

УДК 004

Ю.Омелян, магістр гр. КН-22М-2

*Центральноукраїнський національний технічний університет*

## ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДВИЩЕННЯ НАДІЙНОСТІ ЗБЕРІГАННЯ ДАНИХ У ХМАРІ

У статті розроблено програмне забезпечення, яке призначено для системи підвищення надійності зберігання даних у хмарі. Метою розробки є дослідження та програмна реалізація системи підвищення надійності зберігання даних у хмарі. Об'єктом дослідження є процес підвищення надійності зберігання даних у хмарі. Предметом дослідження є методи підвищення надійності зберігання даних у хмарі. Методи дослідження базуються на методах теорії надійності, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи підвищення надійності зберігання даних у хмарі. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

**Постановка проблеми.** Ріст обсягів даних, що зросли вимоги до надійності зберігання й швидкодії доступу до даних роблять необхідним виділення засобів зберігання в окрему підсистему Обчислювального Комплексу (ОК). Роль і важливість системи зберігання визначаються постійно зростаючою цінністю інформації в сучасному суспільстві, можливість доступу до даних і керування ними є необхідною умовою для виконання бізнес-процесів. Безповоротна втрата даних піддає бізнес серйозній небезпеці. Втрачені обчислювальні ресурси можна відновити, а втрачені дані, при відсутності грамотної спроектованої й впровадженої системи резервування, уже не підлягають відновленню. По даним Gartner, серед компаній, що постраждали від катастроф і які пережили велику необоротну втрату корпоративних даних, 43% не змогли продовжити свою діяльність. Доступ до даних неможливий як у випадку виходу з ладу каналів (доступу) або обчислювальних засобів, так і у випадку відсутності необхідної продуктивності для виконання прикладних завдань. Виділення засобів зберігання даних в окрему підсистему в рамках Обчислювального Комплексу дозволить проєктувальникам сконцентруватися на рішенні проблем забезпечення надійного зберігання й доступу до даних у рамках однієї підсистеми. Крім того, це створює передумови для оформлення системи зберігання даних (СЗД) в організаційно-технічну структуру, що є основою для аутсорсингу послуг з надання засобів зберігання даних.

**Аналіз останніх досліджень і публікацій.** При аналізі останніх досліджень і публікацій [1-20] було виявлено певні прогалини у забезпеченні системи підвищення надійності зберігання даних у хмарі.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи підвищення надійності зберігання даних у хмарі.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем підвищення надійності зберігання даних у хмарі.
- Дослідження системи підвищення надійності зберігання даних у хмарі.
- Програмна реалізація системи підвищення надійності зберігання даних у хмарі.

*Об'єктом дослідження* є процес підвищення надійності зберігання даних у хмарі.

*Предметом дослідження* є методи підвищення надійності зберігання даних у хмарі.

*Методи дослідження* базуються на методах теорії надійності, методах математичної статистики, методах розробки програмного забезпечення.

**Виклад основного матеріалу.** Мережною інфраструктурою, що поєднує велику кількість серверів і пристроїв зберігання, необхідно управляти й, як мінімум, відслідковувати її стан. Сказане не означає, що немає необхідності моніторингу стану, наприклад, двох серверів і одного масиву, підключеного до них прямо. Однак, це можна реалізувати підручними засобами – убудованими утилітами серверів, масиву й операційної системи, безкоштовними (freeware) утилітами або скриптами. Кожне із пристроїв у СЗД має кілька об'єктів, що вимагають керування й контролю стану, наприклад дискові групи й томи в масивів, порти в масивів і комутаторів, адаптери в серверах. Як тільки число об'єктів керування в СЗД починає обчислюватися десятками, керування такою конфігурацією за допомогою "підручних" засобів віднімає в адміністраторів занадто багато часу й сил, і неминуче приводить до помилок. Упоратися з таким завданням можна тільки використовуючи повномасштабну систему керування. Це справедливо для будь-яких великих систем і для великої системи зберігання даних, зокрема. Впровадження системи керування стає особливо актуальним у тих випадках, коли система зберігання даних виділена не тільки структурно й функціонально, але й організаційно.

Система зберігання даних повинна включати наступні підсистеми й компоненти:

– Пристрої зберігання даних: дискові масиви й стрічкові бібліотеки. Сучасні високопродуктивні дискові масиви використовують технологію Fibre Channel для підключення до них серверів і для доступу до дисків усередині масиву. Вони можуть масштабуватися до десятків терабайт дискового простору й мають убудований інтелект для виконання спеціальних функцій, таких як: віртуалізація дискового простору, розмежування доступу до дискового простору, створення Point-In-Time (PIT) копій даних і реплікація даних між масивами. До пристроїв зберігання даних також ставляться всілякі бібліотеки – стрічкові, магнітооптичні й CD/DVD, які в розглядатися не будуть. Визначення поняття Point-In-Time копії даних (PIT-копія, іноді зустрічається скорочення I-T-копія) треба з його назви – це копія даних, зроблена на певний момент часу, і стан даних "заморожено" у момент створення копії. Іноді плутають PIT-копії з "моментальними знімками" (SnapShot), які в дійсності є тільки одним з методів створення PIT-копій. До інших методів створення PIT-копій ставляться методи клонування (clone) даних.

– Інфраструктуру доступу серверів до пристроїв зберігання даних. У цей час, як правило, інфраструктура доступу серверів до пристроїв зберігання даних створюється на основі технології SAN. SAN є високопродуктивною інформаційною мережею, орієнтованою на швидку передачу більших обсягів даних.

– В основі концепції SAN лежить можливість з'єднання кожного із серверів з будь-яким пристроєм зберігання даних, що працюють за протоколом Fibre Channel. Мережа зберігання даних утворюють: волоконно-оптичні з'єднання, Fibre Channel Host Bus Adapters (FC-HBA) і FC-комутатори, у цей час передачі, що забезпечують швидкість, 200 Мбайт/с і відстань між об'єктами, що з'єднуються, до декількох десятків кілометрів. У випадку, якщо відстань між об'єктами перевищує можливості FC-устаткування ні достатньої кількості "темної" оптики, зв'язок між об'єктами можна забезпечити використовуючи технологію ущільненого спектрального мультиплексування DWDM або інкапсулював FibreChannel в інший транспортний протокол, наприклад в TCP/IP. Технологія DWDM (Dense Wavelength Division Multiplexing) дозволяє оптимальним образом застосовувати оптоволоконні ресурси й передавати не тільки трафік Fibre Channel, але також Ethernet і інші протоколи по тим самим оптичних каналах одночасно. При цьому відстані між об'єктами, що з'єднуються, можуть становити сотні й навіть тисячу кілометрів. Докладніше про SAN можна прочитати в [1]. "Темна" оптика – це технічний жаргон, що позначає оптичну магістраль (кабель) на шляху проходження якої не встановлені ніякі активні пристрої. Відсутність таких пристроїв має на увазі, що по кабелю не передається ніяких сигналів. Для оптики таким сигналом є світло, тобто в оптичний кабель не світить ні який пристрій. Звідси й походження терміна. Без

застосування додаткових пристроїв, наприклад FC-ATM конвертерів, FC-комутатори не можуть віддавати пакети по магістралі, де присутні інші активні пристрої.

