DOI 10.37882/2223-2966.2023.06.31

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ РАБОТЫ ХРАНИЛИЩА ДАННЫХ GREENPLUM НА ОСНОВЕ ОБРАБОТКИ ЛОГ-ФАЙЛОВ

INTELLIGENT ANALYSIS OF THE GREENPLUM DATA WAREHOUSE BASED ON LOG FILE PROCESSING

A. Rusakov
D. Gorin
A. Lisyutenko
S. Dolzhenkov
A. Karev
I. Astafyev

Summary. The article provides methods and algorithms for data mining contained in the debugging files of the GreenPlum data warehouse to improve the efficiency of the repository as a whole. One of the important tasks to be solved in this article is the detection of cause-and-effect relationships, as a result of which the GreenPlum storage is experiencing overloads. Based on the results of the study, a software package was designed and implemented in the form of an Internet portal that allows you to download the GreenPlum storage log files and find patterns in them that indicate shortcomings in the selected storage architecture with the possibility of automatic training based on existing precedents.

Keywords: Data storage, GreenPlum optimization, data mining, log file analysis.

Русаков Алексей Михайлович
старший преподаватель,
МИРЭА — Российский технологический университет
rusakov_a@mirea.ru
Горин Денис Станиславович
доцент,
МИРЭА — Российский технологический университет
gorin@mirea.ru
Лисютенко Анастасия Сергеевна
старший преподаватель,
МИРЭА — Российский технологический университет
lisyutenko@mirea.ru
Долженков Сергей Сергеевич
ассистент,
МИРЭА — Российский технологический университет
dolzhenkov@mirea.ru
Карев Андрей Дмитриевич
МИРЭА — Российский технологический университет
karev_ad@mail.ru
Астафьев Иван Андреевич
МИРЭА — Российский технологический университет
ya.astafev-00@yandex.ru

Аннотация: В статье приводятся методы и алгоритмы интеллектуального анализа данных, содержащихся в отладочных файлах хранилища данных GreenPlum для повышения эффективности работы хранилища в целом. Одной из важных решаемых задач в данной статье, является обнаружение причинно-следственных связей, в результате которых хранилище GreenPlum испытывает перегрузки. По результатам исследования спроектирован и реализован программный комплекс в виде Интернет-портала, который позволяет загружать лог-файлы хранилища GreenPlum и находить в них закономерности, указывающие на недостатки в выбранной архитектуре хранилища с возможностью автоматического обучения по имеющимся прецедентам.

Ключевые слова: хранилище данных, оптимизация работы GreenPlum, интеллектуальный анализ данных, анализ лог-файлов.

Введение

Задачей анализа логов системы хранения базы данных является важным аспектом как в научном, так и в практическом плане для всех видов систем. Решение этой задачи позволит увеличить коэффициент производительности, а также уменьшить процент ошибок, которые возникают во время работы системы [1]. За счет анализа хронологии работы с базами данных специалисты-аналитики, а также программисты и системные администраторы могут составить прогноз будущих ошибок, сэкономить ресурсы компании, связанные со временем и эффективностью работы сервера, выявляют причины сбоя сервера, помогают устранить те ошибки, которые не были найдены естественным путем [2]. Немало важным отличием данного способа анализа данных логов систем является обнаружение возникновения критических ситуаций, т.е. поиск и предотвращение возможных ошибок на этапе проектирования задачи и анализа данных в статическом виде, для того чтобы устранить ошибки этого этапа. Как правило при работе с логами выделяют следующие типы:

- системные логи, то есть те, которые связаны с системными событиями;
- серверные логи, регистрирующие обращения к серверу и возникшие при этом ошибки;
- логи баз данных, фиксирующие запросы к базам данных;

— почтовые логи, относящиеся к входящим/исходящим письмам и отслеживающие ошибки, из-за которых письма не были доставлены;
— логи авторизации;
— логи аутентификации;
— логи приложений, установленных на этих операционных системах.

Интерпретация логов является одной из важных задач, которая может повысить работу системы в целом.

Информационная безопасность

Информационная безопасность — обеспечение конфиденциальности и целостности информации, недопущение несанкционированных действий с ней, в частности, ее использования, раскрытия, искажения, изменения, исследования и уничтожения.

Системный подход состоит из четырех составляющих обеспечения безопасности:
— законы и нормативно-правовые акты;
— распределение задач между ИБ-подразделениями;
— политика информационной безопасности;
— техническое и программное обеспечение.

Технология GreenPlum

Одной из важнейших задач для анализа логов является способ анализа логов системой GreenPlum. Существуют ситуации, при которых остается место для продажи ресурсов, в связи с этим большую часть времени система простаивает. В таких случаях делается прогноз, для поиска таблицы с высокой нагрузкой для переноса в быстродействующие хранилища Hadoop. Таким образом, часто используемые таблицы можно хранить в любой другой системе (не обязательно в GreenPlum). Исходя из этого, высвободятся ресурсы, нагрузка на этот кластер уменьшится, а значит повысится окупаемость [3].

