

Robert TUTAJEWICZ

EFEKTYWNOŚĆ ROZPRASZANIA OBLICZEŃ W SIECI KOMPUTEROWEJ DLA ZADAŃ WYSZUKIWANIA DANYCH

Streszczenie. Artykuł zawiera omówienie i wyniki eksperymentów pozwalających określić opłacalność wykonywania zadań wyszukiwania w bazach danych w sposób rozproszony. Do realizacji przetwarzania rozproszonego wykorzystano lokalną sieć komputerową.

EFFICIENCY OF DISTRIBUTED COMPUTING IN COMPUTERS NETWORK FOR DATA RETRIEVING TASKS.

Summary. Experiments relating to distributed tasks of retrieving in databases is presented in the paper. Profitability of this method is discussed. Tests were executed in computers network.

DIE EFFIZIENZ DER VERTEILTEN BERECHNUNGEN IM COMPUTER-NETZWERK FÜR DATENAUSSUCHAUFGABEN

Zusammenfassung. Der Artikel beinhaltet die Analyse und Ergebnisse der Versuche, die ermöglichen, die Zweckmäßigkeit der verteilten Ausführung der Datenaussuchaufgaben zu bestimmen. Das LAN-Netzwerk wurde zu diesem Zweck benutzt.

1. Wstęp

Jednym ze sposobów przyspieszenia wykonywania złożonych zadań jest podzielenie ich na mniejsze fragmenty (podzadania) i wykonywanie każdego z nich współbieżnie na innym komputerze w sieci. Artykuł ten opisuje wyniki takiego podejścia do kilku typowych zadań występujących w bazach danych, ze szczególnym uwzględnieniem złożonych zadań wyszukiwania. Jako kryterium podziału zadania na części zaproponowano wyodrębnienie w zadaniu poszczególnych operacji algebry relacji. W artykule przedstawiono przykłady podziału zadań wyszukiwania, rezultaty przeprowadzonych eksperymentów oraz wnioski z nich wynikające. Omówiono także zastosowane mechanizmy programowe umożliwiające współbieżną realizację zadań.

2. Założenia i oczekiwania

Jak już wspomniano, podzielenie złożonego zadania na fragmenty i wykonywanie kilku z tych fragmentów równolegle na kilku maszynach, powinno wydatnie zmniejszyć czas wykonania całego zadania. Pierwszym problemem, jaki pojawia się przy realizacji zadań wyszukiwania w sposób rozproszony, jest wyodrębnienie w rozwiązaniu owych niezależnych fragmentów, które można by wykonywać równolegle. Okazuje się [1], że w takim przypadku najlepiej przedstawić rozwiązanie za pomocą operatorów algebry relacji, takich jak selekcja, projekcja i łączenie relacji. Wtedy stosunkowo łatwo wydzielić niezależne od siebie fragmenty rozwiązania. Drugi, może nawet istotniejszy, problem jest związany z pracą w sieci. Otóż, realizacja zadań wyszukiwania wymaga transmisji dużej liczby danych między serwerem a komputerem wykonującym obliczenia. Współbieżna realizacja takich zadań spotęguje "tłok" w sieci, czyniąc wątpliwym osiągnięcie przyspieszenia. W tym kontekście głównym celem artykułu jest sprawdzenie możliwości uzyskania przyspieszenia w trakcie równoległej realizacji rozproszonych zadań wyszukiwania. W celu rozstrzygnięcia tej wątpliwości i oceny efektywności rozproszenia wykonano serię eksperymentów.

3. Struktura bazy

W eksperymentach wykorzystywano dane z bazy związanej z pracą dziekanatu. Baza ta składała się z kilkunastu plików. Jednakże nie wszystkie były używane. Dlatego też opis struktury tej bazy został ograniczony jedynie do plików używanych w eksperymentach. Dla ułatwienia zmieniono także nieco nazwy plików (rezygnując z ograniczenia do 8 znaków w nazwie).

