

Оптимізація параметрів функціонування системи керування іт-інфраструктурою датацентру

В. В. Москаленко, С. В. Пімоненко

Розроблено алгоритм навчання системи керування датацентром з використанням системи допусків на значення ознак для кожного з класів розпізнавання. Це дозволяє застосувати нормовані статистики кількості потраплянь ознак у поля допусків для визначення моменту перенавчання системи та підвищити достовірність рішень. Досліджено ефективність використання адитивно-мультиплікативної та ентропійної згорток частинних критеріїв якості функціонування датацентру

Ключові слова: хмарні сервіси, датацентр, інформаційний критерій, машинне навчання, ройовий алгоритм

1. Вступ

Останнім часом корпоративні додатки, електронна пошта, пошукові системи та електронна комерція все частіше розгортаються в обчислювальному середовищі хмарних датацентрів. Конкурентоздатність хмарних провайдерів визначається можливостями безвідмовної роботи датацентру у форматі 24/7. При цьому внаслідок високого рівня споживання електроенергії в хмарних датацентрах провайдери намагаються максимізувати ефективність використання електроенергії шляхом перерозподілу віртуальних ресурсів і відключення простоюючих фізичних серверів. Однак при мінімізації затрат оператор хмари повинен забезпечувати відповідність метрик якості сервісу вимогам угоди про рівень обслуговування (Service Level Agreement, SLA), оскільки порушення умов SLA призводить до штрафних санкцій та відтоку клієнтів [1].

Розроблення ефективних алгоритмів розподілу ресурсів у хмарному середовищі ускладнено відсутністю або неточністю інформації про ресурсні вимоги та поведінку конкретного завдання на навантажених гетерогенних вузлах датацентру. Априорна невизначеність функціонального стану вузла при виконанні завдання і неспроможність точної оцінки часу його виконання можуть призвести до виділення надлишкових ресурсів, які будуть простоювати, знижуючи завантаженість обчислювального середовища, або виділення недостатнього обсягу ресурсів, що призводить до накладних витрат, пов'язаних з процесом введення до експлуатації додаткових ресурсів чи міграцією задач на інший вузол. Крім того, здійснення моніторингу віртуальних машин засобами їх операційних систем чи гіпервізора пов'язане із споживанням додаткових ресурсів та некоректністю роботи засобів моніторингу при значних зниженнях продуктивності.

Найбільш ефективними інструментами підтримки безвідмовного та ефективного функціонування датацентру є аналітичні засоби аналізу архівних даних моніторингу компонентів ІТ-інфраструктури та суб'єктивно-статистичних дос-

ліджень оцінки якості сервісів [2]. Метою аналітичних засобів є формування в процесі машинного навчання чи самонавчання вирішальних правил для вчасного виявлення (активне детектування) чи прогнозування (проактивне детектування) аномальних функціональних станів компонентів ІТ-інфраструктури датацентру та аномальної поведінки користувачів чи хмарних додатків. Отримані вирішальні правила дозволяють в процесі функціонування системи керування датацентром знімати невизначеність щодо функціонального стану ІТ-інфраструктури датацентру і як наслідок підвищити ефективність дій щодо реконфігурації та перерозподілу ресурсів.

Задача розподілу ресурсів датацентру є багатокритеріальною, оскільки необхідно одночасно забезпечити мінімум енергоспоживання, обсягу невикористаних ресурсів та порушень SLA-угод. Однак ці частинні критерії є попарно суперечливими, мають різну розмірність і є нелінійними функціями контрольованих характеристик і конфігурацій ІТ-інфраструктури хмарного датацентру. Подібні задачі в компромісній області мають безліч оптимальних у розумінні Парето рішень. При цьому у задачах пошуку оптимальних за Парето рішень для багатокритеріальних задач оптимізації найбільш перспективним є використання ідей та методів ройового інтелекту, що дозволяє підвищити оперативність рішень для динамічно змінюваних умов функціонування.

Таким чином, розробка нових схем кодування ознак порушення умов SLA в алгоритмах інформаційно-екстремального машинного навчання та способів згортки частинних критеріїв ефективності функціонування датацентру в популярних алгоритмах пошуку є актуальними напрямками дослідження, що спрямовані на зниження енергозатрат та підвищення якості обслуговування кінцевих користувачів.

2. Аналіз літературних даних і постановка проблеми

При розробці автономних датацентрів з властивостями самоконфігурування, самонастроювання та самодіагностики набуває важливого значення задача прогнозування функціонального стану компонентів ІТ-інфраструктури та сервісів. Для прогнозування функціонального стану фізичних серверів при розміщенні на них нової віртуальної машини в працях [3, 4] було запропоновано використовувати багатошарову нейронну мережу з обчисленням ознак розпізнавання за принципом мішка слів (Bag-of-Words). Вхідний математичний опис такої прогностичної моделі формується шляхом пошуку в архівній історії функціонування датацентру моментів зниження продуктивності його вузлів та формування за результатами кластер-аналізу архівних даних моніторингу словника віртуальних машин з різними шаблонами споживання ресурсів. Основними недоліками такого підходу є ігнорування категорійних контекстних даних і помітне зниження оперативності навчання та розпізнавання при розширенні словника ознак та алфавіту класів. При цьому, як було показано в працях [5, 6], зниження продуктивності віртуальних машин нерідко призводить до затримок та некоректності в роботі засобів моніторингу, вбудованих в гіпервізор чи операційну систему віртуальної машини, що є причиною помилкових рішень. В працях [7, 8] розглядається використання класифікатора Байеса для прогнозування зни-

ження продуктивності вузлів хмарного кластера при призначенні на нього задачі з черги. Для уникнення перевантажень та некоректної роботи системи моніторингу авторами пропонується встановлювати конфігураційний поріг доступних ресурсів нижчим за максимальну ємність фізичного вузла. При цьому словник ознак класифікатора містить статичні та динамічні ресурсні ознаки задачі та вузла призначення, а алфавіт класів розпізнавання характеризує успішність виконання задачі. Однак статистичний метод машинного навчання класифікатора обмежує його ефективність в умовах незбалансованих та неоднорідних наборів навчальних даних, що має місце на практиці. Крім того, задача прогнозування зниження функціональної ефективності сформованих вирішальних правил досі не розглядалася.

В працях [9, 10] розглядається застосування грубого двійкового кодування спостережень, що дозволяє уніфікувати подання різнотипних ознак та прискорити обробку вхідних даних в режимах навчання та прийняття рішень. При цьому, в праці [10] було обґрунтовано використання логарифмічного інформаційного критерію функціональної ефективності (КФЕ) навчання для побудови високодостовірних вирішальних правил за малими навчальними вибірками. Запропонована схема кодування ознак дозволяє використовувати нормовану статистику числа потраплянь значень ознак у свої поля контрольних допусків як прогностичну функцію ефективності вирішальних правил [10]. Однак даний підхід передбачає вибір одного, базового, класу, відносно якого визначається верхня і нижня межі контрольних допусків на ознаки розпізнавання. Закодовані спостереження класів розпізнавання характеризують відхилення певного рівня та напрямку лише від базового класу, що призводить до втрати частини значущої статистичної інформації про взаємне розміщення в просторі ознак спостережень будь-якої пари класів. З метою підвищення достовірності та завадозахищеності вирішальних правил може розглядатися альтернативна схема кодування, у якій система контрольних допусків на значення ознак будується відносно кожного класу розпізнавання, однак особливості реалізації та ефективність такого підходу досі не були дослідженими. При цьому підвищення інформаційної спроможності вирішальних правил є актуальною задачею, оскільки ряд частинних критеріїв ефективності функціонування датацентру обчислюються за результатами прогнозу функціонального стану його компонентів та сервісів.

