

## АВТОМАТИЗОВАНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ ОБЛАДНАННЯ РОЗПОДІЛЕНИХ СИСТЕМ

В сучасному світі питання автономності та незалежності у сфері електропостачання, теплопостачання, водопостачання або інтернету стає дедалі популярнішим та перспективнішим [1]. Це пов'язано зі зростаючою свідомістю людей щодо необхідності екологічної сталості та забезпечення своєї безпеки та комфорту у повсякденному житті. А згадуючи про ситуацію в нашій країні, ця тема набуває актуальності та потребує комплексного підходу та вдосконалення відповідних систем та технологій з метою забезпечення максимальної надійності та ефективності їх роботи.

Саме тому для цього використовують найсучасніші технологічні прилади, які мають достатній рівень самостійності та потребують лише мінімальних втручань з боку людини. Зазвичай, для більш зручного використання цих технологічних приладів та комфортного стеження за ними, їх конструкція може припускати можливість передачі даних свого поточного стану через «хмару» у додаток на будь-якому електронному пристрої, наприклад телефоні або комп'ютері [2, 3]. Але у випадках використання технологічних приладів від різних фірм часто виникає проблема несумісності, через що за кожним приладом доводиться стежити у різних додатках [4]. Це незручно і іноді може призвести до неприємних наслідків або навіть катастроф.

Через це виникає потреба у створенні єдиного інтерфейсу, який міг би сприймати інформацію з «хмари» незалежно від фірми-виробника та відображати дані усіх технологічних приладів у будівлі, або у кількох будівлях.

Об'єктом дослідження є хмарні сервіси моніторингу стану віддалених технологічних об'єктів. Предметом дослідження є підходи до підвищення ефективності процесу збору інформації з різних хмарних сервісів. Завданням дослідження є оптимізація процесу збору та первинної обробки інформації в системах моніторингу.

Для виконання поставленого завдання, передбачається створення окремого програмного засобу, який би виконував роль системи моніторингу. За приклад ми можемо взяти аналогічну програму Grafana, яка може вести зчитувати, аналізувати та порівнювати дані з опалювальних котлів [5]. Приклад інтерфейсу програмної системи Grafana наведено на рис. 1.



Рисунок 1 – Інтерфейс програмної системи візуальних даних Grafana

Ця програма написана на мовах програмування Go та TypeScript, тому вона здатна виконувати парсинг. Парсинг (від англ. parsing), або синтаксичний аналіз, – це процес аналізування інформації з метою визначення її структури та витягування корисної інформації [6]. Для вирішення нашої задачі програма повинна спочатку виконувати парсинг, завдяки якому інформація отримуватиме первинну обробку та буде виводитися у вигляді, зручному для користувача.

Парсинг є основною функцією програми та виконує головну роль у моніторингу за поточними даними з технологічного об'єкту, але наша програма передбачає спостереження за декількома об'єктами, які можуть взаємодіяти один з одним та не мають спільного потоку даних [7, 8]. До того ж, для створення більш широкого та докладного аналізу поточних даних та характеру їх змін, потрібно враховувати дані минулих станів технологічних об'єктів. Для збереження цієї інформації потрібно створити базу даних. Основою будь-якої бази даних є сервери, тому включення баз даних до нашої програми передбачає створення власної або використання пропонованої інфраструктури. Архітектура передачі даних до серверів та користувача наведено на рис. 2.