– Систему резервного копіювання й архівування даних. Система призначена для створення резервних копій і відновлення даних. Система резервного копіювання дозволяє захистити дані від руйнування не тільки у випадку збоїв або виходу з ладу апаратури, але й у результаті помилок програмних засобів і користувачів.

– Виконання резервного копіювання є одним з необхідних методів забезпечення безперервності бізнесу. Створення централізованої системи резервного копіювання дозволяє скоротити сукупну вартість володіння ІТ-інфраструктурою за рахунок оптимального використання пристроїв резервного копіювання й скорочення витрат на адміністрування (у порівнянні з децентралізованою системою).

– Програмне забезпечення керування зберіганням даних. Програмне забезпечення призначене для рішення завдань керування зберіганням даних, наприклад, для розмітки дискових томів або підвищення продуктивності доступу до даних прикладного ПЗ. Наприклад, убудоване в масиви Hitachi Lightning 9900V програмне забезпечення Cruise Control збирає статистику по інтенсивності роботи з даними, і виходячи з її ухвалює рішення щодо переміщення даних на диски, продуктивність яких відповідає швидкості звертання до даних.

– Систему керування. Система призначена для моніторингу й керування рівнем якості сервісу зберігання даних. Вона тісно інтегрується із системою керування ОК. Основою системи керування СЗД є засоби керування апаратними ресурсами мережі зберігання даних. Їхня інтеграція з іншими системами дає можливість контролювати ресурси СЗД і управляти ними на всіх рівнях, від дисків у масиві до файлової системи сервера.

Серед підсистем СЗД система резервного копіювання заслуговує на особливу увагу. Як треба з визначення, створення системи резервного копіювання є одним із засобів забезпечення надійного зберігання даних, про які поговоримо нижче. Однак, систему резервного копіювання необхідно включити в СЗД як окрему підсистему не тільки із цієї причини. Обсяг даних, вимірюваний одиницями й десятками терабайтів, вимагає усе більше часу на процедуру резервного копіювання. Класичні засоби резервного копіювання по ЛОМ не встигають виконати цю процедуру й укластися у відведене тимчасове "вікно", що скорочується з наближенням режиму роботи інформаційної системи до "24x7" (наприклад, у системах обслуговуючі регіони із центра). Рішенням зазначеної проблеми є використання SAN для передачі даних резервного копіювання, а також застосування засобів сучасних дискових масивів для створення PIT-копій. У цьому випадку буде потрібно тісна інтеграція системи резервного копіювання з SAN і дисковими масивами.

#### **Завдання які стоять перед системою зберігання даних**

Система зберігання даних призначена для організації надійного зберігання даних, а також відказостійкого, високопродуктивного доступу серверів до пристроїв зберігання. Існуючі в цей час методи по забезпеченню надійного зберігання даних і відказостійкого доступу до них можна охарактеризувати одним словом – дублювання.

Так, для захисту від відмов окремих дисків використовуються технології RAID, які (крім RAID-0) застосовують дублювання даних, збережених на дисках. Рівень RAID-5 хоча й не створює копій блоків даних, але все-таки зберігає надлишкову інформацію, що теж можна вважати дублюванням. Для захисту від логічного руйнування даних (руйнування цілісності бази даних або файлової системи), викликаних збоями в устаткуванні, помилками в програмному забезпеченні або невірних діях обслуговуючого персоналу, застосовується резервне копіювання, що теж є дублюванням даних. Для захисту від втрати даних внаслідок виходу з ладу пристроїв зберігання через техногенну або природну катастрофу, дані дублюються в резервний центр.

Відказостійкість доступу серверів до даних досягається дублюванням шляхів доступу. Стосовно до SAN дублювання полягає в наступному: мережа SAN будується як дві фізично незалежні мережі, ідентичні по функціональності й конфігурації. У кожний із серверів,

включених в SAN, установлюється як мінімум по двох FC-HBA. Перший з FC-HBA підключається до однієї "половинці" SAN, а другий – до іншої. Відмова встаткування, зміна конфігурації або регламентні роботи на одній із частин SAN не впливають на роботу іншої. У дисковому масиві відказостійкість доступу до даних забезпечується дублюванням RAID-контролерів, блоків живлення, інтерфейсів до дисків і до серверів. Для захисту від втрати даних дзеркалюють ділянки кеш-пам'яті, що беруть участь в операції записи, а електроживлення кеш-пам'яті резервують батареями. Шляхи доступу серверів до дискового масиву теж дублюються. Зовнішні інтерфейси дискового масиву, включеного в SAN, підключаються до обох її "половинок". Для перемикання зі шляху, що вийшов з ладу, доступу на резервний, а також для рівномірного розподілу навантаження між всіма шляхами, на серверах установлюється спеціальне програмне забезпечення, що поставляється або виробником масиву (EMC CLARiiON – PowerPath, HP EVA – AutoPath, HDS – HDLM), або третім виробником (VERITAS Volume Manager).

Необхідну продуктивність доступу серверів до даних можна забезпечити створенням виділеної високошвидкісної транспортної інфраструктури між серверами й пристроями зберігання даних (дисковим масивом і стрічковими бібліотеками). Для створення такої інфраструктури в цей час найкращим рішенням є SAN. Використання сучасних дискових масивів з достатнім обсягом кеш-пам'яті й продуктивної, що не має "вузьких місць" внутрішньою архітектурою обміну інформацією між контролерами й дисками, дозволяє здійснювати швидкий доступ до даних. Оптиміальне розміщення даних (disk layout) по дисках різної ємності й продуктивності, з потрібним рівнем RAID залежно від класів додатків (СУБД, файлові сервіси й т.д.), є ще одним способом збільшення швидкості доступу до даних.

