Proactive Health Monitoring and Maintenance of High-Speed Slingshot Fabrics in HPC Environments

Jeff Kabel
HPC Slingshot
Hewlett Packard Enterprise
Acworth, Georgia, USA
jeff.kabel@hpe.com

Mike Cush
HPC Global Support
Hewlett Packard Enterprise
Columbus, OH, USA
michael.cush@hpe.com

Mike Schmit
HPC Global Support
Hewlett Packard Enterprise
Bloomington, MN, USA
michael.schmit@hpe.com

Mike Accola
HPC Global Support
Hewlett Packard Enterprise
Bloomington, MN, USA
michael.accola@hpe.com

***Abstract*— This whitepaper addresses the critical need for maintaining the health of high-speed Slingshot fabrics in high-performance computing (HPC) environments. Identifying and resolving known issues swiftly is essential for optimizing HPC workload performance, yet pinpointing common and emerging problems can be highly challenging. We propose a proactive solution that leverages automated capture of key configuration and performance metrics, coupled with sophisticated event logic, to detect unhealthy components/known bugs within the fabric. This is achieved through the System Diagnostic Utility (SDU), integrated with HPCM and CSM software, which automates data capture and securely transmits it to HPE using HPE Remote Device Access (RDA).**

This solution is complimentary to other system monitoring solutions such as SMS and AIOps. In fact, this solution can also capture data from those tools to consolidate and enhance the level of data captured for analysis. The impact of this approach is significant: it enables faster identification and resolution of issues, thereby enhancing fabric performance and overall HPC job efficiency. Furthermore, the ability to analyze both historical and current data allows for real-time rule additions to address newly discovered bugs. The comprehensive visibility into issues across customer environments also aids HPE R&D in developing timely fixes and enhancements.

Keywords – Slingshot; proactive support; reactive support; monitoring; HPC; system support analytics; system support automation; support case automation.

I. Introduction

A. Problem Statement

The health and performance of an HPC (High-Performance Computing) technology stack are critical to ensuring optimal job execution and scalability. However, issues inevitably arise, often requiring reactive debugging and diagnosis. Reactive support presents several challenges: performance or availability is already impacted, time must be spent on initial triage, and if the problem is not quickly resolved, it escalates into an HPESC case. This process involves gathering data, collaborating with service teams, and sometimes engaging R&D. The longer this process takes, the greater the disruption for end-users. Reactive support cases can be categorized into several levels of complexity, each progressively more difficult to resolve:

- Known Issues: Straightforward resolutions, generally understood by service and R&D and often documented in Customer Advisories.
- Configuration Issues: Problems caused by misconfigurations.
- Edge Cases: Rare and highly specific scenarios.
- Zero-Day Issues: Completely new problems never encountered before.

Recent Slingshot cases escalated to engineering provide insight into the challenges all technologies face and highlight how HPE is leveraging innovation to improve the future of reactive case management. These sample cases primarily fall into the "zero-day" category—new software or hardware issues that arise due to the nearly infinite permutations of environmental conditions that are challenging to preemptively test.

Case 1: Jobs are failing, and CXI errors are being reported.
End-users report MPI job failures, accompanied by specific error messages from the nodes during execution:

```
   [Tue Feb 18 20:48:41 2025] cxi_ss1
0000:21:00.0: cxi0[hsn0]: C_EC_CRIT: C1_HNI
error: pfc_fifo_oflw (46) (was first error at
1739908121:067337283)
   [Tue Feb 18 20:48:41 2025] cxi_ss1
0000:21:00.0: cxi0[hsn0]: C_EC_CRIT:
pfc_fifo_oflw_cntr: 213
   [Tue Feb 18 20:48:41 2025] cxi_ss1
0000:21:00.0: cxi0[hsn0]: C_EC_DEGRD_NS: C_IXE
error: pbuf_rd_err (48) (was first error at
1739908121:067509999)
   [Tue Feb 18 20:48:41 2025] cxi_ss1
0000:21:00.0: cxi0[hsn0]: C_EC_DEGRD_NS:
pbuf_rd_errors: 14
```

Case 2: Node Crashes with Latest Slingshot Host Software (SHS) v12.0.0, USS v1.3

A new setting introduced in USS v1.3 for IOMMU passthrough can cause nodes running SHS v12.0.0 to crash when executing LNET.

