

ДОДАТОК А
СЛАЙДИ ПРЕЗЕНТАЦІЇ

Атестаційна робота магістра

**Дослідження ефективності методів і моделей
резервного копіювання та аварійного
відновлення серверних даних з використанням
хмарного середовища**

Виконала: ст.гр. ІПЗм-18-2

Ніконова А.В.

Керівник проекту:

проф. Лесна Н.С.

Рисунок А.1 - Слайд 1 презентації

Актуальність дослідження. Мета роботи

- **Актуальність.** Обумовлена необхідністю систематизації процесу створення плану післяаварійного відновлення серверних даних та необхідністю вибору найбільш ефективного методу резервного копіювання з урахуванням сучасних метрик
- **Мета роботи.** Виявлення найбільш ефективних методів і моделей резервного копіювання та післяаварійного відновлення серверних даних

Рисунок А.2 - Слайд 2 презентації

Об'єкт і предмет дослідження

- **Об'єкт дослідження:** процес резервного копіювання та післяаварійного відновлення серверних даних
- **Предмет дослідження:** методи, моделі і технології резервного копіювання та післяаварійного відновлення серверних даних у SAP Hybris Commerce

Рисунок А.3 - Слайд 3 презентації

Наукова новизна та практична значимість

- **Наукова новизна.** Уперше запропонована методика та обрані критерії для оцінювання ефективності методів резервного копіювання і післяаварійного відновлення даних
- **Практична значимість.** Результати оцінювання ефективності та розроблені рекомендації дозволяють систематизувати планування післяаварійного відновлення серверних даних та зменшити збитки внаслідок позаштатних ситуацій

Рисунок А.4 - Слайд 4 презентації

Постановка задачі



- Аналіз існуючих методів, моделей і технологій резервного копіювання та післяаварійного відновлення даних у Hybris



- Розробка методики, визначення критеріїв ефективності



- Розробка програмної системи Standby: core-модуль, initialdata-модуль, ossaddon-модуль та storefront-модуль



- Оцінювання ефективності існуючих методів, моделей і технологій



- Розробка рекомендацій

Рисунок А.5 - Слайд 5 презентації

Аналізовані методи і технології резервного копіювання

- Технологія імпорту-експорту ImpEx
 - Менеджер ImpExManager
 - Дамп бази даних
 - Aimprosoft Commerce CBM
- } **ООТВ**
- } **ПЗ сторонніх виробників**

Рисунок А.6 - Слайд 6 презентації

Пошук Парето-оптимальної множини аналізованих методів

№		СУБД-незалежність	Макс. рівень експорту	Тип UI	Простота установки	Підтримка хмарного середовища
1	Технологія ImpEx	Присутня ✓	Каталог ✓	НАС ✓	ООТВ ✓	Відсутня ✗
2	ImpExManager	Присутня ✓	Окремі об'єкти ✗	Відсутній ✗	ООТВ ✓	Відсутня ✗
3	Дамп бази даних	Відсутня ✗	Кластер ✓	Desktop, термінал ✓	Стороннє ПЗ ✗	Відсутня ✗
4	Commerce CBM	Присутня ✓	Сайт ✓	Backoffice ✓	Розширення ✓	Присутня ✓

Рисунок А.7 - Слайд 7 презентації

Критерії оцінювання ефективності. Незалежні і залежні змінні

Незалежні змінні:

Об'єктивна точка відновлення RPO

Обсяг даних за годину

Обсяг RAM і потужність CPU

Наявність транзакцій

Залежні змінні:

Об'єктивний час відновлення RTO

Кількість звернень до БД

Відсоток завантаженості CPU

Рисунок А.8 - Слайд 8 презентації

RPO і RTO - основні критерії ефективності DRaaS



Рисунок А.9 - Слайд 9 презентації

Структура програмної системи

- **Standbycore** - основна логіка програмної системи: сервіси, аспекти, періодичні задачі генерації/порушення цілісності даних
- **Standbyinitialdata** - початкові дані для імпорту необхідних складових ізольованого каталогу
- **Standbyocaddon** - реалізація кросплатформеного RESTful API за допомогою підходу Operation Controle Center
- **Standbyoffice** - UI для управління процесами та графічного відображення результатів

