Міністерство освіти і науки України
Кам’янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп’ютерних наук

Дипломна робота
бакалавра
з теми **«РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ КОМП’ЮТЕРНИХ МЕРЕЖ НА ОСНОВІ ZABBIX»**

Виконав: студент 3 курсу, групи KNms1-B19
спеціальності 122 Комп’ютерні науки
Богуш Дмитро Васильович

Керівник: Понеділок В.В.
старший викладач кафедри комп’ютерних наук, кандидат технічних наук

Кам’янець-Подільський – 2022

ЗМІСТ

ВСТУП 4
РОЗДІЛ 1. КЛАСИФІКАЦІЯ КОМП'ЮТЕРНИХ МЕРЕЖ 6
1.1. Локальні мережі 6
1.2. Глобальні мережі 7
РОЗДІЛ 2. МЕТОДИ МОНІТОРИНГУ КОМП'ЮТЕРНОЇ МЕРЕЖІ 12
2.1. Загальне поняття моніторингу мережі 12
2.2. Види та способи моніторингу мережі 13
2.3. Засоби моніторингу мережі 14
2.4. Напрямки розвитку систем моніторингу 15
2.5. Протоколи керування мережею 16
2.6. Протокол SNMP 18
2.6.1. Функції компонентів SNMP 20
РОЗДІЛ 3. ЗАГАЛЬНИЙ МЕТОД ІНСТРУМЕНТУ ZABBIX ДЛЯ МОНІТОРИНГУ КОМП'ЮТЕРНОЇ МЕРЕЖІ 25
3.1. Загальні відомості про систему моніторингу Zabbix 25
3.1.1. Архітектура Zabbix 25
3.1.2. Переваги Zabbix 27
3.2. Прикладний програмний інтерфейс 28
3.2.1. Переваги архітектури Zabbix 30
3.3. Взаємодія з пам'яттю 30
3.3.1. Кеш-пам'ять 31
3.3.2. Групові операції 31
3.4. Веб-інтерфейс Zabbix 31
3.5. Аналоги Zabbix та їх недоліки 32
3.5.1. Система моніторингу Nagios 32
3.5.2. Система моніторингу Ganglia 33
3.5.3. Система моніторингу Cacti 34
3.5.4. Система моніторингу Munin 34
РОЗДІЛ 4. ІНТЕГРАЦІЯ ТА РЕАЛІЗАЦІЯ МОНІТОРИНГУ ZABBIX У КОМП'ЮТЕРНІЙ МЕРЕЖІ 35
4.1. Встановлення Zabbix Server 35

4.1.1. Встановлення Ubuntu Server 35
4.1.2. Встановлення Zabbix Server 36
4.2. Моніторинг сервера Zabbix 37
4.2.1. SNMP-моніторинг 37
4.2.2. Створення хосту та елементів 37
4.2.3. Моніторинг за допомогою шаблонів 38
4.2.4. Моніторинг маршрутизатора 40
4.3. Моніторинг серверів Zabbix 42
4.4. Топологія мережі 43
4.5. Технічне рішення моніторингу Zabbix 44
ВИСНОВКИ 50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 51

ВСТУП

Сучасний світ техніки важко уявити без комп’ютерної мережі, адже це один з головних компонентів щоденного користування. Від спілкування, до обміну файлами. Від приватних мереж до корпоративних гігантів, які об’єднують до 500 і більше пристроїв на одному підприємстві.

Спектр соціальних груп, що підключаються до мережі Інтернет і що шукають інформацію в WWW, весь час розширюється за рахунок користувачів, що не відносяться до категорії фахівців в області інформаційних технологій. Будь-який користувач, що відчув корисність і незамінність мережі для своєї професійної діяльності або захоплень, приєднується до величезної армії споживачів інформації в «Всесвітній Павутині».

Веб-технологія повністю перевернула наші уявлення про роботу з інформацією та з комп'ютером взагалі. Виявилось, що традиційні параметри розвитку обчислювальної техніки - продуктивність, пропускна спроможність, запам’ятовуючих пристроїв, не враховували головного «вузького місця» системи - інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І лише коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, послідував безпрецедентний вибух інтересу до можливостей обчислювальної техніки.

З розвитком технологій гіпертекстової розмітки в Інтернеті почали з'являтися все більше сайтів, тематика яких була абсолютно різною – від сайтів крупних компаній, що оповідають про успіхи компанії і її провали, до сайтів маленьких фірм, що пропонують відвідати їх офіси в межах одного міста.

Розвиток Інтернет-технологій послужив поштовхом до появи нової гілки в Інтернеті – Інтернет-форумів. Почали з'являтися сайти, і навіть цілі портали, на яких люди зі всіх куточків планети можуть спілкуватися, отримувати відповіді на будь-які питання і, навіть, укладати ділові операції.

Але з цим і з’явилась низка проблем, які потрібно було вирішувати якнайшвидше, такі як: низька продуктивність мережі, низька швидкість передачі даних від серверу до користувача, припинення роботи серверу без фізичного втручання.

Із появою цих проблем з’явилась потреба у відслідковуванні цих помилок у режимі реального часу. З’явився термін «моніторингу» мережі. Постійно контролюючи стан здоров'я та надійність мережі та шукаючи тенденції, система моніторингу відстежує та записує параметри мережі. До них відносяться швидкість передачі даних (пропускна здатність), частота помилок, час простою / час роботи, відсотки часу використання та час відповіді користувачеві та автоматизовані введення та запити. Коли досягнуто заздалегідь визначених порогів параметрів, спрацьовують тривожні сигнали та ініціює процеси управління помилками мережі. Одним із розповсюджених програмних застосунків є Zabbix.

Zabbix - це універсальний інструмент моніторингу, здатний відстежувати динаміку роботи серверів та мережевого обладнання, швидко реагувати на позаштатні ситуації і попереджати можливі проблеми з навантаженням. Система моніторингу Zabbix може збирати статистику в зазначеної робочої середовищі і діяти в певних випадках заданим чином.

У Zabbix є 4 основні інструменти, за допомогою яких можна моніторити певну робочу середу і збирати про неї повний пакет даних для оптимізації роботи.

Сервер - ядро, що зберігає в собі всі дані системи, включаючи статистичні, оперативні і конфігурацію. Дистанційно управляє мережевими сервісами, оповіщає адміністратора про існуючі проблеми з обладнанням, що знаходяться під наглядом.

Проксі - сервер, який збирає дані про доступність і продуктивності пристроїв, який працює від імені сервера. Всі зібрані дані зберігаються в буфер і завантажуються на сервер. Потрібен для розподілу навантаження на сервер.

Агент - програма, яка активно моніторить і збирає статистику роботи локальних ресурсів (накопичувачі, оперативна пам'ять, процесор і ін.) і додатків.

Веб-інтерфейс - є частиною сервера системи і вимагає для роботи вебсервер. Часто запускається на тому ж фізичному вузлі, що і Zabbix.

РОЗДІЛ 1. КЛАСИФІКАЦІЯ КОМП’ЮТЕРНИХ МЕРЕЖ

У даному розділі визначимо якими є комп’ютерні мережі, переваги та недоліки кожної з них.

Комп'ютерна мережа — це два або більше комп'ютерів, що обмінюються інформацією за допомогою ліній зв'язку.

Комп'ютерна мережа дозволяє передавати інформацію з одного комп'ютера на інший, а значить спільно використовувати ресурси, наприклад, принтери, модеми та пристрої зберігання інформації. Великою мережею керує системний адміністратор, який встановлює рівень доступності ресурсів, визначає паролі доступу до ресурсів, права користувачів.

Мережі бувають:

- Локальні – об'єднують комп'ютери, що знаходяться недалеко один від одного, наприклад, що стоять у сусідніх кімнатах, в одному будинку;
- Глобальні - комп'ютери можуть знаходитися у різних містах і країнах. Глобальні мережі, зазвичай, об'єднують кілька локальних мереж.

1.1. Локальні мережі

До локальних мереж (Local Area Network, LAN) зазвичай відносять мережі, комп'ютери яких зосереджені відносно невеликих територіях (зазвичай, в радіусі до 1-2 км). Класичним прикладом локальних мереж є мережа одного підприємства, розташованого в одному або декількох будівлях, що стоять поруч. Невеликий розмір локальних мереж дозволяє використовувати їх побудови досить дорогі і високоякісні технології, що забезпечує високу швидкість обміну інформацією між комп'ютерами.

Основні функції локальної мережі:

- Оптимізація робочого процесу. Домашня локальна мережа, організована, наприклад, офісі, забезпечує його співробітникам можливість

дистанційного обміну даними, і навіть спільного використання всіх видів оргтехніки;

- Можливість віддаленого адміністрування. Так, корпоративна локальна мережа дозволяє одному фахівцю надавати технічну підтримку кількох десятків різних пристроїв;
- Спілкування. Звичайно, повністю замінити «інтернет-коннектинг» локальні мережі не зможуть, але в тих випадках, коли потрібно організувати власний, закритий від зовнішніх користувачів, канал зв'язку (наприклад, форум співробітників корпорації) локальні мережі просто незамінні;
- Економія. Зручніше платити за підключення до інтернету та забезпечити ним всіх співробітникам організації (пристроям користувача) можливість вільного доступу, ніж платити за доступ до всесвітньої павутини кожному співробітнику (гаджету) індивідуально;

Таким чином, локальна мережа – дуже корисний інструмент у будь-якій сфері діяльності. По суті, саме локальні мережі замінили всім відому «голубину пошту» як на будь-якому підприємстві, так і між друзями-знайомими.

1.2. Глобальні мережі

Глобальні мережі (Wide Area Network, WAN) – це мережі, призначені для об'єднання окремих комп'ютерів та локальних мереж, розташованих на значній відстані (сотні та тисячі кілометрів) одна від одної. Оскільки організація спеціалізованих високоякісних каналів зв'язку великої протяжності є досить дорогою, то раніше, у глобальних мережах нерідко використовувалися вже існуючі і не призначені для побудови комп'ютерних мереж лінії (наприклад, телефонні або телеграфні). У зв'язку з цим швидкість передачі даних у таких мережах була істотно нижчою, ніж у локальних.

Основні функції глобальної мережі:

- Транспортні функції глобальної мережі. Світова мережа використовується як транзитний транспортний механізм. Самі дані зберігаються і виробляються в комп'ютерах, що належать локальним мережам, а глобальна мережа їх лише переносить із однієї локальної мережі до іншої.

- Інформаційні функції Інтернет. Це доступ до гіпертекстової інформації Web-вузлів, що робить джерелом даних не окремі комп'ютери, а всю глобальну мережу. Потрібно відзначити і широкомовне поширення звукозаписів, організацію інтерактивних «розмов» - chat, організацію конференцій з інтересів, пошук інформації та багато іншого.

- Електронна комерція. До електронної комерції відносяться: брокерська, рекламна та торговельна діяльність. Брокер зводить покупця і продавця щодо угоди, потім бере комісійні за проведену угоду. Найяскравішою формою електронної комерції є Інтернет–магазини.

Прийнято розрізняти корпоративні мережі, побудовані з використанням:

- виділених каналів;
- комутації каналів;
- комутації пакетів.

Якщо перші два є застарілими та майже не використовуються у сучасному світі, окрім як на великих підприємствах, то комутація пакетів є основною у побудові глобальних комп’ютерних мереж. Прикладом технології мереж з комутацією пакетів є мережі X.25, frame relay, мережі TCP/IP.

Територіальні мережі поділяються на дві великі категорії:

- магістральні мережі;
- мережі доступу.

Магістральні територіальні мережі (backbone wide-area networks) використовуються для утворення однорангових зв’язків між великими локальними мережами, що належать до великих підрозділів підприємства. Магістральні територіальні мережі повинні забезпечувати високу пропускну здатність, оскільки на магістралі поєднуються потоки великої кількості підмереж. Крім того,

магістральні мережі повинні бути постійно доступними, тобто забезпечувати дуже високий коефіцієнт готовності, оскільки по них передається трафік багатьох критично важливих для успішної роботи підприємства додатків (business-critical applications). З огляду на особливу важливість магістральних засобів їм може «прощатися» висока вартість. Так як у підприємства зазвичай є не так багато великих мереж, то до магістральних мереж не пред'являються вимоги підтримки розгалуженої інфраструктури доступу.

