May 2022

Using Low-level Telemetry in Cloud Platforms to Mitigate Latency Risks for High-Performance Computing

Radu Iorga

Aurélien Legrand

Using Low-level Telemetry in Cloud Platforms to Mitigate Latency Risks for High-Performance Computing

ABSTRACT

Latency minimization is critical to high-performance computing (HPC). Network monitoring tools that rely on out-of-band data to control latency cannot assist latency-sensitive network workloads in real time. This disclosure describes techniques that combine in-band network telemetry (INT) with the software-defined network (SDN) controller used by the cloud platform to mitigate HPC latency. INT gathers hardware-level information about buffer and queue utilization. Such information is used by the cloud SDN controller to make changes to the virtual environment. The SDN controller can directly affect decisions of the HPC master node relating to the assignment of tasks to worker nodes. The techniques leverage the deep, hardware-level information about potential latency issues signaled by buffer accumulations to inform cloud-HPC scheduling algorithms.

KEYWORDS

- High-Performance Computing
- Cluster computing
- Network telemetry
- Software-defined network (SDN)
- Spine-leaf topology
- SDN controller
- HPC master
- Congestion control
- Machine learning model
- Buffer utilization
- Queue occupancy
- Network monitoring tool
- Latency-sensitive application
- Top-of-the-rack (ToR) switch
- Virtual machine

BACKGROUND

From a network standpoint, a major challenge in high-performance computing (HPC) is latency minimization. Some HPC applications, e.g., weather-related applications, are sensitive to latency [1]. Currently, network monitoring tools rely on out-of-band data. However, such tools cannot assist latency-sensitive network workloads in real-time.

Traditional low-latency techniques in HPC, e.g., InfiniBand [3], are expensive, compared, e.g., to gigabit ethernet. Moreover, in cloud environments, there is very little InfiniBand support. Another way of minimizing latency in cloud environments is to group working nodes and their associated virtual machines as closely as possible, ideally in the same rack connected by a top-of-the-rack (ToR) switch. However, even in this configuration, depending on traffic patterns, the ToR can become congested and latency can increase.

An existing technique analyzes buffer utilization by extracting ASIC information [4]. However, this technique requires an inordinately large amount of data for each network node, resulting in very high computational expense. Also, it is difficult to correlate buffers with physical ports.

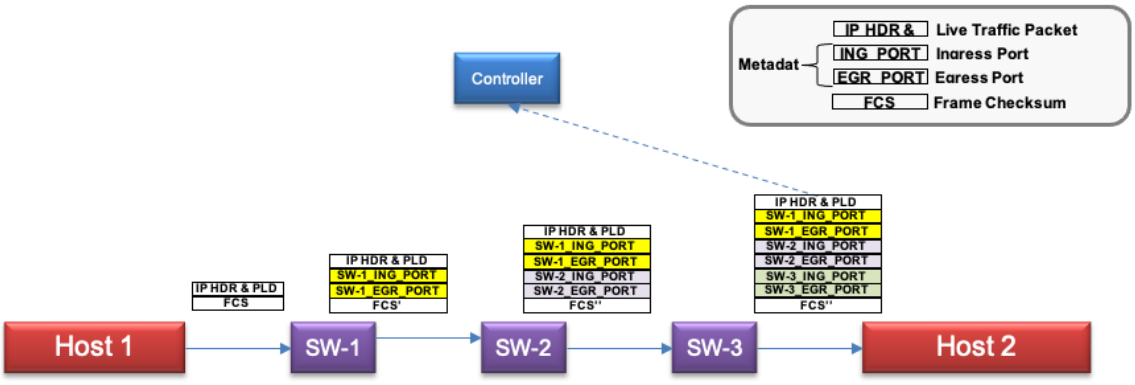

Fig. 1: High-Level view of the INT protocol

Inband Network Telemetry (INT) is a technology that can offer in-depth information about low-level latency. The INT protocol is illustrated in Fig. 1 and is described below.

