

# High Performance Computing as a Self-Service HPCaaS™

Self Service Portal for HPC Infrastructure  
Lifecycle Management, Provisioning and Orchestration of  
Layer 0(PCle) through Layer 7

May 2019



## Introduction

High Performance Compute Centers (HPCC) are expanding throughout major industries as the need for the computational power across larger data sets and workflows grows exponentially. Lower barriers of entry and the need for higher performing centers are accounting for the bottle neck throttling the adoption of workflows in these HPCs. With the advent of technology breakthroughs in the interconnect backplane technology based on industry standard PCIe interface, to solve latency and bandwidth limits, and Lab as a Service automation frameworks, HPC centers can solve both of these issues. HPCCs with a focus on “as a Self Service” portals to schedule, reserve and configure new faster, and greener network infrastructure will allow more users to adopt and consume HPCC resources and deliver solutions to high compute and big data problems at a new pace.

The key to the successful deployment of High Performance Computing as a Self-Service (HPCaaS) is based on Lab as a Service frameworks combined with automation tools to provide a wide range of provisioning and orchestration automation linking the workflow to the infrastructure. The ability to disaggregate and aggregate the network, compute, memory and storage items (ie Layer 0 interconnect) into unique stacks for specific workflows while being managed as a life cycle sandbox will allow users to more efficiently consume the HPCC resources.

This white paper will address the problems facing HPCCs, solutions to the problems, risks and an example solution using the latest available set of tools and methodologies to accomplish the goal of delivering HPCaaS.

## High Performance Computing Center Usage Barriers

HPCC environments encompass a number of different types that share some common characteristics, but also have their own unique requirements:

### **HPC: A Changing and Challenging Data Driven Environment**

The remarkable increase in the amount of data being collected, and that must be analyzed and stored, is driving the rapid adoption of advanced data analytics and Artificial Intelligence (AI) that is challenging the fundamental architectures of today’s ASC data centers, in a way not seen since the 1990s.

AI, and the associated Machine Learning (ML) and Deep Learning (DL) applications are fueling demand for fundamental change in the creation of compute and storage clusters. Faster and larger storage arrays and a rapid proliferation of specialized



compute accelerators, like GPUs, FPGAs and custom ASICs, are creating bottlenecks and configuration problems for the interconnect systems, as the traditional networks were never designed to handle the performance requirements of these workloads and devices.

Further, the rapid pace of change in acceleration technology and AI software fuels the necessity for flexible and easy to upgrade architectures, capable of incorporating new technology without demanding forklift upgrades to expensive equipment. This means disaggregating elements of the traditional server into separate pieces that can be easily shared. But, in order to effectively disaggregate storage and accelerators, the interconnects must support both an exceptionally low latency AND high bandwidth.

Owners and administrators are being pressure by end users to lower barriers of adoption and in turn automate the complex administration activities within the HPPC. Admin support for open source tool automation, delivering easy to use self-service portals and trying to perform re-use and standardization of automation and tools becomes an extremely heavy and costly lift.

And, of course, data center managers want to drive high utilization of expensive new storage and acceleration products to keep both Capex and Opex costs down.

Add all these up and these Advanced Scale Computing, Enterprise, Cloud and Edge data centers need both scale-up and then scale-out resources across the cluster and require a network technology that will grow in both directions.

Two potential solutions both raise new issues. Scaling-up computing systems into ever-larger “super-servers” is generally cost prohibitive for all but the largest academic or government institutions. Scaling-out computer power by replicating servers introduces overprovisioning and underutilization inefficiencies.

A few IT technology industry efforts are aimed at overcoming these issues:

- Numerous companies—from established to startup—are developing a new wave of more efficient, niche accelerators. Rather than general purpose GPUs, these chips work faster and use less power because they are designed for very specific processing tasks.
- SSD storage providers are rapidly improving performance.
- Advanced Scale Computing (ASC) is disaggregating resources into a *composable infrastructure* where compute, storage and accelerator resources can be accessed on-demand.

But, while disaggregating resources is the right solution, now the system interconnect that enables composability becomes critically important. Insufficient speed and high latency in the interconnect networks, creates a bottleneck in scaling-up and scaling-out with ASC. The underlying question for customers then becomes: How can I best manage all these new compute, storage, and network resources to break the constraints of prepacked servers and scale my computing to meet changing requirements, while getting the best bang for my computing buck?



