

The Price of a Sorted Terabyte

A unit-cost model for migrated batch on AWS, Azure and Google Cloud, and where the money actually goes

Bill Ahlbrandt

President and CEO, Ahlbrandt Software

Abstract

When people build the business case for moving batch off the mainframe, they usually price the new platform in servers per month. I wanted to price it per unit of work: what it costs to sort one terabyte, once a night, on Amazon Web Services, Microsoft Azure and Google Cloud at published list prices in September 2026. I built a small model that anyone can rerun and used it to compare two ways of running the same sort. One is an always-on server with provisioned block storage, which is how most rehosted batch looks on day one. The other is a server with local NVMe storage that only exists while the sort runs. On all three clouds the first costs about \$59–62 per sorted terabyte and the second about \$4–5. Nearly all of the difference is idle time. The charges most likely to surprise a migration team have nothing to do with compute. Sending the sorted terabyte back over the internet costs more than the whole always-on sort, and delivering it as millions of small objects costs more than the ephemeral one. At the end there's a checklist for costing a batch migration per unit of work.

What I found

- At list prices, sorting 1 TB a night costs **\$59–62 per terabyte** on an always-on server with block storage and **\$4.02–4.52** on a server that only runs for the sort. The architecture you pick matters 13 to 15 times more than the cloud you pick.
- On the always-on server, about 93 percent of the instance hours you pay for are idle. A three-year commitment cuts the bill roughly in half, to \$29–33. The idle hours get cheaper. They don't get used.
- Sending the sorted terabyte back over the internet costs **\$87–102** at first-tier list rates. That's more than the whole always-on sort. If the input comes from another zone, AWS adds up to \$20; Azure doesn't charge for that any more.
- Memory matters when you pay by the hour. Cut sort memory from 64 GiB to 1 GiB and the passes over the data double, which adds about 40 percent to the ephemeral cost. On an always-on server you don't pay any more, and the batch window gets 1.7 hours longer.
- The published CloudSort record is **\$0.97 per terabyte** (2022). That's a research system, and nobody should treat it as a target. It does show how far a lifted-and-shifted batch server is from what's possible.

1. Why cost per terabyte?

The FinOps Foundation calls FinOps “an operational framework and cultural practice” for getting the most business value out of technology, with engineering and finance working together with the rest of the business.¹ One piece of its framework is *unit economics*: working out “the cost and carbon emissions of a single unit of a business” and comparing it with the value that unit delivers.² For a batch shop the natural unit is the work done each night, like records posted or statements produced. For sort it’s even simpler. The unit is the terabyte sorted.

Unit costs matter because cloud bills are easy to run up and hard to pin on anything. In Flexera’s annual survey of more than 750 cloud decision-makers, respondents estimated that 27 percent of what they spent on infrastructure and platform services was wasted in 2025;³ in the 2026 edition it went up to 29 percent, the first increase in five years.⁴ Those are people’s own estimates, in a survey run by a company that sells cost-management software, so take them as a rough guide. They do fit with what follows. Most of the waste in a migrated batch server doesn’t show up on an invoice that lists instances, volumes and gigabytes, because every one of those lines is correct. You only see it when you divide the bill by the work.

This paper builds on two earlier ones in the series. *What Sort Really Costs on the Mainframe* showed that the mainframe software bill follows the monthly four-hour peak.⁵ *The Batch Window After the Mainframe* showed that once you’re off the mainframe, sort elapsed time comes down to storage throughput and the number of merge passes.⁶ Here I’m asking the question a finance director asks after the migration. What does each sorted terabyte cost now, and why?

2. The model

The workload is one terabyte (10^{12} bytes) sorted once a night, every night. All three providers turn hourly rates into monthly ones at 730 hours, which gives 30.4 nightly runs a month. Prices are Linux, on-demand, in US dollars, for one region each (AWS us-east-1, Azure East US and Google Cloud us-central1), retrieved in September 2026, without tax, free tiers or support plans. I costed two configurations.

