Adopting Heterogeneous Computing Modules: Experiences from a ToUCH Summer Workshop

David P. Bunde*, Kishwar Ahmed[xii], Sridevi Ayloo¶, Tisha Brown-Gaines‖, Joel Fuentes††, Vishwesh Jatala[xi], Ruth Kurniawati‡‡, Işıl Öz[x], Apan Qasem†, Philip J. Schielke‡, Mary C. Tedeschi§, Thomas Y. Yeh**
*Dept. Computer Science, Knox College, Galesburg, IL, USA
[xii]Dept. Electrical Engineering and Computer Science, The University of Toledo, Toledo, OH, USA
¶§Dept. Computer Systems Technology, CityTech, City University of New York, Brooklyn, NY, USA
‖Computer Science Dept., Belmont University, Nashville, TN, USA
††Computer Science Dept., Universidad del Bío-Bío, Chillán, Chile
[xi]Dept. of EECS, Indian Institute of Technology Bhilai, India
‡‡Computer and Information Science Dept., Westfield State University, Westfield, MA, USA
[x]Computer Engineering Dept., Izmir Institute of Technology, Izmir, Turkey
†Dept. Computer Science, Texas State University, San Marcos, TX, USA
‡Dept. Computer Science, Concordia University Texas, Austin, TX, USA
§Dept. Computer Science, St. John's University, New York, NY, USA
§Dept. Information Systems and Statistics, Baruch College, New York, NY, USA
**Computer Science Dept., Pomona College, Claremont, CA, USA
Email: *dbunde@knox.edu, [xii]kishwar.ahmed@utoledo.edu, ¶sayloo@citytech.cuny.edu, ‖tisha.gaines@belmont.edu,
††jfuentes@ubiobio.cl, [xi]vishwesh@iitbhilai.ac.in, ‡‡rkurniawati@westfield.ma.edu, [x]isiloz@iyte.edu.tr,
†apan@txstate.edu, ‡philip.schielke@concordia.edu, §tedeschm@stjohns.edu, **thomas.yeh@pomona.edu

***Abstract*—We present efforts to encourage the adoption of modules for teaching heterogeneous parallel computing through a faculty development workshop. The workshop was held remotely using a novel format to exploit the advantages of a virtual format and mitigate its disadvantages. Adoption at a wide variety of institutions showed module effectiveness and also gathered feedback leading to several module improvements. We also report on the adoptions themselves, which show the importance of supporting adaptation of the modules for diverse settings.**

***Index Terms*—Parallel computing education, heterogeneous computing, propagating educational innovations**

I. Introduction

Modern computing systems are very heterogeneous; accelerators are becoming ubiquitous from the very largest systems [23] down to portable devices such as cellphones [20]. Exploiting heterogeneous parallelism is seen as a key research challenge [3]. Despite its importance, however, heterogeneous parallel computing it not widely taught in CS programs.

The ToUCH project (Teaching Undergraduates Collaborative and Heterogeneous computing) was created to try to fill this gap. It created a collection of modules, each covering some aspect of Heterogeneous Computing (HC). Each module is designed to be mostly self-contained and is scoped so it can be completed in 1–2 days of class time. The idea is that these modules can be added to courses throughout an institution's curriculum. The overall approach is "early and often": students should see HC repeatedly at increasing levels of complexity. Compared with creating a new course, this approach emphasizes that heterogeneous systems occur throughout computing. It also supports incremental adoption by adding modules one at a time, which is thought to be politically easier than making curriculum-wide changes all at once. The modules are available on the project github repository (https://github.com/TeachingUndergradsCHC/modules).

Any effort to change curricula or pedagogy faces a dual challenge. The first is develop the innovation, the modules in this case. The second challenge is to get that innovation adopted by others; even an amazing innovation will have minimal impact if only its creators use it. It is easy to focus on creating a great innovation with the idea that others will naturally adopt it, but this approach typically fails [21]. Recognition of this fact has led to research on the adoption of educational innovations and advice for successfully spreading innovations (e.g. [13], [22]). Since the Edu* community strives to promote the teaching of parallel computing, we need to address the adoption challenge as well as the creation one.

To promote their modules, the ToUCH project ran a faculty development workshop during Summer 2021. Participants were recruited through posts on CS education and HPC email lists. They applied via an online form that asked about their interest in teaching HC, the relevant courses they teach, the modules that interested them, and the parallel computing topics they already teach. Due to the pandemic, the workshop was held virtually, using a non-traditional organization; instead of meeting continuously for one or more days, the workshop was held as a pair of group meetings (both run twice for scheduling

This work was partially supported by the National Science Foundation through awards OAC-1829554, OAC-1829644, and CNS-2104925.

reasons), with organizers meeting each participant individually between the group meetings to help them plan.

This workshop led to numerous module adoptions and also important feedback about how to improve them. We report on these things as well as observations about promoting adoption itself. Our specific contributions include the following:

- The results of a variety of assessments showing module effectiveness.
- A discussion of useful feedback received from instructors adopting the modules.
- Lessons learned about promoting curricular modules.

