Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing
(HotCloud '11). June 14-15, 2011, Portland, OR.

Exertion-based billing for cloud storage access

Matthew Wachs*, Lianghong Xu*, Arkady Kanevsky†, Gregory R. Ganger*
*Carnegie Mellon University, †VMware

Abstract

Charging for cloud storage must account for two costs: the cost of the capacity used and the cost of access to that capacity. For the cost of access, current systems focus on the work requested, such as data transferred or I/O operations completed, rather than the *exertion* (i.e., effort/resources expended) to complete that work. But, the provider's cost is based on the exertion, and the exertion for a given amount of work can vary dramatically based on characteristics of the workload, making current charging models unfair to tenants, provider, or both. This paper argues for exertion-based metrics, such as disk time, for the access cost component of cloud storage billing. It also discusses challenges in supporting fair and predictable exertion accounting, such as significant inter-workload interference effects for storage access, and a performance insulation approach to addressing them.

1 Introduction

A core aspect of cloud computing, especially so-called Infrastructure-as-a-Service (IaaS) cloud computing, is tenants sharing provider services and paying for what they use. For this model to work, cloud providers must bill tenants. From the provider's perspective, higher bills are better, and fairness is irrelevant until tenants are able to hold them accountable for it. From the tenant's perspective, however, bills should accurately reflect the demands each given tenant places on provider resources — workloads that induce less resource utilization should yield lower bills, and bills should not be inflated by activities of other tenants.

Unfortunately, current billing models for cloud storage fall far short of this ideal. Storage capacity billing is reasonable, being based simply on "bytes stored," but storage *access* billing is not. For the latter, billing is usually based on the aggregate count of bytes or I/Os requested, without consideration for the exertion (i.e., resources/effort) required to provide for them. But, it is well-known that storage access efficiency can vary by orders of magnitude, depending on workload characteristics such as locality and transfer size — the cost to the provider for a given byte or I/O count would vary accordingly. As a result, tenant bills for storage access may bear little to no relationship to the actual costs.

This problem with storage access billing has several consequences. Most importantly, it is unfair to tenants with efficient access patterns. Since the provider must recover all costs, a tenant's bill will likely reflect average efficiency across all tenants, rather than that tenant's efficiency. Moreover, it acts as a disincentive for tenants to invest in efficiency. Having programmers improve access efficiency may not make economic sense, if bills will not reflect a benefit. Also, it obstructs a provider's ability to add higher-performance but more expensive storage options (such as SSDs), since current billing models account neither for costs nor for performance.

Storage access billing should be based on some measure of exertion, such as disk time or server time. An exertion-based metric would more accurately reflect the actual cost to the provider of a given tenant's activities. Indeed, for other resources, this approach is already taken — for example, CPU resources are often billed in terms of "CPU hours."

Exertion-based billing for storage access does raise several challenges. For example, storage access can involve a number of resources (e.g., network, server CPU, server cache, disk), and all would ideally be accounted for. Different variants of a resource (such as rotational disks vs. SSDs) have different acquisition and operating costs and should be billed at different rates. Also, different configurations (such as resiliency through RAID or replication) incur differing costs and should also be incorporated into the accounting scheme. More insidiously, access efficiency (and thus exertion for a given workload) can vary widely, in most storage systems, when they are shared by multiple clients (i.e., tenants). Inter-workload interference can reduce efficiency significantly, depending on how the workloads mix. With exertion-based billing, therefore, bills would also vary widely, depending on the activities of other tenants rather than on one's own activities.

Such variation is highly undesirable — indeed, an ideal billing model might yield the same bill each time a tenant applies the same workload, rather than penalizing a tenant for the provider's inability to manage interference. One option might be to create mechanisms to account for the effects of interference in exertion measurements. Instead, we promote an approach based on storage services that bound the impact of interference on efficiency (and thus exertion). As an example, we describe a request scheduler, Argon, that provides this property and thereby allows a cloud storage provider to bill fairly for the inherent costs of each tenant's workload.

2 Storage accounting

A cloud service provider pays to acquire a set of resources and to make them available at a particular point in time to customers. The number of resources (such as CPUs or disks) and the decision to operate a particular subset of the available resources (for example, to power up a server or spin up a disk) during a given period is driven by client demand. Thus, clients should pay for the load they present to each of the resources.

