

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-ДОСТУПНОЇ ЛАБОРАТОРІЇ НАПІВПРОВІДНИКІВ

Роман Назаревич¹, Іван Болеста²

^{1,2} Львівський національний університет імені Івана Франка,
кафедра радіофізики та комп'ютерних технологій, Львів, Україна

¹Email: roman.nazarevych@gmail.com, ORCID: 0009-0003-0104-3280

²Email: ivan.bolesta@lnu.edu.ua, ORCID: 0000-0002-7580-3495

© Назаревич Р., Болеста І., 2025

У роботі представлені підходи до створення автоматизованої інтернет-доступної лабораторії напівпровідників, зокрема керуючого програмного комплексу, що забезпечує інтернет-доступність лабораторії. Описано функціонал і структуру програмно-апаратного комплексу, розробленого з урахуванням відомих рішень, а також варіанти його реалізації з використанням інтернету, хмарних і граничних обчислень. Розглянуто варіанти локальної реалізації з підвищеною стійкістю до форс-мажорних факторів і кіберзагроз. Проаналізовано та запропоновано оптимальні рішення для бази даних і протоколів комунікації, вибрано рішення, що задовольняють вимоги розподіленої системи з обмеженими ресурсами граничних вузлів, зокрема обмеженнями за оперативною пам'яттю та обчислювальними потужностями. Запропонована реляційна база даних SQLite має мінімальні вимоги до ресурсів і водночас надає більшість можливостей реляційних баз даних, а запропонований протокол gRPC є одним із найшвидших і ефективно використовує мережеві ресурси. Розглянуто основні підходи до реалізації граничних обчислень (Cloudlet, MEC, Fog Computing) і вибрано оптимальний варіант для інтернет-доступної лабораторії. Представлено загальну структуру та організацію базової лабораторної, клієнтської та дата-складових частин програмно-апаратного комплексу. Описано високорівневий алгоритм взаємодії основних вузлів системи та розподілу обчислювальних задач між ними. Цей алгоритм, використовуючи переваги гібридних (хмарно-граничних) обчислень, дозволяє прискорити опрацювання результатів експерименту, зменшити навантаження на канал зв'язку з інтернетом і знизити обчислювальні навантаження на хмарний вузол. Крім того, цей алгоритм підтримує роботу в автономному режимі, що дозволяє системі працювати без підключення до інтернету без обмежень функціоналу.

Ключові слова – інтернет-доступна лабораторія, програмно-апаратний комплекс, граничні обчислення, хмарні обчислення, розподілені обчислення, гетерогенні обчислення, протоколи інтернет-зв'язку.

Постановка проблеми

Автоматизація і комп'ютеризація лабораторних досліджень – давній, але актуальний і до сьогодні тренд у розвитку науково-дослідного і навчального процесу, зокрема, у закладах вищої освіти (ЗВО). На сучасному етапі цей тренд доповнюється трендом на забезпечення онлайн-доступності цих процесів у зв'язку з розширенням використання дистанційної роботи і навчання. В літературі (Chushchak et al., 2023; Holubinka & Khudyi, 2024; Savkiv, 2019, та ін.) описано приклади реалізації інтернет-доступних наукових і науково-навчальних лабораторій різного рівня, а також інші рішення подібного функціоналу. Одним із основних сучасних підходів для вирішення таких задач є використання “хмарних обчислень”. Цей підхід зручно використовувати для вирішення таких задач, але він має ряд недоліків які покликана вирішити нова концепція, вона називається “граничні обчислення”. Цей підхід є одним із варіантів розподілених обчислень, основною метою його є

зменшення затримок і навантаження на мережу за рахунок перенесення частини обчислень ближче до джерела даних та клієнта, до так званої “границі”.

Задача створення автоматизованої інтернет-доступної лабораторії напівпровідників актуалізувалась для кафедри радіофізики та комп’ютерних технологій факультету електроніки та комп’ютерних технологій ЛНУ ім. Івана Франка у зв’язку з розширенням використання дистанційної дослідницької роботи і навчання, а також з розвитком нових напрямів досліджень напівпровідникових матеріалів та мікро- і наноелектронних структур, у т.ч. у напрямку фотоніки і наноплазмоніки. Потреби дистанційного доступу викликані різними обставинами, а можливості з’явилися і постійно зростають у зв’язку з розвитком комп’ютерної техніки і цифрової електроніки, можливістю використання швидкісної мережі Інтернет, мереж мобільного зв’язку нових поколінь, що забезпечують повсюдно доступні мобільний інтернет, хмарні сервіси тощо. Для створення автоматизованої інтернет-доступної лабораторії напівпровідників необхідно проаналізувати комплекс задач, які повинні бути при цьому вирішені, врахувати особливості наявних апаратних засобів та програмного забезпечення, передбачити можливості подальшого розвитку та доповнення такої лабораторії як апаратними, так і програмними засобами. Варто також передбачити можливість використання створюваного програмного комплексу для інших подібних лабораторій (за відповідної його адаптації), а також для вирішення подібного кола задач в інших областях застосування. Також слід звернути увагу на надійність функціонування такої лабораторії в різних умовах, у т.ч. форс мажорних, а також на питання кібербезпеки при її функціонуванні. Одним із ключових аспектів дослідження буде вивчення шляхів організації розподілених обчислень для оптимізації роботи онлайн-лабораторії, застосування підходів “граничних обчислень” для цих задач з метою підвищення продуктивності, економічної ефективності, надійності роботи та кібербезпеки. Також заплановано задіяння в он-лайні наявних та нових спеціально створених методик комп’ютерного моделювання для підвищення ефективності дослідницької роботи такої лабораторії, особливо у напрямку фотоніки і наноплазмоніки.

Аналіз останніх досліджень та публікацій

Щодо створення автоматизованої інтернет-доступної лабораторії напівпровідників є певні напрацювання (Bolesta et al., 2019a, 2019b), які враховують потреби лабораторних досліджень і слугують базою для подальших робіт у цьому напрямку. Також є розроблення авторів щодо розподілених моніторингових геоінформаційних систем подібного функціоналу і опрацювання та препроцесингу даних у них (Nazarevych & Nazarevych, 2011; Morozov et al., 2014, Bolesta et al., 2019a). При цьому доцільно врахувати вже розроблені і описані в літературі підходи та рішення з метою вибору найкращих способів та засобів створення програмного забезпечення для такої інтернет-доступної лабораторії (Chushchak et al., 2023; Ivantyshyn & Burov, 2023; Holubinka & Khudyi, 2024, та ін.).

