

PhysaFlow Stranded Capacity Index (PSCI)

An Operational Taxonomy of Unproductive Power & Compute in Modern AI Infrastructure

— PHYSAFLOW ENGINEERING & RESEARCH GROUP

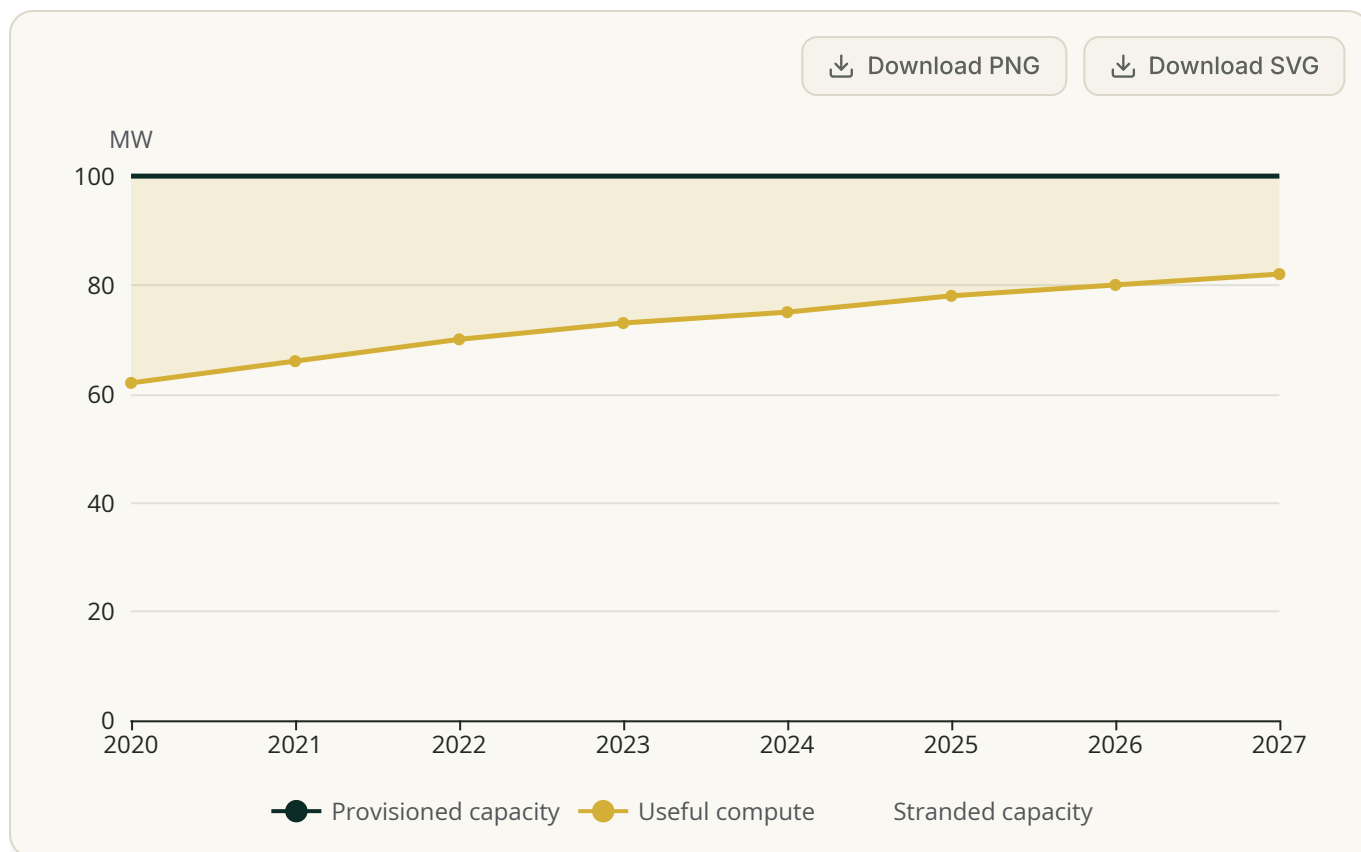
1. Executive Summary & Introduction

1.1. Context & Industry Bottlenecks

Modern AI data centers operate under severe physical and operational constraints. While demand for high-density compute accelerates exponentially, the physical expansion of grid power and cooling capacity is severely bottlenecked. Grid interconnection queues extend five to seven years in most markets, while model complexity and training compute grow annually — a widening mismatch between what the industry wants to consume and what the physical infrastructure can deliver.

1.2. Problem Definition

A critical, yet under-documented issue in contemporary facilities is **Stranded Capacity** — the paid-for, fully provisioned physical and digital capacity that yields zero productive compute due to a lack of multi-layer coordination between the facility, IT hardware, and workload orchestrators. Capacity is stranded when it is contracted, energized, and ready, yet unavailable in practice: power present but cooling saturated, GPU clusters idle from scheduling limits, or resources reserved but never consumed.



The Stranded Capacity Gap: provisioned capacity remains constant at its contractual level while useful compute fluctuates below it; the shaded band between both lines represents stranded capacity. Values are illustrative and do not represent production measurements.

1.3. Scope & Objectives

This report establishes a formal taxonomy to categorize, quantify, and eliminate stranded capacity across all operational layers. It is a reference, not a prescriptive guide, and provides a shared vocabulary for capacity loss across teams. Specific objectives:

Categorize stranded capacity into recurring patterns across the Facility, IT, and Workload layers.

Quantify each pattern with operational signals, cost estimates, and root causes.

