УДК 004.85:519.872.2

І. Ю. Мирошникова, О. М. Новіков
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Проспект Перемоги, 37, 03056 Київ, Україна

Вибір провайдера хмарного сервісу на основі навчання з підкріпленням і репутації провайдерів

Для задачі вибору провайдера хмарного сервісу запропоновано використати підхід на основі навчання з підкріпленням, що додатково враховує репутацію провайдерів, яка визначається за досвідом використання їхніх сервісів та оцінками користувачів. Для провайдерів зі сталими параметрами надання сервісів було показано зменшення кількості кроків, що необхідні для навчання системи, порівняно з підходами на основі навчання з підкріпленням без урахування репутації провайдерів.

Ключові слова: *хмарні обчислення, задача вибору провайдера, навчання з підкріпленням, репутація, брокер, модель аукціону.*

В умовах розвитку інформаційних технологій збільшується навантаження на обчислювальні ресурси інформаційних систем, що спричиняє зростання вимог до характеристик ресурсів і витрат на їхнє обслуговування. Використання хмарних обчислень надає можливість зменшити подібні витрати, оскільки дозволяє утворити з віддалених ресурсів сторонніх провайдерів спільне об'єднання (пул) і оперативно отримувати до нього доступ [1]. Перед інформаційною системою постає задача вибору провайдера серед доступних, ресурси якого буде використано для обробки поточного завдання.

З метою автоматизації прийняття рішення про вибір провайдера, частиною архітектури інформаційної системи стає брокер, який виконує роль посередника між користувачами системи та провайдерами сервісів [2, 3]. Брокер містить підсистему, що приймає рішення відповідно до певної стратегії. При побудові стратегії необхідно враховувати наступні властивості середовища системи: неможливо отримати інформацію про стан всієї системи в цілому (часткова спостережуваність); характеристики сервісів, що надаються провайдерами, можуть бути невідомі на момент прийняття рішень (невизначеність); множина доступних провайдерів може часто змінюватися за період життя системи (динамічність).

Урахувати ці властивості дозволяють стратегії на основі машинного навчання, оскільки за такого підходу налаштування параметрів, що впливають на прийняття

рішень, відбувається протягом всього періоду функціонування системи. Підрозділом машинного навчання є навчання з підкріпленням, застосування якого не потребує наявності попередньо заданої навчальної вибірки, а вхідні зразки агент (особа, що приймає рішення) отримує в результаті взаємодії із середовищем [4].

Задачу розподілу обчислювальних одиниць ресурсу між завданнями у черзі системи було сформульовано в термінах задачі навчання з підкріпленням у роботах [5–7]. При цьому прийняття рішень відбувалося за стратегіями на основі Q-навчання [8], а при оцінюванні ресурсу враховувалися наступні параметри: очікування завдання у черзі на виконання і час виконання цього завдання ресурсом [5]; різниця між зазначеною допустимою затримкою виконання завдання та її реальним значенням, різниця між обсягом обчислювальних одиниць ресурсу: використаних для виконання поточного завдання та наданих за нормою віртуальної організації [6]; чи була завершена і повністю виконана обробка завдання [7].

За умови частих змін у середовищі системи, час адаптації підсистеми прийняття рішень до них може бути значно більшим за час між двома послідовними взаємовиключними змінами. Виникає потреба у зменшенні часу адаптації підсистеми, що можливо за рахунок налаштування керуючих параметрів стратегій [5]. Але такий підхід не є універсальним і залежить від особливостей середовища системи. Також зменшення часу навчання можливо за рахунок уведення репутації провайдерів, як додаткової оцінки провайдера на основі відгуків користувачів і характеристик сервісів [9]. Тому актуальною є розробка та дослідження методів, які поєднують підхід навчання з підкріпленням та поняття репутації, що дозволить зменшити час навчання системи.

Таким чином, мета даної роботи полягає в тому, щоб запропонувати модель підсистеми вибору провайдера сервісу на основі навчання з підкріпленням, яка відрізняється врахуванням репутації провайдерів при прийнятті рішень, що призводить до зменшення часу навчання системи.