Зазвичай як магістральні мережі використовуються цифрові виділені канали зі швидкостями від 2 до 622 Мбіт/с, якими передається трафік IP, IPX або протоколів архітектури SNA компанії IBM, мережі з комутацією пакетів frame relay, ATM, X.25 або TCP/IP. За наявності виділених каналів забезпечення високої готовності магістралі використовується змішана надлишкова топологія зв'язків.

Під мережами доступу розуміються територіальні мережі, необхідних зв'язку невеликих локальних мереж та окремих віддалених комп'ютерів з центральної локальної мережею підприємства. Якщо організації магістральних зв’язків при створенні корпоративної мережі завжди приділялася велика увага, організація віддаленого доступу співробітників підприємства перейшла в розряд стратегічно важливих питань тільки останнім часом. Швидкий доступ до корпоративної інформації будь-якої географічної точки визначає для багатьох видів діяльності підприємства якість прийняття рішень його співробітниками. Важливість цього чинника зростає зі збільшенням кількості працівників, що працюють вдома (telecommuters - телекомм'ютерів), часто перебувають у відрядженнях, і зростання кількості невеликих філій підприємств, що у різних містах і, можливо, різних країнах.

Як окремі віддалені вузли можуть також виступати банкомати або касові апарати, що вимагають доступу до центральної бази даних для отримання інформації про легальних клієнтів банку, пластикові картки яких необхідно авторизувати на місці. Банкомати або касові апарати зазвичай розраховані на взаємодію з центральним комп'ютером через мережу X.25, яка свого часу

спеціально розроблялася як мережа для віддаленого доступу неінтелектуального термінального обладнання до центрального комп'ютера.

До мереж доступу висуваються вимоги, що суттєво відрізняються від вимог до магістральних мереж. Оскільки точок віддаленого доступу підприємство може бути дуже багато, однією з основних вимог є наявність розгалуженої інфраструктури доступу, яка може використовуватися співробітниками підприємства як при роботі будинку, так і у відрядженнях. Крім того, вартість віддаленого доступу має бути помірною, щоб економічно виправдати витрати на підключення десятків або сотень віддалених абонентів. При цьому вимоги до пропускної спроможності в окремого комп'ютера або локальної мережі, що складається з двох-трьох клієнтів, зазвичай укладаються в діапазон кількох десятків кілобіт на секунду (якщо така швидкість і не цілком задовольняє віддаленого клієнта, то зазвичай його зручностями жертвують заради економії коштів підприємства).

Як мережі доступу зазвичай застосовуються телефонні аналогові мережі, мережі ISDN і рідше – мережі frame relay. При підключенні локальних мереж філій також використовуються виділені канали зі швидкостями від 19,2 до 64 Кбіт/с. Якісний стрибок у розширенні можливостей віддаленого доступу стався у зв'язку зі стрімким зростанням популярності та поширеності Internet. Транспортні послуги Internet дешевші, ніж послуги міжміських та міжнародних телефонних мереж, а їх якість швидко покращується.

Програмні та апаратні засоби, які забезпечують підключення комп'ютерів або локальних мереж віддалених користувачів до корпоративної мережі, називаються засобами віддаленого доступу. Зазвичай на клієнтській стороні ці засоби представлені модемом та відповідним програмним забезпеченням.

Організацію масового віддаленого доступу з боку центральної локальної мережі забезпечує сервер віддаленого доступу (Remote Access Server, RAS). Сервер віддаленого доступу є програмно-апаратним комплексом, який поєднує функції маршрутизатора, мосту і шлюзу. Сервер виконує ту чи іншу функцію в залежності від типу протоколу, яким працює віддалений користувач або віддалена мережа.

Сервери віддаленого доступу зазвичай мають багато низькошвидкісних портів для підключення користувачів через аналогові телефонні мережі або ISDN.

РОЗДІЛ 2. МЕТОДИ МОНІТОРИНГУ КОМП'ЮТЕРНОЇ МЕРЕЖІ

В даному розділі визначимо який є моніторинг комп'ютерної мережі, його види та способи, засоби та сучасний розвиток і удосконалення моніторингу, що є важливим у наш час.

2.1. Загальне поняття моніторингу мережі

Моніторинг працездатності обчислювальної мережі – це робота системи, яка виконує постійне спостереження за обчислювальною мережею у пошуках повільних чи несправних систем, а при виявленні таких повідомляє про них адміністратору мережі за допомогою засобів оповіщення. Моніторинг є одним з найважливіших завдань, необхідних організації повноцінного управління обчислювальної мережею. Процес виявлення самих несправностей та формування комплексу заходів може зайняти значний час та суттєво вплинути на функціонування системи автоматизації підприємства загалом. Часті відмови або тривалі періоди непрацездатного стану мережі можуть призвести до повної втрати працездатності системи автоматизації підприємства. Для підвищення оперативності вжиття заходів, здатних повернути обчислювальну мережу в режим штатного функціонування, необхідне проведення моніторингу мережі, який переважно залежить від людського фактору. Професійного досвіду спеціаліста, що експлуатує великі обчислювальні мережі, часто не вистачає для оперативної діагностики мережі та прийняття рішення під час усунення збоїв у роботі.

Деталізуючи поняття обчислювальної мережі, можна сказати, що обчислювальна мережа складається з середовища передачі даних, каналоутворювального обладнання, кінцевого обладнання користувача. Здійснення моніторингу необхідно проводити за кожним елементом обчислювальної мережі, надалі об'єднуючи результат моніторингу єдину оцінку всієї обчислювальної мережі.

2.2. Види та способи моніторингу мережі

Найпростішою системою моніторингу, чи, точніше сказати, командою для моніторингу, що використовується практично у всіх невеликих організаціях, в яких відсутні будь-які програмні або апаратні системи моніторингу, є команда «ping». Контроль здійснюється періодично, при зникненні мережі або постійному режимі до певних вузлів мережі. Після того, як виявляється відсутність зв'язку з будь-яким із вузлів мережі, проводиться уточнююча робота з виявлення конкретної несправності мережі (мережі зв'язку, каналоутворююча апаратура тощо). Проте використання команди «ping» не дозволяє оперативно знайти несправність та потребує постійної операторської присутності. При використанні великих обчислювальних мереж або різнорідних мереж, дана команда може просто не працювати.

Сучасні вимоги до обчислювальних мереж потребують більш точного та гнучкого підходу до моніторингу. Від коректної роботи Web-серверів і серверів баз даних може залежати працездатність внутрішньокорпоративних додатків і важливих зовнішніх сервісів для клієнтів.

Збої та порушення роботи маршрутизаторів можуть порушувати зв'язок між різними частинами корпорації та її філіями. Сервери внутрішньої пошти та мережевих месенджерів, автоматичних оновлень та резервного копіювання, принт-сервери – будь-які з цих елементів можуть страждати від програмних та апаратних збоїв.

Завдання системи моніторингу – це попередження, оскільки перерви у роботі мережі загалом впливають авторитет організації, комерційні організації втрачають заробіток при непрацездатності обчислювальної мережі, а державні організації, втрачають управління підрозділами, отже, непрацездатність обчислювальної мережі може бути прямою загрозою для життя та здоров'я людей.

Тому ці організації використовують різноманітні засоби та продукти для моніторингу.

2.3. Засоби моніторингу мережі

• Системи управління мережею (Network Management Systems) – це централізовані програмні системи, які збирають дані про стан вузлів та комунікаційних пристроїв мережі, а також про трафік, що циркулює у мережі. Ці системи не тільки здійснюють моніторинг та аналіз мережі, але й виконують в автоматичному або напівавтоматичному режимі дії з керування мережею – включення та відключення портів пристроїв, зміна параметрів мостів адресних таблиць, комутаторів та маршрутизаторів тощо. Прикладами систем управління можуть бути популярні системи HP OpenView, SunNet Manager, IBM NetView та ін.

• Засоби управління системою (System Management) часто виконують функції, аналогічні функцій систем управління, але стосовно інших об'єктів. У першому випадку об'єктом управління є програмне та апаратне забезпечення комп'ютерів мережі, а у другому – комунікаційне обладнання. Разом з тим деякі функції цих двох видів систем управління можуть дублюватися, наприклад засоби управління системою можуть виконувати найпростіший аналіз мережевого трафіку.

• Вбудовані системи діагностики та управління (Embedded Systems) – ці системи виконані у вигляді програмно-апаратних модулів, які встановлюються у комунікаційне обладнання, а також у вигляді програмних модулів, вбудованих в операційні системи. Вони виконують функції діагностики та управління лише одним пристроєм, і в цьому їхня основна відмінність від централізованих систем управління. Прикладом засобів цього класу може бути модуль управління концентратором Distributed 5000, що реалізує функції автосегментації портів при виявленні несправностей, приписування портів внутрішнім сегментам концентратора та деякі інші. Як правило, вбудовані модулі управління «за сумісництвом» виконують роль SNMP-агентів, що постачають дані про стан пристрою для систем управління.

• Аналізатори протоколів (Protocol Analyzers) є програмні чи апаратно-програмні системи, які обмежуються, на відміну систем управління, лише

функціями моніторингу та аналізу трафіку в мережах. Хороший аналізатор протоколів може захоплювати і декодувати пакети великої кількості застосовуваних мереж протоколів – зазвичай кілька десятків. Аналізатори протоколів дозволяють встановити деякі логічні умови для захоплення окремих пакетів і виконують повне декодування захоплених пакетів, тобто в зручній для фахівця формі вкладеність пакетів протоколів різних рівнів один в одного з розшифровкою змісту окремих полів кожного пакета.

• Експертні системи акумулюють людські знання про виявлення причин аномальної роботи мереж та можливі способи приведення мережі в працездатний стан. Експертні системи часто реалізуються як окремих підсистем різних засобів моніторингу та аналізу мереж: систем управління мережами, аналізаторів протоколів, мережевих аналізаторів. Найпростішим варіантом експертної системи є контекстно-залежна help-система. Найскладніші експертні системи є так звані основи знань, які мають елементи штучного інтелекту.

• Багатофункціональні пристрої аналізу та діагностики. В останні роки у зв'язку з поширенням обчислювальних мереж виникла необхідність розробки недорогих портативних приладів, що поєднують функції декількох пристроїв: аналізаторів протоколів, кабельних сканерів і навіть деяких можливостей програмного забезпечення мережного управління.

2.4. Напрямки розвитку систем моніторингу

Ще одним із важливих завдань моніторингу є контроль за безпекою обчислювальної мережі. Всі перераховані засоби та продукти моніторингу контролюють роботу елементів обчислювальної мережі, але не контролюють безпеку трафіку, що проходить, і ступінь захищеності елементів мережі та всієї мережі в цілому.

Безпека комп'ютерної мережі (у сенсі захищеності від шкідливих дій) забезпечується двома методами: аудитом і контролем. Аудит безпеки – перевірка налаштування мережі (відкритих портів, доступності «внутрішніх» програм ззовні,

надійності автентифікації користувачів). Сутність контролю безпеки полягає у виявленні аномальних подій у функціонуванні мережі та контролює:

- навантаження на серверне ПЗ та «залізо»: аномально високі рівні завантаження процесора, раптове скорочення вільного місця на дисках, різке збільшення мережного трафіку найчастіше є ознаками мережевої атаки;

- журнали та звіти на наявність помилок: окремі повідомлення про помилки в лог-файлах програм-серверів або журнал подій серверної операційної системи допустимі, але накопичення та аналіз таких повідомлень допомагає виявити несподівано часті або систематичні відмови;

- стан потенційно вразливих об'єктів – наприклад, тих, «захищеність» яких важко проконтролювати безпосередньо (ненадійне стороннє ПЗ, що змінилася/ неперевірена конфігурація мережі): небажані зміни прав доступу до деякого ресурсу або вмісту файлу можуть свідчити про проникнення «ворога».