In accounting for storage costs, there are two reasons why a cloud service provider might need to pay to expand its storage resources. First, clients might require more capacity, thus requiring more platters to store their data. Second, clients might require more data accesses, or more performant data accesses. To accommodate this type of growth, the cloud provider might buy more independent disk heads to service requests in parallel, or more cache, or more bandwidth on the network links to the storage system. Consequently, storage accounting should incorporate both the capacity used by a given workload and the data accesses it makes over time.

Using the right metric. The obvious candidates for metering data accesses are based on throughput and bandwidth. Accounting for MBs transferred or IOs performed is straightforward, and such metrics are easily available from the storage system and visible to both the cloud service provider and the user. Table 1 shows the most popular cloud providers generally charge for capacity and IOs, with some exceptions ([1, 2, 3, 4, 5]). One might also propose accounting based on rates of MB/s or IOPS provisioned for a workload. Unfortunately, performance and transfer metrics are poor proxies for storage exertion.

Different workloads may require different levels of exertion to receive the same throughput or bandwidth. One common example is the distinction between sequential and random accesses. In Table 2, we show that sequential and random workloads require dramatically different exertion to achieve, for example, 1 MB/s. The sequential workload achieves up to 63.5 MB/s bandwidth given dedicated disk access, whereas the random workload only achieves 1.5 MB/s. Based on full-disk performance, the disk exertion to achieve 1 MB/s for the sequential and random workloads is 1.6% and 67%, respectively. If accounting is based on MB/s, then in this case, for the same 1 MB/s performance, the owner of each workload would pay the same, but running the sequential workload would consume significantly less disk time and thus incur less exertion. Accounting based on the number of MBs transferred in a window of time is not fundamentally different; for instance, transferring 1 MB in a second-long window might take 0.016 s (1.6% of 1 s) or 0.67 s (67% of 1 s). The metrics of IOs and IOPS suffer from the same problem; for instance, for a workload with fixed request sizes, IOs and MBs, and IOPS and MB/s, simply differ by a scaling factor (request size). Thus, performance- or transfer-based accounting would be unfair because a user running a low "difficulty level" (for example, sequential) workload would overpay, which deviates from the cloud accounting rule of "pay for what you use."

Provider	Capacity	Data access		
		Reqs.	Bytes	Perf.
Amazon EC2	1			1
Amazon EBS	✓	✓[2]		
Amazon S3	✓	✓[3]	4	
Google App Engine	✓			
Windows Azure	✓	✓		
vCloud Express	✓			

1. VMs are provisioned from a selection of tiers; discrete levels of maximum capacity and performance are available.
2. Requests = IOs.
3. Requests = PUT/GET/etc.
4. Billing for byte transfer only applies when crossing geographical region boundaries, so is ultimately a network charge.

Table 1: Popular cloud service providers generally charge for capacity and IOs.

The aspect of data access that represents the fraction of the disk resource being used by a workload, and the fraction of the resource not available to other workloads, is disk time. Because there is not a strong correspondence between throughput or bandwidth and disk time, throughput and bandwidth represent poor proxies for storage exertion. Because disk time is a simple measure which captures disk exertion precisely, we propose that storage access billing should be performed using an exertion metric that incorporates disk time (alongside exertion for other storage resources such as the caches[1]). Because a provider knows what each storage resource, such as a disk, costs to acquire and operate, using exertion metrics such as disk time allows the provider to bill each user fairly and accurately.

It's not just about locality. The distinction between sequential and random workloads is only one example of the fact that there is not a 1-to-1 mapping between user-visible performance and exertion. Metadata accesses may have very different costs than data accesses. Workloads with high cacheability may achieve high bandwidth and throughput with even less disk exertion than a "very easy" sequential workload. Alternative storage technologies such as solid-state disks present very different performance characteristics, different classifications

[1] Choosing appropriate metrics for exertion for each resource is outside the scope of this paper. For systems that provide resiliency through techniques such as RAID, access cost would include multiple accesses to independent disks, and capacity cost would capture the extra space used. Today, such features might only be imprecisely accounted for as different tiers or classes of storage with different rates.

Workload	Dedicated disk bandwidth	Disk exertion to achieve 1 MB/s
Sequential	63.5 MB/s	1.6%
Random	1.5 MB/s	67%

Table 2: Sequential and random workloads require dramatically different levels of exertion to achieve the same performance or quantity transferred.

of "easy" and "hard" workloads, and different ratios between performance and acquisition costs. All of these factors further compound the limitations of using bandwidth or throughput for accounting and all of these scenarios can be appropriately incorporated into an exertion-based billing model.

Does exertion make sense to users? Admittedly, disk time is not a perfect choice. While it can be thought of as, for instance, "using one quarter of a disk," many users may find it to be hard to interpret or predict. This may lead some to propose that the more directly visible metrics of throughput or bandwidth should be used anyway, despite their flaws. While these metrics are indeed more visible, we argue that they are only marginally more meaningful than disk time to a user. Ultimately, the metric that users value is application progress, which for even moderately complex applications will not be directly represented by any metric visible to the underlying storage system. For instance, a particular user might care about credit card transactions per second when thinking about performance, and cents per credit card transaction when thinking about accounting. A credit card transaction might take a variable number of IOs depending upon data layout, caching, and other effects, and therefore metrics such as IOPS are not especially interpretable from the user's perspective either. In this sense, all of the metrics, including IOPS, bandwidth, and disk exertion are only weakly meaningful to application progress. As a result, we believe using storage exertion, based in part on disk and other storage resource time, as the accounting metric is most reasonable because it is at least more natural for providers.

3 Making exertion fair

For exertion to be a reasonable accounting metric, it should be *fair* and *predictable*. Each workload should operate *efficiently*, accomplishing its work with as low a level of exertion as is practical. In addition, the exertion required to service a workload should be *independent* of the activities of other workloads or other artificial, external influences. One consequence of these attributes is that the cost across multiple runs of the same workload should be the same. Exertion-based accounting should always match the intrinsic nature of the workload rather than transient effects outside a user's control.

3.1 Problem

However, in practice, these goals can be difficult to achieve. Interference between workloads during runtime request scheduling is a common source of poor efficiency and unpredictability. For example, when a sequential workload is scheduled to run concurrently with another workload using the same real disk, their respective access patterns may become interleaved. In this case, the locality of the sequential workload may be lost in the mix. Consequently, its performance may become random-like and it may thus incur excessive exertion as compared to when it runs alone — and excessive exertion compared the natural expectation for sequential workloads. Table 3 shows an example of this scenario using experimental results from sequential and random microbenchmarks running together on a real system [8]. It would be unfair for the user to pay more for an intrinsically "easy" workload whose difficulty was artificially increased by external interference. But the cloud provider would not want to absorb the extra costs caused by an order-of-magnitude–or–more increase in exertion either.

3.2 Solution

Since inter-workload interference may lead to poor predictability and unfairness of exertion-based billing, it must be addressed for the billing mechanism to be acceptable. Rather than suggesting a way to account for or allocate the increase in costs, we believe a more direct and satisfying solution is to avoid the increase in exertion in the first place. Fortunately, effective solutions exist. To address the interference problem, the cloud provider should use a scheduler that limits interference between workloads while retaining efficiency and locality for each of them.

Argon [8] is a locality-preserving scheduler that helps address the interference problem and promotes each of the four goals for exertion: predictability, fairness, efficiency, and independence. Argon has a tuning knob called the *R-value* which represents the minimum efficiency level the system will maintain despite the overhead of sharing the system between workloads. Argon seeks to maintain the property of *performance insulation*: Each workload receiving a fraction f of disk time should receive performance $shared \geq standalone \times f \times R$. Argon is a quality-of-service system which operates over bandwidth performance metrics. However, Argon's focus on efficiency also has implications for exertion, and its use of disk time fractions maps to exertion.

As an example of how Argon manages performance, if the system is configured to maintain $R = 0.9$, then each workload should receive at least 90% of its dedicated-disk performance within its share of disk time and the performance loss due to sharing should be no more than

10%[2]. As a concrete example, a workload that receives 40 MB/s on a dedicated disk and is receiving 25% of a disk's total time would ideally receive 10 MB/s if there were no overhead from sharing the disk, and Argon would ensure at least 9 MB/s if the R-value is set to 0.9. This is in constrast to systems that do not explicitly maintain efficiency or any lower bound on performance due to highly variable and complex sharing overheads.

The performance benefits of Argon directly translate into exertion benefits as well. Because Argon strictly bounds the decrease in performance, it also bounds the increase in exertion (disk time fraction f) necessary to maintain a fixed level of performance $shared$. Manipulating the equation for performance insulation, $f \leq (1/R) \times shared/standalone$. The value $shared/standalone$ represents the exertion level one would ideally expect to yield a workload the performance level $shared$ based on extrapolating from its dedicated-disk performance $standalone$. Argon's performance insulation limits the inflation of exertion above this ideal fraction to $\leq 1/R$. For instance, if 90% efficiency is being maintained, then the overhead in exertion for a workload due to sharing should not exceed $\sim 11\%$.

One of the key techniques used by Argon to maintain efficiency is disk-head timeslicing. Each workload is assigned a controlled fraction of disk head time, which in this context can be interpreted as the disk exertion the workload is allowed to consume. These fractions dictate the relative lengths of per-workload timeslices in a round-robin schedule. The absolute lengths are calculated based on the R-value; timeslices are sized to be long enough that the cost of seeking between respective workloads' data, which is done at the boundary between timeslices, is amortized across a whole timeslice's worth of useful work to yield the desired efficiency level.

Argon also partitions the storage server's cache among workloads and selects appropriate workload cache partition sizes based on their respective access patterns. This allows each workload to maintain a hit rate close to its dedicated-server hit rate, except in the case (which Argon identifies up-front) where the workloads are too demanding to coexist on the same server. As a result, Argon protects workloads from efficiency loss and inflation of exertion due to poor cache sharing as well.

Earlier, we identified four goals for exertion: predictability, fairness, efficiency, and independence. Argon achieves each. Exertion under Argon is *predictable* because it is a simple function of the desired performance level and dedicated-disk performance, scaled up by the reciprocal of the R-value. Exertion under Argon is *fair* because the overhead of sharing is limited and does not fall disproportionately on any one workload. Argon is able to maintain high *efficiency* levels, such as 90%, without unduly affecting the latency of most requests [8]. And Argon promotes the *independence* of exertion values: The behavior of other workloads does not appear in the equation $f \leq (1/R) \times shared/standalone$ that characterizes the exertion needed to achieve a given level of performance under Argon. The influence of other workloads is straightforward and limited to the presence of an R-value less than 1 (representing the overhead of sharing) and to the fact that the fractions assigned to the set of workloads must sum to at most 100% (or else the system will be oversubscribed). The specific behavior of other workloads does not have a complex and unpredictable relationship with a workload's exertion under Argon as it does under systems that do not explicitly bound efficiency loss.

[2]The R-value represents a balance between efficiency and maximum latency / variance in latency.

	Exertion for 1 MB/s	
Workload	**Ideal (no interference)**	**Interference example**
Sequential	1.6%	23%
Random	67%	74%

Table 3: The exertion required to achieve the same level of performance can vary widely if the cloud service provider does not control inter-workload interference.

4 Discussion

A number of other issues arise when billing for storage exertion. This section surveys a few of them.

Are surplus resources free? As described in Section 2, there are two reasons to buy more disks, and thus two reasons to charge for disk use: capacity and performance. We expect providers will assign fees for capacity used and fees for exertion demanded, and bill for the sum of these charges. However, in some scenarios, one of the two might be "essentially free." For instance, if demand for capacity dominates (e.g., as it would for archival workloads), then the provider may own a vast number of platters that are nearly full of data, but whose disk heads are virtually idle. In this situation, marginal exertion may cost essentially zero. If demand for performance dominates, then the provider may own a vast number of spindles that are nearly constantly performing transfers, but whose platters have a large amount of free space. If even a modest amount of free time were available, marginal capacity for low-intensity workloads such as archival storage may similarly have virtually no cost.

If either of these scenarios are true, it may be most appropriate to only bill for the single metric that is causing system growth, and not for the other that is virtually free. Of course, if full but idle disks are spun down then the simplified view of idle resources being available at no marginal cost ceases to be true.

