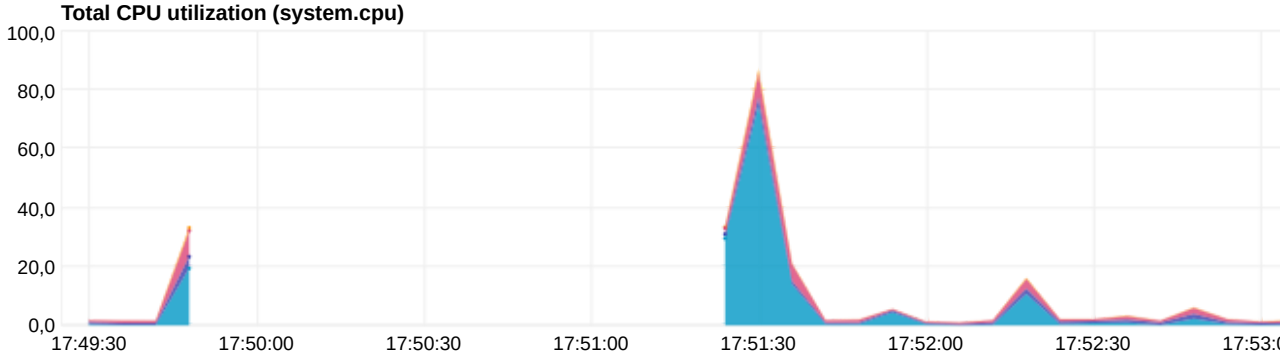
System Overview

Overview of the key system metrics.

cpu

Total CPU utilization (all cores). 100% here means there is no CPU idle time at all. You can get per core usage at the CPUs section and per application usage at the Applications Monitoring section.
Keep an eye on **iowait** (2,5%). If it is constantly high, your disks are a bottleneck and they slow your system down.
An important metric worth monitoring, is **softirq** (0,00%). A constantly high percentage of softirq may indicate network driver issues.

Total CPU utilization (system.cpu)

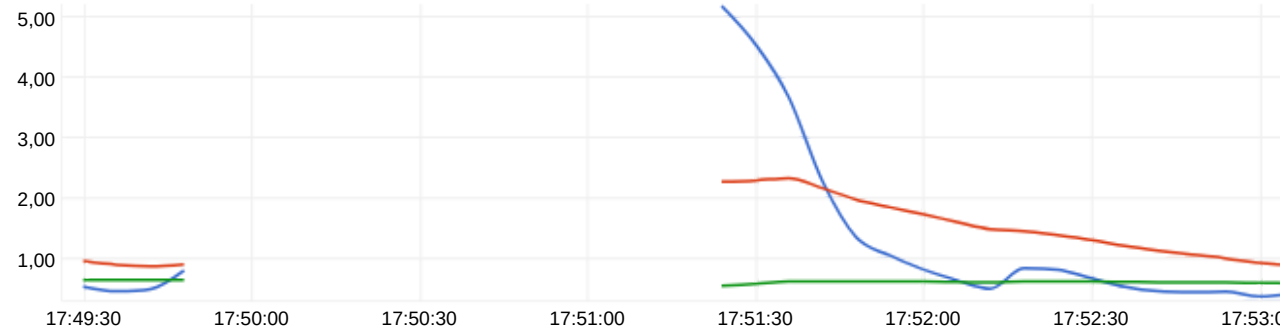


percentage сб, 24 апр. 2021 г. | 17:54:24

softirq 0,0 user 0,8 system 0,8 iowait 0,5

Pressure Stall Information (https://www.kernel.org/doc/html/latest/accounting/psi.html) identifies and quantifies the disruptions caused by resource contentions. The "some" line indicates the share of time in which at least **some** tasks are stalled on CPU. The ratios (in %) are tracked as recent trends over 10-, 60-, and 300-second windows.

CPU Pressure (system.cpu_pressure)



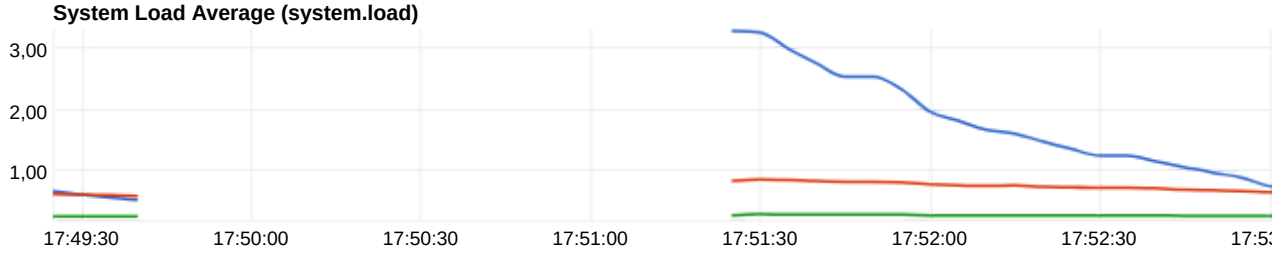
percentage сб, 24 апр. 2021 г. | 17:54:24

some 10 0,69 some 60 0,67 some 300 0,60

load

Current system load, i.e. the number of processes using CPU or waiting for system resources (usually CPU and disk). The 3 metrics refer to 1, 5 and 15 minute averages. The system calculates this once every 5 seconds. For more information check this wikipedia article (https://en.wikipedia.org/wiki/Load_(computing)).

System Load Average (system.load)



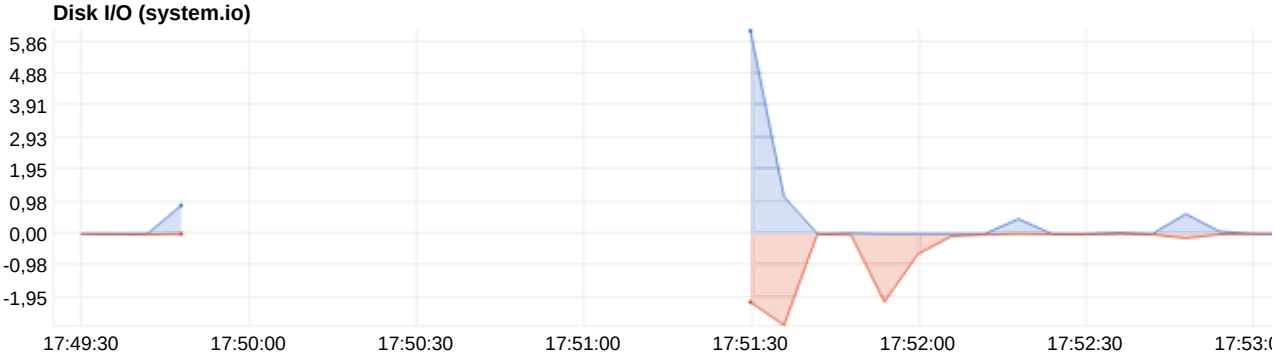
load сб, 24 апр. 2021 г. | 17:54:20

load1 0,20 load5 0,50 load15 0,24

disk

Total Disk I/O, for all physical disks. You can get detailed information about each disk at the Disks section and per application Disk usage at the Applications Monitoring section. Physical are all the disks that are listed in `/sys/block`, but do not exist in `/sys/devices/virtual/block`.

Disk I/O (system.io)

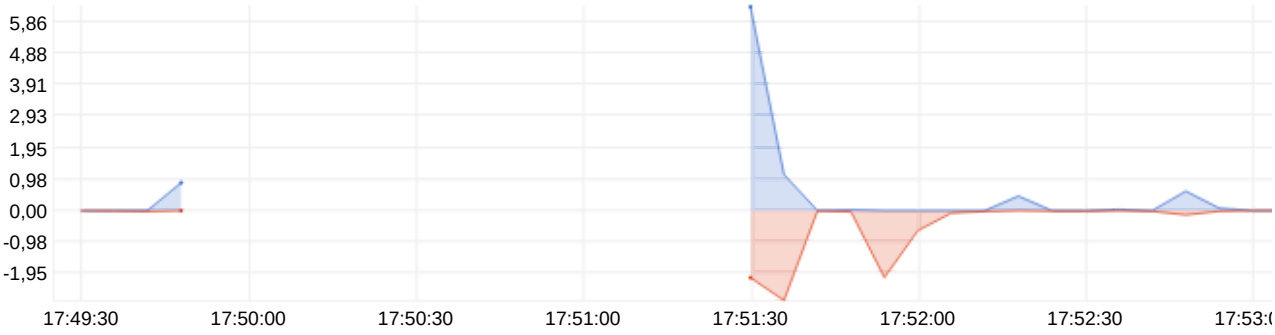


MiB/s сб, 24 апр. 2021 г. | 17:54:24

in 0,00 out -0,01

Memory paged from/to disk. This is usually the total disk I/O of the system.

Memory Paged from/to disk (system.pgpgio)

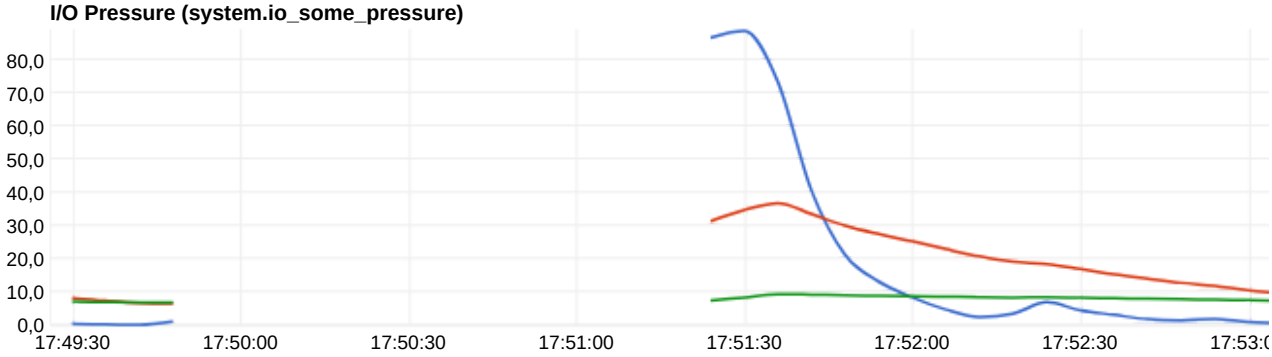


MiB/s сб, 24 апр. 2021 г. | 17:54:24

in 0,00 out -0,01

Pressure Stall Information (https://www.kernel.org/doc/html/latest/accounting/psi.html) identifies and quantifies the disruptions caused by resource contentions. The "some" line indicates the share of time in which at least **some** tasks are stalled on I/O. The "full" line indicates the share of time in which **all non-idle** tasks are stalled on I/O simultaneously. In this state actual CPU cycles are going to waste, and a workload that spends extended time in this state is considered to be thrashing. The ratios (in %) are tracked as recent trends over 10-, 60-, and 300-second windows.

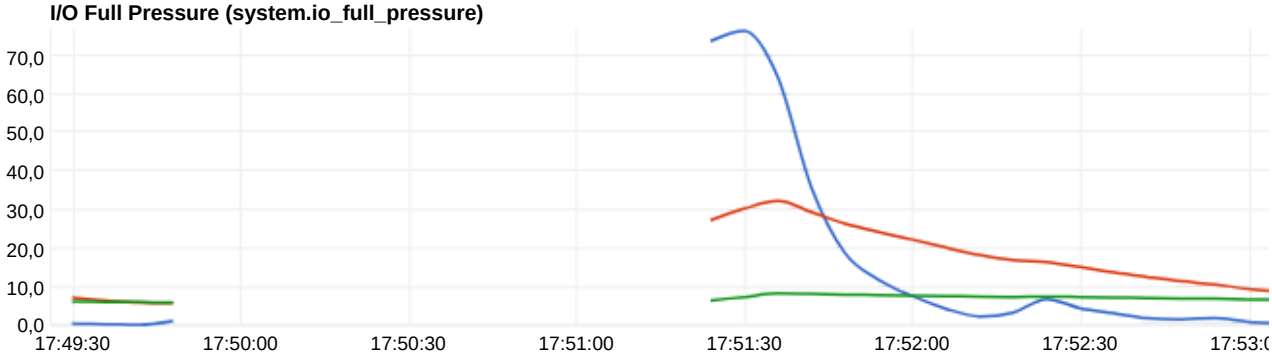
I/O Pressure (system.io_some_pressure)



percentage сб, 24 апр. 2021 г. | 17:54:24

some 10 0,7 some 60 3,0 some 300 5,9

I/O Full Pressure (system.io_full_pressure)



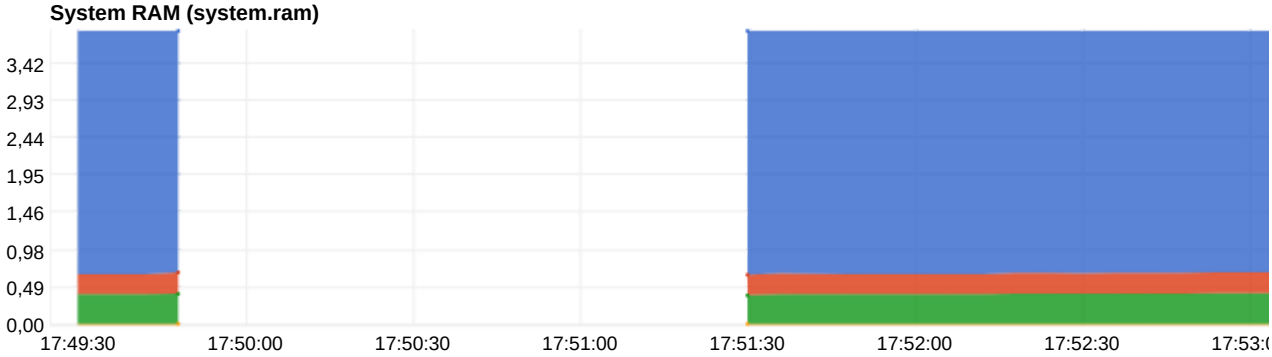
percentage сб, 24 апр. 2021 г. | 17:54:24

full 10 0,7 full 60 2,8 full 300 5,3

ram

System Random Access Memory (i.e. physical memory) usage.

System RAM (system.ram)

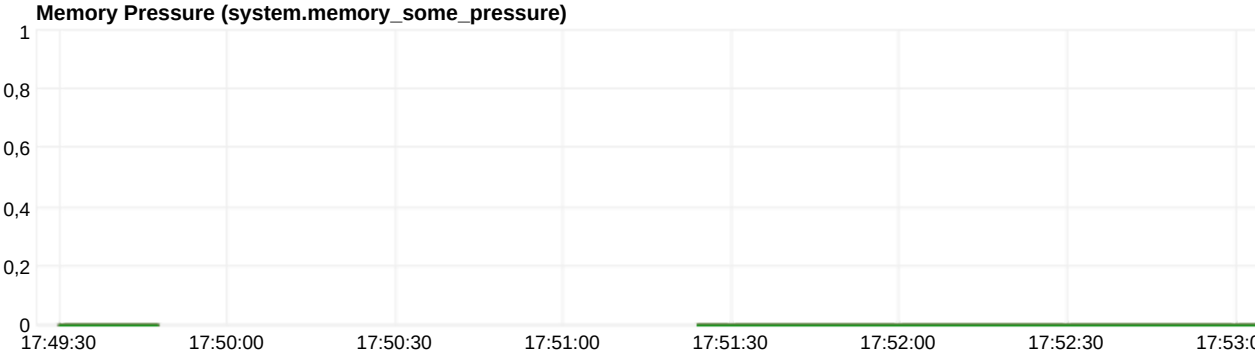


GiB сб, 24 апр. 2021 г. | 17:54:24

free 3,15 used 0,28 cached 0,40 buffers 0,02

Pressure Stall Information (https://www.kernel.org/doc/html/latest/accounting/psi.html) identifies and quantifies the disruptions caused by resource contentions. The "some" line indicates the share of time in which at least **some** tasks are stalled on memory. The "full" line indicates the share of time in which **all non-idle** tasks are stalled on memory simultaneously. In this state actual CPU cycles are going to waste, and a workload that spends extended time in this state is considered to be thrashing. The ratios (in %) are tracked as recent trends over 10-, 60-, and 300-second windows.

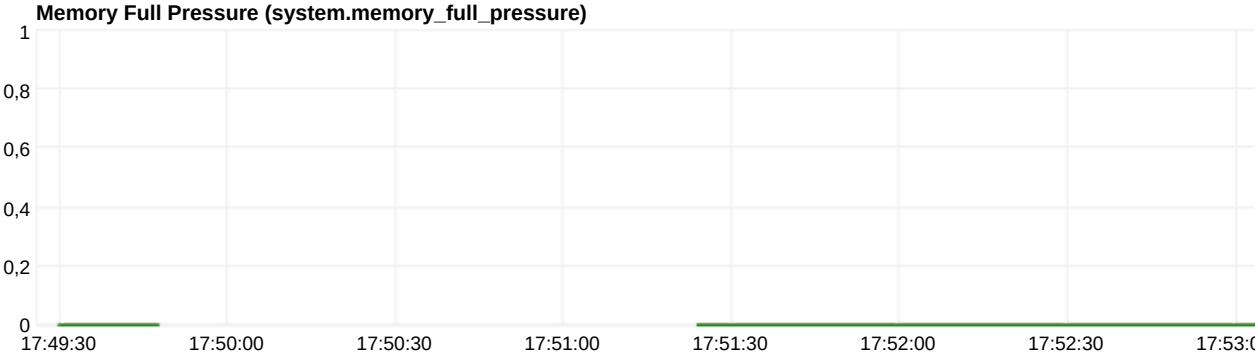
Memory Pressure (system.memory_some_pressure)



percentage сб, 24 апр. 2021 г. | 17:54:24

some 10 0 some 60 0 some 300 0

Memory Full Pressure (system.memory_full_pressure)



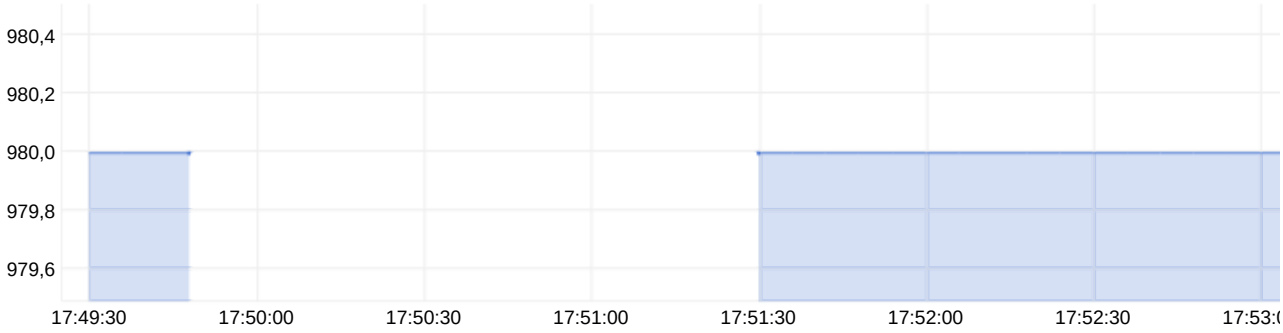
percentage сб, 24 апр. 2021 г. | 17:54:24

full 10 0 full 60 0 full 300 0

swap

System swap memory usage. Swap space is used when the amount of physical memory (RAM) is full. When the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space (usually a disk, a disk partition or a file).

System Swap (system.swap)



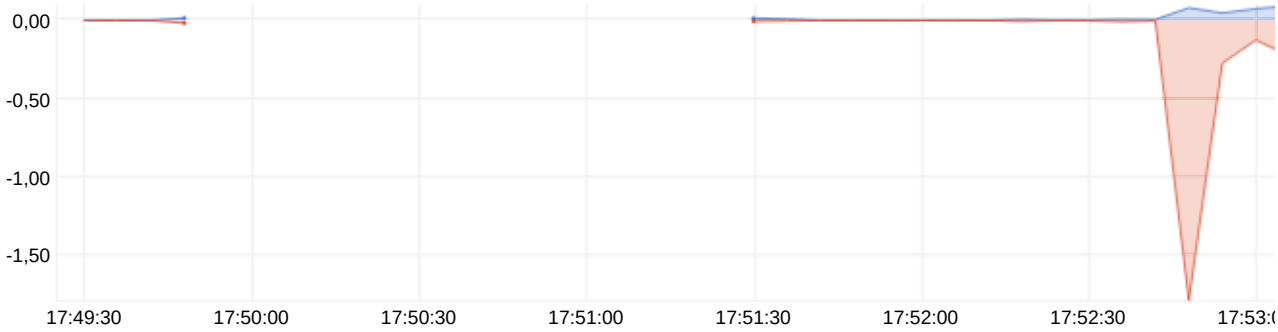
MiB сб, 24 апр. 2021 г. | 17:54:24

free 980,0

network

Total bandwidth of all physical network interfaces. This does not include `lo`, VPNs, network bridges, IFB devices, bond interfaces, etc. Only the bandwidth of physical network interfaces is aggregated. Physical are all the network interfaces that are listed in `/proc/net/dev`, but do not exist in `/sys/devices/virtual/net`.

