

МИРГОРОДСЬКИЙ Андрій

Вінницький національний технічний університет

<https://orcid.org/0009-0007-6764-0060>

e-mail: mirgorodskijav@gmail.com

РОМАНЮК Оксана

Вінницький національний технічний університет

<https://orcid.org/0000-0003-0235-8615>

e-mail: romaniukoksana@gmail.com

РОМАНЮК Олександр

Вінницький національний технічний університет

<https://orcid.org/0000-0002-2245-3364>

e-mail: rom8591@gmail.com

АНАЛІЗ ДИНАМІЧНИХ МОДЕЛЕЙ ЗАБЕЗПЕЧЕННЯ УЗГОДЖЕНОСТІ ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ КЕРУВАННЯ БАЗАМИ ДАНИХ

В роботі проаналізовано сучасні динамічні моделі забезпечення узгодженості даних у розподілених системах керування базами даних. Розглянуто архітектурні особливості та принципи роботи трьох динамічних моделей: контекстно-орієнтованої, що використовує політики узгодженості та дескриптори контексту; CAnDoR, яка застосовує адаптивне розміщення сегментів даних; та R-TBC/RTA з ієрархічною топологією та графовою кластеризацією. Розглянуто варіанти технічної реалізації моделей та їх інтеграції з існуючими розподіленими системами. Проведено порівняльний аналіз моделей за архітектурними, адаптивними та інтеграційними критеріями. Визначено переваги та недоліки кожної моделі, їх складність реалізації та сценарії застосування. Результати дослідження можуть бути використані для розробки нових динамічних моделей узгодженості даних.

Ключові слова: розподілені бази даних, узгодженість даних, динамічні моделі, системи управління базами даних, відмовостійкість, сегментація даних.

MYRHORODSKYI Andrii, ROMANIUK Oksana, ROMANIUK Oleksandr

Vinnitsia National Technical University

ANALYSIS OF DYNAMIC DATA CONSISTENCY MODELS IN DISTRIBUTED DATABASE MANAGEMENT SYSTEMS

The article is devoted to the analysis of modern dynamic data consistency models in distributed database management systems (DBMS). Traditional distributed DBMS used static consistency models that did not consider system state or data access patterns. However, growing data scales, transaction complexity, and performance requirements demand new approaches.

The research analyzes three main dynamic consistency models. The context-oriented model consists of consistency blocks, consistency policies, and context descriptors that dynamically determine consistency levels based on operation context and system state. The CAnDoR model uses data segmentation with continuous measurement of synchronization and response times to automatically distribute segments among nodes. The R-TBC/RTA model addresses network partitioning problems by forming hierarchical tree structures with different consistency guarantees for primary and secondary nodes. Various options for the practical implementation of mechanisms for individual models are also considered.

Comparative analysis was conducted based on architectural criteria (concept, consistency determination, data placement), adaptive criteria (load monitoring, time adaptation, fault tolerance), and integration possibilities. The comparison revealed that simpler models like context-oriented provide higher flexibility and easier integration but depend heavily on implementation, while complex models like R-TBC/RTA offer better fault tolerance guarantees but require more sophisticated implementation. The use of architecture based on more complex models together with separate mechanisms of simpler models is promising, as it may allow the creation of a more universal dynamic model of data consistency.

The results show that all models have effective consistency mechanisms for specific scenarios but differ significantly in implementation complexity. The analysis can be used for developing custom dynamic consistency models and advancing distributed systems research.

Keywords: distributed databases, data consistency, dynamic models, database management systems, fault tolerance, data segmentation.

Стаття надійшла до редакції / Received 11.07.2025

Прийнята до друку / Accepted 19.08.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

З появою та розвитком розподілених систем керування базами даних з'явилися також нові виклики, які неможливо зустріти в класичних монолітних базах даних. До них можна віднести забезпечення узгодженості даних та високої доступності, а також постійно зростаючі вимоги до швидкодії та продуктивності [1].

В минулому розподілені СКБД переважно використовували статичні моделі забезпечення узгодженості даних, що не враховували загальний стан системи або ж природу оброблюваних даних, як-от їх типові схеми доступу, а тому вони базуються на ряді загальних правил та механізмів для забезпечення певного рівня цілісності даних та вирішення конфліктів. Існуючі інформаційно-орієнтовані та клієнтоорієнтовані моделі забезпечення узгодженості покривають більшість класичних сценаріїв застосування для типових розподілених систем або систем керування базами даних. Проте зростання масштабів даних, ускладнення структур транзакцій та підвищення вимог до продуктивності вимагають нових підходів до вирішення цієї задачі [2].

