УДК 681.3.012:621.1

Г.Г. Швачич

ОСОБЛИВОСТІ КОНСТРУЮВАННЯ І МОДЕЛЮВАННЯ ВИСОКОПРОДУКТИВНОГО ІНТЕГРОВАНОГО СЕРЕДОВИЩА НА БАЗІ ПЕРСОНАЛЬНОГО ОБЧИСЛЮВАЛЬНОГО КЛАСТЕРА

Запропонований підхід до розробки сучасних обчислювальних математичних технологій знаходження розв'язку багатомірних нестаціонарних задач металургійного виробництва. Процес моделювання реалізовано на основі застосування багатопроцесорних обчислювальних систем кластерного типа. Висвітлюються питання конструювання персонального обчислювально кластера. Блок обчислювальних вузлів персонального п'яти – вузлового кластера створений на основі використання системної плати PC2500 від VIA з інтегрованими процесорами C3-1,8. ПОК працює під управлінням ОС Linux.

Approach is offered to development of modern calculable mathematical technologies of finding of decision of multidimensional unstationary tasks of metallurgical production. The process of design is realized on basis application of the multiprocessor computer systems of cluster type. The questions of constructing of the personal of calculable cluster are lighted. The block of computing units of the personal five-sites central cluster - is created on the basis use of system boards PC2500 from VIA with integrated processors C3-1, 8. PCC works under control of OS Linux.

Предложен подход к разработке современных вычислительных математических технологий нахождения решения многомерных нестационарных задач металлургического производства. Процесс моделирования реализован на основе применения многопроцессорных вычислительных систем кластерного типа. Освещаются вопросы конструирования персонального вычислительно кластера. Блок вычислительных узлов персонального пяти – узлового кластера создан на основе использования системных плат PC2500 от VIA с интегрированными процессорами C3-1,8. ПВК работает под управлением ОС Linux.

Постановка проблеми досліджень

Застосування паралельних обчислювальних систем (*ПОС*) є стратегічним напрямом розвитку обчислювальної техніки. Ця обставина викликана не тільки принциповим обмеженням максимально можливої швидкодії звичайних послідовних ЕОМ, але і практично постійним існуванням обчислювальних задач, для вирішення яких можливостей існуючих засобів обчислювальної техніки завжди виявляється недостатньо. До таких задач відносяться, наприклад, чисельне моделювання процесів гідродинаміки і металургійної теплофізики [1,2,3], задачірозпізнавання образів, оптимізаційні задачіз великим числом параметрів, моделювання клімату, генна інженерія, проектування інтегральних схем, аналіз забруднення навколишнього середовища [4], рішення широкого кола багатовимірних нестаціонарних задач[5] і т.д. Наприклад, в [6] висвітлюються питання моделювання за допомогою методу кінцевих елементів процесу волочіння дроту в роликових волоках. Задачу моделювання автори здійснювали на основі використання

 ISSN 1991-7848

комерційного програмного забезпечення FORGE3 фірми Transvalor (*Франція*). На комп'ютері з процесором Celeron 2.4 задачарозв'язувалася близько 12 годин. Очевидно, що вказаного класу задачі можуть успішно розв'язуватися тільки за допомогою розподіленого моделювання.

Організація паралельності обчислень, коли в один і той же момент часу виконується одночасно декілька операцій обробки даних, здійснюється, в основному, введенням надмірності функціональних пристроїв (*багатопроцесорності*). В цьому випадку можна досягти прискорення процесу рішення обчислювальної задачі і збільшення продуктивності обчислень, якщо здійснити розділення обчислювального алгоритму на інформаційно незалежні частинії організувати виконання кожної частини обчислень на різних процесорах. Подібний підхід дозволяє виконувати необхідні обчислення з меншими витратами часу, і можливість отримання максимального прискорення обмежується тільки числом наявних процесорів і кількістю "незалежних" частин у виконуваних обчисленнях.

Протеслід зазначити, що нині застосування паралелізму не набуло такого широкого поширення, як це вже багато разів передбачалося багатьма дослідниками. Однією з можливих причин подібної ситуації була до недавнього часу висока вартість високопродуктивних систем (можливість придбання суперЕОМ могли собі дозволити тільки крупні компанії і організації). Сучасна тенденція побудови паралельних обчислювальних комплексів з типових конструктивних елементів (мікропроцесорів, мікросхем пам'яті, комунікаційних пристроїв), масовий випуск яких освоєний промисловістю, понизила вплив цього чинника і зараз практично кожен споживач може мати в своєму розпорядженні багатопроцесорні обчислювальні системи (*БОС*) досить високої продуктивності.

Інша і, мабуть, тепер основна причина стримання масового розповсюдження паралелізму полягає в тому, що для проведення паралельних обчислень необхідне "паралельне" узагальнення традиційної послідовної технології рішення задач на ЕОМ. Так, чисельні методи у разі багатопроцесорних систем повинні проектуватися як системи паралельних і взаємодіючих між собою процесів, що допускають виконання на незалежних процесорах. Вживані алгоритмічні мови і системне програмне забезпечення повинні забезпечувати створення паралельних програм, організовувати синхронізацію і взаємо виключає асинхронних процесів і т.п.

Беручи до уваги відмічене, можна відзначити, що *паралельні обчислення є актуальною, перспективною і привабливою областю застосуванняобчислювальної техніки*. Крім того, паралельні обчислення є складною науково-технічною проблемою. Тим самим знання сучасних тенденцій розвитку ЕОМ і апаратних засобів для досягнення паралелізму, уміння розробляти моделі, методи і програми паралельного рішення задач обробки даних слід віднести до важливих кваліфікаційних характеристик сучасного фахівця з прикладної математики, інформатики і обчислювальної техніки.

Аналіз останніх досліджень і публікацій

Історія обчислювальних кластерів почалася 1994 року. Піонером в цій справі є науково-космічний центр NASA - *Goddard Space Flight Center (GSFC)*, точніше створений на його основі CESDIS *(Center of Excellence in Space Data and Information Sciences)*. Фахівцями GSFC влітку 1994 роки був зібраний перший кластер, що складався з 16 комп'ютерів 486DX4/100MHz/16Mb RAM і трьох паралельно працюючих 10Mbit мережених адаптерів. Даний кластер, який був названий "Beowulf", створювався, як обчислювальний ресурс проекту Eart and Space Sciences Project (*ESS*).

Чотири роки опісля в 1998 році, в Лос-Аламосській національній лабораторії (*США*) астрофізик Майкл Уоррен і інші вчені з групи теоретичної астрофізики побудували суперкомп'ютер, який був Linux-кластером на базі процесорів Alpha 21164A з тактовою частотою 533 Мгц. Спочатку Avalon складався з 68 процесорів, потім був розширений до 140. У кожному вузлі встановлено по 256 Мбайт оперативної пам'яті, жорсткий диск на 3 Гбайт і мережений адаптер Fast Ethernet. Загальна вартість проекту Avalon склала 313 тис. дол., а показана їм продуктивність на тесті LINPACK - 47,7 GFLOPS дозволила йому зайняти 114 місце в 12-й редакції списку Top 500 поряд з 152-процесорною системою IBM RS/6000 SP. У тому ж 1998 році на найпрестижнішій конференції у області високопродуктивних обчислень *Supercomputing'98* творці Avalon представили доповідь "*Avalon: An Alpha/Linux Cluster Achieves 10 Gflops for $150k*", що одержала першу премію в номінації "якнайкраще відношення ціна/продуктивність". В даний час Avalon активно використовується в астрофізичних, молекулярних і інших наукових обчисленнях.