- **A. Always-on, block storage.** A 32-vCPU memory-optimized server (AWS r7i.8xlarge at \$2.1168 an hour; Azure E32s v5 at \$2.016; Google c4-highmem-32 at \$2.0854)^{7–9} runs around the clock with three block volumes (input, work and output) of 1,200 GB each, provisioned at 500 MB/s and 3,000 IOPS. AWS gp3 costs \$0.08 per GB-month plus \$0.04 per MB/s-month above 125 MB/s;¹⁰ Azure Premium SSD v2 costs \$0.00011 per GiB-hour plus \$0.000055 per MB/s-hour above 125 MB/s;^{8, 11} Google Hyperdisk Balanced costs \$0.000109589 per GiB-hour plus \$0.000054795 per MB/s-hour above 140 MB/s.¹² IOPS stay inside each baseline.
- **B. Ephemeral, local NVMe.** A 32-vCPU server with local NVMe storage (AWS i4i.8xlarge at \$2.746, with $2 \times 3,750$ GB; Azure L32s v3 at \$2.784, with 4×1.92 TB; Google c4-highmem-32-lssd at \$2.4964, with 1,875 GiB)^{7–9, 13} is started for the sort and deleted afterward. It reads its input from object storage and keeps its work files on the local drives. The drives come with the instance price, and whatever is on them goes away with the instance.¹⁴ Output goes back to object storage. One generation of input and output (2 TB) is kept in object storage at \$0.023 per GB-month on S3, \$0.0208 on Azure Blob Hot LRS and \$0.020 per GiB-month on Google Cloud Storage.^{8, 15, 16} Requests are made in 16 MiB parts.

Using the model from paper 7, a sort with 64 GiB of memory and a 256-way merge needs one merge pass for a terabyte, so the data crosses storage four times.⁶ For configuration A I assumed 800 MB/s sustained to the volumes, which gives a run of 1.74 hours. Configuration B moves data through object storage at an assumed 1,000 MB/s in each of its two phases and takes 0.86 hours, counting ten minutes to start and remove the server. Both include a 25 percent allowance for CPU time and imperfect overlap. The throughput numbers are assumptions. I didn’t measure them, and paper 7 explains why you have to test them against each instance’s published limits. The model is illustrative. Its Python source computes every figure in this paper, and it’s available on request.

3. What a sorted terabyte costs

Exhibit 1 shows the result for both configurations on each cloud, line by line, so you can check any figure against the price list and swap in your own negotiated rate.

Exhibit 1. Cost per sorted terabyte at list prices, three clouds, two configurations (illustrative)

Per nightly terabyte	A · AWS	A · Azure	A · Google	B · AWS	B · Azure	B · Google
Instance	r7i.8xlarge	E32s v5	c4-highmem-32	i4i.8xlarge	L32s v3	c4-highmem-32-ssd
List rate per hour	\$2.1168	\$2.0160	\$2.0854	\$2.7460	\$2.7840	\$2.4964
Hours billed per night	24.00	24.00	24.00	0.86	0.86	0.86
Compute	\$50.80	\$48.38	\$50.05	\$2.36	\$2.40	\$2.15
Block storage, 3 volumes	\$10.95	\$10.34	\$10.24	–	–	–
Object storage held (2 TB)	–	–	–	\$1.51	\$1.37	\$1.22
Object requests (238,420)	–	–	–	\$0.64	\$0.64	\$0.64
Total per TB sorted	\$61.75	\$58.72	\$60.29	\$4.52	\$4.41	\$4.02

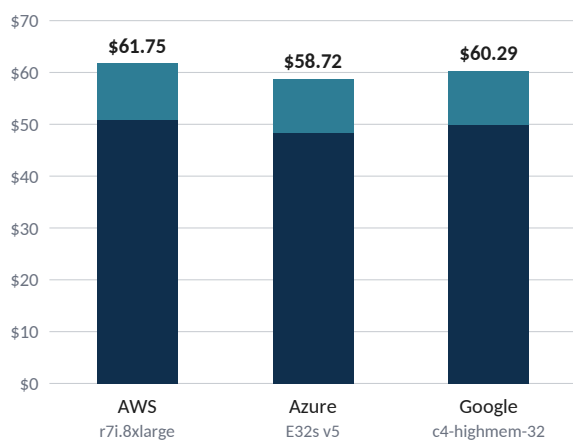
Computed with my model from published on-demand list prices retrieved September 2026 (us-east-1, East US, us-central1).^{7–10, 12, 15, 16} Monthly charges divided by 30.4 nights. Doesn't include tax, free tiers, support, staff, software licenses or data transfer (see Exhibit 3).

