

Adam KOZIATEK

OCENA EFEKTYWNOŚCI KONFIGURACJI PROGRAMOWYCH DOSTĘPNYCH W ŚRODOWISKU NETWARE

Streszczenie. W pracy dokonano oceny efektywności różnych konfiguracji programowych pracujących w środowisku NetWare, których zadaniem jest manipulacja na zbiorach danych. Przeprowadzono analizę wyników eksperymentów w odniesieniu do architektury badanych konfiguracji. Wyboru środowiska badawczego, tj. platformy sprzętowej, systemu operacyjnego i oprogramowania narzędziowego dokonano pod kątem dostępności wymienionych narzędzi. Wydaje się, że uzyskane wyniki są uniwersalne i niezależne od środowiska operacyjnego. Stanowią one bowiem ocenę i porównanie pewnych modelowych architektur programowych i mogą posłużyć jako pomoc przy wyborze konkretnych rozwiązań.

ESTIMATION OF EFFECTIVENESS NETWARE-COMPATIBLE DATABASE MANAGEMENT SOFTWARE CONFIGURATIONS

Summary. The article includes estimation of effectiveness of various database management software configurations, which are NetWare-compatible. The analysis of experimental results has been carried out corresponding to the architecture of the configurations in question. The experiment environment (i.e. hardware platform, operating system and software utilities) has been chosen regarding the availability. The results may be treated as universal and environment independent. As a set of estimations and comparisons of some model software architectures they may be helpful in choosing a specific solution.

BEWERTUNG AUSGEWÄHLTER PROGRAMMKONFIGURATIONEN IN DER NETWARE-UMGEBUNG

Zusammenfassung. Im Artikel wurde die Effizienz verschiedener, in der NetWare-Umgebung laufender Programmkonfigurationen bewertet, deren Aufgabe die Manipulation der Dateien ist. Die durchgeführte Analyse der Experimentenergebnisse bezieht sich auf die Architektur der geforschten Konfigurationen. Die Forschungsumgebung (d.h. Hardware-Plattform, Betriebssystem und Software) wurde im Hinblick auf die Zugänglichkeit o.g. Werkzeuge ausgewählt. Die erreichten Ergebnisse scheinen universell und vom Betriebssystem unabhängig zu sein. Sie bilden nämlich die Bewertung und den Vergleich gewisser Musterprogrammarchitekturen und können als Hinweis bei der Auswahl konkreter Lösungen dienen.

1. Metodologia pomiarów

Jako podstawę do określenia efektywności systemów przyjęto czas wykonania podstawowych operacji na zbiorach danych. Badania przeprowadzono w czterech konfiguracjach programowych:

- środowisko lokalne,
- środowisko sieciowe z lokalnym zarządcą zbiorów danych,
- środowisko sieciowe z przekazywaniem zleceń,
- środowisko sieciowe z wykorzystaniem zdalnych procedur.

Wszystkie te konfiguracje są opisane w następujących punktach.

Pomiary wykonano w różnych konfiguracjach możliwych do zaimplementowania przy spełnieniu następujących warunków:

- jednakowy algorytm pomiaru czasu,
- jednakowy efekt końcowy wykonywanych operacji.

Do pomiaru czasu wykorzystano klasę użytkową Timer, dostarczoną wraz z kompilatorem języka C++ firmy Borland w wersji 3.0. Wydaje się, że tak proste narzędzie nie powinno wносить istotniejszych błędów. Pomiaru czasu dokonywano bezpośrednio przed odwołaniem do pakietu użytkowego (Btrieve; program, uruchomiony na maszynie serwera plików itd.) i bezpośrednio po jego zakończeniu. Dzięki temu uzyskano czas samego odwołania do pakietu użytkowego bez narzutu na obsługę zleceń. Pewną trudność sprawiał pomiar wydajności zdalnych wywołań. Przyjęto tutaj założenie, że celem pomiarów jest ocena metody działania. Zatem jednakowy powinien być efekt wykonanych operacji, zaś sposób ich realizacji może być dowolny. Ważne było, by stan danych był po wykonaniu każdej z badanych operacji taki sam, niezależnie od metody. W zmierzony czas trwania operacji wliczano czas bezpośredniego odwołania się do funkcjonujących narzędzi - pominięto natomiast czas przygotowania danych

potrzebnych do wykonania badanej operacji. Miało to na celu wyeliminowanie tych okresów, w których dane podlegały pewnej obróbce, związanej z samą organizacją badań. Celem badań była również ocena jakości usług świadczonych przez konkretne narzędzia programistyczne.

Inną sprawą był wybór operacji do pomiarów. Subiektywnie zdecydowano, że badane będą operacje najczęściej wykonywane. Nie znaleziono bowiem żadnego kryterium, które określałoby częstotliwość wykonywania danej operacji.

Testowane operacje można podzielić na dwie grupy:

- proste,
- złożone.

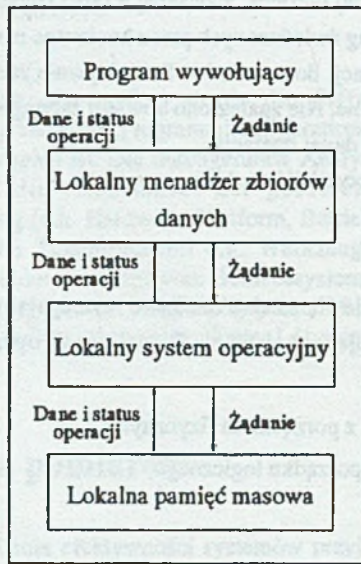
Jako operacje proste rozumie się tutaj takie, które wymagają pojedynczego odwołania do pakietu użytkowego, manipulującego danymi. Do tej grupy operacji z pewnością możemy zaliczyć:

- pobieranie rekordów zgodnie z porządkiem fizycznym,
- pobieranie rekordów według porządku logicznego,
- wprowadzanie rekordów,
- usuwanie rekordów,
- modyfikację istniejących rekordów z podziałem na modyfikację pól, będących składowymi kluczy oraz nie będących składowymi żadnego klucza.

Natomiast jako operacje złożone rozumie się takie, które wymagają wielokrotnego odwołania się do danych w ramach tej samej operacji. Zatem operacjami złożonymi jest zestaw operacji prostych. W badaniach aktualizowano jedno z pól rekordu na podstawie wyników obliczeń, przeprowadzonych z udziałem innych pól tego samego rekordu. Szczegóły zamieszczono w następujących rozdziałach.

2. Warunki techniczne

Badania przeprowadzono na komputerach klasy IBM PC z procesorem firmy Intel 80386 z zegarem 33 MHz. Komputery były wyposażone w twarde dyski magnetyczne. Szczegółowe parametry komputerów, na których przeprowadzano badania, zostały podane w tabeli 1. Pomiarów parametrów technicznych sprzętu komputerowego dokonano z wykorzystaniem programu Power Meter firmy Database Group Inc. wersja 1.5.