Case 3: Post-Upgrade Job Failures with Slingshot v2.3.0
After upgrading to Slingshot v2.3.0, jobs that previously ran successfully now fail due to reported packet drops, despite the system appearing to come online normally.

This paper presents a vision for improvement by harnessing both technological and procedural advancements to revolutionize reactive support. The approach is twofold: first, to enhance efficiency in handling reactive cases, and second, to transition from a reactive model—particularly for complex, zero-day cases—toward proactive problem identification. This transformation will foster continuous improvement in future product releases through a strong feedback loop, fundamentally redefining the reactive case management paradigm.

II. Approach

This paper uses Slingshot as the primary model, but the concepts discussed are applicable to any technology. While learning from field examples in reactive cases is essential, a greater objective is to develop proactive checks that significantly reduce or prevent issues before they occur. The foundation of this approach lies in system state and support telemetry, which can be generated through ongoing monitoring outputs from tools like `fmn-show-status`, Redfish alerts, and telemetry. Reactive debugging and analysis can be performed using tools such as `linkdbg`, `show-flaps`, and `fmn-check-fabric`. Additionally, real-time monitoring and alerting can be achieved with tools like SMS and AIOps to track and report the status of fabric systems.

When addressing reactive support issues, it is critical to automate the collection of this data over time using Slingshot's native interfaces, as well as stateful and log data that may expire over time. Capturing and associating this evidence with the case is vital for effective resolution.

The automation of data capture and secure transport can be efficiently managed with HPE's System Diagnostic Utility (SDU) Toolkit. SDU is integrated into CSM [1], HPCM [2], Slingshot and ClusterStor/Neo [3], featuring the SDU CLI, which collects system state and telemetry data through extensible plugins and generates a support bundle referred to as a collection. The SDU Toolkit also incorporates HPE RDA for secure transport, which is the de facto standard for secure data transmission within HPE.

A. *Distinct Levels of Opt-in*

The System Diagnostic Utility (SDU) [4], its cloud-based back-end data lake, Metis [5], represents the second generation of support technology tailored for the high-performance computing (HPC) market. Building on over a decade of experience with the first-generation System Snapshot Analyzer (SSA) [6], Hewlett Packard Enterprise has strategically evolved its call home technology to maintain its leadership in the HPC market. While data collection and transmission to HPE are integral parts of the support process, this workflow is often susceptible to miscommunication and human error, which can result in incomplete data being gathered. Even when communication is clear, data collection often requires active involvement from multiple parties, a challenge that becomes more pronounced when issues are escalated to HPE's advanced product teams.

SDU is designed to significantly reduce the need for manual data collection and deliver immediate access to this data for all members of the case team. Ideally, a large portion of the required data can be automatically gathered or retrieved through straightforward tool activation on the customer's system, which takes less than ten minutes and requires no downtime. By minimizing the time spent by customers and HPE personnel on data collection, more focus can be placed on resolving the issue itself. Additionally, SDU ensures that identical data views are available both locally on the customer's system and to the HPE case team, enabling both sides to use the same tools for analysis—this capability is especially important for sites operating in disconnected environments.

With its flexible and configurable architecture, HPE aims to establish SDU as a standardized interface across all its HPC platforms. This standardization enables the creation of a new generation of reactive and proactive customer service capabilities, accommodating the needs of both connected and disconnected customers.

SDU operates on an opt-in model. At the first opt-in level, SDU collects essential support data and stores it locally on the file system. A key aspect of this data collection and storage process is its accessibility—site admins and team members can easily review the captured information without requiring advanced expertise in HPC data collection techniques. The data is presented in a single, human-readable format, enabling the use of common POSIX command-line tools for straightforward data analysis. By allowing users to configure the number of stored collections, SDU also facilitates tracking changes over time. For example, it can help answer questions like, "How has the network configuration changed since the last SDU triage collection when the system network was more responsive?" Additionally, engineering teams have embedded tools within their products to highlight known issues, offering valuable insights to both customers and HPE site teams. This concept will be explored in greater detail later in this paper.