Physical Network Interfaces Aggregated Bandwidth (system.net)

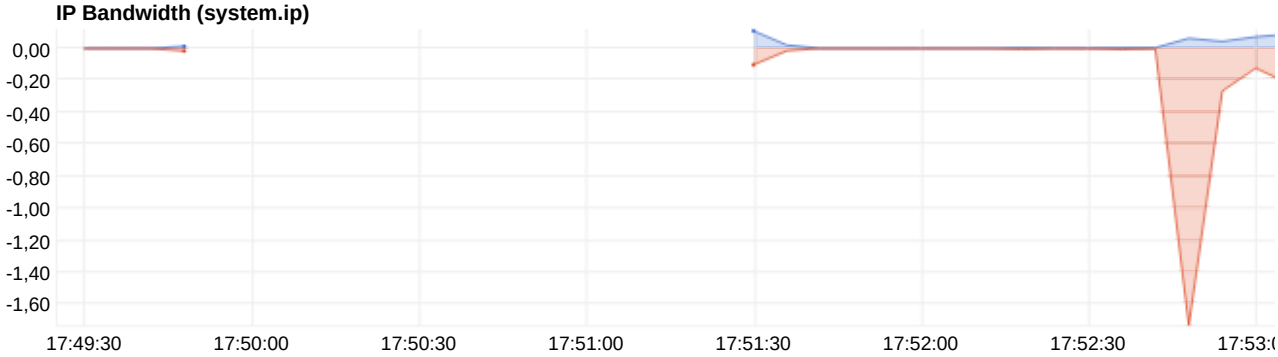


megabits/s сб, 24 апр. 2021 г. | 17:54:24

received 0,09 sent -0,15

Total IP traffic in the system.

IP Bandwidth (system.ip)

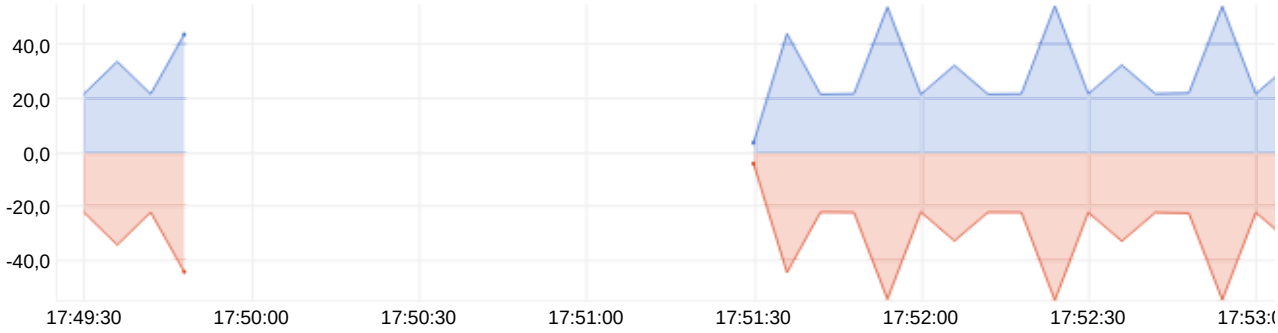


megabits/s сб, 24 апр. 2021 г. | 17:54:24

received 0,08 sent -0,15

Total IPv6 Traffic.

IPv6 Bandwidth (system.ipv6)



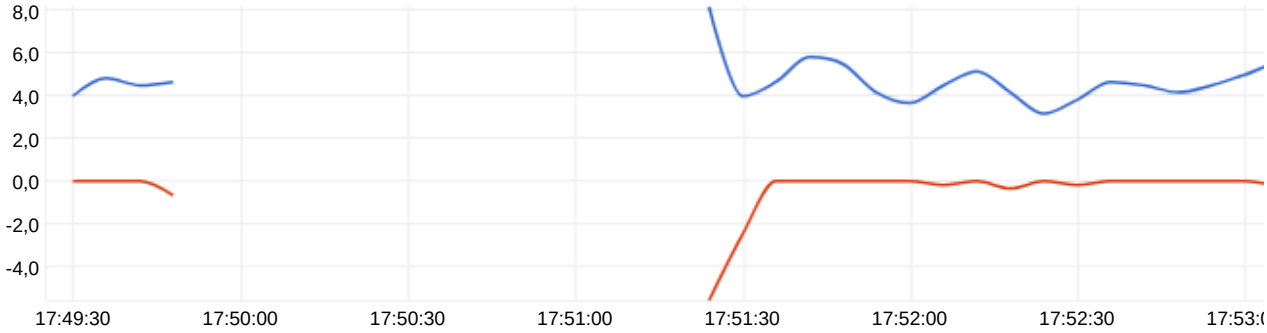
kilobits/s сб, 24 апр. 2021 г. | 17:54:24

received 54,6 sent -54,6

processes

System processes. **Running** are the processes in the CPU. **Blocked** are processes that are willing to enter the CPU, but they cannot, e.g. because they wait for disk activity.

System Processes (system.processes)

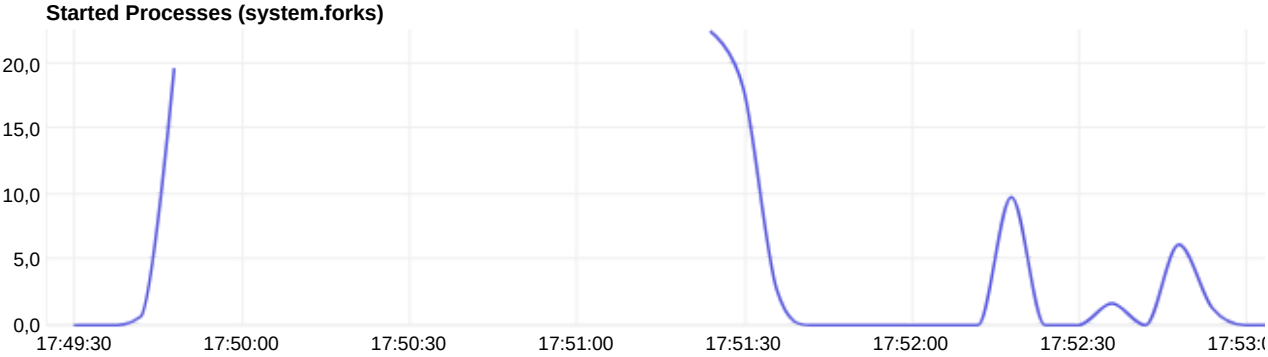


processes сб, 24 апр. 2021 г. | 17:54:24

running 4,5 blocked 0,0

Number of new processes created.

Started Processes (system.forks)

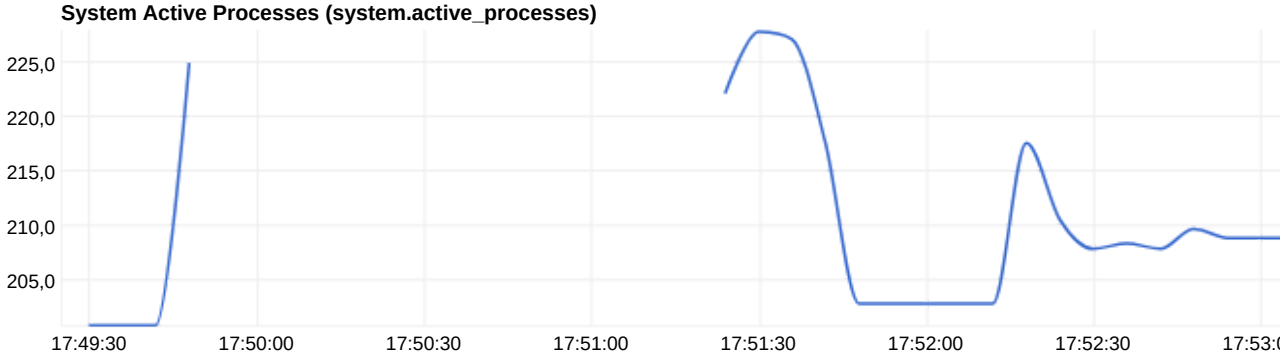


processes/s сб, 24 апр. 2021 г. | 17:54:24

started 0,0

All system processes.

System Active Processes (system.active_processes)



processes сб, 24 апр. 2021 г. | 17:54:24

active 210,0

Context Switches (https://en.wikipedia.org/wiki/Context_switch), is the switching of the CPU from one process, task or thread to another. If there are many processes or threads willing to execute and very few CPU cores available to handle them, the system is making more context switching to balance the CPU resources among them. The whole process is computationally intensive. The more the context switches, the slower the system gets.

CPU Context Switches (system.ctxt)

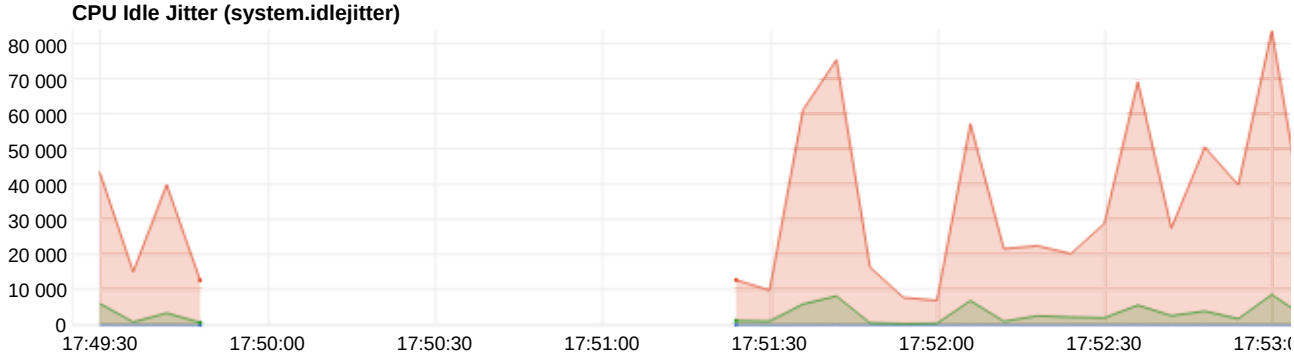
context switches/s сб, 24 апр. 2021 г. | 17:54:24

switches 1 274

idlejitter

Idle jitter is calculated by netdata. A thread is spawned that requests to sleep for a few microseconds. When the system wakes it up, it measures how many microseconds have passed. The difference between the requested and the actual duration of the sleep, is the **idle jitter**. This number is useful in real-time environments, where CPU jitter can affect the quality of the service (like VoIP media gateways).

CPU Idle Jitter (system.idlejitter)

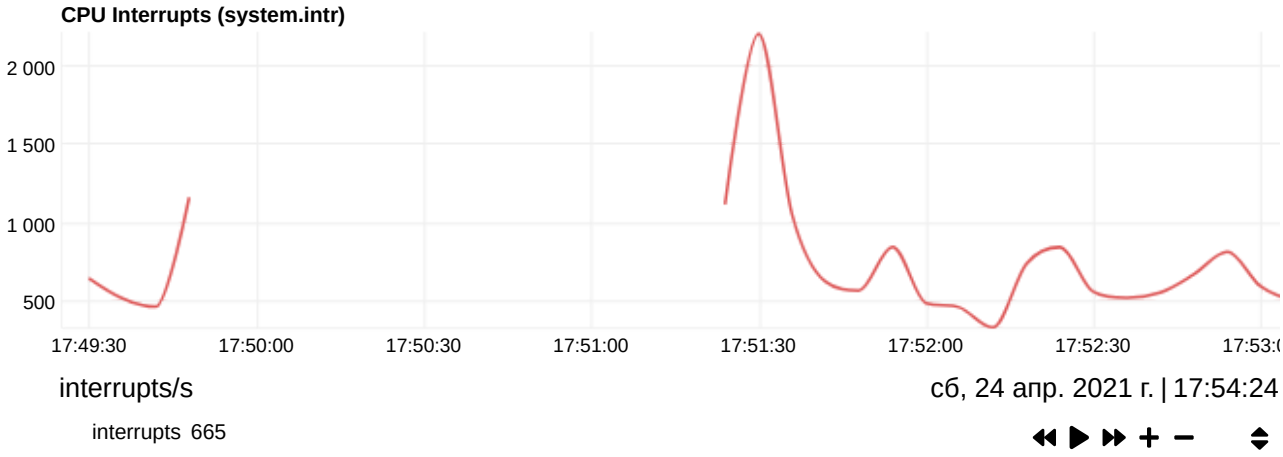


microseconds lost/s сб, 24 апр. 2021 г. | 17:54:24

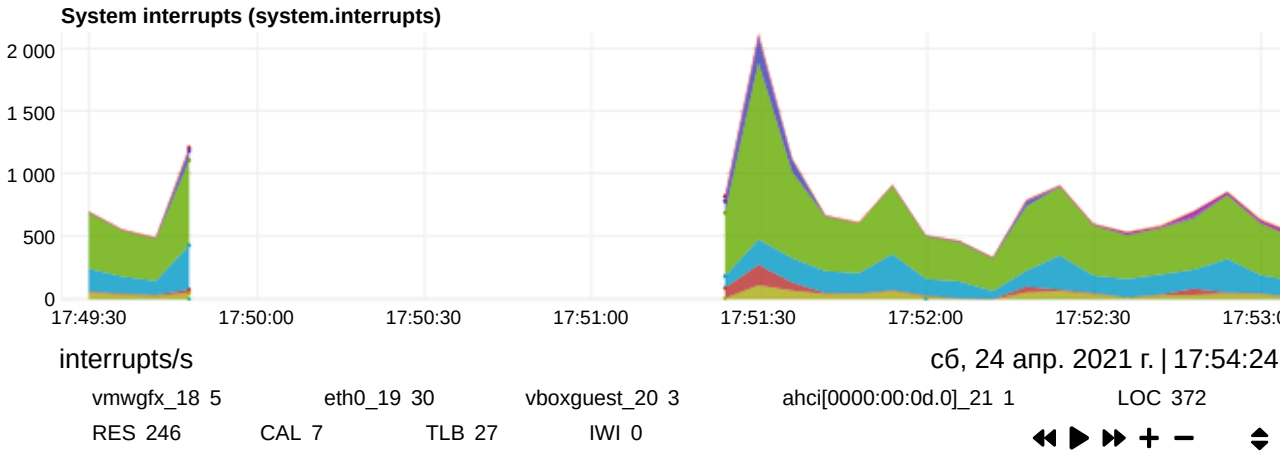
min 64 max 8 903 average 581

interrupts

Total number of CPU interrupts. Check `system.interrupts` that gives more detail about each interrupt and also the CPUs section where interrupts are analyzed per CPU core.

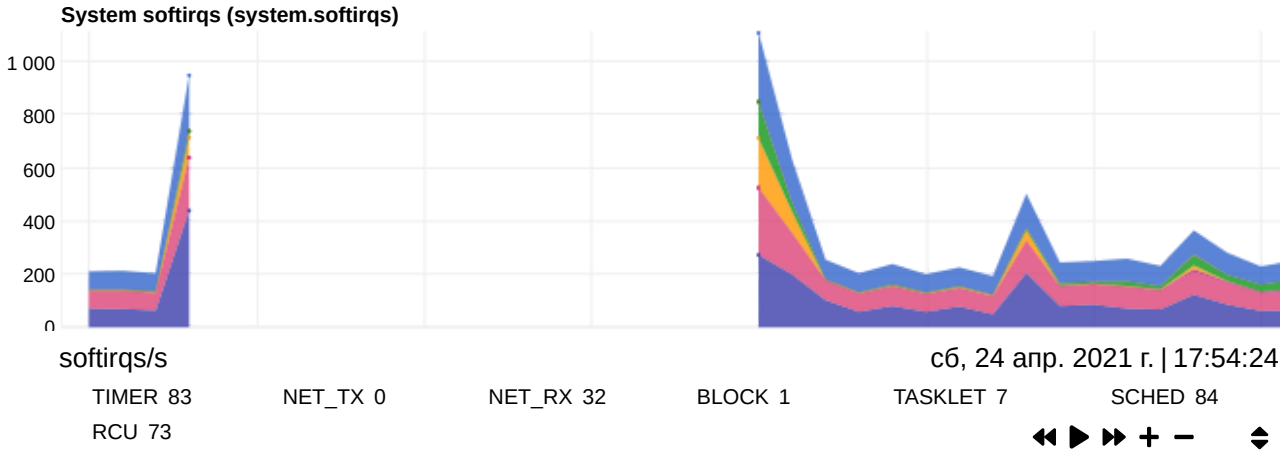


CPU interrupts in detail. At the CPUs section, interrupts are analyzed per CPU core.



softirqs

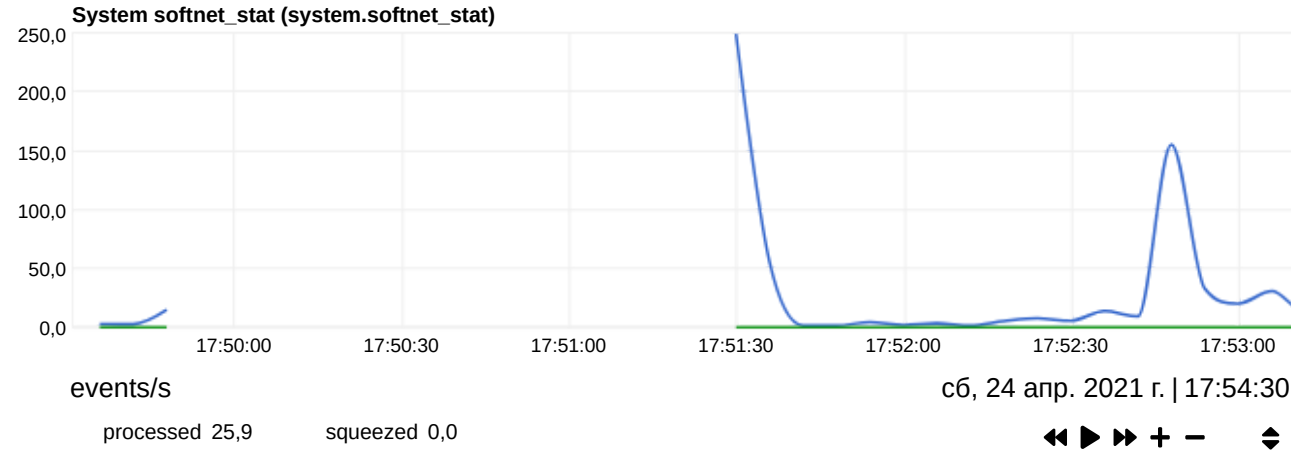
CPU softirqs in detail. At the CPUs section, softirqs are analyzed per CPU core.



softnet

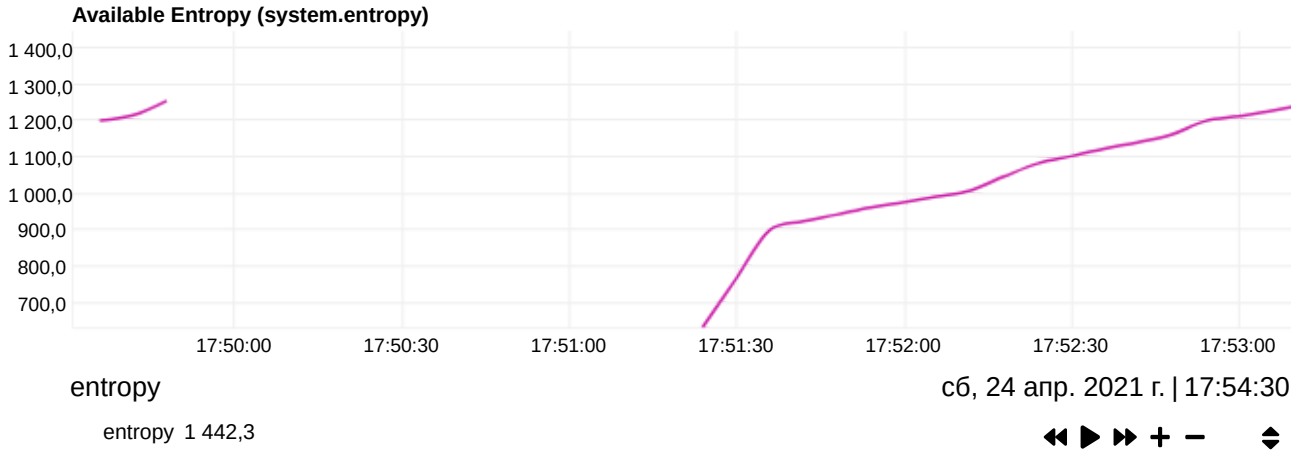
Statistics for CPUs SoftIRQs related to network receive work. Break down per CPU core can be found at CPU / softnet statistics. **processed** states the number of packets processed, **dropped** is the

number packets dropped because the network device backlog was full (to fix them on Linux use `sysctl` to increase `net.core.netdev_max_backlog`), **squeezed** is the number of packets dropped because the network device budget ran out (to fix them on Linux use `sysctl` to increase `net.core.netdev_budget` and/or `net.core.netdev_budget_usecs`). More information about identifying and troubleshooting network driver related issues can be found at Red Hat Enterprise Linux Network Performance Tuning Guide (https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf).



