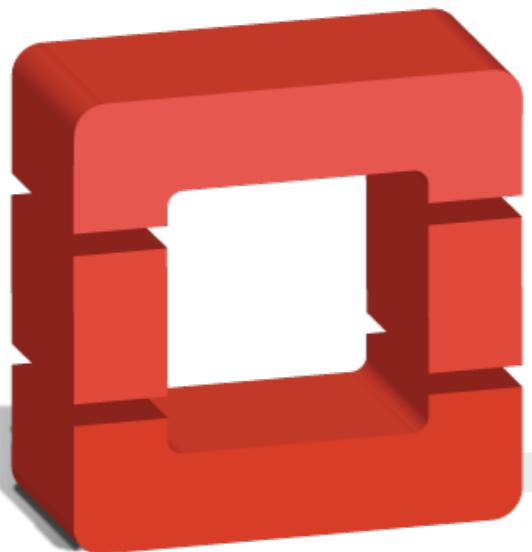


OpenStack® for Enterprise

THE TIPPING POINT COMETH



openstack™

CLOUD SOFTWARE

What Is OpenStack?

There is a lot of buzz in the industry around OpenStack. The open source cloud operating system, first released in 2010, has gained considerable momentum over the past half-decade. Its name is used alongside major players in both the public cloud, as well as in private cloud product environments. These comparisons are made with good reason, as we have observed a significant upswing in organizations taking an interest in finding out about how OpenStack may become a part of their environment and long-term cloud strategy.

This white paper provides an introduction to OpenStack and its brief history, as well as a summary of VMTurbo's contributions to the community.

What Is OpenStack?

Definition and Releases

OpenStack Defined

According to the definition at OpenStack.org:

"OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface."



First released in 2010, OpenStack began as a partner development between NASA and Rackspace®. The intent of the project was to accelerate application development cycles by enabling developers to provision virtual resources on-demand through a self-service portal. The first release, codenamed Austin, controlled only compute and object storage, and was extremely simple relative to the OpenStack we know today. Each subsequent release of OpenStack has been codenamed, marked by a word (often a location) starting with the next letter of the alphabet. Major releases have been consistently launched on a semi-annual basis: Austin, Bexar, Cactus, Diablo, Essex, Folsom, Grizzly, Havana, Icehouse, Juno, Kilo, and (anticipated late 2015) Liberty.



Each OpenStack release also bears additional point releases, formatted (YYYY.E.N), a taxonomy reflecting the year of release, the edition release of that year (1 or 2), and the release number. An example is the Icehouse edition: version 2014.1 (April 17, 2014), 2014.1.1 (June 9, 2014), 2014.1.2 (August 8, 2014), and 2014.1.3 (October 2, 2014). When the next full release comes out, the previous release no longer increments and updates come only as security patches where necessary.

Fixed Release Dates, Variable Features

OpenStack development cycles are fixed on the calendar in advance of the next full release. The OpenStack development process integrates features using a well-orchestrated CI/CD (Continuous Integration/Continuous Deployment) methodology, which accepts only tested, working features into the code base, referred to as the *trunk repository*. The trunk repository constitutes the “official” version of OpenStack for any given point in time.

This aggressive technique has been developed and maintained well, even with thousands of developers submitting code for consideration by OpenStack.

OpenStack Programs

The programs, also called projects, are the different portions of OpenStack that make up the operational components to run a complete cloud environment. Programs are divided into different categories based on how rigorously they have been tested and integrated into the trunk repository.

Programs are categorized as:

- o Integrated (previously Core)
- o Incubation
- o External

Programs were previously referred to as Core, but this has been changed to Integrated. Admittedly, it can be confusing if you read different documentation due to the change in terms along the way.

Incubation programs are those that are not yet considered fully Integrated. This is based on a variety of criteria including testing processes and the maturity of features. Incubation programs go through what is called a graduation process, and are considered Incubation programs for 2 full development milestones prior to attaining Integrated status.

New programs are called External programs and also have a requisite qualification process to become an official OpenStack Incubation program. External programs must have a well-defined scope, and demonstrate a need for the features they will add to the OpenStack ecosystem. A level of maturity in the development process is also required so that a momentum is already in place to keep the program developing and help grow the OpenStack feature set.

