Discussion for Case I

1. From the second column of Table 1, the total numbers of page pulls are relatively regular; even if the blocking factor is 1 (minimum value) or 33 (maximum value), the total number of page pulls is only about 4 times of the smallest number of page pulls. Thus, the page pull curve is very smooth for DBM. Therefore, dynamic block-mapping reduces the page pulls significantly.

2. Compare the page pulls in the third and fourth column in Table 1 for RBM and CBM. For every B, the total number of page pulls for RBM is smaller than that for CBM. The arrays A and C will profit if the mapping strategy is row-major, while array B will profit if the mapping strategy is column-major. Hence, the page pull curve for RBM is above the page pull curve for CBM.

3. When B=1 or 33, the total number of page pulls is 10268 and 10395 for RBM and CBM respectively. If we take the conventional row-major mapping and column-major mapping, the resulting page pulls will be the same as the above values. These page pulls are about one hundred and twenty times of the optimal number of page pulls (87).

Discussion for Case II

1. If P=1 or P=2, the total number of page pulls is very large (equal to the total number of array elements). Here the working set cannot hold 3 blocks (one each for A, B, C); none of the element that have been referenced can be reused.

2. For A, B, and C to reside in main memory, we need 27 pages. For DBM, if the total number of page pulls is 27, we need 20 pages of main memory. For RBM, if the total number of page pulls is 27, we need 18 pages of main memory. For CBM, the corresponding number is 25 pages.

3. When we have at least six pages of main memory, all of the page pulls are acceptable and the page pull curves for DBM, RBM, and CBM are very smooth.

We conclude that the conventional row-major or column-major mapping is not good enough. If the programmer does not design a program carefully, the program I/O behavior may be much worse than a well designed program. However, most of the programmers only concentrate on the correctness of their programs and pay less attention to whether their programs run fast or not. Therefore, if the compiler can parse the source code and analyze which parts of the code can be blocked, then, by applying block-mapping strategy, the data locality can be improved to a great extent.

The fixed block-mapping strategy, either RBM or CBM, works well only if we choose a good blocking factor within some range; in contrast, dynamic block-mapping only depends little on the choice of blocking factor. If the array reference patterns tend to be highly structured, dynamic block-mapping can greatly improve the program I/O behavior.

Bibliography
