

УДК 681.518:004.93.1'

В. В. МОСКАЛЕНКО, А. С. РИЖОВА

Сумський державний університет, Україна

ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНИЙ КЛАСИФІКАТОР З КОМБІНОВАНИМИ КОНТЕЙНЕРАМИ ДЛЯ ПЛАНУВАННЯ ЗАДАЧ В РОЗПОДІЛЕНОМУ ГЕТЕРОГЕННОМУ ОБЧИСЛЮВАЛЬНОМУ СЕРЕДОВИЩІ

Розглядається інформаційний синтез динамічного планувальника незалежних задач, заснований на класифікаційному прогнозі функціонального стану вузлів гетерогенного обчислювального середовища. У статті досліджено алгоритм інформаційно-екстремального навчання прогностичного класифікатора з комбінованими контейнерами за умов незбалансованості навчальних вибірок та різнотипності ознак розпізнавання, що має місце в практичних задачах моніторингу обчислювального середовища. Фізичне моделювання роботи запропонованих алгоритмів реалізовано в обчислювальному середовищі кластера з моделлю розподілених обчислень MapReduce на прикладі задач WordCount, Terasort та Distributed Grep.

Ключові слова: розподілені обчислення, планування задач, черга задач, гетерогенні ресурси, машинне навчання, розпізнавання образів, ознаки розпізнавання, контейнер класу розпізнавання, інформаційний критерій.

Вступ

Потреба в обчислювальних ресурсах суттєво зросла в останні десятиліття у багатьох сферах суспільно-економічної діяльності. У зв'язку з цим широкі поширення отримали системи паралельної і розподіленої обробки даних. Основним компонентом цих систем є планувальник завдань, який ініціює виділення ресурсів для завдань з урахуванням вимог до якості обслуговування користувачів та інтересів провайдерів ресурсів, які зводяться до оптимізації інтегральних характеристик розподіленого середовища, таких як час виконання пакету завдань та завантаження (утилізація) ресурсів. Складність задачі оптимального відображення завдань на ресурси є нелінійною функцією розмірності системи, а методи точного її вирішення мають високу складність і потребують невідповідно великі витрати часу. Тому на практиці використовуються різні евристичні та ймовірнісні алгоритми планування, що дають рішення близьке до оптимального.

Розроблення алгоритмів планування, спрямованих на підвищення ефективності виконання як окремих завдань, так і їх потоку, ускладнено відсутністю або неточністю інформації про ресурсні вимоги та поведінку конкретної задачі обробки даних на навантажених гетерогенних вузлах. 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 [1, 2].

Повторюваний характер задач з однаковою та схожою специфікаціями забезпечує можливість використання ідей і методів машинного навчання та розпізнавання образів з метою підвищення ефективності планувальника задач на основі знань, одержаних в процесі інтелектуального аналізу передісторії виконання задач [3-5]. При цьому гетерогенність обчислювального середовища і наявність параметричної невизначеності інформаційних процесів обумовлюють довільні умови формування образів функціональних станів обчислювальних вузлів. Це призводить до формування в просторі ознак образів складної конфігурації та їх перетину. При цьому більшість методів машинного навчання не пристосовані для аналізу спостережень за процесами розподіленої обробки даних, оскільки спостереження містять різнотипні ознаки, а архівна історія має істотно незбалансованість класів розпізнавання.

У статті розглядається інформаційний синтез динамічного планувальника незалежних задач, основаного на класифікаційному прогнозі функціонального стану вузлів гетерогенного обчислювального середовища. Навчання прогностичного класифікатора запропоновано здійснювати в рамках інформаційно-екстремальної інтелектуальної технології, яка дозволяє в процесі адаптивного кодування різнотипних ознак і оптимізації комбінованих контейнерів класів отримати вирішальні правила з низькою обчислювальною складністю і високою достовірністю [6, 7].

1 Формалізована постановка задачі

Нехай дано алфавіт $\{X_m^0 \mid m = \overline{1, M}\}$ класів розпізнавання, які описують функціональний стан обчислювального вузла в результаті виконання задачі обробки даних. Як реалізації кожного класу розпізнавання розглядаються впорядковані вектори різнотипних ознак, які характеризують особливості задачі та вузла обробки. Відома навчальна матриця типу “об’єкт-властивість”

$$\{y_{m,i}^{(j)} \mid m = \overline{1, M}; j = \overline{1, n_m}; i = \overline{1, N}\},$$

де M – потужність алфавіту класів розпізнавання;

n_m – кількість реалізацій класу X_m^0 ;

N – кількість ознак розпізнавання.

Відомий структурований вектор параметрів функціонування класифікатора:

$$g = \langle \delta_i, x_{k/m}, d_{k/m} \rangle, \quad (1)$$

де δ_i – параметр, що визначає для i -ї ознаки ширину кодуючого рецептивного поля;

$x_{k/m}$ – двійковий еталонний вектор для k -ї гіперсфери контейнера класу X_m^0 , що відновлюється в бінарному просторі ознак Ω_B ;

$d_{k/m}$ – кодовий радіус k -ї гіперсфери комбінованого контейнера класу X_m^0 .

При цьому задано такі обмеження:

$$d(x_m \oplus x_{k/m}) + d_{k/m} < d(x_m \oplus x_c),$$

де $d(x_m \oplus x_{k/m})$ – кодова відстань між двійковим еталонним (усередненим) вектором класу X_m^0 та двійковим еталонним вектором k -го контейнера $x_{k/m}$; $d(x_m \oplus x_c)$ – кодова відстань між двійковим еталонним (усередненим) вектором класу X_m^0 та двійковим еталонним (усередненим) вектором сусіднього класу X_c^0 ; $\delta_i \in [0; \delta_{\max}]$, де δ_{\max} – нормоване рецептивне поле, що визначає область значень параметра δ_i .

Необхідно в процесі навчання планувальника задач визначити оптимальні значення координат вектора параметрів функціонування (1), що забезпечують максимальне значення усередненого за алфавітом класів розпізнавання критерію функціональної ефективності

$$\bar{E} = \frac{1}{M} \sum_{m=1}^M \max_{\{k\}} \{E_m^{(k)}\}, \quad (2)$$

де $E_m^{(k)}$ – інформаційний критерій функціональної ефективності (КФЕ) на k -му кроці навчання планувальника розпізнавати реалізації класу X_m^0 ;

$\{k\}$ – упорядкована множина кроків навчання (відновлення контейнерів класів розпізнавання).