Взагалі відмітимо, що проблема створення високопродуктивних обчислювальних систем належить до найбільш складних науково-технічних завдань сучасності і її розв’язок можливий тільки за умов всебічній концентрації зусиль багатьох талановитих вчених і конструкторів, припускає використання всіх останніх досягнень науки і техніки, і вимагає значних фінансових інвестицій. Проте, досягнуті останнім часом успіхи в цій області вражають. Так, в рамках прийнятої в США в 1995 р. програми "Прискореної стратегічної комп'ютерної ініціативи" (*Accelerated Strategic Computing Initiative - ASCI*) [8] було поставлене завдання збільшення продуктивності суперЕОМ в 3 рази кожні 18 місяців і досягнення рівня продуктивності в 100 трильйонів операцій в секунду (*100 терафлопс*) в 2004 р. Однією з найбільш швидкодіючих суперЕОМ в даний час є комп'ютер SX-6 японської фірми NEC з швидкодією одного векторного процесора близько 8 мільярдів операцій в сек. (*8 Гфлопс*). Досягнуті показники швидкодії для багатопроцесорних систем набагато вищі. Так, система ASCI Red фірми Intel (*США, 1997*) має граничну (*пікову*) продуктивність 1,8 трильйонів операцій в секунду (*1,8 Тфлопс*). Система ASCI Red включає в свій склад 9624 мікропроцесорів PentiumPro з тактовою частотою 200 Мгц, загальний об'єм

 ISSN 1991-7848

оперативної пам'яті 500 Гбайт і має вартість 50 млн. доларів (*тобто вартість 1 Мфлопс складає близько 25 доларів).*

Відмітимо, що проблематика паралельних обчислень є надзвичайно широкою областю теоретичних досліджень і практично виконаних робіт і звичайно підрозділяється на наступні напрямки діяльності:

- *розробка паралельних обчислювальних систем*, даний напрямок присвячений принципам побудови паралельних обчислювальних систем [3, 10, 11,12];
- *аналіз ефективності паралельних обчислень*, даний напрямок присвячений оцінюванню одержуваного прискорення обчислень і ступеня використання всіх можливостей комп'ютерного устаткування при паралельних способах рішення задач [13,14];
- *створення і розвиток паралельних алгоритмів* для вирішення прикладних задач в різних областях практичних додатків [15-18];
- *розробка паралельних програмних систем*, даний напрямок присвячений роботам, пов'язаних з математичним моделюванням паралельних програм [19-22];
- *створення і розвиток системного програмного забезпечення* для паралельних обчислювальних систем, обговорення питань даного напряму присвячене забезпеченню мобільності (перенесимості між різними обчислювальними системами) створюваного прикладного програмного забезпечення [23-25].

Останнім часом наголошується істотний інтерес до побудови персональних обчислювальних кластерів (*ПОК*) на базі стандартних загальнодоступних технологій і компонентів [26,27]. Цей інтерес обумовлений рядом чинників. Відзначимо основні з них. По - перше, зростання, відповідно до потреб ринку, продуктивності таких стандартних мережених технологій як Ethernet (послідовне підвищення швидкості передачі — 10, 100, 1000 Мбіт/с, застосування комутаторів замість моделі з середовищем даних, що розділяються) дозволив розглядати їх як комунікаційне середовище для багатопроцесорних обчислювальних систем. По-друге, одним з важливих чинників стало збільшення популярності вільно поширюваної операційної системи Linux. Ця операційна система спочатку позиціонувалася як варіант UNIX для платформ на базі архітектури Intel, але достатньо швидко з'явилися версії для інших популярних мікропроцесорів, у тому числі і для лідерів з продуктивності протягом останніх років — мікропроцесорів Alpha.

З урахуванням економічних реалій нашої країни використання систем, *побудованих на базі стандартних технологій, стає більш ніж актуально*. Причому залежно від завдань і бюджету проекту можливі достатньорізноманітні варіанти конфігурації. У найбільш доступній конфігурації використовуються стандартні материнські плати для процесорів Intel Pentium III і мережені адаптери Fast Ethernet. Вузли кластера об'єднуються між собою за допомогою комутатора Fast Ethernet на відповідні число

портів. Кількість вузлів і їх конфігурація залежить від вимог, що пред'являються до обчислювальних ресурсів конкретними завданнями і доступних фінансових можливостей.

У зв'язку з відміченим, *основними цілями даної статті є:*

- висвітлення особливостей конструювання персонального обчислювального кластера на базі стандартних мережених технологій;
- особливості розвитку системного програмного забезпечення для персонального обчислювального кластера;
- розробка паралельних алгоритмів для розв'язування широкого кола прикладних задач;
- моделювання задач, які розглядуються, на персональному обчислювальному кластері.

Виклад основного матеріалу досліджень

Деякі особливості конструювання та функціонування кластерних обчислювальних систем

Кластер - це модульна багатопроцесорна система, створена на базі стандартних обчислювальних вузлів, з'єднаних високошвидкісним комунікаційним середовищем.

Нині слова «кластер» та «суперкомп'ютер» у значній мірі синоніми, але до поняття цього апаратні засоби пройшли довгий цикл еволюції. Перші 30 років з часів заснування комп'ютерів, аж до середини 1980-х р., під «суперкомп'ютерними» технологіями розуміли виключно виготовлення спеціалізованих особливо потужних процесорів. Однак поява однокристального мікропроцесора практично стерло різницю між «масовими» та «особливо потужними» процесорами, і з цього моменту єдиним способом створення суперкомп'ютера став шлях об'єднання процесорів для паралельного рішення однієї задачі. Приблизно до середини 1990-х р. основний напрямок розвитку суперкомп'ютерних технологій було пов'язано зі створенням спеціалізованих багатопроцесорних систем із масових мікросхем.

Один з сформованих підходів – технологія SMP (*Symmetric Multi Processing*), мав на увазі об'єднання багатьох процесорів з використанням загальної пам'яті, що сильно спрощувало програмування, але зумовляло високі вимоги до самої пам'яті. Зберегти швидкодію таких систем при збільшенні кількості вузлів до десятків було практично неможливо. Крім того, цей підхід став самим дорогим в апаратній реалізації.

На порядок більш дешевшим та практично з безмежними спроможностями до масштабування стала технологія MPP (*Massively Parallel Processing)*, при якій незалежні спеціалізовані обчислювальні модулі об'єднались спеціалізованими каналами зв'язку, причому й перші й другі розроблялись під конкретний суперкомп'ютер і ні в яких інших цілях не використовувались.

Ідея створення так званого кластера робочих станції фактично стала розвитком технології MPP, бо логічно MPP-система не сильно різнилась від

 ISSN 1991-7848

звичайної локальної мережі. Локальна мережа стандартних персональних комп'ютерів, при відповідному ПЗ, що використовувалась як багатопроцесорний суперкомп'ютер, й стала першою ланкою сучасного кластеру. Ця ідея отримала більш сучасне втілення в середині 1990-х р., коли дякуючи широкому оснащенню ПК високошвидкісною шиною PCI і появою дешевої, але досить швидкодіючої мережі Fast Ethernet кластери стали наздоганяти спеціалізовані MPP-системи за комунікаційним можливостям. Це означало, що повноцінну MPP-систему можна було створити з стандартних серійних комп'ютерів при допомозі серійних комунікаційних технологій, причому така система обходилася дешевше в середньому на два порядки.

Сфера використання кластерних систем нині аніскільки не вужче, ніж суперкомп'ютеров з іншою архітектурою: вони не менш успішно справляються із задачею моделювання самих різних процесів і явищ.

Саме розвиток кластерних технологій зробило високу продуктивність обчислення широко доступною і дозволило самим різним установам скористатись їх перевагами. Наведемо область розподілення обчислень при використанні 500 самих потужних комп'ютерів світу:

44,3% - електрона, автомобільна, авіаційна та ін. галузі важкої промисловості та машинобудування;

20% - наука й навчання, суперкомп'ютерні центри;

18% - доводитися на погодні й кліматичні дослідження;

7% - ядерні, космічні, енергетичні й воєнні державні програми;

3,5% - фінансові компанії й банки та ін.

