УДК 004.3

С.Г. Стіренко

СПОСОБИ РЕАЛІЗАЦІЇ ВІДКЛАДЕНИХ ОБЧИСЛЕНЬ В КЛАСТЕРНИХ СИСТЕМАХ

Анотація: В роботі розглядаються підходи до організації відкладених обчислень в кластерних системах. Запропоновано декілька підходів до підвищення ефективності паралельних обчислень за рахунок оптимізації параметру "час передачі даних", що суттєво відображається на загальної продуктивності системи в цілому. Досліджені можливі використання різних варіантів рішень в залежності від поточних задач.

Ключові слова: відкладення обчислень, обчислювальний процес, системи з локальною пам'яттю.

Вступ

Однією з ключових задач, які необхідно розв'язати при організації обчислювального процесу на обчислювальних системах з локальною пам'яттю є організація передачі даних, які створюються в результаті одних обчислень і необхідні для виконання інших обчислень, між локальною пам'яттю різних вузлів системи. Відкладення передачі даних, що були введені або обчислені в одному вузлі [1,2], а будуть використані в іншому вузлі, є загальною ідеєю яка може бути декількома різними способами. Зокрема для її реалізації необхідно запропонувати методи визначення часу прийняття рішення про відкладення, методи визначення необхідності відкладення, методи визначення тривалості відкладення тощо. При цьому також необхідно враховувати необхідність виконання запропонованих методів та алгоритмів, що їх реалізують, в динаміці так, щоб їх виконання не займало істотну кількість часу, оскільки він буде врахований в загальний час виконання обчислень і може негативно вплинути на коефіцієнт ефективності. Розглянемо декілька відносно простих підходів до визначення цих значень.

Способи реалізації передачі даних

Підхід 1: *без відкладення передачі даних*. Цей підхід пропонується для підтвердження сумісності відкладення передачі даних з існуючими моделями програмування обчислювальних систем з локальною пам'яттю. Він не передбачає власне відкладення передачі даних, а лише передачу даних безпосередньо після їх готовності $t^{i\to j}_{comm,k} = t^{i}_{rdy,k}$. Реалізація підходу без відкладення передачі даних може застосовуватись для перевірки працездатності програмної реалізації та коректності проведення обчислень. Рішення про передачу даних приймається у момент готовності даних.

 ISSN 1560-8956

Гіпотеза 1. Застосування підходу без відкладення передачі даних знизить загальний коефіцієнт ефективності через наявність додаткових розрахунків по визначенню часу початку передачі даних.

Підхід 2: передача даних по запитам. Цей підхід є прямо протилежним до підходу без відкладення передачі даних, так як передбачає початок передачі даних одразу після того, як надійшов запит на їх використання, якщо дані вже готові на цей момент часу, або передачу даних одразу по готовності, якщо запит надійшов раніше.

$$t_{comm,k}^{i\to j} = \begin{cases} t_{req,k}^{j}, & \text{якщо } t_{req,k}^{j} \geqslant t_{rdy,k}^{i} \\ t_{rdy,k}^{i}, & \text{інакше} \end{cases}$$

Застосування цього підходу за умови наявності балансування навантаження потенційно може зменшити час очікування введення та виведення даних у ситуаціях, аналогічних до розглянутих у попередньому підрозділі. Додатковим ефектом може бути також зменшення часу безпосереднього введення та виведення даних за мережею завдяки запобіганню повторних передач даних.

Гіпотеза 2. Застосування підходу передачі даних по запитам може збільшити обсяги пам'яті, що використовуються у вузлах, які переважно відправляють дані.

Гіпотеза 3. Застосування підходу передачі даних по запитам не знижує час очікування введення та виведення даних за відсутності балансування навантаження у випадках, коли запит на використання даних відбувся пізніше ніж вони були готові (в цьому разі можна розпочати передачу у такий момент часу, щоб вона завершилась раніше запиту на використання).

