MEANS OF TELECOMMUNICATIONS
ЗАСОБИ ТЕЛЕКОМУНІКАЦІЙ

УДК 621.396.004 **DOI:10.30837/rt.2023.1.212.18**

Л.О. ТОКАР, канд. техн. наук, О.А. КОЛТАКОВ, В.Є. ЦИЛЮРИК

СТВОРЕННЯ ТЕСТОВОГО СТЕНДУ CALL-ЦЕНТРУ ДЛЯ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ ASTERISK У КЛАСТЕРІ

Вступ

У зв’язку з зростанням попиту на телекомунікаційні послуги сервери голосових викликів та викликів даних повинні забезпечувати більшу пропускну здатність обробки викликів. Інтернет та мобільні пристрої сприяли збільшенню потреб в комутаційних потужностях телекомунікаційних серверів. Через вимогливі додатки можливості існуючих серверів обробки викликів доведено до краю.

Виходячи з цього, одним із рішень є кластеризація серверів викликів. Кластер серверів викликів є групою з декількох окремих серверів викликів, що охоплюють певну географічну область. Щоб подолати зростаючий попит на пропускну здатність, необхідні дослідження параметрів продуктивності кластерів серверів викликів для отримання оцінки про якість та збалансованість роботи центру обробки викликів. Кластеризація серверів у call-центрі забезпечує переваги як масштабованості, так і надійності. Тема роботи є актуальною, що обумовлено необхідністю ефективної обробки викликів серверами у кластері для забезпечення потрібної відмовостійкості та мінімізації експлуатаційних витрат для телефонних компаній.

Call-центр являє собою центр обробки телефонних викликів, принцип побудови якого ґрунтується на маршрутизації викликів агентів за певними правилами, які розробляються в компанії та дозволяють якісно та ефективно обслуговувати клієнтів, а також підтримувати черги дзвінків. Він здатний не тільки приймати та обробляти запити, але й використовувати для контактів з клієнтами звичайну пошту, факсимільний та мобільний зв'язок, Інтернет, SMS (short message service) тощо.

Процес моделювання call-центру являє собою віртуалізацію всієї системи, або деяких її частин. Це необхідно для більш детального розгляду деяких аспектів функціонування мережі, подальшого уникнення проблем при побудові реальної телефонної мережі, виявлення слабких ланок мережі, аналізу трафіку тощо [1]. Особливістю процесу віртуалізації є налаштування АТС для обслуговування клієнтських пристроїв. Для цього доцільно створити віртуальну машину з встановленою на ній віртуальною АТС, наприклад Asterisk, FreePBX або Elastic, налаштувати її в одну мережу з клієнтськими пристроями та маршрут до неї. Варто зазначити, що встановлений SIP-сервер сам не обробляє медіапотоки – це робить окремий медіа-сервер, використовуючи протокол RTP. У реалізаціях IP-АТС завжди SIP-сервер й медіа-сервер перебувають у одному фізичному сервері. Однак системи з великим навантаженням (наприклад, у великих VoIP-операторів) можуть використовувати медіа-сервер, встановлений на іншій фізичній машині, щоб краще справлятися з обробкою сесій. Також можливе розподілення навантаження на кілька медіа-серверів [2].

Основна частина

Основними вимогами щодо стабільного функціонування call-центру можна визначити: відмовостійкість системи; якісний та швидкий доступ до мережі Інтернет; використання системи, яка здатна інтегрувати в собі декілька технологій (передача даних, голосу, відео); безпека дзвінків шляхом шифрування даних; масштабованість; здійснення дзвінків в загальну абонентську мережу PSTN (public switched telephone network) [3].

Балансування навантаження є важливим аспектом кластерів у call-центрі. Один із серверів обробки викликів у кластері може бути перевантажений, у той час як інші сервери обробки викликів можуть працювати нижче за призначений рівень навантаження. У разі такої незбалансованої ситуації можна розподілити додатковий трафік перевантаженого серверу у кластері на слабозавантажений сервер. Такий рівень розподілу навантаження не дозволяє одному серверу викликів стати вузьким місцем у продуктивності.

Аналіз балансування навантаження показав, що методи динамічного балансування навантаження дозволяють використовувати оперативну інформацію про стан системи прийняття рішень для коригування балансування навантаження під час роботи, що впливає на продуктивність системи. Прийняття рішень можливо використовувати на різних рівнях: рівні завантаження ЦП (центрального процесора), доступної пам'яті тощо [4].

У літературі [5] запропоновано використовувати двокаскадний комутатор з балансуванням навантаження для масштабування до швидкості оптоволокна. Такий підхід вважається більш ефективним, ніж інші підходи, такі як i-SLIP, що називається алгоритмом вирішення конфліктів. Його засновано на ітеративному циклічному зіставленні з ковзанням. У літературі [6] показано, що для трафіку з багатопротокольною комутацією за мітками MPLS (multiprotocol label switching) алгоритми динамічного розподілу навантаження кращі за продуктивністю, ніж алгоритм найкоротшого шляху. Зазначено, що алгоритми динамічного балансування навантаження ефективні як з легких, так і з важких умов навантаження.

Алгоритм перенаправлення запитів для кластерів веб-серверів досліджено в [7]. Проведено класифікацію кожного нового запиту на основі очікуваного впливу на ресурси сервера. При цьому кожному запиту надається вага, яка використовується для розрахунку навантаження на сервер. Щоб скоротити час збору інформації про навантаження, вибираються лише два сервери з пулів серверів, і новий запит перенаправляється на найменш завантажений сервер.

У [8] запропоновано використовувати стратегії розподілу навантаження, що засновано не тільки на ресурсах ЦП, але й на врахуванні ресурсів пам'яті. Метою запропонованих стратегій розподілу навантаження є мінімізація як часу простою ЦП, так і кількості відмов сторінок в гетерогенних розподілених системах.

У [9] використано мобільні агенти для забезпечення розподілу навантаження у глобальному мережному середовищі, такому як Інтернет. Мобільні агенти діють як координатори від імені серверів і представляють себе на віддалених майданчиках для координації дій щодо переміщення робочих місць.

