DOI: https://doi.org/10.36910/6775-2524-0560-2025-59-04
УДК: 004.415:004.6
Андрущак Ігор Євгенович, д.т.н., професор
https://orcid.org/0000-0002-8751-4420
Шмаровоз Станіслав Васильович, магістр
Луцький національний технічний університет, м. Луцьк, Україна

ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ МОНІТОРИНГОВИХ СИСТЕМ У КОРПОРАТИВНИХ МЕРЕЖАХ: ТИПОВІ ПРОБЛЕМИ І РІШЕННЯ

Андрущак І. Є., Шмаровоз С. В. Забезпечення надійності моніторингових систем у корпоративних мережах: типові проблеми і рішення. У статті розглядаються актуальні аспекти забезпечення надійності систем моніторингу у корпоративних комп'ютерних мережах. З урахуванням зростаючої складності IT-інфраструктур, надійність моніторингу набуває критичного значення для підтримання безперервності сервісів, своєчасного реагування на інциденти та запобігання фінансовим втратам. Проведено детальний аналіз типових проблем, таких як єдина точка відмови (SPOF), перевантаження системи, хибні спрацювання тригерів, втрати даних та ненадійні канали оповіщення. Наведено статистику з авторитетних джерел і приклади реальних інцидентів, що ілюструють наслідки недосконалого моніторингу в умовах зростаючого навантаження та складності. Окрема увага приділена технічним і архітектурним заходам підвищення стійкості: розподіленим системам, кластеризації серверів, реплікації баз даних, використанню багатоканального нотифікаційного механізму та автоматизації реакції на інциденти. Обґрунтовано доцільність впровадження інтелектуальної фільтрації подій, кореляції сигналів тривоги та регулярного тестування працездатності компонентів. Запропоновано комплексну архітектуру надійної моніторингової системи, що враховує AI-алгоритми, кешування, резервування та адаптивну маршрутизацію повідомлень. Визначено напрями подальших досліджень, зокрема динамічне налаштування порогів спрацювання за допомогою самонавчальних алгоритмів, вивчення впливу людського фактора та інтеграції з інструментами кібербезпеки. Представлені рішення дозволяють значно зменшити ризики критичних збоїв, оптимізувати реагування, підвищити ефективність моніторингу та створити передумови для стабільного розвитку інформаційних систем.
Ключові слова: моніторинг мережі, fault-tolerance, надійність, корпоративна мережа, оповіщення, високодоступність, безперервність сервісу, інцидент-менеджмент, автоматизація, машинне навчання, кіберстійкість, аналітика подій.

Andrushchak I., Shmarovoz S. Ensuring the Reliability of Monitoring Systems in Corporate Networks: Typical Problems and Solutions. The article explores current aspects of ensuring the reliability of monitoring systems in corporate computer networks. Given the growing complexity of IT infrastructures, monitoring reliability is critically important for maintaining service continuity, ensuring timely incident response, and preventing financial losses. A detailed analysis of common problems is provided, including single points of failure (SPOF), system overload, false trigger activations, data loss, and unreliable alerting channels. Statistical data from authoritative sources and real-world examples are cited to illustrate the consequences of inadequate monitoring under high load and complexity. Particular attention is given to technical and architectural measures for increasing system resilience: distributed architectures, server clustering, database replication, multichannel notification mechanisms, and automated incident response. The paper substantiates the implementation of intelligent event filtering, alarm correlation, and regular system health testing. A comprehensive architecture is proposed that incorporates AI algorithms, caching, failover, and adaptive message routing. Future research directions are outlined, including dynamic threshold adjustment using self-learning algorithms, studying the impact of human factors, and integration with cybersecurity tools. The proposed solutions significantly reduce the risk of critical failures, optimize response processes, increase monitoring efficiency, and establish a foundation for the stable development of corporate information systems.
Keywords: network monitoring, fault tolerance, reliability, corporate network, alerting, high availability, service continuity, incident management, automation, machine learning, cyber resilience, event analytics.

