STUDIA INFORMATICA 2001
Volume 22 Number 3 (45)

Przemysław GŁOMB
Instytut Informatyki Teoretycznej i Stosowanej PAN

NARZĘDZIE ADMINISTRACYJNE DLA SYSTEMU ORACLE VIDEO SERVER ZAPROJEKTOWANE Z UWZGLĘDNIENIEM WARUNKÓW OBCIĄŻENIA

Streszczenie. Praca przedstawia opis narzędzia administracyjnego dla serwera Oracle Video Server, stworzonego z wykorzystaniem sieci przekazywania zgłoszeń Oracle Media Net, uwzględniającego charakterystyczne dla tego modelu i profilu zastosowania, zagadnienia efektywności i niezawodności.

AN ADMINISTRATOR'S TOOL FOR ORACLE VIDEO SERVER WRITTEN WITH CONSIDERATION OF STRESSFUL CONDITIONS

Summary. This work describes a tool for administering the content of Oracle Video Server. It uses Oracle Media Net, and considers topics like effectiveness and failure avoidance.

1. Geneza i cel pracy

Geneza pracy wywodzi się z badań prowadzonych w laboratorium multimedialnym Instytutu Informatyki Teoretycznej i Stosowanej PAN. Ważną rolę w nich odgrywa serwer usługi „wideo na żądanie" (ang. *Video on Demmand*) Oracle Video Server. Jest on szeroko wykorzystywany, od badań wymagań transmisji filmowych w sieciach komputerowych, przez analizę właściwości strumieni multimedialnych, do prób ich klasyfikacji i indeksacji. Aby jednak te ostatnie doświadczenia mogły być skutecznie prowadzone, niezbędne było uzupełnienie zestawu narzędzi administracyjnych serwera o moduł dokonujący analizy treści zawartych w bazie filmów.

Opisywane narzędzie usuwa częściowo jedno z najpoważniejszych niedociągnięć systemu OVS, jakim jest brak możliwości indeksacji strumieni multimedialnych. Podczas

uzupełniania zawartości bazy filmów, wszelkie zmiany trzeba później odnotować w bazie opisu ręcznie. Wiąże się to często z koniecznością oglądania całego filmu, np. w celu przydzielenia go do jakieś kategorii. Aby uniknąć takiej straty czasu, zaprojektowano mechanizm automatycznej indeksacji, prezentowany w [5].

Jego praktyczna realizacja wymagała stworzenia nakładki dla serwera, wykorzystującej środowisko sieciowe Oracle Media Net (wywodzące się z modelu CORBA, ang. *COmmon Request Broker Architecture*), uwzględniającej ostre wymagania co do kontroli błędów i obciążania serwera.

Ta praca przedstawia opis wykonania tego rozszerzenia. W następnych rozdziałach prezentowane są kolejno: architektura środowiska Oracle Video Server i Oracle Media Net; lista problemów, z którymi przyszło się uporać; sposób ich rozwiązania; wreszcie opis architektury samego programu, schemat działania i wnioski z jego użytkowania.

