УДК 004.65
DOI https://doi.org/10.32838/2663-5941/2021.5/23

Піткевич П.І.
Білоруський державний університет інформатики та радіоелектроніки

МЕТОДИКА СТВОРЕННЯ РОЗПОДІЛЕНОГО СХОВИЩА ДАНИХ У БАНКІВСЬКІЙ СФЕРІ У РЕАЛЬНОМУ ЧАСІ

У статті розкрито методику створення розподіленого сховища даних в банківській сфері у реальному часі. Визначено роль сховища даних сучасних банківських організаціях, описані ключові завдання, які вирішуються за допомогою розподіленого сховища даних, а також окреслено проблеми проектування систем такого класу. Для вирішення даних проблем запропонована методика створення розподіленого сховища даних в банківській сфері у реальному часі. У даному дослідженні описано структуру розробленої автором універсальної багатокомпонентної моделі розподіленого сховища даних в банківській сфері у реальному часі, а також представлений базовий набір компонент сховища і бізнес-об'єктів. Цей базовий набір може бути використаний як основа або шаблон, який може бути налаштований під потреби конкретної банківської організації. Підкреслено, що ключовими завданнями, які необхідно вирішити в процесі розвитку даного дослідження, є: формалізація обов'язкових і опціональних атрибутів для кожного типу сутності; формалізація типів атрибутів і доменів даних; збагачення базового набору компонент і бізнес-об'єктів; формалізація правил нейминга об'єктів універсальної багатокомпонентної моделі розподіленого сховища даних в банківській сфері у реальному часі; апробація запропонованої методики і універсальної моделі на прикладі вирішення конкретного бізнес-завдання в банківській сфері. Архітектура розгортання розподіленого сховища даних в банківській сфері у реальному часі базується на встановленні зв'язку між чотирма блоками управління. Розподілене сховище даних в банківській сфері у реальному часі складається із двох частин: бази даних інформаційної точки та бази значень. Схематично запропоновано потік даних доступу до даних та потік даних із запитами даних. Наголошено, що дані розподіляються на різні вузли відповідно до правил розподілу, а правила розподілу включають власні хеш-функції та таблиці відображення фрагментації. Зазначається, що для забезпечення високої доступності вузла управління зазвичай потрібно налаштувати кілька вузлів, бо він в основному зберігає ключові налаштування, такі як правила розповсюдження та невелику детальну інформацію, таку як стан кожного активного та резервного вузла, підходить для реалізації на основі подібних архітектур, забезпечуючи впорядковану та послідовну інформацію.

Ключові слова: *база даних, сховище, розподілена система, банківська сфера, реальний час, транзакція, централізоване розгортання, обчислення.*

Постановка проблеми. Технологія баз даних зародилася в кінці 20 століття. Її теорія та технологічний розвиток надзвичайно швидкі, а її застосування набуває все більшого поширення. Важливою галуззю технології баз даних є база даних реального часу. Така база даних побудована за моделями даних у реальному часі. Технологія бази даних у реальному часі є продуктом поєднання системи реального часу та технології бази даних [1].

Спеціалізація та виробництво баз даних у режимі реального часу у сфері передачі даних, зберігання даних, пошуку даних, доступу до даних, обробки даних та відображення даних забезпечують зручну та стабільну підтримку даних для побудови аналітичних додатків на основі історичних даних великої ємності в реальному часі. Це дозволяє прикладній системі повною мірою використовувати цінні історичні дані реального часу з більш високого та глибокого рівня. Зі стрімким розвитком мережевих технологій кількість даних, що генеруються системами реального часу, зростає в геометричній прогресії, а бізнес-додатки висувають все більш високі вимоги до управління даними в режимі реального часу та характеру програм у реальному часі. Наявна автономна технологія бази даних у реальному часі має великі обмеження як у теорії, так і в практичному застосуванні, і більше не може повністю задовольняти поточні потреби.

У даний час використання бази даних у реальному часі у банківській сфері ще базується на централізованому розгортанні. Упровадження технологій розподілених обчислень та технологій зберігання даних цілком може вирішити ці вузькі місця, з якими стикаються автономні бази даних реального часу. Однак питання методики створення розподіленого сховища даних у реальному часі для задоволення своєчасності обробки заявки

на транзакцію та вимог до пропускної спроможності транзакцій є складним моментом.

