УДК 336.761.5

Веретельник В. В., Германенко Л. М., Лелеко І. Г.[1]

РИЗИКИ І ОСОБЛИВОСТІ ПРОВЕДЕННЯ АУДИТУ З ВИКОРИСТАННЯМ ХМАРНОЇ КОМП'ЮТЕРНОЇ ТЕХНОЛОГІЇ ОБРОБКИ ДАНИХ

У статті висвітлено можливості, умови та переваги проведення аудиту в умовах застосування сучасних комп'ютерних технологій. Визначено чинники, що підвищують і знижують аудиторський ризик при використанні комп'ютерних технологій обробки даних. Розглянуто основні процедури, які можуть виконуватися аудитором з використанням хмарної комп'ютерної технології. Розроблено моделі, які можуть бути використані при створенні хмарної комп'ютерної технології обробки даних, управлінні обчислювальними процесами на основі OpenStack і OpenFlow, що реалізує підвищення продуктивності за рахунок ефективного планування завдань і управління структурами даних між ними.

Ключові слова: *аудит, аудиторський ризик, хмарна комп'ютерна технологія, інформаційні системи, програмні продукти, комп'ютерна обробка даних.*

ВСТУП

За останні десятиліття значно підвищилися вимоги до організації системи обліку та звітності. У зв'язку з цим з'явилися нові форми і методи ведення обліку, у тому числі із застосуванням комп'ютерних систем. Аудит фінансової звітності як основного джерела інформації, що дозволяє оцінити фінансовий та майновий стан економічних суб'єктів, з розвитком хмарних комп'ютерних технологій став ще більш значимим для її користувачів.

Аудитор може використовувати комп'ютерні технології в аудиті як при веденні бухгалтерського обліку вручну, так і при автоматизованому обліку суб'єктів підприємницької діяльності. У першому випадку аудитору необхідно вирішувати проблему наявності необхідного програмного забезпечення для аудиту бухгалтерських записів щодо господарських операцій чи підсумкових записів відповідних облікових і звітних документів (регістрів). У другому випадку для аудиту суб'єкта підприємницької діяльності необхідно застосовувати тільки відповідну програму інформаційних технологій.

Питання комплексного дослідження технологій та засобів комп'ютерної обробки даних для аудиту розглядалися у наукових

[1] Рецензент – д. е. н., доцент Шпак Л. О.

працях багатьох зарубіжних та вітчизняних учених, серед яких Е. Чамберс, К. Кловз, Н. І. Рубан, М. Т. Білуха, Г. В. Федорова, В. Ю. Лісіна В. С. Рудницький, А. В. Бондаренко, С. В. Івахненков, Ж. А. Жирна, В. І. Подільський, Н. М. Проскуріна, Р. Вебер, Г. М. Мусіхіна, Ю. Миронова, О. І. Ястребов, В. П. Коваленко та інші. Проте пошук напрямів удосконалення аудиторських процедур та оцінки ризиків у сучасній системі аудиту, у тому числі за рахунок впровадження комп'ютерних технологій обробки даних, залишається актуальним питанням і потребує подальших досліджень.

ПОСТАНОВКА ЗАВДАННЯ

Сучасний аудит дуже тісно пов'язаний з інформаційними технологіями, оскільки зростає ступінь автоматизації облікового процесу на підприємствах, збільшується вплив комп'ютеризованої системи обробки даних бухгалтерського обліку і аудиту на достовірність і повноту інформації, яка формується у звітності. Ефективність аудиторських перевірок істотно підвущується із застосуванням самими аудиторами спеціалізованих програмних продуктів.

Метою статті є дослідження проблеми використання хмарної комп'ютерної технології обробки даних при аудиті; послідовності виконуваних традиційних аудиторських процедур з метою підвищення їх ефективності при взаємодії з людино-машинним інтерфейсом; збору, обробки та зберігання даних у комп'ютерних системах.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розвиток суспільства за останній період характеризується такими стадіями науково-технічного прогресу, як індустріалізація, комп'ютеризація, інформатизація. Нові комп'ютерні технології в аудиті сприятимуть удосконаленню інтелектуалізації аудиту та науковому обґрунтуванню його висновків. Використання інформаційних технологій в аудиті передбачено Міжнародним стандартом аудиту (далі – МСА) 401 «Аудит в умовах комп'ютерних інформаційних систем», Положенням про міжнародну аудиторську практику 1001 «Середовище комп'ютерних інформаційних систем – автономні мікрокомп'ютери» та ін.