## Addressing the Issues – Solutions for a High-Performance Compute Center Service

### PCIe Based Interconnect Network

Starting with a fundamentally different architecture, a PCIe *interconnect network* eliminates the bottleneck. This architecture can construct a multi-rack, composable infrastructure that performs as if every storage and compute element were inside a single box, providing many benefits:

- Native PCI Express (PCIe) communication between hundreds of mixed processing units, cutting latency by eliminating the entire translation layer required by other interconnect network options.
- True Direct Memory Access and sharing of *all* connected processors and memory with point-to-point connections between any two devices, reducing customer investment in these resources.
- Dynamically disaggregation and composable, scalable, and elastic infrastructure.
- A roadmap for end-to-end data speed improvements across the entire interconnect network.
- Legacy support for HPC resources

### Self Service Portal for HPCaaS

Many entities today are struggling with how to efficiently manage their HPC infrastructure with the continuous pressure to reduce cost while increasing performance, capacity, and ease of use for their consumers. These entities have identified the following infrastructure management needs:

- Support of new technologies
  - New Interconnect for networking based on PCIe resources
- Scheduling, reserving, managing, deploying (lifecycle) environments
- Auto provisioning of resources (physical and virtual)
- Supporting converged infrastructure and legacy hardware
  - Support of cross domains (public and private clouds)
- Automation Framework to control provisioning from PCIe thru Layer 7
- IT admin activities
  - Auto-discovery, lab resets, resource health-checks
  - Powering down devices when not in use
  - Support for maintenance modes on resources
  - Integrating Help Desk applications into the workflow of administration of the HPC
  - Spinning up new resources on demand, etc.



- Adding workload provisioning on top of the compute stack optimization of the provisioning and orchestration specific to the stack and the workload
- Supporting multiple tenancies and domains
- Configuration management of HPC resources
- Enabling user automation (configuring and deploying workloads, Virtual machines, containers, DevOps flows, sandboxing, etc.)
- Sharing of intellectual property
  - Support processes, automation, configurations, resources, use cases, etc.
  - Community based and open source focused
- Integration with other tools such Help Desk, Ticketing systems, and Simulators
- Metrics on the processes/activities, resources, usage, and users, to manage the lifecycle of the environment

This list is by no means complete but it does address the majority of the problems seen by these entities. All of the above actions need to be handled in a standardized and centralized approach in order to be effective and consumable by the different roles involving the use of the HPCC infrastructure. The actions need to be managed with a lifecycle approach for both the management of the HPC infrastructure and the activities within the HPCC. The maintenance of resources, the roll back of configurations, and the validation testing of an environment before releasing it to the consumers are all examples of actions that are repeatable yet highly configurable and complex.

Management needs to understand how well the HPCC infrastructure is functioning so that decisions about the maintenance and lifecycle of the range infrastructure can be made from the data analytics available. The infrastructure needs to have tools in place to support not only these actions, but the lifecycle management of these actions as well. Most importantly, the environment needs to support new tools and processes as well as share intellectual property (IP) developed across all levels and users of the environment (administrators, developers, end users, etc.).

## The Risk of Not Implementing HPCaaS

Given the financial cost of a HPCC and its importance to the rest of the organization, it doesn't pay to make the significant capital and operational investments while neglecting the lifecycle management of the infrastructure. The highly manual processes associated with lifecycle management typically used in large Infrastructure labs are the enemy of reliability, repeatability, and auditability. Manual or non-managed processes are often visible in a number of ways:



- *Absence of live inventory visibility.* In most HPCCs, equipment inventory is not tracked in a way that provides live visibility to engineers. While most IT organizations perform asset tracking for financial purposes, what passes for the inventory management used by engineers is a spreadsheet that is often ill-maintained. As a result, it can be difficult to tell without exhaustive work what resources or equipment exists, is being used by whom and what is truly available.
- *Offline topology design.* Since there is no usable inventory visibility, it follows that topology design is done completely offline without regard for resource availability. Visio or other diagramming tools are most common, and basically produce the electronic version of a paper drawing, which is usually then printed to aid in a time-consuming manual hunt for relevant equipment.
- *Chaotic connectivity management and costly errors.* Once inventory is found that is at apparently available, engineers must manually re-cable connections between the equipment. With multiple engineers making adds, moves and changes, typically without up to date documentation, errors such as disconnecting someone else's infrastructure inevitably occur.
- *Underutilized resources wasted resources.* Often servers, memory, storage and networking resources are locked into a particular set of virtualization tools or stacks, greatly limiting the sharing and balancing of these resources for different workloads.
- *Use of custom environments instead of COTS (Commercial Off The Shelf) tools.* Using these custom tools to manage the environment leads to excessive cost, limited use within different groups, and limited industry expertise to lower maintenance costs for the management environment. Different custom tools and environments without a common control interface across different user roles can cause limits in adoption of the environment due to increased training and expertise needed to support multiple management environment tools and processes.
- *Lack of device/stack configuration baselining.* Engineers using the infrastructure must often change OS images, apply patches, and create new configurations on devices. Unfortunately, it's all too easy to forget to set devices or environments back to a baseline state, which means that when the next engineer uses the resource, they may wrongly assume that it is configured at a known baseline state and execute a series of test protocols on an incorrect configuration.
- *Legacy resource support.* HPC leaders must constantly be aware of obsolescence support when adopting new HPC resources to ensure compatible and extend the life cycle of very costly resources.

