BIT 29 (1989), 364–368

AN ADAPTATION OF A ROOT FINDING METHOD TO SEARCHING ORDERED DISK FILES REVISITED

Y. MANOLOPOULOS and G. POULAKAS

Division of Electronics and Computer Engineering, Department of Electrical Engineering, Aristotelian University of Thessaloniki, 54006 Thessaloniki, Greece.

Abstract.

In the present report, Interpolation search, Fast search and Pegasus method are compared with respect to their performance in searching ordered disk files for several key distributions. The aim is to study the effect of the page capacity on searching performance. Cost metric is the number of page accesses and not key comparisons. Numerical results are illustrated and a new approximate formula is derived giving an estimate of the number of page accesses for the case of the Interpolation algorithm under uniform distributions.

CR categories and subject descriptors: D.4.2, D.4.8, F.2.2, H.2.2.

General terms: Algorithms, Experimentation, Performance.

Additional key words and phrases: Interpolation search, Fast search, Pegasus method, root location methods, ordered and unindexed files, page capacity, searching performance.

1. The simulation.

In a recent work by Armenakis et al. [1] a comparison between Interpolation [4, 8], Fast [2, 6] and Pegasus methods [3] for three different key value statistical distributions is performed. The major limitation of this work comes from the fact that, although the term file has been included in the title, there is an implicit assumption that the cost metric is the number of key comparisons and not the number of page accesses. In other words, their results hold only for the case when the page capacity is one record, which is not often met in practice. The motivation of the present report is the reexamination of the three algorithms, the validation of the previous results and the study of the effect of the page capacity on the searching performance. For our experimentation and analysis, the cost is considered in I/0 terms because the CPU computation is relatively negligible. Therefore, the metric is the number of page accesses.

An extensive simulation has been carried out on IBM AT with Turbo Pascal

Received May 1988. Revised November 1988 and February 1989.

version 4.0. It has been tried to follow the main characteristics of the experimentation of [1], e.g. the ordered disk files consisted of 5000, 15000 and 25000 distinct records with key values obeying a uniform, an exponential and a gamma distribution. However, for our experiment the key space is from 1 to 500000, and not from 0 to 1 as in [1]. In addition, it is noted here that the exponential distribution is taken by using a logarithmic method, while the gamma distribution is taken by using a rejection-acceptance method by setting the shape (scale) parameter equal to 2 (1) [5]. As explained earlier, a new parameter has been considered, that of the page capacity. In this simulation the capacity was equal to 40, 20, 10, 5 and 1 record. Only successful search was considered. Besides it is assumed that all records are equiprobably searched. This was achieved by generating five times a sample of 100 randomly chosen numbers from 1 to N, the file size. Afterwards the key value of every random file position was searched. This process was repeated for every searching algorithm, key distribution and file size.

Tables 1 to 3 present results of every distribution taken in consideration. The implementation details produced some deviations from the results of [1]. However, these details have not altered the qualitative results of both reports. From the tables the following six observations are made. The last two of them concern the effect of page capacity on searching performance as a function of the method and distribution involved.

Table 1. *Performance of Interpolation, Fast and Pegasus methods for various capacities and uniform key distribution*

capacity	Interpolation			Fast			Pegasus		
	5000	15000	25000	5000	15000	25000	5000	15000	25000
40	1.65	1.83	1.85	7.64	8.72	9.65	1.67	1.81	1.85
20	1.82	2.08	2.12	9.35	10.29	11.92	1.85	2.04	2.11
10	2.13	2.44	2.47	11.11	11.77	13.97	2.13	2.36	2.46
5	2.55	2.81	2.86	12.37	12.87	15.37	2.50	2.74	2.87
1	3.60	3.89	3.98	14.01	14.58	17.23	3.51	3.78	3.99

Table 2. *Performance of Interpolation, Fast and Pegasus methods for various capacities and gamma key distribution.*

capacity	Interpolation			Fast			Pegasus		
	5000	15000	25000	5000	15000	25000	5000	15000	25000
40	7.17	8.89	10.71	6.75	9.00	10.74	4.57	4.88	5.07
20	8.57	10.21	12.16	8.08	10.31	12.46	4.88	5.19	5.39
10	9.95	11.53	13.60	9.29	11.61	14.04	5.22	5.52	5.72
5	11.33	12.82	15.34	10.28	12.88	15.04	5.59	5.88	6.09
1	14.04	15.34	17.78	11.97	14.60	17.22	6.66	6.97	7.24

Table 3. *Performance of Interpolation, Fast and Pegasus methods for various capacities and exponential key distribution.*

capacity	Interpolation			Fast			Pegasus		
	5000	15000	25000	5000	15000	25000	5000	15000	25000
40	9.97	12.88	23.40	7.78	9.91	14.97	5.15	5.52	6.53
20	12.09	14.92	27.23	9.70	11.87	18.81	5.52	5.83	6.84
10	14.19	16.97	31.16	12.01	13.52	20.93	5.89	6.13	7.14
5	16.25	18.89	34.64	13.42	14.74	22.26	6.29	6.51	7.51
1	20.32	22.80	41.88	15.26	16.56	24.16	7.42	7.59	8.55

(a) The performance of all methods is reconfirmed to be of order log log N. Even for the exponential and the gamma distributions the involved coefficient does not take great values.

