

## Wieloskalowe symulacje biologicznych sieci neuronowych charakteryzujących się wewnętrzną topologią wielowymiarowych torusów z wykorzystaniem PGENESIS

*Large-scale simulations of biological neural networks characterized by internal topology of multi-dimensional torus using PGENESIS*

Monika Kwiatkowska<sup>1</sup> i Łukasz Świerczewski<sup>2</sup>

**Treść:** Praca obejmuje implementacje oraz testy wydajności i skalowania biologicznych sieci neuronowych charakteryzujących się wewnętrzną topologią wielowymiarowych torusów. Do obliczeń wykorzystano równoległą wersję symulatora GENESIS - PGENESIS. Symulacje przeprowadzono w środowisku równoległym na superkomputerze (klaster wydajnościowy o architekturze x86\_64) HP BladeSystem/Actina, Hydra dostępnym w Interdyscyplinarnym Centrum Modelowania Matematycznego i Komputerowego Uniwersytetu Warszawskiego. Testy objęły procesory AMD Opteron 2435, AMD Opteron 6174, AMD Opteron 6272 oraz Intel Xeon X5660. Uwzględniono także aspekt wykorzystania interfejsów sieciowych Infiniband QDR, Infiniband DDR oraz 10Gb Ethernet w komunikacji międzywęzłowej. Dodatkowo wykonano analizę uzyskanego zysku wydajności dzięki zastosowaniu wersji PGENESIS skompilowanej pod kątem wybranego procesora. W pracy skupiono się jedynie na części dotyczącej pomiarów wydajności – nie podjęto jakichkolwiek prób analiz aktywności modelowanych biologicznych sieci neuronowych.

**Słowa kluczowe:** PGENESIS, biologiczne sieci neuronowe, topologia torus

**Abstract:** This paper includes implementation and performance tests and also scaling of biological neural networks characterized by internal topology of multi-dimensional toruses. For calculations there was used a parallel version of the GENESIS - PGENESIS simulator. Simulations were performed in a supercomputer's parallel environment, (a performance cluster with x86\_64 architecture) HP BladeSystem/Actina, Hydra available at the Interdisciplinary Centre for Mathematical and Computational Modeling, Warsaw University. Tests included AMD Opteron 2435, AMD Opteron 6174, AMD Opteron 6272 and Intel Xeon X5660 processors. There was also taken into account the aspect of the use of network interfaces such like Infiniband QDR, DDR Infiniband and 10Gb Ethernet in interstitial communication. In addition, there was performed an analysis on the resulting performance gained by using the PGENESIS version compiled for the selected processor. In this paper author focused only on the section of performance measurement - there weren't taken any attempts of activity analysis of the modeled biological neural networks.

**Keywords:** PGENESIS, biological neural networks, torus topology

### Wprowadzenie

GENESIS (the GEneral NEural Simulation System) [1][2] jest platformą ogólnego przeznaczenia, która została opracowana w celu wspierania biologicznie realistycznych symulacji układów nerwowych, od skomplikowanych modeli pojedynczych neuronów do symulacji dużych sieci. GENESIS implementuje język wysokiego poziomu, który pozwala łatwo rozszerzyć możliwości symulatora, wymieniać, modyfikować i wykorzystywać modele lub ich elementy składowe.

Istnieje także równoległa wersja GENESIS – Parallel GENESIS (PGENESIS) [3][4]. Umożliwia ona zrównoleżenie obliczeń i uruchomienie ich na klastrach komputerowych. Wspierany jest zarówno standard PVM [5][6], jak i nowszy MPI [7]. W tej pracy wykorzystano MPI w wersji 1.4.1p1. GENESIS i PGENESIS było kompilowane kompilatorem GCC [8] 4.8.1.

Do realizacji obliczeń wykorzystano klaster komputerowy

o architekturze x86\_64 HP BladeSystem/Actina, Hydra zainstalowany w Interdyscyplinarnym Centrum Modelowania Matematycznego i Komputerowego Uniwersytetu Warszawskiego. W klastrze tym dostępne są cztery rodzaje węzłów. Dokładną ich specyfikację wraz z ilością zestawiono w Tab. 1.

Tab. 1. Dostępne zasoby sprzętowe na superkomputerze Hydra zainstalowanym w ICM, UW.

Rodzaj procesora	Taktowanie procesora	Liczba procesorów x liczba rdzeni	RAM	Typ połączenia międzywęzłowego	Liczba węzłów
AMD Opteron 2435	2,6 GHz	2 x 6	32 GB	Infiniband DDR + 1Gb Ethernet	96
Intel Xeon X5660	2,8GHz	2 x 6	24 GB	Infiniband QDR + 1Gb Ethernet	120
AMD Opteron 6174	2,2 GHz	4 x 12	256 GB	10Gb Ethernet	30
AMD Opteron 6272	2,2 GHz	4 x 16	512 GB	10Gb Ethernet	16

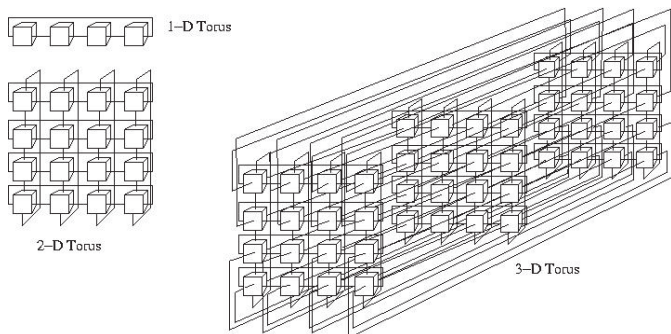
1. Uniwersytet Marii Curie-Skłodowskiej w Lublinie, Wydział Matematyki, Fizyki i Informatyki.

2. Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości w Łomży, Instytut Automatyki i Robotyki.

Jak przedstawiono w powyższej tabeli w systemie dostępnych jest w sumie 262 węzłów obliczeniowych. Największa ich grupa (120 węzłów) posiada po dwa sześciordzeniowe procesory Intel Xeon X5660. Połączone są one w komunikacji międzywęzłowej szybkim interfejsem Infiniband [9][10] QDR oraz 1Gb Ethernet, co daje sumaryczną teoretyczną przepustowość na poziomie 33 Gbit/s. Węzły te posiadają stosunkowo mało pamięci operacyjnej RAM – jedynie 24 GB na węzeł. Drugą pod względem liczebności (96 węzłów) grupą są jednostki wyposażone w procesory AMD Opteron 2435. Posiadają one więcej pamięci operacyjnej RAM (32 GB), jednak charakteryzują się niższą częstotliwością taktowania oraz są połączone znacznie wolniejszą siecią Infiniband typu DDR o przepustowości 16 Gbit/s. Węzłów tych także jest dość dużo – aż 96. Ostatnie dwa typy węzłów obliczeniowych stanowią rozwiązania oparte o procesory AMD Opteron 6174 oraz AMD Opteron 6272. Posiadają one bardzo dużą ilość pamięci RAM (odpowiednio 256 i 512 GB) jednak są połączone jedynie wolną siecią Ethernet o przepustowości 10 Gbit/s.

Kluczową kwestią w przypadku porównywania systemu połączeń węzłów może okazać się nie tylko maksymalna przepustowość ale także opóźnienia. W przypadku urządzeń działających w standardzie Ethernet 10 Gbit/s te mogą wynosić powyżej 40 mikrosekund dla Infiniband DDR czas ten wynosi 2 mikrosekundy, a przy Infiniband QDR spotkamy się z opóźnieniem w granicach jedynie jednej mikrosekundy.

Na potrzeby niniejszej pracy zaimplementowano sieci bazujące na topologiach wielowymiarowych torusów. Przykładowe topologie tego typu dla torusów 1D, 2D i 3D przedstawiono na Ryc. 1.



Ryc. 1. Topologie torus 1D, 2D oraz 3D. Źródło: <http://wiki.expertiza.ncsu.edu/>.

Jak widać na Ryc. 1. topologia torusów charakteryzuje się tym, że każdy z elementów w tej strukturze posiada odpowiednią ilość sąsiadów. Nawet węzły skrajne w tym przypadku są połączone z przeciwległymi sąsiadami.

W PGENESIS zrealizowano strukturę sieci w następujący sposób:

- na każdym węźle jest alokowana siatka o rozmiarze 32x32 neuronów (razem 1024 neuronów), w ramach węzła realizowane jest 30% połączenia pełnego (314572 połączeń),
- strukturę połączeń międzywęzłowych określa topologia torusa o określonej liczbie wymiarów, w tym przy-

padku od węzła do węzła realizowane jest 20% połączenia pełnego (209715 połączeń).

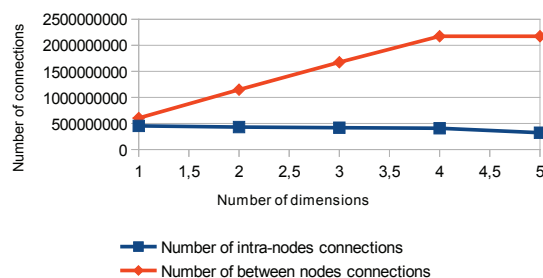
Porównanie charakterystyk największych symulacji jakie można wykonać na jednej partycji procesorów Intel Xeon X5660 na klastrze Hydra przedstawiono w Tab. 2.

Tab. 2. Największe symulacje jakie można zrealizować na jednej partycji systemu Hydra złożonej z 120 węzłów posiadających po dwa sześciordzeniowe procesory Intel Xeon X5660 (razem 1440 rdzeni).

Typ torusa	Rozmiar torusa (ilość procesorów)	Całkowita ilość neuronów	Ilość połączeń wewnątrzwęzłowych / pomiędzy węzłami (suma)
1D	1440 (1440)	1474560	452984832 / 603979776 (1056964608)
2D	37x37 (1369)	1401856	430650163 / 1148400435 (1579050598)
3D	11x11x11 (1331)	1362944	418696396 / 1674785587 (2093481983)
4D	6x6x6x6 (1296)	1327104	407686348 / 2174327193 (2582013541)
5D	4x4x4x4x4 (1024)	1048576	322122547 / 2174327193 (2469606195)

Jak widać w tabeli teoretycznie największą ilość neuronów w sieci można uzyskać stosując topologię torusa 1D. Jest to spowodowane tym, że w implementacji skryptu uwzględniono możliwość realizacji jedynie torusów idealnych – tzn. takich gdzie każdy wymiar ma taki sam rozmiar. Dla torusa dwuwymiarowego tego typu największa struktura posiada 37x37 węzłów, co daje wykorzystanie maksymalnie 1369 rdzeni. Im większy wymiar tym większa utrata możliwości zaangażowania w obliczenia określonej ilości procesorów.

