УДК 004.93

АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ПОБУДОВИ ВИСОКОПРОДУКТИВНОЇ КЛАСТЕРНОЇ СИСТЕМИ

Яровий Андрій, Арсенюк Ігор, Беліченко Сергій

Вінницький національний технічний університет

Анотація

Здійснено аналіз способів розпаралелювання у математичних задачах. За його результатами була підтверджена доцільність та перспективність застосування паралельних обчислень. Наведено використання технологій для паралельних обчислень, таких як MPI (Massive Parallel Interface) та PVM (Parallel Virtual Machine).

Abstract

The parallelization methods analysis of mathematical problems has been done. According to the results confirmed the feasibility and perspective of parallel computing. The usage of technology for parallel computing such as MPI (Massive Parallel Interface) and PVM (Parallel Virtual Machine) is shown.

Вступ

Робота присвячена аналізу програмних засобів організації паралельних та розподілених обчислень. В ній проілюстровано, як можна використати технології для паралельних та розподілених обчислень, такі як PVM (Parallel Virtual Machine) та MPI (Massive Parallel Interface) у кластерних системах. Такий підхід дозволить покращити продуктивність кластерних систем завдяки більш ефективному використанню обчислювальних ресурсів.

Аналіз програмних засобів організації паралельних та розподілених обчислень

Кластер – група комп'ютерів, об'єднаних високошвидкісними каналами зв'язку, яка представляє з точки зору користувача єдиний апаратний ресурс. PVM (Parallel Virtual Machine) – загальнодоступний програмний пакет, який дозволяє об'єднувати різнорідний набір комп'ютерів у загальний обчислювальний ресурс (паралельну віртуальну машину) та надає можливості управління процесами завдяки механізму передачі повідомлень. Існують реалізації PVM для різноманітних платформ: від лептопів (ноутбуків) до суперкомп'ютерів. PVM має більш розширені можливості, ніж його популярний аналог MPI, в плані контролю обчислень. Тут присутня спеціалізована консоль управління паралельною системою та її графічний еквівалент XPVM, що дозволяє наочно продемонструвати роботу усієї системи [1 – 3].

PVM був розроблений в Окріджській національній лабораторії, університету штату Теннесі та університети Еморі. Робота над проектом розпочалась в Окріджській національній лабораторії влітку 1989 року, і в тому ж році була випущена PVM 1.0. Останньою версією PVM є версія 3.4.6, яку було випущено в лютому 2009 року. PVM підтримує програмування на мовах C та C++ шляхом представлення спеціальних бібліотек. PVM є безкоштовним ПЗ та поширюються в рамках ліцензії BSD License [1 – 3].

При реалізації кластерної системи із використанням технології PVM для зменшення витрат часу, пов'язаних із затримкою (латентністю), потрібно дотримуватись таких системних вимог [1, 2]:

– створювати алгоритми, що потребують менше пересилань даних, а також групувати запити та відповіді;

– використовувати інформацію, розташовану відносно близько у гіпермережі;

– кешувати, запитувати відповідь і дублювати інформацію;

– переміщувати дані на обчислювальний вузол, де, власне і виконуються обчислення;

– по можливості, виконувати обчислення там, де зберігаються дані (пов'язано з безпекою та використанням приватних ресурсів та сервісів).

MPI (Message Passing Interface) – програмний інтерфейс (API) для передавання інформації, який дозволяє обмінюватись повідомленнями між процесами, що виконують одну задачу. MPI є найбільш поширеним стандартом інтерфейсу обміну даними у паралельному програмуванні, існують його реалізації для різних комп'ютерних платформ. Використовується при розробці програм для кластерів та суперкомп'ютерів. Основним засобом комунікації між процесами в MPI є передача повідомлень один одному [4, 5].

Стандартизацією MPI займається MPI форум. У стандарті MPI описаний інтерфейс передачі повідомлень, який повинен підтримуватись як на платформі, так і в додатках користувача. Натепер існує багато безкоштовних та комерційних реалізацій MPI. Існують реалізації для мов Java, C та C++ [6].

Базовим механізмом зв'язку між MPI процесами є передача та прийняття повідомлень. Повідомлення містить в собі дані та інформацію, які дозволяють стороні отримувача виконувати їх вибірковий прийом.

Операції прийому та передачі можуть бути як такими, що блокуються, так і такими, що не блокуються. Для операцій, що не блокуються, визначені функції перевірки готовності та очікування виконання.

В першу чергу MPI орієнтований на системи з розподіленою пам'яттю, тобто коли витрати на передавання даних великі, в той час як OpenMP орієнтований на системи з загальною пам'яттю (багатоядерні із загальним кешем). Обидві технології можуть використовуватись спільно, для того, щоб оптимально використати у кластері багатоядерні системи [7].

Іншим способом зв'язку є віддалений доступ до пам'яті RMA (Remote Memory Access), який дозволяє читати та змінювати область пам'яті віддаленого процесора. Локальний процес може переносити область пам'яті віддаленого процесу (усередині вказаного процесами вікна) у свою пам'ять та назад, а також комбінувати дані, які передаються у віддалений процес з наявними в його пам'яті даними (наприклад, шляхом додавання). Усі операції віддаленого доступу звертаються до пам'яті, яка не блокується, однак, при їх виконанні необхідно викликати функції синхронізації [8].

