УДК 004.45 **Технические науки**

Налисник Алексей Николаевич, *студент магистр, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»*

Белов Юрий Сергеевич, *к.ф.-м.н., доцент, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»*

АВТОМАТИЧЕСКАЯ НАСТРОЙКА КОНФИГУРАЦИЙ NOSQL БАЗ ДАННЫХ ДЛЯ МАКСИМИЗАЦИИ ПРОИЗВОДИТЕЛЬНОСТИ И МАСШТАБИРУЕМОСТИ В ПРИЛОЖЕНИЯХ С ИНТЕНСИВНЫМ ИСПОЛЬЗОВАНИЕМ ДАННЫХ

Аннотация: Базы данных NoSQL все чаще используются в качестве комплексных и качественных решений прорывных научных и производственных задач во всем мире. Однако существует ряд ограничений по быстродействию и реконфигурации, что негативно сказывается на производительности, качестве и гибкости системы. В статье рассматривается один из основных способов автоматической настройки оптимальной конфигурации для специфических приложений, использующих большие объемы данных и требующих высокой производительности.

Ключевые слова: Базы данных NoSQL, автоматическая реконфигурация, оптимизация, масштабируемость, распределенные системы, производительности.

Abstract: NoSQL databases are increasingly being used as complete and quality solutions to scientific and industrial problems around the world. However, there are a number of speed and reconfiguration limitations that negatively affect the performance, quality, and flexibility of the system. This article uses one of the main

tuning options for specific applications that use large amounts of data and require high performance.

Keywords: NoSQL databases, automatic reconfiguration, optimization, scalability, distributed systems, performance.

Введение. Одними из немногих примеров приложений для определения и применения конфигураций системы NoSQL для нескольких параллельных целей оптимизации (например, пропускной способности, задержки, стоимости) на основе количественной оценки различных тактик производительности и масштабируемости являются различные виды промежуточного программного обеспечение по автоматической настройке конфигурации (рис. 1). Данные виды программного обеспечения решают проблему оптимизации во время выполнения, отслеживая рост данных, изменения в сочетании чтения/записи/запроса во время выполнения, а также другие системные показатели, которые указывают на неоптимальную производительность. В рассматриваемой разработке используется контролируемое машинное обучение на основе исторической и текущей информации мониторинга и соответствующих конфигураций для выбора наилучших комбинаций высокоуровневых тактик и адаптации систем NoSQL к изменяющимся рабочим нагрузкам [1].

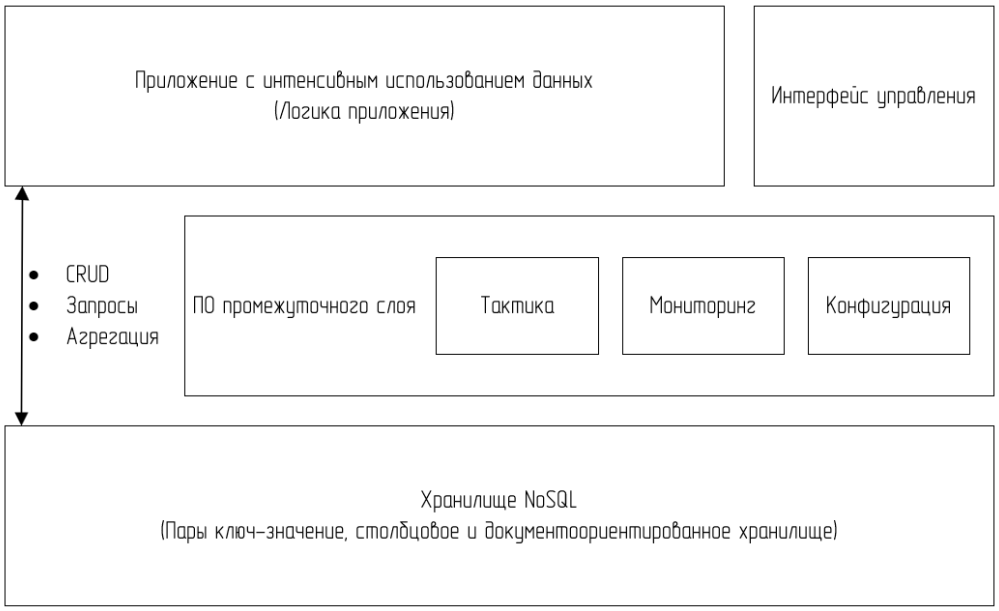

Рис. 1. Архитектура оптимизационного программного обеспечения на базе слоя реконфигурации.

