

УДК 621.391

Б.М. Стрихалюк (канд. техн. наук, доцент),
О.М. Шпур (аспірант), М.О. Селюченко (аспірант)
Національний університет «Львівська політехніка», м.Львів
кафедра телекомунікації
E-mail: bogdan_str@ukr.net, o.shpur@mail.ru, egur2006@yandex.ua

ВИЗНАЧЕННЯ ДОСТУПНОСТІ ПРОГРАМНИХ КОМПЛЕКСІВ У СИСТЕМАХ З СЕРВІСНО-ОРІЄНТОВАНОЮ АРХІТЕКТУРОЮ

Доступність сервісів в сучасних системах обслуговування на основі cloud технології є одним з основних показників якості та надійності. В умовах постійних змін існуючих та впровадження нових методів надання сервісу та залежність доступності від багатьох факторів системи, робить її оцінку не простою задачею. Розділення систем керування та надання послуг, перехід до використання інформаційних технологій та методів створення послуг на основі програмного забезпечення призвели до виникнення систем обслуговування з динамічним середовищем, в якому поведінку та стан більшості компонентів чи процесів дуже тяжко передбачити. В результаті чого постає проблема оцінки доступності компонентів системи обслуговування. Класична теорія систем масового обслуговування для оцінки доступності комутаційних систем використовує методи, перевірені експериментально та на практиці. Для вирішення цієї проблеми ми пропонуємо використовувати метод групового пошуку ефективної доступності, що використовувався свого часу для оцінки доступності в класичних комутаційних системах масового обслуговування.

Ключові слова : *cloud, SOA, доступність.*

Вступ та аналіз існуючих досліджень

Сучасний стан розвитку інформаційних технологій спрямований на розроблення та впровадження нових підходів до організації обчислювального процесу [1].

Практично всі методи побудови сучасних систем та програм базуються на принципах сервісно-орієнтованої архітектури. Сервісно-орієнтована архітектура (SOA) в галузі інформаційних технологій визначає основні принципи побудови програм відповідно до специфіки конкретної задачі [2]. SOA базується на таких основних принципах:

1. Модульність компонентів – кожен компонент є незалежним від всіх інших компонентів і виконує одну визначену функцію. Компонент повинен мати відкритий інтерфейс, який описує синтаксис для виклику функції та результат, який ця функція повертає. Завдяки такому інтерфейсу, компоненти можуть взаємодіяти між собою.

2. В основі взаємодії компонентів лежить другий принцип SOA – обмін інформацією на основі технології комутації повідомлень, завдяки використанню якої будь-який компонент може взаємодіяти з будь-яким іншим компонентом системи використовуючи для звертання відкритий інтерфейс [3].

Будь-який процес можна розділити на логічну послідовність простих функцій, кожна з яких можна реалізувати як окремий програмний компонент. Взаємодіючи між собою за визначеною схемою, ці компоненти можуть реалізувати процес будь-якої складності (рис.1).

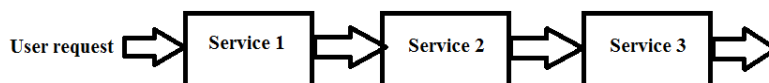


Рисунок 1 - Реалізація логічних процесів на основі SOA

Така архітектура програм, що може використовуватися для надання сервісів клієнтам, володіє значними перевагами у порівнянні з монолітною архітектурою програмного забезпечення.

На основі SOA можна реалізувати будь-яку послугу, яка може бути отримана як сервіс користувачами у віддаленому режимі. Для цього користувач надсилає запит, який поступає на обробку до програмного додатку, що реалізує потрібну послугу. Запит обробляється всіма компонентами послідовно і результат відправляється останнім компонентом назад до користувача. У випадку реалізації такої програми постає важливе питання - забезпечення високого рівня доступності всіх компонентів програми в умовах збільшення кількості користувачів з метою забезпечення умов договору SLA провайдером послуг. Підвищення інтенсивності вхідного навантаження може призвести до блокування того чи іншого компонента програми. Це пов'язано з тим, що кожен компонент виконує функції, які відрізняють за складністю і тривалістю обслуговування. В результаті компонент, який потребує більше часу для виконання обробки запиту або повинен очікувати відповідь від інших допоміжних компонентів системи, наприклад бази-даних, або може стати вузьким місцем цілого процесу обслуговування запиту.