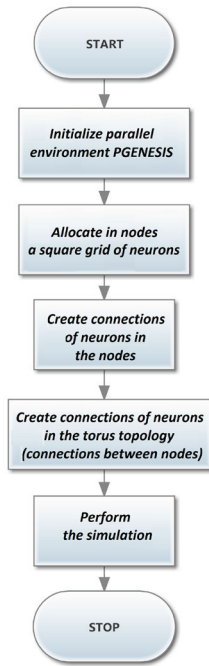
Zmianę ilości połączeń między neuronami przy ustalonej topologii określonego torusa oraz uwzględnieniu maksymalnej liczby dostępnych rdzeni na 1440 przedstawiono na Ryc. 2.



Ryc. 2. Zmiana ilości połączeń między neuronami przy ustalonej topologii określonego torusa oraz uwzględnieniu maksymalnej liczby dostępnych rdzeni na 1440.

## Implementacja

Prosty schemat blokowy szablonu zaimplementowanych skryptów przedstawiono na Ryc. 3.



Ryc. 3. Prosty schemat blokowy szablonu zaimplementowanych skryptów.

```

if ({nodes} == 2)
  if ({mynode} == 1)
    pconnect_two_maps /net1 1 /net2 2 {dim} {dim} {connect_probability_beetwen_node}
    make_synapse /input /net1/cell[0]/dend/Ex_channel 10 0 0
  end
else
  for (i=1;i<nodes+1;i=i+2)
    if ({mynode} == {i})
      if ({i+1}%{nodes} == 0)
        pconnect_two_maps /net{i} {i} /net{nodes} {nodes} {dim} {dim} \
          {connect_probability_beetwen_node}
      else
        pconnect_two_maps /net{i} {i} /net{{i+1}%{nodes}} {{i+1}%{nodes}} {dim} \
          {dim} {connect_probability_beetwen_node}
      end
    end
  end
end

if ({mynode} == 1)
  make_synapse /input /net1/cell[0]/dend/Ex_channel 10 0 0
end
end
  
```

Listing 1. Struktura kod odpowiedzialna za tworzenie połączeń pomiędzy siatkami neuronów o strukturze torusa 1D

```

for (i=1;i<grid_dim+1;i=i+1)
  for (j=1;j<grid_dim+1;j=j+1)
    for (k=1;k<grid_dim+1;k=k+1)
      if ({mynode} == {(i-1)*grid_dim*grid_dim+(j-1)*grid_dim+k})
        if ( {{i+1}%grid_dim} == 0)
          pconnect_two_maps /net{i}-{j}-{k}
            {(i-1)*grid_dim*grid_dim+(j-1)*grid_dim+k} \
            /net{grid_dim}-{j}-{k} \
            {(grid_dim-1)*grid_dim*grid_dim+(j-1)*grid_dim+k} \
            {dim} {dim} {connect_probability_beetwen_node}
        else
          pconnect_two_maps /net{i}-{j}-{k}
            {(i-1)*grid_dim*grid_dim+(j-1)*grid_dim+k} \
            /net{{i+1}%grid_dim}-{j}-{k} \
            {{{i+1}%grid_dim-1}*grid_dim*grid_dim+(j-1)*grid_dim+k} \
            {dim} {dim} {connect_probability_beetwen_node}
        end
      end
    end
  end
end
end
end
end
  
```

Najbardziej zaawansowanym krokiem podczas tworzenia sieci neuronowej jest operacja oznaczona na schemacie (Ryc. 3) jako „Create connections of neurons in the torus topology (connections beetwen nodes)”. Listingi przedstawiające realizację tej procedury w PGENESIS przedstawiono na Listingu 1 (dla torusa 2D) oraz Listingu 2 (dla torusa 3D).

Listing 2. Struktura kod odpowiedzialna za tworzenie połączeń (jedynie po jednej osi) pomiędzy siatkami neuronów o strukturze torusa 3D

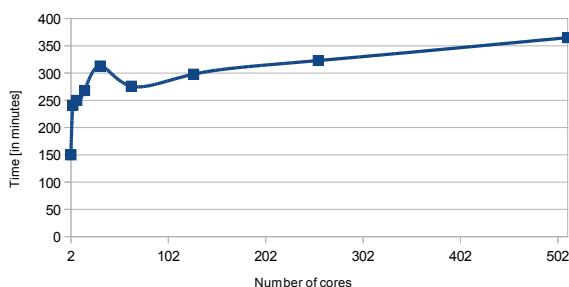
## Zrealizowane symulacje

Na potrzeby pracy badawczej zrealizowano szereg symulacji – zostały one przedstawione w Tab. 3 (torus 1D), Tab. 4 (torus 2D), Tab. 5 (torus 3D), Tab. 6 (torus 4D) oraz Tab. 7 (torus 5D).

Tab. 3. Zestawienie symulacji wykonanych z wykorzystaniem topologii wewnętrznej torus 1D na czterech różnych typach węzłów.

