УДК 004.942(045)

Боровик В.М., канд. техн. наук

ПРІОРИТЕТНЕ УПРАВЛІННЯ ЗАПИТАМИ В МОДЕЛЯХ СЕРВЕРІВ БАЗ ДАНИХ

Інститут комп'ютерних технологій
Національного авіаційного університету

Розроблено математичну модель пріоритетного управління у системах клієнт-сервер з розподіленою обробкою інформації серверів баз даних. Дається метод та алгоритм розрахунку пріоритетів з розподіленою обробкою даних

Вступ

Ефективність моделей "клієнт-сервер" проявились тоді, коли, на відмінність від епохи мейн-фреймів, усі комп'ютерні мережі стали мати свої потужні ресурси, тому треба було так розподілити навантаження на них, щоб максимально використати ці ресурси.

Перед розробниками стали нові можливості побудови ефективних систем обробки інформації.

Мета статті

Головна концепція дослідження є побудова ефективної системи розподіленої обробки інформації для систем баз даних. Предметом нашого дослідження є моделі "клієнт-сервер" у технології баз даних. Задача нашого дослідження оптимізувати управління запитами, зосередивши увагу на практичній реалізації – алгоритму пріоритетної обробки інформації. Розглядаючи систему обробки інформації серверів бази даних (БД) як систему масового обслуговування(СМО) із пріоритетами, використуємо властивості марківських моделей обробки інформації.

Головна мета статті показати, що деяка, на перший погляд, складність оптимізації математичної моделі пріоритетної обробки, реалізується у досить простій практичній формі (алгоритму) управління такими системами.

Об'єктом дослідження є моделі "клієнт-сервер" у технології баз даних, де функції взаємодії між сервером і клієнтом розподілені на декілька частин. Як правило, це двох- та трьохрівневі моделі. Система управління базами даних (СУБД) розміщена на сервері, тому моделі файлового серверу та модель віддаленого доступу до даних, де СУБД розташовується у клієнта, не є об'єктом нашої уваги. Формально, парадигма "клієнт-сервер" застосовується до програмного забезпечення як до окремих процесів, тому і для таких систем можливе застосування наглядаємих задач, але будемо притримуватися зазначених мережних технологій.

Актуальними для цього класу моделей є моделі серверів баз даних.

Для таких моделей виконуються важливі умови існування БД:

– у кожний момент часу повинен відображатися стан предметної області, який визначається не тільки даними, але і зв'язками між ними, тобто це данні, які зберігаються у БД;

– БД повинна відображати деякі правила предметної області, по яким вона функціонує;

– потребується постійний контроль за станом БД, наглядання усіх змін та адекватна реакція на них;

– зміна ситуації в БД повинна чітко та оперативно впливати на хід прикладної задачі.

У цій роботі дається підхід побудови моделі прийняття рішення при обробці запитів клієнтів на сервері БД як СМО. Одночасно, при цьому виконується головне правило Кодда – при розподілі бази даних дозволяється розширення засобів СУБД при логічній незалежності програм користувача від бази даних.

Таку модель підтримують більшість сучасних СУБД, де є механізм зберігання процедур як засобу програмування *SQL*-серверу, механізм тригерів як механізм

наглядання поточного стану інформаційного сховища та механізм обмеження на типи даних, які називають механізмом підтримки доменної структури.

Модель активного серверу БД

В даній моделі бізнес-логіка розподілена між клієнтом та сервером. На сервері бізнес-логіка реалізована у вигляді зберігаємих процедур – спеціальних програмних модулів, які розташовані в БД та керуються безпосередньо СУБД (рис. 1):

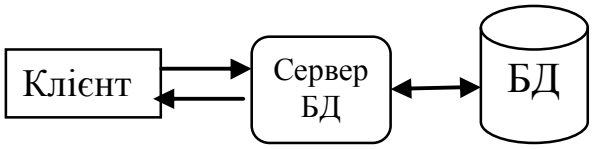

Рис. 1. Структура моделі активного серверу БД

В даній моделі сервер є активним, тому що не тільки клієнт, але і сам сервер, використовує механізм тригерів, а також може бути ініціатором обробки даних. Процедури та тригери зберігаються у словнику БД і можуть бути використані декількома клієнтами, що суттєво зменшує дублювання алгоритмів обробки даних.

