TidalScale

March 2018

Software-Defined Servers

A guide to adding the missing piece in the Software-Defined Data Center puzzle

In this issue

Gartner®

2018: The Year of the Software-Defined Server

Letter from Gary Smerdon, CEO, TidalScale

With data volumes growing at an astounding **62 percent CAGR**, it's easy to see why some organizations are struggling to keep up. They're trying to process and analyze bigger and more complicated problems, which increasingly stresses their computing resources.

At a time when in-memory databases and analytics are pushing data centers beyond their capability, when DevOps initiatives require on-demand access to IT resources, when IoT data overwhelms traditional servers, one part of the solution may be the **Software-Defined Server**. A Software-Defined Server combines all the resources of multiple commodity servers into one virtual server. Applications and operating systems need no modification, and they think they're looking at an actual single hardware system. In a data center, multiples of these virtual servers can be deployed as needed.

In 2017, Software-Defined Servers landed squarely on the radar screens of media and analysts. TidalScale, which pioneered the Software-Defined Server, was named a **Gartner Cool Vendor for 2017**, and a **leading innovator** by other firms. *Red Herring* placed us in its **Top 100 Global** list of innovators. eWeek's **coverage** of TidalScale described our Software-Defined Servers as "the biggest advance in servers since VMware's virtualization."

This year, more organizations will find that the solution to the limits of hardware isn't hardware at all. The Gartner research spotlighted in this newsletter looks closely at the next evolution in infrastructure and data center automation architecture including business value, best use cases and risks.

That next evolution is the Software-Defined Data Center (SDDC). According to the report "an SDDC is becoming critical to achieving the business agility, speed and time to market associated with a digital business. Orchestrating infrastructure such as that required for IoT, DevOps and cloud services is accomplished through software-defined components, enabling greater end-to-end automation, mobility and integration through the use of APIs, and policy orchestration". Also featured is a planning guide that's a must read for anyone thinking of implementing a software-defined data center.

We hope you'll find this research and planning guide helpful as you consider what a software-defined future could mean for you.

The Growing Need for Software-Defined Resources

The pace of technological change is accelerating, creating a future that is less predictable and more difficult to navigate. Organizations that once relied on a strategy of overprovisioning – buying more hardware capacity than they need today in hopes that it will accommodate future workloads – find that approach is increasingly inflexible, unpredictable and expensive. The demands on data centers are evolving too rapidly.

New developments are making old approaches obsolete. Data sets are growing, thanks in part to the increased use of sensors, social media and other sources that proliferate data. Businesses are increasingly relying on in-memory databases and analytics applications to make sense of this data tsunami.

In some cases, even new technologies are failing to fully serve these escalating needs. Next-generation rack-scale hardware designs can help businesses become more flexible and agile, but they lack an easy way to aggregate the most critical resource in the rack – compute and memory – in a logical layer. At the same time, traditional scale-up hardware is almost exclusively designed to deliver more compute capability, when what modern applications need most is a balance of compute *and* memory. This leaves IT having to make big bets on large systems whose price/performance ratio declines as configurations grow larger. What they want is flexibility, but traditional (and expensive) large systems don't provide it. It's a high price to pay for giving up the ability to determine your own future.

Three key needs have most organizations trying to decide on their most promising technology options.

Businesses need results faster. In today's marketplace, deriving useful insight quickly isn't just a good idea. It's a business imperative.

The problem is that data centers aren't always equipped to deliver those insights, at least not quickly enough. Most data centers are comprised of fixed resources and hard-to-scale assets that make it tough to quickly meet fluctuating requirements. Organizations don't have time to wait for the next big system to be acquired or for applications to be rewritten so they can run across distributed servers.

They also need to lower costs. The traditional scale-up model of buying new hardware just doesn't work for many organizations. For one thing, a lot of them can't afford it, due to the non-linearity of server costs as larger servers replace smaller ones. And even if you can, there's the often-futile exercise of trying to anticipate your needs three years from now so you can overprovision today in anticipation of having enough iron to throw at the problems you think you'll face in 2021. But who can possibly predict what they'll face in a year, let alone three?

It doesn't help that traditional software licensing is usually based on core count. That causes organizations running memory-intensive applications to essentially overpay for cores they're not even using just to access the amount of memory they need for their problem. Even if you buy a large scale-up server, hardware vendors typically tie large memory with high processor core counts, along with their associated low single stream performance – a combination that is not always optimal. So, you end up overpaying twice.

IT administrators and DevOps are seeking agility through flexibility. To further derive the greatest ROI from existing investments and to drive down operating costs, they are looking at their data centers differently. They are packaging applications, microservices and data onto containers as a lightweight alternative to traditional virtualization. They are looking to cloud infrastructure-as-a-service (IaaS) providers to access on-demand resources they don't possess in house.

Some are doing all these things. And others are going a step further by establishing a software-defined data center (SDDC).

Source: TidalScale

On-Demand Everything: The Software-Defined Data Center

In their April 21, 2017 report, *"Should Your Enterprise Deploy a Software-Defined Data Center?"* Gartner analysts Dave Russell and Donna Scott note that SDDCs are "becoming critical to achieving the business agility, speed and time to market associated with a digital business."

The notion of an SDDC focuses on flexibility and optimization of existing resources. IT assets become virtual, and this in turn makes them more readily available to changing workloads. An API-enabled software layer can orchestrate how those resources are used, and automate the way they are deployed, even on demand.

Source: TidalScale

Why Software-Defined Servers are the Missing Piece of the SDDC

Most data center managers – and even many end users – are familiar with Software-Defined Networking and Software-Defined Storage. These technology paradigm shifts elegantly deliver the flexibility and cost-effectiveness that they were designed for.

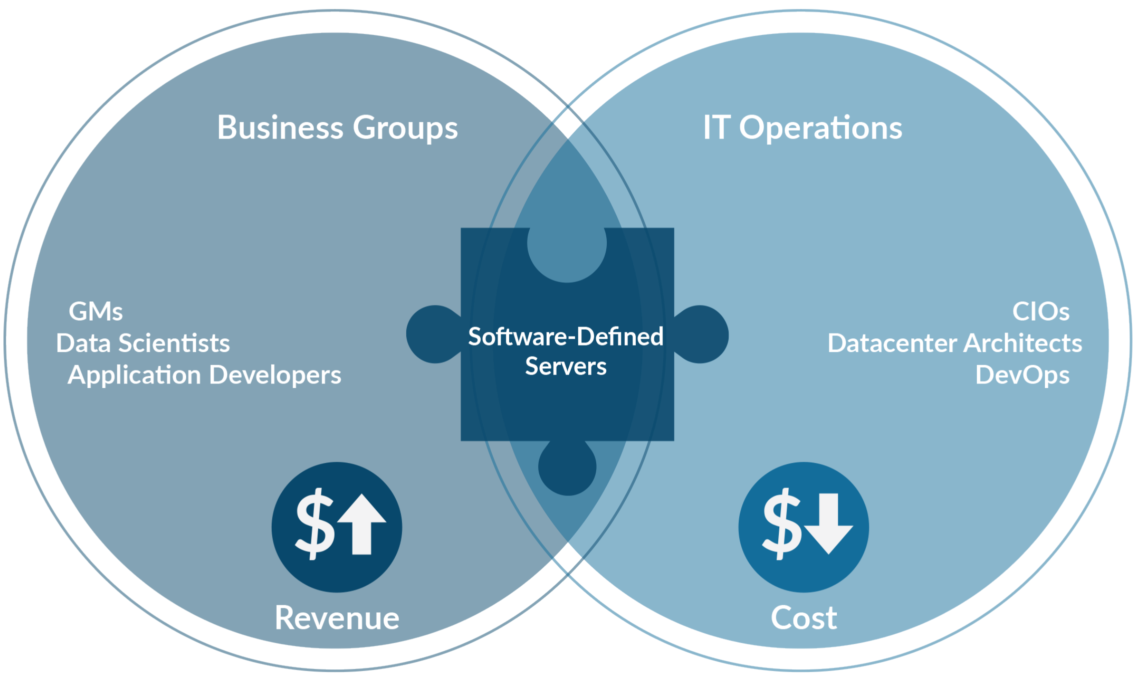
It's astounding, then, that servers have been left behind. They're the one part that has been conspicuously missing from the software-defined puzzle. And as missing pieces go, it's a big one. Servers represent the heart of the data center – the very engine around which all other resources and systems are built.

Servers Left Out

Yet traditional servers today remain a fixed and inflexible asset in data centers. Virtualization has allowed organizations to get more use from a single server, but many applications simply aren't appropriate to run over clusters. And if they could be made to run over clusters, they'd have to be rewritten. Despite the compelling need, servers have been left out of the Software-Defined Data Center.

Software-Defined Servers represent that missing piece. As its name might suggest, a Software-Defined Server uses software to combine multiple physical computers into a single virtual server. The virtual server spans from one server to many physical servers, aggregating all the resources of the combined physical servers, including memory, CPUs, networking and storage. And all those resources can then be made available as part of one large, fast, flexible virtual server.

Software-Defined Servers: A Win for Business and IT



What do Software-Defined Servers offer users of databases and in-memory apps?

On-demand access to all the memory they need. *What this means:* Results sooner and at a lower cost than traditional scaling.