Rozmiar torusa (ilość procesorów)	Czas realizacji symulacji na AMD Opteron 6272	Czas realizacji symulacji na AMD Opteron 6174	Czas realizacji symulacji na AMD Opteron 2435	Czas realizacji symulacji na Intel Xeon X5660
2 (2)	271 min 25,199 s	386 min 43,978 s	369 min 22,822 s	150 min 21,573 s
4 (4)	322 min 3,664 s	303 min 57,682 s	375 min 19,659 s	241 min 12,390 s
8 (8)	374 min 26,470 s	592 min 5,022 s	319 min 12,117 s	250 min 46,845 s
16 (16)	486 min 8,700 s	463 min 29,277 s	352 min 49,962 s	268 min 8,244 s
32 (32)	475 min 12,651 s	599 min 14,141 s	346 min 52,060 s	313 min 45,030 s
64 (64)	526 min 33,446 s	509 min 55,929 s	342 min 39,755 s	276 min 12,626 s
128 (128)	523 min 31,221 s	577 min 54,899 s	400 min 20,326 s	298 min 19,998 s
256 (256)	?	530 min 27,452 s	?	323 min 42,148 s
512 (512)	?	530 min 7,629 s	?	365 min 5,279 s
720 (720)	?	?	?	478 min 17,941 s <b>ERROR NR 25 / 6</b>

Jak już widać w Tab. 3 dla torusa 1D wraz ze wzrostem rozmiaru symulowanej czasu wcale niekoniecznie łączy się także wzrostowy trend czasu jej symulacji. Spowodowane jest to tym, że zarówno dwa najistotniejsze parametry określające rozmiar sieci (ilość neuronów i ilość połączeń między nimi), jak i ilość realizujących symulację procesorów rośnie w sposób liniowy. Przykładowy wykres obrazujący zależność czasu od rozmiaru symulacji na torusie 1D (i zastosowanej ilości procesorów Intel Xeon X5660) przedstawiono na Ryc. 4.



Ryc. 4. Wykres obrazujący zależność czasu od rozmiaru symulacji na torusie 1D oraz zastosowanej różnej ilości procesorów Intel Xeon X5660.

Na Ryc. 4 widać zauważalne maksimum czasu realizacji w przedziale zastosowania od 2 do 128 rdzeni – dokładnie wystąpiło ono podczas symulacji określonej sieci na 32 rdzeniach. Widać, że dla większych symulacji (> 128 rdzeni) czas w niewielkim stopniu proporcjonalnie rośnie. Jak zaznaczono w Tab. 3 podczas wykonywania największych symulacji (wykorzystanych 720 rdzeni) wystąpił błąd. Objawiał się on głównie zwróceniem na ekran poniższej informacji i zakończeniem symulacji:

```
Parlib error 25: barrierall hung in local zone barrier (timed out)
```

```
=====
= BAD TERMINATION OF ONE OF YOUR APPLICATION PROCESSES
= EXIT CODE: 6
= CLEANING UP REMAINING PROCESSES
= YOU CAN IGNORE THE BELOW CLEANUP MESSAGES
=====
```

```
APPLICATION TERMINATED WITH THE EXIT STRING: Aborted (signal 6)
```

Z powodu tego błędu nie wykonywano już większych symulacji. Zarzucono także próby wykorzystania m.in. całej partycji procesorów Intel Xeon X5660 (120 węzłów) co początkowo planowano zrobić. Dodatkowo w Tab. 3 w niektórych polach wstawiono znak „?” - oznacza on, że pomimo blisko miesiąca oczekiwania w systemie kolejkowym zadanie to nie zostało zrealizowane (brak wolnych zasobów sprzętowych). Poniżej zestawiono wyniki dla torusów 2-5D.

Tab. 4. Zestawienie symulacji wykonanych z wykorzystaniem topologii wewnętrznej torus 2D na węzłach posiadających procesory Intel Xeon X5660.

Rozmiar torusa (ilość procesorów)	Czas realizacji symulacji
2x2 (4)	391 min 54,872 s
3x3 (9)	515 min 23,696 s
6x6 (36)	453 min 35,542 s
8x8 (64)	452 min 35,541 s
12x12 (144)	425 min 26,538 s
16x16 (256)	461 min 35,235 s
24x24 (576)	484 min 9,090 s
26x26 (676)	481 min 37,494 s

Tab. 5. Zestawienie symulacji wykonanych z wykorzystaniem topologii wewnętrznej torus 3D na węzłach posiadających procesory Intel Xeon X5660.

Rozmiar torusa (ilość procesorów)	Czas realizacji symulacji
2x2x2 (8)	499 min 29,652 s
3x3x3 (27)	572 min 39,844 s
4x4x4 (64)	590 min 36,026 s
6x6x6 (216)	796 min 28,919 s
8x8x8 (512)	<b>ERROR NR 25 / 6</b> 600 min 29,887 s

Tab. 6. Zestawienie symulacji wykonanych z wykorzystaniem topologii wewnętrznej torusa 4D na węzłach posiadających procesory Intel Xeon X5660.

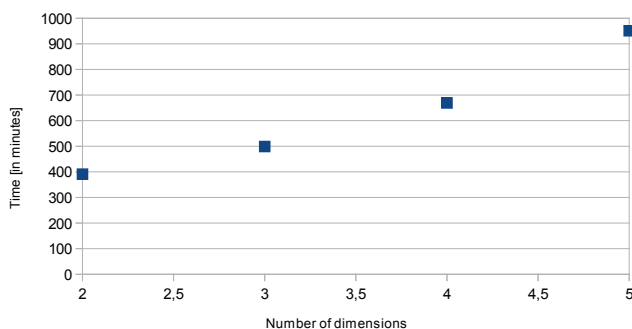
Rozmiar torusa (ilość procesów)	Czas realizacji symulacji
2x2x2x2 (16)	669 min 36,610 s
3x3x3x3 (81)	658 min 28,172 s
4x4x4x4 (256)	727 min 9,920 s
5x5x5x5 (625)	790 min 4,390 s

Tab. 7. Zestawienie symulacji wykonanych z wykorzystaniem topologii wewnętrznej torusa 5D na węzłach posiadających procesory Intel Xeon X5660.

