13.1 ІСТОРІЯ БАГАТОПРОЦЕСОРНОЇ ОБРОБКИ. ФОРМИ ПАРАЛЕЛІЗМУ

Поняття про процесор як про найцінніший ресурс обчислювальної системи, звичайне для перших двох десятиліть розвитку сучасної обчислювальної техніки, зараз вже практично застаріло. Зараз з появою мультипроцесорних архітектур набагато більшого значення набувають проблеми надійності, паралелізму в обчисленнях, оптимальних схем комутації та змагань між процесорами, що намагаються отримати доступ до одних і тих же ресурсів.

З перших днів свого існування комп'ютерна промисловість постійно прагнула до досягнення все більшої і більшої обчислювальної потужності. Наприклад, комп'ютер ENIAC міг виконувати 300 операцій в секунду, з легкістю тисячократно обставляючи будь-який попередній йому калькулятор, але людей і це не влаштовувало. Швидкодія сучасних машин в мільйони разів перевищує можливості ENIAC, але є потреби в ще більшій потужності.

З моменту появи фактично діючих електронно-обчислювальних машин вчені і практики в галузі інформатики працювали над тим, щоб ці машини можна було об'єднувати в так звані багатомашинні комплекси. Справедливо передбачалося, що такі комплекси будуть набагато продуктивніше і ефективніше, ніж кожна окрема обчислювальна машина. При цьому розглядалися різні можливі варіанти об'єднання комп'ютерів між собою.

Для розв'язання великих завдань потрібні все більш швидкі комп'ютери. Є всього два основних способи підвищення швидкодії ЕОМ [38]:

1. За рахунок підвищення швидкодії елементної бази (тактової частоти). Швидкодія процесора зростає пропорційно зростанню тактової частоти, при цьому не потрібно зміни системи програмування і програм користувача.

2. За рахунок збільшення числа одночасно працюючих в одному ЕОМ для розв'язання одного завдання, процесорів тощо, тобто за рахунок паралелізму виконання операцій. Це вимагає використання складних систем паралельного програмування.

Паралельні системи по архітектурі поділяються на два класи:

1. Конвеєрні системи, коли кілька спеціалізованих блоків одночасно працюють над частинами одного потоку команд.

2. Паралельні системи, коли певна кількість команд однієї програми одночасно виконуються множиною АЛУ або процесорів.

Тактова частота. У минулому проблема обчислювальної потужності завжди вирішувалася за рахунок підвищення тактової частоти. На жаль, цьому підвищенню вже починає перешкоджати ряд фундаментальних обмежень. Так як електричний сигнал не може поширюватися швидше за швидкість світла, яка дорівнює у вакуумі приблизно 30 см/нс, а в мідному провіднику або в оптичному кабелі швидкість поширення сигналу дорівнює приблизно 20 см/нс. З цього випливає, що на комп'ютері з тактовою частотою 10 ГГц сигнал не може подолати за один такт сумарну відстань, що перевищує 2 см. Для комп'ютера з тактовою частотою 100 ГГц максимальна сумарна довжина шляху дорівнює 2 мм. Комп'ютер з тактовою частотою 1 ТГц (1000 ГГц) повинен бути менше 100 мкм (0,1 мм), щоб сигнал міг дістатися з одного його кінця до іншого за один такт [9].

Зменшити комп'ютери до таких розмірів, може бути, і можливо, але тоді на заваді стане інша фундаментальна проблема: відвід тепла. Чим швидше комп'ютер, тим більше тепла він виділяє, а чим він менший, тим важче від цього тепла позбутися. Вже зараз на потужних x86-системах системи охолодження, встановлені на процесорі, більше самого процесора.

Конвеєрні системи. Історично ідея конвеєрної обробки стосовно до обчислювальної техніки першою отримала теоретичне обґрунтування і практичне втілення в реальній апаратурі. Конвеєр – найбільш «дешевий» спосіб підвищення продуктивності за рахунок введення паралелізму.