При проведенні аудиту з використанням комп'ютерної обробки даних (далі – КОД) зберігається мета і основні елементи методології аудиту, що забезпечується через дотримання аудиторами правил і стандартів аудиту.

На вітчизняному ринку послуг автоматизації аудиту переважають компанії, що пропонують функціональні (прикладні) інформаційні технології у вигляді таких популярних програм, як «Експрес Аудит: ПРОФ», AuditExpert, «ІНЕК-Аналітик», AuditXP, «Комплекс Аудит», AuditNET, Audit Command Language (ACL), «ITAudit: Аудитор» та ін.

За своєю структурою і змістом дані програми відповідають вимогам сертифікації аудиторських послуг. Крім цього, для задоволення загальних інформаційних потреб аудитора в процесі перевірки широко застосовуються забезпечувальні інформаційні технології, серед яких можна виділити: нормативно-довідкові та інформаційно-пошукові системи («Грант», «Консультант Плюс», «Юрисконсульт», «Ліга: Закон» тощо), інформаційні технології загального призначення і засоби електронного офісу (програмні засоби пакетів Microsoft Office: Word, Excel, Access та інші), спеціалізовані інформаційно-аналітичні та статистичні системи (програмні комплекси «FinAnalitic», «SalesExpert», «Статистик-Консультант», «СтатЕксперт») та інші [1].

В умовах сучасної економіки до якості та безпеки аудиторських програм висуваються підвищені вимоги. Крім того, якісна аудиторська програма повинна мати високий рівень виконання і відповідати іншим вимогам: коректності, надійності, зручності використання, гнучкості, масштабованості, відкритості, безвідмовності, продуктивності. Всі ці показники характеризують рівень якості аудиторської програми. Для перевірки безпеки програм можуть використовуватися як загальні технології тестування і забезпечення якості, так і спеціальні, прийнятні для цих напрямків.

Якщо розглядати етапи складання плану аудиторської перевірки відповідно до стандартів МСА, то потрібно враховувати рівень автоматизації обробки облікової інформації, наявність особливостей інформаційного, програмного та технологічного забезпечення у суб'єкта підприємницької діяльності, способи передачі і зберігання даних, особливості організаційної форми обробки даних, включаючи використання мережевих і локальних систем. Робочі документи, що сформовані у процесі аудиту із застосуванням комп'ютерних програм, можуть зберігатися в аудиторській фірмі, окремо в архівах на машинних носіях або безпосередньо в хмарі.

Підвищити ефективність аудиторських процедур можливо завдяки використанню хмарної комп'ютерної технології обробки даних при проведенні аудиту.

Основні процедури, які можуть виконуватися аудитором для контролю з використанням хмарної комп'ютерної технології обробки даних, включають: спостереження за послідовністю даних, які беруть участь не в одному етапі обробки; контроль попередніх даних; виконання прогнозу та плану результатів перевірки даних; зіставлення результатів перевірки з контрольними даними для окремих операцій і за окремими видами діяльності в цілому; оцінка працездатності програмного й апаратного забезпечення аудиторської роботи за допомогою комп'ютерної обробки даних; оцінка відповідності їх сучасним вимогам; контроль відповідності певного комп'ютерного забезпечення суб'єкта підприємницької діяльності, що перевіряється,

чинному законодавству; використання хмарних комп'ютерних технологій економічним суб'єктом під час проведення аудиту.

У сучасних умовах у більшості організацій (підприємств) здійснення фінансово-господарської діяльності відбувається в умовах КОД. Застосування комп'ютерних технологій для обробки бухгалтерської інформації вимагає від аудиторів професійних знань для ідентифікації та оцінки ризиків, пов'язаних із комп'ютеризованою обробкою даних.

Аудиторський ризик є основним критерієм, який характеризується якістю роботи аудитора, в основі якого лежить його професійна думка. Ризики в системі КОД регулюються стандартом аудиторської діяльності. Поняття ризику аудитора полягає у ймовірності того, що за підсумками аудиторської перевірки бухгалтерська фінансова звітність може містити невиявлені суттєві викривлення після підтвердження її достовірності, або містить суттєві викривлення, коли насправді таких спотворень не існує. Розуміння системи бухгалтерського обліку і внутрішнього контролю є важливим для аудитора, що базується на основі накопиченого досвіду і доповнюється вивченням документів і записів, спостереженням за організацією комп'ютерної обробки операцій. Мета внутрішнього контролю в системі КОД – забезпечення достатньої впевненості у надійності обробки облікової інформації.

