

УДК 681.3

О.М. Данильченко, к.т.н., доц.

С.М. Защипас, аспір.

Житомирський державний технологічний університет

ПЛАНУВАННЯ СТРУКТУРИ КЛАСТЕРНОЇ СИСТЕМИ

В статті розглядається метод планування ефективної конфігурації кластерної системи, базуючись на характеристиках заданої задачі. Наведено спосіб розв'язання практичної задачі планування за допомогою генетичного алгоритму.

1. Постановка задачі про планування кластерної системи

Розпаралелювання та виконання алгоритмів в кластерних системах звичайно призводить до значного підвищення швидкості при розв'язку широкого класу задач. Але для різних задач та різних реалізацій алгоритмів їх розв'язання, ефективність використання кластерних систем залежить як від кластерної системи, так і від параметрів задач, що на ній розв'язуються.

Важливою та невід'ємною частиною кластерної системи [1-3] виступають підсистеми складання розкладу виконання процесів на вузлах кластера.

Загальна задача складання розкладу оптимального виконання процесів відповідно до набору певних умов є NP-повною. Звичайно для розв'язання цієї задачі використовують евристичні алгоритми.

Задача складання розкладу процесів у кластерній системі може бути спрямована з метою досягнення високої продуктивності системи за допомогою мінімізації часу виконання окремої задачі, мінімізації пауз в обміні даними, максимізації утилізації системних ресурсів. В системах, що потребують обробки даних в реальному часі, складання розкладу полягає в знаходженні оптимального розподілу ресурсів для задоволення вимог кожного запиту. В умовах, коли використання кластера відбувається циклічно, з одним і тим же алгоритмом, ставиться комплексна задача планування оптимальної структури кластерної системи та водночас створення розкладу для цієї системи. Оптимальною структурою кластерної системи будемо називати таку структуру, при якій виконання заданого алгоритму не перевищує директивного терміну, а кількість вузлів такої системи є мінімальною.

Для заданої прикладної програми, історія поведінки якої може бути представлена у вигляді графа $G = (V, E)$, де $V = \{P_i, i = 1 : v\}$ – набір вузлів задачі, що представляють собою множину робочих інтервалів-процесів p_i . $v = |V|$ – кількість вузлів та E – набір дуг між цими вузлами, що являють собою залежність даних між процесами.

Причому вузли у G пронумеровані таким чином, що

$$\tau(P_{i+1}) \geq \tau(P_i), i \in [1; v], \quad (1)$$

де $\tau(P_i)$ – час початку виконання i -го процесу.

Директивний термін виконання цієї програми T^{dir} .

Необхідно визначити:

n – число вузлів у кластерній системі, що планується, $M = \{M_i, i = 1 : n\}$ – прив'язок процесів, що складають програму, до вузлів кластера $M_i = \{p_k, k \in [1; v]\}$ – упорядкований список процесів, що виконуються на i -му вузлі, Λ – часову діаграму обміну даними між вузлами.

При цьому повинні виконуватися такі умови:

– час виконання алгоритму не повинен перевищувати директивний термін:

$$T \leq T^{dir}; \quad (2)$$

– число вузлів кластера n повинно бути мінімальним:

$$n \rightarrow \min. \quad (3)$$

2. Алгоритм розв'язання задачі планування кластерної системи

Розроблений алгоритм розв'язання поставленої задачі планування базується на Simple Genetic Algorithm (SGA), який був запропонований Д. Холландом (J.H. Holland) [4].

Псевдокод генетичного алгоритму має такий вигляд:

```

begin GA
  // створення початкової популяції
  initpopulation P;
  // обчислення функції пристосованості для всіх початкових хромосом популяції
  evaluate P ;
  // цикл з перевіркою критерію закінчення алгоритму
  while not done do
    // вибір предків для створення нової популяції
    P' := selectparents P;
    // схрещування генів обраних предків
    crossover P';
    // проведення випадкової мутації
    mutate P' ;
    // обчислення функції пристосованості для хромосом нової популяції
    evaluate P' ;
    P := P';
  end do
end GA.

```

Розглянемо налагодження даного алгоритму для розв'язання поставленої задачі.

Кодування хромосоми

Для однозначного розподілу процесів по вузлах необхідно для кожного процесу p_i мати номер вузла, на якому він виконується (N_i), та його порядковий номер на цьому вузлі (k_i). Закодуємо тепер хромосому Ch таким чином, що „ген” у „хромосомі” – це процес, причому номер процесу i задається місцем гена у хромосомі, а номер вузла N_i – значенням цього гена.

$$Ch = N_1 \cup N_2 \cup \dots \cup N_v. \quad (4)$$

Відповідно до (1) порядок виконання процесу на відповідному вузлі може бути визначений однозначно шляхом сортування процесів вузла по i .

