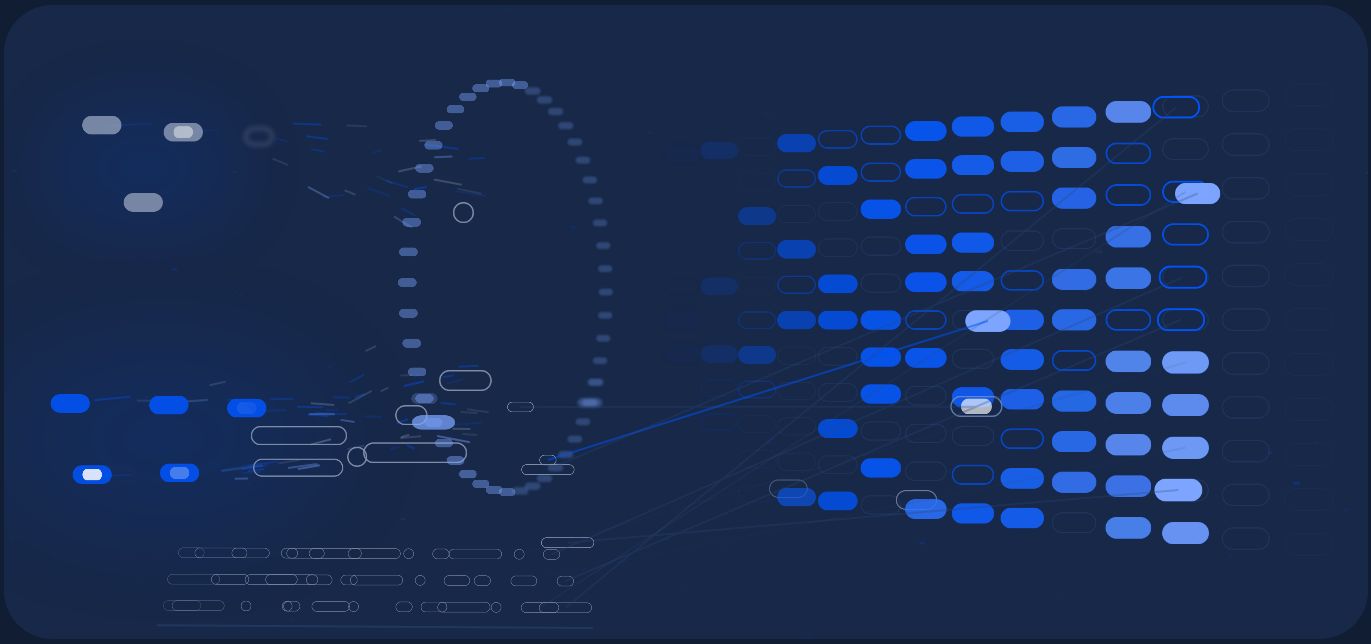
Rillor

RESEARCH REPORT

What research shows about training on spot GPUs

Systems research and provider documents show what spot GPU instances save on model training once lost work, checkpoint overhead and deadlines are counted.



Rillor

Contents

01 Summary

02 Background

03 Method

04 Findings

05 Implications

06 Limits of this analysis

07 Sources

Summary

- Spot and preemptible GPU capacity comes with short notice: two minutes at AWS and Oracle, 0 or 120 seconds plus up to 30 seconds of shutdown at Google, up to 30 seconds at Azure [2, 10, 14, 15, 17]. The interruption statistics the three large clouds publish cannot be compared [6, 11, 14].
- The peer-reviewed training results ran mostly on V100-era instances from 2021 to 2024. Under their own setups, Bamboo reported costs 2.4 times lower than on-demand [22], Parcae 3.24 times cheaper than on-demand [24], and Can't Be Late saved 41% to 48% on an ML training job whose spot discount was 67% [25].
- Lost work and restarts explain the gap between discount and saving. Checkpoint and restart took 77% of training time in Bamboo's GPT-2 test on 64 spot instances [22].
- Results depend on metric, network and checkpoint storage. Bamboo's paper reports 2.4 times, 3.6 times and 1.95 times on different measures [22], and Oobleck reversed Bamboo's ranking against Varuna on a faster network with NVMe storage [23].
- Under a deadline, savings follow the posted discount and the share of time spot capacity is available [25, 30]. A January 2026 preprint found median lifetimes from under 1 hour to over 20 hours by zone for a job on 16 single-H100 spot instances on Google Cloud that stops when any one is preempted, and checkpoint egress at up to 16% of total cost in multi-region fine-tuning [30].
- Dedicated capacity is interrupted too: Llama 3 405B pre-training on up to 16K H100 GPUs saw 419 unexpected interruptions in 54 days [32].

Background

Spot capacity is spare cloud capacity sold at a discount on the condition that the provider can take it back. Each provider sets its own terms.

