Refreshing ATC – USENIX ATC ’19 Program Co-Chairs Message

Dan Tsafrir
Technion – Israel Institute of Technology
and VMware Research

Dahlia Malkhi
VMware Research
and Calibra

1 Introduction

Welcome to ATC ’19: the 2019 USENIX Annual Technical Conference. The scope of ATC covers all practical aspects related to systems software, and its goal is to improve and further the knowledge of computing systems of all scales, from small embedded mobile devices to large data centers, while emphasizing implementations and experimental results.

The ATC ’19 program is the result of tremendous efforts by many in our community. We are most thankful to the authors who submitted their high-quality work and to the reviewers who undertook the challenging task of evaluating hundreds of submissions and providing constructive feedback to the authors. While working on creating the program, we have been repeatedly inspired by our reviewers’ competence, experience, patience, and dedication. Thanks to their efforts, we are happy to report that the excellent program of ATC ’19 achieves its aforementioned goal.

Briefly, we received 356 submissions and accepted 71 (19.9% acceptance rate) through a double-blind, two-rounds review process. The statistics that describe the submitted and accepted papers, along with the details of the review process, are summarized in Table 1 and are further discussed below.

This document is somewhat longer than is typical for a “message from the ATC program co-chairs”. What motivated us to write this detailed report is the many changes that have been introduced to ATC this year, the reasoning underlying them, and the new things we have learned while working on creating the program. The potential target audience for this document is future chairs, or readers who wish to learn more about the process.

2 Changes

We have introduced some notable changes to ATC this year, primarily to meet higher reviewing standards used by other major systems conferences. We discuss these changes next.

2.1 Increased Number of Reviews

Top-tier system conferences typically employ a two-rounds reviewing process in which each submission receives at least three reviews in the first review round (R1), and then, if the submission is promoted to the second round (R2) based on its R1 reviews, it gets at least two additional reviews, amounting to at least five reviews per R2 submission.

count	*description*
i. all submissions (short & full):	
356	submitted (458 registered)
29	violated format, given 24 hour to fix
2	rejected+withdrawn due to said format violations
2	withdrawn before review process ended
352	underwent the full review process
184	promoted to review round #2 (R2)
80	R2 submissions pre-rejected during online discussions
37	R2 submissions pre-accepted during online discussions
67	R2 submissions discussed at PC meeting (accepted 34)
71	accepted (19.9% acceptance ratio)
ii. short submissions:	
32	submitted
1	rejected+withdrawn due to format violations
8	promoted to R2
7	R2 submissions pre-rejected during online discussions
1	discussed at the meeting and accepted
1	full submission accepted as short
iii. committee & reviewing load:	
66	heavy weight PC members; 18–19 reviews per member
28	light weight PC members; 13 reviews per member
22	external review committee (ERC) members; 5 reviews
116	committee members
51	external reviewers; 1 review
iv. reviews:	
3–4	per submission in R1 (at least 2 by heavy members)
5–6	per submission in R2 (at least 4 by heavy members)
1,347	reviews in R1
405	reviews in R2
1,752	total, consisting of 1,097,815 words (6.7MB)
v. authors:	
1,695	all submissions (1,442 unique, with 409 affiliations)
384	accepted (361 unique, with 118 affiliations)

Table 1: *ATC ’19 submissions and reviewing statistics.*

In contrast, until this year, ATC R1 and R2 submissions received only two and four reviews, respectively. Upon investigation, we have learned that the decision to employ fewer reviews than other systems conferences has been made more than a decade ago, by the program co-chairs of ATC 2008.

We and many others believe that making review-round promotion decisions based on only two R1 reviews is less informed, and hence leads to higher variability in the result. We further feel that the minimal number of R2 reviews should be similar to that of the other main system conferences, to allow for a better, more rigorous paper selection process. Consequently, this year, all submissions have indeed received at least three R1 reviews and at least five R2 reviews.

2.2 Double Blindness

Ever since ATC has been established, and until this year, the conference has employed a single-blind reviewing process, whereby reviewers see the names of the authors of the submissions that they review. While simplifying the review process, studies show that single-blindness might lead to bias against minorities and in favor of well-known authors and organizations. For example, Tomkins et al. show that

> *"Reviewers in the single-blind condition typically bid for 22% fewer papers and preferentially bid for papers from top universities and companies. Once papers are allocated to reviewers, single-blind reviewers are significantly more likely than their double-blind counterparts to recommend for acceptance papers from famous authors, top universities, and top companies. The estimated odds multipliers are tangible, at 1.63, 1.58, and 2.10, respectively."* [14]

Similarly, Goues et al. show that

> *"Reviewers with author information were 1.76x more likely to recommend acceptance of papers from famous authors, and 1.67x more likely to recommend acceptance of papers from top institutions. [...] When reviewers knew author identities, review scores for papers with male-first authors were 19% higher, and for papers with female-first authors 4% lower."* [4]

The latter study also shows that reviewers are usually unable to deanonymize authors of submissions by guessing, even if they believe themselves to be experts on a submission's topic.

Accordingly, major systems conferences (including SOSP, OSDI, ASPLOS, Eurosys, FAST, NSDI, and USENIX Security) employ a double-blind reviewing process by keeping author identities concealed from reviewers.

For ATC '19, we employed this policy as well, and we hope future chairs will continue to do so. The ATC '19 call for papers (CFP) requires authors to make a good faith attempt to anonymize their submissions by avoiding identifying themselves or their institution, either explicitly or by implication, e.g., through references, acknowledgments, online repositories that are part of the submission, or direct interaction with committee members. When authors cite their own studies, the CFP specifies two possibilities: either cite them as written by a third party (preferable), or as anonymized supplemental material uploaded to the HotCRP submission management system (most useful when the cited work is currently under review or awaiting publication). Prior publication as a technical report or in an online repository does not constitute a violation of anonymity.

2.3 Author Responses

Most premier systems conferences – e.g., OSDI, SOSP, ASPLOS, USENIX Security – give authors a few days to write a response to the reviews. The authors' response is known as "rebuttal", and it is optional. It allows authors to provide answers to specific questions raised by reviewers and, importantly, to correct factual errors or misunderstanding in the reviews. (It may *not* provide new results or reformulate the presentation.) Some researchers perceive rebuttals as essential for the reviewing process, to keep it fair and transparent [6], and some ACM SIGs encourage program chairs and steering committees of SIG-sponsored events to employ rebuttals, based on feedback from their members [13].

Therefore, for ATC '19, we chose to allow authors to rebut. Similarly to our past experiences in forming programs while serving in committees of conferences that employ rebuttals, our sense is that the author responses have contributed to the ATC '19 process. Primarily because they allowed the reviewers to make better informed decisions in certain cases. But also because they implicitly encouraged reviewers to write more accountable reviews and, importantly, to submit them on time so as to be visible during the authors response period; the latter allowed the online discussion period to start on time with all the required material available.

We used a 500-words soft limit on the size of the rebuttal; reviewers were not required to read more. The reviews were made visible to authors in the rebuttal period, during which reviewers were asked to avoid modifying them. After the rebuttal period ended, reviews became invisible to authors again, allowing reviewers to update them based on the rebuttal, the online discussions, and the program committee (PC) meeting.

2.4 Submission Chairs

The ever-increasing number of submissions to systems conferences (approaching 400 in the last two ATCs) makes it increasingly challenging for everyone involved to create a program. For example, it is challenging for reviewers to bid on hundreds of submissions so as to express review preferences. It is likewise challenging to arrange things such that the submission system accurately reflects conflicts associated with more than a hundred reviewers and an order of magnitude more authors (experience repeatedly shows that many conflicts are missing because reviewers and authors neglect to declare all their conflicts). It is also challenging to manage a "dual track" PC meeting (where the PC is split between two rooms part of the time) in a manner that ensures that all committee members are found in the right room at the right time in order to discuss the submissions they have reviewed. Many other examples exist.

For this reason, we decided to formalize the role of a "submission chair" as part of the official organizers of ATC. The job of the submission chair is to help the program chair in accomplishing tasks such as those listed above by, for example: adding missing conflicts to HotCRP based on DBLP; helping reviewers' bidding by identifying the submissions that cite their papers and communicating this information to the reviewers; checking format violation in uploaded PDFs and communicating with authors to quickly fix those through reformatting and content deletion; helping to ensure that the quality of the reviews assignment is high (HotCRP assignments might be far from optimal); helping to make sure that per-submission administrative tasks are being carried out and progress is achieved, e.g., by following up on submissions that were not yet tagged as passing the "review sufficiency check"; helping in scheduling of the dual track meeting; and serving as scribes during the meeting while making sure the scheduling of PC members in rooms works as expected.

Submission chairs get admin privileges in the HotCRP system in order to carry out their duties. Their role, however, never requires them to make decisions that affect the outcome of the review process. For example, they do not steer online discussions. It is productive for the program chair and submission chair to be geographically located near each other, allowing them to physically meet when the need arises.

2.5 Extended Review Committee (ERC)

Most of the premier systems conferences, which must review a few hundreds of submissions, typically employ a light-heavy program committee model, where "light" PC members review fewer submissions but do not attend the PC meeting, whereas "heavy" members review more submissions and attend the meeting. This model is needed in order to decrease the high reviewing load of PC members, while keeping in mind that the number of people who can sit in one room and conduct a productive discussion is bounded.

Last year, unpredictably, ATC '18 received nearly a hundred additional submissions as compared to ATC '17 (377 submissions as compared to 283 submissions, respectively). To our knowledge, the PC of 2018 was the the first ATC PC to employ the light-heavy model. In previous years, all ATC PC members were "heavy", which was viable because the number of submissions was much lower, albeit, even so, past ATC-s reviewing load was sometimes in the range of 25–30 submissions per member. (Some of us were members of those PCs and still remember the pain.)

