

Piotr BAJERSKI
Miroslaw CHŁOPEK
Henryk JOSIŃSKI
Stanisław KOZIELSKI

RÓWNOLEGŁE WYSZUKIWANIE W ROZPROSZONEJ BAZIE DANYCH W LOKALNEJ SIECI KOMPUTEROWEJ

Streszczenie. W pracy przedstawiono rozwiązanie eksperymentalnego systemu umożliwiającego równoległą organizację obliczeń dla przyspieszenia złożonego wyszukiwania w bazie danych. Baza ta zawiera dane o publikacjach naukowych wraz z tekstami streszczeń tych publikacji. Baza została utworzona jako baza rozproszona na dwóch serwerach systemu Ingres. Do organizacji rozpraszania obliczeń w sieci stacji roboczych Sun i sterowania ich równoległą realizacją wykorzystano system Linda.

PARALLEL DATA RETRIEVAL IN A DISTRIBUTED DATABASE IN A LOCAL NETWORK

Summary. The paper presents an experimental data retrieval system using distributed computations for speeding up execution of complex queries against database which containing information about scientific papers. The database is distributed using two Ingres servers. Experiments were performed using a cluster of Sun workstations. The coordination framework of the system is written using the Linda System.

PARALLELES AUSSUCHEN IN VERTEILTER DATENBANK IM LOKALEN NETZWERK

Zusammenfassung. Im Artikel wurde die Lösung eines experimentellen Systems für parallele Ausführung von Berechnungen zur Beschleunigung des komplexen Aussuchens in einer Datenbank dargestellt. Diese Datenbank enthält Daten über wissenschaftliche Veröffentlichungen inkl. derer Zusammenfassungen und funktioniert als verteilte Datenbank, die auf zwei Ingres-Server basiert. Zur Organisation der Verteilung von Berechnungen im Sun-Rechnernetzwerk sowie zur Steuerung ihrer parallelen Ausführung wurde das Linda-System ausgenutzt.

1. Wprowadzenie

Rozwiązanie wielu aktualnych zadań informatycznych wymaga zastosowania odpowiednio dużych mocy obliczeniowych. Jedną z możliwych dróg zaspokojenia takich potrzeb - obok zakupu komputera dużej mocy lub komputera równoległego - jest wykorzystanie sieci komputerowej do obliczeń równoległych.

Prace nad realizacją obliczeń w takim środowisku prowadzone są intensywnie w wielu ośrodkach [2, 6]. W publikowanych opracowaniach zwraca się uwagę, że uzyskanie efektywnych rozwiązań, a ściślej - uzyskanie znaczących przyspieszeń dla równoległej realizacji programów w sieci może mieć miejsce jedynie dla złożonych czasowo zadań, które po rozproszeniu na poszczególne stacje sieci wymagają minimalnej komunikacji między sobą. Problem przedstawiony w niniejszej pracy należy do zadań właśnie tego typu. Procesy analizujące dane o publikacjach są bowiem rozpraszane wśród dostępnych stacji sieci i niezależnie realizują swe zadania na odrębnych kopiach bazy danych. Do rozwiązania omawianych w pracy zadań wykorzystano system Linda [1, 4].

2. Ogólna charakterystyka systemu przeszukiwania abstraktów

W niniejszej pracy przez *abstrakt* rozumiany jest zwięzły opis publikacji. Zawiera on: nazwiska autorów, tytuł, nazwy tematów, których problematyka jest poruszana, słowa kluczowe, zestaw terminów nie podlegających klasyfikacji oraz streszczenie publikacji.

2.1. Organizacja bazy danych zawierającej abstrakty

W celu przeprowadzenia badań utworzono przy wykorzystaniu Systemu Zarządzania Relacyjnymi Bazami Danych INGRES 6.4 bazę danych zawierającą abstrakty artykułów z dziedziny informatyki. Baza ta składa się z następujących tabel:

a) tabeli R1 zawierającej opisy publikacji (opis atrybutów znajduje się w tab. 1).

