

Fragmentation Tax

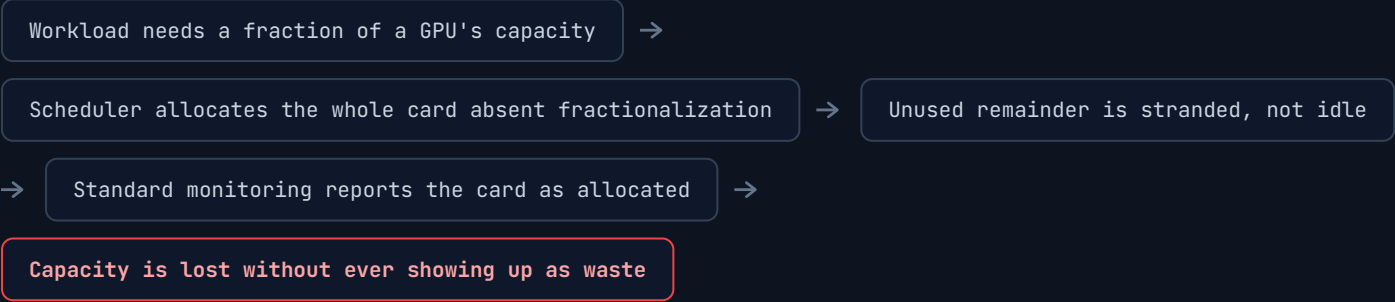
DEFINITION

The capacity permanently lost to allocation fragmentation across GPUs and nodes.

FRAMEWORK FLOW

01 The Condition A workload requires only a fraction of a GPU's compute or memory capacity, but the scheduler allocates the whole card — the default behavior absent fractionalization support.	02 The Boundary Sub-card workloads accumulate across the fleet, each individually justified, each stranding the unused remainder of its assigned card — the point where per-job allocation choices become fleet-wide waste.	03 Failure State A meaningful fraction of provisioned capacity sits inside allocations that are technically "in use" but only partially utilized — capacity that's neither idle nor productive at anything close to full density.	04 Consequence Standard monitoring reports the card as fully allocated; the stranded fraction is lost without ever showing up as waste.
--	--	--	--

Failure State: Whole-Card Allocation to Sub-Card Work



THE DIAGNOSTIC SIGNAL

ALLOCATION OWNERSHIP Ask whether each job's allocation matches its actual resource requirement. Almost never, by default — whole-card allocation is the scheduler's simplest behavior, not a considered choice.	COMPOSITE OWNERSHIP Ask who owns the stranded remainder once a sub-card job is placed on a whole card. Nobody does — it isn't idle, isn't allocated to anyone else, and isn't tracked as its own quantity.
TRIGGER CONDITION Any sub-card workload placed without fractionalization support triggers this; a fleet running only whole-card, fully-saturating jobs never accumulates a fragmentation tax.	REMEDIATION SHAPE Fractionalization support, density-aware scheduling, bin-packing sub-card workloads onto shared cards — architectural, not a monitoring dashboard change.

SCHEDULER MATURITY SPECTRUM

PRIMITIVE Whole-card allocation regardless of workload size, absent fractionalization support — fragmentation tax accumulates unchecked.	TRANSITIONAL Some manual bin-packing or fractionalization in limited cases; fragmentation tax reduced but not systematically managed.
DENSITY-AWARE Scheduler actively places sub-card workloads to maximize density; fragmentation tax is a monitored, managed quantity.	YIELD-OPTIMIZED Fragmentation tax is minimized by design; scheduling decisions are made against effective yield, not raw allocation.

ARCHITECTURAL RELATIONSHIPS

#89 Effective GPU Yield RELATED MODERATE Fragmentation Tax is one of three named yield discounts (alongside allocation drift and reservation overhang) that directly reduce Effective GPU Yield by stranding capacity inside partially utilized allocations.	#90 Capacity Illusion Index RELATED MODERATE Fragmentation is one of the mechanisms — alongside allocation drift and reservation overhang — that converts provisioned GPU capacity into the stranded, non-allocatable capacity the Capacity Illusion Index quantifies.
#91 Phantom Scarcity RELATED MODERATE Co-resident in the GPU Utilization Analyzer — Fragmentation Tax is the direct cause of Phantom Scarcity.	#92 Queue-Idle Paradox RELATED MODERATE Co-resident in the GPU Utilization Analyzer — fragmentation produces the Queue-Idle Paradox.
#94 Economic Density Loss RELATED MODERATE Co-resident in the GPU Utilization Analyzer — Fragmentation Tax is the mechanism; Economic Density Loss is its cost.	
ANCHOR /gpu-utilization-analyzer/ FRAMEWORK HOME Vendor and hardware-class agnostic — the whole-card-vs-sub-card allocation gap applies to any GPU fleet lacking fractionalization support, regardless of accelerator vendor.	

"Every allocated card is correctly counted as in use. Nobody counts the fraction of it that's actually doing work — that's the tax nobody sees on the invoice."