Passive or active learning: the challenges of teaching distributed computing using Raspberry Pi clusters to open distance university students

Daniel Gooch, Jon Rosewell, Douglas Leith & Mike Richards

To cite this article: Daniel Gooch, Jon Rosewell, Douglas Leith & Mike Richards (2022): Passive or active learning: the challenges of teaching distributed computing using Raspberry Pi clusters to open distance university students, Open Learning: The Journal of Open, Distance and e-Learning, DOI: 10.1080/02680513.2022.2118573

To link to this article: https://doi.org/10.1080/02680513.2022.2118573

© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

Published online: 01 Sep 2022.

Article views: 26

View related articles

View Crossmark data
Passive or active learning: the challenges of teaching distributed computing using Raspberry Pi clusters to open distance university students

Daniel Gooch, Jon Rosewell, Douglas Leith and Mike Richards

School of Computing and Communications, The Open University, Milton Keynes, UK

ABSTRACT
Parallel and distributed computing (PDC) is now considered a threshold concept for computing, and is embedded in computing curricula across the globe. While the costs of traditional computing clusters have made developing practical activities challenging, the rise of low-cost computers, particularly the Raspberry Pi, has led to an exploration of how PDC can be taught to students using Raspberry Pi clusters. Building on this work, we report our experiences from developing a series of low-cost Raspberry Pi clusters for use with open distance university students. Based on survey results from 484 students, we argue that our work demonstrates the benefits that remote practical activities can have for teaching PDC concepts, as well as engaging students. We conclude with a discussion of two key challenges: supporting active learning through student-led programming on the clusters, and supporting lower-performing students at a distance.

KEYWORDS
Raspberry Pi cluster; distance learning; distributed architectures; parallel; CS education

Introduction and background
Parallel and distributed computing (PDC) is now considered a threshold concept for computing (Cousin, 2006; John & Thomas, 2014), forming a key part of the ACM/IEEE Curriculum Guidelines for Undergraduate Degree Programmes in Computer Science (Association for Computing Machinery (ACM), 2020). To understand this concept, students need practical exposure to specialist hardware – a computing cluster.

However, this has posed a challenge for academic institutes, with the cost of purchasing and running a cluster for teaching purposes being too expensive for most institutes (Abrahamsson et al., 2013; Adams et al., 2016; Pfalzgraf & Driscoll, 2014). While hosted virtual clusters provide one alternative avenue, by using hosted clusters, students are removed from the physical hardware, which reduces their ability to utilise ‘hands-on exercises in which students must apply abstract concepts [to] improve their understanding of those concepts’ (Adams et al., 2016, p. 83), as well as abstracting away various practical concerns with computing clusters (Cox et al., 2014).

The low cost of Raspberry Pis (https://www.raspberrypi.org) and other single-board computers has led to an exploration of how PDC can be taught to students using clusters built
from these platforms. This has ranged from helping students build their own clusters (Doucet & Zhang, 2017) to benchmarking a large 64-node cluster, designed for teaching (Cox et al., 2014). Clusters containing as many as 300 Raspberry Pis have been developed, with the advantage of lower costs being seen as essential to making such devices available for teaching (Abrahamsson et al., 2013). Others have focussed on benchmarking their developed Raspberry Pi clusters, demonstrating that they display appropriate PDC behaviour and can therefore be used in teaching (Pfalzgraf & Driscoll, 2014). The interest in using these clusters to teach students about PDC has even led to SIG meetings (e.g. (Adams et al., 2016)). As yet, there has been no reported use of Raspberry Pi clusters for open distance university students.

There is some evidence that the use of Raspberry Pis can both motivate students and improve their understanding of PDC concepts. For example, Matthews et al. report on a series of workshops on teaching PDC concepts using Raspberry Pis, and note an increase in understanding of PDC concepts from before and after questionnaires (Matthews et al., 2018). Carratalá-Sáez et al. (2019) report on an optional CS course focussed around a Raspberry Pi cluster, and note that the platform was both engaging for the students, and flexible enough to be used to effectively teach PDC concepts.

