УДК 004.4

Романенко Лев Анатолійович
бакалавр програмної інженерії
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
Романенко Лев Анатольевич
бакалавр программной инженерии
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»
Lev Romanenko
Bachelor of software engineering
The National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»

ЕКСПЕРИМЕНТАЛЬНЕ ПОРІВНЯННЯ РОБОТИ ЗАСОБІВ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ НА РІЗНИХ КЛАСТЕРАХ ДЛЯ АЛГОРИТМУ ФЛОЙДА-УОРШАЛА

ЭКСПЕРИМЕНТАЛЬНОЕ СРАВНЕНИЕ РАБОТЫ СРЕДСТВ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ НА РАЗЛИЧНЫХ КЛАСТЕРАХ ДЛЯ АЛГОРИТМА ФЛОЙДА-УОРШАЛА

EXPERIMENTAL COMPARISON OF THE WORK OF PARALLEL PROGRAMMING TOOLS ON VARIOUS SUPERCOMPUTERS FOR THE FLOYD-WARSHAL ALGORITHM

Анотація. *У даній статті проведено експериментальне порівняння роботи бібліотек паралельного програмування MPI та openMP на суперкомп'ютері Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» та кластері Colfax International. Проаналізовано особливості архітектури обчислювальних машин та її взаємодію з різними інструментами паралельного програмування. На прикладі динамічного алгоритму Флойда-Уоршала показана залежність обчислювальних характеристик від різної кількості потоків виконання та засобів паралельного програмування.*

Ключові слова: *MPI, openMP, алгоритм Флойда-Уоршала, суперкомп'ютер.*

Аннотация. *В данной статье выполнено экспериментальное сравнение работы библиотек параллельного программирования MPI и openMP на суперкомпьютере Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского» и кластере Colfax International. Проанализированы особенности архитектуры вычислительных машин и ее взаимодействие с различными инструментами параллельного программирования. На примере динамического алгоритма Флойда-Уоршала показана зависимость вычислительных характеристик от разного количества потоков исполнения и инструментов параллельного программирования.*

Ключевые слова: *MPI, openMP, алгоритм Флойда-Уоршала, суперкомпьютер.*

Summary. *In this paper an experimental comparison of the work of the parallel programming of MPI and openMP libraries on the supercomputer of the National Technical University of Ukraine «Igor Sikorskiy Kiev Polytechnic Institute» and the Colfax International cluster was conducted. The peculiarities of the architecture of computing machines and its interaction with various parallel programming tools are analyzed. An example of a dynamic Floyd-Worceshal algorithm shows the dependence of computer from a different number of execution flows and parallel programing tools.*

Key words: *MPI, openMP, Floyd-Worshall algorithm, supercomputers.*

Під час написання програм для кластерних систем, доводиться звертати увагу на особливості тієї машини, де дане програмне забезпечення буде виконуватися. Тому метою дослідження було дізнатися, як впливає апаратне устаткування на характеристики обчислень і визначити яка комбінація фізично-логічних засобів буде найкращою. До уваги бралися дві бібліотеки для паралельного програмування такі як MPI та openMP і два еквівалентних за потужністю суперкомп'ютери — Colfax International та кластер КПІ ім. Ігоря Сікорського. Як обчислювальна задача для програмування, був обраний алгоритм Флойда-Уоршала.

Алгоритм Флойда-Уоршала — це динамічний алгоритм для знаходження найкоротших відстаней між усіма вершинами зваженого орієнтованого графа. Його було розроблено в 1962 році Робертом Флойдом і Стівеном Уоршалом.

Основою задачі є орієнтований граф, в якому кожній дузі вершин відповідає позитивна вартість. Загальне завдання знаходження найкоротших шляхів полягає в пошуку для кожної впорядкованої пари вершин будь-якого шляху від початкової вершини в кінцеву, де є мінімальна довжина серед усіх можливих шляхів.

Апаратне забезпечення кластерів на яких виконувалися обчислення. Для виконання обчислень обрано два суперкомп'ютери: Colfax International та кластер Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Кластери мають наступні технічні характеристики.