Для уникнення вузьких місць в процесі обслуговування доцільно здійснювати реплікацію компонентів. З метою підвищення доступності, всі екземпляри компонента доцільно розташовувати на окремих фізично-незалежних серверах, оскільки відмова одного з серверів не вплине на доступність решти екземплярів компонента, які продовжать обслуговування запитів. В реальних системах доцільно здійснювати реплікацію всіх компонентів програми, що гарантує не тільки доступність але і високу якість обслуговування за рахунок можливості здійснення динамічного балансування навантаження (рис. 2).

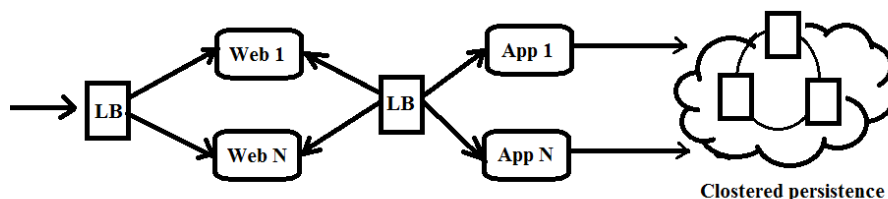


Рисунок 2 - Підвищення доступності програм завдяки реплікації її компонентів

Одним з таких підходів на основі сервісно-орієнтованої архітектури по реалізації реплікацій компонентів програм є cloud системи, що вважається наступним еволюційним кроком у розвитку розподілених обчислень. Метою такого підходу є підвищення ефективності використання розподілених ресурсів, об'єднання їх в єдину високо-продуктивну систему здатну вирішувати масштабні обчислювальні задачі. В cloud-системах використовуються як відомі технології так і нові ідеї та рішення.

Користувачам хмари надаються необхідні сервіси віддалено за допомогою технології віртуалізації. Користувач має доступ до власних даних, але не може управляти і не повинен піклуватися про інфраструктуру, операційну систему і власне програмне забезпечення, з яким він працює. Проте важливим аспектом у наданні cloud послуг є доступність до цих сервісів як у середині хмари так і до хмари загалом, наявність вільних каналів для їх надання та необхідної смуги пропускання для задоволення потреб користувачів.

Однак навіть при великому різноманітті моделей надання сервісів, доступність до компонентів сервісу все одно залишається пріоритетною задачею. Над проблемою доступності компонентів на різних рівнях cloud системи працюють багато вчених. Зокрема, дослідження доступності фізичних серверів в умовах динамічного розгортання віртуальних машин проводиться вченими Hamzeh Khazaei, Jelena Mišić, Vojislav B. Mišić та Nasim Beigi-Mohammadi [4]. Ще одна робота, авторами якої є Adam Grzech та Paweł Świątek, присвячена

дослідженню доступності складних програм на основі сервісно-орієнтованої архітектури. В цій роботі проведено дослідження структури складних програм та способу її оптимізації для підвищення доступності [5].

Постановка проблеми та її вирішення

Сьогодні основним та найбільш оптимальним середовищем для реалізації програм на основі сервісно-орієнтованої архітектури є Cloud-система. В основі таких систем лежить технологія віртуалізації. Перевагами її використання є можливість динамічного розгортання віртуальних машин на серверах в залежності від потреб користувачів та можливість динамічної міграції віртуальних машин з одного сервера на інший..

На кожній VM можна виділити певну кількість програмних компонентів. Кожен програмний додаток споживає однакову кількість апаратних ресурсів для обслуговування запитів. Кожен програмний додаток може обслуговувати запити від користувачів певної інтенсивності. Проте інтенсивність надходження запитів в певний момент часу може збільшитися. Це призведе до зменшення продуктивності програмного компоненту, до якого цей запит був надісланий та появи блокувань на обслуговування таких запитів.

Для уникнення перевантаження компонентів, а саме зменшення ймовірності блокування, ми пропонуємо оцінювати доступність сервісних програм, що побудовані на основі сервісно-орієнтованої архітектури та виконуються в середовищі Cloud-системи, за допомогою методу визначення ефективної доступності в режимі групового пошуку. Для цього ми вводимо такі умови та обмеження.

- кількість функціонально-незалежних, незамінних компонентів програми становить j ;
- у випадку обслуговування запиту компонент буде недоступним для всіх інших запитів, тобто компонент перейде до обслуговування наступного запиту тільки після закінчення повної обробки поточного запиту;
- на одній віртуальній машині можна встановити тільки один компонент чи його екземпляр, незалежно від функціональності, тому для розгортання окремого компоненту використовується нова віртуальна машина;
- на всіх фізичних серверах можуть одночасно виконуватися не більше ніж N віртуальних машин одночасно;
- кількість компонентів Z для всіх функцій програми є однаковою.