Rozmiar torusa (ilość procesów)	Czas realizacji symulacji
2x2x2x2x2 (32)	951 min 58,543 s
3x3x3x3x3 (243)	856 min 40,842 s

Jak widać utrzymywany jest w miarę stały czas realizacji w ramach danego torusa. Dla torusa 2D waha się on w przedziale od 391 do 515 minut, dla torusa 3D od 499 do 796 minut, dla torusa 4D od 669 do 790 minut i wreszcie dla torusa 5D ze względu na największą ilość połączeń także czas realizacji jest najwyższy i wynosi od 851 do 951 minut (dla jedynie dwóch symulacji).

Zmianę czasu realizacji symulacji w odniesieniu do ilości wymiarów torusa podczas wykonywania na Intel Xeon X5660 przedstawiono na Ryc. 5.



Ryc. 5. Czas realizacji symulacji w odniesieniu do ilości wymiarów torusa podczas wykonywania na Intel Xeon X5660

### Analiza zysku uzyskanego dzięki kompilacji PGENESIS pod kątem określonego typu procesora

Podczas badań podjęto także próbę porównania wydajności uzyskanej dzięki zastosowaniu PGENESIS skompilowanego z optymalizacją pod kątem określonego typu procesora z kompilacją bez takiej optymalizacji (kompilacja generic). Czasy wykonywania przez nieoptymalizowane PGENESIS dla sieci o topologii wewnętrznej torusa 1D ujęto w Tab. 8. Ograniczono się w tym przypadku do wykonania symulacji na konfiguracjach jedno-węzłowych (tak aby pominąć dodatkowy narzut czasu spowodowany dodatkową komunikacją za pomocą znacznie wolniejszych interfejsów sieciowych).

Tab. 8. Czasy realizacji symulacji sieci neuronowych o topologii wewnętrznej torusa 1D z wykorzystaniem nieoptymalizowanej pod kątem procesora wersji PGENESIS.

Rozmiar torusa (ilość procesów)	Czas realizacji symulacji na AMD Opteron 6272	Czas realizacji symulacji na AMD Opteron 6174	Czas realizacji symulacji na AMD Opteron 2435	Czas realizacji symulacji na Intel Xeon X5660
2 (2)	272 min 53,974 s	297 min 34,104s	423 min 12,055 s	302 min 58,905 s
4 (4)	215 min 5,974 s	353 min 51,872 s	322 min 0,303 s	239 min 42,939 s
8 (8)	373 min 38,573 s	517 min 29,013 s	405 min 29,657 s	316 min 28,257 s
16 (16)	409 min 57,412 s	681 min 13,005 s	-	-
32 (32)	667 min 50,090 s	469 min 5,884 s	-	-
64 (64)	528 min 17,632 s	-	-	-

Jak widać w Tab. 8 wyniki nie są jednoznaczne – raz szybsza jest wersja zoptymalizowana pod kątem procesora, a raz jest wręcz przeciwnie. Bardziej czytelne porównanie przedstawiono w Tab. 9. W tej tabeli przykładowy zapis + 0,36% oznacza, że podczas realizacji danej symulacji wersja skompilowana z optymalizacją pod kątem danego procesora okazała się o **0,36%** szybsza od zastosowania bez optymalizacji. Analogicznie w przypadku gdy wartość procentową poprzedza minus mamy do czynienia ze spowolnieniem wykonywania symulacji.

Tab. 9. Procentowy zysk lub strata uzyskana dzięki zastosowaniu optymalizacji PGENESIS pod kątem procesora.

Rozmiar torusa (ilość procesów)	Czas realizacji symulacji na AMD Opteron 6272	Czas realizacji symulacji na AMD Opteron 6174	Czas realizacji symulacji na AMD Opteron 2435	Czas realizacji symulacji na Intel Xeon X5660
2 (2)	+ 0,36%	- 29,96%	+ 14,63%	+ 101,33%
4 (4)	- 49,76%	+ 16,50%	- 16,45%	- 0,83%
8 (8)	- 0,26%	- 14,50%	+ 26,95%	+ 26,40%
16 (16)	+ 18,82%	+ 47,08%	-	-
32 (32)	+ 40,42%	- 27,71%	-	-
64 (64)	+ 0,38%	-	-	-

W Tab. 9 możemy zaobserwować bardzo dużą rozbieżność w uzyskanych pomiarach. Według danych empirycznych uruchamiając zoptymalizowane pod kątem procesora PGENESIS w najgorszym przypadku utracimy 49,76% wydajności (AMD Opteron 6272 i rozmiar torusa równy 4) lub zyskamy 101,33% na szybkości realizacji symulacji (Intel Xeon X5660 i rozmiar torusa równy 2). Trudno jest zdecydować, które z rozwiązań daje lepsze rezultaty. Uśredniony zysk (lub strata) wydajności (suma po kolumnach z Tab. 9) dzięki zastosowaniu optymalizacji pod kątem procesora został przedstawiony w Tab. 10.

Tab. 10. Uśredniony zysk/strata wydajności dzięki zastosowaniu optymalizacji na czterech typach węzłów.

Uśredniony zysk/strata wydajności dzięki zastosowaniu optymalizacji na:			
AMD Opteron 6272	AMD Opteron 6174	AMD Opteron 2435	Intel Xeon X5660
+ 9,96%	- 8,59%	+ 25,13%	+ 126,90%

Jak widać w przypadku interpretowanych danych tak jak w Tab. 10 najlepiej pod tym względem wypadają procesory Intel Xeon X5660, a najgorzej AMD Opteron 6174. Tego typu porównanie może być jednak nieco oderwane od faktycznego problemu (zbytne uproszczenie).

