Zbigniew Koza, Maciej Matyka, Sebastian Szkoda
Uniwersytet Wrocławski
Wydział Fizyki i Astronomii

Łukasz Mirosław
Politechnika Wrocławska, Instytut Informatyki
oraz Vratis Sp. z o. o.

Jakub Poła
Uniwersytet Śląski
Wydział Matematyki Fizyki i Chemii

Technologia GPGPU w obliczeniach naukowych

STRESZCZENIE

W artykule przedstawiamy jedną z najbardziej innowacyjnych technologii, która w najbliższych latach może zrewolucjonizować sposób przeprowadzania wysokowydajnych obliczeń naukowo-inżynierskich – wykonywanie obliczeń na masowo równoległych kartach graficznych (tzw. technologia GPGPU). Omawiamy bariery technologiczne, które od kilku lat ograniczają dalszy postęp w wydajności obliczeniowej komputerów. Przestawiamy najważniejsze cechy charakterystyczne architektury współczesnych kart graficznych pod kątem ich zastosowań numerycznych, analizujemy ich zalety i wady. Omawiamy też najważniejsze osiągnięcia wrocławskich fizyków w dziedzinie GPGPU.

WSTĘP

Jeżeli za Encyklopedią PWN przyjąć, że rewolucją jest „wszelka szybka i głęboka zmiana (np. rewolucja obyczajowa, przemysłowa, naukowa, techniczna)” [PWN], jesteśmy właśnie świadkami kolejnej rewolucji, która dokonuje się w przemyśle komputerowym. Jej specyfika polega na tym, że po kilku dekadach niezwykle szybkiego wzrostu mocy obliczeniowej komputerowych jednostek centralnych nadszedł w tej dziedzinie czas stagnacji. O czym bowiem, jeśli nie o stagnacji, świadczy to, że najnowszy procesor firmy Intel, Sandy Bridge, swoją wydajnością obliczeniową przewyższa starszy o półtora roku model Lynnfield o kilka, może kilkanaście procent, podczas gdy jeszcze dziesięć lat temu kolejne generacje procesorów oferowały przyspieszenia rzędu 50% w stosunku rocznym?

Trwające mniej więcej od 2005 r. spowolnienie rozwoju jednostek centralnych rodzi poważne konsekwencje dla dalszego rozwoju cywilizacyjnego, grozi bowiem zablokowaniem jednej z najbardziej owocnych dróg postępu: coraz bardziej wyrafinowanych symulacji komputerowych. Dzięki trwającemu kilka dziesięcioleci bezprecedensowemu rozwojowi technologii mikroprocesorowej oraz przełomowym odkryciom w dziedzinie algorytmiki i inżynierii programowania, modelowanie komputerowe stało się powszechnie stosowną techniką badawczą, której ranga zaczęła dorównywać dwóm tradycyjnym filarom metody naukowej: teorii i eksperymentowi. Dzieje się tak dlatego, że zaawansowane symulacje komputerowe pozwalają przyspieszyć tempo i obniżyć koszty badań w niemal każdej dziedzinie nauki i techniki. Na przykład możliwość wykonywania coraz bardziej zaawansowanych symulacji komputerowych dynamiki płynów (CFD, ang. c*omputer fluid dynamics*) prowadzi do stopniowej marginalizacji

niezwykle kosztownych w budowie i eksploatacji urządzeń – tuneli aerodynamicznych. Skalę tego zjawiska dobrze ilustruje to, że pierwszy prywatny załogowy statek kosmiczny (samolot suborbitalny), SpaceShipOne, został zaprojektowany całkowicie w oparciu o symulacje CFD [Scaled]. Zdolność do błyskawicznego wykonywania ogromnej liczby obliczeń w układach elektronicznych jest niezbędna do działania tomografii komputerowej, symulatorów lotu, symulatorów pola walki, bibliotek cyfrowych, wyszukiwarek internetowych czy smartfonów. Z kolei możliwość wykonywania obliczeń w specjalistycznych ośrodkach obliczeniowych udostępniających klastry złożone z setek lub nawet tysięcy komputerów jest niezbędna do prognozowania zmian klimatycznych Ziemi, analizy danych z eksperymentów fizycznych w Wielkim Zderzaczu Hadronów (LHC), poszukiwania nowych materiałów, czy wspomagania projektowania prototypowego reaktora termonuklearnego (projekt ITER). Bez szybkiego wzrostu rozwoju mocy obliczeniowej komputerów można spodziewać się wyhamowania postępu technologicznego, co przy stopniowym wyczerpywaniem zasobów naturalnych i wciąż rosnących aspiracjach społecznych daje raczej mroczną perspektywę przyszłości. W tym sensie obecna sytuacja w przemyśle komputerowym ma charakter „szybkiej i głębokiej zmiany" o trudnych do przewidzenia konsekwencjach.

