

PIOTR FILIPKOWSKI

Kolegium Analiz Ekonomicznych
Szkoła Główna Handlowa w Warszawie

MICHAŁ HORODELSKI

Wydział Matematyki, Informatyki i Architektury Krajobrazu
Katolicki Uniwersytet Lubelski Jana Pawła II w Lublinie

Zastosowanie rozproszonej platformy modelowania i symulacji do analizy informacji medycznych na bieżąco

1. Wstęp

Ratownictwo medyczne na poziomie przedszpitalnym obejmuje zespół działań mających na celu ratowanie życia i zdrowia ludzi podczas zdarzeń, w których występuje zagrożenie utraty życia lub pogorszenia się stanu zdrowia. Jego celem jest zapewnienie skutecznych medycznych działań ratowniczych wobec każdej osoby znajdującej się w stanie nagłego zagrożenia. Osiągnięcie tak sformułowanego celu wymaga wykonywania zadań związanych z przetwarzaniem informacji i wiedzy, odpowiednich do rodzaju zagrożeń życia lub zdrowia, polegających m.in. na zapewnieniu możliwości natychmiastowego wezwania pomocy przy użyciu środków łączności, niezwłocznego podjęcia właściwych decyzji medycznych i działań ratowniczych na miejscu zdarzenia. Efektywność i skuteczność tych działań zależy od potencjałów informacyjnych i decyzyjnych systemu ratownictwa medycznego na poziomie jednostek.

W dobie rozwoju technik informacyjnych współczesnemu społeczeństwu należy nie tylko dostarczać zaawansowanych narzędzi i technologii w obszarze

funkcjonowania społeczeństwa informacyjnego (SI), ale też zapewnić na odpowiednim poziomie medyczne wsparcie decyzyjne i pokrewnych obszarów e-przestrzeni (ang. *e-space*) decydenta (inteligentnego agenta), którym może być ratownik-lekarz.

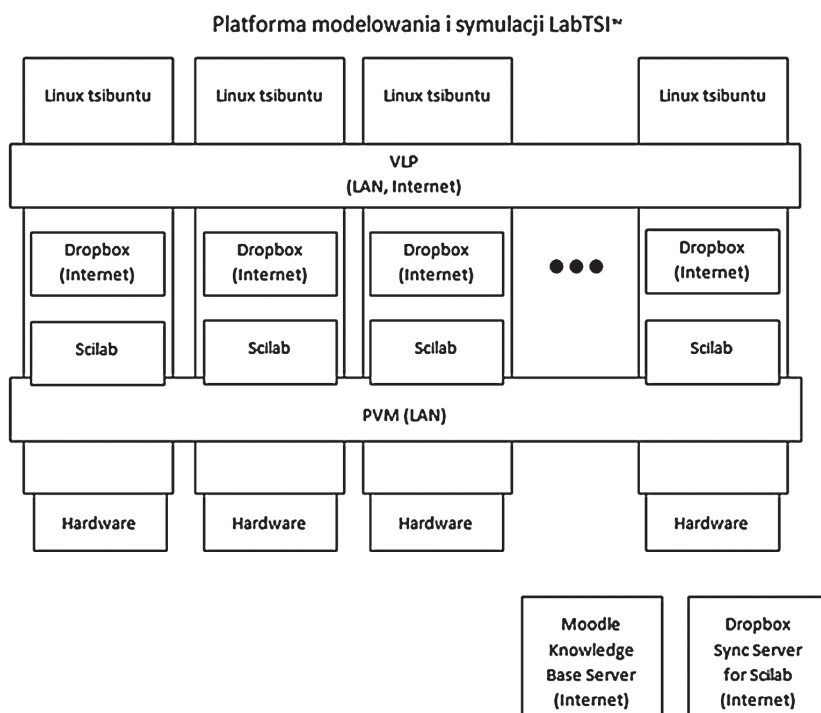
Celem badawczym niniejszej pracy jest poznanie możliwości obliczeniowych wybranej platformy modelowania i symulacji do mobilnego wspomagania analizy obrazów medycznych w warunkach stresu. Jako platformę modelowania i symulacji autorzy rozumieją informatyczny system działających aplikacji, udostępniający wybrane usługi, mający własność zwiększania wydajności działania tych usług dzięki mobilności, elastyczności oraz skalowalności jej mocy obliczeniowej. Informacją medyczną może być zbiór informacji z urządzeń medycznych, w tym każdy obraz wykonany przy pomocy technik medycznych na pacjencie, np. zdjęcie lub prześwietlenie części ciała. Warunki stresu autorzy utożsamiają z sytuacją, w której osoba podejmująca decyzję musi szybko postawić wstępną diagnozę dotyczącą urazu poszkodowanego na podstawie wiedzy oraz analizy informacji medycznych, jakie posiada – bez bezpośredniego dostępu do zasobów specjalistycznego laboratorium.

