

Active Learning in Undergraduate Computer Systems: Before and After the COVID-19 Pandemic

Katrina LaCurts Manya Ghobadi Karen Sollins
Massachusetts Institute of Technology

Abstract

In this paper, we evaluate and reflect on teaching MIT’s upper-level undergraduate computer systems course, **Computer Systems Engineering**, or 6.033. This course is based on the textbook **Principles of Computer Systems** by Saltzer and Kaashoek [21] and a collection of research papers from the literature. The course, dating back to the 1960s, is a complex subject both because of size (about 400 students and over 30 staff members) and complexity of elements, bringing the students through increasingly complex systems: from single processor structures, to the distribution engendered in networking, to distributed functionality and coordination, to the complexities arising from security violations. In addition, 6.033 also meets an MIT *communications requirement*. In recent years, the key pedagogical element of the class has been *active learning*, which involves a great deal of interaction with each student.

On March 15, 2020, six weeks after the start of the semester and one week before Spring break, MIT shut down and gave the staff two weeks to go completely online. Our goal became to retain as much of the course as possible with students scattered literally around the world with varying abilities to join in real-time. This paper reviews many of the key decisions about how we did this and why we made the choices we did. In this paper, we reflect on the success of the residential version of 6.033, as well as opportunities to improve the online version of the course, especially including recovering some of the active-learning opportunities.

1 Introduction to 6.033

Computer Systems Engineering has been in existence in a similar, but evolving form since 1967. It was initially taught experimentally by Profs. Fernando Corbato and Jerry Saltzer. In the next year, it became a standard course in the MIT curriculum taught initially by Profs. Jerry Saltzer and Robert Graham with the course number it still retains: “6.033” [1]. It brings the students through a series of increasingly complex systems, beginning with issues in computing on a single computer, including virtualization and processes [20], to the challenges that arise with the introduction of networks, both in terms of the implications for the computers themselves [14] and the complexity of designing, building, and running a network at the scale of the Internet [6, 16]. After that, the course moves on to issues of distributed functionality [8, 10] and coordination [11], and concludes with the complexities that arise from security violations [13]. The trajectory of the course presents the students with increasing complexity in computer systems and approaches to addressing them. As one might imagine, the actual content of the course has evolved over the decades, as computer systems, networking, and security have all evolved. At this point, each of the four major topics is discussed over a 3-4 week period.

1.1 Format of the Course

The format of the course has remained fairly unchanged, and involves a handful of moving parts:

- Two one-hour **lectures** each week, paired with a reading assignment from the text. Every student in the class—350-400 students in recent years—attends the same lecture section. Lectures serve primarily as a one-way communication channel—though there is often some back-and-forth—to explain the key challenges, concepts and complexity of systems.

- Two one-hour **recitations** each week, in which students discuss, analyze, and critique papers from the literature that complement the topics of that week’s lectures. In recent years, we’ve had sixteen recitation sections, with 22-25 students each. Students are required to read the assigned paper and answer a few conceptual questions before each recitation. They are also graded on their participation in recitation, which requires them to engage in the discussion, not just to be present.
- Weekly **communication tutorials**. MIT requires each student to take two general communications courses and two communications courses specific to the student’s major; 6.033 is one of the latter for EECS students. The weekly tutorials are led by communications instructors, run in smaller sections like the recitations, and teach students to analyze and critique a technical paper. These skills are borne out most readily in the design project, discussed below.

Though 6.033 includes a midterm and a final, as well as a variety of “hands-on” exercises (e.g., having students explore systems using `dig`, a tool for requesting information from the Domain Name System, and `traceroute`, a tool for exploring the path of traffic through the network), the primary assignment in the course is a semester-long **design project**. This design project is done in teams of three students, all of whom have the same tutorial and recitation instructor, and asks students to design a large-scale distributed system over the course of the semester; for example, in Spring 2020, teams designed a network storage system for an extraterrestrial network. Teams write a preliminary design report (due before Spring break), give an oral presentation (due a few weeks after Spring break), and write a final report as well as a peer review of another team’s design. In addition to handling the technical material, students must also develop the skills to work effectively in teams. Both the recitation and communications instructors help them with this management and technical challenge. Note that the design project does *not* involve an implementation, but requires substantial design work, justification for their design, and preliminary evaluations.

