

CSC 464-01 Term I 2004 PARALLEL PROGRAMMING

12:00-12:50 pm MWF Room: E 210

COURSE DESCRIPTION. This course introduces parallel programming techniques. Parallel programming uses multiple computers, or computers with multiple internal processors, to solve a problem at a greater computational speed than using a single computer. In this course we discuss the use of multiple computers that communicate between themselves by sending messages. The computers we use can be different types but must be interconnected by a network, and a software environment must be present for computer message passing. Several software tools are available for message-passing parallel programming. In this course we utilize the MPI software routines.

PREREQUISITES. CSC 172

INSTRUCTOR. Dr. M. S. Skaff Office: E204 Telephone: 993-3376
E-mail: skaffms@udmercy.edu

OFFICE HOURS. 11:00-12:00 am MWF, 4:00-5:00 pm MW, and anytime my office door is open or by appointment.

TEXTBOOK. Parallel Programming by B. Wilkinson and M. Allen, 2nd edition, Prentice-Hall Publisher, 2005, ISBN: 0-13-140563-2

COURSE OBJECTIVES. To understand and learn the concepts involved in parallel processing. In particular, we investigate a variety of networks utilized in parallel processing, message-passing concepts and techniques, embarrassingly parallel computations, and partitioning and divide-and-conquer computations. Moreover, we learn parallel algorithms for a variety of problems where both theoretical and applied ideas are considered.

COURSE OUTCOMES. After taking this course, students will be able to understand:

1. Different types of parallel computers such as a shared memory multiprocessor system, message-passing multicomputer, distributed shared memory computer, and MIMD and SIMD classifications.
2. Architectural features of message-passing multicomputers.
3. Network computers as a multicomputer platform.
4. Basics of message-passing programming.
5. Evaluating and debugging parallel programs.
6. Embarrassingly parallel computations.
7. Partitioning and divide-and-conquer strategies.
8. Pipelined computations
9. Synchronous computations
10. Load balancing and termination detection.
11. Programming with shared memory.

COMPUTER PROJECTS. There will be two projects. The first project will be a report of executing canned MPI applications in the parallel processing laboratory. The second project will utilize MPI commands in solving a problem using both sequential and parallel programming techniques. Students should be familiar with C++ , JAVA and/or object-orient programming concepts.

EXAMINATIONS. There will be **three (3) closed book examinations** to be taken using bluebooks. These bluebooks or exam booklets may be purchased at the bookstore in the student union. The three bluebooks must be submitted by **09/15/04**. Put your name and CSC 464 on the cover. No notes, books, computers, or dictionaries will be allowed on any examination.

All exams will be graded on a curve. Exams may be either take-home or multiple-choice. No makeup exams will be given. The final grade for the course will be based on the best 4 scores achieved during the term. One low score or missed exam will be dropped (see **GRADING** section below).

EXAM DATES. The three exams will be given on the following dates. Mark your calendars now!

EXAM 1. Wednesday September 29, 2004

EXAM 2. Wednesday October 27, 2004

EXAM 3. Wednesday December 1, 2004

There is no final exam for the course.

PROJECTS. Two projects are required for the course. These projects are defined in the packet of material distributed with this syllabus.

Project 1: Project report sections 1-4 due by September 20, 2004

Final project report due by October 18, 2004

Project 2: Project report sections 1-4 due by due by November 8, 2004

Final project report due by December 8, 2004

GRADING. The best 4 out of 5 grades for all examinations and projects will be averaged for a course grade. If any examination or project is not submitted or missed, that score will be eliminated as the “dropped” grade. If more than two grades are missing, a grade of F will be given for the extra missing grades. No makeup exams or extra projects will be given or accepted.

IMPORTANT FACTS. Last day to withdraw with no W – October 1, 2004

Last day to withdraw with a W - November 22, 2004

ACADEMIC INTEGRITY. Everything submitted for grading is expected to be student’s own work. Anything suspected as being otherwise the case will be dealt with according to the College policy.

CSC 464 LABORATORY INSTRUCTIONS

These instructions are to be followed when accessing the parallel programming software in the laboratory.

