Technical Disclosure Commons

Defensive Publications Series

March 2020

LINE RATE HARDWARE FLOW TELEMETRY ARCHITECTURE ON FIBRE CHANNEL APPLICATION-SPECIFIC INTEGRATED CIRCUIT

Harsha Bharadwaj

Vasuki H. A

Rajesh L. G

Recommended Citation

Bharadwaj, Harsha; A, Vasuki H.; and G, Rajesh L., "LINE RATE HARDWARE FLOW TELEMETRY ARCHITECTURE ON FIBRE CHANNEL APPLICATION-SPECIFIC INTEGRATED CIRCUIT", Technical Disclosure Commons, (March 27, 2020)
https://www.tdcommons.org/dpubs_series/3067

LINE RATE HARDWARE FLOW TELEMETRY ARCHITECTURE ON FIBRE CHANNEL APPLICATION-SPECIFIC INTEGRATED CIRCUIT

AUTHORS:
Harsha Bharadwaj
Vasuki H A
Rajesh L G

ABSTRACT

Techniques are provided herein for offering line rate flow telemetry on all ports of a Fibre Channel (FC) switch by implementing a bidirectional flow correlation engine inside an FC switching Application-Specific Integrated Circuit (ASIC). This enables a specialized set of flow analytics solutions to be implemented using machine learning models. These models may be trained with full flow visibility, including every outlier, and may have very high prediction accuracy. This may enable building of a switch integrated solution for this use case without involving external appliances.

DETAILED DESCRIPTION

Traditional Application-Specific Integrated Circuit (ASIC) flow metric collection architectures (e.g., NetFlow, Sflow, etc.) implement a flow cache or a flow table that collect flow marker packets unidirectionally and periodically ships it out of the box to an external flow collector (e.g., an Ethernet NetFlow collector external to the ASIC/switch) where flows are matched/completed and flow metrics (e.g., byte_count, flow anomalies, etc.) are computed. Small Computer System Interface (SCSI) and/or Non-Volatile Memory express (NVMe) analytics may be supported. Unlike flows in the Local Area Network (LAN) that are characterized by relatively long-lived sessions carrying high-volume data, a flow in a Fibre Channel (FC) Storage Area Network (SAN) corresponds to a single Input/Output (I/O) operation between a [Initiator, Target, Logical Unit Number (LUN)] tuple (ITL). Every I/O is a transaction that is completed in the context of an FC exchange and is extremely short-lived. For example, typical I/O sizes are 2K, 4K, or 8K. A 2K I/O consists of just three FC frames and completes in a few microseconds. Due to this flow characteristic, a flow-cache/table method is not suitable due to scale (e.g., the flow table quickly overflowing and missing flows) and practicality (e.g., an extremely fast external

FC flow collector) concerns. As such, an updated flow metrics collection architecture is desirable in the next-generation FC ASICs.

The previous generation FC ASIC provides for a flow collection architecture where the flow marker frames are "tapped" and "spanned" from the data path using Access Control List (ACL) based Switched Port Analyzer (SPAN) to an onboard dedicated Network Processing Unit (NPU) to perform bidirectional matching/correlation and flow metric computation in a software analytics engine. The symmetrical routing of I/O request/response frames to the same port is guaranteed due to the Source Identifier (SID), Destination Identifier (DID), and Exchange Identifier (OXID) based hashing used in Equal-Cost Multi-Path (ECMP) routing and port-channel link selection logic across all network switches.

However, this solution cannot scale for a line rate flow collection (e.g., for 48 ports on a line card) due to the limited compute capacity of the NPU and finite bandwidth for data transfer between the previous generation FC ASIC and the NPU. As a result, a port sampling scheme (e.g., flow metrics collected on a rotation basis per-port) is deployed beyond a certain threshold of total switched traffic from the ASIC. This results in data "dark spots" requiring interpolations and data inaccuracy concerns.

Accordingly, a line rate flow telemetry is required for two specific category of metrics: primary storage metrics (e.g., I/O Operations per Second (IOPS), throughput, inter-I/O gap, I/O size, I/O access pattern, etc.) and latency-related metrics (e.g., I/O completion time, storage data access delays, host-induced I/O delays, busy time, etc.).

With respect to primary storage metrics, the new-age all-flash arrays are extremely fast and service I/Os in one large burst. The bursty nature of I/O cannot be observed with a sampling scheme. An examination of every flow is required to identify these I/O microburst patterns. This can then be used to root-cause fabric congestion caused due to slow drain symptoms as a result of servicing bursty I/O requests. A timely identification and fixing of slow draining devices are critical for a healthy FC.

I/O size and access patterns for storage can help characterize unexplained slowdowns. The block storage devices are well behaved with similar sized I/Os with the same access patterns. With full visibility, the occurrence of unexpected I/O sizes or random access patterns due to application misconfiguration or misbehavior resulting in a "blending" effect on storage media can be quickly identified.

