

INTEGRATING DATA-INTENSIVE SCALABLE COMPUTING INTO THE COMPUTER SCIENCE CURRICULUM

Alice E. Fischer
University of New Haven
West Haven, CT
AFischer@newhaven.edu

John P. Dougherty
Haverford College
Haverford, PA
jd@cs.haverford.edu

Gregory Kesden
Carnegie Mellon University
Pittsburgh, PA
gkesden@cs.cmu.edu

Mark E. Hoffman (Moderator)
Quinnipiac University
Hamden, CT
mark.hoffman@quinnipiac.edu

ABSTRACT

The participants in this panel all attended the Data-Intensive Scalable Computing in Education (DISC-E) workshop at the University of Washington in July 2008¹. Data-Intensive Scalable Computing (DISC) focuses on efficient organization, storage, and processing of massive data sets stored in distributed file systems. For organizations like Google, Amazon, Facebook, and Yahoo! where massive data sets are an everyday issue, knowledge of the related issues and the ability to solve related problems is critical when considering new employees. The goal of the workshop was to “inspire the development of new coursework in large-scale data-intensive application design and cluster computing.” At the workshop, the participants discussed various strategies for integrating DISC into the computer science curriculum and courses at all levels: from introductory computing to upper-level and capstone courses. The participants in this panel will discuss how they see DISC, and parallelism in general, as part of the computer science curriculum. Participants will discuss how they have used what they learned at the workshop in courses and how they foresee using it in the future.

POSITION STATEMENTS:

Alice E. Fischer, University of New Haven

DISC will be presented as a form of parallel computation using independent

1. <http://www.cs.washington.edu/homes/ak/clusterworkshop/>

processes running on a cluster of computers, directed and organized centrally. Implications of being data-centric and scalable will be discussed, along with the types of problems where DISC is an efficient and appropriate solution. Google's map-reduce language and its public-domain analog, Hadoop, will be introduced as a platform for writing DISC applications. A Hadoop program consists of one or more cycles, each comprised of a parallel "map" phase followed by a central "reduce" phase.

Computer Science at the University of New Haven is within the school of engineering. Our courses serve CS, mathematics, and computer engineering majors, and minors, many from Criminal Justice. DISC computing has been introduced into two courses, both at the senior level. Our courses at all levels attempt to blend theory and hands-on practice. In the Structure of Languages course, DISC and Hadoop have been introduced as an important new computing paradigm, and short programs have been written. In the senior-level Computer Ethics course, the easy availability of cheap parallel computation on massive amounts of data is discussed as a new way to do data mining and a new threat to personal security and privacy.

John P. Dougherty, Haverford College

Computer Science at Haverford College provides a curriculum that targets three student demographics: computer science majors, concentrators and minors; concentrators in scientific computing; and non-majors/budding majors in computer science, often via service courses (CS0). DISC is mentioned in CS0, but is found in more substantive detail in the first two demographics.

For majors/concentrators/minors, DISC is highlighted in CS1 as an application of the functional programming model. The course begins with a functional approach to algorithm design and implementation, eventually leading to enumerate and test solutions to such NP-complete classics as the clique problem and graph coloring. Students are expected to express their solutions functionally, and we can state with persuasive examples in DISC that such a functional approach can be more effective cognitively than other approaches (e.g., imperative, object-oriented). At this level we can state that "finding the answer from an extremely large pool" is one way map-reduce can be utilized. We do not dive into more detail than that.

DISC is also noted in the following other courses for majors/concentrators/minors:

- *programming languages*: when a more formal and complete treatment of functional programming is covered
- *computer architecture*: to re-emphasize the importance of considering proximity of data to processing to increase DISC performance
- *software engineering*: inject a curricular module to discuss how to design solutions that utilize the map-reduce paradigm for web-based processing, most likely a simple example using search
- *high-performance scientific computing (HPSC)*: it is noted that while many classic scientific problems are solved with more conventional methods (e.g., matrix operations), DISC is becoming an alternative for data-mining type of algorithms used in biology and genomics

The last course from the list above, HPSC, is the location where students planning to concentrate in scientific computing get the most direct experience with DISC. At this

point our curriculum is not ready for a dedicated elective in DISC, but that is a topic for future consideration. Students are also welcome to select DISC as a topic for a senior capstone thesis and project, part of the graduation requirements for our students at Haverford College.

