ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 35 Nr kol. 1402

Wojciech KADŁUBOWSKI
Politechnika Szczecińska, Instytut Elektroniki i Informatyki

SZACOWANIE EFEKTYWNOŚCI SYSTEMÓW REPLIKACJI BAZ DANYCH - ANALIZA NIEZAWODNOŚCIOWA[1]

Streszczenie. W pracy omówiono metodę szacowania efektywności replikowanych baz danych. Zaprezentowano budowę i działanie takich systemów oraz właściwości wynikające ze stosowania różnych metod replikacji. Przedstawiono modele systemu pozwalające porównywać różne algorytmy replikacji, jak również różne konfiguracje stosujące tę samą metodę replikacji.

EVALUATION OF EFFECTIVENESS OF REPLICATION DATABASE SYSTEMS - RELIABILITY ANALYSIS

Summary. In this paper the method of evaluation of effectiveness of replication database systems were discussed. The architecture and properties of such systems working by different replication methods has been shown. The reliability system model for comparison the different replication algorithms was presented.

1. Wstęp

Przykładem systemu rozproszonego może być replikacja (częściowa lub całkowita) bazy danych. W wypadku replikacji baza danych (lub jej część) zostaje skopiowana i umieszczona na kilku serwerach w sieci (lokalnej lub rozległej). Aplikacje klientów uaktualniają bazę pierwotną, natomiast system sam rozsyła zmiany dokonane w bazie do jej replik.

Ponieważ są to systemy złożone, spełniające różne funkcje użytkowe (np. możliwość aktualizacji danych w danym węźle, a następnie propagacja dokonanych zmian do pozostałych węzłów lub też umożliwienie tylko odczytu danych z systemu w konkretnym węźle),

[1] Praca wykonana w ramach grantu KBN Nr 8T11C01612.

porównanie ich właściwości może nastręczać pewne trudności. Problemem może być porównanie różnych systemów replikacji (np. rozwiązania typu Master - Slave i typu Point - to Point), jak również konkretnych rozwiązań bazujących na wybranym typie replikacji (np. przy czterech węzłach systemu replikacyjnego należy wybrać taką konfigurację, aby uzyskać największą niezawodność przy minimalnych kosztach).

W literaturze spotkać można opisy różnych, szczegółowych (czasem tylko teoretycznych) przypadków replikacji. Daje się jednak odczuć brak uogólnionego modelu umożliwiającego porównywanie algorytmów replikacji. Za pomocą takiego modelu można by dobierać algorytm replikacji (spełniający określone parametry, np.: niezawodność) do zastosowania w konkretnym przypadku.

Wyniki prezentowanej pracy mogą znaleźć zastosowanie praktyczne przy konstrukcji systemu replikacyjnego wymagającego określonego poziomu efektywności i niezawodności.

2. Aspekty techniczne replikacji

Istnieje kilka koncepcji replikacji baz danych. Najczęściej używane są systemy na typu peer-to-peer oraz master-slave.

W systemach master-slave istnieje jedna baza pierwotna, którą użytkownicy mogą uaktualniać oraz bazy wtórne (repliki bazy pierwotnej) umożliwiające tylko odczyt danych. Uaktualnienia dokonane w bazie pierwotnej system replikacyjny rozsyła do baz wtórnych. Zapewnia się w ten sposób spójność danych w całym systemie. Przesyłanie informacji przeznaczonej do uaktualnienia replikacji może odbywać się na kilka sposobów. Informacja ta może być przesyłana natychmiast po uaktualnieniu bazy. Możliwa jest również zastosowanie tzw. migawki (snapshot), czyli przesłania wszystkich zmian dokonanych w bazie w określonej chwili (np. co 10 minut). Czynnikiem powodującym rozpoczęcie procesu uaktualniania replikowanych danych może być również ilość zmian dokonanych w bazie (tablicy, wierszu).

W koncepcji peer-to-peer wszystkie bazy są traktowane równorzędnie. Replikatory mogą wysyłać i odbierać dane od innych replikatorów na zasadzie równoprawnych węzłów systemu; nie ma pomiędzy nimi takiej zależności, jak w przypadku master-slave. Użytkownik może uaktualniać każdą z nich. System sam rozstrzyga ewentualne konflikty związane z próbą uaktualnienia tych samych danych w różnych miejscach równocześnie.

