

Детектор Плагіату v. 2945 - Звіт оригінальності: 14.06.2026 19:48:49

Перевірений документ: 4_ПЗ_Семесенко Ігор.docx Ліцензія: ВОЛОДИМИР МАТІЄВСЬКИЙ

? Тип пошуку: Пошук Перепису ? Знайдено мову: Uk

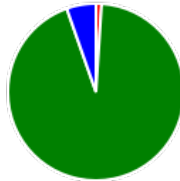
? Тип перевірки: Інтернет-перевірка

TEE і кодування: DocX n/a

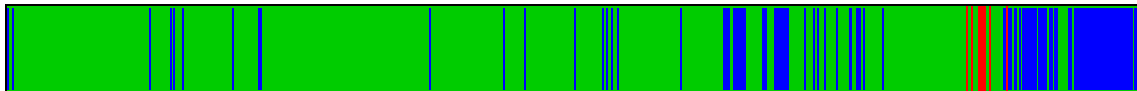
Детальний аналіз документа документа:

? Співвідношення:

Плагіат 1.02% Оригінал 93.61%
Цитати 5.38% ШІ 0%



? Графік розподілу:



? Джерела плагіату: 10

7% 840 1. https://web-crawler.plagiarism-detector.com/get-doc-pt?did=9kilP4y9-jpDIg
7% 766 2. https://web-crawler.plagiarism-detector.com/get-doc-pt?did=_0iiP4y7-zZBkQ
2% 168 3. https://ela.kpi.ua/bitstream/123456789/25275/1/TULR_navch_posibn.pdf

? Дані оброблених ресурсів: 167 - ОК / 15 - Помилок

? Важливі примітки:

Wikipedia:	Google Books:	Сервіси платних робіт:	Античіт:
[не знайдено]	[не знайдено]	[не знайдено]	Знайдена протидія!

? Звіт проти обману UACE:

- Статус: Аналізатор **Увімкнений** Нормалізатор **Увімкнений** схожість символів встановлено **100%**
- Виявлений відсоток забруднення UniCode: **14,1%** з обмеженням: 4%
- Відсоток невизнаних символів після нормалізації: **8,4%**
- Усі підозрілі символи будуть позначені фіолетовим кольором: **Abcd...**
- Знайдено невидимі символи: 0

Рекомендація з оцінки:

Аналізу цього звіту слід приділити особливу увагу! Підозрюється, що цей документ містить значну кількість символів, чужих мові документа. Це є прямим вказівкою на те, що автор документа використав спеціальне програмне забезпечення онлайн - веб - сервіс, щоб ефективно заплутати текст, намагаючись уникнути потенційного виявлення плагіату. Наполегливо рекомендується ескалювати цю справу! У разі сумнівів зверніться до служби підтримки детектора плагіату!

Статистика алфавіту та аналіз символів:

? Активні посилання (URL-адреси, витягнуті з документа):

URL не знайдені

? Виключено:

URL не знайдені

? Включено:

URL не знайдені

0123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100

Детальний аналіз документу:

Міністерство освіти і науки України ДЗ

Знайдено посилань: 0,05%

id: 1

«луганський Національний університет імені тараса шевченка»

Факультет технологій та інформаційних систем (назва факультету, інституту)
Інформаційних технологій та систем (назва кафедри) Пояснювальна записка до
дипломного проекту (роботи) БАКАЛАВРА (освітньо-кваліфікаційний рівень) на тему:
ВИКОРИСТАННЯ [MIKROTIK](#) ДЛЯ СТАРТУ ТА ЗАВЕРШЕННЯ РОБОТИ СЕРВЕРУ [ESXI](#) Виконав:
студент 4 курсу, групи 4KI напрямку підготовки (спеціальності) 123(E7)

Знайдено посилань: 0,02%

id: 2

«Комп'ютерна інженерія»

(шифр і назва напрямку підготовки, спеціальності) Семесенко І.М_ (прізвище та ініціали)
Керівник Могильний Г.А_ (прізвище та ініціали) Рецензент Козуб Ю.Г._ (прізвище та ініціали) Лубни – 2026 ЗМІСТ ВСТУПЗ Розділ 1 Сучасні системи підтримки живлення7 1.1
Огляд сучасних систем підтримки живлення7 1.2 [ESS](#) у системах зберігання енергії8 1.3
Промислові та модульні джерела безперебійного живлення (ДБЖ / [UPS](#)) для серверних приміщень13 1.4 Особливості джерел безперебійного живлення19 1.4.1. Аналіз базових топологій ДБЖ20 1.4.2. Аналіз технологій подвійного та Дельта-перетворення21 1.4.3. Системні характеристики сучасних Онлайн-ДБЖ23 1.4.4. Еволюція акумуляторних систем: [VRLA](#) проти [LiFePO4](#)24 1.4.5. Переваги впровадження Онлайн-ДБЖ у критичній інфраструктурі25 Висновки до розділу26 Розділ 2 Розробка системи керування електроживленням для серверів [Vmware ESX v7](#)29 2.1 Аналіз станів електроживлення та попередні налаштування29 2.1.1. Специфічні налаштування [BIOS/UEFI](#) для серверів29 2.1.2. Особливості процесу вимкнення ([Shutdown Sequence](#))33 2.1.3. Особливості процесу старту ([Boot Sequence](#))33 2.2 Передумови використання [Mikrotik](#)35 2.2.1.Процес виключення35 2.1.2.Процес включення36 2.3 Основні складові системи37 2.4 Загальна послідовність роботи скриптів роутеру [MikroTik](#)39 2.5 Особливості створення скриптів роутеру [MikroTik](#)41 Висновки до розділу49 Загальні висновки52 Список літератури55 Додатки58 Додаток А Текст головного скрипту для старту58 Додаток Б Текст головного скрипту для завершення60 Додаток В Текст першого допоміжного скрипту для старту62 Додаток Д Текст першого допоміжного скрипту для завершення63 Додаток Е Текст інших допоміжних скриптів для старту64 Додаток Є Текст інших допоміжних скриптів для завершення65 ВСТУП Змн. Арк. № докум. Підпис Дата Арк. 3 ІТС.4KI.0326.03-ПЗ Розроб. Семесенко І. Керівник Могильний Г.А. В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. Вступ Літ. Акрушів З ЛНУ Кафедра ІТС, Гр.4KI . Запровадження воєнного стану на території України зумовило масштабну трансформацію системи вищої освіти, змусивши більшість вітчизняних закладів вищої освіти (ЗВО) повністю або частково перейти на дистанційний та гібридний формати організації освітнього процесу. Систематичні оголошення сигналів повітряної тривоги та необхідність перебування учасників освітнього процесу в укриттях створюють безпрецедентні виклики для забезпечення безперервності навчання. За таких умов критично важливим завданням стає реорганізація функціонування навчальних комп'ютерних лабораторій та забезпечення віддаленого, стабільного і безпечного доступу здобувачів освіти до спеціалізованого апаратного та програмного забезпечення [1,2]. Особливої ваги ця проблема набуває у контексті підтримки працездатності серверних систем та високопродуктивних обчислювальних вузлів, які використовуються у навчальному процесі для моделювання, проектування та виконання лабораторних робіт. Регулярні відключення електроенергії, спричинені цілеспрямованим енергетичним терором та масованими атаками на об'єкти критичної інфраструктури України, призводять до аварійного вимкнення серверного обладнання. Для складних віртуалізованих середовищ, зокрема на базі платформи [VMware vSphere](#), раптове зникнення живлення є критичним чинником, що загрожує порушенням цілісності метаданих файлової системи [VMFS](#), руйнуванням віртуальних дисків та втратою результатів досліджень здобувачів освіти. Забезпечення віддаленого доступу до лабораторних потужностей вимагає розв'язання двох взаємопов'язаних задач: автоматизованого дистанційного ввімкнення (ініціалізації) обладнання за допомогою мережевих протоколів (наприклад, [Wake-on-LAN](#)) та організації процедури безпечного,

контрольованого вимкнення ([graceful shutdown](#)) серверних систем у разі тривалої відсутності електропостачання. Вирішень цих питань неможливо без засосування сучасних мережевих технологій [3-6]. Традиційні промислові рішення для моніторингу та керування електроживленням, що базуються на використанні сучасних систем підтримки живлення мають складну структуру [7] та включають багато різних рівней керування [8-10] та не застосовуються у невеликих підрозділах. З іншого боку, більш прості рішення, що використовують промислові джерела безперебійного живлення (ДБЖ) класу [Enterprise](#) із вбудованими мережевими картами керування ([NMC — Network Management Card](#)), також мають надзвичайно високу ринкову вартість та не застосовуються у навчальних закладах у період воєнного стану. Крім того, вони відрізняються низькою програмною гнучкістю та

обмеженими можливостями кастомізації під специфічні програмні-апаратні сценарії конкретного навчального закладу. Для сегменту малих та середніх серверних кластерів, які зазвичай функціонують на кафедрах ЗВО, закупівля таких систем є економічно недоцільною або неможливою в умовах обмеженого бюджетного фінансування. У зв'язку з цим виникає гостра потреба в розробці альтернативних, гнучких та бюджетних автоматизованих систем керування електроживленням. Перспективним напрямом є інтеграція доступної мережевої інфраструктури на базі активного обладнання [MikroTik](#) (із використанням вбудованої мови скриптів [RouterOS](#) та можливостей моніторингу мережі) з побутовими або напівпромисловими портативними зарядними станціями великої потужності, що виконують роль резервних джерел енергії. Такий підхід дозволяє створити адаптивну систему раннього попередження та автоматизованого завершення роботи віртуалізованої інфраструктури з мінімальними фінансовими витратами, що підтверджує актуальність, своєчасність та практичну значущість обраного напрямку дослідження. Об'єкт дослідження — процес автоматизованого керування станом електроживлення та функціонуванням серверної та мережевої інфраструктури навчальних комп'ютерних лабораторій в умовах нестабільного енергопостачання. Предмет дослідження — програмно-апаратні засоби інтеграції мережевого обладнання [MikroTik](#) та резервних джерел живлення для серверних середовищ [VMware vSphere](#). Мета дослідження — підвищення живучості та надійності функціонування навчальних комп'ютерних лабораторій шляхом проектування, розробки та впровадження гнучкої, економічно ефективної автоматизованої системи дистанційного керування живленням на базі інфраструктури [MikroTik](#) та портативних зарядних станцій. Для досягнення поставленої мети необхідно вирішити такі завдання: Проаналізувати існуючі рішення підтримки живлення та протоколи дистанційного керування електроживленням серверного обладнання та виявити їхні недоліки в умовах обмеженого бюджету ЗВО. Дослідити архітектуру та програмний інструментарій керування середовищем віртуалізації [VMware vSphere](#), зокрема механізми програмування для контрольованого завершення роботи віртуальних машин та хостів [ESXi](#). Розробити алгоритми взаємодії між мережевим обладнанням [MikroTik](#) (через скрипти [RouterOS](#)) та сервером [ESX](#) з віртуалізацією для моніторингу стану мережі та ініціалізації вимкнення систем. Спроекувати та реалізувати програмно-апаратну схему інтеграції портативної зарядної станції в загальну систему моніторингу живлення. Провести експериментальне тестування розробленого рішення в умовах навчальної лабораторії та оцінити його технічну ефективність. У першому розділі проведено детальний аналіз сучасних технологічних рішень у сфері забезпечення безперервного та якісного електропостачання критичної ІТ-інфраструктури, обчислювальних центрів та серверних кімнат. На основі аналізу міжнародних стандартів, зокрема [ANSI/TIA-942](#), розглянуто архітектуру, функціональні можливості та ключові компоненти промислових систем зберігання енергії ([ESS](#)) та сучасних модульних джерел безперебійного живлення (ДБЖ) онлайн-типу із подвійним перетворенням. У другому розділі було проведено опис процесу створення та розробка бюджетної, але високоефективної системи керування електроживленням серверів [ESXi](#) на базі програмованих можливостей сучасного мережевого обладнання [MikroTik](#), наведено основні попередні налаштування, процеси створення сертифікатів, алгоритм роботи скриптів. Розділ 1

Сучасні системи підтримки живлення Змн. Арк. № докум. Підпис Дата Арк. 3
ІТС.4КІ.0226.03-ПЗ Розроб. Семесенко І., Керівник Могильний Г.А. В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. Розділ 1 Літ. Акрушів З ЛНУ Кафедра ІТС, Гр.4КІ . 1.1 Огляд сучасних систем підтримки живлення Надійність функціонування сучасних інформаційно-обчислювальних систем, серверних кімнат та високопродуктивних обчислювальних кластерів ([HPC](#)) критично залежить від якості та безперервності електропостачання. Навіть короточасні протидія напруги тривалістю в кілька мілісекунд можуть призвести до збоїв у роботі оперативної пам'яті серверів, пошкодження файлових систем, втрати результатів складних обчислень та тривалого простою дорогого обладнання. Аналіз сучасних науково-технічних літературних джерел та провідних промислових стандартів (зокрема стандарту [ANSI/TIA-942](#) [11] для центрів обробки даних) дозволяє класифікувати сучасні системи підтримки електроживлення на кілька ключових технологічних рівнів та типів. Усі системи підтримки та резервування електроживлення підприємств та установ сьогодні доцільно розподілити на такі чотири взаємопов'язані класи: Системи накопичення енергії промислового масштабу ([Energy Storage Systems](#) — [ESS](#)): Використовуються на рівні підприємства або окремого потужного обчислювального центру для згладжування піків споживання ([Peak Shaving](#)), інтеграції з відновлюваними джерелами енергії та забезпечення довгострокового резерву. Вони базуються на хімічних акумуляторах високої ємності або фізичних накопичувачах. Резервні системи генерації енергії ([Backup Power Generation](#)):

Призначені для тривалого забезпечення електроенергією об'єкта під час аварій у зовнішній мережі, які перевищують час автономної роботи акумуляторних батарей. До цього класу належать дизель-генераторні установки (ДГУ),

газопоршневі установки (ГПУ) та перспективні водневі паливні елементи ([Fuel Cells](#)).

