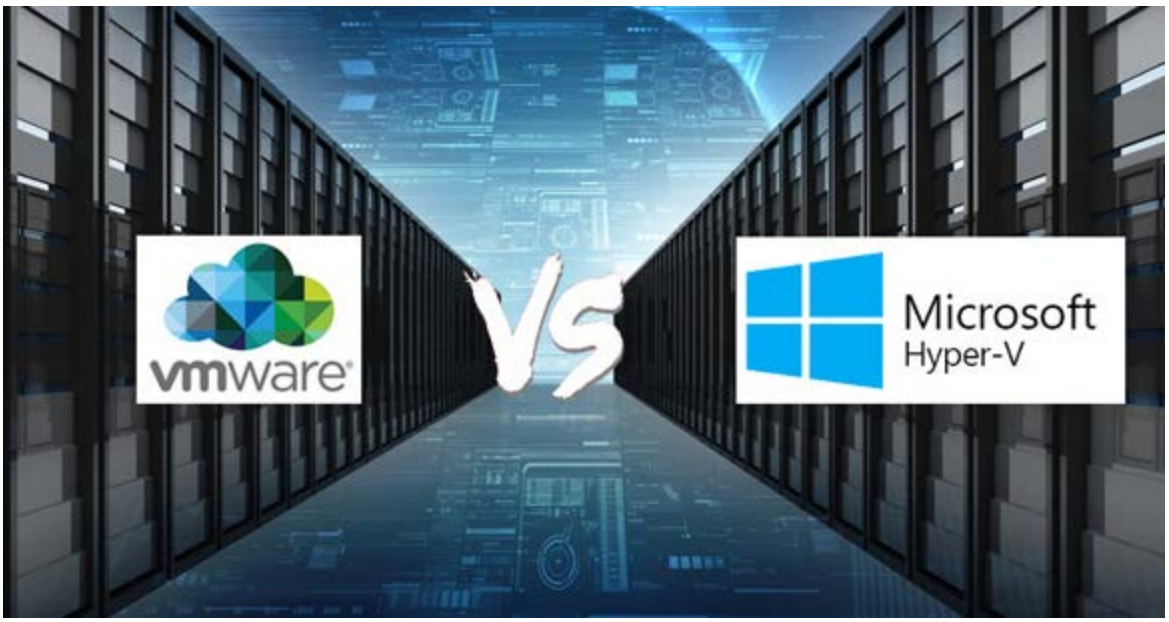
Virtual Systems



Performance Comparison

Project Report

Authors: Daniel Battagani and Varun Subramanya
Graduate Program: MSIT
UMASS Lowell, Massachusetts
July 10th – 29th, 2018

UMASS LOWELL

Table of Contents

Abstract **4**
Introduction **5**
Virtualization 5
Server Virtualization 5
Hypervisor 5
Type 1 Hypervisor 5
Type 2 Hypervisor 6
Desktop Virtualization 6
Client hosted desktops 6
Server hosted desktops 6
VMware Overview 6
VMware ESX Server 7
VMware ESXi Server: 7
Microsoft Hyper-V Overview 7
Feature Comparison **9**
Design Comparison 9
Monolithic 9
Microkernelized 10
CPU Comparison 10
VMware CPU 10
Hyper-V CPU 10
Memory Comparison 11
VMware ESX/ESXi Memory Overcommit and Memory Ballooning 11
Microsoft Hyper-V Dynamic Memory: 11
Background **12**
Testing Environments 12
Environment 1 Design 12
Environment 1 Physical Specifications 12
Environment 2 Design 15
Environment 2 Physical Specifications 15
Methodology **17**
Benchmarking 17
SysBench 17
Measurements **18**
Sysbench Tests 18
CPU TEST 18
MEMORY TEST 19
I/O FILE TEST 22
Conclusion **25**
Appendix **26**
Environment 1 Scripts 26
CPU Test Script 26
Memory Test Script 26
I/O Test Script 26
Environment 2 Scripts 27

CPU Test Script 27
Memory Test Script 27
I/O Test Script 27
Environment 1 Detailed Test Results 28
VMware ESXi – Ubuntu Testing 28
Microsoft Hyper-V – Ubuntu Testing 37
Environment 2 Detailed Test Results 46
VMware ESXi – Ubuntu Testing 46
Microsoft Hyper-V – Ubuntu Testing 56
References 66

Abstract

Virtualization is a technology that runs multiple independent virtual operating systems on a single physical device. There are several virtualization vendors available in the market today with their proprietary products like VMware ESXi, Microsoft Hyper-V, Citrix, Red Hat and Amazon. In this paper, we analyzed VMware (ESXi) and Microsoft (Hyper-V) products with regards to the CPU Performance, Memory Performance and File I/O Performance.

In our analysis for performance comparison, we had used two environments as explained in the Background section to find out the overheads incurred by each hypervisor and Sysbench, as our benchmarking to understand the cross-platform and multi-threaded performance that are important for a system running under intensive load.

We didn't find any hypervisor (ESXi or Hyper-V) outperforming each other, however, we did noticed some differences in the CPU and Memory performance that is slightly better in ESXi at higher CPU values than Hyper-V. I/O tests performances varied differently between the two environments, but this is largely due to the design of the environments we had used for this analysis indicating that it's important to carefully plan and design the virtual environments in terms of load, capacity and speed with respective to the applications hosted in the virtual environments.

Introduction

The virtual Machine Monitor (VMM) is the control system at the core of virtualization. It acts as the control and translation system between the virtual machines (VMs) and the hardware in efficiently controlling the physical platform resources that includes memory translation and I/O mapping. It is the primary software behind the virtualization technology.

This section provides a high level information about the Virtualization technology and hypervisor types we used in this analysis to compare the performance of VMware and Hyper-V on a Dell PowerEdge Server and a Dell laptop.

Virtualization

Virtualization is a broad computing term used for the creation of many virtual resources from one physical resource. As per Wikipedia, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.

Virtualization can take many forms depending on the type of application used, hypervisor used and hardware platforms that the virtualization is built.

Server Virtualization

Server virtualization also knows as Hardware-assisted virtualization or Hardware virtualization is a concept that partitions a physical server into a number of small, virtual servers with the help of virtualization software called “Hypervisors”. The hypervisor interacts directly with the physical server's CPU and disk space. It serves as a platform for the virtual server’s operating systems. The hypervisor keeps each virtual server completely independent and unaware of the other virtual servers running on the physical machine, that is, every virtual machine runs in isolation and has its own OS and applications. VMware ESX/ESXi and Microsoft Hyper-V virtualization comes under server Virtualization.

Hypervisor

A hypervisor or virtual machine monitor (VMM) is computer software, firmware or hardware that creates and runs virtual machines. It is a process that separates a computer’s operating system and applications from the underlying physical hardware.

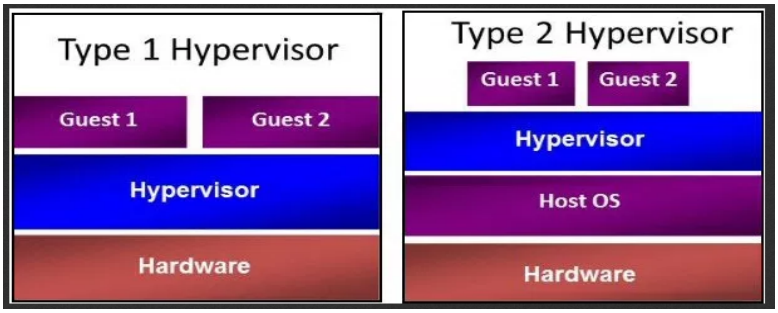
There are two (2) types of hypervisors – Type 1 and Type 2.

Type 1 Hypervisor

Type 1 hypervisors run directly on the system hardware. They are often referred to as a "native" or "bare metal" hypervisors also. It has direct access to the physical hardware, with no operating system getting in the way of the virtual machines using that hardware.

Type 2 Hypervisor

Type 2 hypervisors run on a host operating system just like another program or application in the system. These are designed to install onto an existing operating system.



Desktop Virtualization

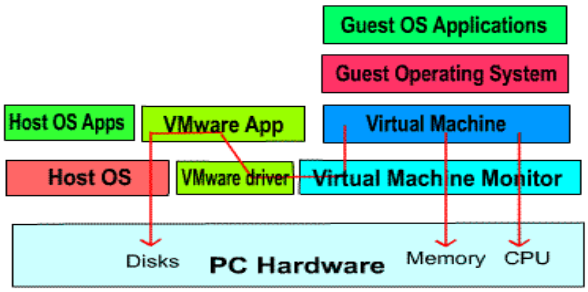
This type of virtualization consists of two (2) types – Client Hosted and Server Hosted.

Client hosted desktops

In this type of Desktop virtualization, a virtual machine runs on the desktop computer. An operating system runs within an application (virtualization software) that emulates or abstracts actual hardware into a standard set of virtual hardware. VMware workstation comes under this desktop virtualization.

VMware workstation

VMware Workstation is a hosted hypervisor that runs on Windows and Linux operating systems.
It is used to run multiple operating systems as virtual machines (VMs) on a single Linux or Windows PC. It works as a bridge between the host and virtual machine for all kinds of hardware resources including hard disks, USB devices and CD-ROMs. All device drivers are installed via the host device.



VMware Workstation Architecture

Server hosted desktops

In this type of Desktop virtualization, the client OS, Applications & Data are kept on Severs in data centers for better utilization of hardware, easy maintenance and accessibility. The technology which enables this is called “Virtual Desktop Infrastructure (VDI)” that leverages protocols like Remote Desktop or a Virtual Desktop Client (VMware View) for access.

VMware Overview

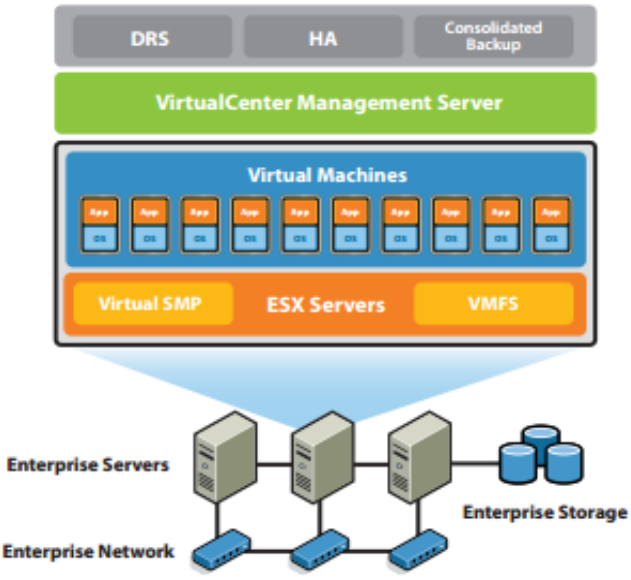
VMware, a subsidiary of Dell Technologies is a virtualization and cloud computing software provider. Its software allows users to create multiple virtual environments, or virtual computer systems, on a single computer or server. VMware product line mainly consists of two types of virtualization – Server and Desktop Virtualization.

VMware's software, hypervisors, intended for servers are bare-metal embedded hypervisors that can run directly on the server hardware without the need of an extra primary OS.

VMware's line of server software includes:

VMware ESX Server

ESX (Elastic Sky X) is an enterprise-level solution, which is built to provide better functionality in comparison to the freeware VMware Server resulting from a lesser system overhead. It is the VMware's enterprise server virtualization platform. In ESX, VMkernel is the virtualization kernel which is managed by a console operating system also called as Server console.



VMware ESXi Server:

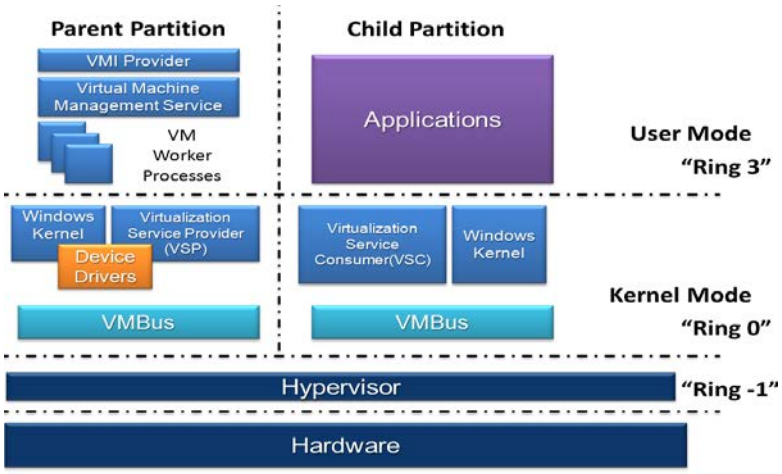
ESXi (Elastic sky X Integrated) is a purpose-built bare-metal hypervisor that installs directly onto a physical server. With direct access to and control of underlying resources, ESXi is more efficient than hosted architectures.

It is similar to the ESX, except that the service console is replaced with BusyBox installation and it requires very low disk space to operate. All the VMware related agents run directly on the VMkernel.

VMware ESX is integrated with VMware vCenter that provides additional solutions to improve the manageability and consistency of the server implementation.

Microsoft Hyper-V Overview

Microsoft Hyper-V, codenamed Viridian and formerly known as Windows Server Virtualization, is a native hypervisor that can create virtual machines on x86-64 systems running on Windows. It is Microsoft's hardware virtualization product and a hypervisor-based virtualization technology.



Microsoft Hyper-V is a server virtualization hypervisor that enables consolidation of a single physical server into many virtual servers, all sharing the hardware resources of the host server and powered by Hyper-V. Hypervisor manages the interactions between the hardware and the virtual machines. This hypervisor controlled access to the hardware gives virtual machines the isolated environment in which they run. Hyper-V operates both as a standalone

solution and as an addition to Windows Servers 2008/2012/2016 R2, however, during installation it redesigns the OS architecture and becomes just like a next layer on the physical hardware.

Hyper-V, isolates part of a physical machine into child partitions and allocates them to different guest operating systems, with Windows Server OS acting as the primary host/parent. Hyper-V also assigns appropriate hardware and software resources for each of the guest operating system it's hosting because they don't have direct access to the hardware resources and rely on Hyper-V.

Feature Comparison

This sections provides a high-level theoretical feature comparison between VMware ESX/ESXi and Microsoft Hyper-V hypervisors with respective to their designs, CPU, and Memory performances.

