ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ ТА СИСТЕМИ

doi:https://doi.org/10.15407/emodel.40.05.049

УДК 004.94; 004.4; 004.62

Е.В. Жаріков, канд. техн. наук
Національний технічний університет України
«Київський політехнічний ін-т ім. Ігоря Сікорського»
(Україна, 03056, пр-т Перемоги, 37,
e-mail: zharikov.eduard@acts.kpi.ua)

Структурна оптимізація моделей прогнозу споживання обчислювальних ресурсів в умовах віртуалізації

Забезпечення заданої якості надання хмарних послуг в умовах нестаціонарних навантажень є одним з основних завдань при управлінні хмарним центром обробки даних. Для забезпечення заданої якості надання сервісу необхідно застосовувати проактивний підхід до управління обчислювальними ресурсами. Запобігти виникненню проблем при недостатньому або надмірному виділенню ресурсів можна за допомогою прогнозування споживання ресурсів віртуальними машинами або контейнерами. Запропоновано адаптивний двоетапний метод прогнозування споживання обчислювальних ресурсів, який забезпечує меншу помилку прогнозу у порівнянні з методом прогнозу за моделлю, отриманою на основі тренувальних даних фіксованого розміру. Результати дослідження запропонованого методу показують, що точність прогнозу зростає в середньому від 2,4 до 23,6 % в залежності від статистичних характеристик часового ряду за даними моніторингу. Підвищення точності прогнозу споживання обчислювальних ресурсів дозволяє зменшити енергоспоживання та кількість порушень угоди про рівень обслуговування клієнтів за допомогою більш точного виділення необхідних ресурсів віртуалізованим застосункам хмарного центру обробки даних.

Ключові слова: хмарні обчислення, прогнозування, часовий ряд, віртуалізація, енергоефективність.

В сучасних умовах конкуренції між провайдерами хмарних послуг все актуальнішим стає питання забезпечення заданої якості надання сервісів кінцевому користувачеві. До кожного сервісу висуваються вимоги високої доступності та продуктивності, які забезпечуються за допомогою масштабування, балансування навантаження, резервування та застосування ефективних методів управління ємністю фізичних та віртуальних ресурсів.

IT-інфраструктура, що забезпечує роботу віртуалізованих застосунків, обслуговує різні комбінації нестаціонарних навантажень. В цих умовах

провайдер хмарних послуг повинен забезпечити виконання угоди щодо дотримання рівня сервісу (service-level agreement (SLA). Зазвичай реактивний підхід до управління обчислювальними ресурсами не забезпечує належного рівня надання сервісу, що відбивається, в решті решт, на кількості клієнтів та репутації провайдера. Актуальним є проактивний підхід до управління обчислювальними ресурсами хмарного центру обробки даних (ЦОД), який полягає у застосуванні методів прогнозування.

Одним із найважливіших механізмів, що забезпечує роботу хмарних обчислювальних систем, є консолідація навантажень в рамках однієї ресурсної одиниці — фізичного сервера (ФС). Завдяки застосуванню віртуалізації [1] такі сутності, як віртуальні машини (ВМ) та контейнери розміщуються на одному ФС для більш ефективного використання наявних фізичних ресурсів. Таким чином, якщо отримати прогнозоване значення споживання ресурсів з боку ВМ або контейнера, з'являється можливість запобігти виникненню проблем недостатнього виділення ресурсів (under-provisioning) та надмірного виділення ресурсів (over-provisioning). В першому випадку виникає уповільнення роботи сервісів і порушення SLA, а в другому — збільшуються витрати електроенергії внаслідок збільшення кількості увімкнених ФС.

Аналіз публікацій. Управління віртуалізованими ресурсами з використанням прогнозування широко використовується в системах управління провайдерів хмарних послуг і висвітлюється в [2—9]. Більшість публікацій присвячено прогнозуванню потреби в ресурсах центрального процесора (CPU), оскільки вважається, що нестача цього ресурсу найбільше впливає на якість надання хмарних послуг. Зокрема, в роботах [10—12] наведено методи, що використовуються виключно для прогнозування навантаження процесора в реальному часі, та проаналізовано різні моделі прогнозу і алгоритми навчання нейронних мереж.

В роботі [6] запропоновано фреймворк прогнозування навантаження PRACTISE, що базується на використанні нейронної мережі. Зроблено порівняльний аналіз фреймворку за методом авторегресії інтегрованого ковзного середнього (ARIKC) [13] та звичайною нейронною мережею. Автори роботи [6] зібрали статистику споживання ресурсів декількох тисяч ВМ за два місяці з інтервалом 15 хв. Але запропонований алгоритм надає прогноз тільки на один день, що не може бути використано при оперативному управлінні ВМ, наприклад, впродовж години. При цьому обсяг тренувальних даних обрано довільно і фіксовано. Інші дані, які можна отримати під час роботи в реальних умовах, не враховуються. Повторне тренування моделі відбувається у відповідь на появу сплесків помилки прогнозування. Але не вказано, скільки вимірів взято для пов-

торного тренування моделі, і яким чином формуються різні розбиття. Крім того, опис моделей АРІКС та звичайної нейронної мережі, навчальні вибірки, а також деталі їх отримання не наведено.

У роботі [7] наведено метод прогнозування споживання ресурсів процесора, базованого на лінійній регресії [14]. Запропонований метод дає можливість отримати короткостроковий прогноз. Для перевірки працездатності використано набір даних PlanetLab [15], що містить статистику споживання процесорного ресурсу ВМ з інтервалом вимірювань п'ять хвилин. Але в роботі не враховано, що процес вхідних і вихідних міграцій на ФС, а також різна кількість і склад ВМ суттєво впливають на споживання ресурсу процесора ФС. Тому після завершення міграцій необхідно знов будувати моделі прогнозу. Крім того, використано фіксовану кількість тренувальних даних розміром 12 вимірів для побудови моделі прогнозу, що не може бути виправданим для всіх сценаріїв споживання процесорного ресурсу в промислових умовах.

