Cloud computing datacenters provide computational services and resources, such as data processing, networking, and storage. Typically, they run large numbers of servers arranged in groups controlled by a group controller. The workloads handled by the datacenter tap the server hardware resources to various extents.

End users of such datacenter services expect consistent workload performance, measurable by industry metrics and standard benchmarks. Besides SPEC* benchmarks, Unixbench* and specialized usage workloads are often used to measure the performance of a virtual infrastructure instance allocated to an Intel or other CPU. Benchmarks developed by end users measure compute and I/O performance to inform the build out of virtual infrastructure for the end user application.

Overview
Enterprise IT is turning to cloud to meet demands ranging from faster service provision to lower costs. But shared cloud environments and service level agreements (SLAs) bring new management challenges.

This whitepaper examines - what better methods can be used to provide service level objectives for cloud virtual machines (VMs), detect noisy neighbors, and deliver business value in Openstack™-based private cloud environments.

Deploying cloud workloads: the challenges
Openstack Scheduling Today (Before Implementing Intel® SAA)

Cloud computing datacenters provide computational services and resources, such as data processing, networking, and storage. Typically, they run large numbers of servers arranged in groups controlled by a group controller. The workloads handled by the datacenter tap the server hardware resources to various extents.

End users of such datacenter services expect consistent workload performance, measurable by industry metrics and standard benchmarks. Besides SPEC* benchmarks, Unixbench* and specialized usage workloads are often used to measure the performance of a virtual infrastructure instance allocated to an Intel or other CPU. Benchmarks developed by end users measure compute and I/O performance to inform the build out of virtual infrastructure for the end user application.

In public Infrastructure-as-a-Service (IaaS) environments resources are provided as X number of cores or vCPUs. For example, in Openstack you can use any of the filters available to shortlist the server you want to serve the workload. The compute filters for capabilities, capacity, etc. allow the scheduler to allocate a workload on the system that is best able to serve it.

A cloud end user has applications with workloads either predominantly utilizing memory and data storage or the server processors. In a static workload pattern, consistent performance is a must or performance degradation will deteriorate over time. With no knowledge of these workload patterns, the initial sizing of the infrastructure becomes important. In a dynamically changing environment, an elastic approach to scale up or down based on performance requirements and assurance needs to be built in.
vCPU vs. SCU: Getting Accurate Performance Measurements

A virtual CPU (vCPU) is a virtual machine allocated to a physical CPU. Gartner sums up the difficulty of obtaining accurate performance measurements based on the vCPU: “AWS is a massive scale cloud provider, with a wide variety mix of servers and processor architectures in existence. Therefore, two instances each with 2 vCPU will not necessarily be equivalent. One instance could reside on top of a 2012-based processor while the other could reside on top of a 2014-based processor. Many people have written about the fact that EC2 processor architecture varies across instance types and across regions, even those described as having the “same specs.”[2]

A service compute unit (SCU) provides a uniform number of compute units for performance assurance as a defined and measurable service level objective (SLO). These units are part of the machine flavor definitions.

With Intel’s SCU, IT has server capacity and accurate metrics for assessment performance on Intel based processors.

<table>
<thead>
<tr>
<th>Service</th>
<th>Server</th>
<th>Data Center</th>
<th>Test Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.c1.medium</td>
<td>CA, US</td>
<td>3.44</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.c1.xlarge</td>
<td>CA, US</td>
<td>6.12</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.m1.large</td>
<td>CA, US</td>
<td>3.37</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.m1.small</td>
<td>CA, US</td>
<td>N/A</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.m1.xlarge</td>
<td>CA, US</td>
<td>4.18</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.m2.2xlarge</td>
<td>CA, US</td>
<td>12.99</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>ec2-us-west-linux.m2.4xlarge</td>
<td>CA, US</td>
<td>22.81</td>
</tr>
</tbody>
</table>

Table 1. Cloud Harmony, March 14 “Pi day,” Aggregate CPU performance metric.

It is necessary to know the end user workload requirements and categorize static patterns (e.g., batch-mode applications) or dynamically changing patterns (e.g., interactive workloads) to arrive at the application performance requirements for the workload.

Whether static or dynamic, the user has no means to specify a unit of required compute performance that is measurable and portable. Currently, a service provider has no means to commit to a performance SLA that is constant across platform generation, sites, and varying power and thermal conditions. Data presented in published research[1] shows that performance predictability in an Amazon EC2 varies between 10% and 35%. There is a small variation within the same processor type, and a larger variation based on different processor types for the same instance request. In Table 1, we see 1% to 17% variation in CPU performance at the same location for the same instance type.

Once the virtual machine and image is selected it is assigned to the bare-metal hardware in a rack (Figure 1).

Managing Cloud Workloads: The Solution

Intel® Service Assurance Manager (Intel® SAA)

Intel® Service Assurance Manager (Intel® SAA) is service management software. It’s targeted for environments where enterprise IT administrators want to:

- Start an on-premise enterprise cloud initiative to run workloads on virtualized and cloud infrastructure with performance service level objectives (SLO) guarantees
- Explore integrated hardware and software cloud systems solutions to reduce time to deploy

Intel SAA maximizes the capabilities of Intel® Xeon®-based platforms to enable a “virtual machines” contention detection solution via scheduler models—providing assured performance with noisy neighbor detection, as well as the ability to port workloads between datacenters without compromising on performance.