Przy rozsyłaniu danych do wielu miejsc stosuje się różne warianty połączeń pomiędzy replikami. Jednym z nich jest połączenie kaskadowe. Innym wariantem są połączenia baz replikowanych w strukturę typu gwiazda. Istnieją również rozwiązania mieszane.

3. Model systemu replikacyjnego

Na rys. 1 przedstawiono uogólniony model systemu replikacyjnego. Na wyższym poziomie, "logicznym" obejmuje on bazy danych interfejsy do bazy na stacjach klientów, aplikacje bazodanowe na serwerach i ewentualnie na stacjach klientów, łącza logiczne oraz elementy reprezentujące replikację i moduły rozwiązywania konfliktów. Łącze logiczne obejmuje łączność między bazami danych, jak również pomiędzy interfejsami klientów i bazami. Podobnie przedstawia się model systemu rozproszonego na niższym poziomie "fizycznym". Zamiast baz danych i aplikacji mamy serwery i stacje robocze, natomiast sieć komputerowa zastępuje łącza logiczne. Przy szacowaniu niezawodności oraz efektywności poszczególnych rozwiązań można brać pod uwagę replikację na poziomie logicznym (np. replikacja bazy danych), jak również na poziomie fizycznym (replikacja serwerów, stacji roboczych).

Jak przedstawiono na rys.1 stacja klienta systemu składa się z kilku elementów: interfejsu użytkownika do bazy danych, stacji roboczej traktowanej jako komputer z systemem operacyjnym, a także aplikacji bazodanowej uruchomionej na stacji roboczej (element opcjonalny). W skład serwera bazy danych wchodzą: serwer (traktowany jako komputer plus system operacyjny), motor bazy danych wraz z plikami bazy, moduł replikatora. Moduł rozwiązywania konfliktów uruchamiany na serwerze nie został zawarty w serwerze bazy, ponieważ nie występuje we wszystkich typach replikacji. Zarówno komunikacja rozległa, jak i lokalna składa się z łączy logicznych (powyżej czwartej warstwy OSI/ISO) i sieci komputerowych (poniżej czwartej warstwy OSI/ISO).

Niezawodność działania stacji klienta jest obliczana jako iloczyn niezawodności poszczególnych elementów.

$$P_W = P_{WS} \cdot P_{INT} \cdot P_{APW} \tag{1}$$

gdzie:

P_W – prawdopodobieństwo działania stacji klienta,

P_{WS} – prawdopodobieństwo działania stacji roboczej,

P_{INT} – prawdopodobieństwo funkcjonowania interfejsu użytkownika do bazy danych,

P_{APW} – prawdopodobieństwo działania aplikacji bazodanowej na stacji roboczej.

Podobnie jest w pozostałych przypadkach.

$$P_L = P_{LL} \cdot P_{SL} \tag{2}$$

gdzie:

P_L – prawdopodobieństwo funkcjonowania komunikacji lokalnej,

P_{LL} – prawdopodobieństwo funkcjonowania lokalnego łącza logicznego,

P_{SL} – prawdopodobieństwo funkcjonowania lokalnej sieci komputerowej.

$$P_{SB} = P_S \cdot P_{DB} \cdot P_{RE} \quad (3)$$

gdzie:

P_{SB} – prawdopodobieństwo działania serwera bazy danych,

P_S – prawdopodobieństwo funkcjonowania serwera,

P_{RE} – prawdopodobieństwo funkcjonowania replikatora,

P_{DB} – prawdopodobieństwo funkcjonowania motoru bazy danych.

$$P_R = P_{LR} \cdot P_{SR} \quad (4)$$

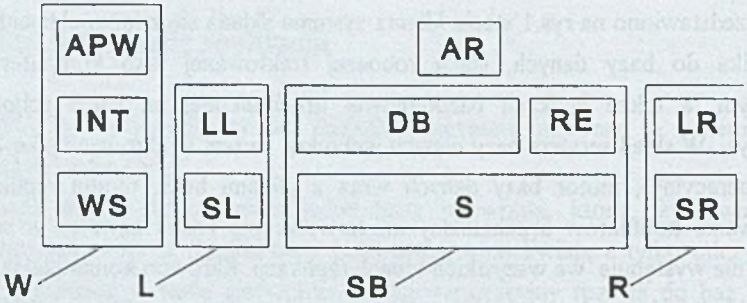
gdzie:

P_R – prawdopodobieństwo działania komunikacji rozległej,

P_{SR} – prawdopodobieństwo działania rozległej sieci komputerowej,

P_{LR} – prawdopodobieństwo funkcjonowania łącza rozległego.

