CLB-деревья: новый способ индексации больших массивов текстов

А. Б. Веретенников, Ю. С. Лукач

В последние годы резко повысилась актуальность обработки больших объемов разнородной текстовой информации. В первую очередь это связано с лавинообразным развитием Интернета, которое привело к появлению громадного количества текстов, представленных самым различным образом: в виде обычных текстов, HTML- и XML-документов, сообщений электронной почты и пр. В частности, юридическая, патентная и новостная информация в Интернете исчисляется уже терабайтами. В этой ситуации существенно возрос интерес к классификации подобных массивов документов и быстрому поиску в них нужной информации. При этом мы понимаем под поиском слов в массиве текстов нахождение всех документов, содержащих искомые слова, и положение найденных слов в этих документах.

Обычно обеспечение быстрого поиска в подобных массивах является комбинацией методов внешнего и внутреннего поиска. Структуры данных для каждого из этих видов поиска изучены достаточно хорошо, однако их эффективные комбинации начали рассматриваться в литературе только в последние годы. Хорошо известно, что для хранения и индексации данных во внешней памяти обычно используются либо инвертированные файлы [1], либо B-деревья и их вариации [2, 3]. С другой стороны, для быстрого поиска во внутренней памяти чаще всего применяются цифровые деревья поиска типа Patricia [4], суффиксные деревья [5, 6], суффиксные массивы [7, 8] и тернарные деревья поиска [9]. Представляется логичным строить новые комбинированные структуры на основе уже апробированных структур. Первой из известных нам попыток создания такой комбинированной структуры были String B-деревья — комбинация B-дерева и Patricia [10, 11].

Авторы разработали новую структуру — CLB-дерево (от cluster list + B-дерево), представляющую собой комбинацию B-деревьев и тернарных деревьев с хранением информации о словах в цепочках связанных блоков кластеров. CLB-дерево предназначается как для поиска отдельных слов, так и для поиска словосочетаний в текстах. При разработке данной структуры мы руководствовались требованиями достижения наибольшего быстродействия как при поиске слов, так и при индексации текста.

Даны теоретические оценки, подтвержденные результатами вычислительных экспериментов и показывающие, что как скорость индексации, так и скорость поиска слов в CLB-деревьях существенно выше, чем у String B-деревьев, не говоря уже о классических B-деревьях.

Литература

[1] *Prywes N. S., Gray H. J.* The organization of a multilist-type associative memory // IEEE Trans. on Communication and Electronics. 1963. Vol. 68. P. 488–492.

[2] *Bayer R., McCreight E.* Organization and maintenance of large ordered indexes // Acta Informatica. 1972. Vol. 1, No. 3. P. 173–189.

[3] *Bayer R., Unterauer K.* Prefix B-trees // ACM Trans. Database Syst. 1977. Vol. 2, No. 1. P. 11–26.

[4] *Morrison D. R.* PATRICIA: Practical algorithm to retrieve information coded in alphanumeric // J. ACM. 1968. Vol. 15. P. 514–534.

[5] *McCreight E. M.* A space-economical suffix tree construction algorithm // J. ACM. 1976. Vol. 23, No. 2. P. 262–272.

[6] *Weiner P.* Linear pattern matching algorithm // IEEE Symp. on Switching and Automata Theory. 1973. P. 1–11.

[7] *Gonnet G. H., Baeza-Yates R. A., Snider T.* Information Retrieval: Data Structures and Algorithms. N. Y.: Prentice-Hall, 1992.

[8] *Manber U., Myers G.* Suffix arrays: a new method for on-line string searches // SIAM J. Comput. 1993. Vol. 22, No. 5. P. 935–948.

[9] *Bentley J. N., Sedgewick R.* Fast algorithms for sorting and searching strings // 8th ACM-SIAM Symp. on Discr. Algorithms. 1997. P. 360–369.

[10] *Ferragina P., Grossi R.* The string B-tree: a new data structure for string search in external memory and its applications // J. ACM. 1999. Vol. 46, No. 2. P. 236–280.

[11] *Ferragina P., Grossi R.* An experimental study of SB-trees // 7th ACM-SIAM Symp. on Discr. Algorithms. 1996. P. 373–397.