The result of these manual processes is inaccuracy, inefficiency, and waste, evident through a number of indicators:



- *Lack of process integrity and repeatability.* Manual processes tend to experience operator errors that compromise process integrity. The lack of repeatability that results means that it is very hard to offer sufficient verification of processes.
- *Poor process documentation.* Manual processes are by nature difficult and time-consuming to capture in documentation for auditing purposes. When changes occur in procedures or processes, it is too easy to miss documentation steps, which can impact the audit trail.
- *Incomplete process reporting.* Process methodologies can generate voluminous results of data. Manual analysis processes struggle to digest this data and provide sufficient reporting for auditing purposes.
- *Imbalanced ratio of setup to actual usage.* Infrastructure engineers can easily spend days in the setup process for a procedure that takes less than an hour to run.
- *Very low asset utilization.* Millions of dollars in capital equipment are typically only 15% to 20% utilized. This represents a huge waste of annual capital depreciation costs.

There are significant implications of the inaccuracy and waste created by manual operating processes in HPC labs:

- *Risk of errors and non-optimized HPC infrastructure due to process integrity, repeatability, and documentation issues.* Even if the processes are painstakingly performed in an accurate fashion most of the time, the inefficiency and slow pace of manual or separate custom processes may make it nearly impossible for allocated personnel to achieve fast HPC infrastructure delivery, which causes a reduced utilization of the infrastructure.
- *HPC lab asset utilization under 20% represents a significant waste of capital depreciation costs.* Low asset utilization also means that as demands for infrastructure deployment grows, the pace of investment in infrastructure lab capacity will rise at a rapid rate. With large infrastructures costing anywhere from \$1K to \$3K per square foot inclusive of equipment costs, this can lead to huge, unnecessary CAPEX outlays over time.

## HPCaaS Implementation

### **GigaIO FabreX™ Interconnect Network**

Starting with a fundamentally different architecture, GigaIO FabreX *interconnect network* eliminates the bottleneck. This architecture can construct a multi-rack, composable infrastructure that performs as if every storage and compute element were inside a single box, providing many benefits:



FabreX provides wider lanes and faster data throughput (256Gbps full duplex). FabreX is administered using DMTF open-source Redfish® that provides a simple interface for configuring computing clusters on-the-fly.

FabreX is 100% compliant with the industry's leading standard, PCI Express, insuring high levels of support and compatibility from an enormous selection of vendors and technologies. Every new storage, acceleration and compute technology fully support the latest PCIe standards ensuring full access to the latest and most capable technologies.

And as IT experts begin to use new architectures for moving compute to the data at the edge of your networks, FabreX is a natural choice, offering the greatest depth of support in storage, compute and acceleration technology, the highest density, lowest power and best performance.

### **TSI Middleware Automation Framework for HPCaaS**

Using a HPCaaS solution to manage the lifecycle of a HPC infrastructure environments can help users achieve dramatically higher accuracy, utilization, and productivity. This will lead to significant CAPEX and OPEX savings, faster infrastructure cycle completion, and sustainably documented processes and reporting for metrics and auditing of the HPC Infrastructure's performance. A sound automation and provisioning solution which delivers a fully integrated, object-oriented software framework for automating development, administration and end user operations on the HPC infrastructure controlling from PCIe(Layer 0) through Layer 7 orchestration and provisioning includes:

- Centralized live infrastructure and resource inventory
- Inventory-aware stack/topology design
- Shared calendar based resource and topology reservation
- Connectivity mapping and automated connectivity control
- Easy to create automated provisioning tasks
- Non-programmer friendly automation workflow creation based on a library of highly reusable, template based, objects that can be created from a wide variety of sources and leveraged to create:
  - Auto-discovery, auto base-lining, and other automated maintenance routines
  - Full test automation workflows
- Community sharable IP for automation and management of HPC infrastructures and processes
- Powerful automated reporting that provides a verifiable and sustainable audit trail
- Resource agnostic (any device, any cloud, any hypervisor) to support new technologies
- Ownership of all logs and datasets produced by the toolset