Our goal for this year was to ensure that the reviewing load of heavy members will not exceed 20 submissions. In parallel, USENIX instructed us to be prepared for an additional sizable increase in the number of submissions. Therefore, to be safe and have some flexibility, we decided to supplement the light-heavy model with an Extended Review Committee (ERC), consisting of members whose review load will be light: about 5 submissions per member.

Notably, due to the light reviewing load, ERC members were easy to draft regardless of their seniority: they typically accepted our invitation (which specified that the expected reviewing load will be 3–7). Additionally, more than a quarter of the ERC members were initially invited to serve as heavy or light members and opted for the lighter alternative instead of declining altogether.

Ultimately, having an ERC was a contributing factor that allowed us to assign four reviewers in R1 to most submissions (without increasing the load on light and heavy members beyond our planned upper bound). Having an initial assignment of four reviews proved to be invaluable when making R2 promotion decisions in the face of multiple late reviews, as three reviews were typically enough to confidently make the call. The ERC members additionally contributed by augmenting the expertise of our pool of reviewers.

2.6 No Abstract Submission Deadline

Last year, in their welcome message, the program co-chairs of ATC '18 stated that

> *"We required authors to submit abstracts a week before the paper submission in the hope of ensuring proper subject area coverage by the program committee and to get an idea of the reviewing load. This did not work. We had over 550 submitted abstracts, meaning almost 40% of the submissions were abandoned. In the end, requiring abstracts to be submitted early did not help with planning due to such a large number of abstracts that did not result in a submission"* [5].

To that we add that requiring committee members to indicate reviewing preferences before the submission deadline would be a waste of their valuable time, as they will inevitably bid on submissions that will not materialize. Stating review preferences given hundreds of finalized submissions is already time-consuming and challenging enough, and needlessly making this task even harder is counterproductive.

Bidding on registered abstracts that will not materialize into submissions would additionally negatively affect the quality of the review assignment, because committee members frequently stop bidding when they feel they have already placed "enough" bids on submissions.

Consequently, this year, we have to cancel the requirement to register abstracts in advance, and we eliminated the corresponding deadline.

2.7 Submission Deadline Closer to New Year

The date at which accept/reject notifications for ATC submissions are sent to authors is typically set by USENIX to around mid April.[1] Accordingly, since 2013, the submission deadline of ATC has been scheduled at the end of January or in early February, which thus far allowed the committee to complete the reviewing process in time to comply with a mid-April author notification date. This year, however, we set an earlier submission deadline: January 10, 2019.

Three issues necessitated this change. First, we needed additional time for the authors response period (Section 2.3) and for the "review sufficiency check" period that preceded it (described in Section 8). Second, as noted in Section 2.6, we had to allocate a few days following the deadline to allow reviewers to place bids on submissions indicating their review preferences; traditionally, such bidding took place before the submission deadline, as authors were required to register an abstract a week in advance.

The third issue that motivated an earlier deadline is the increased number of submissions. To cope with this increase, we allocated two weeks for online committee discussions scheduled before the PC meeting, in order to allow the committee to converge to a decision regarding as many submissions as possible—failing to do so would mean ending up with too many submissions to discuss at the meeting. The increased submission number also required allocating the week following the bidding period in order to assign reviews to members in a manner that would later allow us to reasonably conduct a dual track PC meeting (see details in Section 6).

Scheduling the submission deadline to occur soon after New Year may partially explain this year's somewhat smaller number of submissions as compared to last year: 377 vs. 356 in ATC '18 and ATC '19, respectively.

2.8 Uniform Shepherding

In the past, shepherding in ATC was not used by default. This approach reduces the load from both committee members and authors. A main drawback, however, is the increased likelihood that some of the issues that the reviewers expect authors to address in the camera-ready version remain unresolved.

The alternative approach, used by most of the premier systems conferences, is to assign shepherds to all accepted papers and thereby generally improve quality assurance. As part of our efforts to update the ATC reviewing process in order to make it aligned with that of its sibling conferences, this year, we decided that all accept decisions are conditional and depend on the approval of shepherds.

After the (conditional) accept notification, authors were given a few days to consider how to address the reviewers' comments and email a revision plan to their shepherd. Authors and shepherds then agreed on a timeline that allows the authors to complete the revision, providing enough time for the shepherd to read, consider, and discuss the revision with the authors, while permitting a final round of text polishing if necessary before the camera-ready deadline. At the end of this process, shepherds explicitly "signed off" the inclusion of papers in the program using HotCRP tags, allowing the program chairs to track the progress of turning all conditional accepts to accepts.

2.9 Accept as Short

As members of former ATC PCs, we are aware of full submissions that were accepted to past ATC-s on the condition that their authors will reduce their size to meet the short paper page-limit requirement. ATC program committees made such decisions rarely, limiting them to situations where the alternative is to otherwise reject the paper.

Surprisingly, past ATC call-for-papers were not clear about the possibility to accept as short; the practice was only anecdotally documented in the messages from chairs [2]. Seeing that this practice has been used in the past and may be used in the future, in the interest of transparency, we decided to explicitly declare it in the CFP, which now states that "the program committee may rarely decide to accept a full submission on the condition that it is cut down to fit in the short paper page limit" [19].

This CFP update initiated a discussion with USENIX board members who were concerned that the effort required to transform a full submission to a short paper might be too significant to accomplish between the authors notification date and the camera-ready date. They cited the FAST policy—which states that "the program committee will not accept a full paper on the condition that it is cut down to fit in the short paper page limit" [20]—as potentially preferable.

After consideration, we decided to keep the ATC accept-as-short policy because we believe it produces a significantly better outcome for both the authors and for the community, provided the alternative is to reject. In such rare cases, disallowing the PC to accept as short would result in a lose-lose situation: the authors lose because they are rejected instead of being given a chance to shorten and thereby get accepted; the ATC program loses a short paper; and the systems community loses because the paper would be subsequently resubmitted and hence re-reviewed, requiring the community to spend additional reviewing cycles, whereas reviewing load is already too high.

[1] In odd years, if the appropriate coordination takes place (as is the case this year), ATC notifications occur shortly before the SOSP submission deadline, to allow rejected authors of the former conference to submit an improved version of their study to the latter conference, assuming they have kept working on it while it was under submission at ATC.

2.10 Shorter Presentations

Last year's aforementioned 33% increase in the number of ATC submissions (377 in ATC '18 vs. 283 in ATC '17) and the consequent 27% increase in accepted papers (76 in ATC '18 vs. 60 in ATC '17) motivated the program co-chairs of ATC '18 to avoid hosting "best of the rest" sessions in their program, as well as to generate a longer-than-usual program that ends in the evening of the third day of the conference rather than around lunch time.

Despite having a similarly-sized program this year (71 papers), we wanted to have our cake and eat it too, namely: bring back the "best of the rest" sessions; further add lightning sessions to the program (see Section 2.12); while still end the program around lunch time at the third day, as was done in previous years prior to ATC '18.

To this end, this year, we decided to shorten the presentation time from 25 minutes per paper to 20 minutes. We believe that this change constitutes a reasonable compromise, allowing the conference to accommodate the additional sessions within the traditional time frame, while still providing enough time for presenters to convey the gist of their ideas.

2.11 Poster Requirement

To partially compensate for the shorter presentation time slots, this year, we dedicated the two poster sessions exclusively to accepted papers, and we required all paper-presenting authors to additionally present a poster in one of these sessions. Hopefully, this format will promote and facilitate interaction between authors and attendees who are interested in their work.

2.12 Lightning Sessions

In recent years "lightning sessions" have become standard in top-tier computer architecture conferences (ISCA, ASPLOS, etc.), and this year we decided to adopt them in ATC. Lightning sessions are typically interesting and fun, and, importantly, they are particularly suitable for conferences that have parallel sessions, which inevitably means attendees miss some of the presentations they are interested in. Lightning sessions give attendees a chance to make more informed decisions regarding what interests them the most and which talks are more worthy of their time. Speakers indeed often treat their lightning session presentations as previews aimed at soliciting listeners to attend the associated talks.

A lightning session is a joint session at the beginning of the day, which includes all the talks that will be given on that particular day. After the daily lightning session, the conference splits into its parallel tracks. Shortly before the daily lightning session, the speakers of that day queue in order—they do not sit until they present. Then, each lightning talk is allocated 120 seconds.

Each daily lightning session has a session chair. The chair is responsible for: interacting with speakers to get their slides beforehand; ordering slides on her laptop based on their order in the program, and making sure they display nicely; informing the speakers regarding the order; and regulating time during the session if necessary (we have never witnessed a lightning session chair having to actually exercise this authority).

Lightning speakers are additionally requested to submit lightning videos beforehand, which are made available in the conference web page before the conference. Both lightning presentations and videos are currently available in the ATC '19 technical sessions webpage.

In the past, USENIX conference talks were videoed, a very useful service that largely stopped due to financial reasons. Our hope is that lighting videos, which do not incur video recording costs, can partially provide some of this service: optimally, lightning videos would allow people who wish to only understand the gist of the idea to do so in 120 seconds.

3 Changes to Consider

3.1 Steering Committee

The one remaining notable difference between ATC and its sibling academic systems conferences (USENIX-sponsored: FAST, NSDI, OSDI, USENIX Security; SIGOPS-sponsored: ASPLOS, Eurosys, SOSP) is that ATC does not have a formal, broad, long-term steering committee. To make ATC more valuable to the community, we—nearly all ATC program chairs since 2015—believe that ATC should have such a committee, and we propose to form it, thus completing the transition of ATC into a conference that is governed by policies generally acceptable in the academic systems community.

We propose that the newly formed ATC steering committee will assume all responsibilities typically assigned to such committees, including providing advice and guidance to the current program co-chairs, selecting future program co-chairs, sustaining organizational memory, suggesting and considering new ideas when the need arises; and ultimately shaping the role of ATC. The identity of the steering committee members should be publicized along with call-for-papers to allow interested parties to address the committee with respect to matters that concern the conference long-term.

