导致分析不精确.新的分析框架应该融合程序级分析与系统级分析,提高时间分析的精确性;

GPU 调度和资源管理问题:科技巨头 NVIDIA 公司不公开其 GPU 调度逻辑的详细资料,阻碍了对 NVIDIA GPU 开展实时调度研究和实验.虽然通过黑盒实验的方法可以获得很多不公开的调度规则,但是并不能确定这份调度规则清单是否足够完备,在新架构的 GPU 上是否依然有效.对 SoC 平台上 CPU 和 GPU 访问共享内存的分时隔离技术的研究已经取得了很大的进展,但是 CPU 与 GPU 之间显式或隐式的同步仍然会导致时间不确定性问题.由于 AMD 公司对其 GPU 技术细节的曝露要开放许多,并提供开源驱动 GPU Open[119],因此,一个可行的研究方向是以 AMD GPU+OpenCL[120]为平台来研究 GPU 实时调度[121]和资源管理技术,并研发用于实时 DNN 计算的基础软件.此外,前面综述过的调度或资源管理优化的研究工作存在技术路线不够系统化的问题,可以从 GPU 程序建模分析出发,结合系统的调度和资源分配,综合研究实时性能优化技术;

面向实时系统的网络加速器协同设计问题:无论是通用还是专用网络加速器仅能在一定程度上改善网络性能,并难以设计普适性的网络加速器结构.DNN 和网络加速器的协同设计可以提高两者的契合度,从而设计出性能特征高度匹配的网络与网络加速器整体解决方案,并降低硬件成本.不过,这个方向的研究主要集中在提高系统的平均性能,还未建立起满足实时系统要求的协同设计理论、性能建模与分析方法.考虑到神经网络加速器在未来必将广泛应用于安全攸关领域,面向实时应用的协同设计理论是一个非常有意义的研究方向;

智能实时嵌入式系统可更新问题:传统的实时嵌入式系统基于量体裁衣的方式设计程序,而很少考虑应用或系统更新之后可能会带来违背时间约束的问题(第 2.1 节中关键问题(3)).Wang 教授在文献[46]中指出了 CPS 安全攸关系统安全可更新问题和解决该问题的必要性、理论方向以及技术路线,同理,在 AI 赋能的实时嵌入式系统中,DNN 模型也会不断地更新迭代,那么如何保证模型更新之后的人工智能应用仍然能够满足原初设计的实时约束,将会是一个更具挑战性的理论问题,解决该问题无疑将大大促进人工智能实时嵌入式系统的发展.

References:

[1] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press, 2016.

[2] Molnar C. Interpretable machine learning: A guide for making black box models explainable. 2019. https://christophm.github.io/

[3] Zheng ZY, Gu SY. TensorFlow: A Practical Google Deep Learning Framework. Beijing: Publishing House of Electronic Industry, 2017 (in Chinese).

[4] Aceto L, Ingólfsdóttir A, Larsen KG, Srba J. Reactive Systems: Modelling, Specification and Verification. New York: Cambridge University Press, 2007.

[5] Seshia SA, Sadigh D. Towards verified artificial intelligence. CoRR, 2016, abs/1606.0.

[6] Sun X, Khedr H, Shoukry Y. Formal verification of neural network controlled autonomous systems. In: Proc. of the 22nd Int’l Conf. on Hybrid Systems: Computation and Control (HSCC). ACM, 2019. 147−156.

[7] Seshia SA, Desai A, Dreossi T, Fremont DJ, Ghosh S, Kim E, Shivakumar S, Vazquez-Chanlatte M, Yue X. Formal specification for deep neural networks. In: Proc. of the 16th Int’l Symp. on Automated Technology for Verification and Analysis (ATVA). Springer-Verlag, 2018. 20−34.

[8] Tuncali CE, Fainekos G, Ito H, Kapinski J. Simulation-Based adversarial test generation for autonomous vehicles with machine learning components. In: Proc. of the 2018 IEEE Intelligent Vehicles Symp. IEEE, 2018. 1555−1562.

