ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ

Seria: INFORMATYKA z. 24

Nr kol. 1222

Krzysztof BORYCZKO, Marian BUBAK, Jacek KITOWSKI, Jacek MOŚCIŃSKI, Renata SLOTA

ROZPROSZONE OBLICZENIA NA SIECI STACJI ROBOCZYCH I KOMPUTERZE CONVEX C3210

<u>Streszczenie.</u> W artykule omówiono podejście do obliczeń równoległych dużej skali, obejmujących symulacje metodami automatów komórkowych (gazu siatkowego) i dynamiki molekularnej, realizowanych w sposób rozproszony na sieci komputerowej złożonej ze stacji roboczych i komputera CONVEX C3210. Przedstawiono także ocenę przydatności sieci do obliczeń rozproszonych na podstawie benchmarków SLALOM i NAS EP.

DISTRIBUTED COMPUTING ON NETWORK OF WORKSTATIONS AND A CONVEX C3210 COMPUTER

Summary. In the paper we present an approach to large-scale parallel distibuted computing using cellular automata (lattice gas) and molecular dynamics on a network of workstations and a CONVEX C3210 computer. We also discuss network evaluation on the basis of SLALOM and NAS EP benchmarks.

VERTEILTE BERECHNUNGEN AN WORKSTATIONENNETZEN UND CONVEX C3210 COMPUTER

Zusammenfassung. Im Artikel wurde besprochen die Stellungnahme zu parallelen Berechnungen der großen Skala, umfassend die Simulationen nach Zellenautomaten (Netzgas) und der molekularen Dynamik, realisiert auf verteilte Weise an Workstationennetzen und CONVEX C3210 Computer. Es wurde auch Bewertung der Brauchbarkeit der Netze auf Basis von Benchmark SLALOM und NAS EP dargestellt.

1993

Ne 1601, 1222

1. Wprowadzenie

Ważnym kierunkiem badań w informatyce jest opracowywanie narzędzi umożliwiających szybkie przetwarzanie danych i efektywną realizację obliczeń numerycznych [1]. Zapotrzebowanie na dużą moc obliczeniową wiąże się w szczególności z zagadnieniami należącymi do klasy Grand Challenge Problems [2]. Przeważa opinia, że do realizacji obliczeń naukowych i technicznych dużej skali nie wystarcza jeden typ architektury komputerowej i stąd też pojawiła się koncepcja metakomputera [3] obejmującego systemy wektorowe, grona stacji roboczych i układy wieloprocesorowe o masywnej równoległości, tworzące heterogeniczną sieć, skonfigurowaną i oprogramowaną tak, by dla użytkownika przedstawiała jednorodną całość. Istotne znaczenie mają prace badawcze, których celem jest zebranie praktycznych doświadczeń w zakresie metacomputingu, skoncentrowane obecnie na badaniu wirtualnego komputera sieciowego.

Kilka stacji roboczych (np. IBM RS/6000, HP Apollo, SUN SPARC, DECstation, Silicon Graphics) połączonych siecią wraz z komputerem np. CONVEX jest nową odmianą architektury heterogenicznej o mocy obliczeniowej rzędu mocy superkomputerów, przy znacznie niższej cenie.

Wykorzystanie takiego wirtualnego komputera sieciowego polega na rozłożeniu problemu na kilka zadań i realizowaniu ich na oddzielnych węzłach sieci. O efektywnym wykorzystaniu dużej mocy sieci decyduje właściwe powiązanie algorytmu obliczeniowego, narzędzi programowych i cech charakterystycznych architektury. Stąd też szczególnie ważne są środowiska programowe, umożliwiające wykorzystanie sieci jako systemu wieloprocesorowego.