Підхід 3: відкладення передачі даних на певний час. Даний підхід передбачає відкладення початку передачі даних на певний фіксований час t_d за умови, що на момент готовності даних запит на їх використання ще не надійшов. В інших випадках відкладення передачі даних лише збільшить час очікування введення та виведення, тому виконувати відкладення передачі даних може бути недоцільно.

$$t_{comm,k}^{i\to j} = \begin{cases} t_{rdy,k}^{j} + t_d, & \text{якщо } t_{req,k}^{j} > t_{rdy,k}^{i} \\ t_{rdy,k}^{i}, & \text{інакше} \end{cases}$$

Рішення про відкладення передачі даних приймається у момент готовності даних. Однак можливі варіанти підходу, з повторним відкладанням на фіксований час після виконання балансування навантаження, що вплинуло на вже відкладену передачу. В рамках цього підходу необхідно розглянути декілька способів визначення часу затримки t_d.

Спосіб 3.1: статичне задання часу затримки. Час затримки $t_d = \text{const}$ задається статично для всього процесу обчислень. Перевагою цього способу є відсутність обчислень для визначення часу t_d, однак істотним недоліком є неможливість урахування динамічних характеристик обчислень. Конкретне значення константи може бути розраховане попередньо на основі тестових запусків певного ряду обчислень.

Гіпотеза 4. Застосування відкладення передачі даних на статично заданий час затримки при малих значеннях часу затримки наближається за значенням коефіцієнту ефективності до способу без відкладення передачі даних.

Гіпотеза 5. Застосування відкладення передачі даних на статично заданий час затримки при великих значеннях часу затримки, тобто таких, що перевищують середнє значення різниці між моментом використання даних та моментом їх готовності, знижує коефіцієнт ефективності. Через те, що час початку передачі в середньому буде більше часу передачі, що збільшить сумарний час очікування введення та виведення.

Спосіб 3.2: задання часу затримки як функції від різниці середнього часу між моментом запиту даних та моментом їх готовності. При чому, доцільно розглядати лише випадки, у яких запит на використання даних отриманий пізніше ніж дані готові до використання, так як в іншому випадку середнє значення прямуватиме до нуля. Визначимо

$$t_{diff,k} = \begin{cases} t^j_{req,k} - t^i_{rdy,k}, & \text{якщо } t^j_{req,k} > t^i_{rdy,k} \\ 0, & \text{якщо інакше} \end{cases} \quad (1)$$

Тоді середня різниця

$$\hat{t}_{diff} = avg_k t_{diff,k}$$

в цьому разі можемо визначити час затримки як функцію середньої різниці $t_d = f\left(\hat{t}_{diff}\right)$

Гіпотеза 6. Якщо $f\left(\hat{t}_{diff}\right) > \max\limits_k t_{diff,k}$, то передача даних розпочнеться завжди через певний час після запиту за визначенням (1), що призведе до зниження коефіцієнту ефективності. Якщо ж $f\left(\hat{t}_{diff}\right) < agv_k t_{diff,k}$, то коефіцієнт ефективності має збільшитись за умови виконання обмеження обсягів використовуваної пам'яті.

Найбільш перспективними для дослідження функціями f обчислення часу затримки є монотонно неспадні функції на проміжку $[0; +\infty)$ з обмеженням $f(t) \leqslant \max\limits_k t_{diff,k}$. Серед таких функцій можна розглянути лінійні функції вигляду $kt + b$, нелінійні монотонні функції вигляду

$$at^k + b : k,\ b \in \mathbb{R}$$

$$a\log_k t + b:\ a,b \in \mathbb{R},\ k \in \mathbb{N}$$

$$ab^t + c:\ a,b,c \in \mathbb{R}$$

До розрахунку середнього часу різниці за формулою (1) входять лише ті передачі, які завершились до цього моменту, тому необхідність в оцінюванні часу майбутніх передач або часу завершення певних обчислень відсутня.

Гіпотеза 7. Відкладення передачі даних в перші 5-10% часу виконання обчислень недоцільне, оскільки у цей час переважно виконується передача вхідних даних та похідних від них, які досить швидко використовуються для подальших обчислень.