The members of the committee could, for example, be the USENIX executive director, relevant members of the USENIX board, and the program chairs from the last n ATC instances, such that members who chaired ATC in year $Y-n$ will be replaced by the ATC chairs of year Y shortly after the latter conference takes place. Joining the steering committee will of course be voluntary.

In October 9, 2018, a letter consisting of the content of this subsection has been submitted to the USENIX board. The letter was signed by all the ATC program chairs since 2015 except two (one responded too late and the other serves on

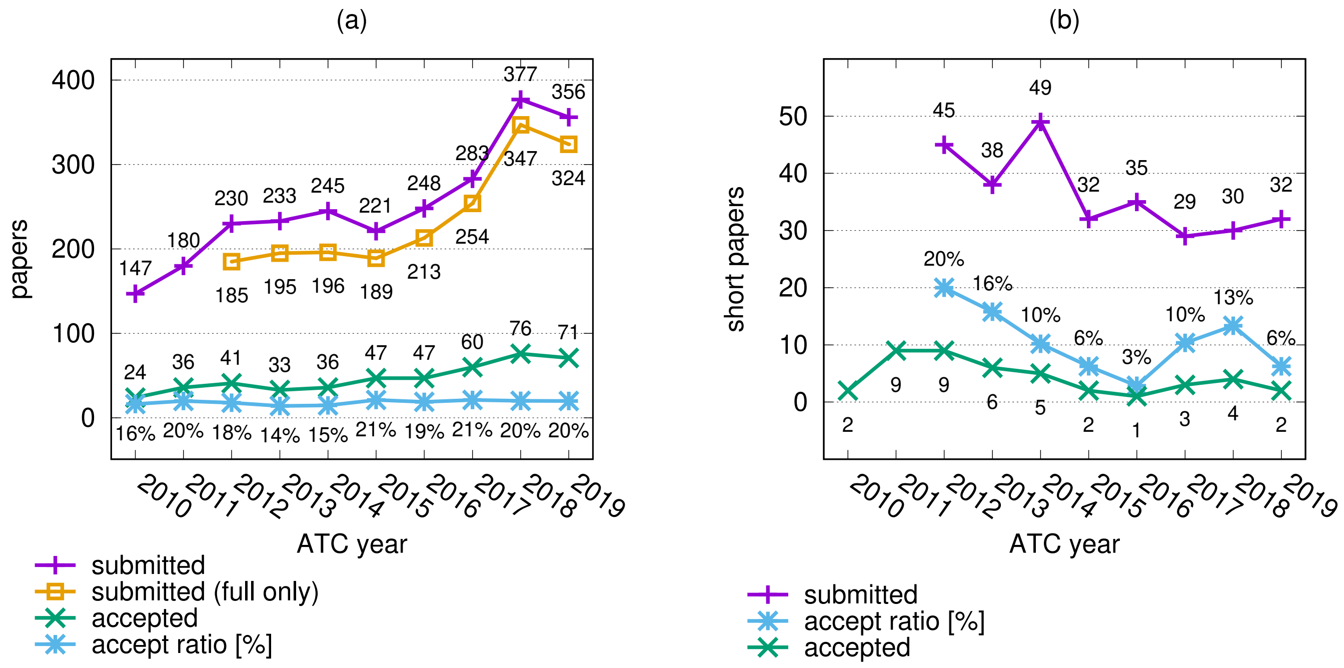

Figure 1: *(**a**) Submission and acceptance statistics of ATC papers (including both full and short) in the last decade, based on the corresponding proceedings' message from the ATC program chairs. (**b**) The same, but only for short papers. In 2010–2011, short submission numbers were not reported. In 2019 and 2013, one and three of the accepted short papers were submitted as full, respectively; we do not have this information for the other years.*

the Usenix board and is one of the decision makers regarding the steering committee issue). All who signed volunteered to serve on the steering committee when it is formed. The response of the relevant people in USENIX seems positive, but a steering committee has not yet been established.

3.2 Test of Time Award

All the premier systems conferences—except ATC—annually recognize historical, influential papers that have stood the test of time. This includes: USENIX Test of Time Award for FAST, NSDI, and USENIX Security [18]; SIGOPS Hall of Fame Award, which is typically handed to SOSP and OSDI papers [17]; Eurosys Test of Time Award [16]; and SIGARCH/SIGPLAN/SIGOPS ASPLOS Influential Paper Award [15].

The procedure to select the awarded papers varies. A common way employed is for the program committee of the conference to nominate influential papers published in that conference around ten years ago, with the final selection being made by the conference's steering committee (which, as noted, ATC still does not have). ATC is much older than ten years and, arguably, has changed its nature and goals over the years. So when/if an ATC test of time award is established, the steering committee will need to decide how to address older papers and handle the backlog. Jeff Mogul documented some of SIGOPS's considerations when establishing its Hall of Fame Award in 2005 and addressing similar questions [12].

3.3 Short Submissions

Figure 1a shows the submission and acceptance numbers of ATC papers in the last decade. Figure 1b shows the same statistics for short papers only. Getting a short submission accepted to ATC is clearly harder. We do not know why and can only speculate about the reason. Perhaps there is a mismatch between PC members' expectations and what can actually be accomplished in the scope of a short paper. Perhaps authors wrongfully believe that the bar for short submissions is lower. And perhaps there is a loose negative correlation between the increasing number of full submissions and the decreasing number of accepted short papers because PC members feel they have stronger papers to accept, relatively speaking.

Regardless of the reason, the fact that ATC PCs have reviewed 29–35 short submissions per year in the last five years only to accept 1–4 of them raises the question of whether the effort is worth it, since the reviewing effort to accept short papers is significantly greater than the effort to accept full papers (3%–13% vs. about 20% acceptance rate for short and short+full submissions in the last five years, respectively).

This year provides an extreme demonstration of how much harder the PC has to work in order to accept short papers. Table 2 specifies the number of reviews that the ATC '19 PC wrote for full and short submissions, as well as the resulting

scenario	*submission type*	*written reviews*	*accepted papers*	*work ratio*
real (worst case)	full	1620	70	23:1
	short	132	1	132:1
extrapolated (best case)	short	132	4	33:1

Table 2: *The number of reviews that the ATC '19 PC wrote for full and short submissions demonstrates that the PC had to work much harder in order to accept a single short paper ("real"). Even if we hypothetically assume that the PC had accepted four short paper instead of one as in last year (best case scenario in the last five years), the reviews-to-accepts work ratio would still be nearly 1.5x higher ("extrapolated").*

number of accepts. It turns out that the PC wrote 132 reviews in order to accept a single short paper, as opposed to writing "only" 23 reviews in order to accept a full submission. Namely, the PC had to work nearly 6x times as hard.

That said, as can be seen in Figure 1b, this year has been especially bad for short submissions. But even if we hypothetically assume the best case scenario across the last five years of accepting four short papers, the corresponding reviews-to-accepts ratio would have been 35:1, which is still nearly 1.5x harder than accepting a full paper.

ATC enjoys a steadily increasing number of full submissions. As a consequence, the reviewing load becomes heavier, requiring bigger PCs that already hardly fit into one room. Considering the relatively low return on investment (a significantly higher reviews-to-accepts ratio), it may make sense for future ATCs to consider to stop soliciting short papers.

We note in passing that, this year, we revised the CFP definition of short submissions to exclude workshop-style papers ("a short paper is not like a workshop paper—it presents a complete idea, which does not require full length to be appreciated" [19]). We introduced this change hoping to increase the short submission success rate by discouraging authors from submitting work that (our experience suggests) ATC reviewers tend to reject. The data shown in Figure 1b suggests this change was ineffective .

3.4 Early Rejects or R1 Rebuttals

The program co-chairs of this year debated about the issue of whether or not to send early reject notifications to authors of submissions who did not make it to R2. The reasoning to oppose sending early rejects was that such notifications might provide an unfair advantage to R1 rejects over R2 submissions that will be rejected later on, because the authors of the former will be free to resubmit their work elsewhere much sooner. Additionally, early rejects might translate to even higher reviewing loads that the community must handle due to said earlier resubmissions. Lastly, and importantly, postponing the R1 reject notification would allow PC members to re-calibrate during the second round and the deliberations and potentially change their opinion.

The reasoning to supported early rejects was that delaying reject notifications would be counterproductive for authors who do not abuse the system but rather leverage the reviewers' feedback to improve their work before they resubmit. Arguably, the ATC reviewing process should not replace one evil ("helping" authors who might abuse the system by ignoring the reviewers' feedback and resubmitting prematurely) with another (allowing authors to believe that they have a chance to get accepted for a good few weeks whereas in fact they do not).

Eventually, since we already introduced many changes to ATC this year (Section 2), we decided to leave things as they are in this particular case and avoid sending early reject notifications. But we encourage future ATC program chairs (and/or the ATC steering committee if it is established) to reconsider.

Because decisions were collectively sent to authors shortly after the PC meeting, R1 rejects were given a chance to write a rebuttal (Section 2.3), which the committee members read and considered. Two R1 rejected submissions were resurrected as a result. These submissions were promoted to R2 and urgently assigned two additional reviewers. In the end, however, both were rejected. We speculate that allowing authors to rebut (also) after R1 (as is done by some conferences) would have had a bigger effect. But doing so would require more labor and an even earlier deadline, which would be closer to New Year, which might result in fewer submissions (see Section 2.7).

3.5 Physical PC Meeting

The number of submissions the PC can discuss in one day (let us denote it as c) is bounded. For example, it takes more than eight hours to discuss $c = 70$ submissions if allocating 7 minutes per submission, as is typical. PCs also usually dedicate 2–3 minutes to present each submission that was pre-accepted in the online discussion phase (ATC '19 had 37 such submissions), and they take about 30 minutes for lunch. It is challenging to squeeze all these activities into one day.

