

Tadeusz NOWICKI, Łukasz WRZOSEK

Wojskowa Akademia Techniczna, ul. Kaliskiego 2, 00-908 Warszawa

E-mail: tadeusz.nowicki@wat.edu.pl, lukasz.wrzosek@wat.edu.pl

Symulator do badania własności systemów wieloprocesorowych

1 Wprowadzenie

Praca związana jest z badaniem własności systemów wieloprocesorowych [1], [2]. Stosowanie takich systemów ma podstawową zaletę, można w nich realizować jednocześnie wiele zadań. Powoduje to znaczące przyspieszenie obliczeń [3]. Napływające do systemu zadania są rozdzielane między rezydujące procesory, które zadania te wykonują. Kluczowym elementem systemu jest tu dobór odpowiedniego algorytmu przyporządkowywania napływających zadań do poszczególnych procesorów. Cechy algorytmów wpływają znacząco na efektywność funkcjonowania całego systemu.

Pierwsze komputery były maszynami sekwencyjnymi. Procesory, które wówczas projektowano, były przeznaczone do wykonywania sekwencji instrukcji, jedna po drugiej. Wywarło to wpływ na cały świat informatyki na wiele lat: na językach programowania, budowie systemów operacyjnych, a nawet na sposobie myślenia ówczesnych inżynierów komputerowych. Podejście to tak głęboko się zakorzeniło w informatyce, że kiedy na świecie upowszechniały się komputery osobiste, wciąż dominującym systemem operacyjnym był DOS. System ten wykonywał w jednej chwili tylko jeden program komputerowy, podczas gdy systemy z rodziny UNIX już od kilkunastu lat potrafiły dzielić czas procesora pomiędzy różne programy. Było to spowodowane tym, że zagadnienia związane z wykonywaniem równoległym są o wiele bardziej skomplikowane i wymagają znajomości złożonych problemów w tematyce wieloprzetwarzania. W celu pełniejszego wykorzystania pojedynczego procesora systemy operacyjne zostały wzbogacone o wielozadaniowość. Umożliwiło to wykonywanie wielu procesów (programów komputerowych) w tym samym odcinku czasu. Zaprogramowany zegar wysyłał przerwanie, którego obsługa zmieniała adres powrotu programu obsługi przerwania. Z punktu widzenia użytkownika oba procesy wykonywane były jednocześnie, bowiem czas przełączania między procesami był niezauważalnie krótki. Wydajność procesora nie wzrosła, ale wydajność całego systemu komputerowego w wielu zastosowaniach zwiększyła się. Gdy jeden proces czekał na jakieś dane, na przykład dane z klawiatury, drugi proces mógł w tym czasie wykonywać obliczenia. Pojedynczy proces nie mógł skorzystać z wielozadaniowości systemu, bowiem był programem sekwencyjnym. Zaczęto więc pisać aplikacje wieloprocesowe tak, aby czynności mogące wstrzymać proces, w oczekiwaniu na dane, były wykonywane w oddzielnym procesie. Wiązało się to z pewnym spadkiem wydajności programów. Aby tym problemom zapobiec wprowadzono przetwarzanie wielowątkowe. Proces w systemie operacyjnym widziany jest jako spis obszarów pamięci kodu i danych oraz stanu procesu. Jeżeli proces aktualnie nie jest uruchomiony, to również zapisany jest stan wszystkich rejestrów w momencie wstrzymania działania procesu. Zatem dwa procesy różnią się tak obszarami pamięci, jak i stanem rejestrów. Wątki natomiast współdzielą obszary pamięci i różnią się jedynie

stanem rejestrów. Mogąc używać tych samych obszarów pamięci w wielu przypadkach uniknięto kopiowania danych, gdyż były one bezpośrednio dostępne dla każdego z wątków. Czas tworzenia nowego wątku, w porównaniu do czasu tworzenia procesu, jest o wiele krótszy. Wraz z wprowadzeniem wielowątkowości pojawił się całkiem nowy i skomplikowany problem spójności danych. Do naruszenia spójności danych dochodzi wtedy, gdy pod jednym adresem pamięci odbywa się zapis, z jednoczesnym odczytem lub zapisem przez inny wątek. Aby rozwiązać ten problem wprowadzono synchronizację między wątkami. Wszystkie te prace stanowiły podstawę konstrukcji systemów równoległego przetwarzania, w szczególności systemów wieloprocesorowych.

