

# **Obiektowe i agentowe strategie przetwarzania wielkich zadań obliczeniowych w sieciach komputerowych**

**Marek Grochowski**

**Opiekun: prof. dr hab. inż. Robert Schaefer**

# Plan pracy doktorskiej

## Teza:

Możliwe jest stworzenie strategii dla rozproszonych środowisk obiektowych i środowisk wieloagentowych na potrzeby obliczeń wielkiej skali spełniającej poniższe wymagania:

1. efektywne zagospodarowanie zasobów zmiennego w czasie heterogenicznego środowiska
2. małe narzuty na zarządzanie i komunikację dzięki zastosowaniu zdecentralizowanych mechanizmów sterowania migracją i dekompozycją
3. skrócenie czasu projektowania i realizacji procesu obliczeniowego (w porównaniu do rozwiązań niskopoziomowych) poprzez wprowadzenie mechanizmów częściowo zautomatyzowanego podziału zadania, jak i tworzenia oraz alokacji jednostek obliczeniowych
4. łatwa współpraca z innymi systemami obliczeń rozproszonych jak i platformami wieloagentowymi

## Przewidywany spis rozdziałów

1. Motywacja podjętej tematyki.
2. Taksonomia problemów związanych z przetwarzaniem rozproszonym.
3. Postawienie problemu.
4. Referencyjne rozwiązania niskopoziomowe.
  - 4.1. Architektury i aplikacje oparte na bibliotekach komunikacyjnych.
  - 4.2. Wydajność, narzuty, skalowalność, szeregowanie.
  - 4.3. Wyniki testów dla SBS-PCG oparte na PVM oraz MPVM w homogenicznej sieci.
5. Rozwiązania obiektowe.
  - 5.1. Architektury i aplikacje oparte o zdalne obiekty zaalokowane statycznie, wykorzystujące obiektowe biblioteki komunikacyjne oraz architektury zorientowane serwisowo.
  - 5.2. Wydajność, narzuty, skalowalność, szeregowanie.
  - 5.3. Wyniki testów dla generatora siatek 3D.
6. Rozwiązania wieloagentowe.
  - 6.1. Architektury platformy i agentów.
    - 6.1.1. rozproszone struktury danych:
      - opis topologii serwerów

- graf podziału agentów
  - serwisy platformy
- 6.1.2. polityki:
- informacyjna
  - migracyjna
  - bezpieczeństwa
- 6.2. Algorytmy.
- 6.2.1. szeregowanie dyfuzyjne
- 6.2.2. transakcyjna migracja oraz komunikacja
- 6.2.3. sterowanie ziarnem
- 6.2.4. miękka synchronizacja
- 6.3. Architektura agenta Smart Solid.
- 6.4. Case study.
- 6.4.1. CAE metoda bezsiatkowa (regularna równoległość)
- dekompozycja w oparciu o opis grafowy dla gramatyk ETPL(k)
  - rozproszona generacja niestrukturalnej siatki 3D typu Delaunay'a
  - solver SBS-PCG
- 6.4.2. HGS (nieregularna równoległość)
- wielopopulacyjny genetyczny algorytm przybliżonego przeszukiwania zbiorów ciągłych
  - modele w pełni synchronizowane
  - modele z miękka synchronizacją wykorzystanie agenta „Leo the professional”
7. Testy.
- 7.1. wydajność vs. ziarno
- 7.2. narzuty technologii wieloagentowej
- 7.3. skuteczność szeregowania
- 7.4. stopień wykorzystania zasobów
- 7.5. skuteczność miękkiej synchronizacji
8. Plan dalszych badań.
- 8.1. Rozwój modelu matematycznego dyfundującego systemu wieloagentowego. Badanie asymptotyki.

- 8.2. Nowe koncepcje algorytmiczne zarządzania podziałem i migracją agentów.
- 8.3. Integracja z technologiami GRID.
  - 8.3.1. serwisy w architekturze OGSA
  - 8.3.2. rozszerzenie modelu szeregowania w celu umożliwienia agentom migracji pomiędzy wieloma lokalnymi sieciami
- 8.4. Rozszerzenie implementacji architektury Smart Solid o wsparcie przez inne platformy wieloagentowe.
- 9. Podsumowanie i wnioski.