Таким образом общая постановка задачи формулируется следующим образом: на основе предоставленных логов хранилища необходимо разработать алгоритм обработки данных и прогнозирования будущей нагрузки. Важной частью задачи является выявление и распределение объектов хранилища по различным критериям.

Greenplum — система управления данными, предназначенная для больших проектов из мира Big Data [2].

Основным отличием системы управления баз данных Grennplum является то, что она использует традиционную реляционную модель хранения данных при этом существует возможность ее масштабирования за счет использования кластерных функций в виде распределения по типу создания Raid-массивов в компьютере.

Greenplum поддерживает реляционную модель данных, сохраняя при этом неизменность данных, а это значит, что она прекрасно подойдет для данных, которые чувствительны к точности и структурности. К примеру, для финансовых операций.

Общая постановка задачи

На основе предоставленных логов хранилища необходимо разработать алгоритм обработки данных и прогнозирования будущей нагрузки. Важно частью задачи является выявление и распределение объектов хранилища по различным критериям.

В качестве исходных данных предоставлен набор логов, которые содержат запросы к базе данных GreenPlum. Нужно предложить метод быстрой обработки этих данных. Выделению отдельных объектов, к которым чаще всего обращаются пользователи. Выявления самых «тяжелых» и самых «горячих» объектов. Прогнозирование нагрузки на систему в разные временные отрезки.

Особенно важной является задача прогнозирования времени исполнения запроса по предоставленным данным [4].

Формат хранения log файла

В качестве примера приведем фрагмент данных (см. рис. 1).

Rn — уникальный номер записи
Loguser — пользователь (dev_* — разработчики, etl_* — загрузчик)
Query — запрос в базу:
— from, join — извлечение данных
— into — запись данных

Rn (bigint)	Loguser (text)	Query (text)
1513	etl_2048	from tbl_142463,join tbl_142465
1361	dev_473	join tbl_33332,join tbl_151403,INTO tbl_385661

```
(format CSV, header, delimiter ',', quote '"', null '', escape '"')
```

Рис. 1. Фрагмент хранения данных в log файле

Обоснование выбора технологий и программных средств для реализации разрабатываемых решений

Выбор технологий для интеллектуального анализа процесса работы хранилища данных на основании обработки лог-файлов GreenPlum представлен на рис. 2.

Аналитическая часть

Программное решение будет состоять из аналитической части, которая будет разрабатываться на языке программирования Python. Планируются использовать следующие основные библиотеки: matplotlib, NumPy, statsmodels.

Десктоп приложение

Для рабочего места аналитика и системного администратора предлагается использовать сразу две технологии, одна в виде десктоп приложения, которая будет работать на основании библиотеки PyQt.

Система мониторинга (Web портал)

Разработка будет проходить с использованием языка JavaScript, и фреймворков Vue.js и Quasar, а также библиотеки Axios, которая отобразит эти результаты в интерактивной форме на web-интерфейсе. Таким образом данное решение могут просматривать еще и пользователи этого хранилища.

Проект GreenPlum реализован с нуля на python. Он состоит из трёх программ. В первую программу входит выпадающий список, который содержит в себе все графики и из диаграммы matplotlib, которая рисует графики. Вторая программа, в которую входит главное окно с меню, с помощью которой подгружается файл и дальше программный алгоритм преобразует данные с помощью библиотеки pandas, в удобный вид.

Третья программа осуществляет прогнозирование.

Если вы захотели собрать первую программу из исходных файлов,то для этого требуется запустить файл main.py, с помощью среды разработки для Python, мы советуем использовать PyCharm последней версии. В первой программе используются соответствующие библиотеки, описанные выше. Открывается окно, в котором нужно выбрать соответствующий график для вычисления нужных параметров.

Если вы захотели собрать вторую программу из исходных файлов, то для этого требуется запустить файл GreenPlum2.py, с помощью среды разработки для Python, которая находится в zip файле GreenPlum2.zip. В качестве среды разработки для Python мы советуем использовать PyCharm последней версии. Во второй программе используются те же библиотеки, что и в первой программе. Открывается окно, в котором нужно в панели меню открыть файл tables.xlsx.

Рис. 2. Технологии для разработки программного решения

Если вы захотели собрать третью программу из исходных файлов,то для этого требуется запустить файл GreenPlum3.py, с помощью среды разработки для Python, которая находится в zip файле GreenPlum3.zip. В качестве среды разработки для Python мы советуем использовать PyCharm последней версии. Все исходные коды программных решений доступны в [5]

Первая программа

Пример работы первой части программного средства можно посмотреть на следующих рисунках.

На рисунках 4 и 6 представлены таблицы, на которых показаны столбчатые диаграммы высоконагруженных таблиц, где в каждом столбце показано число обращений к данной таблице. В исследуемых данных было выявлено четыре таблицы, при рассмотрении которых, наиболее часто обращались (число обращений в среднем составляет около 27000, в то время как к другим таблицам число обращений не превышало 10000).

