Ya. A. Zack, Dokt.-Ing. Deutschland, Aachen, Deutschland, e-mail: yuriy_zack@hotmail.com

Algorithms for Solving the Transport Problem in Fuzzy Data on the Cost of Delivery of Goods

We consider a mathematical model of the transportation problem of linear programming in an environment where the specific transport costs are represented by fuzzy sets, and volume of cargo shipments — are real numbers. On the basis of the proposed methods of comparison and benchmarking fuzzy-sets general type developed fuzzy analogues of potential method. For Fuzzy-sets with the membership function of the triangular and trapezoidal species we present simple formulas and detailed calculation algorithms, which are illustrated in the numerical example.

Keywords: *transportation problem, the basic feasible solutions, the potential method, Fuzzy-set comparison and determine the effectiveness of fuzzy sets*

References

1. **Golshtein E. G., Yudin D. B.** *Zadachi linejnogo programmirovanija transportnogo vida*, Moscow, Fismatgiz, Nauka, 1993, 384 p. (in Russian).
2. **Kantorovitch L. V.** O permeschenii mass. *Dokladi AN SSSR*, 1942, vol. 37. P. 227—229. (in Russian).
3. **Lungu K. N.** *Linejnoje programmirovanije. Rukovodstvo k resheniju zadach*. Moscow: Fismatlit, 2005. 128 p. (in Russian).
4. **Danzing Dzh.** Linejnoje programmirovanije, jego primenenija i obobschenija, Moscow: Progress, 1966, 600 p. (in Russian).
5. **Zack Yu. A.** *Prinyatije reshenij v uslovijach razmitich i nechetkich dannich. Fuzzy-technologii*, Moscow, Librokom, 2013, 352 p.
6. **Zack Yu. A.** Kriterii i metodi sravnenija nechetkich mnozhestv, *Sistemnije issledovanija i informazionnije technologii*, Kiew, 2013, no. 3. pp. 58—68.
7. **Rommelfanger H.-J.** *Entscheiden bei Unschurfe. Fuzzy Decission. Support-Systeme*, Springer Verlag, Berlin-Heidelberg, 1994, 314 p.
8. **Duboi D., Prade H. M.** *Fuzzy sets and systems: theory and applications*, Academic Press, New Jork — London — Toronto, 1980, 393 p.
9. **Chang Y.-H. O.** Hybrid fuzzy least-squares regression analysis and its reliability measures, *Fuzzy Sets and Systems*, 2001, vol. 119, pp. 225—246.
10. **Poleshuk O. M., Komarov E. G.** New defuzzification method based on weighted intervals, *Proceedings of the 27th International Conference of the North American Fuzzy Information Processing Society, NAFIPS'2008.*
11. **Poleshchuk O., Komarov E.** A Fuzzy Nonlinear Regression Model for Interval Type-2 Fuzzy Sets, *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 2014, vol. 8, no. 6, pp. 833—837.
12. **Tanaka H., Uejima S., Asai K.** Linear regression analysis with fuzzy model, *IEEE Transactions on Systems, Man and Cybernetics*, 1982, vol. 12, no. 6, pp. 903—907.
13. **Zack Yu. A.** Fuzzy-regressionnije modeli prognosirovsnija zatrat vremeni i stoimosti grusovich avtomobilnich perevosok, *Logistika segodnya*, 2015, no. 3, pp. 162—172.

УДК 004.91

С. В. Бутаков, канд. техн. наук, доц., e-mail: sergey.butakov@computer.org,
С. В. Мурзинцев, аспирант, e-mail: o.l00@yandex.ru,
А. А. Цхай, д-р техн. наук, проф., e-mail: taa1956@mail.ru,
Алтайская академия экономики и права, Барнаул

Использование горизонтально масштабируемой инфраструктуры при поиске заимствований в тексте

Рассмотрена проблема быстрого сравнения текстовых документов. В прикладных задачах акцент сделан на поиск плагиата и на фильтрацию текстов в системах защиты от утечек информации. Краткий обзор решений, основанных на традиционных СУБД, показал их ограничения с точки зрения масштабируемости системы. В качестве альтернативы предложено использовать нереляционные СУБД с возможностью распределения поискового индекса между узлами системы. Для решения задач текстового поиска в работе предложен вариант представления отпечатков текстов в виде "ключ — значение", выполнена программная реализация данной модели и проведены эксперименты, подтвердившие приемлемость модели с точки зрения реализации на горизонтально масштабируемой платформе.