- a) plik **STUDENCI**: rozmiar rekordu - 527 bajtów
- album* - numer albumu studenta - 5 bajtów
 - nazwisko* - nazwisko studenta - 25 bajtów
 - imię* - imię studenta - 25 bajtów
 - pozostałe* - pozostałe dane o studencie (w rzeczywistości wiele pól, jednak nie znaczących dla dalszych rozważań) 472 bajty.
- b) plik **STYPENDIA**: 62 bajty
- album* - numer albumu studenta - 5 bajtów
 - rok* - rok akademicki, w którym pobrano stypendium - 5 bajtów
 - miesiąc* - miesiąc, za który pobrano stypendium - 2 bajty
 - stypendium* - rodzaj pobieranego stypendium - 1 bajt
 - pozostałe* - pozostałe pola (tu nie znaczące) - 49 bajtów
- c) plik **PRZEDMIOTY**: rozmiar rekordu - 55 bajtów
- przedmiot* - skrót nazwy przedmiotu - 5 bajtów
 - nazwa* - pełna nazwa przedmiotu - 50 bajtów
- d) plik **EGZAMINY**: rozmiar rekordu - 37 bajtów
- album* - numer albumu studenta - 5 bajtów
 - rok* - rok akademicki, w którym student zdawał egzamin - 5 bajtów
 - przedmiot* - skrót nazwy przedmiotu - 5 bajtów
 - ocena* - uzyskana ocena - 3 bajty
 - pozostałe* - pozostałe pola - 19 bajtów
- e) plik **ZALICZENIA**: rozmiar rekordu - 35 bajtów
- W pliku tym występują podobne pola jak w pliku EGZAMINY, z tą jedynie różnicą, że pola *pozostałe* zajmują 17 bajtów.

W badaniach używano dwu wersji opisywanej bazy, różniących się liczbą rekordów w poszczególnych plikach.

4. Sformułowanie zadania i omówienie sposobu rozwiązania za pomocą operatorów algebry relacji

Zadanie pierwsze

Załóżmy, że naszym zadaniem będzie wyszukanie nazwisk wszystkich studentów pobierających w październiku 1990 r. stypendium socjalne, którzy podczas jakiegokolwiek egzaminu uzyskali ocenę niedostateczną. W pliku wynikowym powinny się znaleźć obok nazwiska studenta także nazwy przedmiotów, z których uzyskał on ocenę negatywną na egzaminie.

W rozwiązaniu posłużymy się następującymi operacjami algebry relacji: projekcja relacji R na zbiór atrybutów X , oznaczana jako $\Pi_X(R)$, selekcja relacji R względem danego warunku e , co zapisujemy jako $\sigma_e(R)$, złączenie relacji R i S , zapisywane jako $R \bowtie S$.

Znalezienie nazwisk studentów spełniających opisane wcześniej warunki można zapisać za pomocą algorytmu złożonego z następujących kroków:

1. Wyszukać w pliku¹ (relacji) STYPENDIA numery albumu studentów, którzy w październiku 1990 roku pobierali stypendium socjalne. Wynik zapisać w pliku roboczym WRK1.
2. Połączyć pliki WRK1 i STUDENCI według numeru albumu, co pozwoli uzyskać nazwiska studentów pobierających stypendium socjalne w październiku 1990 r. Wynik zapisać w pliku roboczym WRK2.
3. Wybrać z pliku EGZAMINY rekordy dla, których ocena wynosi 2.0 i wynik umieścić w pliku WRK3.
4. Połączyć pliki WRK3 i PRZEDMIOTY i tym samym otrzymać plik WRK4 zawierający negatywne wyniki z egzaminów wraz z pełnymi nazwami przedmiotów.
5. Połączyć pliki WRK2 i WRK4, aby znaleźć nazwiska studentów pobierających w październiku 1990 roku stypendium socjalne (dane z pliku WRK2), którzy mieli negatywne wyniki z egzaminów (dane z pliku WRK4). Wynik tej operacji zapisany w pliku WRK5 jest odpowiedzią w postawionym zadaniu wyszukiwania.

¹ Pojęcia krotki i rekordu oraz pliku i relacji są tu stosowane zamiennie. Relacja jest jednym z pojęć relacyjnego modelu danych oznaczającym zbiór krotek. W praktyce każdej relacji, jako pojęciu abstrakcyjnemu, odpowiada jakiś plik złożony z rekordów (będących odpowiednikiem pojęcia krotki). Podobnie utożsamia się też pojęcia atrybutu i nazwy pola.

