

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Inżynierii Metali i Informatyki Przemysłowej



PRACA DYPLOMOWA INŻYNIERSKA

pt.

„Opracowanie analizy aspektu
oszczędności energii elektrycznej pod
kątem wykonania złożonych obliczeń
numerycznych”

Imię i nazwisko dyplomanta:

Grzegorz Maj

Kierunek studiów:

Informatyka Stosowana

Nr albumu:

261360

Promotor:

dr hab. inż. Łukasz Rauch

Recenzent:

dr hab. inż. Krzysztof Banaś

Podpis dyplomanta:

Podpis promotora:

Kraków 2017

„Upředzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także upředzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchylające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej "sądem koleżeńskim"”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.”

Kraków, dnia

Podpis dyplomanta.....

Spis treści

Wstęp	4
Część I – Teoretyczna	6
Green Computing	6
Charakterystyka użytych technologii	10
Charakterystyka sprzętu	12
Część II – Badania własne	15
Metody obliczeniowe, algorytmy	15
Analiza uzyskanych wyników	18
Całkowanie numeryczne metodą trapezów	20
Algorytm mnożenia macierzy przez macierz	27
Wnioski.....	31
Bibliografia.....	32
Spis tabel	33
Spis rysunków	33

Wstęp

W dzisiejszych czasach podstawą naszego rozwoju jest technologia, i to właśnie ona od kilkudziesięciu lat otacza nas właściwie na każdym kroku. Świat dzięki niej jest bardziej bezpieczny, działa sprawniej, szybciej i niezawodniej, lecz jest ona także powodem wielkiej rywalizacji. Ludzie żyją w ciągłym pośpiechu a czas jest na wagę złota dlatego też rośnie zapotrzebowanie na coraz szybszy dostęp do informacji oraz szybsze obliczenia i mogło by się wydawać, że to właśnie ten czynnik w rozwoju technologii jest najważniejszy, lecz z szybkością złożonych obliczeń rośnie również zużywana przy tym energia oraz generowane ciepło, a to z kolei prowadzi do częstszych awarii i ogólnej zawodności sprzętu.

Inżynierowie biorący udział w projekcie *Supercomputing in Small Spaces*, użyli równania Arrheniusa, które opisuje zależność między szybkością reakcji a temperaturą, które poparte również testami empirycznymi na swoim klastrze wykazało, że z każdym wzrostem temperatury komputera o 10 °C jego zawodność się podwajała[1]. Wydajność, długa żywotność oraz zapewniona ciągłość pracy sprzętu jest niezwykle ważna, a uzyskujemy ją zmniejszając zużycie energii poprzez zastosowane nowsze technologie oraz optymalizację co dodatkowo przejawia się jako korzyść dla ochrony środowiska dlatego temat ten jest tak bardzo istotny.

Celem niniejszej pracy jest przeprowadzenie analizy aspektu oszczędności energii elektrycznej przy wykonywaniu złożonych obliczeń numerycznych. Zlecone będą zadania obliczeniowe z wykorzystaniem frameworku pozwalającego na obliczenia na heterogenicznych platformach sprzętowych. Chcę do analizy wykorzystać superkomputer Prometeusz, który pozwala na monitorowanie zużycia energii na procesorach oraz kartach graficznych. Jeśli pozwoli na to infrastruktura superkomputera przeprowadzona będzie analiza porównująca architektury sprzętowe różnych producentów pod kątem wydajności i energooszczędności. Dodatkowo badania przeprowadzane będą na superkomputerze Zeus, który cechuje się dużą liczbą jednostek obliczeniowych dostępnych z poziomu jednego węzła. W pracy chciałbym przeanalizować opłacalność użycia wielu kart graficznych oraz procesorów do złożonych obliczeń co przełoży się na większe zużycie prądu, lecz mniejszy czas wykonania zadania, i chciałbym w niniejszej pracy odpowiedzieć na pytania, czy czas który zyskamy przy użyciu wielu kart rekompensuje nam zużycie energii które te karty wnoszą, oraz w jakim momencie i przy ilu urządzeniach dokładanie kolejnych przestanie być opłacalne, a wręcz spowalniać będzie cały proces z powodu narzutu komunikacyjnego.

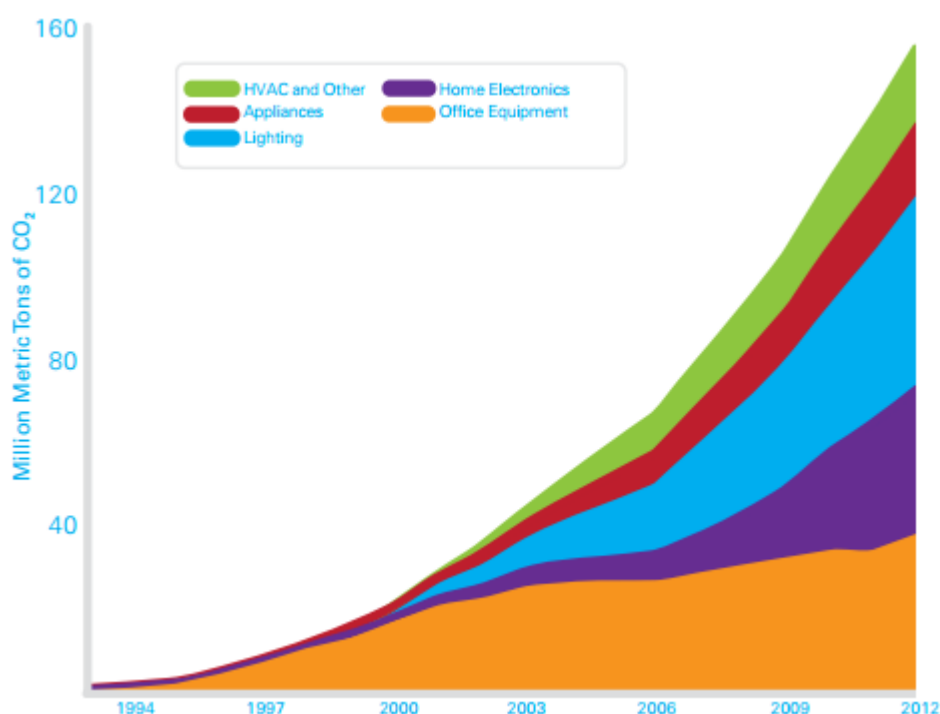
Kolejnym ważnym aspektem przeprowadzanych w niniejszej pracy badań oraz celem do którego będę dążył w trakcie jej trwania jest znalezienie i zaproponowanie zmian optymalizacyjnych, które spowodują zmniejszenie kosztu energetycznego obliczeń.

Dodatkowo ważnym dla mnie jest przeanalizowanie temperatur towarzyszących złożonym obliczeniom, ponieważ uważam, że znaczną częścią kosztów utrzymania sprzętu jest chłodzenie, które przy wyższych temperaturach jest wymagane, aby podzespoły pracowały poprawnie i utrzymywały jak najdłuższą żywotność.

Część I – Teoretyczna

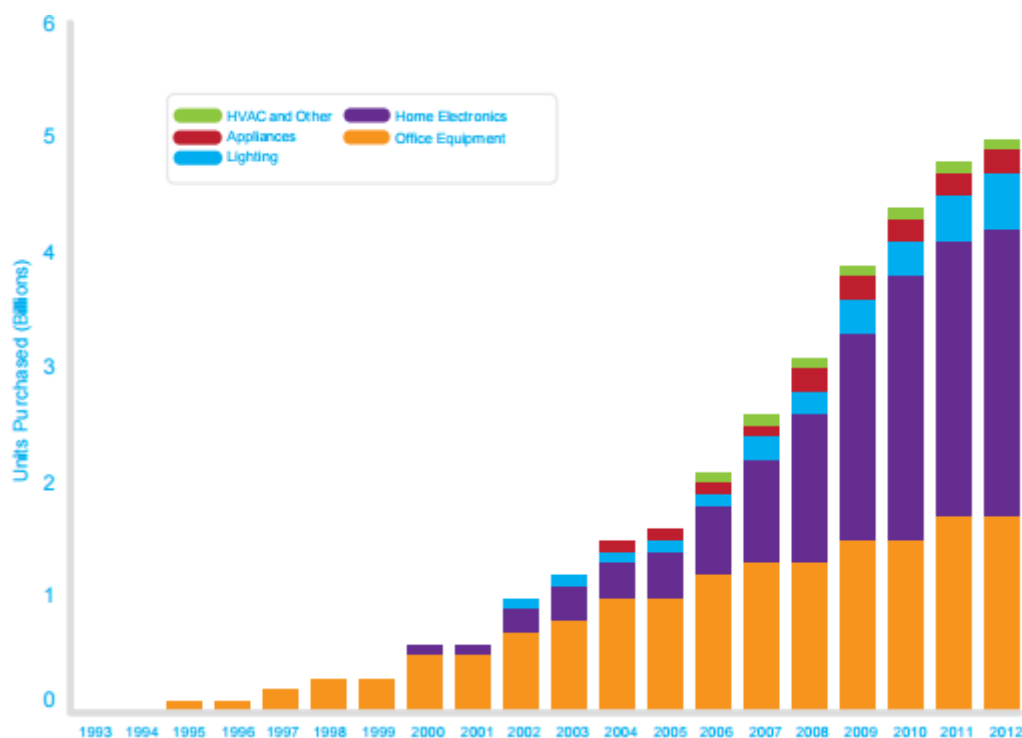
Green Computing

Poruszając temat Green Computingu będziemy mówić o wszelkich sposobach minimalizowania negatywnych skutków użytkowania komputerów oraz wszelkich zasobów z nimi związanych na środowisko naturalne. Głównymi celami do których dąży się w tym temacie są: obniżenie poboru energii elektrycznej stosując optymalizację na poziomie software'u i hardware'u oraz odpowiedzialne zarządzanie odpadami pochodzenia elektronicznego. Jednym z pierwszych działań podjętych na rzecz Green Computingu był powstały w 1992 roku program Energy Star, który promował i pomagał rozpoznawać produkty energooszczędne. Głównym celem programu była redukcja emisji gazów cieplarnianych, a pierwszymi produktami oznaczonymi etykietą Energy Star były komputery i monitory[2]. Jak bardzo program ten przyczynił się do poprawy jakości życia i oddziałał na nasz ekosystem pokazują dane zebrane z dwudziestu ostatnich lat.



Rysunek 1 Roczna ilość niewyemitowanych metrycznych ton gazów cieplarnianych.