Постановка проблеми. У складних корпоративних мережах надійний моніторинг є критично важливим для запобігання збоїв і забезпечення безперервності роботи сервісів. Згідно з дослідженнями Gartner, понад 60% простоїв сервісів у корпоративних IT-системах були спричинені недоліками моніторингових рішень або їх несвоєчасною реакцією на інциденти. Центральні або слабко масштабовані системи часто стають вразливими до збоїв, що призводить до втрати контролю над інфраструктурою або несвоєчасного реагування. Проблема ускладнюється швидким зростанням кількості пристроїв, трафіку та взаємозв'язків у межах сучасних мереж.

Аналіз останніх досліджень і публікацій. У наукових працях та технічних звітах описано переваги використання відмовостійких рішень у побудові систем моніторингу (Zabbix, Nagios, Prometheus). У роботах [1–2] розглянуто підходи до забезпечення fault-tolerance на рівні сервера та мережевих агентів. У публікаціях Zabbix [4] описано типову архітектуру з реплікацією бази даних, горизонтальним масштабуванням та балансуванням навантаження. Проте більшість практичних досліджень зосереджені на налаштуванні окремих інструментів, тоді як питання комплексної надійності часто залишаються поза увагою.

Мета дослідження – виявити основні проблеми надійності моніторингових систем у корпоративних мережах і запропонувати рішення для їх усунення.

Виклад основного матеріалу.

1. Типові проблеми надійності моніторингових систем

1.1 Єдина точка відмови: Проблема Single Point of Failure (SPOF) виникає у випадках, коли відмова одного критичного компонента системи призводить до повної недоступності сервісу моніторингу. Нижче зображено приклад SPOF, де точкою відмови є роутер.

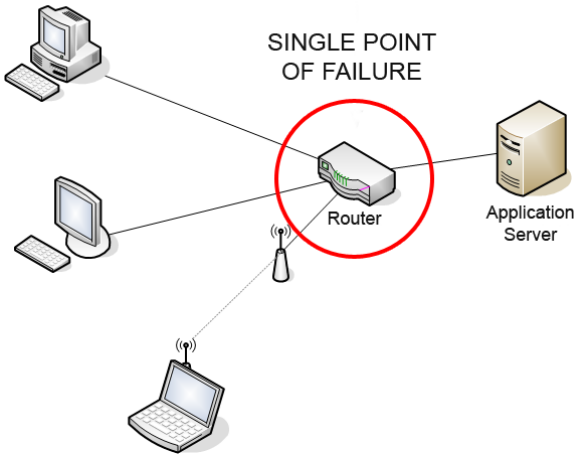

Рис. 1 – Приклад SPOF[5]

Найбільш поширеним SPOF є централізований сервер без резервування, який виконує функції збору, обробки та зберігання метрик. У разі його збоїв – втрачається не лише доступ до історичних даних, а й можливість вчасного виявлення нових інцидентів у мережі.

Типові прояви:

- зникнення всіх активних оповіщень;
- неповні дашборди та втрата оновлення даних у реальному часі;
- неможливість аналітичного аналізу ситуації після відновлення.

Згідно з аналітичним звітом Red Hat [8] за 2023 рік, 30% середніх і великих компаній зазнали принаймні одного критичного інциденту через відсутність відмовостійкості моніторингового компонента. В окремих випадках час простою сягав кількох годин, що завдавало значних фінансових і репутаційних збитків.

Причини появи SPOF:

- централізована база даних без реплікації або кластеризації;
- відсутність резервного сервера або агента;
- один канал доставки сповіщень (наприклад, лише email);
- відсутність балансування навантаження та автоматичного перемикання на резерв.

