An Analysis of Parallel Computing Curricula: A Case Study

Joel Themmen[1] and Brian J. d'Auriol[2]

Department of Computer Science,
The University of Manitoba,
Winnipeg, Manitoba,
Canada, R3T 2N2

Abstract

An analysis of parallel computing curricula has been presented in this paper. The context for the analysis is that parallel computing is a necessary integral component of both undergraduate and graduate education. The analysis itself takes into account varying factors of curriculum content, the extensiveness of the curriculum as well as departmental influencing factors. A graphical notation called a *curriculum map* has been developed to visually illustrate important points of the application of the analysis. The analysis has been applied to five university computer science curriculums in the United States of America and comments regarding the usefulness of specific analysis components have been made. We expect that more informed decisions regarding parallel curricula update can be realized by, in part, employing the analysis presented in this paper.

Keywords: Parallel Computing, Curriculum Analysis, Educational Components

1 Introduction

Parallelism refers to the occurrence of two or more activities simultaneously within the same interval of time. Typical components in computing which provide for parallelism are parallel machines, parallel languages, parallel algorithms and variations of sequential programming tools and techniques tailored to the parallel computing environment. It is widely acknowledged that it is more difficult to work in the parallel computing environment than it is to work in the sequential computing environment [1]. As a result, working in the parallel computing field demands greater expertise and is more costly.

The constant demand by users for more computing power has spurred the development of faster computers. Manufacturers produce chips that can perform faster on a seemingly continuous basis, with the number of transistors doubling every eighteen months [2]. However, implications of physics with regard to size and complexity of chip circuitry limit the continuing trend of ever increasing single processor performance [3]. Even allowing for future processor performance increases, application programmers need to consider other possible computing areas, like parallel computing, for the desired performance gains. Parallel processing will become more important as single CPU performance increases begin to wane. Also, in order for computer scientists to meet the challenges of the present and the future, experience in the field of parallel computing is absolutely required [3]. For future computer scientists to have sufficient

[1]Email: umthemme@cc.umanitoba.ca
[2]Email: bdauriol@cs.umanitoba.ca

exposure to parallel computing, computer science departments should have significant emphasis on parallel computing in their curricula.

It is noted that many computer science departments include undergraduate parallel course content as a part of some related course, for example operating systems [4]. It is also common for many departments to offer only elective courses in parallel computing. These courses are emphasized as important elective courses in some departments, while in other departments the emphasis may reside in other fields. Many undergraduate programs have few stand alone parallel undergraduate computing courses. There is a trend for graduate level courses to be provided at larger schools that offer parallel computing which have more resources or schools which may be able to gain access to additional resources offered by other high performance centers [4].

It is characteristic that parallel computing curriculum is frequently in a state requiring updating. Since parallel computing is a fast changing discipline, defining an appropriate parallel computing curriculum may be difficult. Schools of different size and academic focus respond to these changes in different ways. The analysis discussed in this paper is expected to assist in decision making processes regarding parallel computing curriculum evaluation and modification.

This paper is organized as follows. Subsequent subsections of the introduction present a brief overview of the proposed analysis and the case studies investigated in this paper. The intent is to provide the reader with an understanding of the proposed analysis techniques. Section 2 details the analysis criteria and includes the known core components of parallel computing, as well as the additional factors we include in the analysis. Section 3 presents five case studies illustrating the application of the analysis, while Section 4 discusses results based on these case studies. Comments regarding the general application of the analysis are given in Section 5 and conclusions are drawn in Section 6.

1.1 The Analysis

The analysis consists of categorizing parallel computing curricula according to the following three types of factors: curriculum content, curricula scope, and department profile. Curriculum content refers to the content of parallel computing courses. A review of the literature [1, 3] notes that parallel computing curriculum can be divided into the following five categories: algorithms, architecture, programming, applications and advances in parallel computing. These are considered the key areas for a robust parallel computing education. The curricula scope of a school refers to the relative size and importance of parallel computing at a particular school. Not all schools can (or should) provide the same curriculum with regards to these areas. As a result, it is expected that academic institutions will have varying degrees of involvement in the teaching of parallel computing. The level of involvement is loosely grouped as follows: Small Scope, Medium Scope and Large Scope. Scoping is subjectively determined by considering, in part, the pre and co-requisite structure, the number of courses offered, departmental interest in parallel computing and overall focus on parallel computing. Many of these factors are also important in the third set of factors which profiles the school's background and resources. Factors in this category include course offerings, departmental focus, department size, department expertise, enrollment, pre and co-requisite structure of the course offerings, the internal structure of the course and the facilities that are provided.

