

Intel Manycore Testing Lab - środowisko sprzętowo-programowe do dydaktyki tworzenia i testowania efektywności równolegliczacji oprogramowania

Intel Manycore Testing Lab - hardware and software environment focused on didactic of development and efficiency testing in software paralleling

Łukasz Świerczewski¹

Treść. Współczesny proces dydaktyczny technik programowania często wymaga dostępu zarówno do nowoczesnego sprzętu, jak i oprogramowania. W szczególnej mierze odnosi się to do algorytmów równoległych, których odpowiednie właściwości w dużo większym stopniu można zaobserwować na wydajnych procesorach nowej generacji. Aby stworzyć międzynarodową społeczność akademicką związaną z tą specjalizacją firma Intel udostępniła wirtualne laboratorium testowe (Manycore Testing Lab - MTL). Artykuł przedstawia aspekt architektury oraz praktycznego zastosowania MTL w pracy wieloużytkowej i skupia się na empirycznym potwierdzeniu wzrostu wydajności uzyskanej dzięki programowaniu równoległemu i 10-rdzeniowym procesorom Westmere-EX. Badaniom objęto cztery klasy algorytmów: czysto matematyczny dotyczący problemu Collatza, kryptograficzny 3DES, kwantowy algorytm Grovera oraz klasyczny algorytm genetyczny. Dla zastosowań edukacyjnych dostęp do laboratorium jest bezpłatny, a udostępniane platformy wspierają wszelkie zaawansowane technologie.

Słowa kluczowe: wirtualne laboratorium, wirtualny eksperyment, programowanie równoległe, Manycore Testing Lab

Abstract. The modern didactic process of programming techniques often requires access to the modern hardware and software. In a particular part applies to parallel algorithms, where appropriate properties to a much greater extent can be seen in the new generation of high-performance processors. To create an international academic community associated with this specialization, Intel released a virtual test lab (Manycore Testing Lab - MTL). The paper presents the architectural aspect and the practical application of MTL at work reusable and focuses on empirical confirmation gains obtained through parallel programming and 10-core Westmere-EX processors. The study consisted of four classes of algorithms: for a purely mathematical problem Collatz, 3DES cryptography, quantum Grover algorithm and the classic genetic algorithm. For educational access to the laboratory is free and available to all platforms support advanced technologies.

Key words: virtual laboratory, virtual experiments, parallel programming, Manycore Testing Lab

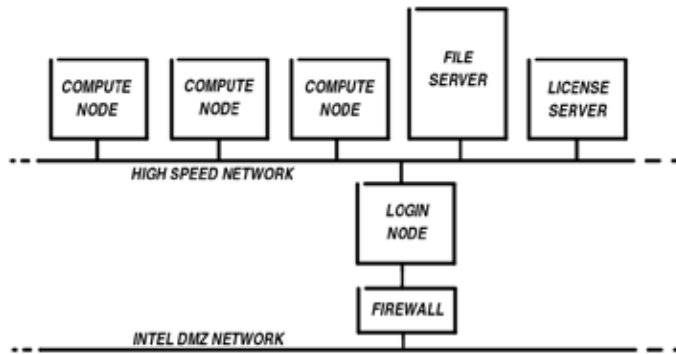
1. Wprowadzenie

Z dostępu do Manycore Testing Lab [1] korzysta ponad 160 uniwersytetów na całym świecie. Intel udostępnia wirtualne laboratoria działające w oparciu o różne systemy operacyjne: Linux lub Windows. Inicjatywa spotkała się z bardzo dużym zainteresowaniem także ze względu na dostęp do nowoczesnego środowiska programistycznego Parallel Studio ze wsparciem najnowszych kompilatorów zoptymalizowanych pod kątem procesorów Intel. Przejaw aktywności w świecie akademickim firmy Intel nie jest nowością. Już wcześniej korporacja nVidia umożliwiła nawiązywanie partnerstw z ośrodkami edukacyjnymi w ramach kształcenia studentów z wykorzystaniem programowalnych akceleratorów graficznych CUDA [14]. Idea nauczania wspierana przez międzynarodowe firmy może wskazywać gałęzie nauki i przemysłu, które będą się bardzo dynamicznie rozwijać.