a. Host1 sends traffic to Host2.
b. Switch-1 (SW1) samples traffic from the original stream, e.g., makes a copy without modifying the original packet. This ensures that the INT packet follows the exact same path as the original packet.
c. SW1 adds metadata to the INT packet and marks the packet as being INT. The metadata includes such as node ID, ingress port and timestamp (in nanoseconds), egress port, timestamp, link utilization, queue occupancy, buffer occupancy, etc.
d. Upon seeing an incoming packet marked as INT, switch-2 (SW2) adds its own metadata.
e. Upon seeing an incoming packet marked as INT, switch-3 (SW3) adds its own metadata. SW3 being the last hop before the host, it does not forward the packet; rather, it redirects it to a controller for analysis. The analysis itself is out of scope for the INT protocol. Note that the last hop switch before the host is known (for example the final leaf switch in a spine/leaf topology).

Information about buffers and queues can be gathered even if the collected information does not apply to the ingress/egress ports of the current flow. In-depth information about the accumulation of buffers inside the network equipment can be valuable to latency sensitive HPC applications.

DESCRIPTION

This disclosure describes techniques that combine in-band network telemetry (INT) [2] with the software-defined network (SDN) controller used by the cloud platform to mitigate HPC latency. INT gathers hardware-level information about buffer and queue utilization. Such information is used by the cloud SDN controller to make subtle changes to the virtual

environment. The changes to the SDN layer only affect latency-sensitive HPC workloads. The SDN controller can directly affect decisions of the HPC master node relating to the assignment of tasks to worker nodes. Effectively, the techniques leverage the deep, hardware-level information about potential latency issues signaled by buffer accumulations to inform cloud-HPC scheduling algorithms.

