AUTOMATYKA • 2011 • Tom 15 • Zeszyt 2

Mirosław Kasper*, Grzegorz Dobrowolski*

Propozycja wysokoskalowalnej metody replikacji danych Theta dla rozproszonych systemów transakcyjnych

1. Wprowadzenie

Metoda replikacji Theta zaprojektowana jest w wielowarstwowej architekturze zapewniającej wysoką dostępność systemu replikacji. Warstwa middleware dla zaproponowanej metody jest rozproszona, co oznacza, że w każdej lokalizacji, w której odbywa się replikacja danych, znajduje się instancja oprogramowania middleware współpracująca z instancją bazy danych. Szczegóły architektury systemu replikacji zbudowanego w oparciu o metodę Theta przedstawione są w sekcji 2 artykułu.

Komponenty middleware realizujące poszczególne zadania związane z procesem wykonywania transakcji w replikach przedstawione są w sekcji 3.

W metodzie replikacji Theta transakcje użytkowników (globalnie uporządkowane są przez niezależny generator unikalnych identyfikatorów) dostarczane są (bez konieczności zachowania jakiegokolwiek porządku) do wszystkich baz danych (replik), gdzie następuje ich uszeregowanie zapewniające uzyskanie identycznych wyników we wszystkich replikach. Middleware używa własnego mechanizmu kontroli współbieżności przetwarzania transakcji. Mechanizm ten realizuje algorytm przeciwdziałający wystąpieniu w bazie konfliktów na poziomie pojedynczych transakcji (*Conflict Prevention, algorithm*), co umożliwia zrównoleglenie przetwarzania transakcji w poszczególnych kopiach danych zgodnie z teorią szeregowalności jednej kopii (*1-Copy Serializability*) [2].

W sekcji 4 artykułu omówione zostały szczegóły równoległego przetwarzania transakcji podczas realizacji procesu replikacji danych, natomiast sekcja 5 zawiera informacje dotyczące algorytmu zapobiegającemu występowaniu konfliktów transakcji.

2. Architektura

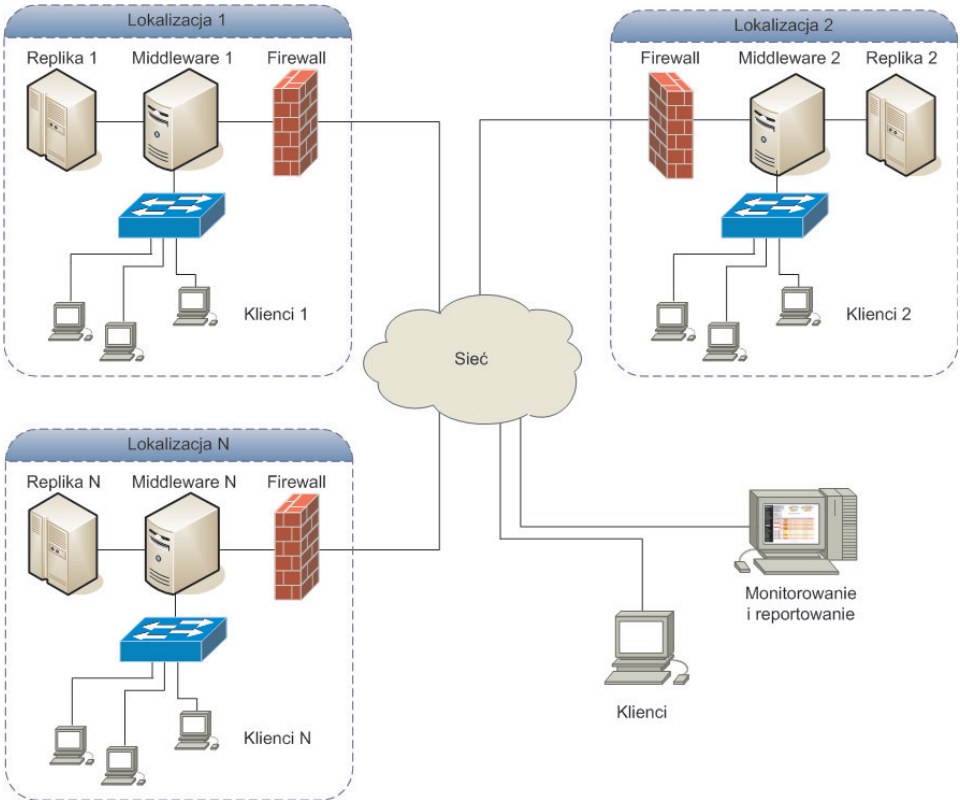
Na rysunku 1 przedstawiona jest ogólna architektura systemu replikacji danych zaimplementowanego w oparciu o metodę replikacji Theta.

* AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Informatyki

Replikacja danych w oparciu o metodę Theta realizowana jest w wielowarstwowej architekturze, w której warstwa middleware oraz instancje baz danych rozproszone są pomiędzy poszczególne lokalizacje w systemie.

System zarządzania bazą danych oraz oprogramowanie działające w warstwie middleware mogą być uruchomione na wspólnej maszynie lub mogą działać na osobnych dedykowanych maszynach. Metoda umożliwia używanie różnych baz danych w poszczególnych lokalizacjach, np. Oracle w lokalizacji 1 i 2, PostgreSQL w lokalizacji 3 oraz IBM DB2 dla lokalizacji 4, 5 i 6. Jednocześnie bazy danych w poszczególnych lokalizacjach mogą wspierać różne technologie związane z lokalnym zabezpieczeniem systemu, np. pojedyncza instancja bazy w lokalizacji 1, klaster wydajnościowy w lokalizacji 2, czy dalsza replikacja danych oparta o mechanizmy Hot Standy czy też replikację sprzętową w lokalizacji 3.

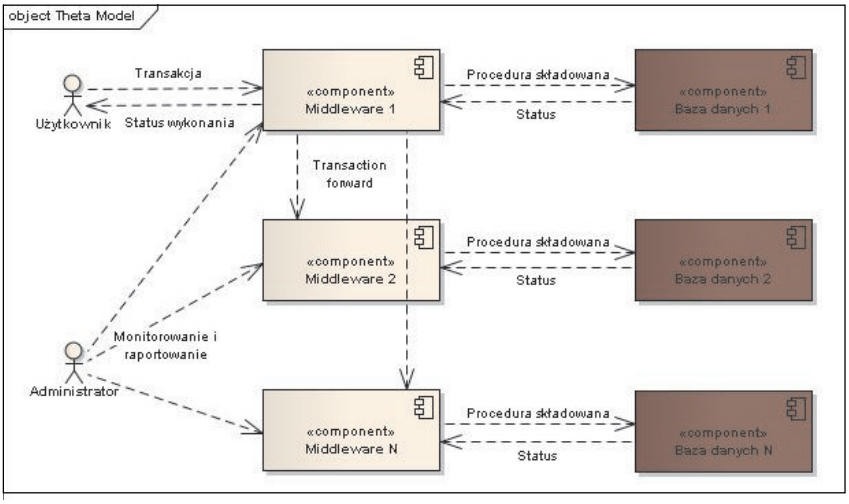
Klienci wykonują swoje transakcje, klienci łączą się z bazą danych za pośrednictwem oprogramowania middleware, które używa sterowników właściwych dla poszczególnych replik danych. Mogą to być zatem sterowniki natywne, np. OCI dla baz Oracle, czy też uniwersalne oprogramowanie zapewniające dostęp do danych w bazie jak np. ODBC, JDBC, OLE-DB itp.



Rys. 1. Architektura systemu replikacji

W celu połączenia z middleware klienci używają specjalnego sterownika zrealizowanego na potrzeby implementacji tej komunikacji dla metody Theta. Sterownik ten (*Theta Connector*) jest stosunkowo prostym i szybkim programem, który automatycznie konwertuje transakcje użytkowników w tablice parametrów i następnie przesyła je do middleware.

Szczegółowy sposób przekazywania danych w systemie zbudowanym na podstawie metody Theta przedstawiony jest na rysunku 2. Jeżeli dostarczona do middleware transakcja jest transakcją nową, wówczas przekazywana jest do wszystkich pozostałych lokalizacji, w przeciwnym wypadku przetwarzana jest tylko lokalnie. Dodatkowo, również transakcje odczytu (*Read Only*) domyślnie przetwarzane są tylko w bazie lokalnej dla tego middleware, do którego transakcja została dostarczona.



Rys. 2. Przepływ danych w systemie z replikacją Theta

Transakcja użytkownika jest przesyłana do warstwy middleware, gdzie następnie jest przekazywana do wszystkich pozostałych lokalizacji. Węzeł, do którego nowa transakcja trafia bezpośrednio od klienta, nazywany jest Zarządcą Transakcji (*Middleware Transaction Manager*). Dana transakcja może być zarządzana tylko przez jednego Zarządcę Transakcji, który inicjowany jest w dowolnej instancji middleware dla tej transakcji.