1. Математична модель системи

Розглянемо систему, що містить одного брокера. Визначимо його стан як кортеж множин сутностей

$$S = < U, P, T >, \tag{1}$$

де U — множина користувачів, яких обслуговує даний брокер; P — множина провайдерів сервісів, які доступні даному брокеру і здатні обробити надіслане користувачем завдання; T — черга надісланих користувачами завдань, що чекають обробки.

Взаємодія брокера та провайдерів у даній системі має в основі модель аукціону з параметрами μ та ν, де μ — фактор співпраці ($0 < \mu < 1$), ν — фактор відмови від співпраці ($-1 < \nu < 0$) [9]. Також для кожного з провайдерів брокер визначає значення функції репутації $r : P \to Rep$, де $Rep \subset \Re$ — множина допустимих значень репутації провайдера, залежить від обраної метрики. Частина архітектури системи, що відповідальна за зберігання поточних даних про репутацію провайдерів, виділена у окрему підсистему (рис. 1).

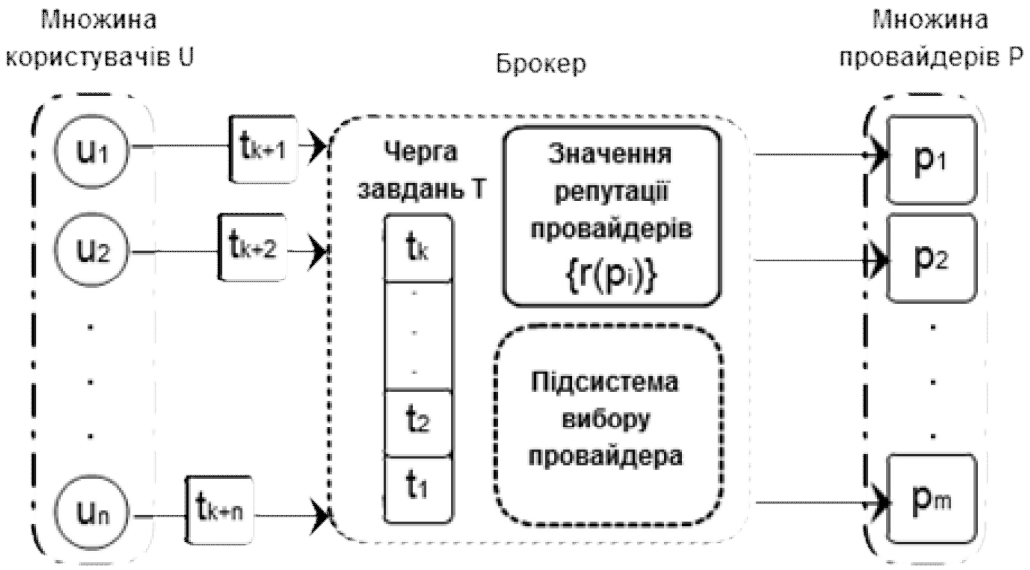

Рис. 1. Архітектура системи з одним брокером

Кожне завдання з черги $t \in T$ брокер надсилає до підсистеми вибору провайдера і отримує провайдера $\hat{p}$, який надасть сервіс для обробки завдання *t*. При цьому підсистема приймає рішення на основі оптимізації наступного критерію:

$$v^u(c, \hat{p}) \to \max, \tag{2}$$

де v^u — дійсна цінність наданого сервісу, яка обчислюється на стороні користувача *u* за деякою функцією $v^u : C \times P \to \Re$; *c* — кортеж параметрів, що характеризують наданий сервіс. Надсилається користувачеві разом з результатом виконання завдання. Перелік параметрів, що входять до *c*, є фіксованим для всіх користувачів брокера, проте залежить від виду функції v^u. Одними з таких параметрів можуть бути час, протягом якого виконувалося завдання, та індикатор невиконання завдання сервісом (наприклад, унаслідок збою).

Для оптимізації наведеного критерію у наступному розділі буде запропоновано стратегію вибору провайдера на основі навчання з підкріпленням, що враховує репутацію провайдерів.

