**РОЗДІЛ 3. РОЗРОБКА ПРОГРАМИ ПРГ2 ДЛЯ ПКС ЛП**

Оскільки математична задача співпадає з тою, що розглядалась в розділі 2.1, то перший крок по рекомендованій методиці[ не матиме відмінностей від уже виконаних. Тому розділ «Розробка паралельного математичного алгоритму» пропущений.???? ВСТАВИТЬ!!!!

**3.1. Розробка алгоритмів процесів**

|  |
| --- |
| **Крок алгоритму**  1. Якщо id = 1, ввести a, MK, MT  2. Якщо id = Р, ввести MO, MB  3. Якщо id = 1, передать Т(Р) дані: α , МК, МTР\*Н/2  4. Якщо id = Р, прийняти від Т(1) дані: α , МК, МTР\*Н/2  5. Якщо id = Р, передати Т(1) данні: МО (Р\*Н-Р\*Н/2), МB(Р\*Н-Р\*Н/2)  6. Якщо id = 1, прийняти від Т(Р) дані: МО (Р\*Н-Р\*Н/2), МB(Р\*Н-Р\*Н/2)  7. Передати з Т(j) в Т(j+1) дані: α, МК, МО ((Р\*Н-Р\*Н/2)-j\*H), МT((Р\*Н-Р\*Н/2)-j\*H), MB((Р\*Н-Р\*Н/2)-j\*H). j=1 .. (P\*H - P\*H/2), j != (P\*H-P\*H/2)  8. Прийняти з Т(j) в Т(j+1) дані: α, МК, МО ((Р\*Н-Р\*Н/2)-j\*H), МT((Р\*Н-Р\*Н/2)-j\*H), MB((Р\*Н-Р\*Н/2)-j\*H). j=1 .. (P\*H - P\*H/2), j != (P\*H-P\*H/2)  9. Передать з Т(j) в Т(j-1) дані: α, МК, МО(Р\*Н/2-j\*H), МT(Р\*Н/2-j\*H), MB(Р\*Н/2-j\*H). j = P .. P\*H/2, j != P\*H/2 (кроки 7 та 9, 8 та 10 відповідно паралельні)  10. Прийняти з Т(j) в Т(j-1) дані: α, МК, МО(Р\*Н/2-j\*H), МT(Р\*Н/2-j\*H), MB(Р\*Н/2-j\*H). j = P .. P\*H/2, j != P\*H/2  11.Розрахунок z(j) = min(MOh);  12. Передати з T(j) в T(j-1) дані: z(j), j = (P\*H- P\*H/2) .. 1, j != 1  13. Прийняти з Т(j) в Т(j-1) дані:z(j), j = (P\*H - P\*H/2) .. 1, j != 1  14. Розрахунок z(j-1) = min(z(j-1), z(j)); j = (P\*H - P\*H/2) .. 1, j != 1  15. Передати з Т(j) в Т(j+1) дані: z(j), j = P\*H/2 .. P, j != P  16. Прийняти з Т(j) в Т(j+1) дані: z(j), j = P\*H/2 .. P, j != P  17. Розрахунок z(j+1) = min(z(j+1), z(j)); j = P\*H/2 .. P, j != P  (кроки 12-17 відповідно паралельні)  18. Передати з Т(1) в Т(P) дані: z(1)  19. Прийняти з Т(1) в Т(P) дані: z(1)  20. Розрахунок z(P) = min(z(1), z(P));  18. Передати з Т(P) в Т(1) дані: z(P)  19. Прийняти з Т(P) в Т(1) дані: z(P)  21. Передати з Т(j) в Т(j+1) дані: z(j). j=1 .. (P\*H - P\*H/2), j != (P\*H-P\*H/2)  22. Прийняти з Т(j) в Т(j+1) дані: z(j). j=1 .. (P\*H - P\*H/2), j != (P\*H-P\*H/2)  23. Передать з Т(j) в Т(j-1) дані: z(j). j = P .. P\*H/2, j != P\*H/2  24. Прийняти з Т(j) в Т(j-1) дані: z(j). j = P .. P\*H/2, j != P\*H/2  (кроки 21-24 відповідно паралельні)  25. Розрахунок MAН = z \* MTH + a \* MK \* MBН  26. Передати з Т(j) в Т(j-1) дані: MА j\*H, j = (P\*H - P\*H/2) .. 1, j != 1  27. Прийняти з Т(j) в Т(j-1) дані: MА j\*H, j = (P\*H - P\*H/2) .. 1, j != 1  28. Передати з Т(j) в Т(j+1) дані: MА j\*H, j = P\*H/2 .. P, j != P  29. Прийняти з Т(j) в Т(j+1) дані: А j\*H, j = P\*H/2 .. P, j != P  (кроки 26-29 відповідно паралельні)  30. Якщо id = 1, передати задачі Т(Р) дані: MА(P\*H - P\*H/2)\*H  31. Якщо id = P, прийняти від задачі Т(1) дані: MА(P\*H - P\*H/2)\*H  32. Якщо id = P, вивести MА  **3.2. Розробка схеми взаємодії процесів** |