Ключевые слова: *фильтрация текста, сравнение текстов, большие данные, обнаружение плагиата, системы предотвращения потери данных*

Введение

Задача поиска совпадающих или близких к совпадению фрагментов текстов изучается на протяжении нескольких последних десятилетий. В большом потоке появившихся исследований можно выделить два основных направления. В работах первого направления совпадение или близость к совпадению фрагментов текстов понимается как совпадение или близость смыслов фрагментов [1, 2].

Решение этой задачи в различных вариантах привело к созданию семейства систем семантического поиска [3, 4] и, как следствие, к улучшению поиска информации в сети Интернет.

Работы, относящиеся ко второму направлению, ориентированы на выявление совпадения или близости фрагментов текста на поверхностном, т. е. символьном, уровне. Такая постановка задачи актуальна, в частности, при поиске плагиата в текстах [5, 6] или при фильтрации конфиденциальной информации в системах защиты от утечек данных [7].

Большинство пользователей глобальной сети Интернет ежедневно используют поисковые машины, но, как правило, текст, сравниваемый с массивом документов, имеет относительно ограниченную длину — максимум несколько десятков слов. Например, поисковый сервис Google ограничивает запрос 32 словами. Данная работа, относящаяся ко второму из указанных направлений, рассматривает задачи сравнения текстов произвольной длины.

Теоретически постановка задачи выглядит следующим образом: сравнить проверяемый текст t' произвольной длины с текстами t в массиве T, где размер массива T может быть достаточно большим, например, несколько миллионов документов. В определенных случаях такое сравнение может быть передано третьей стороне, обладающей высокопроизводительной компьютерной инфраструктурой для осуществления подобного поиска [8]. Однако следует заметить, что в ряде случаев задача поиска не может быть передана третьей стороне, так как сравниваемые тексты могут иметь конфиденциальный характер. В этом случае возникает проблема создания алгоритмов, позволяющих осуществлять поиск на горизонтально масштабируемой платформе, мощность которой может расти, следуя за потребностями организации. Подобная поисковая система должна обладать способностью к эластичному росту, не требующему существенных инвестиций в инфраструктуру, изменения алгоритмов и программного обеспечения.

Одним из возможных решений данной задачи является разработка структур данных и архитектуры поискового сервиса с использованием нереляционных баз данных, работающих на распределенной инфраструктуре [9].

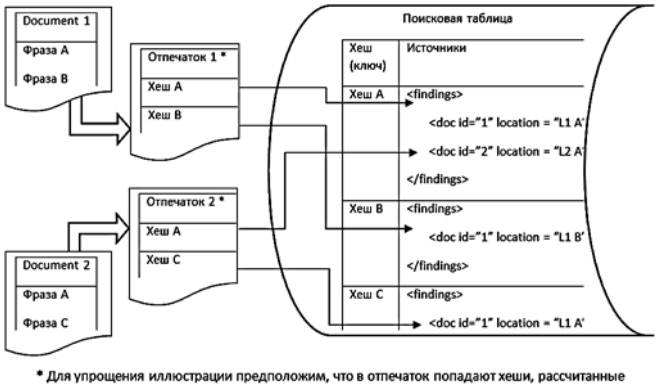
В данной статье рассмотрено подобное решение и показаны его преимущества и возможные ограничения на использование. Структурно работа построена следующим образом: в первом разделе дан краткий обзор существующих методов сравнения текстов с использованием традиционных баз данных [10, 11]. Во втором и третьем разделах рассмотрены представление данных для быстрого поиска и соответствующие эксперименты на больших текстовых выборках.

