

М. А. Устимкина, А. В. Густов, Е. М. Тиманин, Е. В. Еремин, С. В. Макушина, Е. В. Паришина // Медицинский Альманах - № 1 (14) - март 2011 - С. 191-193. **10.** Ткаченко, А. Kaddr.com [Электронный ресурс]: обзоры: обзор Sony Alpha 7 II. Первая в мире полнокадровая камера с матричной 5-осевой стабилизацией [Текст] / А. Ткаченко. Режим доступа: <http://kaddr.com/2015/03/obzor-sony-alpha-7-ii-pervaya-v-mire-polnokadrovaya-kamera-s-matrichnoj-5-osevoj-stabilizatsiej/>

Bibliography (transliterated): **1.** Demin, A. V. (1997). Series binoculars with built-in image stabilization system. Opto-electronic devices and systems. Coll. scientific articles. Issue 97. Edited by prof. E. D. and Professor Pankov, 8 -15. **2.** Froimson, I. M., Tsyvkin, R. V. (2001). Useful action observation devices with image stabilization. Special equipment, 5, 60- 63. **3.** Froimson, I. M. (2002). Image Stabilization in the observational instruments. Special equipment, 6, 16-24. **4.** Nikiforov, V. O., Gutner, I. E., Sergachev, I. V. (2004). Active vibration protection system: development, test results and prospects of the. Mechatronics,

Automation and management, 2, 13-18. **5.** Kolovskiy, M. Z. (1976). Automatic control systems, vibration isolation. Moscow: Nauka. **6.** Frolov, K. V., Fuhrman, F. A. (1980). Applied Theory of vibration isolation systems. Moscow: Mechanical Engineering. **7.** Wang, T., Tian, B., Liang, C., Shi, D. (2008). Blind Image Quality Assessment for Measuring Image Blur. Congress on Imagend Signal Processing. **8.** Droniuk, I. Open Academic Journals Index [Electron resource]: Overall image quality for printed and electronic documents. Access: <http://oaji.net/articles/2015/1872-1428763302.pdf>. **9.** Aleksandrova, E. A., Ustimkina, M. A., Gustov, V., Timanin, E. M., Eremin, E. V., Makushina, S. V., Parshina, E. V. (2011). Accelerometer diagnostic power of tremor in Parkinson's disease during therapy. Medical Almanac number 1 (14) March 2011, 191-193. **10.** Tkachenko, A. Kaddr.com [electronic resource]: review: review Sony Alpha 7 II. The world's first full-frame camera with a matrix of 5-axis stabilization. Access: <http://kaddr.com/2015/03/obzor-sony-alpha-7-ii-pervaya-v-mire-polnokadrovaya-kamera-s-matrichnoj-5-osevoj-stabilizatsiej/>.

Надійшла (received) 27.05.2015

Відомості про авторів / Сведения об авторах / About the Authors

Нечай Сергій Олексійович – кандидат технічних наук, Національний технічний університет України «Київський політехнічний інститут», доцент кафедри приладобудування; тел.: 050-351-63-19; e-mail: nechay_s@ukr.net.

Нечай Сергей Алексеевич – кандидат технических наук, Национальный технический университет Украины «Киевский политехнический институт», доцент кафедры приборостроения; тел.: 050-351-63-19; e-mail: nechay_s@ukr.net.

Nechai Sergii Oleksiiovich – Candidate of Technical Sciences (Ph. D.), Docent, National Technical University Ukraine "Kyiv Polytechnic Institute", Associate Professor at the Department of instrument engineering; tel.: 050-351-63-19; e-mail: nechay_s@ukr.net.

Хильченко Тетяна Валентинівна – студентка, Національний технічний університет «Харківський політехнічний інститут», кафедра приладобудування; тел.: 093-897-33-21; e-mail: xvulunka@mail.ru.

Хильченко Татьяна Валентиновна – студентка, Национальный технический университет «Харьковский политехнический институт», кафедра приборостроения; тел.: 093-897-33-21; e-mail: xvulunka@mail.ru.