How do DevOps and SDDC environments benefit?

More flexible use of IT assets & reduced cost of ownership. And IaaS providers can offer subscribers more server options.

Source: TidalScale

Software-Defined Servers deliver results faster. A big advantage to using them is that they enable you to achieve results more quickly, particularly when using applications and in-memory databases. For instance, say you have 40 different servers across your data center or organization, and each is equipped with 512GB of RAM. They can be combined into a single Software-Defined Server that is indistinguishable from an actual physical server equipped with 20TB of memory. As far as the user, the application, the operating system and the data are concerned, it's all one computer.

Suddenly the large-scale analyses that previously slowed your progress will no longer overwhelm the computer; rather, you size the computer to the problem rather than the other way around. With the right amount of resources available, insights surface sooner.

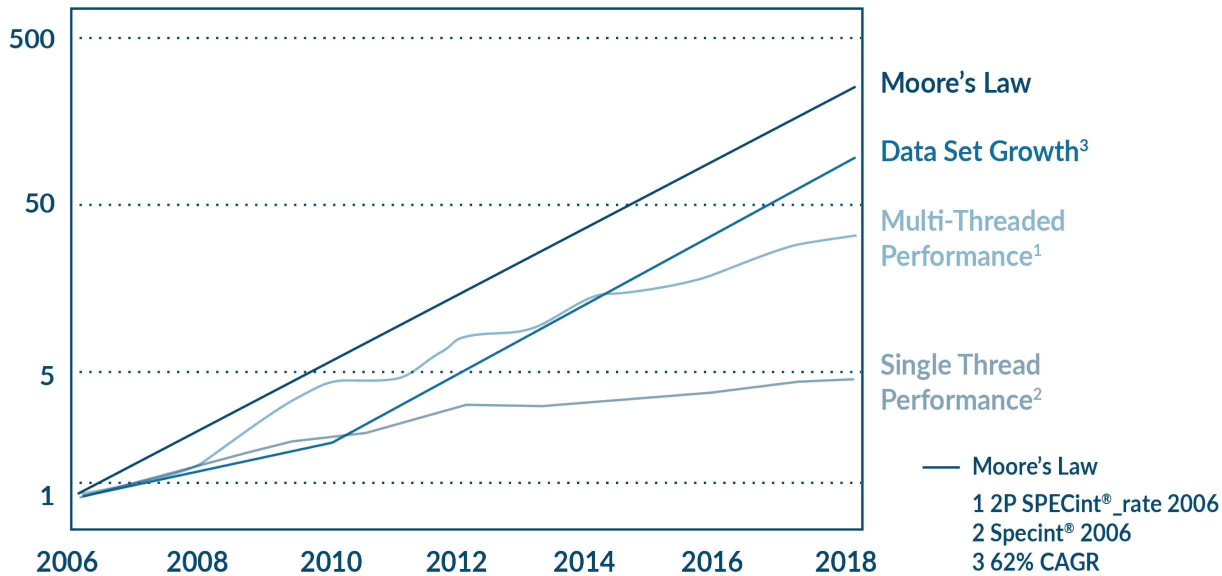
They lower costs. Software-Defined Servers also help drive down costs by extracting greater utility and longer usable life from existing IT resources, including so-called "sweet spot" servers – industry-standard commodity systems designed to handle simple workloads, but that are ill-suited for more taxing problems.

They enable greater data center flexibility. You can combine your population of physical servers in any configuration that gives you the cores, memory and I/O you need to solve the problem at hand. Given the example above, you can create one system with 12TB of memory and four others, each with 2TB. Mathematically speaking, you have thousands of options at your disposal, delivering the kind of operational flexibility and cost savings data center administrators have long been searching for but failing to find.

7 Technology Realities Driving Software-Defined Servers

If Big Data and unpredictable workloads overwhelming traditional servers weren't enough, computing and networking technologies themselves are proving that the time has come for Software-Defined Servers. Moore's Law, which for more than 50 years reliably promised that microprocessor performance would double every 18 months, has been rendered **effectively dead**, with parallelism and other innovations working overtime to try to keep pace with the intensifying data tsunami. In the face of this surging need, there are seven industry trends that constrain and drive the need for agile infrastructure enabled by Software-Defined Servers.

The Rate of CPU Performance Increases is Slowing

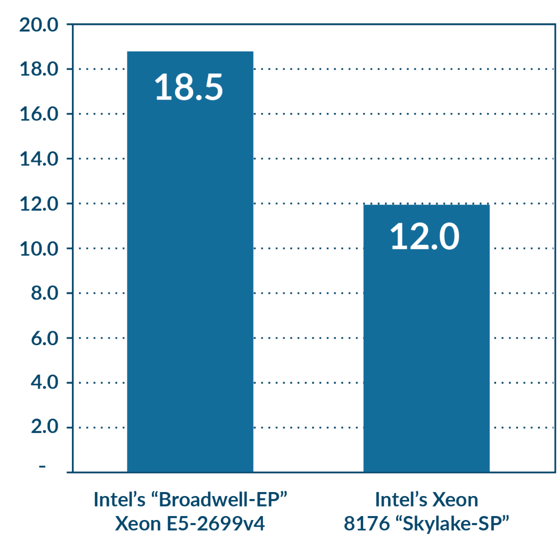


Sources: 1-2, ***spec.org****; 3, IDG*

1. CPU performance increases are slowing overall. The industry response to physical limitations on CPU frequency has been incremental improvements in micro architecture including hyper threading, larger caches, and increasingly large (often impractical) core counts. The net result is a strong rate of improvement in integer and floating point performance per socket, but a much more muted improvement in single thread performance. If all applications were embarrassingly parallel, this might be fine. They are not. One law that is not dead is **Amdahl's Law** which has always identified single thread performance as a fundamental limiter.

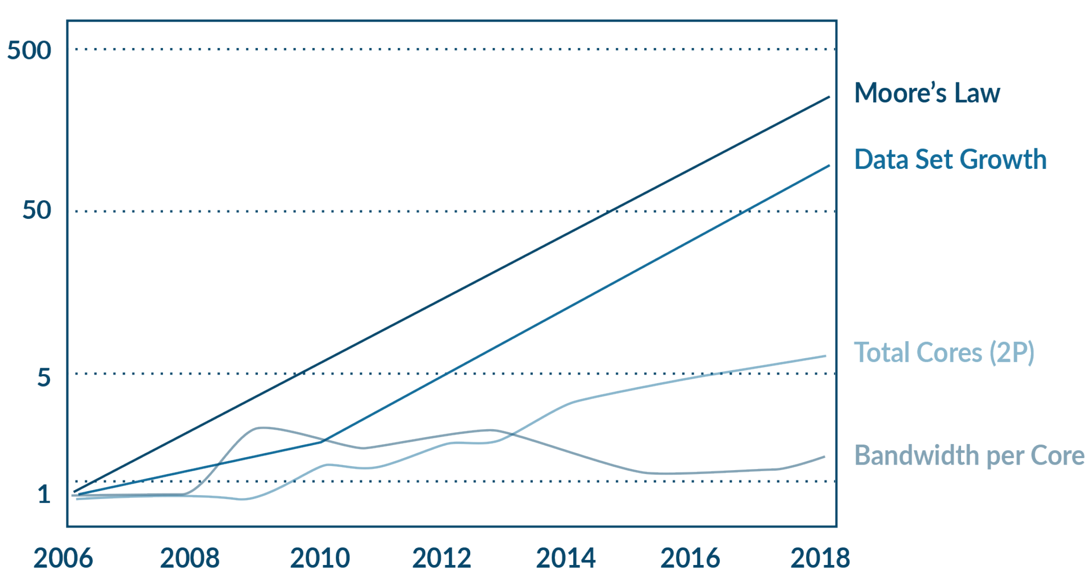
2. Memory bandwidth per core is decreasing. While core counts are increasing rapidly, physics is working against us in scaling memory bandwidth to keep pace. The shared parallel DRAM interface, overall limitations in processor pin counts, and motherboard design complexity limit the aggregate memory bandwidth per socket and severely limit the bandwidth per motherboard. Overall, memory bandwidth per core in recent years is increasing at just 23% per year. Bandwidth per core decreases in the **SkyLake** processor generation. Intel's Xeon 8176 "Skylake-SP" shows only 12MB/sec bandwidth compared to the prior generation's Intel "Broadwell-EP" Xeon E5-2699v4, which sported 18.5MB/sec bandwidth.

