4th Gen Intel® Xeon® Scalable Processors: Reliability, Availability, and Serviceability

Technical Paper

April 2023

Revision 1.0

Document Number: 775813

Contents

Figures

Tables

Revision History

Revision Number	Description	Date
1.0	• Initial release of the document.	April 2023

1 Overview

This paper provides an overview of 4th Gen Intel® Xeon® Scalable processors' (herein after known as the "processor" or "processors") Reliability, Availability, and Serviceability.

Topics include an introduction to Intel® Xeon® processor RAS, RAS management, debug, telemetry features, and Intel's RAS platform application engineering support.

Notice that RAS feature availability may vary by processor SKU and/or implementation/configuration options. Additionally, not all features, although being used, may be visible to end users.

Contact your Intel representative or your system integrator with any further inquiries.

2 Introduction

The term Reliability, Availability, and Serviceability (RAS) for a server system is simply defined as:

- The Reliability of a system typically refers to its ability to continuously produce correct results and ensure data integrity. A reliable system has infrequent component faults and the capability to detect, contain, correct, log, and signal errors.
- The Availability of a system is its ability to remain operational even in the event of errors or faults. An available system has fault handling capabilities to map out failed units or operate in a degraded mode. Available systems minimize application outages and maximize the time that they are up and running.
- The Serviceability of a system generally refers to the process of easily identifying and diagnosing failures as well as ease of repair. The goal is to normally reduce system down time. A serviceable system's faults can be quickly repaired for restoration to a fully operational state.

One may observe that high reliability and high serviceability result in higher availability since a system that does not fail is highly available. If it does encounter a fault, high serviceability ensures that the system comes back up into operation faster, thus increasing the overall availability.

A well architected and robust Intel Xeon processor and server-based platform ingredient RAS is the foundation of high quality of Intel Xeon server-based platform. Intel Xeon servers incorporate additional complementary platform monitoring and manageability technologies such as telemetry, in-field scanners, and predictive failure analyzers that play a crucial role in management of systems for maintaining a reliable operational state as well as autonomous crash loggers that help diagnose system faults for prompt serviceability.

The Reliability, Availability, and Serviceability of a modern data-center server fleet is critical not only to maximize ease of system management for reliable operation but also for maintaining customer Service Level Agreements (SLA) and a reduced Total Cost of Ownership (TCO) of the server platform.

One 2022 Hourly Cost of Downtime survey indicates a single hour of server downtime totals $300,000 or more for 91% percent of mid-sized enterprises (SMEs) and large enterprises. Among that 91% majority, nearly 42% of corporations stated that hourly outage costs can range from one million ($1M) to over five million ($5M). [1]

Intel Xeon server platforms are designed and manufactured to be a rock-solid foundation for even the largest scale computing applications with the highest resiliency against faults based on the following three key tenets:

1. Silicon quality
2. Robust platform hardware, software, and firmware RAS architecture
3. Platform manageability

Intel Xeon server platforms offer a portfolio of RAS features that cater to the needs of a broad range of datacenter deployment models. Intel Xeon RAS features have evolved from collective insights over several generations of large scale-deployments. This experience, coupled with customer feedback, has resulted in a comprehensive set of RAS capabilities, feature selection, and enablement designed to address our customers' unique datacenter needs. Continuing with the trend of customer driven RAS richness and flexibility, the processor added and refined RAS features to meet our customers' evolving RAS needs.

RAS features are co-architected within the CPU silicon architecture, microarchitecture, firmware, and system software stack. Developing a fine-tuned RAS solution requires deep insights into platform's RAS architecture and deployment's fault handling approach. The purpose of this paper is to familiarize the audience with Intel Xeon server platform RAS philosophy and explain RAS capabilities of the processors. For a technical overview of processors readers are referred to Intel article "Technical Overview of the 4th Gen Intel Xeon Scalable processor family" [2].

In addition to a rich platform RAS feature set, this processor-based server platform incorporates a wide range of platform RAS management avenues that are designed specifically for cloud-scale fleet operators to enable automation of system log gathering, log-interpretation, and platform telemetry collection for correlation.

