УДК 004.8(07)

РОЗПОДІЛЕНІ СИСТЕМИ НА АВТОМОБІЛЬНОМУ ТРАНСПОРТІ

В.О. Алексієв, доцент, к.т.н., ст. наук. співр., Є.П. Логачов, студент, ХНАДУ

Анотація. *Розглянуто використання розподілених систем у автомобільному транспорті для вирішення високопродуктивних обчислювальних завдань наукових досліджень транспортної галузі.*

Ключові слова: *розподілені системи, телематика, автомобіль.*

РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ НА АВТОМОБИЛЬНОМ ТРАНСПОРТЕ.

В.О. Алексеев, доцент, к.т.н., ст. научн. сотр., Е.П. Логачёв, студент, ХНАДУ

Аннотация. *Рассмотрено использование распределенных систем в автомобильном транспорте для решения высокопроизводительных вычислительных задач научных исследований транспортной отрасли.*

Ключевые слова: *распределенные системы, телематика, автомобиль.*

DISTRIBUTED SYSTEMS FOR THE ROAD TRANSPORT

V. Aleksiev, associate professor, cand. eng. sc., senior research worker,
E. Logachev, student, KhNADU

Abstract. *The use of distributed systems in road transport solutions for high performance computing research needs of the transport industry is considered in the given article.*

Key words: *distributed systems, telematics, vehicle.*

Вступ

На сучасному рівні розвитку автомобільної галузі спостерігається застосування новітніх технічних технологій, що поєднано з інтелектуалізацією управління рухом та автоматизацією усіх автомобільних вузлів та агрегатів. Ці новітні технології особливо у галузі управління рухом супроводжуються накопиченням великої кількості різноманітної інформації, яку подальше необхідно обробити, а у деяких випадках обробка повинна виконуватися у реальному часі.

Постановка задачі

Без використання потужних обчислювальних ресурсів обробка такої інформації неможлива у малі проміжки часу. Використання звичайних персональних ЕОМ у таких задачах неможлива, з причини недостатньої їхньої продуктивності. Нераціональність використання суперкомп'ютерів спричинена на сьогодні ще високою ціною на них. В даній статті розглянуто, як можна досягти продуктивності суперкомп'ютерів, використовуючи звичайні персональні ЕОМ.

Огляд джерел дослідження

Загальну інформацію застосування розподілених систем, зокрема GRID-технологій, наведено в [1, 2]. В [3] надано аналогію розвитку комп'ютерних ресурсів з розвитком біологічних популяцій, запропоновано методику розрахунку. Теорія та практика програ-

мування паралельних програм з використанням технологій MPI та OpenMP наведено у [4, 5].

Теоретичне обґрунтування

На сьогодні найбільш актуальним способом створення потужних багатопроцесорних комплексів є кластеризація персональних комп'ютерів, що масово випускаються, за допомогою серійного телекомунікаційного обладнання локальних мереж (або більш високопродуктивних з'єднань).

Кластер – це набір комп'ютерів (обчислювальних вузлів), об'єднаних деякою комунікаційною мережею. Кожен обчислювальний вузол має свою оперативну пам'ять і працює під управлінням своєї операційної системи. Найбільш поширеним є використання однорідних кластерів, тобто таких, де всі вузли абсолютно однакові за своєю архітектурою і продуктивністю. Кластерні системи, з огляду на те, що будуються на основі серійних компонентів, характеризуються невисокою вартістю, а для досягнення прийнятного рівня продуктивності користуються зазвичай їх гарною масштабованістю, нарощуючи кількість процесорів. До речі, звичайні локальні обчислювальні мережі можна вважати єдиним комп'ютером, звичайно ж при використанні спеціального програмного забезпечення.

Розпаралелювання – це процес розбиття крупніших завдань, що виконуються додатком, на підзадачі, які обробляються одночасно, але незалежно один від одного. Кожну з таких підзадач можна призначити окремому процесору, а результати їх виконання за необхідністю об'єднати (синхронізувати) для отримання узгодженого результату. Оскільки декілька підзадач виконуються одночасно, час на отримання загального результату скорочується. У цьому і полягає основна перевага розподілених систем.

Як вже було сказано вище, використання новітніх технологій на автомобільному транспорті супроводжується накопиченням великої кількості різноманітної інформації. Однією з найбільш зручних форм представлення інформації є зображення. На сьогодні день розпізнавання образів у зображеннях використовується в таких галузях, як медицина, космонавтика, дактилоскопія і т. ін., продовжує впроваджуватися у багато сфер життя людини. Як приклад використання методів виділення і розпізнавання таких об'єктів можна привести розпізнавання номерів автівок на автостоянках.

Обробка зображень завжди було ресурсномісткою задачею для обчислювальних систем. Наприклад, застосування до зображення фільтра може зайняти хвилини або навіть години, залежно від комплексності фільтра, розміру зображення і швидкості комп'ютера. З точки зору розподілених систем, застосування паралельного програмування при обробці зображень дає ефективну продуктивність. Це відбувається за рахунок того, що алгоритми обробки зображень дуже добре розпаралелюються.

