

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Методи та засоби побудови білінгвових систем
з відмовостійкою архітектурою

Виконав: студент 6 курсу груп СІм-61
_____, и _____
спеціальності _____

123 «Комп'ютерна інженерія»

(шифр і назва спеціальності (напряму підготовки))

_____	Тимчук М.І.
(підпис)	(прізвище та ініціали)
Керівник	Баран І.О.
_____	_____
(підпис)	(прізвище та ініціали)
Нормоконтроль	Тиш Є.В.
_____	_____
(підпис)	(прізвище та ініціали)
Завідувач	Осухівська Г.М.
кафедри	_____
_____	_____
(підпис)	(прізвище та ініціали)
Рецензент	Скоренький Ю.Л.
_____	_____
(підпис)	(прізвище та ініціали)

м. Тернопіль – 2020

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« » 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 123 Комп'ютерна інженерія
студенту Тимчуку Миколі Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та засоби побудови білінгових систем
з відмовостійкою архітектурою

Керівник роботи Баран Ігор Олегович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від «28» вересня 2020 року № 4/7-687

2. Термін подання студентом роботи 21.12.2020

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 Аналітична частина. 2 Теоретична частина. 3. Практична реалізація.

4 Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема, мета, задачі, об'єкт, предмет дослідження. 2. Актуальність. 3. Робота білінгової

мережі в реальному часі. 4 Порівняльний аналіз технологій. 5. Топології Oracle Golden Gate

6. Архітектура Oracle Golden Gate. 7. Відмовостійка архітектура бази даних білінгової системи

8. Режими захоплення змін. 9. Створення екземпляру бази даних. 10. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., доцент	01.11.20	07.12.20
Безпека в НС	Стадник І. Я., професор	01.11.20	09.12.20

7. Дата видачі завдання _____ 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи	28.09.20	Виконано
2	Аналіз літературних джерел	29.09.20-15.10.20	Виконано
3	Обґрунтування актуальності дослідження	15.10-21.10.20	Виконано
4	Аналіз предмету дослідження та предметної області	21.10-30.10.20	Виконано
5	Проведення дослідження методів та засобів аналітичного опрацювання даних	30.10-05.11.20	Виконано
6	Оформлення розділу «Аналітична частина»	05.11-12.11.20	Виконано
7	Оформлення розділу «Теоретична частина»	12.11-22.11.20	Виконано
8	Оформлення розділу «Практична частина»	22.11-10.12.20	Виконано
9	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»	26.11-12.12.20	Виконано
10	Нормоконтроль	11.12-14.12.20	Виконано
11	Попередній захист роботи	15.12.20	Виконано
12	Захист кваліфікаційної роботи	23.12.20	

Студент

(підпис)

Тимчук М.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Баран І. О.

(прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби побудови білінгових систем з відмовостійкою архітектурою
// Кваліфікаційна робота за освітнім рівнем «магістр» // Тимчук Микола Іванович
// Тернопільський національний технічний університет імені Івана Пулюя,
факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра
комп'ютерних систем та мереж, група СІм-61 // Тернопіль, 2020 // с. – 67, рис. –
33, табл. – 5, аркушів А1 – 10, бібліогр. – 25.

Ключові слова: БІЛІНГОВА СИСТЕМА, ВІДМОВОСТІЙКА
АРХІТЕКТУРА, РЕПЛІКАЦІЯ, ЕКЗЕМПЛЯР БАЗИ ДАНИХ

Кваліфікаційна робота присвячена дослідженню методів та засобів побудови відмовостійкої архітектури для білінгових систем. Проведено порівняльний аналіз існуючих програмних середовищ та технологій для вирішення завдання, як результат обрано Oracle Golden Gate. Наведено особливості встановлення Oracle Golden Gate з конфігурацією вихідної і цільової баз даних. Докладно описані програмні механізми реплікації баз даних, приведено процедуру створення екземпляру бази даних Oracle. Також запропоновано способи усунення проблем та конфліктів, які можуть виникнути при використанні розробки. Проведено попереднє налаштування Active-Active реплікації.

Дана робота охоплює ряд важливих аспектів, таких як висока доступність і максимальний захист даних, відмовостійкість, продуктивність, зниження витрат на розгортання, управління і підтримку системи.

ANNOTATION

Methods and tools of billing systems building based on fault-tolerant architecture
// Master thesis // Tymchuk Mykola// Ternopil Ivan Pul'uj National Technical
University, Faculty of Computer Information Systems and Software Engineering,
Department of Computer Systems and Nets, group CIm - 61 // Ternopil, 2020 // p. – 67,
fig. – 33 , table. – 5 , Sheets A1 - 10 , Ref. - 25.

Keywords: BILLING SYSTEM, FAULT-TOLERANT ARCHITECTURE,
REPLICATION, DATA BASE INSTANCE

The thesis deals with the study of methods and means of constructing a fault-tolerant architecture for billing systems. A comparative analysis of existing software environments and technologies to solve the problem was carried out. As a result, Oracle Golden Gate was selected. Features of installation of Oracle Golden Gate with a configuration of initial and target databases are resulted. The software mechanisms of database replication are described in detail, the procedure for creating an instance of the Oracle database is given. There are also ways to resolve issues and conflicts that may arise when using the development were proposed. Pre-configured Active-Active replication was done.

This research work covers a number of important aspects, such as high availability and maximum data protection, fault tolerance, productivity, reduced deployment costs, management and system maintenance.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

BI (Business Intelligence) – бізнес- аналіз даних

DWH (Data WareHouse) – сховище даних

MTTF (Mean Time To Failure) - середній час до відмови

MTTR (Mean Time To Recovery) - середній час відновлення

ODI – Oracle Data Integrator

OGG – Oracle Golden Gate

OLTP (Online Transaction Processing) – обробка транзакцій в реальному часі

ROI (Return On Investment) – коефіцієнт рентабельності інвестицій

TCO (Total Cost of Ownership) - сукупна вартість володіння активом

БД – база даних

БС – білінгова система

Відмовостійкість – властивість технічної системи зберігати свою працездатність після відмови одного або декількох складових компонентів

Гетерогенна система - інформаційна мережа, в якій присутні компоненти ІС різних фірм-виробників, різнотипних технічних засобів або топологій

Екземпляр БД - це засіб доступу до БД Oracle, що складається з набору фонових процесів і пам'яті, яку вони використовують

ІС – інформаційна система

ОС – операційна система

ПЗ – програмне забезпечення

СППР – система підтримки прийняття рішень

Реплікація (Replication) – механізм синхронізації вмісту декількох копій об'єкта (наприклад, вмісту БД)

РЧ – реальний час

СУБД – система управління базами даних

ЦОД – центр обробки даних

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ	6
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА	12
1.1. Роль білінгу в ІС.....	12
1.2. Аналіз основних понять досліджуваної області	16
1.3. Огляд існуючих технологій.....	20
1.4. Висновки до розділу	23
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА	24
2.1. Характеристика середовища OGG	24
2.1.1. Ключові можливості і технологічні відмінності.....	25
2.1.2. Використовувані топології.....	25
2.1.3. Проміжне ПЗ Oracle для BI і ODI.....	26
2.1.4. Застосування для сховищ даних в режимі РЧ.....	27
2.1.5. Використання для СУБД Oracle	28
2.1.6. Відмовостійкість	29
2.1.7. Вимоги до системи.....	30
2.2. Огляд архітектури OGG.....	30
2.2.1. OGG Capture.....	32
2.2.2. OGG Trail Files.....	32
2.2.3. OGG Delivery	33
2.2.4. OGG Manager	34
2.3. Встановлення OGG	35
2.4. Конфігурація вихідної і цільової Oracle БД	37
2.5. Висновки до розділу	38
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ	39
3.1. Архітектура БД БС.....	39

3.2. Захоплення змін та реплікація	40
3.3. Створення OGG екземпляру	47
3.4. Вирішення проблем.....	51
3.5. Попереднє налаштування Active-Active реплікації	54
3.6. Висновки до розділу	56
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	57
4.1. Охорона праці	57
4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки	60
4.3. Висновки до розділу	62
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
ДОДАТОК А. Тези конференції	

ВСТУП

Актуальність теми. В даний час основний підхід до створення інфраструктури високої доступності полягає в розгортанні надлишкового та такого, що часто простоє, обладнання та комплектуючих, ПЗ від різних постачальників. Цей підхід часто пов'язаний з більшими витратами, але не забезпечує очікуваного рівня надійності через слабку інтеграцію компонентів, технологічних обмежень і складності адміністрування. Крім того, багато підприємств проходять через процес збільшення своєї інфраструктури ІТ для підвищення бізнес-ефективності, тому консолідація БД є важливою частиною цього процесу.

На даному етапі, в реаліях глобальної економіки, неможливо досягти консолідації ІС тільки шляхом об'єднання сотень БД в кілька більших або розміщення їх в одному ЦОД чи поєднанням обох рішень. Такі технології, як Oracle Active Data Guard, Oracle RAC і OGG часто використовують для задоволення потреб в аварійному відновленні і реплікації даних. Розподіл робочого навантаження за декількома екземплярами БД може підвищити продуктивність, масштабованість і ефективність ІС, а також цінність ІТ-активів.

Готове рішення повинно бути в першу чергу гнучким, так як у різного роду бізнес-функцій, додатків і користувачів різні вимоги до рівня обслуговування, які до того ж змінюються зі зміною умов ведення бізнесу.

Поставлена в рамках роботи задача проектування, документування і реалізації архітектури високої доступності для білінгових систем вкрай актуальна на сьогоднішній день, так як апаратні помилки, що викликають відмову серверів, є непередбачуваними, викликають припинення бізнес-процесів, що завдає істотної шкоди бізнесу: його позитивній репутації, прибуткам, лояльності клієнтів і т.д.

Метою дослідження є проектування архітектури максимальної доступності, що задовольняє наступним умовам:

- висока ступінь відмовостійкості;

- вилучення даних в РЧ;
- аналіз і реагування в РЧ;
- стійкість до поганих каналів.

В роботі поставлено та розв’язано **наступні задачі:**

- збору даних і реплікації в РЧ;
- детально вивчити середовище OGG (вимоги до системи, в якій воно може бути розгорнуте, архітектура логічної складової, різні топології в практичному застосуванні);
- спроектувати відмовостійку архітектуру БД БС;
- практично реалізувати (установка та конфігурування ПЗ) з тестуванням отриманих результатів.

Об’єкт дослідження: процес розробки відмовостійкої архітектури.

Предмет дослідження: технології та середовища для побудови відмовостійкої архітектури.

Наукова новизна отриманих результатів.

- проведено ґрунтовний порівняльний аналізу існуючих програмних середовищ та технологій для вирішення завдання;
- спроектовано відмовостійку архітектуру БД з мінімальним навантаженням як на транспортну мережу, так і на серверні машини, на яких система була розгорнута;
- запропоновано способи усунення проблем та конфліктів, які можуть виникнути при використанні БД.

Практичне значення одержаних результатів. Впровадження результатів проведеного дослідження дозволить звести час простою до нуля, знижуючи ризики збитків для бізнесу і гарантуючи доступність корпоративних систем в режимі 24/7.

Публікації. Результати дослідження апробовано на VIII науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського

національного технічного університету імені Івана Пулюя (9-10 грудня 2020 р.) у вигляді опублікованих тез:

Тимчук М.І. Особливості використання Oracle GoldenGate для розробки відмовостійкої архітектури баз даних. Інформаційні моделі, системи та технології: Праці VIII наук.-техн. конф. (Тернопіль, 09-10 грудня 2020 р.) Тернопіль, 2020. – С. 121.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 4 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 67 арк. формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА

1.1. Роль білінгу в ІС

ІС є невід'ємною частиною інфраструктури сучасного підприємства і використовується для управління внутрішніми і зовнішніми ресурсами підприємства (фінансовими, матеріально-технічними, людськими), управління витратами і ризиками, розширення зв'язків з замовниками, бізнес аналітики і т.д., даючи в результаті конкурентну перевагу. Таким чином, організації все більше залежать від своєї ІТ-інфраструктури та її неперервної доступності. Прості додатків і недоступність даних напряду призводять до втрати продуктивності і доходів, розчарування замовників і збитку для репутації.

Варто зауважити, що в мережі Інтернеті та мобільних мережах існує тенденція до появи персоналізованих (розширених) послуг. Вони враховують атрибути (характеристики) та специфічні риси (особливості) користувача.

Таким чином, для білінгу необхідна реалізація певної функціональності [1]. Отже білінг - це об'єднання процесів та рішень в інтернет-організаціях чи організаціях зв'язку, які відповідають за збирання інформації щодо застосування послуг телекомунікації, їх тарифікацію і виставлення абонентських рахунків, та власне, опрацювання всіх видів оплат. Оскільки білінг є найбільш важливою частиною роботи комерційного оператора зв'язку (властиво незалежно від впровадженого виду телекомунікації), то такі оператори точно не можуть існувати без білінгу, завдяки якому фактично є можливою стабільна економічна їх діяльність.

Поняття білінгу включає такі складові частини [2]:

– облік (збір) інформації про послуги, надані абонентам мережі («accounting»);

- тарифікація наданих послуг, іншими словами - розрахунок вартості («charging»);
- процес представлення рахунків абонентам (власне «billing»).

В таблиці 1.1 представлені визначення понять обліку, тарифікації та білінгу, які використовуються в телекомунікаційній мережі (ТКМ) та мережі Internet.