Найбільш поширеною моделлю виявлення ризику аудитора є:

$$АР = ВР \times РК \times РН, \quad (1)$$

де *АР* – загальний аудиторський ризик; *ВР* – властивий (внутрішній) ризик; *РК* – ризик контролю; *РН* – ризик невиявлення.

Застосування хмарної комп'ютерної технології обробки даних істотно впливає на організаційну структуру підприємства, яка сприяє специфічним ризикам у системі обліку і внутрішнього контролю. Використання стандартної комп'ютеризованої системи обробки даних змінює процедури запису облікових даних і розширює коло осіб, які отримують доступ до бухгалтерських записів, що може привести до появи таких ризиків, як: відсутність можливості спостереження за рознесенням облікових даних по регістрах, їхнім закриттям і складанням поточної звітності; перешкоди контролю доступу до бази даних і програм системи КОД несанкціонованих користувачів; наявність спеціальних документів на електронних носіях, які розробляються і використовуються всередині спеціалізованими службами організації; зниження можливості контролю й аналізу поточної облікової інформації в режимі реального часу; відсутність регістрів, які містять поетапні записи підсумкових даних обробки операцій певного виду, що обмежує можливості спостереження за їхнім рознесенням перед складанням бухгалтерської звітності;

неможливість архівації проміжних записів, що приводить до постійного оновлення поточної інформації; підвищення ризику несанкціонованих бухгалтерських записів і змін даних тощо.

Введення даних в умовах їх комп'ютерної обробки може одночасно змінювати одразу декілька функцій. Особливості системи значно збільшують вплив помилки введення на бухгалтерську інформацію. При цьому невірна сума буде проведена не за одним рахунком, як при ручній обробці даних, а відразу за кількома. Певна частина бухгалтерських проведень може генеруватися системою КОД самостійно, тобто без складання первинних документів і без втручання фахівців. Це може призвести до появи несанкціонованих записів і розрахунків в обліку, які складно виявити через відсутність спеціальних регістрів чи документів. Прикладом може бути автоматичне нарахування відсотків за заборгованістю покупців чи за позиками.

Бази даних можуть зберігатися тільки в електронному вигляді, що збільшує ризик їхньої втрати внаслідок поломки, зараження комп'ютерними вірусами, несанкціонованого проникнення в інформаційні системи.

Помилки з'являються при використанні комп'ютерних програм на підприємстві у таких випадках: при підготовці вихідних даних, при першому запуску системи; при передачі вихідної інформації від однієї частини програми до іншої; при запиті інформації для отримання додаткових відомостей за окремими угодами; при коригуванні головних файлів, при їхній зміні (додаванні, видаленні або зміні записів); при генеруванні різноманітних форм вихідних звітів чи файлів; при виправленні поточних помилок у даних, які виявлені в результаті реалізації процедур контролю; використанні персоналом підприємства вихідних даних і пристроїв [4].

Аудиторський ризик значною мірою залежить від якості здійснення внутрішнього контролю, а саме від контролю процесу підготовки даних, запобігання формальних помилок під час роботи із системою КОД, контролю за дотриманням у поточній роботі нормативно-правових аспектів чинного законодавства, взаємодії між користувачами системи і службою інформаційної підтримки. Контроль якості має ґрунтуватися на дотриманні співробітниками аудиторської фірми вимог незалежності, чесності, об'єктивності, конфіденційності, професіоналізму при виконання всіх видів робіт в ході аудиту.

Ризик аудитора змінюється в умовах застосування клієнтом спеціальних програмних засобів обробки облікової інформації. Факторами, які можуть підвищувати аудиторський ризик при комп'ютеризованій обробці даних на підприємстві у процесі аудиту, є: децентралізація і географічна розпорошеність системи КОД; відсутність механізмів внутрішнього контролю за функціонуванням

інформаційного облікового середовища підприємства; відсутність заходів з обмеження несанкціонованого доступу до системи КОД.