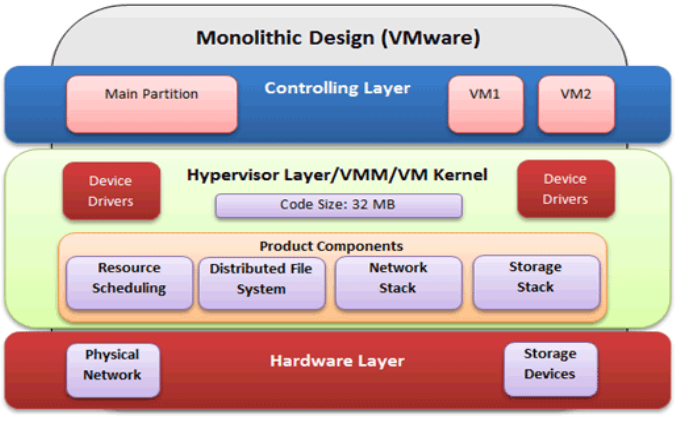
Comparison Table:

Features	Hyper-V	VMware ESXi
Base OS	Windows Server	vmkernel (linux based)
Architecture	Bare-Metal	Bare-Metal
Virtualization Technology	Para Virtualization, Server Virtualization	Full Virtualization, Server Virtualization
CPU Scheduling	VM Reserve, VM limit, relative weight	share-Based, Relaxed Co-Scheduling
SMP Scheduling	CPU base scheduling	CPU aware load balancing
Memory Address Translation	Shadow pagetable, Hardware assisted pagetable	Emulated TLB, Shadow pagetable, Hardware assisted pagetable
Disk Management	Fixed Disks, Pass-Through Disks, Dynamic Disks	Latency-aware, Priority based Scheduler, Storage DRS
Network Management	TCP offload, VM queue, large send offload	Priority based Network I/O, TCP segmentation offload, netqueue, distributed virtual switch

Design Comparison

Monolithic

VMware ESX/ESXi uses “Monolithic” Hypervisor Design. In this design the hypervisor is aware of the device drivers that are hosted in and managed by the "Hypervisor Layer."

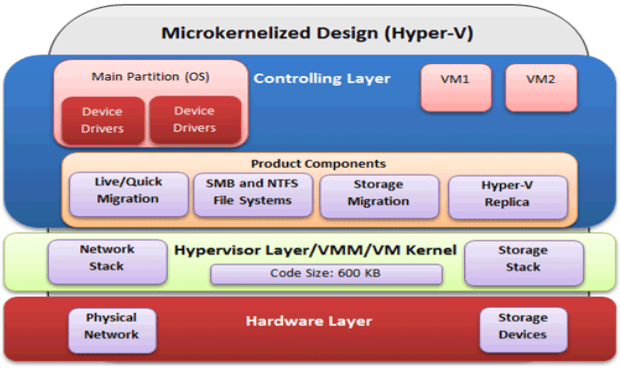


Monolithic design implements a shared driver model from the hypervisor itself. VMware operates its components in the "Hypervisor Layer.

No operating system is required for controlling the components of the virtualization product. This is the biggest advantage over the Microkernelized Hypervisor Design used by the Microsoft Hyper-V, but requires more initialization time than Hyper-V.

Microkernelized

Hyper-V uses “Microkernelized” Hypervisor Design. This design implements a shared Virtualization stack and a distributed driver model that is flexible and secure. Also it does not require the device drivers to be part of the Hypervisor layer. The device drivers operate independently and run in the "Controlling Layer". This approach has less overhead for maintaining and managing the Device Drivers.



The Microkernelized Hypervisor Design allows you to install any other server roles in the "Controlling Layer" apart from Server Virtualization role (Hyper-V). There is less initialization time required as hypervisor code is only about 600 KB in size and as a result, the "Hypervisor Layer" initialize its components fast.

CPU Comparison

As the systems are pushed to their limits, Operating System schedules CPU resources appropriate to the workloads being added resulting in CPU contention issues. A CPU contention is the ability for the system to respond to processing requests. It forces the operating system to more carefully dole out the resources. VMware ESXi and Hyper-V provides manual methods for CPU scheduling to the VMs.

VMware CPU

VMware has CPU scheduling controls for shares, reservation and limit.

- Shares are used to allocate the relative performance of a VM. If a virtual machine has a share value that is half of another, it’s entitled to only half the CPU resources.
- Reservation makes sure that the VM will get the resources specified in MHz.
- Limit is the ability to stop an individual or single VM to consuming unlimited resources.

Hyper-V CPU

Hyper-V has VM reserve, VM limit and Relative weight for CPU scheduling controls.

- VM reserve allows the reservation of a portion of the server’s total processing resources for this virtual machine.
- VM limit make sure that the host processing resources are not consumed by an individual or single VM.
- Relative Weight is a method to determine how much processing power should be consumed by the virtual machine. The relative weight option allows the weighting of this virtual machine against others

Memory Comparison

VMware ESX/ESXi Memory Overcommit and Memory Ballooning

VMware can support more virtual machines on the same hardware than any other x86 hypervisor. Of all x86 bare-metal hypervisors, only VMware ESX supports memory overcommit, which allows the memory allocated to the virtual machines to exceed the physical memory installed on the host. VMware ESX supports memory overcommit with minimal performance impact.

Content-based transparent memory page sharing conserves memory across virtual machines with similar guest OSs by seeking out memory pages that are identical across the multiple virtual machines and consolidating them so they are stored only once, and shared.

VMware ESX enables virtual machines to manage their own memory swap prioritization by using memory ballooning to dynamically shift memory from idle virtual machines to active virtual machines. Memory ballooning artificially induces memory pressure within idle virtual machines as needed, forcing them to use their own paging areas and release memory for more active or higher-priority virtual machines

Microsoft Hyper-V Dynamic Memory:

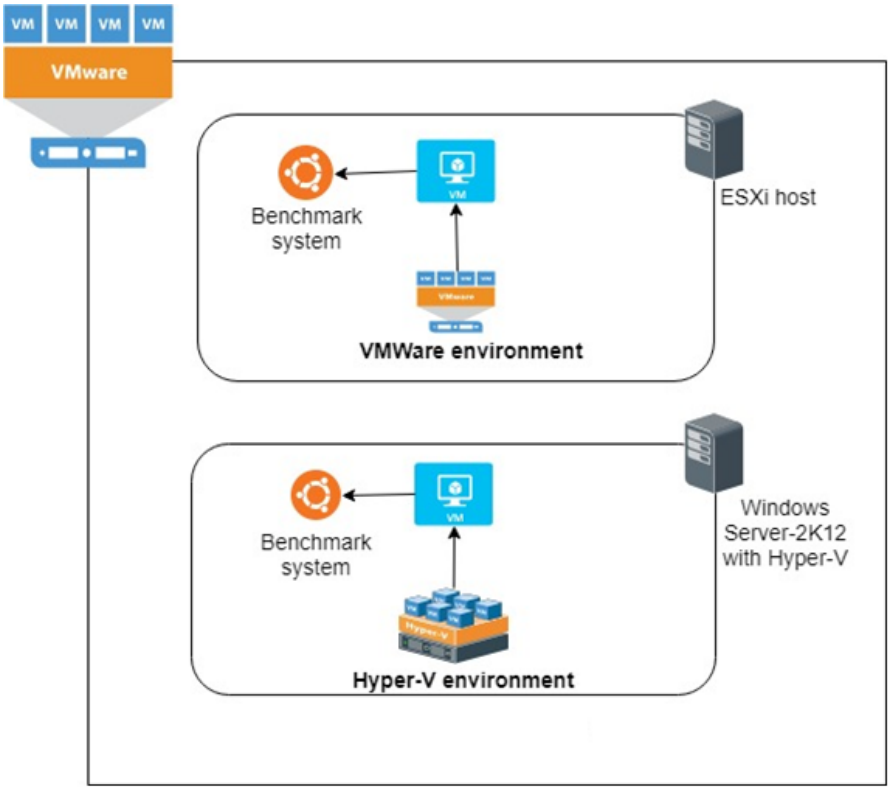
Hyper-V Dynamic Memory works different when compared with VMware memory techniques. Instead of compressing the host RAM, it allows virtual machines to demand more RAM if required. This is sometimes referred as Memory Oversubscription in Microsoft. This is managed by a device driver called “Dynamic Memory Virtual Service Consumer (DM VSC)” that monitor the memory usage in the guest OS. This driver collects information about current RAM requirements and reports it to the host which decides to give or take RAM from the virtual machine.

Background

This section provides information about the different infrastructures environments used in this analysis for the performance comparison.

Testing Environments

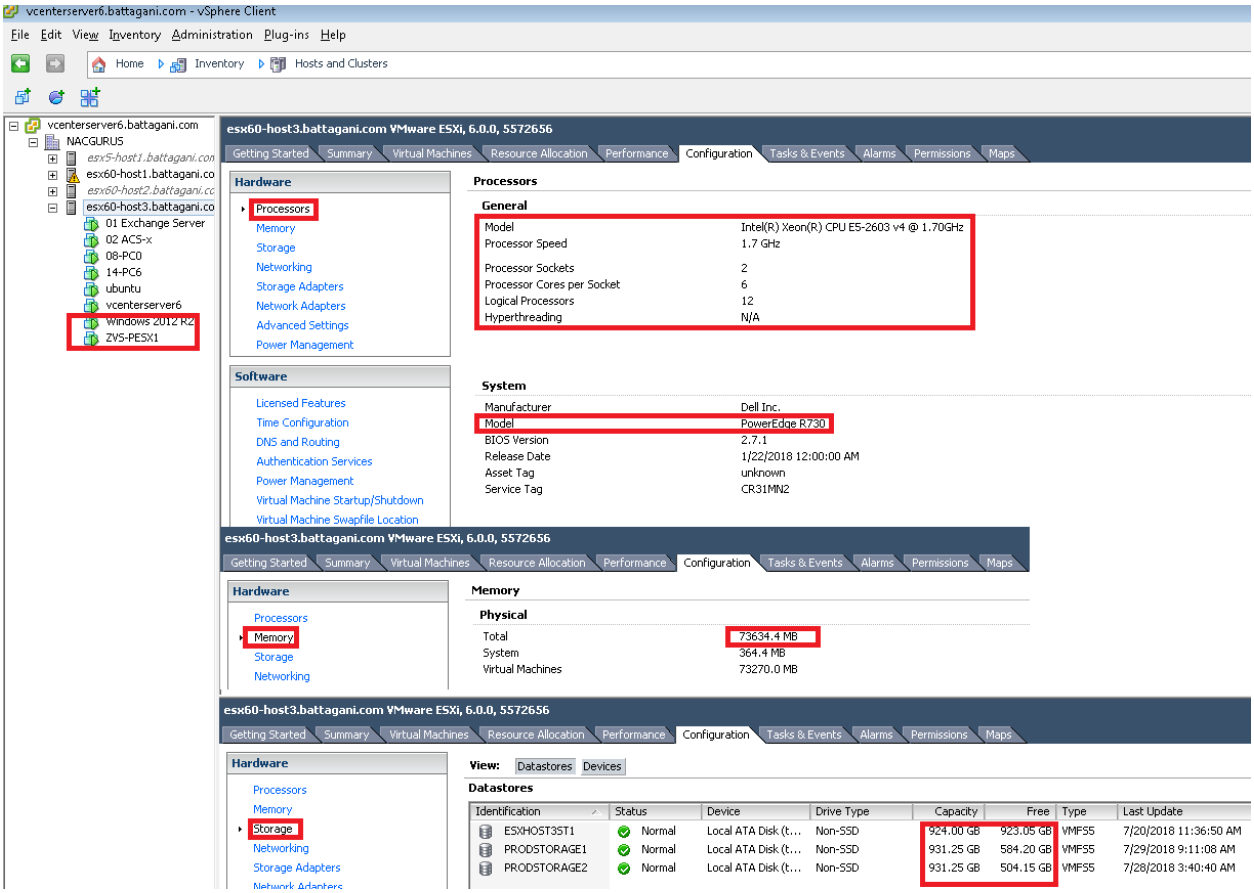
Environment 1 Design



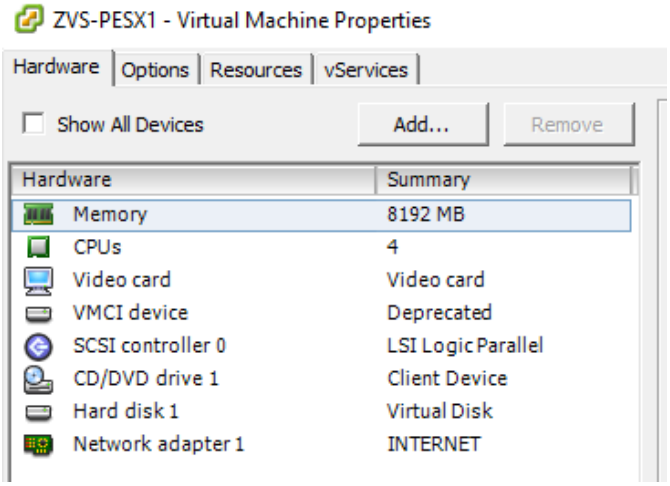
Environment 1 Physical Specifications

This environment uses a Type-1 hypervisor VMware ESXi 6.0 installed on a Dell PowerEdge R730 server. The physical server consists of two processor sockets with six cores per socket, a total of 12 CPUs, 72 GB Memory and three (3) TB disk space.

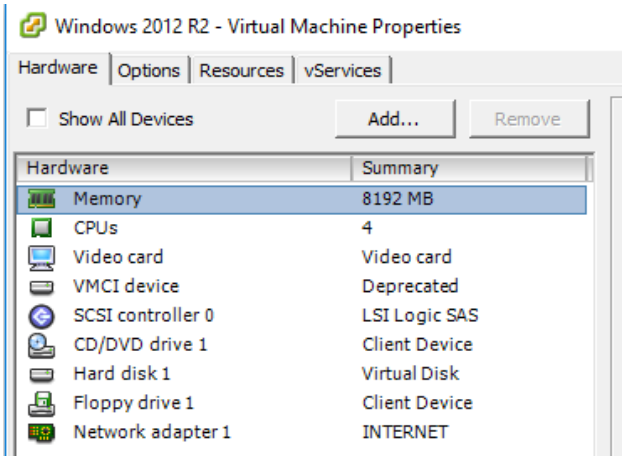
We did a “Nested Virtualization” for fair assessment between VMware and Hyper-V performance comparison using the same hardware. As part of the nested virtualization, we deployed a VMware ESXi host as a VM and Windows 2012 Server with Hyper-V role.



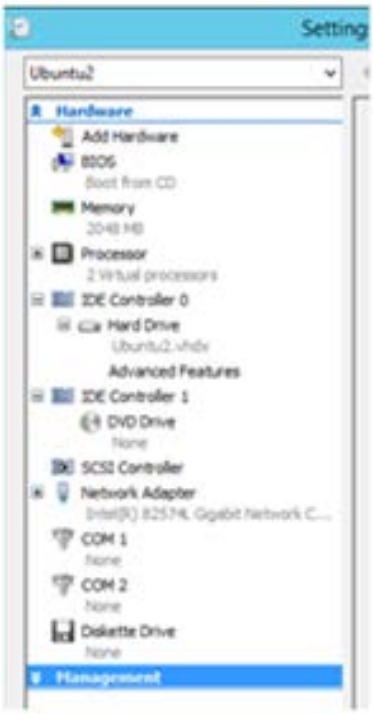
VMware ESXi version 6 host as a virtual machine guest consists of 4 CPUs, 8GB Memory with 200GB Hard Disk Space allocated.