Rozwiązanie zadania można przedstawić za pomocą operatorów algebry relacji w następujący sposób:

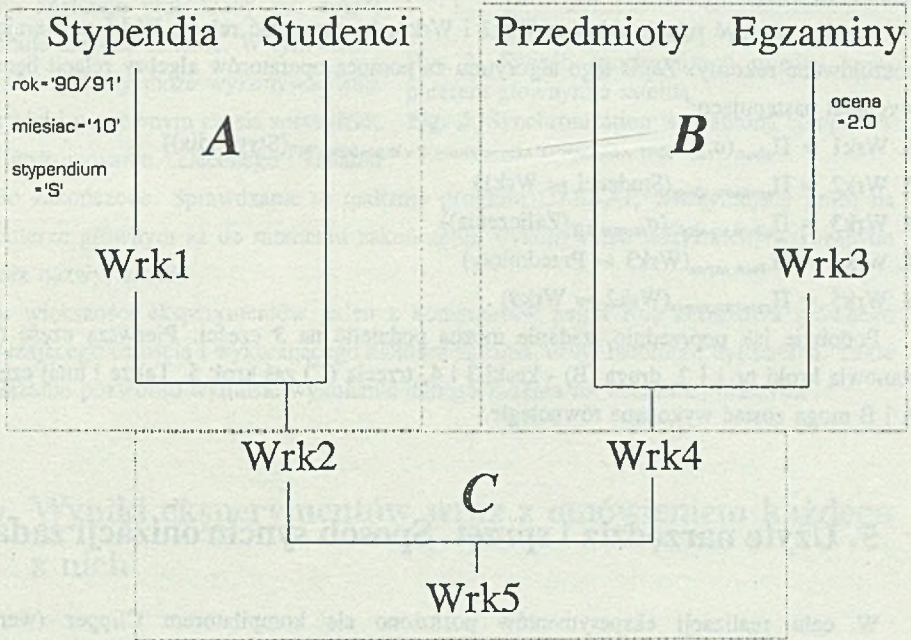
$$\text{Wrk1} = \Pi_{\text{album}}(\sigma_{\text{rok}='90/91' \wedge \text{miesiac}='10' \wedge \text{stypendium}='S'}(\text{Stypendia}))$$

$$\text{Wrk2} = \Pi_{\text{nazwisko, album}}(\text{Studenci} \bowtie \text{Wrk1})$$

$$\text{Wrk3} = \Pi_{\text{album, przedmiot}}(\sigma_{\text{ocena}='2.0'}(\text{Egzaminy}))$$

$$\text{Wrk4} = \Pi_{\text{album, nazwa}}(\text{Wrk3} \bowtie \text{Przedmioty})$$

$$\text{Wrk5} = \Pi_{\text{nazwisko, nazwa}}(\text{Wrk2} \bowtie \text{Wrk4})$$



Rys. 1. Rozwiązanie zadania pierwszego
Fig. 1. Solution of first task

Rozwiązaniem zadania jest zatem relacja Wrk5, zawierająca nazwiska poszukiwanych studentów. Zauważmy, że rozwiązanie to można podzielić na trzy fragmenty, przy czym dwa pierwsze są w ogóle niezależne od siebie. Wyraźnie to widać po przedstawieniu rozwiązania w postaci graficznej. Pierwszy fragment (A), to znalezienie studentów pobierających w październiku 1990 r. stypendium socjalne, drugi (B) - wyszukanie studentów, którzy kiedykolwiek otrzymali na egzaminie ocenę niedostateczną (znajdują się oni w relacji Wrk4). Ostatni etap, to połączenie obydwu relacji (Wrk2 i Wrk4), w wyniku którego otrzymujemy interesujące nas dane.

Zadanie drugie

Podać nazwiska studentów, którzy w grudniu 1990 roku otrzymywali stypendium socjalne lub motywacyjne, a przynajmniej raz uzyskali zaliczenie równe 3.0.

Aby znaleźć rozwiązanie tego zadania, należy postępować podobnie jak poprzednio. Dlatego też całość rozumowania została przedstawiona dość skrótowo. Najpierw należy znaleźć nazwiska studentów pobierających w grudniu 1990 r. stypendium socjalne lub motywacyjne (kroki 1 i 2). Następnie wyszukać wszystkie rekordy relacji ZALICZENIA, w których ocena wynosi 3.0 (krok 3); określić, z jakich to było przedmiotów (krok 4) i wreszcie połączyć relacje robocze Wrk2 i Wrk4, by otrzymać relację Wrk5 zawierającą poszukiwane rekordy. Zapis tego algorytmu za pomocą operatorów algebry relacji będzie wyglądał następująco:

