AvidThink

Sponsored by Hewlett Packard Enterprise | intel

Scaling the Edge with Redfish®

Implications for Redfish in a Blue Ocean

RESEARCH BRIEF

Table of Contents

Research Briefs are independent content created by analysts working for AvidThink LLC. These reports are made possible through the sponsorship of our commercial supporters. Sponsors do not have any editorial control over the report content, and the views represented herein are solely those of AvidThink LLC. For more information about report sponsorships, please reach out to us at **research@avidthink.com**.

About AvidThink™

AvidThink is a research and analysis firm focused on providing cutting edge insights into the latest in infrastructure technologies. Formerly SDxCentral's research group, AvidThink launched as an independent company in October 2018. Over the last five years, over 110,000 copies of AvidThink's research reports (under the SDxCentral brand) have been downloaded by 40,000 technology buyers and industry thought leaders. AvidThink's expertise covers Edge and IoT, SD-WAN, cloud and containers, SDN, NFV, hyper-convergence and infrastructure applications for AI/ML and security. Visit AvidThink at **www.avidthink.com**.

RESEARCH BRIEF

Scaling the Edge with Redfish®

Implications for Redfish in a Blue Ocean

Introduction

SD-WAN, 5G, IoT, NFV, MEC, edge computing: extremely popular buzzwords today that both enterprises and communication service providers (CSPs) are likely to be bombarded with. In particular, many in the business press are claiming that the edge computing market will be even bigger than the cloud market today. Unlike the red seas of today's cloud data centers, the blue ocean that represents the edge market is rife with opportunities. Visions abound of mini or micro data centers running in remote locations: oil rigs in the ocean, cruise liners, telco central offices (COs), and even mobile cell towers. With a market that is forecast to reach $28.8 billion by 2025 at a CAGR of 54%[1], edge computing is hot.

Couple this with the IoT wave, and analyst firm IDC's estimation of 41.6 billion IoT connected devices by 2025[2], and we'll have an explosion of remote platforms, many of which could be IoT gateways running at the edge.

Who is the DMTF?

The Distributed Management Task Force (DMTF) was founded in 1992 and is a trade organization recognized internationally by ANSI and ISO. It creates open manageability standards spanning diverse emerging and traditional IT infrastructures including cloud, virtualization, network, servers, and storage. Current DMTF board of directors includes Broadcom, Cisco, Dell, Hewlett Packard Enterprise, Hitachi, HP, Intel, Lenovo, NetApp, Software AG, Vertiv, and VMware.

With this expansion of compute at the edge, enterprises, cloud providers, and CSPs need new ways of managing these platforms at scale. Network functions virtualization (NFV) brings virtualization efforts to the CSPs, and there's active high-visibility efforts in OpenStack and Kubernetes with both NFV and edge computing. However, there's a foundational piece that many in the industry are unaware of. For OpenStack and Kubernetes to operate, there needs to be an underlying hardware platform. With the scale brought about by the edge, there's a critical need to manage the hardware for all these edge platforms. This research brief by AvidThink aims to shed light on the remote management of this potentially huge collection of edge platforms. In particular, we will dig into the differences between edge platforms and today's data center platforms, challenges with remote management and the DMTF Redfish® standard. We will also focus on efforts underway to create a remote management infrastructure capable of robustly and securely supporting massive 5G, IoT, SD-WAN and edge platform rollouts. For CSPs and cloud providers looking at edge opportunities, understanding remote management challenges and Redfish's capabilities will be a critical step towards designing scalable and manageable edge deployments.

[1] "Edge Computing Market Size Worth $28.84 Billion By 2025" https://www.grandviewresearch.com/press-release/global-edge-computing-market

[2] "Worldwide Global DataSphere IoT Device and Data Forecast, 2019-2023" https://www.idc.com/getdoc.jsp?containerId=prUS45213219

The Edge is not the Data Center

To understand the massive scale that must be dealt with, we start by examining the spectrum of locations that could be classified as the edge. The one thing we know for sure is that the edge is not a data center, and its breadth and diversity, coupled with scale, is a management challenge CSPs and enterprises have not dealt with.