Benchmark typical inefficiency ranges observed in hyperscale environments.

Guide remediation through cross-layer dynamic coordination and real-time telemetry.

Figures and examples in this document are illustrative and do not represent production measurements.

2. Taxonomy of Stranded Capacity

The taxonomy groups stranded capacity into three complementary operational layers — Facility, IT, and Workload — each with its own recurring patterns, symptoms, costs, and root causes.

The layered view matters because stranded capacity rarely announces itself in a single component. It tends to emerge at the seams — a cooling plant sized for steady baseload, an orchestrator blind to thermal state, a power chain carrying static headroom that nobody measures. Reading the report layer by layer makes those seams visible, and the taxonomy gives operators a common language for triage: when capacity sits idle, it points to where to look first, what to measure, and which fix belongs to which discipline — Facility first, then IT, then Workload, mirroring how constraints cascade in practice.

2.1. Facility (Power & Thermal Infrastructure)

The Facility layer covers power delivery and thermal management: transformers, UPS systems, PDUs, chilled water loops, and containment. It is designed with long planning horizons and significant capital investment, and it often becomes the first source of operational constraints as AI demand grows — compute capacity may exist while power distribution, cooling efficiency, or physical infrastructure prevent full utilization.

2.1.1. Static Thermal Headroom (STH)

Over-cooling driven by static safety margins that decouple cooling output from actual dynamic heat generation.

— WHAT YOU SEE

Chilled water loop temperatures running 3–5 °C lower than ASHRAE recommended thresholds.

PUE remains static despite drops in compute utilization.

— WHAT IT COSTS

8–14% excess energy consumption at the chiller plant.

Thousands of gallons of evaporative water loss daily.

— WHY IT OCCURS

Facility operators set static safety margins to prevent localized hot spots during unpredictable GPU power spikes, decoupling cooling output from actual dynamic heat generation.

2.1.2. Power Distribution Stranding (PDS)

Capacity held in transformers, UPS systems, and PDUs by theoretical peak draws that are never reached.

— WHAT YOU SEE

Power Distribution Units (PDUs) capped at 65% utilization while nameplate capacity indicates available headroom.

UPS load factors below design intent despite active compute demand.

— WHAT IT COSTS

Megawatts of contracted grid power locked behind administrative caps that cannot be assigned to new racks.

— WHY IT OCCURS

Conservative static provisioning based on worst-case theoretical peak draws rather than real-time telemetry-backed dynamic capping.

2.1.3. Liquid Cooling Flow Mismatch (LCFM)

Static coolant flow rates applied to dynamic GPU loads, leaving thermal capacity unused or overcooling idle hardware.

— WHAT YOU SEE

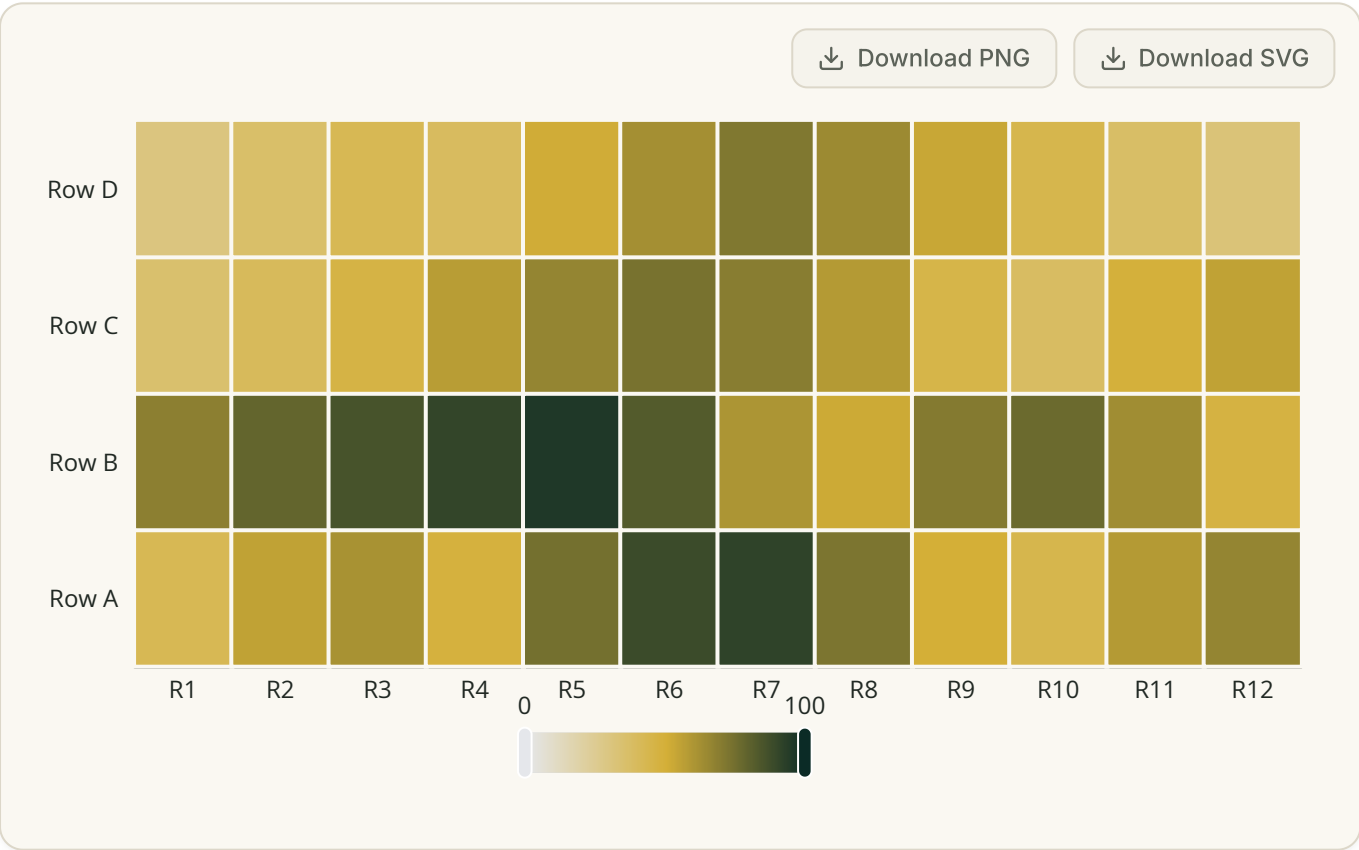
Coolant flow setpoints unchanged across idle and peak workload phases.
GPU inlet temperatures well below design targets while flow remains at nominal.