AWS. Spot Instances run on spare EC2 capacity "for steep discounts in exchange for returning them when Amazon EC2 needs the capacity back" [1]. AWS interrupts when it needs the capacity, when the Spot price rises above a maximum price the customer set, or when a launch or zone constraint can no longer be met, and setting a maximum price makes interruptions more frequent [1]. AWS advises using checkpoints, notes that instance store data is lost on stop or termination, and says an instance can be interrupted before any warning is available [4]. It does not charge for an interruption it causes in the first hour, and charges for the seconds used when it interrupts after the first hour [5]. The Spot Instance Advisor page states savings of up to 90% against On-Demand [6].

Google Cloud. Spot VMs are offered at "up to 91% discounts for many machine types, GPUs, TPUs, and Local SSDs" against standard on-demand prices [10]. They have no minimum or maximum runtime, are excluded from the Compute Engine SLA and are not offered on the A4X machine series or bare metal [10]. Prices can change up to once every day [10]. Spot VMs with GPUs are preempted by default during maintenance and cannot restart automatically, while standard GPU VMs get one hour of notice [10]. Preemption within one minute of creation is not billed [10]. Older preemptible VMs always stop after 24 hours [12]. Flex-start VMs, a third model, run uninterrupted for up to seven days once granted, and Google lists model fine-tuning among their uses [13].

Microsoft Azure. Azure offers Spot VMs with no SLA and evicts them when it needs the capacity or when the price rises above the customer's maximum, unless that maximum is set to -1 [14]. The default Deallocate policy keeps the disks, which count against quota and are charged, with no guarantee of reallocation [14]. The product page states discounts of up to 90% against pay-as-you-go prices [16].

Oracle and a GPU marketplace. Oracle's preemptible capacity "costs 50% less than on-demand capacity in all regions" and, Oracle says, is not suitable for long-running workloads [17]. Its supported GPU shapes are the VM.GPU2 and VM.GPU3 series [17], P100 and V100 shapes [18]. Vast.ai sells interruptible instances on bids, "often 50%+ cheaper than on-demand", paused when outbid with data kept, and its guide pairs multi-day training with reserved instances [19].

Method

What was examined. Published evidence on the net cost of training on spot or preemptible GPUs once interruptions are counted, with the hardware, cloud, year, model, trace and metric behind every figure.

Sources. Provider documentation from AWS [1, 2, 3, 4, 5, 6, 7, 8, 9], Google Cloud [10, 11, 12, 13], Microsoft [14, 15, 16], Oracle [17, 18] and Vast.ai [19], each viewed on 24 October 2026. Peer-reviewed systems papers: CheckFreq at FAST 2021 [20], Varuna at EuroSys 2022 [21], Bamboo at NSDI 2023 [22], Oobleck at SOSP 2023 [23], Parcae at NSDI 2024 [24], Can't Be Late at NSDI 2024 [25], SpotServe at ASPLOS 2024 [26], SkyServe at EuroSys 2025 [27], a Meta reliability study at HPCA 2025 [28] and Ding-Dong Ditch, accepted at IEEE CLOUD 2026 [29]. Preprints and technical reports: SkyNomad [30], RLBoost [31], the Llama 3 report [32] and MegaScale [33]. Two posts by the teams involved: PyTorch on asynchronous checkpointing [34] and the Vicuna team on fine-tuning cost [35].

How. Papers were read in full text and each figure taken from the passage or table cited. Results stay under their own setups and are not averaged. Notice periods come from provider documents only, since several papers state them differently (Finding 8). Discount ceilings are provider claims. Figures marked "derived" are arithmetic on a source's numbers.

Findings

1. Notice periods are short and best effort

Provider	Notice	Delivery	After the notice
AWS Spot	Two minutes before stop or termination [2]	EventBridge and instance metadata, check every 5 seconds [2]	Terminate, stop or hibernate [1]. Hibernation starts at once, with no two-minute warning [2]
Google Cloud Spot VMs	0 seconds (default) or 120 seconds [10]	Metadata, then an ACPI soft-off signal [10]	Up to 30 seconds of best-effort shutdown, then stop (default) or delete [10]
Azure Spot	Up to 30 seconds [14, 15]	Scheduled Events, poll once per second [15]	Deallocate (default) or delete [14]
Oracle preemptible	Two minutes [17]	An instance preemption event [17]	Terminated [17]
Vast.ai interruptible	No period stated [19]	Not stated [19]	Paused, data kept, resumes when priority returns [19]

AWS and Azure describe their notices as best effort, and Google its shutdown period [2, 10, 15]. AWS adds that an instance can be interrupted before a warning is available [4], and that a rebalance recommendation cannot always be sent before the two-minute notice [3].