Tabela 1

Struktura tabeli R1

Nazwa atrybutu	Typ atrybutu	Opis atrybutu
ID_ABSTR	integer4	numer identyfikacyjny abstraktu
TYTUL	char(200)	tytuł artykułu
AUTORZY	char(80)	autor (autorzy) artykułu
KLASYFIKACJA	char(70)	zestaw kodów grup tematycznych, których problematyka jest poruszana w artykule
SLOW_KLUCZ	char(240)	zestaw słów kluczowych
DOW_TERMINY	char(450)	zestaw dowolnych słów
ABSTR_DLUG	integer4	długość streszczenia wyrażona w bajtach
FR_1	char(950)	pierwszy fragment streszczenia

b) tabeli R2 zawierającej kolejne fragmenty streszczenia (opis atrybutów znajduje się w tab. 2).

Tabela 2

Struktura tabeli R2

Nazwa atrybutu	Typ atrybutu	Opis atrybutu
ID_ABSTR	integer4	numer identyfikacyjny abstraktu
NR_REK	integer4	numer kolejnego fragmentu streszczenia
TEKST	char(1995)	treść fragmentu streszczenia

c) tabeli R3 zawierającej słownik grup tematycznych (opis atrybutów znajduje się w tab. 3).

Na określenie szerokości atrybutów FR_1 (R1) oraz TEKST (R2) wpłynęło wymaganie systemu INGRES, by rozmiar pojedynczego rekordu tabeli nie przekroczył wielkości strony danych, która wynosi 2048 bajtów, przy czym 40 bajtów zarezerwowanych jest na informacje systemowe. Dla atrybutów nie stanowiących kluczy w poszczególnych tabelach dopuszczono wystąpienie wartości pustej NULL.

Tabela 3

Struktura tabeli R3

Nazwa pola	Typ atrybutu	Opis atrybutu
KOD_GRUPY	char(7)	kod grupy tematycznej
TEMAT	char(80)	nazwa grupy tematycznej

Pojedynczy rekord tabeli R1 zawiera informacje o artykule, z którego pochodzi streszczenie, a także początkowy fragment tego streszczenia. Jeśli składa się ono z liczby znaków nie przekraczającej 950, to mieści się w całości w jednym rekordzie. W sytuacji przeciwnej pozostała część streszczenia zapisywana jest w kolejnych rekordach tabeli R2 (atrybut TEKST) charakteryzujących się tym samym identyfikatorem (ID_ABSTR) i następującymi po sobie liczbami naturalnymi (począwszy od 1) określającymi numer rekordu (NR_REK). Problematyka poruszana w danym artykule może dotyczyć wielu zagadnień należących do określonych grup tematycznych, które są identyfikowane przez kod złożony z ciągu liter i cyfr.

2.2. Wyszukiwanie abstraktów

Możliwe jest wyszukiwanie abstraktów na podstawie różnych kryteriów. Zadania wyszukiwania mogą dotyczyć autorów, tytułów lub słów kluczowych. Można również szukać artykułów, których streszczenia zawierają podane słowa. Przy tego rodzaju wyszukiwaniach możliwe jest przyspieszenie dostępu do danych przez zastosowanie indeksów lub list inwersyjnych. Istnieją również specjalne metody polegające na klasyfikacji słów kluczowych [5]. Jednak ich użycie staje się problematyczne, gdy uwzględnia się zmianę formy fleksyjnej, zmianę części mowy lub błędy w zapisie. W pracy rozpatruje się metodę wyszukiwania dokumentów na podstawie podobieństwa między zestawem słów zadany przez użytkownika (nazywanym dalej *wzorcem*) a zbiorem słów tworzących jeden z elementów opisu dokumentu.

Niezależnie wyznacza się podobieństwo słów tworzących wzorec do następujących elementów opisu dokumentu: tytuł, streszczenie, grupy tematyczne, słowa kluczowe, dowolne terminy. W następnym kroku sumuje się z odpowiednimi wagami otrzymane współczynniki. Wagi mogą posłużyć na przykład do określenia, że wystąpienie podobieństwa w tytule jest bardziej znaczące niż wystąpienie takiego samego podobieństwa w tekście streszczenia. Wagi te mogą być modyfikowane przez użytkownika.