**Задачі Т(1)-Т(Р)**

На основі алгоритму взаємодії всіх процесів, наведеному в попередньому розділі, було розроблено структурну схему взаємодії процесів (рис. 3.1). Схема дозволяє наочно контролювати зв’язки між процесами, а також дозволяє виявити ситуації в яких програма може виконуватись не повністю (данні будуть відправленні не в усі процеси).

Для демонстрації взаємодії між задачами вибрано 4 задачі: Т(1), Т((P\*H - P\*H/2)), Т(P\*H/2) та Т(Р). Задачі Т(1) і Т(Р) вводять данні. Задачі Т(і) (і = 1 .. Р) виконують обчислення. Задача Т(Р) виводить результат.

Всі позначення на схемі відповідні до позначень пункту 3.1.

**3.3. Розробка програми ПРГ2**

Прогарама написана на мові С++ з використанням бібліотеки MPI та складається з одного модуля: Kurs.cpp

Модуль містить функцію main, в якій зосереджена основна логіка програми: залежності виконання задач від їх номера; також в в модулі місляться процедури задач, в яких описується логіка кожної задачі: пересилання і отримання повідомлень.

Лістинг програми ПРГ2 наведено у додатку Б.

**3.4 Тестування програми ПРГ2**

Для тестування ПРГ2 використовувалось те ж саме апаратне забезпечення, що і для ПРГ1. ЯКЕ?

В якості програмного забезпечення виступали:

* операцій система: Microsoft Windows 7 x86 SP1 (Version 6.7.7601);
* середовище розробки і компіляції C++ програми: Microsoft Visual Studio 2010 (Version 10.0.30319.1 RTMRel).

МПІ-програма запускалася за допомогою утиліти HPC Pack 2008.

Результати тестування і проведених досліджень ефективності розробленої програми наведено в таблицях 3.1 – 3.3.