Single Core Memory Bandwidth (MB/second)



Source: Intel technical specs

Memory Bandwidth Per Core is Shrinking



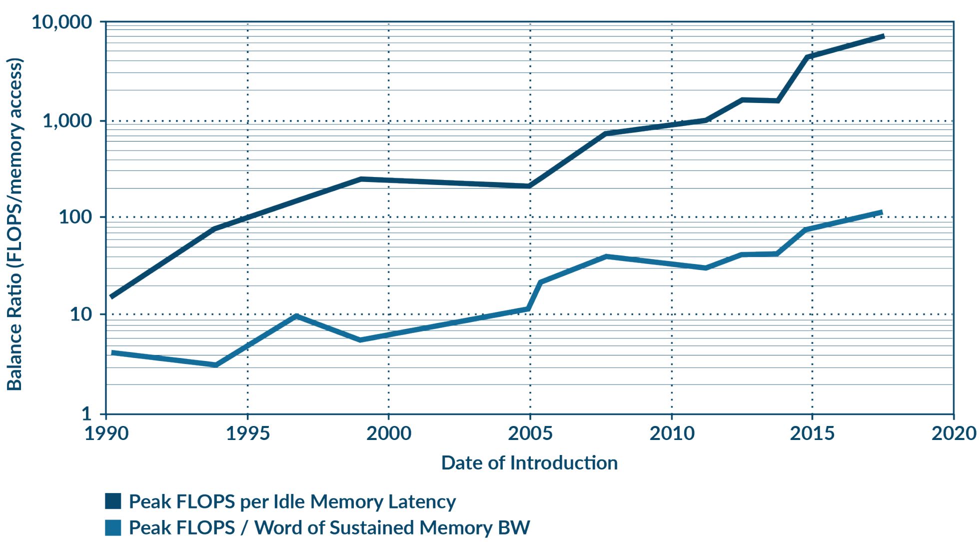
Sources: IDG, Intel technical specs

3. Memory latency is getting worse. Flat to decreasing processor clock speeds with rapidly increasing core counts leads to sluggish increases in memory latency, despite heroic increases in hardware complexity. The result is a three-orders-of-magnitude increase in the performance gap between processor performance improvements and memory performance improvements that is caused by three factors:

- Slow rate of memory pin speed improvements
- Slow rate of increase of memory interface width
- Virtually unchanged DRAM cell cycle time

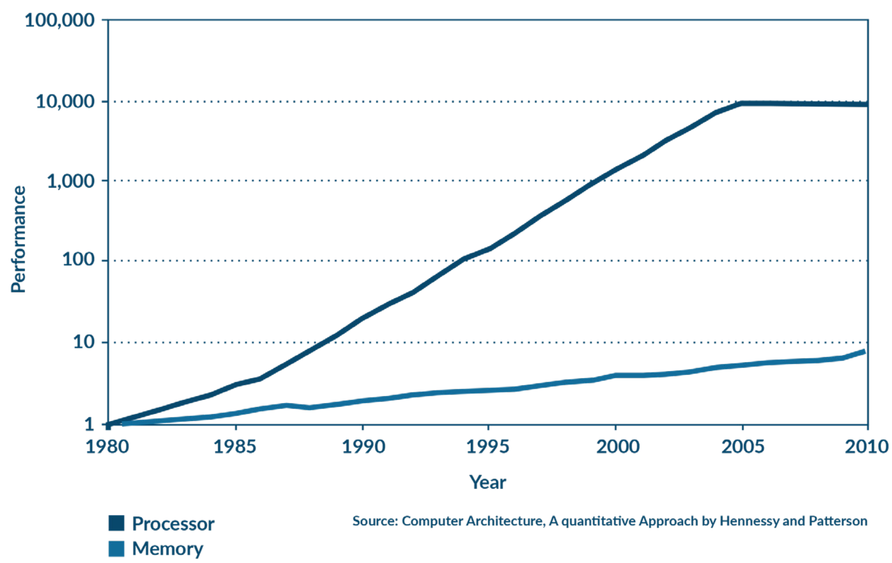
4. Sweet spot servers are compellingly cost effective. The volume sweet spot price points for standard 2-socket servers are significantly more cost-effective compared to 4- or 8-socket servers. On a 3-year TCO analysis of a popular database application on a Software-Defined Server, standard 2 socket servers with cost-effective processors saved 50% of hardware costs and 61% of software licensing costs for a total of 56% in cost savings overall. A software-defined layer that can utilize standard two socket servers as building blocks becomes progressively more cost effective as system size increases. The advantages of using 2-socket servers as the basis for Software-Defined Servers actually increase in the **SkyLake** hardware generation where Intel starts charging a larger premium for big memory server configurations – it takes twice as many sockets to get to the same memory as the Intel Xeon® v4 systems.

Memory Latency is Worsening: (GFLOPs/s) / (Memory Latency)



Source: Texas Advanced Computing Center

The Increasing Gap Between Processor and Memory Performance



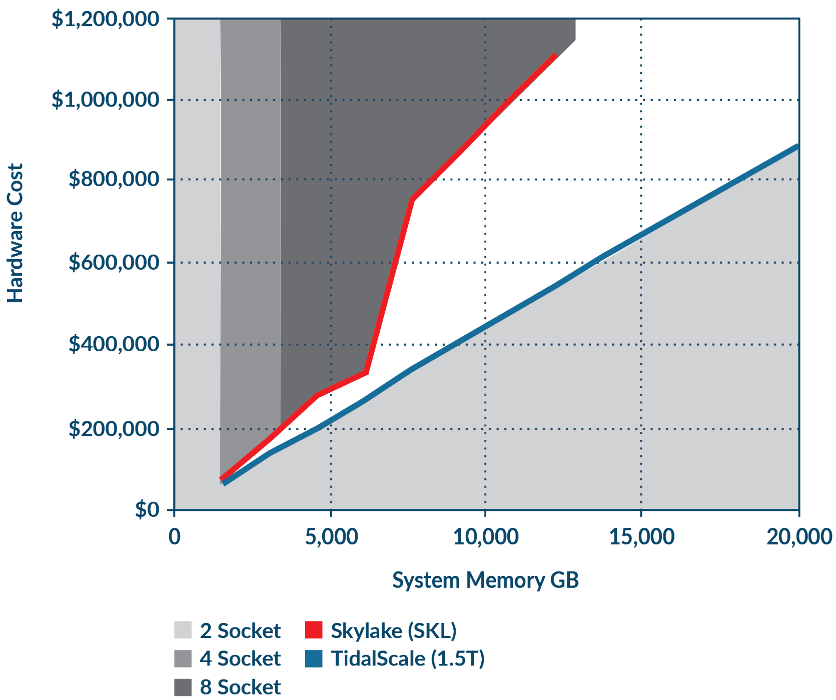
Source: Computer Architecture, A quantitative Approach by Hennessy and Patterson

5. New server designs increase system complexity. In pursuit of performance improvements, hardware designers are introducing a confusing array of new interconnect technologies: **CCIX**, **3D Xpoint**, **GENZ**, **HCM**, **HBM**, **NVMe**. The only thing that is clear about these technologies is that system administrators will be forced to move up a layer in abstraction: these increasingly complex node memory hierarchies strongly suggest the need for the existence of a simplification layer that can enable applications to adapt to whatever technologies are underneath them and extract the best performance automatically. In other words, these hierarchies call for software definition of servers.

6. Composable infrastructure responds to a desire for on-demand resources. Many organizations are deploying **composable infrastructure** platforms because they recognize the business and IT advantages of on-demand resources. Compute resource disaggregation in next-generation rack scale hardware designs is a key part of this effort because it empowers business efficiency and agility. However, most of these platforms lack an easy way to aggregate the most critical resources in the rack, compute and memory, in a logical layer. The need for this missing logical layer is aggravated by shifting user expectations for auto scaling infrastructures implied behind the push to serverless architectures.

7. Machine learning enables software-defined resources to dynamically adapt. Machine learning technologies enable the possibility of software-defined systems tuning themselves to adapt to application needs automatically and dynamically. Affinities

The Growing Cost of Large Memory



Source: TidalScale cost analysis

The low latency requirements for analytics, ad placement, relevance assessment, fraud detection, medical diagnosis, supply chain management, IoT analysis, identification of persons of interest, and more are pushing the requirement for in-memory performance to an extreme.

It's not long before that journey leads to Software-Defined Servers.

between resources like memory, CPU, network, and I/O, can be automatically and efficiently identified so that their virtual placement on physical hardware can deliver optimal performance, and these adjustments can be made orders of magnitude faster than anything currently available. This self-tuning capability is unique to TidalScale.

These technology factors affect in-memory performance solutions and they come at a time when there is a rapid shortening of the time-to-decision window for business users.

Source: TidalScale

Research from Gartner

Should Your Enterprise Deploy a Software-Defined Data Center?

The software-defined data center marks the next evolution in infrastructure and data center automation architecture. However, switching to or deploying an SDDC is not recommended for all IT organizations. I&O leaders must understand the business value, best use cases and risks.

Key Challenges

- I&O organizations are challenged to make I&O faster to respond and scale to meet the business, and to support new application architectures for hybrid cloud solutions, analytics, digital business, the Internet of Things (IoT) and mobile and edge computing.
- Application programming interfaces (APIs) are becoming the “glue” of the digital data center, but an orchestration engine across these APIs is still needed to coordinate the technical capabilities that ultimately provide a business service.
- Infrastructure vendors are actively adding APIs to their platforms, but these are not standard nor open across vendors, or even products from the same vendor.
- In addition to requiring significant deployment work internally, a software-defined data center (SDDC) implementation also requires new IT skill sets and a profound cultural shift.

Recommendations

I&O leaders focused on infrastructure agility:

- Identify business models and IT projects that get the most value from an SDDC, and the technology requirements that make these possible, by using this research.
- Prioritize the SDDC business case based on business outcomes, such as improved time to market, lower operational costs and speed to implement change (agility).
- Start your SDDC initiative with a small, focused use case, learn from it, then scale horizontally.
- Mitigate the top risks of an SDDC project, such as technology immaturity and vendor lock-in, while recognizing that all software and automation incur some degree of lock-in.
- Do not build a complete SDDC orchestration architecture if you are not an early adopter of technology and if you lack deep expertise in I&O engineering and programming; rather, choose to implement specific software-defined components as they mature.