Для спрощення багатокритеріальної задачі керування параметрами функціонування датацентру її, як правило, зводять до однокритеріальної багатопараметричної задачі оптимізації. У праці [11] було запропоновано евристичний алгоритм розміщення віртуальних машин на фізичних серверах, оснований на максимізації адитивної згортки частинних критеріїв, однак використання цієї згортки має сенс лише у випадку випуклості множини допустимих рішень і як показано в праці [12] на практиці це нерідко призводить до нестійкості рішень. В праці [13] пропонується мультиплікативна згортка частинних критеріїв, яка з успіхом використовується у багатьох економічних задачах, однак умовою парето-оптимальності рішень за цією згортокою крім випуклості множини допустимих рішень є увігнутість функції логарифму згортки відносно кожного з частинних критеріїв. В працях [14, 15] показано, що на адекватність способу згор-

тки частинних критеріїв впливає топологічний розподіл аналізованих альтернатив в просторі частинних критеріїв – випуклість чи увігнутість області Парето відносно кожної пари координат. При цьому різні способи згортки частинних критеріїв можуть призводити до суттєво відмінних результатів, а вибір процедури згортки не може бути до кінця формалізований і визначається специфікою задачі, цілями, досвідом та інтуїцією дослідника. Тому дослідження та аналіз алгоритмів формування узагальненого показника ефективності функціонування датацентру є важливими задачами інформаційного синтезу системи керування IT-інфраструктурою датацентру.

3. Ціль і задачі дослідження

Мета даної роботи полягає в підвищенні ефективності функціонування здатної навчатися системи керування хмарним датацентром в умовах гетерогенності фізичних вузлів та сервісів.

Для досягнення поставленої мети пропонується розв'язання таких задач:

- розробити алгоритми інформаційно-екстремального машинного навчання системи керування датацентром з використанням системи контрольних допусків на значення ознак, яка окремо визначається для кожного класу розпізнавання;
- розробити алгоритм прогнозування моменту зниження функціональної ефективності прогностичних вирішальних правил з метою визначення моменту перенавчання системи керування датацентром;
- визначити оптимальний спосіб згортки частинних критеріїв для задачі розміщення віртуальних машин на фізичних серверах хмарного датацентру.

4. Алгоритми функціонування інтелектуальної системи керування IT-інфраструктурою датацентру

Алгоритм функціонування інтелектуальної системи керування датацентром повинен включати в себе процедури моніторингу обчислювального середовища та накопичення знань про взаємозв'язок небажаних функціональних станів середовища з його характеристиками та подіями в ньому, що реєструються та архівуються. При цьому основним джерелом інформації є ключові показники продуктивності (Key Performance Indicators, KPI), ключові показники якості KQI (Key Quality Indicator, KQI) та системні повідомлення, які зчитуються на різних рівнях хмарної системи. На рис. 1 показано узагальнену структурну схему системи керування датацентром.

Оцінка поточного функціонального стану може бути здійснена шляхом перевірки виконання умов договору про рівень обслуговування (Service Level Agreement, SLA), що містить граничні значення цільових параметрів (Service Level Objectives, SLO), до яких належить і сприйняття якості обслуговування (Quality of Experience, QoE). При використанні категоріальних ознак з метою врахування частоти їх появи в тому чи іншому функціональному стані компонента датацентру здійснюється їх частотне перекодування, при якому кожне значення категоріальної ознаки представляють частотами її появи в кожному з класів розпізнавання. При цьому процес кодування векторів спостережень пропонується здійснювати шляхом порівняння значення i -ї ознаки з відповідним

нижнім $A_{L,m,i}$ та верхнім $A_{U,m,i}$ контрольними допусками класу розпізнавання, які розраховуються за формулами

$$A_{L,m,i} = \bar{y}_{m,i} \left[1 - \frac{\delta_{m,i}}{\delta_{\max}} \right], \quad (1)$$

$$A_{U,m,i} = \bar{y}_{m,i} \left[1 + \frac{\delta_{m,i}}{\delta_{\max}} \right], \quad (2)$$

де $\bar{y}_{m,i}$ – усереднене значення ознаки в базовому класі; $\delta_{m,i}$ – параметр поля контрольних допусків для i -ї ознаки розпізнавання; δ_{\max} – максимальне значення параметра поля контрольних допусків.



Рис. 1. Узагальнена структурна схема системи керування датацентром

Формування бінарної навчальної матриці $\{x_{m,i}^{(j)} \mid i = \overline{1, N}; j = \overline{1, n}; m = \overline{1, M}\}$, де N – кількість ознак розпізнавання, n – кількість векторів-реалізацій в класі та M – кількість класів функціонального стану, здійснюється за правилом

$$x_{m, M*(i-1)+k}^{(j)} = \begin{cases} 1, & \text{if } A_{L,k,i} \leq y_{m,i}^{(j)} \leq A_{U,k,i}; \\ 0, & \text{else;} \end{cases} \quad k = \overline{1, M}. \quad (3)$$

Запропонована схема кодування (3) дозволяє збільшити різноманітність двійкових векторів-реалізацій та враховувати рівень та напрям відхилення розподілу векторів-реалізацій кожної пари образів між собою.

Ітераційна процедура оптимізації параметра полів контрольних допусків на ознаки розпізнавання здійснюється шляхом максимізації в процесі навчання системи керування усередненого за алфавітом класів розпізнавання інформаційного критерію функціональної ефективності навчання (КФЕ)

$$\{\delta_i^* | i = \overline{1, n}\} = \arg \max_{G_\delta} \left\{ \max_{G_E \cap G_d} \overline{E} \right\}, \quad (4)$$

де $\overline{E}^{(k)}$ – усереднене значення інформаційного КФЕ; G_δ – область допустимих значень параметра полів контрольних допусків на значення ознак; G_E – допустима область визначення функції інформаційного КФЕ; G_d – область допустимих значень радіусу гіперсферичних контейнерів класів розпізнавання, що відновлюються у бінарному просторі Хеммінга.

Процедура оптимізація радіуса контейнерів є вкладеною в процедуру (4) і може здійснюватися методом прямого перебору з заданим кроком, оскільки кількість кроків такого пошуку є відносно малою.