Źródło: Energy Star Products Retrospective[3]



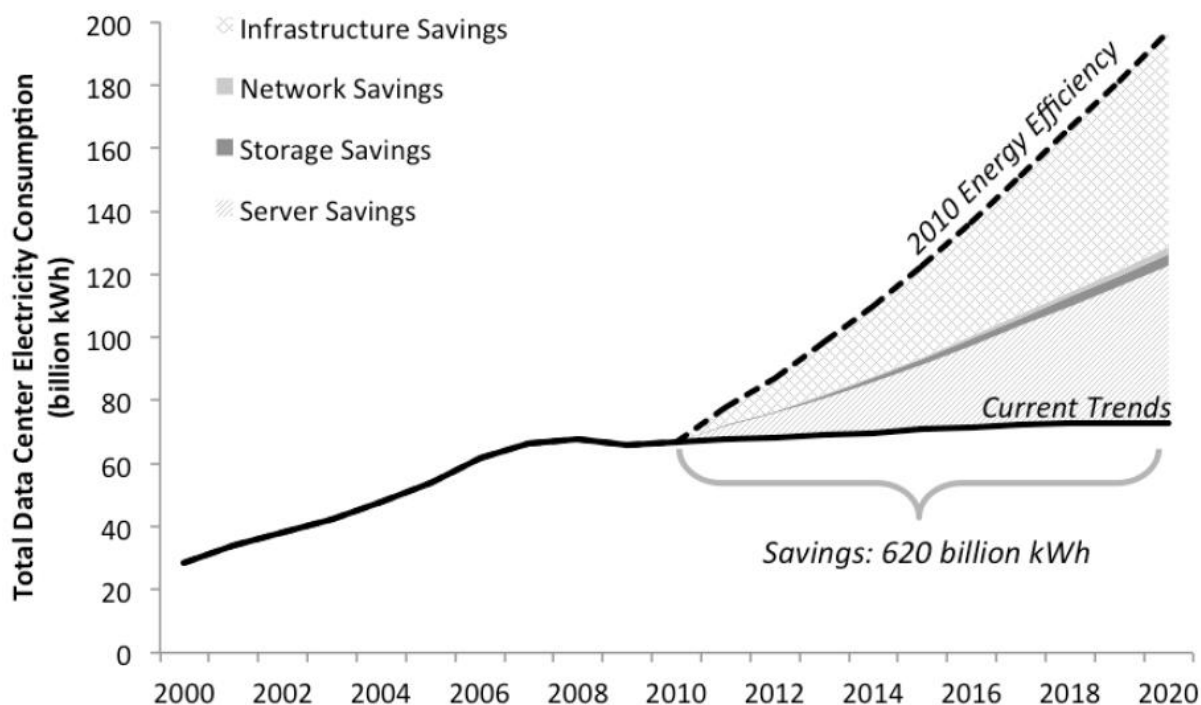
Rysunek 2 Ilość sprzętów z etykieta Energy Star zakupionych w latach 1993-2012 w USA łącznie.
Źródło: Energy Star Products Retrospective[4]

W samej tylko Ameryce rodziny oraz firmy zaoszczędziły około 230 miliardów dolarów na rachunkach, co więcej dzięki Energy Star nie dopuszczono do emisji ponad 1.7 miliarda metrycznych ton gazów cieplarnianych. Do 2012 roku sprzedano ponad 5 miliardów urządzeń z etykietą Energy Star, a w tym momencie zgodne z programem są kuchenki gazowe, żarówki, bojler, a nawet całe domy i setki innych urządzeń[5].

Zużycie energii jest jak widać sporym problemem zarówno dla środowiska naturalnego jak i finansowym dla każdego z nas, a problem ten minimalizowany jest na trzy sposoby. Za pomocą sterowników, gdzie na przykładzie kart graficznych można sterować wyłączeniem chłodzenia przy niewielkim ich użyciu, lub zmniejszeniu taktowania, albo nawet korzystaniu tylko ze zintegrowanej karty graficznej do momentu zajścia potrzeby włączenia dedykowanej. Sprzętowo, z roku na rok pojawiają się coraz nowsze technologie tworzenia urządzeń, a to, że problem jest już bardzo dobrze znany i ludzie są świadomi jego występowania to urządzenia tworzone są w taki sposób żeby jak najlepiej tą energią zarządzać, co również zwiększa żywotność samego sprzętu. Ostatnim sposobem jest odpowiedzialna i mądra obsługa sprzętu przez człowieka czyli dla przykładu decyzja o wyłączeniu lub hibernacji urządzenia gdy nie jest używane.

Spójrzmy jeszcze na centra danych, które jak oszacowano wykorzystują od 1% do 1.5% całej energii elektrycznej na świecie. Placówki te korzystają z energii ponad sto razy większej niż standardowe placówki biurowe [6]. W 2014 roku w samym USA centra danych

pochłonęły około 70 miliardów kWh co stanowiło 1.8 % zużycia energii na terenie całego państwa, a zgodnie z trendami z poprzednich lat wielkość zużycia będzie się zwiększać[7].



Rysunek 3 Zużycie energii elektrycznej w USA w latach 2000-2020
Źródło: US Department of Energy, Lawrence Berkeley National Laboratory [8]

Powyższy wykres pokazuje zużycie energii w samych centrach danych na przestrzeni 20 lat w USA, oraz to, jak szybko zużycie energii wzrastałoby gdyby w sprawach Green Computingu nie czyniono nic. W 2010 roku przewidywany początek kolejnego wzrostu zużycia energii, który przez 8 najbliższych lat wzrósłby dwukrotnie, lecz dzięki kilku ważnym czynnikom sprawiono, że wzrost został ustabilizowany. Przeważająca większość oszczędności na energii elektrycznej w tym przypadku wzięła się z tego, że ilość serwerów używanych przy centrach danych nie zwiększa się już tak bardzo jak było to jeszcze kilka lat wcześniej. Serwery posiadają teraz większą moc obliczeniową i są bardziej oszczędne w zużywanej energii, a dodatkowo w przemyśle zaczęto wykorzystywać wirtualizację serwerów co umożliwiło wykorzystanie jednej fizycznej jednostki do hostowania wielu. Należy dodatkowo pamiętać, że do każdego zakupionego serwera potrzebne były urządzenia zarządzające oraz chłodzenie co dodatkowo generowało koszty i zwiększało zużycie energii elektrycznej.

Między rokiem 2000 a 2005 firmy kupowały z roku na rok średnio 15% więcej serwerów niż w roku poprzednim, a całkowita liczba serwerów w ciągu tych pięciu lat się podwoiła. W tym momencie wzrost ten się ustabilizował, jeszcze do 2010 roku wzrost był w

wysokości 5% a dzięki m.in. wirtualizacji, która około 2008 roku zaczęła się pojawiać na rynku, spadł do 3% i badacze spodziewają się, że na tym poziomie wzrost utrzyma się do 2020 roku[9].

Charakterystyka użytych technologii

W roku 2009 trzy najwydajniejsze superkomputery na świecie – Jaguar, Roadrunner oraz Kraken XT5 nie posiadały kart graficznych. Co więcej biorąc pod uwagę dwadzieścia najszybszych superkomputerów w tamtym roku tylko jeden posiadał GPU. Przed 2010 rokiem obliczenia GPGPU (general-purpose GPU) czyli obliczenia bez użycia grafiki na kartach graficznych były uważane za nowość i praktycznie nie brano ich pod uwagę pod kątem zwiększania wydajności lecz na dzień dzisiejszy wszyscy są coraz bardziej zgodni, że połączenie CPU oraz GPU przy obliczeniach reprezentuje przyszłość obliczeń superkomputerowych.

Można się teraz zastanowić w jaki sposób możemy użyć takich urządzeń hybrydowych, jaki język programowania będzie mógł sobie poradzić z kilkoma różnymi platformami sprzętowymi i różnymi jednostkami obliczeniowymi. Stąd właśnie użyty przeze mnie w analizie aspektu oszczędności energii framework OpenCL który definiuje typy danych, struktury danych oraz funkcję rozszerzając niejako język C oraz C++. Procedury OpenCL mogą być uruchomione i wykonane na jednostkach procesorowych oraz graficznych nie przejmując się różnicami w architekturze oraz technologii różnych producentów. Programowanie równoległe pozwala wykonywać wiele zadań w tym samym czasie na wielu jednostkach obliczeniowych, a we frameworku OpenCl zadania te nazywane są Kernelami. Kernel to nic innego jak odpowiednio zaimplementowana funkcja, która może być uruchomiona na urządzeniach obsługujących framework OpenCl. Kernel taki jest wysyłany do urządzeń za pośrednictwem tak zwanego hosta, który jest zwykłą aplikacją napisaną w C/C++. Host zarządza połączonymi z nim urządzeniami za pomocą kontenerów nazywanych kontekstami, a aby w ogóle stworzyć kernel, host musi sięgnąć do funkcji nazywanej programem, do której przesyła się kod funkcji kernelowej a następnie przydziela się go do kolejki rozkazów (ang. command queue). Za pomocą kolejki tej, host może się komunikować z urządzeniami, a gdy kernel jest zakolejkowany, urządzenie uruchomi funkcję przekazaną w programie[10].

Jedną z głównych zalet używania OpenCl jest fakt, że możemy uruchamiać aplikacje za pomocą setek tysięcy wątków, nazywanych w środowisku OpenCl work-items, a górnym limitem ilości jest maksymalna wielkość typu size_t. Liczba takich work-items jest nazywana Global Size'em. Ważną rzeczą do zapamiętania jest to, że przy generowaniu takich work-item'ów, Global Size powinien być wielokrotnością work-group'y. Work group to nic innego jak zbiór work-item'ów, gdzie każdy z nich posiada swój własny identyfikator grupy. Każdy work-item posiada dwa identyfikatory, pierwszym jest Global ID, który identyfikuje i

odróżnia go od wszystkich innych work-item'ów, drugim jest Local ID, który identyfikuje go wewnątrz konkretnej work-group'y, więc w innych grupach może się on powtórzyć.

Każda work-group'a posiada swój własny blok pamięci nazywany pamięcią lokalną. Dla wielu urządzeń, work-item'y mogą uzyskać dostęp do danych z lokalnej pamięci o wiele szybciej, niż gdyby miały je pobierać z pamięci globalnej, więc aby jak najlepiej wykorzystać pamięć lokalną, jest ważne, by utworzyć jak największą liczbę work-item'ów w work-group'ie jak to tylko możliwe. Liczbę wszystkich work-item'ów w work-group'ie nazywamy Local Size'em, a OpenCL udostępnia możliwość sprawdzenia maksymalnej wartości Local Size dla konkretnego urządzenia[10].