Таблиця 1.1

Порівняння термінологій в ТКМ та Інтернеті

Терміни	Визначення в ТКМ	Визначення в Internet
Облік/Accounting	Процес пропорційного розподілення оплати між середовищем надання послуги, обслуговуючою мережею і користувачем.	Функція, що збирає дані про споживання ресурсів, включає контроль збору даних, транспорт і зберігання даних
Тарифікація/Charging	Функція мережі, за допомогою якої інформація, що дозволяє визначити використання ресурсу і відноситься до оплачуваної події, збирається, приводиться у відповідний формат і передається до білінгової системи, а також здатна монетизувати тарифікаційні параметри і виконувати контроль рахунку абонента під час користування послугою.	Функція, що генерує немонетарну вартість послуги і специфічні параметри, що відносяться до тарифу послуги
Білінг/Billing	Функції, за допомогою якої дані по тарифікації трансформуються у рахунки відповідно до оплати.	Функція, що переводить немонетарні витрати в монетарні одиниці і генерує остаточний рахунок для користувача.

Висока доступність (мінімізація і зведення до нуля недоступності обслуговування і простоїв) і відмовостійкість (постійна доступність, а також

продовження обслуговування при збоях деяких частин ІС) мають першочергове значення для проектування і експлуатації ПЗ БС, підтримують всі бізнес-процеси білінгу. Дані параметри ще важливіші для ПЗ БС, розташованих в хмарних системах [3].

Необхідно також навести одну із схем реалізації білінгової мережі даних (див. рис. 1.1).

Білінгова мережа реалізована так:

- комутатор записує дані про кількість та тривалість з'єднань;
- потім він їх опрацьовує;
- після цього дані поступають в систему для проведення розрахунків з наперед відомими тарифами.

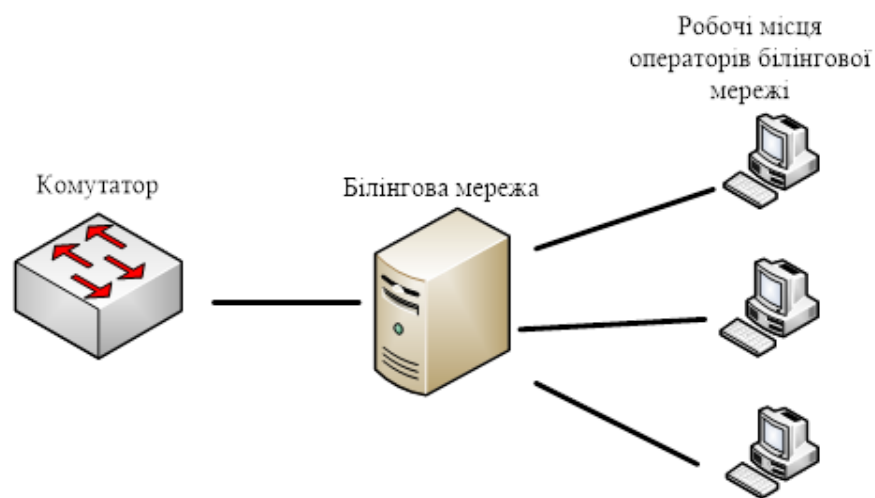


Рис. 1.1. Один з варіантів реалізації білінгової мережі даних

Ця система виконує ідентифікацію виклику та робить необхідні розрахунки, властиво таким чином забезпечуючи формування рахунку абонента за надані послуги. Зрозуміло, що в пам'яті такої мережі повинні міститися не тільки дані щодо тарифів та нормативів, інформація про надані послуги, але також і дані про клієнтів, заключених угодах з абонентами зв'язку чи зі сторонніми постачальниками послуг (при їх наявності). Ще необхідно мати вартість передачі інформації за різними каналами зв'язку.

Крім того, будь-яка власне білінгова мережа потребує наявності БД, в якій буде зберігатися історія всіх платежів для всіх клієнтів. Всі ці дані допомагають управляти процесами оплати та активації/деактивації абонентів. Саме ця функція і є захисною, тому що вона забороняє використання послуг мобільного зв'язку тими особами, які не хочуть платити за них [22].

Таким чином білінг в комунікаційних системах веде роботу в режимі РЧ (див. рис. 1.2). Для збільшення цінності інформації дані необхідно збирати і аналізувати в РЧ для того, щоб пропонувати клієнту актуальну послугу, а не тоді, коли потреба в ній уже може пропасти.

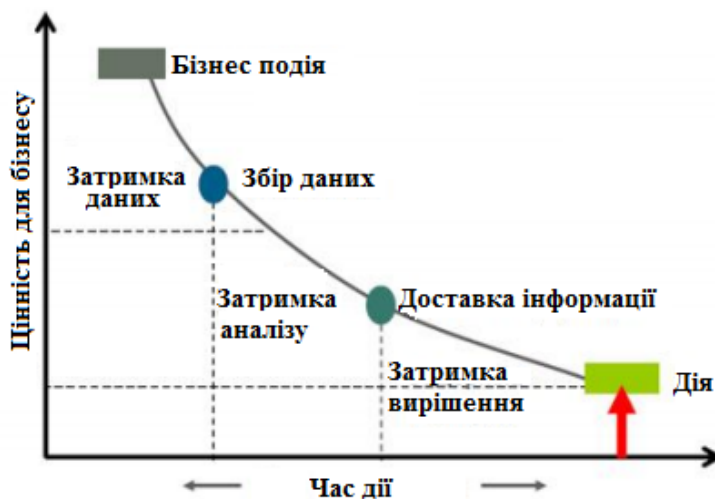


Рис. 1.2. Робота в режимі реального часу

Наявне моментальне списання грошових активів з рахунку абонента за надані йому послуги. Відповідно такий режим роботи вимагає від БС високої надійності ПЗ і стабільної роботи БД, а також й рішення про переходи з стратегічного на операційний рівень ВІ.

Впровадження в існуючу ІС операційного ВІ дозволить вирішити в майбутньому такі завдання, як:

– результат процесу здобування знань про бізнес з перетворених на інформацію даних про ІС;

- оптимізація реклами про пакети послуг від стільникового оператора;
- оптимізація роботи з клієнтом і його утримання в РЧ;
- виявлення шахрайства та поліпшення ланцюжків постачання.

Таким чином, можна виділити дві основні задачі в побудові архітектури:

- рішення під відмовостійкість і високу доступність ІС;
- рішення під оперативний ВІ.

1.2. Аналіз основних понять досліджуваної області

Для подальшого розуміння варто навести основні поняття, які будуть використовуватися в тексті роботи.

Зокрема, корпоративне сховище даних (англ. EDW – Enterprise Data Warehouse) визначається, як предметно-орієнтована інформаційна БД, спеціально розроблена і призначена для підготовки звітів і бізнес-аналізу з метою підтримки прийняття рішень в організації. Будується на базі СКБД і СППР. Дані, що надходять в сховище даних, як правило, доступні тільки для читання.

Тут варто згадати про ETL (англ. Extract, Transform, Load) – дослівний переклад термінів «витяг, перетворення, завантаження»), який є одним з основних процесів при керуванні сховищами даних. Він складається з таких частин: вилучення даних із зовнішніх джерел; здійснення трансформації (перетворення) даних для їх коректного зберігання; виконання завантаження в сховище даних.

Також визначним є поняття бізнес-аналізу даних (англ. ВІ – Business Intelligence). Відповідно до [7], під ВІ мається на увазі процес, який включає наступне:

- перетворення даних в інформацію і знання про бізнес для підтримки прийняття поліпшених і неформальних рішень;
- ІТ (методи і засоби) збору даних, консолідації інформації і забезпечення доступу бізнес-користувачів до знань;

– знання про бізнес, здобуті в результаті поглибленого аналізу детальних даних і консолідованої інформації.

Визначені терміни ВІ та «бізнес-аналітика» дуже часто у літературі застосовуються як слова-синоніми, проте між ними є деякі відмінності. Зокрема, бізнес-аналітика (у вузькому сенсі), на противагу ВІ, властиво опрацьовує вже підготовлені до аналізу дані («очищені»). Також вона використовує статистичні та кількісні інструментальні засоби для обчислення оцінки поточної ситуації та ефективного прогнозування, тому дедалі частіше її називають «поглиблена аналітика».

В роботі вже вживалися значення «несправність» та «відмовостійкість», проте їм не було дано точні визначення. Нижче буде спроба дати більш точне їх визначення і деяких інших використовуваних термінів в області відмовостійкості. Визначення тут засновані на тих, що дані Лапрі [8], Авізенісом і Лапрі [9], Гресем і Рейтером [10].

Коли сконструйована ІС і її модулі, визначено і їх поведінку. Під час роботи ми можемо спостерігати їх поведінку. Момент, коли спостерігається поведінки ІС (або окремих її модулів), що відрізняється від визначеного, називається збоєм. Збій трапляється через помилки, викликані несправністю. Відрізок часу між виникненням помилки і виправленням збою в ІС є затримкою помилки.

Весь процес наведено на рис. 1.3.

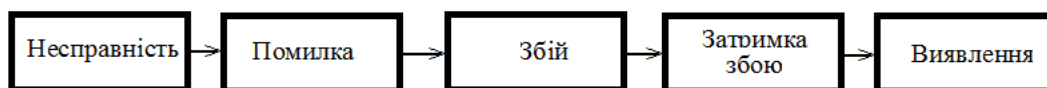


Рис. 1.3. Процес виявлення несправностей

Несправності бувають як апаратними, так і ПО. Надійність модуля вимірює час від початкового моменту до наступної події збою і визначається статистично як МТТФ. Середній час, затрачений на відновлення модуля після виявлення

відмови, називається MTTR. Результатом буде формула (1.1) доступності модуля, як відношення завершення служби до минулого часу:

$$\frac{MTTF}{MTTF + MTTR} \quad (1.1)$$

Не завжди збій ІС викликаний подією, коли щось пішло не так: він може бути викликаний затримкою правильної поведінки компонента ІС - збоєм синхронізації (Timing failure). Це відмова процесу або частини процесу в синхронній розподіленій системі або системі РЧ з дотриманням обмежень, встановлених під час виконання, доставки повідомлень, швидкості дрейфу годин або перекосу годин.

Доступність ІС - це частка пропонованого навантаження, котра обробляється з прийнятним часом відгуку.

У таблиці 1.2 представлена класифікація ІС за різними класами доступності. В даний час, більшість ІС загального призначення працює в класах 3 і 4. Найкращі відмовостійкі системи працюють в класі 5.

Таблиця 1.2

Доступність типових системних класів

Тип ІС	Недоступність (хв/год)	Доступність (%)	Клас
Некерована	52560	90	1
Керована	5256	99	2
Добре керована	526	99,9	3
Відмовостійка	53	99,99	4
Висока доступність	5	99,999	5
Дуже висока доступність	0,5	99,9999	6
Ультра-доступність	0,05	99,99999	7

Для досягнення надійної і доступної ІС можна використовувати два дуже різні підходи: усунення помилок; відмовостійкості.

У той час, як усунення помилок - це запобігання виникнення помилок по конструкції, відмовостійкість - це використання надмірності, щоб уникнути збоїв через несправності. Усунення помилок – це процес важкий і практично неможливий у великих і складних ІС. Це робить відмовостійкість єдиною реалістичною альтернативою класам систем.

Універсальні ІС: універсальні комп'ютери на елітарному комерційними ринку, які застосовують методи відмовостійкості для підвищення загальної надійності.

Високої доступності ІС: системи, призначені для класу доступності 5 або вище, часто є системами БД або комутації телефонних з'єднань. Oracle визначає високу доступність як «обчислювальні середовища, сконфігуровані для забезпечення майже повної готовності». Загальноприйнятим стандартом для високої доступності є «п'ять дев'яток» (99,999% часу безвідмовної роботи).

Довговічні ІС: системи, призначені для роботи впродовж дуже довгого часу без будь-яких проблем з ремонтом. Довговічні системи є типовими мобільними системами, де ремонт на місці є важким або , можливо, неможливо. Прикладами є безпілотні космічні системи, такі як супутники або космічні пошукові машини. Вони відрізняються від інших відмовостійких систем, маючи надмірність не тільки в електричних системах, але і в механічних деталях. Вони також повинні забезпечити правильну роботу протягом тривалих періодів часу.

Критично-обчислювальні ІС: системи, що виконують критичну роботу, де помилкові обчислення можуть мати тяжкі наслідки для людського життя або мати високі економічні наслідки. Прикладами можуть слугувати комп'ютери в космічному шатлі, ядерній установці або системі управління повітряним рухом, де несправність може бути надзвичайно катастрофічною.

Як ІС для свого проектування вибираємо клас високої доступності з підходом відмовостійкості. Необхідно також глибше вникнути в існуючі на

сьогоднішній день методи вирішення поставлених завдань у вибраній області.

1.3. Огляд існуючих технологій

Багато компаній підтримують репліки своїх БД як локально, так і в географічно розподілених ЦОД. Причинами ведення обох типів реплік є:

- безперервність бізнесу і швидке відновлення доступу до даних;
- оптимізація продуктивності для місцевих клієнтів;
- локалізація контенту і кешування;
- адаптація до місцевого законодавства;
- використання ЦОД.

Однак, коли додатки розподілені по різних БД, а також за різними ЦОД, неможливо ефективно використовувати наявні ресурси для забезпечення максимальної продуктивності і доступності, ще не маючи ретельно спроектованої архітектури.

У системі, яка містить кілька реплікованих БД, конкретний сервер БД може викликати повільніший час відгуку через підвищений попит на службу БД або недостатність потужного устаткування і стати вузьким місцем всієї інфраструктури. Тоді як сервери реплік, здатні пропонувати ту ж послугу, можуть бути недостатньо використані.

Підсумковий порівняльний аналіз технологій наведені в таблиці 1.3.

Наведемо основні середовища, які описані в цій таблиці.