1.2 Staffing and Infrastructure

To handle all of the moving parts, 6.033 has a large staff. The head lecturer leads the course, delivering the lectures, managing staff, determining what technical material to cover, etc. In addition, there are 8-9 teaching teams consisting of a recitation instructor (generally a faculty-level PI), a TA, and a communications instructor. Each team is responsible for two sections’ worth of students. One of the recitation instructors often takes the lead on managing the design project, and two of the communication instructors coordinate the communications curriculum. There are also 1-2 administrative TAs, who handle a variety of logistics, and a group of undergraduate graders, who grade the hands-on assignments.

To bring new people into the teaching staff and keep the whole operation moving smoothly, there are two weekly staff meetings: one for the technical staff plus the two communications leads and the other for the communications staff. These meetings are opportunities to share questions and ideas for the coming week’s material—including objectives and goals, teaching challenges, and ideas for how to approach the material—and are an important component of the success of 6.033 (see Section 2.3).

The course also has a great deal of online preparation, including guidance for both students and staff. We utilize the online discussion tool Piazza [3] to raise and address many sorts of detailed questions, including questions about lectures, the reading, the homework assignments, the design project and quizzes. Significant effort goes into making sure that all questions and issues are answered by the staff. We note this here, because it meant that much of the infrastructure required to support the teaching staff in online teaching was already in place.

1.3 The Disruption

On March 15, 2020, MIT closed and sent all students home; we refer to this as “the disruption”. The challenge that this brought to us was to take a complex course of almost 400 students and a staff of over 30, which was primarily based on a great deal face-to-face interaction and active learning, and translate that to remote learning. The size of the course meant that we had students in multiple timezones, with differing levels of Internet access, and different home circumstances (e.g., access to quiet space). Our primary objective was to maintain the interactivity and engagement of the course while supporting students’ different needs. In this paper, we will focus largely on our attempts to translate active-learning exercises to this environment.

2 Active Learning in 6.033

Before discussing how we translated our active-learning exercises to remote learning, it is important to understand how this approach is used in the residential version of 6.033, and why we have chosen it. Active learning has been the main pedagogical approach for 6.033 in the past four years. Active learning is a form of learning that attempts to involve students more directly in the learning process; rather than passively listening to a lecture, students are active participants.

There is much evidence in favor of active learning: it can improve a variety of learning outcomes such as exam scores and study habits [19], can confer disproportionate benefits from STEM students from disadvantaged backgrounds [12], and creates a more inclusive learning environment [15].

2.1 Motivation for Active Learning in 6.033

In addition to all of these benefits, we turned to active learning in 6.033 to solve a specific problem in our recitations. The goal of our recitations has always been to have students discuss technical papers. This is a class about how systems are *designed*, not just about how systems work. One of the ways that we teach that is by having students analyze and critique existing systems, asking question such as: What were the goals of the designers? Does the system meet those goals? Are those goals important? How might we re-design this system today? What trade-offs did the designers make?

A large part of that instruction happens in recitation, and students apply those skills to their own design project. Consequently, it is important that students are engaged during recitation: answering and asking questions and coming prepared. Prior to introducing active learning, we were not meeting these goals. We found that students were not willing to volunteer answers, and that many of them were not reading the papers beforehand. As a result, the recitation instructors spent a significant amount of time lecturing about the paper, creating a negative feedback loop: students knew they were going to get a lecture, so they would not bother reading the paper beforehand, and certainly would not come ready to discuss it. Though the majority of the technical content from the papers—how the system worked—was being delivered, we were not getting to teach the material we needed to focus on: how to read papers, and analyze and critique the systems described in them. Furthermore, it just was not a fun way to teach a systems course.

In cases when discussions *did* happen, many of our instructors encountered the common situation where a handful of students would dominate the discussion. Other students would tune out, or seem visibly uncomfortable volunteering their answers in front of their peers.

2.2 Examples of Active Learning in 6.033

To solve these problems, we turned to active learning. All of the activities we use help us foster an inclusive environment that encourages discussion and participation, allow us to analyze various aspects of a system's design, and convey important technical material. Here, we outline the most common four activities we use.

