

АРХІТЕКТУРА ПОТОКОВИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

Існують труднощі, пов'язані з розв'язанням проблем автоматизації паралельного програмування, необхідного для забезпечення ефективного використання для широкого кола задач матричних (паралельних) ОС, тобто систем типу SIMD (ОКМД).

Конвеєрні і конвеєрно-векторні ОС найбільший ефект дають під час реалізації спеціалізованих систем з фіксованими алгоритмами, за яких досягається оптимальна структура для задачі, яка розв'язується. В ОС універсального призначення склад позицій конвеєра є далеко не оптимальним для кожної даної задачі, до того ж виникають витрати часу на комутацію позицій конвеєра під поточну команду.

В таких умовах актуальними є дослідження нових шляхів побудови високопродуктивних ОС, одними з яких є ОС з управлінням потоком даних (операндів), або, іншими словами, потокові ОС.

У системах з управлінням потоками даних передбачається наявність великої кількості спеціалізованих операційних блоків для певних видів операцій (додавання, множення тощо, окремих для різних типів даних). Дані наділяються покажчиками типу даних (тегами), на основі яких, у міру готовності даних (операндів), до обробки (а не в порядку послідовності команд у програмі) вони завантажуються у відповідні вільні операційні блоки. У разі достатньої кількості операційних блоків може бути досягнутий високий рівень розподілення обчислювального процесу (близький до «потенційного паралелізму» програми). До того ж у самих операційних блоках може бути використана конвеєрна обробка. Таким чином, утворюються умови для реалізації високої продуктивності системи.

В усіх раніше розглянутих ЕОМ і обчислювальних системах порядок виконання операцій над даними у ході розв'язання задачі суворо детермінований, він однозначно визначається послідовністю команд програми.

Принципова відмінність поточкових машин полягає в тому, що команди виконуються не за порядком чергування команд у тексті програми, а в міру готовності їх операндів. Як тільки будуть вираховані операнди команди, вона може захоплювати вільний операційний простір і виконувати задану їй операцію. В цьому випадку послідовність, в якій виконуються команди, вже не є детермінованою.

Розглянемо як приклад обчислення на потоковій ОС коренів квадратного рівняння:

$$ax^2 - bx + c = 0,$$

при $b^2 - 4ac > 0;$

$$x_{1,2} = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Будемо вважати, що на вхід системи надходить група даних a_i, b_i, c_i і за ними обчислюються корені рівняння. Розв'язання може бути отримане за допомогою операційних пристроїв, які виконують операції додавання, віднімання, множення, піднесення до степеня, ділення і вирахування квадратного кореня.

Процес розв'язання можна подати у такій послідовності:

K0: $A1 := 2 \times a;$ K1: $B1 := b^2$ <піднесення до степеня>;	K5: $D1 := \text{sqrt}(D)$ <обчислення квадратного кореня>;
	K6: $B2 := b + D1;$
K2: $C := 4 \times c;$	K7: $B3 := b - D1;$
K3: $C1 := C \times a;$	K8: $X1 := B2/A1;$
K4: $D := B1 - C1;$	K9: $X2 := B3/A1.$

У потокових обчислювальних моделях для описання обчислень використовують орієнтований граф, у якому вершини відображають операції, а дуги показують потоки даних між вершинами графа, які вони з'єднують.

На рис. 1 показаний граф потоку даних, які управляють розв'язанням задачі, що розглядається. Великими кружками показані операційні пристрої, а маленькими – мітки готовності даних. У початковому стані процесу готові всі вхідні дані.

Вершина графа активізується в довільному порядку у разі готовності їх операндів. Можна вважати, що активізація вершини супроводжується поглинанням міток готовності на їх входах. По закінченні операції міткою готовності позначається вихідна дуга вершини. Переміщення міток за графом потоку даних відображає проходження обчислювального процесу.

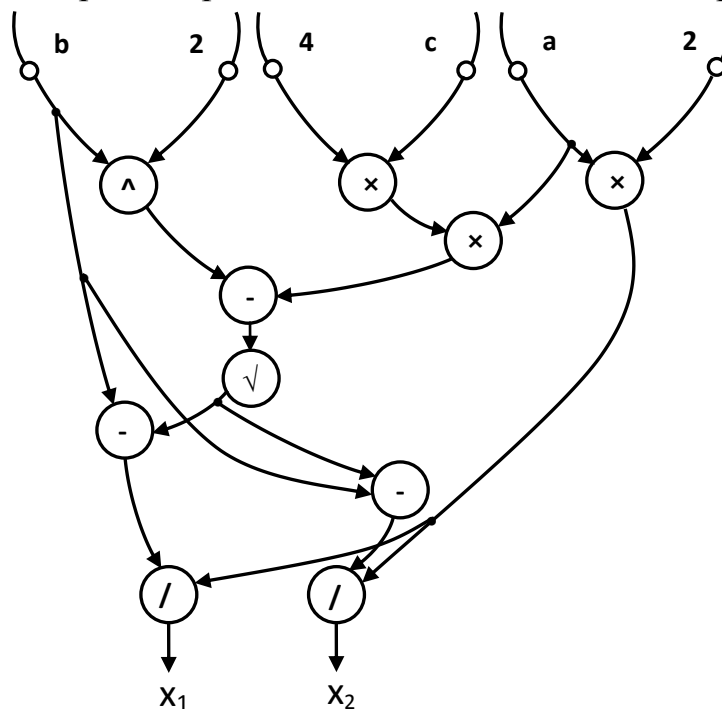


Рисунок 1 - Граф потоків операндів

Виникає запитання, як дані і відповідні команди знаходять один одного? Для відповіді на нього звернемося до рис. 2, який пояснює ідею процесора, що управляє потоком даних.