Програмна реалізація

Процес розбиття задачі на малі частини називається декомпозицією. Розглянемо, як це робиться на практиці. Наприклад, маємо зображення, яке необхідно обробити (рис. 1, а). Розіб'ємо це зображення на малі частини (рис. 1, б), і кожну частину розподілимо на різні комп'ютери. Далі вже кожний комп'ютер одночасно буде обробляти свою маленьку частину, а це у свою чергу скоротить час на обробку.

Одним з основних стандартів програмування кластерних систем вважається MPI. Реалізація під різні операційні системи цього стандарту виконано у пакеті MPICH.

На початку кожної MPI програми викликаються деякі функції. Ці функції ініціалізують процес обміну повідомлень між комп'ютерами та визначають номер кожного вузла, який однозначно ідентифікує вузол серед інших вузлів (комп'ютерів). Цей номер може приймати значення від 0 до «кількості вузлів в кластері» – 1. Як правило вузол з номером 0 звичайно називають головним вузлом, і він керує усім процесом. Приклад загальної структури програми на MPI, яка розподіляє частини зображення на різних комп'ютерах, відповідно до їх номеру наведено нижче.

```
#include <mpi.h>
#include <stdlib.h>

void main()
{
// Змінні для номера процесу та
```

```
// загальної кількості процесів
long int myid, numprocs;
int imgHeight, imgWidth, imgSize;
// Ініціалізація MPI
MPI_Init((void *) 0, (void *) 0);
// Отримуємо номер поточного вузла
MPI_Comm_rank(MPI_COMM_WORLD,
  &myid);
// Отримуємо загальну кількість вузлів
MPI_Comm_size(MPI_COMM_WORLD,
  &numprocs);
// Десь тут отримуємо дані довжини
// та висоти зображення з файла
// та записуємо у змінні
// imgHeight та imgWidth
// imgSize – величина зображення,
// яку буде обробляти кожний вузол
imgSize = imgHeight / numprocs;
// Приклад процесу читання даних
// зображення з файла
for(int i=myid*imgSize;
i<myid*imgSize + imgSize;++i)
{
for(int j=0;j<imgWidth;++j)
{
// Читаємо дані з файла
}
}
// … Тут продовжується тіло програми
// Закінчуємо роботу з MPI
MPI_Finalize(MPI_COMM_WORLD);
}
```

а

б

Рис. 1. а – оригінальне зображення; б – розбиття зображення на маленькі частини

Як бачимо з тексту програми, кожний процес (комп'ютер) буде отримувати свою частину даних зображення. Така реалізація буде ефективною, якщо необхідно обробляти зображення великих розмірів. Наприклад, зображення роздільної здатності 20000×20000 пікселів може потребувати 1200000000 байтів пам'яті! А це дуже багато для більшості комп'ютерів. Вищенаведений приклад показує як просто, використовуючи принцип декомпозиції, можна зменшити вимоги до пам'яті комп'ютера.

Подальшу продуктивність роботи такої програми можна отримати, завдяки додаванню у програму коду, що враховує багатопроцесорність або багатоядерність комп'ютерів, на яких буде працювати програмний додаток. Якщо цього не робити, то програма буде використовувати тільки один процесор багатопроцесорної системи (або одне ядро – багатоядерної).

Одним з найпоширеніших стандартів програмування багатопроцесорних систем з загальною пам'яттю є OpenMP. Головне завдання OpenMP – це полегшення написання програм, орієнтованих на цикли. Використання цих двох технологій одночасно надає більшої продуктивності розподіленій системі, а це у свою чергу дозволить більш ефективно виконувати обчислення високопродуктивних задач.

Висновок

На сьогодні паралельне програмування для розподілених систем набуває популярності. Вже минули часи, коли процесори комп'ютерів були одноядерними, а збільшення їхньої продуктивності відбувалося за рахунок підвищення тактової частоти. Появлення багатопроцесорних та багатоядерних систем вимагає від розробника отримання нових знань щодо адаптації програмного забезпечення під нові процесори.

Створення обчислювальних кластерів – це ще один крок на шляху отримання недорогих продуктивних обчислювальних ресурсів, особливо це стосується організацій, які вже мають деякий парк комп'ютерних ресурсів. В цій статті зображено, як просто програмувати додатки для таких систем.

Література

1. Алексієв В.О. Концепція застосування GRID-технологій на транспорті: Научно-технический журнал «Бионика интеллекта». – Харьков: ХНУРЕ. – №2(69) – 2008 – С. 125–128.
2. Алексієв В.О. Новітня GRID-технологія для вирішення задач дослідження мехатронних систем у автомобільно-дорож–ньому ВНЗ // Вестник ХНАДУ: сб. науч. тр. – Харьков: ХНАДУ – 2007. – Вып. 38. – С. 111–113.
3. Алексієв В.О., Логачов Є.П. Розвиток гетерогенних комп'ютерних мереж у ВНЗ транспортного профілю // Автомобільний транспорт: збірник наукових робіт – №23. – 2009. – С. 159–162.
4. Антонов А.С. Введение в параллельные вычисления. – М.: Изд-во МГУ, 2002. – С. 29–55.
5. Антонов А.С. Параллельное программирование с использованием технологии OpenMP. – М.: Изд-во МГУ, 2009. – С. 27–88.

Рецензент: О.В. Бажинов, професор, д.т.н., ХНАДУ.

Стаття надійшла до редакції 16 вересня 2009 р.