ascilite
ascilite 2012
FUTURE CHALLENGES | SUSTAINABLE FUTURES
25 - 28 November 2012 | Wellington | New Zealand

Impacts of Scheduling Algorithms on Resource Availability

David Lowe
The University of Sydney

Cost and space constraints typically limit the provision of many educational resources, with laboratory apparatus being a common example. This limitation is often ameliorated by utilizing scheduling techniques to manage access over an extended period of time. The specific scheduling algorithms that are used have been shown to have a significant impact on the overall availability of a set of resources and hence the level of access that can be supported. This paper considers ways in which these scheduling algorithms can be enhanced and the resulting impacts. Whilst the results are illustrated through their application to remote laboratory access, the implications are equally applicable to scheduling of access to any constrained resource.

Keywords: Scheduling; Resource Availability; Access; Remote Laboratories; Online Access

Introduction

Laboratories, like many other educational resources, are important tools for supporting student learning whose use is often constrained by financial or logistical constraints. These resources represent a significant financial and logistical investment that can be difficult to develop and maintain, particularly where they involve either significant physical infrastructure or limited software licences (Hofstein & Lunetta, 2004). The resultant limitation in resourcing levels is often dealt with by scheduling student access across an extended time period. This scheduling requires a set of strategies for managing the access and, ideally, optimizing the level of usage.

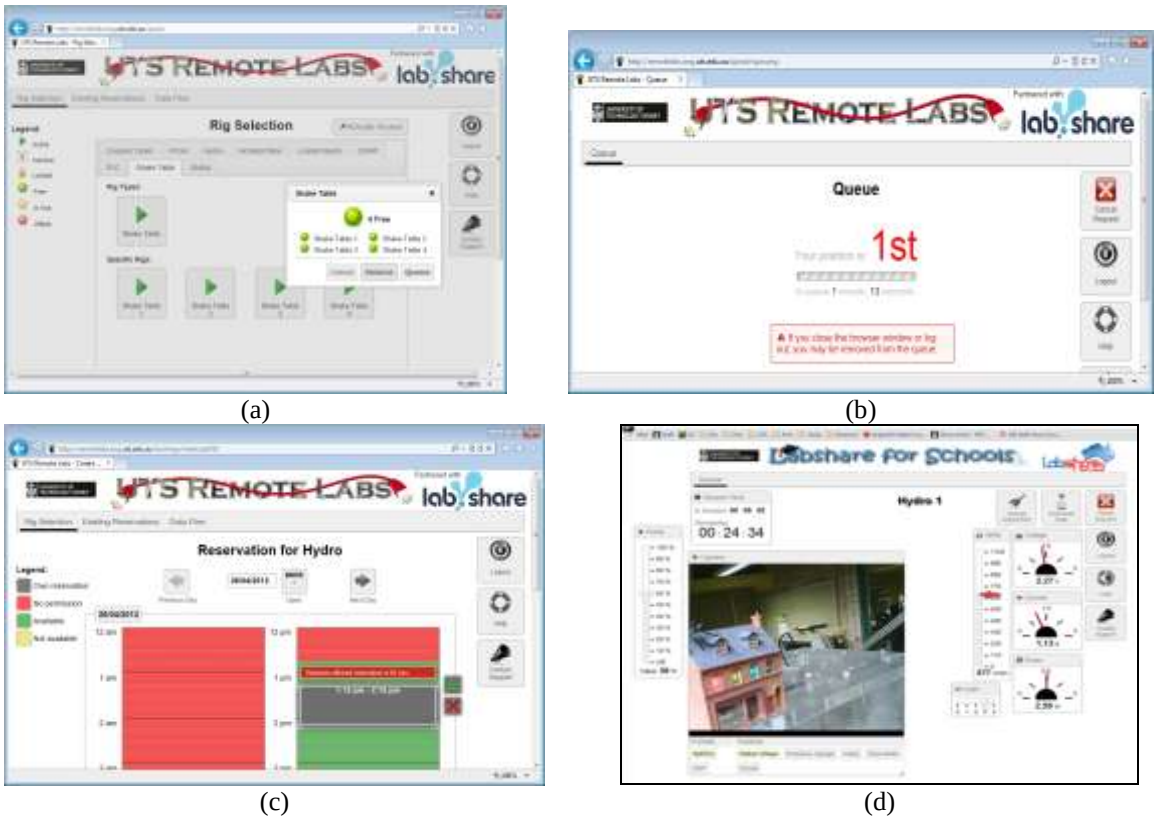
The above issues have been well studied within the context of remote laboratory systems. The rapid evolution and ubiquity of computer and networking technologies and the increasing sophistication of sensors and actuators have led to the emergence of remotely accessed laboratories (or Remote Labs - RLs) as a useful educational tool. Students are able to access, monitor and control physical laboratory experiments across the internet. Various benefits have been shown to arise from their use, including flexibility of access (Gomes & Bogosyan, 2009), the ability to share resources and labs across multiple institutions (Harward et al., 2008; Lowe et al., 2012; Richter, Böhringer, & Jeschke, 2009), security of users, data, and devices (Gravier, Fayolle, Bayard, Ates, & Lardon, 2008), amongst many other benefits.