The Edge is Everywhere

While many have positioned the edge as a binary alternative to cloud, the reality is that the edge is a continuum that extends today's clouds. Our view is that the edge starts from regional data centers and ends at the far edge — potentially mobile cell towers or IoT gateways. For this brief, we do not consider user equipment like drones, sensors, or mobile phones as part of the edge even though terms like user edge or device edge have been bandied around. Here's a list of possible edge locations:

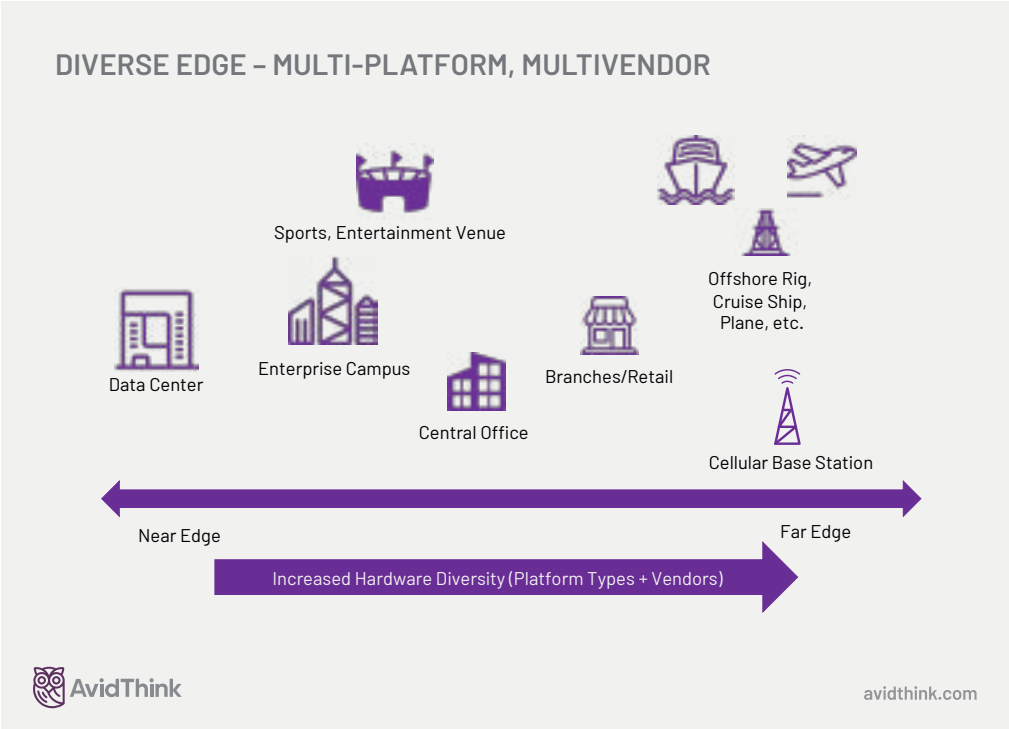
Edge platforms will be located across different geographic areas with different environmental characteristics, and these platforms will vary widely in their configuration and form factors.

- **Enterprise Campuses** — As SD-WAN services take hold, the customer premise equipment (CPE) within an enterprise becomes an edge platform to run enterprise workloads, including IoT gateways or even mobile cores for private cellular networks. Cloud providers like AWS also offer cloud software running on local infrastructure. Initiatives like AWS Outpost, AWS Greengrass, Azure IoT Edge could transform enterprise CPEs into cloud-managed edge platforms.
- **Enterprise Branches or Retail Stores** — CPEs in remote locations, especially retail locations, can host local unified communication (UC) software, IoT gateways, retail POS apps, video analytics for security, and advertising.
- **Sports, Entertainment and Other Public Venues** — Stadiums, concert halls, airports, universities, and smart buildings where the edge can offer localized services. For example, an edge application could allow stadium viewers to watch a game from numerous perspectives and offer them personalized HD content, all without backhauling massive data streams to cloud data centers.
- **Central offices and Regional Data Centers** — Both wireline and wireless CSPs have physical location assets from their existing lines of businesses. Many of these locations are aggregation points for their existing connectivity services. These locations are ideal for edge platform placement.
- **Radio Base Stations** — These represent the far edge on which components, such as the Central Units (CUs) or Distributed Units (DUs), of disaggregated 5G radio access networks could run.
- **Offshore Rigs, Cruise Ships and Airplanes** — In an age where there is universal connectivity, there may be pockets of intermittent or poor connectivity in locations such as cruise ships, planes, mines, farms, oil rigs, trains, pipelines, wind farms, solar power plants, and power grids.