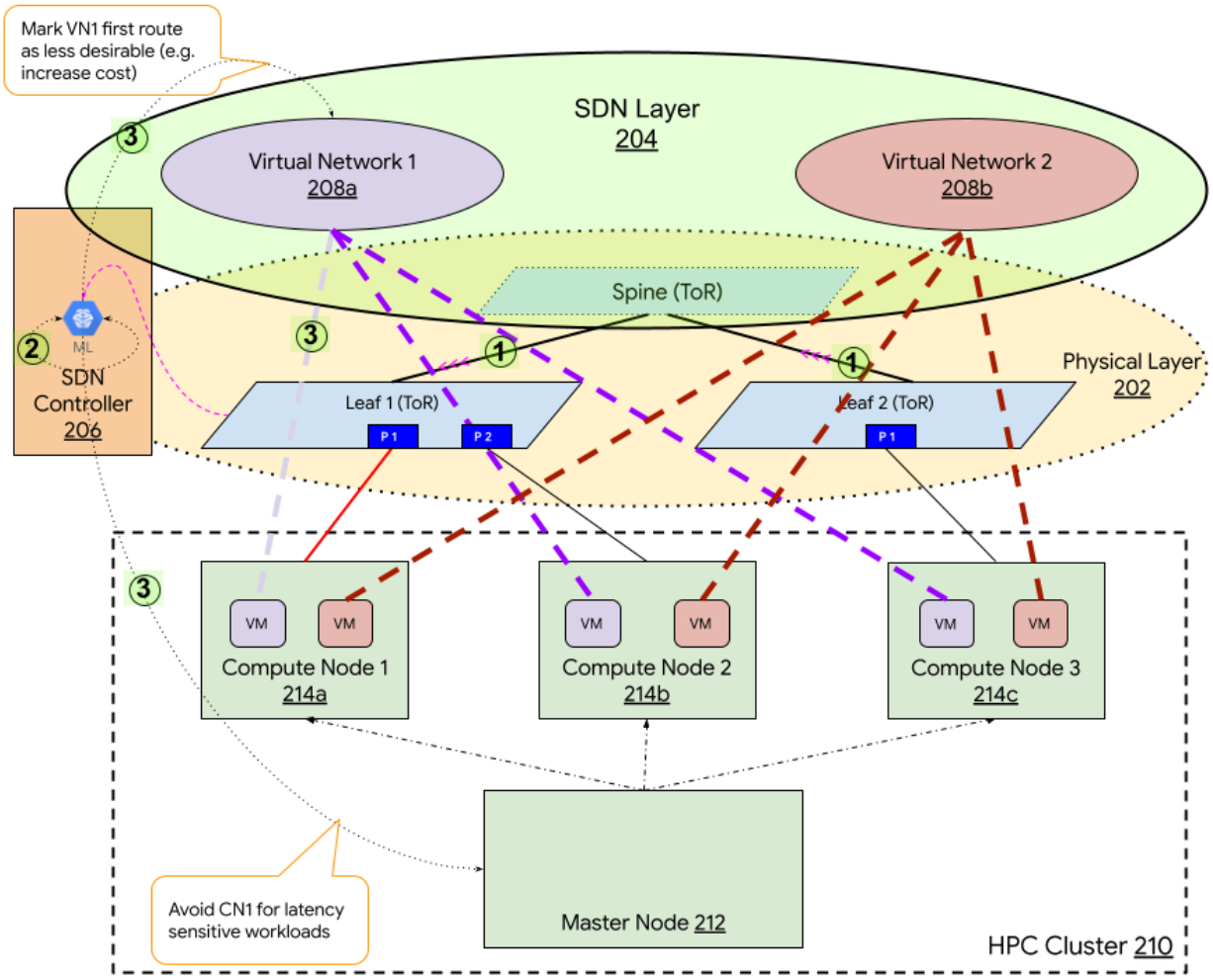

Fig. 2: Cloud HPC architecture with INT support

Fig. 2 illustrates an example of cloud HPC architecture with INT support. Hardware equipment sits in a physical network (or layer, 202), which also runs the INT protocol. The example illustrates a spine-leaf topology, common in data centers, although the techniques

described herein are applicable to any topology. Over the physical network is an overlay (or SDN) layer (204), where the SDN controller (206) creates virtual networks (208a-b). The SDN controller sits between the two layers and receives information from the physical layer. Correlating to Fig. 1, the SDN controller is also the INT controller where node metadata is collected. An HPC cluster (210) comprises a master node (212) and compute nodes (214a-c) where the workload virtual machines (VMs) are running. The compute nodes can host VMs from several different virtual networks, and it is not necessary that all VMs are running HPC latency-sensitive jobs.

In the cloud clusters that run HPC workloads, continuous flows are started along with the associated INT ❶. Depending on the topology the flows cover network nodes serving HPC workloads. Leaf 1 ToR sends metadata to the SDN controller. Inside the SDN controller there can be several ways to analyze the metadata ❷, e.g.,

a. A simple checking mechanism for congestion.
b. A machine learning (ML) model that predicts latency based on historical values of the buffers and queue utilization. This is unlike a simple anomaly detection because, for example, one short spike in buffer utilization is no reason for latency to necessarily increase. The ML model outputs a latency probability vector as illustrated in the example Table 1.

Physical Port	10-100 ns	100-1000 ns	>1ms
P1	90%	80%	60%
P2	70%	60%	40%

Table 1: Example latency probability vectors at ports P1 and P2 predicted by a machine learning model. In this example, there is a higher probability of congestion at Port P1, and a lesser probability of congestion at Port P2

Once latency increase is detected or predicted (e.g., in port P1 of Table 1), several latency-ameliorating actions can take place ❸, for example,

a. Notify the HPC master that P1 of ToR1 is experiencing (or is expected to experience) congestion and increased latency. If the HPC Master has working jobs on compute node 1, it can attempt to move the workload to an unaffected compute node. The master node can also move the workload VM to compute node 2, provided that port P2 of Leaf1 is unaffected by the present or anticipated congestion.
b. The SDN controller can increase the cost of the virtual link using port1 P1 of Leaf1 such that the HPC master does not schedule latency-sensitive jobs on compute node 1. To ensure that non-sensitive scheduling remains unaffected, a latency cost can be imposed, as illustrated in the example of Table 2.