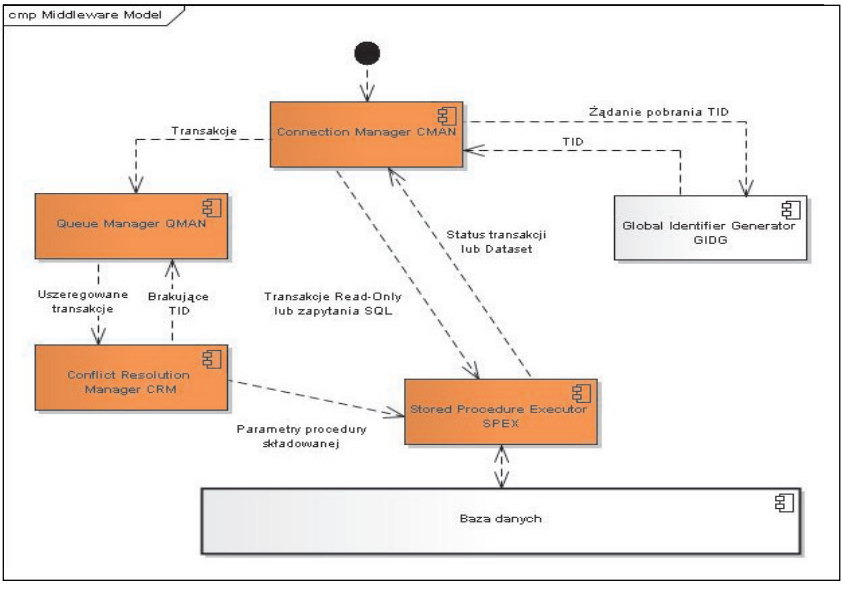
Jednocześnie z przesyłaniem danych do pozostałych lokalizacji Zarządca Transakcji przekazuje transakcję do dalszego procesowania związanego z wykrywaniem konfliktów i następnie wykonaniem w bazie danych jako wywołanie procedury składowanej. Po zakończeniu wywołania odpowiedniej procedury w bazie Zarządca Transakcji otrzymuje właściwy status transakcji z bazy danych i następnie przekazuje go w odpowiedzi do klienta. Pozostałe węzły systemu replikacji nie informują o wykonaniu transakcji ani klienta, ani Zarządcy Transakcji obsługującego tę transakcję. Przekazują one tylko transakcję do wykonania w lokalnej bazie danych i weryfikują status jej wykonania.

System pozwala na konfiguracje bezpośrednich połączeń pomiędzy klientem i dedykowanym middleware. Pomimo faktu, iż użytkownicy mogą łączyć się z dowolnymi instancjami middleware, rekomendowanym rozwiązaniem jest zestawienie połączenia klienta do najbliższego middleware, co oczywiście znacząco obniża obciążenie związane z komunikacją sieciową.

3. Metoda replikacji Theta

W każdej lokalizacji systemu replikacji opartego o metodę Theta znajduje się instancja middleware oraz instancja bazy danych. Możliwe jest również uruchomienie zarówno instancji middleware, jak i bazy danych w konfiguracji klastrowej w celu zapewnienia większej odporności na awarie i/lub zwiększenia wydajności systemu. Oprogramowanie middleware usytuowane jest pomiędzy klientem a bazą danych.

Transakcja klienta w formie tabeli parametrów przekazywana jest do middleware, gdzie następnie realizowany jest proces zapobiegania wystąpieniu konfliktów na poziomie transakcji. Parametry transakcji są następnie dekodowane na wywołania procedur składowanych odpowiednie dla poszczególnych kopii baz danych i następnie wykonywane są w tych replikach. Procedury te mogą wykonywać prostsze operacje wstawiania nowych czy modyfikacji już istniejących danych, jak również bardzo złożone operacje wykonywane na wielu tabelach. Jedna procedura wywoływana przez middleware związana jest z pojedynczą transakcją klienta.



Rys. 3. Komponenty middleware

Middleware składa się z pięciu głównych komponentów, które komunikują się wzajemnie w celu wykonania działań, które zagwarantują takie same wyniki wykonania transakcji we wszystkich zdalnych replikach pomimo to, że są one dostarczane w różnej kolejności. Poniżej znajduje się opis poszczególnych komponentów systemu.