Далее мы рассмотрели структуру лог-файла и сделали два графика представленных на рисунках 3 и 5. На этих графиках видно, что зависимость распределения длины запросов от количества запросов подчиняется экспоненциальному закону, это означает, что данное распределение не является нормально-распределенным (Гауссовским). Таким образом можно сделать допущение к данным исследуемым логам можно применить регрессионный анализ [6].

Вторая программа

Согласно второму заданию происходит поиск неиспользуемых таблиц, в которые все записывается и никогда не читается разработчиком.

Инструкция применения: после открытия файла tables.xlsx есть небольшая задержка, из-за подгрузки данных — просто подождите.

Поиск таблиц

В данном случае мы отсортировали данные при помощи библиотеки Pandas таким образом, чтобы получились следующие столбцы, изображенные на рисунке 7. Etl_into, etl_join, dev_into, dev_join, dev_from, range (etl_into + dev_into).

Третья программа

Прогнозирование нагрузки на кластер. Для прогнозирования нагрузки на кластер после использования алгоритмов машинного обучения, в частности, Байесовский классификатор было принято решение использование обычной нелинейной регрессионной модели четвертого порядка на основе ARIMA [7].

Оценивание параметров модели и вычисление остатков:

$$y_t - \varphi_1 \cdot y_{t-1} - ... - \varphi_p \cdot y_{t-p} = \\ = \delta + \varepsilon_t - \theta_1 \cdot \varepsilon_{t-1} - ... - \theta_p \cdot \varepsilon_{t-p};$$

$$\varepsilon_t \approx iid(0,\sigma^2)$$

Тогда процесс y_t является ARIMA (p,q,d).