Останні дослідження у сфері розподілених обчислень акцентують увагу на необхідності гнучких механізмів, здатних адаптувати рівень узгодженості в залежності від типу запиту, навантаження на систему або контексту виконання операцій [3]. Крім того, окремої уваги заслуговують динамічні та контекстно-залежні моделі узгодженості, які можуть комбінувати різні підходи в рамках єдиної архітектури та переключатись між ними в залежності від поточного сценарію використання.

Проте, більшість існуючих динамічних моделей використовують кардинально різні механізми роботи, що впливають на їх характеристики та можливість при визначенні необхідного режиму узгодженості. Низка таких аспектів залишається недостатньо дослідженою й існує потреба в систематизації підходів до формування контексту виконання запитів, визначення політик узгодженості на основі системних метрик, а також інтеграції таких моделей у реальні СКБД.

Метою даного дослідження є аналіз, порівняння та оцінка характеристик й ефективності сучасних динамічних моделей забезпечення узгодженості даних у розподілених системах керування базами даних.

Задачами дослідження є: аналіз архітектури та особливостей роботи існуючих динамічних моделей; визначення характеристик та порівняння розглянутих моделей, аналіз їх переваг, недоліків, сценаріїв застосування та можливих інтеграцій; визначення подальших напрямів дослідження та ключових характеристик і механізмів, що можуть допомогти в розробці перспективних моделей динамічного забезпечення узгодженості даних.

АНАЛІЗ ІСНУЮЧИХ ДИНАМІЧНИХ МОДЕЛЕЙ

Класичні моделі передбачають роботу всієї системи в одному режимі узгодженості – всі зміни даних обробляються та реплікуються згідно однакової логіки, що дає чіткі гарантії узгодженості для системи та її користувачів – прикладом є інформаційно-орієнтовані моделі узгодженості даних [4]. Проте, така розподілена система, наприклад СКБД, надає однаковий пріоритет для всіх вхідних запитів, чи то виконання комплексних фінансових транзакцій, чи то оновлення додаткової інформації користувача. Це може бути прийнятним варіантом для малонавантажених систем або систем, що орієнтовані на великі об'єми запитів вичитування даних. Проте, застосування динамічної логіки для одночасної підтримки різних режимів узгодженості для різних наборів даних/запитів може покращити швидкодію для високонавантажених систем або систем з обмеженими мережевими можливостями [5]. Наприклад, динамічна модель забезпечення узгодженості може застосовувати строгу узгодженість для високопріоритетних транзакцій та кінцеву узгодженість для решти запитів, що зменшить загальну кількість мережевого трафіку за одиницю часу між всіма вузлами та дозволить проводити реплікацію даних в більш гнучкому режимі.

Досягти динамічної узгодженості можна декількома шляхами. Одна з моделей пропонує ввести контекстно-орієнтовану систему, що складається з трьох компонентів: блоків узгодженості, політик узгодженості та дескрипторів контексту [6].

Блок узгодженості (англ. Consistent Bloc) – це група операцій, що мають однакові або схожі вимоги до узгодженості. Це можуть бути як незалежні операції, що просто виконуються в єдиний короткий проміжок часу, так і взаємопов'язані команди в складі єдиної транзакції, що не можуть бути розділені на різні блоки з різною узгодженістю.

Дескриптор контексту (англ. Context Descriptor) – це компонент, що збирає додаткові відомості про загальний стан системи та окремі параметри, що стосуються операцій в конкретному блоці узгодженості. Контекст може включати в себе середній час обробки запитів, швидкість синхронізації з іншими вузлами, наявні мережеві проблеми або їх відсутність, додаткові дані сесії клієнта тощо. Дескриптор надає необхідний контекст для подальшого прийняття рішень щодо призначення того чи іншого режиму узгодженості для блоку операцій.

Політика узгодженості (англ. Consistency Policy) – це проміжний компонент, що аналізує блок узгодженості та дані з дескриптора контексту. Результат роботи для цього компонента – це модель узгодженості, що буде застосована для блоку операцій. Конкретна реалізація політик узгодженості може містити як жорсткі правила для визначення моделі на основі конкретних параметрів контексту, так і застосовувати більш просунуті й гнучкі інструменти, як-от штучний інтелект чи машинне навчання.

Всі ці три компоненти разом утворюють контекстно-орієнтовану архітектуру на основі політик узгодженості, що може бути легко інтегрована в розподілену систему в якості проміжного сервісу для процесу обробки запитів. Відповідно, розподілена СКБД на такій архітектурі може застосовувати гібридний підхід для

керування даними: користувач може вказати конкретну бажану модель узгодженості для блоку операцій або залишити цей процес для адаптивних політик узгодженості, що врахують усі вхідні дані, включаючи стан системи та контекст операцій, для прийняття рішення.