Khylchenko Tetiana – National Technical University Ukraine "Kyiv Polytechnic Institute", Department of instrument engineering; tel.: 093-897-33-21; e-mail: xvulunka@mail.ru.

УДК 004.652.5

О. В. БУЗОВСКИЙ, А. А. ПОДРУБАЙЛО

ЧАСТИЧНОЕ ПРЕОДОЛЕНИЕ CAP-ТЕОРЕМЫ В РАСПРЕДЕЛЕННОМ ДОКУМЕНТО-ОРИЕНТИРОВАННОМ ХРАНИЛИЩЕ ДАННЫХ

Существующие распределенные хранилища данных в оперативной памяти типа «ключ-значение» объединяют в себе оптимистичность подхода BASE-систем и пессимистичные алгоритмы обновления индексов, рассчитанные на работу в глобальной сети. Данная статья предлагает использование синхронных индексов для повышения консистентности и доступности данных без ущерба для способности системы к горизонтальному масштабированию.

Ключевые слова: CAP-теорема, хранилище данных, консистентность, индекс, доступность, распределенность, ключ-значение, оперативная память, теорема Брюэра, NoSQL.

Введение. Реляционные базы данных обязаны удовлетворять требования ACID[1] (atomicity, consistency, isolation, durability), т.е. ключевым понятием в таких системах является транзакция, как группа объединенных последовательных операций с базой данных. Транзакция не может быть выполнена частично, не нарушает согласованность системы, не влияет на одновременную работу других транзакций[2]. Кроме того, выполненные успешной транзакцией изменения не должны быть потеряны вследствие каких-либо сбоев системы (к примеру, обесточивания либо сбоя в оборудовании). Эти требования хорошо соответствуют традиционным бизнес-моделям, однако в полной мере реализовать их в распределенных

системах, с учетом необходимого быстродействия, крайне сложно.

CAP-теорема, известная также как теорема Брюэра, утверждает, что распределенная система способна обеспечить не более двух из трех свойств: консистентность, доступность, устойчивость к разделению (consistency, availability, partition tolerance) [3-5].

Современные хранилища типа ключ-значение спроектированы с учетом теоремы Брюэра и относятся к BASE-системам, обеспечивая «базовую доступность» и «согласованность в конечном счете» хранимых данных[6,7,8]. Это позволяет обеспечить высокую потенциальную масштабируемость и способность системы к эффективному функционированию

© О. В. Бузовский, А. А. Подрубайло. 2015

при больших расстояниях между узлами. Однако значительная часть реальных случаев использования таких хранилищ демонстрирует разворачивание системы в рамках одной высокоскоростной локальной сети передачи данных. В таких условиях штатные механизмы индексирования распределенных хранилищ, рассчитанные на работу в глобальной сети с большими задержками, становятся неэффективны и приводят к чрезмерным задержкам в обновлении индексов, а, следовательно, к потере консистентности между индексами и индексируемыми данными.

Цель работы. Целью работы является создание метода повышения консистентности и доступности данных, содержащихся в распределенных хранилищах данных в оперативной памяти типа «ключ-значение» без потери устойчивости таких хранилищ к разделению.

Описание методики синхронного индексирования. В данной работе для повышения консистентности и доступности данных в распределенных хранилищах типа «ключ-значение» предлагается ввести синхронные индексы по вторичным полям. Выбор структуры такого индекса должен, с одной стороны, максимально соответствовать идеологии хранения данных, а с другой – обеспечить быстрый и консистентный результат пользовательского запроса.

При этом индекс должен удовлетворять следующим требованиям:

- способностью обеспечить быстрый пользовательский поиск по значению требуемых параметров;
- высокой скоростью операций обновления индекса;
- непрерывной согласованностью индекса и хранимых данных;
- иметь как можно меньшие накладные расходы.

Базовой структурой данных в документ-ориентированных хранилищах является распределенная хэш-таблица[9]. Такая концепция подходит и для хранения индекса, т.к. он ставит в соответствие одному ключу множество значений. Однако, в случае, когда одно и то же значение индексируемого поля имеет множество документов, запись в индексе будет слишком большой. В этом случае для облегчения операций поиска и обновления отдельных участков индекса рекомендуется сегментировать индекс, т.е. разбить каждую запись на сегменты таким образом, чтобы размер каждого сегмента был небольшим.

