Early Adopter – Parallel Computing: Keys to a Future in Computing
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

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

• Brief History, CpS & Motivation
• Course Map & Description
  – Course map
  – CS UG Common Courses
  – CS UG Core & Foundations
  – MS Concentration in Parallel Computing
• < Two year Master’s plan in concert w/ established plan A or plan B
• Thesis topics (plan A) w/ research faculty
• Comprehensive Examination (plan B)

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

• Hampton University’s MS Program in CS
• Emphasis on AI, SE and CpS
• Subset of established facilities
  – Robotics lab created and courses run
  – IA lab created and UG courses run
• Update of program required
  – CpS facility missing, emerging field in CS
  – Many institutions already have CpS (i.e. A&T, W&M)

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

- Computational Science (CpS) facility under construction & expansion
- Summer 2008, Department awarded NSF MRI CNS-0821475
- Summer 2009, NSF REU supplement CNS-0941150
- Fall 2009, Department installed 400 core nVidia™ Tesla™ GPU/ Intel™ Xeon™ CPU cluster
- Spring 2010, research in CpS begins, courses run

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

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

• 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
Undergraduate and Graduate Computational Science†

All CS UG Programs

- CSC 120*
- CSC 151*
- CSC 152*
- CSC 251*
- CSC 204*
- CSC 215*
- CSC 252*
- CSC 205*
- CSC 529

Master’s of Science Concentration in Computational Science

- CSC 612
- CSC 620**
- CSC 623
- CSC 650**

- CSC 681/682, 683, 684, 689, 700
- Master’s Thesis

- CSC 702, electives
- Comprehensive Exam

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

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

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

- **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

- **CSC 252* – 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

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

†cf. HU Acad. Cat. 2008-2010
*core UG
**core Grad + CSC 630, 640, 660
Master’s of Science Concentration in Computational Science†

- **CSC 623 – Architecture**  
  Principles of computer design. Architecture of sequential and parallel computers. Prerequisite: CSC 620 or permission of the instructor

- **CSC 650** – Theory of Computation  
  Chomsky hierarchy. Turing machines, decidability and computational complexity.

- **CSC 651 – Algorithms and Complexity**  
  Turing machines and equivalent computational models. Church’s thesis. Complexity classes. (Parallel algorithms, PRAMs and parallel complexity classes, TBA) Prerequisites: CSC 510, 650 or permission of the instructor

- **Required:**  
  - CSC 630** - Artificial Intelligence  
  - CSC 640** - Software Engineering Foundations  
  - CSC 660** - Programming Languages

- **Thesis option A**  
  - CSC 681 – Research Seminar I / CSC 682 – Teaching Seminar I, CSC 683 – Research Seminar II,  

- **Comprehensive Exam option B**  
  - Electives (2)  
  - CSC 702 – Master’s Comprehensive Exam

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

Research is conduct w/ research faculty on topics among the strengths of the Department

- AI, SE (cf. w/ Boonthum, Mohammad, Hu)
- CpS (cf. w/ Providence)
  - serial & parallel algorithms
  - multi- & many-core, distributed and parallel architectures
  - simulation & modeling, process-oriented, message-passing languages
  - numerical & algebraic computation, data analysis, data visualization

At least one year of research & thesis writing

- Concentration + CSC 681/682, 683, 684, 689, 700

†HU Acad. Cat. 2008-2010
First NSF/TCPP Workshop on Parallel and Distributed Computing Education, 2011
Plan B†

• Comprehensive Examination CSC 702
  – Currently multiple choice questions chosen at
    by pseudo-random generator from general CS,
    AI, SE, & CpS questions.
CSC 205 Interest Results

Chart Area

- Has the topic of parallel computing, parallel programming or parallel algorithms been brought up in this class?
- If you have a dual-core or two-core processor in your computer, does parallel programming seem like a way to exploit both cores?
- Has the notion of SISO, SIMD, MISO and MIMD been presented in this or the pre-requisite computer architecture course?
- Have you been introduced to pipelining as a form of parallelism?
- There are several parallel programming languages: MPI, CUDA and OpenCL, do any of these languages interest you, even if you have never heard of them?
- Most computers have graphics processing units (GPUs) inside or GPUs, would it be useful to learn how to program the GPU specifically?
- Currently, computers are not getting faster at the same pace as in the past. Multi-core and many-core processors are a solution to making them faster. Do you agree?
- Would you take a course in "Parallel Thinking" or "Supercomputing" if it were offered in the Fall 2017?
- Parallel computing is important in computer science.
- Not all computing problems can be solved by parallelism.
Future

• Procure Grad support funding
  – NSF REU supplement and DoD Fall 2011 for UG
  – NSF HBCU RISE proposal April 2012
• Increase pubs quantity
• Form pipelines to R1 institutions
  – Broaden research base and funding (NSF BPC)
• Increase pub quantity
• Plan PhD program
  – Curriculum expansion (NSF CCLI)
• Procure Doctoral support funding
• Start Doctoral program
  – UG research & education (NSF REU Site)
  – NSF HBCU RISE

†HU Acad. Cat. 2008-2010

First NSF/TCPP Workshop on Parallel and Distributed Computing Education, 2011
Undergraduate and Graduate Artificial Intelligence†

All CS UG Programs

CSC 120*

CSC 151*

CSC 152*

CSC 251*  
CS UG Core & Foundation

CSC 204*  
CS UG Core & Foundation

CSC 205*  

CSC 252*  

CSC 539

Master's of Science Concentration in Artificial Intelligence

CSC 510**

CSC 570

CSC 630**

CSC 640**

CSC 632

CSC 631

CSC 639

CSC 702, electives

Master's Thesis

Comprehensive Exam

†HU Acad. Cat. 2008-2010
*core UG
**core Grad + CSC 620, 650, 660
Undergraduate and Graduate Software Engineering†

All CS UG Programs

CSC 120*

CSC 151*

CSC 152*

CSC 251*

CSC 204*

CSC 215*

CSC 549

CSC 570

CSC 649

CSC 642

CSC 669

CSC 702, electives

Master’s Thesis

Comprehensive Exam

Master’s of Science Concentration in Software Engineering

CSC 510**

CSC 540**

CSC 660**

CSC UG Core & Foundation

CSC 640**

CSC 642

CSC 669

CS UG Core & Foundation

‡HU Acad. Cat. 2008-2010

*core UG

**core Grad + CSC 620, 630, 650