Strategic Planning Assumptions

By 2022, programming skills will be the primary hiring criteria for 50% of I&O roles, up from 20% today.

By 2023, the programmatic capabilities of an SDDC will be considered a requirement for 85% of large global enterprises, up from 25% today.

Introduction

A software-defined data center (SDDC) is the foundation of public cloud infrastructure as a service (IaaS) offerings such as those from Amazon Web Services (AWS) and Microsoft Azure. A software-defined approach enables automation and policy enforcement activities to be selected through self-service catalogs or APIs, as well as enforced programmatically through policy and analytics. Private cloud services rarely implement a complete SDDC today (unless they are OpenStack-based); however, they will increasingly do so as SDDC offerings mature.

An SDDC is becoming critical to achieving the business agility, speed and time to market associated with a digital business. Orchestrating infrastructure such as that required for IoT, DevOps and cloud services is accomplished through software-defined components, enabling greater end-to-end automation, mobility and integration through the use of APIs, and policy orchestration.

SDDC Immaturity

A key part of the challenge is that a complete SDDC across many domains (e.g., compute, networking, storage, security, etc.) remains in an immature state for broad-scale internal implementation (needed to access internal data centers and infrastructure). Therefore, only visionary organizations with advanced I&O engineering and architecture skill sets, or companies with a very isolated use case, will have implemented an SDDC in the near term (one to two years). Others that are interested in the agility benefits that an SDDC brings will seek public cloud providers or on- or off-premises managed private clouds to enable the benefits, without having to fully engineer and implement an SDDC themselves.

However, I&O leaders cannot simply go out and buy a readymade SDDC that is ready to deliver business agility and value. Instead, they must deploy, integrate and orchestrate numerous parts, likely from several public cloud IaaS providers as well as on-premises vendors (see Note 1), such as software-defined storage (SDS), software-defined networking (SDN), software-defined compute (SDC), software-defined security (SDSec) and software-defined facilities (SDFs; see Note 2).

I&O Skills Refresh and Cultural Shift Required

In addition to requiring significant deployment work, an internal SDDC implementation also requires new IT skill sets and a cultural shift. For example, the allocation of network, storage and compute resources to a particular workload will require IT staff to program or set policies for the logic of that allocation, rather than relying on a primarily manual set of processes with support from spreadsheets and system management tools. Moreover, with the advent of self-service infrastructure, I&O leaders are increasingly challenged to enable these needs directly through the use of APIs and service catalogs. This requires I&O to predesign and establish offerings for consumption by application developers, another skill set that is lacking in most I&O organizations. Even when an organization decides to more fully deploy a public cloud IaaS, I&O must enable policy and automation, to, for example, ensure adherence to resiliency policies, backup policies and monitoring policies, all while keeping developers as productive as possible. I&O must also work across the aisles with security and applications and enterprise architecture to establish the right capabilities for access and integration across public cloud and company-owned infrastructure (including sensors for IoT).

As a result of cloud computing, DevOps and digital transformations, many I&O organizations have been evolving to more automated SDDC orchestration capabilities. Yet, progress has often been slow, as a lack of skills, challenges with legacy applications and the requirement to sustain the current environment trump the new capabilities. An additional issue is that very few SDDC products fully extend their functionality into public clouds; that support is only beginning. Another problem is a real lack of industry-standard APIs supported by the product vendors, which can restrict an SDDC implementation to being largely vendor-proprietary.

This is often a scary proposition for I&O leaders, because where there is change there is fear, as well as very real risk. Implementation of an SDDC must be done in the context of a business culture that is not solely focused on speed, but one that is also well-versed in delivering business value and implementation of business strategy. An IT culture that is only focused on underlying technical competence, without a strong linkage to the business drivers, will not be a good candidate to implement an SDDC.

Analysis

Learn About Software-Defined Data Centers

An SDDC is an architectural approach to orchestrating IT infrastructure and facilities to deliver business value such as speed to market, increased automation (like continuous integration/continuous deployment [CI/CD]) and greater employee productivity. As shown in Figure 1, an SDDC enables abstraction of the physical infrastructure, reducing the complexity to access physical infrastructure functionality through a virtualization layer. The virtual layer enables access to the physical through another set of APIs that mask the physical underneath (for example, enabling access to storage without having to know exactly which storage you are accessing). The virtual layer can then either be accessed directly via APIs or through an orchestration layer that enables programmability across many infrastructure components via its APIs. At the orchestration layer, policies can be enabled for rule-based orchestration (for example, to add more infrastructure when demand spikes).

The goal of an SDDC is to deliver greater business value than traditional, more manual processes and scripts by increasing the agility and speed of infrastructure through policy-based automation of processes. This includes infrastructure selection, security, provisioning, and ongoing life cycle management and optimization. As a result, operational cost savings are also a benefit by both lessening administrative requirements and enabling reusability of automation. Today, many of the operational cost savings are attributable to individual SDDC technologies, such as SDS, SDN or SDC.

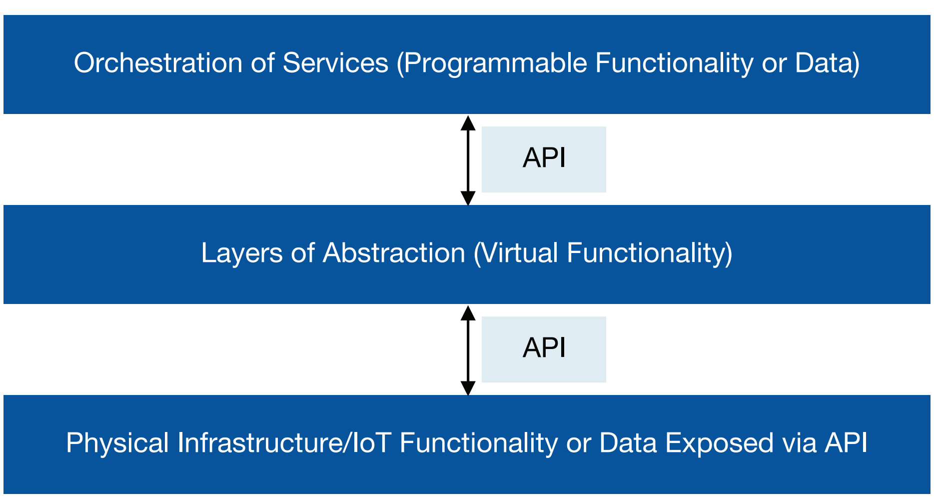
In delivering greater business value, SDDC enables the migration of the software services that support proprietary hardware to a higher-level software service layer, with policies driving configuration, automation and deployment across a distributed environment and the potential for cost savings in capital expenditure by implementing resource pools containing commodity hardware. Moreover, an SDDC may extend to legacy hardware investments, thus increasing their useful life through the abstraction and programmability layers.

Understand SDDC Attributes and Taxonomy

Key attributes of software-defined components and the criteria for an SDDC are as follows:

- **Abstracted** – Of resources from their physical implementation, thus decoupling a resource from its consumer. Abstraction enables the definition of logical models of infrastructure elements that can be instantiated at the time of provisioning and reused, thus enforcing standardization.
- **Instrumented** – Of infrastructure (physical and virtual) for monitoring, publishing and intelligent analytics.
- **Programmable** – Enabling not just hands-off automated provisioning services, but all IT processes through documented APIs (typically RESTful APIs).
- **Automated** – Using the APIs, wiring up the exposed elements using scripts and other automation tools to remove "human middleware" from the equation.
- **Policy-Managed** – Enabling centralized configuration and reconfiguration of the decentralized infrastructure through pre-established policies to meet business needs.

Figure 1. High-Level View of Software-Defined



Source: Gartner (April 2017)

- **Orchestrated** – Beyond script-based automation, enabling automation of tasks across resource domains of compute, network, storage, facilities and security, with linkages to policy management for enforcement and optimization.

The combination of the above capabilities will foster innovations that optimize infrastructure and application use of underlying infrastructure. Examples include selecting the right infrastructure for deterministic SLAs, and dynamically optimizing the infrastructure to increase asset utilization. These capabilities may be acquired independently or as part of a suite of functionality, such as in cloud management platforms (CMPs).

Figure 2 presents the broader taxonomy of SDDC as well as "software-defined anything" (SDx), which is the umbrella term that Gartner uses to describe anything that is software-defined, including applications and infrastructure. The SDDC is a component of software-defined infrastructure (SDI), with SDI building upon and broadening from SDDC to include software-defined devices and software-defined machines, which will be key components of IoT.