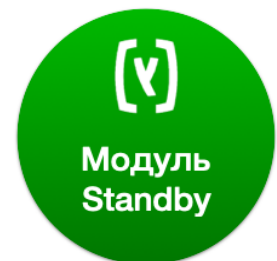


Рисунок А.10 - Слайд 10 презентації

Обрані засоби та технології



Рисунок А.11 - Слайд 11 презентації

Доменна модель рішення

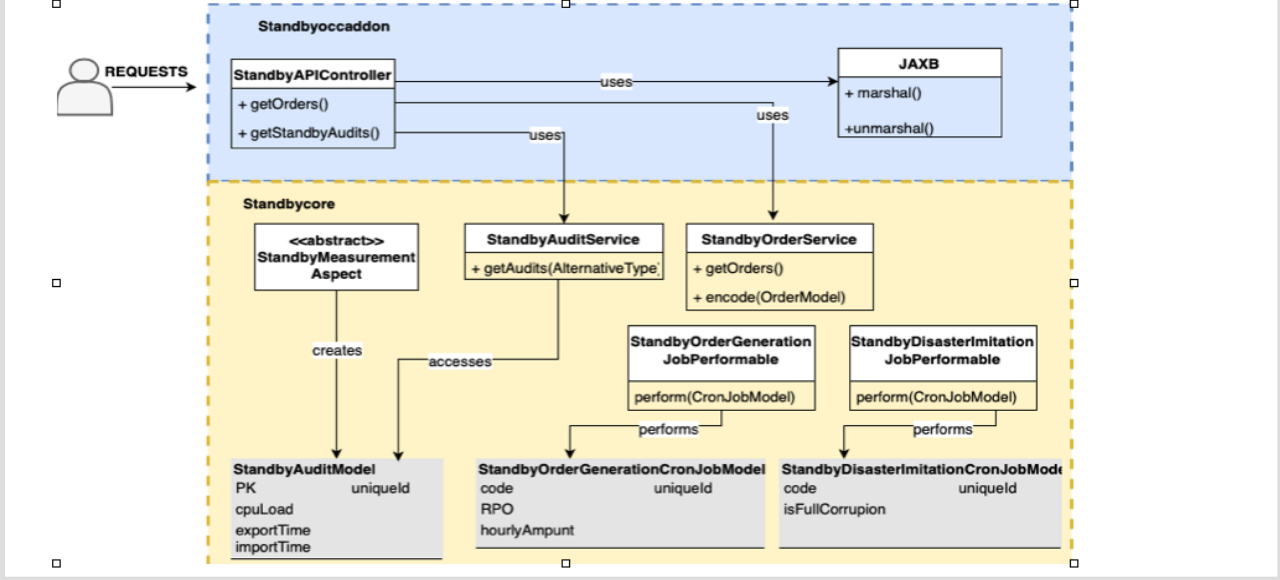


Рисунок А.12 - Слайд 12 презентації

Результати дослідження: Сценарій 1

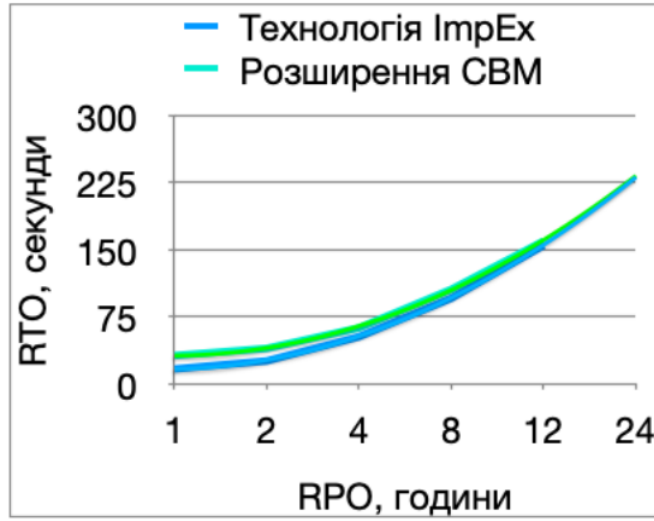


Рисунок А.13 - Слайд 13 презентації

Результати дослідження: Сценарій 2

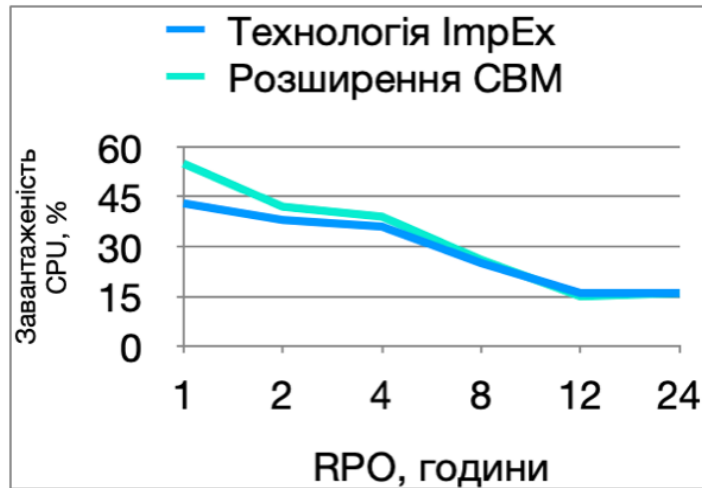


Рисунок А.14 - Слайд 14 презентації

Результати дослідження: Сценарій 3

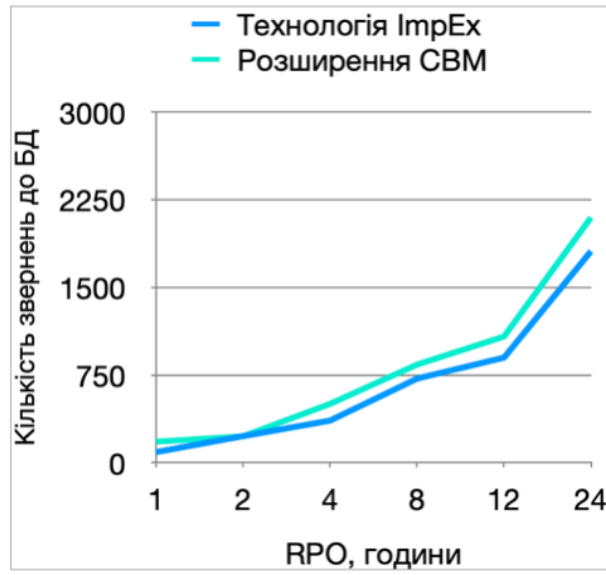


Рисунок А.15 - Слайд 15 презентації

Результати дослідження: Сценарій 4

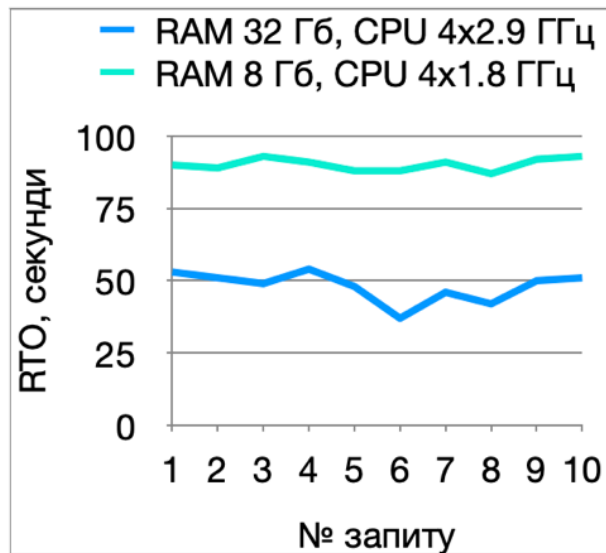


Рисунок А.16 - Слайд 16 презентації

Результати дослідження: Сценарій 5

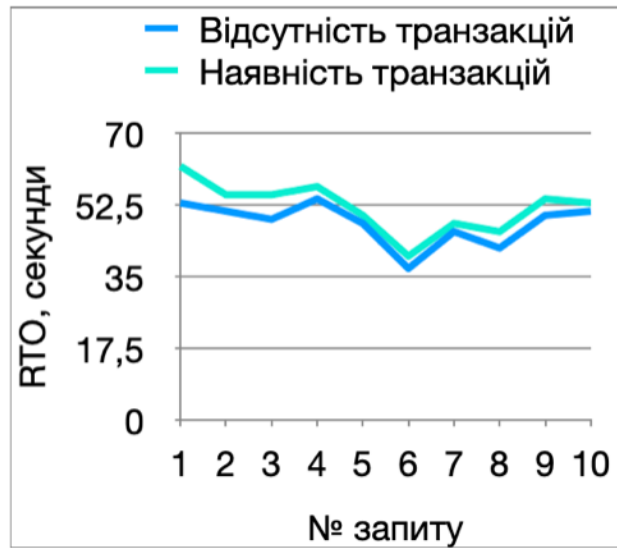


Рисунок А.17 - Слайд 17 презентації

Баланс між витратами через втрату даних та вартістю розробки бекапів

