IBM Developer

Article

Cloud metering and billing

Billing metrics for compute resources in the cloud

☆ Save 👍 Like

Site feedback

By Jason Meiers

Updated August 7, 2011 | Published August 8, 2011

Alignment of IT resources with their cost can determine the profitability and allocation of cost per department or user. If an organization can't identify IT resource costs before and after their use along with what or who is consuming those resources, the correct entity may not be paying for ongoing support to keep the services available and maintained. For example, if a new service is brought online with a common database, it will be impossible to determine who will pay for the database or server space or for long-term capacity planning — a failure that may affect the organization's customers.

Cloud computing alone won't help an organization determine who will pay for what resource, but it can help provide a platform for an infrastructure design that establishes a charge-back model for metering and billing. This article describes the metering and billing options available for well-established cloud computing models as well as models offered by developing technology.

Frequently used acronyms

- **HTTP:** Hypertext Transfer Protocol
- **IT:** Information technology
- **REST:** Representational State

Cloud computing billing

Cookie Preferences

impacts

Each available cloud model has its own spin on how resource allocation is determined, and that spin is different from traditional IT business models in terms of affordability and the expense model in use. Lower cost and improved allocation of IT resources per service changes from *capital expenditure* for the common IT department to *operational expenditure* for the service and user. For example, the number of message queue `GET` and `PUT` operations per request can provide a cost structure for each customer that can in turn be accumulated for a total cost per transaction and ultimately per customer per month (similar to a mobile phone bill).

- **SOA:** Service-oriented architecture
- **SOAP:** Simple Object Access Protocol
- **WSDL:** Web Services Description Language

Accounting for cloud cost allocation in your code

If metering is based on transactions and leveraging the cloud computing cost allocation model, be sure to include cost-specific design patterns in your application code. Application architectures designed without developing patterns to use the cost per use of application resources won't provide the right infrastructure for your organization to employ next-generation cloud computing metering and billing options. For example, developing a next-generation, service-oriented platform and leveraging cloud computing may provide a cost-effective new way to do computing, but that platform may miss the boat in terms of being able to provide innovative solutions that scale up as well as down on demand.

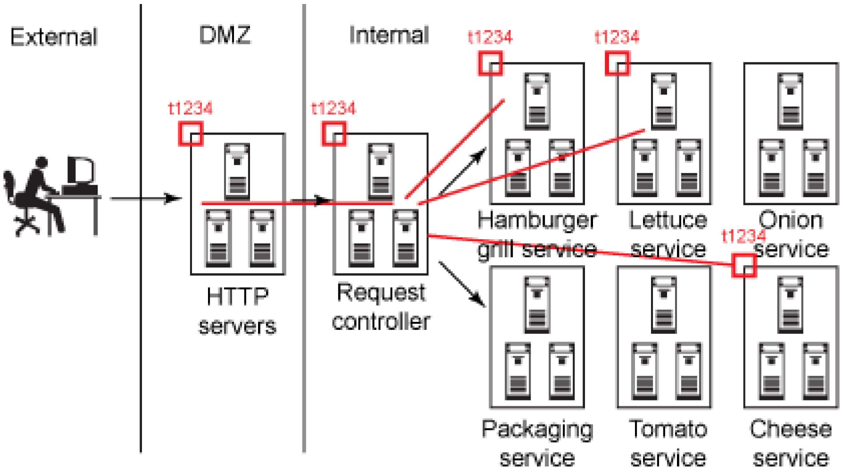
Set a goal of transaction tracking for each HTTP or SOAP request submitted and its associated cost for your cloud-based application. Because the resources — be they server hardware, a database request, a message queue request, or monitoring services — are charged based on actual usage, you must include the transaction consumer ID in each step and resource invocation. For instance, if you call an external service to get data from the database, the associated HTTP request should include the transaction ID as well as the consumer ID for later correlation of these metrics. Of course, you should have an additional thread in the application to capture transaction correlation data so that neither the core transaction performance nor response time is affected.

