Chapter 4

CM Fortran Performance Aspects

Getting the best performance from a CM Fortran program is a combination of three factors:

- what the CM Fortran compiler gives you automatically
- what the compiler can give you if you help it (that is, if you write statements so they can be compiled effectively)
- what you can do on your own (better program design, effective programming strategies, etc.)

This section of the manual will give you tips in all three areas, arranged by category.

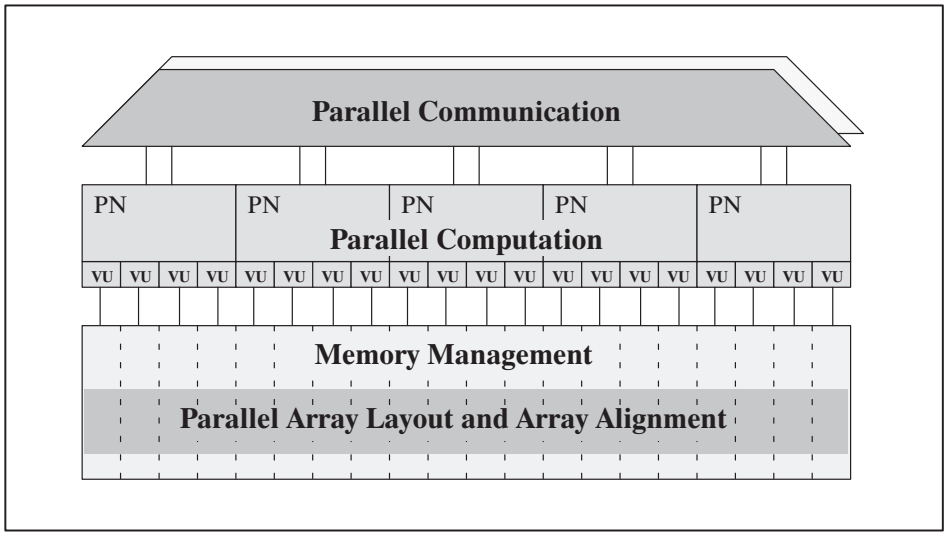

Figure 9. The major categories of CM-5 performance enhancement.

Copyright © 1994 Thinking Machines Corporation

The structure of the CM-5 neatly divides the possible performance enhancement issues into five major categories (see Figure 9):

- *Array layout*: How efficiently the data in each parallel array is distributed among the PEs of the CM. Improving array element locality (the distance between neighboring array elements in the hardware of the CM) can significantly improve the performance of your code. Array layout is discussed in Chapter 5.

- *Array alignment and aliasing*: How well arrays used in parallel array operations match each other in terms of array layout. Aligned arrays put matching elements in the memory of the same PE, allowing array arithmetic to be performed directly as parallel computation. You can also use array aliasing to give the compiler modified information about the data type and layout of an array. Alignment and aliasing are discussed in Chapter 6.

- *Parallel Computation*: How effectively the parallel computations in your code compile into PE code blocks, and how effectively the compiler can optimize your code. Parallel computation is discussed in Chapter 7.

- *Parallel Communication*: The efficiency of the communication calls made by your code. Parallel communication is discussed in Chapter 8.

- *Memory Management*: How effectively your program allocates and deallocates the parallel memory of the CM. Memory management is discussed in Chapter 9.

There are also miscellaneous performance issues, such as the use of optimized CM library routines, global/local program execution, etc. These are discussed separately in Chapter 10.

Version 2.2, October 1994
Copyright © 1994 Thinking Machines Corporation