2. Opis architektury serwera

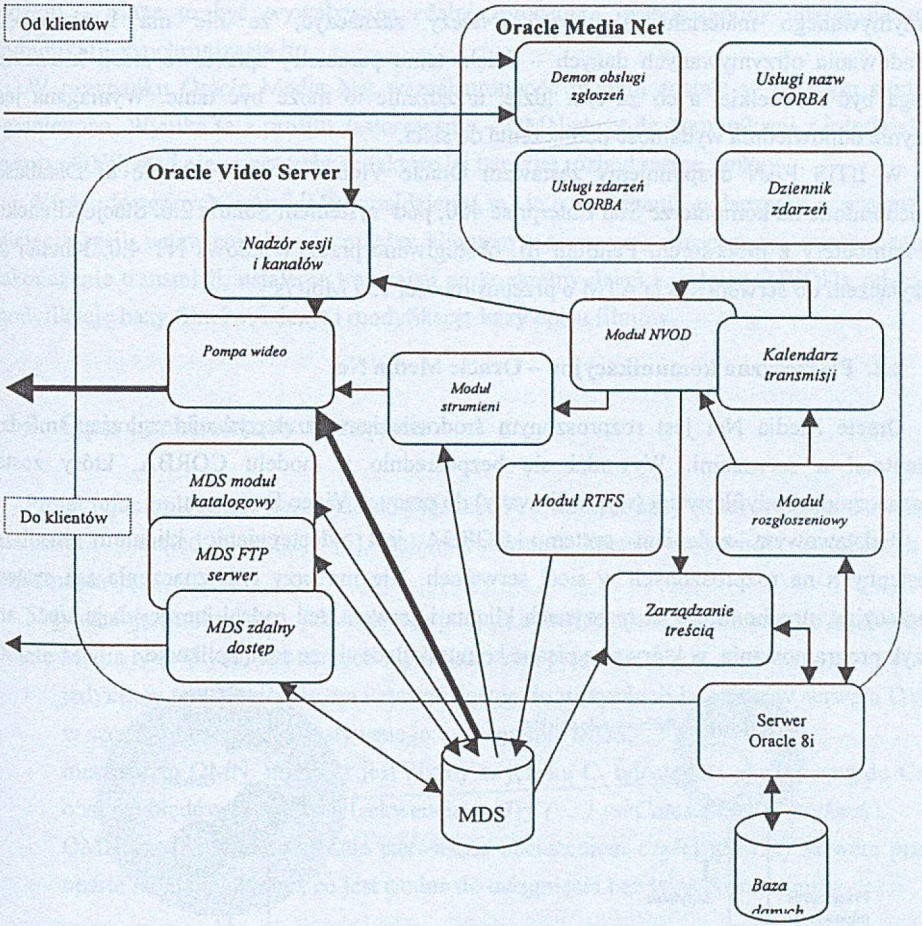
2.1. System Oracle Video Server

System Oracle Video Server jest serwerem multimedialnej bazy danych, przystosowanym do oferowania usług typu „wideo na żądanie" oraz pokrewnych, w tym [9]:

- szeroko rozumianych zdalnych prezentacji, obejmujących np. zdalne nauczanie, transmisję filmów, programów telewizyjnych na żądanie,
- obsługi cyfrowego archiwum informacji multimedialnej,
- przekazywania na żywo do sieci programów, np. transmisji sportowych.

Architektura serwera składa się z następujących elementów [1]:

- obszar przechowywanych danych (MDS, ang. *media data store*), zawierający bazę danych informacji multimedialnej, dostępnej dla klientów, wraz z modułami administracyjnymi – usługą katalogową, modułem zdalnego dostępu oraz serwerem FTP umożliwiającym manipulację jego zawartością. Obszar przechowywania danych może się znajdować na innym komputerze niż sam serwer,
- „pompę wideo" (ang. *video pump*), odpowiedzialną za szybkie przekazywanie danych z dysku w sieć,
- moduł sterowania przebiegiem sesji, przyjmowaniem zgłoszeń od klientów, sterowania przebiegiem transmisji, tworzenia kanałów klient – serwer, obsługi informacji zwrotnej (ang. *stream, session and circuit service*),



Rys.1. Schemat Video Servera
Fig. 1. Video Server's schema

- moduł zarządzania transmisją zleconą (NVOD, ang. *near video on demmand*), polegającą na ustawieniu transmisji danego materiału filmowego na określoną datę i godzinę,
- moduł obsługi transmisji, przekazywanej ze źródła zewnętrznego w czasie rzeczywistym (RTFS, ang. *real time feed service*),
- moduł zarządzający zawartością (ang. *logical content service*), umożliwiający przechowywanie informacji o filmach zgromadzonych w bazie danych. Moduł ten może współpracować z bazą danych Oracle.

OVS jest zbudowany w architekturze klient – serwer. Klient, instalowany na stacjach roboczych, odpowiedzialny jest wyłącznie za nawiązanie połączenia i odtwarzanie na bieżąco