В общем виде структура сегмента индекса представлена на рис. 1. Такой сегмент хранится как обыкновенный документ, что повышает универсальность подхода и легкость его внедрения.

Сегментировать индексы можно по различным принципам. В случае, когда одному значению индексируемого поля соответствует очень много докумен-

тов, запись разбивается на несколько частей. В случае же, когда одному значению соответствует мало документов, записи можно объединять, разработав определенное правило, позволяющее по требуемому значению поля получить идентификатор индексной записи.

На рис. 2 представлен пример такого индекса с объединенными записями, представляющий программу телепередач.

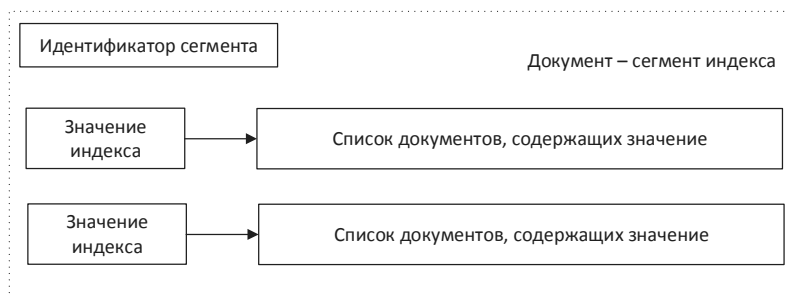


Рис. 1 – Общий вид сегмента синхронного индекса документ-ориентированного хранилища



Рис. 2 – Пример синхронного индекса с объединением записей в сегменте

В этом индексе предполагаются пользовательские запросы вида «получить все телепередачи, активные в заданный промежуток времени на данном канале». Для обработки таких запросов использован составной ключ, включающий идентификатор канала и время. Т.к. в один момент времени на заданном канале может быть активна только одна программа, для экономии предлагается использовать один сегмент для определенного интервала времени (в примере – часа), на протяжении которого существуют активные телепередачи.

Тогда, кроме идентификатора программы, необходимо также хранить действительное время ее начала и длительность, чтобы точно знать, активна ли она в заданный период времени внутри часового промежутка.

При таком подходе одна и та же программа может встречаться в нескольких индексных записях, если она активна сразу в нескольких промежутках времени, что увеличивает избыточность. С другой стороны, предложенная структура удобна при поиске данных для диапазона индексных значений: зная границы

временного промежутка, мы можем сразу вычислить идентификаторы сегментов, используя затем наиболее быстрый метод поиска по ключу.

В случае, когда индексируемые значения не обладают выраженной периодичностью, но требуется обеспечить поиск по диапазону значений, рекомендуется использовать древовидные структуры, в узлах которых содержатся идентификаторы сегментов (рис. 3).

При этом не рекомендуется использовать B+ деревья, как это делается в реляционных базах данных. Как показано в [10], B+ не обеспечивает эффективную работу системы управления базой данных в оперативной памяти, поэтому в данной работе использованы красно-черные деревья.

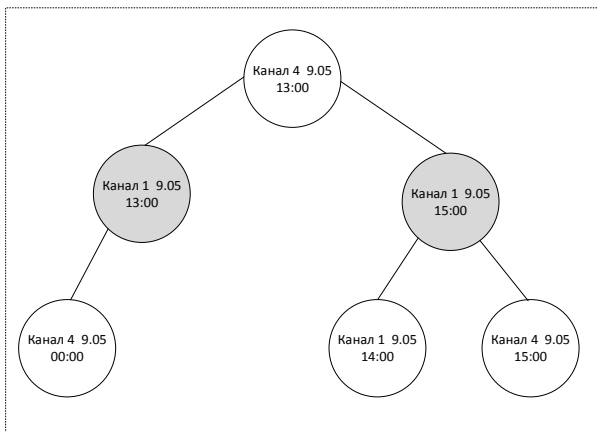


Рис. 3 – Древовидная структура для поиска диапазона индексных значений

Алгоритм обновления индекса. Для обновления индексов при добавлении новых или обновлении существующих документов предлагается использовать следующий алгоритм: вначале предпринимается попытка поиска по ключу старой версии данного документа, уже записанной в данном хранилище. Затем вычисляются индексные записи для старой и новой версий объекта. Под индексной записью здесь понимается элементарная структура, содержащая имя индекса, идентификатор сегмента, к которому относится данная запись и данные, которые должны быть записаны в индекс (рис. 4).

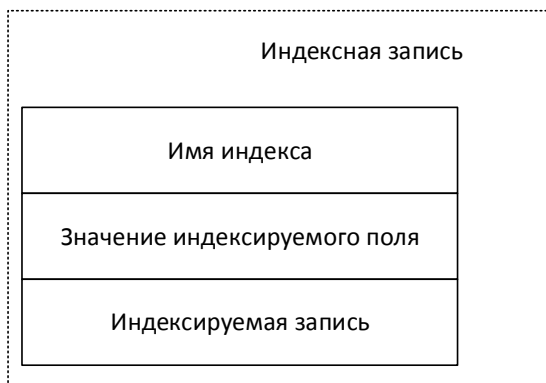


Рис. 4 – Структура индексной записи

Выполняется вычисление разности множества старых и новых индексных записей (если объекта с данным идентификатором ранее не было в хранилище

– множество старых индексных записей принимается пустым). Затем для индексных записей, уже не присутствующих в новом документе, выполняется удаление, а для записей, вновь появившихся – добавление в сегмент.

Описание эксперимента. Для отработки методики синхронного индексирования было выбрано распределенное документ-ориентированное хранилище Couchbase. Данное хранилище демонстрирует высокие скорости чтения/записи документов по ключу, но обладает достаточно несовершенным механизмом выполнения более сложных пользовательских запросов.

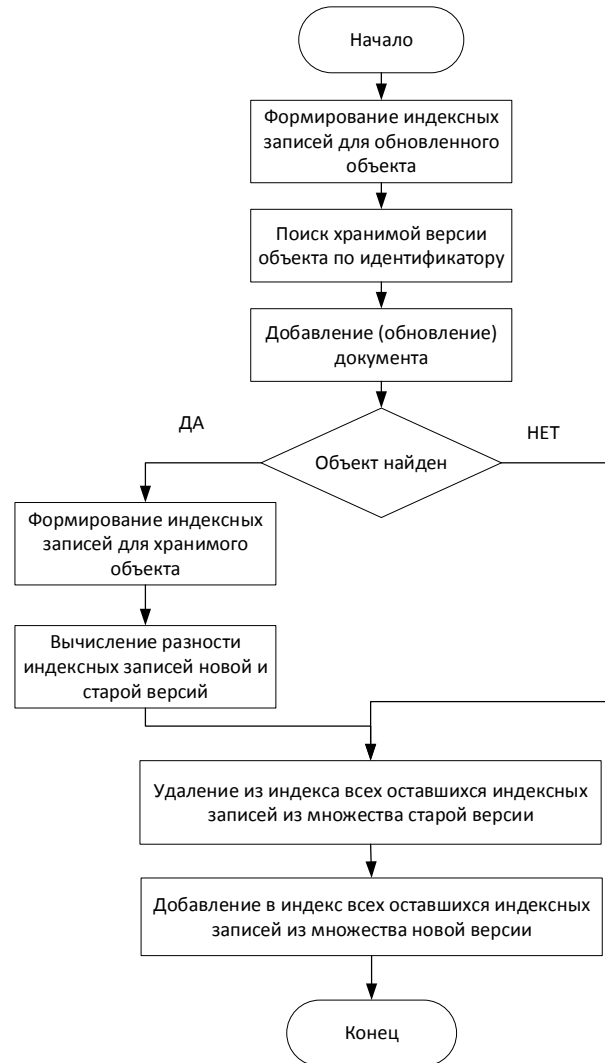


Рис. 5 – Алгоритм обновления индекса при добавлении документа

Для поиска произвольной информации Couchbase предлагает концепцию видов (views), т. е. функций на JavaScript, посредством которых пользователь хранилища определяет содержание индексных записей. Процесс обновления видов запускается только после добавления 5000 новых документов или по истечении пяти секунд после добавления документа, и занимает, для большого хранилища, до десяти минут. Во время обновления видов запросы к ним обрабатывают, возвращая неактуальные (не обновленные) данные. При этом поиск документа по ключу всегда возвращает актуальную версию документа. Таким обра-

зом, учитывая частоту операций добавления документов, Couchbase может служить ярким примером согласованности в конечном счете, демонстрируя отсутствие консистентности между видами и отображаемыми ими документами.

Приведем пример такой ситуации. Предположим, вид отображает документы по содержимому их поля «цвет». Некий документ с идентификатором ID42 содержал в этом поле значение «зеленый», что было отражено соответствующим видом, в котором значению «зеленый» был сопоставлен идентификатор ID42. Затем документ с ID42 был обновлен, и в новой версии поле «цвет» сменило значение на «синий», а, через некоторый период, началось обновление вида. Если пользователь в период с момента обновления документа до окончания процесса обновления вида попытается, воспользовавшись встроенными механизмами, найти документ с цветом «зеленый», он получит идентификатор ID42. При этом, поиск документа по ID42 вернет обновленный документ с цветом «синий». Таким образом, частые операции обновления приводят хранилище в неконсистентное состояние, делая невозможным поиск по чему-либо, кроме идентификаторов документов.

В эксперименте использовались три индекса разных типов: индекс точного соответствия значения поля (индекс А), индекс соответствия значению временных диапазонов (индекс Б), и индекс значений с возможностью поиска по диапазону (индекс В).

Работа с синхронными индексами была реализована при помощи клиентской библиотеки Couchbase Java SDK 2.1 на клиентской стороне. Couchbase сервер состоял из двух узлов, размещенных в одной локальной сети с клиентом.

Методика эксперимента предусматривала добавление в каждый индекс полумиллиона записей, затем 10 тысяч операций выборки, затем добавление еще полумиллиона записей (перекрывание идентификаторов 25 %) с непрерывным выполнением десяти тысяч запросов.

Поведение каждого индекса сравнивалось с поведением соответствующего вида (view), выполнявшего те же задачи.

Результаты эксперимента приведены в табл. 1.

Как видно, время записи документов в пустое хранилище при использовании механизма синхронного индексирования в 1,5-2 раза выше, чем время аналогичной операции с использованием видов. Это объясняется тем, что асинхронная перестройка индексов запускается чуть позже добавления, обеспечивая меньшую нагрузку на аппаратное обеспечение. Наиболее медленно идет добавление в диапазонные индексы, требующие три операции записи и две операции чтения для записи одного объекта.

Выполнение запросов на свободном от других задач хранилище выполняется примерно одинаково как с помощью view, так и посредством синхронных индексов с небольшим опережением первых.

Таблица 1- Результаты измерений времени выполнения и неконсистентности результатов выполнения запросов

Используемые индексы / виды	Добавление $0,5 \cdot 10^6$ записей, с	Выполнение 10^4 запросов, с	Обновление индекса и выполнение запросов, с	Количество неконсистентных результатов, шт
Индекс А	90	100	150	0
Индекс Б	95	107	174	0
Индекс В	184	115	336	0
Индекс А,Б, В	187	110	286	0
View А	61	94	324	110
View В	61	115	321	143
View С	62	110	335	214
View А,В, С	63	107	542	178