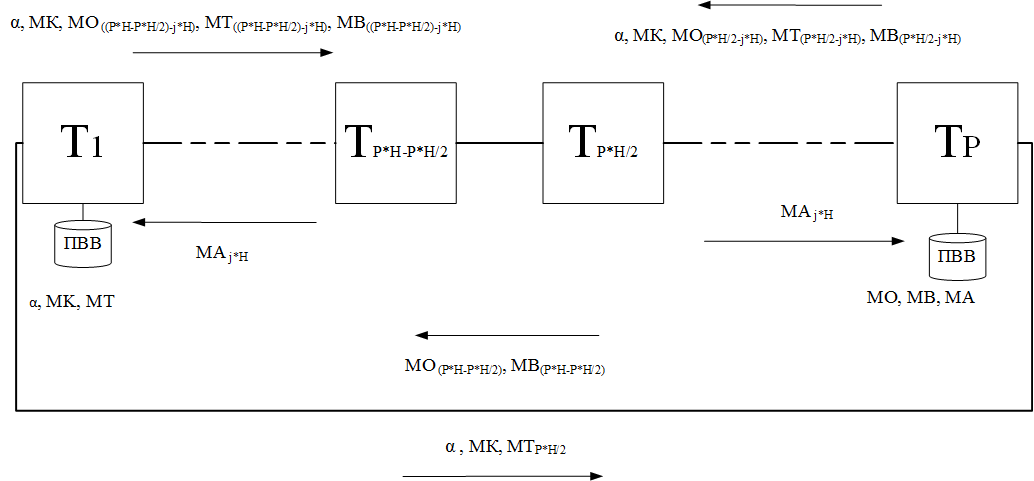


Рис 3.1Стуруктурна схема взаємодії задач для ПРГ2

НАДПИСЬ – ПОД РИСУНОК, А НЕ ЗБОКУ!!!

Таблиця 3.1 – Час виконання програми ПРГ2 ОТКУДА 16 процессоров?

ГИПЕРТРЕЙДИНГ – не пойдет!!!!

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Кількість процесорів (P) | | | | | |
| N | 1 | 2 | 3 | 4 | 10 | 15 |
| 900 | 1700,53 | 1025,75 | 620,25 | 449,11 | 989,52 | 920,20 |
| 1200 | 2200,71 | 1375,67 | 795,99 | 598,81 | 1299,20 | 1350,37 |
| 1500 | 2825,89 | 1729,58 | 1045,55 | 748,51 | 1740,76 | 1678,43 |
| 2400 | 4525,50 | 2800,01 | 1590,89 | 1120,39 | 2756,02 | 2673,92 |

На основі даних із таблиці  виконано розрахунок значень коефіцієнтів прискорення, які наведені в таблиці .

Таблиця 3.2 – Коефіцієнти прискорення для програми ПРГ2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Кількість процесорів (P) | | | | | |
| N | 1 | 2 | 3 | 4 | 10 ? | 15 ? |
| 900 | 1,00 | 1,658 | 2,742 | 3,786 | 1,719 | 1,848 |
| 1200 | 1,00 | 1,5997 | 2,764 | 3,675 | 1,694 | 1,6297 |
| 1500 | 1,00 | 1,634 | 2,703 | 3,775 | 1,623 | 1,684 |
| 2400 | 1,00 | 1,616 | 2,845 | 4,000 | 1,6396 | 1,692 |

Коефіцієнти ефективності (таблиця ) обчислено за даними таблиці .

Таблиця 3.3 – Коефіцієнти ефективності для програми ПРГ2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Кількість процесорів (P) | | | | | |
| N | 1 | 2 | 3 | 4 | 10 | 15 |
| 900 | 100,00% | 82,9% | 91,4% | 94,65% | 17,19 | 12,32% |
| 1200 | 100,00% | 79,985% | 92,13% | 91,875% | 16,94 | 10,86% |
| 1500 | 100,00% | 81,7% | 90,1% | 94,375% | 16,23 | 11,23% |
| 2400 | 100,00 | 80,8% | 94,83% | 100,00% | 16,396 | 11,28% |

Використовуючи таблиці ‑ побудовано графіки зміни коефіцієнтів прискорення і ефективності, па також зміни часу роботи програми в залежності від *N* і .

Рисунок 2.2 – Графік зміни часу виконання програми ПРГ1 в залежності від кількості ядер та значення N