Little fits in these windows. SpotServe needed about 2 minutes to launch and initialize its inference engine [26]. SkyServe's authors write that finding, provisioning and loading a replacement typically takes longer than the warnings [27]. Bamboo does not use the grace period, because its length varies across clouds [22]. In Oobleck's spot tests, Varuna fell back because it could not finish checkpointing within the grace period [23]. An asynchronous checkpoint is safe only after its background write to storage finishes [34].

2. Interruption statistics measure different things

Provider	Metric	Window
AWS	Frequency band per instance pool: under 5%, 5% to 10%, 10% to 15%, 15% to 20%, over 20% [6, 7]	Last 30 days, as the page states [6]

Provider	Metric	Window
Azure	Probability of eviction within the next hour [14]	Last 7 days in the portal, 28 days of eviction rates and 90 days of prices through Azure Resource Graph [14]
Google Cloud	Spot VMs preempted in a day divided by all Spot VMs that stopped that day, including user stops [11]	Daily rates over the past 30 days, prices over the past year [11]

Google's metric is in Preview, is not available for N1 machine types with GPUs, and counts user stops in its denominator, so 0.50 means half of the Spot VMs that stopped that day were preempted [11].

AWS's advisor draws on a public data file. The copy viewed was last modified on 20 March 2026, so the "last 30 days" label could not be confirmed for it. Its Linux entries for 8-GPU instances read [7, 8]:

Instance	GPUs	us-east-1	us-east-2	us-west-2
p5.48xlarge	8 x H100	Under 5%, 57% savings	10% to 15%, 74%	Under 5%, 69%
p5en.48xlarge	8 x H200	Under 5%, 67%	5% to 10%, 76%	Under 5%, 70%
p4d.24xlarge	8 x A100	15% to 20%, 56%	Over 20%, 33%	Over 20%, 45%
p6-b200.48xlarge	8 x B200	10% to 15%, 79%	5% to 10%, 82%	10% to 15%, 82%

The page says average interruption frequency "has historically been <5%" [6], while the file's global field reads "<10%" [7]. AWS notes that the rate a workload sees depends on capacity at the time [6].

Research traces use their own measures. Several show interruptions arriving in batches within a zone or region.

- In Bamboo's 24-hour trace of a 48-node EC2 cluster, hourly preemptions ranged from none to 16 nodes (33%) and averaged 8% to 12%. In a separate 24-hour trace of 64 EC2 spot instances, only 7 of 127 preemption timestamps spanned more than one zone [22].
- Oobleck replayed traces from EC2 P3 and Google Cloud a2-highgpu-1g spot instances in which preemption events came every 7.7 minutes (EC2) and 10.3 minutes (Google Cloud) on average [23].
- Can't Be Late found that in clusters of 4 and 16 instances, all or none were available over 94% of the time, and periods of unavailability lasted hours or days [25].

- Ding-Dong Ditch recorded 2,635 interruption events in six 24-hour experiments on AWS and Azure between December 2024 and March 2026, with 92.9% of co-interruptions within 3 minutes. Its instance types are not stated to be GPU types [29].
- SkyServe found spot GPUs unavailable across all zones of a region 33.1% of the time in one AWS trace, with preemptions correlated within a region and much less across regions [27].
- SkyNomad found simultaneous preemptions across regions rare [30].

Varuna found single-GPU low-priority VMs on Azure easier to obtain than 4-GPU VMs [21].

3. The cost of lost work and checkpoints

Three costs separate a spot job from an uninterrupted one: work done since the last checkpoint, the time to restart, and the overhead of taking checkpoints.

Lost work. CheckFreq (FAST 2021) tested seven models on 8-GPU V100 and GTX 1080Ti servers [20]. Training ResNext101 on ImageNet-1K on one V100 takes 270 hours, about 3.9 hours per epoch, so a checkpoint at each epoch boundary loses about two hours of GPU computation per interruption on average [20]. CheckFreq checkpoints at the level of iterations, in a snapshot and a persist phase, and cut recovery from hours to seconds while keeping runtime overhead within 3.5% [20].

Restart and rollback at scale. Bamboo's authors trained GPT-2 over 64 p3.2xlarge spot instances with a checkpoint-and-restart approach. Restarting, adapting checkpoints to new pipeline configurations and redoing work took 77% of training time, and preemptions often arrived during checkpointing, forcing a rollback to the checkpoint before [22].

Changeover time. Can't Be Late measured a changeover of 4 minutes of provisioning, 5 of setup and 9 of lost progress, about 0.3 hours, for its training job on a p3.2xlarge [25]. SkyNomad measured a cold start of about 6 minutes, including checkpoint loading, for jobs with 100 GB and 500 GB checkpoints [30].