## Skrócony opis zawartości rozdziałów

### 1. Motywacja podjętej tematyki.

Przetwarzanie dużych problemów obliczeniowych wymaga wykorzystania odpowiednio dużej mocy obliczeniowych, które mogą być zaspokojone przez maszyny równoległe bądź przez platformy wirtualne zbudowane w oparciu o wiele komputerów klasy PC czy stacji roboczych połączonych siecią komputerową. Platformy wirtualne są rozwiązaniem znacznie tańszym w porównaniu do komputerów równoległych o architekturze zwartej (UMA, NUMA). Platformy takie mogą być również tworzone na bazie komputerów znajdujących się w niemal każdej lokalnej sieci komputerowej.

Obecnie dostępnych jest wiele różnych platform zaprojektowanych z myślą o przeprowadzaniu obliczeń rozproszonych. Niemal wszystkie platformy wymagają ręcznej konfiguracji komputerów oraz wspierają jedynie dedykowane aplikacje, tzn. takie, które zostały zaimplementowane dla konkretnej platformy.

W pracy zaproponowano wykorzystanie platformy wspierającej systemy wieloagentowe oraz zbudowanie agenta nazwanego Smart Solid, którego celem jest wspieranie dużych zadań obliczeniowych poprzez poszukiwanie wymaganej mocy obliczeniowych w sieci komputerowej. Zadanie obliczeniowe, jako aplikacja wieloagentowa zbudowana z agentów Smart Solid, może być uruchamiana na wielu różnych platformach wieloagentowych bez konieczności dokonywania zmian takiego zadania.

### 2. Taksonomia problemów związanych z przetwarzaniem rozproszonym.

Rozważana jest charakterystyka problemów istotnych dla obliczeń rozproszonych dużej skali przeprowadzanych w sieci komputerów. W szczególności są to zagadnienia związane ze skalowalnością, rodzajem równoległości, sterowaniem obliczeniami, czy ziarnistością w przetwarzanych zadaniach. Istotnym zagadnieniem w tych rozważaniach jest decentralizacja sterowania oraz rozproszone struktury danych.

Taksonomia zespołu omawianych problemów jest przedstawiona wraz z wynikami innych autorów.

### 3. Postawienie problemu i teza pracy

Celem pracy jest znalezienie jak najlepszej strategii na tworzenie aplikacji i systemów przeprowadzających obliczenia rozproszone w sieci komputerów. Aby wprowadzić czytelnika do omawianego zagadnienia charakteryzowane są w skrócie podstawowe techniki konstruowania aplikacji oraz systemów liczących począwszy od niskopoziomowego programowania wykorzystującego komunikację przez wymianę komunikatów, poprzez rozwiązania obiektowe wykorzystujące architektury zorientowane serwisowo, kończąc na paradygmacie systemu wieloagentowego. Rozważane są problemy oraz korzyści występujące przy projektowaniu oraz użytkowaniu aplikacji oraz systemów liczących tworzonych w oparciu o wymienione

techniki, w szczególności związane z wydajnością, skalowalnością, szeregowaniem, łatwością instalacji i utrzymywania takich aplikacji oraz systemów.

Tworzona strategia dla rozproszonych środowisk obiektowych oraz środowisk wieloagentowych musi spełniać następujące wymagania:

1. efektywne zagospodarowanie zasobów zmiennego w czasie heterogenicznego środowiska
2. małe narzuty na zarządzanie i komunikację dzięki zastosowaniu zdecentralizowanych mechanizmów sterowania migracją i dekompozycją
3. skrócenie czasu projektowania i realizacji procesu obliczeniowego (w porównaniu do rozwiązań niskopoziomowych) poprzez wprowadzenie mechanizmów częściowo zautomatyzowanego podziału zadania, jak i tworzenia oraz alokacji jednostek obliczeniowych
4. łatwa współpraca z innymi systemami obliczeń rozproszonych jak i platformami wieloagentowymi

Na podstawie tych rozważań formułowana jest teza pracy.

Weryfikacja postawionej tezy dokonywana jest doświadczalnie poprzez analizy eksperymentów wykonanych dla dwóch wybranych problemów case study charakteryzujących się różnym rodzajem równoległości (patrz rozdz.6.4). Analizy wyników są nastawione na sprawdzenie stopnia zagospodarowania zasobów, narzutów czasowych na sterowanie i komunikację w porównaniu do czasu obliczeń, oraz skuteczności rozpraszania w zależności od wielkości problemów jak i od liczby użytych komputerów.

Dokonywane jest także porównanie eksperymentów przeprowadzanych z systemami liczącymi wykonanymi w różnych architekturach, lecz rozwiązujących takie same problemy CAE.