With this shared distributed use of remote laboratories, scheduling of access has been a significant issue. A number of different strategies have been extensively explored. The two dominant access paradigms mirror those that have been used in many other systems: booking and queuing. In a booking strategy, users are able to select a time slot when they would like to access the system and make a reservation. They return at the specified time and are given access to the system. This is the paradigm that has been adopted by systems such as iLabs (Harward et al., 2008) amongst many others. There are also variations to this technique, such as the scheme used in NetLab (Machotka, Nedic, & Nafalski, 2009) where multiple students can make a concurrent reservation and then share access in a collaborative experiment. In a queuing strategy, such as that supported by Sahara (Labshare, 2010), users request access and are then placed in a queue. They are then provided with access to the first available apparatus that meets their specific request.

Good overviews of these strategies, along with various extensions and hybrids, can be found in the literature (Lowe & Orou, 2012; Orduña, 2011). Typically a booking scheme will be more commonly adopted where the available resources are very limited and student (or teacher, where the lab is being used for demonstration purposes) demand is relatively high – and hence having a confirmed access time is desirable. Conversely, queuing schemes are more commonly used where there is a significant pool of apparatus and students are happy to be allocated any item from the pool. Both of these techniques have benefits and disadvantages. For example, with booking schemes, typically the users will book a session of a specified duration but will very commonly not make use of the whole session. The unused component is then left un-utilised with a resulting waste of capacity. A queuing approach avoids this problem, but does not give users certainty regarding when they may be able to have access.

More recently, the Sahara remote laboratory system has incorporated a scheduling system that allows both queuing and booking to be used in parallel. Figure 1 shows a series of screenshots from the UTS Remote

Laboratory facility illustrating this functionality. The system administrator is able to specify which users can access which scheduling functionality and how different sets of apparatus are handled. For example, with a pool of 6 identical rigs, 4 may be able to be booked or queued, with the other two only available for queued users.



(a) (b)

(c) (d)

Figure 1. Screenshots from The UTS Remote Laboratory facility: (a) Selection of an experimental apparatus and provision of the option to either queue for the next available apparatus, or to make a booking for guaranteed access at a specified time; (c) A user waiting in the queue for access; (c) A user selecting a time slot to make a booking; (d) Accessing a specific experiment that allows students to explore hydro-electric power generation.

Previous work (Lowe & Orou, 2012) has shown that there can be complex interdependencies when both scheduling and queuing are used together. Whilst superficially it may appear that the performance (and hence access for students) would be improved by allowing both scheduling and queuing, analyses of live usage data has shown that significant problems can arise. In particular it was found in this previous work that there would often be significant periods of time where there was a queue of users waiting for access, but apparatus that was not currently in use was not being allocated because there was a later booking and the time available prior to that booking was not sufficient to guarantee the queued users the full session duration to which they were entitled (even if those users would typically not use the full duration).

In this paper we explore the implications of this issue and show how careful adaptation of the scheduling algorithms can lead to significant improvements in the overall resource availability. Whilst the results are discussed in the context of remote laboratories, the findings are generally applicable to the management of access to any restricted resources.

Scheduling Performance Evaluation

To demonstrate the implications of the interplay between booking and queuing systems we developed a Matlab scheduling simulation that emulates the scheduling strategy that is used within the Sahara remote laboratories system. An example of the resultant analysis and visualization is shown in Figure 2. The simulation was validated by testing it with a set of live data drawn from actual usage (277 bookings and 798 queue requests over a period of approximately 7 weeks) and confirming that the simulation allocated the same usage sessions.

To illustrate the implications of the interplay between different scheduling strategies we constructed a sequence

of scenarios based around a pool of 3 items of identical experimental apparatus being used by a total of 200 students across a 2 day period (note that the total availability in this period is therefore 3 rigs x 24 hours x 2 days = 144 hours of use). Each of the 200 students will request access to any one of the three rigs, and then when allocated make use of the rig for a random amount of time defined by $T.k^n$, where T is the maximum allowable period, k is a random value between 0 and 1, and n is skew factor that adjusts the likely usage duration. For this evaluation T=1 hour and n=0.5, which gives an average usage time of 0.67 hours, and hence a total usage time of approximately 133 hours –within the available usage time of 144 hours.

