

Kamil Kuliberda^{*}, Jacek Wiślicki^{*}

Architektura gridu bazodanowego – *Data Grid*

1. Wprowadzenie

Data Grid – grid bazodanowy, jest koncepcją, dzięki której możliwe jest zrealizowanie równoległego, rozproszonego przetwarzania informacji biznesowych pochodzących ze źródeł o charakterze bazodanowym. Oznacza to możliwość połączenia ze sobą niezależnych źródeł danych posiadających swój własny, niezależny schemat formatowania danych. Aby możliwa była wymiana danych pomiędzy organizacjami udostępniającymi swoje dane, należy wykonać formalizację udostępnionych danych, według wcześniej ustalonego globalnego schematu przetwarzanych danych. Schemat taki opisuje charakter i zależności danych dostępnych w gridzie. Jednym ze strategicznych zadań omawianego podejścia jest poprawne zdefiniowanie przezroczystości, dzięki której możliwy jest uproszczony dostęp do informacji. Precyzyjne określenie zasad przezroczystości i ich implementacja jest trudnym zadaniem dla twórców oprogramowania. Zaproponowane w niniejszym artykule rozwiązanie bazuje na koncepcji Aktualizowalnych Perspektyw oraz Teorii Podejścia Stosowego do obiektowych baz danych. Wymienione mechanizmy w nieskomplikowany sposób pozwalają na realizację problemów związanych z przezroczystością dla zbiorów informacji pochodzących z baz danych [1, 4]. Artykuł porusza tylko tematykę baz danych, odpowiednie zastosowanie opisywanego podejścia wzbogacone dodatkowymi usługami i mechanizmami może posłużyć do rozwiązywania innych problemów biznesowych w architekturze rozproszonej np. rozproszonych obliczeń.

2. Szczegóły koncepcji *Data Grid*

Data grid koncentruje się na przetwarzaniu danych udostępnianych w sposób dynamiczny przez współużytkowników sieci grid. Dostęp do zasobów globalnych jest realizowany poprzez mechanizm Wirtualnego Repozytorium, który spełnia także rolę interfejsu integrującego i zarządzającego wszystkimi rozproszonymi zasobami. Jego zadaniem jest również uproszczenie dostępu do rozproszonych, heterogenicznych i nadmiarowych danych, co umożliwia łatwiejsze zarządzanie dostępnością zasobów. Cykliczne aktualizacje

^{*} Katedra Informatyki Stosowanej, Politechnika Łódzka

stanu schematu globalnego wirtualnego repozytorium pozwalają na wyeliminowanie fizycznie niedostępnych zasobów. Komunikacja między współużytkownikami jest realizowana za pomocą odpowiednio zaprojektowanej platformy transportowej szczegółowo opisanej w artykułach [9, 10]. Prezentowana koncepcja zakłada składowanie danych na lokalnych komputerach podłączonych do sieci grid, jednocześnie spełnia założenie przezroczystości, dzięki któremu zarówno dostępne usługi, jak i dane są zarządzane i aktualizowane lokalnie i niezależnie. Po stronie schematu globalnego nie występuje ani ich materializacja, ani kopie.

Dostęp do danych, ich wyszukiwanie oraz ich propagacja odbywają się dzięki dodatkowym mechanizmom wbudowanym w wirtualne repozytorium. Mechanizmy działają zgodnie z teorią podejścia stosowego omówionego w artykułach [1, 5].

Zaimplementowany prototyp obiektowej bazy danych z mechanizmem stosów oraz mechanizmem aktualizowalnych perspektyw [1] pozwala na zbudowanie omawianej koncepcji środowiska grid.

Obecnie istnieją dwa prototypy obiektowych baz danych;

- 1) YAOD (implementacja w języku JAVA),
- 2) ODRA (implementacja w języku MS C#).

W wirtualne repozytorium wbudowana jest również metabaza określająca zależności oraz klasyfikację danych w podstawowym modelu składu obiektów, który powstaje w fazie projektowania sieci grid. Do budowy metabazy można wykorzystać wszystkie mechanizmy znane z innych technologii bazodanowych, które są zaimplementowane w wirtualne repozytorium [3, 4, 8].

