УДК 004.7, 658.012.324

І. І. Пастернак
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

ЗАСОБИ ПЕРЕВІРКИ ВУЗЛІВ КОМУНІКАЦІЙНОЇ МЕРЕЖІ КІБЕРФІЗИЧНОЇ СИСТЕМИ

Проаналізовано принципи побудови комунікаційних мереж. Розглянуто переваги та недоліки сучасних засобів реалізації середовищ діагностики вузлів комунікаційних мереж. Запропоновано засоби перевірки надійності вузлів комунікаційної мережі у кіберфізичних системах.

Ключові слова: комунікаційна мережа, клієнт, сервер, кіберфізична система.

I. Pasternak
Lviv Polytechnic National University,
Computer Engineering Department

MEANS OF UNITS VERIFY THE RELIABILITY OF COMMUNICATION NETWORK SYSTEMS CYBER PHYSICAL

Analyzed the principles of building communication networks. The advantages and disadvantages of existing today means implementing diagnostic environments. A reliable means of verification nodes in a communication network cyber physics system.

Key words: communication network, client, server, cyber physics system.

Вступ

Спеціалізоване середовище комунікаційної мережі об’єднує в собі готову комп’ютерну мережу, яка відповідає за комунікаційне середовище, та програмне забезпечення, яке керує обчислювальними засобами, центром збирання та опрацюванням інформації, системою захисту. Для спеціалізованого середовища діагностики комунікаційної мережі потрібно передусім мати готову або спроектувати нову комп’ютерну мережу, яка підтримуватиме концепцію кіберфізичних систем, а саме захищене зберігання і обмін вимірювальною та службовою інформацією, можливість віддаленого моніторингу та налагодження вузлів, дистанційного керування трафіком. Також у цій комунікаційній мережі на вузлах потрібно встановити керовані комутатори, які підтримують протоколи SNMP, оскільки за допомогою цих протоколів відбуватиметься збирання інформації.

У цьому випадку буде здійснюватися перебудова комп’ютерної мережі одного із інтернет-провайдерів міста Львова. Ця мережа поєднує різні технології проектування комп’ютерних мереж – від Fast Ethernet до технології GEPON. Тим самим було продемонстровано на справжньому прикладі, як звичайну комунікаційну мережу можна облаштувати в кіберфізичну систему.

Аналіз останніх джерел та публікацій

Red Hat Enterprise Linux складений переважно з вільного та відкритого програмного забезпечення, але наявний у доступній для вживання, двійковій формі (наприклад, на CD або DVD дисках) лише для передплатних користувачів. Як і вимагається, Red Hat випускає усі вихідні тексти

своїх продуктів під GNU General Public License та іншими вільними ліцензіями. Розробники CentOS використовують цей вихідний код для створення кінцевого продукту, котрий є дуже подібним до Red Hat Enterprise Linux і вільним для завантаження та використання, однак без відповідної технічної підтримки з боку компанії Red Hat.

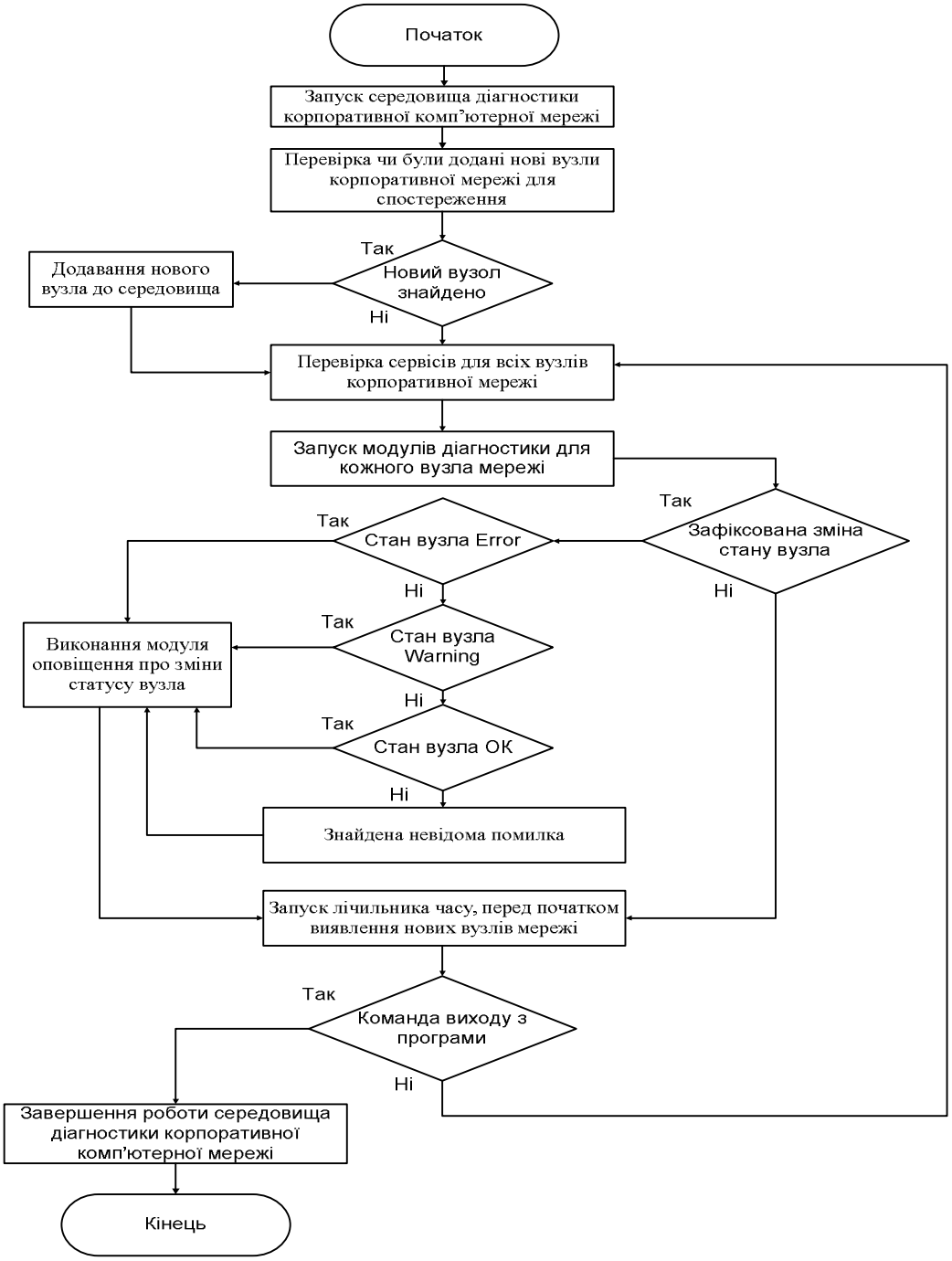

Рис. 1. Схема алгоритму роботи спеціалізованого середовища діагностики КФС

Існують й інші дистрибутиви, основані на вихідних текстах Red Hat Enterprise Linux, однак жоден з них не досяг такого рівня спільноти; загалом CentOS – єдиний дистрибутив, який йде у ногу зі змінами, що вносяться до Red Hat Enterprise Linux. CentOS віддав перевагу програмному забезпеченню для оновлення на основі yum, хоча підтримка up2date також є. Можна використовувати для завантаження та встановлення як додаткові пакунки і залежності, так і спеціальні та періодичні оновлення безпеки з репозиторію на CentOS Mirror Network. CentOS

придатний для використання на базованих на Window платформах, однак його зазвичай використовують як серверну операційну систему для веб-хостингу. Багато великих хостингових компаній використовують CentOS разом із cPanel Control Panel для забезпечення стабільної роботи для своїх веб-додатків [1, 3].