### Analiza spowolnienia obliczeń wynikającego z komunikacji międzywęzłowej

Teoretycznie konieczność zastosowania interfejsów sieciowych w komunikacji międzywęzłowej powinna znacznie spowolnić wykonywanie symulacji realizowanych za pomocą GENESIS. Analizie poddano prosty model oparty o dwuwymiarowy torus o rozmiarze jednego wymiaru równym 2 (2x2 - 4 główne procesy obliczeniowe). Na pojedynczym węźle standardowo alokowano siatkę 32x32 (1024) neuronów. W ramach węzła jak dotychczas realizowano 30% połączenia pełnego, natomiast zmieniano ilość połączeń przy połączeniach międzywęzłowych (od 10% do 100% w kroku co 10%). Wyniki analiz dla procesorów Intel Xeon X5660 oraz Infiniband QDR przedstawiono w Tab. 11. Rezultaty dla AMD Opteron 2435 oraz Infiniband DDR zestawiono w Tab. 12, a dane dla AMD Opteron 6174 ze wsparciem najwolniejszego w zestawieniu protokołu Ethernet 10 Gb można zobaczyć w Tab. 13. Podczas analiz pominięto węzły posiadające procesory AMD Opteron 6272 gdyż posiadają one taki sam standard komunikacji międzywęzłowej co AMD Opteron 6174.

Tab. 11. Porównanie różnych koncepcji przydziału miejsca wykonywania procesów (na jednym węźle lub z podziałem na dwa) z wykorzystaniem procesorów Intel Xeon X5660 oraz interfejsu komunikacji Infiniband QDR.

Wartość procentowa w odniesieniu do połączenia pełnego realizowanego między węzłami	Czas realizacji w przypadku zastosowania jednego węzła	Czas realizacji w przypadku zastosowania dwóch węzłów
	Architektura: 1 (węzeł) x 4 procesy	Architektura: 2 (węzły) x 2 procesy
10%	285 min 1,974 s	303 min 45,377 s
20%	397 min 52,157 s	409 min 14,064 s
30%	489 min 32,580 s	494 min 13,924 s
40%	597 min 42,026 s	602 min 0,824 s
50%	710 min 57,143 s	702 min 44,341 s
60%	823 min 30,265 s	792 min 19,056 s
70%	919 min 49,225 s	909 min 20,841 s
80%	1010 min 34,762 s	1017 min 47,569 s
90%	1149 min 23,754 s	1111 min 44,418 s
100%	1239 min 58,651 s	1512 min 40,198 s

Tab. 12. Porównanie różnych koncepcji przydziału miejsca wykonywania procesów (na jednym węźle lub z podziałem na dwa) z wykorzystaniem procesorów AMD Opteron 2435 oraz interfejsu komunikacji Infiniband DDR.

Wartość procentowa w odniesieniu do połączenia pełnego realizowanego między węzłami	Czas realizacji w przypadku zastosowania jednego węzła	Czas realizacji w przypadku zastosowania dwóch węzłów
	Architektura: 1 (węzeł) x 4 procesy	Architektura: 2 (węzły) x 2 procesy
10%	321 min 46,090 s	329 min 32,030 s
20%	409 min 46,183 s	521 min 24,899 s
30%	549 min 13,830 s	608 min 7,272 s
40%	739 min 56,930 s	743 min 10,726 s
50%	884 min 52,342 s	953 min 10,848 s
60%	840 min 49,289 s	962 min 16,840 s
70%	1154 min 24,513 s	1054 min 41,357 s
80%	1208 min 15,929 s	1270 min 16,345 s
90%	1301 min 9,254 s	1289 min 37,833 s
100%	1613 min 25,423 s	1583 min 44,740 s

Tab. 13. Porównanie różnych koncepcji przydziału miejsca wykonywania procesów (na jednym węźle lub z podziałem na dwa) z wykorzystaniem procesorów AMD Opteron 6174 oraz interfejsu komunikacji Ethernet 10 Gb.

Wartość procentowa w odniesieniu do połączenia pełnego realizowanego między węzłami	Czas realizacji w przypadku zastosowania jednego węzła	Czas realizacji w przypadku zastosowania dwóch węzłów
	Architektura: 1 (węzeł) x 4 procesy	Architektura: 2 (węzły) x 2 procesy
10%	321 min 46,090 s	329 min 32,030 s
20%	409 min 46,183 s	521 min 24,899 s
30%	549 min 13,830 s	608 min 7,272 s
40%	739 min 56,930 s	743 min 10,726 s
50%	884 min 52,342 s	953 min 10,848 s
60%	840 min 49,289 s	962 min 16,840 s
70%	1154 min 24,513 s	1054 min 41,357 s
80%	1208 min 15,929 s	1270 min 16,345 s
90%	1301 min 9,254 s	1289 min 37,833 s
100%	1613 min 25,423 s	1583 min 44,740 s

Jak widać w powyższych tabelach nawet dla teoretycznej najwolniejszej konfiguracji wyposażonej w procesory AMD Opteron 6174 i Ethernet 10 Gb i nie widać zasadniczej różnicy wydajności podczas porównywania dwóch różnych rodzajów przydziałów zasobów sprzętowych. Sumaryczne zestawienie wyników z tabel (suma po kolumnach) Tab. 11, Tab. 12 oraz Tab. 13 przedstawiono w Tab. 14.