Oracle RAC (Oracle Real Application Clusters) - ПЗ для кластеризації і підвищення доступності для БД Oracle. OGG - це високопродуктивне ПЗ для захоплення змін даних, їх перетворення і доставки, забезпечує двосторонню реплікацію на основі журналів СУБД. Oracle ADG (Active Data Guard) - це високопродуктивне ПЗ для створення катастрофостійкої копії БД з можливістю відкриття для читання.

Порівняльний аналіз технологій

Технологія	Переваги	Недоліки
Oracle RAC	Скорочення вартості збільшення масштабу Цілодобова постійна доступність додатків БД	Єдиний кошик Масштабованість і локація виключно всередині локальної мережі
OGG	Мінімальне навантаження на трафік мережі за рахунок вивантаження / доставки / завантаження не самих даних, а змін журналів в РЧ Можливість реплікації даних між різними середовищами (ОС,СУБД) в РЧ Можливість зміни ІС з мінімальними затратами на працю програміста Транзакційна цілісність - стійке до переривання роботи і збоїв Висока гнучкість і масштабованість ІС	Відсутність забезпечення можливості «автоматичної відмови» (як в Oracle ADG)
Oracle ADG	Створення повністю однакових копій вихідної БД на одну або кілька резервних синхронізованих цільових БД Постійне застосування змін, отриманих від вихідних БД, що дозволяє отримувати дані онлайн	Забезпечення повністю ідентичних копій вихідної БД на одну або кілька резервних синхронізованих цільових БД, що створює надмірність даних Постійне застосування змін, отриманих від вихідних БД, що створює додаткове навантаження на промислове середовище Негативний досвід з хмарними технологіями Неможливість використання для роботи одночасно обидві вихідну і цільових перевірок БД, так як цільова БД є запасною для мінімальних тимчасових витрат при аварійному відновленню доступу

Технологія	Переваги	Недоліки
Реплікація	Можливість синхронізації декількох таблиць, а не всієї БД Реплікація між різними версіями БД і навіть між різними платформами	Синхронізація даних з інтервалом від 10 хв Після 16 невдалих спроб передачі даних послідовно процес зупиняється і вимагає ручного перезапуску служби Вирішення побудови даних між різними платформами є значно дорожчим по часу і матеріальним ресурсам
Самостійно розроблене ПЗ	Відносно дешева праця розробника В разі незнання або невміння використовувати існуючі технології	Відносно тривалий час, необхідний на реалізацію Реалізацію нового ПЗ можна порівняти з «винаходом велосипеда» Залежність від програміста у вигляді високих ризиків втрати керованості проектом в разі погано задокументованого ПЗ і супутніх інструкцій по експлуатації Ручне програмування порівняно менш надійне в порівнянні з вже неодноразово перевіреним часом або іншим ПЗ від провідного вендора

Проаналізувавши дані з таблиці 1.3, приходимо до висновку, що найоптимальнішим засобом для вирішення поставленої задачі на проектування буде OGG.

Даний продукт володіє всіма необхідними перевагами, зокрема висока доступність і максимальний захист даних, продуктивність, зниження витрат на розгортання, управління і підтримку системи, а з недоліків – тільки відсутність забезпечення можливості «автоматичної відмови» [15].

1.4. Висновки до розділу

У цьому розділі описано роль білінгу в мережах та ІС. Проаналізовано основні поняття досліджуваної області. Завдяки проведеному ґрунтовному порівняльному аналізу існуючих програмних середовищ та технологій для вирішення завдання на проектування як середовище виконання обрано OGG.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

2.1. Характеристика середовища OGG

У світлі сучасних тенденцій було прийнято рішення про розробку відмовостійкої і високодоступної архітектури для БД в приватній хмарі для захисту критично важливих даних ІС з можливістю реплікації даних в гетерогенних середовищах. Така архітектура вирішує практичні проблеми, пов'язані як з плановими простоями ІС в результаті міграцій БД, оновлення ПЗ, СУБД або ОС сервера, так і в разі несподіваних збоїв внаслідок фізичного пошкодження сховищ, відключення обчислювального майданчика і людських помилок [6].

Найбільш оптимальним рішенням є OGG, котрий забезпечує низькорівневе захоплення, маршрутизацію, перетворення і доставку транзакцій БД в гетерогенних середовищах в майже РЧ, а також дозволяє здійснювати обмін і маніпулювання даними на рівні транзакцій між декількома гетерогенними платформами всередині однієї ІС, шляхом переміщення виконаних транзакцій з журналів повтору, внаслідок чого цілісність транзакції підтримується з затримкою в субсекунди [13].

Саме тому даний продукт був обраний серед безлічі інших для забезпечення рішення постійної доступності, включаючи нульовий час оновлень і міграцій, пряму звітність, оперативну бізнес-аналітику і інтеграцію транзакційних даних системи.

Навіть підприємства, які використовують СУБД Oracle, реалізують свої власні рішення управління розподіленою обробкою даних. Ці рішення не можуть забезпечити повну функціональність, пропоновану вендором, таку як, наприклад, балансування навантаження під час роботи або надійний перехід на інший ресурс у разі збою, оскільки вони не інтегровані в стек СУБД Oracle.

2.1.1. Ключові можливості і технологічні відмінності

Варто навести основні можливості середовища, які і дозволяють використати його для виконання поставленого завдання на проектування.

Отже, OGG - це:

- проміжний продукт, призначений для роботи в гетерогенному середовищі з різними СУБД;
- переміщення тільки зафіксованих даних між різними платформами, що дозволяє використовувати субсекундну затримку;
- можливість переміщення змін по мережі TCP / IP, не вимагаючи наперед встановленого Oracle Net;
- використання власної системи файлів контрольних точок для підтримки цілісності транзакцій і без використання концепції мультиплексування, такої як в БД Oracle;
- швидке переміщення даних в резервну БД, яка може підтримувати аварійне відновлення;
- використання власного порядкового номера (CSN) для визначення транзакції, заснованої на БД Oracle Database SCN.

2.1.2. Використовувані топології

OGG має модульну архітектуру [13], яка забезпечує гнучкість для вилучення і реплікування обраних записів даних, транзакційних змін і змін на мові опису структури даних DDL (Data Definition Language) в різних топологіях (див. рис. 2.1).



Рис. 2.1. Топології OGG

Завдяки такій гнучкості, а також можливості фільтрації, перетворення і призначеної для користувача обробки OGG підтримує багаточисельні бізнес-вимоги, в тому числі:

- відмовостійкість бізнесу і висока доступність;
- первинне завантаження і міграція БД;
- інтеграція даних;
- підтримка прийняття рішень в корпоративних сховищах даних.

2.1.3. Проміжне ПЗ Oracle для BI і ODI

Об'єднання OGG та Oracle Data Integrator (ODI) використовується для доставки даних в режимі РЧ, забезпечуючи швидку інтеграцію даних в РЧ і швидке перетворення об'ємних даних: OGG подає проміжну область сховища даних, а ODI виконує масові перетворення для подачі користувальницьких

таблиць в DW за TCO і високопродуктивне рішення для зберігання даних в РЧ (RTDW) (рис. 2.2).

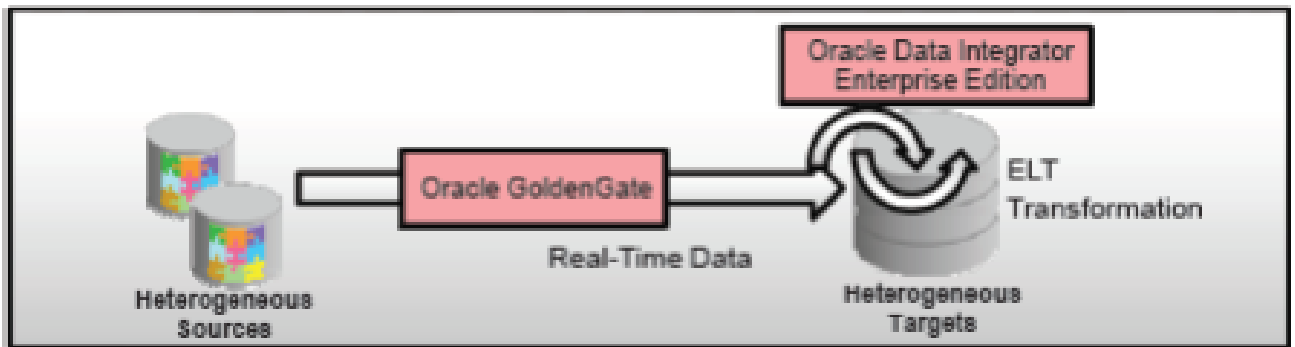


Рис. 2.2. OGG та ODI

2.1.4. Застосування для сховищ даних в режимі РЧ

Для середовищ корпоративних сховищ даних в режимі РЧ Oracle OGG захоплює і приносить змінені дані в сховищі оперативних даних в режимі РЧ. Оскільки рішення базується на журналі роботи, то відбувається мінімальний вплив на джерело, немає пакетних вікон, а переміщення нових транзакцій з вихідної системи досягається за субсекунди (рис. 2.3).

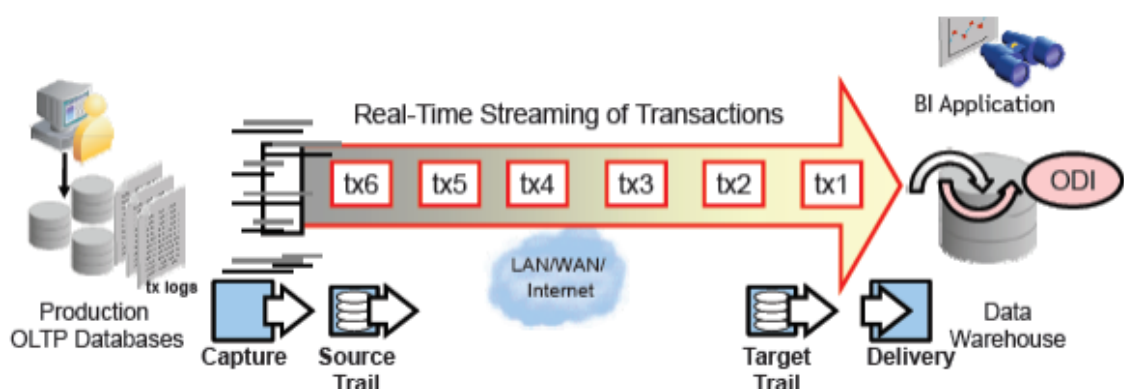


Рис. 2.3. OGG для сховищ даних в режимі РЧ

Під час фази переміщення зберігаються кордони фіксації кожної транзакції для забезпечення цілісності даних. ODI виконує складні перетворення в цільовій

БД для максимальної продуктивності. Іншою перевагою даного підходу є можливість відновлення даних в разі збою при переміщенні даних. Це важлива вимога, оскільки латентність даних зменшується при підтримці аналітичного середовища. Trail файли Oracle OGG, що зберігають змінені дані, є постійними, тому їх можна повторно використовувати (при необхідності) як для цільової так і для вихідної системи.

2.1.5. Використання для СУБД Oracle

Ключовим прецедентом в усуненні запланованого часу простою є оновлення або міграція БД, додатки, ОС або обладнання (див. рис. 2.4). Рішення забезпечує нульовий час простою БД, синхронізуючи БД в різних версіях і забезпечуючи негайне перемикання [14].

Клієнти також можуть використовувати дві версії БД одночасно, щоб зробити оновлення повністю прозорим для користувачів додатків, а також для перевірки узгодженості даних. Також наявна резервна можливість повернутися до старої системи, якщо на то з'явиться необхідність з будь-якої причини.

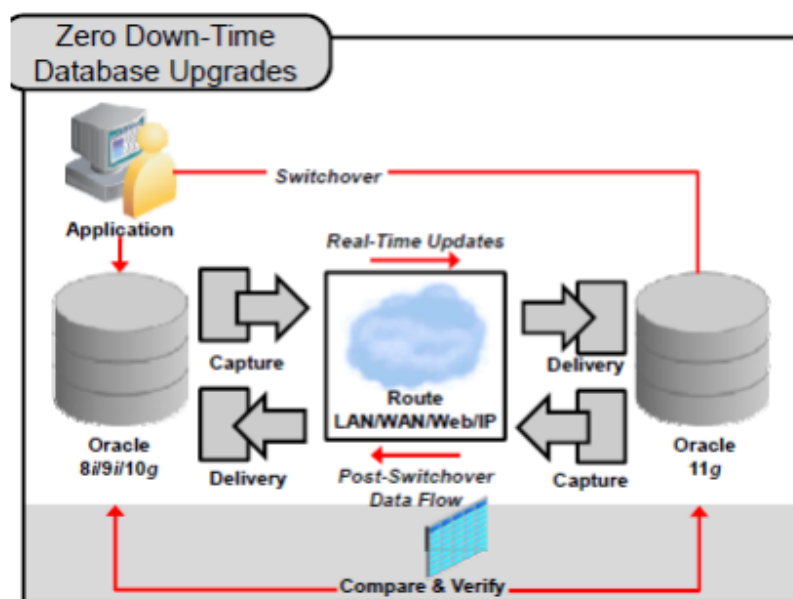


Рис. 2.4. Нульовий час простою під час оновлень і міграцій

2.1.6. Відмовостійкість

Для аварійного відновлення даних Oracle Active Data Guard є життєздатним варіантом тільки для СУБД Oracle. Oracle OGG доповнює цю можливість для платформ, відмінних від Oracle, конфігураціями «Активний-активний», а також крос-OS або крос-БД версіями (див. рис. 2.5). Oracle OGG може підтримувати фізичну резервну, логічну резервну систему або резервну резервну систему.