Obliczenia w zakresie dynamiki przepływów są typowym, ważnym ze względów praktycznych przykładem obliczeń numerycznych dużej skali. Oprócz podejścia tradycyjnego, polegającego na rozwiązywaniu równań Naviera-Stokesa, obecnie coraz częściej zjawiska hydrodynamiczne są badane od podstaw metodami symulacji molekularnej. Stosowane są tu dwie techniki: metoda automatów komórkowych (gazu siatkowego – LGA), polegająca na traktowaniu płynu jako zbioru cząstek o jednostkowej masie, poruszających się z jednostkową prędkością po trójkątnej siatce, oddziałujących jedynie w jej węzlach, i metoda dynamiki molekularnej (MD), której istotą jest rozwiązywanie równań ruchu dla układu o dynamice wynikającej z oddziaływań między cząstkami. Ze względu na rozmiary symulowanych układów (ponad 10⁸ węzlów siatki, ponad 10⁵ cząstek) oraz liczby wymaganych kroków symulacji, obie metody wymagają użycia dużych mocy obliczeniowych. Można je uzyskać korzystając z komputera sieciowego.

2. Sprzęt i środowiska programowe

W tabeli 1 przedstawiono podstawowe dane stacji roboczych i komputera CONVEX C3210, pracujących w sieci Ethernet, na których realizowano badania.

Charakterystyki węzłów sieci							
Komputer (nazwa)	zegar, MHz	RAM, MB	OS				
SUN SPARCstation SLC (SLC)	20	8	SunOS v4.1.1; -03				
SUN SPARCserver 470 (S470)	40	48	SunOS v4.1.1; -O3				
SUN SPARCstation 2 (SS2)	40	32	SunOS v4.1.1; -03				
SUN SPARCstation IPX (IPX)	40	48	SunOS v4.1.3; -O3				
IBM RS/6000-320 (RS6)	20	16	AIX v3.1; -0				
HP Apollo 9000/720 (HP9)	50	16	HP-UX 8.05; -O				
CONVEX C3210 (C2)	40 (ns)	128	ConvexOS 10.0; -O2				

W większości przypadków korzystano ze środowiska programowego PVM – Parallel Virtual Machine [4], opracowanego w Oak Ridge National Laboratory (v2.4 i v3.1), dzięki któremu użytkownik może traktować heterogeniczną sieć jak jedną, wirtualną maszynę. PVM składa się z dwóch zasadniczych elementów: procesu *pvmd*, instalowanego na każdej maszynie sieci, i biblioteki procedur komunikacyjnych. Używano również podobnych do PVM środowisk p4 [5] i Parasoft Express [6] oraz opartego na koncepcji przestrzeni krotek środowiska Network Linda System [7].

3. Symulacja metodą gazu siatkowego na sieci

3.1. Algorytm i sposób zrównoleglenia

Program symulacji metodą gazu siatkowego opracowano opierając się na dwuwymiarowym modelu FHP, w którym cząstki poruszają się w 6 kierunkach po trójkątnej siatce [8]; w każdym węźle może też znajdować się cząstka spoczywająca. Ponadto wprowadzone zostały węzły specjalne: zmieniające kierunek ruchu cząstki na przeciwny, pochłaniające i wprowadzające cząstki do układu. Ewolucja gazu obejmuje etap translacji, podczas którego cząstki gazu przemieszczają się między węzłami, i etap kolizji, w wyniku których następuje zmiana konfiguracji pędów cząstek w poszczególnych węzłach.

Tabela 1

W programie uwzględniono wszystkie kolizje zachowujące masę i energię, co oznacza konieczność rozpatrywania w każdym węźle 76 konfiguracji cząstek. Wartości makroskopowe, opisujące np. przepływ płynu, uzyskuje się przez uśrednianie wartości pędów.

Dzięki kodowaniu w jednym słowie binarnych wartości pędów możliwa jest znaczna redukcja potrzebnej pamięci i czasu obliczeń poprzez użycie operacji logicznych: każda binarna wartość oznacza obecność lub brak w danym węźle cząstki poruszającej się w określonym kierunku. Oznacza to, że nawet na maszynie sekwencyjnej jest możliwa równoległa aktualizacja stanów kilku węzłów.