Intel MTL cechuje budowa wielowęzłowa z wyróżnieniem charakterystycznych nodów logowania (węzeł dostępowy), obliczeniowych, magazynujących informację oraz odpowiedzialnych za zarządzanie licencjami oprogramowania. Schemat budowy systemu został przedstawiony na Ryc. 1.

Węzły obliczeniowe oraz węzeł dostępowy opierają się na platformie zbudowanej z czterech procesorów Intel Xeon E7-4860. Każdy z układów charakteryzuje się częstotliwością taktowania wynoszącą 2.26 GHz, która dzięki technologii Turbo Boost [12] może zostać maksymalnie podniesiona do 2.66 GHz. Pojedynczy procesor posiada 10 rdzeni, jednak za pomocą technologii Hyper-Threading [13] jest możliwa obsługa jednocześnie aż 20 wirtualnych wątków. Dla całej platformy złożonej z czterech procesorów możemy więc uzyskać równoległą obsługę aż 80 wątków. Należy jednak pamiętać, że technologia HT jedynie symuluje działanie tak dużej ilości procesorów. W

¹ Instytut Informatyki i Automatyki, Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości w Łomży, lswierczewski@pwsip.edu.pl



Ryc. 1. Schemat budowy Intel Manycore Testing Lab.
Ryc. 1. Diagram of the Intel Manycore Testing Lab.

praktyce uzyskany wzrost wydajności pomiędzy 40, a 80 wątkami na tego typu platformie może być minimalny lub nawet w skrajnych przypadkach ujemny. Całość uzupełnia 64 lub 256 GB (w zależności od rodzaju węzła) szybkiej pamięci operacyjnej DDR3 o częstotliwości taktowania 1066 MHz.

Na węzłach zainstalowane są systemy operacyjne Windows lub Linux. Wszystkie obliczenia zaprezentowane w tym artykule wykonano na węźle działającym pod kontrolą systemu operacyjnego Linux i wyposażonym w 256 GB pamięci operacyjnej.

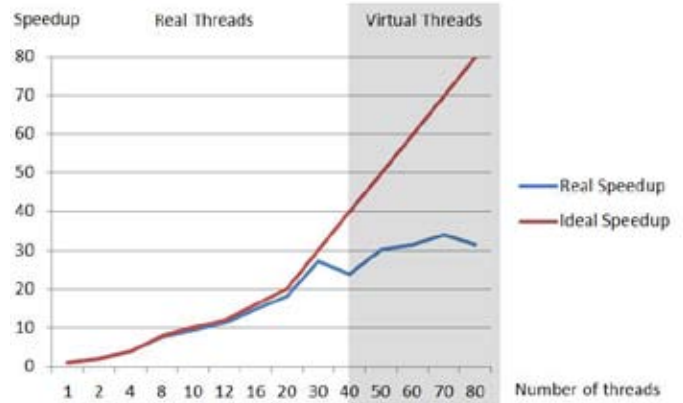
Podczas analiz algorytmów wykorzystano autorską bibliotekę Olib [2] [3]. Zostały w niej zaimplementowane wydajne algorytmy z wykorzystaniem środowiska OpenMP [4], a także nVidia CUDA. Intel MTL wspiera jedynie oprogramowanie pisane z myślą o komputerach równoległych z pamięcią wspólną więc w pracy zostanie wykorzystane jedynie środowisko OpenMP.

