

# РОЗПОДІЛ РЕСУРСІВ МІЖ ПУЛАМИ ТА РОЗПОДІЛ ЧАСУ МІЖ ЗАВДАННЯМИ ВСЕРЕДИНИ ПУЛУ ІЗ ЗАСТОСУВАННЯМ МАШИННОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ

Рева К.В., Родіонов А.М.

*Національний технічний університет України «Київський політехнічний інститут»*

## Анотація

У даній роботі представлений огляд проблеми розподілу ресурсів між декількома пулами, моделюється планувальник ресурсів та робота алгоритмів розподілу часу на завдання всередині пулу

*Ключові слова:* машинне навчання з підкріпленням, розподіл ресурсів між завданнями, TD-learning, Cloud computing

## Вступ

Динамічний розподіл ресурсів між багатьма клієнтами є актуальною задачею [1], [2], особливо у зв'язку зі стрімким розвитком "хмарних обчислень" (Cloud computing), які пропонують користувачам сервіси типу PaaS (Platform-as-a-Service), SaaS (Software-as-a-Service), IaaS (Infrastructure-as-a-Service) [3]. Кожен з цих сервісів, зазвичай, представляє собою пул певних ресурсів, підтримку та прогнозування динаміки використання яких має забезпечувати провайдер "хмарних обчислень".

Для вирішення проблеми динамічного розподілу ресурсів між пулами, в даній роботі розглядається модель на основі машинного навчання. Вона є базою для експериментів з алгоритмами ефективного розподілу часу на завдання всередині кожного пулу.

Моделюється робота алгоритмів розподілу часу та пропонується підхід машинного навчання з підкріпленням як один методів вирішення даної проблеми.

## 1. Постановка задачі

Опишемо базову модель для розподілу ресурсів між пулами. Загальна ідея в тому, що кожен пул вираховує значення ефективності своєї роботи та найбажанішу для нього дію (з урахуванням, що трансфер ресурсів не безкоштовний і бездіяльність може бути вигідніша для конкретного пулу в деякий момент часу) і відправляє ці значення деякому арбітру. Арбітр на основі значень ефективності обирає пул, дія якого принесе найбільшу користь системі, та виконує відповідну дію, що пропонує пул (забираючи, якщо потрібно, ресурси з менш ефективних пулів).

Даний підхід не бере до уваги розподілення часу роботи між конкретними завданнями в конкретному пулі, що при наявності завдань з параметрами (час на виконання, необхідні ресурси), що суттєво

різняються, може привести до їх нерівномірного опрацювання. Має сенс розглянути декілька алгоритмів для розподілу часу на завдання всередині пулу:

- 1) Обробка завдань, базуючись на черзі в залежності від часу прибуття
- 2) Обробка завдань, базуючись на пріоритетній черзі [4] з витісненням завдань, час опрацювання яких перевищив деяке порогове значення
- 3) Обробка завдань, базуючись на підході машинного навчання з підкріпленням

## 2. Загальний опис підходу для базової моделі

Для кращого розрахування ефективності роботи кожного пулу, має сенс використати метод машинного навчання з підкріпленням, хоча існує безліч інших методів та підходів до вирішення даної проблеми [5],[6]. Машинне навчання з підкріпленням - підрозділ машинного навчання, що вивчає як агент повинен діяти в деякому оточенні, щоб максимізувати довготривалу винагороду [7]. В даній задачі, агент - пул, що намагається мінімізувати чергу завдань та максимально ефективно використати наявні ресурси для завдань, що потрібно опрацювати. Оточення, в якому працює агент, зазвичай описується як марковський процес прийняття рішень  $(S, A, r, T)$ , де

- $S$  - фінітна множина станів
- $A$  - фінітна множина дій
- $r : S \times A \times S \mapsto \mathbb{R}$  - функція винагороди
- $T : S \times A \mapsto PD(S)$  - функція переходу станів, що відображає поточний стан та дію агента у набір ймовірнісних розподілів на  $S$

На кожному кроці  $t$ , агент спостерігає стан  $s_t \in S$ , обирає дію  $a \in A$ , змінює свій стан згідно з функцією  $T$ , що залежить тільки від  $s_t$  і  $a_t$ , та отримує винагороду  $r(s_t, a_t, s_{t+1})$ . Ґрунтуючись на та-

кій взаємодії з навколишнім середовищем, агент, який навчається з підкріпленням, повинен виробити стратегію  $\pi : S \mapsto A$ , яка максимізує величину  $R = r_0 + r_1 + \dots + r_n$ . Один із підходів - використання TD(0) (temporal difference(TD) learning). При такому підході, функція довготривалої цінності має вигляд[8]:

$$V^\pi(s_t) \leftarrow V^\pi(s_t) + \alpha_t(r(s_t, \pi(s_t), s_{t+1}) + \gamma V^\pi(s_{t+1}) - V^\pi(s_t))$$

При умові, що агент буде слідувати стратегії  $\pi$ :

$$\pi'(s) = \operatorname{argmax} E[r(s_t, a, s_{t+1}) + \gamma V^\pi(s_{t+1})]$$

Але, беручи до уваги, що простір станів занадто великий для даної задачі, потрібно ввести ряд спрощень, що зменшить складність обчислення функції цінності.