Program równoległy opracowano zgodnie z dekompozycją domenową: każda domena obejmuje pas siatki i w każdej z nich niezależnie realizowana jest ewolucja siatki i wyznaczanie wartości uśrednionych. Synchronizacja obliczeń następuje w każdym kroku, po operacjach wprowadzania oraz usuwania cząstek z układu i rozwiązywania kolizji, a przed realizacją operacji translacji i obejmuje wymianę skrajnych wierszy między sąsiednimi pasami siatki.



Rys. 1. Czas obliczeń LGA dla różnych klastrów stacji roboczych Fig. 1. Execution time of LGA for different homogenous clusters of workstations



Rys. 2. Przyspieszenie dla rozproszonego programu LGA na sieci stacji SLC Fig. 2. Speedup for distributed LGA program on SLC workstation network

Program hostowy przekazuje do programów nodowych opis geometrii symulowanego układu oraz odbiera od nich uśrednione (co krok uśredniania) wartości pędów. Programy nodowe generują stan początkowy symulowanego pasa siatki, a następnie realizują ewolucję i uśrednianie. Ze względu na małą ilość operacji wykonywanych przez program hostowy, na tym samym węźle sieci umieszczony jest również jeden program nodowy.

3.2. Wyniki testów

Przykładowe charakterystyki równolegiego programu LGA przedstawiono na rys. 1 i rys. 2.

Rys. 1 pokazuje czas obliczeń na jeden węzeł sieci i jeden krok obliczeń, τ , uzyskany na trzech izolowanych, nieobciążonych klastrach sieci w środowisku PVM 3.1. Zaczernione znaki pokazują czas ścienny dla programu sekwencyjnego. Każdy program nodowy realizował obliczenia dla pasa obejmującego 307200 węzłów siatki, z uśrednianiem co 25 kroków symulacji. Czas obliczeń nie obejmuje operacji rozesłania opisu geometrii symulowanego układu i wygenerowania stanów początkowych poszczególnych domen. Dla silnie obciążonej sieci SLC uzyskano podobny przebieg jak na rys. 1, z tym, że τ dla jednego procesora wynosił 21.6 μ s, a dla dziewięciu - 2.5 μ s.

Zauważono, że efektywność obliczeń rośnie (wolno) z rozmiarami symulowanego układu, np. 4-krotne powiększenie układu dawało zmniejszenie czasu obliczeń, τ , o ok. 5%. Podobny wpływ ma zwiększanie kroku uśredniania.

4. Dynamika molekularna na sieci heterogenicznej

4.1. Algorytm i sposób zrównoleglenia

Badany algorytm dynamiki molekularnej jest przydatny do symulacji zjawisk w mikrokapilarach: pudło obliczeniowe zawierające N cząstek oddziałujących siłami krótkozasięgowymi ma kształt długiego cylindra. Periodyczne warunki brzegowe nałożono jedynie wzdłuż osi cylindra (z). Na podstawie promienia obcięcia r_C można wyznaczyć liczbę n_C cząstek sąsiednich oddziałujących z daną cząstką. W celu wyznaczenia oddziaływań między cząstkami są one sortowane ze względu na ich współrzędną z, a wyznaczanie oddziaływań opiera się na przesuwaniu pudła obliczeniowego względem jego kopii od 1 do n_C [9, 11].

