

Figure 10. Response time of PART algorithm compared to SW.

The experiments were performed with the same 4 relations of test sets 1–3, but we varied the number of total sites available from 4 to 7. Thus, with 4 sites, PART does not perform any partitioning. The results of these experiments are summarized in figure 10. In comparison with SW, we achieved the following response time improvements: 18–33% for 4 sites, 15–42% for 5 sites, 19–52% for 6 sites, and 24–57% for 7 sites. Notice that increasing the number of sites does not guarantee a lower response time for the SW method for certain join selectivity values. The improvement of response time in PART versus PIPE_CHQ, i.e., the gains obtained by additional sites, were as follows: 3–10% for 5 sites, 13–24% for 6 sites, and 19–37% for 7 sites. It is clear that the speed-up in the response time of PART is not optimal with respect with the number of processors, but the performance of PART is more stable than that of SW.

5.3. Disk-based systems

In order to assess the effectiveness of our algorithm for disk-based systems where graph partitioning needs to be performed, we repeated the experiments based on test sets 1–3

23

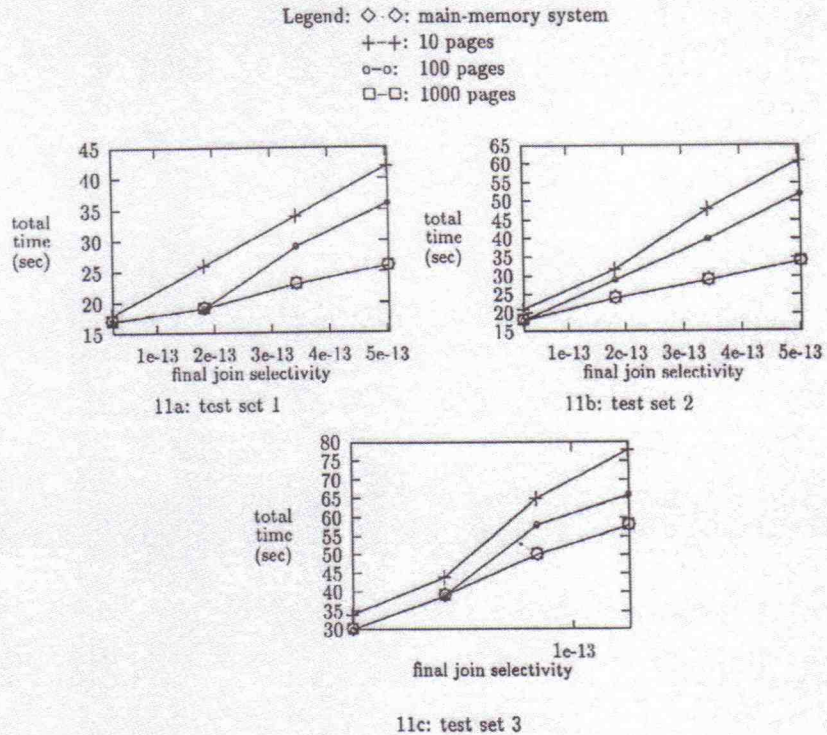


Figure 11. Total time comparison: main-memory vs. disk-based PIPE.CHQ.

while varying the amount of main memory available for the graphs. We experimented with three different settings for the size of the main memory buffer available at each site (including the query site): 10, 100 and 1000 pages respectively, each page being of size 1 Kbyte. In Section 2 we presented two variants of PIPE.CHQ, intended for the extreme cases where all graphs fit in main memory or none of them fit. In practice, we also encounter a number of hybrid cases. One such case occurs when each bipartite graph BG_{R_{i-1}, R_i} fits in main memory at site S_i , but at the query site there is not enough space to store all reduced bipartite graphs. In this case, step 4 of the disk-based PIPE.CHQ is modified, since no page identifiers are present. A second hybrid configuration occurs when some bipartite graphs fit in main memory at their respective sites. At each site S_i the inclusion of $page(id_{i-1})$'s in BG_{R_{i-1}, R_i} depends upon whether the graph at S_{i-1} is main memory resident or not.

Figure 11 compares the total processing times of PIPE.CHQ in a main memory system with those for disk-based systems with various buffer sizes. In all test cases run, a disk-based system with buffers of size 1000 pages behaved identically to a main memory system: all graphs fit in memory; hence, no deterioration in performance occurred. In the case of buffers of size 10 pages, on the average, 31% of the graphs fit in main

24

