DOI: 10.17725/rensit.2023.15.179

Проблемы масштабирования компонентов в системах потоковой обработки данных

Булычев Г.Г., Черных А.В.
МИРЭА-Российский технологический университет, https://www.mirea.ru/
Москва 119454, Российская Федерация
E-mail: geo-bulychev@mail.ru, meidm@yandex.ru
Поступила 24.04.2023, рецензирована 30.04.2023, принята 08.05.2023
Представлена действительным членом РАЕН А.С. Дмитриевым

Аннотация: **Проводится рассмотрение существующих проблем масштабирования компонентов в системах потоковой обработки данных. Предлагаемый алгоритм значительно снижает количество операций создания и удаления компонентов при масштабировании системы. Алгоритм основан на линейной регрессии. Для предложенного алгоритма моделируется практическая нагрузка на систему для подтверждения полученных теоретических результатов.**

Ключевые слова: **потоковая обработка данных, масштабирование, большие данные**

УДК 004.62

Для цитирования: Булычев Г.Г., Черных А.В. Проблемы масштабирования компонентов в системах потоковой обработки данных. *РЭНСИТ: Радиоэлектроника. Наносистемы. Информационные технологии*, 2023, 15(2):179-184. DOI: 10.17725/rensit.2023.15.179.

The problems of scaling components in streaming data processing systems

George G. Bulychev, Alexey V. Chernykh
MIREA-Russian Technological University, https://www.mirea.ru/
Moscow 119454, Russian Federation
E-mail: geo-bulychev@mail.ru, meidm@yandex.ru
Received April 24, 2023, peer-reviewed April 30, 2023, accepted May 08, 2023

Abstract: **The article examines the existing problems of scaling components in streaming data processing systems. The algorithm proposed by the authors significantly reduces the number of operations for creating and removing components when scaling the system. The algorithm is based on linear regression. For the proposed algorithm, the practical load on the system is simulated to confirm the theoretical results obtained.**

Keywords: **streaming data processing, scaling, big data**

UDC 004.62

For citation: George G. Bulychev, Alexey V. Chernykh. The problems of scaling components in streaming data processing systems. *RENSIT: Radioelectronics. Nanosystems. Information Technologies*, 2023, 15(2):179-184e. DOI: 10.17725/rensit.2023.15.179.

Содержание

1. ВВЕДЕНИЕ

В условиях современного информационного общества, характеризующегося гигантскими объемами данных [1], постоянно поступающими в режиме реального времени, актуальность и значимость потоковой обработки данных [2] неуклонно растет. Этот подход к обработке информации предполагает немедленный анализ и обработку данных, в отличие от классического подхода, с сохранением данных в хранилище и постобработкой.

Основным преимуществом систем потоковой обработки данных является возможность оперативного принятия решений на основе свежей и актуальной информации, что позволяет организациям и специалистам приспосабливаться к динамично меняющимся условиям и сохранять конкурентоспособность на рынке. Однако, вместе с ростом масштабов и сложности потоков обработки данных возникает ряд технических и концептуальных проблем. В частности, алгоритмы масштабирования компонентов в системах потоковой обработки данных сталкиваются с требованиями, связанными с эффективностью, надежностью и гибкостью.

Необходимость быстро и эффективно распределять ресурсы между узлами обработки данных создает проблемы в определении оптимального количества ресурсов для каждого узла, чтобы обеспечить высокую производительность и надежность системы. Неправильное масштабирование может привести к снижению производительности, дополнительным затратам на ресурсы и затруднению обработки данных в режиме реального времени.

2. АНАЛИЗ СУЩЕСТВУЮЩИХ АЛГОРИТМОВ

В существующих системах потоковой обработки данных используется алгоритм масштабирования с пороговым значением [3] благодаря высокой скорости принятия решения.

2.1. Алгоритм масштабирования с пороговым значением

Алгоритм масштабирования с пороговым значением основан на определении пороговых значений нагрузки на узлы системы. Если нагрузка на узел превышает определенный порог, система автоматически масштабируется для распределения нагрузки.

Результатом работы алгоритма выступает количество сущностей компонента – величина, характеризующая необходимое число копий компонента для обработки поступающей нагрузки.

Количество сущностей компонента рассчитывается по формуле:

$$C_t = N_{t-1}/M, \quad (1)$$

где C_t – количество сущностей компонентов в момент времени t, N_{t-1} – нагрузка на систему в момент времени $t - 1$, M – пороговое значение нагрузки.

