ParallelAR: An augmented reality app and instructional approach for learning parallel programming scheduling concepts

Marin Abernethy, Oliver Sinnen,
Giuseppe De Ruvo, Nasser Giacaman
Parallel and Reconfigurable Computing lab
Department of Electrical and Computer Engineering
Auckland, New Zealand
mabe628@aucklanduni.ac.nz,{o.sinnen,
g.deruvo, n.giacaman}@auckland.ac.nz

Joel Adams
Department of Computer Science
Calvin College
Grand Rapids, USA
adams@calvin.edu

***Abstract*—Parallel programming has rapidly moved from a special-purpose technique to standard practice. This newfound ubiquity needs to be matched by improved parallel programming education. As parallel programming involves higher level concepts, students tend to struggle with turning the abstract information into concrete mental models. Analogies are known to aid in this knowledge transfer, by providing an existing schema as the basis for the formation of a new schema. Additionally, technology has been proven to increase motivation and engagement in students, which ultimately improves learning. Combining these ideas, this paper presents several contributions that enhance aspects of parallel programming education. These contributions include a set of collaborative learning activities to target fundamental scheduling concepts, a detailed analogy to assist in the understanding of the scheduling concepts, and an augmented reality application to facilitate the collaborative learning activity by bringing the analogy to life.**

I. Introduction

Parallel programming has rapidly moved from a special-purpose technique to standard practice. Hyper-threading and multi-core architectures are now the norm. As the hardware has changed fundamentally, so must the software. We have reached a turning point, where developers must now harness the processing power of these machines through parallel and concurrent programming techniques. Due to this concurrent revolution, the expectations for software developers have changed, and the corresponding curriculum must evolve accordingly. However, teaching and learning even basic programming is difficult. Undergraduate or novice programmers are especially challenged by higher level and abstract concepts, often leading to high drop-out rates in programming courses [1], [2]. Compared to constructing a sequential program, parallel and concurrent programming requires a different, and more complex mental model of control flow [3].

Analogies have been heavily researched as a tool to aid in learning new, unfamiliar or abstract concepts. Specifically, an analogy depends on knowledge from an established domain, known as the source, and applying it to a new domain, called the target [4]. The source can be in the real world, or another concept better understood by the student. This can benefit the learner, making abstract information concrete by providing an existing schema as the basis for the formation of a new schema [5]. Analogies therefore bridge students' understanding by helping them relate the unfamiliar concept to something they are already familiar with.

Additionally, there exists persuasive research suggesting teaching approaches based upon the constructivist learning theories are successful in promoting motivation, critical-thinking skills, and a myriad of other benefits. The approaches explored in this report include Active Learning, Collaborative Learning, and Constructive Alignment.

Technology is known to bolster effective collaborative learning [6], [7]. When used in schools, it can also help bridge educational and social gaps by compensating for unequal access to technologies at home [8]. According to the Pew Research Center, 77% of Americans now own smartphones, while only 35% did in 2011 [9]. This rapid growth in the past six years suggests that mobile technology may be a promising platform to explore. This is particularly the case in the education sector, where the use of technology can be a powerful mechanism for equality.

Augmented reality (AR) may be the solution that brings all of these points together. Mobile applications have the potential to deliver accessible educational solutions to students, regardless of social or economic status. While there are expensive AR headsets on the market, many smartphones are already capable of harnessing the fundamental aspects of AR. Here, AR benefits from the accessibility of mobile technology for education. Furthermore, AR is inherently about augmenting the real world with virtual elements; likewise, analogies are about using real world stories to describe a virtual (abstract) situation.

The contributions of this paper are as follows:

1) A carefully-designed analogy to help instructors explain introductory parallel scheduling concepts to students.

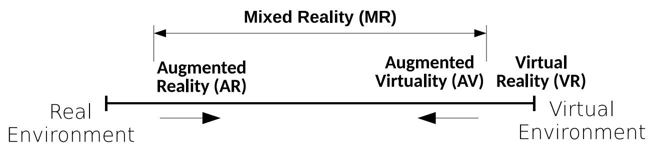

Fig. 1. The Reality-Virtuality Continuum [11], where augmented reality adds virtual enhancements to elements from a real-world environment view. In the case of ParallelAR, the real-world elements are the learning flashcards.

2) A collaborative learning activity to target fundamental scheduling concepts inspired by the Curriculum Initiative on Parallel and Distributed Computing (PDC CI) [10], with downloadable flashcards and recommended workflow instructions to support the learning activity.
3) Alignment of the activity and analogy with Learning Outcomes (LOs) that are inspired by the PDC CI. This alignment aims to encourage educators in integrating the ideas within their PDC course.
4) Finally, an AR app that further facilitates the collaborative learning activity by bringing the analogy to life.

Section II provides a background overview of AR, in addition to a brief account of the educational theories and pedagogies this work leans on. Section III reviews how AR has been used in education to date. Section IV presents the analogy that is easy to explain to students. The Learning Outcomes are outlined in Section V, while the collaborative learning activity is presented in Section VI. Section VII presents the AR app, ParallelAR, before concluding in Section VIII.

II. Background

A. Augmented reality

The term "augmented reality" began to frequently appear in literature without consensus on its definition. Paul Milgram et al sought to bring clarity by establishing the reality–virtuality continuum [11]. This especially helped understanding the relationship between augmented reality (AR), virtual reality (VR) and even augmented virtuality (AV), by recognizing that reality is effectively a spectrum. As shown in Figure 1, the environment at one end consists entirely of real-world objects (as seen through a headset or camera window). At the other end, the environment seen by the user is entirely virtual (virtual reality). In between these extreme points, is the augmentation of an otherwise real environment using virtual elements (augmented reality, AR), or the augmentation of an otherwise virtual environment using real objects (augmented virtuality, as commonly achieved with green screens).

With AR, the virtual elements are superimposed onto the physical environment in real-time by the live feed of the camera. By combining virtual data with real-world data, users have access to contextually relevant and meaningful multimedia content [12]. AR has a complicated taxonomy, with many variations depending on the hardware and software. At a high level, there are two main categories of devices: wearable and non-wearable. Wearable devices include headsets and helmets, while non-wearable devices include those that are portable (e.g. smartphones and tablets) and those that are stationary (e.g. TVs and computers) [13]. Also at a high level, there are two main types of AR; marker-based which uses cameras and visual cues, and marker-less which uses positional data and various other device sensors [14]. Marker-based AR detects and tracks targets using the device's camera. These targets are often 2D images, but can also be 3D or real-world objects. Conversely, marker-less AR typically uses the device's location and orientation to superimpose virtual objects.

The popularity of AR and VR has risen significantly in recent years, with half a million AR/VR headsets sold in 2016 and forecasted sales of 13 million units will be sold in 2020 [15]. Gartner has named both of these technologies as one of the Top 10 Strategic Technology Trends for 2017 [15]. While VR has seen greater funding and development deals in the past, AR is slowly narrowing the gap [16] and many developers think AR will ultimately be bigger than VR [17] due to its versatility and more natural control. While AR research is still in its infancy, it has already been used in a variety of disciplines, such as medicine, manufacturing, aeronautics, robotics, entertainment, tourism, social networking and education [18].

In the educational domain, the AR and VR technologies have also been identified as one of the Top Five Strategic Technologies Impacting K-12 Education in 2017 [19]. According to the report, the high interest is likely driven by lowered costs of the hardware, the increase in the amount of content, and the explosive popularity of games, such as Pokémon GO. Adoption of these tools in elementary school through high school levels is still in its early stages, but are likely to see substantial development over the coming years [19].

B. Educational theories

The contributions made in this paper are based upon several fundamental learning theories and pedagogies, which will be discussed briefly in this section. Firstly, constructivism is the backbone of everything examined here, and for which many teachers regard as the building blocks of teaching and learning [20]. The core concepts behind Cognitive Constructivism is that knowledge is constructed in individuals through a personal process; people cannot be "given" information which they immediately understand and use, but rather they must construct their own knowledge through experience. Social Constructivism, on the other hand, values the construction of knowledge through interaction with teachers or other students [20]. Here, social interaction is an integral part of learning. When students complete projects or activities in a group, each individual internalizes the knowledge at a different rate according to their own experience. It is suggested that social interaction is a catalyst for a more powerful and effective internalization in each individual [20].

Active learning is based upon constructivist theories and is broadly defined as any instructional method that engages students in the learning process. In practice, this means students participate in meaningful learning activities in the classroom while reflecting on what they are doing [21]. Prince et al examined the evidence for the effectiveness of active learning by reviewing the literature for the purpose of aiding engineering instructors in making teaching decisions [21]. While they discussed the limitations of the research at hand, they still conclude that introducing active learning activities into classes can significantly improve information recall, with plenty of evidence supporting the benefits of student engagement.

Active learning is an umbrella term encompassing several variations. For example, Collaborative Learning is an instructional method where students work together in small groups toward a common goal [21]. Students are not only responsible for their own learning, but also that of their peers. In this sense, the success of one student benefits other students as well [22]. Research also suggests that groups achieve higher levels of thought and retain information longer than students who work individually [22]. Not only are students participating in collaborative learning likely to acquire better critical-thinking skills (within the subject) than students who do not participate, but most students believe group work helped them to better understand the material and stimulated their thinking process. Students would credit this to shared responsibility and reduction of anxiety associated with problem-solving [22].

Also built upon constructivist ideals is the notion of Constructive Alignment [23]. The framework transfers aspects of constructivism into classroom decisions on teaching objectives and assessment. The core principle is that aligning curriculum and assessment significantly improves the results of instruction. Constructive alignment emphasizes the importance of teaching methods that engage students in activities that mirror the performance and ways of thinking prescribed in the curriculum objectives [23]. To achieve this in practice, the intended Learning Outcomes (LOs) must be formulated first, and then the assessment criteria developed from them. Finally, activities are organized that will teach the student how to meet the assessment criteria.

C. *Parallel programming education*

The importance of education in the parallel and distributed computing disciplines is recognized by the education community internationally. The Curriculum Initiative on Parallel and Distributed Computing – Core Topics for Undergraduates (PDC CI) [10] has been proposed for computer science and computer engineering undergraduate degrees. The motivation behind the proposed PDC core curriculum is that every computing undergraduate student should achieve a specified skill level regarding PDC-related topics.

Within the published document, various topics and subtopics are outlined, along with the associated LOs and their level of coverage. While these are just recommendations, and it is up to teaching institutes to incorporate them, it does help course instructors cover a relevant curriculum. The contributions presented in this paper concentrate on part of the Performance Issues subtopic contained within the Programming topic, namely Load Balancing and Scheduling & Mapping. The specific LOs that are reinforced by this research are outlined in further detail in Section V.

III. Related work

The number of published studies about AR in education has progressively increased year by year, especially in recent years [24]. Bacca et al [24] did a systematic mapping study on all the AR applications in education to date. Similarly, Bower et al [18] and Wu et al [25] both published the state of the field regarding AR in education, in consecutive years. Both studies discussed the potential of AR, speculating on the opportunities the technology has to enhance learning and teaching, and the limitations that come along with it.

AR technology has the power and flexibility to support several effective pedagogical approaches. These include the constructivist learning theory, as briefly discussed in Section II-B, situated-based, game-based, and inquiry-based learning [18]. In the reviews, it was concluded that AR has positive effects on student motivation and consequently contributes to students meeting LOs [18], [25], [24]. In one study conducted by Tarng and Ou [26], augmented reality technologies were used to teach butterfly ecology in an elementary school. In conjunction with standard instruction (using lecture and textbook materials) that the control group relied on, the experimental group also used an AR application to breed and observe virtual butterflies on their smartphones. The study showed that the AR teaching method had a significant impact on post-test scores, indicating AR's learning potential as a learning tool.

Much of the experimental research on AR for educational purposes is within the mathematical and scientific fields because learning these subjects requires visualization of abstract concepts [25]. This suggests that AR has the ability to promote thinking skills and conceptual understanding of high level concepts, which are normally challenging to imagine [25]. Another advantage of AR comes from the accessibility of mobile technology. Students can engage in the learning process inside and outside the classroom, supporting the practices of different types of learners [12].

However, the state of the field is not all positive acclaims for this emerging technology. There are some considerable limitations that have been reported and the research is still in its very early stages. Evidence of the effects of AR on teaching and learning are often shallow [25], so it is difficult to glean a definitive answer. Some of the concerns include inflexibility of the content [25], the potential for cognitive overload or confusion by the large amount of information [25], the limited number of use-cases to inform teacher practice [18], and the novelty of AR as a distraction rather than a benefit [24]. Furthermore, if an AR system does not work as intended or there are technical difficulties, such as issues maintaining superimposed information [24], teachers are rarely equipped or trained to handle it [18]. This may lead to frustration from both the student and teacher.

Wu et al [25] claims that many of the identified features may not be unique to AR, as they could be found in other environments such as mobile learning. Therefore, it is important to not only explore the capabilities of AR technology, but also how its use could be aligned with different instructional approaches. There are many instances of successful AR applications in other educational domains. For example, Augmented Chemistry uses a Tangible User Interface to compose 3D molecular models [27], or Augmented Reality in Surgery (ARI*SER) for surgical training [28]. LearnAR includes many activities to teach introductory English, mathematics, science, physical education and languages [18], while others helped children categorize animals and modes of transport [29]. AR Books, such as MagicBook, are also becoming popular [30]. Finally, Construct3D for Mathematical Education allows multiple users, such as teachers and students, to collaborate in constructing virtual, geometric shapes [30] [28].

IV. The analogy

This section describes an analogy[1] that can be used to help turn the abstract parallel programming concepts into concrete mental models. The execution of a parallelized program is represented by an office with employees (or contractors) working towards a common goal. Within this overarching analogy, alignments are made between the office environment and specific technical concepts:

- The **office space** represents the **system hardware**.
- Each **desk** represents a **processor core**.
- An **employee/contractor** represents a program **thread**.
- **Hiring** an employee represents **creation** of a thread.
- **Releasing** an employee represents **killing off** of a thread.
- A **piece of paper** represents a **computational task**.
- A **filing cabinet** represents the **central location** of ready-to-execute tasks.

Since a processor core can only handle one thread at a time, there is only one seat per desk. This limits the number of office workers at each desk to one. As an example, the analogy assumes there are four desks in an office (i.e. a quad-core system). If the program is running sequentially, this means only one employee is hired to sit at one of the four desks to complete all the work. The worker is released once the work is completed, which represents the thread being killed off.

However, if the same program is parallelized, the first employee hired (the main thread) will proceed to hire other employees/contractors. This is analogous to creating and starting new threads. The number of threads created depends on the scheduling policy. Regardless, the main thread must wait for these spawned threads to finish their work. In the analogy, the main employee will fall asleep on the office couch while the other employees proceed to work.

Once any thread is created and assigned a task, it is ready to begin working. However, it cannot progress on a task until a processor core is available. This means that a queue might form in the office; once a seat at a desk opens up, the next employee in the queue will sit down and continue their allocated task(s). When there are more threads than there are processor cores, the program is executed by interleaving the work of each thread via time-sharing slices. If it does not complete its work during a time slice, it is paused to allow another thread to progress. So in the office, an employee may only accomplish part of their task before being placed in the back of the queue to relinquish their desk for another worker. This is especially evident in a fully-parallel scheduling policy, where a new thread is allocated for every task (wasting lots of time due to context switching).

A dynamic scheduling policy relies on a central location for storing and assigning tasks to threads during runtime. The analogy uses a filing cabinet to store these tasks, which are represented by pieces of paper (with instructions for the workers on them). When a worker is freed up, they are allocated a task from the filing cabinet (without needing to get off the desk).

The analogy is programming language agnostic, so that it applies to the basic concepts rather than any specific language. While this analogy is at the heart of the ParallelAR app (Section VII), it is also a standalone teaching resource to help explain core concepts to students. The next section introduces the overarching learning outcomes that are addressable by the analogy, with the ParallelAR app further helping in bringing the analogy to life.

V. Learning outcomes

The overarching learning objectives targeted by the resources presented here are inspired by the Curriculum Initiative on Parallel and Distributed Computing [10]:

- *Comprehend that having access to more processors does not guarantee faster execution (Section 8.4, page 28).* A system with only one thread, or one that is inherently sequential, does not benefit from adding more processors.
- *Understand the effects of load imbalances on performance, and ways to balance load across threads or processes (Section 8.3, page 25).* Ideally all threads finish at the same time, however, load imbalances occur if some threads are working longer than others, increasing the overall time of execution.
- *Understand the basic notions of static and dynamic scheduling, mapping and impact of load balancing on performance (Section 8.3, page 24).* A static scheduler allocates tasks before the threads start, while a dynamic scheduler allocates tasks at runtime.
- *Understand different ways to assign computations to threads or processes (Section 8.3, page 24).* Computations can be assigned using a static or dynamic manner, using a fully-parallel approach or recycling of threads via thread pools.
- *Recognize time as a fundamental computational resource that can be influenced by parallelism (Section 8.4, page 27).* Time complexity can be reliably reduced by splitting tasks among separate threads.

In addition to the above LOs reinforcing the PDC CI, the following LOs have also been identified:

- *Understand what overhead is and how it impacts the performance of a parallelized system.* Overhead refers to the amount of time required to coordinate parallel

[1]A full print version is available from the ParalleAR website.

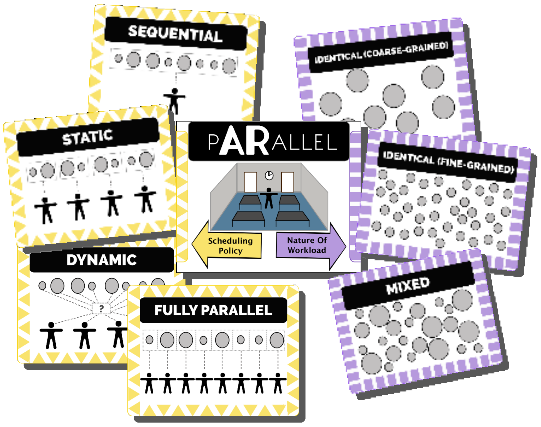

Fig. 2. The full set of flashcards includes four scheduling policies and three workload types. One of each is selected, to facilitate performance discussions.

tasks, as opposed to doing useful work. It ultimately has a negative impact on performance.

- *Recognize the importance of fully utilizing each core in a multi-core system.* Parallelization is most effective and efficient when all processor cores are being used.
- *Understand the impact of a parallel system managing more threads than cores.* A system interleaves execution of tasks according to a time slice, possibly resulting in high rates of context switching.
- *Recognize the benefits of parallelization opposed to a sequential program.*

VI. Collaborative learning with flashcards

Collaborative learning refers to knowledge construction via social interaction, and has the support of many years of research. Students work together in small groups toward a common goal; consequently, students are responsible for one another's learning as well as their own such that the success of one student helps benefit the other students**.** As mentioned in Section II-B, benefits of collaborative learning include critical-thinking skills, information retained for longer, and reduced anxiety associated with problem-solving.

With this in mind, Figure 2 shows the set of learning flashcards that have been designed to encourage collaborative learning to achieve the LOs outlined in Section V (although, of course, they may still be used non-collaboratively). Traditional flashcards are known to engage active recall and are a good learning tool when used properly. Proper use entails actively trying to guess the back side while looking at the front, rather than blankly reading information [31]. Combining this activity with an environment for students to discuss their reasoning, this promotes active recall and knowledge construction.

Strictly speaking, flashcards are traditionally two-sided, where one side has the prompt and the other side has the solution. A classic example is that of studying a new language; the first side has the native language word and the opposite side reveals the translated word. However, our flashcards only have one real side. The "solution side" can come in one of two forms, depending on whether this activity is paired with the ParallelAR app presented in Section VII. If it is, the second side of the flashcards are the virtual animations, enabled by the AR app. Otherwise, the solutions provided by a teacher would stand in their place.

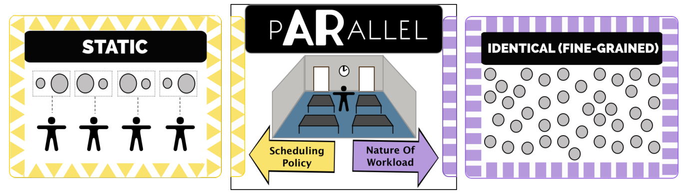

Fig. 3. An example of a possible flashcard combination, using *Static* scheduling for *Identical (Fine-grained)* tasks. Here, students could discuss that high core utilization and good load balancing is achieved, with minimal runtime overhead.

Figure 3 displays an example selection of the flashcards. In the center is the main flashcard, prompting for a pairing of one scheduling policy and the nature of the workload. Each type of scheduling policy flashcard has a yellow and triangle border, likewise, each nature of workload flashcard has a purple and striped border. This design is to clearly differentiate between the two types of flashcards without requiring them to be printed in color. The face of the flashcards have a simple illustration that mimics the analogy described in Section IV to describe what it represents. For instance, the *Static* scheduling policy flashcard depicts which tasks are assigned to which threads (employees). Similarly, the *Identical (Fine-grained)* flashcard shows many small identical tasks. The entire set is shown in Figure 2, and may be downloaded for printing from the ParallelAR website.

A. *Recommended workflow*

Teachers can use these flashcards in a couple of different ways to promote active learning. After being given a basic explanation of these parallel concepts, students could then be asked to work together to determine which scheduling policy will be the most efficient for a given workload. This collaborative activity would ideally generate some discussion and debate, before receiving the correct solution by the instructor. Alternatively, students could discuss (in groups) the anticipated behavior depending on the combination of workload and scheduling policy. To attain the intended LOs, students would need to compare a couple of different configurations. For the latter, a Recommended Workflow has been prepared to guide the learning process with these flashcards; a short snippet is provided in Figure 4, also explaining the flashcards.

The Recommended Workflow provides a suggested ordering in which the combinations should be presented, interleaved with prompt questions to encourage the correct type of discussion, as well as the learning outcomes the activities are targeting. The suggested workflow gradually introduces the different scheduling policies, motivating the need for different policies. For example, the first steps illustrate how a *Fully Parallel* policy (a thread per task) is fine when there are few

Nature of Workload

There are three types of computational workloads:

- Identical (Coarse-grained) There are a total of 8 tasks. Each task is identical, and each task takes 4 seconds to complete.
- Identical (Fine-grained) There are a total of 40 tasks. Each task is identical, and each task takes 1 second to complete.

...

Scheduling Policy

There are four types of scheduling policies:

...

Sample workflow, prompt questions, learning outcomes

1. To get the base performance for coarse-grained tasks, combine the **Sequential** and **Identical (Coarse-grained)** flashcards.
   (a) How much overall time do you think it will take to complete all tasks?
   (b) What do you think the core utilization will be? Estimate a numerical value for the utilization of each core.
2. Now replace **Sequential** with **Fully Parallel** to see the impact of parallelization.
   (a) Do you think there will be an improvement in the overall time? How much?
   (b) Compare each core utilization in this round compared to the previous round.
   (c) ***Learning outcome:*** appreciate the potential of parallelization.
3. To get a base performance for fine-grained computations, select the **Sequential** and **Identical (Fine-grained)** flashcards.
   (a) How much overall time do you think it will take to complete all tasks?
   (b) What do you think the utilization will be for each core?
4. Now replace **Sequential** with **Fully Parallel**, keeping **Identical (Fine-grained)**.
   (a) How much overall time do you think it will take to complete all tasks?
   (b) How does this compare to the parallelization of the **Identical (Coarse-grained)** computations in steps 1 and 2?
   (c) Compare each core utilization in this round compared to the previous rounds.
   (d) Is the core utilization here more or less than when we used **Fully Parallel** with **Identical (Coarse-grained)**?
   (e) ***Learning outcome:*** appreciate the overhead of creating and scheduling a large number of small tasks. With an excessive number of threads, the system may struggle with context switching on the limited number of cores.
5. Now replace **Fully Parallel** with **Static**, keeping **Identical (Fine-grained)**.
   ...

Fig. 4. Snippet from the Recommended Workflow document that explains the flashcards, providing students with guided instructions. Prompt questions and learning outcomes are interleaved to further aid meaningful discussions. The ready-to-print document is available to download from the ParallelAR website.

coarse-grained tasks. However, the following steps illustrate how this is insufficient with many fine-grained tasks. Performance is rectified with a *Static* thread-pool policy, where there are only as many threads as there are processor cores.

The Recommend Workflow and flashcards are provided in a manner to support a collaborative learning activity to get students to think about how scheduling policies would affect performance. While using the ParallelAR app is not essential here, it could be used after students discuss the prompt questions to "experience" the scenarios. Even without the app, a teacher could hand out multiple copies of the flashcards to students in a classroom, and ask the students to discuss in groups. They could then come together and the teacher would ask the students to share their thoughts.

VII. The ParallelAR app

In order to support the analogy and collaborative learning elements of the previous sections, the ParallelAR app is built using AR technology. Figure 5 shows how the app detects and tracks image targets (in this case flashcards), and augments them with virtual enhancements. In this case, the value-added benefit includes watching the analogy come to life (the animation that plays out depends on the flashcard combination), and a summary of the resulting performance. This in-app summary also serves as "answers" to the flashcard combinations, by including the overall task completion times and processor core utilization (as prompted in the Recommended Workflow).

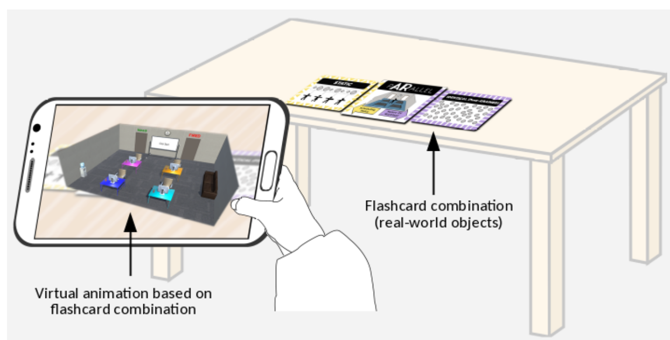

Fig. 5. The ParallelAR app uses the smartphone or tablet's camera to view the flashcard combination. The flashcards are used as real-world objects to trigger virtual behavior on the device (an animated simulation of the analogy for the respective flashcard combination, and summary of the resulting performance).

The primary purpose of the flashcards are to be used as a learning resource to aid collaboration. When combined with the Recommended Workflow, this supports the focus on the identified parallel scheduling LOs. With the ParallelAR app, the student experiences the different combinations in real-time instead of passively being told the pros and cons of different scheduling policies. The inherent nature of AR technology fits perfectly with the purpose of analogies. At its core, AR is about augmenting the real world with virtual elements; analogies are likewise about using real-world stories

Fig. 6. Screenshot of ParallelAR executing (real-world background on mobile screen is cropped out), after combining the *Fully Parallel* and *Identical (Coarse-grained)* flashcards. With eight tasks (numbered *a* to *h*) assigned to eight threads (numbered *1* to *8*) and only four cores, some context switching occurs. The main thread (numbered *0*) is sleeping (on the couch).

to describe virtual or abstract concepts.

The major benefit of the combination of these tools – the AR app, the analogy, and the flashcards – is their accessibility in the classroom. The app is built using low-poly 3D assets, a technique that shows a reduction in file sizes, mesh-density and loading time on Android devices [32]. In this regards, most Android (version 4.1.x+) smartphones and tablets can handle the animations without any problem. One only needs a basic Android device and a black and white printer to participate. Even so, if printing is an issue, the application can recognize the flashcards displayed on another screen (albeit not as well). All the required resources may be downloaded from the ParallelAR website[2].

A. Visualization of the analogy

After the flashcard combination has been successfully recognized by the app, a 3D model of an office will be superimposed onto the main flashcard in the smartphone's camera view. The virtual office is shown in Figure 6, where the number of desks is fixed to four to represent a quad-core system. In this example, the flashcard combination represents the *Fully Parallel* scheduling policy with the *Identical (Coarse-grained)* task workload. This means the main thread will create a new thread for each of the eight tasks to be completed. As in the analogy of Section IV, there is a door labeled "Hired" from which employees (new threads) enter the scene, and one labeled "Released" from which employees exit the scene.

The couch provides a natural place for the main thread to sleep, while waiting for the remaining eight threads to finish their work. As there are more threads than available cores, some context switching occurs. This lowers the core utilization slightly while the threads execute in time slices (i.e. the allocated tasks will not be completed in one time slide). At the time of the screenshot, Core 2 is not highly utilized as thread 5 is about to be switched out with thread *4* (which is making its way towards to Core 2). The water cooler represents a place for threads to queue up while waiting for another core to free up (e.g. threads *6*, *7*, *8*). While subtle, the clock on the back wall continuously spins to represent the passing of real time; this illustrates a common concept when discussing a parallel program – the "wall clock". When the scheduling is *Dynamic*, a filing cabinet with a stack of papers on top (described in Section IV) appears in the center of the room.

The selected workload is always depicted in the app as a "task bar" across the top of the screen, as shown in Figure 6. Each computation that needs to be done is represented by an empty circle. As the computation is completing, the circle fills up. The face of the employee who is assigned to a given computation will appear in the center of the circle. Computations that are not yet assigned are without such a face (in the case of *Dynamic* scheduling). Only a small number of tasks can be worked on at once, depending on the number of threads and cores. When a task is actively in progress (e.g. tasks *a*, *b* and *c* in Figure 6), it is surrounded by the color that matches the core color where the assigned thread is currently placed. This task bar provides visual cues as to how the scheduling policy effects the assignment and order of computations completed.

B. Statistics summary screen

During each simulation run, the overall core usage is displayed (bottom-right corner of Figure 6). The core utilization momentarily drops when threads are context switching on that core (e.g. Core 2). The core utilization is also 0% when no threads are executing on it, as would be the case for three of the four cores if a *Sequential* scheduling policy was used. A low core utilization suggests a possible improvement in the utilization of the system. The students should be able to feel the difference in time of the different runs, as it mirrors real-world time for the simulation to play out. To emphasize these performance measures, a summary screen of the statistics, similar to the one in Figure 7, will be displayed after every run. This summary also includes an average core utilization for each processor core, and an overall system utilization (the average of the core utilizations). The statistics summary screen plays an important roles in helping students confirm their predictions for the combination of flashcards.

VIII. Conclusions

An analogy is a powerful tool to help students relate unfamiliar and abstract concepts to something they are already familiar with. Augmented reality (AR) is the ideal technology platform for an analogy to come to life; like an analogy, AR is inherently about using the real world to build (apply) a virtual (abstract) experience (concept). However, analogies and technology alone are not necessarily sufficient in promoting an active learning activity. For this reason, a collaborative learning activity has been designed and presented.

Following a constructive alignment approach, learning outcomes pertaining to fundamental parallel programming

[2] http://parallel.auckland.ac.nz/education/parallelar

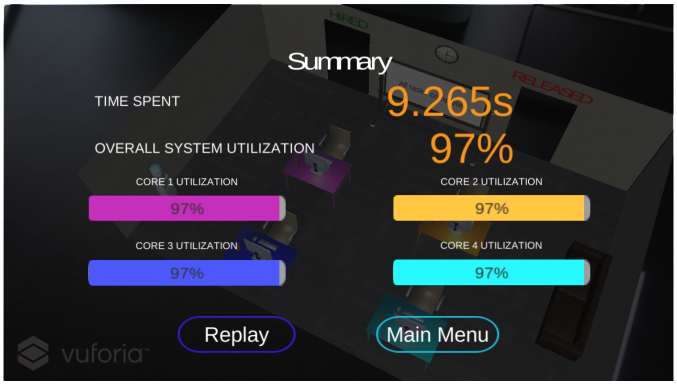

Fig. 7. Screenshot of the ParallelAR statistics summary screen, which is displayed after a run through of Figure 6. Since there were eight threads alternating the coarse-grained tasks, the workload was balanced but there was still some slight overhead due to context switching the threads.

scheduling concepts have been identified from the Curriculum Initiative on Parallel and Distributed Computing. A set of flashcards have been designed and provided, with a Recommended Workflow document to assist completing the activity. Students are encouraged to discuss their anticipated answers, and are provided prompt questions at different stages of the activity. The answers may be provided by the teacher, or the students may use the ParallelAR app to watch the analogy come to life. Performance statistics are summarized at the end of the simulation, to help students self-assess their assumptions to the prompt questions from the collaborative activity.

The collaborative activities and app have been created in a manner to allow future extensions. By adding more flashcards to the collection, this will increase the number of scenarios and lessons students can learn from. For example, another set of flashcards could be added to represent the number of processor cores, or using chunk sizes greater than one when allocating tasks dynamically. This way, the students would have more decisions to make when creating their system using the flashcards, more closely mirroring the process of implementing a parallel program in reality.

References

[1] L. Porter, S. Garcia, H.-W. Tseng, and D. Zingaro, "Evaluating student understanding of core concepts in computer architecture," *Proceedings of the 18th ACM conference on Innovation and technology in computer science education - ITiCSE '13*, p. 279, 2013.

[2] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bulletin*, vol. 37, no. 3, pp. 14–18, 2005.

[3] H. Sutter, "The free lunch is over: A fundamental turn toward concurrency in software," *Dr. Dobb's Journal*, pp. 1–9, 2005.

[4] I. Blanchette and K. Dunbar, "How analogies are generated: The roles of structural and superficial similarity," *Memory & Cognition*, vol. 28, no. 1, pp. 108–124, 2000.

[5] P. R. Simons, "Instructing with analogies," *Journal of Educational Psychology*, vol. 76, no. 3, p. 513, 1984.

[6] B. G. Silverman, "Computer supported collaborative learning (CSCL)," *Computers Education*, vol. 25, no. 3, pp. 81–91, 1995.

[7] Y. J. Dori, J. Belcher, and J. Belcher, "How does technology-enabled active learning affect undergraduate students' understanding of electromagnetism concepts?," *Journal of the Learning Sciences*, vol. 8406, no. November, 2017.

[8] M. Warschauer and T. Matuchniak, "New technology and digital worlds: Analyzing evidence of equity in access, use, and outcomes," *Review of Research in Education*, vol. 34, no. 1, pp. 179–225, 2010.

[9] Pew Research Center, "Mobile fact sheet." http://www.pewinternet.org/fact-sheet/mobile/, 2017.

[10] S. K. Prasad, A. Chtchelkanova, M. G. F. Dehne, A. Gupta, J. Jaja, K. Kant, A. L. Salle, R. LeBlanc, A. Lumsdaine, D. Padua, M. Parashar, 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://www.cs.gsu.edu/~tcpp/curriculum, 2012.

[11] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: a class of displays on the reality-virtuality continuum," *Telemanipulator and Telepresence Technologies*, no. 2351, pp. 282–292, 1994.

[12] M. Billinghurst and A. Duenser, "Augmented reality in the classroom," *Computer*, vol. 45, no. 7, pp. 56–63, 2012.

[13] J. Peddie, "Types of Augmented Reality," in *Augmented Reality*, pp. 29–46, Springer, 2017.

[14] A. Katiyar, K. Kalra, and C. Garg, "Marker Based Augmented Reality," *Advances in Computer Science and Information Technology*, vol. 2, no. 5, pp. 441–445, 2015.

[15] K. Panetta, "Gartner's top 10 strategic technology trends for 2017." https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017, 2016.

[16] CB Insights, "AR vs VR financing: VR leads but AR starts narrowing the gap." https://www.cbinsights.com/research/ar-vr-funding-trends, 2017.

[17] M. Rougeau, "Why virtual reality developers think AR is going to be bigger than VR." https://www.digitaltrends.com/gaming/ar-will-be-bigger-than-vr-according-to-developers, 2017.

[18] M. Bower, C. Howe, N. McCredie, A. Robinson, and D. Grover, "Augmented reality in education - cases, places, and potentials," *Proceedings of the 2013 IEEE 63rd Annual Conference International Council for Education Media, ICEM 2013*, vol. 51, no. 1, pp. 1–15, 2013.

[19] K. J. C. Williams, J.-M. Lowendahl, and G. Morgan, "Top five strategic technologies impacting K-12 education in 2017." https://www.gartner.com/doc/3558520/top-strategic-technologies-impacting-k, 2016.

[20] C. Powell, Katherine and J. Kalina, Cody, "Cognitive and social constructivism: Developing tools for an effective classroom," *Education*, vol. 130, no. 2, pp. 241–250, 2009.

[21] M. Prince, "Does active learning work? A review of the research," *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.

[22] A. A. Gokhale, "Collaborative learning enhances critical thinking," *Journal of Technology Education*, vol. 7, no. 1, 1995.

[23] J. Biggs, "Enhancing teaching through constructive alignment," *Higher Education*, vol. 32, no. 3, pp. 347–364, 1996.

[24] J. Bacca, S. Baldiris, R. Fabregat, and S. Graf, "Augmented reality trends in education: A systematic review of research and applications," *Educational Technology & Society*, vol. 17, no. 4, pp. 133–149, 2014.

[25] H. K. Wu, S. W. Y. Lee, H. Y. Chang, and J. C. Liang, "Current status, opportunities and challenges of augmented reality in education," *Computers and Education*, vol. 62, pp. 41–49, 2013.

[26] W. Tarng and K.-L. Ou, "A study of campus butterfly ecology learning system based on augmented reality and mobile learning," *2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education*, pp. 62–66, 2012.

[27] M. Fjeld and B. M. Voegtli, "Augmented chemistry: An interactive educational workbench," in *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pp. 259–321, IEEE, 2002.

[28] G. Chang, P. Morreale, and P. Medicherla, "Applications of augmented reality systems in education," *Technology & Teacher Education*, vol. 2010, pp. 1380–1385, 2010.

[29] R. Freitas and P. Campos, "SMART: a SysteM of Augmented Reality for Teaching 2nd grade students," in *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction*, pp. 27–30, BCS Learning & Development Ltd., 2008.

[30] K. Lee, "Augmented reality in education and training," *TechTrends*, vol. 56, no. 2, pp. 13–21, 2012.

[31] D. Glenn, "Close the book. Recall. Write it down.," *The Chronicle of Higher Education*, vol. 55, no. 34, pp. 1–5, 2009.

[32] P. Bhawar, N. Ayer, and S. Sahasrabudhe, "Methodology to create optimized 3D models using blender for Android devices," *Proceedings - 2013 IEEE 5th International Conference on Technology for Education, T4E 2013*, pp. 139–142, 2013.