До факторів, що знижують аудиторський ризик, можна віднести: наявність у підприємства ліцензії на використовувані пакети прикладних облікових програм; наявність у аудитора апробованих засобів тестування достовірності алгоритмів, реалізованих у програмних продуктах клієнта; існування дієвої організаційної системи внутрішнього контролю використовуваного програмного забезпечення; наявність централізованої інформаційної політики, що визначається керівництвом підприємства, на якому проводиться аудит; застосування уніфікованого програмного забезпечення усіма підрозділами організації, її філіями і дочірніми компаніями, та робота в єдиному середовищі КОД; формування довгострокового плану розвитку інформаційної системи економічного суб'єкта.

Таким чином, ризик аудитора виступає основним критерієм при оцінці якості його роботи, в основі якого лежить професійна думка аудитора щодо стану організації бухгалтерського обліку та формування фінансової звітності.

Оцінка ризиків у сучасній системі аудиту визначається ефективним управлінням комп'ютерної технології обробки даних, яка передбачає використання моделі хмарної комп'ютерної технології. Хмарний обчислювальний кластер може бути описаний у формі зваженого неорієнтованого мультиграфа виду:

$$Cloud = (Devices, Links, type, w_{\text{n}}, w_{\text{sw}}.,w_{\text{st}}, w_L,Flavors,Users), \quad (2)$$

в якому вершини являють собою множини:

$$Devices = Nodes \text{ u } Switches \text{ uu } Storages \text{ u } Controllers, \quad (3)$$

де $Nodes = (\text{Nj},N_2,\ldots,N_n\}$ i позначає безліч обчислювальних вузлів; $Switches = (\text{Sj},S_2,....,S_m\}$ – комутатор; $Storages = (\text{Fj},F_2,\ldots,F_q\}$ – сховище даних; $Controllers = (H_1,H_2,\ldots,H_z\}$ – контролерів OpenFlow. Ребра мультиграфа $Links = (L-\}$ являють собою двосторонні мережеві зв'язки між пристроями мережі, але допускається наявність декількох паралельних зв'язків між двома пристроями.

Тип мережевого пристрою може бути визначений за допомогою класифікованої функції *type:*

$$Devices- > \{\text{" } node\text{", " } switch\text{", " } storage\text{", " } controller\text{"}\} \quad (4)$$

Відображення w_n для кожного обчислюваного вузла N_i задає вектор його характеристик:

$$^{w}\text{n. } ^{(N}\text{i)} = (^{w}\text{n. stat. } ^{(N}\text{i }^{)}, ^{w}\text{n. dyn. } ^{(N}\text{i, t)}\text{I}, \quad (5)$$

де $^{w}{}_{,,,}\ stat.^{(N}i^{)}$ i $^{w}n.\ dynX^{N}i^{,\ t)}$ (6)

відповідно означають статистичні параметри і динамічні характеристики N. Статистичні параметри вузла представляють вектором:

$$^{w}{}_{,,.}\ stat.^{(N}i^{)} = (^{M}i\ ,\ ^{D}i\ ,\ ^{C}i\ ,\ ^{P}i\ ), \quad (7)$$

який містить розмір його ОЗП M_i, дискової пам'яті (далі – ДП) D_i, число обчислювальних ядер Ci його характеристики продуктивності:

$$Pi = (P_{i1}, P_{i2}, \ldots, P_{iC}.). \quad (8)$$

Динамічні характеристики можуть бути задані вектором:

$$^{w}n.\ dyn.\ ^{(N}i^{,\ t)} = (^{m}i(^{t}I^{d}i^{(t),}\ ^{c}i(^{t}I^{u}i^{(t),}\ ^{vm}i^{(t))}. \quad (9)$$

Тут $m_i(t)$ і $d_i(t)$ описують відповідно об'єм доступної оперативної пам'яті (далі – ОП) і дискової пам'яті вузлами моменту часу $t > 0$; $c_i(t) = (c_{i1}(t),c_{i2}(t),...,c_{iCi}(t))$ – вектор ознак зайнятості обчислювальних ядер ($c_{ik}(t) = 0$ – ядро вільне, $c_{ik}(t) = 1$ – зайняте) у момент часу t.

$$u_i(t) = (u_{i1}(t),u_{i2}(t),...,U_{iCi}(t)). \quad (10)$$

Вектор завантаженості обчислювальних ядер у момент часу t. $vm_i(t)$ описує набір обчислювальних задач, які виконуються на вузлі N_i в момент часу t. Дана інформація може збиратися менеджером мережі хмарного обчислювального центру обробки даних (далі – ЦОД) через регулярні інтервали часу за допомогою протоколу SNMP [5].

