Technology Guide

Intel® Dynamic Load Balancer (Intel® DLB) - High-Precision Network Rate Limiting with Intel® DLB

Authors

Dong Wang

Yunfeng Bi

Niall McDonnell

James Zheng

Intel Corporation

Hengyang Xu

Tonghao Fang

Wenbin Gao

Frank Pang

Yvonne Li

Tencent

1 Introduction

As a leading cloud service provider, Tencent Cloud* provides high-performance, enterprise-class network services for users around the globe. The public cloud demands extremely high-quality service standards. The allocation of compute, memory, network, and storage resources must meet the standards listed in the service-level agreement (SLA), which in turn, impacts the end -user experience. To satisfy the SLA, cloud service providers reduce operating expenses by employing previous resources and minimizing the costs of additional resources. Tencent Cloud, in close association with Intel, can optimize network resource scheduling and allocation based on an innovative and integrated software and hardware approach using the Tencent Gateway (TGW) and Star Lake Lab, Tencent's dedicated hardware engineering laboratory, to leverage the technical advantages of TGW in the network field.

The commonly used method to allocate network resources is to limit the bandwidth of each user, and to maintain user concurrency control and requests at the gateway. This protects the system from congestion caused by request rate overload. The token bucket algorithm is a common rate-limiting mechanism, where tokens are replenished at a set rate and consumed by user requests, with the request being rejected if there are insufficient tokens. However, in the case of multi-core processors, concurrent operations on shared token buckets need to be atomic. Therefore, the software token bucket method for multi-core processors uses a lock to safeguard the token bucket. Because there is a high probability of the lock degrading forwarding performance, some developers adopt a lightweight lock scheme to amortize the cost of the locking.

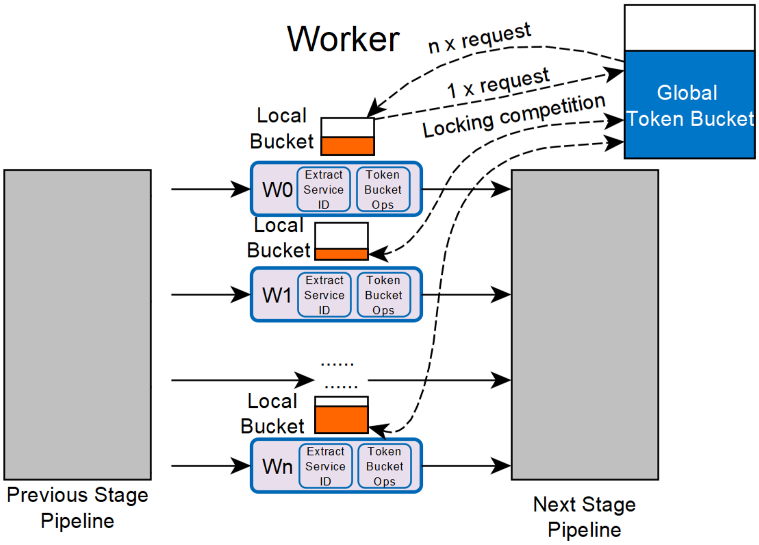


Figure 1. Lightweight Lock Scheme for Rate Limiting

There is another way to remove the lock by distributing packets on the base of service ID using the Intel® Ethernet Flow Director like technology. In this method, every core's load may be not equal because service numbers in network flow and each service's traffic are changed time to time. When a core is overloaded, packets cannot be received in time then dropped. Therefore, this method is not widely used.

This paper examines common lock optimizations for the token bucket in network bandwidth rate-limiting, analyzing their precision shortcomings. Then the paper describes the lockless token bucket approach based on Intel® Dynamic Load Balancer (Intel® DLB) integrated with 4th Gen Intel® Xeon® Scalable processors. Theories and comparative testing demonstrate the advantages of the lockless token bucket approach compared with existing lock optimizations. Lightweight Lock for Rate Limiting.

When multiple processor cores concurrently perform rate limiting on a network data stream, several cores may lock the same token bucket at the same time, at which point, one core can acquire ownership of the token bucket. The resulting lock competition is the main cause of performance degradation. Developers can reduce the probability of lock competition and performance impact on by changing the way token buckets are used and by using certain algorithms. This method is called a lightweight lock. See Figure 1.