1.2 Case Studies

Several universities from across North America were studied. Complete case studies were done on two schools from both the large and medium scope schools and one school from the small scope category. As well, one other school for both the large and medium scoped schools were informally investigated along with three additional small scoped schools. The choice of the selected schools is based on both the school's reputation for parallel computing and/or familiarity of the school with the authors. Consequently, the selection represents a good application of the analysis techniques. The following schools were chosen for the case studies: Cornell University, University of Illinois, Urbana-Champaign (UIUC), Syracuse University, University of Tennessee at Knoxville (UTK), and Ohio Northern University (ONU); while Purdue (large), University of North Carolina-Chapel Hill (medium) and Wright State University (WSU), University of Manitoba (U of M) and Colgate were used for the informal case studies.

2 Analysis Criteria

2.1 Core Components of Parallel Computing

Parallel computing should be a part of every undergraduate's course load [1, 5]. Furthermore, it is our belief that all graduate students should also be required to study in this area. Consequently, both undergraduate and graduate courses should be offered in this area. Parallel computing is useful as an area with its own merits but, moreover, is also important to many other fields due to the performance increases that it offers. A strong background in parallel computing and parallel application development will be essential for scientists in the near future [3].

Parallel computing is typically comprised of the following five major components:

1. **Algorithms:** Many schools teach algorithms based upon the concepts of the von Neumann model of computing, that is, from a sequential point of view. It is interesting to note that we teach — and have been taught — to approach problems in this manner . Yet as human beings, we process problems on a parallel level [6]. For example, in a secretarial pool, a large document will often be distributed among many secretaries in order to speed overall delivery. This process of concentrating on parallel solutions should be encouraged and nurtured.

   It is important that students not think of *converting* a problem into a parallel domain problem but *develop skills* to realize when a solution problem is intrinsically part of the parallel computing domain [1]. For example, matrix multiplication is taught in a sequential fashion usually in a course in linear algebra. However, this is naturally a parallel problem, for example, rows are multiplied by columns distinct of each other. Instruction in parallel algorithms will lead students to such realizations.

2. **Architecture:** Due to the greater diversity of parallel architectures as well as the greater impact that parallel architectures have on parallel programming, knowledge of parallel architectures becomes essential. When studying parallel computing, architectural concerns such as instruction level code, communication, cache coherency issues and interconnection problems are new and often troublesome areas to the undergraduate student [1]. Many

standard undergraduate architecture courses (sequentially based) do not deal with these topics. It is important to recognize this omission from most undergraduate architecture courses. Architecture courses should cover both sequential and parallel architecture.

3. **Programming:** Programming in a parallel computer environment has many similarities to programming in a sequential computer environment. Given knowledge of parallel algorithms, it is possible to augment sequential programming knowledge so that programming in a parallel, computing environment is not an entirely new experience. However, without some exposure to parallel programming, a student will have little confidence in their ability to program in parallel computing environments. The particular language chosen is less important than the fact the some parallel language issues have been taught at the undergraduate level [7].

4. **Applications:** Applications of parallel computing can be considered as the synthesis of some parallel algorithm(s) and some parallel programming. Applications will eventually position parallel computing to a ubiquitous educational concern [3]. As applications demand computing power that can only be resolved by parallel systems, parallel application development skills will become extremely valuable [3].

   Many scientists (specifically, in this case, non-computer scientists) work with problems that could benefit from parallel computing. However, these scientists are often specialists in their fields, not computer science, and it is important to note that their concerns may be different. Parallel computing should be readily available to these scientists as a tool to aid in their work. Providing parallel language programming and application development training would be expected to be of great benefit to scientists in many fields.

5. **Advances in Parallel Computing:** Parallel computing is a dynamic field, that is, significant changes and advancement in the field occur over relatively short periods of time. As a result, it is common that an academic institution will have one or more courses to specifically deal with the latest advances in parallel computing. It seems typical that this course is offered at the graduate level (refer to the case studies in Section 3). Once a solid foundation has been established at the undergraduate level, an 'advances in parallel computing' course can introduce cutting edge topics in selected fields. This course is very important to the research of parallel computing at a school. Often, it is exposure to the "hot" topics in parallel computing that interest and captivate graduate students in the area of parallel computing.