Ідея конвеєрної обробки полягає у виділенні окремих етапів виконання спільної операції, причому кожен етап, виконавши свою роботу, передавав би результат наступного, одночасно приймаючи нову порцію вхідних даних. Отримуємо очевидний виграш в швидкості обробки за рахунок поєднання раніше рознесених в часі операцій.

Припустимо, що в операції можна виділити п'ять мікрооперацій, кожна з яких виконується за одну одиницю часу. Якщо кожну мікрооперацію виділити в окремий етап (або інакше кажуть – ступінь) конвеєрного пристрою, то на п'ятій одиниці часу на різній стадії обробки такого пристрою будуть знаходитися перші п'ять пар аргументів, а весь набір зі ста пар буде оброблений за 5+99 = 104 одиниці часу. В

ідеальному випадку прискорення в порівнянні з послідовним пристроєм зростає в п’ять разів (по числу ступенів конвеєра) [38]. Конвеєрні системи втрачають сенс, коли час передачі інформації з рівня на рівень стає порівнянним з часом обчислень на кожному ступені.

Паралельні системи. З моменту появи фактично діючих електронно-обчислювальних машин вчені і практики в галузі інформатики працювали над тим, щоб ці машини можна було об’єднувати в так звані багатомашинні комплекси. Передбачалося, що такі комплекси будуть набагато продуктивніше і ефективніше, ніж кожна окрема обчислювальна машина, а головне – набагато відмовостійкими. Очевидно, що виконувати вимоги відмовостійкості обчислювальній системі при єдиному процесорі неможливо, тому всі відмовостійкі системи, незалежно від реалізацій і архітектурних рішень, є багатопроцесорними.

Один з підходів до збільшення швидкості полягає в широкомасштабному застосуванні паралельних обчислювальних систем. Ці системи містять багато центральних процесорів, кожний з яких працює на звичайній частоті (яке б значення вона не мала в даний час), але в порівнянні з окремо взятим процесором всі разом вони мають куди більш високу обчислювальну потужність. Зараз вже продаються системи, що складаються з десятків тисяч центральних процесорів. А в лабораторіях вже створені системи з 1 млн центральних процесорів [9].

Неважко буде поставити в одній дуже великій кімнаті тисячу не пов’язаних між собою комп’ютерів за умови, що вистачить на це коштів. Розмістити тисячі комп’ютерів по всьому світу ще легше, оскільки при цьому не потрібно шукати для них відповідну велику кімнату. Проблеми починаються, коли потрібно організувати обмін даними між комп’ютерами для спільної роботи при розв’язанні єдиного завдання. Тому був пророблений великий обсяг роботи по розробці технології з’єднання комп’ютерів, а різні технології з’єднання привели до якісних систем, які відрізняються одна від одної типами систем і різним організаціям програмного забезпечення.

Весь обмін даними між електронними компонентами в кінцевому підсумку зводиться до обміну повідомленнями – чітко визначеними бітовими рядками. Різниця полягає у використовуваних масштабах часу, відстані і логічної організації. На одному полюсі знаходиться багатопроцесорна система із загальним простором

пам’яті, де від двох до тисячі центральних процесорів обмінюються даними через загальну пам’ять. У цій моделі кожен центральний процесор має рівний доступ до всієї фізичної пам’яті і може читати, і записувати окремі слова. Доступ до слова пам’яті зазвичай займає 5-50 нс. Зараз вже немає нічого незвичайного в розміщенні на один кристал центрального процесора більш одного обчислювального ядра з наданням ядрам спільного доступу до основної пам’яті (а іноді навіть і до спільним блокам кеш-пам’яті).

