Stress Testing

It is important to keep the overhead of performance monitoring tools very low to minimize the interference they can impose on user workloads. In this document, we show the overhead of running Intel® Workload Interference Detector (procmon) on running workloads. We first use [stress-NG](https://wiki.ubuntu.com/Kernel/Reference/stress-ng) stress testing tool, with varying CPU load%. Next we use [Intel® Memory Latency Checker (MLC)](https://www.intel.com/content/www/us/en/developer/articles/tool/intelr-memory-latency-checker.html) with various numbers of concurrently running containers, from 1 to 128 (=total number of CPUs).

Host system specs:

1. **System: AWS EC2 m6i.metal system (vCPU = 128, Memory (GiB) = 512)**
2. **OS: Ubuntu 22.04.1 LTS**
3. **Linux Kernel: 5.19.0-1026-aws**

# Stress-NG:

We run stress-NG with varying CPU loads: 30%, 50%, 90%, and 99%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Command:** | **docker run --rm colinianking/stress-ng --cpu 0 --cpu-load <<load>> --timeout 20s --metrics-brief** | | | |  |
|  |  |
|  | **bogo ops** | | | | **Max  Overhead %** |
| **CPU load %** | **30%** | **50%** | **90%** | **99%** |  |
| **No procmon** | **1,210,898** | **1,910,878** | **2,965,798** | **3,164,739** | **-** |
| **Procmon** | **1,217,496** | **1,909,337** | **2,955,713** | **3,148,672** | **0.51%** |
| **Dockermon** | **1,212,594** | **1,903,808** | **2,957,927** | **3,150,668** | **0.44%** |
| **NN\_detect** | **1,219,411** | **1,911,108** | **2,948,380** | **3,142,280** | **0.71%** |

As observed from the table, procmon’s overhead does not exceed 1% when system is running under 99% utilization.

# MLC:

We measure the overhead of procmon.py, dockermon.py, and NN\_detect.py in two cases:

1- One MLC container running on all CPUs

2- 128 MLC containers, 1 CPU per container

The overhead is measured as the performance (Bandwidth in MB/sec) reported by MLC with and without running procmon on the same host. We run the following commands in EC2 m6i.metal system.

# First, we start by running one MLC container that stresses L1 cache on all CPUs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MLC-l1 - 1-container - all CPUs | | | |
|  | No procmon | With procmon | With dockermon | With NN\_detect |
| Bandwidth (MB/Sec) | 6820035.7 | 6785762.8 | 6783973.4 | 6788992.1 |
| overhead % |  | 0.50% | 0.53% | 0.46% |

# Second, we run 128 MLC containers (1CPU/container) and aggregate the bandwidths across the 128 containers

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Without procmon | With procmon | With dockermon | With NN\_detect |
| Bandwidth (MB/Sec) | 7311277.9 | 7250520.9 | 7320665.6 | 7320656.4 |
| overhead % |  | 0.83% | -0.13% | -0.13% |

As we can see, the performance overhead is negligible in both cases. Specifically, even when CPU utilization ~ 100%, the maximum overhead measured if 0.83%. Which shows that procmon is lightweight and suitable in monitoring performance of running workloads without causing any significant noise.