A review of the literature indicates the general opinion that these five categories are germane to a breadth wise comprehension of parallel computing [8, 4, 9, 7, 10, 11, 12, 5]. However, consensus does not exist on the relative importance of the categories. Any institution can offer any subset of these five categories, including none or all five.

2.2 Additional Analysis Categories

No two academic institutions are exactly alike, furthermore, they should not be. The uniqueness of the institution, with regard to parallel computing, will be evident in its chosen approach to parallel computing curriculum. Any analysis performed on the curriculum of any school must take into account the intrinsic differences of the schools.

The curricula scope of a school refers to the relative size and importance of parallel computing at a particular school. We loosely group schools into three scope categories based on and in reference to parallel computing:

1. **Small Scope:** Small scope refers to an academic program in which there are few or no courses offered at the undergraduate or graduate level that specifically focus on parallel computing. Despite the fact that no courses may specifically be parallel in nature, a blending of parallel computing oriented material and other core courses (notably operating systems, computer architecture, distributed computing and Topics in Computer Science) is often found. Specific parallel computing courses may not be offered every year and as a result often do not require pre-requisites for entry. The main goal may be to inform the student that parallel computing exists and to cover the basics of the field.

2. **Medium Scope:** A school with medium scope will likely have several courses that are specifically focused on parallel computing. As well, it is likely that the approach used to teach parallel computing will be much more specific in regards to the five core components of parallel computing. A medium scoped school may not have a separate course for each of the core components but will likely have courses that are entirely parallel in nature and that are comprised by some combination of the core components. There begins to be a separation between the distinct core elements that compose the field of parallel computing. There is higher likelihood that pre and co-requisites exist and are enforced.

3. **Large Scope:** A school that has large scope will offer many parallel computing courses at both the undergraduate and graduate levels. There will exist specific courses to deal with the separate branches of the field. It is expected that the courses will have a relatively high number of students enrolled. Furthermore, the curriculum should be widely recognized and supported within the department. There is usually a strong hierarchy that is enforced between the courses that comprise the undergraduate/graduate course offerings.

With respect to the curricula content, it remains important that the five core components are covered to some degree. A preliminary discussion of the scope categories appears in [13].

The final set of factors that are considered in the analysis constitute a profile of the department. Departmental issues can greatly determine and influence the type of parallel computing education regardless of any curriculum specifications. We consider the following categories to be indicative of the departmental profile:

1. **The number of courses in parallel computing:** The number of courses offered at a school in parallel computing is taken as a measurement of the ability of that department to commit educational resources to parallel computing. A low number of courses will be interpreted as indicating that parallel computing is not emphasized while a higher number would indicate greater emphasis.

2. **Intensity:** *Intensity* refers to the amount of effort that a student must put into a course. A course which requires significant work from the student may be indicative of a stronger emphasis on parallel computing.

3. **Parallel Percentage of a Course:** The percentage of a course that deals with parallel computing can indicate the emphasis on parallel computing within both that course and

the department. Strong departmental or professorial emphasis on parallel computing can indicate importance of parallel computing within the department. We subjectively measure the percentage using a four point scale with one (1) indicating little parallel content, two (2) indicating between 10% and 25% parallel content, three (3) indicating 25% to 50% parallel content and four (4) indicating over 50% parallel content.

4. **The enrollment in parallel courses:** High attendance in parallel computing courses will be taken to be indicate that there is extensive promotion of parallel computing within that department. This can be due to both departmental and student influence, e.g. word-of-mouth recommendations from other students. Note that enrollment may need to be adjusted based on relative scope. Consequently, percentage enrollment may be substituted for comparative purposes.

5. **The pre and co-requisite structure of the courses:** A tight pre and co-requisite structure within a department is taken to indicate that the students are expected to have substantial background in either parallel computing or strongly related material. It is expected that a school that enforces on pre and co-requisites for many courses will offer courses to satisfy those requirements. Without establishing parallel computing prerequisites to courses, the department is allowing the student access to parallel computing courses without ensuring prior parallel computing knowledge. Therefore the possibility exists that the individual student may not have studied parallel computing at the undergraduate level. In some cases, consent-based authorization for entry into a course is used either in conjunction with or alternative to pre and co-requisites.