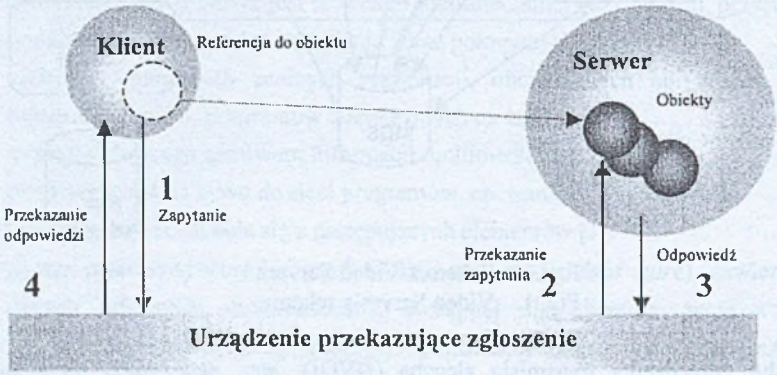
otrzymywanego materiału filmowego. Należy zaznaczyć, że nie ma konieczności składowania otrzymywanych danych – dzięki temu parametry sprzętowe stacji klienckiej mogą być niewielkie, a co za tym idzie, urządzenie to może być tanie. Wymagana jest jedynie odpowiednia wydajność podłączenia do sieci.

W IITiS PAN dysponujemy zestawem Oracle Video Server + Oracle 8i Database, uruchomione na komputerze Sun Enterprise 450, pod systemem Solaris 2.6. Stacje klienckie to komputery z procesorem Pentium III, obsługiwane przez Windows NT 4.0. Klienci są przyłączeni do serwera siecią ATM o przepustowości 155 Mbit /s.

2.2. Płaszczyzna komunikacyjna – Oracle Media Net

Oracle Media Net jest rozproszonym środowiskiem przekazywania zgłoszeń między klientami a serwerami. Wywodzi się bezpośrednio z modelu CORBA, który został nieznacznie zmodyfikowany (wyprofilowany) do pracy z Video Serverem.

Podstawowym zadaniem systemu CORBA jest udostępnianie klientom zasobów dostępnych na rozproszonych w sieci serwerach. Nie ma przy tym znaczenia ani system operacyjny, uruchomiony na maszynach klienta i serwera, ani rodzaj łączących je sieci, ani język programowania, w którym są pisane komunikujące się ze sobą aplikacje.



Rys.2. Idea architektury CORBA
Fig. 2. CORBA architecture idea

Realizowane jest to z udziałem urządzenia przekazującego zgłoszenie (ORB, *object request broker*). Udostępnia ono klientom referencje do zdalnych obiektów, a w przypadku żądania wykonania operacji na danym obiekcie, odnajduje odpowiednie jego wystąpienie, przekazuje do właściwego serwera zgłoszenie / żądanie, a następnie przekazuje rezultat klientowi (rysunek 2).

Urządzenia przekazujące dysponują rozbudowanymi algorytmami równoważenia obciążeń, wyboru serwerów które są w danej chwili dostępne. Obiekty mogą być różnego

rodzaju – może to być wywoływana zdalnie procedura, współdzielony obszar danych, komunikaty, synchronizacja itp.

W przypadku Oracle Media Net wszechstronność modelu została w pewnym stopniu ograniczona. Wynika to z profilu zastosowania – OMN służy do komunikacji z interfejsami serwera OVS, stąd nie są potrzebne niektóre jej bardziej rozbudowane funkcje.

Wśród dostępnych interfejsów znajdziemy m. in.: nawiązanie połączenia z serwerem, inicjacja sesji, ustawienie kanału między klientem a serwerem, rozpoczęcie, wstrzymanie i zakończenie transmisji, ustalenie transmisji na konkretny dzień i godzinę (NVOD), odczyt i modyfikację bazy filmów, odczyt i modyfikację bazy opisu filmów.

3. Opis architektury narzędzia

Projektując opisywane narzędzie do indeksacji zawartości bazy filmów, po opracowaniu analizatora i metod ekstrakcji [5], należy rozwiązać szereg problemów związanych z implementacją.

Najważniejszym krokiem, jaki należało podjąć, była decyzja o wykorzystaniu środowiska Oracle Media Net. Wybór ten narzucił się niejako sam przez się, gdyż:

- jedynie w ten sposób można uzyskać dostęp do wszystkich interfejsów serwera OVS, w szczególności wykorzystywanego w programie MDS,
- mechanizm OMN, mimo iż jest pisany w języku C, udostępnia analogiczną do C++ obsługę błędów i wyjątków (sekwencja *yseTry { ... } yseCatchAll { ... } yseEnd;*),
- OMN zapewnia odpowiednie sterowanie obciążeniem części głównej serwera przez oparte na niej aplikacje, co jest trudne do osiągnięcia bez jej wykorzystania.

