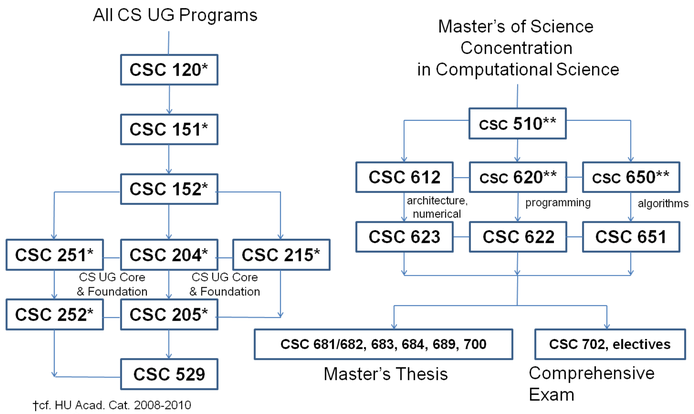
*Department of Computer Science
Hampton University
Hampton, VA 23668

Motivation

- Early experience institutionalizing MS and PhD program in Computational Science at NCAT
  - after PI departure, program faltered
  - want to do the same at HU and remain
- Three graduate courses:
  COMP 732 – Advanced Software Tools
  Spring 2004, 2006
  COMP 733 – Parallel Computing Applications
  Fall 2003
  COMP 770 – Computer Organization and Programming for Scientific Computing
  Fall 2004, Fall 2005
- Cps entails parallel computing and HPC on clusters on up to supercomputers
- HU CpS research begins
- Intel™ has donated Parallel Studio software suite
- AI, SE have established curricula
- CpS does not have an established curriculum
- We are proposing to NSF HBCU RISE to increase PhD production in SMET areas
- Hardware and Software are in place at HU
- Parallel computing abilities are now required of UG and Grad students
  - Parallel systems are now widely available
  - Many applications in simulation & modeling, data analysis, image and signal analysis, data visualization
  - Many sectors: medical, scientific, academic, industrial, commercial, governmental
  - Wide range:
    - small scale department project (< 32 nodes)
    - large scale government project (> 10^5 = 100,000 nodes)
  - New tools are widely available: OpenMP, MPI, CUDA, OpenCL

Overview

We infuse 1) parallel programming into CSC 151, 152,
2) parallel architecture into CSC 204, 205 and
3) parallel algorithms into CSC 251, 252



†cf. HU Acad. Cat. 2008-2010
*core UG
**core Grad + CSC 630, 640, 660

Early Adopter - Parallel Computing: Keys to ***a*** Future in Computing

First NSF/TCPP Workshop on Parallel and Distributed Computing Education, 2011

An Undergraduate to Graduate Student Curriculum / Concentration

MS Program in Computer Science

Stephen V. Providence Ph.D.
Graduate Coordinator
Assistant Professor of Computer Science
Department of Computer Science
Hampton University, Hampton, VA 23668
E-mail: stephen.providence@hamptonu.edu
Voice mail: (757)728-6406
Department: (757)727-5552