КПІ ім. Ігоря Сікорського [2]:
- вузли представлені 4-ядерними процесорами Intel Xeon E5440 2.83ГГц та 8 Гб оперативної пам'яті у кожному;
- операційна система CentOS release 6.4;
- локальний менеджер ресурсів — Slurm.

Colfax Internatioanl [1]:
- вузли представлені 8-одноядерними процесорами Intel Xeon Phi 2.1 ГГц з 4 Гб оперативної пам'яті у кожному;
- операційні системи CentOS, Fedora, Scientific Linux;
- локальний менеджер ресурсів — Brigth Cluster Manager.

Обидва кластера дотримуються політики обмеження ресурсів для користувача. Colfax використовує віртуальне середовище, яке обмежує кількість ресурсів для кожного користувача. Кластер КПІ ім. Ігоря Сікорського обмежує ресурси загально, не з допомогою віртуального середовища. Система обмежує кількість потоків та кількість вузлів для користувача. Програма може бути виконана на одному вузлі, але на декількох потоках, або на декількох вузлах, але з одним потоком.

Collfax використовує специфічний Intel C Compiler [3, с. 41], який розрахований під конкретне апаратне устаткування. Основні можливості компілятора наступні:
- міжпроцедурна оптимізація;
- автоматичне розпаралелювання коду;
- векторизація для SSE, SSE3, SSE4;
- оптимізація з урахуванням профільної інформації.

Intel C++ compiler підтримує стандарт OpenMP 3.0 для написання паралельних програм. Також містить модифікацію OpenMP під назвою Cluster OpenMP, за допомогою якої можна запускати додатки написані відповідно до OpenMP на кластерах, що використовують MPI. Кластер КПІ ім. Ігоря Сікорського має базовий GNU C Compiler. Він використовується як стандартний компілятор для вільних UNIX-подібних операційних систем.

Результати проведених обчислень

Для порівняння засобів паралельного програмування було обрано такі характеристики як час обчислення, коефіцієнт прискорення та коефіцієнт ефективності. Обчислення здійснювалися на наборі векторів розміром 2700 елементів для 8 потоків виконання.

Таблиця 1

Результати вимірювань для кластера КПІ ім. Ігоря Сікорського

К-ть потоків	Час		Прискорення		Ефективність	
	MPI	openMP	MPI	openMP	MPI	openMP
1	41,308186	41,308186	1	1	1	1
2	22,956614	22,731996	1,799402	1,817183	0,899701	0,908591
3	22,587734	22,739856	1,828788	1,816554	0,609596	0,605518
4	16,121003	16,181408	2,562383	2,552818	0,640596	0,638204
5	19,122467	19,319592	2,160191	2,13815	0,432038	0,42763
6	15,877504	21,076918	2,60168	1,959878	0,433613	0,326646
7	18,669774	18,802218	2,21257	2,196985	0,316081	0,313855
8	16,261717	16,25834	2,540211	2,540738	0,317526	0,317592

Результати випробувань для кластера КПІ ім. Ігоря Сікорського наведені в таблиці нижче (табл. 1).

На таблиці з даними можна спостерігати скорочення часу, це природньо, адже збільшується кількість ядер при одному і тому ж об'ємі даних — графік наглядно це показує (рисунок 1). Різниця в часі для обраних бібліотек майже відсутня, а це говорить про рівнозначну швидкість обчислень.

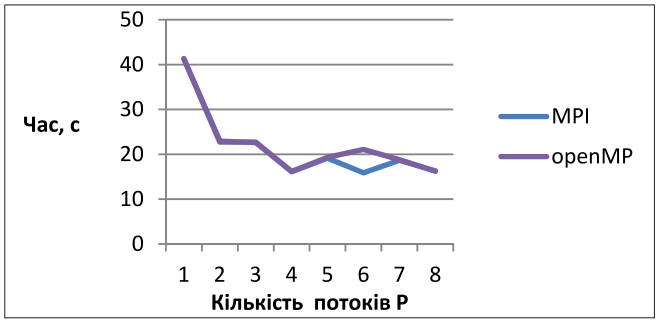

Рисунок 1. Графік залежності часу від кількості потоків обчислення

На таблиці з даними про прискорення (табл. 1), можна бачити що прискорення обчислень збільшується на ядрах 2, 3, 4, а потім коливається в сталому діапазоні. Можливо, така поведінка обчислень на кількості ядер 5, 6, 7, 8 пов'язана з накладними затратами на комунікацію між ними. Виходячи з результатів, такий об'єм даних оптимальніше обчислювати на кількості ядер від 1 до 4, а якщо об'єм даних пропорційно більший, то на кількості ядер від 5 до 8. Знову ж таки обидві бібліотеки показують ідентичні результати для прискорення (рисунок 2).