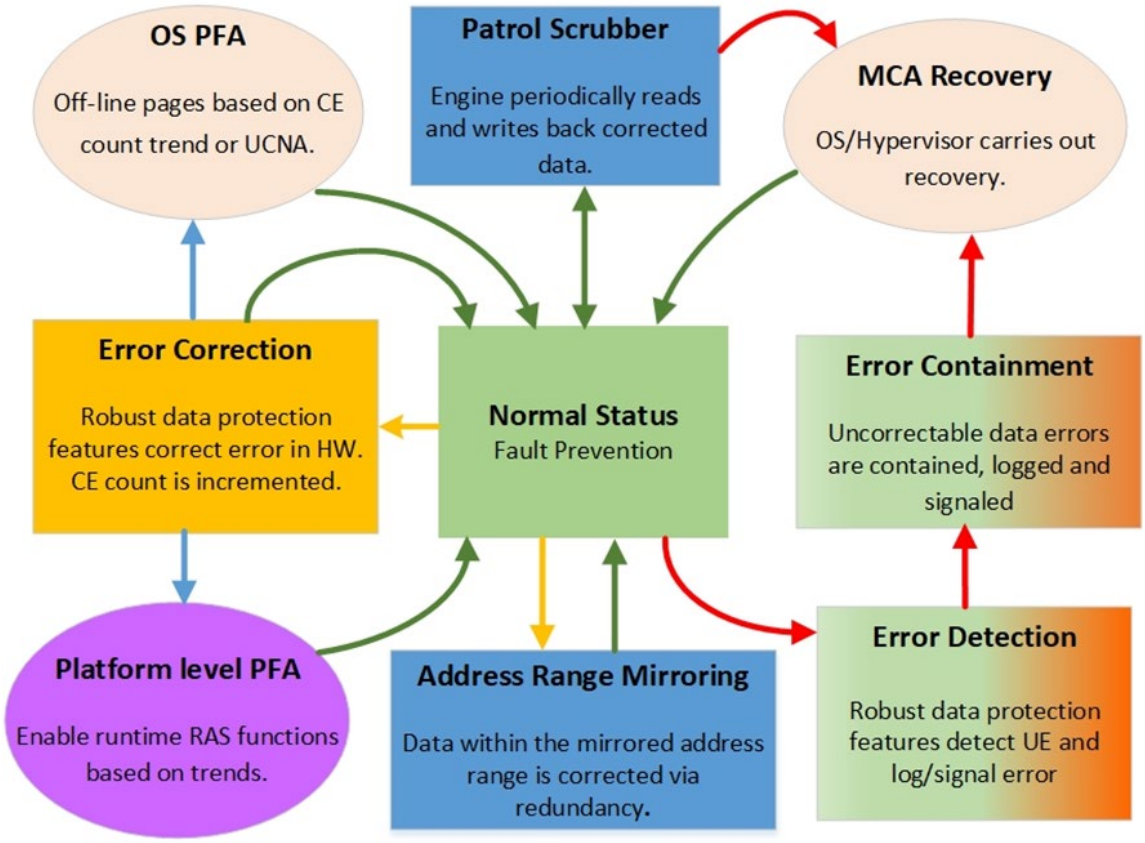
3 Synergy of RAS Features

Intel Xeon processors are equipped with a sophisticated line up of RAS features to optimize the system for a high level of resiliency. These features and techniques range from Error Correcting Code (ECC) protections of data to more advanced capabilities such as Address Range mirroring and Machine Check Architecture (MCA) Recovery. Even with a high level of fault prevention measures, faults may occur. The aim of the processor RAS mechanism is as follows:

- Correct as many faults as possible.
- Apply Predictive Fault Analysis (PFA) to learn from the corrections to prevent future failures.

When an uncorrectable fault does arise, apply recovery mechanisms to limit the impact of the failure to the smallest footprint possible.

