

## PL-Grid - polska infrastruktura przetwarzania danych

*PL-Grid – polish data computing infrastructure*

**Monika Kwiatkowska<sup>1</sup>, Łukasz Świerczewski<sup>2</sup>**

**Treść.** Praca prezentuje polską infrastrukturę przetwarzania danych naukowych PL-Grid. W pracy zaprezentowano zarówno teoretyczny zarys możliwości, jak i praktyczne pomiary empiryczne, które wykonano na platformie. Podczas testów wykorzystano algorytm wyszukiwania nieparzystych liczb dziwnych oraz program poszukujący kontrprzykładu dla hipotezy Brocard'a. Wykonano także proste symulacje algorytmu kwantowego Shora z wykorzystaniem udostępnionych akceleratorów graficznych wspierających technologię CUDA.

**Słowa kluczowe:** przetwarzanie danych, grid, PL-Grid, wirtualne laboratorium.

**Abstract.** This paper presents polish scientific data computing infrastructure PL-Grid. Paper shows both theoretical outline of the possibilities and practical empiric measurements, which were made via platform. During test there were used: odd weird numbers search algorithm and Brocards conjecture counterexample search program. There were also taken some simple simulations of Shor's quantum algorithm using shared graphics accelerators supporting CUDA.

**Key words:** computing, grid, PL-Grid, virtual laboratory.

### 1. Wstęp

Prace nad utworzeniem polskiej infrastruktury PL-Grid rozpoczęto w 2009 roku. W tym roku Akademickie Centrum Komputerowe CYFRONET działające przy AGH w Krakowie zaproponowało innym centrom danych utworzenie krajowego, jednolitego systemu, który miałby służyć polskiej społeczności naukowej.

Infrastrukturą PL-Grid zarządza Konsorcjum PL-Grid, które składa się z następujących instytucji:

- ACK CYFRONET AGH – Akademickie Centrum Komputerowe CYFRONET AGH w Krakowie,
- ICM UW – Interdyscyplinarne Centrum Modelowania Matematycznego i Komputerowego w Warszawie,
- PCSS – Instytut Chemii Bioorganicznej PAN - Poznańskie Centrum Superkomputerowo Sieciowe w Poznaniu,
- CI TASK – Centrum Informatyczne Trójmiejskiej Akademickiej Sieci Komputerowej w Gdańsku,
- WCSS – Wrocławskie Centrum Sieciowo - Superkomputerowe we Wrocławiu.

Wyżej wymienione instytucje w ramach projektu udostępniają naukowcom swoje zasoby obliczeniowe.

Polska Infrastruktura Gridowa jest elementem EGI (ang. European Grid Initiative) [1]. Celem EGI jest integracja narodowych struktur w jeden system.

### 2. Zasoby obliczeniowe

PL-Grid udostępnia w sumie ponad 230 TFlopsów mocy obliczeniowej oraz ponad 3,6 PBajtów miejsca na dyskach twardych. Użytkownicy gridu uzyskują dostęp do maszyn z pamięcią vSMP [2], klastrów komputerowych oraz nowoczesnych kart graficznych wspierających obliczenia CUDA [3]. Zasoby obliczeniowe udostępnione przez ACK CYFRONET oraz CI TASK przedstawiono w Tab. 1.

Jak widać w pierwszym centrum danych możemy odnaleźć aż 1306 węzłów obliczeniowych. W CI TASK do dyspozycji użytkowników jest nieco mniej - 226 węzłów obliczeniowych.

1. Uniwersytet Marii Curie-Skłodowskiej w Lublinie, Wydział Matematyki, Fizyki i Informatyki, pl. Marii Curie - Skłodowskiej 5, 20-031 Lublin

2. Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości, Instytut Automatyki i Robotyki, ul. Akademicka 14, 18-400 Łomża

Tab. 1. Zasoby obliczeniowe udostępnione w ramach projektu PL-Grid przez ACK CYFRONET oraz CI TASK.

Tab. 1. Computing resources shared under PL-Grid project by ACK  
CYFRONET and CI TASK.