Spowolnienie tempa rozwoju możliwości obliczeniowych układów elektronicznych nie jest oczywiście konsekwencją czyjejś świadomej polityki, lecz napotkaniem przez przemysł licznych, fundamentalnych barier technologicznych. Jedną z nich jest bariera termiczna: działanie procesorów wymaga energii, która całkowicie zamienia się w ciepło, tymczasem układy te nie mogą funkcjonować stabilnie w temperaturze wyższej niż ok. 100 °C. Bariera termiczna wiąże się z barierą energetyczną: od ok. 10 lat żaden mikroprocesor nie pobiera więcej niż ok. 130 W, co jest górną granicą dla układów chłodzonych powietrzem (najnowsze czterordzeniowe układy Itanium pobierają do 170 W, a dla procesorów kart graficznych graniczna wartość wynosi ok. 250 W) [Fuller, 2011]. To właśnie osiągnięcie bariery energetycznej stanowi główną przyczynę załamania się szybkiego wzrostu mocy obliczeniowej pojedynczych układów obliczeniowych. Warto więc zastanowić się nad przyczynami tego zjawiska i jego konsekwencjami.

Moc (P) wydzielana na jednostce powierzchni układu scalonego dość dobrze opisywana jest wzorem [Dennard, 1974]

$$P = NCfU^2$$

gdzie N oznacza liczbę tranzystorów na jednostkę powierzchni, C jest ich pojemnością, f jest częstotliwością zegara, a U to napięcie zasilające. Ponieważ pojemność elektryczna k-krotnie zmniejszonych tranzystorów maleje k-krotnie, a ich liczba na jednostkę powierzchni rośnie jak k^2, przez długie lata możliwe było utrzymanie stałej wartości P poprzez równoczesne k-krotne zmniejszenie rozmiaru liniowego tranzystorów, k-krotne zmniejszenie napięcia zasilającego i k-krotne zwiększenie częstotliwości zegara (k oznacza tu pewną liczbę rzeczywistą większą od 1). Głównym czynnikiem decydującym o przyspieszeniu układu było nie tyle zwiększenie liczby tranzystorów, ile zwiększenie częstotliwości zegara. Niestety, metoda ta wyczerpała swój potencjał ze względu na dolne ograniczenie wartości napięcia zasilającego. Dalsze zmniejszanie tego napięcia prowadzi bowiem do bardzo szybkiego wzrostu tzw. upływności tranzystorów, czyli pasożytniczego prądu, który pojawia się w stanie, w którym tranzystor nie powinien przepuszczać prądu. Upływność (I) rośnie wykładniczo z obniżaniem tzw. napięcia progowego bramki (U_{th}), które musi być mniejsze od napięcia zasilania, zgodnie wzorem [Fuller, 2011]

$$I = I_0 \exp(-qU_{th} / \alpha kT)$$

gdzie I_0 jest pewną stałą, q jest ładunkiem elementarnym, $\alpha \approx 1{,}3$, k oznacza stałą Boltzmanna, a T jest temperaturą bezwzględną. Powyższy wzór na upływność ma charakter fundamentalny i obowiązywać będzie przy wszelkich możliwych konstrukcjach opartych na potencjale progowym, a więc i w tak innowacyjnych rozwiązaniach, jak tranzystory nanorurkowe czy grafenowe. Jeśli za ładunek elementarny przyjąć elektron, to w temperaturze pokojowej zmniejszenie napięcia progowego o zaledwie 100 mV powoduje 10-krotny wzrost upływności układu! Upływność była zaniedbywana, póki proces technologiczny nie osiągnął granicy ok. 90 nm, więc komputery pobierały energię tylko podczas zmiany stanu tranzystorów. Obecnie energia potrzebna jest nawet wtedy, gdy tranzystor znajduje się w stanie bezczynnym.

Jakiego rzędu jest to energia? W przypadku procesora zbudowanego we współczesnej technologii 40 nm wykonanie jednej operacji „pomnóż i dodaj" w 64-bitowej arytmetyce zmiennopozycyjnej wymaga zużycia ok. 100 pJ [Fuller, 2011]. Najszybsze karty graficzne potrafią wykonać 0.5×10^{12} takich operacji na sekundę, co oznacza konieczność dostarczenia do układu ok. 50 W mocy. Dlaczego więc te urządzenia pobierają 250 W? Otóż transfer 64 bitów danych wymaga zużycia 12,8 pJ na każdy milimetr połączenia. Przesłanie 4 argumentów operacji „pomnóż i dodaj" na odległość zaledwie 1 mm wymaga wydatkowania ok. 50 pJ, czyli połowy energii potrzebnej na wykonanie samej operacji. Przesłanie takiej ilości danych na odległość rzędu centymetra to wydatek energetyczny przekraczający koszt samych obliczeń o rząd wielkości. Równie kosztowna jest wymiana danych do lub z pamięci zewnętrznej.