Сьогодні можна стверджувати, що кластерні системи успішно застосовуються для всіх задач суперкомп'ютінга - від обчислень для науки й промисловості до управління базами даних. Практично усі програми, що потребують потужніх обчислень, мають зараз паралельні версії, які дозволяють розбивати задачу на фрагменти й обчислювати її паралельно на багатьох вузлах кластеру. Наприклад, для інженерних обчислень традиційно використовуються так звані сіткові методи [3,4,5,12,17,19,21,22,30,31], коли область обчислень розбивається на осередки, кожен з яких є окремою одиницею обчислень. Ці осередки обчислюються незалежно на різних вузлах кластера, а для отримання загальної картини на кожному кроці обчислень виконується обмін даними.

Кластерні рішення – це найбільш економічно зумовлений вибір. На відміну від більшості серверних систем з загальною пам'яттю кластерні рішення легко масштабуються до систем більшої продуктивності. Таким чином, при збільшені обчислювальних вузлів до необхідної продуктивності обчислень не обов'язково придбавати нову систему - можна додати стандартні обчислювальні вузли й легко нарощувати стару.

Кластерні рішення мають найкраще на сьогоднішній день співвідношення ціна/продуктивність та мають істотно більш низьку вартість обслуговування. Це досягається за допомогою спроможності до масштабування й

використанню стандартних загальнодоступних компонентів ціна яких постійно знижується.

Крім того, кластерна архітектура забезпечує відмінну відказостійкість системи: при виході з ладу одного, чи декількох обчислювальних модулів (чи вузлів) кластер не втрачає робото- спроможності, й нові задачі можуть бути запущені на меншому числі вузлів. Несправний вузол легко й швидко виймається з стійки й заміняється новим, який одразу ж включається в роботу. Це можливо дякуючи комутованій топології сучасних системних мереж, коли обмін повідомленнями між двома вузлами може відбуватися багатьма шляхами.

На даний час кластери конструюються з обчислювальних вузлів на базі стандартних процесорів, з'єднаних високошвидкісною системною мережею (інтерконектом), а також, як правило, допоміжною й сервісною мережами. Іноді лідери-виробники пропонують свій формфактор: наприклад, IBM, Verari, LinuxNetworx та інші компанії пропонують обчислювальні вузли на основі блэйд-технологій («*леза*»), які забезпечують високу щільність установки, але дещо удорожчують конструктивні рішення.

Кластер - це складний програмно-апаратний комплекс, і задача його побудови не закінчується об'єднанням великої кількості процесорів в один сегмент. Для того щоб кластер швидко й правильно обчислював задачу, усі комплектуючі повинні бути щільно підібрані один до одного з врахуванням вимог програмного забезпечення, оскільки продуктивність кластерного ПЗ сильно зумовлена від архітектури кластера, характеристик процесорів, системної шини, пам'яті й інтерконекта.

Кластерні системи можуть використовувати дуже різні платформи та типи інтерконектів, і як правило, класифікуються не через набір комплектуючих, а по галузям використання. Виділяють чотири типи кластерних систем:

обчислювальні кластери;

кластери баз даних;

відказостійкі кластери;

кластери для розподілення завантаження.

Сама велика група - обчислювальні кластери. Вона може бути розбита на підгрупи; правда, класифікуються в цій групі вже не обчислювальні машини, а готові програмно-апаратні кластерні рішення. Такі системи «під ключ» мають поперньо встановлене ПЗ, необхідне замовнику для розв'язку його задач. Рішення, оптимізовані для різних програм, різняться підбором компонентів, що забезпечують найбільш продуктивну роботу саме цих програм за найкращим співвідношенні ціна/ якість.

Основні типи готових рішень у світовій практиці:

- промислові кластери для інженерних задач;
- кластери для нафто- й газодобивающей промисловості;

 ISSN 1991-7848

- кластери для досліджень у галузі «наук про життя», або life sciences (пошук нових ліків, генетика, молекулярне моделювання, біоінформатика);
- кластери для стратегічних досліджень (дослідження перегоди й клімату, ядерна фізика й фізика часток, космічні дослідження, оборонні програми);
- кластери для індустрії розваг (комп'ютерна графіка й спец ефекти, комп'ютерні онлайнові ігри);
- кластери для високопродуктивних обчислень у різних галузях науки й навчання.

Деякі особливості прикладного програмного забезпечення кластерних обчислювальних систем

Робота кластерних систем забезпечується чотирма видами спеціалізованих додатків, таких як:

операційні системи (*як правило, Linux*);

засоби комунікації (для обчислювальних кластерів це за

звичай бібліотека MPI *(Message Passing Interface)*);

засоби розробки паралельних додатків;

ПЗ для адміністрування кластерів.

Для написання паралельних програм, використовуються бібліотеки програмування MPI, що забезпечує взаємодію між вузлами кластера. MPI стандартизує набір інтерфейсів програмування, на яких можна робити програми, що легко переносяться на різні кластерні архітектури.

Створення паралельної програми містить у собі дві основних стадії:

послідовний алгоритм піддається декомпозиції (розпаралелюванню), тобто розбивається на незалежно працюючі ланки; для взаємодії між ланками вводяться дві додаткових нематематичних операції: прийом і передача даних;

паралельний алгоритм записується у вигляді програми, у якій операції прийому й передачі записуються в термінах конкретної системи зв'язку між ланками.

Система зв'язку, у свою чергу, містить у собі два компоненти: програмний і апаратний.

Із точки ж зору програмування базових методик дані можуть передаватися:

через розподілювальну пам'ять; синхронізація доступу галузей до такої пам'яті відбувається за допомогою семафорів;

у вигляді повідомлень.

Перший метод є базовим для SMP-машин, другий - для мереж всіх типів.

Стандартом інтерфейсу програміста для кластерів, що підтримують MPP архітектуру, вирішено зробити технологію MPI. Тож утворений MPI Forum, і випущена специфікація, яка повинна задовольняти всі конкретні

розробки. Головна організація проекту - Аргонська національна лабораторія США, саме вона поширює пакет MPICH (*MPI CHameleon*), який перенесений на більшість платформ.

Конструктивна реалізація персонального обчислювального кластера MPP архітектури

Переважна більшість функціонуючих суперобчислювальних установок є фактично багатопроцесорними паралельними обчислювальними системами MPP архітектури (*Massively Parallel Processing*). Багатопроцесорні обчислювальні системи, сконструйовані на локальних мережах, почали називати «кластерними системами» або просто «кластерами». Це пояснюється тим, що логічно MPP - система мало відрізняється від звичайної локальної мережі.

Існує два можливі шляхи побудови кластерного обчислювального комплексу (рис.1):

з'єднання за допомогою локальної мережі (*Ethernet*) персональних ЕОМ (рис. 1а). Причому технічно конфігурації ЕОМ об'єднаних у кластер можуть бути різними й навіть із різними операційними ситемами;

так звані «блейд» серверні рішення (рис.1б), при яких кілька однотипних материнських модулів установлюються в одному корпусі.

За схемою рис. 1а на кафедрі прикладної математики та обчислювальної техніки НМетАУ було сконструйовано відповідний обчислювальний кластер, який дозволив розв’язати проблему математичного моделювання багатовимірювальних задач металургійного виробництва [3,5,12,17,18,29].

Блейд - системи більш компактні й зручні в обслуговуванні, і незначно дорожче в реалізації в порівнянні з першим підходом. Але дякуючи зростаючому попиту та пропозиції «блейд» конфігурацій на нашому ринку, було ухвалене рішення про створення саме «блейд» кластерного пристрою для математичного моделювання багато вимірювальних задач металургійного виробництва.

У якості конструктива було обрано єдиний корпус, що являє собою осередок обчислювальної шафи. Це пов'язане з тим, що, з одного боку, при необхідності можна декілька ПОК розміщати в єдиному корпусі, а з іншого боку - при такому підході забезпечується компактність, успішне охолодження й легкий доступ до гнізд і елементів плат, які налагоджуються. ПОК включає вертикальне, паралельне друг стосовно друга, розташування системних плат, що відповідає ідеї “Blade” - серверів (рис. 2).

