The Need for Computational Pluralism

Deeksha Dangwal
Unaffiliated
Seattle, Washington, USA
deekshadangwal@gmail.com

Abhejit Rajagopal
Allen Institute
Seattle, Washington, USA
abhe.rajagopal@alleninstitute.org

Abstract

Many emerging applications require heterogeneous deployment as a technical necessity: smart city infrastructure processing sensor data with sub-millisecond latency requirements, federated learning enabling privacy-preserving collaboration across institutions, and most recently offline personalized AI-powered devices. Yet, the move towards cloud-first infrastructure has left important gaps in the ecosystem. This position paper argues for *computational pluralism*: preserving viable pathways for cloud, edge, and local execution to coexist as complementary tiers rather than competing alternatives. Beyond technical necessity, pluralism enables properties that no single tier can provide: user control over sensitive data [17], resilience through architectural diversity, reduced environmental impact for latency-tolerant workloads [2], geographically distributed innovation capacity, and ultimately the *autonomy to compute*.

1 Introduction

Cloud computing has delivered transformative benefits; economies of scale amortize infrastructure costs and achieve high utilization through sophisticated resource management techniques [20], elastic resource provisioning enables applications to scale from zero to millions of users, serverless platforms handle trillions of monthly requests [3], and exascale workload aggregation surprisingly simplifies daunting data consistency and data serving challenges [5]. These achievements have rightfully driven widespread cloud adoption. Yet there have been unintended second-order effects.

More applications are becoming accessible *exclusively* through remote services. For example, as of January 2026 Apple's Siri does not work without an internet connection, frontier AI models must be accessed via remote APIs, and popular word processing software cannot run locally without cloud connectivity. This has lead to a decline in demand for local computational capacity. Recently we observed these effects trickle into the hardware manufacturing industry: Micron announced their exit from the consumer memory market in December 2025, explicitly citing AI-driven datacenter growth [1], leading to consumer DRAM prices increasing by over 300%. Nvidia is also rumored to decrease production of consumer graphics cards, citing memory shortages [16]. The result has been a self-reinforcing cycle where *datacenter needs drive hardware reorientation*, making local execution progressively less viable.

Singular reliance on remote cloud execution limits investment in alternative hardware and software topologies that may in fact achieve better performance when integrated at a global scale for specialized applications. For example, querying sensitive demographic information or health records demands higher privacy, where cloud centralization creates unnecessary infrastructure burden [8, 22]. Similarly, some applications require real-time feedback, e.g. a wearable device translating languages in a foreign country, or if a self-driving car needs to make an urgent decision about directions. There are still other applications which do not require an immediate response and for which we may want to choose solutions with a lower carbon or energy cost, such as for edge devices or sensor networks operating in low-power nodes with intermittent internet connections. These applications are by all counts more prevalent than the comparatively small number of large-scale AI models captivating datacenter scaling [7].

Moreover, environmental costs scale with centralization. US datacenters produced an estimated 105 million tons CO2 equivalent (CO2eq) in 2024, 3x more than 2018 levels [11]. Large facilities consume up to 5 million gallons of water *daily* for cooling, equivalent to the water used by town of 50,000 people [23]. While carbon-aware scheduling can reduce emissions by 15-65% through intelligent workload placement [2], a comparatively more efficient solution may be to compute locally, requiring no water for cooling at all.

1.1 Our Position

Our position is that the research community should prioritize and invent systems, abstractions, and policies that make heterogeneous deployment straightforward rather than exceptional. This includes research enabling portability across execution environments, intelligent workload placement frameworks that consider latency, privacy, cost, and carbon alongside raw performance [19], and economic models that preserve hardware ecosystems supporting diverse deployment scenarios.

2 Computational Pluralism

Pluralism as a philosophy supports a worldview of multiplicity; it is a position rooted in pragmatism and context. We define *computational pluralism* not merely as the coexistence of cloud and local deployment, but an ecosystem where multiple viable execution environments create evolutionary pressure for technical advancement. In a pluralistic infrastructure, the same computation can execute in datacenters

for batch processing, on edge devices for latency-critical applications, across federated networks for privacy-sensitive workloads, or on local hardware for rapid prototyping. Each deployment context surfaces distinct optimization challenges that drive innovation.