WHAT IT COSTS

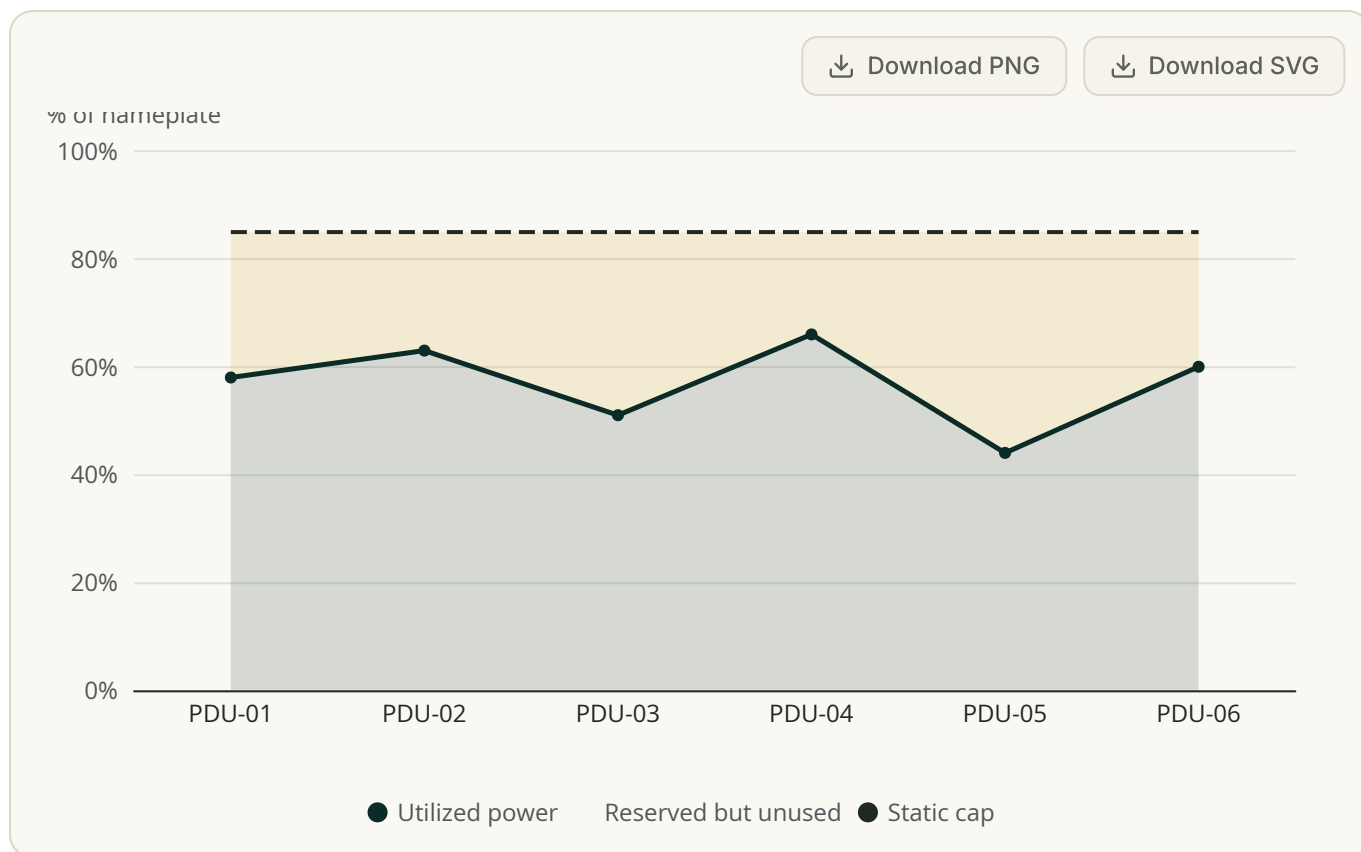
Pumping energy and cooling capacity committed to hardware that does not need it, reducing the thermal envelope available for productive racks.

WHY IT OCCURS

Flow setpoints are commissioned once and rarely rebalanced because cooling control loops do not receive workload telemetry.



Thermal headroom and localized hot spots across a rack grid: cool zones show unused thermal capacity, gold zones sit on static margins, and dark zones are at their thermal limit while adjacent racks stay idle. Values are illustrative and do not represent production measurements.