OGG постійно синхронізується систему резервних БД в режимі РЧ, щоб в разі потреби миттєво переключитися на резервну систему. Вторинна система відкрита для використання «тільки для читання» так само як і «для запису» [6].

Також надається перемикання потоку даних із запасних БД (standby) на вихідну відразу ж після перемикання з режиму очікування в основний. Будь-які дані, які обробляються на запасній БД (Standby) в режимі очікування під час збоїв, переміщуються на вихідну БД як тільки остання стає доступною.

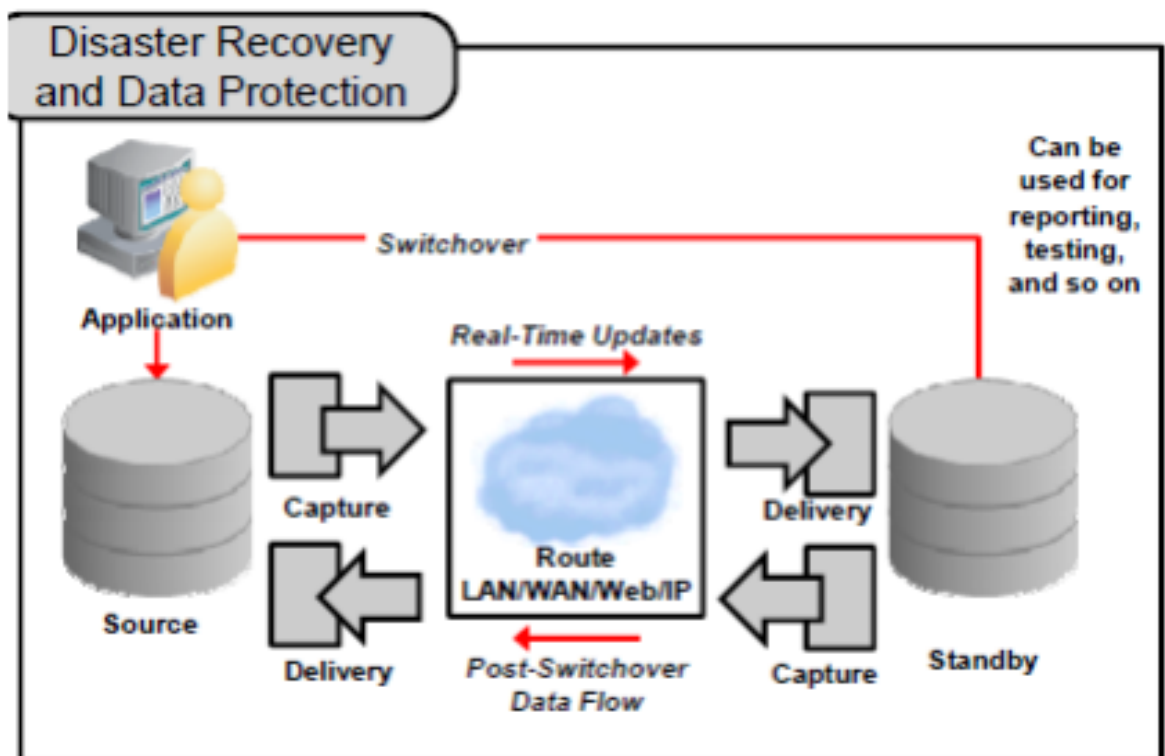


Рис. 2.5. Відмовостійкість

2.1.7. Вимоги до системи

Підтримувані версії СУБД, ОС ПЗ Oracle OGG підтримується на різних платформах, повна інформація про сумісність з різними вендорами СУБД, типами процесорів, версіями ОС і розрядах системи знаходиться в Додатку Б (Матриця сертифікації підтримуваних платформ).

В цілому, вимоги до ресурсів залежать від кількості транзакцій / процесів OGG кожен процес захоплення або доставки потребує 25-55Мб, для процесу integrated capture значення вище і залежить від БД. Вимоги до дискових ресурсів залежать від поточного навантаження і інформації, яка добувається.

Для успішної реплікації даних, особливо в високонавантажених системах, необхідна хороша мережа між вузлами системи. OGG підтримує як IPv4, так і IPv6 TCP IP мережі. Для нормального функціонування OGG необхідні відкриті і незайняті порти TCP / IP.

У разі роботи з декількома БД на 1 сервері необхідна інсталяція окремого OGG для кожного випадку.

В разі установки OGG на Oracle RAC (Oracle Real Application Clusters) інсталяцію слід виконувати на спільно використовуваних дисках.

2.2. Огляд архітектури OGG

Максимальна гнучкість, надійність і продуктивність досягається за рахунок конфігурації, зібраної з незв'язаних модулів, яка і є архітектурою OGG [4].

Така архітектура розбиває процес переносу даних на чотири простих, і в той же час ефективних, етапи (рис. 2.5).

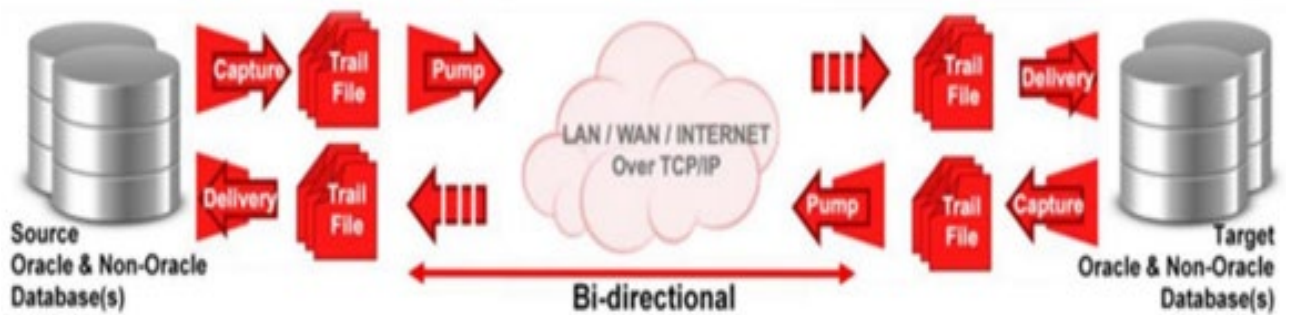


Рис. 2.6. Архітектура OGG [4]

Ключові етапи.

1. Capture (Захоплення) – OGG захоплює операції зі зміни даних, зафіксовані в транзакційних журналах СУБД за допомогою такого, що не порушує роботу СУБД, високопродуктивного методу з низьким додатковим навантаженням.

2. Route (Маршрутизація) – OGG може використовувати різні транспортні протоколи, може компресувати і шифрувати передані зміни. Транзакційні дані можуть розвантажуватися в плоскі файли або JMS черги.

3. Transform (Перетворення) - У будь-якій точці до застосування даних в цільовій системі OGG може здійснювати перетворення і фільтрацію даних за допомогою великої кількості вбудованих функцій.

4. Delivery (Доставка) – OGG застосовує транзакційні зміни до однієї або декількох БД з затримкою менше секунди зі збереженням транзакційної цілісності.

Ключові компоненти. OGG складається із 4 різних модулів і компонент [4]:

- OGG Capture;
- OGG Trail Files;
- OGG Delivery;
- OGG Manager.

В подальшому коротко буде описано кожна з цих складових частин, які власне і забезпечують коректну роботу OGG.

2.2.1. OGG Capture

Модуль Capture отримує тверді транзакції, що включають операції вставки, оновлення та видалення журналу СУБД, та пересилає їх для розповсюдження. Цей модуль не повинен міняти джерело БД або програми для фіксації змін. Модуль Capture використовує ряд методів CDC для підтримки оптимальної роботи. Журнали транзакцій автоматично заповнюються БД незалежно від того, запущений OGG чи ні, і містять будь-які зміни в БД. Немає необхідності створювати додаткові таблиці для роботи механізму захоплення, а додаткове навантаження значно зменшується в порівнянні з фіксацією змін тригерами. Типове додаткове навантаження під час роботи OGG, як правило, становить від 3 до 5%.

Часто потрібно виконати реплікацію не всієї таблиці, а лише її частини. Модуль Capture дозволяє витягувати дані лише з певних таблиць, рядків і стовпців та ігнорувати всі інші записи в журналі транзакцій.

Модуль Capture може надсилати дані через Інтернет, WAN та LAN та оптимізує передачу різними способами. Зазвичай передається лише частина, згенерована джерелом бази даних. Різні рівні стиснення можуть зменшити обсяг переданих даних на 75% і навіть більше. Якщо обсяг змін дійсно дуже великий (більше 100 ГБ / год), ви можете налаштувати кілька модулів Capture, щоб мінімізувати затримку між системою джерела та одержувача.

2.2.2. OGG Trail Files

Trail Files - це унікальний механізм черги в OGG. Він включає останні зміни до портативного, незалежного від платформи, універсального формату даних OGG і може бути перетворений у XML та інші популярні формати для передачі в різні програми. Залежно від вимог, файли шляху можна зберігати у джерелі, на одержувачі або в обох системах.

OGG, який використовує файли записів як кеш, послаблює зв'язок між джерелом та одержувачем, тому можна увімкнути неоднорідну підтримку. На відміну від архітектури процес-процес, ця архітектура дозволяє кожному модулю функціонувати незалежно.

Oracle OGG також пропонує гнучкість у виборі обладнання, ОС, БД для джерел та одержувачів. Клієнти можуть використовувати різні версії файлів Capture, Delivery та Trail у проекті для максимальної гнучкості та простоти використання.

Trail Files можуть бути створені для архівування інформації, яка видаляється з джерела шляхом перетворення операцій видалення та оновлення у вставки в цільовій системі. OGG може також включати зміни в окрему таблицю для цілей аудиту та для задоволення офіційних вимог.

2.2.3. OGG Delivery

Модуль Delivery зчитує всі зміни з Trail Files та негайно застосовує їх до цільової БД. Серед підтримуваних БД: Oracle Database; Microsoft SQL Server; IBM DB2 (LUW), MySQL. Використовуючи адаптери програм OGG, ви можете публікувати дані в системах масового обслуговування (JMS) у форматі XML або інших форматах, а також плоскі файли для завантаження в сторонні системи, такі як ETL.

Delivery застосовує всі зміни в тому самому порядку, в якому вони були зареєстровані у джерелах, щоб забезпечити цілісність кількості посилань та цілісність даних. Крім того, з міркувань послідовності ви використовуєте їх у тому ж контексті транзакції, що і у вихідній системі.

Як і модуль Capture, модуль Delivery може бути налаштований для перетворення даних. За замовчуванням модуль Delivery заповнює всі стовпці цільової системи даними з тих самих вихідних стовпців. Однак досить легко налаштувати OGG для передачі даних з однієї таблиці в кілька або навпаки. Це

може бути використано для нормалізації або денормалізації даних у сховищі або в середовищі OLTP.

Модуль забезпечення використовує ряд спеціалізованих методів для оптимізації завантаження даних у цільову систему. Зазвичай модуль забезпечення виконується локально в цільовій системі, що знімає обмеження в мережі з БД. Введення-виведення також зведено до мінімуму, але властивості вихідних транзакцій збережені. По можливості OGG працює через власний інтерфейс бази даних (Oracle - OCI, MSSQL - OLEDB).

Можна запускати кілька модулів Delivery для підвищення продуктивності та зменшення затримок.

Для більшої гнучкості презентаційний модуль може застосовувати дані відразу або через певний проміжок часу, зберігаючи цілісність транзакцій. Таким чином, є можливість створити додатковий рівень захисту, коли необхідно підтримувати вторинну систему в стані, який збігається із затримкою первинної системи. У цій конфігурації OGG зберігає змінені дані у Trail Files цільової системи, але не додає ці дані до цільової системи, поки всі зміни не досягнуть Trail Files.

Будь-які змінені дані у цільовому файлі слідів можуть бути повністю застосовані до цільової системи, щоб підтримувати їх у стабільному стані. Це може регулювати користувач.

2.2.4. OGG Manager

Для управління процесами OGG може використовуватися інтерфейс командного рядка, який працює через модуль керування адміністратором і робить наступне:

- Визначення параметрів для конфігурації та тонкого налаштування процесів OGG;
- Запуск, зупинка та моніторинг модулів Capture та Delivery;

- Звіти про перевищення визначених меж, а також критичні та інформативні повідомлення;

- Управління ресурсами;

- Управління Trail Files.

Модуль Manager забезпечує явні команди, а також виконує ряд запланованих дій. Наприклад, менеджер процесів використовується для перезапуску компонента OGG, але він також відстежує затримку та видаляє старі файли шляху.

Також забезпечує додаткову гнучкість у відстеженні транзакцій для швидкого пошуку вузьких місць та досягнення оптимальної продуктивності OGG.

2.3. Встановлення OGG

Перед, власне, самим виконанням поставленого в роботі завдання, необхідно встановити саме середовище OGG [5].

Нижче будуть неведені основні кроки для цього.