CMAN (*Connection Manager*) stanowiący interfejs obsługujący połączenia klienta z middleware. Po otrzymaniu transakcji od klienta CMAN dodaje transakcję do lokalnej kolejki wejściowej transakcji i jednocześnie w niezależnych wątkach przekazuje transakcję do pozostałych węzłów systemu replikacji. CMAN zarządza również połączeniami z procesem GIDG.

GIDG (*Global Identifier Generator*) dostarcza unikalny, sekwencyjny identyfikator dla transakcji.

QMAN (*Queue Manager*) jest procesem, który obsługuje wejściową kolejkę transakcji. QMAN wybiera z kolejki wejściowej transakcje z ciągłymi identyfikatorami transakcji i przekazuje je do kolejki wejściowej procesu CRM.

CRM (*Conflict Resolution Manager*) zarządza procesem zapobiegania występowaniu konfliktów na poziomie transakcji w bazie danych. CRM zapewnia odpowiednią kolejność wykonywania transakcji w bazie danych (transakcje te dostarczone są wcześniej w dowolnej kolejności).

SPEX (*Stored Procedure Executor*) wywołuje procedury składowane w lokalnej bazie danych na podstawie danych wejściowych otrzymanych od procesu CRM. W celu wykonania procedur w bazie używa natywnego sterownika lub uniwersalnego API dla poszczególnych replik. Transakcje umieszczone przez proces CRM w wejściowej kolejce SPEX wykonywane są w bazie danych współbieżnie.

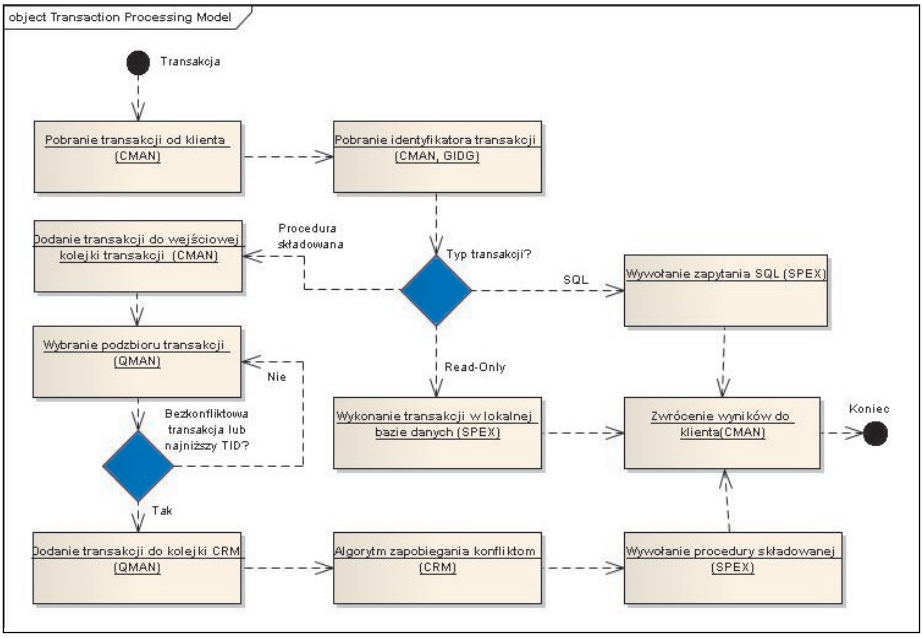
4. Obsługa transakcji

W zależności od typu transakcji oprogramowanie middleware przeprowadza określone operacje. Jeżeli przetwarzana transakcja jest transakcją odczytu danych, wykonywana jest tylko w lokalnej bazie danych. Oczywiście proces zapobiegania występowania konfliktów jest w takiej sytuacji niepotrzebny i transakcja jest wykonywana w bazie danych natychmiast po jej otrzymaniu. W przeciwnym wypadku, gdy transakcja modyfikuje, dodaje lub usuwa dane, transakcje muszą być przeanalizowane pod kątem możliwości wystąpienia konfliktów.