[9] Dreossi T, Donzé A, Seshia SA. Compositional falsification of cyber-physical systems with machine learning components. In: Proc. of the 9th Int’l Symp. on NASA Formal Methods (NFM). Springer-Verlag, 2017: 357−372.

[10] Dreossi T, Ghosh S, Yue X, Keutzer K, Sangiovanni-Vincentelli AL, Seshia SA. Counterexample-Guided data augmentation. In: Proc. of the 27th Int’l Joint Conf. on Artificial Intelligence (IJCAI). 2018. 2071−2078.

[11] Pulina L, Tacchella A. An abstraction-refinement approach to verification of artificial neural networks. In: Proc. of the 22nd Int’l Conf. on Computer Aided Verification (CAV). Springer-Verlag, 2010. 243−257.

[12] Katz G, Barrett CW, Dill DL, Julian K, Kochenderfer MJ. Reluplex: An efficient smt solver for verifying deep neural networks. In: Proc. of the 29th Int’l Conf. on Computer Aided Verification (CAV). Springer-Verlag, 2017. 97−117.

[13] Singh G, Gehr T, Mirman M, Püschel M, Vechev MT. Fast and effective robustness certification. In: Proc. of the Advances in Neural Information Processing Systems 31: Annual Conf. on Neural Information Processing Systems (NeurIPS). Springer-Verlag, 2018. 10825−10836.

[14] PRECISE center for safe AI. https://precise.seas.upenn.edu/safe-autonomy

[15] Amert T, Otterness N, Yang M, Anderson JH, Smith FD. GPU scheduling on the nvidia tx2: Hidden details revealed. In: Proc. of the 2017 IEEE Real-Time Systems Symp. (RTSS). IEEE, 2017. 104–115.

[16] Yang M, Wang S, Bakita J, Vu T, Smith FD, Anderson JH, Frahm JM. Re-Thinking CNN frameworks for time-sensitive autonomous-driving applications: Addressing an industrial challenge. In: Proc. of the 25th IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS). IEEE, 2019. 305–317.

[17] Han R, Zhang F, Chen LY, Zhan J. Work-in-Progress: Maximizing model accuracy in real-time and iterative machine learning. In: Proc. of the Real-Time Systems Symp. 2018. 351–353.

[18] Alcaide S, Kosmidis L, Hernandez C, Abella J. High-Integrity GPU designs for critical real-time automotive systems. In: Proc. of the 2019 Design, Automation & Test in Europe Conf. & Exhibition (DATE). 2019. 824–829.

[19] Bateni S, Liu C. ApNet: Approximation-aware real-time neural network. In: Proc. of the Real-Time Systems Symp. (RTSS). IEEE, 2019. 67–79.

[20] Zhou H, Bateni S, Liu C. S^3DNN: Supervised streaming and scheduling for GPU-accelerated real-time dnn workloads. In: Proc. of the IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS). IEEE, 2018. 190–201.

[21] Yang M, Otterness N, Amert T, Bakita J, Anderson JH, Smith FD. Avoiding pitfalls when using nvidia GPUS for real-time tasks in autonomous systems. In: Proc. of the 30th Euromicro Conf. on Real-Time Systems (ECRTS). Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2018. 1–21.

[22] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 1943,5(4):115–133.

[23] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. In: Proc. of the Psychological Review. 1958. 65–386.

[24] Minsky M, Papert S. Perceptrons—An Introduction to Computational Geometry. MIT Press, 1987.

[25] Sejnowski TT. The Deep Learning Revolution. MIT Press, 2018.

[26] Hinton GE. Learning distributed representations of concepts. In: Proc. of the 8th Annual Conf. of the Cognitive Science Society. Oxford University Press, 1986. 112.

[27] Learning representation by back-propagation errors. Nature, 1986,323:533–536.