Let m denote a member of the PC, and let r denote the number of submissions reviewed by m. Similarly to c, the value of r is bounded. At the risk of overgeneralizing, we roughly approximate that $r = 15$, $r = 20$, and $r = 25$ reviews per member are nowadays considered light, average, and heavy reviewing loads in academic systems conferences, respectively. The value of r cannot be raised arbitrarily.

In contrast to c and r, the total number of submissions that the PC must review (let us denote it as n) is unbounded and keeps increasing. The practical meaning of this increase is that, on average, fewer and fewer of the r submissions that m reviewed are getting discussed at the meeting. Figure 2 demonstrates this trend, assuming $c = 70$ submissions are discussed at the meeting, and that 2/3 and 1/3 of the r submissions assigned to m are reviewed in R1 and R2, respectively. The x axis shows n, and the y axis shows the corresponding

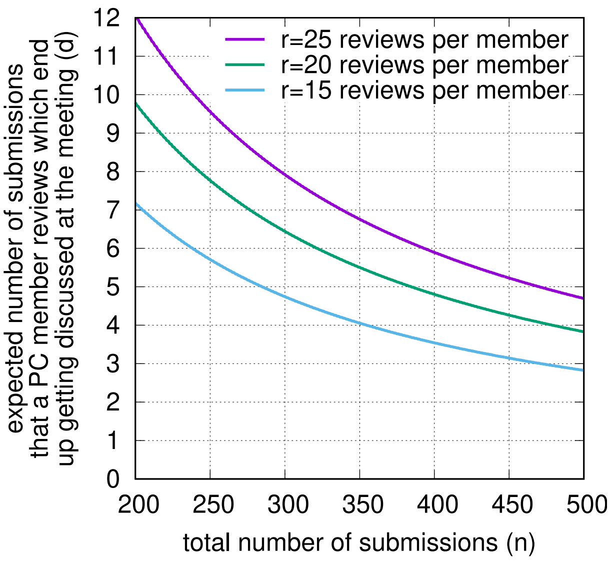

Figure 2: *Increased number of submissions translates to fewer submissions that each PC member gets to discuss at the PC meeting; see Appendix A for details.*

expected number of submissions that have been reviewed by m and ended up being discussed at the meeting (let us denote it as d), which is monotonically decreasing.

In Appendix A, we show that under our assumptions, $d \approx 4rc/3n$ is a reasonable approximation of the expected number of submissions that m reviewed and discussed at the meeting. As noted, because r and c (numerator) are bounded, d asymptotically behaves like $1/n$ (denominator).

Our PCs received $n = 356$ submissions and used an upper bound of 18–19 reviews per heavy member, which more or less corresponds to the line associated with $r = 20$ in Figure 2. In the relevant range of n, we see that $d = 5.5$ submissions discussed at the meeting per member. Because d is just an average, some members discussed more submissions, but others discussed less: as little as 2–3 submissions in certain cases. Flying to California to discuss such a small number of submissions is, arguably, counterproductive.

In 2018, the PC meeting spanned across two days, allowing the committee to make fewer decisions during the online discussions period and instead discuss $c = 124$ submissions in person at the meeting (with $n = 377$ and $r = 18$). Therefore, by our calculation, each member discussed about 8 submissions on average, alleviating the problem somewhat. On the other hand, 8 submissions during two days means 4 submissions per day (as compared to 5.5 per day in 2019), which is not necessarily preferable.

When discussing this issue with some of the members during the PC dinner, it seemed like most agreed that there is a problem: the time overhead and carbon emission associated with physical PC meetings are possibly becoming excessive considering the smaller number of submissions that each member gets to discuss. Still, there was a sense that the program turned out better due to the physical meeting, which allowed the members to calibrate. Additionally, several members—both junior and senior—pointed out that a notable value they get from PC meetings is the chance to network and interact with their peers.

In light of the above, it may be advisable for future program chairs to consider if in-person, physical PC meetings are worth it, at least in their current format. If they decide in favor of physical meetings, one conceivable way to increase their value is, for example, to couple them with workshop-style events, where committee members briefly present their ideas and get feedback from their peers.

4 Assembling the Committee

After we accepted the position of the ATC '19 program co-chairs, we were asked by USENIX to take into account that the number of submissions in 2019 might exhibit the same growth rate as it did in 2018, which would bring us to about 500 submissions (a.k.a. "the nightmare scenario" :-)), requiring $3 \times 500 + 2 \times 250 = 2000$ reviews assuming 50% of the submissions move to R2 (see Section 2.1). A smaller, more conservative estimate of 400 submissions would require $3 \times 400 + 2 \times 200 = 1600$ reviews. In comparison, a sizable heavy PC of 60 members each contributing 20 reviews—a threshold we were hoping and planning not to exceed—provides $60 \times 20 = 1200$ reviews. Taking into account these numbers, we decided to draft a heavy PC, a light PC, and an ERC (see Section 2.5) with target sizes of 65, 25, and 25, respectively.

Drafting about 115 committee members is a challenging task. In preparation for it, we compiled a list of all those who served on PCs in the last three instances of the main systems conferences, such that we had a pool of candidates to helps us (we used: ASPLOS 2017–2019, ATC 2016–2018, Eurosys 2017–2019, FAST 2017–2019, NSDI 2017–2019, OSDI/SOSP 2016–2018, and USENIX Security 2016–2018).

Analyzing this database brought up an interesting insight, which might indicate that our community has scalability issues in terms carrying out the reviewing load. Table 3 shows the relevant statistics. The aggregated sum of the size of the 21 PCs we have included in our analysis is 1118. These membership positions were manned by 655 unique individuals, a finding that could be interpreted to mean that members serve in $1118/655 \approx 1.7$ PCs in three years, on average. A deeper look at the data, however, reveals that 284 individuals participated in two or more of the PCs in our database, and these individuals are responsible for manning 783 (70%) of the 1118 positions. This finding implies that a relatively small group of people shoulders most of the reviewing load.

Figure 3 depicts the histogram of how many of the members in our database (y) served in how many of the PCs that we included (x), which demonstrates the reviewing effort dis-

memberships (aggregated sum of PC sizes)	1,118
number of unique members	655
number of unique recurring members	284

Table 3: *Membership statistics of the PCs of the main systems conferences in the last three years.*

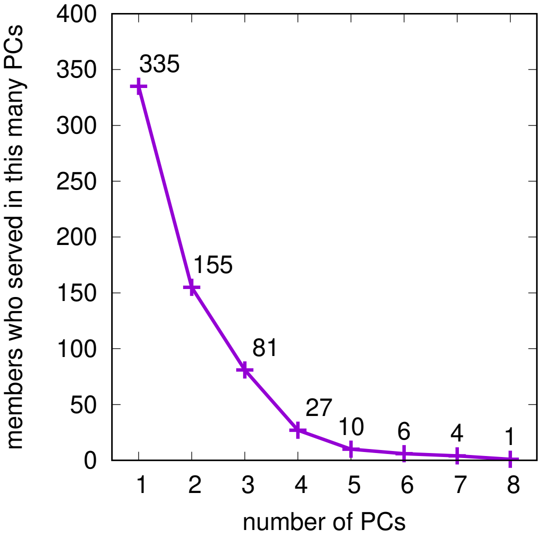

Figure 3: *Histogram showing how many of the members of the PCs of the main systems conferences in the last three years (y axis) served in how many of these PCs (x axis).*

parity. We can see, for example, that one member served in 8 PCs, and four members served in 7.

The list we compiled was helpful in drafting the committee. When sending heavy member invitations, we allowed the candidates to accept as light or ERC, and when sending light member invitations, we allowed the candidates to accept as ERC. The number, type, and outcome of the invitations are specified in Table 4, and the demographic information of the resulting PC is specified in Table 5. Nearly 2/3 of the invitations sent were accepted, and as can be seen, this relatively high success rate is partially because we allowed candidates to opt for roles that involve a smaller reviewing load.

invite type	invite sent	accepted as heavy	accepted as light	accepted as ERC	declined
heavy	131	66	16	3	46
light	22	-	12	3	7
ERC	27	-	-	16	11
sum	180	66	28	22	64

Table 4: *Number of invitations to serve on the ATC '19 committee sent to candidates, and the corresponding responses.*

seniority	junior	31
	senior	63
gender	female	14
	male	80
sector	university	64
	industry	25
	both	5
continent	N. America	69
	Europe	15
	Asia	5
	Middle East	4
	Australia	1

country	USA	62
	Canada	7
	Switzerland	6
	Israel	4
	UK	4
	Germany	2
	Netherlands	2
	Korea	2
	Australia	1
	China	1
	France	1
	Hong Kong	1
	Sweden	1

Table 5: *Demographic information of the PC (heavy and light, excluding program co-chairs).*

5 Conflicts and Bidding

5.1 Missing Conflicts

Before assigning submissions to reviewers, it is important for the submission management system, HotCRP, to have accurate conflict of interest information as defined by the ATC '19 call for papers [19]. In addition to the conflict information that authors and reviewers explicitly specify, HotCRP helps by highlighting potential conflicts based on the information available to it, which is productive. This year, we also used the PC Chair Kit [3] that was written for ISCA '18 to find missing conflicts based on authorship information available via DBLP.[2] The script downloads the relevant DBLP information and checks if there are any co-authors of submission authors from the last n years that are not already listed as HotCRP conflicts.

Our submission co-chairs found 150 such undeclared conflicts and verified them manually. They identified a few false positives (e.g., two researchers with identical name, a summer school report authored by many authors that should not be considered as a real conflict), but the rest of the conflicts were valid.

5.2 Helping Committee Members to Bid

Authors associate topics from a predetermined list with their submissions, and committee members declare their per-topic level of (dis)interest for each such topic. This information is important, because it is utilized by HotCRP to compute a per-member score for each submission, and members use these scores to sort through hundreds of submissions and thereby ease the process of bidding—the act of associating integers with submissions to indicate reviewing preference. HotCRP then uses bids (as well as topic scores when, e.g., bids are absent) to assign reviews to reviewers.