Figure 1 shows an example of a hamburger builder transaction that includes the different SOA services and uses a transaction ID. Agents are deployed on all of the nodes
transaction data for each transaction. Here, t1234 is the transaction ID tha

transaction data for each transaction. Here, t1234 is the transaction ID that identifies the

transaction; each service ties the elapsed CPU time to the transaction ID for later billing and metering.

Figure 1. Transaction example using SOA services and transaction IDs



Operations

The operation of cloud computing metering and billing is provided in some infrastructures (that is, the public infrastructure) and still required in private clouds built on enterprise application server infrastructures. The main differences are the security requirements, as most application-specific billing and metering are similar for private or public cloud computing. Some additional operational infrastructure items are required for metering and billing, however, such as messaging services to capture the usage data. Basically, additional infrastructure items are deployed to manage the use and cost of cloud computing metering and billing resources.

Established service models

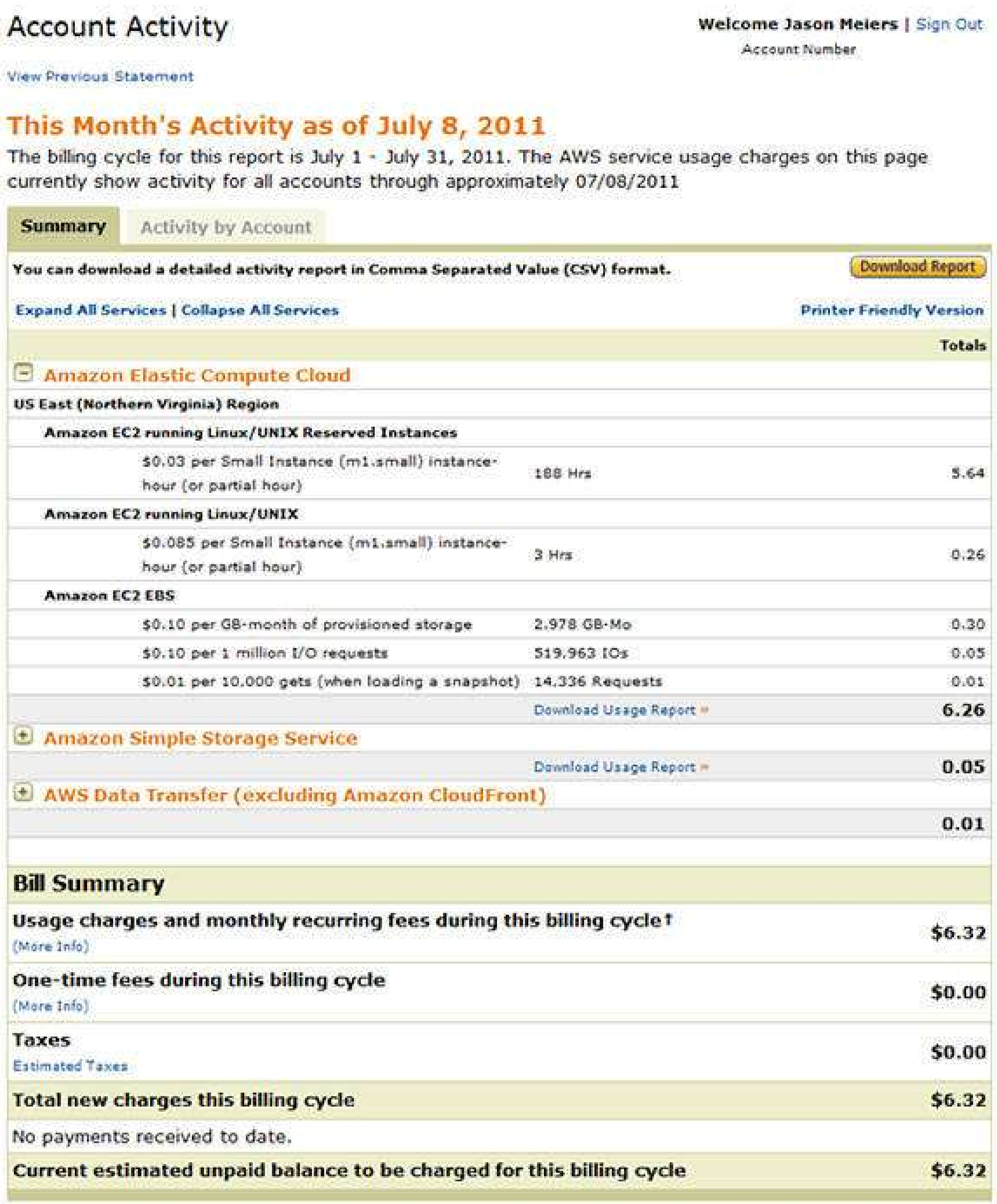
Some service models were initially thought of as more innovative than functional. However, they are established and considered usable for metering and billing in cloud computing infrastructures. It is important to note which models have been established — for example, server billing at US$0.10 per hour rather than large upfront procurement co