Checkpoint size and write time. Llama 3's checkpoints held 1 MB to 4 GB of model state per GPU [32]. PyTorch's distributed asynchronous checkpointing cut a 7B model's checkpoint downtime from an average of 148.8 seconds to 6.3 seconds [34]. The visible pause, copying from GPU to CPU, took 6 to 14 seconds for 7B to 13B models, and the write to disk follows in the background [34]. MegaScale uses the same two-stage design [33].

Checkpoint interval. Meta's reliability study (HPCA 2025) restates a classic result by Daly and Young: under some limiting assumptions, the optimal checkpoint interval is about the square root of 2 times the checkpoint write time times the mean time to failure [28]. The study's cluster parameters include a write overhead of about 5 minutes [28].

Two illustrations show the range (derived, using that formula). With a 5-minute write and the 7.9-hour mean time to failure Meta measured for 1,024-GPU jobs [28], the interval is about 69 minutes. With a 5-minute write and a preemption event every 7.7 minutes, as in Oobleck's EC2 trace [23], the formula gives about 9 minutes,

longer than the gap between failures. The formula's assumptions fail there, and checkpoint and restart spends most of its time writing and rolling back, as Bamboo's 77% measurement shows [22].

Meta estimates that an effective training time ratio of 0.9 on runs of about 12,000 GPUs needs a checkpoint write overhead of about 10 seconds, or a failure rate improved from 6.50 to about 1 per thousand node-days [28].

What is billed. AWS charges for the seconds used when it interrupts an instance after the first hour, so lost work after the first hour is paid for [5]. Google waives charges only for Spot VMs preempted within their first minute [10]. SageMaker managed spot training computes its savings as one minus billable time over training time, where training time excludes intervals lost to an interruption in their first hour of runtime and time spent re-provisioning or waiting for capacity [9]. The figure is the spot discount on retained productive time [9].

4. Redundancy and fast recovery in distributed training

Four papers reduce or replace checkpoint restarts.

Varuna (EuroSys 2022). Varuna trained on Azure low-priority VMs, NC24_v3 or NC6_v3 with V100 16 GB GPUs and 10 Gbps Ethernet, using up to 300 GPUs, and compared them with a cluster of 16 DGX-2 nodes connected by 200 Gbps InfiniBand [21]. On GPT-2 8.3B it reached 0.56 examples per second per GPU on spot VMs against Megatron's 0.48 on the DGX-2 cluster, on VMs about 5 times cheaper, for 5.85 times better cost-performance [21]. The comparison is against dedicated hardware. It does not use on-demand cloud prices.

Bamboo (NSDI 2023). Bamboo adds redundant computation, placed in pipeline bubbles, so that a node can take over the work of a preempted neighbor [22]. It ran on EC2 p3 spot clusters of single-V100 instances, with on-demand at $3.06 per GPU-hour and spot at $0.918 "at the time of our experiments" [22]. The authors replayed 24-hour preemption traces at 10%, 16% and 33% hourly preemption rates across six models [22]. The abstract reports 3.7 times the throughput of checkpointing and costs 2.4 times lower than on-demand [22]. The introduction reports a 3.6 times cost reduction, and the evaluation 1.95 times higher value, where value is throughput per dollar per hour [22]. Its throughput at the 10% rate averaged 15% below on-demand, and reconfiguration took an average of 7% of training time [22].

Bamboo's Table 2 for GPT-2 shows how the result changes with preemption rate [22].

GPT-2 configuration	Time (hours)	Cost ($ per hour)	Total cost (derived)
On-demand, single-GPU instances	4.63	97.92	About $453
Bamboo on spot, 10% hourly preemption	4.64	42.23	About $196

GPT-2 configuration	Time (hours)	Cost ($ per hour)	Total cost (derived)
Bamboo on spot, 16%	6.12	40.39	About $247
Bamboo on spot, 33%	10.08	36.72	About $370

At 33% the run took more than twice as long. On BERT, Bamboo outperformed Varuna by 2.5 and 2.7 times in throughput at the 10% and 16% rates, and Varuna hung at 33% [22].

Oobleck (SOSP 2023). Oobleck builds a set of pipeline templates and runs at least f + 1 logically equivalent pipeline replicas to tolerate any f simultaneous failures [23]. It ran on 30 NVIDIA A40 GPUs, one per node, connected by 200 Gbps InfiniBand, with Varuna checkpointing to a MinIO object store on NVMe [23]. The abstract reports up to 29.6 times the throughput of Bamboo and Varuna [23]. In the paper's Table 2, that figure is GPT-3 6.7B with a failure every 10 minutes, where Oobleck reached 3.55 samples per second and Varuna 0.12. Bamboo could not run any GPT-3 model because of lack of memory [23]. With a failure every 6 hours the two were close, 4.33 against 4.02 [23]. On replayed spot traces, Oobleck outperformed the others by up to 9.1 times on average, and Varuna made no training progress on GPT-3 6.7B [23].

