An Industry Standard Benchmark Consortium

Networking Version 2.0 Benchmark Name: IP Reassembly

Highlights

- **Based on NetBSD kernel code**

Application

When an IP packet is too large to fit within the Maximum Transfer Unit (MTU) of the egress interface, it can no longer be transmitted as a single frame. Rather, the IP packet must be fragmented and transmitted in multiple frames. Dealing with reassembly, the process of reassembling the IP fragments to form the original packet, can place significant resource requirements on systems.

Additionally, with the increasingly heterogeneous networking environment of LANs and WANs, fragmentation becomes increasingly likely.

Benchmark Description

Based on the NetBSD kernel code, the IP Reassembly benchmark simulates the processing performed to handle reassembly. The benchmark simulates the arrival of a large number of IP fragments of varying lengths – ranging from those requiring just a single mbuf to those requiring a cluster. The degree of fragmentation, the number of fragments per IP packet, the arrival order of the fragments, and the number of packets being reassembled in parallel is configured to represent different networking environments.

When a packet "arrives" it is checked for basic correctness. Its packet ID, source, and destination parameters are compared with those of all the packets waiting for reassembly. If the fragment corresponds to a new packet, a new queue is started to hold the additional fragments that will be required to reassemble the packet. If the fragment belongs to a packet reassembly effort already in progress, then the doubly linked list which forms the reassembly queue is traversed to determine where this fragment belongs in the packet. Each fragment contains offset information indicating its relative position to the start of the packet. The new fragment is then inserted into the linked list at the appropriate position, and because fragment overlap is possible, it may be necessary to trim (or even dequeue) adjacent fragments. A check is subsequently made for complete reassembly. If, with the addition of the current fragment, reassembly is complete, fragment concatenation is undertaken and the reassembled packet is passed up the stack.

Analysis of Computing Resources

1. Tests data caches – large memory requirements test cache size and replacement algorithms, while aggressive pointer chasing tests latency.
2. Linked list traversal tests processors' ability to perform loads and compares and stresses processors' branch prediction logic and ability to recover gracefully from misprediction.

Special Notes NetBSD is a secure and highly portable UNIX-like operating system available for many platforms, from high-end servers to embedded and handheld devices. The core routines in this benchmark are based on the networking functionality in the NetBSD operating system.

Optimizations Allowed

Out of the Box / Standard C

Full Fury / Optimized

- For Out of the Box, you may not change the algorithm nor the C code except for the Test Harness, or to get the code to compile. All compiler-related changes must be documented and must not have a performance impact.
- For Out of the Box, if the C compiler can schedule for any additional hardware without code changes, these are allowed. ASM statements are not allowed. All optimized libraries must be part of the standard compiler package, and/or available to all customers
- You may use Test Harness Regular or Test Harness Lite. You may not create your own Test Harness.
- For Optimized, you may re-write the basic algorithm as long as the output is unchanged.
- For Optimized, you may re-write the code in assembler.
- For Optimized, you may use publicly available optimized libraries, use hardware-assist if it is on the same processor as that being benchmarked, and/or inline the code.
- You may not assume that the data files shipped with the benchmark are the only data files that will be run through the certification run by EEMBC. However, the distributions will be as specified above.