На системах, де встановлена БД Oracle, змінні ORACLE_HOME, ORACLE_SID будуть визначені на рівні системи. На інших встановлюються параметри SETENV в Extract і Replicat наступним чином (рис. 2.7):

```
SETENV (ORACLE_HOME = "path to Oracle home location")  
SETENV (ORACLE_SID = "SID")
```

Рис. 2.7. Лістинг параметрів SETENV

У разі використання декількох екземплярів Oracle з процесами захоплення і застосування, вказується для кожної групи процесу окремий файл параметрів, який відображений на рис. 2.8:

```

EXTRACT ora9a
SETENV (ORACLE_HOME = "/home/oracle/ora/product")
SETENV (ORACLE_SID = "oraa")
USERID ggga, PASSWORD ggga
RMTHOST sysb
RMTTRAIL /home/ggs/dirdat/rt
TABLE hr.emp;
TABLE hr.salary;
EXTRACT orab
SETENV (ORACLE_HOME = "/home/oracle/ora/product")
SETENV (ORACLE_SID = "orab")
USERID gggb, PASSWORD gggb
RMTHOST sysb
RMTTRAIL /home/ggs/dirdat/st
TABLE fin.sales;
TABLE fin.cust;

```

Рис. 2.8. Лістинг файлу параметрів

Для встановлення файлів OGG необхідно виконати наступні дії:

- розпакувати архів з дистрибутивом під використовувану версію ОС (ім'я директорії, де встановлений golden gate не містить пробілів);
- з директорії, куди скопійовані дані, потім запустити GGSCI і виконати команду створення директорії :

```

GGSCI> CREATE
SUBDIRS GGSCI>EXIT

```

Для створення менеджера необхідно запустити GGSCI наступним чином:

```

EDIT PARAMS ./GLOBALS - редагування параметрів;
MGRSERVNAME oracle_GG_mgr - задається ім'я менеджера;

```

Install option AUTOSTART - перебуваючи в директорії OGG, виконується команда інсталяції з опцією автостарту сервісу.

Щоб підтримувати послідовності в Oracle, необхідно встановити в БД спеціальні процедури, які виконують FLUSH SEQUENCE при першому старті OGG процесу (під час ініціалізації завантаження).

1. Необхідно підключитися до БД як SYSDBA
2. Створення користувача для підтримки DDL:

```

CREATE USER DDLuser IDENTIFIED BY password ;

```

```
GRANT CONNECT, RESOURCE, DBA TO DDLuser ;
```

3. З GGSCI в кожному файлі GLOBALS вказується параметр схеми з ім'ям користувача, створеного в БД

```
EDIT PARAMS ./GLOBALS
```

```
GGSCHEMA DDLuser
```

4. У SQLPLUS потрібно запустити скрипт `sequence.sql`, який знаходиться в корені OGG директорії

5. Даються необхідні привілеї користувачеві:

```
GRANT EXECUTE on DDLuser .updateSequence TO DBLOGINuser ;
```

6. На цільовій машині даються права користувачеві `replicat` процесу

```
GRANT EXECUTE on DDLuser .replicateSequence TO Replicatuser ;
```

7. Включається додаткове ведення журналу для системної таблиці з послідовністю

```
ALTER TABLE sys.seq $
```

```
ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY)
```

```
COLUMNS;
```

2.4. Конфігурація вихідної і цільової Oracle БД

Для кожного процесу існує свій файл конфігурації. Зокрема базова конфігурація відображена на рис. 2.9.

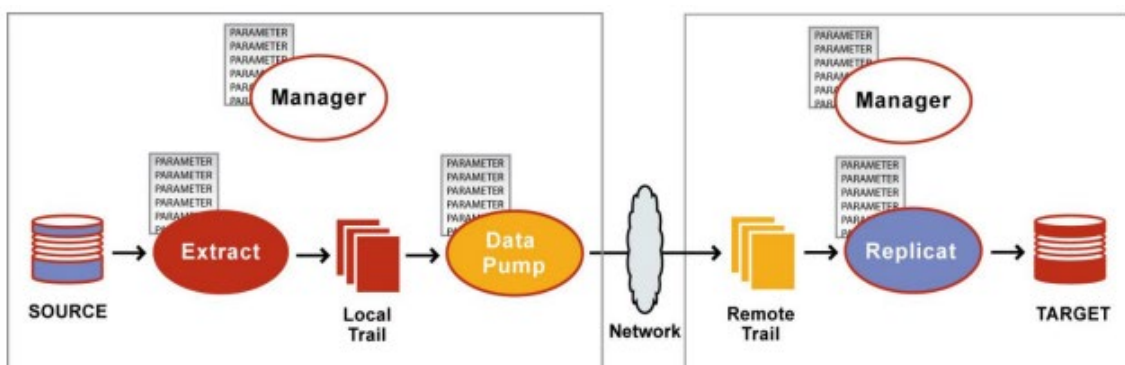


Рис. 2.9. Базова конфігурація [15]

Для конфігурації OGG необхідно визначитися з наступними налаштуваннями [11]:

1. Вибір імен для процесів і файлів
2. Який метод захоплення даних використовувати
3. Призначити користувача всередині СУБД Oracle для OGG:
 - На обох системах необхідно встановити екземпляри OGG
 - Конфігурувати Extract для захоплення змін на вихідній БД
 - Конфігурувати Replicat для застосування змін на цільовій БД

Щодо загальних рекомендацій по іменування процесів, файлів і логів – вони повинні мати осмислені назви для полегшення подальшого процесу конфігурування і підтримки.

Необхідно навести обмеження та вимоги стосовно іменування trail файлів: сама назва trail-файлу може містити тільки 2 символи, наприклад tr, до нього буде автоматично додано 6 символів порядкового номера сиквенсу.

Подальші особливості встановлення та використання OGG з практичною реалізацією завдання будуть відображені в розділі 3.

2.5. Висновки до розділу

У цьому розділі дано повну характеристику OGG: описані ключові можливості і технологічні відмінності, використовувані топології, можливості застосування для сховищ даних в режимі РЧ та для СУБД Oracle, відмовостійкість. Виконано огляд архітектури OGG, зокрема модулів Capture, Trail Files, Delivery, Manager. Також наведено особливості встановлення OGG з конфігурацією вихідної і цільової Oracle БД.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Архітектура БД БС

На рис. 3.1 наведено розроблену відмовостійку архітектуру БД для БС.

Архітектура ІС володіє наступним функціоналом:

- доступ до актуальних даних;
- аналіз інформації в сховищі даних;
- оптимізація роботи в умовах поганого з'єднання каналів зв'язку;
- об'єднання БД під управлінням різних СУБД.

Також дане рішення має наступні переваги:

- мінімальне навантаження як на транспортну мережу, так і на серверні машини, на яких ІС була розгорнута;
- висока гнучкість, що дозволяє вносити зміни в ІС з мінімальними зусиллями;
- легка масштабованість.

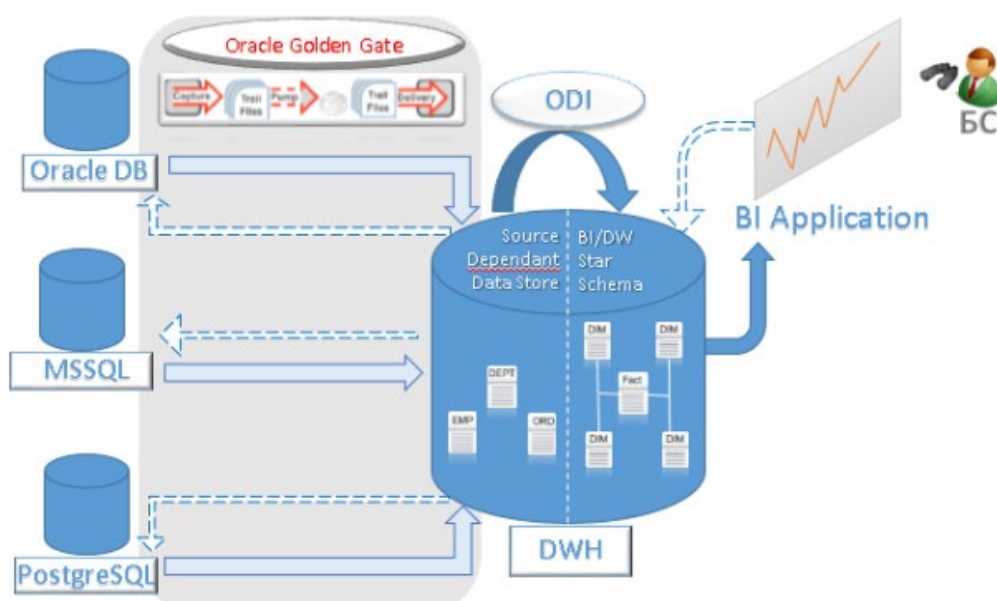


Рис. 3.1. Відмовостійка архітектура БД БС

В подальшому в цьому розділі описані ключові практичні моменти, які виникали при створенні архітектури.

3.2. Захоплення змін та реплікація

У OGG існує 2 режиму захоплення змін [11]:

- classic;
- integrated.

Варто описати кожен з них докладніше з практичної сторони реалізації.

Classic capture. У класичному режимі захоплення змін процес OGG Extract витягує зміни з оперативних і архівних журналів на вихідній системі, або на іншій системі з наданих архівних журналів, не використовуючи механізми СУБД (рис. 3.2). Почасти це добре, оскільки архітектурно реалізовано дуже просто. До того ж цей спосіб має хорошу історію. Цей режим може бути використаний в будь-якій підтримуваний СУБД Oracle [17].

Однак, в класичному режимі OGG більш обмежений в підтримці технологій Oracle, ніж в інтегрованому режимі.

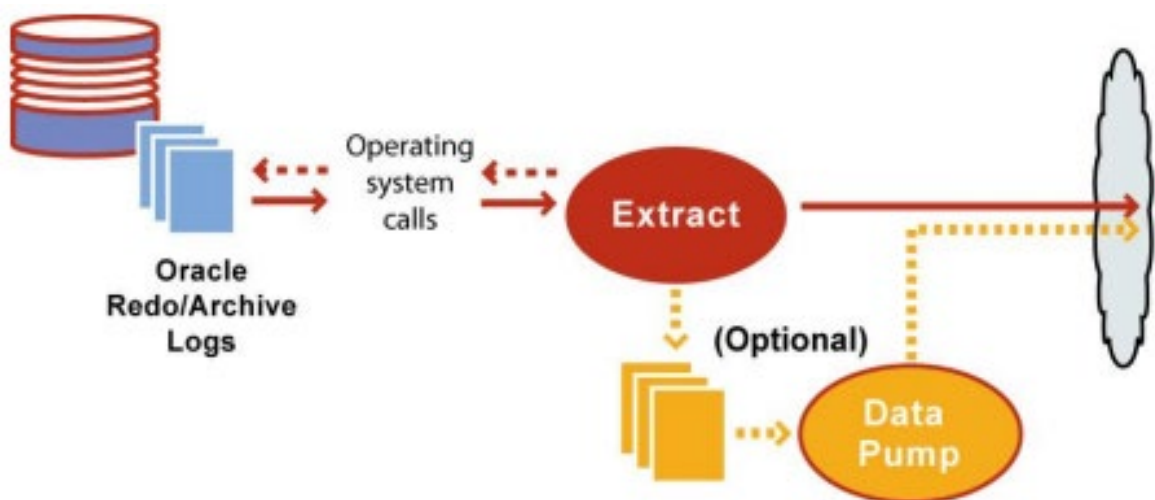


Рис. 3.2. Класичне захоплення

Integrated Capture. В інтегрованому режимі процес OGG Extract задіє спеціальний database logmining server для отримання змін у формі logical change records (LCR) [17]. У інтегрованому режимі OGG підтримує більшу кількість типів даних, режимів зберігання і є простішим у випадку складного джерела (RAC, Exadata, високонавантажені системи) (рис. 3.3).

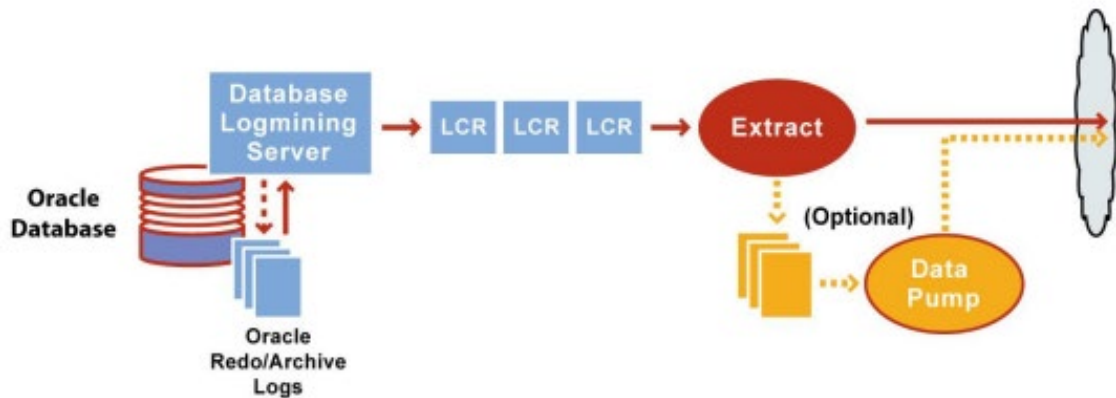


Рис. 3.3. Інтегроване захоплення

В інтегрованому режимі додатково наявно:

- прозора підтримка Oracle RAC, ASM, and TDE без додаткового налаштування;
- Downstream обробка;
- автоматичне перемикання між різними копіями архівних і оперативних журналів;
- швидше виконується фільтрація таблиць;
- ефективніша обробка випадків відновлення бази в "момент-в часі";
- інтеграція з RMAN.

Крім того, для кращої роботи:

- для Exadata слід встановлювати Bundle Patch 3 (версія СУБД буде 11.2.0.3.3)
- для не-Exadata БД необхідний патч.

У випадку з Integrated Capture користувачеві OGG на стороні сервера також знадобляться привілеї [12]:

- Адміністратор OGG - видаються за допомогою пакета `dbms_goldengate_auth`;
- SELECT на подання `V_$DATABASE`.