While edge computing platforms are the foundation of new services across these environments, edge platforms will be located across different geographic areas with different environmental characteristics. Further, platforms will vary widely in their configuration and form factors.

The Edge is Diverse – Multi-platform, multivendor

The edge compute environment is going to unleash a wide variety of new hardware and software platforms. For these mini or micro data centers at the edge, which exist in areas that previously didn't house generalized compute servers, there are a whole host of new considerations.



There isn't really a standard edge platform — whether software (SW) or hardware (HW). Standardization of the Edge Computing/Multi-access Edge Computing (MEC) SW platform architecture is an ongoing process with standard bodies and consortiums such as the European Telecommunications Standards Institute (ETSI) MEC industry specification group, the IEEE-blessed OpenFog Consortium, the Open Edge Computing Initiative, and Kinetic Edge Alliance.

In addition, there are open source HW groups, like the Open Compute Project (OCP) and Telecom Infra Project (TIP), and open source SW foundations, such as the Linux Foundation (including LF Edge) and the OpenStack Foundation, who are using the vehicle of collaborative open source to bring about standardization to edge platforms. However, the platforms that these groups are working on are mostly software platforms and software stacks. The general recognition from these groups is that the underlying hardware platform will be quite diverse.

Since edge workloads are diverse, and because we can't predict in advance the optimal hardware configuration, there are vendors working on composable architectures. These systems look less like standard servers and more like clusters of CPUs, GPUs, FPGAs, memory and storage tied together with high-speed interconnects. These composable systems have the ability to dynamically join groups of each element type into a virtual server instance that's ideally suited to the workload.

No matter which standards or open source groups end up driving the dominant edge software or hardware platform, there are some fundamental underlying factors that drive the diversity at the edge:

- **Environmental Needs** — Most edge locations do not have data center-class power and cooling, nor can they afford space for a full-depth, full-height 19" racks. Edge platforms will come in different form factors; they must handle wider temperature ranges and operate with lower power. As such, components not usually used in data centers might be called into service. These have higher reliability under wide temperature ranges and higher mean time between failures (MTBF).
- **Performance Issues** — Edge locations generally have small power budgets but high I/O processing needs. Therefore, hardware acceleration in the form of GPUs, FPGAs, or ASICs will be part of these hardware platforms.
- **Security Considerations** — Physical security is a problem in some edge locations, and there is sensitivity to theft of platform. Edge platforms will likely have trusted platform modules (TPM) or secure enclave (SE) mechanisms that can store critical key data. Tamper-detection capabilities may also be built into some of these platforms.

As a result of these unique requirements, edge rollouts will need to include diverse hardware platforms with different configurations, and potentially dynamically reconfigurable hardware systems. Regardless, the diversity means there isn't a single vendor that can provide the variety of platforms, so the edge will be almost certainly multivendor.

Remote Management Challenges at the Edge

Managing the decentralized and distributed computing infrastructure that powers the edge will be a significant issue. As orchestration software like OpenStack and Kubernetes are enhanced to accommodate new edge deployments, the underlying hardware platform will need an equivalent distributed management system.

The challenge for remote platform management at the edge is a little different from that of data centers. Certainly, all the standard criteria for remote management of data center infrastructure will exist at the edge, but there will be additional requirements:

- **Multivendor and diverse platforms** — We've already discussed this, but it's important to reiterate. Any solution to manage platforms at the edge will have to accommodate varied hardware that includes CPUs, GPUs, FPGAs, smart NICs, and different memory and storage devices. And with the concepts of software-defined infrastructure taking hold and composable hardware platforms showing up, we could see reconfigurable edge platforms emerge with CPU, GPU, FPGA, and storage modules. In any case, the remote management solution will need to work well across diverse vendors, each with unique capabilities on their platforms.
- **Massive number of distributed sites** — The data center environment is generally one in which there's a high-speed management local area network that ties into the management interfaces of 1000s or 10,000s of servers in a single location. These servers are often from a limited number of vendors and of limited flavors. Looking at a major cloud provider like AWS, which is unique in that it makes its own custom servers, it currently has five major categories of instance types, each with only four to eight instance families. At the edge, the variation in HW platforms will easily number in the hundreds or more. And instead of 1000s of servers in a few locations, we'll have a few servers in 1000s or more locations, under a single administrative domain.
- **Varied connectivity characteristics** — As discussed, there's usually a reliable high-speed management network present in data centers. At the edge, connectivity will be more limited, and will vary from 1Gbps links to satellite links in the multi-Mbps or even multi-100Kbps range. In some cases, these links will be intermittent and may experience periods of downtime.
- **Emphasis on security** — We have discussed the need for a more secure platform at the edge due to the remote locations. In addition to supporting the underlying hardware encryption and tamper-detection and logging capabilities, any remote management solution needs to provide robust encryption, role-based controls, authentication, and resilience against DoS and DDoS attacks. Techniques like a root of trust embedded within the platform, such as Google's OpenTitan project or HPE's silicon root of trust, could become commonplace. Fundamentally, without the ability to rely on a segmented data center network protected by a firewall, the remote solution must engender security better than legacy IPMI.