Так як використовуваних політик та дескриптор може бути декілька (наприклад, для покриття більшого спектру типових сценаріїв) – розподілена система повинна також мати відповідні сервіс-менеджери для керування цими елементами та оновлення необхідних даних, як-от статистики запитів в одному з дескрипторів контексту. Відповідно, також необхідний сервіс-менеджер для групування запитів в блоки узгодженості за часом, належністю до транзакції або іншими параметрами [6].

Таким чином, контекстно-орієнтована модель утворює конвеєр, який можна інтегрувати в початкову стадію обробки запитів СКБД. В залежності від фактичної реалізації, розподілена система може мати службу черги повідомлень або ж інший механізм-буфер для ефективного накопичення вхідних операцій запису [7], які вже будуть організовані в відповідні пакети менеджером блоків узгодженості.

В залежності від очікуваного характеру навантаження цей етап також може включати в себе оптимізацію запиту, що дозволить підвищити ефективність роботи інших компонентів системи [8]. Наприклад, комплексна транзакція може містити оновлення як часто використовуваних даних, так і записів з низьким пріоритетом. Застосування оптимізації на цьому етапі може дозволити виділити окремі підоперації, що повинні належати до окремих блоків узгодженості.

Наступним кроком конвеєра буде передача підготовлених блоків до менеджера політик узгодженості, що повинен працювати разом з менеджером дескрипторів контексту. Якщо попередня оптимізація та встановлення зв'язків уже було виконано в менеджері блоків, то ці два компоненти можуть працювати з пакетами операцій паралельно або ж зовсім можуть бути виділені в окремий мікросервіс, якщо мережеві можливості розподіленої системи дозволяють введення додаткових затримок на цьому етапі [9].

Конвеєр обробки вхідних операцій вичитування також може потребувати змін, але лише з метою збору статистики та іншої інформації для менеджера дескрипторів контексту. В залежності від реалізації, цей сервіс-менеджер може бути доцільно інтегрувати з існуючими сервісами моніторингу цифрової інфраструктури (наприклад, Prometheus) задля проактивного відстеження показників та покращення механізмів раннього виявлення проблем [10].

Приклад реалізації системи з такою моделлю динамічної узгодженості наведено на рисунку 1.

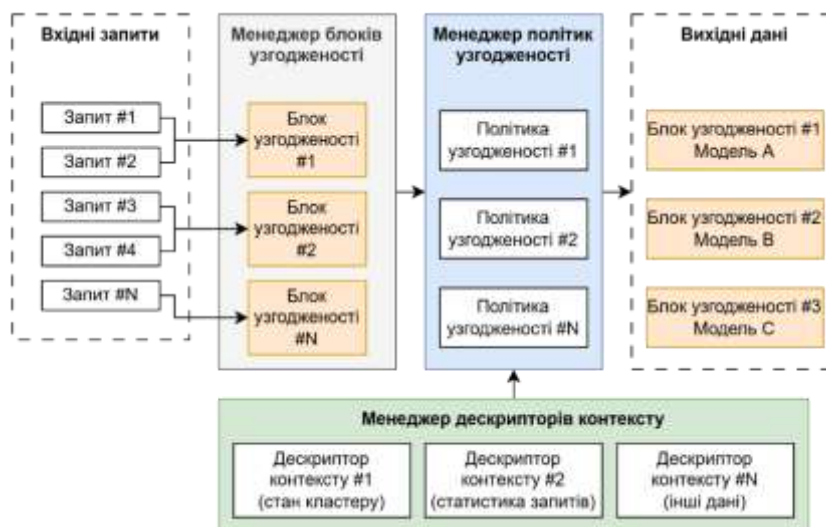


Рис. 1. Розподілена система з контекстно-орієнтованою моделлю динамічної узгодженості

Розглянута модель динамічної узгодженості пропонує лише високорівневу архітектуру для прийняття рішень, її результуюча ефективність залежить від конкретної реалізації політик узгодженості та дескрипторів контексту, як це було розглянуто в аналізі можливих конвеєрів обробки запитів.

Інші моделі, як-от CAnDoR, також застосовують показники розподіленої системи для прийняття рішень, але пропонують більш розширені й адаптивні механізми їх аналізу, що впливають не тільки на режим узгодженості для операцій, але й на структуру самого кластера.

CAnDoR передбачає використання сегментації або поділу даних (англ. Data Sharding), а не повної реплікації на всіх вузлах системи. Основний механізм роботи полягає в постійному вимірюванні двох основних показників: часу синхронізації та часу відповіді – на їх основі автоматично приймаються рішення щодо розподілу сегментів між доступними вузлами системи [11].

