TEACHING PARALLEL AND DISTRIBUTED COMPUTING IN A REVOLTING NATION

Manal Helal

College of Computing and Information Technology (CCIT), Arab Academy for Science, Technology, and Maritime Transport

1 OVERVIEW

Parallel and Distributed Computing (PDC) is currently taught in undergraduate computing curriculum as separate topics in several courses using different technologies. We propose a laboratory setup where all computing courses can be taught with hands-on experiences in a distributed environment. The complexity of PDC can be simplified if broken down into the main technologies leading to it. Computer architecture and organisation, networking courses, programming courses, data structures and algorithms, artificial intelligence, image processing, numerical analysis and computational science, software engineering, project management, and practically all computing knowledge areas can integrate distributed applications, programming, and infrastructure.

Being in a revolting nation that missed many opportunities to keep up with technologies due to the continuous unrest, is a double-edged sword. The young are adventurous and willing to learn, and the bureaucratic administration is resisting any change. A continuous effort from many team players is required to achieve the proposed integration.

The following sections start by the knowledge area according to (ACM/IEEE CS Curriculum RTF, 2012). Then the courses with their relevant category according to (NSF/IEEE-TCPP , 2012), then inside square brackets the CCIT's course code, and the course title. Current course contents are described, and possible PDC topics to include are listed.

2 SDF-SOFTWARE DEVELOPMENT FUNDAMENTALS

2.1 CS1 [CS111]: INTRODUCTION TO COMPUTERS (CORE)

This course introduces numbering systems and problem-solving skills using flow charts and visual basic, and design web page using html. This course can include the following PDC topics:

- Distributed dataflow charts.

Although these two PDC topics are included in the architecture topics in (NSF/IEEE-TCPP , 2012), but to distribute the topics equally without overloading a course, this proposition might be appropriate since these topics require no pre-requisites.

2.2 CS1 [CS143]: INTRODUCTION TO PROBLEM SOLVING AND PROGRAMMING (CORE)

Students are introduced to fundamental data structures including arrays and multidimensional arrays in C language in this course, besides development methods. This course can include the following PDC topics:

- Array language extensions.
- Parallel loops [OpenMP].
- Compiler directives [OpenMP].

Knowledge about these topics at this early stage will make it easier to apply these concepts in the PDC course.

3 PL-PROGRAMMING LANGUAGES

3.1 CS2 [CS243]: OBJECT ORIENTED PROGRAMMING (CORE)

The course is planned to empower the students programming skills in Object Oriented Programming using Java. The focus is on Java Editions and APIs, OOP principles such as inheritance, encapsulation, method overloading and overriding, polymorphism, exception handling, abstract classes and interfaces, GUI and event-driven programming. It is advised in (NSF/IEEE-TCPP , 2012) that it is easier to begin with a sequential model to acquire basic algorithmic problem solving skills. Since the course is already intensive and foundational, it is very difficult to introduce any PDC topics in this one, or it will be on the expense of less content or practice of other core requirements.

3.2 CS2 [CS244]: ADVANCED PROGRAMMING APPLICATIONS (CORE)

The course is planned to introduce the students to various topics to be able to develop applications using state-of-the-art technologies. Topics already include applications of PDC topics such as:

- JDBC and RMI in which client/server architecture is explained.
- Language extensions such as multithreading and synchronization in a shared memory space using Java threads.
- Java web programming such as servlets/JSP, in which the web server multithreaded request/response architecture is explained, servlets life cycle and processes, and the shared environment variables and the local ones to each thread and process.
- Unix/Linux programming environment and tools as a foundation to be used for all PDC topics.
- Python scripting language introduction because it is useful in many configuration scripts for many PDC packages and has its own parallel programming libraries.

This course can include the following PDC topics:

- Processor vector extensions.
- Programming for Hybrid machine models such as CPU and GPU.

3.3 CS2 [CS321]: Systems Programming (Core)

Students learn about the architecture of a hypothetical machine, its assembly language, macro language, programming in assembly language, the structure and design of assemblers, linkers and loaders, and the concepts and theory behind the implementation of high level programming languages. This course can include the following PDC topics:

- SPMD: Understand how SPMD program is written and how it executes.
- SPMD notations: Know the existence of highly threaded data parallel notations (e.g., CUDA, OpenCL), message passing (e.g., MPI), and some others (e.g., Global Arrays, BSP library).
- Concurrency topics.

3.4 CS2 [CS345]: Structure of Programming Languages (Core)

Students are learn about the essentials of imperative programming languages, fundamental issues in language design, overview of programming paradigms, and type systems, and models of execution control: Order of evaluation of sub expressions; conditional execution; iteration; exceptions and exception handling; and parallel composition. This course can include the following PDC topics:

- Semantics and correctness issues.
- Concurrency defects.
- Tools to detect concurrency defects.
- Memory models.