Opierając się na tak zdefiniowanym modelu systemu replikacji na poziomie logicznym i fizycznym przeprowadzono obliczenia.



INT - interfejs użytkownika do bazy danych, W - stacja klienta systemu, WS - stacja robocza (komputer z systemem operacyjnym), SL - sieć komputerowa lokalna, SR - sieć komputerowa rozległa, LL - łącze logiczne lokalne, LR - łącze logiczne rozległe, APW - aplikacja na stacji roboczej, DB - motor bazy danych, pliki bazy danych, RE - moduł replikacji, S - serwer (komputer z systemem operacyjnym), L - komunikacja lokalna (składająca się z LL i SL), SB - serwer bazy (składający się z S, DB, RE, APS), R - komunikacja rozległa (składająca się z LR i SR), AR - moduł rozwiązywania konfliktów.

Rys. 1. Model systemu replikacji bazy danych
Fig. 1. Model of database replikation system

4. Metoda obliczania efektywności systemu replikacyjnego

W przypadku dużych systemów często można spotkać się z sytuacją, w której część elementów systemu jest sprawna i spełnia swoje zadania, a pozostała część nie udostępnia wymaganych serwisów. Możemy określić stany systemu, w których układ jest w pełni

sprawny, częściowo sprawny lub też całkowicie niesprawny. Odpowiednie zbiory stanów systemu oznaczamy przez Z^+, Z^*, Z^-. W układach złożonych można zdefiniować funkcje cząstkowe f_j realizowane przez dany system (na podstawie [1]). Każdy ze stanów systemu realizuje w jakiś sposób funkcje cząstkowe zdefiniowane w tym układzie. Z pracą systemu w stanie z_i przy realizacji funkcji f_j wiąże się określony efekt wyjściowy. Wartość bezwzględną tego efektu uzyskaną w jednostce czasu oznaczamy przez w^j_i. Przyjmuje się, że dla poszczególnych stanów systemu spełnione są relacje:

$w^j_i = w^j_0$ dla $z_i \in Z^+$,

$0 < w^j_i < w^j_0$ dla $z_i \in Z^*$,

$w^j_i = 0$ dla $z_i \in Z^-$.

Możemy również zdefiniować względną wartość wyjściowego efektu działania systemu w stanie z_i przy realizacji funkcji f_j, czyli wskaźnik efektywności φ^j_i.

$$\varphi^j_i = \frac{w^j_i}{w^j_o} \tag{5}$$

Z każdą funkcją cząstkową f_j wiążemy funkcję efektywności F^j realizacji f_j przez system. Dla systemów o długim czasie działania określa się stacjonarną funkcję efektywności systemu realizującego funkcję cząstkową f_j.

$$F^j = \sum_{i=0}^{i=N} \varphi^j_i \cdot P_i \tag{6}$$

gdzie P_i jest prawdopodobieństwem stanu z_i dla systemów o długim czasie działania.

4.1. Określenie algorytmów replikacji

Symetryczna replikacja (peer-to-peer) umożliwia uaktualnianie na równorzędnych zasadach wielu kopii bazy danych przechowywanych w różnych węzłach systemu replikacyjnego. Uaktualnienia mogą być dokonywane w każdym węźle. Następnie są one rozsyłane do wszystkich innych replik. Zmiany w lokalnych kopiach danych są zapamiętywane lokalnie, a następnie periodycznie przesyłane w oddzielnych transakcjach. Uaktualnienia są rozsyłane w krótkich odstępach czasu lub w specyficznych punktach czasowych (np. gdy łączność jest tańsza). W przypadku niedostępności zdalnego węzła zmiany pozostają w lokalnym systemie i są przesyłane w momencie ponownej jego dostępności. Może się zdarzyć, że niektóre kopie będą chwilowo posiadać nieaktualne dane, jednak po pewnym czasie system replikacyjny powinien wszystkie je uaktualnić.