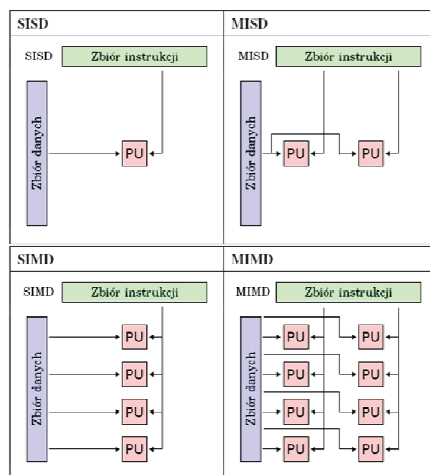
W celu usystematyzowania pojęć z zakresu systemów równoległego przetwarzania danych najczęściej stosuje się podział zaproponowany przez M. Flynna w 1966 roku, zwany taksonomią Flynna:

SISD - pojedynczy strumień rozkazów, pojedynczy strumień danych (single instruction, single data stream). Pojedynczy procesor wykonuje pojedynczy strumień operacji na pojedynczym strumieniu danych. Do tej grupy niegdyś zaliczano systemy jednoprocessorowe. Ten typ przetwarzania praktycznie dzisiaj nie występuje.

SIMD - pojedynczy strumień rozkazów, wiele strumieni danych (single instruction, multiple data stream). Pojedynczy procesor wykonuje pojedynczy strumień operacji, działając jednocześnie na wielu strumieniach danych. Do tej kategorii zalicza się operacje wektorowe, gdzie np. operacja dodawania wektorów składa się z wielu operacji dodawania składowych wektora. Na komputerze typu SIMD dodawanie składowych odbywa się równoległe. Prawie każdy dzisiejszy CPU posiada wyspecjalizowane rejestry i rozkazy do wykonywania operacji wektorowych. Innym rodzajem procesorów zaliczanych do tej kategorii są procesory tablicowe.

MISD - wiele strumieni rozkazów, pojedynczy strumień danych (multiple instruction, single data stream). W założeniu komputer taki miał wykonywać wiele strumieni instrukcji na pojedynczym zbiorze danych, lecz brak zastosowania takiego rozwiązania spowodował, że wciąż brak jest komercyjnych implementacji tej architektury.

MIMD - wiele strumieni rozkazów, wiele strumieni danych (multiple instruction, multiple data stream). Wiele procesorów wykonuje różne zbiory instrukcji na różnych zbiorach danych. Do tej kategorii zaliczono systemy SMP, klastrowe i NUMA. Produkowane w tej technologii systemy komputerowe mogą składać się nawet z setek tysięcy pojedynczych procesorów.



Rys. 1. Taksonomia Flynna – systemy równoległe przetwarzania danych

Fig. 1. Flynn taxonomy – parallel data processing systems

2 Badanie systemów wieloprocessorowych

System wieloprocessorowy można badać na wiele sposobów. Przede wszystkim można zbierać informacje o działaniu gotowego systemu. Najwierniejszym modelem systemu jest system sam w sobie. Jednakże nawet posiadanie takiego systemu nie oznacza, że zna się wszelkie jego parametry. W celu poszerzenia posiadanej wiedzy można badać taki system bezpośrednio. Model fizyczny można modyfikować tak, aby sam dostarczał odpowiednich danych do wyznaczania wartości parametrów efektywnościowych.

Innym sposobem jest podłączenie sond, które będą mierzyć parametry, jednocześnie nie wpływając na system. Ma to swoje niewątpliwe zalety, ale jednocześnie jest bardzo drogie i często nie umożliwia pomiaru wszelkich potrzebnych parametrów. Obie metody również wymagają dostępu do systemu wieloprocessorowego, co w przeważającej części przypadków jest niemożliwe. Systemy są bardzo kosztowne w utrzymaniu i nikt nie pozwoli na uruchamianie nań testów w celu uzyskania kilku statystyk.

Modele symulacyjne mają tę zaletę, że można na nich wykonywać serie eksperymentów, które pod względem strukturalnym odpowiadają eksperymentom na systemie rzeczywistym. Można w ten sposób badać różne przebiegi zdarzeń w systemie, nawet takie, które na rzeczywistym systemie mogłyby doprowadzić go do jego zniszczenia lub być zbyt kosztowne (np. wymagałyby wyłączenia systemu z jego normalnej pracy). Ponieważ zmiennymi niezależnymi są wejścia, a zależnymi wyjścia systemu, to można większe modele budować z mniejszych, wcześniej zaprojektowanych modeli, łącząc wyjścia jednego modelu z wejściami innych modeli. Takie postępowanie pozwala budować model fragmentami, jednocześnie ułatwiając ich weryfikację. Duża część małych modeli może być z powodzeniem wielokrotnie używana do budowy innych modeli symulacyjnych.

Ważne jest, aby model symulacyjny wcześniej zweryfikować, na przykład poprzez rozwiązanie modelu matematycznego drogą analityczną dla szczególnych przypadków. Jeżeli jest to zbyt skomplikowane, to można uprościć model matematyczny tak, aby porównując rozwiązania osiągnięte metodą analityczną i symulacyjną można było zweryfikować poprawność całego modelu. Postępowanie takie umożliwia wczesne wykrycie błędów modelu symulacyjnego (a dokładniej implementacji modelu matematycznego). Zapobiega to sytuacjom, w których po zebraniu wszystkich wyników mogłoby się okazać, że dane wyjściowe zupełnie odbiegają od wartości rzeczywistych.