Stacja robocza

Rys. 1. Środowisko lokalne

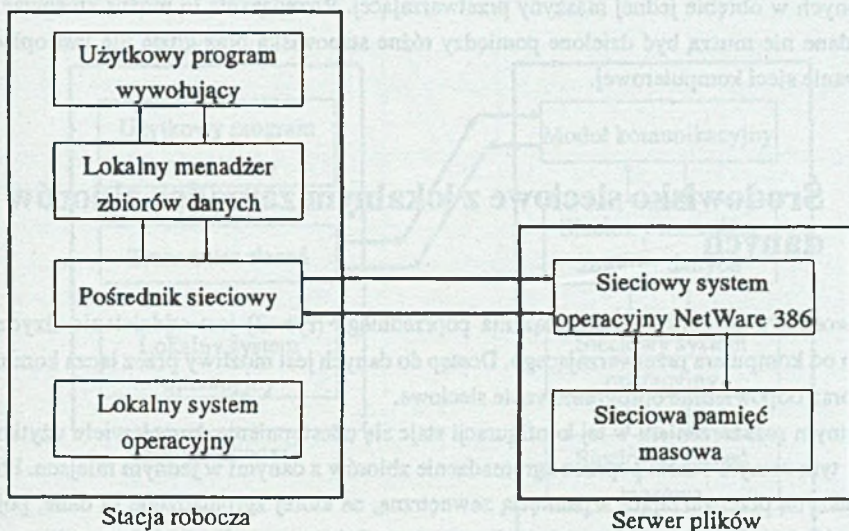
Fig. 1. Local environment

Tabela 1

Parametry środowiska sprzętowego

Parametr	Jednostka miary	Stacja robocza	Serwer plików
Takt zegara	MHz	33	33.08
Moc obliczeniowa procesora	MIPS	7.99	7.94
Pojemność dysku	MB	104.5	327.7
Średni czas dostępu do dysku	ms	23.97	18.09
Transmisja danych	kB/s	1068.98	1914.27

Szczególną uwagę warto zwrócić na średni czas dostępu do dysku oraz czas transmisji danych z dysku do pamięci. Efektywność dysku w stacji roboczej jest mniejsza od dysku zainstalowanego w serwerze plików. Ma to znaczenie przy ocenie wyników konfiguracji lokalnej, w której kluczowe są odwołania do dysku lokalnego.



Rys. 2. Środowisko sieciowe

Fig. 2. Network environment

3. Środowisko lokalne

Cechą tej konfiguracji (rys. 1) jest to, że komputer przetwarzający dane ma bezpośredni dostęp do pamięci masowych, na których dane są zgromadzone. Brak jest zewnętrznych łączy komunikacyjnych pomiędzy procesorem a tą pamięcią. Drugą cechą wyróżniającą jest to, że dostęp ten jest wyłączny, tzn. tylko ten komputer ma dostęp do tych danych i tylko on może je przetwarzać. Procesor korzysta z tych danych w następujący sposób:

- generowane jest zlecenie do lokalnego menadżera zbiorów danych,
- menadżer realizuje zlecenie otrzymane od programu użytkownika; wykorzystuje on przy tym funkcje oferowane przez lokalny system operacyjny,
- sterowanie wraz z wynikami jest przekazywane do programu użytkownika.

Zgodnie z powyższym wśród znaczących składowych czasowych można wyróżnić:

- czas dostępu do danych w pamięci zewnętrznej (czas odczytu lub zapisu danych w takiej pamięci),
- czas przetwarzania danych.

Ta ostatnia składowa powinna być pominięta, skoro przyjęto zasadę eliminacji okresów organizacyjnych potrzebnych do przeprowadzenia badań testowych.

Konfiguracja ta wydaje się jednak mało praktyczna ze względu na lokalność danych zgromadzonych w obrębie jednej maszyny przetwarzającej. Rozwiązanie to można stosować tam, gdzie dane nie muszą być dzielone pomiędzy różne stanowiska oraz gdzie nie jest opłacalne stosowanie sieci komputerowej.

4. Środowisko sieciowe z lokalnym zarządcą zbiorów danych

Nowością w stosunku do rozwiązania poprzedniego (rys. 2) jest oddzielenie fizycznych danych od komputera przetwarzającego. Dostęp do danych jest możliwy przez łącza komunikacyjne oraz odpowiednie oprogramowanie sieciowe.

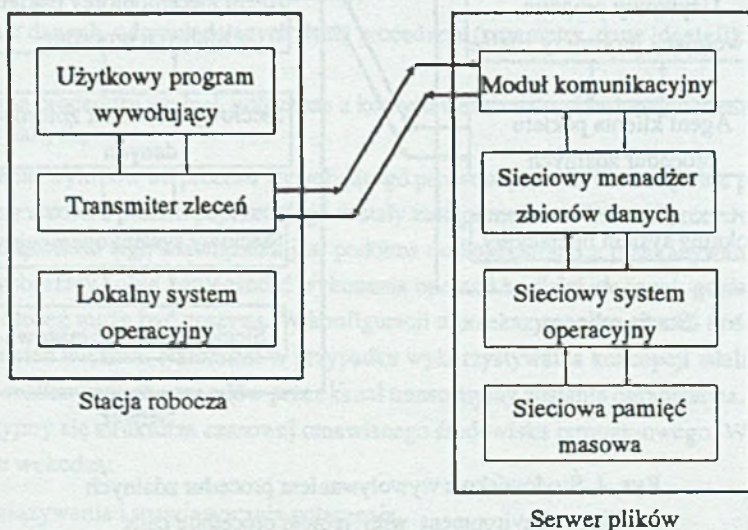
Istotnym rozszerzeniem w tej konfiguracji staje się udostępnienie danych wielu użytkownikom w tym samym czasie poprzez zgromadzenie zbiorów z danymi w jednym miejscu. Pomiedzy maszyną przetwarzającą a pamięcią zewnętrzną, na której zgromadzone są dane, pojawia się kanał komunikacyjny, którym przesyłane są dane. Pojedyncze odwołanie jest realizowane w następujący sposób:

- przekazanie zlecenia do lokalnego menadżera zbiorów z poziomu programu użytkowego (tu: program testowy),
- wykonanie zadania - lokalny menadżer zbiorów wywołuje kilkakrotnie funkcje systemu operacyjnego; każde takie wywołanie jest kontrolowane przez pośrednika sieciowego (*ang. network shell*) i kierowane do sieciowego systemu operacyjnego. Każde takie odwołanie do sieciowego systemu operacyjnego musi być transmitowane przez kanał komunikacyjny, przetworzone na serwerze zbiorów, odpowiedź znowu musi zostać przesłana przez sieć,
- wyniki wykonania operacji zostają przekazane do programu użytkowego.

Analizując tę konfigurację pod względem czasowym, należy wziąć pod uwagę powtarzające się odwołania do sieciowego systemu operacyjnego. Każde z nich wiąże się przecież z przesyłaniem ządania, jego przetworzeniem i przesyłem zwróconym odpowiedzi. Inną cechą jest to, że ządania od wielu użytkowników mogą wzajemnie ze sobą konkurować i blokować chwilowo dostęp do danych. Uwzględniając powyższe można wyróżnić następujące składowe czasowe:

- czas transmisji danych pomiędzy serwerem a stacją roboczą,
- czas przetwarzania zleceń przez serwer plików,
- czas zużywany przez procesy, pracujące na stacji roboczej (program testowy, lokalny menadżer zbiorów danych, pośrednik sieciowy).

Szczególnią wagę ma tutaj czas transmisji danych przez kanał komunikacyjny, tym bardziej, że w przypadku intensywnego wielodostępu zwiększa się obciążenie podsieci transmisyjnej.



