Raport projektu celowego
Obliczenia wielkiej skali i wizualizacja do zastosowań w wirtualnym laboratorium z użyciem klastra SGI

Zadanie WP 2.1. Zdalny dostęp do bibliotek naukowych

WP. 2.1.5. Integracja systemu udostępniania bibliotek matematycznych ze strukturami klastra SGI

Maciej Brzeźniak, Tomasz Makieła

Poznańskie Centrum Superkomputerowo-Sieciowe

Poznań, październik 2004 r.

Uaktualnienie raportu z listopada 2003 r.

Spis treści

1. Informacje ogólne

1.1. Status tego dokumentu

W ramach prac projektu celowego „Obliczenia wielkiej skali i wizualizacja do zastosowań w wirtualnym laboratorium z użyciem klastra SGI" ośrodki PCSS i TASK opracowują System Udostępniania Bibliotek Matematycznych (SUBM).

Cechy i funkcjonalność istniejących systemów udostępniania bibliotek matematycznych (NetSolve [NETSOLVE], Ninf [NINF] opisuje raport nr. 2 p.n. „Przegląd cech i funkcjonalności systemów udostępniania bibliotek matematycznych, raport z testów i analizy kodów źródłowych, raport z wykonania instalacji testowej" [REPORT2]. Zestawienie oprogramowania matematycznego wykorzystywanego w ośrodkach naukowych w Polsce oraz informacje jakie oprogramowanie zostanie włączone do systemu udostępniania zawiera raport nr 1. p.n. „Wykorzystanie oprogramowania matematycznego w Polsce [REPORT1]. Raporty te są efektem prac prowadzonych w pierwszych sześciu miesiącach projektu.

W toku prac w pierwszej części projektu wytypowano system NetSolve jako bazę dla Systemu Udostępniania Bibliotek Matematycznych.

Niniejszy raport zawiera opis koncepcji i projektu integracji systemu udostępniania bibliotek ze środowiskiem klastra SGI a także informacje o implementacji. Część dokumentacji administratora systemu udostępniania dotycząca konfiguracji opracowanych narzędzi integracji (planowana jako deliverable dla zadania WP.2.1.5 projektu) jest składnikiem dokumentacji administratora systemu, która została opublikowana przez PCSS i TASK w końcu października b.r.

Niniejszy dokument jest uaktualnieniem raportu z miesiąca 12 projektu.

1.2. Opis prac

Zadanie WP.2.1.5 ma na celu połączenie Systemu Udostępniania Bibliotek Matematycznych ze strukturami klastra SGIgrid w stopniu umożliwiającym wykorzystanie zasobów klastra SGIgrid przez użytkowników Systemu Udostępniania Bibliotek Matematycznych.

1.2.1. Cechy klastra SGIgrid mające wpływ na SUBM

W klastrze SGI (SGIgrid) system SUBM jest jednym z podsystemów współużytkujących zasoby obliczeniowe. Narzuca to szereg wymagań. Po pierwsze, system nie może zawłaszczać tych zasobów. Po drugie, ich wykorzystanie musi być zgodne z lokalnymi

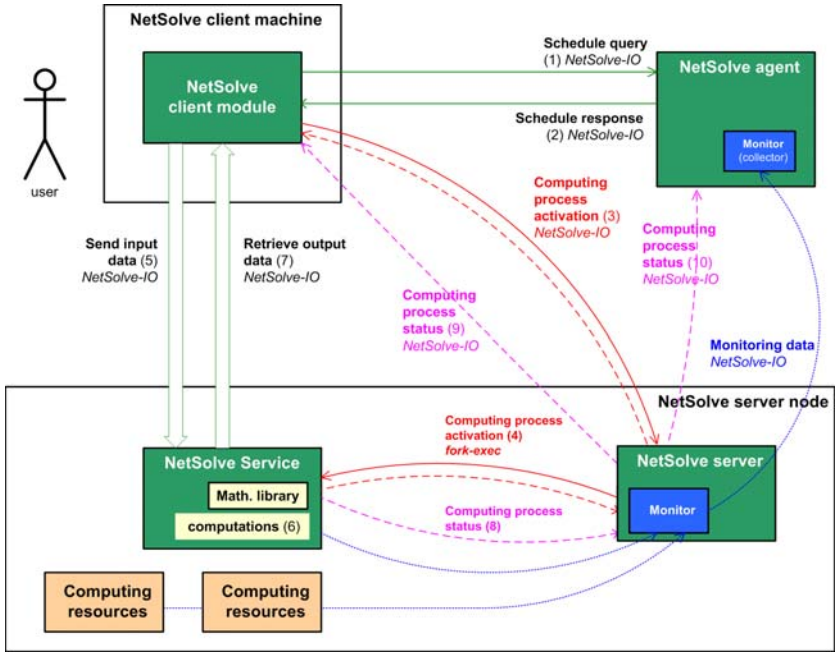
(określonymi dla poszczególnych węzłów i klastrów lokalnych) zasadami użytkowania i dostępu. Zasoby klastra SGIgrid to w maszyny należące do różnych ośrodków obliczeniowych. Są to urządzenia, które nie mogą być dedykowane dla potrzeb klastra SGIgrid. W większości przypadków niemożliwe jest również zakładanie specjalnych kont dla użytkowników klastra SGIgrid na poszczególnych komputerach

W ramach prac nad projektem SGIgrid uczestniczące ośrodki opracowały wspólną koncepcję mechanizmów dostępu użytkowników i aplikacji do zasobów dla klastra SGIgrid. Zadanie WP.2.1.5 obejmuje prace, mające na celu dostosowanie SUBM do zdefiniowanych ogólnych mechanizmów dostępu.

1.2.2. Problemy integracji SUBM i klastra SGIgrid

Jednym z głównych problemów, których rozwiązanie było przedmiotem prac w ramach zadania WP.2.1.5. jest fakt, że system NetSolve, przyjęty jako baza dla SUBM nie ma możliwości wykorzystania zasobów obliczeniowych (węzłów klastrów, pojedynczych maszyn), które są kontrolowane przez zarządców zasobów, takich jak systemy kolejkowe, m.in. system LSF. Jest to spowodowane następującymi cechami NetSolve.

Działanie modułów systemu NetSolve w oryginalnej wersji wymaga uruchamiania procesów-demonów na każdej z maszyn na których mają być uruchamiane obliczenia. Demony stale nasłuchują na wybranych portach na żądania wykonania obliczeń (patrz Rysunek 1). Na większości węzłów klastra SGIgrid taki schemat działania jest niedopuszczalny.



Rysunek 1 Architektura systemu NetSolve

Kolejnym problemem jest autoryzacja użytkowników. W systemie NetSolve zadania na systemie obliczeniowym uruchamiane są z prawami użytkownika, który uruchomił proces-demon. W tym podejściu wszystkie żądania wykonania obliczeń przychodzące z systemu NetSolve do węzła obliczeniowego są traktowane jednakowo, bez rozróżniania na użytkowników końcowych systemu NetSolve. Jedyną metodą ograniczenia zasobów polega na określeniu maksymalnej liczby zleceń uruchomionych w danym czasie przez danego klienta na danym zasobie. Klienta (a właściwie grupę klientów) specyfikuje się bardzo zgrubnie - po nazwie domeny (można wykorzystywać symbole wieloznaczne np. *.edu.pl).
Natomiast klienci systemu NetSolve w stosunku do tego systemu autoryzowani są bardzo prostymi metodami, nie zapewniającymi wymaganego w środowisku rozproszonym (w którym węzły należą do różnych organizacji) poziomu bezpieczeństwa. Również komunikacja w systemie NetSolve odbywa się „otwartym tekstem", bez wzajemnej autoryzacji uczestniczących w niej węzłów. Jest to niedopuszczalne w środowisku klastra SGIgrid.
Prace w ramach zadania WP.2.1.5. zmierzają do rozwiązania tych problemów.

