Introduction to the Special Section on ASPLOS 2024

ASPLOS 2024, the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, took place in San Diego from April 27 to May 1, 2024. It hosted 804 in-person participants, making it the largest ASPLOS to date. Among the participants, 635 attended the main conference (including 290 students) and 368 attended workshops/tutorials (including 157 students) with some overlap between the two groups. In addition, there were 31 remote attendees.

For the second year, ASPLOS'24 employed a review model with three submission deadlines—spring, summer, and fall—along with a major revision mechanism. Overall, ASPLOS'24 received 922 submissions, up from 598 in 2023 and 397 in 2022, corresponding to substantial year-over-year increases of more than 50% in each of the past two years. Table 1 summarizes the submission, acceptance, and major revision statistics for the three submission cycles. We accepted 194 papers, corresponding to an acceptance rate of 21.0%, with nearly 40% of the accepted papers having undergone a major revision.

Major revisions were introduced into multi-cycle ASPLOS as a way to increase acceptance rates by allowing authors to address a list of problems identified by reviewers that would otherwise have resulted in rejection. Authors can then resubmit their work to the subsequent review cycle, knowing that it will be accepted if these problems are resolved. Evidently, however, in 2024, the acceptance rate of multi-cycle ASPLOS, excluding major revisions, dropped considerably relative to the acceptance rates of past single-cycle ASPLOS conferences, from around 20% to 13%. Thus, in ASPLOS'24, major revisions did not increase acceptance rates, but rather compensated for the significant reduction in regular acceptances. One possible explanation for this phenomenon is that reviewers unconsciously use major revisions to raise the acceptance bar and improve submissions that might otherwise have been accepted without revision. Another explanation is that the dramatic increase in submission numbers was partially driven by lower-quality work, resulting in a higher rejection rate.

ASPLOS emphasizes multidisciplinary research, so submissions should ideally (though not necessarily) reflect the synergy of two or more of its three main disciplines: computer architecture, operating systems, and programming languages, broadly interpreted. We were curious about how ASPLOS papers are distributed across these disciplines and, in particular, about the extent to which ASPLOS is multidisciplinary. Traditional submission "topics" do not necessarily correspond to specific disciplines, so the relevant statistics have thus far been inaccurate or anecdotal. Consequently, for the first time at ASPLOS, we instructed authors, upon submission, to identify the primary discipline associated with their work and, if applicable, the secondary and tertiary disciplines.

ACM Reference Format:
Dan Tsafrir and Madanlal Musuvathi. 2026. Introduction to the Special Section on ASPLOS 2024. *ACM Trans. Comput. Syst.* 44, 3, Article 22 (August 2026), 4 pages. https://doi.org/10.1145/3815199



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 0734-2071/2026/08-ART22
https://doi.org/10.1145/3815199

Table 1. The Combined Submission, Acceptance, and Major Revision Statistics of ASPLOS'24

	submitted			*accepted*		
	R1	*R2*	*revision*	*regular*	*revision*	*total*
spring	173	87 (50%)	27 (16%)	28 (16.2%)	22 (12.7%)	50 (28.9%)
summer	409	179 (44%)	33 (8%)	54 (13.2%)	27 (6.6%)	81 (19.8%)
fall	340	183 (54%)	33 (10%)	39 (11.5%)	24 (7.1%)	63 (18.5%)
total	922	449 (49%)	93 (10%)	121 (13.1%)	73 (7.9%)	194 (21.0%)

Out of a total of 922 submissions, ASPLOS'24 accepted 194 papers, resulting in a 21.0% acceptance rate. R1 and R2 denote the first and second review rounds, respectively. The denominator for the percentages is the seasonal R1 submission number.

Table 2. The Primary Discipline of Submitted and Accepted ASPLOS'24 Papers as Reported by Authors

primary discipline	*submitted*	*accepted*	*discipline accept rate*
computer architecture	529 (57.4%)	108 (55.7%)	20.4%
operating systems	249 (27.0%)	41 (21.1%)	16.5%
programming languages	144 (15.6%)	45 (23.2%)	31.2%
total	922 (100.0%)	194 (100.0%)	21.0%

Numbers in parentheses denote percentages out of the total of that column, whereas percentages not in parentheses denote discipline-specific acceptance rates.

Table 2 reports the distribution of primary disciplines, showing that about 57% of ASPLOS'24 submissions are primarily associated with computer architecture, 27% with operating systems, and 16% with programming languages. Interestingly, the acceptance rate for computer architecture submissions (20.4%) is similar to the overall acceptance rate, while the corresponding rate is substantially lower for operating systems (16.5%) and markedly higher for programming languages (31.2%).