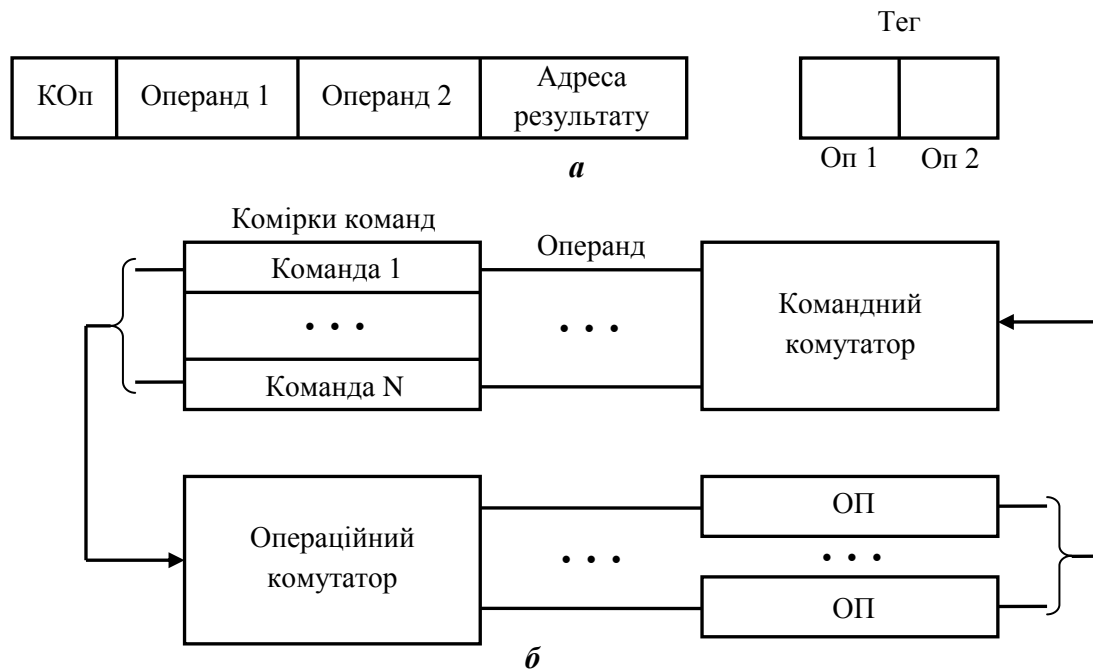


Рисунок 2 - Спрощена структура процесора з управлінням потоком даних:
а – структура команди; б – ОП – операційний пристрій

«Потокова програма» розташовується в масиві комірок команд. Команда, наряду з кодом операції, містить поля, куди заносяться готові операнди, і поле, яке містить адреси команд, в які повинен бути направлений як операнд результат операції. Крім того, кожній команді поставлений у відповідність дворозрядний тег (що розташований в управляючому пристрої), розряди якого встановлюються в 1 при занесенні в тіло команди відповідних операндів.

У стані тегу 11 (обидва операнди готові) ініціюється запит до операційного комутатора на передачу готової команди у відповідний коду операції (і тегу операнда, який визначає тип даних) операційний пристрій.

Результат виконання команди над її безпосередньо адресованими операндами направляється через командний комутатор згідно з вказаними в команді адресами в комірки команд і розташовується в їх поля операндів. Далі вказана процедура циклічно повторюється, до того ж управління цим процесом повністю децентралізоване.

БАГАТОМАШИННІ ТА БАГАТОПРОЦЕСОРНІ КОМП'ЮТЕРНІ СИСТЕМИ

Обчислювальні системи можуть будуватися на базі декількох комп'ютерів або на базі декількох процесорів. В першому випадку ОС буде *багатомашинною*, в другому - *багатопроесорною*.

Багатомашинна ОС містить деяке число комп'ютерів, інформаційно взаємодіючих між собою. Комп'ютери можуть знаходитися поряд один з одним, а можуть бути віддаленими один від одного на деяку, іноді значну відстань (обчислювальні мережі).

В *багатомашинних ОС* кожний комп'ютер працює під управлінням своєї операційної системи (ОпС). А оскільки обмін інформацією між комп'ютерами, що взаємодіють один з одним, виконується під управлінням ОпС, динамічні характеристики процедур обміну дещо погіршуються (потрібен час на узгодження роботи самих ОпС). Інформаційна взаємодія комп'ютерів в багатомашинній ОС може бути організовано на рівні:

- процесорів;
- оперативної пам'яті;
- каналів зв'язку.