Необхідно в режимі екзамену планувальника задач прийняти рішення про належність реалізації, яка відповідає парі “задача-вузол”, до одного з класів алфавіту $\{X_m^0\}$, сформованого на етапі навчання, з метою призначення задачі на ресурс, що задовольняє конфігураційні обмеження, задані користувачем та провайдером ресурсів, та підвищує пропускну здатність і раціональність використання ресурсів обчислювального середовища.

2 Алгоритм функціонування планувальника задач

В гетерогенних розподілених обчислювальних середовищах часто існує невизначена кількість завдань і ресурсів в будь-який момент часу. З метою забезпечення високої пропускну здатності розподіленого обчислювального середовища алгоритм динамічного планування повинен мати невисоку обчислювальну складність і враховувати прогнозоване навантаження вузлів та ймовірність успішного виконання задач. На рис. 1 показано узагальнену схему планувальника задач, який при оптимізації інтегральних характеристик обчислювального середовища використовує прогностичні вирішальні правила, сформовані в процесі машинного навчання за даними передісторії обробки задач.

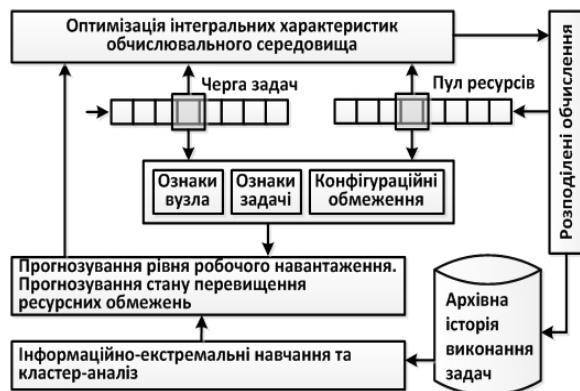


Рис. 1. Планувальник задач з інтелектуальною складовою

Запропонований планувальник задач приймає рішення про призначення задачі на вузол обробки, ґрунтуючись на ресурсних ознаках вузла і задачі та конфігураційних обмеженнях, заданих користувачем та провайдером ресурсу.

Конфігураційні обмеження використовуються алгоритмами оптимізації інтегральних характеристик обчислювального середовища для оцінки дотримання політик обслуговування. Крім цього

вони використовуються алгоритмами машинного навчання при формуванні навчальних вибірок для класу бажаних та класу небажаних рішень щодо призначення задачі на вузол обробки. Конфігураційні обмеження обов'язково містять задані провайдером ресурсні обмеження вузлів та задані чи обрані користувачем вимоги до умов обслуговування. Ресурсні обмеження характеризують ємність мережевих, процесорних і дискових ресурсів вузла та граничний обсяг частки ресурсів, яка може виділятися задачам користувача залежно від його ролі (адміністратор, привілейований користувач, авторизований користувач, гість). Користувач, у свою чергу, визначає бюджет, призначений для оплати тарифікованих ресурсів, граничний час обробки задач (дедлайн) та заданий користувачем пріоритет "важливості".

Ресурсні ознаки задачі формуються на основі унікального ідентифікатора задачі та передісторії її обробки на заданому вузлі. До ресурсних ознак задачі відносяться середнє вибіркоче значення та вибіркочна дисперсія обсягу використаних ресурсів процесора, оперативної пам'яті, пропускну здатності мережі та пропускну здатності для операцій вводу-виводу при роботі з жорстким диском. Додатковими ознаками задачі можуть бути метадані вхідних блоків даних, які корелюють з ресурсними вимогами. Ресурсні ознаки вузла поділяють на статичні, які не змінюються протягом функціонування обчислювального вузла, та динамічні, які змінюються в процесі його функціонування. До статичних характеристик вузла відносяться кількість процесорів, частота процесорів, архітектура процесорів, повний обсяг фізичної оперативної та віртуальної (файл підкачки) пам'яті, обсяг кеш пам'яті, обсяг пам'яті жорстких дисків, тип та версія операційної системи. До динамічних характеристик відносяться кількість використаних на вузлі процесорів, рівень завантаження процесорів, обсяг вільної оперативної пам'яті, інтенсивність операцій вводу/виводу, пропускну здатність мережі для прийому та передачі інформації, обсяг вільної віртуальної пам'яті, залишок дискового простору, споживана потужність та ін.

Класифікаційне прогнозування стану перевищення ресурсних обмежень перед призначенням задачі на обчислювальний вузол дозволяє підвищити продуктивність системи за рахунок скорочення кількості випадків неуспішного виконання задачі, викликаних нестачею ресурсів чи аномальною поведінкою вузла призначення. Класифікація дозволяє зменшити невизначеність, пов'язану з відсутністю або неточністю інформації про ресурсні вимоги і поведінку задачі на навантажених гетерогенних вузлах. Перевищення ресурсних обмежень призво-

дить до накладних витрат, пов'язаних з міграцією задач на інші ресурси з метою повторного їх виконання чи з запитом додаткових ресурсів, введення яких в експлуатацію може бути набагато повільнішим, ніж надходження нових запитів. Якщо планувальнику не вдається знайти для задачі вузол, який буде задовольняти ресурсним обмеженням, то задача буде очікувати появу необхідних ресурсів або буде повернута користувачу для корекції вимог до умов обслуговування. Прогностичні вирішальні правила можуть бути отримані в процесі машинного навчання, де роль учителя виконує прапорець (не)перевищення ресурсних обмежень.

Класифікаційне прогнозування зміни рівня навантаження гетерогенних вузлів при прибутті задач з різними ресурсними вимогами дозволяє зменшити простір пошуку кращого плану виконання задач з точки зору утилізації ресурсів та балансування навантаження. При цьому прогностичні вирішальні правила можуть бути отримані в процесі машинного навчання, де вектор ознак містить характеристики задачі та обчислювального вузла до прибуття задачі, а роль учителя виконує вектор характеристик вузла після прибуття задачі, який замінюють міткою класу рівня навантаження. Класи рівнів робочого навантаження для гетерогенних вузлів отримують шляхом кластер-аналізу векторів динамічних ознак вузла в архівній історії виконання задач. Оскільки кластер-аналіз призначений для ранжування функціональних станів вузла за рівнем навантаження, то динамічні ознаки вузла, що характеризують вільні ресурси, варто подавати у вигляді обсягу зайнятих ресурсів.