Figure 3-1. RAS Virtuous Cycle – Example of RAS Runtime Operations



Intel Xeon RAS features typically work in a virtuous cycle to increase the availability of the system since ultimately, availability translates directly to the Total Cost of Ownership (TCO) and improved SLA of the server platform. It is difficult to illustrate such a virtuous cycle with all the features and capabilities that the processors support. Figure 3-1. RAS Virtuous Cycle – Example of RAS Runtime Operations illustrates a limited example.

The following sections describe each aspect of RAS[1] that makes processors highly resilient.

1 Feature availability may vary by processor SKU and/or implementation/configuration options, additionally not all features, although being used, may be visible to end users, for details please contact your Intel representative or system integrator

4 4th Gen Intel® Xeon® Scalable Processor RAS Features

4.1 Reliability Features

The reliability of an Intel Xeon server-based platform is measured by its ability to detect, correct, and report errors. With the increasing number of transistors and smaller process geometries, reliability has never been more important than it is today. In real terms, reliability translates to ensuring that data integrity is maintained during operations such that the probability of incorrect outcomes in computations is rare. It also means preventing or avoiding failures by applying preventive actions judiciously. 4th Gen Intel Xeon Scalable Processor-based platforms are architected with a philosophy that reliability starts from the lowest level of silicon circuit quality to the highest-level system software reliability capabilities.

Processors are equipped with a variety of reliability features. Within the System-on-Chip (SoC), silicon structures are protected by ECC or parity. Error protection mechanisms protect data as it travels from source to destination. This includes the Intel® Ultra Path Interconnect (Intel® UPI) fabric, which is protected by CRC coupled with a retry mechanism. Advanced Error Detection and Correction (AEDC) improves the fault coverage within the processor core pipeline.

Error containment is another aspect of reliability that ensures that the effects of an uncorrectable data error are limited in scope. Corrupt Data Containment is a method by which an uncorrectable data error is tagged with a "poison" indication. Data that is tagged with a "poison" indication is said to have been poisoned. Once data is poisoned, the poison indication will travel synchronously with the data and be stored along with the data. Corrupt Data Containment is also instrumental in enabling a major availability feature, MCA Recovery, which is discussed in Section 4.2 Availability Features. Asynchronous uncorrectable errors may be contained using the Viral form of containment.

Memory is one of the significant sources of errors in servers. Memory faults may manifest in several ways, ranging from single bit failures to complete device failures. A reliable server must fix as many of these as possible to prevent failure of the entire system. 4th Gen Intel Xeon Scalable Processors support DDR5 10X4, 5X8 and 9X4 DIMMs. The first line of defense against memory errors is the error correction and detection mechanism in the memory controller known as Single Device Data Correction (SDDC). Intel has worked very closely with DRAM vendors and customers to identify the dominant fault types and provide full coverage for DDR5 bounded faults in all DIMM types.

In addition, Intel has developed a new method called Persistent Fault Detection (PFD) that helps detect persistent faults and enhances ECC coverage in all DIMM types. Finally, there exists a host of other RAS features that keep the memory sub-system operational even in the face of failures. Upon the first failure of a DRAM device, Adaptive Double Device Data Correction (ADDDC) can

be activated to keep correcting errors after mapping out the failing device dynamically, at runtime. Fault prevention mechanisms such as Patrol Scrubbing and Demand Scrubbing detect and correct errors in the background and demand access, respectively. These actions provide software with early visibility for possible preventive measures such as page off-lining based on the rate of error corrections. Multiple Row Hammer prevention mechanisms mitigate possible issues that may arise from a high rate of access to a single DRAM row.

Memory mirroring is a technique which duplicates data across channels such that if failure is detected in one part of the mirror, the data can be obtained from the mirrored pair. The processors support two forms of mirroring: Full mirroring and Address Range mirroring. In full mirroring the entire memory is mirrored. While full mirroring is effective in protecting the entire memory, and is useful in certain situations, it comes with a cost since memory capacity is halved. Address Range Mirroring reduces this cost by allowing the Operating System to request the range of memory to be mirrored. In this manner, only critical parts of memory (such as Kernel data) can be protected with less capacity reduction. Protecting this part of memory with Address Range Mirroring has the additional benefit of protecting kernel/hypervisor state in a virtualized environment as it is particularly challenging to recover from uncorrected errors in kernel space (also see MCA Recovery in Section 4.2 Availability Features).