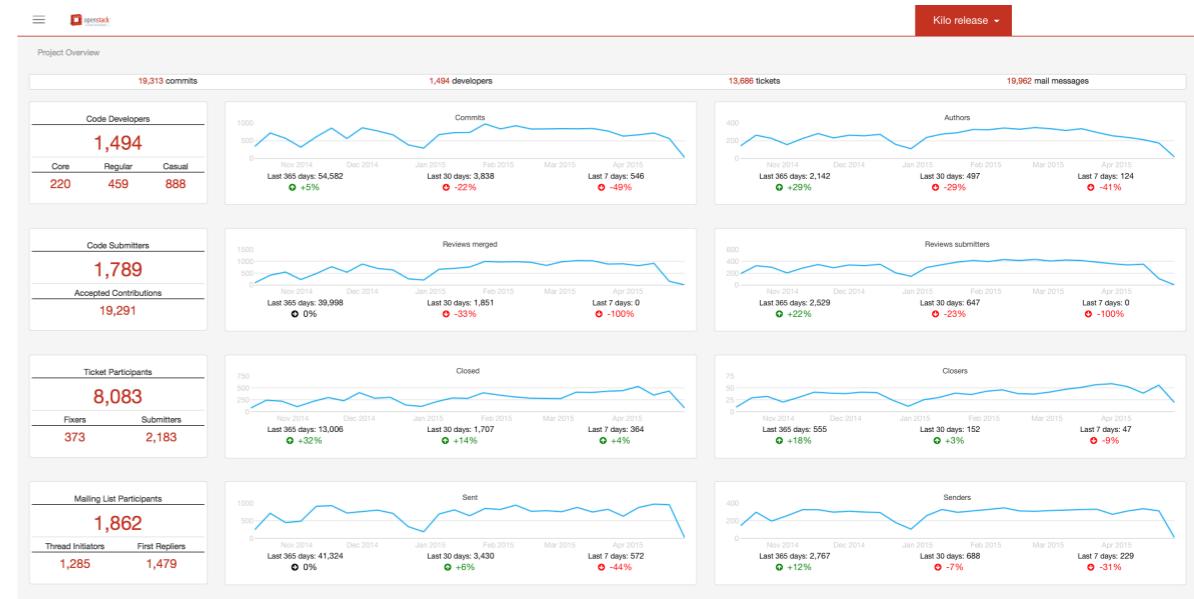
These program levels have been well established since the inception of OpenStack, and are a key part of the successful development and release

process. Most importantly, all program code must adhere to the [Apache License 2.0](#), include documentation, and provide a RESTful API and Python client library.

The OpenStack Developer Ecosystem

Each OpenStack program is led by what is called a PTL, or Program Technical Lead. The PTL is charged with the task of maintaining the codebase and features in the backlog as OpenStack progresses from release to release. The PTL and supporting team work to integrate upstream code submitted by developers. The developer ecosystem can include individual programmers, as well as corporations who provide with their development teams. The OpenStack Foundation itself has developers who maintain and contribute code also to help drive the development.

OpenStack maintains the [Project Overview Dashboard](#), which charts the developer ecosystem and accompanying code commit activities for each release and the project as a whole.



As of the Kilo release, there were 1,494 code developers including 220 core developers, 459 regular developers, and 888 casual developers. The classification is based on participation levels and the amount of time dedicated to the development of OpenStack specifically.

From the overall code submitter pool of 1,789 submitters, there were 19,291 commits overall to the Kilo release. Of note is that all of this development takes place across the world and over a seemingly compressed 6-month development cycle.

Compared to the Juno release, which had 1,433 code developers and 18,358 commits, we can see that contribution is growing across the board. Contribution growth is impressive to say the least, and it reflects the upward momentum of the OpenStack ecosystem altogether.

The OpenStack Foundation

The OpenStack Foundation is the governing body that helps to guide the growth and development of OpenStack. The foundation is comprised of a Board of Directors who manage corporate governance, and a Technical Committee, who provide technical leadership on the OpenStack ecosystem.

By having the OpenStack Foundation provide general guidance, the OpenStack ecosystem is able to be community-contributed while maintaining a direction that keeps the overall development and growth on track toward a specific, agreed-upon set of goals. This has been the key ingredient in maintaining standards while also responding to community-driven change.