1.2.3. Zmiany w zadaniu podczas trwania projektu

Na etapie planowania projektu zakładano, że dostęp do poszczególnych maszyn i węzłów klastra SGI odbywać się będzie wyłącznie za pośrednictwem systemu LSF (ang. Load Sharing Facility, [LSF]). Część prac nad mechanizmami integracji SUBM z klastrem SGIgrid wykonana została przy tym założeniu. Początkowo planowano wykonanie modułów pośredniczących pomiędzy systemem NetSolve i LSF. Miały one realizować dwie główne funkcje. Pierwsza z nich to uruchamianie zadań przez agenta systemu poprzez interfejs do systemu kolejkowego: przesyłanie danych wejściowych do odpowiedniej lokalizacji, uruchamianie procesu obliczeniowego w systemie kolejkowym za pomocą interfejsu systemu LSF oraz pobieranie i odsyłanie danych wynikowych do modułu klienta NetSolve. Druga, to pozyskiwanie z systemu LSF informacji o wydajności obliczeń a także stanie systemu kolejkowego. Informacje te miały zostać wykorzystane przez agenta systemu do szeregowania żądań wykonania funkcji matematycznych.
W listopadzie 2003 roku konsorcjum realizujące projekt podjęło decyzję o zmianie mechanizmu współdzielenia zasobów klastra SGI z planowanego wcześniej systemu LSF na system Globus. Ta decyzja miała znaczny wpływ na przebieg prac w zadaniu WP.2.1.5.

1.2.4. Ostateczny przedmiot prac

W listopadzie 2003 roku ustalono, że dostęp do zasobów klastra SGI odbywać się będzie poprzez system Globus w wersji 2.4. System NetSolve został dostosowany do współpracy z Globusem. Wykorzystanie poszczególnych elementów pakietu narzędzi Globus (ang. Globus Toolkit [Globus]) pozwoliło z jednej strony na dostęp systemu NetSolve do zasobów kontrolowanych przez Globusa, z drugiej pozwoliło spełnić wymagania środowiska SGI takie jak bezpieczeństwo, wymaganie wzajemnej autoryzacji modułów itd.

W przyjętym rozwiązaniu komunikacja pomiędzy modułami systemu udostępniania bibliotek odbywa się z użyciem techniki Globus I/O. System NetSolve został zmodyfikowany tak, że wywołania tradycyjnych funkcji socketowych (open(), accept(), connect(), close()) zostały zastąpione wywołaniami funkcji Globus I/O. Takie podejście zapewnia z jednej strony możliwość szyfrowania danych przesyłanych przez kanał komunikacyjny, z drugiej strony zapewnia wzajemną autoryzację stron uczestniczących w komunikacji. Każde połączenie Globus I/O jest autoryzowane odpowiednimi certyfikatami cyfrowymi obu modułów uczestniczących w komunikacji. Szczegóły projektu i implementacji opisane zostały w punkcie 2.3 raportu.

Ważnym elementem wpływającym na wydajność obliczeń w systemie udostępniania bibliotek jest przesył danych wejściowych i wyjściowych. W wielu przypadkach objętość tych danych mierzy się w dziesiątkach megabajtów lub nawet w gigabajtach. Klaster SGIgrid jest systemem rozproszonym nie tylko w sensie logicznym ale również w sensie geograficznym. Narzut czasowy na przesył dużych zbiorów danych w rozległym środowisku jest spory, nawet przy wykorzystaniu pojemnych łącz. System udostępniania bibliotek taki jak NetSolve przesyła dane do obliczeń w trybie połączeniowym. Przepustowość uzyskiwana w tym trybie jest ograniczona szerokością dostępnego pasma łącza ale także opóźnieniem wnoszonym przez nie.

W pakiecie Globus Toolkit dostępne jest narzędzie GridFTP. Pozwala ono m.in. na transfer danych przy użyciu wielu wątków transmisyjnych. PCSS zdecydowało się na wykorzystanie protokołu GridFTP do przesyłu danych wejściowych i wyników obliczeń między modułem klienta systemu NetSolve a modułem serwerowym NetSolve (uruchamianym na maszynie obliczeniowej). Pozwala to znacznie skrócić czas przesyłania danych a tym samym skrócić czas obsługi żądań kierowanych do systemu udostępniania. Protokół GridFTP poza dużą przepustowością transmisji zapewnia również wzajemną autoryzację stron oraz możliwość szyfrowania przesyłanych danych.

Prace nad modyfikacją mechanizmów systemu NetSolve tak by wykorzystały one protokół GridFTP zostały zakończone. Szczegóły projektu i implementacji opisane zostały w punkcie 2.4 raportu.

Kolejnym elementem systemu NetSolve, który został zmodyfikowany dla zapewnienia integracji z klastrem SGIgrid jest mechanizm uruchamiania obliczeń na serwerach obliczeniowych. Do uruchamiania zadań w klastrze SGIgrid wykorzystano usługę Globus GRAM. Wykorzystanie tej usługi pozwala zrezygnować z umieszczania demonów systemu NetSolve na wszystkich maszynach obliczeniowych. Demony systemu NetSolve pozostały jedynie na wydzielonych komputerach, które realizują funkcje agenta i serwerów systemu NetSolve. Klienci bezpośrednio komunikują się tylko z tymi jednostkami (Globus-IO); dostęp do maszyn obliczeniowych mają jedynie serwery (poprzez GRAM).

Poprzez usługę GRAM możliwy jest dostęp do różnych zasobów obliczeniowych w jednolity sposób, z wykorzystaniem narzędzi GRAM dostępnych z linii komend lub interfejsu programistycznego GRAM. Globus Toolkit zawiera „przejściówki" (ang. *plug-in*) do rozmaitych systemów obliczeniowych. Dzięki temu możliwy jest dostęp i kontrola wykonania zadania na tych systemach: pojedynczych komputerach, klastrach, komputerach SMP i MPP, klastrach LSF i PBS i innych z wykorzystaniem jednolitego interfejsu użytkownika usługi GRAM i przejściówek tej usługi to poszczególnych systemów obliczeniowych.

Zaletą wykorzystania Globus GRAM do uruchamiania zadań w środowisku rozproszonym jest również fakt, że do nowych systemów zarządzania zasobami lub nowych wersji istniejących systemów (np. LSF) rozwijane są kolejne, aktualne „przejściówki".

Prace projektowe i implementacyjne oraz prace testowe związane z wykorzystaniem usługi GRAM dla NetSolve zostały zakończone. Szczegóły projektu i implementacja zostały opisane w punkcie 2.5 raportu.

1.2.5. Relacja do prac prowadzonych na świecie

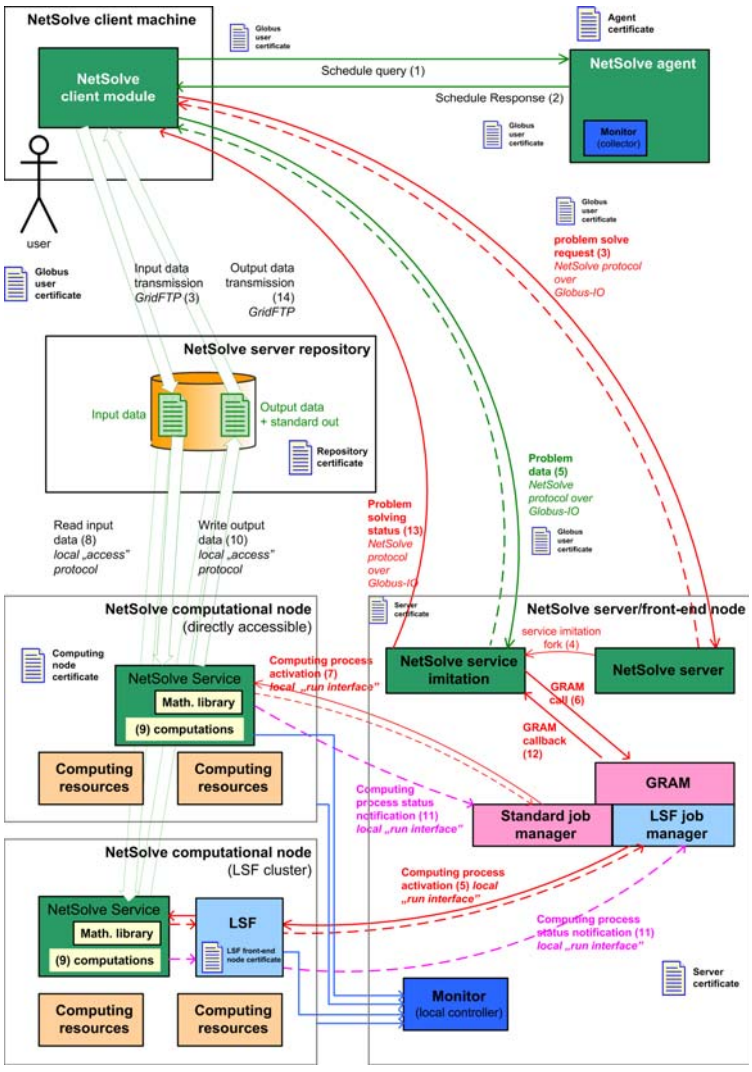
Oparcie systemu NetSolve na usługach pakietu Globus Toolkit jest nowym podejściem. Wprawdzie z pakietem NetSolve dostarczany jest moduł proxy dla Globus (można go uaktywnić przy kompilacji systemu), jednak moduł zawiera zaledwie częściową implementację mechanizmu uruchamiania obliczeń. Nie obejmuje on bezpiecznej komunikacji w ramach systemu NetSolve ani przesyłu danych wejściowych i wyników obliczeń za pomocą szybkich i bezpiecznych mechanizmów. W istniejącym rozwiązaniu nie ma również możliwości konfiguracji mechanizmu dostępu do usług GRAM – dane konfiguracyjne zaszyte są „na sztywno" w kodzie oryginalnego NetSolve.