Disk layout – це схема розподілу дані додатки по дисках. Вона враховує в які рівні RAID організовані диски, число й розміри розділів на дисках, які файлові системи використовуються й для зберігання яких типів даних вони призначені.

Необхідно помітити, що оптимізація налаштувань програмних засобів, як самих додатків, так і операційної системи, дає істотно більший приріст продуктивності системи, ніж використання могутнішої апаратури. Обумовлено це в першу чергу тим, що оптимізація налаштувань усуває "вузькі місця" (bottleneck) на шляхах проходження потоків даних, тоді як нова апаратура робить "горлечко пляшки" ледве ширше й тільки (хоча іноді й цього досить для рішення проблем швидкодії). У рішенні завдання оптимізації може допомогти застосування спеціального ПЗ, у якому реалізовані функції, що враховують особливості взаємодії апаратури, операційної системи й прикладного ПЗ. Прикладом такого ПЗ служить опція Quick I/O файлової системи VxFS. Опція Quick I/O ліцензується в складі пакета VERITAS DataBase Edition (DBE) for ORACLE. Зазначена опція дозволяє СУБД ORACLE використовувати Kernel Asynchronous IO (KAIO) для доступу до файлів даних, що істотно підвищує продуктивність операцій введення-виводу СУБД. Докладніше про це можна прочитати в [4].

Крім досягнення необхідних показників продуктивності, відказостійкості й надійності зберігання даних у СЗД, замовники також прагнуть скоротити сукупну вартість володіння системою (Total cost of ownership – TCO). Впровадження системи керування дозволяє скоротити витрати на адміністрування СЗД і спланувати витрати на її модернізацію. Консолідація технічних засобів також сприяє скороченню витрат на експлуатацію СЗД.

### **Завдання які треба вирішити проектувальникові в процесі створення системи зберігання даних**

У процесі створення СЗД необхідно досягти оптимального співвідношення продуктивності, доступності (надійного зберігання й відказостійкого доступу) і сукупної вартості володіння.

Одним з найбільше часто використовуваних методів забезпечення високої доступності СЗД є дублювання, що підвищує вартість системи. Якщо не враховувати бізнес-вимог замовника до доступності системи, то система стає невиправдано дорогою. Погоня за

непотрібною продуктивністю також приведе до використання дорогих технічних засобів. Крім високих показників продуктивності, доступності й низкою вартості потрібно також забезпечити необхідну функціональність – обсяг зберігання й число серверів, що підключаються.

На жаль, замовники не завжди можуть описати вимоги по продуктивності в кількісних характеристиках, прийнятих для систем зберігання – пропускної здатності (Мбайт/с) або продуктивності (операцій уведення-виводу в секунду – I/O per second (IOPS)). Тому, щоб визначити якщо не кількісні характеристики, те хоча б характер навантаження, проектувальникові необхідно з'ясувати, роботу яких додатків повинна забезпечувати СЗД.

Різні класи додатків створюють різне навантаження на дискову підсистему. Наприклад, для СУБД характерні види навантажень, що залежать від класів завдань – транзакційні системи (Online Transaction Processing (OLTP)) і аналітичні системи (Decision Support Systems (DSS)). Для завдань класу OLTP характерним є великий потік запитів на уведення-вивід невеликих порцій даних. Для завдань класу DSS, навпроти, характерно невелике число запитів на читання більших порцій інформації.

Від того, яку навантаження дає додаток, залежить вибір способу розподілу даних по дисках і визначення обсягу кеш-пам'яті дискового масиву. Так для OLTP-завдань наявність кеш-пам'яті в дисковому масиві може зіграти істотну роль для підвищення продуктивності вводу-виводу. Навпроти, у завданнях класу DSS відбувається зчитування більших обсягів даних, що практично виключає їхнє повторне влучення в кеш-пам'ять (на відміну від OLTP-завдань). Таким чином, кешування зчитувальних даних при обробці DSS-завдань не завжди збільшує їхню продуктивність.

До типів навантаження на СЗД, виробленими завданнями класу OLTP, DSS і файловими сервісами можна віднести інші відомі типи додатків. Так, поштовий сервіс, побудований на базі MS Exchange або Lotus Domino, дає подібне навантаження на СЗД, що й OLTP, оскільки зазначені продукти побудовані на базі СУБД. Поштовий же сервіс, заснований на sendmail, робить навантаження на СЗД, характерну для файлової системи із частою зміною метаданих. Засоби резервного копіювання виконують послідовне читання даному, підлягаючому резервному копіюванню, що робить характер їхніх операцій схожим з DSS-завданнями.

Існує ще один, не згаданий раніше, тип навантаження, характерний для процесів журналювання. Прикладом можуть служити запису журналів транзакцій СУБД або журналів подій операційних систем. До цього класу також можна віднести завдання завантаження даних у БД або сховище даних (Data Warehouse). Навантаження, здійснюване на СЗД цим класом завдань, аналогічна навантаженню DSS тільки для операцій запису. Тут наявність кеш-пам'яті (на запис) у дисковому масиві не збільшує продуктивності запису даних. Дану тезу необхідно пояснити. Звичайне застосування кеш-пам'яті на запис зменшує час, що сервер витрачає на операцію запису й очікування її завершення. Але при записі великого обсягу інформації (завантаження даних у БД) або при записі даних, які пишуться на нове місце, таких як журнал транзакцій, існує висока ймовірність виникнення ситуації, коли кеш-пам'ять на запис уже буде повністю зайнята, а новий записуваний блок не відповідає жодному із уже присутніх у кеш-пам'яті. У цьому випадку масиву прийде звільнити кілька блоків кеш-пам'яті, записавши їхній зміст на диски, перш ніж почати обробку операції, що надійшла.

Визначити класи завдань, які буде обслуговувати СЗД, необхідно не тільки для виробітку політики роботи з кеш, але також для розподілу даних по дисках (disk layout). Для забезпечення надійного зберігання інформації диски організуються в рівні RAID 1, 0+1/1+0 або 5. Самим економічним з погляду використання додаткового (дублюючого) дискового простору є рівень RAID 5. Однак продуктивність RAID 5 нижче, ніж в RAID 1 або 0+1 при частих випадкових змінах даних, характерних для OLTP-завдань, і висока при зчитуванні даних, що характерно для DSS-завдань.

