

2. Торстейнсон П. Криптография и безопасность в технологии.NET/ П.Торстейнсон Г.А. Ганеш; пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2007. – 479 с.
3. Просиз Дж. Программирование для Microsoft.NET/ Дж. Просиз; пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2009. – 704 с.
4. AspectC++. [Електроний ресурс] // Режим доступу: <http://www.aspectc.org>.
5. Нгуен Ван Доан. Средства аспектно-ориентированного программирования для разработки Web-приложений в системе Aspect.NET/ Нгуен Ван Доан, В. О. Сафонов // Вестн. С.-Петербург. ун-та. Сер. 10. –2011. – Вып. 1. – С. 85-105.

Сімоненко В.П.

доктор технічних наук, професор;

Баль В.В.

магістр,

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ОЦІНКА УЗГОДЖЕНОСТІ ІНФОРМАЦІЇ В GRID СИСТЕМАХ

Інформаційна система Grid вирішує питання, які можуть зажадати розгляду всіх джерел інформації (Grid-сервісів), які широко поширені географічно, з тим щоб забезпечити ефективні функції Grid, які можуть використовувати кілька взаємодіючих сервісів. В основному це може бути досягнуто шляхом переміщення запиту до даних (відправлення запиту) або переміщення даних в запит (відправка даних).

Фундаментальний аспект розподілених обчислень є необхідність отримання інформації про структуру і стан мережевих послуг в Грід системах, які широко поширені географічно. Інформація, що описує кожен мережеву послугу Грід системи забезпечується само мережевою службою і, отже є основним джерелом інформації. Інформація відповідає інформаційній моделі яка представляє основні концепції служби Грід системи і відносин між ними, щоб чітко визначити їх семантику. Передбачається, що зміст джерела інформації є актуальним, тобто значення відображають реальний стан Грід системи.

Необхідно дозволити запити, які можуть враховувати деякі або всі джерела інформації, щоб забезпечити ефективні функції Грід системи, які можуть використовувати кілька взаємодіючих служб. Загалом, інформація в Грід системах надзвичайно розподілена, тобто кожен кортеж (екземпляр об'єкта) знаходиться в іншому місці [8]. Запити не є складними з точки зору структури, не більше одного предиката і немає об'єднань, тому результат зазвичай є підмножиною загальної інформації (наприклад, знайти найближчий обчислювальний сервіс для служби зберігання). Метою є мінімізація часу відповіді на запит для виконання багатьох запитів від багатьох клієнтів для багатьох джерел інформації (тобто для вирішення проблеми масштабованості).

2 Виконання запитів інформаційної грід системи

2.1 Вихідні дані

Завдання глобальних інформаційних систем були викладені давно, в статті [10], в якій стверджувалося, що масштабованість, автономність і (не) доступність джерел інформації дозволять використовувати методи вибору грубої сили, вичерпний пошук і глобальну синхронізацію (узгодженість). Неможливо для великої кількості запитів з -за великих затримок і необхідної пропускної здатності. Було також зазначено, що недоступність також є проблемою в міру збільшення масштабів, оскільки ймовірність відмови принаймні по одному компоненту різко зростає, що означає, що принаймні деякі з джерел інформації будуть недоступні, якщо їх буде багато. Через складність глобальної синхронізації вимоги до узгодженості можуть бути ослаблені і, крім того, оскільки джерела інформації можуть бути недоступні, результати можуть бути неповними. Різні компроміси між усіма цими проблемами можуть існувати в різних налаштуваннях і умовах.

У типовій архітектурі обробки запитів інформація, пов'язана з розташуванням джерел інформації, зберігається в каталозі [6]. У середовищі Grid каталог являє собою реєстр або індекс служби, який надає список сервісів разом з їх місцем розташування у вигляді URL (службового запису) [2]. Інші атрибути, такі як тип сервісу, можна використовувати, щоб уникнути звернення до служб, які можуть не мати відношення до запиту, однак для глобального запиту вибору потрібно тільки URL-адресу, що ідентифікує джерело інформації. Слід підкреслити, що інформація про те, які сервіси мають і не повинні знаходитися в інфраструктурі, виходить не від самої служби. Авторитетне джерело інформації про те, які служби беруть участь в інфраструктурі і до якого адміністративного домену вони належать, залежить від політик управління грід-інфраструктурою. Право власності і контроль над такою інформацією має першорядне значення для глобальної інфраструктури і того, як ця інформація управляється, є питанням політики управління. Таким чином, передумовою інформаційної системи Grid є наявність каталогу у вигляді реєстру служби, який є результатом тих політик управління, які можуть бути використані для виявлення всіх джерел інформації в грід-інфраструктурі [2]. Подробиці про те, як ці політики використовуються для створення реєстру служби, виходять за рамки цієї статті, але це область, яка буде корисна для подальших досліджень.