With respect to latency-related metrics, a high I/O latency with no I/O or transport level errors are usually the most difficult problems to root-cause and troubleshoot. Full visibility to I/O latency helps closely monitor application performance and provide real-time feedback the instant abnormalities are seen. Storage protocols such as NVMe are designed for ultra-low latency. An accurate determination of I/O latency is important for performance baselining and deviation tracking.

Traditional disk-based storage devices are known to exhibit bimodal/multimodal latency distribution patterns for I/Os due to a fast cache that front-ends the disk. All I/Os that can be serviced by the cache have distinctly lower latency compared to the I/Os that miss the cache and seek data on the disk.

The aforementioned latency patterns can be missed when data is sampled and interpolated over large time intervals. Only a line rate flow metric extraction combined with a high-frequency (e.g., 500 ms) export enables catering to these use-cases. Further, nanosecond-level accuracy with time metrics is possible only with an ASIC implementation.

The current generation FC ASIC described herein solves this problem by implementing an on-chip analytics engine that runs in parallel to the switching data path without adding any additional data path latency. The current generation FC ASIC ingress/egress parser blocks identify SCSI and NVMe flow "Request" and "Response" frames, and copy out frame headers to an Analytics Metrics Collection (AMC) block in the ASIC. The AMC block maintains state for every I/O and performs bidirectional correlation and computation of flow metrics at line rates simultaneously for all the ports of the ASIC. Since a large number of flow metrics (e.g., 70) are computed, it is also maintained in an on-chip flow database organized per-port, per-IT/ITL for quick retrieval of data. The flow database is set to be periodically pushed to an NPU connected to the current generation FC ASIC and eventually to a software process on the main switch Central Processing Unit (CPU).

With AMC, the FC flows are bidirectionally tracked within the ASIC by maintaining the state per I/O within the ASIC for all ports at line rate. A flow context is created upon obtaining an I/O request frame and is matched with intermediate/response frames to perform analysis and extract a variety of metrics. The state is cleaned up upon observing a response frame or a timeout. There is not necessarily a need for an external flow collector and analyzer with this architecture. An external entity can directly obtain the flow metrics for each ITL. AMC not only performs NetFlow cache equivalent functionality (e.g., flow identification and accounting), but also the external NetFlow collector equivalent functionality (e.g., flow completion, bidirectional correlation, flow analysis, etc.) inside the ASIC.

A multi-stage data collection architecture is provided with fine balance and alignment between hardware and software resources to ensure that every I/O is examined. The current generation FC ASIC flow database is optimally sized to sustain up to thirty seconds worth of flow metrics before the data is flushed to the NPU (after which the flow counters might overflow) for all ports running at line rate.

The NPU that is connected to multiple current generation FC ASICs periodically collects the ASIC flow database snapshots to aggregate a per-line card flow database with metric storage capability for a small fixed number of ITLs (e.g., 20K). The NPU not only acts as a line card level flow database aggregator, but also provides flexibility and programmability to support any future needs such as support flow telemetry for new protocols (e.g., Key Value extension for NVMe (NVMe-KV), byte-addressable NVMe, etc.) or flow telemetry for new type of non-read/write I/Os.

The supervisor hosts a super flow database that is an aggregate of all the line card flow databases to provide a per-switch flow database with metrics storage capability for a large number of ITLs (e.g., 100K). This framework thus enables offering of line-rate flow metrics for 100% I/O monitoring.

To cater to the cases where exporting data from the supervisor may be too slow, a fast path (e.g., every 100 ms) raw format streaming of flow data from a front panel Ethernet port on the line card is also provided. The AMC can append a programmable User Datagram Protocol (UDP) / Internet Protocol (IP) header to flow data encoded Ethernet frames forwarded out of the port.

Figure 1 below illustrates the architecture evolution.