В роботі [8] розглянуто проблему прогнозування споживання процесорного ресурсу як задачу аналізу часових рядів. Запропоновано фреймворк прогнозування, в якому використовутьюся статистичні моделі прогнозу споживання ресурсу процесора з метою завчасного масштабування фізичних ресурсів і запобігання проблемам недостатнього їх виділення. Запропонований фреймворк базовано на використанні нейронної мережі та лінійної регресії у поєднанні з технікою рухомого вікна (sliding window technique). Але для тренування моделі використано штучні набори даних, отримані генератором навантажень TPC-W. Крім того, кількість вимірів навантаження в наборі відносно мала і складає 135 вимірів для тренування і перевірки моделей. Отже, отриману модель не можна використовувати для прогнозування потреби в процесорних та інших ресурсах в промислових умовах. Як саме була побудована модель прогнозу з використанням рухомого вікна, не уточнюється, але зазначена залежність помилки прогнозу від розміру вікна вказує на те, що найкраща якісь прогнозу досягається при різних розмірах вікна для різних використаних методів. Цей факт не дає можливості визначити конкретний розмір рухомого вікна при прогнозуванні потреби в ресурсі.

У роботі [9] запропоновано адаптивний енергоефективний фреймворк для забезпечення фізичними ресурсами застосунків хмарного ЦОД. Він включає кластеризацію типів навантаження та прогнозування кількості запитів для певного класу ВМ за допомогою фільтра Вінера. Для перевірки працездатності та оцінки роботи запропонованого фреймворку використано трейси Google. Кожної хвилини фреймворк вираховує нові коефіцієнти моделей прогнозу для кожного типу навантаження, адаптуючись до

поточних умов роботи ВМ. Але отримані моделі прогнозу виявились налаштованими таким чином, що фізичні ресурси, згідно прогнозу, виділяються із значним запасом. Це дозволяє запобігти порушенню SLA, але зумовлює надмірне виділення ресурсів і збільшення енергоспоживання. Крім того, не враховано вплив міграцій ВМ на споживання обчислювальних ресурсів.

Постановка задачі. Для роботи ФС або ВМ необхідні наступні ресурси: процесорний час, обсяг оперативної пам'яті, обсяг дискового простору, продуктивність підсистеми зберігання, продуктивність підсистеми мережевої взаємодії та ін. Споживання вказаних ресурсів змінюється в залежності від багатьох чинників, які складно врахувати при виборі керуючих впливів. Зазвичай ВМ споживають не весь обсяг ресурсів, замовлений при їх створенні і розгортанні. Максимальне споживання ресурсів виникає епізодично і не завжди одночасно для всіх ВМ, розміщених на ФС. Тому виникає можливість розміщувати більшу кількість ВМ на одному ФС, ніж передбачено максимальним споживанням ресурсів. Але при виникненні пікових навантажень на більшість ВМ наявних фізичних ресурсів ФС не вистачає, що спричиняє збільшення затримки відгуку на запити та збільшення часу виконання застосунків. При цьому порушується SLA для всіх ВМ на такому ФС.

Таким чином, виникає задача прогнозування споживання обчислювальних ресурсів для фізичного або віртуального сервера з метою зменшення кількості порушень SLA та зменшення енергоспоживання через прийняття управлінських рішень щодо розміщення нових ВМ та міграції існуючих.

Ідея комбінованого методу прогнозу. Вибрати параметри і коефіцієнти моделі для прогнозування споживання ресурсів при будь-яких навантаженнях і їх комбінаціях не уявляється можливим. Відтак, потрібно через певний проміжок часу створювати нову модель, застосовуючи нові дані, що надійшли від підсистеми моніторингу. Кількість даних моніторингу, потрібних для отримання моделі прогнозу, необхідно визначити експериментально. Оскільки строгого критерію застосування того чи іншого методу при прогнозуванні споживання обчислювальних ресурсів не знайдено, пропонується комбінований метод з адаптацією параметрів моделі.

Комбінований метод прогнозування споживання обчислювальних ресурсів полягає у застосуванні декількох методів прогнозування з адаптацією розміру тренувальної вибірки і подальшим вибором прогнозу на основі визначеного критерію (табл. 1). В якості критеріїв пропонуються наступні: мінімальний прогноз на попередніх кроках n ($n \in \mathrm{N}^+$), усереднення прогнозів та зважене усереднення прогнозів. Склад робочих методів

прогнозування обирається за результатами попередніх досліджень їх застосування до тестових вибірок. Основним фактором, що впливає на склад робочих методів, є кількість методів та час обчислення прогнозів кожним з них.

Адаптивний метод прогнозування споживання ресурсів. Проблема прогнозування потреби в ресурсах для ФС не є тривіальною. Об'єм фізичних ресурсів, які потребує ФС з працюючими ВМ, дорівнює сумі потреб цих ВМ у віртуальних ресурсах. Отже, є два способи прогнозувати потреби фізичних ресурсів (табл. 1):

1 — використовуючи дані моніторингу споживання ресурсів ФС;

2 — використовуючи дані моніторингу споживання ресурсів всіма ВМ, що виконується на ФС.

Потреба у ресурсах для ВМ може суттєво змінюватися під впливом числа клієнтських запитів і архітектури конкретних застосунків, які працюють всередині ВМ. Розглянути всі можливі комбінації розміщення ВМ з конкретними потребами ресурсів не можливо. Крім того, моделі прогнозу і їх параметри, як правило, повинні бути обрані відповідно до конкретних умов використання в процесі роботи ВМ або ФС. Застосування методу зі структурною адаптацією моделі прогнозу споживання ресурсів за прикладом ВМ обумовлено використанням відповідних даних моніторингу роботи ВМ для експериментального дослідження моделей. Запропонований метод можна використовувати також для прогнозу потреби ресурсів ФС.