Transakcja klienta złożona jest z zestawu globalnych parametrów, nazw wywoływanych procedur składowanych oraz powiązanych argumentów. Po odebraniu transakcji przez proces CMAN, proces ten pobiera identyfikator transakcji od procesu GIDG i dodaje ten identyfikator do zestawu parametrów definiujących transakcję. Następnie tak przygotowana tablica parametrów przekazywana jest do wszystkich węzłów systemu replikacji, gdzie trafia do odpowiednich kolejek wejściowych. W następnym kroku w każdej instancji middleware proces QMAN wybiera z kolejki wejściowej zestaw kolejnych transakcji, które

z kolei umieszcza w kolejce wejściowej procesu CRM. CRM realizuje następnie algorytm przeciwdziałający wystąpieniu w bazie konfliktów na poziomie pojedynczych transakcji, dzięki czemu ustala odpowiednio kolejność wykonywania transakcji, umożliwiając jednocześnie zrównoleglenie ich realizacji. Transakcje te następnie przekazywane są do procesu SPEX, który dekoduje parametry je definiujące na wywołania procedur składowanych i wykonywane w replikach bazy. Transakcje, które są wzajemnie konfliktowe zgodnie z teorią szeregowalności jednej kopii, nie mogą być wykonywane jednocześnie i proces SPEX wywołuje je w bazie sekwencyjnie.

Rysunek 4 przedstawia ideę procesowania transakcji podczas replikacji danych opartej o zastosowanie metody Theta. Zadania wykonywane podczas obsługi transakcji realizowane są przez poszczególne procesy middleware – nazwy procesów zaangażowanych w poszczególne zadania przedstawione są w nawiasach.

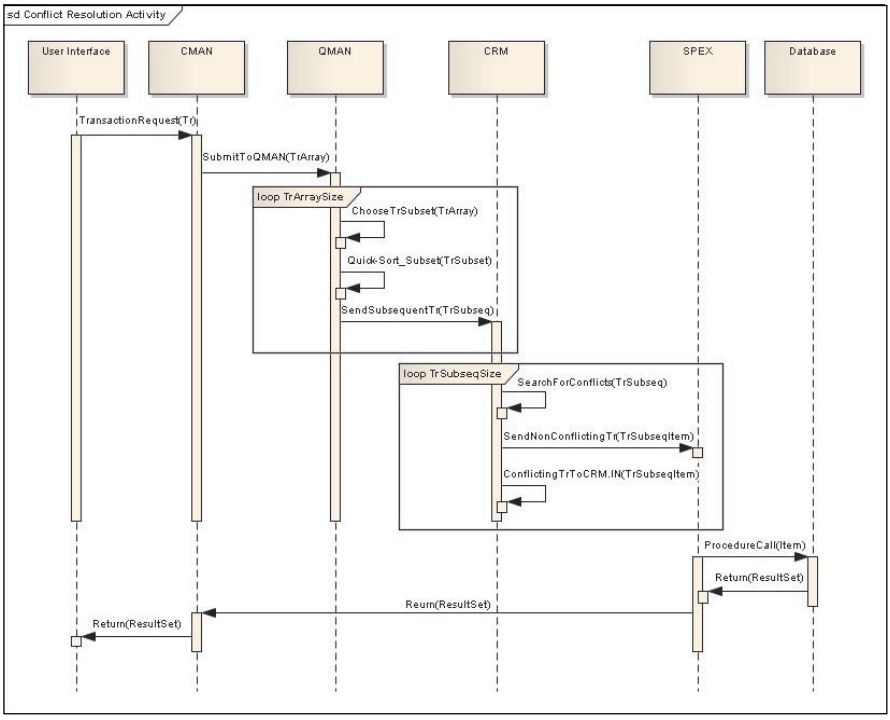


Rys. 4. Obsługa transakcji

W zależności od rodzaju transakcji (odczyt, zapytanie SQL lub procedura składowana) realizowane są kolejne kroki algorytmu przetwarzania transakcji zaprezentowanego na rysunku 4. W przypadku transakcji przekazanej przez Zarządcę Transakcji zdalny proces CMAN nie przesyła jej do kolejnych węzłów, o czym decyduje na podstawie zawartości identyfikatora transakcji – w przypadku przesyłanej transakcji ma on już przydzielony identyfikator.

5. Zapobieganie występowaniu konfliktów

Kluczowym elementem w realizacji metody Theta jest proces zapobiegania występowaniu konfliktów na poziomie transakcji. Diagram procesu zapobiegania konfliktom przedstawiony jest na rysunku 5.



Rys. 5. Diagram procesu zapobiegania konfliktom transakcji

Proces zapobiegania występowaniu konfliktów określa optymalną kolejność wykonywania transakcji, zapewniającą jednocześnie spójność i identyczność danych we wszystkich kopiach (replikach). Właściwa kolejność wykonywania transakcji w bazie danych jest określana niezależnie od kolejności dostarczania transakcji do middleware.