Постановка завдання

Розглянути переваги та недоліки сучасних засобів реалізації середовищ діагностики КФС. Програмно реалізовано модулі, які в короткі терміни дають змогу діагностувати проблему в комунікаційній мережі та усунути її без участі людини. Розробити чіткий механізм виявлення відхилень у роботі комунікаційної мережі та сповіщення про них системного адміністратора.

Основні результати досліджень

Складові та принципи побудови спеціалізованого середовища кіберфізичних систем

Спеціалізоване середовище діагностики та налагодження вузлів комунікаційної мережі кіберфізичних систем має вигляд системи, здебільшого орієнтованої на мережевого адміністратора, оскільки з цього середовища і здійснюватимуться основні маніпуляції з комп'ютерною мережею. Програмне забезпечення розроблене переважно для збирання інформації, віддаленого моніторингу, налаштування вузлів зв'язку в комунікаційній мережі, щоб покращити якість зв'язку і запобігти нестабільній роботі обладнання, яке відповідає за надання послуг. Кіберфізична система розміщується на комунікаційній мережі інтернет-провайдера (рис. 2), що обґрунтовує доцільність цього проекту, оскільки без якісного доступу до мережі Internet сьогодні важко уявити сучасне життя, без надійного з'єднання не могли б працювати банки, супермаркети та інші установи.

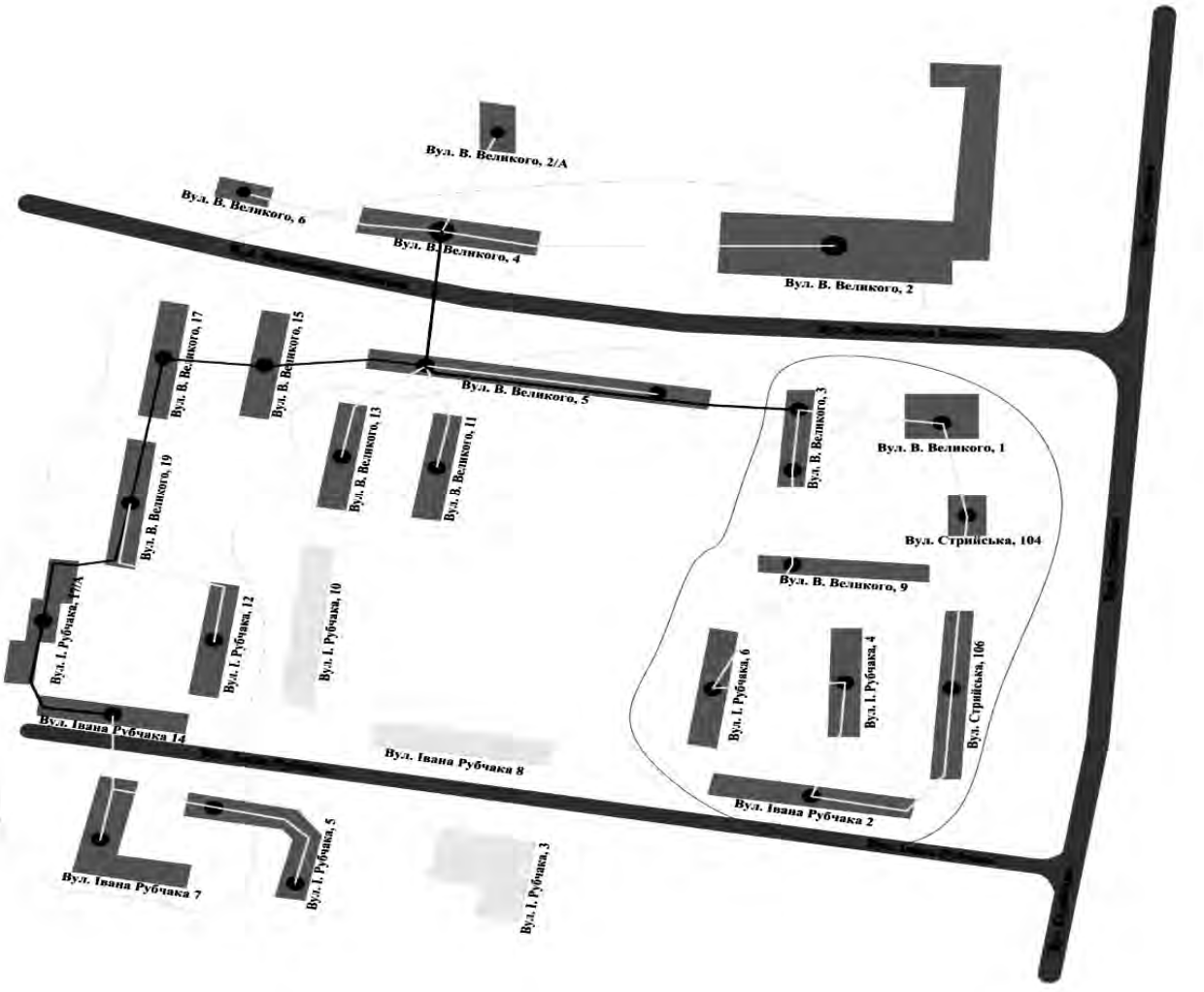

Рис. 2. Модернізація комунікаційної мережі інтернет-провайдера

Ця кіберфізична мережа розгортається в місті Львів та охоплює 32 житлові будинки, котрі виступають вузлами комунікаційної мережі. Будинки розташовані на вулицях Володимира Великого, Стрийській та Івана Рубчака (рис. 2). У комунікаційній мережі задіяні різні технології проектування комп'ютерних мереж, також застосовуються поєднання різних топології мереж, що дає змогу використовувати це середовище практично для будь-яких комп'ютерних мереж.