Charakterystyka sprzętu

Badania będą przeprowadzone na superkomputerach Prometeusz oraz Zeus znajdujących się w krakowskim Cyfronecie należącym do Akademii Górniczo-Hutniczej. Poniżej w tabelach znajdują się ogólne specyfikacje superkomputerów, oraz bardziej szczegółowe informacje odnośnie jednostek obliczeniowych na których przeprowadzane były badania.

Tabela 1 Ogólna Specyfikacja superkomputera Prometeusz

Teoretyczna moc obliczeniowa	1.685 PetaFLOPS
Liczba rdzeni	41 472
Pamięć operacyjna	216 TB w technologii DDR4
Serwery	1728 serwerów HP Apollo 8000
Sieć	InfiniBand o przepustowości 56 Gbit/s

Źródło: opracowanie na podstawie danych z <http://www.agh.edu.pl/info/article/w-agh-uruchomiono-prometheusa-najpotężniejszy-superkomputer-w-historii-polski/> dostęp na dzień 18-10-2016

Tabela 2 Ogólna Specyfikacja superkomputera Zeus

Teoretyczna moc obliczeniowa	169 TFlops
Liczba rdzeni	25 468
Pamięć operacyjna	60 TB
Serwery	HP BL2x220c

Źródło: Opracowanie na podstawie danych z <http://www.cyfronet.krakow.pl/komputery/13345,artykul,zeus.html> oraz [https://pl.wikipedia.org/wiki/Zeus_\(superkomputer\)](https://pl.wikipedia.org/wiki/Zeus_(superkomputer)) dostęp na dzień 18-10-16

Tabela 3 Specyfikacja procesora superkomputera Prometheus

Producent	Intel
Model	Xeon E5-2680 v3
Architektura	Haswell
Liczba rdzeni	12
Nominalna częstotliwość	2500 MHz
Maksymalna częstotliwość (turbo)	3300 MHz
Nominalna częstotliwość AVX	2100 MHz
Maksymalna częstotliwość AVX (turbo)	3100 MHz
Moc obliczeniowa (nominalna) (DP)	480 Gflops
Typ pamięci operacyjnej	DDR4 2133 MHz
Przepustowość pamięci	68 GB/s
TDP	120 W

Źródło: <https://kdm.cyfronet.pl/portal/Prometheus:E5-2680v3> dostęp na dzień 18-10-16

Tabela 4 Specyfikacja kart graficznych superkomputera Prometeusz oraz Zeus

Model	Mikro architektura	Procesory skalarne		Pamięć			Całkowita moc obliczeniowa			TDP W
		Ilość	Zegar MHz	Przepustowość maks. GB/s	Szyna danych bit	Rozmiar GB	Zegar MHz	fp 32 GFLOPS	Fp 64 GFLOPS	
M2050	Fermi	448	1150	148	384	3	1500	1030	515	225
M2090		512	1300	177	384	6	1850	1331	665	225
K40	Kepler	2688	745 base 875 boost	288	384	12	3000	4290 base 5000 boost	1430 base 1660 boost	245

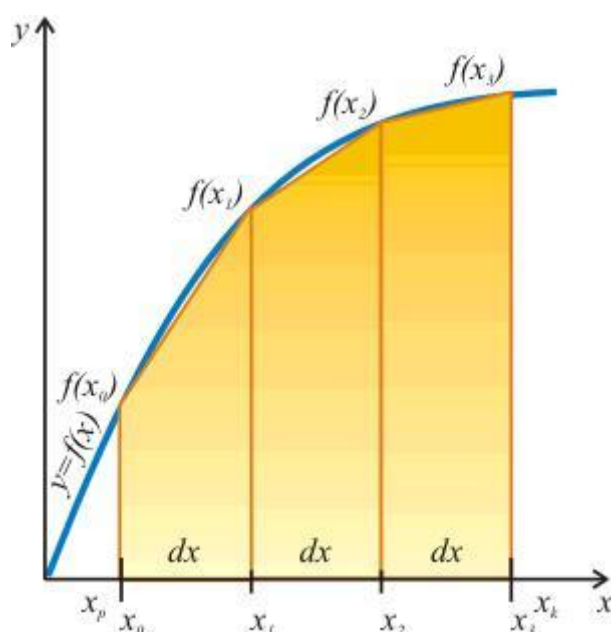
Źródło: Opracowanie na podstawie danych z https://pl.wikipedia.org/wiki/Nvidia_Tesla#cite_note-k40-10 dostęp na dzień 06-01-2017

Z powyższej tabeli karty o modelu M2050 oraz M2090 znajdują się w superkomputerze Zeus, natomiast karta K40 jest umieszczona w superkomputerze Prometeusz i na nich przeprowadzane były obliczenia w tej pracy. W jednym węźle superkomputera Zeus mamy dostęp do aż ośmiu kart graficznych, w obliczeniach korzystam jednak z kart M2090 ponieważ jest ona dostępna i wykrywana z poziomu OpenCl w przeciwieństwie do , natomiast Prometeusz posiada dwie karty z serii Nvidia K40XL oraz dwa procesory Xeon E5-2680 v3

Część II – Badania własne

Metody obliczeniowe, algorytmy

Pierwszym algorytmem który użyję przy badaniach w pracy jest całkowanie numeryczne metodą trapezową. Dla danej funkcji $f(x)$ i rozmiaru próbkowania dx będziemy liczyć długości boków trapezu równe wartości funkcji w punkcie x , wysokość trapezu będzie zawsze równa dx .



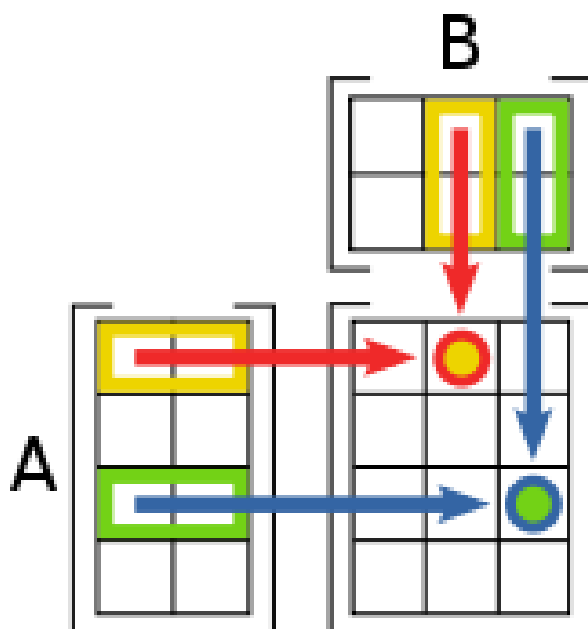
Rysunek 4 Całkowanie numeryczne: metoda trapezów

Źródło: <http://icetim.pl/publikacje/pid>

Algorytm ten wybrałem głównie ze względu na jego większą dokładność względem innych algorytmów całkowania oraz łatwości zrównoleglenia problemu. W moich badaniach, każde urządzenie obliczeniowe otrzyma w zestawie danych przedział, w którym będzie wykonywać obliczenia numeryczne, a wyniki będzie zapisywać do swojej pamięci globalnej urządzenia, skąd później w programie-goście zostanie sczytana i dodana wraz z innymi wartościami obliczonymi przez pozostałe urządzenia. Dodatkowo przekazana jest globalna tablica, która będzie służyła tutaj jako zabezpieczenie przed nadpisywaniem przez wątki danych, ponieważ każdy wątek będzie zapisywał do miejsca w tablicy, które będzie odpowiadało jego identyfikatorowi wątku, a na końcu programu kernela dodana jest bariera, która powoduje, że wątki muszą czekać w danym miejscu kodu do momentu, aż wszystkie wątki zakończą obliczenia. Następnie bariera zostaje zwalniana i tylko wątek o

identyfikatorze równym 0 będzie z tablicy przepisywał dane do bufora karty, który na końcu będzie wykorzystany do przekopiowania z niego danych do pamięci hosta. To podejście pozwala mi zabezpieczyć się przed ewentualnymi warunkami wyścigu wątków na jednostkach obliczeniowych, każde urządzenie ma swój przedział, a każdy wątek swoje unikalne ID, według którego oblicza wartości przynależne do siebie, bez możliwości wejścia w granicę obliczeń innych wątków.

Drugim algorytmem jest mnożenie macierzy przez macierz, wybór ten jest spowodowany ogromną ilością operacji, jaką urządzenie będzie musiało wykonać, aby zakończyć zadanie, a dodatkowo jest to problem 2D co pod kątem samego frameworku OpenCl oraz implementacji i zarządzania wątkami będzie się różnić od poprzedniego algorytmu.



Rysunek 5 Schemat mnożenia macierzy przez macierz

Źródło: <https://pl.wikipedia.org/>

W operacji tej ważne jest zachowanie odpowiednich wymiarów macierzy które będziemy ze sobą mnożyć, jak widać na rysunku 5 macierze mogą się różnić wymiarami ale zawsze liczba elementów w wierszu pierwszej macierzy musi być równa liczbie elementów w kolumnie drugiej macierzy. Przy obliczeniach wykonywanych na jednym urządzeniu problem wydaje się dość prosty do zaimplementowania oraz zrównoleglenia, ponieważ urządzenie dostaje jako dane dwie macierze, a każdemu wątkowi będzie przyporządkowany konkretny wiersz jednej i kolumna drugiej macierzy, macierze się nie zmieniają więc nie ma tu problemu z nadpisaniem danych przez inne wątki, a każdy wątek wykonuje kod programu

polegający na mnożeniu i dodaniu odpowiednich kolumn z wierszami macierzy i zapisania wyniku w dwóch lokalnych zmiennych (każdy wątek ma swoje własne zmienne), a następnie na podstawie swoich identyfikatorów wątków zapisują wyniki obliczeń do odpowiadających im miejsc w globalnej macierzy do pamięci bufora urządzenia, które na końcu po stronie hosta będzie przekopiwane. Algorytm ten cechuje się ogromną liczbą operacji na tych samych danych, przez co zastosowanie w frameworku OpenCL lokalnych grup roboczych z dzieloną pamięcią powinno dać o wiele krótsze czasy wykonania zadania, co warto będzie przetestować w dalszej części pracy.