2. Analizy wydajnościowe Intel MTL

Pierwszym algorytmem jaki wzięto pod uwagę podczas testowania możliwości wirtualnego laboratorium firmy Intel jest słynny problem z teorii liczb: hipoteza Collatza [5]. Mówi ona, że zaczynając od dowolnej liczby naturalnej i generując kolejne wartości ciągu za pomocą wzoru (1) zawsze dojdziemy do liczby 1.

$$c_{n+1} = \begin{cases} \frac{1}{2}c_n & \text{gdy } c_n \text{ jest parzysta} \\ 3c_n + 1 & \text{gdy } c_n \text{ jest nieparzysta} \end{cases} \quad (1)$$

Problem nie został do dnia dzisiejszego rozwiązany pomimo, że wykonano wiele analiz komputerowych [6] [7]. Wyniki zrównoleglenia dla algorytmu Collatza przedstawiono graficznie na Ryc. 2. Bardziej szczegółowe dane można odnaleźć w Tab. 1.



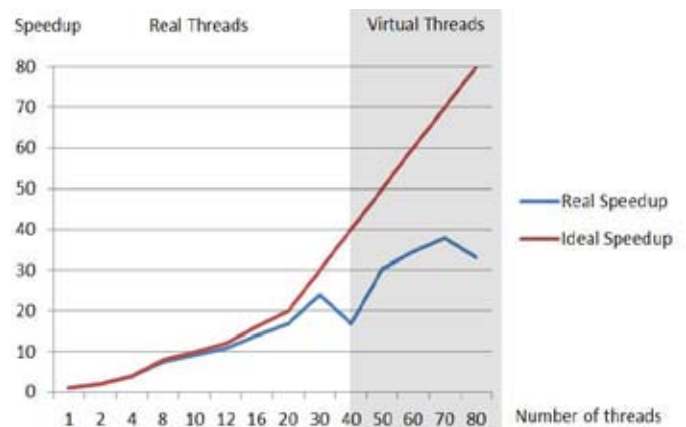
Ryc. 2. Przedstawienie wyników równolegliczności dla problemu Collatza oraz platformy Intel MTL

Ryc. 2. The presentation of the results for the Collatz conjecture and Intel MTL platform parallelization problem

Tab. 1. Tabela przedstawiająca dokładne czasy wykonywania algorytmu rozwiązującego problem Collatza dla różnych ilości wątków na platformie Intel MTL

Tab. 1. Table which shows the exact execution time of the algorithm solving the Collatz conjecture for a different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	782,57	1
2	391,933	1,996693312
4	198,4	3,944405242
8	102,235	7,65461926
10	83,655	9,354730739
12	69,109	11,32370603
16	52,953	14,77857723
20	43,05	18,17816492
30	28,71	27,25774991
40	32,749	23,89599682
50	25,882	30,2360714
60	24,866	31,47148717
70	23,054	33,94508545
80	24,902	31,42598988



Ryc. 3. Przedstawienie wyników równolegliczności dla algorytmu 3DES oraz platformy Intel MTL

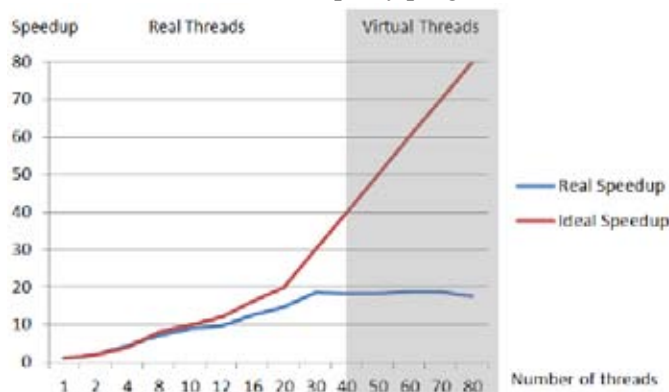
Ryc. 3. The presentation of the results of the 3DES algorithm and Intel MTL platform parallelization

Kolejnym rodzajem algorytmu był Triple-DES [8]. W bibliotece Olib można odnaleźć implementację 3DES ECB umożliwiającą szyfrowanie bloków o rozmiarze 64-bit w sposób równoległy i całkowicie niezależny na wielu procesorach. Szyfrowany plik miał rozmiar 512 MB. Rezultaty zrównoleglenia kodu i rzeczywiste przyśpieszenie można zaobserwować na Ryc. 3. Poszczególne pomiary czasów realizacji szyfrowania zostały umieszczone w Tab. 2.

