

1. Temat zadania

Temat numer 20, „system mnożenia dużych macierzy kwadratowych (lub innych obliczeń algebry liniowej)”

2. Opis protokołu komunikacyjnego

Komunikacja odbywa się przy wykorzystaniu interfejsów gniazd BSD. Przesyłanie danych pomiędzy serwerem, klientem i maszynami obliczeniowymi opiera się o tablicę znaków, która jest podzielona na dwie części: pierwsza informująca o długości wiadomości kończąca się znakiem nowej linii, druga – zawierająca właściwą wiadomość. Ten wzór jest zachowany zarówno wobec komunikacji między serwerem i maszynami obliczeniowymi jak i serwerem i klientami.

3. Opis implementacji, w tym krótki opis zawartości plików źródłowych

2 programy (serwera i maszyny obliczeniowej) zawierają funkcję `readFromADescriptor`, która zajmuje się czytaniem wiadomości w sposób umożliwiający zmianę rozmiaru bufora ładujące dane oraz zapobiega blokowaniu w specyficznych przypadkach omówionych na laboratoriach. Klienci są obsługiwani poprzez serwer zawierający metody klasy `std::thread`, natomiast informacje o gotowości maszyn obliczeniowych do wykonywania obliczeń zapewnia funkcja `select`. Maszyna obliczeniowa wykonuje obliczenia w jednym wątku, tak by zajmowała się w danym momencie wyłącznie obliczeniami. Kod klienta jest napisany w C# i wykorzystuje głównie gotowe metody pokazane na zajęciach i udostępnione na Pańskiej stronie.

Klient wystawia 2 macierze, których iloczyn chce uzyskać. Serwer analizuje gotowość maszyn obliczeniowych i wysyła do każdej niepracującej maszyny odpowiednie wiersze i kolumny macierzy do obliczenia odpowiedniego pola macierzy wynikowej. Następnie maszyny przesyłają odpowiedź w postaci wyznaczonej wartości pola. Serwer łączy wszystkie odpowiedzi w spójną całość i przesyła końcową macierz do klienta. Wszystkie dane są przesyłane jako łańcuch znaków w formacie `.csv`.

4. Sposób kompilacji, uruchomienia i obsługi programów projektu.

Kompilację serwera i maszyny obliczeniowej można wykonać standardowym kompilatorem `g++`. Serwer należy kompilować z dodatkową flagą `-pthread`.

Każda maszyna obliczeniowa musi być uruchomiona z jej unikalnym numerem, ponieważ kod maszyny pobiera z pliku `data.txt` informacje o porcie i adresie IP danej maszyny.

Klient jako napisany w C# powinno się uruchamiać ze stosownym kompilatorem, który na to pozwoli.