6. **The faculty size:** The relative size of a department is indicative of a higher probability of resource commitment to parallel computing. A larger department will likely have more resources and funding to allocate to parallel computing. Since parallel computing is a relatively expensive field, we will interpret that a larger school will be better able to fund adequate parallel computing courses.

7. **The percentage of faculty that has parallel/high-performance computing as a major research topic:** This factor is used to indicate the department's internal focus. If parallel computing is an emphasized area of research within a department then it is likely that parallel computing issues will be an important component of the department curriculum. It is stressed that the faculty members do not have to have parallel/high-performance computing as the primary research area nor must it be the area originally studied.

8. **Departmental Orientation:** The orientation of a department may be spread across all the core components or it may be focused in some subset of those areas. A school may choose to concentrate on some subset of the core components. This departmental focus can greatly determine how both parallel computing and non-parallel computing courses are taught.

9. **Facilities:** Good parallel computing facilities may be indicative of a strong emphasis on parallel computing within a department. Clearly, larger departments have advantages in this category however, smaller schools may be able to obtain access to external facilities (for example, see Section 3). A hands on approach to parallel computing is valuable and can be indicative of the relative importance of parallel computing at that school.

3 Case Studies

Some case studies are accompanied by a *Curriculum Map* which diagrams many of the required analysis factors. The format of the map consists of courses that are offered at a particular school that contain some degree of parallel computing content. The numbering scale used is described in Section 2.2 in the discussion on parallel percentage of a course. Pre-requisites for a course are indicated by an incoming line with a single arrow. A small circle may be used to indicate that more than one pre-requisite is necessary. Co-requisites are represented by lines that have directed arrows at both ends. The blank boxes indicate that pre or co-requisites are required but they are not parallel in nature. Particular field content areas are indicated on each map. Information regarding curriculum and department details for the case studies have been obtained from their respective course calendars and/or their world wide web sites.

We believe that Cornell is an ideal example of an academic institution that has large scope with reference to parallel computing. It covers each of the five categories in substantial detail. We then compare the model to the other schools that we believe exemplify the three scope categories we have developed. The comparison with respect to analysis categories will be somewhat brief since the purpose is to determine the validity of the analysis categories. Subsequently, we will make recommendations on how the analysis categories could be used to provide information that could be useful in modifying curricula for the maximization of parallel computing within said curricula. The analysis does not include *intensity* due to the difficulty in measuring without direct input from staff and students for any particular school.

3.1 Cornell

Cornell University, located in Ithaca, New York, is considered to be one the finest super computing institutions in the world. Cornell has excellent facilities, hardware, staff and students. Cornell is identified as one of the main super computing facilities in the United States of America [14]. Cornell has also been an outstanding contributor to other schools by providing super computing resources to other schools. Cornell also provides many other services as well (example the CTC Virtual Workshop).

For the purposes of this paper, we consider Cornell as an appropriate academic program exemplifying both a large scoped school, as well as the previously discussed curriculum content.

From Figure 1, we see that Cornell covers the five main areas in great detail. A separate course exists for parallel applications (422) at the undergraduate level. Also at the undergraduate level, the operating systems courses are heavily 'flavored' with parallel and distributed computing topics. At a graduate level, separate courses for parallel architecture (516), concurrent programming (613), compiler construction for parallel computers (612), advances in parallel systems (717) all exist. As well, many other courses involve parallel computing to a large extent (for example, 631, multi-media systems, has a focus on multi-media systems on parallel architectures). The strong pre-requisite structure indicates that students are expected to have a substantial level of expertise before admittance into a given course is permitted.

Cornell University currently lists almost 40 faculty members with 12 indicating that parallel computing constitutes some portion of their research. This represents close to one third of the staff. This will help influence many other courses to have a parallel 'flavor' as well as to shape