Głównym problemem podczas prac nad budową systemów gridowych jest wkomponowanie do globalnego schematu i wzajemne zestawienie globalnych klientów oraz dostawców zasobów. Należy zapewnić swobodne, obustronne przetwarzanie danych pomiędzy współuczestnikami gridu – odczyt danych oraz ich pełna aktualizacja u dostawców. Nasza koncepcja definiuje osłony (*wrappers*) jako mechanizmy pozwalające na taką realizację przetwarzania danych. Ich działanie ogranicza się do sterowania i wykorzystywania innych usług (np. relacyjne bazy danych, usługi XML). Osłony pełnią rolę pomostu pomiędzy oryginalną strukturą udostępnianych informacji a mechanizmami nazwanymi perspektywami (*views*), dzięki którym realizowany jest proces formalizowania oryginalnych danych do struktury określonej przez schemat globalny. Wynikiem tego procesu jest schemat kontrybucyjny danych, który integruje do schematu globalnego. Schemat kontrybucyjny jest zbiorem funkcji języka zapytań, które realizują wiązanie pomiędzy sformalizowaną reprezentacją danych a oryginalnymi danymi uzyskanymi z lokalnych zasobów. Przy takich założeniach schemat kontrybucyjny jest wirtualną, niematerializowaną kolekcją danych. Jest to kolekcja informacji biznesowych, które mogą podlegać fragmentacji, charakteryzujących się bezpośrednim dostępem do źródła danych poprzez odpowiednie mechanizmy.

W naszej koncepcji język zapytań jest mechanizmem najwyższego poziomu, dzięki któremu realizujemy dostęp do informacji biznesowych sformalizowanych globalnym schematem wirtualnego repozytorium. Operacje przetwarzania danych dla całego gridu definiowane są językiem zapytań (w naszym przypadku SBQL) [8].

Metadane reprezentują centralny mechanizm architektury integrującej wszystkie dane przetwarzane w gridzie – są kluczowym elementem wirtualnego repozytorium. Ukrywają

wszystkie fizyczne formy współpracy poszczególnych mechanizmów potrzebnych do budowy i zarządzania globalnego indeksu danych w gridzie. W naszym przypadku jest to ujednoczenie struktur danych pochodzących z różnych źródeł i usług, np.: tabel, danych z relacyjnych baz danych, obiektów z innych obiektowych baz danych, atrybutów, dokumentów XML, dokumentów HTML itp. Metadane przechowują wszystkie informacje o szczególnych cechach dostawców informacji oraz samych danych. Integracja metadanych z wirtualnym repozytorium ujednocza schemat przetwarzania danych, co zwiększa udogodnienia w wykorzystaniu oprogramowania i daje możliwość współużytkowania całkowicie różnych modeli danych.

Wirtualne repozytorium może być zaimplementowane za pomocą obiektowej bazy danych, w pełni generycznych aktualizowalnych perspektyw oraz specjalizowanych rozwiązań softwarowych zawierających osłony.

W zależności od rodzaju zasobów można wykorzystać następujące technologie jako źródła danych [3]:

- Web Serwisy jako transport danych i udogodnienie wymiany informacji;
- JDBC jako osłona dla tabel i relacyjnych baz danych z dostępem poprzez język Java;
- ODBC, DCOM jak wyżej, dostęp jest realizowany dzięki technologiom firmy Microsoft;
- różne technologie składowania danych opierające się na schematach XML;
- ogólnie dostępne rozwiązania na bazie języków programowania i języków zapytań.

Omawiana koncepcja zapewnia bezpieczeństwo, prywatność, licencjonowanie oraz działanie zgodnie z przyjętymi przy budowie sieci grid certyfikatami. Część tych aspektów jest realizowane przez wirtualne repozytorium, część przez platformę transportową [9, 10], oraz globalne aplikacje dostępne.