Використання стандартного блоку живлення (*БЖ*) під ATX дозволило зменшити розміри ПОК. Були проведені кілька тестів з різними БЖ, зокрема CHIEFTEC-400W, COOLERMASTER-420W, TARGA-400W. Всі вони з досить гарним запасом по потужності показали приблизно рівні показники. За конструктивними міркуваннями було обрано TARGA. Також був використаний мережений комутатор D-link Ethernet 100Mbit з 8 портами доступу (рис. 3).

 ISSN 1991-7848

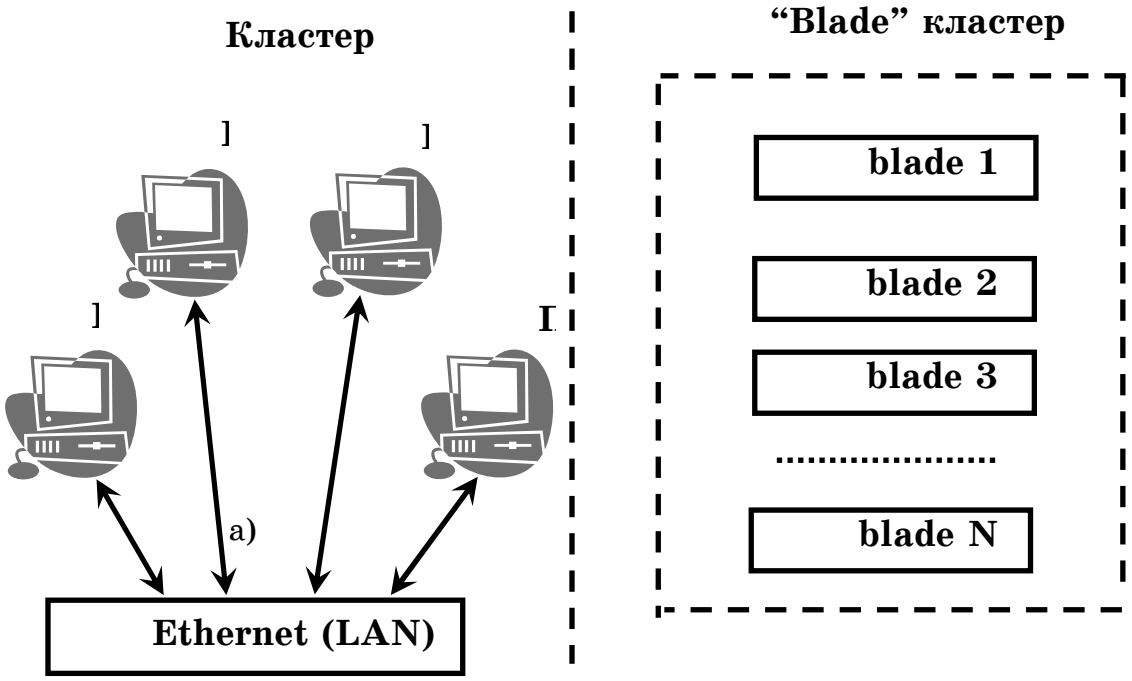

Рисунок 1 - Структурна схема кластерних обчислювальних комплексів: а) побудова кластерного обчислювального комплексу за допомогою з'єднання ПК через локальну мережу (*Ethernet*); б) «блейд» серверне рішення побудови персонального обчислювального кластера

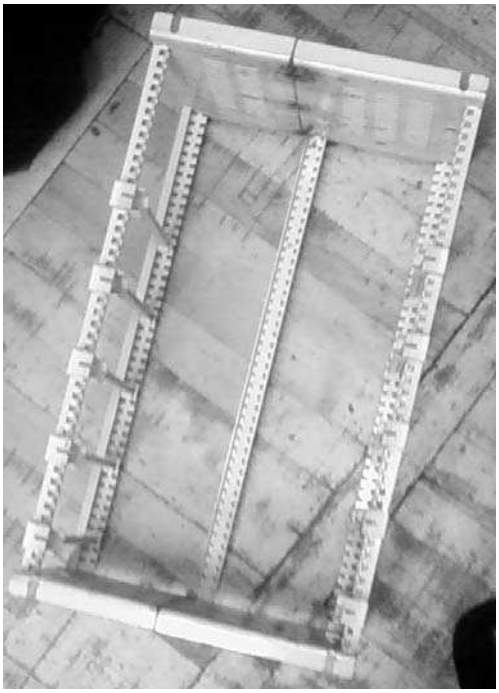

Рисунок 2 - Каркас корпуса «блейд» обчислювального ПОК

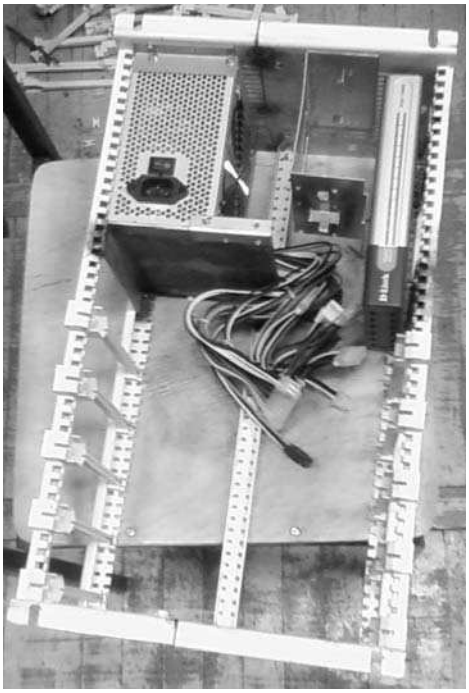

Рисунок 3 - Корпус ПОК з встановленим блоком живлення і мереженим комутатором

У конфігурацію сервера були обрані 5 лез із оглядом на можливу при необхідності розширюваність кластра додатковими лезами (рис. 4).

Блок обчислювальних вузлів персонального п'яти - вузлового кластера створений на основі використання системних плат PC2500 від VIA з інтег-

рованими процесорами С3-1,8 (рис.5), які мають характеристики, що наводяться в табл.1. Такі процесори мають суперекономічне ядро й досить низьку вартість. Основним критерієм вибору даної платформи послужила функція BIOS - «аварійний перезапуск живлення», що дозволяє без додаткової електроніки стартувати леза, а також функція віддаленого завантаження по мережі. На обраних платформах використовується пам'ять типу DDRII-533.