- **Increased need for robustness** — While data centers might be hard to get to, they are generally located so that a qualified engineer can get onsite in a reasonable time. Plus, with scale economies, it makes sense to install a robust rescue or backup path into the data center (via LTE links or via a completely separate and distinct backup network). This is cost prohibitive for remote edge sites. Worse still, a "truck roll" to reach a remote site might involve sending a helicopter with an engineer onto an oil rig, or at the very least, an expensive platform swap by a less-trained onsite technician or non-technical employee. Therefore, whatever remote management solution is deployed, it has to be near bulletproof when it comes to configuration changes, firmware updates, and platform resets.

Adding to the above, the edge platforms will contain a mix of bare-metal deployments, likely with a container-based SW platform, as well as VM-based platforms with hypervisors. This last factor is not unique to the edge, since centralized data centers face the same issues. However, this requirement coupled with the unique characteristics at the edge creates additional management complexity.

Today's Remote Management Reality – the BMC and IPMI

Back in the mid-to-late '90s, as large deployments of servers in data centers became a reality, CPU and server manufacturers recognized they needed a way to monitor and manage the hardware platform on these servers. To achieve this, Intel developed and published the Intelligent Platform Management Interface (IPMI) specification in 1998 along with a design for the Baseboard Management Controller (BMC) module, the main controller within an IPMI subsystem.

The IPMI describes an independent always-on, always-available subsystem that provides management and monitoring capabilities independently of the host system's CPU, firmware, and operating system. IPMI defines a set of interfaces used for out-of-band management and monitoring of servers.

The BMC acts as a rescue platform to reset the server when the system OS is not functioning and essentially force a server restore, all without human direct physical access to the server. Instead of having to physically visit a data center, plug-in a keyboard, display, and mouse or power on or off a machine, data center operators can do so remotely via IPMI. BMCs can remotely monitor and manage large clusters of servers with the Remote Management Control Protocol (RMCP), a specialized wire protocol defined by this specification.

Inadequacies of IPMI

Today, 20-plus years later, IPMI is at version 2.0 and supported by just about every major server manufacturer in the industry, albeit with varying capabilities. Some server manufacturers have their own proprietary implementation that provides more advanced capabilities.

However, IPMI itself is a dated protocol; it isn't API-friendly and therefore not automation-friendly. It is relatively archaic and hard for developers to develop with. Most importantly, it has been demonstrated to be insecure, from a **authentication protocol hack** to a **module weakness**, and multiple public exploits have been announced in the last 7-8 years. In many cases, internet-facing BMCs were exposed, leaving tens of thousands of servers vulnerable to being directly attacked at the firmware level regardless of how robust and secure the operating systems and applications running on top were.

Other protocols, like SNMP, CIM, and WMI also exist to communicate with the BMC, but they are constrained by the IPMI specifications and are also legacy protocols not designed to be API-friendly either.

The Need for Standards-Based Edge Management

As we move to the edge, what we have today with IPMI will not work. Further, given the edge will represent a collection of numerous platforms from diverse vendors, a proprietary management approach will not fit the bill. There needs to be a standards-based infrastructure management framework that all major edge vendors participate in, driving enough momentum to incentivize all edge platform makers to adopt.

Redfish to the Rescue

IPMI is baked into server platforms the world over and migrating from IPMI is no easy task. Nevertheless, over the past few years, with the support of major server and server chip manufacturers, as well as open source organizations, there has been momentum in DMTF's Redfish project. In particular, the OCP's incorporation of Redfish into its hardware management projects gave it a boost for use in web-scale data centers. This gives OCP hardware a valuable out-of-band (OOB) mechanism while providing a shot in the arm for Redfish momentum.

What is Redfish?