#### 4. Referencyjne rozwiązania niskopoziomowe.

Punktem odniesienia dla zagadnień omawianych w pracy są aplikacje rozproszone zrealizowane w technice programowania niskopoziomowego wykorzystujące komunikację realizowaną przez biblioteki PVM oraz MPVM. Wybrane jest zadanie typu CAE rozwiązywane przy użyciu solwera SBS-PCG, który jest implementowany później w innych architekturach, i jest ono traktowane jako problem odniesienia umożliwiające porównanie uzyskanych wyników oraz porównanie parametrów związanych ze skutecznością zastosowanej architektury jak i środowiska, w którym przeprowadzane są obliczenia.

Przedstawione są wyniki testów dla zadania CAE rozwiązywanego przy użyciu solwera SBS-PCG, wraz z ich analizą, przeprowadzone w homogenicznym środowisku złożonym z komputerów w architekturze RISC pracujących pod kontrolą systemu operacyjnego SunOS.

#### 5. Rozwiązania obiektowe.

Opis koncepcji architektur obiektowych wykorzystujących zaalokowane statycznie obiekty z komunikacją w architekturze SOA (Service Oriented

Architecture) w technologii CORBA oraz Jini, komunikujące się ze sobą także za pośrednictwem RMI (Remote Method Invocation). Przedstawione są wyniki testów, wraz z ich analizą, dla rozproszonego generatora trójwymiarowych niestrukturalnych siatek obliczeniowych spełniających warunek Delaunay'a. Testy przeprowadzone były w heterogenicznym środowisku złożonym z komputerów w architekturze Intel oraz MIPS-SGI pracujące pod kontrolą następujących systemów operacyjnych: Linux, IRIX oraz MS Windows.

## 6. Rozwiązania wieloagentowe.

Wykorzystanie platformy wieloagentowej do zbudowania rozproszonej aplikacji obliczeniowej jest w znacznym stopniu ułatwione dzięki dostępnym w takiej platformie mechanizmom rozpraszania obiektów, zarządzania komunikacją czy alokacją agentów. Jeśli dodatkowo do systemu wieloagentowego dostarczone zostaną algorytmy poszukujące zasobów oraz sterujące obciążeniem poprzez podział i migrację agentów to zadanie zbudowania rozproszonego systemu obliczeniowego znacznie się upraszcza. Ponadto zarządzanie takim systemem jak i jego podtrzymywanie staje się czynnością wymagającą znacznie mniej czasu niż to ma miejsce podczas stosowania rozwiązań obiektowych czy niskopoziomowych.

Rozdział ten zawiera także formalny opis zaproponowanego modelu.

### 6.1. Architektury platformy i agentów.

Opisane są architektury platform wspierających systemy wieloagentowe oraz architektury agentów. W szczególności opisane są architektury wspierające zrealizowanie postawionej tezy.

Zaproponowane są struktury danych opisujące stan aplikacji wieloagentowej jak i stan serwerów platformy w sposób rozproszony i zdecentralizowany.

### 6.2. Algorytmy.

Istotnym elementem ułatwiającym tworzenie rozproszonych aplikacji obliczeniowych w systemach wieloagentowych są algorytmy niezbędne dla rozważanych obliczeń rozproszonych takie jak szeregowanie, algorytmy realizujące komunikację pomiędzy migrującymi agentami czy algorytmy sterowania ziarnem.

Przedstawione są własne propozycje poszczególnych algorytmów.

Przykładem szeregowania jest algorytm szeregowania dyfuzyjnego, który został zaprojektowany jako analogia do dyfuzji atomów w sieci krystalicznej. Przez analogię, zdefiniowane zostały podstawowe parametry dyfuzji takie jak energia wiązania czy gradient koncentracji z tym, że dla agentów dyfundujących w sieci komputerów. Reguły dyfuzyjne są wykonywane niezależnie przez każdego z agentów i w taki sam sposób, bez konieczności synchronizacji całego liczącego systemu wieloagentowego. Każdy agent, bezpośrednio po podziale oraz po migracji, odpytuje jedynie serwery znajdujące się w lokalnym otoczeniu po to by wyznaczyć swoje parametry dyfuzyjne, następnie stosując prawo dyfuzji decyduje czy ma pozostać na bieżącym komputerze, czy migrować i na który serwer. Algorytm ten nie wymaga

centralnego sterowania ani centralnej synchronizacji, oraz szybko prowadzi do stanu równowagi w całej sieci komputerów.

