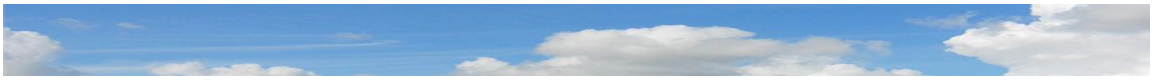


White Paper

Software Applications in the Clouds: Design, Deployment, and Intel Platforms

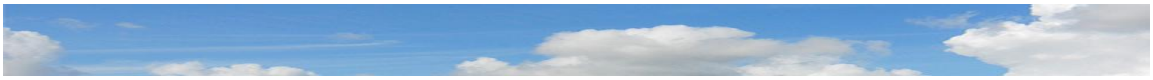
By Kirk Dunsavage, Dave Ott, Rekha Raghu, Stephen Smith

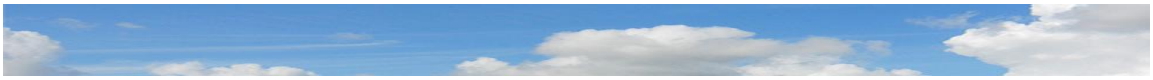
Audience and Purpose of This Paper

Cloud Computing is one of the hottest topics in consumer computing and enterprise IT today. As an aid, Intel has put together some key advice for the Independent Software Developer (ISV) who is developing and delivering cloud applications on Intel hardware.

The audience for this paper are ISVs that are providing software which directly delivers end-user value; i.e., people writing end-user services that are running in the cloud (as opposed to those developing cloud middleware or cloud operating environment services).

What you will know when you get done reading this paper: Some key properties of cloud computing, some high-level principles of software design, development & delivery into the clouds, and finally an understanding of how Intel platform advantages can help you make your applications more effective in the cloud.





1. Cloud Computing – An Overview

1.1 Definition

What is "cloud computing"? Wikipedia describes cloud computing as "a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the internet on a utility basis." Furthermore, "Users need not have knowledge of, expertise in, or control over the technology infrastructure in the 'cloud' that supports them. Cloud computing services are often accessed by the user from a web browser, while a pool of servers somewhere on the Internet executes the software and stores the data for that user."

Let's break down this definition. The first key point is that in cloud computing, users access compute resources remotely over the Internet. Cloud infrastructure providers host services that previously might have been accessed locally from a company's network or even directly on the computers of client users. Not only are compute resources accessed remotely, they are no longer under direct administrative control. The user must now trust a third party for computing services. Both aspects represent a clear departure from traditional enterprise computing paradigms.

Second, cloud computing resources are virtualized. In hosting services, cloud providers abstract the details of where services are running and what hardware resources are involved. In addition, they create they create the illusion that "infinite" resources are available to an application, allowing it to scale up or down to meet user demands as needed. In this way, "cloud" is a metaphor for "an abstract collection of computing entities" servicing a user population over the Internet much like a "virtual data center".

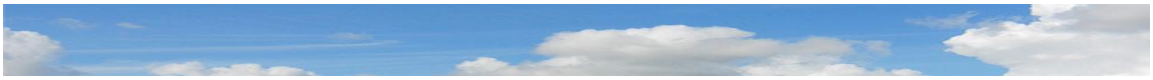
Finally, cloud computing services are delivered on a utility basis. In other words, hosting services are leased from a provider on a pay-as-you-go basis. As we will see later, the terms of this arrangement can vary and may be classified into several different camps.

1.2 Economic Benefits to an Enterprise

Many companies are now using clouds to deliver services. Many ISVs are now using cloud deployment models to deliver their software. Large enterprises, including Intel itself [1] have developed an IT strategy for Cloud Computing. According to a 2009 IDC report, global IT spending on cloud is expected to grow from $17.4 B in 2009 to $44.2 billion in 2013. So, this is the time for an ISV to examine if cloud deployments make sense for their application(s).

Why is this move to the cloud model so popular today? In large part, the answer is economic.

First, the capital investment required for a new enterprise to build and operate a data center is considerable. Not only that, larger data centers are more efficient in amortizing overhead hosts than small data centers as observed by [2]:



> *We argue that the construction and operation of extremely large-scale, commodity-computer datacenters at low cost locations was the key necessary enabler of Cloud computing, for they uncovered the factors of 5 to 7 decrease in cost of electricity, network bandwidth, operations, software, and hardware available at these very large economies of scale.*

Cloud computing thus provides an attractive alternative to new enterprises who wish to quickly deploy software services with little investment. Likewise, it represents an attractive alternative to existing enterprises that might otherwise pay large operational costs to maintain their own computing infrastructure.

Second, the pay-as-you-go (utility) cost model of cloud computing allows enterprises of all sizes to right-size their computing infrastructure and avoid needless costs associated with over-provisioning. In fact, with cloud computing an enterprise can scale up or down their hosting needs *in direct proportion to their customer base*, thus matching their operational expenses to revenue levels, even if they change dynamically over time.

Finally, one might argue that the capital expenditures associated with building computing infrastructure is even more prohibitive when you consider the driving need for many applications to achieve an expansive, global reach. While building a global network of geographically dispersed private data centers is likely to be unrealistic, an enterprise of any size may lease the global infrastructure already offered by today's providers who have been gradually building their infrastructure over many years.

1.3 End-user Benefits

The benefits of cloud computing are not limited to the economics of computing infrastructure and hosting arrangements. In this section we consider end-user advantages that make cloud delivery of application services a compelling proposition.

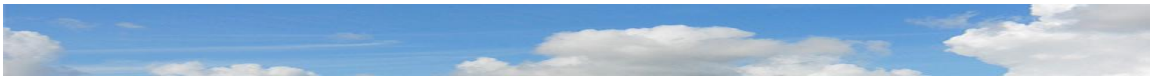
Enhanced User Experience

Cloud implementations of business services build on the paradigm of user interaction of the Web2.0 Era, in which the Web and the Internet becomes a universal standards-based integration platform for these services.

Using cloud delivery, a business will suddenly have numerous services that are able to interact to provide specific functionality with benefits for that business. And it will now be able to quickly build up complex solutions that will reuse existing services while also providing additional business requirements. Such services are woven into ecosystem of software through the web platform.

Available "everywhere"

First, by making all application services available over the Internet, then, any place where a reasonably reliable and fast internet connection is available means that the application's services are also available.



Second, all applications and services are hosted in infrastructure whose maintenance is not provided by the individual user but by a service company that provides this infrastructure for all their hosted customers. This allows high-availability techniques such as virtualized data center across the globe, with automatic backup and distributed caching schemes to minimize transaction latency to be cost-effectively implemented. Thus, the provider's ability to keep computing infrastructure available and healthy is amortized over a large customer population. This can make services cost effectively available 7x24 with acceptable latencies across the globe.

Finally, by embedding service access protocols into a client web browser, all client devices work the same, (modulo their inherent processing power and display/interaction capabilities of the browser). So, as a user, I can access my Google* document from the PC on my desk at home, or from my mobile phone while on in my local coffee shop.

Instant Scalability

Business requirements are highly dynamic and computing services needs to be flexible and nimble to respond to changes. The requirement of supporting 100 transactions per minute during lunch might rise to supporting 500 transaction per minute at 3pm and drop to near zero overnight. With highly flexible cloud implementations, support can be provided for these changed request levels, while still meeting transaction response time and other service guarantees (such as storage requirements, etc.).

The advantages for the end-user in this flexibility are immediately apparent: any service requested meets its guarantees reliably. All of the implement details happen behind the scenes transparently.

Using cloud implementation properties, such as elastic scaling, a service is always available, it always scales to the load you place on it, it responds in reasonable time no matter what the load by increase the compute resource you require, and you are protected from other's consuming your compute resources and thus slowing down your work.