На графіку можна чітко спостерігати пропорційне зменшення ефективності до кількості ядер. Особливі піки спаду ефективності можна спостерігати на непарній кількості ядер. Можливо, що така закономірність зумовлена апаратними особливостями. Загалом спад ефективності можна пов'язати з тими ж таки затратами на комунікацію та синхронізацію між ядрами. Аналізуючи дані коефіцієнта ефективності, можна сказати що на 6 ядрах, бібліотека openMP втрачає показники в порівнянні з MPI (рисунок 3).

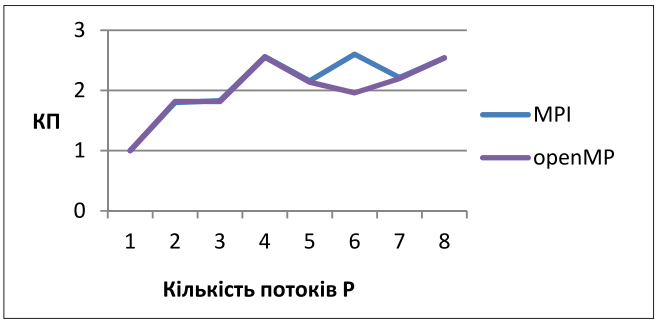

Рисунок 2. Графік залежності коефіцієнта прискорення від кількості потоків обчислення

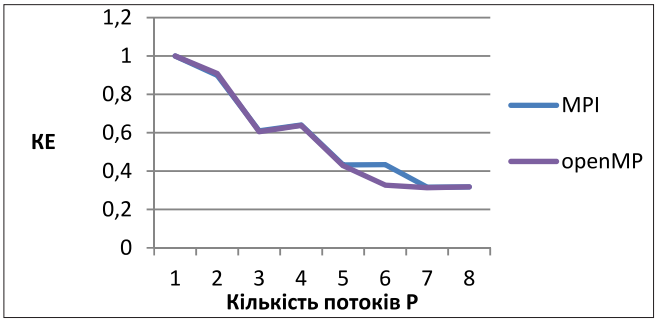

Рисунок 3. Графік залежності коефіцієнта ефективності від кількості потоків обчислення

Результати випробувань для кластера Colfax International наведені в таблиці нижче (табл. 2).

Аналізуючи дані для Colfax, можна спостерігати ті ж тенденції, що і для кластера КПІ ім. Ігоря Сікорського (рис. 4, рис. 5, рис. 6). Відмінність полягає лише в швидкості — Colfax cluster обчислює той же об'єм даних, але на порядок швидше. Також можна відзначити те, що коефіцієнт ефективності openMP на даному кластері більший за MPI.

Таблиця 2

Результати вимірювань для кластера Colfax International

К-ть потоків	Час		Прискорення		Ефективність	
	MPI	openMP	MPI	openMP	MPI	openMP
1	20,390237	20,390237	1	1	1	1
2	11,336521	10,747534	1,798633	1,897201	0,899316	0,948601
3	7,706041	7,563788	2,646007	2,695771	0,882002	0,89859
4	6,22211	5,778772	3,277061	3,528472	0,819265	0,882118
5	4,87605	4,81701	4,181712	4,232965	0,836342	0,846593
6	4,603811	4,403029	4,428991	4,630957	0,738165	0,771826
7	3,781575	3,728906	5,391996	5,468155	0,770285	0,781165
8	3,297379	3,429241	6,183771	5,945991	0,772971	0,743249

