This course is available for student registration only after the approval process has been completed.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Course No.</th>
<th>Credit Hours</th>
<th>Term to be added to the file</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTH</td>
<td>MATH 4682</td>
<td>3.0</td>
<td>Fall 2004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class Hours</th>
<th>Lecture Hours</th>
<th>Lab Hours</th>
<th>Contact Hours (CEU only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Department: Mathematics

<table>
<thead>
<tr>
<th>College/School</th>
<th>(Please check appropriate box)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>College of Engineering-01</td>
</tr>
<tr>
<td></td>
<td>College of Science and Liberal Arts (science)-20</td>
</tr>
<tr>
<td></td>
<td>College of Science and Liberal Arts (liberal arts)-21</td>
</tr>
</tbody>
</table>

Computer Title (restricted to 25 spaces, including blanks):
Introduction to Parallel Processing

Catalog Title: Introduction to Parallel Processing

Catalog Description of Course (limited to 350 characters, including spaces):
A general introduction to parallel algorithms and architectures for parallel computer programming paradigms. Speed and cost for shared memory and distributed memory computers.

In addition, you may attach a course syllabus and/or more detailed description.

Restrictions:
- Prerequisite: (course number)
- Corequisite: (course number)
- Prerequisite: (course number)
- Corequisite: (course number)
- Prerequisite: (course number)
- Corequisite: (course number)

Grades to be issued:
- A, B, C, D, F
- S, U
- P, F
- Other

Additional Restriction (e.g., major, class level, department head approval):

If this course replaces a course currently offered in BANNER, please indicate old course information.

Upon completion of appropriate department approvals, submit form to Chair, Graduate Council, or Chair, Undergraduate Curriculum Committee for approval below and forward to Catalog Coordinator.

Originator: Michael A. Shaw
Date: 2/3/04

Chair, Graduate Council
Date

Chair, Undergraduate Curriculum Committee
Date

Dean or Associate Dean
Date

CATALOG COORDINATOR
Date

REGISTRAR’S USE ONLY

SCARCS
SCADETL
SCAPREQ
SCBASE
SCARRES
Operator in it
Date

DISTRIBUTION:
Original—Registrar
Copy—Academic Unit/SEGS

Florida Institute of Technology • Office of the Registrar
150 West University Boulevard, Melbourne, FL 32901-6975 • (321) 674-8136 • Fax (321) 674-7827
RG-407-003
MTH 4XXX: Introduction to Parallel Processing

Proposed Catalogue Description:
This course is a general introduction to parallel algorithm development. Architectures for Parallel Computers, programming paradigms SIMD and MIMD for shared memory and distributed memory computers. Parallel algorithms for matrix computations, sorting and searching, and various numerical algorithms are presented and analyzed. Analysis of performance of parallel algorithms and scalability of algorithms is discussed. (Prerequisite: Programming ability in Fortran or C)
Proposed Syllabus for New Math Course in Parallel Processing:

\[ H082 \]
MATH 4XXX Introduction to Parallel Processing

Textbook:

Peter S. Pacheco,
Parallel Programming with MPI,

Other References:

MPI-The Complete Reference, Volume 1.

B. Wilkinson and M. Allen,
Parallel Programming Techniques and Applications Using Networked

S.G. Aki,
The Design and Analysis of Parallel Algorithms,
Prentice-Hall, 1989


COURSE SYLLABUS

1. Chap 2, Sec 2.1.1-2.2.2

Computer Architecture: Flynn's taxonomy for Parallel computers, SIMD (Single
Instruction, Multiple Data) and MIMD (Multiple Instruction, Multiple Data)
architectures and their variants. Shared and Distributed memory machines, memory
hierarchy: main CPU, cache and registers.

2. Chap 3, Sec 3.1-3.4
Chap 5, Sec 5.1-5.8

Introduction to MPI: Message Passing Interface. Various MPI interprocessor
Communication functions (protocols) for passing data between processors on a
Distributed Memory machine: Send-Recv, Broadcast, Reduce, Gather, Scatter.
Allgather, Allreduce. Discussion of the SPMD (Single-Program, Multiple Data) programming model to be utilized on the 48-Node Beowulf Cluster at Florida Tech. These MPI functions will be studied and used in connection with parallel programs supplied by Pacheco in C and Fortran.

3. Chap 4, Sec 4.1-4.2
   Chap 5, Sec 5.5, 5.7

Assignments to modify, run and do performance analysis on the following parallel programs will develop student's abilities to write, run and debug parallel programs:
   (1) Parallel computation of the dot (inner) product of two vectors
   (2) Parallel implementation of the Trapezoidal Rule for numerical integration
   (3) Parallel Matrix-Vector multiplication

4. Chap 6, Sec 6.1-6.5

Performance of message-passing functions of MPI can be improved grouping data into a single message. Mechanisms for doing this are using the Count parameter in MPI functions, Derived Data-Types and the MPI_Pack and MPI_Unpack functions.

5. Chap 7, Sec 7.1-7.8

Communicators or Contexts and their Topologies. SPMD Programs with more than one Communicator will be studied in connection with the FOX Algorithm for multiplication of two Matrices.

6. Performance Analysis and Debugging

Throughout the study of Chaps 4,5,7 performance analysis of parallel algorithms will be emphasized. Measurements of communication time, computation time, speed-up measures, scalability, load-balancing and Amdahl's Law will be discussed and emphasized in homework problems. There will be emphasis on predicting performance of parallel programs, based on knowledge about latency of message passing, expected communication overhead, and operation counts. Material on this topic will be taken from Chap 11 and 12 of Pacheco's book, Chap 1 of Wilkinson and Allen's book and other appropriate sources, including documents on the PGI Profilers from the Portland Group. Also, students will learn to make use of the Total View Debugger from Etnus, using on-line documentation.

7. Introduction to the Parallel Library ScaLAPACK
One of the most useful Parallel Libraries installed on Florida Tech’s 48-Node Beowulf Cluster in the ScaLAPACK library for doing most of the usual numerical linear algebra algorithms in parallel on MIMD machines. This involves the BLACS (Basic Linear Algebra Communication Subroutines, which are built on top of the MPI library) and the PBLAS (Parallel Basic Linear Algebra Subroutines) as the underlying routines for communication and elementary BLAS operations (vector, vector-matrix and matrix-matrix operations). The Block Cyclic data distribution which underlies the ScaLAPACK library will be discussed. Some supplied routines from the ScaLAPACK distribution, such as vector-matrix multiplication, matrix multiplication, and Gaussian elimination for solving linear equations, will be experimented with and performance testing on them done.

TERM PROJECT:

In addition to regular homework exercises, each student will write a term project involving the writing, debugging and performance analysis on a parallel algorithm. Topics will vary depending on student’s backgrounds but may be selected from the following general areas: (i) Sorting Algorithms, (ii) Numerical Linear Algebra Algorithms such as LU factorizations, QR factorizations, least squares algorithms, or Fast Fourier Transform algorithm (iii) image processing algorithms, (iv) searching and optimization algorithms, (v) modification and development of some of the MPI functions. Algorithms for the first four subjects can be found in Wilkinson and Allen, Chaps 9, 10, 11, 12 and in S. Aki, Chaps 4, 5, 8, 9, 13. For MPI functions currently implemented, the general reference is Snir, et. al., MPI-The Complete Reference, Vol. 1.