2. Стратегія вибору провайдера

Для того, щоб застосувати навчання з підкріпленням при виборі провайдера, розглянемо підсистему вибору провайдера як агента, який намагається максимізувати загальну отриману винагороду. Множина доступних агенту дій полягає у виборі одного з провайдерів. Згідно з наведеним у (2) критерієм оптимізації, вважаємо винагородою за дію цінність наданого сервісу.

2.1. Прогнозування майбутньої винагороди на основі TD-навчання

Для прогнозування цінності сервісу вводиться функція очікуваної цінності

$$f : U \times T \times P \to \Re , \tag{3}$$

де U, P, T — множини сутностей з (1). Тобто, дійсне число $f(u_c, t_c, p_c)$ — очікувана оцінка якості виконання завдання типу t_c, надісланого користувачем u_c, сервісом, наданим провайдером p_c.

Оновлення значень функції очікуваної цінності відбувається за правилом TD-навчання [10]:

$$\Delta = v^u(c, \hat{p}) - f(u, t, \hat{p}),$$

$$f(u, t, \hat{p}) \leftarrow f(u, t, \hat{p}) + \alpha\Delta ,$$

де α — швидкість навчання ($0 \le \alpha \le 1$). Початкове значення встановлюється рівним $\alpha_0 = 1$, надалі α покроково зменшується до заданого мінімального значення $\alpha_{\min}$.

2.2. Вибір між використанням і дослідженням

Прогноз цінності наданого сервісу відбувається на основі попереднього досвіду використання сервісів, наданим цим провайдером. Оскільки навчання триває все життя системи, під час прийняття рішень агент має обирати між дослідженням середовища та використанням набутого досвіду. Такий вибір відбувається за ε-жадібною стратегією [11]: з імовірністю ε брокер обирає провайдера $\hat{p}$ випадковим чином з P, з імовірністю $(1-\varepsilon)$ вибір відбувається за описаним правилом. Початкове значення параметра ймовірності ε встановлюється рівним $\varepsilon_0 = 1$, надалі з часом ε зменшується до заданого мінімального значення $\varepsilon_{\min}$.

2.3. Вибір провайдера

Для виконання завдання типу t_c брокер обирає з множини авторитетних провайдерів P_r провайдера $\hat{p}$, який надає сервіс з максимальною очікуваною цінністю

$$\hat{p} = \arg\max_{p \in P_r} f(u_c, t_c, p), \tag{4}$$

де arg — оператор, такий що $\arg f(u, t, p) = p$. Якщо множина авторитетних провайдерів порожня, брокер має здійснювати вибір на множині всіх доступних провайдерів P. На рис. 2 наведено псевдокод алгоритму вибору провайдера.

chooseProvide (t_c, u, P_r, P)
// t_c: тип надісланого завдання
// u: користувач, що надіслав завдання t
// P_r: множина авторитетних провайдерів
// P: множина всіх доступних провайдерів
// визначити множину пошуку провайдерів
if ($P_r = \varnothing$) $P_{search} = P$
else $P_{search} = P_r$
$\hat{p} = \arg\max_{p \in P_{search}} f(u, t, p)$
return $\hat{p}$

Рис. 2. Вибір провайдера для опрацювання завдання

2.4. Репутація провайдерів

Нехай $\theta \in \Re$ — задана константа, що визначає поріг репутації. На основі θ брокер виділяє множину P_r авторитетних провайдерів:

$$P_r = \{p \in P \mid r(p) \geq \theta\} \subseteq P.$$

Відповідно до моделі оновлення репутації у роботі [12], також можна було би виділити множину неавторитетних провайдерів, оцінка репутації яких стала нижчою за певне фіксоване значення. Але провайдери з такої множини більше не розглядаються при прийнятті рішень. У задачі вибору провайдера виділення такої множини недоцільне, оскільки зміна на краще характеристик сервісів провайдера з цієї множини не буде відмічена брокером.

Провайдеру p_{new}, який щойно став доступним для брокера, тобто ще не отримував завдання на обробку, встановлюється деяке задане початкове значення репутації $r(p_{new}) = r_0 \in Rep$.