At list prices, the always-on configuration costs \$58.72–\$61.75 per sorted terabyte and the ephemeral one \$4.02–\$4.52. Between providers, within one configuration, the spread is about ten percent. Between configurations on the same provider it's a factor of 13 to 15. For a sort workload, the architecture you pick matters more than the cloud you pick.

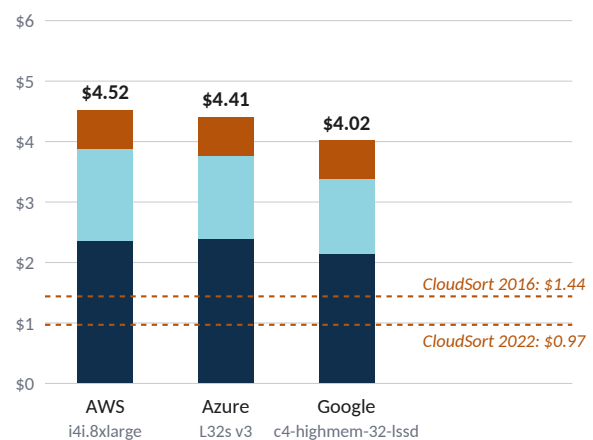
Exhibit 2 shows why. On the always-on server, compute is 82–83 percent of the nightly cost, and the sort uses 1.7 of the 24 hours you pay for. About 93 percent of the instance hours buy nothing for this workload. The block volumes add roughly \$10 a night whether anything reads them or not, because provisioned capacity and throughput are billed by the hour. On the ephemeral server the instance costs more per hour (\$2.7460 against \$2.1168 on AWS), but you only pay for 0.86 hours, and object storage replaces block storage at a lower rate per gigabyte.

Exhibit 2. Cost components per sorted terabyte (illustrative; note the different scales)

A. Always-on server, block storage



B. Ephemeral server, local NVMe



■ Compute (instance hours) ■ Block storage (provisioned) ■ Object storage held ■ Object storage requests

Same data as Exhibit 1. Panel B shows the published CloudSort records for comparison.¹⁷ The label under each bar is the instance type.

This comparison is only fair if the always-on server does nothing else. In practice a rehosted batch server often runs the whole overnight schedule, and its cost should be shared across the jobs it runs, in proportion to the hours or resources each one uses. The finding holds up after that correction. However you allocate it, the hours when the server runs nothing belong to no job, and somebody still pays for them.

For a lower bound, the Sort Benchmark's CloudSort category measures the list-price cost of sorting 100 TB of 100-byte records on a public cloud. The record dropped from \$4.51 per terabyte (TritonSort, 2014) to \$1.44 (NADSort, 2016) and \$0.97 (Exoshuffle-CloudSort, 2022, on Amazon EC2 and S3).¹⁷ These are research systems sorting synthetic data at scale, and no production batch job should expect to match them. The ephemeral configuration costs about four to five times the 2022 record. The always-on one costs about sixty times.

4. Commitments and spot capacity

Every provider discounts committed use. AWS offers Compute Savings Plans at up to 66 percent and EC2 Instance Savings Plans at up to 72 percent below on-demand, for one or three years.¹⁸ Microsoft quotes up to 72 percent for reserved virtual machines. It estimates savings of 36 to 72 percent, and the top figure is based on a three-year reservation of a large M-series machine.¹⁹ For the instances in configuration A, the published three-year rates come out 60.5 percent below on-demand on AWS and 60.5 percent on Azure. Google's three-year resource-based commitment for c4-highmem-32 is 55.0 percent below.⁷⁻⁹ Put a commitment on the always-on server and the cost comes down to \$29–33 per sorted terabyte. That's a big saving, and it's still 7 to 8 times the ephemeral cost.

The FinOps split between rate and usage explains why. A commitment lowers what you pay for each hour. It doesn't change how many hours you buy. Put one on a server that's idle 93 percent of the time and you've made the idle time cheaper and harder to get rid of, because the commitment runs for its whole term whether you need the server or not. Right-size the usage first, then buy commitments for the baseline that's left.