Промислові та модульні джерела безперебійного живлення (ДБЖ / [UPS](#)) для серверних приміщень: Виконують роль першої лінії захисту чутливого обладнання. Вони забезпечують миттєву фільтрацію перешкод, стабілізацію напруги та безперервне живлення на час, необхідний для запуску резервних генераторів або коректного завершення роботи операційних систем серверного кластера. Побутові та малі комерційні системи накопичення та безперебійного живлення: Застосовуються для локального захисту робочих станцій, мережевого обладнання провайдерів ([Edge Computing](#)) та невеликих офісних серверів. Вони мають меншу потужність і спрощену топологію. 1.2 [ESS](#) у системах зберігання енергії У своїй основі Система зберігання енергії ([ESS](#)) вловлює енергію, надійно зберігає її та вивільняє пізніше, коли це необхідно. Сприймайте це як удосконалений банк електричної енергії. Основоволожний принцип: Від'єднати час виробництва енергії від часу її споживання. Це дуже потужна можливість. Два ключові параметри визначають електричну [ESS](#): Енергетична ємність (кВт·год / МВт·год): Повна енергія, яку може вмістити система. Визначає, як довго, скільки енергії або скільки сонячної енергії вона накопичує. Номінальна потужність (кВт / МВт): Максимальна швидкість потоку енергії. Швидкість заряджання/розряджання

 Знайдено посилань: **0,02%**

id: 3

«прямо зараз».

Критично важлива для пікових навантажень або підтримки мережі. Розуміння співвідношення потужності та продуктивності є необхідним для правильної оцінки та визначення розмірів [ESS](#) для конкретного застосування. Перехід до чистої енергетики у багатьох країнах неможливий без сучасних накопичувачів енергії. Зростання [ESS](#) є життєво важливим у зв'язку зі змінними вимогами до мереж та характеристиками відновлюваних джерел енергії. Забезпечення надійної інтеграції відновлюваних джерел енергії: Сонячна енергія/вітер працюють із перебоями. [ESS](#) накопичує енергію за її надлишку і вивільняє її за низького виробітку, але високого попиту, роблячи відновлювані джерела енергії передбачуваними та диспетчеризованими для операторів енергосистем. Зміцнення стабільності та стійкості електромереж: [ESS](#) забезпечує практично миттєву підтримку частоти, напруги та швидкого резервного живлення, роблячи енергосистему більш стійкою до коливань та відключень. Забезпечення значної економії коштів: Заряджайте електроенергію, коли тарифи на неї низькі (поза піком, сонячна енергія), і скидайте під час дорогих піків (оптимізація [TOU](#) — часу використання). Для підприємств [Peak Shaving](#) (зрізання піків) скорочує високі платіжі за потужність (плату за попит). Забезпечення надійного резервного живлення: Безперешкодно відключається від мережі під час перебоїв у роботі ([Islanding](#) — ізоляція) і відразу ж живить критично важливі навантаження для будинків та підприємств, забезпечуючи необхідну стійкість, часто швидше, ніж генератори. Прискорення зусиль із декарбонізації: Забезпечує вищий рівень інтеграції відновлюваних джерел енергії та знижує залежність від

 Знайдено посилань: **0,01%**

id: 4

«пікових»

станцій на забруднюючому викопному паливі, скорочуючи викиди парникових газів. [ESS](#) — це складна, інтегрована система, що складається з ключових компонентів, які працюють разом розумно та безпечно: Накопичувач енергії: Резервуар енергії Елемент, що акумулює енергію (наприклад, літій-іонні акумулятори, проточні батареї). Вибір впливає на продуктивність та безпеку. Система перетворення енергії ([PCS](#)): Керування потоком енергії

 Знайдено посилань: **0,02%**

id: 5

«Електричний інтерфейс».

Керує потоком електроенергії в (заряджання) та з (розряджання) накопичувача (наприклад, двонаправлений інвертор). Перетворює постійний струм на змінний і навпаки. Ефективність та швидкість роботи мають вирішальне значення. Система керування батареєю ([BMS](#)): Забезпечення здоров'я та безпеки батареї КРИТИЧНО важлива для акумуляторних систем. Виступає в ролі охоронця та контролера здоров'я. Постійно перевіряє напругу, температуру, струм у комірках. Забезпечує роботу в безпечних межах, балансує елементи, оцінює [SOC/SOH](#) (стан заряду / стан здоров'я). Забезпечує надійний захист від небезпечних умов. Система керування енергоспоживанням ([EMS](#)): Інтелектуальне керування системою

 Знайдено посилань: **0,01%**

id: 6

«Мозок».

Збирає дані (ціни, мережа, використання, стан [BMS](#)) і повідомляє [PCS](#), коли і скільки заряджати/розряджати залежно від стратегії (економія, обслуговування мережі). Її складність — ключ до максимізації вартості. Баланс заводу енергії ([BOP](#)) та інфраструктура: Допоміжне обладнання для безпечної та надійної роботи: трансформатори, розподільні пристрої, проводка, терморегулювання (охолодження/нагрівання), пожежогасіння, корпуси. Ці компоненти утворюють тісно інтегровану систему для керування, виконання, контролю та підтримки операцій зі

зберігання енергії. За технологією та застосуванням можливо виділити наступні типи систем зберігання енергії: **ESS** охоплює різні технології та масштаби. Системи зберігання енергії в акумуляторах (**BESS**) посідають чільне місце. Таблиця 1.1 Технології зберігання енергії з поясненнями Тип технології Як зберігається енергія Типові приклади використання Основні міркування Акумуляторна Електрохімічні реакції Будинки, підприємства, комунальні послуги, зарядка **EV** Модульні, швидкі. Термін служби, безпека конструкції, вартість. Механічна Потенційна/кінетична енергія Великі комунальні служби (гідроелектростанції), системи швидкого реагування (маховики) Масштабні, довговічні. Залежить від географії. Термічна Тепло або холод у матеріалах Промисловість, великі сонячні електростанції, **HVAC** Для опалення/охолодження. Менше прямого використання електричної мережі. Хімічна Енергія хімічних зв'язків Концепції довготривалого зберігання (водень) Великий потенціал тривалого зберігання. Низька ефективність. **BESS** лідирує завдяки зниженню вартості літій-іонних акумуляторів та простоті впровадження. Ринок віддає перевагу літій-залізо-фосфату (**LFP**) для підвищення безпеки та терміну служби порівняно зі щільністю енергії, що є дуже важливим для мережевих/комерційних застосувань, порівняно з іншими літій-іонними (**NMC**) акумуляторами. Таблиця 1.2 Масштаб розгортання **ESS** Масштабування / розгортання Типовий користувач(і) Головна цінність Характеристики Житлова (**Home**) **ESS** Домовласники Нижчі рахунки (сонячна енергія, **TOU**), резервне копіювання Компактні, встановлюються вдома, часто з сонячними батареями. **C&I** (Комерційна та промислова) **ESS** Підприємства, фабрики Економіка пікового навантаження, економіка електроенергії, резервне копіювання Підбирається з урахуванням навантаження на об'єкт, окупність інвестицій. Масштаб мережі (**Utility-scale**) **ESS** Комунальні служби, оператори Стабільність мережі, інтеграція відновлюваних джерел енергії, потужність Дуже велика, підключена до високовольтної мережі. **ESS** приносить практичну користь завдяки конкретним застосуванням: Скорочення рахунків за електрику: (Економіка на піках та енергоспоживанні не є актуальною для мережевих інфраструктур). Найбільш переконлива економічна вигода. Скорочення дорогих платежів за потужність для підприємств за рахунок розрядження під час пікових навантажень (**Peak Shaving**). Використання дешевшої енергії, накопиченої в періоди низьких тарифів, у дорогі пікові періоди (оптимізація **TOU**), використовуючи правила комунальних тарифів. Забезпечення безперервності роботи: (Резервне живлення – головний фактор). Забезпечує критично важливу стійкість. Безперешкодно відключається від мережі під час перебоїв і негайно подає живлення на певні критичні навантаження для будинків/підприємств, які потребують безперебійної роботи. Часто працює швидше, ніж генератори. Максимізація інвестицій у сонячну енергію: (Використовується тільки у великих дата-центрах). Зберігайте надлишок полуденної сонячної енергії замість того, щоб експортувати її за низьким тарифом (кредитом). Використовуйте накопичену сонячну енергію пізніше (у вечірній пік), коли це необхідно (зсув сонячного часу), підвищуючи власне споживання та рентабельність сонячної енергії. Забезпечення підтримки та стабільності мережі: Комунальні **ESS** забезпечують важливі послуги: швидке регулювання частоти, підтримку напруги, швидке створення резервних потужностей, що життєво важливо для стабільності енергосистеми з більш мінливими відновлюваними джерелами енергії. Вони діють як

Знайдено посилань: **0,02%**

id: 7

«віртуальні електростанції».

Цінність очевидна – економія витрат, безперервність електропостачання, ефективність використання відновлюваних джерел енергії, стабільність мережі. Головний та суттєвий недолік – дуже велика вартість, фінансові затрати не будуть окупатися для малих дата центрів та серверних кімнат у ЗВО. Фактично ці рішення впроваджуються тільки на крупних підприємствах. 1.3 Промислові та модульні джерела безперебійного живлення (ДБЖ / **UPS**) для серверних приміщень У сучасну епоху глобальної цифрової трансформації, стрімкого розвитку хмарних обчислень та інтеграції штучного інтелекту в усі сфери людської діяльності безперервність бізнес-процесів стала безпосередньо залежною від стабільності IT-інфраструктури. Навіть секундне коливання напруги, короткочасне просідання чи повна відсутність електропостачання в мережі здатні призвести до колосальних фінансових збитків, незворотної втрати конфіденційних даних та фізичного пошкодження дорогого серверного й мережевого обладнання. У цих жорстких умовах традиційні монолітні системи захисту живлення, які десятиліттями панували на ринку, стрімко втрачають свою актуальність, поступаючись місцем інноваційним рішенням, серед яких промислові та модульні джерела безперебійного живлення (ДБЖ / **UPS**) стали безумовним галузевим стандартом для крупних серверних кімнат, крупних центрів обробки даних (ЦОД), медичних установ та автоматизованих виробництв. На відміну від класичних монолітних ДБЖ, які є єдиними неподільними блоками фіксованої потужності, сучасна модульна архітектура побудована за принципом гнучкого технологічного конструктора. В основі такої системи лежить спеціальна монтажна шафа (силовий кабінет), яка містить загальні шини розподілу живлення, інтерфейси управління та набір уніфікованих силових модулів, що працюють паралельно, рівномірно розподіляючи між собою загальне підключене навантаження. Це принципово змінює підхід до проектування систем гарантованого електропостачання, оскільки дозволяє відмовитися від складного

прогнозування розвитку IT-інфраструктури на десятиліття вперед, яке раніше змушувало компанії купувати дорогі надлишкові потужності

» Знайдено посилань: **0,02%**

id: 8

"із запасом",

що роками працювали вхолосту, марно споживаючи електроенергію та знижуючи загальну ефективність системи. Модульний підхід реалізує концепцію поступових інвестицій, відому як

» Знайдено посилань: **0,01%**

id: 9

"Pay-as-you-grow"

(Плати в міру зростання), коли на початковому етапі бізнес купує лише шафу з необхідною кількістю вільних слотів та мінімальним набором модулів для поточного споживання, а при встановленні нового серверного обладнання просто додає нові силові блоки протягом кількох хвилин без проведення масштабних електромонтажних чи будівельних робіт.

Рис.1.1 Модульні **UPS** Важливою перевагою такого рішення є радикальне підвищення ремонтпридатності за рахунок підтримки технології гарячої заміни (**Hot-Swap**), яка повністю нівелює ризики, пов'язані з плановим обслуговуванням або аварійними ремонтами. У класичних монолітних системах вихід з ладу одного внутрішнього компонента вимагає зупинки всієї системи або тривалого й ризикованого переведення критичного навантаження на обхідну лінію (байпас), піддаючи чутливе обладнання прямому впливу нестабільної зовнішньої мережі. У модульному ДБЖ будь-який силовий модуль можна безпечно вилучити зі слота прямо під час роботи всієї системи без зупинки процесів та без ризику для сервісного персоналу завдяки захищеним ізольованим роз'ємам. Це дозволяє досягти рекордно низького середнього часу відновлення працездатності (**Mean Time to Repair**), який скорочується з кількох годин чи навіть днів очікування сервісного інженера з паяльним обладнанням до банальної заміни несправного блоку на резервний протягом 5-10 хвилин звичайним персоналом IT-відділу. Надійність модульних ДБЖ також базується на гнучких схемах внутрішнього резервування за принципом $N+1$ або $N+X$, де N — це мінімальна кількість силових модулів, необхідна для забезпечення поточної потужності споживачів, а 1 або X — кількість надлишкових блоків. Якщо для живлення серверної кімнати потрібно 120кВт потужності, то замість встановлення двох монолітних ДБЖ по 120кВт кожен за надзвичайно дорогою схемою повного дублювання $2N$ (240кВт загальної встановленої потужності), замовник може обрати модульний кабінет із чотирма модулями по 40 кВт кожен, де три модулі покривають основне навантаження, а четвертий виступає як резервний. У разі виходу з ладу будь-якого з чотирьох блоків, решта три миттєво й безперервно перебирають навантаження на себе, гарантуючи нульовий час простою, при цьому вартість створення такої відмовостійкої системи зростає лише на ціну одного додаткового модуля (додаткові +33% до бюджету замість +100% у випадку монолітного дублювання). Сучасні модульні системи характеризуються безпрецедентно високою енергоефективністю, що безпосередньо впливає на сукупну вартість володіння (**Total Cost of Ownership**). Класичні ДБЖ демонструють свій максимальний коефіцієнт корисної дії (ККД) лише при високому завантаженні на рівні 70-90%, але оскільки більшість реальних серверних систем у середньому завантажені лише на 15-30%, ККД моноліту різко падає, призводячи до колосальних втрат енергії, яка перетворюється на тепло і вимагає додаткових витрат на кондиціонування приміщення. Натомість інтелектуальні модульні ДБЖ оснащені запатентованими алгоритмами розумного сплячого режиму (**Smart Sleep** або **Module Sleep**), які безперервно аналізують споживання та переводять частину силових модулів у режим глибокого очікування, залишаючи активними лише стільки блоків, скільки потрібно для забезпечення поточної потреби з урахуванням заданого резервування. Активні модулі при цьому працюють у точці свого максимального ККД, який у сучасних трифазних безтрансформаторних системах подвійного перетворення онлайн (**Double Conversion Online**) сягає вражаючих 96,5-97,5%, а у разі раптового сплеску споживання сплячі блоки миттєво активуються за частки мікросекунди без найменшого ризику для споживачів. Максимальній оптимізації піддається і фізичний простір, оскільки сучасні IT-приміщення та комерційні ЦОД мають надзвичайно високу вартість квадратного метра площі. Модульні ДБЖ відрізняються рекордною питомою щільністю потужності: замість розміщення величезних окремих шаф випрямлячів, інверторів, ручних байпасів та акумуляторних стійок, модульна система інтегрує силову частину, статичний байпас, інтелектуальний контролер та навіть батарейні лінійки в межах одного стандартного телекомунікаційного кабінету шириною 19 дюймів, звільняючи цінну площу для встановлення додаткових серверних стійок, що безпосередньо збільшує дохідність бізнесу. Крім того, перехід на модульні безтрансформаторні технології дозволив досягти одиничного вихідного коефіцієнта потужності (**Unity Power Factor**, $PF=1.0$), що означає повну відповідність активної потужності в кіловатах (кВт) повній потужності в кіловольт-амперах (кВа), дозволяючи підключати більше сучасних серверів із блоками живлення з компенсацією реактивної потужності без необхідності штучного заниження номінальних параметрів ДБЖ. Революційним кроком у розвитку модульних систем стало також активне