2.5. Протоколи керування мережею

Розділимо засоби моніторингу та аналізу обчислювальних мереж на декілька крупних класів:

Системи управління мережею (Network Management Systems) - централізовані програмні системи, які збирають дані про стан вузлів і комунікаційних пристроїв мережі, а також дані про трафік, циркулюючий в мережі. Ці системи не тільки здійснюють моніторинг і аналіз мережі, але і виконують в автоматичному або напівавтоматичному режимі дії по управлінню мережею - включення і відключення портів пристроїв, зміна параметрів мостів адресних таблиць мостів, комутаторів і маршрутизаторів і т.п. Прикладами систем управління можуть служити популярні системи HPOpenView, SunNetManager, IBMNetView.

Засоби управління системою (System Management). Засоби управління системою часто виконують функції, аналогічні функціям систем управління, але по відношенню до інших об'єктів. В першому випадку об'єктом управління є

програмне і апаратне забезпечення комп'ютерів мережі, а в другому - комунікаційне устаткування. Разом з тим, деякі функції цих двох видів систем управління можуть дублюватися, наприклад, засоби управління системою можуть виконувати найпростіший аналіз мережного трафіку.

Вбудовані системи діагностики і управління (Embedded systems). Ці системи виконуються у вигляді програмно-апаратних модулів, встановлюваних в комунікаційне устаткування, а також у вигляді програмних модулів, вбудованих в операційні системи. Вони виконують функції діагностики і управління тільки одним пристроєм, і в цьому їх основна відмінність від централізованих систем управління. Прикладом засобів цього класу може служити модуль управління концентратором Distrebuted 5000, реалізовуючий функції автосегментації портів при виявленні несправностей, приписування портів внутрішнім сегментам концентратора і деякі інші. Як правило, вбудовані модулі управління «за сумісництвом» виконують роль SNMP-агентів, що поставляють дані про стан пристрою для систем управління.

Аналізатори протоколів (Protocol analyzers). Є програмними або апаратно-програмними системами, які обмежуються на відміну від систем управління лише функціями моніторингу і аналізу трафіку в мережах. Хороший аналізатор протоколів може захоплювати і декодувати пакети великої кількості протоколів, вживаних в мережах - звичайно декілька десятків. Аналізатори протоколів дозволяють встановити деякі логічні умови для захоплення окремих пакетів і виконують повне декодування захоплених пакетів, тобто показують в зручній для фахівця формі вкладеність пакетів протоколів різних рівнів один в одного з розшифровкою змісту окремих полів кожного пакету.

Устаткування для діагностики і сертифікації кабельних систем. Умовно це устаткування можна поділити на чотири основні групи: мережні монітори, прилади для сертифікації кабельних систем, кабельні сканери і тестери (мультиметри).

- Мережеві монітори (звані також мережевими аналізаторами) призначені для тестування кабелів різних категорій. Слід розрізняти мережні монітори і аналізатори протоколів. Мережні монітори збирають дані тільки про статистичні

показники трафіку - середньої інтенсивності загального трафіку мережі, середньої інтенсивності потоку пакетів з певним типом помилки і т.п.

- Призначення пристроїв для сертифікації кабельних систем, безпосередньо виходить з їх назви. Сертифікація виконується відповідно до вимог одного з міжнародних стандартів на кабельні системи.

- Кабельні сканери використовуються для діагностики мідних кабельних систем.

- Тестери призначені для перевірки кабелів на відсутність фізичного розриву.

- Експертні системи, дані системи акумулюють виявлення причин аномальної роботи мереж і можливі способи приведення мережі в працездатний стан. Експертні системи часто реалізуються у вигляді окремих підсистем різних засобів моніторингу і аналізу мереж: систем управління мережами, аналізаторів протоколів, мережних аналізаторів. Найпростішим варіантом експертної системи є контекстно-залежна help-система. Складніші експертні системи є так званими базами знань, що володіють елементами штучного інтелекту. Прикладом такої системи є експертна система, вбудована в систему управління Spectrum компанії Cabletron.

- Багатофункціональні пристрої аналізу та діагностики. Останніми роками, у зв'язку з повсюдним розповсюдженням локальних мереж виникла необхідність розробки недорогих портативних приладів, що суміщають функції декількох пристроїв: аналізаторів протоколів, кабельних сканерів і, навіть, деяких можливостей ПО мережевого управління.

2.6. Протокол SNMP

Всю інформацію протокол SNMP отримує з бази керуючої інформації (ManagementInformationBase, MIB). MIB представляє собою базу даних стандартизованої структури. База даних має деревоподібну структуру, а всі змінні класифіковані за тематикою. Кожне піддерево містить певну тематичну підгрупу

змінних. Найбільш важливі компоненти, що відповідають за роботу мережевих вузлів, об'єднані в підгрупі MIB-II.

Існують два типи MIB: стандартні і фірмові. Стандартні MIB визначені комісією з діяльності Інтернет (Internet Activity Board, IAB), а фірмові - виробником пристрою. У базах даних, зазначених у таблиці 1, присутня безліч змінних, які можуть бути корисні для діагностування мережі і мережевих пристроїв.

У MIB кожен об'єкт має ім'я і тип. Назва об'єкту характеризує його становище в дереві MIB. При цьому ім'я дочірнього вузла включає в себе ім'я батьківського вузла і задається цілим числом.

SNMP - протокол прикладного рівня. Він призначений для обміну інформацією між мережевими пристроями. За допомогою цього протоколу, мережевий адміністратор може виробляти аналіз мережевого устаткування, знаходити і вирішувати безліч мережевих проблем.

З виходом SNMPv 3, користувачам стали доступні нові служби, такі як: обмеження доступу, захист даних і аутентифікація користувача.

Крім цього, варто відзначити, що SNMPv 3 перейняв модульну архітектуру від своїх попередників. Це забезпечує підтримку попередніх версій SNMP і, не дивлячись на те, що SNMPv 1 і SNMPv 2 не підтримують аутентифікацію і шифрування, у Вас буде можливість керування пристроями, які підтримують ці версії.

У SNMPv 3 вже не застосовуються терміни «агент» і «менеджер», тепер використовуються терміни «сутності». Як і раніше одна сутність знаходиться на керованому пристрої, а друга займається опитуванням додатків.

У сутностей-агентів і сутностей-менеджерів тепер є ядро, яке виконує чотири основні функції:

1. функції диспетчера;
2. обробка повідомлень;
3. функції безпеки;
4. контроль доступу.

Диспетчер - це проста система управління вхідним і вихідним трафіком. Для кожного вихідного блоку даних (PDU) він визначає тип необхідної обробки (SNMPv 1, SNMPv 2, SNMPv 3) і передає блок даних відповідного модуля в системі обробки повідомлень.

Система обробки повідомлень отримує від Диспетчера вихідні блоки даних (PDU), додає до них відповідний заголовок і повертає їх назад Диспетчеру.

Система безпеки відповідає за шифрування і аутентифікацію. Всі вихідні повідомлення перед відправкою спочатку передаються із системи обробки повідомлень в систему безпеки, де всі шифруються поля в заголовку повідомлення, блок даних (PDU), генерується код аутентифікації і додається до заголовку повідомлення. Після цього повідомлення передається назад у систему обробки повідомлень. Точно така ж операція, але в зворотному порядку проводиться для всіх вхідних повідомлень.

Система контролю доступу управляє службами аутентифікації для контролю доступу до MIB виходячи з вмісту блоків даних.

2.6.1. Функції компонентів SNMP

Стандарт SNMP створено для вирішення задач обробки помилок та аналізу продуктивності і надійності.

Обробка помилок. Виявлення, визначення і усунення наслідків збоїв і відмов у роботі мережі. На цьому рівні виконується реєстрація повідомлень про помилки, їх фільтрація, маршрутизація і аналіз на основі деякої кореляційної моделі.

Аналіз продуктивності і надійності. Оцінка на основі статистичної інформації таких параметрів, як час реакції системи, пропускна спроможність каналів зв'язку, інтенсивність трафіку в окремих сегментах мережі, імовірність спотворення даних, коефіцієнт готовності служб мережі.

Результати такого аналізу дозволяють контролювати угоду про рівень обслуговування (Service Level Agreement, SLA).

Згідно з SNMP, управління повинне бути простим, нехай навіть ціною втрати потужності, масштабованості і захищеності. Тому при розробці стандартів SNMP враховувалися наступні умови:

• Широка сфера застосування. Системи під управлінням SNMP можуть бути будь-якими і можуть бути скрізь: від принтерів до мейнфреймів;

• Простота додавання керуючих функцій. Керована система обмежена в функціональності управління, дуже проста і не може контролювати себе. Замість цього всі керовані системи контролює складна керуюча система, функціональність якої можна розширювати;

• Стійкість у критичних ситуаціях. Наприклад, при перевантаженні і проблемах в мережі, тобто при множинних помилках.

Архітектуру розподіленої системи можна описати в термінах обробних елементів (або компонентів), що з'єднують елементів (або з'єднувачів) і елементів даних. Перерахуємо складові елементи системи управління SNMP:

1. компоненти:

• агент;

• менеджер;

2. з'єднувачі:

• транспортний протокол;

• Протокольні блоки даних (ProtocolDataUnits, PDU) і повідомлення SNMP;

3. дані:

• керуюча інформація MIB;

Проаналізуємо архітектуру SNMP з позиції досягнення поставлених перед SNMP цілей. Для цього використовуємо поняття архітектурного стилю мережевого програмного забезпечення. Архітектурний стиль - це узгоджений набір архітектурних обмежень, накладених на ролі і особливості архітектурних елементів (компонентів, з'єднувачів і даних) і відносин між ними, яка проявляється у будь-якій архітектурі, і яка задовольняє цьому стилю.

Архітектура SNMP передбачає побудову системи управління за схемою «менеджер-агент», тобто використання архітектурного стилю «клієнтсервер».

Система SNMP містить безліч керованих вузлів, на кожному з яких розміщується досить простий сервер - агент SNMP, а також, принаймні, один вузол, що містить складного клієнта - менеджера SNMP.

Менеджер взаємодіє з агентами за допомогою протоколу SNMP з метою обміну керуючою інформацією. В основному, ця взаємодія реалізується у вигляді періодичного опитування менеджером множини агентів, так як агенти всього лише надають доступ до інформації, але не знають, що їм з нею робити. Видно, що система, побудована за такими принципами, втрачає масштабованость, оскільки є виділений клієнт, який займається опитуванням всіх серверів. Зате така схема забезпечує простоту реалізації систем під управлінням SNMP.

Для підвищення масштабованості та адміністративної керованості вводиться поняття проксі-агента, який може переправляти операції протоколу SNMP, а також поняття менеджера проміжного рівня, який приховує несуттєві подробиці керуючої інформації від систем управління мережами верхнього рівня, інтегруючи одержувані від агентів дані. Це дозволяє створювати багаторівневі системи управління, що відповідають архітектурному стилю «багаторівневий клієнт-сервер».

Більш детальна класифікація компонентів:

1. Менеджер:

• Менеджер проміжного рівня;

• Система управління мережами;

2. Агент:

• Мінімальний агент;

• Проксі-агент;

Тепер розглянемо дані, якими маніпулюють системи SNMP, тобто керуючу інформацію. У SNMP кожний керований пристрій, на якому розташований агент, представляє свою керуючу інформацію у вигляді змінних. Такими змінними можуть бути: ім'я системи, час з моменту її перезапуску, записи в таблиці маршрутизації і т. д. В загальному випадку змінні можна розділити на: скалярні змінні та таблиці змінних.

