

```
mpdtrace -l
```

które powoduje, że lokalny demon podaje swój komputer i port w postaci `<host>` `<port>`.

Następnie zarejestruj się na każdym z pozostałych komputerów, przejdź do właściwego katalogu `install/bin` i wykonaj:

```
mpd -h <hostname> -p <port> &
```

gdzie `<hostname>` i `<port>` należą do oryginalnego uruchomionego przez Ciebie `mpd`. Na każdym komputerze, po uruchomieniu `mpd`, wykonaj:

```
mpdtrace
```

aby zobaczyć, które komputery są nadal w pierścieniu.

7. Testuj właśnie utworzony pierścień poleceniem:

```
mpdtrace
```

Efektom tego polecenia powinna być lista komputerów, na których aktualnie chodzą demony MPD. Można sprawdzić również, jak szybko komunikat obiegnie cały pierścień poleceniem:

```
mpdringtest
```

Ale ponieważ dzieje się to błyskawicznie, lepiej jest ten dystans przedłużyć tysiącrotnie:

```
mpdringtest 1000
```

D. Testowanie uruchomienia zadania na wielu procesach.

1. Testuj czy w pierścieniu można uruchomić zadanie na wielu procesach. Zrób to poleceniem o składni:

```
mpiexec -n <number> hostname
```

Liczba procesów nie musi zgadzać się z liczbą komputerów w pierścieniu, jeżeli procesów jest więcej, będą się „zawijać”. Możesz to zobaczyć, wyświetlając etykiety procesów poleceniem:

```
mpiexec -l -n 30 hostname
```

albo podając pełną ścieżkę:

```
mpiexec -l -n 30 /bin/hostname
```

2. Uruchom zadanie MPI, korzystając z przykładu z katalogu instalacyjnego albo z katalogu `mpich2-1.0.3/examples`. Wybierz zadanie `cpu`, obliczające liczbę π metodą równoległego całkowania numerycznego:

```
mpiexec -n 5 cpu
```

Liczba procesów, jak już wiesz, nie musi odpowiadać liczbie komputerów. Przykład `cpu` sam poda, na ilu procesach się liczy. Nie ma potrzeby podawania komputerów, w obliczeniach mogą brać udział te, które są w pierścieniu.

3. Zwróć uwagę, jak wiele opcji ma polecenie `mpiexec`. Możesz je wszystkie poznać podając polecenie:

```
mpiexec --help
```

4. Typowym przykładem jest polecenie `mpiexec`, w postaci gwarantującej, że proces z numerem 0 uruchomi się akurat na Twoim komputerze:

```
mpiexec -n 1 master : -n 19 slave
```

albo

```
mpiexec -n 1 -host mymachine : -n 19 slave
```

Zadanie do samodzielnego wykonania

Dokonaj instalacji pakietu MPICH2 na dwóch, trzech, lub czterech komputerach, pracujących pod systemem operacyjnym Linux. Wykorzystaj wiadomości z niniejszego ćwiczenia, oraz z dwóch dokumentów, dostępnych w podkatalogu `/doc/userguide` i `/doc/installguide`, albo możliwych do pobrania ze strony: <http://www-unix.mcs.anl.gov/mpi/mpich2/>. Uruchom w tym środowisku wskazany przykład.

Podsumowanie

Podczas tego ćwiczenia zainstalowałeś implementację standardu MPI-2 o nazwie MPICH2, pod systemem operacyjnym Linux. Jest to kolejne Twoje doświadczenie z MPI. Czas na programowanie w MPI.

Co powinieneś wiedzieć:

- Jak dokonać instalacji implementacji MPICH2 na komputerze pracującym pod systemem operacyjnym Linux.
- Jak uruchomić przykładowy program w środowisku MPI.