

Исследование методов и разработка решения для повышения отказоустойчивости высоконагруженных IT-систем с использование Fallback cache**Семисалов Д.В. (ИТМО)****Научный руководитель – Зубаков А.Н. (ИТМО)**

Введение. Процесс сжатия данных в высоконагруженных IT-системах представляет собой сложную задачу, требующую баланса между эффективностью компрессии, нагрузкой на вычислительные ресурсы и скоростью работы системы [1]. В распределённых базах данных, таких как Apache Cassandra, механизмы сжатия играют ключевую роль в оптимизации дискового пространства и снижении затрат на хранение данных [2]. Однако выбор оптимального алгоритма и его параметров существенно влияет на производительность системы в целом. В частности, комбинация сжатия на уровне приложения и на стороне базы данных может существенно повысить эффективность использования ресурсов [3]. Исследование посвящено анализу взаимодействия алгоритмов компрессии на различных уровнях обработки данных, а также разработке адаптивного метода настройки параметров сжатия, позволяющего изменять параметры алгоритмов Snappy и Deflate в зависимости от характеристик входных данных. Такая методика обеспечивает оптимизацию сжатия без избыточного увеличения нагрузки на процессор и дисковую подсистему, что особенно важно в условиях интенсивного потока запросов к базе данных [4].

Основная часть. В ходе работы были решены несколько ключевых задач, связанных с исследованием эффективности алгоритмов сжатия на разных уровнях обработки данных. Были проведены тестирования, включавшие три различных сценария компрессии: Использование только встроенного в Cassandra алгоритма Deflate, обеспечившего снижение объёма данных в среднем на 60%, но при этом увеличившего нагрузку на процессор при записи [5]. Применение алгоритмов компрессии на стороне приложения (LZ4, Snappy, Zstd) перед записью данных в базу, что позволило уменьшить объём передаваемых данных на 30–40%, снизив нагрузку на Cassandra. Комбинированное сжатие, включавшее предварительную компрессию на уровне приложения и последующее сжатие Deflate в Cassandra. Этот метод дал наибольший выигрыш по дисковому пространству — более 80%. Особое внимание было уделено влиянию различных алгоритмов на производительность системы. Было выявлено, что при использовании Snappy на стороне приложения в комбинации с Deflate в Cassandra наблюдалось наибольшее сокращение объёма данных (85%) при сохранении высокой скорости обработки. Это объясняется особенностями Snappy, который минимизирует накладные расходы на процессор и сохраняет структуру данных, что позволяет Deflate достигать максимального уровня компрессии. В то же время LZ4 и Zstd, хотя и показали достойные результаты, оказались менее эффективными из-за либо ограниченной степени сжатия (LZ4), либо высокой нагрузки на CPU при обработке данных (Zstd). Дополнительно в рамках исследования был разработан метод адаптивной настройки параметров Snappy и Deflate в зависимости от характеристик входных данных. Алгоритм анализа данных позволил изменять уровень компрессии в реальном времени, адаптируя систему к текущей нагрузке и типу информации. Это позволило сократить избыточные вычислительные операции и повысить общую производительность системы.

Выводы. Проведённый анализ подтвердил, что комбинированное использование алгоритмов Snappy и Deflate обеспечивает максимальную степень сжатия данных при минимальных накладных расходах на производительность. Разработанная методика адаптивной настройки параметров сжатия позволила динамически изменять алгоритмы компрессии в зависимости от характеристик данных, что улучшило баланс между степенью сжатия, нагрузкой на процессор и скоростью работы системы. В дальнейших исследованиях планируется расширить тестирование на другие типы данных, а также провести нагрузочное

тестирование с использованием Gatling для оценки масштабируемости разработанного метода в условиях реального трафика.

Список использованных источников:

1. Гарев К. В. Сравнительный анализ методов сжатия табличных данных // Программные продукты и системы. – 2024. – №2. – С. 170–177. – DOI: 10.15827/0236-235X.146.170-177.
2. Choudhary S. M., Patel A. S., Parmar S. J. Study of LZ77 and LZ78 data compression techniques // International Journal of Engineering Sciences & Innovative Technology. – 2015. – Т. 4, № 3. – С. 45–49.
3. Shannon C. E. A mathematical theory of communication // The Bell System Technical Journal. – 1948. – Т. 27, № 3. – С. 379–423. – DOI: 10.1002/j.1538-7305.1948.tb01338.x.
4. Oswal S., Singh A., Kumari K. Deflate compression algorithm // International Journal of Engineering Research and General Science. – 2016. – Т. 4, № 1. – С. 430–436.
5. Adjero D., Bell T., Mukherjee A. The Burrows-Wheeler Transform: Data compression, Suffix Arrays, and Pattern Matching. – New York: Springer, 2008. – 352 с. – DOI: 10.1007/978-0-387-78909-5.