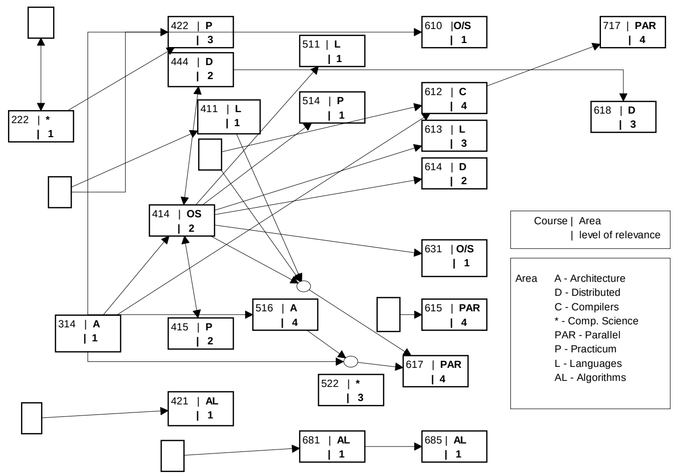

Figure 1: **Cornell University Undergraduate/Graduate parallel course map.**

the general direction of the faculty. Department focus at Cornell appears to broad based. The department has sufficient resources to offer a depth wise education in all of the co re components. We see courses offered in all of the core component areas

Cornell student representation in parallel courses is substantial. Approximately 40% of all course work done at the graduate level will deal directly with parallel processing according to recent student statistics [15].

3.2 University of Illinois, Urbana-Champaign

The University of Illinois, Urbana-Champaign (UIUC) is another of the designated high performance super computing centers in the United States of America [14]. UIUC is another example of a large scoped school. UIUC provides a variety of parallel hardware to the student. Resources are comprehensive. As expected, due to its large scope nature, UIUC's parallel curriculum covers many fields in great detail. UIUC provides theoretical, application and programming domains for the student.

UIUC also offers many parallel computing courses. From Figure 2, at least four courses are offered where parallel computing is the core subject covered. As well, there exists an additional six courses where significant sections of the course relate to parallel computing. Course number 320 (also offered as Computer Engineering 302) provide instruction in parallel programming as well as covering the basic computer architectures that will be used for programming. 320 (302) is aimed at scientists and engineers and will cover the applications area as well as the programming aspect of parallel computing. Part of the contents of 370 includes parallel algorithms and their analysis. 373 advances these topics into more depth. There are the courses 433 and 454 which are entirely devoted to the study of parallel algorithms. Several 'topics' courses are offered at both the undergraduate and graduate levels at UIUC.

A strong pre-requisite link exists throughout the curriculum. It is noted that some of the

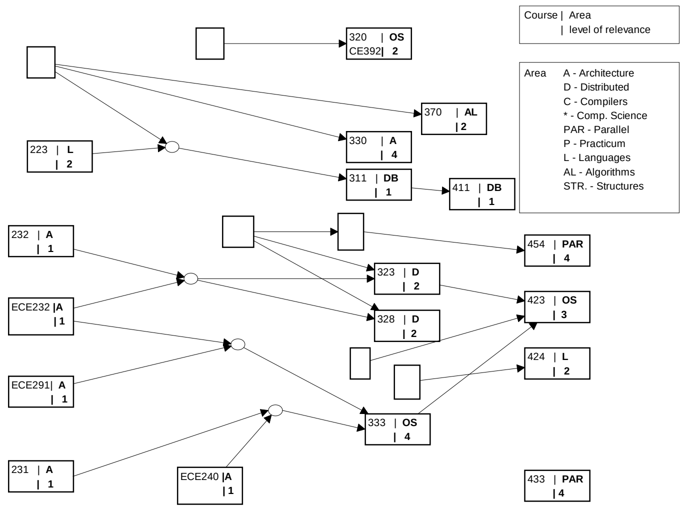

Figure 2: **UIUC Undergraduate/Graduate parallel course map.**

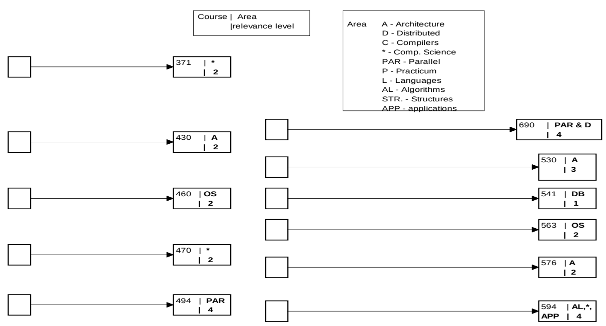

Figure 3: **Tennessee Undergraduate/Graduate parallel course map.**

graduate parallel courses (423 for example requires 323 and 333) require previous work done in the parallel computing field as an undergraduate. This is indicative of a tightly coupled pre-requisite that has a specific focus on parallel processing. It is demanded that the incoming graduate students already have a certain skill set before taking certain graduate courses. This implies that the curriculum reflects the concept that parallel processing should be a part of the student's academic lives before they are in graduate school.