Характеристика рабочей нагрузки и отображение конфигурации систем

Производительность и масштабируемость приложений, интенсивно использующих данные, зависят не только от развертывания и конфигурации распределенной системы хранения NoSQL, но и от рабочей нагрузки, которую приложение, интенсивно использующее данные, возлагает на базовую систему хранения с точки зрения операций чтения, записи и запросов. Однако выбор наиболее эффективной технологии, настройка масштабируемости развертывания и настройка параметров во время выполнения для оптимального предоставления услуг остаются сложными задачами. Чтобы настроить распределенные системы хранения NoSQL для повышения производительности и масштабируемости, промежуточное программное обеспечение отслеживает рост данных, изменения в сочетании чтения/записи/запросов во время выполнения, а также другие системные показатели, свидетельствующие о неоптимальной производительности и нарушениях SLA. Используя контролируемое машинное обучение на основе исторической и текущей информации мониторинга и соответствующих конфигураций, промежуточное программное обеспечение затем сопоставляет рабочие нагрузки приложений с конфигурациями распределенных систем, чтобы адаптироваться к изменяющимся рабочим нагрузкам. Общий подход показан на рисунке 2 [5].

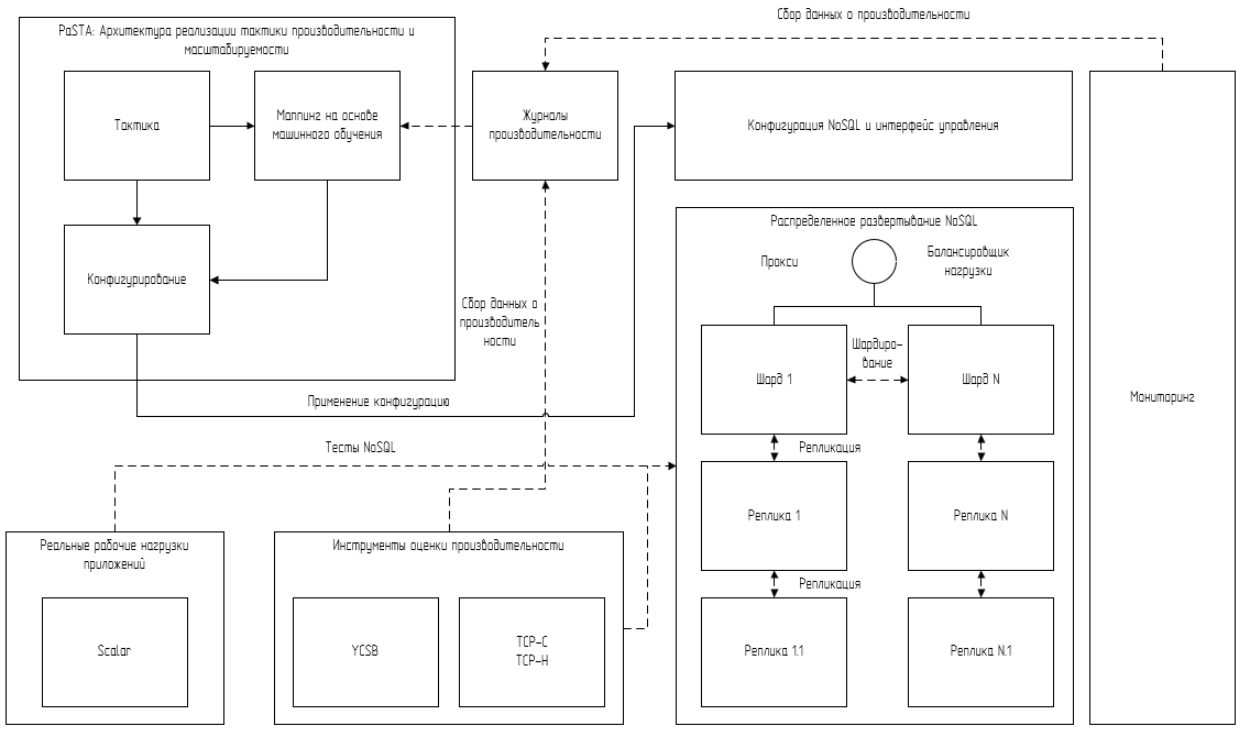

Рис. 2. Сопоставление тактик производительности и масштабируемости на основе машинного обучения с помощью конфигурации времени выполнения в системах NoSQL.