W węzłach typu slave tworzone są kopie tylko do odczytu, pozwalające na replikowanie tablicy głównej w innych węzłach systemu rozproszonego. Często nazywa się je migawkami, ponieważ mogą być periodycznie odświeżane, odzwierciedlając ostatni spójny stan tablicy

głównej. Bazując na jednej tablicy głównej w węźle master można stworzyć dowolną liczbę migawek tylko do odczytu w węzłach typu slave. Użytkownik nie może uaktualniać migawek tylko do odczytu w węźle slave. Wszelkie zmiany może on dokonywać tylko w tablicy głównej węzła master. Zmiany te są następnie propagowane przez system replikacyjny. Istnieją pewne modyfikacje związane z węzłami typu slave. Oprócz standardowego węzła typu slave istnieją także węzły slave z bazą chronioną tylko do odczytu oraz bazą uaktualnianą.

W trybie pracy chronionym baza danych w węźle slave otrzymuje informacje z innych węzłów replikacyjnych. Węzeł replikacyjny posiada moduł pozwalający na przyjęcie poprawnych (dopuszczalnych) danych z kilku serwerów typu master. Tryb chroniony replikacji wykorzystywany jest w węzłach zawierających dane tylko do odczytu, do których kilka innych węzłów replikacyjnych przesyła uaktualnienia danych.

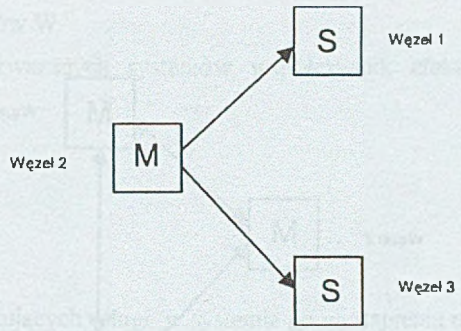
Uaktualniana baza w węźle slave różni się od migawki tylko do odczytu tym, że można w niej dokonywać uaktualnienia (w specjalnej perspektywie dostępnej dla użytkowników). Uaktualnienia dokonane w migawce są propagowane do powiązanej z nią tablicy głównej w węźle slave. W węźle master system porównuje stare dane przysłanych z migawki (tzn. wartości tych wierszy przed dokonaniem zmian) z wartościami bieżącymi tego węzła. Różne wartości oznaczają wystąpienie konfliktu. Wywoływana jest wtedy procedura rozwiązywania konfliktów. Działanie tej procedury może spowodować zmianę lub nawet usunięcie niektórych wierszy w migawce po jej odświeżeniu. Uaktualnienia dokonane w tablicy głównej w węźle master są przesyłane do migawki przy użyciu mechanizmu odświeżania, tak więc niepotrzebne są procedury wykrywania i rozwiązywania konfliktów w węźle migawki.

Obliczenia zostały przeprowadzone dla konfiguracji trzech serwerów replikacyjnych połączonych dwoma łączami rozległymi. Badane były cztery rodzaje algorytmów replikacyjnych:

- master - slave (rys. 2),
- peer - to - peer (rys. 4),
- węzeł slave chroniony (rys. 3),
- węzeł slave uaktualniany (rys. 5).

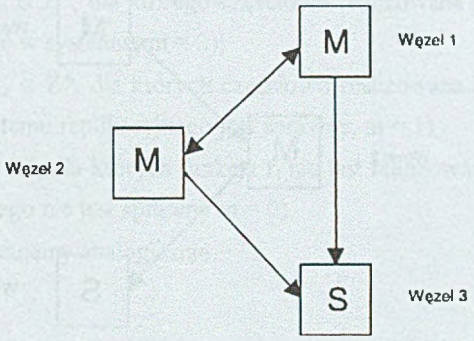
Na rys. 2 - 5 przedstawiono powiązania logiczne pomiędzy poszczególnymi węzłami.

Zakładamy, że dla każdej konfiguracji z rys. 2 - 5 węzeł systemu można przedstawić za pomocą modelu z rys. 1 (w bazach standardowych węzłów slave oraz w bazach uaktualnianych slave nie występuje moduł rozwiązywania konfliktów).