Багатопроцесорна обробка зародилася в середині 1950-х в ряді компаній (IBM, Control Data Corporation). На початку 1960-х Burroughs Corporation представила симетричний мультипроцесор типу MIMD з чотирма CPU, що має до шістнадцяти модулів пам’яті, з’єднаних координатним з’єднувачем (перша архітектура SMP, Symmetric MultiProcessing – технологія симетричної мультипроцесорності) [9]. Широко відомий і успішний комп’ютер CDC 6600 був представлений в 1964 році і забезпечував CPU десятьма підпроцесорами (периферійними процесорами). В кінці 1960-х Honeywell випустила іншу симетричну мультипроцесорну систему з восьми CPU Multics.

У той час як багатопроцесорні системи розвивалися, інші технології також йшли вперед, зменшуючи розміри процесорів і збільшуючи їх здатність працювати на значно більшій тактовій частоті. Багатопроцесорні системи, що розглядаються в цьому розділі, широко використовуються для вирішення багатьох задач в науці, промисловості, а також інших областях людської діяльності.

Ще одна область розвитку, що має відношення до досліджуваного питання, – це неймовірно бурхливе зростання мережі Інтернет. Система, що складається з тисячі комп’ютерів, розосереджених по всьому світу, не відрізняється від системи, що складається з тисячі комп’ютерів, що знаходяться в одному приміщенні, хоча затримки по часу і інші технічні характеристики у цих двох систем розрізняються. Ці системи також будуть коротко розглянуті в цьому розділі.

ФОРМИ ПАРАЛЕЛІЗМУ

Паралелізм – це можливість одночасного виконання більш однієї арифметико-логічного операції або програмної гілки. Вивчення ряду алгоритмів і програм показало, що можна виділити такі основні **форми паралелізму** [38]:

– дрібнозернистий паралелізм;

– крупнозернистий паралелізм.

Дрібнозернистий паралелізм (паралелізм суміжних операцій або скалярний паралелізм) забезпечується за рахунок паралелізму всередині базових блоків (5-20 команд), які є частинами програм, що не містять умовних і безумовних переходів. Цей вид паралелізму реалізується блоками одного процесора (різними АЛП, помножувачами, блоками звернення до пам'яті, зберігання адреси, переходів тощо) і навіть при оптимальному плануванні паралелізм не може бути більшим.

Крупнозернистий паралелізм забезпечується за рахунок паралелізму незалежних програмних гілок, підпрограм, потоків (ниток) всередині програм і реалізується процесорами або ядрами багатоядерних процесорів. Крупнозернистий паралелізм включає векторний паралелізм і паралелізм незалежних гілок.

Векторний паралелізм. Найбільш поширеною в обробці структур даних є векторна операція (природний паралелізм).

Паралелізм незалежних гілок. Суть паралелізму незалежних гілок полягає в тому, що в програмі розв'язання великого завдання можуть бути виділені програмні частини, незалежні за даними.

Ефективність паралельних обчислень можна обчислити за допомогою закону Амдала. Джин Амдал розробляв в IBM комп'ютерні архітектури, але популярність йому приніс його закон, в якому розраховується максимально можливе поліпшення (прискорення – R) системи при поліпшенні її частини. Закон використовується для обчислення максимального теоретичного поліпшення роботи системи при використанні декількох процесорів:

$$\textbf{Прискорення (R)} = 1/(F + (1-F)/N).$$

Використовуючи це рівняння, можна обчислити максимальне поліпшення продуктивності системи, що використовує N процесорів і фактор F, який показує, яка

частина системи не може бути розпаралелена (частина системи, яка є послідовною за своєю природою).

Через те, що не всі в задачі можуть бути розпаралелені і є непродуктивні витрати в управлінні процесорами, прискорення виявляється трохи менше.

Нехай, наприклад, F = 0,2 (що є реальним значенням), тоді прискорення не може перевищувати 5 при будь-якому числі процесорів, тобто максимальне прискорення визначається потенційним паралелізмом завдання.

Якщо система має кілька архітектурних рівнів з різними формами паралелізму, то якісно загальне прискорення в системі буде:

$$\mathbf{R = r1 \times r2 \times r3,}$$

де ri – прискорення деякого рівня.