Недоліком даної моделі є велика загрузка серверу:

– моніторинг подій, пов'язаних із тригерами;

– забезпечення виконання внутрішньої програми кожного тригера;

– запуск процедур по запитам клієнтів;

– повертання даних за вимогами;

– забезпечення всіх функцій СУБД тощо.

Ставиться задача, використовуя механізм обробки запитів (тригери та процедури), зменшити навантаження на клієнта, переклавши більшу частину обробки запитів на сервер БД. Такі моделі іноді називають "тонкий клієнт".

Моделі пріоритетної обробки серверів баз даних

В даній роботі розглядаються:

1) *однорівнева модель* для одного комп'ютера, при якій управління виконується між клієнтом і сервером на одному комп'ютері (рис. 2), при чому умовно на сервері розташовується СУБД, БД, а у клієнта – клієнтський додаток та управління клієнтськими додатками (програма "диспетчер"):

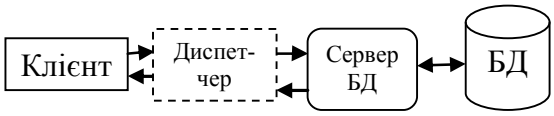

Рис. 2. Структура однорівневої моделі

2) *двохрівнена модель серверу БД* – дану модель підтримують більшість сучасних СУБД (*Informix, Ingres, Sybase, Oracle, MS SQL Server*). У цих моделях виконання додатків розподілено на два етапи, відповідно, програма-диспетчер розподілена на дві частини – обробка запитів для встановлення в чергу (клієнтська частина) та виконання запитів-додатків на сервері БД (рис. 3).

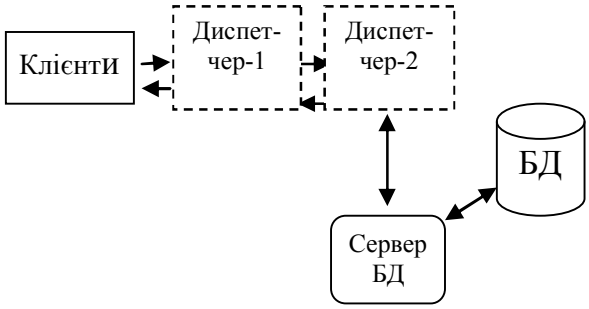

Рис. 3. Структура двохрівневої моделі

Постановка задачі

Головним математичним об'єктом досліджень даного класу моделей систем є СМО з кінцевим числом можливих станів у стаціонарному режимі. Процес керування такими системами розглядається як пошук керуючих параметрів безупинного марківського ланцюга.

Із теорії кінцевих ланцюгів Маркова відомо, що усі стаціонарні ймовірності станів $\pi(\cdot)>0$, тобто більше нуля. Оптимальне значення керуючих елементів – $\delta(\cdot)$ можуть бути рівні тільки 0 або 1. Таким чином, оптимізаційна задача зводиться до пошуку екстремуму функції:

$$L^* = f(\pi^*(\cdot)) = \max\{L'' = f(\pi(\cdot)) :$$
$$Q(\pi(\cdot), \delta(\cdot)) = 0;$$

$$0 < \pi(\cdot) \le 1;$$
$$\delta(\cdot) \in \{0,1\}; \sum_{P} \pi(\cdot) = 1\},$$

де $\pi^*(\cdot)$ – шукаємі оптимальні стаціонарні ймовірності стану системи;
Q – характеризує функціональний зв'язок стаціонарних ймовірностей стану та керуємих пріоритетів $\delta(\cdot)$;
L^* – оптимальне значення критерію ефективності;
P – множина усіх можливих стаціонарних ймовірностей стану, визначаємих типом та розмірністю моделі.