Parcae (NSDI 2024). Parcae adjusts parallelization ahead of predicted preemptions and optimizes "liveput", the expected throughput across possible preemption scenarios [24]. It replayed segments of a 12-hour trace from a 32-instance AWS p3.2xlarge cluster on 32 V100 instances [24]. On average it delivered 2.59 times the throughput of Varuna and 3.0 times that of Bamboo, and was 3.24 times cheaper than on-demand instances [24].

Three of the four ran on V100 instances and one on A40s in a private cluster. None used H100 or later GPUs.

5. Deadlines and fallback to on-demand

A job with a deadline has to switch to on-demand when spot capacity stays away too long. Two papers model this.

Can't Be Late (NSDI 2024). The authors collected spot availability traces every 10 minutes over three months in nine AWS availability zones for V100 and K80 GPU instances [25]. Availability could change from 100% to 0% within hours, and unavailability could last hours or days [25]. Prices varied about 5% on average over any week, with a spot V100 at about $1 per hour [25]. Their policy, Uniform Progress, uses spot capacity when it is available and switches to on-demand when the job falls behind a uniform pace toward its deadline [25].

For the ML training workload, a RoBERTa pre-training job on one p3.2xlarge in us-west-2b with 72 hours of computation, spot was priced at $0.92 per hour, 67% below on-demand [25]. The job checkpointed twice per epoch to S3 [25]. Against $233.5 on on-demand only, the policy cost $138.2 with a tight deadline of 84 hours (41% less) and $122.0 with a loose deadline of 100 hours (48% less) [25]. On the loose deadline a

greedy policy saved 15% and an optimal one 55% [25]. The 84% saving in the abstract belongs to a CPU bioinformatics job with a 91% spot discount, and a CPU analytics job with a 55% discount saved 27% and 32% [25].

SkyNomad (preprint, January 2026). SkyNomad uses spot capacity across regions for GPU batch jobs with deadlines [30]. Its traces cover 16 a3-highgpu-1g instances (1 x H100 each) across 13 Google Cloud zones, for a gang-scheduled job that is preempted if any instance is [30]. Median lifetimes ranged from under 1 hour in europe-west1-c to over 20 hours in asia-south2-b [30].

Its deployments fine-tuned Qwen3-4B on AWS 4 x L4 and 4 x A10G instances and Qwen3-14B on 8 x A100, each about 30 hours of compute with a 45-hour deadline and checkpoints of 100 GB and 500 GB [30]. SkyNomad saved 10% to 55% against the "best single-region baseline" and 47% to 69% against the average single region [30]. These figures include its overheads: egress for moving checkpoints between regions made up to 16% of total cost [30]. Moving 500 GB of checkpoints cost roughly $10 from the US to Europe and about $40 from Asia to the US [30]. The weakest region had no spot capacity for more than 70% of the time, and on-demand fallback drove 74% of its total cost [30].

Provider tools. SageMaker managed spot training does not fall back to on-demand. Without spot capacity the job waits until its maximum wait time and stops [9].

Fine-tuning on spot instances. Evidence specific to fine-tuning is thin. The Vicuna team reported in March 2023 that training on 8 A100 GPUs with managed spot instances, auto-recovery and zone switching cut the cost of its 7B model from $500 to around $140 and its 13B model from around $1K to $300 [35]. The post gives no count of interruptions or lost work. Google's Flex-start VMs offer a different trade: up to seven days uninterrupted once granted, with requests that fail when resources are unavailable [13].

6. Serving and rollout on spot capacity

SkyServe's authors note that training can pause and resume from checkpoints to meet a deadline, while serving must stay available [27].

SpotServe (ASPLOS 2024) served OPT-6.7B, GPT-20B and LLaMA-30B on g4dn.12xlarge instances (4 x T4), replaying segments of a 12-hour AWS g4dn spot trace [26]. It cut P99 latency 2.4 to 9.1 times against existing serving systems [26]. On GPT-20B, against on-demand at 3.9 USD per hour and spot at 1.9 USD per hour, it saved up to 54% of cost, with average latency up to 18% higher and P99 latency up to 90% higher than on-demand [26].

SkyServe (EuroSys 2025) spreads spot replicas across regions and clouds and falls back to on-demand [27]. It cut cost by 43% on average against on-demand replicas [27]. A pure spot deployment through an AWS Autoscaling Group saw 49.4% of requests fail or time out [27].

RLBoost (preprint, v3 April 2026) runs reinforcement learning rollout, which its authors call stateless and embarrassingly parallel, on spot 2 x H100 instances, and keeps training on reserved 8 x H100 instances [31]. With Qwen3 8B, 14B and 32B models, it raised training throughput 1.51 to 1.97 times and cost efficiency, measured as tokens trained per budget, by 28% to 49% against on-demand only [31]. Its prices are averages across providers and regions: $83.79 per hour for on-demand 8 x H100 and $5.32 per hour for spot 2 x H100 [31]. Preemptions were replayed from Bamboo's trace [31], which came from EC2 p3 (V100) spot instances [22]. With a spot discount of about 30%, RLBoost still improved cost efficiency [31].