If designed properly, the HPCaaS architecture avoids the pitfalls of script-based approaches to automation, which cannot scale due to their high maintenance costs. Best of breed commercial solutions deployed by industry leading organizations worldwide provide them with the fastest path to successfully and sustainably automated framework environments. This is the path which leading power utilities, enterprises, government and military agencies, telecom service providers, and technology manufacturers have chosen to transform chaotic manually driven environments into highly efficient infrastructure operations. These organizations have the ability to:

- Manage infrastructure inventory including Servers, GPUs/Accelerators, Flash/Storage, L1/L2/L3 switches, and virtual resources such as virtual machines, virtual switches, and containers in a live, searchable database of resource objects tagged with searchable attributes. This capability eliminates manual searching for equipment in racks, and allows engineers to interface with the datacenter infrastructure efficiently via software. An inventory and resource management tool with object support and hierarchies can represent relatively simple nested resources such as chassis, blades, and ports, or complex, pre-integrated resources stacks such as converged infrastructure and workflow based, stack solutions. (See Figure 1 Web Portal and See Figure 2 Resource Library and Search)
- Create variable or abstract topologies via a software GUI that allows drag and drop of resource objects onto a canvas, visually ascertain availability, design, and sanity check connectivity. Save the entire topology as a high-level object in the resource library, so that it can be reused later or by other engineers.
- Schedule resources and entire topologies through a common calendaring system, preventing use case disruptions. (See Figure 3 Scheduling)
- Manage connectivity remotely by generating patching or cabling requests to lab administrators, or if PCIe/Layer 1 switches are in use, to automatically connect and build topologies.
- Make device and service provisioning error free by building automation objects for common provisioning tasks, and execute them from a graphical test topology view. Device/VM provisioning can include uploading OS images, resetting device configurations to baselines, or creating routing adjacencies between virtual switches. (See Figure 4 Resource Automation Commands)
- Create auto-discovery and auto-baselining processes that leverage an extensive array of control interfaces, GUI automation and scripting capabilities to streamline the management of inventory and device states to a compliant baseline.
- Automate complex provisioning and configuration management tasks in a fully documentable and repeatable fashion. Automation can be created through integration of existing automation scripts as objects, as well as creation of new automation objects through screen, GUI, and other capture processes.



- Support for including open source tools, APIs and existing scripts into the environment and managing them as objects to facilitate reuse, centralization and standardization of IP.
- Build, configure, and rapidly deploy virtual networked environments through an easy to use, multi-tenancy web portal GUI. Embed other tools to allow integrated workflow applications and other tools into a single pane of glass environment for easy to consume interfaces for end users. (See Figure 5 Complex HPC Stack Environment)
- Use compute, storage, and networking resources in an optimal and hyper-converged deployment for shared resource pools to meet on-demand needs of users.
- Support Rack Scale Design (RSD) for aggregation/disaggregation of HPC resources.
- Support virtual resources from any hypervisor and any physical resource vendor, allowing for a universal user interface for all consumers of the infrastructure. (See Figure 6 App and Service Support for Clouds)
- Allow multi-tenancy in a secure virtual and physical environment to meet security compliance validation. (Encryption of passwords, SSO, single file virtual network containers)
- Generate comprehensive audit compliant result reports.
- Produce custom business intelligence dashboards to allow for managers to analyze and collate data from the testing activities and metrics for input into planning initiatives. (See Figure 7 BI dashboard)

## The HPCaaS's Beneficial Impact on the HPCC Infrastructure

Adoption and deployment of a HPCaaS methodology supporting FabricX interconnect technology on your HPCC leads to significant, positive impacts:

- ***Sustainable auditability.*** With automation comes built-in documentation of automation processes since the object-oriented method of creating, modifying, and maintaining template elements creates an ongoing and live documentation for process composition and methodology. Automated equipment maintenance processes with documented schedules provide proof of the compliance of the testing environment. Automated results analysis offers robust reporting that offer solid proof of compliance and compliance efforts. Complete control of all data sets produced by the framework allows for control and ownership of all metrics and outputs produced by the toolset.
- ***A dramatic increase in the velocity of infrastructure delivery.*** Organizations routinely report time savings upwards of 70% in their deployment cycles once they have



automated the process of allocating devices, device/VM provisioning, running automation processes, and generating reports.