Базовий набір параметрів для Extract, що працює в класичному режимі, зображено на рис. 3.4.

```
EXTRACT finance
USERID ggadmin, PASSWORD oracle
EXTTRAIL ./dirdat/lt
TABLE ggtest.*;
```

Рис. 3.4. Лістинг базового набору параметрів для Extract в режимі classic

Базовий набір параметрів для Extract, що працює в інтегрованому режимі наведено на рис. 3.5.

```
EXTRACT financerp
USERID ggadmin, PASSWORD oracle
TRANLOGOPTIONS MININGUSER ggmining, MININGPASSWORD oracle
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 164,
DOWNSTREAM_REAL_TIME_MINE y, parallelism 3)
EXTTRAIL ./dirdat/lt TABLE ggtest.*;
```

Рис. 3.5. Лістинг базового набору параметрів для Extract в режимі integrated

Необхідно навести коментарі для представлених вище параметрів [12].

`TRANLOGOPTIONS MININGUSER ggmining, MININGPASSWORD oracle` - задає інформацію для `logmining server` (`MININGUSER` – користувач, під яким буде виконуватися `mining`, `MININGPASSWORD` пароль цього користувача).

`TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 164, DOWNSTREAM_REAL_TIME_MINE y)` - вказує кількість SGA пам'яті, яку використовує `logmining server`, кількість процесів, що виконують `logmining`, а

також режим роботи real-time або archived-log mode, якщо використовується downstream mining data-base.

Щоб додати новий Extract використовується команда наведена на рис. 3.6.

```
dblogin userid ggadmin, password oracle
delete financep
add extract financep, INTEGRATED tranlog, begin now
ADD extrail ./dirdat/lt, EXTRACT financep, MEGABYTES 50
```

Рис. 3.6. Лістинг команди для додавання новго Extract

Також видно, що при створенні нового Extract в БД жодні процеси не створюються. Вони будуть створюватися за допомогою наступної команди, яка і робить всю заключну роботу по конфігурації СУБД Oracle:

```
REGISTER EXTRACT financep DATABASE
```

Результат:

```
2020-11-24 15:00:40 INFO OGG-02003 Extract FINANCEP successfully
registered with database at SCN 32789317.
```

Фрагмент коду з вивантаження файлу alert.log наведено на рис. 3.7.

```
Logminer Bld: Done Thu Nov 24 15:00:56 2020
Setting XOUT_CLIENT_EXISTS to Y for Capture: OGG$CAP_FINANCEP
Setting XOUT_CLIENT_EXISTS to Y for Capture: OGG$CAP_FINANCEP
Thu Nov 24 15:00:56 2020
GoldenGate CAPTURE CP02 for OGG$CAP_FINANCEP started with
pid=44, OS id=16330
First applied SCN of the GoldenGate path from capture
(OGG$CAP_FINANCEP) to propagation () to apply (OGG$FINANCEP) is set
to SCN: 2789317 (0x2a8fc5.00000000)
.....
```

Рис.3.7. Лістинг фрагменту коду з вивантаження файлу alert.log

Якщо тепер включити трасування, буде наступне.

1. Перше, що робить OGG, - налаштовує деякі внутрішні, недокументовані параметри в пакеті dbms_xstream_gg_adm.

2. Потім викликається цілком стандартний `dbms_capture_adm.build(:result)`. Ця операція повинна скинути в redo log БД словник СУБД. У Streams це використовувалося для того, щоб процеси застосування змін знали про структуру СУБД джерела.

3. Далі створюється черга повідомлень, таблиця для неї, процес CAPTURE, який буде цю чергу наповнювати. І останнім кроком з допомогою `DBMS_XSTREAM_GG_ADM.ADD_OUTBOUND` створюється `outbound`-процес (по суті це Apply-процес), що дозволяє ззовні читати дані, які захоплені в процес Capture. Зовнішнім підписником на ці зміни буде процес Extract.

4. Передостаннім кроком видаляються правила для створеного процесу Capture.

5. Ну і останній крок - дозволити використовувати OUTBOUND.

Запуск Extract. Після запуску в журналі з'являється повідомлення:

```
2020-12-01 21:25:48 INFO OGG-02036 Integrated capture successfully attached  
to logmining server OGG $ FINANCEP.
```

Що стосується зміни списку видобутих таблиць, то Extract при перезапуску автоматично оновлює правила процесу Capture.

У Streams є режим "Downstream" (рис. 3.8), в якому БД джерела пересилає журнали потоком режимі на цільову систему, а потім ці журнали розбираються «на льоту» і пишуться в чергу для подальшого застосування.

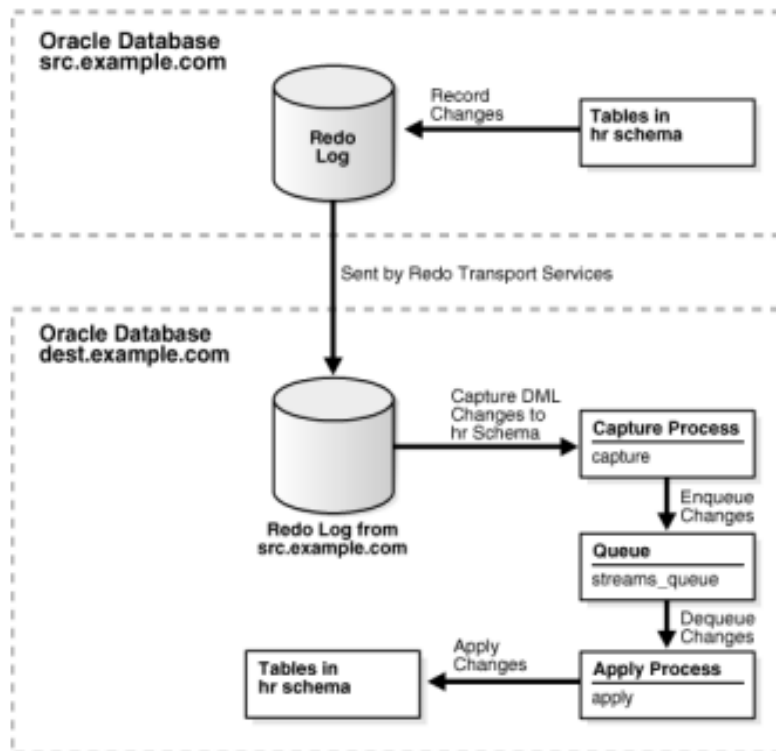


Рис. 3.8. Конфігурація реплікації з використанням режиму Downstream захоплення

Підготовка джерела для downstream реплікації (gg1orcl.local) [17].

1. Установка використання глобальних імен:

```
Alter system set global_names = TRUE scope = BOTH;
```

2. Права адміністратора OGG

```
grant connect, resource, dba to ggadmin identified by oracle;
```

```
exec dbms_goldengate_auth.grant_admin_privilege ('GGADMIN');
```

3. Конфігурація доставки REDO з вихідної на цільову базу:

– налаштування TNS-alias на джерелі для приймача;

– копіювання файлу паролів з джерела на приймач;

– на джерелі конфігурується додатковий LOG_ARCHIVE_DEST_n, код

наведено на рис 3.9:

```

ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2 = ENABLE;
ALTER SYSTEM
SET LOG_ARCHIVE_DEST_2 = 'SERVICE = gg2orcl.local LGWR SYNC
NOREGISTER
VALID_FOR = (ONLINE_LOGFILES, PRIMARY_ROLE)
DB_UNIQUE_NAME = gg2orcl ';
ALTER SYSTEM SET LOG_ARCHIVE_CONFIG = 'DG_CON-
FIG = (gg1orcl, gg2orcl) ';

```

Рис. 3.9. Лістинг конфігурування додаткового LOG_ARCHIVE_DEST_n

Підготовка downstream БД.

1. Установка використання глобальних імен:

```
Alter system set global_names = TRUE scope = BOTH;
```

2. Права адміністратора OGG

```
grant connect, resource, dba to ggadmin identified by oracle;
exec dbms_goldengate_auth.grant_admin_privilege ('GGADMIN');
GRANT SELECT ON V_$ DATABASE TO GGADMIN;
```

3. Створення standby redo log (вони повинні бути не менше, ніж журнали на джерелі; кількість груп має бути не менше, ніж на джерелі) наведено на рис. 3.10.

```

ALTER DATABASE ADD STANDBY LOGFILE GROUP 4
( '/U01/oradata/gg1orcl/slog4.rdo') SIZE 50M;
ALTER DATABASE ADD STANDBY LOGFILE GROUP 5
( '/U01/oradata/gg1orcl/slog5.rdo') SIZE 50M;
ALTER DATABASE ADD STANDBY LOGFILE GROUP 6
( '/U01/oradata/gg1orcl/slog6.rdo') SIZE 50M;

```

Рис. 3.10. Лістинг процесу створення standby redo log

4. Вказується область для архівації standby-журналів, код на рис. 3.11:

```

ALTER SYSTEM SET
LOG_ARCHIVE_DEST_2 = 'LOCATION = /u01/ oradata/gg1orcl /arc
VALID_FOR = (STANDBY_LOGFILE, PRIMARY_ROLE) ';
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2 = ENABLE;
ALTER SYSTEM SET LOG_ARCHIVE_CONFIG = 'DG_CON-
FIG = (gg1orcl, gg2orcl) ';

```

Рис. 3.11. Лістинг коду області для архівації standby-журналів

Налаштування Extract для захоплення змін до downstream БД.

1. Увійдіть у вихідну БД за допомогою DBLOGIN і в downstream базу за допомогою програмного коду з рис. 3.12:

```
DBLOGIN USERID ggadmin @ src, PASSWORD oracle
MININGDBLOGIN USERID ggadmin @ dest, PASSWORD oracle
REGISTER EXTRACT etesti DATABASE
ADD EXTRACT etesti INTEGRATED TRANLOG, BEGIN NOW
```

Рис. 3.12. Лістинг коду входу у вихідну БД

2. Файл параметрів буде стандартним, але окремі з них потрібно буде додатково вказати (рис. 3.13):

```
EXTRACT xa
USERID ggadmin@gg1orcl.local, PASSWORD oracle
TRANLOGOPTIONS MININGUSER ggadmin@gg2orcl.local, MININGPASS-
WORD oracle
TRANLOGOPTIONS INTEGRATEDPARAMS (downstream_real_time_mine Y)
EXTTRAIL dirdat / ta
table ggtest.employees;
```

Рис. 3.13. Лістинг задання додаткових параметрів

Даний режим дозволяє:

- збирати зміни в РЧ;
- створювати на джерелі практично нульове навантаження.

3.3. Створення OGG екземпляру

Для запуску екземпляра необхідний працюючий процес менеджера на всіх системах, які входять в реплікацію (рис. 3.13).

Для запуску менеджера необхідно вказати додаткові параметри менеджера.

```
GGSCI> EDIT PARAMS MGR.
```


Базові параметри для Extract

Extract group name	Group name – ім'я Extract
User id, password pw	Інформація для під'єднання до БД, encryption options для шифрування паролю
TRANLOGOPTIONS MININGUSER user id, MININGPASSWORD pw	Для інтегрованого захоплення вказує інформацію по під'єднанню downstream mining database, якщо така використовується
EXTTRAIL pathname	Шлях до локального Трейсу, куди екстракт викладає захоплені дані
SEQUENCE owner.sequence;	Вказуються послідовності, які повинні відслідковуватися
TABLE owner.table;	Ім'я таблиці може бути замінено на *, схему замінювати не можна
TABLEEXCLUDE owner.table	Для виключення таблиць з експортованої схеми, якщо використовується schema. *

Конфігурація Replicat. Дані дії виконуються на цільовій системі. Для процесу застосування даних до БД необхідна таблиця з checkpoint-ами в цільовій БД, це необхідно для успішного відновлення після збоїв обладнання / ОС. У більшості випадків достатньо створення таблиці за замовчуванням.

Для запуску GGSCI виконується команда:

```
DBLOGIN, USERID db_user [, PASSWORD pw [encryption options]]
```

Далі створюю таблицю з чекпоінтами

```
ADD CHECKPOINTTABLE owner.table
```

Далі необхідно вказати цю таблицю у файлі параметрів процесу Replicat

```
EDIT PARAMS ./GLOBALS
```

```
CHECKPOINTTABLE owner.table
```

Для поліпшення продуктивності, за замовчуванням використовується асинхронний commit з nowait опцією. Можливо відключити асинхронний commit with nowait використовуючи DBOPTIONS параметр з DISABLECOMMITNOWAIT опцією в файлі параметрів Replicat (табл. 3.2).

Параметри, що відносяться до базової конфігурації Replicat

REPLICAT group name	Ім'я групи процесу застосування логів
USERID user id, PASSWORD pw	Опис даних для підключення до БД
DBOPTIONS SUPPRESSTRIGGERS, DEFERREFCONST	SUPPRESSTRIGGERS – відключає спрацьовування тригерів при використанні реплікації, з метою прибрати подвійне спрацьовування тригерів DEFERREFCONST – перемикає константи в стан DEFERRABLE з метою відкласти перевірку цілісності до завершення транзакції Replicat
ASSUMETARGETDEFS	Означає припущення того, що БД / схеми ідентичні, інакше необхідно описувати дані з допомогою фала DEFGEN
DISCARDFILE full_ pathname	Повний шлях до файлу, куди REPLICAT буде скидати записи, які неможливо застосувати до цільової БД
MAP owner.table, TARGET owner.table;	Описує зв'язок між вихідною і цільовою БД.

Запуск процесів реплікації.