2.2.1 Group Work to Class-wide Discussion

In almost every recitation, we have the students think about and discuss a question in small groups, and then bring things back to a class-wide discussion. For example:

- In our recitation on content distribution networks, the students read the Akamai Network paper [17] which describes various goals of an Internet-scale content distribution network (such as performance, reliability, and scalability). We divide students into groups and have each group think about what aspects of the system support each of those goals.
- In one of our recitations focused on the Unix operating system, we focus specifically on naming in the Unix filesystem. We ask each group of student to list what “things” have names, and challenge groups to come up with as many things as possible.

In both of those examples, we bring the class back together for a discussion, and try to have each group contribute. Talking in the small groups beforehand makes the shyer students more confident, and asking each group to contribute at the end means the discussion does not get dominated by 2-3 people. It also

allows us the time and space to discuss more content than we could otherwise; for instance, *all* of the design goals of a CDN, rather than just one.

2.2.2 Debates

Debates in recitation have been exceptionally successful, both because the students enjoy them and because they raise many issues that can take a long time to raise in a discussion. As one example, the students read Dave Clark’s updated and annotated version of **The Design Philosophy of the DARPA Internet Protocols** [7]. For the corresponding recitation, we pose two debatable propositions:

1. Having a VERY narrow waist is a good idea
2. The network would benefit from more state in the network

We then divide the section of students into four teams—one “pro” and “con” team for each proposition—and give each team a few minutes to prepare their arguments. We then handle each proposition in turn: the pro team presents, then the con team, and then the two teams prepare and present counter-arguments. Students who are arguing for or against the second proposition will vote for the most convincing argument in the first (and vice versa). We end with some follow-up discussion as to why these arguments were (or were not) convincing.

This approach works well in many papers, because there’s almost always a “debateable” question. It also works well when students read two papers on the same topic. In the past we have read Jim Gettys’ paper on Bufferbloat [9], a particular, localized network traffic congestion problem, and Marc Allman’s [5] response. The students debated which author made the most convincing argument.

One of the things that makes this activity productive is that we assign students a position. There is no pressure for students pick the “best” position themselves, nor do we have to worry about a particular position only having a few students in favor of it. Typically, all teams come up with convincing arguments, further driving home the point that there are not easy solutions to these questions.

2.2.3 Diagramming

One of the skills students develop for their design project is drawing a system diagram. We reinforce this skill in recitation by having them draw diagrams of the systems described in our papers. To do this, we break the students into 4-6 small groups and ask them to draw a particular part of a system. The students must consider the interfaces between abstractions or layers as well as how their component operates. This activity works well when we are discussing systems that have a lot of hierarchy and/or layering, such as the Domain Name System (DNS) or the Unix filesystem.

Once students become comfortable with this activity, we see lots of creativity here. There are no “right” or “wrong” answers, only opportunities to discuss and possibly enhance what they say. It often allows us to point out that there is no “correct” way to abstract the details of a system; different levels of abstractions allow us to understand different things.

2.2.4 Acting Out Systems

Now we come to many students’ favorite approach: acting out systems. The best example of this is in our MapReduce [8] recitation (although Raft [18] gets high marks as well).¹ MapReduce is an approach to dividing-and-conquering the processing of extremely large amounts of data across many “worker” machines. Students enter the recitation knowing the basics of MapReduce: the role of the master and workers and the basic progress of a MapReduce job. However, they often have questions about failures and edge cases; e.g., what happens when a machine goes offline in the middle of a job?

To have them act out MapReduce, we start by assigning a few small groups of students to be the workers, and one group of students (or sometimes the instructor) to be the master. The “job” we run is usually counting the number of each type of candy in a large bag. Each worker group has a specific task to work on—a specific handful of candy to count—and the master runs the clock. We plan for various failure

¹We are indebted to Frans Kaashoek at MIT, and Xi Wang at the University of Washington, for inspiring our MapReduce activity.

scenarios, where the instructor shuts down a particular machine during a job. As that happens, we pause the class and ask them what should happen next: should the master reassign the job? If so, to which worker? Where is the intermediate data? How does the new worker access it?

This activity has proved to be a good way to make sure students understand the basics of the MapReduce system, and to reinforce how it handles edge cases. Moreover, it allows us to easily discuss alternative designs: e.g., what if all data was replicated via GFS, the Google File System [10]? What if none was? It is important to note that this is *both* a highly enjoyable activity and one where we are able to cover the necessary technical content as well as discuss the goals of the system.