Redfish is a standard that defines simple, secure infrastructure management for data centers. It aims to provide an easy-to-implement standard for access to the same types of subsystem data that IPMI provides. However, it leverages common web technologies to make it fit better into today's programmatic toolchain and automation management frameworks.

Version 1.0 of the Redfish specification passed in August 2015 and there's been significant momentum since. We are now at version 1.8 of the specification. Redfish is designed to be an open, unified API that addresses the shortcomings of IPMI and naturally fits into modern DevOps frameworks.

As part of its design tenet, Redfish leverages existing web standards and best practices like OData from OASIS, an international standards organization, which defines how to query and update data over RESTful interfaces. And Redfish separates protocol from the data model, allowing each to evolve or change independently. From a protocol perspective, Redfish uses HTTP and TLS (standard web-based transports), SSDP (Simple Service Discovery Protocol) from uPnP (universal Plug and Play) for service discovery and HTTP-based alert subscription.

Leveraging JSON and RESTful interfaces, Redfish benefits from the wide variety of tools that already exist for web developers to test, interact with, and manage large-scale web-based applications. Through this RESTful interface, Redfish provides the ability to retrieve IPMI-type data, including things like:

Leveraging JSON and RESTful interfaces, **Redfish benefits from the wide variety of tools that already exist for web developers to test, interact with, and manage large-scale web-based applications.**

- Server identification and asset tag information, as well as system configuration, NIC MAC addresses, and topology.
- The health of the server, including temperature, fan speed, power supply status, power consumption, and hard drive status.

Redfish can also perform the operational functions IPMI used to provide, including:

- Turning power on and off.
- Changing the boot order and boot devices.
- Accessing the serial console via SSH.
- Handling event notifications and logging.
- Managing BMC network settings and local user accounts.

The unique attributes of Redfish and its ability to take on all the functions IPMI used to provide make it appropriate as a secure, modern, multinode capable replacement for IPMI.

Redfish Security Tenets

As for security, Redfish includes support for two authentication methods: HTTP(S) basic authentication (basic auth) and

session-based authentication. Basic auth is used generally for one-shot commands, where the username and password are passed as HTTP(S) headers in the actual request. Session-based authentication requires obtaining a token first and then reusing that session token in all subsequent requests, with the need to delete that session to free up session slots at the end. The latter is used for longer-lived sessions. Further, in addition to local users, corporate directories like LDAP and Microsoft Active Directory can be used for credentials. Most implementations will protect the majority of resources behind authentication, making it harder to attack. And use of standard HTTP(S) protocols allows the use of HTTP(S) proxies and NGFW inspection on the traffic to protect these resources.

Benefits of Redfish

By using a format common to web developers today (JSON), and leveraging the well-understood HTTP(S) protocol, Redfish benefits from a common web security model, and from development tools that operate on JSON-formatted RESTful interfaces. The human-readability also helps developers get up to speed quickly, allowing Redfish commands to be tested and executed from browser plugins designed to exercise web-based APIs.

The reduced complexity of this approach, when compared to IPMI, allows a wider ecosystem to interact and integrate with the Redfish protocol and speeds adoption. Now, infrastructure management can be performed using the same development tool chains and skill set as other DevOps tasks today.

Certainly, the biggest benefit to Redfish is its potential ubiquity. As a non-vendor-specific, standards-based and already well-adopted standard, it provides a rallying point for all server and device manufacturers to congregate around to ensure a common management infrastructure for the next-generation of deployments in data centers and at the edge.

Having said that, just as different vendors had varied implementations for IPMI, particularly in areas like graphical console access or console over LAN, we'll see some variations in Redfish implementations, particularly around maturity of support for all the operations. Nevertheless, AvidThink expects Redfish to take over from IPMI, especially for edge computing deployments.

Current Status of Redfish

As indicated, Redfish is at version 1.8 today. And some of the major server manufacturers already have strong support for the standard. Cisco, Dell, HPE, and Lenovo all support it, as do Intel and Supermicro motherboards. At the same time, manufacturers will still have their own proprietary management systems. Those such as Cisco IMC, Dell IDRAC, HPE iLO and Lenovo XCC will continue to exist, but as Redfish matures, some may choose strategically to take an open, standards-oriented path and go all in with Redfish. In fact, Redfish allows for Redfish-compliant vendor extensions that conform to the overall Redfish specification. For example, in addition to the basic functions, HPE iLO includes extensions to provide for UEFI system configuration and SMART storage status among others.

The DMTF is trying to make adoption easier by providing simulated environments where users can familiarize themselves online or use simulation libraries to test out their automation harnesses. For example, there's a **Mockup Utility** that can collect all Redfish resources and schema, and a **Python CLI Redfish** tool to demonstrate resource-oriented basic operations. Finally, DMTF provides the aptly-named **Redfish Tacklebox** that collects together a set of utilities to perform common management tasks like power on/reset, checking system inventory, and updating firmware. There are also vendor resources that provide scripts, tools that work within Ansible, Chef, Puppet, and other popular automation frameworks for using Redfish to manage their equipment. Another related project is Linux Foundation's **OpenBMC project**, which provides an open source Linux distribution for equipment vendors to kickstart their BMC implementation.

Redfish for the Edge — Closing the Gap

As Redfish matures, just like any other management standard, such as IPMI or SNMP in the past, there's the problem of dealing with bloat. The danger of accommodating all suggestions and contributions from the users is that Redfish ends up

with a kitchen sink situation. Especially with the rise of edge computing, where hardware variations will be much larger, trying to accommodate all that diversity would be tricky.

Some in the industry are pushing for Redfish to be more modular, by focusing on a core set that every vendor will adhere to and then keep all other families of functions as modular — whether function specific, e.g., only relevant for systems with GPUs, FPGAs etc., or vendor specific.

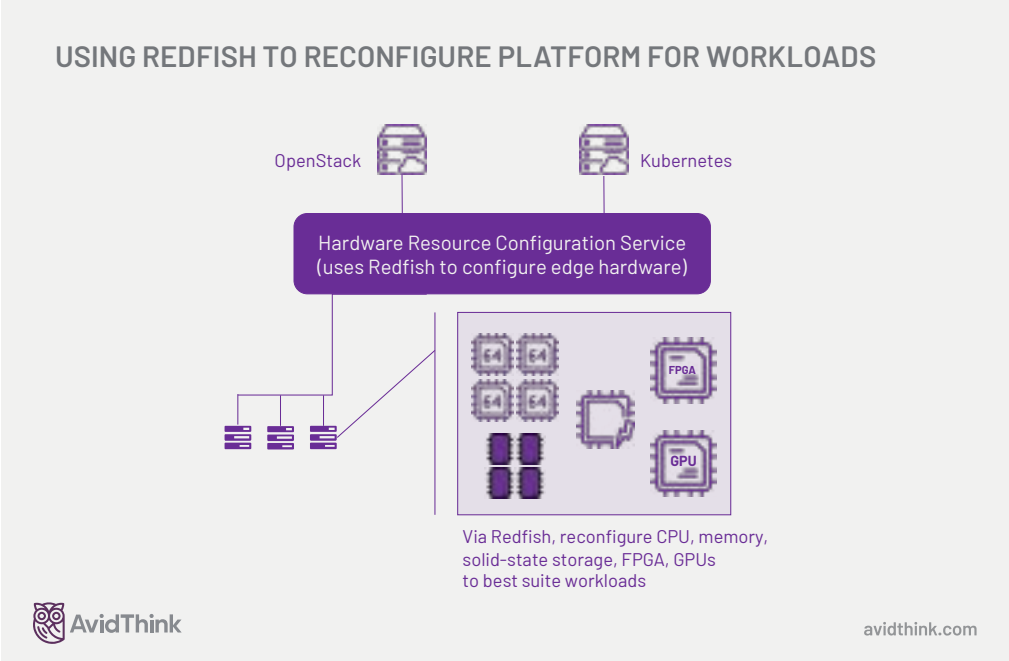
AvidThink anticipates the evolution to higher-level orchestration systems that will also manage the underlying hardware configuration (either directly or via an intermediary resource configuration service).

Integrating with Management and Orchestration Stacks

With DevOps automation and the use of orchestration and management platforms like Kubernetes, OpenStack, and VMware's vCloud as well as other commercial equivalents, it stands to reason that we would want to integrate the underlying hardware management system with the software orchestration system.

Ideally, we would have a two-way flow of information between both systems allowing, for example, OpenStack or Kubernetes to first provision the underlying hardware — say, associating appropriate storage elements to compute elements or carving out GPU resources — and then load the appropriate OS and workloads on top. Some of the bare metal provisioning projects within the OpenStack framework, like OpenStack Ironic, have support for Redfish. And with open source projects like OpenStack's Metal3, the same framework can be used for bare metal container infrastructure with Kubernetes.