Мотивация для использования информации мониторинга из инструментов сравнительного анализа, не зависящих от приложений, заключается в том, что они обычно охватывают более широкий спектр, хотя и смоделированных рабочих нагрузок приложений. Таким образом, промежуточное программное обеспечение может использовать грубую аналитическую информацию, чтобы справляться с невидимыми рабочими нагрузками реальных приложений.

Есть несколько ограничений, связанных с универсальными инструментами сравнительного анализа технологий, которые позволяют принимать обоснованные решения по адаптации или реконфигурации. Например, они не могут дать никаких гарантий того, что требования QoS для конкретных приложений в отраслевых примерах, будут выполнены. Кроме того, они обычно проверяют только общий знаменатель функциональности системы хранения, обычно операции CRUD на основе поля первичного ключа. Невозможно систематически сравнивать влияние вторичных индексов, агрегаций или более сложных запросов. Кроме того, тесты на стороне клиента

не отслеживают потребление ресурсов на серверах (ЦП, память, сеть), что затрудняет объяснение узких мест с точки зрения технологий NoSQL по сравнению с насыщением ресурсами.

Основные элементы параметров для формирования конфигурации

Набор функций, используемых в качестве входных данных для конвейера машинного обучения для сопоставления рабочих нагрузок приложений с конфигурациями системы NoSQL, состоит из следующих функций:

- *Потребление ресурсов.* Для каждого узла в кластере собираются системные метрики об использовании ЦП, памяти и сети, и эти показатели агрегируются по разным скользящим окнам (5 с, 30 с и 5 мин).
- *Транзакции данных.* Количество операций чтения, сканирования, записи, обновления и запроса, их продолжительность, размер обработанных данных, снова агрегированных за те же временные интервалы.
- *Конфигурация системы хранения и переход.* Постоянно поддерживаются два набора атрибутов (описывающих тактику производительности и масштабируемости), первый из которых представляет текущую конфигурацию, а второй список представляет предполагаемое состояние системы. Таким образом отслеживаются переходы конфигурации.
- *Ключевые показатели производительности.* Эти числовые атрибуты показывают, в какой степени SLA для конкретного приложения были выполнены (или нарушены).
- *Информация о состоянии.* Количество пользователей и данных в системе NoSQL, а также является ли текущая конфигурация системы стабильной или изменяющейся (включая текущую продолжительность изменения в секундах).

С помощью вышеуказанного набора атрибутов получается непрерывный оперативный обзор состояния и производительности распределенной системы NoSQL. Для развертывания кластера из 5 узлов получается около 200–230 атрибутов, которые передаются алгоритму машинного обучения для отображения конфигурации. Причина такого большого количества атрибутов

заключается в том, что определенные категориальные характеристики (например, текущая и планируемая конфигурация системы хранения) расширяются с помощью прямого кодирования в формат, который делает их более подходящими для алгоритмов классификации и регрессии [4].

Были декларированы несколько тактик повышения производительности и масштабируемости для конкретных случаев применения, однако изменения конфигурации в этих случаях эффективны только в определенных условиях эксплуатации. Например, репликация для рабочих нагрузок с интенсивным выполнением операций записи увеличивает сетевой трафик между узлами реплик, что ставит под угрозу пропускную способность приложения, а также может негативно повлиять на задержку запросов из-за управления очередями или согласованностью и разрешения конфликтов. Разделение также приводит к снижению производительности, поскольку служба поиска прокси должна определить, на какой сегмент должна быть направлена операция, или даже объединить результаты запросов из разных сегментов. Вторичные индексы в основном полезны для рабочих нагрузок с интенсивным чтением (с минимальными накладными расходами на обслуживание индекса), когда сами индексы могут храниться в памяти. Таким образом, существует компромисс между использованием памяти для (частичного) кэширования или вторичных индексов.