Ośrodek	Nazwa systemu	Liczba węzłów	Liczba procesorów w węźle	Liczba rdzeni w procesorze	Liczba rdzeni w węźle	GHz	Wielkość pamięci węzła
ACK CYFRONET AGH	Zeus	256	2	4	8	2,5	16 GB
		256	2	6	12	2,26	16 GB
		342	2	6	12	2,66	24 GB
		234	2	6	12	2,4	24 GB
		6	2	6	12	2,4	192 GB
	Zeus vSMP	64	2	6	12	2,66	96 GB
	Zeus GPGPU	24	2	6	12	2,93	72 GB
		20	2	6	12	2,4	96 GB
	Zeus BigMem	104	4	16	64	2,3	256 GB
CI TASK	Galera PLUS	192	2	6	12	2,27	16 GB
		2	32	6	192	2,93	1000 GB
		32	2	6	12	2,93	72 GB

### 3. Uczestnictwo i granty

Aby stać się użytkownikiem systemu PL-Grid należy być naukowcem z tytułem co najmniej doktora. Konto mogą uzyskać także np. studenci, którzy zostaną zarejestrowani jako podopieczni przypisanego im pracownika naukowego. Aby wykonywać obliczenia naukowe trzeba mieć aktywny grant obliczeniowy. Po przydzieleniu konta użytkownik automatycznie (bez jakichkolwiek dodatkowych formalności) otrzymuje grant osobisty, który uprawnia do wykorzystania 1000 godzin obliczeniowych i 40 GB pamięci masowej. Grant ten podlega odnowieniu co 6 miesięcy. Jeżeli zasoby udostępnione w ramach grantu osobistego nie są wystarczające można wystąpić o przydzielenie grantu właściwego. Grant ten podlega rozliczeniom co ok. 6 miesięcy oraz po jego zakończeniu. W przypadku zakończenia grantu użytkownik ma obowiązek poinformowania PL-Grid o wynikach badań naukowych oraz publikacjach, które dzięki obliczeniom powstały.

### 4. Wirtualne laboratoria

PL-Grid umożliwia także dostęp do tzw. gridów dziedzinowych z bioinformatyki lub chemii. Użytkownicy zainteresowani bioinformatyką mogą uzyskać dostęp do usługi dotyczącej analizy danych genetycznych. Pozwala ona na uruchomienie programów służących do analizy danych bioinformatycznych, które zostały uzyskane za pomocą sekwenatora genomowego. W ramach PLGrid ICM udostępnia także usługę umożliwiającą długoterminowo i bezpiecznie przechowywać duże ilości informacji genetycznych. System archiwizacji jest oparty o rozwiązanie

rozproszone Platon U4, które zapewnia zarówno wysoką niezawodność, jak i trwałość przechowywanych danych. Na platformie PL-Grid odnajdziemy także usługę InSili-coLab [4] dzięki, której możliwe jest uruchomienie złożonych eksperymentów obliczeniowych z chemii kwantowej.

Użytkownik może uzyskać także dostęp do wielu innych programów naukowych takich jak: Wolfram Mathematica [5], MATLAB [6], ANSYS Fluent [7] lub nawet Blender [8].

### 5. Liczby Weirda - klastry komputerowe

Większość zasobów obliczeniowych jakie są udostępniane w ramach PL-Grid obejmuje klastry komputerowe [9]. Programowanie tego typu komputerów jest dużo trudniejsze od pisania programów na maszyny z pamięcią wspólną – wymagana jest znajomość specjalnego środowiska MPI.

W teorii liczb liczba dziwna [11] (ang. weird number) to liczba naturalna, której suma wszystkich dzielników jest większa od jej samej. Dodatkowym warunkiem jest to, że suma wszystkich lub niektórych dzielników nie może być równa tej liczbie. Najmniejszą liczbą dziwną jest 70, ponieważ jej dzielniki to: 1, 2, 5, 7, 10, 14 i 35, a ich suma wynosi 74 i nie ma takiego podzbioru dzielników, których suma jest równa 70. Liczbą weirda nie jest natomiast 12, której dzielniki to 1, 2, 3, 4, 6. Ich suma wynosi 16, jednak 2+4+6 wynosi 12. Do dnia dzisiejszego nie odnaleziono żadnej nieparzystej liczby dziwnej i nie wiadomo czy takie istnieją. Spadek czasu realizacji przeszukiwania zakre-

su  $10^{11}$  liczb w poszukiwaniu nieparzystej liczby dziwnej dla różnej ilości procesów na klastrze komputerowym PL-Grid WCSS Supernova przedstawiono w Tab. 2.

