

- Вісник Національного університету "Львівська політехніка". "Логістика", № 749; 2012: 530 – 537.
3. Bublyk M., Udovychenko T., Medvid R. Concept of smart specialization in the context of the development of Ukraines economy. Economics. Ecology. Socium, №3(2); 2019: 55-61.
 4. Bublyk M. Economic evaluation of technogenic losses of business entities on fuzzy logic based opportunities. Zarzadzanie organizacja w warunkach niepewnosci teoria i praktyka, 2013: 19 – 29.
 5. Koshtura D., Bublyk M., Matseliukh Y., Dosyn D., Chyrun L., Lozynska O., Karpov I., Peleshchak I., Maslak M., Sachenko O. Analysis of the demand for bicycle use in a smart city based on machine learning. CEUR workshop proceedings, Vol.-2631; 2020: 172-183.

УДК 519.8

ВИЯВЛЕННЯ НАДЛИШКОВОГО ВИКОРИСТАННЯ ПАМ'ЯТІ ПРОГРАМНИМИ ДОДАТКАМИ

Мітіков М., аспірант, mitikov.m22@fpm.dnu.edu.ua, ДНУ

Гук Н.А., доктор фіз.-мат. наук, професор huk_n@fpm.dnu.edu.ua

Розвиток інформаційних технологій поширив всебічне застосування програмних додатків в багатьох сферах сьогодення. Необхідність оброблювати все більші обсяги інформації може компенсуватись збільшенням розрахункової потужності [1] або оптимізацією існуючих програмних реалізацій [2].

Враховуючи лінійне зростання вартості одиниці оперативної пам'яті у хмарному хостингу, складається хибне враження можливості лінійного масштабування загальної вартості володіння. Існуючі пропозиції віртуальних машин пропонуються з розмірами оперативної пам'яті (ОЗП) кратними ступеню двійки (8ГБ – 16ГБ –..). При потребі одного додаткового гігабайта понад наявної пам'яті виникає необхідність масштабування на наступну пропозицію мінімум у 2 рази дорожчу за попередню (Рис. 1).

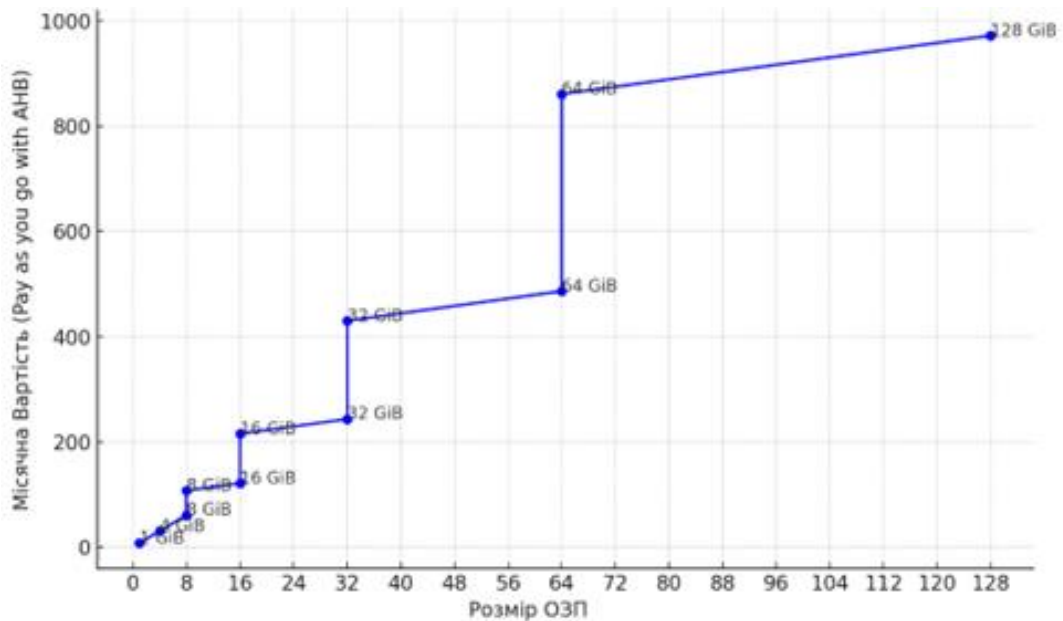


Рисунок 1 – Вартість хмарної системи від розміру оперативної пам'яті

У роботі [2] наведено підходи до зменшення використання пам'яті. За результатами досліджень [3] пропонується використання шаблону “пул об'єктів”, що може значно зменшити час відгуку високопродуктивних багатопотокових застосунків, особливо у середовищах з обмеженою пам'яттю. Але в зазначених роботах чітко не визначено як ідентифікувати ситуації для застосування запропонованих підходів. Однак, з розумінням причин неоптимального розподілу пам'яті та використанням відповідних стратегій для їх уникнення, можна підвищити стабільність, ефективність та вартість роботи системи.

Такі сценарії важко виявити, оскільки вони можуть не проявлятися одразу. Часто проблема з пам'яттю виявляється, коли система починає працювати повільно або нестабільно.

Для виявлення дублювання даних необхідно проаналізувати пам'ять програми і виділити об'єкти, які мають властивість незмінності. Властивість незмінності може задаватися як на рівні мови програмування, так і проявляється для існуючого коду через відсутність викликів методів, які можуть змінювати стан об'єкту. Знаходження властивості незмінності є ключем до зменшення витрат пам'яті через можливість повторного використання однакових об'єктів, але потребує наявності коду виконуваної програми.

Однак найбільш важким для виявлення сценарій збільшеного використання пам'яті є завеликий зарезервований розмір полів об'єктів. Цей сценарій не виявляється жодним існуючим аналізатором через неможливість співставити діапазон реальних даних поля в системі з кодовою базою.

Для усунення зазначених проблем в роботі пропонується використання знімків пам'яті, оскільки в середині знімку пам'яті відображається інформація про операції, що виконуються, стан потоків виконання, об'єкти, якими ці потоки керують, та програмний код. Враховуючи значну кількість об'єктів в пам'яті промислових програмних додатків, ручний пошук поміж тисяч різних типів не є доцільним.

В роботі створюється алгоритм пошуку типів, які призводять до надлишкового використання пам'яті. Алгоритм групує екземпляри об'єктів за типами, виділяє діапазон значень для кожного з полів. Враховуючи діапазон значень, алгоритм пропонує мінімальний достатній розмір для збереження даних, гарантуючи що існуючі значення вмістяться в запропонований діапазон на основі даних певної промислової системи.

Висновок. Визначено, що використання знімків пам'яті доцільно для виявлення її витоків та інших проблем у функціонуванні програмного забезпечення. Автоматизація процесу сканування знімків пам'яті є ефективним методом для аналізу роботи високонавантажених систем. Існуючі методи аналізу фокусуються саме на витоках пам'яті. Надмірне використання пам'яті може залишатись неідентифікованим через відсутність алгоритмів визначення мінімального необхідного обсягу пам'яті. Створено програмний продукт який визначає оптимальні розміри типів даних на основі знімків пам'яті. Обсяги зменшення використаної пам'яті підтверджено серією експериментів на промислових системах.

Список використаних джерел

1. B. Gregg, «2.7.3 Scaling solutions,» в Systems Performance, Second Edition, Boston, Addison-Wesley, 2021, p. 929.
2. J. L. Bentley, Writing efficient programs, Englewood Cliffs, N.J. : Prentice-Hall, 1982.
3. S. E. Ioannis T. Christou, «To Pool or Not To Pool? Revisiting an Old Pattern,» Athens Information Technology, Marousi, 2018.