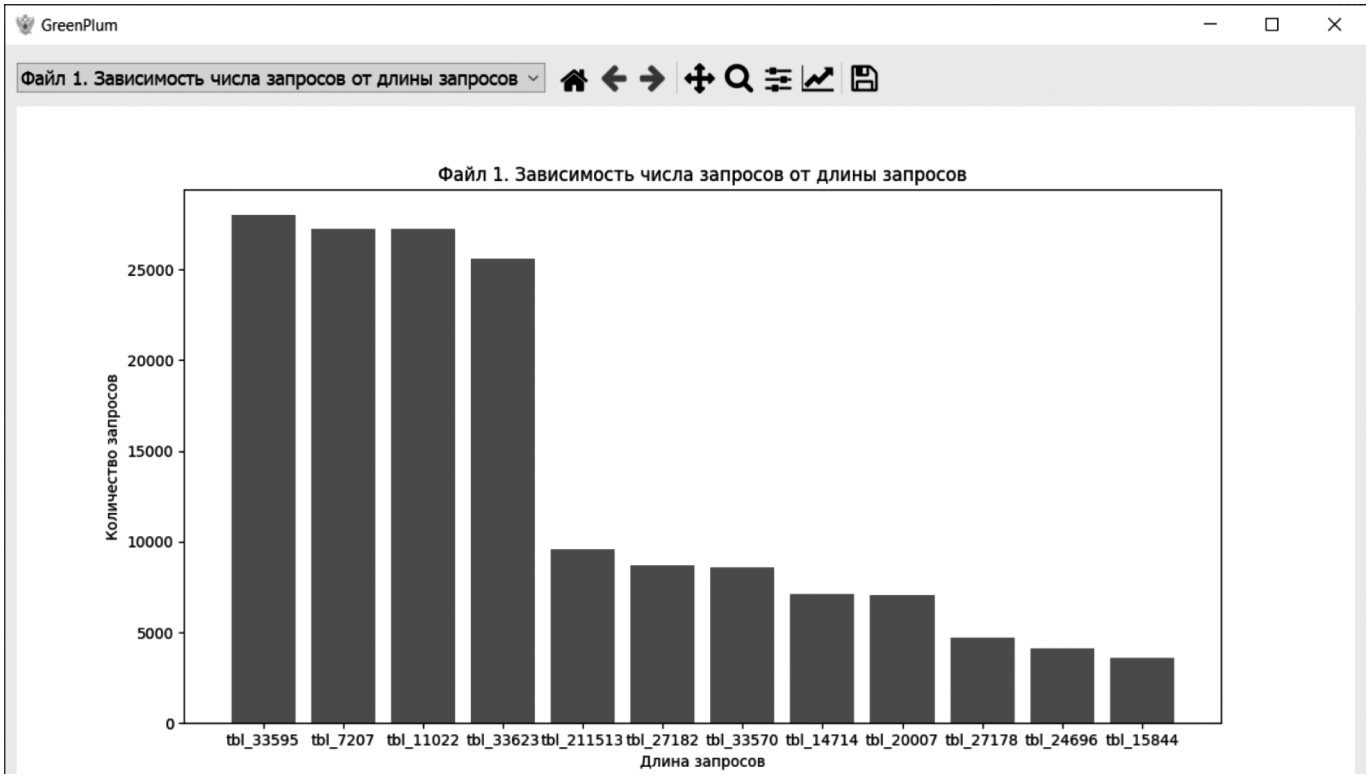

Рис. 3. Зависимость числа запросов от длины запросов

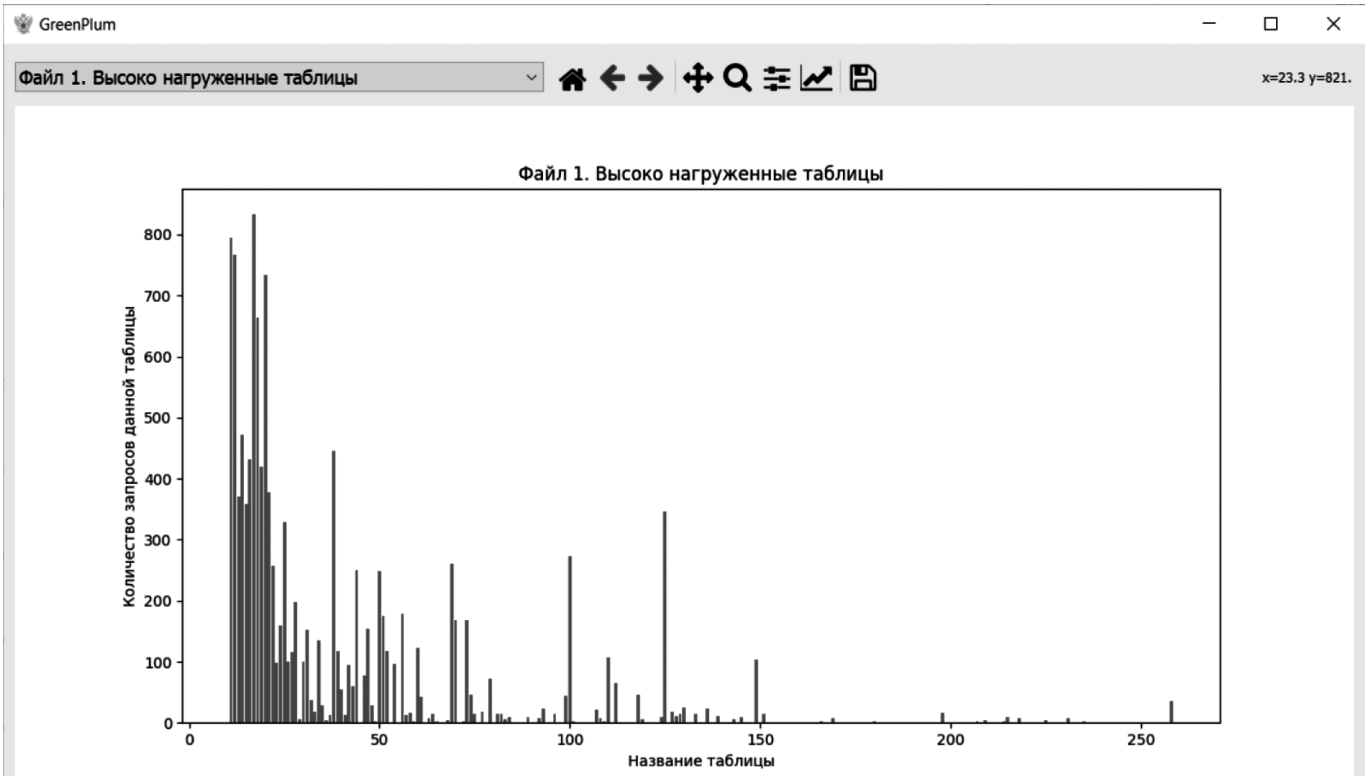

Рис. 4. Высоконагруженные таблицы

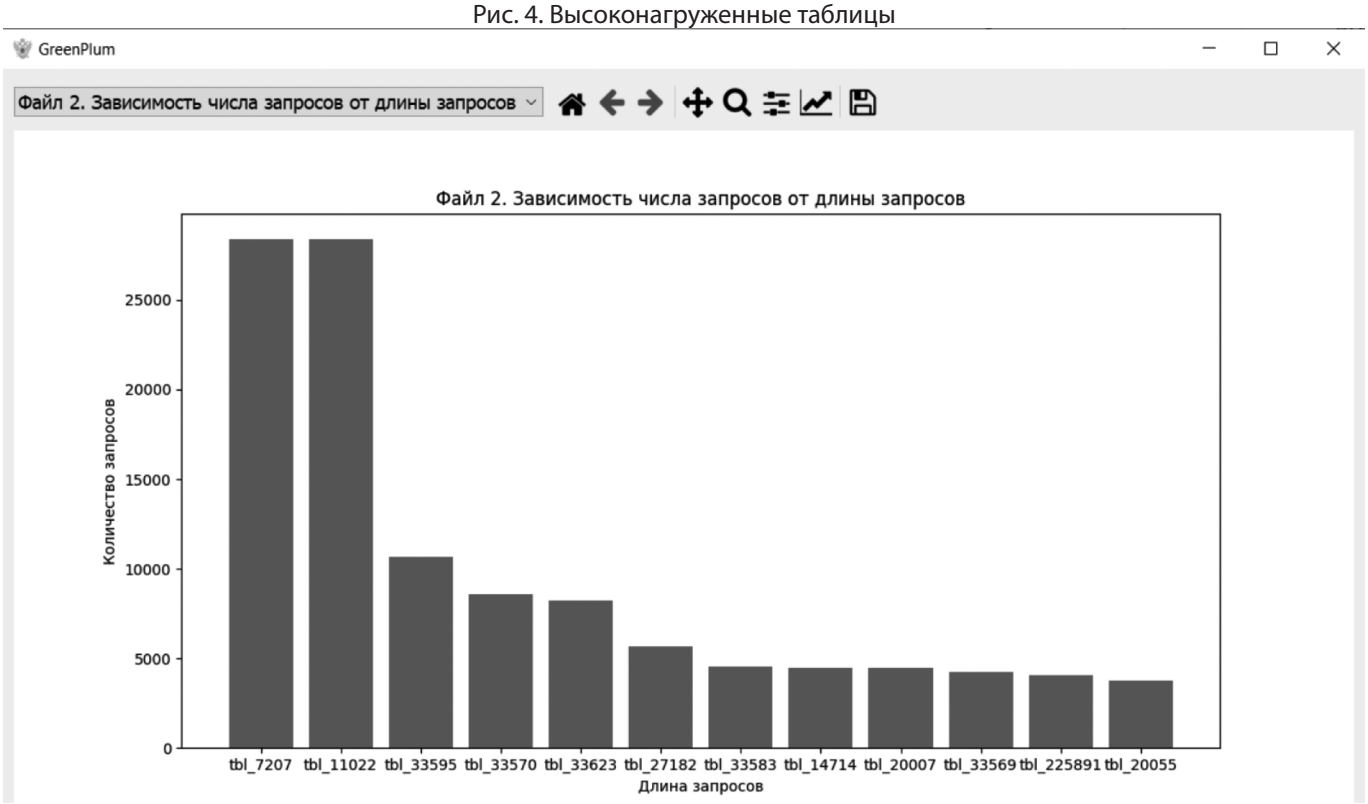

Рис. 5. Зависимость числа запросов от длины запросов

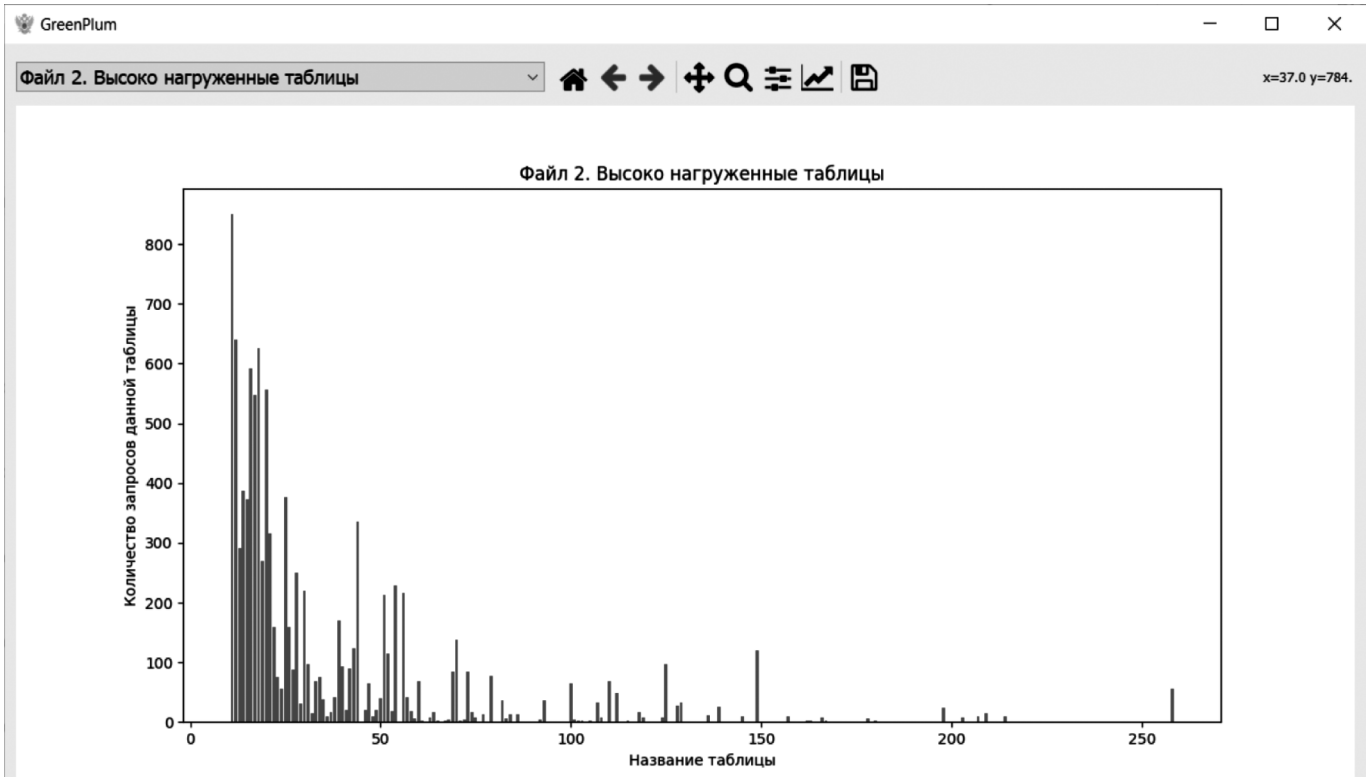

Рис. 6. Высоконагруженные таблицы

GreenPlum2

Меню

	table_name	etl_into	etl_join	etl_from	dev_into	dev_join	dev_from	range