У роботі створено схему організації call-центру компанії. Основним елементом виступає кластер з декількох SIP-серверів (якщо компанія здійснює досить велику кількість дзвінків). Сервер може бути як фізичний, так і віртуальний (орендована віртуальна машина (ВМ). SIP-сервери зв’язуються з CRM системою, яка призначена для автоматизації стратегій взаємодії із замовниками (клієнтами). Разом вони формують ядро системи, яке має швидкісне з’єднання з мережею Інтернет, де серверам виділяються номери, що виділені SIP-провайдером. Офіс call-центру, з’єднаний з ядром, складається з IP-телефонів та комп’ютерів. Він може територіально знаходитись як в одному будинку з ядром мережі, так і в іншому місці.

В якості програмного забезпечення запропоновано обрати Asterisk. Модульна архітектура Asterisk дозволяє легко підключати в комутаційне поле будь-яку бізнес-логіку, написану практично будь-якою мовою програмування, або реалізовану власною мовою діалплану Asterisk. Серед основних функцій Asterisk слід зазначити: підтримку як протоколів IP-телефонії, так й традиційних ліній зв'язку; підтримку відеозв'язку; підтримку шифрування розмов; наявність простих й добре документованих інтерфейсів для інтеграції з іншими системами; підтримку всіх базових та розширених функцій АТС; наявність готових дистрибутивів [10]. Asterisk може працювати практично на будь-якій платформі Linux та на деяких

інших ОС (операційних системах), таких як Solaris, BSD, MacOS X. На рис. 1 представлено схему організації call-центру компанії.

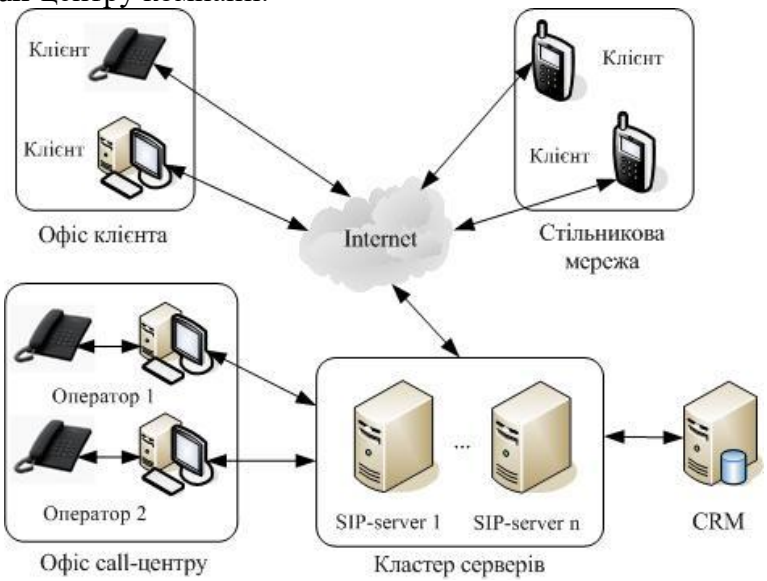

Рис. 1. Схема організації call-центру компанії

В роботі створено мережну модель call-центру, яка визначає загальну структуру, що моделюється: об'єкти в системі, а також їх фізичне розташування, взаємозв'язки та конфігурації. Мережну модель call-центру показано на рис. 2.

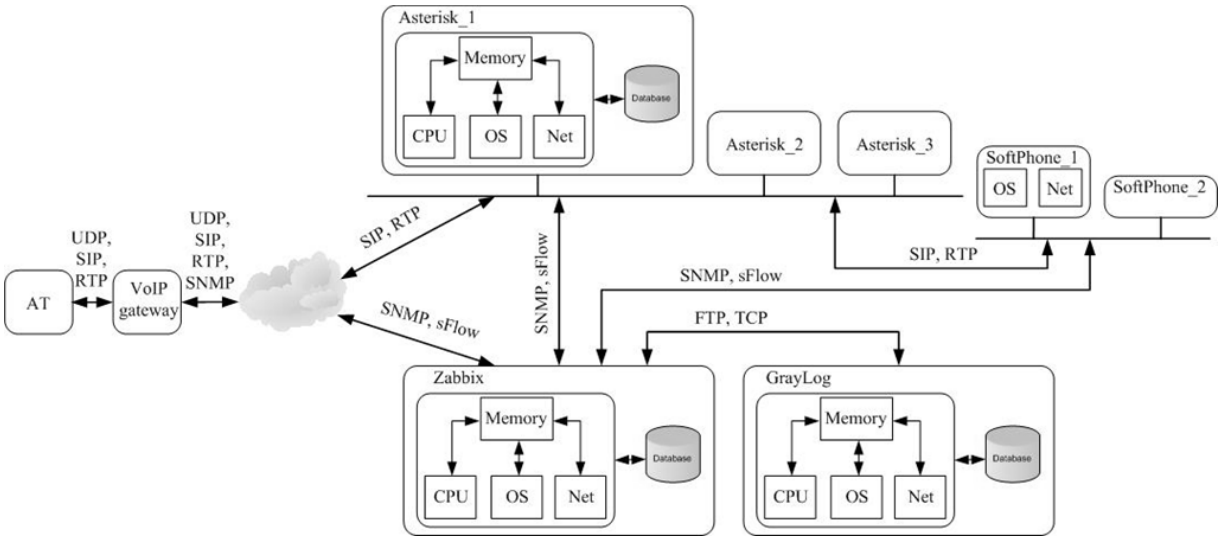

Рис. 2. Мережна модель call-центру

Основними структурними блоками домену мережі є підмережі, вузли та канали зв'язку. У цьому розумінні один сервер виклику становить об'єкт підмережі в домені мережі. Кластер створюється в мережному домені шляхом з'єднання кількох серверів викликів через канали зв'язку.