Для пошуку оптимальних шляхів і способів вирішення перелічених задач нами проаналізовано описані в наукових публікаціях (Chushchak et al., 2023; Ivantyshyn & Burov, 2023; Niswar et al., 2024, та ін.) рішення в цьому напрямку, результати власних попередніх робіт (Bolesta et al., 2019a; Nazarevych & Nazarevych, 2011; Morozov et al., 2014), у т.ч. наявні на кафедрі напрацювання щодо цих питань (Bolesta et al., 2019a, 2019b, та ін.) Для розроблення загальної структури програмного забезпечення проаналізовано наявну структуру апаратних засобів лабораторії (Bolesta et al., 2019a), перспективи її розвитку та відомі рішення подібного функціоналу (Chushchak et al., 2023; Savkiv, 2019) щодо організації інтернет-доступних лабораторій, порталів та інформаційних мереж, у т.ч. раніше створені авторами (Nazarevych & Nazarevych, 2011; Morozov et al., 2014). Звернуто увагу на особливості комп’ютеризації конкретних лабораторних установок (Bolesta et al., 2019a, 2019b; Nazarevych & Nazarevych, 2011), загальну організаційну структуру таких комплексів (Chushchak et al., 2023; Ivantyshyn & Burov, 2023; Nazarevych & Nazarevych, 2011), особливості та оптимізацію організації баз даних (Holubinka & Khudyi, 2024; Kuzev et al., 2020), організацію багаторівневого адміністраторського та клієнтського доступу (Nazarevych & Nazarevych, 2011; Morozov et al., 2014; Bolesta et al., 2019a), питання надійності та безпеки функціонування системи (Faizulin & Nazarevych, 2024; Nazarevych & Nazarevych, 2011). Детальніший аналіз публікацій з конкретних питань наведено нижче у відповідних підрозділах, де розглядається вибір і обґрунтування конкретних технічних рішень.

Детальний огляд способів організації граничних обчислень, потенційних проблем і рішень наведено в (Alsurdeh, 2020; Niswar et al., 2024; Skvorc et al., 2014, та ін.). Цю інформацію нами проаналізовано для вибору оптимізованих підходів і реалізацій щодо створення програмного забезпечення інтернет-доступної лабораторії напівпровідників, така задача є актуальною для кафедри радіофізики та комп'ютерних технологій факультету електроніки та комп'ютерних технологій ЛНУ ім. Івана Франка.

Для формулювання конкретних вимог до створюваного програмного забезпечення (ПЗ) інтернет-доступної лабораторії напівпровідників проаналізовано наявні методичні напрацювання щодо створення такої лабораторії в локальному виконанні (Bolesta et al., 2019a, 2019b, та ін.). Лабораторія покликана досліджувати напівпровідникові матеріали та мікро- і наноелектронні структури, у т.ч. у напрямку фотоніки і наноплазмоніки. Для цього створюється лабораторна установка, що забезпечує формування необхідних електричних, теплових та інших режимів тестування досліджуваних об'єктів, багатоканальне вимірювання цих режимів і параметрів, моніторинг їх змін у часі з необхідним розділенням за величинами і часовою детальністю, з фіксацією (за необхідності) різних оптичних ефектів (спектрів випромінювання і поглинання, діаграм розсіювання тощо). Далі проводиться опрацювання результатів експериментів та їх порівняння з результатами відповідних теоретичних розрахунків, що надає інформацію для модифікації відповідних мікроелектронних технологій і отримання у підсумку необхідних за характеристиками напівпровідникових та наноплазмонних матеріалів та мікро- і наноелектронних структур. При цьому приходиться враховувати, що більшість із готових програм, які можна було б використати як загальну основу чи в окремих модулях створюваного ПЗ, існують або у формі програм з відкритим кодом на мовах нижчого рівня, або у формі готової консольної програми з обмеженим набором вхідних параметрів, або є пропріетарним програмним забезпеченням (Laboratory..., 2025; Lumerical..., 2025, та ін.). Інша проблема полягає в тому, що експериментальні результати немає можливості порівнювати з результатами комп'ютерного моделювання, у т.ч. проведеного за авторськими методиками (Bolesta et al., 2019a, 2019b).

Формулювання цілі статті

Метою роботи є розроблення структури програмного комплексу автоматизованої інтернет-доступної лабораторії напівпровідників та наноструктур, для проведення досліджень у галузях фотоніки і наноплазмоніки, що забезпечуватиме її інтернет-доступність як щодо керування лабораторним експериментом, так і щодо подальшого опрацювання отримуваних результатів. За результатами аналізу поставлених перед лабораторним комплексом задач сформульовано вимоги до функціональних характеристик, апаратного та програмного забезпечення програмного комплексу.

Виклад основного матеріалу

На основі аналізу перелічених вище даних розроблено загальну апаратно-програмну структуру лабораторного комплексу, загальні алгоритми роботи і взаємодії його складових, принципи організації бази даних експериментів та ін. Система будується на принципах мікросервісної архітектури з використанням локальних рішень, можливостей Інтернету і хмарних сервісів. Сформовано концепцію програмної реалізації окремих модулів програмного комплексу та вирішення проблемних питань, зокрема, шляхом проведення необхідних досліджень для пошуків оптимізованих варіантів програмно-алгоритмічних реалізацій. При опрацюванні перелічених питань основну увагу приділено забезпеченню необхідного рівня доступності та надійності взаємодії всіх систем і підсистем комплексу через мережу Інтернет, пов'язані з цим питання вибору доступних і найбільш ефективних програмних продуктів та середовищ тощо.

Загальна структура і функціонал лабораторного комплексу

Лабораторний комплекс повинен забезпечувати проведення лабораторних експериментів з дослідження мікро- і наноелектронних структур та матеріалів, напівпровідникових пристроїв, зокрема, світлодіодів, вивчення ефектів і явищ у галузі фотоніки і наноплазмоніки. Для цього він має об'єднувати комп'ютеризовану лабораторну установку, клієнтські термінали і базу даних лабораторних експериментів. У штатному режимі роботи передбачено зв'язок між цими частинами через локальну мережу, мережу Інтернет і хмарні сервіси (рис. 1). При реалізації комплексу використовується поєднання хмарних і граничних обчислень, зокрема, гібридну модель хмарно-граничні обчислення (Alsurdeh, 2020) У цьому варіанті обчислення розподіляються між хмарию і граничними вузлами системи. В такій конфігурації особливої ваги набирає ефективне планування і розподіл задач між вузлами системи.

Функціональні можливості програмної частини комплексу – лабораторна, клієнтська і дата-складові, об'єднані програмним ядром у єдину систему. Лабораторна частина комплексу є комп'ютеризованою лабораторною установкою (див. рис. 1) і забезпечує реалізацію лабораторного експерименту, вимірювання необхідних параметрів досліджуваного пристрою (отримання первинних даних) і запис цих даних у базу даних разом з протоколом самого експерименту.

Комп'ютеризована лабораторна установка в загальному об'єднує систему керування експериментом (електронну, електронно-механічну, електронно-оптичну та ін.), об'єкт досліджень, вимірювальну систему. Система керування експериментом реалізує весь хід експериментів з об'єктом досліджень, задаючи необхідні режими його роботи і тестування, вимірює необхідні характеристики об'єкта в процесі експерименту, забезпечує керування ходом експерименту через систему керування з контролем реалізації програми експерименту, а також забезпечує прийом даних з вимірювальної системи з можливістю керування процесом вимірювань. У випадку віддаленого керування ходом експерименту через клієнтський інтерфейс лабораторна частина комплексу може працювати в режимі віддаленого доступу.