The second opt-in level enables the secure transfer of collected data to HPE via Remote Device Access (RDA). This feature securely copies the local view of the support data to HPE, where critical system details are automatically populated into the asset records within the case management system, establishing a "source of truth" for service and R&D teams. For secure sites, a limited subset of the data can be captured in a compact JSON file, allowing for flexible transmission through any method preferred by the site, if/when sharing is desired. This level is highly customizable; customers can choose to share all collections or selectively upload specific data sets to HPE. The emphasis is on empowering customers with choices to suit their unique needs.

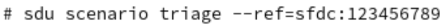
Automating the capture of support telemetry significantly reduces the time and effort required from customers and site team members, allowing them to focus on resolving the issue at hand. By securely collecting and transporting accurate "source of truth" data to HPE, and automatically associating it with a specific case, service and engineering teams gain immediate access to the necessary information to address the problem efficiently. This is accomplished through several mechanisms:

– **Automated Telemetry Collection**:

  Automated collection of support telemetry provides the service team with instant visibility into the customer’s unique configuration, eliminating the need for back-and-forth questions—only actionable answers are available to the case team.

– **Built-in Scenarios for Targeted Data Collection**:

  A service engineer may request a triage scenario using SDU. This scenario, manually executed by the customer or site team, captures all required information to address the issue and securely shares it within the case with a private case comment using a single command:

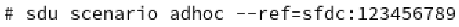


```
# sdu scenario triage --ref=sfdc:123456789
```

  NOTE: This level of detail is not typically suitable when attempting to attach to a case directly as it exceeds the limitation of SFDC.

– **Immediate Sharing Specific Data via 'Adhoc Scenario'**:

  Targeted support telemetry can be quickly gathered and shared using the "adhoc scenario" mechanism. This involves copying or linking command outputs or file content into a predefined adhoc directory, followed by executing the command:



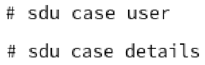
```
# sdu scenario adhoc --ref=sfdc:123456789
```

  This scenario, manually executed by the customer or site team, captures all information in the adhoc directory and securely shares it within the case with a private case comment.

– **Case Creation via Command Line**:

  A new feature current going through beta testing with a few early customers allows for the creation of HPESC cases directly from the command line. This feature captures the necessary data, securely transports it to HPE, and automatically opens a case without requiring the user to leave the system's command line interface. Example commands include:

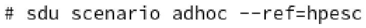
  (2) commands to provide information for the case:



```
# sdu case user

# sdu case details
```

  The command to initiate the data collection, secure transport, and instructions for the SDU back end to create the case on your behalf:



```
# sdu scenario adhoc --ref=hpesc
```

These tools jointly streamline the data collection process, reducing the time required by up to 95% and significantly decreasing the mean time to resolution (MTTR). While this approach is highly beneficial in reactive cases, it is even more critical for enabling the vision of proactive support, where issues can be identified and addressed before they impact operations.

Conventional methods for securely transferring the necessary information to effectively troubleshoot cases are inadequate, whether through attaching details directly to the case or using tools like HPRC [8] (commonly referred to as HPE Dropbox).

The SDU offers several advantages over the existing HPRC File Transfer Service currently used by many customers, particularly in terms of functionality:

– **Method to Move Data to HPE**
  - **HPRC**: Data transfers often involve multiple steps—moving files from the machine to another server or laptop, then to the HPRC landing site. Afterward, an HPC case team member must manually transfer the data to the internal area used by the service team.
  - **SDU**: With a single, secure outbound connection (port 443) configured to midway.ext.hpe.com, data transfer is direct to the final destination, streamlining the process.

– **Method to Collect Data for Transport**
  - **HPRC**: No built-in functionality for data collection.
  - **SDU**: Includes CLI scenarios such as daily, triage and adhoc to automatically gather the necessary data for troubleshooting issues.

– **Updates to HPE Salesforce Asset Facts**
  - **HPRC**: No capability to update asset configuration data.
  - **SDU**: Automatically updates asset configuration data in HPE Salesforce when daily scenarios are uploaded.

- **Data Arrival Notifications**
  - **HPRC**: No notification system in place for when data is received.
  - **SDU**: Automatically generates notifications in the associated case when data arrives in HPE's backend data lake, keeping engineers informed.