M - węzeł systemu replikacyjnego zawierający serwer bazy typu master,
S - węzeł systemu replikacyjnego zawierający serwer bazy typu slave.

Rys. 2. Konfiguracja Master - Slave
Fig. 2. Master Slave servers configuration

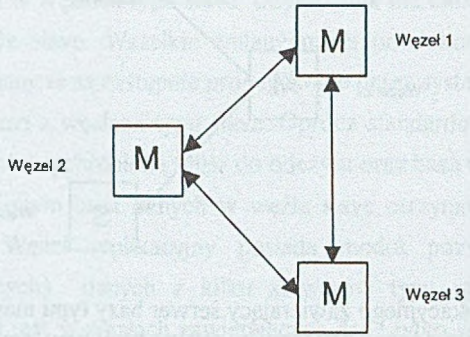


Rys. 3. Konfiguracja z chronionym serwerem Slave
Fig. 3. Protected Slave server configuration

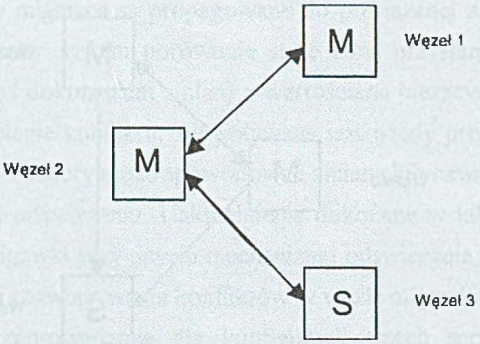
4.2. Określenie miar efektywności systemu replikacyjnego

Dla rozpatrywanych systemów replikacyjnych zdefiniowano następujące funkcje cząstkowe:

f_1 - m-węzłowa realizowalność usług n-węzłowego systemu replikacyjnego zdefiniowana następująco: dane są spójne w działających i mających ze sobą łączność m-węzłach n-węzłowego systemu replikacyjnego (nie uwzględniana jest chwilowa niespójność danych w różnych węzłach wynikająca z procesu replikacji) oraz bazy mogą oferować usługi przewidziane algorytmem replikacji,



Rys. 4. Konfiguracja Peer - to - Peer
Fig. 4. Peer - to - Peer configuration



Rys. 5. Konfiguracja z uaktualnianym serwerem Slave
Fig. 5. Updatable Slave configuration

f_2 - użytkownicy węzła k mogą aktualizować dane w bazie w swoim węźle (w swojej replice bazy), węzeł k zawiera spójne dane z działającymi m węzłami systemu, z którymi ma łączność,

f_3 - użytkownicy węzła k mogą odczytywać dane z bazy w swoim węźle (ze swojej repliki bazy), węzeł k zawiera spójne dane z działającymi m węzłami systemu, z którymi ma łączność.

W obliczeniach przyjęto występowanie awarii z wszystkich klas S_i zdefiniowanych następująco:

S_1- awarie modułów AR,

S_2- awarie serwerów bazy SB,

S_3- awarie sieci lokalnych L,

S_4- awarie łączy rozległych R,

S_5- awarie stacji klientów W.

Ustalamy, że dla rozważanych systemów współczynnik efektywności φ^j_i przyjmuje następujące wartości:

$$\varphi^1_i = \frac{w^1_i}{w^1_o} \tag{7}$$

$w^1_0 = n$

$w^1_i = m$

m - ilość węzłów realizujących usługi w systemie po wystąpieniu uszkodzeń klas S_t,

n - całkowita ilość węzłów zdefiniowanych w systemie replikacyjnym (w rozpatrywanej konfiguracji n = 3),

$\varphi^1_0 = 1$ dla stanu $z_0 \in Z^+$, dla którego w pełni realizowana jest funkcja f_1, tzn. dla $w^1_0 = 3$ (wszystkie węzły są całkowicie sprawne i system bazy replikowanej można uznać za spójny, m=3),

$\varphi^1_1 = 0{,}66$ dla stanu $z_1 \in Z^*$, dla którego częściowo realizowana jest funkcja f_1, $w^1_1 = 2$ (tzn. dwa węzły są sprawne w systemie, m = 2),

$\varphi^1_2 = 0{,}33$ dla stanu $z_2 \in Z^*$, dla których częściowo realizowana jest funkcja f_1, $w^1_2 = 1$ (tzn. tylko jeden węzeł systemu replikacyjnego jest sprawny, m = 1),

$\varphi^1_3 = 0$ dla stanu $z_3 \in Z^-$, dla których funkcja f_1 nie jest realizowana, $w^1_3 = 0$ (tzn. żaden węzeł systemu replikacyjnego nie jest sprawny m = 0).