Równoległy algorytm skonstruowano opierając się na dekompozycji domenowej: pudło obliczeniowe podzielono wzdłuż osi z na luźno powiązane domeny. W każdej z nich są niezależnie wyznaczane: oddziaływania między cząstkami, między cząstkami i ścianami pudła, rozwiązywane są równania ruchu, wyznaczane uśrednione wartości wielkości termodynamicznych i sortowane cząstki. Komunikacja między domenami związana jest z przemieszczaniem się cząstek, wyznaczaniem oddziaływań na granicy domen, przekazywaniem wartości uśrednionych w każdej domenie w celu wyznaczenia charakterystyk globalnych układu oraz równoważeniem obciążenia [12]. Poszczególne domeny są przyporządkowane węzłom sieci. Do wyznaczania wartości sil konieczne jest kopiowanie danych dotyczących n_C cząstek z sąsiedniej domeny.

Programy hostowe i nodowe, napisane w języku C, są podobne: różnice są związane z koniecznością realizacji przez program hostowy operacji wejścia – wyjścia, wyznaczania globalnych wartości i równoważenia obciążenia.





Rys. 3. Czas obliczeń programu MD dla różnych jednorodnych klastrów stacji roboczych

Fig. 3. Execution time of MD program for different homogenous clusters of workstations Rys. 4. Porównanie środowisk oparte na programie MD; do SLC dołączano: S470, SS2, SS2

Fig. 4. Comparison of environments using MD program; to SLC (host) – S470, SS2, SS2 were added

Dla zwiększenia efektywności obliczeń realizowanych na sieci heterogenicznej, w warunkach zmiennego obciążenia, zastosowano procedurę dynamicznego równoważenia obciążenia (load-balancing – LB) poszczególnych wezłów sieci. Celem LB jest uzyskanie tego samego czasu ściennego (wall-clock) na wszystkich wykorzystywanych węzłach sieci w kolejnych krokach symulacji. Można to uzyskać poprzez zmianę liczby cząstek w poszczególnych domenach (węzłach sieci). Ze względu na liniową zależność czasu obliczeń od ilości cząstek, liczba cząstek δ_k , o którą w danym kroku powinna być zmodyfikowana liczba cząstek w węźle k, zależy od stosunku ściennego czasu obliczeń danego węzła do wartości średniej czasów ściennych realizacji obliczeń na każdym węźle. Na początku symulacji każda domena zawiera tę samą liczbę cząstek. Równoważenie obciążenia jest realizowane przez program hostowy.

4.2. Wyniki testów

Na rys.3 przedstawiono znormalizowany czas obliczeń (na jeden krok symulacji i na jedną cząstkę), τ , uzyskany na trzech różnych, homogenicznych klastrach stacji roboczych.

Każda z domen symulowanego układu zawierała 32004 cząstek. Badania przeprowadzono na nieobciążonych klastrach. We wszystkich przypadkach uzyskano prawie liniową zależność przyspieszenia od liczby węzłów sieci i efektywność bliską 1.



- Rys. 5. Czas obliczeń w kolejnych krokach dla klastra obejmującego S470, C2, 2*SS2, HP9, 2*RS6
- Fig. 5. Wall-clock time for every timestep, T, for the network of 7 machines: S470, C2, 2*SS2, HP9, 2*RS6

Uzyskane czasy obliczeń są zbliżone do czasów osiąganych na superkomputerach.

Rys. 4 przedstawia porównanie czasów wykonania dla różnych środowisk programowych na heterogenicznej sieci. Do hosta SLC stopniowo włączano: S470, SS2, SS2. Badania wykonano przy typowym obciążeniu, a w trakcie obliczeń działała procedura dynamicznego równoważenia obciążenia. Środowiska PVM (v2.4), p4 i Express dają zbliżone czasy wykonania, natomiast wygoda w programowaniu w środowisku Linda okupiona jest mniejszą efektywnością obliczeń.

Przydatność dynamicznego równoważenia obciążenia ilustruje rys. 5, a jego wpływ na zmianę ilości cząstek na poszczególnych węzłach sieci obejmującej komputer CONVEX C3210 przedstawiono na rys. 6 i rys. 7. Obliczenia zostały wykonane z wykorzystaniem dwóch wersji środowiska PVM. Przedstawiony na obu rysunkach czas τ uzyskano przez bieżące uśrednianie w oknie 100 kroków. Widać, że niektóre ze stacji roboczych uzyskały obciążenie wynikające z maksymalnej dostępnej pamięci.