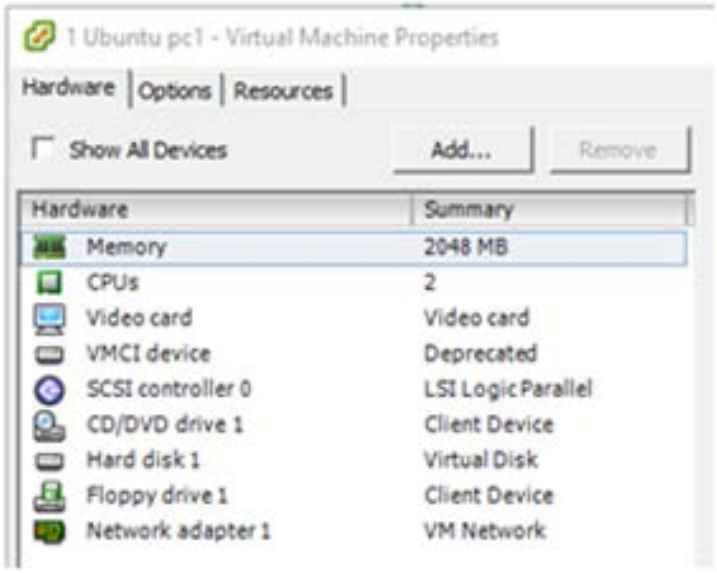


A Windows 2012 Server virtual machine guest with Hyper-V role consists of 4 CPUs, 8GB Memory with 200GB Hard Disk Space allocated was deployed

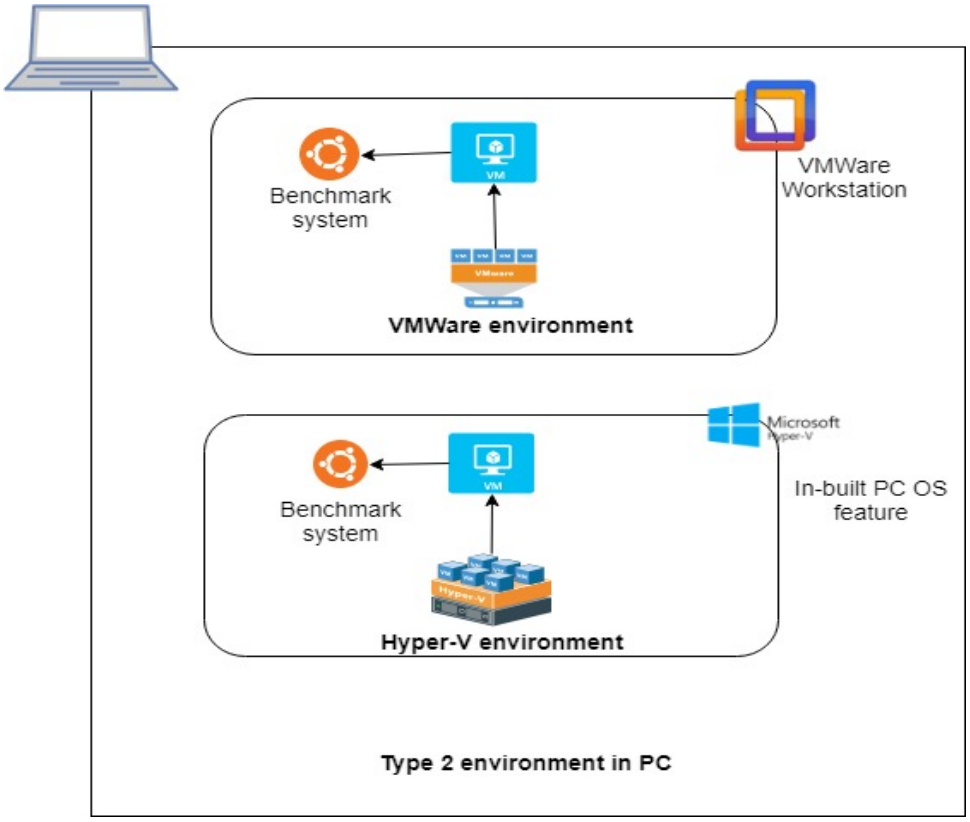


An Ubuntu desktop 18.04 version was deployed on both the VMware ESXi host and Hyper-V for carrying out the benchmark tests. This consists of 2 CPUs (2 CPUs with one core per socket), 2 GB Memory and 25GB Hard Disk space allocated.





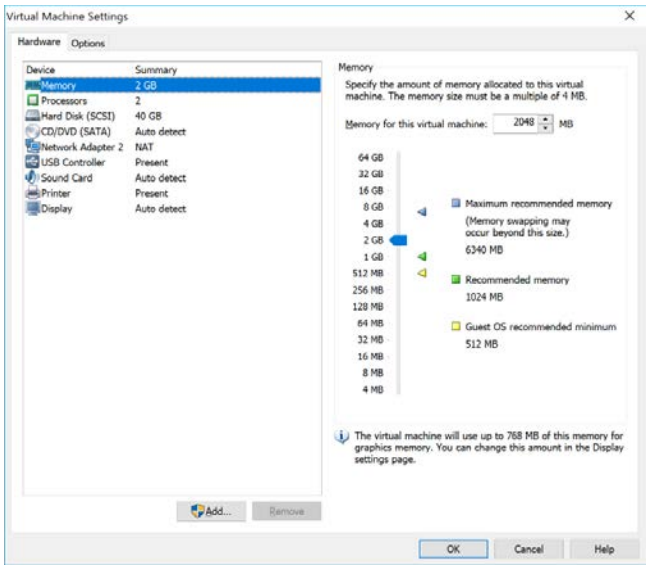
Environment 2 Design

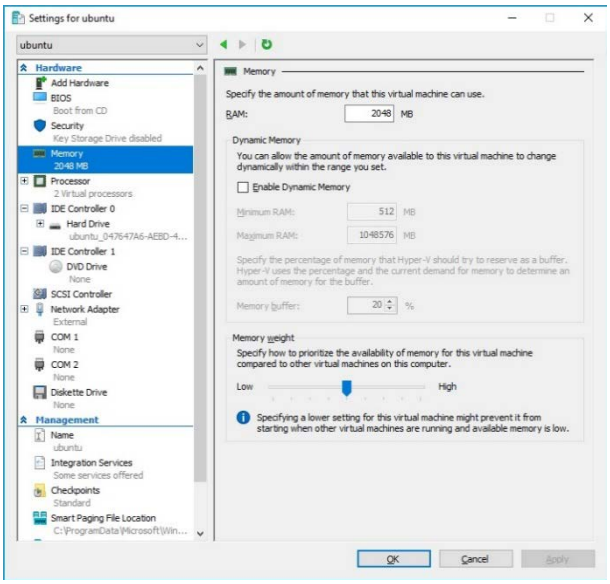


Environment 2 Physical Specifications

On contrast to environment 1, the underlying system in environment 2 is a personal PC -Microsoft Surface Pro 4 with Intel(R) Core (TM) i5-6300U CPU @ 2.40GHz. The system runs Windows 10 64 Bit version on which VMware Workstation and Hyper-V manager (in-built to OS) installed to evaluate the performance.

An Ubuntu desktop 18.04 version was deployed on both the VMware ESXi host and Hyper-V for carrying out the benchmark tests. This consists of 2 CPUs (2 CPUs with one core per socket), 2 GB Memory and 25GB Hard Disk space allocated as shown in the screenshots below.


Virtual Machine Settings
Hardware
Options
Device
Summary
Memory
2 GB
Processors
2
Hard Disk (SCSI)
40 GB
CD/DVD (SATA)
Auto detect
Network Adapter 2
NAT
USB Controller
Present
Sound Card
Auto detect
Printer
Present
Display
Auto detect
Add...
Remove
Memory
Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.
Memory for this virtual machine:
2048
MB
64 GB
32 GB
16 GB
8 GB
4 GB
2 GB
1 GB
512 MB
256 MB
128 MB
64 MB
32 MB
16 MB
8 MB
4 MB
Maximum recommended memory
(Memory swapping may occur beyond this size.)
6340 MB
Recommended memory
1024 MB
Guest OS recommended minimum
512 MB
The virtual machine will use up to 768 MB of this memory for graphics memory. You can change this amount in the Display settings page.
OK
Cancel
Help


Settings for ubuntu
ubuntu
Hardware
Add Hardware
BIOS
Boot from CD
Security
Key Storage Drive disabled
Memory
2048 MB
Processor
2 Virtual processors
IDE Controller 0
Hard Drive
ubuntu_047647A6-AEBD-4...
IDE Controller 1
DVD Drive
None
SCSI Controller
Network Adapter
External
COM 1
None
COM 2
None
Diskette Drive
None
Management
Name
ubuntu
Integration Services
Some services offered
Checkpoints
Standard
Smart Paging File Location
C:\ProgramData\Microsoft\Win...
Memory
Specify the amount of memory that this virtual machine can use.
RAM:
2048
MB
Dynamic Memory
You can allow the amount of memory available to this virtual machine to change dynamically within the range you set.
Enable Dynamic Memory
Minimum RAM:
512
MB
Maximum RAM:
1048576
MB
Specify the percentage of memory that Hyper-V should try to reserve as a buffer. Hyper-V uses the percentage and the current demand for memory to determine an amount of memory for the buffer.
Memory buffer:
20
%
Memory weight
Specify how to prioritize the availability of memory for this virtual machine compared to other virtual machines on this computer.
Low
High
Specifying a lower setting for this virtual machine might prevent it from starting when other virtual machines are running and available memory is low.
OK
Cancel
Apply

Methodology

This section provides information about the measurement methods used for performance analysis.

Benchmarking

A computerized test for measuring the properties of the particular technology is called benchmarking. The properties might include speed, performance, transfer rate, etc. Since there are so many benchmarking tools available for the analysis of a complex entity such as virtualization, we decided to use the “Sysbench” benchmarking to measure (using generic workloads) CPU, Memory, and Disk I/O.

The reason we choose SysBench benchmarking as it is a cross-platform and multi-threaded benchmark tool for evaluating OS parameters that are important for a system running under intensive load. The idea of this benchmark suite is to quickly get an impression about system performance without setting up complex database benchmarks or even without installing a database at all, making it easy for our testing.

SysBench

SysBench is an open source tool that can be used for almost all kind of operating system and multi-threaded benchmarking. The benchmark runs directly on the system and does not need any load generators. It is a simple scriptable tool to use and it runs specified number of threads by default that is one and can be changed as needed for the specific tests being performed. System performance is generally measured based on the CPU memory and Input and output (I/O) of data of given system.

SysBench runs a specified number of threads and they all execute requests in parallel. The actual workload produced by requests depends on the specified test mode. You can limit either the total number of requests or the total time for the benchmark, or both. Available test modes are implemented by compiled-in modules, and SysBench was designed to make adding new test modes an easy task.

Measurements

This section provides information about the measurements performed for performance comparison. For full test results, please refer to the sections Environment 1 Detailed Test Results and Environment 2 Detailed Test Results.

Sysbench Tests

CPU TEST

The CPU test is one of the benchmarks test in SysBench. The test tries to find if number is prime or not and uses a maximum prime number value that's defined when the test is run to determine how long it takes the system to calculate the numbers. When running with the CPU test, sysbench will verify prime numbers by doing standard division of the number by all numbers between 2 and the square root of the number.

We decided to use this test as the CPU test is very processor intensive and will generally run the CPU at 100% during the test making it a good test case for the CPU performance measurement.

In this test each request consists in calculation of prime numbers up to a value specified by the --cpu-max-primes option. All calculations are performed using 64-bit integers. Each thread executes the requests concurrently until either the total number of requests or the total execution time exceed the limits specified with the common command line options.

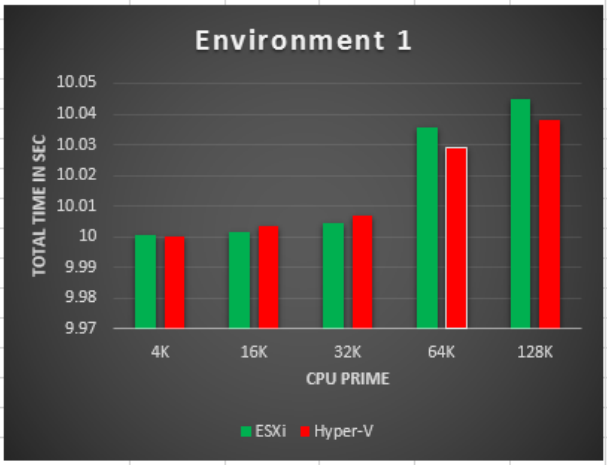
Command Executed: sysbench --test=cpu --cpu-max-prime=4000 run
Result:

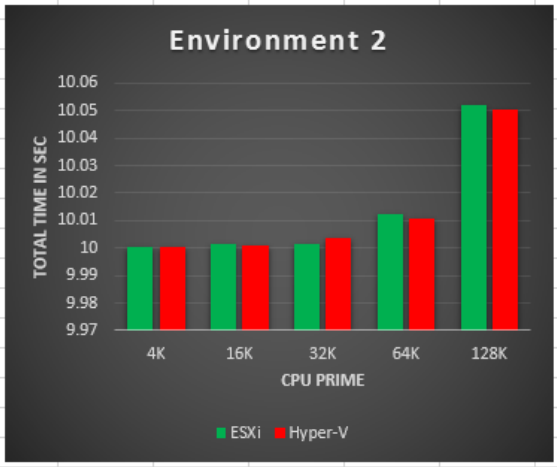
```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 4000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:  1817.00
General statistics:
    total time:                          10.0005s
    total number of events:              18182
Latency (ms):
         min:                                    0.49
         avg:                                    0.55
         max:                                    9.28
         95th percentile:                        0.73
         sum:                                 9985.84
Threads fairness:
    events (avg/stddev):           18182.0000/0.00
    execution time (avg/stddev):   9.9858/0.00
```

Total time is the end-to-end time, and as such includes the overhead of shared memory access for the threads. So we took the total time vs CPU Prime value for our comparison. Tests were repeated several times with different CPU Prime values (4000, 16000, 32000, 64000, 128000) on two Ubuntu desktops hosted in ESXi and Hyper-V environments 1 and 2 to do the fair assessment. Following is the graphical representation of the CPU test results in both environments.

CPU Test Results Graph:

	Environment 1		Environment 2	
CPU PRIME	ESXi	Hyper-V	ESXi	Hyper-V
4K	10.0005	10.0001	10.0003	10.0004
16K	10.0015	10.0033	10.0014	10.0012
32K	10.0043	10.007	10.0017	10.0035
64K	10.0356	10.0289	10.0122	10.011
128K	10.045	10.0379	10.052	10.0502





From the graph, it is quite evident that CPU intensive test is faster on Hyper-V machine where system calls are directly being executed without VMM Intervention. However, there is slight difference in tests on higher CPU max-prime number where too many systems calls to be made where Hyper-V lags the performance whereas VMware VM provided consistence performance. To prove that, we did run the test multiple times and found when Max Prime value is higher that 32K, VMware provided significant improvement performance. It proves that the VMware provides better performance when compared Hyper-V even though it has kernel architectural advantage.

