Report on the Third International Workshop on Teaching Performance Analysis of Computer Systems 2024

Vittoria de Nitto Personé
Tor Vergata University of Rome
denitto@ing.uniroma2.it

Y.C. Tay
National University of Singapore
dcstayyc@nus.edu.sg

1. INTRODUCTION

Teaching is one of the most essential activities of academics, and leading knowledge and critical thinking is crucial for a healthy and productive society. However, the context is complex. The last two decades were characterised by an economic crisis that negatively impacted many educational systems, and industry interests are driving social transformation. Inevitably, institutes of higher learning are changing their role, perhaps unawares. Meanwhile, new generations of students are also changing their learning modes.

The workshop series on Teaching Performance Analysis of Computer Systems (TeaPACS) started in 2021. The need for an in-depth discussion on performance education naturally came fifty years from the birth of the discipline, as computing has evolved at a frenetic pace since then (see the 2021 "Message from the Chairs").

The workshop has a particular structure: it has invited speakers and two discussions open to the audience. This year's schedule was constrained by the lunch and coffee breaks, so the question-and-answer sessions were separated from the talks.

Ideally, for effective participation, TeaPACS should engage the attendees for a full day to listen to the talks and contribute to the discussions. However, TeaPACS is usually held together with all other exciting workshops. This inevitably reduces the audience size.

In this third edition, four speakers contributed their experiences and ideas. The central theme of the discussions was the relationship between performance education and data-driven approaches. The issue of the new generation and the difficulties in engaging their interest and attention is always an essential point of the discussions. Despite the small audience, the discussions were interactive and intensive.

This report briefly describes all contributions and the two discussions open to the audience.

2. TALKS

Dieter Fiems (Ghent University) started with the first invited talk. He started by pointing out that the current pressure on courses and research in performance modeling recalls similar pressure previously from simulations. Performance evaluation has now settled into a mix of modeling and simulation, and we may see a similar synergy between modeling and (machine) learning. Despite all efforts, student enrolment in performance-related courses keeps dropping. For example, a "Performance of Fun System" course cannot compete with a "Fun System" course for student interest. Part of the reason may be in the time scale: the gap between the description of a system and modeling its performance is large compared to that between rules and strategies in game theory. Quoting De Saint-Exupéry ("... perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away ..."), Dieter observes that just as we look for the simplest model that can explain the dynamics, we may need to drop some classical content from performance evaluation courses, and focus on what is minimally necessary for understanding how stochastic systems work.

In the second talk, Michela Meo (Politecnico di Torino) began with her vision on the purpose of education, and the objectives for STEM (Science, Technology, Engineering and Mathematics) in particular. For illustration, she described her master's course on "ICT for Smart Societies". Networking is central to smart home/transport/grid/etc., and her course uses it to illustrate the STEM approach with the observation-modeling-verification cycle, and exercise the students' critical and creative thinking. Some elementary queuing theory (M/M/m and Pollaczek-Khintchin formula) is used to draw conclusions on bandwidth splitting, traffic shaping, resource allocation, etc. Lab exercises help the students understand the complementary roles that mathematical modeling and discrete event simulation can play in understanding network behaviour. Student feedback shows that while they like the hands-on practice and problem solving, they also appreciate the theory. Their diverse background is a challenge, so the instructor needs to give them time to learn.

The next talk was by Ana-Lucia Varbanescu (University of Twente), who considers performance engineering to be the use of systematic and quantitative techniques to meet performance requirements. As with apprentice training, these techniques come from some toolbox: lectures introduce the tools, lab exercises are for students to test these tools and projects let them use the tools to solve problems. For data-intensive problems, analytical modeling (e.g. roofline, bottleneck analysis, queueing models) and machine learning (e.g. data collection, deep neural networks) are tools for, say, feasibility analysis. The students should use models to decide if some solution is feasible, use simulators and benchmarks to generate data for analysis, select methods for tuning the solution, and explain where the resulting performance is coming from. Treating it like a puzzle makes it appealing to students. She has learnt not to underestimate live demos (let's see what happens when we modify this code ...) as the suspense triggers students. The other lesson learnt was that having such a practical course is very intensive in terms of keeping up-to-date and providing sufficient tools, expertise, and feedback to the students. (Unfortunately, Ana could not finish her extended abstract before the camera-ready deadline.)