Zespół pracujący nad systemem NetSolve w Innovative Computing Labs na University of Tennessee w Stanach Zjednoczonych wyraził zainteresowanie pracami prowadzonymi w PCSS w kontekście integracji systemu NetSolve z usługami pakietu Globus. [NETSNEWS]

2. Integracja

2.1. Integracja z usługami Globus

Integracja systemu udostępniania bibliotek (NetSolve) z klastrem SGIgrid polega na takiej modyfikacji systemu NetSolve by korzystał on z mechanizmów takich jak Globus I/O do komunikacji wewnętrznej, GridFTP do przesyłania danych wejściowych i wyników obliczeń oraz usługi GRAM do uruchamiania procesów obliczeniowych na serwerach. Schemat integracji systemu NetSolve z mechanizmami systemu Globus zawiera Rysunek 3.



Rysunek 3 Schemat integracji systemu NetSolve z mechanizmami Globus

Klient, agent i serwer systemu wyposażeni zostali w certyfikaty cyfrowe, którymi posługują się do bezpiecznej komunikacji przez Globus I/O. Repozytorium jest nowym obiektem wykorzystywanym podczas przetwarzania zadań w NetSolve: klient składuje do niego dane wejściowe dla obliczeń i odbiera z niego wyniki. Proces-serwis pobiera dane wejściowe z repozytorium i składuje w nim wyniki obliczeń. Szczegóły integracji NetSolve z mechanizmami systemu Globus przedstawione są w punktach 2.1-2.3 raportu.

2.2. Dostęp do bibliotek naukowych w klastrze SGIgrid

Na wcześniejszych etapach prac nad projektem rozważano integrację zmodyfikowanego NetSolve z brokerem SGIgrid, modułem RAD oraz mechanizmami VUS.

Ostatecznie PCSS i TASK, w porozumieniu z zespołami odpowiedzialnymi za Broker, moduł RAD oraz mechanizmy VUS, zrezygnowały z integracji NetSolve z tymi mechanizmami. Zdecydowano, że dostęp do zdalnych bibliotek matematycznych będzie udostępniany jako niezależna usługa w klastrze SGIgrid.

W zmodyfikowanym NetSolve dostęp do zasobów odbywa się za pośrednictwem systemu Globus. Dzięki temu usługa dostępu do bibliotek matematycznych pozostaje zgodna z ogólnymi zasadami zarządzania zasobami w SGIgrid uzgodnionymi przez konsorcjum realizujące projekt.

Za wydzieleniem systemu dla dostępu do bibliotek matematycznych jako oddzielnej usługi przemawiają przede wszystkim przesłanki dotyczące wydajności przetwarzania zadań w tym systemie. Na wydajność przetwarzania w NetSolve w znacznym stopniu wpływa sprawność szeregowania zadań. Podczas szeregowania zadań NetSolve brane są pod uwagę specyficzne, charakterystyczne tylko dla wywołań funkcji matematycznych czynniki, m.in. informacje o wydajności poprzednich wywołań funkcji na poszczególnych zasobach obliczeniowych, szereg parametrów dotyczących zadania obliczeniowego (rozmiar zadania, wersja algorytmu, złożoność obliczeniowa, wartości parametrów modyfikujących działanie funkcji matematycznej itp.) oraz specyficzne dane dotyczące stanu systemów obliczeniowych. Dane dotyczące stanu systemów zawierają m.in. informacje o liczbie procesów w systemie, dostępnej i zajętej pamięci, obszarze wymiany (ang. *swap*), liczbie wymian na sekundę, liczbie operacji wejścia/wyjścia na sekundę w systemie itp. Wprowadzanie tych informacji do systemu informacyjnego brokera jest bezcelowe. Są one pobierane z systemów obliczeniowych za pomocą specjalnych, wydzielonych, dedykowanych dla NetSolve metod. Z racji ich objętości przetwarzane są specjalnymi metodami przez Agenta NetSolve. Ich

przetwarzanie przez Broker SGIgrid nie gwarantowałoby odpowiednio krótkiego czasu odpowiedzi systemu NetSolve na zapytanie o przydział zasobów do obliczeń. W toku prac nad projektem PCSS i TASK przeprowadziło analizę i szereg testów wydajnościowych związanych z operacjami na systemie informacyjnym brokera. Z analizy i testów wynika, że wydajność tych operacji jest nieporównywalnie niższa niż operacje na lokalnej bazie danych o wydajności agenta systemu NetSolve. Dlatego całość procesów związanych z szeregowaniem zadań wykonania zdalnego funkcji matematycznych wykonuje w rozwiązaniu opracowanym przez PCSS i TASK Agent NetSolve. Więcej informacji na temat mechanizmów szeregowania w zmodyfikowanym NetSolve zawiera raport 6 zadania WP.2.1.6 projektu celowego.

2.3. Komunikacja (GlobusIO)

2.3.1. Wstęp

System komunikacji w NetSolvie 2.0 opiera się na gniazdach TCP oraz w wersji na platformy *nix'owe na lokalnych połączeniach (wykorzystujących protokoły z dziedziny Unixa, tzw.ang. Unix domain protocols).

Możemy wyodrębnić następujące składniki systemu:

- agenta
- serwer
- serwisy – są to zadania uruchamiane przez serwer
- klienta
- proxy – to łącznik między klientem a systemem NetSolve.

Dodatkowo, ważne z punktu widzenia komunikacji, możemy wyodrębnić procesy spełniające zadanie koła ratunkowego, tak zwane procesy *lifelink*. Ich zadaniem jest powiadamianie systemu o nieprawidłowościach w działaniu poszczególnych składników systemu (w szczególności chodzi tu o *proxy* oraz uruchomiony serwis). Mechanizm ten pozwala utrzymać stabilność systemu w przypadku problemów z którymś z jego elementów, nie stanowiących jego głównego trzonu.

2.3.2. Moduł Globus-IO

W rozproszonym klastrze SGI niedopuszczalny jest model komunikacji, w którym brakuje podstawowych mechanizmów bezpieczeństwa a dane przesyłane są w postaci “czystego tekstu”. Zaimplementowany system komunikacyjny, korzystający z biblioteki GlobusIO,

został wyposażony w mechanizmy szyfrowania oraz autentykacji. Dostosowanie systemu NetSolve wymagało zmiany również samego protokołu komunikacyjnego.

Biblioteka GlobusIO opiera się na mechanizmach gniazd, oferując jednolity interfejs dla operacji wejścia/wyjścia. Dotyczy on zarówno transmisji sieciowej (typu strumieniowego i datagramowego) jak i operacji na plikach. Pozwala w łatwy sposób tworzyć połączenia, ustawiać, modyfikować i odczytywać atrybuty tych połączeń. Dodatkowo udostępnia funkcje do obsługi komunikacji asynchronicznej, nieblokującej (wykorzystane są tutaj mechanizmy powiadamiania tzw. *callback'u*).

Dostarczone mechanizmy i funkcje pozwalają w łatwy, bezpieczny i odporny na błędy sposób zaimplementować protokół komunikacyjny.