Как видно из формулы, алгоритм не прогнозирует нагрузку и оперирует только известными величинами. Данная особенность приводит к "отставанию" алгоритма от текущей нагрузки. При скачкообразном графике нагрузки система будет подстраиваться к нагрузке с отставанием, производя излишнее количество операций создания и удаления сущностей компонентов.

3. АЛГОРИТМЫ С ПРОГНОЗИРОВАНИЕМ НАГРУЗКИ

Для эффективного масштабирования компонентов необходимо снизить количество операций создания и удаления сущностей компонентов. Для выполнения данного требования необходимо реализовать алгоритм прогнозирования нагрузки. Проблему прогнозирования нагрузки можно свести к проблеме прогнозирования временных рядов

(так как нагрузка напрямую связана с временным рядом).

Однако в контексте существующей задачи алгоритм должен иметь минимальное время задержки принятия решения и использовать минимальное число ресурсов. В противном случае алгоритм будет работать с сильным запозданием или со значительным повышением потребления ресурсов, тем самым нивелируя экономию, вызванную снижением числа операций создания или удаления сущностей компонентов.

Данные особенности не позволяют использовать алгоритмы, основанные на нейронных сетях [4], случайном лесе [5] и большинство других алгоритмов машинного обучения [6,7]. Однако алгоритм линейной регрессии [8] и экспоненциального сглаживания [9] подпадают под данные требования.

3.1. Алгоритм линейной регрессии

Линейная регрессия — это статистический метод машинного обучения, используемый для моделирования связи между зависимой переменной и одной или несколькими независимыми переменными. В контексте прогнозирования нагрузки зависимая переменная может быть нагрузкой на систему, а независимые переменные – факторы, влияющие на нагрузку (например, время суток, день недели, объем трафика).

Математически линейная регрессия описывается как:

$$C_{t+1} = \frac{\overline{Y}_{t+1}}{M}, \qquad \overline{Y}_{t+1} = \delta \cdot N_t + \varepsilon, \tag{2}$$

где C_{t+1} – количество сущностей компонентов в момент времени $t + 1$, $\overline{Y}_{t+1}$ – прогнозируемая нагрузка в момент времени $t + 1$, M – пороговое значение нагрузки, N_t – нагрузка на систему в момент времени t, δ, ε – коэффициенты регрессии.

Подбор значения коэффициентов линейной регрессии производится методом наименьших квадратов.

3.2. Алгоритм экспоненциального сглаживания

Экспоненциальное сглаживание — это метод прогнозирования временных рядов, который учитывает все наблюдения в прошлом, присваивая им экспоненциально убывающие веса. Таким образом, более новые наблюдения имеют большее влияние на прогноз, чем старые.

Для решения существующей задачи оптимальным будет алгоритм тройного экспоненциального сглаживания (метод Хольта-Винтерса [10]), так как данный алгоритм учитывает тренды и сезонность, а для большинства обрабатываемых сообщений характерны данные особенности.

Математически алгоритм описывается как:

$$\begin{aligned} C_{t+h} &= \frac{\overline{Y}_{t+h}}{M}, \\ \overline{Y}_{t+h} &= A(t) + h \cdot B(t) + S(t - p + 1 + (h-1) \bmod p), \\ A(t) &= \alpha \cdot (N_t - S(t-p)) + (1-\alpha) \cdot (A(t-1) + B(t-1)t), \\ B(t) &= \beta \cdot (A(t) - A(t-1)) + (1-\beta) \cdot B(t-1), \\ S(t) &= \gamma \cdot (N_t - A(t)) + (1-\gamma) \cdot S(t-p), \end{aligned} \tag{3}$$

где C_{t+h} – количество сущностей компонентов в момент времени t + h, $\overline{Y}_{t+h}$ – прогнозируемая нагрузка в момент времени $t + 1$, M – пороговое значение нагрузки, N_t – нагрузка на систему в момент времени t, A – уравнение, описывающее сглаженный ряд, B – уравнение для оценки тренда, S – уравнение для оценки сезонности, α – постоянная величина, определяющая влияние сглаженного ряда, β – постоянная величина, определяющая влияние тренда, γ – постоянные величины, определяющие влияния сезонности, p – период сезонности.

Для функционирования алгоритма необходимо определить постоянные величины. Однако для большинства компонентов невозможно их оценить заранее.