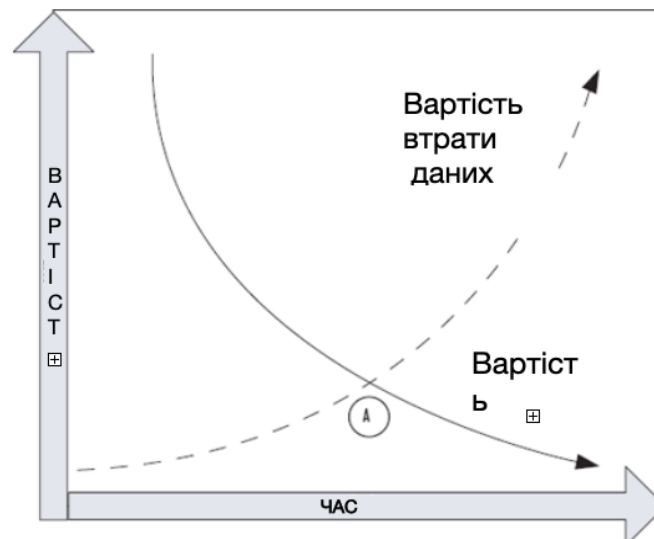


Рисунок А.18 - Слайд 18 презентації

Рекомендації

- Для **RPO ≤ 4 годин** бажаним є використання **ООТВ-технології ImpEx** завдяки значному виграшу щодо RTO, % навантаження на CPU, кількості звернень до БД
- Для **RPO > 4 годин** рекомендується розширення **SBM Commerce** завдяки невеликому програшу у RTO/CPU та підтримці хмарного середовища
- **Точка оптимального балансу** між вартістю втрати даних та вартістю створення резервних копій: **RPO ≈ 8 годин**
- **Підвищення потужності** сервера є недорогим способом **зниження RTO**
- Рекомендується **використання транзакцій**, оскільки вони незначним чином збільшують показник RTO

Рисунок А.19 - Слайд 19 презентації

Апробація

- Стаття “Дослідження методів і моделей резервного копіювання та післяаварійного відновлення у SAP Hybris Commerce” у міжнародному науковому журналі “Science Online”



Рисунок А.20 - Слайд 20 презентації

Висновки

- **Проведений моніторинг** стану проблеми та наявних методів і моделей резервного копіювання та післяаварійного відновлення в Hybris, **виявлена необхідність** розробки гнучкої системи оцінювання ефективності даних методів
- **Запропонована методика** та визначені критерії оцінювання
- **Спроектований і реалізований модуль Standby** для SAP Hybris Commerce, який дозволяє виділити найбільш ефективні методи створення резервних копій із подальшим їх відновленням
- **Проведене оцінювання ефективності** методів, моделей і технологій резервного копіювання і післяаварійного відновлення даних
- **Розроблені рекомендації** щодо створення плану післяаварійного відновлення у кластері Hybris

Рисунок А.21 - Слайд 21 презентації

[Демонстрація програмної системи]