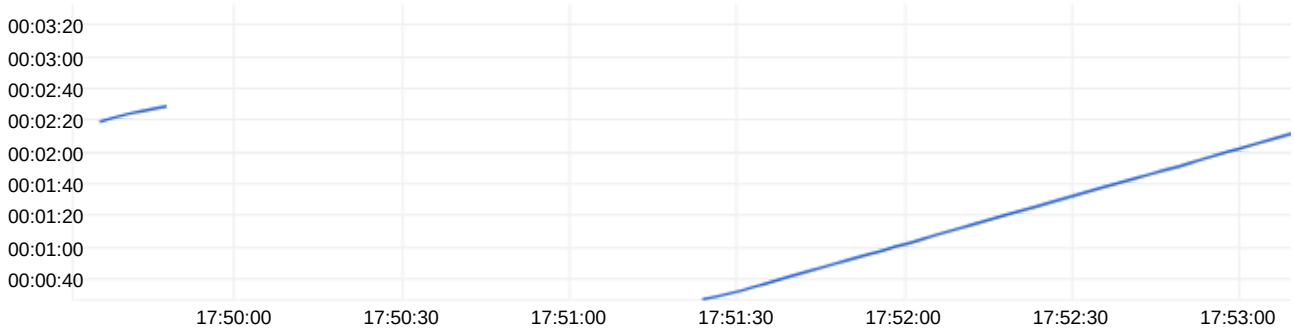
entropy

Entropy (https://en.wikipedia.org/wiki/Entropy_(computing)), is a pool of random numbers (/dev/random (https://en.wikipedia.org/wiki//dev/random)) that is mainly used in cryptography. If the pool of entropy gets empty, processes requiring random numbers may run a lot slower (it depends on the interface each program uses), waiting for the pool to be replenished. Ideally a system with high entropy demands should have a hardware device for that purpose (TPM is one such device). There are also several software-only options you may install, like `haveged`, although these are generally useful only in servers.



uptime

System Uptime (system.uptime)



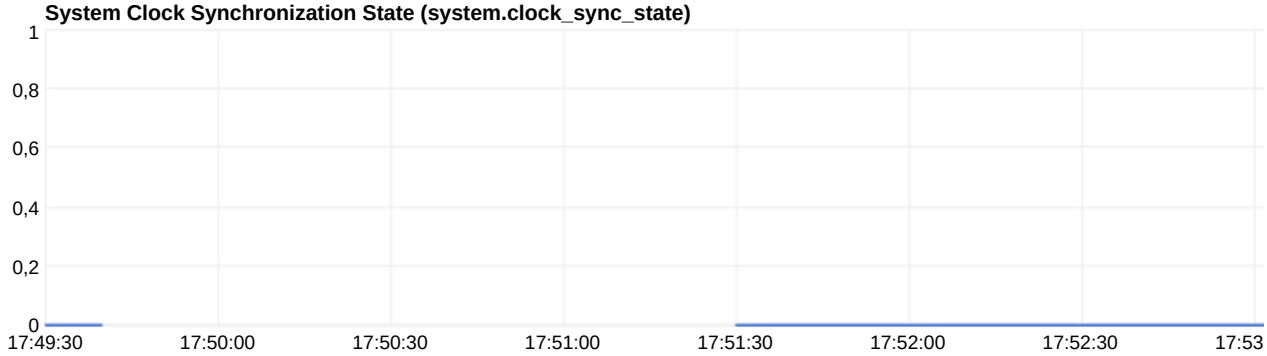
time сб, 24 апр. 2021 г. | 17:54:30

uptime 00:03:33

clock synchronization

State map: 0 - not synchronized, 1 - synchronized

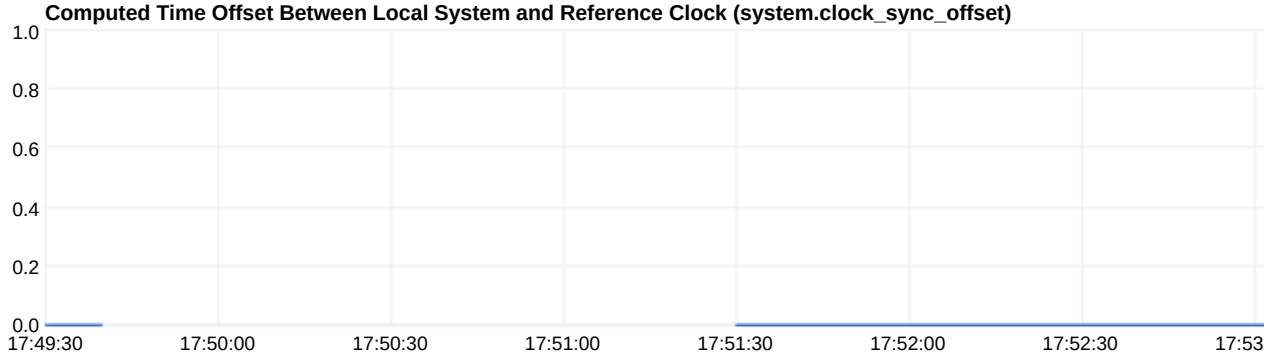
System Clock Synchronization State (system.clock_sync_state)



state сб, 24 апр. 2021 г. | 17:54:20

state 0

Computed Time Offset Between Local System and Reference Clock (system.clock_sync_offset)

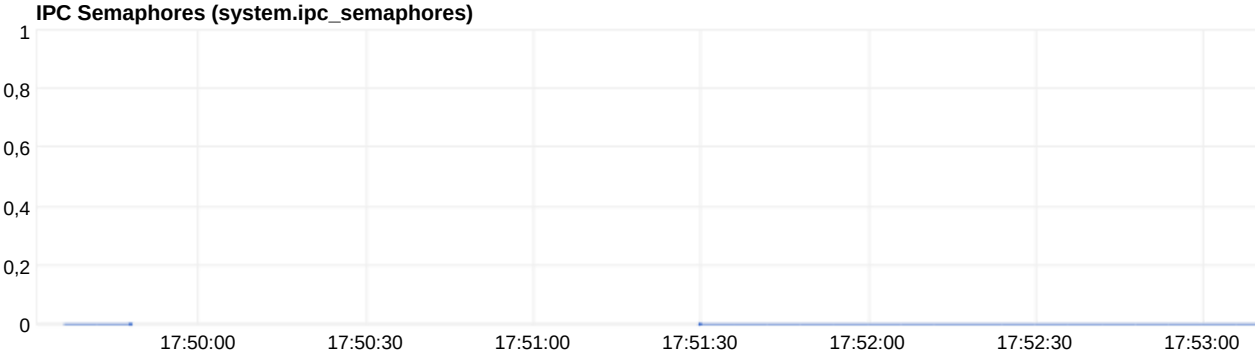


milliseconds сб, 24 апр. 2021 г. | 17:54:20

offset 0.0

ipc semaphores

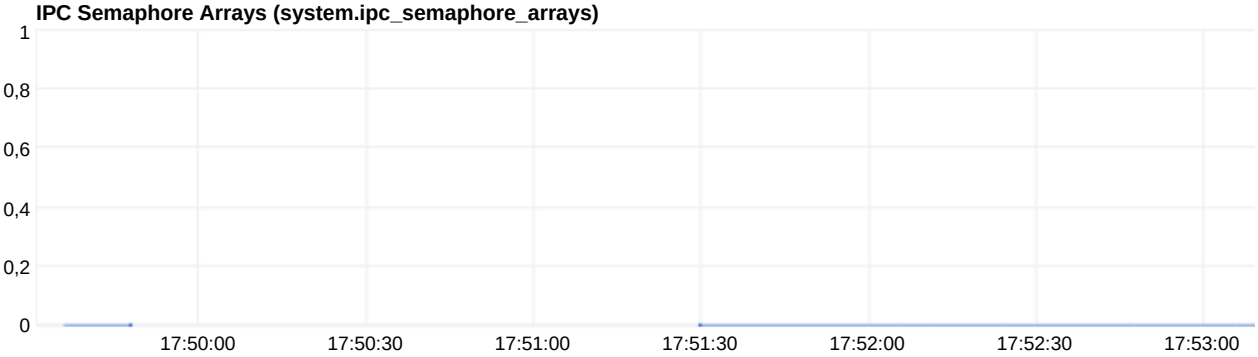
IPC Semaphores (system.ipc_semaphores)



semaphores сб, 24 апр. 2021 г. | 17:54:30

semaphores 0

IPC Semaphore Arrays (system.ipc_semaphore_arrays)

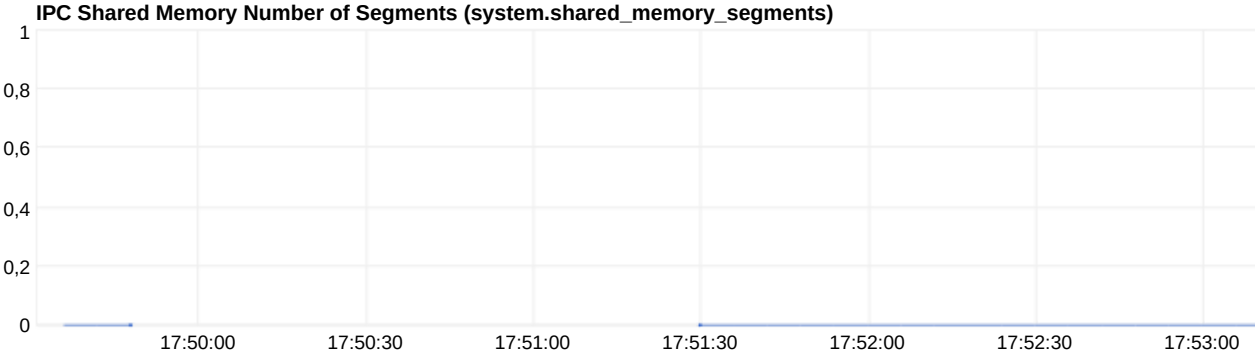


arrays сб, 24 апр. 2021 г. | 17:54:30

arrays 0

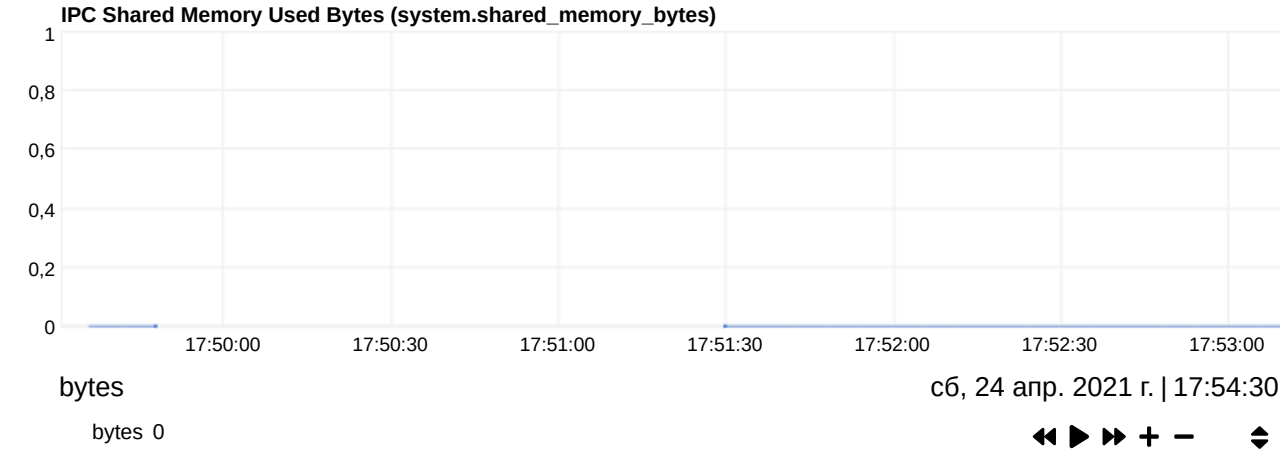
ipc shared memory

IPC Shared Memory Number of Segments (system.shared_memory_segments)



segments сб, 24 апр. 2021 г. | 17:54:30

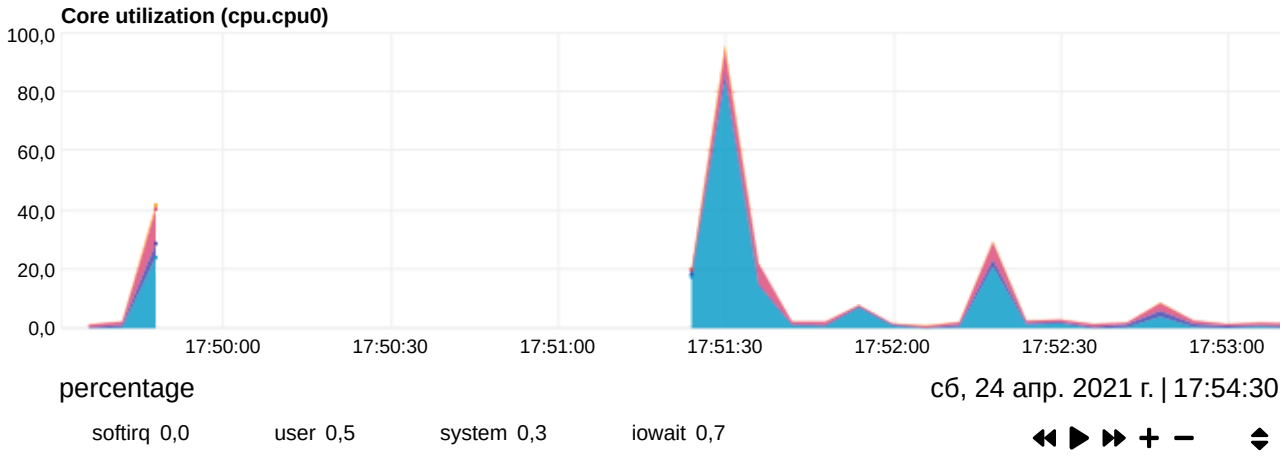
segments 0

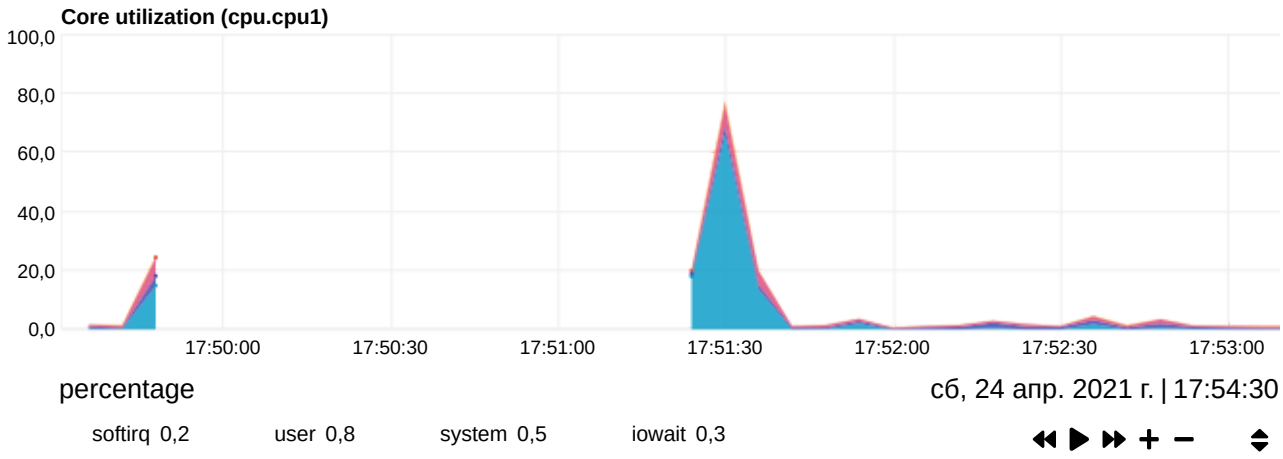


CPUs

Detailed information for each CPU of the system. A summary of the system for all CPUs can be found at the System Overview section.

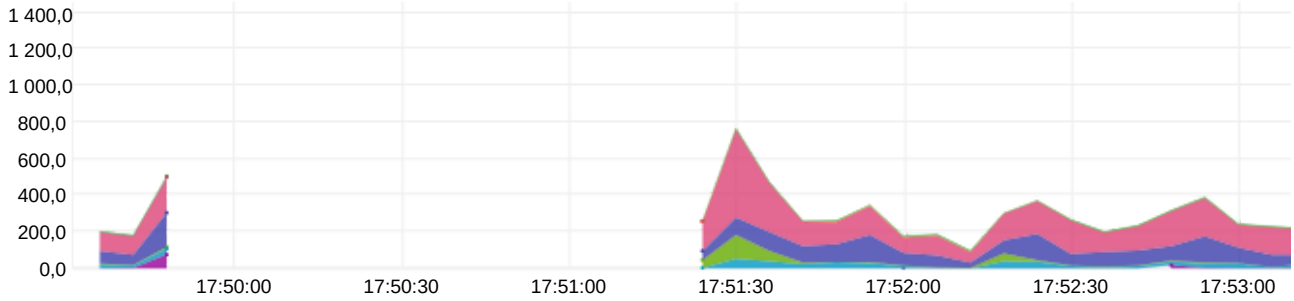
utilization





interrupts

CPU0 Interrupts (cpu.cpu0_interrupts)

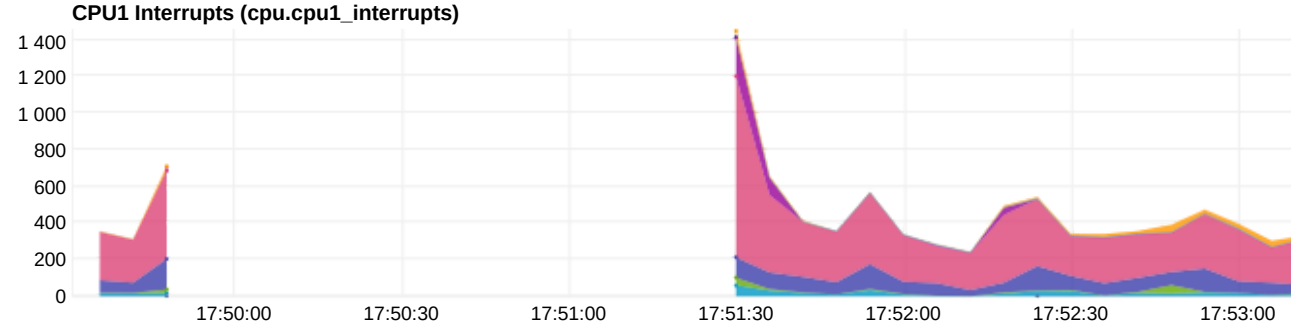


interrupts/s сб, 24 апр. 2021 г. | 17:54:30

vmwgfx_18 5,0 eth0_19 0,0 LOC 111,5 RES 94,3 CAL 2,2 TLB 25,3

IWI 0,0 ahci[0000:00:0d.0]_21 1,0

CPU1 Interrupts (cpu.cpu1_interrupts)



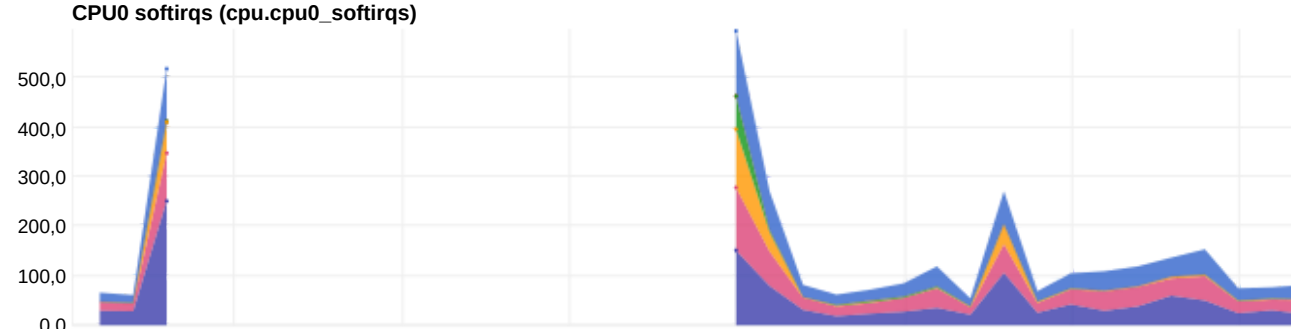
interrupts/s сб, 24 апр. 2021 г. | 17:54:30

eth0_19 25 vboxguest_20 3 ahci[0000:00:0d.0]_21 0 LOC 259 RES 47

CAL 1 TLB 11 IWI 0

softirqs

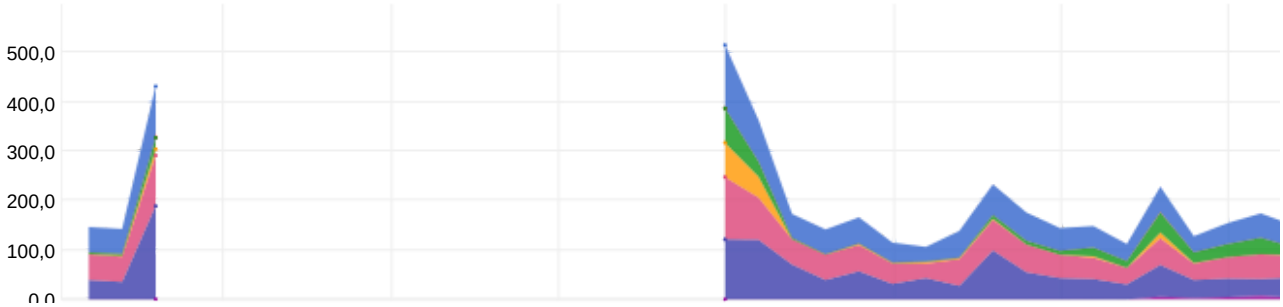
CPU0 softirqs (cpu.cpu0_softirqs)



softirqs/s сб, 24 апр. 2021 г. | 17:54:30

TIMER 23,5 NET_TX 0,2 NET_RX 0,7 BLOCK 0,7 SCHED 24,2 RCU 31,5

CPU1 softirqs (cpu.cpu1_softirqs)



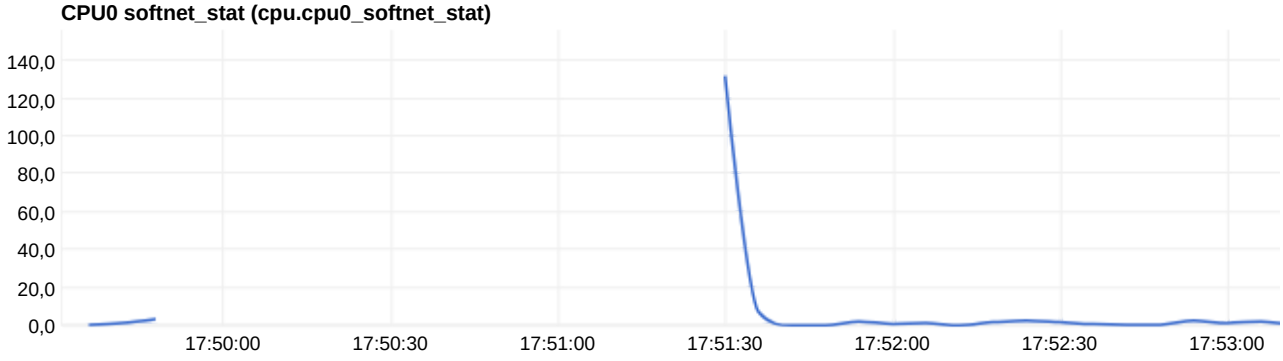
softirqs/s сб, 24 апр. 2021 г. | 17:54:30

TIMER 53,7 NET_TX 0,0 NET_RX 25,4 BLOCK 0,3 SCHED 54,4 RCU 45,3
TASKLET 5,0

softnet

Statistics for per CPUs core SoftIRQs related to network receive work. Total for all CPU cores can be found at System / softnet statistics. **processed** states the number of packets processed, **dropped** is the number packets dropped because the network device backlog was full (to fix them on Linux use `sysctl` to increase `net.core.netdev_max_backlog`), **squeezed** is the number of packets dropped because the network device budget ran out (to fix them on Linux use `sysctl` to increase `net.core.netdev_budget` and/or `net.core.netdev_budget_usecs`). More information about identifying and troubleshooting network driver related issues can be found at Red Hat Enterprise Linux Network Performance Tuning Guide (https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf).

