

АНАЛІЗ СЕРЕДОВИЩА ФУНКЦІОНУВАННЯ ПАРАЛЕЛЬНИХ БАЗ ДАНИХ

к.т.н. І.Д. Овсянніков, С.Ю. Гайдаров, С.А. Притула, О.М. Гіневський
(подав проф. А.В. Корольов)

Запропоновано підхід до аналізу середовища функціонування паралельних баз даних при використанні різних типів архітектур базової обчислювальної мережі.

Технологія паралельних баз даних дозволяє множині процесорів розділяти доступ до єдиної бази даних. Розподіл завдань по множині процесорних ресурсів та паралельне їх використання дозволяє досягти більш високого рівня пропускної спроможності транзакцій, які дозволяють підтримувати велику кількість одночасно працюючих користувачів та прискорити виконання складних запитів. Існують три різні архітектури, що підтримують паралельні бази даних.

1. Симетрична багатопроцесорна архітектура із загальною пам'яттю SMA (Shared Memory SMP Architecture). Ця архітектура підтримує єдину базу даних, що працює на багатопроцесорному сервері під управлінням однієї процесорної системи. Збільшення продуктивності таких систем забезпечується нарощуванням числа процесорів, пристроїв оперативної та зовнішньої пам'яті.

2. Архітектура із загальними (розподіленими) дисками SDA (Shared Disk Architecture). Це типовий випадок побудови кластерної системи. Ця архітектура підтримує єдину базу даних при роботі з декількома комп'ютерами, об'єднаними в кластер (звичайно, такі комп'ютери називаються вузлами кластерів), кожний із яких працює під управлінням своєї копії операційної системи. У таких системах всі вузли розподіляють доступ до загальних дисків, на яких і розташовується єдина база даних. Продуктивність таких систем може збільшуватись, як шляхом нарощування числа процесорів та об'ємів оперативної пам'яті у кожному вузлі кластера, так і за допомогою збільшення кількості самих вузлів.

3. Архітектура баз розподілення ресурсів SNA (Shared Nothing Architecture). Як і в архітектурі з загальними дисками, в цій архітектурі підтримується єдиний образ бази даних при роботі з декількома комп'ютерами, що працюють під управлінням своїх копій операційних систем. Однак, в цій архітектурі кожний вузол системи має свою опе-

ративну пам'ять та особисті диски, які не розподіляються поміж окремими вузлами системи. Практично в таких системах розділяється тільки загальний комунікаційний канал між вузлами системи. Продуктивність таких систем може збільшуватись шляхом додавання процесорів, об'ємів оперативної та зовнішньої (дискової) пам'яті в кожному вузлі, а також шляхом нарощування кількості таких вузлів.

Розглядаючи процес обробки транзакцій баз даних для запропонованих варіантів архітектури мережі, як цілого, з кінцевим числом вимог, постійним для всієї мережі, а при функціонуванні перерозподіленого у підсистемах відповідно до запропонованої архітектури, можна для його аналізу застосувати апарат замкнених мереж масового обслуговування. При цьому обов'язково повинні виконуватися наступні умови:

$$r_{i0} = r_{j0} = 0, \quad i = \overline{0, m}, \quad j = \overline{0, m},$$

де r_{i0} , r_{j0} – ймовірність виходу відповідних вимог із системи, яка містить m підсистем (вимоги не виходять з системи, а перерозподілюються по її підсистемах), матриця передач даних має розмірність $m \times m$ та інтенсивність середнього сумарного потоку на вході будь – якої підсистеми дорівнює середній сумарній інтенсивності вихідного потоку з даної підсистеми, тобто

$$\lambda_j = \sum_{i=1}^m r_{ij} \cdot \lambda_i, \quad j = \overline{0, m},$$

де λ_i - сумарна інтенсивність вимог на виході відповідної підсистеми;
 r_{ij} - ймовірність вступу вимог із i - ї підсистеми по закінченні обслуговування до j - ї підсистеми.

Дану систему у матричній формі можна записати, як $R^t \lambda = \lambda$, де R^t – транспонована матриця передач, а λ - вектор - стовпець інтенсивностей потоків вимог, що проходять крізь підсистеми у встановленому режимі, тобто

$$\lambda_j = k_j \cdot \lambda_1 \Rightarrow k_j = \lambda_j / \lambda_1,$$

де k_j - коефіцієнт передачі від джерела вимог до j - го вузла.

Для визначення ймовірностей станів замкнутої мережі у даному випадку можна використати теорему Джексона [1]:

$$p(n_1, n_2, \dots, n_m) = A \prod_{j=1}^m p_j(n_j),$$

де n_j - складові вектора стану мережі;

A - додатковий множник, який враховує замкненість при визначенні ймовірностей станів мережі.

Повна декомпозиція задачі аналізу характеристик замкненої мережі масового обслуговування неможлива, так як у мережі міститься кінцеве число вимог ($\sum_{j=1}^m n_j = N$).

Відмітимо, що

$$\sum_{q \in Q} P(n_1^{(q)}, n_2^{(q)}, \dots, n_m^{(q)}) = 1,$$

де Q – множина усіх векторів q розмірності m з невід’ємними цілими коефіцієнтами, сума яких дорівнює N .

Тоді [2]

$$A = \left[\sum_{q \in Q} \prod_{j=1}^M P_j(n_j^{(q)}) \right]^{-1},$$

Враховуючи, що $p_j(n_j) = d_j(n_j) \cdot p_j(0)$ [3], розрахуємо ймовірності:

$$\begin{aligned} P(n_1, n_2, \dots, n_m) &= \frac{\prod_{j=1}^m p_j(n_j)}{\sum_{q \in Q} \prod_{j=1}^M p_j(n_j^{(q)})} = \\ &= \frac{\prod_{j=1}^m d_j(n_j) \prod_{j=1}^m p_j(0)}{\prod_{j=1}^m p_j(0) \sum_{q \in Q} \prod_{j=1}^M d_j(n_j)} = \frac{\prod_{j=1}^m d_j(n_j)}{\sum_{q \in Q} \prod_{j=1}^M d_j(n_j)}. \end{aligned}$$

Отримані вирази дозволяють проаналізувати різні варіанти мережного середовища функціонування програмних баз даних в системах обробки транзакцій у режимі on-line.

ЛІТЕРАТУРА

1. Кофман А., Анри – Лабордер А. Методы и модели исследования операций. – М.: Мир, 1977. – 432 с.
2. Захаров В.Н., Поспелов Д.А., Хазацкий В.Е. Системы управления. – М.: Энергия, 1977. – 424 с.
3. Нечипоренко В.И. Структурный анализ систем. – М.: Сов. радио, 1977. – 216 с.