Модели хранения текстовых данных для быстрого сравнения

Большой класс методов быстрого текстового поиска по документам представлен методами, основанными на n-граммах (или шинглах). Данные методы предполагают создание отпечатка документа, который бы позволял быстро идентифицировать совпадающие части при попарном сравнении документов. За последние 30 лет разработаны десятки алгоритмов, основанных на шинглах. Подробный обзор данных алгоритмов с точки зрения производительности выполнен в работе [7]. Одним из известных алгоритмов, основанных на шинглах, является алгоритм Winnowing [12]. Пример применения Winnowing-подобного алгоритма для решения задачи сравнения текстов в системе подробнее рассмотрен в работе [13]. Показано, что выбранный шингл (n-грамма) помещается в таблицу базы данных в виде триады "<хеш> — <документ> — <позиция хеша в документе>" и представляется записями как минимум в двух реляционно-связанных таблицах. Примеры, приведенные в работах [13, 14], иллюстрируют высокую скорость роста базы данных шинглов при росте числа сравниваемых документов и, как следствие, скоростные ограничения, накладываемые реляционной моделью.

В качестве решения проблемы данная работа предлагает нереляционную модель типа "*ключ — значение*", где хеш подстроки (n-граммы) документа будет выступать в качестве ключа, а значение будет представлять собой двумерную таблицу, содержащую значения *"<документ> — <позиция хеша в документе>"* для всех документов, имеющих совпадающий текст. Таким образом, для поиска будет использоваться только одно ключевое значение, представленное значением хеш функции от n-граммы. Следствием этого является возможность распределения поиска по узлам кластера (шардам базы данных) для выполнения параллельных запросов к нескольким узлам одновременно [13].

Пример записи двух документов с совпадающим фрагментом представлен на рис. 1 [13]. В данном



* Для упрощения иллюстрации предположим, что в отпечаток попадают хеши, рассчитанные для каждой фразы

Рис. 1. Пример хранения данных в поисковой таблице [13]

примере в поисковую таблицу помещено два отпечатка документов с двумя совпадающими фрагментами (*n*-граммами).

Представленная теоретическая модель была реализована программно и протестирована в ряде экспериментов. Результаты тестирования рассмотрены ниже в следующем разделе.

Экспериментальные результаты

Эксперименты, выполненные в данной работе, направлены на подтверждение применимости использования распределенных баз данных для поиска схожих документов. Применимость оценивалась через два параметра — возможность масштабирования инфраструктуры и скорость работы системы. Скорость работы системы оценивалась во время загрузки базы сравнения в систему и во время сравнения одного документа с массивом текстов, загруженных в базу данных.

Возможность масштабирования инфраструктуры оценивалась через построение прототипа программной системы с использованием различных конфигураций горизонтально масштабированной базы данных. Были построены и протестированы инфраструктуры без разделения записей по индексу, а также с разделением записи по индексу между тремя и шестью шардами. При тестировании программы на всех трех инфраструктурах не требовалось изменять код приложения, что дает основание утверждать, что прототип системы может работать при разделении записей между фактически неограниченным числом узлов. Данное свойство обеспечивает возможность эластичного масштабирования системы при росте базы данных.

Схема кластера с тремя узлами хранения данных (шардами) показана на рис. 2 (см. третью сторону обложки). Кластер включает сервер приложения, работающий с сервером-маршрутизатором, который, в свою очередь, работает с конфигурационными серверами и с узлами хранения (шардами). В качестве физического оборудования для всех компонентов системы в экспериментах использовались неспециализированные персональные компьютеры офисного назначения.

В качестве базы данных сравнения была использована очищенная выборка статей из англоязычного сегмента Википедии (en.wikipedia.org). Документы в выборке были очищены от специфической разметки (тэгов) системы управления контентом MediaWiki (www.mediawiki.org). Исходя из предположения что документы, проверяемые на плагиат или на конфиденциальность, должны нести некую смысловую нагрузку, из полученного набора документов были удалены небольшие документы размером менее 100 слов. Полученная после подобной очистки выборка состояла из 3 917 453 документов, общий размер которых равен 9,84 Гбайт.

Оценка времени загрузки. Первый эксперимент имел целью оценить время загрузки исходного массива в базу данных. Данная операция не является критической для поисковой системы. В реальных условиях массовая загрузка большого числа документов выполняется при вводе системы в эксплуатацию, и в дальнейшем проводится рутинное обновление загруженного массива.

Среднее время загрузки (время/размер, мс/Кбайт) текста для различных конфигураций хранилища составляет: без шард — 87 мс/Кбайт, 3 шарда — 21 мс/Кбайт, 6 шардов — 11 мс/Кбайт. Зависимость времени загрузки от размера документа показана на графиках, приведенных на рис. 3—5.