Кожний *OpenFlow*-комутатор, також як і вузол, має статичні параметри і динамічні характеристики:

$$^{w}sw.\ ^{(S}J^{)} = (^{w}sw.\ stat.\ ^{(S}J^{),\ w}sw.\ dyn.\ ^{(S}j^{,\ t))}\ . \quad (11)$$

Статичні параметри Sj включають такі значення:

$$^{w}sw.\ stat.\ ^{(S}J^{)} = (^{Et}j^{,\ Pc}j^{,\ OF}j^{,\ Ts}-^{),}, \quad (12)$$

де *Etj є ("100Mbit Ethernet",* "1 *Gbit Ethernet",* "10 *Gbit Ethernet"} визначає* підтримувану версію протоколу Ethernet; *Pcj* – кількість портів комутатора; *OF-є*("1.0", "1.1", "1.2", "1.3"} – підтримувана версія протоколу OpenFlow. У комутаторі таблиця *OpenFlow* має максимум *Ts* – записів про потоки.

Динамічні характеристики комутатора можуть бути представлені вектором:

$$^{w}sw.\ dyn.^{(S}J^{,\ t)} = (^{Ft}J^{(t),}\ Qj^{(t}\ I^{\ Pt}J^{(t))}, \quad (13)$$

де $Ft\ j\ (t)$ – стан таблиці потоків (flowtable) O*penFlow* у момент часу t.

Кожний запис-правило *OpenFlow* (flowentry) має такий вигляд:

$$Rule_t = (Match, Counters_l, Actions_l), \quad (14)$$

де Match – наявність декількох паралельних зв'язків між двома прилаштуваннями; $Counters_l$ – статичні лічильники; $Actions_l$ – дії, які виконуються над пакетом.

Всі пакети, що надходять у комутатор, зіставляються з усіма правилами з таблиці потоків. Якщо необхідне правило знайдено (його *Match відповідає заголовкам пакета*), тоді виконуються всі дії $Actions_{t,}$. Ці правила і оновлюють значення лічильників $Counters_l$, в іншому випадку пакет відправляється контролеру, асоційованому з комутатором.

Контролер відповідальний за визначення способу обробки пакетів, для яких не знайшлося підходящих правил у таблиці комутатора. Після прийняття рішення контролер додає або видаляє правила у таблицях потоків даного та інших комутаторів.

Протокол *OpenFlow* версії 1.0 підтримує такі поля для задання частини *Match* правил комутації: номер вхідного порта, Ethernet-адреси джерела і отримувача, тип Ethernet-пакета, ідентифікатор *VLAN*, IP-адреса джерела і отримувача, тип протоколу IP, *TypeofService (ToS)* біти протоколу *IP* і *TCP/UDP*-порти джерела і отримувача.

Кожному полю *Match* може задаватися конкретне значення або *«ANY»* (будь-яке значення). Наявність даних полів дозволяє аналізувати заголовки пакетів на рівнях L2–L4, що дає можливість на основі *OpenFlow* реалізувати комутацію, маршрутизацію, мережевий брандмауер, систему виявлення вторгнень і т. п.

Actions може містити такі дії [6]:

– Forward – відправити пакет на конкретний порт комутатора;

– Forwardall – відправити пакет на всі порти;

– Forwardcontroller – виконати інкапсуляцію пакета і відправити його *OpenFlow*-контролеру;

– Forwardlocal – відправити пакет у локальний мережевий стек комутатора;

– Forwardtable – виконати дії у таблиці;

– Forwardinport – відправити пакет назад у вхідний порт;

– Forwardnormal – обробити пакет без *OpenFlow* стандартними засобами комутатора;

– Forwardflood – відправити пакет по мінімальному покриваючому дереву, що не включає порт;

– Enqueue – помістити пакет у задану чергу QoS;

– Modifyfield – змінити задане поле заголовка пакета.

Порожній список дій *Actionsl* означає видалення пакета. Найчастіше використовується така дія, як комутація пакета на зазначений порт. *OpenFlow* підтримує лічильники для кожного запису про кожний потік окремо. Countersl містить такі значення: загальне число пакетів, оброблених даними правилом; загальний розмір усіх

пакетів у байтах, оброблених даними правилом; тривалість дії правила після додавання в комутатор у секундах і мілісекундах.

Дані метрики дають уявлення про використання записів про потоки.