MEMORY TEST

This Memory test is be used to benchmark sequential memory reads or writes. Depending on command line options each thread can access either a global or a local block for all memory operations. The memory test in sysbench, allocates a memory buffer and then read or write from it, each time for the size of a pointer (so 32bit or 64bit), and each execution until the total buffer size has been read from or written to. This is then repeated until the provided volume (--memory-total-size) is reached.

Multiple threads (--num-threads) can be provided with different sizes in buffer (--memory-block-size) and the type of requests (read or write, sequential or random).

Command Execution: sysbench --test=memory --memory-block-size=1K --memory-scope=global --memory-total-size=100G --memory-oper=write --num-threads=<value of threads"

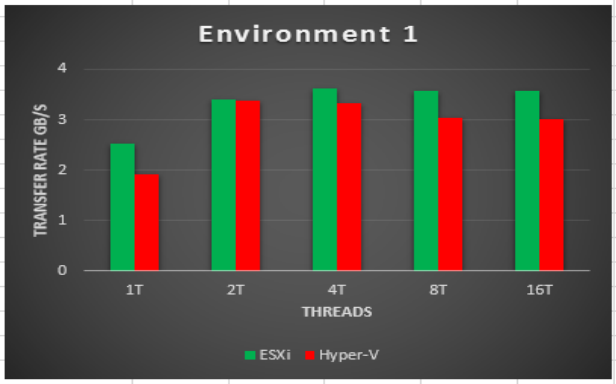
Result:

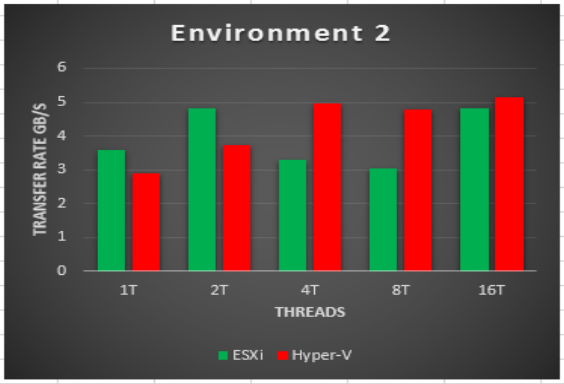
```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 24744251 (2472977.91 per second)
24164.31 MiB transferred (2415.02 MiB/sec)
General statistics:
    total time:                          10.0004s
    total number of events:              24744251
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    0.60
         95th percentile:                        0.00
         sum:                                 4558.86
Threads fairness:
    events (avg/stddev):           24744251.0000/0.00
    execution time (avg/stddev):   4.5589/0.00
```

The important number to compare is the amount of the transfer rate (transferred) per specified threads and the total time taken to complete the test. Tests were repeated with different Thread values (1, 2, 4, 8 and 16) and were performed for Memory Write and Memory Read on two Ubuntu desktops hosted in ESXi and Hyper-V environments 1 and 2 to do the fair assessment. Following is the graphical representation of the test results in both environments.

Memory Write Test Results Graph:

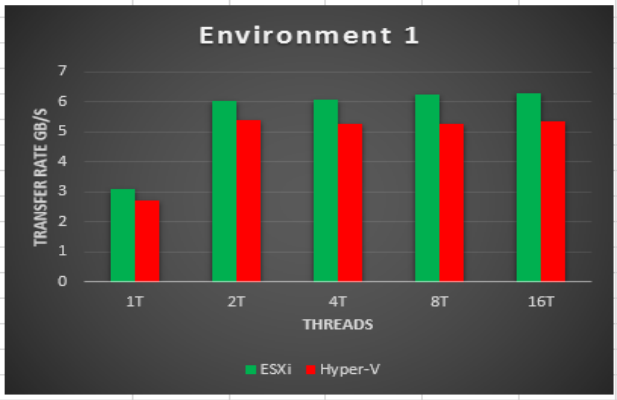
	Environment 1		Environment 2	
Memory Write	ESXi	Hyper-V	ESXi	Hyper-V
1T	2.53	1.92	3.57	2.9
2T	3.41	3.37	4.82	3.72
4T	3.61	3.32	3.31	4.98
8T	3.57	3.03	3.03	4.77
16T	3.57	3.01	4.81	5.14

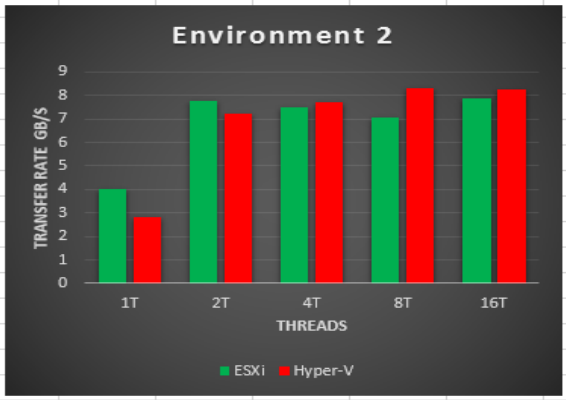




Memory Read Test Results Graph:

	Environment 1		Environment 2	
Memory Read	ESXi	Hyper-V	ESXi	Hyper-V
1T	3.12	2.71	4.01	2.85
2T	6.02	5.39	7.75	7.24
4T	6.09	5.27	7.5	7.73
8T	6.25	5.27	7.05	8.32
16T	6.29	5.37	7.87	8.29





From the Memory Read and Memory Write test result graphs, it proves that performance is better on ESXi where it provides more throughput in transferring the data from and to memory/RAM. Higher throughput helps CPU can execute the instructions faster there by providing a significant increase in system performance.

I/O FILE TEST

File I/O test is used to perform various kinds of file I/O workloads. This is similar to Disk I/O test but with less complex in terms of output. In this test we can perform sequential reads and writes, as well as random reads and writes. Sysbench File I/O allows to modify block size for I/O, and the ability to toggle direct I/O, sync, async, and various other IO related functions.

Staging I/O Test

At the prepare stage SysBench creates a specified number of test files with a specified total size, then at the run stage, each thread performs specified I/O operations on this set of files. When the global --validate option is used with the file I/O test, SysBench performs checksums validation on all data read from the disk. On each write operation the block is filled with random values, then the checksum is calculated and stored in the block along with the offset of this block within a file. On each read operation the block is validated by comparing the stored offset with the real offset, and the stored checksum with the real calculated checksum. The duration of the test is given through the --max-time option (in seconds).

Command Execution:

sysbench --test=fileio --file-total-size=4G –file-num=64 prepare
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndwr --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=1 run
sysbench --test=fileio --file-total-size=4G --file-num=64 cleanup

Result:

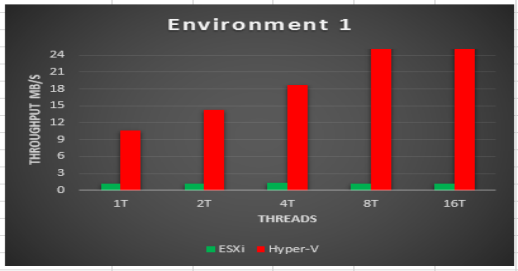
```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     309.95
    fsyncs/s:                     197.64
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.21
General statistics:
    total time:                   60.0044s
    total number of events:       30460
Latency (ms):
         min:                     0.00
         avg:                     1.97
         max:                     117.66
         95th percentile:         9.56
         sum:                     59953.86
```

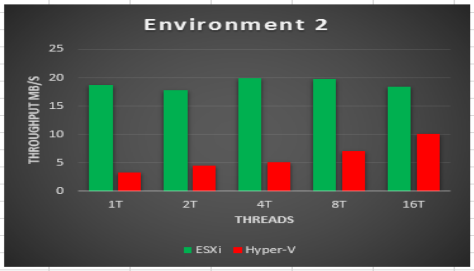
```
Threads fairness:
    events (avg/stddev):          30460.0000/0.00
    execution time (avg/stddev):  59.9539/0.00
```

The important number to compare is the amount of the throughput per specified threads. Tests were repeated with different Thread values (1, 2, 4, 8 and 16) and were performed for Random Write, Random Read and Random Read Write on two Ubuntu desktops hosted in ESXi and Hyper-V environments 1 and 2 to do the fair assessment. Following is the graphical representation of the test results in both environments.

Random Write Test Results Graph:

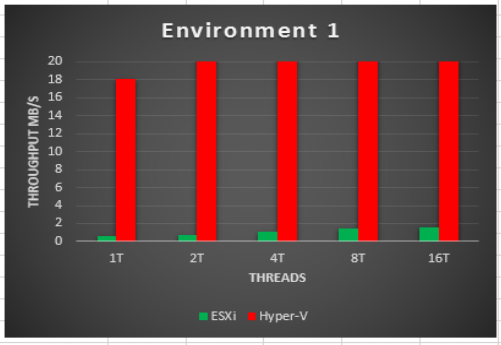
	Environment 1		Environment 2	
Random Write	ESXi	Hyper-V	ESXi	Hyper-V
1T	1.21	10.6	18.77	3.33
2T	1.22	14.21	17.88	4.45
4T	1.34	18.65	19.88	5.08
8T	1.22	26.18	19.78	7.05
16T	1.24	41.31	18.44	10.12

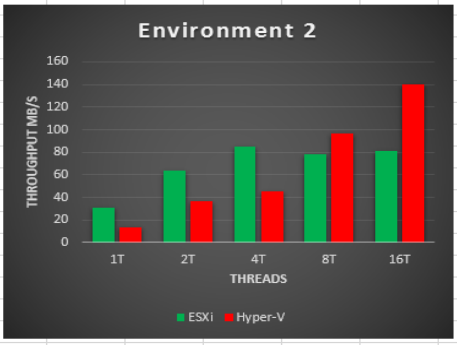




Random Read Test Results Graph:

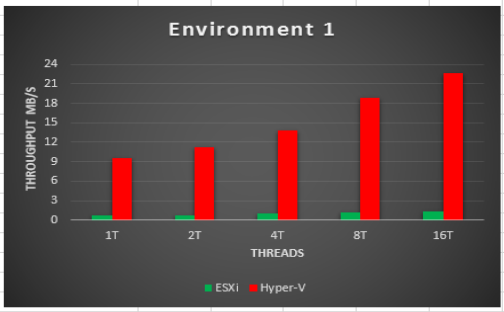
	Environment 1		Environment 2	
Random Read	ESXi	Hyper-V	ESXi	Hyper-V
1T	0.6	18.08	30.98	13.22
2T	0.69	28.33	63.98	36.41
4T	1.04	52.33	84.57	45.09
8T	1.41	75.26	78.5	96.57
16T	1.62	105.52	80.94	140

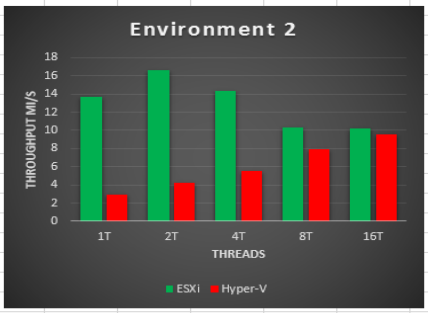




Random Read Write Test Results Graphs:

	Environment 1		Environment 2	
Random Read Write	ESXi	Hyper-V	ESXi	Hyper-V
1T	0.73	9.58	13.69	2.98
2T	0.79	11.18	16.65	4.25
4T	1.02	13.75	14.4	5.58
8T	1.28	18.82	10.35	7.93
16T	1.33	22.63	10.25	9.61





From the I/O test result graphs, we realized the following factors did affected IP read and write performance of the benchmark VMs:

- **Converse result:** There was opposite result between two different environments. Environment 1 is a nested virtualization where any IO requested from the VM is passed through two different VMMs before it reached the underlying physical hardware. However, Environment 2 is straightaway had direct access to underlying hardware.

- **Huge difference in Performance**: Although environment1 result was misleading whereas Environment 2 did proves that Microkernel based VM performance is significantly low as it has to pass through the parent partition through VMBus where as in Monolithic Kernel based VMM directly interact with guest VM through emulated IO device.

Conclusion

Based on the results obtained during the tests, VMware ESXi has slightly better results in CPU performance at higher CPU values when compared with Hyper-V. This is due to VMware ESXi having a powerful CPU scheduling mechanism in place that ensures that virtual machines receive attention from the system.

Memory performance is also better in VMware compared to Hyper-V due to VMware's ability to do memory over commit. VMware's memory management technology enables ESX to automatically reclaim back any physical memory that is not in use and enables another VM to use it, so no memory is wasted. I/O tests performances varied differently between the two environments due to the underlying infrastructure chosen for the analysis.

We observed that that no hypervisor (ESXi or Hyper V) outperformed overwhelmingly each other in all the tests that were performed, although tests clearly indicate that Type 1 hypervisors have great advantages over Type 2 hypervisor solutions, due to the direct access to the system's resources. Hyper-V is micro-kernelized hypervisor leveraging para-virtualization together with full-virtualization (hardware-assisted) while VMware ESXi is a monolithic hypervisor leveraging full-virtualization approach (hardware-assisted).

Ultimately, the choice must be made with consideration for features and cost and that the organizations must select a solution based on its requirements.