Також різні рівні RAID забезпечують різні рівні відказостійкості дискової пам'яті до збоїв окремих дисків. Так RAID 5 не рятує від двох послідовних відмов дисків, втім, як і RAID 0+1, якщо це диски різних половин "дзеркала". Найбільше відказостійким є рівень RAID 1+0 (попарне "дзеркалювання" дисків і потім їх "striping"), тому його рекомендується застосовувати для зберігання критичних даних, наприклад, журналів транзакцій СУБД. Практика показує, що для зберігання файлових систем і даних DSS-завдань досить використовувати RAID 5. Однак, якщо файлова система часто змінюється як, наприклад, у поштових системах sendmail, те має сенс використовувати журнальовану файлову систему або файлову систему з окремо збереженими метаданими, наприклад файлову систему Sun QFS. Тоді для зберігання журналів або окремих метаданих краще застосовувати RAID 1 або 1+0.

Striping – метод розміщення даних на дисках, при якому послідовно, що йдуть блоки, даних, складові логічний том, записуються по черзі на кожний фізичний диск, що входить у дискову групу. У такий спосіб досягається більша продуктивність, оскільки операції читання й запису на диски можуть вироблятися паралельно в порівнянні з варіантом, коли всі блоки логічного тому записується на один диск. Докладніше про striping і його вплив на продуктивність уведення-виводу можна прочитати в [3].

Докладніше про вплив кеш-пам'яті й disk layout на продуктивність уведення-виводу завдань класу OLTP і DSS можна прочитати в [3].

Для "великих" систем актуальною стає оптимізація налаштувань СЗД, спрямована на підвищення швидкодії для досягнення заданого рівня сервісу. Під "великий" розуміється така система, у якій обробляється значний обсяг даних – одиниці й десятки терабайт, і/або до СЗД підключені десятки серверів. Для невеликих систем проблеми зі швидкодією можна вирішити застосуванням більше продуктивної апаратури. В "великих" системах такий підхід може виявитися неприйнятним або у зв'язку з тим, що буде потрібно дуже дорога апаратура, або у зв'язку з тим, що вже досягнуть межа апаратної продуктивності. Єдиним рішенням у цьому випадку є оптимізація. Для оптимізації продуктивності СЗД бажано мати можливість управляти налаштуваннями на всьому шляху проходження даних від процесора до дисків і назад. Для СУБД ORACLE така оптимізація полягає в можливості використовувати КАІО, а також можливості відключити кеш файлової системи для файлів даних ORACLE (оскільки СУБД ORACLE кешує дані у власній області пам'яті SGA). Для цієї мети можна використовувати згаданий раніше пакет VERITAS DBE. Якщо в системі експлуатуються OLTP– і DSS-завдання (що є типовим для більшості систем), то для оптимізації продуктивності дискового масиву бажано мати можливість управляти налаштуваннями кеш-пам'яті для кожного логічного диска (LUN) окремо. Це необхідно для того, щоб для тих дисків, де розташовані дані OLTP-завдання, використовувати кеш (і бажано великого обсягу), а для дисків з даними DSS-завдання використання кеш-пам'яті відключити. Однак, якщо для OLTP– і DSS-завдань використовуються ті самі таблиці даних, те такі налаштування втрачають зміст доти, поки не буде вирішене питання про фізичне рознесення даних завдань по різних дисках, а виконання самих завдань перенесено на різні сервери. Це можна реалізувати засобами СУБД, наприклад, за допомогою експорту даних у файл і імпорту їх в іншу базу. Якщо обсяг даних великий і синхронізацію баз даних OLTP– і DSS-завдань треба проводити досить часто, цей варіант може виявитися неефективним. Тут може допомогти технологія створення РІТ-копій даних, реалізована в більшості сучасних дискових масивів.

Вище говорилося, що СЗД є важливою ланкою в забезпеченні заданого рівня сервісів, надаваних інформаційною системою. Рівень сервісу залежить не тільки від продуктивності СЗД, питання забезпечення якої тільки що обговорювалися, але також від готовності й надійного зберігання даних, інакше кажучи, від доступності СЗД. Виходячи з бізнес-вимог до інформаційної системи, необхідно визначити режими її роботи (24x7, 12x5 і т.п.), ступінь критичності даних залежно від ступеня критичності сервісів, що використовують ці дані,

вимоги до готовності й надійності даних залежно від ступеня їхньої критичності й режимів роботи системи.

Розглянемо для приклада роботу інформаційної системи комерційного банку. У банку експлуатується Автоматизована Банківська Система (АБС), що обслуговує фінансові транзакції клієнтів банку (OLTP-завдання). Режим роботи банку 8: 00-20:00. Банк має кілька філій у ряді регіонів України, які працюють із АБС головного офісу в термінальному режимі. Робочі годинники АБС становлять 6: 00-22:00. Припустимий час простою АБС – не більше 1 години. Припустимо втратити дані АБС не більш ніж за 0,5 години, оскільки за цей період вони можуть бути повторно уведені в систему з паперових носіїв (фактично це обумовлено бізнес-вимогою за часом проходження фінансової транзакції). Також у банку експлуатуються аналітичні завдання (DSS) на основі даних з АБС. Робочий час аналітиків 9:00-18:00. Припустимий час простою сервісів аналітичних завдань не більше 4-х годин. Завантаження даних з АБС в аналітичну базу (синхронізація) відбувається по закритті "операційного дня" в 23:00. Таким чином, відставання аналітичних даних від АБС становить поточний "операційний день". У випадку втрати даних аналітичних завдань вони повинні бути відновлені на момент останнього закритого "операційного дня". Строк відновлення аналітичних даних залежить від того, у який момент трапився збій у системі. Якщо збій відбувся ранком, то аналітичні дані повинні бути відновлені не пізніше, ніж за 4 години. Якщо збій трапився після обіду або ввечері, то відновлення повинне бути завершено до ранку наступного дня, плюс в аналітичну базу повинні бути завантажені дані останнього "операційного дня". Для виконання OLTP- і DSS-завдань використовується СУБД ORACLE 8i.

Аналізуючи бізнес-вимоги до інформаційної системи з наведеного приклада, виходить, що СЗД банку повинна забезпечувати роботу двох типів завдань – OLTP і DSS. Дані OLTP-завдання є критичними для системи в період з 6:00 до 23:00 (ураховується завантаження даних з АБС в аналітичну базу) і повинні забезпечувати високу готовність (простий не більше 1 години). Вимоги по надійності також високі – втрати не більше півгодини роботи. Навпроти, дані DSS-завдання не настільки критичні й вимоги по готовності не настільки високі, але повинна бути забезпечена висока надійність – втрати не припустимі.

У попередньому розділі були зазначені методи забезпечення доступності СЗД: дублювання апаратних компонентів і дублювання даних, що включає застосування різних рівнів RAID, резервне копіювання й реплікацію в резервний центр.