Додатково до оновлення функції очікуваної цінності, оцінка репутації $r(\hat{p})$ провайдера $\hat{p}$ також потребує оновлення. Нехай $V^u \in \Re$ — оцінка (рівень) якості, яку користувач вимагає від сервісу. Введемо також параметр *k*, що дозволить узагальнити правило оновлення оцінки репутації провайдера [9]:

$$k = \begin{cases} \mu, \text{якщо } \delta \geq 0, \\ \nu, \text{якщо } \delta < 0, \end{cases} \text{ де } \delta = v^u(c, p) - V^u.$$

Тоді правило оновлення оцінки репутації провайдера $\hat{p}$ формулюється наступним чином:

$$r(\hat{p}) \leftarrow \begin{cases} r(\hat{p}) + k(1 - r(\hat{p})), \text{якщо } r(\hat{p}) \geq 0, \\ r(\hat{p}) + k(1 + r(\hat{p})), \text{якщо } r(\hat{p}) < 0. \end{cases}$$

Після оновлення оцінки репутації її значення порівнюється з порогом θ. Якщо $r(\hat{p}) \geq \theta$ і $\hat{p} \notin P_r$, провайдер $\hat{p}$ додається до множини авторитетних. Відповідно, якщо $r(\hat{p}) < \theta$ і $\hat{p} \in P_r$, провайдер має бути вилучений з множини авторитетних провайдерів P_r.

Загальна послідовність дій брокера при обробці завдань з черги наведена на рис. 3, де chooseProvider відповідає передачі вказаних параметрів до підсистеми вибору провайдера.

globalSystemLifetime (T, u, P, θ, r_0, $\varepsilon_{\min}$, ε_{step})
// T: черга завдань брокера
// u: користувач, що надіслав завдання t
// P: множина всіх доступних провайдерів
// θ: поріг репутації
// r_0: початкове значення репутації для нового провайдера
// $\varepsilon_{\min}$, ε_{step}: відповідно мінімальне та крокове значення параметра імовірності для ε-жадібної стратегії

$P_r \leftarrow \varnothing$
$\varepsilon \leftarrow 1$
for each $p_i \in P$:
    $r(p_i) \leftarrow r_0$
    $f(u, t_c, p_i) \leftarrow 0$

for each $t \in T$
    $t_c \leftarrow getType(t)$ // визначити тип завдання

    // обрати між дослідженням середовища і використанням досвіду
    ε_1 = getRandomInteger(0,1)
    if ($\varepsilon_1 < \varepsilon$) // обрати випадковим чином провайдера з P
        $\hat{p}$ = getRandomProvider(P)
    else // обрати провайдера з множини авторитетних
        $\hat{p}$ = **chooseProvider** (t_c, u, P_r, P)

    // надіслати провайдеру $\hat{p}$ завдання t
    // визначити кортеж c характеристик наданого сервісу
    // отримати від $\hat{p}$ і надіслати користувачеві u результат обробки завдання t разом з c
    // отримати від користувача u оцінку наданого сервісу $v^u(c, \hat{p})$ і різницю між бажаною та отриманою цінністю сервісу δ

    // оновити ε

if $(\varepsilon > \varepsilon_{\min})$ $\varepsilon \leftarrow \varepsilon - \varepsilon_{step}$
if $(\varepsilon < \varepsilon_{\min})$ $\varepsilon \leftarrow \varepsilon_{\min}$

// оновити прогноз цінності сервісу провайдера $\hat{p}$
$f(u, t_c, \hat{p}) \leftarrow f(u, t_c, \hat{p}) + \alpha \cdot (v^u(c, \hat{p}) - f(u, t_c, \hat{p}))$

// оновити значення репутації провайдера $\hat{p}$
if $(\delta \geq 0)$ $k = \mu$
else $k = \nu$

$r(\hat{p}) \leftarrow r(\hat{p}) + k(1 - \text{sgn}(r(\hat{p})) \cdot r(\hat{p}))$

// оновити множину авторитетних провайдерів
if $(r(\hat{p}) \geq \theta$ і $\hat{p} \notin P_r)$
$P_r \leftarrow P_r \cup \{\hat{p}\}$
if $(r(\hat{p}) < \theta$ і $\hat{p} \in P_r)$
$P_r \leftarrow P_r \setminus \{\hat{p}\}$

Рис. 3. Життєвий цикл брокера під час обробки завдань з черги

3. Результати моделювання

Виконаємо моделювання системи з одним брокером. Вважаємо, що завдання надходять рівномірно від усіх користувачів, і їх можна моделювати пуассонівським процесом. Приймаємо, що всі завдання, які надходять на обробку, здатний обробити кожен з доступних брокеру провайдерів.