Gregory Kesden, Carnegie Mellon University

The School of Computer Science at Carnegie Mellon offers a diverse collection of majors, minors, and concentrations in the computing sciences. Our courses, at all levels and across areas, emphasize *Computational Thinking*, as well as depth in domain. Our systems courses have a long established tradition of blending a study of the over-arching principles of computer systems and system software with ground-up, large-scale implementation, such as of operating systems, network protocol stack implementations, and distributed databases.

The intellectual building blocks for reasoning about problem-solving under the Map-Reduce paradigm have long been introduced in our core programming language course, which is steeped in functional programming. The first use of DISC in our courses came in the Fall 2007 special offering of a masters-level course in *Internet Search Technologies* taught by collaborators in industry. A discussion of the design, architecture, capabilities, and limits of cloud computing as realized through the Hadoop framework and the Map-Reduce paradigm was introduced into our senior-level *Distributed Systems* course in Spring 2008. The Fall 2008 inaugural offering of *Fundamentals of Systems*, also at the senior-level, included significant discussion and project work in DISC and made extensive use of Hadoop and computing provided both by partners in industry and by campus and personal computers.

Moving forward, we anticipated expanding coverage of DISC in each of our *Distributed Systems* and *Fundamentals of Systems* courses. A special topics or elective course, exclusively in DISC, is being discussed for Spring 2010. This course, in contrast to our present offerings, would have the opportunity to offer truly in-depth coverage of algorithms and idioms for DISC in several application domains and to explore areas of active research. Opportunities for introducing parallel thinking, including DISC, are actively being explored for our first-year courses, which are taught to both majors and non-majors. We believe that parallel thinking and large-scale data processing, at the heart of DISC, are interesting tools for exploring *Computational Thinking* at all levels of the curriculum.

Mark E. Hoffman, Quinnipiac University

At Quinnipiac University we offer a major and minor in Computer Science. At this early stage we have integrated DISC into the senior project where one student is currently working on a project to implement a cluster as a Hadoop test bed. The test bed will be used to go beyond the algorithms implemented at the workshop (word count, index, and page rank) to implement a new algorithm on a large data set. During the spring 2009 instance of our data structures (CS2) course we plan to explore standard examples of word count and index on relatively small data sets then consider how new strategies like DISC are necessary to handle massively-large data sets. We are also considering a future project to implement a single computer virtual cluster to quick installation and experimentation with Hadoop.

ABOUT THE PANELISTS:

Alice E. Fischer is a professor of Computer Science and chair of the department of Electrical & Computer Engineering and Computer Science at the University of New Haven. She earned her Ph.D. from Harvard University in 1985 and has taught at UNH since 1982. Her interests are in software engineering, computer languages, and the impact of computing technology on society. Dr. Fischer has published two textbooks in these areas.

John P. Dougherty is presently an assistant professor of Computer Science at Haverford College where he investigates computing education issues, including the composition of the introductory course for undergraduates, outreach to K-12 programs in computing, as well as materials in computing for non-computing undergraduates. He also studies parallel scientific computing, including dependability/performance issues using cluster approaches. Finally, he is working to establish connections between information technology and society, especially accessible computing.

Gregory Kesden is an associate teaching professor in the Computer Science Department and the Director of Undergraduate Laboratories for the School of Computer Science at Carnegie Mellon, where he has been a member of the faculty since 1999. He specializes in the management of resources in concurrent and hazardous environments, including areas of distributed systems, operating systems, and networks. He regularly teaches upper-division, master's-level, and introductory courses. He coaches Carnegie Mellon's programming teams and facilitates other student activities. He completed both his undergraduate and graduate work in computer science at Clemson University in South Carolina.

Mark Hoffman is a Professor of Computer Science, and has been at Quinnipiac University since 2001. He has taught courses in Computer Architecture and Organization, Operating Systems, Data Structures, and Algorithms, among others. Dr. Hoffman earned his Ph.D. from Polytechnic University in Brooklyn, NY in 2001. Prior to joining Quinnipiac University he worked for 16 years in IT, after spending 14 years teaching high school mathematics and science. Dr. Hoffman's research interests Computer Science education and curriculum, computer literacy, and the impact of the Internet and computing technology on society.