Zrównoleglenie problemu mnożenia macierzy przez macierz rozwiązałem poprzez przesłanie całych dwóch macierzy do wszystkich dostępnych urządzeń obliczeniowych, a następnie tak zaimplementowałem kod po stronie kernela, że każde urządzenie korzysta tylko ze swojej części drugiej macierzy, i każde ma dokładnie taką samą część pod względem wielkości co inne urządzenia, także problem rozdzielony jest po równo.

Analiza uzyskanych wyników

Dokonanie pomiarów energii na kartach graficznych oraz procesorach możliwe było tylko i wyłącznie za pośrednictwem superkomputera Prometeusz. Dostępny był bowiem na nim narzędzie NVIDIA System Management Interface, który pozwala zarządzać i monitorować kartami graficznymi firmy Nvidia. Skupiłem się na zebraniu głównie dwóch najważniejszych dla mnie danych - mocy poszczególnych kart w trakcie rozpoczynania pracy, czyli jeszcze przed wysłaniem z hosta rozkazu dotyczącego uruchomienia kernela, i już w trakcie obliczeń zmiennoprzecinkowych. Druga dana dotyczy temperatury urządzeń biorących udział w obliczeniach, ponieważ jest to dość istotne do jakich temperatur podczas wzmożonej pracy potrafią dojść tak bardzo wydajne urządzenia, dodatkowo należy wziąć pod uwagę testy empiryczne i obliczenia matematyczne inżynierów z programu Supercomputing in Small Spaces, o których już wspominałem we wstępie pracy, a którzy zauważyli, że wzrost temperatury wpływa znacząco na awaryjność sprzętu. Dane zebrane za pomocą narzędzia NVIDIA System Management Interface są próbkowane co sekundę, co jest dość sporym przedziałem czasowym i niestety tutaj narzędzie to ogranicza zebrane wyniki i dalszą analizę. Pobór energii z procesora umożliwił skrypt sacct zawarty w systemie zarządzania klastrami Slurm. Zwraca on dla zakończonego zadania całkowitą energię zużytą przy pracy procesora.

W niniejszej pracy zabrakło pomiarów dotyczących kart oraz procesorów firmy AMD, ponieważ żaden z dostępnych klastrów nie posiadał urządzeń tej firmy, przez co niemożliwym było porównanie architektury różnych producentów sprzętu. Dodatkowo na superkomputerze Zeus nie udało się uzyskać pomiarów zużycia energii, gdyż nie ma tam narzędzia jak NVIDIA System Management Interface, i niestety nie ma możliwości zainstalowania go na tym klastrze, więc nie pozostało mi nic innego jak zmierzyć czasy wykonania zadań na poszczególnych ilościach kart graficznych i oszacowania opłacalności wykorzystania wielu kart przy obliczeniach zmiennoprzecinkowych. Zeus nie posiadał również zainstalowanego pakietu OpenCL SDK, przez co procesory na tym klastrze nie były dostępne dla obliczeń w OpenCL, dlatego też nie znalazła się w pracy ich specyfikacja. Obliczenia na Zeusie oraz ich charakter są głównie szacunkowe z uwagi na brak możliwości uzyskania zużycia, dlatego też ograniczyłem się do obliczeń dla jednego algorytmu na tym klastrze.

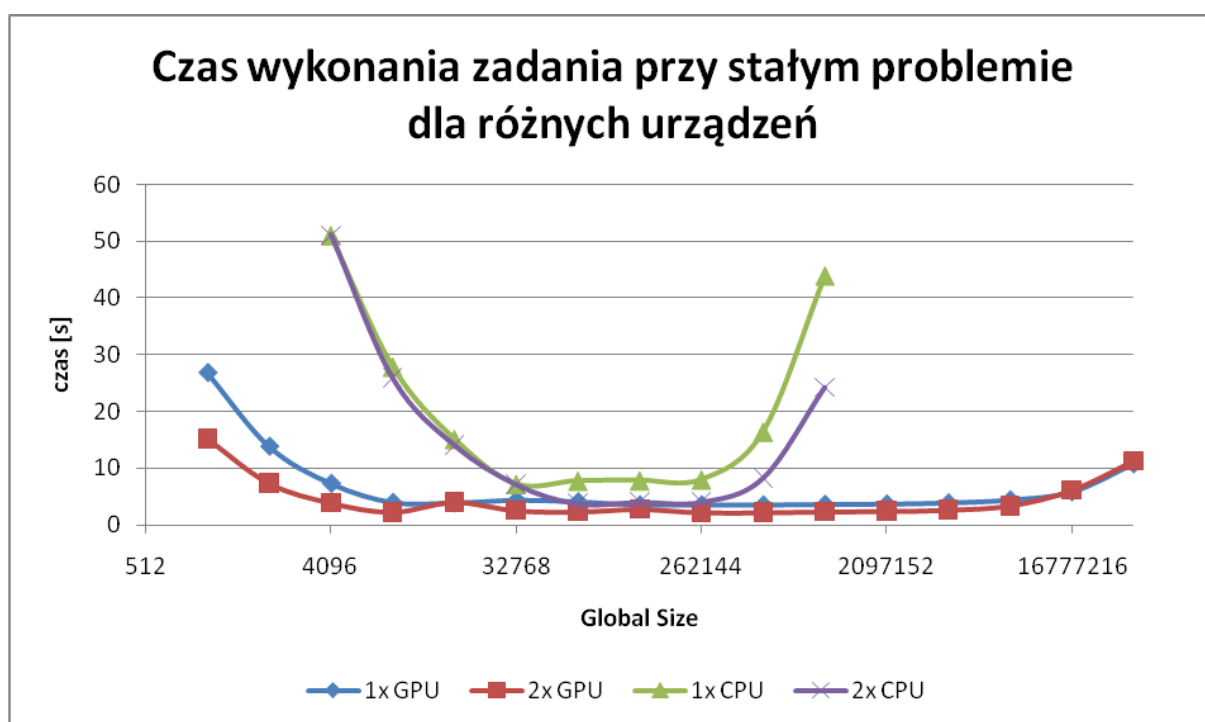
Analiza i badania przeprowadzone były w taki sposób, aby każde urządzenie na którym uruchamiano programy, było w miarę możliwości maksymalnie wykorzystane. Każda karta graficzna oraz procesor miała podczas obliczeń ustawioną maksymalną możliwą

wartość Local Size oraz uruchomioną maksymalną ilość rdzeni, oczywiście w zależności od wartości Global Size, która niejako determinuje, czy urządzenia są mocniej bądź słabiej eksploatowane.

Całkowanie numeryczne metodą trapezów

Założenia				
Przedział obliczeń	dx	Global Work Size	Local Work Size	Liczba rdzeni
0 - 65536	0.000001	zmienny	maksymalny	maksymalna

PROMETEUSZ - Wyniki – czas wykonania

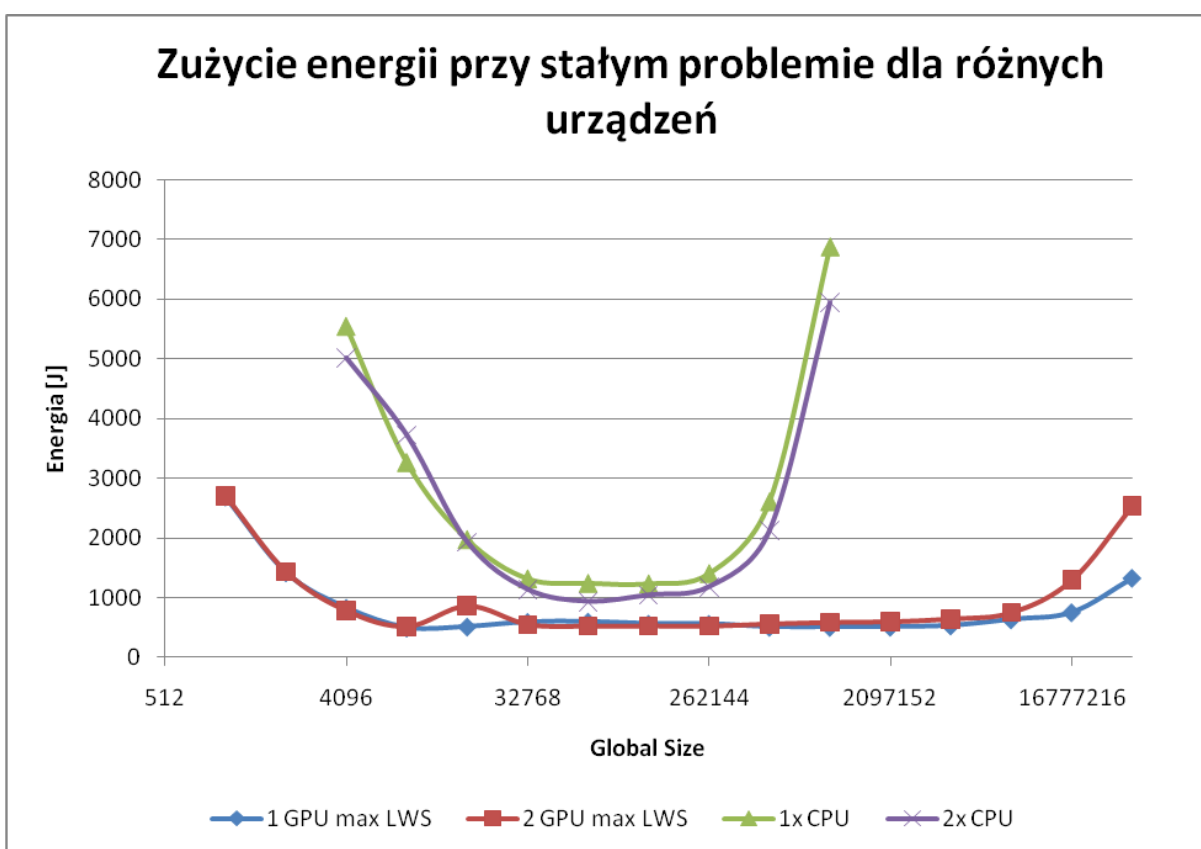


Źródło: Opracowanie własne

Powyższy wykres pokazuje wyniki dotyczące czasu wykonania przy stałym problemie. Pomiary zacząłem od wartości nieprzypadkowych, a związanych ściśle z rozmiarem Local oraz Global Size dla urządzeń. Karty graficzne dostępne na klastrze Prometheusz posiadały maksymalny Local Size równy 1024, procesory natomiast posiadały maksymalną wartość na poziomie 4096. Jak wiadomo rozmiar Global Size musi być podzielny bez reszty przez Local Size, czyli nie może być za razem mniejszy od niego, dlatego wyniki rozpoczynają się od różnych wartości, które są również minimalnymi wartościami możliwymi do ustawienia na konkretnych urządzeniach przy założeniu, że używam maksymalnej wartości Local Work Size.