3. Zagadnienie przezroczystości (*transparency*)

Istotnym problemem przy budowie systemów rozproszonych jest zapewnienie przezroczystości danych, zasobów, usług itd. Przezroczystość pozwala na ukrycie wszystkich niesformalizowanych oryginalnych danych za wirtualnym schematem globalnym, dostępnym dla całego gridu. W naszym wypadku „wirtualność” jest powiązana z przetwarzaniem danych, co oznacza reprezentacje danych zgodnie z definicją języka zapytań. Takie podejście oznacza brak materializacji danych na podstawie zapytania, w zamian za to występuje powiązanie do danych oryginalnych. Termin wirtualny jest tutaj bezpośrednio związany z terminem przezroczysty. Przezroczystość rozwiązuje wiele skomplikowanych szczegółów technicznych dotyczących środowiska rozproszonych danych i usług. Przede wszystkim podnosi elastyczność zarządzania i utrzymania gridu. Dzięki przezroczystości można w uproszczony sposób realizować połączenia pomiędzy klientami a dostawcami danych, wiedząc jednocześnie, że każdy z nich może w łatwy sposób wykorzystywać globalny potencjał przetwarzania danych. Wyższe formy przezroczystości takie, jak: fragmentacja danych, nadmiarowość, kopie czy różnorodne dostarczanie usług tworzą nową kulturę w wymianie informacji, gdzie użytkownik gridu otrzymuje wiele korzyści i zdobywa wiele nowych doświadczeń związanych z przetwarzaniem wirtualnych danych [6, 7].

Przezroczystość posiada wiele różnych form, w skrócie przedstawimy tylko te najważniejsze, które mają istotne znaczenie w proponowanym przez nas rozwiązaniu. Jednoczesne osiągnięcie wszystkich form przezroczystości jest dość trudne i stanowi wielkie wyzwanie dla projektantów oprogramowania.

Poniżej przedstawione zostały podstawowe formy przezroczystości:

- Przezroczystość nazw danych i usług – zdolność identyfikacji określonych danych bez potrzeby rozpoznawania ich prawdziwych nazw. Dane wirtualne są dostępne dzięki mechanizmom perspektyw, które określają ich nazwy synonimowe. Istnieje możliwość uzyskania całych kolekcji danych. Ten rodzaj przezroczystości dotyczy także nazewnictwa klientów oraz dostawców danych.
- Przezroczystość lokalizacji danych i usług – opisuje zależności pomiędzy lokalizacją fizycznych zasobów danych a wirtualną definicją lokalizacji, dostępną dla współuczestników gridu. Ta przezroczystość daje możliwość dostępu do danych bez wiedzy o ich fizycznej lokalizacji.
- Przezroczystość dostępu do danych i usług – daje możliwość dostępu do źródeł danych poprzez różne metody dostępu oraz różne interfejsy, może definiować także niektóre metody określające prawa dostępu.
- Przezroczystość współbieżności – trudny do zrealizowania mechanizm pozwalający na poprawne rozproszone, równoległe przetwarzanie danych w zależności od złożoności funkcji przetwarzania.
- Przezroczystość implementacji danych i usług – ukrywa zależności pomiędzy strukturami danych sformalizowanymi dla wirtualnego repozytorium a strukturami danych występującymi u dostawców danych i usług.
- Przezroczystość skalowalności – pozwala na nieograniczony wzrost i redukcję zasobów wirtualnego repozytorium. Jest to możliwe do osiągnięcia dzięki mechanizmowi kontrybucji gridu, który pozwala na generyczną przebudowę i aktualizację istniejącego już schematu gridu.
- Przezroczystość fragmentacji poziomej i pionowej – jedna z najważniejszych form przezroczystości dotycząca zasobów lokalnych gridu. Pozwala na przetwarzanie i łączenie pofragmentowanych obiektów w kolekcje oraz ich prezentację jako połączone obiekty. Takie podejście pozwala na dostęp do określonych fragmentów kolekcji, przy uwzględnieniu narzuconych ograniczeń. Możliwe jest przetwarzanie każdej dostępnej partii danych pofragmentowanych bez ingerowania w ich inną część.
- Przezroczystość nadmiarowości danych i usług – rozwiązuje podobny problem jak przezroczystość fragmentacji, ponadto pozwala na poprawne przetwarzanie i dostęp do zasobów danych zawierających powtarzające się nadmiarowe dane względem schematu globalnego. Ważną cechą realizacji tej przezroczystości jest odpowiednia reakcja na zmiany w obiektach, które muszą być uwzględnione dla każdej lokalizacji zawierającej takie same nadmiarowe dane.
- Przezroczystość kopii danych i usług – rozwiązuje problem podobny do powyższego, jednak przetwarzanie dotyczy zasobów zawierających takie same dane z identyczną strukturą obiektów.