Оскільки умови роботи сервера (віртуального або фізичного) залежать від багатьох випадкових факторів, як зовнішніх, так і внутрішніх, необхідно адаптуватися до різних умов при прогнозуванні споживання ресур-

Таблиця 1. **Порівняння способів отримання прогнозу споживання ресурсів**

Спосіб	Перевага	Недолік	Обмеження
Прогнозування споживання ресурсів для ФС	Одноразова побудова моделей прогнозу та одноразове обчислення прогнозу	Вхідні та вихідні міграції спотворюють реальне споживання ресурсів ВМ	Потрібно враховувати дані моніторингу, отримані після завершення вхідних та вихідних міграцій
Прогнозування споживання ресурсів для всіх ВМ на ФС	Кількість історичних даних моніторингу кожної ВМ не залежить від кількості міграцій. Дані про споживання ресурсу (і попередні моделі з прогнозами) можуть мігрувати разом з ВМ	Побудова моделей прогнозу та обчислення прогнозів виконується для кожної ВМ. Відбувається накопичення помилки прогнозу в цілому для ФС	Немає

сів, використовуючи структурну адаптацію. В даному випадку структурна адаптація полягає в тому, щоб вибрати кращий метод та кращу модель прогнозу (її параметри) в залежності від поточних умов експлуатації сервера [16]. Прогноз споживання кожного ресурсу виконується окремо і не залежить від методів, обраних для прогнозу споживання інших ресурсів. Розглянемо роботу запропонованого методу на прикладі прогнозу споживання процесорного ресурсу ВМ.

На першому етапі роботи методу, серед обраних методів прогнозу, для прогнозування споживання ресурсів на наступний крок використовується той метод, що дав найкращий прогноз на поточному кроці. Поточним кроком назвем стан системи, коли надійшли фактичні дані про споживання ресурсу і є можливість обрахувати середню абсолютну похибку прогнозу в процентах. Отже, для вибору управління на поточному кроці обираємо прогноз, отриманий методом, що дав мінімальну середню абсолютну похибку в процентах на поточному кроці. На другому етапі визначаються параметри моделей та їх структура. Використаємо вбудовані в мову програмування R [17] алгоритми підбору параметрів моделей та запропонуємо новий алгоритм підбору довжини вікна тренувальних даних. При цьому розмір вікна тренувальних даних обираємо такий, за допомогою якого на попередньому кроці було отримано прогноз з мінімальною середньою абсолютною похибкою в процентах.

Перший і другий етапи методу базовано на припущенні, що найякісніший прогноз на наступному кроці буде отримано тим методом і тією моделлю, за допомогою яких отримано найкращий прогноз на поточному кроці. Обидва етапи роботи методу є адаптивними, на першому етапі до поточних умов адаптується метод, на другому етапі адаптується кількість тренувальних даних цього методу.

Отримані моделі використовуються для обчислення прогнозованого значення на один крок вперед. Зазвичай для перевірки адекватності отриманої моделі необхідно проаналізувати залишки моделі (residuals). Якщо за допомогою автокореляційної функції і щільності розподілу встановлено, що залишки некорельовані і мають нормальний розподіл (або наближений до нього), то модель є адекватною.

В умовах автоматичного прогнозування потреб в ресурсах тренувальні дані постійно надходять від підсистеми моніторингу і використовуються для створення нової моделі прогнозу на кожному кроці управління. У запропонованому методі зроблено припущення про те, що неможливо обрати структуру і параметри моделі прогнозу споживання обчислювального ресурсу, використовуючи деякий довільний набір (набори) послідовних спостережень використання цього обчислювального ресурсу,

отриманий в одній або декількох ВМ. Це обумовлено, зокрема, складністю отримання таких початкових даних: розмір тренувальної вибірки, кількість тренувальних вибірок, кількість і тип ВМ, з яких отримано вибірки.

Таким чином, у запропонованому методі неможливо на кожному кроці аналізувати адекватність моделей прогнозу загальноприйнятим способом. Тому пропонується визначати точність отриманої моделі прогнозу за допомогою середньої абсолютної похибки в процентах MAPE (Mean Absolute Percentage Error) [18], не перевіряючи її адекватність. Використання метрики MAPE обумовлено необхідністю порівняння точності прогнозу, отриманого різними методами на різних наборах тренувальних даних. Якщо MAPE <10%, то отриманий прогноз має високу точність, при 10% < <MAPE <20% точність прогнозу добра, при 20% < MAPE < 50% точність прогнозу задовільна, при MAPE > 50% точність прогнозу незадовільна.

Базуючись на прогнозованих значеннях, менеджер ФС може приймати рішення про виділення більшого об'єму ресурсу для ВМ або міграції ВМ в межах однієї стійки. Прогнозовані значення враховуються також на верхньому рівні управління для визначення ФС, на яких буде розміщено нові або мігруючі ВМ.

Методи прогнозування, використані у дослідженні. В запропонованому адаптивному методі склад робочих методів, що використовуються безпосередньо для обрахування прогнозу, не визначений. Кількість методів і їх тип пропонується підбирати в залежності від типу ресурсу, споживання якого потрібно прогнозувати. Для порівняльного аналізу обрано чотири методи, що застосовуються для прогнозування споживання процесорного ресурсу.

Метод простого експоненціального згладжування (simple exponential smoothing (SES)) використовується для прогнозування часових рядів у випадках, якщо дані спостереження не мають вираженого тренду і сезонності або ці характеристики можуть з'являтися в даних тимчасово [19]. При обрахуванні прогнозу ступінь врахування попередніх значень ряду зменшується за експоненціальним законом: $\hat{y}_{t+1|t} = \alpha y_t + \alpha(1-\alpha)\, y_{t-1} + +\alpha(1-\alpha)^2 y_{t-2} + L$, де $0<\alpha<1$ — параметр згладжування; $\hat{y}_{t+1|t}$ — прогнозоване значення на один крок вперед в момент часу t; y_t, y_{t-1}, y_{t-2} — значення часового ряду, що входять до тренувальної вибірки.

Підбір параметрів моделі SES і обчислення прогнозу на один крок виконується функцією *ses* (*ts*, *h* = 1) в мові R, де *ts* — часовий ряд. При цьому функція вирішує нелінійну оптимізаційну задачу підбору параметрів і початкових значень моделі через мінімізацію суми квадратичних похибок. Наступним кроком є отримання похибки прогнозу MAPE викликом функції *accuracy*(*fitses$mean*[1],*testts*), де *fitses$mean*[1] — прогно-

зоване значення; *testts* — фактичне значення, отримане від підсистеми моніторингу.