Випадковий характер вказаних показників використовується при аналізі завантаження системи. Для баз даних виконання процедури обробки запитів залежить від об'єму інформації та складності алгоритму обробки, які задіяні у кожному випадку, тому одна і таж процедура може виконуватися різний час. Крім того, пуассонівський розподіл часу доступу і обробки запитів дає критичні по завантаженню параметри системи, що гарантує успішну роботу системи у реальному часі. Запити на обробку від клієнта до серверу поступають тільки у вигляді команд ініціалізації процедур, які разом з даними знаходяться на сервері. Якщо сервер вільний (може прийняти наш запит), запит приймається на виконання. В протилежному випадку ні, то він чекає на виконання свого часу у черзі.

Розглядається ситуація управління запитами, які очікують своєї обробки. Фізична інтерпретація черги має інформаційний характер і може пояснюватися по-різному. У нашому випадку деталі цього механізму не мають принципового значення. На даний момент протоколи обробки запитів постійно ускладнюються, а загальною тенденцією є пошук оптимальних методів контролю роботи системи з урахуванням усіх можливих критеріїв. Наша задача – дати приклад рішення вибору запиту для обробки на сервері при альтернативних ситуаціях вибору.

У подальшому використовується апарат марковських процесів, який дозволяє у стаціонарних режимах роботи розробляти ясні у інтерпретації детерміновані алгоритми керування такими системами. Відомо, що випадквий процес є марковським, якщо для усіх фазових траєкторій ймовірність попадання системи у деякий наступний стан залежить тільки від того, в якому стані вона знаходиться у даний час і не залежить від попередньої історії. Будемо розглядати систему у моменти розмноження при попаданні у чергу пуассонівських вхідних потоків, маючих властивість відсутності післядії. На етапах "загибелі" (сидіння чи перескоків), тобто етапів їх обслуговування, час якого розподілений по експоненціальному закону, одержуємо ланцюг, маючий марковську властивість. Для однорідного марковського ланцюга, тобто того, де відсутня залежність стану від часу, одержуємо для неприводимого випадку (кожний стан може бути досягнений із будь-якого іншого) можливість досягнення границі $\lim_{t\to\infty} P_{ik}(t) = \pi_k$, не залежного від початкового стану ланцюга, де $P_{ik}(t)$ – ймовірність попадання системи у стан "k", якщо до того вона перебувала у стані "i". Множина $\pi_k, (k = \overline{1,K})$ є граничний розподіл ймовірностей станів, називаємих далі стаціонарними ймовірностями станів. Ланцюг у такому вигляді є ергодичним.

У нашій моделі сервер інтерпретується як "прибор", запит – як "заява", виконання процедури (запиту) – обслуговування процедури, назва процедури – тип заяви, час виконання процедури – час виконання заяви.

$\vec{I} = (i_1, i_2, \ldots, i_s, \ldots, i_n)$ – кількість заяв по кожному s-му типу у черзі. У подальшому індексований параметр "i" буде визначати кількість заяв даного типу у черзі, а без індексу – просто номер заяви. При k=0 – прибор вільний, а при $\vec{I}\,|\, i_s = 0, (s = \overline{1,n})$ – у черзі відсутні заяви s-то типу. Вводиться обмеження на загальну довжину черги: $\sum_{s=1}^{n} i_s \le R$, де R – обме-

ження на загальну чергу та окремої черги по кожному типу.

Пояснимо динаміку переходів на граф-схемі (рис. 4), на якій зображена спрощена модель для одного прибору, двох типів потоків та обмеженні в черзі по одному місцю на кожний тип потоків $M_2|M_1|r_i(i=\overline{1,2})$. Кругами відображаються усі можливі стани процесу, а стрілками – можливі перехідні інтенсивності (умовні та безумовні). Після цього для рівноважного випадку складаються рівняння усіх стаціонарних ймовірностей станів.

