

ОЦЕНКА ЭФФЕКТИВНОСТИ БАЗ ДАННЫХ NOSQL ДЛЯ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНОГО СИТУАЦИОННОГО ЦЕНТРА

ASSESSMENT OF THE EFFECTIVENESS OF NOSQL DATABASE FOR INTEGRATED SITUATIONAL CENTER PROBLEMS

**V. Simankov
M. Drilenko**

Summary. Increasing requirements for scalability and elasticity of data storage for web applications has made Not Structured Query Language NoSQL databases more invaluable to web developers. One of such NoSQL Database solutions is Redis. A budding alternative to Redis database is the SSDB database, which is also a key-value store but is disk-based. The aim of this research work is to benchmark both databases (Redis and SSDB) using the Yahoo Cloud Serving Benchmark (YCSB). YCSB is a platform that has been used to compare and benchmark similar NoSQL database systems. Both databases were given variable workloads to identify the throughput of all given operations. The results obtained shows that SSDB gives a better throughput for majority of operations to Redis's performance.

Keyword: NoSQL Redis SSDB database Yahoo Cloud Serving Benchmark (YCSB)

Симанков Владимир Сергеевич

*Д.т.н., профессор, Кубанский Государственный
Технический Университет (Краснодар)
vs@simankov.ru*

Дриленко Максим Владимирович

*Аспирант, Кубанский Государственный Технический
Университет (Краснодар)
mvdrilenko@gmail.com*

Аннотация: Повышение требований к масштабируемости и эластичности хранения данных для веб-приложений сделало неструктурированные базы данных языка запросов NoSQL более ценными для веб-разработчиков. Одним из таких решений для баз данных NoSQL является Redis. Многообещающей альтернативой базе данных Redis является база данных SSDB, которая также является хранилищем значений ключей, но является дисковой. Целью данной работы является оценка обеих баз данных (Redis и НПО), используя облака в Yahoo служба эталоном (YCSB). YCSB - это платформа, которая использовалась для сравнения и тестирования аналогичных систем баз данных NoSQL. Обеим базам данных были предоставлены переменные рабочие нагрузки для определения пропускной способности всех заданных операций. Полученные результаты показывают, что SSDB дает лучшую пропускную способность для большинства операций производительности Redis.

Ключевые слова: NoSQL , Redis , SSDB база данных, облачное хранилище Yahoo, Бенчмаркинг (YCSB).

Введение

Ситуационный центр — информационная аналитическая система, базирующаяся на распределенной системе баз данных, которые должны иметь высокую производительность. Проблемы производительности зависят не только от скорости обработки, но и от возможных уязвимостей безопасности данных, которые возникают из-за характера и характеристик больших данных (большой объем, скорость, разнообразие и достоверность данных) [2]. Одной из таких уязвимостей является необходимость запросов к зашифрованным базам данных без снижения производительности приложений [3, 4].

Целью данного исследования является нахождение оптимальной системы управления базами данных на основе NoSQL, соответствующую трём важным критериям при обработке данных СЦ [1]:

- ◆ Быстродействие (по скорости чтения/записи/изменения строки);
- ◆ Адаптируемость (к потребностям ситуационных центров);

- ◆ Надежность (возможность организации взаимодействия распределенных структур).

В результате анализа представленных систем управления базами данных (СУБД) выбраны две представляющие интерес СУБД NoSQL, а именно SSDB и Redis, которые распространяются по лицензии BSD и соответствуют указанным критериям. При помощи средств компьютерного моделирования разработана соответствующая программная модель.

Redis и SSDB — это хранилища структуры данных в памяти с открытым исходным кодом (лицензия BSD), используемое в качестве базы данных, кэша и брокера сообщений [7]. Модель данных является ключ-значение, хотя поддерживаются различные типы данных: строки, списки, наборы, сортированные наборы, хэши, гиперлоги и растровые изображения [10]. Средства имеют встроенную репликацию, она может быть реплицирована с помощью master-slave баз (основной-подчиненной) модели [2], то есть master (основная база) может иметь несколько slave (подчиненных баз) [13, 3], чем обеспечивают доступ к изменяемым структурам данных с помощью набора команд, которые отправляются с использо-