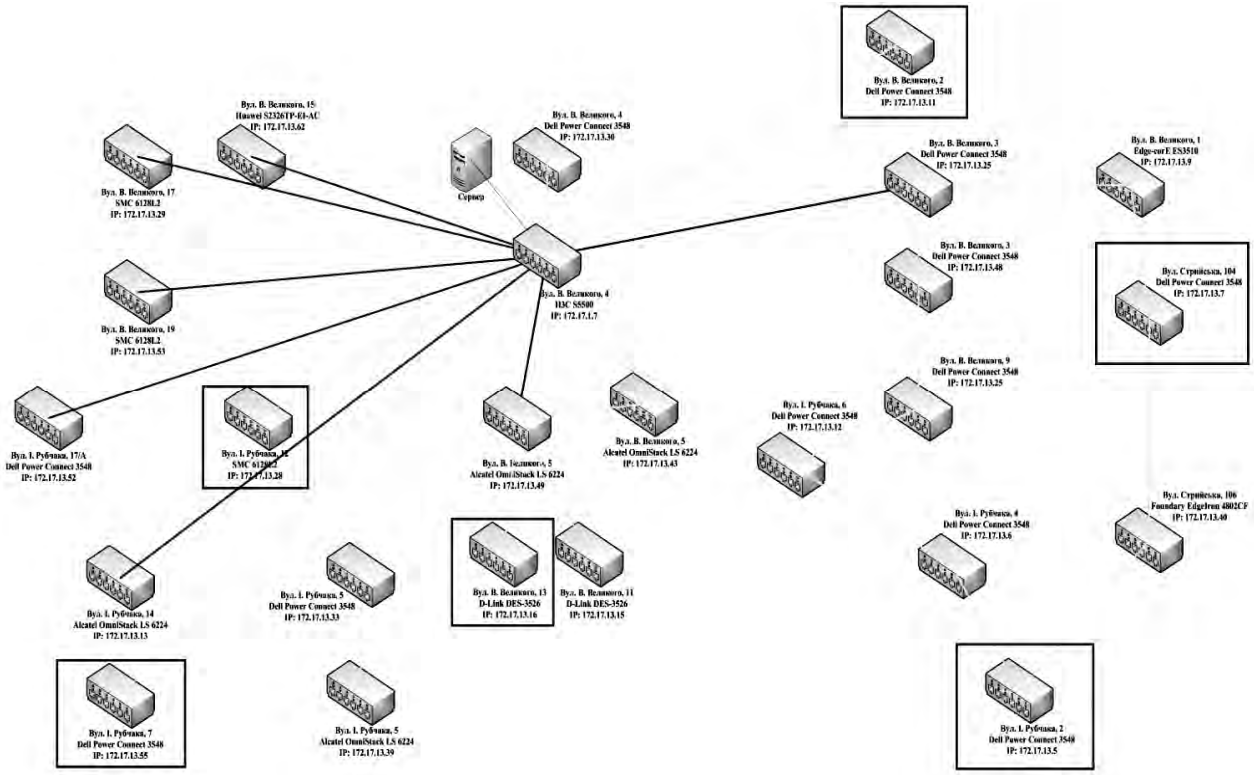

Рис. 3. Вузли комунікаційної мережі

Для того, щоб перевірити надійність використання протоколу SNMP як збирача інформації з вузлів комунікаційної мережі, а саме мережевих комутаторів, котрі там розташовані, порівняємо результати команди ping. У першому випадку результати будуть надані спеціалізованим середовищем кіберфізичних систем, в іншому випадку мережа самостійно відправлятиме команду ping на вузли керування. Отримані результати зможемо порівняти (рис. 3).

Структура середовища діагностики КФС

Спеціалізоване середовище діагностики вузлів корпоративної комп'ютерної мережі КФС працює за алгоритмом, наведеним у вступі. Перед розгорненням спеціалізованої системи моніторингу комунікаційної мережі потрібно підготувати середовище для цього. Середовищем КФС буде операційна система сім'ї Linux, а саме CentOS 7. Цю операційну систему вибрано через високу надійність, швидкодію та всі необхідні функції, які дають змогу реалізувати поставлене завдання. Операційна система CentOS – вільнодоступний дистрибутив Linux на основі комерційного дистрибутиву Red Hat Enterprise Linux компанії Red Hat.

Red Hat Enterprise Linux складений переважно з вільного та відкритого програмного забезпечення, але наявний у доступній для вживання двійковій формі (наприклад, на CD або DVD дисках) лише для передплатних користувачів. Як і вимагається, Red Hat випускає усі вихідні тексти своїх продуктів під GNU General Public License та іншими вільними ліцензіями. Розробники CentOS використовують цей вихідний код для створення кінцевого продукту, який є дуже подібним до Red Hat Enterprise Linux і вільним для завантаження та використання, однак без відповідної технічної підтримки з боку компанії Red Hat.

У результаті інсталяції операційної системи маємо у своєму розпорядженні апаратну консоль, де можна переглянути версію операційної системи за допомогою команди “cat /etc/*release”. Після інсталяції операційнії системи налаштовуємо мережеві інтерфейси на отримання IP-адреси від DHCP сервера та виконуємо команду # yumupdate для оновлення репозиторіїв та оновлення інстальованого програмного забезпечення до останніх версій.

Перед початком розгорнення системи моніторингу встановлюємо усі необхідні компоненти, які будуть потрібні для встановлення Nagios. Насамперед інсталюємо Apache сервер для того, щоб надалі керувати Nagios через веб-інтерфейс. Робимо це за допомогою команди:

yuminstallhttpd

Запускаємо веб-сервер та дозволяємо його автозавантаження під час старту операційної системи:

systemctl start httpd

systemctl enable httpd

Створюємо тестову сторінку для перевірки роботи веб-сервера та доступаємось до неї через браузер з іншого комп’ютера в тій самій підмережі. Після цього було інстальовано php-модуль для Apache, щоб мати обробник php-сценаріїв системи моніторингу, які використовуватимуться у керуванні системою через веб-сервер.

yum -y install php php-mbstring php-pear

systemctl restart httpd

Веб-сервер готовий до роботи з Nagios. Тепер потрібно підготувати базу даних, куди середовище діагностики записуватиме всі свої зміни про стан вузлів корпоративної мережі КФС. Для цього скористаємось MySQL. Оскільки CentOS 7 використовує в репозиторіях дистрибутив бази даних від MySQL, як MariaDB, то нам для того щоб інсталювати MySQL в чистому вигляді, потрібно додати його в репозиторії. Зробимо це так:

rpm–Uvhhttp://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm

Тепер, коли MySQL є в репозиторіях, інсталюємо його, використовуючи такі команди:

yum -y install mysql-community-server

MySQL інсталювався зі всіма компонентами, які потрібні для його роботи. Тепер аналогічно із веб-сервером запускаємо сервер баз даних та дозволяємо його запуск під час старту операційної системи. Маючи сконфігурований MySQL сервер, потрібно створити пароль для входу під користувачем root (unix системний акаунт зі всіма правами, як SYSTEM у WINDOWS). Для цього запускаємо ініціалізацію сервера баз даних:

/usr/bin/mysql_secure_installation

Також, для зручності, відключаємо розширені права доступу (ACL) та фаєрвол, щоб мати змогу надалі без проблем конфігурувати сервер. Це нам дає змогу безперешкодно здійснювати зміни на сервері, й у разі виникнення проблем з конфігурацією одразу вимикати з можливих причин фаєрвол та розширені права доступу операційної системи CentOS. Тепер, маючи все для розгортання спеціалізованої системи моніторингу, можна переходити до цього. Передусім потрібно створити користувача для спеціалізованого середовища діагностики, від імені якого середовище працюватиме. Систему моніторингу Nagios розгорнуто як основу для подальшої конфігурації та розроблення додаткових модулів.