Абонентський термінал являє собою пристрій, який формує виклики до АТС на виділений її зовнішній номер. На даному сегменті мережі визачено тип трафіку – це UDP, SIP, RTP, SNMP. VoIP шлюз призначено для підключення телефонних апаратів до АТС через IP-мережу та передачі й обробки голосового трафіку.

Asterisk являє собою сервер АТС, що керує голосовим трафіком та викликами в мережі, у складі якого слід відзначити блок пам’яті, центральний процесор, мережний адаптер,

операційну систему та базу даних. Характер трафіку визначається переважно протоколами SIP та RTP. В моделі показано три сервери, які об'єднано в кластер.

У якості серверу моніторингу, що збирає необхідні метрики з хостів мережі та записує всі події до log серверу GrayLog, обрано open source продукт Zabbix. Такий вибір обумовлено простотою підключення для моніторингу будь-яких параметрів інформаційних систем у IT-інфраструктурах. Запропоновано застосування Zabbix`а для дослідження кластеру серверів викликів. Тип трафіку визначено протоколами SNMP, sFlow та SIP. Сервер логів GrayLog з'єднано з сервером моніторингу Zabbix. Логи зберігаються в його виділеній базі даних, до якої можна звернутися у будь який момент. Основний тип трафіку: TCP, FTP.

Налаштування тестового стенду починається з налаштування віртуальної машини. У якості середовища обрано VMWare.

У роботі використано технологію віртуалізації, яка дасть змогу побудувати мережу call-центру та провести необхідне тестування. У якості платформи для налаштування мережі використано гіпервізор VMWare ESXI 6.7 та кліент vCenter. Гіпервізор VMware ESXi – це потужний інструмент, який емулює апаратні ресурси, дозволяє безпечно виконувати машинні інструкції, ізолює та поділяє ресурси віртуальних машин [11]. В табл. 1 наведено обрані параметри віртуальних машин для налаштування.

Таблиця 1

Name	CPU	Memory	OS	IP-address	Hard Disk
1	2	3	4	5	6
Asterisk_1	4	6 Gb	Ubuntu 20.04	192.168.200.112	60 Gb
Asterisk_2	4	6 Gb	Ubuntu 20.04	192.168.200.104	60 Gb
Asterisk_3	4	6 Gb	Ubuntu 20.04	192.168.200.109	60 Gb
Zabbix	6	6 Gb	Ubuntu 20.04	192.168.200.107	60 Gb
Gray_Log	4	6 Gb	Ubuntu 20.04	192.168.200.110	60 Gb
SoftPhone_1	2	4 Gb	Ubuntu 20.04	192.168.200.108	60 Gb
SoftPhone_2	2	4 Gb	Ubuntu 20.04	192.168.200.111	60 Gb
Mikrotik	1	130 Mb	RouterOS	192.168.200.254	500 Mb
SiPp	4	6 Gb	Ubuntu 20.04	192.168.100.10	60 Gb

Вікно налаштування параметрів ВМ (віртуальної машини) наведено на рис. 3.

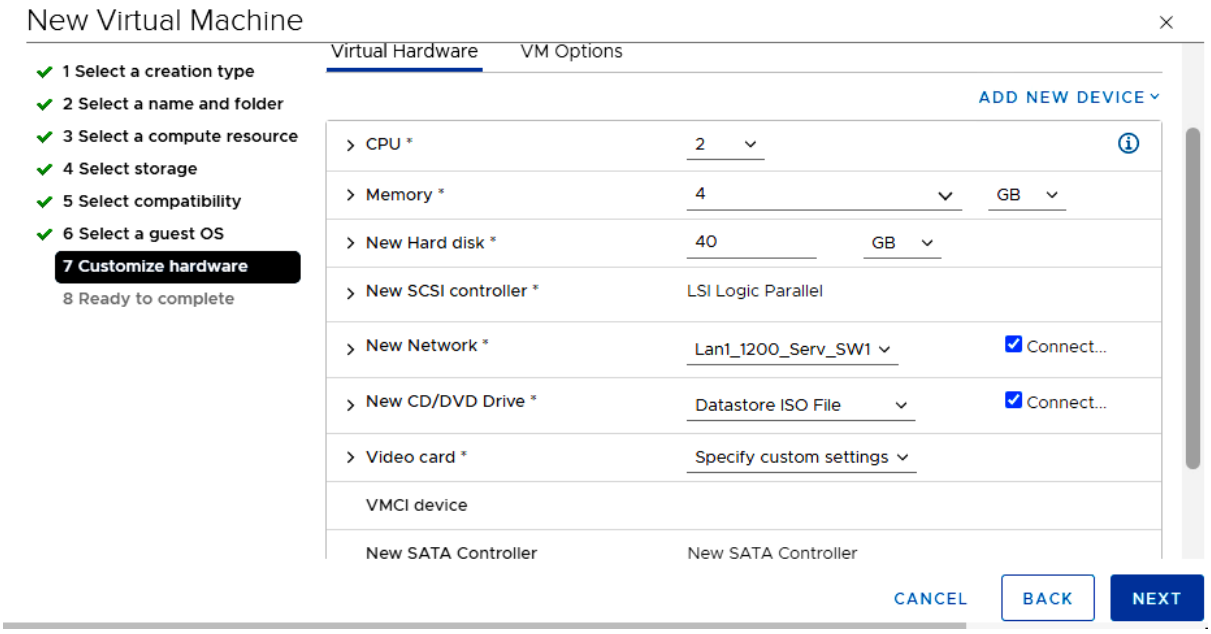

Рис. 3. Налаштування параметрів ВМ

При створенні ВМ використано наступні параметри: CPU (central processing unit), memory, місце на жорсткому диску, мережний адаптер та ISO образ ОС (операційна система). Перевагу надано ОС Ubuntu Server 20.04 за таких ознак: безкоштовність, досить велике коло користувачів, легкість для розгортання будь яких мережних або програмних сервісів, постійне оновлення, стабільність роботи [12]. В якості ядра мережі call-центру обрано образ RouterOS від компанії Mikrotik, основні функції якого наступні: маршрутизація трафіку в мережі, видавання адрес хостам, виконання ролі firewall для безпеки локальної мережі.