Virt Link ID	Virt Network ID	Physical link	Cost	Latency cost
1	1	L1/P1	100	1000
2	1	L1/P2	100	500
1	2	L1/P1	100	1000

Table 2: Example of latency costs

The imposition of a latency cost causes the HPC scheduling algorithm to consider the latency of a virtual path. Based on the vector of latency probabilities, costs can be adapted such that the HPC master can schedule to the least risky compute node. For example, if virtual network 2 is not hosting a latency-sensitive workload, nothing changes for virtual network 2. If

port P2 of Leaf1 is unaffected by the actual/predicted congestion nothing changes for port P2 of Leaf1 (this is reflected in the latency cost).

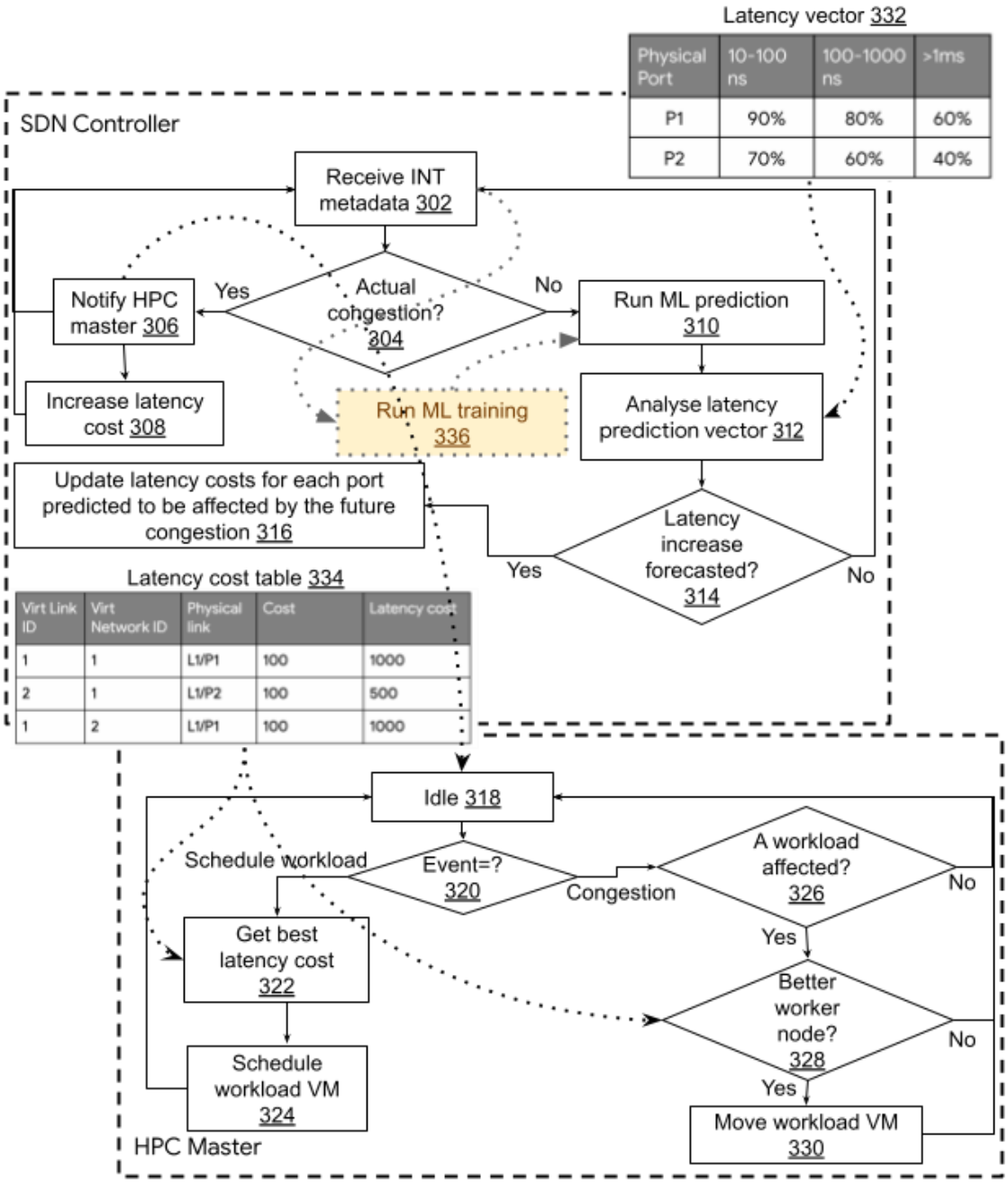

Fig. 3: Workflows at the SDN controller and at the HPC master node

Fig. 3 summarizes the workflows at the SDN controller and at the HPC master node. At the SDN controller, INT metadata (302) is used to train (offline and/or online) a machine

learning (ML) model (336). During operation, INT metadata is tested to detect congestion (304). If congestion is detected, the HPC master node is notified (306) and the latency cost is increased (308). If no congestion is detected, the ML predictor is run (310). The result of ML prediction, e.g., a latency vector (332), is analyzed (312). If an increased latency is predicted (314), the latency costs in the latency cost table (334) are updated for ports predicted to be affected by congestion in the near future (316).

The workflow at the HPC master node is idle (318) until it receives a notification from the SDN controller. Depending on the event (320), the HPC master node schedules the workload by getting the best latency cost (322) from the latency cost table and by selecting an appropriate virtual machine (324). If congestion is detected at the SDN controller, the HPC master node identifies affected workloads (326). If there are any affected workloads, a better worker node is sought (328) from the latency cost table, and, if a better worker node is found, the workload is moved to the appropriate virtual machine (330).