В результаті цього можна побудувати схему обслуговування запиту, доступність якої можна оцінити за допомогою класичної моделі багатоланкових систем типу час-простір-...-простір-час. Така схема відображена на рис.3.

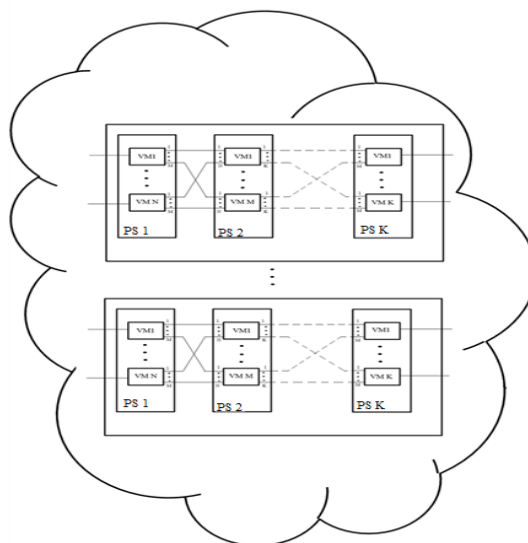


Рисунок 3 - Модель багатоланкової cloud системи

Для оцінки доступності пропонуємо використовувати метод ефективної доступності в режимі групового пошуку (метод «КЛІГС») [6].

Метод КЛІГС отримав скорочену назву від англійських слів, що означають «розрахунок багатоланкових систем групового шукання». Метод ґрунтується на поняттях середньої доступності d_{sr} , середньої недоступності $\overline{d_{sr}}$, максимальної доступності d_{max} комутаторів останньої ланки від будь-якого вільного входу першої ланки. За допомогою цих понять визначається ефективна доступність d_{ef} , значення якої дає змогу обчислити ймовірність втрат, користуючись модифікованою формулою Пальма-Якобеуса. Формула Пальма-Якобеуса описує зайняття з'єднувальних ліній в неповнодоступному пучку, який можна описати за допомогою розподілу Ерланга, отриманого ним для зайняття будь-яких D ліній в повнодоступному пучку.

Даний метод визначення доступності системи може бути адаптований для таких систем як cloud. Припустимо, що наша cloud система має 5 фізичних серверів (ланок), які можуть одночасно містити і обслуговувати не більше ніж 4 віртуальних машини кожен. Відповідно до розробленої схеми (рис.4) обслуговування запиту на одному фізичному сервері можуть бути встановлені віртуальні машини, на яких виконуються компоненти тільки одного типу. Такі сервери утворюють ланку. Запит в процесі обслуговування проходить по черзі через усі ланки. Після обробки запиту k -м компонентом у ланці i , запит відправляється до будь-якого вільного компоненту у ланці $i+1$. При цьому приймається, що протягом часу від моменту надходження запиту на окремий компонент першої ланки до моменту закінчення обслуговування запиту компонентом останньої ланки, всі компоненти, через які проходить запит не можуть здійснювати обробку інших запитів відповідно до типу компоненту, який виконує відповідну обробку запиту. Процес обслуговування запиту послідовністю сервісів в усіх ланках також відображена на рис.4.

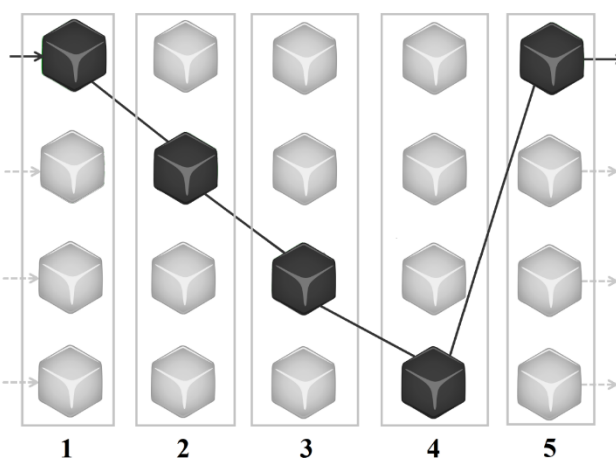


Рисунок 4 - Процес обслуговування запиту послідовністю сервісів

Використання методу для cloud систем передбачає введення таких позначень:

m_i – кількість логічних зв'язків віртуальної машини i -тої ланки з віртуальною машиною ланки $i+1$ (для нашого випадку $m_i = 4$);

k_i – кількість віртуальних машин i -го сервера (ланки);

s – кількість ланок (фізичних серверів) cloud системи;

q_r – кількість логічних виходів з віртуальних машин останньої ланки в напрямку до користувача (для нашого випадку $q_r = 4$) ;

v_r – кількість зв'язків в напрямку між VM та користувачем (для нашого випадку $v_r = 1$);