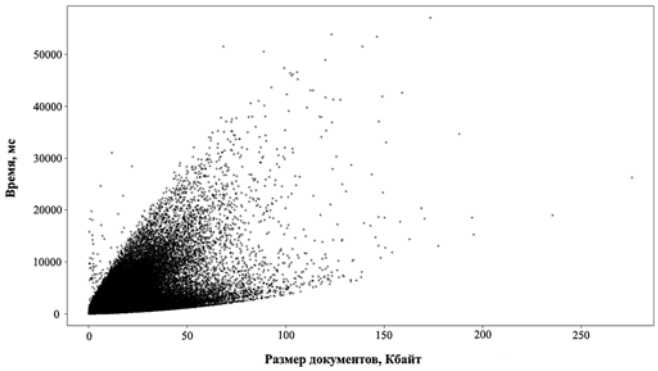


Рис. 3. Время записи в зависимости от размера документа (без шард)

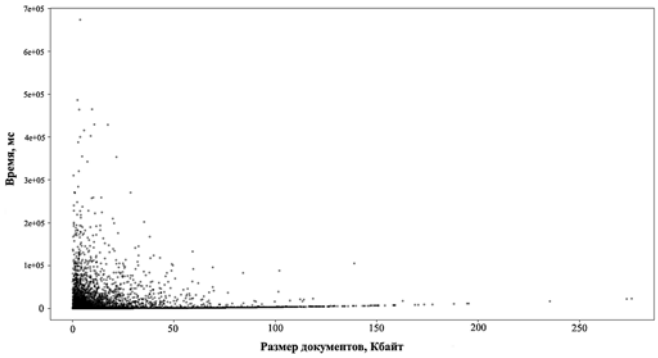


Рис. 4. Время записи в зависимости от размера документа (3 шарда)

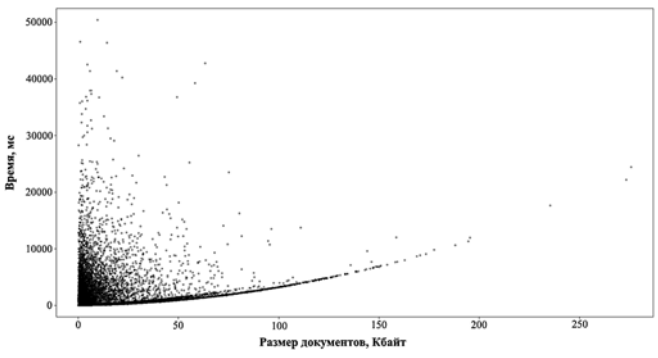


Рис. 5. Время записи в зависимости от размера документа (6 шард)

Однако, как показал анализ, данные графики не учитывают объем данных, уже загруженных в базу данных. На графиках явно видно, что время загрузки зависит не только от размера документа, так как значение времени значительно варьируется при загрузке документов приблизительно одного размера. Особенно хорошо это просматривается на графике, который находится на рис. 3.

С учетом того, что при загрузке проводится обновление индексов, размер базы данных является ключевым фактором, влияющим на производительность операции добавления. Графики, приведенные на рис. 6—8, показывают относительное время записи документов в зависимости от числа документов, уже загруженных в базу данных.

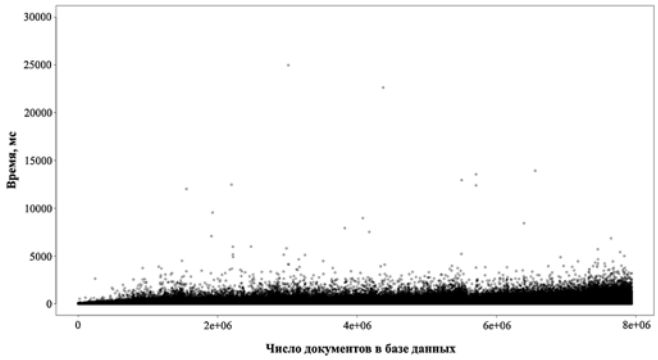


Рис. 6. Время поиска в зависимости от размера базы данных (без шард)

