Software Defined Infrastructure (SDI) is Intel’s Future for the Datacenter

SDI Leverages Virtualization and Consolidation to Increase Agility

Executive Summary

Moor Insights & Strategy believes that IT is on the cusp of a major datacenter architecture transition. This transition will be driven by 24x7 global business reach, dramatically increased use and depth of business intelligence (BI) and predictive analytics (Big Data), and pushing sensors and intelligence into our physical world in the form of the “Internet of Things” (from datacenters to wearable consumer electronics). It is impossible to predict exact technology directions even in a three to five year timeframe, but the industry is starting to form a good, high-level framework for the future of IT operations.

Software defined datacenters (SDDC) are the next generation operational target for IT. SDDC is the flexible, agile infrastructure framework that will promote IT to a full business partner capable of creating value as well as increasing operational efficiencies. Though the term SDDC was coined by VMware, we believe the key concepts of SDDC are broader. Specialized hardware will have a role in future datacenter architecture, but infrastructure software will mediate workload access flexibly to scalable hardware.

Intel’s concept of software defined infrastructure (SDI) extends our definition of SDDC. It is a reevaluation of system architecture driven by the requirements of business flow, workloads, and specific applications—not by a menu of hardware available to purchase at the moment. SDI will transform mainstream enterprise datacenters. SDI has the potential to displace classic mainframes and highly available transaction processing systems by the end of this decade.

The first step on the SDDC path is to understand the basics of “software defined” architectures, how orchestration and composition can be overlaid on current datacenter infrastructure, and how they are likely to affect datacenter architecture in the coming quarters. The second step is to track SDI progress over the next two years, including Intel’s directions and investments in system software and rack scale architecture.

We recommend understanding and then evaluating Intel’s first generation SDI orchestration and composition software suite. It seems like a good means to increase enterprise datacenter flexibility and agility while leveraging current hardware infrastructure investments and lifecycle management practices.

State of Today’s Converged Enterprise Datacenter

The first big IT architecture transition was from mainframes and minicomputers to microcomputers (RISC and x86). The shift in systems architecture, software development, and IT skills through that transition was driven by the widespread availability of consumer personal computers. The current transition is similarly driven by the move from microprocessor-based microcomputers and personal computing to cloud-enabled mobility and the widespread democratization of cloud-based app development targeting mobile, wearable, and ambient (built into our environment) computing platforms.

At a more technical level, the IT industry began with general-purpose business accounting architectures originally designed to automate calculating ballistics trajectories and tabulating census data (mainframes and minicomputers). Then it moved to systems purpose-configured for specific applications (RISC and then x86 servers). And then it moved back again to general-purpose pools of virtual machines (VMs, x86 predominantly with VMware). Through this deep history, the prevailing datacenter investment philosophy has been that hardware is the expensive, scarce resource. The prevailing philosophy assumes that it is much more cost effective to employ software developers to stitch together hardware systems rather than it is to design flexible hardware systems that minimize developer effort.

Today the industry has reached a high state of maturity for this model: the virtualized, converged enterprise datacenter. The components of this mature enterprise IT technology base are:

- **X86 instruction set architecture (ISA)**. Intel’s processors dominate the market at roughly 95% share of server shipments. AMD’s compatibility with x86 software development tools gives software developers a unified compilation target that spans almost all of the servers shipped into the market.
- **Ethernet switching**. Ethernet has evolved to meet increasing datacenter demands and has now displaced every other connectivity standard for most enterprise datacenter deployments. Cisco is the clear market leader with about two-thirds of market revenue; HP is second at under half of Cisco’s share.
- **Hypervisor**. VMware is currently the market leader with about half of market revenue, and Microsoft’s Hyper-V is in second place and gaining. Two-thirds of enterprise datacenters are virtualized today; we are past the tipping point and starting to convert the long tail. Most virtualized enterprise datacenters are virtualized for workload consolidation, but note that for many scale-out datacenters, virtualization is increasingly used to manage application lifecycles, diagnose health and status of applications, *etc*. instead of consolidation.
- **NAS and SAN**. EMC is the leader in external disk storage systems but has a less commanding lead than leaders in the above categories enjoy. NetApp has half EMC’s share. IBM, Hitachi, HP, and others vie for a slice of a larger “other” category than the above technologies. In the total disk storage market, Dell places third behind EMC and NetApp.