The authors all work at an open distance university, where there are additional challenges to making use of Raspberry Pi clusters to teach PDC. Firstly, the profile of our students is somewhat different from what is typical at a face-to-face institute. The majority of our students are studying part-time, meaning there can be significant time gaps between students being introduced to a topic and having to use and develop that skill (Kim et al., 2013; Smith et al., 2018). Many of our students are mature students (only 10% of undergraduates are under the age of 21), with 72% already in work. Due to the open nature of admissions, some of our students are complete novices with no prior computing experience, while others are practitioners with many decades of experience. This learner profile makes it difficult to provide additional content to support motivation, necessitating that the core teaching material is as engaging as possible.

Secondly, the distance nature of our institute makes it challenging to provide direct access to hardware, particularly when it is neither time nor cost effective to build a cluster for each of the 1500 students on an introductory year 1 course.

We thus undertook a project to explore:

(1) Is it possible to provide remote access to a Raspberry Pi cluster?
(2) Does such access motivate, engage and educate the students who use the cluster?
(3) What are the benefits and challenges of developing and deploying such a tool?

Through these questions we contribute an experience report of our attempt to utilise Raspberry Pi clusters to teach PDC concepts to open distance university students. We contribute findings that highlight the value of the clusters in supporting the engagement of our students, as well as outlining challenges for other practitioners wanting to make more use of physical hardware as teaching tools.

Module context

This work took place in the context of a first-year University module at The Open University, titled ‘Technologies in practice’ (TM129). The module is compulsory for
students studying any of the named degrees from the School of Computing and Communications, and is optional on other qualifications. The module has no prerequisites. The module is presented twice a year (starting in October and February). On average, around 82% of the students are male. Flexible study patterns mean that a significant minority of our students (18%) are studying the module having not previously studied at The Open University. Around 73% are studying on a named degree from the School of Computing and Communications.

The module covers three different topics, split into blocks, covering Robotics, Networking and Operating Systems. The module is expected to take 300-hours’ total study time (30 credits), split evenly across each of the three blocks. The Operating Systems block uses Raspberry Pi Desktop O/S to teach the fundamentals of Operating Systems. Updated for the 2020 October cohort, the module introduced content on multitasking and distributed computing within the O/S block, with the clusters forming the bulk of the practical activities around this topic.

Construction of the clusters

Our Raspberry Pi clusters were constructed based on the OctaPi instructions, released under a Creative Commons Licence from GCHQ (see https://projects.raspberrypi.org/en/projects/build-an-octapi). Eight Raspberry Pis are connected in a private network, using a router and a switch. One of the Raspberry Pis acts as the lead, while the other seven Raspberry Pis are used as servers, to provide results back to the lead Raspberry Pi. Programmes are written in Python and run on the lead Pi; the dispy package is used to distribute activities across the cores in the cluster. Three programmes have been developed for the cluster, available on GitHub (https://github.com/dg7692/TM129).

Two of the programmes are linked to search problems introduced earlier in the module – the travelling salesperson and password hashing. As complete search problems, both of these are ideal for teaching PDC concepts.

We wanted to include a programme that was graphical, providing a contrast to the other two text-based programmes. The image combiner programme takes as its input three images, with non-overlapping obstructions. By comparing the RGBA values pixel-by-pixel across the three images, and selecting the median, an image is constructed without any obstructions (as the median will always be one of the two images without the obstruction). 12 triplets of images have been produced; 10 are photographs and 2 are digitally drawn images. They have then been scaled to four sizes (full [7416 × 3136], mid [1854 × 784], small [1112 × 470], and tiny[742 × 314]). As an example, Figure 1 shows a pair of Christmas ducks, obstructed by an Easter duck. All of the images are available with the programmes on GitHub.

Student activities

Our approach to developing the student activities was inspired by Ghafoor et al. (2019) and their work on designing plugged and unplugged activities to support the teaching of PDC concepts.

For the travelling salesperson programme, students were posed with three activities:
(1) Calculate the number of comparisons needed to find the minimal distance between 5 and 7 locations. Run the cluster using 14 cores, and time how long it takes to find the best route for 5 locations. Estimate how long the cluster will take to visit Y locations. Check your calculation by using the cluster.