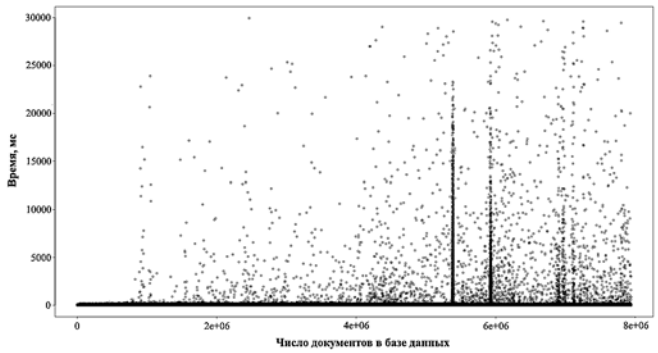


Рис. 7. Время поиска в зависимости от размера базы данных (3 шарда)

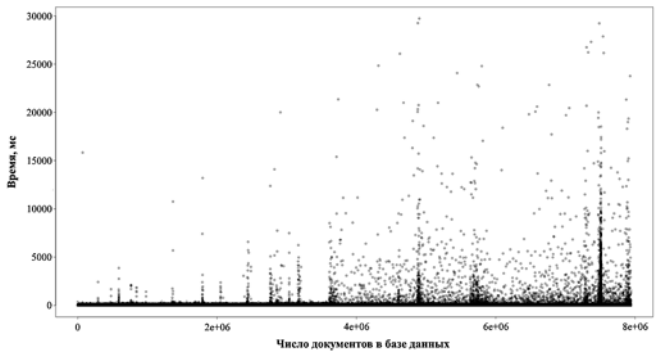


Рис. 8. Время поиска в зависимости от размера базы данных (6 шард)

Как видно из данных графиков, рост относительного времени, требуемого на загрузку в зависимости от объема уже загруженных данных, близок к линейному. При увеличении числа узлов хранения с одного до трех и с трех до шести заметно, что относительное время загрузки показывает меньший рост при увеличении числа узлов хранения. Это позволяет утверждать, что с точки зрения загрузки система является горизонтально масштабируемой.

На графиках, изображенных на рис. 7 и 8, отчетливо видно, что в момент перераспределения индекса по шардам время загрузки уменьшается. В случае с 3 шардами перераспределение происходит реже, но занимает больше времени по сравнению с работой кластера из 6 шард.

Оценка времени поиска. Второй эксперимент ставил целью оценку времени поиска по базе данных. Для проведения оценки массив из 3 917 453 документов был загружен в базу данных, установленную на загруженные ранее архитектуры. Для осуществления поиска из исходного массива были случайным образом отобраны 1000 документов и выполнен их поиск. Результатом каждого поиска было 100 % совпадения набора хешей документа, но эксперимент оценивал не качество, а время поиска. Графики времени поиска в зависимости от размера проверяемого документа приведены на рис. 9—11. Ниже представлено среднее время поиска на килобайт текста (время/размер, мс/Кбайт): без шард — 159,1 мс/Кбайт, 3 шарда — 77,7 мс/Кбайт, 6 шардов — 78,6 мс/Кбайт. Из этого можно сделать вывод, что среднее время без использования распределенной инфраструктуры как минимум в два раза выше, чем при хранении данных в распределенных узлах. Кроме того, график на рис. 9 показывает значительную нелинейность при росте объема проверяемого документа. В то же время графики на рис. 10 и 11 хорошо приближаются линейным представлением. Кроме того, предварительный анализ последних графиков показывает, что заметного роста времени поиска с увеличением числа шард не происходит, а даже наблюдается ее некоторое снижение при использовании 6 узлов вместо 3.

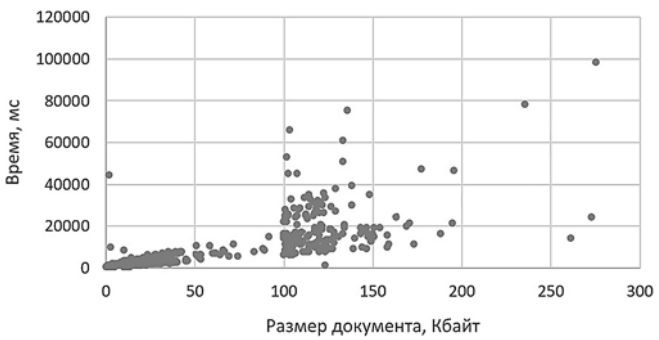


Рис. 9. Время поиска в зависимости от размера документа (без шард)