$$1. \text{Wrk1} = \Pi_{\text{album}}(\sigma_{\text{rok} = '90/91' \wedge \text{miesiac} = '12' \wedge (\text{stypendium} = 'S' \vee \text{stypendium} = 'M')}(\text{Stypendia}))$$

$$2. \text{Wrk2} = \Pi_{\text{nazwisko, album}}(\text{Studenci} \bowtie \text{Wrk1})$$

$$3. \text{Wrk3} = \Pi_{\text{album, przedmiot}}(\sigma_{\text{ocena} = '3.0'}(\text{Zaliczenia}))$$

$$4. \text{Wrk4} = \Pi_{\text{album, nazwa}}(\text{Wrk3} \bowtie \text{Przedmioty})$$

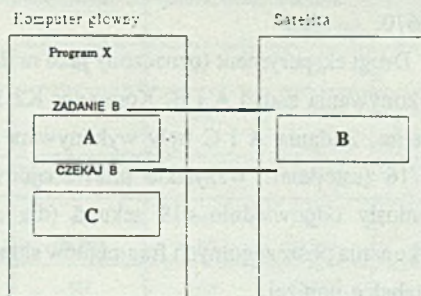
$$5. \text{Wrk5} = \Pi_{\text{nazwisko, nazwa}}(\text{Wrk2} \bowtie \text{Wrk4})$$

Podobnie jak poprzednio, zadanie można podzielić na 3 części. Pierwszą część (A) stanowią kroki nr 1 i 2, drugą (B) - kroki 3 i 4, trzecią (C) zaś krok 5. Także i tutaj części A i B mogą zostać wykonane równolegle.

5. Użyte narzędzia i sprzęt. Sposób synchronizacji zadań

W celu realizacji eksperymentów posłużono się kompilatorem Clipper (wersja Summer'87). Przeprowadzono je w sieci Novell, przy średnim obciążeniu sieci, na komputerach klasy PC z procesorami 80386SX. W większości eksperymentów używano dwóch komputerów, różniących się częstotliwością zegara. Pierwszy, zwany dalej K1/16, był taktowany zegarem o częstotliwości 16 MHz, drugi (K2/18) nieco szybszy - 18 MHz. W eksperymentach wykorzystano pakiet kilku programów, umożliwiający zdalne uruchamianie programów na komputerach połączonych w sieć [3]. Idea działania tego pakietu opiera się na wyróżnieniu jednego komputera w sieci jako tzw. komputera głównego, rozdzielającego zadania między pozostałe komputery, zwane dalej satelitami. Komputer główny wysyła opis zadania (w postaci nazwy zadania oraz nazwy programu do uruchomienia wraz z ewentualnymi parametrami), które jest następnie wykonywane przez któryś z satelitów.

Rysunek ilustruje sposób synchronizacji między komputerem głównym a satelitą. Na każdym komputerze-satelicie uruchamiany jest program, który oczekuje na zlecenia przekazywane w postaci opisu zadania z komputera głównego i wykonuje je. Opis ten jest przesyłany do satelity za pomocą programu ZADANIE. Następnie satelita analizuje przekazany mu opis i wykonuje zlecone zadanie. W tym czasie komputer główny może wykonywać inne czynności i po pewnym czasie sprawdzić, czy wykonywanie zleconego zadania zostało zakończone. Sprawdzanie to realizuje program CZEKAJ, wstrzymujący pracę na komputerze głównym aż do momentu zakończenia wykonywania wszystkich, wskazanych poprzez nazwy, zadań.



Rys. 2. Sposób synchronizacji między komputerem głównym a satelitą

Fig. 2. Synchronization way among computers

W większości eksperymentów jeden z komputerów pełnił rolę komputera głównego zarządzającego całością i wykonującego niektóre zadania, drugi natomiast był satelitą. Takie rozwiązanie pozwoliło wymusić wykonanie danego zadania na wskazanej maszynie.

6. Wyniki eksperymentów wraz z omówieniem każdego z nich

W tej części zostaną omówione szczegółowo poszczególne eksperymenty i przedstawione ich wyniki. Większość eksperymentów przeprowadzono dla dwu różnych zestawów plików z danymi, różniących się ilością rekordów (np. plik opisujący wyniki egzaminów w jednym przypadku zawierał 7007 rekordów, a w drugim 12210).