Warto przypomnieć, że przemysł komputerowy nie po raz pierwszy natknął się na barierę energetyczną i dotychczas zawsze skutkowało to całkowitą zmianą procesu technologicznego. Dominująca obecnie technologia budowy tranzystorów, CMOS, w połowie lat 80. ubiegłego wieku wyparła technologię NMOS, gdyż umożliwiała kilkudziesięciokrotną redukcję poboru energii. Z tego samego powodu kilkanaście lat wcześniej układy scalone zbudowane w technologii NMOS wyparły konstrukcje składające się z pojedynczych tranzystorów. Te z kolei wcześniej wyparły prądożerne lampy, a lampy – przełączniki mechaniczne. Obecna sytuacje jest szczególna w tym sensie, że po raz pierwszy nie widać na horyzoncie żadnej technologii, która mogłaby zastąpić dominującą technologię w czasie krótszym niż 10 lat.

Kolejną barierą ograniczającą postęp w zakresie mocy obliczeniowej komputerów jest bariera pamięci. O ile w latach 90. ubiegłego wieku prędkość procesorów rosła o ok. 50% w stosunku rocznym, prędkość pamięci rosła zaledwie o ok. 10%. Ponadto po zatrzymaniu wzrostu wydajności pojedynczych procesorów, od 2004 r. zaczęto budować układy złożone z kilku rdzeni obsługiwanych przez tę samą pamięć. Co więcej, od wielu lat każdy rdzeń może wykonywać kilka operacji arytmetycznych jednocześnie. W tych warunkach dysproporcja pomiędzy teoretyczną wydajnością procesorów i praktyczną przepustowością pamięci RAM stała się jednym z głównych czynników ograniczających wydajność obliczeniową współczesnych komputerów. Producenci mikroprocesorów usiłują minimalizować efekt bariery pamięci m.in. poprzez instalowanie modułów pamięci podręcznej bezpośrednio na układzie scalonym mikroprocesora oraz rozbudowywanie jego układów sterujących o funkcje umożliwiające przewidywanie zapotrzebowania programu na dane i pobieranie ich z wyprzedzeniem. W niektórych modelach procesorów, np. Itanium Montecito, na pamięć podręczną przypada ponad 90% tranzystorów!

Warto jeszcze wspomnieć o jeszcze jednej barierze – barierze prędkości światła. Współczesne komputery taktowane są zegarami o częstotliwości rzędu 3 GHz, czyli 3 miliardów cykli na sekundę. Prędkość światła wynosi 300 000 km/s, czyli 30 miliardów cm/s. Oznacza to, ze w jednym takcie zegara światło przemierza ok. 10 cm. Uwzględniając konieczność synchronizacji różnych podzespołów procesora oraz to, że większość połączeń elektrycznych między różnymi punktami nie może przebiegać po liniach prostych, dochodzimy do wniosku, że obecne konstrukcje osiągnęły swój maksymalny rozmiar ok. 1cm. Jasne się też staje, że pamięć RAM, odległa od procesora o kilka cm i posiadająca rozmiary liniowe przekraczające 10 cm nie może być taktowana równie

szybkim zegarem, jak procesor główny. Bariera prędkości światła powoduje też, że jeżeli prawdziwe są doniesienia prasowe o możliwości skonstruowania tranzystorów grafenowych taktowanych zegarem 1 THz, a więc 300 (!) razy szybszych od współczesnych tranzystorów wykonanych w technologii CMOS, to rozmiary liniowe procesora grafenowego taktowanego takim zegarem nie mogłyby przekraczać 0.05 mm, same zaś tranzystory musiałyby zostać umieszczone w strukturze trójwymiarowej (zamiast używanej obecnie struktury planarnej), co w połączeniu z wymogiem masowości produkcji wydaje się żądaniem z gatunku science fiction.

Wszystkie te bariery technologiczne wskazują na konieczność szukania zupełnie nowatorskich rozwiązań, zasadniczo różniących się od dominującej w ostatnich latach, lecz technologicznie przestarzałej architektury i386. Najwydajniejsze komputery przyszłości będą składać się z sieci niezależnych od siebie układów wykonujących *lokalnie* obliczenia z ogromną prędkością i komunikujących się z innymi ultraszybkimi układami za pomocą stosunkowo wolnych łącz. W szczególności wysoki koszt energetyczny przesyłu danych wymusi stosowanie architektur, w których dane będą przechowywane i przetwarzane lokalnie w niezależnych od siebie „centrach obliczeniowych”. W szczególności oznacza to koniec epoki architektur szeregowych i bezpowrotne wejście w erę obliczeń równoległych. To z kolei wymagać będzie ogromnej pracy w zakresie algorytmiki i inżynierii oprogramowania.