[28] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. Proc. of the Advances in Neural Information Processing Systems 25. Curran Associates, Inc., 2012. 1097–1105.

[29] Zhou Z. Machine Learning. Beijing: Tsinghua University Press, 2016 (in Chinese).

[30] Nielsen M. Neural Networks and Deep Learning. Determination Press, 2015.

[31] TensorFlow. https://www.tensorflow.org/

[32] PyTorch. https://pytorch.org/

[33] Caffe. http://caffe.berkeleyvision.org/

[34] Peng JT, Lin J, Bai XL. In-Depth Understanding of Tensorflow Architecture Design and Implementation Principles. Beijing: Posts & Telecom Press, 2018 (in Chinese).

[35] Cook S. CUDA programming: A Developer's Guide to Parallel Computing with GPUS. San Francisco: Morgan Kaufmann Publishers Inc., 2013.

[36] CUDA zone. https://developer.nvidia.com/cuda-zone

[37] Meet jetson, the platform for ai at the edge. https://developer.nvidia.com/embedded-computing

[38] Chen GL, Sheng M, Yang G. A survey of hardware-accelerated neural networks. Journal of Computer Research and Development, 2019,56(2):240–253 (in Chinese with English abstract).

[39] Farabet C, Martini B, Corda B, Akselrod P, Culurciello E, LeCun Y. NeuFlow: A runtime reconfigurable dataflow processor for vision. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2011. 109–116.

[40] Zhang C, Prasanna VK. Frequency domain acceleration of convolutional neural networks on CPU-fpga shared memory system. In: Proc. of the ACM/SIGDA Int'l Symp. on Field-Programmable Gate Arrays (FPGA). ACM, 2017. 35–44.

[41] Gao M, Pu J, Yang X, Horowitz M, Kozyrakis C. TETRIS: Scalable and efficient neural network acceleration with 3D memory. In: Proc. of the Int'l Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS). ACM, 2017. 751–764.

[42] Kim D, Kung J, Chai SM, Yalamanchili S, Mukhopadhyay S. Neurocube: A programmable digital neuromorphic architecture with high-density 3D memory. In: Proc. of the 43rd ACM/IEEE Annual Int'l Symp. on Computer Architecture (ISCA). IEEE Computer Society, 2016. 380–392.

[43] Shafiee A, Nag A, Muralimanohar N, Balasubramonian R, Strachan JP, Hu M, Williams RS, Srikumar V. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In: Proc. of the 43rd ACM/IEEE Annual Int'l Symp. on Computer Architecture (ISCA). IEEE Computer Society, 2016. 14–26.

[44] Xu H, Mueller F, Carolina N. Work-in-Progress: Making machine learning real-time predictable. In: Proc. of the 2018 IEEE Real-Time Systems Symp. (RTSS). IEEE, 2018. 157–160.

[45] Kim H, Nam H, Jung W, Lee J. Performance analysis of CNN frameworks for GPUS. In: Proc. of the IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS). IEEE, 2017. 55–64.

[46] Wang Y. Towards customizable CPS: Composability, efficiency and predictability. In: Duan Z, Ong L, eds. Proc. of the 19th Int'l Conf. on Formal Engineering Methods (ICFEM). Vol.10610. Xi'an: Springer-Verlag, 2017. 3–15.

[47] Abdullah J, Dai G, Yi W. Worst-Case cause-effect reaction latency in systems with non-blocking communication. In: Proc. of the 2019 Design, Automation & Test in Europe Conf. & Exhibition (DATE). 2019. 1625–1630.

[48] Wilhelm R, Engblom J, Ermedahl A, Holsti N, Thesing S, Whalley DB, Bernat G, Ferdinand C, Heckmann R, Mitra T, Mueller F, Puaut I, Puschner PP, Staschulat J, Stenström P. The worst-case execution-time problem—Overview of methods and survey of tools. ACM Transactions on Computer Systems, 2008,7(3):36:1–36:53.