Podobieństwo dwóch zbiorów słów wyznacza się sumując współczynniki podobieństwa wszystkich słów z jednego zbioru do wszystkich słów z drugiego zbioru. Współczynnik podobieństwa między dwoma wyrazami jest wyznaczany w następujący sposób:

- a) jeżeli ciągi znaków są identyczne, współczynnik podobieństwa wynosi A,
- b) w przypadku gdy jeden z nich jest podciągiem drugiego, współczynnik podobieństwa wynosi B,
- c) jeżeli żaden z poprzednich przypadków nie zachodzi, współczynnik podobieństwa wyznaczany jest na podstawie kosztu przekształcenia jednego ciągu w drugi.

Koszt przekształcenia ciągów jest równy minimalnemu kosztowi ciągu przekształceń elementarnych (wstawień, usunięć i zamian liter) potrzebnych do transformacji jednego ciągu w drugi (jest on znany pod nazwą *dystansu Levensteina*). Koszt jest tym większy, im bardziej ciągi się różnią. Przy jednakowym koszcie każdej z operacji elementarnych, wynoszącym 1, jest on co najwyżej równy liczbie znaków dłuższego z ciągów. Koszt równy 0 oznacza, że ciągi są tożsame.

Dla przypadku c współczynnik podobieństwa jest wyznaczany na podstawie poniższej formuły:

$$C * \left(\frac{l-k}{l}\right)^2$$

gdzie: k jest otrzymanym kosztem przekształcenia, l oznacza liczbę znaków dłuższego z wyrazów, a C jest wagą definiowaną przez użytkownika. Tak więc współczynnik podobieństwa może przyjmować wartości z zakresu od 0 do C .

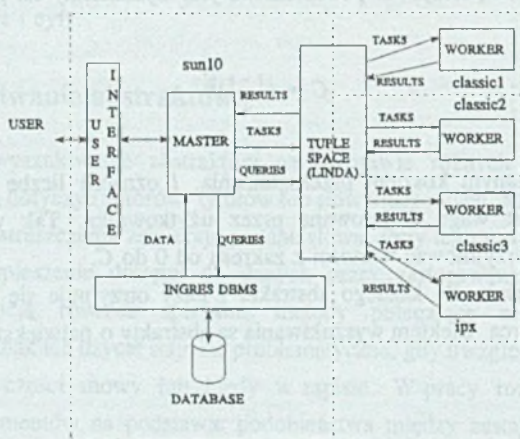
Przy takim podejściu dla każdego abstraktu z bazy otrzymuje się współczynnik jego podobieństwa do wzorca. Efektem wyszukiwania są abstrakty o największym współczynniku podobieństwa.

3. Organizacja wyszukiwania

Część programu odpowiedzialna za synchronizację procesów i wymianę danych między nimi została napisana w języku C-Linda. Obliczenia numeryczne zostały zaprogramowane w języku C. System Linda oparty jest na modelu współdzielonej asocjacyjnej pamięci (nazywanej *przestrzenią krotek*) i udostępnia operacje umożliwiające zapis danych do przestrzeni krotek, pobranie i odczyt danych z tej przestrzeni oraz tworzenie nowych procesów. Zapisywane i odczytywane ciągi danych nazywane są *krotkami*.

W rozważanym zadaniu wyszukiwania należy przejrzeć całą bazę danych i każdy rekord porównać z zadaniem wzorcem. Ze względu na charakter porównywania nie można zastosować żadnej ze znanych metod indeksacji. Przy dekompozycji problemu przyjęto, że elementarnym podzadaniem jest wyznaczenie współczynnika podobieństwa danego abstraktu do wzorca. Otrzymane w ten sposób podzadania są niezależne.

