А. Д. Бондарчук, Д. И. Пумпур

# ETL-ПРОЦЕССЫ В ХРАНИЛИЩАХ ДАННЫХ GREENPLUM

*Рассматриваются возможности увеличения производительности обработки «сырых» данных за счёт горизонтальной масштабируемости системы на примере ETL-процессов в СУБД Greenplum*

## Введение

Мы сегодня живём и работаем в эпоху больших данных, которые генерируют не только информационные системы, с которыми работают люди, но и умные устройства и датчики Интернета вещей, а также множество других неодушевлённых машин. Для получения и обработки таких объёмов данных лучше всего подходит метод интеграции ETL. Он позволяет получать данные, проверять их, унифицировать, сохранять для последующей подготовки на их основе аналитической информации.

## I. ETL

ETL (Extract, Transform, Load) – один из основных процессов в управлении хранилищами данных, который подготавливает необработанные «сырыес данные для использования корпоративными системами, как правило, для отчётности и аналитики. Типичный вариант использования ETL – экспорт данных из внешних источников в хранилище данных. Основные этапы ETL-процесса приведены на рисунке 1.

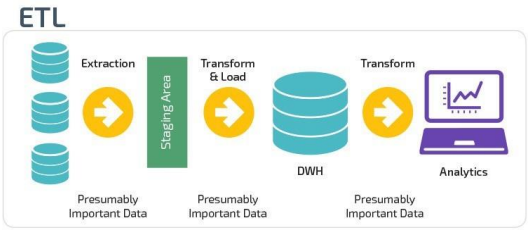


Рис. 1 – Основные этапы ETL-процесса

Extract. Это шаг, на котором датчики принимают на вход данные из различных источников (логов пользователей, копии реляционной БД, внешнего набора данных), а затем передают их дальше для последующих преобразований.

Transform. Это «сердцес любого ETL, этап, во время которого применяются бизнес-логика, фильтрация, группировка и агрегирование, чтобы преобразовать «сырыес данные в готовый к анализу набор данных. Эта процедура требует понимания бизнес-задач и наличия базовых знаний в области.

Load. Наконец, обработанные данные загружаются и отправляются в место конечного использования. Полученный набор данных может быть использован конечными пользователями, а может являться входным потоком к ещё одному ETL.

## II. Greenplum

Greenplum – open-source продукт, массивно-параллельная реляционная СУБД для хранилищ данных с гибкой горизонтальной масштабируемостью и столбцовым хранением данных на основе PostgreSQL. Благодаря своим архитектурным особенностям и мощному оптимизатору запросов, Greenplum отличается особой надёжностью и высокой скоростью обработки SQL-запросов над большими объёмами данных, поэтому эта MPP-СУБД широко применяется для аналитики Big Data в промышленных масштабах. При этом она хорошо ложится на облачный ландшафт и позволяет интегрироваться с ним намного глубже, чем другие базы данных, адаптируясь под экосистему.

## III. ETL в Greenplum

Greenplum как современное хранилище данных может обрабатывать большой объём данных, который обычно исчисляется петабайтами, но она не может генерировать такое количество данных самостоятельно. Данные, как правило, генерируются миллионами пользователей или встроенными устройствами. В идеале все источники передавали бы данные в Greenplum напрямую, но в действительности это невозможно, поскольку данные являются основным активом компании, а Greenplum – лишь одним из многих инструментов, которые можно использовать для создания продукта с помощью активов данных. Одним из распространённых решений этой проблемы является использование промежуточной системы для хранения данных. Когда Greenplum готова загрузить данные из промежуточной системы, важнейшей задачей становится эффективная загрузка данных – одна из основных задач ETL-инструментов. Пользователь сможет использовать сгенерированные данные после определённой задержки. Таким образом, ETL-инструменты помогают Greenplum надежно и эффективно загружать данные из внешнего источника.

## IV. ETL-инструменты Postgres

Для Greenplum доступны такие ETL-подобные инструменты, как pg_dump, copy.