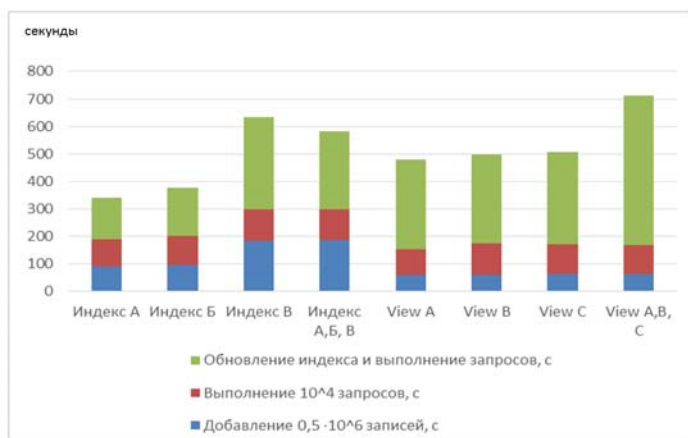


Рис. 6 – Сводная гистограмма времени выполнения, с

Наконец, во время одновременной загрузки и чтения данных асинхронные индексы показывают замедление произво-

дительности вследствие блокировок при обновлении.

На сводной гистограмме (рис. 6) показано суммарное время выполнения всех операций для каждого из вариантов индексов и видов.

Выводы. Метод синхронного индексирования распределенных документо-ориентированных хранилищ данных показал свою применимость при условии, что клиент и сервер находятся в одной локальной сети. Предложенный метод позволил добиться повышения скорости обработки запросов, и, одновременно, консистентности их результатов при одновременной загрузке данных в хранилище. Однако, синхронный индекс, позволяющий запросы по диапазонам, показал уменьшение скорос-

ти обробки запиту по порівнянню з асинхронними індексами, по тому такої вид індекса рекомендується використовувати тільки в тому випадку, коли узгодженість індексів має первинне значення.

Список літератури: 1. *Cattell, R.* Scalable SQL and NoSQL Data Stores [Text] / *R. Cattell* // SIGMOD Record – 12.2010. – Vol. 39, № 4. – P. 12-27. 2. *Gray, J.* The Transaction Concept: Virtues and Limitations [Text] / *J. Gray* // Proceedings of the 7th International Conference on Very Large Databases – 1981. – P. 144-154. 3. *Brewer, E.* CAP Twelve Years Later: How the "Rules" Have Changed [Text] / *Eric Brewer* // IEEE Computer. – 02.2012. – Vol. 45, № 2. – P. 23-29. 4. *Gilbert, S.* Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services [Text] / *Seth Gilbert, Nancy Lynch* // ACM SIGACT News. – 06.2002. – Vol. 33, № 2. – P. 51-59. 5. *Birman, K., Freedman, D., Huang, Q., Dowell, P.* Overcoming CAP with consistent soft-state replication [Text] / *Kenneth Birman, Daniel Fridman, Qi Huang, Patrick Dowell* // IEEE Computer – 02.2012 – Vol. 45, № 2 – P. 50-58. 6. *Gilbert, S.* Perspectives on the CAP Theorem [Text] / *Seth Gilbert, Nancy Lynch* // IEEE Computer. – 02.2012. – Vol. 45, № 2. – P. 30-36. 7. *Pritchett, D.* BASE: an ACID alternative [Text] / *Dan Pritchett* // Queue - Object-Relational Mapping – 06.2008. – Vol.6, №3. – P.48-55. 8. *Bailis, P., Ghodsi, A.* Eventual consistency today: limitations, extensions and beyond [Text] / *Peter Bailis, Ali Ghodsi* // Communications of the ACM – 05.2013. – Vol.56, №5. – P. 55-63. 9. *Gupta, M. K., Verma, V., Verma, M. S.* In-Memory Database Systems - A Paradigm Shift [Text] / *Mohit Kumar Gupta, Vishal Verma, Megha Singh Verma* // In-

ternational Journal of Engineering Trends and Technology (IJETT) – 12. 2013.– P. 333-336. 10. *Шапоренков, Д. А.* Эффективные методы индексирования данных и выполнения запросов в системах управления базами данных в основной памяти: дис. кандидата физ.-мат. наук: 05.13.11 [Текст] / *Шапоренков Дмитрий Александрович.* – СПб., 2006 – 126 с.