Оскільки обчислення виконуються в системі з локальною пам'яттю, необхідно розглянути декілька стратегій по обчисленню необхідних середніх значень. При цьому основний вплив має фактор (де-)централізованості збирання статистики. Оскільки розглядається обчислювальна система з потенційно неоднорідними зв'язками між вузлами, середній час різниці $t_{diff}^{i\to j}$ може суттєво відрізнятися для різних пар вузлів i, j, тому доцільно при визначенні часу затримки перед безпосереднім початком передачі даних t_d враховувати лише ті значення, які відповідають передачам даних між тією ж самою парою вузлів. Таким чином, кожний вузол має збирати значення t_{diff} для даних, які передавалися з та на нього, при цьому обмін цими даними між вузлами не потрібен. Останнє дозволяє також зменшити час, що витрачається на роботу інфраструктури даної системи тобто меншим чином негативно вплинути на коефіцієнт ефективності обчислень.

Підхід 4: *відкладення передачі з оцінкою часу передачі та обчислень.* Даний підхід передбачає можливість відкладення початку передачі даних на певний час t_d, аналогічно до підходу 3

$$t_{comm,k}^{i\to j} = \begin{cases} t_{rdy,k}^{j} + t_d, & \text{якщо } t_{req,k}^{j} > t_{rdy,k}^{i} \\ t_{rdy,k}^{i}, & \text{інакше} \end{cases}$$

однак час затримки перед безпосередньою передачею даних є функцією оціненої тривалості передачі та/або обчислень на вузлі, з якого передаються дані, та вузлі, на який передаються дані. В даному підході необхідно оцінювати тривалість передачі та час, через який може відбутися перше звернення до певних даних, так як точні значення на момент прийняття рішення про відкладення невідомі: запит на звернення ще не надійшов, передачу даних ще не виконано. Існує ряд методів, які дозволяють оцінити цей час [3,4], однак найбільш простим з них є застосування моделі з абстракціями задач, підзадач та пов'язаних з ними даних, запропоноване в [5].

В цій моделі вважається, що система розв'язує певну задачу T_0 (тобто послідовність обчислювальних дій), яку можна розбити на частини для їх паралельного виконання, причому кожна частина може бути розбита на менші частини рекурсивно. Для визначення частин вводиться поняття підзадача z-го порядку T_z, який характеризує кількість рекурсивних розбиттів до отримання даної підзадачі. Розбиття підзадачі визначається за допомогою оператора split

$$\mathrm{split}T_z^{j,\ldots,k} \to \left\{T_{(z+1)}^{j,\ldots,k,i}\right\},\ i \in (1, N_{(z+1)}^{j,\ldots,k}) \tag{2}$$

де $N_{(z+1)}^{j,\ldots,k}$ – кількість підзадач, на які була розбита вихідна підзадача $T_{(z+1)}^{j,\ldots,k}$.

Послідовне застосування оператора розбиття до підзадач, представлених у вигляді множини послідовностей дій, призводить до формування у кожної підзадачі послідовності Γ: $T_{z)}^{\gamma_1,\gamma_2,\gamma_3,\cdots}$, довжина якої визначає порядок підзадачі $z = \dim\Gamma$, а кожний i-тий елемент послідовності характеризує номер підзадачі i-го порядку, з якої було отримано дану підзадачу. Послідовність Γ називається шляхом розбиття. Для вихідної задачі, яку для збереження спільності можна вважати підзадачею нульового порядку, шлях розбиття є пустою послідовністю $\langle\rangle$.

Використання подібної моделі програмування дозволяє при прийнятті рішення про відкладення передачі даних розглядати інформацію про порядок підзадачі та шлях розбиття, використовуючи яку можна формулювати оцінку часу виконання обчислень. Для оцінки часу виконання передачі даних, можна використовувати як обсяг інформації, що передається, так і близькість відповідних шляхів розбиття и застосуванні даного підходу можна розглянути декілька способів обчислення часу затримки перед безпосереднім відправленням даних.