[49] Davis RI, Burns A. A survey of hard real-time scheduling for multiprocessor systems. ACM Computing Surveys, 2011,43(4): 35:1–35:44.

[50] Hestness J, Keckler SW, Wood DA. A comparative analysis of microarchitecture effects on CPU and gpu memory system behavior. In: Proc. of the 2014 IEEE Int'l Symp. on Workload Characterization (IISWC). IEEE, 2014. 150–160.

[51] Posluszny D. Avoiding pitfalls when using nvidia GPUS for real-time tasks in autonomous systems. In: Proc. of the 30th Euromicro Conf. on Real-Time Systems (ECRTS). IEEE, 2018. 1–21.

[52] Reineke J, Wilhelm R. Impact of resource sharing on performance and performance prediction. In: Proc. of the Design, Automation & Test in Europe Conf. (DATE). European Design and Automation Association, 2014. 1–2.

[53] Capodieci N, Cavicchioli R, Bertogna M, Paramakuru A. Deadline-Based scheduling for GPU with preemption support. In: Proc. of the 2018 IEEE Real-Time Systems Symp. (RTSS). IEEE, 2018. 119–130.

[54] Forsberg B, Marongiu A, Benini L. GPUguard: Towards supporting a predictable execution model for heterogeneous SoC. In: Proc. of the 2017 Design, Automation and Test in Europe (DATE). 2017. 318–321.

[55] Bavoil L. SetStablePowerState.exe: Disabling GPU boost on windows 10 for more deterministic timestamp queries on nvidia GPUS. 2016. https://developer.nvidia.com

[56] Shams S, Platania R, Lee K, Park SJ. Evaluation of deep learning frameworks over different HPC architectures. In: Proc. of the Int'l Conf. on Distributed Computing Systems. IEEE, 2017. 1389–1396.

[57] Mojumder SA, Louis MS, Sun Y, Ziabari AK, Abellán JL, Kim J, Kaeli D, Joshi A. Profiling DNN workloads on a volta-based DGX-1 system. In: Proc. of the 2018 IEEE Int'l Symp. on Workload Characterization (IISWC). 2018. 122–133.

[58] Stephenson M, Sastry Hari SK, Lee Y, Ebrahimi E, Johnson DR, Nellans D, O'Connor M, Keckler SW. Flexible software profiling of GPU architectures. ACM SIGARCH Computer Architecture News, 2015,43(3):185–197.

[59] Shen D, Song SL, Li A, Liu X. CUDAAdvisor: LLVM-based runtime profiling for modern GPUS. 2018. 214–227.

[60] Farooqui N, Kerr A, Eisenhauer G, Schwan K, Yalamanchili S. Lynx: A dynamic instrumentation system for data-parallel applications on GPGPU architectures. In: Proc. of the IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS). IEEE, 2012. 58–67.

[61] Qi H, Sparks ER, Talwalkar A. Paleo: A performance model for deep neural networks. In: Proc. of the ICLR. 2017. 1–10.

[62] Dong S, Gong X, Sun Y, Baruah T, Kaeli D. Characterizing the microarchitectural implications of a convolutional neural network (CNN) execution on GPUS. 2018. 96–106.

[63] Madougou S, Varbanescu AL, De Laat C, Van Nieuwpoort R. A tool for bottleneck analysis and performance prediction for GPU-accelerated applications. In: Proc. of the 2016 IEEE 30th Int'l Parallel and Distributed Processing Symp. (IPDPS). IEEE, 2016. 641–652.

[64] Ali W, Yun H. Protecting real-time GPU kernels on integrated CPU-GPU SoC platforms. In: Proc. of the 30th Euromicro Conf. on Real-Time Systems (ECRTS). Vol.106. Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2018. 19:1–19:22.

[65] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <0.5MB model size. 2016. 1−13.

[66] Wang Y, Li H, Li X. Real-Time meets approximate computing: An elastic CNN inference accelerator with adaptive trade-off between qos and qor. In: Proc. of the 54th Annual Design Automation Conf. 2017. 2017. 33:1−33:6.