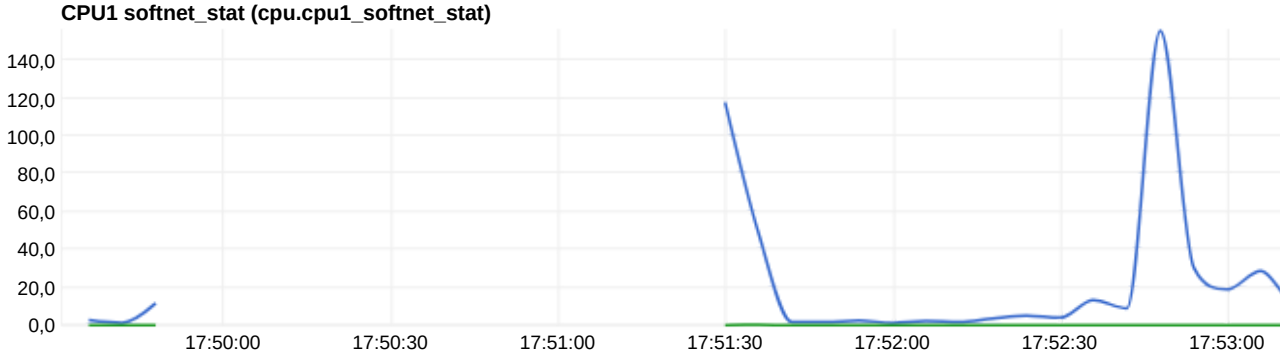
CPU0 softnet_stat (cpu.cpu0_softnet_stat)



events/s сб, 24 апр. 2021 г. | 17:54:30

processed 1,3

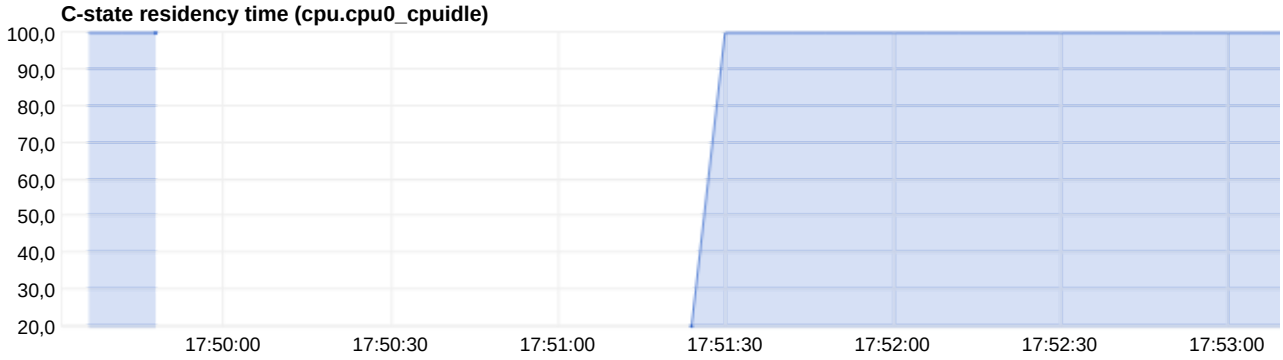
CPU1 softnet_stat (cpu.cpu1_softnet_stat)

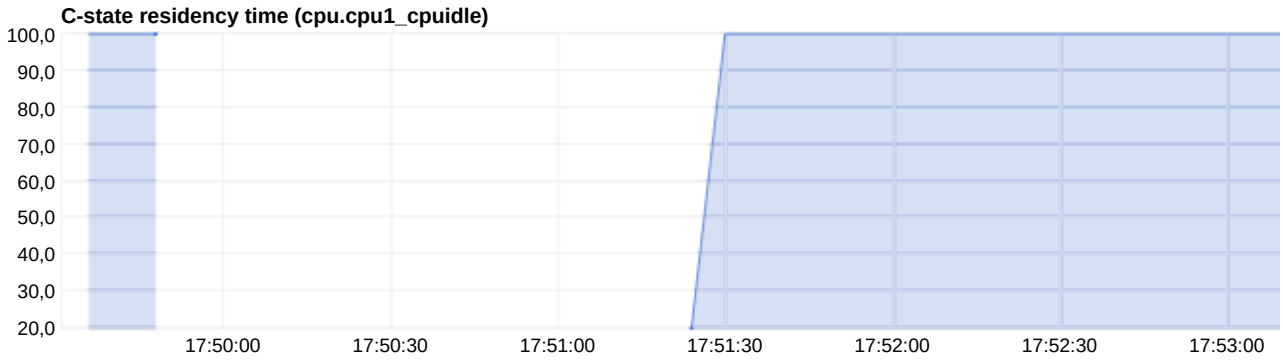


events/s сб, 24 апр. 2021 г. | 17:54:30

processed 24,5 squeezed 0,0

cpuidle



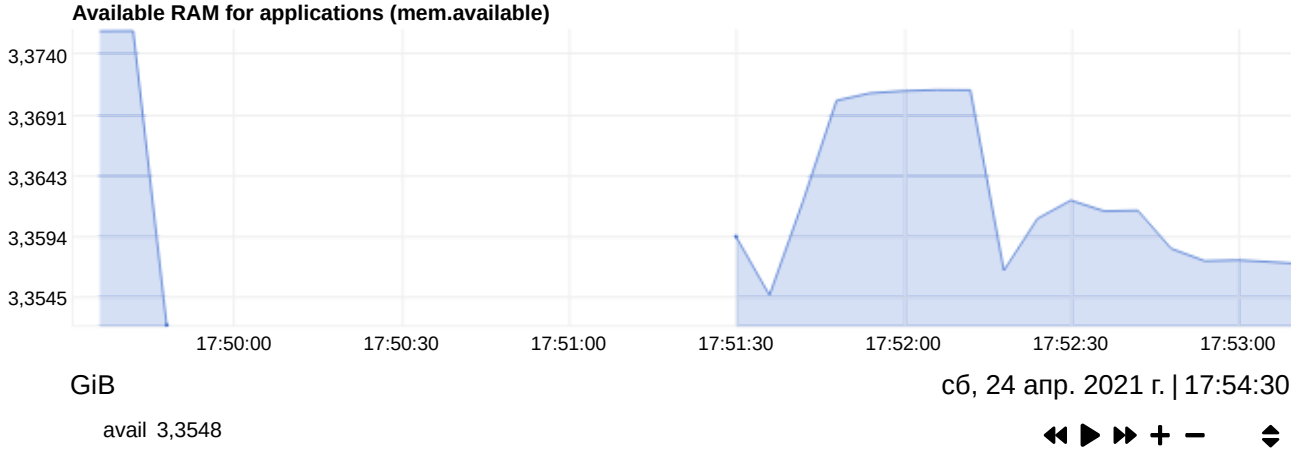


Memory

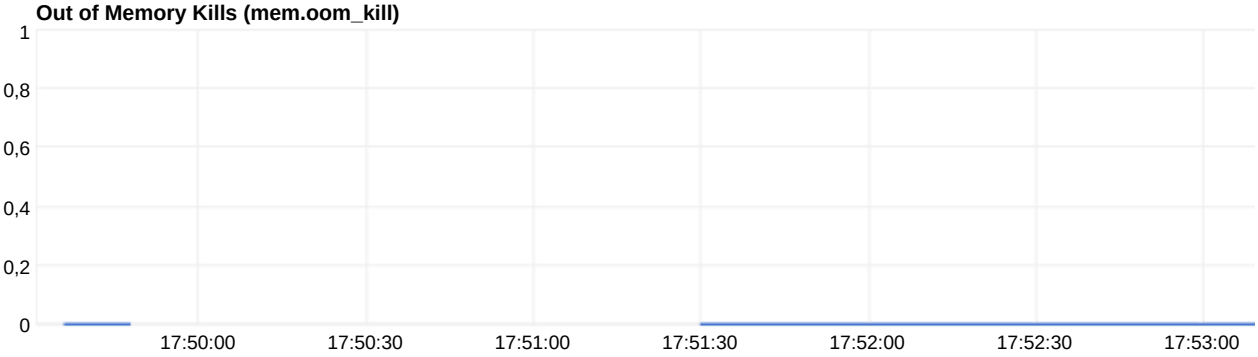
Detailed information about the memory management of the system.

system

Available Memory is estimated by the kernel, as the amount of RAM that can be used by userspace processes, without causing swapping.



Out of Memory Kills (mem.oom_kill)

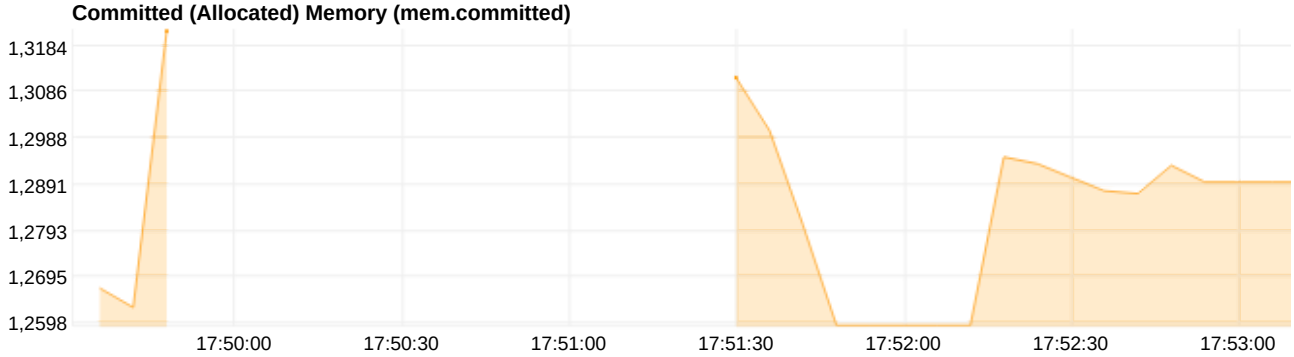


kills/s сб, 24 апр. 2021 г. | 17:54:30

kills 0

Committed Memory, is the sum of all memory which has been allocated by processes.

Committed (Allocated) Memory (mem.committed)

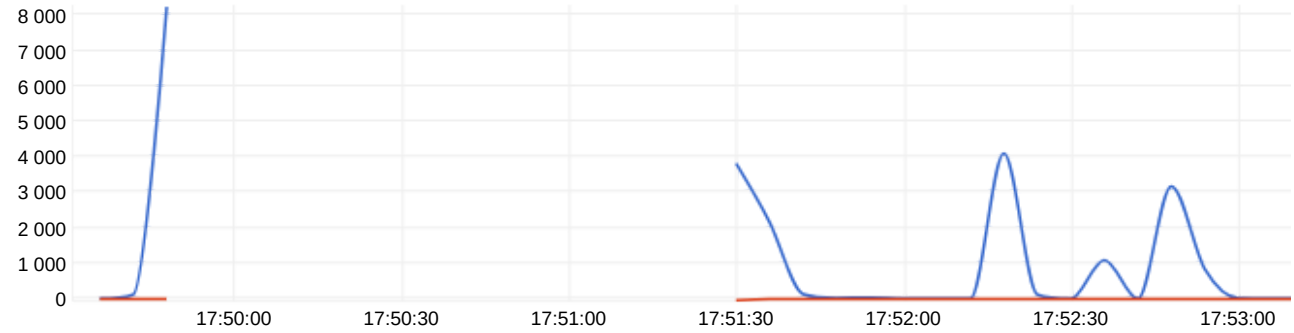


GiB сб, 24 апр. 2021 г. | 17:54:30

Committed_AS 1,2896

A page fault (https://en.wikipedia.org/wiki/Page_fault) is a type of interrupt, called trap, raised by computer hardware when a running program accesses a memory page that is mapped into the virtual address space, but not actually loaded into main memory. If the page is loaded in memory at the time the fault is generated, but is not marked in the memory management unit as being loaded in memory, then it is called a **minor** or soft page fault. A **major** page fault is generated when the system needs to load the memory page from disk or swap memory.

Memory Page Faults (mem.pgfaults)



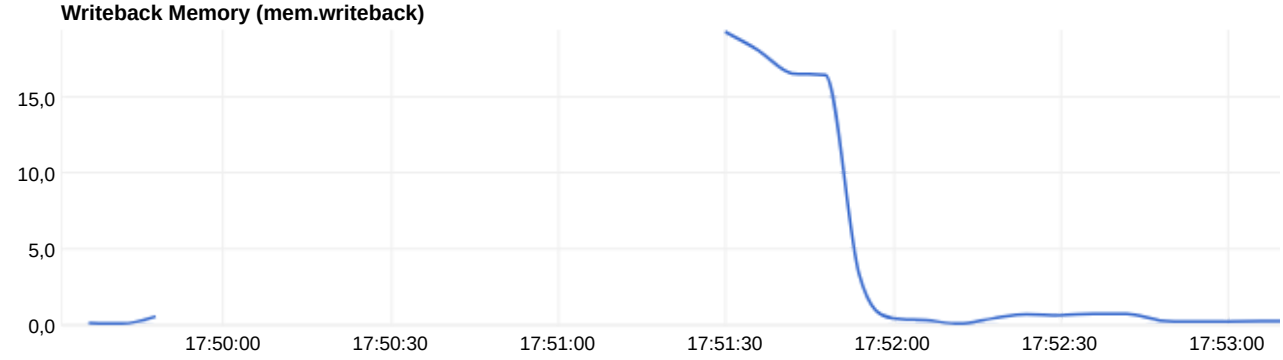
faults/s сб, 24 апр. 2021 г. | 17:54:30

minor 1 major 0

kernel

Dirty is the amount of memory waiting to be written to disk. **Writeback** is how much memory is actively being written to disk.

Writeback Memory (mem.writeback)

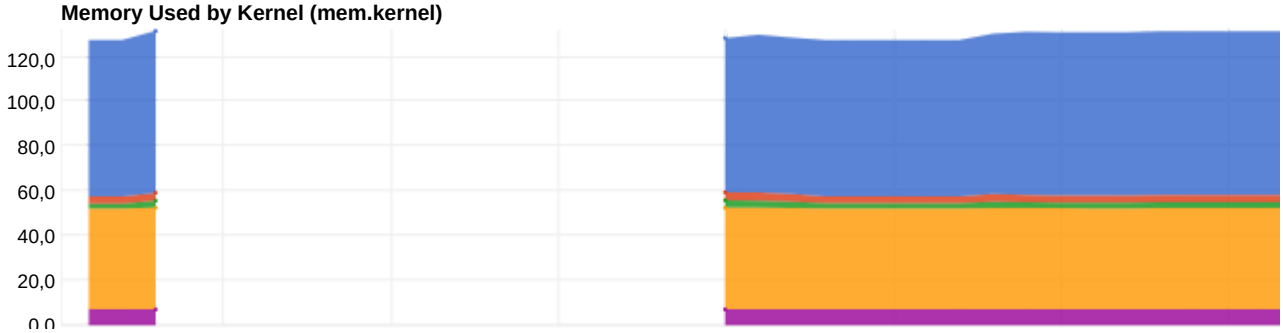


MiB сб, 24 апр. 2021 г. | 17:54:30

Dirty 0,1

The total amount of memory being used by the kernel. **Slab** is the amount of memory used by the kernel to cache data structures for its own use. **KernelStack** is the amount of memory allocated for each task done by the kernel. **PageTables** is the amount of memory dedicated to the lowest level of page tables (A page table is used to turn a virtual address into a physical memory address). **VmallocUsed** is the amount of memory being used as virtual address space.

Memory Used by Kernel (mem.kernel)



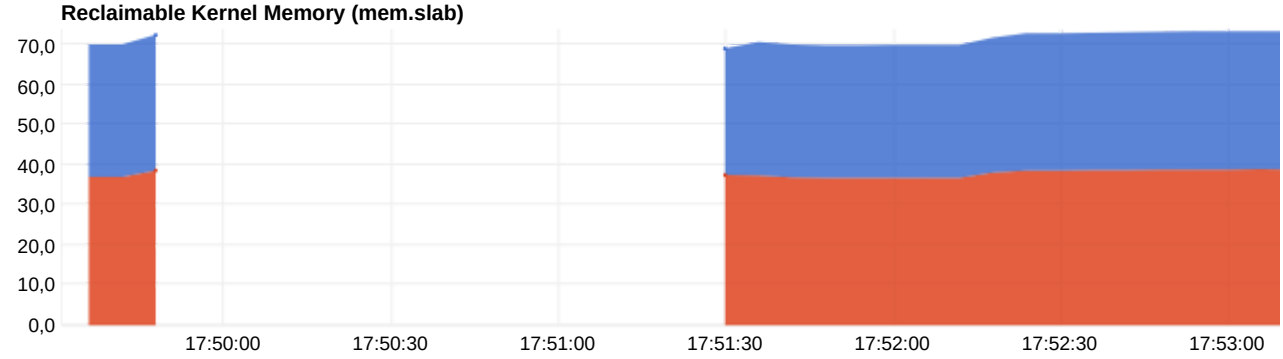
MiB сб, 24 апр. 2021 г. | 17:54:30

Slab 73,5 KernelStack 3,3 PageTables 2,6 VmallocUsed 45,3 Percpu 7,2

slab

Reclaimable is the amount of memory which the kernel can reuse. **Unreclaimable** can not be reused even when the kernel is lacking memory.