To enable customers' validation of their RAS related SW stack and error handling environment, a secure DRAM error injection mechanism is available. This mechanism allows the injection of corrected or uncorrected errors into DRAM based on a given system address while the system is booted in debug mode.

Table 4-1. Condensed List of Reliability Features supported by 4th Gen Intel® Xeon® Scalable Processors|

Subsystem	Reliability Feature[2]	Description
CPU	Error detection and correction	Capability to detect and correct errors across silicon and interconnects.
CPU	Advanced Error Detection and Correction (AEDC)	Improves fault coverage within the processor core.
Intel® Ultra Path Interconnect (Intel® UPI)	Link level retry and CRC	Intel UPI link Cyclical Redundancy Check (CRC) and dynamic retry mechanism.
DDR5 and HBM[3]	Persistent Fault Detection (PFD)	Memory controller's capability to detect persistent faults and enhance ECC correction and detection.
DDR5 and HBM	Error Correction Code (ECC) support	Sophisticated error correction and detection techniques. Support for 10x4, 5x8 and 9x4 DDR5 DIMMs and HBM memory.

2 Feature availability may vary by processor SKU and/or implementation/configuration options, additionally not all features, although being used, may be visible to end users, for details please contact your Intel representative or system integrator

3 High Bandwidth Memory (HBM) available on Intel® Xeon® CPU Max Series.

Subsystem	Reliability Feature[2]	Description
DDR5	Adaptive Double Device Data Correction (ADDDC)	A technique invoked upon impending device failure to prevent system failure.
DDR5	Command Address Parity and Retry	Error detection on the command/address bus coupled with a retry mechanism for correction.
DDR5	Write Data CRC and Retry	CRC protection for write data on the data bus coupled with a retry mechanism for correction.
DDR5 and HBM	Post Package Repair (PPR)	Capability to replace rows identified by the platform as failed rows with spare rows in DDR5 DIMMs or HBM.
DDR5 and HBM	Command/Address parity check and retry	Command/Address bus protection via parity. Transaction retry for recovery.
DDR5 and HBM	Demand and Patrol Scrubbing	Scrubbing (DRAM and HBM) to correct latent errors in memory and provide means for corrective action from Operating System (OS)/Virtual Machine Manager (VMM) or the platform.
HBM	Memory bank sparing	A sparing technique to replace a failing bank.
HBM	Partial cache line sparing	A technique to repair stuck single bit errors.
DDR5	Advanced Memory Test (AMT)	An enhanced memory test capability to check memory health.
PCIe[4]	Enhanced Downstream Port Containment (eDPC)	eDPC as per the PCIe specification.
Compute Express Link* (CXL*)[5] and PCIe	CXL and PCIe link CRC error check and retry	CXL/PCI specification compliant handling of Cyclical Redundancy Check (CRC) check and retry.
System	Core disable for FRB	Disable failed cores at boot time.
System	Poison mode for data error containment	Support for poison mode of operation to contain uncorrected data errors.
System	Viral mode for data error containment	Enhanced error containment mode of operation to improve error containment.

4 PCIe is registered trademark of PCI SIG.
5 CXL is registered trademark of CXL Consortium.

4.2 Availability Features

Availability is a measure of a system's capability to remain operational even in the presence of faults. A highly available system remains operational by employing a robust set of mechanisms such as disabling malfunctioning components, continuing to operate at a reduced capacity or predicting and preventing failures before they happen.

A resilient system is designed with a set of mechanisms that will let the system keep running in the face of errors. When a reboot is required, being able to quickly identify the cause of the fault, rectify it and bring the system back up in working order as quickly as possible is also important. In this sense, availability has dependencies on reliability and serviceability features.

Other fault prevention mechanisms such as processor Built-In Self-Test (BIST) also help identify failures before they show up in a running system. Processor BIST runs at boot time and any cores that fail BIST can be mapped out by the platform to keep the system healthy. Similarly, Intel UPI supports Dynamic Link Width Reduction where upon detecting a failed lane, it reduces the link width so that the system can operate without the failed lane.

