

ABSTRACT

Dynamic Memory Management Algorithms in a
Paged Memory Environment. (May 1976)

David Sherwin Burris, B.S., Southwest Texas State University

M.A., Southwest Texas State University

Chairman of Advisory Committee: Dr. Udo W. Pooch

Two algorithms are presented which dynamically cluster pages of a problem program based on past program behavior (i.e., reference string patterns) in a demand paged virtual memory environment. The objective of these algorithms is to minimize the number of page faults encountered by a program during execution, while at the same time to use memory page frames efficiently. Dynamic clusters of "time and reference" related pages are built during a program execution time.

Whenever a page fault for the i -th instruction page occurs, in this time evolving environment, the pages of the cluster associated with the i -th page are compared to the pages currently in real (physical) memory. Thus during the page fault, the demand page, and any associated clustered pages not currently in physical memory are placed into memory. Page frames holding pages not in the current cluster are returned to the memory management system. Thus the physical amount of memory allocated to a processing program is dependent upon the size of the cluster associated with the instruction page at that time. When the current instruction page ceases to hold the next instruction, but the next sequential instruction page J is currently in real memory, pages not in instruction page J 's domain (cluster) may be returned to the memory management system.

Simulation results of program behavior operating under these dynamic clustering strategies indicate that improvements in the page fault rate and in the space time product (memory utilization) may be achieved. Both algorithms require fewer page faults, and consequently have a lower space time product per executed instruction than most

currently implemented algorithms that utilize a fixed number of page frames per problem programs (i.e., fixed allocated partition, stack, or region).

The first algorithm, the LMM (Locality Matrix Model), an extension of work by Hedges and Pooch (10), is used to determine inherent program locality and predict dynamic program behavior. The LMM separates instruction from data references. Furthermore, strength coefficients between weakly or loosely coupled pages may be used to refine the cluster population.

The second algorithm, DCM (Dynamic Cluster Model), creates two separate locality measures for each program page, and then at the time of a page fault, merges this information. For each instruction page i , the collected information reflects:

- (1) The set of pages to which control may transfer when page i ceases to hold the current instruction.
- (2) The set of data pages referenced when page i is the current instruction page.

Whenever a page fault occurs these two sets of pages are merged and brought into physical memory during the demand page fault. Again, strength coefficients between weakly or loosely coupled pages may be used to refine the cluster population. Both algorithms appear feasible for hardware implementation using current technology.