УДК: 004.738.5:004.056.53

А. В. Мельник, к.п.н.
Житомирський державний університет ім. І.Франка, Житомир

ОПТИМІЗАЦІЯ ЕФЕКТИВНОСТІ КЕШУВАННЯ В ІНФОРМАЦІЙНО-ОРІЄНТОВАНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ АНАЛІЗУ ТРАФІКУ І ВИКОРИСТАННЯ АДАПТИВНИХ АЛГОРИТМІВ

Інформаційно-орієнтовані мережі зазвичай містять велику кількість вузлів та ресурсів, що надають послуги, які користувачі можуть запитувати. Один із способів покращити продуктивність таких мереж - це використання кешування, що дозволяє зберігати копії даних вузла в локальній пам'яті та використовувати їх для задоволення майбутніх запитів. Проте, існує декілька факторів, які можуть погіршити ефективність кешування в інформаційно-орієнтованих мережах, таких як великий обсяг даних, що потребують кешування, нестабільність запитів, а також несприятливі зміни у мережі.

У літературі існує велика кількість адаптивних алгоритмів кешування, які дозволяють змінювати стратегію кешування в залежності від змінних умов мережі та даних. Розглянемо деякі з них:

1. Алгоритм кешування з часовими мітками (Time-based cache algorithm) Цей алгоритм використовує часові мітки для визначення того, коли дані востаннє використовувалися. Дані, які були використані нещодавно, зберігаються в кеші, а ті, які не використовувалися довший час, видаляються. Один з недоліків цього методу - він не враховує популярність даних, тому дані, які були запитані менше, можуть бути видалені надто рано.

2. Адаптивний алгоритм кешування LRU-K (Least Recently Used K) - цей алгоритм використовує історію використання даних для прийняття рішення про збереження або видалення даних з кешу. Він враховує не лише останній запит, а й K попередніх запитів. Це дозволяє покращити точність передбачення попиту на дані, зберігати більш популярні дані в кеші та видаляти менш популярні.

3. Адаптивний алгоритм кешування LFU-K (Least Frequently Used K) - цей алгоритм використовує історію використання даних для визначення їх популярності. Дані, які були запитані менше, видаляються з кешу, а більш популярні зберігаються. Аналогічно

до LRU-K, LFU-K враховує K попередніх запитів для точнішого визначення популярності даних.

4. Адаптивний алгоритм з попереднім прогнозуванням (Adaptive Pre-fetching algorithm) Цей алгоритм передбачає, які дані будуть запитані в майбутньому, та зберігає їх у кеші. Це дозволяє зменшити час на обробку запитів, оскільки запитані дані вже знаходяться в кеші. Однак, якщо прогнозування неправильне, це може призвести до зайвого використання пам'яті та зменшення ефективності кешування.

5. Адаптивний алгоритм з резервуванням (Adaptive Reservation algorithm) Цей алгоритм зарезервовує деяку кількість пам'яті для найбільш популярних даних та використовує LRU-стратегію для менш популярних даних. Це дозволяє зберігати найбільш важливі дані в кеші, тоді якщо пам'ять стає обмеженою, менш популярні дані будуть видалені першими. Однак, якщо популярність даних змінюється динамічно, то цей алгоритм може виявитися менш ефективним.

6. Алгоритм кешування зі змінним розміром кешу (Variable-size cache algorithm) Цей алгоритм дозволяє змінювати розмір кешу в залежності від популярності даних. Якщо деякі дані стають більш популярними, то розмір кешу збільшується, щоб зберегти ці дані. Це дозволяє ефективніше використовувати доступну пам'ять. Однак, цей алгоритм може бути витратним з точки зору обчислювальних ресурсів, оскільки потребує частого перерахування розміру кешу.

Кожен з цих алгоритмів має свої переваги та недоліки і вибір певного алгоритму залежить від конкретних умов та вимог до системи кешування.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Adaptive probabilistic caching technique for caching networks with dynamic content popularity URL: https://www.sciencedirect.com/science/article/abs/pii/S0140366418302925?via%3Dihub1 (дата звернення 12.03.2023)

2. Cache Replacement Algorithms: How To Efficiently Manage The Cache Storage URL: https://dev.to/satrobit/cache-replacement-algorithms-how-to-efficiently-manage-the-cache-storage-2ne1 (дата звернення 02.03.2023)