- **Performance**
  - FabreX delivers the industry's lowest latency AND the highest effective bandwidth. Latency from system memory of one server to system memory of any other is less than 200ns – true PCIe performance across the entire cluster. The current Gen 3 implementation delivers 256Gbits/sec bandwidth, soon to scale up to 512Gbits/sec with PCIe Gen 4.
- **Flexibility**
  - FabreX can unite an unprecedented variety of resources, connecting accelerators of all types including GPUs, TPUs, FPGAs and SoCs to other compute elements or storage devices, such as NVMe, PCIe native storage, and other I/O resources. FabreX can span multiple servers and multiple racks to scale up single-host systems and scale out multi-host systems, all unified via the FabreX software.
- **Efficiency**
  - Featuring 100% PCI-SIG compliance, the FabreX switch can integrate heterogenous computing, storage and accelerators into one symmetrical system-area cluster fabric, so you can do more with less. Patented GigaIO technology strips away unnecessary conversion, software layers and overheads that add latency to legacy interconnects.
- **Significant savings in infrastructure CAPEX and OPEX.** Organizations deploying infrastructure automation software report increases of 50% to 200% in device utilization, leading to capital budget savings, less depreciation waste, as well as accompanying savings in space, power, and cooling costs.
  - **Open Platform, Standards-Based**
    - FabreX is built on, and 100% compliant with, the industry's most widely adopted standard, PCI Express, insuring low risk, easy integration and long life. Further, the FabreX operating system easily integrates third-party applications with its open-source design, including the DMTF open-source Redfish® APIs to provide unprecedented integration with a range of third-party applications for fabric automation, orchestration, resource allocation and job management.
  - **Memory-Centrix Fabric**
    - FabreX is the next generation, memory-centric fabric for a changing compute world. Effortlessly connect new memory / storage products, the multitude of new accelerators and your choice of processors either directly attached or via server configs like NVMe-oF (Non Volatile Memory Express over Fabric).
  - **Driving Down Cost**



- The result is lower Capex and Opex through less hardware, higher utilization of resources, lower power consumption, reduced power and less cooling. Avoid overprovisioning and add just the elements you need. Maximize utilization of the footprint of your data center and contribute to your bottom line.

## Conclusion

HPCCs are under tremendous pressure to maintain a sustainable compliance regimen for continuously evolving and increasing the usage and optimization of the deployment of services and resources to their user community. Deploying a HPCaaS set of tools and services to automate the lifecycle management activities of the HPCC can dramatically increase the usage and optimization of the resources within the HPCC, allowing entities to deliver the optimized HPC infrastructure faster, less expensive and with greater performance to their user community. Using this approach, to managing the infrastructure ensures that the HPCC is reliable, efficient, repeatable, and highly auditable. Entities using a HPCaaS approach can build a sustainable platform for delivering infrastructure that leverages their users' performance and capability to ensure that their bottom lines are maximized.

## About the Authors

**Charles T. Reynolds** is the founder and CTO of Technical Systems Integrators, Inc. (TSI). TSI is a leading provider of Infrastructure Management solutions serving customers globally since 1987. Chuck holds a BS in Computer Science/EE from Duke University and a MS in Engineering Management from Florida Institute of Technology (FIT). Chuck has been a member of ITEA, AFCEA, IEEE, and has over 30 years of experience in electronic design, testing, and infrastructure management tools. To learn more about TSI's product solutions, please reach out to us:

<https://www.tsieda.com>

<https://www.tsieda.com/blog>

**Steve Campbell** is the Chief Marketing Officer for GigaIO. Steve has held senior VP Marketing positions for HPC and Enterprise vendors. Steve was most recently Vice President of Marketing and Solutions of the Hitachi Server Group. Previously, Steve served as VP of Marketing at Sun Microsystems Enterprise Systems Products Group responsible for the mid-range and high-end Sun Fire and Sun Enterprise servers. He was executive sponsor for several high-profile customers in Asia and also responsible for Sun's High-Performance Computing initiatives and the Data Center Insight Programs



leading solution programs for data center consolidation and mainframe migration. Before joining Sun, Steve was a founding partner in a technology consulting company working with early stage technology and Internet start-ups helping raise over \$300M. He served on the boards of and as interim CEO/CMO of several early-stage technology companies. Steve was Vice President of Marketing at FPS Computing and held executive positions at Convex. For more information on FabreX: <https://www.gigaio.com>

