**4. Експериментальний розділ**

В попередніх розділах даної МКР виконувалося дослідження можливості побудови конфігурованого сховища даних на базі розподіленої платформи Hadoop та проектування програмного рішення, на основі виконаних досліджень. Система була спроектована і побудована з урахуванням всіх основних вимог що ставляться до типових сховищ даних, а також з підтримкою простої конфігурації системи в залежності від предметної області, в якій вона буде використовуватися. В даному розділі виконується перевірка коректності роботи системи та її функціональне тестування.

Тестування продуктивності системи не входить в межі даної роботи і потребує наявності достатньо великих обсягів експерементальних даних та відповідної інфраструктури (кластера великої кількості машин). Таке тестування повинне проводитися за наявності деталізованих вимог до сховища даних та з урахуванням інфраструктурних можливостей. Майже лінійна масштабованість платформи Hadoop вже була підтверджена в лабораторних та реальних умовах і є описана в багатьох наукових доповідях.

Даний розділ покриває тільки функціональне тестування системи і виконує валідацію лише базового функціоналу, що є характерним для усіх сховищ даних, незалежно від предметної області. Такий функціонал включає наступні елементи:

* коректне завантаження структурованих даних з СКБД в розподілену файлову систему;
* коректне завантаження окремих файлів з файлових серверів в розподілену файлову систему;
* побудова таблиць фактів з заданої множини вхідних даних;
* побудова таблиць вимірів різних типів з заданої множини вхідних даних;
* надання інтерфейсу до підготованого сховища даних у вигляді, зручному для використання споживачами даних.

Оскільки розроблене рішення є фреймворком і не містить в собі ніяких вбудованих правил обробки, окрім як правил формування таблиць вимірів різних типів, то характер даних для виконання тестування не є важливим. Основною вимогою до цих даних є наявність таких полів:

* дата появи запису;
* унікальний ідентифікатор запису;
* значення яке несе цінність в процесі опрацювання даних (значення яке використовується для побудови аналітичного звіту).

Для валідації системи було згенеровано штучні дані, які відповідають всім вищевказаним вимогам. Кількість тестових даних не є великою, адже вони необхідні тільки для перевірки функціональної правильності системи.

Набір тестових даних складається з чотирьох таблиць наповнених штучно згенерованими даними. В кожній таблиці містяться дані, що використовуватимуться для побудови окремої логічної частини вітрини даних: таблиця фактів, таблиці вимірів першого, другого та третього типів. Ідентичні дані містяться в текстових файлах на локальній файловій системі, яка виконує роль файлового сервера. Для легкості тестування дані були вибрані ідентичними. Записи збережені у форматі розділеному вертикальними лініями (‘|’). Кожне поле в записі відділяється від сусіднього вертикальною лінією. Поширеним форматом є також розділяння комами, але такий спосіб не завжди підходить, адже інколи значення полів можуть містити символи коми. Система сховища даних буде розгорнута у віртуальному середовищі VirtualBox, з використанням уже готового, доступного у вільному доступі, віртуального образа Hortonworks HDP 2.2 Sandbox. База даних з тестовими таблицями працює на хості, засобами якого виконується віртуальне середовище. Для тестових цілей була обрана open-source СКБД MySQL. Текстові файли з тестовими даних знаходяться в локальній файловій системі того ж хоста, на якому працює СКБД.

Запуски виконуються шляхом надсилання послідовностей виконання (описаних в форматі xml, з урахуванням вимог, що ставляться засобом координації виконання Oozie) через командну стрічку на сервер-координатор. Послідовності виконання повинні знаходиться в розподіленій файловій системі(HDFS), а конфігураційний файл з параметрами виконання – на локальній файловій системі, звідки виконується запуск.

Запуски на виконання проводилися в кілька етапів. Спочатку для кожного джерела даних(таблиця в СКБД або текстовий файл на файловійс системі) окремо. Після кожного запуску перевірявся результат кожного з проміжних кроків та послідовності дій загалом. Також виконувався запуск комібонованої послідовності. В межах цієї послідовності виконувалася обробка з усіх джерел даних за один запуск.

Запуски виконувалися тричі для кожного з виду вхідних даних. Таким чином унеможливлювалося випадкове співпадіння отриманих та очікуваних результатів.

Також для перевірки роботи системи використовувалися дані, з завідомо введеними помилками (дані без необхідних полів в записах). Такі запуски виконувалися для підтвердження коректної обробки системою помилкових ситуацій та некоректних файлів.

Результати роботи кожного з запусків порівнювалися з очікуваними результатами. Всі запуски були успішні а результат роботи співпав з тим, який очікувався при запуску.

Після запуску програмного рішення на розподіленій файловій системі було отримано дані, що раніше знаходилися в БД та на окремому файловому сервері. Дані були приведені до вигляду, якого вимагає структура сховища даних та перетворені згідно з описаними вище правилами. Після одержання даних в HDFS, до них був наданий SQL-інтерфейс. Це було виконано за допомогою створення зовнішньої таблиці засобами інструменту Apache Hive. Після цього до даних можна було звертатися як інтерактивно (шляхом викоання SQL-запитів), так і шляхом інтеграції з зовнішніми аналітичними інструментами, які через ODBC-міст зверталися до збережених даних. Швидкість виконання таких запитів для невеликої кількості збереженої інформації є порівняно малою, що пояснюється накладними витратами на кожний з таких запусків. Але перевагою такого рішення є те, що при значному зростанні кількості даних, до яких виконуються запити, час доступу зростає дуже повільно.

Великою перевагою наявності невеликих віртуальних кластерів на локальній машині є можливість тестування окремих частин програми та системи вцілому одразу під час розробки. Функціональне тестування такого коду може відбуватися негайно після його написання, без потреби надсилання його на віддалений кластер. Таке тестування не дозволяє виявити деяких проблем, що можуть зявитися при запуску в справжньому розподіленому середовищі. Тестування в розподіленому середовищі є невідємною частиною процесу розробки таких програмних рішень, як сховища даних і потребує ретельного виконання. Але такий спосіб валідації системи може бути виконаний вже після попередньої перевірки правильності роботи системи локально.

**ВИСНОВОК**

В цьому розділі було проведено тестування та валідацю коректності роботи системи, розробленої під час виконання даної МКР. Основною метою проведених експериментів було встановити правильність виконання поставлених до системи вимог. Було сформульовано основні сценарії для тестування та у відповідності з ними проведено ряд запусків системи з набором штучно-згенерованих даних. Для уникнення можливості випадкового співпадіння отриманих результатів з очікуваними, запуски відбувалися кілька разів декількома способами. Після запуску, результати кожного з проміжних кроків та кінцеві результати порівнювалися з очікуваними. На всіх етапах тестування результати запусків співпадали з очікуваними, що свідчить про коректність роботи системи вцілому.

Дані експерименти не включали перевірки продуктивності, стійкості до навантажень а також інших поширених видів тестувань програмних продуктів. Основною метою було перевірити функціональну правильність виконуваних операцій та підтвердити можливість виконання всіх необхідних операцій збудованою системою. Як можна робити висновок з отриманих результатів – сховища даних на основі платформи Hadoop можуть повністю покрити функціонал, що до цього часу забезпечувався програмними системами на основі СКБД та програм для перетворення даних. Як було описано раніше, такі рішення можуть набагато легше масштабуватися і потребують менших затрат на побудову апаратної інфраструктури.