Off-peak usage. Another consideration involving

marginal cost involves provisioning for the peak. For instance, telephone networks are provisioned for the maximum number of concurrent calls, and many of the dominant costs are associated with building a system that can handle this high watermark. The incremental cost of a single call may be vanishingly small. As a result, some providers offer discounted rates during off-peak hours, and some mobile providers provide unlimited free "night and weekend minutes." Usage during peak hours incurs a higher charge because such usage contributes to the high watermark level. Similarly, storage in a cloud (and other resources in the cloud) may be provisioned for peak demand. The incremental cost of keeping a drive spinning during off-peak hours may not be neglibible, but there may still be motivation to charge a reduced rate for usage that is not part of peak load.

Competition among providers. A standard billing model based on storage exertion and diverse rates for it over time and across different providers would introduce market forces [6]. Once providers publish their rates, customers have the freedom to compare them and choose the provider that offers the lowest overall cost for their particular workload. A healthy market will require providers to compete for customers. In order to be competitive, they will need to bill for exertion in a predictable, fair, efficient, and independent manner. Thus, there will be an incentive and an imperative to implement efficient request scheduling, effective on-disk data layout policies, and other such approaches that minimize the overhead of virtualization.

Common units across resources. Disk exertion can be expressed as the percentage of disk time assigned to a given workload. Percentages are also natural metrics for other cloud resources, such as CPU time, memory footprints, and network bandwidth reservations. While there is no meaningful conversion between percentages for one resource and percentages for another, the use of a common unit enables certain types of analysis. For instance, customers or providers could assess whether their use of resources is balanced or bottlenecked based on percentages. If a workload uses CPU time more heavily than memory or storage, then an alternative algorithm that chooses a different spot on the space-time tradeoff (e.g., increased memoization) would be suitable. In addition, by considering exertion along with capacity and by being able to do a bottleneck analysis, different storage systems or technologies can better be compared. For instance, if a workload is bandwidth-heavy but capacity-light, billing for exertion might better allow the costs of upgrading to SSDs, or increasing cache sizes, to be justified.

Choosing storage systems. Exertion charging may improve providers' ability to choose among different storage systems and capabilities. For instance, higher-end storage systems may provider better performance and/or resiliency at higher prices. With more predictable and meaningful exertion charges, providers may better be able to assess whether higher-end systems are cost-effective. It may allow providers to better justify specialized storage for specific usage types (such as archival storage) or simply to choose a general-purpose system better matched to the applications in use. Cost differentiation may also clarify which resources (among the set available at a provider) to use for which workloads.

Other related work. Various authors, such as Wang *et al.* [9] and Li *et al.* [7], have also looked at accounting in a cloud environment. Wang proposes using metrics such as cost in addition to more traditional metrics such as throughput. Both cite real examples of cost varying significantly from run to run of the same workload. We focus on one cause of run-to-run variance, and suggest a request scheduler that reduces it. Wang demonstrates that users "comparison shop" among vendors, and Li shows that cost and performance for the same task can differ significantly across different cloud vendors.

5 Summary

Cloud storage access billing should be exertion-based, charging tenants for the costs actually induced by their I/O activities rather than an inaccurate proxy (e.g., byte or I/O count) for those costs. Although challenges exist in moving to such exertion-based billing, doing so rewards efficiency and can be fair to tenants. One set of challenges, related to inter-workload interference, can be addressed with explicit performance insulation, such as is provided by the Argon storage server. With such mechanisms, exertion-based billing can be fair and repeatable, increasing tenant ability to account for costs and justify efficiency improvements.

Acknowledgements

We thank Orran Krieger for discussions which inspired this work, and our colleagues in the Parallel Data Lab for helping expand on our initial ideas.

References

[1] http://aws.amazon.com/ec2/pricing/.

[2] http://aws.amazon.com/s3/pricing/.

[3] http://code.google.com/appengine/docs/billing.html.

[4] http://www.microsoft.com/windowsazure/pricing/.

[5] http://vcloudexpress.terremark.com/pricing.aspx.

[6] O. Krieger et al. Enabling a marketplace of clouds: Vmware's vcloud director. *ACM SIGOPS Operating Systems Review*, 44(4):103–114, 2010.

[7] A. Li et al. Cloudcmp: Shopping for a cloud made easy. In *HotCloud*, 2010.

[8] M. Wachs et al. Argon: performance insulation for shared storage servers. In *FAST*, 2007.

[9] H. Wang et al. Distributed systems meet economics: Pricing in the cloud. In *HotCloud*, 2010.