7. Interruptions without spot: a large-run baseline

Llama 3. Meta trained Llama 3 405B on up to 16K H100 GPUs with 80 GB of HBM3 each [32]. During a 54-day snapshot of pre-training there were 466 job interruptions: 47 planned and 419 unexpected [32]. About 78% of the unexpected interruptions were attributed to confirmed or suspected hardware issues, and GPU issues accounted for 58.7% [32]. Significant manual intervention was needed only three times, and Meta reported more than 90% effective training time [32]. Automated cluster maintenance, such as firmware and kernel upgrades, caused at least one interruption daily [32]. The 419 unexpected interruptions equal about 7.8 a day, or one about every 3.1 hours (derived).

Meta's research clusters. Across 11 months, 4 million jobs and over 150 million A100 GPU hours, Meta found mean time to failure of 7.9 hours for 1,024-GPU jobs against 47.7 days for 8-GPU jobs, and projected 1.8 hours for 16,384-GPU jobs and 0.23 hours for 131,072-GPU jobs [28]. Hourly checkpoints were typical for larger jobs, giving an average of half an hour of lost work per failure [28].

MegaScale. Its authors report a production run of a model with hundreds of billions of parameters over several weeks that recovered from failures over 100 times [33]. After a crash, training caught up to its earlier progress within 15 minutes from the latest checkpoints, keeping effective training time above 90% [33].

At these scales a job already restarts several times a day, before any spot reclaims, which arrive in batches (Finding 2).

8. Where the sources disagree

Notice periods. SpotServe says AWS gives a 30-second grace period [26]. AWS documents two minutes [2]. SkyServe says 30 seconds on Google Cloud [27]. Google documents 0 or 120 seconds of notice, then up to 30 seconds of shutdown [10], and the 30-second figure matches the shutdown period of legacy preemptible VMs [12]. CheckFreq cites GPU VMs preempted at least every 24 hours on Google Cloud [20], which describes the 24-hour limit of preemptible VMs [12]. Spot VMs have no maximum runtime [10].

Price terms. Can't Be Late reports that Google Cloud spot prices change only once every 30 days [25]. Google's current page says they can change up to once every day [10]. When the terms changed is not dated in the sources reviewed.

Bamboo's headline. Besides the figures in Finding 4, the paper's introduction reports 2.48 times higher value in simulation [22]. Each figure measures something different.

Bamboo against Varuna. On BERT, Bamboo found itself 2.5 to 2.7 times faster than Varuna [22]. Oobleck found Bamboo slower than Varuna and called the gap surprising. Its authors attributed it to their 200 Gbps network and NVMe storage against Bamboo's p3 instances with up to 10 Gbps networking and S3, and to Bamboo's evaluation giving Varuna the same batch configuration as Bamboo, which throttled Varuna's throughput [23].

Size of the discount. The papers report the spot discount of their own time and hardware: preemptible VMs 6 to 8 times cheaper (CheckFreq, 2021) [20], low-priority VMs about 5 times cheaper (Varuna, 2021) [21], spot at about 30% of on-demand on EC2 (Bamboo, 2023) [22], V100 spot about 3 times cheaper than on-demand on AWS (Can't Be Late, 2024) [25], spot at 8% to 50% of on-demand (SkyServe, October 2024) [27] and 3 to 10 times lower cost (SkyNomad, 2026) [30]. The provider ceilings are up to 90% at AWS [6] and Azure [16], up to 91% at Google [10] and a fixed 50% at Oracle [17].

Implications

For ML engineers planning a training run. The studies found net savings where the posted discount was large against lost and repeated work [25], checkpoints wrote and restored quickly on fast storage [20, 23], the deadline left slack [25], capacity came from several zones or regions [27, 30] and stateless work ran apart from stateful training [31]. Where preemption was frequent and checkpoints slow, checkpoint and restart consumed most of the run [22]. The Daly and Young interval links checkpoint write time to the expected interruption interval [28].

For platform and capacity leads. The interruption figures providers publish are not interchangeable, and Google's excludes N1 GPU VMs [6, 11, 14]. Interruptions arrive in batches within a zone [22, 25, 29], so a gang-scheduled job loses several nodes at once. Cross-region placement reduced correlated preemption in two studies [27, 30], at an egress cost of up to 16% of the total in one of them [30]. Notice periods of 0 seconds (Google's default) to two minutes are shorter than the restart times measured in the papers, which run from about 6 to 18 minutes [10, 25, 30].