Table 3 summarizes the interdisciplinarity information provided by ASPLOS'24 authors, regardless of per-discipline rank. Specifically, the primary and secondary/tertiary disciplines (if available) are treated solely as indicators of interdisciplinarity, ignoring their relative importance within each submission. Overall, fewer than half of the papers are interdisciplinary. In particular, 46% and 45% of the submitted and accepted papers, respectively, are associated with more than one ASPLOS discipline. It should be noted that authors declaring their work as corresponding to more than one discipline does not necessarily mean that this is indeed the case. The actual interdisciplinarity rate may therefore be lower. (The program co-chairs of ASPLOS'25 asked reviewers to assess whether the authors' discipline declarations were reasonable, but it appears that this information is unavailable.)

ASPLOS'24 introduced several notable changes and new procedures, mostly in response to the aforementioned substantial increase in the number of submissions. These include: (1) significantly increasing the program committee size to over 220 members, more than twice that of the previous year; (2) foregoing synchronous PC meetings and instead making all decisions online; (3) overhauling the review assignment process by employing a constraint solver to optimize assignment "goodness" while accounting for multiple independent metrics, including textual similarity between submission contents and papers authored by reviewers, reviewer self-assessed expertise for a carefully selected subset of submissions, citations within submissions to reviewer papers, and author-declared best-matching topics; (4) introducing the new role of Program Vice Chairs; and (5) identifying a systematic issue in the review of quantum computing submissions, describing how we addressed it, and explaining how we believe it should be handled in the future. These issues and many others are described in detail in our "messages from the program chairs" across the four ASPLOS'24 volumes [1–4]. In addition, a SIGARCH trip report is available [5].

Table 3. ASPLOS'24 Interdisciplinarity Information Provided by Authors

		number of submissions					*percent of submissions*				
	discipline	*arch*	*os*	*pl*	*all*	*total*	*arch*	*os*	*pl*	*all*	*total*
submitted	computer arch.	343	217	126	14	700	37%	24%	14%	2%	76%
	operating systems	217	104	64	14	399	24%	11%	7%	2%	43%
	prog. languages	126	64	54	14	258	14%	7%	6%	2%	28%
	all three	14	14	14			2%	2%	2%		
	total (w/ duplicates)	700	399	258		1357	76%	43%	28%		147%
	total (w/o)					922					100%
	non-interdisciplinary (main diagonal)					501					54%
	interdisciplinary (lower/upper triangular)					421					46%
accepted	computer arch.	73	35	30	4	142	38%	18%	15%	2%	73%
	operating systems	35	13	18	4	70	18%	7%	9%	2%	36%
	prog. languages	30	18	21	4	73	15%	9%	11%	2%	38%
	all three	4	4	4			2%	2%	2%		
	total (w/ duplicates)	142	70	73		285	73%	36%	38%		147%
	total (w/o)					194					100%
	non-interdisciplinary (main diagonal)					107					55%
	interdisciplinary (lower/upper triangular)					87					45%

The 4x4 shaded matrices are symmetric: the main diagonal consists of non-interdisciplinary submissions, and the lower and upper triangular entries correspond to interdisciplinary submissions associated with either two disciplines or all three, as indicated by the row and column labels.

Selecting the best papers from the 194 papers accepted by ASPLOS'24 was nontrivial, effectively requiring us to run another mini-conference-style review process. We prepared a document, included in this issue, that describes the selection methodology we employed [6]. We believe that this methodology constitutes a fair process that upholds high standards of quality while emphasizing consensus and expert judgment; we hope it will prove useful to others. The process resulted in six awarded papers [7–12]. The authors of three of these papers chose to extend their work with additional material for this special section of ACM Transactions on Computer Systems [13–15]. The extended versions were re-reviewed and accepted by a subset of the original reviewers. We believe that readers will find them even more compelling.

Dan Tsafrir
Technion – Israel Institute of Technology, Haifa, Israel
email: dan@cs.technion.ac.il

Madan Musuvathi
Microsoft Research, Redmond, United States
email: madanm@microsoft.com

Program Co-Chairs

References

Documents Describing ASPLOS 2024

[1] Madan Musuvathi and Dan Tsafrir. 2024. ASPLOS 2024 Volume 1 (Spring Cycle): Message from the Program Chairs. DOI: https://doi.org/10.1145/3617232. (2024).

[2] Madan Musuvathi and Dan Tsafrir. 2024. ASPLOS 2024 Volume 2 (Summer Cycle): Message from the Program Chairs. DOI: https://doi.org/10.1145/3620665. (2024).