This document is part of the Network Transformation Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	November 2022	Initial release.
002	December 2022	Minor updates made in the Abbreviations table.
003	January 2023	Updated for public distribution on Intel Network Builders.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
API	Application Programming Interface
CPU	Central Processing Unit
DIP	Destination IP
DPDK	Data Plane Development Kit
DUT	Device under test
Intel® DLB	Intel® Dynamic Load Balancer
IOV	Input/output Virtualization
LTS	Long Term Support
QSFP	Quad Small Form-factor Pluggable
SDK	Software Development Kit
SLA	Service Level Agreement
TGW	Tencent Gateway

1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
Intel® DLB Programming Guide	613545 NDA material. Contact you Intel representative.
Intel® DLB Software Development Kit	686372 Contact you Intel representative.
DPDK Eventdev (Intel® DLB) Development Guide	http://doc.dpdk.org/guides/eventdevs/dlb2.html

2 Overview

The lightweight lock scheme uses local token buckets in each processor core, backed by a global token bucket. The global bucket refills tokens at a set rate, and the local buckets prefetch tokens from the global bucket in batches to amortize the cost; the tokens are then eventually consumed from the local bucket by user requests.

In the token generation to token consumption process, only the prefetch operations from the global bucket to the local buckets need to be atomic, and the number of prefetches per packet is greater than the actual number of tokens consumed per packet. When processing the same number of messages, the number of times the token bucket is locked in the lightweight locking technique is obviously lower than that in the traditional single global token bucket technique. Therefore, with the increase in the number of processor cores, the lightweight lock for rate limiting can reduce the contention and deliver better performance.

2.1 Challenges Addressed

The lightweight lock scheme contains two key parameters:

- The token-generation rate of the global token bucket, that is, the target rate after the limiting rate.
- Batch size, the number of tokens prefetched from the global bucket when there are insufficient tokens in the local bucket.

When the global token bucket generates tokens at a low rate, a situation arises wherein the number of tokens generated within a unit time cannot meet all the batch prefetch requests of the local token bucket. A local token bucket that cannot be replenished causes messages to be discarded because there are insufficient tokens. However, there may still be unconsumed tokens in other local token buckets, and these discarded messages do not exceed the rate limit, resulting in the limited rate being lower than the target rate.

For this reason, a rate fluctuation can occur after the limiting rate, and so the focus must be on ensuring accuracy when optimizing rate limiting. See Figure 2.