Система змодельована для наступних співвідношень кількості користувачів брокера до кількості доступних йому провайдерів: 300 000:30, 30 000:30.

Поріг репутації, згідно з яким визначається множина авторитетних провайдерів: $\theta = 0,6$. Початкове значення репутації для провайдера, який ще не оброблював запити, встановлюється рівним $r_0 = 0$.

Налаштування параметрів навчання в системі: мінімальне значення параметра ймовірності $\varepsilon_{\min} = 0,3$, мінімальна швидкість навчання $\alpha_{\min} = 0,2$. Після кожного прийняття рішень відбувається оновлення ε: якщо параметр ε не досяг мінімального значення, то він зменшується на $\varepsilon_{step} = 0,1$. Аналогічно змінюється і швидкість навчання — з кроком $\alpha_{step} = 0,05$.

Кортеж параметрів *c*, що характеризують наданий сервіс, представлений однією характеристикою — якість сервісу. Вимоги користувачів (значення V^u) до цієї характеристики рівномірно розподілені між користувачами. Функція v^u, за якою користувач обчислює дійсну цінність наданого сервісу, лінійно залежить від якості *q*. Вважаємо, що кожний провайдер надає завжди сервіс з фіксованою якістю, значення якої також генерується за рівномірним розподілом.

Константи, що характеризують взаємодію провайдерів і брокера у даній системі, встановлюються рівними наступним значенням: фактор співпраці $\mu = 0,4$, фактор відмови від співпраці $\nu = -0,2$ — за аналогією до системи ринку у роботі [9].

Моделювання проводилося для чотирьох стратегій, які були визначені наступним чином. **Random** — брокер завжди обирає провайдера випадковим чином з усіх доступних (P_r). **RL**, **Reputation** та **Reputation (R)** обирають між дослідженням середовища та використанням набутого досвіду за ε-жадібною стратегією. Різниця між ними полягає у правилі використання досвіду: Reputation (R) обирає провайдера з максимальною репутацією із множини авторитетних, Reputation — провайдера з очікуваною цінністю з множини авторитетних, RL — провайдера з максимальною очікуваною цінністю сервісу з множини усіх провайдерів.

На рис. 4 відображено результати моделювання системи, в якій брокер обслуговує: (а) 30 000 користувачів та 30 провайдерів, (б) 300 000 користувачів і 30 провайдерів. Дані усереднено за 1000 прогонів, у кожному з яких оброблено 1000 запитів.

Розглянемо результати моделювання для стратегій RL, Reputation та Reputation (R), що наведені на рис. 4. Для цих стратегій приблизно перші 25 запитів для обох масштабів системи відбувається накопичення досвіду. Це пояснюється тим, що початкове значення параметра ймовірності $\varepsilon_0 = 1$, тому певний період система частіше обиратиме дослідження середовища. Поведінка системи зі стратегіями RL та Reputation приблизно однакова в період накопичення досвіду.

Стратегія Reputation (R) є кращою за стратегії RL та Reputation приблизно через 50 оброблених запитів, — це дає підстави вважати її перспективною для швидкої реакції системи на різкі зміни середовища. Проте у довгостроковій перспективі ця стратегія показує значно нижчу середню оцінку сервісів: через 300 запитів кращі результати показують вже стратегії RL та Reputation. Причому для системи з меншою кількістю користувачів (рис. 4,*б*) стратегія Reputation показує кращі за Reputation (R) результати вже приблизно через 50 запитів.

Також необхідно відмітити відмінності між RL та Reputation. Приблизно через 200 запитів для системи з більшою (300 000) кількістю користувачів (рис. 4,*а*) ці стратегії показують приблизно однакові результати. До цього вищу середню оцінку сервісів показує стратегія Reputation. Для системи з 30 000 користувачів цей період зростає до 300 запитів.