Reclaimable Kernel Memory (mem.slab)

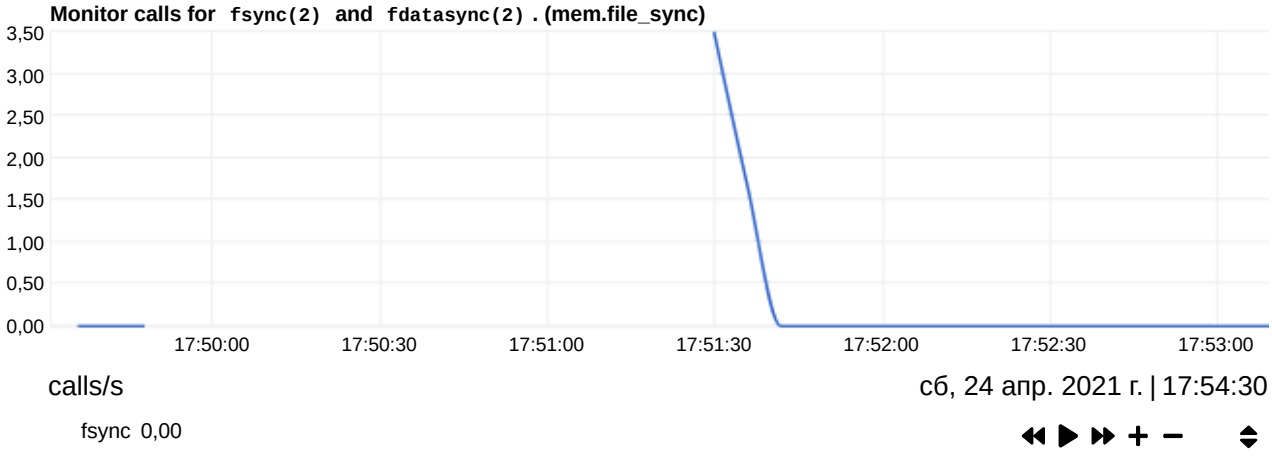


MiB сб, 24 апр. 2021 г. | 17:54:30

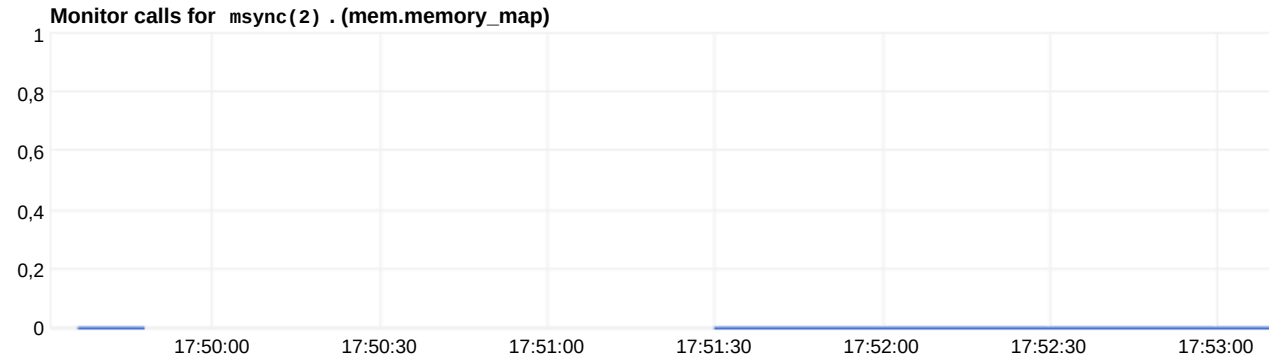
reclaimable 34,5 unreclaimable 39,0

synchronization (eBPF)

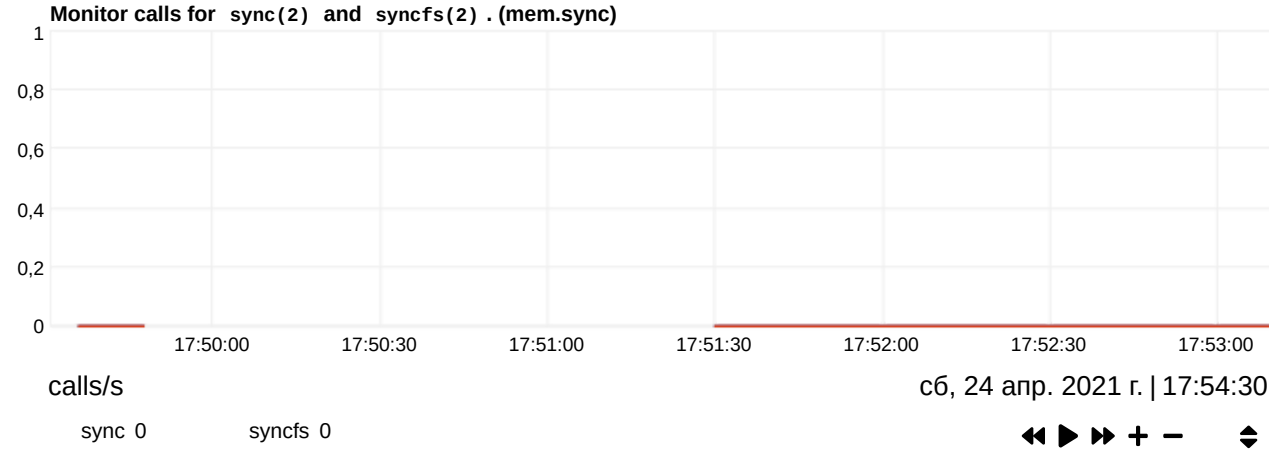
System calls for fsync() and fdatasync() (https://man7.org/linux/man-pages/man2/fsync.2.html) transfer all modified page caches for the files on disk devices. These calls block until the device reports that the transfer has been completed.



System calls for msync() (https://man7.org/linux/man-pages/man2/msync.2.html) which flushes changes made to the in-core copy of a file that was mapped.

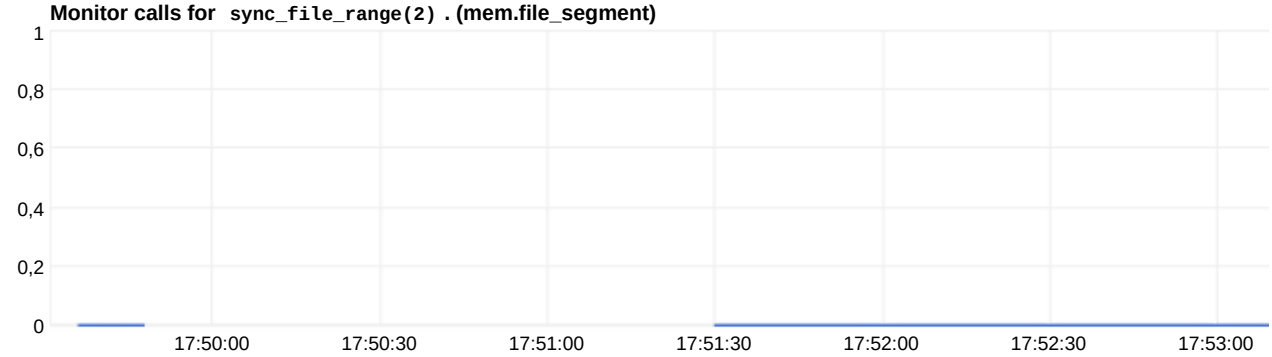


System calls for sync() and syncfs() (https://man7.org/linux/man-pages/man2/sync.2.html) which flush the file system buffers to storage devices. Performance perturbations might be caused by these calls. The `sync()` calls are based on the eBPF syncsnoop (https://github.com/iovisor/bcc/blob/master/tools/syncsnoop.py) from BCC tools.



System calls for sync_file_range() (https://man7.org/linux/man-pages/man2/sync_file_range.2.html) permits fine control when synchronizing the open file referred to by the file descriptor fd with disk. This system call is extremely dangerous and should not be used in portable programs.

Monitor calls for `sync_file_range(2)`. (mem.file_segment)



calls/s сб, 24 апр. 2021 г. | 17:54:30

sync_file_range 0

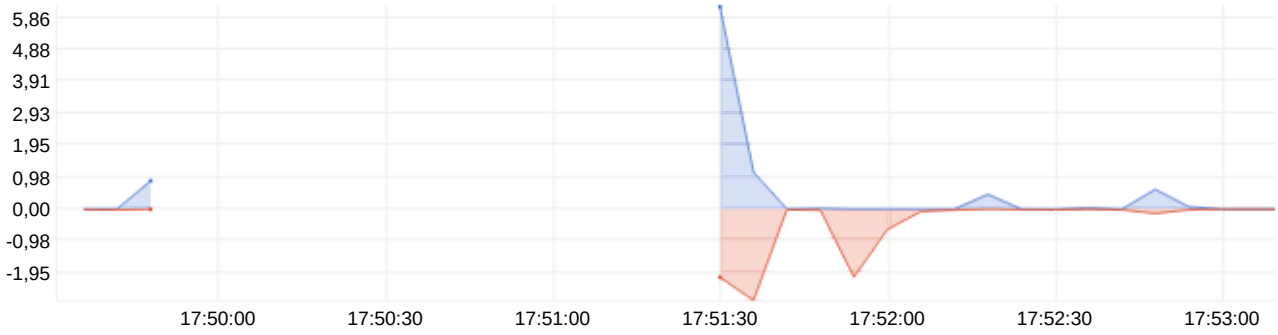
Disks

Charts with performance information for all the system disks. Special care has been given to present disk performance metrics in a way compatible with `iostat -x`. netdata by default prevents rendering performance charts for individual partitions and unmounted virtual disks. Disabled charts can still be enabled by configuring the relative settings in the netdata configuration file.

sda

Amount of data transferred to and from disk.

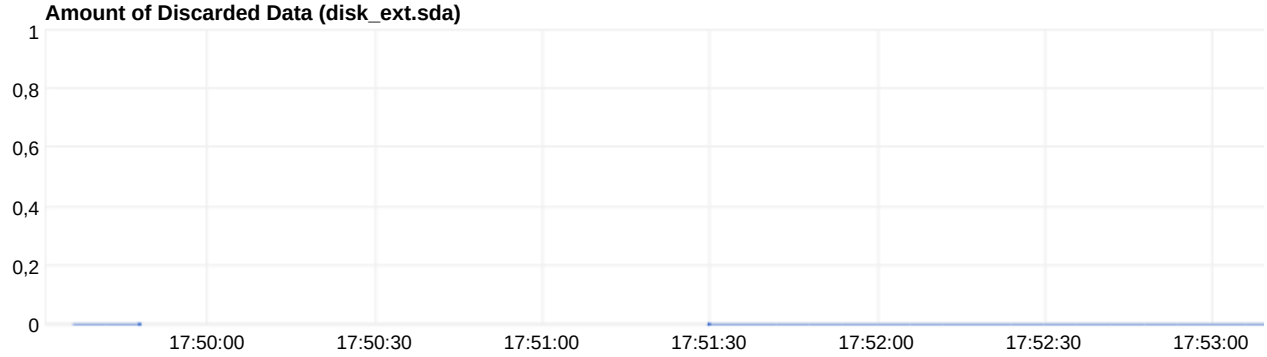
Disk I/O Bandwidth (disk.sda)



MiB/s сб, 24 апр. 2021 г. | 17:54:30

reads 0,03 writes -0,00

Amount of Discarded Data (disk_ext.sda)

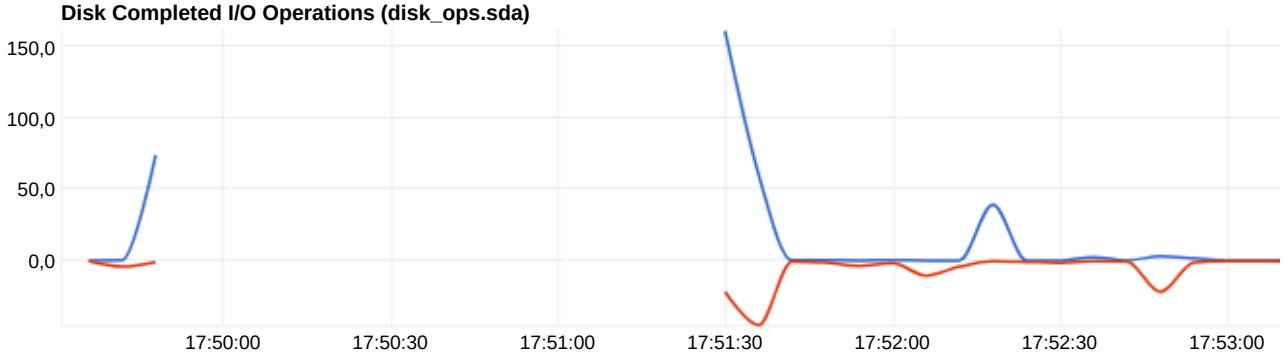


KiB/s сб, 24 апр. 2021 г. | 17:54:30

discards 0

Completed disk I/O operations. Keep in mind the number of operations requested might be higher, since the system is able to merge adjacent to each other (see merged operations chart).

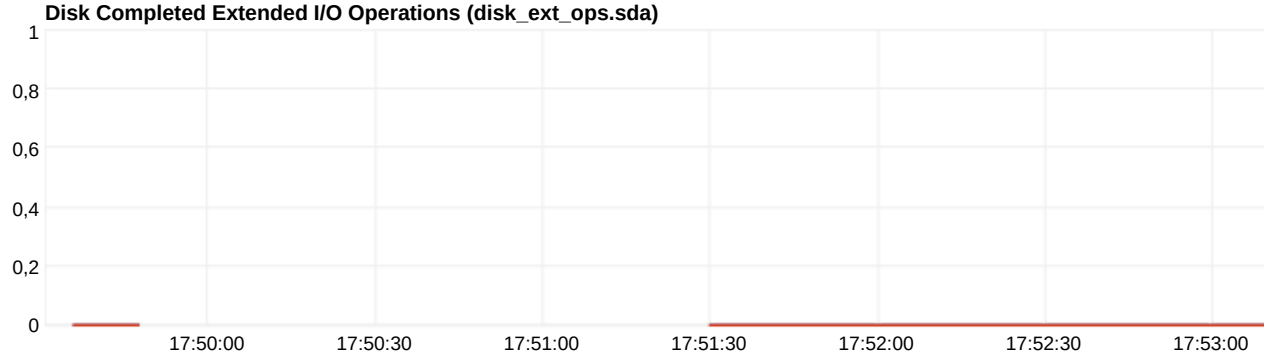
Disk Completed I/O Operations (disk_ops.sda)



operations/s сб, 24 апр. 2021 г. | 17:54:30

reads 0,3 writes -0,5

Disk Completed Extended I/O Operations (disk_ext_ops.sda)

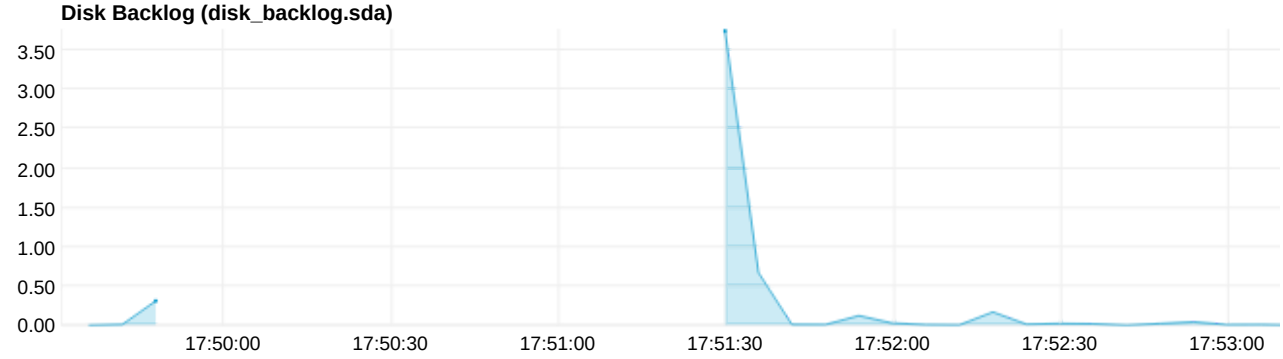


operations/s сб, 24 апр. 2021 г. | 17:54:30

discards 0 flushes 0

Backlog is an indication of the duration of pending disk operations. On every I/O event the system is multiplying the time spent doing I/O since the last update of this field with the number of pending operations. While not accurate, this metric can provide an indication of the expected completion time of the operations in progress.

Disk Backlog (disk_backlog.sda)

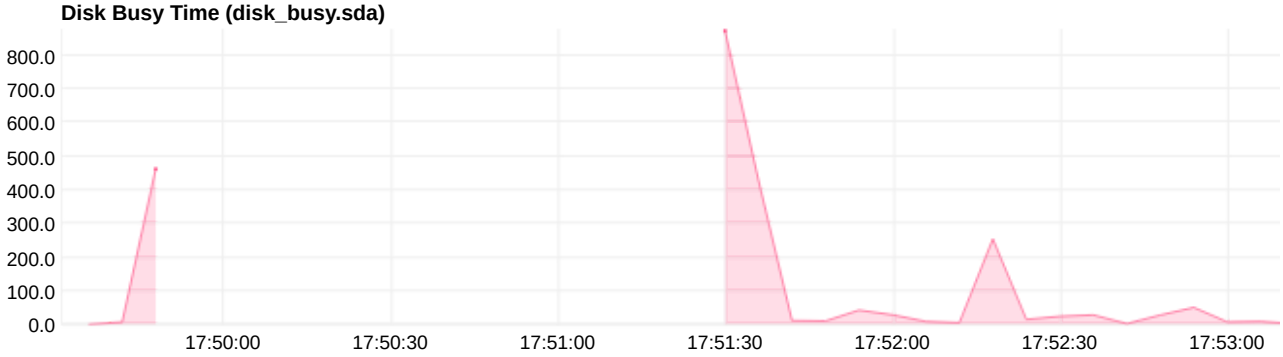


seconds сб, 24 апр. 2021 г. | 17:54:30

backlog 0.02

Disk Busy Time measures the amount of time the disk was busy with something.

Disk Busy Time (disk_busy.sda)

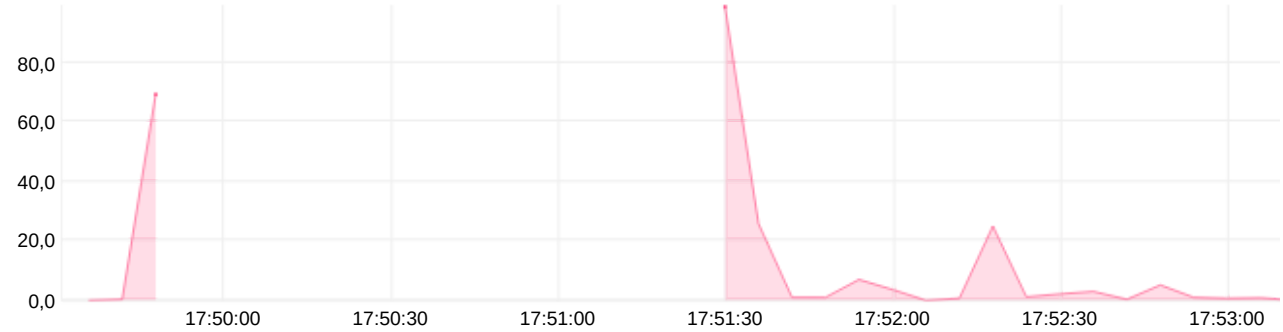


milliseconds сб, 24 апр. 2021 г. | 17:54:30

busy 14.0

Disk Utilization measures the amount of time the disk was busy with something. This is not related to its performance. 100% means that the system always had an outstanding operation on the disk. Keep in mind that depending on the underlying technology of the disk, 100% here may or may not be an indication of congestion.

Disk Utilization Time (disk_util.sda)

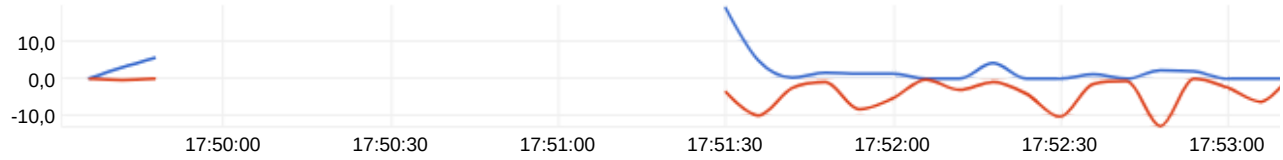


% of time working сб, 24 апр. 2021 г. | 17:54:30

utilization 1,3

The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

Average Completed I/O Operation Time (disk_await.sda)

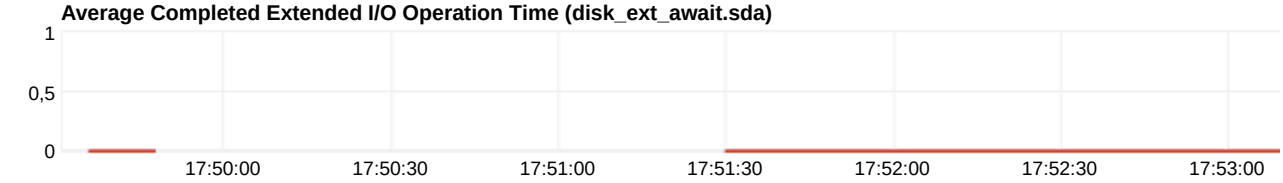


milliseconds/operation сб, 24 апр. 2021 г. | 17:54:30

reads 3,0 writes -3,2

The average time for extended I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

Average Completed Extended I/O Operation Time (disk_ext_await.sda)

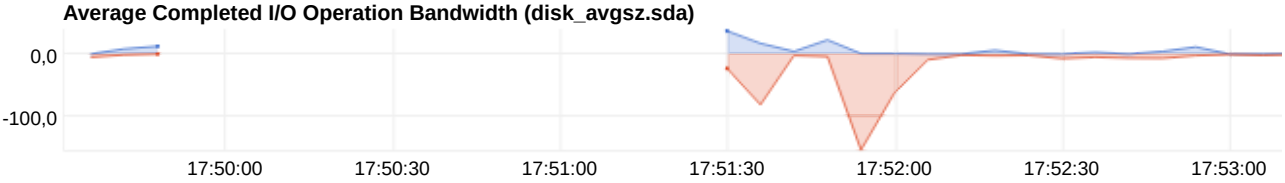


milliseconds/operation сб, 24 апр. 2021 г. | 17:54:30

discards 0 flushes 0

The average I/O operation size.