Опис методів інтеграції спеціалізованих рішень у вибране середовище діагностики КФС

Конфігураційні файли вузлів комп’ютерної мережі зберігаються у директорії /usr/local/nagios/etc/servers. Саме сюди додаємо конфігураційні файли для кожного вузла корпоративної мережі КФС та його параметри. Головний файл конфігурації самої системи міститься у /usr/local/nagios/etc/nagios.conf. Тут конфігуруємо параметри самої системи, такі як шлях до логів системи, шаблонів конфігурацій вузлів корпоративної мережі, файлів кешування даних, загальних сервісів, користувача системи та групи, від якої запускається середовище тощо. Користувачі для доступу до сайта прописуються в файлі /usr/local/nagios/etc/htpasswd.users, який створено у попередньому розділі. Паролі в файлі зберігаються у захешованому вигляді, наприклад:

nagiosadmin:{SHA} Gd0+f++5bX5eiNS+YZjCWwCwx20=. Це дає змогу забезпечити безпеку, коли до сервера мають доступ багато людей так, що вони не зможуть побачити явно пароль для кожного користувача та залогуватись на сайт під ним.

Директорія /usr/local/nagios/etc/objects/ містить шаблони конфігураційних файлів. Файл /usr/local/nagios/etc/cgi.cfg містить інформацію про місцезнаходження головного конфігураційного файла, про директорію веб-сервісу та його веб-адресу, розмежування прав доступу до сайта та про інші налаштування, пов'язані з веб-сервісом. У файл /usr/local/nagios/etc/default.cfg збираємо усі стандартні для всіх серверів налаштування, такі як оголошення періодів часу сповіщень чи нагляду за вузлами мережі, групи вузлів корпоративної мережі, контакти осіб, яким надходитиме сповіщення і шаблони конфігураційних файлів. У файл /usr/local/nagios/etc/resource.cfg вносяться системні змінні, такі як ім'я користувача, шлях до певної директорії чи конфігураційного файла. Цей інструмент виявиться дуже корисним у випадку, коли ми перенесемо директорію з конфігураціями в інше місце чи захочемо перейменувати користувача. Для цього нам потрібно буде пізніше внести зміни тільки в цей файл і решта залежностей збережуться коректними. Інакше кажучи, щоб не шукати, де прописані всі назви об'єктів, передаємо їх через змінні і змінюємо тільки в одному місці. Також у цей файл записується шлях до компонентів системи [5, 8]. Файл /usr/local/nagios/etc/commands.cfg створено для опису команди виклику модулів системи і сповіщень. Розробляючи кожен модуль, додаємо його саме сюди.

Засоби перевірки надійності вузлів комунікаційної мережі в спеціалізованому середовищі діагностики КФС

Середовище діагностики Nagios є модульним, а тому всі розроблені модулі працюють незалежно один від одного, що збільшує гнучкість системи та безперебійність її роботи загалом. У кожного модуля своя роль, будучи незалежним, він відповідає за окремі функції, що зображено на рис. 4.

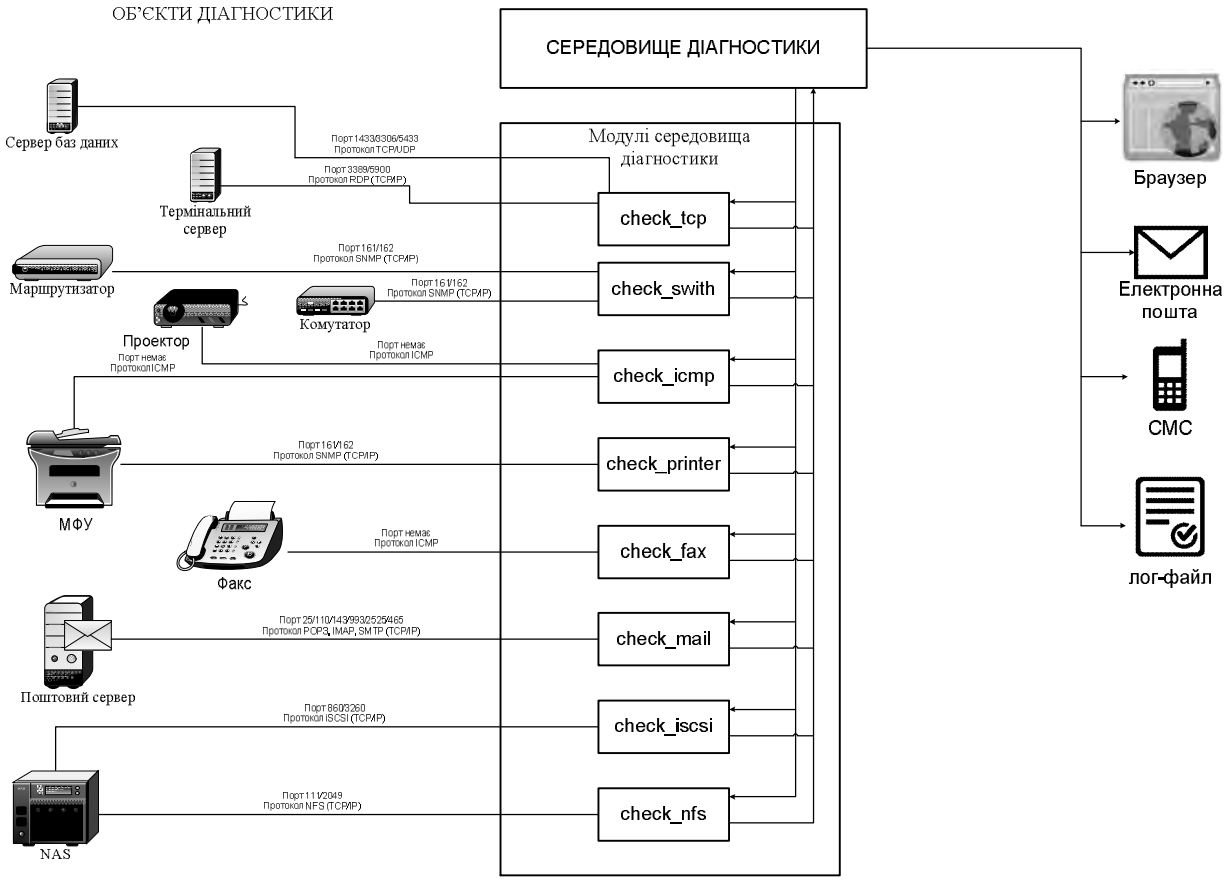

Рис. 4. Взаємодія модулів середовища діагностування КФС з мережею

Усі розроблені модулі поміщено за директорією /usr/local/nagios/libexec. Модулі викликаємо, перед тим описавши їх в конфігураційному модулі commands.cfg, де описано спосіб виклику модуля та передавання йому необхідних параметрів. За допомогою опису модуля маємо змогу передати йому такі параметри, як ім'я сервера, статус, адреса, дата та час, контакти, тип нотифікацій, аліас сервера, ім'я сервісу тощо.

Додавання модуля і його опис виглядають так:

```
define command{
    command_name    check_service
    command_line    check_service $ARG1$ $HOSTADDRESS$
    }
```

Розглянемо детальніше структуру описування модуля:

- define – оголошуємо модуль;
- command – вказуємо, що тип оголошення саме модуль;
- command_name – ім'я модуля (для зручності вказуємо в назві модуля функцію, за яку він відповідає);
- command_line – власне описано спосіб виклику модуля;
- check_service $ARG1$ $HOSTADDRESS$ – рядок виклику модуля з один аргументом та адресою сервера.

