Secure Web-Based Access for Productive Supercomputing

US Department of Defense High Performance Computing (HPC) Modernization Program communities are increasingly in need of access to highly capable computing, networking, storage, and software resources from their user enclaves that are administratively prevented from installation of applications due to malicious software risks. More generally, users are gravitating towards web-based workflows and away from traditional command line interfaces (CLI). The HPC Portal enables a productive and secure computational science environment.

Introduction

Enabling access to supercomputing is a complex process for the scientists and engineers focused on addressing large-scale problems for the US Department of Defense (DoD). Increasingly, network enclave security mechanisms block the traditional Kerberos mechanism of authenticating desktop clients to server supercomputers. To mitigate the risk of users being unable to access supercomputers, the DoD High Performance Computing Modernization Program (HPCMP) provides a Web-based software-as-a-service (SaaS) access approach known as the HPC Portal (https://portal.hpc.mil). The Portal supports the life cycle of design and development through operations (DevOps) as a software stack hosted on multiple virtual machines (VMs) for development, continuous integration, quality assurance (QA), staging, and production deployment. A SaaS approach benefits the user communities in several ways, including zero install, configuration, or maintenance updates on the user's part; a simplified and enhanced security model; a simplified user experience with or without a command line interface (CLI); central authorization of software read and execute controls; and potential access from Web-connected devices. Currently, the HPC Portal provides file and job management, CLI access, and commercial and government-off-the-shelf (COTS and GOTS, respectively)-hosted applications.

Representative Supercomputing Workflows

The HPC Portal enables access for a spectrum of user experience levels and workflows, from engineering students who are new to supercomputing simulation, to the existing HPCMP community of expert Linux users and developers. In response to this span of communities, the HPC Portal provides both HTML5-based Web applications as well as a lightweight Web-based shell terminal (http://code.google.com/p/shellinabox). In addition, for the many COTS and GOTS applications not provided as Web applications, remote desktop applications are made available via HTML5 VNC rendering. Although remotely rendered desktop applications have the advantage of zero code porting, they have the disadvantage of a lag in responsiveness that is inherent in VNC systems. From the user perspective, the HPC Portal would ideally perform at the same level of responsiveness as a locally installed desktop application. Hence, to improve user-visible responsiveness, dedicated HTML5 Web applications have been developed. The HPC Portal's support of standards-based Web browsers offers the benefit of not requiring additional user client security software or firewall settings. Security is handled through two-factor authentication, with hardware tokens and HPCMP account passwords. Both DoD common access card (CAC) smartcards and HPCMP-managed YubiKey mechanisms are supported. All network traffic is handled via the browser over the standard TCP/IP port using secure commercial protocols. An example of an HPC Portal workflow can be described with the HPCMP Computational Research and Engineering Acquisition Tools (CREATE) Kestrel fixed-wing Multiphysics application, one capability of which is simulation of an aircraft's aerodynamic performance. Generally, as a first step in this use case, the user generates surface and volume meshes representing the aircraft on local or remote desktop applications. Next, the user opens the Kestrel Web application and starts a new job, which includes uploading or linking to the mesh file as well as setting simulation conditions such as Mach number, altitude, simulation time-step size, and other parameters. The user can specify the job's HPC account, queue, number of CPUs, and maximum runtime duration. Kestrel's design enables responsive input verification and guidance for users without the need to resort to a CLI. The user can visually inspect the mesh with WebGL-based 3D graphics, which greatly reduces network transactions and improves responsiveness. Once the configuration is complete, the user submits the job, and the HPC Portal launches it on the designated supercomputer. The user can monitor job progress through the queued and run states and can monitor convergence of the solution in real time as the job executes. The user can also halt the job at any time, as well as request a graceful stop with restart files generated at a given iteration. Result files are available after the job completes, with solution visualizations available. In addition to these real-world communications solutions, the HPC Portal provides functional benefits that go beyond traditional command-line HPC tools. At the US Air Force Academy, aerospace researchers employ

multidimensional parameter sweeps to explore problem spaces such as flight envelopes for novel configurations. For other DoD users, the Portal workflow manager lets scientists define job sequences composed of separate applications that pipeline their data products into other applications. The HPC Portal's 3D visualization and full-featured file management capabilities simplify research and acquisition workflows across distributed DoD and defense industry enclaves.

Architectural Approach

The HPC Portal is deployed at all five HPCMP DSRC sites (https:\\centers.hpc.mil), each of which contains one or more supercomputers. The Portal deployments connect users and applications to HPC node pools via Representational State Transfer (REST) services that are internal to a particular deployment. As users enter with a Web browser, the Portal Apache HTTPD instance routes them into appropriate application frameworks (e.g. Apache Tomcat, Node, Python, etc.) acting as OpenID authentication clients by redirecting users to the HPCMP OpenID Server and then caching authenticated credentials for a time period thereafter. Application frameworks allow for a variety of implementation approaches for Web and VNC applications. Early versions of DoD's HPC Portal relied on the Liferay application framework to manage the overall Portal look and feel and also to provide a host environment for HPC web application front ends. The current implementation relies on efficient single page web applications, with core applications based on Facebook's "React" JavaScript library. More generally, within this microservice architecture, web servers that suit implementation languages of choice can be quickly instantiated as needed. VNC applications are delivered with a modified Guacamole VNC service, which provides a host environment for delivery of native, desktop-based applications. These applications are spawned in the user space on DSRC nodes made available for interactive processing. While VNC applications are used mostly for HPC pre- and post-processing, web applications can overlay full HPC run life cycles through the Portal REST services (PRS) application programming interface (API). The PRS API enables various levels of communication between Portal web applications and HPC interactive and batch compute nodes, including file management and abstractions for HPC job submission and monitoring. The PRS acts as a common API for all HPC Portal applications, greatly simplifying development of new applications. These abstraction layers simplify application access to all core HPC-related tasks such as job creation, submission, monitoring, runtime, and spool-down operations. PRS support spans from single-run jobs to restarts through execution of many related HPC jobs. The HPC Portal REST API enables Web application back ends to exist in practically any programming language. Future architectural changes continue and include scaling HPC Portal web services across multiple nodes, automated user management, and zero-downtime deployments.