Наступним етапом є налаштування АТС Asterisk та створення кластеру серверів.

Встановлення АТС Asterisk на сервер Ubuntu здійснювалося з готових репозиторіїв. Готовий пакет asterisk встановлено з офіційних джерел [13, 14]. Надалі йде перевірка стану служби sudo systemctl status asterisk та проводиться конфігурація сервера. Продукт налаштовується дуже гнучко й має масу можливостей. Основними файлами конфігурації виступають sip.conf extensions.conf. Файл sip.conf відповідає за налаштування внутрішніх й зовнішніх каналів SIP в Asterisk. При цьому використано об'єкти конфігурації – піри. В налаштуваннях діє принцип успадкування, як і в більшості конфігів Asterisk. Подальші налаштування зводяться до глобальних налаштувань драйвера SIP Asterisk, які поширюються на всі об'єкти, але можуть бути перевизначені для окремих бенкетів у їх категоріях.

Слід зазначити один із найважливіших конфігураційних файлів Asterisk – extensions.conf, у якому визначається обробка та маршрутизація вхідних та вихідних викликів. Цей файл керує поведінкою всіх з'єднань, що проходять через АТС. Зміст файлу extensions.conf розбито на секції, в яких можуть бути або визначені статичні налаштування та визначення або команди плану набору, що виконуються; в цьому випадку вони називаються контекстами. Секції, призначені для статичних налаштувань, називаються general і globals, а імена контекстів визначаються системним адміністратором системи.

В роботі було прийнято рішення обмежитись лише двома віртуальними телефонами для перевірки викликів в мережу. На наступному кроці налаштовано Sip-транки для об’єднання серверів Asterisk в кластер. Для цього на кожному з серверів введено новий пір, який буде використовувати сервер для підключення до наступного. Тобто налаштовується реєстрація серверів меж собою, але без підтвердження. На рис. 4 показано процес перевірки реєстрації серверів між собою на прикладі Asterisk_1.

```
aster1*CLI> sip show peers
Name/username              Host                                  Dyn Forcerport Comedia    ACL Port     Status
ion
201/201                    192.168.200.113                        D  Auto (No)  No             57085    Unmonitored
202/202                    192.168.200.100                        D  Auto (No)  No             50981    Unmonitored
asterisk2                  192.168.200.104                           Auto (No)  No             5060     OK (1 ms)
asterisk3                  192.168.200.109                           Auto (No)  No             5060     OK (1 ms)
4 sip peers [Monitored: 2 online, 0 offline Unmonitored: 2 online, 0 offline]
```

Рис. 4. Інформація про реєстрацію в системі на прикладі Astersik_1

Подальшим кроком є налаштування SoftPhone. В якості IP телефонів в роботі використано програмне забезпечення Zoiper5. На відміну від більшості тестових стендів, на ВМ встановлено графічну оболонку XFCE для взаємодії з програмою.

Після цього проводилося створення Zabbix серверу та додавання host моніторингу. Основна задача сервера Zabbix – це моніторинг всієї мережі call-центру, зняття необхідних метрик, наприклад стану каналів, стану SIP-транків, навантаження ВМ тощо. Доменне ім'я серверу в роботі не виділяється. Підключення виконується за його IP-адресою та портом. Для обробки коду та створення динамічного контенту для веб-сервера встановлюється та налаштовується веб-сервер apache в середовищі Ubuntu, база даних MySql та PHP (personal home page) [15].

На наступному етапі розгорнуто платформу за допомогою репозиторіїв. Zabbix має чотири основні елементи, за допомогою яких можна моніторити певне робоче середовище й

збирати про нього повний пакет даних для оптимізації роботи: сервер, проксі, агент, веб-інтерфейс є частиною сервера системи й вимагає для роботи веб-сервер [16]. Для тестування та моніторингу обладнання в мережі потрібно додати відповідні параметри. На рис. 5 показано процес додавання нового host до системи.