На рівні таблиці *Ftj (t)* є такі лічильники: кількість активних правил, кількість пошуків пакетів, кількість зіставлених пакетів.

Дані значення можуть збиратися через регулярні інтервали часу за допомогою протоколу *OpenFlow.*

У формулі *Qj (t) = {Qjkr (t)}* позначає набір черг пакетів, асоційованих з конкретним портом комутатора і значенням *ToS.* Вони використовуються, щоб забезпечити згідно з *QoS* мінімальну гарантовану пропускну спроможність для заданих мережевих зв'язків.

З кожною подібною чергою *Qjkr (t)* пов'язані такі метрики: кількість переданих пакетів, число переданих байтів і кількість помилок, що виникають при переповненні.

$$Ptj\ (t) = \{Ptn\ (t), Ptj2\ (t), ..., Pt]\ Pcj\ (t)\} \quad (15)$$

являє собою набір динамічних характеристик портів комутатора S. До їхнього числа належать: стан порту (включений або виключений), загальні кількості отриманих і переданих пакетів, загальне число отриманих байтів, загальне число переданих байтів, загальні кількості отриманих та переданих повідомлень про видалення пакетів, загальні кількості отриманих і переданих помилок, загальна кількість отриманих помилок вирівнювання Фрейма, загальна кількість отриманих *CRC*-помилок, загальне число колізій.

Дані метрики також збираються за допомогою протоколу *OpenFlow* через регулярні інтервали часу. Вони допомагають виявляти перевантаження мережевих зв'язків і комутаторів, їхні відмови.

Мережеві сховища містять обчислювальні завдання (образи примірників віртуальних машин), бази даних додатків, а також інфраструктурні компоненти хмарної обчислювальної системи. Кожне сховище має такий вектор значень, що є частиною формули:

$$Wst.\ (Fk) = (Vlk, vlk\ (t), rk\ (t), wk\ (t)), \quad (16)$$

де *Vlk* – загальний розмір сховища; *vlk (t)* – розмір вільної його частини у момент часу *t; rk (t) і wk (t)* – відповідність встановлення середньої швидкості зчитування і запису даних на момент часу *t.*

Для кожного мережевого зв'язку *L(j) Links* функції *wl (Lij)* визначається вектор статичних параметрів і динамічні характеристики:

$$^{w}l.^{(l}ij^{)} = {}^{(B}ij^{,\,b}ij^{(t}X^{lat}ij^{(t\wedge}, \quad (17)$$

де *Bj* – максимальна пропускна здатність мережевого зв'язку, яка вимірюється за умови присутності мережевої конкуренції; *bj (t)* –

пропускна здатність у момент часу *t*; *latch (t)* – значення латентності у момент часу *t*.

Останні дві характеристики враховують мережеву конкуренцію. Типи можливих примірників віртуальних машин задаються набором *Flavors = [Fl1, Fl2, ..., Fle}* із формули (1). Кожен тип характеризується такими параметрами:

$$Flk = (CL, Mrk, DrkX), \quad (18)$$

де *Crk* – кількість віртуальних процесів; *Mrki Drk* – відповідно обсяги ОП і ДП. Всі користувачі хмарного обчислювального ЦОД формують безліч

$$Users = \{U1, U2, ..., Uv\}. \quad (19)$$

Дана модель адекватно описує хмарні обчислювальні ЦОД із довільними топологіями, у тому числі з розподіленими сегментами, керованими окремими контролерами *OpenFlow*. Модель орієнтована на підтримку версії 1.0 протоколу *OpenFlow*, що дозволяє за рахунок зворотної сумісності моделювати довільні контролери і *OpenFlow*-комутатори. Вибір даної версії також мотивований тим, що сьогодні більш нові версії протоколу не підтримувані апаратними комутаторами.

Кожна обчислювальна задача у хмарному обчислювальному ЦОД може бути представлена у вигляді набору примірників віртуальних машин. Вони пов'язані між собою повнозв'язковим чином, тобто кожен екземпляр може по мережі обмінюватися пакетами з будь-яким іншим. Обчислювальне завдання описується таким вектором:

$$Jk. = (Uk, Ik, Bmin.kX), \quad (20)$$

$$Ik = (Ik1, Ik2, ..., Ikgt), \quad (21)$$

де *Uk* – користувач – власник завдання; *(Ik1, Ik2, ..., Ikgt)* – набір екземплярів віртуальних машин; *Bmink* – мінімальна гарантована пропускна здатність зв'язків між віртуальними машинами, що задається користувачами.