- **Microsoft Windows Server and Red Hat Enterprise Linux (RHEL)**. Not too long ago, the OS was the point of contact between applications and hardware. Now that role is filled by a hypervisor. While the OS still matters, applications vendors often specify which OS versions they will run with in a VM.

Datacenter servers have standardized on a set of dominant design attributes:

- **2P**. Two-socket symmetric multiprocessing (cache coherent shared memory space) are the market workhorses.
- **Thin chassis**. 2P/1U and 2U "pizza boxes" now dominate enterprise server shipments; 2P blades are in third place.
- **Converged and "engineered" systems**. Because all of the above have been so homogenized and standardized, vendors are reacting by integrating their own hardware and software infrastructure bundles to address specific applications like SAP and Exchange.

There is now very little practical differentiation between converged server products. Datacenter buyers state that staying with the same brand is three times (3x) more important than choosing the server platform that best fits their application (InformationWeek Data Center Decision Time survey, Sep 2013). The success of VCE Vblocks and Cisco UCS is in large part due to a now-level playing field where brand matters more than technology.

Attention is now focused on the cost of creating, provisioning, managing, moving, and retiring VMs. And that's where we start to feel the impact of cloud technologies, because now we're talking about "orchestration". And the industry is about to change its mind about whether hardware or software is more expensive.

What Do We Mean by "Software Defined"?

"Application ecosystems" emerged in the consumer market with Apple's iOS with iTunes App Store and Google's Android with Play Store. It's the idea that apps are "stickier" than devices. People stay with their apps for long periods of time while they replace the underlying hardware at a fast pace. App developers participating in those ecosystems code to an abstracted hardware model, and hardware vendors create new platforms that run those apps faster, cheaper, or with longer battery life.

Developers and end customers can only invest their time and resources in a couple of dominant app ecosystems in each technology market. It's not that each app developer would find it hard to port their high value apps to another app ecosystem (that is an expense we call "opportunity cost"). It's that once an app ecosystem gains critical mass it becomes difficult to entice enough developers to port or move to another ecosystem so that it gains critical mass. Once developers move away from an ecosystem (Blackberry, for instance), it is nearly impossible to win them back.

SDDCs borrow this notion that applications come first. The idea of "software defined" datacenter attributes emerged from hyperscale service operators who separated physical datacenter hardware from the responsibilities of delivering a robust service. In their quest for reliable yet low cost components, these vendors created a new model for resiliency: physical operations are continuous and uninterrupted in the face of all possible sources of downtime **in addition to** reconfiguration of policies and control.

Web giants like Amazon Web Services (AWS), who had already implemented SDDC, took an early lead in the merchant cloud service provider (CSP) market. Google and Microsoft, also early leaders in SDDC, have been trying to grow competing CSP services. But while all three are growing revenue, AWS continues to grow its share of market. The challenge for new entrants in the CSP market is how to compete with these web giants who have raised the bar for hiring internal software developer resources to all but unobtainable heights.

That's where OpenStack comes into play. Its goal is simple; mashing-up their website yields: *an open source cloud computing platform for public and private clouds that is simple to implement, massively scalable, and feature rich*.

OpenStack is an SDDC framework that will enable folks who own physical datacenters—folks who have paid for buildings, physical servers, storage, networking, and utility contracts, such as hosters or enterprise IT datacenter operators—to stand up their own private cloud. OpenStack was founded by Rackspace and NASA precisely for that purpose. It also provides the core software basis to compete with AWS and other CSPs if they so wish (and without hiring droves of software developers), although it is not a complete solution to compete as a CSP.