Tab. 2. Tabela przedstawiająca dokładne czasy wykonywania algorytmu 3DES dla różnych ilości wątków na platformie Intel MTL
 Tab. 2. Table showing the exact execution times of the 3DES algorithm for different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	2729,085	1
2	1374,803	1,985073498
4	710,359	3,841839126
8	361,902	7,540950313
10	298,679	9,137184067
12	253,223	10,77739779
16	196,869	13,86244152
20	161,698	16,87766701
30	113,643	24,01454555
40	161,69	16,87850207
50	89,382	30,5328254
60	79,076	34,51217816
70	72,011	37,89816834
80	82,319	33,15255287

Aktualnie dość młodą i szybko rozwijającą się gałęzią informatyki są algorytmy kwantowe. Nie zbudowano dotychczas w pełni funkcjonalnego komputera kwantowego jednak znamy kilka algorytmów, których działanie można symulować na współczesnych maszynach. Jednym z nich jest algorytm Grovera [9] [10]. Główną operacją która podlega zrównolegleniu podczas symulacji tego algorytmu jest przejście rejestru kubitów przez bramkę kwantową, które jest definiowane jako mnożenie wektora przez macierz. Dla platformy MTL symulację wykonano dla rejestru kwantowego o rozmiarze 16 kubitów co wymusza operowanie na wektorze o długości 2^{16} oraz macierzy o rozmiarze $2^{16} \times 2^{16}$. Podczas pracy program alokował aż 96



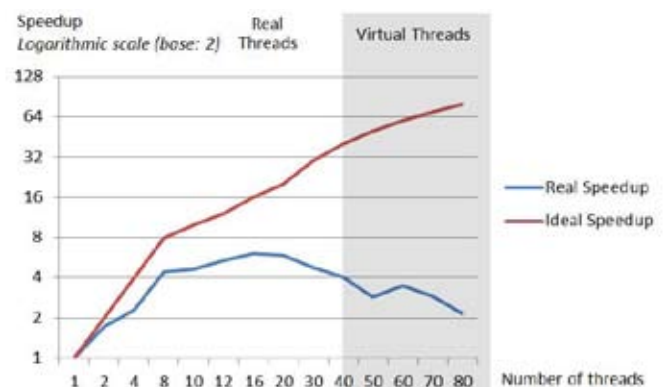
Ryc. 4. Przedstawienie wyników równoleglizacji dla symulacji kwantowego algorytmu Grovera oraz platformy Intel MTL
 Ryc. 4. The presentation of the results of simulation quantum Grover's algorithm and Intel MTL platform parallelization

GB pamięci operacyjnej. Wyniki zostały przedstawione na Ryc. 4. Bardziej szczegółowe dane można odnaleźć w Tab. 3.

Tab. 3. Tabela przedstawiająca dokładne czasy wykonywania symulacji kwantowego algorytmu Grovera dla różnych ilości wątków na platformie Intel MTL
 Tab. 3. Table which shows the exact execution times of simulation quantum Grover's algorithm for different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	7065,58	1
2	3670,555	1,924935058
4	1569,955	4,500498422
8	1017,367	6,944966762
10	792,365	8,917077357
12	728,098	9,704160704
16	565,739	12,48911601
20	484,207	14,59206496
30	384,308	18,38520145
40	386,253	18,29262168
50	387,401	18,23841446
60	376,061	18,78838805
70	378,095	18,68731403
80	400,883	17,62504272

Podczas badań wzięto także pod uwagę potencjalne możliwości zrównoleglenia algorytmu genetycznego [11]. Wykonane pomiary dotyczyły populacji o rozmiarze 16384, krzyżowania typu two-point z prawdopodobieństwem równym 50%. Prawdopodobieństwo zajścia mutacji bit inversion wynosiło 1%. Dodatkowo podczas analiz uwzględniono selekcję za pomocą koła ruletki i skalowanie liniowe. Rezultaty dla algorytmu genetycznego zostały przedstawione w Ryc. 5 i Tab. 4.