Метод Хольта та демпфірований метод тренду. Тренд Хольта та демпфірований метод тренду є методами, що розширюють можливості методу SES. Тренд Хольта та демпфірований метод тренду позначені відповідно holt та dholt. Якщо в даних спостереження є наявний тренд, ці методи можуть дати якісний прогноз [20]. Рівняння для прогнозу методом Хольта має вигляд

$$\hat{y}_{t+1|t} = l_t + b_t,$$

$$l_t = \alpha\, y_t + (1-\alpha)(l_{t-1} + b_{t-1}),\ b_t = \beta(l_t - l_{t-1}) + (1-\beta)\, b_{t-1}\ ,$$

де l_t — оцінка рівня часового ряду в момент часу t; b_t — оцінка тренду часового ряду в момент часу t; α — параметр згладжування для рівня часового ряду, $0 \le \alpha \le 1$; β — параметр згладжування для тренду часового ряду, $0 \le \beta \le 1$. Параметри згладжування і початкові значення l_t та b_t для моделі прогнозу оцінюються через мінімізацію суми квадратичних похибок. Підбір параметрів моделі за методом Хольта і обчислення прогнозу на один крок виконується функцією *holt*(*ts*, *h* = 1) в R, де *ts* — часовий ряд. Значення похибки прогнозу MAPE повертає функція *accuracy*(*fitholt$mean*[1], *testts*), де *fitholt$mean*[1] — прогнозоване значення, *testts* — фактичне значення, отримане від підсистеми моніторингу.

Емпіричні дослідження часових рядів показали, що метод Хольта дає «надмірний» прогноз. Значно кращі результати дає демпфірування тренду Хольта через введення спеціального параметру демпфірування [21]. Рівняння однокрокового прогнозу методом демпфіруваного тренду Хольта має вигляд

$$\hat{y}_{t+1|t} = l_t + \phi b_t,$$

$$l_t = \alpha\, y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1}),\ b_t = \beta(l_t - l_{t-1}) + (1-\beta)\,\phi b_{t-1}.$$

Для коефіцієнта ϕ зазвичай встановлюють значення в діапазоні [0,8 — 0,98]. Підбір параметрів моделі за методом демпфіруваного тренду Хольта і обчислення прогнозу на один крок виконується за допомогою функції *holt*(*ts*, *damped* = TRUE, *phi* = 0,9, *h* = 1) в R, де *ts* — часовий ряд. Похибку прогнозу MAPE повертає функція *accuracy*(*fitdholt$mean*[1], *testts*), де *fitdholt$mean*[1] — прогнозоване значення, *testts* — фактичне значення, отримане від підсистеми моніторингу.

Метод авторегресії інтегрованого ковзного середнього (AutoARIMA). В системах управління дані з підсистем моніторингу надаються у вигляді часового ряду. Кількість елементів часового ряду, що розглядаються як

тренувальні дані для створення моделі, визначається вимогами до адекватності моделі та точності прогнозу. Аналіз часових рядів з наборів [22] показав, що умови стаціонарності для великих вибірок завжди не виконуються, а для частин цих вибірок іноді можуть виконуватися. Тому застосування методів авторегресії (АР), ковзного середнього (КС) або їх комбінації (АРКС) обмежено. Якщо для досліджуваного часового ряду виконуються умови стаціонарності, то для прогнозування можна обрати одну з моделей (або їх комбінацію). Модель АРКС описано наступним рівнянням:

$$y_t = \phi_0 + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t,$$

де ϕ_0, ϕ_i — коефіцієнти моделі авторегресії порядку p; θ_i — коефіцієнти моделі КС порядку q; ε_t — білий шум ($E(\varepsilon_t) = 0$, $\mathrm{Var}(\varepsilon_t) = \sigma_\varepsilon^2$, $\mathrm{Cov}(\varepsilon_t, \varepsilon_s) = 0$, $t \neq s$).

Для прогнозу наступного значення часового ряду за допомогою моделі АРКС використовується вираз із застосуванням оператора зсуву:

$$\hat{y}_{t+1|t} = \sum_{i=0}^{p-1} \phi_i L^i y_t + \sum_{i=0}^{q-1} \theta_i L^i \varepsilon_t + \varepsilon_{t+1},$$

де L^i — оператор зсуву, визначений як $L^i y_t = y_{t-1}$; ϕ_i, θ_i — коефіцієнти моделі. Якщо часовий рад, отриманий від підсистеми моніторингу, є нестаціонарним, для прогнозу треба отримати модель АРІКС [13] порядку d через взяття перших ($d = 1$) або других ($d = 2$) різниць між членами ряду. Зазвичай щоб отримати стаціонарний ряд і скористатися моделлю авторегресії ковзного середнього, на практиці достатньо взяти перші різниці членів ряду. Отже, модель АРІКС(p, d, q) має вигляд

$$\left(1 - \sum_{i=0}^{p-1} \phi_i L^i\right)(1-L)^d y_t = \left(1 + \sum_{i=0}^{q-1} \theta_i L^i\right) \varepsilon_t.$$

Підбір параметрів моделі АРІКС, включаючи окремі випадки (АР, КС, АРКС), виконується автоматично одним із варіантів алгоритму Hyndman—Khandakar [23] через виклк функції *auto.arima*() в R. Для отримання моделі за допомогою цієї функції використовується тест на стаціонарність, мінімізація інформаційного критерію Акаіке та метод максимальної правдоподібності.

Аналіз початкових даних. Для аналізу роботи запропонованого методу прогнозування використано набір даних, отриманий системою моніторингу ВМ розподіленого хмарного ЦОД Bitbrains [22]. Набір даних

Bitbrains містить показники роботи 1750 ВМ із розподіленого ЦОД, який спеціалізується на постачанні послуг хостингу та бізнес-розрахунків для підприємств. Кожен файл набору даних (трейс) містить показники споживання однією ВМ основних обчислювальних ресурсів таких як процесорний час, обсяг оперативної пам'яті, продуктивність підсистеми зберігання та продуктивність підсистеми мережевої взаємодії [24]. Для дослідження з кожного трейсу взяті тільки дані про споживання процесорних ресурсів. Дані про споживання процесорного ресурсу ВМ являють собою часовий ряд, тривалість якого складає один місяць (заміри значень зроблено з інтервалом 5 хв.). Однак у цьому дослідженні при виконанні обчислень було використано дані за чотири доби, тобто 1152 значення часового ряду.