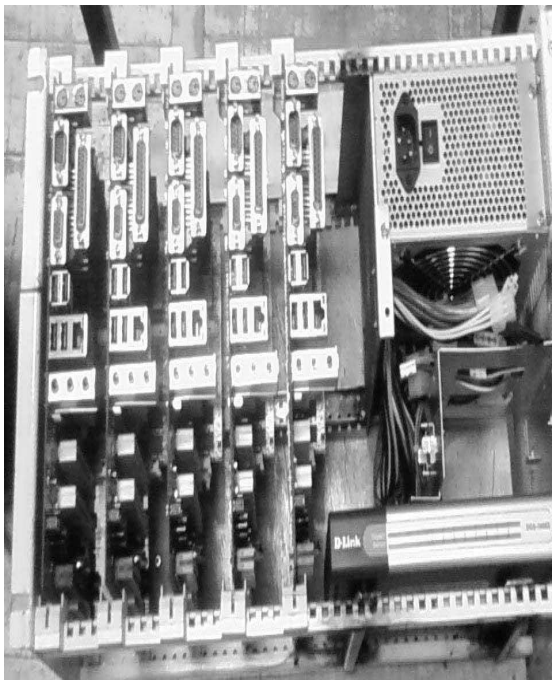

Рисунок 4 - Блок ПОК із встановленими лезами

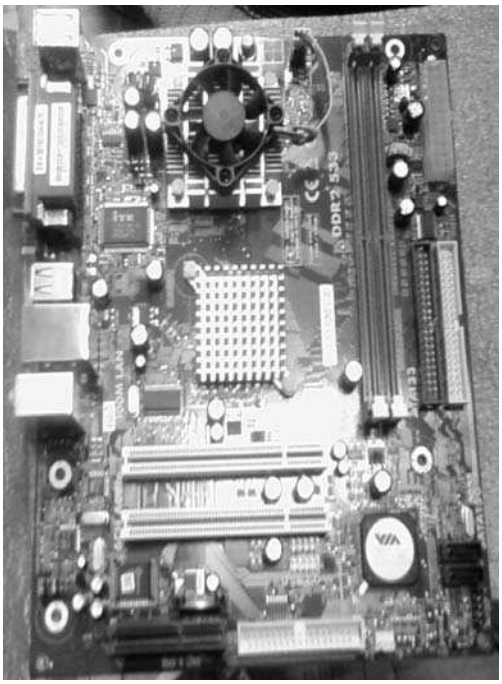

Рисунок 5 - Лезо кластера

За наведеними принципами був сконструйований ПОК, який представлено на рис. 6.

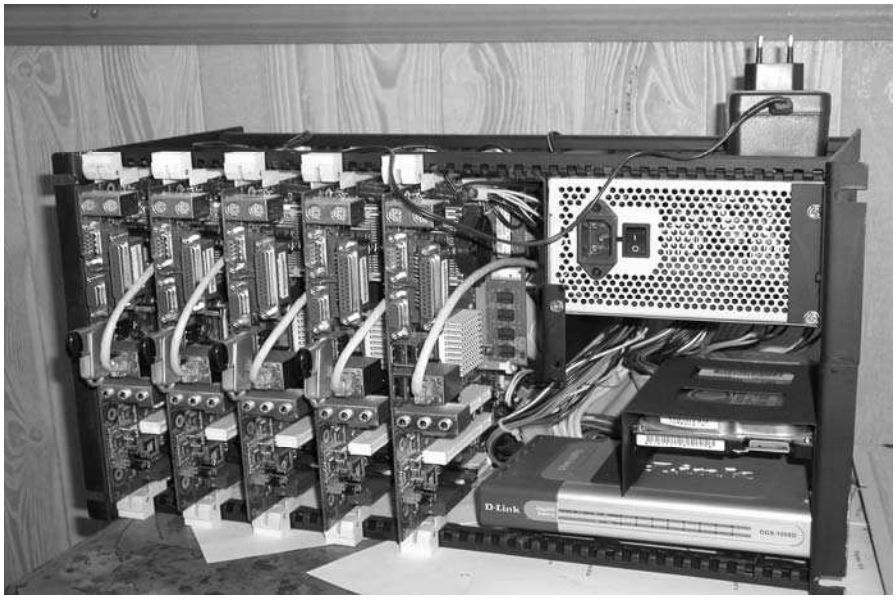

Рисунок 6 - ПОК в зборі

 ISSN 1991-7848

Персональний обчислювальний кластер, який зображено на рис.6, має розміри: ширина 19', висота 10,9', глибина 9'. Вага пристрою приблизно 7 кг.

Таблиця1

Технічні характеристики лез персонального обчислювального кластера

Процесор	VIA C7-D 1.5GHz
Чипсет	VIA CN700 + VT8237R Plus
Пам'ять	2 x DDRII slots, Up to 2GB
Вбудоване відео	VIA UniChrome Pro IGP(VIA CN700)
Аудіо	Realtek ALC655(support 6 channels)
Мережна плата	VIA VT6103L PHY 10/100 Base-T Ethernet
Порти	1x 10/100 LAN port 1x VGA connector 4x USB 2.0 ports 1x Line out / Line in / mic jack 2x PS2 connectors 1x Parallel Port (LPT port) 1x Serial Port (Com port)
Контролери	2x PATA connectors(Up to ATA133) 2x SATA connectors(Up to SATA150)
Гнізда на платі	2x USB 2.0 connectors (for 4 additional USB 2.0 ports) 1x Front-audio connector (Mic and Line Out) 1x Front-panel connector 2x PCI Slots 1x Floppy drive connector 1x CD Audio-in connector 2x Fan connectors: CPU/Sys FAN 1x ATX 20pin Power Connector 1x +12V 4pin Power Connector 1x CNR Slot 1x IR connector

Організація «блейд» кластера складається в об'єднанні лез через комутатор, який встановлюється у одному ж корпусі з лезами в єдину мережу Ethernet. Для блейд сервера досить одного вінчестера на якому розташований образ системи, що завантажується, і при цьому використовується механізм мережного завантаження «Network boot». При включенні живлення мережний комутатор роздає IP адреси всім вузлам кластера, при цьому відбувається початкова ініціалізація й кластер готовий до роботи (рис.7).

Після завантаження операційної системи доступ до ПОК можна отримати за стандартними мережними протоколми (*telnet, ssh, rsh*), як до звичайного ПК. Дякуючи чому для організації суперкомп'ютера на основі робочого ПК і ПОК необхідний лише мережний зв'язок між ними, який може бути організований як через локальну LAN так і через глобальну мережу internet (рис.8).