Bibliography (transliterated): 1. *Cattell, R.* (2010). Scalable SQL and NoSQL Data Stores. SIGMOD Record, Vol. 39, № 4, 12-27. 2. *Gray, J.* (1981). The Transaction Concept: Virtues and Limitations. Proceedings of the 7th International Conference on Very Large Databases, 144-154. 3. *Brewer, E.* (2012). CAP Twelve Years Later: How the "Rules" Have Changed. IEEE Computer, Vol. 45, № 2, 23-29. 4. *Gilbert, S., Lynch, N.* (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, Vol. 33, № 2, 51-59. 5. *Birman, K., Freedman, D., Huang, Q., Dowell, P.* (2012). Overcoming CAP with consistent soft-state replication. IEEE Computer, Vol. 45, № 2, 50-58. 6. *Gilbert S., Lynch N.* (2012). Perspectives on the CAP Theorem. IEEE Computer, Vol. 45, № 2, 30-36. 7. *Pritchett, D.* (2008). BASE: an ACID alternative. Queue - Object-Relational Mapping, Vol.6, №3, 48-55. 8. *Bailis, P., Ghodsi, A.* (2013). Eventual consistency today: limitations, extensions and beyond. Communications of the ACM, Vol.56, №5, 55-63. 9. *Gupta, M. K., Verma, V., Verma, M. S.* (2013). In-Memory Database Systems - A Paradigm Shift. International Journal of Engineering Trends and Technology (IJETT), 333-336. 10. *Shaporenkov D. A.* (2006). Effective methods of data indexing and querying in main memory database systems. (PhD dissertation).

Поступила (received) 26.05.2015

Відомості про авторів / Сведения об авторах / About the Authors

Бузовський Олег Володимирович – доктор технічних наук, професор, Національний технічний університет України «Київський політехнічний інститут», професор кафедри обчислювальної; тел.: 050-554-95-45; e-mail: obuza38@gmail.com.

Бузовський Олег Владимирович – доктор технических наук, профессор, Национальный технический университет Украины «Киевский политехнический институт», профессор кафедры вычислительной техники; тел.: 050-554-95-45; e-mail: obuza38@gmail.com.

Vuzovskyi Oleh – Doctor of Technical Sciences, Full Professor, National Technical University of Ukraine "Kyiv Polytechnic Institute", professor of the department of computer technology; tel.: 050-554-95-45; e-mail: obuza38@gmail.com.

Подрубайло Олександр Олександрович – асистент, Національний технічний університет України «Київський політехнічний інститут», кафедра обчислювальної техніки; тел.: 050-381-43-34; e-mail: sxl_sas@gmail.com.

Подрубайло Александр Александрович – ассистент, Национальный технический университет Украины «Киевский политехнический институт», кафедра вычислительной техники; тел.: 050-381-43-34; e-mail: sxl_sas@gmail.com.

Podrubailo Oleksandr – assistant, National Technical University of Ukraine "Kyiv Polytechnic Institute", professor of the department of computer technology; tel.: 050-381-43-34; e-mail: sxl_sas@gmail.com.

УДК 656.13

Р. Б. РОГАЛЬСЬКИЙ

ПРОБЛЕМИ ТА ШЛЯХИ УДОСКОНАЛЕННЯ ПАСАЖИРСЬКОГО ОБСЛУГОВУВАННЯ СУЧАСНИХ МІСТ

Проаналізовано проблеми, які виникають, під час організації пасажирського обслуговування на території сучасних великих та значних міст. Основну увагу зосереджено на містобудівних, функціонально-планувальних та транспортних чинниках, які впливають на закономірності у формуванні пішохідних та пасажирських потоків. Наведено шляхи покращання показників роботи маршрутної мережі громадського транспорту.

Ключові слова: транспортний район, рухомість населення, пасажирський потік, транспортні дослідження.

Вступ. Сфера громадського пасажирського транспорту відіграє важливу роль у забезпеченні життєдіяльності сучасних міст, особливо великих та значних. Громадський транспорт забезпечує задоволення потреб у щоденному пересуванні містом великої частини його мешканців. Якісне виконання гро-

мадським транспортом послуг впливає на підвищення економічної ефективності виробництва, зростання продуктивності праці, сприяє підвищенню культурного рівня населення, а також раціональному використанню вільного часу [1]. Сучасні стандарти життя зумовлюють зростання потреб населення в

© Р. Б. Рогальський. 2015