

# DIGGING INTO THE SOFTWARE DEFINED DATA CENTER

---

The software defined data center is a relatively new buzzword embraced by the likes of EMC and VMware. For an introduction to the concept see my article over at Network Computing (<http://www.networkcomputing.com/data-center/the-software-defined-data-center-dissect/240006848>.) This post is intended to take it a step deeper as I seem to be stuck at 30,000 feet for the next five hours with no internet access and no other decent ideas. For the purpose of brevity (read: laziness) I'll use the acronym SDDC for Software Defined Data Center whether or not this is being used elsewhere.)

First let's look at what you get out of a SDDC:

## **Legacy Process:**

In a traditional legacy data center the workflow for implementing a new service would look something like this:

1. Approval of the service and budget
2. Procurement of hardware
3. Delivery of hardware
4. Rack and stack of new hardware
5. Configuration of hardware
6. Installation of software
7. Configuration of software
8. Testing
9. Production deployment

This process would vary greatly in overall time but 30-90 days is probably a good ballpark (I know, I know, some of you are wishing it happened that fast.)

Not only is this process complex and slow but it has inherent risk. Your users are accustomed to on-demand IT services in their personal life. They know where to go to get it and how to work with it. If you tell a business unit it will take 90 days to deploy an approved service they may source it from outside of IT. This type of shadow IT poses issues for security, compliance, backup/recovery etc.

### **SDDC Process:**

As described in the link above an SDDC provides a complete decoupling of the hardware from the services deployed on it. This provides a more fluid system for IT service change: growing, shrinking, adding and deleting services. Conceptually the overall infrastructure would maintain an agreed upon level of spare capacity and would be added to as thresholds were crossed. This would provide an ability to add services and grow existing services on the fly in all but the most extreme cases. Additionally the management and deployment of new services would be software driven through intuitive interfaces rather than hardware driven and disparate CLI based.

The process would look something like this:

1. Approval of the service and budget
2. Installation of software
3. Configuration of software
4. Testing
5. Production deployment

The removal of four steps is not the only benefit. The remaining five steps are streamlined into automated processes rather than manual configurations. Change management systems and

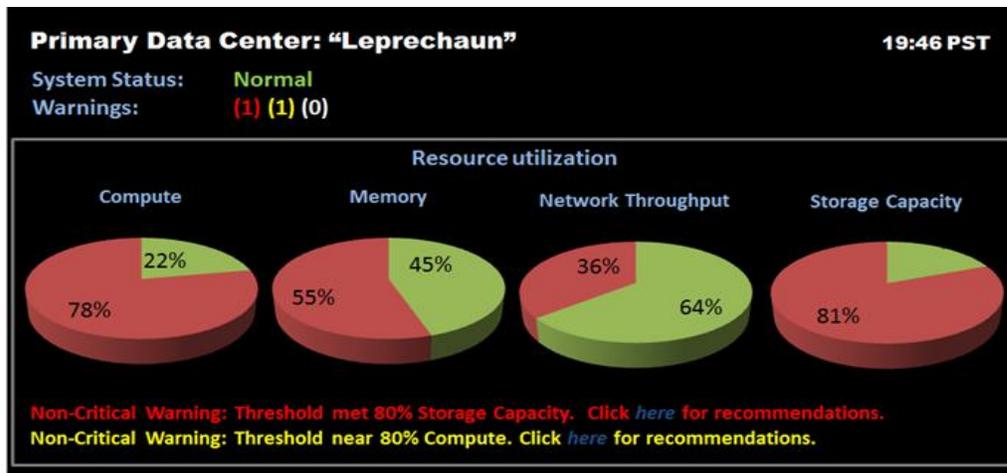
trackback/chargeback are incorporated into the overall software management system providing a fluid workflow in a centralized location. These processes will be initiated by authorized IT users through self-service portals. The speed at which business applications can be deployed is greatly increased providing both flexibility and agility.

### **Isn't that cloud?**

Yes, no and maybe. Or as we say in the IT world: 'It depends.' SDDN can be cloud, with on-demand self-service, flexible resource pooling, metered service etc. it fits the cloud model. The difference is really in where and how it's used. A public cloud based IaaS model, or any given PaaS/SaaS model does not lend itself to legacy enterprise applications. For instance you're not migrating your Microsoft Exchange environment onto Amazon's cloud. Those legacy applications and systems still need a home. Additionally those existing hardware systems still have value. SDDC offers an evolutionary approach to enterprise IT that can support both legacy applications and new applications written to take advantage of cloud systems. This provides a migration approach as well as investment protection for traditional IT infrastructure.

### **How it works:**

The term 'Cloud operating System' is thrown around frequently in the same conversation as SDDC. The idea is compute, network and storage are raw resources that are consumed by the applications and services we run to drive our businesses. Rather than look at these resources individually, and manage them as such, we plug them into a management infrastructure that understands them and can utilize them as services require them. Forget the hardware underneath and imagine a dashboard of your infrastructure something like the following graphic.