впровадження літій-іонних акумуляторних батарей (наприклад, на основі літій-залізо-фосфатної хімії [LIFEPO4](#)) як альтернативи класичним герметичним свинцево-кислотним акумуляторам ([VRLA](#)). Літій-іонні батареї мають у 3-4 рази довший термін служби (10-15 років порівняно з 3-5 роками для [VRLA](#)), здатні стабільно працювати при вищих температурах без деградації ємності, що дозволяє економити на охолодженні акумуляторної кімнати, мають у кілька разів меншу вагу та об'єм при аналогічній ємності, а також підтримують надшвидкий заряд до 80% ємності всього за годину, забезпечуючи високу готовність системи до повторних відключень зовнішньої мережі. Інтегрована інтелектуальна система управління батареями ([Battery Management System](#)) на рівні кожного окремого модуля безперервно контролює напругу, струм та температуру кожного гальванічного елемента, запобігаючи тепловому розгону та гарантуючи абсолютну пожежну безпеку промислового об'єкта. Широкий спектр ефективного практичного застосування модульних ДБЖ охоплює не лише ІТ-сферу, де гнучке масштабування є життєво необхідним для забезпечення нульового часу простою хмарних серверів у режимі 24/7/365. На сучасних високотехнологічних промислових підприємствах такі системи захищають складні автоматизовані лінії, робототехніку та чутливі контролери АСУ ТП, які є вкрай вразливими до імпульсних завад, гармонійних спотворень та перепадів напруги, що виникають через роботу потужного промислового обладнання та двигунів по сусідству. У сфері охорони здоров'я модульні ДБЖ гарантують безперебійне функціонування життєзабезпечувальної апаратури в операційних блоках та високоточних діагностичних комплексів, таких як магнітно-резонансні (МРТ) та комп'ютерні томографи (КТ), де навіть мікросекундне зникнення напруги може призвести до збою програмного забезпечення, втрати результатів дослідження чи створення загрози для життя пацієнта, а технологічна можливість проводити регулярне обслуговування ДБЖ без його повного вимкнення знімає потребу в зупинці роботи медичного центру. Сучасні промислові рішення також неможливо уявити без глибокої [IoT](#)-інтеграції та інтелектуального цифрового моніторингу, які перетворюють ДБЖ з пасивного заліза на проактивний елемент інфраструктури, що інтегрується в загальні системи управління будівлею ([Building Management System](#)) та програмні комплекси моніторингу дата-центрів ([Data Center Infrastructure Management](#)) за допомогою відкритих промислових протоколів [SNMP](#), [Modbus TCP/RTU](#) або хмарних сервісів. Завдяки вбудованим датчикам та алгоритмам предиктивної аналітики, система самостійно контролює життєвий цикл критично важливих компонентів, таких як фільтруючі електролітичні конденсатори, вентилятори охолодження та акумуляторні комірки, завчасно повідомляючи технічний персонал про необхідність проведення регламентних робіт задовго до того, як потенційне зношення деталей зможе спровокувати реальний аварійний збій. Додатковий екологічний моніторинг параметрів вологості та температури всередині шафи дозволяє точно прогнозувати термін служби батарей, захищаючи обладнання від перегріву чи утворення конденсату. Таким чином, перехід від застарілих монолітних концепцій до передових модульних рішень є не просто тимчасовою технологічною модою, а стратегічним, фінансово обґрунтованим бізнес-рішенням для крупних та потужних датацентрів, яке мінімізує операційні ризики, оптимізує капітальні витрати та закладає надійний фундамент для довгострокового динамічного розвитку будь-якої сучасної технологічної компанії чи промислового підприємства. Головний та суттєвий недолік – дуже велика вартість, фінансові затрати не будуть окупатися для малих дата центрів та серверних кімнат у ЗВО. Фактично ці рішення впроваджуються тільки на крупних підприємствах. Крім того, слід відзначити, що у таких системах використовується програмне забезпечення, що розраховано на моніторинг потужності та слабо орієнтовано на значні та часті відключення світла. Головна мета цього програмного забезпечення – це швидкий перерозподіл споживання енергії, а не керування процесами вмикання та вимикання серверних систем [ESX 1.4](#) Особливості джерел безперебійного живлення Джерела безперебійного живлення (ДБЖ, англ. [UPS](#) — [Uninterruptible Power Supply](#)) у сучасному технологічному ландшафті виконують роль фундаментального елемента відмовостійкості. Вони не просто забезпечують тимчасове живлення від акумуляторів під час аварійних відключень, а виступають першою лінією активної оборони для чутливого електронного обладнання.

Рис. 1.2 Системи [UPS](#) Сучасна система електропостачання піддається десяткам деструктивних впливів: від високочастотних завад і імпульсних сплесків, викликаних комутаційними процесами на підстанціях або ударами блискавок, до довготривалих просідань та гармонійних спотворень. Для [IT](#)-директорів, архітекторів дата-центрів та провідних інженерів глибоке розуміння топології, внутрішньої архітектури та фізичних процесів у ДБЖ є обов'язковою умовою для проектування надійних систем класу [Tier III](#) та [Tier IV](#). 1.4.1. Аналіз базових топологій ДБЖ Правильний вибір системи енергозахисту потребує чіткого аналізу класичної тріади топологій, які визначені міжнародним стандартом [IEC 62040- 3](#) [12]. Резервні ДБЖ ([Offline / Standby](#)) Це базовий клас пристроїв, призначений для індивідуального захисту обладнання з низьким рівнем критичності. У нормальному режимі роботи навантаження отримує живлення безпосередньо із зовнішньої мережі через пасивний фільтр електромагнітних завад і варисторний захист від перенапруг. Фізика процесу та перемикання: Коли вхідна напруга виходить за встановлені межі (наприклад, падає нижче 180В або перевищує 250В), внутрішній контролер фіксує

аварію і за допомогою механічного реле перемикає споживача на роботу від інвертора, що живиться від внутрішньої батареї. Технічні обмеження: Час перемикання становить від 4 до 12 мс. Це допустимо для більшості персональних комп'ютерів завдяки конденсаторам у їхніх імпульсних блоках живлення (так званий [hold-up time](#), який за стандартом [Intel ATX](#) має бути не менше 16мс). Проте для високочутливих промислових контролерів або вимірювальних приладів така затримка може бути критичною та призвести до перезавантаження. Форма сигналу: Більшість резервних ДБЖ при роботі від батареї генерують апроксимовану (ступінчасту) синусоїду. Це створює підвищене теплове навантаження на трансформаторні блоки живлення та електродвигуни через наявність високочастотних гармонік у вихідному сигналі. Лінійно-інтерактивні ДБЖ ([Line-Interactive](#)) Крок уперед у порівнянні з резервними системами. Головна відмінність цієї топології — наявність вбудованого автотрансформатора з перемикальними обмотками, відомого як блок автоматичного регулювання напруги ([AVR — Automatic Voltage Regulator](#)). Принцип стабілізації: Завдяки [AVR](#) пристрій здатний здійснювати східчасте коригування (підвищення або зниження) вхідної напруги на 10-15% без переходу на акумуляторний ресурс. Це значно подовжує життєвий цикл батарей, оскільки в умовах постійно заниженої напруги в мережі ДБЖ просто комутує відповідні обмотки автотрансформатора. Параметри перемикання: Час реакції реле [AVR](#) становить близько 2-6 мс. Сумісність із [Active PFC](#): Моделі середнього та вищого класу генерують чисту синусоїдальну напругу при роботі від батареї. Це надзвичайно важливо для сучасних серверів із блоками живлення, що мають активну корекцію коефіцієнта потужності ([Active PFC](#)). Використання ступінчастої синусоїди з [APFC](#)-блоками часто призводить до аварійного вимкнення ДБЖ через надвисокі пускові струми в момент перемикання. Онлайн-ДБЖ ([Online / Double-Conversion](#)) Технологія безперервної дії, розроблена для захисту інфраструктури критичного значення. Тут реалізовано концепцію повної ізоляції споживача від зовнішнього джерела живлення. Незалежно від стану вхідної мережі, навантаження отримує ідеально стабілізовані параметри струму, які генеруються локально внутрішнім інвертором.

1.4.2. Аналіз технологій подвійного та Дельта-перетворення

У сфері високих потужностей (від 10кВт до кількох мегават), яка охоплює промислове обладнання та великі дата-центри, ведеться конкуренція між двома основними підходами до побудови онлайн-систем. Онлайн-ДБЖ із подвійним перетворенням ([Double-Conversion](#)) Класична схема працює за послідовним принципом: Перший етап ([AC-DC](#)): Вхідний випрямляч приймає нестабільну змінну напругу з мережі, випрямляє її та згладжує пульсації. Одночасно з цим відбувається підзарядка акумуляторної батареї через інтегрований контролер. Другий етап ([DC-AC](#)): Постійний струм з шини [DC](#) надходить на інвертор, який за допомогою широтно-імпульсної модуляції (ШИМ) на базі швидкодіючих [IGBT](#)-транзисторів знову формує ідеальну синусоїду з фіксованою частотою (50-60Гц) та стабільним вольтажем. Архітектурна перевага: Повний розрив фази. Будь-які імпульсні перешкоди, гармоніки чи коливання частоти з боку електромережі фізично не можуть пройти через [DC](#)-ланку. Час перемикання на батарею дорівнює точно 0 мс, оскільки інвертор постійно підключений до шини постійного струму, де батарея перебуває в режимі очікування

Знайдено посилань: **0,01%**

id: **10**

«гарячого»

розряду. Енергетичні втрати: Оскільки 100% енергії проходить через два послідовні силові перетворювачі, загальний ККД системи історично становив близько 90-93%. У сучасних моделях преміум-класу за рахунок використання трирівневих інверторів цей показник вдалося підняти до 95-96.5%. Онлайн-ДБЖ із дельта-перетворенням ([Delta-Conversion](#)) Ця топологія була розроблена як технологічна альтернатива для оптимізації енергоспоживання у надпотужних системах. Замість того щоб повністю регенерувати весь обсяг потужності, система аналізує вектор вхідної напруги та компенсує лише відхилення від еталона. Принцип компенсації дельти: Спеціальний дельта-трансформатор, підключений послідовно між входом та виходом ДБЖ, додає або віднімає необхідну векторну величину напруги. Головний інвертор регулює вихідну напругу, а дельта-інвертор контролює вхідний коефіцієнт потужності, зводячи нелінійні спотворення струму практично до нуля. Економічний ефект: Оскільки через перетворювачі проходить лише частина загальної потужності (зазвичай не більше 15-20%, яка безпосередньо становить відхилення параметрів мережі від номіналу), теплові втрати різко знижуються. ККД таких систем стабільно тримається в діапазоні 97-98%. Технічний компроміс: Складніша схема керування та відсутність повної гальванічної розв'язки між вхідним фідером і навантаженням. У промислових умовах з високим рівнем індустриальних завад або при роботі від нестабільних дизель-генераторів відсутність фізичної ізоляції може призвести до проникнення високочастотного шуму на вихід ДБЖ, що вимагає встановлення додаткових ізолювальних трансформаторів.