M – загальна кількість виходів (для даної системи $M=4$);

$v_{i,i+1}$ – загальна кількість проміжних ліній між сусідніми ланками;

Основними поняттями, якими будемо оперуватися для визначення доступності до програмних компонентів будуть поняття середньої доступності до додатків d_{sr} , середньої недоступності, максимальної доступності віртуальних компонентів системи від кількості запитів на програмний додаток, що надходять від користувачів d_{max} та ймовірність втрат запитів.

Отже, значення середньої доступності до додатків визначатиметься:

$$d_{sr_k} = \left[\frac{m_i - y_k}{m_i} \right]^{s-1},$$

де y_k – обслужене навантаження компонентом k -тої ланки. Для даної системи обслужене навантаження одним компонентом ланки не може перевищувати 1 Ерл. Інакше це буде свідчити про те, що будуть одночасно зайняті всі виходи компонента ланки, що обслуговує це навантаження. Враховуючи, що для обслуговування запиту резервується послідовність компонентів в кожній ланці, то звідси випливає, що така послідовність буде постійно зайнята обслуговуванням запитів при максимальній інтенсивності навантаження, яку може обслужити така послідовність. Прийmemo, що інтенсивність поступлення запитів на один компонент рівна 100 запитів/год та еквівалентна навантаженню в один Ерланг

Максимальна доступність визначатиметься виразом

$$d_{max} = \prod_{i=1}^{s-1} \frac{m_i}{m_{i+1}},$$

і дорівнює добутку частки логічних зв'язків віртуальних машин i -тої ланки з віртуальною машиною ланки $i+1$ до логічних зв'язків віртуальних машин $i+1$ -ої ланки ланки з віртуальною машиною ланки $i+2$. Оскільки на кожному фізичному сервері виконується однакова кількість компонентних додатків, то кількість логічних зав'язків між i -тою та $i+1$ -ю ланкою буде рівною кількості логічних зав'язків між $i+1$ -ю та $i+n$ -ю ланкою, тобто $d_{max} = 1$.

Середня недоступність рівна:

$$\bar{d}_{sr} = d_{max} - d_{sr},$$

і відповідає середній кількості недоступних VM, що дорівнює різниці між кількістю VM, доступних при відсутності запитів і заданій кількості обслужених запитів.

У цьому методі ефективна доступність визначається як сума двох доданків

$$d_e = d_{1e} + d_{2e},$$

де $d_{1e} = d_{sr} q_r$ і є середньою кількістю логічних виходів з віртуальних машин останньої ланки в напрямку до користувача, які можна займати, і створювати середню доступність. Другий доданок ефективної доступності задається виразом

$$d_{2e} = \bar{d}_{sr} q_r \frac{y_{0r} v_{s-1,s} - y_0}{v_r M},$$

де y_{0r} – обслужена кількість запитів; y_0 – загальна кількість запитів.

Після формування всіх необхідних умов, за допомогою аналітичних і статистичних методів розрахунку було отримано наступні результати :

Таблиця 1

Результати проведених розрахунків

Обслужене навантаження компонентами i -тої ланки, зап/год	Середня доступність, %	Середня недоступність, %	Ефективна доступність
1	2	3	4
100	0.316	0.684	9.469
120	0.24	0.76	10.079

Продовження табл. 1

1	2	3	4
140	0.179	0.821	10.572
160	0.13	0.87	10.963
180	0.092	0.908	11.268
200	0.063	0.938	11.5
220	0.041	0.959	11.672
240	0.026	0.974	11.795
260	0.015	0.985	11.88
280	8.1e-3	0.992	11.935
300	3.906e-3	0.996	11.969
320	1.6e-3	0.998	11.987
340	5.063e-4	0.999	11.996
360	10e-5	1	11.999
380	6.25e-6	1	12

Провівши аналіз отриманих результатів можна сказати, що:

1. Доступність cloud системи напряду залежатиме від кількості реплікацій компонентних додатків одного типу;
2. Обслужене навантаження компонентами кожної ланки залежить від кількості запитів, які надсилатимуть користувачі, та відповідно, і від кількості користувачів cloud системи;
3. Ефективна доступність, яка характеризує наявність вільних логічних зв'язків між компонентами ланок, буде напряду залежати від кількості обслужених заявок та кількості проміжних логічних зв'язків між сусідніми ланками.