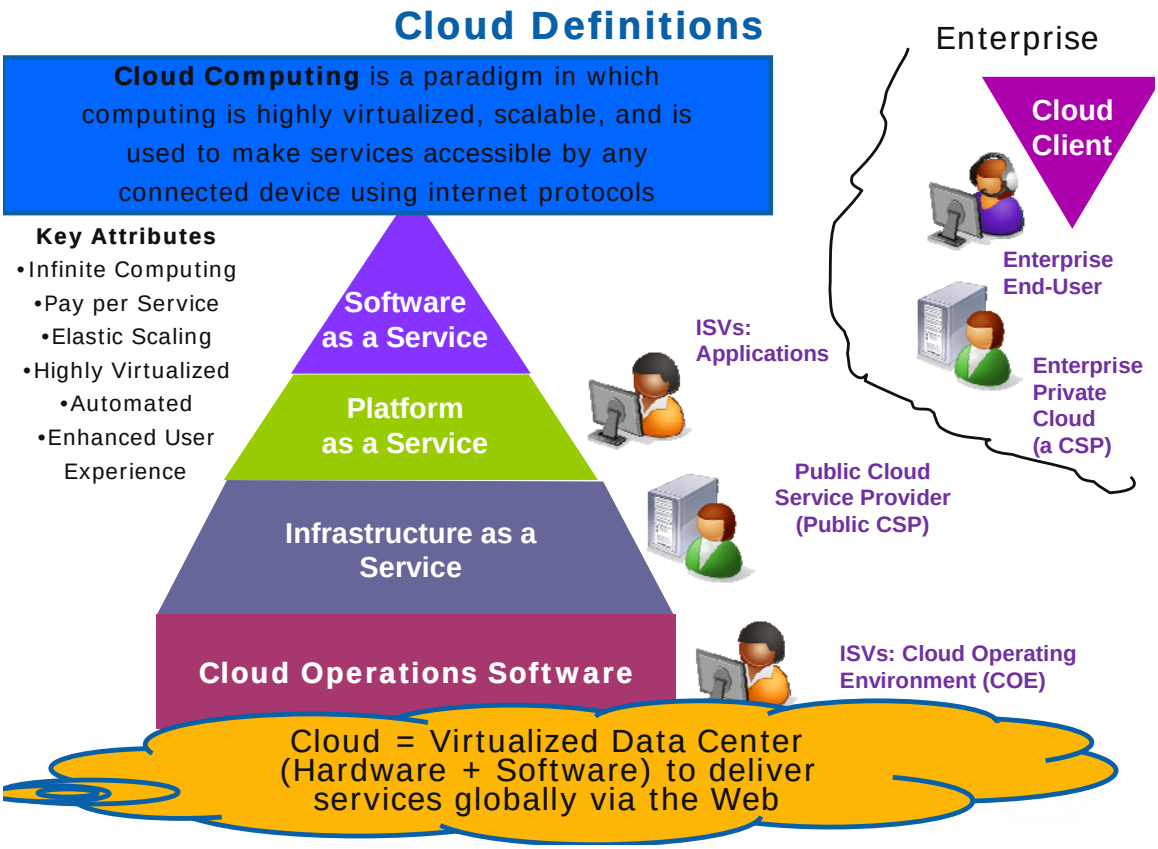
Centralized Cloud Maintenance

Since the cloud service provider provides the software and infrastructure, the end users are removed from the maintenance tasks. In general, policy-based maintenance, security, and upgrade procedures can be applied in a uniform manner across the application's user-base. For instance, users immediately get access to the latest software patches and features without any effort on their part.

The survey in [3] shows that many companies move to managed services get access to better technology than they can purchase in product form. Likewise, 81% of respondents said that better security was a key driver to move to Cloud-based services.

1.4 Cloud Terminology

Figure 1 below contains a quick of many of the key concepts of cloud computing.



* Other names and brands may be claimed as the property of others.

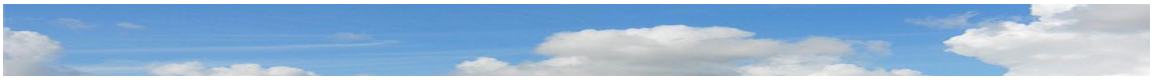
We have already mentioned some key terms. First, ISVs build applications which are then hosted on the clouds. These hosting services we call Cloud Services Providers (CSP). For consumers, these are generally Public CSP, which host many customers simultaneously (multi-tenancy) in a few world-wide large data centers. This is in contrast to a business enterprise, where that business' IT department may use Enterprise Private Cloud to service their internal needs. We should also mention that another class of ISVs exists. These are the folks that build the middleware of the cloud, which we call a Cloud Operating Environment (COE). We will not treat this class of ISVs in this paper.

Figure1 lists some of the key positive attributes of cloud computing. We will cover the middle triangle in Figure 1 in the next sub-section, which describes some of service provider models in cloud computing.

Cloud Service Provider Taxonomy

There is a simple basic taxonomy for public clouds that we will use in the rest of this paper. This taxonomy breaks cloud provider offerings into three basic models dependent on the type and level of service offered. These are: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Software as a Service (SaaS) model: In the SaaS model, software application services and/or full application licenses are available on-demand (over the Internet) via standard Web



technologies/interfaces and are located and provisioned in the service provider's network. This is in contrast to the customers' install-purchased software resident on their own network computers. Generally, SaaS is deployed with some "pay-as-you-go" model, with costs being based on actually usage. SalesForce.com's* CRM application is a prime example of SaaS.

Platform as a Service (PaaS) model: In the PaaS model, the CSP provides an integrated environment to design, develop, test, deploy and support custom applications. The environment and its features are typically only available via a cloud deployment model. Google*'s App Engine and SalesForce's Force.com platform are the prime examples of PaaS. The whole point of these cloud platforms is in being able to design, build, and deploy quality apps that scale well at reasonable cost with minimal development time.

Infrastructure as a service (IaaS): In IaaS models, the cloud service provider dynamically leases out "hardware platforms". These platforms are usually virtual machine images that may come with OS and special middleware (databases, etc.) preloaded, which then execute on the CSP's network of real machines.

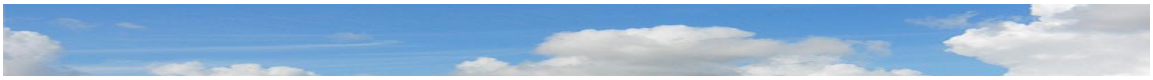
Amazon* Web Services (AWS) has clearly emerged as the de facto standard for IaaS. AWS's Elastic Compute Cloud EC2 offers its users the ability to spin up platforms consisting of an OS preloaded with applications onto "T-Shirt sized" on virtual machines. Other services (which merge into the PaaS model) include online storage (Simple Storage Service or S3) and queuing and monitoring services.

Attached to Amazon is a growing ecosystem of companies, like RightScale*, that add additional features to ease AWS usage and add management and service solutions for businesses. In addition, other companies have extended the AWS framework to work on other cloud provider offerings. Eucalyptus* is an open source project, backed via a commercial support company, that that lets enterprises implement their private clouds that are compatible with many AWS-inspired infrastructure-as-service API bindings.

Note that you, as an ISV, might use any or all of these models in delivering your application. For instance, you might use an IaaS model to host your basic application logic on Amazon's AMI machine images. Then, you might make use of Amazon's PaaS features to build a reliable and scalable persistence layer for your application. And you might use third party SaaS offerings to offer bill services to your users.

You have to make choices on how you build your applications and which CSP delivery model to employ in all these models, the cloud offering from Intel is still important to you. If you use IaaS offerings close to the "bare metal", Intel's features and performance optimizations can be vitally important to delivering cost effective contributions. For service providers offering highly abstracted software hosting frameworks (PaaS), hardware features and performance hardware performance characteristics may seem more pertinent to the authors of the middleware framework. Yet, you will find some "gray box" [4] understanding of these will still be important for you to effectively deliver low-cost applications.

* Other names and brands may be claimed as the property of others.



1.5 Summary of Remainder of Paper

In the rest of this paper, we focus on ISV's interested in placing applications into the clouds. We characterize architectural properties of an application that work well in the clouds, technical considerations in achieving those cloud deployments, and potential questions and issues that might arise. We highlight things Intel is doing to make your cloud deployments easier and more cost effective. We present an argument about why your use of Intel hardware in your development and testing is still vital to your success. Finally, we close with a simple checklist to follow to host in the clouds.