Як критерій ефективності машинного навчання класифікатора розглядається модифікація інформаційної міри Кульбака [10]:

$$E_m = \left[1 - (\alpha_m^{(k)} + \beta_m^{(k)}) \right] * \log_2 \left[\frac{2 - (\alpha_m^{(k)} + \beta_m^{(k)})}{(\alpha_m^{(k)} + \beta_m^{(k)})} \right], \quad (5)$$

де α_m – помилка другого роду при розпізнаванні реалізацій класу X_m^o ; β_m – помилка другого роду.

Допустима область визначення функції інформаційного КФЕ (5) обмежена нерівностями $\alpha_m < 0,5$, $\beta_m < 0,5$ та $d_m < d(x_m \oplus x_c)$, де d_m – радіус гіперсферичного контейнера класу X_m^o , $d(x_m \oplus x_c)$ – кодова відстань між усередненим вектором класу X_m^o та найближчого до нього усередненим вектором класу X_c^o .

Для прогнозування моменту зниження функціональної ефективності інформаційно-екстремальних вирішальних правил на останньому етапі машинного навчання здійснюється відображення двійкової матриці $\|x_{m,i}^{(j)}\|$ на множину вільних статистик, інваріантних до широкого сімейства законів розподілу ймовірностей, з подальшим формуванням варіаційного ряду екстремальних порядкових статистик (ЕПС) $\langle \{S_{m,n}^*\} \rangle$. При цьому як одновимірною статистичною характеристикою вибіркової множини розглядається нормована статистика числа потраплянь ознак у свої поля контрольних допусків за n випробувань [10]

$$S_{m,n} = \sum_{j=1}^n \left(\frac{k_{m,j} - \bar{k}_{m,n}}{s_{m,n}} \right)^2, \quad m = \overline{1, M}, \quad (6)$$

де $k_{m,j}$ – число успіхів при j -му випробуванні; $\bar{k}_{m,n}$ – вибіркове середнє числа успіхів після n випробувань; $s_{m,n}^2$ – вибіркова незміщена дисперсія за n випробувань.

Статистика (6) має розподіл χ^2 і залежить тільки від обсягу випробувань n . Вільна статистика $S_{m,n}$ є членом варіаційного ряду – порядковою статистикою, ранг якої визначається номером кроку навчання. При цьому нижні та верхні довірчі межі блоків варіаційного ряду $\langle \{S_m^*\} \rangle$ рекомендується обчислювати поділивши відстані між сусідніми ЕПС на рівні частини.

У робочому режимі системи керування визначається поточний клас функціонального стану віртуальної машини та формується поточна ЕПС. Якщо ЕПС виходить за межі варіаційних блоків, то приймається рішення про необхідність перенавчання системи, внаслідок зміни структури споживання ресурсів віртуальними машинами датацентру.

Кожне z -те рішення P_z щодо розподілу фізичних ресурсів датацентру між R віртуальними машинами, які очікують рішення в часовому вікні Δt , кодується вектором натуральних чисел

$$P_z = \langle p_{z,1}, \dots, p_{z,r}, \dots, p_{z,R} \rangle, \quad (7)$$

де $p_{z,r}$ – номер фізичного вузла на який призначається r -та віртуальна машина.

В робочому режимі система керування повинна приймати оптимальне оперативне рішення P_z^* щодо розподілу ресурсів датацентру. Оцінка оптимальності рішення відбувається на кожному кроці пошуку глобального екстремуму згортки частинних критеріїв, до яких належить рівень енергоспоживання та прогнозованих порушень умов SLA, обсяг невикористаних ресурсів та інші.

У загальному випадку для надання рівномірності впливу кожного з частинних критеріїв на значення згортки необхідно вирівняти діапазони зміни значень частинних критеріїв шляхом масштабування і зведення їх значень до безрозмірної шкали $[0,1]$ за правилом

$$k_i' = \begin{cases} 0, & k_i \leq k_i^{\min}; \\ \frac{(k_i - k_i^{\min})}{(k_i^{\max} - k_i^{\min})}, & k_i^{\min} < k_i < k_i^{\max}; \\ 1, & k_i > k_i^{\max}; \end{cases}$$

де k_i^{\min} , k_i^{\max} – відповідно нижня та верхня межі допустимої області значень i -го частинного критерію.

Під час нормування та формування формули згортки необхідно враховувати, що часткові критерії не є односпрямованими: частина часткових критеріїв має бути максимізована, частина – мінімізована. Тому часткові критерії поділяють на стимулятори (які мають бути максимізовані) та дестимулятори (які повинні мінімізуватися). Формула нормування стимуляторів може бути спрощена і мати вигляд

$$k'_i = \frac{k_i}{k_i^{\max}}. \quad (8)$$

Формула нормування дестимуляторів може бути спрощена аналогічно

$$k'_i = \frac{k_i^{\min}}{k_i}. \quad (9)$$

Для охоплення широкого кола задач було розроблено велику кількість модифікацій адитивно-мультиплікативних згорток як класичного виду, так і побудованих на основі поліному Колмогорова-Габора [16]. В працях [15, 16] було запропоновано експоненціальну та ентропійну згортки, які охоплюють широке коло задач, мають просту формулу обчислення і мінімальну кількість параметрів. З урахування (8) та (9) адитивно-мультиплікативну згортку можна представити у вигляді

$$F = \sum_{i=1}^{K_1} \omega_i \frac{k_i}{k_i^{\max}} + \sum_{i=K_1+1}^{K_2} \omega_i \frac{k_i^{\min}}{k_i} + \prod_{i=1}^{K_1} \left(\frac{k_i}{k_i^{\max}} \right)^{\omega_i} \prod_{i=K_1+1}^{K_2} \left(\frac{k_i^{\min}}{k_i} \right)^{\omega_i}, \quad (10)$$

де ω_i – вага (пріоритет) i -го критерію, для якого повинна виконуватися умова

$$\sum_{i=1}^{K_1+K_2} \omega_i = 1.$$

Формула згортки, побудована за принципом інформаційної ентропії, відбиває змістовне наповнення поняття корисності як інформаційної категорії і має такий вигляд

$$F = \sum_{i=1}^{K_1} \omega_i \left(\frac{k_i}{k_i^{\max}} \right)^{\omega_i} + \sum_{i=K_1+1}^{K_2} \omega_i \left(\frac{k_i^{\min}}{k_i} \right)^{\omega_i}. \quad (11)$$

Значення ваги кожного частинного критерію може бути обчислене на основі методів, які ґрунтуються на попарному порівнянні критеріїв чи аналітичній залежності показників важливості критеріїв, та формальних методів, таких як

метод базового критерію чи метод Черчмена-Акоффа [17]. Проте в умовах априорної невизначеності відповідно до принципу Бернуллі-Лапласа можна прийняти вагу критеріїв однаковою і рівною

$$\omega_i = \frac{1}{K_1 + K_2}.$$

Гетерогенність, ієрархічність інформаційно-телекомунікаційного середовища, багатофакторність та нестаціонарність процесів споживання ресурсів обумовлюють нелінійність, багатоекстремальність та високу розмірність оптимізаційної задачі. З метою підвищення оперативності пошуку як оптимального рішення щодо розподілу ресурсів, так і оптимального вектора параметрів прогностичних вирішальних правил має сенс використати популяційні алгоритми, в основі яких лежать ідеї, запозичені в природи, а також базові постулати універсальності, фундаментальності, властиві самоорганізації природних систем. В даній роботі пропонується обрати популяційний алгоритм пошуку косяком риб (Fish School Search, FSS), який відрізняється простотою реалізації, інтерпретабельністю та високою швидкістю збіжності [18]. В цьому алгоритмі позиція агента в N-вимірному просторі рішень представлена у вигляді числового вектору $P^{[N]}$ довжиною N, який відповідає вектору параметрів, що оптимізується.