The hardware resources become raw resources to be consumed by the IT services. For legacy applications this can be very traditional virtualization or even physical server deployments. New applications and services may be deployed in a PaaS model on the same infrastructure allowing for greater application scale and redundancy and even less tie to the hardware underneath.

### **Lifting the kimono:**

Taking a peak underneath the top level reveals a series of technologies both new and old.

Additionally there are some requirements that may or may not be met by current technology offerings. We'll take a look through the compute, storage and network requirements of SDDC one at a time starting with compute and working our way up.

Compute is the layer that requires the least change. Years ago we moved to the commodity x86 hardware which will be the base of these systems. The compute platform itself will be differentiated by CPU and memory density, platform flexibility and cost. Differentiators traditionally built into the hardware such as availability and serviceability features will lose value. Features that will continue to add value will be related to infrastructure reduction and enablement of upper level management and virtualization systems. Hardware that provides flexibility and programmability will be king here and at other layers as we'll discuss.

Other considerations at the compute layer will tie closely into storage. As compute power itself has grown by leaps and bounds our networks and storage systems have become the bottleneck. Our systems can process our data faster than we can feed it to them. This causes issues for power, cooling efficiency and overall optimization. Dialing down performance for power savings is not the right answer. Instead we want to fuel our processors with data rather than starve them. This means having fast local data in the form of SSD, flash and cache.

Storage will require significant change, but changes that are already taking place or foreshadowed in roadmaps and startups. The traditional storage array will become more and more niche as it has limited capacities of both performance and space. In its place we'll see new options including, but not limited to migration back to local disk, and scale-out options. Much of the migration to centralized storage arrays was fueled by VMware's vMotion, DRS, FT etc. These advanced features required multiple servers to have access to the same disk, hence the need for shared storage. VMware has recently announced a combination of storage vMotion and traditional vMotion that allows live migration without shared storage. This is available in other hypervisor platforms and makes local storage a much more viable option in more environments.

Scale-out systems on the storage side are nothing new. Lefthand and Equallogic pioneered much of this market before being bought by HP and Dell respectively. The market continues to grow with products like Isilon (acquired by EMC) making a big splash in the enterprise as well as plays in the Big Data market. NetApp's cluster mode is now in full effect with OnTap 8.1 allowing their systems to scale out. In the SMB market new players with fantastic offerings like Scale Computing are making headway and bringing innovation to the market. Scale out provides a more linear growth path as both I/O and capacity increase with each additional node. This is contrary to traditional systems which are always bottle necked by the storage controller(s).

We will also see moves to central control, backup and tiering of distributed storage, such as storage blades and server cache. Having fast data at the server level is a necessity but solves only part of the problem. That data must also be made fault tolerant as well as available to other systems outside the server or blade enclosure. EMC's VFCache is one technology poised to help with this by adding the server as a storage tier for software tiering. Software such as this places the hottest data directly next to the processor with tier options all the way back to SAS, SATA, and even tape.

By now you should be seeing the trend of software based feature and control. The last stage is within the network which will require the most change. Network has held strong to proprietary hardware and high margins for years while the rest of the market has moved to commodity. Companies like Arista look to challenge the status quo by providing software feature sets, or open programmability layered onto fast commodity hardware. Additionally Software Defined Networking (<http://www.definethecloud.net/sdn-centralized-network-command-and-control>) has been validated by both VMware's acquisition of Nicira and Cisco's spin-off of Insieme which by most accounts will expand upon the CiscoOne concept with a Cisco flavored SDN offering. In any event the race is on to build networks based on software flows that are centrally managed rather than the port-to-port configuration nightmare of today's data centers.

This move is not only for ease of administration, but also required to push our systems to the levels required by cloud and SDDC. These multi-tenant systems running disparate applications at various service tiers require tighter quality of service controls and bandwidth guarantees, as well as more intelligent routes. Today's physically configured networks can't provide these controls. Additionally applications will benefit from network visibility allowing them to request specific flow characteristics from the network based on application or user requirements.

Multiple service levels can be configured on the same physical network allowing traffic to take appropriate paths based on type rather than physical topology. These network changes are require to truly enable SDDC and Cloud architectures.

Further up the stack from the Layer 2 and Layer 3 transport networks comes a series of other network services that will be layered in via software. Features such as: load-balancing, access-control and firewall services will be required for the services running on these shared infrastructures. These network services will need to be deployed with new applications and tiered to the specific requirements of each. As with the L2/L3 services manual configuration will not suffice and a 'big picture' view will be required to ensure that network services match application requirements. These services can be layered in from both physical and virtual appliances but will require configurability via the centralized software platform.

Source: <http://www.definethecloud.net/digging-into-the-software-defined-data-center/>