[67] Sainath TN, Kingsbury B, Sindhwani V, Arisoy E, Ramabhadran B. Low-Rank matrix factorization for deep neural network training with high-dimensional output targets. In: Proc. of the IEEE Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2013. 6655−6659.

[68] Lebedev V, Ganin Y, Rakhuba M, Oseledets I, Lempitsky V. Speeding-Up convolutional neural networks using fine-tuned cp-decomposition. 2014. 1−11.

[69] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions. 2014.

[70] Zhang X, Zou J, Ming X, He K, Sun J. Efficient and accurate approximations of nonlinear convolutional networks. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2015. 1984−1992.

[71] Zhang X, Zou J, He K, Sun J. Accelerating very deep convolutional networks for classification and detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2016,38(10):1943−1955.

[72] Denil M, Shakibi B, Dinh L, Ranzato M, De Freitas N. Predicting parameters in deep learning. 2013. 1−9.

[73] Han S, Mao H, Dally WJ. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. 2015. 1−14.

[74] Han S, Pool J, Tran J, Dally WJ. Learning both weights and connections for efficient neural networks. 2015. 1−9.

[75] Courbariaux M, Hubara I, Soudry D, El-Yaniv R, Bengio Y. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1. 2016.

[76] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: Efficient convolutional neural networks for mobile vision applications. 2017.

[77] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proc. of the 30th IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017. 1800−1807.

[78] Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2018. 6848−6856.

[79] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted residuals and linear bottlenecks. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2018. 4510−4520.

[80] Ma N, Zhang X, Zheng HT, Sun J. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In: Proc. of the 15th European Conf. on Computer Vision (ECCV). LNCS 11218. Springer-Verlag, 2018. 122−138.

[81] Gholami A, Kwon K, Wu B, Tai Z, Yue X, Jin P, Zhao S, Keutzer K, Berkeley UC. SqueezeNext: Hardware-aware neural network design. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR). IEEE Computer Society, 2018.

[82] Yao Q, Wang M, Chen Y, Dai W, Yi QH, Yu FL, Wei WT, Qiang Y, Yang Y. Taking human out of learning applications: A survey on automated machine learning. 2018. 1−26.

[83] Zoph B, Le Q V. Neural architecture search with reinforcement learning. 2016. 1−16.

[84] Tan M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, Le QV. MnasNet: Platform-aware neural architecture search for mobile. 2018.

[85] Xu H, Mueller F. Hardware for machine learning: Challenges and opportunities. In: Proc. of the Real-Time Systems Symp. IEEE, 2019. 157−160.

[86] Berezovskyi K, Bletsas K, Andersson B. Makespan computation for GPU threads running on a single streaming multiprocessor. In: Proc. of the 24th Euromicro Conf. on Real-Time Systems (ECRTS). IEEE, 2012. 277−286.

[87] Berezovskyi K, Bletsas K, Petters SM. Faster makespan estimation for GPU threads on a single streaming multiprocessor. In: Proc. of the 2013 IEEE 18th Conf. on Emerging Technologies & Factory Automation (ETFA). IEEE, 2013. 1−8.

[88] Berezovskyi K, Santinelli L, Bletsas K, Tovar E. WCET measurement-based and extreme value theory characterisation of CUDA kernels. In: Proc. of the 22nd Int’l Conf. on Real-Time Networks and Systems (RTNS). ACM, 2014. 279.

[89] Betts A, Donaldson A. Estimating the wcet of GPU-accelerated applications using hybrid analysis. In: Proc. of the Euromicro Conf. on Real-Time Systems. IEEE, 2013. 193−202.

[90] Punniyamurthy K, Boroujerdian B, Gerstlauer A. GATSim: Abstract timing simulation of GPUS. In: Proc. of the Design, Automation & Test in Europe Conf. (DATE). IEEE, 2017. 43−48.