В алгоритмі FSS косяк риб є агрегацією агентів популяції, які рухаються приблизно з однією й тією ж швидкістю і орієнтацією, підтримуючи приблизно однакову відстань між собою. Індивідуальний успіх кожної риби в процесі пошуку рішення характеризується її вагою, що відіграє роль пам'яті. Кожна ітерація пошуку виконує дві групи операторів – оператори годування та оператори плавання.

Оператор годування формалізує успішність дослідження агентами тих чи інших областей “акваріуму” і полягає в обчисленні ваги z-го агента, яка пропорційна нормалізованій різниці значень фітнес-функції на наступній та поточній ітераціях

$$w_z[k+1] = w_z[k] + \frac{J(P_z[k+1]) - J(P_z[k])}{\max(J(P_z[k+1]), J(P_z[k]))}, \quad z = \overline{1, Z},$$

де $P_z[k+1]$, $P_z[k]$ – позиція z-го агента в багатовимірному просторі рішень на k-й та (k+1)-й ітерації алгоритму FSS.

Максимально можливе значення ваги агента w_z у алгоритмі FSS обмежується значенням $w_{\max} > 0$. При цьому під час ініціалізації популяції всім агентам присвоюється вага, що рівна $w_{\max} \cdot 0,5$.

В алгоритмі FSS розрізняють три види плавання – індивідуальне, інстинктивно-колективне та колективно-вольове. Ці види плавання здійснюються послідовно один за одним в окремі інтервали часу

$$(t_1, t_2], (t_2, t_3], (t_3, t_4), \quad t_1 < t_2 < t_3 < t_4, t_4 = t_1 + 1.$$

Під час індивідуального плавання агентів відбувається їх переміщення, що має рівноймовірний випадковий характер. При цьому за одну ітерацію алгоритму FSS крок індивідуального плавання виконується фіксовану кількість разів. Компоненти кроку переміщення V_z^{ind} рівномірно розподілені в заданому інтервалі v_{\max}^{ind}

$$V_z^{ind} = U(0;1)v_{\max}^{ind}, \quad z = \overline{1, Z},$$

де $U(0;1)$ – випадкове число із діапазону $(0;1)$.

В процесі інстинктивно-колективного плавання на кожного з агентів чинять вплив всі інші агенти популяції і цей вплив пропорційний індивідуальним успіхам агентів. При цьому позиції агентів обчислюються за формулою

$$P_z^{t_3} = P_z^{t_2} + \frac{\sum_j V_j^{ind}(t_2)(J(P_j^{t_2}) - J(P_j^{t_1}))}{\sum_j J(P_j^{t_2}) - J(P_j^{t_1})}, \quad j = \overline{1, Z}. \quad (12)$$

Колективно-вольове плавання полягає у зміщенні всіх агентів у напрямку поточного центру тяжіння популяції за умови збільшення сумарної ваги косяка риб в результаті індивідуального та інстинктивно-колективного плавання. Якщо сумарна вага зменшилась, то зміщення відбувається в протилежному напрямку. Колективно-вольове плавання виконується за правилами

$$P_z' = P_z^{t_3} \pm v^{vol}(P_z^{t_3} - P_c^{t_3}), \quad z = \overline{1, Z}, \quad (13)$$

де $P_c^{t_3}$ – координати центра тяжіння косяка риб, що визначаються за формулою

$$P_c^{t_3} = \frac{\sum_z w_z^{t_3} P_z^{t_3}}{\sum_z w_z^{t_3}}.$$

У формулі (6) знак плюс використовується за умов $\sum_z w_z^{t_3} > \sum_z w_z^{t_3-1}$, а знак мінус – в протилежному випадку. При цьому розмір кроку переміщення агентів v^{vol} є випадковою величиною

$$v^{vol} = v_{\max}^{vol} U(0;1),$$

де v_{\max}^{vol} – знакододатне значення максимально допустимої довжини кроку переміщення при колективно-вольовому плаванні.

Таким чином, запропонований алгоритм функціонування системи керування датацентром оснований на зведенні багатокритеріальної задачі оптимізації до однокритеріальної та пошуці за популяційним алгоритмом косяка риб глобального максимуму узагальненого критерію ефективності функціонування датацентру. При цьому такі частинні критерії як рівень дотримання умов SLA знаходяться шляхом прогнозування, де прогностичні вирішальні правила формуються в процесі інформаційно-екстремального машинного навчання за вибілковими даними моніторингу і автоматично корегуються при зміні структури споживання віртуальними машинами фізичних ресурсів датацентру.

5. Результати фізичного моделювання інтелектуальної системи керування ІТ-інфраструктурою дата центру

Для фізичного моделювання було створено датацентр на базі 20 серверів з процесорами типу Intel (в тому числі Atom і Xeon), які відрізняються між собою кількістю ядер, розміром кеш-пам'яті, тактовою частотою та споживанням енергії. На серверах розгортаються віртуальні машини, одна частина з яких виконує функції віртуальних вузлів Hadoop кластера для обробки гетерогенних задач розподіленої обробки даних за технологією MapReduce, таких як PiEstimator (оцінка числа Pi з точністю 1-го мільйонного знаку), WordCount (обчислення частоти появи слів в 15GB даних); Sort (сортування 18GB даних), Grep (пошук співпадінь за випадково обраним регулярним виразом в 6GB даних), TeraSort (сортування 1GB даних), Kmeans (кластер-аналіз 6GB числових даних). Інша частина віртуальних машин виконує функції веб-серверів, на базі яких працюють сервіси онлайн доступу до книг та додатки типу Micro Web App чи PHP/MySQL. Робоче навантаження на сервери генерується клієнтами, що запрограмовані на формування сумарної мінімальної потреби в ресурсах, яка змінюється в часі за законом, який показано на рис. 2.