Rys. 7. Czas obliczeń τ i rozkład cząstek dla 8 węzłów sieci dla PVM v3.1 Fig. 7. Execution wall-clock time, τ , and particle distribution between 8 processing nodes under PVM v3.1

5. Ocena sieci za pomocą benchmarków

5.1. Omówienie benchmarków

Powszechnie stosowanym sposobem oceny osiągów sprzętu komputerowego są benchmarki, używane do porównań maszyn sekwencyjnych i wektorowych. Rozproszone obliczenia na sieci stacji roboczych i komputerze CONVEX C3210

Prowadzone są próby zastosowania ich do maszyn wieloprocesorowych i układów rozproszonych. Do oceny sieci jako narzędzia do intensywnych obliczeń zaadaptowano dwa benchmarki: SLALOM [13, 14] i EP z zestawu NAS [15, 16].

Zasadniczą częścią benchmarku SLALOM jest rozwiązywanie układu równań liniowych powstałych z dyskretyzacji równań różniczkowych cząstkowych dla obszaru podzielonego na n fragmentów. SLALOM służy do pomiaru ilości operacji wykonywanych w czasie ok. 1 minuty, podając zarówno liczbę fragmentów obszaru, dla których uzyskano rozwiązanie, jak i osiągnięte MFLOPS.

W omawianych badaniach dostosowano równoległy benchmark przeznaczony pierwotnie na komputer nCUBE. Zastosowano statyczne równoważenie obciążenia oparte na obciążeniu poszczególnych węzłów sieci w chwili rozpoczęcia testu.

Benchmark EP (Embarrassingly Parallel) generuje pary liczb losowych o rozkładzie normalnym i wyznacza rozkład wartości większej ze składowych. Rozmiar problemu określony jest przez liczbę generowanych par $n = 2^m$. Zrównoleglenie (algorytmiczne) polega na rozbiciu pojedynczej pętli na pętle wykonywane na różnych węzłach, przy czym długość każdej z tych pętli zależy od obciążenia danego węzła.

Zastosowano dwie metody statycznego równoważenia obciążenia. Pierwsza z nich (LB-1) jest oparta na pomiarze czasu wykonywania zestawu operacji zmiennoprzecinkowych, druga (LB-2) daje rozdział na poszczególne węzły sieci zgodnie z wykonaniem EP dla mniejszego problemu.

5.2. Wyniki

Wyniki testów z wykorzystaniem benchmarku SLALOM przedstawiono w tabeli 2.

Ocena sieci za pomocą benchmarku SLALOM (środowisko PVM v.2.4)							
3.62	Komputery	bez LB		z LB			
No.	w sieci	Fragmenty	MFLOPS	Fragmenty	MFLOPS		
1.	S470, C2, SS2, SLC	42	0.61	nd all <u>o ques</u> to	10 10 10 10 10 10		
2.	S470, C2, RS6, SLC	42	0.63	151	2.2		
3.	S470, C2, RS6, RS6	129	1.72	387	4.1		
4.	S470, C2, RS6, HP9	131	1.70	394	4.3		
5.	S470, C2, 4*RS6, 2*SS2	107	1.47	475	6.3		

Tabela 2

71

Dla porównania, dla nCUBE 2 z 4 procesorami (20 MHz, Fortran+coded BLAS) osiągi wynoszą 560/3.8 (fragmentów/MFLOPS) [14]. Odpowiada to w przybliżeniu wynikowi 475/6.3 dla sieci z 8 węzłami (pozycja 5 w tabeli 5.2.). Dla MasPar MP-1 (12.5 MHz, C) z 2048 procesorami uzyskano 1055/22 [14].