- **Data Size Limitations**
  - **HPRC**: While it supports larger files than the 2GB limit for case attachments in HPESC, storage limitations can restrict how long files are retained.
  - **SDU**: Supports multiple large files (10s to more than 100s GB) without the storage constraints of HPRC. Its backend data lake offers virtually unlimited storage, accommodating multiple multi-GB files for a single issue, far exceeding HPRC's capabilities.

The third opt-in level is designed for customers who wish to grant HPE engineers secure remote access to their systems through RDA's Interactive Device Access (IDA) feature. This capability facilitates the use of support tools such as SSH, SCP, VNC, RDC, and web UIs to interact with customer devices in real-time. It allows support engineers to collaborate on the same device and provide hands-on assistance. Importantly, customers maintain full control over enabling this feature, deciding who can connect and specifying the type of connection allowed. This opt-in level is entirely independent of the other levels and can be enabled as a standalone option if necessary.

B. Summary of security Features

SDU Toolkit

- The SDU Toolkit operates within an OCI container, ensuring isolation from the host system.
- The SDU Command Line Interface (CLI) is written in Python, and fully transparent for review.
- Customers retain complete control over the upload of support telemetry data.
- Data collections are stored locally, allowing customers to inspect them prior to any upload.
- Collection objects are hashed and validated against the collection manifest to ensure integrity.
- Standard use cases do not involve the collection of Personally Identifiable Information (PII) or customer-specific data.

Secure Transport

- Remote Device Access [6] (RDA) is recognized as one of the most secure data transport methods in the industry.
- It is widely deployed globally across hundreds of thousands of devices.
- See section **Remote Device Access (RDA) Security Architecture** for more details.

SDU Backend (Metis)

- Data is the heart of the solution, so we encrypt all data at rest using 256-bit AES encryption and it is FIPS 140-2 compliant. Data in transit allows for system-wide TLS 1.2 protocols as well as IKEv2 and SSH2.
- We leverage cloud provided tooling which highlights security related issues and provides an overall assessment of Metis' security posture and compliance against a variety of policies (ex. SOC TSP, ISO 27001, PCI DSS 3.2.1). Along with this feature set, it also provides vulnerability scanning and reporting via the open-source tool Qualys. These scans are performed on all VMs running in the Metis infrastructure and aggregates the results into an easy-to-use dashboard.
- Support telemetry for each asset is stored within its own dedicated container.
- There is a storage key vault per environment (PRO, ITG, and DEV), so access control can be handled per environment. We rotate storage account keys using a tick tock pattern based upon an even/odd check of the instance number. At the end of a roll forward to a new instance, the storage account key that is no longer being referenced is regenerated.
- Access to uploaded data is strictly limited to authorized HPC service and R&D team members.

Remote Device Access (RDA) Security Architecture

The HPE Remote Device Access (RDA) platform is a state-of-the-art solution designed to deliver highly secure, scalable, and compliant remote transport and access for device management. Leveraging advanced encryption, trust-based access controls, and robust operational security, the platform ensures data confidentiality, integrity, and availability while aligning with global regulatory standards.

This section provides a detailed overview of the key capabilities, architecture, security framework, and operational controls of the RDA platform, emphasizing measures in place to safeguard customer environments and maintain trust.

Key Capabilities

The Remote Device Access (RDA) platform provides a variety of features aimed at enabling secure and efficient device management.

The System Diagnostic Utility (SDU) currently utilizes the Asynchronous File Transport (AFT) and Interactive Device Access (IDA) capabilities of the Remote Device Access (RDA) platform, but additional critical features can be enabled as opt-in possibilities to enhance functionality. AFT facilitates reliable file transfers, such as telemetry and crash dumps, from devices to HPE for analysis. IDA allows real-time interactive support sessions using tools like SSH, VNC, and web UIs. Other key capabilities of the platform include the Content Delivery Service (CDS), which enables push/pull content delivery for firmware updates, patches, and diagnostics while incorporating malware checks and activity logging; Synchronous Data Transport (SDT), offering a reverse proxy-like service for secure exposure of REST APIs and web content; Mass Notification Bus (MNB), which enables HPE to send notifications, such as firmware updates, to devices; and Device Awareness Service (DAS), which monitors lifecycle events like connection, disconnection, and registration.