У наведеному прикладі ІС банку можна рекомендувати використовувати RAID 1+0 для файлів даних і журналів транзакцій OLTP-завдання, при цьому необхідно розташувати файли даних і журнали транзакцій на різних LUN. Така схема дозволить управляти продуктивністю (якщо використовуваний масив може управляти кеш-пам'яттю для окремих LUN) і забезпечить високу надійність зберігання даних. Для даних DSS-завдання рекомендується використовувати RAID 5. Цього цілком достатньо для надійного зберігання даних DSS-завдання й продуктивності при читанні даних. Для відказостійкого доступу до даних у серверах АБС необхідно встановити як мінімум по 2 FC-HBA й підключити їх до різних контролерів дискового масиву. При цьому компоненти дискового масиву повинні бути продубльовані, а ділянки кеш-пам'яті, використовувані для операцій записи, задзеркальовані й захищені від збоїв живлення.

Про вибір масиву певного класу й з певними характеристиками мова йтиме в наступному розділі. Але необхідно відзначити, що не обов'язково всі компоненти масиву повинні бути дубльовані, якщо за зазначений припустимий період простою вони можуть бути замінені, а дані при необхідності відновлені з резервних копій.

Тип додатка впливає на те, як буде здійснюватися резервне копіювання. Наприклад, для резервного копіювання СУБД ORACLE засобами Recovery Manager (RMAN) рекомендується використовувати окремий сервер (а, отже, і окремий екземпляр бази даних і дисковий простір для нього), де буде розміщений RMAN Recovery Catalog. Для резервного

копіювання файлових систем цього не потрібно. Щоб відновити базу даних ORACLE необхідно мати копії журналів транзакцій, для чого рекомендується активізувати в СУБД ORACLE режим архівування журналів транзакцій (ARCHIVELOG). Для архіву журналів транзакцій буде потрібно виділити дисковий простір. Для його захисту від руйнування рівня RAID 5 буде досить. Який тип резервного копіювання використовувати (повне або інкрементальне) залежить від того, за який час можна буде скопіювати базу даних і журнали транзакцій зі стрічок на диски, і чи укладається повне резервне копіювання у відведене тимчасове вікно. Використання повного щоденного резервного копіювання дозволяє відновити базу швидше, ніж, наприклад, застосування повного щотижневого й щоденного інкрементального. При розрахунку часу відновлення СУБД ORACLE рекомендується врахувати, що дані з стрічки копіюються на диски повільніше, ніж записуються з дисків на стрічки, оскільки треба записувати метадані файлової системи [2]. Також треба врахувати, що після копіювання файлів даних і журналів транзакцій зі стрічок на диски СУБД ORACLE повинна буде виконати процедуру відновлення бази – RECOVERY, тобто "накотити" всі незавершені транзакції з журналів.

У наведеному прикладі ІС банку журнали транзакцій необхідно буде копіювати щопівгодини. Якщо зміни в базі даних АБС не великі (за день здійснюється невелике число фінансових транзакцій), то процедура RECOVERY буде виконуватися швидко (не більше десятка хвилин). Якщо база даних і журнали транзакцій можуть бути скопійовані зі стрічок на диски менш чим за 50 хвилин, досить буде робити повне резервне копіювання бази раз у добу після закриття "операційного дня". Але якщо ці умови не виконуються, то буде потрібно використовувати більше складні технології, такі як Storage Checkpoint, реалізовані в VERITAS DBE, або засобу створення PIT-копій.

Як говориться, збій збою ворожнеча. До цього моменту обговорювалися методи відновлення даних після "незначних" збоїв – руйнування даних у результаті відмов окремих елементів апаратури (дисків, контролерів і т.п.), помилок ПЗ або дій персоналу/користувачів системи. Збій у роботі системи може відбутися через аварію, причиною якої може бути вихід з коштуючи технічного засобу цілком (дискового масиву або сервера) або, що ще гірше, техногенна (пожежа, затоплення) або природна (землетрус, повінь) катастрофа. Від того, які вимоги пред'являються до СЗД по строках відновлення після аварій, застосовуються різні схеми резервування даних, що впливає на вибір технічних і програмних засобів і, в остаточному підсумку, на вартість рішення.

У прикладі з ІС банку, якщо потрібно забезпечити відновлення працездатності АБС протягом 1 години після техногенної катастрофи (наприклад, пожежі) у приміщенні комплексу, то, у резервному центрі необхідно крім серверів мати резервний дисковий масив, дублікати стрічок з резервними копіями й стрічковою бібліотекою достатньої потужності. Якщо обсяги даних АБС і інтенсивність їхніх змін не великі, то постійне резервне копіювання й регулярне перевезення дублікатів стрічок у резервний центр буде достатнім для захисту даних від аварії. Але у випадку більших обсягів даних (сотні гігабайт) і частих змінах у базі (великій кількості фінансових транзакцій) зазначена схема не ефективна, особливо, якщо врахувати, що резервне копіювання дає істотне додаткове навантаження на апаратний комплекс і, отже, може привести до неприпустимого зниження його продуктивності. Альтернативним рішенням у наведеному прикладі може служити передача даних між масивами, розташованими на різних площадках, що реалізується за допомогою реплікації або дзеркалювання.

Реплікація може виконуватися програмним забезпеченням, установленим на серверах (програмна реплікація), або убудованим у дискові масиви програмним забезпеченням "прозора" для серверів (апаратна реплікація). Для організації реплікації або дзеркалювання необхідно створити високопродуктивну інфраструктуру передачі даних, об'єднуючу обидві площадки. Як правило, для цього застосовують SAN [1].

При розробці проекту СЗД для збереження вкладених інвестицій необхідно врахувати наявні технічні засоби й найбільш оптимальним образом вписати їх у нову систему. Треба



також врахувати наявність, навички й досвід обслуговуючого персоналу. Якщо кваліфікації персоналу недостатньо для обслуговування високотехнологічної системи, то крім організації навчання, ретельного пророблення регламентів збірки детальних інструкцій і іншої робочої документації, необхідно впровадити систему керування СЗД. Система керування не тільки полегшить роботу, але ще й знизить імовірність помилкових дій персоналу, які можуть привести до втрати даних.

Не секрет, що наявність плану розвитку системи полегшує її впровадження й знижує витрати на модернізацію, у порівнянні з хаотичним розвитком. З іншого боку, бізнес піддається змінам – міняються завдання, з'являються нові користувачі й т.п. Тому, навіть добре спроектована СЗД згодом перестане задовольняти бізнес-вимогам.