All Clouds Open Clouds

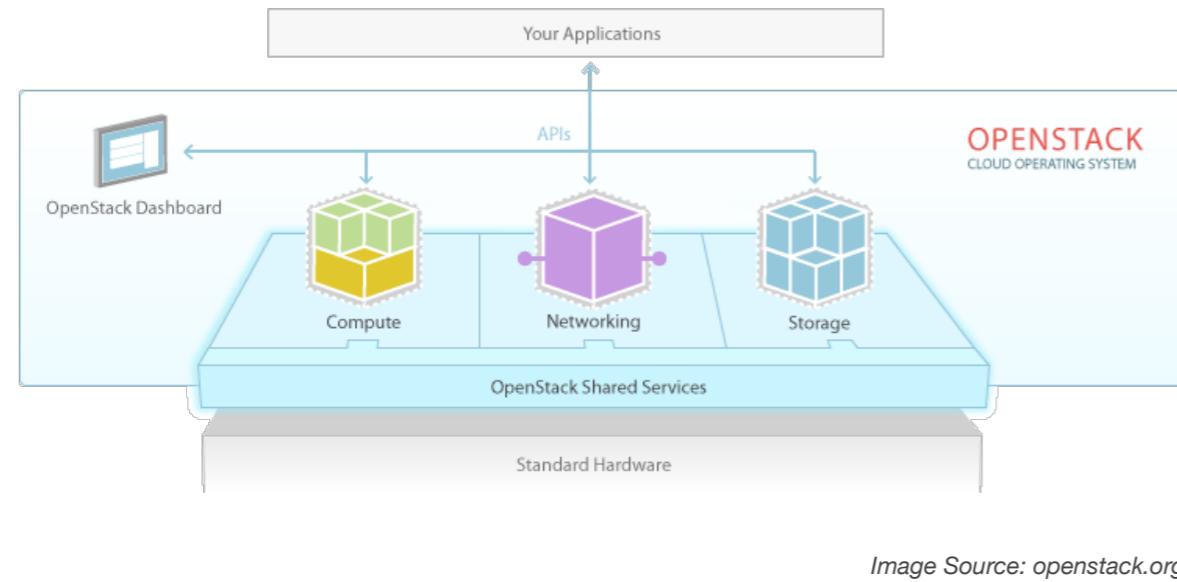
So what is the ultimate vision for OpenStack? If we revisit the definition, it seems self-evident that the notion is to develop OpenStack to full feature parity with propriety cloud platforms, while maintaining the flexibility and customizability that only open source can offer.

The project has come a long way since Austin. In the next section, we will summarize the full menu of OpenStack Programs (not to be confused with Releases), which drive toward this parity.

OpenStack Programs

Understanding the Programs

Since its inception in 2010, the program list (sometimes referred to as projects) has grown to include numerous integrated programs. The following list describes each of the integrated programs up to Kilo.



Nova: OpenStack Compute

Perhaps the most central of OpenStack projects, Nova is the heart of the compute platform that drives the ability to run and manage guest instances. Nova has a scheduler and drivers to connect and manage hypervisors on hosts to run instances. This can be confusing, as *Nova is not a hypervisor itself*.

The underlying hypervisor can include VMware vSphere® ESX(i), Microsoft Hyper-V®, KVM, Citrix XenServer® **and some others**, including the Docker container platform. Because OpenStack is hypervisor agnostic, organizations can actually run multiple hypervisors and control them all together as a part of a single cloud infrastructure.

Nova also provides networking support for the OpenStack cloud with the ability to run Layer 3 networks and provide DHCP services to the running instances. Nova network was once considered to be heading towards “legacy” status, but it was found that Nova networking was sufficient for many customers and has since held and gained traction with contributions from the developer ecosystem.



Cinder: OpenStack Block Storage

While Nova instances can run using simple ephemeral storage close to the hypervisor, the need for persistent block storage is a common one. The Cinder program manages the abstraction of storage, which can be either hardware or software driven.

Many drivers exist for commercial storage vendor platforms, evidence that there are feature-specific support levels within each vendor driver.

Another misconception about OpenStack is around different features and capabilities such as mirroring, snapshots, and other features that we find in many enterprise storage platforms. Cinder itself does not provide these features, but it

performs the management of features, which are offloaded to the underlying storage platform.



Swift: OpenStack Object Storage

Distributed object storage is most commonly attributed to the widely used Amazon S3 (Simple Storage Service) platform. Objects are written to storage, remain fully accessible, and are given a fully addressable URI to retrieve them as needed.

Object storage in OpenStack works in the same way with a distributed JBOD (Just a Bunch Of Disks) methodology. The Swift infrastructure uses algorithms to distribute copies of objects across multiple storage endpoints to provide protection. This is very different from traditional RAID (Redundant Array of Independent Disks).