2. ISV "Cloud Ready" Applications

While cloud computing offers many new advantages and opportunities, designing or modifying software to meet its deployment context can seem overwhelming. In this section, we discuss application types and technical considerations for making software "cloud ready".

2.1 Which applications are best suited for Clouds?

Are some applications better suited for cloud computing contexts? We believe the answer is "yes", and that it is important for software designers and adopters to understand why. Below we discuss key attributes that make enterprise software strongly matched to the cloud computing context. These attributes both analyze cloud applications currently already in deployment today as well as provide a set of guidelines for architecting new software in a way that exploits the strengths of cloud computing.

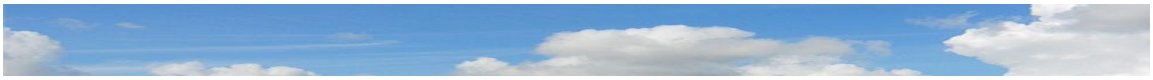
Audience Reach

Reach refers to the size or extensiveness of a software product's audience, either in terms of user volume ("seats") or number of business clients. The more an enterprise application is designed for use by a large audience of users, the better matched it is for the cloud computing environment. Conversely, the more software is designed to solve highly specialized problems for a very narrow audience, the less well-matched it may be.

Software serving a broad audience is most often highly generalized in its underlying logic and capabilities, while at the same time offering a considerable degree of customization at the user or client organization level. A canonical example is CRM (Customer Relations Management) since almost every business, large or small, has similar needs with regard to managing customer accounts, though the details of which will vary by organization. An application poorly matched to the cloud paradigm might be a domain-specific medical application written for a particular clinic or research organization to solve a problem unique to that organization and with little transferrable value.

High Availability

High availability refers to ease of software access. Software is highly available when the same software tools and environment can be accessed in a variety of user settings (home, work, business travel, guest or public computers, etc.) and geographic locations (corporate sites, regions of the United States, Europe, Asia, etc.). In general, the more an enterprise application supports high availability, the better suited it is for the cloud computing



environment. Conversely, the more software is designed for limited access and highly controlled physical environments, the less well-suited it is.

Web applications like search engines or online retailers provide a vivid example. Each is accessible with a simple Web browser and without regard to the user's compute setting or geographic location. The point here is not merely that application software exploits Internet connectivity and Web-based location independence, but that application designers have developed sophisticated cloud-based infrastructures to ensure a consistent, high-quality user experience whenever and wherever a user connects. An application that is poorly matched to cloud computing in this respect might be a highly guarded file server storing corporate trade secrets. The server is not intended for access outside the company network or even from many places within the network.

Scalable and Elastic

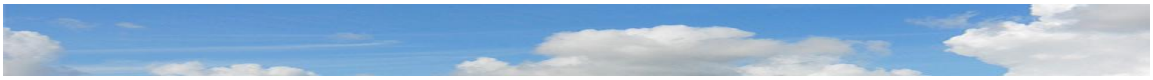
While the reach attribute considers the audience of an enterprise application, the *scalability* property addresses the structure of the software application itself. A scalable application is one that can be dynamically resized to increase the amount of data it processes, the number of users it services, or, stated more generally, the amount of work it is capable of accomplishing per unit time. *Elasticity* refers to an application's ability to scale up with increased demand and scale down with decreased demand. Since cloud hosting services are typically metered, the ability to adapt sizing to usage patterns is important. This may mean scaling up as the software audience grows, scaling up and down to follow seasonal or event-driven patterns of use, or simply following diurnal patterns of computer usage by a user population co-located in the same geographic region.

In general, the more an enterprise application is capable of scaling up and down, the better suited it is for the cloud computing environment. Conversely, the more software is designed to handle a single, well-defined load level, the less well-suited it is. Once again, good examples of scalable application services are plentiful on the Web, for example, online retailers or social networking sites. Poor examples can often be seen in in-house, custom-built applications within a company. While filling their intended requirements well, they are often not designed to scale in an external computing environment outside the company.

Platform, Location, and Framework Independence

An application that is *platform* and *location independent* can be run on any hardware system, anywhere. That is, it does not make strong assumptions about the underlying hardware system or its exact location within a LAN or WAN. The more an application can achieve such independence, the more it is highly suited for deployment in a cloud computing environment. Conversely, the more an application in tied to specific hardware platforms on which they have been developed and optimized, or tied to a specific network configuration, the less well-suited it is.

Framework independence refers to flexibility in the software context of deployment. An application that can be deployed using a wide variety of software frameworks is highly suited to the cloud computing environment. For example, an application could be deployed as a Microsoft* Azure service, using Google App Engine, or on Amazon Web Services' EC2.



An application strongly coupled to a specialized software environment and with a long list of requirements would be a poor match for the cloud computing. For example, a particular application that can only run on the RISC architecture and assumes a legacy operating system version with outdated libraries for compatibility.

Latency and Data Consistency Tolerant

Applications and services running in the cloud computing context typically represent an increase in physical distance between users and actual computing infrastructure. As such, applications that are a good match for cloud computing are tolerant of propagation delay and network latency due to the shared nature of computer networks generally. In contrast, an application that is highly sensitive to delays (e.g., a real-time media capture system) would be a poor match for the cloud computing environment. Applications like the latter, however, can often be restructured to place delay-sensitive elements at the client and non-delay sensitive elements within the cloud where the benefits of cloud computing (e.g., high availability) can be fully exploited.

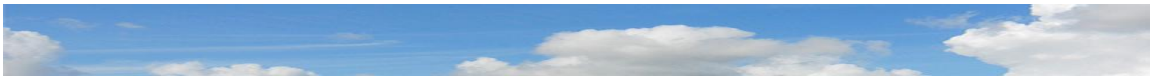
Applications that are highly distributed geographically are also a challenge in terms of *data consistency*. The more an application can tolerate temporary inconsistencies in global state and stored data, the better suited it is for highly distributed deployment in a cloud environment. For example, content updates or customer orders in one region may take time to propagate to other regions, but users may continue to use the system without long update delays. Conversely, applications that demand a stringent consistency model, both in global state and stored data may be poorly matched. For example, some real-time database systems are difficult to deploy in a highly distributed computing context. The interested reader can refer to [6] for an interesting review of general distributed computing issues which many developers tend to ignore.

Web-based Protocols

Applications that make use of client-server communication mechanisms and Web-based protocols are a strong match for the cloud computing environment. Such applications use TCP port 80 since, as the canonical port for HTTP Servers, it provides a universal portal through most every company- and consumer-based firewall. This wide connectivity gives the natural benefit of location independence and "almost everywhere" reach. For an application that uses a desktop GUI-based approach with private connections to backend servers, a move to HTML protocol interactions with backend components is critical. Re-designing the GUI with an AJAX-based web browser interface support might likewise be timely.

Service-Oriented Architectures

Applications that have been partitioned into well-defined services using service-oriented architecture (SOA) design techniques are also well-matched to the cloud computing environment. Services benefit from the inherent properties of cloud computing (reach, high availability, location independence) while client applications built on top of such services benefit from the inherent flexibility and loose coupling provided by service-oriented



architectures. Much could be said about how to go about partitioning an application using the SOA approach. The reader is referred to [5] for more information and advice.

2.2 Application Readiness Technical Considerations

Assuming that an enterprise application is well-suited for deployment in a cloud computing environment, the question now becomes, "what technical issues should a design or migration team consider to make sure their application is cloud ready?" In this section, we raise a number of issues that developers need to address to ensure a successful cloud deployment.

Resource Usage Requirements

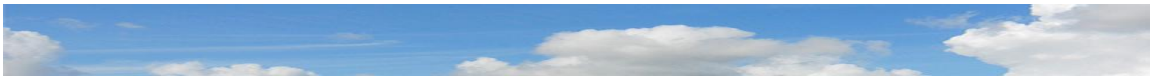
An application running in a single-user environment may have loosely defined resource requirements that speak to minimal hardware and operating system requirements for the application to run well. In the cloud computing context, however, an application's resource requirements need a more explicit profile for several reasons. First, the hosting service needs this information to effectively provision compute resource within a data center that is shared by many clients simultaneously. From the customer's standpoint, explicit resource profiles help avoid both over-provisioning, which creates needless expenses, and under-provisioning, which hurts the software user experience. From a cloud administrator's point of view, explicit sizing information is needed to create virtual machines, a topic we discuss in the next section.