1	tbl_27182	475	4718	0	0	0	0	475
2	tbl_29495	468	64	0	0	0	0	468
3	tbl_29497	468	0	0	0	0	0	468
4	tbl_28425	379	0	0	0	0	0	379
5	tbl_27356	379	0	0	0	0	0	379
6	tbl_26783	378	0	0	0	0	0	378
7	tbl_29461	377	0	0	0	0	0	377
8	tbl_29458	373	0	0	0	0	0	373
9	tbl_26784	366	0	0	0	0	0	366
10	tbl_26792	365	0	0	0	0	0	365
11	tbl_28670	365	0	365	0	0	0	365
12	tbl_29481	351	157	0	0	0	0	351
13	tbl_27619	324	0	0	0	0	0	324
14	tbl_26786	295	0	0	0	0	0	295
15	tbl_29415	288	0	0	0	0	0	288
16	tbl_361734	288	0	181	0	0	0	288
17	tbl_27483	279	0	0	0	0	0	279
18	tbl_27486	275	0	0	0	0	0	275
19	tbl_27761	274	0	0	0	0	0	274
20	tbl_26778	268	0	0	0	0	0	268
21	tbl_28675	266	0	0	0	0	0	266
22	tbl_28674	266	0	0	0	0	0	266
23	tbl_29485	263	0	0	0	0	0	263
24	tbl_28436	262	0	0	0	0	0	262
25	tbl_29457	261	105	0	0	0	0	261
26	tbl_29486	261	0	0	0	0	0	261
27	tbl_29962	260	0	0	0	0	0	260