Swift, being the oldest of the programs (along with Nova), is regarded as one of the most mature. It has an independent release cycle and is also one of the simplest programs to upgrade due to its distributed topology and ring architecture.



Neutron: OpenStack Networking

Perhaps the most-talked-about program in OpenStack is Neutron. Once dubbed Quantum, it was renamed to Neutron after the Grizzly release to eliminate any legal conflicts with the storage vendor holding the same name.

Neutron provides advanced networking capabilities to OpenStack including support for Layer 2 and Layer 3 networking with extended support to allow SDN (Software-Defined Networking) platforms such as VMware® NSX™, Cisco® ACI, and Open vSwitch to operate stretched Layer 2 networks.

The key feature for Neutron is the design, which includes the ML2 (Modular Layer 2) plugin support that can be used by any supporting vendor. This standardized approach allows for the best in interoperability.

NFV (Network Function Virtualization) features are also present in OpenStack thanks to the capabilities of Neutron.

Horizon: OpenStack Dashboard

One of the core tenets of a cloud environment is the ability to provide “on-demand self-service”. This is done for OpenStack using the web-based Horizon dashboard program. Using Horizon, both administrators and consumers of the OpenStack cloud can request resources, and manage existing instances and features as allocated by rights using Keystone.

The simple and effective user interface is becoming a focus of developers because of the importance of ease-of-use in adoption of any cloud technology.

Keystone: OpenStack Identity

The Keystone program handles authentication and authorization. Using a built-in directory, or extending to attach to an existing LDAP-compliant directory (if supported), OpenStack operators are able to create and assign users and roles to use OpenStack services.

Using a token system, Keystone will authenticate a user, provide a time-based token, and all future transactions by that user session will pass the token between the other programs where the rights are checked for authorization to perform different functions.

Glance: OpenStack Image

For those familiar with VMware, think of images as VM templates. As with Amazon Web Services™ (AWS) or other cloud products like Microsoft® Azure, images are

the catalogue of instance templates that you use as a baseline when deploying personalized instances.

Images can include base instances with a simple operating system, or can also be fully customized application builds. Using the Glance program, one can upload images on a per-tenant and per-cloud basis. The tenants can also create Glance images for use within their own personal tenant area.

Ceilometer: OpenStack Telemetry

Telemetry is the measurement and metering of instances. The Ceilometer program provides instrumentation and usage information about the running instances and hosts. This detail can be used to feed into a chargeback system and provide pricing per-tenant.

Heat: OpenStack Orchestration

The Heat program handles orchestration within OpenStack. There are many other orchestration tools that work alongside an OpenStack deployment such as Puppet, Chef and Ansible, but the native orchestration can now be done within Heat.

Using a YAML-based template format called HOT, administrators and tenants alike can build Heat stacks to be used for repeated deployments and automation within the OpenStack cloud.

Trove: Database Service (DBaaS)

While traditional relational databases can be run inside a guest instance, there are many cloud-based database alternatives that have risen in popularity. Azure offers the same features as traditional on-premises MS SQL server in their cloud platform. This is precisely what the Trove program is built to do for OpenStack.

The database service can be used by any instance in the tenant, which allows for a distributed database design without the need to manage the underlying operating system and dependencies.

Sahara: Data Processing (Big Data)

Since Icehouse, the release that preceded Juno, the Data Processing program codenamed Sahara was elevated from Incubation to an Integrated program. This exciting program has many watching how it is forming and is also attracting a lot of contributions. By providing an as-a-Service type of presentation for a Big Data platform, we could see the overall data processing field move more to the mainstream as people have a new way to deploy and test.

OpenStack for Enterprise: The Tipping Point

Recently, OpenStack has made headlines at the expense of the legacy clouds it's replacing. ***"(Company) Dumps (Vendor X) Contract In Favor of OpenStack Cloud"*** is not uncommon news these days. Take your pick: Wal-Mart, PayPal, Cisco Webex, Best Buy; all have built their clouds on OpenStack. In the next section, we'll take a look at a few of these enterprise implementations, and explain why the tipping point is here.

Enterprise Adoption

5 Years, 11+ Releases, and 1 Revolution

Well, revolution may be an overstatement, but it is fair to say that there is a fundamental shift in the industry, facilitated by all that OpenStack has brought to us. From its roots as a shared project with NASA and Rackspace, OpenStack has grown significantly in both its capabilities and its reach across organizations.

Before we look at the *What* of any technology such as “What can it do?” or “What does it produce?”, we have to understand the *Why*. Understanding why something exists is fundamental in proving the business value it provides. Remember that value is the whole reason technology products exist.

