схеми організації ієрархії та етапів побудови цільових рішень та логіки їх формування.

1. *Пальчевський Б.О*. Дослідження технологічних систем. - Львів: Світ. 2001 – 232с.
2. *Месарович М., Такахара Я*. Общая теория систем: математическая основа. – М.: Мир, 1978. – 311с.
3. *Катренко А*. Системний аналіз об'єктів та процесів комп'ютеризації. – Новий світ. 2000. – 424 с.
4. *Лямец В.И., Тевяшев А.Д.* Системный анализ. – Харків ХНУРЭ. 2004 – 448с.
5. *Горбатов В.А*. Теория частично упрощенных систем. М.: Сов. Радио, 1976 – 336с.
6. *Поспелов Д.А*. Логико-лингвистические модели в системах управления. - М.: Энергоиздат, 1981 – 232с.

Поступила 18.08.2014р.

УДК 681.3, 621.3

О.М.Колодчак, НУ «Львівська політехніка», каф. ЕОМ

МЕТОД РОЗПОДІЛУ РЕСУРСІВ В СИСТЕМАХ, ПОБУДОВАНИХ НА ХМАРНИХ ТЕХНОЛОГІЯХ, НА ОСНОВІ МОДИФІКОВАНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ

Розглянуто метод розподілу ресурсів в системах, побудованих на хмарних технологіях. В основі методу був використаний модифікований генетичний алгоритм. Зроблені висновки щодо перспектив використання запропонованого методу.

Ключові слова: хмарні технології, віртуальна машина, модифікований генетичний алгоритм, центр обробки даних.

The method of allocation of resources is considered in the systems, built on cloudy technologies. In basis of method was used a genetic algorithm is modified. The done conclusions are in relation to the prospects of the use of the offered method.

Keywords : cloudy technologies, virtual machine, a modified genetic algorithm, DPC.

Вступ

Останні роки все більшої популярності набувають так звані хмарні технології або хмарні обчислення (cloud computing). Хмарні технології — це технологія, яка надає користувачам Інтернету доступ до комп'ютерних ресурсів сервера і використання програмного забезпечення як онлайн-сервіса. Тобто навіть зі смартфона, який має підключення до інтернету зможна виконувати складні обчислення, опрацьовувати свої дані використовуючи потужності віддаленого сервера. Для пересічного користувача, здавалось б це

не дуже потрібно, але от невеличким фірмам, установам, які не мають змоги придбати і обслуговувати власні сховища даних або опрацювання складних обрахунків, це безумовно вихід з ситуації. Зрозуміло, що все це не безкоштовно і за використання різноманітних послуг слід платити, але судячи з розвитку інформаційних технологій(ІТ) таку технологію вважають перспективною. Можна отриматим наступні послугі: інфраструктура як послуга IaaS (Infrastructure as a Service), платформа як послуга PaaS (Platform as a Service) та програмне забезпечення як послуга SaaS (Software as Service).

IaaS зазвичай надає уніфіковані апаратні і програмні ресурси, але в деяких випадках і на інфраструктурному рівні для встановлення прикладного забезпечення (ПЗ) з оплатою по мірі використання. Замовлена інфраструктура може динамічно масштабуватися. Типові приклади - Amazon EC2 (Elastic Cloud Computing) Service і Amazon S3 (Simple Storage Service).

PaaS надає більш високий рівень сервісу, що дозволяє розробляти, тестувати і впроваджувати власні програми. Вбудована масштабованість накладає обмеження на тип розробляються. Приклад - сервіс Google App Engine, що дозволяє впроваджувати Web-додатки на тій же системі, на якій працюють власні додатки Google.

SaaS пропонує готову аплікацію,яку використовує клієнт як послугу. Канонічний приклад - Salesforce та її онлайнова система управління відносинами з клієнтами. Аналітики Forrester з'ясували, що серед різних варіантів хмарних середовищ переважає SaaS.

Оскільки в ІТ-інфраструктурах центрів обробки даних(ЦОД) зосереджені потужні і досить дорогі ресурси, виникає проблема їх ефективного використання. Це стає можливим за рахунок розподілу, управління і диспетчерування ресурсів і навантаженням ЦОД на основі комплексу відповідних математичних моделей і методів. Формалізація проблеми ефективного використання ресурсів привела до задач лінійного і нелінійного, булевого і цілочисельного, а також змішаного програмування з широким вибором критеріїв оптимізації і врахуванням ресурсних, часових, технологічних та інших обмежень. Одним з потужних засобів вирішення задачі оптимізіції на сьогоднішній день є генетичні алгоритми (ГА) та їх модифікації.