2.2.5 Summary

We utilize a number of other active-learning exercises in 6.033, but these represent our four most common exercises. Some require a bit of planning; for instance, we've iterated on our MapReduce activity for years and have to purchase the requisite materials (candy) in advance. Others do not; we hold small-group discussions on the fly in almost every recitation.

These exercises have dramatically increased the participation and engagement in 6.033 recitations. We are able to cover all of the necessary technical content and encourage discussion of various design principles in each recitation. We have found that the vast majority of students come to recitation having read the paper, and we rarely have issues with a single student dominating a discussion.

2.3 Training Instructors for Active Learning

In order to execute this level of active learning in 6.033, we provide a great deal of support for all of our staff. Although the majority of our instructors are faculty-level PIs (e.g. faculty, senior research scientists, senior lecturers), not all are; we have had graduate students teach recitations, and we encourage all of our TAs (typically masters students or undergraduates) to teach two recitations during the semester. We've also had brand new faculty or research staff teach with us, with very little teaching background. We have been able to effectively train all of these instructors in large part because of the amount of dedication the entire staff puts into planning, in the following ways:

- We make it known to all staff that active learning is a core component of 6.033 right from the start. The head lecturer meets with new staff members individually to explain why we use active learning in 6.033 (evidence for its success, how it supports the learning objectives in 6.033, the improvements we've seen in 6.033 students since starting to teach this way).
- The head lecturer provides guidance for each recitation, both in terms of what content to focus on, and what active-learning exercises have worked well in the past. This means that staff who are new to active learning do not have to come up with their own exercises from scratch. We reinforce the idea that active learning exercises do not have to be complex, last a long time, nor involve props; simply asking students to turn to a partner and discuss a question, and then bringing them back to the full group for a discussion, is an effective active-learning exercise. While we encourage all of our instructors to use active learning from the first day of class, we also encourage new instructors to start with the simplest active-learning exercises.
- We dedicate time in each of our weekly staff meetings to discussing successes and failures of active learning and suggesting new activities. This discussion builds a lot of collective support for teaching; the more senior instructors help train newer instructors, and the new instructors bring fresh perspectives. Between this time in our staff meetings, and the provided guidance for each recitation, it's rare that any instructor has to come up with a brand new active-learning technique for a particular recitation; we all learn and benefit from each other. Moreover, creating a space where we discuss our successes and failures in teaching has made us all more willing to bring things up that aren't working, and to ask for help.
- To add more individual support, our head lecturer or a more senior instructor will often observe individual sections to offer feedback to instructors. In doing this, it's very rare that we find anything to criticize; most frequently, we find instructors trying out creative and effective techniques, and these

observations help disseminate those good ideas to the rest of the staff. New instructors get additional individualized support as needed.²

2.4 Cost of Active Learning

In our shift towards active learning, the highest cost came during the first semester we tried it. Instructors had to be convinced to teach this way and we had to plan the majority of activities from scratch. This level of planning required a fair amount of additional time and work from the head lecturer and the most senior staff members.

In subsequent semesters, the cost of active learning has been low, as there is less from-scratch planning. Teaching this way does require instructors to cede some control over their classroom in a way that a lecture doesn't. We make space for students to discuss, critique, and ask questions; occasionally they ask us questions that we can't answer. However, we believe strongly that this benefits the students, and thus the entire class.

3 Transforming from the Classroom to Online

As active learning has proved very successful in the in-class version of 6.033, one of our primary goals in the move to remote learning was maintaining the recitation environment. We were faced with the challenges of moving the entire course online in two weeks, but also with the challenge of many students being in different timezones, with varying levels of Internet access, and with varying levels of access to quiet space for learning.

3.1 Overall Structure of 6.033, Post-disruption

Luckily for us, many aspects of the course were already online. We already used Piazza [3] for class-wide Q&A (both public and private), our own homegrown document submission site, and Gradescope [2] for online grading. To make everything more manageable for the students and staff, we made a few schedule changes:

- Lectures were pre-recorded and posted online so that students could watch them at a time that was convenient for them. Pre-recording lectures also freed our head lecturer up during the week to focus on handling unexpected logistical challenges, as well as on teaching recitations.³
- We moved to discussing one technical paper per week, and divided the week into roughly two halves: in the first, they consumed content (lectures and papers); in the second half, they discussed it.
- We broke recitations and tutorials into smaller groups of 5-8 students each (in the case of tutorials, 2-3 design project teams), and scheduled these to accommodate the students, who were spread across many timezones. Despite the geographic spread of our students, we were able to hold the majority of these sections in normal working hours on the east coast of the US.