Program oparto na schemacie "nadzorca-wykonawca". Proces nadzorcy był odpowiedzialny za: obsługę dialogu z użytkownikiem (pobranie wzorca wyszukiwania), umieszczenie wzorca w przestrzeni krotek, utworzenie wykonawców, nadzorowanie obliczeń, rozpoznanie końca obliczeń, zakończenie procesów wykonawców, zebranie wyników częściowych i wyprowadzenie zbioru abstraktów będącego rezultatem wyszukiwania. Utworzono dwa typy wykonawców. Wykonawcy pierwszego typu zajmowali się odczytem abstraktów z bazy danych i umieszczaniem ich w przestrzeni krotek. Procesy te były również odpowiedzialne za sterowanie zapelnieniem przestrzeni krotek. Tworzone krotki zawierały w kolejnych polach identyfikator dokumentu i elementy jego opisu.

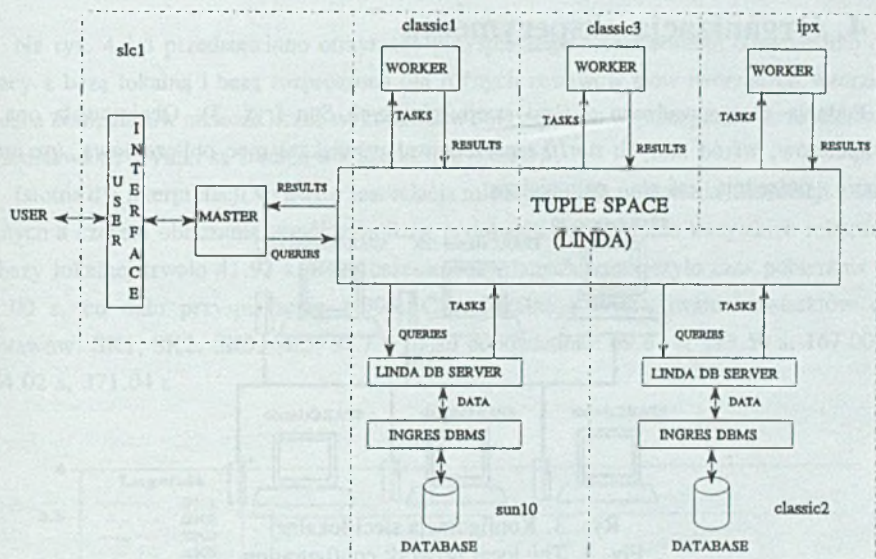


Rys. 1. Struktura systemu wyszukiwania podczas pracy z bazą lokalną

Fig. 1. The structure of the retrieval system while working with a local database

Wykonawca drugiego typu obliczał współczynnik podobieństwa pomiędzy abstraktem a wzorcem. Jego działanie rozpoczynało się odczytaniem wzorca z przestrzeni krotek. Następnie wykonawca powtarzał ciąg operacji: pobranie zadania do wykonania, obliczenie współczynnika podobieństwa oraz wyprowadzenie krotki z wynikiem, zawierającej identyfikator dokumentu i odpowiadający mu współczynnik podobieństwa. Działanie wykonawcy kończyło się, gdy pobrane zadanie zawierało znacznik końca pracy.

Na rys. 1 przedstawiono schemat struktury systemu dla pracy z bazą lokalną, a na rys. 2 dla pracy z bazą rozproszoną. Podczas pracy z bazą lokalną proces nadzorcy spełniał również funkcję procesu pobierającego opisy dokumentów z bazy danych i umieszczającego je w przestrzeni krotek. Linia przerywaną rozdzielono obszary zadań poszczególnych komputerów.



Rys. 2. Struktura systemu wyszukiwania podczas pracy z bazą rozproszoną

Fig. 2. The structure of the retrieval system while working with a distributed database

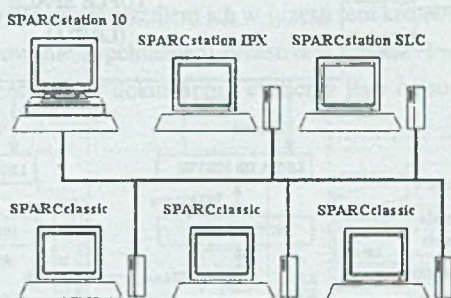
Problem ochrony przestrzeni krotek przed przepełnieniem rozwiązano stosując strategię "wysoki-niski poziom wody" [1]. Idea ta polega na utrzymywaniu liczby opisów abstraktów znajdujących się w przestrzeni krotek pomiędzy dolnym a górnym limitem. Górny limit zapewnia, że przestrzeń krotek nie zostanie przepełniona, dolny, że wykonawcy będą cały czas zajęci.