Рисунок А.22 - Слайд 22 презентації

ДОДАТОК Б

ЛІСТИНГ КОДУ

ImportExportServiceTimeMeasurementAspect:

```

package ua.nure.anikonova.standby.aspect;

import com.puig.core.enums.StandbyAlternativeType;
import com.puig.core.model.StandbyAuditModel;
import de.hybris.platform.servicelayer.impex.ExportConfig;
import de.hybris.platform.servicelayer.impex.ExportResult;
import de.hybris.platform.servicelayer.impex.ImportConfig;
import de.hybris.platform.servicelayer.impex.ImportResult;
import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Pointcut;

/**
 * An aspect for time measurement of {@link
 de.hybris.platform.servicelayer.impex.impl.DefaultImportService} execution
 *
 * @author<a href="mailto:anna.nikonova@nure.ua">Anna Nikonova</a>
 * @since 20200415 for [Disaster recovery methods and models research Diploma]
 */
@Aspect
public class ImportExportServiceTimeMeasurementAspect extends
StandbyMeasurementAspect {

    @Pointcut("execution(*de.hybris.platform.servicelayer.impex.impl
.DefaultImportService.importData(..))
    && args(importConfig)")
    public void doImportData(final ImportConfig importConfig) {
        //import method pointcut
    }

    @Pointcut("execution(*de.hybris.platform.servicelayer.impex.impl
.DefaultExportService.exportData(..))
    && args(exportConfig)")
    public void doExportData(final ExportConfig exportConfig) {
        //export method pointcut
    }

    @Around("doImportData(importConfig)")
    public Object aroundDoImportData(final ProceedingJoinPoint pjp, final
        ImportConfig importConfig) throws Throwable {
        final long startTime = System.currentTimeMillis();
        final ImportResult importResult = (ImportResult) pjp.proceed();
        //method execution
        final long endTime = System.currentTimeMillis();
        final double cpuLoad = getProcessCpuLoad();

        //save audit to DB
        final StandbyAuditModel audit =
            modelService.create(StandbyAuditModel.class);
        audit.setImportTime(endTime - startTime);
        audit.setImportCPULoad(cpuLoad);
        audit.setAlternativeType(StandbyAlternativeType.IMPORT_EXPORT_SERVICE);
        audit.setMethodName(getMethodName(pjp));
        audit.setChecksumAfter(getOrdersChecksum());
        return importResult;
    }
}

```

```

    }

    @Around("doExportData (exportConfig)")
    public Object aroundDoExportData (final ProceedingJoinPoint pjp, final
        ExportConfig exportConfig) throws Throwable {
        final long startTime = System.currentTimeMillis();
        final ExportResult exportResult = (ExportResult) pjp.proceed();
        final long endTime = System.currentTimeMillis();
        final double cpuLoad = getProcessCpuLoad();

        final StandbyAuditModel audit =
            modelService.create (StandbyAuditModel.class);
        audit.setExportTime (endTime - startTime);
        audit.setImportCPULoad (cpuLoad);
        audit.setAlternativeType (StandbyAlternativeType.IMPORT_EXPORT_SERVICE);
        audit.setChecksumBefore (getOrdersChecksum());
        audit.setMethodName (getMethodName (pjp));
    return exportResult;
    }
}

```

StandbyMeasurementAspect (абстрактный класс):

```

protected String getOrdersChecksum() {
    final CatalogVersionModel catalogVersion =
        catalogVersionService.getCatalogVersion (STANDBY_CATALOG_ID,
            ONLINE_VERSION);
    return CollectionUtils
        .emptyIfNull (standbyOrderService.getOrders (catalogVersion))
        .stream()
        .map (standbyOrderService::encode)
        .collect (Collectors.joining ("\n"));
}

```

StandbyApiController:

```

@Secured ("ROLE_TRUSTED_CLIENT")
@RequestMapping (value = "/standby/orders", method = RequestMethod.GET)
@ResponseBody
@ApiOperation (value = "Get encoded orders checksums.",
    notes = "Used to scrap all orders from Standby Test Catalog",
    authorizations =
        {@Authorization (value = "oauth2_client_credentials")})
@ApiBaseSiteIdAndCurrAndLangParam
public List<String> getStandbyOrders() {
    final CatalogVersionModel catalogVersion =
        catalogVersionService.getCatalogVersion (STANDBY_CATALOG_ID,
            ONLINE_VERSION);
    final List<OrderModel> orders =
        standbyOrderService.getOrders (catalogVersion);
    final List<String> encodedOrders =
        CollectionUtils.emptyIfNull (orders).stream()
            .map (standbyOrderService::encode)
            .collect (Collectors.toList());
    return encodedOrders;
}