(b) The interpolation method is very stable for the case of uniform distribution. As expected, for the gamma and especially for the exponential distribution the performance deteriorates with increasing file sizes. Results of the gamma distribution always lie between the results of the other ones.

(c) Fast search is the worst method for uniform keys. However, its performance is close (superior) to that of Interpolation for the gamma (exponential) distribution. In this case results of the gamma distribution does not lie between the results of the other ones. On the contrary, our experiments seem to indicate that no certain position may be taken concerning the performance of the method under the uniform and gamma distributions. Exponential distribution always gives higher results.

(d) The Pegasus method emerged as the most stable algorithm. Its performance is almost identical to that of Interpolation for uniform keys but clearly outperforms it for gamma and especially the exponential distribution. The Pegasus method compared to the Fast search is superior in all cases. It is remarkable that in the case of exponential data its performance results are very close even with increasing file sizes. As for Interpolation search, here again results of the gamma distribution always lie between the results of the other ones.

(e) When considering the effect of page capacity on performance for all methods, as expected, the number of page accesses decreases with increasing page capacity. Comparing the results of page capacity 1 to those of page capacity 40 we observe a gain of at least $\approx 40\%$ (up to $\approx 55\%$) for the Interpolation method. Similar figures for fast search (Pegasus method) are from $\approx 37\%$ to $\approx 45\%$ ($\approx 23\%$ to $\approx 54\%$) gain in page accesses. This gain is a measure of the convergence speed of the algorithms. Therefore, as a consequence it seems that Interpolation search converges to the final position relatively faster than the other methods.

(f) From the key distribution point of view the following observations are made. For uniform distributions the gain varies from $\approx 55\%$ to $\approx 40\%$, while for

gamma (exponential) distributions the gain varies from $\approx 49\%$ to $\approx 30\%$ ($\approx 51\%$ to $\approx 23\%$). In an analogous manner, from our experiments, it means that the uniform distribution plays a positive role with respect to the fast convergence when compared to the other distributions.

2. Epilogue.

In this section a new formula is derived under the assumption that the keys, which obey a uniform distribution, are searched with the Interpolation method. This formula is based on an expected value analysis and gives an estimate of the number of page accesses as a function of the file size and the page capacity.

According to [8] the expected distance between the ith probe position and the final position is less than $N^{2^{-i}}$ $(i > 1)$. Besides, the average error of the first probe position is less than $(\sqrt{N})/2$. The page accesses continue up to the point that the page containing the desired record has been reached. Expectedly the desired record will be the middle one of the last page. Therefore:

$$N^{2^{-i}} \leq \text{cap}/2 \Leftrightarrow i \leq \log_2 \log_2 N - \log_2 \log_2 (\text{cap}/2)$$

where cap is the page capacity. The above equation gives an approximation of the necessary page accesses. Table 4 illustrates a comparison of the values produced by the new formula together with the results produced by our simulation. For the parameter values of Table 4 the deviation varies from $\approx 1\%$ to $\approx 26\%$. Future research should derive more accurate approximations. A possible extension of this report would deal with the study of these three, as well as other algorithms on batched searching using the methods described in [7].

Table 4. *Comparison of analysis and simulation of Interpolation search for various capacities and uniform key distribution.*

capacity	5000		15000		25000	
	simul.	analys.	simul.	analys.	simul.	analys.
40	1.65	1.51	1.83	1.68	1.85	1.76
20	1.82	1.89	2.08	2.06	2.12	2.14
10	2.13	2.40	2.44	2.58	2.47	2.65
5	2.55	3.22	2.81	3.39	2.86	3.47

Acknowledgements.

The authors would like to thank the anonymous referees for their helpful comments and specifically the one to whom the formula of the last section belongs.

REFERENCES

[1] Armenakis A. C., Garey L. E. and Gupta R. D.: *An adaptation of a root finding method to searching ordered disk files*, BIT, Vol. 25, pp. 562–568, 1985.

[2] Burton F. W. and Lewis N. G.: *A robust variation of interpolation search*, Information Processing Letters, Vol. 10, pp. 198–201, 1980.

[3] Dowell M. and Jarratt P.: *The Pegasus method for computing the root of an equation*, BIT, Vol. 12, pp. 503–508, 1972.

[4] Gonnet G. H., Rogers L. D. and George J. A.: *An algorithmic and complexity analysis of interpolation search*, Acta Informatica, Vol. 13, pp. 39–52, 1980.

[5] Knuth D. E.: *The Art of Computer Programming*, Vol. 2, *Seminumerical Algorithms*, 2nd edition, Addison-Wesley, Reading, MA., 1981.

[6] Lewis G. N., Boynton N. J. and Burton F. W.: *Expected complexity of fast search with uniformly distributed data*, Information Processing Letters, Vol. 13, No. 1, pp. 4–7, 1981.

[7] Manolopoulos Y., Kollias J. G. and Burton F. W.: *Batched interpolation search*, The Computer Journal, Vol. 30, No. 6, pp. 565–568, 1987.

[8] Perl Y., Itai A. and Avni H.: *Interpolation Search – a log log N search*, Communications of the ACM, Vol. 21, No. 7, pp. 550–553, 1978.