Instructions for Committee Members We requested committee members to favor bidding on submissions for which

[2] More accurately, we used a fork of that kit [7].

they can provide expert or knowledgeable reviews, rather than on submissions that they find interesting but do not fall in their area of expertise.

We additionally requested committee members to limit the range of the numeric values they use to express preference to -20 to 20. The HotCRP system does not compare preference values of different users in the automatic review assignment algorithm and so members need not use the same scale. Some review assignments, however, are inevitably done manually by program chairs, and then having a common scale is helpful.

Defining Topics Last year, in ATC 2018, the aforementioned predetermined list consisted of 62 topics, as opposed to years 2017 and 2016, at which ATC used a list consisting of 17 topics. Some speculate that having this many topics is cumbersome, overly verbose, and unhelpful [9], and we seriously considered minimizing the list and consolidating topics when defining it for 2019. But a closer look at the historical data (from ATC '18, as well as from ASPLOS '19, which used a similarly sized list) indicated that authors and reviewers do use most topics in the longer lists.

Considering that (1) the task of bidding is really hard when there are hundreds of submissions, and that (2) PC members do primarily rely on topics when bidding as a way to cope with this submission volume, we eventually decided that it might be counterproductive to shrink the topic list and risk making bidding harder. A concise (or at least coarser grained) list could be preferable, and mining past data more seriously may provide evidence that support this hypothesis. But as we currently do not know, we decided to stick with the more sizable, finer grained list (although we made changes).

Figure 4 shows the 59 topics used in ATC '19, ranked by the number of submissions that used them. It could be argued that even our least popular topic ("cryptography", which was associated with only three submissions) is worthwhile, because it is preferable for the associated submissions to be reviewed by the appropriate committee members who are actually capable of doing it, and it seems reasonable to speculate that the odds of that happening would have been smaller without the topic.

Grouping Topics Given that there are dozens of topics, it makes sense to group related topics when they are presented to authors and committee members within HotCRP, which makes using them easier. In ATC '18, the program co-chairs did so in an ad hoc manner by adding grouping prefixes to topic strings that are separated from the topic names by a colon (for example: "storage:deduplication", "storage:disk (CMR, SMR, etc.)", "storage:erasure coding", and so on). In ATC '19, we used the same notation but also kindly requested the HotCRP maintainer to directly support the concept, which he did [9], making the HotCRP presentation of grouped topics more elegant, usable, and effective. The topic groups we used are: general, devices, networking, OS, PL/SE (abbreviation of programming languages and software engineering), security, storage, systems, and techniques/aspects.

total number of citations of committee papers	1266
average number of citations per member	11.6
median number of citations per member	7
standard deviation	11.5
citations of top-most cited member	67
citations of 2nd-most cited member	61
citations of 3rd-most cited member	43

Table 6: *Statistics of citations of committee member papers found in the ATC '19 submissions and communicated to members to help with their bidding.*

Pinpointing Submissions that Cite Members As noted, having to place bids to decide which submissions to review is becoming more challenging due to the increasing number of submissions. Merely reading the titles of 300–400 submissions is time-consuming, and many reviewers need more information than just the title to decide to bid. Attempting to ease the process of bidding, we generated for, and shared with each committee member a list that specifies all the ATC '19 submissions that cite that member's papers. The list was generated by our submission co-chairs using the aforementioned PC Chair Kit [7].

Table 6 provides some statistics about the citations we have found. Since there are more than a thousand of them, hopefully, they provided a usable signal to some of the committee members.

Dealing with Unpopular Submissions Despite the fact that nearly 90% of the committee members placed positive bids on 20 submissions or more (and 2/3 of the members placed positive bids on 40 submissions or more), some submissions were associated with relatively few positive bidders. Perhaps unsurprisingly, some submissions are much more popular than others. The line associated with "before" in Figure 5 depicts the disparity of popularity. The x axis shows the rank of each submission based on the the number of members that bade positively on it, and y axis shows the corresponding number of bids.

Focusing on the bottom right, we can see that 60 submissions received only 6 positive bids or less, which would have likely hampered the review assignment process. We therefore labeled these 60 as "lowbids" in HotCRP and asked our committee members to consider positively bidding on some of them if they are within their domain of expertise, stating that if everyone does this truthfully, no one will be tasked with arbitrary submission assignments. The line associated with "after" in Figure 5 demonstrates that this request was effective. (Albeit the data is distorted somewhat by the fact that the "after" line additionally accounts for bids we solicited before the beginning of R2.)

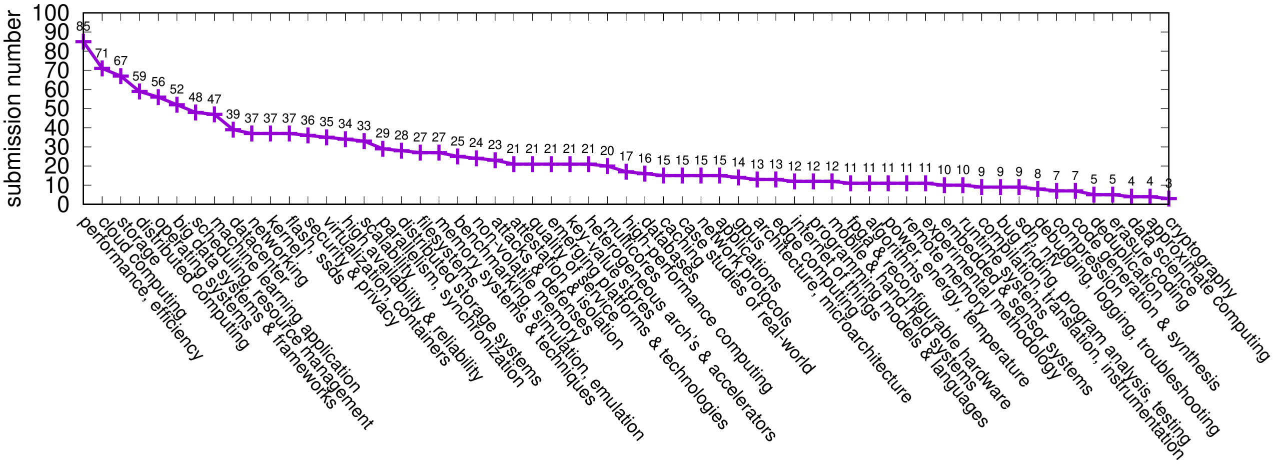

Figure 4: *The ATC '19 topics (without their grouping prefix) ranked by the number of submissions that used them.*

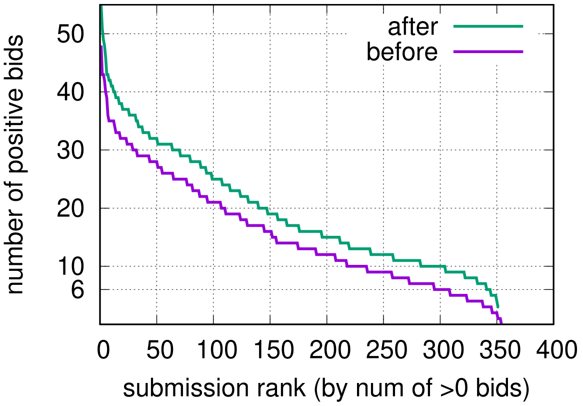

Figure 5: *Disparity of popularity among ATC '19 submissions.*

Interestingly, out of those 60 "unpopular" submissions, only three (5%) were accepted to the ATC '19 program, which is 4x lower than the overall acceptance rate. Perhaps this poor success rate suggests that bidding information could be leveraged somehow to make the reviewing process more efficient? A positive answer to this question would be helpful, because the ATC '19 committee wrote 281 reviews for these particular 60 submissions, which is a lot of effort in order to accept only three.

On the other hand, one of these three has been awarded best paper, which is another demonstration of what all of us already know: popularity isn't everything... :)

6 Planning for a Dual-Track PC Meeting

Due to the increasing number of submissions to system conferences, in order to be able to finish the PC meeting on time, several recent program chairs resorted to splitting the meeting into two parallel tracks for part of the time, such that each track is simultaneously headed by a different co-chair. Assuming that the number of submissions is not going down any time soon, it seems like dual-track meetings are here to stay.

However, properly organizing a dual-track meeting is challenging. Notably because it may affect how submissions are assigned to reviewers, as it is nontrivial to arranges things such that *all* PC members are *always* found in the right room at the right time while the meeting takes place.

Currently, there is no standard, generally accepted best-practice for how to arrange a successful dual-tack PC meeting. Program chairs typically need to apply creativity and to spend much effort to come up with an appropriate model they feel would work and would be suitable for their committee. For this reason, before we describe the model we used, we survey the models of dual-track meetings used by program chairs before us, and we briefly discuss their pros and cons. Hopefully, this discussion would be useful for future chairs when deciding upon the model that works best for them, as the state of the art of dual-track PC meetings evolves.

ASPLOS '17 Model The first PC that we are aware of that split into tracks occurred spontaneously at the PC meeting of ASPLOS '17, when attending members and chairs realized it was not realistic for them to finish on time. They therefore split in an ad hoc manner to flexible, parallel discussion groups. The approach was reported to have worked: the program was ready at the end of the day, and the members lived to tell the tale.

ASPLOS '18 Model In the subsequent year, having experienced the difficulties from the previous year, the program co-chairs of ASPLOS '18 carefully planned for the dual-track meeting. They split their PC members into two disjoint equally-sized sets M_0 and M_1, with the stated goal of having equal expertise in both, in all the relevant conference topics. They likewise split the submissions into two equally sized sets S_0 and S_1, and they exclusively assigned submissions from S_i to M_i, such that no PC member reviewed outside of her sub-committee's pool of submissions. Consequently, by design, running the dual track meeting was easy.