1. Pick any computer to be the host computer. All other computers are called Nodes. Go to the E:drive (local drive not C) on the host computer. Create a new folder call **TEMP** (use any name you like).
2. Click on **START**. Click on **MY COMPUTER**. Click on **local drive (C:)**. Click on **PROGRAM FILES**. Click on **MPICH**.
3. Click on **SDK** folder. Click on **EXAMPLES** folder. Click on **NT** folder.
4. Copy the **contents** of the **NT** folder to the **TEMP** folder you created in step 1.
5. Start **Visual C++** version 6.0 by clicking on **START**, **PROGRAM FILES**, and **Microsoft Visual Studio 6.0**.
6. In the visual C++ window, **CLOSE** the **Tip of the Day** window (if it appears). Click **FILES**. Click **OPEN "Workfile"**. Browse to **TEMP** folder in the E:drive and double click **EXAMPLES.DSW**.
7. Right Click on the specific **PROJECT** application you wish to run (eg., **cpu**, **mpptest**, ...). Then left click **SET AS ACTIVE PROJECT**.
8. Click on **PROJECT**. Click **SETTINGS**.
9. Click on **C/C++**.
Under **SETTINGS FOR**: select **WIN 32 DEBUG**.
Under **CATEGORY**: select **CODE GENERATION**
Under **RUN TIME LIBRARY**: select **DEBUG MULTITHREADED**
10. Click on **C/C++**
Under **SETTINGS FOR**: select **WIN 32 RELEASE**
Under **CATEGORY**: select **CODE GENERATION**
Under **RUN TIME LIBRARY**: select **MULTITHREADED**
11. Click on **C/C++**
Under **SETTINGS FOR**: select **"All Configurations"**
Under **CATEGORY**: select **"Preprocessor"**
Under **ADDITIONAL INCLUDE DIRECTORIES**: delete or clear field and enter **E:\PROGRAM FILES\MPICH\SDK\include**
12. Click on **LINK**.
Under **SETTINGS FOR**: select **"All Configurations"**
Under **CATEGORY**: select **"Input"**
Under **ADDITIONAL LIBRARY PATH**: enter the same path as in step 11 except replacing **"include"** with **"lib"**
13. Click on **LINK**.
Under **SETTINGS FOR**: select **"Win 32 Debug"**
Under **CATEGORY**: select **"General"**
Under **OBJECT/LIBRARY MODULES**: Insert (or precede) with **"ws2_32.lib mpichd.lib"** (be sure to have spaces between each entry)

14. Click on **LINK**.
Under **SETTINGS FOR:** select “**Win 32 Release**”
Under **CATEGORY:** select “**General**”
Under **OBJECT/LIBRARY MODULES:** Inset (or precede) with “**ws2_32.lib mpich.lib**” (be sure to have spaces between each entry.)
15. Click on **OK**
16. Select the project to be processed. Right click on the project name and select “**Set as active project**”. Click on **BUILD** and executable name.exe (eg., **build cpi.exe**)
17. If there are no errors from BUILD operation, The executable (eg., cpi.exe) resides in E:\TEMP\BASIC\PDEBUG. Copy this executable to the **TEMP** folder in E: drive on the HOST computer.
18. Go to the **C:\Program Files\mpich** folder and click the **MPD** folder. Click the **BIN** folder. Copy the **MPIRun.exe** program from the **BIN** folder to the HOST computer **TEMP** folder in the E:drive.
19. Copy the **TEMP** folder on the E:drive in the Host Computer to a Floppy. Then copy the contents of the Floppy to E:\TEMP folder in each Node computer
20. Next we need to configure the computers required to run the application program. To do this we click on **START**, click on **PROGRAM FILES**, click on **MPICH** on the HOST computer. Then click on **MPD** folder. Click on **MPICH Configuration Tool**. **Select** the computer ID’s for each computer to be used as Nodes. The Host computer ID is already on the screen. Click the ID’s of those computers to be used as Nodes. (Highlight all those to be selected and Click Add).
21. To run the application: go to the **TEMP** folder on the HOST computer. In **DOS** mode at the **E:\TEMP>** prompt, for all applications except **Mandel** type **mpirun -np k -logon executable name** where k=number of nodes+1. For example, if one host and 2 nodes are used, then k=3. We then type **mpirun -np 3 -logon cpi**. For the **MANDEL** application we would use **mpirun -np k -localonly mandel**.

NOTE: User name and password will change. Currently, it is User name: dpl_ ,
Password: dpl

CSC 464 PROJECT 1

BACKGROUND. To experience parallel programming techniques we need software that can implement parallel programming concepts. In this project, you will learn about MPI software as you execute canned programs in the parallel programming laboratory. For each canned program, one computer will be designated as the host computer and all other computers will be called nodes. For project 1, you will select one of the canned programs, execute the program, and write a report documenting what you observed. The instructions needed to execute the laboratory programs and the report outline requirements are given below.

CANNED APPLICATIONS. There are six canned applications in the laboratory system. These are (1) cpi (2) fpi (3) mandel (4) mpptest (5) netpipe, and (6) systest. Some of these may not execute correctly. By trying to run all of them, you will find those that execute correctly and those which you can explain in your report.

For example, the mandel application is a program which uses a graphics package to display the mandelbrot set on the screen. It is a demonstration of parallel graphic programming by drawing a typical shape. The mpptest is an application that measures the communication performance of a message passing system.

REPORT OUTLINE REQUIREMENTS. The project report must follow the outline below:

1. **TITLE PAGE.** Center title of report to the page with your name and CSC 464 in the lower right hand corner.
2. **SUMMARY FOR EXECUTIVE READER.** A brief (less than one page) summary of the content and purpose of the report.
3. **TABLE of CONTENTS.** Show page numbers for major sections of the report.
4. **MAIN CONTENTS of PAPER.** This section is the principal information of the report.
5. **OUTPUT.** This section documents any computer output that exists.
6. **REFERENCES.** List any references that are utilized in the paper including textbooks, library books, or internet information.