УДК 004.9 DOI:10.35803/1694-5298.2021.4.556-563

КАРТАНОВА А.ДЖ., ИМАНБЕКОВ Т.И.

[1]КГУСТА им. Н. Исанова, Бишкек, Кыргызская Республика

KARTANOVA A.DZH., IMANBEKOV T.I.

KSUCTA n. a. N. Isanov, Bishkek, Kyrgyz Republic

a.kartanova@mail.ru imanbekov_t@mail.ru

ОБЗОР МЕТОДОВ ОПТИМИЗАЦИИ ПРОИЗВОДИТЕЛЬНОСТИ ПРОЦЕССА ETL

OVERVIEW OF OPTIMIZATION METHODS FOR PRODUCTIVITY OF THE ETL PROCESS

Берилиштер базаларындагы жана берилиштер кампаларындагы процесстерди, операцияларды башкаруунун жана тездетүүнүн маанилүү суроолорунун бири - бул ETL - процесстер, маалыматтарды алуу, трансформациялоо жана жүктөө процесси. Бул процесстерди жөнгө салбастан, берилиштер кампасынын долбоорлорун ишке ашыруусу кымбат, татаал жана көп убакытты талап кылат.

Бул макалада ETL процесстеринин натыйжалуулугун оптималдаштыруу методдорунун серептери жана изилдөөлөрү келтирилген, ETL тутумунун иштешинин эң маанилүү көрсөткүчү - бул маалыматтарды иштеп чыгуу убактысы жана ылдамдыгы. ETL процесстеринин агымынын жалпыланган структурасынын маселелери каралып, ETL процессин оптималдаштыруу архитектурасы сунуш кылынат жана ETL тутумдарында маалыматты параллель иштетүүнүн негизги ыкмалары келтирилген, булар процессинин натыйжалуулугун жакшырта алат. Бүгүнкү күндүн эң актуалдуу болуп саналган берилиштер кампалары үчүн ETL-процесстерин аткаруу көйгөйү кеңири каралды.

Өзөк сөздөр: *берилиштер кампалары, өндүрүмдүүлүк, маалыматтардын сапаты, ETL процесси, маалымат агымдары.*

Одним из важных аспектов в управлении и ускорении процессов, операций в базах данных и хранилищах данных являются ETL процессы, процесс извлечения, преобразования и загрузки данных. Без оптимизации этих процессов реализация проектов в области хранилищ данных является дорогостоящей, сложной и трудоемкой задачей.

В данной работе приведен обзор и исследование методов оптимизации производительности ETL процессов, показано, что наиболее важным показателем работы ETL системы является время и скорость обработки данных. Рассмотрены вопросы обобщенной структуры потоков процесса ETL, предложена архитектура оптимизации процесса ETL и изложены основные методы параллельной обработки данных в ETL системах, позволяющие улучшить ее производительность. Подробно рассмотрена проблема производительности процессов ETL для хранилищ данных, как наиболее актуальная на сегодняшний день.

Ключевые слова: *хранилища данных, производительность, качество данных, процесс ETL, потоки данных.*

One of the important aspects in management and acceleration of processes, operations in databases and data warehouses is ETL processes, the process of extracting, transforming and loading data. These processes without optimizing, a realization data warehouse project is costly, complex, and time-consuming.

This paper provides an overview and research of methods for optimizing the performance of ETL processes; that the most important indicator of ETL system's operation is the time and speed of

data processing is shown. The issues of the generalized structure of ETL process flows are considered, the architecture of ETL process optimization is proposed, and the main methods of parallel data processing in ETL systems are presented, those methods can improve its performance. The most relevant today of the problem is performance of ETL processes for data warehouses is considered in detail.

Key words: *data warehouses, performance, data quality, ETL process, data streams.*

Введение. Ключевым активом предприятий и организаций являются систематизированные данные, хранящиеся в надлежащем формате, для того, чтобы лица, принимающие бизнес-решения (аналитики данных, руководители предприятий и менеджеры) могли принимать более быстрые и качественные решения. Поэтому хранилища данных становятся все более важным элементом в системах поддержки принятия решения. Организации вкладывают большие средства в создание и обслуживание своих хранилищ данных.

