

Systemy równoległe i rozproszone

Projekt 2

opis

Wojciech Majkut
Magdalena Piłat

Opis programu

Program został napisany w języku C++/C z wykorzystaniem standardu oprogramowanie rozproszonego CORBA. Program składa się z dwóch serwerów oraz jednego klienta - zarządcy.

- Interfejs do operacji zdalnych został napisany w języku IDL. Znajduje się w pliku ***TemperatureDataInterface.idl*** i zawiera deklarację jednej metody ***randomWalk***. Metoda ta wyznacza temperaturę w pojedynczym węźle siatki. Parametrami wejściowymi są: tablica 2D zawierająca temperatury w węzłach siatki w danej iteracji, nr wiersza węzła oraz nr kolumny węzła. Metoda zwraca wyznaczoną wartość temperatury dla zadanego węzła.
- Namiastka - plik ***TemperatureDataImplementation.cpp***. Plik zawiera procedurę *stub* służącą do przekazania argumentów wywołania do serwera, a następnie do odebrania wyników zdalnej procedury ***RandomWalk***
- Implementacja - plik ***RandomWalk.c***. Plik w języku C zawierający implementację metody Monte Carlo wyznaczenia temperatury w danym punkcie siatki
- Serwer tj. program świadczący usługi wykonania operacji - plik ***temperature_server.cpp***. W pliku tym inicjowany jest broker, następnie zainicjowany i wywołany jest POA. Tworzony jest obiekt zarządzający POA tzw. POA Manager. Tworzony jest obiekt klasy implementującej operację ***randomWalk*** oraz servant CORBA. Następnie uzyskiwana jest referencja do serwisu *Naming* i utworzony obiekt jest zapisywany w serwisie. Na końcu manager aktywowany.
- Klient tj. zarządca - plik ***temperature_client.cpp***. Wczytuje on referencje do serwerów z serwisu *Naming*, a następnie wywołuje metodę ***randomWalk*** przez nich udostępnioną. Obsługa serwera jest opakowana w wątek reprezentowany przez obiekt klasy ***SideThread***, dziedziczącej po ***JTCThread***. W każdej iteracji tworzone są nowe obiekty wątków, obliczana jest temperatura w siatce oraz trwa oczekiwanie na zakończenie wątków. Obliczenia trwają aż do osiągnięcia określonego błędu w różnicy średniej temperatury pomiędzy kolejnymi iteracjami. Wyniki zapisywane są w pliku ***dane.txt***.

Obsługa programu

Do programu został dołączony plik Makefile.

- **make all** - generowanie deklaracji na podstawie pliku idl, kompilacja serwera oraz klienta
- **make runserver** - uruchomienie naming service oraz dwóch serwerów o nazwach **server1** oraz **server2** na **localhost** oraz porcie **3456**.
- **make runclient** - uruchomienie klienta z argumentami **server1** oraz **server2** odpowiadającymi uruchomionym uprzednio serwerom. Klient podłączony jest do naming service uruchomionego przez runserver.
- **make gnuplot** - generowanie wykresu i zapisywanie go w pliku **cieplo.png**

NR portu można łatwo zmienić poprzez zmianę wartości zmiennej PORT w pliku Makefile.