4. МОДЕЛИРОВАНИЕ

4.1. Ключевые показатели

Использование алгоритма должно приводить к:

- снижению количества операций создания и удаления компонентов;
- снижению задержки в обработке сообщений с момента поступления в очередь. Для систем

потоковой обработки данных критически важным параметром является время обработки каждого сообщения;

- снижению максимального и среднего размера очереди. Моделирование системы проводится на примере одного компонента, однако в целевой системе количество компонентов и очередей может исчисляться тысячами и при значительном росте размера очереди каждого компонента суммарное потребление памяти системой может нивелировать экономию ресурсов, полученную за счет снижения количества операций масштабирования.

Целевой алгоритм должен показать минимальное значение указанных параметров при моделировании.

Алгоритм должен использовать минимальные ресурсы для прогнозирования нагрузки.

Для оценки всех вышеперечисленных параметров выбраны следующие метрики:

- Количество операций создания и удаления сущностей компонентов.
- Среднее время задержки обработки сообщения.
- Максимальный размер очереди.
- Средний размер очереди.
- Изменение объема потребляемой памяти.

4.2. Контекст моделирования

Рассмотрим систему, состоящую из одного компонента – обработчик сообщений в публичных репозиториях кода ($A1$) и очереди ($Q1$), в которую поступают сообщения для обработки (**Рис. 1**). Обработка данных сообщений позволяет выявить утечки исходного кода или конфиденциальной информации компаний.

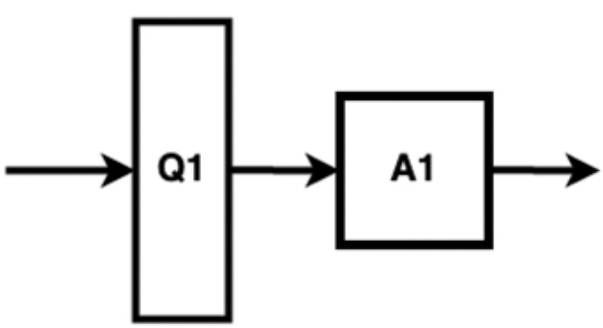

Рис. 1. *Система обработки с очередью.*

Таблица 1

Параметры моделирования

Параметр	Значение
Производительность A1	50 сообщений в секунду
Суммарное количество сообщений	30 715 323
Время запуска или удаления сущностей	1 секунда

В процессе моделирования каждый алгоритм получает информацию о текущей нагрузке и возвращает количество сущностей компонентов, которые необходимо запустить или удалить в следующий момент времени.

В качестве источника нагрузки выбраны события на github.com за период с 2022-01-01 15:00:00 до 2022-01-12 02:51:46.

Дополнительные параметры моделирования отражены в **Таблице 1**.

4.3. Моделирование алгоритма с пороговым значением

При моделировании алгоритма с пороговым значением потребление памяти было фиксировано.

В результате моделирования (**Таблица 2**) была достигнута минимальная задержка в обработке сообщений при, однако, колоссальном числе операций создания и удаления компонентов.

4.4. Моделирование алгоритма экспоненциального сглаживания

Для моделирования алгоритма необходимо определить три константы:

- влияние сглаженного ряда;
- влияние тренда;
- влияние сезонности.

Таблица 2

Результаты моделирования алгоритма с пороговым значением

Параметр значение	Количество операций
Количество операций масштабирования	81 746 850
Средний размер очереди	3.534165161
Среднее время задержки сообщений	0.156226077
Максимальный размер очереди	140

Таблица 3

Результаты моделирования алгоритма экспоненциального сглаживания

Параметр значение	Количество операций
Количество операций масштабирования	148 122 776
Средний размер очереди	10.47011898
Среднее время задержки сообщений	0.46282659
Максимальный размер очереди	415

Однако до формирования нагрузки и анализа корректно определить данные параметры невозможно. Для моделирования были выбраны случайным образом следующие значения:

- влияние сглаженного ряда – 0.4;
- влияние тренда – 0.1;
- влияние сезонности – 0.02.

В результате моделирования алгоритма потребление памяти было фиксировано.

Алгоритм показал результаты (**Таблица 3**) значительно хуже, чем алгоритм с пороговым значением. Количество операций выше в 1.8 раз, а средняя задержка больше чем в два раза по сравнению с алгоритмом с пороговым значением.

4.5. Моделирование алгоритма линейной регрессии

При моделировании алгоритма линейной регрессии потребление памяти было фиксировано.

Алгоритм показал (**Таблица 4**) снижение числа операций создания и удаления компонентов более чем в 740 раз. Однако время задержки сообщений и размер очереди возросли. Полученный результат можно объяснить тем, что алгоритм проводит эффективное сглаживание всплесков и спадов нагрузки к среднему значению и корректно определяет тенденции.