Cloud Services Catalog

Cloud Services

Catalog

Cloud Scheduler

Service Request

Machine Flavors & App

Service level objectives

Plug-In

SDI - Cloud Machines

Service Assurance Administrator

Resource Scheduling

Machine placement to match resource needs and service management objectives.

Datacenter Infrastructure

Figure 1. End user service flow.
Performance Assurance: Why Service Compute Units Matter

On server platforms, there are partitionable shared resources. Resources are partitionable for the same CPU and threads in the same core, and different cores in the same CPU. There are also multiple generations of CPUs and instruction sets.

To provide consistent performance on any Intel® server platform, Intel created the service compute Unit (SCU). Part of Intel SAA, the SCU is, a consistent unit of measurement for use by the server and VMs. With the SCU, the service provider can monitor virtual machines SLO in the form of Intel SAA units.

In Intel SAA, an SLO is defined for each virtual instance (i.e., a particular number of SCUs), which will be multiplied with the allocated virtual cores to provide the “allowed” total consumption to the VM. For example, if the Instance flavors defined and SLOs are as follows, the SCU SLO management engine contains:

1. A Compute Consumption metric based on the CPUs ability to execute instructions.
2. And a Compute Performance Metric to qualify the SCUs consumed.

Each generation provides increasing performance throughput, based on more SCUs. So, one SCU signifies the equal amount of performance range in any generation of an Intel Xeon CPU. For example, for Sandybridge (SNB), 1 SCU is the same as for Haswell (HSX). However, as each new CPU generation improves in performance, it does offer a greater number of SCUs.

<table>
<thead>
<tr>
<th>Instance Type and RAM</th>
<th>CPU (Virtual Cores x SCU)</th>
<th>Sample SLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra large +30 GB</td>
<td>4 virtual cores with 4 SCUs</td>
<td>High</td>
</tr>
<tr>
<td>Extra large +15 GB</td>
<td>4 virtual cores with 2 SCUs</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 2. Sample SLOs

Intel® SAA Advantages

Intel SAA—a service management framework built and integrated into the OpenStack cloud management environment—brings advanced management capabilities to private cloud enterprise deployments, including:

1. Creating a service management environment to define service level objectives (SLOS) and “tether” to OpenStack flavors prior to launching the workload. Intel SAA defines where SCU is integrated into an application.
2. Building an OpenStack list of compatible hosts able to run the image, based on capacity as measured using SCU metric and system characteristics, and to sort hosts by “fitness” (a metric for prioritizing hosts).
3. Matching workloads on platforms based on capability, capacity, and SLO.
4. Monitoring allocated machines and workloads using deep platform telemetry to detect performance bottlenecks, noisy neighbors, and SLO violations in order for the cloud administrator to remediate performance issues.
5. Assign workloads to run on secure servers with Intel® Trusted Execution Technology (Intel® TXT)3,—providing hardware trust assurance with boot attestation and white-listing for trusted pools.

Performance Assurance: Why Service Compute Units Matter

On server platforms, there are partitionable shared resources. Resources are partitionable for the same CPU and threads in the same core, and different cores in the same CPU. There are also multiple generations of CPUs and instruction sets.

To provide consistent performance on any Intel® server platform, Intel created the service compute Unit (SCU). Part of Intel SAA, the SCU is, a consistent unit of measurement for use by the server and VMs. With the SCU, the service provider can monitor virtual machines SLO in the form of Intel SAA units.

In Intel SAA, an SLO is defined for each virtual instance (i.e., a particular number of SCUs), which will be multiplied with the allocated virtual cores to provide the “allowed” total consumption to the VM. For example, if the Instance flavors defined and SLOs are as follows, the SCU SLO management engine contains:

1. A Compute Consumption metric based on the CPUs ability to execute instructions.
2. And a Compute Performance Metric to qualify the SCUs consumed.

Each generation provides increasing performance throughput, based on more SCUs. So, one SCU signifies the equal amount of performance range in any generation of an Intel Xeon CPU. For example, for Sandybridge (SNB), 1 SCU is the same as for Haswell (HSX). However, as each new CPU generation improves in performance, it does offer a greater number of SCUs.