Для достижения цели расширенной бизнес-аналитики хранилище данных работает с данными, собранными из нескольких источников. Исходные данные могут поступать из систем собственной разработки, приобретенных приложений, сторонних синдикатов данных и других источников. Это может включать транзакции, производство, маркетинг, человеческие ресурсы и многое другое. В современном мире больших данных данными могут быть миллиарды отдельных кликов на веб-сайтах или массивные потоки данных от датчиков, встроенных в сложное оборудование и т.п.

Билл Инмон является одним из авторов концепции хранилищ данных (Data Warehouse). В 1990 году он дал определение хранилищу данных, как «предметно ориентированные, интегрированные, неизменчивые, поддерживающие хронологию наборы данных, организованные для целей поддержки управления», призванные выступать в роли «единого и единственного источника истины», обеспечивающие менеджеров и аналитиков достоверной информацией, необходимой для оперативного анализа и принятия решений [1].

Ральф Кимбалл другой из авторов концепции хранилищ данных, описывал хранилище данных как «место, где люди могут получить доступ к своим данным», а также сформулировал и основные требования к хранилищам данных: поддержка высокой скорости получения данных из хранилища, поддержка внутренней непротиворечивости данных, возможность получения, сравнения так называемых срезов данных, наличие удобных утилит просмотра данных в хранилище, полнота и достоверность хранимых данных, поддержка качественного процесса пополнения данных [2].

В общем, хранилище данных - это собрание огромных объемов данных, которые используются для поддержки системы принятия решений, и где применяются множество различных технологий, в первую очередь ориентированных на то, чтобы выполнять следующие функции: извлечение данных из разрозненных источников, их трансформация и загрузка в хранилище, администрирование данных хранилища, извлечение данных из хранилища, аналитическая обработка и представление данных конечным пользователям.

В связи с этим в средах хранилищ данных существуют различные проблемы, такие как извлечение данных из множества различных источников, интеграция, трансформация и консолидация огромных объемов данных, поступающих из множества различных разнородных систем. Обработка потоков данных от источника данных к получателю – хранилищу данных решаются комплексом программных средств получившее обобщенное название ETL (от англ. extraction, transformation, loading — «извлечение», «преобразование», «загрузка»).

Архитектуру хранилища данных, с позиции процесса ETL, можно представить в виде трех компонентов, таких как источник данных, содержащий структурированные данные в виде таблиц, совокупности таблиц или просто файла (формата TXT, TSV, CSV), промежуточная область, содержащая вспомогательные таблицы, создаваемые временно и

исключительно для организации процесса выгрузки и получатель данных – хранилище данных или база данных, в которую должны быть помещены извлеченные данные, как показано на рис.1.

Задача процесса ETL - извлечь и очистить данные из отдельной исходной системы, интегрировать их и загрузить в систему хранилища данных. Основная цель процесса ETL - предоставить нужные данные нужным людям в нужное время. Поскольку процесс ETL должен завершить свое выполнение в указанные временные рамки, для его достижения необходимы методы оптимизации. Создание потенциально организованных данных является одной из важнейших задач хранилищ данных, поэтому их создание - трудоемкая и сложная процедура. Без оптимизации процессов ETL разработка хранилищ данных является дорогостоящей, сложной и трудоемкой.

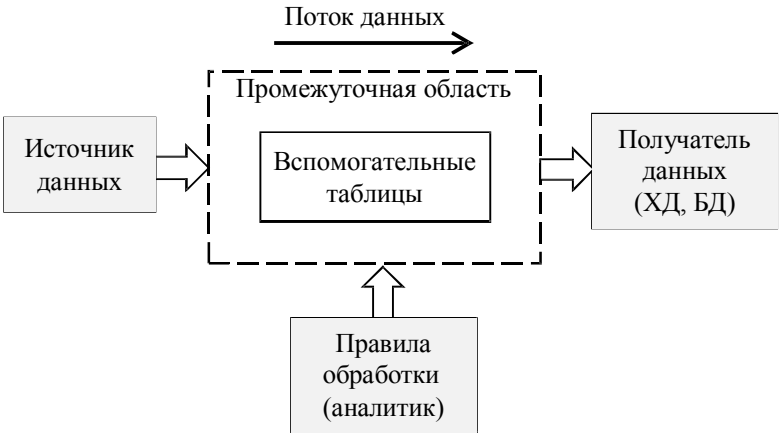

Рис. 1. Потоки данных в ETL

ETL — комплекс методов, реализующих процесс переноса исходных данных из различных источников в аналитическое приложение или поддерживающее его хранилище данных, как показано на рис.2.