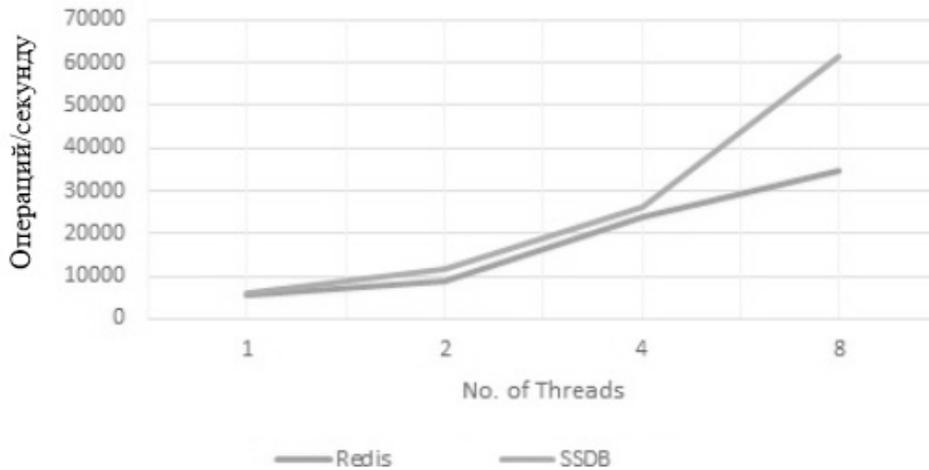


Рис. 1. Модификация файлов

ванием модели сервер-клиент с TCP-сокетами и простым протоколом [11].

Указанные инструменты можно использовать в качестве кэша LRU (least recently used) — это алгоритм, при котором вытесняются значения, которые дольше всего не запрашивались, используя приближенный алгоритм LRU для вытеснения старых данных при добавлении новых [9]. Он также предлагает возможность написания сценариев с использованием Lua — мощного, легкого скриптового языка, написанного на C и встроенного в оба указанных средства [8]. Redis и SSDB поддерживают автоматический переход, если master база не работает, как ожидалось, с этого начинается процесс обработки отказа, где slave становится master, дополнительные slave также перенастроены на новые master [12]. Сегментирование выполняется через кластеры платформ, где данные автоматически распределяются по нескольким узлам [14].

Среди ключевых преимуществ рассматриваемых продуктов — заявленная более быстрая и эффективная производительность, чем устаревшие системы управления реляционными базами данных (СУБД) [1]. Базы данных NoSQL также адаптированы к облачным вычислениям, что позволяет осуществлять масштабирование “по требованию» (эластичность) и упрощенную разработку приложений [2]. Тем не менее, есть ограничения в разработке больших данных и веб-приложений, отмечая, что не все базы данных NoSQL создаются одинаково, когда речь идет о производительности [3]. Утверждается [4], что, поскольку решения NoSQL не являются в достаточной мере всеобъемлющими и развиваются с разной скоростью, администраторы баз данных должны тщательно выбирать между NoSQL и реляционными базами данных в соответствии с их конкретными потребностями с точки зрения согласованности, производительности, безопас-

ности, масштабируемости, затрат и других нефункциональных критериев.

Планирование эксперимента и спецификации системы

Оценка производительности баз данных NoSQL с любыми стандартными тестами осуществляется с использованием двух подходов: трассировки и генерации векторной нагрузки (операций с базами данных) [12, 13]. Системы тестирования на основе трассировки используют фактическую рабочую нагрузку приложений, созданную из приложений. В то время как векторные тестовые системы создают поведение приложения, используя векторы и применяя векторы, используя известные статистические модели распределения, имитируя фактический запрос приложения и ответ в аппаратной или виртуальной платформе. Инструменты оценки производительности можно классифицировать на встроенные и пользовательские. Примером первого является redis-benchmark. Пользовательские инструменты тестирования включают Yahoo! Cloud System Benchmark (YCSB) [4], BigBench [4] и GraySort [4]. Целью этого процесса оценки производительности является сравнение производительности баз данных Redis и SSDB NoSQL с использованием экземпляров одного узла. Результаты подтверждают утверждения о пригодности SSDB в качестве альтернативы Redis, как предложили авторы [11]. Выбранным инструментом — является YCSB — спецификация и программный пакет с открытым исходным кодом для оценки возможностей поиска и обслуживания компьютерных программ. Он часто используется для сравнения относительной производительности систем управления базами данных NoSQL, архитектура на основе плагинов YCSB и простота расширения с помощью скриптов [10] делают его великолепным выбором. Согласно публикации [7], описывается два уровня тестов YCSB, а именно: производительность и масштабирование. Основное вни-