Na pierwszy rzut oka patrząc na powyższe wyniki rzuca się fakt, że Global Size równy Local Size'owi nie jest optymalny niezależnie od urządzenia, głównie dlatego, że nie wykorzystujemy takiego urządzenia w pełni. Procesory na wykresie pokazują znacząco dłuższe czasy, gdy porównamy je z kartami graficznymi, co przekłada się też na większe zużycie energii. Widać też wyraźnie pewien zakres Global Size, który można nazywać optymalnym dla urządzeń na których pracowałem, ja obrałem za optymalną, wartość 32768 zarówno dla procesorów jak i kart graficznych, i to głównie pod kątem zużywanej energii, nie czasu wykonania.

PROMETEUSZ - Wyniki – zużycie energii



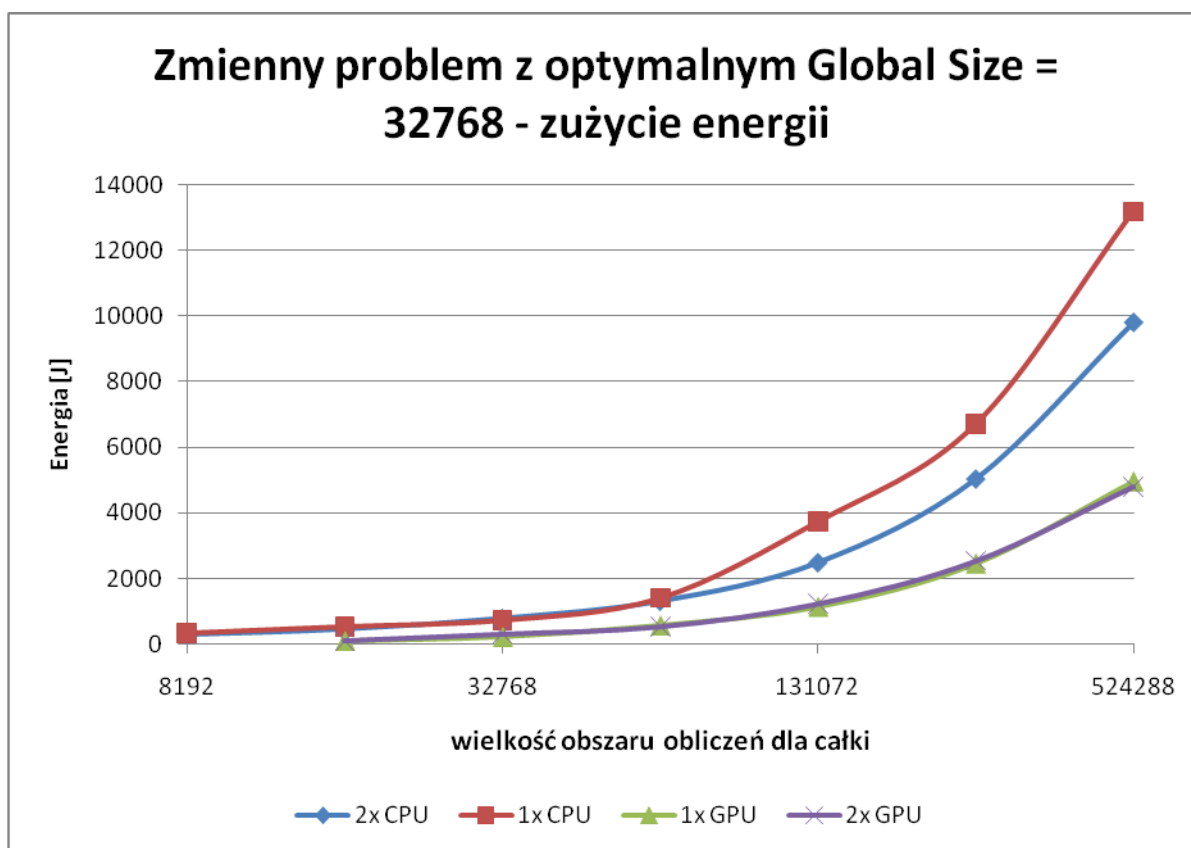
Źródło: Opracowanie własne

Jak słusznie zauważył Matthew Scarpino w swojej książce *OpenCL in action*, „dzisiejsze superkomputery pokazują, że żaden procesor nie posiada tak miażdżącej mocy obliczeniowej w przeliczeniu na Wat jak karty graficzne” i doskonale widać to na powyższym wykresie, ponieważ w żadnym jego punkcie procesory nie osiągają wyników lepszych niż dwukrotnie większe od kart graficznych. Karty natomiast przez większość obliczeń zachowują podobne zużycie energetyczne niezależnie czy do obliczeń była użyta jedna czy dwie. Dwukrotnie mniejszy czas dla dwóch kart jednocześnie wiąże się z dwukrotnie

większym zużyciem, pomijając bardzo duże wielkości Global Size, które nie są optymalne z powodu dużej ilości wątków, które czekają na zasoby obliczeniowe oraz sam fakt konieczności zarządzania nimi w dużej ilości.

Na szczególną uwagę według mnie zasługuje fakt, że wszystkie powyższe wykresy były tworzone w oparciu o jeden niezmienny rozmiar problemu, ten sam sprzęt, a tylko dzięki zmianie parametrów, wyszukaniu najbardziej optymalnej wartości jesteśmy w stanie dwukrotnie zmniejszyć zużycie energii na kartach graficznych, a dla procesorów nawet o ponad cztery razy.

PROMETEUSZ – Wyniki – optymalna wartość Global Size - zmienny problemu – zużycie energii

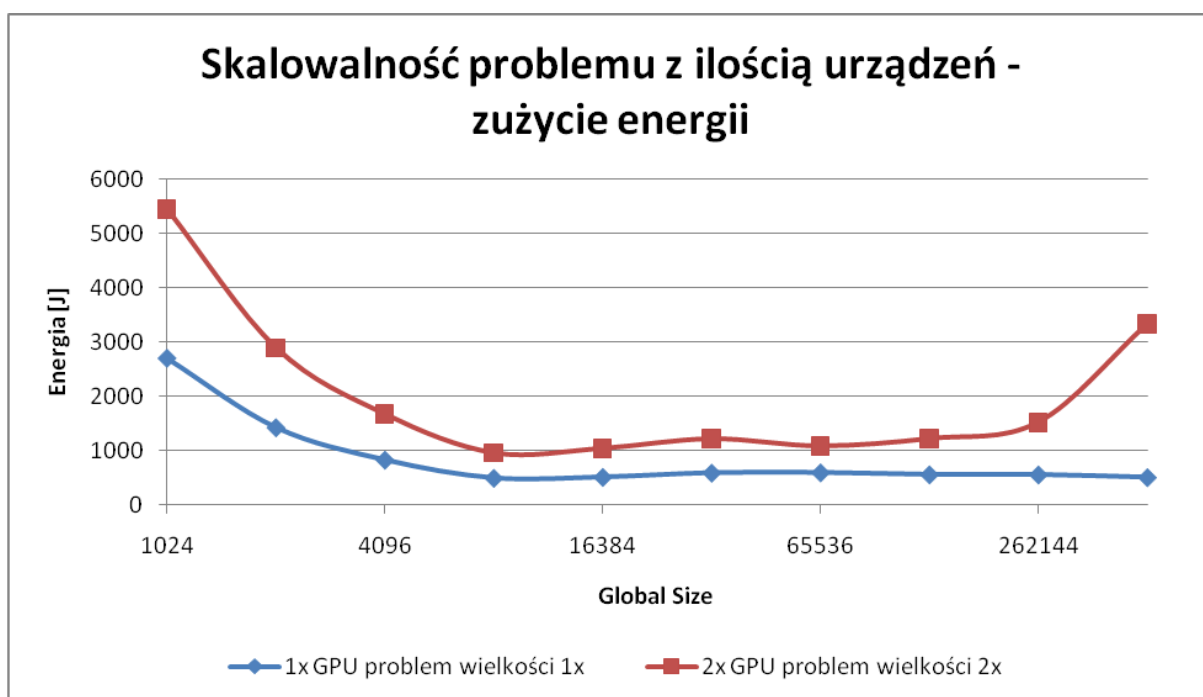


Źródło: Opracowanie własne

Powyższy wykres ukazuje nam, że odpowiednia parametryzacja może przynieść efekty w postaci dwóch procesorów pracujących z mniejszą kosztownością energetyczną niż jeden procesor, uważam, że jest tak ponieważ posiłkując się wykresem pierwszym dotyczącym czasów wykonania, parametr 32768 dla jednego procesora był najbardziej optymalnym, ze wszystkich sprawdzonych przeze mnie (wyższe wartości zaczęły go spowalniać), przez co wykonywał najwięcej pracy w dużo szybszym czasie niż dla innych

parametrów, lecz odbywało się to kosztem zużycia energii jaka musiała być włożona, aby procesor pracował na najwyższych obrotach. Dwa procesory natomiast powyżej tej wartości wybranej przeze mnie jako optymalna dalej zmniejszały czas wykonania zadania oraz zużywalność energii co wyraźnie widać na dwóch pierwszych wykresach. Oznacza to właśnie, że dwa procesory przy takim parametrze dalej nie pracują ze swym pełnym potencjałem, co również przy podziale problemu na dwa osobne procesory skutkuje całkowitym mniejszym kosztem energetycznym. Przy kartach graficznych nie możemy zobaczyć tak wyraźnych różnic, ponieważ przy tym parametrze karty pracują bardzo wyrównanie, i tak jak już zaznaczałem – mniejszy czas na dwóch kartach skutkuje większym zużyciem energii.