(2) Repeat the previous activity, using 8 cores and 28 cores. Estimate how much quicker the programme runs using each additional core. Students can optionally view a discussion from the module team which highlights that ‘While each core can process data, there are various overheads in running the algorithm in parallel. These overheads include the time needed to distribute the data across the cluster through the LAN, and having to share various system resources across the cores on a single Raspberry Pi. Additionally, for a larger number of cores, the algorithm has to split the list of all routes into a large number of sub-lists, and check a larger number of potential solutions, all of which takes time.’

(3) Here is the outline for the algorithm. Making one simple change, how could you make the algorithm twice as fast? Students can optionally view a discussion from the module team which highlights that ‘In this form of the travelling salesperson problem, the distance between two locations is the same regardless of ordering – so travelling from location 1 to location 2 is the same as travelling from location 2 to location 1. This means that the list of all possible routes is twice as long as it needs to be: the distance of travelling route 1-2-3-4-5-6-7-8-9-1 is the same as the distance of travelling route 1-9-8-7-6-5-4-3-2-1. Therefore you can make the algorithm twice as fast by halving the size of the list of all possible routes. This can be achieved by removing all reverse-ordered routes.’

For these activities, students do not need to access the code, with the activities focussed on developing an understanding of the limitations of parallelism in speeding up computations.

For the image combiner programme, students were given two activities:

Figure 1. Example image for the image combiner script, with the Easter duck obstructing the Christmas ducks.
(1) Given the single threaded algorithm, produce the pseudo-code for the parallel algorithm. Students can optionally access the pseudo-code from the module team.

(2) ‘Ghosting’ can appear in some of the outputs, where artefacts of the obstructions remain. A more complex algorithm would also compare the pixel to its neighbours. Why is this more challenging in a parallel algorithm? Students can optionally view a discussion from the module team that explains ‘The “ghosting” is an artefact of the conditions of the images. Digital photographs will be affected by the lighting conditions surrounding the objects when they are photographed, as well as the shadows cast by the objects in the photograph. Images created on a computer are not affected by lighting conditions and are exact duplicates, meaning the “ghosting” never occurs. The main difficulty with comparing a pixel to its neighbours using the cluster is that on the cluster the image is divided. Therefore, to compare all of a pixel’s neighbours, additional rows and columns must be distributed across the cluster, and subsequently filtered out when the output image is being constructed. While possible, it is an additional complication.’

Finally, for the password hashing program, students were given two activities:

(1) Collaboratively collect the timing data for different password and number of core combinations. Using the crowd-sourced data, plot a graph of speed-up factor against the number of cores in the cluster. Summarise and interpret your results.

(2) Having outlined the algorithm to students, they are asked about the assumption made in the algorithm about the hash function and whether the assumption makes the existing algorithm limited.

(3) Students are asked what are the two key limitations of the parallel algorithm? Students can optionally view a discussion from the module team that explains ‘As soon as the algorithm finds a pair of words that produces a matching hash, it can stop looking: there is no point in looking for additional matches by searching the rest of the dictionary. At the moment, the code has a variable called “success” but this is not used to exit the searching. If the “success” variable was used to exit the searching, and the pair of words appeared early in the dictionary, then one task would find them and could return much more quickly. However, if the programme as a whole waits for all tasks to complete, then the overall performance will not improve. In that case, a single Raspberry Pi would find the values quicker than the cluster.’

Sufficient explanation was required when describing each of the activities such that it was not necessary for students to access the code to understand or answer any of the questions.

Accessing the clusters

To provide remote access to the clusters, we made use of the OpenSTEM Labs infrastructure at our university (Richardson et al., 2019). 10 clusters (eight built with Pi 4s, two built with Pi 3B+s) were installed into racks, with webcams pointed at each clusters. Students could use the booking infrastructure to secure a 25 minutes slot on a cluster, and could book as many slots as they desired.
Students accessed the clusters through a web interface (see, Figure 2). This allowed students to select which programme they wanted to run, the number of cores on the cluster to use, and set the parameters for the selected programme. As output, the student sees the time the programme took to run on both an individual Raspberry Pi, and on the cluster using the number of cores selected. The student also saw the output from the programme, either the password hashing result, the minimal and maximal travelling salesman route, or the non-occluded image.

