Two-way Replacement Selection*

Xavier Martinez-Palau, David Dominguez-Sal, Josep Lluis Larriba-Pey

DAMA-UPC, Departament d'Arquitectura de Computadors
Universitat Politècnica de Catalunya
{xmartine,ddomings,larri}@ac.upc.edu

Sorting is in the heart of many high performance processes, in particular those of a database management system. Since inputs are typically large and the memory available to a sort process is limited, a good out of core sorting algorithm has an important impact on the performance of the final application.

Replacement Selection (RS) is one of the most commonly used run generation strategies. The objective of RS is to sort a stream of records as they come, producing another stream of released data records called "run", which is sorted. The algorithm uses a heap: upon the arrival of a new record, RS releases the first record in the heap, and stores the new record in it. The runs generated are merged in a second phase. This second phase is faster when there is a smaller number of runs. Thus, it is important to generate long runs.

RS has played a prominent role in external sorting situations because it fulfills most of the desirable features. First, unlike other sorting methods, it is able to sort data in a streamed fashion, using one heap to perform the job. Second, it generates runs which double the size of the memory available for random data, and infinite runs for already sorted data. This reduces the number of runs, allowing the merge process to reduce its fan in, and the chances to perform multiple I/O passes during the merge phase. Finally, although it is not the fastest in-memory sorting strategy, it offers a good trade off for its value: it generates smaller I/O by creating larger runs at the cost of possibly more in-core computational effort, compared to other faster in-core methods.

However, for input data with certain characteristics, like inputs sorted inversely with respect to the desired output order, the length of the runs generated by RS is downsized, being only as large as the available memory. We solve this problem, proposing Two-way Replacement Selection, a general strategy that allows to obtain runs which are at least the size of those generated by RS and, in many cases, more than double the size of the memory available for sorting no matter the dataset, improving the good features mentioned above for RS.

In our paper, we present Two-Way Replacement Selection (2WRS), which is a generalization of RS obtained by implementing two heaps instead of the single heap used by RS, and adding two optional buffers, called input and victim buffers. The heaps of 2WRS adapt to the data characteristics, one intends to capture the growing values and the other one intends to capture the decreasing values. The strategy is to place each newly arrived record in the correct heap. Also, the heaps grow or shrink depending on the nature of the data. So, in case there are more growing than decreasing data, it grows the growing data

* This is an extended abstract of [1]

heap, and shrinks the decreasing data heap, and conversely. The appropriate management of these two heaps allows generating runs at least as large as twice the available memory in a stable way, i.e. independent from the datasets.

The input buffer of 2WRS is filled with records from the input. At each step, the algorithm outputs the top record of one of the two heaps (selected at random), and then inserts the next record into one heap. This second heap selection is done by an input heuristic that uses the contents of the input buffer to choose which heap holds the next record. Due to the restrictions on the values allowed for records in each heap, it is possible that a new recors cannot be stored in either heap. In this case the new record is stored in the victim buffer.

We prove that the runs generated by 2WRS are at least as large as the runs generated by RS and, for several structured inputs larger. This means that the run length of 2WRS is always equal or better than that of RS.

We performed a statistical analysis of 2WRS, using the analysis of variance (ANOVA) methods with five different sets of input data. A configuration of 2WRS sets the input heuristic used, which buffers are used, and the percentage of available memory dedicated to buffers. From this analysis we extract several conclusions. First, the use of a second heap is beneficial, as it allows the generation of large runs independently of the distribution of the input data. Second, having slightly smaller heaps in exchange for buffers is beneficial performance-wise. Finally, the analysis shows that 2WRS generates runs at least as large as those generated by RS, and larger with some input data, which confirms the theoretical analysis. The statistical analysis also allows finding the best input heuristic, buffer configuration and the optimal buffer size.

We also tested experimentally the time performance of 2WRS with respect to RS. The results show that 2WRS is only 10% slower than RS when the input data is randomly distributed, but with structured inputs, 2WRS achieves speedups of almost 3, thanks to the generation of longer runs. These experiments also show the usefulness of using both buffers. The experiments have been repeated for different input sizes and memory dedicated to the sorting operation. We have been able to sort datasets with strong memory limitations (6 orders of magnitude larger) three times faster than the regular RS. Furthermore, we obtain similar speedups with different scaleups of the input.

Finally, 2WRS maintains the heap and run generation architecture of RS that allows for improvements already proposed in the literature for RS, which include variable key support, read ahead strategies or hierarchical data sorting among others.

Acknowledgments The members of DAMA-UPC thank the Ministry of Science and Innovation of Spain and Generalitat de Catalunya, for grant numbers TIN2009-14560-C03-03 and SGR-1187 respectively.

References

[1] X. Martinez-Palau, D. Dominguez-Sal, J.L. Larriba-Pey. *Two-way replacement selection*, PVLDB, 3(1): 871-881 (2010)