Результати досліджень

Під час проведення експериментальних досліджень було побудовано обчислювальний кластер, що містив як стаціонарний комп'ютер, так і мобільні комп'ютерні пристрої.

В якості тестової задачі на обчислювальному кластері було реалізовано метод Монте-Карло, зокрема для обрахування значення числа π. Використання кластерної системи дозволило отримати приріст швидкодії на 13% порівняно зі стаціонарним комп'ютером та на 45% швидше порівняно із ноутбуком. Графічна ілюстрація методу наведена на рисунку 1 [9].

Висновки

Отримані результати показують доцільність та перспективність застосування технологій паралельних та розподілених обчислень, зокрема, MPI та PVM для розв'язання складних математичних задач [10, 11].

У перспективі планується використати отримані результати у подальших дослідженнях з метою створення високопродуктивного обчислювального кластеру на основі GPGPU – технологій [8].

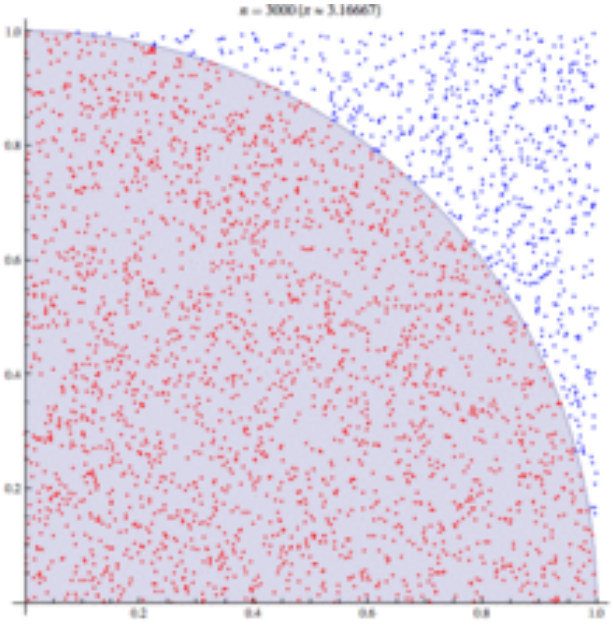

Рисунок 1 – Результати реалізації методу Монте-Карло
в межах обчислювального кластера

Список використаних джерел:

1. Хьюз К. Параллельное и распределенное программирование на C++. / К. Хьюз, Т. Хьюз [Пер. с англ.] – М.: ИД «Вильямс», 2004. – 672 с.

2. Аксак Н. Г. Паралельні та розподілені обчислення : Підручник / Н. Г. Аксак, О. Г. Руденко, А. М. Гуржій. – Х. : Компанія СМІТ, 2009. – 480 с.

3. Воеводин Вл. В., Жуматий С. А. // Вычислительное дело и кластерные системы / Вл. В. Воеводин, С. А. Жуматий – М.: МГУ, 2007. – 150 с.

4. Эхтер Ш. Многоядерное программирование / Ш. Эхтер, Д. Робертс – СПб.: "Питер", 2010. – 316 с.

5. Биллиг В. А. Параллельные вычисления и многопоточное программирование / В. А. Биллиг – М.: НОУ "Интуит", 2016. – 310 с.

6. Message Passing Interface [Електронний ресурс] : [веб-сайт]. – режим доступу: http://mpi-forum.org/ (дата звернення 01.07.2016). – назва з екрана.

7. Яровий А. А. Паралельно-ієрархічне перетворення прямоподібних зображень на основі Multi-GPU систем / А. А. Яровий, О. О. Кулик, Н. І. Кокряцька // Інформаційні технології та комп'ютерна інженерія. – 2015. – № 3. – С. 72 – 80.

8. Яровий А. А. Паралельно-ієрархічне перетворення інформаційних середовищ на основі гетерогенної кластерної системи // Вісник ВПІ. – 2011. – № 2 (95). – С. 120 – 127.

9. Беліченко С. О. Аналіз підвищення продуктивності обчислень на основі застосування кластерної системи [Електронний ресурс] : [веб-сайт]. – режим доступу: http://ir.lib.vntu.edu.ua/handle/123456789/10852 (дата звернення 14.06.2016). – назва з екрана.

10. Яровий А. А. Методологічні особливості побудови паралельно-ієрархічних та ієрарх-ієрархічних мереж на основі кластерних систем з розподіленою обробкою інформації // А. А. Яровий, Оптико-електронні інформаційно-енергетичні технології. – 2010. – №1 (19). – С. 69 – 79.

11. Яровой А. А. Параллельно-иерархические сети на основе кластерной CPU-ориентированной аппаратной платформы А. А. Яровой, Л. И. Тимченко, С. В. Наконечная // Современный научный вестник. Серия: Современные информационные технологии. – 2014. – № 8 (204) – С. 50 – 56.