The number of IT vendors supporting OpenStack and the weight of their resource investments has given OpenStack the momentum to become a vendor-neutral *de facto* applications programming interface (API) for OpenStack private clouds to burst to OpenStack hosters or public clouds.

The components of SDDC (networking, storage, security, *etc.*) each abstract their architecture into two "planes": control and operations.

- The **control plane** manages operational and business policies, whether they are continuous, discrete, or one-time. Higher level system control functions are referred to as "orchestration" as they span a wide range of control functions and may bridge the control planes of different software defined domains. The control plane tells the operations plane how to behave.
- The **operations plane** delivers a continuous service in the face of ongoing control plane instructions of any type. Operations planes are specific to each software defined domain. In the absence of new control plane input, the operations plane continues to function based on its most recent set of instructions. In software defined networking (SDN) this is a "data plane".

This separation of policy (control) and function (operation) follows "policy-based automation" which separates business and operational policies from the mechanics of performing those functions. This SDDC architecture (control plane and data plane architecture) was originally designed for the largest hyperscale hardware deployments, characterized by massive replication of commodity hardware, and in Google's case, self-built from independently sourced components.

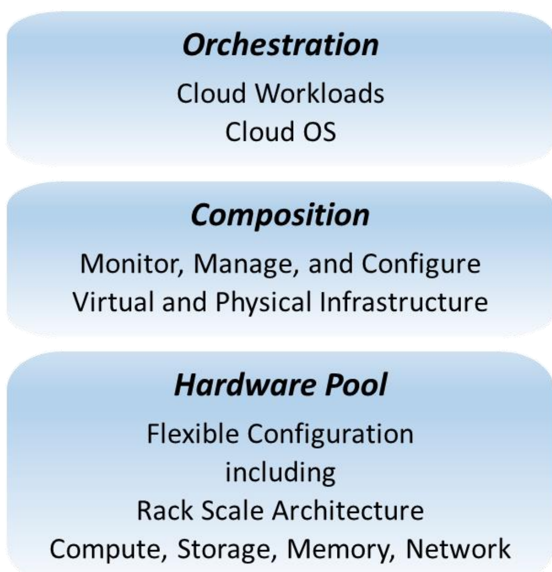
Where SDI Fits

Intel cites three catalysts for developing SDI: software defined networking (SDN), network function virtualization (NFV), and software defined storage (SDS). These are all examples of policy based automation. Data communications and SDDC vendors and customers have deployed and tested several product models for separating control and operations planes; their consensus is that the architectural template works well.

There are three layers to Intel's SDI model:

- **Orchestration**: a policy engine that allows higher level frameworks to manage composition dynamically without interrupting ongoing operations—a control plane.
- **Composition**: a low-level layer of system software that continually and automatically manages the pool of hardware resources. The term "data plane" has been used by the networking community for many years, but it does not translate well to other datacenter domains; we believe "composition" more accurately spans domains.
- **Hardware Pool**: an abstracted pool of modular hardware resources.

Figure 1: Intel's 3-Layer Software Defined Infrastructure Model



Similarly, the DMTF Open Software Defined Datacenter (OSDDC) Incubator's work-in-progress definition of SDDC lists these features:

a) *A pool of compute, network, storage and other resources*
b) *Discovery of resource capabilities*
c) *Automated provisioning of logical resources based on workload requirements*
d) *Measurement and management and of resources consumed*
e) *Policy-driven orchestration of resources to meet service level obligations of the workloads*

Intel's goal is to give datacenter customers greater flexibility and efficiency by enabling them to dynamically "right size" application resource allocation as needed (custom resource allocation per application image) and thus enable provisioning of a new service in minutes. This represents a vast improvement in service time and quality of service (QoS) and also big improvement in reducing human effort and electrical power consumption to "stand up" those new services.

The most interesting attribute of SDI from an enterprise IT viewpoint is that it does not specify exotic, new hardware architectures or even updates to existing hardware capabilities. In fact, the high-level architectural definition and software design work being done is remarkably hardware and OS/hypervisor neutral.