Кожен екземпляр віртуальної машини *Ikj* визначається таким набором значень:

$$Ikj = (Imkj, Flkj, stkj (t), hkj (tIakj (tIcnkj (t)). \quad (22)$$

Тут *Imkj* задає дисковий образ екземпляра віртуальної машини, *Flkj* визначає тип екземпляра, стан задається значенням *stkj (t) є {"queued", "running "," migrating ","Stopped", "terminated"}.hkj (t)* – номер вузла, на якому виконують у момент часу t, якщо *stkj (t) Ф "running" istkj (t) Ф "migrating", mohkj (t) = -1* (неіснуючому вузлі). Номер вузла може змінюватися в часі у разі міграції віртуальної

машини. Відображення екземпляра обчислювального ядра вузла *Nh {t)* задається безліччю

$$akj\ (t) = \{akj1\ (tXakj2\ (tl\ ..\ ^\ akjCrFjj\ (t)\}, \quad (23)$$

де *akji (t)* – номер ядра вузла, зайнятого у момент часу *t*; *cnkj (t)* являє собою набір метрик екземпляра віртуальної машини, включаючи час її запуску, час зупинки, сумарний процесорний час.

Запропоновані моделі орієнтовані на підтримку хмарної системи *OpenStack* і повністю відображають особливості її функціонування, включаючи підтримку *IaaSi* – багатоорендовану архітектуру. Вибір даної системи мотивований її популярністю, відкритістю, документуванням, наявністю вихідних кодів, а також можливістю її адаптації під довільні архітектури обчислювальних систем. Основні недоліки *OpenStack*: неефективний алгоритм планування екземплярів віртуальних машин, який виконує вибір вузлів для запуску на основі їхнього ранжування за лінійним згортанням характеристик; відсутність ефективних засобів планування та локалізації мережевого трафіка між екземплярами віртуальних машин; відсутність гарантій щодо пропускної здатності віртуальної мережі.

Справжня робота спрямована на усунення даних недоліків за рахунок використання адаптованих варіантів раніше розроблених авторами алгоритмів планування завдань для кластерних обчислювальних систем, програмних комп'ютерних систем (далі – ПКС) на основі протоколу *OpenFlow*, а також мережевих засобів забезпечення *QoS*.

Запропоноване рішення на основі *OpenStack* і *OpenFlow*.

На рис. 1 [7, с. 121] наведена схема планування завдань, що запропонована у рамках цієї роботи. Сірим кольором позначені елементи, які розроблені авторами.

Опишемо основні елементи схеми. Обчислювальні вузли пов'язані за допомогою комутаторів із підтримкою *OpenFlow*. Використання даних комутаторів дозволяє управляти потоками даних груп віртуальних машин із метою їхньої топологічної локалізації та заниження мережевих конкуренції між ними.

Диспетчер займає центральне місце у хмарній обчислювальній системі, він вирішує проблеми управління і планування обчислювальних задач.

Контролер *OpenFlow* являє собою центральну компоненту ПКС. Він зосереджує у собі всю програмну логіку управління маршрутизацією пакетів; використовується для формування таблиць комутації в *OpenFlow*-комутаторах на основі розрахованих потоків даних між призначеними на фізичні вузли обчислювальними завданнями. Також використовується для збору мережевої статистики з комутаторів *OpenFlow* і передачі її службі управління мережею.