However, no matter how good the correction capability and fault prediction of the system, uncorrectable errors can occur in a large deployment pool over an extended period. When they occur, a resilient server survives by containing the impact of the fault. Processors employ a feature called MCA recovery for this purpose. There are two forms of MCA Recovery: Execution path recovery and non-execution path recovery. In non-execution path recovery, the error is encountered outside of the execution path. The OS/hypervisor may carry out recovery by off-lining the page associated with the failure. Non-execution path recovery is important in that it gives the OS time to deal with the error condition before it gets to the consumption point.

Execution path recovery manages the case where the poisoned data reaches the consumption point. The error is presented to the OS/hypervisor in such a way that it can take one of three possible actions. First, determine if it can clear the condition that caused the error and if so, let the operation continue from where it was interrupted. Second, if that fails, terminate only the guest or application that encountered the error, thus avoiding a system crash and letting all other operations on the system continue. Finally, if neither of those two options are possible, it will bring down the system. Localized Machine Check Recovery avoids the broadcast of the machine check recovery event to all threads in the system, thus focusing attention from the OS/hypervisor only on the affected operation.

Other availability features such as Seamless firmware update, which allows the update of microcode and other firmware without requiring a reboot are discussed under the RAS Management section.

The following table describes availability specific features supported by the processors:

Table 4-2. Condensed List of Availability Features supported by 4th Gen Intel Xeon Scalable Processors

Subsystem	Availability Feature[6]	Description
CPU	Processor BIST	During the power-on initialization, the processor's built-in self-test engine provides the result of the self-test ran on its internal cache structure.
Intel UPI	Intel® UPI dynamic link width reduction	Dynamic Intel UPI link width reduction to half width to recover from full link width CRC failures.
DDR5 and HBM	Memory disable and map-out	Allows memory initialization and booting to OS even when fault occurs (Fault Resilient Boot).
DDR5	Address Range Memory mirroring	Mirroring only portions of memory critical to the system's operation.
CXL and PCIe	Hot-plug	Specification compliant managed and surprise hot-plug flows for removal/addition of a card at the system runtime.
PCIe	Link retraining and recovery	Ability to re-train the link at either smaller width or lower speed when corrected errors reach threshold.
System	Predictive Failure Analysis	OS or platform firmware preemptive action based on failure prediction algorithms.
System	MCA Recovery – Execution path	Software assisted recovery from uncorrectable data errors on the path of program execution.
System	MCA Recovery – Non-Execution path	Software assisted recovery from uncorrected data errors outside of program execution.
System	Local Machine Check (LMCE) Recovery	Software assisted recovery to only affected logical processor receiving corrupted data (poison).

4.3 Serviceability Features

The speed and ease by which a fault is diagnosed and repaired to bring the system back to operation is fundamental to the total cost of ownership and eventual availability of the system. A highly serviceable system must have methods to provide detailed error logs identifying where a failure occurred so that repairs can be performed with as little disruption as possible.

The processors provide prodigious amounts of error logs to identify failed or failing components. The memory sub-system provides detailed error logs that help identify where the failure occurred. Within the SoC detailed logs are made available to the OS and the platform via Machine Check Architecture (MCA)[3] logs, Advanced Error Reporting (AER), and platform specific logs that help with the diagnosis of failures. Furthermore, tools such as the Autonomous Crash

6 Feature availability may vary by processor SKU and/or implementation/configuration options, additionally not all features, although being used, may be visible to end users, for details please contact your Intel representative or system integrator

Dump[7] (Intel® ACD) utility make it easier to diagnose faults. It captures a rich set of debug state to accelerate diagnosis of errors and provides actions for customers to take. Intel works closely with customers in providing these tools and any hands-on help needed.

Error diagnosis may take place in either a Firmware First Mode (FFM) or out-of-band means from a Baseboard Management Controller (BMC). FFM mode and out-of-band management are described under RAS Management section.