A main concern with this model is that it splits the expertise and thus runs the risk of arbitrarily preventing the most appropriate experts who happen to belong to M_i from reviewing submissions that happen to belong to the "wrong" pool $S_{(i+1) \bmod 2}$.

ASPLOS '19 Model In an effort to alleviate this drawback, the program co-chairs of ASPLOS '19 employed the following approach in deciding how to define M_i and S_i. ASPLOS is an interdisciplinary venue of three communities: SIGARCH (50% sponsorship), SIGOPS (25% sponsorship), and SIGPLAN (25% sponsorship). Accordingly, the chairs initially divided their PC into M_{OS} and M_{PL} containing members from the operating systems community and the programming languages community, respectively. They then searched for an "optimal" division of the PC members from the architecture community into two parts, each added to the initial M_{OS} and M_{PL} to form two equally-sized M_{OS}^{arch} and M_{PL}^{arch} sets that, together, comprise the entire PC.

The said optimality was achieved as follows. The chairs and their helpers used a script that exhaustively enumerated all the possible equally-sized M_{OS}^{arch} and M_{PL}^{arch} group partitions. For each partition, they assigned every submission to the group that maximizes the submission's "affinity" (a combination of reviewer citations, topic score, and normalized bids). Then, they scored that partition by aggregating the affinity across all submissions within their assigned group. The final partition was the one that scored the highest by this metric.

They then calculated the "partitioning penalty" for each submission, which is the total affinity of the submission for the whole PC minus its affinity to the group it was assigned to. They assigned high partitioning penalty papers to the whole PC, thus adding a requirement for a *joint session* at the meeting, in addition to the dual track. To make workload for the two groups even, they took the most highly penalized papers from the larger group and assigned them to the whole PC.

The ASPLOS '19 model is more careful in how it splits S_i and M_i as compared to the ASPLOS '18 model, trying to minimize the penalty associated with splitting. It additionally supports submissions that are discussed jointly. Still, while minimized, the penalties do exist.

We note in passing that the ASPLOS '19 program co-chairs received extensive help in planning for their dual-track meeting from individual whose role was similar to what we formalized as "submission chairs" (Section 2.4).

ATC '18 Model The program co-chairs of ATC '18 decided not to split the PC beforehand and globally assign reviews across all members without any constraints. This approach is simple and entirely eliminates the penalties of splitting. The cost, however, is shifting all the administrative complexity to the PC meeting itself: it raises the question of how to run the dual-track meeting without resorting to the ASPLOS '17 model, which seems to have heavily relied on luck.

The ATC '18 program co-chairs did not rely on luck. They were successful in planning the dual-track PC meeting after (1) all the reviews have been uploaded, (2) the online discussions have been concluded, (3) the list of submissions to be discussed at the meeting have been finalized, and (4) it became known which PC members will call-in rather than attend physically.

The PC meeting timeline was divided into several consecutive sessions T_i $(i = 1, 2, ...)$, such that in each session T_i the PC was split into two groups T_i^i and T_i^{ii} that met in parallel. The group membership changed across sessions, so group T_1^i was different than group T_2^i, for example.

In some sessions, groups T_i^i and T_i^{ii} were disjoint. But in other sessions, some PC members were instructed to physically move to the other group at some point, but such transitions were limited to one move per one member per session. In such non-disjoint sessions, PC members were asked to be aware of the discussion schedule so as to know when to make the transition. But inevitably this did not always work, and so occasionally members were called from the other room. Still, the program co-chairs reported that, overall, the movement between rooms was minimal and not distracting.

One ATC '18 co-chair concluded that "if I would repeat, I would not change what [we] did because it worked fine, and the PC didn't seem to be bothered to move around." But the other co-chair reported that "I would avoid doing what we did in the future even though it worked amazingly well. We lucked out [...], and we barely pulled it off."

Similarly to ASPLOS '19, the ATC '19 program co-chairs received extensive help in scheduling the PC meeting from individuals whose role was similar to what we formalized as submission chairs.

ATC '19 Model Like the program co-chairs of ATC '18, we wanted to refrain from the penalties and complexities involved in splitting the PC beforehand in a manner that affects how reviews are assigned. But we also wanted to completely avoid the aforementioned transitions between rooms, the occasional missing members that must be fetched from elsewhere, and—perhaps most importantly—the sense of uncertainty associated with the "barely pulled it off" sentiment quoted above. We achieved all these goals as described next.

Immediately after the submission deadline passed, the committee members placed their bids, and missing conflicts were identified and uploaded, we repeatedly applied the following simulation procedure.

1. Using standard HotCRP functionality, simulate assigning three R1 reviewers to all submissions as if for real.

2. Randomly select 50% of these submissions (177 in our case) to be the simulated R2 submissions; let us denote this random set as S_2.

3. Using HotCRP functionality yet again, simulate assigning two additional R2 reviews by heavy members to all the submissions in S_2.

4. Randomly select 50% of the S_2 submissions (88 in our case) to be the simulated set of submissions to be discussed at the meeting; denote this random set as S_3.

5. Using a constraint solver, find a split of the heavy PC into two groups that allow for the longest simulated dual-track parallel session of submissions from S_3 (without any transitions of members between the two groups); submissions that cannot be discussed in parallel in this split, will be discussed in a simulated joint session.

6. Compute the time it takes to run these simulated parallel and joint sessions, assuming a 6–7 minutes discussion per submission. If the simulated meeting takes less than eight hours, declare success; otherwise declare failure.

Our submission co-chairs repeated the above procedure multiple times using multiple random selections, and they verified that it *always* declared success. We therefore gained confidence that scheduling our dual-track meeting using a constraint solver is doable, despite using a global review assignment. This was indeed the case in the actual PC meeting.

Before running the above experiment, we did not know whether or not it would be successful, and we were prepared to get a negative result. In this case, we planned to use the framework we developed to attempt to understand the root cause of the failure, and to try to devise constraints for the baseline HotCRP review assignment algorithm that would resolve the underlying issue. Thankfully, we did not have to do that.

HotCRP Multi Live-Meeting Trackers HotCRP has a useful live meeting tracker feature, which helps program chairs run the meeting by keeping attendees in sync, presenting the current and next submissions discussed and the relevant conflicts. The problem was that HotCRP assumed a single track meeting, making the tracker unusable in the case of dual tracks. Thankfully, again, the HotCRP maintainer was willing to accommodate our request to add support for multiple live-meeting trackers [10], which we indeed used in our meeting.

7 Review Assignment Improvements

The review assignment is done by HotCRP using a min-cost max-flow algorithm [8, 11]. This assignment utilizes member bids and topic scores in order to distribute the reviews among reviewers in a manner that attempts to be balanced and fair, both in terms of number of reviews assigned to each member, and in terms of the bidding preferences, such that everyone would hopefully get as many of their top bids as possible.

The review assignment process of the individual conferences frequently involves some constraints that must be taken into account when the assignment takes place. In the case of the first review round of ATC '19, these were: (1) each PC member gets an assignment of 13 reviews; (2) each ERC member gets an assignment of 5 reviews; and (3) each submission gets at least 2, and at most 3, reviews by heavy members.

There is no way we are aware of to express multiple constraints such as these all at once in HotCRP (nor in the underlying min-cost max-flow algorithm, we believe). Instead, a sequence of assignments is conducted that is applied to the various types of members: first heavy, then light, then ERC, and some creativity is involved to get the desired outcome, which is an assignment that adheres to all the constraints.

With the goal of checking the quality of the resulting assignment, we have defined the per-reviewer "goodness" metric as follows. Let n be the number of reviews assigned to the reviewer, namely, in our case, n is 13 and 5 for PC and ERC members, respectively. The goodness metric measures how many of the reviewer's most-preferred n submissions, associated with her highest bid values, were actually assigned to that reviewer. For example, if an ERC member was assigned her five most preferred submissions, then her goodness is $5/5 = 100\%$, but if she was assigned only one of them, then her goodness is $1/5 = 20\%$.

The line that approaches 0% in the bottom right of Figure 6 shows the goodness produced by the default HotCRP assignment algorithm for all PC/ERC members. The committee members are ranked based on their review goodness value, from highest to lowest, and this rank is displayed along the x axis; the y axis shows the goodness value of the corresponding members. The drop towards zero at the right indicates that the default algorithm might produce an unfair assignment when used as described above. Some members get all their top picks and some get none, with 31 members (more than 1/4 of the committee) members getting less than 60% of their top picks. Moreover, the default algorithm made 38 and 6 assignments where the bid placed by the corresponding members was zero or negative, respectively.

For these reasons, we implemented a script that helps improve the assignment as follows. Let r_i be a reviewer, s_i be some submission that r_i was assigned to review, and $b(r_i, s_i)$ be the numeric bid value that r_i placed on s_i. Our script initially attempts to exploit the fact that the default algorithm does not produce a stable marriage [21]. Namely, it is possible to find a subset of n reviewers r_i ($i = 0, 1, ..., n$), each assigned with a certain submission s_i, such that if r_i hands s_i to $r_{(i+1) \bmod n}$ and reviews $s_{(i-1) \bmod n}$ instead, then: (i) no conflict of interest is violated; (ii) $b(r_i, s_i) \leq b(r_i, s_{(i-1) \bmod n})$, namely, the new assignment is at least as good as the previ-

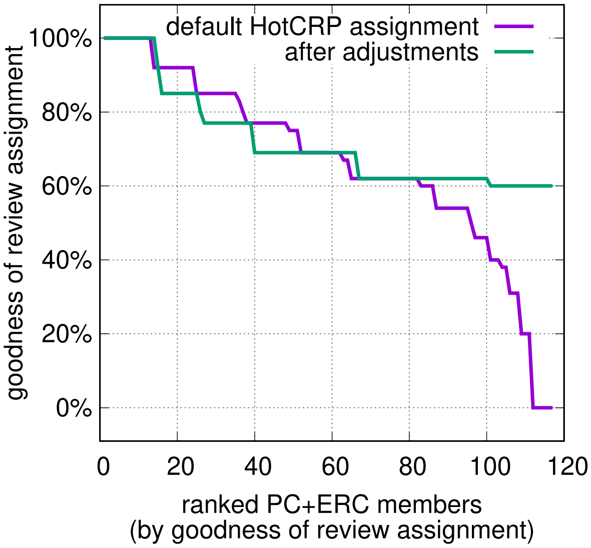

Figure 6: *Per-member goodness of the default HotCRP review assignment, which we improved, obtaining a lower bound of 60% through (i) review swaps that improved the assignments for all reviewers involved, or (ii) at the expense of reviewers who enjoy a much higher goodness value.*

ous for all reviewers involved; (iii) there exist at least one k $(0 \leq k < n)$ for which $b(r_i, s_i) < b(r_i, s_{(i-1) \bmod n})$, namely, the new assignment is better than the old for at least one reviewer; and (iv) each submission still gets at least two and at most three heavy reviewers.