Аналіз останніх досліджень і публікацій. Принципи задоволення потреб зі збереження та передачі даних у банківській сфері протягом багатьох років вивчало чимало вчених. Автори підходили як до питання трансформації цифрової сфери у банківську, так і до механізмів розгортання хмарних середовищ для забезпечення діяльності банківських установ.

Цифрова трансформація вітчизняного банківського середовища в умовах розвитку фінтех-екосистеми розкрита А.І. Гулей та С.А. Гулей [2]. Прикладні аспекти поширення хмарних сховищ даних описані у [3].

У дисертації [4] проведено аналіз можливості впровадження хмарних технологій для забезпечення діяльності банківських установ та підтримки функціонування бізнес-процесів. Розглянуто проблеми та переваги хмарних технологій на різних рівнях архітектурного ландшафту банку з урахуванням специфіки нормативно-правового регулювання діяльності фінансової установи. Автором реалізовано підвищення ефективності обробки інформації регламенту операційного дня банку шляхом модернізації інформаційної архітектури банку на основі впровадження хмарних технологій.

І.І. Бородій, Я.С. Парамуд та В.В. Сав'як [5] розглянули принципи побудови програмної системи формування агрегованих даних, а також основні принципи побудови програмних систем формування агрегованих даних. Науковцями проведено їхній порівняльний аналіз, запропоновано альтернативний принцип побудови програмної системи. За цим принципом побудови можна усунути проблеми швидкої та надійної обробки даних, масштабування, автоматизації роботи складових частин програмної системи, якості та безпеки даних.

Із зарубіжних авторів варто відзначити таких, як: Apostu A., Rednic E., Puican F. [6], Martins, Pedro & Sá, Filipe & Caldeira, Filipe & Abbasi, Maryam [7], Zissis D., Lekkas D. [8], Barkhordari, Mohammadhossein & Niamanesh, Mahdi [9], Noaman, Amin & Yousef, Amin [10], Ambodo B. S., Suryanto R., Sofyani H. [11], Yang, Weiwen & Qu, Yanzhen [12], Billel, ARRES & Nadia, Kabachi & Boussaid, Omar [13], Krishnaveni S. & Hemalatha, M. [14] та ін.

Проте, враховуючи описані наукові набутки за темою, питання розкриття методики створення розподіленого сховища даних у банківській сфері у реальному часі залишається відкритим та потребує детального опрацювання.

Постановка завдання. Розкрити методику створення розподіленого сховища даних у банківській сфері у реальному часі.

Викладення основного матеріалу дослідження. Розподілена архітектура розгортання сховища даних в банківській сфері у реальному часі показана на рисунку 1.

В архітектурі розгортання вузли управління є адміністратором усього розподіленого сховища даних в банківській сфері у реальному часі і в основному зберігають інформацію про метадані системи, включаючи ключову інформацію, таку як режим розповсюдження даних, стан кожного вузла та стан узгодженості активних та резервних вузлів.

Рис. 1. Архітектура розгортання розподіленого сховища даних в банківській сфері у реальному часі

Розподілене сховище даних в банківській сфері у реальному часі складається із двох частин: бази даних інформаційної точки та бази значень, як показано на рисунку 2.

Вузли планування належать до рівня розподіленого доступу. Уніфікований інтерфейс дозволяє програмам отримувати доступ до розподіленого сховища даних в банківській сфері у реальному часі як цілісної логічної сутності. Крім того, вузол планування також належить до розподіленого рівня розташування, який є розподільником та збирачем даних. В основному він відповідає за розподіл даних, збір результатів запитів та планування завдань. Коли розподілений вузол здійснює запити та отримує доступ до даних на декількох вузлах, обробка одночасного доступу дозволяє паралельно обробляти кілька запитів даних на декількох вузлах зберігання, тим самим забезпечуючи ефективний розподілений доступ до даних.

Вузли даних належать до розподіленого рівня зберігання. Кожен вузол даних запускається та керує екземпляром бази даних. Вузол даних відповідає за фактичне зберігання всіх системних даних бази даних, отримує дані від вузла планування, виконує розкладене завдання запиту, а результат виконання повертається до прикладної програми через вузол планування. Кількість вузлів даних обмежена лише важкими умовами, такими як пропускна здатність Інтернету та фізичні умови приміщення обладнання. Кожен вузол даних зберігає лише дані, що належать до відповідного розділу, і логічно еквівалентні. Активні та резервні вузли даних реалізують надмірність даних між вузлами.