Маючи уявлення про самі модулі середовища діагностики корпоративної комп'ютерної мережі КФС, можна переходити до їх написання під різні сервіси. Деякі модулі вже існують як готове рішення у середовищі діагностики мережі Nagios, а деякі нам довелось дописувати під певні сервіси, за якими потрібно спостерігати. Усі модулі середовища розроблено за допомогою інтерпретатора bash та perl в операційній системі сім'ї Linux. Ми розробили та перепрограмували під наші потреби такі модулі, як: check_ntp_peer – перевірка часового ntp сервера; check_swap – перевірка файла підкачування ОС; check_arping.sh – пінгування сервера за MAC-адресою; check_ifstatus – перевірка статусу мережевого інтерфейсу; check_tcp – перевірка портів за tcp протоколом; check_imap – перевірка поштового протоколу imap; check_time – перевірка часу за сервером; check_udp – перевірка доступності портів за udp протоколом; check_ups – перевірка безперебійних блоків живлення; check_load – перевірка завантаженості ресурсів сервера; check_ping – перевірка доступності сервера за ICMP протоколом; check_uptime – перевірка часу від моменту увімкнення; check_dhcp – перевірка DHCP-сервера; check_pop – перевірка поштового протоколу POP; check_users – перевірка кількості користувачів, залогованих у системі; check_disk – перевірка об'єму диска та його наповненості; check_procs – перевірка запущених процесів; check_disk_smb – перевірка доступності спільних ресурсів за протоколом samba; check_website – перевірка доступності веб-сайта за доменом імені або конкретним URL; check_ftp – перевірка роботи FTP-сервера; check_smtp – перевірка роботи поштового сервера за протоколом SMTP; check_ssh – перевірка доступності 21 порту.

Усі модулі середовища діагностики корпоративної комп'ютерної мережі знаходять у директорії /usr/local/Nagios/libexec/, яка показана на рис. 5.

Кожен з модулів має права на виконання для користувача, від якого запущене середовище діагностування, а тому проблем із запуском системи та перевіркою вузлів всіма модулями не повинно виникати. Схему алгоритму роботи модулів спеціалізованого середовища діагностики корпоративної мережі показано на рис. 6.

Перевірка вузла починається із запуску модуля середовища діагностики з вказівкою про місцезнаходження модуля, який необхідно перевіряти. Після цього модуль перевіряє вузли і передає інформацію про його стан у середовище діагностики. Якщо немає інших сервісів для перевірки, то модуль завершує свою роботу.

```
[root@nagios libexec]# ls
check_apt         check_ifoperstatus   check_ntp_peer  check_swap
check_arping.sh   check_ifstatus       check_ntp_time  check_tcp
check_breeze      check_imap           check_nwstat    check_time
check_by_ssh      check_ircd           check_oracle    check_udp
check_clamd       check_jabber         check_overcr    check_ups
check_cluster     check_load           check_ping      check_uptime
check_dhcp        check_log            check_pop       check_users
check_disk        check_mailq          check_procs     check_wave
check_disk_smb    check_mrtg           check_real      check_website
check_dummy       check_mrtgtraf       check_rpc       negate
check_file_age    check_nagios         check_sensors   urlize
check_flexlm      check_nfs_export.pl  check_simap     utils.pm
check_ftp         check_nntp           check_smtp      utils.sh
check_http        check_nntps          check_spop
check_icmp        check_nt             check_ssh
check_ide_smart   check_ntp            check_ssmtp
[root@nagios libexec]#
```

Рис. 5. Модулі середовища діагностування комунікаційної мережі КФС

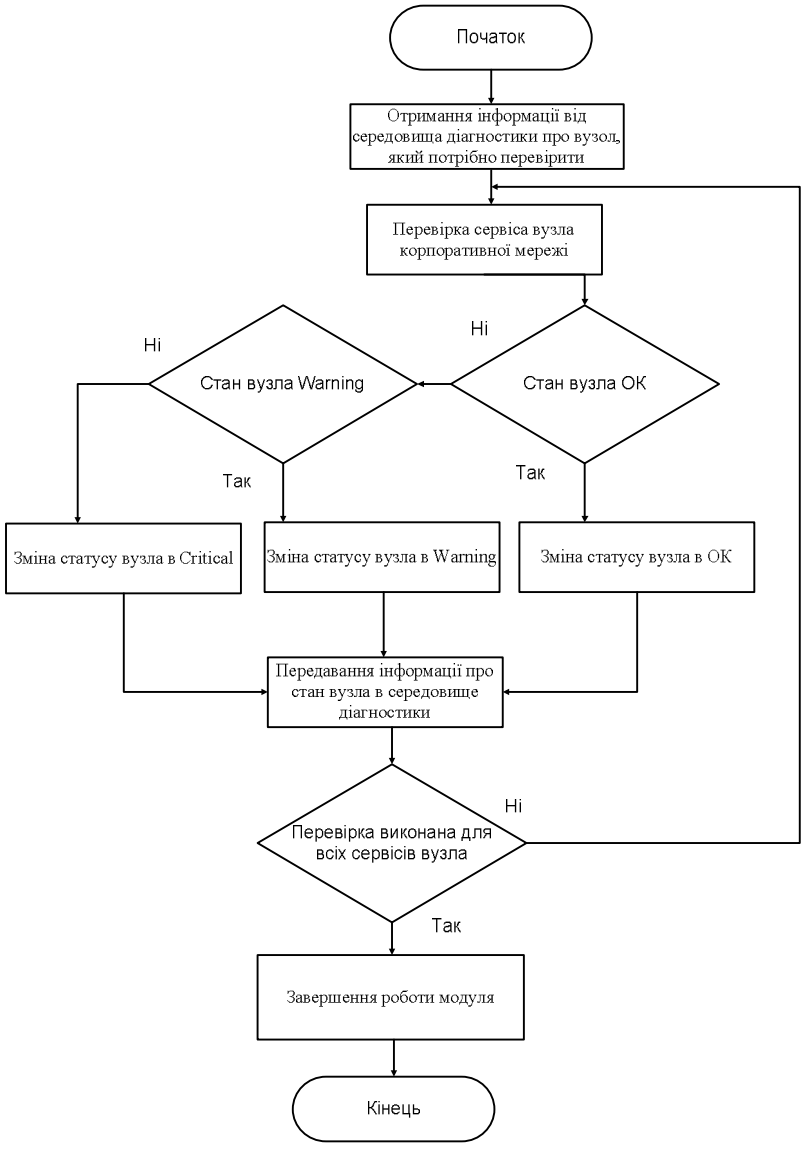

Рис. 6. Схема алгоритму роботи модулів середовища діагностики

Реалізація функціональної частини спеціалізованого середовища діагностики КФС

Після написання модулів системи конфігуруємо кожен вузол корпоративної мережі так, щоб його сервіси перевірялись нашими модулями і сигналізували про події чи відхилення в роботі, параметри яких не відповідають тим, що описані у модулі чи вказані при виклику одного з модулів [6, 9].

На кожен вузол корпоративної мережі КФС було створено окремий конфігураційний файл, який містить інформацію про ім'я вузла корпоративної мережі, його IP-адресу, сервіси, які відстежують на цьому вузлі та параметри, яким він повинен відповідати у разі нормальної роботи.