При безпосередній взаємодії процесорів один з одним інформаційний зв'язок реалізується через регістри процесорної пам'яті і вимагає наявності в ОпС дуже складних спеціальних програм.

Взаємодія на рівні оперативної пам'яті (ОП) зводиться до програмної реалізації загального поля оперативної пам'яті, що дещо простіше, але також вимагає суттєвої модифікації ОпС. Під загальним полем мається на увазі рівнодоступність модулів пам'яті: всі модулі пам'яті доступні всім процесорам і каналам зв'язку.

На рівні каналів зв'язку взаємодія організовується найбільш просто і може бути досягнуто зовнішніми по відношенню до ОпС програмами-драйверами, що забезпечують доступ від каналів зв'язку однієї машини до зовнішніх пристроїв інших (формується загальне поле зовнішньої пам'яті і загальний доступ до пристроїв вводу-виводу).

Все вищесказане ілюструється схемою взаємодії комп'ютерів в двохмашинній ОС, представленої на рис. 3.

Зважаючи на складність організації інформаційної взаємодії на 1-у і 2-у рівнях в більшості багатомашинних ОС використовується 3-й рівень, хоча і динамічні характеристики (в першу чергу швидкодія), і показники надійності таких систем істотно нижче.

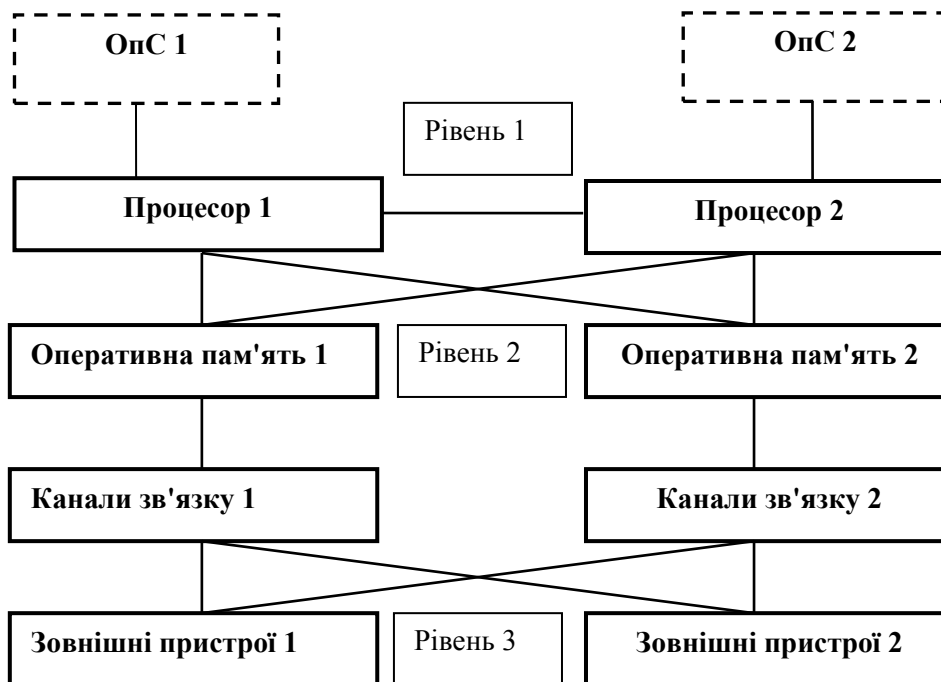


Рисунок 3 - Схема взаємодії комп'ютерів в ОС

В *багатопроцесорній ОС* є декілька процесорів, інформаційно взаємопов'язаних між собою або на рівні регістрів процесорної пам'яті, або на рівні ОЗП. Цей тип взаємодії використовується в більшості випадків, бо організовується значно простіше і зводиться до створення загального поля оперативної пам'яті для всіх процесорів. Загальний доступ до зовнішньої пам'яті і пристроїв вводу-виводу забезпечується звичайно через канали ОЗП. Важливим є і те, що багатопроцесорна обчислювальна система працює під управлінням єдиної ОпС, загальної для всіх процесорів. Це істотно покращує динамічні характеристики ОС, але вимагає наявності спеціальної, дуже складної ОпС.

Схема взаємодії процесорів в ОС показана на рис. 4.



Рисунок 4 - Схема взаємодії процесорів в ОС

Швидкість і надійність багатопроцесорних ОС в порівнянні з багатомашинними, взаємодіючими на 3-у рівні, істотно підвищуються,

по – перше, із-за більш швидкого обміну інформацією між процесорами, більш швидкого реагування на ситуації, що виникають в системі;

по – друге, зважаючи на більший ступінь резервування пристроїв системи (система зберігає працездатність, поки працездатні хоча б по одному модулю кожного типу пристроїв).

Типовим прикладом масових багатомашинних ОС можуть служити комп'ютерні мережі, прикладом багатопроцесорних ОС - суперкомп'ютери.