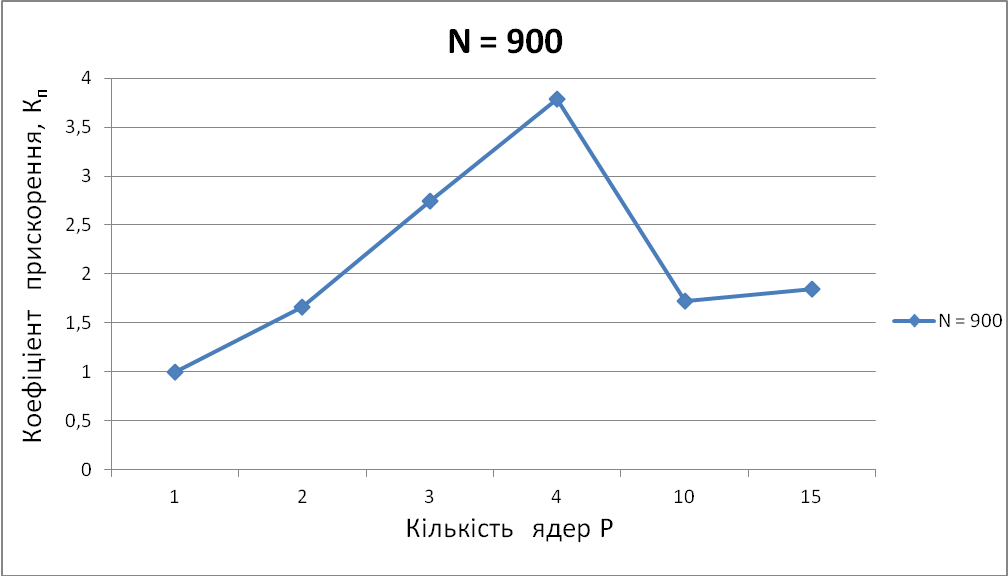
****

Рисунок 2.3 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N = 900

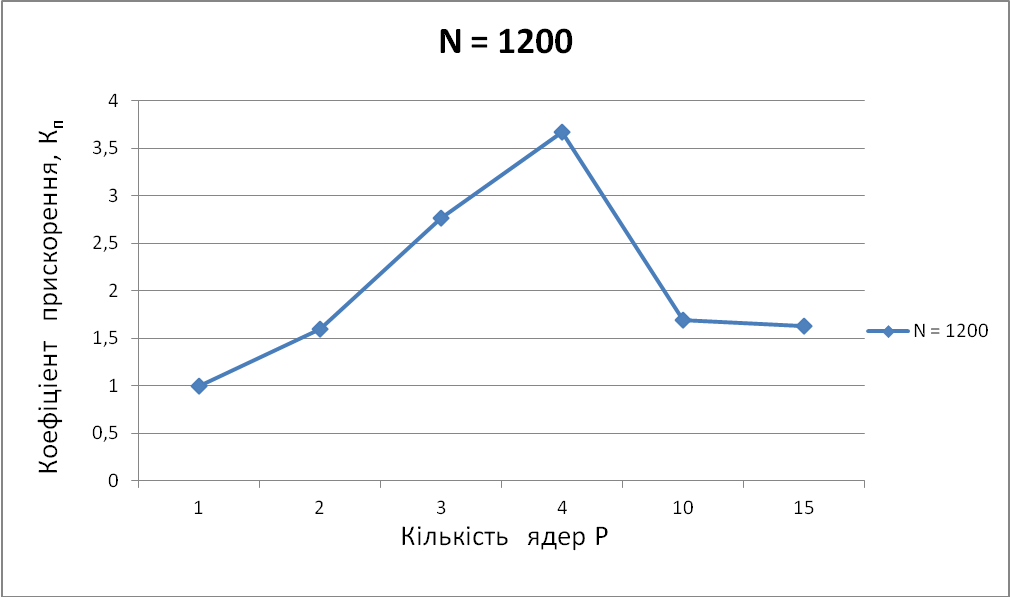
****

Рисунок 2.4 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N = 1200