Рис. 7. Вариант работы второй программы

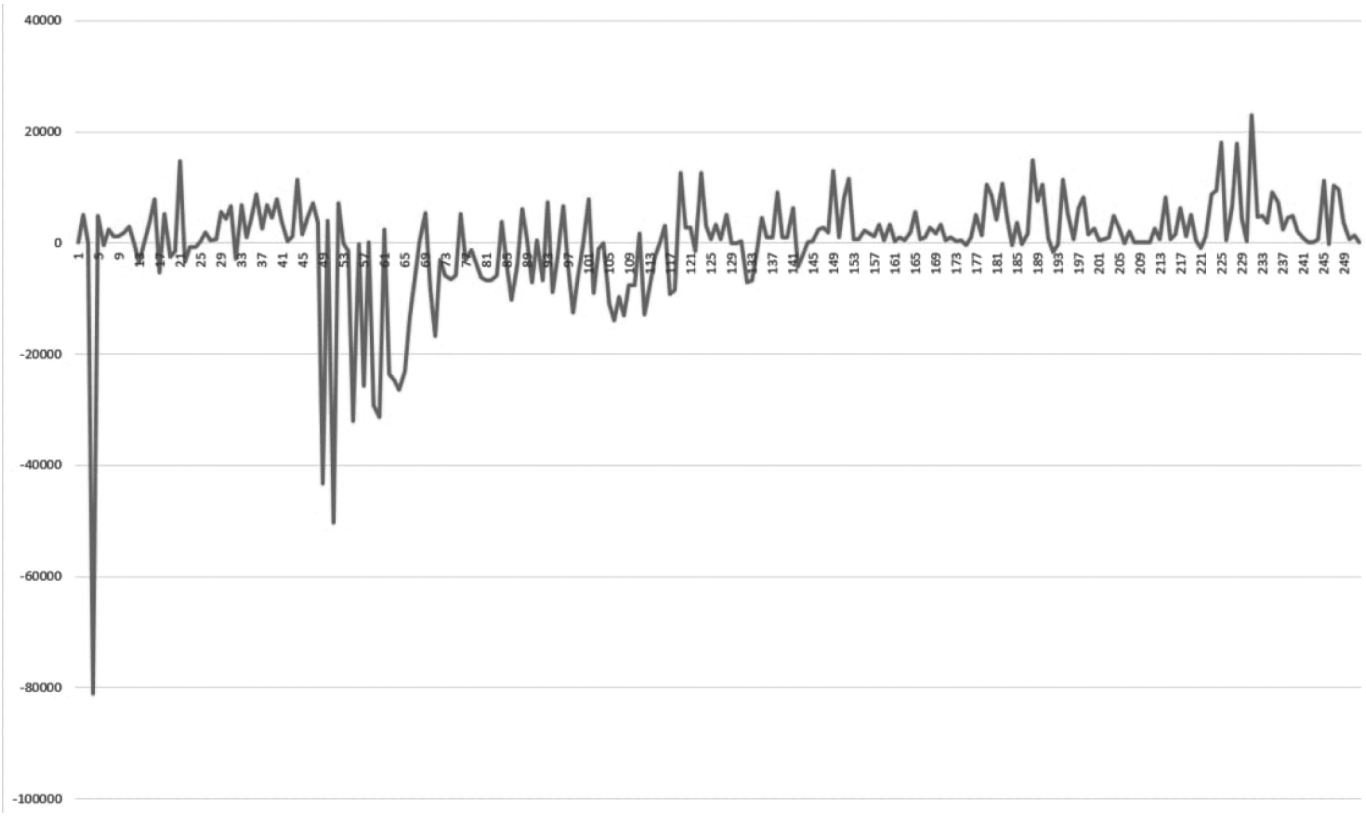

Рис. 9. Прогресс доступа к данным

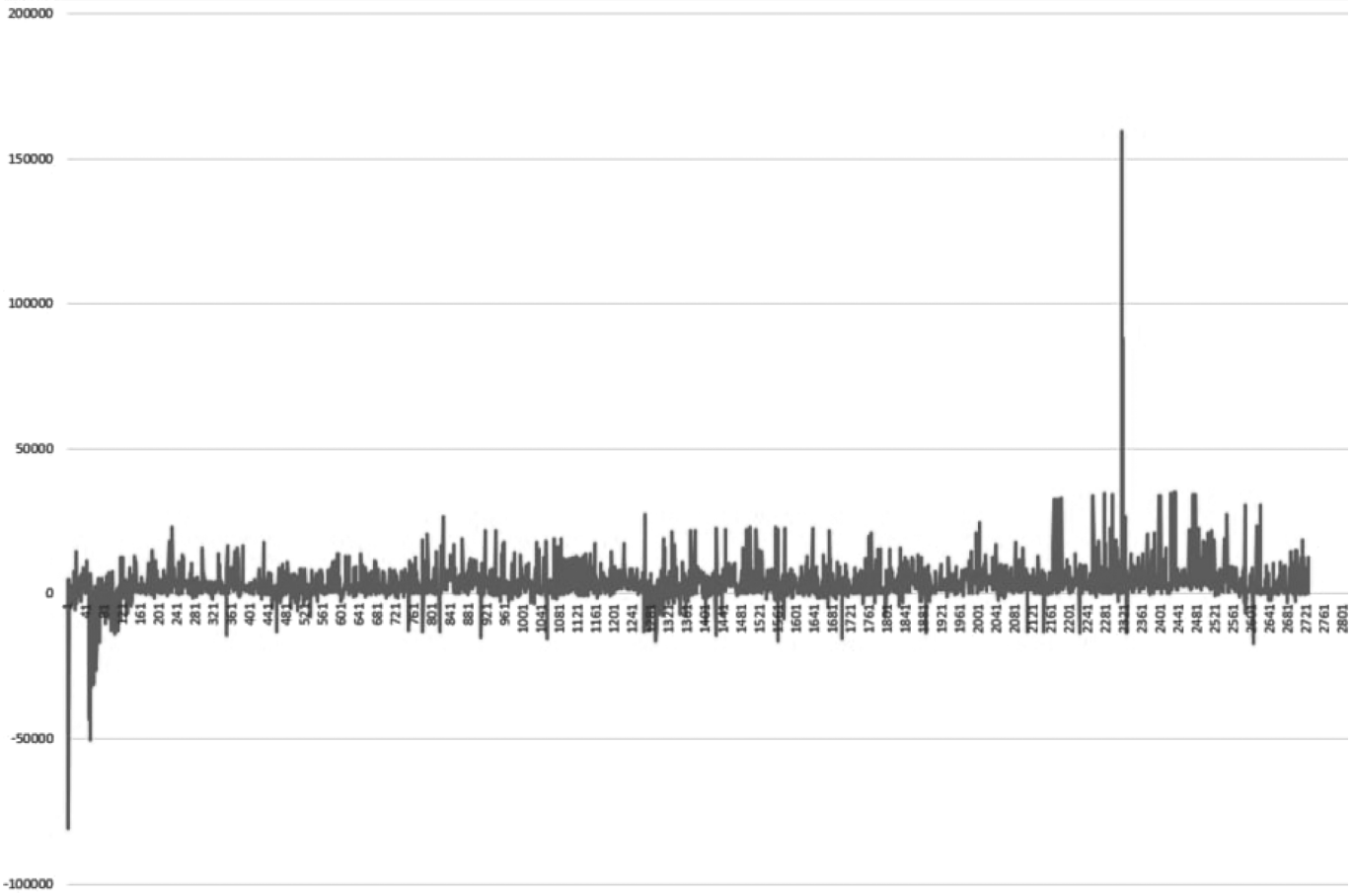

Рис. 10. Прогресс доступа к данным

Каждая из полученных моделей проверяется на соответствие исходным данным. Из тех моделей, которые адекватны данным, отыскивается наиболее простая модель, то есть модель, которая имеет наименьшее количество параметров [8].