System INGRES umożliwia tworzenie rozproszonej bazy danych, na którą składa się szereg baz lokalnych znajdujących się na różnych serwerach sieci komputerowej, i która może być rozumiana jako globalna perspektywa obejmująca elementy wielu baz lokalnych. Globalny katalog systemowy zawierający informację o wszystkich tabelach, perspektywach i statystykach bazy rozproszonej zapewnia przeźroczystość lokacji (użytkownik nie potrzebuje wiedzieć, gdzie bazy składające się na bazę rozproszoną są fizycznie przechowywane). Lokalna baza może być także obsługiwana autonomicznie przez serwer lokalny. Dla celów

badawczych rozproszono bazę abstraktów tworząc dwie bazy lokalne: jedną na serwerze *sun10*, drugą na serwerze *classic2*. Zastosowano fragmentację poziomą tabel R1 i R2 (liczebności rekordów w odpowiadających sobie tabelach obu baz są do siebie zbliżone) oraz powielono tabelę R3.

4. Organizacja eksperymentu

Badania przeprowadzono w sieci stacji roboczych Sun (rys. 3). Obejmowała ona 6 komputerów, wśród których *sun10* reprezentował największą moc obliczeniową, *ipx* oraz *classic* - pośrednią, zaś *slc* - najmniejszą.



Rys. 3. Konfiguracja sieci lokalnej
Fig. 3. The local network configuration

Podczas pracy z bazą lokalną proces nadzorca został uruchomiony na komputerze *sun10*, na którym znajdował się również serwer bazy danych i proces pobierający dane z bazy danych. Pozostałe węzły obliczały współczynniki podobieństwa. Komputery były dołączane w następującej kolejności: *classic1*, *classic2*, *classic3*, *ipx*. Podczas pracy z bazą rozproszoną proces nadzorca był uruchamiany na komputerze *slc1*. Serwery bazy danych były zlokalizowane na komputerach *sun10* i *classic2*. Pozostałe węzły wykonujące porównania były dołączane w następującej kolejności: *classic1*, *classic3*, *ipx*. Na jednym komputerze był uruchamiany tylko jeden proces wykonawcy. W badaniach wykorzystano zestaw około 800 abstraktów, który był pamiętany w bazie lokalnej i rozproszonej. Baza rozproszona powstała przez podzielenie bazy lokalnej na dwie równe części. W pomiarach użyto pięciu wzorców (zestawów słów kluczowych; cyfra w nazwie wzorca oznacza liczbę słów we wzorcu):

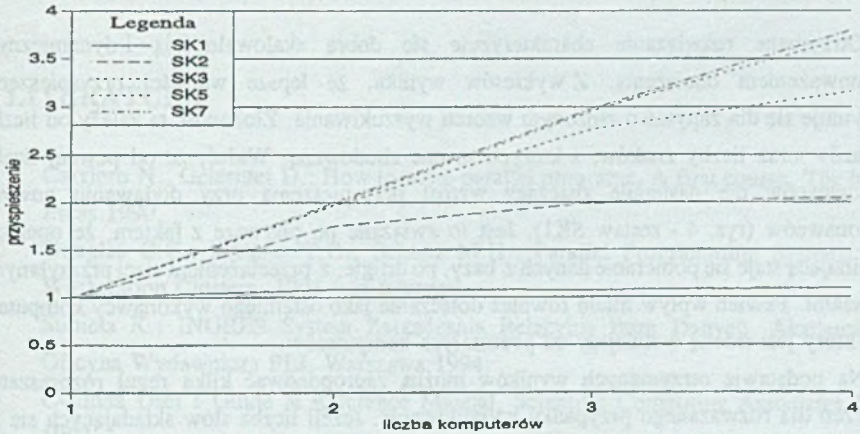
- SK1 - "Linda"
- SK2 - "Linda computing"
- SK3 - "Linda computing paralellism"