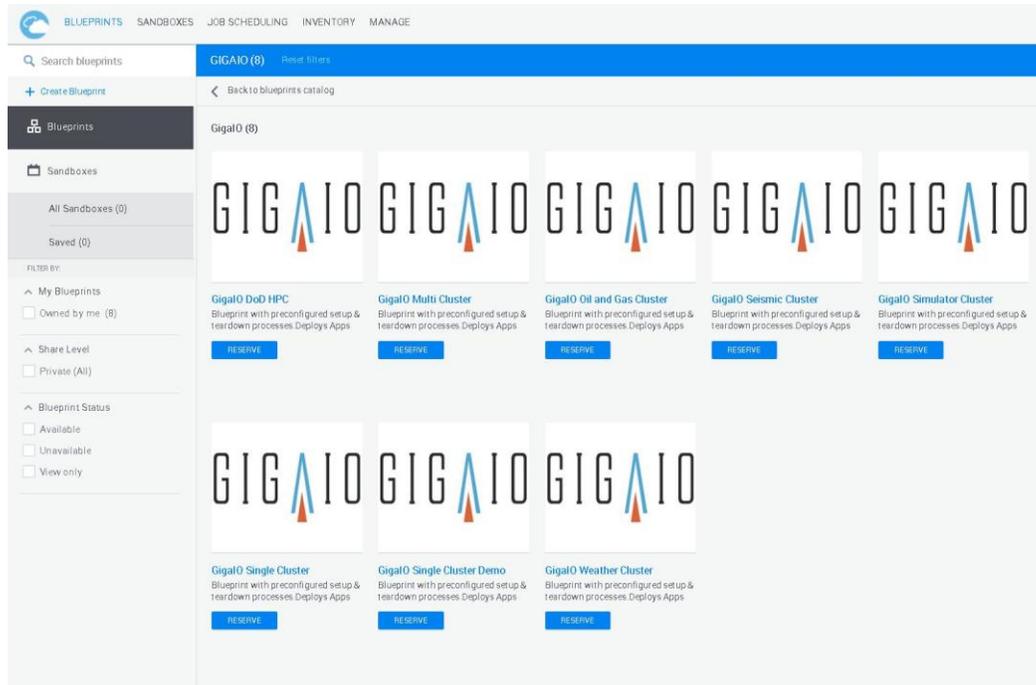


Figure 1: Web Portal for Self Service HPC Catalog

NAME	IP	OS	ARCH	RESOURCES	STATUS	ACTIONS
Fluxus1	10.13.1.1	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus2	10.13.1.2	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus3	10.13.1.3	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus4	10.13.1.4	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus5	10.13.1.5	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus6	10.13.1.6	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus7	10.13.1.7	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus8	10.13.1.8	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus9	10.13.1.9	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus10	10.13.1.10	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus11	10.13.1.11	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus12	10.13.1.12	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus13	10.13.1.13	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus14	10.13.1.14	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus15	10.13.1.15	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus16	10.13.1.16	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus17	10.13.1.17	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus18	10.13.1.18	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus19	10.13.1.19	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus20	10.13.1.20	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus21	10.13.1.21	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus22	10.13.1.22	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus23	10.13.1.23	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus24	10.13.1.24	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus25	10.13.1.25	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus26	10.13.1.26	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus27	10.13.1.27	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus28	10.13.1.28	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus29	10.13.1.29	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus30	10.13.1.30	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus31	10.13.1.31	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus32	10.13.1.32	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus33	10.13.1.33	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus34	10.13.1.34	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus35	10.13.1.35	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]
Fluxus36	10.13.1.36	CentOS 7	x86_64	1x Intel Xeon E5-2680 v4, 1x NVIDIA K80	Available	[Icon] [Icon]