В даному випадку, час синхронізації – це період часу, що необхідний для оновлення всіх релевантних вузлів в системі з урахуванням обраного режиму узгодженості. Наприклад, строга узгодженість зазвичай

передбачає синхронну реплікацію даних, тому час обробки запиту буде включати в себе очікування готовності всіх вузлів. Коли як більш слабкі моделі узгодженості можуть застосовувати асинхронну реплікацію або просто проводити її менш регулярно, що також буде впливати в сторону зменшення усередненого часу на синхронізацію вузлів. Крім того, сегментація даних дозволяє гнучко змінювати кількість реплік на вузлах – більше копій будуть вимагати довшого часу синхронізації, що буде впливати на швидкість системи в цілому.

Інший показник, час відповіді, більше корелюється з фізичним розміщенням вузлів та частотою запитів до конкретних вузлів або ж наборів даних. Адаптивна модель повинна враховувати які вузли частіше обробляють запити та які саме дані запитують чи оновлюють користувачі – в такому випадку є сенс розмішувати ці дані на найближчих до клієнтів серверах для максимізації швидкодії. При цьому, варто не забувати, що реальна розподілена система, а тим паче СКБД, повинна балансувати між найкращою швидкістю обробки запитів та доступністю даних, включаючи аварійні ситуації.

Тому, CAnDoR враховує обидва показники для знаходження найкращого варіанту розміщення сегментів даних на вузлах системи. Ця модель також дозволяє використовувати одночасно різні моделі узгодженості для різних операцій [11], але не визначає додаткових вимог чи деталей цього механізму – логіка може бути різною в залежності від кінцевої реалізації. Обраний режим узгодженості для конкретної операції буде впливати на коефіцієнти, що використовуються при виборі вузлів для розміщення сегментів даних.

Таким чином, для розміщення сегментів, де переважають дані для операцій зі строгою узгодженістю, необхідно пріоритизувати зменшення швидкості синхронізації. Для сегментів, що використовуються лише задля кінцевої узгодженості, пріоритет може бути зміщено в сторону меншого часу доступу. Незалежне вимірювання часу для кожного запиту або групи запитів у складі транзакції дозволяє динамічно реагувати на зміни в патернах доступу до даних – з часом розміщення сегментів на вузлах може змінюватись задля підтримки необхідного рівня швидкодії.

Отже, в порівнянні з першою моделлю, CAnDoR пропонує не лише можливість одночасної роботи з різними моделями узгодженості, а й механізм динамічної оптимізації вузлів, що може підвищити ефективність роботи розподілених СКБД або інших типів розподілених інформаційних систем.

Проте, обидві моделі не враховують варіанти вирішення однієї з найбільш розповсюджених проблем розподілених систем – розділення мережі або втрата зв'язку з частиною кластера. Цю проблему намагається вирішити нова R-TBC/RTA модель.

Ключова риса цієї моделі подібна до CAnDoR – в основі R-TBC/TRA лежить механізм для автоматичного визначення оптимального розміщення даних. Проте, ця модель працює не з окремими сегментами на вузлах, а може змінювати топологію розподіленої системи в цілому. Подібно до механізму роботи алгоритмів консенсусу, вузли кластера формують кворум, в якому є центральний/головний вузол, що обробляє запити запису даних та ініціює синхронну реплікацію даних з іншими вузлами [12]. Цей кворум формує набір вузлів, що працюють в режимі строгої узгодженості та зберігають відповідні для цього дані. Наприклад, розподілена СКБД банку може зберігати основні фінансові дані саме на цьому наборі вузлів.

Проте, окрім основного кворуму, можуть бути додаткові вузли, що синхронізуються не з центральним вузлом, а з одним з вузлів зі строгою узгодженістю даних. Таким чином, ця модель пропонує формувати з вузлів розподіленої системи дерево, в якому корінь та напрямку з'єднані з ним вузли використовують строгую узгодженість, а решта вузлів – можуть застосовувати інші механізми реплікації та кінцеву узгодженість. Такі вторинні вузли можуть швидше обробляти запити через асинхронну реплікацію даних [13], а за умови зберігання на них некритичних даних їх втрата не буде впливати на роботу всього кластера.

Окрім застосування підходу з первинних та вторинних вузлів з різними гарантіями узгодженості, ця модель також пропонує алгоритм для автоматичного вертикального та горизонтального сегментування даних – GPA-MDC (англ. Graph Partitioning Algorithm - Multidimensional Data Clustering, алгоритм розбиття графа з багатовимірною кластеризацією даних).