У конфігураційному файлі насамперед оголошується ім'я сервера та його адреса, після цього перевіряють декілька сервісів. Сервіси перевіряють модулями check_nt з аргументом UPTIME для перевірки часу, протягом якого піднятий сервер, аргументом CPULOAD для перевірки завантаженості сервера, MEMUSE для перевірки зайнятості оперативної пам'яті, USEDDISKSPACE для відстеження заповненості жорсткого диска, а також модуль check_tcp з параметром 49453, який виступає портом, на який слухають MicrosoftSQL-server на цьому вузлі[10].

Наступні вузли для спостереження конфігуруються так само, тільки для кожного з них вказують різні модулі, розроблені та розміщені в директорію /usr/local/nagios/libexec. Усі конфігураційні файли поміщені у директорії /usr/local/Nagios/etc/servers та мають розширення .cfg. Після додавання нового конфігураційного файла в нашу систему моніторингу корпоративної мережі чи редагування конфігураційного файла систему потрібно перезавантажувати, щоб вона перечитала ці зміни. Перезавантажуємо систему командою: # systemctlrestartNagios. Після цього зміни здійснюються.

Перевірка працездатності спеціалізованого середовища діагностики КФС

Після розгортання середовища діагностики КФС, створення конфігураційних файлів для всіх вузлів корпоративної комп'ютерної мережі та підключення до них розроблених модулів, які було розроблено, це все ми бачимо у веб-інтерфейсі нашої системи, показаному на рис. 7.

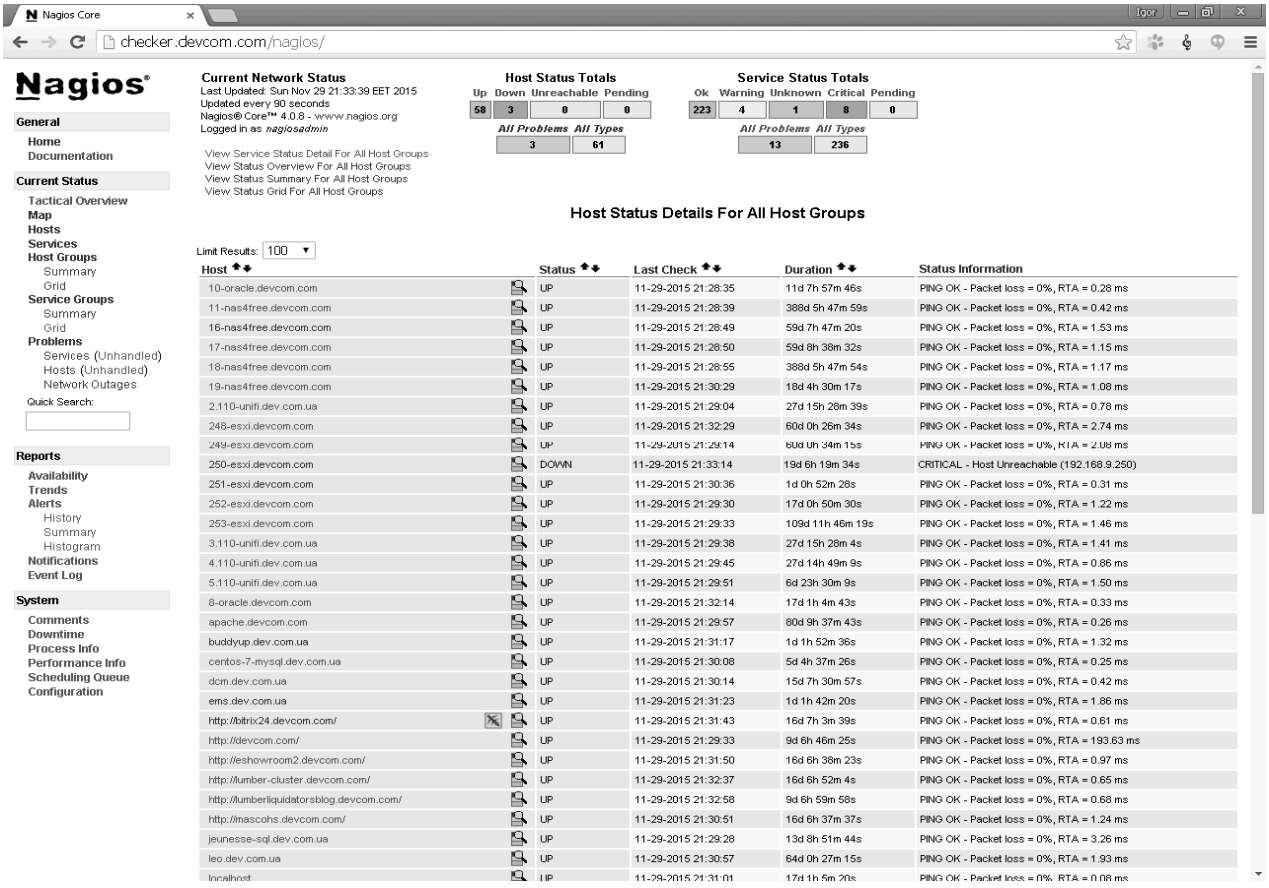

Рис. 7. Веб-інтерфейс середовища Nagios зі створеними вузлами корпоративної комп'ютерної мережі КФС

Також на скріншоті бачимо, що один вузол корпоративної мережі, а саме 250-esxi.devcom.com, вимкнений більш ніж 19 днів (колонка Duration) та на іншому вузлі комп'ютерної мережі відключені нотифікації про статус вузла, про що нам говорить відповідне позначення [7, 10].

Якщо нас цікавить детальніша інформація про вузол, ми можемо її отримати, клацнувши по відповідному сервері чи вузлі корпоративної мережі, який нас цікавить. Після цього у нас відкриється інше вікно з детальнішою інформацією про вузол, що ми і бачимо на рис. 8.