- Przezroczystość lokalnych indeksów danych i usług – bardzo trudna do uzyskania w praktyce, dzięki szczególnym przypadkom, jakimi są mechanizmy indeksowania danych, które występują w poszczególnych usługach i źródłach danych. Transakcje na obiektach mogą wykorzystywać indeksy. Wirtualne repozytorium, które jest zbiorem danych wirtualnych, wykorzystuje indeksy zbudowane właśnie na takich danych, stąd dane wirtualne oraz ich odwzorowanie na dane rzeczywiste niekoniecznie może być takie same. Nieodpowiednie zaprojektowanie tej przezroczystości może doprowadzić do fałszywych rezultatów obliczeń.
- Przezroczystość dostawców danych i usług – istnieje możliwość wystąpienia wielu dostawców danych udostępniających takie same dane, ale o innej strukturze. Ten rodzaj przezroczystości zapewnia poprawne przetwarzanie danych na wszystkich zasobach dostępnych dzięki wirtualnemu repozytorium, nawet jeśli struktura danych rzeczywistych ma inną postać u każdego z dostawców tych danych [6].

4. Architektura wirtualnego repozytorium

Przetwarzanie danych w oparciu o wirtualne repozytorium napotyka kilka skomplikowanych problemów. Jednym z nich jest aktualizacja wirtualnych danych widzianych przez wirtualne repozytorium. Stan badań tego problemu jest wciąż otwarty i praktycznie niezbadany. Podobne problemy dotyczą bezpieczeństwa i infrastruktury transakcji, które znajdują się w wyższej warstwie architektury ponad wirtualnym repozytorium. Inny problem dotyczy zagadnień wydajności, a w szczególności – globalnej optymalizacji zapytań, które są wynikiem interakcji użytkowników z systemem.

Skład komponentów wirtualnego repozytorium przedstawia rysunek 1.



Rys. 1. Abstrakcyjna architektura wirtualnego repozytorium

Proponowany skład komponentów wirtualnego repozytorium (rys. 1) przedstawiony jest poniżej.

- Globalna infrastruktura zabezpieczeń i zarządzania repozytorium znajdująca się na wierzchołku wirtualnego repozytorium jest odpowiedzialna za zarządzanie wszystkimi zasobami gridu i przyznawaniem uprawnień dla współuczestników gridu.
- Mechanizm globalnego przetwarzania transakcji jest odpowiedzialny za synchronizowanie współbieżnego dostępu do rozproszonych zasobów.
- Mechanizm globalnej integracji zasobów ma za zadanie integrację pofragmentowanych obiektów w jedną spójną wirtualną całość, pozwala na swobodne przetwarzanie zasobów heterogenicznych, dowiązuje do wirtualnego schematu globalnego zasoby nadmiarowe oraz kopie, stosuje wszystkie dostępne mechanizmy do osiągnięcia jak najwyższego poziomu przezroczystości gridu.
- Zarządzanie metadanymi, przechowuje globalny oraz lokalne schematy danych oraz współzależności występujące pomiędzy zasobami lokalnymi oraz globalnymi, ze szczególnym uwzględnieniem specyficznych różnic pomiędzy zasobami lokalnymi.
- Zarządzanie ontologią zawiera wszystkie dane dotyczące zorientowanymi na użytkownika metazasobami wiedzy, które dotyczą klasyfikacji obiektów, ich kategoryzacji, reprezentują słowniki, mapy tematyczne i bazy wiedzy. Metazasoby są bezpośrednio powiązane z zasobami, które pozwalają użytkownikom na ich wyszukiwanie i przetworzenie zgodnie pewnymi koncepcyjnymi wzorcami.
- Zarządzanie wsparciem dla dostępu składa się z mechanizmów optymalizujących zapytania i indeksy, które charakteryzują się wsparciem dla zwiększenia wydajności gridu, zwiększenia jego dostępności oraz skalowalności.