4 DS-Discrete Structures

4.1 CS2 [CS202]: Discrete Structures (Core)

Students learn about the mathematical structures fundamental to computer science. Topics discussed include logic of compound and quantified statements, number theory and methods of proof, sequences and mathematical induction, set theory, counting methods, functions and relations. This course can include the following PDC topics:

- Semantics and correctness issues - Tasks and threads.
- Computation (partitioning & load balancing).
- Computation decomposition strategies.

5 AL-Algorithms and Complexity

5.1 DS/A [CS212]: Data Structures and Algorithms (Core)

Students learn about the organization of data and the algorithms that act upon them. The topics of linked lists, stacks, queues, trees and graphs as well as hashing are introduced. Applications to data structure searching and sorting, memory allocation and file management are included. This course can include the following PDC topics:

- Algorithmic paradigms topics.
- Data distribution, layout, locality, false sharing.

5.2 DS/A [CS311]: Theory of Computation (Core)

This course introduces the fundamental mathematical models of computation. The course presents both inherent capabilities and limitations of these computational models as well as their relationships with formal languages. Topics covered include: Finite automata and regular languages, Deterministic and nondeterministic computations, Context-free grammars, languages, and pushdown automata, Turing machines, recursive and recursively enumerable sets, undesirability, introduction to computability and complexity theory. This course can include the following PDC topics:

- PDC algorithmic problems topics.
- Non-determinism.
- Load balancing.
- Static & dynamic scheduling and mapping.

5.3 DS/A [CS312]: Computing Algorithms (Core)

This course introduces students to the analysis and design of computer algorithms. Topics covered include searching, sorting, selection, graph structures, traversal algorithms, P/NP complete problems, analysis of worst case running time of algorithms, basic properties of randomized algorithms, how to use the major algorithms, data structures and design paradigms, and introduction about the design of parallel algorithms. This course can include the following PDC topics:

- Parallel and distributed models and complexity.
- Performance monitoring tools.
- Performance metrics topics.

6 AR-Architecture and Organization

6.1 Systems [CC216]: Digital Logic Design (Core)

Students learn about design and analysis of digital logic circuits with applications to digital computer. It covers: Number systems (this includes the PDC topics of Floating-point representation), binary arithmetic and codes, logic gates, Boolean algebra and logic simplifications, Design and realization of combinational circuits, Functions of combinational circuits logic: Flip-Flops, analysis design and realization of counters, analysis and realization of shift registers, Computer – aided engineering. This course can include the following PDC topics:

- Flynn's Taxonomy.
- Data vs. control parallelism topics.

6.2 Systems [CC243]: Computer Architecture (Core)

This course exposes the student to computer design & organization. It aims for the student to understand the software/hardware interface, instructions, processor, modules & performance issues. This course can include the following PDC topics:

- Shared vs. distributed memory topics.
- Memory hierarchy topics.
- Performance metrics topics.
- Power consumption.

7 NC-Networking and Communication

7.1 Networking [CC231]: Introduction to Computer Networks (Core)

Students learn computer networks architecture and protocols with special emphasis on the Internet. The course covers the IP protocol stack including application, transport, network, and link layers, use of packet sniffing and protocol analysis tools. This course can include the following PDC topics:

- Gossip: Recognize how all-to-all communication simplifies certain computations.
- Fault tolerance topics.
- Network topologies and diameter of the various graph structures: linear, ring, mesh/torus, tree, hypercube, clique, and crossbar.

7.2 Networking [CS331]: Network Protocols & Programming (Elective)

Students learn network programming using the application-programming interface known as sockets, several design alternatives for client/server applications, developing network-aware applications that involve unicast and multicast communications from the grounds up. This course can include the following PDC topics:

- Performance modeling topics.

8 IM-Information Management

8.1 Algo 2 [IS373]: Database Systems (Core)

Students learn relational Database analysis, design, normalization, and implementation. Concurrent transactions are discussed along with related problems. Storage and indexing of data are discussed. This course already discusses distributed databases and client server approaches. This course can include the following PDC topics:

- Cloud databases and NoSQL environments.
- Distributed transactions.

9 SE-Software Engineering

9.1 SwEngg [SE291]: Introduction to Software Engineering (Core)

This course provides an introduction to software engineering disciplines with emphasis on: software life cycle, System Models, Requirements Specification, Architecture Requirements, Software Design, Rapid Software Development, Verification, Validation and Testing of software. This course can include the following PDC topics:

- Task graphs.