Understanding the resource usage requirements of an application takes careful study and may involve more than one strategy. Minimally, an application's use cases must be cataloged and then exercised within an environment instrumented to gather resource profiling data, including CPU, memory, disk I/O, and network I/O requirements. A user model can be constructed, and then simulation software developed that drives application use in a realistic way. Application code may also be engineered to log resource usage levels internally or, better yet, to run at prescribed resource usage levels that can be configured explicitly by an administrator.

Quantifying application resource needs can be complicated. Many applications have complex functionality and a single set of numbers may not describe the range of behavior possible in realistic usage scenarios. One approach may be to look for maximum or average levels of usage, or to distill complex usage patterns to sets of summarizing use cases. Another problem is that of scaling. Resource usage may depend entirely on the number of users which cannot be predicted beforehand. In this case, the profiling goal becomes to quantify constant overhead and then to determine a per-user resource usage estimation. Cloud provisioning can then be parameterized to follow from the number of current users and projected load given current trends and historical data.

Virtualization

Virtualization is a cornerstone technology for cloud computing environments. In part, this is because Virtual Machine Monitors (VMMs or *hypervisors*) provide a convenient framework for multiplexing system resources among multiple applications, each running in its own operating system environment. Virtual Machines (VMs), as such encapsulations are called,



provide many additional benefits as well: they are capable of running on heterogeneous hardware platforms without recompiling, they can be easily replicated or deleted to scale an application up or down dynamically, they can be migrated from one physical system to another to take advantage of under-utilized or better-performing hardware, and they make possible the consolidation of applications with heterogeneous operating system requirements on the same system.

Not only is virtualization frequently used as an underlying technology, but many cloud computing providers define their computing services, provisioning units, and billing arrangements directly in terms of VMs. This is especially true for IaaS providers (e.g., Amazon EC2) who offer abstracted computing environments with a choice of operating system. Environments may be utilized without knowing the exact location or the hardware specifications of the underlying physical platform. Best of all, it may be scaled up or down in an elastic manner based upon user demand, with billing arrangements following a pay-as-you-go charging discipline.

With this in mind, enterprise application developers need to insure that their cloud-based application is "virtualization ready". First, they must understand their application resource requirements (see the previous section) and be ready to make them explicit in for the purposes of VM sizing. This may include a "fitting" process whereby an application is matched with a set of possible VM sizes offered by the cloud provider.

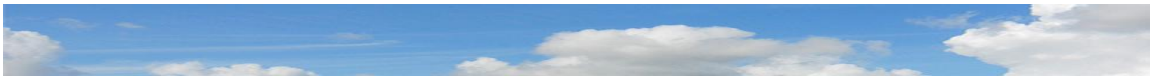
Second, application developers should also study the performance of their applications within virtualized environments in order to understand its behavior compared to native execution environments. While hardware virtualization technology and VMM software are continually evolving, there is inherent overhead in virtualizing system resources and multiplexing them among numerous VMs. Application developers should understand this impact and isolate performance bottlenecks that stem from virtualization, for example, system calls that needlessly invoke VM exits and hamper application performance because of it.

Finally, application designers need to insure that system-level calls and programming techniques do not make inadvertent assumptions about the underlying hardware platform. A common example is making strong assumptions about the results of system time calls, not realizing that virtual time as delivered by a VMM can be somewhat elongated or compressed.

Scalability and Elasticity

The notions of *scalability* and *elasticity* were explained in the previous section. In general, they refer to an application's ability to increase or decrease it capacity for work dynamically, based on workload demand. A simple problem scenario might be a Web-based retailing application that experiences surges in workload periodically due to holiday shopping patterns and national economic trends.

A fundamental strategy employed in almost every highly scalable application is that of *parallelism*. Applications handling a large volume of work may typically be decomposed into smaller units of work, many or all of which can be executed concurrently. *Data parallelism*



refers to the decomposition of complex data sets into smaller units or "chunks", each of which can be operated on in parallel before reconstructing a global result. Common examples include image processing and vector or matrix processing in scientific applications. *Task parallelism* refers to the decomposition of complex services into independent tasks, each of which can be executed in parallel. Web servers or databases, for example, can service multiple users in parallel.

A programmer can apply multi-threading wherever data or task parallelism presents itself. Fortunately, threading libraries are widely available on virtually every operating system platform and by most modern programming language run-time environments (e.g., Java, C#). Furthermore, today's operating systems will automatically manage the assignment of threaded tasks to available cores on today's multi-core server platforms to exploit hardware parallelism of the underlying system. The effective combination of data/task decomposition, multi-threading, and multi-core processor hardware contribute to effective vertical scaling on a single system.

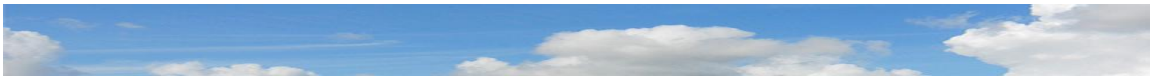
Data parallelism may be further expanded through horizontal scaling, or the use of additional servers. Applications with a high degree of task and data independence can often easily benefit by increasing the number of application instances available on a networked cluster. Load balancing or handoff techniques can be used to distribute user load across available servers. Message passing may be used to coordinate processing between differentiated components of a distributed application. Replication across large geographic distances often requires periodic synchronization, the frequency of which is application dependent.

Instrumentation for making elastic scaling decisions may be demand-, performance-, or resource-driven in some combination. Demand-driven adjustments may create a trigger based on the user or request number thresholds. Given well-understood processing capabilities, application instances can be created or deleted to adjust to a smoothed average demand. Performance-driven adjustments may use response time as a trigger for application scaling changes. This approach is often tuned to quality of service requirements taken from a user's perspective. Finally, resource-driven adjustments consider system resource utilization levels like CPU and memory as a trigger for adding or removing application instances.

Provider Software Frameworks

Cloud computing providers vary in the software framework provided to deploy cloud applications on their infrastructure. At the time of this writing, Amazon Elastic Compute Cloud (EC2) supports the creation of custom VMs (called Amazon Machine Images, or AMIs) that can be replicated on demand. Additional storage, database, and queuing services are available to support an application as well. (See http://aws.amazon.com/ec2.) Google App Engine, in contrast, supports an abstracted application deployment framework using Java* and Python* runtime environments. Part of their deployment scheme includes automatic scaling and load balancing services. (See http://code.google.com/appengine.)

In general, provider frameworks can be thought of as lying on a continuum between bare infrastructures and highly structured, highly abstracted runtime environments. In the



former, application developers have the freedom to choose among operating systems, programming languages, runtime environments, and supporting software components like databases or web servers can be selected and customized as needed. At the same time, the burden of software installation and configuration falls upon application developers directly, thus requiring considerable expertise to manage the complexity.

In contrast, the structured runtime environments offer easier development and deployment environments since the provider has abstracted operating system and runtime environment considerations entirely from the user. Furthermore, they may provide templates and libraries that guide application creation and manage deployment and scaling complexity with easy user configuration. The downside, of course, is flexibility. Such environments may not offer the desired programming language, or may not support certain libraries or system APIs that are assumed by an application. This may be particularly hard on pre-existing applications that would ideally be ported to the cloud computing environment rather than re-written from the ground up. In general, such environments also require a tighter coupling between an application implementation and the deployment framework itself which may not be desirable.

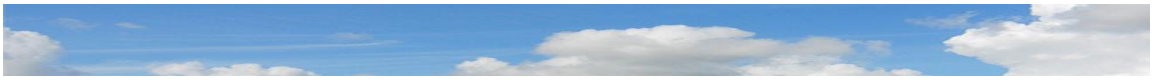
Each application developer will need to choose the best alternative on the continuum between bare infrastructure and highly structured runtime environments. Among the considerations are the tradeoffs between ease of deployment, flexibility, and degree of coupling between the application and its deployment framework.