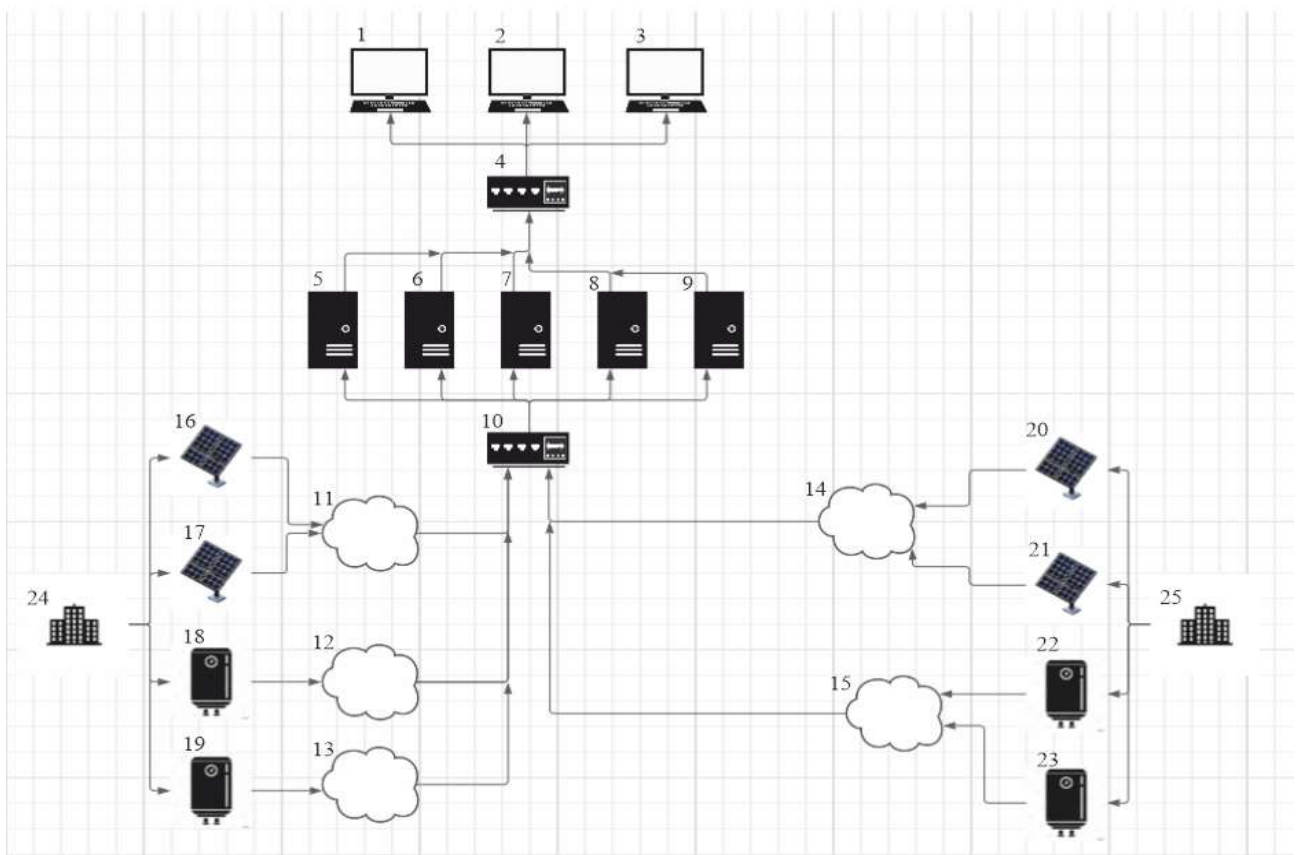


Рисунок 2 – Архітектура мережі передачі даних з технологічних приладів до користувача через різні хмарні сервіси:

1,2,3 – електронні пристрої користувача; 4,10 - балансувальники; 5,6,7,8,9 – сервери; 11,12,13,14,15 – хмарні сервіси, створені фірмами-виробниками технологічних приладів; 16,17,18,19,20,21,22,23 – технологічні прилади (16,17,20,21 – сонячні панелі, 18,19,22,23 – котли опалення, як приклад), 24,25 – будівлі, в яких встановлені технологічні прилади.

Для одночасної реалізації парсингу і роботи з базами даних в одній програмі, доцільно обрати мову програмування C# [9]. Для парсингу ця мова програмування має бібліотеки HtmlAgilityPack та AngleSharp. HtmlAgilityPack є бібліотекою для парсингу HTML-сторінок, яка дозволяє отримувати доступ до елементів сторінки та їх атрибутів. AngleSharp є бібліотекою, яка дозволяє парсити не тільки HTML-сторінки, але й XML-файли та інші

формати даних. Також мова С# має розширення LINQ, яке дозволяє легко обробляти та фільтрувати дані, що отримані з веб-сайту.