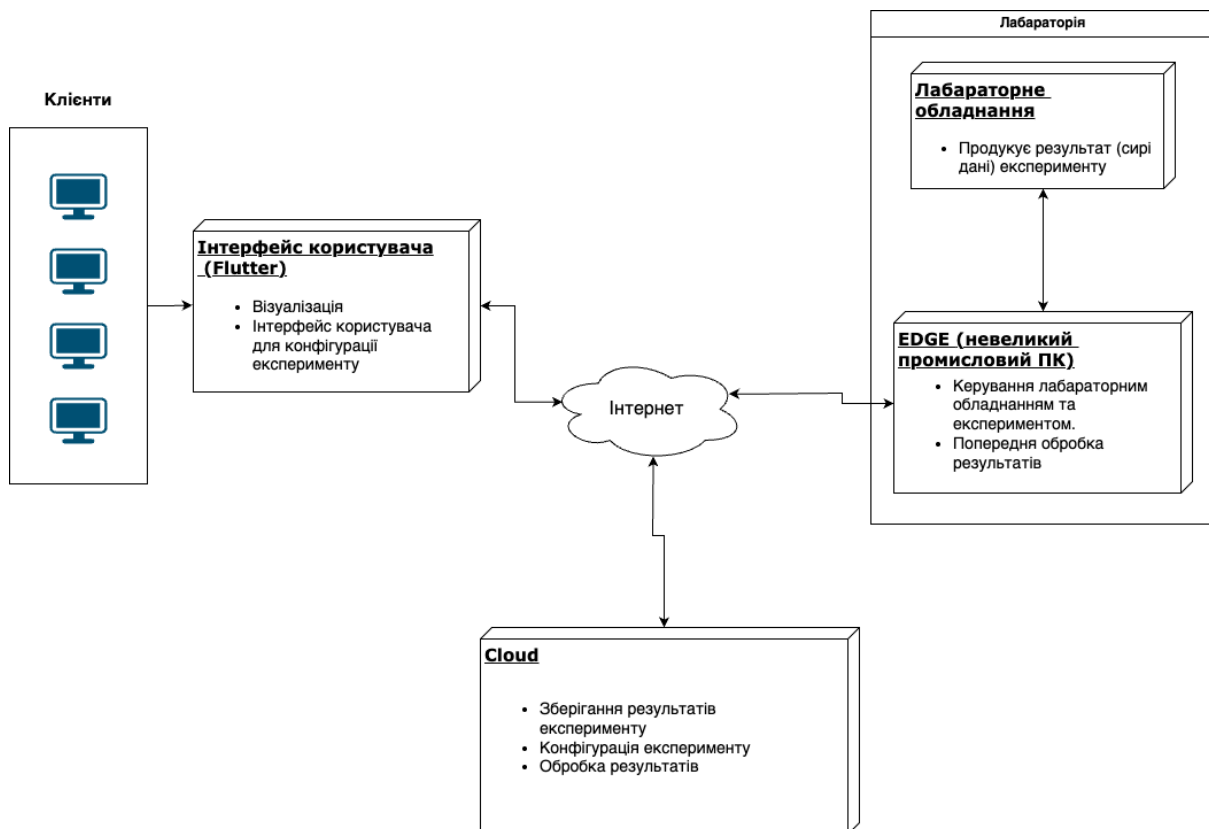


Рис. 1. Функціональна структура лабораторного комплексу (пояснення в тексті)

Клієнтська частина забезпечує задання необхідних параметрів лабораторного експерименту, керування його ходом у часі (за необхідності) у взаємодії з лабораторною частиною і подальше опрацювання отриманих результатів у взаємодії з дата-частиною. Дата-частина комплексу забезпечує збереження первинних результатів експериментів і результатів їх опрацювання в базі даних з видачою за запитом з клієнтських терміналів. Опрацювання результатів може здійснюватися як на ресурсах лабораторної частини комплексу, так і на ресурсах клієнтської чи дата-частин, на ресурсах обчислювального кластеру або на хмарних ресурсах провайдера послуг.

За формажорних обставин (відсутність електроживлення від мережі / відсутність інтернету) передбачено можливість автономної роботи лабораторного програмно-апаратного комплексу в мінімальній / базовій конфігурації (без підключення до мережі Інтернет) і з базовим функціоналом шляхом взаємодії лабораторної, клієнтської і дата-складових через локальну мережу і з автономним живленням (автономний генератор / система, що містить сонячні батареї – конвертери – акумулятори). В цьому випадку функціонал хмари (сховище і опрацювання даних) може реалізуватись як на клієнтській частині комплексу (персоналізовано), такі і (для загального доступу клієнтам) на лабораторну частину комплексу (EDGE – Граничний вузол) або на Cloudlet – невеликому локальному сервері дата-центру (детальніше див. далі).

Оскільки лабораторія покликана реалізовувати як навчальні, так і науково-дослідні програми, клієнтами можуть бути як студенти на лабораторних заняттях, так і дослідники – на наукових експериментах. Вони ж можуть бути (поряд із спеціальним лаборантом-оператором) операторами лабораторної установки. Задачі оператора – встановлення в установку дослідних зразків, підключення відповідних виконавчих елементів і вимірювальних каналів, включення установки і налаштування робочих режимів, контроль відпрацювання програми експерименту, оперативний зв'язок з клієнтами.

Лаборант-оператор може бути і адміністратором лабораторного комплексу. Задачі адміністратора – планування і розподіл навантаження лабораторного комплексу в часі по клієнтах і їх дослідних програмах (у взаємодії з клієнтами), оперативний зв'язок з клієнтами для вирішення різних питань у ході реалізації лабораторних експериментів, керування доступом до результатів з бази даних тощо. Адміністрування на початковому етапі створення лабораторії планується здійснювати з лабораторної частини комплексу (EDGE – Граничний вузол).

Клієнтська частина програмного забезпечення може встановлюватись на клієнтських комп'ютерах лабораторії (як локальних, так і віддалених), а також на комп'ютери (ноутбуки, планшети, смартфони) користувачів, яким надано відповідний доступ до роботи в створюваній інтернет-доступній лабораторії.

На початковому етапі розвитку проекту, в якості лабораторної установки виступає система спектрофотометричного обладнання StellarNet для експериментів з реєстрації люмінесценції наноструктур для органічних світлодіодних структур з поляризованим світленням (Bolesta et al., 2019a, 2019b). Керування експериментом забезпечує лабораторна частина комплексу (EDGE – Граничний вузол), встановлена на невеликий промисловий ПК на архітектурі x86 з OS Windows, це продиктовано вимогами програмного забезпечення для керування лабораторним обладнанням. В подальшому передбачена можливість нарощування кількості лабораторних установок у складі лабораторії.