It is noted that 16 of 43 faculty members list parallel processing as an area of research, indicative of a high level of research in parallel computing. Once again, this indicates that parallel computing is an important part of the department at UIUC.

3.3 University of Tennessee

The University of Tennessee at Knoxville (UTK) is classified as a school with medium scope. From the course offerings (see Figure 3), we note that parallel computing has more of a presence

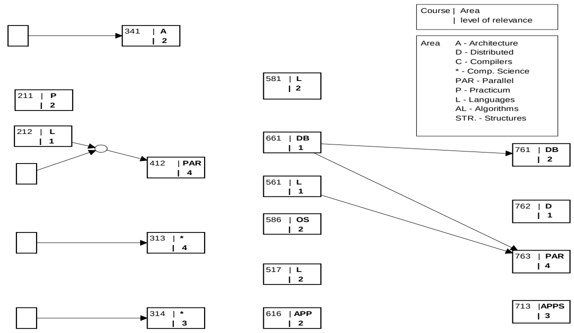

Figure 4: **Syracuse Undergraduate/Graduate parallel course map.**

at the graduate level than the undergraduate level. There is a strong emphasis on computational science at UTK. At the graduate level 530 covers many architectural issues of both serial and parallel computers. 430, 460 and 470 deal with hardware, parallel computation and the application of parallel computers to scientific problems, respectively, while 494 deals entirely with parallel computing. From Figure 3, it is noted that most of the parallel courses at UTK do not require parallel courses as pre-requisites.

UTK covers applications in great detail. Architecture is covered both at the undergraduate level (430) and in even greater detail at the graduate level. Algorithm analysis is covered in a course listed jointly with the math department (371). Topics courses are covered by several iterations of the 594 course (594 can cover different topics, year dependent).

17 of 38 faculty members list parallel processing as an area of research. As well, the university is also closely affiliated with Oak Ridge Institute, a research center that conducts research in the parallel computing field (amongst others).

UTK offers several courses in the computational science field as well. Typically these are very heavily based on parallel computing and subsequent applications of parallel computing (for example, 594). A total of five courses at the undergraduate and six courses at the graduate level, indicate the level of interest of parallel computing at UTK. The emphasis at the graduate level is expected with the high proportion of faculty that express interest in the parallel computing field.

UTK has several courses that are strictly parallel in nature (494, 594, 690). When the parallel component is not the only component (examples, 430, 530, 563), the emphasis is high thereby bringing parallel computing into many classes.

UTK facilities include a CM-5 (32 processor), a MP-2 (16,000 processor), a iPSC860, a Sequent Symmetry and others.

In summary we see that our analysis model is reasonable with respect to UTK's existing parallel computing curriculum.

3.4 Syracuse University

Syracuse is another good example of an academic institution that has medium scope. There are dedicated courses at both the graduate and undergraduate level (see Figure 4) with a strong emphasis on computational science. At the undergraduate level course numbers 211, 313 and 314 and at the graduate level course numbers 616 and 730 are based primarily in the field of computational science. Course numbers 412 (undergraduate) combines parallel algorithms, architecture and programming into one course. 761 combines databases with multiple processors while 762 and 763, respectively, cover distributed computing and parallel processing systems. There are also several other courses (for example 211 and 586) that combine parallel computing with other related courses.

Graduate parallel course entry requirements are often graduate standing and undergraduate programming experience and not specific undergraduate parallel courses.

Of Syracuse's 32 faculty members, 13 indicate research interests parallel computing . Consequently, there is a high interest in parallel computing in a large faculty. It is noted that faculty member, Geoffrey Fox, is largely responsible for the heavy emphasis on scientific computation at Syracuse. NPAC is co-ordinated and run by Dr. Fox. This strong emphasis on scientific parallel application development makes parallel computing very visible at Syracuse both at the undergraduate and graduate levels.

Syracuse has very good facilities, as would be expected from a school with a focus in computational science. The parallel computing hardware at Syracuse includes a CM5, an iPSC, an SP2 and a multiprocessor SGI Challenge L with distributed computing being supplied by a SUN Ultra SPARC cluster, DEC Alpha cluster and a QuadPC cluster.