1.4.3. Системні характеристики сучасних Онлайн-ДБЖ

Сучасне покоління онлайн-систем безперебійного живлення вийшло далеко за межі простих джерел аварійного струму. Вони є складними інтелектуальними комплексами. Чиста синусоїдальна напруга Високоточна цифрова техніка, медичне обладнання (наприклад, МРТ-сканери), а також сервери з блоками живлення високої щільності вимагають стабільного коефіцієнта нелінійних спотворень напруги ([THDu](#) 2%). Онлайн-

ДБЖ гарантують гладку синусоїдальну форму сигналу при будь-яких типах навантаження — як активному (резистивному), так і реактивному (індуктивному/ємнісному). Це нівелює ризик виникнення паразитних вихрових струмів та перегріву магнітопроводів у підключеному обладнанні. Статичний та сервісний байпас ([Bypass](#)) Незважаючи на високу надійність онлайн-систем, критично важливо мати механізми обходу на випадок перевантаження або внутрішньої аварії: Статичний (електронний) байпас: Швидкодіє тиристорний перемикач, який здатний за частки мікросекунди закоротити вхід із виходом у разі короткого замикання в навантаженні або виходу з ладу внутрішнього інвертора ДБЖ. Це рятує критичні споживачі від знеструмлення, переводячи їх на пряме живлення від мережі. Ручний (сервісний) байпас: Механічний перемикач, що дозволяє повністю ізолювати внутрішні силові блоки ДБЖ від напруги для проведення регламентного технічного обслуговування або заміни плат без зупинки роботи підключеного дата-центру. Масштабованість та схеми резервування ($N+X$) Сучасна [IT](#)-інфраструктура будується за принципом поетапного розширення. Модульні трифазні ДБЖ дозволяють нарощувати потужність шляхом додавання уніфікованих силових блоків у загальну раму (шасі) у режимі [Hot-Swap](#) (гаряча заміна): Схема $N+1$: Передбачає встановлення одного додаткового модуля понад мінімально необхідну кількість (N). Наприклад, якщо для роботи [IT](#)-навантаження потрібно 100кВт, система комплектується трьома модулями по 50 кВт кожен (100 кВт+50 кВт резерву). У разі виходу з ладу будь-якого модуля, два інших безперешкодно розподіляють навантаження між собою, зберігаючи повну працездатність системи. Схема $2N$ (або $2(N+1)$): Створення двох абсолютно незалежних паралельних ліній живлення, кожна з яких здатна самостійно забезпечити 100% потужності об'єкта. Цей підхід усуває будь-які єдині точки відмови ([Single Points of Failure](#)) в інфраструктурі електропостачання. 1.4.4. Еволюція акумуляторних систем: [VRLA](#) проти [LiFePO4](#) Невід'ємною складовою будь-якої системи безперебійного живлення є накопичувач енергії. Останніми роками в індустрії відбувається тектонічний зсув від класичних рішень до літєвих технологій. Таблиця 1.3 Технології акумуляторів Характеристика Свинцево-кислотні ([VRLA / AGM](#)) Літій-залізо-фосфатні ([LiFePO4](#)) Термін служби (календарний) 3-5 років (для стандартних серій) 10-15 років Кількість циклів (при 80% [DoD](#)) Температурна чутливість Оптимально 20С. Кожні +10С зменшують ресурс вдвічі Стабільна робота в діапазоні від 0С до +40С без деградації Вага та габарити Важкі, потребують посиленних перекриттів підлоги На 60% легші та займають до 40% менше площі Система моніторингу Потребує зовнішніх датчиків для детальної діагностики Інтегрована інтелектуальна система керування ([BMS](#)) Хоча початкові капітальні інвестиції ([CAPEX](#)) у літій-залізо-фосфатні батареї є вищими, загальна вартість володіння ([TCO](#)) на горизонті 10 років виявляється значно нижчою. Це досягається за рахунок відсутності потреби у регулярній заміні акумуляторних масивів кожні кілька років та зниження витрат на кондиціонування приміщень акумуляторних залів. 1.4.5. Переваги впровадження Онлайн-ДБЖ у критичній інфраструктурі Мінімізація апаратних ризиків та оптимізація [TCO](#): Повна стабілізація параметрів струму запобігає мікроруйнуванням напівпровідникових кристалів у блоках живлення серверів та мережевих комутаторів. Інженерна статистика доводить, що використання високоякісного онлайн-живлення знижує рівень відмов заліза на 25-30%, що безпосередньо подовжує термін експлуатації обладнання. Гарантія цілісності даних: Раптове зникнення напруги навіть на кілька мікросекунд здатне перервати транзакцію у базі даних, порушити цілісність [RAID](#)-масивів або пошкодити системні файли під час запису з кешу. Нульовий час перемикання онлайн-ДБЖ повністю нівелює цей ризик, гарантуючи безперервність запису. Енергетична та фінансова передбачуваність: Модульна архітектура сучасних трифазних онлайн-систем дозволяє масштабувати енергетичний захист паралельно з ростом фізичної інфраструктури компанії. Це виключає необхідність

” Знайдено посилань: **0,01%**

id: 11

"заморожувати"

великі кошти у надлишковій потужності ДБЖ на перших етапах будівництва об'єкта, забезпечуючи оптимальне планування капітальних інвестицій. Головний та суттєвий недолік – велика вартість, фінансові затрати не можуть запроваджуватися в умовах військового стану. Особливості роботи серверів на базі [ESX](#) потребують більш гнучкого програмного забезпечення, якого не має у сучасних ДБЖ. Фактично ці рішення можуть бути запроваджені у серверних кімнатах ЗВО. Але аналіз останнього періоду подавання електроенергії в умовах дуже частих (декілька разів на день) та тривалих (до 5 годин) відключень свідчить, що потрібно використовувати більш гнучке програмне забезпечення, яке буде ефективно вимокати та вимикати серверні потужності в залежності від критичності ситуацій у системі електроживлення у невеликих підприємствах. Висновки до розділу У першому розділі роботи проведено детальний аналіз сучасних технологічних рішень у сфері забезпечення безперервного та якісного електропостачання критичної IT-інфраструктури, крупних обчислювальних центрів та серверних кімнат. На основі аналізу міжнародних стандартів, зокрема [ANSI/TIA-942](#), розглянуто архітектуру, функціональні можливості та ключові компоненти промислових систем зберігання енергії ([ESS](#)) та сучасних модульних джерел безперебійного живлення (ДБЖ) онлайн-типу із подвійним перетворенням. За результатами проведеного аналітичного огляду можна зробити такі

основні висновки: Сучасні модульні ДБЖ та промислові системи [ESS](#) (особливо на базі літій-залізо-фосфатних акумуляторів [LiFePO₄](#)) забезпечують найвищий рівень надійності, пропонуючи гнучке масштабування за принципом

Знайдено посилань: **0,01%**

id: 12

"Pay-as-you-grow"

, внутрішнє резервування потужності за схемами $\$N+1\$$ або $\$N+X\$$, високий ККД завдяки інтелектуальним режимам спячки ([Module Sleep](#)), а також глибоку [IoT](#)-інтеграцію для предиктивного моніторингу. Проте, головним недоліком таких комплексів є їхня надзвичайно висока ринкова вартість (як капітальні витрати [CAPEX](#), так і операційні [OPEX](#)). Впровадження повноцінних промислових та модульних шаф потребує значних фінансових інвестицій, які є непосильними для багатьох організацій та установ. З огляду на високу вартість брендових систем гарантованого живлення (таких як [APC](#), [Eaton](#), [Vertiv](#) тощо), виникає гостра потреба в розробці альтернативних, фінансово доступних систем керування. Ефективним шляхом розв'язання цієї проблеми є перенесення інтелектуальних функцій керування на рівень наявної мережевої інфраструктури. Сучасне мережеве обладнання, зокрема маршрутизатори компанії [MikroTik](#) під управлінням [RouterOS](#), має потужні вбудовані можливості програмування та автоматизації (інструменти [Netwatch](#), мова скриптів, підтримка контейнеризації або виконання запитів [API](#)). Використання програмованих роутерів як центральних контролерів дозволяє реалізувати інтелектуальні алгоритми моніторингу стану мережі та керування живленням серверів без необхідності закупівлі спеціалізованих дорогих систем автоматизації та [EMS \(Energy Management Systems\)](#). В умовах тривалого воєнного стану в Україні, нестабільної роботи об'єднаної енергетичної системи через постійні руйнування інфраструктури, проблема безперебійної роботи серверів стала критичною. Водночас невеликі підрозділи (локальні державні установи, освітні заклади, волонтерські та гуманітарні центри, малі підприємства) з обмеженим бюджетом не мають коштів на придбання промислових рішень. Для таких об'єктів, які зазвичай експлуатують невелику кількість серверних стійок або окремих машин, критично важливо забезпечити максимальну тривалість автономної роботи від звичайних побутових або комерційних систем безперебійного живлення. Оскільки єдиним безкоштовним та ефективним ресурсом у таких умовах є оптимізація споживання, існує реальна необхідність у розробці нової інтелектуальної системи автоматизованого каскадного та пріоритетного керування живленням серверів. Така система за допомогою скриптів на базі роутера [MikroTik](#) (використовуючи [Netwatch](#) або [Ping](#)-моніторинг зовнішніх вузлів та шлюзів) зможе: у реальному часі визначати факт зникнення зовнішньої мережі 220В або критичного падіння ємності локальних батарей; через мережу (протоколи [SSH](#), [Winbox API](#), [SNMP](#) або [VMware vCenter API](#)) автоматично, за заздалегідь визначеним пріоритетом, коректно зупиняти другорядні віртуальні машини, тестові середовища та менш важливі сервери; консервувати енергію акумуляторів виключно для функціонування критично важливих сервісів підрозділу (бази даних, зв'язок, системи безпеки). Таким чином, мета бакалаврської роботи — розробка бюджетної, але високоефективної системи керування електроживленням серверів на базі програмованих можливостей сучасного мережевого обладнання [MikroTik](#) — є повністю обґрунтованою, актуальною та має високу практичну цінність для забезпечення живучості ІТ-інфраструктури в умовах військового стану та обмеженого фінансування. Розділ 2

Розробка системи керування електроживленням для серверів [Vmware ESX v7](#) Змн. Арк. № докум. Підпис Дата Арк. 3 ІТС.4КІ.0624.03-ПЗ Розроб. Семесенко І. Керівник Могильний Г.А. В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. Розділ 2 Літ. Акрушів 3 ЛНУ Кафедра ІТС, Гр.4КІ. 2.1 Аналіз станів електроживлення та попередні налаштування Розглянемо особливості роботи серверів. За основу будемо брати інформаційну структуру невеликого підрозділу, у якій 2–6 Хостів, [ESXi 7.3](#), [vCenter 7.3u](#), активний [DRS](#), але вимкнений [HA](#), має дуже специфічну поведінку в умовах раптового зникнення живлення та наступного відновлення. Оскільки [HA \(High Availability\)](#) вимкнено, у разі падіння одного з хостів [Dell](#) або [Supermicro](#) через проблеми з живленням, його віртуальні машини не будуть автоматично перезапущені на інших виживілих хостах. Вони просто

Знайдено посилань: **0,01%**

id: 13

"помруть"

разом із залізом. [DRS \(Distributed Resource Scheduler\)](#) відповідає лише за балансування навантаження працюючих ВМ (через [vMotion](#)) і не вміє реанімувати вимкнені машини. Нижче наведено технічні особливості налаштування [BIOS](#) для серверів [Dell PowerEdge](#) та [Supermicro](#), а також архітектурні нюанси, які необхідно внести до наукової роботи. 2.1.1. Специфічні налаштування [BIOS/UEFI](#) для серверів На відміну від десктопної [Windows](#), для серверів [ESXi](#) режим [WoL](#) зазвичай вважається антипаттерном через високі вимоги до безпеки та надійності. Керування стартом тут реалізується через [IPMI \(Dell iDRAC / Supermicro IPMI/BMC\)](#) або автоматичний запуск. Для серверів [Dell PowerEdge](#) (керування через [iDRAC](#)) у [BIOS \(System Setup - System BIOS\)](#) необхідно виставити: [AC Power Recovery](#): Встановити у значення [Last](#) або [On](#). Таким чином, якщо [UPS](#) вимкнувся, а потім відновив подачу живлення на стійку, сервери мають ініціювати запуск автоматично (або

повернутися до останнього стану). [AC Power Recovery Delay](#): Вистави [Immediate](#) для першого сервера (де розгорнуто [vCenter](#)) та [Random](#) або [Fixed Delay](#) (наприклад, 60, 120 сек) для решти хостів. Це запобігає одночасному пусковому току ([Inrush Current](#)) на [UPS](#) та каскадує старт. [Deep Sleep Mode](#): Обов'язково [Disabled](#). Серверний блок живлення повинен постійно живити плату [iDRAC](#), щоб ти міг керувати сервером віддалено в будь-який момент. Для серверів [Supermicro](#) (керування через [IPMI/BMC](#)) у [BIOS \(Advanced - ACPI Settings / IPMI Configuration\)](#): [Restore on AC Power Loss](#): Встанови у значення [Last State](#) або [Power On](#). [Power Button Function](#): 4 [Seconds Override](#) (для безпечного примусового вимкнення залізом, якщо ОС зависне). [Network Stack](#): Якщо планується запуск через [iPXE](#)/мережу — [Enabled](#), інакше [Disabled](#) для прискорення часу проходження [POST](#). Сервери [Supermicro](#) відомі своїм тривалим [POST](#)-тестом (іноді до 5–7 хвилин), що критично враховувати в таймінгах [UPS](#). Таким чином, для автоматизації старту хостів [ESXi](#) та автоматичного відновлення живлення було проведено переналаштування [BIOS](#). У [BIOS](#) було встановлено автоматичне відновлення живлення та підтримку пакетів [WOL](#) з метою подачі команди від роутера [MikroTik](#). Слід відзначити, що у багатьох серверів, залежно від виробника та моделі сервера ці налаштування розташовані у різних меню [BIOS](#). В інформаційній системі, що досліджувалась налаштування параметра

» Знайдено посилань: **0,04%**

id: 14

«[Restore on AC Power Loss](#)»

зображено на рис. 2.1. Налаштування параметру [WOL](#) на сервері [DELL](#) зображено на рисунках 2.2-4, а для сервера [SUPERMICRO](#) на рисунках 2.5-7.