Na rys. 8 przedstawiono wyniki benchmarku EP dla 5 węzłów sieci. Uzyskany czas ścienny wynosi 125 s. Dla tego benchmarku uzyskano czasy cpu: 315.4 s na IBM RS/6000– 320 i 11.6 s na Cray Y-MP.

6. Podsumowanie

Uzyskane wyniki wskazują na przydatność sieci komputerowej do realizacji obliczeń dużej skali w sposób rozproszony. Opracowane algorytmy równolegle umożliwiają realizację symulacji układów o dużej liczbie cząstek (MD) i węzłów (LGA). Zostały one wykorzystane w symulacji wybranych zjawisk fizycznych [17, 18].



Rys. 8. Czas ścienny dla EP na sieci komputerów (kolejno włączane: S470, C2, RS6, SLC, RS6)

Fig. 8. Wall-clock time for EP kernel run on the network of computers (included in the order: S470, C2, RS6, SLC, RS6)

Opierając się na przedstawionych w pracy wynikach prowadzi się badania innych, ważnych z praktycznego punktu widzenia metod obliczeniowych (np. metoda elementów skończonych [19]) oraz przydatności innych środowisk programowych (np. TOPSYS Rozproszone obliczenia na sieci stacji roboczych i komputerze CONVEX C3210

- Tools for Parallel Systems, PCN - Program Composition Notation [20]).

Autorzy dziękują mgr inż. W. Aldzie i mgr inż. D. Nikołowowi za szereg cennych uwag i pomoc w realizacji symulacji.

Praca była częściowo finansowana z projektów badawczych KBN PB 8 S503 021 05 i PB 8 S503 022 05.

LITERATURA

- Tvrdik, P. (ed.): ISIPCALA'93 International Summer Institute on Parallel Computer Architectures, Languages, and Algorithms, Prague, Czech Republic, July 5-9, 1993.
- [2] Wilson, K.G., Computational Science and Grand Challenges, Cray Channels, Summer, 1992, pp.5-6 (A Cray Research, Inc. Publications).
- [3] Metacomputing. HP CONVEX a shared vision, Seminar, June 17th 1993, University of Warsaw.
- [4] Sunderam, V.S., PVM: a framework for parallel distributed computing, Concurrency: Practice and Experience, 2 (1990) 315-339.
 Geist, A. Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., PVM 3.0 A users' guide and reference manual, ORNL/TM-12187, USA (February, 1993).
- [5] Butler, R., and Lusk, E., User's guide to the p4 programming system, Argonne National Laboratory, ANL-92/17, October 1992.
- [6] Express v3.0 Introductory Guide.
 Express v3.0 Operational Manual, ParaSoft Corp., Pasadena (1990).
- [7] Carriero, N.J., and Gelernter, D.H., Linda in context, Comm. ACM 32 (1989) 444-458.
 Bjornson, R., Kolb, C., and Sherman, A., Ray Tracing with Network Linda, SIAM News, 24 1 (1991).
 Carriero, N., Gelernter, D., Kaminsky, D., Westbrook, J., Adaptive paralelism with Piranha, Yale University, 1993 (draft).
- [8] Boghosian. B.M., Lattice gas illustrate the power of cellular automata in physics, Computers in Physics, Nov/Dec 1991, pp.585-590.