The architecture of the RDA platform incorporates several participants, including customer devices, HPE Midways, secure key managers, trusted servers, and technicians. It supports multiple connection modes, such as Customer-Initiated TLS, HPE-Initiated TLS over IPsec/SSH, and clientless REST APIs. Trust management assigns devices and participants trust levels (A to F) based on identity verification and authentication strength. To ensure robust security, the platform employs multi-layered protection across the network/transport, tunnel, session, and secure layers. Communications are encrypted using IPsec and TLS, authenticated tunnels provide end-to-end security, Midways manage secure support sessions, and modern cryptographic algorithms like ECC and SHA-256 safeguard encryption and authentication. Operational security measures, such as patch management, penetration testing, and 24/7 cyber defense monitoring, further enhancing reliability. The Public Key Infrastructure (PKI) system issues, maintains, and revokes X.509 certificates for device and server identity authentication, while fraud prevention mechanisms protect against unauthorized access, identity duplication, and black-market sales. Compliance with global standards such as GDPR and CCPA ensures adherence to regulatory requirements.

Authentication and authorization mechanisms in the RDA platform employ a layered approach to securely manage access to devices and systems. Authentication methods include X.509 certificates managed by HPE's PKI, multi-factor authentication (MFA) using DigitalBadge certificates, YubiKey, and OATH tokens, SSH key-based authentication for HPE-initiated connections, TLS authentication through a three-phase PKI process, and anonymous access for minimal trust-level interactions with public resources. By integrating role-based permissions, trust levels, and encryption, DigitalBadges simplify secure communications while ensuring robust protection against fraud and unauthorized access. Authorization is granted based on trust levels, roles, and session-specific access controls. Two-step authorization includes Midway-level checks for user roles and session types, followed by device-level validation for user trust and session timing. Trust levels (A to F) determine the strength of authentication, and role-based access control (RBAC) ensures participants are limited to authorized actions. Session-level authorization requires explicit approval and validation through access control lists (ACLs), while a chain-of-trust combines tunnel and session attributes to establish a systemic trust score.

Access Control Lists (ACLs) are integral to the RDA platform's security, enforcing rules that govern participant access to resources and actions. ACLs identify participants through unique Station IDs, certificates, or SSH keys and apply rules based on identity, role, trust level, resources, and activities. Enforcement occurs at both the Midway server and device level, ensuring comprehensive oversight through two-step verification. Trust levels influence ACL decisions, with higher trust levels enabling privileged access. ACLs are dynamically updated in response to changes in participant trust, session attributes, or customer configurations, providing fine-grained control and adaptability. For example, a support technician may be authorized for firmware updates and diagnostics, while an anonymous user may access only public, read-only resources.

Midways serve as the central communication hub in the RDA architecture and are secured through a multi-layered approach. Authentication relies on X.509 certificates managed by HPE's PKI, including certificates for server verification, tunnel authentication, and device ownership signing. A three-phase PKI authentication process ensures secure tunnel establishment. Network security is enhanced through IPsec encryption, controlled IP parameters, tightly managed ports and addresses, and firewall rules to restrict unauthorized traffic. Intrusion detection systems (IDS) monitor and mitigate potential cyber threats, while compartmentalization isolates Midways in secure network zones. Role-based access control (RBAC) restricts operations based on participant roles, and Enterprise Secure Key Managers (ESKM) safeguard root certificates and keys. Fraud detection mechanisms identify and block suspicious activity, such as duplicate devices or spoofed connections. Operational security measures include regular patching, penetration testing, and 24/7 monitoring by HPE's Global Cyber Defense Center (CDC). All Midway activities are logged and audited for compliance, and physical security at HPE data centers ensures hardware protection against tampering.

In summary, the RDA platform delivers highly secure, scalable remote access solutions for device management. By integrating robust authentication, authorization, and access control mechanisms with layered security measures, the platform ensures data confidentiality, integrity, and availability while maintaining compliance with global standards. Through its comprehensive capabilities, such as AFT, IDA, CDS, SDT, MNB, and DAS, along with stringent Midway security protocols and dynamic ACL enforcement, RDA provides a trusted and flexible framework for managing remote connectivity and device interactions.