10 PBD-PLATFORM-BASED DEVELOPMENT

10.1 LANG [CS433]: WEB PROGRAMMING (CORE)

This course is a comprehensive introduction of common, primarily open-source, technologies used to develop and maintain server sites on the Web. A variety of client-side and server-side technologies are covered. This course can include the following PDC topics:

- Web search.
- Web services.

10.2 LANG [IS433]: MOBILE COMPUTING APPLICATIONS (ELECTIVE)

This course involves the design and development of mobile application for cell phones, PDAs, and related remote computing devices. After an introduction to mobile computing infrastructures and Mobile Application Software Development tools and Frameworks, the students will be introduced to we-based mobile application architecture using XML, UML and ASP.NET framework. This course can include the following PDC topics:

- Pervasive and Mobile computing.

10.3 LANG [CS343]: GAME PROGRAMMING (ELECTIVE)

The aim of this course is to introduce students to some of the techniques of modern approaches to AI such as probabilistic reasoning, learning paradigms (statistical, reinforcement …etc.). This course can include the following PDC topics:

- Describe uses of SIMD/Vector (same operation on multiple data items), e.g., accelerating graphics for games.
- Cellular automata such as in life game.
- Collaborative computing such as multiplayer games using multiple devices.

10.4 LANG [CS454]: MULTIMEDIA ACQUISITION AND COMMUNICATIONS (ELECTIVE)

This course introduces students to the necessary protocols for supporting multimedia communication through network such as the real time streaming protocol, real time control protocol and real time protocol. The students will be able to acquire and capture voice and video in real time and setting up their network for transmission. Different methods of source coding are also introduced. This course can include the following PDC topics:

- Latency.
- Bandwidth.
- Circuit switching.
- Packet switching.
- Routing.
- Multithreaded streaming video application using Java Media Framework (JMF).

10.5 LANG [CS453]: VIRTUAL ENVIRONMENTS

Basic concepts. Virtual worlds. Hardware and software support. World modeling. Geometric modeling. Light modeling. Topics include: Kinematics and dynamic models; other physical modeling modalities; Multi-sensor data fusion. Anthropomorphic avatars. Animation: modeling languages, scripts, and real-time computer architectures. VE interfaces. This course can include the following PDC topics:

- Social networking.

11 IAS-SECURITY AND INFORMATION ASSURANCE

11.1 SYSTEMS [CS421]: COMPUTER SYSTEM SECURITY

The course is an introduction to computer and network security. The course encompasses the study of security mechanisms for secrecy, integrity, and availability. Topics include basic cryptography and its applications, security in computer networks and distributed systems and control and prevention of viruses and other rogue programs. In addition, hands-on experience will be provided through a series of programming assignments. This course can include the following PDC topics:

- Security and privacy.

12 PD-PARALLEL AND DISTRIBUTED COMPUTING

12.1 DISTSYSTEMS [CS425]: DISTRIBUTED SYSTEMS (ELECTIVE)

This course presents an introduction to distributed systems principles and paradigms. Key principles in the distributed systems arena are presented including: communication, processes, naming, synchronization, consistency and replication, and fault tolerance. In addition, different paradigms are outlined including object-based systems, distributed file systems, and document-based systems. A practical component of the course will allow students to experiment with a simple distributed system including modification of some of its components. This course can include the following PDC topics:

- Distributed memory topics & MPI APIs.
- Parallel Functional/logic languages (Optional).
- Cluster Computing.
- Cloud/grid Computing.
- Peer-to-Peer Computing.

13 ELECTIVE/ADVANCED ALGORITHM DESIGN AND ANALYSIS (CS7)

The following are courses that can be considered algorithms for various applications that can benefit from PDC. They are listed arranged by knowledge area and what they courses currently offer. Then a list of proposed PDC applications for all of them is presented.

13.1 IS: INTELLIGENT SYSTEMS

Algo 2 [CS366]: Introduction to Artificial Intelligence (Core)

Students learn about the history and goals of AI methods. Topics include Representation and search; Knowledge based systems; Logic (Propositional and Predicate) as a representation language; Prolog as an example of an AI language; and an Introduction to Machine Learning.

Algo 2 [CS467]: Advanced Artificial Intelligence (Elective)

This course includes topics such as: Planning, Probabilistic reasoning and Bayes Nets, Learning, Statistical Learning and Reinforcement Learning, Neural Networks.

Algo 2 [CS367]: Robotics Applications (Elective)

This course provides an introduction to the world of robotics and their software programming applications. Topics of interest include the application of microcontrollers and sensors in robotics and from the points of view of hardware and software. The course aims at introducing the students to the mathematical background behind the equations of motion of dynamics and kinematics.

Algo 2 [CS461]: Software Agents (Elective)

A basic introduction to the analysis and design of intelligent agents, software systems which perceive their environment and act in that environment in pursuit of their goals. The module builds on previous AI modules and acts as an introduction to the problems of combining the techniques covered in these modules into a single intelligent agent with broad competence.