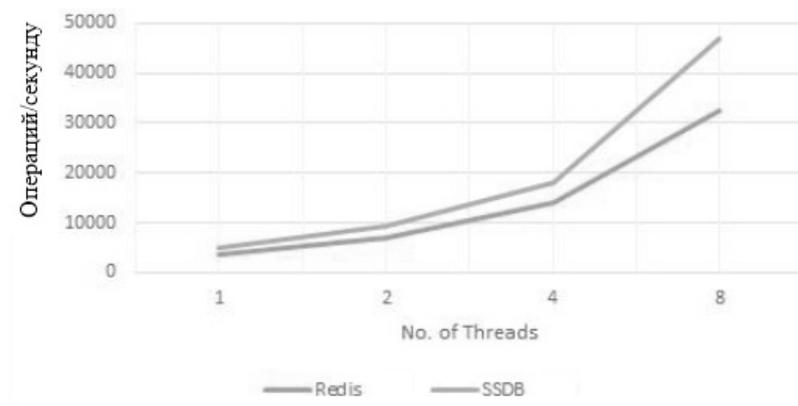


Рис. 2. Запись данных

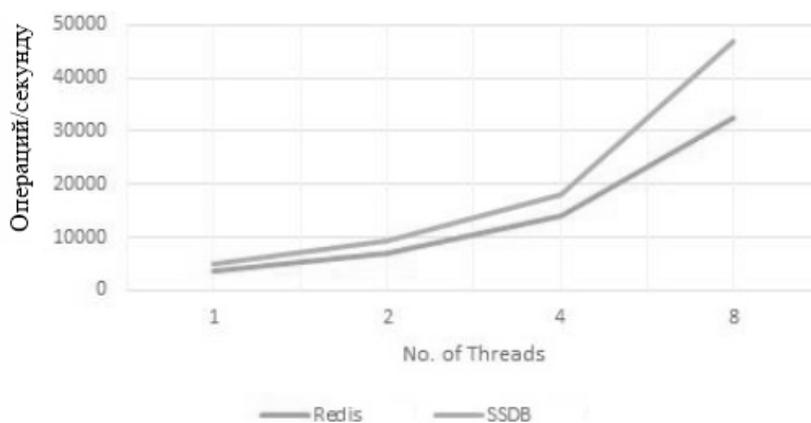


Рис. 3. Чтение данных

мание в настоящем исследовании уделяется оценке производительности SSDB по сравнению с Redis для ряда рабочих нагрузок, которые наиболее характерны при работе ситуационного центра. Эти рабочие нагрузки имитируют различные типы поведения запросов веб-приложений, такие как интенсивное чтение и запись. Рассматриваемые рабочие нагрузки включают:

Модификация (Update)

В этой рабочей нагрузки 50% операций чтения и на 50% записи.

Нагрузки Б (Write)

В этой рабочей нагрузки 95% операций чтения и остальные 5% это запись.

Рабочая Нагрузка С (Read)

Рабочая нагрузка с 100% операций чтения.

Рабочая Нагрузка D (Write & Update)

Нагрузки с 95% операций чтения и 5% операций вставки. Рабочая нагрузка вставляет новые записи и са-

мые последние вставленные записи являются наиболее популярными.

Рабочая Нагрузка F

Рабочая нагрузка, при которой клиент будет осуществлять чтение, изменение и обновление записи (Update). Ниже приведены сведения о конфигурации системы и спецификации для процесса оценки производительности.

Проведенные машинные эксперименты реализованы в при одинаковой конфигурации и спецификации системы.

Результаты и анализ

Нагрузка А. Модификация (Update)

Результаты сложных операций обновления (с одновременным изменением записей, 1000 записей). В результате SSDB превосходит Redis по количеству потоков и его пропускная способность расходится с увеличением количества потоков. Это говорит о том, что увеличе-

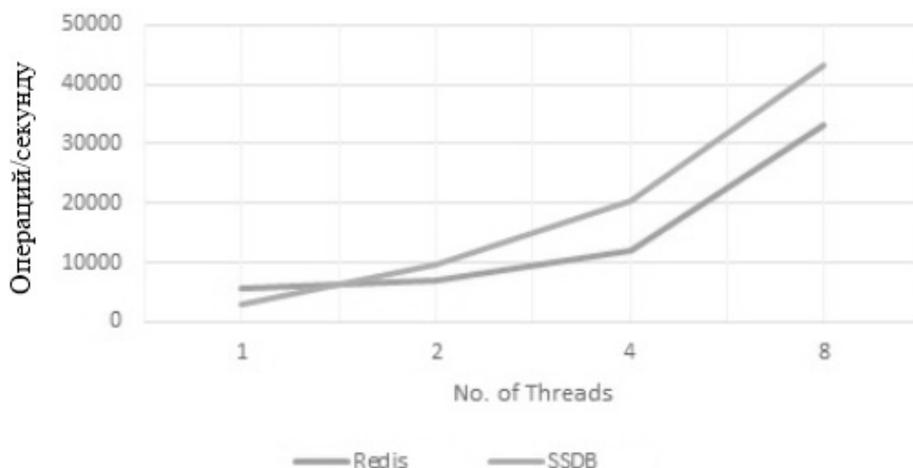


Рис. 4. Операция чтения и записи

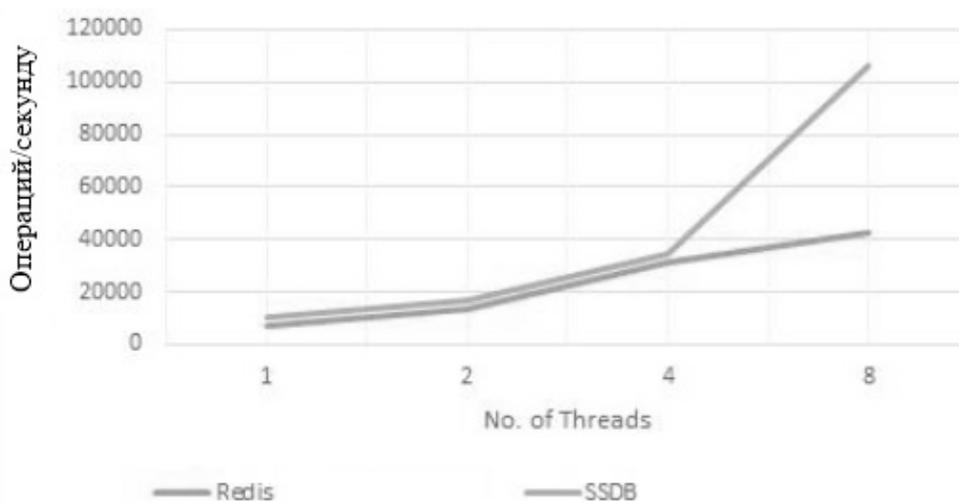


Рис. 5. Операции «чтение-модификация-запись»

ние потоков программного обеспечения увеличит производительность SSDB (рис. 1).

Нагрузки Б. Запись (Write)

В результате рабочей нагрузки, как показано на рис. 2, имеется сходство с операцией обновления. SSDB превосходит Redis для операций чтения. Однако это преимущество производительности, по-видимому, теряется, поскольку количество записей приближается к 10000. Redis явно восстанавливает потерянное на этом уровне.

Нагрузка С. Чтение (Read)

Как показано Сравнение Redis и SSDB с 1000 записями и 1000 операциями, пропускная способность SSDB отличается от пропускной способности Redis по мере увеличения числа развернутых потоков (рис. 3).

Нагрузка D.

Запись и модификация (Write & Update)

Сравнение Redis и SSDB с 1000 записями и 1000 операциями показывает, как в результате пропускная способность SSDB превосходит Redis. Это также показывает повышение пропускной способности, когда число потоков увеличивается до 8 (рис. 4).

Нагрузка F. Чтение-модификация-запись (Read-Update-Write)

В рабочей нагрузке «чтение-изменение-запись» (сравнение Redis и SSDB с 1000 записями и 1000 операциями), имеются оттенки сходства с рабочей нагрузкой D. SSDB отстает от Redis соразмерно с количеством потоков от 1, 2 и 4 соответственно. Однако расхождение наиболее заметно когда потоков 8.

Проведенное сравнение эффективности осуществления операций базируется на применении интенсивных алгоритмов обработки данных. Кроме того, совокупность проведенных экспериментов показывает, что увеличение числа потоков увеличивает скорость обработки информации, в среднем, на 86% (при добавлении каждого дополнительного потока).

Вывод

В проведенном исследовании рассмотрены системы управления базами данных формата NoSQL, такие как Redis и SSDB. Выявлены отличительные особенности выбранных программных средств. Для серии проведенных машинных экспериментов использовались только единичные экземпляры узлов обеих баз данных. Задействовались шесть общих нагрузок YCSB.

Установлено, что SSDB показывает себя эффективнее (по результатам 5 экспериментов, циклы A-F), чем Redis. Вместе с тем, заметно сокращение числа операций чтения-изменения и записи. Это показывает, что при реше-

нии задач ситуационного центра следует рассмотреть вопрос об использовании SSDB для масштабирования в сценариях приложений, где выполняются обновления, операции чтения, изменения и записи. В то время как Redis следует использовать для коротких или небольших (<10 000) запросов сканирования диапазона, что делает его непригодным для использования в СЦ.

Кроме того, при решении задач ситуационного центра, направленных на масштабирование и одновременную обработку неструктурированной информации при наличии количества потоков обработки более восьми с дальнейшей репликацией извлеченных и обработанных сведений для предоставления конечному потребителю SSDB является наиболее производительным инструментом по результатам проведенных экспериментов, а именно количеству обрабатываемых операций.

Таким образом, сравнительный анализ двух выбранных программных продуктов указывает на целесообразность использования SSDB для решения задач обработки данных в интеллектуальном ситуационном центре.

ЛИТЕРАТУРА

1. C.U. Kumarasinghe, K.L.D.U. Liyanage, W.A.T. Madushanka and R.A.C.L. Mendis. (2015, September). Performance Comparison of NoSQL Databases in Pseudo Distributed Mode: Cassandra, MongoDB & Redis [Online].
2. B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with ycsb". In Proceedings of the 1st ACM symposium on Cloud computing (New York, NY, USA, 2010), SoCC '10, ACM, pp.143–154.
3. Y. Abubakar, T. S. Adeyi, and I. G. Auta, "Performance Evaluation of NoSQL Systems using YCSB in a Resource Austere Environment," International Journal of Applied Information Systems, vol. 7, pp. 23–27, Sep. 2014.
4. Oussous, F. Z. Benjelloun, A. A. Lahcen, S. Belfkih, "Comparison and Classification of NoSQL Databases for Big Data," in Proc. of the 2015 International Conference on Big Data, Cloud and Applications, BDCA 2015, 25–26 May 2015, Tetuan, Morocco [Online]. Available: ResearchGate, <https://www.researchgate.net>. [Accessed: 24 June 2016].
5. H. Khazaei, M. Fokaefs, S. Zareian, N. Beigi-Mohammadi, B. Ramprasad, M. Shtern, P. Gaikwad, and M. Litoiu. "How do I choose the right NoSQL solution? A comprehensive theoretical and experimental survey." In: Submitted to Journal of Big Data and Information Analytics(BDIA)(2015).[Online]. Available: https://www.researchgate.net/profile/Hamzeh_Khazaei/publication/282679529_How_Do_I_Choose_The_Right_NoSQL_Solution_A_Comprehensive_Theoretical_And_Experimental_Survey/links/5618781808ae044edbad2437.pdf. Accessed: Jun. 29, 2016.
6. E. Anderson, X. Li, M. A. Shah, J. Tucek, and J. J. Wylie, "What consistency does your key-value store actually provide?" HotDep, vol. 10, pp. 1–16, 2010.
7. B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny. Workload analysis of a large-scale key-value store.
8. In Proceedings of the SIGMETRICS'12, June 2012.
9. K. Ma, A. Abraham, "Toward lightweight transparent data middleware in support of document stores" in WICT 2013: Proceedings of the third World Congress on Information and Communication Technologies (2013)
10. C. Chasseur, Y. Li, and J. M. Patel. Enabling JSON document stores in relational systems. In Proceedings of the 16th International Workshop on the Web and Databases (WebDB), pages 1–6, 2013.
11. R. Cattell, "Scalable sql and nosql data stores," ACM SIGMOD Record, vol. 39, no. 4, pp. 12–27, 2010. [11] D. J. Abadi. Column stores for wide and sparse data. In CIDR, Asilomar, CA, USA, 2007.
12. V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional expansion for keyword search on graph databases," in Proc. of VLDB Conference, 2005, pp. 505–516.
13. J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," In Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, Oct. 2011, pp. 363–366.