Flipping the Bit: Applications Come First

The biggest change in perspective for SDDC architecture is that an application and its service level needs come first. System resources are configured to supply optimum resources to an application so that expenses are kept under control while service guarantees are met.

This application-first perspective contrasts the current enterprise IT strategy of over-provisioning a pool of VMs for worst case application demands (both off-the-shelf and in-house). In some ways, the current strategy is reminiscent of buying servers twenty years ago when each server ran one application and was configured for the worst case peak load anticipated for that application; similarly, enterprises now provision a pool of hardware for the most challenging applications they expect to run in any VM (buying the flexibility to run any application in any VM). Because of the variability in resource demands among those business applications, the average hardware server node resources dedicated to a VM are overbuilt for most of the applications running in those VMs. Although enterprises have consolidated those individual, lightly utilized servers into a pool of virtual servers with much higher hardware utilization, there is still a lot of room for improvement. Over-provisioning incurs wasted capital and operational expenses.

Web giants provision their applications at such scale that using identically configured VMs to run all of their applications is inefficient. They compete to deliver increasingly better QoS at ever lower prices, so they can't afford to overprovision. They have developed a strategy of creating differently configured hardware archetypes to cover broad classes of applications. They buy each archetype at high enough volume that they don't lose their bulk hardware purchasing power while they gain efficiencies in capital and operational spending. This attitude toward hardware spending is one of the root drivers for SDDC architecture.

The eventual goal for SDDC and SDI is the ability to configure a logical VM from fine-grained system resources. SDI is dependent on hardware modularity and a software framework to compose fine-grained hardware modules into a VM that is tuned for a specific application's runtime needs.

In the short-term, SDI has strong potential for enterprise IT in creating hardware server node profiles—useful for different generations of server hardware purchases over the years, so that applications can be better matched with VMs running on a range of legacy server configurations. This might enable IT to easily reuse and on-the-fly provision older hardware for less demanding applications, automatically.

Orchestration and Service Assurance

Looking at SDI from the top down—from an application perspective—the orchestration layer understands each application's QoS requirements and creates both policy controls and requests to create, manage, or shut down individually configured VMs. Service assurance is then the ability to set policies at an appropriate level of detail in the control plane and in real-time enabling dynamic management of hardware resources in the operations plane.

Orchestration policies can control application availability, security, QoS, performance, and consumable resources (power, airflow, network bandwidth, *etc*.). These policies are used to configure, provision, operate, and adjust then decompose an application's system resources for the duration of its service life.

At scale, an important feature of service assurance is to monitor hardware and software infrastructure actively and intelligently. To ensure that QoS requirements are continually met, hardware and software health and status (including VM utilization) must be continually monitored. This continual monitoring requires physical and logical sensors plus metadata and real-time data storage structures that maintain the capabilities, current status, and history of every component in a SDDC.

Continual monitoring of this stream of status data, coupled with automatic policy enforcement, enables the orchestration layer to do several things. It can solve service issues faster and continually optimize the hardware resources needed for each application to keep resource utilization at an acceptable level. It also can manage dynamic addition of new applications into a pool of resources while maintaining the service levels of running applications.

There is also an aspect of Big Data and predictive modeling to orchestration. Continually monitoring application performance and QoS under a variety of operational conditions provides the data that an orchestration system needs to anticipate service problems in the future and to boost resource utilization efficiencies over time.

Orchestration is the least-developed of the major SDI components. OpenStack's "Heat" component is one example of an orchestrator. Many system and software vendors are designing orchestrators for OpenStack, such as HP's Cloud Agents, and are feeding the source code back into the open source community. We expect rapid development of orchestrator feature sets over the next few years.