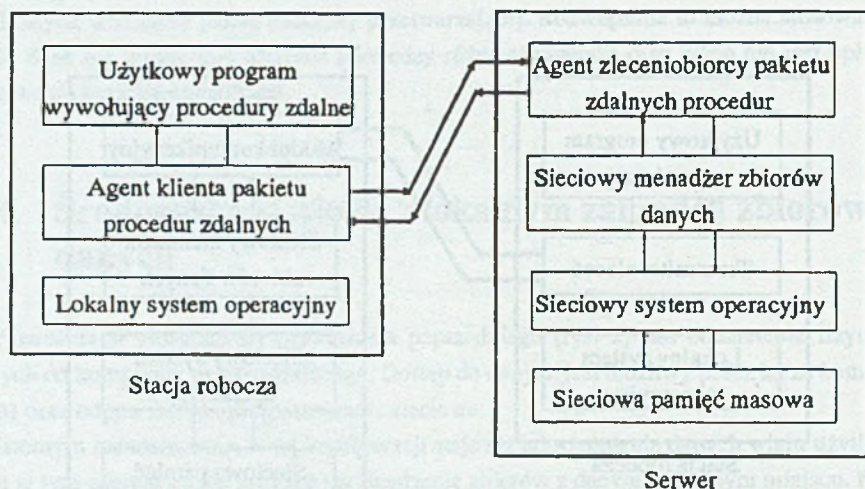
Rys. 3. Środowisko z przekazywaniem zleceń

Fig. 3. Environment with transmitted orders

5. Środowisko sieciowe z przekazywaniem zleceń

Ta konfiguracja podobna jest do swojej poprzedniczki (rys. 3). Podobnie jak w konfiguracji poprzedniej, oddzielono tu miejsce składowania danych od miejsca ich przetwarzania oraz możliwy jest dostęp do danych przez wielu użytkowników. Tym razem jednak zarządca zbiorów danych nie rezyduje już na stacji roboczej a na serwerze plików. Na stacji uruchomiony jest jedynie program, zwany transmitterem zleceń, który otrzymawszy zlecenie od programu użytkowego przekazuje je do sieciowego menadżera zbiorów danych. Zatem w odróżnieniu od konfiguracji poprzedniej transmisje nie są już tak częste, gdyż odwołanie do sieciowego systemu operacyjnego jest realizowane tylko jeden raz - wraz z transmisją zlecenia. Szczegółowo wygląda to następująco:

- generacja zlecenia przez program użytkowy,
- przyjęcie żądania przez transmitter zleceń i przekazanie go do modułu komunikacyjnego na serwerze,
- przetworzenie zlecenia przez sieciowego menadżera zbiorów z wykorzystaniem funkcji sieciowego systemu operacyjnego,



Rys. 4. Środowisko z wywoływaniem procedur zdalnych

Fig. 4. Environment with remote procedure calls

- przekazanie wyników wykonanej operacji poprzez moduł komunikacyjny i transponder zleceń do programu użytkowego.

W omawianym środowisku całe zlecenie jest przesyłane przez kanał komunikacyjny i dopiero na serwerze jest ono przetwarzane - w ten sposób ilość transmisji została ograniczona do minimum.

Struktura czasowa każdego takiego dostępu do danych wygląda następująco:

- czas przesyłu zlecenia od transpondera zleceń (stacja robocza) do modułu komunikacyjnego (serwer plików),
- czas obsługi zlecenia przez sieciowego menadżera plików.

Ten ostatni czas może być powiększony o narzut związany z wielodostępem.

6. Środowisko sieciowe z wywoływaniem zdalnych procedur

Koncepcja procedur zdalnych opiera się na następującej zasadzie: program na jednej maszynie cyfrowej wywołuje procedurę, która będzie wykonywana na innym komputerze (rys. 4). Wywołanie takiej procedury dla programu inicjującego jej działanie praktycznie nie różni się od wywołania procedury lokalnej. Pojedyncze odwołanie do danych można opisać w kolejnych krokach:

- wywołanie procedury zdalnej przez program użytkowy,
- nawiązanie połączenia pomiędzy pośrednikiem procedur zdalnych, pracującym na maszynie klienta a jego odpowiednikiem na serwerze,
- przesłanie danych, odpowiadających danej procedurze (parametry, dane identyfikacyjne itp.),
- wykonanie procedury zdalnej, połączone z korzystaniem z usług sieciowego menadżera zbiorów danych,
- przekazanie wyników do procesu wywołującego procedurę zdalną i rozwiązanie połączenia.

Przekazy zleceń z punktu poprzedniego zostały zastąpione wywołaniami procedur zdalnych. Zatem efektywność tego rozwiązania jest podobna do konfiguracji z przekazywaniem zleceń. Jednak wyobraźmy sobie konieczność wykonania operacji bardziej złożonej, gdzie ilość pojedynczych zleceń może być znaczna. W konfiguracji z przekazywaniem zleceń ilość transmisji będzie również większa. Natomiast w przypadku wykorzystywania koncepcji zdalnych procedur liczba realizowanych przesyłów przez kanał transmisyjny zostanie ograniczona.

Przyjrzyjmy się strukturze czasowej omawianego środowiska programowego. W ramy łącznego czasu wchodzi:

- czas nawiązywania i rozwiązywania połączenia,
- czas transmisji danych przez podsieć komunikacyjną tam i z powrotem,
- czas wykonywania procedury zdalnej na serwerze.

Pierwsza z wymienionych składowych czasowych będzie prawdopodobnie znacznie obniżać efektywność w przypadku operacji prostych. Zaś to, że transmisje danych będą wykonywane jedynie dwukrotnie dla każdego wywołania sprawi, iż będzie to rozwiązanie bardziej efektywne w odniesieniu do operacji złożonych.

Zwróćmy jeszcze uwagę na fakt, gdzie realizowane są obliczenia. W każdej z dotychczas omawianych konfiguracji przetwarzanie danych odbywało się na stacji roboczej - w konfiguracji z wywoływaniem procedur wszelkie obliczenia odbywają się na maszynie serwera sieciowego. Konsekwencją tego faktu jest to, że w poprzednich konfiguracjach wymogi techniczne stawiane stacji roboczej były znaczne (w zależności od zadania). Stacja robocza musiała obsługiwać zarówno komunikację z użytkownikiem, jak i wykonywać obliczenia. Natomiast w środowisku z wywoływaniem procedur zdalnych wymogi stawiane stacjom roboczym są mniejsze. Odwrotnie ma się ta sprawa, jeśli chodzi o serwer sieciowy. W środowisku procedur zdalnych jego większa moc obliczeniowa może być znacznie efektywniej wykorzystana niż w poprzednich rozwiązaniach.

7. Dostęp jedno- i wieloużytkowy

Badania przeprowadzono w dwóch trybach:

- z pojedynczym dostępem do danych,
- z jednoczesnymi odwołaniami do danych.

W wariancie z pojedynczym dostępem do danych zadbano o to, by do danych odwoływał się jeden i tylko jeden proces użytkowy. Chodziło w tym trybie głównie o pomiar czasu trwania operacji jednostkowych w celu oceny efektywności stosowanych narzędzi. Inny był cel przeprowadzenia badań w trybie wieloużytkowym. Tu głównym zadaniem była ocena efektywności danej konfiguracji w zależności od zwiększającego się obciążenia.