Spot capacity is the opposite deal. You get a deep discount, and the provider keeps the right to take the machine back. AWS advertises savings of up to 90 percent and gives two minutes' notice of an interruption;²⁰ Azure gives 30 seconds, offers no availability guarantee and lists batch processing among the workloads it suits;²¹ Google advertises discounts of up to 91 percent.²² On the retrieval date the Azure Spot price for an L32s v3 was \$0.586 an hour, 79 percent below pay-as-you-go.⁸ Sorts are rarely checkpointed, so an interrupted sort starts over. Let's say the discount is 70 percent, one run in ten gets interrupted half-way, and the rerun goes on on-demand capacity to protect the window. Then spot saves \$1.32–1.47 per terabyte, about a third of the ephemeral cost. Each interruption also adds most of an hour to that night's run. Spot is fine for sorts with slack. Sorts on the critical path of the batch window should run on demand.

5. Memory, merge passes and what time costs

Paper 7 showed that the memory a sort can use, and the number of files it merges at once, decide how many times the data crosses storage.⁶ Now that arithmetic has a price on it. With 64 GiB and a 256-way merge a terabyte needs four passes. With 4 GiB and a 16-way merge it needs six, and with 1 GiB and 16-way, eight. On the ephemeral server each extra pair of passes adds about 0.35 hours and \$0.87–0.97 per terabyte, so the eight-pass case costs roughly 40 percent more. On the always-on server the extra passes don't cost any money, because you're paying for the hours anyway. They do add 0.9 and 1.7 hours to the run. So a cost problem turns into a batch-window problem, which is what paper 7 is about.

It works the other way too, when you buy memory the sort can't use. On all three clouds the memory-optimized 32-vCPU instance costs 31 or 32 percent more an hour than the general-purpose one with half the memory. A terabyte sort with a wide merge needs one pass on either, so in the always-on configuration the general-purpose server would save \$11.52–12.10 a night. What you need to size is the memory each sort is allowed to use, multiplied by the number of sorts that run at once. The size of the server is the wrong number to look at.

6. Moving the data: zones, egress and requests

Data transfer inside a zone is free on all three clouds, and so is inbound transfer from the internet.^{8, 23, 24} Transfer between zones in the same region isn't always free. AWS charges \$0.01 per GB “in each direction”,²³ so a terabyte delivered from a server in another zone costs \$20.00. Google charges \$0.01 per GiB between zones, \$9.31 per terabyte.²⁴ Microsoft announced in May 2024 that Azure would not charge for transfer between availability zones.²⁵ So a zone-redundant design that puts the file transfer server in one zone and the batch server in another can add a third to the always-on cost on AWS, or more than four times the ephemeral cost.

Internet egress is the biggest single number in this paper. At first-tier list rates (\$0.09 per GB on AWS, \$0.087 on Azure and \$0.11 per GiB on Google's Premium Tier)^{8, 15, 24} sending a sorted terabyte back to an on-premises system costs \$87–102. That's more than the always-on sort, and twenty to twenty-five times the ephemeral one. Volume tiers bring the rate down somewhat at 30 TB a month, and dedicated interconnects have their own lower rates, which I haven't modeled here. For the design, sort the data where it will be used and move it across the boundary once.

Object storage also charges per request. List rates are the same on all three providers: \$0.005 per 1,000 writes and \$0.0004 per 1,000 reads (Azure quotes \$0.05 and \$0.004 per 10,000).^{8, 15, 16} With 16 MiB parts, uploading, sorting and reading back a terabyte takes 238,420 requests and costs \$0.64. If the same terabyte arrives as ten million 100 KB objects, the charge goes up by \$53.68, which is more than the whole ephemeral sort. Mainframe data sets are big sequential files, but migration tooling that splits output by customer or by record group can produce exactly this pattern.

Exhibit 3. How sensitive the cost per sorted terabyte is, dollars per night (illustrative)