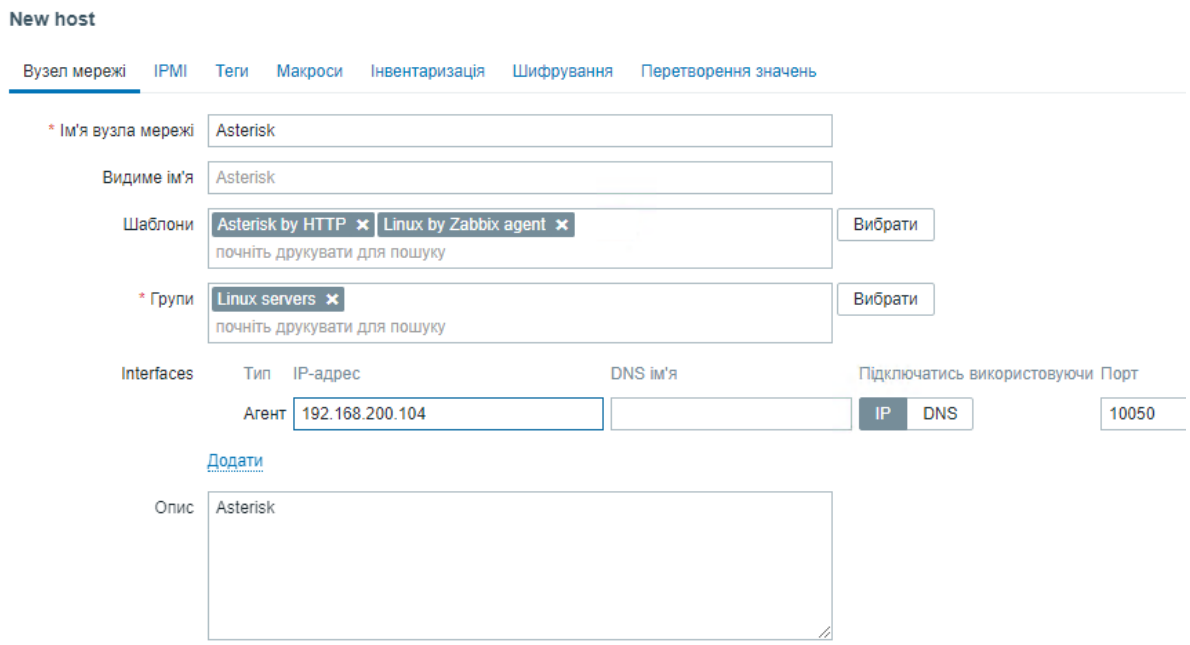

Рис. 5. Процес додавання host моніторингу

Крім інструмента моніторингу, який збирає необхідні метрики в системі та надає їх адміністратору, Zabbix може бути певним чином сконфігуровано для оперативного реагування на певні прояви системи. Робиться це за допомогою скриптів та тригерів. Тому в подальших налаштуваннях слід звернути увагу на тригери в Zabbix, які є логічними висловлюваннями, що відображають стан системи. Висловлювання, що використовуються в тригерах, є дуже гнучкими. Адміністратор може використовувати їх для створення складних логічних тестів з огляду на статистику з моніторингу [17]. Функції тригерів дозволяють посилатися на зібрані значення, поточний час та інші фактори. Стан (вираз) тригера перераховується щоразу, коли Zabbix сервер отримує нове значення даних, якщо це значення даних є частиною висловлювання. Zabbix підтримує такі важливі висловлювання тригерів: невідома важливість; в інформаційних цілях; попереджувальний; середня проблема; сталося щось важливе; надзвичайний [18]. На рис. 6 показано приклад тригерів, які використовуються для Asterisk, їх можна вибрати з шаблону при налаштуванні моніторингу host.

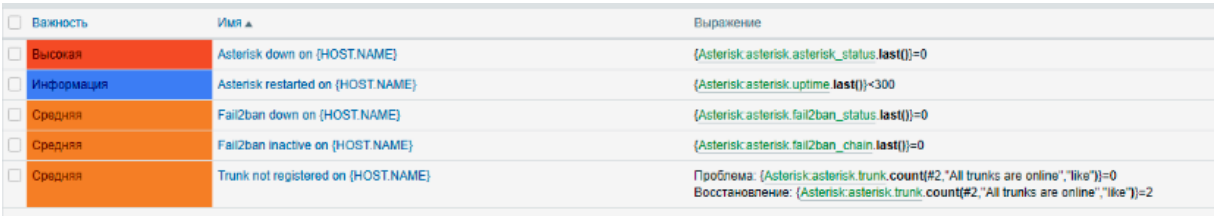

Рис. 6. Тригери для серверів Asterisk

У шаблоні присутні шість елементів даних, які визначаються в агенті, п’ять тригерів та один графік.

Віджети, графіки та карти мереж в Zabbix потрібні для візуалізації даних. Це, насамперед, полегшує сприйняття інформації адміністратором системи.

Після налаштувань для візуалізації даних в Zabbix отримано графіки та карту мережі. Карта мережі являє собою змодельовану структуру мережі call-центру. На рис. 7 показано графік з трафіком серверу для Asterisk_1.

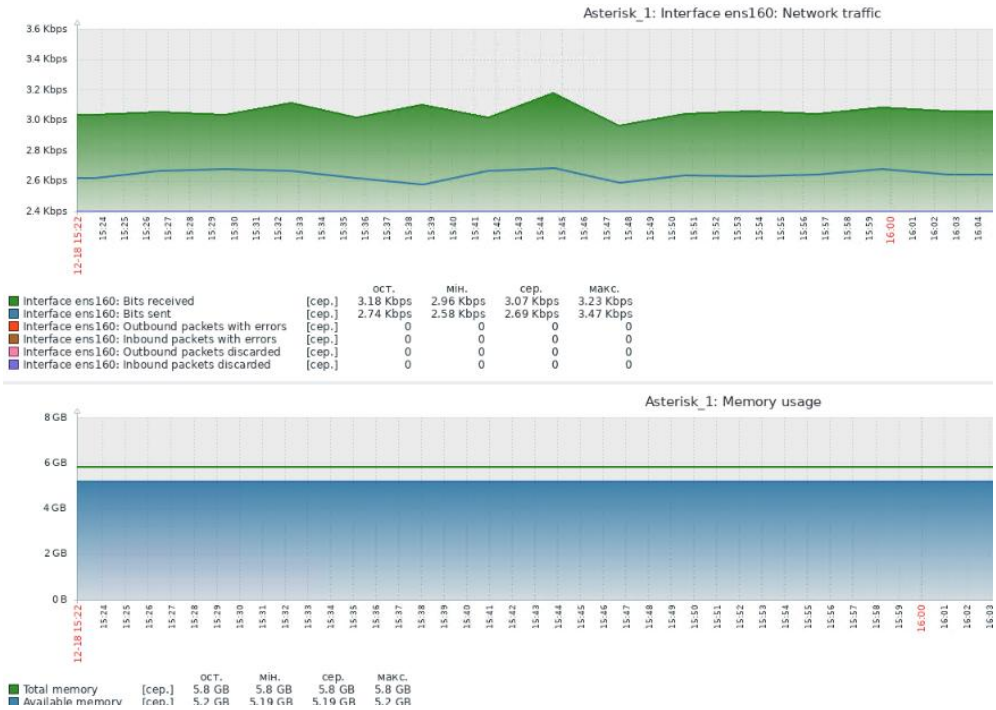

Рис. 7. Графіки пропускної здатності інтерфейсу ens160 та пам’яті сервера Astersik_1

На рис. 8 показано топологію мережі call-центру (карту мережі call-центру), яку отримано в результаті налаштувань.