Також передбачено додаткове підключення в систему довільної кількості клієнтських комп'ютерів (як через Інтернет, так і через локальну мережу), у т.ч для додаткового лабораторного обладнання, так і бути додатковими обчислювальними потужностями в системі розподілених обчислень даного комплексу.

Використання при розробці ПЗ архітектури мікросервісів забезпечує просте нарощування функціоналу лабораторного комплексу, що дає можливість запроваджувати нову функціональність у вигляді мікросервісів у загальний програмний пакет або в окремі програмні модулі без істотних змін чи переробок існуючого ПЗ та функціоналу. Це дає можливість легко нарощувати як апаратний склад лабораторного комплексу з своїми програмними модулями – драйверами, так і використовувати нові

методи моделювання, лабораторних експериментів і опрацювання даних, швидко інтегрувати їх з існуючим ПЗ і методами лабораторних досліджень. Мікросервіси об'єднані у функціональні блоки – введення даних, моделювання експерименту, фізичний експеримент, опрацювання результатів. При цьому оптимізується також програмно-апаратне навантаження на різні ресурси комплексу. Наприклад, мікросервіс моделювання методом дискретного диполя (DDA) вимагає більше ресурсів пам'яті та ЦП для розпаралелювання обчислень, а мікросервіс імпорту даних вимагає відповідної конфігурації сховища даних (Bolesta et al., 2019a).

Вибір оптимальної архітектури розподілених граничних обчислень

Для обґрунтування вибору загальної схеми функціонування ПЗ лабораторного комплексу розглянемо основні архітектури для реалізації парадигми “Граничних обчислень”: Cloudlet, MEC (Multi-access Edge Computing) та Fog Computing (Chamas et al., 2017).

Cloudlet – це невеликий локальний дата-центр, який знаходиться ближче до кінцевих користувачів і забезпечує швидше опрацювання даних порівняно з традиційною хмарною інфраструктурою. Cloudlet діє як проміжний рівень між користувацькими пристроями та великими хмарними дата-центрами.

MEC – це архітектура, розроблена для обчислень на краю мережі мобільного зв'язку (4G/5G). Основна мета MEC – знизити затримку шляхом опрацювання даних безпосередньо на базових станціях операторів мобільного зв'язку. ETSI (European Telecommunications Standards Institute) є головним регулятором, який розробляє та стандартизує MEC. У 2014 році в ETSI було створено спеціальну MEC Industry Specification Group (ISG), яка визначає стандарти, архітектуру та можливості MEC.

Fog Computing розширює ідею Edge Computing, додаючи рівень проміжних вузлів між кінцевими пристроями та хмарою. “Туманні обчислення” дозволяють опрацьовувати дані в розподіленій системі перед їх передачею в хмару.

Таблиця 1.

Порівняння Cloudlet, MEC та Fog Computing

Архітектура	Де розташовані обчислення?	Основне призначення	Типові сценарії використання
Cloudlet	Поблизу кінцевих користувачів (маленькі дата центри)	Зменшення навантаження на хмару, швидка обробка	Кампуси, лікарні, відео аналітика
MEC	На рівні базових станцій мобільної мережі	Обробка даних для 5G, зниження затримки	Автономні автомобілі, AR/VR, розумні міста
Fog Computing	Між IoT-пристроями та хмарою (розподілені вузли)	Масштабовані розподілені обчислення	Промисловість 4.0, транспорт, безпека

Виходячи із загальних вимог, наведених у розділі “Загальна структура і функціонал лабораторного комплексу” та проведеного аналізу впливає, що оптимальною архітектурою буде Cloudlet. У структурі створюваної інтернет-доступної лабораторії Cloudlet буде проміжним рівнем між лабораторною установкою, клієнтськими комп'ютерами і хмарними сервісами, він виконуватиме функції локального дата-центру (база даних і обробка). MEC може використовуватись для забезпечення взаємодії з клієнтами інтернет-доступної лабораторії через мобільний Інтернет. Fog Computing призначений для масштабованих розподілених обчислень, на сьогодні таких задач для інтернет-доступної лабораторії не планується.

Загальний алгоритм функціонування ПЗ лабораторного комплексу

Завданням програмного забезпечення є, як уже зазначалося, забезпечення зв'язку між лабораторною, клієнтською і дата-складовими лабораторного програмно-апаратного комплексу і поєднання їх функціонально в одне ціле. Зокрема, це керування ходом лабораторних експериментів (спільно з лабораторною та клієнтською частинами), запис результатів експериментів у базу даних (з відповідними атрибутами і протоколами експериментів – взаємодія з лабораторною, клієнтською і дата-складовими комплексу), видача даних клієнту і на опрацювання (якщо опрацювання даних здійснюється окремими спеціальними обчислювальними ресурсами), ведення бази первинних даних і результатів їх опрацювання та взаємодія між нею та клієнтом (взаємодія з лабораторною, клієнтською і дата-складовими комплексу).

На рис. 2 у вигляді діаграми послідовностей (Sequence diagram) зображена схема взаємодії основних компонентів системи а також зв'язки та послідовність основних процесів, які відбуваються у цих компонентах.

1. Запуск експерименту (Cloud → Edge Node 1):
 - a. Хмара (Cloud) ініціює експеримент, надсилаючи команду запуску до Edge Node 1.
 - b. Завдання (A) – це процес керування експериментом на рівні хмари, що включає генерацію команд для лабораторного обладнання.
2. Керування лабораторним обладнанням Edge Node 1 → Lab:
 - a. Edge Node 1 отримує команду від хмари та виконує завдання (B).
 - b. Завдання (B) передбачає надсилання команди для керування лабораторним обладнанням та запуску експерименту.
 - c. Лабораторне обладнання (Lab) отримує команду та починає виконання експерименту.
 - d.