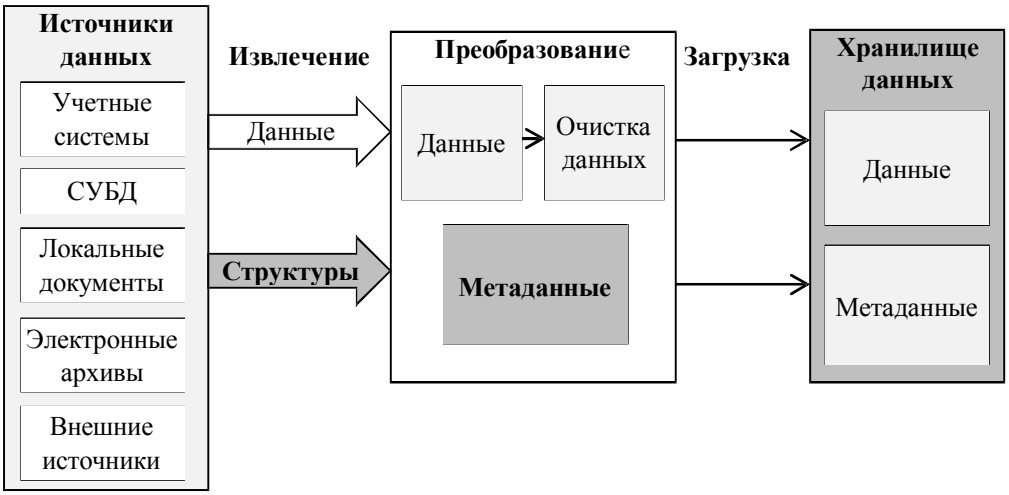

Рис. 2. Обобщенная структура процесса ETL

Методика решения. При выполнении процесса ETL доступно множество методов оптимизации. Эти методы различаются технологией оптимизации процесса ETL, но все они направлены на ускорение выполнения процесса ETL с оптимальным использованием

системных и временных ресурсов. В ходе исследования были изучены различные методы оптимизации процесса ETL.

В работе Сюфэн Лю и Надим Ифтихар [3] предложили метод оптимизации процесса ETL, направленного на реализацию разделения одного потока данных задания ETL на несколько параллельных независимых потоков данных процесса ETL. Авторы предлагают распределенную обработку процесса ETL, то есть многоузловую обработку процессов, обрабатывающих огромный объем данных, и распределение нагрузки рабочего процесса на распределенную многосерверную архитектуру ETL. Конечно, результаты экспериментов показывают, что предлагаемый метод позволяет достигнуть производительности процесса в 4,7 раза больше, чем у аналогичных инструментов ETL, но предлагаемая установка распределенной многосерверной архитектуры требует больших затрат и не подходит для организаций малого и среднего уровня.

Другой метод, предложенный Алкисом Симитсисом, Панос Вассилиадисом и Тимосом Селлисом [4], направлен на перестройку логического преобразования с целью повышения производительности и поддержания точности данных. В этом методе авторы предложили логическую оптимизацию процессов ETL, моделируя ее как задачу поиска в пространстве состояний, где каждое состояние представляет собой конкретную схему рабочего процесса в виде графа. Далее создается пространство состояний с помощью набора правильных переходов между состояниями, задаются условия генерации состояний, корректности и применимости переходов и применяются предложенные алгоритмов поиска, направленных на минимизацию затрат на выполнение рабочего процесса ETL.

Много работ посвящено моделирования архитектуре хранилища, процесса ETL. Например, в работе [5] авторы представили новое решение в разработке стандартной концептуальной модели для усовершенствования процесса ETL, реализующего операций извлечения, преобразования и загрузки. При построении модели, они основывались на трех подходах в моделировании процессов: первый - это отображение терминов и инструкций, второй - концептуальное моделирование, а третий - моделирование в среде UML.

В статье [6] авторы предложили улучшить традиционную архитектуру хранилища данных с возможностью функционирования хранилища данных в режиме реального времени. Так как постоянное и бесперебойное функционирование хранилища данных в онлайн формате позволяет получать актуальные данные с желаемой скоростью передачи.