Основні кроки алгоритму включають в себе категоризацію даних за важливістю (тобто, необхідними гарантіями цілісності та узгодженості), обрахунок потенційних зв'язків між елементами для мінімізації розподілених транзакцій, що будуть вимагати одночасної комунікації декількох вузлів, а також формування результатуючих груп сегментів для розміщення на вузлах. Аналогічно до класичних підходів сегментації даних, цей алгоритм також враховує типові патерни доступу при горизонтальній сегментації та пов'язані атрибути/колонки при вертикальній сегментації.

Хоча один з критеріїв роботи GPA-MDC є мінімізація розподілених транзакцій, але R-TBC/TRA в цілому намагається створювати сегменти, що мають спільні дані. Це дозволяє забезпечити стійкість розподіленої системи до відмови вузла або мережних проблем без створення ідентичних вузлів-копій і може допомогти в кращому розподіленні навантаження.

Аналогічно до інших розглянутих моделей, R-TBC/TRA містить механізм для подальшої адаптації розподіленої системи в часі. За результатами вимірювання мережних показників, як-от затримка, доступність вузлів, часткові втрати мережних пакетів, може змінюватись не тільки розміщення окремих сегментів на вузлах, а й відбуватись реструктуризація всієї системи. Наприклад, первинні вузли (що працюють над даними

та операціями зі строгою узгодженістю) можуть змінити свою роль на вторинну (з підтримкою лише кінцевої узгодженості або іншої слабшої моделі), що може бути необхідно для забезпечення доступності менш критичних даних або при зміні пріоритету цих даних. Аналогічно, може відбуватись зміна зв'язків між первинними та вторинними вузлами з метою оптимізації часу синхронізації між ними.

ПОРІВНЯННЯ МОДЕЛЕЙ УЗГОДЖЕНОСТІ

Розглянемо більш детально обрані для порівняння характеристики моделей забезпечення узгодженості даних.

В першу чергу, варто відзначити базову концепцію кожної з моделей та їх механізм визначення рівня узгодженості. Ці характеристики дозволяють коротко підсумувати базові принципи роботи та архітектурні особливості, що також стануть частиною повноцінної розподіленої СКБД, в якій буде використовуватись та чи інша модель.

Розміщення даних описує методи зберігання інформації на вузлах – чи модель передбачає певні правила або ж структурування сегментів чи вузлів. В залежності від обраного підходу можуть існувати додаткові обмеження або ж вимоги до розподілених систем, що важливо враховувати при інтеграції моделі з існуючою або ж розробкою нової СКБД. Ці три критерії можна умовно віднести до архітектурних критеріїв порівняння.

Моніторинг навантаження та адаптація у часі – ці критерії підсумовують можливості моделі узгодженості відслідковувати власний стан або зміни в тенденціях запитів та можливість реагувати на ці зміни. Моделі з вищим рівнем моніторингу будуть мати більше даних для роботи інших складових компонентів архітектури, включаючи механізм адаптації у часі. З іншого боку, наявність додаткових даних не завжди означає, що модель або метод зможуть ефективно використати ці дані. Проте, ці показники все-одно важливі для порівняння теоретичного потенціалу моделей узгодженості.

Стійкість до збоїв описує закладені в моделі узгодженості механізми боротьби з програмними, мережевими та іншими проблемами, що є однією з ключових проблем сучасних розподілених систем [14]. Ці три показники можна віднести до адаптивних критеріїв порівняння.

Можливість інтеграції з існуючими СКБД підсумовує наявні теоретичні відомості про технічну реалізацію методів. Деякі з моделей в відповідних першоджерелах розроблялись та тестувались в складі існуючих розподілених систем, в залежності від сценарію використання це може бути кращим варіантом, аніж розробка власної СКБД з підтримкою динамічної узгодженості даних.

Додатковий аналіз переваг та недоліків розглянутих моделей разом з іншими критеріями дозволить сформувати список можливих сценаріїв застосування та особливостей використання (і розробки) СКБД на основі обраних моделей узгодженості.

Підсумуємо особливості роботи всіх розглянутих моделей за допомогою цих критеріїв, що наведено в таблиці 1.