З метою покращення інтегральних характеристик обчислювального середовища найбільш пріоритетна задача призначається на вузол, де вона найімовірніше успішно виконається (прогнозується найменше робоче навантаження). При цьому пріоритет z -ї задачі враховує як час знаходження в черзі t_z , так і заданий користувачем пріоритет "важливості" p_z , і обчислюється за правилом:

$$P_z = \begin{cases} p_z t_z, & \text{якщо } 0 < t_z \leq T_z \text{ та } C_z > 0; \\ 0, & \text{якщо } t_z > T_z, \text{ або } C_z = 0, \end{cases}$$

де T_z – заданий користувачем для задачі граничний час виконання, після закінчення якого задача стає неактуальною і видаляється з черги;

C_z – залишок коштів, після успішного виконання попередніх задач на тарифікованих ресурсах.

Розглянемо основні кроки алгоритму динамічного розподілу обчислювальних задач, відсортованих в черзі за значенням пріоритету P_z .

1. Ініціалізація лічильника задач в черзі: $z := z + 1$.

2. Формування множини вузлів $\{H_{r,z} | r = \overline{1,R}\}$, на яких прогнозовано виконання z -ї задачі без перевищення ресурсних обмежень.

3. Якщо прийнятні вузли відсутні, то перехід до кроку 1, інакше до кроку 4.

4. Формування множини вузлів $\{H_{u,z} | u = \overline{1,U}\} \subset \{H_{r,z} | r = \overline{1,R}\}$, на яких прогнозовано менший рівень робочого навантаження.

5. Призначення z -ї задачі на будь-який вузол із множини $\{H_{u,z} | u = \overline{1,U}\}$ та видалити її з черги.

6. Якщо черга не пуста, то перейти до кроку 1, інакше до кроку 7.

7. ЗУПИН.

Таким чином, процес планування зводиться до обслуговування пріоритетованої черги задач шляхом обчислювально-простих класифікаційних прогнозів результату розподілу.

3 Алгоритм функціонування інформаційно-екстремального класифікатора

Спостереження за процесами розподілених обчислень включають в себе як кількісні, так і категоріальні ознаки. Тому запропоновано застосування адаптивного кодування ознак, в якому здійснюється уніфікація різнотипної інформації двійковим поданням, що враховує ймовірнісні характеристики як кількісних, так і категоріальних ознак. Процес кодування ознак в рамках інформаційно-екстремальної технології полягає у пошуку оптимальних меж рецептивних полів, які визначають обмеження на ймовірність появи значень ознаки в базовому класі розпізнавання $X_i^0 \in \{X_m^0\}$ (рис. 2). На рис. 2а показано, що верхня та нижня межі рецептивного поля для кількісної ознаки розпізнавання визначається відносно середнього вибіркового значення ознаки в базовому класі $X_B^0 \in \{X_m^0\}$.

На рис. 2б показано, що для кожної категоріальної ознаки верхня межа $P_{B,i}$ поля контрольного допуску рівна максимальній (модальній) відносній частоті появи значень ознаки в базовому класі $X_B^0 \in \{X_m^0\}$, а нижня межа $P_{H,i}$ залежить від параметра поля контрольних допусків. При цьому кодування кількісних ознак полягає у порівнянні їх числових значень $y_{m,i}^{(j)}$ з верхнім $A_{B,i}$ та нижнім $A_{H,i}$ порогоми, а кодування категоріальних ознак полягає у порівнянні відносної частоти $f_{m,i}^{(j)}$ появи значення

$y_{m,i}^{(j)}$ в i -й ознаці реалізацій класу з верхньою $P_{B,i}$ та нижньою $P_{H,i}$ межами частот. Тому формування бінарної навчальної матриці

$$\{x_{m,i}^{(j)} | i = \overline{1,N}; j = \overline{1,n}; m = \overline{1,M}\}$$

здійснюється за правилами:

$$x_{m,i}^{(j)} = \begin{cases} 1, & \text{if } A_{H,i} \leq y_{m,i}^{(j)} \leq A_{B,i}; \\ 0, & \text{if else;} \end{cases}$$

$$x_{m,i}^{(j)} = \begin{cases} 1, & \text{if } P_{H,i} \leq f_{m,i}^{(j)} \leq P_{B,i}; \\ 0, & \text{if else.} \end{cases}$$

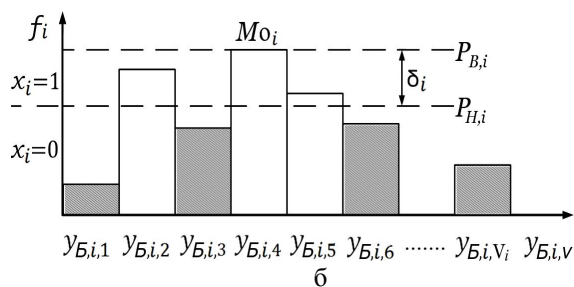
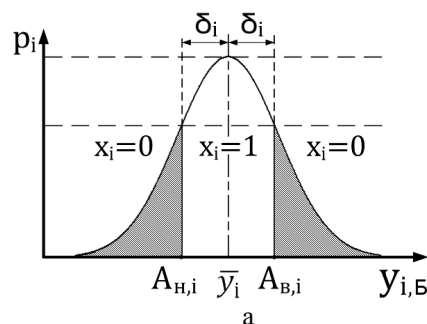


Рис. 2. Нижня і верхня межі рецептивного поля для i -ї ознаки базового класу:
а – кількісна ознака; б – категоріальна ознака

Оптимізація параметра $\delta = \delta_i, i = \overline{1,N}$ рецептивного поля, що реалізує механізм адаптивного двійкового кодування ознак, здійснюється за ітеративною процедурою [6]

$$\delta^* = \operatorname{argmax}_{G_\delta} \left\{ \frac{1}{M} \sum_{s=1}^M \left[\max_{G_E \cap G_d} E_m \right] \right\}, \quad (3)$$

де G_δ – область допустимих значень параметра рецептивного поля;

G_d – область допустимих значень радіуса гіперсферичного контейнера класу X_m^0 ;

G_E – допустима область визначення функції критерію (1), що обмежена нерівностями $D_{1,m} \geq 0,5$ і $D_{2,m} \geq 0,5$;

E_m – критерій ефективності навчання класифікатора розпізнавати реалізації m -го класу.