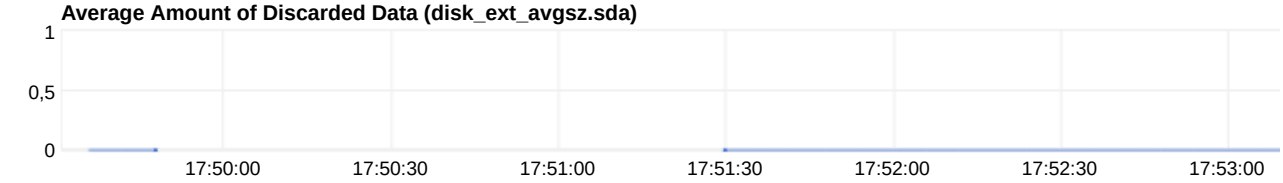
Average Completed I/O Operation Bandwidth (disk_avgsz.sda)



KiB/operation сб, 24 апр. 2021 г. | 17:54:30

reads 13,7 writes -0,8

Average Amount of Discarded Data (disk_ext_avgsz.sda)

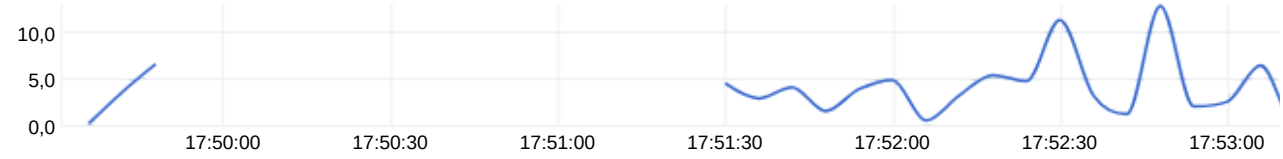


KiB/operation сб, 24 апр. 2021 г. | 17:54:30

discards 0

The average service time for completed I/O operations. This metric is calculated using the total busy time of the disk and the number of completed operations. If the disk is able to execute multiple parallel operations the reporting average service time will be misleading.

Average Service Time (disk_svctm.sda)

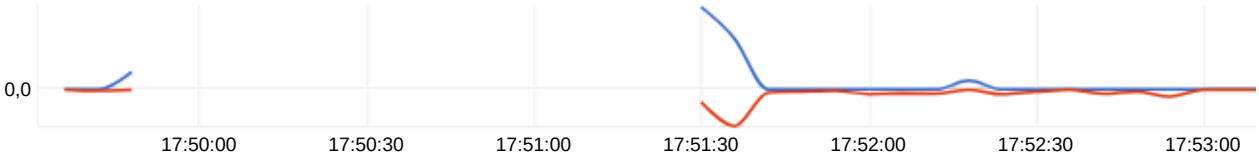


milliseconds/operation сб, 24 апр. 2021 г. | 17:54:30

svctm 5,3

The number of merged disk operations. The system is able to merge adjacent I/O operations, for example two 4KB reads can become one 8KB read before given to disk.

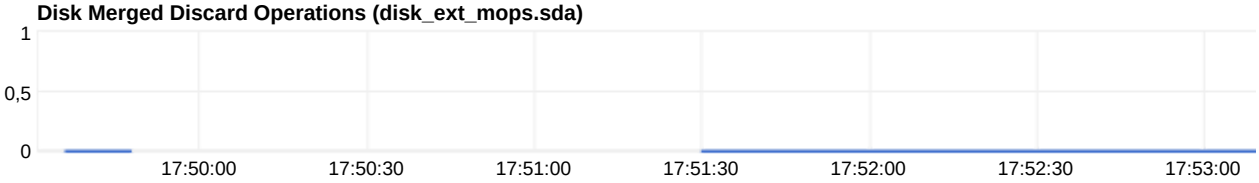
Disk Merged Operations (disk_mops.sda)



merged operations/s сб, 24 апр. 2021 г. | 17:54:30

reads 0,0 writes -0,3

Disk Merged Discard Operations (disk_ext_mops.sda)

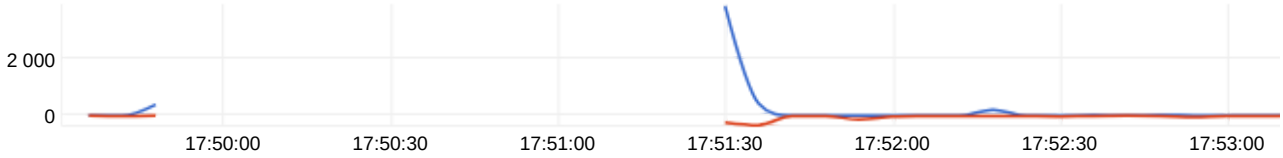


merged operations/s сб, 24 апр. 2021 г. | 17:54:30

discards 0

The sum of the duration of all completed I/O operations. This number can exceed the interval if the disk is able to execute I/O operations in parallel.

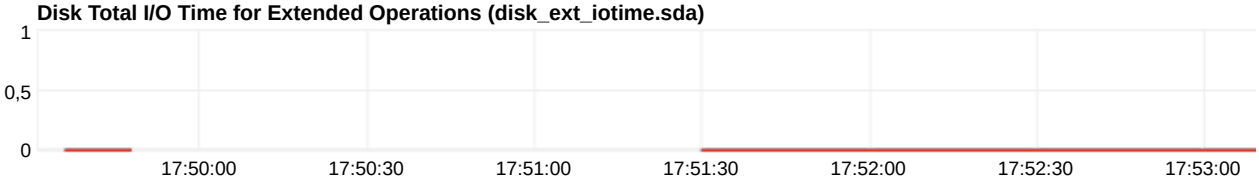
Disk Total I/O Time (disk_iotime.sda)



milliseconds/s сб, 24 апр. 2021 г. | 17:54:30

reads 6 writes -10

Disk Total I/O Time for Extended Operations (disk_ext_iotime.sda)



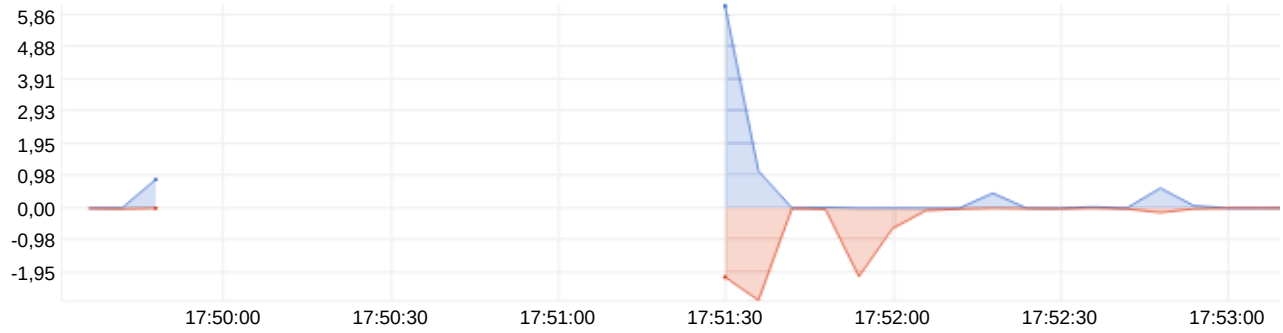
milliseconds/s сб, 24 апр. 2021 г. | 17:54:30

discards 0 flushes 0

vgvagrant-root

Amount of data transferred to and from disk.

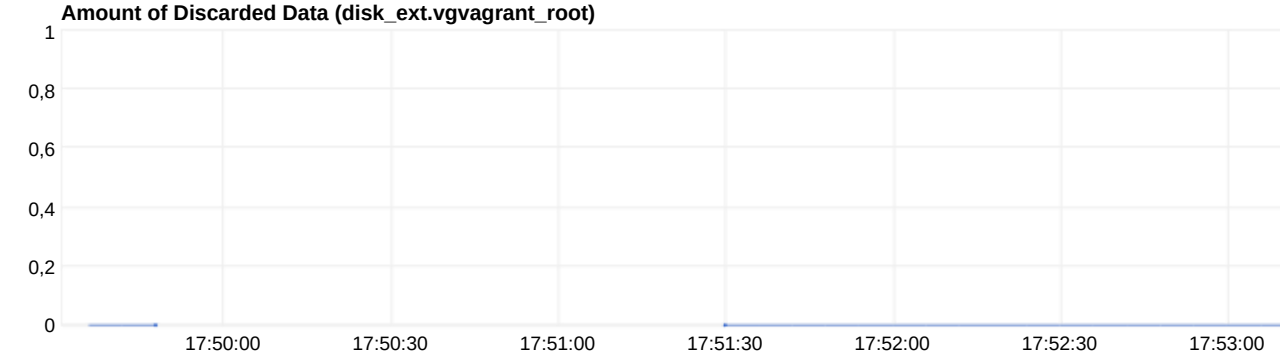
Disk I/O Bandwidth (disk.vgvagrant_root)



MiB/s сб, 24 апр. 2021 г. | 17:54:30

reads 0,03 writes -0,00

Amount of Discarded Data (disk_ext.vgvagrant_root)

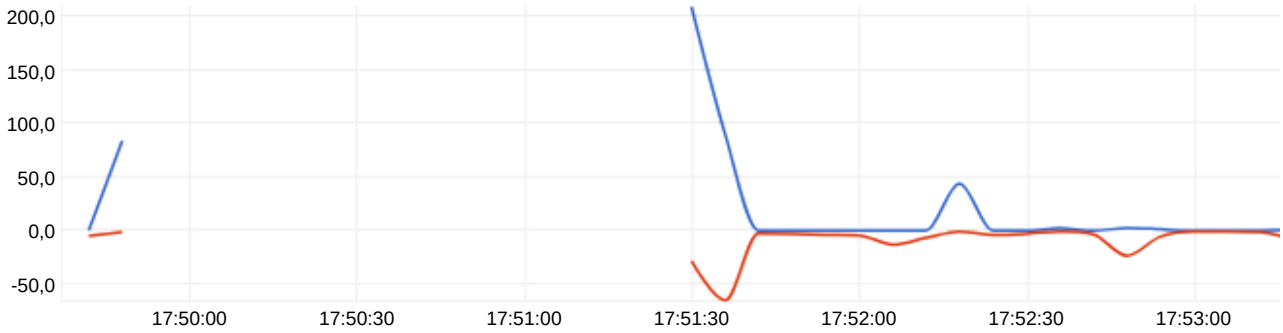


KiB/s сб, 24 апр. 2021 г. | 17:54:30

discards 0

Completed disk I/O operations. Keep in mind the number of operations requested might be higher, since the system is able to merge adjacent to each other (see merged operations chart).

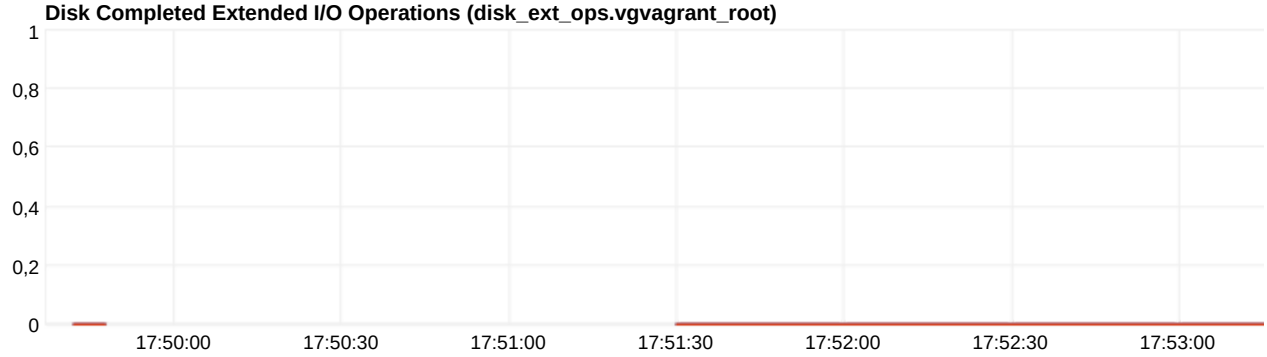
Disk Completed I/O Operations (disk_ops.vgvagrant_root)



operations/s сб, 24 апр. 2021 г. | 17:54:36

reads 0,2 writes -1,0

Disk Completed Extended I/O Operations (disk_ext_ops.vgvagrant_root)



operations/s сб, 24 апр. 2021 г. | 17:54:36

discards 0 flushes 0

I/O operations currently in progress. This metric is a snapshot - it is not an average over the last interval.

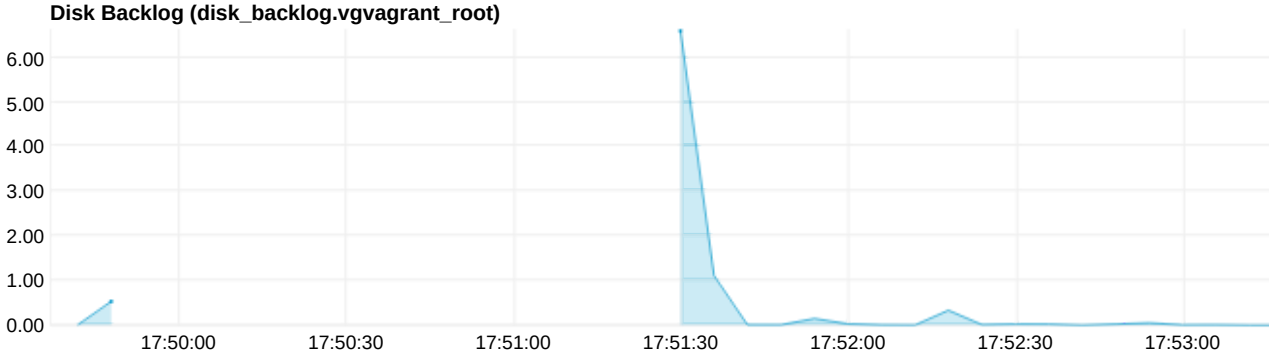
Disk Current I/O Operations (disk_qops.vgvagrant_root)

operations сб, 24 апр. 2021 г. | 17:54:36

operations 0,00

Backlog is an indication of the duration of pending disk operations. On every I/O event the system is multiplying the time spent doing I/O since the last update of this field with the number of pending operations. While not accurate, this metric can provide an indication of the expected completion time of the operations in progress.

Disk Backlog (disk_backlog.vgvagrant_root)

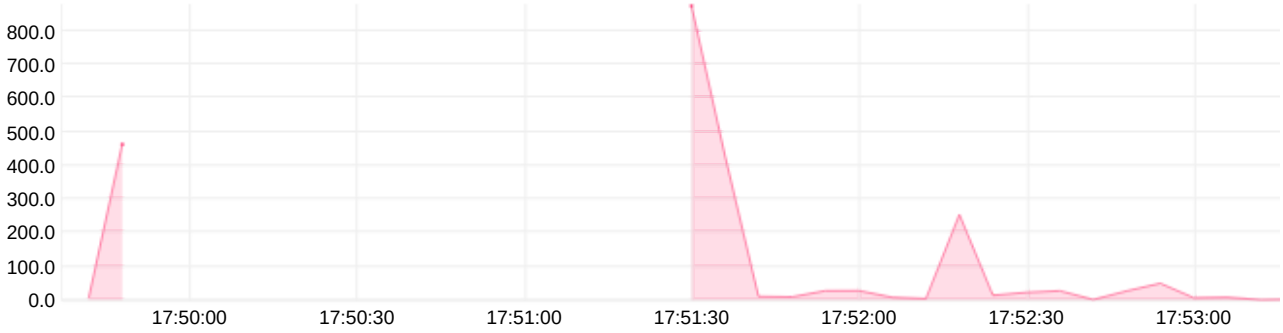


seconds сб, 24 апр. 2021 г. | 17:54:36

backlog 0.01

Disk Busy Time measures the amount of time the disk was busy with something.

Disk Busy Time (disk_busy.vgvagrant_root)

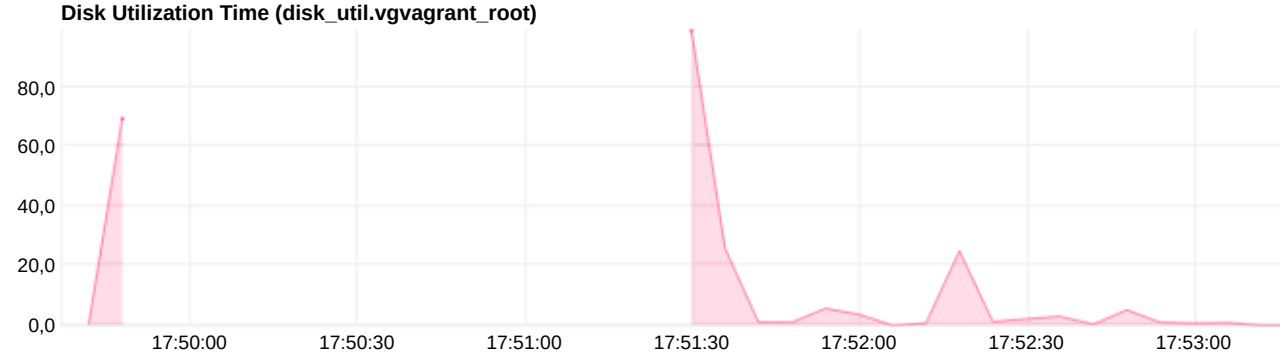


milliseconds сб, 24 апр. 2021 г. | 17:54:36

busy 11.3

Disk Utilization measures the amount of time the disk was busy with something. This is not related to its performance. 100% means that the system always had an outstanding operation on the disk. Keep in mind that depending on the underlying technology of the disk, 100% here may or may not be an indication of congestion.

Disk Utilization Time (disk_util.vgvagrant_root)



% of time working сб, 24 апр. 2021 г. | 17:54:36

utilization 1,0

The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

Average Completed I/O Operation Time (disk_await.vgvagrant_root)

milliseconds/operation сб, 24 апр. 2021 г. | 17:54:36

reads 0,0 writes -1,7

The average time for extended I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

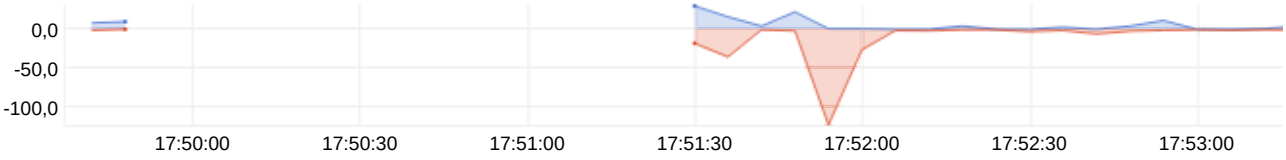
Average Completed Extended I/O Operation Time (disk_ext_await.vgvagrant_root)

milliseconds/operation сб, 24 апр. 2021 г. | 17:54:36

discards 0 flushes 0

The average I/O operation size.

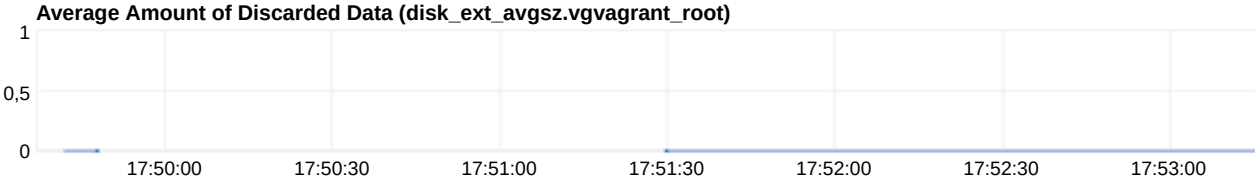
Average Completed I/O Operation Bandwidth (disk_avgsz.vgvagrant_root)



KiB/operation сб, 24 апр. 2021 г. | 17:54:36

reads 1,3 writes -0,7

Average Amount of Discarded Data (disk_ext_avgsz.vgvagrant_root)

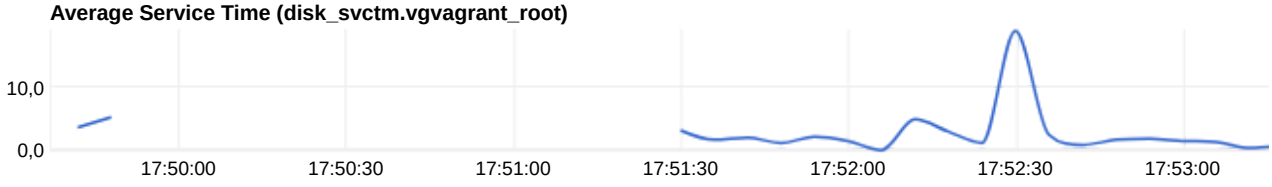


KiB/operation сб, 24 апр. 2021 г. | 17:54:36

discards 0

The average service time for completed I/O operations. This metric is calculated using the total busy time of the disk and the number of completed operations. If the disk is able to execute multiple parallel operations the reporting average service time will be misleading.

Average Service Time (disk_svctm.vgvagrant_root)

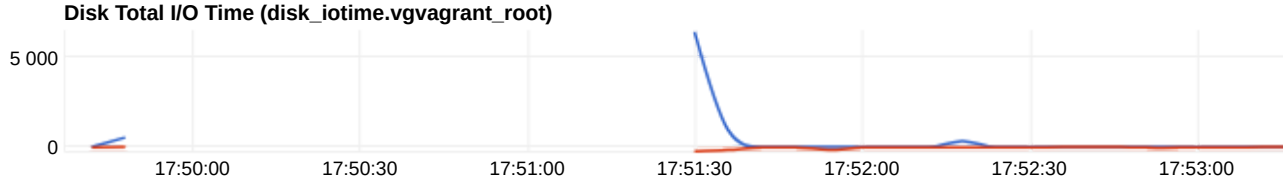


milliseconds/operation сб, 24 апр. 2021 г. | 17:54:36

svctm 2,3

The sum of the duration of all completed I/O operations. This number can exceed the interval if the disk is able to execute I/O operations in parallel.