Таблиця 1

Порівняння моделей динамічної узгодженості даних

Критерій	Контекстно-орієнтована модель	CAnDoR	R-TBC/RTA
Концепція	Контекстна адаптація політик узгодженості	Сегментація даних з адаптивним розміщенням	Ієрархічна топологія з графовою кластеризацією
Визначення рівня узгодженості	Динамічно для кожного блоку на основі політик та контексту	Динамічно на рівні сегментів даних	Динамічно на рівні вузлів системи (ієрархічного дерева)
Розміщення даних	Довільне	Адаптивне, на основі метрик доступності	Дерево з різними пріоритетами та ролями вузлів
Моніторинг навантаження	Довільний, через дескриптори контексту	Відслідковування часу синхронізації та відповіді	Вимірювання мережових показників
Адаптація у часі	Можлива, залежить від реалізації	За допомогою динамічного розміщення сегментів	За допомогою реструктуризації дерева
Стійкість до збоїв	Не передбачена, залежить від реалізації	Не передбачена, залежить від механізму реплікації сегментів	Передбачена на рівні побудови дерева
Можливість інтеграції з існуючими СКБД	Висока, може бути проміжною службою	Середня, потребує змін в логіці розміщенні даних	Нижча, потребує специфічної топології та структури зберігання даних
Основна перевага	Відносна простота реалізації, можливість глибокої кастомізації	Відносна простота реалізації, потенційна сумісність з СКБД з сегментацією даних	Висока стійкість, що передбачена на архітектурному рівні
Основне обмеження	Висока залежність від кінцевої реалізації розподіленої системи	Висока залежність на точність зібраних метрик	Висока складність реалізації

Варто відмітити, що через залежність від технічної реалізації цей аналіз є теоретичним і базується на відкритих даних, тому його варто використовувати в контексті порівняння моделей між собою.

Тим не менш, систематизація характеристик і показників моделей за цими критеріями дозволяє відслідкувати ряд особливостей. В першу чергу, на основі такого порівняння можна відслідкувати тенденцію – більш прості моделі, як-от контекстно-орієнтована, мають більшу гнучкість, але їх ефективність роботи та інші характеристики складно прогнозувати до моменту фактичної реалізації розподіленої системи.

Більш комплексні моделі, як-от R-TBC/RTA, дають більш чітке розуміння поведінки в багатьох сценаріях, але їх реалізація є більш складною та вимогливою. Модель CAnDoR займає проміжну позицію з більшим ухилом до простоти контекстно-орієнтованої моделі.

Відносно сценаріїв застосування, контекстно-орієнтована модель є найбільш універсальною та залишає багато деталей, включаючи архітектурних, на розсуд фінальної технічної реалізації розподіленої системи або ж інтеграції з існуючою. Інші моделі ж будуть сумісні з СКБД з застосуванням сегментації даних та можуть потребувати більш глибокої інтеграції з цими механізмами роботи.

Більш того, варто відмітити більшу залежність CAnDoR та R-TBC/RTA від зібраних даних моніторингу. Контекстно-орієнтована модель може працювати з довільними політиками узгодженості, що потенційно може передбачати розробку окремої системи узгодженості на основі бізнес-правил для конкретного сценарію застосування. Решта моделей більше покладаються на зібрані дані та показники системи, тому їх підтримка таких довільних розширень може бути обмежена.

В цілому, R-TBC/RTA виглядає найбільш перспективно через вбудовану стійкість до збоїв та більш комплексний механізм сегментації, проте ця модель є найбільш складною в реалізації. Можливо, доцільним рішенням буде інтегрувати її ієрархічну архітектуру з деякими механізмами інших моделей, як-от політиками узгодженості, для розробки більш гнучкої та відмовостійкої розподіленої системи.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

І ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ У ДАНОМУ НАПРЯМІ

У межах проведеного дослідження було здійснено ґрунтовний аналіз сучасних динамічних моделей забезпечення узгодженості даних у розподілених системах керування базами даних. Основною відмінністю таких моделей від класичних підходів є можливість адаптації режиму узгодженості в залежності від контексту виконання операцій, типу транзакції та системного навантаження.

Наукова новизна дослідження полягає в систематизації характеристик моделей динамічної узгодженості даних з різною архітектурою та їх порівнянні за рядом відібраних різносторонніх критеріїв. Було проаналізовано архітектуру та механізми роботи трьох моделей: контекстно-орієнтованої, що динамічно визначає режим узгодженості для блоку операцій на основі політик та контексту; CAnDoR, що за допомогою метрик доступності вузлів системи може динамічно змінювати розміщення даних при сегментації; R-TBC/RTA, що може побудувати комплексну топологію кластера розподіленої системи на основі мережевих показників та цільового призначення даних. Також розглянуто варіанти технічної реалізації динамічних моделей узгодженості, як-от інтеграція контекстно-орієнтованої моделі в конвеєр обробки та оптимізації вхідних запитів.