Вкладений в процедуру (6) алгоритм здійснює оптимізацію кодового радіусу d_m для простого гіперсферичного контейнера з еталонним (усередненим) бінарним вектором x_m для кожного класу розпізнавання.

У випадку апіорно некласифікованої навчальної матриці, яку необхідно розбити на класи еквівалентності, упорядковані за якісною шкалою, формування і оптимізацію вирішальних правил запропоновано здійснювати в процесі інформаційно-екстремального кластер-аналізу. При цьому оптимізація геометричних параметрів вирішальних правил здійснюється в бінарному просторі вторинних ознак, а межі кодуемого рецептивного поля для i -ї ознаки обчислюється за формулами:

$$A_{H,i} = y_{\min,i}, \quad A_{B,i} = y_{\min,i} + \delta_i,$$

де $y_{\min,i}$ – мінімальне значення обсягу використання i -го ресурсу в архівних даних моніторингу.

Розглянемо основні етапи реалізації алгоритму інформаційно-екстремальної кластеризації даних моніторингу на M кластерів.

1. Формування множини двійкових векторів $\{x^{(z)} \mid z = \overline{1, n}\}$, що є потенційними початковими центрами контейнерів, шляхом копіювання двійкової матриці $\{x_i^{(j)}\}$, видалення в ній однакових векторів та сортування решти векторів в порядку збільшення кількості одиничних координат.

2. Формування початкових центрів контейнерів $\{x_m \mid m = \overline{1, M}\}$ за правилом

$$x_m = \begin{cases} x^{(1)}, & m = 1; \\ x^{(n^{(m-1)}/[M-1])}, & 1 < m < M; \\ x^{(n)}, & m = M. \end{cases}$$

3. Кластер-аналіз двійкової матриці класу $\{x_i^{(j)}\}$ за алгоритмом k -медіан [8], з використанням дистанційної міри Хеммінга, та формування множини оптимальних центрів кластерів x_m^* .

4. Оптимізація радіуса контейнерів здійснюється за процедурою $d_m^* = \arg \max_{G_d} \{\bar{E}\}$ за умови, що кількість охоплених контейнерами векторів-реалізацій є більшою ніж 90% від загального обсягу.

5. ЗУПИН.

Основним недоліком існуючих алгоритмів машинного навчання в рамках ІЕІ-технології є неспроможність побудувати оптимальні в інформацій-

ному сенсі гіперсферичні контейнери для навчальних вибірок з витягнутим та багатомодальним розподілом векторів-реалізацій в умовах непередбачуваної інформативності ознак, що характерно для режимів донавчання та самонавчання [6, 7]. Усунути цей недолік запропоновано шляхом застосування комбінованих контейнерів, утворених об'єднанням декількох гіперсфер, які більш точно описують в бінарному просторі ознак область складної конфігурації відповідно до розподілу, властивого навчальним даним (рис. 3).

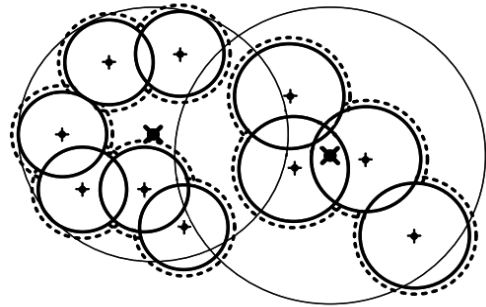


Рис. 3. Комбіновані контейнери двох класів розпізнавання в бінарному просторі ознак

Запуск процедури уточнення меж контейнерів шляхом комбінування гіперсфер здійснюється у випадку, коли не вдається отримати безпомилкові за навчальною матрицею вирішальні правила при оптимальних межах рецептивних полів. Центри і межі гіперсфер, які формують комбінований контейнер в бінарному просторі, знаходять в процесі інформаційно-екстремального кластер-аналізу при незмінних оптимальних параметрах рецептивних полів. Розглянемо основні кроки побудови комбінованих контейнерів для інформаційно-екстремального класифікатора.

1. Ініціалізація лічильника класів: $m := m + 1$.

2. Ініціалізація лічильника кількості кластерів для контейнера класу: $K_m := K_m + 1$.

3. Пошук оптимальних центрів $\{x_{m/k}^* \mid k = \overline{1, K_m}\}$ і радіусів $\{d_{m/k} \mid k = \overline{1, K_m}\}$ гіперсфер комбінованого контейнера класу X_m^0 в процесі інформаційно-екстремального кластер-аналізу його двійкової навчальної матриці $\{x_m^{(j)}\}$.

4. Якщо всі варіанти кількості кластерів перебрані, то знаходиться оптимальна кількість кластерів за процедурою $K_m^* = \arg \max_{\{K_m\}} \{E_m\}$, інакше перехід до кроку 2.

5. Якщо $m < M$, то перехід до кроку 1, інакше до кроку 6.

6. ЗУПИН.

Незбалансованість та неоднорідність навчальних даних знижує повну достовірність вирішальних правил. Тому в рамках інформаційно-екстремальної технології запропоновано початкове навчання на малих випадкових вибірках реалізацій класів (однакового розміру) і в поступовому покращенні отриманих вирішальних правил в процесі ковзного контролю на повному наборі даних. Неправильно розпізнані реалізації класів використовуються для послідовного доповнення навчальної вибірки (рис. 4).

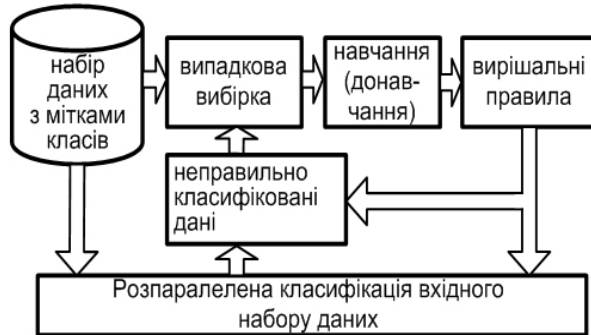


Рис. 4. Схема навчання при незбалансованому навчальному наборі даних

В даному підході пропуск напівдублюючих (майже співпадаючих) реалізацій здійснюється шляхом попереднього тестування вирішальних правил на кожній реалізації з наступним донавчанням у випадку неправильної класифікації. Це дозволяє збільшити різноманітність реалізацій образів не допускаючи "екстремальної" незбалансованості. При цьому висока узагальнююча здатність інформаційно-екстремальних вирішальних правил повинна забезпечити скорочення кількості ітерацій донавчання, а обчислювальна ефективність вирішальних правил забезпечує швидке сканування (при ковзному контролі) набору даних.

Як критерій ефективності навчання класифікатора розглядається модифікація інформаційної міри Кульбака [7], в якій відношення правдоподібності представлено у вигляді відношення повної ймовірності правильного прийняття рішень P_{true} до повної ймовірності помилкового прийняття рішень P_{false} . Для випадку двох альтернативних і рівно ймовірних гіпотез міра Кульбака має вигляд:

$$E_m = \left[P_{\text{true},m} - P_{\text{false},m} \right] \log_2 \frac{P_{\text{true},m}}{P_{\text{false},m}} = \begin{cases} P_{\text{true},m} = 0,5D_{1,m} + 0,5D_{2,m} \\ P_{\text{false},m} = 0,5\alpha_m + 0,5\beta_m \\ \alpha_m = 1 - D_{1,m}; D_{2,m} = 1 - \beta_m \end{cases}$$

$$= \left[D_{1,m} - \beta_m \right] \log_2 \left(\frac{1 + \left[D_{1,m} - \beta_m \right]}{1 - \left[D_{1,m} - \beta_m \right]} \right), \quad (4)$$

де $D_{1,m}$ – перша достовірність для класу X_m^0 ;

$D_{2,m}$ – друга достовірність;

α_m – помилка другого роду;

β_m – помилка другого роду;

n – кількість реалізацій в навчальній вибірці класу X_m^0 .

При розрахунку оцінок точнісних характеристик контейнера класу X_m^0 згідно з принципом k -найближчих "сусідів" сусіднім образом можна вважати набір з n реалізацій, найближчих до еталонного вектора x_m

$$\{x_c^{(j)} \mid j = \overline{1, n}\} \in \left[\bigcup_{c=1}^M X_c^0 \right] \setminus X_m^0.$$

Робоча (допустима) область визначення функції інформаційного критерію обмежена нерівностями $D_{1,m} \geq 0,5$, $D_{2,m} \geq 0,5$ та $n \geq n_{\text{min}}$, де n_{min} – мінімально допустимий обсяг векторів-реалізацій в класі.

Нормовану модифікацію критерію (4) представимо у вигляді

$$\hat{E}_m = \frac{E_m}{E_{\text{max}}}, \quad (5)$$

де E_{max} – значення критерію при $D_{1,m} = 1$ і $\beta_m = 0$.

У режимі екзамену рішення про належність вектора-реалізації $x^{(j)}$ одному з класів алфавіту

$\{X_m^0\}$ приймається шляхом обчислення геометричної функції належності

$$\mu_m = \max_{\{k\}} \{ \mu_{m/k} \},$$

де $\mu_{m/k}$ – функція належності вектора $x^{(j)}$ до k -го контейнера класу X_m^0 , обчислюється за правилом:

$$\mu_{m/k} = 1 - \frac{d(x_{m/k}^* \oplus x^{(j)})}{d_{m/k}^*},$$

де $d(x_{m/k}^* \oplus x^{(j)})$ – кодова відстань від центра k -го контейнера класу X_m^0 до вектора $x^{(j)}$.

Таким чином, алгоритм інформаційно-екстремального навчання полягає в ітераційній процедурі наближення глобального максимуму інфор-

маційного КФЕ (2) до його граничного значення шляхом оптимізації параметрів рецептивних полів та геометричних параметрів контейнерів. При цьому підвищення достовірності класифікатора розглядається як за рахунок уточнення меж контейнерів шляхом комбінування гіперсфер, так і шляхом підвищення репрезентативності навчальної вибірки в процесі ковзного контролю тестових даних.

4 Результати фізичного моделювання

Оцінка ефективності запропонованого підходу здійснювалась в експериментальному середовищі, що включає 6 попередньо налаштованих віртуальних машин `hadoop-vm-appliance-0.18`, які працюють під управлінням VMware ESX Server [9]. Одна віртуальна машина була налаштована як керуючий вузол, який виконує як функції служби метаданих (NameNode) для керування файловими операціями в розподіленій файловій системі HDFS, так і функції планувальника задач (jobTracker) [9, 10]. Решта віртуальних машин виступають в ролі підпорядкованих вузлів, виконуючи функції зберігання блоків даних (DataNode) та функції виконання задач обробки даних (TaskTracker) [10].

Кожна з віртуальних машин має ім'я `hadoopX`, де $X \in \{1, \dots, 6\}$ та призначену мережеву IP-адресу `10.100.20.X`, яку отримує через DHCP-протокол у відповідності до заданої MAC-адреси. З метою надання гетерогенності підпорядкованим вузлам для кожного з них задано різну апаратну конфігурацію, наприклад, різні значення обсягу пам'яті, пропускну здатності мережі та ін. Кожен вузол працює під управлінням операційної системи Ubuntu Linux 8.04.1. При цьому на кожному вузлі окремо відредатовано файл `/etc/sysctl` з метою встановлення різного обмеження на максимальну кількість одночасно відкритих файлів і максимальну кількість паралельних процесів. Відповідно до призначення вузлів з метою забезпечення координації між керуючим і підпорядкованими вузлами на кожному з вузлів відредатовано файли `/etc/hostname`, `/etc/hosts`, `conf/masters`, `conf/slaves` та `conf/hadoop-site.xml`, які відповідають за запуск відповідних програмних демонів за замовчуванням. З метою підвищення зручності моніторингу розподіленого обчислювального середовища було додатково встановлено і налаштовано систему Ganglia. Система Ganglia включає демон `Gmond`, що встановлюється окремо на кожному підпорядкованому вузлі для локального моніторингу, демон `Gmetad` для централізованого збору і зберігання даних моніторингу розподіленого обчислювального середовища та веб-сервер візуалізації Ganglia Web Frontend зі скриптами і деякими утилітами, зокрема `RRDtool` для роботи з кільцевою

базою даних [10]. Для інтеграції в систему Ganglia метрик Hadoop до директорії `hadoop/src/core/org/apache/hadoop/metrics/ganglia` було додано файли `GangliaContext.java` та `GangliaContext31.java`, скомпільовано Hadoop та відредатовано файл `/etc/hadoop/conf/hadoop-metrics.properties`. В табл. 1 показано розподіл запущених демонів на вузлах кластеру.

Таблиця 1
Активовані демони на вузлах
віртуального кластеру

Компонент	Віртуальні машини					
	1	2	3	4	5	6
MapReduce monitor	x					
MapReduce worker		x	x	x	x	x
HDFS monitor	x					
HDFS worker		x	x	x	x	x
Gmetad	x					
Gmond		x	x	x	x	x

Керуючому вузлу задано найбільший обсяг оперативної пам'яті – 800 Мбайт, а обсяг оперативної пам'яті підпорядкованих вузлів не перевищує 512 Мбайт, оскільки демони моніторингу (MapReduce monitor та HDFS monitor) потребують більше оперативної пам'яті, а демони обробки запитів (MapReduce worker та HDFS worker) потребують більше обчислювальних ресурсів.

Формування вхідного математичного опису запропоновано здійснювати на основі зібраної статистики виконання таких MapReduce задач, як WordCount, Terasort та Distributed Grep. Трасування задач проводилось за умов використання ємнісного планувальника, де вузол для виконання задач обирався за результатами перевірок неперевикнення порогу кількості запущених на вузлі задач та достатності обсягу вільної дискової пам'яті. При цьому обслуговування черги задач здійснювалося згідно політики FCFS (First Come, First Serve). Журнали, сформовані утилітою `RRDtool` системи Ganglia, складаються з векторів ресурсних ознак. Ці журнали конвертуються в формат, зручний для обробки інформаційно-екстремальним класифікатором. База містить 150000 записів про успішне виконання завдань та 18800 випадків перевантаження. З них було сформовано вибірки обсягом $n = 200$ векторів-реалізацій для успішного та неуспішного виконання задач. Словник ознак містить 12 ресурсних ознак задачі та 9 ресурсних ознак вузла, тобто $N = 21$. Для кластер-аналізу було використано вибірки по 1000 векторів на вузол.

Формування вирішальних правил для прогнозування перевантаження вузлів здійснювалося в процесі машинного навчання за процедурою (3). Графіки зміни значення усередненого нормованого КФЕ (5) в процесі оптимізації параметра рецептивних полів показано на рис. 5.

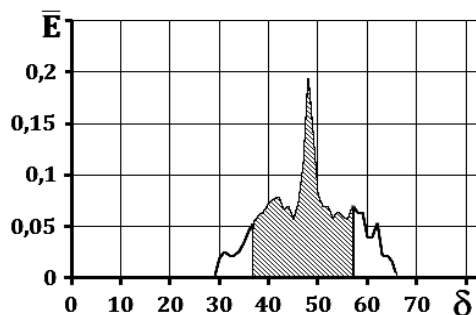


Рис. 5. Графіки зміни максимумів КФЕ в процесі оптимізації параметра рецептивних полів на ознаки розпізнавання

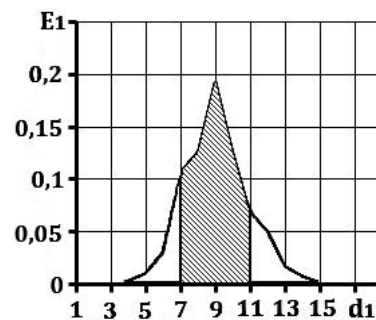
Аналіз рис. 5 показує, що оптимальний параметр поля контрольних допусків дорівнює $\delta^* = \pm 48\%$ при значенні максимуму усередненого нормованого критерію $\bar{E}^* = 0,191$. Таким чином, отримані в процесі паралельної оптимізації параметра $\delta_i = \delta, i = \overline{1, N}$ вирішальні правила не є безпомилковими за навчальною матрицею.

Оскільки метою інформаційно-екстремального навчання є відновлення в просторі ознак оптимальних контейнерів класів розпізнавання, то на рис. 6 наведено графіки залежності нормованого КФЕ (5) від радіусів гіперсферичних контейнерів класів X_1^0 та X_2^0 .

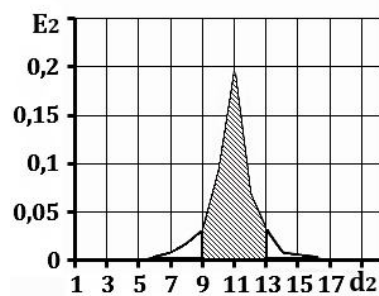
Аналіз рис. 6 показує, що оптимальні значення радіусів гіперсферичних контейнерів класів X_1^0 , X_2^0 відповідно рівні: $d_1 = 9$, $d_2 = 11$ (в кодових одиницях). При цьому міжцентрова відстань контейнерів рівня $d(x_1 \oplus x_2) = 14$. З метою підвищення достовірності вирішальних правил здійснюється уточнення меж контейнерів шляхом комбінування гіперсфер меншого радіусу, центри яких знаходять в процесі кластер-аналізу. Процес пошуку оптимальної кількості гіперсфер при побудові комбінованих контейнерів показано на рис. 7.

Аналіз рис. 7 показує, що оптимальні значення кількості гіперсфер для побудови комбінованих контейнерів класів X_1^0 та X_2^0 рівні відповідно $K_1^* = 5$ та $K_2^* = 5$. Процес оптимізації радіусів гіперсфер, що утворюють комбіновані контейнери,

при оптимальних значеннях кількості гіперсфер показано на рис. 8.

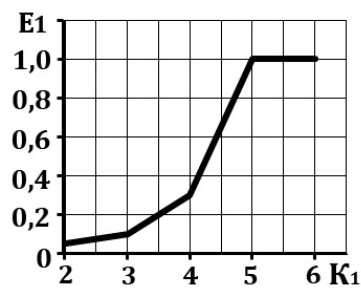


а

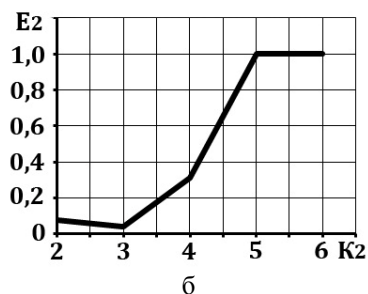


б

Рис. 6. Оптимізація радіусів гіперсферичних контейнерів: а – X_1^0 ; б – X_2^0



а



б

Рис. 7. Графіки залежності нормованого КФЕ від кількості гіперсфер, що формують контейнер класу: а – X_1^0 ; б – X_2^0

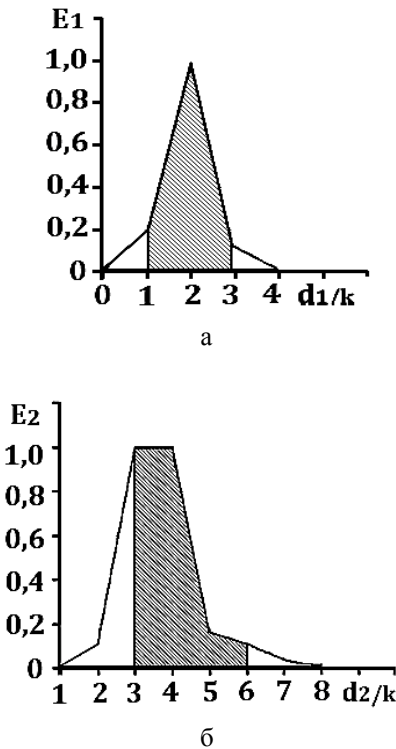


Рис. 8. Графіки залежності нормованого КФЕ від значення кодового радіусу гіперсфер контейнера класу: а – X_1^0 ; б – X_2^0

Аналіз рис. 8 показує, що оптимальні значення радіусів гіперсфер, що утворюють комбіновані контейнери класів X_1^0 та X_2^0 відповідно рівні $d_{1/k}^* = 2$, $d_{2/k}^* = 3$ (в кодових одиницях). При цьому вдалось побудувати безпомилкові за навчальною матрицею вирішальні правила.

З метою підвищення достовірності вирішальних правил здійснюється сканування всього набору даних та ітераційне доповнення навчальної матриці, що дозволяє реалізувати донавчання, не допускаючи “екстремальної” незбалансованості класів. На рис. 9 показано процес донавчання, де кожному різкому спаду значення критерію відповідає знаходження неправильно класифікованих векторів-реалізацій. При цьому кожному піку значення критерію відповідає результат донавчання класифікатора.

Аналіз рис. 9 показує, що в процесі ковзного контролю архівних даних моніторингу було виявлено 5 помилкових класифікацій. Корекція вирішальних правил була виконана в процесі 205 кроків зміни параметра рецептивного поля. В результаті отримано безпомилковий за навчальною і тестовою матрицями класифікатор для прогнозування перевантаження вузлів обробки.

Формування прогностичних вирішальних правил для оцінки рівня робочого навантаження вузла при прибутті потенційної задачі відбувається шля-

хом інформаційно-екстремальної кластеризації відповідної вибірки. На рис. 10 показано процес кластер-аналізу спостережень за ресурсними ознаками першого вузла при заданій кількості класів $K = 3$.

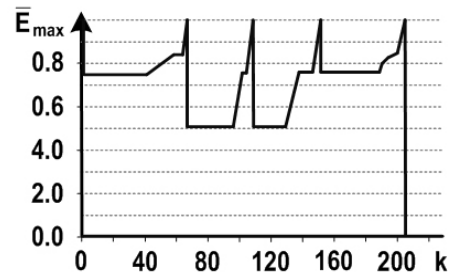


Рис. 9. Графіки залежності максимумів нормованого КФЕ від кількості кроків зміни рецептивного поля в процесі донавчання

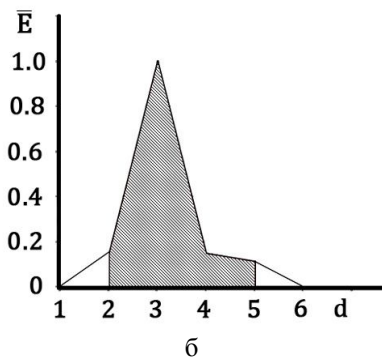
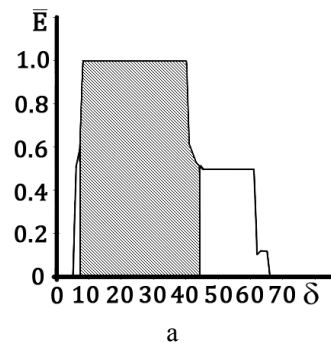


Рис. 10. Графіки зміни нормованого КФЕ при кластер-аналізі спостережень за ресурсними ознаками першого вузла:
а – оптимізація параметра рецептивного поля;
б – оптимізація радіусу контейнерів

Аналіз рис. 10 показує, що в процесі кластер-аналізу вдається побудувати чіткі прогностичні вирішальні правила для 4-х класів, що характеризують рівень навантаження першого вузла. Аналогічні результати при кластер-аналізі даних моніторингу інших вузлів.

На рис. 11 показано результати виконання задач в обчислювальному середовищі Nadoop кластера при використанні планувальника з вирішальними правилами за замовчуванням (П1), які оснований на

обмеженні кількості задач на вузлі і з прогностичними правилами (П2) та (П3), які оснований на інформаційно-екстремальних класифікаторах з гіперсферичними та комбінованими контейнерами відповідно.

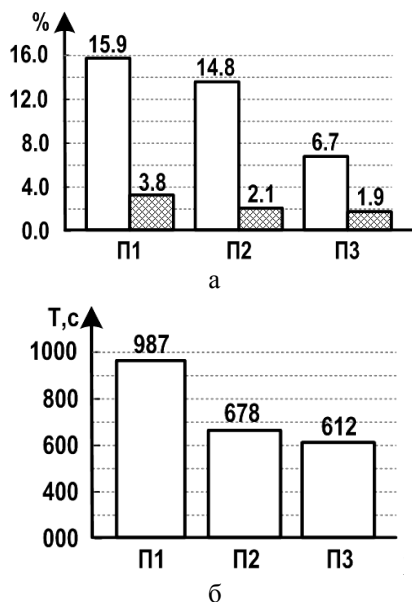


Рис. 11. Вплив вирішальних правил планувальника на пропускну здатність кластера:
а – задачі, які не виконались вчасно (заштриховано) чи призвели до перевантаження;
б – час завершення виконання всіх задач

Аналіз рис. 11 показує, що застосування прогностичних вирішальних правил дозволяє підвищити пропускну здатність обчислювального середовища за рахунок зменшення накладних витрат, пов'язаних з міграцією задач. При цьому алгоритм дозволяє раціональніше використовувати ресурси, що пов'язано з врахуванням реальної місткості вузла для гетерогенних задач. В результаті відпадає необхідність в спекулятивному виконанні задач, коли одна й та сама задача запускається одночасно на декількох вузлах з метою підвищення імовірності її вчасного і успішного виконання.

Таким чином, в рамках інформаційно-екстремальної інтелектуальної технології синтезовано безпомилкові за навчальною матрицею вирішальні правила для планувальника задач, що дозволяє при плануванні враховувати ресурсні вимоги і поведінку гетерогенних задач в гетерогенному обчислювальному середовищі.

Висновки

В рамках інформаційно-екстремальної інтелектуальної технології синтезовано безпомилкові за навчальною матрицею і обчислювально-ефективні

вирішальні правила для планувальника задач, що дозволило підвищити пропускну здатність обчислювального середовища за рахунок раціонального використання його ресурсів.

За результатами фізичного моделювання доведено перевагу використання комбінованих контейнерів класів функціонального стану обчислювальних вузлів у порівнянні з гіперсферичними, оскільки вони дозволяють точніше оцінити межу місткості вузлів для гетерогенних задач.

Література

1. Golubev, I. A. *Metadata-driven task scheduling in computer clusters [Text]* / I. A. Golubev, M. S. Kupryanov // *Proceedings of 9th International Conference on Computer Science and Information Technologies*. – Yerevan, Armenia, 2013. – P. 249-252.
2. *A study on using uncertain time series matching algorithms for MapReduce applications [Text]* / N. B. Rizvandi, J. Taheri, R. Moraveji, A. Y. Zomaya // *Journal of Concurrency and Computation: Practice and Experience. Special Issue in Cloud Computing Scalability*. – John Wiley Publisher, 2012. – 19 p.
3. Vladusic, D. *Using the black-box approach with machine learning methods in order to improve job scheduling in GRID environments [Text]* / D. Vladusic // *WSEAS Transactions on Systems and Control*. – 2009. – I. 6, v. 4. – P. 259-270.
4. Garcia, J. L. B. *Improved Self-management of DataCenter Systems Applying Machine Learning [Text]* / J. L. B. Garcia // *Ph. D. Thesis*. – Barcelona, Catalunya (Spain) : Polytechnic University of Catalonia. – 2013. – 155 p.
5. Helmy, T. *Independent Job Scheduling by Fuzzy C-Means Clustering and Ant Optimization Algorithm in a Computation GRID [Text]* / T. Helmy, Z. Rasheed // *International Journal of Computer Science*. – 2010. – V. 37, I. 3. – Massachusetts, USA : MIT-Press. – P. 531-536.
6. Довбиш, А. С. *Основи проектування інтелектуальних систем [Text]* / А. С. Довбиш. – Суми : СумДУ. – 2009. – 171 с.
7. Dovbysh, A. S. *Information-Extreme Algorithm for Optimizing Parameters of Hyperellipsoidal Containers of Recognition Classes [Text]* / A. S. Dovbysh, N. N. Budnyk, V. V. Moskalenko // *Journal of automation and information sciences*. – New York : Begell House Inc. – 2012. – V. 44, I. 10. – P. 35-44.
8. *Adapting K-Medians to Generate Normalized Cluster Centers [Text]* / J. B. Anderson, D. S. Gross, D. R. Musicant, A. M. Ritz, T.G. Smith, L.E. Steinberg // *Proceedings of the Sixth SIAM International Conference on Data Mining*. – Philadelphia : University City Science Center, 2006. – P. 165-175.
9. Cosulschi, M. *Running Hadoop applications in virtualization environment [Text]* / M. Cosulschi,

- M. Gabroeanu, A. Sbircea // Annals of the University of Craiova, Mathematics and Computer Science Series. – Craiova, Romania : University of Craiova, 2012. – Volume 39(2). – P. 322-333.*
- 10. Romer, T. Autoscaling Hadoop Clusters [Text] / T. Romer // Master's thesis. – Tartu, Estonia : University of Tartu, 2010. – 44 p.*

Надійшла до редакції 19.05.2015, розглянута на редколегії 18.06.2015

**ИНФОРМАЦИОННО-ЭКСТРЕМАЛЬНЫЙ КЛАССИФИКАТОР
С КОМБИНИРОВАННЫМИ КОНТЕЙНЕРАМИ ДЛЯ ПЛАНИРОВАНИЯ ЗАДАЧ
В РАСПРЕДЕЛЕННОЙ ГЕТЕРОГЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ**

V. V. Moskalenko, A. S. Rizhova

Рассматривается информационный синтез динамического планировщика независимых задач, основанный на классификационном прогнозе функционального состояния узлов гетерогенной вычислительной среды. В статье исследован алгоритм информационно-экстремального обучения прогностического классификатора с комбинированными контейнерами в условиях несбалансированности обучающих выборок и разнотипности признаков распознавания, что имеет место в практических задачах мониторинга вычислительной среды. Физическое моделирование работы предложенных алгоритмов реализовано в вычислительной среде кластера с моделью распределенных вычислений MapReduce на примере задач WordCount, Terasort и Distributed Grep.

Ключевые слова: распределенные вычисления, планирование задач, очередь задач, гетерогенные ресурсы, машинное обучение, распознавание образов, признаки распознавания, контейнер класса распознавания, информационный критерий.

**INFORMATION-EXTREME CLASSIFIER WITH COMBINED CONTAINER FOR INDEPENDENT
JOB SCHEDULING IN DISTRIBUTED HETEROGENEOUS COMPUTING ENVIRONMENTS**

V. V. Moskalenko, A. S. Rizhova

Informational synthesis of Dynamic Task Scheduler based on the classification-driven prediction of node's functional state in heterogeneous computing environment is considered. Information-extreme algorithm of machine learning for prognostic classifier with combined containers under conditions of imbalanced dataset and feature space heterogeneity that take place in the practical tasks of monitoring computing environment is proposed. Physical modeling is implemented into cluster computing environment with MapReduce computing model on the example such computational tasks as WordCount, Terasort and Distributed Grep.

Keywords: distributed computing, scheduling, queue, heterogeneous resource, machine learning, pattern recognition, feature vector, recognition container of class, information criterion.

Москаленко В'ячеслав Васильович – канд. техн. наук, ст. викл. кафедри комп'ютерних наук, Сумський державний університет, Суми, Україна, e-mail: ai.sys.dev@gmail.com.

Рижова Алена Сергіївна – аспірант кафедри комп'ютерних наук, Сумський державний університет, Суми, Україна, e-mail: alenarizhova@gmail.com.