```

ДОДАТОК В

НАУКОВІ ПУБЛІКАЦІЇ

Стаття у міжнародному науковому журналі “Science Online”

International Electronic Scientific Journal “Science Online” <http://nauka-online.com/>

Інформаційні технології

УДК 004.7

Ніконова Анна Вадимівна*студентка**Харківського національного університету радіоелектроніки***Лесна Наталя Совєтівна***кандидат технічних наук,**професор кафедри програмної інженерії**Харківський національний університет радіоелектроніки***ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ І МОДЕЛЕЙ
РЕЗЕРВНОГО КОПІЮВАННЯ ТА ПІСЛЯВАРІЙНОГО
ВІДНОВЛЕННЯ ДАНИХ У SAP HYBRIS COMMERCE**

***Анотація.** У роботі розглянуті сучасні методи створення бекапів із подальшим післяаварійним відновленням у популярній платформі Hybris. Особлива увага дослідження приділена проблемі пошуку оптимального балансу між необхідною частотою створення резервних копій даних та продуктивністю сервера в контексті електронної комерції.*

***Ключові слова:** електронна комерція, післяаварійне відновлення як послуга, ефективність, DRaaS, RPO, RTO, Hybris, Java.*

Проблема і актуальність дослідження. 2020 рік став періодом винятково швидкого зростання обсягів сховищ даних - як активних, так і резервних, що використовуються на вимогу у випадку пікових навантажень на сервері [1]. Це глобальне явище пов'язане переважно із потребами бізнесу, обумовленими розвитком електронної комерції. Водночас, 71% американських ІТ-компаній визнають, що аспектом їхньої поточної інфраструктури зберігання, який у першу чергу потребує

International Electronic Scientific Journal “Science Online” <http://nauka-online.com/>

Рисунок В.1 - Сторінка 1 статті

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

вдосконалення чи розширення, є саме здатність до післяаварійного відновлення [2].

Із корпоративної точки зору, втрата або порушення цілісності інформації може коштувати сотні тисяч доларів у грошовому, часовому й навіть репутаційному еквівалентах [3].

Огляд поточного стану об'єкта дослідження. Найважливішим критерієм для бізнесу є надійний стек технологій обраної платформи. Найпопулярнішим програмним рішенням для електронної комерції є SAP Hybris Commerce, на якій реалізовані веб-магазини таких великих корпорацій, як Porsche, Volkswagen, Nikon, Philips, Indesit, Deutsche Post, Pasa Rabbane та інші.

Hybris надає багато готових і потужних рішень для управління продуктами, автентифікації і контролю доступу для користувачів та груп, процесу створення замовлень від додавання продукту в кошик до оплати й доставки. Платформа дозволяє створення користувацької бізнес-логіки. Існує також багато додаткових модулів для інтеграції із хмарними провайдерами (Amazon S3, Microsoft Azure), медіа-ресурсами (CloudFlare), сервісами оплати (PayPal, LiqPay) тощо. Також Hybris забезпечує швидке перемикання між різними типами СУБД (MySQL, Oracle SQL, SAP HANA, Postgre SQL, HSQL) без необхідності зміни коду.

У той же час на даний момент у платформі не існує єдиного стандартизованого методу створення резервних копій та подальшого відновлення даних із бекапів. Така задача є нетривіальною, і майже на кожному новому проєкті розробники витрачають понад 40-80 годин на написання логіки користувацьких сервісів для експорту та імпорту даних. Окрім того, поки що не існує програмних засобів для оцінки ефективності тих чи інших методів створення резервних копій у середовищі Hybris. Тому процес планування післяаварійного відновлення є досить проблематичним і неструктурованим.

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Рисунок В.2 - Сторінка 2 статті

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Пошук оптимальної технології для експорту та імпорту даних із метою резервного копіювання є об'єктом дослідження ІТ-компаній, які співпрацюють із SAP Hybris Commerce. Так, українська компанія Aimprosoft створила модуль Commerce Cloud Backup and Migration (Commerce CBM), який автоматизує створення бекапів і їх використання на рівні сайтів чи версій каталогів [4].