The webcam showed a live stream of the cluster. The lead Pi was equipped with an LED display to show the state of the programme as it runs. The webcam makes it clear to students that they are conducting an experiment with real dedicated hardware rather than receiving simulated or pre-recorded results.

As the web interface is heavily integrated into the booking system and infrastructure associated with the OpenSTEM Labs, it is neither appropriate nor possible to provide access to the code base.

![Figure 2. Screenshot of the interface used to access the cluster programs. The screenshot is demonstrating the image combiner program.](image)
As an indication of the speed at which the cluster operates, Figures 3 and 4 show benchmark times for each of the programmes, with the travelling salesman programme being run on both four and nine locations, and the image combiner programme being run on a tiny, small and mid-sized image.

The cluster has 28 cores to use at any one time (seven Raspberry Pis with quad core processors). The small increase in time for 32 cores illustrates how simply splitting the program into lots of chunks and distributing them is costly, mainly due to networking demands and the need to process/combine the results on the lead Raspberry Pi.

Method

We had two goals in developing these clusters. The first was related to the necessity of using clusters to teach PDC concepts in an effective manner, using real tools. The second was to use the clusters as a mechanism for increasing engagement in the teaching material, and interest in Computer Science in general.

One of the cluster activities was flagged as an ePortfolio activity, one of six across the O/S block. Students have to report on two of the six ePortfolio activities as part of their coursework. It was thus possible for students to avoid being assessed on the cluster activities by submitting alternative ePortfolio activities. Given the optionality of the activities, we have no way of determining student performance. Therefore, we developed a questionnaire to examine student perceptions of their experience of using the clusters.

Selecting a measure for assessing the efficacy of the clusters was not straightforward. We consulted the csedresearch.org website, which hosts a repository of CS education evaluation measures. We filtered the results by demographic (9th-12th and Undergraduate), year published (2010–2019) and number of questions (1–30). We then
consulted the 38 instruments before selecting two to modify – the NCWIT grade 4–12 computing programme participant post-survey (https://www.ncwit.org/file/grade-4-12-computing-programme-participant-post-survey) and the Computing Attitudes Survey (Dorn & Tew, 2015).

We added three free-text questions to the modified questionnaires which asked ‘What did you enjoy about using the clusters?’ ‘What did you dislike about using the clusters?’ and ‘Do you have any other comments to share with us about the clusters?’.

The clusters were provided to all students in the October 2020 and February 2021 cohorts, covering 993 and 1082 students respectively at the time of use. Completing the questionnaire was a completely optional activity.

Eight students from the February cohort were interviewed to try and elicit information that would complement the questionnaire results. The analysis of the interview transcripts did not provide any insight beyond that gleaned from the questionnaires, and for clarity we thus focus on reporting the questionnaire results.

Our study was reviewed and passed by the Student Research Project Panel (SRPP) which at The Open University is the ethics review board for all projects involving taught students. Informed consent was taken at the start of the questionnaire, and before the phone interviews began.

**Analysis**

The Likert-scale questions included in the questionnaire were analysed through considering descriptive statistics applied to the results from both cohorts.

An inductive open coding approach was used to identify concepts and themes within the responses to the free-text questions in the questionnaire (Braun & Clarke, 2006). The answers were subjected to a line-by-line analysis by the first author. Through this initial
analysis, concepts were identified and labelled within the data. No codes existed prior to the analysis; they were created through constant comparison of the data and the application of labels to the text.

These codes were subsequently categorised into unifying themes by the first author. These themes were then discussed in conjunction with the other authors, to ensure that the developed themes corresponded with their interpretation of the data.

Results

Across the time period the clusters were available, 239 unique visitors from the October 2020 cohort booked a total of 483 sessions on the clusters, with only 22 of those sessions unattended. 125 students completed the questionnaire.

For the February 2021 cohort, 245 unique visitors booked a total of 483 sessions, with 19 not attended. 126 students completed the questionnaire. Table 1 shows the breakdown of the number of sessions on the clusters booked by students.

Given the optionality of the activity, the number of students who used the clusters, at around 25% of each cohort, is an indication of some interest in the clusters from students.