We also experimented with additional online tools: a Piazza forum for each instructor's students (not just the class of 400+ students), as well as Slack for a different "chat" experience, and Zoom for live discussions.

3.2 Successes and Challenges in Active Learning Online

After the disruption, we conducted six recitations: four on topics in distributed systems, two on topics in security. We focused primarily on keeping students engaged in discussion, and spent less time designing more complex active learning exercises. A number of things worked well for us:

²As an example, one of the most senior recitation instructors sometimes holds individual meetings with brand new instructors to work through the specifics of the papers and possible teaching approaches for the week. This is especially helpful when new instructors don't have a background in systems.

³In Spring 2020, our head lecturer was also teaching recitations; this is not the normal structure of the course.

- In the residential version of 6.033, we ask the students a set of relatively straightforward questions to think about (and respond to) before each recitation. After the disruption, we made these questions more substantial, and allowed them to form the basis for our recitation discussions. For example, in our recitation on the processor security attack Meltdown [13], students pondered three questions beforehand:

1. Footnote 10, along with the end of Section 1, reference responsible disclosure practices. What ethical responsibility do you think researchers in this area of computer security have? How should they disseminate knowledge about new attacks?
2. Meltdown was a big deal when it was discovered. Why do you think that was? Does this change your assumptions about the security of your devices? How big of a concern are attacks such as Meltdown (and Spectre, which is also mentioned in the paper) compared to other security issues?
3. This paper describes the details of one rather specific attack. What higher-level lessons about security did you take away from it?

In addition to asking them to give thought to these questions, we also asked each student to publically reflect on one of these questions on their instructor’s forum, or to respond to another student’s reflection. Because this task was asynchronous, students could complete it whenever was convenient for them, though ideally they did so before their recitation group met. Overall, students thought deeply about the high-level takeaways of the paper, and made connections to other areas of the class. The responses were quite good, and in many cases generated some back-and-forth between students on their recitation forums.

- Once students got used to using videochat, small-group discussions worked fairly well, especially coupled with the fact that students had questions to think about ahead of time. In addition to the reflection questions, we gave them examples of other questions we would discuss during recitation (one of the examples for Meltdown: “The paper (Section 6.4) mentions that ARM and AMD CPUs do not appear susceptible to Meltdown, and posits that it could be that the current implementation of Meltdown is too slow. Why does the speed of the Meltdown code matter here?”). Because students could think about these questions ahead of time, most of them had answers prepared. In fact, the discussions were active enough that we could have accommodated more students per session.
- Flexibility in scheduling made it possible for all students to participate. In fact, some students spoke up *more* in video discussions than in person. We believe strongly that accommodating different student circumstances was integral to helping them learn the technical material; after all, if students could not even attend recitation, they were not going to get as much out of the class.
- Having tutorials and recitations with smaller groups of students allowed us to check in on everyone. Tutorials provided us an opportunity to keep on top of how each design project team was faring (e.g., were they working well together and progressing on the project). Recitations allowed us to do a more general check-in, and occasionally opened up opportunities for us to support students in other ways (e.g., finding out that their Internet connection was slow and figuring out alternate ways for them to participate without video).

Despite these successes, some things did not work as well as we had hoped.

- Although students *eventually* got comfortable using videochat, it took a few weeks. This was unfortunate, as we only had six recitations left in the semester after the disruption. The majority of staff agreed that the recitations got better each week as students got more used to the environment.
- Because of staffing constraints, we had to decrease the time for each recitation. This meant there was not as much time for questions about technical content (*how* the system works, rather than *why*). We anticipated this problem, and encouraged students to ask such questions on Piazza before recitation so that instructors could answer them quickly. However, students were reluctant to do that, even though we allowed them to be anonymous to other students in asking their questions.

3.3 Impact of the Disruption on Students

Overall, the students engaged very well in debating issues on Piazza in preparation for the Zoom-based discussions, and the engagement in these discussions remained high (in many cases higher than when we were on campus). However, we were unable to deliver as in-depth an experience with the technical details. Although the student scores on the final exam were on par with previous semesters', there was a general sense from the staff that the students did not get all of the technical content from each paper. Additionally, because we dropped from two papers per week to one, students had a narrower perspective on each topic.