73

- [9] Mościński, J., Kitowski, J., Rycerz, Z.A., and Jacobs, P.W.M., A vectorized algorithm on the ETA 10-P for molecular dynamics simulation of large number of particles confined in a long cylinder, Comput. Phys. Commun., 54 (1989) 47.
- [10] Mościński, J., Bargieł, M., Kitowski, J., Skotniczny, Z., Rycerz, Z.A., and Jacobs, P.W.M., Vectorized molecular dynamics algorithms for very large number of particles, Proc. of 1989 International Conference on Supercomputing, June 5-9, 1989, Crete, Greece (ACM Publications, Baltimore, USA).
- [11] Kitowski, J., and Mościński, J., Microcomputers against Supercomputers? On the geometric partition of the computational box for vectorized MD algorithms, Mol. Simul., 8 (1992) 305-319.
- [12] Boryczko, K., Bubak, M., Kitowski, J., Mościński, J., and Pogoda, M., Molecular dynamics and lattice gas parallel algorithms for transputers and networked workstations, EUROMECH Coll.287, Sept. 21-25, 1992, Cagliari, Italy; also Transport Theory and Statistical Physics, in print.
- [13] Gustafson, J., Rover, D., Elbert. S., and Carter, M., SLALOM Surviving Adolescence, Supercomputing Review, Dec. 91 (1991) 54.
- [14] The SLALOM Benchmark Report, unpublished results (1990).
- [15] Bailey, D.H., Barszcz, E., Barton, J.T., Browning, D.S., Carter, R.L., Dagum, L., Fatoohi, R.A., Frederickson, P.O., Lasinski, T.A., Schreiber, R.S., Simon, H.D., Venkatakrishnan, V., and Weerantuga, S.K., The NAS parallel benchmarks – summary and preliminary results, private information.
- Bailey, D.H., Barton, J., Lasinski, T.A., and Simon, H.D., The NAS parallel benchmarks, Report RNR-91-002 Rev.2 (1991), NASA Ames Research Center, Moffett Field.
 NAS parallel benchmark results, Proc. Supercomputing'92 (1992) 386-393.
 NAS parallel benchmark results. IEEE Parallel and Distributed Technology, Feb. 1993 43-51.
- [17] Mościński, J., Boryczko, K., Kitowski, J., Pogoda, M., Alda, W., Bubak, M., Dzwinel, W., Słota, R., and Noga, M., Distributed molecular dynamics on a cluster of workstations and Convex supercomputer. Proc. of the Convex Worldwide User Group Conference, March 21-25, 1993, Richardson.

- [18] Alda, W., Dzwinel, W., Kitowski, J., Mościński, J., and Yuen, D.A., Penetration mechanics via molecular dynamics, Supercomputer Institute Research Report, UMSI 93/58 (1993), University of Minnesota, and Army High Performance Computing Research Center, report 93-037 (1993).
- [19] Polednia, B., Bubak, M., Chrobak, R., Mościński, J., Pietrzyk, M., Program do wyznaczania odkształceń metodą elementów skończonych – implementacja w języku C i próba zrównoleglenia, seminarium Zastosowanie komputerów w przetwórstwie metali, 16. listopada 1993, AGH Kraków.
- [20] Foster, I., and Tuecke, S., Parallel programming with PCN, Argonne National Laboratory, ANL-91/32 rev.2, January 1993.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do redakcji 20 września 1993.

Abstract

In the paper we present results of investigations of parallel algorithms for two numerically intensive computing methods: lattice gas automata (LGA) and molecular dynamics (MD) for distributed computing on a heterogenous network of workstations with CONVEX C3210 computer (Tab. 1). LGA considers the fluid as an ensemble of particles of unit mass moving with unit velocity on a triangular lattice, while in MD fluid is modelled by a set of particles interacting via a given potential.

Mainly geometric (domain) decomposition of the problem was used in the LGA and MD programs. Aspects of practical implementation on the network are discussed. Characteristics of the algorithms (execution time and speedup) are shown (Fig. 1 and 2 for LGA, Fig. 3 for MD). The following network-oriented software for distributed parallel processing was applied: Parallel Virtual Machine. Parasoft Express, p4 Programming System and Network Linda System (Fig. 4). Some of the algorithms incorporate load-balancing procedure for efficiency increase (Fig. 5. 6 and 7).

For the purpose of network evaluation a parallel version of standard SLALOM benchmark and EP kernel from NAS parallel benchmark were adopted (Tab.2, Fig.8).

The reported experiments show the efficiency of distributed approach to a class of large scale computing.