2. Wybrana platforma modelowania i symulacji

Wykorzystywana Platforma Modelowania i Symulacji LabTSI® (w skr. PMiSL) jest systemem wspólnie działających podsystemów, a więc cechuje ją modułowość i czytelność struktury, dzięki której jest m.in. możliwe dostarczenie środowiska rozproszonego do przyspieszania wykonywania złożonych obliczeń. Podsystemy platformy odpowiadają za obsługę urządzeń heterogenicznych, komunikację, rozpraszanie obliczeń oraz szyfrowanie. Obecnie platforma ta rozwijana jest przez Europejskie Centrum Technologii Społeczeństwa Informacyjnego, a pierwsze prace nad tą platformą miały miejsce w Laboratorium LabTSI® w Katedrze Technologii Społeczeństwa Informacyjnego Katolickiego Uniwersytetu Lubelskiego Jana Pawła II.

Nie trudno zauważyć, że złożoność informacyjna świata jest coraz większa i powiększa się z każdym dniem. Dobrze rozumieją to osoby trudniące się modelowaniem komputerowym. Fakt coraz większej ilości informacji oraz rosnące oczekiwania co do odwzorowania rzeczywistości utrudnia modelowanie zjawisk i procesów otaczającego nas świata ze względu na pojawiającą się znaczącą złożoność obliczeniową. Jeżeli takie dokładne modele interesujących nas zjawisk będą przetwarzane w sposób sekwencyjny, to czas ich symulacji może być bardzo długi.

PMiSL jest wynikiem synergii jej składowych modułów programowych. Podstawowym elementem owej platformy jest system operacyjny tsibuntu¹ (obecna wersja to tsibuntu 2013 beta). Na rysunku 1 przedstawiono schemat pierwszej realizacji PMiSL, która została stworzona do pracy wyłącznie w sieci LAN. Platforma może być zainstalowana w infrastrukturze sprzętowej bez ingerencji w system operacyjny znajdujący się na danym węźle infrastruktury. Co więcej, obecne opracowanie rozszerzenia PMiSL o topologię gwiazdy w sieci WAN umożliwia jej bezpieczną pracę dzięki wykorzystaniu nowego modułu realizującego połączenia w wirtualnej sieci prywatnej (VPN). Pozwala to na pokonanie fizycznych barier przy instalacji platformy w infrastrukturze sprzętowej na węzłach jednej sieci LAN wyłącznie w jednym miejscu. Możliwe są więc również mobilne połączenia węzłów sieci, przy zachowaniu najwyższych standardów bezpieczeństwa pracy oraz certyfikowanych i szyfrowanych połączeń.



Rysunek 1. Platforma modelowania i symulacji – LabTSI®

Źródło: A. Janicki, *Platforma modelowania i symulacji LabTSI™*, Wydawnictwo KUL, Lublin 2011, s. 10.

¹ A. Janicki, *Platforma modelowania i symulacji LabTSI™*, Wydawnictwo KUL, Lublin 2011, s. 9–13.

Modułem dostarczającym środowisko programowania jest narzędzie Scilab, open-source'owa alternatywa dla rozwiązań jak np. Matlab, znanych w środowisku naukowców i inżynierów. Dzięki temu narzędziu użytkownik platformy ma dostęp do intuicyjnego języka skryptowego, który można wykorzystać do implementacji nowo opracowanych pomysłów, modeli procesów decyzyjnych, inżynierskich, medycznych oraz wielu innych². Dostęp do konsoli pozwala w szybki sposób przypomnieć sobie składnię języka Scilab i testować fragmenty kodu bez potrzeby pracy w trybie debugowania. Scilab umożliwia także programowanie przy wsparciu języków Fortran i C oraz tworzenie rozszerzeń w postaci ToolBox, zamykających profesjonalnie zbiory opracowań, bibliotek czy implementacji.

Modułem synchronizującym dane platformy jest aplikacja Dropbox, dzięki której dane symulacji możemy przechowywać w jednym miejscu w sieci WAN oraz mieć do nich dostęp z każdego węzła platformy. Jest to realizacja wymaganej pamięci rozproszonej przy tego typu rozwiązaniach systemowych. Rozwiązanie Dropbox jest wygodne, ale niekoniecznie darmowe. W przypadku wyższych standardów bezpieczeństwa (np. wojskowych) wymiana danych odbywa się drogą niezależną od rozwiązań firm trzecich.

Moduł dostępu do zdalnego pulpitu dla każdego węzła jest realizowany przy pomocy darmowej wersji rozwiązania komercyjnego TeamViewer lub open-source'owego programu ThinVNC w przypadku pracy w sieciach o wyższych standardach bezpieczeństwa. Dzięki temu modułowi platforma, po przeprowadzeniu odpowiednich bezpiecznych procedur dostępowych, może być zarządzana z każdego węzła rozproszonego systemu na zasadach zdalnego pulpitu. Dostęp taki jest wygodniejszy dla osób mniej zaznajomionych z pracą w konsoli oraz umożliwia obserwację wizualnych efektów działania algorytmu rozpraszającego obliczenia.

Newralgicznym modułem realizującym rozpraszanie obliczeń jest wirtualna maszyna równoległa (ang. *Parallel Virtual Machine* – PVM). Dzięki niej przy wykorzystaniu zbioru bibliotek procedur PVM użytkownik platformy może programować aplikacje m.in. w języku skryptowym w sposób rozproszony. Dzięki temu rozwiązanie problemu złożoności obliczeniowej na tej platformie jest realizowane przez zrównoleglenie czasochłonnych realizacji modeli i przyspieszanie obliczeń.