Importantly, many emerging applications require pluralistic compute as a technical necessity rather than a preference. Smart city infrastructure cannot route every sensor reading through distant datacenters when traffic light timing requires sub-millisecond decisions. Smart glasses performing real-time visual understanding cannot tolerate the latency, energy, and bandwidth costs of streaming video to the cloud for every inference. Modern satellite systems in low Earth orbit must process data physically distributed across a mesh with intermittent connectivity to ground stations [6, 21]. Autonomous vehicles cannot depend on stable cellular connections for safety-critical perception [15]. These applications demand a computational fabric that spans edge devices, local processing, and selective cloud coordination. Overreliance on any single deployment model makes entire classes of applications technically infeasible.

2.1 Technical Challenges

As massively distributed hybrid execution becomes more viable, the paradigm of data centralization for machine learning are shifting. Recently, agentic systems are exploiting computational heterogeneity by coupling a common compute model, e.g. LLMs, with agent-specific context (local data), skills (pre-compiled software), and compute capabilities (local accelerators, RAM, etc.) [4, 18]. We believe, the success of such systems will rely on fully utilizing the compute fabric available, i.e., *embracing computational pluralism*.

The primary challenge is coordinating compute, including managing algorithmic tradeoffs, e.g. when to exchange weights and results, how much to prioritize local over global context, which specialized resources to allocate and when. Thus, realizing computational pluralism requires addressing fundamental gaps in systems abstractions, scheduling algorithms through hardware-software co-design.

I. Balancing conflicting constraints
Applications increasingly demand optimization across latency, privacy, energy, and carbon simultaneously. A video analytics pipeline might achieve the lowest latency by processing frames on the cloud but minimize privacy risk through on-device inference or federated learning. Federated learning frameworks [12] optimize for training speed through participant selection but do not consider per-device energy budgets or carbon footprint of distributed training. We need (a) *placement algorithms and programming abstractions* that expose these tradeoffs to developers, (b) enable *per-application priority specification* (e.g., "optimize for privacy subject to 100ms latency bound"), and (c) *dynamically adapt as conditions change*, e.g. when grid carbon intensity varies, network bandwidth fluctuates, or device battery levels deplete, etc.

II. Managing resource-limited heterogeneous devices
Modern edge devices increasingly integrate heterogeneous compute units: CPUs, NPUs, DSPs, and accelerators often without traditional OS support. MCUNet demonstrates viable inference on microcontrollers with 320KB SRAM [14], and TensorFlow Lite Micro enables deployment on systems 2-3 orders of magnitude smaller than mobile phones [10]. The challenge is dynamic workload partitioning: should a visual processing pipeline execute CNN layers on the NPU or CPU? When should the pipeline query the cloud? How do we balance compute allocation when optimizing for minimum latency versus minimum energy differs fundamentally? Whereas peak performance burns power budgets, energy-optimal scheduling may miss real-time deadlines. These edge scenarios demand microsecond decisions with milliwatt power budgets. This requires co-designed hardware performance counters exposing fine-grained energy costs, software schedulers reasoning about power-latency Pareto frontiers at the instruction level, and dynamic orchestration of heterogeneous compute units.

III. Portability
While frameworks such as ONNX provide some standardization, deployment portability across tiers, e.g. cloud vs. edge microcontrollers, is a manual and brittle process. Today, developers must manually quantize for edge devices, rewrite batch processing for heterogeneous clusters, and detect and handle transpilation errors. This can be solved using automatic lifting and transpilation: taking a high-level inference script (e.g. PyTorch) and automatically generating implementations for various targets. The challenge is semantic preservation, and the ability to reason about tradeoffs, e.g. should this layer execute locally (low latency, privacy preserved) or remotely (higher accuracy, energy cost)? How do we verify that a transpiled edge implementation maintains sufficient accuracy relative to the cloud version? How do we manage this error when models are partially running on each tier? Compiler infrastructures like TVM [9] and MLIR [13] provide lowering paths from high-level frameworks to hardware backends, but assume a single target platform.

3 Conclusion

Cloud computing has fundamentally transformed how we build and deploy software, and its centralized infrastructure will remain essential for applications requiring massive scale. The challenge is ensuring that cloud dominance does not inadvertently foreclose architectural approaches serving distinct and important needs: privacy-preserving local inference, latency-critical edge processing, offline-capable systems to handle intermittent connectivity, and carbon-aware placement across geographic regions. While a failure to address these execution scenarios will definitely not crumble the global computing industry, the upside of prioritizing a vision of computational pluralism is a more connected, interoperable, compute fabric that scales for everyone.

