About This Manual

Objectives of This Manual

This manual provides information to help users increase the performance of their CM-5 C* programs.

Intended Audience

This manual is intended for programmers who are familiar with C* and the architecture of the CM-5. Some understanding of DPEAC and SPARC assembly language is helpful but not required.

Revision Information

This is a new manual.

Organization of This Manual

The manual is organized as follows:

Chapter 1 **The Compilation and Execution Models**
This chapter describes how C* programs are compiled for the CM-5 and discusses how the compiler constructs PN code blocks to carry out parallel computation on the processing nodes.

Chapter 2 **Timing and Profiling Code**
This chapter discusses the use of the **`CM_timer`** facility and Prism's performance analysis facility for timing and profiling C* programs.

Copyright © 1993 Thinking Machines Corporation

Chapter 3 General Performance Tips
This chapter gives some general advice about how to improve performance.

Chapter 4 Performance Tips for Parallel Computation
This chapter focuses on how to increase the performance of code that performs parallel computation (that is, operations that take place independently in each processor).

Chapter 5 Performance Tips for Parallel Communication
This chapter focuses on how to increase the performance of code that performs parallel communication (that is, operations that require transferring data between processors).

Chapter 6 Reducing Memory Usage
This chapter explains how C* uses memory on the CM-5, and gives some tips on reducing memory usage.

Appendix Examining Generated Assembly Code
The appendix looks in detail at the scalar and parallel assembly code generated for a simple C* program.

Related Documents

For further information on CM-5 C*, see the *C* Programming Guide* and the *CM-5 C* User's Guide*.

See the *CM-5 Technical Summary* for information on the CM-5's architecture.

For information on DPEAC, see the *DPEAC Reference Manual*.

Copyright © 1993 Thinking Machines Corporation

Notation Conventions

The table below displays the notation conventions observed in this manual.

Convention	Meaning
`bold typewriter`	UNIX and CMOST commands, command options, and filenames, when they appear embedded in text. Also programming language elements, such as keywords, operators, and function names, when they appear embedded in text.
italics	Argument names and placeholders in function and command formats.
`typewriter`	Code examples and code fragments.
`%` **`bold typewriter`** `regular typewriter`	In interactive examples, user input is shown in **`bold typewriter`** and system output is shown in `regular typewriter` font.

Copyright © 1993 Thinking Machines Corporation