Zaprezentowane rozwiązanie jest podejściem ogólnym i oryginalnym. Kilka opisanych wcześniej mechanizmów wspomagających działanie aplikacji gridowych i webowych jest już zaimplementowanych w obecnie rozwijanej platformie prototypowej ODRA.

Platforma ta jest naszym głównym mechanizmem w rozwoju prezentowanej koncepcji:

- Rozwijania potężnego obiektowego modelu bazy danych, cechującego się możliwością ukrycia wszystkich aspektów występujących podczas integracji heterogenicznych zasobów zawierających różne formaty danych; model będzie dostępny jako język służący do budowy schematu globalnego oraz schematów lokalnych współdzielonych zasobów.
- Rozwijania potężnego obiektowego języka zapytań jako interfejsu dla przetwarzania zasobów.
- Rozwijania mechanizmu obiektowych, aktualizowalnych perspektyw, które posłużą do rozwiązywania wielu procesów: do globalnej integracji zasobów, do organizacji schematów lokalnych na lokalnych serwerach, jako interfejsy najniższego poziomu do nadawania uprawnień użytkownikom, jako dodatkowy mechanizm bezpieczeństwa, bazujący na przeciążeniu generycznych operacji dla obiektów wirtualnych.
- Rozwijania protokołu komunikacyjnego z platformą transportową, który umożliwi fizyczną integrację klientów i dostawców danych z wirtualnym repozytorium.
- Rozwijania metodologii, która umożliwi utrzymać dyscyplinę procesów tworzenia konkretnego wirtualnego repozytorium dla wymaganych celów biznesowych, poprzez dobrze zdefiniowane, mierzalne i możliwe do skontrolowania mechanizmy.

Pomimo to, że literatura zawiera wiele informacji na temat opisywanych tutaj zagadnień i problemów, których zaawansowanie jest niewielkie i dalekie od jednolitego i uniwersalnego rozwiązania, nasze badania (bazujące na generycznym modelu obiektowej bazy danych oraz aktualizowalnych perspektywach) dotyczące podejść do unifikacji języków zapytań i aktualizowalnych perspektyw oraz sprowadzenia ich do konstrukcji poprawnie zdefiniowanych mechanizmów gridowych stwarza dużą szansę na otrzymanie ważnych teoretycznych i praktycznych wyników, które znajdą swoje miejsce wśród innych badań tego typu.