Our analysis model is reasonable with respect the existing parallel computing curriculum offered at Syracuse University

3.5 Ohio Northern University

Ohio Northern University (ONU), located in Ada, Ohio, is classified as a small scoped school. As a small college, ONU does not have many of the resources that a larger university would have. It has no parallel machines on campus. Students at ONU, as part of the high performance computer infrastructure in Ohio due in large part to the Ohio Super computing Center, do have access to the following high performance computing facilities: a CRAY Y-MP8/8128, a CRAY T3D, a Convex SPP12200/EX-16, an IBM POWER parallel System - SP2, and a SGI Power CHALLENGE, and a Quantum Chemistry Engine (QCE). ONU offers only one strictly parallel course, 442 and several other courses that contain some parallel content (236 and 436, respectively an architecture and OS course). In 442, students are exposed to parallel algorithms, parallel programming, parallel architectures and parallel applications [16].

4 Results

Based on the case studies presented in the previous section, the following characteristics of the scope categories are identified.

1. **Large Scoped Schools**
   - strong co and pre-requisite structure to course content,
   - high enrollment in parallel computing courses,
   - percentage of course content that is parallel in nature is high,
   - faculty research in parallel computing is quite strong, and
   - orientation is both theoretical and practical
2. **Medium Scoped Schools**
   - co and pre-requisite structure of parallel course content is not as stringent as with the high scoped school,
   - parallel computing course attendance is less than at a highly scoped school for most courses although certain courses will have quite high attendance,
   - percentage of course content that is parallel in nature is high for some courses but parallel content now shares some courses with other topics,
   - faculty research in parallel computing is quite strong, and
   - orientation is both theoretical and practical
3. **Small Scoped Schools:** Small scoped schools may have extensive variability in each analysis category. A majority of schools fall into this scope category. ONU is an excellent example of a school that, despite moderate on site facilities, provides an excellent education in parallel computing. In small scoped schools, the number of faculty may greatly vary, the areas of research may greatly vary. The emphasis of parallel computing within a given department is not always related to the aforementioned two factors. A small scoped school may also be a school of considerable size and resources that does not emphasize parallel computing. Consequently, it is difficult to establish a set of guidelines for a small scoped school.

We see that we can choose the appropriate scope model for a given institution. A key factor for all schools is to provide exposure to parallel computing through parallel computing courses and other related courses as early as possible in the undergraduate curriculum. It is felt that exposure to parallel computing can start as early as the freshman year [1,2,3,5,6,13]. Relevant aspects of parallel computing could and should be brought into the mainstream curriculum. In this manner, a school classified as small scoped can cover the same topics as a large scoped school, just to different depths. The small scoped school need not provide separate courses but must still ensure breadth wise coverage through existing courses with respect to parallel computing.

Parallel computing facilities with abundant resources seem quite willing to share these resources (see Cornell Theory Center and Texas at Austin, for example) with schools that cannot provide the resources for their students directly on campus. As well, the advent of network based parallel computers will allow schools with local area networks to simulate parallel machines.

5 Application of the Curricula Analysis

The analysis presented in earlier sections can be applied to an academic institution in order to assist with the determination or evaluation of their parallel computing curriculum. It is likely that, if a school has large scope, then it will significantly cover all five categories. Graduate and undergraduate courses will be extensive and will cover parallel computing in a depth wise manner. We expect research in parallel computing to be very important at these schools. From our observations we also note that a medium scoped school will likewise cover the five categories, although it may not have a dedicated course for each category (e.g. UTK). We are likely to see more graduate courses than undergraduate courses and a fairly strong research component as well. At a small scoped school we begin to see significant segments of parallel computing not addressed. Typically, research areas at the school do not involve parallel computing to a large extent.

Our case studies have included schools that have ample resources, both in terms of faculty and equipment, however, many schools do not have the capability to focus on one aspect of one department to the same extent as large and medium scoped schools. Thus, a large number of schools fall into the small scope. This does not mean that a comprehensive approach to parallel computing curriculum cannot occur. We believe that the analysis procedure discussed in this paper can accommodate the variability of small scope schools.

6 Conclusion

An analysis of parallel computing curricula has been presented in this paper. The context for the analysis is that parallel computing is a necessary integral component of both undergraduate and graduate education and thus, significant emphasis should be placed on including such instruction in a curriculum. Comments supporting this viewpoint have been presented. The analysis itself takes into account varying factors of curriculum content, the extensiveness of the curriculum as well as departmental influencing factors. A graphical notation called a *curriculum map* has been developed to visually illustrate important points of the application of the analysis. The analysis has been applied to five university computer science curriculums in the United States of America and comments regarding the usefulness of specific analysis components have been made.