The following table summarizes some of the serviceability features supported.

Table 4-3. Condensed List of Serviceability Features supported 4th Gen Intel Xeon Scalable Processors

Subsystem	Serviceability Feature[8]	Description
CPU	Time-out timer schemes	Transaction timeout schemes to signal error in event of non-terminating transaction.
CPU	Error reporting through MCA 2.0 (eMCA 2.0)	Enhanced error reporting to support Firmware First Model (FFM).
CPU	Error reporting (MCA, AER) – core, uncore, and IIO	Coherent domain and non-coherent domain ability to log and signal errors.
CPU	Out-of-band (OOB) Access to logs	Side-band access to RAS error reporting out-of-band through various protocols.
DDR5	Failed DIMM Isolation	Detailed error logs to help identify where failures occurred
CXL and PCIe	CXL/PCIe Error reporting via IOMCA	Capability to report CXL/PCIe errors via MCA.

7 Feature availability may vary by processor SKU and/or implementation/configuration options, additionally not all features, although being used, may be visible to end users, for details please contact your Intel representative or system integrator

8 Feature availability may vary by processor SKU and/or implementation/configuration options, additionally not all features, although being used, may be visible to end users, for details please contact your Intel representative or system integrator

5 RAS Management, Debug, and Telemetry

This section provides an overview of the RAS management, root cause analysis, and telemetry features.

5.1 Error Handling Modes

Traditionally there have been two methods for error handling in server platforms, namely Firmware-First mode (FFM) and OS-First (also known as OS Native Mode). The decision of which method to use is dependent on the system integrator's usage model. For most cloud usages at the system level, especially for Bare Metal-as-a-Service (BaaS) In-band and Out-of-Band (OOB) management are becoming the primary means for server RAS manageability, hence the processors provide various options to support a wide range of the desired customer usage models. BMC, with a dedicated network interface, is one of the most preferred ways of OOB management in the server platform.

5.1.1 In-Band Error Management

On Intel platforms, the preferred method for in-band RAS management is the Enhanced MCA (eMCA) Gen 2 mode, which is an optional mode that was first introduced in the Intel® Xeon® Processor E7-V3 family. This infrastructure provides an opportunity to deliver richer error logs of all severities (CE, UCNA, SRAR, UCE) to higher level software at the time of signaling. The eMCA Gen2 feature [4] provides enhanced error reporting to support various attributes that align with Firmware First Mode (FFM) of error handling.

Additionally, the processors support FFM as well as Native OS-based error handling in the IO domain allowing customers to design their systems based on their usage model.

5.1.2 Out-of-Band Error Management

Out-of-Band (OOB) RAS management is driven by management hardware on the platform (such as BMC), irrespective of the host CPU based error handling. In the processors, the support for this mode of management can be achieved in two ways. First option is to use the eMCA Gen2 mode, harvest the errors in the platform firmware (System Management Mode or SMM) and then push them to BMC.

If any RAS feature management is needed via BMC, it must be in coordination with SMM. The other option is to disable eMCA Gen2 mode and capture partial error information as well as manage some RAS features from OOB. The first option has an advantage of providing complete error telemetry of the platform while the second option provides limited telemetry but avoids platform firmware dependency. For the OOB management Intel provides access to a rich set of error logging and RAS policy registers that can be obtained from the

platform by an OOB agent such as BMC over the PECI interface. This allows OEMs, enterprise customers, and CSPs to manage some PCIe and memory RAS errors as well as monitor and gather error telemetry for several error sources.

5.2 Debug and Root Cause Analysis

In case of fatal errors such as IERR and MCERR conditions it is important to triage quickly, reliably, and precisely to minimize server downtime. Having efficient diagnosis reduces the Mean-Time-To-Debug (MTTD) and helps service the system as soon as possible. Intel provides a technology for crash data collection in the processors, known as Autonomous Crash Dump (Intel® ACD)[9].