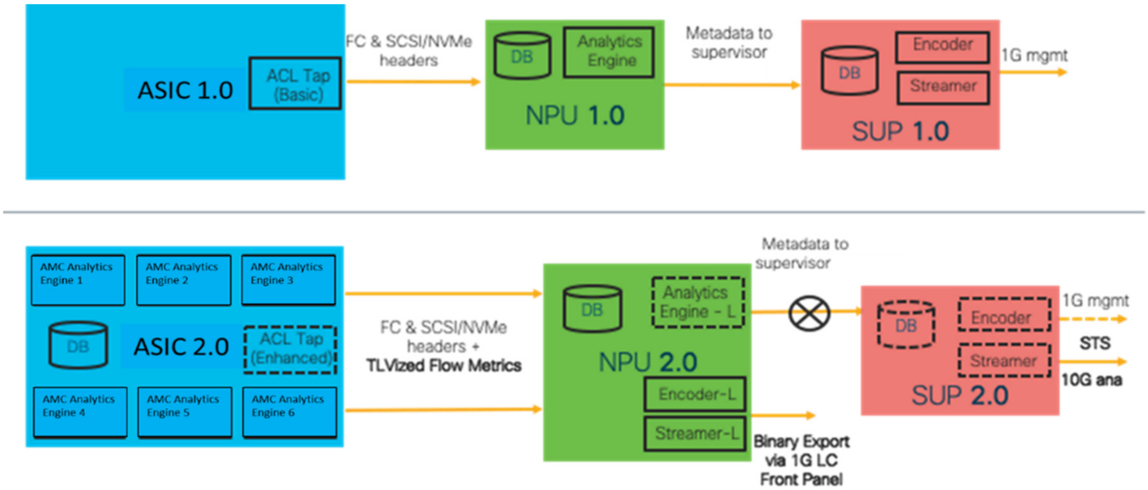

Figure 1

The current generation FC ASIC AMC block may include a line rate analytics engine that supports full rate analytics. Power saving architectures for lookup tables are used. Two custom multi-bucket hash tables that provide high utilization may be used: up to 24K ITLs and up to 960 open exchanges at any time. They may be occupancy-based and hash-addressable using multiple polynomials to maximize utilization. Type-Length-Values (TLVs) for the tables may be packetized for export at programmable times (e.g., greater than or equal to 500ms to less than or equal to 30 seconds).

"IT" mode data tracking may be performed when the ITL table capacity is exceeded.

ITO table entries may be dynamically created and deleted throughout the life of the I/O.

A timer may be provided to clear an incomplete/timeout I/O.

If there are no outstanding I/Os, the ITLs from the ITL table may be opportunistic cleared during export.

Every packet may be timestamped at AMC entry as close as possible to the packet data path to accurately compute different time related metrics with nanosecond accuracy.

In one example, there are no rate/average metrics to avoid compute-intensive or high gate count division operations.

Instead, totals and count/time may be accumulated for external computation over larger time intervals.

A programmable Linear-Feedback Shift Register (LFSR) may pseudo-randomly sample I/Os for binning of critical time metrics in the NPU to characterize distributions via histograms.

Figure 2 below illustrates a block diagram of the AMC computation logic.

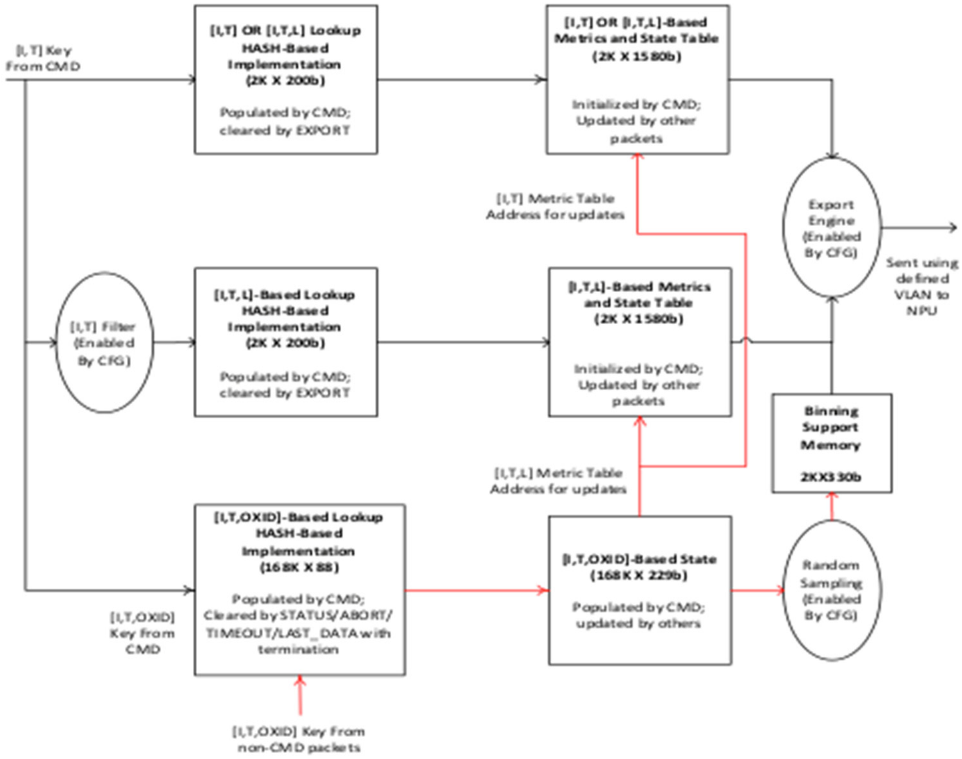

Figure 2

When combined with traditional switch interface metric streaming, line rate FC flow telemetry streaming enables building a suite of analytics applications hosted on a big-data-ready SAN Data Center Network Manager (DCNM-SAN). The DCNM-SAN receives flow telemetry information from switches. These applications could cater to a variety of specialized use cases such as real-time and proactive monitoring of application I/O performance, baselining storage device performance, infrastructure optimization recommendations, anomaly detection, etc. These applications may be real time and provide 100% flow visibility.