Istnieje wiele przykładów nowatorskich rozwiązań technologicznych w dziedzinie wysokowydajnych obliczeń naukowych. Należy do nich m.in. eksperymentalny projekt Intela, w ramach którego w jednym układzie scalonym umieszczono 80 rdzeni obliczeniowych [Intel]. Innym jest heterogeniczny procesor Cell znany m.in. z konsol do gier PlayStation 3, ale i pierwszego superkomputera na liście Top500, japońskiego „K Computer”, który w teście linpack osiągnął wydajność 8 petaflop/s (8×10^{15} operacji zmiennoprzecinkowych na sekundę) [Top500]. W ostatnich latach duże nadzieje wiąże się także z inną wysokowydajną alternatywą dla architektury i386 – kartami graficznymi (GPU, ang. *Graphics Processing Units*).

Dalszą część artykułu poświęcamy właśnie wykorzystaniu kart graficznych w obliczeniach naukowo-inżynierskich. Opisujemy najważniejsze elementy architektury tych urządzeń oraz kilka przykładów ich zastosowań zarówno na świecie, jak i w środowisku wrocławskim. Na koniec formułujemy też kilka wniosków co do przyszłości tej technologii.

ARCHITEKTURA GPU

Architekturę współczesnych kart graficznych przedstawimy na przykładzie karty GeForce GTX 580 firmy nVidia. Karta ta zawiera 16 tzw. multiprocesorów, z których każdy posiada 32 procesory skalarne zwane przez producenta ,,rdzeniami CUDA’’. Łącznie urządzenie to posiada więc imponującą liczbę 512 rdzeni obliczeniowych. Mimo że są to znacznie prostsze jednostki od rdzeni CPU, każdy z nich może wykonywać te same operacje arytmetyczne co CPU. W konsekwencji karta ta swoją teoretyczną mocą obliczeniową przewyższa najszybsze wielordzeniowe CPU o rząd wielkości. Ta potężna moc ma jednak swoją cenę. Każdy multiprocesor posiada zaledwie dwie jednostki sterujące, każda obsługująca 16 rdzeni (tzw. *half-warp*). Oznacza to, że w danej chwili grupy 16 rdzeni otrzymują do wykonania ten sam rozkaz. Co prawda multiprocesor może chwilowo wyłączyć część rdzeni, dzięki czemu możliwe jest wykonywanie różnych instrukcji przez rdzenie tego samego multiprocesora, jednak proces ten realizowany jest sekwencyjnie i prowadzi do znacznej utraty mocy obliczeniowej urządzenia. W sumie jeden rozkaz kierowany jest do 32 rdzeni multiprocesora, zwanych *warpem*, które są wykonywane w dwóch cyklach zegara. Taka organizacja multiprocesorów prowadzi do znacznego spadku zapotrzebowania procesora na instrukcje, co z kolei zwalnia znaczną część przepustowości pamięci karty na potrzeby transferu danych. Moc obliczeniową kart graficznych zwiększa dodatkowo to, że mają zaimplementowaną instrukcję FMA, która w jednym cyklu zegara mnoży dwa argumenty i dodaje wynik do trzeciego.

Posiadają też one możliwość przeprowadzania obliczeń zarówno w pojedynczej, jak i podwójnej precyzji (aczkolwiek w typowych kartach graficznych serii GeForce prędkość obliczeń w podwójnej precyzji sztucznie ograniczono, prawdopodobnie ze względów marketingowych).

