References

[1] M. K. Aguilera, K. Keeton, S. Novakovic, and S. Singhal. Designing far memory data structures: Think outside the box. In *Workshop on Hot Topics in Operating Systems*, HotOS'19, pages 120–126, 2019.

[2] E. Amaro, C. Branner-Augmon, Z. Luo, A. Ousterhout, M. K. Aguilera, A. Panda, S. Ratnasamy, and S. Shenker. Can far memory improve job throughput? In *European Conference on Computer Systems*, EUROSYS'17, pages 1–16, 2020.

[3] ARM. Neon programmer guides for armv8-a, Accessed 2020/06/10. https://developer.arm.com/architectures/instruction-sets/simd-isas/neon.

[4] P. Bailis. Communication Costs in Real World Networks, Accessed 2020/06/10. http://www.bailis.org/blog/communication-costs-in-real-world-networks/.

[5] L. Barroso, M. Marty, D. Patterson, and P. Ranganathan. Attack of the killer microseconds. *Communications of the ACM*, 60(4):48–54, 2017.

[6] B. N. Bershad, S. Savage, P. Pardyak, E. G. Sirer, M. E. Fiuczynski, D. Becker, C. Chambers, and S. Eggers. Extensibility safety and performance in the spin operating system. In *ACM Symposium on Operating Systems Principles*, SOSP'95, pages 267–283, 1995.

[7] The ccix consortium, Accessed 2020/09/24. https://www.ccixconsortium.com/.

[8] Compute express link, Accessed 2020/09/24. https://www.computeexpresslink.org/.

[9] J. Do, S. Sengupta, and S. Swanson. Programmable solid-state storage in future cloud datacenters. *Communications of the ACM*, 62(6):54–62, 2019.

[10] A. Dragojević, D. Narayanan, O. Hodson, and M. Castro. Farm: Fast remote memory. In *Symposium on Networked Systems Design and Implementation*, NSDI'14, pages 401–414, 2014.

[11] D. Firestone, A. Putnam, S. Mundkur, D. Chiou, A. Dabagh, M. Andrewartha, H. Angepat, V. Bhanu, A. Caulfield, E. Chung, et al. Azure accelerated networking: Smartnics in the public cloud. In *Symposium on Networked Systems Design and Implementation*, NSDI'18, pages 51–66, 2018.

[12] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker. Network requirements for resource disaggregation. In *Symposium on Operating Systems Design and Implementation*, OSDI'16, pages 249–264, 2016.

[13] The gen-z consortium, Accessed 2020/09/24. https://genzconsortium.org/.

[14] E. Gershuni, N. Amit, A. Gurfinkel, N. Narodytska, J. A. Navas, N. Rinetzky, L. Ryzhyk, and S. Sagiv. Simple and precise static analysis of untrusted linux kernel extensions. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI'19, 2019.

[15] J. Gu, Y. Lee, Y. Zhang, M. Chowdhury, and K. G. Shin. Efficient memory disaggregation with infiniswap. In *Symposium on Networked Systems Design and Implementation*, NSDI'17, pages 649–667, 2017.

[16] K. Hamidouche, A. Venkatesh, A. A. Awan, H. Subramoni, C.-H. Chu, and D. K. Panda. Exploiting gpudirect rdma in designing high performance openshmem for nvidia gpu clusters. In *IEEE Transactions on Parallel and Distributed Systems*, TPDS'15, pages 78–87, 2015.

[17] G. C. Hunt and J. R. Larus. Singularity: rethinking the software stack. *ACM SIGOPS Operating Systems Review*, 41(2):37–49, 2007.

[18] R. Imaoka. Using ping to test AWS VPC network latency within a single region, Accessed 2020/06/10. https://richardimaoka.github.io/blog/network-latency-analysis-with-ping-aws/.

[19] K. Kaffes, T. Chong, J. T. Humphries, A. Belay, D. Mazières, and C. Kozyrakis. Shinjuku: Preemptive scheduling for μsecond-scale tail latency. In *Symposium on Networked Systems Design and Implementation*, NSDI'19, pages 345–360, 2019.

[20] A. Kalia, M. Kaminsky, and D. G. Andersen. Using rdma efficiently for key-value services. In *ACM Special Interest Group on Data Communications*, SIGCOMM'14, pages 295–306, 2014.

[21] A. Kalia, M. Kaminsky, and D. G. Andersen. Fasst: Fast, scalable and simple distributed transactions with two-sided (rdma) datagram rpcs. In *Symposium on Operating Systems Design and Implementation*, OSDI'16, pages 185–201, 2016.

[22] A. Li, S. L. Song, J. Chen, X. Liu, N. Tallent, and K. Barker. Tartan: evaluating modern gpu interconnect via a multi-gpu benchmark suite. In *2018 IEEE International Symposium on Workload Characterization (IISWC)*, pages 191–202. IEEE, 2018.

[23] B. Li, Z. Ruan, W. Xiao, Y. Lu, Y. Xiong, A. Putnam, E. Chen, and L. Zhang. Kv-direct: High-performance in-memory key-value store with programmable nic. In *ACM Symposium on Operating Systems Principles*, SOSP'17, pages 137–152, 2017.