Tab. 14. Porównanie różnych koncepcji przydziału miejsca wykonywania procesów - sumaryczny czas realizacji symulacji na trzech typach węzłów i różnych rodzajach połączeń między nimi.

Sumaryczny czas realizacji symulacji					
Intel Xeon X5660 + Infiniband QDR		AMD Opteron 2435 + Infiniband DDR		AMD Opteron 6174 + Ethernet 10 Gb	
Sumaryczny czas realizacji w przypadku zastosowania jednego węzła	Sumaryczny czas realizacji w przypadku zastosowania dwóch węzłów	Sumaryczny czas realizacji w przypadku zastosowania jednego węzła	Sumaryczny czas realizacji w przypadku zastosowania dwóch węzłów	Sumaryczny czas realizacji w przypadku zastosowania jednego węzła	Sumaryczny czas realizacji w przypadku zastosowania dwóch węzłów
Architektura: 1 (węzeł) x 4 procesy	Architektura: 2 (węzły) x 2 procesy	Architektura: 1 (węzeł) x 4 procesy	Architektura: 2 (węzły) x 2 procesy	Architektura: 1 (węzeł) x 4 procesy	Architektura: 2 (węzły) x 2 procesy
7618 min	7851 min	9018 min	9312 min	10748 min	10301 min

W Tab. 14 widzimy, że zgodnie z oczekiwaniami dla konfiguracji posiadających szybki interfejs Infiniband sumaryczny czas realizacji w przypadku zastosowania jednego węzła jest niższy niż w przypadku zastosowania dwóch. Różnice te są jednak dla tej skali czasowej niewielkie i wynoszą ok. od 200 do 300 minut. Natomiast dla konfiguracji AMD Opteron 6174 + Ethernet 10 Gb wyniki czasowe są całkowicie odwrotne – rozwiązanie wykorzystujące jeden węzeł okazało się o ponad 400 minut wolniejsze. Aby mieć pewność co do wniosków końcowych przeprowadzono także symulację sieci o znacznie mniejszej gęstości połączeń w ramach węzła (wykorzystano jedynie 0.003% połączenia pełnego oraz węzły połączone interfejsem Ethernet 10 Gb). Wyniki dla takiego podejścia zostały zaprezentowane w Tab. 15.

Tab. 15. Porównanie różnych koncepcji przydziału miejsca wykonywania procesów (na jednym węźle lub z podziałem na dwa) z wykorzystaniem procesorów AMD Opteron 6174 oraz interfejsu komunikacji Ethernet 10 Gb dla sieci gdzie zrealizowano 0.003% połączenia pełnego w ramach węzła.

Wartość procentowa w odniesieniu do połączenia pełnego realizowanego między węzłami	Czas realizacji w przypadku zastosowania jednego węzła  Architektura: 1 (węzeł) x 4 procesy	Czas realizacji w przypadku zastosowania dwóch węzłów  Architektura: 2 (węzły) x 2 procesy
10%	197 min 48,125 s	183 min 26,121 s
20%	355 min 0,894 s	366 min 48,142 s
30%	506 min 56,474 s	503 min 46,749 s
40%	614 min 54,221 s	617 min 43,608 s
50%	858 min 21,759 s	720 min 3,312 s
60%	924 min 13,016 s	792 min 44,519 s
70%	1039 min 2,005 s	914 min 55,103 s
80%	1294 min 9,567 s	983 min 38,876 s
90%	1432 min 28,770 s	1161 min 26,895 s
100%	1206 min 25,549 s	1258 min 32,250 s

Powyższe pomiary jedynie potwierdziły, że jeżeli posiadamy w miarę nowoczesny interfejs sieciowy to nie jest on jakimkolwiek ograniczeniem dla symulacji w GENESIS.

## Wnioski i perspektywy dalszych badań

Początkowo w założeniach pracy był test możliwości uruchomienia symulacji charakteryzujących się topologią wielowymiarowych torusów na komputerach, które taką topologię połączeń międzywęzłowych realizują w sposób sprzętowy – Blue Gene/P [11] oraz Blue Gene/Q [12][13]. Z przyczyn czysto technicznych okazało się to jednak nie do zrealizowania. Na komputerach klasy Blue Gene występował problem z prawidłową kompilacją PGENESIS. Nie było problemów z uruchomieniem samego GENESIS – wersja sekwencyjna jednak w przypadku analiz potrzebnych do wykonania na potrzeby artykułu jest niewystarczająca. Oficjalnie do wersji 2.3.1 PGENESIS dodano plik makefile współpracujący jedynie z pierwszą generacją Blue Gene/L [14] – bez modyfikacji nie działa on jednak poprawnie na nowszych maszynach. Z powodu tych komplikacji zostaliśmy zmuszeni do wykorzystania klastra komputerowego x86\_64 HP BladeSystem/Actina, Hydra. Być może w przyszłości uda się poprawnie skompilować PGENESIS na nowszych Blue Gene dzięki czemu możliwy będzie powrót do tematu.

Podczas badań okazało się, że rozmiar symulowanej sieci w przypadku torusa o określonej liczbie wymiarów nie wpływa w zauważalny sposób na czas symulacji. Dla przykładu symulacja sieci o topologii torusa 2D i wymiarze 2x2 wykonywała się ok. 391 minut. Dla wymiaru 24x24 operacja symulacji zajęła już ok. 484 minut, ale dla 26x26 zaledwie ok. 481 minut.