1.2 Надмірне навантаження: одна з найпоширеніших технічних проблем у системах моніторингу, особливо у великих корпоративних середовищах із великою кількістю вузлів. Перевантаження виникає тоді, коли обсяг запитів до системи перевищує її обчислювальні або мережеві можливості. Наприклад, у системі з 10 000 пристроїв і частотою опитування 30 секунд, упродовж хвилини генерується понад 300 000 запитів. Якщо система не оптимізована, це призводить до:

- значного уповільнення збору та обробки даних;
- затримок у спрацюванні тригерів та генерації сповіщень;
- пропуску критичних подій через перевантаження черги обробки;
- повного падіння сервісу у пікові періоди навантаження.

Надмірне навантаження часто є наслідком:

- відсутності горизонтального масштабування;
- недостатньої оптимізації запитів або агрегації даних;
- погано налаштованих метрик або надмірної кількості тригерів;

– використання повільної СУБД або перевантажених дисків для зберігання логів.

За оцінками Gartner, 37% відмов у системах моніторингу в середніх і великих компаніях у 2022–2023 роках були спричинені саме надмірним навантаженням на моніторингові сервери.

1.3 Хибні спрацювання: Найпоширеніша причина недовіри до системи – **завелика кількість некоректних оповіщень**. За даними Splunk [7], **до 40% всіх сповіщень** виявляються хибними

Причини:

– погано налаштовані порогові значення;
– невраховані шаблони «нічного» трафіку чи планових оновлень;
– події з короткочасними відхиленнями (сплески навантаження).

Наслідки**:**

– IT-фахівці ігнорують сповіщення («alert fatigue»);
– затримка в реагуванні на реальні інциденти;
– псевдоінциденти можуть запускати ланцюгову автоматизацію.

1.4 Втрати даних*:* Це критична проблема для систем, які повинні зберігати **повну історію подій і метрик**. Типові причини:

– відмова бази даних (MySQL/PostgreSQL) без реплікації;
– розрив з'єднання між агентом і сервером;
– заповнення диска логами або бекапами.

Типові наслідки:

– втрата аналітичної цілісності (неповні графіки);
– неможливість провести ретроспективне розслідування інциденту;
– втрата довіри до системи як до джерела істини.

1.5 Ненадійне оповіщення*:* Система може успішно виявити інцидент, але **не повідомити про нього відповідальних осіб**, якщо нотифікація працює лише через один канал (наприклад, email). За даними SolarWinds [6], **27% критичних інцидентів залишились непоміченими** саме через збій у системі сповіщення.

– застосування лише SMTP без fallback-каналу;
– відсутність журналу надсилання;
– обмеження API Telegram/Slack, що спричиняє rate-limit errors.

2. Шляхи підвищення надійності

Надійність системи моніторингу визначається її здатністю продовжувати функціонувати навіть за умов часткових збоїв, втрати зв'язку або перевантаження. Підвищення надійності потребує поєднання технічних, архітектурних та організаційних заходів. Нижче наведено ключові напрямки з розгорнутим поясненням.

2.1 Розподілена архітектура: передбачає розміщення компонентів системи моніторингу на кількох вузлах, які взаємодіють між собою. Це дозволяє зменшити навантаження на кожен окремий компонент і усунути залежність від одного центрального вузла. Наприклад, при розподіленні функцій збору даних, аналітики і оповіщення на різні сервери – кожен з них можна оновлювати чи обслуговувати окремо без зупинки всієї системи. Приклад розподіленої архітектури наведено нижче, на рисунку 2.

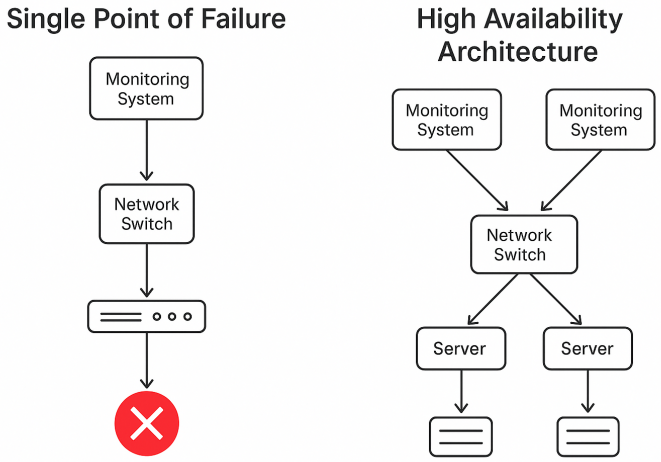

Рис.2 – Приклад розподіленої архітектури

2.2 Резервування компонентів: реалізація активного або пасивного дублювання критичних вузлів. Це може включати:

– використання кластерних рішень для серверів моніторингу (Zabbix HA, Prometheus Federation);

– реплікацію баз даних (наприклад, PostgreSQL з hot standby);

– резервні інтерфейси для надсилання оповіщень (SMS, Telegram, Slack). У разі збою основного компонента резервна система автоматично або напівавтоматично підхоплює навантаження.

2.3 Інтелектуальна фільтрація подій: дозволяє зменшити кількість хибнопозитивних сповіщень, які викликають перевантаження персоналу або втрату довіри до системи. Серед технік фільтрації:

– кореляція подій на основі прецедентів (наприклад, тривога виникає лише при комбінації кількох факторів);

– застосування алгоритмів машинного навчання (Isolation Forest, DBSCAN) для виявлення аномалій у поведінці мережі;

– розумне групування оповіщень за джерелом або типом інциденту.

2.4 Регулярний аудит та тестування: система моніторингу має сама проходити перевірку на працездатність. Це може включати:

– симуляцію подій або збоїв (synthetic events);

– аудит конфігурацій тригерів і каналів зв'язку;

– перевірку логів на наявність недоставлених повідомлень. Регулярне тестування дозволяє запобігти відмовам, які могли б залишитися непоміченими до критичного моменту.

2.5 Автоматизація реакції: у сучасних мережах важливо не лише виявляти проблему, а й реагувати на неї автоматично. Наприклад:

– запуск скрипта перезапуску служби у разі падіння;

– блокування IP-адреси при виявленні аномальної активності;

– автоматичне перемикання маршруту на резервний канал;

– інтеграція з системами автоматизації (Ansible, SaltStack, Puppet). Це суттєво скорочує час усунення інциденту та зменшує навантаження на IT-персонал.

Реалізація навіть частини наведених заходів дозволяє суттєво підвищити стійкість системи моніторингу, скоротити час простоїв і підвищити рівень довіри до її сповіщень.

3. Комплексна архітектура надійної системи моніторингу зображена на рисунку 3.1 та має містити:

– агенти на хостах з можливістю кешування локальної інформації при втраті зв'язку;

– центральний сервер збору подій з реплікованою БД;

– резервний сервер, автоматично активований при відмові основного;

– модуль фільтрації на основі правил і AI;

– багатоканальний модуль нотифікацій (email, Telegram, SMS, Slack API);

– панель адміністратора з можливістю запуску скриптів реагування.

