

**Використання обчислювального кластера з ОС Linux у вищому навчальному закладі**  
Сідун Н.М., Чичкарьов Є.А.

*Маріуполь, Донецька область, Україна, Приазовський державний технічний університет, email: influence@meta.ua*

The contribution is devoted to the usage of open source software in teaching process at technical university during the courses of parallel programming. The peculiarities are considered of implementation of cluster system on the base of teaching laboratory, equipped with personal computers.

В теперішній час виникає нагальна потреба у вирішенні актуальних завдань фундаментальної і прикладної науки, для аналізу і дослідження яких необхідні високопродуктивні системи, що дозволяють в найкоротші терміни проводити розрахунки і моделювання процесів. Перспективним напрямком є використання обчислювальних систем на основі кластерів, що представляють собою набір обчислювальних вузлів, пов'язаних високошвидкісної мережею і об'єднаних в логічне ціле спеціалізованим програмним забезпеченням.

В даній роботі проаналізовано можливості розгортання обчислювального кластера під управлінням ОС Linux, представлені оцінки швидкості обчислень при використанні різних компіляторів стосовно вирішення рівнянь з приватними похідними і деяких тестових завдань.

Обчислювальний кластер - це набір обчислювальних вузлів, призначених для вирішення розподілених завдань в паралельному режимі, з продуктивністю в сотні мільярдів операцій в секунду. Кожен обчислювальний вузол має свою оперативну пам'ять і працює під управлінням своєї операційної системи [1,2]. Програми, які виконуються на кластері, повинні бути адаптовані для використання в кластерному оточенні, для того щоб отримати переваги в продуктивності.

Багато з існуючих у світі обчислювальних кластерів працюють під управлінням операційних систем сімейства UNIX (у тому числі систем з відкритим кодом - Linux і FreeBSD). Серед них є як вільно поширювані системи, так і комерційні системи [1,2]. Використання UNIX-платформ може дати деякі переваги у вигляді оптимізації початкових витрат на придбання програмного забезпечення і реалізувати максимальну продуктивність. Крім цього, частина спеціалізованого програмного забезпечення для обчислювальних кластерів працює тільки на UNIX-платформах. Розробка програм для обчислювального кластеру здійснюється за допомогою вільно розповсюджуваних бібліотек і компіляторів, існують утиліти, що дозволяють забезпечити віддалений моніторинг завантаженості обчислювальних вузлів і доступ для копіювання даних і запуску виконуваних модулів. Основою кластера є не стільки операційна система, скільки комунікаційна середовище, що забезпечує можливість части-

нам паралельної програми, що виконується на різних комп'ютерах, ефективно взаємодіяти між собою. Відомі засоби для побудови кластера мають реалізації як для ОС сімейства UNIX (Linux, FreeBSD та ін.), так і для ОС Майкрософт [1].

Побудова кластера на базі Windows HPC Server 2008 або Windows Compute Cluster Server 2003 спрощує завдання його адміністрування (фактично від адміністратора потрібно тільки налаштувати керуючий вузол, забезпечивши його роботу в режимі контролера домена і запусивши необхідні служби). Дані рішення дозволяють забезпечити інтеграцію кластерного ПЗ з іншою інфраструктурою організації. Однак плюсом Linux в якості кластерної ОС є «прозорість» для користувача і системного адміністратора, що дозволяє швидше і простіше вирішувати всі виникаючі проблеми [1,2]. Оскільки бібліотеки обміну повідомленнями при реалізації паралельних обчислень є кросплатформеною, то вибір операційної системи (Windows vs Linux) не визначає працездатність кластера. Однак слід врахувати той факт, що Linux є помітно менш ресурсомісткою системою. Наприклад, при використанні PelicanHPC GNU Linux система займає в оперативній пам'яті не більш 40Мб (за результатами тесту). Вся інша пам'ять доступна паралельній програмі. Це дуже важливий фактор в тому випадку, коли кластер використовується з метою моделювання процесів на як можна більш докладній сітці. Очевидно, що при використанні багатопроцесорних систем каменем спотикання стає проблема так звано-го «розпаралелювання алгоритмів» для програмування на багато-процесорної структурі.

Для оцінки ефективності розпаралелювання, дослідження впливу числа використовуваних ядер на час вирішення, а також аналізу швидкодії програм, скомпільованих різними засобами, в якості тестової використовувалася крайова задача Діріхле для одновимірного рівняння теплопровідності, для вирішення якої використовувалися паралельний варіант явної різницевої схеми [1,3] і метод прямих з дискретизацією по просторовій координаті [4], що дозволяє звести задачу інтегрування диференціального рівняння з приватними похідними до задачі інтегрування системи звичайних диференціальних рівнянь (ОДУ). Для інтегрування системи ОДУ використовувалася паралельна реалізація методу Рунге-Кутта [2,5]. На думку [6], дана схема забезпечує високу обчислювальну ефективність. Програмна реалізація всіх необхідних алгоритмів виконана на Фортрані, С ++ і Octave (аналог Matlab з відкритим кодом). Компіляція та збирання програм виконувалася з використанням gfortran 4.5 та g++ або Intel ® Fortran Compiler 11 та icc. Кількість використовуваних в обчисленнях ядер змінювалося від 1 до 16 (при підключенні всіх вузлів). Обчислювальний експеримент виконувався на сітках змінної розмірності (кількість вузлів варіювалося на кілька порядків).