CONCLUSION

This disclosure describes techniques that combine in-band network telemetry (INT) with the software-defined network (SDN) controller used by the cloud platform to mitigate HPC latency. INT gathers hardware-level information about buffer and queue utilization. Such information is used by the cloud SDN controller to make changes to the virtual environment. The SDN controller can directly affect decisions of the HPC master node relating to the assignment of tasks to worker nodes. The techniques leverage the deep, hardware-level information about potential latency issues signaled by buffer accumulations to inform cloud-HPC scheduling algorithms.

REFERENCES

[1] “Interconnect Analysis: 10GigE and InfiniBand in High Performance Computing” available online at https://www.hpcadvisorycouncil.com/pdf/IB_and_10GigE_in_HPC.pdf accessed Apr 26, 2022.

[2] “In-band Network Telemetry (INT) Dataplane Specification” available online at https://p4.org/p4-spec/docs/INT_v2_1.pdf accessed Apr 26, 2022.

[3] “InfiniBand - Wikipedia” available online at https://en.wikipedia.org/wiki/InfiniBand accessed Apr 26, 2022.

[4] “BroadView” available online at https://github.com/Broadcom-Switch/BroadView-Instrumentation/blob/master/BroadView%20Instrumentation%20Specification.pdf accessed Apr 26, 2022.

[5] Jamaliannasrabadi, Saba. "High performance computing as a service in the cloud using software-defined networking." PhD diss., Bowling Green State University, 2015.

[6] Pelle, István, Francesco Paolucci, Balázs Sonkoly, and Filippo Cugini. "Latency-sensitive edge/cloud serverless dynamic deployment over telemetry-based packet-optical network." *IEEE Journal on Selected Areas in Communications* 39, no. 9 (2021): 2849-2863.