Do poprawnego działania systemu NetSolve wraz z modułem GlobusIO potrzebny jest odpowiednio zainstalowany i skonfigurowany system Globus w wersji 2.4. Podczas kompilacji/instalacji systemu NetSolve potrzebne są pliki z pakietu SDK Globusa zawierające oprócz podstawowych bibliotek i narzędzi, dodatkowe pliki nagłówkowe niezbędne w procesie kompilacji. W późniejszym wykorzystaniu systemu NetSolve wymagany jest tylko dostęp do bibliotek odpowiednich modułów Globusa. Dobra konfiguracja Globusa musi obejmować instalację odpowiednich części systemu GLOBUS (GlobusCommon, GlobusIO), ustawienie ścieżek do bibliotek Globusa oraz innych narzędzi ($GLOBUS_PATH, $GLOBUS_LOCATION), oraz konfigurację certyfikatów (dla użytkownika i systemu).

2.3.3. Szczegóły implementacji

Implementacja warstwy komunikacyjnej wykorzystującej moduł GlobusIO została zaprojektowana w taki sposób, aby w jak najmniejszym stopniu ingerować w oryginalny kod NetSolve'a. Dodatkowo starano się modyfikować kod w sposób przejrzysty aby możliwe było łatwe znalezienie różnic pomiędzy zmodyfikowaną wersją a oryginałem. Pozwoli to, w przyszłości, szybko zmienić nowe wersje NetSolve tak, by wykorzystywały opracowaną przez PCSS warstwę komunikacyjną. Również dostosowanie systemu do wykorzystania innego modułu komunikacyjnego powinno być łatwiejsze (np. podczas reimplementacji kodu zgodnie z wymaganiami Globus wersji 3 i wykorzystania modułu GlobusXIO - ang. *Globus eXtensible Input Output library*).

Warstwa komunikacyjna wykorzystująca GlobusIO jest dostępna jako opcja. Podczas kompilacji źródeł NetSolve (a dokładniej podczas wykonania standardowej konfiguracji poleceniem ./configure) można ją uaktywnić podając opcję –with_globusio. Spowoduje to uaktywnienie modułu GlobusIO w systemie NetSolve. W dalszym etapie kompilacji tak

skonfigurowanego systemu wykorzystywane są zasoby systemu Globus (pliki nagłówkowe, biblioteki, standardowe narzędzia pomocnicze) – dlatego instalacja Globusa jest tutaj niezbędna do dalszego działania.

Szyfrowanie oraz autentykacja protokołu transmisji zostały również zaimplementowane jako opcje z możliwością ich wyłączenia (standardowo obie są aktywne). W tym wypadku – ponieważ są to opcje, które nie powinny być wyłączane, gdyż wówczas nie ma właściwie różnicy pomiędzy standardową a zmodyfikowaną wersją komunikacji w NetSolvie, p wyłączeniu szyfrowania i autentykacji decydują zmienne globalne, zdefiniowane w kodzie aplikacji.

Na potrzeby znajdowania i poprawiania błędów, testowania zmodyfikowanego systemu NetSolve, a w szczególności protokołu komunikacyjnego opartego na GlobusIO, dodano mechanizm wyświetlania i logowania informacji. Przeznaczony jest on przede wszystkich dla programistów (uaktywniany opcją podczas kompilacji).

Implementacja komunikacji opartej na bibliotece GlobusIO została tak zaprojektowana, aby maksymalnie odzwierciedlała istniejącą warstwę komunikacyjną. Dodane nowe funkcje, są komplementarnymi odpowiednikami funkcji, które znajdują się w standardowym kodzie NetSolve'a. Poniżej, w tabeli Tabela 1 przedstawiono porównanie funkcji.

NETSOLVE	**GLOBUSIO-NETSOLVE**
PODSTAWOWE FUNKCJE KOMUNIKACYJNE	
`BindToFirstAvailablePort()`	`gio_bindToFirstAvailablePort()`
`establishSocket()` `establishLocalSocket()`	`gio_establishSocket()`
`AcceptConnection()`	`gio_acceptConnection()`
`ContactHost()` `ContactLocalHost()`	`gio_contactHost()`
`connectToSocket()` `connectToLocalSocket()`	`gio_connectToSocket()`
ZAAWANSOWANE FUNKCJE KOMUNIKACYJNE (operujące przez *obiekt* NS_Communicator)	
`acceptTransaction()`	`gio_acceptTransaction()`
`initTransaction()`	`gio_initTransaction()`
`newCommunicator()`	`gio_newCommunicator()`
`Recv8BitFlag()`	`gio_recv8BitFlag()`
`send8BitFlag()`	`gio_send8BitFlag()`
FUNKCJE DODATKOWE	
`reportLogToAgent()`	`Gio_reportLogToAgent()`

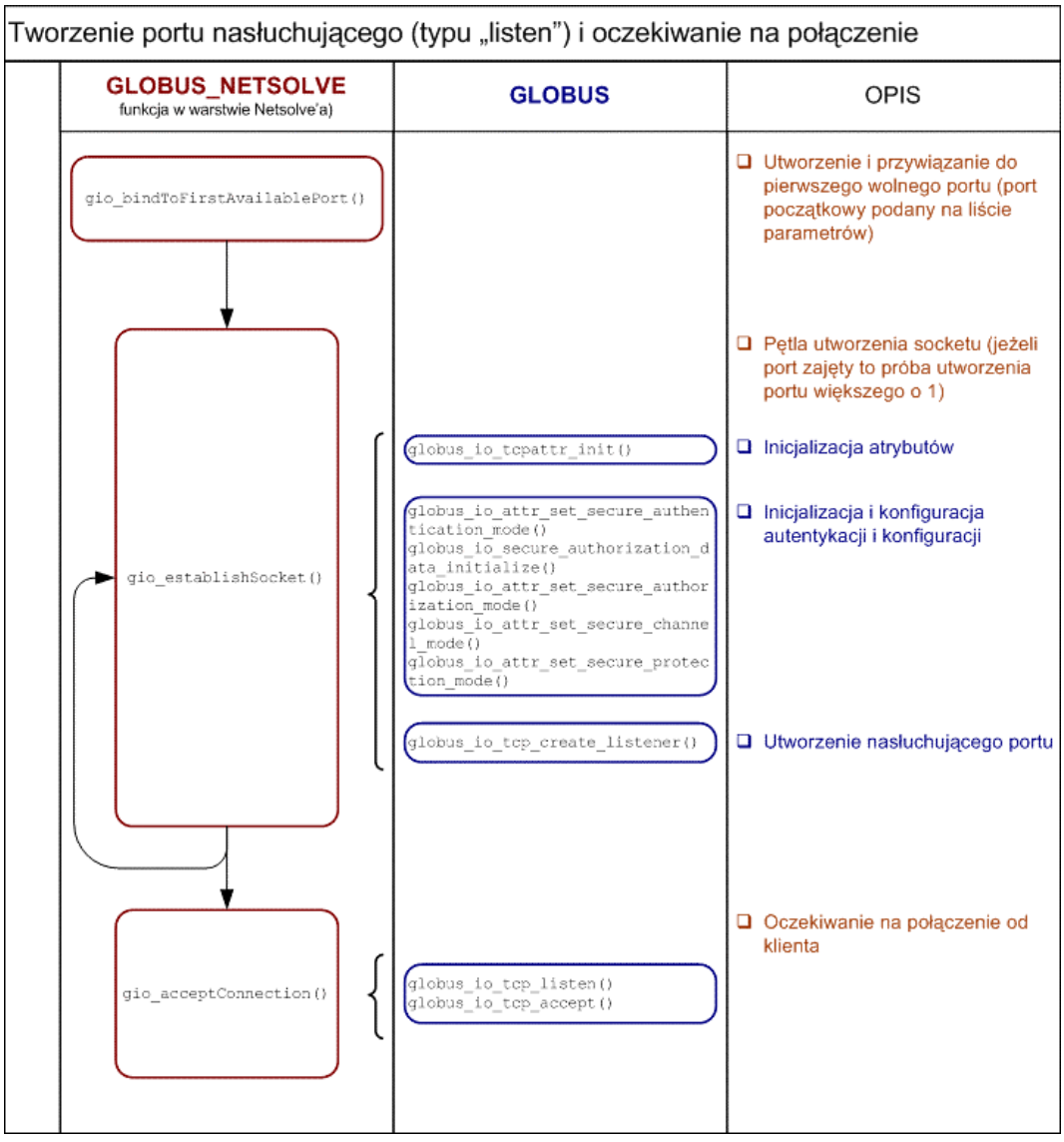
`startProxy()`	`Gio_startProxy()`
`forkStandardService()`	`Gio_forkStandardService()`
`forkStandardNoFileService()`	`Gio_forkStandardNoFileService`
`forkStandardFileService_sequence()`	`Gio_forkStandardFileService_sequence()`
`forkStandardFileService()`	`Gio_forkStandardFileService()`