Схема даних описується структурою керуючої інформації (Structure of Management Information, SMI). Схема даних визначається, як виглядає керуюча інформація, тобто описує її синтаксис. SMI базується на Abstract Syntax Notation One. Конкретні набори керуючої інформації для різних типів пристроїв, протоколів і т. д. описуються базами керуючої інформації (Management Information Bases, MIBs). Бази MIB визначають, яка управляюча інформація існує. Наприклад, для пристрою, що підтримує IP, MIB описує таблицю маршрутизації, прапорець активації функції маршрутизації, число переданих і прийнятих пакетів, число помилок різного характеру і т. д. Таким чином, кожен пристрій містить набір значень змінних, визначених у деякій кількості MIB, описаних за правилами SMI. Цей набір змінних і є даними, що управляє інформацією для протоколу SNMP.

Важливим питанням є іменування змінних. У SNMP кожній змінній присвоюється унікальний ідентифікатор об'єкта (Object Identifier, OID).

Простір імен OID є ієрархічним і контролюється організацією по розподілу номерів в Інтернеті (Internet Assigned Numbers Authority, IANA). Кожен компонент імені є числом. В текстовому вигляді імена записуються як десяткові числа, розділені крапками, зліва направо. Числам можуть бути поставлені у відповідність текстові рядки для зручності сприйняття. У цілому, структура імені схожа на систему доменних імен Інтернету (Domain Name System, DNS).

MIB визначає набір змінних, тобто певну гілку дерева OID, що описує керуючу інформацію в певній галузі. Наприклад, гілка 1.3.6.1.2.1.1 описує загальну інформацію про систему. Опишемо деякі змінні з цієї гілки:

• sysDescr (1.3.6.1.2.1.1.1) - короткий опис системи;

• sysUpTime (1.3.6.1.2.1.1.3) - час з моменту останнього перезапуску;

• sysName (1.3.6.1.2.1.1.5) - назва системи.

Змінні і відомості про їхній тип визначені також в MIB. А самі типи змінних - в SMI.

Крім безпосередньо даних, необхідно ввести операції над ними. Набір цих операцій змінювався і розширювався в міру розвитку SNMP. Основними операціями є:

- зчитування змінної;

- запис змінної;

- зчитування змінної, наступної за заданою змінною (необхідне для перегляду таблиць змінних).

Операції над даними в SNMP схожі на віддалене налагодження деякої програми: стан системи описується певним набором змінних, які можна переглядати та змінювати.

РОЗДІЛ 3. ЗАГАЛЬНИЙ МЕТОД ІНСТРУМЕНТУ ZABBIX ДЛЯ МОНІТОРИНГУ КОМП'ЮТЕРНОЇ МЕРЕЖІ

3.1. Загальні відомості про систему моніторингу Zabbix

Zabbix – це універсальний інструмент моніторингу, здатний відслідковувати динаміку роботи серверів та мережевого обладнання, швидко реагувати на позаштатні ситуації та попереджати можливі проблеми із навантаженням. Система моніторингу Zabbix може збирати статистику у зазначеному робочому середовищі та діяти у певних випадках заданим чином.

Zabbix – це абсолютно безкоштовний інструмент моніторингу мережі. Немає обмежень у можливостях та кількості контрольованих пристроїв. Офіційно дозволено вносити зміни на рівні вихідного коду. Крім того, Zabbix підтримує будьякий розмір мережевої установки: це може бути невелика мережа чи архітектура на рівні підприємства.

Zabbix був створений у 1998 році. В той час на ринку було лише двоє гравців: HP Open View та IBM BMC. Однак рішення, які вимагалися були дуже дорогими та занадто складними для обслуговування та налаштування інструментів для моніторингу з відкритим кодом.

3.1.1. Архітектура Zabbix

Zabbix має 4 основні інструменти, за допомогою яких можна моніторити певне робоче середовище і збирати про нього повний пакет даних для оптимізації роботи.

• Сервер – ядро, що містить у собі всі ці системи, включаючи статистичні, оперативні та конфігурацію. Дистанційно керує мережевими сервісами, повідомляє адміністратора про існуючі проблеми з обладнанням, що перебуває під наглядом.

• Проксі — сервіс, що збирає дані про доступність та продуктивність пристроїв, який працює від імені сервера. Усі зібрані дані зберігаються у буфер і

завантажуються на сервер. Потрібен для розподілу навантаження на сервер. Завдяки цьому можна зменшити навантаження на процесор і жорсткий диск. Для роботи проксі Zabbix окремо потрібна база даних.

• Агент - програма (демон), яка активно моніторить та збирає статистику роботи локальних ресурсів (накопичувачі, оперативна пам'ять, процесор та ін.) та додатків.

• Веб-інтерфейс є частиною сервера системи і вимагає для роботи веб-сервер. Часто запускається тому ж фізичному вузлі, як і Zabbix.

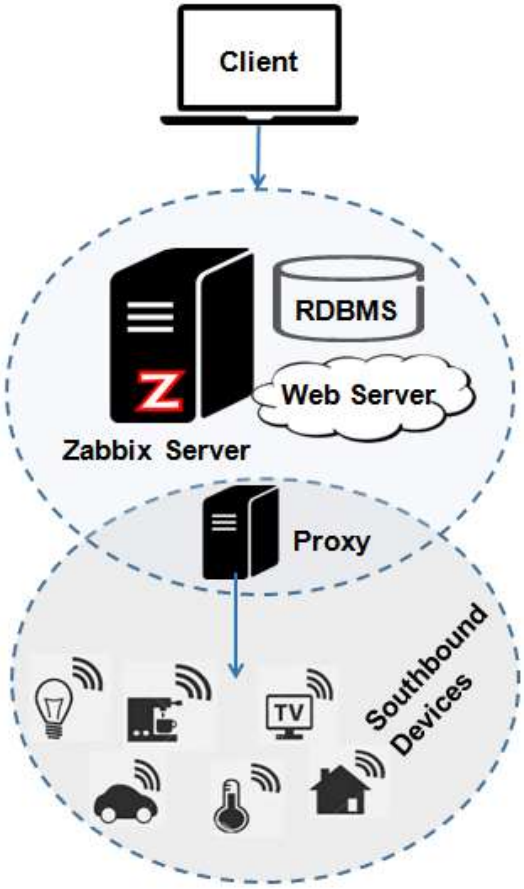

Рис. 3.1 – архітектура Zabbix

Загалом, комбінація цих компонентів дозволяє Zabbix підтримувати три типи моніторингу: проста перевірка, агент Zabbix та зовнішня перевірка. Проста перевірка перевіряє наявність різних служб, таких як SMTP або HTTP, без додаткових установок на хості. Агент Zabbix локально контролює робоче навантаження обладнання. Зовнішня перевірка здійснює віддалений моніторинг за допомогою SNMP, TCP та ICMP через IPMI, SSH.

Хотілось би зазначити, що моніторинг хосту також може здійснюватися без проксі. У цьому випадку всі дані моніторингу з хосту збиратимуться безпосередньо сервером Zabbix. Крім того, графічний інтерфейс Zabbix, сервер Zabbix, медіасервер та база даних можуть бути встановлені в одній машині. Такий метод важливий для малих та середніх мереж.

3.1.2. Переваги Zabbix

Zabbix задовольняє вимоги надійного інструменту моніторингу комп'ютерної мережі приблизно на 90 відсотків. Він здійснює як моніторинг на основі агентів, так і без агентів. Можна знайти такі функції, як виявлення низького рівня, автоматичне виявлення та логічне групування. Усі вищезазначені функції роблять Zabbix надійним інструментом моніторингу мережі, який повністю задовольняє вимоги мережі будь-якого розміру. Однак Zabbix не підтримує прогнозування тенденцій. Ця функція не була включена командою Zabbix, оскільки вона знижує загальну ефективність. Zabbix - надійний та передбачуваний інструмент моніторингу мережі. Якщо Zabbix попередить користувача про деякі помилки, то він може бути на 100 відсотків впевнений, що така проблема існує. Ті самі принципи надійності застосовуються до відновлення та візуалізації.

Крім того, однією з головних переваг Zabbix є його масштабованість, оскільки вона застосовна для середовищ будь-якого розміру. Принцип масштабованості застосовується до продуктивності та зручності використання інтерфейсу. Однак можливості Zabbix не обмежуються лише у IT. Zabbix як інструмент моніторингу мережі можна порівняти з мозком, який отримує поточну інформацію: введення від датчиків, цілих чисел даних та потокових файлів. Тригери аналізують усі ці дані. Коли створюється вихідний результат із тригерів, результати можуть бути різними. Це може бути мак-адреса пристрою, температура процесора або навіть попередження або команда для запуску автономного сценарію.

3.2. Прикладний програмний інтерфейс

Прикладний програмний інтерфейс (API) може бути надзвичайно повільним, особливо коли мова йде про операції, пов’язані із зв’язуванням шаблонів. Наприклад, існує 10 000 хостів, і менеджер мережі хотів би пов’язати їх із простим шаблоном. Це займе приблизно 10-20 хвилин, залежно від обладнання. Крім того, це створить занадто багато запитів SQL. Їх кількість може сягати навіть мільйонів. Інша проблема полягає в тому, що немає суворої перевірки та слабкого звітування про помилки. Наприклад, користувач допустив помилку під час набору виклику API, що перетворилося на загальну помилку. В результаті користувач не знає, яка саме помилка сталася з API. На даний момент API знаходиться на інтерфейсній частині. Планується перемістити API на сторону сервера Zabbix. В результаті це призведе до значного поліпшення продуктивності.

Безпека Zabbix. Головним методом у Zabbix, є шифрування. Zabbix не підтримує шифрування та автентифікацію на пряму. В результаті користувачі повинні застосовувати сторонні інструменти, наприклад Stunnel та Open VPN. Однак вони не належним чином інтегровані із Zabbix, і їх важко підтримувати, особливо для великих середовищ.

Причина полягає в тому, що шифрування є складним процесом, оскільки воно має генерувати сертифікати та ключі. Ось чому команда Zabbix хотіла б запровадити функції захисту, які можна було б легко застосувати для будь-якої мережі, тобто використання відкритих ключів шифрування, таких як SSL або TLS для агентів. Причиною цього є те, що це може спричинити небажаний слід у мережі.

Зв’язок між агентами вже досить великий для роботи мережі. Якщо додати речі, пов’язані з шифруванням, це негативно вплине на роботу мережі та сервера Zabbix.

Zabbix на рівні коду. Перший прототип Zabbix був випущений як проект корпоративного банку. Однак, щоб бути успішним комерційним продуктом, потрібно було змінити всю архітектуру. В результаті було обрано таку структуру.

Мова С використовувалася для всіх критичних частин, таких як сторона сервера, сторона агента та сторона проксі. Для інтерфейсу було обрано PHP. Він використовувався для візуалізації та веб-інтерфейсу. Для бази даних було обрано SQL. Структура з’явилася в Zabbix та стала базовою для всіх наступних версій.

Zabbix та мова програмування С. Мова С - мова низького рівня. Завдяки хорошим навичкам програмування можна створити ефективний код. Ось чому Zabbix швидкий і не вимагає великих обчислювальних ресурсів. Крім того, мова С дозволяє створювати додаток без будь-яких залежностей. Однак під час розробки додатка на мові С слід враховувати такі функції: управління пам’яттю, журнали та спільні ресурси. Як результат, це уповільнює швидкість.

PHP у Zabbix. PHP - мова високого рівня. Його головна перевага в тому, що він доступний для всіх платформ. На відміну від, PHP має деякі недоліки. Це динамічно набрана мова. Часто, коли змінну визначають як масив. Однак наступним рядком може бути ціле число, а наступний рядок може перетворитися на об’єкт. В результаті це створює всілякі проблеми. Крім того, через той факт, що PHP є інтерпретованою мовою, помилка, як правило, виникає під час виконання. Іншими словами, немає компіляції, де можна протестувати додаток.