PROMETEUSZ – Wyniki – skalowalność problemu – zużycie energii

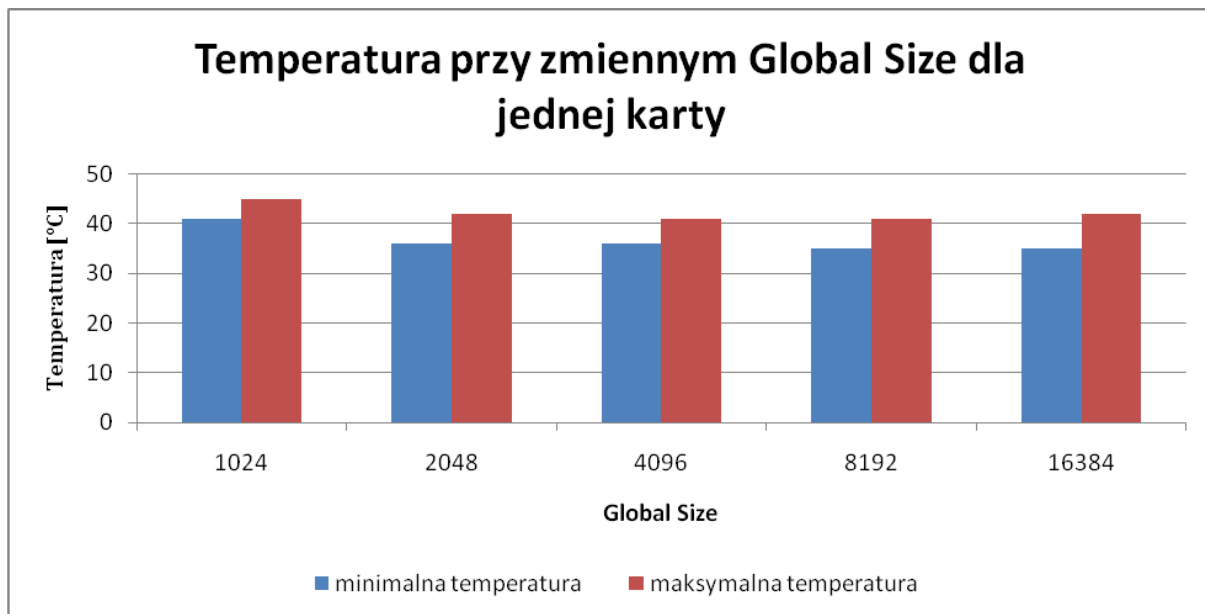


Źródło: Opracowanie własne

Badania, których wynikiem jest powyższy wykres przeprowadziłem głównie aby sprawdzić, czy przy tak równo pracujących energetycznie jak i czasowo kartach zauważymy jak duży jest narzut komunikacyjny kart między sobą oraz między programem-hostem. Spodziewałem się przy takim scenariuszu uzyskać wyniki niewiele większe niż dwukrotność wyników z jednej karty i w większości miałem rację. Gdy przeanalizujemy rezultaty dla dwóch kart, i przeskalujemy je do dwa razy mniejszego problemu, to dwie karty wychodzą odrobinę gorzej niż jedna, są to różnice setnych sekundy, a wraz ze zwiększającą się wartością Global Size jesteśmy w stanie zauważyć zwiększający się narzut zarządzania coraz

to większą ilością wątków, gdzie w ostatnim punkcie wykresu różnica jest sześciokrotna, a pod względem czasu niemal czterokrotna.

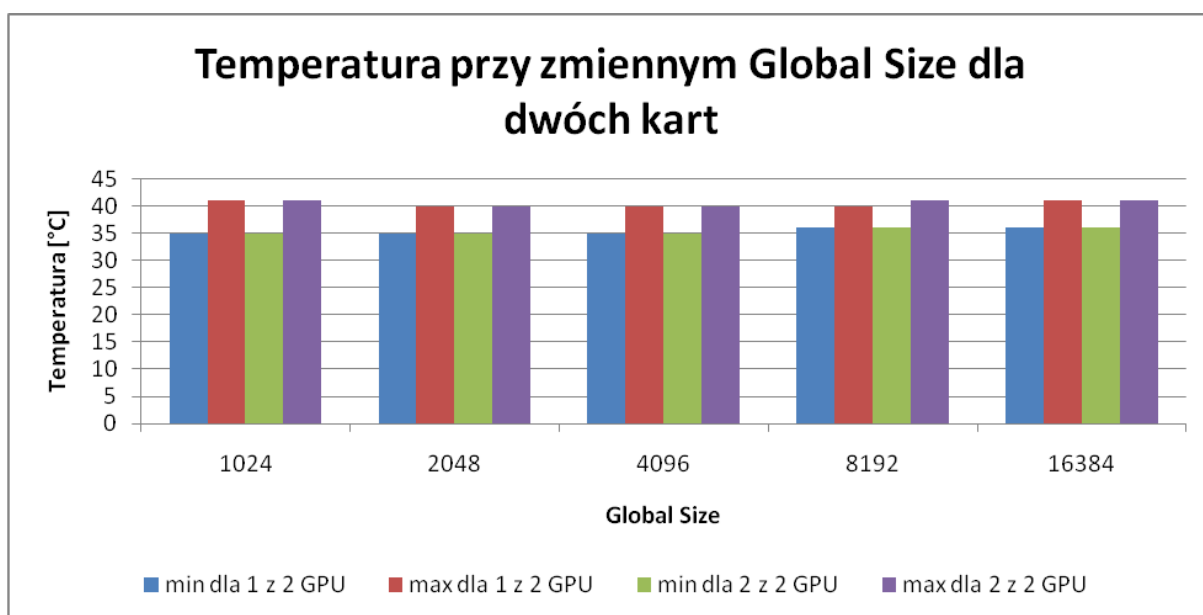
PROMETEUSZ – analiza temperatury – jedna karta



Źródło: Opracowanie własne

Algorytm całkowania mimo, że okazał się czasochłonny nie wpływał znacząco na wydzielanie się ciepła na karcie, nieco wyższe jednak wyniki można zaobserwować dla najniższej wartości Global Size, lecz może się to dotyczyć długości obliczeń, które trwały dwukrotnie dłużej niż przy Global Size 2048. Dopuszczam też możliwość, że karta na której dokonywałem obliczenia tuż przed była eksploatowana przez innego użytkownika klastra, przez co zaczęła pracę z wyższego pułapu temperatury.

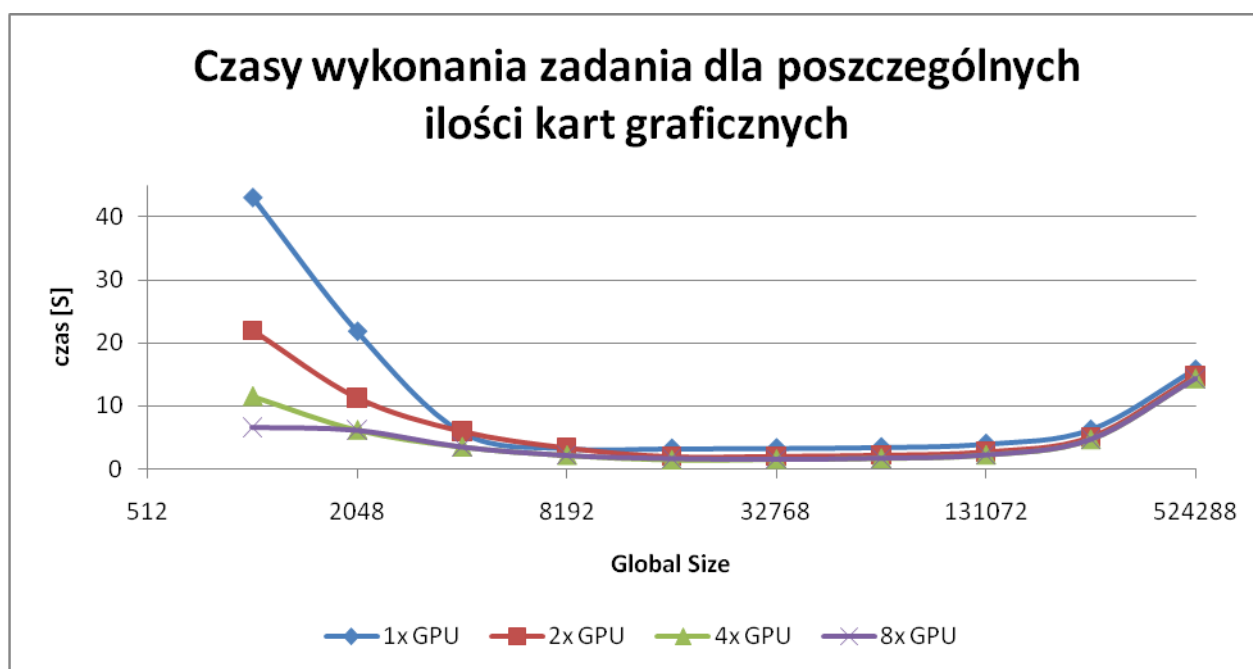
PROMETEUSZ – analiza temperatury – dwie karty



Źródło: Opracowanie własne

Powyższe wyniki potwierdzają, że minimalny Global Size nie wpływa na wzrosty temperatur. Różnice między minimalnymi i maksymalnymi wartościami są rzędu pięciu stopni Celsjusza, jest to znikomy przyrost i dopuszczalny przy tego typu obliczeniach. Długość zadania przy całej również nieznacząco wpływa na wzrost temperatury, dla przykładu zadanie które wykonywało się przy bardzo dużym Global Size sto trzydzieści sekund na jednej karcie posiadało wzrost temperatury rzędu ośmiu stopni Celsjusza, po zmniejszeniu Global Size problem wykonywał się osiem sekund, a wzrost ciepła był tego samego poziomu, podobnego zresztą do wyników z powyższego wykresu.