Cookie Preferences

Infrastructure as a service and billing and metering services

Historically, the high cost of provisioning servers and infrastructure limited the ability to develop software as a service (SaaS) applications. For example, it would take weeks if not months to plan, order, ship, and install new server hardware in the data center. Today, new billing and metering models allow procurement of hardware and operating systems — known as *infrastructure as a service* (IaaS) — in less than a minute (see Figure 2).

Figure 2. Monthly account activity for an IaaS platform



Account Activity

Welcome Jason Meiers | Sign Out
Account Number

View Previous Statement

This Month's Activity as of July 8, 2011

The billing cycle for this report is July 1 - July 31, 2011. The AWS service usage charges on this page currently show activity for all accounts through approximately 07/08/2011

You can download a detailed activity report in Comma Separated Value (CSV) format.

		Totals
⊟ Amazon Elastic Compute Cloud		
US East (Northern Virginia) Region		
Amazon EC2 running Linux/UNIX Reserved Instances		
$0.03 per Small Instance (m1.small) instance-hour (or partial hour)	188 Hrs	5.64
Amazon EC2 running Linux/UNIX		
$0.085 per Small Instance (m1.small) instance-hour (or partial hour)	3 Hrs	0.26
Amazon EC2 EBS		
$0.10 per GB-month of provisioned storage	2.978 GB-Mo	0.30
$0.10 per 1 million I/O requests	519.963 IOs	0.05
$0.01 per 10.000 gets (when loading a snapshot)	14.336 Requests	0.01
	Download Usage Report »	6.26
⊞ Amazon Simple Storage Service		
	Download Usage Report »	0.05
⊞ AWS Data Transfer (excluding Amazon CloudFront)		
		0.01

Bill Summary

Usage charges and monthly recurring fees during this billing cycle† (More Info)	$6.32
One-time fees during this billing cycle (More Info)	$0.00
Taxes Estimated Taxes	$0.00
Total new charges this billing cycle	$6.32
No payments received to date.	
Current estimated unpaid balance to be charged for this billing cycle	$6.32

The primary concepts of IaaS include:

- Servers per hour serving an on-demand model
- Reserved servers for better planning
- Higher and lower compute resource units based on application performance
- Volume-based metering on the number of instances consumed
- Prepaid and reserved infrastructure resources

- Clustered server resources

The billing for most of these elements is on a per-month basis, where each server is decommissioned and returned within a few minutes as initially provisioned. Billing charges accrued over the whole month include instances of servers running for the complete 30 days as well as servers running only up to one minute. Each compute cycle is charged a complete hour regardless of whether it ran for one minute or one hour.

Site feedback

The advanced billing and planning with reserved instances enables lower monthly as well as hourly costs to make compute resource models with known usage patterns and established baselines available as needed. In a model where servers are reserved in advance, an initial investment is required to secure specific servers in certain areas to minimize the hourly usage of virtual machines (VMs). In some cases, the initial investment can reduce the hourly price by up to 50 percent.

In most cases, scaling back instances during non-peak hours and scaling up during peak hours or seasons help to improve availability and response times. In general, if applications are tuned correctly, you would achieve a transaction per second rate that can scale horizontally with the number of servers added to the cloud computing infrastructure. The only concern is third-party resources that are not scaled with the infrastructure exponentially — for example, the database, authentication services, and other services that the scalable infrastructure accesses.