- SK5 - "Linda computing paralellism transputer synchronization"
- SK7 - "Linda computing paralellism transputer synchronization clustering database"

5. Otrzymane wyniki

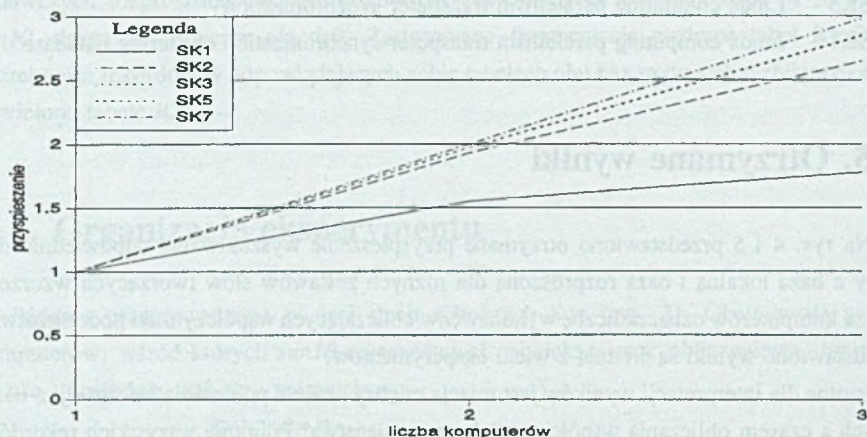
Na rys. 4 i 5 przedstawiono otrzymane przyspieszenie wyszukiwania odpowiednio dla pracy z bazą lokalną i bazą rozproszoną dla różnych zestawów słów tworzących wzorzec. Liczba komputerów oznacza liczbę wykonawców obliczających współczynniki podobieństwa. Przedstawione wyniki są średnią z wielu eksperymentów.

Istotna dla interpretacji wyników jest relacja między czasem pobierania informacji z bazy danych a czasem obliczania współczynników podobieństwa. Pobranie wszystkich rekordów z bazy lokalnej trwało 41.92 s. Rozproszenie bazy danych zmniejszyło czas pobierania do 21.00 s, co dało przyspieszenie 1.996. Czas lokalnego wyszukiwania abstraktów dla zestawów: SK1, SK2, SK3, SK5, SK7 wynosił odpowiednio: 69.82 s, 113.50 s, 167.00 s, 284.02 s, 371.04 s.



Rys. 4. Przyspieszenie wyszukiwania podczas pracy z bazą lokalną

Fig. 4. The speedup of the query execution while working with a local database



Rys. 5. Przyspieszenie wyszukiwania podczas pracy z bazą rozproszoną

Fig. 5. The speedup of the query execution while working with a distributed database

6. Podsumowanie

Otrzymane rozwiązanie charakteryzuje się dobrą skalowalnością i dynamicznym równoważeniem obciążenia. Z wykresów wynika, że lepsze wartości przyspieszenia otrzymuje się dla zapytań o złożonym wzorcu wyszukiwania. Złożoność ta zależy od liczby wyrazów oraz liczby znaków, z których są one zbudowane. Widać, że od pewnej liczby wykonawców nie następuje znaczący wzrost przyspieszenia przy dodawaniu nowych wykonawców (rys. 4 - zestaw SK1). Jest to związane po pierwsze z faktem, że operacją dominującą staje się pobieranie danych z bazy, po drugie, z przeciążeniem sieci przesyłanymi pakietami. Pewien wpływ miało również dołączanie jako ostatniego wykonawcy komputera *ipx*, który jest trochę wolniejszy od pozostałych komputerów.