For finance and procurement teams reading savings claims. A provider's discount ceiling, a tool's reported savings and a job's net saving are different numbers. SageMaker's formula leaves out lost first-hour intervals and capacity waits [9]. Can't Be Late's ML training job saved 41% to 48% on a 67% discount [25]. A complete comparison counts lost compute that is billed [5], fallback capacity [25, 30], egress [30] and the time added to the run [22]. Apart from Varuna's comparison with a dedicated DGX-2 cluster [21], the studies reviewed compare spot with on-demand prices, never with reserved or committed cloud prices.

Rillor's GPU and neocloud pricing method keeps spot and preemptible prices apart from on-demand and reserved prices, and each observation in its schema carries the GPU class, region, term, source and UTC capture time.

Limits of this analysis

- Most peer-reviewed training results ran on V100 instances [20, 21, 22, 24, 25], with T4 for serving [26] and A40 in a private cluster [23]. The H100 evidence is an availability trace in a preprint [30] and H100 instances driven by an older replayed trace [31].
- No study reviewed here measures net savings for multi-node pre-training on H100, H200, B200 or GB200 spot capacity on a public cloud.
- The interruption figures providers publish cannot be converted into one rate, and this report measured no interruption rate for any GPU pool.
- The AWS advisor data file viewed was last modified on 20 March 2026 [7]. Whether the advisor page renders that file today could not be confirmed.
- No source reviewed quantifies how often interruptions arrive without the documented notice.
- Fine-tuning evidence is limited to one team post [35] and one preprint [30]. The closest single-GPU training job, in Can't Be Late, is RoBERTa pre-training [25].
- The Daly and Young formula is taken as restated by the Meta study [28]. The original papers were not reviewed.
- Discount ceilings are provider claims. Current spot prices are outside this report's scope.
- Egress costs come only from SkyNomad's January 2026 figures [30].

Sources

1. Spot Instance interruptions. Amazon EC2 User Guide. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-interruptions.html

2. Spot Instance interruption notices. Amazon EC2 User Guide. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-instance-termination-notices.html

3. EC2 instance rebalance recommendations. Amazon EC2 User Guide. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/rebalance-recommendations.html

4. Prepare for Spot Instance interruptions. Amazon EC2 User Guide. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/prepare-for-interruptions.html

5. Billing for interrupted Spot Instances. Amazon EC2 User Guide. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/billing-for-interrupted-spot-instances.html

6. Spot Instance Advisor. Amazon Web Services, viewed 24 October 2026. https://aws.amazon.com/ec2/spot/instance-advisor/

7. Spot Instance Advisor data file (spot-advisor-data.json). Amazon Web Services, last modified 20 March 2026. https://spot-bid-advisor.s3.amazonaws.com/spot-advisor-data.json

8. Specifications for Amazon EC2 accelerated computing instances. Amazon EC2 Instance Types. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/ec2/latest/instancetypes/ac.html

9. Managed Spot Training in Amazon SageMaker AI. Amazon SageMaker Developer Guide. Amazon Web Services, viewed 24 October 2026. https://docs.aws.amazon.com/sagemaker/latest/dg/model-managed-spot-training.html

10. Spot VMs. Compute Engine documentation. Google Cloud, last updated 22 October 2026. https://docs.cloud.google.com/compute/docs/instances/spot

11. View the preemption rate and pricing for Spot VMs (Preview). Compute Engine documentation. Google Cloud, last updated 22 October 2026. https://docs.cloud.google.com/compute/docs/instances/view-spot-preemption-price

12. Preemptible VM instances. Compute Engine documentation. Google Cloud, last updated 22 October 2026. https://docs.cloud.google.com/compute/docs/instances/preemptible

13. About Flex-start VMs. Compute Engine documentation. Google Cloud, last updated 22 October 2026. https://docs.cloud.google.com/compute/docs/instances/about-flex-start-vms

14. About Azure Spot Virtual Machines. Microsoft Learn. Microsoft, viewed 24 October 2026. https://learn.microsoft.com/en-us/azure/virtual-machines/spot-vms

15. Scheduled Events for Linux VMs in Azure. Microsoft Learn. Microsoft, viewed 24 October 2026. https://learn.microsoft.com/en-us/azure/virtual-machines/linux/scheduled-events

16. Azure Spot Virtual Machines. Microsoft Azure, viewed 24 October 2026. https://azure.microsoft.com/en-us/products/virtual-machines/spot

17. Preemptible Instances. Oracle Cloud Infrastructure Documentation. Oracle, viewed 24 October 2026. https://docs.oracle.com/en-us/iaas/Content/Compute/Concepts/preemptible.htm

18. Compute Shapes. Oracle Cloud Infrastructure Documentation. Oracle, viewed 24 October 2026. https://docs.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm

19. Instance Types. Vast.ai Documentation. Vast.ai, viewed 24 October 2026. https://docs.vast.ai/guides/instances/choosing/instance-types

20. Mohan, Phanishayee and Chidambaram. CheckFreq: Frequent, Fine-Grained DNN Checkpointing. 19th USENIX Conference on File and Storage Technologies (FAST 21). USENIX, February 2021. https://www.usenix.org/system/files/fast21-mohan.pdf

21. Athlur et al. Varuna: Scalable, Low-cost Training of Massive Deep Learning Models. EuroSys 2022, arXiv 2111.04007 v2. arXiv, 15 November 2021. https://arxiv.org/abs/2111.04007

22. Thorpe et al. Bamboo: Making Preemptible Instances Resilient for Affordable Training of Large DNNs. 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23). USENIX, April 2023. https://www.usenix.org/system/files/nsdi23-thorpe.pdf

23. Jang et al. Oobleck: Resilient Distributed Training of Large Models Using Pipeline Templates. ACM SOSP 2023, arXiv 2309.08125 v2. arXiv, 7 November 2023. https://arxiv.org/abs/2309.08125

24. Duan et al. Parcae: Proactive, Liveput-Optimized DNN Training on Preemptible Instances. NSDI 2024, arXiv 2403.14097. arXiv, 21 March 2024. https://arxiv.org/abs/2403.14097

25. Wu et al. Can't Be Late: Optimizing Spot Instance Savings under Deadlines. 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24). USENIX, April 2024. https://www.usenix.org/system/files/nsdi24-wu-zhanghao.pdf

26. Miao et al. SpotServe: Serving Generative Large Language Models on Preemptible Instances. ASPLOS 2024, arXiv 2311.15566. arXiv, 27 November 2023. https://arxiv.org/abs/2311.15566

27. Mao et al. SkyServe: Serving AI Models across Regions and Clouds with Spot Instances. EuroSys 2025, arXiv 2411.01438 v2. arXiv, 3 March 2025. https://arxiv.org/abs/2411.01438

28. Kokolis et al. Revisiting Reliability in Large-Scale Machine Learning Research Clusters. 2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA), arXiv 2410.21680 v2. arXiv, 6 February 2025. https://arxiv.org/abs/2410.21680

29. Kim et al. Ding-Dong Ditch: Peeking Into Spot Instance Availability. Accepted at IEEE CLOUD 2026, arXiv 2604.16457 v2. arXiv, 27 May 2026. https://arxiv.org/abs/2604.16457

30. Li et al. SkyNomad: On Using Multi-Region Spot Instances to Minimize AI Batch Job Cost. arXiv 2601.06520. arXiv, 10 January 2026. https://arxiv.org/abs/2601.06520

31. Wu et al. RLBoost: Harvesting Preemptible Resources for Cost-Efficient Reinforcement Learning on LLMs. arXiv 2510.19225 v3. arXiv, 8 April 2026. https://arxiv.org/abs/2510.19225

32. Llama Team, AI @ Meta. The Llama 3 Herd of Models. arXiv 2407.21783 v3. arXiv, 23 November 2024. https://arxiv.org/abs/2407.21783

33. Jiang et al. MegaScale: Scaling Large Language Model Training to More Than 10,000 GPUs. arXiv 2402.15627. arXiv, 23 February 2024. https://arxiv.org/abs/2402.15627

34. Pasqualin et al. Reducing Model Checkpointing Times by Over 10x with PyTorch Distributed Asynchronous Checkpointing. PyTorch, 12 June 2024. https://pytorch.org/blog/reducing-checkpointing-times/

35. The Vicuna Team. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. LMSYS Org, 30 March 2023. https://lmsys.org/blog/2023-03-30-vicuna/

About this report

Published by Rillor on 7 November 2026. Superintelligence (SI) tools assist research and drafting, and every figure cites its source. Rillor builds and secures SI systems, supplies the data they learn from and delivers the infrastructure they run on. We apply SI to financial and compute markets.

Cite as: Rillor. What research shows about training on spot GPUs. 7 November 2026. https://rillor.com/insights/spot-gpu-training-research

Notices

Rillor is not a registered investment adviser or commodity trading advisor and does not provide investment or trading advice. Research, forecasts, data and software described on this site are for research and engineering use. Nothing here is an offer or recommendation to buy or sell any security, commodity interest or digital asset. Past or simulated results do not indicate future results.

NVIDIA, AMD, Ceph, Kubernetes and related product names are trademarks of their owners and are used only to identify products. No affiliation or endorsement is implied.

Work with Rillor

Send the GPU classes, regions and pricing types you need, and how often you need them. Contact us at rillor.com/contact?type=index or contact@rillor.com. More on GPU and neocloud pricing: rillor.com/markets/gpu-pricing