На сервері №1 запускаю та перевіряю, що процес запустився

```
START EXTRACT orcl1_e
```

```
VIEW REPORT orcl1_e
```

На сервері №2 запускаю процес застосування змін

```
START REPLICAT rep
```

```
VIEW REPORT rep
```

Відповідно зупинка виконується командою

```
STOP {EXTRACT | REPLICAT} group_name
```

Є також варіант із зупинкою методом «вбивання» процесу:

```
KILL {EXTRACT | REPLICAT} group_name
```

Даний вид зупинки може порушити цілісність, тому що НЕ будуть проставлені checkpoint-и. Видалення проводиться таким же простим способом: спочатку зупиняється процес, далі delete {EXTRACT | REPLICAT} group_name

Створення таблиці EMPLOYEES для реплікації наведено на рис. 3.15.

```
connect system / oracle
create table test.EMPLOYEES as select * from hr.EMPLOYEES;
ALTER TABLE test.EMPLOYEES ADD PRIMARY KEY (employee_id);
insert into employees select * from employees;
```

Рис. 3.15. Лістинг коду створення таблиці EMPLOYEES для реплікації

3.4. Вирішення проблем

Перевірка вихідної схеми/бази. На сайті підтримки Oracle опубліковані скрипти, які допомагають перевірити можливість реплікації схеми або БД, а також знайти можливі проблемні об'єкти.

Скрипт перевіряє:

– включене або не включене мінімальне додаткове «журналювання» (Supplemental Logging);

– які є обмеження цілісності (ПК, FK);

– які є типи даних, чи є об'єктні типи;

– чи є таблиці, для яких не визначено ПК.

Вирішення конфліктів. Найбільш затребуваний тип реплікації OGG - двостороння реплікація. Такий спосіб використання OGG дозволяє створити 2 незалежних відмовостійких ЦОД.

Дві основні проблеми:

– одні й ті ж дані змінюються в різних центрах;

– один центр відключається, ми переводимо користувачів в другій, вони там змінюють дані, але при відновленні роботи першого центру виявляється, що користувачі повторили частина операцій в обох центрах.

Управління конфліктами.

Конфлікти можуть виникати різні:

- конфлікт унікальності - трапляється, коли Replicat намагається вставити рядок, що порушує обмеження цілісності PRIMARY KEY або UNIQUE;
- конфлікт зміни - трапляється, коли один і той самий рядок змінюється приблизно в один і той же час в двох ЦОД. Replicat виявляє конфлікт за результатами порівняння старих значень (значень before), збережених в trail-файлі, і значень в цільовій БД;
- конфлікт видалення - трапляється, коли в одній базі рядок видаляється, а в іншій він оновлюється або теж видаляється. Як наслідок, Replicat не може знайти рядок з заданим первинним ключем.

Багато уникати виникнення конфліктом наступними способами:

- конфігурувати sequence з діапазонами, що не пересікаються;
- жорстко рознести користувачів з різних БД - наприклад, за регіонами;
- знижувати затримку в реплікації.

Якщо конфлікти неминучі, то застосовуємо CDR, який має готові процедури вирішення:

- вирішення конфлікту унікальності для INSERT;
- вирішення конфлікту "no data found" для UPDATE, коли рядок існує, але значення BEFORE в одному або декількох стовпцях відрізняються від поточних значень в БД;
- вирішення конфлікту "no data found" для UPDATE, коли рядок не існує;
- вирішення конфлікту "no data found" для DELETE, коли рядок існує, але значення BEFORE в одному або декількох стовпцях відрізняються від поточних значень в БД;
- вирішення конфлікту "no data found" для DELETE, коли рядок не існує.

Нижче на рис. 3.16 наведено синтаксис вирішення конфліктів в директиві MAP для Replicat:

```

[, COMPARECOLS (
{ON UPDATE | ON DELETE}
{ALL | KEY | KEYINCLUDING (<col> [...]) | ALLEXCLUDING
(<Col> [...])}
[...]]
)]
[, RESOLVECONFLICT (
{INSERTROWEXISTS | UPDATEROWEXISTS | UPDATEROWMISSING | DEL-
ETEROWEXISTS |
DELETEROWMISSING}
({DEFAULT | <resolution name>},
{USEMAX (<res_col>) | USEMIN (<res_col>) | USEDELTA | DISCARD
| OVERWRITE | IGNORE})
[, COLS (<col> [...])]
)]

```

Рис. 3.16. Лістинг коду вирішення конфліктів в директиві MAP для Replicat

Наступні типи даних можуть бути використані для вирішення конфлікту [18, 19]:

- numeric;
- date;
- timestamp;
- char / nchar;
- varchar / nvarchar;

Тобто ці стовпці можуть бути включені в директиви виявлення конфлікту COMPARECOLS, GETBEFORECOLS, а також в директиви дозволу конфлікту RESOLVECONFLICT в USEMIN і USEMAX. Якщо потрібно використати USEDELTA, то підтримуються тільки числові поля. Вирішення конфліктів не виконується в режимі BATCHSQL - якщо виникає конфлікт, то Replicat переходить в режим GROUPTRANSOPS, а потім в режим одиночних транзакцій.

3.5. Попереднє налаштування Active-Active реплікації

На рис. 3.17 наведено режим для стандартного налаштування реплікації [18, 20].

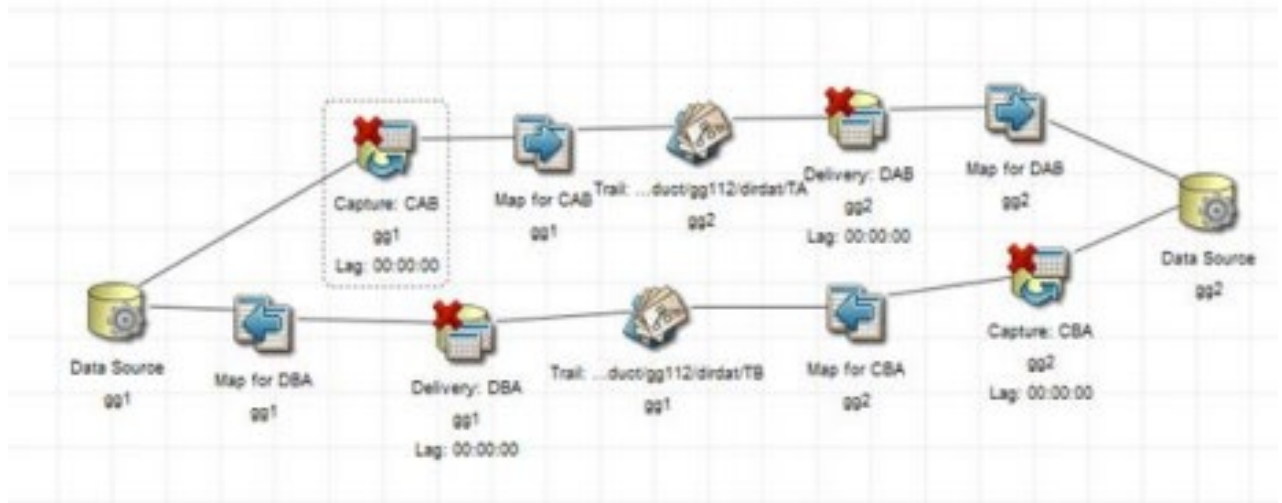


Рис. 3.17. Режим для стандартного налаштування реплікації

В результаті налаштоване по 2 REPLICAT, 2 EXTRACT і 2 TRAIL. Потрібно уникнути зациклення реплікації, тому в Extract необхідно додати:

- для Oracle: `TRANLOGOPTIONS EXCLUDEUSER <user name>` (User_name - це користувач, під яким працює OGG);
- для MSSQL: не потрібно додаткового налаштування;
- для MySQL `TRANLOGOPTIONS FILTERTABLE <table_name>` (table_name - це checkpoint таблиця).

В результаті є наступні параметри для Extract:

- extract CAB (рис. 3.18)

```

userid ggadmin, password oracle
discardfile /u01/app/oracle/product/gg112/dirrpt/CAB.dsc,
purge
tranlogoptions excludeuser ggadmin
rmthost gg2, mgrport 7809
rmttrail / u01 / app / oracle / product / gg112 / dirdat / TA
table ggtest.employees8;

```

Рис. 3.18. Лістинг коду для extract CAB

– replicat DAB (рис. 3.19)

```

discardfile /u01/app/oracle/product/gg112/dirrpt/DAB.dsc, purge
assumetargetdefs
userid ggadmin, password oracle
map ggtest.employees8, target ggtest.employees8;

```

Рис. 3.19. Лістинг коду для replicat DAB

Для тестів була створена наступна таблиця:

```

create table ggtest.employees8 (id number primary key, ename
varchar2 (40), change_time date, comment1 varchar2 (400));

```

Додавання вирішення конфліктів.

Виявлення конфліктів буде здійснюватися за рахунок порівняння значень BEFORE в trail-файлі з поточними значеннями в БД. Вирішується завдання шляхом виконання наведених нижче команд.

1. Щоб змусити Oracle писати в журнал значення BEFORE тих стовпців, які потрібно порівнювати - зазвичай Oracle пише тільки ті стовпці, які змінюються. Це завдання вирішується за допомогою ADD TRANDATA:

```

DBLOGIN userid ggadmin, password oracle
ADD TRANDATA ggtest.employees8 COLS (ename, change_time)

```

2. Щоб OGG переносив на цільову систему не тільки значення після, але і до, необхідно додати в параметри Extract наступної Директиви в TABLE, код якої наведено на рис. 3.20:

```

table ggtest.employees8,
GETBEFORECOLS
(
ON UPDATE ALLEXCLUDING (comment1),
ON DELETE ALLEXCLUDING (comment1)
);

```

Рис. 3.20. Лістинг коду директиви TABLE

Тепер потрібно задати, як будуть вирішуватися конфлікти. Програмний код наведено на рис. 3.21.

```

MAP ggtest.employees8, TARGET ggtest.employees8,
COMPARECOLS (ON UPDATE ALLEXCLUDING (comment1), ON DELETE AL-
LEXCLUDING (comment1)),
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX
(Change_time))),
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX
(Change_time))),
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)),
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)),
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD))
);

```

Рис. 3.21. Лістинг коду для вирішення конфліктів

3.6. Висновки до розділу

У цьому розділі практично розроблена відмовостійка архітектура БС. Описані два режими захоплення змін (classic та integrated). Докладно описані програмні механізми реплікації БД, приведено процедуру створення OGG екземпляру. Також запропоновано способи усунення проблем та конфліктів, які можуть виникнути при використанні БД. Проведено попереднє налаштування Active-Active реплікації.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Метою кваліфікаційної роботи магістра є дослідження методів та засобів побудови білінгових систем з відмовостійкою архітектурою. Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників з розробки ПЗ комп'ютерних систем, в тому числі і фахівців з дослідження методів та інструментальних засобів, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [30]. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Згідно Вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до цих вимог;

– переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 22.08.2005 N 161, зареєстрованого в Міністерстві юстиції України 05.09.2005 за N 990/11270 (НАПБ Б.06.004-2005);

– Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду України від 28.10.98 N 247 (далі - ДБН В.2.5-56:2014, з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014. Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, ДБН В.2.5-56:2010, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу

провідників, температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При дослідженні методів та засобів побудови білінгових систем з відмовостійкою архітектурою, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні

вимоги». Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп'ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців з дослідження методів та засобів побудови білінгвових систем з відмовостійкою архітектурою.

4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки

Екологічне співтовариство розробило сімейство інструментів комплексної екологічної оцінки. Програмне забезпечення і послуги (ESS), комерційна група ПАСА, включаючи AirWare (для повітряних проблеми якості), WaterWare (для якості води), CityWare (якість повітря і води в контексті великих міст) і EIAxpert (для надання допомоги із загальним впливом на навколишнє середовище). Функціональність в цілому схожа на RAISON, хоча з великим акцентом на моделювання і меншим акцентом на керування даними. Знову ж таки, інструменти ESS розроблені як модульні набори інструментів (доступні спеціальні системи для вирішення конкретних завдань). Компоненти включають стандартні імітаційні моделі, включаючи моделі ISC і PBM Агентства з охорони навколишнього середовища США, управління даними, в тому числі ГІС, аналіз даних (наприклад, аналіз часових рядів даних спостережень), візуалізація, а також оптимізація [23].

Іноді немає готових моделей, придатних для конкретного застосування, але тягар розробки нової програми на Фортрані або С / С ++ є надмірним. Розробка

моделі оточення може відносно легко реалізувати власні моделі комп'ютерів і не турбуватися про включення процедур для вирішення рівнянь, візуалізації і т. д. Як правило, за допомогою цих інструментів користувач просто повинен вказати свою модель, використовуючи або математичні рівняння, або спеціальні графічні символи або значки, які безпосередньо представляють поведінку системи.

На даний момент є розроблені моделі комп'ютерного забезпечення процесу для оцінки радіаційної та хімічної обстановки.

GEMS – це система на основі моделей, яка підтримує оцінки схильності і ризику, надаючи доступ до одиночних і мультимедійних моделям експозиції, фізико-хімічні властивості методи оцінки, статистичний аналіз, графічні та картографічні програми з відповідними даними на навколишнє середовище, джерела, рецептори і популяції. У розробці з 1981 року, GEMS надає аналітикам інтерактивний, легко досліджуваний інтерфейс для різних моделей, програм і даних, які необхідні для оцінки хімічного впливу і ризику [24].

HSPF – це комплексний пакет для моделювання кількості і якості стоків з багатоцільових водозборів і процесів радіації, що відбуваються в потоках або повністю змішаних озерах. Це дозволяє інтегроване моделювання землі і ґрунту, процесів забруднення при гідравлічній і осадово-хімічній взаємодії. Результатом моделювання є тимчасові дані витрати стоку, концентрація поживних речовин і пестицидів, а також дані кількості і якості води в будь-якій точці водозбору. Алгоритми якості води включають динаміку BOD / DO, вуглець, азот і фосфор. Процеси трансформації, які включені в модель це: гідроліз, фотоліз, окислення, випаровування, сорбція і біодеградація. Вторинні або «дочірні» хімічні речовини також моделюються.