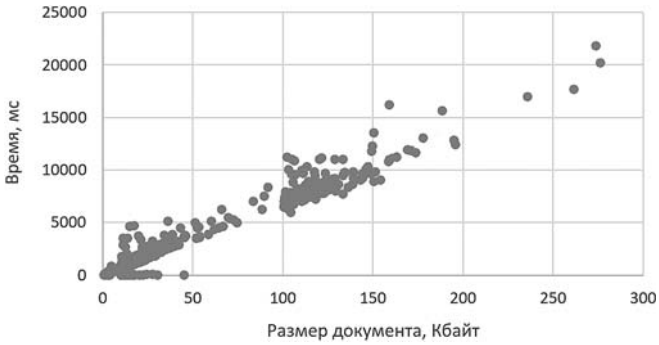


Рис. 10. Время поиска в зависимости от размера документа (3 шарда)

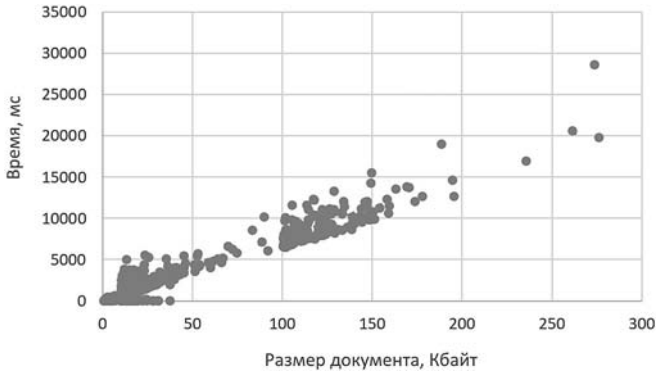


Рис. 11. Время поиска в зависимости от размера документа (6 шард)

Данный факт требует дополнительных исследований и может быть объяснен тем, что для данного объема проверяемых документов (~4 млн текстов) 6 узлов хранения значений отпечатка являются избыточными.

Практические выводы по результатам эксперимента и заключение по результатам работы представлены в следующем разделе.

Заключение

Количественная оценка времени поиска говорит о ее приемлемости для практических приложений, связанных с поиском плагиата или задачами защиты данных от утечек. Время 80 мс/Кбайт проверяемого текста может быть транслировано в односекундную задержку для текста длиной 6 страниц, что является приемлемым уровнем задержки для подобных систем. Если в качестве примера взять типичную задержку в 24 ч для систем проверки на плагиат, то можно утверждать, что пропускная способность подобной системы будет немного ниже 85 000 документов из 6 страниц в сутки, что является приемлемым для систем большого размера, например, обслуживающих один или несколько высших учебных заведений.

Следует отметить, что в данном случае система проводит лишь предварительный отбор документов из большого массива. Алгоритмы, основанные на шинглах, не являются наилучшими с точки зрения детального сравнения документов. Следовательно, предварительный отбор должен быть дополнен детальным сравнением проверяемого документа и предварительно выбранных документов. С учетом того, что объем сравнения снизится со всей выборки в несколько миллионов документов до набора документов, которые гарантированно включают совпадающие фрагменты, детальное сравнение не должно вызвать значительных задержек во времени. Кроме того, процессы предварительного отбора и детального сравнения будут выполняться на различных подсистемах в параллельном режиме, что снизит их взаимозависимость.

Можно указать два направления дальнейших исследований: определение момента необходимости добавления узла при росте числа документов в базе данных и проведение эксперимента на большем числе узлов. Решение первой задачи возможно путем разработки модели оптимизации стоимости инфраструктуры при сохранении времени поиска. Эксперимент позволит верифицировать полученную модель на реальных данных.

Исследование выполнено при финансовой поддержке гранта совместной научной программы "РФФИ-Алтайский край", проект № 14-07-98000.

Список литературы

1. **Fomichov V. A.** K-calculuses and K-languages as power formal means to design intelligent systems processing medical texts // Cybernetica (Belgium). 1993. Vol. 36, N. 2. P. 161—182.