W ciekawy i symptomatyczny sposób zorganizowano obsługę pamięci. Po pierwsze, każdy multiprocesor posiada pulę 2^{15} (tj. ok. 32 tysięcy) 32-bitowych rejestrów, które można łączyć w pary w celu obsługi liczb 64-bitowych. Dodatkowo każdy multiprocesor posiada 64 KB ultraszybkiej pamięci zajmowanej wspólnie przez bufor pamięci podręcznej poziomu L1 oraz tzw. pamięć współdzieloną, w której można przechowywać zmienne i tablice. Każdy multiprocesor, oprócz bufora pamięci podręcznej na instrukcje, posiada też liczący 8 KB szybki bufor pamięci stałej oraz 6 do 8 KB podręcznego bufora pamięci tekstur, który można wykorzystać do przyspieszenia dostępu do danych. Z punktu widzenia programisty, każdy multiprocesor podzielony jest na tzw. bloki, którym sprzętowo przydzielana jest odpowiednia pula rejestrów, bufora L1 i pamięci współdzielonej. W celu minimalizacji wpływu opóźnienia pamięci zewnętrznej, zwykle na każdym multiprocesorze uruchamianych jest wiele bloków, które dzielą się na wątki obsługiwane przez pojedyncze rdzenie, przy czym liczba aktywnych wątków może sięgać aż 24 576 (1536 na multiprocesor). Zarządzanie wątkami odbywa się sprzętowo, przy czym wątki należące do różnych bloków mogą się ze sobą komunikować tylko za pośrednictwem pamięci zewnętrznej. Wszystkie multiprocesory mają dostęp do bufora pamięci podręcznej o pojemności 768 KB. Multiprocesory mają też dostęp do pamięci globalnej karty. Pamięć ta jest wbudowana w kartę, dlatego jest dużo szybsza (około 5 razy) od pamięci używanych przez CPU, jednak jest jej dużo mniej – maksymalnie 6 GB dla karty Tesla C2070. Poza tym maksymalną prędkość transferu danych do lub z pamięci karty uzyskuje się w ściśle określonych warunkach, gdy wszystkie rdzenie warpa odwołują się do blisko położonych komórek pamięci głównej. Próba wczytywania przez różne rdzenie warpa danych z losowo wybranych położeń pamięci DRAM prowadzi do znacznej degradacji przepustowości pamięci. Łączność procesora karty graficznej z zarządzającym nim procesorem CPU lub innymi procesorami GPU kart podłączonych do tej samej płyty głównej wymaga pośrednictwa pamięci DRAM karty oraz złącza PCI-Express. Łączność procesora GPU z innymi komputerami lub kartami graficznymi wymaga pośrednictwa pamięci RAM karty, złącza PCI-E, pamięci RAM CPU i normalnej sieci komputerowej.

W powyższym opisie rozpoznać można główne elementy architektury superkomputerów przyszłości. Zasadnicze obliczenia wykonywane są w licznych, działających jednocześnie, stosunkowo niewielkich rdzeniach obliczeniowych. Pamięć ma wyraźną strukturę hierarchiczną, począwszy od ultraszybkich rejestrów i pamięci współdzielonej, poprzez nieco wolniejszą pamięć RAM karty, jeszcze wolniejsze łącze PCI-E, po pamięć RAM CPU i przyłączone doń wielokrotnie wolniejsze inne nośniki pamięci, np. dyski twarde i sieć, przy czym im wyżej w hierarchii znajduje się dany rodzaj pamięci, tym jest szybszy, ale i kosztowniejszy, a więc i dostępny w mniejszych ilościach.

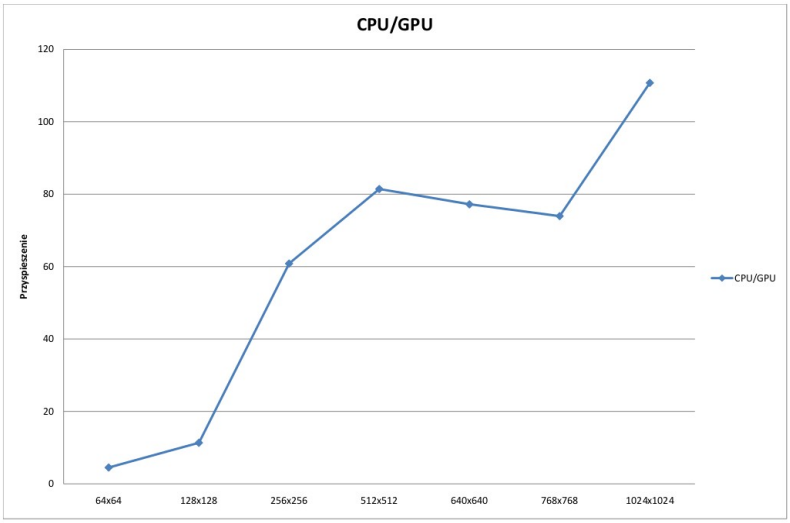
Jedną z główną przyczyn trudności w upowszechnieniu GPU jako koprocesorów matematycznych drugiej generacji jest ich niezgodność z dominującym paradygmatem procesora sekwencyjnego, a w szczególności z architekturą i386. Z tego powodu nie ma możliwości bezpośredniego przeniesienia istniejących programów na nową architekturę. Programy trzeba przygotowywać w zupełnie nowych językach, np. nVidia CUDA lub OpenCL, przy czym nader często przenosiny programów na masowo równoległą architekturę GPU wymagają opracowania nowych struktur danych i/lub nowych algorytmów, co wyklucza możliwość automatyzacji tego zadania. Oznacza to konieczność poświęcenia grubych tysięcy roboczogodzin bez jakiejkolwiek gwarancji, że istniejąca obecnie technologia GPU utrzyma się w przyszłości dostatecznie długo, by taki wysiłek miał ekonomiczne uzasadnienie.