На основі результатів отримана залежність між середньою доступністю системи та обслуженим нею навантаженням (рис. 5).

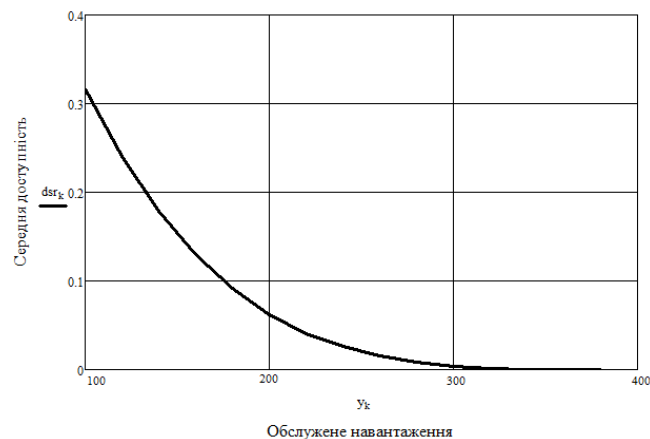


Рисунок 5 - Залежність середньої доступності системи та обслуженим навантаженням

Дана залежність свідчить про те, що чим більша інтенсивність поступлення запитів на обслуговування, тим менша середня доступність кожної компоненти, а як результат і системи в цілому. Однак така доступність компонентів залежить і від кількості ланок системи (фізичних серверів) (рис.6).

Чим більша кількість ланок системи і чим більша інтенсивність поступлення запитів на обслуговування, тим менша середня доступність кожної компоненти в кожній ланці.

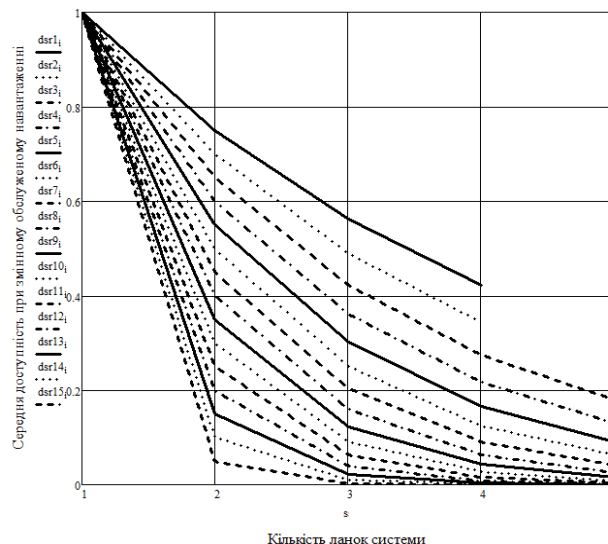


Рисунок 6 - Залежність значень середньої доступності від кількості ланок системи при змінному навантаженні

Що стосується ефективної доступності, то спочатку слід добре розібратися, що таке є доступність сервісних програм, що побудовані на основі сервісно-орієнтованої архітектури та виконуються в середовищі Cloud-системи. Як свідчить класичне визначення цього поняття доступність це властивість інформаційного ресурсу, яка полягає в тому, що користувач та/або процес, який володіє відповідними повноваженнями, може використовувати цей ресурс відповідно до правил, встановлених політикою безпеки не очікуючи довше заданого (прийняттого) інтервалу часу. Суть цієї властивості полягає в тому, що потрібний інформаційний ресурс знаходиться у вигляді, необхідному користувачеві, в місці, необхідному користувачеві, і в той час, коли він йому необхідний. Отож з точки зору систем з сервісно-орієнтованою архітектурою доступність це наявність вільного з'єднувального віртуального каналу, який здатний забезпечити доступ до необхідних ресурсів користувачам в будь який момент часу не очікуючи довше визначеного інтервалу часу. Ресурс повинен надаватися без будь яких затримок та перебоїв.

Тоді ефективною доступністю системи є наявність вільних логічних виходів з віртуальних машин довільної ланки в напрямку до наступної ланки або користувача, які можна займати, і створювати середню доступність при обслуговуванні певної кількості запитів. Цей параметр напряму залежить від інтенсивності поступлення запитів у систему (рис.7).