Provisioned versus utilized power at PDU level: the shaded gold band represents reserved-but-unused capacity held by static caps. The dashed green line marks the contractual provisioning limit that telemetry-backed capping could reassign. Values are illustrative and do not represent production measurements.

2.2. IT & Hardware Infrastructure

Even when sufficient power and cooling are available, limitations within compute platforms, hardware allocation strategies, or infrastructure management processes can prevent effective utilization of installed capacity — compute resources that are technically available but cannot be efficiently assigned to productive workloads.

2.2.1. Unbalanced Thermal Throttling (UTT)

Frequency reduction in specific nodes caused by regional thermal misalignment, degrading whole-cluster throughput.

— WHAT YOU SEE

GPU clocks stepping down on individual racks while adjacent racks run at full boost.
Temperature deltas across the cluster exceeding 10 °C under identical workloads.

— WHAT IT COSTS

Reduced training throughput: slow nodes hold back synchronized steps across the entire job.

— WHY IT OCCURS

Cooling distribution is imbalanced while scheduling is thermal-agnostic, so hot nodes throttle instead of the workload moving or the cooling adapting.

2.2.2. Underutilized Interconnect Bandwidth (UIB)

Network latency and inefficiency that keep high-end GPUs idle waiting for data.

— WHAT YOU SEE

Fabric utilization well below capacity while GPU compute utilization drops during all-reduce and checkpoint phases.

Long tail latencies on specific paths despite low average utilization.

— WHAT IT COSTS

Expensive GPU-seconds burned on data movement instead of computation across every training step.

— WHY IT OCCURS

Topology-unaware placement or static routing spreads traffic unevenly, and orchestration does not co-schedule data movement with compute.

2.2.3. Power Cap Buffering (PCB)

Rigid per-server power caps that prevent optimal performance even when the facility has headroom available.

— WHAT YOU SEE

Servers pinned below their allowed power envelope while PDUs report surplus capacity.

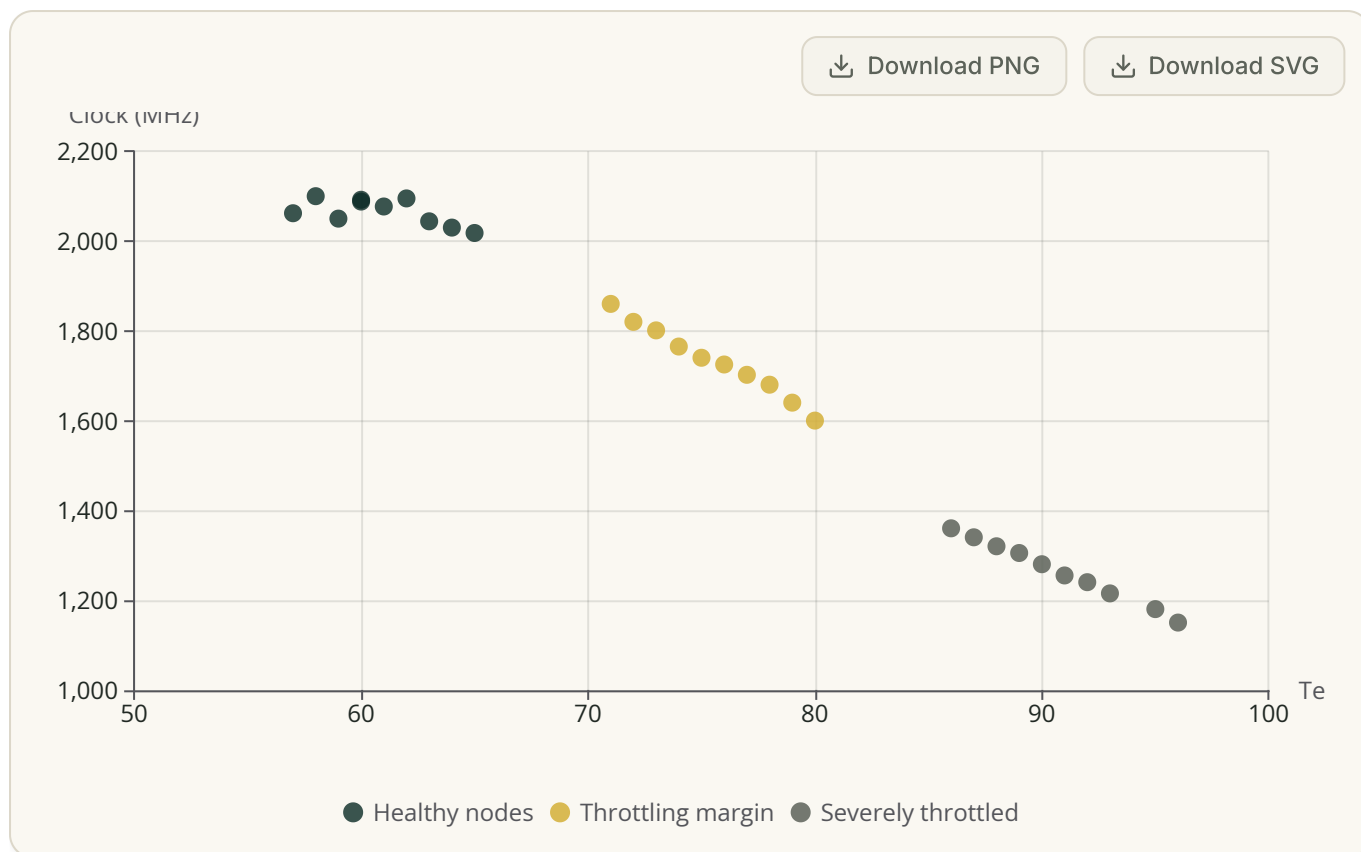
Sustained power capping events on nodes that never trip facility limits.