Engagement in other parts of the course remained high as well. By the time the students left campus, they had already established cooperation patterns for working on their design projects. They continued to communicate effectively, which was evidenced in the strong results we saw in the final design project reports. In fact, neither exam scores nor design-project scores decreased substantially compared to prior years, despite the fact that MIT moved to a pass/fail grading system for Spring 2020.

More important than their assignment scores is the fact that we were able to maintain a sense of community in the class, and provide support for students with difficult circumstances, despite the disruption. Every single student had face-to-face time with their instructors each week; no student got "lost" despite the fact that we no longer saw each other in person.

3.4 Impact of the Disruption on Staff

6.033's existing infrastructure made our transition to online learning fairly smooth for the staff. This infrastructure includes the fact that the majority of documentation and guidance for the class was already online, but also the fact that our weekly staff meetings have always been organized and productive, and include dedicated time to discuss teaching that week's material. We certainly had different challenges to discuss this semester than we usually do, but the space to figure out how to handle them already existed.

The amount of time recitation instructors spent on the class remained roughly the same as before the disruption. Even though instructors held more individual discussions each week (with small groups of students), this was offset by the fact that these discussions were shorter, and that we only discussed one paper per week. However, the demands on instructors changed. They had to work across several different types of media (Piazza, Zoom), and make an effort to engage students in multiple ways. Having so many discussions about the same paper was at times difficult; most instructors prepared 3-4 topics for these discussions in order to avoid repeating the same thing 6-8 times in a short period. Finally, although the majority of our recitations were still held during normal working hours, not all were; some instructors held recitations as early as 8:00am and as late as 8:00pm to accommodate different timezones.

Pre-recording lectures to post online required a significant up-front burden on our head lecturer, but freed her up to handle the many unexpected logistical issues that arose during the second half of the semester. In Spring 2020, the head lecturer was also serving as a recitation instructor (i.e., was delivering lectures as well as teaching two recitation sections), which was a strong motivation for pre-recording lectures. The TAs and graders had the same responsibilities as before the disruption, and were able to manage them all very well remotely.

4 Moving Forward and Possible Improvements

In thinking about improvements for the next time we teach remotely—which will likely be in Spring 2021, as 6.033 is only offered once per year—we have focused largely on making sure students are getting as much out of the technical papers as possible, while maintaining engagement in the course.

4.1 Structural Changes to the Course

One of the biggest surprises from this spring was the fact that our recitations worked well with larger groups of 10-12 students (we learned this on the occasions when a scheduling conflict meant we had to merge two groups for a week). Scheduling recitations with more students at once will also allow us to go back to having longer recitations, which will allow more time to tackle technical material.

Along with this, we have considered slightly more radical structural changes to recitations:

- Holding recitations twice a week, but discussing the same paper in each of the two meetings. The first meeting would be specifically to understand how the system works, the second would be for analysis and critique. Even though we are normally able to get through both of these aspects of a system during a single in-person recitation, we recognize that in a world of remote learning, it is more difficult for students to digest the technical content of a paper; perhaps because they do not have peers to study with, or quiet space to read. Having two sessions would give students more flexibility to digest a paper fully. Both sessions would involve active-learning, with some exercises tailored to understanding different parts of the system, and others tailored to critiquing it.
- Meeting twice a week, again to discuss the same paper each time, and asking students to lead their second meeting in small groups. This idea was inspired by the structure of the Swarthmore Honors Program [4], and gives students more responsibility for engagement. In the first weekly meeting, the recitation instructor would lead the recitation as normal. In the second, students would meet in small groups of 5-6, and one of the students from the group would lead the discussion. The lead students would prepare discussion questions ahead of time to share with the rest of the group, and would then lead the group discussion under the supervision of an instructor or TA. We anticipate that giving students this responsibility will encourage them to read the paper more closely and ask their own questions about it.
- Grouping students into recitation sections based on the timezones in which they reside. Doing so will enable the students to maintain interactions with their fellow classmates throughout the week, outside of the recitations, and will make it easier for us to match instructors and TAs with sections of students (i.e., staff who cannot teach in the evenings will not be assigned students who can only feasibly meet in the evenings).