5.2.1 In-Band Error Harvesting

MCA Banks and status registers can be collected in band. Crash Log in the processors offers the ability to collect CrashLog records in the absence of BMC, by harvesting the CrashLog record after a warm reset. CPU crash log and PCH crash log are independent of configuration, triggers, flows as well as content. PCH Crash Log is preserved over global reset so that the reset causes emanating from the PCH can be triaged post global reset.

5.2.2 Out-of-Band Error Harvesting

When the core is unable to access MCA Registers, OOB error harvesting is used to collect MCA Banks, status registers, and other debug state after a fault. The access is provided through JTAG and the PECI bus. Intel ACD is an OOB debug data collection technology that gathers a larger set of fault logs during system hang and catastrophic error conditions. Intel ACD requires BMC FW to be the agent that detects the fatal error condition by observing the pins such as CATERR# assertion and reads the critical system state, error log, and control information before system reset. This data is critical in accurately debugging the system as it is captured at the point of failure in a customer's fleet instead of reproducing the error scenario. The collected crash data is restructured in JSON format for further debug. Intel provides a decoding tool named CrashDump Summarizer which uses the internal decision trees to triage and root cause the error.

9 Feature availability may vary by processor SKU and/or implementation/configuration options, additionally not all features, although being used, may be visible to end users, for details please contact your Intel representative or system integrator.

4th Gen Intel Xeon Scalable processors improve the state-of-art by introducing Crash Log, which minimizes customer enabling requirements and provides customers with more structured data collection. It employs an active mechanism where disaggregated SoC Crash Log agents dump the fault state in local SRAM for later extraction. For security reasons, no user-data is captured as part of these logs. Crash Log technology provides a faster collection of debug state, and the ability to extract as a block unit. Intel Crash Log technology is a SoC initiated aggregation of the error telemetry and it is collected using different triggers such as IERR or MCERR. Crash Log is

collected as part of the ACD Firmware when available. An input selection structure for ACD allows Intel customers to configure some of the debug state to be collected.

5.2.3 Intel® At-Scale Debug

The Intel® At-Scale Debug (Intel® ASD) feature allows for the use of any host system to run a debug tool stack whilst connecting to a target system across the network. It allows customers to reduce debug complexity at the manufacturing, qualification, and production phases of hardware deployments. Intel ASD is designed to enable customer self-sufficiency at scale to perform critical data collection, minimize the need for the traditional use of (Intel® In-Target Probe (Intel® ITP) - which may require physically accessing systems, and allow greater flexibility for instrumentation, reproduction, and environmental conditions surrounding the debug process.

5.3 Intel® Platform Monitoring Technology

Intel® Platform Monitoring Technology (Intel® PMT) is a standardized way of exposing telemetry that was invented to deliver a shareable, unified interface for devices on both server and client platforms. It provides shared access between in-band and out-of-band interfaces and is highly configurable with various API options. Intel PMT is supported on the processors when coupled with BMC chip or other microcontroller supporting platform manageability and SoC connectivity over serial PECI or PCIe as well as Intel PMT service software integrated with BMC/micro-controller. Intel PMT is supported on the processors when coupled with BMC chip or other microcontroller supporting platform manageability and SoC connectivity over serial PECI or PCIe as well as Intel PMT Service software integrated with BMC/microcontroller. OS support for Intel PMT telemetry is also available on Linux*.

5.4 Seamless Firmware Update

Traditional system upgrade technologies negatively impact the server availability and service blackout time. In a virtualized environment, the virtual machine (VM) must continue to run at a specified SLA regardless of a firmware update. The objective of Seamless Firmware Update technology is to minimize the impact of system resets that are driven by firmware updates. And when reset is unavoidable, reduce the reset duration to the minimum possible [5].

Intel’s Seamless Firmware Update capability allows customers to update firmware components such as microcode, ACM and so forth. without affecting the workload and by maximizing uptime to meet SLAs. This feature was introduced in 3rd Generation Intel Xeon processors and has increased in scope in the processors with additions of SMM driver update infrastructure and OS transparent update control. Seamless Firmware Update employs various mechanisms such as eliminating the image staging. This essentially reduces firmware update time from multiple minutes to seconds, minimizing system downtime [6].