[3] Madan Musuvathi and Dan Tsafrir. 2024. ASPLOS 2024 Volume 3 (Fall Cycle): Message from the Program Chairs. DOI: https://doi.org/10.1145/3620666. (2024).
[4] Madan Musuvathi and Dan Tsafrir. 2025. ASPLOS 2024 Volume 4 (Accepted Major Revisions of Fall Cycle): Message from the Program Co-Chairs. DOI: https://doi.org/10.1145/3622781. (2025).
[5] Akanksha Jain and Daniel Wong. 2024. ASPLOS 2024 Trip Report. Retrieved from https://www.sigarch.org/asplos-2024-trip-report. (May 2024).

How the Best Papers Were Selected

[6] Dan Tsafrir and Madan Musuvathi. 2026. Selecting ASPLOS best papers: A guide. *ACM Transactions on Computer Systems (TOCS)* 44, 3, Article 21 (2026), 1–7. DOI: https://doi.org/10.1145/3815200

ASPLOS 2024 Best Papers

[7] Chang Chen, Xiuhong Li, Qianchao Zhu, Jiangfei Duan, Peng Sun, Xingcheng Zhang, and Chao Yang. 2024. Centauri: Enabling efficient scheduling for communication-computation overlap in large model training via communication partitioning. In *ASPLOS'24: International Conference on Architectural Support for Programming Languages and Operating Systems*. Vol. 3, ACM, New York, NY, USA, 178–191. DOI: https://doi.org/10.1145/3620666.3651379
[8] Tianao Ge, Tong Zhang, and Hongyuan Liu. 2024. ngAP: Non-blocking large-scale automata processing on GPUs. In *ASPLOS'24: International Conference on Architectural Support for Programming Languages and Operating Systems*. Vol. 1, ACM, New York, NY, USA, 268–285. DOI: https://doi.org/10.1145/3617232.3624848
[9] Bhargav Reddy Godala, Sankara Prasad Ramesh, Gilles A. Pokam, Jared Stark, Andre Seznec, Dean Tullsen, and David I. August. 2024. PDIP: Priority directed instruction prefetching. In *ASPLOS'24: International Conference on Architectural Support for Programming Languages and Operating Systems*. Vol. 2, ACM, New York, NY, USA, 846–861. DOI: https://doi.org/10.1145/3620665.3640394
[10] Hao Ling, Heqing Huang, Chengpeng Wang, Yuandao Cai, and Charles Zhang. 2024. GIANTSAN: Efficient memory sanitization with segment folding. In *ASPLOS'24: International Conference on Architectural Support for Programming Languages and Operating Systems*. Vol. 2, ACM, New York, NY, USA, 433–449. DOI: https://doi.org/10.1145/3620665.3640391
[11] Samuel Thomas and James Bornholt. 2024. Automatic generation of vectorizing compilers for customizable digital signal processors. In *ASPLOS'24: International Conference on Architectural Support for Programming Languages and Operating Systems*. Vol. 1, ACM, New York, NY, USA, 19–34. DOI: https://doi.org/10.1145/3617232.3624873
[12] Hünkar Can Tunç, Ameya Prashant Deshmukh, Berk Cirisci, Constantin Enea, and Andreas Pavlogiannis. 2024. CSSTs: A dynamic data structure for partial orders in concurrent execution analysis. In *ASPLOS'24: International Conference on Architectural Support for Programming Languages and Operating Systems*. Vol. 3, ACM, New York, NY, USA, 223–238. DOI: https://doi.org/10.1145/3620666.3651358

Extended TOCS Versions

[13] Tianao Ge, Tong Zhang, and Hongyuan Liu. 2026. Towards scalable and non-blocking automata processing on GPUs with ngAP. *ACM Transactions on Computer Systems (TOCS)* 44, 3, Article 23 (2026), 1–33. DOI: https://doi.org/10.1145/3748646
[14] Hao Ling, Heqing Huang, Chengpeng Wang, Yuandao Cai, and Charles Zhang. 2026. GiantSan: Efficient operation-level memory sanitization with segment folding. *ACM Transactions on Computer Systems (TOCS)* 44, 3, Article 24 (2026), 1–30. DOI: https://doi.org/10.1145/3742426
[15] Hünkar Can Tunç, Yifan Dong, Ameya Prashant Deshmukh, Berk Cirisci, Constantin Enea, and Andreas Pavlogiannis. 2026. Efficient dynamic concurrency analysis with collective sparse segment trees. *ACM Transactions on Computer Systems (TOCS)* 44, 3, Article 25 (2026), 1–31. DOI: https://doi.org/10.1145/3773085