Допомогти в рішенні визначених проблем може знову ж впровадження все тої ж системи керування, що дозволить відстежити зміну навантажень на систему зберігання, врахувати утилізацію дискового простору, спрогнозувати потреби в його нарощуванні й т.д.

Підсумовуючи вище сказане одержуємо, що для визначення необхідних параметрів СЗД і підтримки їх у заданих рамках необхідно чітко визначити класи, режими роботи й характеристики завдань, роботу яких повинна забезпечити СЗД, необхідний обсяг зберігання даних і його можливий приріст, кількість і платформи (UNIX, Windows і ін.) серверів, що підключаються. Іншими словами, при створенні СЗД проектувальник повинен виходити від завдань і бізнес-вимог замовника.

### **Розробка структурної схеми**

У попередньому розділі обговорювалися питання забезпечення продуктивності, доступності, масштабованості, оптимізації й експлуатації СЗД. Виходячи із цього, можна визначити, якими властивостями повинен володіти масив, щоб забезпечити рішення зазначених завдань. Вимоги наявності в масиву певних властивостей або характеристик можна розділити на категорії. Однак та саме властивість масиву може попадати в різні категорії, оскільки воно може бути використане для рішення різних завдань.

Структурна схема системи зображена на рисунку 1.

Приведемо перелік вимог, що часто зустрічається на практиці, але не претендує на повноту, розбитий по структурним категоріям.

Структурні вимоги:

- Ємність "сирого" (raw), тобто без розмітки на рівні RAID, дискового простору масиву повинна становити N ТБ. Якщо Вам зустрілася вимога до дискового обсягу масиву в такому формулюванні, то це означає, що планування disk layout ще не проводилося. У протилежному випадку формулювання було б інше: стільки-то дисків того-то обсягу й такий-то швидкості обертання, стільки-то дисків іншого обсягу й т.д.

- Число LUNs, підтримуваних дисковим масивом. Дана вимога можна чітко сформулювати знову ж тільки після планування disk layout. Але число необхідних LUN можна "грубо" поррахувати по числу серверів, що підключаються до дискового масиву, з обліком виконуваних ними класів завдань. Наприклад, сервер ORACLE – 3 LUNs (дані, журнали, архів журналів), файл-сервер – 1 LUN, сервер sendmail – 2 LUNs (файли й журнал файлової системи) і т.п.

- Число серверів, що підключаються, і платформи серверів, що підключаються.

- Можливість створення PIT-копії даних засобами масиву. Дана функціональність масиву може знадобитися, якщо, наприклад, прийняте проектне рішення про завантаження даних з OLTP-завдання в DSS-завдання засобами масиву. Функція створення PIT-копій може бути реалізована різними методами – через "моментальний знімок" (SnapShot) або через повне копіювання даних (clone). Різниця між цими методами полягає в тому, що SnapShot заощаджує дисковий простір, оскільки для його створення потрібно всього лише місце для бітової карти й деякого пула для збереження старих значень змінених блоків. Навпроти, clone вимагає того ж (корисного) обсягу, що й копіюємий LUN. Однак, якщо вихідний LUN підданий частим змінам, те необхідний для підтримки SnapShot обсяг дискового простору може істотно зрости. Якщо з копією LUN, створеної за допомогою SnapShot буде вестися

інтенсивна робота (велика кількість запитів на уведення-вивід), це може знизити продуктивність обміну даними з вихідним LUN. Копія LUN за допомогою SnapShot створюється моментально (звідси й назву – "моментальний знімок"), оскільки процес "копіювання" полягає тільки в створенні бітової карти. Для створення clone потрібне певний час, оскільки відбувається повне копіювання блоків даних. У цей момент навантаження по уведенню-виводу на копіюємий LUN істотно зростає. Існують проміжні способи створення PIT-копій, коли спочатку створюється SnapShot, а потім він поступово перетворюється в clone. Проектувальник повинен урахувати всі ці особливості методів створення PIT-копій і у вимогах чітко вказати який метод планується використовувати.