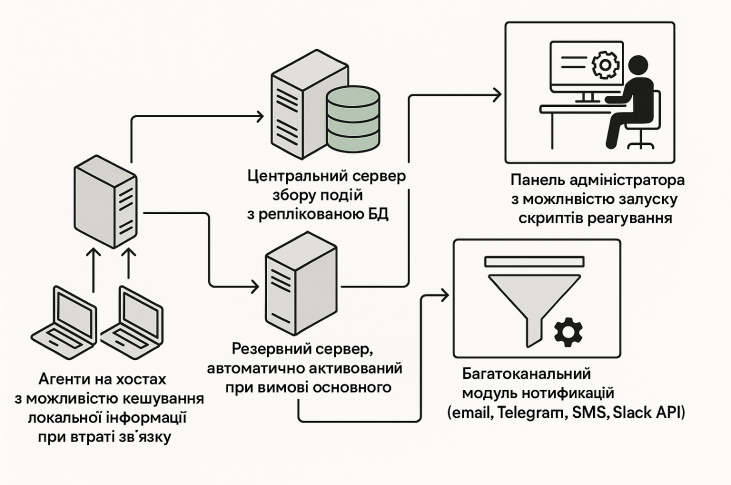

Рис. 3 – Комплексна архітектура надійної системи моніторингу

Висновки та перспектива подальших досліджень. Надійність моніторингових систем є ключовим фактором стабільної роботи сучасної IT-інфраструктури. У статті проаналізовано найпоширеніші загрози для таких систем, серед яких – єдина точка відмови, надмірне навантаження, хибні спрацювання, втрати даних та ненадійне оповіщення. Кожна з цих проблем здатна призвести до серйозних порушень безперервності бізнес-процесів, значних фінансових збитків або втрати контролю над критично важливими ресурсами.

Запропоновані технічні та архітектурні рішення – зокрема впровадження розподіленої інфраструктури, резервування компонентів, застосування інтелектуальних алгоритмів фільтрації подій, регулярне тестування та автоматизована реакція на інциденти – дозволяють суттєво підвищити стійкість моніторингових систем до збоїв.

Особливу увагу в майбутніх дослідженнях слід приділити використанню самонавчальних систем для динамічного налаштування порогів, а також вивченню впливу людського чинника на ефективність реагування. Крім того, перспективним є розвиток мультиканальних систем оповіщення з пріоритетним маршрутизуванням тривог та аналізом зворотного зв'язку.

Комплексний підхід до проєктування і розвитку систем моніторингу дозволяє не лише уникати інцидентів, а й створює основу для довготривалої надійної та безпечної експлуатації корпоративної IT-інфраструктури.

Список бібліографічного опису

1. Barabanov A., Chen M., Gupta R. High Availability in Network Monitoring Systems. Journal of Network and Systems Management. 2020. Vol. 28, No. 3. P. 467–484.
2. Олійник О. О. Надійність інформаційних систем: проблеми і підходи. Вісник НТУУ «КПІ». 2021. № 4. С. 45–52.
3. Nagios Core Documentation. URL: https://www.nagios.org/documentation/ (дата звернення: 02.03.2025).
4. Zabbix Architecture Overview. URL: https://www.zabbix.com/documentation/current/manual/architecture (дата звернення: 02.03.2025).
5. Single point of failure // Wikipedia. URL: https://en.wikipedia.org/wiki/Single_point_of_failure (дата звернення: 11.03.2025).
6. SolarWinds. IT Trends Report 2024 URL: https://www.solarwinds.com/ (дата звернення: 15.05.2025).
7. Splunk. Alert Noise Reduction Study, 2023. URL: https://www.splunk.com/ (дата звернення: 22.04.2025).
8. Red Hat. State of Enterprise Infrastructure 2023 . URL: https://www.redhat.com/ (дата звернення: 03.05.2025).

References

1. Barabanov A., Chen M., Gupta R. High Availability in Network Monitoring Systems. *Journal of Network and Systems Management*. 2020. Vol. 28, No. 3. P. 467–484.
2. Oliinyk O. O. Reliability of Information Systems: Problems and Approaches. *Bulletin of NTUU "KPI"*. 2021. No. 4. P. 45–52.
3. Nagios Core Documentation. URL: https://www.nagios.org/documentation/ (accessed: 02.03.2025).
4. Zabbix Architecture Overview. URL: https://www.zabbix.com/documentation/current/manual/architecture (accessed: 02.03.2025).
5. Single point of failure. Wikipedia. URL: https://en.wikipedia.org/wiki/Single_point_of_failure (accessed: 11.03.2025).
6. SolarWinds. IT Trends Report 2024. URL: https://www.solarwinds.com/ (accessed: 15.05.2025).

7. Splunk. Alert Noise Reduction Study, 2023. URL: https://www.splunk.com/ (accessed: 22.04.2025).
8. Red Hat. State of Enterprise Infrastructure 2023. URL: https://www.redhat.com/ (accessed: 03.05.2025).