[24] S. Li, H. Lim, V. W. Lee, J. H. Ahn, A. Kalia, M. Kaminsky, D. G. Andersen, O. Seongil, S. Lee, and P. Dubey. Architecting to achieve a billion requests per second throughput on a single key-value store server platform. In *International Symposium on Computer Architecture*, ISCA'15, pages 476–488, 2015.

[25] M. Liu, T. Cui, H. Schuh, A. Krishnamurthy, S. Peter, and K. Gupta. Offloading distributed applications onto smartnics using ipipe. In *ACM Special Interest Group on Data Communications*, SIGCOMM'19, 2019.

[26] C. Mitchell, Y. Geng, and J. Li. Using one-sided rdma reads to build a fast, cpu-efficient key-value store. In *USENIX Annual Technical Conference*, ATC'13, 2013.

[27] R. Mittal, V. T. Lam, N. Dukkipati, E. R. Blem, H. M. G. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats. Timely: Rtt-based congestion control for the datacenter. In *ACM Special Interest Group on Data Communications*, SIGCOMM'15, 2015.

[28] Netronome. Nfp-6000 intelligent ethernet controller family, Accessed 2020/06/10. https://www.netronome.com/static/app/img/products/silicon-solutions/PB_NFP6000.pdf.

[29] R. Neugebauer, G. Antichi, J. F. Zazo, Y. Audzevich, S. López-Buedo, and A. W. Moore. Understanding pcie performance for end host networking. In *ACM Special Interest Group on Data Communications*, SIGCOMM'18, pages 327–341, 2018.

[30] A. Panda, S. Han, K. Jang, M. Walls, S. Ratnasamy, and S. Shenker. Netbricks: Taking the v out of nfv. In *Symposium on Operating Systems Design and Implementation*, OSDI'16, pages 203–216, 2016.

[31] Pci-sig specifications library, Accessed 2020/09/24. https://pcisig.com/specifications.

[32] D. A. Popescu. *Latency-driven performance in data center*. PhD thesis, University of Cambridge, 2019.

[33] Y. Shan, Y. Huang, Y. Chen, and Y. Zhang. Legoos: A disseminated, distributed os for hardware resource disaggregation. In *Symposium on Operating Systems Design and Implementation*, OSDI'18, pages 69–87, 2018.

[34] J. Shi, Y. Yao, R. Chen, H. Chen, and F. Li. Fast and concurrent rdf queries with rdma-based distributed graph exploration. In *Symposium on Operating Systems Design and Implementation*, OSDI'16, 2016.

[35] A. Shpiner, E. Zahavi, V. Zdornov, T. Anker, and M. Kadosh. Unlocking credit loop deadlocks. 2016.

[36] D. Sidler, Z. Wang, M. Chiosa, A. Kulkarni, and G. Alonso. Strom: smart remote memory. In *European Conference on Computer Systems*, EUROSYS'20, pages 1–16, 2020.

[37] G. Singh, L. Chelini, S. Corda, A. J. Awan, S. Stuijk, R. Jordans, H. Corporaal, and A.-J. Boonstra. Near-memory computing: Past, present, and future. *Microprocess. Microsystems*, 71, 2019.

[38] M. Technologies. Mellanox innova-2 flex open programmable smartnic, Accessed 2020/06/10. https://www.mellanox.com/sites/default/files/doc-2020/pb-innova-2-flex.pdf.

[39] M. Technologies. Nvidia mellanox bluefield-1 smartnic, Accessed 2020/06/10. https://www.mellanox.com/files/doc-2020/pb-bluefield-smart-nic.pdf.

[40] M. Technologies. Nvidia mellanox bluefield-2 smartnic, Accessed 2020/06/10. https://www.mellanox.com/files/doc-2020/pb-bluefield-2-smart-nic-eth.pdf.

[41] M. Technologies. Rdma aware networks programming user manual, Accessed 2020/06/10. https://www.mellanox.com/related-docs/prod_software/RDMA_Aware_Programming_user_manual.pdf.

[42] L. A. Torrey, J. Coleman, and B. P. Miller. A comparison of interactivity in the linux 2.6 scheduler and an mlfq scheduler. *Software - Practice and Experience*, 37:347–364, 2007.

[43] P. R. . A. Vahdat. Plotting a Course to a Continued Moore's Law - Keynote, Accessed 2020/06/10. https://youtu.be/6wq6g_vi6yw.

[44] K. Vipin and S. A. Fahmy. Fpga dynamic and partial reconfiguration: a survey of architectures, methods, and applications. *ACM Computing Surveys (CSUR)*, 51(4):1–39, 2018.

[45] H. Wang, S. Potluri, D. Bureddy, C. Rosales, and D. K. Panda. Gpu-aware mpi on rdma-enabled clusters: Design, implementation and evaluation. *IEEE Transactions on Parallel and Distributed Systems*, 25:2595–2605, 2014.

[46] Xilinx. Xilinx alveo u280, Accessed 2020/06/10. https://www.xilinx.com/publications/product-briefs/alveo-u280-product-brief.pdf.

[47] J. Xue, Y. Miao, C. Chen, M. Wu, L. Zhang, and L. Zhou. Fast distributed deep learning over rdma. In *European Conference on Computer Systems*, EUROSYS'19, 2019.

[48] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang. Congestion control for large-scale rdma deployments. In *ACM Special Interest Group on Data Communications*, SIGCOMM'15, 2015.