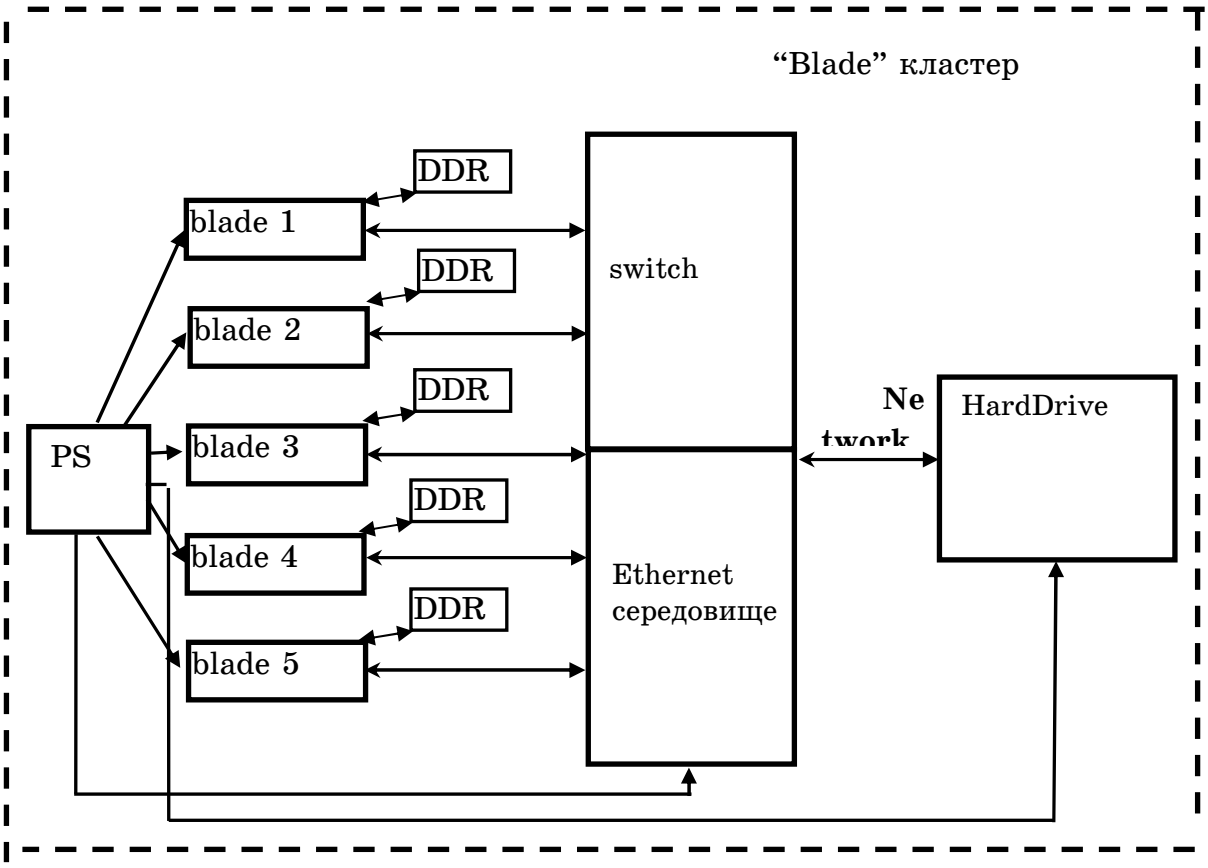

Рисунок 7 - Загальна блок-схема функціонування ПОК

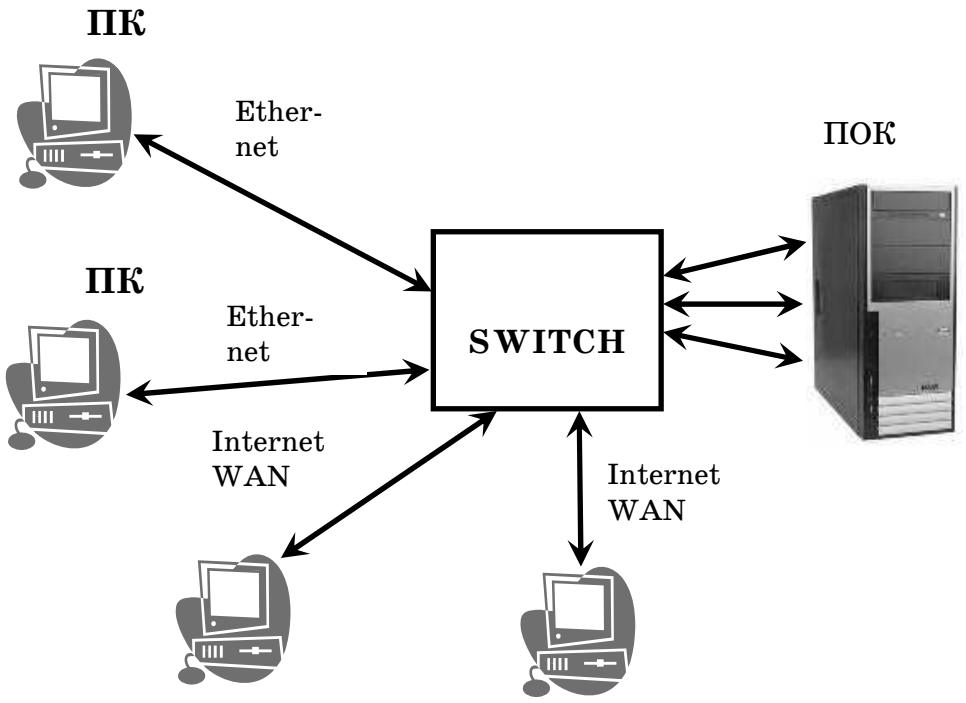

Рисунок 8 - Можливі режими зв’язку робочого ПК з ПОК

 ISSN 1991-7848

Системне програмне забезпечення ПОК

Виробники «блейд»- серверів надають список операційних систем, що підтримують роботу «блейд» кластера – зазвичай це Windows і Linux.

Після ретельного аналізу операційних систем, що поставляються з обраним лезом, було прийняте рішення використання операційної системи Fedora-6. Серед значних переваг якої є:

- безкоштовність - Fedora це Linux дистрибутив;
- економність до апаратних ресурсів і швидкодія - незаперечні переваги всіх Linux - систем у порівнянні з Windows;
- підтримка драйверів мережних карт лез - із запропонованих Linux - систем, Fedora-6 має необхідний набір драйверів у поставці.

Крім того, Fedora-6 має в комплекті інструментарій для моніторингу й аналізу паралельних обчислень на кластері, що безперечно буде корисно для насупних досліджень.

Як програмний інструментарій для паралельних обчислень був обраний стандарт MPI (*Message Passing Interface*) з відкритою базовою реалізацією mpich-1. У його склад входять, як правило, два обов'язкових компоненти:

- бібліотека програмування для мов C, C++ й Фортран,
- завантажник файлів до запуску.

Крім того, може бути присутнім довідкова система, командні файли для полегшення компіляції/компонування програм і т.д.

У стандарті MPI відсутнє все зайве, наприклад, немає засобів автоматичного перенесення й побудови копій файлу, що виконується, у мережі. У такому стандарті також немає ніяких засобів автоматичної декомпозиції, немає відладчика. Тобто це система міжпроцесового зв'язку в чистому виді, і не більше того.

Для MPI прийнято писати програму, що містить код всіх галузей відразу. MPI-завантажником запускається вказана кількість екземплярів програми. Кожний екземпляр визначає свій порядковий номер у запущеному колективі, і залежно від цього номера й розміру колективу виконує ту, або іншу гілку алгоритму. Така модель паралелізму називається Single program/Multiple data *(SPMD*), і є часткою моделі Multiple instruction/Multiple data (*MIMD*).

Кожна гілка має простір даних, повністю ізольоване від інших гілок. Обмінюються даними гілки тільки у вигляді повідомлень MPI.

Всі гілки запускаються завантажником одночасно, як процеси Юнікса. Кількість гілок фіксована, у ході роботи породження нових гілок неможливо.

Хоча з теоретичної точки зору для організації обміну даними досить усього двох операцій (прийом і передача), на практиці все є набагато складніше - для цього існує порядку 40 функцій.

Дві найпростіші (але й найбільш повільні з точки зору швидкодії) функції - MPI_Recv і MPI_Send. Але MPI - має досить розгалужений інструментарій функцій для обміну повідомленнями.

В MPI добре продумане об'єднання гілок у колективи. По суті, таке ділення служить тієї ж мети, що й введення ідентифікаторів для повідомлень: допомагає надійніше відрізняти повідомлення друг від друга. У більшості функцій MPI є параметр типу "коммунікатор", якому можна розглядати як номер колективу. Він обмежує область дії даної функції відповідним колективом. Коммунікатор колективу, що містить у собі всі гілки програми, створюється автоматично при старті й називається MPI_COMM_WORLD.

MPI повинен знати про типи переданих даних остільки-оскільки при роботі в мережах на різних ПК дані можуть мати різну розрядність (наприклад, тип int - 4 або 8 байт), орієнтацію (молодший байт розташовується в пам’яті першим на процесорах Intel, останнім - на всіх інших). Тому всі функції передачі-прийому в MPI оперують не кількістю переданих байт, а кількістю значень, тип яких задається параметром функції: MPI_INTEGER, MPI_REAL і т.д.

Однак, користуючись базовими функціями можна передавати або масиви, або одиночні дані (як окремий випадок масиву). А як передавати дані агрегатних типів, наприклад, структури? В MPI є механізм конструювання користувальницьких об'єктів на базі вже наявних (як користувальницьких, так і вбудованих).

Більше того, розробники MPI створили механізм конструювання нових типів навіть більше універсальний, чим наявний у мові програмування.

Виграш від використань механізму конструювання типів очевидний - краще один раз викликати функцію передачі-прийому зі складним шаблоном, чим двадцять разів - із простими.

Як інструментарій MPI сам по собі є засобом:

- складним: специфікація на MPI-1 містить 300 сторінок;
- спеціалізованим: це універсальна система зв'язку.

Особливості паралельних обчислень на ПОК

Для реалізації програмних обчислень на ПОК була використана мова системного програмування С и стандартна mpich-1 бібліотека.