SDDC is broken down into its component parts – compute, storage, network and facilities – with applications that exploit software-defined services on top, requesting infrastructure services through the technical services APIs. Security and management are cross-domain services that span applications, application services and infrastructure, and are also API-accessible. Each component could be accessible and managed through its own orchestration layer that offers a set of APIs and policy capabilities, with the desire being to have a single orchestration engine on top of all (a "managers of managers" approach). A catalog of logical blueprints or policy templates enables reusability of a particular infrastructure configuration (such as a "flavor" in OpenStack Compute [Nova], or using OpenStack Heat to generate an infrastructure template, such as a workload-specific storage profile for low latency and high input/output operations per second, with certain SLA guarantees). Examples of policy templates in public cloud IaaS include AWS CloudFormation templates and Microsoft Azure Resource Manager templates.

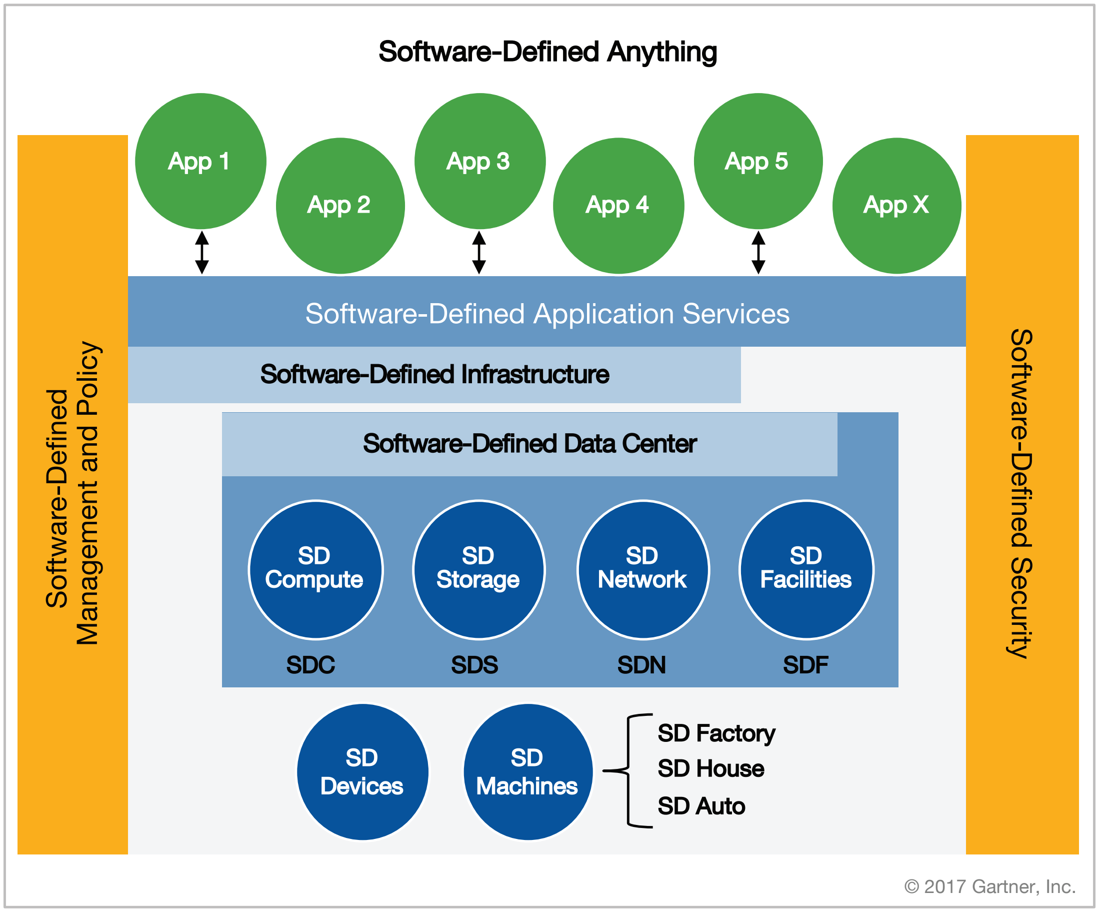
Establish an SDDC When the Business Case Makes Sense

The business value of an SDDC comes through its ability to speed provisioning, streamline IT management processes, and enable cost containment through greater speed of software provisioning and changes, enabling faster time to market for all software, including for digital products and services. Before building an internal SDDC, I&O leaders should determine the ROI of an SDDC based on those three value propositions.

An internally engineered SDDC (as opposed to a cloud-based SDDC) allows for abstraction, orchestration and programmability of on-premises (or colocation-based) infrastructure. For example, this would allow developers to provision and scale infrastructure through an orchestration layer without having I&O personnel required for manual provisioning steps. An internally engineered SDDC would also enable programmability by I&O personnel to, for example, add new capacity to the pool of resources or to dynamically optimize the pool of resources for greatest asset utilization.

It will be critical for IT organizations to develop a cloud strategy to determine how much accessibility to internal programmable infrastructure will be necessary. For example, for enterprises with significant private cloud infrastructure, access by developers will likely be necessary, although that access could be abstracted further by using a platform as a service (PaaS) layer. With PaaS, the I&O team can enable policies to the infrastructure through the PaaS layer, but the developers will not need to access the IaaS layer directly. For enterprises making significant use of public cloud IaaS, most of the developer access will be made directly through the public cloud provider's APIs and

Figure 1. High-Level View of Software-Defined



SD = software-defined
Source: Gartner (April 2017)

orchestration layer. However, there likely will be some requirement for integration of public and on-premises networking and security, therefore requiring internal SDDC, but at a smaller scale than full private cloud IaaS would require.

Table 1 provides a framework for gauging the potential ROI and benefits of an internally engineered SDDC deployment, which should first be justified based on the degree of programmability needed by internal constituents, as discussed above. I&O leaders should note that the ROI of an SDDC will largely come through speed of accessibility and programmability for developers, enabling faster time to market. As a side benefit, I&O personnel costs will be streamlined and reduced because prepackaging repeatable IaaS capabilities for developers will free up the manual activities so common for I&O personnel today. These cost savings will be seen over the long term because most software-defined-based applications will be new, leaving legacy applications and infrastructure to continue to support (in a mostly manual way, until the applications are rationalized and modernized). An internally engineered SDDC can also lead to lower capital expenditure as infrastructure resources are better-utilized and commodity infrastructure can replace more expensive proprietary hardware.

Table 1. SDDC Value and ROI

SDDC Business Value	How This Provides ROI
Faster Time to Delivering Business Value The ability to provide infrastructure abstracted into technical services, enabling functionality required by developers to deliver strategic value. This could include, for example, implementation of IoT to detect customers in stores, renovating the core of IT to enable digital capabilities on top, and reducing friction for developers to access and program infrastructure for more frequent application releases.	■ Enables developers to access self-service via catalog or API, predesigned infrastructure capabilities. This offers greater developer productivity and faster time to market for software. ■ Enables infrastructure policies such as deployment across availability zones, monitoring and backup to be automatically configured, increasing compliance and developer productivity.
Enabling a "Platform" Approach The ability to write and program different aspects of your infrastructure in a way that enables agility, flexibility and mobility, and reduces vendor lock-in (if using open APIs).	■ Reduces overall implementation and configuration complexity and costs. ■ Enables the ability to swap in and out components as needed without application changes.
Faster Provisioning The ability of an SDDC to spin up storage, compute and networking resources in a rapid and standard manner.	■ Federated, cross-domain policy-based configuration and enforcement across distributed infrastructure and data centers reduces the need for manual activities associated with provisioning, configuration and policy enforcement. ■ Reusability and composability of services through API and policy enablement lead to faster and more agile IT service provisioning, which has the opportunity to compress the application development life cycle.
Streamlined IT Management Processes The ability of an SDDC to automate otherwise manual IT processes.	■ Well-documented API access to infrastructure enables greater control, orchestration and automation across distributed infrastructure, eliminating the need for manual activities and custom scripting. ■ Reduces human error through automation.
Cost Containment The use of an SDDC to consolidate systems, enable independent commodity components and better-manage data center growth, plus the two cost-saving benefits listed above.	■ By abstracting the consumed services from the infrastructure, switching out hardware is made easier, allowing for commodity hardware. ■ As agility is increased through an SDDC, overprovisioning of hardware resources can be reduced. ■ Some components of an SDDC may be able to be extended to legacy infrastructure (for example, enabling management of both new and legacy storage under the same APIs, policies and orchestration), thus extending its operational life and reducing its maintenance (labor) costs. ■ Organizations with significant operational expenses in the public cloud may save money through implementation of an on-premises SDDC (trading capex for opex), but only if they have the inclination and expertise to integrate early stage technologies.

Source: Gartner (April 2017)

Implementing an SDDC in both public cloud and on-premises implementations will, however, require new skills, roles and likely professional services. One key new role required is an IaaS product manager, along with delivery managers, to allow repeatable IaaS functionality to be developed and enabled for developers.

Identify the Business Models and IT Projects That Get the Most Value From an SDDC

Faster provisioning, streamlined IT management and cost containment via an SDDC will not always provide a satisfactory ROI. This section looks at the use-case scenarios in which I&O leaders are most likely to realize the value of an SDDC:

- **Small, "greenfield" projects.** New projects that are smaller in size and scope can offer clearer requirements that will more easily allow for proper SDDC product selection. For example, an SDDC project is often added on the back of an already-existing cloud computing initiative. A private cloud may have been implemented, and the enterprise finds that the existing networking is too slow or not scalable enough, or not able to reclaim addresses. At this point, an initiative to investigate and implement SDN would be in order. As legacy data and infrastructure are not involved, coexistence with previous systems will not be an issue. These types of projects allow for a software-defined rollout in a constrained deployment domain (for example, one cell/domain of a virtual desktop infrastructure [VDI] project or analytics) or for a particular application workload. After initial success, expanded usage can be considered. By implementing an SDDC narrowly, it reduces the investment required to trial new technologies and processes. It also allows for on-the-ground training and resulting documentation of lessons, which can be used to assess whether to extend the SDDC to additional use cases.