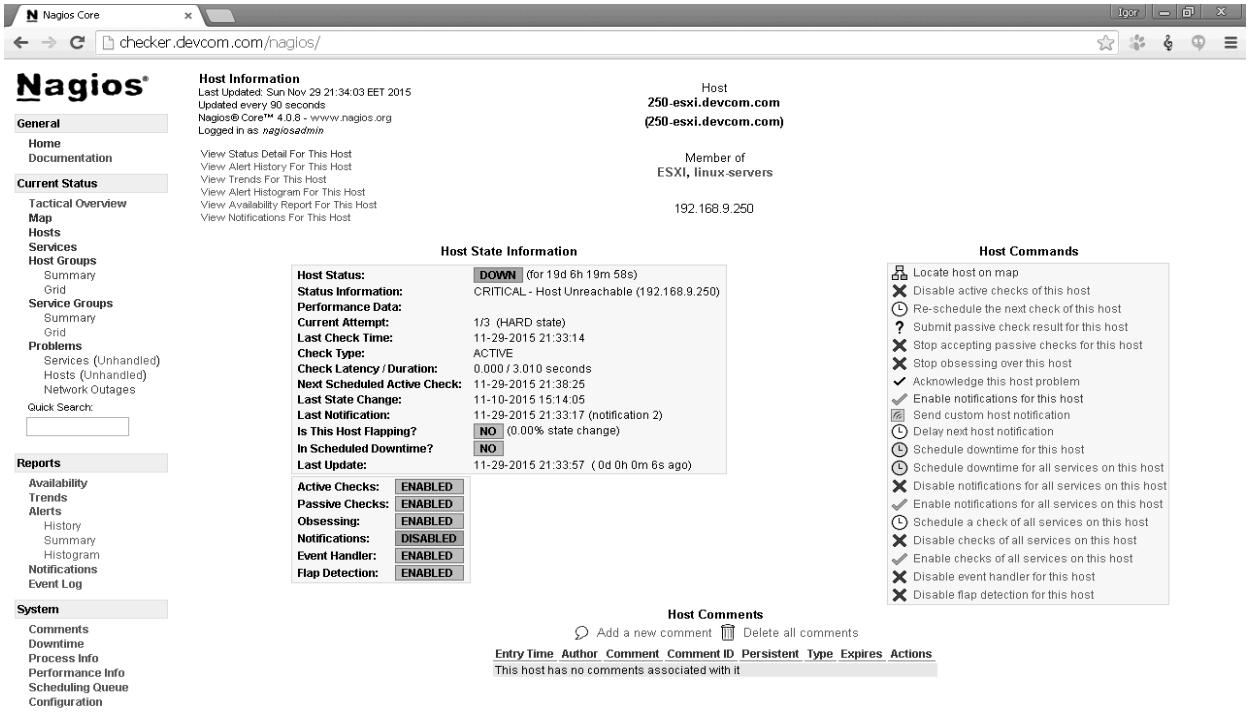

Рис. 8. Вікно з детальнішою інформацію про вузол корпоративної комп'ютерної мережі КФС

У цьому вікні, як бачимо, можна виконувати різні дії з налаштуваннями діагностики цього вузла, такі як заборонити діагностувати вузол протягом певного часу, зовсім заборонити діагностику як вузла, так і всіх його сервісів, тимчасово заборонити сповіщення про вихід з ладу вузла чи абсолюто вимкнути усі сповіщення, які стосуються цього вузла. Також ми бачимо, що можна перейти на сторінку, де описано всі сервіси цього вузла, які на цей момент діагностуються. Це показано на рис. 9.

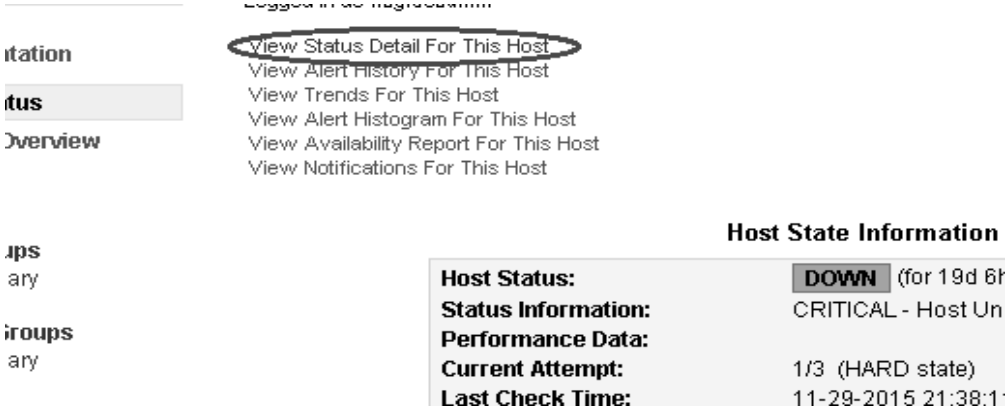

Рис. 9. Метод переходу до перегляду всіх сервісів поточного вузла мережі

Після переходу на нову сторінку можна виконувати ті самі налаштування до кожного окремого сервісу, натиснувши на ньому, як і у випадку з конкретним вузлом корпоративної комп'ютерної мережі (рис. 10).

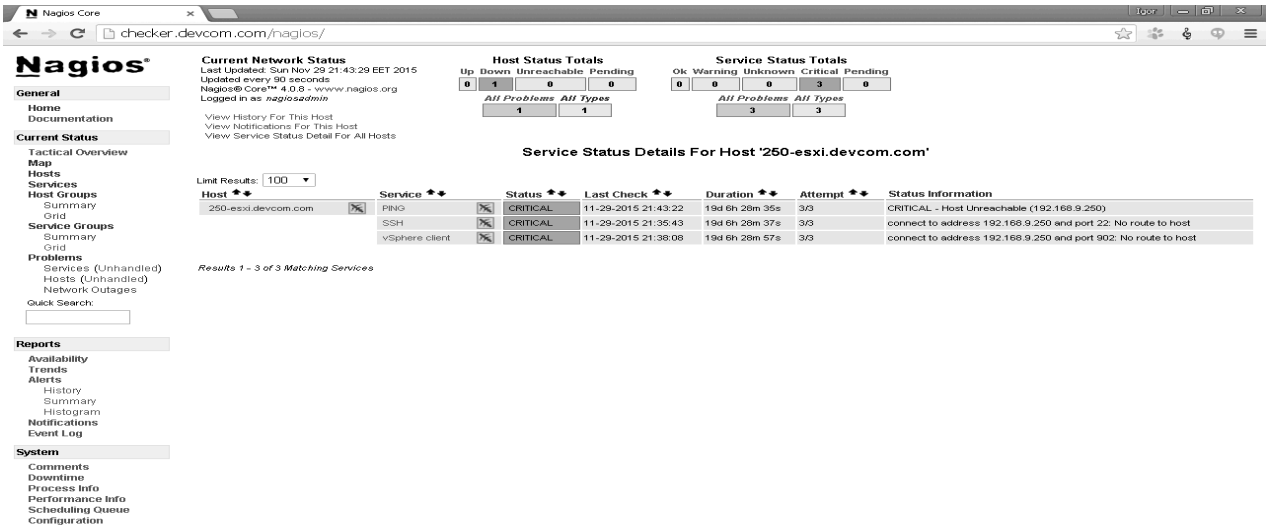

Рис. 10. Веб-сторінка усіх сервісів вузла корпоративної комп'ютерної мережі 250-esxi.devcom.com

Бачимо, що нотифікації на всіх сервісах і сервері вимкнені, про що сигналізують іконки біля них. Після виходу з ладу вузла корпоративної комп'ютерної мережі отримуємо про це повідомлення на електронну пошту, яка під час конфігурації самого середовища була сконфігурована. Тут ми можемо побачити повідомлення про час виходу з ладу вузла, його назву, статус на зараз (OK, down чи warning) та його IP-адресу (рис. 11).

Рис. 11. Електронний лист з повідомленням про вихід з ладу одного з вузлів корпоративної мережі КФС

Налаштувавши вузли корпоративної комп'ютерної мережі КФС на діагностику, ми перевірили роботу середовища діагностування і побачили, що всі сервіси функціонують і готові для подальшого тестування. Усі конфігурації з увімкненнями нотифікацій про статус хоста та увімкнення або вимкнення його діагностування можемо здійснювати також з консолі, не використовуючи веб-інтерфейс. Веб-інтерфейс було розгорнуто для зручності та наочності роботи всієї мережі, а також для пришвидшення конфігурації середовища діагностування.