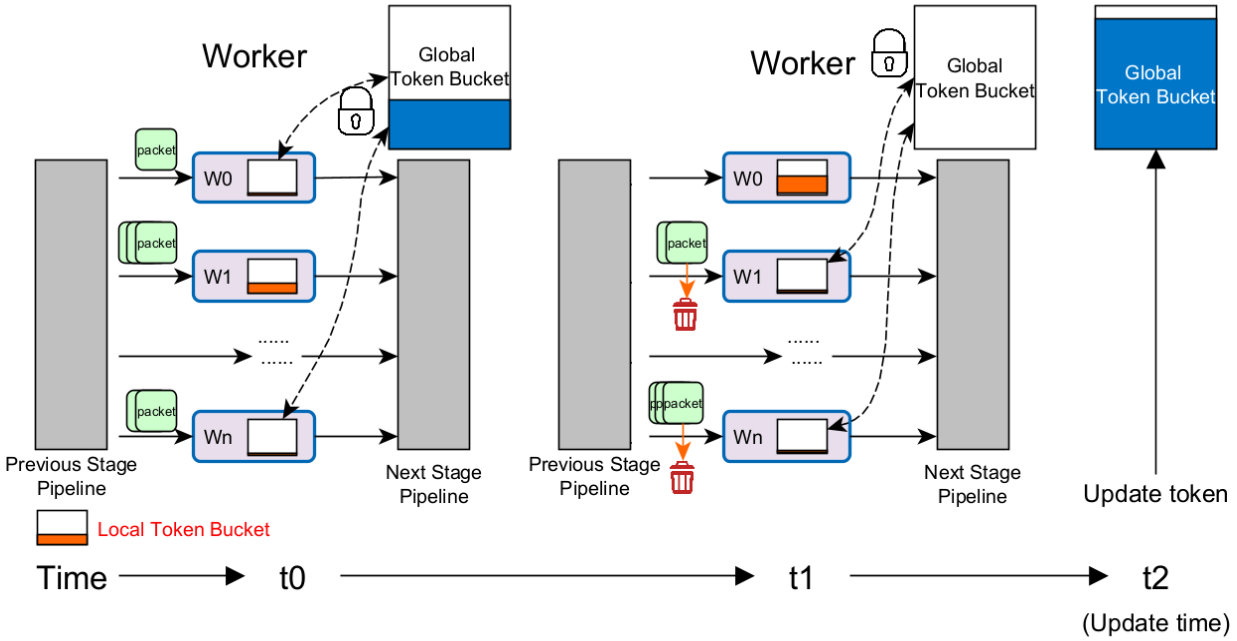


Figure 2. Fluctuation of Limiting Rate Caused by Lightweight Lock

2.2 Technology Description

2.2.1 Intel® DLB Technology for Lockless Rate Limiting

This section describes Intel® Dynamic Load Balancing (Intel® DLB) for lockless rate limiting.

Thanks to technological advances, every subsequent generation of the CPU has significantly more cores than the previous generation. Taking full advantage of multiple cores requires better software concurrency, which poses a huge challenge to software optimization. To resolve this, Intel® DLB technology is introduced in the 4th Gen Intel® Xeon® Scalable processors, to effectively address the performance challenges of highly concurrent software architectures.

Intel® DLB is a hardware queue manager integrated into the CPU. The software interacts with Intel® DLB through enqueuing and dequeuing—the enqueuing side is called the producer and the dequeuing side is called the consumer.

Intel® DLB has two principal features—dynamic processing and load balancing. Load balancing aims to solve the problem of unbalanced processor core load caused by uneven distribution of processing data among processor cores. Unlike the statical scheduling algorithms used in some software solutions, while distributing processing data, Intel® DLB dynamically selects the most suitable core based on each processor core load.

To ensure dynamic processing, Intel® DLB provides four queue models to meet the needs of different application scenarios:

Direct Queue: For multiple producers but only one consumer. No load balancing occurs.

Unordered Queue: For multiple producers and consumers.

The order of tasks is not important, and each task is assigned to the processor core with the currently lowest load.

Ordered Queue: For multiple producers and consumers, and the order of tasks is important. When multiple tasks are processed by multiple processor cores, they must be rearranged in the original order.

Atomic Queue: For multiple producers and consumers, where tasks are grouped according to certain rules. These tasks are processed using the same set of resources and the order of tasks within the same group is important.

See Figure 3.

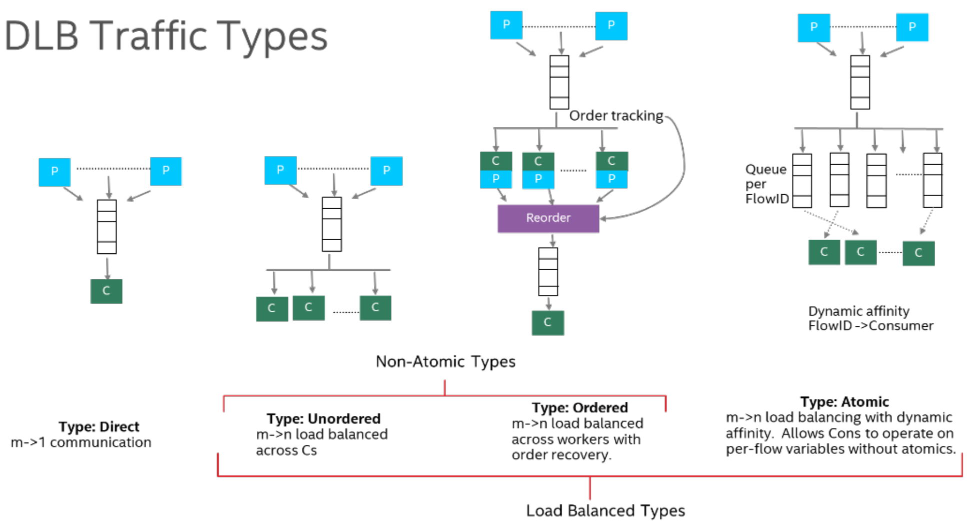


Figure 3. Four Queue Models of Intel® DLB

As described above, existing methods to optimize the performance of rate limiting focus on reducing the cost of the locking, and consequently lead to precision problems. Another approach is to use a lockless method. Such a scheme can be realized by issuing specific rules to the network adapter or by scheduling network messages in the same stream to the same processor core according to predetermined software algorithms to access the same token bucket on the same processor core. The flaw in these approaches is that the message scheduling rules are static and cannot be dynamically adjusted according to the load of the processor core. Therefore, some processor cores may be overloaded due to sudden network traffic, resulting in packet loss.