OpenStack began as an enabling technology that was built with strong focus to answer the need for on-demand virtual resources accessible by application developers. This core has held strong and the growth in the programs to support OpenStack have maintained the primary tenet of on-demand, API-accessible resources to drive applications.

At some point, however, OpenStack crossed-over from development-focused to production-ready. Its evolution has been driven through a combination of communal and commercial code contributions, which intentionally or not, have launched OpenStack to much closer parity with commercial cloud platforms offered by longstanding industry leaders.

The evidence? [Major companies, including Walmart and PayPal](#), have made headlines by going all-in on OpenStack, further begging the question “Why?” Their reasons are compelling.

Why OpenStack?

The fundamental reason that technology exists is to enhance an existing process that is typically people-based. Technology for the sake of technology has a limited shelf life because it doesn’t answer a specific business need. Business drives technology, which in turn drives business. That order is specific.

A 2015 survey conducted by VMTurbo polled more than 1,200 IT professionals spanning all verticals with cloud environments ranging in size from less than 500 to more than 20,000 virtual machines. The survey found that 45.1 percent of respondents were either already running or actively considering OpenStack deployments in a test, dev, or production environment. Of those, the avoidance of vendor lock-in (39.1 percent) and enhanced agility (30.4 percent) were the primary drivers of their open cloud exploration.

In the case of [Walmart](#), the increasingly competitive e-commerce space necessitated cloud innovation. Amandeep Juneja, Senior Director of Cloud Operations and Engineering at Walmart Labs, cited that E-Commerce 3.0 requires flexible architecture whose scale economies mirror those of the business it supports. In 2014, when Walmart committed to reimagining its e-commerce cloud, Juneja sought a platform that “would best support [Walmart’s] application developers, enabling them to rapidly build all kinds of applications, including mobile, WebApps, and RESTful APIs for vendors. A platform that would empower product managers to iterate over new product ideas in an agile manner. A platform that would enable Walmart to respond to customer needs more efficiently.”

OpenStack was selected as the cloud platform of choice for several key reasons. Those Juneja public cited in his Walmart Labs blog post included:

- Avoiding long-term lock with any single private cloud platform vendor
- Using open source flexibility as a source of competitive advantage for Walmart (fueled, of course, by an arsenal of highly-skilled developers)
- The Community. OpenStack's community is alive, well, and supported by the best talent in both Silicon Valley and around the world

The size of Walmart's deployment? A casual 100,000+ cores on Nova, serving Walmart's 140,000,000 weekly U.S. store and website visitors. Walmart Labs also claims to be developing a SDN framework off of Neutron, a block storage environment with Cinder, and an already-in-production multi-petabyte Swift deployment.

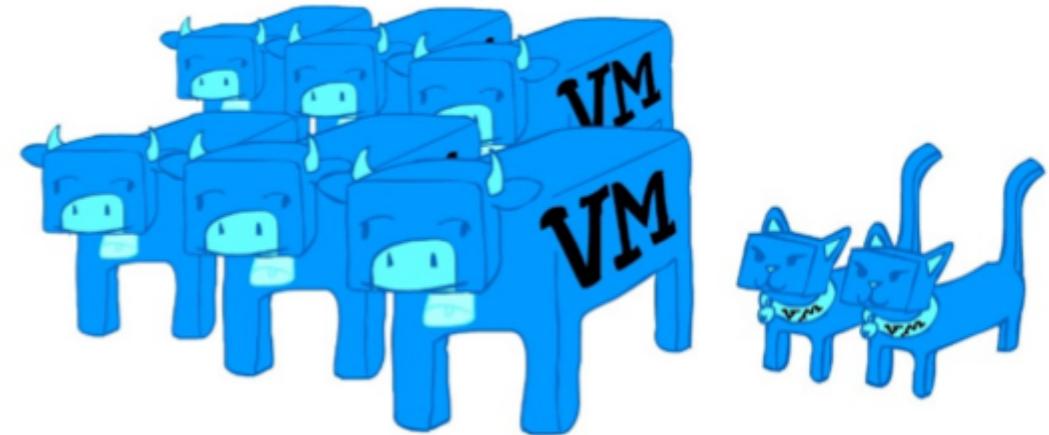
Walmart is not alone. Online payment platform PayPal in 2015 completed transition of its entire front-end cloud to OpenStack. Like Walmart, [PayPal's](#) decision was fueled by the need for greater ability and agility to innovate to meet customer needs.