— WHAT IT COSTS

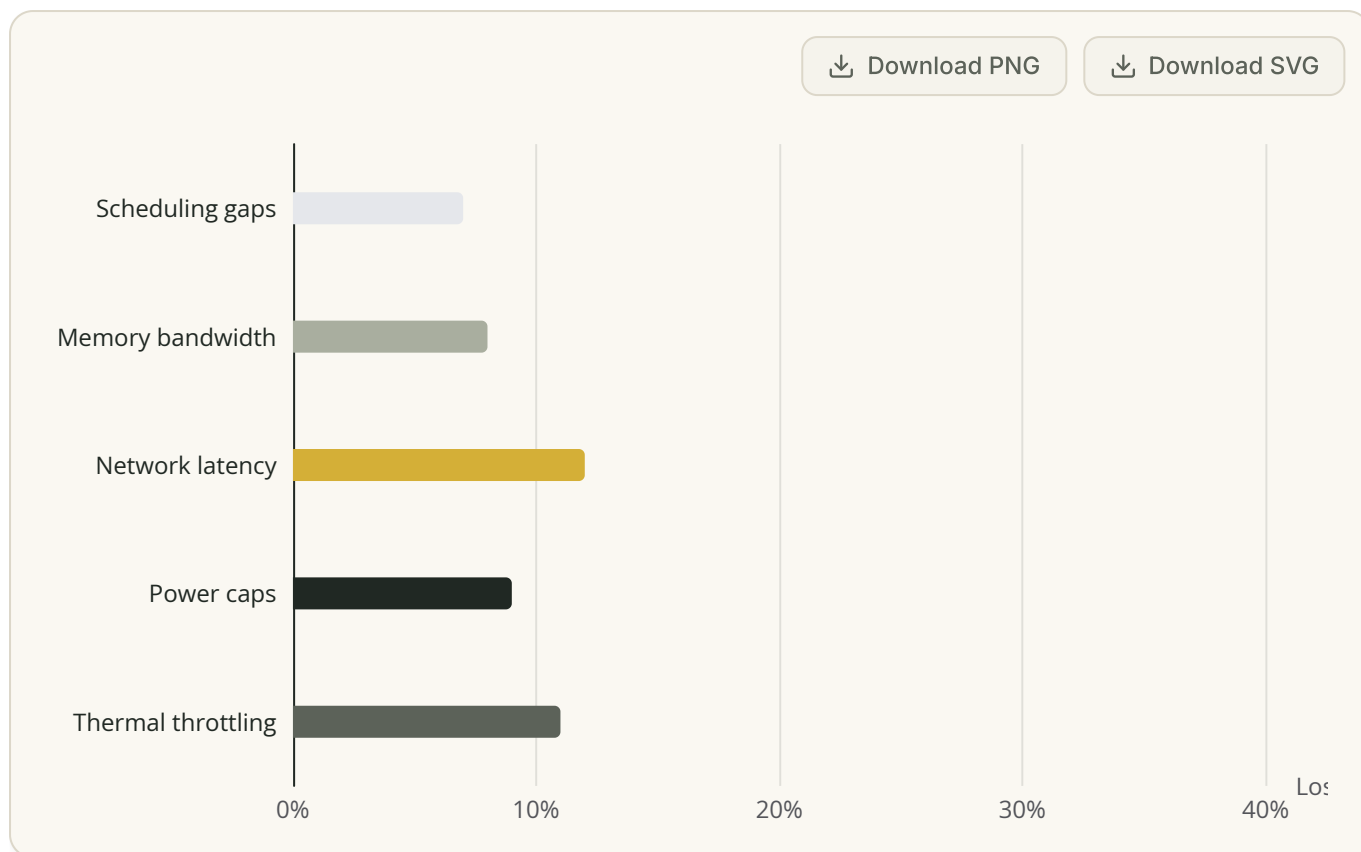
Predictable performance loss bounded only by the cap, not by actual thermal or power constraints.

— WHY IT OCCURS

Power caps are assigned statically at provisioning time and never revisited against real telemetry, so safety buffers become permanent performance ceilings.



GPU clock frequency versus temperature across cluster racks: healthy nodes cluster at high clocks and moderate temperatures, while thermally misaligned nodes drop clocks sharply — the signature of unbalanced thermal throttling. Values are illustrative and do not represent production measurements.



Model FLOPs Utilization (MFU) loss breakdown: memory bandwidth, network latency, static power caps, and thermal throttling each consume a share of the theoretical FLOPs peak; scheduling gaps account for the remainder. Values are illustrative and do not represent production measurements.

2.3. Workload & Orchestration

The Workload layer addresses orchestration, scheduling, and workload placement. Even with power, cooling, and hardware fully available, scheduling decisions — or their absence — can leave the most expensive compute hardware in the facility idle, waiting on synchronization, thermal awareness, or overly conservative reservations.

2.3.1. Non-Aware Thermal Scheduling (NATS)

Planning of massive AI jobs without considering the thermal or energy state of the facility.

— WHAT YOU SEE

Jobs packed into thermally constrained zones while chilled zones sit empty.
Hot spots forming at a predictable cadence that mirrors the job schedule.

— WHAT IT COSTS

Throttling cascades that reduce job throughput despite nominal capacity being available elsewhere.