Вимоги до даних для моделі можуть бути досить широкими в залежності від конкретного застосування.

Модель MMSOILS – це методологія оцінки впливу на людину і ризику для здоров'я, пов'язаних з викидами забруднень з небезпечних відходів. Мультимедійна модель, що стосується перенесення хімічної речовини в ґрунті

води, поверхневі води, атмосферу і накопичення в їжі. Шляхи впливу на людину, які розглянуті в методології включають: потрапляння в ґрунт, вдихання летких речовин в повітря і тверді частинки, шкірний контакт, прийом питної води і т.д. Ризик, пов'язаний із загальною дозою опромінення, розраховується на основі хімічної токсичності [25].

4.3. Висновки до розділу

В цьому розділі проаналізовано важливі питання охорони праці та безпеки в надзвичайних ситуаціях, висвітлено питання комп'ютерного забезпечення процесу оцінки радіаційної та хімічної обстановки.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була реалізована БС для вирішення поставлених завдань збору даних і реплікації в РЧ в критично важливих транзакційних системах. Серед інших технологій, автором було розроблено рішення на основі OGG, використання якого допомогло вирішити питання відмовостійкості і високої доступності системи.

Для того щоб не навантажувати основні сервера при аналізі даних або при збереженні резервних копій, використовується безперервна реплікація даних. На даний момент досить багато компаній пропонують свої рішення по реплікації даних, в тому числі і компанія Oracle з середовищем OGG.

Розроблена архітектура володіє наступними характеристиками:

- висока ступінь відмовостійкості;
- вилучення даних в РЧ;
- аналіз і реагування в РЧ;
- стійкість до поганих каналів.

Основні результати:

- забезпечено збір даних і реплікацію в РЧ;
- детально вивчено середовище OGG (вимоги до системи, архітектура логічної складової, різні топології в практичному застосуванні);
- спроектувано відмовостійку архітектуру БД БС;
- практично реалізовано (істаляція та конфігурування ПЗ) з тестуванням отриманих результатів.

Завдяки постійному доступу до транзакційних даних, реалізована розробка дозволить швидко і точно перетворювати їх в інформацію, яка необхідна для здійснення діяльності підприємства. Це дозволить оптимізувати бізнес-процеси і обслуговування клієнтів, відображати ринкові можливості, що дасть змогу компаніям і підприємствам залишатися конкурентно здатними і розвиватися, постійно збільшуючи прибуток.

Така архітектура вирішує практичні проблеми, пов'язані як з плановими простоями ІС в результаті міграцій БД, оновлення ПЗ, СУБД або ОС сервера, так і в разі несподіваних збоїв внаслідок фізичного пошкодження сховищ, відключення обчислювального майданчика і людських помилок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Татарчук М.І. Корпоративні інформаційні системи: Навч. посібник.— К.: КНЕУ, 2005. 291 с.
2. Операторська платформа надання послуг: Електронне навчальне видання. Конспект лекцій / Л. С. Глоба, О.М. Дяденко, В.Ф. Чердинцева. К.: НН ІТС НТУУ «КПІ», 2012. 191 с.
3. Павленко Л.А. Корпоративні інформаційні системи: Навч. посібник. Харків: ВД „ІНЖЕК”, 2003. 260 с.
4. Использование Oracle GoldenGate 11g для репликации Microsoft SQL Server Database. URL: <http://oraclegis.com/blog/2012/01/30/использование-oracle-goldengate-11g-для-microsoft-sql-server-database/> (дата звертання: 07.12.2020).
5. Installing Oracle Golden Gate. URL: https://docs.oracle.com/en/middleware/goldengate/core/19.1/installing/setting-oracle_home-and-oracle_sid.html (дата звертання: 07.12.2020).
6. Тимчук М.І. Особливості використання Oracle GoldenGate для розробки відмовостійкої архітектури баз даних. Інформаційні моделі, системи та технології: Праці VIII наук.-техн. конф. (Тернопіль, 09-10 грудня 2020 р.) Тернопіль, 2020. – С. 121.
7. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. М.: Нолидж, 2001. 443 с.
8. J.-C. Laprie. Dependable computing and fault tolerance: Concepts and terminology. *In The 15th International Symposium On Fault-Tolerant Computing*, pages 2–11, 1985.
9. A. Aviżienis and J.-C. Laprie. Dependable computing: From concepts to design diversity. *Proceedings of the IEEE*, 74(5), 1986.
10. J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers, 1993

11. Robert G. Freeman. Oracle GoldenGate 11g. Handbook. McGraw-Hill Education. 2013. 448 p.
12. Алапати Сэм Р. Oracle Database 11g. Руководство администратора баз данных. Вильямс (Apress). 2010. 1440 с.
13. Рик Гринвальд, Роберт Стаковьяк, Джонатан Стерн. Oracle 11g. Основы. 4-е издание. 2009. 455 с.
14. Стив Бобровский. Oracle Database XE для Windows. Эффективное использование. Лори. 2009. 512 с.
15. Hank Tullis, Paul Elbow, Chaitanya R. Geddam. Beginning Oracle GoldenGate. Apress. 2015. 400 p.
16. Oracle Golden Gate Installing and Configuring Oracle GoldenGate for Oracle Database. URL: https://docs.oracle.com/goldengate/1212/gg-winux/GIORA/process_mode.htm#GIORA212 (дата звертання: 08.12.2020).
17. Ankur Gupta. Oracle Goldengate 11g Complete Cookbook. Packt Publishing. 2013. 362 p
18. Документація по Oracle Golden Gate. URL: <https://docs.oracle.com/goldengate/1212/gg-winux/GIORA/img/integratedcapture.png> (дата звертання: 12.12.2020).
19. Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала, том 1. Основы. – 7-е изд. : Пер. с англ. – М. : Издательский дом «Вильямс», 2006. – 286 с.
20. Prusinski B., Phillips S. Expert Oracle GoldenGate. Apress, 2011. P. 35-39.
21. Unit Testing with SQL Developer // Oracle / Integrated Cloud Applications and Platform Services URL: https://docs.oracle.com/cd/E15846_01/doc.21/e15222/unit_testing.htm/ (дата звертання: 21.11.2020)
22. Дич Л.З. Биллинговые системы в телекоммуникациях / Л.З. Дич. – М.: Радио и связь, 2003. – 229 с.
23. Зеркалов Д.В. Охорона праці в галузі: Загальні вимоги. Навчальний посібник. К.: Основа. 2011. 551 с.

24. USNRC. 1979. Расчет выбросов радиоактивных материалов в газообразных и жидких сточных водах из реакторов с кипящей водой (код BWRGALE), NUREG-0016/

25. Толлок А.О. Крюковська О.А. Безпека життєдіяльності: Навч. посібник. – 2011. – 215 с.

ДОДАТОК А
Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



9–10 грудня 2020 року

ТЕРНОПІЛЬ
2020

А. Волоха, Л. Дмитроца РЕЗУЛЬТАТИ МОНІТОРИНГУ ТА АВТОМАТИЗАЦІЇ КЕРУВАННЯ СЕРВЕРАМИ В ВИСОКОНАВАНТАЖЕНИХ СИСТЕМАХ	
A. Volokha, L. Dmytrotsa RESULTS OF MONITORING AND AUTOMATION OF SERVER CONTROL IN HIGHLY LOADED SYSTEMS	116
В. Шмарай АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ УПРАВЛІННЯ ПРОЕКТАМИ	
V. Shmarai ANALYSIS OF PROJECT MANAGEMENT INFORMATION SYSTEMS	118
В. Яцишин, А. Сеньків «МЕТОДИ ТА ЗАСОБИ ІНТЕЛЕКТУАЛЬНОЇ АУТЕНТИФІКАЦІЇ НОМЕРНИХ ЗНАКІВ АВТОМОБІЛІВ ПРИ КЕРУВАННІ АВТОМАТИЧНИМ ПЛАГБАУМОМ»	
V. Yatsyshyn, A. Senkiv METHODS AND TOOLS OF INTELLIGENT AUTHENTICATION OF CAR LICENSE PLATES AT AUTOMATIC BARRIER CONTROL	119
М. Тимчук ОСОБЛИВОСТІ ВИКОРИСТАННЯ ORACLE GOLDENGATE ДЛЯ РОЗРОБКИ ВІДМОВУСТІЙКОЇ АРХІТЕКТУРИ БАЗ ДАНИХ	
M. Timchuk PECULIARITIES OF USING THE ORACLE GOLDEN GATE TO DEVELOP A FAULT-TOLERANT DATABASE ARCHITECTURE	121
В. Яцишин, В. Степчук ТРЕНДИ ТА ЇХ РОЛЬ В АНАЛІЗІ МАРКЕТИНГОВИХ ДАНИХ	
V. Yatsyshyn, V. Stepchuk TRENDS AND THEIR ROLE IN THE ANALYSIS OF MARKETING DATA	122
І. Чорняк ОПТИМІЗАЦІЯ ТА МОДЕЛЮВАННЯ БЕЗДРОТОВОЇ МЕРЕЖІ LTE В СЕРЕДОВИЩІ «FORSK ATOLL»	
I. Chorniak OPTIMIZATION AND MODELING LTE WIRELESS NETWORK IN «FORSK ATOLL»	123
В. Яцишин, В. Хацюр ПЕРЕВАГИ КОМПОНЕНТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ	
V. Yatsyshyn, V. Khatsiur BENEFITS OF COMPONENT-ORIENTED PROGRAMMING	124
Ю. Лещишин, І. Міська, Т. Назаренюч СТРУКТУРНЕ МОДЕЛЮВАННЯ ЦИФРОВИХ КАНАЛІВ ЗВ'ЯЗКУ	
Y. Leshchynshyn, I. Mis'ka STRUCTURAL MODELING OF DIGITAL COMMUNICATION CHANNELS	126
Ю. Лещишин, І. Міська, Т. Назаренюч СТВОРЕННЯ ВБУДОВАНИХ СИСТЕМ НА БАЗІ СТРУКТУРНО- ПАРАМЕТРИЧНИХ МОДЕЛЕЙ ЦИФРОВИХ КАНАЛІВ ЗВ'ЯЗКУ	
Y. Leshchynshyn, I. Mis'ka EMBEDDED SYSTEMS CREATION ON THE BASIS OF STRUCTURAL - PARAMETRIC MODELS OF DIGITAL COMMUNICATION CHANNELS	127

УДК 004.658.3

М.І. Тимчук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ОСОБЛИВОСТІ ВИКОРИСТАННЯ ORACLE GOLDENGATE ДЛЯ РОЗРОБКИ ВІДМОВОСТІЙКОЇ АРХІТЕКТУРИ БАЗ ДАНИХ

UDC 004.658.3

M.I. Timchuk

PECULIARITIES OF USING THE ORACLE GOLDEN GATE TO DEVELOP A FAULT-TOLERANT DATABASE ARCHITECTURE

Для створення надійної і доступної БД можна використовувати два дуже різні підходи: усунення помилок (запобігання виникненню помилок по конструкції); відмовостійкості. (використання надмірності, щоб уникнути збоїв через несправності). Усунення помилок – це процес важкий і практично неможливий у великих і складних БД. Це робить відмовостійкість єдиною реалістичною альтернативою. Для забезпечення відмовостійкості необхідно використати таке середовище розробки, яке підтримує високу доступність і максимальний захист даних, продуктивність, зниження витрат на розгортання, управління і підтримку проекту.

Усім цим вимогам найкраще відповідає Oracle GoldenGate (OGG). OGG – це проміжний продукт, призначений для роботи в гетерогенному середовищі з різними СУБД; переміщення тільки зафіксованих даних між різними платформами, що дозволяє використовувати мінімальну затримку; можливість переміщення змін по мережі TCP/IP, не вимагаючи наперед встановленого Oracle Net; використання власної системи файлів контрольних точок для підтримки цілісності транзакцій і без використання концепції мультиплексування, такої як в БД Oracle; швидке переміщення даних в резервну БД, яка може підтримувати аварійне відновлення.

OGG має модульну архітектуру, яка забезпечує гнучкість для вилучення і реплікування обраних записів даних, транзакційних змін і змін на мові опису структури даних DDL в різних топологіях/ Завдяки такій гнучкості, а також можливості фільтрації, перетворення і призначеної для користувача обробки, OGG підтримує багаточисельні бізнес-вимоги, в тому числі: відмовостійкість бізнесу і висока доступність; первинне завантаження і міграція БД; інтеграція даних; підтримка прийняття рішень в корпоративних сховищах даних. OGG постійно синхронізує систему резервних БД в режимі реального часу, щоб в разі потреби миттєво переключитися на резервну систему. Вторинна система відкрита для використання «тільки для читання» так само, як і «для запису». Також надається перемикання потоку даних із запасних БД (standby) на вихідну відразу ж після перемикання з режиму очікування в основний. Будь-які дані, які обробляються на запасній БД (Standby) в режимі очікування під час збоїв, переміщуються на вихідну БД як тільки остання стає доступною. Максимальна гнучкість, надійність і продуктивність досягається за рахунок конфігурації, зібраної з нез'язаних модулів, яка і є архітектурою OGG. Така архітектура розбиває процес переносу даних на чотири простих, і в той же час ефективних, етапи: Capture (Захоплення); Route (Маршрутизація); Transform (Перетворення); Delivery (Доставка).

На сайті підтримки Oracle опубліковані скріншти, які допомагають перевірити можливість реплікації схеми або БД, а також знайти можливі проблемні об'єкти. Конфлікти можуть виникати різні: унікальності; зміни; видалення. Багато уникати виникнення конфліктом наступними способами: конфігурувати sequence з діапазонами, що не пересікаються; жорстко рознести користувачів з різних БД, наприклад, за регіонами; знижувати затримку в реплікації.