Przypomnijmy, że rozwiązanie pierwszego zadania dało się rozdzielić na trzy fragmenty:

- A - znalezienie studentów, pobierających w październiku 1990 stypendium socjalne,
- B - wyszukanie studentów, którzy kiedykolwiek na egzaminie mieli ocenę niedostateczną,
- C - połączenie otrzymanych wyników (relacji), by znaleźć nazwiska studentów spełniających obydwa warunki.

Pierwszą wersją rozwiązania jest rozwiązanie tradycyjne (sekwencyjne, oznaczone w tabelce numerem 1), w całości wykonane na jednym komputerze (K2/18). W tym przypadku

wykonanie całości zadania zajęło odpowiednio: dla mniejszej bazy - 513 sekund, dla większej - 1670.

Drugi eksperyment (oznaczony jako nr 2), to rozwiązanie wykorzystujące zrównoleglenie wykonywania zadań A i B. Komputer K2/18 pracował jako komputer główny, K1/16 jako satelita. Zadania A i C były wykonywane na K2/18, zadanie B natomiast na komputerze K1/16 (satelicie). Wszystkie pliki znajdowały się na dysku serwera. Czasy wykonania wyniosły odpowiednio 419 sekund (dla mniejszej bazy) i 1368 (dla większej). Czasy wykonania poszczególnych fragmentów składających się na rozwiązanie zostały zamieszczone w tabelce poniżej.

Kolejne dwa eksperymenty (3 i 4) różnią się od poprzednich tylko lokalizacją plików z danymi. Eksperyment numer 3 odpowiada eksperymentowi nr 1, 4 natomiast eksperymentowi nr 2. Tym razem zadanie A wykonano wykorzystując pliki umieszczone na dysku lokalnym komputera K2/18. Przy takim rozwiązaniu nie dostrzeżono jakiegokolwiek różnicy w czasach wykonania między tą a poprzednią grupą.

Operacja	Mniejsza baza				Większa baza	
	1	2	3	4	1	2
A	213	213	212	212	769	779
B	129	149 ^{wn}	128	150 ^{wn}	372	435 ^{wn}
A i B (razem)	340 (100%)	213 (62.65%)	339 (100%)	212 (62.54%)	1143 (100%)	780 (68.24%)
C	171	203 ^{wn}	171	202 ^{wn}	527	588 ^{wn}
Całe zadanie	513 (100%)	419 (81.68%)	512 (100%)	419 (81.84%)	1670 (100%)	1368 (81.92%)

Operacje wykonywane na komputerze K1/16 wyróżniono przez umieszczenie indeksu (W) obok czasu wykonania operacji. Wszystkie czasy podano w sekundach.

Druga grupa eksperymentów, związana z drugim zadaniem wyszukiwania, dotyczyła badania wpływu rozmieszczenia poszczególnych podzadań na czas wykonania całości. Wszystkie pliki znajdowały się na dysku serwera sieci. W zamieszczonej tabelce znajdują się wyniki przeprowadzonych eksperymentów.

Operacja	Mniejsza baza				Większa baza		
	A1	A2	A3	A4	A1	A2	A4
A	320	323	385 ⁽⁹¹⁾	392 ⁽⁹¹⁾	1267	1289	1524 ⁽⁹¹⁾
B	394	456 ⁽⁹¹⁾	408	391	1417	1667 ⁽⁹¹⁾	1421
A i B (razem)	714 (100%)	462 (64.7%)	408 (57.1%)	402 (56.3%)	2687 (100%)	1667 (62.04%)	1524 (56.72%)
C	285	334 ⁽⁹¹⁾	335 ⁽⁹¹⁾	287	1424	1459 ⁽⁹¹⁾	1344
Całe zadanie	995 (100%)	857 (86.13%)	742 (74.57%)	679 (68.24%)	4034 (100%)	3128 (77.54%)	2867 (71.07%)

Operacje wykonywane na komputerze K1/16 wyróżniono przez umieszczenie indeksu (W) obok czasu wykonania operacji. Wszystkie czasy podano w sekundach.

Pierwszy eksperyment (oznaczony w tabelce jako A1), to tradycyjne rozwiązanie sekwencyjne, program w całości uruchomiony na komputerze K2/18. Łączny czas wykonania całego zadania wyniósł tu odpowiednio dla mniejszej bazy 995 s, dla większej 4034 s. Eksperyment A2 to zadanie wykonane z wykorzystaniem 2 maszyn, przy czym komputer K2/18 pracował jako satelita (wykonano na nim część A), a K1/16 jako komputer główny (części B i C). W wyniku rozproszenia zadania udało się nieco zmniejszyć czas wykonania całości w stosunku do wersji sekwencyjnej. Zysk wynosił odpowiednio 2 min 18 s (dla mniejszej bazy) i 15 min 6 s (dla większej).