Дана стаття присвячена опису методу оптимального розподілу ресурсів в системах, побудованих на хмарних технологіях, за допомогою модифікованого генетичног алгоритму (МГА). Розглядався випадок в якому були враховані наступні пункти: ресурси одного фізичного сервера поділяються декількома віртуальними машинами (ВМ); застосовується технологія повної віртуалізації чи паравіртуалізації. У разі збільшення кількості клієнтських запитів до сервера застосувань відповідно збільшиться і кількість запитів від сервера застосувань до сервера баз даних. Отже виникає потреба збільшення ресурсів обох ВМ за рахунок решти ВМ. Збільшення ресурсів лише однієї з цих ВМ не призведе до значного підвищення ефективності, оскільки інша буде грати роль вузького місця, а

збільшення ресурсів однієї ВМ за рахунок іншої, навіть навпаки, може призвести до зменшення загальної ефективності роботи трирівневих застосувань внаслідок ще більшого загострення проблеми.

В результаті була розроблена нова методика розгортання ВМ на серверах з використанням МГА.

Постановка проблеми

Проаналізувавши ринок хмарних технологій, можна сказати, що основна його маса зосереджена на реалізації віртуальних машин в хмарі.

Існуючі технології тимчасового збільшення ліміту споживання ресурсу не здатні вирішити проблему динамічної зміни ресурсів. Тобто машина недоотримує ресурси тоді, коли вони їй найбільше потрібні.

Інша проблема - коли ресурси, фактично, довелося сплатити, але використовувати не вдалося. Віртуальній машині вночі просто не потрібно стільки ресурсів. Вона простоює, але оплата за неї проводиться повністю.

Виникає проблема: людина змушена замовляти ресурсів більше, ніж потрібно в середньому, для того, щоб переживати без проблем піки. В інший час ресурси простоюють. Провайдер бачить, що сервер не навантажений, починає продавати ресурсів більше, ніж є. У якийсь момент, наприклад, через пік навантаження на декількох клієнтів, провайдер порушує свої зобов'язання.

Саме відповіддю на цю проблему і стала ідея динамічного маштабування в хмарі. Хмарні технології замість лімітів і квот, які оплачуються незалежно від реального використання, надають користувачам можливість використовувати ресурси без обмеження, з оплатою реального споживання.

Виходячи з цього потрібно сформувати способи виділення ресурсів та їх підрахунку, способи оплати даних послуг та розробити методи, які б допомогли це реалізувати. Вищезгадані технології дозволяють забезпечити динамічне виділення ресурсів та їх моніторинг, але нерозкритим залишається питання із випадками, коли необхідно виділяти більшу або меншу кількість ресурсів та періодами зняття показників для подальшої оплати клієнтів.

Огляд існуючих рішень

Накопичений досвід експлуатації великих розподілених ІТС на початку нового тисячоліття призвів до формування уявлень щодо ІТ-інфраструктури(ІТ-І) ІТС як складного об'єкта управління.

На сьогодні ринок систем управління(СУ) ІТ-І достатньо насичений продуктами відомих виробників. Засади управління ІТ-інфраструктурою викладені в IT Infrastructure Library (ITIL), на якій базується концепція IT Service Management (ITSM). Базовими функціями СУ ІТ-І є моніторинг, управління навантаженням і ресурсами, адміністрування ІТ-інфраструктури. Автори ITIL і ITSM виходять з позицій підтримки системи бізнес-процесів компанії. Принциповими положеннями є підтримка життєвого циклу ІТС, створення єдиного інформаційного середовища, в якому централізовано

підтримуються процеси розгортання і експлуатації IT-інфраструктури.

Провідні позиції посідають компанії Microsoft з MS Data Center, IBM з лінійкою продуктів Tivoli, HP з OpenView Service Desk, до складу якої входить Automation Manager. На ринку представлені продукти багатьох інших виробників. Компоненти СУ IT-I включають до продуктів інших класів, насамперед систем збереження даних, наприклад Veritas Storage Foundation від Symantec, Rainfinity від EMC та інші.