- **New data centers.** When designing and building out new data center infrastructures, the software, hardware and especially process changes associated with broader SDDC implementations can be more easily justified, as all data center architectures and functions are examined with an eye toward implementing the latest-known best practices. Data center build-out or refurbishing is an opportunity to revisit all past decisions and to begin to introduce gradual to more-complete levels of software-defined capabilities. Some of Gartner's enterprise clients have said that the new data center became the focus for developing top-to-bottom processes for new cloud-native and optimized services. In that sense then, it becomes a very large greenfield environment for new application development and programmable infrastructure. Depending on the results and plans for both new development and refactoring legacy systems, retrofitting additional data centers may be the logical next step.

- **Cloud service infrastructure.** Cloud IaaS services are built on many of the same concepts as software-defined. Abstraction (that is, virtualization) and policies are fundamental to the cloud. I&O can make use of cloud IaaS and the SDDC layer through orchestration. This may be enough for applications that are built only on the IaaS, but any integration to on-premises infrastructure or to other cloud providers will require implementation of a broader SDDC layer that enables cross-cloud integration. In most cases, I&O will be required to be the orchestrator of orchestrators, meaning offering higher-level multicloud and on-premises IaaS integrations, meeting compliance and security requirements. We recommend that I&O become well-versed first with orchestration of each cloud environment before orchestrating across clouds, so as to build on complexity as you learn.

Note: You don't have to do this alone. Consider hiring service providers with advanced cloud and SDDC skill sets to build what needs to be built, and to transfer skills and capabilities to I&O personnel.

Know the Required Characteristics to Make SDDC Possible

The ultimate goal of an SDDC is to enable an IT infrastructure wherein network, storage, compute and security adapt to the needs of applications in an automated way. However, simply deploying and integrating software-defined components (SDS, SDN, SDC, SDSec and SDF), along with other SDDC elements, is not enough. Several technological factors weigh on the ability of an SDDC to deliver on the promise of the automated, adaptable infrastructure. These factors are:

- **The openness of the APIs across physical and virtual storage, networking and compute components.** APIs will form the interconnect between the physical resources of the data center and the application. These APIs must be well-defined (ideally public) and well-behaved (stable), but, at the same time, robust (complete) enough to provide the needed control points and feedback mechanisms to automate and orchestrate operations. Today, the completeness of APIs in any one domain, much less cross-data center, are varied, and they can be brittle and subject to changes that cause issues for solutions that have leveraged them.
    - Most vendors offer proprietary APIs that inhibit cross-silo/technology orchestration, or their orchestration capabilities are optimized for their platform, making it difficult to reach into other providers. Focus on minimizing these constraints during product selection. The marketplace needs to continue to push vendors for greater support for stable, published APIs. It is useful to note that the momentum behind OpenStack offers a common set of APIs from the cloud consumer perspective – meaning that a developer using these APIs will find them to be reasonably stable and open. Any vendor that offers a plug-in to OpenStack must meet the API requirements. Therefore, as long as standard APIs are used, enterprises can migrate from one infrastructure vendor to another without impacting the applications and processes that were built on top of these APIs.
    - Because vendors offer features beyond those that OpenStack requires, it means that migration will require disruption if those features are used in the orchestration. The degree of benefit will depend on the degree of standardness – and that will always be counter to vendors' profit motives. Thus, API programs overall could be viewed at times as either uplifting or hampering feature exploitation, depending on whether the capability is offered, extended or unsupported by the API.
- **The degree of interoperability across storage, networking, compute, security and facilities components.** Interoperability can be challenging today in any one area, such as storage, but becomes greater in necessity and complexity as the many domains of the data center are combined in an attempt to deliver complete functional capabilities that I&O personnel don't have to build themselves. If interoperability across the solutions is lacking, reduced functionality and vendor lock-in become real concerns as compute, networking, storage and overall management (including operation, security and data center facilities) will then have to be treated in silos. Alternatively, going with one supplier (such as one public cloud provider or one integrated system provider) can ease the development of prebuilt orchestration capabilities, but at the expense of a lock-in to a single provider.

- **The degree of commoditization across storage, networking and compute components.** The word "commoditization" sometimes is misinterpreted as implying that something has a low value. Oil is a good example that clearly has a high degree of value, but is a true commodity in that it can be sourced from many suppliers. The availability of an API set is a key ingredient in allowing an SDDC to flourish across a multivendor ecosystem. If APIs can be leveraged across low-level components to enable building blocks of physical resources, then reduction or elimination of vendor lock-in becomes more realistic. This is because APIs enable orchestration across the physical and virtual resources, which can lead to a broader range of choices and greater asset reuse. At the end of the day, this may be a case of "choose your lock-in."

- **The degree to which enterprises implement roles and skills to harness the power of an SDDC.** An SDDC does not implement itself. It requires thought-leading infrastructure platform architects and specialists to redesign the infrastructure and establish how it will be programmed for use by IT and application development. Organizations that view infrastructure as their core competence will likely have the most success in redesigning infrastructure and processes utilizing the SDDC to gain business benefit. Those that don't possess these skills should seek to gain programmability through public or managed private cloud services. However, even with public or managed private cloud, I&O personnel need to develop prebuilt I&O building blocks and functionality that development can use to increase developer productivity while maintaining compliance, security and policies.

Plan to Mitigate the Top Risks of an SDDC Project Failure

Unfortunately, many emerging I&O projects fail during the implementation phase. This is a very visible and costly time to discover what can be very fundamental issues. Planning is the key to ensuring that risk can be minimized and that avoidance of issues can be maximized. Other factors can also deter a successful SDDC deployment. These include:

- **Lacking a process/culture to deploy and manage an SDDC.** Simply changing out a legacy infrastructure for a set of software-defined products is unlikely to yield the desired benefits of an SDDC. Put frankly, a broken process, although better-automated, is still a broken process. Before an activity is automated and self-service is implemented, the process associated with the IT service needs to be completely rethought and optimized. This may require new skills and a different culture than what exists in the data center today. Bimodal IT, DevOps and software-defined are example areas where current methods and staffing capabilities will need to be augmented with new skills and processes in order to be properly implemented and deliver the value. Alternatively, an organization could elect to go with a vendor-specific reference architecture, such as what VMware has with VMware Validated Designs.

- **Addressing skills/training.** Because some elements of a software-defined approach are still emerging, and can lack turnkey solutions, skills and training are often lacking. I&O personnel developing these capabilities will need to have programming skills, not just administrative skills. Addressing skills and training is accomplished through retraining, hiring and attrition. I&O personnel who do not want to learn programming should expect to have a short career in any organization that is pursuing digital business. To further mitigate these risks, plan to have a few of your best enterprise architects learn by doing, as well as attend conferences for training and to meet others to collaborate with. I&O leaders can also download OpenStack community code, attend OpenStack conferences, experiment with public cloud IaaS, and attend Gartner conferences where they can network with attendees and share lessons and best practices. They can also get access to public cloud services and practice orchestration of prebuilt components. Additionally, contributing back to open-source projects and communities will help foster the right culture.

- **Not being ready for an SDDC when the time for an SDDC comes.** Nearly all IT organizations are under pressure to deliver faster. An SDDC represents the way to orchestrate increasingly differentiated application functionality on top of infrastructure in an automated, policy-based way. Regardless of whether exposing on-premises infrastructure capabilities or taking advantage of public cloud, I&O must gain software-defined expertise broadly across the organization. I&O personnel that do not get on board with software-defined will be custodians of legacy – and likely outsourced. Only then should I&O architects assess internal capabilities. Preparing for an internal SDDC will take time – even more so for organizations that are not well-automated today and that are more manual in their IT services delivery. The first step is understanding the concepts of software-defined. Then, I&O leaders should examine the available SDDC solutions, typically starting with one component, one software-defined domain or one process that can benefit. Lastly, after addressing the items in this section, they should plan an intercept course, if and when deploying SDDC solutions are appropriate.

- **The technology underlying internal SDDC lacks maturity**. Software-defined in any segment is still nascent in adoption, as evidenced by the positioning of these technologies in Gartner Hype Cycle documents. Even more established software-defined areas, such as SDN and SDS, are still gelling and experiencing primarily early-stage adoption levels for visionary enterprises, especially those that write a lot of software and host it internally. Even enterprises that adopt significant public cloud IaaS will likely find they need to integrate to on-premises infrastructure and security. It is critical to understand your cloud strategy in order to know which components of an SDDC to prioritize for internal engineering. It is important to look at the specific capabilities in data center segments that are of high interest, and to decide if currently available products deliver enough functionality, interoperability and production-proven deployment history to be considered for a particular environment. Implementing in phases is recommended, and SDS can be a compelling starting point, as the capabilities offered can compare favorably to traditional storage solutions.