III. Process

Having outlined a critical prerequisite—securely transferring support telemetry to HPE—let's explore what happens next. Throughout the development of HPC Call Home, our efforts have been dedicated to creating methods that streamline the process of uncovering valuable insights for customers, Global Remote Support (GRS), the Engineering Resolution Team (ERT), Product Management, and R&D. By rapidly identifying and emphasizing "notable events," we deliver several key value propositions.

Proactive alerting for known events can be implemented through various methods, such as utilizing on-system debug tools like `fmn-check-fabric`, which is part of the Slingshot debug package, or through post-SDU collection processing within HPE's backend systems. HPE Call Home processing enables proactive alerting by integrating system-level debug tools from different product streams. A key benefit of performing proactive checks within the Call Home pipeline is the flexibility it offers as these checks can be deployed immediately, unlike on-system tools that require a software release or patch. Furthermore, running these checks in the Call Home pipeline addresses the challenge of some customers not consistently using local debug tools.

"Emerging events" are newly identified or previously unseen occurrences, often stemming from early-stage zero-day bugs, environmental factors, or process-related errors. These events are typically the most difficult to resolve, as they are unprecedented and demand thorough root cause analysis.

To adopt a more proactive approach, the aim is to transform these unknown "emerging events" into known events, integrating the insights back into site tools like `fmn-check-fabric`. A broader goal is to leverage these learnings to evaluate the prevalence of such events across customers and systems. By gathering detailed statistics, we can enhance product quality and enable R&D to focus on accelerating and improving future product innovations.

This entire process relies on high-quality data, which is most effectively captured at the moment the issue occurs. By combining routine SDU data collections with manual collections when necessary, we ensure the data required to drive this process is available. Collaboration with customers during the identification of emerging events and ensuring this data is accessible to R&D for root cause analysis greatly expedites resolution. Automating the data capture process further optimizes efficiency and streamlines the workflow.

As real-time data flows into the Metis backend, a rules engine—integrated into the Metis operational pipeline—is used to identify fingerprints of both known and emerging events within the securely transmitted SDU collection data. This approach originated with the ClusterStor L3 Engineering Resolution Team, where a pilot project was initiated to rapidly identify customer systems experiencing specific conditions. The pilot demonstrated significant success, paving the way for the development of a more robust and formalized process, now known as the Interesting Events Rules Engine.

The concept was later introduced to the Slingshot R&D team as a potential solution to reduce their workload on customer cases. It quickly became evident that this approach offered tremendous value not only to the team but also to customers, enabling the identification of known issues and providing actionable insights that informed the development of Slingshot products based on real-world customer scenarios.

The primary objective of the Interesting Events Rules Engine is to deliver a flexible mechanism for detecting unique patterns in customer deployments—which can vary significantly—that may lead to performance degradation or even imminent failures. Detection is designed to operate on real-time data from customer systems, with an additional capability to analyze historical data previously uploaded and stored in the Metis data lake.

While basic string pattern matching proved useful, it fell short when addressing complex conditions requiring refinement. To overcome this, we implemented advanced logical tests that can be combined to create a sophisticated rules-based engine. String parsing is powered by common regular expressions, making it straightforward to create new rules or update existing ones, ensuring the system remains adaptable and easy to manage.

The Interesting Events Rules-Based Engine is not a standalone application but rather a shared class utilized within the Metis pipeline. Rules are used to define the characteristics of the interesting events being identified. Multiple rules can be combined to create more complex rules; for example, Rule C is considered TRUE if both Rule A and Rule B are TRUE.

Actions specify what should occur when an interesting event is detected. The currently implemented actions include:

- Sending an email
- Sending a Slack notification
- Future capability: Automatically open a support case

Each rule can be associated with multiple actions. Both rules and actions are designed to be general purpose and reusable, ensuring flexibility and adaptability.

IV. Cases revisited

Let’s revisit the zero-day cases we initially discussed through a new perspective—HPC Call Home's interesting events processing. This system is proactively detecting these conditions across all customers who participate in the HPC Call Home framework.