Створення СУ IT-I породило низку наукових проблем, насамперед розподілу ресурсів, управління навантаженням, балансування та ін. Успішно вирішено проблему розподілу ресурсів в межах єдиної апаратної системи. Значних успіхів досягнуто у розробленні моделей розподілу ресурсів у мережах. Насамперед запропоновано низку підходів для передбачуваного планування ресурсів процесора і розподілу мережевої пропускної здатності, пам'яті, дискового простору і загальнодоступних сервісів на одиночних серверах. Але управління групами серверів і кластерних систем вимагає додаткових досліджень. Є специфічні комбінації технологій ЦОД, для яких немає відповідних рішень, наприклад розподіл ресурсів між ЦОД.

Потрібно побудувати моделі для управління ресурсами в зазначених випадках. Крім того, необхідне комплексне рішення для розподілу і управління ресурсами ЦОД, яке можна було б застосовувати в будь-якій ситуації, налаштовуючи його на особливості середовища застосування. Тому необхідно розробляти нові ефективні методи управління навантаженням і ресурсами в різних умовах функціонування ІТС.

Постановка задачі

Нехай при наданні сервісу IaaS ресурси віртуального серверу розподіляються нодами або ВМ, а облік їх використання здійснюється в нодо-годинах. Користувач при підписанні договору з провайдером на надання сервісу вказує фіксовану кількість нод, якою він може користуватись в будь-який момент часу гарантовано, і яка буде оплачуватись ним навіть під час простою. Крім того, нехай користувач має можливість указати кількість нод, що будуть надані йому додатково, у випадку потреби з його сторони та наявності відповідних ресурсів у хмарі. Ці додаткові ноди оплачуються замовником лише по факту їх використання. І ще, користувач має можливість вибрати тип ВМ з можливих, які надає даний провайдер. Зрозуміло, що під час збільшення клієнтських запитів до ресурсів замовника, він зацікавлений у отриманні додаткових ресурсів для забезпечення високої якості надання сервісів кінцевим клієнтам. Провайдер також зацікавлений у наданні додаткових ресурсів замовнику, оскільки їх використання оплачується окремо. При цьому провайдер має забезпечити надання гарантованих ресурсів іншим замовникам, тобто у якості додаткових ресурсів можуть використовуватись лише ті, які не були гарантовано надані жодному замовнику. Необхідно розробити моделі і методи розподілу ресурсів і навантаження хмарних ЦОД, що відповідають наведеним вище особливостям

хмарних IT-інфраструктур базуються на прийнятних для провайдерів критеріях і враховують ресурсні, технологічні та інші обмеження.

Опис модифікованого генетичного алгоритму для пошуку оптимальних рішень при розподілі ресурсів

Кожна ВМ може бути розгорнута на одному сервері. Для кодування генів перейдемо від матриці $n*m$ булевих змінних до вектору довжини m дискретних змінних, кожний елемент якого містить номер сервера $i=1,...,n$, на якому розміщена відповідна ВМ.

Наприклад:

$$x_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$y_j = [3 \quad 2 \quad 4 \quad 2 \quad 1 \quad 3 \quad 1 \quad 1]$$

Такий спосіб кодування генів дозволяє, по-перше, зменшити розмірність задачі, а по-друге, забезпечити автоматичне виконання обмежень щодо завантаження сервера. При цьому операція мутації буде відповідати переносу ВМ з одного сервера на інший, а операція кросовера буде відповідати обміну декількох ВМ між серверами.

Дуже важливим поняттям у генетичних алгоритмах є функція пристосованості (fitness function), яка інакше називається функцією оцінки. Вона являє міру пристосованості даної особини в популяції. В нашому випадку, функція пристосованості повинна приймати значення для найбільш рівномірного розподілу ВМ по серверам, коли всі основні й більшість додаткових вимог користувачів забезпечені ресурсами. В якості функції пристосованості будемо використовувати максимальна кількість нод, які забезпечує сервер. Якщо всі основні і додаткові вимоги до ресурсів забезпечені, то функція пристосованості буде дорівнювати кількості нод, що були вказані у заявках користувачів і були виділені під ВМ (інтервал (0, k)), та не перевищувати максимальну кількість, що може надати конкретний сервер .