ZEUS – Wyniki – czas wykonania



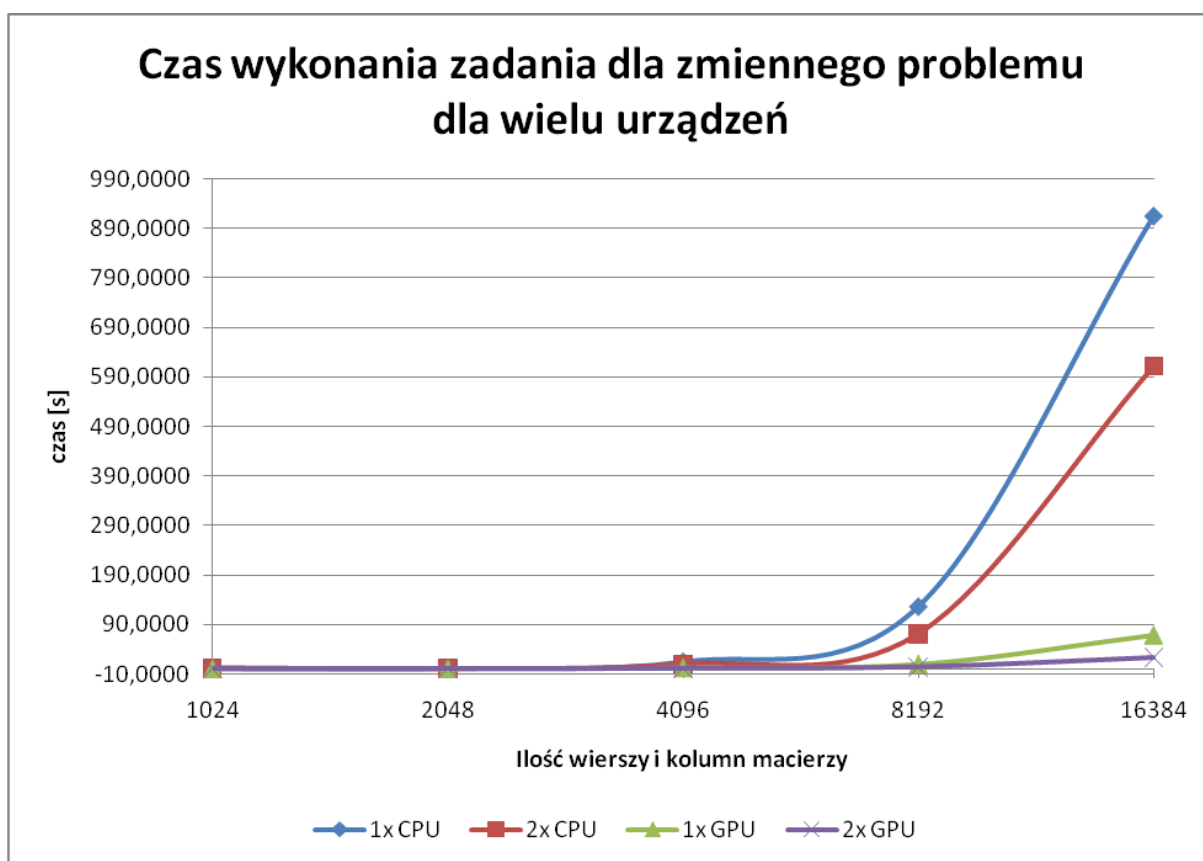
Źródło: Opracowanie własne

Dla klastra Zeus nie jesteśmy w stanie dokładnie zbadać zużycia energii, dlatego analiza ta ma charakter spekulacyjny. Gdyby zależało nam tylko na czasie wykonania zadania, to dokładając kart do stałego zadania zyskamy sensowne zauważalne przyspieszenie, lecz analizując wykres nigdy nie jest to przyspieszenie na tyle duże, aby móc mówić o jakiegokolwiek oszczędności energii. Przyspieszenie to poza tym widoczne jest tylko dla najniższego Global Size, który jak wcześniej wywnioskowałem nie może być parametrem optymalnym, bo z pewnością nie wykorzystujemy wtedy pełnego potencjału kart. Wyniki te jednak potwierdzają niejako konkluzję analizy na klastrze Prometeusz, ponieważ na innym sprzęcie widzimy tę samą zależność, jeden niezmienny rozmiar problemu, ten sam sprzęt przez całość trwania obliczeń, a zmiana parametrów potrafi znacząco wpłynąć na czas wykonania zadania, co przełoży się na niższą konsumpcję energii w czasie.

Algorytm mnożenia macierzy przez macierz

Założenia			
Rozmiar macierzy	Global Work Size	Local Work Size	Liczba rdzeni
zmienny	równy wielkości macierzy	maksymalny	maksymalna

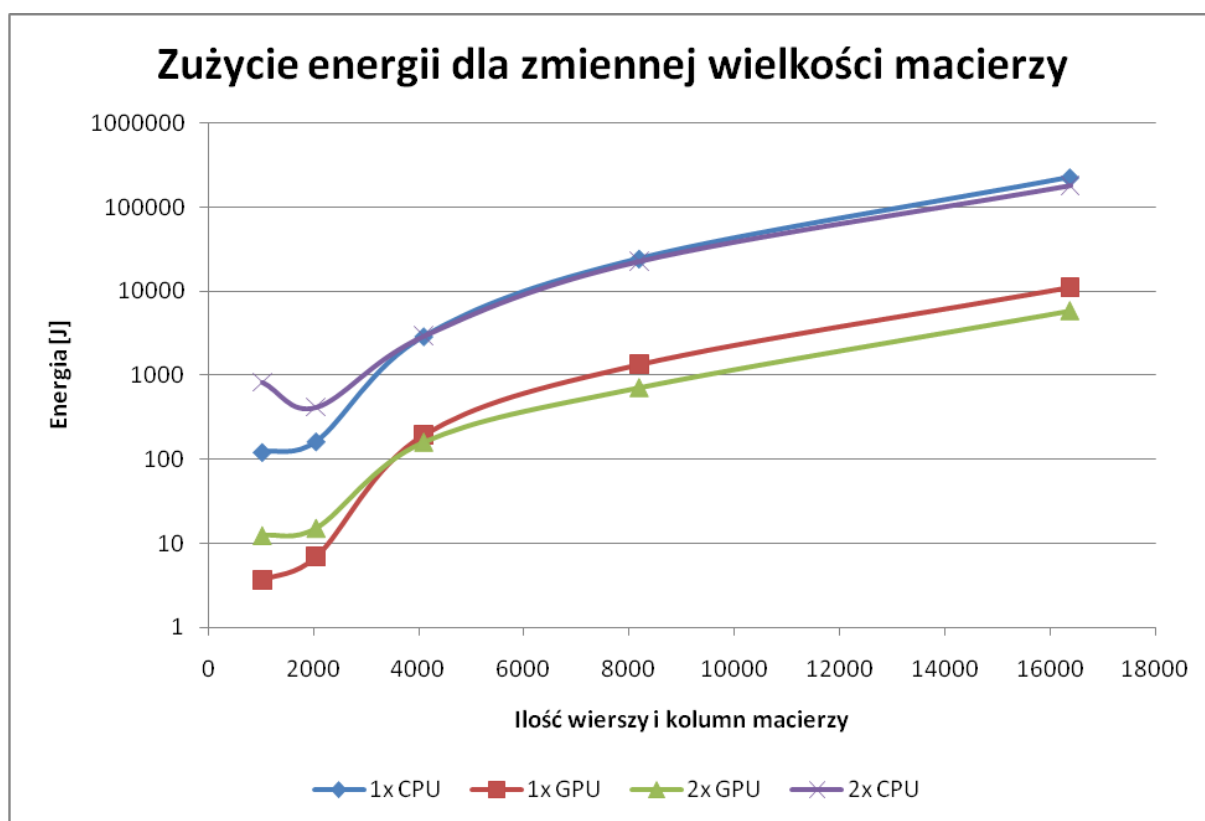
PROMETEUSZ - Wyniki – czas wykonania



Źródło: Opracowanie własne

Wyniki mnożenia macierzowego prezentują się dla małych rozmiarów macierzy bardzo podobnie, lecz osobiście zaskoczyło mnie to, że po raz pierwszy po tylu analizach procesor wykazał mniejszy czas obliczeń, niż karta graficzna, okazuje się, że dla bardzo małych rozmiarów danych jak macierze kwadratowe o rozmiarze 1024 oraz 2048 pod względem czasowym bardziej opłaca się używać procesora. W przypadku tego algorytmu każda zmiana problemu na większy jest o wiele bardziej zauważalna, tak jak przy małych czasach nie widać wielkich różnic, tak przy większych różnice są ogromne, w ostatnim punkcie wykresu, aż dziesięciokrotne.