W trakcie formowania założeń do artykułu zakładano, że PGENESIS będzie działało prawidłowo na dowolnej liczbie procesorów – przynajmniej do rozmiarów jednej partycji złożonej ze 120 węzłów posiadających po dwa procesory Intel Xeon X5660, co daje łącznie 1440 rdzeni obliczeniowych. W praktyce okazało się jednak, że już przy wykorzystaniu 60 węzłów tego typu (720 rdzeni obliczeniowych) symulacja kończy się błędem synchronizacji procesów i przedwczesnego zakończenia przynajmniej jednego z nich. Największą symulację prawidłowo udało się zakończyć na 676 rdzeniach (57 węzłów). Nie zdiagnozowano co dokładnie było problemem – samo GENESIS, środowisko programowe na klastrze Hydra czy też sprzęt. Nie wykluczone, że na innym klastrze komputerowym symulacje o większych rozmiarach przebiegłyby prawidłowo. Autorom nie udało się odnaleźć informacji o symulacjach cechujących się rozmiarem sieci ponad 1,5 miliarda neuronów (i kilka bilionów połączeń między nimi) wykonanych za pomocą symulatora PGENESIS.

Najciekawszym efektem pracy jest jednak uzasadnienie, że PGENESIS nie limituje się na możliwościach sieci łączącej węzły (operacje I/O), a możliwościach obliczeniowych współczesnych procesorów. W badaniach uwzględniono jedynie interfejsy Infiniband QDR/DDR oraz Ethernet 10 Gb. Nie wiadomo jak sprawa wyglądałaby podczas zastosowania znacznie wolniejszych standardów Ethernet 1 Gb, które nadal często stosowane w taniach klastrach typu Beowulf [15].

Warto także uściślić ułomną naturę pomiarów, które wy-

konano. W przypadku gdy alokowano całe węzły to oczywiście symulacja wykorzystywała w pełni ich możliwości obliczeniowe. Jednak gdy rezerwowano jedynie część węzła/węzłów to na pozostałej części mogły (ale nie musiały) wykonywać się zadania innych użytkowników. Oczywiście w pewnym stopniu wpłynęły one na czas przeprowadzonych symulacji. W przypadku gdy na węzle rezydowałyby tylko nieliczne procesy to przy procesorach Intel Xeon X5660 istniałaby dodatkowo możliwość automatycznego wykorzystania technologii Turbo Boost [16], która zwiększa częstotliwość taktowania wykorzystywanych rdzeni.

## Podziękowania

Obliczenia wykonano w Interdyscyplinarnym Centrum Modelowania Matematycznego i Komputerowego (ICM) Uniwersytetu Warszawskiego w ramach grantu obliczeniowego nr G55-11.

## Literatura

1. Bower, James M., David Beeman. The book of GENESIS: exploring realistic neural models with the General NEural Simulation System. Electronic Library of Science, The, 1995.
2. Bower, James M., David Beeman. "Constructing realistic neural simulations with GENESIS." *Neuroinformatics*. Humana Press, (2007), 103-125.
3. Goddard, Nigel H., Greg Hood. "Large-scale simulation using parallel GENESIS." *The Book of GENESIS*. Springer New York, (1998), 349-379.
4. Hood, Greg. "Using P-GENESIS for Parallel Simulation of GENESIS Models." *Brains, Minds and Media* 1.2 2005.
5. Geist, Al, ed. PVM: Parallel virtual machine: a users' guide and tutorial for networked parallel computing. MIT press, 1994.
6. Sunderam, Vaidy S. "PVM: A framework for parallel distributed computing." *Concurrency: practice and experience* 2.4 (1990), 315-339.
7. Gropp, William, Ewing Lusk, and Anthony Skjellum. *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. MIT press, 1999.
8. de Hoon, Michiel JL, et al. "Open source clustering software." *Bioinformatics* 20.9 (2004), 1453-1454.
9. Pfister, Gregory F. "An introduction to the InfiniBand architecture." *High Performance Mass Storage and Parallel I/O* 42 (2001), 617-632.
10. Liu, Jiuxing, Jiesheng Wu, and Dhableswar K. Panda. "High performance RDMA-based MPI implementation over InfiniBand." *International Journal of Parallel Programming* 32.3 (2004), 167-198.
11. Almasi, Gheorghe, et al. "Overview of the IBM Blue Gene/P project." *IBM Journal of Research and Development* 52.1-2 (2008), 199-220.
12. Haring, Ruud A., et al. "The ibm blue gene/q compute

chip." *Micro*, IEEE 32.2 (2012), 48-60.

13. Chen, Dong, et al. "The IBM Blue Gene/Q interconnection network and message unit." *High Performance Computing, Networking, Storage and Analysis (SC)*, 2011 International Conference for. IEEE, 2011.
14. Gara, Alan, et al. "Overview of the Blue Gene/L system architecture." *IBM Journal of Research and Development* 49.2.3 (2005), 195-212.
15. Becker, Donald J., et al. "BEOWULF: A parallel workstation for scientific computation." *Proceedings, International Conference on Parallel Processing*. Vol. 95. 1995.
16. Charles, James, et al. "Evaluation of the Intel® Core™ i7 Turbo Boost feature." *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*. IEEE, 2009.

## Acknowledgment

Interdisciplinary Centre for Mathematical and Computational Modeling (ICM), Warsaw University, Poland is acknowledged for providing the computer facilities under the Grant No. G55-11.