ZASTOSOWANIA

Mimo opisanych powyżej trudności, wykorzystanie procesorów graficznych w obliczeniach naukowo-inżynierskich, zwane technologią GPGPU (ang. *General-Purpose Computing on Graphics Processing Units*), systematycznie zyskuje na znaczeniu. Dość powiedzieć, że wśród dziesięciu najszybszych obecnie (wrzesień 2011) superkomputerów na świecie, aż trzy swoją wysoką pozycję zawdzięczają kartom graficznym [Top500]. Na specjalnej stronie firmy nVidia znaleźć można setki przykładów prac naukowych, w których obliczenia na kartach graficznych z powodzeniem zastosowano w tak różnych dziedzinach nauki i techniku, jak CFD, analiza rynków finansowych, metody numeryczne, przemysł wydobywczy, dynamika molekularna, badania sejsmiczne, obrazowanie medyczne czy chemia obliczeniowa [CudaZone]. Osiągnięte przyspieszenia względem CPU wynoszą od kilku do nawet tysiąca (!) razy. Co prawda programiści Intela [Lee2010] szybko wytknęli autorom wielu z tych doniesień, iż porównywali oni wydajność implementacji na GPU do amatorskiej, jednordzeniowej implementacji na dość słabym CPU, tym niemniej można oczekiwać, że przyspieszenie obliczeń o rząd wielkości jest zupełnie realne.

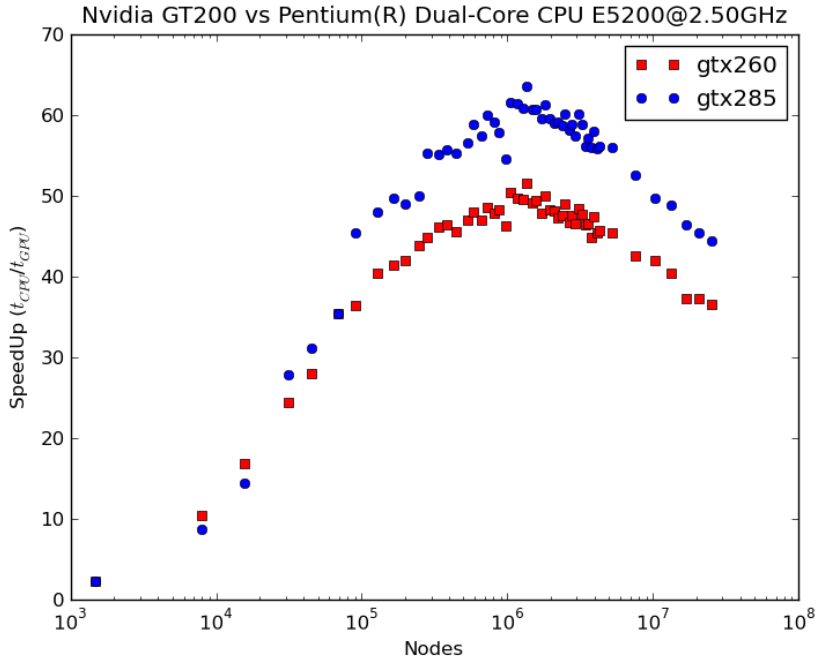
Na rynku pojawia się też coraz więcej zaawansowanych produktów wykorzystujących technologię GPGPU. Należą do nich m.in. Amber (zestaw narzędzi do modelowania molekularnego, stosowany głównie w przypadku makromolekuł biologicznych), Mathematica (komercyjny system obliczeń symbolicznych i numerycznych), Jacket for Matlab (komercyjne środowisko do wykonywania obliczeń naukowych i inżynierskich), NAMD (aplikacja służąca do symulacji układów biomolekularnych metodami dynamiki molekularnej), SpeedIT for OpenFOAM (wtyczka do popularnego środowiska obliczeniowej mechaniki płynów).

Od kilku lat technologia GPGPU wykorzystywana jest także w badaniach prowadzonych na Wydziale Fizyki i Astronomii Uniwersytetu Wrocławskiego. I tak, w badaniach nad możliwością przyspieszenia algorytmu LBM (ang. *lattice Boltzmann model*) rozwiązywania równań przepływu płynu, dla dostatecznie dużych siatek obliczeniowych osiągnięto ponad stukrotne przyspieszenie obliczeń (Rys. 1) [Poła, 2010].



Rys. 1. Przyspieszenie symulacji przepływu cieczy metodą LBM w zależności od rozmiaru siatki obliczeniowej uzyskane dzięki zastosowani technologii GPGPU.