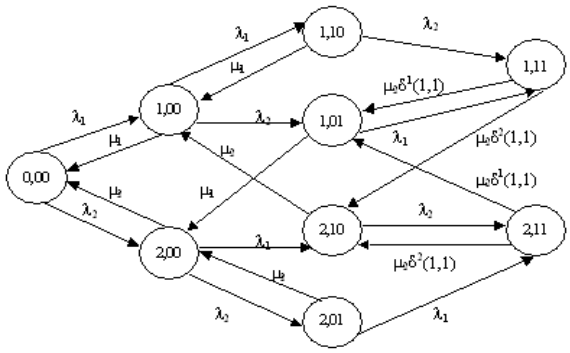

Рис. 4. Граф-схема можливих переходів

Рівняння для усіх $\pi(k,\vec{I})$:

1) $\pi(0,00)(\lambda_1+\lambda_2)=\pi(1,00)\mu_1+\pi(2,00)\mu_2$;
2) $\pi(1,00)(\lambda_1+\lambda_2+\mu_1)=$ $\pi(0,00)\lambda_1+\pi(1,10)\mu_1+\pi(2,10)\mu_2$;
3) $\pi(2,00)(\lambda_1+\lambda_2+\mu_2)=$ $\pi(0,00)\lambda_2+\pi(2,01)\mu_2+\pi(1,01)\mu_1$;
4) $\pi(1,10)(\lambda_2+\mu_1)=\pi(1,00)\lambda_1$;
5) $\pi(1,01)(\lambda_1+\mu_1)=$ $\pi(1,00)\lambda_2+\pi(2,11)\mu_2\delta^1(11)+\pi(1,11)$ $\mu_1\delta^1(11)$;
6) $\pi(2,10)(\lambda_2+\mu_2)=$ $\pi(2,00)\lambda_1+\pi(2,11)\mu_2\delta^2(11)+$ $\pi(1,11)\mu_1\delta^2(11)$;
7) $\pi(2,01)(\lambda_1+\mu_2)=\pi(2,00)\lambda_2$;
8) $\pi(1,11)[\mu_1\delta^1(11)+\mu_1\delta^2(11)]=$ $\pi(1,10)\lambda_2+\pi(1,01)\lambda_1$;
9) $\pi(2,11)[\mu_2\delta^1(11)+\mu_2\delta^2(11)]=$ $\pi(2,10)\lambda_2+\pi(2,01)\lambda_1$.

При неальтернативних ситуаціях на етапах "розмноження" поступають пуассонівські вхідні потоки заяв, при тому виконуються обмеження на число місць у черзі. У момент закінчення обслуговування неальтернативна ситуація відповідає існуванню у черзі заяв тільки одного типу. Для цієї ситуації, яка відноситься до тривіального випадку моделей "розмноження та загибелі", стан системи являє собою лінійний ланцюг. Закон вибору заяв на обслуговування у цьому випадку може бути вільним, що не змінює розподіл стаціонарних ймовірностей станів системи. Для визначеності будемо задавати, що для черги з заявами одного типу буде виконуватися дисципліна обслуговування *FCFS* – "перший прийшов – перший обслугований".

При наданні інформації "прибор вільний" може з'явитися ситуація, коли у черзі знаходяться заяви більше одного типу. Для вирішення таких конфліктних ситуацій використовується пріоритетний параметр: $\delta^s(\vec{I})$, $(s=\overline{1,n})$, який визначає можливість вибору заяви *s*-го типу для обслуговування на приборі. Звичайно, що якась заява обов'язково буде вибрана, тому:

$$\sum_{s=1}^{n} \delta^{s}(i_1, i_2, ..., i_n)\, u(i_s) = 1,$$

де $u(x) = \begin{cases} 1, \text{ ÿêùî} & x > 0, \\ 0, \text{ ÿêùî} & x \le 0. \end{cases}$

Тепер ми можемо у векторній формі записати систему рівнянь для моделі.