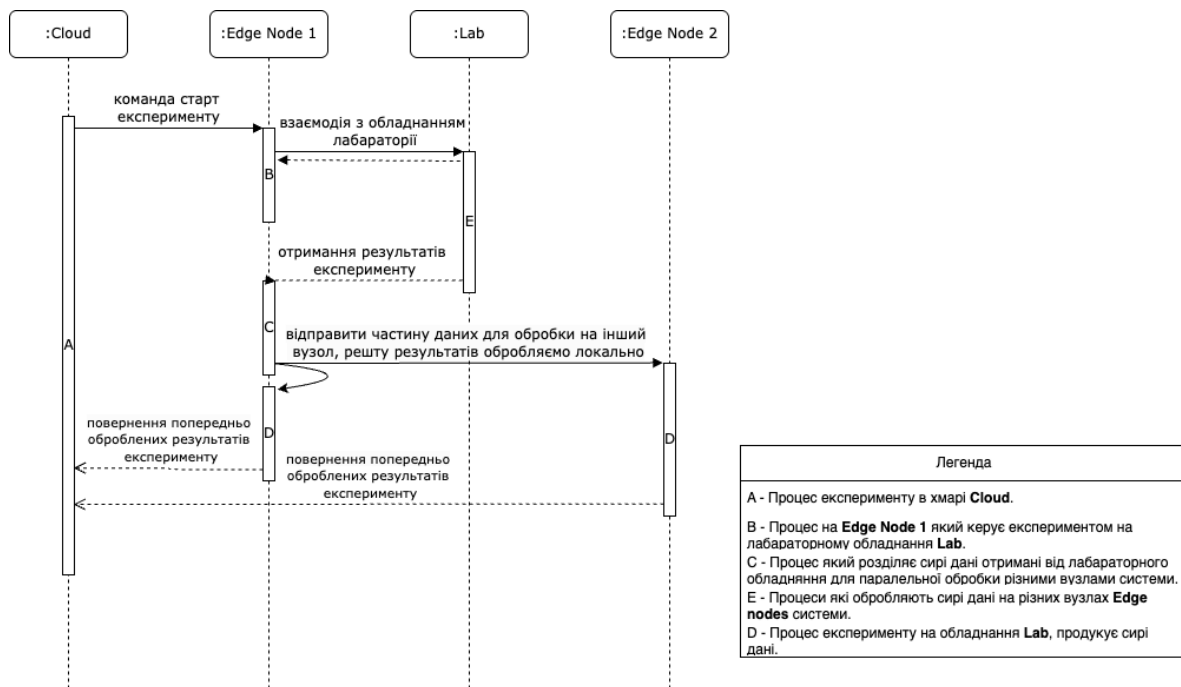


Рис. 2. Діаграма послідовності проведення експерименту і взаємодії між вузлами системи.

3. Виконання експерименту (Lab):
 - a. Лабораторне обладнання (Lab) проводить експеримент (D) та генерує первинні експериментальні дані.
 - b. Завдання (D) – це безпосередній процес отримання експериментальних даних, які можуть включати фізичні вимірювання, електронні сигнали чи інші результати дослідження.

4. Передача експериментальних даних Lab→ Edge Node 1
 - a. Після завершення експерименту лабораторне обладнання надсилає отримані результати назад до Edge Node 1.
 - b. Edge Node 1 приймає ці результати та переходить до їх опрацювання.
5. Попереднє опрацювання даних та розподіл між вузлами (Edge Node 1→ Edge Node 2)
 - a. Edge Node 1 виконує завдання (C) – розподіляє отримані дані між різними вузлами опрацювання.
 - b. Частина даних опрацьовується локально на Edge Node 1, а інша частина передається на Edge Node 2 для подальшого опрацювання.
6. Опрацювання даних на додатковому вузлі (Edge Node 2):
 - a. Edge Node 2 отримує частину первинних (сирих) даних від Edge Node 1.
 - b. Виконується завдання (E) – попередня обробка отриманих експериментальних даних.
7. Передача опрацьованих даних назад (Edge Node 2 → Edge Node 1)
 - a. Edge Node 2 завершує опрацювання та передає результати назад до Edge Node 1.
 - b. Edge Node 1 об'єднує всі попередньо опрацьовані дані.
8. Передача результатів у хмару (Edge Node 1 → Cloud)
 - a. Edge Node 1 передає попередньо опрацьовані результати назад у Хмару (Cloud).
 - b. Хмара отримує дані та завершує експеримент.

*Особливості організації бази даних для розподілених обчислень
з використанням гібридної Cloud Edge моделі.*

Що стосується бази даних, то (за плануванням на даний час) вона повинна забезпечувати зберігання первинних лабораторних даних з прив'язкою до:

- об'єкту досліджень (тип, № зразка, характеристики тощо);
- типу досліджень / визначеної лабораторної установки (якщо установок стане більше, ніж одна, а також з урахуванням можливості проведення різних досліджень на одній установці);
- параметрів лабораторного експерименту (режимів тестування і дослідження об'єктів, особливостей часового ходу досліджень тощо);
- типу і параметрів вимірювального каналу (каналів).
- часу і дати експерименту, ідентифікаційного номера експерименту.

Подібно буде організовано зберігання результатів опрацювання лабораторних даних, деталізація цих вимог буде розроблена пізніше на основі накопичених результатів тестових експериментів та інших напрацювань.

До бази даних додаються також апіорні дані про самі об'єкти досліджень. Наприклад, для досліджень наноплазмонних ефектів вхідними даними можуть бути наноструктурні параметри матеріалів (тип (матеріал) і форма наночастинок, їх концентрація і просторовий розподіл, агрега-тованість – димери, тримери тощо, фрактальність, кластеризація, ієрархія структур тощо), їх оптичні, діелектричні, теплові та інші характеристики (Bolesta et al., 2019a, 2019b).

Первинні дані та результати опрацювання і аналізу можуть бути у вигляді файлів (наприклад, протоколів експериментів), табличних і графічних даних – спектрограм, фотографій, у т.ч з електронного мікроскопа, відео тощо, коментарів (текстових, аудіо, відео) та ін.

Функціональні вимоги до СКБД такі (враховуючи наявні напрацювання (Bolesta et al., 2019a, 2019b, та ін.)):

1. Легка вага та низьке споживання ресурсів:
 - a. БД має працювати на пристроях із обмеженими ресурсами (мало оперативної пам'яті, повільний процесор).
 - b. Підтримка роботи без окремого серверного процесу (наприклад, SQLite).
2. Підтримка автономної роботи:
 - a. Можливість роботи без постійного підключення до мережі.

- b. Синхронізація з хмарою або іншими вузлами після відновлення з'єднання.
- 3. Висока швидкість опрацювання запитів:
 - a. Оптимізовані операції читання/запису, особливо для потокових даних та часових рядів.
 - b. Підтримка in-memory (наприклад, Redis) для швидкого опрацювання в реальному часі.
- 4. Надійність і цілісність даних:
 - a. Журналювання операцій (Write-Ahead Logging) для запобігання втраті даних.
 - b. Реплікація для збереження даних на кількох вузлах.
- 5. Масштабованість та підтримка розподіленої архітектури:
 - a. Важливо для IoT-інфраструктури та хмарних синхронізованих систем.
 - b. СКБД повинна підтримувати можливість роботи в кластері або розподіленому середовищі (наприклад, Apache Cassandra).
- 6. Підтримка різних моделей даних:
 - a. Реляційні (SQL) – коли важлива структурованість і транзакційність.
 - b. NoSQL (ключ-значення, документи) – для гнучких і високошвидкісних застосувань.
 - c. Часові ряди – для IoT-систем (наприклад, InfluxDB).

Нижче наведено перелік СКБД, які підходять для обчислень на периферії (Edge Computing). Вони легкі, ефективні та можуть працювати на пристроях з обмеженими ресурсами (Grzesik & Mrozek, 2020; Kuzev et al., 2020).