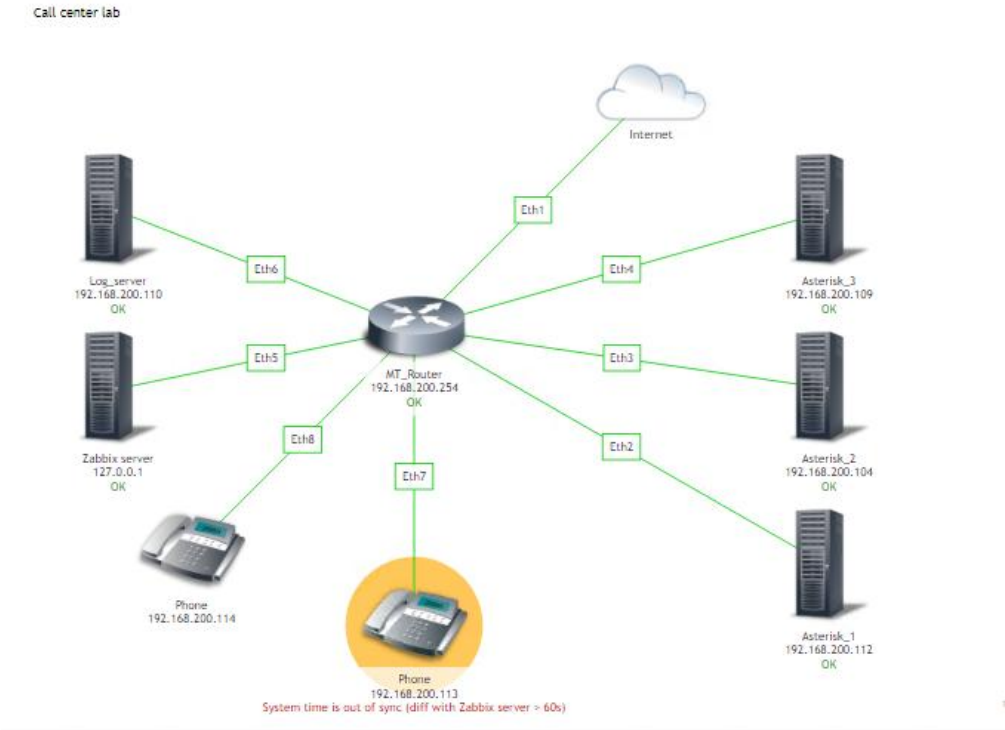

Рис. 8. Карта мережі call-центру

Для прикладу телефон з адресою 192.168.200.113 підсвічено жовтим кольором (в його системі заздалегідь був налаштований інший час). Видно, що даний тригер відображається на мапі як попереджувальний та відображає коротку інформацію адміністратору.

У роботі проведено аналіз та керування балансуванням системою Zabbix. Для цього показано процес тестування навантаження викликами на три сервери Asterisk та реалізація можливостей Zabbix як інструмента балансування навантаження. Розраховано кількість викликів, з якими здатний працювати call-центр.

Тестування навантаження відбувається за допомогою утиліти SIPp, яка є потужною утилітою для створення навантаження на SIP обладнання. Зазвичай SIPp використовується для перевірки відмови системи IP-телефонії, виявлення максимально допустимого навантаження або DDoS-атак (distributed denial-of-service). Сценарій сесії в SIPp описується в XML (extensible markup language) файлі. Можливо скористатися одним із безлічі сценаріїв, що розповсюджуються в комплекті з SIPp, або створити свій сценарій сесії в SIPp.

У роботі створено спеціальний SIP-peer з ім'ям sipp для прийняття викликів від SIPp. В Створено власний сценарій сесії в SIPp, приклад налаштування параметрів якого наведено на рис. 9.

```
sipp@phone:/etc$ cd /etc/sipp-3.3
sipp@phone:/etc/sipp-3.3$ sudo sipp 192.168.200.112 -s 201 -i 192.168.100.10 -d 2h -l 60 -aa -mi 10.10.10.2 -rtp_echo -nd -r
10
```