- **Vendor lock-in.** Despite open-source standards, such as OpenStack, physical infrastructure APIs are not standard. OpenStack or a cloud management platform may help you to reduce vendor lock-in, but won't eliminate it. In addition, since there are no standards on physical infrastructure APIs, writing to them will result in lock-in. Understanding that lock-in, and establishing migration costs or exit plans, will help in choosing vendors and technologies that reduce, but don't eliminate, vendor lock-in. Also, recognize that vendors that dictate an architectural blueprint can lead to trade-offs or compromises that should be considered ahead of deployment. You will want to look out for trading a hardware lock-in for a software lock-in, which has always been the case, so choose the lock-in consciously. Public cloud IaaS has similar lock-in issues. As a result, IT organizations should consider lock-in when developing their cloud strategies.

Evidence

This research is based on:

- Over 1,000 client inquiries handled by the authors and peer analysts regarding software-defined year after year, since 2013
- Gartner Data Center Conference kiosk and session polling results from December 2013 to 2016

Note 1

Example SDDC Vendors

All traditional data center infrastructure vendors are offering various SDDC-oriented solutions, emphasizing their core strengths. None are complete SDDC solutions (meeting all criteria across all areas) and, with a lack of standards, most will not interoperate. As a result, buyers should be cautious.

The following are sample vendors offering solutions for SDC, SDS, SDN, SDF and SDSec:

- SDC: Docker, Hewlett Packard Enterprise (HPE), Huawei, Intel, Microsoft, Nutanix, Oracle, Parallels, Red Hat, Unisys and VMware
- SDS: Atlantis Computing, DataCore, Dell EMC, FalconStor, Hedvig, HPE, IBM, Maxta, Microsoft, Nexenta, Primary Data, Red Hat, Scality, StarWind Software, StorMagic, SUSE, Veritas and VMware
- SDN: Big Switch Networks, Brocade, Cisco, HPE, Juniper Networks, NEC, Nuage Networks and VMware
- SDF: IO, Nlyte Software, Schneider Electric and Vertiv
- SDSec: Catbird, Certes Networks, CloudPassage, Fortinet, Illumio, Security First Corp., Trend Micro, Unisys, vArmour, Vidder and VMware

Note 2

Software-Defined Facilities

SDFs is the least mature area of the SDDC, and Gartner believes that compute, network and storage will be the primary SDDC areas of focus over the next many years. As SDFs mature, data center infrastructure management (DCIM) tool vendors will add more API enablement to their solutions, as well as analytics, to enable facility metrics to be taken into account for data center optimization. An example is moving workloads when power is low on a rack.

Source: Gartner Research, G00316477, Dave Russell, Donna Scott, 27 April 2017

How TidalScale's Software-Defined Servers Work

TidalScale pioneered the Software-Defined Server. To enable a single system image running across multiple commodity servers, TidalScale invented the TidalScale HyperKernel. The HyperKernel binds multiple physical computers into a single, coherent virtual system that allows you to run an unmodified guest operating system. No changes to your existing applications are required. All the resources associated with the aggregated systems – memory, cores, storage and I/O – are available to you and your application. In a data center, you can configure as many of whatever size aggregated systems are needed on the fly.

Inverse hypervisor. The TidalScale HyperKernel is an *inverse hypervisor technology*. An instance of the HyperKernel runs on each physical server. Unlike traditional virtualization, where multiple virtual machines run on a single physical server, TidalScale provides a single virtual machine that runs across multiple physical servers. Each Software-Defined Server appears as a single system, even though it may be composed of dozens of individual physical servers. This is because each TidalScale HyperKernel instance communicates with the others, combining those physical machines together via a commodity Ethernet network. The Ethernet acts as a combined memory and I/O bus. The combined physical servers behave as if they were a single large server.

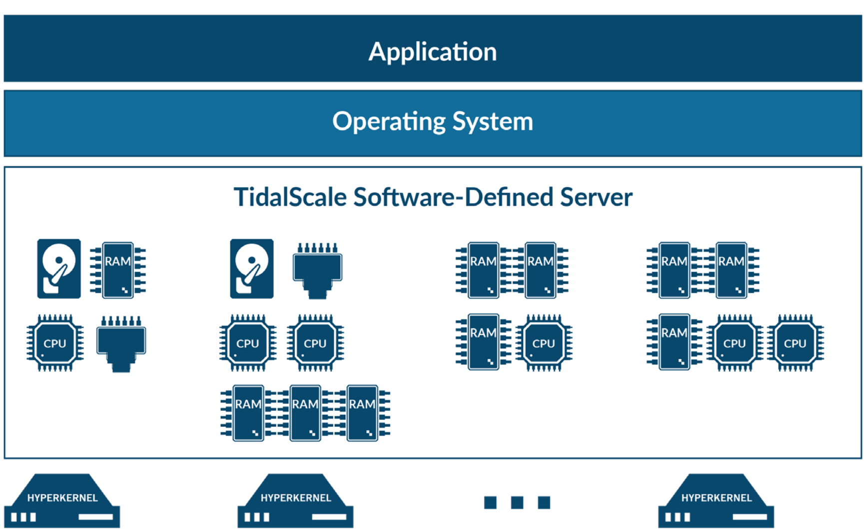
Taken together, individual instances of the TidalScale HyperKernel sit between the Guest Operating System and the set of physical servers. Low-level and fast, the combined HyperKernels act as the hardware level interface to the Guest OS.

TidalScale WaveRunner

Using the TidalScale WaveRunner point-and-click control panel, users can create and boot a new Software-Defined Server in as little as five minutes. A TidalScale Software-Defined Server:

- **Configures on the fly.** One can adjust the size of a Software-Defined Server simply by adding and subtracting physical servers via a GUI or a RESTful API.
- **Self-optimizes.** The HyperKernel uses patented machine learning technology to automatically migrate memory pages and virtual CPUs to where the application needs them to optimize performance.
- **Accelerates application performance.** The HyperKernel takes advantage of the lower latencies of Ethernet networks and the memory management hardware acceleration in the latest CPUs (Intel's VT-x and VT-d) to algorithmically and automatically tune the performance of your Software-Defined Servers.
- **Simplifies SDDC administration.** Virtually anyone, no matter his or her technical expertise, can use the WaveRunner point-and-click control panel to quickly and easily configure and boot new Software-Defined Servers on the fly, monitor virtual server status, and control and configure storage and network infrastructure presented to the guest OS.

The TidalScale Software-Defined Server Stack



Source: TidalScale

Who Benefits from Software-Defined Servers?

Users of in-memory databases analytics applications. *The processing latency of cache-enabled DRAM is roughly 1,000 times faster than flash memory. It's little wonder that in-memory applications and databases are increasingly popular. Applications typically perform better when they access DRAM rather than storage. For instance, on the memory-intensive R benchmark, a Software-Defined Server composed of five 128GB nodes delivered **300X** the performance of a single 128GB bare metal server.*

Organizations that run containers. *A popular, lightweight alternative to virtualization, containers offer a way to match resources on a server to the workload generated by applications and micro-services. But containers aren't without their shortcomings. They lack the mobility, orchestration and security advantages of traditional virtualization—advantages that running containers on Software-Defined Servers in large part restores. In addition, container users will be able to realize new efficiencies (with **throughput gains of more than 20X**) and reduce communication overhead.*

Cloud IaaS customers. *Many organizations turn to cloud infrastructure-as-a-service providers to deliver some of the benefits of an SDDC. But most services limit the size of their largest server instances to no more than 2TB, and still others provide pre-established configurations that are often too large for the workload at hand that results in users renting cloud server resources they don't really need. Software-Defined Server capabilities can be applied to IaaS services to right-size resources, ensuring customers get as much computer as they need, while paying only for what their application demands.*

Enterprise users looking to cut licensing costs. *A significant benefit of right-sizing Software-Defined Servers is that business application users only pay for the cores or memory that their enterprise applications require. No longer do you have to buy servers and install software licenses on hardware deployments sized for 2-3 years in the future. In addition, Software-Defined Servers can be configured to expose only as many cores as the application needs today. This enables provisioning of lower-core count, higher performance CPUs on large memory systems and avoids the hardware and software expense of overprovisioning.*

DevOps managers. *Software-Defined Servers offer four key benefits to DevOps. First, the ability to **combine multiple commodity servers on the fly** from ordinary sweet spot cloud servers using standard Ethernet networks. Second, the **flexibility to change system size on a moment's notice**. Third, a **clean REST API** that lets data center automation infrastructure create, modify and disassemble Software-Defined Servers on demand. **And fourth, the ability to increase cloud hardware utilization by an order of magnitude**.*