Security

The security goals for this DoD program are to ensure confidentiality, integrity, and availability— commonly referred to as the CIA triad. To ensure that the HPC Portal meets these goals, security expertise is engaged from concept development through operations to assure that every component is vetted from a security perspective. This process requires that the necessary controls be implemented such that user data won't be vulnerable to eavesdropping or corruption. However, one of the weakest links in a security chain is the user client. This potential client vulnerability originates from the lack of control that the HPCMP security administration has on the client system accessing the Portal's servers. The HPC Portal alleviates this risk by providing a secured nexus for all services to the user via a Web browser. Utilizing a properly configured Web server, the HPCMP can ensure that users only access the Portal via an encrypted channel (https) with a certificate signed by a trusted DoD certificate authority. Execution of end-user applications on a centrally controlled system, as opposed to multiple desktops, ensures all identified vulnerabilities are properly mitigated. As long as users have a supported Web browser on their client system, they can perform tasks normally executed on their local client within the Portal's secure environment. The browser is the only weak link in the application communication chain: all the data within the Portal environment is protected by various security controls, including authentication, secure code, common component inventory, data confidentiality, and approved ports, protocols, and services (PPSs).

The HPC Portal's security starts at the front end, employing OpenID for user authentication. The OpenID service used within the HPCMP ensures that the user will face two-factor authentication when logging into the Portal. It's important to note that OpenID authenticates the user, but doesn't grant authorization or access privileges. OpenID uses either the DoD-

issued CAC or HPCMP Kerberos credentials, but its primary method for authentication is public-key infrastructure via the use of the CAC. Using OpenID ensures that no passwords will be transmitted to the Portal, with an additional advantage of authenticating and managing user sessions. To verify that the HPC Portal code is written securely, projects are scanned via static and dynamic analysis mechanisms. Static code analysis is an examination of source code, byte code, or application binary for conditions that can create a security vulnerability. Although a static code analysis is useful in identifying one category of potential security issues, a more comprehensive understanding can be gained during execution. Dynamic code analysis examines a running program while probing the processes for adverse behavior. Although it's time-consuming to run the source and binary of the various programs through these processes, doing so can potentially thwart accidental or malicious attempts to compromise the application. By integrating this process within the application's development life cycle, the time to deployment is reduced with the benefit of a more secure runtime environment. To ensure that any security vulnerabilities identified in individual components are tracked, an inventory of all components is maintained and compared against the National Vulnerability Database (NVD). Data is among the most important objects on any computer or network, including the Portal, so to protect it, all communication paths are encrypted. Security personnel ensure that only strong algorithms and properly-sized keys are used to ensure the encryption isn't cracked; they also run sniffing programs to view data as it is transferred between components to ensure that encryption is enabled.

DevOps: Develop, Test, Deploy, Operation, and Monitor

A development environment that meets the needs of the entire HPC Portal team is essential for the timely delivery of a quality product. From this perspective, the team includes DoD HPCMP representatives, software developers, IT administrators, QA, support, and security teams. The development environment supports the key HPC Portal goals of providing productive and reliable access to HPC resources. While there is much commonality, each DSRC has varying requirements and architectural differences. Examples include hosting different applications, variable installation paths and user configurations, and operating system differences on the supercomputer service nodes. Faced with these requirements and challenges, the HPC Portal team has created a staged development environment that is a "pipeline to production". This pipeline rolls out new features or fixes through increasing layers of control and scrutiny prior to production releases. The development environment implemented for the HPC Portal enables members to iterate quickly, test automatically, and maintain configuration management. It supports both the HPC Portal core team and the external groups developing new Web applications. As a feature matures from concept to production release, increasing levels of control in terms of access and configuration are applied. Concurrently, scrutiny from the QA and security teams ramps up. Applied as a whole, this allows for rapid software development while protecting users from an inadequately tested or secured product. Each Portal is instantiated as a replica of the DSRC architecture. A development cluster provides a representative supercomputer back end, allowing realistic behavior for user interactions and job submissions while ensuring that developer codes will work correctly on production systems. When all stakeholders agree, updates are pushed to the production systems during scheduled downtime, where another round of QA feature and regression testing is quickly completed before returning to operations. The HPC Portal team is exploring strategies and best practices to achieve zero-downtime upgrades for production users. For the operational portals, monitoring is provided by periodically running the automated regression tests. In an overall DevOps approach, the team is able to rapidly produce quality software products, secure access, and provide high reliability and performance. To improve HPC Portal operations, the team is automating processes in testing, user management, deployments, and monitoring.

Conclusions

The HPC Portal has enabled cost-effective, secure access to supercomputing capabilities from uniquely protected network enclaves for the first time. The use of modern DevOps practices and agile collaboration among development, security, supercomputing administrators, and stakeholders has led to widespread adoption by the DoD's scientific and engineering communities. This success for DoD programs' usage of supercomputing has led to increased utilization of HPC Portal services and support within a cost-constrained environment.