Рис. 9. Команда для старту відправки викликів SIPp з заданими параметрами

Особливістю сценарію сесії в SIPp є можливість Asterisk авторизувати SIPp не за паролем, а за IP-адресою, яку вказано в полі host. На рис. 10 показано фрагмент роботи утиліти SIPp при заданих параметрах навантаження.

```
1 calls (limit 2)                    Peak was 2 calls, after 30 s
1 Running, 1 Paused, 3 Woken up
0 dead call msg (discarded)          390 out-of-call msg (discarded)
3 open sockets
1309 Total RTP pckts sent            0.000 last period RTP rate (kB/s)

                           Messages  Retrans   Timeout   Unexpected-Msg
    INVITE ---------->         7         0         0
       100 <----------         7         0         0         0
       180 <----------         7         0         0         0
       200 <----------  E-RTD1 6         0         0         0

       ACK ---------->         6         0
           [ NOP ]
     Pause [   8000ms]         6                             2
           [ NOP ]
     Pause [   1000ms]         4                             0
       BYE ---------->         4         0         0
       200 <----------         4         0         0         0

------ [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

Рис. 10. Фрагмент роботи утиліти SIPp

В процесі навантаження кількість викликів поступово збільшувалась на 50. Виявилось, що один сервер Asterisk з поточними його параметрами здатний обробити максимум 915 одночасних викликів. Результати тестування показано на графіку (рис. 11).

Виходячи з результатів, отриманих на графіку, для сервера Asterisk було встановлено порогове значення завантаженості CPU в 90 % (йому відповідає 850 викликів). Саме при ньому для уникнення повного перевантаження сервера потрібно запускати в роботу додаткові сервери.

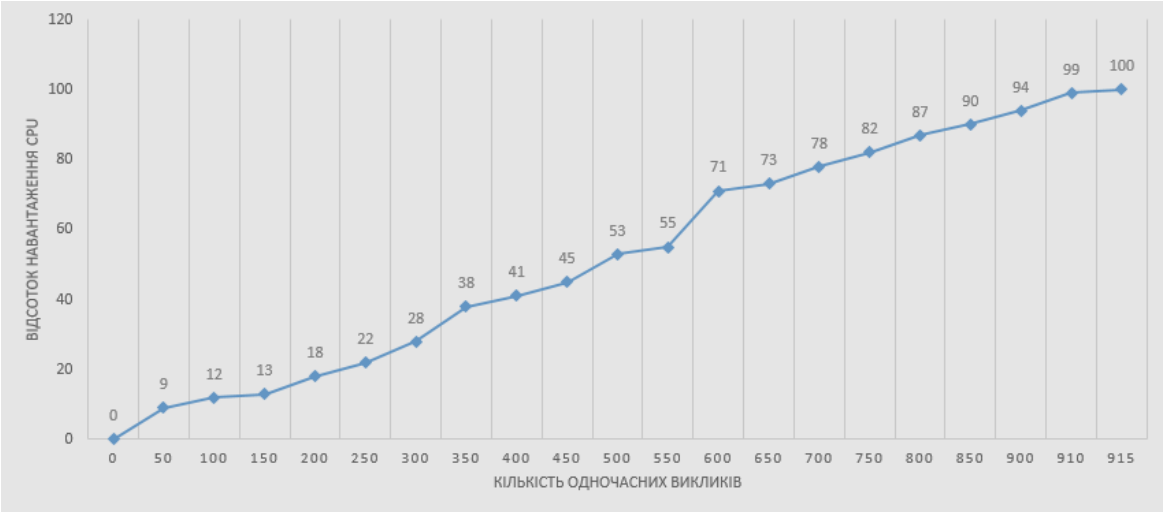

Рис. 11. Графік залежності навантаження CPU від кількості викликів

За допомогою Zabbix запущено процес балансування навантаження на кластер SIP серверів, виходячи з метрик стану CPU та стану SIP-транку. Для того, щоб Zabbix мав змогу перевіряти стан CPU та стан транку, розроблено необхідний скріпт та додано його в існуючі параметри моніторингу host.

В мережі call-центру ВМ Asterisk_1 виступає як в ролі АТС, так і в ролі сервера балансування навантаження [19]. В якості програмного забезпечення, яке відповідає за балансування, на сервері встановлено та налаштовано модуль Kamailio. За замовчуванням його вимкнено.

Для реалізації заданих умов перевірки використано можливості скриптів та тригерів в Zabbix. При виявленні заданого відсотка завантаженості CPU (в даному випадку 90 %) налаштовано скрипт, за яким Zabbix відправляє команду до сервера Asterisk на увімкнення даного модуля. На рис. 12 показано процес створення тригеру, котрий реагує на завантаження CPU host Asterisk.

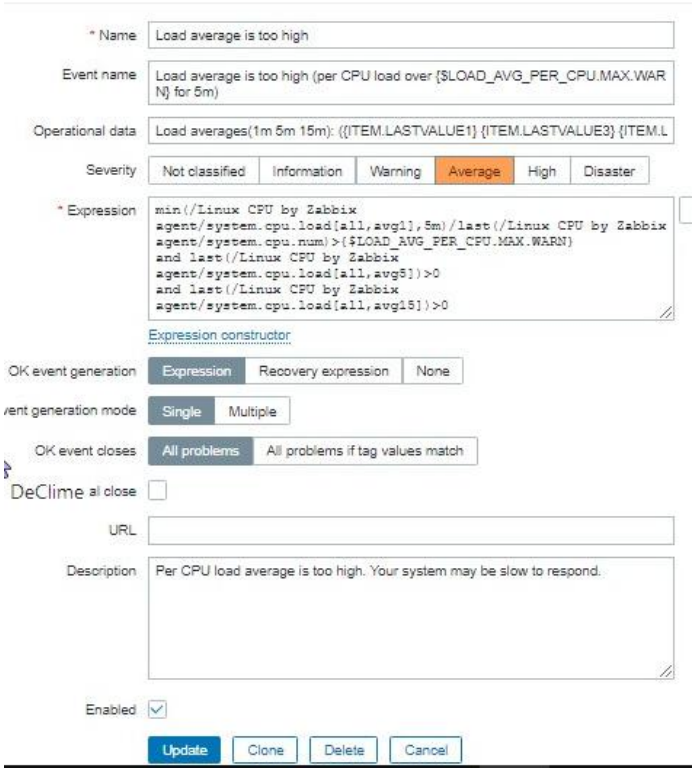

Рис. 12. Процес створення тригеру навантаження CPU

Таким чином, в мережі call-центру налаштовано можливості тестового стенду з використанням Zabbix для балансування навантаження в кластері серверів Asterisk. Було виявлено, що кластер серверів Asterisk здатний обслуговувати 2550 викликів одночасно.

Висновки

Проаналізовано проблеми телекомунікаційних серверів у центрах обробки викликів, що пов`язані з питаннями забезпечення більшої пропускної здатності. Одним з рішень є кластерізація серверів викликів.

Розглянуто особливості процесу моделювання call-центру. Доведено, що для огляду деяких аспектів функціонування мережі необхідна віртуалізація всієї системи або деяких її частин. Це безсумнівно корисно для огляду функціонування мережі, виявлення проблем з трафіком та аналізом параметрів слабких ланок мережі. Проаналізовано особливості діяльності та реалізації мережі call-центру. У якості ядра даної структури розглянуто кластер серверів з використанням віртуальної IP-АТС Asterisk. Такий вибір обумовлено модульною архітектурою Asterisk та можливістю роботи на багатьох ОС. Показано, що одним із основних аспектів якісного функціонування call-центру є балансування навантаження серверів.

Проведено аналіз балансування навантаження з використанням різних алгоритмів та стратегій.

Для розробки та налаштування тестового стенду в роботі створено схему організації call-центру компанії, основним елементом якої виступає кластер SIP-серверів, та мережну модель call-центру. Мережна модель визначає загальну структуру об'єктів, а також їх фізичне розташування, взаємозв'язки та конфігурації. Для створення мережі call-центру та необхідного тестування використано технологію віртуалізації. У якості платформи для налаштування мережі використано гіпервізор VMWare ESXI 6.7 та кліент vCenter. Проведено налаштування АТС Asterisk та створення кластеру серверів. В мережі call-центру ВМ Asterisk_1 виступає як в ролі АТС, так і в ролі сервера балансування навантаження.