Równie obiecujące wyniki uzyskano przy próbie zastosowania technologii GPGPU do przyspieszenia symulacji przepływu płynu w modelu gazu sieciowego FHP [Szkoda, 2010]. Przedstawione na rysunku 2 wyniki tych badań wskazują, że dla odpowiednio dużych układów karta graficzna jest w stanie wykonywać algorytm FHP ponad 60 razy szybciej od pojedynczego rdzenia CPU. Technologię GPGPU zastosowano też do symulacji gazu sieciowego metodą wymiany replik [Gołembiewski, 2011], uzyskując 35-krotne przyspieszenie obliczeń.



Rys. 2. Przyspieszenie symulacji przepływu cieczy metodą FHP w zależności od liczby węzłów w siatce obliczeniowej uzyskane dzięki zastosowani technologii GPGPU.

Z kolei w pracy [Malecha, 2011] zbadano możliwość przyspieszenia obliczeń biomedycznych. W tym celu, wykorzystując dostępność kodu źródłowego pakietu symulacyjnego OpenFoam, przeniesiono w nim na GPU czasochłonne podprogramy służące do iteracyjnego rozwiązywania układów równań liniowych i za pomocą tak zmodyfikowanego oprogramowania rozwiązano zagadnienie przepływu krwi w rozwidleniu aorty brzusznej. Symulacje przeprowadzono dla trzech różnych sytuacji fizycznych: (a) przepływ potencjalny (nielepki) i stacjonarny (brak pulsacji ciśnienia); (b) przepływ lepki i stacjonarny oraz (c) przepływ lepki i niestacjonarny. Ponieważ wydajność metod iteracyjnych zależy niemal wyłącznie od efektywności podprogramu do mnożenia macierzy rzadkiej przez wektor (spmv, ang. *sparse matrix-vector*), zbadano efektywność implementacji tej funkcji w technologii GPGPU na 8 macierzach testowych, otrzymując przyspieszenie od 3 do 14 razy względem procedur biblioteki Intel MKL uruchamianej na czterordzeniowym procesorze firmy Intel. Przeniesienie metod iteracyjnych biblioteki OpenFoam na GPU zaowocowało przyspieszeniem rzędu 4-6 razy dla przepływu potencjalnego i 3 do 4 razy dla lepkiego przypadku stacjonarnego rozwiązanego metodą SIMPLE (ang. *Semi-Implicit Method for Pressure-Linked Equations*). W przypadku przepływu niestacjonarnego technologia GPGPU nie dała żadnego przyspieszenia, gdyż przypadek ten wymaga częstej wymiany dużych ilości danych między CPU i GPU, co niweluje przyspieszenie samych obliczeń.

Oczywiście powyższe wyniki należy traktować z pewną dozą ostrożności, obciążone są bowiem wspomnianymi powyżej wadami metodologicznymi, w szczególności w trzech pierwszych przykładach wybrane do porównania implementacje na CPU wykorzystywały tylko jeden rdzeń

dość przeciętnego CPU, a w nim tylko jedną, główną jednostkę arytmetyczno-logiczną. Warto jednak zauważyć, że użyta do tych porównań implementacja GPU również nie była specjalnie optymalizowana (omawiane tu kody powstały w ramach prac magisterskich), a użyte procesory graficzne należały do „budżetowych" urządzeń średniej generacji – można jednak przypuszczać, że nawet najlepsza implementacja na CPU wciąż byłaby co najmniej o rząd wielkości wolniejsza od „studenckiej" implementacji na GPU. Z kolei w przypadku symulacji biomedycznych uzyskanie naprawdę dużego przyspieszenia wymagałoby przeniesienia na GPU całości obliczeń, co – jeśli w ogóle jest możliwe i ekonomicznie uzasadnione – wymagałoby wielu lat pracy. Z tego punktu widzenia uzyskanie przyspieszenia rzędu 5 razy tylko poprzez reimplementację stosunkowo niewielkiego fragmentu kodu CFD wydaje się bardzo obiecującym osiągnięciem.

WNIOSKI I PERSPEKTYWY

Prowadzone od ok. 2 lat na Wydziale Fizyki i Astronomii Uniwersytetu Wrocławskiego prace poświęcone badaniu możliwości wykorzystania ogromnej mocy obliczeniowej współczesnych kart graficznych do symulacji naukowo-inżynierskich nie prowadzą do jednoznacznych wniosków. Z jednej strony technologia ta umożliwia szybkie osiągnięcie znaczących przyspieszeń obliczeń numerycznych, z drugiej jednak strony mamy świadomość, popartą doświadczeniem, że nie jest i nie może być to technologia uniwersalna, gdyż nie wszystkie problemy można efektywnie rozwiązać w architekturze masowo równoległej.