- Case 1
  - Summary: Jobs are failing, and CXI errors are being reported by nodes.
  - As data is uploaded to the Metis data lake, the output from the fmn-show-status command reveals that the

    number of interfaces in the cassini-policy does not align with the number of ports in the qos-II-be_be_et-cassini-policy. This discrepancy allows us to notify the administrator to verify and ensure the correct port-policies are applied to the fabric.
  - In this instance, the issue was resolved reactively by analyzing the collected data.
- Case 2
  - Summary: A new issue was identified with Slingshot Host Software (SHS) v12.0.0 and USS v1.3 on Blanca Peak blades.
  - This issue could be proactively detected by regularly analyzing collected data, particularly for systems with Blanca Peak blades installed. Administrators could then be warned to either remain on SHS v11.0.0 or apply specific settings to mitigate issues observed with the newer SHS release.
  - By monitoring which configurations customers are using, the Slingshot team gains valuable insights into the adoption rates of various releases, enabling them to prioritize testing and determine where fixes need to be applied.
- Case 3
  - After upgrading to Slingshot v2.3.0, jobs that previously worked correctly began to fail.
  - This issue could be proactively identified by continuously parsing collected data, which includes reports of Switch ASIC errors. A new type of "Empty Route" issue was discovered in Slingshot v2.3.0, causing traffic stalls under specific conditions when a switch is reset. By parsing the ASIC error output daily, we can detect if this error is occurring on a switch within the fabric. Once identified, the site administrator is informed of the issue, provided with a new Customer Advisory, and directed to an updated fmn-debug RPM that includes a tool to better detect this specific problem in the future.

In all these cases, once the "fingerprint" of a complex zero-day issue is identified, an Interesting Events action can be established to quickly detect and address the problem across all customers. Additionally, the insights gained from these discoveries can be used to improve included tools, provide valuable input for new customer advisories, and offer feedback to Slingshot engineering to enhance future releases.

Anticipated Outcomes of the Interesting Events Rules Engine

- Automatically escalate health-related events to the service team or site team for faster resolution.
- Highlight known issues that have not yet been incorporated into local tools, such as fmn-check-fabric.
- Elevate known issues that are already recognized and defined in local tools, like `fmn-check-fabric`.
- Detect problematic or suboptimal software/hardware combinations and configurations.
- Avoid waiting until multiple cases (e.g., eight or more) are reported to address an issue.
- Proactively search for the issue in historical case data to identify patterns or missed detections.
- For emerging issues, use Interesting Events to assess the scope and scale of the problem by identifying how many customers are affected by a zero-day software bug and analyzing similarities or differences in their configurations to fully understand the issue.
- Once emerging events are clearly identified, transition them into proactive checks in on-system tools and real-time monitoring through SDU/Metis.
- Generate detailed emerging event reports that provide valuable insights for R&D, enabling the development of new processes, updated documentation, customer advisories, fixes, enhancements, and improved logging. This allows R&D to spend less time resolving repetitive cases and more time innovating new features.
- Utilize Interesting Events to support customer advisories, particularly for those who may not review advisories promptly.
- Free up R&D resources to focus on creating new and enhanced product features, especially after root causes for emerging events have been identified—reducing duplication of effort across multiple customers experiencing similar issues.
- Improve the quality of logs, making them clearer, more detailed, and more actionable.

Conclusion

This paper presents a vision for progress by harnessing both technological and procedural innovations to revolutionize reactive support. The strategy is centered on two key goals: first, enhancing efficiency in handling reactive cases, and second, transitioning from a reactive approach—particularly for complex, zero-day cases—to a proactive model of problem detection and resolution. This proactive shift will foster continuous improvement in future product releases through a strong feedback loop, redefining the traditional framework of reactive case management.

We are thrilled to introduce SDU as part of our ongoing commitment to advancing support in the complex landscape of

HPC. Building on a decade of success with our first-generation support technology, SSA, we are confident that SDU, our second-generation solution, will deliver exceptional value in ways that were previously unattainable.