В данной статье [7] представлено сочетание методов распараллеливания и использования общей кэш-памяти в распределенной системе хранилища данных. Согласно приведенной оценки, предложенный метод продемонстрировал улучшение производительности процесса ETL по времени на 7,1% для инструмента оптимизации Kettle и на 7,9% для инструмента Talend. Следовательно, распараллеливание может значительно улучшить процесс ETL. В результате, очевидно, что процессы управления и интеграции большими данными стали осуществляться просто и с приемлемой скоростью.

Результаты исследований. Рассмотрены различные подходы к оптимизации производительности процесса ETL, основанные на управлении потоками данных в ETL для оптимизации скорости извлечения, качества преобразованных данных и времени загрузки.

Методы секционирования и распараллеливания применяются в основном при оптимизации операций извлечения и загрузки. Поскольку процесс извлечения данных занимает много времени, он является первоначальным для параллельного выполнения последующих трех фаз. В то время как данные извлекаются, другой процесс преобразования выполняется, как только данные получены и они готовы для загрузки, в то время как загрузка данных начинается без ожидания завершения предыдущих этапов процесса ETL.

В ETL приложениях реализованы три метода параллельного преобразования данных.

Разбиение (секционирования) данных (data partitioning) - для обеспечения параллельного доступа, один последовательный файл разбивают на более мелкие файлы данных. Конвейерная обработка (data pipeling)) - это процесс разработки (подготовки, производства), программный конвейер, который позволяет одновременно работать нескольким компонентам на одном и том же потоке данных, например, идет обработка значения записи №1, но в то же время, происходит добавление двух полей в запись №2. Компонентная обработка (component processing) - одновременный ход нескольких операций над данными на разных потоках, в одном и том же задании, например, сортировка одного входного файла, в то же время удаление дубликатов другого файла. Обычно все три типа параллелизма работают объединено/параллельно в одном задании.

ETL инструменты, как правило, используются в широком круге специалистов, начиная от менеджера, бизнес-аналитика и заканчивая руководителями, ответственными за управление компанией, и желающих быстро импортировать большие наборы данных, провести анализ и получить визуальные отчеты. ETL приложения стали удобным инструментом с реализацией параллельной обработки, на который можно положиться, чтобы получить максимальную производительность процесса ETL при работе с большими объемами данных. ETL инструменты в большинстве случаев содержат графический интерфейс, который помогает пользователям легко преобразовывать данные, с помощью визуального картографа данных, в отличие от написания больших программ для анализа файлов и редактирования типов данных.

Примером ETL программного обеспечения является Talend - это инструмент ETL для интеграции данных. Он предоставляет собой программное решение для подготовки данных, обеспечения качества данных, интеграции данных в приложение, управления данными и большими данными.

Kettle (K.E.T.T.L.E - Kettle ETTL Environment) был недавно приобретен группой Pentaho и переименован в Pentaho Data Integration. Kettle - это ведущее приложение ETL с открытым исходным кодом, классифицируется как инструмент ETL. Однако концепция классического процесса ETL (извлечение, преобразование, загрузка) была немного изменена в Kettle, поскольку он состоит из четырех элементов, ETTL, что означает: извлечение данных из исходных баз данных, транспорт данных, преобразование данных, загрузка данных в хранилище данных.

В работе [7] приведены результаты экспериментальных исследований по оптимизации процесса ETL в хранилище данных за счет сочетания распараллеливания и использования общей кэш-памяти. На рис. 3 показана рекомендованная архитектура управления данными из баз данных и процесса ETL, где используются две комбинированные стратегии распараллеливания и использования общей кэш-памяти для оптимизации операций ETL.