3.1. Założenia, które należało uwzględnić podczas pisania programu

Zagadnienia te dzieliły się na trzy podstawowe grupy:

- problemy związane z charakterem obciążenia właściwym dla serwera multimedialnego,
- charakterystyczne właściwości środowiska Oracle Media Net (a w zasadzie modelu CORBA),
- specyficzne problemy, związane z konkretną wersją OVS (3.0.4).

3.1.1. Specyfika obciążenia serwerów multimedialnych

Serwery usług multimedialnych różnią się znacząco od klasycznych w dwóch obszarach [6]:

- konieczność odczytu, transmisji, a często i zapisu danych, w czasie rzeczywistym, z zachowaniem zależności czasowych obowiązujących te dane,
- konieczność składowania dużych bloków danych multimedialnych oraz zapewnienia wymaganych przez nie dużych szybkości przesyłu.

Dane multimedialne, składowane na takich serwerach, są pobierane przez klientów bez żadnego dodatkowego przetwarzania (inaczej niż np. w konwencjonalnych bazach danych, gdzie wymagane jest np. wykonanie drzewa zapytania SQL). Połączenie klienta z bazą trwa za to dłużej (rzędu dziesiątek minut albo godzin), ich rotacja będzie odpowiednio mniejsza.

W związku z tym, przy projektowaniu narzędzi wykorzystujących serwer multimedialny, przy założeniu, aby możliwie mało zakłócać jego pracę, przede wszystkim należy zmniejszyć obciążenie dysków (ilości odczytów) [7]. Minimalizacja wykorzystanego czasu obliczeniowego procesora ma znaczenie drugorzędne – jest ona w dalszym ciągu ważna, ale nie tak, jak w przypadku tradycyjnych, tekstowych baz danych.

W prezentowanym programie zostało to zrealizowane przez pobieranie porcji danych analizowanego filmu przez interfejs MDS, wykorzystywany w zasadzie wyłącznie przez administratora. Dzięki temu unika się zmniejszania przepustowości dostępnej na „pompie wideo", a tym samym, efektywnie dostępnej dla klientów.

Ma to stosunkowo duże znaczenie, gdyż często występują badania, w których aplikacje - klienci wykorzystują przez długi czas (czasami rzędu dni) całe dostępne pasmo efektywne. Dodatkowe, sztuczne redukowanie go przez uruchamianie narzędzi administracyjnych byłoby raczej niewskazane.

3.1.2. Zagadnienia projektowania aplikacji z wykorzystaniem modelu OMN / CORBA

Programowanie w OMN wykazuje wszystkie charakterystyczne cechy, jakie znaleźć można w modelu CORBA [3], z których najważniejsze, to:

- wszechstronne możliwości przekazywania informacji między klientami a serwerami, w tym takich elementów, jak: przekazywanie parametrów, właściwości i informacji między rozproszonymi aplikacjami, przesyłanie informacji i komunikatów, zdalne wywoływanie procedur i zarządzanie, wykorzystanie globalnej przestrzeni nazw,
- duża ilość udostępnianych platform sprzętowych, systemów operacyjnych i technologii sieciowych dostępnych dla projektanta,
- konieczność dokładnego projektowania aplikacji, z uwagi na duży „obszar swobody" przy pisaniu programów, mogący być w przyszłości źródłem błędów, przy wykorzystaniu aplikacji w innych środowiskach.

Najważniejszymi elementami programu pisanego w tym modelu są zdalne wywołania odpowiednich usług na serwerze. Duże znaczenie ma odpowiednie zaprojektowanie sekwencji tych wywołań, tak aby uniknąć niepożądanych czynników:

- zwiększenia obciążenia serwera i zakłócenia pracy klientów przez zadanie serwerowi więcej pracy niż to jest konieczne,
- destabilizacji serwera przez niepoprawne wywołanie odpowiednich usług (przekazanie złych parametrów, wykonanie wywołań „wyrwanych z kontekstu").