— WHY IT OCCURS

Schedulers select nodes from a free-list, ignoring thermal and power telemetry, so placement actively creates the constraints the facility then has to manage reactively.

2.3.2. Barrier Synchronization Idle (BSI)

Dead time for high-performance GPUs while waiting for synchronization with slower nodes.

— WHAT YOU SEE

GPU utilization dropping in step with all-reduce and barrier phases of distributed training.

Total step time significantly exceeding the compute phase of the slowest node.

— WHAT IT COSTS

Expensive GPU-seconds spent idle on every training step, multiplied across thousands of steps.

— WHY IT OCCURS

Distributed training is only as fast as its slowest member; without topology-aware scheduling and co-design, stragglers force the entire cluster to wait.

2.3.3. Over-Provisioned Job Allocation (OPJA)

Excessive resource reservation by developers to avoid out-of-memory (OOM) failures.

— WHAT YOU SEE

Jobs requesting far more memory and accelerators than their measured steady-state usage.

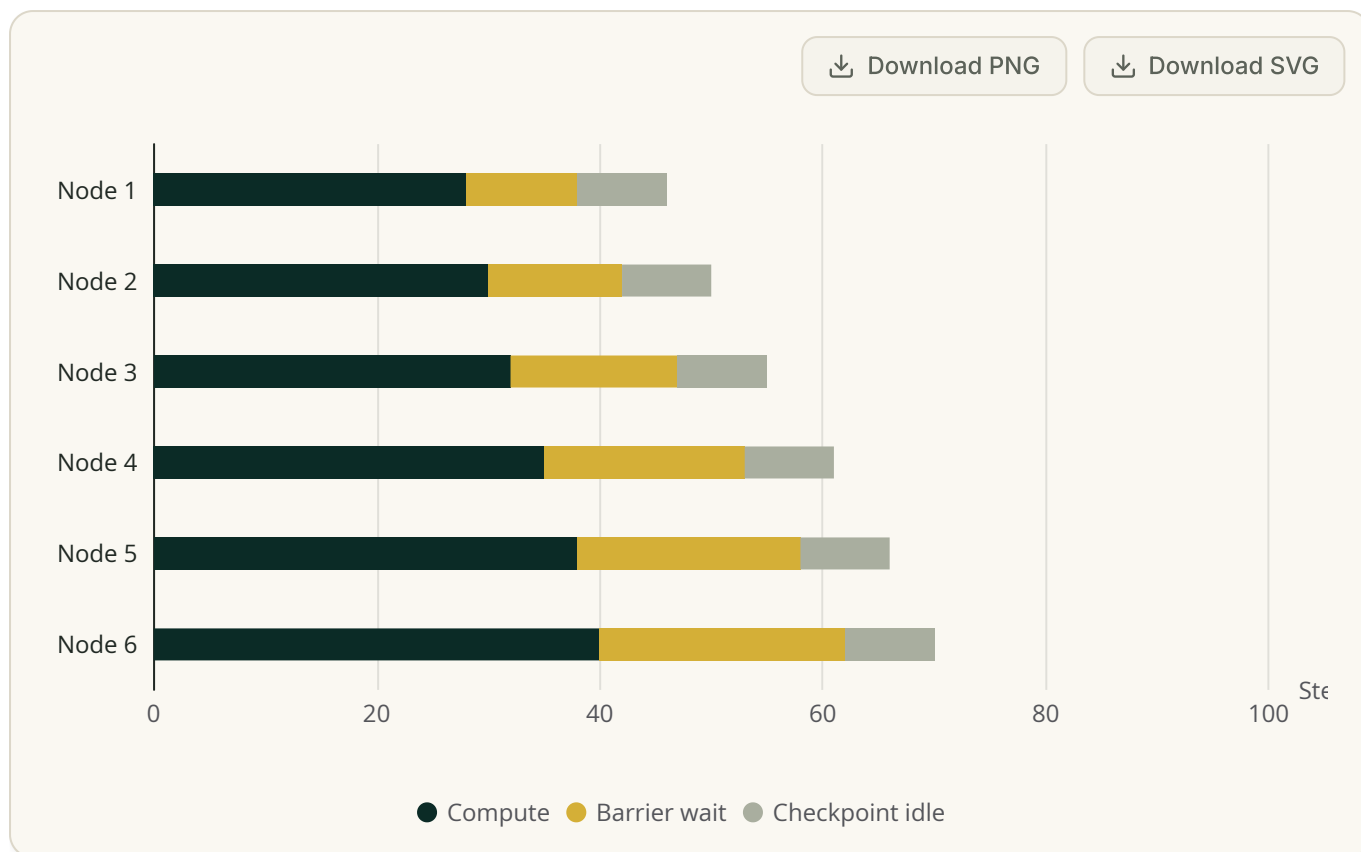
Queue lengths growing while aggregate cluster utilization stays moderate.

— WHAT IT COSTS

Reserved-but-unused resources that cannot be granted to other jobs — stranded capacity created by the scheduler's own allocation policy.

— WHY IT OCCURS

Developers size requests for worst-case peaks as insurance against OOM, and the scheduler has no incentive or mechanism to reclaim idle reservations.



Multi-node training step timeline: the green segment is actual compute, gold is barrier synchronization wait, and gray is checkpoint idle. Nodes with weaker cooling or slower fabrics become stragglers and hold back every step of the job. Values are illustrative and do not represent production measurements.