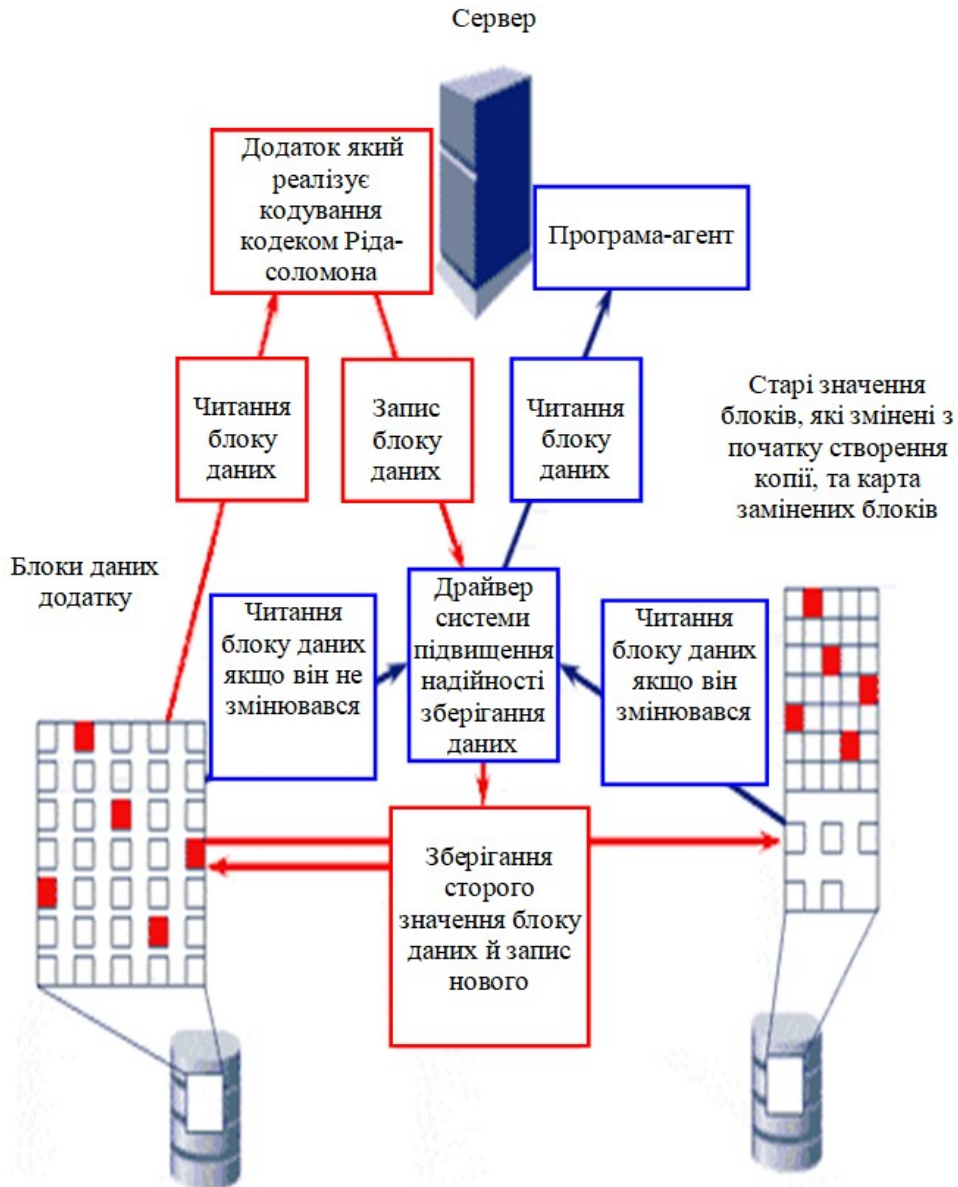


Рисунок 1 – Структурна схема системи

Вимоги до продуктивності:

– Дисковий масив повинен забезпечувати продуктивність  $N$  IOPS, а агрегована пропускна здатність масиву повинна становити  $M$  МБ/с. Як ми вже відзначали, одержати такі цифри не просто. Якщо існує прототип системи або вибір дискового масиву здійснюється для модернізації існуючої СЗД, то можна провести "натурні" виміри продуктивності й апроксимувати їх для очікуваного росту навантаження на СЗД. Якщо система створюється з "нуля", то можна спробувати одержати ці цифри як вимоги виробника прикладного ПЗ (що практично не реально) або звернутися до виробників масивів, щоб ті провели визначення

необхідних параметрів масиву (sizing). Звичайно виробники мають методики "грубої" оцінки необхідної продуктивності. Але вхідними даними для цих методик, як правило, служать очікуване число транзакцій і їх "вага" (light, medium, heavy), які теж не завжди можна визначити.

– Специфічні функції керування кеш-пам'яттю масиву. Наприклад, до таких функцій ставляться:

- можливість закріплення ділянки кеш-пам'яті за конкретним LUN (може придатися для розміщення в кеш часто використовуваних службових таблиць бази даних);
- відключення використання кеш на запис і/або читання для конкретного LUN (може знадобитися для DSS-завдань);
- забезпечення рівня сервісу у вигляді заданого рівня продуктивності (IOPS) або пропускної здатності (МБ/с) для зазначеного сервера.

Вимоги по відказостійкості й надійності зберігання даних:

- Підтримка потрібних рівнів RAID. Як правило, це рівні 1, 0+1, 1+0 і 5.
- Наявність дисків "гарячої заміни" (hot-spare). Механізми використання hot-spare дисків можуть бути різні. Наприклад, можливий варіант, коли у випадку відмови диска дані з дисків порушеної RAID-групи копіюються на hot-spare диск. Але також можливий варіант, коли немає спеціально виділеного hot-spare диска – всі диски містять дані, але при цьому на всіх дисках виділена резервна область, куди копіюються дані з ушкодженою RAID-групи. Визначення необхідного методу знову ж за проєктувальником.

– Захист ділянок кеш-пам'яті, що обслуговують операції запису. За винятком тих випадків, коли відключений кеш на запис, сервер одержує підтвердження завершення операції запису відразу після влучення даних у кеш-пам'ять ще до запису їх на диск. Для забезпечення цілісності даних звичайно застосовуються наступні методи:

- Дзеркалювання ділянок кеш-пам'яті, що обслуговують операції запису.
- Підтримка батареями кеш-пам'яті в плинні N годин або збереження її вмісту на диски у випадку відключення зовнішнього живлення. Який із зазначених варіантів визначити у вимогах – завдання проєктувальника.

– Дублювання всіх компонентів і відсутність єдиної точки відмови (SPOF). Ступінь важливості цієї вимоги залежить від режиму роботи системи й вимог до доступності сервісів. Однак, не треба забувати, що сам масив є SPOF, якщо він не задубльований іншим масивом.

– Можливість створення PIT-копій даних для використання їх у системі резервного копіювання. У ряді систем, де обробляються більші обсяги даних (терабайти), а сервіси повинні бути доступні 24x7 при більших навантаженнях, необхідно застосовувати Serverless резервне копіювання. Для цього використовується механізм створення PIT-копій засобами дискового масиву.

Вимоги по обслуговуємі:

- Можливість заміни компонентів масиву "на ходу" без зупинки системи. Виконання цієї вимоги важливо для систем, що працюють у режимі 24x7.

Вимоги по масштабованості:

– Нарощування дискового простору до N ТБ без заміни раніше встановлених дисків. Таке формулювання дозволяє "убити двох зайців" – забезпечити необхідну функціональність СЗД при росту обсягів оброблюваних даних і зберегти зроблені інвестиції. Тут може бути додана вимога: "без втрати продуктивності". Архітектура масиву може стати "вужьким місцем" і привести до того, що при черговому додаванні дисків продуктивність масиву істотно знизиться, що вплине на рівень якості сервісу.

– Розширення розміру LUN шляхом додавання нових дисків без руйнування збережених даних. Це вимога важливо не тільки для систем, що працюють у режимі 24x7, але також коли є дефіцит кваліфікованого персоналу, здатного здійснити розширення дискового простору при відсутності в масиву даної функції. Бажано, щоб операційна система, дані якої зберігаються на розширюваному LUN, могла автоматично розширити свою файлову систему.

- Збільшення числа серверів, що підключаються, до N.
- Збільшення обсягу кеш-пам'яті до N ГБ без заміни раніше встановлених модулів.

Вимоги по керованості:

- Керування політикою використання кеш-пам'яті для різних LUN. Може знадобитися при "тонкому" налаштуванні масиву.
- Наявність засобів збору статистики про роботу масиву.
- Наявність убудованих засобів оптимізації роботи масиву. Це досить специфічна вимога, однак, наявність таких засобів може допомогти, коли буде потрібно оптимізація, а кваліфікованого персоналу, здатного її виконати, не буде.
- Інтеграція засобів управління масиву із уже розгорнутою системою керування, наприклад HP OpenView.