Współczynnik φ^2_i definiujemy analogicznie:

$$\varphi^2_i = \frac{w^2_i}{w^2_o} \tag{8}$$

$w^2_0 = n$

$w^2_i = m$

$\varphi^2_0 = 1$ dla stanu $z_0 \in Z^+$, dla którego cały system jest sprawny (funkcja f_1 w pełni realizowana), sprawne stacje klientów systemu we wszystkich m = n =3 węzłach są w stanie uaktualniać dane w bazie lokalnej (w swoim węźle), $w^2_0 = 3$

$\varphi^2_1 = 0{,}66$ dla stanu $z_1 \in Z^*$, dla którego funkcja f_1 jest częściowo realizowana oraz sprawne stacje klientów systemu znajdujące się w m = 2 sprawnych węzłach są w stanie uaktualniać dane w bazie lokalnej (w swoim węźle) , $w^2_1 = 2$

$\varphi^2_2 = 0{,}33$ dla stanu $z_2 \in Z^*$, dla których funkcja f_1 jest częściowo realizowana oraz sprawna stacja klienta systemu znajdująca się w sprawnym węźle (m = 1) jest w stanie uaktualniać dane w bazie lokalnej (w swoim węźle) , $w^2_2 = 1$

$\varphi^2_3 = 0$ dla stanów $z_3 \in Z^-$, dla których nie jest realizowana funkcja f_1 lub żadna stacja klienta systemu nie jest w stanie uaktualniać dane , $w^2_3 = 0$.

Analogicznie do φ^2_i jest definiowany współczynnik φ^3_i, który przyjmuje wartości:

$\varphi^3_0 = 1$ $\varphi^3_1 = 0{,}66$ $\varphi^3_2 = 0{,}33$ $\varphi^3_3 = 0$

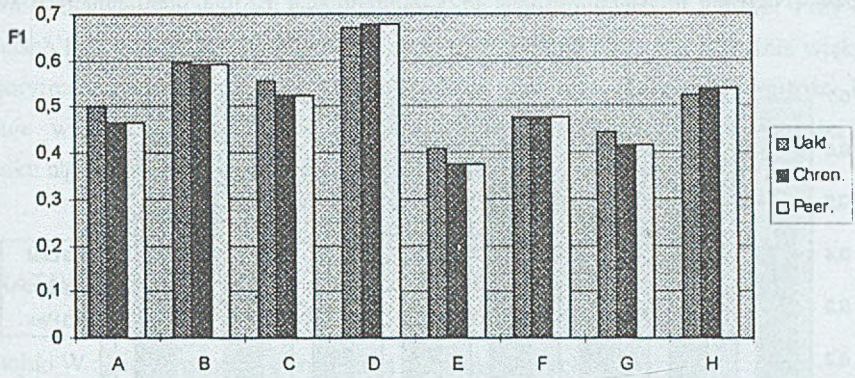
Należy zauważyć, że wyżej przedstawione stany z_i są stanami zagregowanymi, np. z_1 może być agregacją kilku stanów spełniających warunek pracy dwóch węzłów w rozpatrywanej konfiguracji systemu replikacyjnego.

5. Omówienie wyników

Oszacowania zostały wykonane dla następujących zestawów danych:

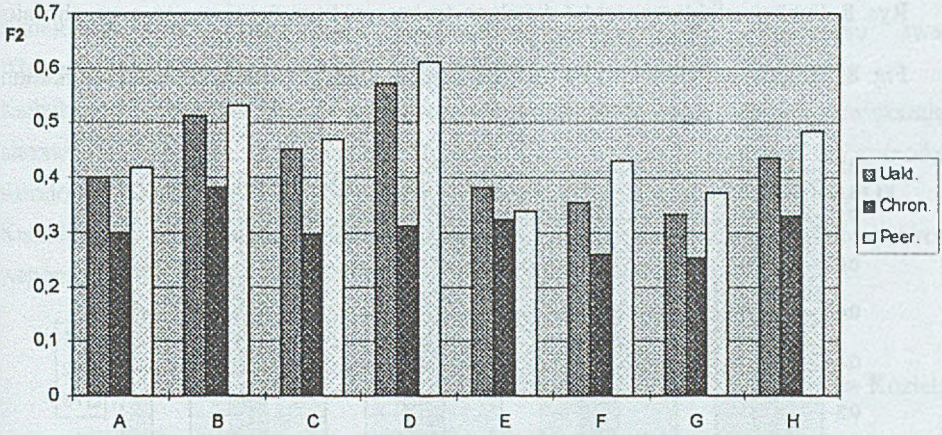
A	P_L=0,9	P_{SB}=0,95	P_R=0,7	P_{AR}=0,65	P_W=0,9
B	P_L=0,9	P_{SB}=0,95	P_R=0,7	P_{AR}=0,85	P_W=0,9
C	P_L=0,9	P_{SB}=0,95	P_R=0,9	P_{AR}=0,65	P_W=0,9
D	P_L=0,9	P_{SB}=0,95	P_R=0,9	P_{AR}=0,85	P_W=0,9
E	P_L=0,9	P_{SB}=0,75	P_R=0,7	P_{AR}=0,65	P_W=0,9
F	P_L=0,9	P_{SB}=0,75	P_R=0,7	P_{AR}=0,85	P_W=0,9
G	P_L=0,9	P_{SB}=0,75	P_R=0,9	P_{AR}=0,65	P_W=0,9
H	P_L=0,9	P_{SB}=0,75	P_R=0,9	P_{AR}=0,85	P_W=0,9
I	P_L=0,9	P_{SB}=0,95	P_R=0,7		P_W=0,9
J	P_L=0,9	P_{SB}=0,75	P_R=0,7		P_W=0,9
K	P_L=0,9	P_{SB}=0,95	P_R=0,9		P_W=0,9
L	P_L=0,9	P_{SB}=0,95	P_R=0,9		P_W=0,9

Obliczenia wartości funkcji efektywności F1, F2, F3 dla czterech algorytmów replikacji przedstawionych na rys.2-5 wykonano podstawiając do wzoru (5) różne wartości prawdopodobieństw działania poszczególnych elementów (zestawy danych A-L). Funkcja efektywności F1 przyjmuje największe wartości dla replikacji master-slave. Można zauważyć, że jej wartości w przypadku algorytmów peer-to-peer oraz slave chroniony są identyczne. Wynika to z faktu, iż w obu przypadkach we wszystkich węzłach występuje moduł AR. Przy małych wartościach P_{AR} funkcja F1 dla algorytmu typu uaktualniany slave jest wyraźnie większa niż dla algorytmów peer-to-peer oraz slave chroniony, natomiast dla dużych wartości P_{AR} F1 przyjmuje wartości porównywalne dla wszystkich trzech algorytmów.



Rys. 6. Funkcje efektywności F1 dla algorytmów replikacji typu peer-to-peer, chroniony slave oraz uaktualniany slave

Fig. 6. Effectiveness funcions F1 for peer-to-peer, updatable and protected replikation

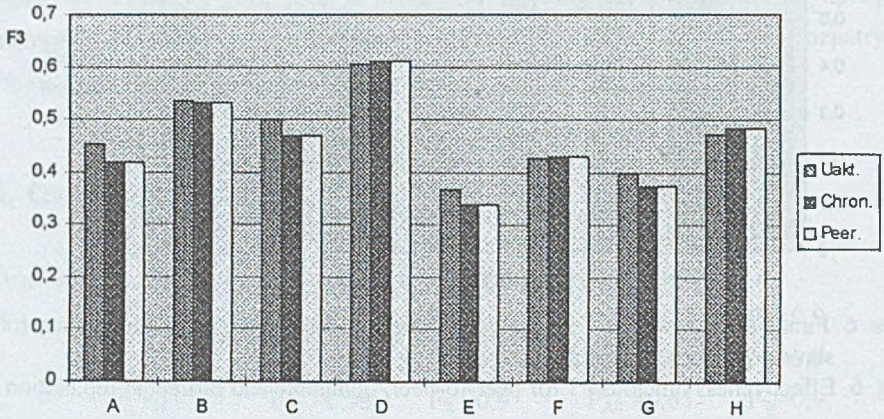


Rys. 7. Funkcje efektywności F2 dla algorytmów replikacji typu peer-to-peer, chroniony slave oraz uaktualniany slave