а)

б)

Рис. 2.1. Налаштування параметра

» Знайдено посилань: **0,04%**

id: 15

«[Restore on AC Power Loss](#)»

а) сервер [DELL](#)

б) сервер [SUPERMICRO](#)

Рис. 2.2. Меню [BIOS](#) сервера [DELL](#)

» Знайдено посилань: **0,02%**

id: 16

«[Device Setting](#)»

Рис. 2.3. Меню [BIOS](#) сервера [DELL](#) параметри адаптера

Рис. 2.4. Меню [BIOS](#) сервера [DELL](#) параметр [WOL](#)

Рис. 2.5. Меню [BIOS](#) сервера [Supermicro Advanset](#)

Рис. 2.6. Меню [BIOS](#) сервера [Supermicro](#) параметри адаптера

Рис. 2.7. Меню [BIOS](#) сервера [Supermicro](#) параметр [WOL](#) 2.1.2. Особливості процесу вимкнення ([Shutdown Sequence](#)) Оскільки [vCenter](#) 7.3 керує [DRS](#), процес зупинки є делікатним. Якщо просто

» Знайдено посилань: **0,02%**

id: 17

"висмикнути розетку"

або

» Знайдено посилань: **0,01%**

id: 18

"погасити"

хости одночасно, [DRS](#) може заблокувати міграції, а база даних [vCenter](#) отримає пошкодження ([data corruption](#)). Таким чином, треба урахувати наступні особливості Поетапне вимкнення ВМ ([Guest OS](#)): Спочатку вимикаються некритичні ВМ, потім інфраструктурні (Контролери домену, [DNS](#), Бази даних) на останньому етапі всі інші віртуальні машини. Визначення локації [vCenter](#): Оскільки [HA](#) вимкнено, [vCenter Server Appliance \(VCSA\)](#)

» Знайдено посилань: **0,01%**

id: 19

"живе"

на конкретному фізичному хості. Скрипт має знати або йому необхідно призначити хост, на якому саме зараз запущено [vCenter](#). Вимкнення [vCenter](#) та основних інфраструктурних машин: [vCenter](#) зупиняється останнім або передостаннім серед усіх ВМ, безпосередньо перед зупинкою самого хоста, на якому він запущений. Переведення хостів [ESXi](#) в [Maintenance Mode](#) або зупинка: Виконується команда вимкнення хостів через [SSH \(poweroff\)](#) або [IPMI API](#). 2.1.3. Особливості процесу старту ([Boot Sequence](#)) Тут виникає головна проблема конфігурації без [HA](#): після відновлення живлення хости завантажуються, але ВМ не увімкнуться самі, тому що [vCenter](#) лежить вимкненим, а [DRS](#) без [HA](#) не знає, що треба щось запускати. Рішення та налаштування: Щоб система піднялася автономно, необхідно налаштувати функцію [VM Startup/Shutdown](#) безпосередньо на рівні самого хоста [ESXi](#) (через

Host Client, а не **vCenter**). Локальний **Auto-Start** на **ESXi**: На кожному хості, потрібно зайти в налаштування хоста - **Manage** - **Autostart** і увімкнути його. Пріоритет для **vCenter** та інфраструктурних машин (Контролери домену, **DNS**, Бази даних, **WEB** сервіси): Внести ці **VM** та **vCenter** у список автозапуску цього конкретного хоста з найвищим пріоритетом (**Priority 1-N**). Затримка запуску (**Start Delay**): Виставити затримку запуску для інших **VM** (наприклад, 300 секунд), щоб **vCenter** та інфраструктурні машини встигли повністю завантажити свої сервіси (сервіси **vCenter 7.3** піднімаються близько 3-8 хвилин). Ввімкнення решти інфраструктури: Як тільки **vCenter** став доступним, зовнішній скрипт або заплановане завдання розішле команди на запуск решти віртуальних машин по кластеру. Таких етапів може бути декілька. В досліджуемій інфраструктурі запропоновано 2 додаткові етапи. В цілому діаграму станів зображено на малюнку 2.8

Рис. 2.8 Діаграма стану для серверної інфраструктури У запропонованій діаграмі станів процесу старту/завершення інформаційної системи перевага надається процесу завершення. Оскільки некоректне завершення може привести до збою роботи всієї системи в цілому. Головна ідея – поетапний старт віртуальних машин з постійною перевіркою електроживлення після кожного етапу. У випадку зникнення живлення розпочинається етап завершення роботи системи, який йде до повного вимкнення всіх віртуальних машин, завершення роботи хостів та не може бути перерваним. Таким чином, за відсутності координації софт-комплексу **VMware**, модель системи керування живленням (мережевий роутер) повинна брати на себе роль координатора станів. Вона має жорстко прив'язувати послідовність подачі живлення на сервери (**Dell/Supermicro**) до архітектурного розташування керуючого вузла (**vCenter**) та використовувати локальні механізми **ESXi Autostart** для подолання обмежень вимкненого **HA**-функціоналу. 2.2 Передумови використання **Mikrotik** Оскільки **MikroTik** має власний **UPS** (або живиться від надійної лінії), він не залежить від стану серверів і може за допомогою скриптів (**/system script**), планувальника (**/system scheduler**), утиліти **Netwatch** та інструментів **tool fetch** повністю координувати сервера **Dell**, **Supermicro** та **VM vCenter**. Нижче наведено особливості та логіку керування такою інфраструктурою безпосередньо з **RouterOS**.

2.2.1.Процес виключення Коли **MikroTik** отримує сигнал про зникнення електроенергії (або через **USB/SNMP**, або через **Netwatch**, фіксуючи зникнення живлення), він покроково гасить інфраструктуру через мережу. Оскільки **HA** вимкнено, **MikroTiku** треба вказати точно, де лежить **vCenter** та головні інфраструктурні віртуальні машини. Оскільки **vCenter 7.3** працює як віртуальна машина (**VCSA**), **MikroTik** може виконати команду поетапного вимкнення всіх віртуальних машин та самого **vCenter** через **API** або **SSH**. Після того, як **vCenter** згасне, **MikroTik** починає завершувати роботу самих хостів **ESXi**. •Для **Dell (iDRAC)**: **MikroTik** надсилає команду через **SSH** (або **RACADM API** безпосередньо на **IP**-адресу **iDRAC**) **poweroff** через **SSH** безпосередньо на самі хости **ESXi** •Для **Supermicro (IPMI)**: Використовується мережевий запит до розширення **IPMI** (через **SMCIPMITool**, якщо є проміжний легкий лінукс-вузол) або надсилання команди **poweroff** через **SSH** безпосередньо на самі хости **ESXi** 7.3. У пропонуемій системі вибрано, що для всіх серверів використовувати первні команди через **SSH** безпосередньо на самі хости **ESXi** 7.3. 2.1.2.Процес включення Оскільки функція **HA** вимкнена, сервери після появи світла самі не піднімуть віртуальні машини в правильному порядку. Роутер робить це каскадно за допомогою вбудованої утиліти **/tool wol**. **MikroTik** фіксує, що живлення відновилося. Скрипт запускає таймер затримки, щоб переконавшись, що мережа живлення стабільна і світло не зникне знову через хвилину. Щоб не перевантажити **UPS** пусковими токами, **MikroTik** вмикає сервери по черзі за їхніми **MAC**-адресами: Фрагмент коду # Вмикаємо перший хост (де лежить **vCenter**) **/tool wol mac=11:22:33:44:55:66 interface=bridge-local :delay 180s** # Вмикаємо другий хост **/tool wol mac=AA:BB:CC:DD:EE:FF interface=bridge-local** Коли хост **ESXi** завантажився (**MikroTik** бачить його по **ping**), локальний механізм **ESXi Autostart** автоматично вмикає інфраструктурні, основні віртуальні машини та **vCenter 7.3**. Потім з певною затримкою, поетапно інші віртуальні машини. Впровадження концепції

” Знайдено посилань: **0,06%**

id: **20**

"Мережевий роутер як оркестратор живлення обчислювального кластера"

для кластерів, **RouterOS** може виступати зовнішнім детермінованим автоматом станів. Оскільки операційна система роутера є монолітною та надзвичайно відмовостійкою, вона забезпечує вищу надійність керування, ніж якби скрипти крутилися всередині самого кластера (який ми якраз і гасимо). 2.3 Основні складові системи Загальні засади створення цієї системи викладено у роботі [13]. До складу мережевої інфраструктури, яка входить до предмету дослідження, належать: пороговий роутер **MikroTik RB750Gr3** з **RouterOS 6.49** [14], який під'єднаний до мережі Інтернет та двох окремих мереж: гостьової та основної, в кожній з яких є проміжні комутатори. Два сервери (**DELL** та **SUPERMICRO**) приєднані до обох мереж, та з'єднані між собою адаптерами (10 Гбіт) для швидкого трафіку **vSAN** та **vMotion** (рис.2.9).

Рис. 2.9 Сервера ЛНУ імені Тараса Шевченка У процесі попереднього аналізу запропоновано, що основні етапи та завдання створення системи керування живленням можна звести до таких кроків: 1.Провести розподіл віртуального середовища на етапи старту, наприклад, мінімальний, базовий та повний рівень. 2.Дослідити споживання

електричної потужності, час її відновлення та періоди затримки між стартами компонентів віртуального середовища. 3.Розробити процедури (скрипти) для роутера ([MikroTik](#)), які дозволять керувати поетапним (каскадним) стартом та завершенням роботи активних віртуальних машин. У першу чергу потрібно виділити етапи старту та завершення інформаційної системи. В інформаційній системі, що розглядається, було виділено три етапи: •Мінімальний – сервер [vCenter](#)[15], сервери з підтримкою [MS AD](#) та [Web](#)-сервер. •Базовий – охоплює мінімальний та додатково файловий сервер та сервер [RDP](#). •Повний – включає базовий та додатково сервер [VMware Horizon Connection Server](#) та сервер [VMware Unified Access Gateway](#)[16]. Для дослідження, було задіяно побутову зарядну станцію [Bluetti](#) (рис. 2.10) потужністю 2000 [W](#).

Рис. 2.9 Побутова зарядна станція Практичні випробування виявили певні нюанси в експлуатації цієї зарядної станції. Її вихідна потужність заряджання становить близько 500 [W](#), тоді як сумарне споживання двох серверів коливається в межах 400–800 [W](#) залежно від поточного навантаження. Відповідно, за умови повного живлення обох серверів від станції, її акумулятор не встигає відновлюватися і плавно розряджається. Цю проблему розв'язали завдяки наявності двох блоків живлення (БЖ) у кожному сервері. Один БЖ кожного сервера підключили до зарядної станції, а інший — напряму до мережі 220 [V](#). Таке рішення дозволило знизити навантаження на станцію (за наявності світла) до 270–400 [W](#), завдяки чому батарея повністю заряджається за 4 години. У разі знеструмлення сервери повністю переходять на живлення від станції, а швидкість розряджання її акумулятора становить 35 %/год. Визначені параметри дають змогу точно розрахувати час автономної роботи серверного обладнання. 2.4 Загальна послідовність роботи скриптів роутеру [MikroTik](#) Відомо, що загальний час старту/завершення серверної інфраструктури зі значною кількістю віртуальних машин може тривати 20-25 хвилин. Протягом цього часу живлення може зникати декілька разів. На засадах діаграми станів прийнято рішення, що перевага віддається процесу завершення системи та пропонується наступна послідовність дій (рис.2.10): Роутер [RB750Gr3](#) з [RouterOS](#) 6.49 відстажує наявність електроживлення за допомогою системи [NetWach](#), яка запускає основні скрипти

» Знайдено посилань: **0,01%** id: **21**

«старту»

або

» Знайдено посилань: **0,01%** id: **22**

«завершення»

системи. Основний скрипт

» Знайдено посилань: **0,01%** id: **23**

«завершення»

реалізує логіку Відзначити у системі подію

» Знайдено посилань: **0,02%** id: **24**

«йде завершення»

Якщо подія

» Знайдено посилань: **0,02%** id: **25**

«йде завершення»

вже існує -перериває роботу - це другий запуск процесу

» Знайдено посилань: **0,01%** id: **26**

«завершення»,

якщо подія

» Знайдено посилань: **0,02%** id: **27**

«йде старт»

також існує видаляє подію

» Знайдено посилань: **0,02%** id: **28**

«йде старт»

Підтримка працездатності певний час, у випадку

» Знайдено посилань: **0,04%** id: **29**

«світло відновлено – йде старт»

– видаляє подію

» Знайдено посилань: **0,02%** id: **30**

«йде завершення»

та завершує роботу, у випадку

» Знайдено посилань: **0,07%** id: **31**

«світло не відновлено, немає події – йде старт»

розпочинає поетапний процес завершення ВМ третього етапу з певними затримками та викликає додаткові скрипти . розпочинає поетапний процес завершення ВМ другого етапу з певними затримками та викликає додаткові скрипти. розпочинає поетапний процес завершення ВМ першого етапу з певними затримками та викликає додаткові скрипти. розпочинає поетапний процес завершення всіх активних ВМ з певними затримками та викликає додаткові скрипти. Видаляє подію

” Знайдено посилань: **0,02%** id: **32**

«йде завершення»

Основний скрипт

” Знайдено посилань: **0,01%** id: **33**

«старту»

реалізує логіку Відзначити у системі подію

” Знайдено посилань: **0,02%** id: **34**

«йде старт»

Якщо подія

” Знайдено посилань: **0,02%** id: **35**

«йде старт»

вже існує -перериває роботу - це другий запуск процесу

” Знайдено посилань: **0,01%** id: **36**

«старт»

Якщо є подія

” Знайдено посилань: **0,02%** id: **37**

«йде завершення»

- чекає коли подія зникне - головне коректне

” Знайдено посилань: **0,01%** id: **38**

«завершення»,

перевіряє наявність події

” Знайдено посилань: **0,02%** id: **39**

«йде старт»,

якщо HI - завершує роботу - світло знов вимкнули Викликає додатковий скрипт

” Знайдено посилань: **0,02%** id: **40**

«перший етап»

старту, який з певними затримками подає пакет **Wol** на всі сервери **ESX** Якщо є подія

” Знайдено посилань: **0,02%** id: **41**

«йде завершення»

- перериває роботу - головне коректне

” Знайдено посилань: **0,01%** id: **42**

«завершення»,

видаляє подію

” Знайдено посилань: **0,02%** id: **43**

«йде старт»

та завершує роботу Викликає додатковий скрипт

” Знайдено посилань: **0,02%** id: **44**

«другий етап»

старту, який з певними затримками запускає віртуальні машини Якщо є подія

” Знайдено посилань: **0,02%** id: **45**

«йде завершення»

- перериває роботу - головне коректне

” Знайдено посилань: **0,01%** id: **46**

«завершення»,

видаляє подію

” Знайдено посилань: **0,02%** id: **47**

«йде старт»

та завершує роботу Викликає додатковий скрипт

”