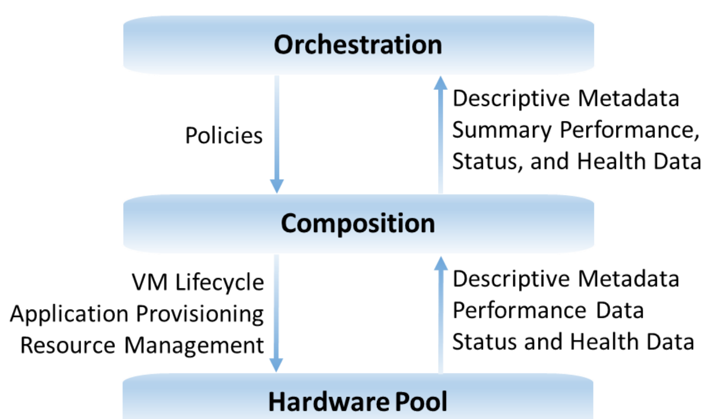
Composition vs. Interoperation

Today's IT infrastructure is designed for interoperation. Individual server nodes, switches, and storage subsystems are not dynamically reconfigurable, if they are reconfigurable at all. Migrating an application instance from a VM on one server node to a VM on another is a matter of matching the hardware capabilities of the source and target nodes to a high degree of conformance; migration then requires rerouting all of the network and storage plumbing so that the application image believes it has exactly the same system resources it started with after it is moved to its target VM. SDN is helping out quite a bit in managing the plumbing parts of VM migration, but server node configurations are a different matter. VM migration is a matter of finding as exact a copy of the originating server node as possible.

Composition is different. It is a broader implementation of provisioning management. Composition uses orchestration policies to automatically create, configure, provision, operate, adjust, and then decompose logical VMs (dynamic federations of compute, storage and network resources assembled to run an application instance).

The composition layer acts on metadata and performance data it receives from the components of a logical VM, but only to implement the operational policies set by the orchestration layer. The composition layer also passes metadata and performance data to the orchestration layer for real-time analysis and decisions about when and how to change operational policies for the composition layer.

Figure 2: SDI High-Level Summary



This means that SDI compute, network, and storage components must report enough information to both the composition layer and the orchestration layer to enable operational policy enforcement, control, and predictive analytics. Given that flow of descriptive, sensed data, composition enables a fine-grain logical VM orchestration that is not possible with older generations of IT infrastructure.

Composition enables the logical disaggregation of compute, network, and storage resources. Intel's vision is that they will be enumerated separately and then can be assembled into smaller logical VMs if needed. A rack full of servers can then be viewed as a pool of resources and can be managed at a lower level than in a current virtualized environment.

Intel started down this path at their 2013 Developer Forum. They announced that they were working with Amazon to surface detailed information about the Intel hardware behind AWS instances. Intel has since expanded their Cloud Technology Program with 15 additional CSP partners (participating CSP partners are recognized with a "Powered by Intel Cloud Technology" badge). We can expect to see more and deeper configuration and telemetry information exposed over time to Intel's software and services partners.

Disaggregated Hardware and Scale

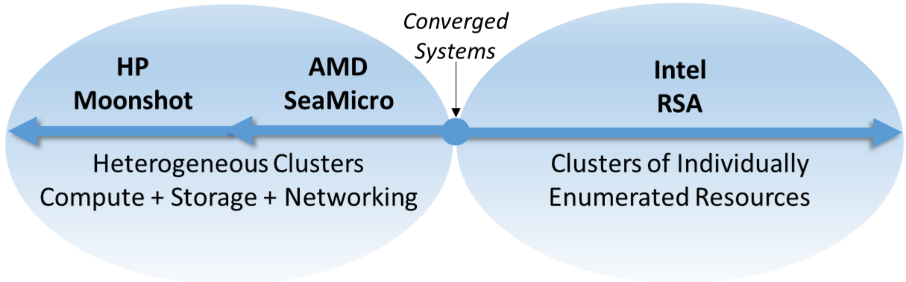
Treating datacenter infrastructure as a pool of abstracted resources is a difficult challenge, but it is central to SDDC and SDI. Using converged systems as a metaphoric anchor point, if we look in one direction we see AMD's SeaMicro chassis containing a sea of small, identical server nodes connected by their Freedom Fabric 3D torus mesh fabric topology. AMD has deployed several different processor cards for the SeaMicro chassis, including cards with Intel processors, but cards are deployed homogeneously with each chassis.