Інформаційна база точок тегів містить таблицю основної інформації точок обстеження з тегом точки (назва_точки), що містить основну інформацію про конфігурацію точки тегу, наприклад опис точки тегу, алгоритм стиснення. Користувач може запитувати основну інформацію точки тегу з цієї інформаційної бази даних тегів. База даних цінностей містить кеш цінностей у реальному часі та сховище історичних даних. Кожен запис відображає позначку часу, значення та якість даних реального часу, які генеруються точкою тегу. Користувачі можуть запитувати значення даних у реальному часі з бази даних цінностей. Тому два основні виміри розподіленого сховища даних в банківській сфері у реальному часі – це точки тегів і час даних. Якщо потрібно розподілити всі дані розподіленого сховища даних у реальному часі на декілька вузлів, необхідно почати з цих двох вимірів.

Зберігання метаданих: Таблиця точок тегів зберігається як таблиця метаданих на вузлі планування, і кожен вузол планування містить повну інформацію про точку тегу.

Резервне копіювання декількох вузлів планування. Вузол управління контролює стан вузлів планування та використовує потік синхронізації для виконання відновлення.

Потік даних доступу до даних показаний на рисунку 3. Дані надсилаються з сервера додатків або клієнта на вузол планування. Вузол планування надсилає дані до різних вузлів основних даних відповідно до правил розподілу даних, а вузол основних даних пересилає дані до резервного вузла під час процесу зберігання.

Інформація про тег точки

Тег точки Ім'я 1	Тег точки Опис 1	Тег точки ID 1	Алгоритм стиснення 1	…
Тег точки Ім'я 2	Тег точки Опис 2	Тег точки ID 2	Алгоритм стиснення 2	…
…				
Тег точки Ім'я n	Тег точки Опис n	Тег точки ID n	Алгоритм стиснення n	…

Значення бази даних

Тег точки Ім'я 1	Мітки часу	Значення	Статус	Мітки часу	Значення	Статус	…
Тег точки Ім'я 2	Мітки часу	Значення	Статус	Мітки часу	Значення	Статус	…
…							
Тег точки Ім'я n	Мітки часу	Значення	Статус	Мітки часу	Значення	Статус	…

Рис. 2. Таблиця точок міток і таблиця значень даних

Потік даних виглядає так, як показано на рисунку 4. Запит та умови запиту надсилаються від сервера додатків або клієнта до вузла планування. Вузол планування використовує моменти тегів і діапазони часу, які беруть участь в умовах запиту. За допомогою правил розповсюдження фільтруються відповідні вузли даних, розділяються та реорганізовуються декілька підзапитів для розподілу на декілька вузлів даних, кілька вузлів даних обробляють запити паралельно, завершують результати та повертають результати до вузла планування. Вузол планування чекає всіх виділених підзадач. Після того, як запит поверне результат, він здійснить процес агрегації та надішле результат на сервер додатків або клієнта.

Це передбачає виділення первинних і вторинних вузлів. Кожен підзапит може бути надісланий лише до вузла даних або резервного вузла, за винятком випадків, коли запит буде продовжувати надсилатися з вузла планування на інший вузол для виконання запиту (одночасно повідомляти про стан вузла управлінському вузлу). Вузол планування може вибрати вузол розповсюдження, визначаючи зайнятість активного та резервного вузлів, а вузол даних періодично повідомляє про поточний активний стан та рівень зайнятості вузла управління. Вузол планування також періодично синхронізує