Appendix

To simplify the performance tests, we scripted the sysbench test commands. These scripts were executed on Environment 1 and Environment 2. This section provides information about these sysbench scripts used and all the results in detail for the tests performed for this analysis

Environment 1 Scripts

CPU Test Script

```
#!/bin/bash
for prime in 4000 16000 32000 64000 128000 ;do
#CPU Test
echo "Performing CPU test cpu-${prime}"
sysbench --test=cpu --cpu-max-prime=${prime} run > /home/daniel/Desktop/results/cpu-${prime}
done
```

Memory Test Script

```
#!/bin/bash
for thread in 1 2 4 8 16 ;do
#Memory test
echo "Testing mem-${thread}T"
sysbench --test=memory --memory-block-size=1K --memory-scope=global --memory-total-size=100G --memory-oper=write --num-threads=${thread} run > /home/daniel/Desktop/results/memory-write-T-${thread}
sysbench --test=memory --memory-block-size=1K --memory-scope=global --memory-total-size=100G --memory-oper=read --num-threads=${thread} run > /home/daniel/Desktop/results/memory-read-T-${thread}
done
```

I/O Test Script

```
#!/bin/bash
# Testing 1GB file
sysbench --test=fileio --file-total-size=4G --file-num=64 prepare
for thread in 1 2 4 8 16 ;do
# Testing I/O Random Write
echo "Testing RW-${thread}T"
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndwr --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=${thread} run > /home/daniel/Desktop/results/RW-${thread}T
#Testing I/O Random Read
echo "Testing RR-${thread}T"
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndrd --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=${thread} run > /home/daniel/Desktop/results/RR-${thread}T
#Testing I/O Random Read/Write
echo "Testing RRW-${thread}T"
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndrw --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=${thread} run > /home/daniel/Desktop/results/RRW-${thread}T
done
# Delete all the test files after the testing.
sysbench --test=fileio --file-total-size=4G --file-num=64 cleanup
```

Environment 2 Scripts

CPU Test Script

```
#!/bin/bash
for prime in 4000 16000 32000 64000 128000 ;do
#CPU Test
echo "Performing CPU test cpu-${prime}"
sysbench --test=cpu --cpu-max-prime=${prime} run > /root/results/cpu-${prime}
done
```

Memory Test Script

```
#!/bin/bash
for thread in 1 2 4 8 16 ;do
#Memory test
#Sysbench Memory test
echo "Performing memory test mem-${thread}T"
sysbench --test=memory --memory-block-size=1K --memory-scope=global --memory-total-size=100G --memory-oper=write --num-threads=${thread} run > /root/results/memory-write-T-${thread}
sysbench --test=memory --memory-block-size=1K --memory-scope=global --memory-total-size=100G --memory-oper=read --num-threads=${thread} run > /root/results/memory-read-T-${thread}
done
```

I/O Test Script

```
#!/bin/bash
# Testing 1GB file
sysbench --test=fileio --file-total-size=4G --file-num=64 prepare
for thread in 1 2 4 8 16 ;do
# Testing I/O Random Write
echo "Performing test RW-${thread}T"
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndwr --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=${thread} run > /root/results/RW-${thread}T
# Testing I/O Random Read
echo "Performing test RR-${thread}T"
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndrd --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=${thread} run > /root/results/RR-${thread}T
# Testing I/O Random Read/Write
echo "Performing test RRW-${thread}T"
sysbench --test=fileio --file-total-size=4G --file-test-mode=rndrw --max-time=60 --max-requests=0 --file-block-size=4K --file-num=64 --num-threads=${thread} run > /root/results/RRW-${thread}T
done
# Delete all the test files after the testing
sysbench --test=fileio --file-total-size=4G --file-num=64 cleanup
```

Environment 1 Detailed Test Results

VMware ESXi – Ubuntu Testing

CPU Test

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 4000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:  1817.00
General statistics:
    total time:                          10.0005s
    total number of events:              18182
Latency (ms):
         min:                                    0.49
         avg:                                    0.55
         max:                                    9.28
         95th percentile:                        0.73
         sum:                                 9985.84
Threads fairness:
    events (avg/stddev):           18182.0000/0.00
    execution time (avg/stddev):   9.9858/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 16000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:   296.60
General statistics:
    total time:                          10.0015s
    total number of events:              2968
Latency (ms):
         min:                                    3.29
         avg:                                    3.37
         max:                                    5.12
         95th percentile:                        3.36
         sum:                                 9997.34
Threads fairness:
    events (avg/stddev):           2968.0000/0.00
    execution time (avg/stddev):   9.9973/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 32000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:   108.92
General statistics:
    total time:                          10.0043s
    total number of events:              1090
Latency (ms):
         min:                                    8.52
         avg:                                    9.17
         max:                                   15.21
         95th percentile:                       12.30
         sum:                                10000.73
Threads fairness:
    events (avg/stddev):           1090.0000/0.00
    execution time (avg/stddev):   10.0007/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 64000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:    43.02
General statistics:
    total time:                          10.0356s
    total number of events:              431
Latency (ms):
         min:                                   22.22
         avg:                                   23.23
         max:                                   33.20
         95th percentile:                       31.94
         sum:                                10013.01
Threads fairness:
    events (avg/stddev):           431.0000/0.00
    execution time (avg/stddev):   10.0130/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 128000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:    16.62
General statistics:
    total time:                          10.0450s
    total number of events:              167
Latency (ms):
         min:                                   58.03
```

```
      avg:                    60.14
      max:                    83.63
      95th percentile:        77.19
      sum:                 10043.41
Threads fairness:
   events (avg/stddev):       167.0000/0.00
   execution time (avg/stddev):  10.0434/0.00
      sum:                 10076.69
Threads fairness:
   events (avg/stddev):       149.0000/0.00
   execution time (avg/stddev):  10.0767/0.00
```

Memory Test

Memory Read

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 30509220 (3049353.38 per second)
29794.16 MiB transferred (2977.88 MiB/sec)
General statistics:
    total time:                 10.0003s
    total number of events:     30509220
Latency (ms):
      min:                    0.00
      avg:                    0.00
      max:                    0.60
      95th percentile:        0.00
      sum:                 3076.18
Threads fairness:
   events (avg/stddev):       30509220.0000/0.00
   execution time (avg/stddev):  3.0762/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 58890387 (5886124.45 per second)
57510.14 MiB transferred (5748.17 MiB/sec)
General statistics:
    total time:                 10.0002s
    total number of events:     58890387
Latency (ms):
      min:                    0.00
      avg:                    0.00
      max:                   90.24
      95th percentile:        0.00
      sum:                 6013.46
Threads fairness:
   events (avg/stddev):       29445193.5000/520650.50
   execution time (avg/stddev):  3.0067/0.04
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 59552649 (5952782.33 per second)
58156.88 MiB transferred (5813.26 MiB/sec)
General statistics:
    total time:                 10.0006s
    total number of events:     59552649
Latency (ms):
      min:                    0.00
      avg:                    0.00
      max:                   16.05
      95th percentile:        0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 61161931 (6113201.07 per second)
59728.45 MiB transferred (5969.92 MiB/sec)
General statistics:
    total time:                 10.0006s
    total number of events:     61161931
Latency (ms):
      min:                    0.00
      avg:                    0.00
      max:                   48.05
      95th percentile:        0.00
```

```
        sum:                  12316.19
Threads fairness:
    events (avg/stddev):          14888162.2500/134130.36
    execution time (avg/stddev):  3.0790/0.15
```

```
        sum:                  24610.39
Threads fairness:
    events (avg/stddev):          7645241.3750/195566.40
    execution time (avg/stddev):  3.0763/0.17
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 61508112 (6147384.02 per second)
60066.52 MiB transferred (6003.30 MiB/sec)
General statistics:
    total time:                  10.0013s
    total number of events:              61508112
Latency (ms):
         min:                    0.00
         avg:                    0.00
         max:                   44.05
         95th percentile:        0.00
         sum:                46184.03
Threads fairness:
    events (avg/stddev):          3844257.0000/70800.20
    execution time (avg/stddev):  2.8865/0.23
```

Memory Write

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 24744251 (2472977.91 per second)
24164.31 MiB transferred (2415.02 MiB/sec)
General statistics:
    total time:                  10.0004s
    total number of events:              24744251
Latency (ms):
         min:                    0.00
         avg:                    0.00
         max:                    0.60
         95th percentile:        0.00
         sum:                 4558.86
Threads fairness:
    events (avg/stddev):          24744251.0000/0.00
    execution time (avg/stddev):  4.5589/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 33393613 (3337888.77 per second)
32610.95 MiB transferred (3259.66 MiB/sec)
General statistics:
    total time:                  10.0003s
    total number of events:              33393613
Latency (ms):
         min:                    0.00
         avg:                    0.00
         max:                  110.19
         95th percentile:        0.00
         sum:                11952.17
Threads fairness:
    events (avg/stddev):          16696806.5000/72668.50
    execution time (avg/stddev):  5.9761/0.12
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 35360346 (3534751.77 per second)
34531.59 MiB transferred (3451.91 MiB/sec)
General statistics:
    total time:                          10.0004s
    total number of events:              35360346
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   16.04
         95th percentile:                        0.00
         sum:                                23686.01
Threads fairness:
    events (avg/stddev):           8840086.5000/69122.55
    execution time (avg/stddev):   5.9215/0.20
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 34926254 (3491409.04 per second)
34107.67 MiB transferred (3409.58 MiB/sec)
General statistics:
    total time:                          10.0005s
    total number of events:              34926254
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                  446.54
         95th percentile:                        0.00
         sum:                                45412.44
Threads fairness:
    events (avg/stddev):           4365781.7500/58260.87
    execution time (avg/stddev):   5.6766/0.14
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 34944836 (3493103.11 per second)
34125.82 MiB transferred (3411.23 MiB/sec)
General statistics:
    total time:                          10.0009s
    total number of events:              34944836
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                  953.50
         95th percentile:                        0.00
         sum:                                93666.93
Threads fairness:
    events (avg/stddev):           2184052.2500/75602.06
    execution time (avg/stddev):   5.8542/0.19
```

File I/O Test

Random Write

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
```

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
```

```
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     309.95
    fsyncs/s:                     197.64
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.21
General statistics:
    total time:                   60.0044s
    total number of events:       30460
Latency (ms):
         min:                     0.00
         avg:                     1.97
         max:                     117.66
         95th percentile:         9.56
         sum:                     59953.86
Threads fairness:
    events (avg/stddev):          30460.0000/0.00
    execution time (avg/stddev):  59.9539/0.00
```

```
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     313.25
    fsyncs/s:                     200.38
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.22
General statistics:
    total time:                   60.0121s
    total number of events:       30826
Latency (ms):
         min:                     0.00
         avg:                     3.89
         max:                     269.08
         95th percentile:         17.95
         sum:                     119966.63
Threads fairness:
    events (avg/stddev):          15413.0000/75.00
    execution time (avg/stddev):  59.9833/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     343.20
    fsyncs/s:                     219.53
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.34
General statistics:
    total time:                   60.0188s
    total number of events:       33777
Latency (ms):
         min:                     0.00
         avg:                     7.10
         max:                     167.74
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     313.18
    fsyncs/s:                     199.44
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               1.22
General statistics:
    total time:                   60.0244s
    total number of events:       30772
Latency (ms):
         min:                     0.00
         avg:                     15.60
         max:                     798.57
```

95th percentile: 30.26 sum: 239978.52 Threads fairness: events (avg/stddev): 8444.2500/941.24 execution time (avg/stddev): 59.9946/0.01	95th percentile: 68.05 sum: 480121.25 Threads fairness: events (avg/stddev): 3846.5000/428.32 execution time (avg/stddev): 60.0152/0.00
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random write test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 318.20 fsyncs/s: 202.63 Throughput: read, MiB/s: 0.00 written, MiB/s: 1.24 General statistics: total time: 60.0199s total number of events: 31263 Latency (ms): min: 0.00 avg: 30.71 max: 4921.74 95th percentile: 144.97 sum: 960228.99 Threads fairness: events (avg/stddev): 1953.9375/335.75 execution time (avg/stddev): 60.0143/0.00	

Random Read

WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started!	WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started!

```
File operations:
    reads/s:                      153.03
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  0.60
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0081s
    total number of events:       9184
Latency (ms):
         min:                     0.00
         avg:                     6.53
         max:                     100.14
         95th percentile:         11.04
         sum:                     59976.46
Threads fairness:
    events (avg/stddev):          9184.0000/0.00
    execution time (avg/stddev):  59.9765/0.00
```

```
File operations:
    reads/s:                      177.75
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  0.69
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0079s
    total number of events:       10667
Latency (ms):
         min:                     0.00
         avg:                     11.25
         max:                     175.84
         95th percentile:         18.28
         sum:                     119973.78
Threads fairness:
    events (avg/stddev):          5333.5000/16.50
    execution time (avg/stddev):  59.9869/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      267.06
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  1.04
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0336s
    total number of events:       16034
Latency (ms):
         min:                     0.00
         avg:                     14.97
         max:                     350.55
         95th percentile:         38.25
         sum:                     240038.03
Threads fairness:
    events (avg/stddev):          4008.5000/84.17
    execution time (avg/stddev):  60.0095/0.01
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      362.09
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  1.41
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0326s
    total number of events:       21739
Latency (ms):
         min:                     0.00
         avg:                     22.08
         max:                     311.39
         95th percentile:         69.29
         sum:                     480050.30
Threads fairness:
    events (avg/stddev):          2717.3750/65.93
    execution time (avg/stddev):  60.0063/0.01
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
```

```
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      415.73
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  1.62
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0631s
    total number of events:       24972
Latency (ms):
         min:                     0.00
         avg:                     38.46
         max:                     1627.96
         95th percentile:         132.49
         sum:                     960349.30
Threads fairness:
    events (avg/stddev):          1560.7500/46.19
    execution time (avg/stddev):  60.0218/0.02
```

Random Read/Write

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      111.99
    writes/s:                     74.66
    fsyncs/s:                     118.64
Throughput:
    read, MiB/s:                  0.44
    written, MiB/s:               0.29
General statistics:
    total time:                   60.0026s
    total number of events:       18319
Latency (ms):
         min:                     0.00
```

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      121.51
    writes/s:                     80.98
    fsyncs/s:                     129.04
Throughput:
    read, MiB/s:                  0.47
    written, MiB/s:               0.32
General statistics:
    total time:                   60.0077s
    total number of events:       19896
Latency (ms):
         min:                     0.00
```

```
      avg:                  3.27
      max:                125.77
      95th percentile:     10.27
      sum:              59958.07
Threads fairness:
   events (avg/stddev):          18319.0000/0.00
   execution time (avg/stddev):  59.9581/0.00
```