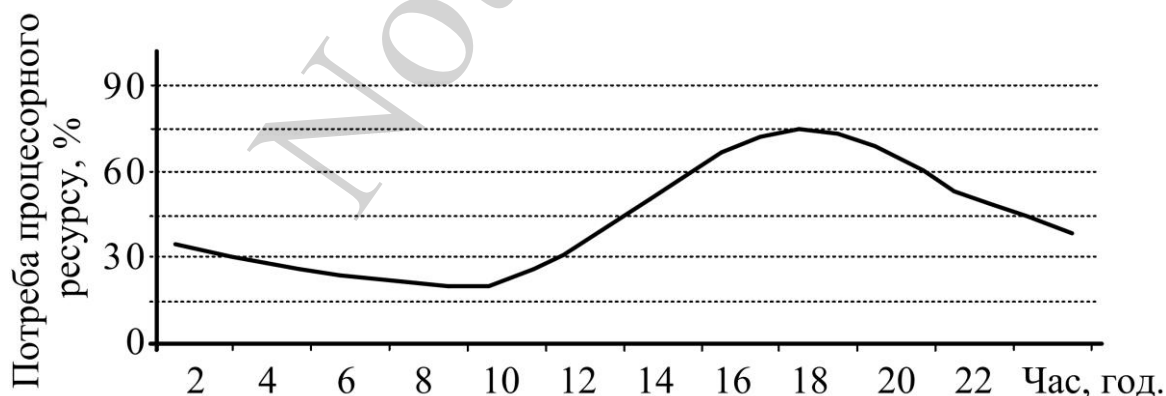


Рис. 2. Графік зміни навантаження датацентру протягом доби

Як інструмент розгортання хмарної платформи датацентру використано вільне програмне забезпечення Apache CloudStack, написане на мові java [19]. Функцію гіпервізора виконує XenServer [19]. Сховище даних керується за допомогою NetApp та Cumulus [19]. Apache CloudStack підтримує різноманітні

алгоритми керування розміщенням віртуальних машин, наприклад, `firstfit`, `random`, `worstfit` та інші, проте в даному дослідженні було розроблено власний планувальник шляхом наслідування від класу `nova.scheduler.driver.Scheduler` та перевизначення згідно із метою дослідження наступних методів `schedule_run_instance` та `select_destinations`.

Задача консолідації віртуальних машин датацентру поділяється на чотири підзадачі: детектування недовантаження серверів; детектування перевантаження серверів; відбір віртуальних машин для переміщення; розміщення віртуальних машин на серверах. Для простоти порівняння спочатку було реалізовано алгоритм розміщення віртуальних машин MBFD (Modified Best Fit Decreasing), основна ідея якого полягає в сортуванні віртуальних машин, що підлягають міграції, в порядку спадання їх ресурсних потреб і їх призначенні на найбільш енергоефективні сервери, що мають достатній обсяг ресурсів [20]. При цьому за замовчуванням алгоритм детектування недовантаження знаходить найменш завантажені сервери і намагається перемістити його віртуальні машини за алгоритмом MBFD на інші сервери. Для дослідження запропонованих алгоритмів їх ефективність буде порівнюватися з алгоритмом MBFD.

В даному дослідженні пропонується здійснювати оптимізацію розподілу віртуальних машин шляхом популяційного пошуку за інтегральними критеріями (10) та (11). При цьому з метою врахування гетерогенності віртуальних машин та серверів пропонується перед переміщенням прогнозувати порушення умов SLA внаслідок конкуренції віртуальних машин на спільні фізичні ресурси.

Навчання системи керування прогнозуванню порушень умов SLA внаслідок конкуренції віртуальних машин за ресурси фізичних серверів здійснюється як для рівня IaaS (Infrastructure-as-a-Service), так і для рівня SaaS (Software-as-a-Service). Тому для запитів клієнтів було прикріплено умови користувацького SLA у вигляді таких показників:

- максимальний час, протягом якого користувач згоден очікувати результат, T_d ;
- ціна яку користувач готовий заплатити за послугу, V_d ;
- розмір компенсації за порушення дедлайну, C_p ;
- обсяг файлів, що надсилають користувачі, F_s ;
- довжина запиту користувача (в мільйонах інструкцій, що виконуватимуться на віртуальному вузлі), M_i .

Кожен фізичний вузол в експериментальному середовищі було описано ресурсним SLA у вигляді показників:

- час розгортання віртуальної машини, T_{vm} ;
- вартість години використання віртуальної машини, C_{vm} ;
- вартість переміщення даних між користувачами та віртуальною машиною, C_{fs} ;
- швидкість опрацювання задач користувача в мільйонах інструкцій за секунду, M ;
- швидкість передачі даних між користувачем та фізичним вузлом, S_{fs} .

У зв'язку з обмеженістю ресурсів датацентру прогнозування порушень умов SLA внаслідок конкуренції віртуальних машин за спільні фізичні ресурси дозволяє приймати рішення оптимальні у вартісному сенсі. При цьому повторюваний харак-

тер задач, що вирішуються додатками віртуальних машин, забезпечує можливість застосування методів машинного навчання для аналізу log-даних трасування роботи віртуальних машин для синтезу прогностичної моделі. Вхідний математичний опис такої моделі включає результати кластер-аналізу даних трасування роботи віртуальних машин за алгоритмом k-середніх. Утворені таким чином групи (класи) віртуальних машин характеризують існуючі в датацентрі шаблони споживання різного типу ресурсів фізичного сервера. При цьому словник ознак класифікатора віртуальних машин включає середній обсяг використання ресурсу процесорів, оперативної пам'яті, файлу підкачки, мережевого каналу, дискового простору, середню інтенсивність операцій вводу-виводу з дисковою пам'яттю.

Для навчання прогностичного класифікатора порушення умов SLA внаслідок розміщення обраної віртуальної машини на обраному фізичному вузлі пропонується використати такий словник ознак:

- нормалізовані відносно потреб обраної віртуальної машини обсяги вільних ресурсів (CPU, RAM, I/O Disk, Network) обраного фізичного вузла;
- нормалізовані кількості віртуальних машин кожного класу (6 кластерів) на обраному фізичному вузлі з урахуванням обраної віртуальної машини, що планується на ньому розмістити;
- розрахунковий час виконання запиту $T_e = T_d - T_{vm} - F_s / S_{fs} - M_i / M$;
- розрахунковий залишок бюджету $B_r = B_d - C_{fs} * F_s / S_{fs} - C_{vm} * M_i / M$.

Визначення належності вектора розглянутих ознак до одного з класів порушення умов SLA з метою формування навчальної вибірки здійснюється за результатами фіксації в даних моніторингу таких подій: задача направлена планувальником на хостову машину для запуску; для задачі виконується перепланування (міграція); задача виконана без порушень умов SLA.

На рис. 3 показано графік залежності усередненого значення нормованого інформаційного КФЕ (5) від кількості ітерацій оптимізації параметра поля контрольних допусків на значення ознак (4) за ройовим алгоритмом пошуку.