In December 2011, PayPal was routing 20% of incoming requests to its small, OpenStack-based private cloud - at the peak of the holiday shopping seasons. Says Sri Shivananda, Vice President, Global Platform and Infrastructure, "Since, we have continued to build on top a broad base of OpenStack technologies. With our private cloud infrastructure, we've been able to deploy new Java applications and provision infrastructure capacity within minutes - instead of days. It is critical that we have infrastructure that is available on demand for our developers to deploy to. At the same time, it is important we have highly available infrastructure that is consistently manageable at scale."

OpenStack For the World, Not Versus the World

The long-standing description of how cloud deployments differ from traditional application deployment is the oft-compared "pets versus cattle". Traditional

applications and servers were treated as pets, and a modern scale-out application is more like nameless, utilitarian cattle.



The industry has evolved to recognize that there is not a clear divide between these two descriptions. Many enterprise organizations are beginning to embrace cloud methodologies and change the culture of development to utilize more scale-out, N-Tier deployments. As the shift towards this application deployment style occurs, OpenStack and similar enabling technologies become much more viable, and important.

The divide between application-centric companies and traditional IT organizations is becoming less and less with each passing year. There are many who say that every organization will become a technology organization within a few years or else they risk being pushed out of the market; to this end, OpenStack is making cloud adoption viable for all.

Another challenge within the technology ecosystem is the comparisons against other products such as VMware vCloud®, Amazon Web Services and Microsoft Azure among others. OpenStack is different in many ways than any of its supposed competitors, but there is not an apples-to-apples comparison truthfully.

OpenStack as an overall product ecosystem has features and capabilities that can be individually compared to other products in the market. There are also distinct

reasons why the programs have been built to be loosely coupled and independent in many ways. Parts of OpenStack can be productized such as Swift and others can be deployed in smaller scale to achieve a specific purpose.

The flexibility of OpenStack as a product set gives it a very interesting value to the industry and to its customers. This is why OpenStack is often viewed as almost Hydra-like with many heads that can each operate independently. This could be particularly important as OpenStack is pitted against other software and infrastructure tools in the coming months and years.

Despite resistance that the only clouds should be public clouds, many organizations are embracing cloud methodologies while using on-premises deployments to maintain privacy, or to leverage their existing capital investment in high-performance data center equipment.

There are powerful use cases for private clouds that preclude the use of public cloud resources, and this has become generally accepted now. Commercial offerings are available from companies such as Red Hat, Mirantis, and recently EMC, HP, and VMware as examples. This is adding to the potential for significantly increasing the adoption of OpenStack to add value to businesses.

Is OpenStack in Your Plan?

If your organization is considering OpenStack, you are faced with numerous decisions to make from the beginning:

- Will you greenfield the deployment, or partner with a vendor-managed distribution (i.e. Red Hat, HP, Mirantis, or other)?
- Will you deploy on KVM or on a proprietary hypervisor?
- Which release and projects will you adopt?
- On what proportion of your infrastructure will you implement and what is your rollout strategy?

- How will you monitor, manage, or control the OpenStack environment?

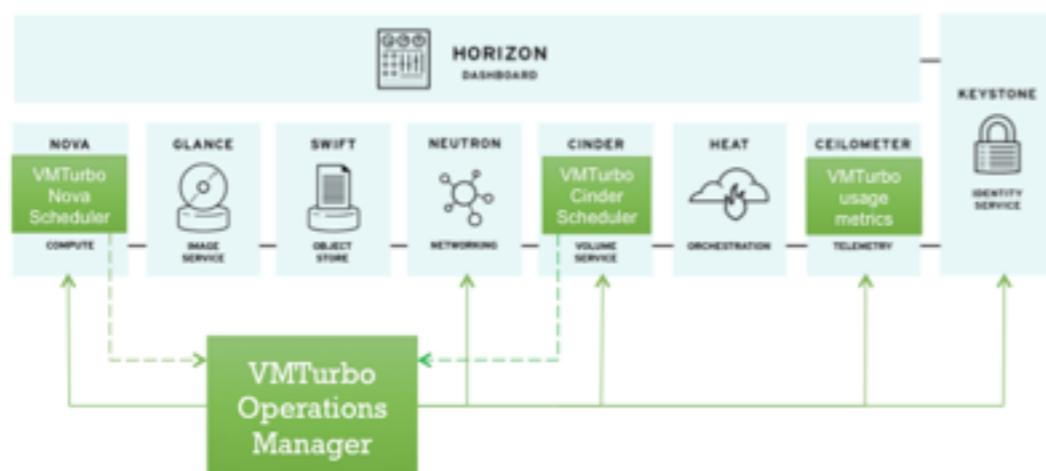
If OpenStack is part of your company's strategic roadmap, we invite you to learn more about the contributions and demand-driven decision automation VMTurbo brings to the OpenStack ecosystem in the next section.