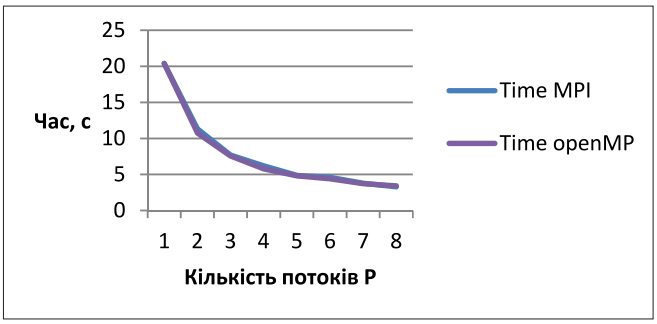

Рисунок 4. Графік залежності часу від кількості потоків обчислення

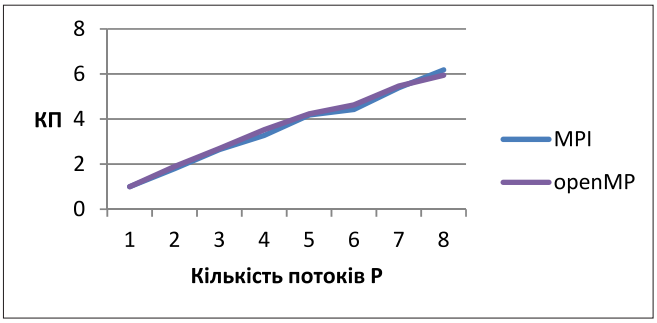

Рисунок 5. Графік залежності коефіцієнта прискорення від кількості потоків обчислення

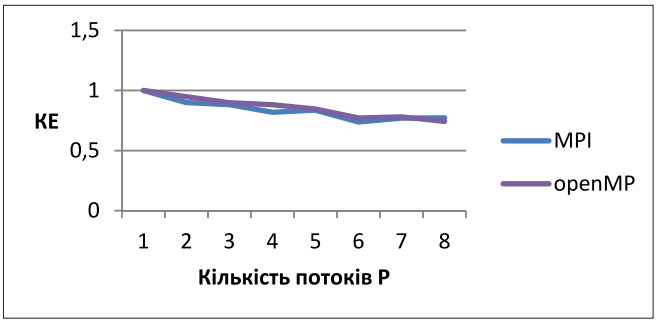

Рисунок 6. Графік залежності коефіцієнта ефективності від кількості потоків обчислення

Результати та висновки. Як показали дослідження, коефіцієнт ефективності бібліотек openMP та MPI варіюється в залежності від обладнання на якому вони застосовуються. Загалом різниця в показниках не суттєва, але на великих наборах даних буде відчутна.

Компілятор відіграє важливу роль в швидкості обчислень. Апаратно орієнтовні компілятори э більш оптимізовані. Таку тенденцію можна побачити аналізуючи час за який було оброблено один і той же об'єм даних на кластерах КПІ ім. Ігоря Сікорського та Colfax International. Апаратно орієнтовні компілятори є ефективніші, в порівнянні з базовими-універсальними, тому що мають змогу виконувати високорівневі, а також цільові оптимізації під процесори на які вони розраховані. Як результат, такі компілятори в парі з їх процесорами працюють швидше. Це можна побачити на прикладі роботи Colfax International.

Технологія MPI менш вимоглива до засобів зв'язку між процесорами на якій вона може бути ефективно реалізована, але більш вимоглива до програміста, важча в застосуванні ніж технологія OpenMP.

Апаратне устаткування, на базі якої можлива ефективна реалізація OpenMP (SMP-сервери), коштує дорого і погано масштабується. Обладнання, придатне для реалізації MPI (спеціалізовані мережі обчислювальних кластерів), набагато дешевше, і масштабується практично необмежено, але має більш низьку ефективність.

Отже вибір засобу паралельного програмування повинен залежати від наявного обладнання та апаратного програмного забезпечення.

Література

1. Colfax International [Електронний ресурс] — Режим доступу: http://www.colfax-intl.com/nd/index.aspx

2. Центр суперкомп'ютерних обчислень НТУУ «КПІ ім. І. Сікорського» [Електронний ресурс] — Режим доступу: http://grid.kpi.ua/index.php/ru/national-resource-centre/10-centr-superkompyuternih-obchislen.html

3. Parallel programming and optimization with intel xeon PHI coprocessors/ A. Vladimirov, R. Asai, V. Karpusenko, 2013–2015.