The first point is primarily aimed at those who could use the modules in their courses. The last two are useful for those creating pedagogical materials or promoting their adoption.

The remainder of the paper is organized as follows: Section II presents related work about teaching heterogeneous computing and propagating educational innovations. Section III describes the ToUCH modules. Section IV discusses how they were adopted. Section V presents assessment results and adopter feedback. Section VI discusses the lessons learned about adoption itself. We conclude in Section VII.

II. Related Work

In spite of the increasing prominence of heterogeneous systems, there have been relatively few efforts to teach HC to CS undergraduates [19]. Carloni *et al.* designed an upper-level elective course in which heterogeneous programming is taught using a system-level design approach [7]. The course primarily caters to computer engineering and EE undergraduates. Fuentes *et al.* present a course in which HC is taught using DPC++ [11]. This course uses several ToUCH modules, as discussed in § IV of this paper. Frachtenberg introduced an upper-level course at Reed College that focuses on heterogeneous architectures [10]. The key takeaways from their experiences are reflected in the ToUCH approach.

Broader initiatives for teaching parallel computing have started to incorporate heterogeneous computing content. The Center for Parallel and Distributed Computing Curriculum Development and Educational Resources (CDER) promotes parallel and distributed computing education. Their most recent curricular recommendations list several HC topics, including SIMD/vector instructions and accelerators (GPUs and in general) as well as related topics such as power/energy and edge computing [8], [18]. The CSinParallel collection includes several modules that mention HC topics [1], [5], [6], [12], [15], [16]. These modules are primarily designed to teach students CUDA programming and do not emphasize or expose students to the underlying HC concepts. They have also used the Raspberry Pi as a platform for introductory computing and for building a cheap cluster [16], but the focus is on the Pi as a cheap Linux system, not to explore the ARM architecture.

III. Modules

ToUCH modules divide into 4 categories, each labeled with a letter: Fundamentals (A), Algorithms (B), Architecture (C), and Programming models (D). We refer to modules by an ID formed from this letter and a number. The following modules had been created by the ToUCH project:

- **A1 Heterogeneous Computing: Elementary Notions**. Slides that motivate heterogeneity, give examples of processors using it, and show how it complicates scheduling.
- **A2 Task Mapping on Soft Heterogeneous Systems**. Slides and a hands-on demonstration of the importance of task mapping when cores have different speeds.
- **A3 Pollack's Rule as a Justification for Heterogeneous Computing**. Slides and written problems showing how having "fat" and "thin" cores benefits diverse workloads.
- **B1 Hybrid Algorithms**. Adds accelerator offloading to the dynamic multithreading programming model from CLRS [9] to analyze hybrid parallel divide-and-conquer algorithms. Has written and programming exercises.
- **C1 Introduction to ARM**. Slides and assembly programming exercises to introduce the ARM architecture (using a Raspberry Pi) as a contrast to MIPS. Includes material on the NEON SIMD coprocessor and Thumb mode.
- **C2 GPU Memory Hierarchy**. Slides and a CUDA programming lab in which students use tiling to improve the memory performance of matrix multiplication. Uses CUDA shared memory as a programmer-managed cache.
- **D1 Introduction to CUDA Programming**. Slides to introduce CUDA and an image processing lab.
- **D2 Heterogeneous Programming with OpenMP**. Slides and tutorials on using OpenMP for GPU task offloading.

Each module was used by at least one workshop participant. In addition, one participant created a new module by porting module D1 to use SYCL instead of CUDA:

- **D3 Introduction to SYCL Programming**

Several modules (C2, D1, D2) have labs that can use cloud resources (Google Colab). This removes the need for local machines with GPUs.

Modules A1, A2, A3, and C1 have been presented previously [19], but this is the first publication with the others.

IV. Adoption

The modules were adopted at 10 universities with varying characteristics, from small liberal arts colleges to large research institutions. The modules were introduced in courses at different levels from CS1 to Masters. The courses had combined enrollment of 396 students. This section captures the key details about the adoptions. Table I provides a high-level summary. Except when stated otherwise, the modules were adopted in the Fall 2021 term.

A. St. John's University

At St. John's University, the A1 module was adopted in two sections of Introduction to Computer Fundamentals. The students were freshmen and materials from the module were included in a discussion of the anatomy of a computer.