1. Реляційні СКБД (SQL-based):
 - a. SQLite – легка, автономна та широко використовується у вбудованих системах.
2. NoSQL(об'єктні) СКБД:
 - a. RocksDB – високопродуктивне сховище ключ-значення, оптимізоване для флеш-пам'яті.
 - b. Apache Cassandra (для розподілених рішень) – стійка до відмов, але вимагає більше ресурсів.
 - c. Redis – швидка, працює в оперативній пам'яті, чудово підходить для кешування та реального часу.
 - d. MongoDB Embedded – легка документно-орієнтована база даних для IoT та периферійних обчислень.
3. СКБД для часових рядів (для IoT та аналітики):
 - a. InfluxDB – оптимізована для опрацювання часових рядів, ідеальна для IoT.

На першому етапі реалізації вирішено використати реляційну СКБД SQLite, оскільки вона легка, займає мало пам'яті та ресурсів (наприклад, середня база до 100 МБ даних використовує 10–50 МБ RAM, 5–10% CPU), підтримує різні архітектури та ОС. В майбутньому для оптимізації можливо будуть використані інші не реляційні СКБД, наприклад RocksDB або InfluxDB, які краще підходять для великих обсягів даних.

Вибір протоколів інтернет-зв'язку

В загальному апаратно-програмна організація взаємодії лабораторної, клієнтської і дата-складових лабораторного комплексу повинна забезпечувати надійну їх взаємодію через інтернет в процесі проведення лабораторних експериментів і подальшої обробки даних. У випадку віддаленого керування ходом експерименту з клієнтського комп'ютера важливими може бути мінімізація затримки передачі даних і керуючих сигналів, це в першу чергу стосується швидкоплинних у часі лабораторних експериментів з динамічною зміною в часі параметрів тестувань і характеристик досліджуваних об'єктів. Для забезпечення цих вимог планується використати надійні та ефективні способи організації та протоколи інтернет-зв'язку. Основні критерії вибору:

- Мінімізація затримки – важливо для режиму реального часу.
- Ефективність використання мережевих ресурсів – впливає на продуктивність.
- Надійність та безпека – захист даних від втрат і стороннього доступу.

Проаналізуємо основні відомі варіанти розглянуті в цих публікаціях (Alsurdeh, 2020; Niswar et al., 2024; Skvorc et al., 2014):

1. gRPC – сучасний високопродуктивний RPC-фреймворк від Google, що використовує Protocol Buffers для серіалізації та HTTP/2 для транспорту:
 - a. Висока швидкість та ефективність за рахунок бінарного формату;
 - b. Підтримка двонаправленого потокового зв'язку;
 - c. Низька затримка, що може бути критично для керування експериментами;
 - d. Генерація клієнтського коду для різних мов програмування.
2. REST – архітектурний стиль, який використовує HTTP запити:
 - a. Простота реалізації та використання;
 - b. Широка підтримка в різних системах;
 - c. Висока масштабованість;
 - d. Кешування відповідей.
 - e. Недоліки: відсутність нативної підтримки потокової передачі даних.
3. SOAP – протокол обміну структурованими повідомленнями на основі XML:
 - a. Високий рівень стандартизації;
 - b. Вбудовані механізми безпеки та обробки помилок.
 - c. Недоліки: надлишковість формату XML знижує швидкість передачі даних.
4. MQTT/CoAP – протоколи для IoT та обмежених пристроїв:
 - a. MQTT – легкий протокол публікації/підписки, оптимізований для ненадійних мереж;
 - b. CoAP – протокол для обмежених пристроїв, схожий на HTTP, але оптимізований для M2M комунікацій.
 - c. Обидва протоколи мають низькі накладні витрати і енергоспоживання.
5. WebSockets – протокол повнодуплексного зв'язку через TCP-з'єднання:
 - a. Постійне двонаправлене з'єднання;
 - b. Низька затримка для передачі даних в реальному часі;
 - c. Сумісність з існуючою веб-інфраструктурою.

Для лабораторного комплексу, враховуючи необхідність мінімізації затримок та підтримки двонаправленого зв'язку, пропонується використовувати gRPC для взаємодії між хмарною частиною та граничними вузлами, що забезпечить високу швидкість передачі даних та підтримку потокового зв'язку. Для веб-інтерфейсу клієнтської частини доцільно використовувати WebSockets для відображення даних у реальному масштабі часу. Для зв'язку з обмеженими пристроями лабораторії (сенсори, контролери) можна застосувати MQTT, який забезпечує надійну комунікацію навіть при нестабільному з'єднанні. Таким чином, гібридне використання різних протоколів дозволить оптимізувати комунікацію між різними складовими лабораторного комплексу відповідно до їх специфічних потреб та обмежень.

Забезпечення надійності роботи та кібербезпеки

Вимоги до надійності функціонування лабораторного комплексу передбачають як організацію чіткого адміністрування його роботи з уникненням різного роду конфліктів при взаємодії між окремими його частинами, так і стійкість комплексу до різних позаштатних ситуацій – збоїв у роботі окремих його частин, помилок операторів та клієнтів, перебоїв електроживлення та інтернет-зв'язку тощо, такі питання є актуальними для подібних систем (Chushchak et al., 2023; Faizulin & Nazarkevych, 2024; Nazarevych & Nazarevych, 2011).

Для забезпечення надійності системи в умовах розподілених (граничних) обчислень необхідно реалізувати багаторівневу систему стійкості до відмов. По-перше, на кожному з граничних вузлів (Edge Nodes) має бути імплементована система автономного відновлення з можливістю перезавантаження обладнання та програмного забезпечення. Для цього доцільно використовувати технології контейнеризації (Docker), які дозволяють швидко розгортати та відновлювати сервіси (Javed et al., 2020).

Для запобігання втрат даних при збоях передачі між вузлами системи, необхідно впровадити механізми буферизації та тимчасового зберігання даних на граничних вузлах з подальшою синхронізацією при відновленні зв'язку. Подібний підхід реалізований у системах обміну повідомленнями, таких як Apache Kafka або RabbitMQ, де гарантується доставка повідомлень навіть при тимчасовій недоступності отримувача (Javed et al., 2020).

Важливим аспектом надійності є також автоматичний моніторинг стану всіх компонентів системи з використанням таких інструментів як Prometheus та Grafana для збору метрик та візуалізації стану системи. Доцільно налаштувати систему сповіщень, яка повідомлятиме адміністраторів про потенційні проблеми до того, як вони призведуть до відмови системи.

Також необхідно передбачити певні заходи щодо кібербезпеки, зокрема, захист від несанкціонованих втручань у роботу комплексу (Chushchak et al., 2023; Javed et al., 2020). Одним із способів підвищення кібербезпеки при роботі комплексу є чітке керування рівнем доступу і контроль за ним, ведення відповідних протоколів роботи комплексу, це одночасно допоможе підвищити ефективність його використання.