Tab. 2. Spadek czasu realizacji przeszukiwania zakresu  $10^{11}$  liczb w poszukiwaniu nieparzystej liczby dziwnej dla różnej ilości procesów na klastrze komputerowym PL-Grid WCSS Supernova.

Tab. 2. The decrease of search time of numbers from range of  $10^{11}$  in search of an odd weird number for different numbers of processes on a computer cluster PL-Grid WCSS Supernova.

Ilość procesów	Czas realizacji obliczeń
1	6241 s
2	3142 s
3	2765 s
4	1674 s
8	967 s
16	521 s
24	411 s
32	282 s
40	252 s
48	223 s
56	198 s
64	176 s
128	89 s

Program ten zrównoleglił się prawie idealnie. Doskonale widać to podczas porównywania wydajności uzyskanej przy 64 i 128 procesach. Dla 64 procesów czas realizacji zadania wyniósł 176 sekund, a dla 128 spadł niemal dwukrotnie – do 89 sekund.

## 6. Symulacja algorytmu Shora - akceleratory graficzne

Informatyka kwantowa jest dość nową gałęzią informatyki. Niedawno, w 1994 roku Peter Shor zaprezentował kwantowy algorytm faktoryzacji o złożoności wielomianowej. W pracy [12] zaprezentowano możliwość przeprowadzenia symulacji algorytmu Shora na kartach graficznych wspierających standard CUDA oraz OpenCL [13]. Wyniki uzyskane w pracy [12] (sześć pierwszych kart graficznych) oraz pomiary uzyskane na platformie PL-Grid zaprezentowano w Tab. 3.

Tab. 3. Czas wykonywania faktoryzacji liczby 711 za pomocą symulacji algorytmu Shora z wykorzystaniem OpenCL (dla AMD) oraz CUDA (dla nVidii). Źródło: [12] rozwinęte o wyniki własne uzyskane na platformie PL-Grid.

Tab. 3. Time of factoring number 711 with help of Shor's algorithm simulation, using OpenCL (for AMD) and CUDA (for nVidia). Source: [12] expanded by own results gained on the PL-Grid platform.

	Czas minimalny	Czas średni	Czas maksymalny
nVidia Tesla C2050	7,12 s	24,568 s	832,80 s
nVidia Tasia C1060	8,08 s	26,342 s	976,50 s
nVidia GeForce GTS250	12,67 s	58,786 s	1864,32 s

nVidia GeForce 9800 GT	58,09 s	276,895 s	2654,44 s
nVidia GeForce 9800 GX2	60,84 s	290,438 s	2743,01 s
AMD Radeon 7770	5,54 s	25,875 s	743,76 s
PLGrid CYFRONET Zeus GPGPU nVidia Tesla M2050	7,09 s	22,785 s	825,97 s
PLGrid CYFRONET Zeus GPGPU nVidia Tesla M2090	7,11 s	22,568 s	876,82 s
PLGrid ICM UW Hydra nVidia GeForce GTX 480	8,06 s	29,765 s	1007,98 s
PLGrid ICM UW Hydra nVidia Tesla K20	5,26 s	20,502 s	792,43 s

Najnowocześniejszą kartą graficzną w zestawieniu jest nVidia Tesla K20 dostępna w ICM UW i to ona uzyskała zdecydowanie najlepsze wyniki czasowe. Nieco gorzej (ale i tak najlepiej zaraz za K20) wypadły akceleratory graficzne nVidia Tesla M2050 oraz nVidia Tesla M2090 zainstalowane w klastrze komputerowym Zeus GPGPU. Nieco słabszą kartą graficzną jest GeForce GTX 480 – okazał się niewiele wolniejszy od nVidia Tesla C1060.

## 7. Hipoteza Brocard'a

W teorii liczb hipoteza Brocard'a mówi, że zawsze można znaleźć co najmniej cztery liczby pierwsze pomiędzy  $(p_n)^2$ , a  $(p_{n+1})^2$  dla  $n > 1$  gdzie  $p_n$  określa  $n$ -tą liczbę pierwszą. Przykładowe dane prezentujące hipotezę dla pierwszych wartości  $n$  przedstawiono w Tab. 4.