```
      avg:                  6.03
      max:                184.67
      95th percentile:     17.32
      sum:              119961.83
Threads fairness:
   events (avg/stddev):          9948.0000/13.00
   execution time (avg/stddev):  59.9809/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                157.19
   writes/s:               104.76
   fsyncs/s:               167.42
Throughput:
   read, MiB/s:              0.61
   written, MiB/s:           0.41
General statistics:
   total time:                 60.0117s
   total number of events:          25769
Latency (ms):
      min:                    0.00
      avg:                    9.31
      max:                  326.82
      95th percentile:       31.37
      sum:                239967.83
Threads fairness:
   events (avg/stddev):          6442.2500/284.77
   execution time (avg/stddev):  59.9920/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                197.70
   writes/s:               131.72
   fsyncs/s:               209.97
Throughput:
   read, MiB/s:              0.77
   written, MiB/s:           0.51
General statistics:
   total time:                 60.0411s
   total number of events:          32388
Latency (ms):
      min:                    0.00
      avg:                   14.82
      max:                 1586.48
      95th percentile:       57.87
      sum:                480072.37
Threads fairness:
   events (avg/stddev):          4048.5000/193.47
   execution time (avg/stddev):  60.0090/0.01
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
```

File operations: reads/s: 203.82 writes/s: 135.71 fsyncs/s: 217.03 Throughput: read, MiB/s: 0.80 written, MiB/s: 0.53 General statistics: total time: 60.0787s total number of events: 33440 Latency (ms): min: 0.00 avg: 28.72 max: 2160.62 95th percentile: 196.89 sum: 960455.11 Threads fairness: events (avg/stddev): 2090.0000/126.12 execution time (avg/stddev): 60.0284/0.02	

Microsoft Hyper-V – Ubuntu Testing

CPU Test

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 4000 Initializing worker threads... Threads started! CPU speed: events per second: 1727.50 General statistics: total time: 10.0001s total number of events: 17281 Latency (ms): min: 0.49 avg: 0.58 max: 10.28 95th percentile: 0.73 sum: 9984.15 Threads fairness: events (avg/stddev): 17281.0000/0.00 execution time (avg/stddev): 9.9841/0.00	sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 16000 Initializing worker threads... Threads started! CPU speed: events per second: 281.16 General statistics: total time: 10.0033s total number of events: 2814 Latency (ms): min: 2.64 avg: 3.55 max: 13.82 95th percentile: 4.74 sum: 9997.72 Threads fairness: events (avg/stddev): 2814.0000/0.00 execution time (avg/stddev): 9.9977/0.00
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 32000 Initializing worker threads... Threads started! CPU speed: events per second: 113.97 General statistics: total time: 10.0070s total number of events: 1141 Latency (ms): min: 8.49 avg: 8.77 max: 12.97	sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 64000 Initializing worker threads... Threads started! CPU speed: events per second: 32.99 General statistics: total time: 10.0289s total number of events: 331 Latency (ms): min: 22.16 avg: 30.29 max: 41.54

95th percentile: 9.06 sum: 10003.53 Threads fairness: events (avg/stddev): 1141.0000/0.00 execution time (avg/stddev): 10.0035/0.00	95th percentile: 38.94 sum: 10024.58 Threads fairness: events (avg/stddev): 331.0000/0.00 execution time (avg/stddev): 10.0246/0.00
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 128000 Initializing worker threads... Threads started! CPU speed: events per second: 12.84 General statistics: total time: 10.0379s total number of events: 129 Latency (ms): min: 58.05 avg: 77.80 max: 130.48 95th percentile: 101.13 sum: 10036.04 Threads fairness: events (avg/stddev): 129.0000/0.00 execution time (avg/stddev): 10.0360/0.00	

Memory Test

Memory Read

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 26506547 (2649730.38 per second) 25885.30 MiB transferred (2587.63 MiB/sec) General statistics: total time: 10.0001s total number of events: 26506547 Latency (ms): min: 0.00 avg: 0.00 max: 9.00 95th percentile: 0.00 sum: 3003.87 Threads fairness: events (avg/stddev): 26506547.0000/0.00 execution time (avg/stddev): 3.0039/0.00	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 52686942 (5266663.50 per second) 51452.09 MiB transferred (5143.23 MiB/sec) General statistics: total time: 10.0004s total number of events: 52686942 Latency (ms): min: 0.00 avg: 0.00 max: 10.02 95th percentile: 0.00 sum: 5944.36 Threads fairness: events (avg/stddev): 26343471.0000/971668.00 execution time (avg/stddev): 2.9722/0.00
WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8

```
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 51574101 (5155075.86 per second)
50365.33 MiB transferred (5034.25 MiB/sec)
General statistics:
    total time:                          10.0013s
    total number of events:              51574101
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   27.12
         95th percentile:                        0.00
         sum:                                12113.82
Threads fairness:
    events (avg/stddev):           12893525.2500/486732.67
    execution time (avg/stddev):   3.0285/0.14
```

```
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 51499654 (5147279.87 per second)
50292.63 MiB transferred (5026.64 MiB/sec)
General statistics:
    total time:                          10.0010s
    total number of events:              51499654
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   52.03
         95th percentile:                        0.00
         sum:                                22943.94
Threads fairness:
    events (avg/stddev):           6437456.7500/792907.47
    execution time (avg/stddev):   2.8680/0.08
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 52538058 (5251251.97 per second)
51306.70 MiB transferred (5128.18 MiB/sec)
General statistics:
    total time:                          10.0019s
    total number of events:              52538058
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                  113.90
         95th percentile:                        0.00
         sum:                                47361.88
Threads fairness:
    events (avg/stddev):           3283628.6250/134044.94
    execution time (avg/stddev):   2.9601/0.24
```

Memory Write

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
```

```
Initializing worker threads...
Threads started!
Total operations: 18791068 (1878361.67 per second)
18350.65 MiB transferred (1834.34 MiB/sec)
General statistics:
   total time:                          10.0002s
   total number of events:              18791068
Latency (ms):
        min:                                    0.00
        avg:                                    0.00
        max:                                    9.28
        95th percentile:                        0.00
        sum:                                 4408.94
Threads fairness:
   events (avg/stddev):           18791068.0000/0.00
   execution time (avg/stddev):   4.4089/0.00
```

```
 scope: global
Initializing worker threads...
Threads started!
Total operations: 32970621 (3295781.65 per second)
32197.87 MiB transferred (3218.54 MiB/sec)
General statistics:
   total time:                          10.0004s
   total number of events:              32970621
Latency (ms):
        min:                                    0.00
        avg:                                    0.00
        max:                                   12.83
        95th percentile:                        0.00
        sum:                                11395.27
Threads fairness:
   events (avg/stddev):           16485310.5000/789829.50
   execution time (avg/stddev):   5.6976/0.16
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Running memory speed test with the following options:
 block size: 1KiB
 total size: 102400MiB
 operation: write
 scope: global
Initializing worker threads...
Threads started!
Total operations: 32521497 (3250491.99 per second)
31759.27 MiB transferred (3174.31 MiB/sec)
General statistics:
   total time:                          10.0006s
   total number of events:              32521497
Latency (ms):
        min:                                    0.00
        avg:                                    0.00
        max:                                   25.01
        95th percentile:                        0.00
       sum:                                 22551.91
Threads fairness:
   events (avg/stddev):           8130374.2500/250516.25
   execution time (avg/stddev):   5.6380/0.15
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Running memory speed test with the following options:
 block size: 1KiB
 total size: 102400MiB
 operation: write
 scope: global
Initializing worker threads...
Threads started!
Total operations: 29636476 (2962060.45 per second)
28941.87 MiB transferred (2892.64 MiB/sec)
General statistics:
   total time:                          10.0010s
   total number of events:              29636476
Latency (ms):
        min:                                    0.00
        avg:                                    0.00
        max:                                   56.51
        95th percentile:                        0.00
        sum:                                45169.20
Threads fairness:
   events (avg/stddev):           3704559.5000/159422.21
   execution time (avg/stddev):   5.6462/0.34
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Running memory speed test with the following options:
 block size: 1KiB
 total size: 102400MiB
 operation: write
 scope: global
Initializing worker threads...
Threads started!
Total operations: 29423954 (2941032.10 per second)
28734.33 MiB transferred (2872.10 MiB/sec)
General statistics:
   total time:                          10.0016s
   total number of events:              29423954
Latency (ms):
```

min: 0.00 avg: 0.00 max: 113.94 95th percentile: 0.00 sum: 92073.38 Threads fairness: events (avg/stddev): 1838997.1250/37892.60 execution time (avg/stddev): 5.7546/0.29	

File I/O Test

Random Write

WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random write test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 2714.79 fsyncs/s: 1737.08 Throughput: read, MiB/s: 0.00 written, MiB/s: 10.60 General statistics: total time: 60.0002s total number of events: 267133 Latency (ms): min: 0.00 avg: 0.22 max: 40.03 95th percentile: 0.57 sum: 59657.89 Threads fairness: events (avg/stddev): 267133.0000/0.00 execution time (avg/stddev): 59.6579/0.00	WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random write test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 3638.05 fsyncs/s: 2328.20 Throughput: read, MiB/s: 0.00 written, MiB/s: 14.21 General statistics: total time: 60.0004s total number of events: 358003 Latency (ms): min: 0.00 avg: 0.33 max: 105.26 95th percentile: 0.75 sum: 119465.92 Threads fairness: events (avg/stddev): 179001.5000/3378.50 execution time (avg/stddev): 59.7330/0.00
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size	WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size

```
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     4774.55
    fsyncs/s:                     3055.08
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               18.65
General statistics:
    total time:                   60.0010s
    total number of events:       469822
Latency (ms):
         min:                     0.00
         avg:                     0.51
         max:                     46.77
         95th percentile:         1.37
         sum:                     239336.59
Threads fairness:
    events (avg/stddev):          117455.5000/1507.62
    execution time (avg/stddev):  59.8341/0.01
```

```
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     6702.16
    fsyncs/s:                     4288.46
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               26.18
General statistics:
    total time:                   60.0060s
    total number of events:       659553
Latency (ms):
         min:                     0.00
         avg:                     0.73
         max:                     100.20
         95th percentile:         2.66
         sum:                     479006.54
Threads fairness:
    events (avg/stddev):          82444.1250/2118.23
    execution time (avg/stddev):  59.8758/0.01
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     10575.24
    fsyncs/s:                     6767.32
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               41.31
General statistics:
    total time:                   60.0038s
    total number of events:       1040694
Latency (ms):
         min:                     0.00
         avg:                     0.92
         max:                     203.56
         95th percentile:         3.75
         sum:                     958131.09
Threads fairness:
```

events (avg/stddev): 65043.3750/2422.94 execution time (avg/stddev): 59.8832/0.01	

Random Read

WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 4627.35 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 18.08 written, MiB/s: 0.00 General statistics: total time: 60.0028s total number of events: 277672 Latency (ms): min: 0.00 avg: 0.21 max: 45.26 95th percentile: 0.28 sum: 59603.55 Threads fairness: events (avg/stddev): 277672.0000/0.00 execution time (avg/stddev): 59.6035/0.00	WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 7253.61 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 28.33 written, MiB/s: 0.00 General statistics: total time: 60.1535s total number of events: 436364 Latency (ms): min: 0.00 avg: 0.27 max: 123.66 95th percentile: 0.35 sum: 119390.72 Threads fairness: events (avg/stddev): 218182.0000/1471.00 execution time (avg/stddev): 59.6954/0.00
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations:	WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations:

reads/s: 13370.63 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 52.23 written, MiB/s: 0.00 General statistics: total time: 60.0880s total number of events: 803497 Latency (ms): min: 0.00 avg: 0.30 max: 1014.88 95th percentile: 0.36 sum: 238945.11 Threads fairness: events (avg/stddev): 200874.2500/3209.90 execution time (avg/stddev): 59.7363/0.03	reads/s: 19267.66 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 75.26 written, MiB/s: 0.00 General statistics: total time: 60.1622s total number of events: 1159244 Latency (ms): min: 0.00 avg: 0.41 max: 1019.02 95th percentile: 0.53 sum: 478431.01 Threads fairness: events (avg/stddev): 144905.5000/1051.74 execution time (avg/stddev): 59.8039/0.02
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 27011.85 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 105.52 written, MiB/s: 0.00 General statistics: total time: 60.1525s total number of events: 1624948 Latency (ms): min: 0.00 avg: 0.59 max: 923.26 95th percentile: 0.70 sum: 957708.86 Threads fairness: events (avg/stddev): 101559.2500/1991.50 execution time (avg/stddev): 59.8568/0.02	

Random Read/Write

WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1	WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2

```
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      1471.60
    writes/s:                     981.06
    fsyncs/s:                     1568.94
Throughput:
    read, MiB/s:                  5.75
    written, MiB/s:               3.83
General statistics:
    total time:                   60.0003s
    total number of events:       241315
Latency (ms):
         min:                    0.00
         avg:                    0.25
         max:                    128.18
         95th percentile:        0.42
         sum:                    59644.77
Threads fairness:
    events (avg/stddev):          241315.0000/0.00
    execution time (avg/stddev):  59.6448/0.00
```

```
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      1716.86
    writes/s:                     1144.57
    fsyncs/s:                     1830.60
Throughput:
    read, MiB/s:                  6.71
    written, MiB/s:               4.47
General statistics:
    total time:                   60.0004s
    total number of events:       281545
Latency (ms):
         min:                    0.00
         avg:                    0.42
         max:                    524.39
         95th percentile:        0.73
         sum:                    119561.10
Threads fairness:
    events (avg/stddev):          140772.5000/2763.50
    execution time (avg/stddev):  59.7805/0.01
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      2110.82
    writes/s:                     1407.19
    fsyncs/s:                     2250.84
Throughput:
    read, MiB/s:                  8.25
    written, MiB/s:               5.50
General statistics:
    total time:                   60.0014s
    total number of events:       346163
Latency (ms):
         min:                    0.00
         avg:                    0.69
         max:                    186.49
         95th percentile:        1.39
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      2891.34
    writes/s:                     1927.50
    fsyncs/s:                     3083.16
Throughput:
    read, MiB/s:                  11.29
    written, MiB/s:               7.53
General statistics:
    total time:                   60.0102s
    total number of events:       474234
Latency (ms):
         min:                    0.00
         avg:                    1.01
         max:                    222.12
         95th percentile:        2.97
```

sum: 239422.77 Threads fairness: events (avg/stddev): 86540.7500/766.31 execution time (avg/stddev): 59.8557/0.01	sum: 479321.24 Threads fairness: events (avg/stddev): 59279.2500/557.43 execution time (avg/stddev): 59.9152/0.00
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random r/w test Initializing worker threads... Threads started! File operations: reads/s: 3477.22 writes/s: 2317.98 fsyncs/s: 3708.73 Throughput: read, MiB/s: 13.58 written, MiB/s: 9.05 General statistics: total time: 60.0108s total number of events: 570380 Latency (ms): min: 0.00 avg: 1.68 max: 189.42 95th percentile: 7.84 sum: 959149.72 Threads fairness: events (avg/stddev): 35648.7500/304.50 execution time (avg/stddev): 59.9469/0.00	

Environment 2 Detailed Test Results

VMware ESXi – Ubuntu Testing

CPU Test

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 4000 Initializing worker threads... Threads started! CPU speed: events per second: 3029.08 General statistics: total time: 10.0003s total number of events: 30297	sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 16000 Initializing worker threads... Threads started! CPU speed: events per second: 467.75 General statistics: total time: 10.0014s total number of events: 4679

Latency (ms): min: 0.27 avg: 0.33 max: 26.40 95th percentile: 0.47 sum: 9951.44 Threads fairness: events (avg/stddev): 30297.0000/0.00 execution time (avg/stddev): 9.9514/0.00	Latency (ms): min: 1.79 avg: 2.13 max: 5.56 95th percentile: 2.91 sum: 9986.21 Threads fairness: events (avg/stddev): 4679.0000/0.00 execution time (avg/stddev): 9.9862/0.00
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 32000 Initializing worker threads... Threads started! CPU speed: events per second: 180.52 General statistics: total time: 10.0017s total number of events: 1806 Latency (ms): min: 4.63 avg: 5.53 max: 16.46 95th percentile: 6.91 sum: 9992.86 Threads fairness: events (avg/stddev): 1806.0000/0.00 execution time (avg/stddev): 9.9929/0.00	sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 64000 Initializing worker threads... Threads started! CPU speed: events per second: 70.00 General statistics: total time: 10.0122s total number of events: 701 Latency (ms): min: 12.54 avg: 14.28 max: 23.75 95th percentile: 17.01 sum: 10006.87 Threads fairness: events (avg/stddev): 701.0000/0.00 execution time (avg/stddev): 10.0069/0.00
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Prime numbers limit: 128000 Initializing worker threads... Threads started! CPU speed: events per second: 25.93 General statistics: total time: 10.0520s total number of events: 260 Latency (ms): min: 35.17 avg: 38.55 max: 72.03 95th percentile: 49.21 sum: 10023.01 Threads fairness: events (avg/stddev): 260.0000/0.00 execution time (avg/stddev): 10.0230/0.00	