Tabela 1 Porównanie funkcji w oryginalnym NetSolvie i zmodyfikowanym

2.3.3.1. Scenariusze komunikacyjne z wykorzystaniem modułu GlobusIO

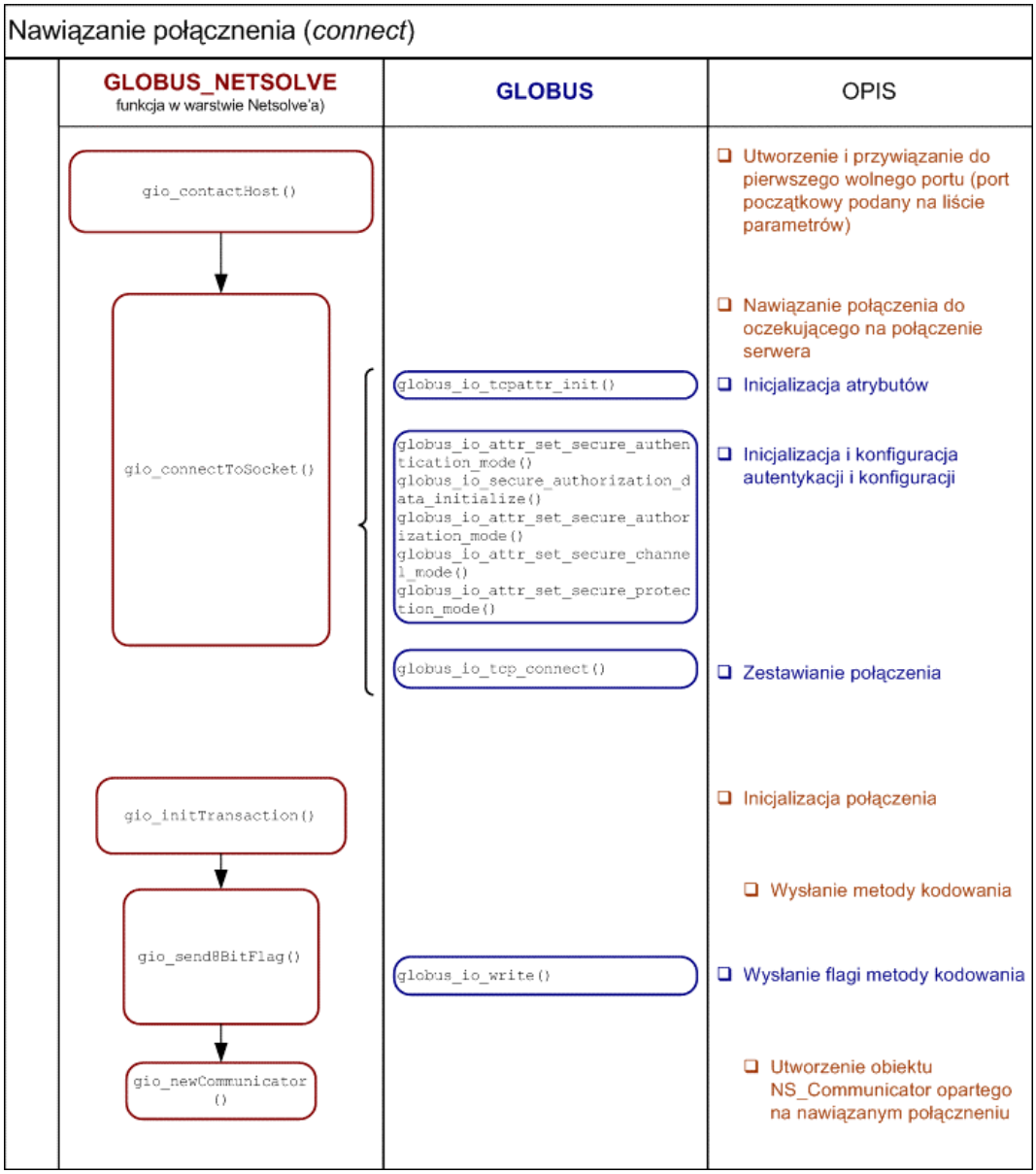
Poniżej przedstawione są scenariusze komunikacji w ramach przetwarzania NetSolve. Podczas komunikacji używana jest technika GlobusIO.

- tworzenie i inicjalizowanie portu nasłuchującego typu „listen”



Rysunek 6 Tworzenie i inicjalizowanie portu nasłuchującego typu „listen”

- nawiązywanie połączenia od strony klienta



Rysunek 7 Inicjalizacja i nawiązywanie połączenia od strony klienta

2.4. Przesyły danych do obliczeń (GRIDFtp)

2.4.1. Wstęp

W systemie NetSolve zostały zdefiniowane różne typy obiektów, które są przekazywane w systemie jako parametry wejściowe i wyniki podczas wykonywania obliczeń przez serwisy. Własności problemów (m.in. ich obiekty stanowiące parametry wejściowe i wyjściowe) są specyfikowane w plikach .PDF (Problem Description File). W NetSolve udostępniono następujące typy obiektów: wartość skalarna, wektor, macierz, macierz rozrzedzona (w formacie CRS – Compressed Row Storage), plik, spakowany plik, funkcja (UPF – User

Provided Function), ciąg znaków (STRING), lista ciągów znaków. Pierwsze cztery typy obiektów mogą przechowywać różne typy danych. Dostępne typy to integer, character, byte, float, double, single precision complex, double precision complex. Pozostałe obiekty są typu character.

Przesyłanie danych w systemie NetSolve odbywa się w każdym przypadku (nawet dla obiektów typu plik) z wykorzystaniem standardowych metod sieciowego protokołu połączeniowego (read(), write()). Wydłuża to dość znacznie czas rozwiązywania problemów, szczególnie dla dużych danych, w mocno rozproszonym środowisku, z ograniczonymi łączami komunikacyjnymi. W problemach matematycznych bardzo często mamy do czynienia z danymi wejściowych i wyjściowych o dużej objętości.

2.4.2. GridFTP – uniwersalny protokół transmisji

Mając na uwadze specyfikę systemu w jakim będzie zainstalowany moduł udostępniania bibliotek matematycznych, postanowiono udoskonalić metodę transmisji. GridFTP jest protokołem, który znacznie rozszerza dobrze znany protokół FTP równocześnie posiadający jego dobre cechy:

- jest to sformalizowany, dobrze opisany i rozumiany protokół;
- z architekturą pozwalającą w łatwy sposób na jego rozszerzenie;
- przesyłanie odbywa się pomiędzy klientem a serwerem;
- umożliwia przesyłanie w trybie third party transfers – przesyłanie danych pomiędzy dwoma zdalnymi serwerami kontrolowane z lokalnej maszyny.

Dodatkowo, protokół GridFTP wspiera:

- wielowątkowe przesyłanie danych, która wydatnie zwiększa przepustowość,
- przesyłanie danych fragmentami,
- bezpieczeństwo poprzez zaimplementowanie autoryzacji klientów i serwerów biorących udział w przesyłaniu danych,
- kontrolę nad poprawnym przesyłaniem danych poprzez metody wykrywania i naprawiania błędów podczas transmisji.

Dzięki tym funkcjom możliwe jest podniesienie jakości przesyłania, prędkości a także automatyczne zaimplementowanie nowych funkcji jak np. zwiększonego bezpieczeństwa przesyłania danych.

2.4.3. Szczegóły implementacji

Interfejs protokołu GridFTP składa się z dwóch modułów: globus-ftp-control i globus-ftp-client. Pierwszy z nich dostarcza metod pozwalających na nisko-poziomową kontrolę transferu. Drugi natomiast zawiera funkcje wywoływane w aplikacjach chcących korzystać z protokołu GridFTP (również w naszym zmodyfikowanym NetSolve'ie).

Podobnie jak w implementacji warstwy komunikacyjnej GlobusIO, podczas projektowania transmisji GridFTP starano się w jak najmniejszy sposób ingerować w oryginalne kody NetSolve. Obsługa przesyłania danych przez GridFTP została maksymalnie odseparowana od reszty kodu, równocześnie zachowując standardy i rozwiązania użyte w oryginalnym kodzie systemu NetSolve. Pozwoli to na łatwe (w przyszłości) dostosowanie naszego kodu do nowej wersji NetSolve'a.