Pg_dump – это инструмент командной строки, который позволяет извлекать данные из таблицы в файл. Это официальное решение от Postgres для резервного копирования. Он может выгружать данные в простой или сжатый файл, также он может выгружать только определение таблицы.

Copy – это SQL-команда Postgres. Её задача – обмен данными между таблицей и файлом. Он может выгружать содержимое таблицы с требуемым разделителем, escape-символом, заголовком в файл и наоборот. Данная команда может быть выполнена клиентом. Это означает, что он может использовать файл, расположенный на хосте psql, для таблицы на удалённом сервере.

У этих средств есть общий недостаток в виде отсутствия возможностей MPP (massive parallel processing, массово-параллельная архитектура). Все они должны связываться с мастером GPDB и использовать мастер для отправки данных. Мастер станет узким местом, если есть много сегментов, ожидающих сигнала от мастера. Рабочий процесс выполнения запроса показан на рисунке 2.

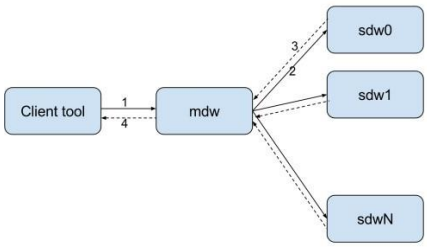


Рис. 2 – Процесс выполнения запроса в Postgres

1. Клиент отправляет запрос внешней таблицы мастеру (mdw).
2. Мастер отправляет команду каждому сегменту (sdwn).
3. Сегменты возвращают результат мастеру.
4. Мастер возвращает результат клиенту.

## V. ETL-инструмент gpfdist компании Greenplum

Gpfdist может загружать данные параллельно, позволяя каждому сегменту напрямую взаимодействовать с сервером gpfdist и не нужно сначала отправлять все данные мастеру, что является важной особенностью этого ETL-инструмента. Рабочий процесс выполнения запроса показан на рисунке 3.

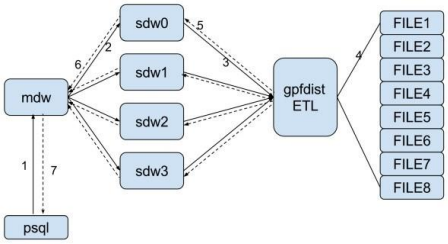


Рис. 3 – Процесс выполнения запроса в gpfdist

1. Клиент отправляет запрос внешней таблицы мастеру (mdw).
2. Мастер отправляет команду каждому сегменту (sdwn).
3. Каждый сегмент отправляет запрос на ETL / gpfdist-сервер с HTTP-запросом.
4. Gpfdist считывает файл в соответствии с информацией о пути. Затем он анализирует и разбивает каждую строку файла.
5. Gpfdist параллельно отправляет результат в разные сегменты.
6. Сегменты отправляют результат мастеру, когда он закончит чтение.
7. Мастер возвращает результат клиенту.

## VI. Выводы

Выбирая СУБД для решения enterprise-задач, необходимо составить в голове чёткое понимание того, какие задачи должна будет решать система, и учесть ряд факторов. Один из ключевых – возможность обрабатывать большие объёмы данных. В контексте enterprise этот фактор обычно принимает более резкую форму: возможность обрабатывать любые объёмы данных. Речь идёт о горизонтальной масштабируемости системы. Когда такая потребность есть, массивно-параллельные (MPP) СУБД зачастую оказываются рентабельнее, чем single-node. Для увеличения объёмов хранилища нет необходимости покупать очередную дорогую мощную машину на замену старой – достаточно добавить несколько более слабых. Как показано выше, одной из наиболее подходящих в таком случае СУБД является Greenplum – мощный и гибкий инструмент для аналитической обработки больших объёмов данных.

*Бондарчук Артем Дмитриевич*, студент кафедры информационных технологий автоматизированных систем БГУИР, artembondarchuck@yandex.by.

*Пумпур Дмитрий Игоревич*, студент кафедры информационных технологий автоматизированных систем БГУИР, dima210876@mail.ru.

*Научный руководитель: Трофимович Алексей Фёдорович*, заместитель декана факультета информационных технологий и управления БГУИР, старший преподаватель.