Якщо ресурсами забезпечені всі основні і лише частина додаткових вимог користувачів, то функція пристосованості буде дорівнювати сумі кількості нод, що були фактично виділені під ВМ (інтервал (k, l)) користувачів на серверах.

Така функція пристосованості відображає реальну економічну ситуацію, коли провайдер збільшує свій дохід від надання якнайбільшої кількості нод на замовлення клієнта, і несе відповідальність у разі невиконання своїх обов'язків по наданню гарантованої кількості ресурсів.

Модифікований генетичний алгоритм (МГА) має ті ж параметри, що і традиційний, але відрізняється використанням деякий операторів. Наприклад, в МГА оператор мутації не застосовується до рішень щодо забезпечення ресурсів, отриманих в результаті операції кросовера, якщо вони мають

прийнятні для розподілу рішення; після кожної операції, в результаті якої отримані нові результати розподілу, проводиться оцінка результатів і оновлюється буфер збереження найліпших результатів(оптимальних рішень), якщо були отримані ліпші за існуючі результати. МГА показаний на Рис.1. Роботу МГА можна розділити на такі етапи:

1. випадковим чином формується деяка початкова множина рішень розподілу ресурсів(можливі варіанти розгортання ВМ на серверах);
2. далі виконуються послідовно операції кросовера та мутації, примочу мутація застосовується до рішень з гіршими параметрами; рішення з ліпшими показниками зберігається в буфері збереження оптимальних рішень;
3. наступним кроком є перевірка параметра МГА, який відповідає за кількість ітерацій – кількість поколінь: якщо він показує, що роботу МГА завершено, то проводиться остаточна оцінка рішень в буфері збереження оптимальних рішень, та вибирається рішення з ліпшими показниками, які відповідають критеріям, поставленим до забезпечення ресурсів з самого початку у вхідних даних, якщо ж ні, то робота МГА повертається до виконання операції кросовера над отриманою популяцією (множиною ріщень щодо розподілу ресурсів) від попереднього циклу.

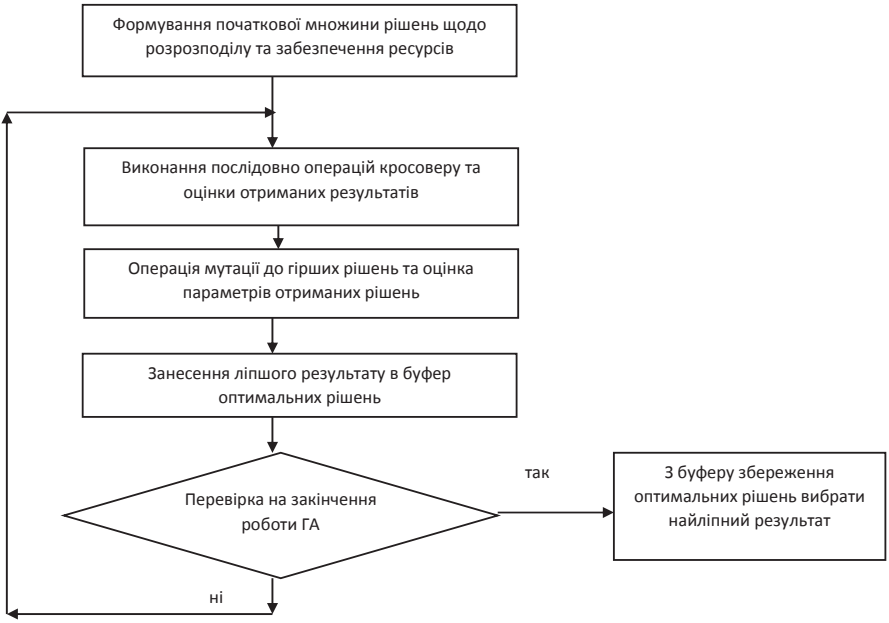

Рис.1. Модифікований генетичний алгоритм

Список основних параметрів МГА:

- потужність множини рішень ***Kp*** (чисельність популяції),
- кількість рішень, згенерованих на кожній ітерації,
- кількість батьківських пар ***Kf,***
- правило ***B*** вибору двох рішень,
- тип використовуваного оператора глобального пошуку (кросовера) ***C***,
- тип використовуваного оператора локальної зміни (мутації) ***M***,
- процедура добору ***S***.

