

АЛГОРИТМИ ОПТИМІЗАЦІЇ ОБРОБКИ ВЕЛИКИХ ДАНИХ У СИСТЕМАХ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ

Хоменко А.О., науковий керівник к.фіз.-мат.н., доцент Кириченко В.В.

У сучасних інформаційних системах, що працюють з екологічними даними, надзвичайно важливим є забезпечення швидкого та ефективного доступу до великих обсягів інформації. Особливо це актуально для систем моніторингу якості атмосферного повітря, які збирають, обробляють та відображають значні масиви даних у реальному часі.

Одним з базових методів є кешування. Використання кешу дозволяє зменшити кількість звернень до бази даних, завдяки збереженню результатів запитів у пам'яті. У контексті Python/Django [1] це можна реалізувати за допомогою декораторів, як-от `@cache_page`, а для збереження — використовувати NoSQL-рішення, наприклад, Redis. Утім, кешування має і недоліки — можливу втрату кешованих даних через обмеження пам'яті, а також проблеми з актуальністю інформації при несвоєчасному оновленні кешу.

Застосування Redis як кеш-сховища є поширеною практикою завдяки його високій швидкодії та підтримці різних типів структур даних. У системі моніторингу якості повітря Redis дозволяє кешувати запити до найбільш завантажених маршрутів, наприклад, до головної сторінки з актуальними вимірюваннями. Такий підхід дає змогу зменшити навантаження на SQL-сервер, знизити час відповіді й підвищити стабільність сервісу в умовах великої кількості одночасних запитів.

Однак кешування слід використовувати обережно. Якщо обсяг кешованих даних перевищує доступну оперативну пам'ять, Redis [4] починає видаляти старі записи згідно з політикою витіснення, що може призвести до втрати критичних даних. Крім того, важливо налаштувати механізми очищення або оновлення кешу при зміні базових даних, аби уникнути ситуацій, коли користувачі отримують застарілу інформацію.

Ще одним важливим підходом є попередня агрегація даних. Вона передбачає попередній розрахунок агрегованих значень (середніх, сумарних) на рівні години чи дня, що зменшує навантаження на систему в момент запиту. Реалізація попередньої агрегації можлива за допомогою планувальників завдань, таких як `cron` або `Celery periodic tasks` [2].

Для підвищення надійності можна комбінувати підхід попередньої агрегації з кешуванням, кешуючи вже агреговані дані. Це дозволяє одночасно зменшити обсяг обчислень під час запиту та забезпечити швидкий доступ до результатів. Також слід запровадити механізми моніторингу запланованих завдань, щоб уникнути ситуацій з несвоєчасним оновленням даних у кеші.

На рисунку 1 наведено порівняльну діаграму ефективності використання різних методів оптимізації. Згідно з отриманими результатами, кешування дозволяє скоротити час відповіді сервера з 300 мс до 150 мс та знизити навантаження з 100% до 70%. Попередня агрегація демонструє ще кращі показники — 100 мс та 60% відповідно. Найвищу ефективність забезпечує поєднання обох методів: відповідь — лише 50 мс, навантаження — 30%.

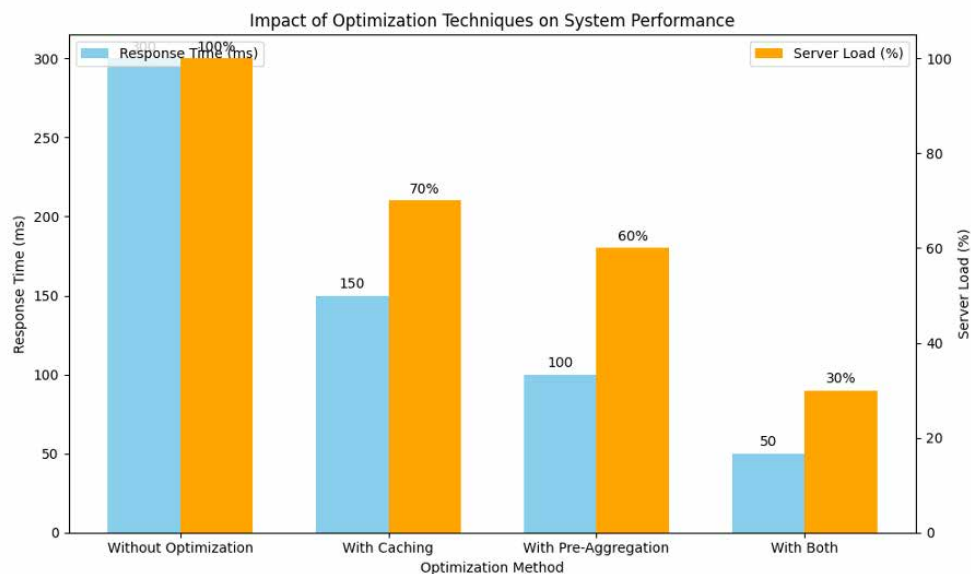


Рис. 1 Порівняння різних методів оптимізації

Крім того, у процесі обробки великих даних не менш важливим є сортування. За результатами експериментальних досліджень [5], найбільш ефективними для великих обсягів інформації є Quicksort у реалізації на Cython, а також сортування за допомогою NumPy. Обидва підходи мають складність $O(n \log n)$ і забезпечують швидку обробку даних. NumPy має перевагу у простоті інтеграції [3], натомість Cython може забезпечити максимальну продуктивність при належному налаштуванні.

Висновки. Комплексне застосування кешування, попередньої агрегації та ефективного сортування дає змогу суттєво оптимізувати роботу інформаційних систем, що працюють із великими даними. У системах моніторингу якості повітря це забезпечує стабільність, оперативність та високу точність під час надання користувачам аналітичної інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація Django. URL: <https://docs.djangoproject.com/>
2. Офіційна документація Celery. URL: <https://docs.celeryq.dev/en/stable/>
3. Офіційна документація NumPy. URL: <https://numpy.org/doc/stable/>
4. Офіційна документація Redis. URL: <https://redis.io/docs/latest/>
5. Хоменко А.О., Кириченко В.В. Ефективність алгоритмів сортування з акцентом на обробку української кирилиці в Python і Cython / Актуальні питання розвитку науки та техніки в умовах глобалізації: матеріали міжнародної науково-практичної конференції / ВСП “Боярський фаховий коледж НУБІП України” / головний редактор Кропивко С.В. / Доступно на: <https://drive.google.com/file/d/1u96UrSBMmOvkuGBmiNYtkDFmbtfWUXEj/view>

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**



**ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ
АСПЕКТИ РОЗРОБКИ
КОМП'ЮТЕРНИХ СИСТЕМ**

**ЗБІРНИК НАУКОВИХ ПРАЦЬ ЗА МАТЕРІАЛАМИ
VII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ І АСПІРАНТІВ
*24 квітня 2025 року***

Київ 2025

УДК 004

Відповідальна за випуск: М.І. Лендел

Збірник наукових праць за матеріалами VII Всеукраїнської науково-практичної конференції студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ '2025», 24 квітня 2025 року, НУБіП України, Київ. – 452 с. (електронне видання)

Відповідальність за зміст публікацій несуть автори.

Передрук матеріалів, а також використання їх будь-якій формі допускається лише з дозволу авторів