References

[1] 2025. Micron announces exit crucial consumer business. https://investors.micron.com/news-releases/news-release-details/micron-announces-exit-crucial-consumer-business

[2] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon explorer: A holistic framework for designing carbon aware datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 118–132.

[3] Alexandru Agache, Marc Brooker, Alexandra Iordache, Anthony Liguori, Rolf Neugebauer, Phil Piwonka, and Diana-Maria Popa. 2020. Firecracker: Lightweight virtualization for serverless applications. In *17th USENIX symposium on networked systems design and implementation (NSDI 20)*. 419–434.

[4] Anthropic Engineering Team. 2025. How We Built Our Multi-Agent Research System. Anthropic Engineering Blog. https://www.anthropic.com/engineering/multi-agent-research-system

[5] James Bornholt, Rajeev Joshi, Vytautas Astrauskas, Brendan Cully, Bernhard Kragl, Seth Markle, Kyle Sauri, Drew Schleit, Grant Slatton, Serdar Tasiran, et al. 2021. Using lightweight formal methods to validate a key-value storage node in Amazon S3. In *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*. 836–850.

[6] Jiaqi Cao, Shengli Zhang, Qingxia Chen, Houtian Wang, Mingzhe Wang, and Naijin Liu. 2023. Computing-aware routing for leo satellite networks: A transmission and computation integration approach. *IEEE Transactions on Vehicular Technology* 72, 12 (2023), 16607–16623.

[7] Robin Chataut, Alex Phoummalayvane, and Robert Akl. 2023. Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. *Sensors* 23, 16 (2023), 7194.

[8] Jim Q Chen and Allen Benusa. 2017. HIPAA security compliance challenges: The case for small healthcare providers. *International Journal of Healthcare Management* 10, 2 (2017), 135–146.

[9] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. 2018. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*. 578–594.

[10] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Tiezhen Wang, et al. 2021. Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of machine learning and systems* 3 (2021), 800–811.

[11] Gianluca Guidi, Francesca Dominici, Jonathan Gilmour, Kevin Butler, Eric Bell, Scott Delaney, and Falco J Bargagli-Stoffi. 2024. Environmental burden of United States data centers in the artificial intelligence era. *arXiv preprint arXiv:2411.09786* (2024).

[12] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient federated learning via guided participant selection. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*. 19–35.

[13] Chris Lattner, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques Pienaar, River Riddle, Tatiana Shpeisman, Nicolas Vasilache, and Oleksandr Zinenko. 2021. MLIR: Scaling compiler infrastructure for domain specific computation. In *2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*. IEEE, 2–14.

[14] Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. 2020. Mcunet: Tiny deep learning on iot devices. *Advances in neural information processing systems* 33 (2020), 11711–11722.

[15] Guoman Liu, Jing Sheng, and Zhen Tao. 2025. Application and design of a decision-making model in ethical dilemma for self-driving cars. *Scientific Reports* 15, 1 (2025), 8187.

[16] PC Magazine. [n. d.]. https://www.pcmag.com/news/nvidia-might-cut-rtx-50-gpu-supply-by-up-to-40-in-2026-due-to-memory-shortages

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[18] Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O'Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, Katarina Slama, Lama Ahmad, Paul McMillan, Alex Beutel, Alexandre Passos, and David G. Robinson. 2023. *Practices for Governing Agentic AI Systems*. Technical Report. OpenAI. https://cdn.openai.com/papers/practices-for-governing-agentic-ai-systems.pdf

[19] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. 2024. On the limitations of carbon-aware temporal and spatial workload shifting in the cloud. In *Proceedings of the Nineteenth European Conference on Computer Systems*. 924–941.

[20] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proceedings of the tenth european conference on computer systems*. 1–17.

[21] Shangguang Wang and Qing Li. 2023. Satellite computing: Vision and challenges. *IEEE Internet of Things Journal* 10, 24 (2023), 22514–22529.

[22] Ruoyu Wu, Gail-Joon Ahn, and Hongxin Hu. 2012. Towards HIPAA-compliant healthcare systems. In *Proceedings of the 2nd acm sighit international health informatics symposium*. 593–602.

[23] Miguel Yañez-Barnuevo. [n. d.]. https://www.eesi.org/articles/view/data-centers-and-water-consumption