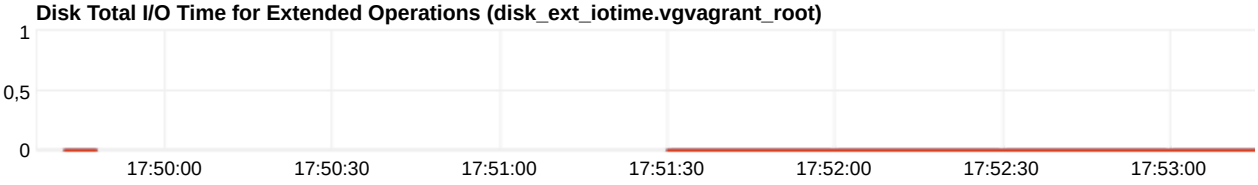
Disk Total I/O Time (disk_iotime.vgvagrant_root)



milliseconds/s сб, 24 апр. 2021 г. | 17:54:36

reads 0 writes -10

Disk Total I/O Time for Extended Operations (disk_ext_iotime.vgvagrant_root)



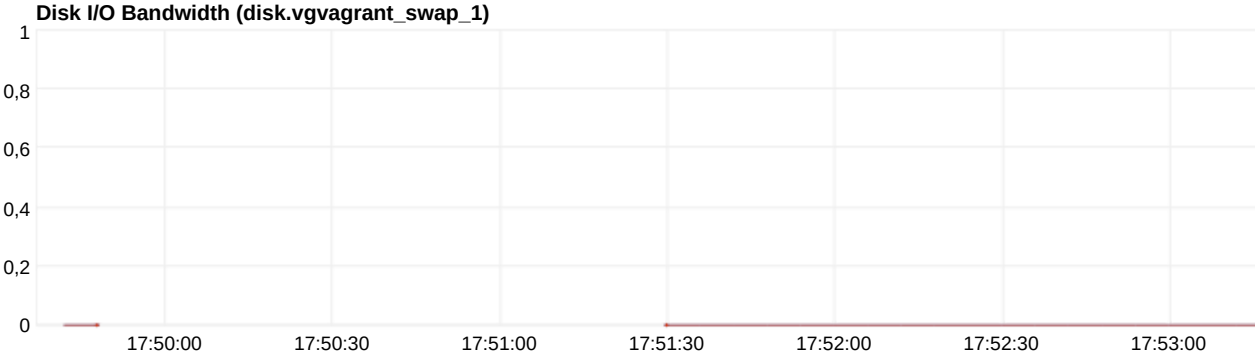
milliseconds/s сб, 24 апр. 2021 г. | 17:54:36

discards 0 flushes 0

vgvagrant-swap 1

Amount of data transferred to and from disk.

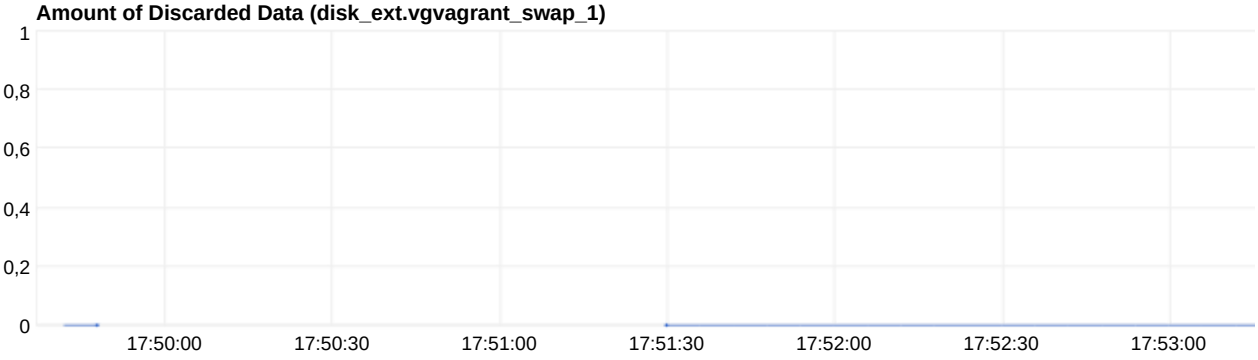
Disk I/O Bandwidth (disk.vgvagrant_swap_1)



KiB/s сб, 24 апр. 2021 г. | 17:54:36

reads 0 writes 0

Amount of Discarded Data (disk_ext.vgvagrant_swap_1)

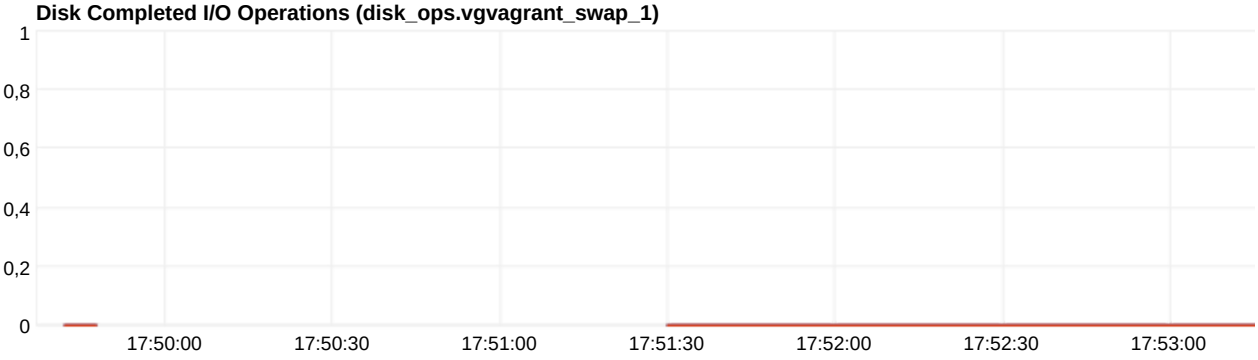


KiB/s сб, 24 апр. 2021 г. | 17:54:36

discards 0

Completed disk I/O operations. Keep in mind the number of operations requested might be higher, since the system is able to merge adjacent to each other (see merged operations chart).

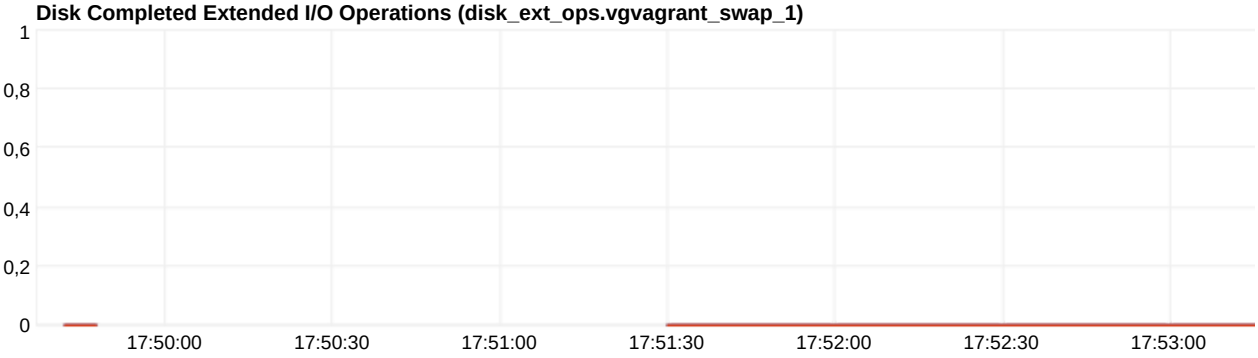
Disk Completed I/O Operations (disk_ops.vgvagrant_swap_1)



operations/s сб, 24 апр. 2021 г. | 17:54:36

reads 0 writes 0

Disk Completed Extended I/O Operations (disk_ext_ops.vgvagrant_swap_1)

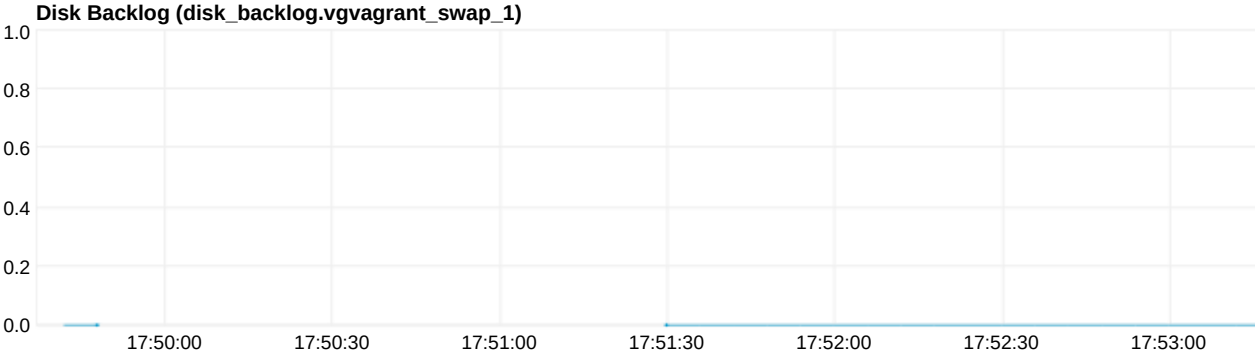


operations/s сб, 24 апр. 2021 г. | 17:54:36

discards 0 flushes 0

Backlog is an indication of the duration of pending disk operations. On every I/O event the system is multiplying the time spent doing I/O since the last update of this field with the number of pending operations. While not accurate, this metric can provide an indication of the expected completion time of the operations in progress.

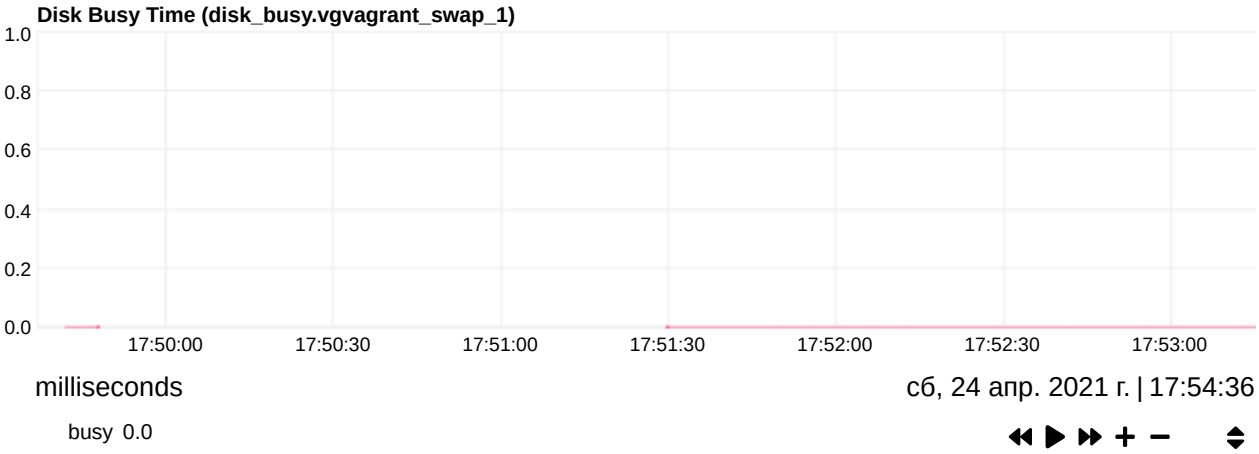
Disk Backlog (disk_backlog.vgvagrant_swap_1)



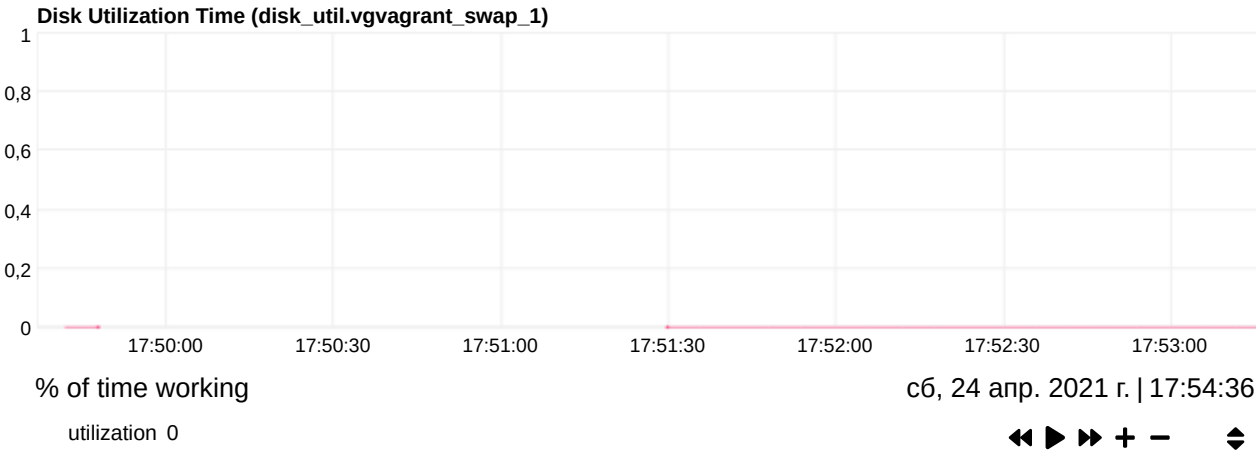
milliseconds сб, 24 апр. 2021 г. | 17:54:36

backlog 0.0

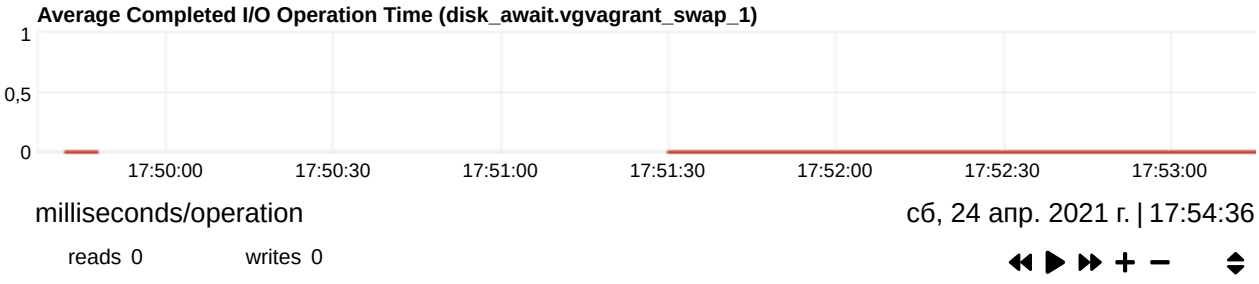
Disk Busy Time measures the amount of time the disk was busy with something.



Disk Utilization measures the amount of time the disk was busy with something. This is not related to its performance. 100% means that the system always had an outstanding operation on the disk. Keep in mind that depending on the underlying technology of the disk, 100% here may or may not be an indication of congestion.



The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.



The average time for extended I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

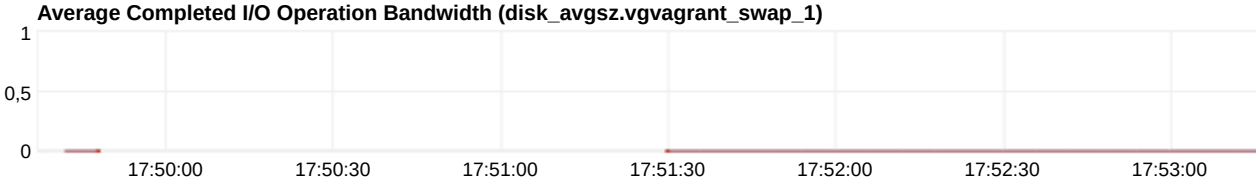
Average Completed Extended I/O Operation Time (disk_ext_await.vgvagrant_swap_1)

milliseconds/operation сб, 24 апр. 2021 г. | 17:54:36

discards 0 flushes 0

The average I/O operation size.

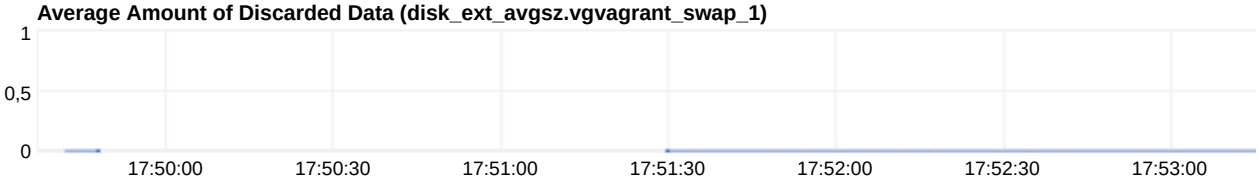
Average Completed I/O Operation Bandwidth (disk_avgsz.vgvagrant_swap_1)



KiB/operation сб, 24 апр. 2021 г. | 17:54:36

reads 0 writes 0

Average Amount of Discarded Data (disk_ext_avgsz.vgvagrant_swap_1)

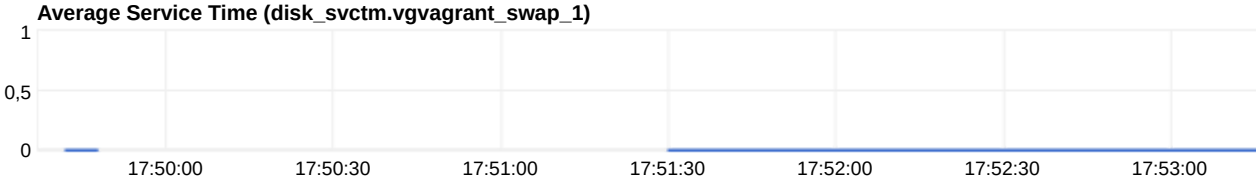


KiB/operation сб, 24 апр. 2021 г. | 17:54:36

discards 0

The average service time for completed I/O operations. This metric is calculated using the total busy time of the disk and the number of completed operations. If the disk is able to execute multiple parallel operations the reporting average service time will be misleading.

Average Service Time (disk_svctm.vgvagrant_swap_1)

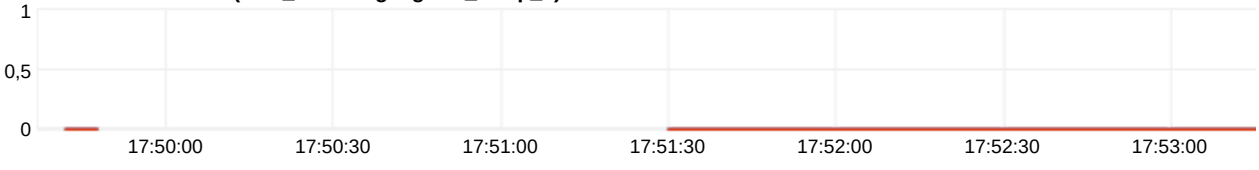


milliseconds/operation сб, 24 апр. 2021 г. | 17:54:36

svctm 0

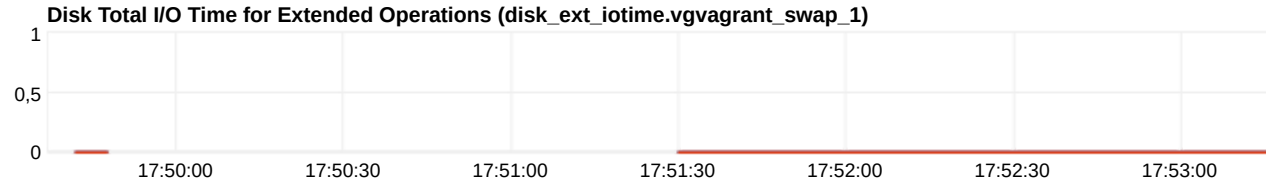
The sum of the duration of all completed I/O operations. This number can exceed the interval if the disk is able to execute I/O operations in parallel.