<table>
<thead>
<tr>
<th>Instance Type and RAM</th>
<th>CPU (Virtual Cores x SCU)</th>
<th>Sample SLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra large +30 GB</td>
<td>4 virtual cores with 4 SCUs</td>
<td>High</td>
</tr>
<tr>
<td>Extra large +15 GB</td>
<td>4 virtual cores with 2 SCUs</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 2. Sample SLOs

Intel® SAA Advantages

Intel SAA—a service management framework built and integrated into the OpenStack cloud management environment—brings advanced management capabilities to private cloud enterprise deployments, including:

1. Creating a service management environment to define service level objectives (SLOS) and “tether” to OpenStack flavors prior to launching the workload. Intel SAA defines where SCU is integrated into an application.
2. Building an OpenStack list of compatible hosts able to run the image, based on capacity as measured using SCU metric and system characteristics, and to sort hosts by “fitness” (a metric for prioritizing hosts).
3. Matching workloads on platforms based on capability, capacity, and SLO.
4. Monitoring allocated machines and workloads using deep platform telemetry to detect performance bottlenecks, noisy neighbors, and SLO violations in order for the cloud administrator to remediate performance issues.
5. Assign workloads to run on secure servers with Intel® Trusted Execution Technology (Intel® TXT)3,—providing hardware trust assurance with boot attestation and white-listing for trusted pools.
Intel® SAA Use Cases

- Create VM Flavors: A flavor is an available hardware configuration for a server. The SCU is used to create VM flavors for the end user to choose (Table 2, section 2). Once the cloud user chooses the flavor (Figure 1), the VM of that size and SCUs are allocated on a server.

- Plan Platform Allocations: SCUs are also used by the cloud provider to plan the size instances that can be allocated to each hardware platform.

How Intel SAA Works

Intel SAA is built to work with cloud orchestrator Openstack. The architecture and control flow is shown in Figure 2. The workload is bundled into one or more virtual machines by choosing a flavor with an SLO measurement. The workload SLO is used by the Intel SAA Controller agent to create a filter for the Openstack NOVA* scheduler. It schedules the VMs to the appropriate CPUs sockets in the nodes. The workload placement module hosts the SLO database, data analysis and collection agents and reporting of workload characteristics, and VM usage and performance trends to the Intel SAA UI. At each host, there is a node agent that uses a reservation share and limit mechanism to allocate the appropriate Intel SAA SCUs and resources according to the flavor chosen. It monitors the data on each VM, and reports SLO violations and captured in the log files.

Shared Resources, Performance Contention, and Noisy Neighbors

Within a server host there are two broad classes of resources: partitionable resources (e.g., CPU cycles) and non-partitionable resources (e.g., cache, bandwidth).

Operating systems have strong support for allocation and scheduling of partitionable resources, which ensures VMs get their fair share of allocation.

The cloud controllers in OpenStack, allocate resources across hundreds of machines in the cluster using the filter (Figure 3). They also use conventional partitionable resources, such as available memory, to make scheduling decisions. The primary challenge lies in allocating non-partitionable shared resources. These are currently unmanaged based on their operating environments. This means allocation is nondeterministic in nature. For example, the share of cache that a VM uses, depends not only on its requirements, but on other neighboring VMs on the same socket at any point in time.

Noisy neighbors are VMs that cause degrading performance and contention side-effects to other running VMs residing on the same server. They are a publically acknowledged problem with few current solutions.
Noisy Neighbors: Example #1

Client A’s virtual machine orchestrator (VM0) lands on a host with performance matching its SLO of 2 SCUs (Figure 4). As new VMs from this or other accounts land on this host, the throughput starts dropping (measured as a performance score). At the same time, the contention score goes up (Figure 5). As a result, you may see a transaction of VM0 1 or 2 slow down. As seen in Figure 6, the transaction of VM0 or the victim slows down by 700 seconds, an approximately 100% slowdown if all neighbors are aggressors.

Noisy Neighbors: Example #2

1. VM1 hosts a task scheduler used in a mail server. It finds the shortest path in the network to deliver emails.
2. The graph in Figure 5 shows the victim VM slowing down almost 50% when running with contention.
3. After taking action we see that the victim’s completion time becomes equal to its solo run.

In Figure 6a the victim is allocated 2 SCUs, and reaches a consumption of 1.32 when it runs without contention.

When aggressors are added, the SCU score comes down and workload throughput is reduced by 50%. This is an extreme case, where multiple aggressors are present. Intel SAA will start flagging aggressor behavior on its dashboard. The service provider administrator can then take an action to move the workload and get the SCU consumption back to the desired level (Figure 6b).

Figure 4: a. Throughput going down; b. contention going up, and; c. transactions becoming slower for the victim due to noisy neighbors.

Figure 5: The victim VM slows down nearly 50% when running with contention. Intel SAA monitors, identifies, and meets SLO in a noisy neighbor situation (Figures 6a & 6b).
Conclusion

SCU is a more accurate way of measuring the performance of VM and server capacity when datacenters are populated with different generations of Intel® CPU platforms. Service providers should consider deploying Intel SAA to provide better customer insight on how workloads are performing against end user contract SLAs, especially, if customer workloads are overusing the stated resource outlined in the contract. Using SCU as a capacity management tool—rather than over-provisioning servers to anticipate VM bursting performance requirements—can also increase server VM density.

Enhancing physical and virtual cloud infrastructure with Intel SAA results in increased manageability and efficiency for enterprise workloads. Intel SAA provides essential software tools for OpenStack, enabling creation of a software-defined infrastructure with enhanced service level objectives.

Learn more at: intel.com/assurance.