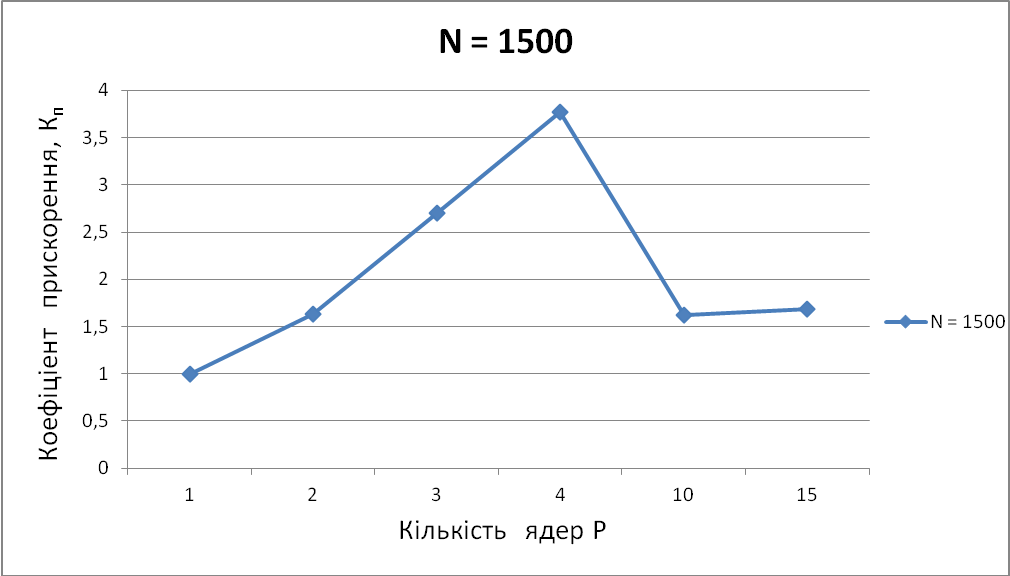
****

Рисунок 2.5 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N = 1500

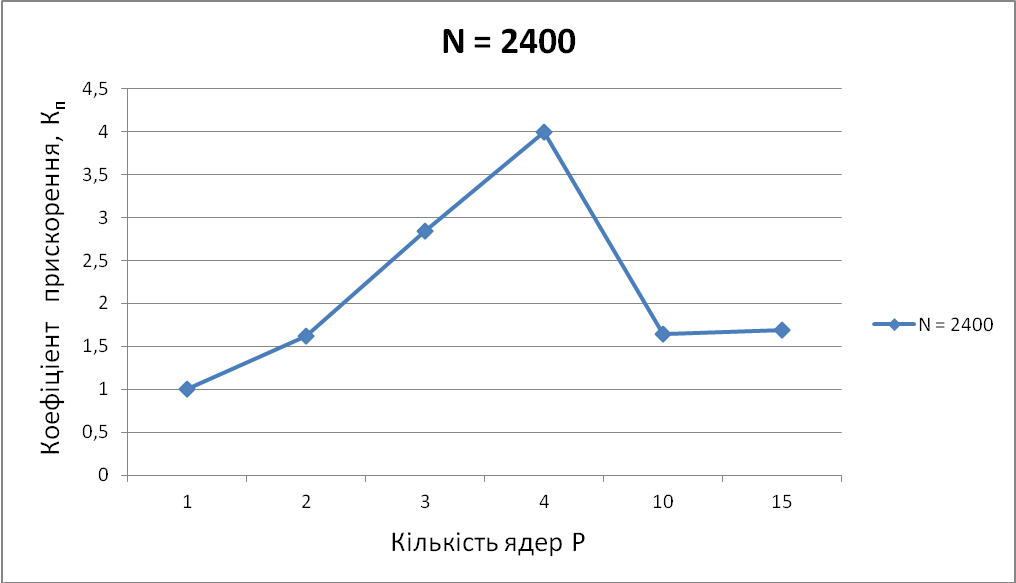


Рисунок 2.6 – Графік зміни коефіцієнту прискорення програми ПРГ1 в залежності від кількості ядер при N = 2400

