SYLLABUS

ECE 493: Introduction to Parallel Systems 3(3,0)
Section 1 - Fall Semester, 2004

Goals:
Understand how to program with MPI
Understand important numerical and non-numberical algorithms
Introduce performance analysis concepts

Instructor: Walt Ligon, 656-1224, 300-D Riggs Hall,
walt@clemson.edu

Office Hours: MW 3:45 to 5:30 or by appointment

Required Text: Parallel Programming in C with MPI and OpenMP
By: Michael J. Quinn,

Grading:
Mid-Term Exam: 20%
2nd Exam: 20%
Assignments and Projects: 40%
Final Exam: 20%

Attendance:
Attendance is required
No late work accepted.
Make-up tests by prior appointment only.
Wait 15 minutes for late instructor.

Academic Honesty:
All work on quizzes, tests, exams, design assignments, projects,
and labs is to be wholly your own. Possessing, using, providing, or
exchanging improperly acquired written, verbal, or electronic
information will be considered a violation of the academic honor
code. Violations will result in a grade of F for the semester.
Study Guide 1

Parallel Computing
Goal: speeding up computation
Interesting problems communicate
Ways to communicate
shared memory
message passing

Message Passing
send/receive data
issues
synchronization
blocking/non-blocking
synchronous
asynchronous
buffering
none
limited
infinite
explicit
naming
direct/indirect
symbolic
symmetric/asymmetric
data size and type
collective communication

MPI
basic calls
    Init, Comm_rank, Comm_size, Send, Recv, Finalize
communicators, size ranks
basic data types
tags
semantics
    non-overtaking
    progress
    no fairness
    limited resources
Sendrecv, Sendrecv_replace
IO models
    master task
    independent
    hybrid
    parallel IO
collective operations
    Barrier, Bcast, Reduce, Allreduce, Scatter, Gather, Allgather, Alltoall, Reduce_scatter,
    Scatterv, Gatherv, Allgatherv, Alltoallv, Scan
modes
    normal
    buffered
    synchronous
    ready
non-blocking IO
    Isend, Issend, Ibsend, Irsend, Irecv
    Test, Wait, Testany, Waitany, Testall, Waitall, Testsome, Waitsome
MPI (cont)

dervied (user-defined) datatypes
- Type_contiguous, Type_vector, Type_hvector, Type_indexed, Type_hindexed,
  Type_struct
- Type_commit, Type_free, Type_size, Type_extent

groups
- What are they and why do we have them?
  Group_incl, Group_excl, Group_rank, Group_size

communicators
- What are they and why do we need them?
  Comm_create, Comm_group, Comm_split

topologies
- What are they and how do they help programmers?
  Cart_create, Cart_coords, Cart_rank, Cart_shift

Parallel Program Design

Foster's methodology
- partitioning
- communication
- agglomeration
- mapping

Decomposition
- data
  for 2D structures:
    by row, by column, checkerboard
  interleaved
  block

functional

Performance Analysis

serial and parallel runtime
- speedup
- Amdahl's Law
- efficiency
  isoefficiency
  Overhead function
  Scalability function