Проведено порівняння моделей на основі архітектурних (концепція, визначення рівня узгодженості, розміщення даних), адаптивних (моніторинг навантаження, адаптація у часі, стійкість до збоїв) та інших (можливість інтеграції, переваги, недоліки) критеріїв. Порівняння дозволило відслідкувати тенденцію між гнучкістю поведінки моделей, залежністю від реалізації та загальним рівнем закладених механізмів гарантування узгодженості на рівні архітектури моделі. Більш прості моделі, як-от контекстно-орієнтована, передбачають більш простий функціонал та складність реалізації, але потенційно мають вищу гнучкість та потенціал інтеграції в уже існуючі системи. Коли як R-TBC/RTA та інші подібні системи є більш складними, але мають кращі гарантії відмовостійкості.

Отримані результати показують, що всі розглянуті моделі мають механізми гарантування узгодженості даних, що можуть бути ефективними для певних сценаріїв використання та сильно відрізняються складністю реалізації. Результати аналізу можна використати на практиці для розробки нових динамічних моделей забезпечення, що зможуть об'єднати покращені гарантії відмовостійкості та гнучкість застосування в різних сценаріях. Також подальшого дослідження заслуговують інші методи розподілених систем, як-от сегментація даних, що грає важливу роль в двох з трьох розглянутих моделей узгодженості. Це може дозволити знайти нові шляхи покращення швидкодії та інших характеристик розподілених систем керування базами даних.

Література

1. TDSQL: Tencent distributed database system / Y. Chen та ін. Proceedings of the VLDB Endowment. 2024. Т. 17, № 12. С. 3869–3882. URL: <https://doi.org/10.14778/3685800.3685812>.
2. Adaptive trade-off between consistency and performance in data replication / H. Sun та ін. Software: Practice and Experience. 2016. Т. 47, № 6. С. 891–906. URL: <https://doi.org/10.1002/spe.2462> (дата звернення: 30.06.2025).

3. A comparative analysis of adaptive consistency approaches in cloud storage / A. Khelaifa та ін. Journal of Parallel and Distributed Computing. 2019. Т. 129. С. 36–49. URL: <https://doi.org/10.1016/j.jpdc.2019.03.006> (дата звернення: 30.06.2025).
4. Almeida P. S. A framework for consistency models in distributed systems. arXiv preprint. 2024. URL: <https://doi.org/10.48550/arXiv.2411.16355>.
5. Consistency models in distributed systems: A survey on definitions, disciplines, challenges and applications / H. N. S. Aldin та ін. arXiv preprint. 2019. URL: <https://doi.org/10.48550/arXiv.1902.03305>.
6. Mahfoud Z., Nouali-Taboudjemat N. Enhancing Data Consistency via a Context-Aware Dynamic Adaptive Model. International Journal of Computing and Digital Systems. 2024. Т. 15, № 1. С. 543–554. URL: <https://doi.org/10.12785/ijcds/160141> (дата звернення: 30.06.2025).
7. Grötschla F., Lehmann Miotto G., Sipos R. Design of a request/response buffering application for I/O intensive workloads. Journal of Physics: Conference Series. 2023. Т. 2438, № 1. С. 012025. URL: <https://doi.org/10.1088/1742-6596/2438/1/012025> (дата звернення: 30.06.2025).
8. Ramu V. B. Optimizing Database Performance: Strategies for Efficient Query Execution and Resource Utilization. International Journal of Computer Trends and Technology. 2023. Т. 71, № 7. С. 15–21. URL: <https://doi.org/10.14445/22312803/ijctt-v71i7p103> (дата звернення: 30.06.2025).
9. Ramu V. B. Performance Impact of Microservices Architecture. The Review of Contemporary Scientific and Academic Studies. 2023. Т. 3, № 6. URL: <https://doi.org/10.55454/rcsas.3.06.2023.010> (дата звернення: 30.06.2025).
10. sai K. Enhanced Visibility for Real-time Monitoring and Alerting in Kubernetes by Integrating Prometheus, Prometheus, Grafana, Loki, and Alerta. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 2024. Т. 08, № 06. С. 1–5. URL: <https://doi.org/10.55041/ijrsrem35639> (дата звернення: 30.06.2025).
11. Mauffret E., Vernier F., Monnet S. CAnDoR: Consistency Aware Dynamic data Replication. 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), м. Cambridge, MA, USA, 26–28 верес. 2019 р. 2019. URL: <https://doi.org/10.1109/nca.2019.8935051> (дата звернення: 30.06.2025).
12. Advanced consistency management of highly-distributed transactional database in a hybrid cloud environment using novel R-TBC/RTA approach / J. Dizdarevic та ін. Journal of Cloud Computing. 2021. Т. 10, № 1. URL: <https://doi.org/10.1186/s13677-021-00230-0> (дата звернення: 30.06.2025).
13. Singh A., Shanker U. Database replication techniques: a review. International journal of research review in engineering science & technology. 2016. Т. 5, № 4. С. 16–19. URL: <https://doi.org/10.13140/RG.2.2.13868.44165> (дата звернення: 30.06.2025).
14. Development of a high availability method for configuration management software / A. V. Myrnodskyy та ін. Optoelectronic Information-Power Technologies. 2023. Т. 46, № 2. С. 64–75. URL: <https://doi.org/10.31649/1681-7893-2023-46-2-64-75> (дата звернення: 30.06.2025).