8. Uwagi implementacyjne

Omawiając poszczególne konfiguracje celowo nie podawano konkretnych nazw stosowanych narzędzi. W różnych systemach mogą być stosowane różne narzędzia, natomiast wyróżnione tu środowiska mogą być traktowane nieco ogólniej. Obecnie opisane zostaną te, które były wykorzystywane w badaniach. W różnych konfiguracjach pewne elementy oprogramowania były podobne. I tak:

- Na stacji roboczej działał system operacyjny DOS firmy Microsoft w wersji 5.0.
- Jako sieciowy system operacyjny zastosowany został system NetWare 386 w wersji 3.11.
- W kanale transmisyjnym funkcjonował protokół komunikacyjny IPX/SPX, opracowany przez firmę Xerox.

8.1. Środowisko lokalne

Menadżerem zbiorów danych był pakiet BTRIEVE 5.10, zaimplementowany przez firmę WATCOM.

8.2. Środowisko sieciowe z lokalnym menadżerem zbiorów

Rolę lokalnego menadżera zbiorów spełniał program BTRIEVE w wersji 5.10, zaimplementowany przez firmę WATCOM. Pośrednikiem sieciowym był program NETX, wersja 3.22 - produkt firmy Novell.

8.3. Środowisko sieciowe z przesyłem zleceń

Jako transmitter zleceń wykorzystano program BREQUEST w wersji 5.16 firmy Novell. Funkcje modułu komunikacyjnego spełniał moduł ładowny BSPXCOM.NLM. Przekazywał on zlecenia do sieciowego menadżera zbiorów danych, którym był moduł ładowny BTRIEVE.NLM.

8.4. Środowisko sieciowe z wywoływaniem procedur zdalnych

Wykorzystywany był pakiet procedur zdalnych wraz z biblioteką sieciową firmy Novell w dwóch wariantach:

- dla systemu DOS w wersji 1.1,
- dla systemu NetWare 386.

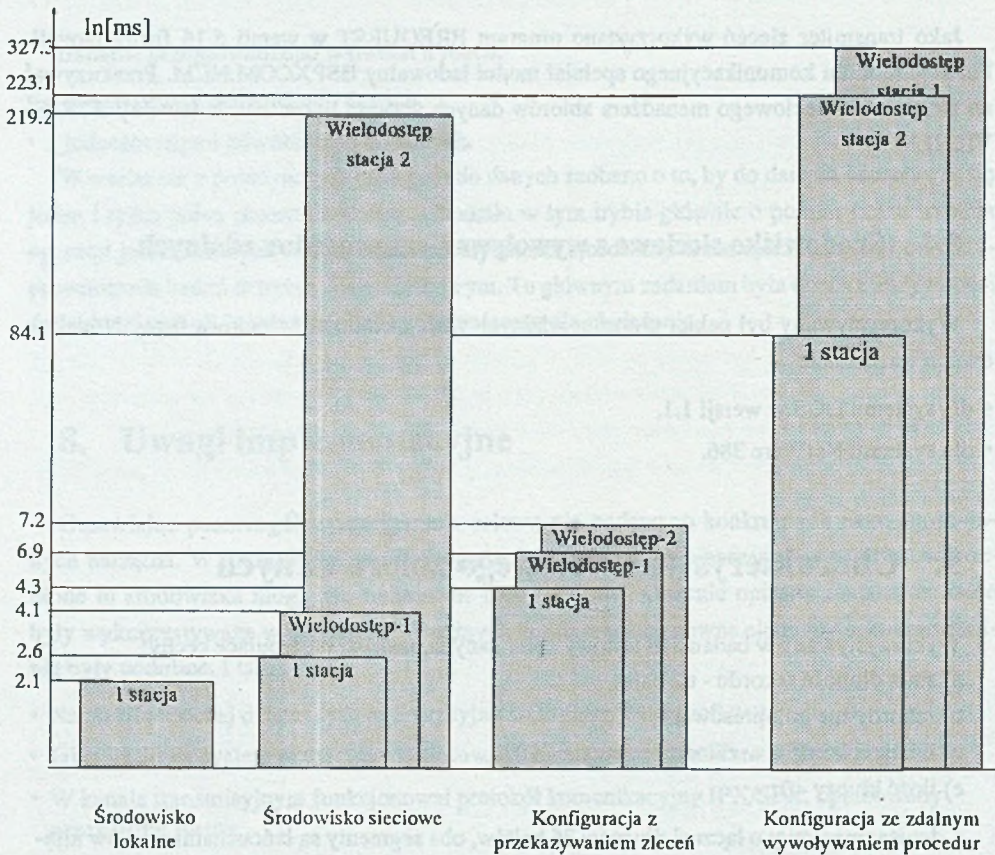
9. Charakterystyka testowego zbioru danych

Wykorzystywany w badaniach testowy zbiór danych posiadał następujące cechy:

- a) stała długość rekordu - 62 bajty,
- b) rekordy nie kompresowane,
- c) liczba rekordów w zbiorze testowym - 1000,
- e) ilość kluczy - 3:
 - dwusegmentowy o łącznej długości 36 bajtów, oba segmenty są łańcuchami znaków alfanumerycznych,
 - jednosegmentowy o długości 4 bajtów, typ - numeryczny, rzeczywisty,
 - jednosegmentowy o długości 4 bajtów, typ - numeryczny, autoinkrementowy.

10. Wyniki badań

Pomiary dokonywane były w pięciu niezależnych seriach. Na podstawie czasów trwania operacji w każdej serii obliczano czas średni, będący średnią arytmetyczną czasów cząstkowych. Na osi rzędnych zaznaczono zlogarytmizowane (logarytm naturalny) średnie arytmetyczne z każdej serii. W ramach każdej konfiguracji przedstawiono wyniki w trybie dostępu wyłącznego do zbioru oraz wyniki uzyskane na każdej ze stacji w trybie wielodostępu.



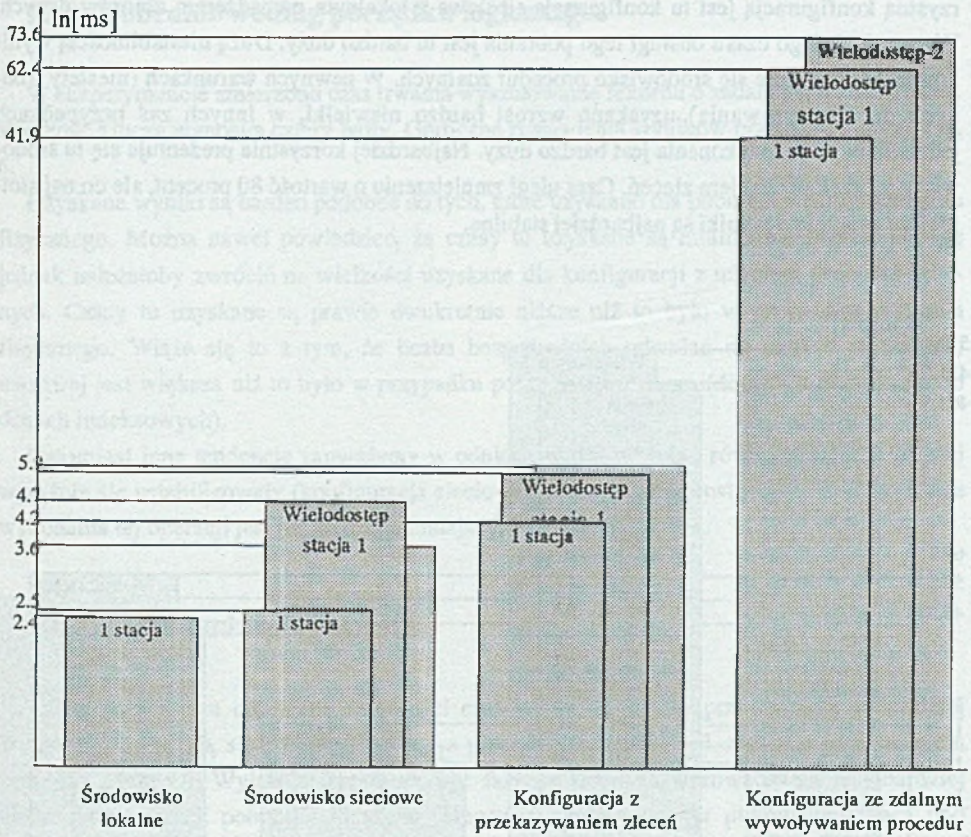
Rys. 5. Pobranie według porządku fizycznego