Знайдено посилань: 0,02%	id: 48
«третій етап»	
старту, який з певними затримками запускає віртуальні машини Видаляє подію	
Знайдено посилань: 0,02%	id: 49
«Йде старт»	
Рис. 2.10 Послідовність роботи скриптів 2.5 Особливості створення скриптів роутеру MikroTik Загалом було створено 6 скриптів для старту та 8 для завершення (рис. 2.11). Рис. 2.11 Скрипти для автоматизації старту/завершення Текст головного скрипту для старту наведено у Додатку А Текст головного скрипту для завершення наведено у Додатку Б Текст першого допоміжного скрипту для старту наведено у Додатку В Текст першого допоміжного скрипту для завершення наведено у Додатку Д Текст інших допоміжних скриптів для старту наведено у Додатку Е Текст інших допоміжних скриптів для завершення наведено у Додатку Є До основних особливостей відноситься наступне. 1. Щоб фіксувати появу та зникнення електроживлення, достатньо налаштувати утиліту Netwatch на моніторинг будь-якого пристрою в гостьовій мережі, який живиться напругу (без ДБЖ). Налаштування здійснюється через меню Tools - Netwatch або в терміналі командою: <code>/tool netwatch add host= IP_addr_quest net timeout= sec interval= sec</code> Після цього роутер автоматично визначатиме зміни стану: відновлення живлення запускатиме скрипт Up, а вимкнення — скрипт Down. (рис. 2.12). Netwatch скрипти мають обмеження у розмірі та розглядаються як один рядок. Тому створюйте окремі скрипти (підскрипти) та обов'язково завершуйте команду символом	
Знайдено посилань: 0,02%	id: 50
<;>	
Для запуску скрипта (підскрипта) у фоновому режимі скористайтесь командою - <code>:execute name_script</code> ; (рис. 2.12) скрипт Up : log info	
Знайдено посилань: 0,02%	id: 51
"Event UP-----"	
; :execute script=	
Знайдено посилань: 0,01%	id: 52
"StartSum"	
; скрипт Down : log info	
Знайдено посилань: 0,02%	id: 53
"Event DOWN-----"	
; :execute script=	
Знайдено посилань: 0,01%	id: 54
"DownSum"	
;	
Рис. 2.12 Створення Netwatch 2. При створенні додаткових скриптів (підскриптів) врахуйте певні привілеї (Policy) скриптів. Їх повинно бути чотири: read, reboot, write, test. Будь-які інші комбінації Policy недопустимі (рис. 2.13). Рис. 2.13 Створення скрипта 3. У скриптах для запуску ESXi (перший етап старту) скористайтесь командою <code>/tool wol mac= MAC_ADDR_ESXi interface= Name_intf_connect_ESXi</code>; 4. Скрипти не мають параметрів. Для передавання параметрів скористайтесь Address Lists. Таким чином, для передавання параметрів додайте інформацію у Address Lists з певним ім'ям, у скрипті зчитайте інформацію з Address Lists та за необхідністю видаляйте Address Lists. Наприклад, для роботи з Address Lists з ім'ям powerDOWN: отримання IP адреси <code>/ip firewall address-list get [find list=</code>	
Знайдено посилань: 0,01%	id: 55
"powerDOWN"	
] address; знищення Address Lists з ім'ям powerDOWN - <code>/ip firewall address-list remove [find where list=</code>	
Знайдено посилань: 0,01%	id: 56
"powerDOWN"	
]; додавання до Address Lists з ім'ям powerDOWN ще однієї IP адреси <code>/ip firewall address-list add list=</code>	
Знайдено посилань: 0,01%	id: 57
"powerDOWN"	
address =127.0.0.1; додати подію	
Знайдено посилань: 0,02%	id: 58
«Йде завершення»	

/ip firewall address-list add list=	id: 59
Знайдено посилань: 0,01%	
"powerDOWNrun"	
address=127.0.0.1; видалити подію	
Знайдено посилань: 0,02%	id: 60
«Йде завершення»	
/ip firewall address-list remove [find where list=	
Знайдено посилань: 0,01%	id: 61
"powerDOWNrun"	
]; з'ясувати чи є подія	
Знайдено посилань: 0,02%	id: 62
«Йде завершення»	
(0) :if ([:len [/ip firewall address-list find where list=	
Знайдено посилань: 0,01%	id: 63
"powerDOWNrun"	
] 0) do={ } додати подію	
Знайдено посилань: 0,02%	id: 64
«Йде старт»	
/ip firewall address-list add list=	
Знайдено посилань: 0,01%	id: 65
"powerUPrun"	
address=127.0.0.1; чекати зникнення події	
Знайдено посилань: 0,02%	id: 66
«Йде завершення»	
:local counter 0; :while (([:len [/ip firewall address-list find where list=	
Знайдено посилань: 0,01%	id: 67
"powerDOWNrun"	
] 0)) do={ :set counter (\$counter + 1); :log info (	
Знайдено посилань: 0,06%	id: 68
"Waiting event - powerDOWNrun to clear... : "	
. \$counter); :delay 10s; } 5. У скриптах, для підключення до хостів ESXi та виконання певної команди на хості використовуйте команду /system ssh-exec user= user_ESXi address= IP_addr_ESXi command=	
Знайдено посилань: 0,01%	id: 69
"Command_ESXi"	
; Встановлено, що при віддаленому підключенні з роутера MikroTik з RouterOS 6.49 неможливо автоматично (у скрипті) підключитися до віддаленого хосту (ESXi) за допомогою пароля. Для автоматичного, програмованого підключення треба використовувати тільки сертифікати. Таким чином, потрібно додатково налаштувати службу SSH на кожному хосту у файлі /etc/ssh/sshd_config: дозволити підключатися користувачу root – параметр PermitRootLogin yes заборонити підключатися з використанням паролів – встановити параметр PasswordAuthentication no встановити відповідний сертифікат для користувача root на кожен хост ESXi у каталозі /etc/ssh/keys-root/ , де створити, або відредагувати файл authorized_keys, у якому кожен окремий рядок – це окремий сертифікат. Слід відзначити, що сертифікати ESXi та MikroTik з ОС RouterOS 6.49 не сумісні між собою. ESXi використовує більш сучасні сертифікати типу OpenSSH (RFC 4716) [17], а MikroTik з ОС RouterOS 6.49 – сертифікати RSA PEM. Одним зі шляхів вирішення цього питання є відключення (у файлі /etc/ssh/sshd_config) параметра FipsMode no. Однак це збільшує ризики безпеки. В цілому питання сумісності сертифікатів вирішено у RouterOS v7.12. Однак за певних обставин існує можливість згенерувати сумісний сертифікат і для RouterOS v6.49. Розглянемо тільки основні етапи: Згенерувати сертифікат у будь-якій ОС. Наприклад у ОС Windows ssh-keygen -t rsa -b 3072 -m PEM -C	
Знайдено посилань: 0,01%	id: 70
"adminifm11"	
-f C:\esx Встановити на MikroTik підтримку сучасних алгоритмів шифрування за допомогою команди у терміналі: /ip ssh set strong-crypto=yes host-key-size=3072 Скопіювати файли сертифікатів та роутер MikroTik Імпортувати сертифікати користувачу (admin) MikroTik, від імені якого будуть запускатися скрипти за допомогою команди: /user ssh-keys private	

`import user=admin private-key-file=file public-key-file=file.pub` Для організації поетапного завершення певних віртуальних машин треба скористатися командою, наприклад, для зупинки віртуальної машини з ім'ям `ubun_webs_new` :`local targetIP 192.168.100.18; /system ssh-exec user=root address=$targetIP command=`

Знайдено посилань: **0,13%**

id: 71

```
"vim-cmd vmsvc/power.shutdown \$(vim-cmd vmsvc/getallvms | grep -E '^[0-9]+' + ubun_webs_new' | awk '{print $1}')
```

; Для організації поетапного старту певних віртуальних машин треба скористатися командою, наприклад, для старту віртуальної машини з ім'ям `ubun_webs_new` :`/system ssh-exec user=root address=$targetIP command=`

Знайдено посилань: **0,14%**

id: 72

```
"vim-cmd vmsvc/power.on \$(vim-cmd vmsvc/getallvms | grep -E '^[0-9]+' + ubun_webs_new' | awk '{print $1}')
```

Для перевірки наявності віртуальної машини, шляхом `ping` скористайтесь наступним прикладом :`local targetIP 192.168.100.18; if ([/ping address=$targetIP count=3] 0) do={ } Для завершення всіх активних віртуальних машин скористався наступною командою /system ssh-exec user=root address=$targetIP command="V=$(vim-cmd vmsvc/getallvms); for n in $(esxcli vm process list | awk -F: ' /Display Name:/ {print $2}'); do i=$(echo "\$V" | grep -E \`

Знайдено посилань: **0,02%**

id: 73

```
"^[0-9]+ +\$V\"
```

| `awk '{print $1}'; [ -n "\$V" ] && vim-cmd vmsvc/power.shutdown \$i; done` Загальна логіка роботи цієї команди та головна складність під час керування `VM` через `CLI` в `ESXi` полягає в тому, що: 1. Інструмент `esxcli` бачить лише запущені наразі процеси `VM` і знає їхні зрозумілі `Display Name` (імена в інтерфейсі). 2. Інструмент керування живленням `vim-cmd` нічого не знає про імена процесів — йому для виконання команд (зокрема й вимкнення) потрібен внутрішній ідентифікатор машини — `VMID` (число). Цей скрипт пов'язує ці дві системи: він бере список тільки працюючих машин, знаходить для кожної з них її унікальний `VMID` і надсилає команду на безпечне вимкнення (`Guest Shutdown`). Для зручності розгорнемо рядок у читабельний `Bash`-скрипт: `V=$(vim-cmd vmsvc/getallvms) for n in $(esxcli vm process list | awk -F: ' /Display Name:/ {print $2}'); do i=$(echo`

Знайдено посилань: **0,05%**

id: 74

```
"$V" | grep -E "^[0-9]+ +\$n"
```

```
| awk '{print $1}'; [ -n
```

Знайдено посилань: **0,01%**

id: 75

```
"$i"
```

] && vim-cmd vmsvc/power.shutdown \$i Done

 Цей код починається з отримання глобального списку `VM` `V=$(vim-cmd vmsvc/getallvms)`; Команда `vim-cmd vmsvc/getallvms` виводить список взагалі всіх зареєстрованих на хості віртуальних машин. Перша колонка цього виводу — це якраз потрібний нам `VMID`. Скрипт один раз виконує цю важку команду і зберігає весь текст у змінну `V`, щоб не опитувати хост на кожній ітерації циклу. Потім – цикл за іменами тільки запущених `VM` `for n in $(esxcli vm process list | awk -F: ' /Display Name:/ {print $2}'); do ... done esxcli vm process list` — повертає конфігурацію тільки тих `VM`, які активні (виділяють ресурси). Якщо машина вимкнена, тут її не буде. `awk -F: ' /Display Name:/ {print $2}'` — фільтрує вивід. Ключ `-F: ' '` розділяє рядки за двокрапкою з пробілом. Скрипт шукає рядок `Display Name:` і забирає другу частину (`$2`) — тобто чисте ім'я віртуальної машини. Цикл `for n in ...` починає по черзі обробляти кожне знайдене ім'я, записуючи його в змінну `n`. Потім Співставлення Імені та `VMID` `i=$(echo`

Знайдено посилань: **0,05%**

id: 76

```
"$V" | grep -E "^[0-9]+ +\$n"
```

| `awk '{print $1}';` Для поточного імені машини (`$n`) скрипт шукає її `VMID` у збереженому тексті змінної `V`: `grep -E`

Знайдено посилань: **0,02%**

id: 77

```
"^[0-9]+ +\$i"
```

— регулярний вираз. Він шукає рядок, який починається з цифр (`^[0-9]+`), за якими йде один або кілька пробілів (`+`), а потім йде ім'я нашої машини (`$n`). Це захищає від хибних спрацьовувань (наприклад, якщо ім'я однієї машини частково дублює ім'я іншої). `awk '{print $1}'` — із знайденого рядка забирає перше слово (першу колонку), яким і є числовий `VMID`. Значення пишеться в змінну `i`. Потім – перевірка та відправка команди на вимкнення [`-n`

Знайдено посилань: **0,01%**

id: 78

```
"$i"
```

] && vim-cmd vmsvc/power.shutdown \$i; [-n

” Знайдено посилань: 0,01%	id: 79
"\$ "	
] — перевіряє, чи не порожня змінна i (чи знайшовся VMID). && — оператор	
” Знайдено посилань: 0,01%	id: 80
« ».	
Якщо перевірка успішна (VMID знайдено), виконується наступна команда. vim-cmd vmsvc/power.shutdown \$i — надсилає сигнал всередину ОС віртуальної машини (через VMware Tools) на коректне завершення роботи (ACPI Shutdown). Це найбезпечніший спосіб (не	
” Знайдено посилань: 0,03%	id: 81
«висмикування шнура з розетки»	
). Для остаточної зупинки певного хоста ESXi скористаємось командой /system ssh-exec user=root address=\$targetIP command=	
” Знайдено посилань: 0,01%	id: 82
"poweroff"	

Висновки до розділу У другому розділі кваліфікаційної роботи бакалавра було проведено детальний аналіз, розробку та проектування програмно-апаратної архітектури нової системи керування електроживленням для серверної інфраструктури невеликого підрозділу на базі гіпервізорів **VMware ESXi v7**. На основі проведених досліджень та інженерно-технічних рішень було отримано такі основні результати: Розроблено та проаналізовано діаграму станів процесів старту та завершення серверів, яка враховує специфіку поведінки віртуалізованого середовища в умовах раптового зникнення та подальшого відновлення основного електропостачання. Побудована модель станів дозволила формалізувати логіку роботи системи керування, чітко визначити тригери переходу між нормальним режимом, автономним живленням від акумуляторних батарей ДБЖ, процесом каскадного вимкнення та безпечного поетапного запуску. Це дало змогу мінімізувати ризики виникнення аварійних ситуацій і пошкодження даних при критичному зниженні ємності акумуляторів. Проведено глибокий аналіз системних налаштувань серверного обладнання (зокрема, платформ **Dell PowerEdge** з контролерами **iDRAC** та **Supermicro** з інтерфейсами **IPMI/BMC**). Запропоновано використати технологію **Wake-on-LAN (WoL)**. Запропоновано вибір мережевого обладнання під керуванням **RouterOS (MikroTik)** як центрального інтелектуального контролера системи керування електроживленням. Завдяки високій апаратній надійності, низькому власному енергоспоживанню, вбудованим інструментам моніторингу мережі (**Netwatch**), потужній системі скриптинг, планувальнику завдань (**Scheduler**) та інтегрованому клієнту **SSH**, маршрутизатор **MikroTik** здатний виконувати роль автономного