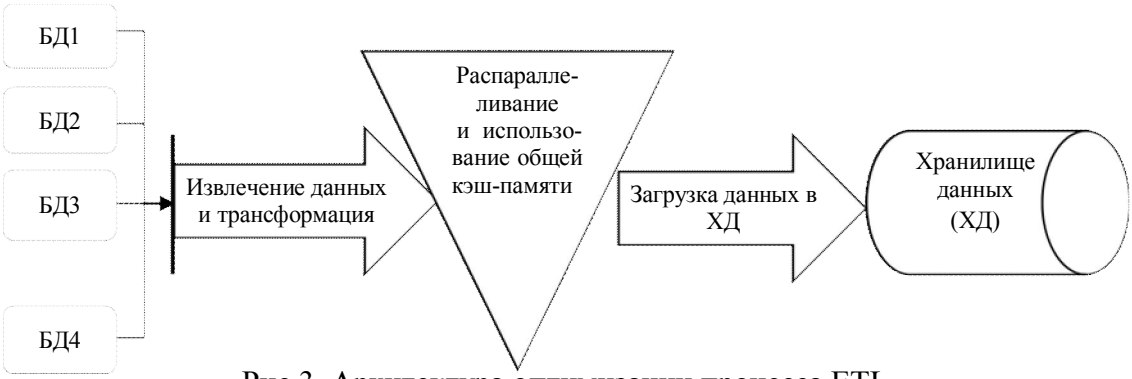

Рис 3. Архитектура оптимизации процесса ETL

Способ распараллеливания процессов в системе в данном исследовании основан на ядрах процессора, существующих в операционных средах. Отметим, что исходные данные их разных источников, распределенных баз данных извлекаются и загружаются в целевое оперативное пространство, при этом каждый процесс выбирается неактивным ядром обработки, после завершения обработки ядро ставится в очередь и готово к приему следующего процесса. Ядра поочередно получают и параллельно обрабатывают данные, затем передают их в хранилище данных.

В эксперименте использовали пять различных баз данных с разными объемами информации и с разным количеством записей. Кроме того, были исследованы общая кэш-память, различное количество ядер, а также последовательная и параллельная реализация процесса. Результаты представлены в табл.1.

Показано, что разница во времени последовательной реализации процесса ETL и параллельной обработки с общей кэш-памятью очень значительна. Функциональность параллельного процесса обработки в среднем в 263 раза лучше, чем при последовательном процессе обработки данных, как показано на рис.4-5.

Таблица 1 - Результаты вычислительных экспериментов

Объем, (Мб.)	Количество записей	Время, (сек.)	
		Последовательной обработки	Параллельной обработки
0,14305	100	5133	14
0,71526	500	28 111	27
1,43051	1000	54 834	39
2,86102	2 000	106 960	52
7,15256	5 000	208 638	65
14,3051	10 000	406 974	77
28,6102	20 000	793 853	90
42,9153	30 000	1 548 508	103

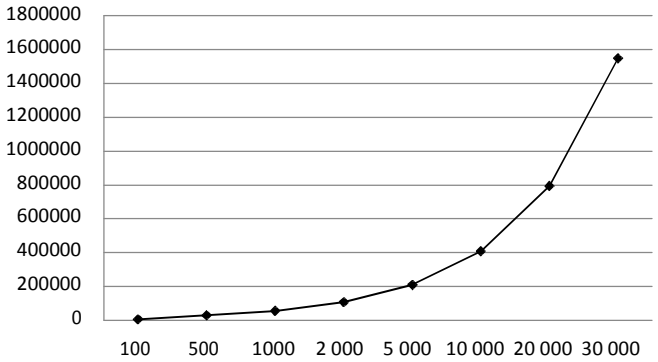

Рис.4. Зависимость времени последовательной обработки от количества записей

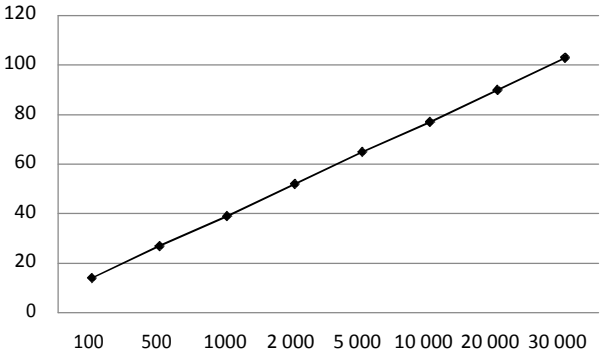

Рис.5. Зависимость времени параллельной обработки от количества записей

В табл. 2 показаны экспериментальные результаты по исследованию времени обработки потока данных инструментами ETL Kettle, Talend по сравнению с предложенным методом.