Fig. 5. Retrieve record by physical order

10.1. Pobranie według porządku fizycznego

Fizyczne położenie rekordu danych w zbiorze zależy od czasu, w którym rekord był do tego zbioru wprowadzany. Jednocześnie dzięki istnieniu koncepcji indeksowania rekordy mogą być dostępne zgodnie z obowiązującą regułą logicznego uporządkowania. Stąd rozróżnienie dwóch typów pobrania:

- pobranie według porządku fizycznego, w którym rekordy pobierane są zgodnie z ich umiejscowieniem w zbiorze danych,
- pobranie według porządku logicznego - tu rekordy są pobierane według określonego indeksu, który określa pewien porządek logiczny.



Rys. 6. Pobranie według porządku logicznego

Fig. 6. Retrieve record by logical order

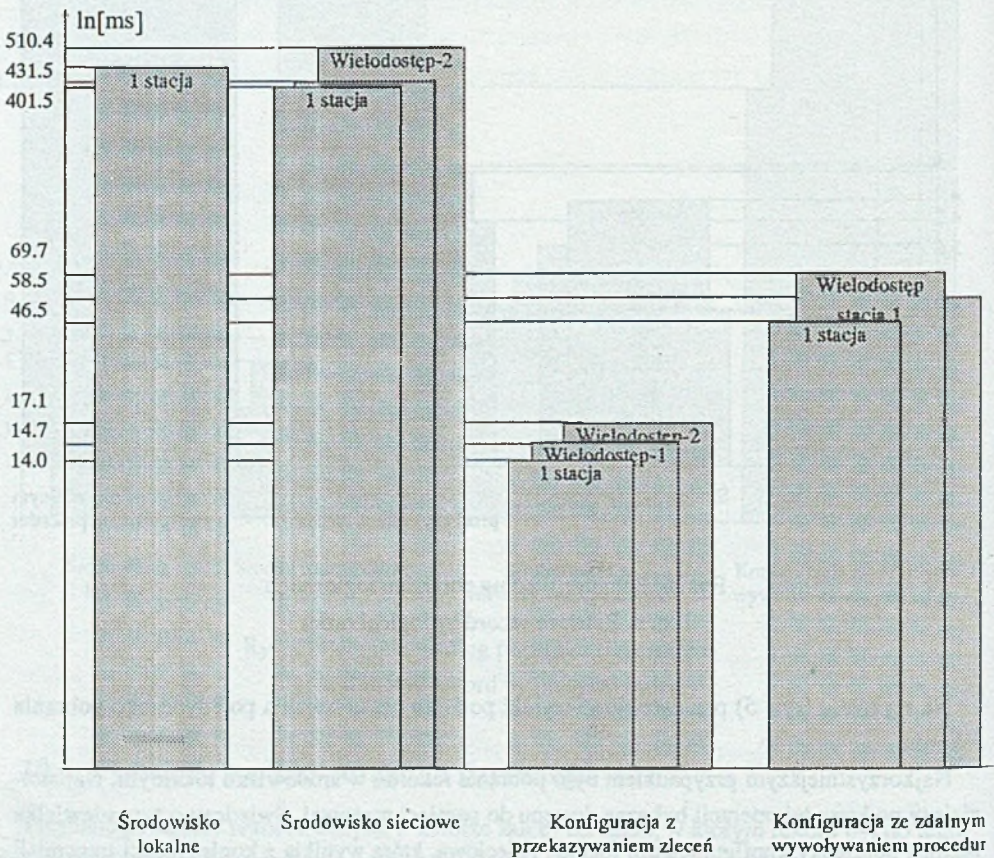
Na wykresie (rys. 5) przedstawiono wyniki pomiaru czasu trwania pojedynczego pobrania rekordu.

Najkorzystniejszym przypadkiem było pobranie rekordu w środowisku lokalnym. Najistotniejszy podczas tej operacji był czas dostępu do pamięci masowej. Świadczy o tym niewielka różnica pomiędzy konfiguracjami: lokalną i sieciową, która wynikała z konieczności transmisji danych przez kanał komunikacyjny w konfiguracji sieciowej (rys. 5).

Znacznie gorszy wynik uzyskany w konfiguracji z procedurami zdalnymi jest konsekwencją faktu, że przy każdym wywołaniu procedury zdalnej następuje nawiązanie połączenia pomiędzy programem klienta (tu: program testowy) a programem na serwerze.

Zwróćmy teraz uwagę na czasy uzyskane dla dwóch procesów użytkowych, chcących wykonać operację pobrania w jednej chwili czasu. Średni czas jednostkowy wykonania operacji jest w tym trybie większy w porównaniu z wariantem jednon użytkowym. Najbardziej nieko-

rzystną konfiguracją jest tu konfiguracja sieciowa z lokalnym menadżerem zbiorów danych. Wzrost średniego czasu obsługi tego pobrania jest tu bardzo duży. Dużą niestabilnością wyników charakteryzuje się środowisko procedur zdalnych. W pewnych warunkach (niestety trudnych do sprecyzowania) uzyskano wzrost bardzo niewielki, w innych zaś przypadkach jednostkowy czas wykonania jest bardzo duży. Najbardziej korzystnie prezentuje się tu środowisko z przekazywaniem zleceń. Czas uległ zmniejszeniu o wartość 80 procent, ale co najistotniejsze osiągnięte wyniki są najbardziej stabilne.



Rys. 7. Wprowadzanie rekordu
Fig. 7. Inserting records to database

10.2. Pobranie według porządku logicznego

W eksperymencie zmierzono czas trwania wyszukiwania rekordu o zadanej wartości klucza. Wielkość klucza wynosiła cztery bajty. Graficzne zestawienie wyników przedstawiono na rys. 6.