13.2 GV: GRAPHICS AND VISUAL COMPUTING

Algo 2 [CS452]: Computer Graphics (Elective)

This course is currently being taught theoretically using the mathematical foundation to understand basic graphics constructs, 2D transformation, and introduction to 3D transformations.

13.3 CN-COMPUTATIONAL SCIENCE

Algo 2 [CS301]: Numerical Methods (Core)

This course provides an introduction to numerical methods and their applications to solve science and engineering problems, such as roots of equations of one variable, systems of linear equations, differentiation, integration, interpolation and regression. In addition, convergence and error analysis of numerical methods is covered, with practical experiences of implementing numerical methods and assessing resulting errors will be acquired through a number of programming assignments.

Algo 2 [CS403]: Optimization Techniques (Elective)

This course introduces Solution of Ordinary differential equations, Optimization Models in Operations Research, Linear and Non-linear models, Simplex Search for Linear Programming, Duality and Sensitivity in Linear Programming, Multi-objective Optimization and Goal Programming, Unconstrained Nonlinear Programming, Selected Methods for Constrained Nonlinear Programming: Lagrange Multiplier Methods and Penalty and Barrier Methods.

These courses already include PDC topics such as:

- Linear systems
- Matrix operations topics.
- Consistency in distributed transactions.

These courses can benefit from the following PDC applications for hand-on experiences by the students:

- Hands-on training of big data analysis such as logistic regression on apache hadoop clusters using apache mahout scalable machine learning and data mining library.
- Parallel Computing toolbox in matlab to apply the various numerical analysis methods in parallel.
- High-Performance and Parallel Computing with R to apply statistical and graphics computing in parallel.
- CUDA and OpenCL libraries and development environments for image processing and scientific computation.
- R also supports GPU computing.

14 GRADUATION PROJECTS

A number of parallel and distributed processing applications were submitted as graduation projects ideas. During fall 2012/Spring 2013, a graduating group connected four laptops in an Apache Hadoop cluster and successfully applied a few data mining operations in Apache Mahout such as recommendation and k-means clustering algorithms. Other ideas include but not limited to: parallel search using Solr over hadoop, Apache UIMA, GPU programming for image processing and scientific computation, multi-agent distributed systems for collaborative computing applications such as games.

15 PROJECT EVALUATION

In CCIT, we are working to apply for the ABET accreditation, and is currently using the ABET forms to measure the student outcomes in each course learning objective, and give a cumulative score to the course offering in each semester. Upon successful integration of the PDC topics in the courses as proposed, we will measure the effectiveness and identify areas of concerns and enhance performance in future offerings. We also use student surveys, faculty reviews, and employers of alumni surveys to evaluate the performance of the programs and courses we offer. We currently don't have compiler, and functional/logic programming courses. Core topics of these areas are discussed in other courses. If the PDC integration is proven to be overloading the existing courses, we can introduce new courses as required. The courses contents mentioned do not cover all topics covered. Currently we are merging the current contents with the curriculum in (ACM/IEEE CS Curriculum RTF, 2012).

16 BUDGET

It will be very motivating to receive a grant to establish an NSF-sponsored Distributed Systems Laboratory (DSL) to train students on real-time hands-on experiences using high capacity servers connected with fast Ethernet and ATM networks. We currently have 5 laboratories in the college; each has about 20 commodities PC with several dual-core, and is expecting to renew one laboratory to i5 processors. Java multithreading and distributed computing, and MPI APIs library, OpenMP, pthreads libraries are already installed. We can install apache hadoop cluster in one or more complete laboratories as we already did that in a cluster of laptops. We do not have dedicated servers' machines to give the students hand-on experiences on installing and managing different types of servers, such as web servers, database servers. We also don't have multi-core tablets and cell phones of different OSs such as Android and iOS instead of using Emulators. We also don't have an In-circuit emulator (ICE) to debug embedded systems projects and FPGA-based hardware emulators. Students purchase their own video game console emulators and devices in graduation projects requiring them. It will greatly help to have some game console emulators available for laboratory work for the various courses & projects.

CCIT also has one TESLA GPU in one supercomputer that was a grant to one professor in the department and is dedicated for research. Having another state-of-the-art GPU of similar or different architecture available to undergraduate laboratories will empower student's skills and expertise. Training faculty staff will require a budget of $1000, purchasing books and materials will require a budget of $500, and a travel budget of around $1200 to attend the conference.

17 BIBLIOGRAPHY

ACM/IEEE CS Curriculum RTF. (2012). *Computer Science Curricula 2013* . IEEE Computer Society.

NSF/IEEE-TCPP . (2012). NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates .