УДК 004.4'24

ДОБРЯНСЬКИЙ Б. І.,
САВАСТРУ С. В.

МОДЕЛЮВАННЯ ПРОГРАМНОЇ КОНФІГУРАЦІЇ ETL-ПРОЦЕСУ

У даній роботі було здійснено огляд підходів до організації, проведення та моніторингу ETL-процесів у загальному випадку. Проаналізовано основні переваги та недоліки кожного з підходів. Надано перелік рекомендацій до використання кожного з підходів для підвищення загальної ефективності організації роботи ETL-процесів. Проведено моделювання конфігурації ETL-процесу.

ETL, ETL-ПРОЦЕС, КЛАУД-ТЕХНОЛОГІЯ, БАЗА ДАНИХ

This research examines the existing general approaches of ETL-process configuration, execution and monitoring. Defines the main advantages and disadvantages of each approach. Provides a list of key recommendations for each approach that should be taken into account in order to increase the general efficiency of the ETL-process management. Develops design of ETL-process configuration.

ETL, ETL-PROCESS, CLOUD-TECHNOLOGY, DATABASE

1. Вступ

На сьогодні, у часи, коли більшість організацій надає перевагу розгортанню своєї інфраструктури у клауді, дуже важливим є адаптація вже звичних та дуже важливих рішень до цього середовища. Одним з них є організація та проведення ETL-процесів, адже міграція даних між сховищами з попередньою з фільтрацією та трансформацією є одною з базових складових для пришвидшення роботи аналітиків, або для автоматизації міграції бази даних, наприклад, при переході на нову версію якогось модулю системи, що, як ми знаємо, є невід'ємним елементом процесу розробки. У першому випадку аналітик матиме змогу швидко налаштувати та виконати ETL-процес замість того, щоб під'єднуватися до бази напряму та виконувати усі операції вручну, а потім ще й завантажувати у кінцеву базу. У другому випадку це значно спрощує процедуру міграції між сховищами даних, навіть без змушеної зупинки роботи елементів інфраструктури, що взаємодіють з даними сховищами. Можливими користувачами системи можуть стати підприємства, внутрішня інфраструктура яких розгорнута у клауді. Це є оптимальним рішенням, бо така платформа стане повноцінним елементом їхньої інфраструктури. У іншому випадку їм треба буде приєднуватись до сторонніх системи, що скоріше за все призведе до появи вразливості до атак на систему, або ж до значного ускладнення обслуговування такої системи.

2. ETL-процес

Кінцевою метою нашої роботи є розробка клауд-платформи для створення, підтримки та моніторингу ETL-процесів. Для початку розберемося з ти, що таке ETL-процес. В загальному випадку під ETL-процесом розуміється процес, який складається з трьох етапів[1]:

- витягування даних (extract);
- трансформація даних (transformation);
- завантаження даних до кінцевого сховища даних (load).

На першому етапі відбувається під'єднання до початкових сховищ даних та витягування з них даних відповідно до початкової конфігурації ETL-процесу. На другому етапі ми проводимо фільтрацію та трансформацію даних, що отримали на першому. На цьому ж етапі ми проводимо поєднання даних з кількох джерел даних, якщо така задача наявна. Наприклад, якщо одна складова даних лежить у одному сховищі, а друга – у іншому сховищі, тоді нам треба зробити поєднання даних за деякою ознакою. На етапі вигрузки даних нам потрібно під'єднатися до цільового сховища даних та завантажити до нього дані[2].

3. Підходи до виконання ETL-процесів

Існує 2 основні підходи до виконання ETL-процесу:

- написання та мануальне виконання окремих скриптів на для конкретного ETL-процесу;
- використання готового програмного забезпечення, що дозволяє конфігурувати та виконувати ETL-процес за допомогою графічного інтерфейсу.

Розглянемо кожний підхід окремо.

Написання скриптів під конкретний ETL-процес звичайно має дуже великий ступінь гнучкості. По-перше, знімається більшість обмежень на джерела даних та кінцеві сховища даних. Головною умовою можливості роботи з ними стає наявність інструментів роботи з цим джерелом у обраній для написання ETL-процесу мові програмування. Аналогічно покращується гнучкість роботи на етапі трансформації даних. Але ж звичайно є й недоліки у такому підході[3].

Перший й найголовніший недолік – людина має вміти програмувати та знати мову програмування, що підійде для таких цілей на доволі високому рівні. А далеко не кожна людина, якій потрібні результати роботи ETL-процесу є програмістом.

Другий недолік – складність конфігурації на подальшої модифікації ETL-процесу, що забиратиме величезну кількість часу.

Також значно ускладнюється його виконання. Звичайно, якщо і джерела даних, і сховища даних у межах доступності з комп'ютера користувача, то виконати його можна доволі швидко. Проблеми виникають, коли ці бази даних знаходяться на віддаленому сервері або у хмарі. Тоді користувач буде вимушений заходити до цього серверу або хмари та виконувати ті скрипти там. Такий підхід є занадто складним та ризикованим з точки зору інформаційної безпеки.

Ще одним недоліком такого підходу можна виділити неможливість моніторингу стану ETL-процесу.

Усі ці недоліки значно перекриваються переваги у гнучкості. Тож у більшості випадків підхід є неоптимальним. Використовувати його доцільно лише тоді, коли такий ETL-процес є одноразовим, тож його написання та виконання забере менше часу, ніж налаштування програмного забезпечення для автоматичного виконання ETL-процесу.

Тепер розглянемо інший підхід, а саме використання готових програмних рішень для налаштування ETL-процесу. Головною перевагою такого підходу є налаштування за допомогою графічного інтерфейсу. Зазвичай, користувач подібного програмного забезпечення має можливість налаштовувати джерела даних зі списку вже підтриманих. Те ж саме відноситься й до кінцевих сховищ даних. Далі вже саме програмне забезпечення аналізує обрані джерела даних та пропонує обрати, наприклад, потрібні таблиці. Так само зазвичай налаштовується й етап трансформації. Тобто, ми бачимо вже першу й основну перевагу – користувач не обов'язково має знати якусь мову програмування, щоб налаштувати ETL-процес. Відповідно зменшується час конфігурації та полегшується його модифікація, а графічний інтерфейс також дозволяє візуалізувати елементи ETL-процесу. Те ж саме стосується й процесу його виконання. Використовуючи готове програмне рішення для ETL-процесу ми можемо у реальному часі проводити моніторинг стану процесу під час виконання та зберігати результати виконання.

Однак, слід зауважити, що зазвичай такі застосунки є пропрієтарними, платформо орієнтованими або непристосованими до клауду.

4. Аналіз існуючих технічних рішень для організації ETL-процесів

Розглянемо більш детально, які типи рішень для ETL-процесів існують. Такі рішення можна поділити на 3 категорії:

- Enterprise рішення;
- Open-Source рішення;
- Cloud-Based рішення.

Також іноді окремо виділяють внутрішні розробки компаній подібних рішень, що вони використовують у своїх бізнес процесах, але, за відсутності публічної інформації про них, робити це ми не будемо. Тож розглянемо кожен тип окремо.

Enterprise рішення для ETL-процесів зазвичай є найбільш досконалими з усіх існуючих, пропонують розвинений графічний інтерфейс та підтримують велику кількість різноманітних типів сховищ даних. Вони розробляються та підтримуються комерційними організаціями. Однак ціна за використання такого роду застосунків досить велика. Також розробники часто

перевантажують рішення різноманітним функціоналом, що робить програму більш складною для користувача та більш вимогливою до системних ресурсів.

Open-Source рішення для ETL-процесів на відміну від попередніх вільні для користування. Також кожен користувач може продивитися код такого рішення, щоб впевнитися у відсутності вразливостей. Однак, такі рішення зазвичай мають меншу кількість функціоналу та не гарантують постійну підтримку та повноту заповнення документації. Хоча в даному випадку обмеженість функціоналу не є значним недоліком, бо основні інструменти, що використовуються найчастіше там присутні. Також в більшості випадків Open-Source рішення можна доволі легко встановити клауду, зібравши перед цим програму у образ Docker.

Cloud-Based рішення для ETL-процесів – рішення які надаються провайдерами клауду. Головною їх перевагою швидкість роботи, доступність та гнучкість сервісів, тож система сама буде адаптуватись під потрібну кількість ресурсів для виконання ETL-процесу. Також, якщо користувач тримає свої дані у того ж клауд провайдера, то мають місце додаткові оптимізації під час виконання ETL-процесів. Значним недоліком такого підходу є те, що Cloud-Based рішення працює лише у середі обраного клауд провайдера[4]. Тобто, якщо дані для аналізу знаходяться у клауді іншого провайдера, їх треба буде перенести на у той, де буде виконуватися ETL-процес.

Табл. 1. Аналіз відомих програмних продуктів для роботи з ETL-процесами

	Talend Open Studio	**HevoData**	**SQL Server Integration Services (SSIS)**
Категорія	Open Source	Enterprise	Cloud Based
WEB-інтерфейс	Відсутній	Присутній	Відсутній
Можливість інтеграції з клаудом	Відсутня	Відсутня	Присутня
Можливість розширення за допомогою плагінів	Присутня	Відсутня	Присутня
Підтримка джерел даних (СКБД, CRM-системи, текстові файли різних форматів)	70 джерел	90 джерел	16 джерел
Вбудоване планування виконання ETL-процесів	Відсутнє	Присутнє	Відсутнє
Вбудований моніторинг виконання ETL-процесів	Відсутній	Присутній	Відсутній

5. Моделювання конфігурації ETL-процесів

Конфігурацію ETL-процесу можна визначити, як сукупність зв’язаних кроків його виконання. Можна виділити три основні кроки його виконання, а саме витягнення даних із джерел, фільтрація та трансформація даних та збереження даних до сховища[5]. Для кожного з кроків виділяємо відповідний елемент конфігурації:

- виймач даних;
- перетворювач даних;
- завантажувач даних.

Усі елементи так чи інакше взаємодіють один з одним. Виймач даних містить параметри з’єднання з джерелом даних та таблицю, дані з якої потрібні під час виконання ETL-процесу. Перетворювач містить посилання на джерела даних з числа виймачів та інших присутніх перетворювачів, структуру вихідних даних та конфігурацію фільтрації та перетворення даних у визначеному форматі. Завантажувач даних містить посилання на джерела даних аналогічно до перетворювача та параметри з’єднання з кінцевим сховищем даних.

Сукупність таких елементів, що відносяться до одного ETL-процесу, формують його конфігурацію. Таким чином конфігурацію ETL-процесу можна представити у вигляді напрямленого графу

без циклів. Приклад конфігурації та зв'язків у структурі ETL-процесу зображено на рисунку 1. Слід зауважити, що елементи відповідальні за вивантаження даних з джерел та за завантаження даних до сховищ мають спільну рису – кожен з них взаємодіє з базою даних тим чи іншим чином. В такому випадку доволі очевидним стає рішення виділити керування даними для взаємодії з базами даних окремо від цих кроків. Ми можемо зберігати інформацію про з'єднання з базами даних окремо та при конфігурації обирати джерела даних з тих, що вже присутні у системі. Звичайно, усі параметри, потрібні для з'єднання з базою даних потрібно зберігати у зашифрованому вигляді.

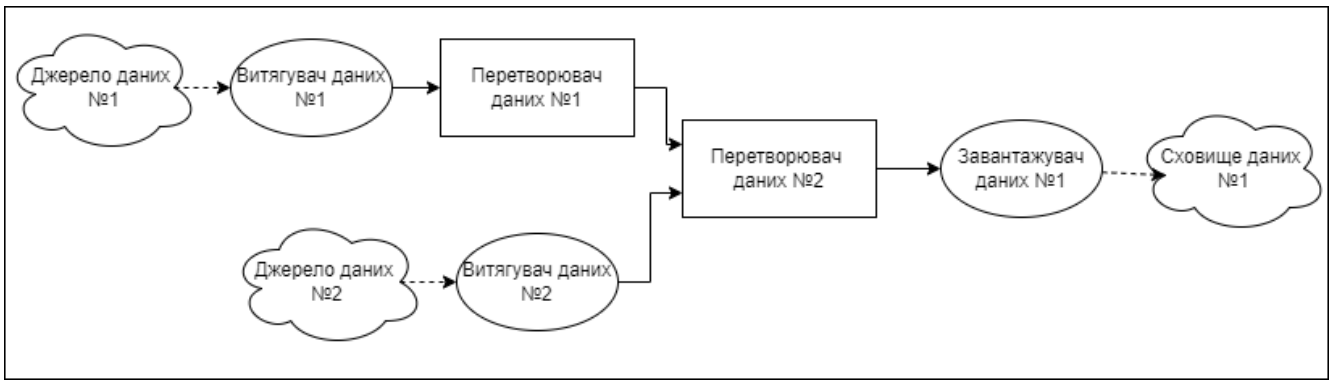

Рис. 1. Приклад конфігурації ETL-процесу

Висновки

Нами було здійснено огляд існуючих підходів до організації, проведення та моніторингу ETL-процесів. Проаналізовано основні переваги та недоліки кожного з них. Надано перелік рекомендацій до використання кожного з підходів для підвищення загальної ефективності організації роботи ETL-процесів. Виконано порівняння існуючих успішних проектів. Проведено моделювання конфігурації ETL-процесу та зображено у вигляді напрямленого графу без циклів.

Список літератури

1. Ralph., Kimball, (2004). The data warehouse ETL toolkit : practical techniques for extracting, cleaning, conforming, and delivering data.
2. David Loshin (2012). ETL (Extract, Transform, Load).
3. Zhao, Shirley (2017). "What is ETL? (Extract, Transform, Load) | Experian".
4. Rankins, Ray; Bertucci, Paul; Jennsen, Paul (2002). Microsoft SQL Server 2000 Unleashed (2 ed.).
5. Theodorou, Vasileios (2017). "Frequent patterns in ETL workflows: An empirical approach".