Scenario 1 involved all 200 students accessing the apparatus by queuing for access, and making the access request at a random point in the 2 day period. Figure 2 shows the resultant behaviour for one of the 3 rigs. As can be seen there are periods when a randomly greater number of students request access and hence the queue length increases, and there are periods when fewer students request access and hence the queue diminishes. Over the 48 hour period the average queue length is 0.84, there are only a few short period when the queue exceeds 5 users, the average waiting time is 0.61 hours, and the maximum waiting time for any student is 1.76 hours.

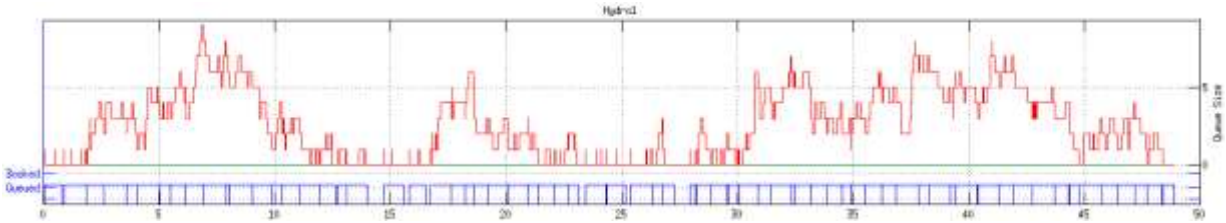


Figure 2. Example of scheduling analysis (Scenario 1) for a hypothetical remote laboratory. The top curve shows the queue length over time (the time axis is in hours) and the bottom curve shows the status of the rig – specifically whether it is idle, in use by a user who queued for access, or in use by a user with a reservation (in this case there are no users who made reservations).

In Scenario 1 no student reserved a specific timeslot, and so no student had a guaranteed access time. To illustrate the consequence of allowing students to make a reservation we considered Scenario 2, where half the students (100) accessed the rigs by making a reservation rather than queuing for access. A reservation will always be for the full time allowed (1 hour) but the users with reservations will still only use the rig for a variable amount of time and then release it back into the pool – potentially to be allocated immediately to a student waiting in the queue. This means that the average usage time (and hence the total usage) remained the same as in Scenario 1. The result of this change is shown in Figure 3. The behaviour in this case is starkly different. Even though the overall requested usage time has remained the same, the existence of the reservations has meant that there are significant periods when rigs are idle, but users are not allocated from the queue because there is less than an hour until the next reservation (because both queued and booked users are allowed to use up to an hour, the system will not allocate them to a rig if there is a reservation within the next hour). The result is an average queue length of 12.55, an average wait of 21.7 hours, and a maximum wait of 31.5 hours. Whilst there are active reservations the queue length continues to grow and does not diminish until after the reservations are all completed. Most significantly, over the first 48 hours, each of the rigs was in use for an average of 29.65 hours, and hence an average of 18.35 hours (38%) of the time was idle despite the existence of queued users.

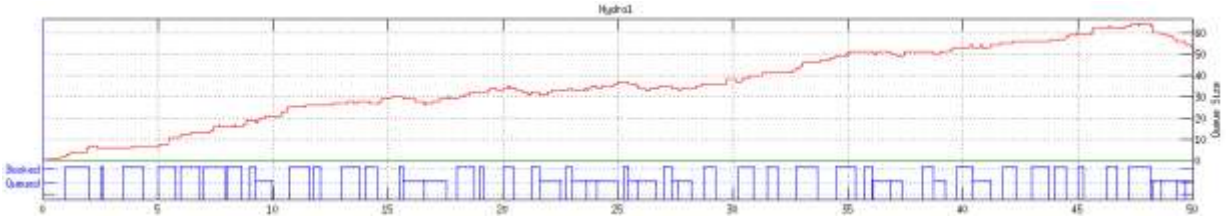


Figure 3. Scheduling analysis for scenario 2. In this case half of the 200 students make a reservation prior to access, with the result that there are substantial unused periods (in between reserved timeslots) and hence a queue that continues to lengthen throughout the period of use.