При складанні програмних кодів було використано модульне програмування. Модульне програмування володіє тією перевагою, що кожна підпрограма може налагоджуватись окремо і це лежить в основі правильної декомпозиції.

Звичайна програма складається з розділів:

- розділ оголошень і співвідношень;
- розділ текстів процедур і функцій;
- розділ основного блоку програми.

 ISSN 1991-7848

При реалізації паралельного механізму обчислень закладено логіку розподілення в основний блок програми.

При запуску програми в результаті ініціалізації кожний mpich вузол одержує унікальний ідентифікатор (номером від 0 до 255). На основі цього ідентифікатора ґрунтується алгоритм розподілення (рис.9).

Нульовий процес (як правило це процес запущений на звичайній ЕОМ) виступає в ролі координатору обчислень master, організовуючи пересилання даних і завдань іншим slave процесам. При цьому нульовий процес сам не виконує ніяких обчислень.

Алгоритм роздачі завдань master ПК полягає в циклічному переборі доступних mpich процесів з роздачею даних зумовлених поточним станом обчислення, відсилання вказівки на обчислення, прийому результату й перемикання на наступний вузол.

При розв’язуванні багато вимірювальної задачі [3,4,5,12,18,30,32,33] в тілі програми виділяються три головних функції - прямого й зворотного прогону по координатам X, У и Z. Саме вони є мінімально доступним завданням на обчислення для кожного slave вузла.

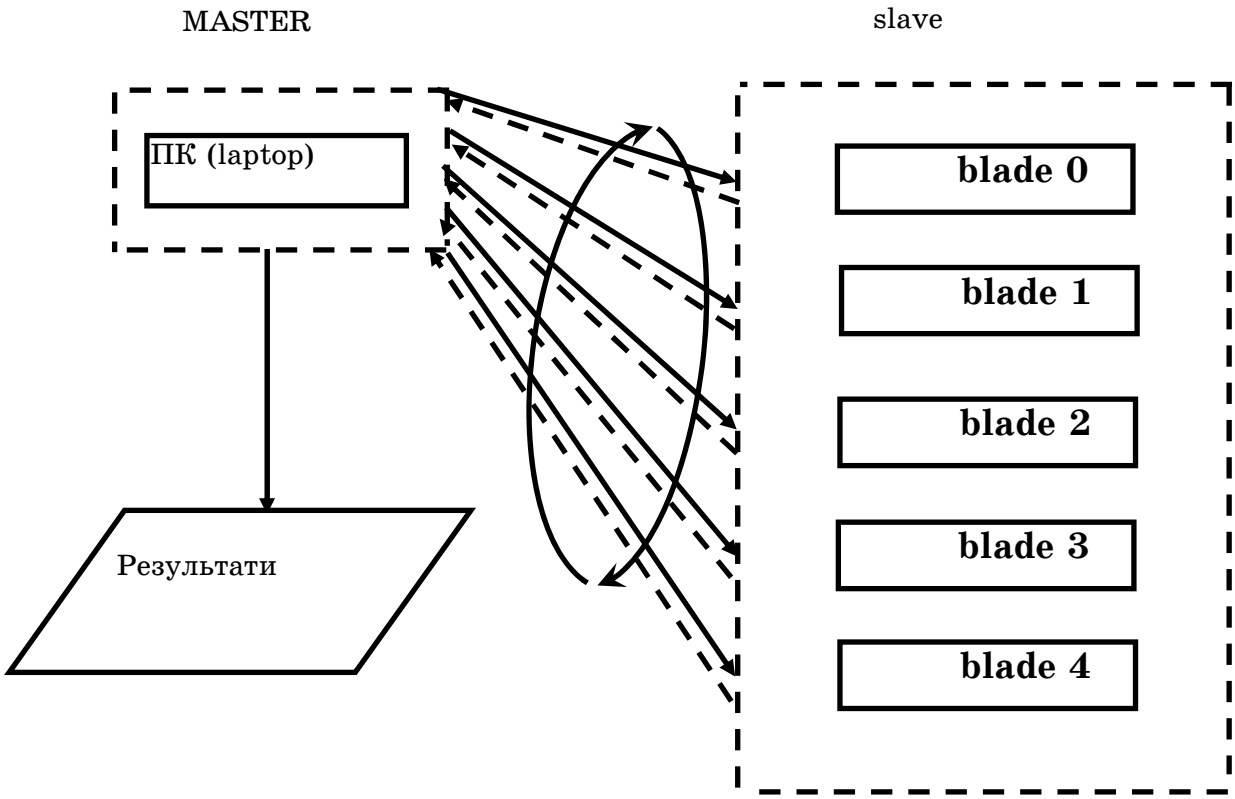

Рисунок 9 - Алгоритм паралельних обчислень ПОК

Для організації даного підходу введені три ідентифікатори обчислень і один додатковий для завершення відомих процесів:

```
#define ACTION_QUIT 0
#define ACTION_RYFP 1
#define ACTION_RXFP 2
#define ACTION_RZFP 3
```

На початку функціонування основного блоку виробляється ініціалізація MPI процесів засобами mpich бібліотеки й визначається ідентифікатор поточного процесу із числом зареєстрованих вузлів:

```
MPI_Init (&argc, &argv);
MPI_Comm_rank (MPI_COMM_WORLD, &myrank);
MPI_Comm_size (MPI_COMM_WORLD, &size);
```

Після ініціалізації в кожному вузлі відбувається первісна ініціалізація початкових умов і даних обчислень, провадиться підготовка вузла координатора й введених вузлів до початку роботи.

Після цього відбувається логічний поділ потоку обчислення на 2 галузі (*master й slave*):

```
if (myrank == 0) {
// код координатора обчислень - master
} else {
// код всіх ведених вузлів - slave
}
```

Код для всіх slave вузлів складається з нескінченного циклу в якому відбувається очікування повідомлення, що задається від master ПК. Як було відзначено вище - можливих завдань усього 4:

```
} else {
    while (1) {
MPI_Recv (&action_code, 1, MPI_INT, 0, 99, MPI_COMM_WORLD, &status);
if (action_code == ACTION_QUIT) {
    MPI_Finalize();
    return 1;
}
if ( (action_code == ACTION_RYFP) ||
(action_code == ACTION_RXFP) ||
(action_code == ACTION_RZFP) ) {
    // обчислення
}
}
}
```

При надходженні одного з кодів ACTION_RYFP або ACTION_RXFP або ACTION_RZF виконуються обчислення відповідного прогону й відправляються результати координаторові. А при надходженні коду ACTION_QUIT відбувається завершення роботи вузла й вихід із програми.

У той час поки в гілці slave вузлів проходить очікування спрямованих повідомлень, у master гілці керування обчислювальним процесом відбувається роздача завдань по циклі всім вузлам по черзі. І по завершенню всіх операцій головний вузол ініціює розсилання повідомлень на завершення всіх ведених процесів:

```
code = ACTION_QUIT;
```

```
for (i=1; i<size; i++) {
   MPI_Send (&code, 1, MPI_INT, i, 99, MPI_COMM_WORLD);
}
```

Після цього, master що залишився єдиним запущеним процесом, виводить результати обчислень і завершує свою роботу.

Алгоритмізація паралельних обчислень і обчислювальні експерименти

Особливості конструювання паралельних обчислювальних алгоритмів для ПОК, що розглядується, детально висвітлюється в [28,29]. Ефективність запропонованого підходу для проведення обчислювальних експериментів підтверджується рішенням задач нестаціонарної теплопроводності [3,5,12,17,18], деякими особливостями моделювання зворотних задач дослідження теплофізичних властивостей матеріалів [30,31], задач прогнозу екологічних систем під впливом природних і антропогенних чинників [4,32,33]. Деякі принципові особливості алгоритмізації і моделювання вказаного класу задач автор планує освітити в найближчих публікаціях.

Висновки і перспективи подальших досліджень

Основним науковим результатом представленої статті є розробка нових ефективних обчислювальних технологій кластерного типа для розв’язування багатомірних нестаціонарних задач металургійного виробництва. При цьому:

1. Запропонований, проаналізований і реалізований новий підхід для розв’язування багатомірних нестаціонарних задач металургійного виробництва на основі паралельних комп'ютерних технологій кластерного типа. Доведена універсальність його по відношенню до розв'язування широкого класу задач металургійного виробництва.
2. Запропонований персональний обчислювальний кластер MPP архітектури, як модульна багатопроцесорна система, що побудована на основі стандартних обчислювальних вузлів, з'єднаних високошвидкісним комунікаційним середовищем.
3. Запропонований кластерний обчислювальний комплекс побудований на основі «блейд» серверних рішень, при яких кілька однотипних материнських модулів установлюються в одному корпусі.
4. Для ефективного функціонування персонального обчислювального кластера було прийняте рішення використання операційної системи Fedora-6, яка має ряд значних переваг перед іншими. Як програмний інструментарій для паралельних обчислень був обраний стандарт MPI (*Message Passing Interface*) з відкритою базовою реалізацією mpich-1.
5. Розроблений і протестований високоефективний комплекс програм для розв’язування широкого класу задач металургійного виробництва.

ЛІТЕРАТУРА

1. Роуч П. Вычислительная гидродинамика / Пер. с англ. – М.: Мир, 1980. – 616 с.
2. Коздоба Л. А. Вычислительная теплофизика. – Киев: Наук. Думка, 1992. – 224 с.

3. Иващенко В.П., Швачич Г.Г., Шмукин А.А. Параллельные вычисления и прикладные задачи металлургической теплофизики // Системні технології. Регіональний збірник наукових праць. – Випуск 3(56).- Том 1.- Дніпропетровськ, 2008.- С. 123-138.
4. Швачич Г.Г. К вопросу конструирования параллельных вычислений при моделировании задач идентификации параметров окружающей среды // Математичне моделювання. №2(14), 2006. - С.23-34.
5. Швачич Г.Г. О параллельных компьютерных технологиях кластерного типа решения многомерных нестационарных задач // Materiбly IV mezinбrodnн vědecko – praktickб konference «Vědeckэ potencibl světa - 2007». – Dнl 7. Technickй vědy. Matematika. Fyzika. Modernн informačnн technologie. Vэstavba a architektura: Praha. Publishing House «Education and Science» s.r.o – P. 35-43.
6. Миленин А., Дыя Х., Мускальски З., Пилярчик Я. Моделирование с помощью метоа конечных элементов процесса волочения проволоки в роликовых волоках // Метизы 2(12), 2006. - С. 30-33.
7. Официальная страница проекта Beowulf - http://www.beowulf.org/.
8. Andrews G.R. Foundations of Multithreading, Parallel and Distributed Programming. Addison-Wesley, 2000 (русский перевод Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. - М.: Издательский дом "Вильяме", 2003).
9. Quinn M. J. Designing Efficient Algorithms for Parallel Computers. - McGraw-Hill, 1987.
10. Roosta, S.H. Parallel Processing and Parallel Algorithms: Theory and Computation. Springer-Verlag, NY. 2000.
11. Shvachych G.G. Prospects of construction highly-productive computers systems on the base of standard technologies // IV Intrenational Conference “Strategy of Quality in Indastry and Education”.- May 30 –June 6, 2008, Varna, Bulgaria . – Proceedings. Volume 2. – P. 815-819.
12. Швачич Г.Г., Шмукин А.А. О технологии параллельного компьютерного моделирования на многопроцессорных вычислительных комплексах кластерного типа // Математичне моделювання. №2(17), 2007. - С. 99-106.
13. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2001.
14. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. Parallel Programming in OpenMP. - Morgan Kaufinann Publishers, 2000.
15. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. Parallel Programming in OpenMP. - Morgan Kaufinann Publishers, 2000.
16. Dimitri P. Bertsekas, John N. Tsitsiklis. Parallel and Distributed Computation. Numerical Methods. - Prentice Hall, Englewood Cliffs, New Jersey, 1989.
17. Швачич Г.Г., Шмукин А.А. Особенности конструирования параллельных вычислительных алгоритмов для ПЭВМ в задачах тепло- и массообмена // Восточно-европейский журнал передовых технологий. №2, 2004. - С. 42-47.
18. Швачич Г.Г., Колпак В.П., Соболенко М.А. Математическое моделирование скоростных режимов термической обработки длинномерных изделий // Теория и практика металлургии. Общегосударственный научно-технический журнал. № 4-5(59-60). 2007. – С. 61-67.
19. Воеводин В.В. Математические модели и методы в параллельных процессах. – М.: Наука. 1986. – 29 с.
20. Miller R., Boxer L. A Unified Approach to Sequential and Parallel Algorithms. Prentice Hall, Upper Saddle River, NJ. 2000.

21. Швачич Г.Г. О алгебраическом подходе в концепции распределенного моделирования многомерных систем // Теория и практика металлургии. Общегосударственный научно-технический журнал. № 6(61). 2007. – С. 73-78.
22. Швачич Г.Г. Математическое моделирование одного класса задач металлургической теплофизики на основе многопроцессорных параллельных вычислительных систем // Математичне моделювання. №1(18), 2008. - С. 60-65.
23. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002. – 608 с.
24. Miller R., Boxer L. A Unified Approach to Sequential and Parallel Algorithms. Prentice Hall, Upper Saddle River, NJ. 2000.
25. Швачич Г.Г. Об одном подходе к решению проблемы латентности вычислительных кластеров MPP архитекруры // Материали за 5 – а международна научна практична конференция, «Ставайки съвременни наука»,- 2007. Том 10. Математика. Физика. Съвременни технологии на информации. Физическа култура и спорт. София. «Бял ГРАД-БГ» ООД.- С. 27-35.
26. Баканов В.М. Персональный вычислительный кластер как недостающее звено в технологии проведения сложных технологических расчетов // Метизы 2(12), 2006. - С. 33-36.
27. Shvachych G.G. Prospects of construction highly-productive computers systems on the base of standard technologies // IV Intrenational Conference “Strategy of Quality in Indastry and Education”.- May 30 –June 6, 2008, Varna, Bulgaria . – Proceedings. Volume 2. – P. 815-819.
28. Shvachych G.G., Shmukin A.A. Peculiarities of parallel computational algorithm synthesizing for personal electronic computer (pec) in heat – and – mass echange problems // Eastern-europeanjournal of enterprise technologies. Num.2, 2004, - P. 15-29.
29. Швачич Г.Г., Шмукин А.А. Графовые модели построения параллельных численных методов решения больших задач // Комп'ютерне моделювання та інформаційні технології в науці, економіці та освіті: Збірник наукових праць.- Кривий Ріг: КЕІ КНЕУ, 2005.- С.231-233.
30. Швачич Г.Г., Шмукин А.А. Определение теплофизических свойств материалов обратными методами // Материалы Международной научно-методической конференции "Проблемы математического моделирования". - Днепродзержинск, 2004.
31. Швачич Г.Г., Шмукин А.А., Протопопов Д.В. Определение теплофизических свойств материалов средствами математического моделирования // Материалы VШ Международной научно-практической конференции "Наука и образование 2005". Том 61. Техника.- Днепропетровск: Наука и образование, 2005. - С. 64-66.
32. Швачич Г.Г. Математическое моделирование динамики окружающей среды на основе применения параллельных вычислительных систем // Матеріали П Міжнародної науково-практичної конференції «Сучасні наукові дослідження-2006». Том 11. Математика. - Дніпропетровськ, 2006. - С. 61-66.
33. Швачич Г.Г. Моделирование задач идентификации окружающей среды средствами параллельного программирования // Матеріали 10 ювілейної міжнародної науково-методичної конференції "Проблеми математичного моделювання". Тези доповідей.- Дніпродзержинськ, 2006. - С. 230.