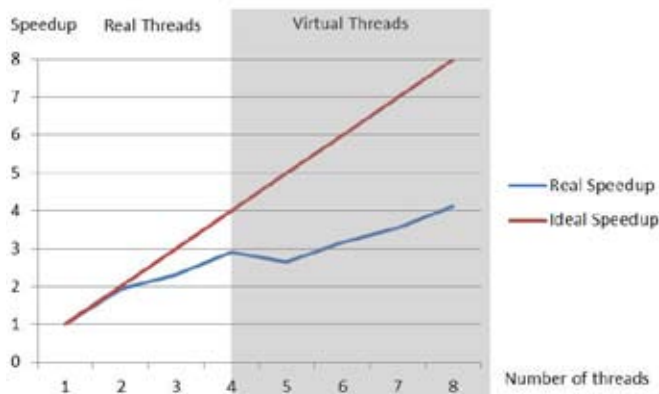


Ryc. 5. Przedstawienie wyników równoleglizacji dla realizacji algorytmu genetycznego oraz platformy Intel MTL
 Ryc. 5. The presentation of the results of a genetic algorithm and Intel MTL platform parallelization

Tab. 4. Tabela przedstawiająca dokładne czasy realizacji algorytmu genetycznego dla różnych ilości wątków na platformie Intel MTL
Tab. 4. Table which shows the exact execution times of a genetic algorithm for different number of threads on Intel MTL platform

Number of threads	Time [in seconds]	Real Speedup
1	9074	1
2	5276	1,719863533
4	3976	2,282193159
8	2037	4,454590083
10	1955	4,641432225
12	1678	5,407628129
16	1498	6,05740988
20	1539	5,896036387
30	1902	4,770767613
40	2241	4,04908523
50	3162	2,86970272
60	2601	3,488658208
70	3113	2,914873113
80	4218	2,15125652

Dla uzyskania dodatkowego punktu odniesienia względem innej platformy sprzętowej testy wydajności algorytmu genetycznego wykonano także na komputerze wyposażonym w jeden procesor Intel i7 920 i 24 GB pamięci operacyjnej DDR3. Wszystkie parametry algorytmu genetycznego pozostały takie same jak w przypadku platformy Intel MTL i czterech procesorów opartych o rdzeń Westmere-EX. Wyniki dla dodatkowej platformy zostały zaprezentowane na Ryc. 6 i Tab. 5.



Ryc. 6. Przedstawienie wyników równolegliczacji dla algorytmu genetycznego oraz platformy sprzętowej opartej o procesor Intel i7 920

Ryc. 6. The presentation of the results of a genetic algorithm and hardware platform based on the Intel i7 920 processor parallelization

Tab. 5. Tabela przedstawiająca dokładne czasy wykonywania algorytmu genetycznego dla różnych ilości wątków na platformie sprzętowej opartej o procesor Intel i7 920

Tab. 5. Table which shows the exact execution times of a genetic algorithm for different number of threads on the hardware platform based on the Intel i7 920 processor

Number of threads	Time [in seconds]	Real Speedup
1	7276	1
2	3753	1,938715694
3	3134	2,321633695
4	2507	2,902273634
5	2762	2,634322954
6	2294	3,171752398
7	2057	3,537190083
8	1767	4,117713639