Спосіб 4.1. Виконати відкладення на максимально можливий час мінус оцінений час передачі. Для цього необхідно оцінити час передачі поточних даних та час, у який відбудеться запит на їх використання. Для оцінки часу запиту на використання певних даних необхідно визначити час початку обчислень, що їх використовують. В моделі з використанням підзадач, цей час може бути оцінений як сума часу виконання всіх підзадач, що знаходяться в черзі перед тією, що використовує поточні дані $\sum t_c^l$. Причому час виконання підзадачі може бути оцінений як функція часу виконання всіх підзадач на цьому вузлі з урахуванням їх шляху розбиття $t_c^{\Gamma^l} = f(t_c^{\Gamma^{l-1}}, t_c^{\Gamma^{l-2}}, \ldots, t_c^{\Gamma^1}, \Gamma^{l-1}, \Gamma^{l-2}, \ldots, \Gamma^1)$. Можна вважати, що час виконання обчислень підзадач однакового порядку підпорядкований нормальному розподілу $N(\mu, \sigma^2)$ за центральною граничною теоремою, оскільки на нього впливає велика кількість фа-

кторів, де математичне очікування μ є зваженим середнім арифметичним

$$\mu = \frac{\sum_{l=1}^{L} d_{\Gamma,\Gamma^l} t_c^{\Gamma}}{\sum_{l=1}^{L} d_{\Gamma,\Gamma^l}}$$

середньоквадратичне відхилення σ є зваженою характеристикою

$$\sigma = \sqrt{\frac{1}{\sum_{l=1}^{L} d_{\Gamma,\Gamma^l}} \sum_{l=1}^{L} d_{\Gamma,\Gamma^l} (t_c^{\Gamma^l} - \mu)^2}$$

$t_c^{\Gamma^1}$ – відомий час виконання обчислень підзадачі із шляхом розбиття Γ^l; L – кількість підзадач, що були обчислені на даний момент; $\Gamma = \langle \gamma_i \rangle$ – шлях розбиття підзадачі, час виконання якої оцінюється; d_{Γ,Γ_l} – ваговий коефіцієнт, що обчислюється за формулою

$$d_{\Gamma,\Gamma_l} = b^{\dim\Gamma - \dim\Gamma_l} \sum_{i=1}^{\min(\dim\Gamma - \dim\Gamma_l)} \delta(\gamma_i - \gamma_i^l).$$

δ – функція Дірака, b – коефіцієнт розгалуження задач на підзадачі, тобто середня кількість підзадач після розбиття. Вагова функція враховує той факт, що близькі за шляхом розподілу підзадачі найбільш ймовірно мають близький час виконання обчислень для більшості задач. При цьому, якщо порівняння виконується з підзадачею меншого порядку, її час розрахунків необхідно розділити на відповідну кількість підзадач більшого порядку, а у випадку порівняння з підзадачею більшого порядку – помножити на відповідну кількість підзадач, що враховується показниковою функцією коефіцієнту розгалуження.

Оцінка часу запиту k-го блоку даних $t_{req,k}^j$ береться як сума випадкових значення з нормальним розподілом $N_C^j(\mu,\sigma)$ для всіх підзадач, що мають бути виконані j-тим вузлом. Аналогічним чином визначається нормальний розподіл для оціненого часу передачі даних $N_T^{i\to j}(\mu,\sigma)$, за яким вибирається оцінка часу передачі даних $t_T^{i\to j}$. Після чого час затримки початку передачі даних визначається як

$$t_d = t_{req,k}^i - t_{T,k}^{i\to j}$$

Гіпотеза 8. Відкладення на максимально можливий час з використанням оцінки часу за нормальним розподілом може зменшувати коефіцієнт ефективності у випадку задач, що розбиваються на підзадачі з обсягами обчислень, що підпорядковуються деякому періодичному закону.