Security

Security remains a contentious subject in cloud computing because the apparent "trust" required in using hosted services. For example, clients must trust the physical security of their provider's data centers, the integrity and privacy of their database infrastructure, the robustness of their authentication systems, the effectiveness of their network intrusion detection mechanisms, and the thoroughness of their employee screening practices. Application isolation is also a natural concern among organizations considering cloud hosting options since the multi-tenancy model inherently assumes the sharing of resources with other applications.

Interestingly, many analysts point out that cloud computing are no less secure than most private, in-house IT services. First, network accessibility is typically a requirement for internally hosted applications, thus leaving them similarly vulnerable to network-based attacks. Meanwhile, however, cloud providers often have considerably more expertise and capital resources to implement robust network security solutions compared to in-house IT services. Virtualization also provides a powerful mechanism for application isolation within multi-tenancy arrangements. The use of encryption for message passing between servers and for storage systems and databases provides the same level of security used within in-house systems, especially when keys are managed by application developers and not cloud computing providers.

Still, many organizations find security to be a difficult subject. There is no substitute for closely examining the security practices and policies of a potential cloud provider, or for



leveraging robust security technologies within application design. As the industry evolves, also look for new standards to emerge that address this issue.

3. Deployment Considerations

Once you have an application you have developed for a cloud environment, the next step in the process is the actual deployment of that application. Deployment into a cloud environment involves several key issues that need to be addressed, some of which match those considerations for deployment in traditional scenarios, but others that are certainly unique to cloud environments. This section will help clarify and articulate several of the most important areas of concern for cloud application deployment.

3.1 Ability to Monitor & Control

A key aspect of running applications in the cloud is the fundamental ability to be able to monitor and control applications currently running there. A Cloud Service Provider that is maintaining the cloud infrastructure requires the ability to bring on new hardware and retire old hardware as needed to support changing application demands. The provider also needs to be able to control applications whether than means limiting application resource usage under certain critical conditions, moving applications to different hardware as new hardware is spun up, configuring new virtual machines to run ISV applications, etc. To actually perform this level of control, a provider needs to also be able to actually monitor applications. Without these two key abilities, the basic usefulness of the cloud tends to fall apart.

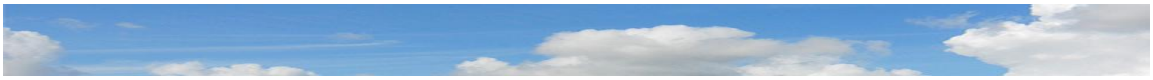
Several of the current CSP's provide well-defined mechanisms to monitor applications in their environments.

- In Amazon's E2C environment, the AWS Management Console is a convenient point-and-click web interface that permits one to view and manage all of the resources being utilized within your cloud environment. It now also ties into the AWS CloudWatch interface which then also provides insight into the real-time operational metrics of your application including CPU utilization, and disk and network I/O for each of your E2C instances.
- With Google's AppEngine environment, the Admin Console allows you to create new applications, view data logs, analyze traffic to/from your application, view tasks, as well as test new versions of your application. To actually monitor metrics such as CPU utilization, data store and memory caching details, you have to utilize pre-defined APIs in your application to do so.

Therefore, mechanisms clearly exist to do just about anything you might need to do around your cloud application. The major difference, as clearly shown by the two examples cited, is in the level of difficulty and user-friendliness afforded by each specific environment.

3.2 Metering & Billing

Closely related to the ability to monitor and control your cloud application, is the key ability to not only monitor and bill appropriately for your customers but the ability to keep track of the resources your own application is using. We will refer to the former as "customer metering" and the later as "cloud provider metering".



In terms of "cloud provider metering", CSP's have built in very clear and accurate infrastructure usage and metering capabilities that they use to track the number, types, and amounts of resources you cloud application is consuming.

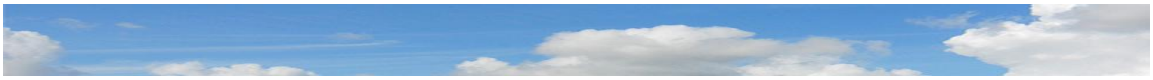
- Google's AppEngine infrastructure, for example, tracks all of this information internally for any application running on top of AppEngine and tracks this usage against a fixed set of quotas. Your application will run essentially for free until its resource usage passes a certain threshold and then Google will begin to bill you for the resource usage beyond that point. Google's Checkout billing infrastructure is then tied directly to its metering/quota infrastructure so they know exactly how much to charge you for use of their infrastructure.
- As with the Google infrastructure, Amazon's E2C environment also tracks all resource usage for your application and provides a much more elaborate billing scheme that Google's AppEngine. With the E2C environment, you can chose (and pay accordingly) for different types of compute instances based on how resource intensive your application is as well as the intended operating system to be used along with the intended data transfer rates for your application. To actually monitor the real-time performance of your application via Amazon's CloudWatch service, an additional charge will apply. Overall however, it is very clear that Amazon has an intensive infrastructure in-place to accurately track the application usage details for your cloud applications.

With respect to "customer metering", this is an entirely different story. The key issue here is whether or not the particular CSP chosen to host your cloud application actually provides a mechanism for applications running within its environment to meter and bill customers using those applications. This is a tricky situation and requires additional infrastructure that might or might not be trivial to implement depending on the level of the cloud taxonomy the particular CPS is working at (e.g. IaaS, PaaS, etc.). On the one hand, Google's AppEngine environment seems to provide little in the way of mechanisms for a cloud application developer to properly meter and bill for usage of their application, whereas Amazon's E2C environment provides specific support for this via their DevPay service. Therefore, this is another key differentiating factor that needs to be considered.

3.3 Verification

Given that many of the current environments provided by the existing set of CSP's are built upon well-known and preexisting technologies, many tools and techniques that already exist can then be applied to cloud applications.

For example, Google's AppEngine environment provides a choice for cloud application developers to base their application on Java or Python, both well-known and established languages with numerous and varied development and testing tools in existence that can then be applied in the cloud space. Tools like JUnit and Eclipse can very easily be applied to cloud applications developed for this environment and Google also provides sets of its own internally developed testing scripts for developer use as well.



With respect to Amazon's E2C environment, given that it is at the PaaS level of the cloud taxonomy, choices here for development, testing, and debugging are even more varied and widespread and can be found elsewhere. Amazon doesn't impose necessarily any specific development languages but does provide the AWS Toolkit for Eclipse which lends itself to supporting Java development.

In addition, several third party vendors are now building and supplying tools to help test cloud applications. SOASTA's* CloudTest application is one such application testing environment designed to exercise and test a cloud application. CloudTest provides a UI-based all-in-one testing tool that can test any sort of cloud application, whether it is based upon SOAP, HTTPS, JSON, Ajax, etc. so it provides yet another option.

All in all, development for a cloud environment adds a few new twists to the development picture, but isn't all that much different than developing outside of the cloud. The difference lie more in how performance is managed in a cloud environment for applications.

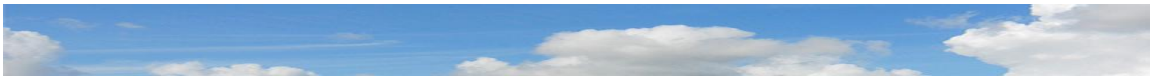
3.4 Performance

As opposed to the verification and testing of a cloud application, the performance monitoring of such an application can be very different relative to that of a non-cloud application. Given that both you and your customer's will be charged for use of cloud-based services and obviously expect a certain level of responsive and performance for what is being paid for, it is fundamentally key to be able to monitor and manage the performance of cloud applications and today's offerings definitely don't fall flat in this area.