Plotting the use of the cluster over time shows no surprises; a slow start from people studying the material before its official start date, a peak when the study week starts, a further peak halfway through that week, and a peak when the coursework is due. There is steady use of the clusters across the time period they are available to students.

Tables 2 through 4 show the median and (interquartile range) for responses to the Likert-scale questions in the questionnaire.

Focusing first on the clusters themselves, we asked students on a scale of 1 to 5, where 1 is strongly disagree and 5 is strongly agree, how much do you agree with the statements in Table 2.

We then asked students about their skills confidence as per the questions in Table 3; rated on a scale of 1 to 5, where 1 is not at all confident and 5 is very confident.

The final set of questions focussed on study interests as per Table 4; rated on a scale of 1 to 5, where 1 is not at all and 5 is very much.

Our interpretation of these figures is that the cluster activities appear to have been motivational and enjoyable. However, the efficacy of the clusters in terms of making the students competent in solving software problems or developing parallel algorithms is more questionable, potentially due to the passive nature of operating them (i.e. through a fixed interface, rather the programming the clusters directly). We return to this in our discussion.

Table 1. Percentage of visitors from each cohort against the number of sessions booked.

<table>
<thead>
<tr>
<th>Number of sessions</th>
<th>October 2020 cohort percent</th>
<th>February 2021 cohort percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45.2</td>
<td>49.0</td>
</tr>
<tr>
<td>2</td>
<td>30.1</td>
<td>25.5</td>
</tr>
<tr>
<td>3</td>
<td>13.8</td>
<td>14.2</td>
</tr>
<tr>
<td>4</td>
<td>5.4</td>
<td>5.9</td>
</tr>
<tr>
<td>5</td>
<td>2.5</td>
<td>2.9</td>
</tr>
<tr>
<td>6</td>
<td>1.7</td>
<td>0.8</td>
</tr>
<tr>
<td>7–8</td>
<td>0.8</td>
<td>1.7</td>
</tr>
<tr>
<td>11</td>
<td>0.4</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 2. Responses to the questions broadly relating to engagement with the discipline and the clusters.

<table>
<thead>
<tr>
<th></th>
<th>October 2020 cohort</th>
<th>February 2021 cohort</th>
</tr>
</thead>
<tbody>
<tr>
<td>After I completed the first cluster activity, I had difficulty solving the other cluster activities.</td>
<td>1 (2)</td>
<td>1 (2)</td>
</tr>
<tr>
<td>I found the challenge of solving the cluster activities motivating.</td>
<td>3 (1)</td>
<td>4 (1)</td>
</tr>
<tr>
<td>I enjoyed solving the cluster activities.</td>
<td>4 (1)</td>
<td>4 (2)</td>
</tr>
<tr>
<td>I am interested in learning more about parallel and distributed computing.</td>
<td>4 (2)</td>
<td>4 (2)</td>
</tr>
<tr>
<td>I am interested in learning more about computer science.</td>
<td>4 (1)</td>
<td>5 (1)</td>
</tr>
</tbody>
</table>

Table 3. Responses to the questions relating to student skill confidence.

<table>
<thead>
<tr>
<th></th>
<th>October 2020 cohort</th>
<th>February 2021 cohort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solve computer software problems.</td>
<td>3 (1)</td>
<td>4 (1)</td>
</tr>
<tr>
<td>Develop algorithms that work in parallel.</td>
<td>3 (1)</td>
<td>3 (2)</td>
</tr>
<tr>
<td>Solve problems using the clusters.</td>
<td>3 (2)</td>
<td>3 (1)</td>
</tr>
</tbody>
</table>

Table 4. Responses to the questions relating to future study interests.

<table>
<thead>
<tr>
<th></th>
<th>October 2020 cohort</th>
<th>February 2021 cohort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the Raspberry Pi clusters in other OU modules</td>
<td>4 (1.25)</td>
<td>4 (2)</td>
</tr>
<tr>
<td>Take future classes in parallel and distributed computing.</td>
<td>4 (1)</td>
<td>4 (1)</td>
</tr>
<tr>
<td>Take future classes in computer science.</td>
<td>4 (1)</td>
<td>4 (1)</td>
</tr>
</tbody>
</table>

Thematic analysis results

We report the three key themes that emerged from our thematic analysis of the questionnaire results, using illustrative quotes from our students. We provide indicative numbers of how many students voiced a particular view, in the format x(y) to distinguish between October 2020 (x) and February 2021 (y) students.