Moduł przesyłania danych z wykorzystaniem GridFTP jest dostępny w naszym systemie jako opcja – podczas konfiguracji należy dodać –with_gridftp w wywołaniu standardowej komendy: ./configure (przed kompilacją systemu).

Standardowo, do przesyłania danych poprzez protokół GridFTP wykorzystywana jest stała liczba równolegle uruchomionych połączeń (obecnie = 4). Stała ta jest zdefiniowana w kodzie programu. Jej optymalna wartość powinna być wyznaczona podczas serii testów w konkretnej instalacji NetSolve. W późniejszym czasie, sposób definiowania tej wartości może zostać zmieniony tak by możliwe było dynamiczne określanie liczby równoległych połączeń np. przez agenta na podstawie gromadzonych danych z wcześniejszych połączeń pomiędzy klientem i serwerem.

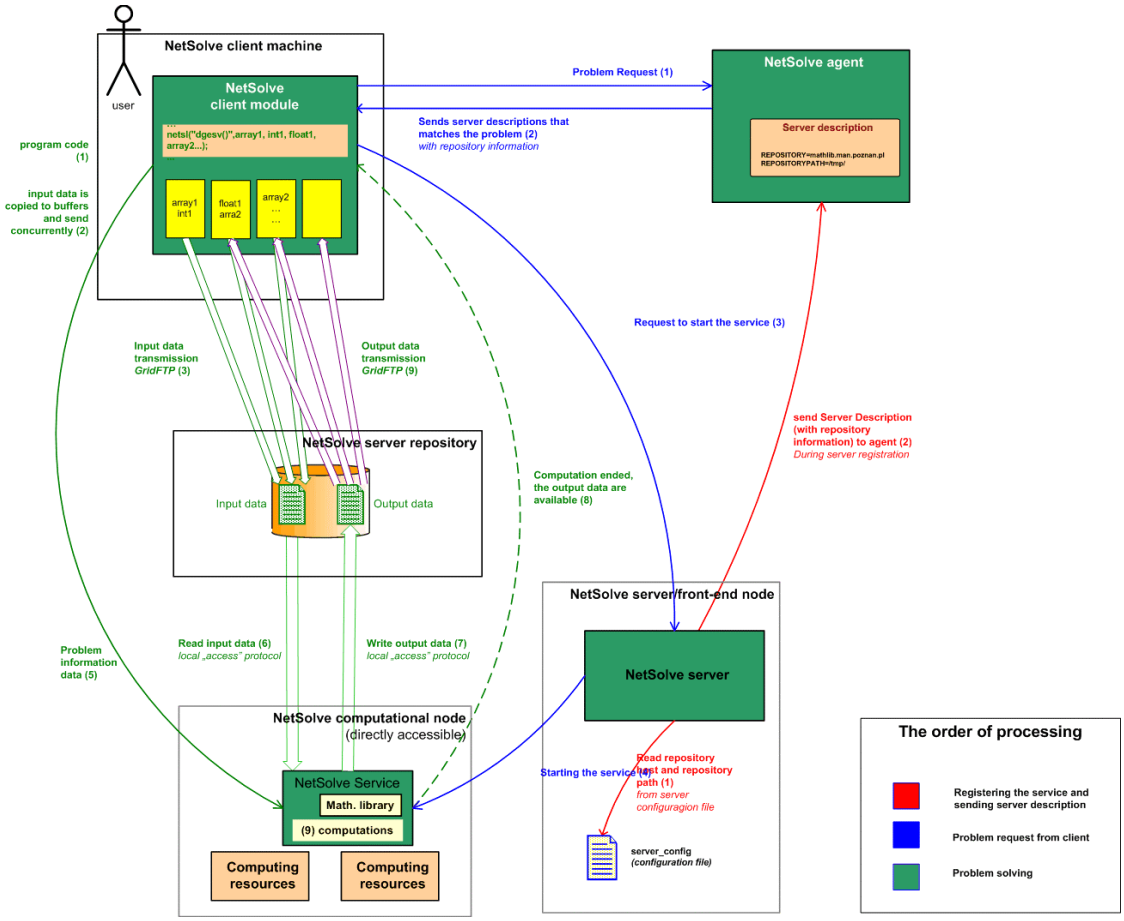
Przesyłanie danych poprzez protokół GridFTP odbywa się bezpośrednio z pamięci operacyjnej do pliku na zdalnym serwerze. Po stronie klienta, alokowane są bufory (ich liczba zależy od maksymalnej liczby równoległych połączeń), do których pakowane są poszczególne obiekty stanowiące parametry wejściowe lub wyjściowe. Wypełnianie bufora trwa do czasu aż nie jest on zapełniony – wówczas, natychmiast, zawartość bufora jest wysyłana do maszyny docelowej (dla opcji wysyłania danych do serwera). Bufor jest zwalniany, gdy klient otrzymuje powiadomienie o udanej operacji wysłania bufora. Może on być wówczas wykorzystany do kolejnego zapełnienia bufora nowymi danymi. Wypełnianie kolejnych buforów odbywa się równolegle, dopóki jakikolwiek bufor ma jeszcze wolne miejsce. Sytuacja podczas transferu wyników jest bardzo podobna. Klient inicjuje połączenie do serwera i wydaje polecenie przesłania pliku z wynikami. Plik ten jest przesyłany bezpośrednio do bufora pamięci po stronie klienta.

Oprócz funkcji API dostępnych w modułach globus-ftp-control i globus-ftp-client wykorzystujemy standardowe mechanizmy pakietu Globus jak: mechanizm wywołań zwrotnych (callback), semafory (mutex) oraz zmienne warunkowe (condition).

Podobnie jak w przypadku modułu komunikacyjnego korzystającego z GlobusIO dodano mechanizm wyświetlania i logowania informacji dotyczącej przesyłania danych protokołem GridFTP, przeznaczony dla programistów. Może być on wykorzystywany podczas prac nad testowaniem i debugowaniem kodu aplikacji (uaktywniany jest odpowiednią opcją podczas kompilacji).

Przesyłanie danych od/do klienta i serwisu odbywa się poprzez dedykowane repozytorium, którego adres i ścieżka gdzie dane będą zapisywane i odczytywane jest ustalana w standardowym pliku konfiguracyjnym systemu NetSolve (plik *server_config*). Dodano dwa nowe klucze: @REPOSITORY i @REPOSITORYPATH, które konfigurują repozytorium przypisane do danego serwera. Podczas rejestrowania serwera u agenta informacja o repozytorium jest przekazywana do agenta, który zapisuje ją w swoich tablicach opisujących dany serwer.

Na rysunku przedstawiono schemat przesyłania danych w systemie NetSolve przy użyciu biblioteki GridFTP. Pokazano na nim istotną część zmiennych, struktur i komunikatów, które są wykorzystywane w zmienionej wersji systemu.



Rysunek 8 Schemat przesyłania danych w systemie NetSolve z wykorzystaniem GridFTP

2.5. Uruchamianie zadań (GRAM)

W standardowym kodzie NetSolve'a, zadania uruchamiane są przez serwer, który wykorzystuje do tego funkcję systemową execv(). W architekturze systemu SGIgrid, w skład którego mogą wchodzić serwery obliczeniowe różnych typów, pracujące pod kontrolą różnych systemów, takie podejście jest niewystarczające. Poza tym zadanie uruchamiane jest jako proces systemowy a nie jako zadanie, które będzie podległe np. systemowi kolejkowemu wykorzystywane na danej jednostce obliczeniowej.

Dzięki wykorzystaniu modułu GRAM, w łatwy sposób stanie się możliwe uruchamianie zadań pochodzących z systemu NetSolve pod kontrolą lokalnych zarządców zasobów, odpowiednich dla danego systemu obliczeniowego. Moduł Globus GRAM oferuje wbudowane 'wtyczki' dla różnych systemów, m.in.: LSF, LoadLeveler, PBS, Condor.

W implementacji zdalnego uruchamiania zadań wykorzystamy moduł globus_gram_client z pakietu Globus'a 2.4.