Amazon's E2C environment provides numerous mechanisms to monitor performance of instances running in its environment via the AWS Management Console and its CloudWatch service. There are definitely the key tools for doing just about anything within the Amazon cloud environment and permit a user of E2C to closely monitor all happenings with respect to their software and then take appropriate actions as necessary. Google's AppEngine provides the Admin Console that corresponds to Amazon's Management Console to do similar things. As was also mentioned before, third party software vendors have also jumped into the fray here as the previously discussed SOASTA CloudTest tool also provides a certain level of performance management for cloud applications.

A side-benefit of being able to monitor the performance of your cloud application is the ability then to actually act on that information dynamically as you see fit. Amazon's E2C environment automatically supports elastic load balancing by default and then also leverages the capabilities of its CloudWatch service to then provide additional value-add for its customers by supplying other infrastructure capabilities like auto-scaling. This feature automatically supports scale-up or scale-down of hardware and software requirements for a particular application based on an initial set of parameters determined by the application developer.

* Other names and brands may be claimed as the property of others.



3.5 Software Maintenance

Finally, the last major deployment consideration centers on software updating and maintenance. All in all, given that today's CSP's are providing a base layer of infrastructure and software to build your cloud applications upon, this area seems to have been left mainly up to the developer and existing updating and maintenance technologies. As such I would consider this an area of improvement for the existing CSP's and a differentiating area for any new CSP's that come into the picture.

4. What is Intel doing in Cloud Computing?

To recap, we have already discussed the software and deployment considerations to make an application "Cloud Ready". Next, we will discuss what Intel is doing to help address some of your cloud computing requirements and challenges.

Clouds demand performance, energy efficiency, virtualized infrastructure, and specialized technologies, platforms and software, all optimized for the large scale datacenter deployments found in cloud service provider infrastructures. Intel is driving several optimizations and initiatives to support large scale server deployments in the cloud with the primary intent to:

- deliver optimized platform for performance, energy efficiency and virtualization
- deduce data center power consumption
- work with the Cloud industry to help set standards and drive initiatives

4.1 Cloud-Optimized Platforms

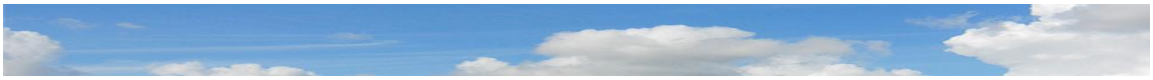
Our latest innovation in processors includes Intel® Xeon® processor 5500 series [7, 8] which provides a foundation for IT management to refresh existing or design new data centers to achieve greater performance while using less energy and space, and dramatically reducing operating costs.

The Intel® Xeon® processor 5500 series offers several features that help it make the best performing server in the industry. Some of these features include:

1. intelligent performance that automatically varies the processor frequency to meet the business and application performance requirements.
2. automated energy efficiency that scales energy usage to the workload to achieve optimal performance/watt and reduce operating costs.
3. flexible virtualization that offers best-in-class performance and manageability in virtualized environments to strengthen the infrastructure and reduce costs.

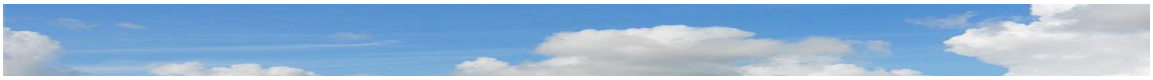
Performance

Application performance is critical for day-to-day business operations, as well as creating new products and reaching new customers and markets like cloud. The Intel Xeon processor 5500 series, with Intel® microarchitecture code-named Nehalem, brings intelligent performance to the most trusted server architecture. The Intel Xeon processor 5500 series brings together a number of innovative technologies to enable intelligent performance: **Intel® Turbo Boost Technology** delivers performance on demand, letting processors operate above the rated frequency under certain conditions to speed specific workloads. Intel®



Turbo Boost Technology converts any available power headroom into higher frequencies. In those situations where the application requires maximum processing power, the Intel Xeon processor 5500 series increases the frequency in the active core when conditions such as load, power consumption and temperature permit it. **Intel® Hyper-Threading Technology** provides 2 threads per core, benefits from larger caches and up to 144 GB memory capacity, delivering greater throughput and responsiveness for multi-threaded applications. **Intel® QuickPath Technology** and an integrated memory controller with next-generation DDR3 memory and 8-MB L3 cache speed traffic between processors and I/O controllers for bandwidth-intensive applications. Servers based on the Intel Xeon processor 5500 series can have up to 2.25x the memory capacity compared to the previous server generation [7].

Intel® Xeon® Processor 5500 Series Features	Description	Benefits
Higher Performing Cores	A number of architectural improvements enable greater instruction level parallelism and more efficient overall processing.	Increased throughput and better performance
Intel® QuickPath Technology with Integrated Memory Controller	A new high-speed point-to-point interconnect subsystem increases peak communication bandwidth many fold among processing cores, memory, and I/O devices.	Faster access to data increases core utilization, which enables better processing efficiency and faster time to results
Shared Level-3 Cache	The large, 3rd level cache stores more data on the processor die and delivers it faster and more efficiently to the processor cores.	
Intel® Hyper-Threading Technology	Each core can process two simultaneous software threads (16 threads for a standard server with two quad-core processors).	Configuring the software to support 16 threads (versus the maximum of 8 on the previous generation server) provides performance boost
Intel® Turbo Boost	Technology Processor clock frequency can be dynamically adjusted to boost performance without exceeding the processors thermal design point (TDP).	Helps1 achieve higher performance gain
Native DDR3 Memory Support up to 144 GB	Enables larger memory configurations using fast, affordable, industry-standard	As data sets and processing workloads increase, larger memory



	DDR3 memory modules.	configurations could potentially deliver additional performance advantages

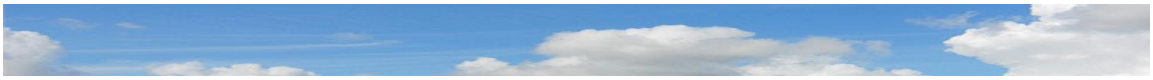
Energy Efficiency

Efficiency is one of the most important requirements of cloud infrastructure – as every watt, every server, and every square foot of the datacenter must be fully utilized to maintain profitability. Industry benchmarking of performance per watt – as measured by SPECPower* – also testifies to leadership of the Intel Xeon processor 5500 series in efficient performance, as servers based on this new generation of processors have leapfrogged the previous top dual-socket score (based on prior-generation Intel Xeon processor architecture) by at least 64 percent [7]. In addition, the Intel Xeon processor 5500 can deliver up to 2.25X the computing performance at roughly the same system power compared to the previous-generation Intel Xeon processor 5400 series. For cloud architects seeking to maximize performance and efficiency, the choice of Intel Xeon processor 5500 series is quite clear.

Intel® Intelligent Power Technology enables policy-based control that allows processors to operate at optimal frequency and power, boosting performance for critical workloads while lowering TCO through overall reduction in power consumption. Within a single server, Intel® Intelligent Power Technology minimizes power consumption when server components are not fully utilized. We will discuss this in detail in section 4.2. Two critical components are:

- **Integrated Power Gates** based on Intel® 45nm high-k metal gate silicon technology allow individual idling cores to be reduced to near-zero power, independent of other operating cores, reducing idle power consumption to 10 watts, versus 16 or 50 watts in prior-generations of Intel quad-core processors. This feature reduces server idle power consumption by up to 50 percent [7] versus the previous generation of two-socket server processors.
- **Automated Low-Power States** together with the operating system, will help achieve CPU clock frequency scaling by automatically putting the processor and memory into the lowest available power states to meet the requirements of the applications currently running. Processors are enhanced with more and lower CPU power states, and the memory and I/O controllers have new power management features.

To understand how the Intel Xeon processor 5500 series realizes its power efficiency, first consider the Advanced Configuration and Power Interface (ACPI), an open industry specification for operating system-directed power management. In this scheme, the operating system conspires with system hardware to choose low processor power states (C-states) whenever idle intervals occur during operation. Intel Xeon processor 5500 offers a new C6 state for use within this ACPI control scheme, achieving considerable power savings through various techniques that power down internal hardware components temporarily. Likewise, Intel Xeon processor 5500 offers a range of processor performance states (P-states) that support dynamic processor frequency adjustments. Once again through ACPI, the operating system can conspire with system hardware to adjust the processor frequency



to lower power consumption whenever processor demand wanes or remains at low (but not idle) levels.