Is there a way to remove the lock protecting the global token bucket while ensuring multi-core load balancing in multi-core processors?

The Intel® DLB atomic queue can perform lockless rate limiting in multi-core scenarios.

By grouping network messages to be processed according to the rate limit network data stream to which they belong, the Intel® DLB atomic queue can schedule messages belonging to the same group to the same processor core for processing. In addition, the atomic queue dynamically selects processor cores for each stream. When there are multiple network data streams, traffic is evenly distributed to each processor core, ensuring load balancing among multiple cores in the processor.

In lockless rate limiting, the processor cores are divided into two groups—producers, and consumers—from the perspective of queue operations. The producer generates the flow ID required by the atomic queue for each message, and then forwards the message to the atomic queue in Intel® DLB, which distributes messages across the consumer threads while maintaining atomicity. After the consumer obtains the message from the atomic queue, it can safely access the global token bucket corresponding to the flow ID in lockless mode, to complete rate limiting operations.

In lockless rate limiting, because only global token buckets are used, there is no precision problem for low rate-limiting rate caused by local token bucket reservation and high rate-limiting rate caused by prefetch tokens. See Figure 4.

Table 3. Device Under Test (DUT) Configuration

Test Date	Test by Intel as of 11/4/2022
CPU	4th Gen Intel® Xeon® Scalable Processor 8458P
Memory	512 GB
Network Adapter	Intel® Ethernet Network Adapter E810-CQDA2
Network Adapter Firmware Version	2.30 0x80005d1b 0.0.0
Microcode	0x2b0000a1
Operating System	Ubuntu* 20.04.3 LTS
Linux* Kernel Version	5.4.0-131-generic
DPDK Version	20.11+DLB patch

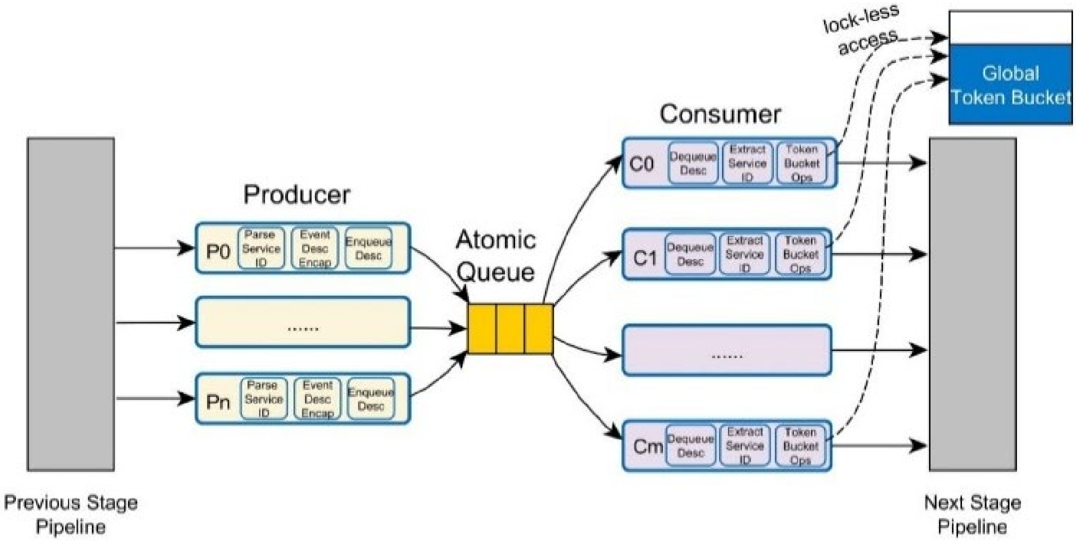


Figure 4. Intel® DLB Lockless Scheme for Rate Limiting

3 Comparison Testing

This section describes the comparison test principles, topology, and configuration.