Для тестування кластеру серверів викликів, які змодельовано як географічно розподілені сегменти телекомунікаційної мережі, запропоновано використати open source продукт Zabbix. Для візуалізації даних в Zabbix отримано характеристику пропускної здатності для сервера Asterisk_1 та карту мережі, яка фактично являє собою змодельовану структуру мережі call-центру.

Показано процес тестування навантаження викликами на три сервери Asterisk та реалізація можливостей Zabbix для балансування навантаженням. Тестування навантаження відбувається за допомогою утиліти SIPp, яка є потужною утилітою для створення навантаження на SIP обладнання. Створено власний сценарій сесії в SIPp для прийняття викликів. При цьому розрахована кількість викликів, з якими здатний працювати call-центр. Виявлено, що один сервер Asterisk з поточними його параметрами здатний обробити максимум 915 одночасних викликів.

Запущено процес балансування навантаження на кластер SIP серверів, виходячи з метрик стану CPU та стану SIP-транку. Виявлено, що кластер серверів Asterisk здатний обслуговувати 2550 викликів одночасно.

Список літератури:

1. Токар Л.О. Особливості побудови віртуальних АТС // Радіотехніка. 2022. Вип. 208. С. 55 – 64. doi:10.30837/rt.2022.1.208.06.

2. Voxlink. URL: https://voxlink.com/ (дата звернення 14.12.2022).

3. X. Xiao, J. Sun, J. Yang. Operation and maintenance(O&M) for data center: An intelligent anomaly detection approach // Computer Communications. 2021. Vol. 178. pp. 141 – 152. doi:10.1016/j.comcom.2021.06.03.

4. K. Gardner, J. Abdul Jaleel, A. Wickham, S. Doroudi. Scalable load balancing in the presence of heterogeneous servers // Performance Evaluation Review. 2020. Vol. 48, no. 3. pp. 37 – 38. doi:10.1145/3453953.3453961.

5. A. Siokis, K. Christodoulopoulos, N. Pleros, E. Varvarigos. Electro-optic switches based on space switching of multiplexed WDM signals: Blocking vs non-blocking design trade-offs // Optical Switching and Networking. 2017. Vol. 25. pp. 40 – 56. doi:10.1016/j.osn.2017.

6. D. Medhi, K. Ramasamy. Routing and Traffic Engineering using MPLS", in Network Routing. 2018. chapter 23. pp. 766 – 785. doi: 10.1016/B978-0-12-800737-2.00027-2.

7. Ataie Reza Ehsan, Sayed Entezari-Maleki, Etesami Ehsan, Egger Bernhard, Sousa Leonel, Movagharg Ali. Modeling and evaluation of dispatching policies in IaaS cloud data centers using SANs // Sustainable Computing, Informatics and Systems. 2022. Vol. 33. pp. 88 – 102. doi:10.1016/j.suscom.2021.

8. C. Li, Q. Cai, Y. Lou. Optimal data placement strategy considering capacity limitation and load balancing in geographically distributed cloud // Future Generation Computer Systems. 2022. Vol. 127. pp. 142 – 159. doi:10.1016/j.future.2021.08.014.

9. M. Ali, S. Bagchi. Probabilistic normed load monitoring in large scale distributed systems using mobile agents // Future Generation Computer Systems. 2019. Vol. 96. pp. 148 – 167. doi:10.1016/j.future.2019.01.053.

10. Баскаков І. В., Пролетарський А. В., Мельников С. А., Федотов Р. А. IP-телефонія у комп'ютерних мережах // Інтернет-університет інформаційних технологій, 2020. 227 с.

11. О.А. Колтаков, Л.О. Токар. Віртуалізація ресурсів підприємства // Матеріали IV Міжнар. студ. наук. конф. Наука сьогодення: від досліджень до стратегічних рішень. 17 черв. 2022. С.178 – 180.

12. Граннеман Скотт., Linux. Карманный справ очник. Sams Publishing, 2019. 464 с.

13. Pelayo Nuno, Carla Suárez, Eva Suárez. A Diagnosis and Hardening Platform for an Asterisk VoIP PBX // Security and Communication Networks. 2020. pp. 1 – 14. doi:10.1155/2020/8853625.

14. Linux Open Source Software Technologies. URL: https://losst.pro/ (дата звернення 19.01.2023).

15. Uytterhoeven Patrik, Olups Rihards. Zabbix 4 Network Monitoring. Third Edition Packt, 2019. 798 p.

16. Van Baekel Brian, Liefting Nathan. Zabbix 6 IT Infrastructure Monitoring. Packt Publishing, 2022. 506 p.

17. ZABBIX 6.2 Improve your monitoring performance. URL: https://www.zabbix.com /(дата звернення 22.12.2022).

18. A. Pradana, I. Widiasari, R.Efendi. Implementasi Sistem Monitoring Jaringan Menggunakan Zabbix Berbasis SNMP // Security and Communication Networks. 2022. Vol. 19(2). pp. 248 – 262. doi:10.24246/aiti.v19i2.248-262.

19. Andrea Clementia, Emanuele Nataleb, Isabella Ziccardi. Parallel Load Balancing on constrained client-server topologies // Theoretical Computer Science. 2021. Vol. 8952021. pp. 16-33. doi:10.1145/3350755.3400232.

Надійшла до редколегії 15.02.2023

Відомості про авторів:

Токар Любов Олександрівна – канд. техн. наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри інфокомунікаційної інженерії ім. В.В. Поповського (ІКІ); Україна; e-mail: liubov.tokar@nure.ua; ORCID: https://orcid.org/0000-0002-7780-1928

Колтаков Олександр Анатолійович – компанія IT-Lance, системний адміністратор, Україна, email: oleksandr.koltakov@nure.ua

Цилюрик Вадим Євгенович – Харківський національний університет радіоелектроніки, магістр кафедри інфокомунікаційної інженерії ім. В.В. Поповського, Україна, email: vadym.tsyliuryk@nure.ua