Virtualization

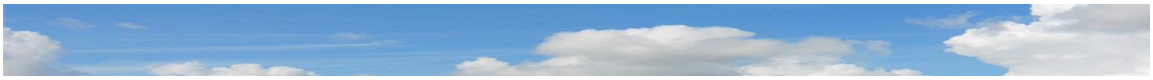
Virtualization is the foundation to cloud. Cloud computing models that provide virtualized hosting environments are highly dependent on the capabilities of their servers to provide sufficiently powerful virtual machines. From the operator's perspective, the more virtual machines that can be realistically provisioned on a given server, the better the return on investment. With hardware virtualization, combined OS/Application images can run without change on non heterogeneous hardware, be consolidated down to a cost-effective reduced set of hardware, and even be migrated to high-performance system as required.

Intel believes in a holistic approach and has accelerated the hardware virtualization innovations called Intel® Virtualization Technology [9] to offer a umbrella of optimized features in processor (Intel® VT-x), chipset (Intel® VT-d) and network (Intel® VT-c). Intel works with VMware*, Microsoft*, Citrix*, Parallels* and many other virtualization software vendors to help ensure that Intel virtualization technologies are broadly supported in today's and tomorrow's solutions, so they deliver high value while being completely transparent to IT organizations and end-users.

Intel® Xeon® Processor 5500 Series Virtualization Features	Description	Benefits
Processor: Intel® Virtualization Technology (Intel® VT-x) • Intel® VT FlexMigration • Intel® VT FlexPriority • Intel® VT Extended Page Tables	Hardware support for transferring platform control between the VMM and guest OSs, so when VMM intervention is required, handoffs are faster, more reliable and more secure	Helps to improve the fundamental flexibility and robustness of software-based virtualization solutions, including support for migrating VMs between servers.
Chipset: Intel® Virtualization Technology for Directed I/O (Intel® VT-d)	Enables the VMM to securely assign specific I/O devices to specific guest OSs	Eliminates the performance overhead by reducing the VMM involvement in managing I/O traffic.
Network: Intel® Virtualization Technology for Connectivity (Intel® VT-c) • Virtual Machine Device Queues (VMDq) • Virtual Machine Direct Connect (VMDc)	Extensive hardware assists into the I/O devices	Optimized network performance under virtualization

* Other names and brands may be claimed as the property of others

The Intel Xeon processor 5500 series enables that kind of maximized return by delivering a 2X leap [7] in virtualization performance vs. the previous generation Intel Xeon processor 5400 series, allowing virtualized clouds to squeeze even more capability out of their infrastructure.



4.2 Data Center Power Management

Intel has incorporated new power technologies into the Intel Xeon processor 5500 platform that address the challenges of managing power at the data center level.

Intel Intelligent Power Node Manager (Node Manager) is an out-of-band (OOB) power management policy engine embedded in Intel server chipsets that dynamically adjusts platform power to achieve configured performance-power ratio. Through the manipulation of ACPI-based processor performance states (P-states) and throttling states (T-states). Node Manager works with the BIOS and the OS to dynamically adjust platform power consumption. By setting user-defined platform energy policies in a coordinated manner, Node Manager can help data center operators increase server rack density while staying within defined power thresholds at an aggregate level.

Intel® Dynamic Power Technologies Data Center Manager (DCM) is a software middleware product that scales the capabilities of Node Manager to manage power at the data center level where hundreds of servers are simultaneously deployed. DCM is not an application, but a software middleware layer leveraged by management applications in communicating with Node Manager-capable platforms through IPMI or DCMI, and handling coordinated adjustments in individual platform configuration in order to achieve aggregate power management policies. DCM also provides support for power and thermal monitoring at varying levels of server aggregation and event-based alerting. With these building blocks, management software may be developed to manage power in sophisticated ways across an entire data center.

Intel Xeon 5500 platforms have also been specifically designed to operate at higher air temperatures – which are ideal for dense datacenters and cloud infrastructure – allowing operators to run their datacenters at warmer temperatures, lower cooling costs, and boost TCO.

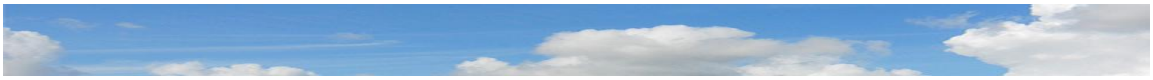
Intel's power management features helps data centers to keep the cost lower which in return gets passed on to cloud service providers and end users. So, as cloud application developer, it is critical that you develop applications that utilize power efficiently thus reducing the cost of deployment.

4.3 Industry Collaborations

Intel is very active with the cloud ecosystem to ensure Intel platforms meet current and future requirements.

At a product level, we work closely with a wide range of OEMs and large end users to provide the broadest range of optimized, dense systems for large data centers including single and dual socket servers, blades, rack optimized designs, and container-based compute modules. These OEMs are delivering innovative platform solutions to meet the unique needs in hosting Internet-scale clouds.

At an industry level, Intel is also working to help in resolving cloud adoption challenges that the industry is facing [11].



One of the challenges can be "where to start?" With the myriad of hardware options and variety of software solutions finding a starting point can be daunting. For example:

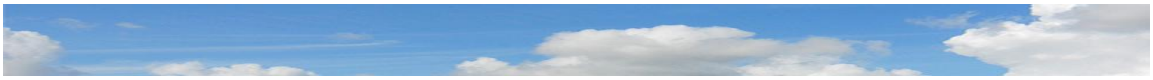
- What server configurations are optimal?
- How to structure the network?
- What is the optimal storage configuration?
- What is the cloud management stack that best suits my needs?

Intel recognizes this need and has formed the Intel® Cloud Builder program [12] to help in this 'getting started' phase. Developed with leading cloud independent software vendors (ISVs), the program provides a starting point to setup, deploy and manage a cloud infrastructure. The Intel® Cloud Builder program produces materials, including documented reference architectures and best known methods on how to build and operate a cloud by running the ISV software stack on a test bed hosted by Intel. The primary goal is to simplify the effort to deploy cloud-based solutions for service providers, hosters and enterprise customers looking to use cloud architectures. The program provides tools and best practices for cloud service providers to create a cloud environment based on a defined software and hardware stack. Industry leading ISVs have joined the Intel® Cloud Builder program.

Being in the early stages of adoption and development, there is not much clarity on how cloud will integrate with an organization's in-house databases, applications, work flows etc. Another challenge would be how multiple clouds will talk to each other (interoperability). As a customer, no one would like to get locked in with a single vendor. Industry standards help enable integration of cloud services into an organization, compose services from different providers into a flexible solution and faster deployment. Several industry standards bodies are working on cloud and data center standards including DCMI, DMTF, SNIA, NIST Open Grid Forum, CSA and others.

DMTF formed the Open Cloud Standards Incubator [13] to assess the impacts of cloud computing on management and virtualization standards and to make recommendations for extensions to better align with the requirements of cloud environments. Intel is co-chairing the Open Cloud Standards group and is actively participating in the definition of cloud interoperability standards. The DMTF's Open Cloud Standards incubator will develop a suite of DMTF informational specifications that deliver architectural semantics to unify the interoperable management of enterprise computing and cloud computing. This may include extensions to existing DMTF specifications including the Common Information Model (CIM), Open Virtualization Format (OVF), WBEM Protocols, member submissions and investigation of opportunities for collaboration with other industry standards bodies. The scope of this activity is focused on mainly cloud resource management aspects of Infrastructure as a Service (IaaS) with some work touching on Platform as a Service (PaaS) including SLAs, QoS, utilization, provisioning and accounting and billing.