Na podstawie otrzymanych wyników można zaproponować kilka reguł rozpraszania obliczeń dla rozważanego przypadku wyszukiwania. Jeżeli liczba słów składających się na wzorec wyszukiwania jest mała, opłacalne jest rozproszenie bazy danych i wykonywanie przeszukiwania lokalnie na serwerach bazy. Wiąże się to z faktem, że operacją dominującą jest pobieranie danych. Przy średniej liczbie słów we wzorcu optymalnym rozwiązaniem jest użycie części dostępnych komputerów do pobierania danych z bazy, a pozostałej części - do obliczania współczynników podobieństwa. Przy dużej liczbie słów porównywanie na tyle dominuje nad operacjami pobierania, że wpływ rozproszenia bazy danych na redukcję czasu wyszukiwania ulega zatarciu i przyspieszenie zależy od liczby komputerów obliczających

współczynniki podobieństwa. Wartości liczbowe odpowiadające użytym pojęciom: "mała liczba", "średnia liczba", "duża liczba" zależą od takich parametrów środowiska obliczeniowego jak: przepustowość sieci, szybkość komputerów dokonujących porównań, wydajność serwerów bazy danych.

Warto zwrócić uwagę na fakt, że ze wzrostem złożoności wzorca podanego w pytaniu maleje obciążenie sieci. Tak więc im bardziej złożony jest warunek wyszukiwania, tym więcej stacji można użyć do porównywania przy proporcjonalnym wzroście przyspieszenia.

Przy zadawaniu wzorca należałoby wprowadzić możliwość traktowania kilku słów jako całości w procesie wyszukiwania. Zaproponowany algorytm preferuje abstrakty zawierające dużą liczbę słów. Dlatego należałoby wprowadzić do wzoru na współczynnik podobieństwa abstraktu do wzorca wagę zależną od liczby słów. Celowe również wydaje się pomijanie tych współczynników podobieństwa między słowami, które mają wartości poniżej pewnego progu.

W pracy skoncentrowano się tylko na wybranych zagadnieniach problematyki wyszukiwania w bazach danych, tzn. przyspieszaniu poprzez ich rozpraszanie i równoległe wykonywanie. Przedstawiony system nie może pretendować do roli systemu użytkowego, został on opracowany jako system eksperymentalny. Uzyskane rezultaty mogą być natomiast z powodzeniem zastosowane w projektowaniu systemu użytkowego wykorzystującego komputer wieloprocesorowy lub sieć lokalną.

LITERATURA

- [1] Carriero N., Gelernter D.: How to write parallel programs, A first course. The MIT Press 1990.
- [2] Douglas C.C., Mattson T.G., Schultz M.H.: Parallel Programming Systems for Workstation Clusters. 1993 (via Internet).
- [3] Subieta K.: INGRES System Zarządzania Relacyjną Bazą Danych. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994.
- [4] C-Linda User's Guide & Reference Manual. Scientific Computing Associates Inc. 1993.
- [5] Wakulicz-Deja A.: Podstawy systemów wyszukiwania informacji. PWN, Warszawa-Lódź 1990.
- [6] Zieliński K.: Środowiska programowania rozproszonego w sieciach komputerowych. Księgarnia Akademicka, Kraków 1994.

Recenzent: Dr inż. Maciej Bargielski

Wpłynęło do Redakcji 5 kwietnia 1996 r.

Abstract

The paper presents an experimental database system, using parallel architecture for speeding up execution of complex data retrieval problems. Parallel algorithm implementation in a local network prove to be efficient only if time complex tasks with insignificant communication overhead are concerned. The problem of exhaustive database search shown in this paper is an example of such a task.

The database contains abstracts and basic information about scientific papers. The distributed database was created using two local Ingres database. The coordination framework of the system dealing with creation and synchronization of processes as well as with data exchange between them is written in C-Linda language. The experiments were carried out in a local network consisting of six Sun workstations.

Chapter 3 contains detail description of retrieval process. Chapters 5 and 6 present results and discussion of relations between complexity of retrieval pattern, number of workers and database distribution.

The experiments show that using parallel processing and database distribution for exhaustive database search gives a significant speedup. The more complex is the search pattern the more workstations can be used in parallel yielding proportional increase in speedup.