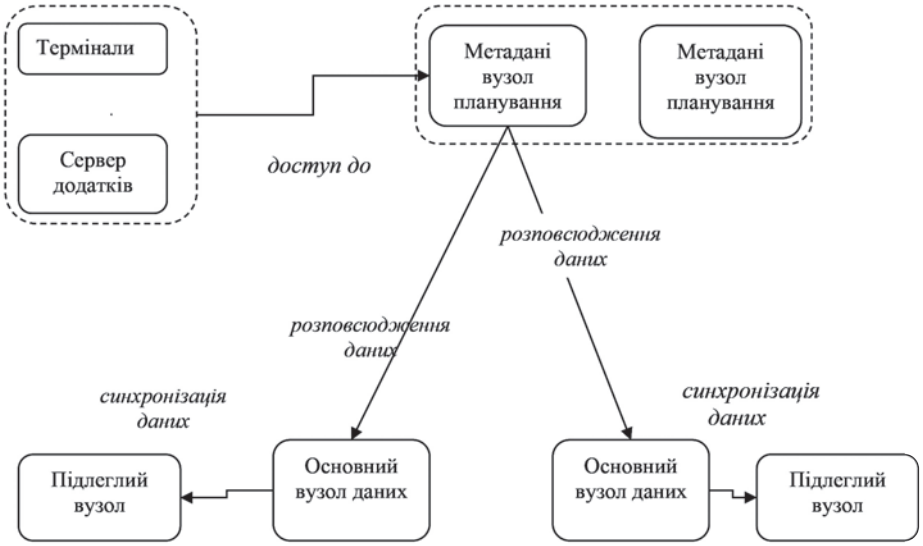

Рис. 3. Потік даних доступу до даних

Рис. 4. Потік даних із запитами даних

всі статуси вузла даних з вузла управління. Показник зайнятості можна зручно виміряти на основі середнього завантаження процесора, середнього використання мережі, поточного використання диска та поточного використання пам'яті.

При розробці правил розповсюдження необхідно враховувати актуальність точки тегу даних (якщо є можливість розмістити частину операцій із запитами безпосередньо всередині вузла даних, це, очевидно, важливіше, ніж концентрувати дані обчислення після передачі даних у вузол планування), а також врахувати ефективність паралельної обробки розподіленої архітектури.

Дані розподіляються на різні вузли відповідно до правил розподілу. Правила розподілу включають власні хеш-функції та таблиці відображення фрагментації. Точки міток та дані можуть бути розподілені до хеш-функції та таблиці відображення фрагментації.

Якщо правило розповсюдження не встановлено заздалегідь до запуску всього розподіленого сховища даних в банківській сфері у реальному часі, таблиця відображення фрагментації автоматично формується відповідно до кількості вузлів даних після запуску. Перед доступом до точок тегів і даних параметри, пов'язані з хеш-функцією, ще потрібно встановити через клієнт управління. Правила розповсюдження безпосередньо передаються на вузол управління для зберігання. Вузлу планування потрібно отримати правило розповсюдження від вузла керування перед розповсюдженням точки мітки та даних. Після того, як вузол планування вперше отримає правило розподілу, воно буде збережено в пам'яті. Не потрібно повторно отримувати правило розподілу кожного разу, коли дані розподіляються. Якщо вузол даних змінюється під час розширення або скорочення системи, правило розповсюдження потрібно змінити, вузол планування потрібно перезапустити.

Оскільки загальна кількість даних змінюється, може виникнути необхідність розглянути питання про розширення або скорочення вузлів даних. Зміна вузлів даних повинна враховувати вирівнювання та перерозподіл даних.

Система реального часу використовує таблицю відображення фрагментації та спосіб розподілу хеш-розподілу у два шари. Стратегію перерозподілу можна розробити так, що хеш -функція не змінюється, а лише змінює таблицю відображення фрагментації. Наприклад, нова таблиця відображення автоматично генерується відповідно до нової кількості вузлів, і вузол планування порівнює різницю між старою та новою таблицями відображення фрагментації, щоб встановити відповідний шлях руху фрагментів кожного вузла даних і розподіляє його для кожного даного вузол.

Більш ефективний підхід полягає у рівномірному розподілі даних по всій системі баз даних. Для забезпечення високої доступності вузла управління зазвичай потрібно налаштувати кілька вузлів. Він в основному зберігає ключові налаштування, такі як правила розповсюдження та невелику детальну інформацію, таку як стан кожного активного та резервного вузла, підходить для реалізації на основі подібних архітектур, забезпечуючи впорядковану та послідовну інформацію.