AvidThink anticipates the evolution to higher-level orchestration systems that will also manage the underlying hardware configuration (either directly or via an intermediary resource configuration service). Many proprietary and commercial orchestration solutions will expand to all layers of an edge data center: from the underlying hardware to the orchestration systems to managing and monitoring the applications.

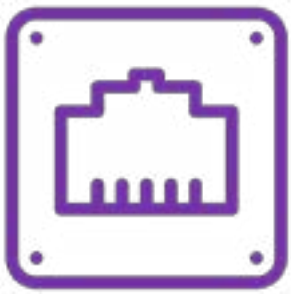


Addressing Non-compute Components

Aside from Redfish for compute, there's a related initiative called Swordfish™ from the Storage Networking Industry Association (SNIA). The SNIA is a storage industry trade organization that develops and promotes vendor-neutral architectures, standards, and educational services that facilitate the management, movement, and security of information.

Swordfish piggybacks on the DMTF Redfish specification, borrowing the same elements of a RESTful interface with JSON. It is designed to manage storage elements in cloud and data center environments. The list of member companies includes vendors already involved in Redfish, plus storage companies: AMD, ARM, Cisco, Dell, Fujitsu, Hitachi, HPE, Intel, Lenovo, NetApp, Supermicro, and other corporations.

On the networking side unfortunately, we come up a little short. With the SDN and Network Automation movement, we've had controller-based network controls and a wave of intent-based network configuration initiatives but little to show for it in the way of standardization beyond the OpenFlow and P4 protocols, which are more focused on changing the behavior of networks. The closest to Redfish and Swordfish is the OpenConfig movement in networking originally supported by Google and now by vendors like Cisco and Juniper. OpenConfig's goal is to create vendor-neutral model-based network configuration written in YANG, a data modeling language. However, OpenConfig is focused on a layer of operation slightly higher than the networking hardware itself, not only will it handle inventory and telemetry but also deal with higher-level abstractions like routing tables and routing protocols.



There are certainly higher-level orchestration systems like Linux Foundation's ONAP and ETSI's OSM, those tend to operate at a higher-level of abstraction today but there are efforts (e.g. ONAP Composable Disaggregated Infrastructure support) to integrate these with Redfish to provision the underlying hardware to serve virtual network functions. However, configuration and management of physical network functions are currently not within scope for these projects.

Hierarchical Scaling for the Edge

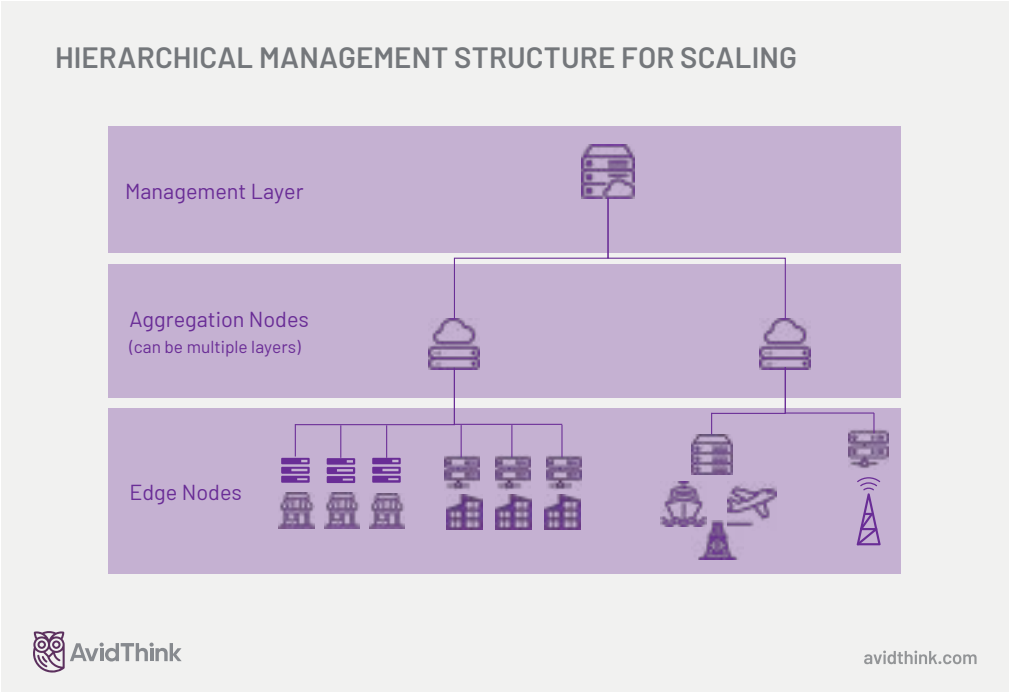
One of the issues around Redfish — and the same charges could be leveled against IPMI — is the need to manage at scale. Even if we dealt with the challenges described above, there's a need for a robust framework for managing thousands or tens of thousands of edge compute platforms. To get to the scale needed, we need to create a layer of abstraction through hierarchy and aggregation as part of the management system.