### 6.3. Architektura agenta Smart Solid.

Agent Smart Solid został zaprojektowany w celu zrealizowania oczekiwanych i opisywanych wcześniej wymagań. Architektura agenta Smart Solid pozwala na wykorzystanie dostarczonych algorytmów jak też dołączenie dedykowanych implementacji innych algorytmów dla aplikacji realizujących obliczenia rozproszone.

Istotną cechą architektury agenta Smart Solid jest zapewnienie separacji na dwóch poziomach. Na pierwszym z nich odseparowana jest logika aplikacji wieloagentowej od właściwego zadania obliczeniowego po to by:

- a) można było w łatwy sposób wykorzystać cechy systemu wieloagentowego do automatycznego alokowania zadań obliczeniowych,
- b) można było dokonywać zmian w kodzie zadania obliczeniowego bez ingerencji w konstrukcję agenta.

Na drugim poziomie, dzięki odseparowaniu logiki agenta od specyfiki wykorzystywanej platformy wieloagentowej, architektura agenta Smart Solid pozwala także na uruchomienie takiego agenta na różnych platformach wieloagentowych. Separacja ta umożliwiła podmiannę implementacji wspierającej agenta na jednej platformie na implementację wspierającą agenta na innej platformie.

### 6.4. Case study.

Jako case study wybrane zostały dwa różne problemy charakteryzujące się równoległością regularną (rozproszony system CAE) oraz nieregularną, stochastyczną równoległością (HGS – genetyczny algorytm optymalizacji globalnej).

## 7. Testy.

W celu zbadania i udowodnienia skuteczności zaproponowanych architektur oraz algorytmów przeprowadzonych zostało szereg testów porównawczych.

Przeprowadzono następujące analizy:

- wydajność vs. ziarno dla wieloagentowych aplikacji CAE oraz HGS,
- oszacowanie narzutów czasowych związanych ze sterowaniem i komunikacją dla wieloagentowych aplikacji CAE oraz HGS,
- określenie skuteczności szeregowania dyfuzyjnego – porównanie z szeregowaniem typu Round-Robin w nieobciążonym dedykowanym środowisku dla aplikacji HGS,
- stopień wykorzystania zasobów dla wieloagentowych aplikacji CAE oraz HGS,
- określenie skuteczności miękkiej synchronizacji dla aplikacji HGS.

Przeprowadzone eksperymenty wykazały, że narzuty wynikające z zastosowania systemu wieloagentowego wraz z szeregowaniem dyfuzyjnym wynoszą ok. 5% czasu obliczeń i mają coraz mniejszą wartość dla coraz większych zadań obliczeniowych.



Skuteczność szeregowania dyfuzyjnego została potwierdzona w testach dla aplikacji HGS, w których skuteczność (speedup) szeregowania dyfuzyjnego została porównana ze skutecznością szeregowania typu Round-Robin dla nieobciążonej sieci komputerów.

Równomierne wykorzystanie zasobów (well balanced load), w przeprowadzonych eksperymentach z zastosowaniem szeregowania dyfuzyjnego, było szybko osiągalne.

#### 8. Plan dalszych badań.

Prowadzimy prace nad rozwojem modelu matematycznego dyfundującego systemu w celu umożliwienia badania asymptotyki takiego systemu.

Planowane są prace nad rozszerzeniem strategii szeregowania dyfuzyjnego (oraz modelu dyfundującego systemu) w celu umożliwienia migracji agentów, w sytuacji wysycenia zasobów sieci lokalnej, pomiędzy wieloma sieciami tworzącymi oddalone środowiska obliczeniowe.

Niezbędne jest również rozszerzenie opisu topologii połączeń na wiele platform wieloagentowych i zaprojektowanie oraz implementacja serwisów w architekturze OGSA (Open GRID Services Architecture) w celu wykorzystania idei GRIDu do udostępniania aplikacji wykonanych w oparciu o agenta Smart Solid pomiędzy różnymi platformami wieloagentowymi (np. IBM Aglets, ADAJ, Java Party, NOMADS) i umożliwienie migracji agentów za pośrednictwem Internetu pomiędzy wieloma oddalonymi sieciami lokalnymi.

## Obecny stan zaawansowania

Większość prac implementacyjnych jest wykonana jak też większość eksperymentów oraz analiz została przeprowadzona. Praca wymaga jeszcze szczegółowego opisanie rozważanych rozwiązań oraz taksonomii poruszanych problemów, syntetycznego opisu proponowanych strategii, oraz doprecyzowania planu dalszych badań.