3 Wskaźniki efektywności systemów wieloprocesorowych

Efektywność każdego systemu, w tym i wieloprocesorowego, musi być mierzona. Ponieważ w dużej mierze zależy to od samego systemu, postarano się wymienić tylko takie parametry, które bezpośrednio mogłyby służyć jako kryteria porównawcze systemów wieloprocesorowych. Do typowych parametrów efektywnościowych zalicza się:

- Czas odpowiedzi (ang. response time) - definiowany jest jako odstęp czasu między odebraniem zapytania przez system, a jego odpowiedzią. Nie jest to do końca sprecyzowana definicja, gdyż w rzeczywistości utworzenie zapytania nie jest nieskończenie krótkie tak samo, jak utworzenie odpowiedzi. Tak więc można czas odpowiedzi liczyć do momentu rozpoczęcia odpowiedzi (z punktu widzenia systemu), lub do zakończenia odpowiedzi (punkt widzenia użytkownika).
- Przepustowość (ang. throughput) - jest to bezpośredni wskaźnik wydajności. Zazwyczaj liczony jest w liczbie zapytań, jakie system jest w stanie obsłużyć w jednostce czasu. W systemach wsadowych może to być liczba zadań wykonanych w jednostce czasu. Zazwyczaj przepustowość systemu rośnie wraz z tym, jak rośnie obciążenie systemu. Jednocześnie czas odpowiedzi systemu wydłuża się. Przy pewnej wartości obciążenia czas odpowiedzi zaczyna bardzo szybko rosnąć. Wartość przepustowości przy tym obciążeniu nazywamy użyteczną pojemnością (ang. usable capacity).
- Nominalna pojemność (ang. nominal capacity) - jest to maksymalna wartość przepustowości przy idealnym strumieniu zapytań. System zazwyczaj nie osiąga w warunkach roboczych pojemności nominalnej.
- Efektywność (ang. efficiency) - stosunek przepustowości systemu dla używalnej pojemności do przepustowości przy pojemności nominalnej. W systemach wieloprocesorowych może to też być stosunek przepustowości idealnego systemu wieloprocesorowego do przepustowości rzeczywistego systemu wieloprocesorowego.

4 Algorytmy rozdziału zadań w systemach wieloprocesorowych

Zarządca (Manager) w symulatorze zajmować się będzie rozdzielaniem zadań do kolejek procesorów. Jego działanie jest cykliczne. Zadania mogą być przydzielane do procesorów zgodnie z następującymi algorytmami przydziału zadań:

- *losowy* - zadania przydzielane są procesorom losowo,

- *zmodyfikowany losowy* - zadania również przydzielane są procesorom losowo, ale prawdopodobieństwo przydzielenia procesorowi zadania jest proporcjonalne do jego szybkości,
- *najkrótsza kolejka* - zadanie przydzielane jest do procesora o najkrótszej kolejce wejściowej,
- *ważona najkrótsza kolejka* - zadanie przydzielane jest do procesora o najkrótszej, uwzględniając jego szybkość, kolejce wejściowej,
- *po kolei* - zadania przydzielane są procesorom po kolei.

Zgodnie z nimi Zarządca umieszcza zadania i odpowiedzi w odpowiedniej kolejce.

5 Założenia projektowe dla symulatora systemu wieloprocessorowego

Zadany system jest systemem wieloprocessorowym typu klaster. Klaster ten składa się z wielu maszyn jednoprocessorowych, zwanych dalej węzłami, połączonych bardzo szybko magistralą. Węzły posiadają na tyle dużo pamięci roboczej, że można założyć, iż nie zabraknie jej podczas dowolnego obciążenia systemu. Oprócz tego przyjęto, że procesory w węzłach charakteryzują się określoną szybkością. Magistrala jest buforowana tak, że można przyjąć, iż jest nieskończenie pojemna. Parametrem charakteryzującym magistralę jest jej szybkość.

Do magistrali również podłączony jest komputer zarządzający całym klastrem, zwany dalej zarządcą. Komputer ten spełnia zadanie planisty zadań, ustalając na podstawie przyjętego algorytmu, do którego węzła zadanie przypisać i następnie je wysłać poprzez magistralę. Zarządca odbiera zadania poprzez łącza zewnętrzne, które charakteryzują się założoną szybkością. Odbieranie zadań z wielu łącz jest zrealizowane algorytmem podziału czasu (ang. time sharing).

Celem budowy modelu było porównanie różnych algorytmów przydziału zadań do węzłów. Algorytmy takie mogą dawać różne wyniki przy różnych obciążeniach systemu. Innym parametrem, jaki zamierzano poznać, było skalowanie systemu w miarę wzrostu liczby procesorów, jak również wpływ na to skalowanie pojedynczej magistrali w systemie.