Kolejny eksperyment (A3) różnił się od poprzedniego rozmieszczeniem poszczególnych podzadań na komputerach. Wykonano go tylko dla mniejszej bazy. Tym razem zadanie B wykonano na K2/18, a zadania A i C na komputerze K1/16, który nadal był komputerem głównym. Łączny czas wykonania takiego zadania wyniósł 742 s, co stanowiło 74.57% czasu wykonania dla wersji sekwencyjnej (A1), a zysk wyniósł już 4 min 13 s. Zauważmy, że osiągnięto poprawę jedynie dzięki zamianie zadań wykonywanych równolegle na poszczególnych maszynach.

By jeszcze bardziej zmniejszyć czas wykonania całości, należało zadanie C (które nie może być wykonywane równolegle z pozostałymi) wykonać na komputerze szybszym, tzn. K2/18. Eksperyment A4 odpowiada takiemu właśnie rozwiązaniu, podzadania B i C wykonywane są na komputerze K2/18, pracującym jako komputer główny, zadanie A wykonuje natomiast satelita - komputer K1/16. Dla takiego rozmieszczenia uzyskano następujące wyniki:

- dla mniejszej bazy: czas wykonania całości - 679 s, co stanowi 68.24% czasu dla wersji sekwencyjnej, czas wykonania został zatem zmniejszony o 5 min 15 s.

- dla większej bazy: czas wykonania całości - 2867 s, czyli 71.07% czasu dla wersji sekwencyjnej, zmniejszenie czasu wykonania o 19 min 27 s.

7. Wnioski dotyczące efektywności metody

Podsumujmy dotychczasowe wyniki, umieszczając w jednej tabeli, dla każdego z eksperymentów, czasy wykonania zadań (a właściwie tylko ich części: podzadań A i B) w wersji sekwencyjnej i równoległej oraz wyrażony w procentach stosunek czasu wykonania podzadań A i B w wersji równoległej do czasu wykonania tychże podzadań w wersji sekwencyjnej (obrazujący na ile zrównoleglenie niektórych części zadania, pozwala przyspieszyć wykonanie całości; występujący w tabeli pod nazwą procentowego współczynnika zysku).

Eksperyment	Czas wykonania zadań A i B w wersji sekwencyjnej	Czas wykonania zadań A i B w wersji równoległej	Procentowy współczynnik zysku
Zadanie pierwsze			
1 i 2 (mniejsza baza)	340	213	62.65%
3 i 4 (mniejsza baza)	339	212	62.54%
1 i 2 (większa baza)	1143	780	68.24%
Zadanie drugie (jako wersja sekwencyjna zawsze zadanie A1)			
A2 (mniejsza baza)	714	462	64.7%
A3 (mniejsza baza)	714	408	57.1%
A4 (mniejsza baza)	714	402	56.3%
A2 (większa baza)	2687	1667	62.04%
A4 (większa baza)	2687	1524	56.72%

Czasy wykonania podano w sekundach.

Zauważmy, że w przypadku niektórych eksperymentów współczynnik zysku osiągnął wartości w granicach 55 - 60 procent, podczas gdy teoretycznie jego wartość minimalna może wynieść 50 procent. Uzyskane wartości są zatem bliskie wartości minimalnej, co dowodzi opłacalności takiej metody wyszukiwania. W przypadku bardziej złożonych zadań, gdzie możliwe byłoby wykonywanie równoległe nie dwu, lecz większej ilości podzadań, uzyskane wyniki powinny być jeszcze nieco lepsze.

W wyniku przeprowadzonych eksperymentów okazało się, że zupełnie nie ma znaczenia lokalizacja plików na dysku lokalnym, czy też na sieciowym (oczywiście przy założeniu, że parametry obydwu dysków są do siebie zbliżone i dysk sieciowy nie jest nadmiernie obciążony przez inne zadania; porównaj wyniki dla eksperymentów 1, 2 i 3, 4). Duży wpływ na czas wykonania całego zadania może mieć natomiast umiejętne rozdzielanie podzadań pomiędzy komputery o różnych mocach obliczeniowych. Na przykład, tylko zamieniając rolami używane w eksperymentach dwa komputery, udało się zmniejszyć czas wykonania całego zadania z 857 s do 679 (eksperymenty A2 i A4, wyniki dla mniejszej bazy), a części wykonywanych równoległe (czyli podzadań A i B) z 462 do 402 s. Zaleca się, dążyć do tego, aby czasy wykonania poszczególnych podzadań, wykonywanych jednocześnie, jak najmniej różniły się pomiędzy sobą.