Майже усі з них можуть динамічно змінюватися від ітерації до ітерації. Операція кросовера була вибрана одноточкова без фіксованої точки розриву. Операція мутації застосовувалась тільки до гірших результатів, мутації піддавався біт, вибраний випадково. Очевидно, що сім параметрів - це досить багато для алгоритму. То, наскільки вдалим виявиться застосування МГА при вирішенні поставленої задачі, у більшості випадків буде визначатися їхнім вдалим настроюванням.

Кінцева імплементація найоптимальнішого рішення розгортки віртуальних машин на серверах відбувається після роботи МГА та отримання за його допомогою цього рішення.

Висновки

Дослідження ефективності запропонованого алгоритму проводились наступним чином. Моделювалась робота кластера із 10 серверів, ресурси кожного з яких були поділені на 18 нод. МГА вирішував задачу розподілу ресурсів для випадків невеликих та середніх ВМ відносно розміру сервера. Було проведено 2 серії експериментів, які відрізнялись розкидом гарантованої кількості нод. Вимоги до віртуальних машин обирались відповідно з табл.1.

Таблиця 1. Вихідні дані проведення експерименту.

Серія експериментів 1 (невеликий розкид вимог користувачів)				Серія експериментів 2 (середній розкид вимог користувачів)			
n_{0min}	n_{0max}	n_{min}	n_{max}	n_{0min}	n_{0max}	n_{min}	n_{max}
1	3	1	3	1	2	1	3
2	3	2	7	1	3	1	5
2	4	3	7	1	4	1	6
2	6	3	8	1	5	1	7
4	6	3	9	1	6	1	8

Для ВМ, які мають незначні потреби у ресурсах відносно до розміру серверів, МГА дає хороші результати по використанню нод, близькі до максимально можливих (18 нод *10 серверів = 180 нод у кластері). При збільшенні вимог користувачів, що мають бути гарантовано забезпечені провайдером, більші ВМ стає складніше щільно розгорнути на серверах. Але ефективність МГА

залишається стабільною.

1. *Исаев Сергей.* Популярно о генетических алгоритмах. web: http://www.chat.ru/~saisa/index.html.
2. *Мельник А.О., Сокіл О.М.* Проектування оптимізованих структур комп'ютерних пристроїв підбором необхідних елементів бібліотеки з використанням генетичних алгоритмів // Вісник Нац. Ун-ту „Львівська політехніка". – 2004. - №523. – с.101-108.
3. *Теленик С.Ф., Ролик О.І., Букасов М.М., Лабунський А.Ю.* Моделі управління віртуальними машинами при серверній віртуалізації// Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. - К.: «ВЕК+», 2009. - № 51. - С. 147-152.
4. *Колодчак О.М., Мельник А.О.* Модифікований генетичний алгоритм для проектування цифрових пристроїв. // Комп'ютерні системи та мережі. Вісник НУ „Львівська політехніка". – №546. – Л.: 2005. – с.69-75.

Поступила 11.08.2014 р.

УДК 004.451.36:681.5

Сабат В. І., Українська академія друкарства, м. Львів

АНАЛІЗ РИЗИКІВ В АВТОМАТИЗОВАНИХ СИСТЕМАХ ДОКУМЕНТООБІГУ

Анотація. У статті проаналізовано метод визначення рівнів ризику для загроз в автоматизованій системі документообігу (АСДО) на всіх етапах життєвого циклу документів.

Ключові слова. Системи документообігу, ризики, загрози, вразливості.

Abstract. The article analyzes the method of determining the level of risk to threats in an automated workflow system (AWS) at all stages of life cycle of documents.

Keywords. Workflow system, risks, threats, vulnerabilities.

Вступ. Функціонування будь-якої системи документообігу зводиться до забезпечення зручних та комфортних умов роботи з документами для усіх суб'єктів системи від замовника чи адміністратора до виконавця. Такі умови роботи з документами можливі лише при налагодженій, надійній та ефективній системі захисту і доступу до будь-якого документу в АСДО.

На сьогодні більшість систем документообігу окрім електронних документів містять паперові, для яких також існує розроблена певна процедура їхнього функціонування в інформаційній системі. Але для того, щоб забезпечити надійність та ефективність роботи з великими масивами інформації, що міститься в документах, необхідно здійснити перехід від паперових до