Data Center Manageability Interface (DCMI) specification [14] explicitly addresses the unique requirements of server platform management within Internet Portal Data Centers (IPDC) and other High Density Data Centers where large numbers (into the tens of thousands) of servers are deployed. This specification seeks to help High Density Data Centers reduce the



cost and complexity of server platform management by defining a concise set of platform monitoring and control capabilities to deliver reliable, stable, interoperable, cross-vendor manageability targeted to the needs of this segment. The DCMI specifications (derived from IPMI from DMTF) define a uniform set of monitoring, control features and interfaces that target the common and fundamental hardware management needs of server systems that are used in large deployments within data centers, such as Internet Portal data centers. This includes capabilities such as secure power and reset control, temperature monitoring, event logging, and others.

5: Consider Intel Platforms When Building for Clouds

In this section, we consider the relationship between the ISV who develops and deploys applications for the cloud environment and Intel hardware platforms which are widely deployed within cloud computing infrastructures.

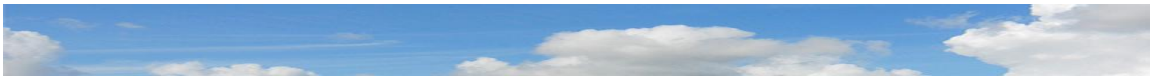
At first glance, it may not seem immediately obvious why an ISV needs to consider Intel platforms in their application development, testing, and deployment process. After all, many cloud computing environments work hard to abstract hosting services and hide the underlying details of server hardware platforms (and even operating systems).

Virtually all applications, however, need to be concerned about performance and scalability. Intel platforms be incorporated into the development and test cycle for several reasons. First, application developers need to obtain realistic measurements of application performance on hardware platforms representative of the deployment environment. This allows developers to understand the number of users, sessions, requests, etc. that a single application instance is capable of handling. It also provides a way to test the quality of service (QoS) under various load levels within a realistic environment.

Testing with Intel platforms also provides an opportunity to more accurately measure resource usage. Platform-specific information on CPU utilization, memory usage, disk I/O, network I/O, and so on can be used to provide VM sizing information to cloud service providers, communicate application requirements, and to estimate the costs of deployment. Resource usage information may also be tied to scaling trigger mechanisms that make automated adjustments to accommodate user demand.

Another aspect of performance testing is that of virtualization. As described earlier, Intel platforms provide a host of virtualization features that impact cloud application performance within virtualized cloud computing environments. Working directly with Intel platforms to test basic function under virtualized conditions, obtain performance numbers, and to identify virtualization-specific performance bottlenecks are all highly recommended.

It was mentioned previously that cloud hosting environments may range in their degree of abstraction. For those hosting environments that provide server infrastructure directly (expose "bare metal"), there may be opportunities to apply Intel platform-specific optimizations to improve performance and scalability. Intel compilers, threading libraries,



(e.g., IPP), performance analyzers, (e.g., VTune, Thread Checker), and other software products can prove invaluable in exploiting hardware capabilities to their fullest.

Finally, note that application performance considerations are not merely academic within cloud computing. Just as hardware platform capacity and efficiency considerations impact the cost of cloud computing hosting services, so too does application capacity and performance impact the cost passed on to ISV customers within the broader software business model. Applications that are highly optimized for virtualization and cloud computing performance may do much more work and take much less resources than non-optimized applications. The result is considerable savings within the utility-based fee structure of most hosting arrangements. This savings, in turn, may significantly impact both the cost of application hosting and the competitiveness of service pricing within a competitive market.

To summarize, to maximize the benefit of deploying an application in the clouds, we recommend that an ISV:

- Understand, architect, develop, and test using the capabilities of the Intel platform infrastructure to ensure cost-effective performance, and
- Specify Intel platforms to CSPs which will host your deployed applications to ensure cost-effective scaling.

6: Summary and Call to Action Checklist

This paper has considered cloud computing from the standpoint of ISVs developing software for cloud deployment. In particular, we have looked at what defines cloud computing and what motivates its adoption. We highlighted various attributes of "cloud ready" software and technical issues that must be addressed in the design process. We also considered deployment issues like monitoring, metering, and testing. Finally, we described what Intel is doing to improve the performance, power efficiency, and virtualization-readiness of server platforms on which cloud computing runs.

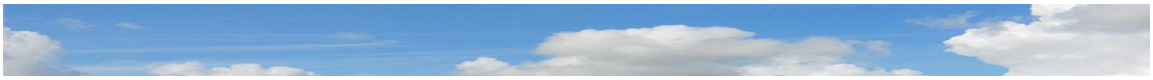
Here is a short summary list of tasks that ISVs need to consider then deploying their applications to the clouds:

Determine Goals, Objectives and Customer:

Be clear about what you are trying to accomplish and who you are trying to serve. Without a clear picture of your end-goals as an ISV here, your deployment effort will be troubled from the start.

Establish Architecture and Service Definitions:

Clearly define your application architecture and required services. Without a good foundational architecture and understanding of the services to be supported by your application, the likelihood that your application will support its intended purpose and do it well is rather small.



Develop Using Good Design Principles:

The more time and effort you put up front as an ISV developing an application, the fewer issues and problems you will have down the road. Taking the time and effort to utilize well-established design principles in architecting your application will permit it to most efficiently operate within cloud environment and thus scale well, etc.

Test in Virtualization and Mini-Cloud Environments:

ISVs all think they probably know what their application will do, but few truly get it right without actually running it. This is even truer when dealing with virtualized and cloud environments. Virtualized environments can expose issues with applications that have tied themselves to specific hardware platforms or other such things and moving that to the cloud just exasperates the problem. To side-step these problems, it is best to fully test your application in a virtualized environment to get a good understanding of how it operates when running within a virtual machine, and then test within a true mini-cloud environment to experience how your application responds to typical cloud demands (e.g. scaling up, scaling down, etc.).

Plan for Proper Service Deployment & Delivery:

Along with the actual development and testing of your application for the cloud, as an ISV you need to also not overlook the planning for your service deployment and delivery. This is an easy area to overlook but coincidentally it is just about the most important one! Be sure to consider all areas of deployment and delivery that could affect proper execution of your application or risk big problems in the cloud.

7: References

[1] Developing an Enterprise Cloud Computing Strategy, Intel Corp, http://download.intel.com/it/pdf/320566.pdf

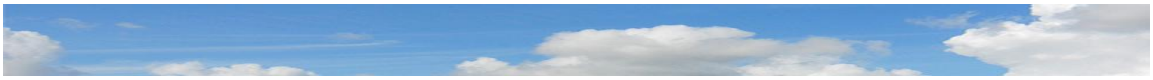
[2] Above the Clouds, A Berkley View of Cloud Computing, Technical Report No. UCB/EECS-2009-28, http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[3] Enterprise Acceptance of Software-as-a-Service: End-User Survey, Aug 2009, Infonetics Research.

[4] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and Gray-box Strategies for Virtual Machine Migration. In Proc. NSDI, April 2007. https://www.usenix.org/events/nsdi07/tech/full_papers/wood/wood.pdf

[5] C. Pautasso & E. Wilde, From SOA to REST: Designing and Implementing RESTful Services, WWW2009 Conference, http://dret.net/netdret/docs/soa-rest-www2009/

[6] Arnon Rotem-Gal-Oz. Fallacies of distributed computing explained. http://www.rgoarchitects.com/Files/fallacies.pdf, 2006.



[7] Why the Intel® Xeon® Processor 5500 Series is the Ideal Foundation for Cloud Computing , http://communities.intel.com/docs/DOC-4213 .

[8] Intel Xeon 5500 Series Software Industry Testimonials
http://www.intel.com/business/software/testimonials/xeon5500.htm

[9] Intel Virtualization Technology
http://www.intel.com/technology/virtualization/
http://download.intel.com/business/resources/briefs/xeon5500/xeon_5500_virtualization.pdf

[10] Baidu POC
http://software.intel.com/sites/datacentermanager/intel_node_manager_v2e.pdf

[11] Intel in cloud computing Wiki
http://communities.intel.com/docs/DOC-4230

[12] Intel(r) Cloud Builder Program
http://communities.intel.com/docs/DOC-4292

[13] DMTF Open Cloud Incubator
http://www.dmtf.org/about/cloud-incubator

[14] DCMI Specification
http://www.intel.com/technology/product/DCMI/

-- END --