Looking a bit farther in that direction, past SeaMicro, we see HP's Moonshot. Its flexibly interconnected chassis offers three different interconnect fabrics and topologies. Moonshot has potential to blend different server and storage "cartridges" to create, in theory, more optimal hardware configurations for different applications.

We'll call the directional approach that AMD and HP have taken "fabric-based". Both AMD and HP are following a path of smaller, less expensive (cost and resource consumption), integrated server nodes that still containing compute, memory, and network or fabric I/O capability. They are modular, but their hardware resource abstraction is tied to a blend of hardware capabilities.

Going back to our converged anchor point, if we look in the opposite direction we'll see Intel's Rack Scale Architecture (RSA). Intel is aiming to completely separate and abstract compute, storage (eventually including main memory) and networking as independent hardware resources.

Figure 3: Directional Approaches to Server Hardware Resources



Looking up the stack at the composition and orchestration layers sitting above our pool of hardware resources, we can see the impact of choosing directions. On one side, the fabric-based system is managing potentially heterogeneous clusters of mixed compute, storage, and networking. On the other side is completely separated clusters of individually enumerated resources. The cluster approach is closer to today's architectures and doesn't require radically new software design and orchestration techniques. By inference, Intel's RSA approach eventually will require radical system software redesign, but in the short-term RSA will look quite a bit like the AMD and HP architectures it refers to as "microservers".

How Rack Scale Architecture (RSA) Applies to SDI

We have already written about Intel's RSA, and so we defer deep technical details to our previous paper. We will focus here on RSA as applied to SDI.

RSA starts with the assumption that a full rack full of server, storage, and networking is the basic unit of datacenter scaling. This is not for small business. To the contrary, we believe that to consider deploying RSA a service provider must have deployed a large-scale service already and now seeks to reduce costs through datacenter customization and application specific optimization. "Rack scale" is aimed at thousands of racks each containing thousands of hardware resources. While fabric-based architectures ostensibly are aimed at the same rack scale, they are housed in smaller chassis increments.

Within an RSA rack, the basic unit of scaling is a "tray". A tray contains a single type of component resource; it is all processors, all storage, all memory, or all network aggregation. The primary goals of RSA are software composable service agility (discussed above) plus easier hardware serviceability and upgradability. Intel and Facebook believe that these trays will decouple component replacement and upgrade cycles so that one component type will not be held hostage to anther component's more frequent upgrade or service cycles.

The only reason to upgrade the tray itself is to upgrade the physical interface components connecting the tray's logical signaling, power delivery, mechanical stability, and thermal management infrastructure.

All of the components in a tray, as well as the tray and rack infrastructure (passive or active), should be instrumented to sense power consumption and thermal behavior, utilization of available performance, security status, location within a rack and the larger datacenter, *etc*. These components should also report metadata enumerating their functions and the expected performance characteristics for each of those functions. The more metadata and real-time performance data exposed, the more accurately the composition layer can enforce operational policies. As we mentioned above, Intel has already started down this path and promises more information during the year.

The concept of "hot swappable" components now makes a lot more sense for RSA than previous or competing architectures. The composition layer should be able to detect issues with a component easily and proactively take that component offline for replacement or upgrade while the rest of the components in its tray continue to operate normally.

For the first generation of RSA rack architecture, processor cards will host dedicated local main memory with a traditional DIMM socket (most likely a mobile SODIMM slot for a smaller physical footprint). This is a short step from current converged architectures and should have a lot in common with fabric-based architectures in terms of preserving current server system software and application design best practices.

The biggest challenge for RSA is also its strongest long-term competitive advantage: a restructuring of datacenter storage hierarchies, from local system memory to cold storage. Disaggregating memory from processors is a huge systems design challenge, and it will require rethinking the relationship of operating systems and hypervisors to memory and storage components.