Крім того, синхронізувати стан вузла пам'яті з вузлом управління. Вузол планування в основному зберігає інформацію метаданих точок тегів. Додавання, видалення та модифікація точок міток виконується у вигляді транзакцій, і необхідно гарантувати атомність та сильну послідовність однієї операції. Тобто кожен функціонуючий вузол планування повинен забезпечувати успішну синхронізацію з усіма іншими нормальними вузлами планування. В іншому випадку потрібно або встановити відповідний статус для вузла, який не вдається синхронізувати, щоб була можливість відновити синхронізацію пізніше, або виконати операцію скасування на успішно виконаному вузлі. Крім того, наступні операції повинні чекати завершення всіх синхронних операцій попередньої операції. Інформація про метадані вузла планування поділена на блоки, і всі операції мають чітку послідовність, тому нормальний вузол повинен бути абсолютно однаковим для кожного блоку та послідовності блоків. Вузол управління регулярно виявляє ненормальні вузли планування та виконує синхронні операції відновлення.

Дзеркальний механізм копіювання та резервування даних використовується для забезпечення високої доступності кластера. Дизайн повинен відповідати режиму реального часу та відмовостійкості. Однак теорія CAP говорить, що консенсус, доступність та розподіл не можуть задовольнити обидва, а слабка узгодженість має свою сферу застосування, особливо у сценаріях, де потрібні високі вимоги реального часу та низькі вимоги щодо своєчасної узгодженості. Стан основних і дзеркальних вузлів даних контролюється вузлом управління. Вузол планування отримує статус активних та резервних вузлів даних через вузол управління, який може призначати вузли запису та вузли запиту. У той же час стан вузлів даних під час обробки завдань у реальному часі також вчасно надсилається до вузла управління.

Автоматична передача даних від головного вузла до дзеркального вузла не гарантує надійної узгодженості, що лише забезпечує остаточну узгодженість. Вузол планування відправляє завдання запиту відповідно до навантаження.

Щоб не впливати на роботу всієї системи в режимі реального часу, для відновлення винятків використовується режим онлайн-синхронізації. Тобто дані не впливають на нормальне читання та запис під час процесу синхронізації.

Записи модифікації, створені ведучим вузлом у процесі синхронізації даних, записуються у вузлі основних даних у режимі журналу. Після завершення синхронізації журнал та дані оновлення аналізуються. Зміна даних у процесі аналізу все ще додається до кінця журналу, доки не буде завершено весь аналіз журналу. Вузол управління встановлює статус активних та резервних вузлів даних на звичайний.

Висновки і перспективи подальших досліджень. У роботі розкрито методику створення розподіленого сховища даних в банківській сфері у реальному часі. Оскільки продуктивність, надійність, масштабованість та інші вимоги до розподіленого сховища даних у реальному часі стають все більш високими у банківській сфері, у цій статті пропонується методика створення розподіленого сховища даних у системах даних реального часу шляхом поєднання розподілених технологій та технологій баз даних реального часу. Одночасно наводиться набір рішень щодо надмірності даних, перерозподілу даних та узгодженості даних.

Перспективи подальших досліджень ґрунтуються на реалізації структурного підходу до створення розподіленого сховища даних в банківській сфері у реальному часі на базі реального підприємства.

Список літератури:

1. Проблеми безпеки універсальних платформ управління даними / С.О. Спасітєлєва, Ю.Д. Жданова, І.В. Чичкань. *Кібербезпека: освіта, наука, техніка*. 2019. 2(6). С. 122–133.
2. Гулей А.І., Гулей С.А. Цифрова трансформація вітчизняного банківського середовища в умовах розвитку фінтех-екосистеми. *Український журнал прикладної економіки*. 2019. Т. 4. № 1. С. 6–15.
3. Іванюк О. Прикладні аспекти поширення хмарних сховищ даних. *Фінансово-кредитна система України в умовах інтеграційних і глобалізацій них процесів*: збірник тез доповідей Всеукраїнської науково-практичної конференції студентів та аспірантів (28 квітня 2021 року, м. Черкаси) / ЧНІ Університету банківської справи. Черкаси, 2021. С. 13–16.
4. Баглай Р.О. Інформаційна архітектура банку на основі хмарних технологій : автореф. дис. ... канд. техн. наук : спец. 05.13.06 ; наук. керівник Роскладка А.А. ; Київ. нац. торговельно-екон. ун-т. Харків, 2019. 20 с.
5. Бородій І.І., Парамуд Я.С., Сав'як В.В. Принципи побудови програмної системи формування агрегованих даних. *Вісник Національного університету «Львівська політехніка». Серія: Комп'ютерні системи та мережі.* Львів : Видавництво Національного університету «Львівська політехніка», 2018. № 905. С. 25–32.
6. Apostu A., Rednic E., Puican F. Modeling Cloud Architecture in Banking Systems. *Procedia Economics and Finance*. 2012. Vol. 3. P. 543–548. doi: http://doi.org/10.1016/s2212-5671(12)00193-1.
7. Martins, Pedro & Sá, Filipe & Caldeira, Filipe & Abbasi, Maryam. Distributed Data Warehouse Resource Monitoring. 2021. 10.1007/978-3-030-68285-9_24.
8. Zissis D., Lekkas D. Addressing cloud computing security issues. *Future Generation Computer Systems*. 2012. Vol. 28, No. 3. P. 583–592. doi: http://doi.org/10.1016/j.future.2010.12.006.
9. Barkhordari, Mohammadhossein & Niamanesh, Mahdi. Hengam a MapReduce-Based Distributed Data Warehouse for Big Data: A MapReduce-Based Distributed Data Warehouse for Big Data. International Journal of Artificial Life Research. 2018. No. 8. P. 16-35. doi: 10.4018/IJALR.2018010102.
10. Noaman, Amin & Yousef, Amin. Distributed data warehouse architecture and design [microform]. 2021. URL : https://www.researchgate.net/publication/36220007_Distributed_data_warehouse_architecture_and_design_microform/citation/download (Last accessed: 17.03.2021).
11. Ambodo B. S., Suryanto R., Sofyani H. Testing of Technology Acceptance Model on Core Banking System: A Perspective on Mandatory Use. *Jurnal Dinamika Akuntansi.* 2018. Vol. 9, No. 1. P. 11–22. doi: http://doi.org/10.15294/jda.v9i1.12006.
12. Yang W., Qu Y. ETL Pipeline Resource Predictions in Distributed Data Warehouses. 2021. URL : https://www.researchgate.net/publication/228541235_ETL_Pipeline_Resource_Predictions_in_Distributed_Data_Warehouses/citation/download (Last accessed: 17.03.2021).
13. Arres B., Kabachi N., Boussaid O. A Data Pre-partitioning and Distribution Optimization Approach for Distributed Data Warehouses. 2015. URL : https://www.researchgate.net/publication/294736835_A_Data_Pre-partitioning_and_Distribution_Optimization_Approach_for_Distributed_Data_Warehouses/citation/download (Last accessed: 17.03.2021).
14. Krishnaveni S. & Hemalatha, M. Dependency-Based Query Scheduling in Distributed Data Warehouse Environment. 2013. doi: 10.1007/978-3-319-03844-5_50.

Pitkevich P.I. METHODS OF CREATING A DISTRIBUTED DATA STORAGE IN THE BANKING SPHERE IN REAL TIME

The article reveals the method of creating a distributed data warehouse in the banking sector in real time. The role of data storage of modern banking organizations is defined, the key tasks which are solved by means of the distributed data storage are described, and also problems of designing of systems of such class are outlined. To solve these problems, a method of creating a distributed data warehouse in the banking sector in real time is proposed. This study describes the structure developed by the author of a universal multicomponent model of distributed data storage in the banking sector in real time, as well as presents a basic set of storage components and business objects. This basic set can be used as a basis or template, which can be customized to the needs of a particular banking organization. It is emphasized that the key tasks that need to be addressed in the development of this study are: formalization of mandatory and optional attributes for each type of entity; formalization of attribute types and data domains; enrichment of the basic set of components and business objects; formalization of the rules of naming objects of the universal multicomponent model of distributed data storage in the banking sector in real time; testing of the proposed methodology and universal model on the example of solving a specific business problem in the banking sector. The real-time deployment architecture of distributed data warehousing in the banking sector is based on establishing a connection between the four control units. Distributed real-time data storage in the banking sector consists of two parts: an information point database and a value database. The data flow of data access and the data flow with data requests are schematically proposed. It is emphasized that the data is distributed to different nodes according to the distribution rules, and the distribution rules include their own hash functions and fragmentation display tables. It is noted that to ensure high availability of the control node, it is usually necessary to configure several nodes, because it mainly stores key settings, such as distribution rules and small details, such as the status of each active and backup node, suitable for implementation based on similar architectures. orderly and consistent information.

Key words: *database, storage, distributed system, banking, real-time, transaction, centralized deployment, computation.*