6 RAS Platform Application Engineering Support

A reliable data center platform must be architected with a RAS mindset. Many of Intel's RAS features are architecturally designed into the platform's hardware and software. Therefore, the customer RAS requirements form the foundation for Intel Xeon RAS architecture. Intel Xeon processor RAS enablement engineers work with system manufacturers throughout the Intel Xeon-based platform life cycle to devise an optimum RAS solution that meets each customer's unique needs.

RAS is a multi-faceted discipline, which combined with system deployment and error handling guidelines, can make a well-tuned RAS solution a daunting undertaking. Intel RAS engineering works to reduce RAS development challenges by enabling our system manufacturers to start planning RAS early in platform development cycles. Intel publishes a rich set of collaterals describing RAS architecture, register set, hardware signaling and BIOS/OS RAS flows. RAS engineering support continuously engages with OEM/CSP platform development teams through interactive deep-dive discussions, to provide RAS feature education while receiving feedback to guide customer specific adaptation.

Quick and precise testing of RAS features is critical to keep up with the contemporary accelerated development cycles, and the Intel Xeon processor RAS engineering team publishes a comprehensive set of RAS testing guides, debug tools, and a RAS integration and validation guide (IVG). This guide, generally referred to as the RAS IVG, serves as a getting-started manual for RAS validation by providing illustrated RAS unit-tests. Through-out an Intel Xeon-based platform lifecycle, the Intel RAS engineering team continuously works with system manufacturers and customers for the best experience on Intel Xeon-based servers to help them properly understand and optimize system RAS error handling.

Intel realizes that today's diverse cloud-scale data-center fleets have increased heterogeneous vendor, silicon, and ingredient interoperability requirements. For efficient cloud-scale fault management the RAS solutions need to be adopted and standardized for vendor agnostic end-to-end infrastructure monitoring and automation. Towards that goal, Intel is leading Open Compute Project (OCP) initiatives for developing standards-based error handling infrastructure requirements and specifications for platform and vendor agnostic error reporting. [7]

7 Acronyms and References

7.1 Acronyms

The following is a list of acronyms and their definitions.

Table 7-1. Acronyms

Acronym	Definition
ACM	Authenticated Code Module
AER	Advanced Error Reporting
API	Application Programming Interface
ASD	At-Scale-Debug
BIST	Built-In Self-Test
BMC	Baseboard Management Controller
CATERR	Catastrophic Error
CE	Corrected Error
CRC	Cyclical Redundancy Check
CXL	Compute Express Link
eMCA	Enhanced Machine Check Architecture
FFM	Firmware First Mode
FRB	Fault Resilient Boot
FRU	Field Replaceable Unit
FW	Firmware
HBM	High Bandwidth Memory
IIO	Integrated Input/Output
IVG	Integration and Validation Guide
JSON	JavaScript Object Notation - a lightweight data-interchange format
JTAG	Joint Test Action Group (interface)
LMCE	Local Machine Check Error
MCA	Machine Check Architecture
OCP	Open Compute Project
OEM	Original Equipment Manufacturer
OOB	Out-of Band
PCIe	PCI Express (Peripheral Component Interconnect Express)
PECI	Platform Event Controller Interface
PFA	Predictive Failure Analysis
SLA	Service Level Agreement
SoC	System on Chip

Acronym	Definition
UCE	Uncorrected Error
UCNA	Uncorrected No Action
VMM	Virtual Machine Manager

7.2 References

[1] 2022 ITIC Reports & Surveys – Information Technology Intelligence Consulting (itic-corp.com)

[2] Technical Overview of The 4th Gen Intel® Xeon® Scalable Processor Family

[3] Intel® 64 and IA-32 Architectures Software Developer's Manual

[4] Error Reporting Through eMCA Gen-2

[5] Platform Boot Time Reduction Enhancements (OCP Global Summit, 2022)

[6] Cloud Downtime Reduction via Firmware Update Innovations (OCP Global Summit, 2022)

[7] OCP Hardware Fault Management Infrastructure Requirements Proposal (OCP Global Summit, 2022)