Німецька компанія SAP, яка є власником сімейства продуктів Hybris, працює над покращенням показників технологій імпорту, представлених "із коробки". До них належать методи із шару сервісів та традиційний менеджер імпорту-експорту, представлений на рівні Jalo. Існують дослідження білоруської компанії EPAM, присвячені підвищенню швидкості імпорту даних за рахунок відмови від стандартного протоколу ImpEx та використання альтернативної технології, яка однак потребує додаткової Java-конфігурації і не є гнучкою.

Як бачимо, всі існуючі методи резервного копіювання в Hybris потребують значних ресурсів сервера. Водночас цінність втрачених даних може бути дуже високою. Саме тому необхідним є пошук оптимального балансу між частотою створення бекапів, яка визначається об'єктивною точкою відновлення RPO, та швидкістю системи, яка описується комбінацією різних метрик (рівень завантаженості CPU, кількість звернень до бази даних тощо). Очікуваним результатом є така комбінація, при якій досягається найнижчий рівень об'єктивного часу RTO, тобто проміжку часу від аварії до повного відновлення стану системи.

Мета дослідження. Дана робота містить дослідження існуючих методів резервного копіювання та післяаварійного відновлення даних у Hybris. Метою дослідження є виявлення оптимального балансу між показником RTO та швидкістю комп'ютера, а також формування рекомендацій щодо розроблення плану післяаварійного відновлення для SAP Hybris Commerce шляхом створення тестового середовища, імітації

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Рисунок В.3 - Сторінка 3 статті

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

порушення цілісності даних та заміру показників ефективності використаних методів при різних конфігураціях.

Дослідження методів резервного копіювання та післяаварійного відновлення з використанням технології імпорту-експорту ImpEx та розширення Commerce SVM здійснюється на основі реалізації періодичних задач генерації тестових даних замовлень в ізольованій версії каталогу. Генерація відбувається із нормальним розподілом зі змінними значеннями RPO, обсягу замовлень на сайті за годину, можливістю синхронного й асинхронного виконання. Прототипом аварії на сервері є моделювання порушення цілісності даних - повного або часткового - в тестовому середовищі. Заміри ефективності роботи сервісів та утиліт виконані за допомогою аспектно-орієнтованого програмування, яке підтримується в Hybris засобами AspectJ.

Для обчислення відсотку втрати даних при післяаварійному імпорті реалізовані сервіс та API з функцією експорту тестових даних. HTTP-запит для експорту здійснюється до та після "аварії". Отримана відповідь у форматі JSON десериалізується засобами JAXB, таким чином порівнюється стан об'єктів. Також засоби Java дозволяють оцінити відсоток завантаженості CPU при використанні того чи іншого методу.

Для серії експериментів використані задачі періодичної генерації сутності OrderModel - замовлень на сайті - зі значеннями RTO: 1, 2, 4, 8, 12, 24 години, та фіксованим щогодинним коефіцієнтом нових значень у базі даних. За час RTO прийнятий час, необхідний для здійснення імпорту із бекап-носія, створеного даною альтернативою під час останнього експорту.

Для отримання достовірних результатів для кожної комбінації RPO та альтернативи, що аналізується, виконано 10 запусків задачі. Найпріоритетнішим завданням є визначення залежності об'єктивного часу

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Рисунок В.4 - Сторінка 4 статті

відновлення RTO від об'єктивної точки відновлення RPO. Усереднені значення серії експериментів показані на рисунку 1.

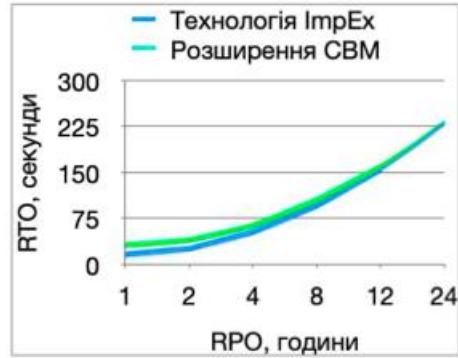


Рис. 1. Залежність RTO від RPO

Також важливим є встановлення залежності рівня навантаження на процесор від обраної RPO. Результати серії експериментів показані на рисунку 2.

