Технології надання доступу до сервісів розподіленої хмарної системи

Сівак А.С.
Науковий керівник: к.т.н., доц. Муляр І.В.
Хмельницький національний університет

На сучасному етапі розвитку української держави актуальною є проблема надання доступу до сервісів розподіленої хмарної системи [1]. Актуальність дослідження зумовлена необхідністю підвищення доступності хмарних сервісів на базі клієнт-серверної й одноранговой розподіленої хмарних архітектур. У контексті даного дослідження необхідним є пошук алгоритмів і засобів використання хмарних систем і мережних додатків, провайдерами хмарних послуг, а також підприємствами, які розгортають власні хмарні інформаційні сервіси.

Так принцип розподілу ресурсів між вузлами хмарної системи з децентралізованою структурою з метою підвищення оперативності відповіді на запит покладено в основу технології надання доступу до сервісів одноранговой розподіленої хмарної системи.

Варто зазначити, що в локальній мережі або в мережі Інтернет розподіл реалізується за допомогою протоколу Kademlia [2] і містить процеси публікації, реплікації і надання доступу до ресурсів. Інформація, яка зберігається на дисковому просторі робочої станції, у вигляді відповіді на запит розподіленої хмарної системи, надається в спільне користування іншим вузлам-учасникам. Шляхом модифікації ресурсних записів процес розподілення ресурсів координується за допомогою DNS-сервера. На робочих станціях всіх вузлів-учасників одногранної розподіленої хмарної системи встановлено спеціалізоване програмне забезпечення, яке організовує взаємодію вузлів. З урахуванням завантаженості каналів зв'язку, доступ до ресурсів для вузлів, які не беруть участі в процесі, надається за допомогою їх

Розглянемо взаємодію основних компонентів для отримання доступу до веб-ресурсу http://domain-name.com/resource-name. Перш за все вузол-учасник виконує стандартний запит до ресурсу http://domain-name.com/resource-name через веб-браузер на своїй робочій станції, далі через прикладне ПЗ виконується запит до додаткового ресурсного запису DNS-сервера ідентифікатора ресурсу ID_res та до ідентифікаторів базових вузлів реплікації. Це необхідно для запуску стандартного процесу пошуку DHT-lookup «FIND_NODE» найближчого за XOR-метрикою вузла реплікації ресурсу з ідентифікатором ID_res на наступному етапі. Наступним кроком буде виконання запиту «FIND_VALUE» на надання доступу до ресурсу до вузла з відповідним ідентифікатором. На прикінцевому етапі через веб-браузер і збереження відповіді на запит відбувається завантаження контенту для запитувача вузла на дисковому просторі цього вузла.

Розглянемо основні типи ідентифікаторів: ідентифікатор вузла (ID_node), глобальний ідентифікатор ресурсу (доменне ім'я) й унікальний

ідентифікатор ресурсу (ID_res) в одноранговій мережі.

Незалежно від ролі вузла, при додаванні кожного нового в систему, для нього генерується унікальний ідентифікатор вузла ID_node і дві пари публічних і приватних (відкритих і закритих) ключів: Kpub1, Kpriv1, Kpub2, Kpriv2 на основі алгоритму RSA-256 або RSA-128. Перша ключова пара використовується для стандартних процесів комунікації, друга – для процесу валідації ідентифікатора вузла з метою захисту від підміни особистості.

Ідентифікатор вузла генерується у такий спосіб [3]:

$$ID\_node = Hash\ (Kpub1 + Sig_{Kpriv2(Kpub1)})$$

де *ID_node* – ідентифікатор вузла; *Kpub1* – відкритий ключ 1; *Kpriv2* – закритий ключ 2; *Hash (Data)* – оператор обчислення хеш-функції даних Data; $Sig_{Kpriv2(Kpub1)}$- цифровий підпис даних *Kpub1*,який отримано за допомогою ключа *Kpriv2*.

Унікальний ідентифікатор ресурсу – хеш-функція його вмісту, що обчислюється за допомогою SHA (2)-256:

$$ID\_res = Hash\ (Content\ (res)),$$

де *ID_res* – ідентифікатор ресурсу; *Hash (Data)* – оператор обчислення хеш-функції даних Data; *Content (res)* – вміст ресурсу res.

Публікація ресурсу може здійснюватися тільки власником цього ресурсу. Даний процес полягає у відкритті доступу до інформації, що публікується, додаванні DNS-запису відповідності адреси ресурсу і доменного імені. Після отримання доступу безпосередньо до ресурсу або його частини, інформація зберігається на дисковому просторі вузла-учасника у вигляді відповіді на запит, і такий вузол може надати цей ресурс за запитом іншим вузлам мережі за допомогою протоколу Kademlia, тобто виступити в ролі сервера. Для цього його ідентифікатор додається в глобальну розподілену хеш-таблицю (DHT) ідентифікаторів ресурсів [2].

У разі, якщо інший вузол-учасник однорангової розподіленої хмарної системи запитує той самий ресурс, він отримує доступ до нього за ідентифікатором найближчого за XOR-метрикою [2] вузла реплікації, що володіє ресурсом частково або повністю, , що містяться в ресурсних записах DNS для даного ресурсу. Якщо, на дисковому просторі вузол-учасник запитує всі частини ресурсу його робочої станції, то буде закешовано весь ресурс повністю. Тоді цей вузол буде мати повну репліку ресурсу, а його адреса буде додана до основного ресурсного запису (A-запису) DNS-сервера. Отже, значна кількість вузлів-учасників однорангової розподіленої хмарної системи, зі збільшенням популярності ресурсу, може ним поділитися. Для виконання серверних функцій кожен вузол-учасник має надати в спільне користування частину власних апаратних ресурсів – дискового простору й потужності центрального процесора. Якщо вузли не є учасниками системи, вони можуть отримати доступ до ресурсу завдяки способу взаємодії з сервером власника ресурсу або вузлами, що володіють повною реплікою

ресурсу, за їхніми адресами з урахуванням завантаженості каналів зв'язку і станції власника ресурсу або вузлів.