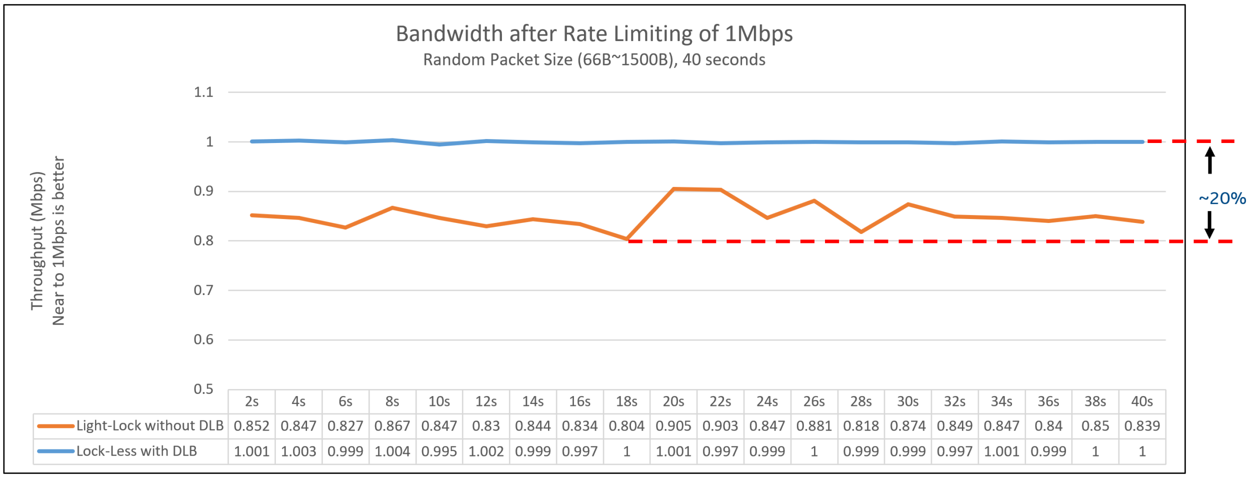
3.1 Test Principles and Topology

During the test, different network data streams requiring rate limiting are distinguished by destination IP addresses (DIP). Network data streams with different DIP addresses are sent to the device under test (DUT) through the network tester, and the packets are returned to the network tester after being processed by the DUT. The network tester calculates the receiving rate of data packets every 2 seconds, for 20 consecutive times (a total of 40 seconds), and then records the result.

The rate limiting software sets the rate of each DIP address to 1 Mbps. When the data flow rate sent by the network tester exceeds 1 Mbps, the rate limiting software discards some network messages to observe the rate limiting accuracy. Two tests were conducted. In the first test, the software using the lightweight lock for rate limiting is run on the DUT and the test results are recorded. In the second test, the software using lockless rate limiting is run on the DUT and the test results are recorded. The rate limiting accuracy of the two techniques is compared based on the results of the two tests.

3.2 Test Configuration

The DUT uses the pre-production 4th Gen Intel® Xeon® Scalable processors integrated with Intel® DLB and two Intel® Ethernet Network Adapters E810-CQDA2, each of which uses a 100 Gbps interface connected to two test ports of the network tester. See Figure 6. The configuration of the DUT is shown in Table 3.



	2s	4s	6s	8s	10s	12s	14s	16s	18s	20s	22s	24s	26s	28s	30s	32s	34s	36s	38s	40s
Light-Lock without DLB	0.852	0.847	0.827	0.867	0.847	0.83	0.844	0.834	0.804	0.905	0.903	0.847	0.881	0.818	0.874	0.849	0.847	0.84	0.85	0.839
Lock-Less with DLB	1.001	1.003	0.999	1.004	0.995	1.002	0.999	0.997	1	1.001	0.997	0.999	1	0.999	0.999	0.997	1.001	0.999	1	1