3. Analiza możliwości równolegliczacji na platformie Intel MTL

Najwyższe przyspieszenia uzyskano podczas zrównoleglenia algorytmu 3DES i programu odpowiedzialnego za generowanie ciągu liczb według formuły Collatza. W pierwszym przypadku przyspieszenie równe ok. 37,89 wygenerowano podczas wykorzystania 70 wątków. W drugim przypadku maksymalne odnotowane przyspieszenie wyniosło ok. 33,94. Rezultat ten uzyskano także podczas intensywnego korzystania z 70 wątków. Wyniki te świadczą o potencjale jaki posiada technologia Hyper-Threading. Pomimo, że platforma posiada fizycznie 40 procesorów to intuicyjne uruchomienie programu z wykorzystaniem 40 wątków, daje gorsze wyniki niż w przypadku 70 wątków. Należy jednak pamiętać, że w tradycyjnym przypadku uruchomienie większej ilości wątków niż mamy dostępnych procesorów musi doprowadzić do ich przełączania między jednostkami obliczeniowymi przez planistę systemowego co zawsze skutkuje tylko spadkiem wydajności.

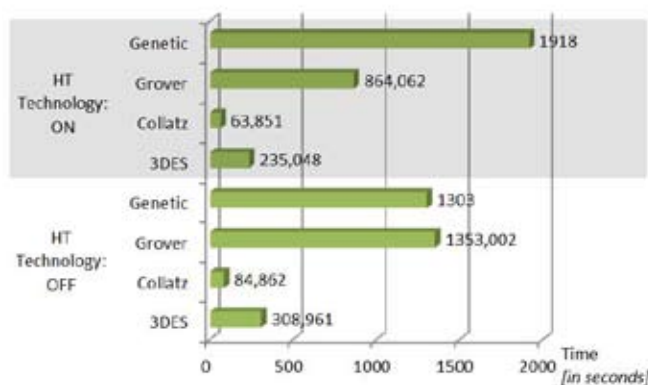
Uzyskane rezultaty nie wyglądają jednak tak dobrze w przypadku symulacji kwantowego algorytmu Grovera i algorytmu genetycznego. Dla algorytmu Grovera przyspieszenie wynoszące 18,78 uzyskano dla 60 wątków. Łatwo jednak zauważyć, że różnice między wynikami czasowymi dla przedziału od 30 do 70 wątków są minimalne. Na zaawansowanej platformie Intel MTL algorytm zachowuje się dość dziwnie. Przyspieszenie osiąga granicę już przy 16 wątkach z wartością zaledwie 6,05. Testy algorytmu z tego powodu przeprowadzono także na znacznie prostszej konfiguracji sprzętowej dostępnej dla przeciętnego użytkownika. Wyniki dotyczące procesora Intel i 7 920 ukazują, że podczas wykorzystania platformy złożonej tylko z jednego wielordzeniowego CPU trend wzrostu wydajności jest prawidłowy – najlepsze przyspieszenie rzędu 4,11 uzyskano przy wykorzystaniu wszystkich 8 wątków. Tak mało efektywne działanie algorytmu genetycznego może być spowodowane dość mało efektywną implementacją tego rozwiązania w bibliotece Olib.

Implementacja ta może okazać się wystarczająca dla komputerów, które posiadamy w domu jednak nie wykorzystują w pełni możliwości zaawansowanych i drogich platform jaką jest Intel MTL.

4. Wpływ technologii hyper-threading na czasy realizacji zadań

Podczas analiz sprawdzono także dokładnie wpływ technologii Hyper-Threading na wydajność. Firma Intel nie umożliwia użytkownikowi wyłączenia tej technologii z poziomu BIOSu. Ograniczono się więc do wykorzystania programu taskset dostępnego w środowisku systemu operacyjnego Linux. Umożliwia on zdefiniowanie na stałe z jakich procesorów dostępnych w systemie ma korzystać uruchamiany program. Na platformie złożonej z czterech procesorów posiadających fizycznie 10 rdzeni i wirtualizację 20 wątków uruchomiono oprogramowanie tak aby korzystało tylko z 10 wątków na procesor (w sumie 40 wątków dla całej maszyny). Takie ograniczenie powinno dać podobne wyniki jak wyłączenie wsparcia dla HT z poziomu BIOSu.