At a certain number of started servers, a discount occurs because of the large volume of virtual servers running — for example, when you reserve 100 VMs. This bulk discount helps the cloud computing provider plan for capacity demands and therefore minimizes the cost and risk of on-demand instances. Similarly, prepaid instances help the cloud computing provider estimate capacity and minimize the on-demand risk of running out of resources or sitting on too many unused instances. Discounts and usage often expire if the resources are not consumed within a certain amount of time. For instance, prepaid instances could be used for the baseline compute resource (a web server for the corporate intranet that is outward facing).

In larger deployments, starting and stopping instances and being billed by cluster utilization consolidates cost and management of IaaS. Because the management of single servers and resource utilization increase with enterprise applications, billing by cluster — possibly including custom resources such as routers and other devices and services — helps to reduce cost for management.

Cookie Preferences

Platform as a service and billing and metering services

Platform as a service (PaaS) billing and metering are determined by actual usage, as platforms differ in aggregate and instance-level usage measures. Actual usage billing enables PaaS providers to run application code from multiple tenants across the same set of hardware depending on the granularity of usage monitoring. For example, the network bandwidth, CPU utilization, and disk usage per transaction or application can determine PaaS cost.

Site feedback

The primary concepts for PaaS metering and billing include:

- Incoming and outgoing network bandwidth
- CPU time per hour
- Stored data
- High availability
- Monthly service charge

The bandwidth of incoming and outgoing network traffic determines the usage per user and creates a metric for billing and metering. The bandwidth metric is helpful, because web applications can be larger depending on their content. For instance, for most web services that return simple WSDL and RESTful payloads, the number of rows may not be significant compared to transactions that include pictures, video, and audio media.

Transaction and HTTP request metering based on CPU time per hour, minute, or second is the most accurate billing and metering model, as each transaction can be measured for total cost. Because you can't pin-point which transaction user is consuming a given amount of CPU resources per request, it's difficult to allocate resources at the user level. Therefore, a simple and effective measure for billing and metering is to determine the amount of stored data the user is consuming. Doing so helps in capacity planning, billing, and metering for services such as storage as a service, where data is stored in larger amounts on servers across the infrastructure. In such a case, a billing model based on gigabytes used determines what the costs of the service per month will be.

As in any enterprise application, the quality of service doubles (in most cas

Cookie Preferences

and price of implementation — sometimes more than double, because the infrastructure is replicated and includes additional infrastructure items to support high availability. High-availability billing and metering enables improved quality of service based on actual demand in cases where demand can be anticipated.

Advanced platforms that have a limited instance-level ability to provide metering and billing often opt to provide generalized billing models in which there is a flat fee to run application code. Such platforms typically include requirements for secure code that doesn't have long-running, CPU-consuming transactions as well as other, built-in security measures to curb utilization on the infrastructure — for example, a platform in which application code is deployed as a file and the underlining run time is provided with enhanced security measures and scalability by the platform as a service provider.

Site feedback

SaaS and billing and metering services

The traditional concept for billing and metering SaaS applications is a monthly fixed cost; in some cases, depending on the amount of data or number of "seats," the billing and pricing are optimized. The number of users is determined by the number of users the organization allows to access the SaaS applications, which increases the price of the monthly fee; in some cases, if certain volumes are met, there is a discount. For instance, sales software provided as a service would cost US$50 per month per sales agent for a company using the application.

The primary concepts for SaaS billing and metering include:

- Monthly subscription fees
- Per-user monthly fees

The monthly subscription fee is a fixed cost billed per month, often for a minimum contracted length of agreement of one year. The billing model per month changes the high initial investment from a software capital cost to a monthly operational expense. This model is especially appealing to small and medium-sized organizations to help them get started with the software required for their business initiatives. Scalability, or pay-as-you-grow models, are helpful for organizations that start with a small initial investment and a few users and grow as demand grows. In some cases, these organizations can scale down while providing access to the same data.