SQL в Zabbix. Під час розробки самої першої версії, вибір був між SQL та Round Robin. Найбільшим недоліком Round Robin було те, що він агрегує інформацію в базі даних, і існує необхідність встановити правило агрегування заздалегідь. Коли вимоги змінюються, доступ до вихідних даних більше не існує. Ось чому для Zabbix було обрано SQL. SQL - це механізм зберігання транзакцій. Це означає, що можна зробити величезну кількість змін, і все одно структура буде атомною. Як результат, це забезпечує узгодженість обмежень на рівні бази даних. Сам механізм перевіряє, коли дані суперечливі. Більше того, SQL має стандарт API. На Zabbix можна запускати MySQL, PostgreSQL, Oracle, DB2 та SQLite.

А якщо дивитись з іншого боку, масштабувати традиційні бази даних SQL досить складно. Масштабування операцій зчитування є відносно простим. Для деяких розширень потрібно внести зміни на рівні програми, але складно для великих операцій.

3.2.1. Переваги архітектури Zabbix

Поєднання мови програмування C, PHP та SQL позитивно впливає на Zabbix. Як результат, Zabbix є досить маленьким додатком, майже не має залежностей. Використання мови C є ключовим компонентом продуктивності Zabbix. Крім того, цей інструмент моніторингу мережі вимагає низького використання ресурсів.

Архітектура Zabbix забезпечує розділення функцій. Наприклад, збір даних відокремлений від інших компонентів, таких як хост, елементи та агенти. Як результат, можна збирати дані, не впливаючи на інші компоненти. Більше того, Zabbix - це багатопроцесний додаток. Такі компоненти, як Zabbix Server, Zabbix Proxy та Zabbix Agent, правильно масштабуються до кількості ядер. Якщо апаратне забезпечення замовника має 32 або навіть 128 ядер, Zabbix масштабує його, щоб отримати всі переваги апаратного забезпечення замовника.

Архітектура Zabbix надає значні переваги загальній продуктивності, все-таки слід зазначити деякі недоліки. У Zabbix використовуються дві різні технології для інтерфейсу. Незважаючи на те, що PHP використовується в інтерфейсі, на (0,1) на мові C. Результат - додаткові виклики та вплив на швидкість. Іноді трапляються ситуації, коли командам Zabbix доводиться створювати дублікати коду як на C, так і на PHP.

Крім того, дублювання коду частково впливає на регресії. Іншим питанням архітектури Zabbix є те, що використовується інтерфейсний PHP, який є динамічно введеною та інтерпретованою мовою. В результаті це спричиняє деякі додаткові проблеми, наприклад, регресії та невдалі проблеми. Теоретично, якщо PHP замінити іншою мовою, весь клас проблем зникне.

3.3. Взаємодія з пам’яттю

Zabbix використовує деякі спеціальні прийоми для підвищення ефективності роботи на рівні пам’яті. У розділі опишемо, як Zabbix може отримувати дані в обхід

бази даних. Крім того, буде представлено, як кілька вкладок поєднуються в одній масовій операції.

3.3.1. Кеш-пам’ять

Zabbix використовує техніку кешування. Кеш - це шар між сервером Zabbix або Zabbix Proxy та базою даних. Хорошим прикладом може бути кеш конфігурації. Для отримання даних із бази даних дзвінки не спрямовуються до бази даних.

Дані конфігурації беруться з кешу. Крім того, Zabbix має кеш значень. Це значно покращує продуктивність обчислювальних ресурсів. Для того, щоб отримати дані для оцінки тригерів, вони беруться безпосередньо з пам'яті, а не здійснюють прямий виклик до бази даних.

3.3.2. Групові операції

Групові операції - це ще одна особливість, яка покращує продуктивність Zabbix - це кеш історії запису. Замість того, щоб робити множинні вставки та оновлення до бази даних, вони поєднуються як одна масова операція. Однак, як техніка готівки, масові операції використовуються лише на задній панелі. Ось чому він реалізований лише на Zabbix Server та Zabbix проксі.

3.4. Веб-інтерфейс Zabbix

Поточна навігація інтерфейсу занадто складна. У користувачів, для яких Zabbix є новимнкою, можуть виникнути проблеми з веб-інтерфейсом. Деякі основні операції можуть зайняти багато часу навіть для досвідчених користувачів. Для базових операцій потрібно занадто багато кліків. Наприклад, адміністратор мережі хоче створити елемент, а після цього тригер. Перш за все, слід створити елемент. Потім користувач повинен повернутися назад і вибрати тригер. Якщо зв’язок втрачено, і користувач не пам’ятає ключ елемента, це створить додаткові

ускладнення. Коли елемент і тригер створені для тестування, користувач повинен перейти на моніторинг. Користувач повинен пам’ятати про те, яким був хост для елемента. В результаті простий процес стає кошмаром для системного адміністратора.

Інша проблема полягає в тому, що інформація від’єднана. Наприклад, елемент та його конфігурації знаходяться в одному місці, а інформацію про моніторинг останніх даних можна знайти в іншому місці. Якщо користувачі хочуть побачити графік то, їм доведеться перейти в інше місце.

3.5. Аналоги Zabbix та їх недоліки

Системи парасолькового моніторингу та управління подіями вивели моніторинг на принципово новий рівень. Порівнювати їх із нішевими рішеннями не зовсім коректно — вони вирішують різні завдання, але перехід до «парасольок» безумовно level-up для компанії. Подібне ПЗ дозволяє спростити роботу технічних фахівців за рахунок створення єдиної системи управління IT-службою, що включає пошук взаємозв'язків між подіями, згенерованими в системах різних типів, і прозору логіку управління, і автоматизацію дій.

3.5.1. Система моніторингу Nagios

Nagios спочатку була створена під ім'ям Netsaint, розроблена Етаном Галстадом. Він же підтримує та розвиває систему сьогодні, спільно з командою розробників, які займаються як офіційними, так і неофіційними плагінами (програмними модулями) для розширення можливостей системи моніторингу.

До переваг Nagios можна віднести:

- велика кількість плагінів, що розширюють базовий функціонал;
- ліцензію GNU;
- використання циклічної бази даних RRD, за даними якої будуються графіки;

- використання демону (програми, що працює у фоновому режимі без прямого спілкування з користувачем) cron–планувальника завдань у UNIX-подібних операційних системах – для періодичного виконання завдань у певний час.

Недоліками є:

- занадто великий інтервал між перевірками та вимірами параметрів;

- перезапуск системи після зміни файлу конфігурації (~10-15 хвилин);

- RRD усереднює дані, тому неможливо сказати, яке було точне значення параметрів, наприклад місяць тому (тобто немає можливості перегляду історії змін кількісних характеристик.).

- відсутність можливості переглядати внутрішні збої системи (наприклад, коли закінчилися діалогові чи фонові процеси).

3.5.2. Система моніторингу Ganglia

Ganglia - масштабована розподілена система моніторингу кластерів паралельних та розподілених обчислень та хмарних систем з ієрархічною структурою. Дозволяє спостерігати статистику та історію (завантаженість процесорів, мережі) обчислень у реальному часі для кожної з машин, що спостерігаються.

На кожній машині запускається демон gmond, який збирає системну інформацію (швидкість процесора, використання пам'яті тощо) та посилає її на певну машину. Машина, яка отримує інформацію, може відображати її, а також передавати деяку узагальнену форму даних вгору ієрархії. Саме завдяки цій ієрархічній схемі Ganglia так добре масштабується. Накладні витрати, пов'язані з роботою gmond, дуже малі, тому цей код можна запускати на всіх машинах кластеру без шкоди для продуктивності.

До недоліків можна віднести відсутність оповіщень в аварійних ситуаціях та управління доступом до системи.

3.5.3. Система моніторингу Cacti

Cacti - відкрита система моніторингу з Web-інтерфейсом, що дозволяє користувачеві опитувати послуги через задані інтервали часу. Cacti збирає статистичні дані за певні часові інтервали та дозволяє відобразити їх у графічному вигляді.

Переважно використовуються стандартні шаблони для відображення статистики із завантаження процесора, виділення оперативної пам'яті, кількості запущених процесів, використання вхідного/вихідного трафіку. Cacti може бути розширена за рахунок скриптів або програм, що виконуються.

Недоліком є відсутність відстеження внутрішніх збоїв.

3.5.4. Система моніторингу Munin

Munin – система моніторингу додатків, яка представляє отримані дані у графіках через Web-інтерфейс. Основний акцент зроблено створення плагінів, число яких налічує вже понад 500. Використовуючи Munin, можна контролювати продуктивність комп'ютерів, мережі, додатків.

Для роботи з даними використовується RRDtool. Munin має архітектуру власник/вузол, коли всі вузли опитуються через рівні проміжки часу. Отримувана інформація зберігається у файлах RRD.

РОЗДІЛ 4. ІНТЕГРАЦІЯ ТА РЕАЛІЗАЦІЯ МОНІТОРИНГУ ZABBIX У КОМП'ЮТЕРНІЙ МЕРЕЖІ

4.1. Встановлення Zabbix Server

Даний розділ буде слугувати для демонстрації теоретичних навичок, здобутих під час навчання, на практиці, а саме, встановлення Zabbix Server поверх Ubuntu Server. Після успішного встановлення цих програмних застосунків ми матимемо змогу увійти до графічного інтерфейсу Zabbix.

Рис. 4.1 – Схема послідовності інсталяції

4.1.1. Встановлення Ubuntu Server

Під час розгляду можливих для встановлення версій Ubuntu мною було обрано Ubuntu Server 14.04.2 LTS. Його буде встановлено як віртуальну машину у віртуальному вікні Oracle. Під час встановлення потрібно виконати додаткові кроки в налаштуванні для подальшого використання у роботі із Zabbix Server:

- Інсталювати Open SHH Server. За допомогою цього ми матимемо змогу підключатися до Ubuntu Server через Putty.

- Призначити IP-адресу Ubuntu Server на інтерфейсі eth1. Обираємо за адресу 192.168.1.10, яка буде слугувати ще й адресою нашого Zabbix Server.

- Останнім кроком є встановлення та налаштування менеджера SNMP на сервері Ubuntu. Це дозволило б серверу Zabbix контролювати пристрої за допомогою SNMP. Для того, щоб це можна було зробити, у командному рядку Linux слід набрати вказаний нижче список команд:

  - sudo apt-get install libsnmp-mib-compiler-perl
  - sudo apt-get install snmp-mibs-downloader

- sudo apt-get install libsnmp-base
- sudo apt-get install libsnmp-dev
- sudo apt- get install snmp
- sudo apt-get install snmpd

Після виконання даних кроків - сервер Ubuntu із SNMP встановлено та налаштовано. Наступним етапом є встановлення та налаштування Zabbix Server.

4.1.2. Встановлення Zabbix Server

Займемось встановленням Zabbix Server поверх Ubuntu Server. Для того, щоб завантажити та встановити сервер Zabbix, вказані команди потрібно ввести у командному рядку сервера Ubuntu.

- sudo wget http://repo.zabbix.com/zabbix/2.4/ubuntu/pool/main/z/zabbix-release/zabbix-elease_2.4-1 + trusty_all.deb
- sudo dpkg -i zabbix-release_2.4-1 + trusty_all.deb
- sudo apt-get

В даному випадку встановлено пакет Zabbix v2.4 який включає такі компоненти, як Zabbix Server, інтерфейс Zabbix (GUI), Zabbix Proxy, Medial Server та Zabbix Database. Крім того, я встановив Zabbix Agent локально на Zabbix Server. Це можна зробити за допомогою цієї команди: sudo apt-get install zabbix-agent zabbix-server-mysql zabbix-frontend-php snmpd php5-mysql php5-curl. Zabbix Agent здійснюватиме локальний моніторинг процесів на сервері Zabbix.

Перш ніж почати користуватися Zabbix, залишається два кроки. Перший - це налаштування часового поясу відповідно до регіону, де встановлений Zabbix Server. Це можна зробити за допомогою команди:

- sudo vi / etc / network / interfaces.

Другим кроком є перезапуск сервера Apache. Це можна зробити, набравши команду: sudo service apache2 restart.Таким чином Zabbix Server було встановлено та налаштовано. Крім того, Zabbix Agent було встановлено локально на Zabbix Server. Відтепер у нас є змога використовувати графічний інтерфейс Zabbix.