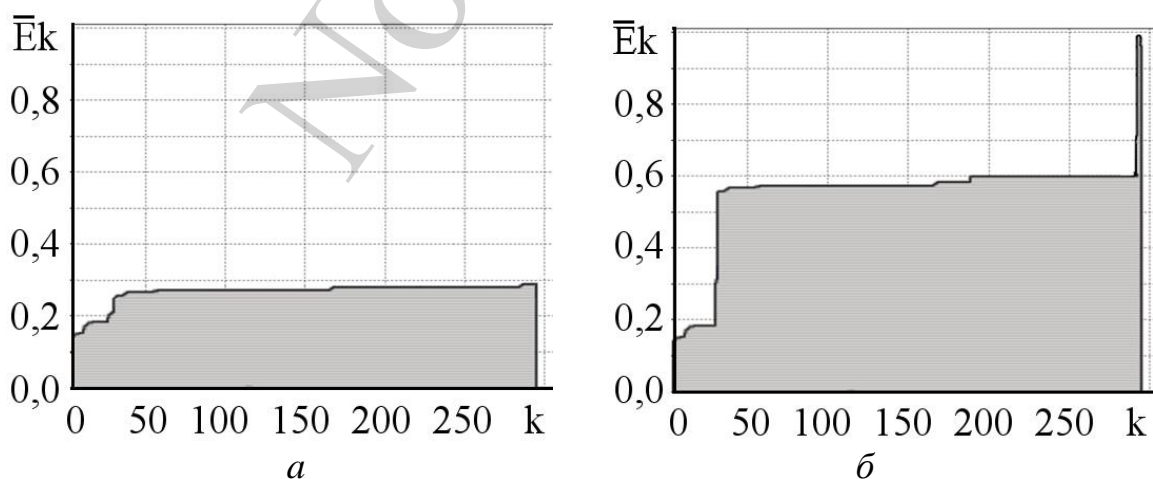


Рис. 3. Графік зміни максимумів усередненого за алфавітом класів значення нормованого критерію (5) в процесі ройової оптимізації системи контрольних допусків: *а* – однорівнева система контрольних допусків; *б* – багаторівнева система контрольних допусків

Аналіз рис. 3, а показує, що інформаційно-екстремальне машинне навчання за триційним алгоритмом з єдиним базовим класом не дозволяє отримати високодостовірні вирішальні правила, а відповідне значення глобального максимуму усередненого нормованого КФЕ навчання класифікатора становить $\bar{E}^* = 0,29$, що відповідає такому значенню точності $\overline{P_{true}}^* = 0,92$. Аналіз рис. 3, б показує, що побудова системи контрольних допусків для кожного класу розпізнавання $\{\delta_{i,m}^*, i = \overline{1, N}; m = \overline{1, M}\}$ дозволяє отримати граничне значення інформаційного критерію $\bar{E}^* = 1,0$ та отримати безпомилкові за навчальною матрицею вирішальні правила, тобто $\overline{P_{true}}^* = 1,0$.

На рис. 4 наведено графіки залежності нормованого КФЕ (5) від радіусів контейнерів відповідних класів при оптимальній системі контрольних допусків на значення ознак розпізнавання.

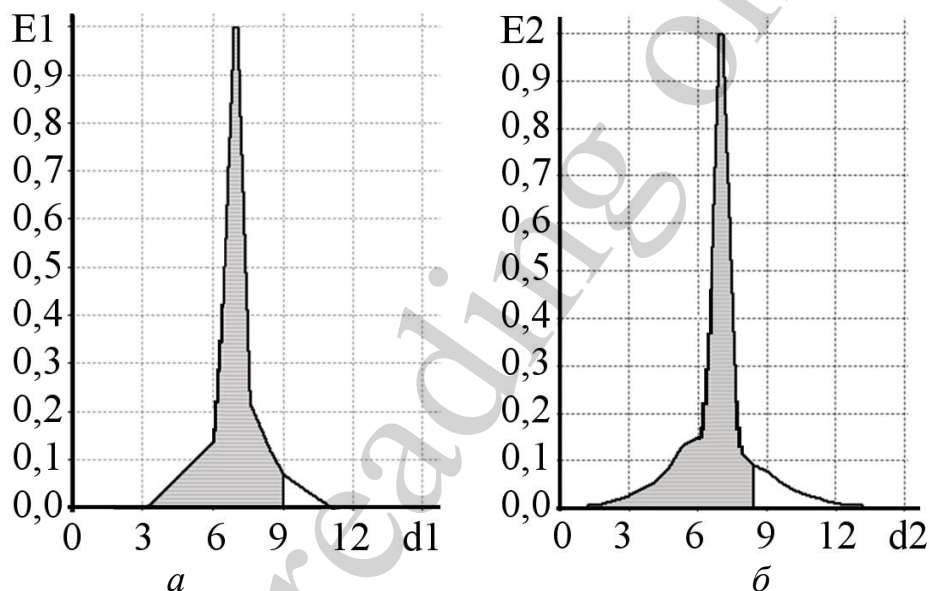


Рис. 4. Графіки залежності нормованого КФЕ (5) від радіусів контейнерів класів: а – X_1^o ; б – X_2^o

Аналіз рис. 4 показує, що максимальні значення КФЕ навчання для класів X_1^o та X_2^o дорівнюють $E_1^* = 1,0$ та $E_2^* = 1,0$ відповідно, а оптимальні значення радіусів відповідних контейнерів класів розпізнавання – $d_1^* = 7$, $d_2^* = 7$ (в кодових одиницях). Граничні значення КФЕ для класів X_1^o та X_2^o свідчать про побудову для них безпомилкових за навчальною матрицею вирішальних правил. При цьому міжцентрова кодова відстань становить $d(x_1 \otimes x_2) = 14$, що свідчить про компактність векторів-реалізацій в субперцептуальному та чіткість розбиття в двійковому просторі Хеммінга.

На рис. 5 показано залежність операційних витрат (у відносних одиницях), які включають сумарну вартість міграції віртуальних машин, компенсацій за порушення SLA та витрати на електроенергію, на кожному кроці прийняття рі-

шень за алгоритмом MBFD та його модифікаціями з прогнозуванням порушення SLA та оптимізацією за критеріями (10) та (11) відповідно.

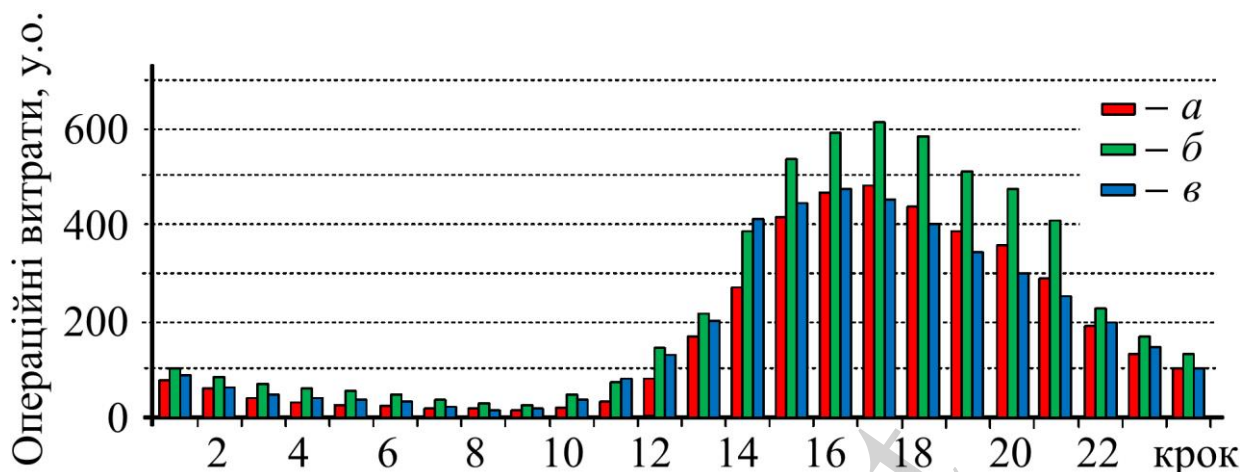


Рис. 5. Гістограма зміни операційних витрат для різних алгоритмів планування залежно від навантаження датацентру на відповідному кроці прийняття рішень: *a* – MBFD; *б* – прогностична модифікація MBFD з критерієм (10); *в* – прогностична модифікація MBFD з критерієм (11)

Аналіз рис. 5 показує, що використання прогностичних вирішальних правил дозволяє зменшити операційні витрати внаслідок зменшення кількості порушення умов SLA та зменшення простоюючих фізичних ресурсів без перевантаження навантажених фізичних вузлів. При цьому під час збільшення навантаження датацентру більш оптимальною є згортка критеріїв (10), а при зменшенні навантаження датацентру кращі результати можна отримати при використанні згортки (11).

Таким чином, розроблений алгоритм інформаційно-екстремального машинного навчання для прогнозування порушення SLA дозволяє підвищити ефективність мінімізації операційних витрат щодо керування ресурсами датацентру. При цьому одна з розглянутих згорток частинних критеріїв оптимізації є більш ефективною при збільшенні навантаження датацентру, а інша – при зменшенні потреб у фізичних ресурсах датацентру.

6. Обговорення результатів імітаційного моделювання

Як видно з рис. 2, 5, зростання операційних витрат системи керування датацентром пропорційне сумарним ресурсним потребам користувачів в обслуговуванні, однак використання популяційного пошуку з прогнозуванням порушень умов SLA дозволяє зменшити рівень витрат порівняно з алгоритмом MBFD. При цьому згортка критеріїв (10) виявилась більш чутливою до рівня енергоспоживання, що дозволяє підвищити ефективність планування на етапі наростання навантаження датацентру, в той час як згортка (11) є більш чутливою до рівня простоюючих фізичних ресурсів, що обумовлює її ефективність при зниженні навантаження датацентру.

Запропонована модифікація алгоритму інформаційно-екстремального машинного навчання за вибіркою, що сформована в процесі моніторингу роботи датацентру, керованого за алгоритмом MBFD, порівняно з традиційним алгоритмом дозволяє підвищити функціональну ефективність системи керування та отримати безпомилкові за навчальною матрицею вирішальні правила. Проте внаслідок розширення набору сервісів, що розгортаються на базі віртуальних машин датацентру, з'являються нові шаблони споживання фізичних ресурсів, що призводить до зміни статистичних характеристик образів і відповідно зменшення функціональної ефективності сформованих до цього вирішальних правил. На рис. 6, *а* показано одержані при оптимальних параметрах навчання порядкові статистики (б) за першими $n=100$ векторами-реалізаціями навчальної матриці для заданого алфавіту класів розпізнавання. На рис. 6, *б* показано зміну в часі екзаменаційних порядкових статистик у своїх варіаційних блоках впродовж n розпізнавання при зростанні попиту до нових сервісів, які не використовувалися при зборі навчальної вибірки. При цьому криві ЕПС червоного кольору відповідає статистиці для класу X_1^o , а крива синього кольору відповідає статистиці для класу X_2^o .

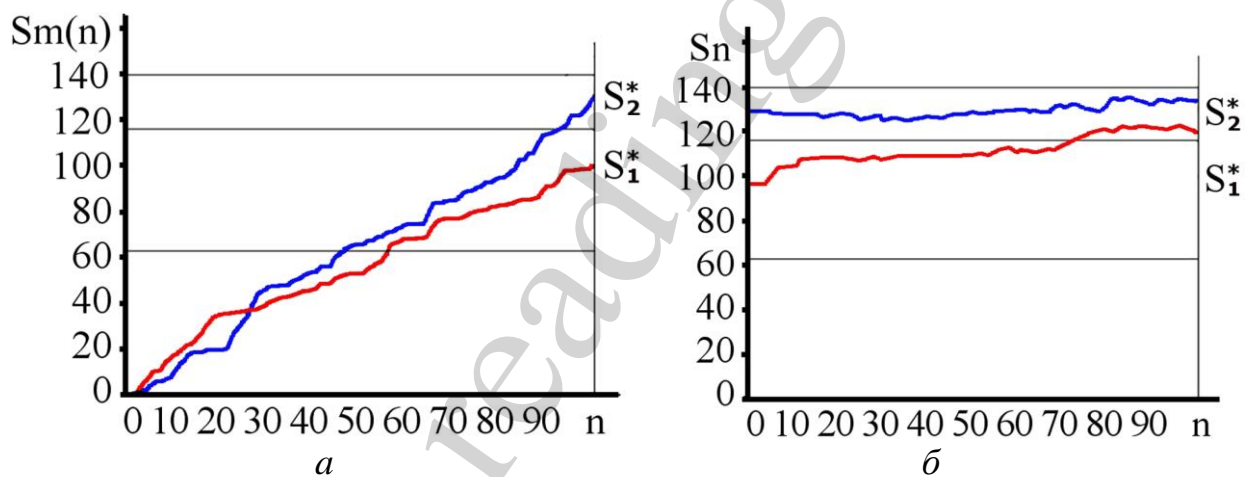


Рис. 6. Графіки залежності ЕПС: *а* – від кількості випробувань при оптимальних параметрах навчання; *б* – від кількості періодів екзамену в процесі зростання попиту на нові сервіси

Аналіз рис. 6, *а* показує що клас X_1^o , який характеризує задоволення умов SLA, має найменше значення ЕПС, рівне $S_1^* = 98,5$. Класу X_2^o , що характеризує функціональний стан порушення умов SLA, відповідає більше значення ЕПС, рівне $S_2^* = 131$. Аналіз рис. 6, *б* показує, що після зміни структури споживання ресурсів протягом 72 розпізнавань функціонального стану одновимірною статистичною характеристикою S_1 реалізацій класу X_1^o переходить у варіаційний блок класу X_2^o , що створює статистичну невизначеність. З метою запобігання зниженню функціональної ефективності системи керування в цей момент слід ініціювати її перенавчання.

Таким чином, розроблене інформаційне та алгоритмічне забезпечення інтелектуальної системи керування датацентром дозволяє визначати момент зниження її функціональної ефективності та адаптуватися до нових умов функціонування.

7. Висновки

1. Показано перевагу у застосуванні в інформаційно-екстремальному навчанні запропонованого алгоритму двійкового кодування ознак розпізнавання, у якому на відміну від традиційного алгоритму система контрольних допусків визначається для кожного з класів розпізнавання, а не лише для базового, що дозволяє збільшити кодову відстань між центрами контейнерів класів та підвищити достовірність вирішальних правил.

2. Доведено за результатами фізичного моделювання ефективність запропонованого алгоритму планування розміщень віртуальних машин на фізичних серверах з використанням інтегрального критерію оптимізації, отриманого за допомогою згортки частинних критеріїв. При цьому було відмічено, що адитивно-мультиплікативна згортка є більш чутливою до рівня енергоспоживання, що дозволяє підвищити ефективність планування на етапі наростання навантаження датацентру, в той час як ентропійна згортка є більш чутливою до рівня простуючих фізичних ресурсів, що обумовлює її вищу ефективність при зниженні навантаження датацентру.

3. Показано, що використання екстремальних порядкових статистик у вигляді нормованих статистик числа потраплянь ознак у свої поля контрольних допусків дозволяє в робочому режимі системи керування виявити появу в датацентрі нових шаблонів споживання фізичних ресурсів внаслідок активного використання нових сервісів датацентру, що дозволяє вчасно донавчити або перенавчити систему керування і підтримувати її функціональну ефективність на високому рівні.

Литература

1. Cao, Z. Dynamic VM consolidation for energy-aware and SLA violation reduction in cloud computing [Text] / Z. Cao, S. Dong // Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2012. – P. 363–369. doi: 10.1109/pdcat.2012.68

2. Sharma, B. Applications of Data Mining in the Management of Performance and Power in Data Centers [Text] / B. Sharma // Technical Report, Department of Computer Science and Engineering. The Pennsylvania State University, 2009. – P. 1–5.

3. Caglar, F. Towards a performance interference-aware virtual machine placement strategy for supporting soft real-time applications in the cloud [Text] / F. Caglar, S. Shekhar, A. Gokhale // Proceedings of the 3rd International Workshop on Real-time and Distributed Computing in Emerging Applications, 2014. – P. 15–20.

4. Delimitrou, C. Paragon: QoS-aware scheduling for heterogeneous datacenters [Text] / C. Delimitrou, C. Kozyrakis // Proceedings of the 18th international conference on Architectural support for programming languages and operating systems. – 2013. – Vol. 41. – P. 77–88. doi: 10.1145/2451116.2451125

5. Hayashi, T. Performance Degradation Detection of Virtual Machines via Passive Measurement and Machine Learning [Text] / T. Hayashi, S. Ohta // International Journal of Adaptive, Resilient and Autonomic Systems. – 2014. – Vol. 5, Issue 2. – P. 40–56. doi: 10.4018/ijaras.2014040103
6. Bodik, P. Fingerprinting the Datacenter: Automated Classification of Performance Crises [Text] / P. Bodik, M. Goldszmidt, A. Fox, D. B. Woodard, H. Andersen // Proceedings of the 5th European conference on Computer systems, 2010. – P. 111–124. doi: 10.1145/1755913.1755926
7. Nanduri, R. Job Aware Scheduling Algorithm for MapReduce Framework [Text] / R. Nanduri, N. Maheshwari, R. Raja, V. Varma // 2011 IEEE Third International Conference on Cloud Computing Technology and Science, 2011. – P. 724–729. doi: 10.1109/cloudcom.2011.112
8. Kandalintsev, A. Profiling Cloud Applications with Hardware Performance Counters [Text] / A. Kandalintsev, R. L. Cigno, D. Kliazovich, P. Bouvry // The International Conference on Information Networking, 2014. – P. 52–57. doi: 10.1109/icoin.2014.6799664
9. Dovbysh, A. S. Information-Extreme Method for Classification of Observations with Categorical Attributes [Text] / A. S. Dovbysh, V. V. Moskalenko, A. S. Rizhova // Cybernetics and Systems Analysis. – 2016. – Vol. 52, Issue 2. – P. 224–231. doi: 10.1007/s10559-016-9818-1
10. Dovbysh, A. S. Learning decision making support system for control of nonstationary technological process [Text] / A. S. Dovbysh, V. V. Moskalenko, A. S. Rizhova // Journal of automation and information sciences. – 2016. – Vol. 48, Issue 6. – P. 39–48. doi: 10.1615/jautomatinfscien.v48.i6.40
11. Chen, L. MTAD: A Multitarget Heuristic Algorithm for Virtual Machine Placement [Text] / L. Chen, J. Zhang, L. Cai, R. Li, T. He, T. Meng // International Journal of Distributed Sensor Networks. – 2015. – Vol. 2015. – P. 1–14. doi: 10.1155/2015/679170
12. Salmasnia, A. A new desirability function-based method for correlated multiple response optimization [Text] / A. Salmasnia, M. Bashiri // The International Journal of Advanced Manufacturing Technology. – 2015. – Vol. 76, Issue 5-8. – P. 1047–1062. doi: 10.1007/s00170-014-6265-x
13. Altinoz, O. T. A multiobjective optimization approach via systematical modification of the desirability function shapes [Text] / O. T. Altinoz, A. E. Yilmaz, G. Ciuprina // 8th International symposium on advanced topics in electrical engineering, 2013. – P. 1–6. doi: 10.1109/atee.2013.6563481
14. Sanginova, O. Comparative analysis of some computational schemes for obtaining a compromise solution [Text] / O. Sanginova // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 1, Issue 4 (73). – P. 10–18. doi: 10.15587/1729-4061.2015.35607
15. Shengnan, Z. Multi-response robust design based on improved desirability function [Text] / Z. Shengnan, W. Jianjun // International Conference on Grey Systems and Intelligent Services, 2015. – P. 515–520. doi: 10.1109/gsis.2015.7301911

16. Kushwaha, S. A Modified Desirability Function Approach for Mean-Variance Optimization of Multiple Responses [Text] / S. Kushwaha, S. Sikdar, I. Mukherjee, P. K. Ray // International Journal of Software Science and Computational Intelligence. – 2013. – Vol. 5, Issue 3. – P. 7–21. doi: 10.4018/ijssci.2013070101
17. Yoo, D. G. Rehabilitation Priority Determination of Water Pipes Based on Hydraulic Importance [Text] / D. G. Yoo, D. Kang, H. Jun, J. H. Kim // Water. – 2014. – Vol. 6, Issue 12. – P. 3864–3887. doi: 10.3390/w6123864
18. Parpinelli, R. Theory and New Applications of Swarm Intelligence [Text] / R. Parpinelli. – InTech, 2012. – 204 p. doi: 10.5772/1405
19. Jain, S. A Comparative Study for Cloud Computing Platform on Open Source Software [Text] / S. Jain, R. Kumar, Anamika, S. K. Jangir // An International Journal of Engineering & Technology. – 2014. – Vol. 1, Issue 2. – P. 28–34.
20. Kaur, A. Energy optimized VM placement in cloud environment [Text] / A. Kaur, M. Kalra // 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016. – P. 141–145. doi: 10.1109/confluence.2016.7508103

Not a reprint