Спосіб 4.2 Виконувати передачу в середині проміжку $[t_{rdy,k}^i, t_{req,k}^j]$, тобто

$$t^{i\to j}_{comm,k} = \begin{cases} t^{j}_{req,k} - t^{i}_{rdy,k}, & \text{якщо } t^{j}_{req,k} > t^{i}_{rdy,k} \\ t^{i}_{rdy,k}, & \text{інакше} \end{cases}$$

При цьому оцінка часу, у який відбудеться запит на використання даних виконується з використанням нормального розподілу аналогічно до попереднього способу.

Гіпотеза 9. Застосування відкладення передачі даних до середини оціненого проміжку очікування демонструє більші значення коефіцієнта ефективності, аніж відкладення на максимально можливий час для випадків, в яких останній є малоефективним через нерівномірність обсягів обчислень.

Для більш точних оцінок часу майбутнього запиту на використання даних та часу передачі даних між вузлами, існує можливість в рамках даного підходу використати різні ймовірнісні моделі, зокрема графові. Крім того, для визначення параметрів цих моделей можна застосувати широко вивчені на даний момент методи машинного навчання [6].

Слід зазначити, що після виконання балансування навантаження або відміни частини обчислень, необхідно заново розглянути можливість відкладення даних для обох запропонованих способів, оскільки оцінка часу передачі даних та часу запиту на їх використання могла змінитися через зміну вузла, на якому будуть виконуватись обчислення.

Висновки

Запропоновано ряд підходів до визначення часу початку передачі даних. Серед них слід відзначити два підходи, які є відповідно нижнім і верхнім обмеженням на доцільні значення часу початку передачі даних: передавати дані відразу по готовності, тобто фактично на виконувати відкладення, та передавати дані за запитом, тобто відкладати передачу на максимально можливий термін. Обмеження знизу є гарантією збереження коректності обчислень, оскільки дані будуть підготовані перед їх пересилкою для використання, а вихід за обмеження зверху через необхідність додаткового очікування після запиту на використання даних вносить у час виконання затримку, під час якої обчислення не виконуються, що в результаті знижує коефіцієнт ефективності. Запропоновано також два підходи до більш гнучкого обчислення часу початку передачі даних, один з яких базується на статистичній обробці попередніх значень часу очікування введення та виведення та обсягів пам'яті під час розв'язання даної задачі, а інший – на ймовірнісній оцінці часу першого запиту даних та часу, необхідного для передачі цих даних між вузлами. В рамках кожного з підходів може бути запропоновано декілька різних способів визначення невідомих величин, а саме можна змінювати алгоритми статистичної обробки або ймовірнісні моделі прогнозування.

Висунуто ряд гіпотез про роботу запропонованих підходів взагалі, які дозволять експериментально підтвердити коректність пропозицій та тверджень. Конкретні значення певних параметрів, що застосовуються у запропонованих способах також мають бути визначені емпіричним шляхом та можуть бути замінені або покращені під час роботи програми із застосуванням методів машинного навчання.

Біблиографічний список

1. Стіренко С.Г. Організація відкладених обчислень в кластерних системах / С.Г. Стіренко // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. – 2013. –№. 59. – С. 12–17.
2. Стіренко С.Г. Модель технології програмування паралельних систем / С.Г. Стиренко // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. – 2013. –№. 58. – С. 158-164.
3. Tanenbaum, Andrew S. Distributed systems. Principles and paradigms. / Andrew S Tanenbaum, Maarten van Steen. New Jersey, USA : Upper Saddle River, NJ: Pearson/Prentice Hall, 2007. – 1296 p.
4. Hennessy, John L. Computer architecture: a quantitative approach / John L Hennessy, David A Patterson. – Amsterdam, Noord-Hooland, Netherlands : Elsevier, 2011. – 824 p.
5. Представление задач в системах распараллеливания с изменяемой зернистостью / Г. М. Луцкий, С. Г. Стиренко, А. И. Зиненко, Д. В. Грибенко // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. – 2012. – Т. 55. – С. 212–216.
6. Russell, S.J. Artificial intelligence: a modern approach / S.J. Russell, P. Norvig. Prentice Hall series in artificial intelligence. – New York, NY, USA: Pearson Education/Prentice Hall, 2010.

Отримано 26.10.2013 р.