2. **Fomichov V. A.** Integral formal semantics and the design of legal full-text databases // Cybernetica (Belgium). 1994. Vol. 37, N. 2. P. 145—177.

3. **Lei Y., Urea V., Motta E.** SemSearch: A Search Engine for the Semantic Web / Staab, Steffen, Svatek, Vojtech (Eds.) // Proc. 15th International Conference on Knowledge Engineering and Knowledge Management "Managing Knowledge in a World of Networks", Podebrady, Czech Republic, October 6—10, 2006, Proceedings. Lect. Notes in Comp. Sci., Springer, 2006. P. 238—245.

4. **Fomichov V. A., Kirillov A. V.** A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures // Artificial Intelligence: Methodology, Systems, and Applications, 15th International Conference, AIMSA 2012. Varna, Bulgaria, September 2012, Proceedings. Springer, Lecture Notes in Artificial Intelligence. 2012. Vol. LNAI 7557. P. 296—304.

5. **HaCohen-Kerner Y., Tayeb A.** Experiments with Filtered Detection of Similar Academic Papers // Artificial Intelligence: Methodology, Systems, and Applications, 15th International Conference, AIMSA 2012. Varna, Bulgaria, September 2012, Proceedings. Springer, Lecture Notes in Artificial Intelligence. 2012. Vol. LNAI 7557. P. 1—13.

6. **Alzahrani S. M., Naomie S., Ajith A.** Understanding plagiarism linguistic patterns, textual features, and detection methods // Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions. 2012. Vol. 42, N. 2. P. 133—149.

7. **Faro S., Lecroq Th.** The exact online string matching problem: A review of the most recent results // ACM Computing Surveys. 2013. Vol. 45, N. 13. P. 42—50. DOI=http://dx.doi.org/10.1145/2431211.2431212

8. **Демиденко Н. Д., Кулагин В. А., Шокин Ю. И.** Моделирование и вычислительные технологии распределенных систем. Новосибирск: Наука, 2012. С. 142—148.

9. **Торган Ю. Н., Зубрилина Т. В.** Использование нереляционного подхода в распределенной системе баз данных // Науч-

но-технические ведомости Санкт-Петербургского государственного политехнического университета. 2012. Т. 5. № 157. С. 15—20.

10. **Ganascia J. G., Lungo A. D.** Automatic detection of reuses and citations in literary texts // Lit Linguist Computing. 2014. Vol. 29, N. 3. P. 412—421.

11. **Pohuba D., Dulik T., Janku P.** Automatic evaluation of correctness and originality of source codes // In 10th European Workshop on Microelectronics Education (EWME). Tallinn, 2014. P. 49—52.

12. **Schleimer S., Wilkerson D., Aiken A.** Winnowing: Local Algorithms for Document Fingerprinting // Proc. of the ACM SIGMOD International Conference on Management of Data. San Diego. 2003. P. 76—85.

13. **Цхай А. А., Бутаков С. В., Мурзинцев С. В., Ким Л. С.** Обнаружение плагиата с использованием нереляционных баз данных // Вестник алтайской науки. 2015. № 1. С. 280—285.

14. **Дягилев В. В., Цхай А. А., Бутаков С. В.** Архитектура сервиса определения плагиата, исключающая возможность нарушения авторских прав // Вестник Новосибирского государственного университета. Сер. "Информационные технологии". 2011. Т. 9. № 3. С. 23—29.

S. V. Butakov, Associate Professor, e-mail: sergey.butakov@computer.org,
S. V. Murzintsev, Postgraduate student, e-mail: o.l00@yandex.ru,
A. A. Tskhai, Full Professor, e-mail: taa1956@mail.ru,
Altai Academy of Economics and Law, Barnaul, Russia

Detecting Text Similarity on a Scalable Cluster

The paper addresses the problem of fast text comparison in massive datasets. Specific application areas in the project included plagiarism detection and text filtering in data loss protection systems. Survey of existing solutions based on relational databases in these areas outlined scalability limitations that may affect the performance. Solution proposed in this paper suggests using non-relational no-SQL databases with potential distribution of the search workload between nodes on the database cluster. To facilitate text search on distributed cluster, it was suggested to use "key-value"-like data structure for text representation. The work outlines details of the proposed data structure, describes developed software prototype and performed experiments. The latter confirmed applicability of the proposed solution on the distributed infrastructure with three and six nodes in terms of comparison quality and speed.