5. Aspekty techniczne realizacji koncepcji *Data Grid*

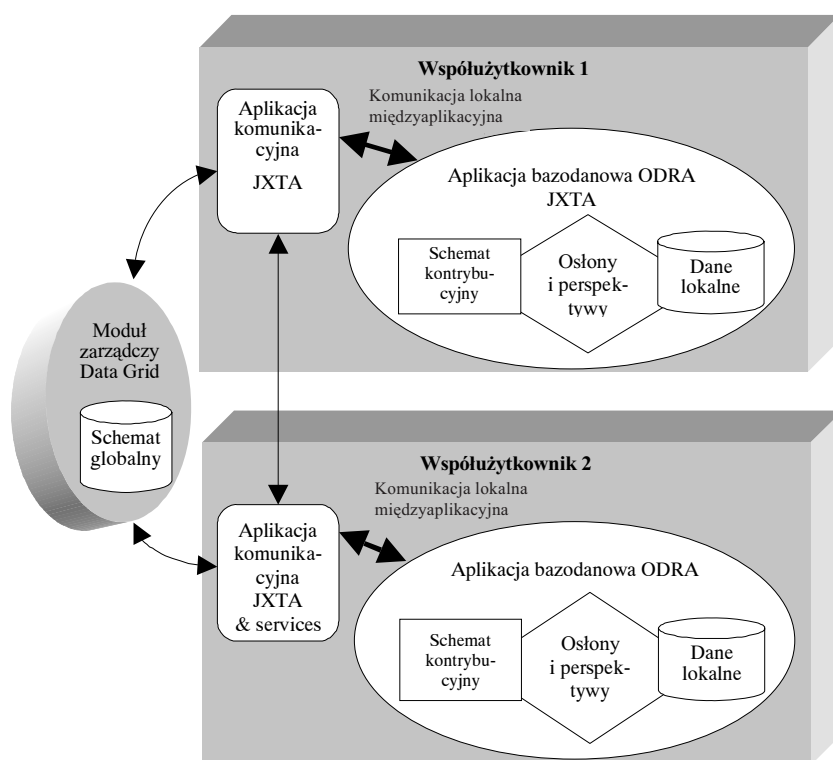
Techniczne aspekty realizacji opisywanej koncepcji zakładają współpracę kilku technologii. Głównym aspektem jest prawidłowa konstrukcja wirtualnego repozytorium zgodnie z dwoma pozostałymi odpowiedzialnymi za zarządzanie zasobami danych i usług (osłony) oraz komunikacją i wymianą danych wraz z zagadnieniami bezpieczeństwa i zarządzaniem komunikacją (platforma transportowa). Rozwiązanie opiera się na współlistnieniu odseparowanych aplikacji (modułów) współdziałających ze sobą za pomocą wewnętrznych mechanizmów pozwalających na ograniczony i odpowiednio określony zakres przetwarzania danych. Poszczególne moduły mogą być rozwijane i implementowane zgodnie z ustaleniami projektantów systemu gridowego.

Nasza propozycja zakłada niżej wymienione moduły.

- Wirtualne repozytorium – fizycznie dostępne poprzez aplikacje o wcześniej opisanej architekturze. Częścią repozytorium będzie aplikacja klienta i dostawcy usług i danych oraz aplikacja zarządzająca gridem.
- Osłony integracyjne – to mechanizmy pozwalające na prosty dostęp do lokalnych zasobów gridu, które swoje schematy lokalne kontrybuują do wirtualnego repozytorium. Każdy z klientów gridu i dostawców zasobów musi być zaopatrzony w osłony zgodnie ze swoim wyborem lub zgodnie z regułami danej sieci grid. Osłony mają bezpośredni dostęp do interfejsów usług udostępnianych przez dostawców informacji. Interfejsy gwarantują dostęp do kolekcji danych, które dany dostawca udostępnia w gridzie. Lokalne dane mogą być zarządzane zgodnie z możliwościami na jakie pozwalają udostępnione przez dostawców interfejsy [4].
- Platforma transportowa – określa niezależne środowisko aplikacyjne odpowiedzialne za swobodne, rozproszone, współbieżne przetwarzanie transakcji gridowych. Platforma w szczególności powinna gwarantować nieograniczony fizycznie dostęp do sieci grid dla współużytkowników sieci (klientów oraz dostawców) oraz zapewnić bezpieczną i niezawodną komunikację oraz wymianę informacji. Rozwiązanie bazuje na architekturze sieci każdy-z-każdym (*Peer-to-Peer*) zarządzanej centralnie. Takie rozwiązanie powinno zapewnić między innymi następujące operacje: unikalną identyfikację współużytkowników sieci, nazewnictwo współużytkowników, komunikację

między współużytkownikami, bezpieczeństwo połączeń itp. Innymi ważnymi aspektami jest utrzymanie przezroczystości lokalizacji zasobów dla współdziałających użytkowników, zapewnienie skalowalności sieci, zapewnienie niezależności od fizycznej konfiguracji sieci i adresacji oraz nazewnictwa. Wszystkie powyższe aspekty mogą być rozwijane dzięki, wieloprotokołowej, w pełni programowalnej platformie bazującej na technologii Project JXTA [11].