Pierwsze zagrożenie ominięto przez dokładne zaplanowanie momentów odwołań do odpowiednich usług. I tak, poza wykonywanym na początku nawiązaniem połączenia i rozłączeniem się podczas wyjścia z programu, jedyne wykorzystywane interfejsy to:

- interfejs zawartości (*logical content*)– uaktywniany raz na początku oraz na życzenie użytkownika, w przypadku np. konieczności zapisu wyników analizy,
- interfejs dostępu do bazy filmów (*MDS*) – uaktywniany, gdy użytkownik wydał komendę analiza, wykorzystywany sekwencyjnie, w celu pobrania porcji danych filmu do analizy.

Drugie zagrożenie ominięto projektując rozbudowane drzewo wywołań usług. Każdy jego węzeł to wywołanie odpowiedniej usługi, a liście oznaczają kolejne możliwe rezultaty akcji. Każdy z rezultatów jest obsłużony, co, w zależności od sytuacji, prowadzi do:

- w przypadku pozytywnym – wywołania kolejnej usługi,
- w przypadku negatywnym – w zależności od miejsca – powtórzenia ostatniego wywołania, całej sekwencji, odwołania do innych usług (np. odłączenie się od interfejsu), czy wreszcie całkowitego zerwania połączenia z serwerem.

3.1.3. Charakterystyczne właściwości systemu OVS w wersji 3.0.4

Większość elementów, które należało uwzględnić podczas projektowania, wymieniono powyżej, tym niemniej, być może z racji wczesnej wersji serwera, występują pewne elementy nietypowe:

- zerwanie sesji bez jej zamknięcia (otwarcie sesji jest jednym z etapów nawiązania połączenia, po inicjacji zmiennych środowiskowych, a przed uaktywnianiem interfejsów), powoduje destabilizację serwera, co jest szczególnie widoczne podczas pisania aplikacji klienckiej, wykorzystującej interfejs strumieni – kilkakrotne uruchomienie programu, w którym pojawia się błąd, powoduje konieczność restartowania serwera (inne interfejsy są bardziej odporne),
- odpowiednio duża liczba wystąpień błędów (nieobsłużonych wyjątków dowolnego typu) prowadzi do spadków wydajności (serwer działa wolniej, trzeba restartować).

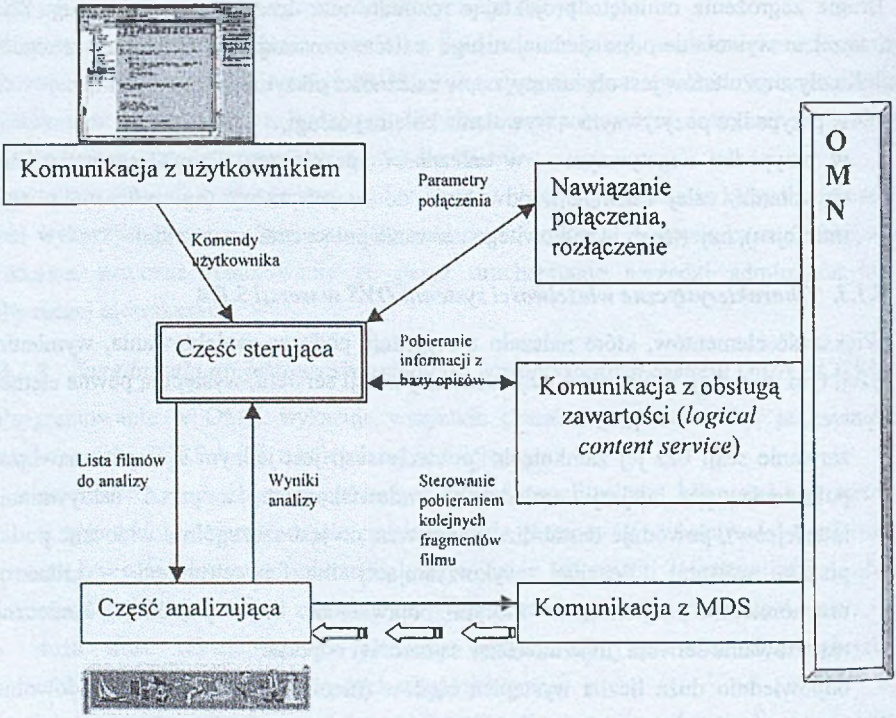
Zaproponowane w poprzednim punkcie rozwiązanie, uzupełnione o kilkustopniowe wychwytywanie wyjątków (kontrola przy każdym wywołaniu usługi i przy wykonywaniu procedur je zawierających), pozwoliło ominąć to zagrożenie.