Щоб не порівнювати всі існуючі на ринку масиви, було б зручно розбити їх на класи. Тоді на основі отриманих вимог можна вибрати потрібний клас і вже порівнювати масиви тільки цього класу. Класи масивів придумувати не треба, вони вже визначені самим ринком, це: початковий клас (low-end), середній клас (mid-range) і вищий клас (high-end).

Масиви зазначених класів відрізняються, у першу чергу, не кількісними характеристиками, а функціональністю й архітектурою. До функціональності low-end масивів можна віднести підтримку різних рівнів RAID і можливість дублювання контролерів (якщо це не JBOD). Від масивів класу mid-range уже потрібна підтримка LUN-masking і створення PIT-копій. А в масивах класу high-end на додаток до зазначених можливостей також реалізовані апаратна реплікація, підтримка OS/390 (zOS) і керування якістю сервісу (на рівні продуктивності в IOPS або пропускну здатності в Мбайт/с).

Але все-таки основним критерієм, по якому можна віднести масив до одного із класів mid-range або high-end, є архітектура. Багато виробників заявляють, що mid-range масиви мають модульну архітектуру, а high-end масиви – монолітну. Це не зовсім вірно. Модульна або монолітна "архітектура" говорить про конструктивний масив – збирається з окремих блоків або шаф. У дійсності архітектуру всіх mid-range масивів (і багатьох low-end) можна характеризувати як "двоконтролерну із загальною шиною".

Для high-end масивів характерна що комутирується або матрична архітектура. Очевидно, що в даній архітектурі ні "вузьких місць", тоді як в mid-range архітектурі вузькими місцями є: контролер, оскільки кожний контролер обслуговує свої RAID-групи (набір дисків, на яких реалізований один рівень RAID), шина між контролерами, обмежене число FC-AL петель до дисків, розташування дисків RAID-групи "уздовж" однієї петлі FC-AL. В high-end масивах RAID-групи розташовуються "поперек" FC-AL петель. Наприклад, в high-end масивах Hitachi RAID-група складається з 4-х або восьми дисків, де кожний диск підключається до двох різних петель від двох різних дисків контролерів. Така конфігурація дозволяє виконувати операції запису-читання із всіх дисків RAID-групи паралельно, чого не можна домогтися в mid-range масивах, коли диски однієї RAID-групи розташовані уздовж однієї петлі й доступ до них здійснюється по черзі.

Іноді ще для high-end масивів говорять про "cache-centric" архітектуру, підкреслюючи тим самим, що центральною ланкою є кеш-пам'ять, до якого мають доступ всі контролери масиву, тоді як в mid-range масивах кеш-пам'ять жорстко прив'язаний до певного контролера.

Зазначені відмінності в архітектурі приводять до втрати продуктивності при масштабуванні mid-range масивів, чого не спостерігається в high-end масивів при додаванні нових дисків. Хоча сучасні mid-range масиви мають високі характеристики масштабованості: дозволяють установлювати до двох-трьох сотень дисків, розподіляючи їх по декількох FC-AL петлях, а також нарощувати кеш-пам'ять до 8 ГБ, все-таки "вузьким місцем" залишається їхня архітектура, що є обмежником масштабованості.

Якщо дотримуватися зазначеної класифікації, то тільки масиви HDS сімейства 9900 і масиви EMC сімейства Symmetrix можна віднести до класу high-end. Масиви HP EVA і IBM ESS 800 (Shark), які позиціонуються виробниками як high-end масиви, мають архітектуру, типову для mid-range масивів.

## Віртуалізація

Останнім часом у маркетинговій літературі всі частіше зустрічається поняття "віртуалізації в системах зберігання", що визначається як приховання від серверів фізичного розташування даних на дисках і подання всього дискового простору як якогось загального пула блоків.

Цей пул уже, у свою чергу, "нарізається" на логічні (віртуальні) диски (Logical Unit Number – LUN), які "видні" серверам як фізичні. У дійсності подібну організацію дискової пам'яті давно вже дозволяє створювати менеджер томів VERITAS Volume Manager. Даний тип віртуалізації одержав назву "host-based" віртуалізація.

Практично всі сучасні дискові масиви виконують функцію створення з наборів фізичних дисків логічних дисків (LUN), що одержала назва "disk array-based" віртуалізація. Це легко визначити на підставі того факту, що ряд масивів підтримують число LUNs більше, ніж фізичних дисків, як, наприклад, у недавно анонсованому масиві Sun StorEdge 3510. Питання полягає в тому, наскільки це зручно. Адміністратори воліють мати можливість управляти фізичним розміщенням файлів даних СУБД ORACLE по фізичних дисках для досягнення оптимальної продуктивності й відказостійкості. Налаштування СУБД під оптимальну продуктивність можуть стати проблематичної, якщо контролер дискового масиву не дозволяє управляти розміщенням RAID-груп на конкретних дисках.

Крім зазначених двох типів віртуалізації – "host-based" і "disk array-based", існує ще один тип – "SAN-based". У цьому типі віртуалізації приховання від серверів фізичного розташування даних здійснюється або за допомогою спеціальних пристроїв, розташованих між FC-комутаторами SAN ("in-band" віртуалізація), або засобами самих FC-комутаторів, що зчитують інформацію про конфігурацію віртуального дискового простору із зовнішнього пристрою ("out-off-band" віртуалізація).

У цей час продуктів "SAN-based" віртуалізації на ринку мало й говорити про їхнє промислове впровадження поки не доводиться. Можливо, потреба в цьому типі віртуалізації з'являться тоді, коли обсяги даних підприємств зростуть настільки, що для їхнього оперативного зберігання не буде вистачати декількох high-end дискових масивів.

**Висновки.** У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів підвищення надійності зберігання даних у хмарі. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем підвищення надійності зберігання даних у хмарі; Досліджена система підвищення надійності зберігання даних у хмарі; На основі отриманих результатів досліджень створена програмна реалізація системи підвищення надійності зберігання даних у хмарі. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання підвищення надійності зберігання даних у хмарі. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

## Список літератури

1. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
2. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings, Volume 3530*, 2023, pp. 256-265.
3. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
4. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
5. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer,

- Cham. pp 169-184.
6. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
  7. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.
  8. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
  9. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.
  10. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136.
  11. Smirnov, O., Drievieva, H., Drieviev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.
  12. Smirnov, O., Drievieva, H., Drieviev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.
  13. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.
  14. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
  15. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.
  16. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.
  17. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
  18. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
  19. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.
  20. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.
  21. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.