4.2. Моніторинг сервера Zabbix

Як було згадано раніше, агент Zabbix був встановлений локально в Zabbix Server, що дало підвищення продуктивності у ньому. Процеси встановлення хосту, елемента та тригера виконались автоматично.

4.2.1. SNMP-моніторинг

Для того, щоб розпочати моніторинг за допомогою протоколу SNMP потрібно виконати три послідовні кроки, а саме: створити хост, елемент та графік.

Рис. 4.2 – послідовність моніторингу SNMP

Хост – це мережевий пристрій, який буде контролювати сам Zabbix. Елемент буде визначати для Zabbix те, що я буду контролювати на хості. Під час створення елементів буде створена низка елементів для моніторингу використання пам’яті, центрального процесору, інтерфейсів та пропускної здатності.

4.2.2. Створення хосту та елементів

Першим кроком є створення хоста. Це мережевий пристрій або послуга, які контролюватимуться через сервер Zabbix. Перший маршрутизатор буде мати ім‘я sw-kpnu-fm, дана назва буде використовуватися в контексті моніторингу та середовища Zabbix загалом. За допомогою імені хоста ми зможемо відрязняти пристрої між собою. sw-kpnu-fm контролюватиметься через SNMP, оскільки на ньому неможливо встановити Zabbix Agent локально. Крім того потрібно вказати IP-адресу пристрою, відповідно до плану логічної мережі IP-адреса маршрутизатора sw-kpnu-fm - 10.10.30.100.

Елемент акумулює та відображає в собі дані від хоста, наприклад процесор, пам’ять і навантаження пропускної здатності. Для моніторингу використання ЦП потрібно створити два елементи. Перший - це простій процесора. Він вимірює обсяг доступних ресурсів. Другий – використання процесора.

Моніторинг процесора на маршрутизаторі здійснювався за допомогою SNMPv2. Інтерфейс хоста призначається автоматично. Ключ створюється вручну, і назва його має бути унікальною. Далі слід призначити номер OID. Це унікальний номер, який використовується для імені конкретного процесу або параметра. Кожен процес має власний унікальний номер OID. Zabbix Server, щоб отримати певний параметр з пристрою, надсилає на цей пристрій номер OID цього параметра. Він відповідає на Zabbix вже результатами параметрів. Номер OID для простою CPU становить 1.3.6.1.4.9.2.1.59.0. Номери OID можна знайти в офіційній документації від виробника пристрою.

Створимо елементи для завантаження процесора. Крім того, елементи пам’яті та пропускної здатності також були реалізовані на sw-kpnu-fm.

4.2.3. Моніторинг за допомогою шаблонів

Ручна конфігурація елементів та графіків - це трудомісткий процес. Для ручного створення елементів потрібна детальна мережева документація яка повина щонайменше містити типи пристроїв, імена та статуси. Крім того, адміністратор мережі повинен мати повний перелік OID SNMP для кожного пристрою. Такі обмеження фактично унеможливлюють масштабованість моніторингу мережу, тому на практиці часто використовуються шаблони.

Шаблон - це набір сутностей, які можна застосувати до будь-якого хоста. Шаблон може складатися з елементів,тригерів та графіків. У Zabbix за замовчуванням є список шаблонів. Однак більше шаблонів можна завантажити з офіційної веб-сторінки Zabbix або навіть створити самому.

Вони мають дві ключові переваги. Перша - як уже згадувалося раніше, шаблони можуть містити елементи за замовчуванням, тригери та графіки. Друга

перевага полягає в тому, що один шаблон можна використовувати для необмеженої кількості хостів. Саме тому призначення параметрів моніторингу для хостів здійснюється одним кліком. Крім того, якщо є необхідність внести деякі зміни у моніторинг, зміни повинні вноситися лише в шаблон. Хости, які використовують цей шаблон, автоматично підтягнуть дані зміни. Як результат, шаблони забезпечують гнучкість моніторингу.

В даному розділі ми налаштуємо шаблон для хосту sw-kpnu-fm. Створений шаблон також буде використовуватися для процесів моніторингу другого хосту - sw-kpnu-fm-29-0. На останньому кроці ми будемо використовувати шаблони для моніторингу процесів на Ubuntu Server.

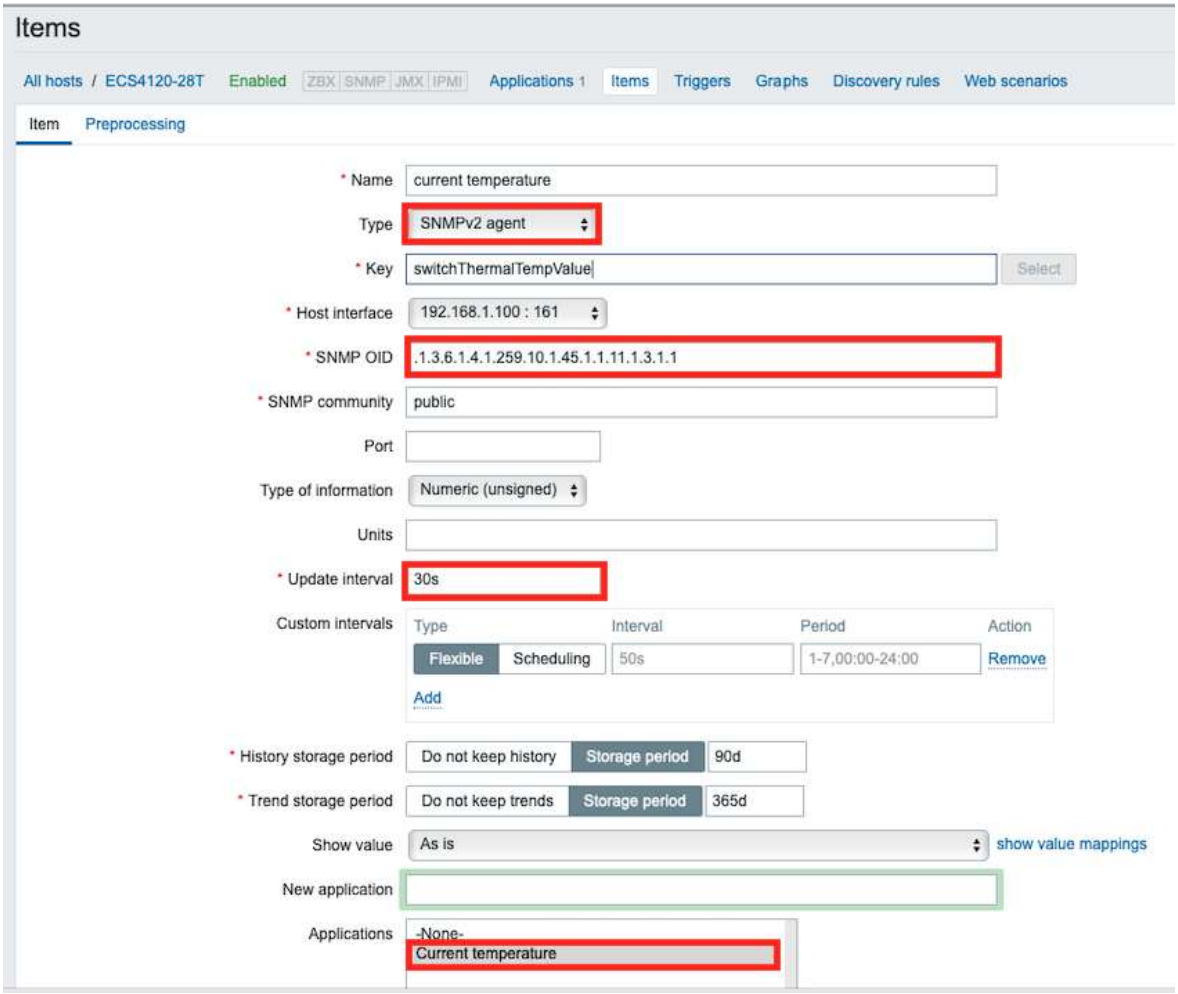

Рис. 4.3 – налаштування шаблону для sw-kpnu-fm

Після того, як ми налаштували SNMP для sw-kpnu-fm я налаштував відображення графіку (дашборду), який буде відображати температуру пристрою.