References

1. Chen, Y., Pan, A., Lei, H., Ye, A., Han, S., Tang, Y., Lu, W., Chai, Y., Zhang, F. and Du, X., (2024). TDSQL: Tencent distributed database system. Proceedings of the VLDB Endowment [online]. 17(12), 3869–3882. Available from: doi: 10.14778/3685800.3685812
2. Sun, H., Xiao, B., Wang, X. and Liu, X., (2016). Adaptive trade-off between consistency and performance in data replication. Software: Practice and Experience [online]. 47(6), 891–906. [Viewed 30 June 2025]. Available from: doi: 10.1002/spe.2462
3. Khelaifa, A., Benharzallah, S., Kahloul, L., Euler, R., Laouid, A. and Bounceur, A., (2019). A comparative analysis of adaptive consistency approaches in cloud storage. Journal of Parallel and Distributed Computing [online]. 129, 36–49. [Viewed 30 June 2025]. Available from: doi: 10.1016/j.jpdc.2019.03.006
4. Almeida, P. S., (2024). A framework for consistency models in distributed systems. arXiv preprint [online]. Available from: doi: 10.48550/arXiv.2411.16355
5. Aldin, H. N. S., Deldari, H., Moattar, M. H. and Ghods, M. R., (2019). Consistency models in distributed systems: A survey on definitions, disciplines, challenges and applications. arXiv preprint [online]. Available from: doi: 10.48550/arXiv.1902.03305
6. Mahfoud, Z. and Nouali-Taboudjemat, N., (2024). Enhancing Data Consistency via a Context-Aware Dynamic Adaptive Model. International Journal of Computing and Digital Systems [online]. 15(1), 543–554. [Viewed 30 June 2025]. Available from: doi: 10.12785/ijcds/160141
7. Grötschla, F., Lehmann Miotto, G. and Sipos, R., (2023). Design of a request/response buffering application for I/O intensive workloads. Journal of Physics: Conference Series [online]. 2438(1), 012025. [Viewed 30 June 2025]. Available from: doi: 10.1088/1742-6596/2438/1/012025
8. Ramu, V. B., (2023). Optimizing Database Performance: Strategies for Efficient Query Execution and Resource Utilization. International Journal of Computer Trends and Technology [online]. 71(7), 15–21. [Viewed 30 June 2025]. Available from: doi: 10.14445/22312803/ijctt-v71i7p103
9. Ramu, V. B., (2023). Performance Impact of Microservices Architecture. The Review of Contemporary Scientific and Academic Studies [online]. 3(6). [Viewed 30 June 2025]. Available from: doi: 10.55454/rcsas.3.06.2023.010
10. sai, K., (2024). Enhanced Visibility for Real-time Monitoring and Alerting in Kubernetes by Integrating Prometheus, Prometheus, Grafana, Loki, and Alerta. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT [online]. 08(06), 1–5. [Viewed 30 June 2025]. Available from: doi: 10.55041/ijrsrem35639
11. Mauffret, E., Vernier, F. and Monnet, S., (2019). CAnDoR: Consistency Aware Dynamic data Replication. In: 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), 26–28 September 2019, Cambridge, MA, USA [online]. IEEE. [Viewed 30 June 2025]. Available from: doi: 10.1109/nca.2019.8935051

12. Dizdarevic, J., Avdagic, Z., Orucevic, F. and Omanovic, S., (2021). Advanced consistency management of highly-distributed transactional database in a hybrid cloud environment using novel R-TBC/RTA approach. *Journal of Cloud Computing* [online]. 10(1). [Viewed 30 June 2025]. Available from: doi: 10.1186/s13677-021-00230-0
13. Singh, A. and Shanker, U., (2016). Database replication techniques: a review. *International journal of research review in engineering science & technology* [online]. 5(4), 16–19. [Viewed 30 June 2025]. Available from: doi: 10.13140/RG.2.2.13868.44165
14. Myrholdskyy, A. V., Romanyuk, O. V., Romanyuk, O. N. and Titova, N. V., (2023). Development of a high availability method for configuration management software. *Optoelectronic Information-Power Technologies* [online]. 46(2), 64–75. [Viewed 30 June 2025]. Available from: doi: 10.31649/1681-7893-2023-46-2-64-75