Tab. 4. Przykładowe dane prezentujące hipotezę Brocard'a.  
Tab. 4. Example data presenting Brocards conjecture.

n	$p_n$	$(p_n)^2$	$p_{n+1}$	$(p_{n+1})^2$	Liczby pierwsze	Ilość liczb pierwszych
1	2	4	3	9	5, 7	2
2	3	9	5	25	11, 13, 17, 19, 23	5
3	5	25	7	49	29, 31, 37, 41, 43, 47	6
4	7	49	11	121	53, 59, 61, 67, 71	5

Napisane oprogramowanie do obliczeń wykorzystuje listę liczb pierwszych. Lista ta na początku jest generowana za pomocą sita Eratostenesa. Krok ten jest najbardziej wymagającym obliczeniowo etapem tego programu. Także głównym ograniczeniem jest sama lista liczb pierwszych, ponieważ program dla dużych parametrów wejściowych potrzebuje bardzo dużo pamięci operacyjnej. Szczegółowe porównanie systemu PL-Grid CI TASK Galera PLUS vSMP z komputerem UMCS opartym o procesor Intel Core i7 950 i wykorzystującym 24 GB pamięci RAM podczas analizy hipotezy Brocard'a zaprezentowano w Tab. 5.

Tab. 5. Wyniki czasowe jakie uzyskano na systemie PL-Grid CI TASK Galera PLUS oraz komputerze UMCS z procesorem Intel Core

i7 950 podczas analizy hipotezy Brocard'a.

Tab. 5. *The results obtained at the PL-Grid CI TASK Galera PLUS system and UMCS computer with Intel Core i7 950 while analyzing Brocards hypothesis.*

Rozmiar wykorzystanej pamięci	PLGrid CI TASK Galera PLUS vSMP	UMCS Intel Core i7 950 24 GB RAM	Rozmiar sita Eratostenesa	Ostatnia liczba n, dla której sprawdzono hipotezę
1 GB	119 s	121 s	8589934592	8951
5 GB	665 s	670 s	42949672960	18566
7 GB	948 s	958 s	60129542144	21551
15 GB	1244 s	2166 s	128849018880	30680
22 GB	3251 s	3294 s	188978561024	36536
50 GB	11999 s	-	429496729600	53227
75 GB	> 259200 s	-	644245094400	-
100 GB	> 259200 s	-	858993459200	-

Jak widać pomimo nieznacznie niższej częstotliwości taktowania (2,93 GHz) procesor zainstalowany w systemie vSMP klastra komputerowego Galera PLUS udostępnionego w ramach PL-Grid okazał się szybszy od komputera opartego na procesorze Core i7 950 (3.07 GHz). Jednak pomimo posiadania teoretycznie aż 1000 GB pamięci (dla użytkownika dostępne ok. 920 GB) okazało się, że operowanie na tak dużych ilościach danych nie jest możliwe. Wygenerowanie sita Eratostenesa już dla 75 GB wykroczało poza czas kolejki, który wynosi 72 godziny. W czasie 72 godzin każde zadanie na systemie vSMP powinno się zakończyć – w przeciwnym wypadku zostanie automatycznie zabite przez platformę. Kod odpowiedzialny za sito Eratostenesa zaprezentowano na Listingu 1.

Listing. 1. Możliwa implementacja sita Eratostenesa z wykorzystaniem operacji bitowych.

Listing. 1. *Possible implementation of the sieve of Eratosthenes using bit operations*

```
void eratosthenes_sieve_bit(char numbers[], unsigned long long int limit)
```

```
{
    unsigned long long int i, j;

    clear_bit(numbers[0], 0);
    clear_bit(numbers[0], 1);

    for (i=2; i<limit; i++)
    {
        set_bit(numbers[i/8], i%8);
    }

    for (j=4; j<limit; j+=2)
    {
        clear_bit(numbers[j/8], j%8);
    }

    for (i=3; i<sqrt(limit); i+=2)
    {
        if ( check_bit(numbers[i/8], i%8) )
        {
            for (j=2*i; j<limit; j+=i)
            {
                clear_bit(numbers[j/8], j%8);
            }
        }
    }
}
```

```
}
}
}
}
```

W kodzie wykorzystano makra `clear_bit()`, `set_bit()` oraz `check_bit()` umożliwiające wykonywanie operacji na bitach. Najprawdopodobniej system vSMP źle sobie radzi z tego typu sekwencyjnym odczytem/zapisem do pamięci i dlatego wynikł tak długi okres realizacji zadań. Należy pamiętać, że platforma vSMP to głównie oprogramowanie wirtualizacyjne, które umożliwia utworzenie jednego komputera z dużą wirtualną pamięcią, z dziesiątek zwykłych serwerów. Dane, które nie są dostępne w ramach danego węzła obliczeniowego muszą być pobrane za pośrednictwem znacznie wolniejszej sieci z innej maszyny.