Również perspektywy tej technologii nie są jasne. Niezwykle ważne jest to, że posiada ona wsparcie jednego z największych gigantów w branży informatycznej, firmy nVidia, która od 6 lat w regularnych odstępach 2-3 letnich wprowadza na rynek nową architekturę swoich kart graficznych o parametrach znacznie przewyższających poprzednie generacje tego sprzętu. W pewnym sensie przypomina to sytuację z lat 90. XX w., gdy zakupione zaledwie 3 lata wcześniej komputery były egzemplarzami niemal muzealnymi. Zasadnicza różnica polega na tym, że tym razem co dwa-trzy lata mamy do czynienia z istotnym postępem technologicznym o niemal rewolucyjnym charakterze, co musi znajdować swoje odzwierciedlenie w sposobie pisania programów. Zwiększa to koszty tworzenia oprogramowania i w pewnym sensie nadaje technologii GPGPU permanentny status technologii eksperymentalnej. Częste modyfikacje architektury powodują, że pomimo deklaracji firmy nVidia o utrzymaniu wstecznej zgodność kolejnych wersji swojego środowiska programistycznego CUDA, nie jest jasne, czy za kilka lat napisane dziś programy wciąż będą programami napisanymi optymalnie, zwłaszcza wobec niedawnych zapowiedzi firmy o połączeniu w jednym układzie kart graficznych z procesorem ARM (tzw. projekt Denver). Mamy więc wiele znaków zapytania, jednak z całą pewnością możemy powiedzieć, że przyszłość wysokowydajnych obliczeń naukowo-inżynierskich należy do nowatorskich rozwiązań opartych na podobnej do współczesnych GPU architekturze masowo-równoległej. Poświęcając energię eksperymentalnej technologii GPGPU ryzykuje się stratę czasu i środków, jednak niepodjęcie tego wyzwania oznaczałoby pewność oddania gry walkowerem.

PODZIĘKOWANIA

Niniejsza praca powstała częściowo dzięki wsparciu Ministerstwa Nauki i Szkolnictwa Wyższego projektem badawczym Nr N N519 (Z.K., M.M., S.Sz.).

BIBLIOGRAFIA

[CudaZone] http://developer.nvidia.com/category/zone/cuda-zone

[Dennard, 1974] Robert H. Dennard, Fritz H. Gaensslen, Hwa-Nien. Yu, V. Leo Rideout, Ernest Bassous, and Andre R. LeBlanc, *Design of ion-implanted MOSFETS with very small physical dimensions*, IEEE Journal of Solid State Circuits 9(5), 256–268 (1974).

[Fuller, 2011] Samuel H. Fuller and Lynette I. Millett (Eds) *The Future of Computing Performance: Game Over or Next Level?*, The National Academies Press, Washington D.C., 2011.

[Gołembiewski, 2011] Jarosław Gołembiewski, *Symulacje gazu sieciowego z oddziaływaniem trójciałowym metodą wymiany replik. Implementacja w technologii CUDA.* praca magisterska, Uniwersytet Wrocławski (2011).

[Intel] http://techresearch.intel.com/ProjectDetails.aspx?Id=151

[Lee, 2010] V. W. Lee at al., *Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU*, ISCA '10 Proceedings of the 37th annual international symposium on Computer architecture, ACM, New York, 2010.

[Malecha, 2011] Z. Malecha, L. Miroslaw, T. Tomczak, Z. Koza, M. Matyka, W. Tarnawski, D. Szczerba, *GPU-based simulation of 3D blood flow in abdominal aorta using OpenFOAM* Arch. Mech. 63(2), 137-161 (2011).

[Poła, 2010] Jakub Poła, *Symulacja przepływu cieczy metodą gazu sieciowego Boltzmanna przy użyciu technologii nVidia CUDA*, praca magisterska, Uniwersytet Wrocławski (2010).

[PWN] Encyklopedia PWN, http://encyklopedia.pwn.pl/haslo.php?id=3967471.

[Scaled] http://www.scaled.com/projects/tierone/faq

[Szkoda, 2010] Sebastian Szkoda, *Implementacja modelu FHP w technologii NVIDIA CUDA*, praca magisterska, Uniwersytet Wrocławski (2010).
http://www.ift.uni.wroc.pl/~sebastian.szkoda/msc.html

[Top500] http://www.top500.org/lists/2011/06/press-release

SUMMARY

APPLICATION OF GPGPU TECHNOLOGY IN COMPUTATIONAL SCIENCE

One of the most innovative technologies is presented that can revolutionize high-performace computing – general-purpose computations on graphics processing units (GPGPU). Technological barriers that prevent a further rapid progress in computer performance are discussed. The most important architectural features of modern graphics cards are reviewed from the high-performance computing viewpoint as well as their advantages and disadvantages over the traditional design. Recent achievements of Wroclaw physicists in this field are also discussed.