3.2. Konstrukcja rozszerzenia

Opisywane rozszerzenie ma stanowić pomost między serwerem OVS z jednej strony, a narzędziem do analizy filmów z drugiej. W chwili obecnej został on przygotowany do pracy z modułem opartym na [5], ale jego ewentualna zmiana bądź uzupełnienie jest stosunkowo proste.

Program składa się z trzech głównych części:

- kontroli / komunikacji z użytkownikiem, zawierającej moduły – komunikacji z użytkownikiem i sterujący,
- moduł analizy, zawierający narzędzia do wydobywania informacji z filmu,
- zestaw procedur przygotowanych do łączenia się z OVS, pogrupowanych w moduły przypisane do konkretnego interfejsu.



Rys.3. Architektura projektowanego rozszerzenia OVS
Fig. 3. Architecture of designed OVS enchancement

Działanie programu polega na:

- wykonaniu sekwencji wywołań do nawiązania połączenia z serwerem, po czym następuje uruchomienie drugiego wątku, obsługującego graficzny interfejs użytkownika,
- pobieraniu od użytkownika zadań i wykonywaniu ich, przez wywołanie odpowiednich interfejsów sieciowych,
- zamknięciu połączenia po wydaniu komendy „wyjście".

Architekturę rozszerzenia przedstawia rysunek 3. Zostało ono napisane w języku C, z wykorzystaniem kompilatora *gcc*, pracuje w środowisku Unix.

3.2.1. Moduły - komunikacji z użytkownikiem i sterujący

Moduł komunikacji z użytkownikiem składa się z graficznego pośrednictwa, zbudowanego z wykorzystaniem biblioteki Motif. Użytkownik ma do wyboru szereg komend: wyświetlenie stanu połączenia z serwerem, pobranie listy filmów znajdujących się w bazie danych wraz z opisami i wynikami poprzednich analiz, wskazanie kilku z nich i wykonanie na nich analizy, zapis wyników analizy do bazy danych i wyjście z programu. Polecenia te są przekazywane wątkowi sterującemu z wykorzystaniem pamięci dzielonej (niezbędną synchronizację zapewniają semafory).

Część sterująca, zaraz po uruchomieniu, dokonuje próby nawiązania połączenia z serwerem. Następnie uruchamia drugi wątek, przekazując mu informację na temat rezultatu próby. W przypadku pozytywnym następuje uruchomienie części graficznej, w innym – następuje zamknięcie obu części i wykonanie sekwencji rozłączeniowej.

Zadaniem części sterującej jest interpretacja otrzymywanych od użytkownika poleceń. Wykonywane jest to, w zależności od polecenia, albo bezpośrednio, przez wywołanie odpowiednich funkcji z części odpowiedzialnej za komunikację z OVS, albo pośrednio, przez uruchomienie modułu analizy, a następnie oczekiwanie na wyniki.

3.2.2. Moduł analizy

Moduł analizy zbudowano w oparciu o programowy dekoder MPEG – 2 (w tym formacie zapisywane są filmy w bazie filmów OVS), stworzony przez MPEG Software Simulation Group. Został on odpowiednio zmodyfikowany, uzupełniony o narzędzie do zbierania danych z kolejno dekodowanych klatek. Danymi pobieranymi są wektory ruchu z każdej klatki, na ich podstawie wyliczane są opisujące je parametry, a z nich następnie tworzony jest opis filmu w formie ciągu współczynników. Opis ten jest następnie przekazywany przez część sterującą użytkownikowi oraz zapamiętywany, w celu ewentualnego późniejszego zapisania.

3.2.3. Procedury komunikacji z OVS

Wyodrębnienie poszczególnych części, odpowiedzialnych za połączenie z OVS, w oddzielne moduły, miało na celu uproszczenie struktury programu, jako że są to elementy dość złożone:

- część odpowiedzialna za nawiązanie połączenia z serwerem i rozłączenie – para procedur - mechanizm nawiązania połączenia jest wielostopniowy; najpierw trzeba połączyć się z OMN, następnie zainicjować zmienne środowiskowe Ovs, potem zainicjować na potrzeby połączenia poszczególne interfejsy; rozłączanie musi przebiegać w odwrotnej kolejności,
- część odpowiedzialna za komunikację z modułem MDS – jest to zestaw procedur; sprawdzanie, czy dany film jest w bazie danych, wysłanie poleceń jego otwarcia, zamknięcia i przewinięcia do wskazanego miejsca,
- część odpowiedzialna za komunikację z modułem opisu (logical content) – para procedur - odbierająca i zapisująca blok opisu filmu, uzupełniony ewentualnie o informacje z poprzednich analiz.

