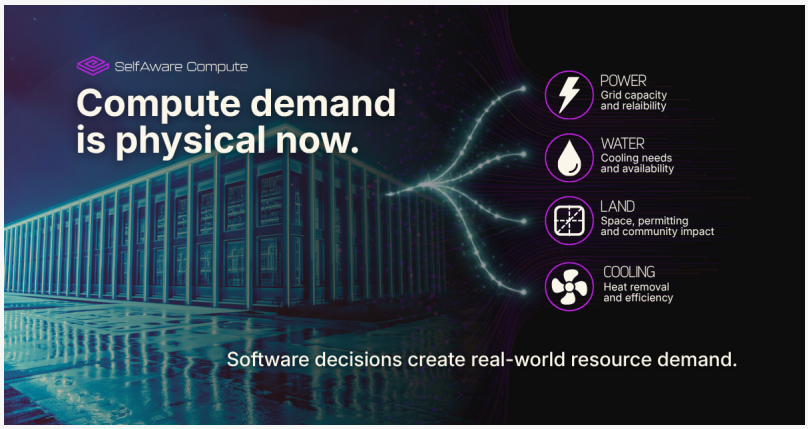
SelfAware Compute Points to Software Execution as Missing Layer in Data Center Efficiency

Execution-pathway optimization improves compute efficiency, resource utilization, and energy use by helping software run more efficiently.



Data center growth is putting pressure on power, water, land, cooling, chips, capital, and public trust. SelfAware Compute focuses on the software execution layer underneath that demand.

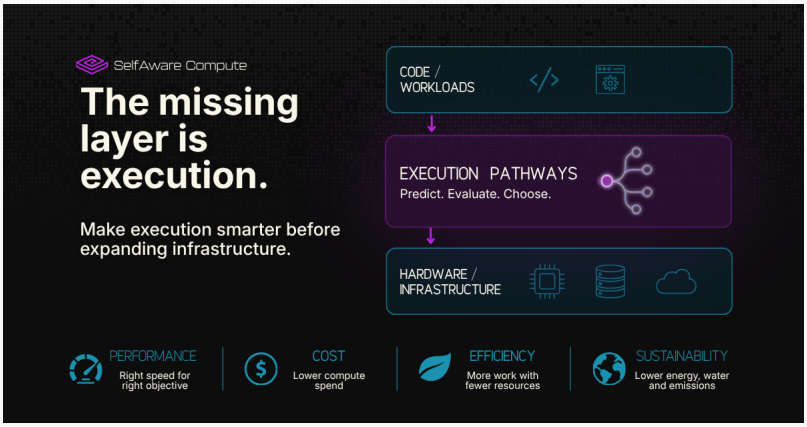
SCOTTSDALE, AZ, UNITED STATES, June 17, 2026 /EINPresswire.com/ -- As data center growth emerges as a defining infrastructure issue, SelfAware Compute points to a less visible part of the efficiency problem. Software execution remains difficult to predict and control at exactly the moment data center decisions are becoming some of the most expensive, visible, and consequential infrastructure investments being made.

Across the United States, companies are investing heavily in new data centers and the physical resources required to support them. The buildout is tied to AI, cloud services, enterprise computing, and broader digital infrastructure. Communities are asking whether these projects are being planned with enough visibility into how the compute will actually be used.

The scale is becoming harder to ignore. According to Lawrence Berkeley National Laboratory, U.S. data center electricity use reached 176 terawatt-hours in 2023 and is estimated to reach 325 to 580 terawatt-hours by 2028. Using the EPA's household electricity conversion, that projected range is roughly comparable to the annual electricity use of 32 million to 57 million U.S. homes. Water is part of the same infrastructure burden. The Environmental and Energy Study Institute reports that larger data centers can use up to 5 million gallons of water per day for cooling, roughly the daily water use of a town of 10,000 to 50,000 people.

This is no longer only a software performance problem. Compute demand is solidly part of the physical world, where software decisions become infrastructure decisions. The next era of digital

infrastructure cannot be built by treating inefficient execution as someone else's problem to absorb.