Після того, як джерела інформації були виявлені, запит все одно повинен бути виконаний. Виникає питання: перемістити запит в джерело інформації (відправлення запиту) або перенести інформацію в запит (перенесення даних). Можна використовувати два підходи до відправки даних з бази даних [6]; Кешування і реплікації. Кешування ініціюється клієнтом і, в нашому контексті, запускається при виконанні запиту; Він зазвичай використовує синхронний протокол, заснований на визнанні недійсності [6], помилковим підходом. Реплікація ініціюється сервером і, в нашому контексті, ініціюється самим джерелом інформації і виконується, навіть якщо ніякі запити не обробляються; Він спрямований на підтримку узгодженості квазі-копії [2] інформації з

використанням асинхронного протоколу. У загальному випадку реплікація намагається перемістити інформацію поруч з великою групою клієнтів, щоб вони могли отримати доступ до неї по дешевці при першому запиті [1]. Кешування дозволяє клієнту отримати доступ до інформації швидше, коли ця інформація використовується повторно [4]; Перший запит буде відчувати накладні витрати, в той час як всі інші подальші запити будуть корисні. У більшості випадків витратна сума набагато менше потенційної вигоди, коли є багато запитів, які можуть принести користь

2.2 Час відповіді на запит

Час відповіді на запит – це час, витрачений на відправку запиту клієнтом в джерело інформації, коли результат запиту було повернуто назад. Оскільки час відповіді на запит може бути пов'язано з конкретними проектними рішеннями, такими як використовуваний мову запитів і структура даних кеша, його необхідно виміряти. Одним з факторів, що впливають на продуктивність запиту і його час відгуку, є кількість запитів, які можуть виконуватися одночасно; Тому цей аспект також варто вивчити. В якості захисту від запитів, які займають занадто багато часу або зависають нескінченно, для кожного запиту зазвичай вказується таймаут. Відсутність стає великою проблемою, оскільки масштаби ростуть; Очікується, що кілька джерел інформації можуть бути недоступні, якщо їх багато. Отже, у міру збільшення кількості джерел збільшується ймовірність того, що конкретний запит буде тайм-аут за даний період часу. Це говорить про те, що для великої кількості джерел інформації значення тайм-ауту може значно вплинути на продуктивність. В результаті середній час відгуку запиту для паралельних запитів повинно бути знайдено емпірично і буде вимірюватися для різних сценаріїв.

2.3 (α, β) –валютність

Якість інформації, що повертається може бути кількісно оцінена за допомогою метрики (α, β) – валютність, яка забезпечує ймовірність α , що випадково вибране значення об'єкта, що зберігається в інформаційній системі Grid, є поточним щодо пільгового періоду β . Ймовірність α може бути розрахована з використанням (1), де β – вік інформації, а λ – константа для типу об'єкта.

$$\alpha = e^{-\lambda\beta} \quad (1)$$

У разі доставки запиту значення для часу відгуку запиту може бути використано для градієнтного періоду β в рівнянні (α, β) . У разі доставки даних значення β в (α, β) – валютність рівнянні є затримка між зміною джерела інформації та оновленням інформаційної служби.

У разі кеша, де є обмеження часу, протягом якого дані можуть знаходитися в кеші (Time To Live – TTL), мінімальне значення α визначається значенням TTL. Отже, для обчислення оптимального значення TTL можна використовувати концепцію (α, β) – валютність, щоб α не зменшується нижче бажаного значення. Для заданого рівня якості α максимальний TTL β можна розрахувати, використовуючи (2).

$$\beta = \frac{\ln \alpha}{-\lambda} \quad (2)$$

3 Висновок

У цій статті були розглянуті основні концепції обробки запитів в середовищі Grid для визначення різних підходів. Основна проблема рішення запиту за багатьма джерелами інформації – проблема пошуку даних, яка широко вивчалася в літературі по обробці розподілених запитів. Існує два основні підходи до розподіленої обробки запитів; Відправлення запиту (переміщення запиту до даних) і відправка даних (переміщення даних в запит). Не дивно, що час відгуку запитів нижче для підходу до відправки даних, ніж для підходу доставки запитів, хоча фактична різниця в продуктивності може залежати від інших факторів, включаючи реалізацію. Навіть при розгляді (α, β) – валютність через час, витраченого на розподілений запит, воно не обов'язково дає певні переваги, і, отже, підхід до відправки даних повинен бути прийнятий.

Список використаних джерел:

1. Aiftimieci, C., Aimar, A., Ceccanti, A., Cecchi, M., Meglio, A.D., Estrella, F., Fuhrmam, P., Giorgio, E., Konya, B., Field, L., Nilsen, J.K., Riedel, M., White, J.: Towards next generations of software for distributed infrastructures: the european middleware initiative. In: Proceedings of the 8th International Conference on E-Science, pages 1–10, Chicago, IL, USA (2012).
2. Neocleous K., Dikaiakos MD., Fragopoulou P., et al. Failure Management in Grids: the Case of the EGEE Infrastructure // Parallel Processing Letters. – vol. 17. – 2007. – PP. 391-410.
3. Alonso, R., Barbara, D., Garcia-Molina, H.: Data caching issues in an information retrieval system. ACM Trans. Database Syst. 15(3), 359–384 (1990).
4. Andreozzi, S., Burke, S., Field, L., Litmaath, M.: GLUE schema version 1.3.
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. Addison Wesley, Harlow, England (1999).
6. Cancio G. The Datagrid architecture version 2 / G. Cancio, C. Steve, T. Folkes, S. Fisher, W. Hoschek, D. Kelsey, T. Folkes // EDMS 439938, CERN. – 2001. Режим доступа: <http://acs.lbl.gov/~hoschek/publications/edg-architecture.pdf>.
7. Bird, I.: Computing for the large hadron collider. Annu. Rev. Nucl. Part. Sci. 61(1), 99–118 (2011).
8. Bird, I., Jones, B., Kee, K.F.: The organization and management of grid infrastructures. Computer 42(1), 36–46 (2009).
9. Casey, J., Cons, L., Lapka, W., Paladin, M., Skaburskas, K.: A messaging infrastructure for WLCG. J. Phys. Conf. Ser. 331(6), 062015 (2011).
10. Ordille, J.J., Miller, B.P.: Database challenges in global information systems. In: Proceedings of the ACM SIGMO international conference on Management of data, pages 403–407, Washington, D.C., United States, p. 1993 (1993).