### 3. Основні положення базової моделі

Ідея в тому, щоб наблизити функцію цінності. Базуючись на [8], намагаємось наблизити істину функцію цінності за допомогою:

$$\hat{V}(s, p) = \sum_{i=1}^M p^i \varphi^i(s)$$

де:  $p_{t+1}^i = p_t^i + \alpha_t[r(s_t, \pi(s_t), s_{t+1}) + \gamma \hat{V}(s_{t+1}, p_t) - \hat{V}(s_t, p_t)] \varphi^i(s_t)$

є параметрами навчання, що налаштовуються з часом, а

$$\varphi^i(s) = \frac{w^i(x)}{\sum_{i=1}^M w^i(x)}$$

являє собою базис для розкладу параметрів.  $M$  - кількість категорій, на які можна класифікувати кількість ресурсів, що використовуються даним пулом,  $w^i$  - деякий базисний набір функцій. Також, якщо вважати перехід між станами миттєвим, то можна задати стратегію вибору найліпшої дії як [8]:

$$\pi'(s) = \operatorname{argmax} [r(s_t, a, \bar{s}_t) + V^\pi(\bar{s}_t)]$$

Функція миттєвої винагороди повинна залежати від значення передбачуваної довжини черги для даного пулу ( $B$ ), а також від ціни трансферу ресурсів між пулами ( $TC$ ):

$$r(s_t, s_{t+1}) = B(s_t, s_{t+1}) + TC(s_t, s_{t+1})$$

### 4. Попереднє моделювання

Використовуючи вищевказану модель, було отримано ряд результатів для різного набору параметрів. Далі розглянемо результати моделювання одного з пулів з початковим розподілом одного з ресурсів (6,5,7) відповідно для трьох пулів та 10000 ітерацій. Час на опрацювання кожного завдання коливається від 5 до 10 ітерацій, а частота прибуття нових завдань не перевищує 0.3 Зміна довжини черги для 2-го

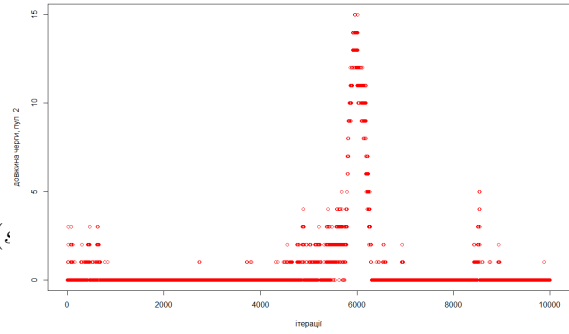


Рис. 1. Залежність довжини черги в другому пулі від номеру ітерації

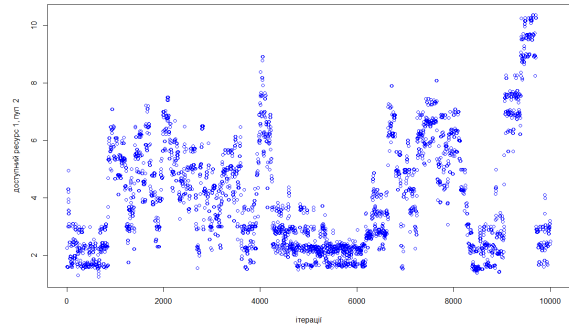


Рис. 2. Залежність кількості доступного ресурсу для другого пулу від номеру ітерації

пулу показана на рис.1, а відповідна їй кількість доступних ресурсів для даного пулу показана на рис.2. Як можна побачити, різке збільшення довжини черги для даного пулу, зумовлене відбиранням ресурсів з цього пулу, було подолано досить швидко відносно загального часу роботи. Система при виявленні нестабільності змогла відновити нормальний стан для даного пулу.

### 5. Обробка завдань всередині пулу

Описана вище модель дозволяє досить ефективно розподіляти ресурси між пулами, але при даному підході не береться до уваги ефективність розподілу часу на обробку завдань всередині самого пулу.

Проблема виникає, коли час виконання значної кількості завдань перевищує середній час виконання інших завдань хоча б в декілька разів. Значною в даному випадку може вважатися така кількість завдань, що забирає з пулу кількість ресурсів, достатніх для зміни поведінки алгоритму базової моделі. В такому випадку нарощується черга завдань зі значним часом простою відносно ситуації з нормальним розподілом часу виконання.