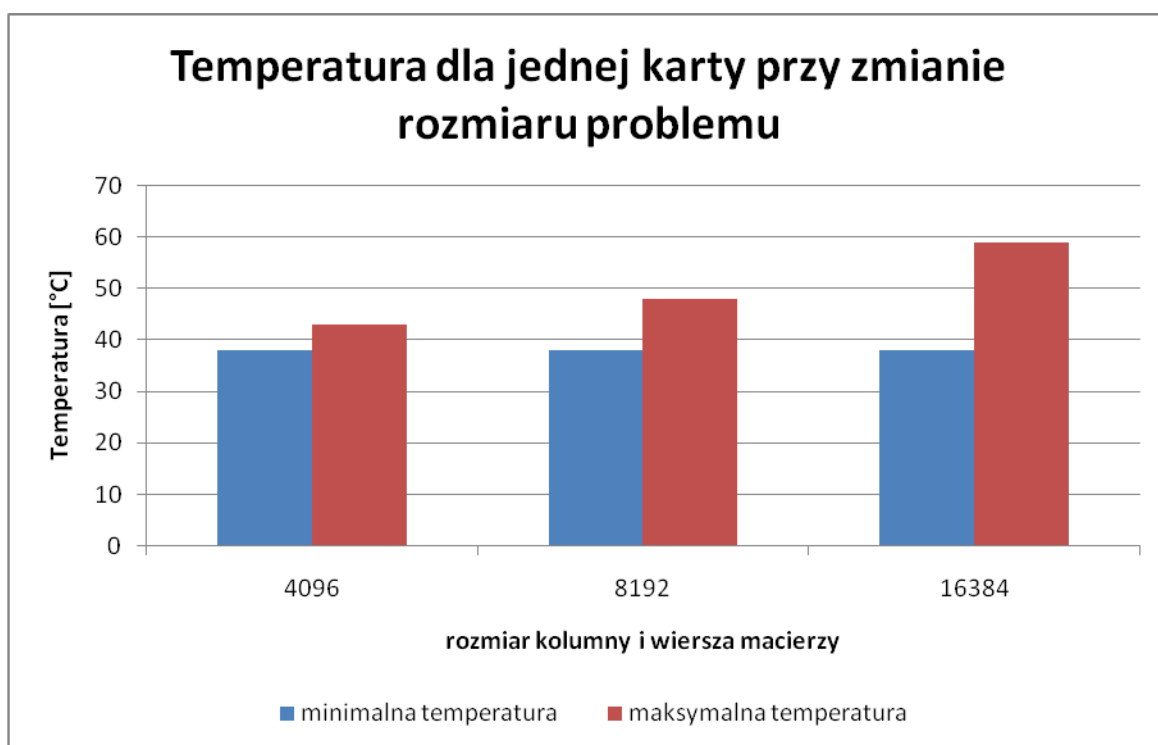
PROMETEUSZ - Wyniki – zużycie energii



Źródło: Opracowanie własne

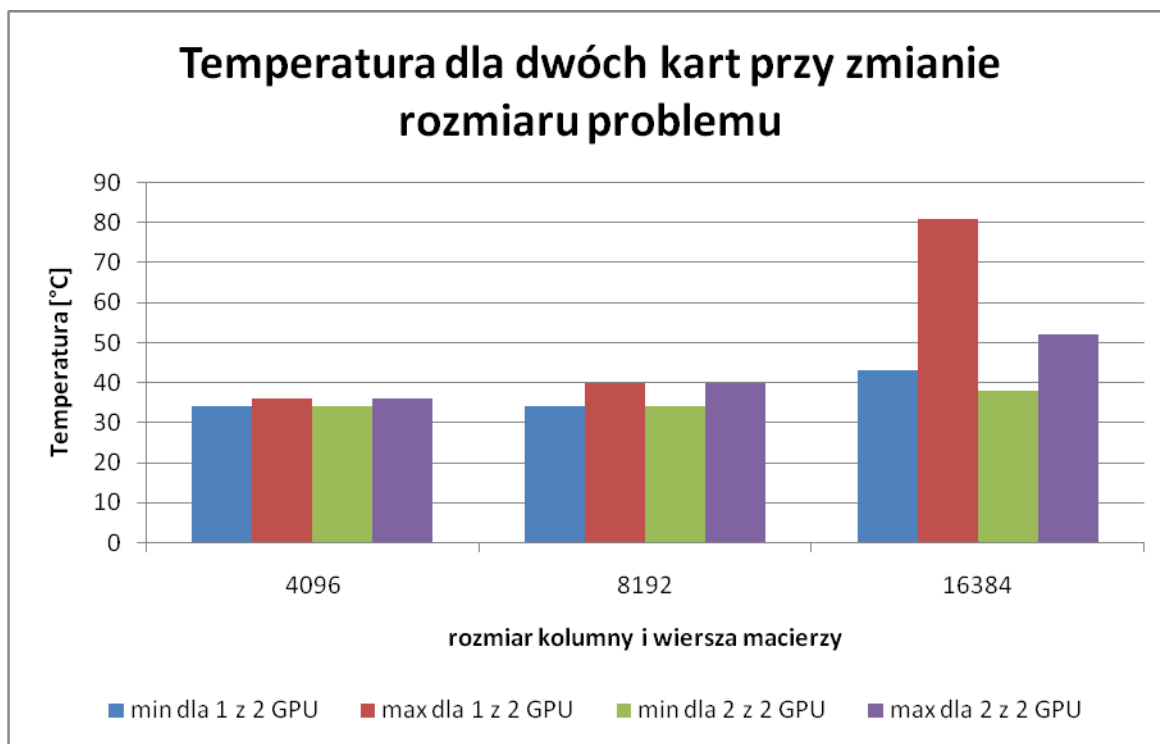
Bardzo niskie zużycie energii dla kart graficznych przy małym rozmiarze problemu wynika z braku dokładności narzędzia do pomiaru zużycia energii, który próbuje zużycie co sekundę, a dwa pierwsze kroki dla kart trwały mniej niż sekundę, dlatego narzędzie zarejestrowało zużycie spoczynkowe karty w wysokości 14 Wat i wyniki początkowe nie są miarodajne. Zważywszy na ilość wykonanych przeze mnie obliczeń wiem, że podczas obliczeń karty te średnio zużywają około 100 W, a przyjmując nawet maksymalną wartość jaką zarejestrowałem dla tego typu kart, która wynosiła 189 W to dalej było by to trzykrotnie mniejsze zużycie niż procesorów. Dalsze zwiększanie problemu skutkuje zauważalnymi różnicami w wydajności dla dwóch kart względem jednej, i widać korzyść z używania większej ilości kart, tak ze względu na mniejszy koszt energetyczny jak i krótszy czas wykonania zadania, i to krótszy trzykrotnie. Zakładam, że było tak za sprawą charakterystyki tego algorytmu, a dokładniej wielokrotnego użytkowania tych samych danych, a co za tym idzie tych samych obszarów pamięci.

PROMETEUSZ – analiza temperatury



Źródło: Opracowanie własne

Przy małej wielkości problemu, wzrost temperatury był znikomy, mniejszy niż przy całkowaniu, lecz przy zwiększeniu rozmiaru macierzy o cztery, ilość operacji i podział problemu musiały wpłynąć na wzrost temperatury. Różnica dziesięciu stopni robi wrażenie, zwłaszcza, gdy przypomnimy sobie wyniki inżynierów z projektu *Supercomputing in Small Spaces* o których wspominałem we wstępie tej pracy. Dziesięć stopni różnicy potrafi się przełożyć na dużo większą zawodność sprzętu, a gdy spojrzymy na kolejny rozmiar problemu, temperatura wzrosła o 20 stopni Celsjusza. Należy wziąć wtedy pod uwagę zużycie energii nie tylko na samym węźle obliczeniowym, lecz jednostkach chłodzących układy, co dodatkowo generuje koszty.



Źródło: Opracowanie własne

Ten sam rodzaj problemu, te same wielkości, dodatkowo problem podzielony po równo na dwa urządzenia. Na dwóch pierwszych seriach danych można zauważyć, że problem rozdzielił się po równo dla każdego z urządzeń, a nawet zmniejszyły się znacznie względem jednej karty, która osiągnęła dziesięć stopni przyrostu przy rozmiarze macierzy kwadratowych równych 8192 wiersze i kolumny. Spodziewałem się rozłożenia temperatury również przy najwyższym założonym przeze mnie rozmiarze macierzy, lecz tutaj temperatura sięgnęła aż osiemdziesięciu jeden stopni Celsjusza, powtórzyłem kilkukrotnie zadania z tym samym efektem, co potwierdziło, że nie był to przypadek bądź błąd. Niestety nie jestem w stanie przeanalizować tego przyrostu merytorycznie, uważam wręcz, że takie temperatury nie powinny występować na kartach, dlatego podejrzewam jakąś awarię chłodzenia lub konkretnej jednostki obliczeniowej.

Wnioski

Projekt inżynierski przedstawiony w tej pracy miał za zadanie opracować analizę aspektu oszczędności energii elektrycznej, pod kątem złożonych obliczeń numerycznych. Po przeprowadzonych testach wnioskuję, że złożone obliczenia powinny być przeprowadzane na kartach graficznych, ponieważ stosunek mocy obliczeniowej do zużytej na te obliczenia energii jest o wiele korzystniejszy niż w przypadku procesorów. Jednostka procesora jest zoptymalizowana pod sekwencyjne przetwarzanie szeregowo, i posiada stosunkowo małą liczbę rdzeni natomiast karty graficzne posiadają ich tysiące, o wiele wydajniejszych niż procesorowe, i stworzone z myślą o przetwarzaniu równoległe wielu zadań.

Uważam, że zadania bardziej złożone, mocniej obciążające obliczeniowo powinny zatem być uruchamiane na urządzeniu do tego przystosowanym, które daje na tym polu dobre efekty, natomiast na procesory powinny być przydzielane zadania mniejsze, sekwencyjne, nie dające się zrównoleglić, dzięki czemu najwydajniej wykonamy obliczenia tak czasowo jak i energetycznie. Przydział odpowiedniego zadania do odpowiedniego urządzenia jedną z propozycji które nasuwają się po przeprowadzonych badaniach, lecz same urządzenia muszą być też dobrze do tych zadań wykorzystane. Jak udało mi się zauważyć w projekcie zarówno procesory jak i karty graficzne wykonując złożone obliczenia, pomimo jednakowej implementacji algorytmów i stałego problemu potrafią zużywać różną ilość energii, dlatego należy znaleźć odpowiednie parametry, aby zużycie to zoptymalizować, i tego udało mi się podczas projektu inżynierskiego dokonać. Różnica w zużyciu energii dla najmniej optymalnego parametru Global Size, a wybranego przeze mnie była pięciokrotna, co pokazuje skalę oszczędności jakie można wprowadzić do niezoptymalizowanych systemów.

Projekt ten planuję w przyszłości dalej rozwijać ponieważ dotyka on tematyki ważnej, często ludziom znanej, lecz równie często ignorowanej. Widzę w tym spory potencjał do zmian, oraz szerokie perspektywy rozwoju, zwłaszcza że jeszcze tak wiele można w tym kierunku zrobić.

Bibliografia

- [1] Feng, W.-C. (2014). *The Green Computing Book - Tackling Energy at Large Scale*. Blacksburg: CRC Press.
- [2] Energystar: <https://www.energystar.gov/about/history> [online] – 18-10-2016
- [3] Energystar:
https://www.energystar.gov/ia/about/20_years/ES%20Anniv%20Book_030712%20for%20web.pdf?1094-68db – [online] 23-10-16
- [4] Energystar:
https://www.energystar.gov/ia/about/20_years/ES%20Anniv%20Book_030712%20for%20web.pdf?1094-68db
- [5] Energystar:
https://www.energystar.gov/ia/about/20_years/ES_20th_Anniv_brochure_spreads.pdf
-[online] – 23-10-16
- [6] Smith, B. E. (2014). *Green Computing - Tools and Techniques for Saving Energy, Money and Resources*. New York: CRC Press.
- [7] Datacenters:
<https://datacenters.lbl.gov/resources/united-states-data-center-energy-usage>
[online] – 23-10-16
- [8] Arman Shehabi,S.S. (2016). *United States Data Center Energy Usage Report*. California: Ernest Orlando Lawrence Berkley National Laboratory.
- [9] DataCenterKnowledge:
<http://www.datacenterknowledge.com/archives/2016/06/27/heres-how-much-energy-all-us-data-centers-consume/> - [online] 23-10-16
- [10] Scarpino, M. (2012). *OpenCL in action*. New York: Manning Publications C.o.

Spis tabel

Tabela 1 Ogólna Specyfikacja superkomputera Prometeusz	12
Tabela 2 Ogólna Specyfikacja superkomputera Zeus	12
Tabela 3 Specyfikacja procesora superkomputera Prometeusz.....	13
Tabela 4 Specyfikacja kart graficznych superkomputera Prometeusz oraz Zeus	14

Spis rysunków

Rysunek 1 Roczna ilość niewyemitowanych metrycznych ton gazów cieplarnianych.....	6
Rysunek 2 Ilość sprzętów z etykieta Energy Star zakupionych w latach 1993-2012 w USA łącznie.....	7
Rysunek 3 Zużycie energii elektrycznej w USA w latach 2000-2020	8
Rysunek 4 Całkowanie numeryczne: metoda trapezów	15
Rysunek 5 Schemat mnożenia macierzy przez macierz.....	16