Порядок виконання процесів, призначених на один вузол, визначається в ході аналізу динаміки функціонування обчислювальної системи з використанням списочних розкладів. Формується список процесів, готових до виконання (тобто процеси, від яких вони залежать), ініціалізація процесів із цього списку здійснюється відповідно до їх пріоритетів.

Критерій пристосованості

За умовою задачі в даній роботі алгоритм повинен закінчуватися при виконанні умов (2)–(3).

Критерій пристосованості у даному випадку може бути визначений за допомогою формули:

$$K = C_1 K_T + C_2 K_P, \quad C_1 + C_2 = 1, \quad (5)$$

де C_1, C_2 – вагові коефіцієнти, від підбору яких залежить якість знайденого розв'язку; K_T, K_P – відповідно критерії оптимальності для умови (2) та для умови (3):

$$K_T = \begin{cases} \frac{T^{dir}}{T}, & T^{dir} \leq T; \\ 1, & T^{dir} > T \end{cases}; \quad (6)$$

$$K_P = 1 - \frac{n}{v}, \quad (7)$$

де n – число вузлів, використаних у даному розкладі.

Критерій зупинки

Оптимальним розв'язком поставленої задачі буде такий, для якого виконується умова:

$$n = \frac{\sum_{i=1}^v t_i}{T^{dir}}, \quad (7)$$

де t_i - час виконання i -го процесу.

Тому у випадку виконання цієї умови відбудеться зупинка генетичного алгоритму. Але можлива ситуація, коли розв'язок, для якого виконується умови (8), може не існувати взагалі. У такому випадку циклічно відбувається покращення критерію пристосованості. Якщо за задане число ітерацій покращення не відбувається, то алгоритм зупиняється, а за розв'язок береться хромосома популяції, критерій пристосованості якої є найбільшим.

Селекція

Для селекції було використано схему рулетки (Roulet-Wheel Selection) [5] з використанням принципу елітизму.

Схрещування

В алгоритмі використовується двочкова операція схрещування SGA [4]–[5].

Мутація

Операція мутації була дещо модифікована таким чином, щоб значення гена, який змінюється, не перевищував максимальну можливу кількість процесорів.

3. Результати обчислювального експерименту та висновки

Тестування алгоритму для задач розмірністю до 100 процесів показали, що шляхом підбору коефіцієнтів C_1, C_2 та ймовірності мутації цей алгоритм може бути універсально налагоджений на складні структури програм, представлені графом G , таким чином, що якісне розв'язання, близьке до оптимального, знаходиться не більш як за 1000 ітерацій.

Оптимальні параметри алгоритму обчислені під час експерименту:

розмір популяції		50
ймовірність мутації		0,9%
ймовірність схрещування		55%
вагові коефіцієнти	C_1	0,4
	C_2	0,6

Також на число ітерацій алгоритму та розв'язок впливають характеристики поставленої задачі (відхилення часу виконання процесів від середнього значення, характеристики взаємодії між процесом E).

Запропонований алгоритм може бути використаний для планування структури обчислювального кластера на основі даних про роботу заданого алгоритму та створення розкладів виконання цього алгоритму.

Даний алгоритм можна застосовувати для інших подібних задач лише шляхом зміни функції пошуку критерію пристосованості та підбору вагових коефіцієнтів.

ЛІТЕРАТУРА:

1. Larry Smarr, Charles E. Catlett. Metacomputing. Communications of the ACM, 35(6):44–52, June, 1992.
2. Что такое Beowulf? <http://www.parallel.ru/computers/reviews/beowulf.html>
3. Савяк В. Эффективные кластерные решения. <http://www.ixbt.com/cpu/clustering.shtml>

4. *Michael D. Vose*. The Simple Genetic Algorithm: Foundations and Theory. – MIT Press, Cambridge, MA, 1999.
5. *Goldberg, D.E.* Genetic algorithms in search, optimization, and machine learning. – Addison-Wesley, Reading, Mass., 1989.

ДАНИЛЬЧЕНКО Олександр Михайлович – кандидат технічних наук, доцент, завідувач кафедри програмного забезпечення обчислювальної техніки Житомирського державного технологічного університету.

Наукові інтереси:

- теорія розкладів;
- теорія складності екстремальних задач;
- паралельні обчислення.

Тел.: (0412) 418-542

E-mail: dan@ziet.zhitomir.ua

ЗАЩИПАС Сергій Миколайович – аспірант Житомирського державного технологічного університету.

Наукові інтереси:

- теорія розкладів;
- паралельні обчислення;
- операційні системи.

E-mail: serge@ziet.zhitomir.ua

Подано 11.03.2004