Change from the base case	Basis	AWS	Azure	Google
Ephemeral, local NVMe (base B)	total	\$4.52	\$4.41	\$4.02
B with 4 GiB sort memory, merge 16-way (6 passes)	change	+\$0.95	+\$0.97	+\$0.87
B with 1 GiB sort memory, merge 16-way (8 passes)	change	+\$1.91	+\$1.93	+\$1.73
B on spot: 70% discount, 10% of runs interrupted and rerun on demand	change	-\$1.45	-\$1.47	-\$1.32
B with input arriving as 10 million 100 KB objects	change	+\$53.68	+\$53.68	+\$53.68
Always-on, block storage (base A)	total	\$61.75	\$58.72	\$60.29
A on a general-purpose instance (half the memory)	change	-\$12.10	-\$11.52	-\$12.10
A with a three-year commitment on the instance	change	-\$30.75	-\$29.27	-\$27.53
A with daily snapshots of the output volume kept 7 days	change	+\$11.51	+\$11.51	+\$10.72
Either: input delivered from another zone in the region	change	+\$20.00	\$0.00	+\$9.31
Either: sorted output returned over the internet	change	+\$90.00	+\$87.00	+\$102.45
Either: third-party software at \$100 per vCPU-year, 32 vCPUs	change	+\$8.77	+\$8.77	+\$8.77

Computed with my model. Spot and interruption rates, object sizes and the license rate are made-up inputs; the license rate is a round number and isn't the price of any product. Egress at first-tier list rates. Snapshot rate \$0.05 per GB-month on all three providers.^{8, 10, 12}

7. What the model leaves out, and a checklist

Most first estimates leave out three costs. The first is **protection**. Snapshots are incremental, but a volume that gets rewritten every night has no unchanged blocks to share. Keeping seven daily snapshots of the output volume adds \$10.72–11.51 per terabyte at the \$0.05 per GB-month list rate that all three providers charge.

The second is **operations**. Monitoring, log ingestion and support plans are priced per gigabyte, per metric or as a share of the bill. Each one is small per terabyte, and it never goes away. The third is **software licensed by capacity**.

Schedulers, runtimes, agents and utilities are often licensed per core or per vCPU of the machine they're installed on, busy or not. For every \$100 per vCPU-year, a 32-vCPU server carries \$8.77 a night, about twice the ephemeral infrastructure cost. If you pick an instance for its memory or its local drives, its vCPU count comes along with it, and so does its license count.

Exhibit 4 turns the model into questions a migration team can answer before cutover. Most of the evidence already exists. SMF type 16 and type 30 records on the source system give the volume and duration of each sort step, and the provider's published price list gives every rate. Write down when you pulled each price. List prices and product limits change, and you should rerun a unit-cost model whenever they do.

Exhibit 4. A checklist for costing a batch migration in unit terms

Item	What to find out	Why it matters
1. Unit of work	Bytes sorted per night and at month-end, per job and in total	The denominator of every unit cost
2. Run shape	Elapsed time, memory per sort, merge order and passes (paper 7)	Billed hours on ephemeral servers; window on always-on
3. Residency	Always-on or ephemeral; what else shares the server; hours actually busy	93% of always-on instance hours were idle here
4. Storage	Block capacity, provisioned throughput and hours provisioned; local NVMe for work files	Provisioned storage is billed whether used or not
5. Data path	Source and destination of every file: zone, region, on-premises; rate each way	Egress can exceed the cost of the sort
6. Object layout	Object count and size for inputs, outputs and intermediates	Small objects multiply request charges
7. Commitments	Baseline usage after right-sizing; term and flexibility	Commitments lower the rate; the hours stay the same
8. Spot	Which sorts have slack; rerun cost; notice period	Interruptions restart the sort
9. Protection	Snapshot, replica and recovery copies; retention; changed bytes per night	Rewritten volumes defeat incremental snapshots
10. Licenses	Metric of every third-party product on the server: cores, vCPUs, hours	Capacity licenses follow instance size, busy or idle
11. Review	Retrieval date of every price; owner of the unit-cost report	Prices change, so this is never finished

Items 1 and 2 can be measured on the source system from SMF records before migration.

8. Bottom line

- **Cost the work.** A sorted terabyte is a unit that finance and operations can both read, and it shows you costs that a list of instances and volumes hides.
- **Idle time is the biggest cost of lifted-and-shifted batch.** At list prices, the same sort costs about thirteen to fifteen times as much on an always-on server as on one that only exists while it runs.
- **Discounts follow architecture.** Commitments lower the rate for hours you've already chosen, and spot suits sorts with slack. Neither one will do what switching the server off does.
- **Watch the data path.** Cross-zone transfer, internet egress and small objects can each cost more than the sort. Sort the data where it will be used and move it across the boundary once.
- **Measure before you migrate.** The inputs to a unit-cost model are on your source system today, and the price list is public. The arithmetic takes an afternoon.