The script is repeatedly applied to the member currently associated with the lowest goodness value, who assumes the role of r_k defined in constraint (iii). The script attempts to find a submission switch as defined above, using $n = 2$ and $n = 3$. If no such swap exist, the script relaxes constraint (ii) so as to tolerate goodness reductions due to the swap, provided that the reviewers that suffer the reduction still enjoy a high goodness value after the switch.

Our script initiated 748 HotCRP events to adjust the original default assignment, as specified in Table 7. In the end, as shown in Figure 6, we were able to ensure a minimal goodness value of 60% to all members (namely, PC members got at least 8 of their top-13 preferences assigned to them, and ERC members got at least 3 of their top-5). Additionally, we were able to arrange things such that all committee members were exclusively assigned submissions associated with their positive bids, with two types of rare exceptions: (1) reviewers whose number of positive bids was smaller than 13 for PC or smaller than 5 for ERC; and (2) submissions with only one positive bid by a heavy PC member. In the latter case, the heavy member with the highest topic score was assigned as the second heavy reviewer.

Processing of the review assignment for R2 was similar albeit somewhat more challenging to improve, due to having fewer usable bids, because only heavy members were assigned reviews, and also because of the additional constraint that we could only assign submissions to members who did not yet review them in R1.

HotCRP events	*purpose*
215	eliminate assignments with zero or negative bids
12	at most 3 heavy reviewers per submission
494	increase low goodness to promote fairness
748	sum

Table 7: *Number of individual HotCRP events affecting review assignment that were generated by our script to improve upon the default assignment of R1.*

Out of the 5–6 additional R2 reviews assigned to heavy members, the initial HotCRP review assignment assigned about 1/4 of the members with 1–5 submissions with which they associated a zero or negative bid. Anecdotally, one such member started off with *all* of his assignments having negative bids. Subsequently, we were able to adjust things such that all committee members were assigned submissions that are exclusively associated with their positive bids, with a few exceptions similar to those found in R1. Overall, half of the heavy PC members were assigned at least three of their (remaining) top picks, and all the them were assigned at least two of their top picks.

8 Reviewing Process

We employed a double-blind reviewing process consisting of two rounds, and we followed standard procedures for handling conflicts of interest. The PC consisted of 66 heavy and 28 light members, assisted by 22 ERC members. Additionally, 51 external reviewers contributed when specific expertise was required. The committee members were allowed to submit papers to the conference; the program co-chairs and submission co-chairs avoided it.

Table 1 summarizes the reviewing process. Out of 458 HotCRP registrations, we received a total of 356 submissions, divided into 324 full submissions (11 pages plus references) and 32 short submissions (5 pages plus references).

Format Violations We visually inspected all the submitted PDFs as well as used the HotCRP style checker to identify 29 submissions that violated the formatting rules. These were given a day to rectify the problem without making any content modifications; if fixing increased the size beyond the page limit, authors were required to remove (never change) content to meet the limit. All violating submissions complied except two, which were then rejected and withdrawn by the co-chairs.

Round 1 In Review Round 1 (R1), the PC members mostly contributed 13 reviews, and the ERC members mostly contributed 5 reviews. Out of all R1 submissions, 277 were assigned four reviewers, and 75 were assigned three reviewers. Regardless, all of the submissions were assigned at least two reviews by heavy members (typical), and at most three. The committee wrote a total of 1,347 R1 reviews.

Round 2 We promoted 184 submissions to Review Round 2 (R2). We assigned each R2 submission with two additional reviewers from the heavy PC. A submission was promoted to R2: (i) if two or more reviewers gave it a positive score ("weak accept" or above); (ii) if a single positive reviewer decided that she supports promotion after considering the other reviews and despite of them, and, if she has so chosen, discussing the matter with the other, negative reviewers; or (iii) if the submission had fewer than three reviews due to late members.

To qualify to be the aforementioned "single positive reviewer", a member must have assigned a score of "accept" or "strong accept". For submissions with three (rather than four) reviews, a "weak accept" also qualified, provided the associated expertise was at least "knowledgeable" or the confidence was "high". Out of the 40 single-supporter submissions (24 with one "accept" or higher), we promoted 17 to R2 (13 with "accept" or higher). The committee wrote 405 R2 reviews and a total of 1,752 reviews in the two review rounds.

Review Sufficiency Check A few days before the rebuttal period, we applied a Review Sufficiency Check (RSC) procedure to all R2 submissions, to ensure that the reviews provide sufficient feedback to authors, as well as sufficient information to the committee to make an informed decision regarding the submission. To this end, for each R2 submission, we appointed one of the reviewers who is a heavy PC member as the "lead" of the submission. Leads were responsible for conducting the RSC by: (1) reading all the associated reviews; (2) asking the relevant reviewers to revise their reviews when the need arises (e.g., by calling out subjective claims that a submission is incremental without adequate citations of prior work, by identifying unclear statements, etc.); and (3) deciding together with the other reviewers if additional reviews are needed when expertise is low.

Online Discussions After the authors uploaded their rebuttals, we discussed the submissions online. Our goal until the meeting was to: (1) revise reviews if needed due to rebuttals; (2) revive R1 submissions if their rebuttals justify it (this happened in only two cases); (3) discuss submissions and attempt to reach consensus, color-tagging them as red to indicate preliminary reject, green to indicate preliminary accept, and yellow to indicate that reviewers are unable to reach consensus, so the submission should be discussed at the meeting; and (4) for red submissions that have a rebuttal, as well as for green submissions, write a post-discussion summary comment, which will be made visible to authors after the PC meeting, briefly explaining the primary reasons for rejections and possibly ways to improve (red), or what is required for the camera-ready (green). Such a summary was eventually written for all submissions that uploaded a rebuttal.

Reviewers who changed their mind about a submission due to the rebuttal or to the other reviews were asked to consider adding a "post-rebuttal feedback" section to their review and explain why. (We requested not to make substantive changes to reviews outside this section, as the reviews have already been seen by the authors and so any changes need to be clearly identified and justified.)

All the submissions, including R1, were assigned discussions leads, whose job was to drive discussion, write the summaries, and ensure progress. We asked leads to make an honest effort to ensure that the opinions of non-heavy reviewers were adequately voiced and represented at the meeting. Non-heavy members were warmly encouraged to champion submissions that they believe should be accepted, and all reviewers were encouraged not to feel pressured to adopt a common denominator point of view, and not to hesitate to go against the majority. Reviewers were encouraged to reflect on each others' opinions, e.g., by considering previous work or confirming an opinion from an expert.

We asked the reviewers to stay positive when possible (particularly when it comes to out-of-the-box ideas) and to keep in mind that we should be looking for reasons to accept a paper rather than reject.

When reviewers were unable to reach consensus (yellow), the online discussion was expected to reconcile as many differences among the reviewers as possible, leaving only a few substantive differences for a focused PC meeting discussion. Namely, tagging yellow was not used as a way to procrastinate or reduce work, because it is impossible to discuss all R2 submissions in one day. The meeting was planned to be dedicated primarily to those submissions that actually require it, focusing on differences that the reviewers had already identified as important.

When making decisions, we requested reviewers to assume shepherding but not for adding new results. (All accepted papers were indeed assigned shepherds, responsible for making sure that revision expectations are met.) Of the R2 submissions, we pre-rejected 80, pre-accepted 37, and tagged 67 as yellow to discuss at the meeting.

During the online discussions, we recognized that about a dozen R2 submissions might not have reviews with enough expertise, so we urgently solicited additional reviews from relevant experts after the rebuttal period. In these cases, we emailed the authors and allowed them to rebut the additional review(s), copy-pasting their response as a comment in the HotCRP relevant page.

Program Committee Meeting The PC meeting took place between 8am–6pm, 12 April 2019, in the VMware campus in

Palo Alto, CA. The program co-chairs, submission co-chairs, and 60 heavy PC members attended the meeting in person, five called in, and one could not participate. The meeting consisted of a morning joint session (8am–12pm), a split session in two rooms (12:30pm–3pm), and an afternoon joint session (3:15pm–6pm), followed by a lively PC dinner.

The split session composition was determined with the help of a constraint solver as described in Section 6. The partition was completely disjoint, and no members transitioned between rooms while it took place. We discussed 12 green (preliminary accept) and 25 yellow (discuss) submissions in the morning joint session, and 7 green and 12 yellow submissions in the afternoon joint session. In the split session, one group discussed 8 green and 16 yellow submissions, and the other group discussed 10 green and 14 yellow submissions. We allocated 3 and 7 minutes discussion time for each green and yellow submissions, respectively.

Out of the 67 yellow submissions discussed, the PC accepted 34, which, together with the 37 preliminary accepts, resulted in a program of 71 papers, of which 2 are short. Accept decisions were reached by consensus, except in two cases that required a PC vote.