Figure 5. Comparison of Test Results

Table 4. Traffic Configuration in Network Tester

TRAFFIC GROUP	NUMBER OF IPS	RANDOM DIP SETTINGS	TOTAL BANDWIDTH OF THE GROUP (MBPS)	AVERAGE BANDWIDTH OF A SINGLE USER (MBPS)
Ordinary User Group	16384	192.168.0.0 / 0.0.63.255	8192	0.5
Malicious User Group	4096	192.168.64.0 / 0.0.15.255	6144	1.5
Observation Group	1	192.168.192.10 / 0.0.0.0	2	2

4 Network Tester Traffic Model

To closely recreate a real scenario, the test used random length messages from 66 bytes to 1,500 bytes. The traffic model included background traffic and observation traffic. The background traffic consisted of 80% ordinary users who do not exceed their rate and 20% malicious users who do try to exceed their rate. Due to the limitation of the network tester, traffic from all IP addresses was not collected. Therefore, we monitor the traffic of an observation group, which contained only a single DIP address. The network tester calculated the observation group received traffic rate to observe the rate limiting effect of a single DIP address.

4.1 Comparison and Analysis of Test Results

The test was conducted in November 2022. Figure 5 shows the test result chart, in which 20 data sampling points of lightweight lock rate limiting are connected by the orange line, and 20 data sampling points of lockless rate limiting are connected by the blue line.

As can be seen from the figure, the overall effect of lockless rate limiting is quite stable and accurate, and the overall error is less than 1%. However, lightweight lock rate limiting causes the flow rate to be small and to fluctuate greatly with an error of up to 20%.

Tests showed that Intel® DLB lockless rate limiting ensures higher rate limiting accuracy compared to lightweight lock rate limiting.

4.2 Flexibility and Universality of Lockless Rate Limiting

Existing applications developed in Linux kernel space, user space, or DPDK framework can be optimized with the Intel® DLB software development kit (SDK) and DPDK software library. Therefore, lockless rate limiting using Intel® DLB can also be applied to a variety of applications.

Intel® DLB also supports IO virtualization technologies such as Intel® Scalable I/O Virtualization (Intel® Scalable IOV), and Single Root I/O Virtualization (SR-IOV), enabling a single Intel® DLB device to be virtualized into multiple virtual devices and shared with multiple virtual machines or containers.

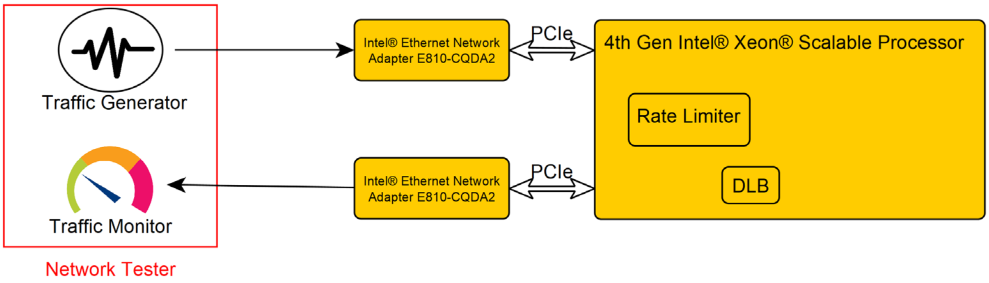

Traffic Generator
Traffic Monitor
Network Tester
Intel® Ethernet Network Adapter E810-CQDA2
Intel® Ethernet Network Adapter E810-CQDA2
PCIe
PCIe
4th Gen Intel® Xeon® Scalable Processor
Rate Limiter
DLB

Figure 6. Test Principles and Topology

5 Summary

Intel® DLB is a new acceleration technology introduced in the 4th Gen Intel® Xeon® Scalable processors with rich hardware queue management capabilities that bring new possibilities for software optimization.

Intel® DLB lockless rate limiting eliminates the need to use locks to protect the global token bucket. As a result, there is no need to consider lock optimization. Compared with the existing lightweight lock rate limiting, this paper demonstrates that the advantages of lockless rate limiting are clearly and intuitively demonstrated in terms of accuracy and performance.

Based on analysis and testing, it can be concluded that Intel® DLB lockless rate limiting, with the addition of a rich Intel® DLB SDK, achieves accurate rate limiting in the kernel mode and user mode of Linux, within the DPDK framework, and can be flexibly applied to different scenarios.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

0123/DN/WIT/PDF 759720-003US