Memory Test

Memory Read

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Running memory speed test with the following options:	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time

```
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 39180455 (3917297.32 per second)
38262.16 MiB transferred (3825.49 MiB/sec)
General statistics:
    total time:                          10.0001s
    total number of events:              39180455
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    5.04
         95th percentile:                        0.00
         sum:                                 3158.26
Threads fairness:
    events (avg/stddev):           39180455.0000/0.00
    execution time (avg/stddev):   3.1583/0.00
```

```
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 75734978 (7569509.73 per second)
73959.94 MiB transferred (7392.10 MiB/sec)
General statistics:
    total time:                          10.0002s
    total number of events:              75734978
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                    6.01
         95th percentile:                        0.00
         sum:                                 6486.37
Threads fairness:
    events (avg/stddev):           37867489.0000/95996.00
    execution time (avg/stddev):   3.2432/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 37919051 (3788170.04 per second)
37030.32 MiB transferred (7153.01 MiB/sec)
General statistics:
    total time:                          10.0002s
    total number of events:              37919051
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   45.85
         95th percentile:                        0.00
         sum:                                13088.86
Threads fairness:
    events (avg/stddev):           9479762.7500/282207.05
    execution time (avg/stddev):   3.2722/0.15
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 68926769 (6891256.86 per second)
67311.30 MiB transferred (6729.74 MiB/sec)
General statistics:
    total time:                          10.0005s
    total number of events:              68926769
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   67.87
         95th percentile:                        0.00
         sum:                                26439.11
Threads fairness:
    events (avg/stddev):           8615846.1250/130229.61
    execution time (avg/stddev):   3.3049/0.15
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: read
  scope: global
Initializing worker threads...
Threads started!
Total operations: 76930443 (7691220.98 per second)
75127.39 MiB transferred (7510.96 MiB/sec)
```

```
General statistics:
   total time:                          10.0007s
   total number of events:              76930443
Latency (ms):
        min:                                  0.00
        avg:                                  0.00
        max:                                277.76
        95th percentile:                      0.00
        sum:                              50446.01
Threads fairness:
   events (avg/stddev):           4808152.6875/97318.59
   execution time (avg/stddev):   3.1529/0.17
```

Memory Write

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 34943779 (3493707.65 per second)
34124.78 MiB transferred (3411.82 MiB/sec)
General statistics:
   total time:                          10.0001s
   total number of events:              34943779
Latency (ms):
        min:                                  0.00
        avg:                                  0.00
        max:                                  3.93
        95th percentile:                      0.00
        sum:                               4495.10
Threads fairness:
   events (avg/stddev):           34943779.0000/0.00
   execution time (avg/stddev):   4.4951/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 47147751 (4713603.63 per second)
46042.73 MiB transferred (4603.13 MiB/sec)
General statistics:
   total time:                          10.0002s
   total number of events:              47147751
Latency (ms):
        min:                                  0.00
        avg:                                  0.00
        max:                                  6.79
        95th percentile:                      0.00
        sum:                              10816.30
Threads fairness:
   events (avg/stddev):           23573875.5000/40791.50
   execution time (avg/stddev):   5.4081/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 32414893 (3239330.02 per second)
31655.17 MiB transferred (3163.41 MiB/sec)
General statistics:
   total time:                          10.0041s
   total number of events:              32414893
Latency (ms):
        min:                                  0.00
        avg:                                  0.00
        max:                                 66.68
        95th percentile:                      0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 29628662 (2962265.47 per second)
28934.24 MiB transferred (2892.84 MiB/sec)
General statistics:
   total time:                          10.0003s
   total number of events:              29628662
Latency (ms):
        min:                                  0.00
        avg:                                  0.00
        max:                                 58.02
        95th percentile:                      0.00
```

sum: 19154.02 Threads fairness: events (avg/stddev): 8103723.2500/478338.60 execution time (avg/stddev): 4.7885/0.16	sum: 37110.15 Threads fairness: events (avg/stddev): 3703582.7500/138014.07 execution time (avg/stddev): 4.6388/0.20
WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 47024972 (4701278.13 per second) 45922.82 MiB transferred (4591.09 MiB/sec) General statistics: total time: 10.0011s total number of events: 47024972 Latency (ms): min: 0.00 avg: 0.00 max: 284.77 95th percentile: 0.00 sum: 84985.06 Threads fairness: events (avg/stddev): 2939060.7500/56067.71 execution time (avg/stddev): 5.3116/0.24	

File I/O Test

Random Write

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random write test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 4804.79 fsyncs/s: 3074.20 Throughput: read, MiB/s: 0.00 written, MiB/s: 18.77 General statistics: total time: 60.0002s total number of events: 472760	sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random write test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 4576.52 fsyncs/s: 2928.29 Throughput: read, MiB/s: 0.00 written, MiB/s: 17.88 General statistics: total time: 60.0004s total number of events: 450303

```
Latency (ms):
     min:                                    0.00
     avg:                                    0.13
     max:                                   80.68
     95th percentile:                        0.28
     sum:                                59436.30
Threads fairness:
   events (avg/stddev):           472760.0000/0.00
   execution time (avg/stddev):   59.4363/0.00
```

```
Latency (ms):
     min:                                    0.00
     avg:                                    0.26
     max:                                  119.43
     95th percentile:                        0.74
     sum:                               119159.27
Threads fairness:
   events (avg/stddev):           225151.5000/61.50
   execution time (avg/stddev):   59.5796/0.01
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                      0.00
   writes/s:                     5089.80
   fsyncs/s:                     3256.99
Throughput:
   read, MiB/s:                  0.00
   written, MiB/s:               19.88
General statistics:
   total time:                   60.0004s
   total number of events:       500827
Latency (ms):
     min:                                    0.00
     avg:                                    0.48
     max:                                   75.93
     95th percentile:                        1.58
     sum:                               238814.01
Threads fairness:
   events (avg/stddev):           125206.7500/7546.54
   execution time (avg/stddev):   59.7035/0.02
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                      0.00
   writes/s:                     5064.32
   fsyncs/s:                     3240.35
Throughput:
   read, MiB/s:                  0.00
   written, MiB/s:               19.78
General statistics:
   total time:                   60.0060s
   total number of events:       498347
Latency (ms):
     min:                                    0.00
     avg:                                    0.96
     max:                                  126.27
     95th percentile:                        9.73
     sum:                               478698.90
Threads fairness:
   events (avg/stddev):           62293.3750/6620.50
   execution time (avg/stddev):   59.8374/0.02
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
```

```
File operations:
    reads/s:                      0.00
    writes/s:                     4720.44
    fsyncs/s:                     3020.43
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               18.44
General statistics:
    total time:                   60.0091s
    total number of events:       464573
Latency (ms):
         min:                     0.00
         avg:                     2.06
         max:                     66.47
         95th percentile:         14.73
         sum:                     958816.20
Threads fairness:
    events (avg/stddev):          29035.8125/2953.58
    execution time (avg/stddev):  59.9260/0.01
```

Random Read

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      7930.47
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  30.98
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0037s
    total number of events:       475871
Latency (ms):
         min:                     0.00
         avg:                     0.12
         max:                     92.71
         95th percentile:         0.30
         sum:                     59235.48
Threads fairness:
    events (avg/stddev):          475871.0000/0.00
    execution time (avg/stddev):  59.2355/0.00
```

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      16378.73
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  63.98
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0037s
    total number of events:       982843
Latency (ms):
         min:                     0.00
         avg:                     0.12
         max:                     188.17
         95th percentile:         0.29
         sum:                     118511.72
Threads fairness:
    events (avg/stddev):          491421.5000/9125.50
    execution time (avg/stddev):  59.2559/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
```

```
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      21651.02
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  84.57
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0217s
    total number of events:       1299601
Latency (ms):
         min:                     0.00
         avg:                     0.18
         max:                     893.03
         95th percentile:         0.42
         sum:                     233772.47
Threads fairness:
    events (avg/stddev):          324900.2500/5459.54
    execution time (avg/stddev):  58.4431/0.07
```

```
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      20095.10
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  78.50
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0042s
    total number of events:       1205851
Latency (ms):
         min:                     0.00
         avg:                     0.39
         max:                     1018.63
         95th percentile:         1.18
         sum:                     468873.33
Threads fairness:
    events (avg/stddev):          150731.3750/2996.76
    execution time (avg/stddev):  58.6092/0.12
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      20721.92
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  80.94
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0052s
    total number of events:       1243545
Latency (ms):
         min:                     0.00
```

avg: 0.75 max: 949.41 95th percentile: 2.66 sum: 937701.70 Threads fairness: events (avg/stddev): 77721.5625/1052.95 execution time (avg/stddev): 58.6064/0.15	

Random Read/Write

WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random r/w test Initializing worker threads... Threads started! File operations: reads/s: 2102.70 writes/s: 1401.80 fsyncs/s: 2241.87 Throughput: read, MiB/s: 8.21 written, MiB/s: 5.48 General statistics: total time: 60.0072s total number of events: 344832 Latency (ms): min: 0.00 avg: 0.17 max: 54.21 95th percentile: 0.43 sum: 59350.29 Threads fairness: events (avg/stddev): 344832.0000/0.00 execution time (avg/stddev): 59.3503/0.00	WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random r/w test Initializing worker threads... Threads started! File operations: reads/s: 2557.30 writes/s: 1704.85 fsyncs/s: 2727.31 Throughput: read, MiB/s: 9.99 written, MiB/s: 6.66 General statistics: total time: 60.0013s total number of events: 419392 Latency (ms): min: 0.00 avg: 0.28 max: 128.75 95th percentile: 0.62 sum: 119108.09 Threads fairness: events (avg/stddev): 209696.0000/2087.00 execution time (avg/stddev): 59.5540/0.01
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random r/w test	WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode

```
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      2211.15
    writes/s:                     1474.07
    fsyncs/s:                     2357.54
Throughput:
    read, MiB/s:                  8.64
    written, MiB/s:               5.76
General statistics:
    total time:                   60.0106s
    total number of events:       362708
Latency (ms):
         min:                     0.00
         avg:                     0.66
         max:                     142.51
         95th percentile:         2.52
         sum:                     238221.95
Threads fairness:
    events (avg/stddev):          90677.0000/2255.33
    execution time (avg/stddev):  59.5555/0.03
```

```
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      1590.59
    writes/s:                     1060.31
    fsyncs/s:                     1695.88
Throughput:
    read, MiB/s:                  6.21
    written, MiB/s:               4.14
General statistics:
    total time:                   60.0108s
    total number of events:       260882
Latency (ms):
         min:                     0.00
         avg:                     1.83
         max:                     234.44
         95th percentile:         11.45
         sum:                     478439.08
Threads fairness:
    events (avg/stddev):          32610.2500/753.72
    execution time (avg/stddev):  59.8049/0.01
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      1574.95
    writes/s:                     1049.94
    fsyncs/s:                     1679.75
Throughput:
    read, MiB/s:                  6.15
    written, MiB/s:               4.10
General statistics:
    total time:                   60.0068s
    total number of events:       258317
Latency (ms):
         min:                     0.00
         avg:                     3.71
         max:                     294.12
         95th percentile:         16.71
         sum:                     958402.11
Threads fairness:
    events (avg/stddev):          16144.8125/220.74
    execution time (avg/stddev):  59.9001/0.01
```

Microsoft Hyper-V – Ubuntu Testing

CPU Test

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 4000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:  3026.34
General statistics:
    total time:                          10.0004s
    total number of events:              30272
Latency (ms):
         min:                                    0.27
         avg:                                    0.33
         max:                                   10.09
         95th percentile:                        0.46
         sum:                                 9944.12
Threads fairness:
    events (avg/stddev):           30272.0000/0.00
    execution time (avg/stddev):   9.9441/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 16000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:   468.11
General statistics:
    total time:                          10.0012s
    total number of events:              4683
Latency (ms):
         min:                                    1.80
         avg:                                    2.13
         max:                                    6.64
         95th percentile:                        2.71
         sum:                                 9983.32
Threads fairness:
    events (avg/stddev):           4683.0000/0.00
    execution time (avg/stddev):   9.9833/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 32000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:   173.19
General statistics:
    total time:                          10.0035s
    total number of events:              1733
Latency (ms):
         min:                                    4.75
         avg:                                    5.77
         max:                                   13.46
         95th percentile:                        7.84
         sum:                                 9993.28
Threads fairness:
    events (avg/stddev):           1733.0000/0.00
    execution time (avg/stddev):   9.9933/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 64000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:    69.22
General statistics:
    total time:                          10.0100s
    total number of events:              693
Latency (ms):
         min:                                   12.68
         avg:                                   14.44
         max:                                   21.99
         95th percentile:                       16.41
         sum:                                10004.51
Threads fairness:
    events (avg/stddev):           693.0000/0.00
    execution time (avg/stddev):   10.0045/0.00
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Prime numbers limit: 128000
Initializing worker threads...
Threads started!
CPU speed:
    events per second:    25.87
General statistics:
    total time:                          10.0502s
    total number of events:              260
Latency (ms):
         min:                                   34.74
         avg:                                   38.64
         max:                                  107.80
```

95th percentile: 45.79 sum: 10046.31 Threads fairness: events (avg/stddev): 260.0000/0.00 execution time (avg/stddev): 10.0463/0.00	

Memory Test

Memory Read

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 27917273 (2791132.41 per second) 27262.96 MiB transferred (2725.72 MiB/sec) General statistics: total time: 10.0002s total number of events: 27917273 Latency (ms): min: 0.00 avg: 0.00 max: 11.84 95th percentile: 0.00 sum: 3165.84 Threads fairness: events (avg/stddev): 27917273.0000/0.00 execution time (avg/stddev): 3.1658/0.00	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 70783985 (7077073.63 per second) 69124.99 MiB transferred (6911.20 MiB/sec) General statistics: total time: 10.0001s total number of events: 70783985 Latency (ms): min: 0.00 avg: 0.00 max: 29.02 95th percentile: 0.00 sum: 6438.96 Threads fairness: events (avg/stddev): 35391992.5000/50033.50 execution time (avg/stddev): 3.2195/0.02
WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 75581793 (7556803.76 per second) 73810.34 MiB transferred (7379.69 MiB/sec) General statistics: total time: 10.0001s total number of events: 75581793 Latency (ms): min: 0.00 avg: 0.00 max: 36.03 95th percentile: 0.00 sum: 12831.46 Threads fairness:	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 81283911 (8126998.39 per second) 79378.82 MiB transferred (7936.52 MiB/sec) General statistics: total time: 10.0002s total number of events: 81283911 Latency (ms): min: 0.00 avg: 0.00 max: 48.13 95th percentile: 0.00 sum: 25325.86 Threads fairness:

events (avg/stddev): 18895448.2500/116437.26 execution time (avg/stddev): 3.2079/0.08	events (avg/stddev): 10160488.8750/220364.80 execution time (avg/stddev): 3.1657/0.14
WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: read scope: global Initializing worker threads... Threads started! Total operations: 80981679 (8096502.47 per second) 79083.67 MiB transferred (7906.74 MiB/sec) General statistics: total time: 10.0004s total number of events: 80981679 Latency (ms): min: 0.00 avg: 0.00 max: 88.64 95th percentile: 0.00 sum: 48925.55 Threads fairness: events (avg/stddev): 5061354.9375/49693.71 execution time (avg/stddev): 3.0578/0.21	