безпеки живлення без необхідності розгортання окремих енергоємних та вразливих серверів моніторингу. Розроблено алгоритм поетапного завершення роботи інфраструктури, ключовою особливістю якого є послідовне та безпечне зупинення сервісів. Алгоритм забезпечує першочергову коректну зупинку гостей віртуальних машин (за допомогою вбудованих засобів автоматизації та інтеграції з **VMware Tools**), що гарантує збереження цілісності баз даних, транзакційних файлових систем та метаданих на рівні сховищ даних (**SAN/NAS**). Лише після успішного вимкнення всіх ВМ алгоритм ініціює вимкнення самих хостів-гіпервізорів **ESXi** та перехід системи живлення в режим очікування або глибокого сну. Реалізовано надійний механізм безпечної взаємодії між керуючим пристроєм та серверами за допомогою протоколу **SSH**. Описано та впроваджено процес генерації та імпорту відкритих криптографічних ключів (**RSA/ECDSA**), що повністю виключає необхідність збереження паролів адміністратора у відкритому вигляді всередині скриптів на **MikroTik**, суттєво підвищуючи рівень інформаційної безпеки системи. Розроблено та протестовано практичні програмні компоненти автоматизації на стороні **MikroTik** та **ESXi CLI**. Зокрема, створено скрипт обробки списку віртуальних машин, який за допомогою системної утиліти **vim-cmd**, регулярних виразів **grep** та потокового фільтра **awk** динамічно визначає ідентифікатори **VMID** активних віртуальних машин та надсилає їм команду плавного вимкнення **power.shutdown**. Це дозволило повністю автоматизувати рутинні операції адміністрування та забезпечити високу швидкість реакції системи на аварійні події. Таким чином, розроблені у другому розділі рішення формують повний теоретичний та інженерний базис для створення високонадійної системи керування живленням, яка впроваджує інтелектуальну координацію між джерелами безперебійного живлення, активним мережевим обладнанням та віртуалізованими обчислювальними ресурсами підприємства. Загальні висновки Змн. Арк. № докум. Підпис Дата Арк. З ІТС.4КІ.0624.03-ПЗ Розроб. Семесенко І. Керівник Могильний Г.А. В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. Загальні висновки Літ. Акрушів З ЛНУ Кафедра ІТС, Гр.4КІ . У кваліфікаційній роботі бакалавра розв'язано актуальне науково-практичне завдання щодо підвищення

живучості та надійності функціонування серверної інфраструктури навчальних комп'ютерних лабораторій закладів вищої освіти (ЗВО) в умовах нестабільного енергопостачання, спричиненого воєнним станом в Україні. Мети дослідження було досягнуто шляхом проектування, розробки та впровадження гнучкої, економічно ефективної автоматизованої системи дистанційного керування живленням на базі наявної мережевої інфраструктури [MikroTik](#) та портативних джерел безперебійного живлення. За результатами виконаних досліджень та розробок сформульовано такі основні висновки: Проаналізовано сучасні технологічні рішення у сфері забезпечення безперервного та якісного електропостачання ІТ-інфраструктури відповідно до вимог міжнародних стандартів (зокрема, [ANSI/TIA-942](#)). Виявлено, що промислові системи зберігання енергії ([ESS](#)) на базі літій-залізо-фосфатних акумуляторів ([LiFePO₄](#)) та модульні онлайн-ДБЖ з подвійним перетворенням мають надвисоку ринкову вартість. Це робить їх впровадження економічно недоцільним або неможливим для невеликих підрозділів та кафедр ЗВО в умовах обмеженого бюджетного фінансування під час воєнного стану. Обґрунтовано перенесення інтелектуальних функцій керування на рівень наявної мережевої інфраструктури. Маршрутизатори компанії [MikroTik](#) під керуванням операційної системи [RouterOS](#) обрано як центральні контролери системи завдяки їхній низькій вартості, високій апаратній надійності, мінімальному власному енергоспоживанню та потужним вбудованим інструментам автоматизації (моніторинг [Netwatch](#), планувальник завдань

[Scheduler](#), мова скриптів та клієнт [SSH](#)). Це дозволило уникнути закупівлі дорогого спеціалізованого програмного забезпечення класу [EMS \(Energy Management Systems\)](#) та додаткових серверів моніторингу. Побудовано формалізовану діаграму станів процесів старту та завершення серверів. Розроблена модель станів чітко визначає тригери переходу між нормальним режимом, роботою від акумуляторів, каскадним вимкненням другорядних сервісів та безпечним поетапним відновленням роботи інфраструктури. Це дозволило врахувати специфіку поведінки складних віртуалізованих середовищ у разі раптового зникнення та подальшого відновлення мережі 220В, мінімізуючи ризик пошкодження метаданих файлової системи [VMFS](#) та віртуальних дисків на платформі [VMware vSphere](#). Розроблено та впроваджено алгоритм поетапного завершення роботи інфраструктури на базі гіпервізорів [VMware ESXi v7](#). Ключовою особливістю алгоритму є пріоритизація та каскадне вимкнення гостьових віртуальних машин за допомогою інструментів інтеграції [VMware Tools](#). Завдяки цьому забезпечується збереження транзакційної цілісності баз даних та файлових систем на сховищах [SAN/NAS](#) перед вимкненням безпосередньо фізичних хостів-гіпервізорів. Для зворотного процесу ініціалізації серверного обладнання запропоновано та налаштовано технологію віддаленого ввімкнення [Wake-on-LAN \(WoL\)](#) через контролери [iDRAC \(Dell\)](#) та [IPMI/BMC \(Supermicro\)](#). Реалізовано надійні програмно-апаратні механізми автоматизації та інформаційної безпеки. Створено практичні скрипти обробки списку віртуальних машин на стороні [RouterOS](#), які за допомогою [SSH](#)-запитів до [CLI ESXi](#), системної утиліти [vim-cmd](#), а також потокових фільтрів [grep](#) та [awk](#) динамічно виявляють ідентифікатори активних віртуальних машин ([VMID](#)) та надсилають команду [power.shutdown](#). Безпека взаємодії між роутером [MikroTik](#) та серверами забезпечується за допомогою безпарольного доступу на основі імпорту відкритих криптографічних ключів ([RSA/ECDSA](#)), що усуває потребу зберігання конфіденційних даних у відкритому тексті скриптів. Практична значущість отриманих результатів полягає у створенні готового до розгортання бюджетного рішення, яке здатне інтегрувати доступні портативні зарядні станції та побутові ДБЖ у загальний контур керування віртуалізованою ІТ-інфраструктурою підрозділу. Впровадження розробленої системи дозволяє оптимізувати енергоспоживання, подовжити час автономної роботи критично важливих сервісів за рахунок консервації енергії батарей та повністю усунути аварійні вимкнення серверів, гарантуючи високу живучість інфраструктури комп'ютерних класів у кризових умовах сьогодення. Список літератури Змн. Арк. № докум. Підпис Дата Арк. 3 ІТС.4КІ.0624.03-ПЗ Розроб. Семесенко І. Керівник Могильний Г.А. В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. Список літератури Літ. Акрушів 3 ЛНУ Кафедра ІТС, Гр.4КІ. Могильний Г.А., Аналіз програмно-апаратних засобів створення системи з віддаленим доступом до навчальних комп'ютерних лабораторій закладів середньої освіти. № 1 (277) (2023):

 Знайдено плагиату: **0,06%** <https://web-crawler.plagiarism-detect...> id: 84

Вісник Східноукраїнського національного університету імені Володимира Даля

URL: <https://doi.org/10.33216/1998-7927-2019-256-8-5-19> (дата звернення: 10.04.2026).

Могильний Г.А., Семенов М.А., Кіреєв В.Ю. Впровадження системи віддаленого доступу до інформаційних ресурсів комп'ютерних лабораторій. № 2 (272) (2022):

 Знайдено плагиату: **0,06%** <https://web-crawler.plagiarism-detect...> id: 85

Вісник Східноукраїнського національного університету імені Володимира Даля

URL: <https://doi.org/10.33216/1998-7927-2022-272-2-7-14> (дата звернення: 10.04.2026).

[Computer Networks: A Systems Approach](#) / Брюс С. Деві, Ларрі Л. Петерсон. [Elsevier](#), 2021. 848 с. Комп'ютерні мережі / Коробейнікова Т.І., Захарченко С.М. Львівська політехніка, 2022.

228 с. Комп'ютерні мережі. Книга 1. Технології комп'ютерних мереж / Євсєєв С.П., Дженюк Н.В. Новий світ-2000, 2024. 471 с. [Systems Programming: Designing and Developing Distributed Applications](#) / Річард Ентоні. Elsevier, 2015. 548 с. Про WTHD [

Знайдено плагіату: **0,71%** <https://web-crawler.plagiarism-detect...> + 6 ресурсів! id: **86**

Електронний ресурс] – Режим доступу: <https://www.wthdne.com/uk/> (дата звернення: 05.05.2026). Система живлення [Електронний ресурс] – Режим доступу: <https://www.wthdne.com/uk/power-supply-system> (дата звернення: 05.05.2026). Система накопичення енергії [Електронний ресурс] – Режим доступу: <https://www.wthdne.com/uk/energy-storage-system> (дата звернення: 05.05.2026). Джерело живлення [ups](#) [Електронний ресурс] – Режим доступу: <https://www.wthdne.com/uk/ups-power-supply> (дата звернення: 05.05.2026). [Ans/tia-942 standard](#) [Електронний ресурс] – Режим доступу: <https://karnodatacenter.com/files/TIA-942-B-2017%20Rev%20Full.pdf> (дата звернення: 05.05.2026). [INTERNATIONAL STANDARD IEC 62040-](#) [Електронний ресурс] – Режим доступу:

<https://www.scribd.com/document/847436937/IEC-62040-3-2021> (дата звернення: 05.05.2026). Г.А. Могильний, М.А. Семенов, В.Ю. Донченко, І.М. Швець, С.М. Донченко [Автоматизація процедур каскадного керування живленням серверів VMWARE VSPHERE засобами мережевої інфраструктури. № 3 \(301\) \(2026\):](#)

Знайдено плагіату: **0,06%** <https://web-crawler.plagiarism-detect...> id: **87**

Вісник Східноукраїнського національного університету імені Володимира Даля

URL: <https://doi.org/10.33216/1998-7927-2026-301-3-14-24> (дата звернення: 10.04.2026).

RouterOS Documentation [Електронний ресурс] URL:

<https://help.mikrotik.com/docs/spaces/ROS/pages/328059/RouterOS> (дата звернення:

08.01.2026). [vmware-vsphere-7-0](#) [Електронний ресурс] URL:

<https://techdocs.broadcom.com/content/dam/broadcom/techdocs/us/en/pdf/vmware/vsphere/vsphere/vmware-v-7-0.pdf> (дата звернення: 13.11.2026). [Horizon 8 Documentation: Installation and Configuration Guide](#). [Електронний ресурс] URL:

<https://www.google.com/search?q=https://docs.omnissa.com/bundle/horizon-installation/> (дата звернення: 12.11.2026). [RFC 4716. The Secure Shell \(SSH\) Public Key File Format](#) [Електронний ресурс] URL: <https://datatracker.ietf.org/doc/html/rfc4716> (дата звернення: 08.01.2026).

Додатки Змн. Арк. № докум. Підпис Дата Арк. 3 ІТС.4КІ.0624.03-ПЗ Розроб. Тітов Е.І. Керівник Могильний Г.А. В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. Додатки Літ. Акрушів З ЛНУ Кафедра ІТС, Гр.4КІ . Додаток А Текст головного скрипту для старту :[log info](#) (

Знайдено посилань: **0,05%** id: **88**

"[NetwatchStart: RUN start script - StartSum](#) !" "

); :if ([[:len [/ip firewall address-list find where list=

Знайдено посилань: **0,01%** id: **89**

"[powerUPrun](#)"

]] = 0) do={ :log info

Знайдено посилань: **0,07%** id: **90**

"[NetwatchStat: Address list powerUPrun is absent \(empty\). OK!!](#)"

; /ip firewall address-list add list=

Знайдено посилань: **0,01%** id: **91**

"[powerUPrun](#)"

address=127.

Знайдено плагіату: **0,03%** <https://web-crawler.plagiarism-detect...> id: **92**

0.0.1;

#####1

:local counter 0; :local maxRepeats 100; :while ([[:len [/ip firewall address-list find where list=

Знайдено посилань: **0,01%** id: **93**

"[powerDOWNrun](#)"

]] 0) && (\$counter \$maxRepeats)) do={ :set counter (\$counter + 1); :log info (

Знайдено посилань: **0,08%** id: **94**

"[NetwatchStart: Waiting for adres-list - powerDOWNrun to clear... :](#) "

. \$counter); :delay 10s; } :if (\$counter = \$maxRepeats) do={ :log warning "[NetwatchStart: Wait timeout! ERROR addr-list - powerDOWNrun still exists after 100 attempts STOP process powerUP.](#)"; :error "[NetwatchStart: Stopping script to timeout addr-list - powerDOWNrun NO delete.](#)"

Знайдено посилань: **0,03%** id: **95**

```

"; } else={ :log info "
NetwatchStart: NO addr-list powerDOWNrun , proceeding with the script.
"» Знайдено посилань: 0,35% id: 96
"; } #####
:local ips {192.168.100.22;192.168.100.12;127.0.0.1;192.168.100.22;192.168.100.12}; :local
delays {2 ;40 ;260 ;100 ;10}; :local delaysCommon 300 ;
##### :local st [/tool
netwatch get [find host="
192.168.102.4
"» Знайдено посилань: 0,04% id: 97
"] status]:if ($st = "
up") do={ :log info "NetwatchStart: Status netwath is UP continue proceeding Start. in script
STARTSUM."
##### :for
i from=0 to=([:len $ips] - 1) do={ :local currentIP ($ips - $i); :local currentDelay ($delays - $i);
:delay 1s; :if ([/ip firewall address-list find list=
"» Знайдено посилань: 0,01% id: 98
"powerUP"
address=$currentIP]) = 0) do={ /ip firewall address-list add list=
"» Знайдено посилань: 0,01% id: 99
"powerUP"
address=$currentIP } :delay 1s; :log info (
"» Знайдено посилань: 0,07% id: 100
"NetwatchStart: Waiting $currentDelay for $currentIP... for list $i !"
); :if ([/ip firewall address-list find where list=
"» Знайдено посилань: 0,01% id: 101
"powerDOWNrun"
]) = 0) do={ :delay $currentDelay; } else={ :log info (
"» Знайдено посилань: 0,07% id: 102
"NetwatchStart: !!!!!!!!!!!!! Check event down server STOP starting process !"
); :set i value=[:len $ips]; } ###:local scr value=(
"» Знайдено посилань: 0,03% id: 103
"Start" . [[:tostr ($i+1)]
); :local scr value=(
"» Знайдено посилань: 0,01% id: 104
"Start"
. ($i+1)); :delay 1s; :if ([/ip firewall address-list find where list=
"» Знайдено посилань: 0,01% id: 105
"powerDOWNrun"
]) = 0) do={ :delay 1s; :log info (
"» Знайдено посилань: 0,03% id: 106
"NetwatchStart: Call scrName $scr !"
); ##:execute script=$scr; /system script run ($scr); :delay 3s; /ip firewall address-list remove
[find where list=
"» Знайдено посилань: 0,01% id: 107
"powerUP"
]; } else={ :log info (
"» Знайдено посилань: 0,07% id: 108
"NetwatchStart: !!!!!!!!!!!!! Check event down server STOP starting process !"
); :set i value=[:len $ips]; /ip firewall address-list remove [find where list=
"» Знайдено посилань: 0,01% id: 109
"powerUP"
]; } ##### end for } :delay $delaysCommon;
##### end if check status netwach check }
##### /ip firewall address-list remove
[find where list=
"»