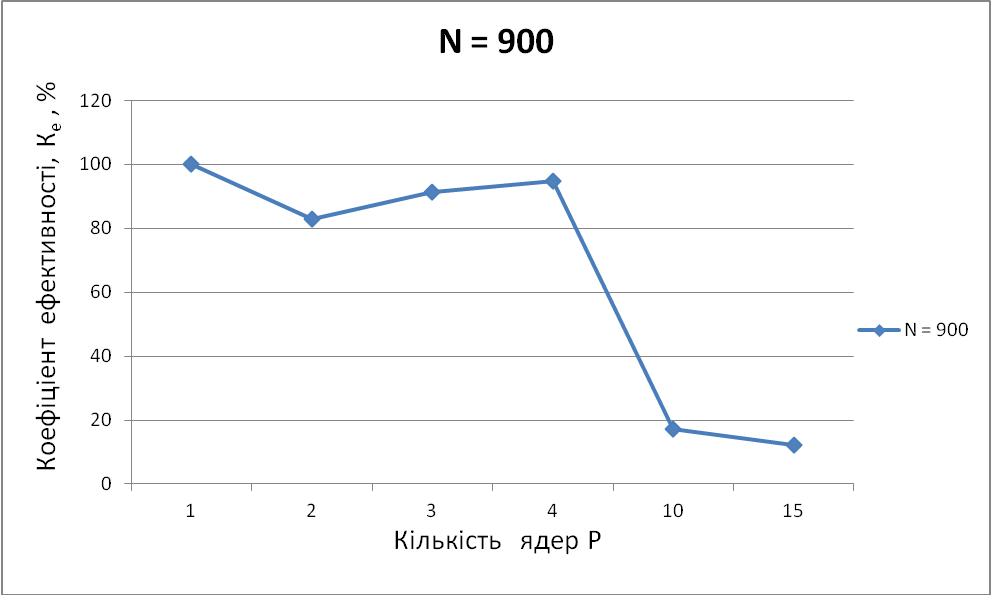
****

Рисунок 2.7 – Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N = 900

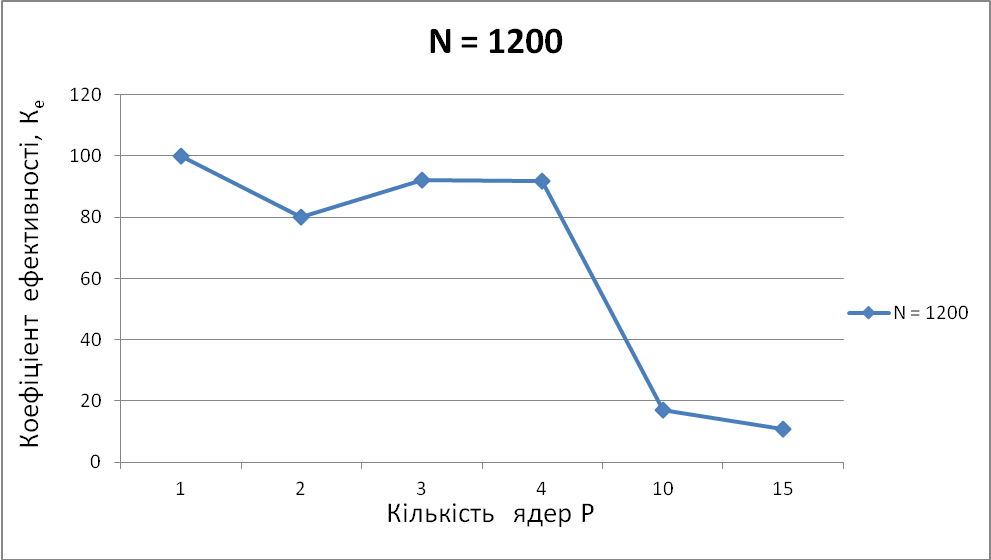
****

Рисунок 2.8– Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N = 1200

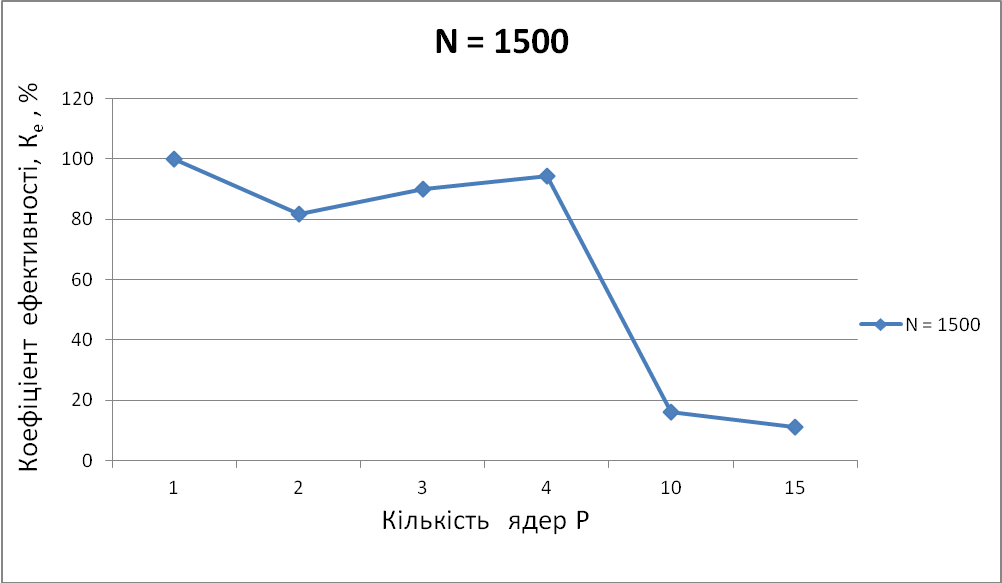
****

Рисунок 2.9– Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N = 1500

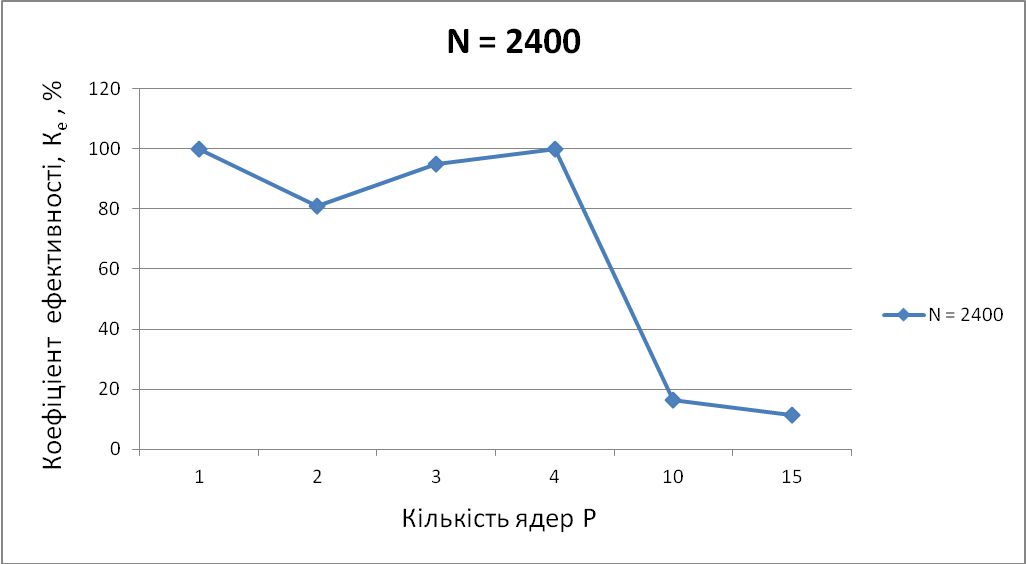


Рисунок 2.10– Графік зміни коефіцієнту ефективності програми ПРГ1 в залежності від кількості ядер при N = 2400

**3.5. Висновки до розділу 3**

Виконано розробку програми ПРГ2 для ПКС ЛП з використанням мови С++ і бібліотеки МРІ. Тестування програми показало наступне:

* Використання багатоядерної ПКС та програми ПРГ2 забезпечує скорочення часу обчислення заданої математичної задачі. Значення К*п* лежать в межах від 1,5997 до 3,786.
* Максимальне значення К*п* = 3,786 забезпечує ПКС з Р = 4 та N = 900.
* Мінімальне значення К*п* = 1,5997 виявлено у ПКС з Р = 2 та N = 1200.
* З ростом N К*п* лінійно збільшується
* Значення К*е* змінюється від 79,985% до 94,65%.
* Найефективніше програма ПРГ2 виконується ПКС з Р = 4. При цьому К*е* = 94,65%.
* При зростанні Р від 1 до 2 К*е* різко спадає до 81,5%. Далі від Р = 2 до Р = 4 лінійно зростає від 81,5% до 93,63%.
* При збільшенні кількості потоків (більше ніж процесорів) показники Кп та Ке різко зменшуються.