У контексті розподілених обчислень особливо важливим є захист комунікації між вузлами системи. Для всіх з'єднань передбачається шифрування за допомогою TLS/SSL з використанням сучасних криптографічних протоколів. Для аутентифікації та авторизації планується використовувати системи управління ідентифікацією (IAM) з підтримкою багатофакторної автентифікації. В граничних вузлах (Edge Nodes) планується впровадити систему логування безпеки з централізованим збором та аналізом логів, що дозволить виявляти потенційні загрози на ранніх стадіях.

Особливої уваги потребує захист передачі керуючих команд до лабораторного обладнання. Планується впровадити систему валідації команд для запобігання виконанню потенційно небезпечних операцій, а також реалізувати механізми фізичного обмеження можливостей обладнання (наприклад, апаратні обмежувачі параметрів) для запобігання пошкодженню обладнання навіть у випадку успішної хакерської атаки (Javed et al., 2020).

Для захисту від атак типу "людина посередині" при передачі даних між компонентами системи планується використовувати механізми взаємної автентифікації та цифрові сертифікати з валідацією через довірені центри сертифікації. Для додаткового захисту можливе використання VPN-тунелів між віддаленими компонентами системи.

Одним з варіантів вирішення питань надійності роботи лабораторного комплексу та його кібербезпеки може бути (за необхідності) організація роботи комплексу як локальної автономної відокремленої системи з забезпеченням взаємодії між лабораторною установкою, локальними клієнтськими робочими місцями і локальною базою даних (вузлом Cloudlet) через локальну мережу і з живленням від автономної системи, наприклад, такої – автономний генератор / система: сонячні батареї – конвертери – акумулятори.

Питання опрацювання даних лабораторних експериментів.

Для опрацювання даних лабораторних експериментів передбачено використовувати як відомі готові програмні пакети відповідного функціоналу, так і спеціальні програмні засоби. Серед відомих можна назвати програму *Excel* (у першу чергу для опрацювання табличних даних), програму *PowerGraph* (зокрема, для реєстрації, опрацювання та аналізу часових рядів), програмні середовища *Matlab*, *COMSOL* та ін. Серед спеціально створених програм одними з найбільш швидких та ефективних рішень можуть бути конвертери файлів, вони дають можливість перетворити файли даних з формату з виходу лабораторної установки у формат, що є вхідним для наявних програм опрацювання, забезпечуючи таким чином спряження лабораторної та клієнтської частин лабораторного комплексу на рівні форматів даних і ефективність використання відомих (часто безкоштовних) програм опрацювання даних. Автори мають досвід розробки таких конверторів даних (Morozov et al., 2014), такий підхід на практиці демонструє свою ефективність.

До складу лабораторного комплексу залучено потужний модуль математичного моделювання досліджуваних фізичних ефектів, він кардинально підвищує результуючу ефективність дослідницької роботи комплексу (Bolesta et al., 2019a, 2019b). Так, для досліджень з наноплазмоніки моделюються нанокompозити та їх оптичні реакції (Demchuk et al., 2017). Лабораторна установка дає оптичні відгуки реальних нанокompозитів. Змодельовані і фактичні спектри порівнюються між собою, і розв'язується обернена задача – знаходження параметрів кластерів наночастинок (зокрема, розмірів самих наночастинок і їх кластерів), що відповідають експериментальним спектрам нанокompозитів. Таким чином отримується важлива інформація про базові та функціональні характеристики нанокompозитів без складних електронномікроскопічних та інших досліджень. Зокрема, у статтях (Bolesta et al., 2019a, 2019b) автори продемонстрували розрахунки спектрів поглинання для таких цілей методом дискретних диполів. Цей метод, а також методи різницевого часових доменів, апроксимації теорії Мі, квазістатичного наближення реалізовано з використанням мікросервісів та хмарних обчислень, їх інтегровано до загального модуля опрацювання результатів експериментів у варіанті локальної реалізації лабораторного комплексу. Очевидно, що вони без проблем будуть функціонувати і в інтернет-доступному варіанті такого комплексу.

Висновки

На основі проведеного аналізу наукових публікацій і власних напрацювань розроблено структуру і загальні алгоритми роботи програмно-апаратного комплексу автоматизованої інтернет-доступної лабораторії напівпровідників, у т.ч. для досліджень у галузі фотоніки і наноплазмоніки. Пропрацьовано питання взаємодії функціональних частин комплексу і сформульовано відповідні вимоги до програмного забезпечення та загальні принципи його функціонування. Також окреслено вимоги до організації бази даних та проаналізовано відомі способи їх забезпечення. Проаналізовано проблематику реалізації інтернет-доступності комплексу, у т.ч. дистанційного задіяння спеціально розроблених засобів комп'ютерного моделювання фотонних і наноплазмонних ефектів, протоколи зв'язку, питання опрацювання даних, надійності роботи та кібербезпеки, на основі проведеного аналізу запропоновано оптимізовані підходи до реалізації цих задач.

Розроблено загальний алгоритм функціонування програмно-апаратного комплексу автоматизованої інтернет-доступної лабораторії напівпровідників. На першому етапі реалізації комплексу серед проаналізованих баз даних для використання вибрано реляційну базу даних SQLite.

Щодо протоколів зв'язку, то планується випробувати і використовувати різні – оптимальні для конкретних задач і функцій. Зокрема, gRPC для взаємодії між хмарною частиною та граничними вузлами, WebSockets для відображення даних у реальному часі, для зв'язку з лабораторною установкою – MQTT, який забезпечує надійну комунікацію навіть при нестабільному з'єднанні.

Для забезпечення надійності роботи і кібербезпеки системи передбачено використання відомих перевірених рішень – технології контейнеризації, шифрування за допомогою TLS/SSL, системи управління ідентифікацією (IAM) з підтримкою багатофакторної автентифікації та ін.

Завдяки достатній універсальності, можливостям нарощування і реконфігурації створюване програмне забезпечення можна буде ефективно використовувати як при подальшому розвитку програмно-апаратного комплексу автоматизованої інтернет-доступної лабораторії напівпровідників, так і для інших подібних лабораторій (за відповідної його адаптації), а також для вирішення подібного кола задач в інших областях застосування.

СПИСОК ЛІТЕРАТУРИ

1. Alsurdeh, R. (2020). Resource provisioning and scheduling algorithms for hybrid workflows in edge cloud computing (Doctoral thesis). Western Sydney University. <http://hdl.handle.net/1959.7/uws:61320>.
2. Bolesta, I., Kushnir, O., Bavdys, M., Khvyshchun, I., & Demchuk, A. (2019a). Computational-Measurement System "Nanoplasmonics". Part 1: Architecture. In Proceedings of the 2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT) (pp. 51–54). IEEE. <https://doi.org/10.1109/ELIT.2019.8892288>.