In closing, to reduce resolution times and accelerate the transition to a proactive support model, HPE seeks your partnership. By enabling SDU and securely transmitting support telemetry, we will not only gather the data needed to create fingerprints for known issues but also unlock the ability to proactively reach out to customers who are currently or may soon be affected by issues identified through the Interesting Events engine. Our L3 and engineering teams are actively developing interesting events and rules to enhance the support experience, and we are excited to continue working together to achieve these advancements.

Acknowledgment

We extend our heartfelt gratitude to the customers who have engaged with our first- and second-generation solutions, System Snapshot Analyzer (SSA) and System Diagnostic Utility (SDU). Your contributions and participation in advancing HPC call-home solutions have been invaluable. The feedback and telemetry processing you provided have been instrumental in driving the improvements seen in System Diagnostic Utility (SDU) and Metis. We would also like to express our appreciation to our partners on the RDA team, whose expertise has refined the security and monitoring of a robust remote connectivity service integrated into our solution. Their dedicated efforts, in collaboration with HPE Cybersecurity, have been crucial in achieving the high level of security and reliability we deliver today. Finally, we acknowledge the invaluable contributions of HPC Global Service, Engineering Resolution, HPC R&D, and manufacturing teams, whose insights and guidance have shaped our solutions. A special thanks goes to the L3 HPC Storage Service team and the Slingshot engineering team for their unwavering commitment to excellence and customer satisfaction in guiding our transition into proactive support.

References

[1] *"HPE Cray Supercomputing EX with CSM System Diagnostic Utility (SDU) Administration Guide (3.3.7) (S-8035)," Published: January 2025," Hewlett Packard Enterprise, 2023. [Online]. Available: https://support.hpe.com/hpesc/public/docDisplay?docId=dp00005619en_us&docLocale=en_US. [Accessed: Mar. 10, 2025].*

[2] *"HPE Performance Cluster Manager Software Administration Guide," Hewlett Packard Enterprise, 2023. [Online]. Available: https://support.hpe.com/hpesc/public/docDisplay?docId=dp00005155en_us&docLocale=en_US. [Accessed: Mar. 10, 2025].*

[3] *"HPE Cray Supercomputing Storage Systems E1000 7.x Administration Guide S-9601," Published: January 2025," Hewlett Packard Enterprise, 2025. [Online]. Available: https://support.hpe.com/hpesc/public/docDisplay?docId=sd00006113en_us&page=GUID-7FFCB729-CA9E-4999-A6C9-37EDC7131667.html&docLocale=en_US. [Accessed: Mar. 10, 2025].*

[4] *"Getting Started with HPC Cray System Diagnostic Utility (SDU)," Hewlett Packard Enterprise, 2025. [Online]. Available: https://support.hpe.com/hpesc/public/docDisplay?docId=sf000105984en_us&docLocale=en_US. [Accessed: May 5, 2025.]*

[5] *"Metis (mythology)," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Metis_(mythology). [Accessed: Mar. 10, 2025].*

[6] *"Reducing Mean Time to Resolution (MTTR) for Complex HPC-Based Systems with Next Generation Automated Service Tools," Cray User Group, 2024. [Online]. Available: https://cug.org/proceedings/cug2024_proceedings/includes/files/pap133s2-file1.pdf. [Accessed: Jan. 10, 2025].*

[7] *"HPE Remote Device Access Security Technical Paper v1.27," Hewlett Packard Enterprise, 2023. [Online]. Available: https://support.hpe.com/hpesc/public/docDisplay?docId=a00006791en_us. [Accessed: Apr. 18, 2025].*

[8] *"HPRC File Transfer Service," Hewlett Packard Enteprise, 2023. [Online]. Available: https://hprc-h1.it.hpe.com/hprc/. [Accessed: Mar. 27, 2025.]*

[9] *"Rule-based system," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Rule-based_system. [Accessed: Mar. 10, 2025].*

"HPE 2024, HPE Cray Supercomputing Redfish Crawler for Linux Operating Systems," Hewlett Packard Enterprise, 2025. [Online]. Available:. [Accessed: May 1, 2025]

This paper utilized ChatHPE to assist refining sentence structure and summarization. All outputs generated by ChatHPE were reviewed, verified, and modified by the authors to ensure accuracy and originality. ChatHPE was cited in accordance with APA guidelines.