8. Wyniki dotyczące zrównoleglenia operacji indeksowania

Jedną z bardziej czasochłonnych operacji w programach wykorzystujących bazy danych jest operacja indeksowania. Przykładowo w systemie obsługującym dzickanat, każdorazowo w momencie rozpoczynania pracy odbywa się indeksowanie kilkunastu plików. Podjęto więc próbę rozdzielania tej operacji na dwa komputery. Na jednym wykonywano indeksowanie pliku Zaliczenia, na drugim natomiast wszystkie pozostałe operacje indeksowania. Taki sposób podziału wynikał z faktu, że wspomniane indeksowanie pliku Zaliczenia zajmowało w przypadku wykonywania wszystkich operacji na jednym komputerze ponad 56 procent czasu wykonywania całości. Wykonanie wszystkich operacji indeksowania na jednym komputerze zajęło 187 s, w przypadku rozdzielenia na dwa komputery czas trwania całości wyniósł 121 s. Rozproszenie pozwoliło więc wyraźnie zmniejszyć czas wykonania całości (do ok. 65% czasu wykonania dla wersji sekwencyjnej) także i w tym przypadku.

9. Jak daleko można posunąć się z rozpraszaniem

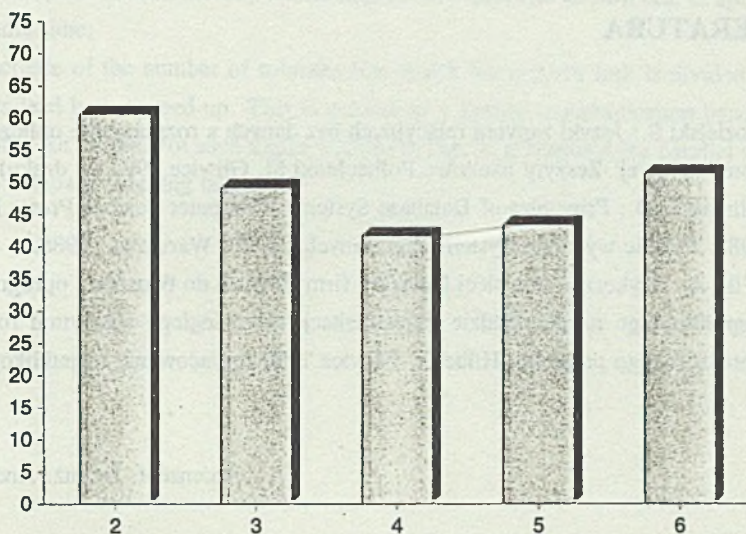
We wszystkich omawianych do tej pory eksperymentach zadanie rozdzielano na zaledwie dwa komputery. Powstaje zatem pytanie, czy rozdzielanie zadania na większą liczbę maszyn będzie zawsze prowadziło do przyspieszenia wykonania całości zadania. Zauważmy, że realizacja zadań wyszukiwania bądź indeksowania wymaga przesyłu dużej ilości danych

poprzez sieć. Przy ciągłym zwiększaniu ilości równolegle wykonywanych zadań, sieć staje się "wąskim gardłem" całego systemu. Ostatnia seria eksperymentów pozwoliła określić, począwszy od ilu zadań wykonywanych równolegle dalsze rozpraszanie przestaje być opłacalne ze względu na ograniczoną przepustowość sieci. Aby wyeliminować wpływ innych czynników, wszystkie zadania były identyczne i wykonane były na maszynach o identycznych parametrach (386SX, 2 MB RAM, zegar 33 MHz). Przeprowadzone eksperymenty polegały na tym, że jednocześnie na kilku komputerach uruchamiano identyczne zadania indeksowania kopii pliku Studenci i mierzono czas, jaki upłynął do momentu zakończenia wszystkich zadań. Liczba komputerów, na których uruchamiano owe zadania, zmieniała się od jednego do sześciu. W tabeli zestawiono wyniki tych eksperymentów. Uzyskane czasy porównano z czasami wykonania takiej samej ilości zadań na jednej maszynie w sposób sekwencyjny. Stosunek tych czasów, zwany dalej procentowym współczynnikiem przyspieszenia umieszczono w ostatniej kolumnie tabeli.