3. Bolesta, I., Kushnir, O., Bavdys, M., Khvyshchun, I., & Demchuk, A. (2019b). Computational-Measurement System "Nanoplasmonics". Part 2: Structure of Microservices. In Proceedings of the 2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT) (pp. 55–58). IEEE. <https://doi.org/10.1109/ELIT.2019.8892345>.
4. Chamas, C. L., Cordeiro, D., & Eler, M. M. (2017). Comparing REST, SOAP, Socket, and gRPC in computation offloading of mobile applications: An energy cost analysis. In Proceedings of the 2017 IEEE 9th Latin-American Conference on Communications (LATINCOM) (pp. 1–6). IEEE. <https://doi.org/10.1109/LATINCOM.2017.8240185>.
5. Chushchak, I., Veres, O., Chyrun, L., Kolyasa, L., & Vlasenko, O. (2023). Information System for the Educational Center. Journal of Lviv Polytechnic National University. Information Systems and Networks, 14, 23–45. <https://doi.org/10.23939/sisn2023.14.023>.
6. Demchuk, A., Bolesta, I., Kushnir, O., Kolych, I. (2017). The Computational Studies of Plasmon Interaction. Nanoscale Research Letters, vol. 12:273, (7 pp). <https://doi.org/10.1186/s11671-017-2050-8>.
7. Faizulin, O., & Nazarkevych, M. (2024). Methods and Means of Analyzing Application Security via Distributed Tracing. Journal of Lviv Polytechnic National University. Information Systems and Networks, 16, 69–87. <https://doi.org/10.23939/sisn2024.16.069>.
8. Grzesik, P., & Mrozek, D. (2020). Comparative Analysis of Time Series Databases in the Context of Edge Computing for Low Power Sensor Networks. Lecture Notes in Computer Science, 371–383. https://doi.org/10.1007/978-3-030-50426-7_28.
9. Holubinka, V., & Khudyi A. (2024). Enhancing Database Query Performance: Analysis of Indexing Techniques. Journal of Lviv Polytechnic National University. Information Systems and Networks, 15, 65–73 (in Ukrainian). <https://doi.org/10.23939/sisn2024.15.065>.
10. Ivantyshyn, D., & Burov, Ye. (2023). Prototype of Intellectual System for Research of Space Weather Parameters. Journal of Lviv Polytechnic National University. Information Systems and Networks, 14, 348–356 (in Ukrainian). <https://doi.org/10.23939/sisn2023.14.348>.
11. Javed, A., Heljanko, K., Buda, A., & Främling, K. (2020). CEFIoT: A fault-tolerant IoT architecture for edge and cloud. arXiv preprint arXiv:2001.08433. <https://doi.org/10.48550/arXiv.2001.08433>.
12. Kuzev, M., Kostal, P., & Urban, L. (2020). Performance of databases in IoT applications. In SAMI 2020 - IEEE 18th World Symposium on Applied Machine Intelligence and Informatics, Proceedings (pp. 127-130). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00041>.
13. Laboratory Software for the Healthcare Industry (2025) [Online]. Available at: <https://www.labware.com/>.
14. Lumerical, Vancouver, Canada. Lumerical: High-Performance Photonic Simulation Software. (2025). [Online]. Available at: <https://www.lumerical.com/products/fdtd/>.
15. Morozov, Y., Nazarevych, R., Nazarevych, A., Struk, Y., & Markhyvka, V. (2014). Software of internet portal of geophysical monitoring. Journal of the National University "Lviv Polytechnic" "Computer Science and Information Technologies", 800, 228–238.
16. Mytsyk, B. H., Nazarevych, A. V., Bashtevych, M. V., & Nazarevych, R. A. (2011). Contactless capacitive displacement sensors in deformation geophysical studies. Selection and Processing of Information, 35(111), 69–76 (in Ukrainian).
17. Nazarevych, A. V., & Nazarevych, R. A. (2011). Prototype of internet-accessible system of geodynamic monitoring data gathering and processing. Geodynamics, 11(2), 216–218 (in Ukrainian). <https://doi.org/10.23939/jgd2011.02.216>.
18. Nazarevych, R., Markhyvka, V., Struk, Y., & Nazarevych, A. (2011). Conversion and preprocessing of data from deformation monitoring. Journal of Lviv Polytechnic National University "Computer Science and Information Technologies", 694, 334–340 (in Ukrainian).
19. Newman, S., Building Microservices: Designing Fine-Grained Systems, Sebastopol: O'Reilly Media, 2015.
20. Niswar, M., Safruddin, R. A., Bustamin, A., & Aswad, I. (2024). Performance evaluation of microservices communication with REST, GraphQL, and gRPC. International Journal of Electronics and Telecommunications, 70(2), 429–436. <https://doi.org/10.24425/ijet.2024.149562>.
21. Savkiv, L. H. (2019). Information technology for express analysis of data of geoelectromagnetic observations. (Candidate's (PhD) thesis). Lviv Polytechnic National University, Lviv, Ukraine (in Ukrainian).
22. Skvorc, D., Horvat, M., & Srbljic, S. (2014). Performance evaluation of WebSocket protocol for implementation of full-duplex web streams. Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1003–1008. <https://doi.org/10.1109/MIPRO.2014.6859715>.

**SOFTWARE OF INTERNET-ACCESSIBLE
SEMICONDUCTOR LABORATORY**

Roman Nazarevych¹, Ivan Bolesta²

^{1,2}Ivan Franko National University of Lviv,

Electronics and Computer Technologies Department, Lviv, Ukraine

¹email: roman.nazarevych@gmail.com, ORCID: 0009-0003-0104-3280

²email: ivan.bolesta@lnu.edu.ua, ORCID: 0000-0002-7580-3495

© Nazarevych.R., Bolesta I., 2025

The paper presents approaches for creating an automated internet-accessible semiconductor laboratory, specifically the control software system that ensures the internet accessibility of the laboratory. The functionality and structure of the software-hardware complex, developed with consideration of known solutions, are described, as well as its implementation options using the internet, cloud, and edge computing. Local implementation options with enhanced resilience to force majeure factors and cyber threats are also considered. Optimal solutions for databases and communication protocols are analyzed and proposed, with solutions selected to meet the requirements of a distributed system with limited resources at edge nodes, specifically limitations in memory and computational power. The proposed relational database SQLite has minimal resource requirements while providing most of the capabilities of relational databases, and the proposed gRPC protocol is one of the fastest and most efficient in utilizing network resources. The main approaches to implementing edge computing (Cloudlet, MEC, Fog Computing) are reviewed, and the optimal option for the internet-accessible laboratory is selected. The overall structure and organization of the basic laboratory, client, and data components of the software-hardware complex are presented. A high-level algorithm for the interaction of the main system nodes and the distribution of computational tasks between them is described. This algorithm, leveraging the advantages of hybrid (cloud-edge) computing, accelerates the processing of experiment results, reduces the load on the internet connection channel, and lowers the computational load on the cloud node. Additionally, this algorithm supports operation in an offline mode, allowing the system to function fully without an internet connection, without limiting its functionality.

Keywords – internet-accessible laboratory, software-hardware complex, edge computing, cloud computing, distributed computing, heterogeneous computing, internet communication protocols.