While this implementation is specific to FC, it could also be extended to telemetry blocks of Ethernet ASICs. It can be used when Ethernet is transporting any block storage protocol (e.g., FC over Ethernet (FCoE), Internet SCSI (iSCSI), etc.). The read and write I/O constructs are the same, independent of the underlying transport.

The AMC conserves significant front panel port bandwidth by sending fully prepared flow metrics. This becomes even more profound for short lived flows. It also uses aggregation and export to provide a non-sampled full visibility silicon solution using moderate resources. The AMC constantly aggregates and maintains the minimum and maximum for all I/O metrics for up to 48K ITL tuples per line card, thereby ensuring that no information is lost. A flexible frequency (e.g., ranging from milliseconds to tens of seconds) of export of this ready-made information from the ASIC to the CPU (software) allows for selection of an export frequency based on metrics accuracy, outlier visibility needs, and CPU processing capabilities. The software on the CPU stores all the metrics for the long term. The ASIC consumes a constant amount of resources to compute and maintain all the I/O metrics irrespective of the total traffic switched and CPU export frequency.

The analytic functions mainly involve computing a variety of flow metrics (e.g., I/O metrics). Appendix 1 includes metrics tables for the current generation FC AMC. In addition to computing metrics for typical reads/writes, the analytic functions may also address several variances of I/Os for both SCSI and NVMe storage protocols (e.g., first burst writes, first burst write with discard (NVMe only), optimized reads, multi-sequenced writes, etc.).

In summary, techniques are provided herein for offering line rate flow telemetry on all ports of a FC switch by implementing a bidirectional flow correlation engine inside an FC switching ASIC. This enables a specialized set of flow analytics solutions to be implemented using machine learning models. These models may be trained with full flow visibility, including every outlier, and may have very high prediction accuracy. This may enable building of a switch integrated solution for this use case without involving external appliances.

Appendix 1 - List of Metrics

Command (CMD) - Based Metrics (updated with every I/O CMD packet)

Metric	Size
Total Read-IO Count	28b
Total Write-IO Count	28b
Total Sequential Read-IO Count	28b
Total Sequential Write-IO Count	28b
Total Read-IO Bytes	28b
Total Write-IO Bytes	28b
Read-IO-Size Min	23b
Read-IO-Size Max	23b
Write-IO-Size Min	23b
Write-IO-Size Max	23b
Read-IO Intergap Time Min	32b
Read-IO Intergap Time Max	32b
Write-IO Intergap Time Min	32b
Write-IO Intergap Time Min	32b
Total Read-IO Intergap time	34b
Total Write-IO Intergap time	32b

Initiate I/O Packet Based Metrics

Metric	Size
Read-IO Initiation Time Min	32b
Read-IO Initiation Time Max	32b
Write-IO Initiation Time Min	32b

Write-IO Initiation Time Max	32b
Write-IO Host Delay Time Min	32b
Write-IO Host Delay Time Max	32b
Write-IO Array Delay Time Max	32b
Total Read-IO Initiation Time	34b
Total Write-IO Initiation Time	34b
Total Write-IO Host Delay Time	34b
Total Write-IO Array Delay Time	34b
Total Write-IO First Burst Count	34b

I/O Status Packet Based Metrics

Metric	Size
Read-IO Completion Time Min	32b
Read-IO Completion Time Max	32b
Write-IO Completion Time Min	32b
Write-IO Completion Time Max	32b
Write-IO Sequences Min	16b
Write-IO Sequences Max	16b
Total Read-IO Completion Time	34b
Total Write-IO Completion Time	34b
Total Write-IO Sequences Count	34b
Total Busy Period	34b
Read_Fault_Counter0	6b
Read_Fault_Counter1	6b
Read_Fault_Counter2	6b
Read_Fault_Counter3	6b
Read_Fault_Counter4	6b

Write_Fault_Counter0	6b
Write_Fault_Counter1	6b
Write_Fault_Counter2	6b
Write_Fault_Counter3	6b
Write_Fault_Counter4	6b

Miscellaneous Metrics

Metric	Size
Peak Outstanding Read-IO Count	16b
Peak Outstanding Write-IO Count	16b
Read-IO Aborts	8b
Write-IO Aborts	8b
Read-IO Failures	8b
Write-IO Failures	8b
Read-IO Timeouts	8b
Write-IO Timeouts	8b