[91] GPGPU-Sim. http://www.gpgpu-sim.org/

[92] Bakhoda A, Yuan GL, Fung WWL, Wong H, Aamodt TM. Analyzing CUDA workloads using a detailed gpu simulator. In: Proc. of the IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS). IEEE, 2009. 163−174.

[93] Wang X, Zhang W. Cache locking vs. partitioning for real-time computing on integrated CPU-GPU processors. In: Proc. of the 35th IEEE Int'l Performance Computing and Communications Conf. (IPCCC). IEEE, 2016. 1−8.

[94] Picchi J, Zhang W. Impact of l2 cache locking on GPU performance. In: Proc. of the SoutheastCon 2015. IEEE, 2015. 1−4.

[95] Huangfu Y, Zhang W. Warp-Based load/store reordering to improve GPU data cache time predictability and performance. In: Proc. of the 19th IEEE Int'l Symp. on Real-Time Distributed Computing (ISORC). IEEE, 2016. 166−173.

[96] Huangfu Y, Zhang W. Warp-Based load/store reordering to improve gpu time predictability. JCSE, 2017,11(2).

[97] Chen G, Guan N, Lü MS, Wang Y. State-of-the-Art survey of real-time multicore system. Ruan Jian Xue Bao/ Journal of Software, 2018,29(7):2152−2176 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5580.htm [doi: 10.13328/j.cnki.jos.005580]

[98] Kato S, Lakshmanan K, Rajkumar R, Ishikawa Y. TimeGraph: GPU scheduling for real-time multi-tasking environments. In: Proc. of the 2011 USENIX Conf. on USENIX Annual Technical Conf. ACM, 2011. 2.

[99] Kato S, Lakshmanan K, Kumar A, Kelkar M, Ishikawa Y, Rajkumar R. RGEM: A responsive GPGPU execution model for runtime engines. In: Proc. of the 32nd Real-Time Systems Symp. (RTSS). IEEE, 2011. 57−66.

[100] Basaran C, Kang KD. Supporting preemptive task executions and memory copies in GPGPUS. In: Proc. of the 24th Euromicro Conf. on Real-Time Systems (ECRTS). IEEE, 2012. 287−296.

[101] Zhong J, He B. Kernelet: High-throughput gpu kernel executions with dynamic slicing and scheduling. IEEE Trans. on Parallel Distrib. Syst., 2014,25(6):1522−1532.

[102] Verner U, Schuster A, Silberstein M, Mendelson A. Scheduling processing of real-time data streams on heterogeneous multi-GPU systems. In: Proc. of the 5th Annual Int'l Systems and Storage Conf. (SYSTOR). ACM, 2012.

[103] Verner U, Mendelson A, Schuster A. Batch method for efficient resource sharing in real-time multi-GPU systems. In: Proc. of the 15th Int'l Conf. on Distributed Computing and Networking (ICDCN). Springer-Verlag, 2014. 347−362.

[104] Verner U, Mendelson A, Schuster A. Scheduling periodic real-time communication in multi-GPU systems. In: Proc. of the 23rd Int'l Conf. on Computer Communication and Networks (ICCCN). IEEE, 2014. 1−8.

[105] Kim J, Andersson B, De Niz D, Rajkumar R. Segment-Fixed priority scheduling for self-suspending real-time tasks. In: Proc. of the 34th Real-Time Systems Symp. (RTSS). IEEE, 2013. 246−257.

[106] Chen G, Zhao Y, Shen X, Zhou H. EffiSha: A software framework for enabling effficient preemptive scheduling of GPU. In: Proc. of the PPoPP. 2017. 3−16.

[107] Wang J, Rubin N, Sidelnik A, Yalamanchili S. Dynamic thread block launch: A lightweight execution mechanism to support irregular applications on GPUS. ACM SIGARCH Computer Architecture News, 2015,43(3):528−540.