2. Оценивание модели и проверка ее адекватности.

Прогнозирование временного ряда. После поиска ряда моделей, следует выполнить прогнозирование на несколько шагов по времени с оцениванием крайних границ прогнозируемых значений.

Благодаря этой модели была построена следующее решение:

rn	loguser	tbl_nan	duratio	predict_duration	difference
4467	dev_332	from tbl_7	63,108	200	136,892
4512	etl_2048	from tbl_8	31,022	5126,372929	5095,350929
4514	dev_332	from tbl_7	174,179	200	25,821
4522	dev_332	from tbl_2	100987,4	20001,26099	-80986,17801
4530	etl_2048	from tbl_8	146,984	5126,372929	4979,388929
4532	etl_2048	from tbl_8	1491,244	1144,93639	-346,3076098
4543	etl_2048	from tbl_1	339,079	2693,73319	2354,65419
4548	etl_2048	from tbl_8	16,879	1144,93639	1128,05739
4550	etl_2048	from tbl_8	12,398	1144,93639	1132,53839
4552	etl_2048	into tbl_85	3450,567	5326,372929	1875,805929
4554	etl_2048	from tbl_2	808,493	3716,1208	2907,6278
4557	dev_332	from tbl_7	173,476	200	26,524
4561	etl_2048	into tbl_85	8878,601	5326,372929	-3552,228071
4590	etl_2048	from tbl_8	1788,173	1617,775571	-170,3974286
4592	etl_2048	from tbl_8	2279,249	6071,455429	3792,206429
4596	etl_2048	from tbl_8	1927,034	9765,17525	7838,14125
4607	etl_2048	into tbl_85	7112,98	1817,775571	-5295,204429
4609	etl_2048	from tbl_8	2599,589	7913,047	5313,458
4622	etl_2048	into tbl_85	8730,629	6271,455429	-2459,173571
4624	etl_2048	from tbl_8	1830,462	484,1488571	-1346,313143
4633	dev_359	from tbl_3	2897,266	17596,7845	14699,5185

Рис. 8. Фрагмент выборки базы данных для поиска нагруженных таблиц

Была построена следующая таблица, в которой помимо стандартного лога прописывается два дополнительных поля predict_duration и difference.

Difference – это разница между значением, которое спрогнозировано и тем, что по факту является. Если система показывает положительный результат, то это говорит о нехватке ресурсов кластера, а если отрицательный, то о нормальной работе системы. Для данного значения difference были построены графики на рис. 9 и 10. На рисунке 9 видно, что прогресс доступа к данным показывает о том, что в это время был момент простоя, и кластер был не нагружен, поскольку присутствует много отрицательных значений. Рисунок 10 показывает график, в котором кластер был перегружен и начал давать отказы.

Заключение

Таким образом используя авторегрессионную модель ARIMA и соответствующие алгоритмы обработки данных из лог-файлов системы GreenPlum мы сможем прогнозировать остаточный ресурс данного кластерного решения и находить нагруженные таблицы тем самым повышая эффективность использование данного программного решения.

ЛИТЕРАТУРА

1. (Мохигул А., Мохинур А. ПОНЯТИЕ BIG DATA И ЕГО ОСНОВНЫЕ ХАРАКТЕРИСТИКИ //RESEARCH AND EDUCATION. — 2022. — С. 596.
2. Han J., Pei J., Tong H. Data mining: concepts and techniques. — Morgan kaufmann, 2022.
3. Павлович Н.В. Оптимизация запросов в Greenplum. — 2022.
4. E.E. Yusufoglu, M. Ayyildiz and E. Gul, «Neural network-based approaches for predicting query response times», 2014 International Conference on Data Science and Advanced Analytics (DSAA), Shanghai, China, 2014, pp. 491–497, doi: 10.1109/DSAA.2014.7058117.
5. Русаков А.М. Исходный код проекта интеллектуальный анализ работы хранилища данных greenplum на основе обработки лог-файлов. https://github.com/RusAl84/greenplum_log_analysis. (2023).
6. Fyfe S. et al. Future flavours from the past: Sensory and nutritional profiles of green plum (Buchanania obovata), red bush apple (Syzygium suborbiculare) and wild peach (Terminalia carpentariae) from East Arnhem Land, Australia // Future Foods. — 2022. — Т. 5. — С. 100136.
7. Gilbert K. An ARIMA supply chain model // Management Science. — 2005. — Т. 51. — № 2. — С. 305–310.
8. Shumway R.H. et al. ARIMA models // Time Series Analysis and Its Applications: With R Examples. — 2017. — С. 75–163.

© Русаков Алексей Михайлович (rusakov_a@mirea.ru); Горин Денис Станиславович (gorin@mirea.ru);
Лисютенко Анастасия Сергеевна (lisyutenko@mirea.ru); Долженков Сергей Сергеевич (dolzhenkov@mirea.ru);
Карев Андрей Дмитриевич (karev_ad@mail.ru); . Астафьев Иван Андреевич (ya.astafev-00@yandex.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»