Таблица 2 - Сравнительные результаты экспериментов трех методов

Объем, (Гб)	Время обработки, (м/с)		
	Рекомендуемый метод	Инструмент Kettle	Инструмент Talend
1	927	2500	2500
2	1900	5100	5000
3	3150	7000	6700
4	6300	12500	12000
5	12200	15000	13200
6	13000	16000	14000
7	15500	20000	17000
8	20200	24000	22000

На рис. 6 показан сравнительный анализ рекомендованного метода с другими инструментами оптимизации ETL.

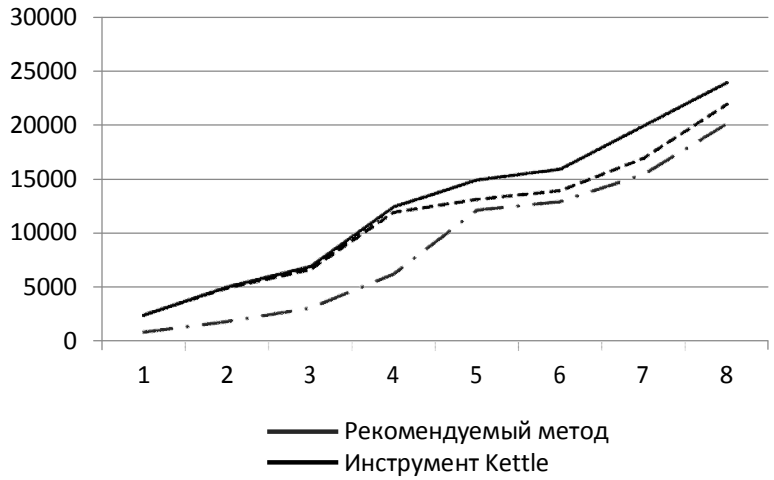

Рис.6. Сравнительный анализ методов оптимизации производительности ETL

Как показано, предложенный метод демонстрирует улучшение скорости примерно на 7,1% по сравнению с инструментом оптимизации Kettle и на 7,9% с инструментом Talend.

Выводы. В ходе исследования был сделан обзор методов оптимизации производительности ETL процессов, показано, что наиболее важным показателем работы ETL системы является время обработки данных, или, более точно, пропускная способность системы. Доказано, что эффективным способом увеличить производительность системы является выполнение процессов извлечения-обработки-загрузки данных ETL системы в параллели. Под этим подразумевается параллелизация чтения (записи) данных из систем-источников (приемников) и использование методов параллельного преобразования данных. Очевидно, на сегодняшний день направление хранилищ данных бурно развивается. Разрабатываются программные средства, как платформы для создания хранилищ данных, так и для управления процессом ETL, интеллектуального анализа данных. Проблемы оптимизации производительности процессов ETL для хранилищ данных остаются актуальными.

Список литературы

1. Inmon W. Building the Data Warehouse. – New York: John Willey & Sons, 1992.
2. Kimball R. The Data Warehouse Lifecycle Toolkit, 2nd Edition: Practical Techniques for Building Data Warehouse and Business Intelligence Systems/ Kimball R., Ross M., etc. – John Wiley & Sons, 2008.
3. Xiufeng Liu, Nadeem Iftikhar. An ETL optimization framework using partitioning and parallelization.SAC '15 Proceedings of the 30th Annual ACM Symposium on Applied Computing (pp. 1015-1022). USA: ACM New York, NY, USA ©2015.
4. Simitsis, A., Vassiliadis, P., and Sellis, T. Optimizing ETL Processes in Data Warehouses. In Proc. of ICDE, pp. 564-575, 2005.
5. S. H. A. El-Sappagh, A. M. A. Hendawi, A. H. El Bastawissy, "A proposed model for data warehouse ETL processes", Journal of King Saud University Computer and Information Sciences, Vol. 23, No. 2, pp.91-104, 2011.
6. Cuzzocrea A., Ferreira N., Furtado P. Enhancing Traditional Data Warehousing Architectures with Real-Time Capabilities. In Proc. of ISMIS, pp. 456-465, 2014.
7. M. Faridi Masouleh ,M. A. Afshar Kazemi, M. Alborzi A., Toloie Eshlaghy Optimization of ETL Process in Data Warehouse Through a Combination of Parallelization and Shared Cache Memory. Engineering, Technology & Applied Science Research, Vol. 6, No. 6, pp.1241-1244, 2016.