Software execution sits between workloads and infrastructure. SelfAware Compute analyzes execution pathways so teams can make better decisions before adding more hardware, power, or cooling.

For decades, the default answer to rising software demand was more hardware. Faster processors, more cores, more servers, more cloud capacity. That approach got the industry far. But it is no longer enough.

The data center conversation focuses on the very real constraints of the physical layer. When workloads cannot be predicted, scheduled, or optimized around a defined objective, organizations compensate by adding extra capacity to reduce the risk of failure, delay, or performance loss. SelfAware works at the software layer underneath that demand.

> When execution cannot be predicted, systems compensate with excess hardware, memory, cooling and budget. Software should explain what it will do before those costs are committed."
>
> *Kevin Howard, Chief Technologist, SelfAware Compute*

Building on the history of Massively Parallel Technologies, SelfAware Compute is the company's modern focus. The current work centers on a core question in parallel computing: how should software execution be understood, predicted, and controlled before more compute resources are committed?

"When execution cannot be predicted, systems compensate with excess," said Kevin Howard, chief technologist at SelfAware Compute. "Extra hardware, memory, cooling, and budget all become safety margins. The question is whether software can explain what it is going to do before those costs are already committed."

That uncertainty has a cost in both money and resources. More runtime means more power. More memory pressure pushes teams toward larger systems than a workload needs. More heat increases operating expense and strains local water systems. Better control over execution does not eliminate infrastructure costs, but it gives teams a clearer basis for deciding which resources are needed, which strategy fits the objective, and where excess spend and resource use are avoidable.

SelfAware's technical approach starts with execution pathways, the routes software takes

through branches, loops, functions, and data-dependent behavior. Instead of treating code as a single fixed object, SelfAware analyzes the paths that activate and how they behave under specific conditions. That model informs decisions about runtime, memory use, energy use, cost, and deployment behavior.

SelfAware Compute helps software understand how it runs, making execution more visible, predictable, and controllable across existing code and infrastructure.

“If you cannot know what path of execution the code is going to take before you run it, then you cannot predict the behavior of the system,” Howard said. “Once we know what pathway is going to be activated, we can start making better decisions about how the system should execute.”

This approach complements compilers, profilers, infrastructure schedulers, and data center planning tools. Compilers translate and optimize instructions. Profilers show what happened during a run. Infrastructure tools place and manage workloads. SelfAware focuses on the decision layer before execution, where software evaluates how it should run based on the objective, input, and target environment, instead of waiting until after compute has already been spent.

The objective changes from run to run. The fastest path is not always the best path. The lowest-resource path is not always the slowest. In some cases, using fewer resources produces the better outcome for the goal, not a compromise against it.

Data center buildout has become a global pressure point. The argument is often framed as a choice between building more capacity or limiting demand. SelfAware adds a third path, making execution smarter before the next layer of infrastructure is committed.

For enterprises, technical teams, and infrastructure operators, predictable software execution supports better decisions about where work runs, how resources are allocated, and when additional capacity is actually required.

SelfAware targets existing code and infrastructure. Many valuable systems run on legacy code, existing hardware, and operational requirements that cannot be rewritten every time infrastructure strategy changes.

The industry needs more creative ways to reduce the impact of data center growth. SelfAware Compute addresses a layer that is usually overlooked: software execution that understands

itself.

SelfAware continues to explore multiple paths for bringing its Optimization Engine to market, including enterprise engagements, technical briefings, and service-based evaluation models. To learn more or request a technical briefing, visit SelfAware Compute or contact Shannon Kendall at shannon.kendall@selfawarecompute.com.

About SelfAware Compute
SelfAware Compute is the current public-facing brand for Massively Parallel Technologies. The company focuses on predictive software optimization, execution-pathway analysis, and compute efficiency. SelfAware helps software understand how it runs so teams can make better execution decisions with existing code and infrastructure where possible.

Shannon Kendall
SelfAware Compute
email us here
Visit us on social media:
LinkedIn
X
Bluesky
YouTube