PMiSL była już z powodzeniem wykorzystywana w obszarze medycznym do modelowania zachowań Państwowego Ratownictwa Medycznego za pomocą modelu systemu agentowego, co skłania autorów do szukania kolejnych zastosowań w tymże obszarze³.

² <http://atoms.scilab.org> (data odczytu 22.11.2013).

³ P. Filipkowski, M. Janicki, D. Oleszczuk, *Agent Technologies in LabTSI™ – Use in medical rescue*, „The Polish Journal of Aviation Medicine and Psychology” 2012, vol. 3 (18),

Podsumowując, należy stwierdzić, że PMiSL jest kompleksową i sprawdzoną platformą modelowania i symulacji działającą w rozproszonym środowisku, zapewniającą przy tym bezpieczne przetwarzanie danych oraz udostępniającą usługi z zakresu przetwarzania informacji, modelowania złożonych procesów, rozpraszania i przyspieszania obliczeń oraz wdrażania inteligentnych rozwiązań.

3. Narzędzia badawcze

Idea **PVM** opiera się na pojęciu maszyny wirtualnej jako pewnej dodatkowej warstwy oprogramowania, w ramach której poszczególne komputery, podłączone do systemu równoległego, nazywane węzłami, są postrzegane jako jednostki przetwarzające, a cała maszyna wirtualna – jako komputer równoległy z pamięcią rozproszoną⁴. PVM dostarcza zestawu narzędzi, dzięki którym wytwarzanie oprogramowania wykorzystującego programowanie rozproszone jest o wiele prostsze. Przetwarzanie w PVM polega na wykonaniu programu równoległego na węzłach maszyny wirtualnej. Jednostką atomową w obliczeniach PVM jest proces. Dzięki odpowiedniej funkcji PVM programy są dzielone na procesy i rozprowadzane na poszczególne węzły maszyny wirtualnej. Podstawową funkcją środowiska PVM jest dostarczanie mechanizmów rozpraszania procesów, ponadto zapewnienie komunikacji, synchronizacji i identyfikacji zadań niezależnie od ich fizycznej lokalizacji. PVM zapewnia także możliwość dynamicznej zmiany konfiguracji.

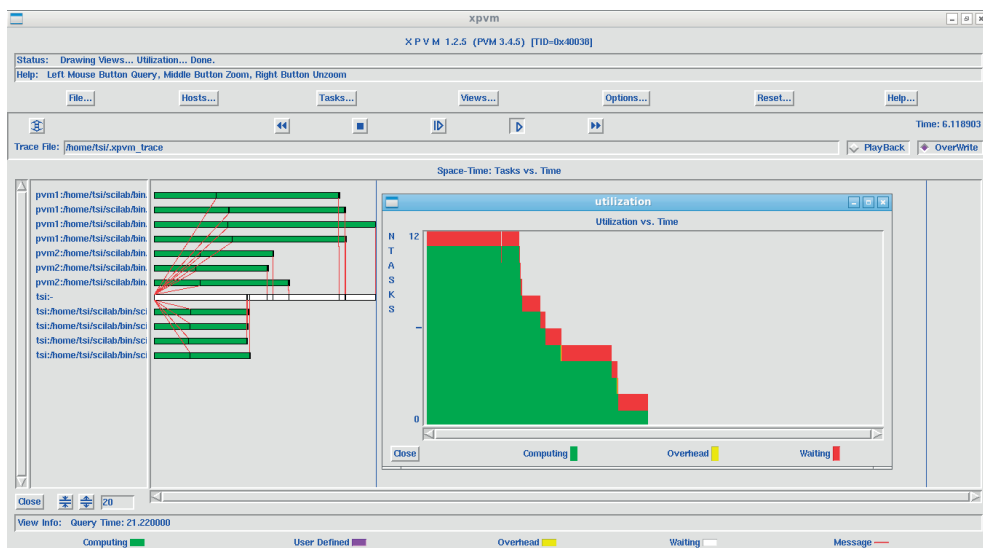
Do głównych zalet środowiska PVM należą: możliwość stosunkowo taniego zwiększania mocy obliczeniowej udostępnianej dla wykonywanych programów i prezentacja graficzna rozpraszanego programu oraz dydaktycznie przyjazny model przekazywania wiadomości na temat systemów rozproszonych.

PVM składa się z dwóch elementów. Pierwszym jest demon PVM, zwykle proces `pvm3` lub `pvm3d` uruchamiany na maszynach dołączonych do systemu rozproszonego. Drugim jest zbiór bibliotek umożliwiających wykonywanie równoległe pożądaných programów. Najczęściej używanymi poleceniami są: `add` (dodawanie nowego węzła), `delete` (usuwanie węzła), `conf` (sprawdzenie bieżącej konfiguracji maszyny wirtualnej) oraz `ps` (wyświetlanie listy zadań).

s. 37–48.

⁴ A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V.S. Sunderam, *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing, Scientific and Engineering Computation*, The MIT Press, Cambridge 1994, s. 2–9.