Memory Write

sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 28366629 (2836037.71 per second) 27701.79 MiB transferred (2769.57 MiB/sec) General statistics: total time: 10.0002s total number of events: 28366629 Latency (ms): min: 0.00 avg: 0.00 max: 11.35 95th percentile: 0.00 sum: 4458.69 Threads fairness: events (avg/stddev): 28366629.0000/0.00 execution time (avg/stddev): 4.4587/0.00	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Running memory speed test with the following options: block size: 1KiB total size: 102400MiB operation: write scope: global Initializing worker threads... Threads started! Total operations: 36409865 (3640079.18 per second) 35556.51 MiB transferred (3554.76 MiB/sec) General statistics: total time: 10.0001s total number of events: 36409865 Latency (ms): min: 0.00 avg: 0.00 max: 43.91 95th percentile: 0.00 sum: 10525.87 Threads fairness: events (avg/stddev): 18204932.5000/327024.50 execution time (avg/stddev): 5.2629/0.05
WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4	WARNING: --num-threads is deprecated, use --threads instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8

```
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 48663732 (4865210.92 per second)
47523.18 MiB transferred (4751.18 MiB/sec)
General statistics:
    total time:                          10.0002s
    total number of events:              48663732
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   32.04
         95th percentile:                        0.00
         sum:                                21655.80
Threads fairness:
    events (avg/stddev):           12165933.0000/225284.84
    execution time (avg/stddev):   5.4139/0.08
```

```
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 46611305 (4660255.22 per second)
45518.85 MiB transferred (4551.03 MiB/sec)
General statistics:
    total time:                          10.0003s
    total number of events:              46611305
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                   60.21
         95th percentile:                        0.00
         sum:                                42152.63
Threads fairness:
    events (avg/stddev):           5826413.1250/120127.70
    execution time (avg/stddev):   5.2691/0.14
```

```
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Running memory speed test with the following options:
  block size: 1KiB
  total size: 102400MiB
  operation: write
  scope: global
Initializing worker threads...
Threads started!
Total operations: 50303225 (5029283.50 per second)
49124.24 MiB transferred (4911.41 MiB/sec)
General statistics:
    total time:                          10.0005s
    total number of events:              50303225
Latency (ms):
         min:                                    0.00
         avg:                                    0.00
         max:                                  100.02
         95th percentile:                        0.00
         sum:                                85655.11
Threads fairness:
    events (avg/stddev):           3143951.5625/81116.28
    execution time (avg/stddev):   5.3534/0.21
```

File I/O Test

Random Write

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
```

```
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
```

```
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     853.27
    fsyncs/s:                     545.24
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               3.33
General statistics:
    total time:                   60.0009s
    total number of events:       83917
Latency (ms):
         min:                     0.00
         avg:                     0.71
         max:                     177.72
         95th percentile:         4.10
         sum:                     59834.71
Threads fairness:
    events (avg/stddev):          83917.0000/0.00
    execution time (avg/stddev):  59.8347/0.00
```

```
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     1301.35
    fsyncs/s:                     832.15
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               5.08
General statistics:
    total time:                   60.0061s
    total number of events:       128041

Latency (ms):
         min:                     0.00
         avg:                     1.87
         max:                     89.46
         95th percentile:         10.09
         sum:                     239676.14
Threads fairness:
    events (avg/stddev):          32010.2500/1170.66
    execution time (avg/stddev):  59.9190/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     1804.66
    fsyncs/s:                     1154.18
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               7.05
General statistics:
    total time:                   60.0068s
    total number of events:       177564
Latency (ms):
         min:                     0.00
         avg:                     2.70
         max:                     152.33
         95th percentile:         12.30
         sum:                     479658.99
Threads fairness:
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      0.00
    writes/s:                     2591.25
    fsyncs/s:                     1657.53
Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               10.12
General statistics:
    total time:                   60.0082s
    total number of events:       254968
Latency (ms):
         min:                     0.00
         avg:                     3.76
         max:                     65.39
         95th percentile:         16.12
         sum:                     959529.68
Threads fairness:
```

events (avg/stddev): 22195.5000/612.83 execution time (avg/stddev): 59.9574/0.00	events (avg/stddev): 15935.5000/620.05 execution time (avg/stddev): 59.9706/0.00
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random write test Initializing worker threads... Threads started! File operations: reads/s: 0.00 writes/s: 2591.25 fsyncs/s: 1657.53 Throughput: read, MiB/s: 0.00 written, MiB/s: 10.12 General statistics: total time: 60.0082s total number of events: 254968 Latency (ms): min: 0.00 avg: 3.76 max: 65.39 95th percentile: 16.12 sum: 959529.68 Threads fairness: events (avg/stddev): 15935.5000/620.05 execution time (avg/stddev): 59.9706/0.00	

Random Read

WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 1 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 3383.82 writes/s: 0.00	WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 2 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 9320.53 writes/s: 0.00

fsyncs/s: 0.00 Throughput: read, MiB/s: 13.22 written, MiB/s: 0.00 General statistics: total time: 60.0010s total number of events: 203040 Latency (ms): min: 0.00 avg: 0.29 max: 90.32 95th percentile: 0.69 sum: 59635.00 Threads fairness: events (avg/stddev): 203040.0000/0.00 execution time (avg/stddev): 59.6350/0.00	fsyncs/s: 0.00 Throughput: read, MiB/s: 36.41 written, MiB/s: 0.00 General statistics: total time: 60.0328s total number of events: 559580 Latency (ms): min: 0.00 avg: 0.21 max: 210.04 95th percentile: 0.39 sum: 119062.89 Threads fairness: events (avg/stddev): 279790.0000/833.00 execution time (avg/stddev): 59.5314/0.00
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 4 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 11543.37 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 45.09 written, MiB/s: 0.00 General statistics: total time: 60.0298s total number of events: 692972 Latency (ms): min: 0.00 avg: 0.34 max: 938.60 95th percentile: 0.75 sum: 238975.98 Threads fairness: events (avg/stddev): 173243.0000/331.86 execution time (avg/stddev): 59.7440/0.01	WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 8 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size Block size 4KiB Number of IO requests: 0 Read/Write ratio for combined random IO test: 1.50 Periodic FSYNC enabled, calling fsync() each 100 requests. Calling fsync() at the end of test, Enabled. Using synchronous I/O mode Doing random read test Initializing worker threads... Threads started! File operations: reads/s: 24722.64 writes/s: 0.00 fsyncs/s: 0.00 Throughput: read, MiB/s: 96.57 written, MiB/s: 0.00 General statistics: total time: 60.0098s total number of events: 1483672 Latency (ms): min: 0.00 avg: 0.32 max: 1019.55 95th percentile: 0.81 sum: 477953.14 Threads fairness: events (avg/stddev): 185459.0000/2338.53 execution time (avg/stddev): 59.7441/0.01
WARNING: --num-threads is deprecated, use --threads instead WARNING: --max-time is deprecated, use --time instead sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3) Running the test with following options: Number of threads: 16 Initializing random number generator from current time Extra file open flags: 0 64 files, 64MiB each 4GiB total file size	

```
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random read test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      35872.35
    writes/s:                     0.00
    fsyncs/s:                     0.00
Throughput:
    read, MiB/s:                  140.13
    written, MiB/s:               0.00
General statistics:
    total time:                   60.0106s
    total number of events:       2152944
Latency (ms):
         min:                     0.00
         avg:                     0.44
         max:                     1057.42
         95th percentile:         1.25
         sum:                     957151.73
Threads fairness:
    events (avg/stddev):          134559.0000/2044.61
    execution time (avg/stddev):  59.8220/0.01
```

Random Read/Write

```
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      457.96
    writes/s:                     305.31
    fsyncs/s:                     487.81
Throughput:
    read, MiB/s:                  1.79
    written, MiB/s:               1.19
General statistics:
    total time:                   60.0031s
    total number of events:       75071
Latency (ms):
         min:                     0.00
         avg:                     0.80
         max:                     274.65
         95th percentile:         3.89
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
    reads/s:                      652.96
    writes/s:                     435.31
    fsyncs/s:                     696.38
Throughput:
    read, MiB/s:                  2.55
    written, MiB/s:               1.70
General statistics:
    total time:                   60.0020s
    total number of events:       107085
Latency (ms):
         min:                     0.00
         avg:                     1.12
         max:                     268.80
```

```
      sum:                    59821.59
Threads fairness:
   events (avg/stddev):          75071.0000/0.00
   execution time (avg/stddev):  59.8216/0.00
```

```
      95th percentile:            5.88
      sum:                   119677.36
Threads fairness:
   events (avg/stddev):          53542.5000/449.50
   execution time (avg/stddev):  59.8387/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                  856.79
   writes/s:                 571.17
   fsyncs/s:                 912.93
Throughput:
   read, MiB/s:              3.35
   written, MiB/s:           2.23
General statistics:
   total time:                  60.0088s
   total number of events:          140490
Latency (ms):
      min:                    0.00
      avg:                    1.71
      max:                   92.44
      95th percentile:        8.43
      sum:                239613.08
Threads fairness:
   events (avg/stddev):          35122.5000/151.57
   execution time (avg/stddev):  59.9033/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                 1218.83
   writes/s:                812.47
   fsyncs/s:                1299.15
Throughput:
   read, MiB/s:              4.76
   written, MiB/s:           3.17
General statistics:
   total time:                  60.0090s
   total number of events:          199863
Latency (ms):
      min:                    0.00
      avg:                    2.40
      max:                  103.96
      95th percentile:       10.46
      sum:                479601.93
Threads fairness:
   events (avg/stddev):          24982.8750/135.01
   execution time (avg/stddev):  59.9502/0.00
```

```
WARNING: --num-threads is deprecated, use --threads instead
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Extra file open flags: 0
64 files, 64MiB each
4GiB total file size
Block size 4KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
Threads started!
File operations:
   reads/s:                 1475.90
```

writes/s: 983.80 fsyncs/s: 1574.03 Throughput: read, MiB/s: 5.77 written, MiB/s: 3.84 General statistics: total time: 60.0056s total number of events: 242053 Latency (ms): min: 0.00 avg: 3.96 max: 440.36 95th percentile: 16.41 sum: 959448.12 Threads fairness: events (avg/stddev): 15128.3125/201.74 execution time (avg/stddev): 59.9655/0.00	

References

Environment setup
https://www.settlersoman.com/how-to-installrun-hyper-v-host-as-a-vmnested-on-vsphere-5-or-6/
https://www.altaro.com/vmware/deploying-nested-esxi-is-easier-than-ever-before/

Bench Marking and Sysbench
https://www.howtoforge.com/how-to-benchmark-your-system-cpu-file-io-mysql-with-sysbench
https://wiki.mikejung.biz/Sysbench
https://forum.cockroachlabs.com/t/sysbench-oltp-performance-test/1150
http://www.mellanox.com/blog/2016/02/set-vmware-vmotion-into-fast-motion-over-high-speed-interconnect/
https://wiki.gentoo.org/wiki/Sysbench
Kopytov, A., SysBench Manual. http://sysbench.sourceforge.net/docs/, Accessed on April 12, 2012.

Research
http://blog.siphos.be/2013/04/comparing-performance-with-sysbench-part-2/
https://www.golinuxhub.com/2016/09/how-to-monitor-network-bandwidth-in.html
https://www.programering.com/a/MjN3IDNwATc.html
https://en.wikipedia.org/wiki/Virtualization
https://en.wikipedia.org/wiki/Virtual_machine
https://en.wikipedia.org/wiki/Hypervisor
https://www.vmware.com/topics/glossary/content/hypervisor
https://www.vmware.com/pdf/virtualization.pdf
https://www.techopedia.com/definition/688/server-virtualization
https://redswitches.com/blog/different-types-virtualization-cloud-computing-explained/
https://www.vmware.com/products/esxi-and-esx.html
https://www.vmware.com/pdf/vsphere4/r40/vsp_40_intro_vs.pdf
https://searchvmware.techtarget.com/definition/VMware
https://www.vmware.com/pdf/vsphere4/r40/vsp_40_intro_vs.pdf
http://www.vmwarearena.com/difference-between-vmware-esx-and-esxi/
https://www.zdnet.com/article/desktop-virtualization-vs-virtual-desktop-infrastructure/
https://blogs.vmware.com/vsphere/2009/06/esxi-vs-esx-a-comparison-of-features.html
https://www.ricohidc.com/kb/an-overview-of-vmware-esx-server-architecture/
https://www.vmware.com/products/workstation-pro.html
https://en.wikipedia.org/wiki/VMware_Workstation
http://www.apmdigest.com/sites/default/files/images/VMvSphereHyperV_Whitepaper.pdf
https://www.extremetech.com/computing/72186-virtual-machines-vmware-part-i/6
https://en.wikipedia.org/wiki/Hyper-V
https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/
https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview
https://www.altaro.com/hyper-v/what-is-hyper-v/

https://www.znetlive.com/blog/server-virtualization-software-comparison-microsoft-hyper-v-vs-vmware-vsphere-vs-citrix-xenserver-vs-kvm/
https://www.quora.com/What-is-the-difference-between-VMware-and-Hyper-V
https://www.atlantech.net/blog/hyper-v-vs.-vmware-which-is-best
https://4sysops.com/archives/microsoft-hyper-v-dynamic-memory-vs-vmware-memory-overcommit/
https://searchservervirtualization.techtarget.com/tip/Hyper-V-dynamic-memory-allocation-vs-VMware-memory-overcommit
https://blogs.vmware.com/virtualreality/2008/06/a-look-at-some.html - do not inlcude this
https://www.quora.com/What-is-the-difference-between-VMware-and-Hyper-V
https://www.loomsystems.com/blog/vmware-vs.-microsoft-hyper-v-is-vmware-still-far-ahead
https://software.intel.com/en-us/articles/the-advantages-of-using-virtualization-technology-in-the-enterprise
https://www.computerweekly.com/tip/Which-is-cheaper-Hyper-V-or-VMware