Cristina L. Abad (Escuela Superior Politécnica del Litoral in Guayaquil-Ecuador) gave the final invited talk. She has emphasised to her students that workloads cannot be an afterthought ("garbage in, garbage out"). A review of the computing curricula recommendations by ACM and IEEE shows that the brief mention of workloads in 2013 was removed in 2020, before appearing again in 2023. Even so, it is silent on workload modeling. There are examples of courses (and textbooks) on performance evaluation and monitoring that teaches this modeling, but these are usually electives that have small enrolments. Instead, one could teach it as part of systems courses — that would fit the curricula guidelines, support performance engineering in practice, and motivate enrolment in performance modeling courses. Cristina gave examples of how workload modeling is incorporated into systems courses with a mix of experimentation with diverse workloads, homework on simulating a networked system, critique of workloads in research publications, and a project that implements a system and evaluates its performance.

3. QUESTION & ANSWER SESSION FOR DIETER AND MICHELA

Lishan Yang (George Mason University) asked the first question: How do we attract good students to courses on queueing theory? For Michela, her course is compulsory. In Dieter's experience, this is not an issue for students in Mathematics and Statistics: attracting CS undergraduates is hard; the same is true for master's courses. Ana faces a similar problem: The Dutch curriculum requires just one course in Mathematics, for which the enrolment is about 5%. In general, students from Asia have a more substantial background.

Gianfranco Balbo (University of Torino) has taught in the US and Italy, and he saw shrinking enrolment in the electives — we have to justify the need for mathematics and theory to the students (e.g. consider simulations). Given the current curriculum structures, Cristina asked if other professors see the need for performance courses.

Ana asked: What mathematics do we want the students to know? Even for those with a good background in mathematics, the amount can be overwhelming. Michela finds that her engineering students do not have enough training on probability; perhaps we can put some necessary short courses online.

Ana further asked: What about textbooks, evaluations and examinations? Michela only uses exercises and lab work to test what the students have learned (e.g. from simulations).

Lishan followed up with her question: How about graduate courses (on performance)? Dieter noted that these courses usually have no credit (but it needs to be clarified if that is a plus or minus for attracting them).

4. QUESTION & ANSWER SESSION FOR ANA AND CRISTINA

Andrea Marin (Universitá Ca' Foscari Venezia) joined this session and pointed out that CS courses talk about performance qualitatively, e.g., caching policies without minding the correlation. Perhaps we could have a reference book with small results on performance; when first-come-first-serve works well, when processor sharing is better, etc. The material need not be mathematically tricky but can be integrated into several CS courses.

Cristina agreed that colleagues often need to get it (the impact of correlation). It would be good to have performance experts in systems put together chapters in a textbook.

Ana noted that we are inspired by lecture notes by some colleagues, but we could perhaps think bigger. We could accompany the notes with slides and tools. This could be a community effort, perhaps with help from PhD students. In contrast, we now look online for relevant material ad hoc, but there needs to be more correct information. She recalls a student contradicting her in class, because of some (bad) YouTube video that the student had seen.

Gianfranco commented that many instructors don't know the historical background to scheduling and queueing theory, etc., so cannot tip off the students when such theories are relevant at particular points in their courses (e.g. processor sharing theory and time slicing).

Vittoria de Nitto Personé (Tor Vergata University of Rome) asked Ana: what is the theoretical part of her course? Ana answered there is a 1-hour introduction on queueing theory, followed later by introducing other theories as needed. The course structure had the hardware, then the software, followed by performance engineering.

Relating to Gianfranco's point, Ana remarked that she was asked to teach a course on computer organisation to AI students, who have diverse backgrounds. She chose as textbook Bryant and O'Hallaron's "Computer Systems: A Programmer's Perspective". It does precisely what Gianfranco suggested: it talks about the impact of the decisions that one makes in a programming a system. It is a peculiar book that is excellent in introducing students to performance and a well-done way of teaching computer architecture to students who are not computer engineers. It weaved in performance in a way that you can't avoid at every level of the architecture.

Dieter asked: How can we keep students focused on performance? Ana has weekly project meetings to keep them in line, asking them to explain the performance (good or bad) constantly steering them. In Cristina's case, where they have a project on cloud computing, they must have a design of the architecture, and they have to make at least one improvement to what they've learnt (e.g. load balancing) and evaluate the change.

Y. C. Tay (National University of Singapore) would like to know how they make sure projects demonstrate what they want the students to learn. Ana does that through the project report. The students have a list of projects to choose from, and a rubric specifies when they have to deliver what. Sometimes, they make a bad choice and encounter difficulties, but that also makes them curious and interested. Some may skip assignments, but they usually do not drop the project because they have latitude in time management and choosing their solutions (in the end, many want to do their best). She gave another example from graph processing where the students realised the choice of workload determines the performance: Should they choose a generic solution, but modeling is a problem, or narrow down to a specific class of graphs? They learn they have to give up something to make progress. Cristina similarly requires a report and has a midterm meeting.