One additional structural change that we will likely make is to go back to some version of live lectures. Pre-recorded lectures sufficed to convey the technical content, but lost all of the spontaneity and energy that are normally present in 6.033 lectures. To accommodate timezones, we anticipate doing two sessions of each lecture (one at the MIT-specified time, and another to accommodate students who can't make the first time), and posting recordings online for students who need to watch asynchronously.

4.2 Specific Active-Learning Exercises

Many of our standard active-learning activities did not immediately translate to remote learning, and with the short amount of time to prepare for the move to remote learning, we were not able to accommodate exercises such as acting out MapReduce. However, four months later, we have many ideas:

- We made little use of breakout rooms in Zoom during our recitation sections. We tried them occasionally at first, but since students were still getting used to videochatting, the rooms did not prove productive. However, now that students are used to an online environment, we believe they would work well for facilitating discussions with small groups of students, and then coming back together for a class-wide discussion. These breakout rooms can expand the students' team work beyond their normal in-class grouping experience, which is often based on the fact that they often sit next to their close friends.
- We also believe breakout rooms would be useful for putting students into smaller groups and asking them to draw system diagrams. Although it is too early to claim one technology as a winner, we have found that tablets are essential for drawing figures, especially to share in groups (no matter what drawing software we tried, drawing with a mouse or a trackpad was painful). In the absence of good software for making truly collaborative drawings, we plan to experiment with a modification of this active-learning exercise: having one student be in charge of drawing a figure, while 2-3 other students describe to them what to draw.
- As part of our pre-recitation questions, we occasionally asked students how *they* would design an activity to act out that week's system if we were still on campus. This question always got enthusiastic responses, so we imagine extending it to ask them how they would design an activity to act out a system

remotely and then to execute those activities. There are a few technical challenges to overcome—how to assign labels to students with different roles in the system—but again, we believe that encouraging student engagement in leading/designing the recitation will be beneficial.

4.3 Supplemental Resources

In addition to the larger changes described above, we believe that providing students with supplemental resources for recitations could help their understanding of the material. For instance, towards the end of the semester, we began to provide them with a list of questions they should be able to answer about the paper after they read it; we anticipate doing that immediately at the start of our next remote semester. We also occasionally provided supplemental notes on the papers. These notes were well received, and did not appear to discourage the students from also reading the papers.

5 Final observations and thoughts

Active learning has been remarkably successful in the residential version of 6.033. Students engage with the recitation content much more than they did prior to active learning. They are able to master the technical content while engaging in analysis and critique of the systems, and also while having fun. We have always been pleasantly surprised by just *how much* they enjoy some of these activities (one student’s review of our MapReduce recitation: “Why can’t every recitation be like this?”).

After the move to remote learning, engagement in 6.033 remained high. Attendance and participation in recitation was on par with the residential version of the class, which we see as a major success. Even though some active-learning activities did not translate well, and some students were frustrated with the shorter recitation sections, the majority of students appreciated the staff’s dedication to accommodating different circumstances, and to making sure each student continued to get face-time with instructors.

Most importantly, we believe that accommodating these different circumstances is crucial to delivering technical content in a meaningful way. More than just helping students get face-to-face time with instructors despite their timezone, our use of active learning helped us maintain a sense of community with our students even though they were scattered around the world. We worried at the start of the disruption that students would become disengaged in the course; for instance, would they put in minimal effort on their design project? Would they cease attending recitations? Neither of these things happened, even though MIT moved to a pass/fail grading system. This is at the forefront of our minds as we prepare for the possibility of a second remote offering of 6.033.

6 Acknowledgements

We would like to thank the entire staff of Spring 2020 semester, including: Atissa Banuazizi, Steve Bauer, Caroline Beimford, Adam Belay, Eric Bradford, Olivia Brode-Roger, Johnny Bui, Michael Cafarella, Amy Carleton, Keith Clavin, Sydney Gibson, Peter Griggs, Eric Jiang, Dave Larson, Alex Lynch, Kevin Marinelli, Michael Maune, Urmi Mustafi, Juergen Schoenstein, Howard Shrobe, Christabel Sitienei, Jessie Stickgold-Sarah, Rebecca Thorndike-Breeze, Michael Trice, Crystal Wang, Diana Wofk, and Yunyi Zhu.