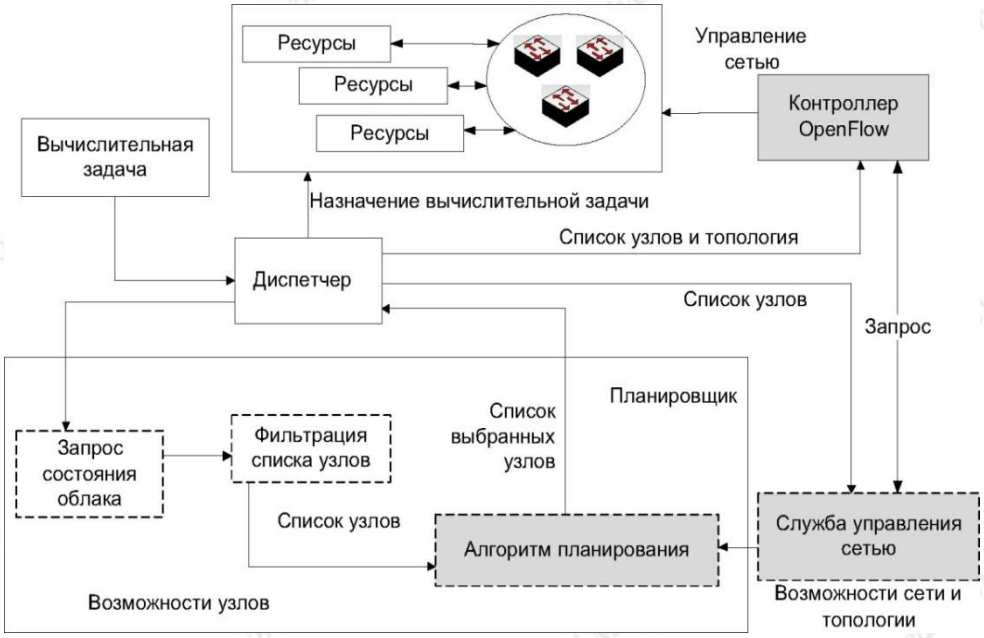

Рис. 1. Схема планування і поліпшення варіанта системи *OpenStack*

Джерело: створено авторами на основі [7]

Служба управління мережею – служба, що реалізує механізми збору статичної та динамічної інформації по хмарному обчислювальному ЦОД. Включає виявлення топології мережі на основі протоколу *LLDP*, отримання статистики від контролера *OpenFlow*, збір інформації про завантаження обчислювальних ядер і пам'яті вузлів за допомогою *SNMP* [7]. Ця служба надає динамічні характеристики для моделі хмарного обчислювального ЦОД. Планувальник завдань на основі запиту користувача на виділення ресурсів для обчислювальної задачі, а також відомостей про стан хмарної обчислювальної системи добирає оптимальні обчислювальні вузли для запуску віртуальних машин. Він спирається на описані в рамках цієї роботи моделі, а також на алгоритми планування, які є модифікованими варіантами алгоритмів *BackfillMDM* і *BackfillSDM* для кластерних систем.

ВИСНОВКИ

Отже, сучасний аудит тісно пов'язаний з інформаційними технологіями. Використання хмарної комп'ютерної технології обробки даних при проведенні аудиту істотно підвищує ефективність аудиторських процедур. Розроблені моделі можуть бути використані при створенні хмарної комп'ютерної технології обробки даних, управлінні обчислювальними процесами на основі *OpenStack* і

OpenFlow, що реалізує підвищення продуктивності за рахунок ефективного планування завдань і управління структурами даних між ними. Подальші дослідження припускають реалізацію системи управління мережевими ресурсами для обчислювального ЦОД і експериментальні дослідження ефективності запропонованих рішень.

СПИСОК ЛІТЕРАТУРИ

1. Ходаківська Л. О. Комп'ютерні технології аудиту в умовах розвитку сучасних інформаційних систем / Л. О. Ходаківська, К. С. Ходаківська [Електронний ресурс]. – Режим доступу: www.pdaa.edu.ua/sites/default/files/nppdaa/spec/136.pdf. 2. Зоріна О. А. Автоматизація аудиту в Україні: проблеми та перспективи розвитку / О. А. Зоріна // Бухгалтерський облік, аналіз та аудит: проблеми теорії, методології, організації. – 2008. – № 2 – С. 25. 3. Кот О. С. Теоретико-методичні аспекти аудиту в умовах комп'ютерних технологій / О. С. Кот // Вісник Університету банківської справи Національного банку України. – 2008. – № 3. – С. 192–195. 4. Адамс Р. Основы аудита : пер. с англ. / Р. Адамс ; под ред. Я. В. Соколова. — М. : Аудит, ЮНИТИ, 2010. 5. Полежаев П. Н. Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора / П. Н. Полежаев // Параллельные вычислительные технологии (ПАВТ'2010) : труды международной конференции. – Челябинск : ЮУрГУ, 2010. – С. 287–298. 6. Кнут Д. Искусство программирования. Основные алгоритмы / Дональд Кнут. – Том 1. – [3-е изд.]. – М. : Вильямс, 2006. – 720 с. 7. Марков А.А. Теория алгорифмов / А. А. Марков, Н. М. Нагорный. – [2-е изд., испр. и доп.]. – М. : Фазис, 1996. – 448 с.

Дата надходження до редакції – 10.09.2015 р.