5. DISCUSSION1: PERFORMANCE EDUCATION IN A DATA-DRIVEN WORLD

Vittoria started the first discussion by asking: How can we integrate black box approaches to give the competencies needed to analyse system behaviour? How can we preserve the essential knowledge in performance modeling and evaluation? How do we bridge the gap between tractable theory and practical problems?

Dieter noted that, in the classical approach, we start with assumptions. This is different for the data-driven approach — it would need another can of theories. Suppose we want to analyse some "what if" situation in a data-driven world. This is impossible unless you create some synthetic data, which is problematic. He recalled an industry project where they had network traces and were synthetically generating new ones (playing them at different times, etc.), but what they were creating had nothing to do with real traffic. There was so much correlation everywhere that got lost in the fake traces, so they got completely different results. In contrast, performance modeling shines when it answers questions like "what if the data is different?"

Michela is sceptical about black boxes. For us designers and analysts, we must know what is behind the data, and use domain knowledge. She gave an example of failures in a distributed system, where trees may fall on power lines in a storm. Designing a solution for such a system will benefit from domain knowledge that can provide rules. The designer must know how to combine various modules to provide a solution. If an issue arises, you then know what to check.

Ana supports using black boxes as tools. In her course, students have a lot of data. They were free to use any model they wanted, but they had to figure out what happened and why it happened. She feels that essential knowledge is maintained with this data-driven approach. She can still decide what the students should know. Her course is very practical, and they start with real problems; it doesn't matter how the students do it, but they must explain what they do, so there is no gap problem.

Cristina agrees that in the systems community, they must be able to explain themselves. They like to use traces, and students must be able to explain: Why is this trace different from that trace? Are there any models for the difference?

Lishan, also from systems, likes to package the problems as identifying bottlenecks. Given data from a simulation, how do you make sure that the data makes sense, whether it is a master's student justifying to a boss, or a PhD student describing their research. She doesn't see a gap problem either since the theory is supporting whatever we do in practice. For a problem like cache simulation, the data may not be large. However, for something like the response time of an Amazon server, there may be a large amount of data. Whether the approach is data-driven may thus depend on the nature of the project.

Gianfranco doesn't believe in black boxes per se. He thinks we should make the student an educated user of black boxes. He draws an analogy with simulators — they can be beneficial tools, but there are many examples of misusing them. A simulation that runs for two days on a supercomputer represents just one point in a huge space. Whether the results make sense may require guidance from a model.

Tay asked for opinions on students' current rush to black boxes. Why should they learn queueing theory when they can feed numbers to a black box and get answers?

Ana gave examples of how students are indeed keen on black boxes. However, she also has students who want to be exposed to white-box modeling before such techniques disappear. This is similar to a phenomenon where some students transfer from CS to AI and then back to CS because they worry they are not getting enough CS training.

Michela supports having students use a combination of tools. They should be aware of possible solutions and trained to be critical of any tool.

An audience member gave an example of flying a plane: when things go wrong, the pilot must be well-trained to handle the situation. Similarly, using simulators or black boxes, some white box training is necessary to understand and assess their results.

Ana asked, "What type of data do we need?" The design space is often huge, and we never have the correct data. She feels it is an exciting field to explore. She is learning from her students, who figure out what data they need and what data to collect. There is a massive trial and error phase. We do not yet have any theoretical understanding of how to exploit these black boxes (recall Dieter's point on "can of theories").

Cristina wrapped up the discussion by revisiting the question of how we can teach performance modeling to students who are lacking in terms of mathematical background. She suggests the paper-reading approach, where the professor curates a selection of systems papers that have models, and students choose what papers to present. Those with solid backgrounds may pick the more complex models, and maybe even develop their research projects from there. Those with weaker backgrounds may pick papers that are more implementation-oriented, and learn the related model one-on-one from the professor. This way, students can pick the level of mathematics that they are comfortable with, and need not deal with examinations (that can be a turnoff).

6. DISCUSSION2: MOTIVATING STUDENTS

Vittoria again kicked off the discussion by posing some questions: How do we cater to a mix of students' backgrounds (engineering, business, etc.)? What tools (solvers? simulators? polling?) can we use to engage student interest and attention? What exercises (e.g. lab work on load balancing and content delivery networks) can we design for students who focus on functionality, not performance?

In Cristina's experience, little anecdotes get the students' attention. She takes on consulting jobs with the industry (despite the stress) partly to collect problems and stories to motivate the students. She also tells them about former students who used what they learned in their jobs.