We would also like to thank the many folks who have been instrumental for 6.033’s success over the years, including Frans Kaashoek, Jerry Saltzer, Hari Balakrishnan, Mark Day, Dina Katabi, Sam Madden, Robert Morris, Martin Rinard, Pete Szolovits, Nickolai Zeldovich, and many, many more.

Finally, we would like to thank Jerry Saltzer for both valuable feedback and historical details, and our reviewers for their thoughtful comments.

References

- [1] 6.033 Course Website. <http://web.mit.edu/6.033/www/index.shtml>.
- [2] Gradescope (web service). <https://www.gradescope.com/>.
- [3] Piazza (web service). <https://piazza.com/>.
- [4] Swarthmore’s Honors Program. <https://www.swarthmore.edu/honors-program>.

- [5] ALLMAN, M. Comments on Bufferbloat. *SIGCOMM Comput. Commun. Rev.* 43, 1 (Jan. 2012), 30–37.
- [6] CLARK, D. The Design Philosophy of the DARPA Internet Protocols. *Symposium Proceedings on Communications Architectures and Protocols* (1988), 106–114.
- [7] CLARK, D. Annotated Version of The Design Philosophy of the DARPA Internet Protocols, March 2013. <http://web.mit.edu/6.033/www/papers/darpa.pdf>.
- [8] DEAN, J., AND GHEMAWAT, S. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM* 51, 1 (Jan. 2008), 107–113.
- [9] GETTYS, J., AND NICHOLS, K. Bufferbloat: Dark Buffers in the Internet. *Commun. ACM* 55, 1 (Jan. 2012), 57–65.
- [10] GHEMAWAT, S., GOBIOFF, H., AND LEUNG, S.-T. The Google File System. *SIGOPS Oper. Syst. Rev.* 37, 5 (Oct. 2003), 29–43.
- [11] GREENBERG, A., HJALMTYSSON, G., MALTZ, D. A., MYERS, A., REXFORD, J., XIE, G., YAN, H., ZHAN, J., AND ZHANG, H. A Clean Slate 4D Approach to Network Control and Management. *SIGCOMM Comput. Commun. Rev.* 35, 5 (Oct. 2005), 41–54.
- [12] HAAK, D. C., HILLERISLAMBERS, J., PITRE, E., AND FREEMAN, S. Increased Structure and Active Learning Reduce the Achievement Gap in Introductory Biology. *Science* 332, 6034 (2011), 1213–1216.
- [13] LIPP, M., SCHWARZ, M., GRUSS, D., PRESCHER, T., HAAS, W., FOGH, A., HORN, J., MANGARD, S., KOCHER, P., GENKIN, D., YAROM, Y., AND HAMBURG, M. Meltdown: Reading Kernel Memory from User Space. *27th USENIX Security Symposium (USENIX Security 18)* (Aug. 2018), 973–990.
- [14] METCALFE, R. M., AND BOGGS, D. R. Ethernet: Distributed Packet Switching for Local Computer Networks. *Commun. ACM* 19, 7 (July 1976), 395–404.
- [15] MILLER, S., PFUND, C., PRIBBENOW, C., AND HANDELSMAN, J. Scientific Teaching in Practice. *Science (New York, N.Y.)* 322 (12 2008), 1329–30.
- [16] MOCKAPETRIS, P. V. RFC0882: Domain Names: Concepts and Facilities, 1983.
- [17] NYGREN, E., SITARAMAN, R. K., AND SUN, J. The Akamai Network: A Platform for High-Performance Internet Applications. *SIGOPS Oper. Syst. Rev.* 44, 3 (Aug. 2010), 2–19.
- [18] ONGARO, D., AND OUSTERHOUT, J. In Search of an Understandable Consensus Algorithm. *2014 USENIX Annual Technical Conference (USENIX ATC 14)* (June 2014), 305–319.
- [19] PRINCE, M. Does Active Learning Work? A Review of the Research. *J. Engr. Education*, 223–231.
- [20] RITCHIE, D. M., AND THOMPSON, K. The UNIX Time-Sharing System. *Proceedings of the Fourth ACM Symposium on Operating System Principles* (1973), 27.
- [21] SALTZER, J. H., AND KAASHOEK, M. F. *Principles of Computer System Design: An Introduction*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009.