VMTurbo & OpenStack

VMTurbo's Demand-Driven Control platform enables customers to manage cloud and enterprise virtualization environments to assure application performance while maximizing resource utilization. VMTurbo's patented decision-engine technology dynamically analyzes demand from applications, containers, network and VDI to adjust configuration, resource allocation and workload placement to meet service levels and business goals. With this unique understanding into the dynamic interaction of demand and supply, VMTurbo closes the loop in IT operations by automating the decision-making process to maintain an environment in a healthy state.

The OpenStack Control Project

In early 2014, VMTurbo committed a series of longterm investments in OpenStack development, ultimately striving to bring the open source project to feature and performance parity with today's leading proprietary cloud platforms.



Specifically, VMTurbo contributions to the Nova, Ceilometer, and Cinder projects bring capabilities to the OpenStack ecosystem that are unavailable from any other vendor or Git. A full description of the use cases, methods, and benefits are below. For access to all VMTurbo OpenStack contributions, please visit <http://www.github.com/vmturbo>.

Use Case #1: Demand-Driven Lifecycle Control

VMTurbo brings Demand-Driven Control to Open Clouds leveraging the same Economic Scheduling Engine proven in VMware vSphere® and Microsoft System Center environments. This engine, which models the cloud and all of its physical and virtual elements as a market supply chain of resource consumers and producers, aims to continuously satisfy a set of conflicting criteria known as the *Desired State*: a state where application performance is assured while the environment is utilized as efficiently as possible.

Clouds running VMTurbo have been proven to deliver up to **37% faster application response times** and **40% higher guest:host ratios**, above and beyond the initial efficiencies of virtualization.

By instrumenting the **OpenStack Nova Scheduler**, VMTurbo brings Demand-Driven Lifecycle Control to OpenStack instances, from initial provisioning through eventual decommissioning. What this means is that as OpenStack instances are introduced into the environment and as demand conditions fluctuate on existing instances, VMTurbo continuously determines and initiates the best placement of instances in relation to each other in order to maintain the Desired State.

Demand-Driven Lifecycle Control empowers organizations to build the highest-performing, most efficient OpenStack cloud possible.

Use Case #2: Tiered Service Offerings

In OpenStack, virtual hardware templates are referred to as **flavors**. With VMTurbo, it is possible to tier OpenStack instance service levels through pre-defining CPU share flavors and allocating incrementally larger VCPU resources to each. VMTurbo will place tiered instances with respect to their flavor, assuring tiered performance while utilizing the physical infrastructure as efficiently as possible.

from OpenStack.org:

cpu_shares. Specifies the proportional weighted share for the domain. If this element is omitted, the service defaults to the OS provided defaults. There is no unit for the value; it is a relative measure based on the setting of other VMs. For example, a VM configured with value 2048 gets twice as much CPU time as a VM configured with value 1024.

Use Case #3: Minimize Software Licensing Costs

Many organizations are opting for OpenStack specifically to curb or avoid proprietary virtualization technology licensing costs. VMTurbo is capable of controlling OpenStack environments running atop any of the OpenStack-compatible hypervisors: KVM, ESXi™, Hyper-V®, and XenServer®, to mitigate the costs associated with these licenses. It is not limited to only hypervisors, however.

Through a combination of consolidation and rule-based workload placement, VMTurbo has the ability to responsibly minimize software-licensing costs for hypervisors, applications and databases licensed per physical host and per physical CPU socket.

Hypervisors

Commercially available hypervisors are often bundled as a free component of a larger private cloud management suite. Suites, and their paid constituents, are most often priced per physical CPU socket and less frequently, per physical host.

VMTurbo's consolidation analysis demonstrates the extent to which customers may eliminate hypervisor licenses by consolidating their virtual workload to a subset of their existing infrastructure. Although this savings is not factored into the ROI calculation, many organizations do realize this value by choosing to eliminate unnecessary hypervisor licensing after purchasing VMTurbo.