Перевірка цілісності ресурсу та ідентичність реплік проводиться шляхом обчислення хеш-функції отриманого ресурсу і зіставлення зі значенням ідентифікатора цього ресурсу. Крім того, на проміжному етапі перед завантаженням контенту виробляється процес валідації вузла, що надає ресурс. Вузол, що надає ресурс, для підтвердження свого ідентифікатора відправляє повідомлення такого формату [2]:

ID_node	
Kpub1	SigKpriv2(Kpub1)
Kpub2	SigKpriv2(Kpub2)

Data

На першому кроці валідації ідентифікатора вузла виконується перевірка автентичності переданих відкритих ключів Kpub1 і Kpub2 за рахунок механізму цифрового підпису. Якщо перевірка проходить успішно, то далі перевіряється істинність співвідношення [3]:

$$ID\_node = Hash\ (Kpub1 + Sig_{Kpriv2}(Kpub1))$$

де *ID_node* – ідентифікатор вузла, переданий в повідомленні; *Kpub1* – відкритий ключ 1; *Hash (Data)* – оператор обчислення хеш-функції даних *Data*; *SigKpriv2(Kpub1)* – цифровий підпис даних *Kpub1*, отриманий за допомогою ключа *Kpriv2*.

Якщо рівність виконується, то збір інформації проходить успішно і вузол-відправник вважається успішно ідентифікованим і підтвердженим.

Незалежно від ролі, будь-який вузол-учасник однорангової розподіленої хмарної системи отримує високошвидкісний доступ до всіх ресурсів системи, завдяки можливості отримати ресурс від найближчого вузла-учасника, який має копію цього ресурсу. Описані процеси додавання нового вузла, збору інформації, публікації ресурсу й отримання доступу до ресурсу подано у вигляді поетапної послідовності дій з надання доступу до сервісів розподіленої хмарної системи за допомогою інформаційних потоків, взаємодії процесів оброблення інформації та об'єктів, які є частиною цих процесів.

Таким чином, архітектура реалізує новий підхід до розподілу ресурсів в хмарної мережі з децентралізованою структурою, об'єднуючи переваги технологій GRID, хмарних обчислень і пірінгових мереж. Її особливістю є самоорганізація процесу реплікації ресурсів засобами робочих станцій вузлів-учасників системи: даний процес не вимагає втручання адміністратора або сторонніх механізмів.

Література

1. Технологии Web, Grid, Cloud для гарантоспособных ИТ-инфраструктур[Текст] : монография / В. С. Харченко и др; Харьков. нац.

аэрокосм. ун-т им. Н. Е. Жуковского «ХАИ», 2013. 868 с.

2. Zhou, S. Liu, and G. Huang, Kad-D: An Improved Model Based on Kademlia, Multimedia Information Networking and Security (MINES), 2011 Third International Conference on, 2011. P.123-127.

3. Основы зеленой ИТ-инженерии. Моделирование облачных систем. Практикум / Харченко В.С., Дрозд А.В., Поночовный Ю.Л., Яновская О.В., Яновский М.Э., Кривцов А.Ю., Иванченко О.В. / Под ред. Харченко В.С. Министерство образования и науки Украины, Нац. аэрокосмический ун-т им. Н.Е. Жуковского «ХАИ». 2016. 168 с.

Аналіз мережевого трафіку за допомогою сніфер-програм

Сташков Д.В.
Науковий керівник – к.т.н. доц. Бойчук В.О.
Хмельницький національний університет

Аналізатор трафіку, сніфер – це програма, яка перехоплює мережевий трафік який проходить через мережеву карту. Програма призначена для діагностики мережі, завдяки чому вона часто використовується системними адміністраторами. З іншого боку, сніфери також використовуються для несанкціонованого доступу до даних і перехоплення паролів. Ці системні програмні засоби працюють на рівні мережевого адаптера NIC (Network Interface Card).

Перехоплення трафіку може здійснюватися:

– звичайним «прослуховуванням» мережевого інтерфейсу (метод ефективний при використанні в сегменті концентраторів (хабів) замість комутаторів (світчів), інакше метод малоефективний, оскільки на сніфер потрапляють лише окремі фрейми);

– підключенням сніфера в розрив каналу;

– відгалуженням (програмним або апаратним) трафіку і спрямуванням його копії на сніфер;

– через атаку на канальному (MAC-spoofing) або мережевому рівні (IP-spoofing), що приводить до перенаправлення трафіку або всього трафіку сегменту на сніфер з подальшим поверненням трафіку в належну адресу.

За замовчуванням мережева плата комп'ютера бачить тільки то, що призначене саме для неї. Однак сніфери встановлюють її в режим прийому всіх пакетів (promiscuous mode) та змушують мережеву плату приймати всі пакети, незалежно від того, кому вони адресовані. Перехоплений трафік передається декодеру пакетів, який ідентифікує та розкладає пакети за відповідними рівнями. Залежно від можливостей конкретного сніферу, перехоплена інформація про пакети може пізніше аналізуватися та фільтруватися.