4. Wyniki działania programu

Po napisaniu, program został przetestowany w dwóch etapach: testowania niezawodności i generowanego obciążenia.

Testowanie niezawodności polegało na kilkakrotnym uruchomieniu klienta OVS (aplikacji wyświetlającej film), potem kilkudziesięciokrotnym uruchomieniu opisywanego rozszerzenia, a następnie ponownym uruchomieniu klienta. Podczas uruchamiania rozszerzenia wydawano losową serię komend. Okazało się, że rozszerzenie nie zawiesza serwera, ani nie destabilizuje jego pracy. Było to potwierdzeniem słuszności przyjętych założeń co do kontroli błędów.

Następnym etapem było uruchomienie nakładki z równolegle działającymi klientami – tu również wyniki były poprawne, tj. nie zaobserwowano różnic w jakości transmisji z uruchomionym rozszerzeniem i bez.

5. Podsumowanie i wnioski

Prezentowane rozszerzenie dobrze spełnia swoje zadanie, przede wszystkim dzięki poprawnie skonstruowanemu mechanizmowi obsługi błędów. Należy zauważyć, że wszelkie programy wykorzystujące interfejs CORBA, ze względu na jego uniwersalność, muszą

bardzo rygorystycznie kontrolować przebieg wykonania wywołań, zwłaszcza że zdarza się, że wystąpienie błędów jest niepowtarzalne, tzn. pojawia się raz na kilka wywołań programu.

W chwili obecnej program jest gotowy, następnym krokiem będzie uzupełnienie go o bardziej wyspecjalizowane metody analizy zawartości filmu.

Program wymaga również pewnego przetworzenia kodu źródłowego, który w chwili obecnej jest dość wyprofilowany pod kątem obecnego zastosowania; w takiej postaci zmiana modułów analizy może być utrudniona np. z powodu skomplikowanej struktury synchronizacyjnej.

LITERATURA

1. Furbush G., Stone D.: Oracle Video Server Developer's Guide. Oracle corp, 1998.
2. Maring S.: Oracle Media Net Developer's Guide. Oracle corp, 1997.
3. Meier R.: A Framework Providing Fault Tolerance Using the CORBA Trading Service. Departament of Computer Science, Trinity College, Dublin 1998.
4. Gokhale A, Schmidt D.: Evaluating CORBA Latency and Scalability Over High-Speed ATM Networks. Proc. ICDCS '97. Baltimore 1997.
5. Głomb P.: Projekt mechanizmu analizy i indeksowania informacji multimedialnej dla środowiska Oracle Video Server. Studia Informatica, vol 21, s. 275 – 286. Gliwice, 2000.
6. Gemmel J., Vin H., Kandlur D., Rangan V.: Multimedia Storage Servers: A Tutorial and Survey. IEEE Computer, 1995.
7. Vin H., Goyal A., Goyal P.: Algorithms for Designing Multimedia Servers. Proc. of First IEEE International Conference on Multimedia Computing and Systems, s. 234 – 243. Boston, 1994.
8. Maffeis S., Schmidt D.: Constructing Reliable Distributed Communication Systems with CORBA. IEEE Comunications, vol 14 no 2, luty 1997.
9. Vin H.: Multimedia System Architecture. Proc International Symposium on Photonic for Industrial Applications. Boston, 1994.

Recenzent: Dr inż. Katarzyna Stąpor

Wpłynęło do Redakcji 27 marca 2001 r.

Abstract

This paper presents a description of the administrator's tool for Oracle Video Server. It's primary use is implementing various methods of movie content analysis, [5] in particular. An overview of Oracle Video Server and Oracle Media Net (a CORBA – derived distributed communication network) is presented, then a discussion of important facts that have to be considered while writing server's extensions (multimedia servers workload profile and consequences of CORBA philosophy for programmers). After that, program architecture is described, along with its behavior and conclusions, based on implementation.