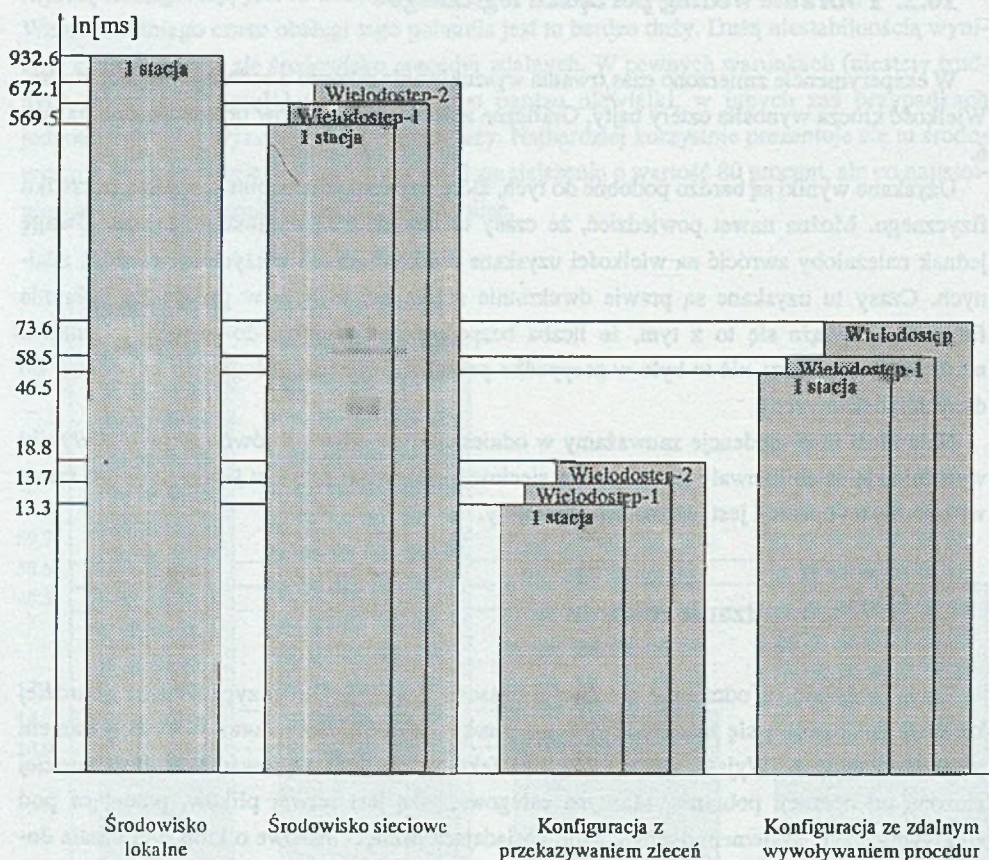
Uzyskane wyniki są bardzo podobne do tych, które uzyskano dla pobrania według porządku fizycznego. Można nawet powiedzieć, że czasy tu uzyskane są minimalnie niższe. Uwagę jednak należałoby zwrócić na wielkości uzyskane dla konfiguracji z użyciem procedur zdalnych. Czasy tu uzyskane są prawie dwukrotnie niższe niż to było w przypadku pobrania fizycznego. Wiąże się to z tym, że liczba bezpośrednich odwołań do danych w pamięci masowej jest większa niż to było w przypadku pobrania fizycznego (dochodzą odwołania do danych indeksowych).

Natomiast inne tendencje zauważamy w odniesieniu do odwołań równoczesnych. Wyniki wyraźnie się ustabilizowały (konfiguracja sieciowa), natomiast przyrost jednostkowego czasu wykonania tej operacji jest już znacznie mniejszy.

10.3. Wprowadzanie rekordu

Zarysowały się tu odmienne zależności czasowe (rys. 7). Oto przypadkami najbardziej korzystnymi okazują się być środowiska: z przekazywaniem zleceń oraz z wywoływaniem procedur zdalnych. Wyjaśnieniem może być fakt, że operacja wprowadzania jest bardziej złożona od operacji pobrania. Maszyna usługowa, jaką jest serwer plików, pracująca pod efektywniejszym systemem operacyjnym, posiadająca pamięci masowe o krótszym czasie dostępu, wykonuje szybciej wszelkie operacje. Operacja bardziej złożona to więcej odwołań do pamięci zewnętrznych, to więcej obliczeń - wszystko to zwiększa łączny czas wykonania. Zauważmy, że wynik najbardziej niekorzystny osiągnięto tu w środowisku lokalnym. Podstawową tego przyczyną jest pamięć zewnętrzna o słabych parametrach technicznych (główną rolę gra tutaj czas dostępu do dysku). Trudno byłoby jednak nie zauważyć, że narzut czasowy wnoszony przez sieciowe elementy przedstawianych konfiguracji jest coraz mniej znaczący, górę zaś biorą takie elementy, jak czas dostępu do pamięci zewnętrznych czy moc obliczeniowa jednostek wykonujących dane operacje.

Przeanalizujmy jeszcze wydajność systemów dla operacji równoczesnych. Efektywność dla dwóch stacji pogorszyła się bardzo nieznacznie w porównaniu z pracą z jednym procesem użytkowym. Daje się zauważyć tendencję mniejszego wzrostu łącznego czasu wykonania operacji bardziej złożonej.

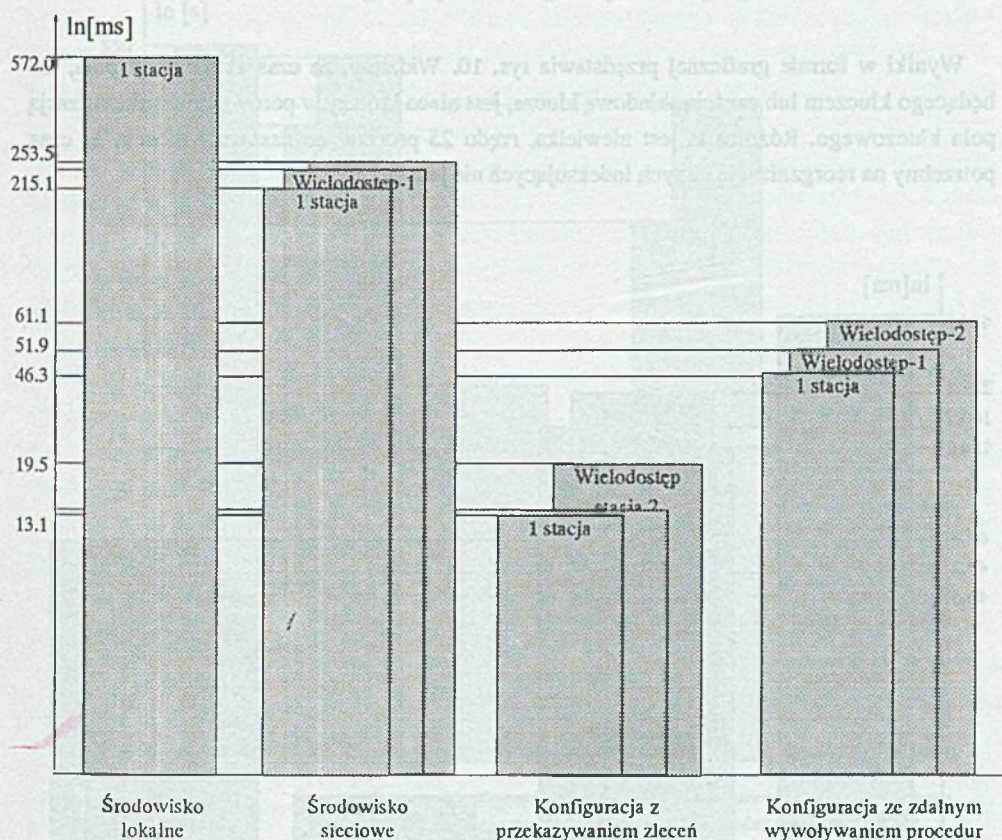


Rys. 8. Usuwanie rekordu

Fig. 8. Removing records from database

10.4. Usuwanie rekordów

Operację usuwania rekordów również można zaliczyć do tych nieco bardziej złożonych. Otrzymane tutaj rezultaty (rys. 8) potwierdzają wysnute wcześniej wnioski. Należy jedynie zwrócić uwagę na uzyskane w niektórych seriach błędy. Otóż w trakcie próby dostępu do danego rekordu inny proces wykonywał już operację usuwania na tym rekordzie. Powstał zatem konflikt w dostępie. Rodzi to pewną uwagę, że w konfiguracjach z przekazywaniem zleceń oraz zdalnych procedur mogą występować konflikty w dostępie do rekordów. Wprowadza to konieczność zabezpieczania programów użytkowych przed taką ewentualnością, co prawdopodobnie pociągnie za sobą zwiększenie czasu obsługi i obniżenie efektywności tych metod obsługi zleceń.