Таблица 4

Результаты моделирования для линейной регрессии

Параметр значение	Количество операций
Количество операций масштабирования	109 596
Средний размер очереди	7.624188911
Среднее время задержки сообщений	0.337023616
Максимальный размер очереди	539

4.6. Выводы по итогам моделирования

Алгоритм с экспоненциальным сглаживанием неприменим для данной задачи, так как для корректной работы необходимо строго определить начальные константы.

Алгоритм линейной регрессии показал значительно снижение числа операций создания и удаления компонентов, по сравнению с классическим алгоритмом с пороговым значением: 109 596 против 81 746 850 операций. Однако увеличивается время задержки сообщений с 0.156 до 0.337 секунды.

Использование алгоритма линейной регрессии позволяет достичь значительного снижения потребляемых ресурсов с минимальным увеличением времени задержки.

5. ЗАКЛЮЧЕНИЕ

Использование алгоритма линейной регрессии для масштабирования компонентов позволяет значительно снизить число операций создания и удаления компонентов (более чем в 700 раз), однако увеличивается задержка в обработке сообщений. Данную особенность можно нивелировать добавлением временного окна, на основе которого будет производиться прогнозирование нагрузки.

При использовании предложенного алгоритма для прогнозирования нагрузки и алгоритма маркировки [11] для маршрутизации сообщений в системе потоковой обработки данных возможно достичь снижения количества операций масштабирования компонентов более чем в 800 раз.

Предложенные алгоритмы могут быть встроены в системы, построенные на микросервисной [12] архитектуре. Благодаря разработанным алгоритмам потребление ресурсов значительно снижается как при обработке больших объемов данных, так и при решении прикладных задач.

ЛИТЕРАТУРА

1. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 //

https://www.statista.com/statistics/871513/worldwide-data-created/ (дата обращения: 24.04.2023)

2. Loshin D. *ETL (Extract, Transform, Load): Business Intelligence*. USA, Silver Spring, Morgan Kaufmann Publ., 2012, 400 p.
3. Chandrima R, Kashyap B, Sandeep A, Manjusha P. Horizontal Scaling Enhancement for Optimized Big Data Processing: *Proceedings of IEMIS*. *Emerging Technologies in Data Mining and Information Security,* 2019, 639-649 p.p. DOI: 10.1007/978-981-13-1951-8_58.
4. Oancea B, Ciucu S. Time series forecasting using neural network. //https://arxiv.org/pdf/1401.1333.pdf (дата обращения: 24.04.2023)
5. Random Forests for Time Series //https://hal.science/hal-03129751/file/Block_bootstrap_for_random_forests.pdf (дата обращения: 24.04.2023)
6. Machine Learning Algorithms for Time Series Analysis and Forecasting //https://arxiv.org/abs/2211.14387 (дата обращения: 24.04.2023)
7. Gianluca B, Souhaib B, Yann-Aël B. Machine Learning Strategies for Time Series Forecasting. *Lecture Notes in Business Information Processing 138*, 2013, 62-77 p.p. DOI: 10.1007/978-3-642-36318-4_3.
8. Dastan H, Adnan M. A Review on Linear Regression Comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends,* 2020, 140-147 p.p. DOI: 10.38094/jastt1457.
9. Handanhal V. Forecasting With Exponential Smoothing – What's The Right Smoothing Constant? *Review of Business Information Systems,* 2013, 117-126 p.p. DOI:10.19030/rbis.v17i3.8001.
10. Chatfield C. The Holt-Winters Forecasting Procedure. *Journal of the Royal Statistical Society,* 1978, 264-279 p.p. DOI: 10.2307/2347162.
11. George G. Bulychev, Alexey V. Chernykh. Problems of Message Routing Algorithms in Streaming Data Processing Systems. *RENSIT: Radioelectronics. Nanosystems. Information Technologies,* 2022, 14(3):279-290e. DOI: 10.17725/rensit.2022.14.279.
12. Al-Debagy O, Martinek P. A Comparative Review of Microservices and Monolithic Architectures. *18th IEEE International Symposium on Computational Intelligence and Informatics*, 2018. DOI: 10.1109/CINTI.2018.8928192.

Булычев Георгий Гавриилович
доктор физ.-мат. наук, профессор
МИРЭА-Российский технологический университет
78, просп. Вернадского, Москва 119454, Россия
E-mail: geo-bulychev@mail.ru

Черных Алексей Валерьевич
аспирант
МИРЭА-Российский технологический университет
78, просп. Вернадского, Москва 119454, Россия
E-mail: meidm@yandex.ru.