Нормуюча умова буде записана у такому вигляді:

$$\pi(0,\vec{0}) + \sum_{k=1}^{n}\sum_{\vec{i}} \pi(k,\vec{i}) = 1.$$

В одержаній системі рівнянь $u(\Omega^s)$ задає обмеження на число місць у черзі.

Запис $\pi(k,\vec{i}\,|i_s = i_s - 1)$ визначає перехід із стану $(k, i_1, i_2,\ldots,i_s,\ldots,i_n)$ у $(k, i_1, i_2, \ldots,i_{s-1},\ldots,i_n)$, а $\pi(k,\vec{i}\,|i_s = i_s + 1)$ із стану $(k, i_1, i_2,\ldots,i_s,\ldots,i_n)$ у $(k, i_1, i_2,\ldots,i_{s+1},\ldots,i_n)$, що означає вихід заяви із черги чи її вхід.

Критерій ефективності

Розібравши механізм зміни стану системи, ми показали можливі переходи

із одного стану у інший без урахування оцінки ефективності їх функціонування по тому чи іншому алгоритму. Для визначення оптимальності функціонування треба ввести критерії ефективності.

Одержані у результаті рішення задачі значення δ(·) і будуть шукаємими параметрами управління.

У якості критерію виберемо мінімізацію часу перебування заяв у черзі (втрата від чекання). Для визначення цього критерію введемо показник витрат із-за перебування *l*-ї заяви *s*-го типу у черзі – β_l^s. Тоді сумарні втрати у одиницю часу у системі від чекання будуть складати:

$$L = \sum_{s=1}^{n} \sum_{l=1}^{R} \beta_l^s \sum_{k=1}^{n} \sum_{\vec{i}} \pi(k, \vec{i} | i_s = l).$$

Із опису рівнянь ми бачимо , що у критеріях ефективності існує тільки лінійна залежність від π(·), а нелінійна складові (множина π(·) на δ(·)) входять в обмеження в обмеження для усіх стаціонарних ймовірностей стану системи. Для вирішення цієї проблеми використовується стандартний прийом заміни існуючих нелінійний складові, що дозволить у подальшому використати методи лінійного програмування. Введення вказаних додаткових змінних потребує й нових обмежень з урахуванням конкретної моделі.

Метод визначення оптимальних пріоритетів

Головним математичним об'єктом досліджень даної роботи є СМО з кінцевим числом можливих станів у стаціонарному режимі. Процес керування такими системами розглядається, як пошук керуючих параметрів безупинного марковського ланцюга.

Тепер треба знайти методи визначення шукаємих параметрів – оптимальних пріоритетів. Оптимальний план управління марковським ланцюгом з кінцевим числом можливих станів має детерміновану структуру, як було показано ще в роботі [4].

Таким чином, стаціонарні ймовірності π(·)>0, тобто суворо більше нуля. Важливим фактом для вибору методу є той факт, що оптимальні значення керуючих параметрів δ(·) можуть бути тільки 0 або 1. Цей параметр і буде використовуватися у програмі "Диспетчер" для однорівневої системи та у програмі "Диспетчер-2" двурівневої системи, де розглядається більш складна ситуація з пріоритетом вибору з черги на сервер (прибор).

Висновки

Розглянута методологія може бути використана для побудови пріоритетного управління запитами у системах клієнт-сервер. Результати розрахунків представляються у вигляді таблиці чи бази даних ситуаційних пріоритетів, які попередньо розраховуються по параметрам системи.

Список літератури

1. *Дейт Дж.* Введение в системы баз данных, 8-е изд. – М.: Вильямс, 2006. – 1328 с.

2. *Меликов А.З., Пономаренко Л.А., Паладюк В.В.* Телетрафик: Модели, методы, оптимизация. – К.: Политехника, 2007. – 256 с.

3. *Карпова Т.* Базы данных :модели, разработка, реализация. – СПб.: Питер, 2003. – 304 с.

4. *Wolf P., Danzig G.B.* Linear programming in Markov chain. Operation research, 1963. – Vol.10. – No.5. – P.702–710.

Подано до редакції 06.04.2010