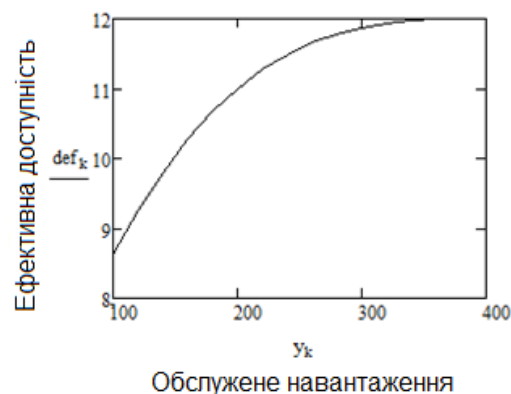


Рисунок 7 - Залежність значення ефективної доступності та обслугованим навантаженням

Як видно із залежності на рис.8 при збільшенні обслуженого навантаження компонентами i -тої ланки системою виділяється більша кількість ліній для обслуговування надісланих запитів користувачів без втрати якості надання ресурсів.

Однак доступність системи залежить не тільки від кількості запитів на ресурси системи, які надходять від користувачів, а й від кількості віртуальних та фізичних компонентів системи. У нашому випадку це кількість фізичних серверів (ланок) cloud системи та кількість віртуальних машин i -го сервера (ланки). Саме від цих параметрів залежить кількість вхідних і вихідних логічних каналів системи та кількість логічних зв'язків між ланками, що доступні користувачам для отримання того чи іншого ресурсу системи. Зміна кількості фізичних серверів дасть змогу збільшити не тільки кількість програм, які доступні користувачам через cloud систему, а й швидкість обробки компонентів та надання необхідного ресурсу користувачам. Більша кількість віртуальних машин збільшить кількість проміжних з'єднувальних ліній між ланками системи, що дасть змогу пришвидшити процес обробки запитів. На рис. 8 приведена залежність ефективної доступності системи від кількості ланок.

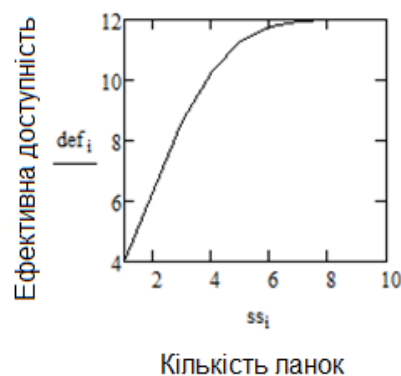


Рисунок 8 - Залежність ефективної доступності від кількості фізичних серверів (ланок) системи

Чим більша кількість ланок cloud системи тим ефективна доступність при заданій інтенсивності надходження запитів більша, тобто для обслуговування системою виділяється більша кількість логічних каналів.

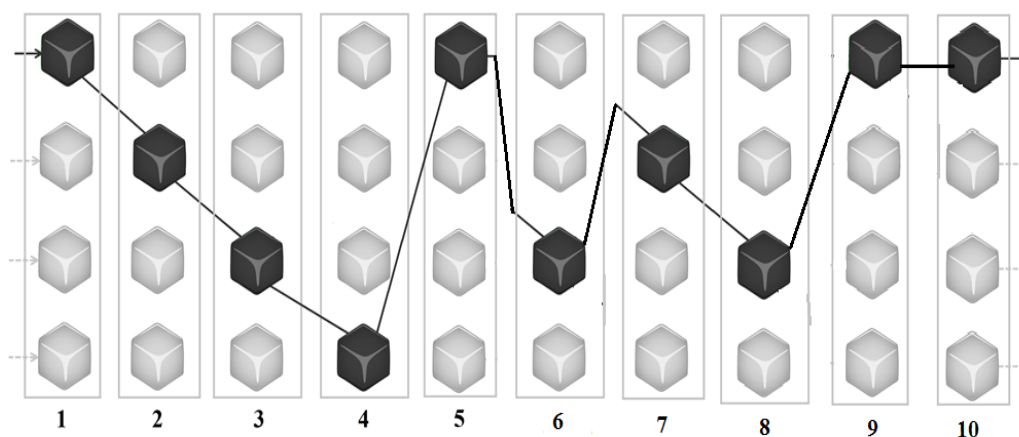


Рисунок 9 - Процес обслуговування запитів більшою кількістю серверів

Висновок

Практично всі методи побудови сучасних систем та програм базуються на принципах сервісно-орієнтованої архітектури. Сьогодні основним та найбільш оптимальним

середовищем для реалізації програм на основі сервісно-орієнтованої архітектури є Cloud-система. Для уникнення перевантаження компонентів, а саме зменшення ймовірності блокування, у роботі запропоновано оцінювати доступність сервісних програм, що побудовані на основі сервісно-орієнтованої архітектури та виконуються в середовищі Cloud-системи, за допомогою методу визначення ефективної доступності в режимі групового пошуку на основі класичної моделі багатоланкових систем типу час-простір-...-простір-час.