9 Best Paper Selection

The best paper award selection process proceeded in two phases. In the first phase, we combined several signals. One was an explicit ranking by reviewers marking papers worthy of consideration for best-paper; any paper marked for such consideration by two or more PC members was passed to the second phase. Additionally, we considered general review ranks and deliberations (both online and during the PC meeting), moving several additional top-ranking papers to the second phase. Last, we collected explicit nominations by PC members for the best paper award.

At the end of the first phase, we generated a short-list of eight papers. At this stage, we appointed a swat team of six PC members consisting of senior and experienced members of the systems research community. During a period of four weeks, the team read papers, and we deliberated each one separately for best-paper worthiness. Conflicted members were excluded from discussions of the relevant papers. We did not place a quota on the number of best-paper awards. Generally, the committee favored papers with original or surprising contribution, and/or ones that would spark interest and establish a new direction for follow on works.

At the end of the second stage, we elected three papers to receive best-paper awards for USENIX ATC '19.

Acknowledgments

The ATC '19 program is the result of the efforts of many. We thank the authors for submitting their work, and the committee members and external reviewers for working so hard to review the submissions. We are deeply indebted to our awesome submission co-chairs, Lalith Suresh and Gerd Zellweger, and also to Igor Smolyar, who helped whenever needed. We also thank the Lightning Talks co-chairs, Deniz Altinbuken and Aasheesh Kolli, and the Best of the Rest co-chairs, Amy Tai and Chia-Che Tsai. We thank Erez Zadok for managing submissions for which both program co-chairs were conflicted.

We thank Eddie Kohler for authoring and maintaining HotCRP, and for supporting us and promptly adding the features we needed. We thank the program co-chairs of ATC '18, Haryadi Gunawi and Benjamin Reed, for being responsive and providing lots of useful information. We are grateful to Emmett Witchel, who co-chaired ASPLOS '19, went through everything a few months before us, and served as a source of much needed knowledge and emotional support. We thank Sarita Adve for accurately documenting her excellent review process in ASPLOS '14 [1], which was quite helpful. We thank Emery Berger for suggesting the idea of test of time award for ATC, and Vijay Chidambaram for so nicely articulating the case for double blindness—much of the text in Section 2.2 originated from him. We also thank Or Hershkovitz for reviewing the math in Appendix A and for finding and elegantly fixing a bug.

We thank the USENIX staff for their outstanding conference management, and notably Casey Henderson, Hakim Weatherspoon, and Angela Demke Brown for their thoughtful advice and guidance; Angela and Hakim additionally reviewed this document (in very short notice), and they provided valuable and much appreciated feedback that helped us improve it.

Lastly, we thank VMware for sponsoring and hosting the PC meeting (and for paying for drinks at the PC dinner), and Sandra Barreto, Lori Blonn, and Sean Crotty for helping to organize the meeting.

References

[1] Sarita Adve. ASPLOS '14: Program chair's message. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages iv–ix, 2014. https://dl.acm.org/citation.cfm?id=2541940.

[2] Andrew Birrell, , and Emin Gün Sirer. Message from the 2013 USENIX Annual Technical Conference program co-chairs. In *ATC '13: USENIX Annual Technical Conference*, page vi, 2013. https://www.usenix.org/sites/default/files/atc13_message.pdf.

[3] Mario Drumond, Mark Sutherland, and Babak Falsafi. PC chair kit. https://github.com/mdrumond/pc-chair-kit.

[4] C. Le Goues, Y. Brun, S. Apel, E. Berger, S. Khurshid, and Y. Smaragdakis. Effectiveness of anonymization in double-blind review. *Communications of the ACM (CACM)*, 61(6):30–33, May 2018. http://doi.org/10.1145/3208157.

[5] Haryadi Gunawi and Benjamin Reed. Message from the 2018 USENIX Annual Technical Conference program co-chairs. https://www.usenix.org/sites/default/files/atc18_message.pdf, 2018.

[6] Gernot Heiser. Peer review: Anonymity should not be at the expense of transparency. https://microkerneldude.wordpress.com/2015/02/13/peer-review-anonymity-should-not-be-at-the-expense-of-transparency/, Feb 2015. Accessed: Jul 2019.

[7] Tyler Hunt. Fork of PC chair kit. https://github.com/tylershunt.

[8] Samir Khuller and Richard Matthew McCutchen. Assigning papers to reviewers. https://mattmccutchen.net/match/index.html, 2013.

[9] Feature request: ability to group topics. GitHub HotCRP issue https://github.com/kohler/hotcrp/issues/153, Dec 2018.

[10] Feature request: live-meeting-tracker for dual-track meetings. GitHub HotCRP issue https://github.com/kohler/hotcrp/issues/154, Dec 2018.

[11] Eddie Kohler. HotCRP source file mincostmaxflow.php. https://github.com/kohler/hotcrp/blob/master/lib/mincostmaxflow.php.

[12] Jeffrey C. Mogul. Policies for the SIGOPS hall of fame award. *SIGOPS Operating Systems Review*, 42(3):132–135, Apr 2008. https://doi.org/10.1145/1368506.1368525.

[13] Guidelines for the program chair of a SIGPLAN event. https://www.sigplan.org/Resources/Guidelines/ProChair/.

[14] Andrew Tomkins, Min Zhang, and William D. Heavlin. Reviewer bias in single- versus double-blind peer review. *Proceedings of the National Academy of Sciences (PNAS)*, 114(48):12708–12713, 2017. https://doi.org/10.1073/pnas.1707323114.

[15] SIGARCH/SIGPLAN/SIGOPS ASPLOS influential paper award. https://www.acm.org/sig-awards.

[16] Eurosys test-of-time award. http://www.eurosys.org/awards/tot-10-award.

[17] SIGOPS – the hall of fame award. https://www.sigops.org/awards/hof.

[18] USENIX test of time awards. https://www.usenix.org/conferences/test-of-time-awards.

[19] USENIX ATC '19 call for papers. https://www.usenix.org/conference/atc19/call-for-papers.

[20] FAST '19 call for papers. https://www.usenix.org/conference/fast19/call-for-papers.

[21] Wikipedia. Stable marriage problem. https://en.wikipedia.org/wiki/Stable_marriage_problem.

n	number of submissions submitted to the conference
n_2	number of submissions promoted to R2
m	an individual PC member
r	number of reviews written by m
r_1	number of reviews written by m in R1
r_2	number of reviews written by m in R2
c	number of submissions discussed at the PC meeting
d	reviewed by m and discussed at the PC meeting

Table 8: *Notation.*

Appendix A Submissions Discussed by Each Member at the Meeting

Let n denote the number of papers that have been submitted to the conference. Let n_2 denoted the total number of R2 submissions that have been promoted from R1. Let m denote one PC member, and assume that m has reviewed exactly r submissions out of the n. Further assume that the number of m's R1 and R2 reviews are r_1 and r_2, respectively ($r = r_1 + r_2$). Let c be the total number of submissions that have been *discussed* at the PC meeting, and let d denote how many of these c submissions have been reviewed by m ($d \leq r$). These notations are summarized in Table 8. Recall that Figure 2 shows that as n grows, d decreases, to the point that m has little to do at the PC meeting because d is small. The computation underlying Figure 2 assumes a typical setup for systems conferences where $n_2 = n/2$ (half of the submissions have been promoted to R2), $r_1 = \frac{2}{3} \cdot r$ and $r_2 = \frac{1}{3} \cdot r$ (two thirds of m's reviews are written during R1), and the number of discussed submissions is $c = 70$. With our assumptions, an intuitive approximation of d on average is

$$d \approx r_1 \cdot \frac{c}{n} + r_2 \cdot \frac{c}{n_2} = (r_1 + 2r_2) \cdot \frac{c}{n} = \frac{4rc}{3n} \qquad (1)$$

because (1) the probability that a single R1 submission that has been reviewed by m will be discussed at the meeting is

c/n, and, similarly, (2) the probability that a single R2 submission that has been reviewed by m will be discussed is approximately c/n_2, if disregarding the fact that the latter probability is in fact affected by the specific number of R1 submissions reviewed by m that have made it into R2. (For example, if all the submissions that m reviewed in R1 were promoted to R2, then the latter probability should actually be $\frac{c}{n_2-r_1}$, seeing that m cannot be assigned R2-submissions that she has already reviewed in R1.)

Figure 2, however, does not depict the approximation of d but rather computes it accurately, as follows. Let $p(n,c,r_1,k)$ denote the probability that exactly k of the r_1 submissions that m reviewed in R1 have been discussed at the meeting, then

$$p(n,c,r_1,k) \;=\; \binom{r_1}{k}\cdot\binom{n-r_1}{c-k}\div\binom{n}{c}. \qquad (2)$$

Thus, $e(n,c,r_1)$, which is the expected number of submissions that m reviewed in R1 and were discussed at the meeting, can (also) be computed with the following summation

$$e(n,c,r_1) \;=\; \sum_{k=0}^{r_1} p(n,c,r_1,k)\cdot k. \qquad (3)$$

Now, by using Equations 2–3 and the law of total probability, we can compute $e_2(n,c,r_1,r_2)$, which is the expected number of submissions that m reviewed in R2 and were discussed at the meeting, as follows

$$e_2(n,c,r_1,r_2) \;=\; \sum_{k=0}^{r_1} p(n,n_2,r_1,k)\cdot e(n_2-k,c,r_2). \qquad (4)$$

Notice that Equation 4 uses $p(n,\mathbf{n_2},r_1,k)$ instead of the earlier $p(n,\mathbf{c},r_1,k)$, because here the probability corresponds to the event that k of the r_1 submissions reviewed by m in R1 were promoted to R2. Using Equations 3–4, we conclude that

$$d \;=\; e(n,c,r_1)+e_2(n,c,r_1,r_2), \qquad (5)$$

which allows us to compute d accurately instead of approximating it. That said, in the range plotted in Figure 2, the difference between the real value of d (Equation 5) and its approximation (Equation 1) is always smaller than 0.52, which is reasonably close.