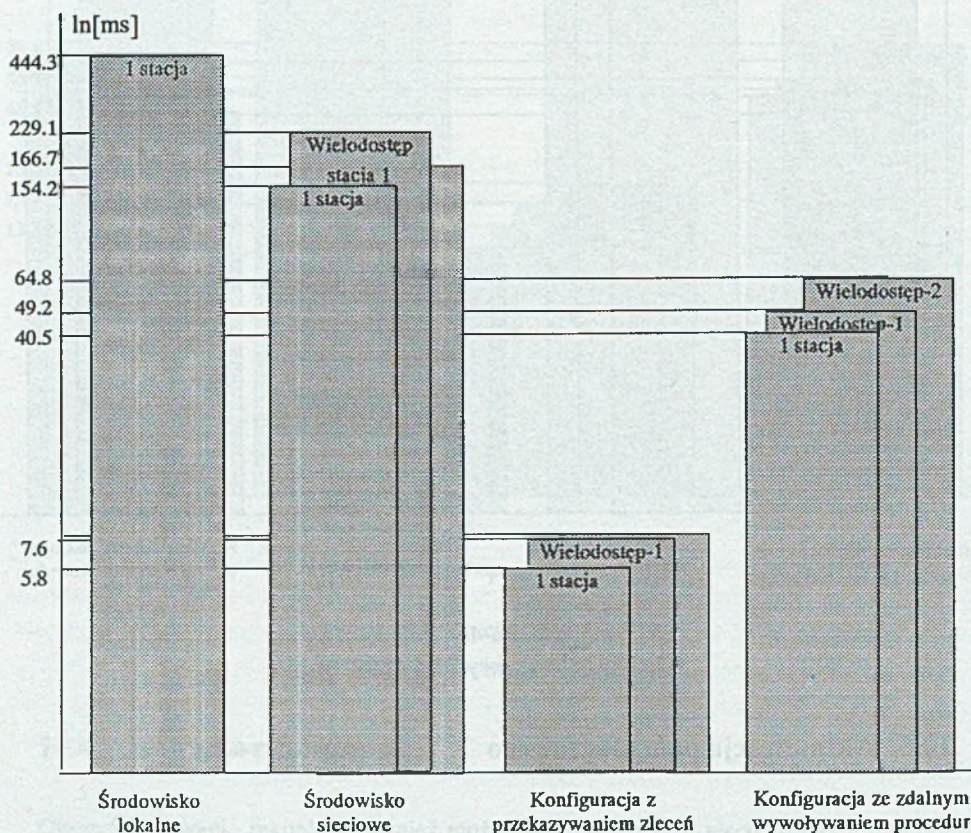
Rys. 9. Aktualizacja pola kluczowego
 Fig. 9. Update a key field of record

10.5. Aktualizacja pola kluczowego

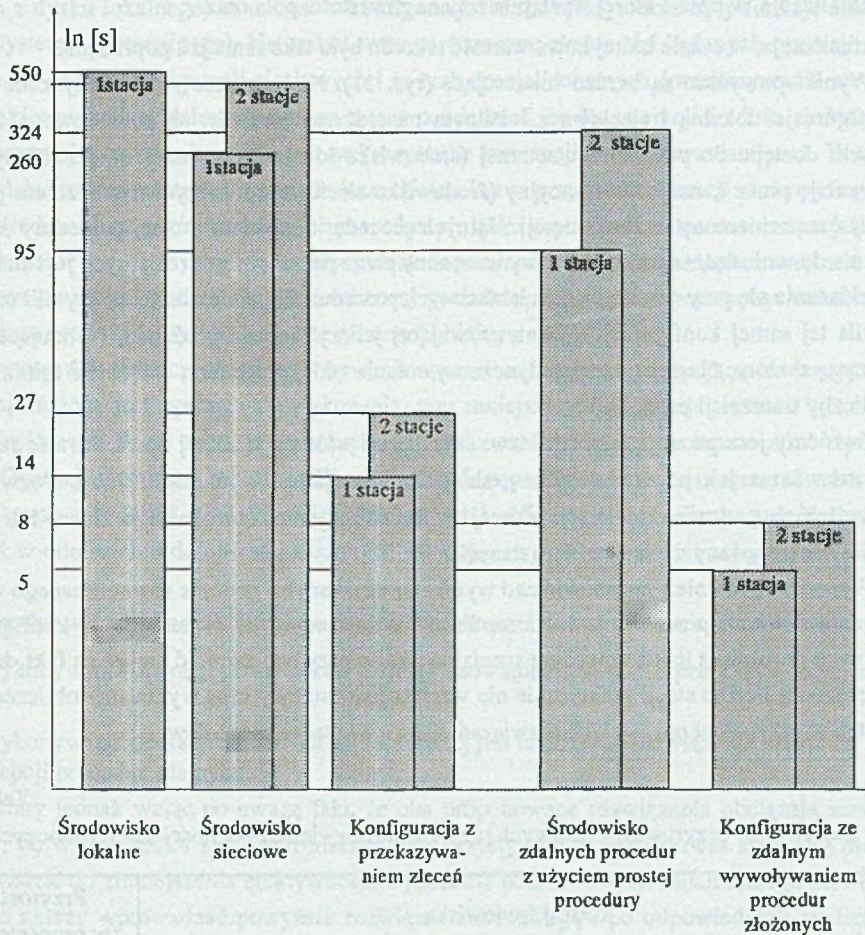
Zmiana zawartości pola, będącego kluczem bądź segmentem klucza, wymaga zmiany nie tylko danych rekordu w zbiorze - konieczna jest także aktualizacja danych indeksujących. Będzie to powodowało częstsze niż przy aktualizacji zwykłego pola odwołania do pamięci zewnętrznej (rys. 9). Otrzymane tu wyniki potwierdzają wcześniejsze wnioski.

10.6. Aktualizacja pola zwykłego (nie będącego kluczem)

Wyniki w formie graficznej przedstawia rys. 10. Widzimy, że czas aktualizacji pola, nie będącego kluczem lub częścią składową klucza, jest nieco krótszy w porównaniu z aktualizacją pola kluczowego. Różnica ta jest niewielka, rzędu 25 procent, co nasuwa wniosek, że czas potrzebny na reorganizację danych indeksujących nie jest duży.



Rys. 10. Aktualizacja pola zwykłego
Fig. 10. Update a normal field of record



Rys. 11. Operacja złożona
Fig. 11. Complex operation

10.7. Operacja złożona

Operacją złożoną w tym eksperymencie było pobranie rekordu, wykonanie pewnych nieskomplikowanych operacji arytmetycznych na polach rekordu oraz zapis wyników do zbioru danych. Zapis nie był wprowadzeniem nowego rekordu, a jedynie aktualizacją rekordu wcześniej pobranego. Aktualizowane było pole kluczowe. Powyższa operacja była wykonywana dla każdego rekordu w zbiorze. Liczba rekordów w każdej próbie była jednakowa. W pomiarach w wariancie jednonużytkowym dokonywano dwóch rodzajów aktualizacji:

- aktualizacja, w czasie której wystąpiła zmiana zawartości pola oraz
- aktualizacja, w czasie której nowa wartość rekordu była taka sama jak poprzednia.