Дослідження навантаження на вузли комунікаційної мережі КФС

Припустимо, що існує потреба перевіряти, наскільки навантажений той чи інший вузол корпоративної комп'ютерної мережі КФС чи час доступу до вузла. На цей випадок також маємо такий показник доступності вузла, як затримка доступу до вузла. Це час, через який було отримано

відповідь від сервера чи будь-якого мережевого обладнання у нашій мережі. У цій статті сконфігуровано багато сайтів для діагностики. Усі сервіси підняті на Apache сервері з певною IP-адресою. Для того, щоб відтворити завантаженість сервера та побачити, як система реагує на такі випадки, можна навантажити сервера процесом циклу. Створюємо скрипт, під час запуску якого наш сервер буде завантажуватись на 100 %. Для цього достатньо завантажити одне ядро серверного процесора будь-яким процесом. Щоб це зробити, запускаємо таку команду: # cat /dev/zero > /dev/null. Тобто пишемо нулі в нікуди і так до нескінченності. У результаті отримуємо завантаження процесора на 100 %. Після навантаження веб-сервера спостерігаємо за попередженнями, які відправляє на електронну адресу середовище діагностики корпоративної комп'ютерної мережі КФС (рис. 12).

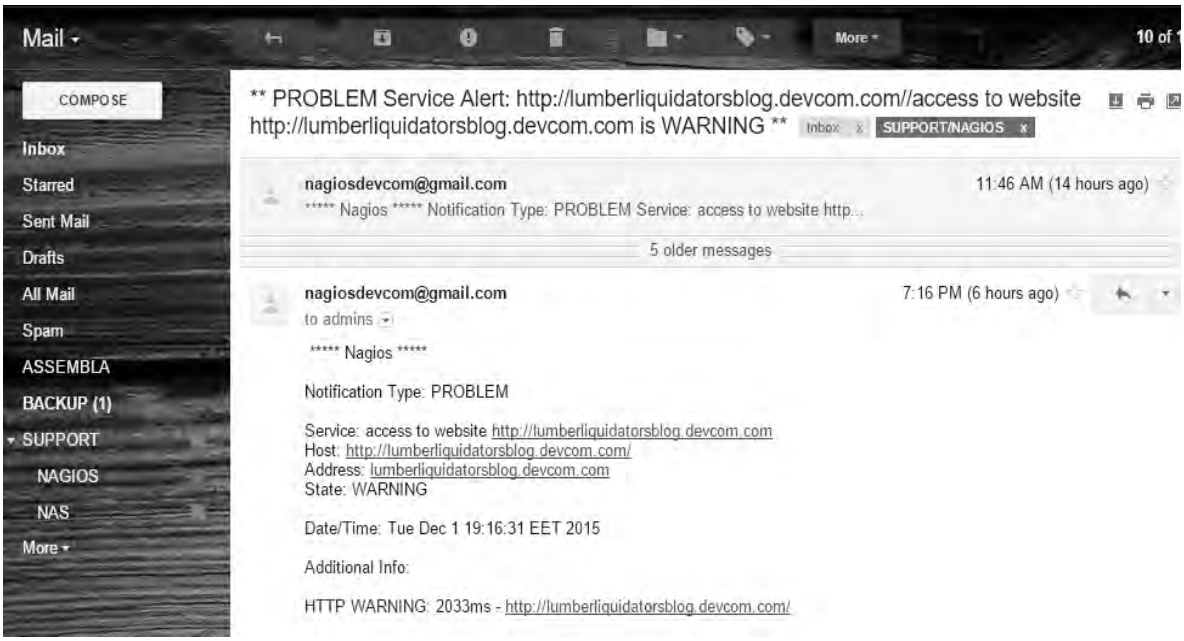

Рис. 12. Попередження про труднощі доступу до веб-сервісу

На цьому скріншоті бачимо, що веб-сервіс почав відгукуватися на запити впродовж більше ніж 2 с і на цей випадок отримуємо повідомлення у вигляді попередження про це. Так можна не тільки перевіряти доступність всіх серверів, але й наскільки ця доступність відповідає загальним вимогам до поставленого завдання.

Висновки

У статті описано принципи побудови комунікаційних мереж КФС та їхню розподіленість. Досліджено роботу спеціалізованого середовища діагностики КФС та змодельовано комунікаційну комп'ютерну мережу. Одночасно описано процес діагностики вузлів комунікаційної мережі та варіанти їх відхилень під час роботи. Здійснено діагностику мережі – об'єкта дослідження та змодельовано роботу КФС у реальних умовах. Проведено первинну перевірку функціонування КФС загалом. Розроблено модулі для перевірки багатьох сервісів вузлів комунікаційної комп'ютерної мережі КФС, які повідомляють про роботу мережі загалом та використовуються для діагностики майбутніх неполадок у комунікаційній мережі КФС. Налаштовано функцію сповіщення про вихід з ладу одного чи певної кількості вузлів комунікаційної комп'ютерної мережі, системного адміністратора за допомогою поштової. Наведено алгоритм тестування комунікаційної мережі в спеціалізованому середовищі діагностики КФС. Змодельовано роботу спеціалізованої системи діагностики КФС та налагодження комунікаційної комп'ютерної мережі. Досягнуто потрібного рівня стабільності роботи середовища за кількості вузлів до 500 одиниць комунікаційної комп'ютерної мережі КФС. Розроблений алгоритм діагностування комунікаційної

комп'ютерної мережі КФС з використанням модулів перевірки сервісів можна використати як основу для складніших багаторівневих комунікаційних комп'ютерних мереж КФС з великою кількістю вузлів.

1. Chris Giametta "Pro Flex on Spring", 2009. – P.445. 2. Robert Dzh. Oberg "Tehnologija COM + Osnovy i programirovanie = Understanding and Programming COM+: A Practical Guide to Windows 2000 First Edition". – M.: Viljams, 2000. – P. 480. 3. Lipaev V. V. Obespechenie kachestva programnyh sredstv. Metody i standarty. – M. : Sinteg, 2001. – P. 246. 4. Makgregor Dzh., Sajks D. Testirovanie obektno-orientirovannogo programnogo obespechenija. – K: Diasoft, 2002. – P. 432. 5. Tamre L. Vvedenie v testirovanie programnogoobespechenija. – M.: Viljams, 2003. – P. 368. 6. Tatarchuk M. I. Korporatyvni informatsijni systemy: Navch. posibnyk, 2005. – P. 245. 7. Muhamedzjanov N. Java. Server applications" – Izdatelstvo: SOLON – R, 2003. – P. 267. 8. Orfali Robert, Den Harki. JAVA and CORBA in client server applications. 9. Duglas Kamer, Devid L. Stivens Seti TCP/IP, tom3. Razrabotka prilozhenij tipa klient/server, Viljams, 2002. – P. 592. 10. Flenov M. E. Web-server glazami hakera: Problemy bezopasnosti Web-serverov; Oshybki v stsenarijah na PHP, Perl, ASP; SQL-inektsii, 2005. – P. 365. 11. Melnyk A. O. Kiberfizychni systemy: problemy stvorennja ta naprjamy rozvytku. // Visnyk Natsionalnogo Universytetu "Lvivska politehnika" "Kompjuterni systemy ta merezhi". – 2015. – No. 692. – P. 100–107. 12. Mijushkovych Je. Ja., Grebenjak A. V., Garamud Ja. S. Telekomunikatsijni pidsystemy kiberfizychnyh system // Visnyk Natsionalnogo Universytetu "Lvivska politehnika" "Kompjuterni systemy ta merezhi", 2016, No. 857. – P. 65–74.