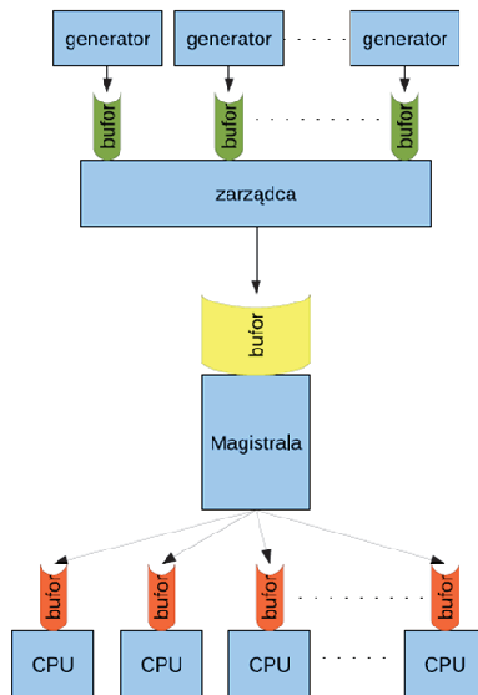
System jest wykorzystywany do gromadzenia sporej liczby danych (na przykład danych uzyskanych ze złożonych obserwacji), po ich uprzednim przetworzeniu. Takie obciążenie systemu jest o tyle charakterystyczne, że dane napływają do systemu, ale system nie wysyła odpowiedzi po realizacji zadania. Dane napływają do systemu w postaci pewnych paczek, które nazywać będziemy zadaniami. Wykonywanie zadań jest asynchroniczne, co jest bardzo istotne. Budując symulator można było pominąć fakt odpowiedzi systemu na zakończone zadanie.

Na potrzeby symulatora przyjęta następujące parametry systemu:

- liczba procesorów,
- szybkości procesorów,
- szybkość magistrali,
- algorytm przydziału zadań,

- rozkład czasu przybywania zadań do systemu,
- rozkład wielkości zadania,
- rozkład złożoności obliczeniowej zadania,
- szybkość łącz, którymi napływają zadania.

Na podstawie powyższych informacji zaproponowano model symulacyjny systemu, którego schemat przedstawia rysunek Rys.2.



Rys. 2. Schemat systemu wieloprocessorowego

Fig. 2. Diagram of a multiprocessor system

Do porównywania różnych konfiguracji systemu posłużono się wskaźnikami takimi, jak:

- czas przebywania zadania w systemie, czyli czas reakcji systemu,
- liczba zadań przebywających w systemie w poszczególnych etapach przechodzenia ich przez system oraz ich suma,
- rozłożenie obciążenia między procesory w postaci stosunku liczby zadań wykonanych na danym procesorze do liczby zadań przetworzonych przez system.

6 Konstrukcja symulatora systemu wieloprocessorowego

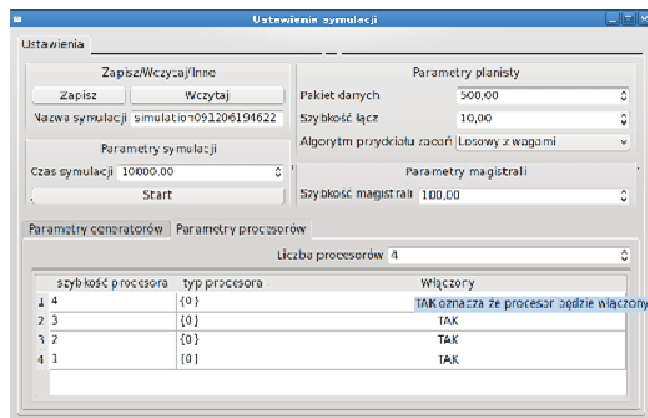
Do konstrukcji symulatora systemu wieloprocessorowego przyjęto następujące założenia:

- charakterystyka zadań: zadania są niezależne i tworzone w generatorze, a wykonywane w procesorach modułów systemu wieloprocessorowego mogą być przypisane do jednego typu procesora lub nie, oraz cechują się czasem wykonania na wzorcowym procesorze i czasem przesyłania przez wzorcową magistralę,
- parametry zadań są następujące: identyfikator zadania, czas wykonania na procesorze wzorcowym, czas przesyłania przez wzorcową magistralę, typ zadania, priorytet zadania, itd.,
- źródła zadań (generatory): jest wiele źródeł zadań symulowanych poprzez generatory zadań, które są niezależne, generują co pewien losowy czas zadanie, wygenerowane zadania trafiają do kolejek wejściowych zarządcy zadań, zadane są rozkłady czasu między kolejnymi zadaniami oraz rozkład generowanych typów zadań,
- zarządca - odbiera zadania z kolejek wejściowych algorytmem time sharing i umieszcza je w kolejce magistrali systemowej,
- każde zadanie zostaje przypisane do odpowiedniego procesora poprzez nadanie mu identyfikatora tego procesora,
- badane są następujące algorytmy przydziału zadań: losowy, ważony losowy, najkrótsza kolejka, ważona najkrótsza kolejka oraz po kolei,
- ustalany jest czas odbierania zadań z kolejek (time sharing), po którym przenosi się zadania do następnej kolejki,
- założenia odnośnie magistrali: magistrala przesyła zadania do procesorów zgodnie z kolejnością w kolejce zadań, czas przesyłania zadania przez magistralę zależy od szybkości magistrali względem magistrali wzorcowej,
- moduł (procesor): moduły wykonują przydzielone im zadania, na wejściu modułu znajduje się kolejka zadań, a czas wykonywania zadania zależy od czasu wykonywania zadania na procesorze wzorcowym i szybkości procesora względem procesora wzorcowego,
- parametry procesora: identyfikator procesora, maksymalna szybkość, typ procesora, długość kolejki, szybkość procesora względem procesora wzorcowego.

7 Implementacja symulatora systemu wieloprocessorowego

Symulator zbudowany został w oparciu o bibliotekę SIMLIB/C++. Biblioteka ta jest projektem uczelnianym napisanym w języku C++ w Instytucie Informatyki i Inżynierii na Uniwersytecie Technicznym w Brnie (Czechy). Prace nad nią trwają od 1990 roku i jest to projekt nadal rozwijany. Można z niej korzystać pod takimi systemami operacyjnymi jak: Linux (wymaga kompilatora GNU C++), FreeBSD oraz Windows. Została udostępniona na licencji GNU LGPL.

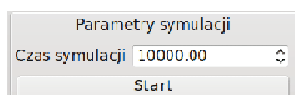
Okno główne programu (Rys.3) służy do ustawiania symulacji. Wczytanie ustawień przebiega w sposób następujący: użytkownik wciska przycisk wczytaj, w nowo otwartym oknie należy wskazać plik z ustawieniami, po zaznaczeniu pliku i zatwierdzeniu okno zniknie, a okno główne zostanie wypełnione zapisanymi danymi. Zapis przebiega w sposób analogiczny.



Rys. 3. Okno główne symulatora

Fig. 3. The main window of simulator

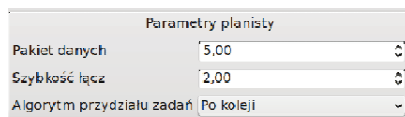
W oknie Parametry symulacji jest jedno pole, które służy do ustawiania czasu symulacji. Określa to, w jakim momencie najpóźniej symulacja się zakończy, może bowiem zakończyć się wcześniej, jeżeli wszystkie procesy zostaną przerwane. W tej sekcji znajduje się również przycisk „Start”, po kliknięciu którego rozpoczyna się symulacja (Rys.4).



Rys. 4. Parametry symulacji

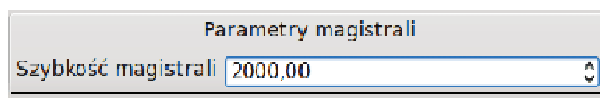
Fig. 4. Simulation parameters

Pole oznaczone jako „pakiet danych” (Rys.5) służy do konfiguracji tego, ile maksymalnie danych jednorazowo jest odbieranych przez zarządcę z jednej kolejki wejściowej zadań, zanim przełączy się do następnej. Pole „Szybkość łącz” zawiera liczbę określającą to, ile razy łącza zarządcy są szybsze od łącza wzorcowego. W liście nazwanej „Algorytm przydziału zadań” użytkownik może wybrać metodę przydziału zadań stosowaną przez zarządcę.



Rys. 5. Parametry zarządcy
Fig. 5. Manager parameters

W oknie Parametry magistrali (Rys.6), przy użyciu pola „Szybkość magistrali”, użytkownik może ustawić szybkość magistrali. Jest ona wyrażona przez liczbę określającą to, ile razy magistrala szybsza jest w stosunku do szybkości łącza wzorcowego.

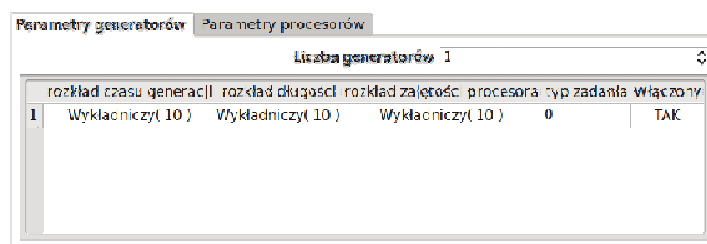


Rys. 6. Parametry magistrali
Fig. 6. Bus parameters

W zakładce Parametry generatorów (Rys.7) ustawienia dotyczą generatorów strumieni zadań napływających do systemu. W kolumnach tabeli konfiguruje się odpowiednio:

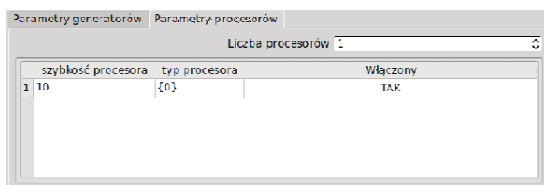
- rozkład czasu do wygenerowania zdania,
- rozkład czasu przesyłania zadania przez wzorcowe łącze,
- rozkład czasu wykonywania zadania na wzorcowym procesorze,
- typ generowanych zadań (zadanie danego typu może zostać wykonane tylko przez procesor, który ten typ zadań obsługuje),

a w ostatniej kolumnie ustala się, czy generator będzie generował zadania.



Rys. 7. Wybór rozkładów prawdopodobieństw
Fig. 7. The choice of probability distributions

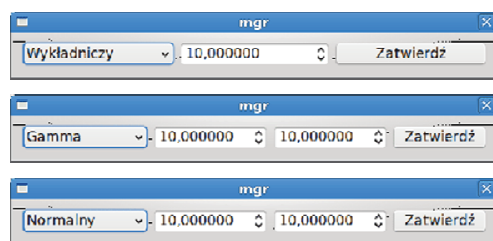
W zakładce na rysunku Rys.8 w kolumnie pierwszej ustawia się liczbę określającą to, ile razy dany procesor szybszy jest od procesora wzorcowego, w drugiej natomiast typy zadań, jakie ten procesor może realizować.



Rys. 8. Parametry procesorów

Fig. 8. Processors characteristics

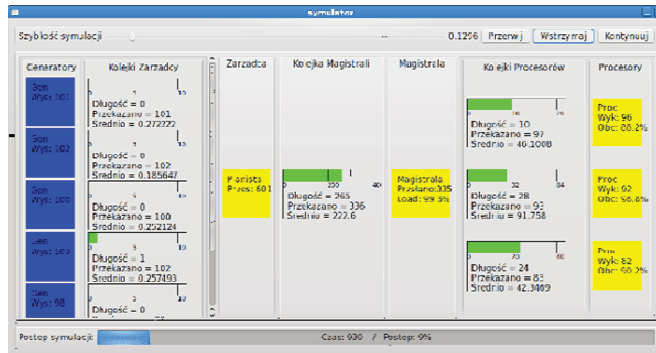
Z kolei w oknie konfiguracji rozkładu (Rys.9) istnieje możliwość wyboru parametrów rozkładów: wykładniczego, gamma i normalnego.



Rys. 9. Wybór parametrów rozkładów prawdopodobieństw

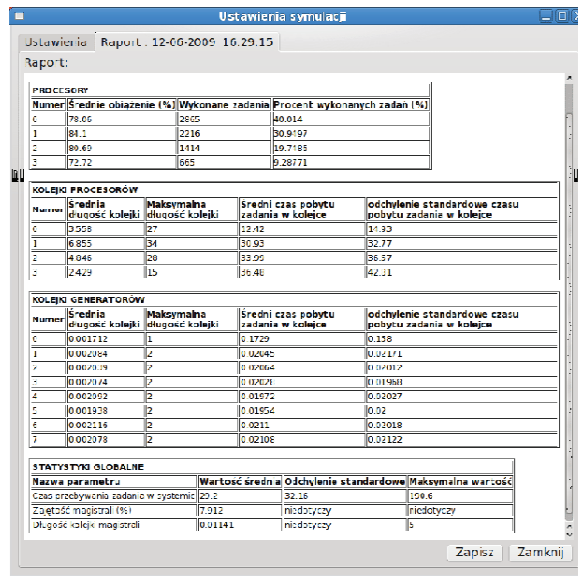
Fig. 9. The choice of probability distribution parameters

W oknie zobrazowania eksperymentu przy użyciu symulatora (Rys.10) umieszczone są informacje o generatorach zadań określające liczbę zadań wysłanych przez dany generator. W kolejkach zarządcy umieszczone są wszelkie kolejki przedstawione w postaci wykresu poziomego o zmiennej skali (dostosowuje się do długości kolejki). Poniżej każdej z kolejek przedstawione są: aktualna długość kolejki, liczba zadań, które zostały z kolejki pobrane i średnia długość kolejki. Każdy poziomy słupek przedstawia długość kolejki. W polu zarządcy widnieje liczba zadań, jakie zostały harmonogramowane i przekazane do wysłania przez magistralę. Specjalny kolor oznacza, że planista aktualnie pobiera zadanie z jednej z kolejek wejściowych. W kolejce magistrali są elementy obrazujące bufor wejściowy magistrali. Magistrala jest elementem obrazującym magistralę systemową. Różni się od elementu planisty tym, że jego tle znajduje się dodatkowo liczba przekazująca, jaki procent czasu magistrala jest zajęta przesyłaniem zadań. Kolejki procesorów są elementami obrazującymi bufory procesorów przechowujące zadania. Procesory, których wygląd, jak i znaczenie tych elementów, jest podobne do fragmentów GUI obrazujących magistralę z tym, że reprezentują procesory systemu.



Rys. 10. Okno obrazowania eksperymentu symulacyjnego
Fig. 10. The window of experiment visualization

Powstałe w wyniku eksperymentów symulacyjnych raporty można zapisać na dysk w postaci dokumentu „HTML”. Mają one postać jak na rysunku Rys.11.



Rys. 11. Wyniki eksperymentów symulacyjnych
Fig. 11. Results of simulation experiment

Statystyki dotyczą każdego z procesorów biorących udział w symulacji. Każdy procesor to jeden wiersz, a tabela zawiera następujące kolumny:

- numer - numer identyfikujący procesor,

- średnie obciążenie (%) - jest to wartość wyrażona w procentach określająca to, przez ile czasu w ciągu całej symulacji procesor był zajęty przetwarzaniem zadań,
- wykonane zadania - liczba przetworzonych zadań w ciągu symulacji,
- procent wykonanych zadań (%) - w tej kolumnie można odczytać to, jaki procent ze wszystkich zadań przetworzonych przez system było wykonanych na procesorze.

Statystyki kolejek procesorów przedstawiają statystyki buforów procesorów, a kolejne kolumny tabeli zawierają odpowiednio:

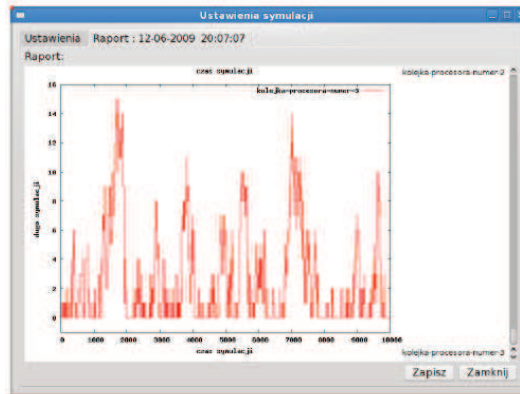
- numer - numer identyfikujący odpowiedni procesor tego bufora,
- średnia długość kolejki - średnia zajętość bufora zadań w ciągu całej symulacji,
- maksymalna długość kolejki - największa długość kolejki zarejestrowana podczas symulacji,
- średnia długość pobytu zadania w kolejce - ile średnio czasu minęło od pojawienia się zadania w buforze do jego pobrania,
- odchylenie standardowe czasu pobytu zadania w kolejce - statystyka komplementarna do wielkości wcześniejszej.

Statystyki kolejek generatorów przedstawiają statystyki buforów wejściowych zarządcy zadań; wartości statystyk są takie same jak dla kolejek procesorów.

Statystyki globalne zebrane w tej tabeli to summaryczne wartości pewnych parametrów systemu:

- wartość średnia - wartość średnia parametru podczas symulacji,
- odchylenie standardowe - odchylenie od wartości średniej,
- wartość maksymalna - największa wartość parametru zarejestrowana podczas symulacji.

Wygląd okienka z wykresem w raporcie dla wybranej charakterystyki przedstawia rysunek (Rys.12).



Rys. 12. Zobrazowanie dynamiki wybranej charakterystyki
 Fig. 12. Visualization of selected dynamic characteristic

Z kolei statystyki procesorów zobrazowane są w tabelach, jak przedstawiono to na rysunku Rys.13.

Numer	Średnie obciążenie (%)	Wykonane zadania	Procent wykonanych zadań (%)
0	64.33	1280	21.4047
1	48.41	1370	22.9097
2	21.07	1321	22.0903
3	15.92	1005	16.806
4	9.858	584	9.76589
5	4.195	286	4.78261
6	1.537	105	1.75585
7	0.254	26	0.434783
8	0.04678	2	0.034448
9	0.007612	1	0.0167224
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0

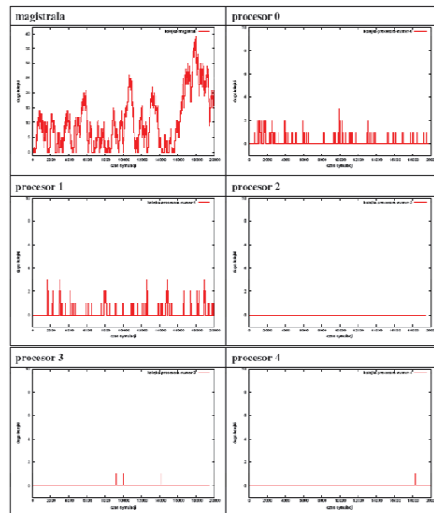
Rys. 13. Zobrazowanie statystyk procesorów
 Fig. 13. Visualization of CPU statistics

Statystyki kolejek procesorów zobrazowane są w tabelach, jak na rysunku Rys. 14.

Numer	Średnia długość kolejki	Maksymalna długość kolejki	Średni czas pobytu zadania w kolejce	Odkchylenie standardowe pobytu
0	0.4107	5	32.09	41.45
1	0.3157	8	23.05	47.33
2	0.0373	3	2.824	11.42
3	0.01313	2	1.306	5.928
4	0.006801	3	1.165	6.936
5	0.001439	2	0.5032	4.978
6	0	1	0	0
7	0	1	0	0
8	0	1	0	0
9	0	1	0	brak danych
10	brak danych	brak danych	brak danych	brak danych
11	brak danych	brak danych	brak danych	brak danych
12	brak danych	brak danych	brak danych	brak danych
13	brak danych	brak danych	brak danych	brak danych
14	brak danych	brak danych	brak danych	brak danych
15	brak danych	brak danych	brak danych	brak danych

Rys. 14. Zobrazowanie statystyk dla kolejek procesorów
 Fig. 14. Visualization of CPU queue statistics

Mozna też przedstawić wykresy zajętości kolejek, co pokazane zostało na rysunku Rys.15.



Rys. 15. Wykresy zajętości kolejek
 Fig. 15. Charts of queues load

W tabeli na rysunku Rys.16 uwidocznione są obciążenia procesorów przy zastosowaniu algorytmów: A) „Po kolei”, B) „Losowy”, C) „Losowy z wagami”, D) „Najkrótsza kolejka”, E) „Najkrótsza kolejka z wagami”.

Number procesora	A)	B)	C)	D)	E)
0	78,62	50,38	47,68	63,23	64,33
1	47,99	47,79	43,03	48,23	48,41
2	20,03	4,01	2,51	20,55	21,07
3	12,45	3,75	3,2	15,45	15,92
4	7,1	3,66	2,91	8,01	9,86
5	2,6	3,89	3,4	4,3	4,2
6	0,63	3,54	3,31	2,08	1,56
7	0,05	3,94	2,56	0,51	0,25
8	0	4,21	2,41	0,18	0,05
9	0	4,05	3,76	0,05	0,01
10	0	4,42	2,89	0	0
11	0	3,75	2,58	0	0
12	0	3,38	4,03	0	0
13	0	4,4	2,1	0	0
14	0	0,85	2,63	0	0
15	0	0,78	2,9	0	0

Rys. 16. Charakterystyki obciążenia procesorów

Fig. 16. The characteristics of CPU load

7 Uwagi końcowe

W pracy przedstawiono projekt i implementację symulatora programowego pozwalającego dla zadanych parametrów systemu wieloprocesorowego uzyskiwać oszacowania wartości wskaźników efektywności jego funkcjonowania. Pokazano sposób analizy systemów wieloprocesorowych na podstawie wyników uzyskanych z eksperymentów symulacyjnych.

Przedstawiona została tu jedynie implementacja symulatora systemów wieloprocesorowych. Następnym krokiem w badaniach będzie opracowanie planów eksperymentów symulacyjnych do badania własności tej klasy systemów. Wyniki pozwolą na ocenę funkcjonowania algorytmów rozdziału zadań między poszczególne procesory. Pozwolą również na zbadanie wrażliwości systemu na różnego typu strumienie zadań obliczeniowych. Pozwoli to też na ocenę wpływu parametrów sieci komputerowej na pracę całego systemu.

Literatura

1. Jain R.: *The Art of Computer Systems Performance Analysis. Techniques for Experimental Design, Measurement, Simulation and Modeling.* John Wiley & Sons, Inc., Littleton, Massachusetts, 1991
2. Null L. and Lobur J.: *Computer Organization and Architecture.* Jones and Bartlett Publishers, Sudbury, Massachusetts, 2003
3. Stallings W.: *Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność.* WNT, Warszawa, 2004

Streszczenie

W pracy przedstawiono projekt i implementację symulatora programowego systemu wieloprocessorowego. Omówiono charakterystyki takiego systemu. Zdefiniowano poszczególne elementy modelu systemu. Zaprezentowano możliwości symulatora w zakresie badań efektywności systemów wieloprocessorowych. Pokazano szczegółowe charakterystyki, jakie można uzyskać posługując się opracowanym symulatorem.

Simulator for multiprocessor systems efficiency investigation

Summary

In this work project and implementation of software simulator of multiprocessor system is presented. Selected characteristics of such system are discussed. Every elements of model of multiprocessor system are defined. Possibilities of simulator from the system efficiency point of view are presented. Many characteristics which can be obtained during simulation experiments of multiprocessor system exploitation are shown.