Одним із варіантів вирішення даної проблеми може бути витіснення завдань, що виконуються в даний момент, назад в чергу, якщо їх час виконання перевищить деяке порогове значення.

Чергу ж має сенс сортувати по деякому критерію, що може включати не тільки час перебування в чер-

Тип підходу	Середній час знаходження в черзі
Проста черга	46.50
Черга з витісненням	15.12

Табл. 1. Таблиця залежності часу знаходження завдань в черзі від типу підходу до розподілу часу для 200 ітерацій

Тип підходу	Середній час знаходження в черзі
Проста черга	240.07
Черга з витісненням	122.91

Табл. 2. Таблиця залежності часу знаходження завдань в черзі від типу підходу до розподілу часу для 1000 ітерацій

зі, а й кількість необхідних ресурсів для виконання чи необхідний час для виконання.

Для спрощення, використаємо тільки критерій часу перебування в черзі. В такому випадку, витіснені завдання будуть потрапляти в кінець черги, а на їх місце будуть взяті завдання в порядку їх потрапляння в чергу. Даний випадок відрізняється від випадку простої черги тільки наявністю витіснення перевиконаних завдань.

Далі змодельовано поведінку одного з пулів для 200 і 1000 ітерацій та обмеження на опрацювання завдання в 10 ітерацій. Час на опрацювання кожного завдання збільшено, тепер він коливається від 5 до 80 ітерацій, а частота прибуття нових завдань не перевищує 0.8

Як видно з табл. 1 та табл. 2, середній час знаходження в черзі для підходу з витісненням значно менший, ніж аналогічна величина в підході з чергою без витіснення.

Отже, у випадку витіснення перевиконаних завдань, значно зменшується середнє значення часу знаходження завдань в черзі, тобто пул виділяє час на обробку завдань більш рівномірно, виключаючи ситуацію, коли множина завдань з довгим часом виконання змушує чекати велику кількість інших завдань. Але потрібно більше експериментів, щоб визначити оптимальні початкові параметри.

Також потрібно зазначити, що при даному підході, кількість опрацьованих завдань не падає відносно випадку без витіснення (табл. 2)

Надалі має сенс ввести агента на основі машинного навчання з підкріпленням, що міг би розподіляти

Тип підходу	Середня кількість виконаних завдань на 1000 ітерацій
Проста черга	942.8
Черга з витісненням	925.1

Табл. 3. Таблиця залежності кількості виконаних завдань від типу підходу до розподілу часу

час на завдання в пулі більш оптимально в контексті як середнього часу простою завдань в черзі, так і кількості ресурсів на обробку, зберігши при цьому підхід витіснення в чергу тих завдань, які перевищують порогове значення часу виконання. Такий підхід повинен збільшити гнучкість системи у випадку появи аномальних завдань як по часу виконання, так і по вимогам до ресурсів.

## 6. Висновки

Було проведено моделювання планувальника ресурсів, в основі якого лежить машинне навчання з підкріпленням. Крім того, була змодельована робота планувальника ресурсів з різними підходами щодо розподілення часу на завдання всередині самого пулу. Було продемонстровано ефективність підходу з витісненням перевиконаних завдань та запропоновано введення внутрішнього агента для кожного пулу, що буде намагатися оптимально розподілити час на обробку завдань.

## Література

1. Oracle Cloud's Platform as a Service: Innovate, Integrate, and Collaborate — <http://www.oracle.com/us/solutions/cloud/platform-as-a-service/oracle-cloud-paas-brief-2411348.pdf> —
2. Шмойлов Дмитрий Викторович — ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ: АКТУАЛЬНОСТЬ И ПРОБЛЕМЫ — Электронное научное издание "Электроника и информационные технологии" — ГОУВПО "Мордовский государственный университет им. Н.П. Огарева" — 2011
3. Why "PaaS" is seen and considered as the future of cloud computing? — <http://marketrealist.com/2014/07/why-paas-is-considered-the-future-of-cloud-computing/> —
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms — Third Edition, The MIT Press Cambridge, Massachusetts London, England, 2009 — ISBN 978-0-262-53305-8 — Priority queues, pp. 162–169.
5. А.В. Прохоров, Е.М. Пахнина — Мультиагентные технологии управления ресурсами в распределенных вычислительных средах — Second International Conference "Cluster Computing" CC 2013 — с. 184-190
6. С.В. Минухин, С.В. Баранник — Метод планирования ресурсов в распределенной гетерогенной системе и исследование его практической реализации — Міжнародна конференція "Високопродуктивні обчислення" НРС-UA'2012 — с. 247-254
7. Обучение с подкреплением / Р.С. Саттон, Э. Г. Барто; пер. с англ — М. : БИНОМ. Лаборатория знаний, 2011. — 399 с.

8. D. Vengerov. A Reinforcement Learning Approach to Dynamic Resource Allocation. Eng. Appl. of AI, 20 (3) — 2007. — pp.383-390