Реализация промежуточного программного обеспечения

Промежуточное программное обеспечение включает три уровня. На верхнем уровне оно предлагает владельцу приложения панель управления, поддерживающую наблюдение за производительностью и соответствием QoS/SLA, и возможное одобрение изменений конфигурации, предлагаемых ядром приложения. Компонент мониторинга производительности промежуточного программного обеспечения собирает показатели потребления ресурсов для узлов в кластере с помощью готовых инструментов мониторинга производительности. Кроме того, специальные инструменты мониторинга производительности базы данных NoSQL собирают статистику времени

выполнения и показатели самих транзакций. Необработанные измерения и нормализованные метрики передаются в компонент мониторинга политик SLA, который проверяет соответствие целям QoS/SLA, настроенным архитектором приложения в компоненте администрирования политик SLA через панель управления. Компонент мониторинга политик SLA автономно запускает компонент выбора и сопоставления конфигураций, который использует знания и входные данные из базы знаний и репозитория конфигураций текущих и предыдущих конфигураций [2].

На нижнем уровне промежуточное программное обеспечение включает в себя поддержку мониторинга, чтобы дополнить основные функции онлайн-измерениями для фактического количественного определения характеристик производительности и масштабируемости работающих приложений. Эта важная поддержка мониторинга не тесно интегрирована с ядром промежуточного программного обеспечения, чтобы гибко использовать существующие и новые компоненты мониторинга третьих сторон. В основе архитектуры промежуточного программного обеспечения лежит ядро, которое может автономно конфигурировать и реконфигурировать уровень хранения для оптимизации характеристик производительности приложения, интенсивно использующего данные, во время выполнения. Компонент базы знаний по тактике содержит общие сведения об эталонных тестах и список тактик, которые оказывают положительное или отрицательное влияние на пропускную способность с точки зрения количества операций NoSQL в секунду, распределение задержки этих операций NoSQL, потребление ресурсов и тип операции. И наконец, компонент автоматизации развертывания подготавливает и настраивает узлы репликации или сегментов в кластере NoSQL [3].

Заключение. Программное обеспечение для автоматической настройки конфигураций NoSQL баз данных используется в узкоспециализированных проектах, где есть необходимость максимизировать производительность и масштабируемость на больших наборах данных, например, для исследования метагеномики или для анализа больших данных из озер данных. Решение

нахождения лучшей конфигурации и способы поиска постоянно улучшаются и модифицируются за счет применения новых развивающихся технологий, таких как машинное обучение, что даёт прорывные, устойчивые и необходимые решения в области репликации и реконфигурации.

Библиографический список:

1. Липатова С.Е. Белов Ю.С. Принцип работы контроллера Statefulset Kubernetes для управления доступностью приложений с отслеживанием состояния на основе микросервисов // Высокие технологии и инновации в науке. В сборнике: Избранные статьи Международной научной конференции. 2022. С. 112–115.

2. Налисник А.Н. Белов Ю.С. Оптимизация производительности хранилищ данных NoSQL // Наукоемкие технологии в приборо- и машиностроении и развитие инновационной деятельности в вузе. Материалы Всероссийской научно-технической конференции. Издательство МГТУ им. Н. Э. Баумана. 2022. Т. 2. С. 214–217.

3. Das S., Nishimura S., Agrawal D., El Abbadi A. Albatross: lightweight elasticity in shared storage databases for the cloud using live data migration // Proceedings of the Very Large Database Endowment. 2021. pp 494–505.

4. Ghosh M., Wang W., Holla G., Gupta I. Morphus: Supporting Online Reconfigurations in Sharded NoSQL Systems // IEEE 12th International Conference on Autonomic Computing. 2019. pp. 1–9.

5. Klimovic A., Litz H., Kozyrakis C. Selecta: Heterogeneous cloud storage configuration for data analytics // USENIX Annual Technical Conference. 2018. pp. 759–773.

6. Mainak G., Wenting W., Gopalakrishna H., Indranil G. Morphus: Supporting Online Reconfigurations in Sharded NoSQL Systems // IEEE 12th International Conference on Autonomic Computing. 2021. pp. 1–10.

7. Müller I., Marroquín R., Alonso G. Lambada: Interactive data analytics on cold data using serverless cloud infrastructure // Proceedings of the 2020 ACM

SIGMOD International Conference on Management of Data. 2020. pp. 115–130.

8. Preuveneers D., Joosen W. Automated Configuration of NoSQL Performance and Scalability Tactics for Data-Intensive Applications // 1st Workshop on Performance Analysis of Big Data Systems. 2020. pp. 5–10.