[108] Hosseinimotlagh S, Kim H. Thermal-Aware servers for real-time tasks on multi-core GPU-integrated embedded systems. In: Proc. of the 25th IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS). IEEE, 2019. 254−266.

[109] Nugteren C, Van den Braak GJ, Corporaal H, Bal HE. A detailed GPU cache model based on reuse distance theory. In: Proc. of the 20th Int'l Symp. on High Performance Computer Architecture (HPCA). IEEE, 2014. 37−48.

[110] Liang Y, Li X. Efficient kernel management on GPUS. ACM Transactions on Computer Systems, 2017,16(4):115:1−115:24.

[111] Park JJK, Park Y, Mahlke S. Dynamic resource management for efficient utilization of multitasking GPUS. ACM SIGARCH Computer Architecture News, 2017,45(1):527−540.

[112] Elliott GA, Ward BC, Anderson JH. GPUSync: A framework for real-time GPU management. In: Proc. of the Real-Time Systems Symp. 2013. 33−44.

[113] Pellizzoni R, Betti E, Bak S, Yao G, Criswell J, Caccamo M, Kegley R. A predictable execution model for cots-based embedded systems. In: Proc. of the 17th Real-Time and Embedded Technology and Applications Symp. (RTAS). IEEE, 2011. 269−279.

[114] Alhammad A, Pellizzoni R. Time-Predictable execution of multithreaded applications on multicore systems. In: Proc. of the Design, Automation & Test in Europe Conf. (DATE). European Design and Automation Association, 2014. 1−6.

[115] Abdelouahab K, Pelcat M, Serot J, Berry F. Accelerating CNN inference on fpgas: A survey. 2018.

[116] Kwon K, Amid A, Gholami A, Wu B, Asanovic K, Keutzer K. Co-Design of deep neural nets and neural net accelerators for embedded vision applications. 2018. 1−6.

[117] Yang Y, Huang Q, Wu B, Zhang T, Ma L, Gambardella G, Blott M, Lavagno L, Vissers K, Wawrzynek J, Keutzer K. Synetgy: Algorithm-hardware co-design for convnet accelerators on embedded fpgas. 2018. 23−32.

[118] Gao M, Yang X, Pu J, Horowitz M, Kozyrakis C. TANGRAM: Optimized coarse-grained dataflow for scalable NN accelerators. In: Proc. of the 24th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS). 2019. 807−820.

[119] GPUOpen. https://gpuopen.com/

[120] OpenCL. https://www.khronos.org/opencl/

[121] Luo Y, Li S, Sun K, Renteria R, Choi K. Implementation of deep learning neural network for real-time object recognition in opencl framework. In: Proc. of the Int'l SoC Design Conf. (ISOCC). IEEE, 2017. 298−299.

附中文参考文献:

[3] 郑泽宇,顾思宇.TensorFlow:实战 Google 深度学习框架.北京:电子工业出版社,2017.

[29] 周志华.Machine Learning:机器学习.北京:清华大学出版社,2016.

[34] 彭靖田,林健,白小龙.深入理解 TensorFlow 架构设计与实现原理.北京:人民邮电出版社,2018.

[38] 陈桂林,胜马,阳郭.硬件加速神经网络综述.计算机发展与研究,2019,56(2):240−253.

[97] 陈刚,关楠,吕鸣松,王义.实时多核嵌入式系统研究综述.软件学报,2018,29(7):2152−2176. http://www.jos.org.cn/1000-9825/5580.htm [doi: 10.13328/j.cnki.jos.005580]

张政馗(1984－),男,博士,讲师,主要研究领域为实时系统模型检测,实时系统设计与时间分析.

庞为光(1989－),男,硕士,主要研究领域为实时嵌入式系统,实时人工智能系统.

谢文静(1994－),女,学士,主要研究领域为实时系统时间分析,GPU 性能分析.

吕鸣松(1980－),男,博士,副教授,主要研究领域为实时系统时间分析,实时操作系统.

王义(1961－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为形式化方法,实时嵌入式系统.