TABLE I
SUMMARY OF MODULE ADOPTIONS

Course	Institution	n	Modules
Intro to Computer Fundamentals	St. John's University	59	A1
Computer System Management and Support	CityTech, City Univ. New York (CUNY)	5	A1
Networking Fundamentals	CityTech, City Univ. New York (CUNY)	5	A1
Intro to Computer Security	CityTech, City Univ. New York (CUNY)	48	A1
Information Systems Development Project	Baruch College	25	A1
Programming II	Belmont University	27	A1
Programming Languages	Belmont University	30	A1
Intro to Computing Systems	Knox College	8	A1, A3, D1
CS 2	Westfield State University	15	A1, A2
Data Structures	Westfield State University	8	A1, A2
Computer Organization	Pomona College	12	A3, D2
Parallelization of Programs	Indian Institute of Technology Bhilai	9	D1, C2
Computer Architecture	Izmir Institute of Technology	100	C1, C2
Heterogeneous Computing	Universidad del Bío-Bío	21	A1, A3, B1, C1, C2, D1, D3
High Performance Computing	University of South Carolina Beaufort	13	A1
Computer Architecture	University of South Carolina Beaufort	11	C1

B. CityTech, City University of New York (CUNY)

Module A1 was also used in three courses at CityTech:

- Networking Fundamentals (9 students)
- Computer Systems Management and Support (6 students)
- Intro to Computer Security (2 sections of 24 each)

All these are low-level courses in Information Systems; all students were undergraduates. The first two courses are required and the third is an elective. The module and a CACM article [25] were used to introduce HC with a focus on the complexity of modern (and future) computing systems. This fits naturally in the first two courses. For Computer Security, it was a "special topic" with the justification that securing a system requires understanding it and that complexity increases the potential for breaches.

C. Baruch College

In the Computer Information Systems department at Baruch College, the A1 module and an introductory article [25] were provided as a self-study introduction to heterogeneous computing in an masters-level capstone course. Subsequently, the students learned Intel oneAPI and used it for a term project.

D. Belmont University

At Belmont University, module A1 was adopted in two sections each of Programming II and Programming Languages. Programming II is a sequence course in Java that expands on data structures, recursion, and algorithm analysis. Programming Languages is an upper-level course covering functional, logic, and OO programming languages. Each section had about 15 students, mostly sophomores in Programming II and juniors or seniors in Programming Languages.

E. Knox College

At Knox College during Winter 2022, modules A1, A3, and D1 were tested with a special session of students who had taken Introduction to Computing Systems the previous year (Winter 2021). This is a required systems course and the one in the major that most heavily discusses parallelism; coverage includes pthreads, OpenMP, and concurrency problems such as producer-consumer. It is mostly taken by sophomores and juniors. The course normally discusses heterogeneous parallel computing, including module A3, but it was omitted in Winter 2021 in response to the pandemic. All students from that term were invited to the special session unless they were studying abroad or had graduated. Of the 16 students invited, 8 participated. The modules were covered on a weekend afternoon and students were paid for their time.

F. Westfield State University

At Westfield State University, modules A1 and A2 were adopted in the CS2 and Data Structures classes. Both are sophomore-level classes. CS2 is a prerequisite of Data Structures. There were 15 students in CS2 and 8 in Data Structures. In both classes, this was the students' first exposure to PDC material. In CS2, the material was covered as an advanced topic in the semester's last week. In Data Structures, it was in weeks 6–7, before coverage of shared-memory parallel programming in Java and Pyjama [14].

Both classes used the slides and also the Colab notebook allowing experimentation with Amdahl's law. In Data Structures, module A2's Jupyter notebook was used for an in-class demonstration.

G. Pomona College

At Pomona College, modules A3 and D2 were used in Computer Organization, an upper-division elective using a standard text [17]. All students had taken a three-course intro CS sequence and half had taken a computer systems course.

Materials from module A3 were used after introducing homogeneous chip multiprocessors (CMPs). Later, after introducing OpenMP, module D2 was used in a discussion of accelerators (GPUs, FPGAs, and ML accelerators).

H. Indian Institute of Technology Bhilai (IIT Bhilai)

At IIT Bhilai, modules D1 and C2 on CUDA were adopted in Parallelization of Programs during the 2021-22-M semester. The course had previously covered only compile-time and runtime parallelization. It was an online course with 9 students.

The course began by introducing parallel programming, architectures, and programming models. The modules were used to illustrate SIMD and optimizations on GPUs. Students did the programming tasks from the modules and others in addition, including some to speed up applications using GPUs. They were also encouraged to present a research paper about using GPUs for data-parallel applications.

I. Izmir Institute of Technology

At Izmir Institute of Technology in Turkey, modules C1 and C2 were added to Computer Architecture in the Computer Engineering department. It is a third-year undergrad course that teaches assembly language, single-cycle processor design, pipelining, and caching using Patterson and Hennessy [17]. It had 100 students, 99 in computer engineering and 1 in electrical engineering taking the course as an elective.

Module C1 was covered in week 6, after basic performance issues and MIPS assembly language. This part of the course did not have lab sessions or programming assignments. 55 students attended the online lecture on this module.

Module C2 on the GPU memory hierarchy was added in week 12, right after caching. The students were new to CUDA so the instructor did not cover all the details, like synchronization. 19 students attended the online lecture. There was also a lab session where a TA introduced CUDA and demonstrated the naive and tiled matrix multiplication codes.

J. Universidad del Bío-Bío

At the Universidad del Bío-Bío in Chile, many of the modules were used to create Heterogeneous Computing, a new undergraduate elective about parallel programming for heterogeneous architectures. Intel's OneAPI was the primary development framework. 21 students, all in the fourth year of the CS BS program, took the course. They had previously covered parallelism and shared-memory programming.

The class was divided into the following four main units, with the modules used in each listed in parentheses:

- Heterogeneous architectures (A1, C1, and C2).
- Programming models (D1 and D3).
- Algorithms (B1).
- Performance (A3).

Hands-on exercises and labs were completed throughout the course and a quiz was given on each unit. Students programmed discrete/integrated GPUs, FPGAs, and CPUs using Intel Devcloud. The course also included two guest talks on GPU Computing and High-Performance Computing, respectively. All the lectures, labs, project, and quizzes are posted online at https://jfuentes.github.io/classes/hc. More details are available in a paper specifically about this course [11].

K. University of South Carolina Beaufort

At the University of South Carolina Beaufort, module A1 was adopted in High Performance Computing and module C1 was used in Computer Architecture. These courses are required for the Computational Science and Information Science and Technology majors, respectively. High Performance Computing had 13 students (2 graduate students, 10 seniors, and 1 junior). Computer Architecture had 6 seniors and 5 juniors.

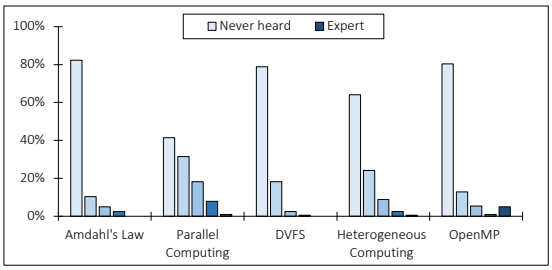


Fig. 1. Module A1 combined pre-assessment scores for courses at CUNY, St. John's University, Baruch College, and Belmont University. (n=203)

V. Observations from Assessment

Now we present the lessons from these adoptions, both about module effectiveness and how to improve them.

A. Need for the Modules

The most common adoption was using module A1 to introduce the big-picture idea of heterogeneous computing. Many classes adopting this module used the same pre-assessment survey, which had been provided during the workshop. It asks students how familiar they are with parallel concepts on a scale from 1="Never heard of it" to 5="I'm an expert".

The combined results for many of the adoptions are shown in Figure 1. We present the combined results rather than results per class or institution because all the results are qualitatively similar; some of the students have heard about parallel computing, but they know very little about any of the other concepts. Even the masters students are not more knowledgeable about these topics. We interpret these results as demonstrating the need for the modules; clearly students are not learning about these concepts from other sources.

B. Effectiveness of A1 and A2

Now we look at evidence that students learned from the modules. Unfortunately, due to a misunderstanding, none of the courses whose results appear in Figure 1 gave a post-test to collect such evidence. The instructors of these courses did feel that the module was well-received and useful for teaching basic concepts, however.

A pre/post test arrangement was used for the adoption of module A1 at Knox College. The pre-test survey was the same as used by the other adopters. The post-test has the same questions except the one about OpenMP, which is only to assess background since module A1 does not cover OpenMP. The results are shown in Figure 2. The pre-test results are notably better than those in Figure 1, which we attribute to the students being recruited from a course that had significant parallel computing coverage (though less on heterogeneous systems) and possibly selection bias since only a subset of the original class participated in the modules. The post-test shows that the students felt they had learned from the A1 module.

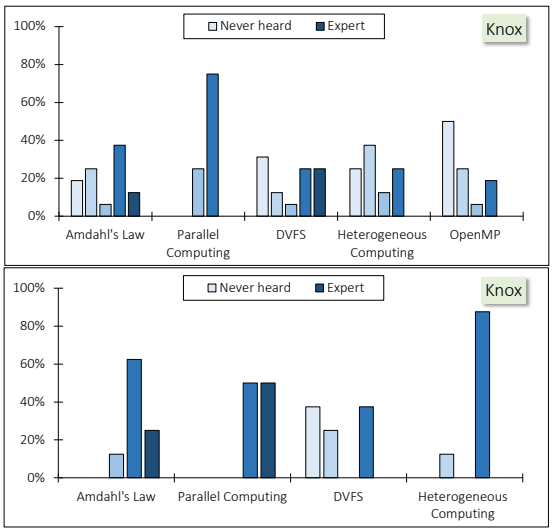


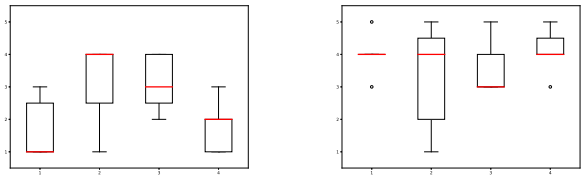
Fig. 2. Module A1 pre-assessment (top) and post-assessment (bottom) scores for students at Knox College. (n=8)

The post-test also included short-answer questions to check the actual level of student understanding; student answers were consistent with their perceived level of understanding.

The adoption at Westfield University used a separate set of questions to demonstrate improved interest and sense of self-efficacy from modules A1 and A2. The questions are shown in Figure 3 and Likert-scale answers for the Data Structures class are shown in Figure 4. There were not enough students who took the survey from the CS 2 class, likely because the material was taught at the end of the semester. Students in both classes also completed a summative assessment asking for speed-up and Amdahl's law calculations. Students showed strong mastery of the material, with the majority of students able to perform these calculations without errors.

I feel that I have a good understanding of PDC concepts
I think that PDC will be important to me in my future career
I think I will enjoy learning about PDC in this class
I understand how I can speed up a program using parallelism

Fig. 3. Survey questions



(a) Before (b) After

Fig. 4. Formative assessment result for the Data Structures class

C. Effectiveness of A3 (Pollack's Rule)

The other introductory module is A3, which examines different processor configurations, estimating their performance analytically to show the advantage of heterogeneous core sizes in the presence of a varied workload. It is an introduction to performance estimation and also motivational.

To assess this adoption, students were asked to perform two calculations: use Pollack's rule [4] to estimate core performance in a specific system and then use Amdahl's law to estimate the running time of a partially-parallelizable program on this system. These tasks are analogous to worked exercises from the module but for a different system. On each task, 6 students (of 8) correctly performed the calculation, one set up half the calculation correctly but not the other half, and one student showed some understanding but made a significant conceptual error. (Different students struggled on each part; the two students who missed the first part were correct on the second part except for using their incorrect results from the first part). Thus, the module seems to be teaching how to perform these calculations.

The students were also asked what they learned about the motivation for heterogeneous systems. Five said something about different systems being better for different jobs or something about the tradeoff between number of cores and core performance, which is the point of the module. One said he had already known why heterogeneous systems are used, but his answer didn't identify the motivation. The other two did not address the "why" part of the question and only talked about the calculations. Thus, the results suggest that the module is fairly successful in motivating the use of heterogeneous systems, but that the motivational aspect of the module could use some emphasis to ensure that students see the reason for the calculations.

The other adoption of module A3 was at Pomona College, where we did not have IRB permission to examine student work, but the instructor made an important observation. Implicit in the module is the assumption that the total die area is fixed. The module creator had made this assumption automatically, but it is not explicitly stated in the module. Coming at the module with fresh eyes, the adopter identified this issue and suggested a broader framing of the module as a discussion of the choice between a monolithic core, homogeneous cores, and heterogeneous cores. Because of this feedback, we have made the assumption about fixed die size explicit and we plan to broaden the framing. Thus, this feedback has already made the module more portable to other instructors and will also lead to an overall improvement.

D. Effectiveness of C1, D1, and C2 (ARM and CUDA)

Our evaluation of the ARM and CUDA modules (C1, D1, and C2) is complicated because several of their assessments were over multi-module adoptions. At IIT Bhilai, student scores on assignments related to the modules (D1 and C2) were compared to their overall scores. There was a nearly linear relationship, with students scoring around 5% higher on the modules than the rest of the course.

Assessment at the Izmir Institute of Technology also used student performance on assignments. For module C1, a general

TABLE II
RESULTS FROM ATTITUDINAL SURVEYS

Course	Institution	Modules	helpful & engaging	impacted interest	compared to rest of course
Computer Architecture	Izmir Inst. Tech.	C1, C2	3.17 / 4	2.99 / 4	2.96 / 4
High Performance Computing	Univ. SC Beaufort	A1	3.58 / 4	4.18 / 5	4.42 / 5
Computer Architecture	Univ. SC Beaufort	C1	3.6 / 4	4.1 / 5	4.63 / 5
Heterogeneous Computing	Univ. Bío-Bío	A1, A3, B1, C1, C2, D1, D3	3.86 / 4	3.67 / 4	—

question about heterogeneous computing was on a written in-class exam. Of the 91 students taking the exam, 20 did not answer the question, only 14 received full credit, and the average score was 2.79/10 while the overall exam average was 51/100. Clearly the students struggled with this material. The instructor attributes this to using the module too early in the term. In particular, it appeared before pipelining, which is important because one of the features of ARM discussed in the module is conditional instructions, an alternative to branches that can improve pipeline performance in some cases. Students also struggled with the SIMD instructions since this was their first exposure to parallelism. This was a learning experience for the instructor, who will use the module later in the term going forward, and for the ToUCH team, who will re-examine module prerequisites and make sure they are sufficiently prominent in the module descriptions.

The adoption of module C2 at Izmir Institute of Technology was assessed with an optional part of the last programming assignment. 20 students submitted this part and the average score was 35/50 while the mandatory part was completed by 66 students with an average score of 80/100. The lower average, particularly for an optional assignment, suggests that the students still found the material challenging, but the students were much more engaged with this material than module C1. They asked more questions in lecture and seemed to see that a software developer who understands the underlying computer architecture could take fuller advantage of the memory hierarchy. It was a very good experience for students interested in both hardware and software.

Interestingly, this module was also presented without all the intended prerequisites. The students had seen thread parallelism and OpenMP, but not CUDA even though the module uses it; the instructor felt that this was not a difficult leap for the students to make. It likely also helped that the instructor had just covered blocking for serial matrix multiplication, which is very similar to the blocking the module asks them to do in CUDA. The module was not designed with the assumption that students would have previously seen blocking; the instructor found an alternate prerequisite for this module.

The adoption of module D1 at Knox College was assessed by looking at how well students were able to finish its image processing lab. The lab took them longer than a normal period (1.5–2 hours), but most students completed the assignment, which was blurring an image by averaging the color of nearby pixels. They also discovered an issue with the lab, which was that it did not specify the desired behavior on the boundary; this has now been clarified in the module materials.

E. Suitability for Use in a Specialized Course

Last, we consider the use of modules as part of a course entirely on HC at the Universidad del Bío-Bío. Table III shows the evaluation activities, the modules each covers, and student grade statistics (the range is 1–100). The students generally mastered the material. More details can be found in Fuentes et al. [11].

TABLE III
SCORES FROM UNIVERSIDAD DEL BÍO-BÍO COURSE

Assessment	ToUCH Modules	# Students	Mean	St. Dev.
Quiz 1	A1, C1, C2	21	88.9	11.7
Quiz 2	D1, D3	21	83.9	13.9
Quiz 3	B1, D3	21	92.7	9.1
Quiz 4	A3	21	87.2	15.4
Lab 1	C2, D3	20	91.7	14.9
Lab 2	A2, C2, D3	19	92.5	19.7
Project 1	B1, D3	19	84.4	28.8
Project 2	D3	20	78.8	22.1

F. Attitudinal Changes

In addition to showing that students learned from the modules, some instructors gave a survey to see how the students felt about the modules. Table II reports the average score for the following questions:

- Were the class activities helpful and engaging? (1="not helpful", top score="very helpful")
- Did the material covered have any impact on your interest in Computer Science? (1="decreased my interest", top score="increased my interest a great amount")
- How would you rate your learning experience in the module compared to the rest of the class? (1="module experience was worse", top score="module experience was significantly better")

Different instructors used different top scores for some of the questions; this is the number after the slash in each table entry.

In addition, the students in Computer Architecture at the Izmir Institute of Technology students were also asked if they would consider learning more about heterogeneous computing (for example, by taking an elective course). The answer counts were "Yes"=7, "Maybe"=10, and "No"=6.

The students liked the modules. All the numbers in Table II are above the midpoint of their range. The lowest scores are from the Izmir Institute of Technology and, as explained

in Section V-D, module C1 was used too early in that course. That issue will be fixed going forward.

G. General Observations

We also made a couple of observations that are more general than a single module. The first is that the initial versions of the slides didn't provide enough information about how they were intended to be used. The text on a slide often does not include the transitions between slides or even explicitly give the purpose of the slide, both important for an adopting instructor trying to convey the "story" of a slide deck. Some of the stories were demonstrated during the workshop when a subset of the slides were presented by the module creators, but time was limited. Potential adopters also need pedagogical content knowledge about each slide's purpose and the misconceptions it addresses, neither of which is obvious when seeing a presentation. To fill these gaps, the slide notes or other module materials must be significantly expanded to support adopters. As with other feedback, the ToUCH project is working to address this.

Another general observation is that many adopters significantly changed the materials while adopting them. At first, there seems to be tension between this observation and the need for more information about the creators' intentions about module use; why add more information if adopters will just change the module anyway? The goal of adding information is not to enforce "correct" module use, however. Changes are needed so materials can fit different student backgrounds, different courses, and instructors with different goals or teaching styles. It is a best practice to support adaptation of materials being propagated [13] even though some adopters make changes breaking the materials [2], [24]. Key to avoiding this is exactly the pedagogical content knowledge described above; knowing the module's design and each part's purpose allows adapters to make informed choices.

Finally, we found that Colab was popular among the adopters since it freed them from hardware limitations. That said, one commented that using the modules on local hardware would be preferable; using GPUs on Colab requires extra steps and virtualization complicates performance measurement.

VI. Meta-Observations

This section presents observations about the process of adoption itself. The first is that the modules were adopted quite differently than expected by their creators, who were not expecting uses such as module A1 going into a security course or the modules being used en masse for a specialized heterogeneous computing course. Module creators can't "expected the unexpected", but our experience shows that they must expect and support adaptation of their materials. In particular, the modules were designed to introduce specific topics, but there are different kinds of introductions. The module aimed at the lowest-level students (A1) was provided to graduate students. This is a quite different audience, but the topic was still new to those students and the delivery was modified to suit their advanced level by adding another article and making it a before-class reading. Less extreme was the use of modules in courses that subsequently spent significant time on the topic, which happened in a number of cases. One suggestion to support module use in more advanced settings is to identify or create more advanced resources. These particularly benefit instructors covering the topic more deeply, but any instructor can benefit from a more advanced introduction.

We also clearly saw that adopting modules is an iterative process. Several instructors suggested improvements to make the modules work better in their context and others plan to adjust how they use the modules. For example, although each module should only be used in one required class in a program, two colleges experimented with module A1 in multiple classes. This gives more students exposure to the new material and also helps determine where the material best fits.

The workshop format was also an experiment. The original decision to use Zoom instead of an in-person workshop was due to COVID, but this allowed participation by a much broader audience, including several faculty from outside the US, which would have been cost-prohibitive for an in-person workshop. Since attendees spanned so many time zones, each part of the workshop was actually done twice. This meant long days for the organizers, but also made each group smaller, which facilitated a productive interaction among organizers and participants despite the event being virtual.

Another part of the experimental workshop format was splitting it over multiple days. It was two group meetings a week apart. In the first, the organizers introduced the ToUCH project and modules. Then breakout rooms with one organizer in each were used for questions and demonstrations. In the second group meeting, attendees presented their plans. Between these meetings, the organizers met with each attendee at least once to discuss their plans, answer questions, and provide other resources. The gap between meetings was intended to replace an in-person event's "work time". Attendees could schedule their time as needed, but there was a clear expectation to start incorporating the modules into their course. Having to present their plans helped push attendees to move beyond thinking about the modules; from our own experience, the organizers know that it's easy to think something is a good idea without actually giving it time from our busy schedules.

Our final format experiment was to directly incentivize module use and the reporting back of assessment results. The incentive was both part of the workshop stipend and joining a paper about the modules (this paper). Others, including the CDER parallel education workshops, have split workshop stipends, but we are not familiar with others doing collaborative publication with attendees. The idea is that people are motivated by more than money. Most faculty are expected to publish and this helps with that, as well as exposing faculty to the idea of publishing educational work, something few do in graduate school. As such, it attempts to engage another part of our faculty identity (researcher) to motivate teaching HC.

With these format experiments, the workshop's net effect is that 10 instructors adopted at least one ToUCH module and reported on the results. (Nine of them helped write this paper,

1 opted out.) Seventeen instructors attended the first part of the workshop. Thirteen of them attended the second meeting, but 3 still have not yet adopted a year later. They have nearly all expressed interest in the materials and some have specific reasons for not adopting yet (e.g. changing job roles), but we believe that most instructors who will adopt have done so.

Our sense is that 10/17 is a reasonable workshop success rate, particularly for instructors attending and changing their courses during the pandemic, but a comparison is hard since we are not aware of published figures from other workshops. Certainly, a higher rate would be better. Looking forward, we will continue reaching out to all attendees to support adoption and continued use the modules. We also encourage other groups running workshops to report their adoption rate.

Despite our feeling that the workshop was reasonably successful, we have mixed feelings about virtual workshops going forward. The lack of travel and more flexible scheduling allow broader participation. That said, we think collaboration is easier once people have met and gotten to know each other at least a bit. Thus, we recommend holding workshops in person, but suggest our format (individual meetings between two group meetings) when virtual meetings are necessary.

One area where the workshop could have improved is assessment. The organizers initially planned for attendees to design assessment appropriate to their institutions and course. This is unrealistic since most faculty are new to assessment and inconsistent with the goal of writing a combined paper. Instead, organizers should provide assessment instruments to provide consistency; the wide use of the questions provided for A1 shows that most attendees would prefer this. At a minimum, attendees need specific assessment design support.

VII. Conclusion

From the perspective of the ToUCH project, we consider this workshop and the resulting adoptions to be a success. The modules were introduced by a variety of instructors who increased the awareness of heterogeneous computing among their students. We have collected data on several aspects of the modules that generally supports their effectiveness and suitability. We also got valuable feedback which is being used by the project to improve the modules.

References

[1] J. Adams, R. Brown, and E. Shoop. Patterns and Exemplars: Compelling strategies for teaching parallel and distributed computing to CS undergraduates. In *Proc. 3rd NSF/TCPP workshop on parallel and distributed computing education (EduPar)*, 2013.

[2] T.M. Andrews, M.J. Leonard, C.A. Colgrove, and S.T. Kalinowski. Active learning not associated with student learning in a random sample of college biology courses. *CBE Life Sciences Education*, 10:394–405, 2011.

[3] A. Banerjee, S. Basu, E. Brunvand, P Mazumder, R. Cleaveland, G. Singh, M. Martonosi, and F. Pembleton. Navigating the seismic shift of post-Moore computer systems design. *IEEE Micro*, 41:162–167, Nov.–Dec. 2021.

[4] S. Borkar. Getting gigascale chips: Challenges and opportunities in continuing Moore's Law. *Queue*, 1(7):26–33, 2003.

[5] R.A. Brown and E. Shoop. Modules in community: Injecting more parallelism into computer science curricula. In *Proc. 42nd SIGCSE Technical Symp. Computer Science Education (SIGCSE TS)*, pages 447–452, 2011.

[6] Richard Brown, Elizabeth Shoop, Joel Adams, Curtis Clifton, Mark Gardner, Michael Haupt, and Peter Hinsbeeck. Strategies for preparing computer science students for the multicore world. In *Proceedings of the 2010 ITiCSE working group reports*, pages 97–115, 2011.

[7] L.P. Carloni, E.G. Cota, G. Di Guglielmo, D. Giri, J. Kwon, P. Mantovani, L. Piccolboni, and M. Petracca. Teaching heterogeneous computing with system-level design methods. In *Proc. Workshop Computer Architecture Education*, pages 1–8, 2019.

[8] A. Chtchelkanova, S. Das, C. Das, F. Dehne, M. Gouda, A. Gupta, J. Jaja, K. Kant, A. La Salle, R. LeBlanc, A. Lumsdaine, D. Padua, M. Parashar, S. Prasad, V. Prasanna, Y. Robert, A. Rosenberg, S. Sahni, B. Shirazi, A. Sussman, C. Weems, and J. Wu. NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing— core topics for undergraduates. version 1, http://tcpp.cs.gsu.edu/curriculum/?q=system/files/NSF-TCPP-curriculum-version1.pdf, 2012.

[9] T. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

[10] E. Frachtenberg. Experience and practice teaching an undergraduate course on diverse heterogeneous architectures. In *Proc. Workshop on Education for High-Performance Computing (EduHPC)*, 2021.

[11] J. Fuentes, D. López, and S. González. Teaching heterogeneous computing using DPC++. In *Proc. 12th NSF/TCPP workshop on parallel and distributed computing education (EduPar)*, 2022.

[12] P. Garrity, T. Yates, R. Brown, and E. Shoop. WebMapReduce: An accessible and adaptable tool for teaching map-reduce computing. In *Proc. 42nd SIGCSE Technical Symp. Computer Science Education (SIGCSE TS)*, pages 183–188, 2011.

[13] C. Henderson, R. Cole, J. Froyd, D. Friedrichsen, R. Khatri, and C. Stanford. *Designing educational innovations for sustained adoption: A how-to guide for education developers who want to increase the impact of their work*. Increase the Impact, Kalamazoo, MI, 2015.

[14] R. Kurniawati. Teaching parallel programming with Java and Pyjama. In *Proc. 53rd ACM Technical Symposium on Computer Science Education (SIGCSE TS)*, volume 2, page 1109, 2022.

[15] S.J. Matthews, J.C. Adams, R.A. Brown, and E. Shoop. CSinParallel Project. http://csinparallel.org/.

[16] S.J. Matthews, J.C. Adams, R.A. Brown, and E. Shoop. Portable parallel computing with the Raspberry Pi. In *Proc. 49th SIGCSE Technical Symp. Computer Science Education (SIGCSE TS)*, pages 92–97, 2018.

[17] D.A. Patterson and J.L. Hennessy. *Computer Organization and Design MIPS Edition: The Hardware/Software Interface*. Morgan Kaufmann, 6th edition, 2020.

[18] S.K. Prasad, T. Estrada, S. Ghafoor, A. Gupta, K. Krishna, C. Stunkel, A. Sussman, R. Vaidyanathan, C. Weems, K. Agrawal, M. Barnas, D.W. Brown, R. Bryant, D.P. Bunde, C. Busch, D. Debs, E. Freudenthal, J. Jaja, M. Parashar, C. Phillips, B. Robey, A. Rosenberg, E. Saule, and C. Chen. NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing— core topics for undergraduates. version 2-beta, https://tcpp.cs.gsu.edu/curriculum/?q=system/files/TCPP%20PDC%20Curriculum%20V2.0beta-Nov12.2020.pdf, 2020.

[19] A. Qasem, D.P. Bunde, and P. Schielke. A module-based introduction to heterogeneous computing in core courses. *J. Parallel and Distributed Computing*, 158:56–66, 2021.

[20] Y.S. Shao, B. Reagen, G. Wei, and D. Brooks. The aladdin approach to accelerator design and modeling. *IEEE Micro*, 35(3):58–70, 2015.

[21] C. Stanford, R. Cole, J. Froyd, C. Henderson, D. Friedrichsen, and R. Khatri. Analysis of propagation plans in NSF-funded education development projects. *J. Sci. Educ. Technol.*, 28:418–437, 2017.

[22] C. Taylor, J. Spacco, D.P. Bunde, Z. Butler, H. Bort, C.L. Hovey, F. Maiorana, and T. Zeume. Propagating the adoption of CS educational innovations. In *Proceedings Companion of the 23rd Annual ACM Conf. on Innovation and Technology in Computer Science Education (ITiCSE-WG)*, pages 217–235, 2018.

[23] Top500 List. November 2021 list. http://www.top500.org/, viewed Dec 2021.

[24] C. Turpen, M. Dancy, and C. Henderson. Perceived affordances and constraints regarding instructors' use of Peer Instruction: Implications for promoting instructional change. *Phys. Rev. Education Research*, 12(010116), 2016.

[25] M. Zahran. Heterogeneous computing: Here to stay. *CACM*, 60(3):42–45, 2017.