**Obliczenia równolegle**

Rosnące w ostatnich latach zapotrzebowanie na metody rozwiązywania bardzo dużych zadań obliczeniowych zaowocowało gwałtownym rozwojem obliczeń równoległych. W obliczeniach równoległych, dzięki jednoczesnemu wykonywaniu obliczeń przez wiele procesorów, można uzyskać znaczne skrócenie czasu oczekiwania na wynik końcowy w stosunku do tych samych obliczeń wykonywanych przez jeden procesor na komputerze sekwencyjnym.

*(tutaj wstaw tabelę)*

Podstawową wielkością charakteryzującą wydajność obliczeń równoległych jest przyspieszenie S (ang. speedup). Jest to stosunek całkowitego czasu obliczeń, przeprowadzonych za pomocą sekwencyjnej implementacji algorytmu na pojedynczym procesorze do czasu obliczeń na N-procesorach. Przyspieszenie oznacza zatem ile razy szybciej można wykonać program na N-procesorach niż na jednym procesorze.

*(tutaj wstaw wzór)*

W obliczeniach równoległych bardzo dużą rolę odgrywa komunikacja między procesami. Do realizacji komunikacji najczęściej stosowane są dwie biblioteki: PVM (ang. Parallel Virtual Machine) i MPI (ang. Message Passing Interface).

**Standard MPI**

MPI (ang. Message Passing Interface) jest standardem interfejsu do przesyłania komunikatów w środowisku rozproszonym (rzeczywiste lub wirtualne maszyny równoległe) opracowanym w latach 1993-1994 przez grupę badaczy z USA i Europy reprezentujących przemysł i instytucje naukowe (grupa ta nosi nazwę ”MPI Forum” - [http://www.mpi-forum.org](http://www.mpi-forum.org/)). Głównymi celami podczas opracowywania standardu były: przenośność kodu między różnymi systemami, duża wydajność i funkcjonalność. Z punktu widzenia programisty MPI jest zbiorem procedur i funkcji wywoływanych z programów napisanych w językach C, C++ lub Fortran, a służących do wysyłania i odbierania komunikatów oraz do synchronizacji procesów. MPI definiuje tylko bibliotekę do komunikacji pozostawiając pozostałe aspekty, takie jak inicjalizacja procesów i sterowanie, indywidualnym programistom. Istnieje wiele implementacji MPI na różne platformy sprzętowe m.in. Cray 3D, IBM SP, IBM RS/6000, Hitachi SR2201, DEC Alpha, Intel x86, Intel Paragon.

**WMPI**

Na każdym komputerze uruchomiony jest ”daemon” WMPI, czyli program pośredniczący w wymianie informacji między procesami. Każdy ”daemon” zawiera tablicę zadań wszystkich procesów, którymi zarządza. Procesy komunikują się z lokalnymi ”daemonami” poprzez TCP, natomiast ”daemony” znajdujące się na różnych komputerach używają do komunikacji UDP.

Napisanie programu wykorzystującego WMPI i Microsoft Visual C++ 6.0 wymaga w pierwszej kolejności stworzenia nowego projektu. W tym celu wybieramy: ”File - New - Projects - Win 32 Console Application” i podajemy nazwę projektu - ”Project name:” (np. ”MyApp”). System automatycznie wpisuje ścieżkę dostępu i katalog, w którym umieszczone zostaną wszystkie pliki tworzące projekt - ”Location:” (np. ”Location: C:\MyApp”). Ścieżkę tę możemy zmienić.

Po naciśnięciu ”OK.”, jako typ tworzonego projektu (”What kind of Console Application do you want to create? ”), wybieramy ”An empty project” (jest to ustawienie domyślne). Kolejną czynnością jest zmiana typu biblioteki DLL. Wybieramy “Project - Settings - C/C++”, a następnie ”Category: Code Generation”i “Use run-time library: Multithreaded DLL”.

**Kompilacja i uruchomienie programu**

Po napisaniu programu należy przeprowadzić jego kompilację do postaci wykonywalnej naciskając klawisz ”F7” lub wybierając polecenie: ”Build - Build nazwa.exe”, gdzie ”nazwa” jest nazwą tworzonego projektu.

Programy wykorzystujące implementację WMPI uruchamiane są przy użyciu podsystemu ”P4”. Przed uruchomieniem systemu należy przygotować specjalny plik konfiguracyjny (tzw. ”procgroup file”) określający wszystkie działające w nim procesy. Jest to plik tekstowy o odpowiedniej składni.

Jako nazwa komputera może występować nazwa NetBIOS (np. ”ketim32”), domena, pod którą występuje komputer (np. ”ketim35.pb.bialystok.pl”) lub numer IP (np. ”212.33.69.110”). Plik konfiguracyjny odczytywany jest tylko raz podczas startu, zatem niemożliwe jest rekonfigurowanie systemu podczas jego działania (np. dodanie nowych komputerów).

Start systemu odbywa się poprzez uruchomienie pierwszego programu, który jest jednocześnie pierwszym procesem w systemie (nazywany jest ”big master” - ”bm”). Uruchamia on pozostałe procesy (”slave”) i konfiguruje tzw. wirtualny komputer równoległy (ang. the virtual parallel machine) zgodnie z tym, co zostało określone w pliku konfiguracyjnym. Plik konfiguracyjny może być wyspecyfikowany na trzy sposoby.