Таким чином, на основі аналізу отриманих результатів моделювання можна зробити наступні висновки. Якщо обирати стратегію для системи з динамічним середовищем, де різкі зміни є частим явищем і необхідно швидко реагувати на них, доцільно використовувати стратегію, що обирає провайдерів за критерієм максимальної репутації (Reputation (R)). Якщо швидкість реагування на зміни не є критичним параметром і важливішими є результати у довгостроковій перспективі, можна використовувати стратегії RL та Reputation. Причому стратегія Reputation досягає прийнятних оцінок сервісів (приблизно 75 % від максимально отриманих за час моделювання) за 6–10 % часу моделювання системи, в той час як RL досягає таких оцінок за 20–30 % часу моделювання. Отже, застосування стратегії, що вра-

ховує репутацію додатково до навчання з підкріпленням, дійсно зменшує час навчання системи.

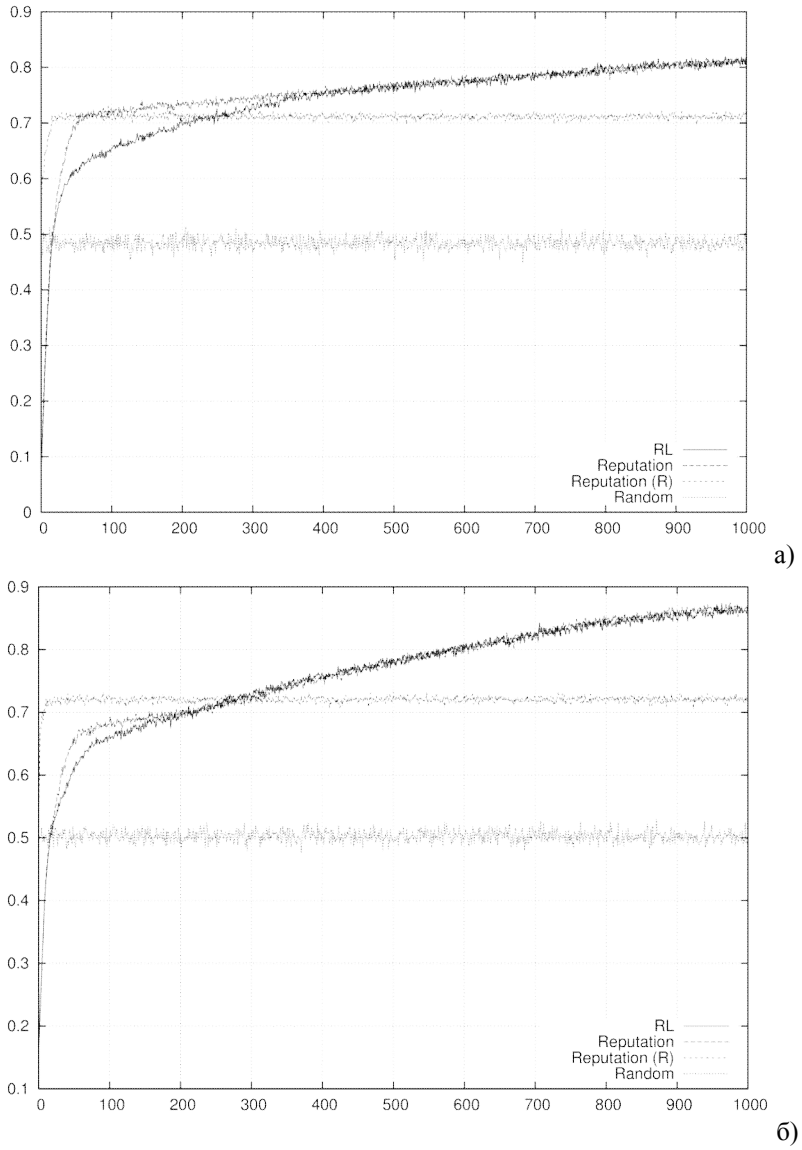

Рис. 4. Усереднена оцінка сервісу, отримана від користувача, для різних стратегій вибору провайдера. Співвідношення кількості користувачів до провайдерів: а) 300 000:30, б) 30 000:30

Висновки

У роботі для вирішення задачі вибору провайдера сервісу запропоновано застосувати підхід на основі навчання з підкріпленням, який враховує репутацію провайдерів. Було побудовано модель підсистеми вибору провайдера сервісу та відповідну стратегію, згідно з якою підсистема прийматиме рішення.