Cookie Preferences

Up-and-coming service models

Secondary service models are in progress, and many have standardized billing and metering models that have gained acceptance in all levels of business. Because SaaS is gaining acceptance, it is possible that these up-and-coming models will improve in adoption, as well. For example, SaaS providers can use database as a service (DaaS) and monitoring as a service (MaaS), and these models are gaining traction for cloud computing and with SaaS IT-focused companies.

Site feedback

DaaS and billing and metering services

The difference between traditional enterprise database infrastructures and software infrastructures is the built-in scalability and billing for what you actually use. DaaS infrastructures employ these concepts:

- Instances of database servers
- Scalable cloud computing database services

The database instances that exist today in large enterprise infrastructures started with an infrastructure as a service platform using license agreements that already exist. This groundwork helps the implementation of software license agreements in DaaS models. For example, customers with existing licenses can run the same database instances per core in a cloud computing infrastructure.

Databases built to leverage cloud computing scalability are available and billed in actual usage, often based on the number of requests executed on the server. This model helps to determine the actual usage of software and infrastructures for databases. Sometimes, DaaS providers may bill for database utilization by including the elapsed CPU time since one request used more CPU time than what is typical. For example, a long-running insurance transaction may include hundreds of milliseconds of response time, with thousands of rows inserted, where financial payment transactions may use less, having end-to-end response times in the 200 millisecond range.

Cookie Preferences

MaaS and billing and metering services

Adding MaaS to an existing monitoring infrastructure aligns with availability requirements for infrastructure services. MaaS employs these concepts:

- External service monitoring
- Instances of monitoring infrastructure
- Elapsed CPU time

Site feedback

Monitoring though the use of external services has been available for a while, providing availability checking for IT compute resources though pinging or synthetic transactions from the software developer's data center. This service is often billed monthly and based on actual usage as well as intervals the monitors execute the transaction and cycles the data is collected. For example, when transaction are monitored on the business website, each HTTP request is added to the monitoring infrastructure provider and billed as a complete package of 200 URLs. This solution does not require the organization to have administrators on staff to manage the monitoring infrastructure and is billed on a per-month basis.

More complex infrastructures for monitoring the complete infrastructure as a software service maintained by the customer can be provided as a service offered by the vendor or software development partner, billed and metered on an as-needed basis. This required administration of the monitoring infrastructure at the software and operating system layer typically includes the hardware and infrastructure, as well. For billing, customers can either pay a monthly fee or reuse of existing enterprise licenses.

MaaS based on elapsed CPU time determines the actual usage of each request and is consolidated at the end of each month. Without determining exact usage, it is difficult to provide scalable solutions for both small and large consumers, as usage can differ. For example, in event management, where filters for each event are processed for each transaction request, the table of measure is an accumulation across the composite transaction services for the elapsed CPU time.

Conclusion

Cookie Preferences

Metering and billing for SaaS adds offers models that align with business objectives, providing detailed accounting requirements for business units in larger organizations as well as lower initial investments for startup companies and small businesses. The large initial investment and procurement of software with SaaS is shifting to fit what is actually used and enabling new projects to leverage enterprise-class software for which a budget may have not been previously available. In addition, scalability for larger volumes of transaction load is no longer available only to enterprises.

Site feedback

Similarly, the shift from capital expenditure to operational expenditure enables more precise billing and metering models that meet accounting requirements based on department usage. For example, the sales department is now able to add new users based on actual usage without increasing the complexity and cost of procuring new hardware, software, and administrative resources.

Legend ⓘ

Categories

Cloud Financial services Platform as a service

Table of Contents

Cookie Preferences

Build
Smart ↓
Build
Secure ↑

IBM Developer

About

FAQ

Third-party notice

Follow Us

Twitter

LinkedIn

Facebook

YouTube

Explore

Newsletters

Code patterns

APIs

Articles

Tutorials

Open source projects

Videos

Events

Site feedback

Community

Career Opportunites

Privacy

Terms of use

Accessibility

Cookie preferences

Sitemap

Cookie Preferences