Programiści do wizualizacji procesu rozpraszania w środowisku PVM mogą wykorzystać przeznaczony do tego celu program XPVM, dzięki któremu możliwe jest zarządzanie maszyną wirtualną oraz wizualizacja wyników działania algorytmu rozpraszającego (rysunek 2). Podsumowując, należy stwierdzić, że PVM oferuje: zarządzanie zadaniami i zasobami, metody komunikacji pomiędzy węzłami, tolerancję na uszkodzone węzły systemu oraz wizualizację rozpraszania procesów.



Rysunek 2. Przykład wizualizacji rozpraszania programu na PMiSL

Źródło: opracowanie własne.

Scilab to środowisko programistyczne pozwalające na profesjonalne przetwarzanie danych do celów inżynierskich i naukowych. Gdy zaistnieje potrzeba szybkiego przetestowania czy sprawdzenia pewnego pomysłu, narzędzie takie jak Scilab nadaje się do tego wyśmienicie. Wygodne środowisko Scilab, zbliżone składnią do takich komercyjnych rozwiązań jak Matlab, Mathematica, WolframAlpha, pozwala na prowadzenie obliczeń na podstawie rachunku macierzowego. Dzięki temu możliwe jest przetwarzanie nie tylko pojedynczych czy skalarnych danych, ale całych macierzy danych⁵.

Założek Scilab powstał w firmie IRIA, potem INRIA i DIGITEO, a obecnie jego rozwój jest możliwy dzięki firmie Scilab Enterprises i wsparciu organizacji

⁵ <http://www.scilab.org> (data odczytu 22.11.2013).

INRIA⁶. Aktualnie dostępna jest wersja 5.4.1 Scilaba oraz wersja 5.5.0 beta, w której twórcy zmienili politykę interfejsu, zbliżając ją wyglądem do rozwiązań Matlaba.

Scilab jest narzędziem wieloplatformowym, można go uruchomić zarówno podsystemami z rodziny Windows, jak i podsystemem operacyjnym Linux czy MacOS. Do niekwestionowanych zalet tego środowiska należy zaliczyć przede wszystkim: bezpłatną licencję, otwarte kody źródłowe, wysoką wydajność narzędzia, zbliżoną składnię do komercyjnego rozwiązania Matlab, programowanie w języku skryptowym wysokiego poziomu składnią także zbliżoną do języka C, dostępność rozszerzeń zwanych Atomsami (dawniej ToolBoxami), wygodną manipulację macierzami, dostępność dobrych i solidnych bibliotek matematycznych skompilowanych w języku C, możliwość współpracy z językami Fortran i C, kontrolę wykonywania kodu, wizualizacje wyników oraz możliwość wizualnego programowania okienkowego. W dobrze zorganizowanym laboratorium badawczym Scilab może posłużyć do szybkiego przetestowania koncepcji nowo opracowanego modelu przez skrócenie procesu standaryzacji tworzonego rozwiązania.

Główną biblioteką wykorzystywaną w ramach omawianych badań jest PVM Parallel Toolbox. Dzięki poprawce tego rozszerzenia możliwe było połączenie Scilaba 5.3.3 z wirtualną maszyną równoległą. Stosując odpowiednie techniki programowania równoległego oraz składni Scilaba, można programować w środowisku rozproszonym. Wykorzystano także dodatkowe biblioteki *signal toolbox* i *general toolbox*, autorstwa G. Peyrego⁷, kompatybilne z wersją Scilab 5.3.3, pozwalającą na wczytywanie obrazów medycznych do tego środowiska.

Podsumowując, należy stwierdzić, że Scilab to darmowe środowisko programistyczne, pozwalające na proceduralne przetwarzanie danych oraz programowanie równoległe, rozproszone i współbieżne.

4. Plan badań

W związku z możliwym pojawieniem się scenariusza pracy ratowników w Państwowym Ratownictwie Medycznym, w którym w procesie podejmowania decyzji medycznych niezbędna byłaby zaawansowana analiza informacji

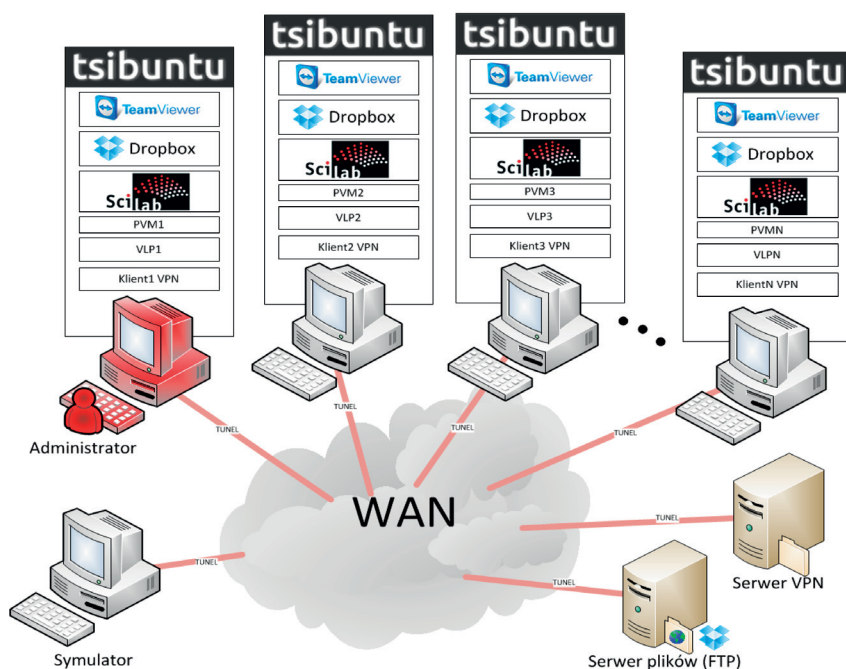
⁶ <http://www.scilab.org/scilab/history> (data odczytu 22.11.2013).