Враховуючи, що навантаження на кожну ВМ має свій особливий характер, з усього набору даних для досліджень було обрано шість трейсів ВМ з номерами 599, 795, 840, 850, 904 та 957. Трейси ВМ обрано таким чином, щоб статистичні показники кожного з них суттєво відрізнялись (табл. 2). Трейси не включають ніякої інформації про навантаження і додатки, що виконуються всередині ВМ.

За допомогою візуального аналізу часових рядів кожної ВМ (рис. 1), не використовуючи формальних методів перевірки на стаціонарність, можна зробити висновок про нестаціонарність наведених часових рядів.

Таблиця 2. **Статистичні показники часових рядів споживання процесорного ресурсу**

Покажчик	ВМ599	ВМ795	ВМ840	ВМ850	ВМ904	ВМ957
Середнє	10,37598	0,83502	25,63766	5,44349	1,20278	1,75686
Стандартна похибка	0,82311	0,0379	0,02682	0,08682	0,01092	0,07531
Медиана	1	0,5	25,63333	4,63333	1	1,11667
Мода	1	0,5	25,83333	4,3	1	1
Стандартне відхилення	27,84026	1,282	0,90721	2,9365	0,3694	2,54714
Дисперсія	775,08024	1,64354	0,82303	8,62302	0,13645	6,48794
Ексцес	5,10384	41,5692	-0,07084	18,9043	15,0006	36,97383
Асиметричність	2,6587	5,78652	0,0455	3,94964	3,29004	5,37852
Інтервал	95,88333	14,2	5,76667	26,06667	3,53333	31,01667
Мінімум	1	0,5	22,8	2,03333	1	1
Максимум	96,88333	14,7	28,56667	28,1	4,53333	32,01667
Сума	11870,11667	955,26557	29329,48004	6227,35615	1375,97619	2009,84549
Кількість вимірювань	1152	1152	1152	1152	1152	1152

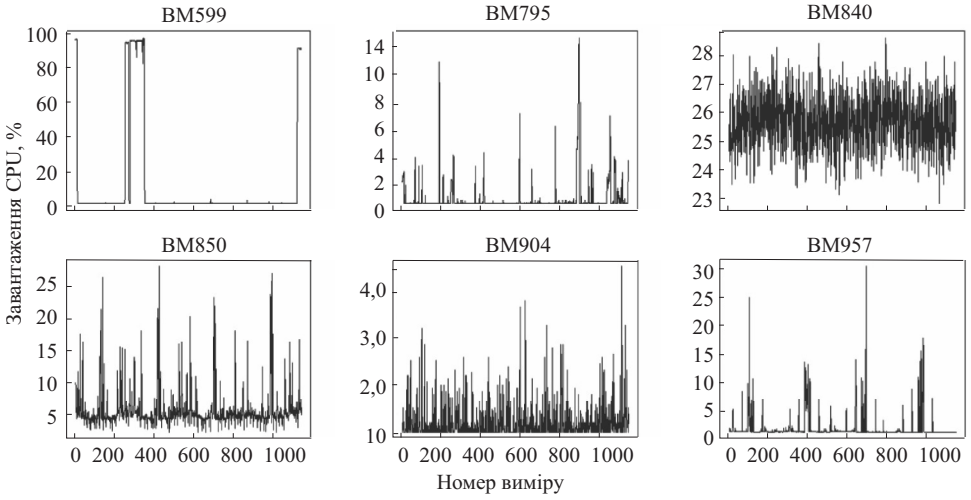

Рис. 1. Часові ряди вимірів споживання процесорного ресурсу ВМ

Водночас, можна припустити, що на деяких відрізках часу в кожному з трейсів часовий ряд є стаціонарним, тому виправданим є використання методів, наведених у попередньому розділі. Крім того, оскільки дослідити всі трейси і отримати відповідні моделі прогнозу не можливо, в даному разі аналіз ряду на стаціонарність не використовується.

Результати експериментальних досліджень. Підбір моделей для прогнозування споживання процесорного ресурсу виконано за допомогою мови програмування R (версія 3.01) [17] та пакету forecast [25]. Для перевірки працездатності запропонований метод було спрощено через виконання першого етапу вручну. Отже, роботу обраних методів прогнозу, описаних у попередньому розділі, досліджено на прикладі шести часових рядів. За результатами аналізу показників якості обрано метод прогнозу, який використовувався на другому етапі роботи. Відповідні параметри моделі для цього методу підбирались через вибір розміру вікна тренувальних даних. При цьому на другому етапі роботи методу інші методи прогнозування не розглядалися.

Аналіз методів прогнозування SES, AutoARIMA, HOLT та DHOLT проведено для змінного розміру вікна тренувальних даних. Ідея полягає в тому, що на кожному кроці управління слід обраховувати моделі прогнозу для декількох розмірів вікон тренувальних даних (рис. 2). При цьому розмір максимального $w_{\max}$ та мінімального $w_{\min}$ вікон визначається підбором. Дослідження показали, що при розмірі вікна менше восьми якість прогнозу обраними методами є незадовільною.