Wyniki powyższe są bardzo interesujące (rys. 11). Niekorzystne wyniki uzyskane przez konfiguracje: lokalną i sieciową z lokalnym menadżerem zbiorów są powodowane dużym czasem dostępu do pamięci zewnętrznej (środowisko lokalne) oraz czasem, potrzebnym na transmisję przez kanał komunikacyjny (środowisko sieciowe z lokalnym menadżerem pliku). Duży czas zmierzony w konfiguracji zdalnych procedur z użyciem prostej procedury zdalnej skłania do wniosku, że narzut czasowy wnoszony przez pakiet procedur zdalnych jest znaczący i uwidacznia się przy częstych wywołaniach tych procedur. Zupełnie odmienne wyniki uzyskano dla tej samej konfiguracji, jednak używającej jednej procedury zdalnej, realizującej całą operację złożoną. Następuje tu pojedyncze wywołanie takiej procedury; osiągnięto minimalizację liczby transmisji przez sieć.

Zwróćmy jeszcze uwagę na czas tzw. aktualizacji wtórnej, w której nowa wartość rekordu jest taka sama jak poprzednia. Czas takiej aktualizacji jest wydatnie krótszy. Pozwala to stwierdzić, że podczas aktualizacji wtórnej nie jest dokonywana reorganizacja kluczy lub nawet nie jest dokonywany zapis do zbioru danych.

Wypada się również zastanowić nad wynikami uzyskanymi podczas równoczesnego wykonywania obliczeń przez stacje robocze. Tu także najkorzystniej prezentuje się konfiguracja zdalnych procedur z jedną procedurą specjalną. Warto przy tym zwrócić uwagę na fakt, że przy zwiększonej liczbie stacji praktycznie nie wzrasta jednostkowy czas wykonania obliczeń. Nie jest tak w przypadku pozostałych rozwiązań. Ilustruje to tabela wzrostów:

Tabela 2

Zestawienie przyrostów czasowych przy obsłudze zwiększonej ilości stacji roboczych

Rodzaj środowiska	Przyrost (w procentach)
Środowisko sieciowe	85
Środowisko z przekazem zleceń	70
Środowisko zdalnych procedur z użyciem prostych procedur zdalnych	185
Środowisko zdalnych procedur z użyciem procedury specjalnej	3

11. Ocena wyników

Otrzymaliśmy wyniki, które pozwalają nam ocenić stosowane konfiguracje programowe. Ogólnie zarysowała się następująca tendencja: operacje bardzo proste, oparte na pojedynczym przesyle przez łącza komunikacyjne najefektywniej są wykonywane w środowiskach lokalnym i sieciowym z lokalnym menadżerem zbiorów danych, a więc tam, gdzie stosowane są standardowe mechanizmy dostępu do danych (w konfiguracji lokalnej był to pojedynczy odczyt

danych z dysku lokalnego, natomiast w konfiguracji sieciowej użyteczne były funkcje sieciowego systemu operacyjnego). Natomiast wraz ze wzrostem złożoności badanych operacji dostrzega się zalety konfiguracji sieciowych i to tych specjalizowanych (przekazywanie zleceń oraz pakiet procedur zdalnych). Można wręcz powiedzieć, że dla operacji średnio złożonych najkorzystniejszym rozwiązaniem jest wykorzystanie koncepcji przekazywania zleceń. W tym środowisku zmierzone czasy były najkrótsze. Natomiast w przypadku operacji złożonych górę bierze środowisko z pakietem zdalnych procedur, ale tylko w przypadku stosowania specjalizowanych procedur zdalnych. Nie należy jednak zapominać o tym, że pamięć zewnętrzna na wykorzystywanej w badaniach stacji roboczej posiadała niekorzystne parametry techniczne (duży czas dostępu do dysku), co może zaburzyć obiektywność ocen.

W przypadku jednoczesnego wykonywania operacji zauważono następującą tendencję: im bardziej złożona jest operacja do wykonania, tym mniejsza jest progresja wzrostu łącznego czasu wykonania danej operacji w funkcji liczby jednocześnie działających procesów użytkowych. Operacje złożone są mniej wrażliwe na wielodostęp. Jeśli zaś chodzi o ocenę równowagi konkretnych rozwiązań, to najbardziej stabilna jest konfiguracja z przekazywaniem zleceń, chociaż w odniesieniu do operacji złożonych ustępuje ona wyraźnie pakietowi procedur zdalnych.

Na podstawie uzyskanych wyników można zasugerować pewne zalecenia praktyczne:

- do wykonywania operacji prostych celowe jest stosowanie rozwiązań z przekazywaniem zleceń,
- do wykonywania operacji złożonych najkorzystniej jest realizować rozwiązanie oparte na koncepcji procedur zdalnych.

Musimy jednak wziąć po uwagę fakt, że oba proponowane rozwiązania obciążają serwer plików, co w przypadku słabszych maszyn, wykorzystywanych jako serwer sieciowy, może doprowadzić do zmniejszenia efektywności wykonania podstawowych zadań takiego serwera. Dlatego należy wprowadzać powyższe rozwiązania z rozwagą i po odpowiednich analizach obciążenia serwera sieci.

LITERATURA

- [1] Netware 386 v.3.11 *Concepts*, 1991.
- [2] Network C for DOS *Btrieve Programmer's Manual for C v.5.1*, 1990.
- [3] Btrieve Installation and Operation, 1991.
- [4] C Network Compiler/386 v.7.0 *Language Reference*, 1989.
- [5] C Network Compiler/386 v.7.0 *NetWare Loadable Module Library Reference*, 1989.
- [6] C Network Compiler/386 v.7.0 *Client/Server Tutorial*, 1989.
- [7] NetWare RPC User's Manual, 1989, 1991.
- [8] NetWare RPC NetWare 386 and SPX Supplement, 1989.

- [9] Wolisz A.: *Podstawy lokalnych sieci komputerowych - Oprogramowanie komunikacyjne i usługi sieciowe*, tom 2. WNT, Warszawa 1992.

Recenzent: Dr inż. Maciej Bargielski

Wpłynęło do Redakcji: 16 stycznia 1995 r.

Abstract

The work contains the analysis of effectivity of various configurations of software used for manipulating data files in the NetWare environment.

Four model configurations that are often occurring in real systems were chosen. These are:

- local system
- network with the local file manager
- sending requests
- remote procedures

The models were described in the chapters from 3 to 6. The base for comparison was measured execution time of some chosen operations performed in these four models. The results are shown with bar charts in logarithmic scale. The following conclusions were obtained after the analysis: The simple operations consisting in a single data access are performed more efficiently in models which are less technically advanced i.e. local system and network with the local file manager. The opposite can be observed about the complex operations with multiple data access and multiple operating system calls. For these the advanced solutions, such as sending requests and remote procedures, show the advantage. The remote procedures environment is particularly suitable for multirecord processing. However, this solution puts an additional burden onto the server and this may influence the server's performance even for regular tasks.

The work also states that for more complex operations the increase of the elapsed execution time with the same increase of the number of users concurrently accessing data is less than the one for simple operations. The conclusion is easy - the requests and remote procedure models are more suitable for multiple access environments. The author maintains that the examined configuration models are so universal that the given conclusions may apply to other network environments.