⁷ Źródło bibliotek: https://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/graphics_2_synthesis_wavelets (data odczytu 22.11.2013).

medycznych (w tym również obrazów), założono, że czas od przekazania zbiorów danych do uzyskania wyników nie powinien przekroczyć 90 sekund.

Przyjęto, że jeden z komputerów przenośnych będących w dyspozycji zespołu ratownictwa medycznego jest podłączony do platformy modelowania i symulacji LabTSI za pośrednictwem mobilnych, pakietowych połączeń danych.

Konfiguracja połączeń platformy LabTSI została przygotowana tak, że możliwa jest praca w układzie rozproszonym (rysunek 3).



Rysunek 3. Konfiguracja platformy LabTSI dla symulowanego przypadku analizy informacji medycznych

Źródło: P. Filipkowski, M. Horodelski, *Platforma modelowania i symulacji*, Seminarium WIML, Materiały ECTSI, Warszawa 2013.

Zakładany scenariusz zespołu ratownictwa medycznego jest następujący:

1. Odebranie dyspozycji wyjazdu do zdarzenia losowego od dyspozytora.
2. Transport ratowników do miejsca zdarzenia oraz porządkowanie informacji na temat zdarzenia.
3. Przyjazd na miejsce zdarzenia i przejęcie poszkodowanego.
4. Wykonanie rozpoznania, w tym wykonanie dokładnej analizy stanu zdrowia poszkodowanego.

5. Rozwiązywanie problemu decyzyjnego i podjęcie decyzji o doborze dalszych działań ratujących życie.
6. Działania ratujące życie.
7. Transport poszkodowanego do szpitalnego oddziału ratunkowego.
8. Przejęcie poszkodowanego przez szpitalny oddział ratunkowy.

Proces wspomagania decyzji ratowników będzie miał początek w punkcie piątym zakładanego scenariusza. Wykonana zdalna analiza obrazów będzie bezpośrednio wpływać na decyzje o dalszym przebiegu procesu ratowniczego.

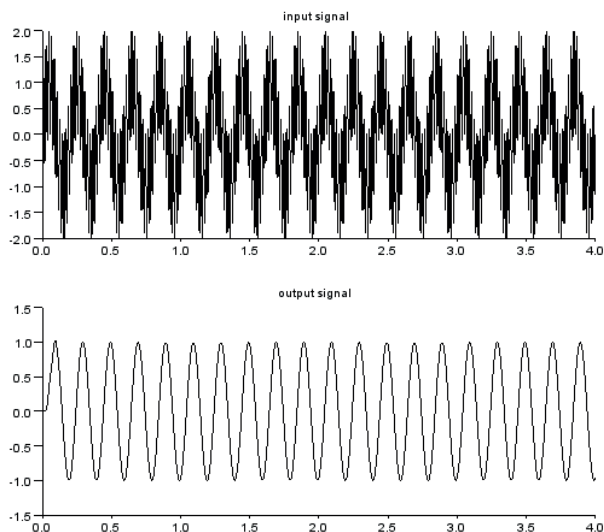
5. Badania symulacyjne

Badania symulacyjne przeprowadzono dla dwóch typów danych. Pierwszy typ to dane w postaci szeregów czasowych pochodzących np. z medycznych urządzeń pomiarowych, drugi typ danych to obraz będący zdjęciem lub prześwietleniem części ciała człowieka.

O ile przetwarzanie szeregów czasowych na urządzeniach mobilnych typu laptop nie jest większym problemem, o tyle w przypadku netbooków, tabletów czy smartfonów może okazać się pewnym wyzwaniem. Przetwarzanie obrazów jest bardziej energochłonne niż przetwarzanie danych szeregów czasowych, tak więc z pewnością potrzebne byłyby urządzenia o większej wydajności. Ponadto w warunkach polowych nie zawsze jest dostęp do energii elektrycznej, dlatego problem zarządzania energią elektryczną w takich warunkach może mieć kluczowe znaczenie dla powodzenia prowadzonych działań, a tym samym dla bezpieczeństwa poszkodowanych i ratowników.

Autorzy kierowani chęcią podniesienia jakości prowadzonych analiz danych medycznych oraz troską o energię będącą w zasięgu urządzeń mobilnych proponują obliczenia rozproszone w warunkach działania zespołów ratownictwa medycznego. Działanie takie zapewnia dostęp do urządzeń o stałym zasilaniu oraz znacznie większej mocy obliczeniowej.

Pierwsze badania przetwarzania rozproszonego danych z szeregów czasowych przy wykorzystaniu transformaty Fouriera na PMiSL były przeprowadzone w Wojskowym Instytucie Medycyny Lotniczej w 2013 r. Prezentowane wyniki prowadzonych symulacji są niejako kontynuacją uzyskanego wyniku, a jako metodę przetwarzania danych z szeregów czasowych wykorzystano metodę filtrowania sygnałów cyfrowych przy wykorzystaniu filtrów cyfrowych IIR (ang. *Infinite Impulse Response*). Modelowy wynik filtrowania w Scilabie przedstawiono na rysunku 4.



Rysunek 4. Wynik filtrowania (*output signal*) zaszumionego sygnału okresowego (*input signal*)

Źródło: opracowanie własne.

W obliczeniach została wykorzystana funkcja IIR⁸, która zapewnia niską złożoność obliczeniową oraz niewielkie zapotrzebowanie na pamięć operacyjną. Ze względu na dużą elastyczność w kształtowaniu przebiegu funkcji za pomocą ilorazu wielomianów znacznie łatwiej uzyskać pożądaną charakterystykę, używając filtru IIR niskiego rzędu niż filtru FIR (ang. *Finite Impulse Response*).

W prowadzonych obliczeniach symulacyjnych podczas projektowania filtru dolnoprzepustowego wykorzystano filtr Butterwortha, działający w dziedzinie częstotliwości, który charakteryzuje się maksymalnie płaską charakterystyką amplitudową w paśmie przenoszenia.

Metoda analizy obrazu jest to operacja mająca na celu wydobycie pewnych istotnych informacji z obrazu bądź usunięcie ich nadmiaru. Bardzo często gdy jakość zdjęcia rentgenowskiego jest słaba, ekspozycja i kontrast na zdjęciu są zbyt małe, utrudnia to widoczność struktur anatomicznych i ich konturów⁹. W ramach badań skupiono się na drugim zagadnieniu. Wykorzystano metodę desaturacji o poziomie szarości obliczanym przy pomocy średniej arytmetycznej. Informacja o kolorach zapisana jest w trzech subpikselach każdego piksela

⁸ http://help.scilab.org/docs/5.3.3/en_US/iir.html (data odczytu 22.11.2013).

⁹ J. Jakubaszko, *Medycyna ratunkowa. Nagłe zagrożenia pochodzenia wewnętrznego*, Wydawnictwo Medyczne Górnicki, Wrocław 2003, s. 366.

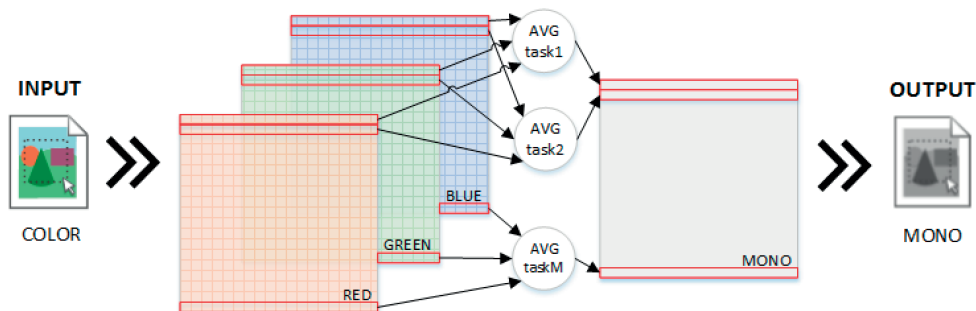
badanego obrazu w formacie BMP. Wczytany obraz do Scilaba jest przechowywany w obiekcie zawierającym informacje o trzech kanałach: red, green i blue. Aby uzyskać obraz monochromatyczny w wybranej metodzie, należy uśrednić wartość piksela na podstawie wartości jego subpikseli. Nowy obraz może być wyliczany przy pomocy następującego opisu:

Niech będzie dana struktura przechowująca trzy macierze $A = \{a_{xyz}\}$, gdzie $x \in \{1, 2, \dots, n\}$, $y \in \{1, 2, \dots, m\}$ i indeks $z \in \{1, 2, 3\}$ będący identyfikatorem kolejnych macierzy oraz macierz wynikowa $B = \{b_k\}$ o wymiarach $n \times m$. Wtedy informacja o nowym obrazie powstaje w wyniku obliczenia poszczególnych elementów macierzy B:

$$b_{jk} = \frac{1}{3} \sum_{z=1}^3 a_{jkz}, \quad (1)$$

gdzie $j \in \{1, 2, \dots, n\}$, $k \in \{1, 2, \dots, m\}$.

W celu realizacji metody zaimplementowano funkcję `desaturacja()`, a następnie przy wykorzystaniu technik programowania równoległego przekształcono implementację w skrypt `pvm_desaturacja.sce`, rozpraszający obliczanie macierzy **B** i zapisujący wynik działania metody na dysku w postaci obrazu monochromatycznego.