Po zakończeniu procesu zapobiegania występowaniu konfliktów transakcje bezkonfliktowe wykonywane są w bazie współbieżnie, podczas gdy transakcje konfliktowe są odpowiednio szeregowane i wykonywane oddzielnie.

6. Podsumowanie

Metoda replikacji Theta zaprojektowana jest w architekturze wielowarstwowej z rozproszoną warstwą middleware, dzięki czemu zapewnia wysoką skalowalność systemu

replikacji. W metodzie nie są stosowane rozproszone blokady dla transakcji (*transaction distributed locks*), dzięki czemu wykrywanie i rozwiązywanie zakleszczeń (*deadlocks*) odbywa się tylko na poziomie pojedynczej instancji bazy danych. Wyeliminowanie konieczności używania rozproszonych blokad w metodzie znacznie zwiększa jej wydajności, gdyż nie ma konieczności stosowania transakcji rozproszonych.

Mechanizm rozwiązywania konfliktów zaprojektowany na potrzeby metody Theta jest stosunkowo prosty w działaniu, co w połączeniu z niską złożonością czasową zapewnia wysoką efektywność metody. System replikacji oparty o metodę Theta jest kompatybilny z większością obecnie dostępnych systemów baz danych. Jednocześnie metoda oparta jest na wykonywaniu procedur składowanych (*Stored Procedures*) w zdalnych replikach, co umożliwia jej bezproblemowe używanie w środowiskach heterogenicznych działających na różnych platformach sprzętowych, z różnymi systemami operacyjnymi i systemami zarządzania baz danych. Zastosowanie prekompilowanych w bazie procedur składowanych przekłada się również na podwyższenie wydajności replikacji danych.

Zaprojektowany na potrzeby metody proces zapobiegania występowaniu konfliktów umożliwia dostarczanie transakcji według dowolnej kolejności, dzięki czemu obciążenie systemu związane z obsługą komunikacji jest w bardzo dużym stopniu zredukowane, co znacząco podwyższa całkowitą efektywność procesu replikacji danych.

Metoda replikacji Theta używa zarówno komponentów opisanych w literaturze czy też zaimplementowanych w istniejących rozwiązaniach (np. globalne identyfikatory transakcji, architektura z rozproszoną warstwą middleware) oraz nowych komponentów (algorytm zapobiegania występowaniu konfliktów, sposób komunikacji wewnątrz middleware i pomiędzy procesami zdalnymi). Dzięki zastosowanym rozwiązaniom metoda oferuje wysoką skalowalność replikacji danych, przy jednoczesnym zachowaniu dużej wydajności i odporności na awarie. Nowa metoda jest również przystosowana do pracy na różnych platformach sprzętowych wspierających różnorodne oprogramowania.

Praca finansowana z projektu INDECT, w ramach VI Programu UE.

Literatura

[1] Armendariz J.E., Decker H., Munoz-Escoi F.D., Irun-Briz L., de Juan-Marin R., *A middleware architecture for supporting adaptable replication of enterprise application data Design*. Springer, 2005.

[2] Bernstein P.A., Hadzilacos V., Goodman N., *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[3] Bougettaya A., Malik Z., Rezgui A., Korff L., *A Scalable Middleware for Web Databases*. Codd & Date, Inc., 2006.

[4] Carey M.J., Livny M., *Conflict detection tradeoffs for replicated data*. ACM Trans. Database Syst., 1991.

[5] Coulon C., Pacitti E., Valduriez P., *Consistency management for partial replication in a high performance database cluster*. ICPADS, 2005.

[6] Defago X., Schiper A., Urban P., *Total order broadcast and multicast algorithms: Taxonomy and survey*. 2004.
[7] Gray J., Helland P., O'Neil P., *The dangers of replication and a solution*. ACM SIGMOD, 1996.
[8] Kasper M., *Middleware based replication for database systems*. Automatyka (półrocznik AGH), 2008.
[9] Lin Y., Kemme B., Patino-Martinez M., *Middleware based data replication providing snapshot isolation*. ACM Press, 2005.
[10] Saito Y., Shapiro M., *Optimistic replication*. ACM Computing Surveys, 2005.
[11] Wiesmann M., *Group communications and database replication: techniques, issues and performance*. PhD thesis, Ecole Polytechnique Federale De Lausanne, 2002.