Keywords: *text filtering, text comparison, big data, plagiarism detection, no-SQL, data loss prevention systems*

References

1. **Fomichov V. A.** K-calculuses and K-languages as power formal means to design intelligent systems processing medical texts http://cat.inist.fr/?aModele=afficheN&cpsidt=4211600, *Cybernetica* (Belgium), 1993, vol. 36, no. 2, pp. 161—182.

2. **Fomichov V. A.** Integral formal semantics and the design of legal full-text databases, *Cybernetica* (Belgium), 1994, vol. 37, no. 2, pp. 145—177.

3. **Lei Y., Urea V., Motta E.** SemSearch: A Search Engine for the Semantic Web. Staab, Steffen, Svatek, Vojtech (Eds.), *Proc. 15th International Conference on Knowledge Engineering and Knowledge Management "Managing Knowledge in a World of Networks"*, Podebrady, Czech Republic, October 6—10, 2006, Proceedings. Lect. Notes in Comp. Sci., Springer, 2006, pp. 238—245.

4. **Fomichov V. A., Kirillov A. V.** A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures, *Artificial Intelligence: Methodology, Systems, and Applications, 15th International Conference, AIMSA 2012.* Varna, Bulgaria, September 2012, Proceedings. Springer, Lecture Notes in Artificial Intelligence, 2012, vol. LNAI 7557, pp. 296—304.

5. **HaCohen-Kerner Y., Tayeb A.** Experiments with Filtered Detection of Similar Academic Papers, *Artificial Intelligence: Methodology, Systems, and Applications, 15th International Conference, AIMSA 2012.* Varna, Bulgaria, September 2012, Proceedings. Springer, Lecture Notes in Artificial Intelligence, 2012, vol. LNAI 7557, pp. 1—13.

6. **Alzahrani S. M., Naomie S., Ajith A.** Understanding plagiarism linguistic patterns, textual features, and detection methods, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, 2012, vol. 42, no. 2, pp. 133—149.

7. **Faro S., Lecroq Th.** The exact online string matching problem: A review of the most recent results, *ACM Computing Surveys*, 2013, vol. 45, no. 13, pp. 42—50, DOI=http://dx.doi.org/10.1145/2431211.2431212

8. **Demidenko N. D., Kulagin V. A., Shokin Ju. I.** *Modelirovanie i vychislitel'nye tehnologii raspredelennyh system: monografija* [Modeling and computational technology distributed systems]. Novosibirsk: 2012, pp. 142—148.

9. **Torgan Ju. N., Zubrilina T. V.** Ispol'zovanie nereljacionnogo podhoda v raspredelennoj sisteme baz dannyh [Using the relational approach in a distributed database system], *Nauchno-tehnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politehnicheskogo universiteta*, 2012, vol. 5, no. 157, pp. 15—20.

10. **Ganascia J. G., Lungo A. D.** Automatic detection of reuses and citations in literary texts, *Lit Linguist Computing*, 2014, vol. 29, no. 3, pp. 412—421.

11. **Pohuba D., Dulik T., Janku P.** Automatic evaluation of correctness and originality of source codes, *In 10th European Workshop on Microelectronics Education (EWME)*, Tallinn, 2014, pp. 49—52.

12. **Schleimer S., Wilkerson D., Aiken A.** Winnowing: Local Algorithms for Document Fingerprinting, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, San Diego, 2003, pp. 76—85.

13. **Tskhai A. A., Butakov S. V., Murzincev S. V., Kim L. S.** Obnaruzhenie plagiata s ispol'zovaniem nereljacionnyh baz dannyh [Plagiarism detection using non-relational databases], *Vestnik altajskoj nauki*, 2015, no. 1, pp. 280—285.

14. **Djagilev V. V., Tskhai A. A., Butakov S. V.** Arhitektura servisa opredelenija plagiata, iskljuchajushhaja vozmozhnost' narushenija avtorskih prav [The architecture of the service definition of plagiarism, which excludes the possibility of copyright infringement], *Vestnik Novosibirskogo gosudarstvennogo universiteta, serija "Informacionnye tehnologii",* 2011, vol. 9, no. 3, pp. 23—29.