Exposure to real hardware

One of the clearest themes was that our students enjoyed and were motivated by having remote access to physical hardware, with 26(18) students noting their enthusiasm: ‘It was really interesting to be able to use real clusters instead of having it virtualized’; ‘It was really exciting to be able to actually see a cluster working and see the real effects of working with multiple cores. It was great to be able to try this out for myself, not just to read the theory about it’. We conjecture that providing the live webcam and the cluster’s LED display was important for giving students a sense of access to actual physical hardware.

5 students pursued this further and suggested that providing students with the facility to programme directly on the clusters would have been better than the current setup: ‘This may be beyond the scope of this module but it would be nice to develop our own programmes for use on the cluster to further test our understanding’, with 2 others suggesting we could mail equipment out. 0(1) student suggested use of regional branches and physical access to the clusters.

Engagement with the concept of PDC

12(15) students discussed their interest in parallel computing being piqued by the activities, and the topic area in general. Two of these students also drew on outside
experiences such as setting up servers or running clusters at work, and contrasting those experiences with the low-cost option provided: ‘I have access to high availability clusters in my day job, but these are only typically 2 node clusters and interacting with them is nothing like interacting with the Pi clusters, which I imagine are very similar in use to supercomputers. I’ve always wanted to be able to use a supercomputer in some way, and using the clusters got me a little closer’. 0(4) students found the activities dull.

One of the activities was based around crowdsourcing the limits of the gains of parallelism; students were provided with a shared spreadsheet and a goal of exploring the range of cores and the resulting competition speed for the password hashing programme. By each contributing their data, the students could then collectively investigate the speed-up factor and limitations of providing additional cores. This activity, with its focus on group work (generally disliked by distance students (Sharples et al., 2016)), and shared data analysis approach is different from other tasks they would have completed in the year 1 curriculum.

9(3) of our students discussed this aspect directly, in a positive light, variously arguing: ‘I did enjoy the aspect of gathering the data from the clusters and seeing the different timings and the patterns as the number of cores increased with some unexpected results’ and ‘I liked the last activity. It involved aggregating data in a spreadsheet and finding the pattern’.

17 students commented on expectation setting, with comments considering the length of the activities as being ‘too much reading for someone with dyslexia’ through to not being clear enough on what the students should expect to see: ‘at first I was unsure exactly how it worked, but once I booked a slot I figured it out very quickly’. 2(2) of the students noted that they found the activities challenging; and given the typical bias for completing questionnaires, it is likely that a significant proportion of the 233 students who used the clusters but did not complete a questionnaire also found the activities too challenging.

**Practicalities of accessing the remote hardware**

19(11) students directly commented on the simplicity of accessing the clusters through the OpenSTEM Lab infrastructure, and that such a facility could provide access to computing hardware that otherwise distance students couldn’t use: ‘I enjoyed the ease of use and the interface. I also enjoyed the opportunity to explore parallel computing in this way as I would never have been able to do this at home’. 1 student further noted that the system worked well with screen readers, an important requirement for supporting the accessibility of the material.

**Discussion and conclusion**

Returning to our stated aims, our results clearly demonstrate one method of providing remote access to a Raspberry Pi cluster. The feedback from our students indicates that in some cases the clusters are motivating and engaging. Further work is clearly needed to examine more closely whether use of the clusters is supporting students’ learning of PDC concepts. The use of physical hardware was a novelty to many of these students, and clearly engaged a large proportion of the students who used the system. Our results show similarities with face-to-face students, in that the clusters appear to be effective for motivating open distance students (Carratalá-Sáez et al., 2019; Matthews
et al., 2018). Given the range of benefits that accrue from student engagement (Morgan et al., 2018), we are pleased with the feedback we received, including general praise from the student body: ‘Genuinely some of the best activities I’ve done so far through [year 1]. The direct interaction with remote hardware felt engaging and is something I’d like to see again’. One student was so enthused by the concepts taught that they contacted the module team to discuss their work in simulating their own cluster using VirtualBox on their local machine.