У результаті проведення обчислювального експерименту встановлено, що за інших рівних умов (конкретні значення граничних умов, кількість вузлів сітки, алгоритм вирішення задачі (1)) час обчислення сильно зале-

жить від способу оптимізації коду з урахуванням використовуваної архітектури процесора. При виконанні розрахунків час обчислення програми, скомпільованою Intel® Fortran Compiler, виявилось менше, ніж з використанням gfortran 4.5. Зниження часу рахунки за рахунок використання компілятора Intel склало від 5-10% до 100-110%, в залежності від способу оптимізації, обраному при компіляції за допомогою gfortran. Досліджуваний кластер був зібраний з 7 системних блоків ідентичною конфігурації (DualCore Intel Pentium, 1800 MHz cache 1 Mb; RAM: 1Gb DDR2-667; 1 HDD 250Gb SATA 2) і одного керуючого блоку (2 CPU Intel Core 2 Duo P8400, 2266 MHz cache 3 Mb; RAM: 4Gb DDR2-667; 1 HDD 320Gb SATA 2). Всі вузли кластера були об'єднані в локальну мережу на основі технології Gigabit Ethernet (фізичний тип з'єднання - «зірка»), яка забезпечує пропускну здатність 16 Gbit/s. Теоретична обчислювальна потужність кластера склала 851 GFlops, а кращий результат при використанні тестів Linpack склав 634 GFlops. Для окремих експериментів використовувалася та ж конфігурація кластера, але з використанням мережі Fast Ethernet. В якості операційної системи на всіх вузлах була встановлена ОС Linux Ubuntu 11.04 i386. Випробувана також робота кластера із завантаженням PelicanHPC GNU Linux. До складу системного програмного забезпечення був використаний пакет засобів для паралельного програмування Intel Cluster Toolkit Compiler Edition (компілятор Intel C++ та Fortran; бібліотека Intel MPI Library; інструменти Intel Trace Analyzer і Intel® Trace Collector; бібліотека Intel Math Kernel; еталонні тести Intel MPI), а також паралельний відлагоджувач Total View Debugger. Як технології паралельного програмування на даному кластері були доступні технології OpenMP і MPI (mpich-1.1.0, mpich2-1.2p1, openmpi-1.3). З інших засобів розробки встановлені GNU C++, GNU Fortran.

При варіюванні числом використовуваних обчислювальних вузлів встановлено, що найбільший приріст швидкості обчислень досягається при переході з обчислень на одному вузлі до обчислень на 2-3 вузлах. Подальше збільшення кількості обчислювальних вузлів і ядер дає істотно менший відносно зменшення часу рахунку. Оптимальне по приросту ефективності обчислень число використовуваних для розрахунку ядер процесорів залежить від швидкості міжпроцесорних комунікацій. З ростом пропускну здатності мережних з'єднань (перехід з Fast Ethernet на Gigabit Ethernet) оптимальна кількість обчислювальних ядер закономірно збільшилася. Тестування методів вирішення тієї ж задачі (рівняння теплопровідності з граничними умовами Діріхле) з використанням відкритої реалізації m-мови Matlab - пакета Octave - показала, що за рахунок векторизації і распараллеівання швидкість виконання інтерпретуються програм на Octave цілком порівнянна зі швидкістю виконання скомпільованих програм на C++ (навіть при використанні компілятора Intel). Проте швидкість обчислень для програм на m-мові виявилася досить чутливою до стилю програмування.

## **Література**

1. Сбитнев Ю.И. Кластеры / Ю.И. Сбитнев. – Екатеринбург, 2009. – 119 с.
2. Высокопроизводительные вычисления на кластерах: Учебн. пособие/ Под ред. А.В. Старченко. – Томск: Изд-во Том. ун-та, 2008. – 198 с.
3. Воеводин В. В. Параллельные вычисления / В. В. Воеводин - СПб.: БХВ - Петербург, 2002.- 608 с.
4. Самарский А.А. Введение в численные методы /А.А.Самарский. – Спб.:Лань, 2009. – 288 с.
5. Ващенко Г.В. Параллельная реализация явных методов Рунге-Кутты / Г.В. Ващенко, Е.А. Новиков. - Вестник Красноярского государственного аграрного университета. – 2010.- № 2. – С.14-18.
6. Швачич Г.Г. О концепции неограниченного параллелизма в задачах теплопроводности / Г.Г. Швачич, А.А. Шмукин // Восточно-Европейский журнал передовых технологий. - 4 (10). - 2004. – с. 132-135.

## **Новий підхід до створення INITRAMFS**

*Сітало В.С., Ясько М.М.*

*Дніпропетровський національний університет імені Олеся Гончара, факультет прикладної математики, [root@stvad.org](mailto:root@stvad.org)*

Considered the principles of operation, uses and benefits of live systems. Proposed a new approach to the construction of the initial file system and described the benefits of this approach. Described the steps necessary to implement the approach. Approach demonstrated on the example of Puppy Linux.

Одним із засобів ширшого розповсюдження вільного програмного забезпечення, а конкретніше для ширшого ознайомлення користувачів з операційними системами GNU/Linux, є використання LiveCD. LiveCD – це операційна система, яка завантажується зі змінного носія (CD, DVD, USB-накопичувача і т.п.) та не вимагає для свого функціонування встановлення на жорсткий диск. Цим же поняттям позначають і носії з такими ОС (в залежності від носія розрізняють LiveCD, LiveDVD і LiveUSB).

LiveCD дозволяють швидко почати роботу з комп'ютером, уникаючи тривалого процесу установки ОС в сталу пам'ять (вінчестер). Запуск LiveCD звичайно триває декілька хвилин, в той час як перший запуск (встановлення) традиційних ОС часто вимагає від однієї до декількох годин.

Першим LiveCD, який базувався на Linux, був Yggdrasil Linux, вперше виданий у 1992р. На більш пізньому етапі розвитку з'явилась можливість