## 8. Wnioski

PL-Grid stanowi bardzo ciekawą propozycję dla współczesnego naukowca, który często do pracy może wykorzystywać komputery o dużych mocach obliczeniowych. Praca prezentuje aspekt wykorzystania tej platformy i zarówno pod względem symulacji algorytmu kwantowego Shora, jak i poszukiwania nieparzystej liczby dziwnej uzyskano interesujące rezultaty czasowe. Pod znakiem zapytania można jedynie pozostawić wykorzystanie systemu vSMP do rozwiązywania hipotezy Brocard'a. Należy jednak dodać, że Infrastruktura PL-Grid umożliwia wykorzystanie dwóch systemów vSMP – zainstalowanego w klastrze CI TASK Galera PLUS oraz ACK CYFRONET Zeus. Przetestowano jedynie możliwości pierwszego systemu z całkowitym pominięciem możliwości systemu Zeus, który być może byłby szybszy. Ciekawe podejście dotyczące przemysłu rolno-spożywczego (centra informacji) chociaż z trochę odmiennego punktu widzenia zaprezentowano w [14].

## Podziękowania

Praca została wykonana z wykorzystaniem Infrastruktury PL-Grid.

Praca została przygotowana z wykorzystaniem zasobów superkomputerowych udostępnionych przez Wydział Matematyki, Fizyki i Informatyki Uniwersytetu Marii Curie-Skłodowskiej w Lublinie.

## Literatura (References)

- [1] D. Kranzlmüller, J. Marco de Lucas i P. Öster. "The European Grid Initiative (EGI)." *Remote Instrumentation and Virtual Laboratories*. Springer US, 2010, s. 61-66.
- [2] Schmidl, Dirk, et al. "How to scale nested openmp applications on the scalemp VSMP architecture." *Cluster*

- Computing (CLUSTER), 2010 IEEE International Conference on*. IEEE, 2010.
- [3] J. Nickolls, et al. "Scalable parallel programming with CUDA." *Queue* 6.2 (2008): 40-53.
- [4] J. Kocot, et al. "InSilicoLab—Managing Complexity of Chemistry Computations." *Building a National Distributed e-Infrastructure—PL-Grid*. Springer Berlin Heidelberg 2012, s. 265-275.
- [5] S. Wolfram, *Mathematica: a system for doing mathematics by computers*. Addison-Wesley, Reading, 1988.
- [6] M. S. Grewal i Angus P. Andrews. *Kalman filtering: theory and practice using MATLAB*. Wiley. com, 2011.
- [7] A. Fluent, "12.0 User's Guide." *User Inputs for Porous Media* (2009): 6.
- [8] R. Hess, *The essential Blender: guide to 3D creation with the open source suite Blender*. No Starch Press, 2007.
- [9] R. Buyya, "High Performance Cluster Computing: Architectures and Systems (Volume 1)." *Prentice Hall, Upper SaddleRiver, NJ, USA* 1 (1999), s. 999.
- [10] S. J. Benkoski i P. Erdős, "On weird and pseudoperfect numbers." *mathematics of computation* 28.126 (1974), s. 617-623.
- [11] Vandersypen, Lieven MK, et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." *Nature* 414.6866 (2001): 883-887.
- [12] Ł. Świerczewski, Symulacja funkcjonalnego systemu kwantowego na równoległych komputerach klasycznych IV generacji. (2013).
- [13] J. E. Stone, D. Gohara i Guochun Shi. „OpenCL: A parallel programming standard for heterogeneous computing systems." *Computing in science & engineering* 12.3 (2010), s. 66.
- [14] H. Zarzycki, E. Fronczak, *A Practical Approach To Computer System Architecture For An Agro-Food Industry Information Center*. PSZW, nr 33, s. 248-255, Bydgoszcz 2010.