Source: TidalScale

Tips for Establishing Your Own SDDC **A PLANNING GUIDE**

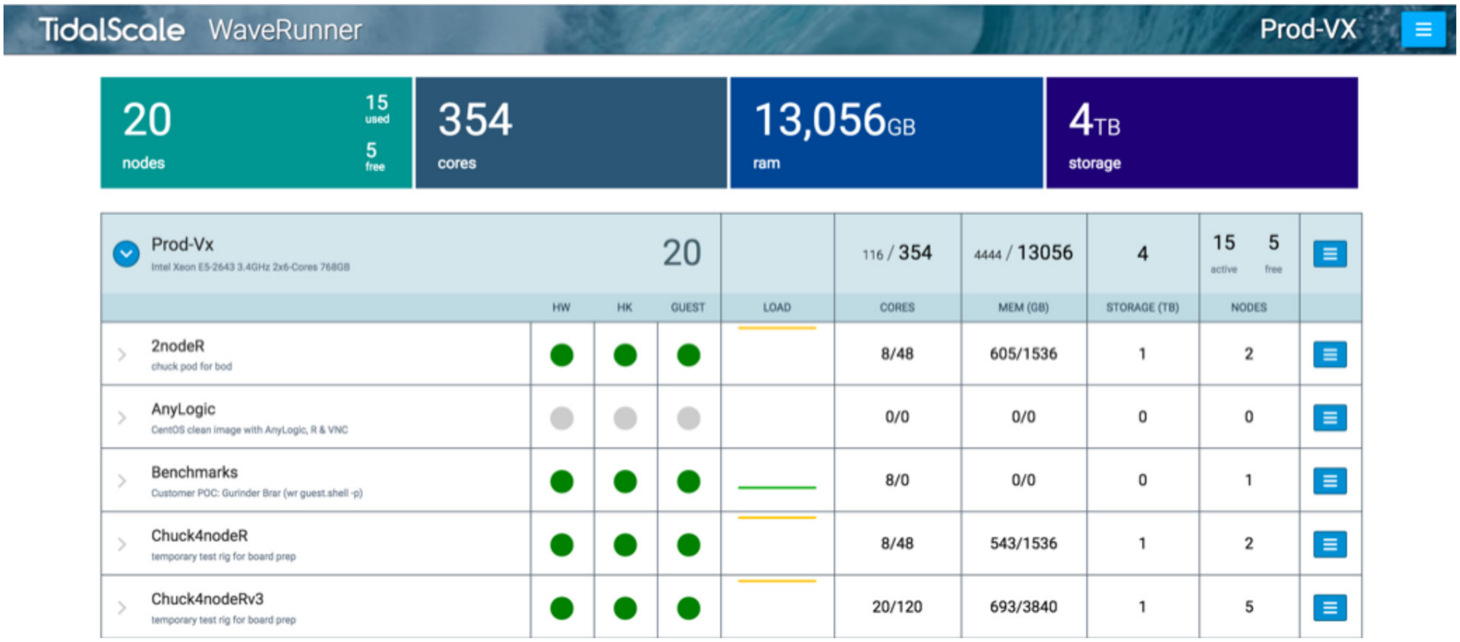
Establishing your own Software-Defined Data Center isn't the arduous task many might envision. Solutions for software-defined networks and storage have existed for years and are well known to IT administrators. But what about Software-Defined Servers?

Fortunately, TidalScale makes this piece easy as well. TidalScale **WaveRunner** is the administration server that allows you to quickly and easily configure, boot and monitor your Software-Defined Servers along with other software-defined resources in your datacenter. The WaveRunner GUI turns the management of TidalPods into an intuitive, point-and-click experience. Using WaveRunner, administrators can easily manage and optimize resources across hundreds of servers, storage volumes, and networks.

Configuration and process considerations. To achieve optimal utilization, configure worker nodes to be stateless. Stateless worker nodes enable any guest OS to use any hardware resource, which increases guest mutability on server resources. Here are some configuration suggestions that support stateless deployments:

- Keep servers "stateless" so they can be used for any workload. For instance, avoid local storage except for transient data. Instead, keep all persistent data volumes and boot volumes as volume images in TidalScale WaveRunner's ZFS server or in a separate NAS.

TidalScale WaveRunner Control Panel



Source: TidalScale

- Allocate servers only when a workload needs to be run. At other times, leave the servers unassigned and available in the TidalPool. An even better solution is to use a job scheduler like Jenkins to grab resources only when applications need to run. In TidalScale's internal testing infrastructure, we achieved a 20X test throughput improvement using simple script integration between Jenkins and WaveRunner and using Containers to manage test images.

Picking cost effective worker nodes for your Software-Defined Servers. Determining which servers to use depends on the size of the system you need and the budget you have. If you have existing servers on hand, you can use them to create large Software-Defined Servers at tremendous cost savings when compared to buying new large hardware systems.

However, if your situation calls for the purchase of some new systems, consider these tips. Smaller nodes can be used to build single-rack Software-Defined Servers up to 60TB in size at a very reasonable $17-18/GB. Higher end systems may cost more up-front but they can be used to assemble a Software-Defined Servers reaching 120TB capacity in a single rack at a reasonable $18/GB. In both cases, it pays to use the higher density 64GB and 128GB DIMMs if they are available for the target server. The $/GB sweet spot servers are any of these servers loaded with 64GB or 128GB DIMMs.

Whatever size servers are chosen for TidalPools, a few simple, rack-level design considerations are important for maximizing resource utilization in a data center:

- For each TidalScale worker node, provide 10GB and 1GB top-of-rack switches.
- Provide high-bandwidth connectivity between the WaveRunner server and worker nodes to realize the best iSCSI performance for boot and data volumes hosted to guests by WaveRunner.

- TidalScale does not have a specific requirement for local storage drives on worker nodes. For convenience and flexibility, shared storage from the WaveRunner server is recommended. For local storage, a single SSD on each node (for example, 480GB) for logging purposes or local scratch space is recommended but not mandatory. Each node does not need to have the same number of local storage drives, or the same size for the drives.
- Standardize worker node storage configuration; for example, 1 SSD for transient data (minimum), additional SSDs and hard drives for workloads as needed. Worker node storage devices need to be pre-partitioned and then added to Software-Defined Server configurations using the wr device.add command.

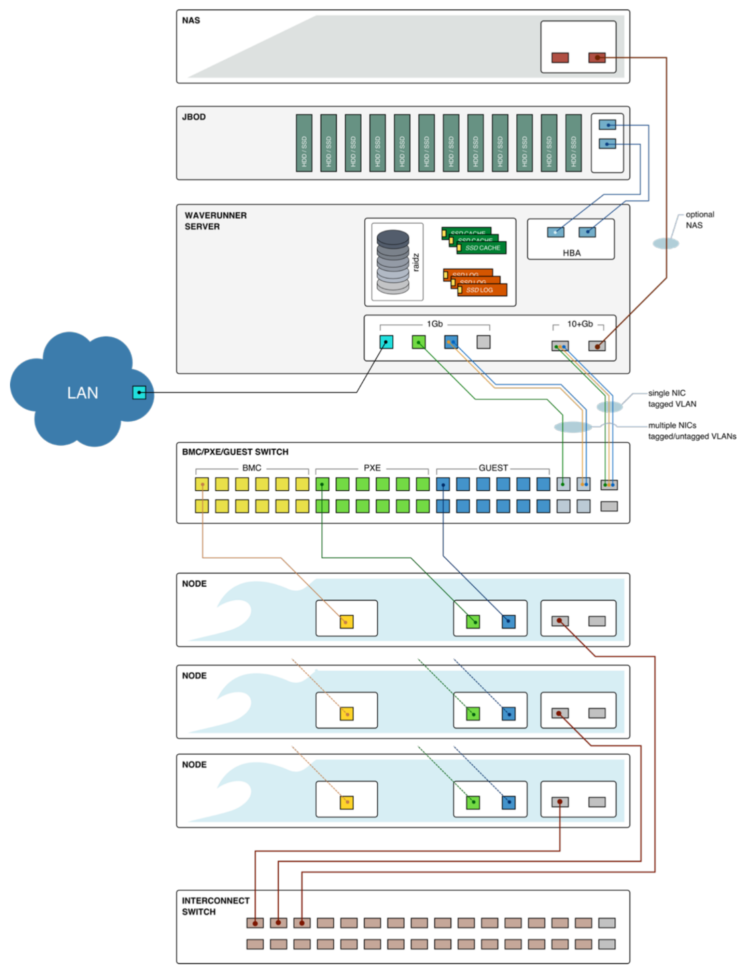
Select a Physical Server to Host WaveRunner

WaveRunner, the TidalScale administration server, is a separate physical system with TCP/IP network connections to all the compute resources under its control: servers, switches, and network attached storage. If WaveRunner is acting as the Network gateway and iSCSI Target then it is recommended that this server be a physical system. Otherwise, the WaveRunner server itself can be a virtual machine.

All the resources of the Software-Defined Servers in this deployment can now appear as a single coherent system. This brings new flexibility to data centers, both on-premise and in the cloud, and enables data center administrators to derive more useful life and increased value from their existing resources.

Example WaveRunner Network Diagram

WaveRunner Example Network with PXE fast path, attached JBOD & external NAS



Source: TidalScale

About TidalScale

TidalScale is the leading provider of Software-Defined Servers. By right-sizing servers on the fly, TidalScale helps organizations achieve results sooner and at a lower cost than traditional scaling options, while bringing new levels of flexibility to data centers. TidalScale solutions deliver in-memory performance at any scale, are self-optimizing, use standard hardware and are compatible with all applications and operating systems – and they achieve all this with no changes to applications or operating systems. TidalScale transforms the economics of the data center and the traditional time constraints of working with big data.

Contact us

Learn more at **tidalscale.com**.
Or contact us directly at 650-535-2204.

TidalScale