.NET Framework має багато вбудованих засобів для роботи з реляційними базами даних, такими як Microsoft SQL Server, Oracle, MySQL та інші [10]. Для цього в .NET Framework є технології ADO.NET та LINQ to SQL, які дозволяють зчитувати та зберігати дані в базі даних.

ADO.NET є технологією, яка дозволяє створювати з'єднання з базою даних, виконувати запити до бази даних та обробляти результати запитів. ADO.NET також містить багато інших функцій, таких як використання транзакцій, обробка помилок, оптимізація запитів та інші.

LINQ to SQL є технологією, яка дозволяє створювати об'єкти-сутності на основі таблиць бази даних та здійснювати з ними роботу на рівні об'єктно-орієнтованої програми.

Висновок. Застосування сучасних інформаційних технологій дозволяє здійснювати моніторинг параметрів розподілених об'єктів, що використовують різні хмарні сервіси. В якості подальшого дослідження, планується створення методів зворотного зв'язку програми та користувача з технологічними об'єктами, що зробить програму системою моніторингу та управління. Це дозволить програмі взаємодіяти з обладнанням у випадках його критичного перенавантаження або дасть можливість користувачу дистанційно керувати технологічним процесом.

#### ЛІТЕРАТУРА:

1. Енергетичні ресурси та потоки / Шидловський А.К., Віхорев Ю.О., Гінайло В.О. та ін. К.: Українські енциклопедичні знання, 2003. 472 с.
2. Яковицький І.Л. Технологія «хмарних обчислень» як інструмент створення інформаційної інфраструктури управління. *Комунальне господарство міст*. №102. С. 320-327.
3. Cloud Automation: Why, Where and How. URL: <https://bluexp.netapp.com/blog/cloud-automation-why-where-and-how-cvo-blg>
4. Шишак А.В., Пупена О.М. На шляху до Індустрії 4.0: інтеграція існуючих АСУТП з хмарними сервісами. *Автоматизація технологічних і бізнес-процесів*, №10, Т. 1, 2018. С. 33-39.
5. Grafana Dashboard Monitoring - Store & Visualize Your Metrics. URL: [https://newrelic.com/lp/grafana-monitoring?utm\\_medium=cpc&utm\\_source=google&utm\\_campaign=EVER-GREEN\\_NB\\_SEARCH\\_GRAFANA\\_EMEA\\_CENTRAL\\_EN&utm\\_network=g&utm\\_keyword=grafana&utm\\_device=c&\\_bt=469549587759&\\_bm=e&\\_bn=g&gclid=CjwKCAjwoIqhBhAGEiwArXT7K2ewTFuTloWp-nt2ta7b2aW8DQKPaоXEX\\_jZo\\_AAMa3T\\_8wBoate8xoCH2cQAvD\\_BwE](https://newrelic.com/lp/grafana-monitoring?utm_medium=cpc&utm_source=google&utm_campaign=EVER-GREEN_NB_SEARCH_GRAFANA_EMEA_CENTRAL_EN&utm_network=g&utm_keyword=grafana&utm_device=c&_bt=469549587759&_bm=e&_bn=g&gclid=CjwKCAjwoIqhBhAGEiwArXT7K2ewTFuTloWp-nt2ta7b2aW8DQKPaоXEX_jZo_AAMa3T_8wBoate8xoCH2cQAvD_BwE)
6. Синтаксичний аналіз. URL: [https://www.wiki-data.uk-ua.nina.az/Синтаксичний\\_аналіз.html](https://www.wiki-data.uk-ua.nina.az/Синтаксичний_аналіз.html)
7. Earley J. An efficient context-free parsing algorithm. *ACM* 13, 2 (Feb 1970). – P. 94–102.
8. Швороб І.Б. Порівняльний аналіз методів синтаксичного розбору текстів. *Вісник Національного університету "Львівська політехніка". Збірник наукових праць. Серія: Інформаційні системи та мережі*. № 814, 2015. С. 197 -202.
9. The AZ of Programming Languages: C#. URL: <https://www.computerworld.com/au/>
10. Crane R., Resnick S., Bowen C. Essential Windows Communication Foundation (WCF): For .NET Framework 3.5. *Pearson Education*. 2008. 608 p.