Applications and Databases

Commercially available application suites and databases are often licensed per client or per physical CPU socket. For those licensed per socket, pricing is not quoted as a function of actual utilization, but rather, of potential utilization. For example, a virtualized Oracle® Database with 4 vCPUs residing in a 20-host cluster containing 40 physical CPU sockets would of necessity be licensed for 40 sockets, because the potential exists that the database will use each and every one of the sockets (over some period of time) as it migrates.

By creating rules in VMTurbo, customers can confine applications and databases to a subset of their infrastructure i.e. specific hosts. The Desired State is maintained and potential utilization licensing costs are subsequently minimized.

Given the consolidation efficiencies VMTurbo creates, many customers chose to run their databases on freed-up bare metal servers. If, however, they wish to keep the database virtualized, VMTurbo can safely multiplex workloads on the same host where the database is pinned. The

performance of those co-located workloads will neither hinder nor be hindered by sharing the same physical resource.

Use Case #4: Workload High Availability, Affinity & Anti-Affinity Rules

Users accustomed to operating vSphere- or System Center-based cloud environments are likely familiar with the concepts of High Availability (HA), Affinity Rules, and Anti-Affinity Rules.

VMTurbo enables organizations to deliver High Availability, Affinity Rules, and Anti-Affinity Rules to their OpenStack cloud through a series of logical server groupings and segmentations that heed the interdependencies of distributed applications. Users can define image interdependencies, or specific instance interdependencies, enabling VMTurbo to place instances in a manner that will minimize inter-tier latency.

Use Case #5: Shared Nothing Live Migration

Within the OpenStack ecosystem, if you deploy only the Compute Service (Nova), by default, users do not have access to any form of persistent storage. The disks associated with VMs are "ephemeral," meaning that (from the user's point of view) they effectively disappear when a virtual machine is terminated.

Since ephemeral-backed OpenStack instances are not confined to any particular persistent storage SAN or NAS (which would require Cinder or Swift services), VMTurbo enables shared nothing live migrations of OpenStack instances for performance enhancement, efficiency enhancement, or host maintenance purposes.

This function brings capability to open clouds not natively available without manual scripting or orchestration - which cannot accommodate real-time demand scenarios.

Use Case #6: Host Maintenance Mode

Related to Shared Nothing Live Migration, VMTurbo is capable of evacuating all instances from selected hosts by making a simple API call. Once instances have evacuated the target host or hosts, VMTurbo acts on them using its Demand-Driven algorithm to assure performance while maximizing efficiency on the active infrastructure. Once the host is brought back online, it again becomes a viable option on which OpenStack instances can run.

Use Case #7: Forced Host Selection

When an OpenStack instance is launched, either the end user (or orchestrated flow) must make three decisions: the image to be launched (i.e. the type of application), the flavor of instance (i.e. size and configuration of the template), and the instance name (i.e. a custom name which replaces the UUID). Once the image is launched, the burden falls on Nova Scheduler to find an available compute node on which the instance can run.

VMTurbo has instrumented the Nova Scheduler to force placement on specific hosts, enabling operational maintenance during unplanned guest evasions. Since OpenStack natively abstracts-away specific hardware decisions - presenting aggregated pools of compute and storage - such forced placement becomes a beneficial option, now available with VMTurbo.

Demand-Driven Control for Open Clouds

The promise and momentum of OpenStack are real. As more and more enterprises adopt cloud models to suit the needs of their business, open source continues to gain prevalence, Community members seek to accelerate that movement, and organizations like VMTurbo strive to further the state of open clouds wherever possible.

About VMTurbo



© 2015 VMTurbo, Inc. All Rights Reserved. All trademark names are the property of their respective companies.

VMTurbo's Demand-Driven Control platform enables customers to manage cloud and enterprise virtualization environments to assure application performance while maximizing resource utilization. VMTurbo's patented decision-engine technology dynamically analyzes demand from applications, containers, network and VDI and adjusts configuration, resource allocation and workload placement to meet service levels and business goals. With this unique understanding into the dynamic interaction of demand and supply, VMTurbo is the only technology capable of closing the loop in IT operations by automating the decision-making process to maintain an environment in a healthy state.

The VMTurbo platform first launched in August 2010 and now has more than 25,000 users, including many of the world's leading money center banks, financial institutions, social and e-commerce sites, carriers and service providers. Using VMTurbo, our customers, including JP Morgan Chase, Salesforce.com and Thomson Reuters, ensure that applications get the resources they need to operate reliably, while utilizing their most valuable infrastructure and human resources most efficiently.