Для оцінки доступності запропоновано використовувати метод ефективної доступності в режимі групового пошуку (метод «КЛІГС»).

Провівши аналіз побудови складних програм на основі SOA можна сказати, що :

1. Доступність cloud системи напряду залежить від кількості реплікацій компонентних додатків одного типу;

2. Обслужене навантаження компонентами залежати від кількості запитів, які надсилатимуть користувачі, та відповідно, і від кількості користувачів cloud системи;

3. Ефективна доступність, яка характеризує наявність вільних логічних зв'язків між компонентами ланок, буде напряду залежати від кількості обслужених заявок та кількості проміжних логічних зв'язків між сусідніми ланками.

На основі отриманих даних в результаті застосування методу КЛІГС, для визначення доступності складних програм, отримана залежність між середньою доступністю системи та обслуženим нею навантаженням. Дана залежність свідчить про те, що чим більша інтенсивність поступлення запитів на обслуговування, тим менша середня доступність кожної компоненти, а як результат і системи в цілому. Однак така доступність компонентів залежить і від кількості ланок системи (фізичних серверів). Чим більша кількість ланок системи і чим більша інтенсивність поступлення запитів на обслуговування, тим менша середня доступність кожної компоненти в кожній ланці.

Ефективною доступністю системи є наявність вільних логічних виходів з віртуальних машин довільної ланки в напрямку до наступної ланки або користувача, які можна займати, і створювати середню доступність при обслуговуванні певної кількості запитів. Цей параметр напряду залежить від інтенсивності поступлення запитів у систему. При збільшенні обслуженого навантаження компонентами і-тої ланки системою виділяється більша кількість ліній для обслуговування надісланих запитів користувачів без втрати якості надання ресурсів. Чим більша кількість ланок cloud системи тим ефективна доступність при заданій інтенсивності надходження запитів більша, тобто для обслуговування системою виділяється більша кількість логічних каналів.

Список використаної літератури

1. N. Kryvinska Next Generation Networks - Service Delivery and Management / N. Kryvinska, C. Strauss; Electronic Business series, vol. 7, International Academic Publishers, Peter Lang Publishing Group, 2011; ISBN-978-3-631-60871-5; ISSN: 1868-646X.
2. N. Kryvinska Converged Network Service Architecture: A Platform for Integrated Services Delivery and Interworking. Electronic Business series edited by C. Strauss, vol. 2, International Academic Publishers, Peter Lang Publishing Group, 2010; ISBN-13:978-3631595251; ISSN:1868-646X.
3. N. Kryvinska Integrated management platform for seamless services provisioning in converged network/ N. Kryvinska, Do van Thanh, C. Strauss, Inderscience Publishers, International Journal of Information Technology, Communications and Convergence (IJITCC), ISSN (Online): 2042-3225, ISSN (Print): 2042-3217, Vol. 1, No. 1, 2010, pp. 77-91.
4. Khazaei, H. Performance of Cloud Centers with High Degree of Virtualization under Batch Task Arrivals/ Khazaei, H., Misic, J., Misic, V; Parallel and Distributed Systems, IEEE Transactions on, On page(s): 2429 - 2438 Volume: 24, Issue: 12, Dec. 2013 Grzech, A.,

5. Swiatek, P. Complex Services Availability in Service Oriented Systems / Inst. of Comput. Sci., Wroclaw Univ. of Technol., Wroclaw, Poland / Systems Engineering (ICSEng), 2011 21st International Conference on Date 16-18 Aug. 2011: p. 227 – 232.
6. Б.С.Лившиц, А.П.Пшеничников, А.Д.Харкевич Теория телетрафика – М.: Связь, 1979, - с. 119.