Figure 2: HPC Resource Library and Search

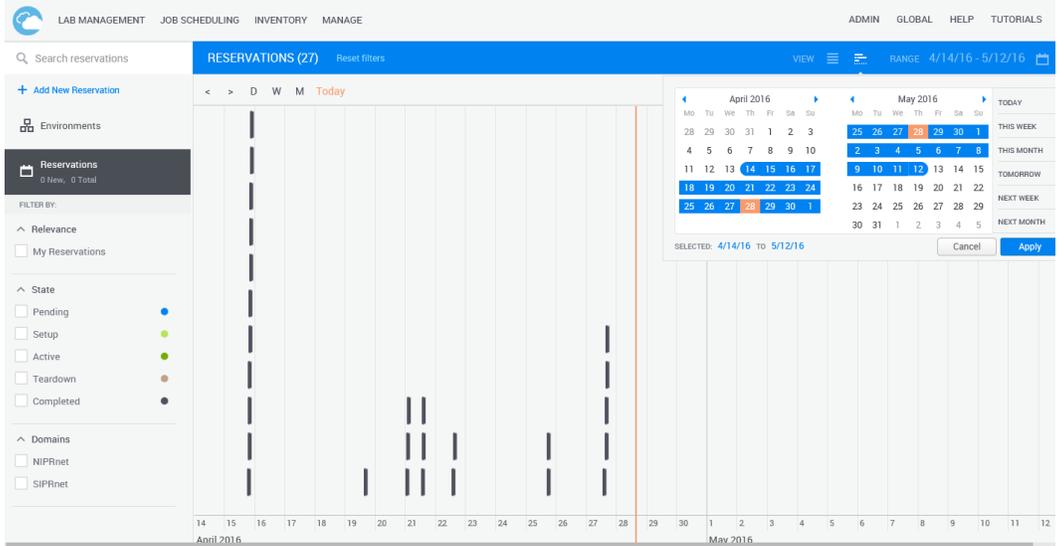


Figure 3: Scheduling of HPC Sandboxes

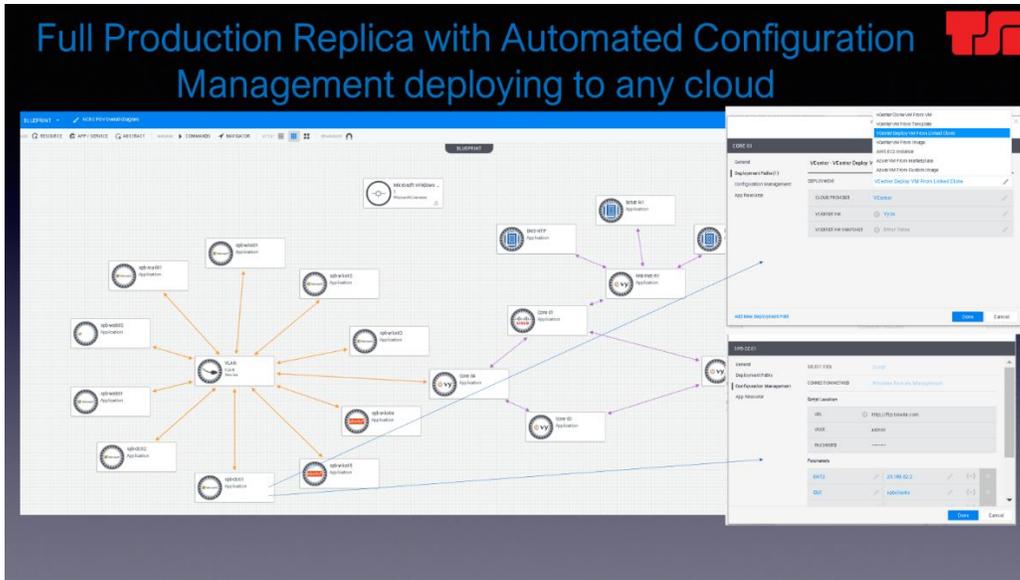


Figure 4: Resource Automation Commands

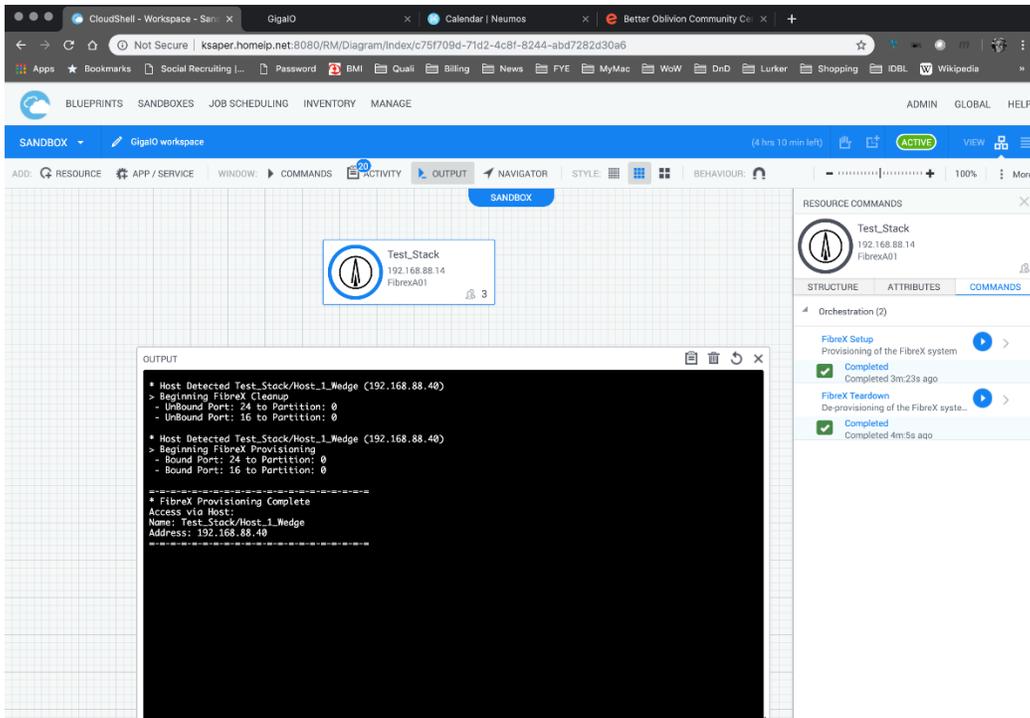


Figure 5: Complex HPC Stack Sandbox