Ana's course is for 1st-year master's students, so the former students are still around in their second year. She has these 2nd-year students repeat their project talks to the 1st-year students, who get to grill them. The other way of motivating them is live demonstrations (tools, coding, etc.) — anything live appears to get their attention.

Dieter commented that there is a risk of bugs! He sometimes plays animation, but they can be counter-productive if they are too long. Vittoria also uses animation, e.g. simulation.

Tay observed that the examples above are about motivating students who are already in the course. What about attracting students to a course?

Ana thinks packaging is essential. Students did not care when she talked about energy efficiency, but they suddenly became interested when she referred to it as energy waste. When teaching AI students, she struggles with having them appreciate system performance when their focus is on AI performance (accuracy).

Cristina used to run a seminar where the students were actively recruited. It was not for credit, so the students only got knowledge (and free food). They would read papers and watch video presentations, and she would explain the parts of the performance they didn't understand. When she had a pipeline of these students, she could get those who started working to talk to the current students about the performance work that they were doing.

Tay sees an existential problem: if we cannot motivate a significant number of students to take performance courses, then they get cut, and it may be tough to remount those courses. We have to keep on maintaining enrolment. Can we somehow involve the industry to help us motivate the students?

Andrea has three visiting lectures from the industry. Even though they may talk about benchmarking, students get to see what they learn (whether the benchmarks use an open or closed system, how the queueing forms a network, etc.).

Ana finds that the problems that companies are interested in may be too complicated, but any anecdote helps. Internships also help but tend to focus on functionality rather than performance. If Cristina knew a company had a performance problem and fixed it, she would bring it to class and ask them specifically to talk about that problem. Andrea hears from graduates about performance problems in their companies, he would invite them to describe their experience in class.

Tay then followed up with a question to Cristina about getting workloads from the industry. She has tried, with mixed results, but thinks we should keep trying. They can take too long to deliver (waiting for approvals, etc.), but their workloads are a potential gold mine for training students, publication, and harvesting citations.

Dieter sees the problem as not finding performance issues in a company, but finding the people who speak the same language in performance and are on the same wavelength. There is also a risk of industry visitors talking more about functionality and trying to recruit students.

Tay commented that Dieter's point recalls a point made by Mor Harchol-Balter at the first TeaPACS about engineers not having the same terminology, the right way of framing their performance problem, etc. — just agreeing on the language is a problem. The industry visitor may end up confusing our students.

The SIGMETRICS/Performance 2024 Conference has more industry participants than usual, so Tay asked Andrea (the General Chair): Why do they care, and how can we leverage their interest? Andrea came across a comment that SIGMETRICS is disconnected from the industry, so the organisers made a special effort to establish a connection. Their focus may be on simulations, but we can point out what their simulation is missing out (e.g. optimality, or where an analytical model can provide a view of where to go), help them do something better than what they are doing. In the future, SIGMETRICS will work on strengthening this industry connection.

Cristina thinks it is essential to teach performance modeling in performance analysis. Once you work out a model (an optimisation problem, or queueing model, or Markov chain), solving and analysing it is a separate problem that can go to a different person. Similarly, in the industry, it is fine if they are not performance experts, but they need to know what they need, and have the correct vocabulary to describe their needs and get the right person to help. She sees a need to educate the industry, help them see that they need to think beyond functional requirements, and think about performance before their system crashes because it cannot handle the load.

Ana sees a similar issue in the high-performance computing industry. They approach academia for help, and one solution is to give master's students to them as interns for performance engineering projects. With enough interaction, they now have a shared vocabulary for formulating and discussing performance problems. Going back to her talk, she feels the community needs to agree on what material (methods, tools, etc.) to push into existing courses (in operating systems, networking, etc.). Keeping this body of knowledge and repository up-to-date requires considerable effort.

Cristina thinks that it is harder for students nowadays to see a production system with a performance problem. How can we make it easier for them to understand this type of model or tool can be used to solve that type of problem (apart from reading many papers)? A toolbox or corpus and a list of techniques (like in high-performance computing) would help.

Ana recently came across a survey of the industry that tries to find out what is helpful for them and what they expect and want. Such an exercise will have an immediate impact on what we teach. If there is a missing connection between industry and SIGMETRICS, it may be because they think our students would not make suitable employees.

7. CONCLUSION

TeaPACS has so far collected the thoughts and contributions of 13 invited speakers and many participants from the Performance/Sigmetrics community (industry, academics, and students) in its first three editions. This workshop series aims to focus attention and raise awareness, rather than draw conclusions or formulate guidelines.

Moreover, the presentation slides, extended abstracts and workshop reports are collected and available at the TeaPACS repository (`https://teapacs.github.io/2024/`), so that the discussion on Performance Education can be shared with the community.