3. PSCI Measurement Methodology & Benchmarks

3.1. Key Metrics & Formulas

The PSCI formalizes stranded capacity as a measurable ratio between what is provisioned and what becomes productive compute.

$$\text{PSCI (\%)} = (\text{Provisioned capacity} - \text{Productive compute}) / \text{Provisioned capacity} \times 100$$

Each operational layer contributes a sub-index, so losses can be attributed and prioritized:

PSCI-facility: thermal and power capacity contracted but not delivered to IT loads.

PSCI-it: hardware capacity installed but not converted into FLOPs.

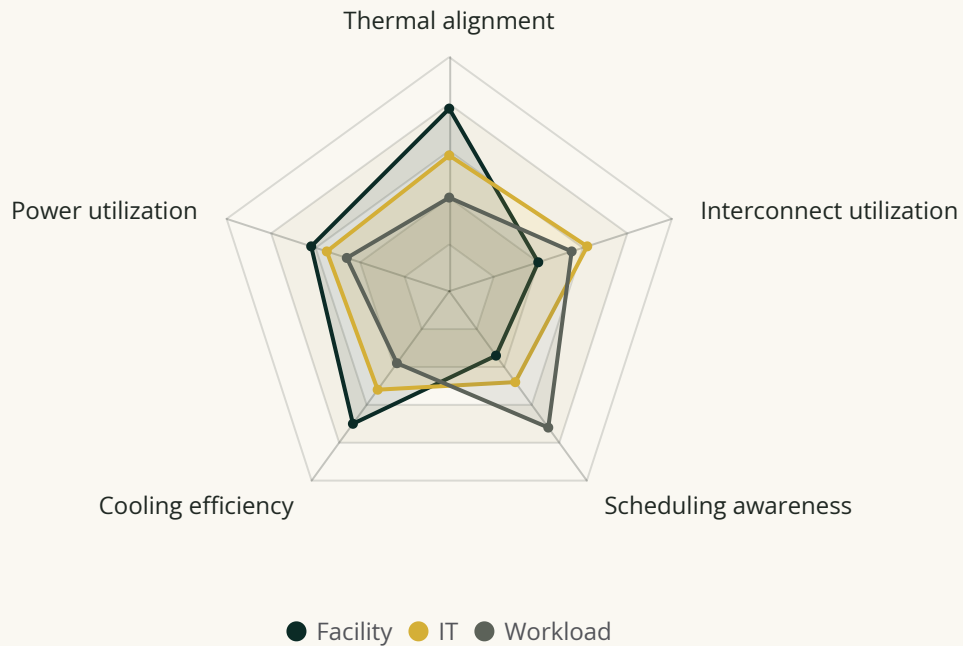
PSCI-workload: schedulable capacity lost to orchestration inefficiency.

MFU vs. PUE alignment. PUE measures facility energy efficiency but is blind to compute output; MFU measures what fraction of theoretical FLOPs a workload achieves but is blind to energy. PSCI relates both: a facility with excellent PUE and a workload with low MFU still strands the majority of its contracted compute. Operators should track PUE and MFU together, with PSCI as the single reconciliation metric.