Fig. 7. Effectiveness funcions F2 for peer-to-peer, updatable and protected replikation

Brak modułu AR w uaktualnianym węźle slave możemy potraktować jako moduł AR o prawdopodobieństwie działania P_{AR} =1. Stąd też dla dużych wartości P_{AR} (dążących do jedności) uzyskujemy porównywalne wartości F1 dla wymienionych trzech algorytmów. Tak więc przy poprawnym działaniu modułu AR (tzn. rozwiązywaniu wszelkich konfliktów zachodzących w bazie replikowanej) różnice w wartości F1 dla wszystkich rozpatrywanych algorytmów zmniejszają się. Wartość funkcji efektywności F2 jest największa dla algorytmu

pee-to-peer. Niewiele mniejsze wartości przyjmuje funkcja F2 dla uaktualnianych węzłów slave.



Rys. 8. Funkcje efektywności F3 dla algorytmów replikacji typu peer-to-peer, chroniony slave oraz uaktualniany slave

Fig. 8. Effectiveness funcions F3 for peer-to-peer, updatable and protected replikation

Rys. 9. Funkcje efektywności F1, F2, F3 dla algorytmu replikacji typu master-slave

Fig. 9. Effectiveness funcions F1, F2, F2 for master - slave replikation

Dużo mniejsze wartości można zaobserwować dla węzła chronionego slave, a najmniejsze dla algorytmu master slave. Można zauważyć, że wartość funkcji efektywności F2 dla algorytmu chroniony slave w konfiguracji zawierającej małą liczbę serwerów master oraz dużą liczbę serwerów slave maleje, a przy dużej liczbie serwerów master oraz małej liczbie serwerów slave rośnie. W funkcji F3 dają się zauważyć podobne zależności jak w przypadku

funkcji F1. Największą wartość przyjmuje ona dla algorytmu master - slave. Przy małych wartościach P_{AR} funkcja F3 dla algorytmu typu uaktualniany slave jest wyraźnie większa niż dla algorytmów peer-to-peer oraz slave chroniony, natomiast dla dużych wartości P_{AR} F3 przyjmuje wartości porównywalne dla wszystkich trzech algorytmów. Wartości F3 w przypadku algorytmów peer-to-peer oraz slave chroniony są identyczne.

LITERATURA

1. Zamojski W.: Niezawodność i eksploatacja systemów, Wrocław 1981.
2. Wood A.: Predicting client/server availability, IEEE Computer, 1995.
3. Kadłubowski W.: Wykorzystanie replikacji baz danych w systemach rozproszonych, II Krajowa konferencja "Komputerowe wspomaganie badań naukowych KOWBAN '95", Wrocław 1995.
4. Kadłubowski W.: Odtwarzanie systemu replikowanego po wystąpieniu awarii. III Seminarium "Sieci komputerowe", Gliwice 1996.
5. Kadłubowski W.: Replikacja sprzętowa i programowa jako element zwiększający niezawodność systemu, II Konferencja "Informatyka na wyższych uczelniach dla gospodarki narodowej", Gdańsk 1996.
6. Kadłubowski W.: Metody replikacji baz danych, II Krajowa konferencja "Komputerowe wspomaganie badań naukowych" Polanica Zdrój, Wrocław 1996.

Recenzent: Prof. dr hab. inż. Stanisław Kozielski

Wpłynęło do Redakcji 28 stycznia 1998 r.

Abstract

In this paper the method of evaluation of effectiveness of replication database systems were discussed. The architecture and properties of such systems working by different replication methods has been shown. The reliability system model for comparison the different replication algorithms was resented. In chapter 2. the replication algorithms were discussed and in chapter 3. the reliability models of replication system components were presented.

In chapter 4. the method of evaluation of effectiveness of system was demonstrated (see formula 5). Diagrams presented on Fig.6 - 9. shows the effectiveness functions for component's parameters.