About the author

Bill Ahlbrandt is President and CEO of Ahlbrandt Software, which he founded in 1993. Before that he was a systems programmer, Director of Information Security and Director of R&D at Acxiom. He has been working on mainframe sort and batch for more than thirty years, and his company's sort engine runs mainframe sort jobs on Linux and other platforms for banks, insurers and government agencies.

Full disclosure: my company sells sort software for platforms other than the mainframe. Every price in this paper is a cloud provider's published list price. I don't say anything here about what any sort product costs or how it's licensed.

References

1. FinOps Foundation, "What is FinOps?", finops.org, accessed September 2026.
2. FinOps Foundation, FinOps Framework, domain "Quantify Business Value", capability "Unit Economics"; as documented in Microsoft Learn, "Quantify business value", accessed September 2026.
3. Flexera, *2025 State of the Cloud Report*, March 2025 (n > 750), as reported by CIO Dive, 19 March 2025. Vendor survey; waste figure is respondents' own estimate.
4. Flexera, *2026 State of the Cloud Report*, press release and blog, 18 March 2026 (n = 753). Vendor survey; waste figure is respondents' own estimate.
5. B. Ahlbrandt, *What Sort Really Costs on the Mainframe*, Ahlbrandt Software white paper, 2026.
6. B. Ahlbrandt, *The Batch Window After the Mainframe*, Ahlbrandt Software white paper, 2026.
7. Amazon Web Services, Amazon EC2 On-Demand and Reserved Instance pricing, US East (N. Virginia), Linux: r7i.8xlarge, m7i.8xlarge, i4i.8xlarge. Figures as compiled from the AWS price list by economize.cloud, retrieved 25 September 2026.
8. Microsoft, Azure Retail Prices API (prices.azure.com), region eastus: Esv5, Dsv5 and Lsv3 virtual machines (pay-as-you-go, reservation, Spot); Premium SSD v2; Blob Storage Hot LRS; Bandwidth; Standard HDD LRS snapshots. Retrieved 25 September 2026.
9. Google Cloud, "General-purpose machine family pricing" (C4), us-central1, retrieved 25 September 2026.
10. Amazon Web Services, "Amazon EBS pricing" (gp3 volumes, snapshots), aws.amazon.com/ebs/pricing, retrieved 25 September 2026; us-east-1 gp3 throughput rate corroborated by CloudZero, "EBS Pricing Explained", 2026.
11. Microsoft Learn, "Understand Azure Disk Storage billing", accessed September 2026.
12. Google Cloud, "Disk and image pricing" (Hyperdisk Balanced, Local SSD, snapshots), us-central1, retrieved 25 September 2026.
13. Microsoft Learn, "Lsv3 sizes series", accessed September 2026.
14. Amazon Web Services, *Amazon EC2 User Guide*, "Instance store temporary block storage for EC2 instances", accessed September 2026.
15. Amazon Web Services, "Amazon S3 pricing", US East (N. Virginia), aws.amazon.com/s3/pricing, retrieved 25 September 2026.
16. Google Cloud, "Cloud Storage pricing", us-central1, retrieved 25 September 2026.
17. Sort Benchmark, sortbenchmark.org, CloudSort results: TritonSort (2014), NADSort (2016), Exoshuffle-CloudSort (2022), accessed September 2026.
18. Amazon Web Services, "Compute Savings Plans and EC2 Instance Savings Plans pricing", accessed September 2026.
19. Microsoft, "Azure Reserved Virtual Machine Instances", azure.microsoft.com, accessed September 2026.
20. Amazon Web Services, "Amazon EC2 Spot Instances", and *Amazon EC2 User Guide*, "Spot Instance interruption notices", accessed September 2026.
21. Microsoft Learn, "Use Azure Spot Virtual Machines", accessed September 2026.
22. Google Cloud, Compute Engine documentation, "Spot VMs", accessed September 2026.
23. L. Rolim and L. F. Silva da Silva, "Exploring data transfer costs for AWS Network Load Balancers", AWS Networking and Content Delivery Blog, 8 April 2025.
24. Google Cloud, "Network pricing" (VPC data transfer and internet egress), retrieved 25 September 2026.
25. DatacenterDynamics, "Microsoft removes egress fees for moving data between availability zones in same Azure cloud region", 22 May 2024.