Call to Action

The next three to five years are a pivotal period in the IT industry's transition to SDDC and new SDI architectures. What happens in the next few years will determine the dominant architectural directions for the next two or more decades.

Intel hasn't yet formally launched SDI products with their industry partners and hardware customers. The key question when they do launch will be, "Can we overlay SDI technologies or products on our current enterprise datacenter infrastructure, or are we talking about new greenfield datacenter build-out only?"

We recommend understanding and then evaluating Intel's first generation SDI orchestration and composition software suite. That suite should run across a very wide range of Intel-based datacenter infrastructure—from Atom-based "microservers" to the biggest, most recent flagship Xeon-based machines.

Key points are:

- Choose open, industry standards-based SDI software suppliers that match your risk profile of platform support and utility and that run on commercially available hardware.
- Evaluate moving from dedicated hardware appliances to virtual appliances. Networking and storage will benefit the most from this shift in the near-term.

For enterprise IT customers who are planning to stay with Intel's Xeon brand and are not planning on ripping and replacing entire datacenters: SDI orchestration and composition seems like a good means to increase enterprise datacenter flexibility and agility while leveraging current hardware infrastructure investments and lifecycle management practices.

For datacenter operators planning incremental or entirely new datacenter build-out: Keep up-to-date with RSA technologies, and evaluate products as they enter the market. On the software side, given Intel's work in the OpenStack community, we expect that Intel will contribute key pieces of their orchestration and composition layers into OpenStack and possibly other cloud stack forums. We also believe Intel will eventually do the same for Intel-branded SoC metadata and real-time reporting mechanisms. Also, Intel is likely to partner with Red Hat and other cloud software stacks to ensure Intel's SDI layers and processor metadata and performance data interfaces are integrated into the latest and most widely used open source enterprise-class cloud stack distributions.

Have we changed our opinion on the potential for alternate ISAs in SaaS and public cloud deployments? Absolutely not! However, it is impossible for us to gloss over fundamental business reality. For enterprise datacenter markets, x86 is the only ISA that is likely to be a factor in the market for many more years. Intel's longer-term challenge will be to enable SDI to span high-end Xeon performance down to Atom and possibly even Knights Landing feature sets and performance levels. This is not really a technical challenge as much as it is a business challenge.

Important Information About This Paper

Authors
Paul Teich, CTO, Senior Analyst at Moor Insights & Strategy.

Editors
Scott McCutcheon, Acting Director, Research, at Moor Insights & Strategy.
Patrick Moorhead, President & Principal Analyst at Moor Insights & Strategy.

Inquiries
Please contact us here if you would like to discuss this report and Moor Insights & Strategy will promptly respond.

Citations
This note or paper can be cited by accredited press and analysts, but must be cited in-context, displaying author's name, author's title and "Moor Insights & Strategy". Non-press and non-analysts must receive prior written permission by Moor Insights & Strategy for any citations.

Licensing

Disclosures
Moor Insights & Strategy provides research, analysis, advising, and consulting to many high-tech companies mentioned in this paper including Intel, who commissioned this paper. No employees at the firm hold any equity positions with any companies cited in this document.

DISCLAIMER
The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors. Moor Insights & Strategy disclaims all warranties as to the accuracy, completeness or adequacy of such information and shall have no liability for errors, omissions or inadequacies in such information. This document consists of the opinions of Moor Insights & Strategy and should not be construed as statements of fact. The opinions expressed herein are subject to change without notice.

Moor Insights & Strategy provides forecasts and forward-looking statements as directional indicators and not as precise predictions of future events. While our forecasts and forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could cause actual results to differ materially. You are cautioned not to place undue reliance on these forecasts and forward-looking statements, which reflect our opinions only as of the date of publication for this document. Please keep in mind that we are not obligating ourselves to revise or publicly release the results of any revision to these forecasts and forward-looking statements in light of new information or future events.