Rysunek 5. Proces tworzenia nowej macierzy z podziałem na zadania

Źródło: opracowanie własne.

Skrypt dzieli obliczanie całej macierzy na obliczanie jej poszczególnych wierszy. W ten sposób sekwencyjne obliczanie kolejnych wierszy nowej macierzy jest zastępowane przez równoległe, co przyspiesza proces przekształcający obraz kolorowy w obraz o odcieniach szarości.

Opracowany program rozproszony działa na podstawie paradygmatu programowania o nazwie *crowd computing* typu *master-slave*¹⁰.

6. Wyniki badań symulacyjnych

Badania danych szeregów czasowych przeprowadzono dla dwóch konfiguracji PMiSL – sekwencyjnej i rozproszonej. Zastosowano konfigurację sekwencyjną uruchomioną na jednym netbooku, konfigurację sekwencyjną uruchomioną na jednym laptopie oraz konfigurację rozproszoną uruchomioną na czterech laptopach. W przypadku obliczeń dla danych szeregów czasowych założono wysoką jakość analizowanych sygnałów oraz ich znaczną liczebność. Założono konieczność filtrowania 1000 sygnałów dla 10 różnych ustawień filtra (łącznie 10 000 wektorów). Wyniki czasów obliczeń zebrano w poniższych tabelach.

Tabela 1. Wyniki czasów obliczeń dla danych szeregów czasowych na netbooku

Urządzenie	Liczba węzłów obl. (szt.)	Średni czas analizy (s)
Netbook	1	4990

Źródło: opracowanie własne.

Tabela 2. Wyniki czasów obliczeń dla danych szeregów czasowych na laptopie i7

Urządzenie	Liczba węzłów obl. (szt.)	Średni czas analizy (s)
Laptop i7	1	180

Źródło: opracowanie własne.

Tabela 3. Wyniki czasów obliczeń dla danych szeregów czasowych na czterech węzłach w sieci WAN przy dużym obciążeniu sieci

Urządzenie	Liczba węzłów obl. (szt.)	Średni czas analizy (s)
Laptop i7 Laptop i5 Laptop i3 Laptop T5550	4	80

Źródło: opracowanie własne.

¹⁰ A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. S. Sunderam, op.cit., s. 34–37.

Natomiast w badaniu czasu rozpraszania obliczeń związanych z analizą obrazów medycznych przeprowadzono kilka testów w różnych konfiguracjach. Badano czas dla przetworzenia grupy obrazów, gdzie poszczególne węzły jako zadanie otrzymywały do przetworzenia całe obrazy, a także w drugim sposobie rozsyłano zadania mające na celu osobne obliczanie fragmentów jednego obrazu. Poniżej znajdują się wyniki przetwarzania pojedynczego obrazu.

Tabela 4. Wyniki czasów obliczeń dla danych graficznych w sieci LAN

Urządzenie	Liczba węzłów obl. (szt.)	Średni czas analizy (s)
Laptop i3	1	84
Laptop i3 Laptop T5550	2	48

Źródło: opracowanie własne.

Tabela 5. Wyniki czasów obliczeń dla danych graficznych w sieci WAN przy dużym obciążeniu sieci

Urządzenie	Liczba węzłów obl. (szt.)	Średni czas analizy (s)
Laptop i7 Laptop i5 Laptop i3 Laptop T5550	4	70

Źródło: opracowanie własne.

Przetworzenie obrazu na jednym komputerze jest zdecydowanie bardziej czasochłonnym procesem niż przetwarzanie zrównoleglone na dwóch komputerach. Okazało się także, że czas obliczeń przy dużym obciążeniu sieci WAN był krótszy od czasu obliczeń wykonanych na jednym komputerze.

Przy założonym stałym poziomie jakości decyzji zysk uzyskany w czasie można traktować jako korzyść na rzecz przetrwania poszkodowanego bądź koszt na rzecz uzyskania wyższej jakości decyzji, a tym samym obsługi poszkodowanego.

7. Podsumowanie

W pracy przedstawiono możliwości współczesnych platform modelowania i symulacji oraz przykłady ich zastosowania w ratownictwie medycznym. Informatyka wykorzystywana w służbie człowiekowi jest ważnym obszarem technologii społeczeństwa informacyjnego, istotnie wpływającym na jakość

życia. W niniejszym artykule zaprezentowano przewagę mobilnych rozwiązań w układzie rozproszonym nad stacjonarnymi oraz korzyści płynące z zaawansowanych metod komputerowego wspomaganie decyzji w warunkach stresu.

Ze względu na to, że przeżycie zależy od czasu udzielenia pomocy zgodnie z tzw. złotą godziną¹¹, o życiu poszkodowanego mogą decydować nawet sekundy. Cytując wypowiedź J. Jakubaszki z wywiadu dla „Dziennika Zachodniego”: „Przecież liczy się czas”¹². W czasie udzielania pomocy, a nawet w trakcie transportu poszkodowanego, jak pokazują wyniki przeprowadzonego badania, zyskany czas można wykorzystać na rzecz podniesienia jakości działań ratowniczych. Pobrane dane medyczne można przesłać do platformy modelowania i symulacji, gdzie w krótkim czasie można przeprowadzić ich analizę i podjąć decyzję o stosownym dalszym postępowaniu zespołu ratownictwa medycznego czy ewentualnych przygotowaniach w SOR. Wykorzystanie platform symulacyjnych do analizy próbek medycznych na bieżąco ma również znaczenie ekonomiczne w kontekście poprawnie i szybko postawionej diagnozy oraz trafnie podjętych działań ratujących życie.