```


Знайдено посилань: 0,01%	id: 110
"powerUP"	
]; /ip firewall address-list remove [find where list=	
Знайдено посилань: 0,01%	id: 111
"powerUPrun"	
]; } else={ :log info	
Знайдено посилань: 0,08%	id: 112
"NetwatchStart: Address list 'powerUPrun' is exist. ignore 2 start ERROR!!!!"	
; } :log info (	
Знайдено посилань: 0,04%	id: 113
"NetwatchStart: END script - StartSum !"	
; Додаток Б Текст головного скрипту для завершення log info (	
Знайдено посилань: 0,05%	id: 114
"NetwatchDown: RUN start script - DownSum !"	
; # check for double-start (no list powerDOWNrun) :if ([[:len [/ip firewall address-list find where list=	
Знайдено посилань: 0,01%	id: 115
"powerDOWNrun"	
] = 0) do={ :log info	
Знайдено посилань: 0,07%	id: 116
"NetwatchDown: Address list 'powerDOWNrun is absent (empty). OK!!"	
; /ip firewall address-list add list=	
Знайдено посилань: 0,01%	id: 117
"powerDOWNrun"	
address=127.0.0.1; # ch scet :local svetch 8; #kolichecsnvo vidkluchen :local kolv 4; :local del1 (((\$svetch*12)/(\$kolv +1)); ;if (\$del1 85) do={ :set del1 85; } :set del1 ((\$del1*3600)/50); :local del2 (\$del1/60); #:put \$del1; #:put \$del2; :local del3 0;	
#####	
#####	
:while (\$del3 \$del1) do={ :local st [/tool netwatch get [find host=	
Знайдено посилань: 0,03%	id: 118
"192.168.102.4"	
] status] :set del3 (\$del3+\$del2); #:put \$del3; :delay \$del2; :if (\$st =	
Знайдено посилань: 0,01%	id: 119
"up"	
) do={ :log info	
Знайдено посилань: 0,1%	id: 120
"NetwatchDown: Status netwath is UP NO continue proceeding DOWN. in script DownSum."	
; /ip firewall address-list remove [find where list="powerDOWNrun"]; :error " NetwatchDown: Stopping script - change powerUP .	
Знайдено посилань: 0,04%	id: 121
"; } else= { :log info "	
NetwatchDown: First timeout \$del3. in script DownSum. sum time \$del1	
Знайдено посилань: 0,17%	id: 122
"; } }#. # :local counter 0; :local maxRepeats 100; # 100 :while ([[len [/ip firewall address-list find where list="	
powerUPrun"]]) && (\$counter \$maxRepeats)) do={ # :set counter (\$counter + 1); #: :log info ("NetwatchDown: Waiting for adres-list - powerUPrun to clear... : " . \$counter); #: :delay 10s; } #	
:if (\$counter = \$maxRepeats) do={ :log warning "NetwatchDown: Wait timeout! ERROR addr-list - powerUP still exists after 100 attempts STOP process powerDown." # :error " NetwatchDown: Stopping script to timeout addr-list - powerUP NO delete."; } else={ :log info "NetwatchDown: NO addr-list powerUPrun , proceeding OK." } :local ips	
{192.168.100.12;192.168.100.22;192.168.100.127;192.168.100.12;192.168.100.22;192.168.100.12;192.168.100.12;192.168.100.12}	
:local delays {2s;60s;10s;100s;50s;240s;10s}; :local delaysCommon 240s; :for i from 0 to=	
([[:len \$ips] - 1) do={ :local currentIP (\$ips- \$i); :local currentDelay (\$delays- \$i); # 1. ADD powerDOWN :if ([[:len [/ip firewall address-list find list=	
Знайдено посилань: 0,01%	id: 123

```

"powerDOWN"
address=$currentIP]] = 0) do={ /ip firewall address-list add list=
Знайдено посилань: 0,01% id: 124
"powerDOWN"
address=$currentIP } :delay 1s; # 2.) :log info (
Знайдено посилань: 0,07% id: 125
"NetwatchDown: Waiting $currentDelay for $currentIP... for list $ !"
); :delay $currentDelay; # 3. Run enter script start+iterac :local scr value=(
Знайдено посилань: 0,01% id: 126
"Down"
. ($i+1)); :log info (
Знайдено посилань: 0,03% id: 127
"NetwatchDown: Call scitpName $scr !"
); :delay 1s; ##:execute script=$scr; /system script run ($scr); :delay 3s; /ip firewall address-list
remove [find where list=
Знайдено посилань: 0,01% id: 128
"powerDOWN"
] } # pause for end downing procesing :delay $delaysCommon; /ip firewall address-list remove
[find where list=
Знайдено посилань: 0,01% id: 129
"powerDOWN"
]; /ip firewall address-list remove [find where list=
Знайдено посилань: 0,01% id: 130
"powerDOWNrun"
]; } else={ :log info
Знайдено посилань: 0,08% id: 131
"NetwatchDown:Address list 'powerDOWNrun' is exist. ignore 2 start ERROR!!!"
; } :log info
Знайдено посилань: 0,04% id: 132
"NetwatchDown: END script - DownSum !"
; Додаток В Текст першого допоміжного скрипту для старту :log info
Знайдено посилань: 0,03% id: 133
"NetwatchStart: start script Start1"
; :local targetIP [/ip firewall address-list get [find list=
Знайдено посилань: 0,01% id: 134
"powerUP"
] address] ; # 192.168.100.22 :if ([ :len [/ip firewall address-list find where list=
Знайдено посилань: 0,01% id: 135
"powerDOWNrun"
]] = 0) do={ ##### /tool wol mac=3C:EC:EF:7A:08:2E
interface=bridge; :log info
Знайдено посилань: 0,03% id: 136
"NetwatchStart: WOL Start1"
; } else={ :log info (
Знайдено посилань: 0,08% id: 137
"NetwatchStart: !!!!! Check event down server STOP end script Start1 !"
); } :log info
Знайдено посилань: 0,03% id: 138
"NetwatchStart: end script Start1"
; Додаток Д Текст першого допоміжного скрипту для завершення :local targetIP [/ip firewall
address-list get [find list=
Знайдено посилань: 0,01% id: 139
"powerDOWN"
] address] ; :log info (

```

"Знайдено посилань: 0,05% "NetwatchDown: start script Down1 for addr=".	id: 140
\$targetIP); if ([/ping address=\$targetIP count=3] 0) do={ #---192.168.100.12-----; /system ssh-exec user=root address=\$targetIP command=	
"Знайдено посилань: 0,13% "vim-cmd vmtoolsd/power.shutdown \\$(vim-cmd vmtoolsd/getallvms grep -E '[0-9]+' +conn1 awk '{print \$1}')"	id: 141
;:log info	
"Знайдено посилань: 0,04% "NetwatchDown: VM conn1 script Down1"	id: 142
;:delay 10s; /system ssh-exec user=root address=\$targetIP command=	
"Знайдено посилань: 0,13% "vim-cmd vmtoolsd/power.shutdown \\$(vim-cmd vmtoolsd/getallvms grep -E '[0-9]+' +ubun_webs_new awk '{print \$1}')"	id: 143
;:log info	
"Знайдено посилань: 0,04% "NetwatchDown: VM ubun_webs_new script Down1"	id: 144
;/system ssh-exec user=root address=\$targetIP command=	
"Знайдено посилань: 0,13% "vim-cmd vmtoolsd/power.shutdown \\$(vim-cmd vmtoolsd/getallvms grep -E '[0-9]+' +newuga awk '{print \$1}')"	id: 145
;:log info	
"Знайдено посилань: 0,04% "NetwatchDown: VM newuga script Down1"	id: 146
;:delay 10s; /system ssh-exec user=root address=\$targetIP command=	
"Знайдено посилань: 0,13% "vim-cmd vmtoolsd/power.shutdown \\$(vim-cmd vmtoolsd/getallvms grep -E '[0-9]+' +new-xcent1 awk '{print \$1}')"	id: 147
;:log info	
"Знайдено посилань: 0,04% "NetwatchDown: VM new-xcent1 script Down1"	id: 148
;:delay 10s; /system ssh-exec user=root address=\$targetIP command=	
"Знайдено посилань: 0,13% "vim-cmd vmtoolsd/power.shutdown \\$(vim-cmd vmtoolsd/getallvms grep -E '[0-9]+' +servad_vm_old awk '{print \$1}')"	id: 149
;:log info	
"Знайдено посилань: 0,04% "NetwatchDown: VM servad_vm_old script Down1"	id: 150
; } else={ :log info	
"Знайдено посилань: 0,08% "ESXi host \$targetIP is unreachable. SSH 1 command aborted."	id: 151
} #/system ssh address=\$targetIP user=root command=	
"Знайдено посилань: 0,01% "	id: 152
output-to-file=del; :log info (	
"Знайдено посилань: 0,06% "NetwatchDown: end script Down1 for ip addr=".	id: 153
\$targetIP); Додаток Е Текст інших допоміжних скриптів для старту log info	
"Знайдено посилань: 0,03% "NetwatchStart: start script Start2"	id: 154
;:local targetIP [/ip firewall address-list get [find list=	
"Знайдено посилань: 0,01% "powerUP"	id: 155
] address]; #192.168.100.12 if ([!len [/ip firewall address-list find where list=	
"Знайдено посилань: 0,01%	id: 156

```

"powerDOWNrun"
]] = 0) do={ ##### /tool wol mac=20:04:0F:EE:3D:B4 interface=bridge;
:log info
"NetwatchStart: WOL Start2"
; } else={ :log info (
"NetwatchStart: Check event down server STOP end script Start2 !"
); } :log info
"NetwatchStart: end script Start2"
;
#####
# Типова структура – шаблон для інших
#####
:log info
"NetwatchStart: start script Start0"
; :local targetIP [/ip firewall address-list get [find list=
"powerUP"] address]
; :if ([len [/ip firewall address-list find where list=
"powerDOWNrun"
]] = 0) do={ :local k 0; :log info
"NetwatchStart: end script Start0"
; } else={ :log info (
"NetwatchStart: Check event down server STOP end script Start0 !"
); } :log info (
"NetwatchStart: end script Start0 !"
); Додаток Є Текст інших допоміжних скриптів для завершення :local targetIP [/ip firewall
address-list get [find list=
"powerDOWN"
] address]; :log info (
"NetwatchDown: start script Down2 for IP add=".
$targetIP); #192.168.100.22 :if ([/ping address=$targetIP count=3] 0) do={ /system ssh-exec
user=root address=$targetIP command=
"vim -cmd vmsvc/power.shutdown \$(vim -cmd vmsvc/getallvms | grep -E '[0-9]+' +servRDP1 |
awk '{print \$1}')"
:log info
"NetwatchDown: VM servRDP1 script Down2"
; /system ssh-exec user=root address=$targetIP command=
"vim -cmd vmsvc/power.shutdown \$(vim -cmd vmsvc/getallvms | grep -E '[0-9]+' +new_serv2"
| awk '{print \$1}')"
:log info
"NetwatchDown: VM new_serv2 script Down2"

```

```

; /system ssh-exec user=root address=$targetIP command=
Знайдено посилань: 0,13% id: 172
"vim-cmd vmsvc/power.shutdown \$(vim-cmd vmvc/getallvms | grep -E '[0-9]+' +servad1 |
awk '{print $1}')"
:log info
Знайдено посилань: 0,04% id: 173
"NetwatchDown: VM servad1 script Down2"
; } else={ :log info
Знайдено посилань: 0,07% id: 174
"ESXi host $targetIP is unreachable. SSH2 command aborted."
} #/system ssh-exec user=root address=192.168.100.22 command=
Знайдено посилань: 0,01% id: 175
"poweroff"
; :log info (
Знайдено посилань: 0,06% id: 176
"NetwatchDown: end script Down2 for ip addr=".
$targetIP);
#####
# Типова структура - шаблон для інших
#####
:local targetIP [/ip firewall address-list get [find list=
Знайдено посилань: 0,01% id: 177
"powerDOWN"
] address]; :log info (
Знайдено посилань: 0,05% id: 178
"NetwatchDown: start script Down0 for addr=".
$targetIP); :if ([/ping address=$targetIP count=3] 0) do={ :log info (
Знайдено посилань: 0,03% id: 179
"NetwatchDown: YES addr=".
$targetIP); #---192.168.100.12-----; } else={ :log info
Знайдено посилань: 0,08% id: 180
"ESXi host $targetIP is unreachable. SSH 0 command aborted."
} #/system ssh address=$targetIP user=root command=
Знайдено посилань: 0,01% id: 181
"ls"
output-to-file=del; :log info (
Знайдено посилань: 0,06% id: 182
"NetwatchDown: end script Down0 for ip addr=".
$targetIP);

```

Відмова від відповідальності:

Цей звіт повинен бути правильно інтерпретований та проаналізований кваліфікованою особою, яка несе відповідальність за оцінку!

Будь-яка інформація, наведена у цьому звіті, не є остаточною та підлягає ручному огляду та аналізу. Дотримуйтесь вказівок: [Рекомендації з оцінки](#)

Детектор Плагіату - Право знати автора! ☐ SkyLine LLC