The analysis consists of evaluating a curriculum in three areas corresponding to the core educational content of parallel computing, the scope and departmental influences, respectively. It has been established that, of these, the first two are largely significant and orthogonally address the important issues of a curriculum. The third area of evaluation assists in qualifying the profile of the department.

Procedurally, applying the analysis to a particular curriculum for *evaluation* purposes requires first establishing the scope of the curriculum as well as detailed course content. Furthermore, based on departmental concerns, the areas of parallel computing that are to be further development must be identified. We expect that more informed decisions regarding curricula update can be realized by, in part, employing the analysis presented in this paper.

References

[1] Marcin Paprzycki and Janus Zalewski. Teaching parallel computing without a separate course. In *Proceedings of the Conference on Parallel Computing for Undergraduates*, Colgate University, Hamilton, New York, June 18-19 1994. Edited by Chris Nevison.

[2] Robert L. Hummel. Eight ways to the future. *BYTE*, 21:85–88, December 1996. Issue No. 12.

[3] Geoffrey Fox, Roy D. Williams, and Paul C. Messina. *Parallel Computing Works*. Morgan Kauffman Publishers, Inc., San Francisco, 1994.

[4] Russ Miller. The status of parallel computing. document available via FTP://ftp.cs.buffalo.edu/users/miller, 1995.

[5] Donald Johnson, David Kotz, and Fillia Makedon. Teaching parallel computing to freshmen. In *Proceedings of the Conference on Parallel Computing for Undergraduates*, Colgate University, Hamilton, New York, June 18-19 1994.

[6] Bart Happel and Jacob Murre. Design and evolution of modular neural network architectures. *Neural Networks*, 7:985–1004, 1994.

[7] Marcin Paprzyki, Ryszard Wasniowski, and Janusz Zalewski. Parallel and distributed computing education: A software engineering approach. Rosalind L. Ibrahim, editor, In *Proceedings of 8th CSEE '95 Conference*, pages 187–204, New Orleans, La, USA, March 29-April 1 1995. Springer-Verlag.

[8] P. Takis Metaxas. How easy is to think parallel? a new approach for introducing parallelism. In *Proceedings of the Conference on Parallel Computing for Undergraduates*, Colgate University, Hamilton, New York, June 18-19 1994.

[9] Chris Nevison. Parallel computing in the undergraduate curriculum. Prepared slides for guest lecture at OSC, Columbus, Ohio, available at URL=http://www.osc.edu/pcurric/UGPARC.PPT.pdf, May 17 1996.

[10] Arnold L. Rosenberg. Thoughts on parallelism and concurrency in computing curriculum. In *Proceedings of the Wellesley Forum on Parallel Computing Curricula*, Wellesley College, Wellesley, Mass., USA, March-April1 1995.

[11] Marcin Paprzycki and Janusz Zalewski. Introduction to parallel computing. In *Proceedings of the Fifth Annual South Central College Computing Conference*, George A. Benjamin (Ed.), Amarillo, Texas, USA, April 15-16 1994.

[12] Pavel Tvrdik. Theoretical background of parallel computing for undergraduate. In *Proceedings of the Conference on Parallel Computing for Undergraduates Students*, Colgate University, Hamilton, New York, June 18-19 1994.

[13] Brian d'Auriol. Effective teaching of practical parallel programning skills. Prepared slides for a talk at the OSC, Columbus, Ohio, available at URL=http://www.osc.edu/pcurric/wsu.596.ps, May 17 1996.

[14] Marcin Paprzycki. Incorporating high performance computers into mathematics curriculum. In *Proceedings of the Fifth Annual International Conference on Technology in College Mathematics*, William Rainey Harper College and Northern Illinois University, Rosemont, Illinois, USA, November 12-15 1992.

[15] Computer science courses and enrollment statistics. Cornell Computer Science Department, Fall 95, Spring 96. Electronic Mail Correspondence.

[16] HPCWire Private, liberal arts colleges now teach parallel computing. Alan Beck (Ed.), Tabor Griffin Communications, 8445 Camino Santa Fe, San Diego, CA 92121 (http://www.tgc.com/HPCwire.html), March 28, 1997.