The third scenario shows how this issue can be addressed by a change in the allocation strategy. The booking and queuing requests were kept the same as for scenario 2, but the allocation algorithm was changed to incorporate two modifications. The first change was to allow a reservation to be delayed by up to a fixed amount

(in much the same way that if you have an appointment with a doctor for 3pm, you may be admitted to see the doctor at some time after the reservation time). In this scenario, the commencement of the reservation could be deferred by anything up to 12 minutes. This allows queued users to be more readily inserted into gaps between reservations. The second modification was applied when there were short periods prior to the commencement of a reservation. In this case the shorter period was offered to each queued user in sequence. In the simulation a queued user would randomly accept only if the offered period was longer than the duration of their usage (with the likelihood of acceptance increasing at the offered session got longer). The result of these changes is shown in Figure 4. As can be seen by comparing this to Figure 3, the impacts on the user experience are significant. The average queue length is 1.33 and the average wait time is 1.80 hours. This is longer than the average wait time of 0.61 hours when no reservation at all are allowed and derives primarily from users having to wait for reservations to complete, but is still relatively short. Most significantly, the overall usage level remained almost identical to the case with no reservations and it was possible to make maximum usage of the resources.

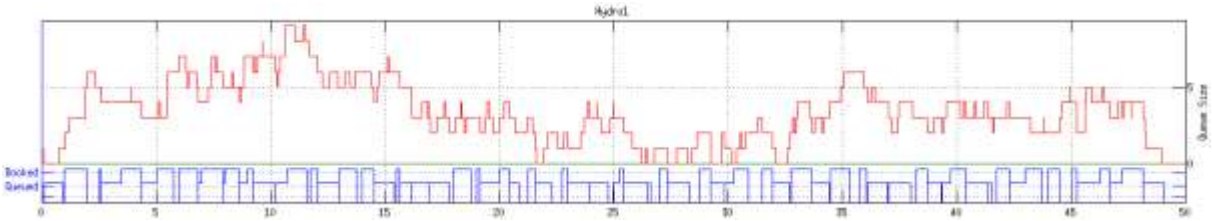


Figure 4. Scheduling analysis for scenario 3, incorporating changes to the allocation algorithm.

Conclusions

This paper has discussed the implication of merging booking and queuing strategies for managing access to laboratory resources. It was shown that if care is not taken a highly suboptimal allocation can result that leads to a significant reduction in capacity and major impacts on the user experience. Conversely, relatively simple adaptations to the allocation strategies can ameliorate these problems and allow the benefits of both queuing and booking to be leveraged.

References

Gomes, L., & Bogosyan, S. (2009). Current Trends in Remote Laboratories. *Industrial Electronics, IEEE Transactions on*, *56*(12), 4744-4756. IEEE.

Gravier, C., Fayolle, J., Bayard, B., Ates, M., & Lardon, J. (2008). State of the art about remote laboratories paradigms-foundations of ongoing mutations. *iJOE*, *4*(1), 19.

Harward, V. J., del Alamo, J. A., Lerman, S. R., Bailey, P. H., Carpenter, J., DeLong, K., Felknor, C., et al. (2008). The iLab shared architecture: A Web Services infrastructure to build communities of Internet accessible laboratories. *Proceedings of the IEEE*, *96*(6), 931.

Hofstein, A., & Lunetta, V. (2004). The laboratory in science education: Foundations for the twenty-first century. *Science Education*, *88*(1), 28-54.

Labshare. (2010). Sahara Labs. Retrieved June 3, 2010, from http://sourceforge.net/projects/labshare-sahara/

Lowe, D., Conlon, S., Murray, S., Weber, L., Villefromoy, M. D. L., Lindsay, E., Nafalski, A., et al. (2012). LabShare: Towards Cross-Institutional Laboratory Sharing. In A. Azad, M. Auer, & J. Harward (Eds.), *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines* (1st ed., pp. 453-467). Hershey, PA, USA: IGI Global.

Lowe, D., & Orou, N. (2012). Interdependence of Booking and Queuing in Remote Laboratory Scheduling. *REV2012 - Remote Engineering & Virtual Instrumentation* (p. In Press). Bilbao, Spain: IAOE.

Machotka, J., Nedic, A., & Nafalski, A. (2009). Collaborative Remote Laboratories as an Educational Tool to Enhance Student Centred Learning in Engineering and Science Education in the 21st Century. In B. Goclowska & Z. Lojewski (Eds.), *Computer Science Applied in Education* (pp. 96-121). Maria Curie-Slodowska University Press, Lublin, Poland.

Orduña, P. (2011). Scheduling schemes among Internet Laboratories ecosystems. *REV 2011: 8th International Conference on Remote Engineering and Virtual Instrumentation* (pp. 1-6). Brasov, Romania.

Richter, T., Böhringer, D., & Jeschke, S. (2009). LiLa: A European Project on Networked Experiments. *REV 2009: 6th International Conference on Remote Engineering and Virtual Instrumentation*. Bridgeport, USA: IAOE.

Author contact details:

Prof David Lowe, david.lowe@sydney.edu.au

Please cite as: Lowe, D. (2012) Impacts of Scheduling Algorithms on Resource Availability, Ascilite 2012, Wellington, NZ. (pp. 575- 579).