Bibliografia

1. De Meo P., Quattrone G., Ursino D., *A Multiagent System for Assisting Citizens in Their Search of E-Government Services*, „Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on” 2008, vol. 38, issue 5, September, wersja elektroniczna z bazy IEEE Xplore.
2. Filipkowski P., Janicki M., Oleszczuk D., *Agent Technologies in LabTSI™ – Use in medical rescue*, „The Polish Journal of Aviation Medicine and Psychology” 2012, vol. 3 (18).
3. Geist A., Beguelin A., Dongarra J., Jiang W., Manchek R., Sunderam V.S., *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing, Scientific and Engineering Computation*, The MIT Press, Cambridge 1994.
4. Jakubaszko J., *Medycyna ratunkowa. Nagłe zagrożenia pochodzenia wewnętrznego*, Wydawnictwo Medyczne Górnicki, Wrocław 2003.
5. Jakubaszko J., *Ratownik medyczny*, Wydawnictwo Medyczne Górnicki, Wrocław 2007.

¹¹ J. Jakubaszko, *Ratownik medyczny*, Wydawnictwo Medyczne Górnicki, Wrocław 2007, s. 115.

¹² *Ratownicy nie mają prawa stwierdzić zgonu w karetce*, http://www.dziennikzachodni.pl/artykul/962020,prof-jakubaszko-ratownicy-nie-maja-prawa-stwierdzic-zgonu-w-karetce,id,t.html?cookie=1#czytaj_dalej (data odczytu 22.11.2013).

6. Janicki A., *Platforma modelowania i symulacji LabTSI™*, Wydawnictwo KUL, Lublin 2011.
7. Kisiała W., *Fundusze strukturalne Unii Europejskiej na rozwój systemu ratownictwa medycznego w Polsce*, „Biuletyn” Instytutu Geografii Społeczno-Ekonomicznej i Gospodarki Przestrzennej UAM, nr 8, *Praktyczne aspekty badań regionalnych – varia*, t. 2, red. P. Churski, Uniwersytet im. Adama Mickiewicza w Poznaniu, Bogucki Wydawnictwo Naukowe, Poznań 2009.
8. Kołodziński E., *Komputerowe wspomaganie procesów informacyjno-decyzyjnych ratownictwa*, II Konferencja Naukowa Zarządzanie Kryzysowe nt. „Ratownictwo w sytuacjach kryzysowych”, Akademia Morska, Szczecin, 18 czerwca 2004.
9. Kostin A., Ilushechkina L., *Modeling and Simulation of Distributed Systems*, World Scientific, Singapore 2010.
10. Olariu S., Maly K., Foudriat E.C., Yamany S.M., *Wireless support for telemedicine in disaster management*, ICPADS '04 Proceedings of the Parallel and Distributed Systems, Tenth International Conference, IEEE Computer Society, Washington DC 2004, wersja elektroniczna z bazy IEEE Xplore.
11. Peters J., *Assessment of ambulance response performance using a geographic information system*, „Social Science & Medicine” 1999, vol. 49.
12. Scilab, *Help, Help On-line*, Scilab Enterprises, INRIA, ENPC, with contributors, Tue Apr 02 17:36:25 CEST 2013.
13. Tadeusiewicz R., *Informatyka medyczna*, Instytut Informatyki, UMCS, Lublin 2011.
14. Zubairi J., Misbahuddin S., *Ambulatory Data Aggregation and Reduction for Transmission Over Limited Bandwidth Network*, Proc. CTS 2009, May 2009, wersja elektroniczna z bazy IEEE Xplore.

Źródła sieciowe

1. <http://www.scilab.org> (data odczytu 22.11.2013).
2. http://help.scilab.org/docs/5.3.3/en_US/iir.html (data odczytu 22.11.2013).
3. <http://www.scilab.org/scilab/history> (data odczytu 22.11.2013).

* * *

Applied distributed modeling and simulation platform for the analysis of medical information in real time

Summary

The paper presents selected methods of processing medical information relevant to the type of threat to life or health using telecommunication systems, in order to create opportunities to take appropriate immediate medical decisions and emergency

action. The aim of the research is to recognise computational capabilities of the selected modeling and simulation platform, and then use it to provide mobile support for the analysis of medical information under stress. For the purpose of this paper, one possible scenario of the medical services' action has been chosen and presented alongside with its implementation on the LabTSI[®] platform. The paper argues that the use of modeling and simulation environment and of distributed programming enable a successful reduction of the information processing time, so important in emergency medical services.

Keywords: modeling and simulation platform, paramedic, scilab, pvm, agent technology, VPN, client/server paradigm, distributed programming, parallel programming, Finite and Infinite Impulse Response Filter, desaturation method