In the past, especially with IPMI, managing servers was a relatively flat operation. A single control point would reach out to thousands of servers in a data center and issue commands, usually dealing with one at a time or forking some number of threads to parallelize the efforts. Since each of the servers would have relatively static hardware configurations, and because the number of server types were limited, these management operations didn't have to be particularly sophisticated.

To manage the edge, with vast geographical distribution, varied server types, and where configurable platforms in the form of composable platforms might be deployed, a more sophisticated approach needs to emerge.

Learning from past approaches for managing large scale infrastructure, Redfish management systems will have to incorporate hierarchy, adding aggregation nodes that will act as delegates and proxies as part of the overall infrastructure. Similar ideas exist in other management systems which use concepts of domains and forests, such as Microsoft's Active Directory and its categories of domain controller roles. Some early open source efforts like **RedDrum** tried to provide aggregation points on a per-server-rack basis, but these efforts are better suited for large data centers with lots of server in a single-rack. Leading server vendors are today working on utilizing Redfish as a foundation and building on top of that with a hierarchical management infrastructure that better fits the edge.

A general form of this type of hierarchical management would look like the following:



As depicted above, the design involves an aggregation layer that allows the upper management layer to control large numbers of southbound devices by sending a unified set of Redfish instructions to these aggregation nodes, which take on the task to communicate with and configure each edge device under their purview. Likewise, these aggregation nodes would gather appropriate monitoring and telemetry from southbound nodes and send them to the upper layers, perhaps with some level of filtering or pre-processing.

One could imagine having multiple layers of proxies as needed to handle larger-scale deployments. Each aggregation node can manage disconnected end devices and wait for them to come back online to configure them or update their status, without tying up central control systems upstream. Some of the complexity of southbound devices could also be abstracted and managed at the aggregation layers. These aggregation nodes could be organized geographically, or even by device types. Aggregation by device types, e.g., all IoT gateways from a single vendor for a specific use case, could substantially reduce the complexity of sending different configuration commands to different clusters.

There's plenty of innovation that will be needed for edge hardware platform management and lots of opportunity to be captured. Vendors and standards bodies will likely drive innovations in this space for the next few years as the edge comes into focus for cloud providers, enterprise, and CSPs.

Implications for CSPs and Edge Providers

For CSPs and edge cloud providers, it's important to realize that in addition to investment in orchestration and management platforms based on OpenStack and Kubernetes, attention needs to be paid to underlying remote hardware platform management. Redfish is likely the best standards-based candidate for that platform management, but there's still significant effort before it's ready for the scale and diversity of the edge. Furthermore, vendor support for Redfish can be uneven, and there's the reality that proprietary hardware management systems will continue to exist.

There's plenty of innovation that will be needed for edge hardware platform management and lots of opportunity to be captured.

AvidThink recommends that CSPs and edge cloud providers design orchestration and automation strategies that accommodate different and diverse hardware management systems from multiple vendors, while putting pressure on these vendors to improve their support for Redfish. Likewise, CSPs, edge cloud providers and vendors should collaborate to evolve Redfish to accommodate the management scale needed at the edge.

The Future of Redfish at the Edge

The edge computing market promises to be a sizable blue ocean opportunity for all. As we roll out edge use cases while dreaming up unique applications, we need to realize that these use cases and the type of deployments have implications for the underlying platform. Whether software or hardware, the edge has different needs from that of the data center. Redfish is a critical part of ensuring that we have a manageable, scalable, and extensible hardware platform management solution. Closing the gap between the requirements of the edge and the rapidly maturing Redfish standard will bring much needed innovation in this space.

AvidThink, LLC
1900 Camden Ave
San Jose, California 95124 USA
avidthink.com

Rev A