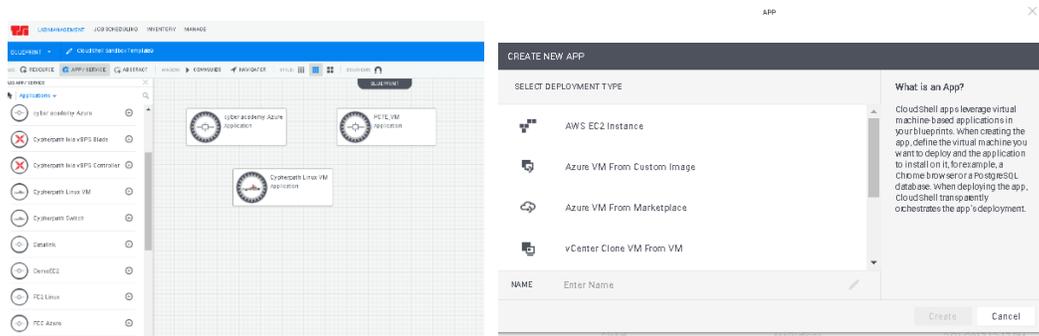


Figure 6: App and Service Support for Clouds

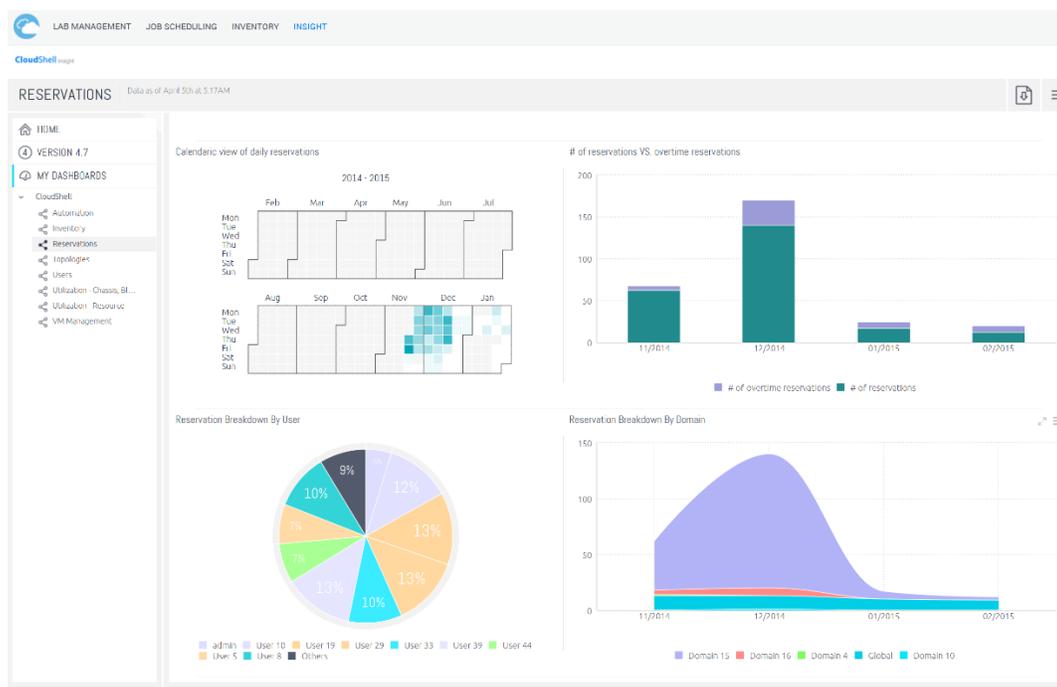


Figure 7: BI Dashboard (example of reservations dashboard)

[www.gigaio.com](http://www.gigaio.com)

[www.tsieda.com](http://www.tsieda.com)