Wyniki takiego eksperymentu zaprezentowano na Ryc. 6. Jedyne algorytm genetyczny działał gorzej podczas symulowanego wyłączenia technologii Hyper-Threading. Wszystkie pozostałe algorytmy wykonywały się zauważalnie szybciej dzięki zastosowaniu HT.



Ryc. 6. Porównanie czasów realizacji przedstawionych algorytmów symulacją nieaktywnej i aktywnej technologii Hyper-Threading na Intel MTL

Ryc. 6. The comparison of execution times of a showed algorithms using inactive and active Hyper-Threading technology on Intel MTL

5. Wnioski końcowe

Zaprezentowane wyniki świadczą o dużych możliwościach wirtualnego laboratorium Intela. Jedyne działanie algorytmu genetycznego pozostawia wiele do życzenia. Algorytm 3DES z przyśpieszeniem w granicach 37,89 niewiele odbiega od zrównoleglenia doskonałego, dzięki któremu powinniśmy w tym przypadku uzyskać przyśpieszenie równe 40. Wysoce prawdopodobne, że problem z algorytmem genetycznym leży nie w wąskim gardle platformy, a w samym kodzie oprogramowania.

Laboratorium z całą pewnością umożliwia efektywną wizualizację możliwości programowania równoległego i nowoczesnych procesorów wielordzeniowych. Dostęp do MTL w trakcie kursów z technik programowania równoległego może znacząco podnieść ich atrakcyjność.

Literatura (References)

- [1] Intel Manycore Testing Lab, URL: <http://software.intel.com/en-us/articles/intel-many-core-testing-lab/> [ostatni dostęp: 10.07.2012 r.]
- [2] Olib Library, URL: <http://goldbach.pl/olib/> [ostatni dostęp: 10.07.2012 r.]
- [3] Ł. Świerczewski OLib - High Performance Library Containing Mathematical Functions & Algorithms. International Supercomputing Conference, Hamburg 2012.
- [4] R. ChandrA, R. Menon, L. Dagum, D. Kohr, D. Maydan, J. McDonald, Parallel Programming in OpenMP. Morgan Kaufmann 2000.
- [5] R. K. Guy, Unsolved problems in Number Theory. Springer (2004): 336–337.
- [6] J. Simons, B. De Weger, Theoretical and computational bounds for m-cycles of the $3n + 1$ problem, Acta Arithmetica 2005.
- [7] Tomas Oliveira e Silva, Computational verification of the $3x+1$ conjecture, URL: <http://www.ieeta.pt/~tos/3x+1.html>, [ostatni dostęp: 10.07.2012 r.]
- [8] U.S. DEPARTMENT OF COMMERCE / NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY: (1977) Data Encryption Standard (DES), Federal information processing standards publication
- [9] L. K. Grover, A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (1996): 212.
- [10] M.A. Nielsen, I.L. Chuang, Quantum computation and quantum information. Cambridge University Press 2000, Chapter 6.
- [11] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag 1999.
- [12] J. Charles, P. Jassi, N.S. Ananth, A. Sadat, A. Fedorova, Evaluation of the Intel® Core™ i7 Turbo Boost feature, Workload Characterization, 2009. IISWC 2009. IEEE International Symposium
- [13] J.R. Bulpin, A.I. Pratt, Multiprogramming performance of the Pentium 4 with Hyper-Threading, In the Third Annual Workshop on Duplicating, Deconstructing and Debunking (WDDD2004) held at ISCA '04. (2004): 53-62.
- [14] D. Krik, NVIDIA CUDA software and GPU parallel computing architecture, ISMM 2007.