Zupełnie inne podejście do uruchamiania/zlecania wykonania zadania spowoduje zmiany w protokole komunikacyjnym. Część komunikatów, które wysyłał serwis (proces obliczeniowy

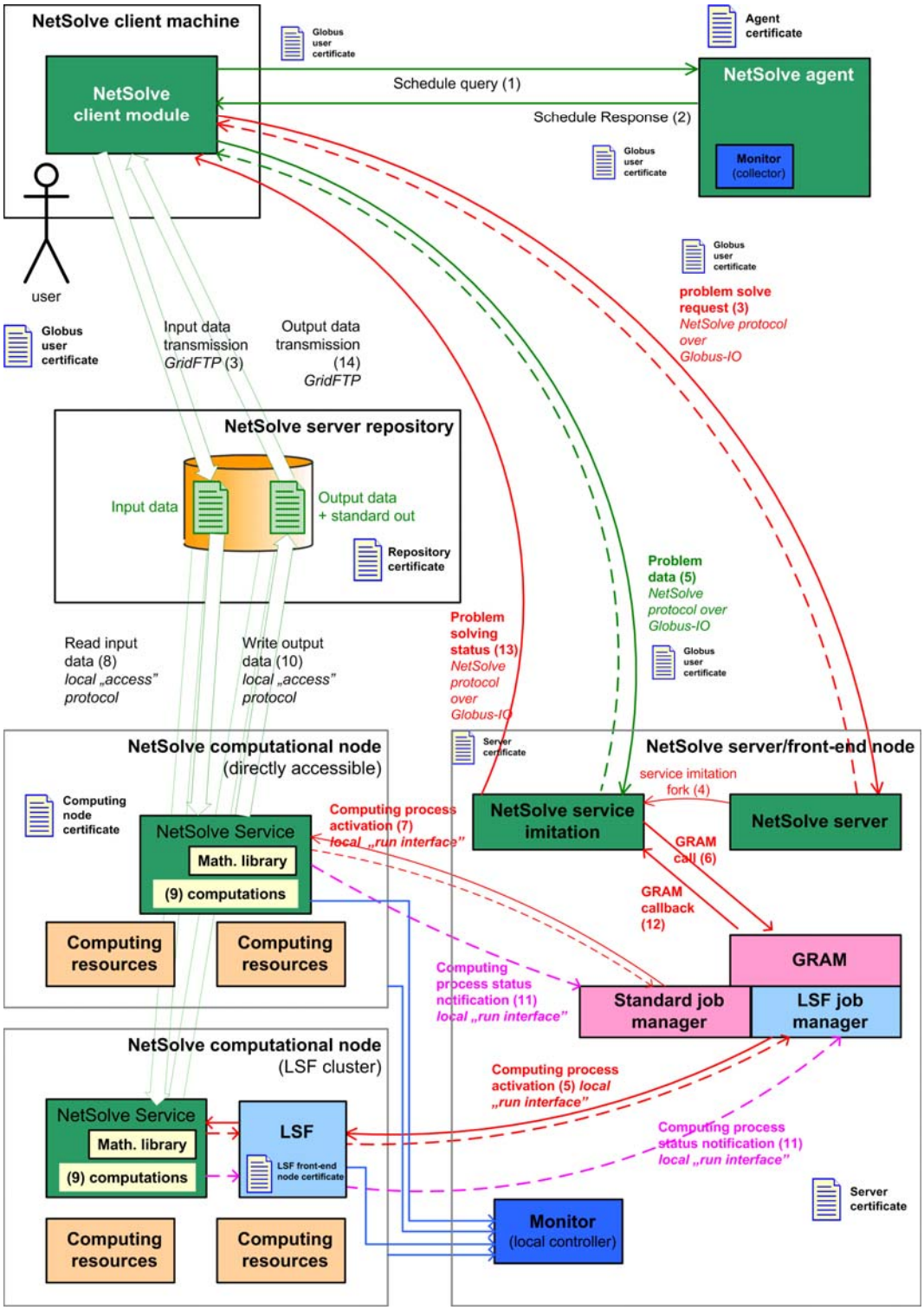
uruchamiany przez proces-serwer NetSolve) zostanie usunięta lub przeniesiona do modułu serwera. Również powiadamianie o stanie zadania, jego zakończeniu musi być zaimplementowane tak by odpowiadało wymogom modułu GRAM.

Uruchamianie zadań poprzez GRAM stanowi również podstawę do integracji mechanizmów rozwijanych w ramach zadania "Udostępniania dostępu do bibliotek naukowych" z mechanizmami opracowanymi w zadaniu "Zarządzania kontami użytkowników wraz z rozliczaniem zasobów" projektu SGI (VUS).

Schemat procesu przetwarzania zadania przy wykorzystaniu usługi SGIgrid przedstawia Rysunek 9. Aktywacji procesu obliczeniowego dokonuje moduł klienta NetSolve – podobnie jak w niezmodyfikowanej wersji programu NetSolve. Żądanie wysyłane jest do serwera, który uruchamia swój proces potomny, imitujący działanie serwisu. Klient nawiązuje połączenie z imitacją serwisu – i współpracuje z nim jak ze standardowym serwisem NetSolve. Imitacja uruchamia zadanie poprzez GRAM, czeka na wynik i zwraca status zadania klientowi. Dzięki takiemu podejściu, moduł klienta jak i protokół komunikacyjny NetSolve'a pozostał niezmieniony.

Komunikacja pomiędzy klientem i imitacją serwisu przebiega poprzez protokół GlobusIO. Komunikaty podpisane są certyfikatem cyfrowym użytkownika końcowego (czyli klienta). Zmodyfikowany serwer uruchamia usługę (serwis) GRAM podpisując żądanie certyfikatem klienta dzięki wykorzystaniu mechanizmu delegacji certyfikatów dostępnego w Globusie. Opis zadania przekazywany jest w języku RSL. Opis zawiera m.in.:

- nazwę serwisu-programu wykonującego obliczenia
- identyfikator globalny zadania (global_id)
- identyfikator żądania (request_id)
- lokalizację pliku z danymi wejściowymi dla zadania (lokalizacja w repozytorium)



Rysunek 9 Schemat przetwarzania zadań NetSolve przy użyciu usługi GRAM

Nowym elementem wykorzystywanym podczas przetwarzania jest *Repozytorium* (*NetSolve server repository*). Przed uruchomieniem zdalnego procesu obliczeniowego klient NetSolve umieszcza w repozytorium obiekty wejściowe dla obliczeń. Z repozytorium są one pobierane przez aktywowany później proces-serwis (NetSolve service). Po zakończeniu obliczeń serwis

umieszcza obiekty wynikowe w odpowiednim pliku w repozytorium. Moduł klienta, po odebraniu komunikatu o zakończeniu obliczeń pobiera dane z tego pliku.

W oryginalnym systemie NetSolve nie wyróżniano repozytorium jako oddzielnego obiektu. Dane wejściowe i wyjściowe obliczeń umieszczano na tym samym węźle, na którym znajdował się moduł serwera NetSolve. Wyróżnienie repozytorium jako oddzielnego obiektu zostało narzucone przez fakt, że obiekty wejściowe i wyjściowe muszą być dostępne dla procesu-serwisu, który może działać na innym niż serwer NetSolve i innym niż serwer GRAM systemie obliczeniowym. W opisie zadania przekazywanym przez klienta NetSolve serwerowi GRAM podana jest lokalizacja repozytorium (nazwa hosta, nr portu itd.) oraz ścieżka dostępu. Nazwy plików generowane są na podstawie identyfikatorów zadania i żądania (para ta jest unikalna dla danego zlecenia zdalnego wykonania funkcji)

2.6. Autentykacja i autoryzacja w zmodyfikowanym NetSolve

Jednym z podstawowych problemów oryginalnego systemu NetSolve były ograniczone możliwości autentykacji i autoryzacji użytkowników korzystających z systemu. Jedynym mechanizmem ograniczającym dostęp do zasobów NetSolve był tzw. *restriction index*, który ustalany był w plikach konfiguracyjnych serwera. Pozwalał on ograniczyć liczbę uruchamianych jednocześnie na danym serwerze zadań dla danego klienta lub grupy klientów, ew. zupełnie zablokować określonych klientów. Klienci identyfikowani byli na podstawie ich nazwy domenowej.

W zmodyfikowanym systemie NetSolve, stosowane są metody autentykacji i autoryzacji dostępne w implementacji Globusa, a oparte na certyfikatach wystawianych przez godne zaufania centrum. Certyfikaty są wystawiane z góry założonym czasem życia, dzięki czemu możemy udostępniać zasoby danemu klientowi na pewien czas.