We were more surprised about the positive feedback from multiple tutor groups about the group-work element of crowdsourcing data. Group work is generally disliked by distance students (Sharples et al., 2016), and is something module teams within the School of Computing and Communications tend to approach very carefully. The report from Sharples et al. (2016), highlights how students at The Open University don’t like group work where they are dependent on others, and managing that process for distance students is challenging. Our results indicate that one mechanism for supporting some level of enthusiastic collaboration is by structuring work around a novelty where the stakes are low, in such a manner that no student is dependent on the input of any other specific student. By lowering the burden for successful collaboration, we can provide an introduction to the benefits of collaboration – which incidentally align with the parallelism concepts being taught through the cluster activities.

Focussing on our third aim, our experiences also highlight a number of challenges and future directions for further work.

Firstly, there is a clear challenge of how to extend the project so that students can develop their own software and run it directly on the clusters. While our results show some success in supporting the teaching of PDC concepts, much of the practical aspects of the activities are essentially passive, with students operating them through the selection of inputs from a fixed interface, rather than developing their own programmes directly. Because of the remote access, providing the ability to run code directly on the cluster is a security risk; and if anything goes wrong, it takes time to rebuild the cluster from scratch. One extension we are currently exploring is whether we can create and run a virtual machine across the cluster, providing a sandbox environment in which students can test code. While this would support a more active learning approach, it would remove some of the motivational benefits of running code directly on the hardware. We believe there is an interesting area of work to be explored in comparing and contrasting the benefits and shortcomings of using virtual and physical clusters in teaching PDC concepts, particularly in how to use their pros and cons to develop hybrid teaching methods that make use of both kinds of clusters. Fundamentally, our findings highlight the shortcomings of passive learning approaches when using novel hardware; while they may be engaging, it is harder to assert that they impact on students’ learning.

Our second identified challenge was in supporting students who were struggling with the material. Given the activities were optional and non-assessed, there was no support from tutors. As well as limiting support to students, this also limits the data we have available to assess students’ understanding of the PDC concepts being taught. Furthermore, the distance nature of our students’ learning creates a distance from support. Troubleshooting can be challenging, even for simple problems which would normally be solved by a TA in a lab tutorial (Smith et al., 2018). This is an inherent challenge of trying to provide support at a distance, with no mechanisms for quick bug
fixes or answering clarification questions. Such learning requires a level of resilience from students – which at year one can't necessarily be assumed.

As others have noted, designing effective interventions to support low-performing students in introductory CS courses is challenging, with few clear strategies to use (Liao et al., 2019). While we were trying to use the practical activities to support a conceptual understanding, some of our lower performing students may have benefited more had our focus not been too ingrained within the advantages that practical hands-on activities could provide.

Given the broader need to support resilience, as an important predictor of academic success (Prickett et al., 2020), we argue that the challenge we faced highlights a productive area for further work. By exploring how to appropriately support distanced hands-on practical work to develop resilience, we could develop insights into how to support the resilience of all students, regardless of whether they are studying face-to-face or at a distance.

As a final observation, when reflecting on the project, we note that there are specific challenges of working with open distance students which can slow the adoption of novel teaching practices. The construction of the clusters and developing the activities took time: academics also need to be prepared for the time committed to getting materials in front of students, particularly working through the development of a suitable infrastructure, considering accessibility needs and adapting the materials appropriately, and the time needed for long-term maintenance requirements across multiple presentations a year, each with high numbers of students. Academics and instructors need to be aware of these needs before beginning to deploy such systems, to ensure they have suitable capacity to see projects through to fruition.

Parallel and distributed computing is embedded in computing curricula across the globe. Through our experiences in developing a low-cost Raspberry Pi cluster for use with open distance university students, we have demonstrated the benefits that remote practical activities can have for teaching PDC concepts, as well as engaging students. We believe that the benefits we've observed, and the challenges we've identified, can help academics and instructors in developing their own practices.

Note


Acknowledgments

We would like to thank our reviewers for helping us refine our drafts, and to all of the students on TM129 who have helped contribute to this research.

Disclosure statement

No potential conflict of interest was reported by the author(s).
References