Описану систему було змодельовано для середовища зі сталими характеристиками надання сервісів провайдерами, пуассонівського потоку завдань від корис-

тувачів і трьох різних стратегій вибору провайдерів. У результаті моделювання було показано, що із запропонованою стратегією система має у середньому втричі менший час адаптації до середовища, ніж зі стратегіями, що основані на навчанні з підкріпленням, але не враховують репутацію. Водночас застосування стратегії на основі врахування репутації без навчання з підкріпленням не показала прийнятних результатів у довгостроковій перспективі. Таким чином, було показано, що в стратегіях вибору провайдера на навчанні з підкріпленням доцільно додатково враховувати репутацію провайдерів сервісів, оскільки це призводить до зменшення часу навчання системи.

Подальшими напрямками досліджень є оцінювання ефективності запропонованого підходу за умови середовища системи, характеристики якого можуть часто змінюватися за час, що є порівняно малим по відношенню до часу навчання системи. Зокрема, моделювання середовища, в якому характеристики провайдерів значно змінюються за життя системи або змінюється множина доступних брокеру провайдерів.

1. *Mell P.* The NIST definition of cloud computing / P. Mell, T. Grance [Електронний ресурс]: NIST Special Publication 800–145. — USA National Institute of Standards and Technology. — 2011. — Режим доступу: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

2. *Designing* a resource broker for heterogeneous grids / S. Venugopal, K. Nadiminti, H. Gibbins, R. Buyya // Software: Practice and Experience. — 2008. — Vol. 38, N 8. — P. 793-825.

3. *Dynamic* cloud resource reservation via cloud brokerage / W. Wang, D. Niu, B. Li, B. Liang // Proc. of the 2013 IEEE 33-rd Internation. Conf. on Distributed Computing Systems ICDCS'13. — 2013. — P. 400–409.

4. *Sutton R.* Reinforcement learning: ani / R. Sutton, A.G. Barto // MIT Press. — Cambridge, MA, 1998.

5. *Galstyan A.* Resource allocation in the grid with learning agents / A. Galstyan, K. Czajkowski, K. Lerman // Journal of Grid Computing. — 2005. — Vol. 3, N 1. — P. 91–100.

6. *Grid* differentiated services: a reinforcement learning approach / J. Perez, C. Germain-Renaud, B. Kegl, C. Loomis // Cluster Computing and the Grid, 2008 CCGRID'08. — 8-th IEEE International Symposium on. — 2008. — P. 287–294.

7. *A novel* multi-agent reinforcement learning approach for job scheduling in Grid computing / J. Wu, X. Xu, P. Zhang, C. Liu // Future Generation Computer Systems. — 2011. — Vol. 27, N 5. — P. 430-439.

8. *Watkins C.J.C.H.* Q-learning / C.J.C.H. Watkins, P. Dayan // Machine learning. — 1992. — Vol. 8, N 3. — P. 279-292.

9. *Tran T.T.* A reputation-oriented reinforcement learning strategy for agents in electronic marketplaces / T.T. Tran, R. Cohen // Computational Intelligence. — 2002. — Vol. 18, N 4. — P. 550–565.

10. *Barto A.G.* Temporal difference learning / A.G. Barto // Scholarpedia. — 2007. — **2**(11):1604.

11. *Vermorel J.* Multi-armed bandit algorithms and empirical evaluation / J. Vermorel, M. Mohri // Proc. of the 16-th European Conf. on Machine Learning. ECML'05. — 2005. — P. 437–448.

12. *Yu B.* A social mechanism of reputation management in electronic communities / B. Yu, M.P. Singh // In M. Klusch and L. Kerschberg editors — Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence. — 2000. — Vol. 1860. — P. 154–165.

Надійшла до редакції 06.10.2014