Poniżej omówioną architekturę współpracy modułów przedstawia rysunek 2.



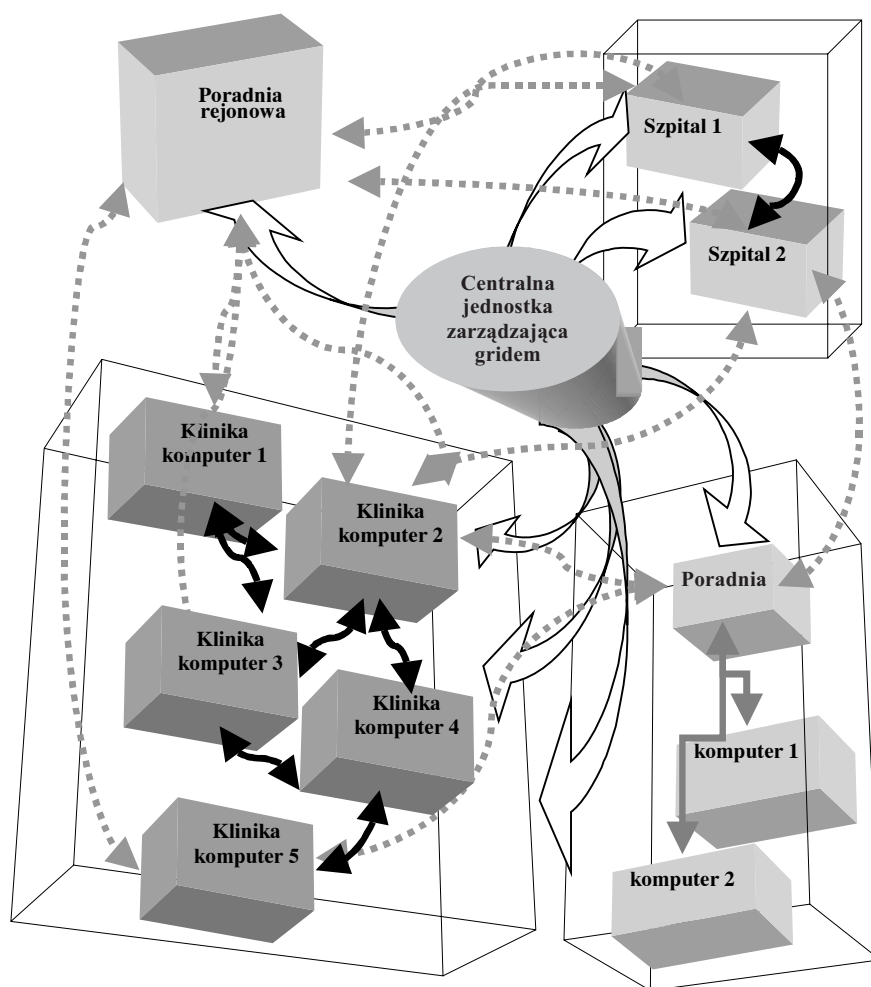
Rys. 2. Architektura logiczna koncepcji *Data Grid*

6. Prosty przykład gridu bazodanowego

Poniższy przykład przedstawia biznesowe podejście do problemu rozproszonego przetwarzania danych. Na rysunku 3 przedstawiono powiązania w aplikacji gridu bazodanowego dla przykładu usług medycznych. W rzeczywistości istnieje wiele różnych ośrodków zdrowia o różnym charakterze świadczonych usług, wzięliśmy pod uwagę przychodnie rejonowe, poradnie, szpitale, kliniki.

Globalna sieć zdrowia może być utworzona poprzez połączenie w jedną sieć grid oddzielnie działających ośrodków zawierających zasoby danych takich, jak m.in.:

- listy pacjentów,
- historie ich chorób,
- listy używanych przez nich leków.