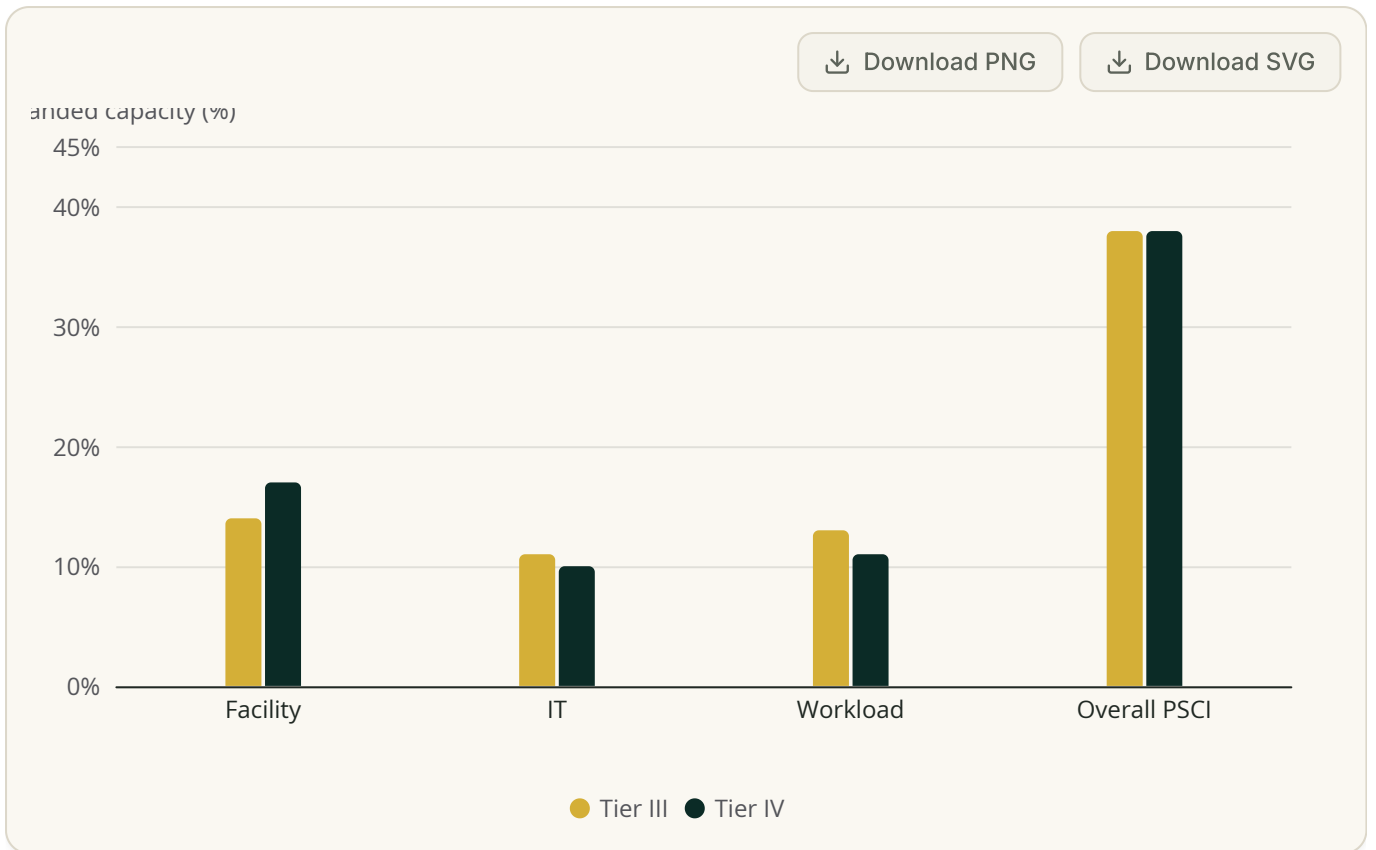
3.2. Baseline Hyperscale Benchmarks

Across surveyed hyperscale environments, typical stranded capacity ranges from **22% to 38%** of contracted capacity, depending on design maturity and operational tooling. The dominant loss layer varies by facility generation: older facilities lose more at the Facility layer, while newer, thermally efficient facilities shift the balance toward Workload losses.

Layer	Typical loss range	Dominant pattern	---	---	---	Facility	8–15%	Static Thermal Headroom, Power Distribution Stranding
IT	6–12%	Unbalanced Thermal Throttling, Power Cap Buffering	Workload	6–14%	Barrier Synchronization Idle, Over-Provisioned Allocation			

[Download PNG](#)[Download SVG](#)

Stranded capacity footprint across operating dimensions: a high score means high efficiency in that dimension. The Facility layer typically scores well on power delivery but poorly on scheduling alignment, while the Workload layer shows the inverse — most stranded capacity sits in the dimensions the operator does not instrument. Values are illustrative and do not represent production measurements.



PSCI benchmark range across Tier III and Tier IV data centers: Tier IV design redundancy adds resilience but also more provisioned capacity that can strand; operational tooling, not certification class, is the main differentiator. Bars show the midpoint of typical loss ranges. Values are illustrative and do not represent production measurements.

4. How to Cite This Document

4.1. Academic & BibTeX Formats

APA (7th edition):

PhysaFlow Engineering & Research Group. (2026). *PhysaFlow Stranded Capacity Index (PSCI): An operational taxonomy of unproductive power & compute in modern AI infrastructure*. PhysaFlow.

BibTeX:

BIBTEX

Copy

```
@techreport{physaflow2026psci,  
author    = {{PhysaFlow Engineering & Research Group}},  
title     = {PhysaFlow Stranded Capacity Index (PSCI): An Operational Taxonomy of Unproductive Power},  
year      = {2026},  
institution = {PhysaFlow},  
type      = {Technical report},  
note      = {Source: PhysaFlow Stranded Capacity Index (PSCI)}  
}
```

4.2. Media & Industry Citation Standards

Inline / web:

PhysaFlow Engineering & Research Group, *PhysaFlow Stranded Capacity Index (PSCI)*, PhysaFlow, 2026. Available: <https://dev.physaflow.com/reports/stranded-capacity-index>

IEEE:

P. Engineering and R. Group, *PhysaFlow Stranded Capacity Index (PSCI)*, PhysaFlow, Tech. Rep., 2026.

Usage note: Figures, charts, and downloadable graphics in this report must retain the attribution `Source: PhysaFlow Stranded Capacity Index (PSCI)` when reproduced in any form — academic, media, or industry material.

5. References & Key Literature

Uptime Institute. (2024). *Annual Data Center Industry Survey*. Uptime Institute Intelligence.

Uptime Institute. (2023). *Data Center Power and Cooling: Managing the Demand Frontier*.

IEEE. (2022). *IEEE 3000 Series: Power Systems Design Standards for Industrial and Commercial Facilities*.

ASHRAE. (2021). *Thermal Guidelines for Data Processing Environments (TC 9.9)*.

NVIDIA Corporation. (2023). *NVIDIA Architecture and Data Center Design Reference Architectures*.

EPRI. (2023). *Powering Intelligence: Data Center Load Growth and Grid Implications*.

Jouppi, N. P., et al. (2017). *In-Datcenter Performance Analysis of a Tensor Processing Unit* (ISCA). IEEE.

Patterson, D., et al. (2021). *Carbon and Energy Footprint of the Current Phase of Deep Learning Research and Development*. arXiv:2105.01226.

Chowdhury, M., et al. (2023). *Optimus: Hardware-Accelerated GPU Co-Scheduling*. arXiv.

PhysaFlow. (2026). *PhysaFlow Stranded Capacity Index (PSCI): Internal Field Measurements* (whitepaper).

Reference values, benchmarks, and loss ranges quoted in this report are illustrative aggregations unless a primary source is explicitly cited.