

*Ярослава Сікора,*

кандидат педагогічних наук, доцент

[iaroslava.sikora@gmail.com](mailto:iaroslava.sikora@gmail.com)

*Богданна Якимчук,*

кандидат технічних наук

[yakumchuk.bl@gmail.com](mailto:yakumchuk.bl@gmail.com)

(Житомирський державний університет імені Івана Франка)

## СПОСОБИ РЕАЛІЗАЦІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

Теорія паралельних обчислень характеризується великою різноманітністю ідей, підходів до вирішення завдань, розробки паралельних програм. Часто завдання вимагають урахування їх особливостей для розробки ефективних програм розв'язання. Це, у свою чергу, призводить до розробки різноманітних та численних спеціалізованих засобів паралельного програмування.

Нині застосовується кілька парадигм паралельного програмування:

- використання низькорівневих бібліотек передачі повідомлень для звичайних мов програмування;
- спеціальні вказівки: анотації, прагми, або паралельні оператори, – зазвичай, реалізовані у вигляді коментарів, – які вставляються у послідовну програму та допомагають компілятору, який враховує ці вказівки, згенерувати відповідну паралельну програму;
- нові паралельні мови (зокрема, розширення послідовних мов новими конструкціями для опису паралельних обчислень).

Одним з найбільш популярних засобів паралельного програмування, заснованих на традиційних мовах програмування, є технологія OpenMP. В її основі послідовна програма, а для створення паралельної версії використовується набір директив, функцій та змінних середовища. Це дозволить паралельній програмі коректно працювати на різних комп'ютерах з розподіленою пам'яттю, які підтримують OpenMP API.

Цей стандарт розробляється некомерційною організацією OpenMP ARB

(Architecture Review Board), із залученням представників компаній-розробників SMP-архітектури та ПЗ. OpenMP підтримує роботу з мовами Fortran та C/C++.

В OpenMP реалізуються паралельні обчислення з використанням багатопотоковості. Головний (master) потік створює набір «підпорядкованих» (slave) потоків, між якими розподіляється задача. Потоки виконуються паралельно на багатопроцесорній машині, зокрема, кількість процесорів не обов'язково має бути більша або дорівнювати кількості потоків [1, с. 167].

MPI – стандарт паралельної обробки даних, було створено в 1994 році групою Message Passing Interface Forum, а у 1997 році з'явився стандарт MPI-2. Внаслідок своєї складності, стандарти MPI та MPI-2 не були в повному обсязі реалізовані у жодній системі. Тому при теоретичному вивченні та практичному застосуванні рекомендується використовувати створені в Арагонській національній лабораторії (США) реалізації стандартів MPI та MPI-2 – MPICH та MPICH-2 відповідно.

Створення реалізації MPICH-2 здійснювалося з головною метою – здійснити ефективну підтримку багатоплатформності, із урахуванням як кластерних систем від суперкомп'ютерів (Blue Gene) до загальнодоступних (настільні системи, багатоядерні процесори), так і комутаційних мереж (Ethernet 10 Гбіт/с, InfiniBand, Myrinet, Quadrics). У якості програмних засобів, зазначені стандарти реалізуються у вигляді бібліотек MPI, що містять понад 300 різноманітних програмних функцій.

До спільних рис бібліотеки MPI та OpenMP можна віднести безоплатність та можливість використовувати однакові мови програмування (C, C++ та Фортран). Проте, у MPI та OpenMP є значні відмінності. В межах паралельної архітектури залежно від способу використання основної пам'яті можна реалізувати два основні варіанти паралельних обчислень. Бібліотека OpenMP орієнтується на системи зі спільною пам'яттю (shared memory), за допомогою якої здійснюється обмін даними між окремими процесорами (ядрами центрального процесора). Також можлива паралельна архітектура з розподіленою (індивідуальною) пам'яттю (distributed memory). У таких кластерних системах

процесори працюють незалежно один від одного, що унеможливорює безпосередній обмін даними як між ними, так і через основну пам'ять.

Здійснення зв'язку між окремими процесами, що виконуються на різних ядрах кластера, потребує спеціального механізму обміну повідомленнями (інформацією) між ними. З цією метою було розроблено стандарт обміну повідомленнями MPI між різними процесорами під час їхньої паралельної роботи, де різноманітні комунікатори є посередниками при цьому обміні інформацією.

MPI базується на архітектурі MIMD, в той же час, технологія обчислень OpenMP зорієнтована на архітектуру SIMD за класифікацією Флінна. В MPI паралелізм реалізовується завдяки роботі багатьох незалежних процесів, в OpenMP – за рахунок роботи багатьох потоків одного процесу.

Стандарт MPI складніше практично реалізувати. Якщо в OpenMP необхідно додати до вже налагодженої послідовної програми паралельні директиви, то в MPI потрібно написати і налагодити принципово іншу програму. Окрім складності програмування до недоліків MPI можна також віднести те, що поки не існує реалізацій MPI, які повною мірою забезпечують суміщення обмінів з обчисленнями [2, с. 64-65].

Отже, OpenMP – це метод програмування на пристроях загальної пам'яті. Це означає, що паралелізм виникає коли кожен паралельний потік має доступ до всіх даних. MPI – це спосіб програмування на пристроях із розподіленою пам'яттю. Це означає, що паралелізм відбувається там, де кожен паралельний процес працює у власному просторі пам'яті. Досвід вивчення паралельних обчислювальних процесів засвідчив, що обидві технології можуть використовуватися спільно, щоб оптимально використовувати в кластері багатоядерні системи.

### **Список використаних джерел та літератури**

1. Коцовський В. М. Теорія паралельних обчислень : навч. посіб. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с.
2. Семеренко В. П. Технології паралельних обчислень : навч. посіб. Вінниця : ВНТУ, 2018. 104 с.