CS UG Core & Foundations†

- CSC 204* – Computer Architecture, Systems and Organization I
  Binary number representation and arithmetic. Computer structure. Addressing techniques. Storage allocation. Subroutine linkage. Relocatability and program segmentation, bit manipulation. Operating system supplied I/O routines and interface using a systems programming language and assembly language. Macros.
  Prerequisites: CSC 152, MAT 117
- CSC205* – Computer Architecture, Systems and Organization II
  Intermediate logic design including truth tables, logic diagrams, Boolean functions and K-maps. Computer architecture including CPU design, memory organization, I/O processing including programmed I/O, interrupt I/O, and direct memory access. (Multi-computers, Flynn's taxonomy, MESI cache coherence, UMA & NUMA multiprocessing, Spring 2010) Coding. Prerequisite: CSC 204
- CSC 529 – Topics in Parallel Programming
  A treatment of topics not routinely covered by other courses. (foundations parallel programming in MPI & OpenMP, Spring 2010)
  Prerequisite: Permission of the instructor
- CSC 215* – Discrete Structures
  Set theory, logic and combinatorics. Relations and functions. Proof techniques, including mathematical induction. Introduction to graph theory. Ethical principles in computing.
  Prerequisites: CSC 152, MAT 117 (pre-calculus)
- CSC 251* – Data Structures and Algorithm Analysis I
  Representation of compiler related data structures. Contiguous linked and hashed representations. Empirical and abstract analysis of time and space of competing representation. Space optimization. Specification, design, implementation, and verification of linear and hierarchical; abstract data types, including stacks, lists, queues and trees. Basic techniques to algorithm design and analysis; ethical principles in computing.
  Prerequisite: CSC 152, Co-requisite: CSC 215
- CSC252* – Data Structures and Algorithm Analysis II
  An in-depth survey of data structures and algorithms, exploring their design, running efficiency, and applications. Advanced methods for interval and external sorting and searching. Implementation of relational data types including directed and undirected graphs. Advanced algorithms, which may include parsing, breadth-first and depth-first graph traversals, minimum weighted paths and information, flow analysis, (parallel algorithms), ethical principles in computing. Prerequisites: CSC 204, 215, 251

About the Cluster....

The cluster is a Hybrid Cluster. It includes both GPU and CPU nodes.
•1 nVIDIA Tesla GPU with 240 Core
•10 processing nodes with 8 cores per node
-2 Intel Xeon Quad Cores
• Xen Virtualization allows these 8 core to be doubled and work as though there are 16 cores per node
•400 Core Total

Zeus Visualization

Zeus CPU - GPU Nodes

Zeus KVM

Student Parallel Computing Research

Modified Sieve Code Segment

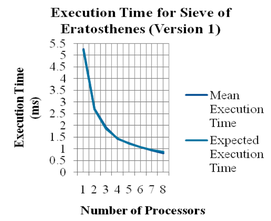
```
elapsed_time += -MPI_Wtime(); // Starts Timer

if(p > 1)
MPI_Bcast(&prime,1,MPI_Int,0,MPI_COMM_WORLD)

elapsed_time += MPI_Wtime(); // Stops Timer
```

•The modified sieve algorithm was run 5 times for 2 ,4,6, and 8 processes. The values were then averaged to determine the overall broadcasting time for the entire system.

Benchmarking Results



*Department of Computer Science
Hampton University
Hampton, VA 23668

All CS UG Program Courses†

- CSC 120* – Introduction to Computers
  This course helps the student understand how computers can be used to enhance his or her personal, academic or professional life. A hands-on approach is used to introduce students to various software packages for word processing, spreadsheet applications, and database management.
- CSC 151* – Computer Programming I
  Computer basics; goals of quality software concepts of input/output, constants, variables, expressions; program control structures including iterations, sequence, selection; concepts of object oriented programming; use of a high level language; concepts of event-driven programming; introduction to Graphical User Interface components; introduction to input/output for files; introduction to arrays; ethical principles in computing.
- CSC 152* – Computer Programming II
  Problem solving techniques. Principles of good programming style, documentation, and robustness. Introduction to Data Structures. Object-based and Object-Oriented Programming. Exception handling files and streams. Ethical issues in computing. Prerequisite: CSC 151

Master's of Science Concentration in Computational Science†

- CSC 510** – Mathematical Foundations
  Propositional and Predicate Calculus. Proof techniques. Queuing theory. Mathematical formulations of data structures. Basic models of computation expressions and grammars. Prerequisites: CSC 215, 252
- CSC 612 – Numerical Computation
  Numerical and optimization methods useful for simulation, graphics, and image processing. Computation statistics and Monte Carlo methods. Signal analysis foundations. Prerequisite: CSC 510 or permission of the instructor
- CSC 620** – Operating Systems
  Relation between architectures and operating systems. Multiprogramming timesharing, multiprocessing, distributed processing and real time processing. Inter-process communication and synchronization. Resource allocation and related problems.
- CSC 622 – Parallel Processing
  Parallel processing models and architectures. Concurrent processes and controls. Parallel algorithms and their analysis. Prerequisite: CSC 620 or permission of the instructor

CSC 205 Interest Results