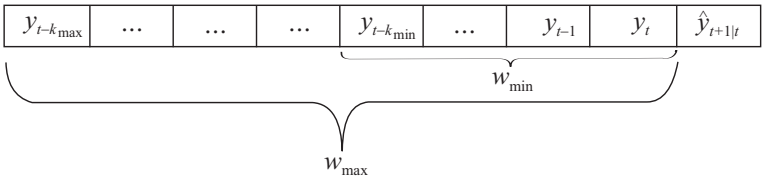

Рис. 2. Змінна довжина вікна тренувальних даних

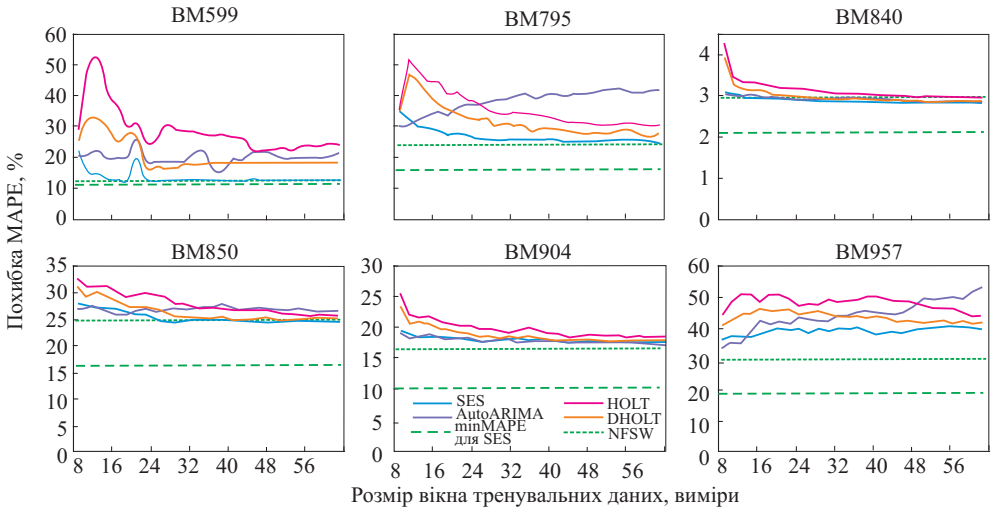

Рис. 3. Графіки якості прогнозу споживання процесорного ресурсу

При аналізі кожного з обраних методів підібрано моделі прогнозу для 30 вікон тренувальних даних (від восьми до 62 з кроком два) і шести трейсів роботи різних ВМ. Відтак, мінімальний розмір досліджуваного вікна становить $w_{\min} = 8$, максимальний — $w_{\max} = 62$. Для кожної з 30 моделей обраховано значення MAPE і побудовано залежність помилки прогнозу MAPE від довжини вікна тренувальних даних (рис. 3).

Результати аналізу свідчать про те, що якість прогнозу в значній мірі залежить від статистичних характеристик трейсу. Цей висновок підтверджує припущення про неможливість обрати один метод прогнозу для використання в промислових умовах роботи фізичних і віртуальних серверів. Крім того, як бачимо, подальше збільшення розміру вікна тренувальних даних не зумовлює суттєвого покращення якості прогнозу. Для шести обраних трейсів і різних розмірів вікон тренувальних даних найкращі прогнози отримано за допомогою моделей простого експоненціального згладжування. Для трейсу BM840 метод SES показав дуже гарний результат, а для трейсів BM599 та BM904 — добрий результат. Однак для

трейсів BM795, BM850 та BM957 прогноз виявився задовільним, що потребує додаткових заходів щодо покращення і використання прогнозів. Таким чином, для перевірки роботи запропонованого адаптивного методу прогнозування споживання ресурсів на другому етапі його роботи використано метод простого експоненціального згладжування.

Для дослідження ефективності другого етапу роботи запропонованого методу обчислено мінімальну похибку (minMAPE) серед 30 розмірів вікон тренувальних даних для кожного трейсу. Тобто на кожному кроці управління для підбору моделі прогнозу обирається розмір вікна тренувальних даних, при якому похибка MAPE є мінімальною, потім обраховується середня похибка для всіх кроків управління, від 63 до 1151. Значення minMAPE обраховано з метою порівняння зі значеннями, отриманими на другому етапі адаптивного методу.

Далі, на кожному кроці управління, обирається прогнозоване значення, отримане моделлю. Параметри моделі обраховано через розмір вікна тренувальних даних, на якому отримано мінімальну похибку на попередньому кроці управління. Обраховано середню похибку (NFSW) для всіх кроків управління, від 63 до 1151. Для трейсів BM795, BM904 та BM957 запропонований адаптивний метод показує кращі результати порівняно зі звичайним методом SES відповідно на 10,7; 7,8; 23,6% . А для трейсів BM599, BM840 та BM850 при розмірі вікна тренувальних даних більше ніж 24 якість прогнозу наближена до якості роботи методу SES, тобто в середньому прогноз кращий відповідно на 9,3; 2,7; 2,4%.

Таким чином, запропонований адаптивний двоетапний метод прогнозування споживання обчислювальних ресурсів дозволяє отримати прогнозовані значення з меншою помилкою прогнозу, ніж при використанні одного методу прогнозу з фіксованим розміром тренувальних даних для обчислення моделі прогнозу.

Висновки

Якісне прогнозування потреби в обчислювальних ресурсах дозволяє застосувати проактивний підхід до управління *IT*-інфраструктурою зі зменшенням енергоспоживання та кількості порушень угоди про рівень обслуговування клієнтів. При дослідженні запропонованого методу використано метод простого експоненціального згладжування, метод Хольта, демпфірований метод тренду, моделі авторегресії інтегрованого ковзного середнього. Аналіз та моделювання виконано з використанням статистичних даних Bitbrains [22] на прикладі прогнозування споживання процесорного ресурсу. Якісний аналіз результатів дослідження показав, що точність прогнозу в значній мірі залежить від статистичних характеристик