Komunikacja w całym systemie (nie tylko pomiędzy klientem a systemem, ale również między poszczególnymi składnikami systemu) jest prowadzana przy użyciu Globus-IO poi uprzedniej autentykacji poszczególnych stron połączenia. Dodatkowo strony komunikacji autoryzują użytkownika związanego z danym certyfikatem poprzez sprawdzenie, czy jego Distinguished Name występuje w grid-mapfile. Nie

Dzięki funkcjonalności opartej na pliku grid-mapfile, prowadzana jest autoryzacja poszczególnych klientów podczas wykonywania zadań zlecanych poprzez usługę GRAM oraz podczas wymiany danych wejściowych i wyjściowych (GridFTP). Autoryzacja odbywa się przez sprawdzenie możliwości dopasowania tzw. Distinguished Name, dla którego wystawiony jest certyfikat użytkownika, do wpisu w pliku grid-mapfile. Zlecenia

uruchamiane przez GRAM oraz transmisje danych wejściowych i wyjściowych odbywają się z uprawnieniami tego użytkownika lokalnego danej maszyny, na którego, zgodnie z grid-mapfile, mapowany jest dany Distinguished Name.

3. Narzędzia testowe do weryfikacji poprawności integracji

Środowisko, do którego dostosowywany jest system udostępniania bibliotek matematycznych jest środowiskiem heterogenicznym. Pracujące w klastrze SGI komputery kontrolowane są przez systemy operacyjne Irix i Linux. Wykorzystywane na tych komputerach biblioteki matematyczne pochodzą z różnych źródeł (np. NAG Fortran Library na maszynach SGI [NAG], wersje bibliotek LAPACK i BLAS pochodzące z repozytorium oprogramowania NetLib [NETLIB] na części maszynach Linux'owych a także Intel Math Kernel Library na wybranych komputerach Linux'owych.

Prowadzone w ramach zadania WP.2.1.5 prace w znacznym stopniu zmieniają strukturę systemu i sposób działania systemu NetSolve. Zmiany te mają wpływ na wydajność przetwarzania żądań – obserwowane jest spowolnienie obsługi żądań w momencie z powodu przejścia z prostej, nieszyfrowanej i nieautoryzowanej komunikacji opartej o gniazdka systemu Unix na komunikacje z użyciem Globus IO. Z kolei zmiana techniki przesyłania danych wejściowych i wynikowych z pojedynczych strumieni danych na komunikację przez GridFTP powoduje przyspieszenie przesyłu danych podczas obliczeń w systemie rozległym geograficznie

3.1. Testy funkcji biblioteki LAPACK

Dla sprawdzenia poprawności działania zmienionego mechanizmu zdalnych obliczeń a także oceny wydajności i porównania z wydajnością w tradycyjnym systemie NetSolve PCSS opracowało modyfikację pakietu funkcji testujących służącego do testowania obliczeń z użyciem biblioteki matematycznej LAPACK.

Do przeprowadzenia testów środowiska wybrana została biblioteka LAPACK oferująca zestaw funkcji obliczających równania liniowe różnego typu. W implementacji wykorzystano zestaw programów testujących (dostępnych razem z biblioteką LAPACK) obliczających liniowe równania na różnego typu macierzach złożonych z liczb typu rzeczywistego.

Środowisko testowe składa się z trzech elementów:

- zmodyfikowanego zestawu programów testujących,
- narzędzi do prekompilacji kodu,

- zmodyfikowanego systemu NetSolve, oferującego dostęp do funkcji LAPACKa uruchamianych na serwerach obliczeniowych.

Programy testujące, napisane w języku FORTRAN, stanowią element wejściowy dla prekompilatora dostarczonego przez ośrodek TASK (WP.2.1.3). Otrzymany na wyjściu kod, w którym standardowe wywołania funkcji LAPACKa zostają zamienione na wywołania tychże funkcji dostępnych w systemie NetSolve, jest następnie kompilowany z biblioteką klienta systemu NetSolve i uruchamiany w systemie. Standardowe programy testowe, które są dostępne w bibliotece LAPACK, korzystają z generatora losowego macierzy. Wygenerowane macierze stanowią zmienne wejściowe potrzebne do testów. Aby zapobiec problemom związanym z interpretacją wyników a także aby testować dokładnie te same działania matematyczne przy zmieniających się innych parametrach systemu, do programów testowych zostały dodane funkcje pozwalające na zapisanie wygenerowanych macierzy do pliku a następnie ich odczytanie z plików (podczas kolejnych uruchomień procedury testowej).

3.2. Instalacja testowa a wdrożenie

Testy zmodyfikowanego systemu NetSolve przeprowadzone były w instalacji testowej obejmującej dwie lokalizacje: Poznań i Gdańsk. Instalacja testowa została omówiona w raporcie końcowym zadania WP.2.1.1. Testy wykazały poprawne działanie mechanizmów integracji NetSolve z usługami systemu Globus. Na etapie wdrożenia projektu celowego prowadzone będą dalsze testy zmodyfikowanego systemu NetSolve w środowisku klastra.

4. Podsumowanie

Prace integracyjne w ramach podzadania WP.2.1.5 zostały zakończone. Zaistniałe opóźnienie (przesunięcie prac do 24. miesiąca projektu) spowodowane jest decyzją o zmianie architektury systemu podjętą przez konsorcjum realizujące projekt podczas jego trwania.

Wprowadzone zmiany są uzasadnione ważnymi przesłankami technicznymi. Powstały produkt jest bardziej niż w pierwotnej, planowanej wersji dostosowany do wymagań środowiska SGIgrid i zgodny ze standardami budowy współczesnych środowisk przetwarzania rozproszonego. Wymagało to niestety dodatkowych nakładów pracy i przez to przesunięcia terminu zakończenia zadania.

Niniejszy raport jest uzupełnieniem raportu z miesiąca 12 projektu. Dokumentuje on szczegóły mechanizmów integracji systemu udostępniania bibliotek matematycznych z innymi elementami klastra SGIgrid. Dokumentacja użytkownika i administratora systemu

opublikowana została wraz z pakietem oprogramowania dostępnym na stronie www zadania pod adresem: mathlib.psnc.pl.

Bibliografia

[GLOBUS] Globus: A Metacomputing Infrastructure Toolkit. I. Foster, C. Kesselman. Intl J. Supercomputer Applications, 11(2):115-128, 1997.
www.globus.org

[LSF] Load Sharing Facility. http://www.platform.com/products/LSFfamily

[NINF] M. Sato, H. Nakada, S. Sekiguchi, S. Matsuoka, U. Nagashima, H. Takagi: Ninf: A Network based Information Library for a Global World-Wide Computing Infrastracture, HPCN'97 (LNCS-1225), pp. 491-502", 1997"
(http://ninf.apgrid.org/papers/hpcn97/hpcn97-paper.pdf)

[NETLIB] www.netlib.org

[NETSOLVE] Network-Enabled Solvers and the NetSolve Project. H. Casanova, J.J. Dongarra, and K. Moore, SIAM News, January, 1998, Vol 31, No. 1.
Strona domowa projektu NetSolve. http://icl.cs.utk.edu/netsolve

[NETSNEWS] Informacje o nowościach w projekcie NetSolve. Strona domowa projektu NetSolve. http://icl.cs.utk.edu/netsolve/news/index.html

[NINF] Ninf: A Network based Information Library for a Global World-Wide Computing Infrastracture. Mitsuhisa Sato, Hidemoto Nakada, Satoshi Sekiguchi, Satoshi Matsuoka, Umpei Nagashima and Hiromitsu Takagi. HPCN'97 (LNCS-1225), pp. 491-502, 1997.

[SGIREPORT1] Raport projektu celowego. Obliczenia wielkiej skali i wizualizacja do zastosowań w wirtualnym laboratorium z użyciem klastra SGI. Zadanie WP 2.1. Zdalny dostęp do bibliotek naukowych. Wykorzystanie bibliotek matematycznych w Polsce.
http://mathlib.psnc.pl/documents.html

[SGIREPORT2] Raport projektu celowego. Obliczenia wielkiej skali i wizualizacja do zastosowań w wirtualnym laboratorium z użyciem klastra SGI. Zadanie WP 2.1. Zdalny dostęp do bibliotek naukowych. Przegląd cech i funkcjonalności systemów udostępniania bibliotek matematycznych, raport z testów i analizy kodów źródłowych, raport z wykonania instalacji testowej.
http://mathlib.psnc.pl/documents.html