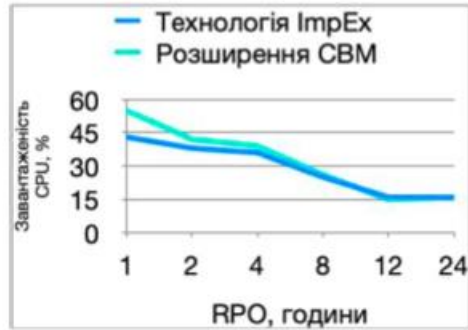


Рис. 2. Залежність завантаженості CPU від RPO

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Для визначення оптимальної точки між RTO, збільшення якої еквівалентне збільшенню вартості втрати даних, та RPO, ріст значень якої означає зниження вартості бекапів (тобто навантаження на сервер), слід визначити точку перетину графіків залежності RTO від RPO та RTO від навантаження на систему, яке у контексті даної моделі формується як відсоток завантаженості процесора. Така оптимальна точка позначена на рисунку 3 як А.

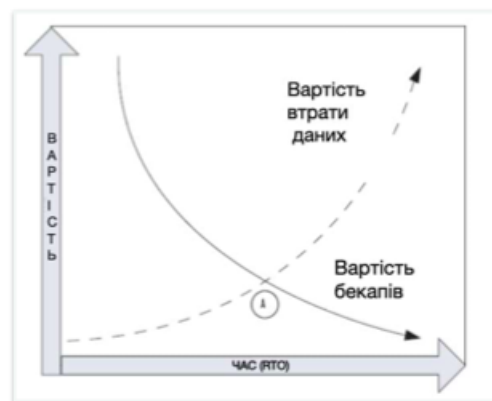


Рис. 3. Оптимальна точка балансу між RTO та RPO

Після нормалізації отриманих графіків залежностей було виявлено, що оптимальна точка балансу знаходиться приблизно на рівні 8-годинного RPO. Тобто із заданими коефіцієнтами щогодинної кількості нових даних (замовлень) на сервері та технічними характеристиками системи найкращою (із точки зору балансу між безпекою даних та швидкістю сервера) опцією є створення бекапів тричі на добу, тобто раз на 8 годин.

Висновки. У ході дослідження була розглянута проблема визначення ефективності методів резервного копіювання та післяаварійного відновлення даних на сервері SAP Hybris Commerce.

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Створена програмна система для генерації записів у тестовому середовищі, запуску задач експорту, імітації порушення цілісності даних, а також імпорту. Реалізовано визначення швидкодії системи (рівня завантаженості CPU), часу роботи методів імпорту (післяварійного відновлення з бекап-носія).

Завдяки серії експериментів визначено рівень залежності RTO від RPO, а також відсоток завантаженості CPU від RPO для OOTB-технології ImpEx та розширення Aimprosoft Commerce CBM. Встановлено, що для низьких (до 4 годин) значень RPO бажаним є використання OOTB-методів. Водночас на великих RPO оптимальність розширення CBM практично наближається до значень "коробочних" технологій. Враховуючи факт автоматичної генерації скриптів та можливості прямого запису резервних копій у хмару, CBM Commerce рекомендується для використання у системах з RPO, більшим за 4 години.

Також експериментально виявлено точку оптимального балансу між вартістю втрати даних та вартістю створення резервних копій. При заданих характеристиках системи такою точкою є RPO, близьке до 8 годин, тобто здійснення процесу бекапу тричі на добу.

Література

1. Gai S. Building a Future-Proof Cloud Infrastructure: A Unified Architecture for Network, Security, and Storage Services. Boston, MA: Addison-Wesley, 2020. 118 p.
2. Harvard Business Review Press. HBR Guide to Data Analytics Basics for Managers. Boston, MA: Harvard Business Review Press, 2018. 256 p.
3. Stevens B. DRaaS basics: Veeam special edition. Vaar: Veeam Press, 2015. 110 p.
4. Machnev A. Functional Specification Document: Hybris Commerce CBM v.1.0. Kharkiv: Aimprosoft, 2020. 4 p.

International Electronic Scientific Journal "Science Online" <http://nauka-online.com/>

Рисунок В.7 - Сторінка 7 статті



Рисунок В.8 - Свідоцтво про публікацію