трейсу. Результати кількісного аналізу показують, що точність прогнозу, отриманого за допомогою запропонованого методу, зростає в залежності від статистичних характеристик трейсу. Найкращий прогноз споживання процесорного ресурсу отримано за допомогою методу простого експоненціального згладжування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Barham P., Dragovic B., Fraser K. et al.* Xen and the art of virtualization // In ACM SIGOPS operating systems review. 2003, Vol. 37, No. 5, p. 164—177.
2. *Chen G., He W., Liu J., et al.* Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services // In NSDI. 2008, Vol. 8, p. 337—350.
3. *Padala P., Hou K.Y., Shin K.G., et al.* Automated control of multiple virtualized resources // Proc. of the ACM European conference on Computer systems (EuroSys'09), ACM, 2009, p. 13—26.
4. *Gross, G., Galiana, F.D.* Short-term load forecasting // Proc. of the IEEE. 1987, 75(12), p. 1558—1573.
5. *Xiao Z., Song W., Chen Q.* Dynamic resource allocation using virtual machines for cloud computing environment // Transactions on Parallel and Distributed Systems, IEEE. 2013, Vol. 24, No. 6, p. 1107—1117.
6. *Xue J., Yan F., Birke R., et al.* PRACTISE: Robust prediction of data center time series // 11th International Conference on Network and Service Management (CNSM). IEEE, 2015, p. 126—134.
7. *Farahnakian F., Liljeberg P., Plosila J.* LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers // 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 2013, p. 357—364.
8. *Islam S., Keung J., Lee K., Liu A.* Empirical prediction models for adaptive resource provisioning in the cloud // Future Generation Computer Systems. 2012, Vol. 28, No. 1, p. 155—162.
9. *Dabbagh M., Hamdaoui B., Guizani M., Rayes A.* Energy-efficient resource allocation and provisioning framework for cloud data centers // IEEE Transactions on Network and Service Management. 2015, Vol. 12, No. 3, p. 377—391.
10. *Naseera, S., Rajini, G.K., Prabha, N.A., Abhishek G.*A comparative study on CPU load predictions in a computational grid using artificial neural network algorithms // Indian Journal of Science and Technology. 2015, Vol. 8, No. 35, p. 1—5.
11. *Naseera S., Rajini G.K., Reddy P.S.K.* Host CPU Load Prediction Using Statistical Algorithms a comparative study // International Journal of Computer Technology and Applications. 2016, 9 (12), p. 5577—5582.
12. *Dinda P.A.* Design, implementation, and performance of an extensible toolkit for resource prediction in distributed systems // IEEE Transactions on Parallel and Distributed Systems. 2006, Vol. 17, No. 2, p. 160—173.
13. *Box G.E., Jenkins G.M., Reinsel G.C., Ljung G.M.* Time series analysis: forecasting and control // 5th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015, p. 712.
14. *Montgomery D.C., Peck E.A., Geoffrey G.G.* Introduction to linear regression analysis // John Wiley & Sons, 2015, p. 612.
15. *Park K., Pai V.S.* CoMon: a mostly-scalable monitoring system for PlanetLab // ACM SIGOPS Operating Systems Review, 2006, p. 65—47.
16. *Telenyk S., Zharikov E., Rolik O.* Architecture and Conceptual Bases of Cloud IT Infrastructure Management // Advances in Intelligent Systems and Computing. 2017, Vol. 512, p. 41—62.

17. *R Core Team* R: A language and environment for statistical computing // R Foundation for Statistical Computing. Vienna, Austria, 2018, URL https://www.R-project.org/.
18. *Jorgensen M.* Experience with the accuracy of software maintenance task effort prediction models // IEEE Transactions on Software Engineering. 1995, Vol. 21, p. 674—681.
19. *Hyndman R., Koehler A.B., Ord J.K., Snyder R.D.* Forecasting with exponential smoothing: the state space approach // Springer Science & Business Media, 2008, p. 359.
20. *Holt C. C.* Forecasting seasonals and trends by exponentially weighted moving averages // International journal of forecasting. 2004, Vol. 20, No. 1, p. 5—10.
21. *Gardner Jr E.S., McKenzie E.D.* Forecasting trends in time series // Management Science. 1985, Vol. 31, No. 10, p. 1237—1246.
22. *GWA-T-12* Bitbrains [Online] Available from: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains [Accessed September 12, 2018].
23. *Hyndman R.J., Khandakar Y.* Automatic time series forecasting: The forecast package for R // Journal of Statistical Software. 2008, 27(1), p. 1—22. Retrieved from https://www.jstatsoft.org/article/view/v027i03
24. *Shen S., van Beek V., Iosup A.* Statistical characterization of business-critical workloads hosted in cloud datacenters // 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). IEEE, 2015, p. 465—474.
25. *Hyndman R., Bergmeir C., Caceres G. et al.* Forecast: Forecasting functions for time series and linear models // R package version 8.3, 2018, <URL: http://pkg.robjhyndman.com/forecast>.

Отримано 12.09.18

REFERENCES

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A. and Warfield, A. (2003), "Xen and the art of virtualization", *ACM SIGOPS operating systems review*, Vol. 37, no. 5, pp. 164-177.
2. Chen, G., He, W., Liu, J. et al. (2008), "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services", *NSDI*, Vol. 8, pp. 337-350.
3. Padala, P., Hou, K.Y., Shin, K.G. et al. (2009), "Automated control of multiple virtualized resources", *Proc. of the ACM European conference on Computer systems* (EuroSys'09), ACM, pp. 13-26.
4. Gross, G. and Galiana, F.D. (1987), "Short-term load forecasting", *Proceedings of the IEEE*, 75(12), pp. 1558-1573.
5. Xiao, Z., Song, W. and Chen, Q. (2013), "Dynamic resource allocation using virtual machines for cloud computing environment", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, no. 6, pp. 1107-1117.
6. Xue, J., Yan, F., Birke, R. et al. (2015), "PRACTISE: Robust prediction of data center time series", *11th International Conference on Network and Service Management* (CNSM), IEEE, pp. 126-134.
7. Farahnakian, F., Liljeberg, P. and Plosila, J. (2013), "LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers", *39th EUROMICRO Conference on Software Engineering and Advanced Applications* (SEAA), IEEE, pp. 357-364.
8. Islam, S., Keung, J., Lee, K. and Liu, A. (2012), "Empirical prediction models for adaptive resource provisioning in the cloud", *Future Generation Computer Systems*, Vol. 28, no. 1, pp. 155-162.