Liczba komputerów	Czas wykonania wersji sekwencyjnej	Czas wykonania wersji równoległej	Współczynnik procentowy
1	17 s	17.00 s	100.0%
2	34 s	20.59 s	60.57%
3	51 s	25.05 s	49.12%
4	68 s	28.35 s	41.69%
5	85 s	36.91 s	43.42%
6	102 s	52.45 s	51.42%

Łatwo zauważyć, że istnieje pewna optymalna liczba komputerów, na które można rozproszyć wykonanie całego zadania (w naszym przypadku 4, zob. także rys. 3). Dalsze rozdzielanie nie tylko nie poprawia wydajności, ale może nawet ją pogorszyć. Przyczyną takiego zjawiska jest przepustowość sieci. Oczywiście, uzyskana wartość optymalna może się nieco różnić w każdym przypadku (w zależności od parametrów komputerów włączonych

w sieć i parametrów samej sieci). Jednakże sama zasada istnienia takiej wartości wydaje się niezmienna.



Rys. 3. Zależność czasu wykonania równoległej operacji indeksowania od ilości zadań (w stosunku do realizacji sekwencyjnej)

Fig. 3. Dependence of executing time of parallel indexing upon the task number

10. Podsumowanie wyników

Wyniki opisanych tu badań wskazują na to, że możliwe jest znaczne przyspieszenie wykonywania złożonych zadań w bazach danych poprzez rozdzielenie ich na mniejsze podzadania i wykonywanie tych podzadań równoległe na kilku maszynach. Istotne jest jednak, by tak dzielić zadanie, aby czasy wykonania wszystkich równoległe wykonywanych podzadań były zbliżone. Dodajmy, że spełnienie tego wymagania nie zawsze jest możliwe, dlatego też uzyskane przyspieszenie nie zawsze będzie znaczące.

Ponadto okazało się, że najbardziej opłacalne jest najczęściej rozdzielanie zadania na zaledwie kilka podzadań wykonywanych równolegle (zob. punkt poprzedni). Zauważmy także, że nawet dla dość złożonych zadań wyszukiwania, najczęściej możliwe jest rozdzielanie ich na niewielką ilość podzadań.

LITERATURA

- [1] Kozielski S.: Języki zapytań relacyjnych baz danych a rozpraszanie obliczeń w sieci komputerowej. Zeszyty naukowe Politechniki Śl. Gliwice 1993 (w druku).
- [2] Ullman J. D.: Principles of Database Systems. Computer Science Press, Rockville, 1983. Polskie wydanie: Systemy baz danych. WNT, Warszawa 1988.
- [3] Wilk A.: Wykorzystanie sieci NetWare firmy Novell do tworzenia oprogramowania współbieżnego na przykładzie implementacji równoległego algorytmu rozwiązania uproszczonego problemu Hilberta. Gliwice 1990 (opracowanie niepublikowane).

Recenzent: Dr inż. Andrzej Wilk

Wpłynęło do Redakcji 19 listopada 1993 r.

Abstract

The division of a complex task into a number of subtask and parallel realization of them should speed up execution of whole task. This article presents the results of application of this idea to the retrieving tasks in databases. The method of dividing the task into subtasks is given. The separation of operations of relations algebra from complicated task is presented as the division criterion. Figure 1 illustrates how to divide a complex retrieving task into subtasks. Out of the task A, B and C, the tasks A and B can be executed in parallel, whereas the task C can be executed only after termination of A and B. Then the synchronization method is presented (figure 2).

The next section discusses the experimental results. The execution times of each subtask are presented in the tables for more realization of the complex task.

We have found that

- a proper division of the a complex retrieving task into subtask can speed up its execution substantially (e.g. the time to execute the task sequentially is 995 secs, and in parallel 679 secs; note that only the tasks A and B are executed in parallel);
- an important issue is load balancing, i.e. to assign the computations to processors of different speed in such the way that each processor executes its own task in aproximately the same time;
- the increase of the number of subtasks into which the original task is divided, does not always lead better speed-up. This is caused by a limited communication bandwidth of a network. An example of such a case is given in Fig. 3. It concerns the parallel realization of the database indexing tasks.