References

1. Kryvinska, N., Strauss, C. (2011), *Next Generation Networks - Service Delivery and Management*, Book, Electronic Business series, vol. 7, International Academic Publishers, Peter Lang Publishing Group, ISBN-978-3-631-60871-5; ISSN: 1868-646X.
2. Kryvinska, N. (2010), *Converged Network Service Architecture: A Platform for Integrated Services Delivery and Interworking*. Electronic Business series edited by C. Strauss, vol. 2, International Academic Publishers, Peter Lang Publishing Group; ISBN-13:978-3631595251; ISSN:1868-646X.
3. Kryvinska, N., Do van Thanh, Strauss, C. (2010), "Integrated management platform for seamless services provisioning in converged network", *Inderscience Publishers, International Journal of Information Technology, Communications and Convergence (IJITCC)*, ISSN (Online): 2042-3225, ISSN (Print): 2042-3217, Vol. 1, No. 1, 2010, pp. 77-91.
4. Khazaei, H., Mistic, J., Mistic, V. (2013), "Performance of Cloud Centers with High Degree of Virtualization under Batch Task Arrivals", *Parallel and Distributed Systems*, IEEE Transactions on, Volume: 24, pp. 2429 – 2438.
5. Swiatek, P. (2011) "Complex Services Availability in Service Oriented Systems, *Proc. 21st International Conference on Date 16-18 Aug. 2011*, pp. 227 – 232.
6. Lyvshyts, B.S., Pshenychnykov, A.P., Kharkevych A.D. (1979), *Teoriya teletrafyka*, Svyaz', Moskva, Russia, pp. 119.

Надійшла до редакції:
25.04.2014 р.

Рецензент:
докт. техн. наук, доц. Пелішок В.О.

Б.М. Стрыхалюк, О.Н. Шпур, М.А. Селюченко

Национальный университет «Львовская политехника»

Определение доступности программных комплексов в системах с сервисно-ориентированной архитектурой. Доступность сервисов в современных системах обслуживания на основе cloud технологии является одним из основных показателей качества и надежности. В условиях постоянных изменений существующих и внедрение новых методов предоставления сервиса и зависимость доступности от многих факторов системы, делает ее оценку не простой задачей. Разделение систем управления и предоставления услуг, переход к использованию информационных технологий и методов создания услуг на основе программного обеспечения привели к возникновению систем обслуживания с динамической средой, в которой поведение и состояние большинства компонентов или процессов очень тяжело предсказать. В результате чего возникает проблема оценки доступности компонентов системы обслуживания. Классическая теория систем массового обслуживания для оценки доступности коммутационных систем использует методы, проверенные экспериментально и на практике. Для решения этой проблемы мы предлагаем использовать метод группового поиска эффективной доступности, использовавшийся в свое время для оценки доступности в классических коммутационных системах массового обслуживания.

Ключевые слова: cloud, SOA, доступность

B.M. Strykhalyuk, O.M. Shpur, M.O. Seliuchenko
Lviv Polytechnic National University

Detecting the availability of software complexes in systems with service-oriented architecture.

The availability of services in the current service systems based on cloud technology is one of the main indicators of quality and reliability. Under conditions of constant change of existing systems and introducing new methods of service provisioning, system accessibility assessment which depends on many factors is not an easy task to solve. Separation of management systems and services, the transition to use of information technologies and methods for creating service-based software systems have led to the emergence of dynamic service environment in which the behavior and state of most components or processes are very difficult to predict. As a result, there is a problem of assessing service components accessibility. The classical theory of queuing systems for assessing the availability of switching systems used methods tested experimentally and in practice. To solve this problem, we propose to use an effective method of finding availability group, that has been used once to assess the accessibility of classical switching systems. Effective availability means presence of virtual logical outputs from virtual machines of any link in the direction to the next link or the user, which can be occupied and used to create an average availability while servicing certain amount of requests. This parameter directly depends on incoming requests intensity in the system. When processed load by i-link is being increased, bigger amount of links is provided for processing incoming requests from users without quality of service losses. The greater number of links in cloud system the more efficient is availability by determined incoming requests intensity that is more logical channels are provided for processing.

Keywords: cloud, SOA, availability.



Стрихалюк Богдан Михайлович, канд. техн. наук, доцент кафедри «Телекомунікації» Національного університету «Львівська політехніка» (м. Львів, вул. С. Бандери, 12). Наукові інтереси: теоретичні основи побудови та функціонування мереж наступного покоління NGN та cloud-технологій.



Шпур Ольга Миколаївна, аспірантка кафедри «Телекомунікації» Національного університету «Львівська політехніка» (м. Львів, вул. С. Бандери, 12). Наукові інтереси: особливості побудови та функціонування мереж на основі сервісно-орієнтованої архітектури, mesh- та cloud-технологій.



Селюченко Мар'ян Олександрович, аспірант кафедри «Телекомунікації» Національного університету «Львівська політехніка» (м. Львів, вул. С. Бандери, 12). Наукові інтереси: особливості побудови та функціонування мереж на основі сервісно-орієнтованої архітектури, формування композитних додатків у мережах з технологією cloud.