Rys. 3. Przykład architektury *Data Grid*

Każdy z wymienionych ośrodków dysponuje danymi związanymi z dziedziną zdrowia, w jakiej się specjalizuje. Głównym zadaniem jest tutaj połączenie ze sobą odseparowanych danych różnego typu w jeden globalny schemat dostępny dla wszystkich aktywnych współużytkowników takiej sieci.

Z technicznego punktu widzenia każdy z komputerów występujących w poszczególnych ośrodkach przy połączeniu do gridu staje się współużytkownikiem gridu, automatycznie staje się częścią globalnego schematu. Centra zdrowia mogą posiadać podsieci, wtedy każda z podsieci może być w sieci grid widziana jako oddzielny współużytkownik sieci. W prezentowanym podejściu, każdy komputer podłączony niezależnie do sieci grid staje się częścią jego schematu globalnego. W takim przypadku należy zrealizować proces formalizacji danych dla każdego lokalnego schematu. Przedstawiony przykład jest elastyczny pod względem badań, mogą w nim pojawić się wszystkie omawiane wcześniej złożoności schematów lokalnych takie jak fragmentacje, nadmiarowości czy kopie.

7. Podsumowanie

W artykule przedstawione zostało podejście do modelu gridu bazodanowego, opierającego się o teorię podejścia stosowego do baz danych. Naszkicowano główne pomysły zbudowania modelu rozproszonego obiektowych baz danych przetwarzającej równolegle transakcje w architekturze rozproszonej.

Model opiera się o mechanizmy takie, jak:

- wirtualne repozytorium,
- osłony,
- aktualizowalne perspektywy,
- platformę transportową wspieraną architekturą P2P.

Obecny stan prac nad proponowanym w artykule rozwiązaniem koncentruje się nad opracowaniem spójnej koncepcji budowy mechanizmów osłon dla różnych technologii przetwarzania danych oraz częściowej implementacji prototypów omawianych w artykule modułów oprogramowania.

Literatura

- [1] Kozakiewicz H., Leszczyłowski J., Płodzień J., Subieta K.: *Updateable Object Views*. ICS PAS Reports 950, October 2002
- [2] Habela P., Kaczmarek K., Kozankiewicz H., Lentner M., Stencel K., Subieta K.: *Data-Intensive Grid Computing Based on Updatable Views*. ICS PAS Report 974, June 2004
- [3] Kaczmarek K., Habela P., Subieta K.: *Metadata in a Data Grid Construction*. Workshop on Emerging Technologies for Next generation GRID (ETNGRID-2004), 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), University of Modena and Reggio Emilia, Italy, June 14-16, 2004, Proceedings published by IEEE
- [4] Kozankiewicz H., Stencel K., Subieta K.: *Integration of Heterogeneous Resources through Updatable Views*. Workshop on Emerging Technologies for Next generation GRID (ETNGRID-2004), 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), University of Modena and Reggio Emilia, Italy, June 14-16, 2004, Proceedings published by IEEE

-
- [5] Subieta K.: *Teoria i konstrukcja obiektowych języków zapytań*. Warszawa, Wydawnictwo PJWSTK 2004, ISBN 83-89244-28-4 (522 s.)
 - [6] Moore R., Merzky A.: *Persistent Archive Concepts*. Global Grid Forum GFD-I.026. December 2003
 - [7] Foster I., Kesselman C., Nick J., Tuecke S.: *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Global Grid Forum, June 22, 2002
 - [8] Kozankiewicz H., Stencel K., Subieta K.: *Implementation of Federated Databases through Updatable Views*. Proc. EGC 2005 – European Grid Conference, Springer LNCS, 2005
 - [9] Kuliberda K.: *Podstawowe mechanizmy architektury Grid Computing*. Półrocznik AGH Automatyka, t. 8, z. 3, 2004, 497–503
 - [10] Kuliberda K., Sekulska-Nalewajko J., Kuzański M.: *Technologia Grid Computing – Nowa architektura sieci komputerowych*. XII Konferencja Sieci i Systemy Informatyczne 2004, Politechnika Łódzka, Łódź, 2004
 - [11] www.jxta.org