9. Dabbagh, M., Hamdaoui, B., Guizani, M. and Rayes, A. (2015), "Energy-efficient resource allocation and provisioning framework for cloud data centers", *IEEE Transactions on Network and Service Management*, Vol. 12, no. 3, pp. 377-391.
10. Naseera, S., Rajini, G.K., Prabha, N.A. and Abhishek, G. (2015), "A comparative study on CPU load predictions in a computational grid using artificial neural network algorithms", *Indian Journal of Science and Technology*, Vol. 8, no. 35, pp. 1-5.
11. Naseera, S., Rajini, G.K. and Reddy, P.S.K. (2016), "Host CPU Load Prediction Using Statistical Algorithms a comparative study", *International Journal of Computer Technology and Applications*, 9 (12), pp. 5577-5582.
12. Dinda, P.A. (2006), "Design, implementation, and performance of an extensible toolkit for resource prediction in distributed systems", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, no. 2, pp. 160-173.
13. Box, G.E., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. (2015), Time series analysis: fore-casting and control, 5th ed. Hoboken, NJ, USA: John Wiley & Sons.
14. Montgomery, D.C., Peck, E.A. and Vining, G.G. (2015), Introduction to linear regression analysis, John Wiley & Sons.
15. Park, K. and Pai, V.S. (2006), "CoMon: a mostly-scalable monitoring system for PlanetLab", *ACM SIGOPS Operating Systems Review*, pp. 65-47.
16. Telenyk, S., Zharikov, E. and Rolik, O. (2017), "Architecture and Conceptual Bases of Cloud IT Infrastructure Management", *Advances in Intelligent Systems and Computing*, Springer, Vol. 512, pp. 41-62.
17. R Core Team (2018), "R: A language and environment for statistical computing", *R Foundation for Statistical Computing*, Vienna, Austria, URL, available at: https://www.R-project.org/.
18. Jorgensen, M. (1995), "Experience with the accuracy of software maintenance task effort prediction models", *IEEE Transactions on Software Engineering*, Vol. 21, pp. 674-681.
19. Hyndman, R., Koehler, A.B., Ord, J.K. and Snyder, R.D. (2008), Forecasting with exponential smoothing: the state space approach, Springer Science & Business Media.
20. Holt, C.C. (2004), "Forecasting seasonals and trends by exponentially weighted moving averages", *International journal of forecasting*, Vol. 20, no. 1, pp. 5-10.
21. Gardner Jr E.S. and McKenzie E.D. (1985), "Forecasting trends in time series", *Management Science*, Vol. 31, no. 10, pp. 1237-1246.
22. GWA-T-12 Bitbrains, available at: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains (accessed September 12, 2018).
23. Hyndman, R.J. and Khandakar, Y. (2008), "Automatic time series forecasting: The forecast package for R", *Journal of Statistical Software*, 27(1), pp. 1-22, available at: https://www.jstatsoft.org/article/view/v027i03.
24. Shen, S., van Beek, V. and Iosup, A. (2015), "Statistical characterization of business-critical workloads hosted in cloud datacenters", *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (CCGrid), IEEE, pp. 465-474.
25. Hyndman, R., Bergmeir, C., Caceres, G. et al. (2018), "Forecast: Forecasting functions for time series and linear models", *R package version 8.3*, <URL, available at: http://pkg.robjhyndman.com/forecast>.

Received 12.09.18

Э.В. Жариков

СТРУКТУРНАЯ ОПТИМИЗАЦИЯ МОДЕЛЕЙ ПРОГНОЗА ПОТРЕБЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ В УСЛОВИЯХ ВИРТУАЛИЗАЦИИ

Обеспечение заданного качества предоставления облачных услуг в условиях нестационарных нагрузок является одной из основных задач при управлении облачным центром обработки данных. Для обеспечения заданного качества предоставления сервиса необходимо применять проактивный подход к управлению вычислительными ресурсами. Предотвратить возникновение проблем при недостаточном или чрезмерном выделении ресурсов можно с помощью прогнозирования потребления ресурсов виртуальными машинами или контейнерами. Предложен адаптивный метод прогнозирования потребления вычислительных ресурсов, обеспечивающий меньшую ошибку прогноза по сравнению с методом прогноза с помошью модели, полученной на тренировочных данных фиксированного размера. Результаты исследования предлагаемого метода показывают, что точность прогноза возрастает в среднем от 2,4 до 23,6% в зависимости от статистических характеристик временного ряда по данным мониторинга. Повышение точности прогноза потребления вычислительных ресурсов позволяет уменьшить энергопотребление и число нарушений соглашения об уровне обслуживания клиентов посредством более точного выделения необходимых ресурсов виртуализированным приложениям облачного центра обработки данных.

К л ю ч е в ы е с л о в а: облачные вычисления, прогнозирование, временной ряд, виртуализация, энергоэффективность.

E.V. Zharikov

STRUCTURAL OPTIMIZATION OF THE RESOURCE CONSUMPTION FORECASTING MODELS IN VIRTUALIZED ENVIRONMENT

Providing a given quality of cloud services under non-stationary workload is one of the main tasks in managing a cloud data center. To ensure a given service quality, it is necessary to apply a proactive approach to managing computing resources. It is possible to prevent problems from insufficient or excessive allocation of resources by forecasting the consumption of resources by virtual machines or containers. An adaptive method for forecasting the consumption of computational resources is proposed, which provides a smaller forecasting error compared to the forecasting method using a model obtained from fixed-size training data. The results of the study of the proposed method show that the forecast accuracy increases on average from 2.4% to 23.6% depending on the statistical characteristics of the time series according to monitoring data. Increasing the accuracy of forecasting the consumption of computing resources allows to reduce power consumption and the number of violations of the customer service level agreement through more accurate allocation of the necessary resources to virtualized cloud data center applications.

K e y w o r d s: cloud computing, forecasting, time series, virtualization, energy efficiency.

ЖАРІКОВ Едуард В'ячеславович, канд. техн. наук, доцент кафедри АСОІУ Національного технічного університету України «Київський політехнічний ін-т ім. Ігоря Сікорського». В 1994 р. закінчив Східно-Український державний університет. Область наукових досліджень — ІТ інфраструктура, віртуалізація, хмарні обчислення, центри обробки даних, комп'ютерні мережі.