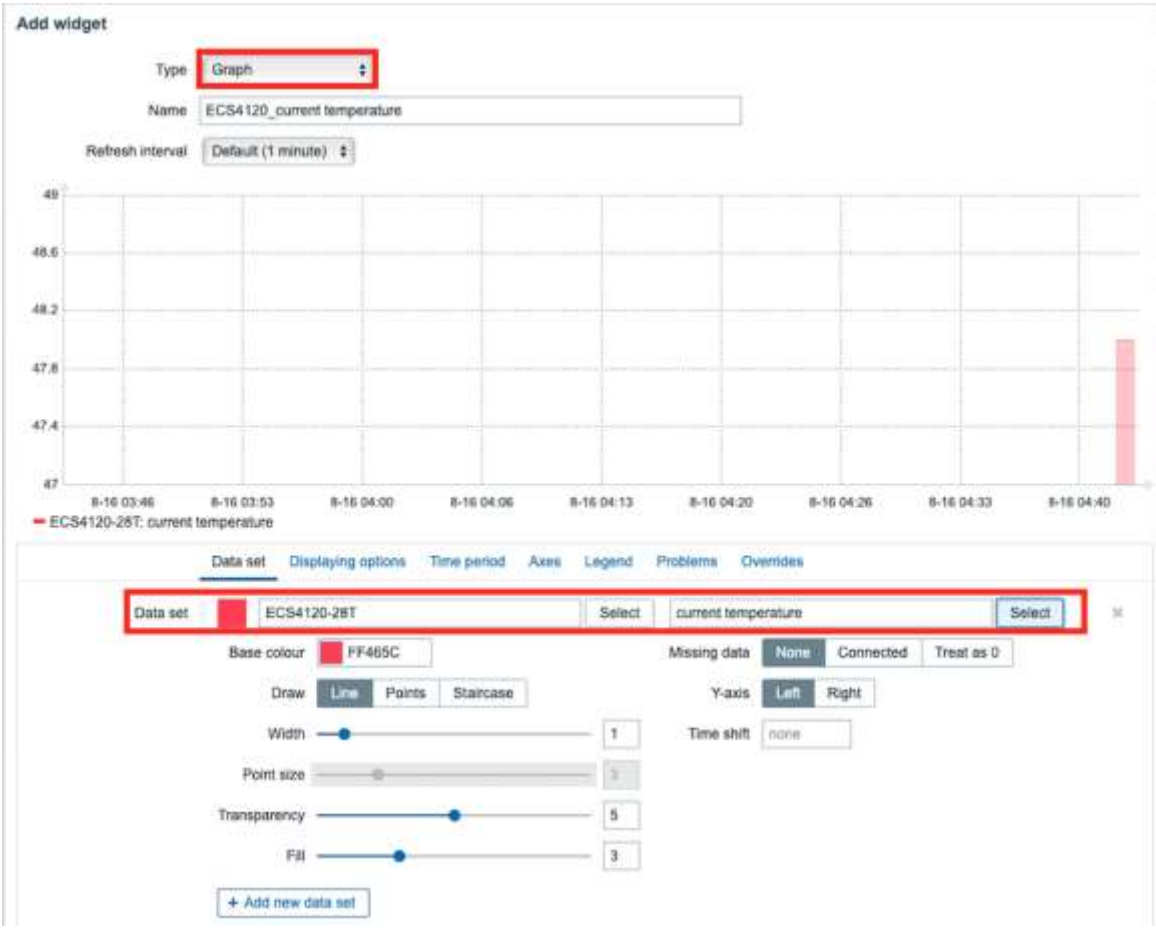

Рис. 4.4 – налаштування відображення графіку для sw-kpnu-fm

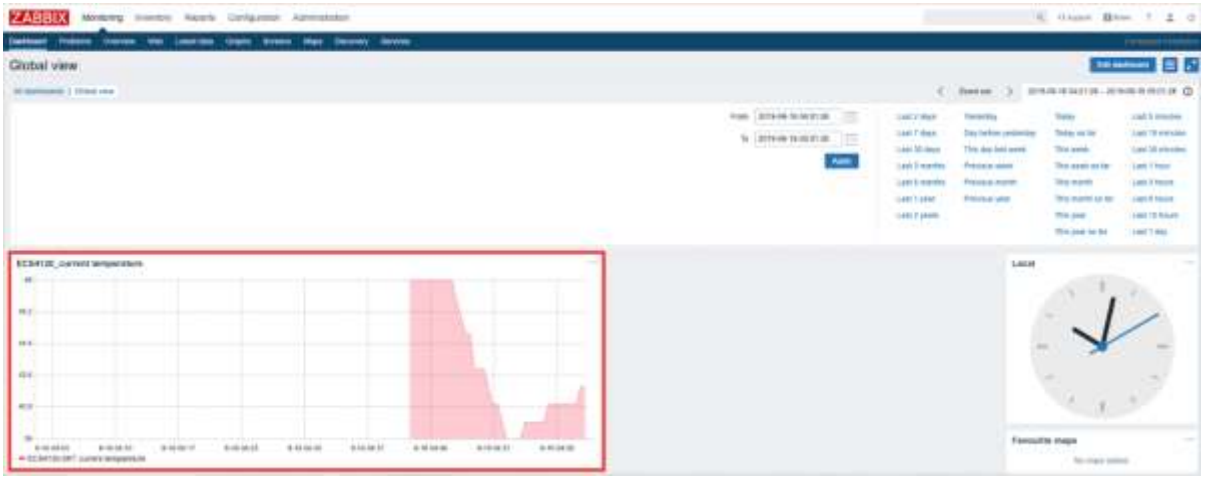

Рис. 4.5 – загальний вигляд графіка з відображенням температури пристрою

4.2.4. Моніторинг маршрутизатора

У даному розділі організуємо моніторинг sw-kpnu-fm за допомогою шаблонів, обравши шаблон пристроїв SNMP. Моніторинг цього шаблону базується

на протоколі SNMP. Оскільки ми використовуємо цей шаблон на sw-kpnu-fm, потрібно вказати спільноту SNMP. Як результат, шість нових елементів були автоматично створені на sw-kpnu-fm. Крім того, також додано графіки та тригери. Однак цей шаблон автоматично додає виявлення та моніторинг інтерфейсів.

Для відслідковування додаткових елементів, наприклад завантаження процесора та використанням пам'яті, можемо скористатись одним із двох способів:

Перший - додати ці елементи на шаблонні пристрої SNMP. Однак це додасть обмежень щодо використання цього шаблону для інших пристроїв. Ось чому ми додаємо ще два шаблони на sw-kpnu-fm.

У списку шаблонів можна знайти шаблонні процесори SNMP та шаблон SNMP пам'ять. Однак вони не містять жодних предметів, тому нам доводиться створювати елементи для обох шаблонів вручну. Перш за все, ми клонували оригінальні шаблони. Тоді потрібно створити елементи так само, як це було зроблено для хоста. Для вимірювання продуктивності процесора потрібно створити два елементи: CPU в режимі очікування та завантаження процесора. Для використання пам’яті потрібно використовувати пам’ять та вільну пам’ять. З таблиці 4 я взяв OID SNMP для необхідних елементів. Слід взяти до уваги, що назва та ключ для нових створених елементів повинні бути унікальними. Коли створюються нові елементи, потрібно вказати спільноту SNMP. Це можна зробити вручну на вкладці макросів. Там необхідно зазначити, що рядок (SNMP_COMMUNITY) який дорівнює значенню вказаному імені. На основі нових елементів можна створювати графіки. Після цього на sw-kpnu-fm можна призначити новий шаблон.

Для моніторингу маршрутизатора sw-kpnu-fm-29-0, першим кроком є його заведення до списку хостів. Наступним кроком є призначення шаблонів, які були використані. Як бачимо один набір шаблонів може використовуватися необмеженою кількістю хостів. Слід враховувати єдине - хости мають бути пристроями одного типу від одного вендору.

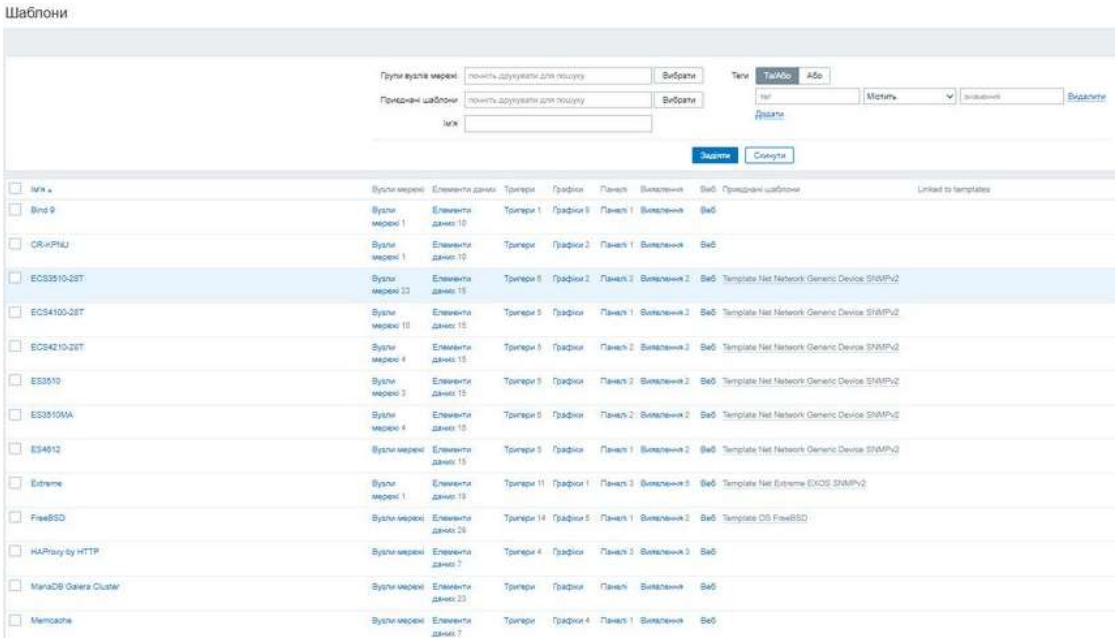

Рис. 4.6 – перелік створених шаблонів для мережі

4.3. Моніторинг серверів Zabbix

Можливості моніторингу Zabbix не обмежується мережевими пристроями, сервери також можна контролювати за допомогою шаблонів. За замовчуванням Zabbix пропонує шаблони для операційних систем на базі Linux та Windows. Моніторинг можна здійснити двома способами: SNMP або Zabbix Agent.

У даному розділі ми відстежуватимемо Ubuntu Server із шаблоном SNMP. На початку деякі конфігурації потрібно робити на сервері Ubuntu. Зазначимо - Ubuntu Server встановлюється як віртуальна машина. Спочатку ми оновлюємо список пакетів Ubuntu Server за допомогою команди: sudo aptget update. Далі потрібно встановити сервер SNMP: sudo apt-get install snmpd snmp. Це дозволить Zabbix Server контролювати Ubuntu Server за допомогою SNMP.

Коли сервер SNMP встановлений, потрібно внести деякі зміни до конфігурації. Редагування потрібно виконати у файлі конфігурації за допомогою цієї команди: sudo vi /etc/snmp/snmpd.conf. Існує необхідність прокоментувати рядок про адресу агента: #agentAddress udp: 127.0.0.1: 161. Це слід зробити через те, що сервер Ubuntu не збирається контролювати через агент Zabbix. Крім того, потрібно додати рядок: agentAddress udp : 161, udp6: [:: 1]: 161. Ця команда вказує

номер порту 161. Це номер порту UDP для SNMP. Останній рядок, який потрібно додати, - це спільнота SNMP: rocommunity. Потрібно перезапустити службу SNMP: sudo service snmpd restart.

Наступним кроком є додавання сервера Linux як хоста в графічному інтерфейсі Zabbix. Іменем хоста було обрано Linux Server 1. Шаблон для моніторингу нового хоста був обраний SNMP OS Linux. У цьому шаблоні потрібно вказати спільноти SNMP, він містить усі необхідні елементи, тому додаткових редагувань робити не потрібно. Шаблон автоматично створює елементи та графіки хоста. Як результат, сервер Linux можна контролювати.

Можемо зробити висновок, що у цьому розділі ми змогли використовувати шаблони для моніторингу процесів на sw-kpnu-fm. Той самий набір шаблонів був використаний на sw-kpnu-fm-29-0.

Крім того, ми мали можливість контролювати ефективність сервера Ubuntu, встановленого як віртуальна машина. Шаблони приносять значну перевагу для мережі середнього та корпоративного рівня. Вони допомагають спростити процес створення елементів для хоста. Для шаблону потрібно налаштувати елементи лише один раз. Після цього елемент може бути легко використаний багатьма хостами. Крім того, якщо потрібні зміни у моніторингу, їх потрібно зробити лише один раз за шаблоном.

Хости, які використовують цей шаблон, будуть автоматично оновлені, замість того, щоб змінювати елементи на кожному хості вручну.

4.4. Топологія мережі

Для мереж на рівні підприємства важливо організувати пристрої в групи. Це забезпечує чіткість та прозорість моніторингу та управління. Можна групувати пристрої на основі географічного положення, за типом пристроїв та їх функціональністю.

На рисунку 4.7 чітко відображається топологія мережі фізико-математичного факультету.

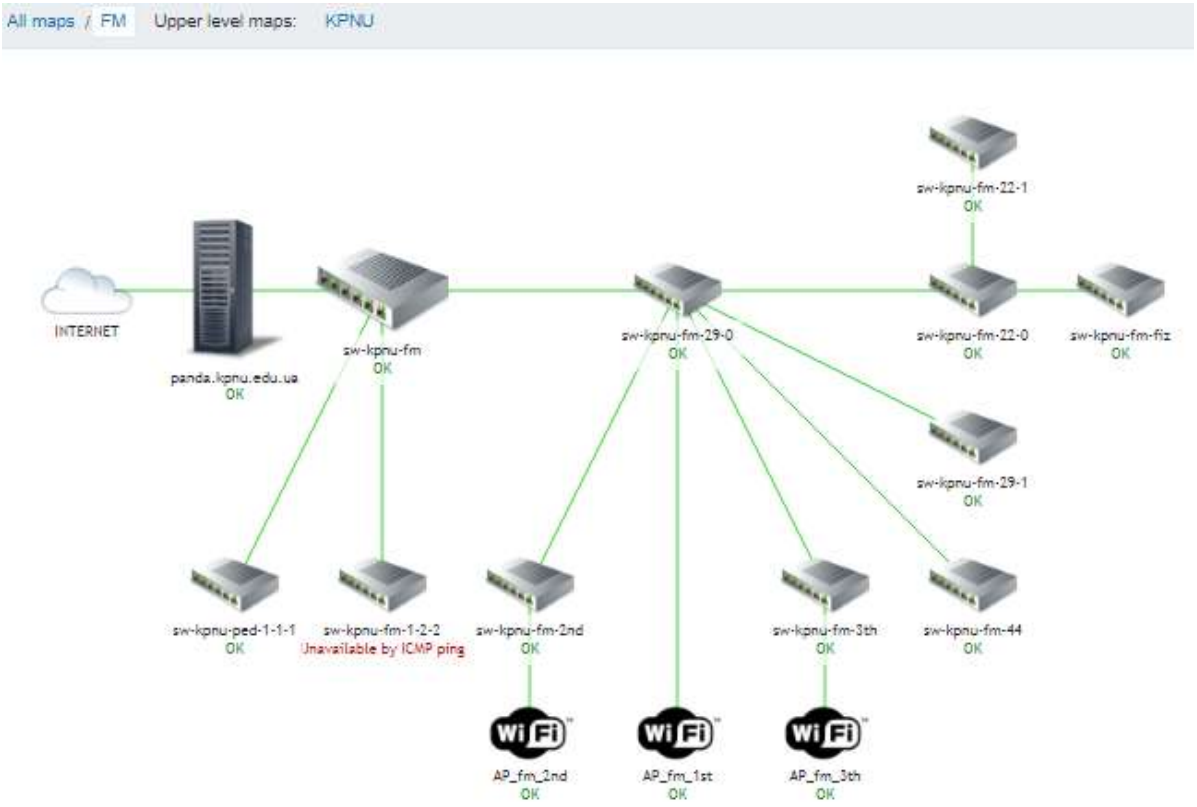

Рис. 4.7 – топологія мережі фізико-математичного факультету

4.5. Технічне рішення моніторингу Zabbix

Для тестування моніторингу Zabbix, необхідно розглянути як він працює:

Рис. 4.8 – Мережа моніторингу Zabbix

Пристрій з одного боку каналу (Client) періодично посилає ряд запитів на пристрій (Server) на іншій стороні каналу, отримує відповіді (або не отримує) і зберігає результати. Запити бувають наступних типів:

- HTTP GET request at a target URL
- HTTP GET request for metadata at a target URL
- ICMP echo request to a target address (the default)

- ICMP timestamp request to a target address
- UDP ping packets to a target device
- UDP timestamp requests to a target address
- TCP ping packets to a target device

Перші два типи запитів відносяться, очевидно, не зовсім до якості каналу, а скоріше до доступності та швидкодії web-сервісів, останні 4 запити є розширеними і вимагають підтримки RPM з боку пристрою з роллю Server. Крім підтримки RPM, для цих тестів так само потрібно і розширена ліцензія. У нашому випадку для ролі Client та ролі Server - використовуєм комутатори ex2200 з базовою ліцензією і роль RPM Server для розширених тестів ми застосовувати не можемо. Тому в цьому розділі обмежимося запитами типу ICMP echo request. Тим більше що це набагато більш універсальний сценарій. Роль Server може виконати абсолютно будь-який мережевий пристрій, який вміє відповідати на ping. Схема тестування зображена на рисунку 4.9.

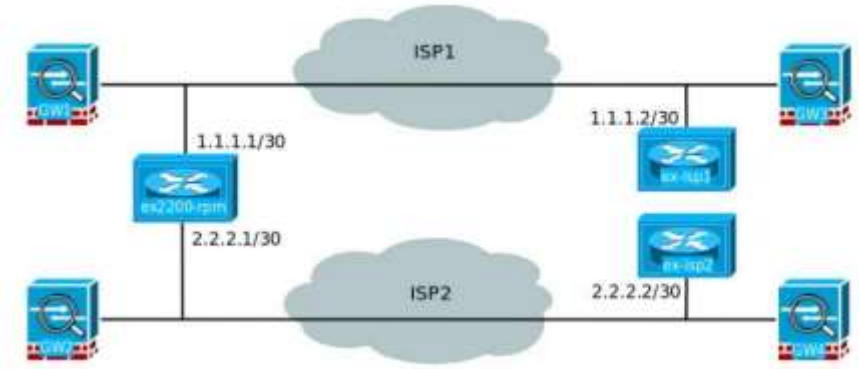

Рис. 4.9 – Схема тестування

Є два L2 канала між офісами від двох операторів зв'язку. Client розміщений зліва. В принципі достатньо було б використовувати по одному пристрою з кожного боку, але так склалося, що до моменту організації тестування пристрою ex-isp1 і ex-isp2 вже використовувалися на цій ділянці мережі.

Тепер можемо перейти до конфігурування RPM. На пристрої ex2200-rpm прописуємо наступну конфігурацію:

```
iddqd@ex2200-rpm> show configuration services rpm
probe Gee {
```

```
test Jitter {
probe-type icmp-ping-timestamp;
target address 2.2.2.2;
probe-count 15;
probe-interval 1;
test-interval 15;
source-address 2.2.2.1;
data-size 1400;
thresholds {
successive-loss 2;
}
hardware-timestamp;
}
}
probe BARS {
test Jitter {
probe-type icmp-ping-timestamp;
target address 1.1.1.2;
probe-count 15;
probe-interval 1;
test-interval 15;
source-address 1.1.1.1;
data-size 1400;
thresholds {
successive-loss 2;
}
hardware-timestamp;
}
}
```

Конфігурація self-explanatory особливих пояснень не потребує. Після чого робимо commit and-quit і через хвилину вже можна збирати результати тестів.

Тепер переходимо до налаштувань Zabbix для моніторингу RPM тестів. Вбудована функціональність SNMP в Zabbix недостатня для автовизначення RPM тестів. Для автовизначення Zabbix використовує метод snmp walk. Де в якості параметрі, які використовуються SNMP індекси та їх значення. Наприклад, для пошуку по об'єкту ifDescr, пишимо наступне:

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB :: ifDescr

IF-MIB :: ifDescr. 4 = STRING: WAN

IF-MIB :: ifDescr.7 = STRING: LAN1

IF-MIB :: ifDescr.11 = STRING: LAN2

Метод discovery в Zabbix виявить індекси 4,7,11 і їх значення WAN, LAN1 і LAN2. А от для виявлення тестів RPM, Juniper не надав такого зручного об'єкта. Найбільш відповідний об'єкт, що вдалося виявити – це об'єкт jnxRpmResSampleValue. Після якого таблиця повернення об’єкта виглядить наступним чином:

iddqd @ ex2200-rpm> show snmp mib walk jnxRpmResSampleValue

jnxRpmResSampleValue.3.71.101.101.6.74.105.116.116.101.114.1 = 1989

jnxRpmResSampleValue.3.71.101.101.6.74.105.116. 116.101.114.2 = -424

jnxRpmResSampleValue.3.71.101.101.6.74.105.116.116.101.114.3 = 810

jnxRpmResSampleValue.4.66.65.82.83.6.74.105.116.116.101.114.1 = 3352

jnxRpmResSampleValue.4.66.65.82.83.6.74.105.116.116.101.114.2 = 1612

jnxRpmResSampleValue.4.66.65.82.83.6.74.105.116.116.101.114.3 = 971

З наведеної таблички видно, що jnxRpmResSampleValue - це MIB об'єкт, по якому ми проходимо, цифри .3.66.101.101.6.74.105.116.116.101.114 - це назва нашого тесту. Крім того, можемо доказати. В якості SNMP index (останнє після точки число) виступає порядковий номер параметрів тесту:

1 - RTT

2 - Round trip jitter

3 - Round trip interarrival jitter

Якщо подивитись на результати RPM тесту та зіставити з числами, які повернув snmpwalk, щоб дізнатися OID (тобто цифрове значення) MIB об'єктів в JunOS таких як jnxRpmResSampleValue, jnxRpmResultsSampleTable, jnxRpmHistorySummaryTable та будь-яких інших, можна запустити команду: show snmp mib walk jnxRpmResSampleValue. Тобто, скрипт авто визначення. Отже, без зовнішньої допомоги Zabbix з виявленням RPM тестів не впорається. Необхідно для зручності надати допомогу у вигляді зовнішнього скрипта, скрипт написаний на Python 2.7 zbx_junper_rpm та використовує всього одну зовнішню бібліотеку – pysnmp. Данна бібліотека присутня у більшості дистрибутивів Ubuntu та її можна поставити командою: apt install python-pysnmp4 чи у будь-якому дистрибутиві через PIP менеджер командою: pip install pysnmp.

На вхід скрипта подаються 2 параметра hostname і community та повертається JSON в наступному вигляді:

```
 {
"data ": [
{
" {#RPMTEST} ":" Jitter ",
" {#RPMUUID} ":" 4.66.65.82.83.6.74.105.116.116.101.114 " ,
"{#RPMOWNER}": "BARS"
}
{
"{#RPMTEST}": "Jitter",
"{#RPMUUID}": "3.71.101.101.6.74.105.116.116.101.114", { {1}}
"{#RPMOWNER}": "Gee"
}
]
}
```

Призначені для користувача макроси {#RPMUUID}, {#RPMOWNER} і {#RPMTEST} далі використовуються в назвах елементів, їх ключах, триггерах і навіть графіках. Скрипту потрібно зробити chmod + x і розмістити в директорію

для зовнішніх скриптів Zabbix у нашому випадку це директорія: / etc / zabbix / etc / externalscripts. Далі не будемо розписувати налаштування Zabbix, а просто переходимо до шаблонів. Шаблон включає в себе 3 елементи: RTT, Jitter і PacketLoss, для кожного RPM тесту, які він виявляє за допомогою скрипта RPM, RPM тест не передбачає вимірювання Jitter, то цей параметр просто автоматично виключається з моніторингу. Для більш просунутих тестів шаблон можна буде доопрацювати за вимогами інфраструктури, яка його використовує. При цьому скрипт не потребуватиме змін, тобто усі три елементи масштабування та виведені на один графік, який так само створюється автоматично.

ВИСНОВКИ

Під час виконання даної роботи було інтегровано програмний застосунок Zabbix у комп’ютерну мережу університету, створено низку шаблонів для моніторингу стану пристроїв, які являються частиною цієї мережі.

Під час роботи з Zabbix я дослідив, що він має низку переваг над іншими застосунками, такі як: відкритий вихідний код, без вкладень у програмне забезпечення, сервер має низькі вимоги щодо продуктивності пристрою, підтримка декількох пристроїв із кількома шаблонами моніторингу, підтримка розподіленого централізованого управління з функцією автоматичного виявлення може реалізувати автоматичний моніторинг, коли елемент, що відстежується, більше, ніж черга з декількох серверів, може бути прийнятий пасивний стан, і відстежуваний клієнт активно завантажує елемент, що відстежується з сервера, а потім вивантажує дані на сервер.

Zabbix задовольняє вимоги надійного інструменту моніторингу комп’ютерної мережі приблизно на 90 відсотків. Він здійснює як моніторинг на основі агентів, так і без агентів. Усі вищезазначені функції роблять Zabbix надійним інструментом моніторингу мережі, який повністю задовольняє вимоги мережі будь-якого розміру.

У підсумку, можна зазначити, що під час виконання даного дипломного проекту усі завдання та цілі, які були поставлені переді мною, як перед виконавцем, були виконаними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zabbix True Open Source. [Електронний ресурс] – Режим доступу до ресурсу: http://www.zabbix.com/true_open_source.php.

2. KELLY, J.: An Examination of Pattern Matching Algorithms for Intrusion Detection Systems. Master's thesis, Ottawa Carleton Institute for Computer Science, Carleton University, Canada, 2006.

3. Проектування пакетів. [Електронний ресурс] – Режим доступу до ресурсу: http://www.networkworld.com/article/2338253/infrastructure-management.

4. Моніторинг мережі. [Електронний ресурс] – Режим доступу до ресурсу: http://www.helpsystems.com/intermapper/network-monitoring.

5. Zabbix. Документація. [Електронний ресурс] – Режим доступу до ресурсу: https://www.zabbix.com/documentation.

6. Автоматичне виявлення на рівні управління мережею та послугами. [Електронний ресурс] – Режим доступу до ресурсу: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1194192& abstractAccess=no&userType=inst.

7. Zabbix автоматичне виявлення. [Електронний ресурс] – Режим доступу до ресурсу: http://www.zabbix.com/auto_discovery.php

8. Моніторинг безпеки мережі Zabbix. [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/215509

9. Network Monitoring Approaches: An Overview [Електронний ресурс] — Режим доступу: (PDF) Network Monitoring Approaches: An Overview (researchgate.net)

10. Rohde & Schwarz (Ed.) (2006). R&S ETX DTV Monitoring Receiver operating manual, 2068.0909.12 – 02. Munich: Rohde & Schwarz.

11. Zabbix. Відкриття низького рівня. [Електронний ресурс] – Режим доступу до ресурсу: http://habrahabr.ru/company/zabbix/blog/203050