Disk Total I/O Time (disk_iotime.vgvagrant_swap_1)



milliseconds/s сб, 24 апр. 2021 г. | 17:54:36

reads 0 writes 0

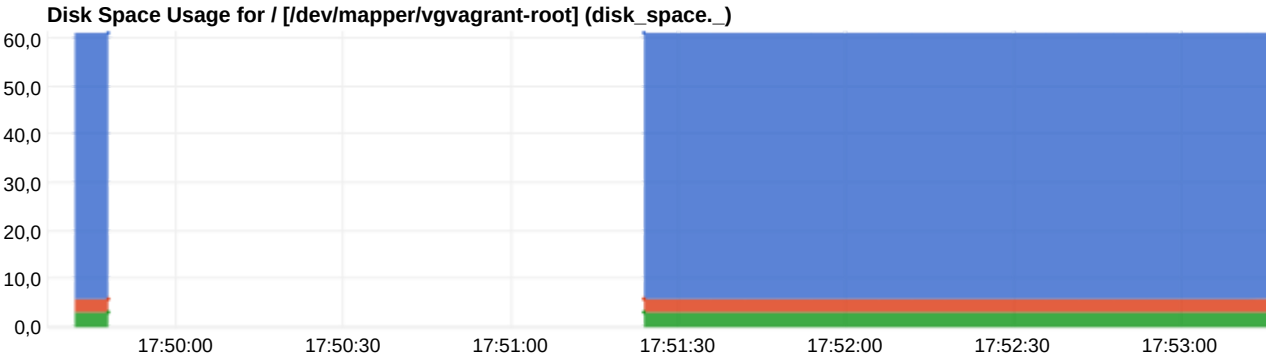


milliseconds/s сб, 24 апр. 2021 г. | 17:54:36

discards 0 flushes 0

/

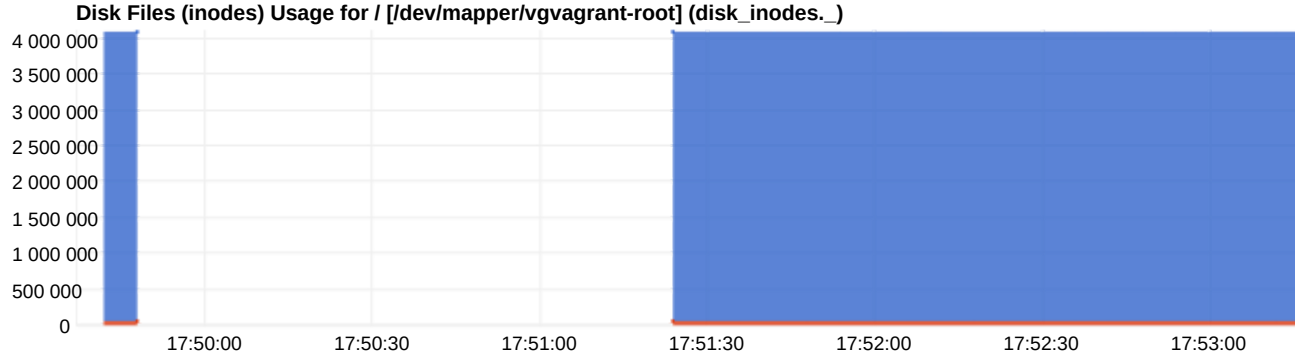
Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.



GiB сб, 24 апр. 2021 г. | 17:54:36

avail 55,4 used 2,8 reserved for root 3,1

inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.



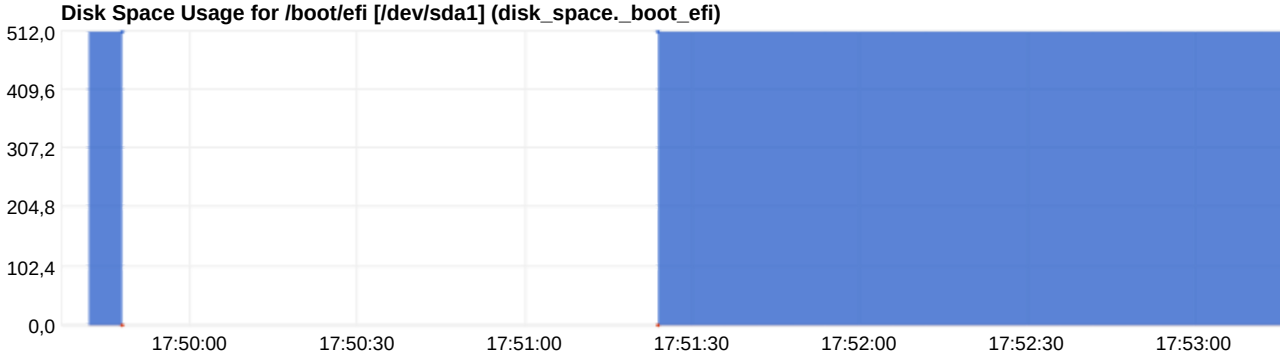
inodes сб, 24 апр. 2021 г. | 17:54:36

avail 4 043 535 used 60 657

/boot/efi

Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.

Disk Space Usage for /boot/efi [/dev/sda1] (disk_space._boot_efi)



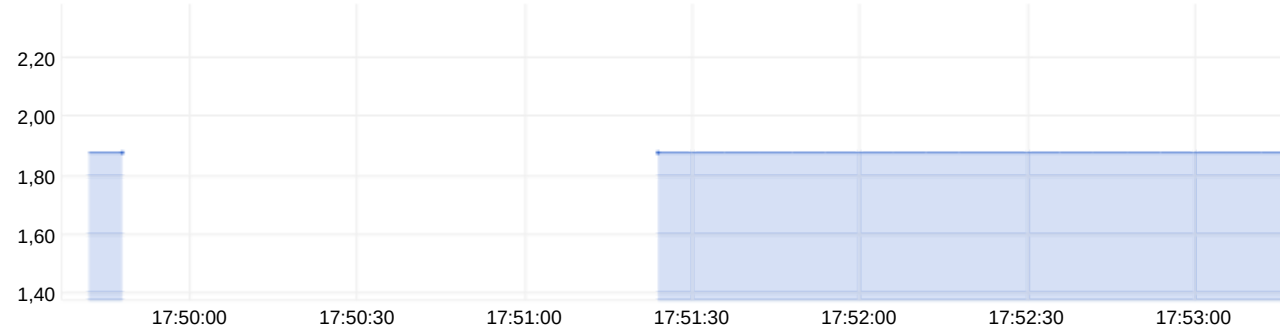
MiB сб, 24 апр. 2021 г. | 17:54:36

avail 511,0 used 0,0

/dev

Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.

Disk Space Usage for /dev [udev] (disk_space._dev)

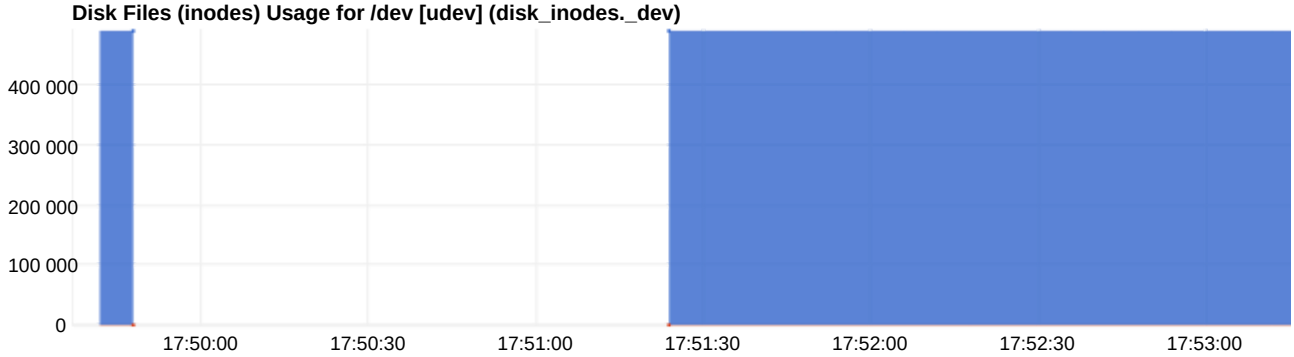


GiB сб, 24 апр. 2021 г. | 17:54:36

avail 1,88

inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.

Disk Files (inodes) Usage for /dev [udev] (disk_inodes._dev)

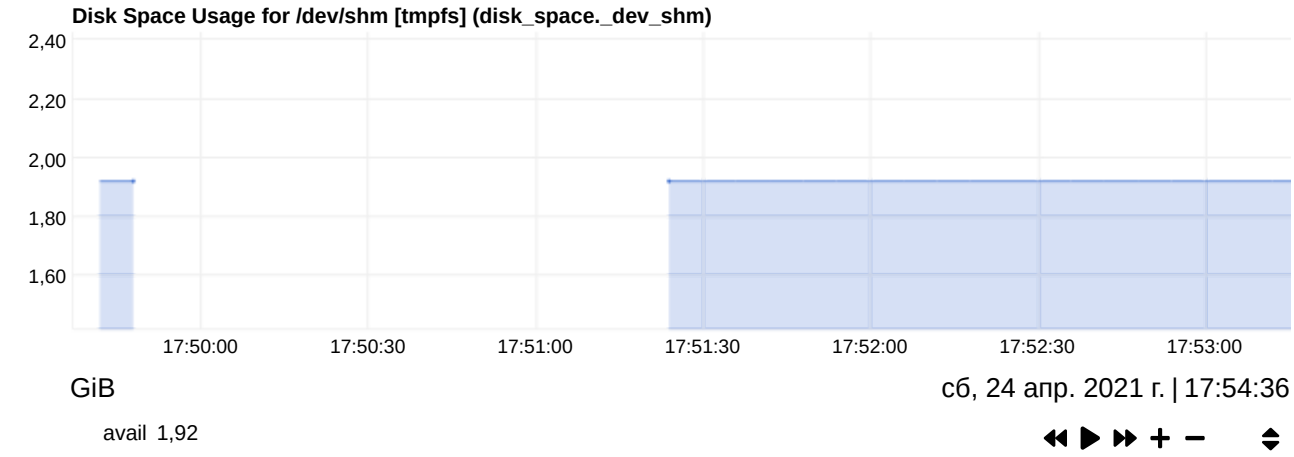


inodes сб, 24 апр. 2021 г. | 17:54:36

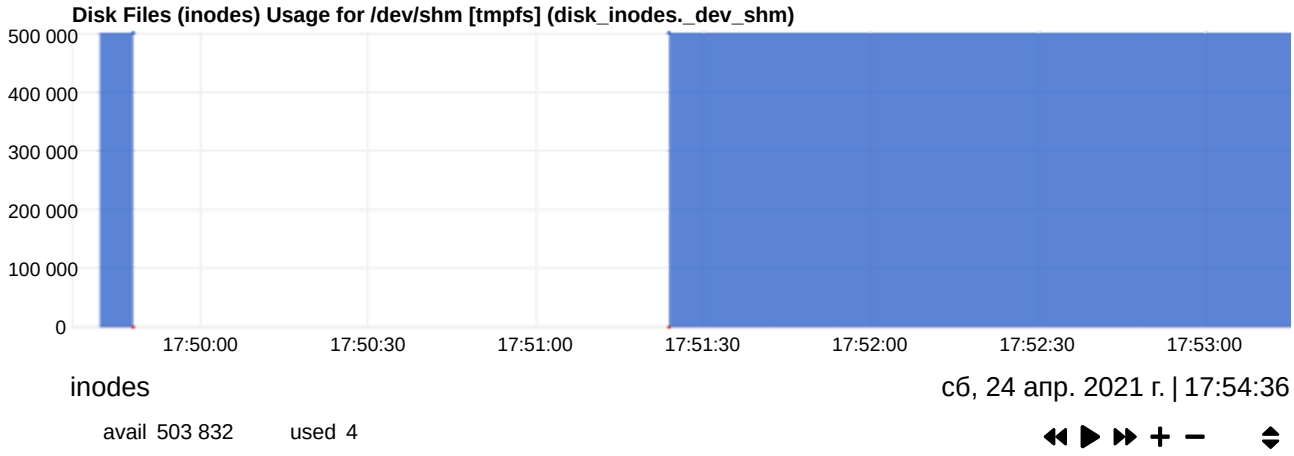
avail 492 360 used 442

/dev/shm

Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.

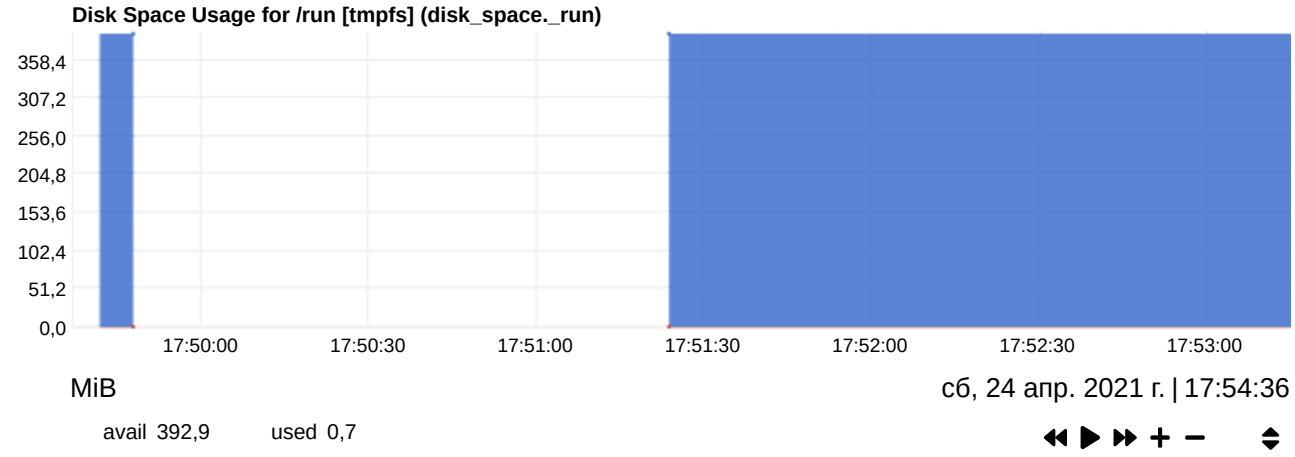


inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.

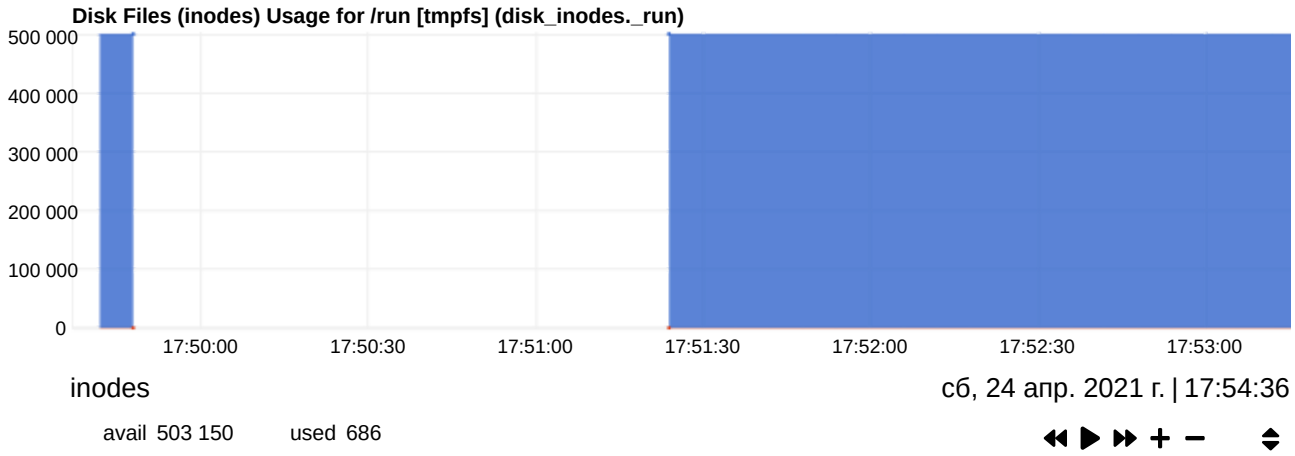


/run

Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.

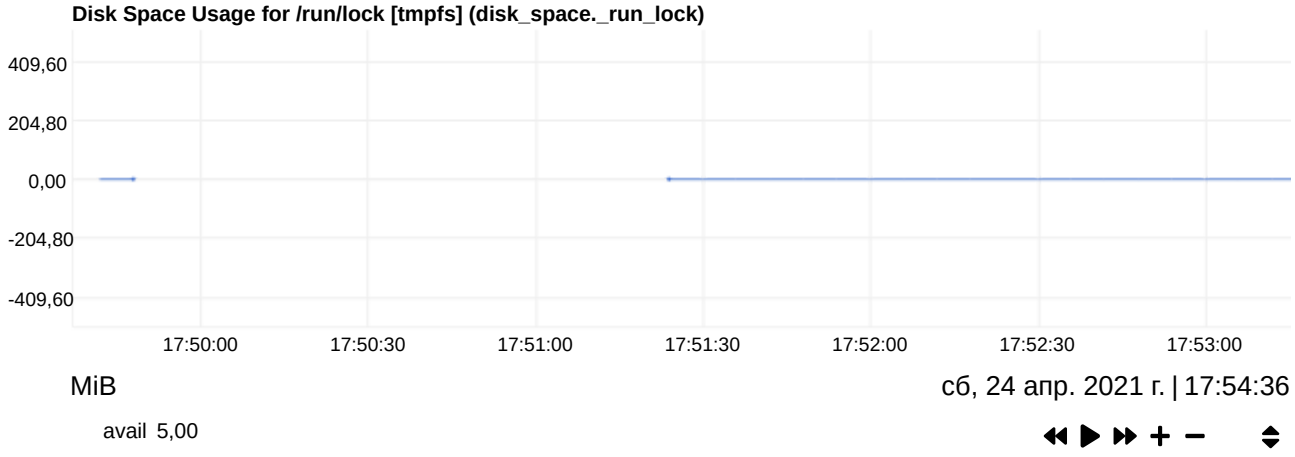


inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.



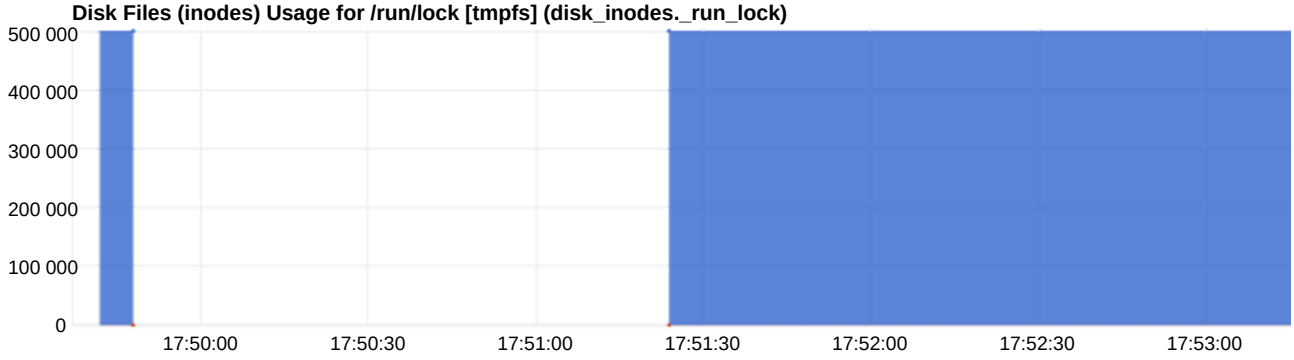
/run/lock

Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.



inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.

Disk Files (inodes) Usage for /run/lock [tmpfs] (disk_inodes._run_lock)



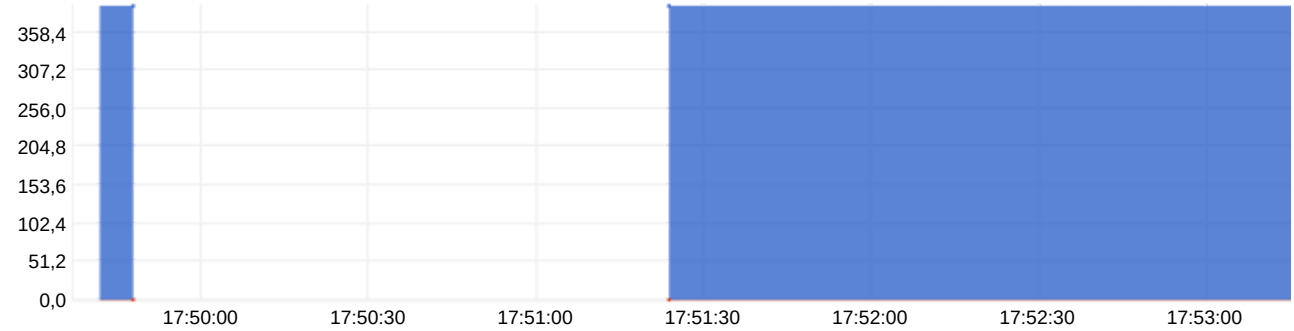
inodes сб, 24 апр. 2021 г. | 17:54:36

avail 503 833 used 3

/run/netdata

Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.

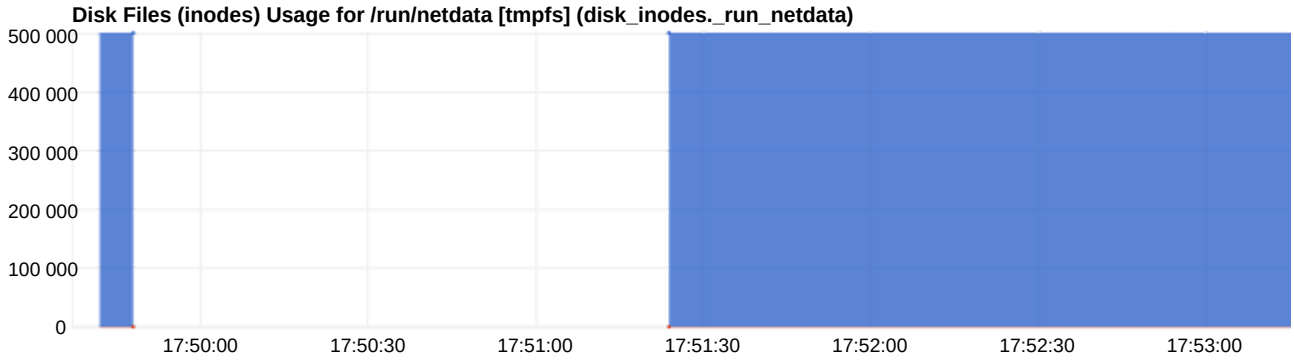
Disk Space Usage for /run/netdata [tmpfs] (disk_space._run_netdata)



MiB сб, 24 апр. 2021 г. | 17:54:36

avail 392,9 used 0,7

inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.

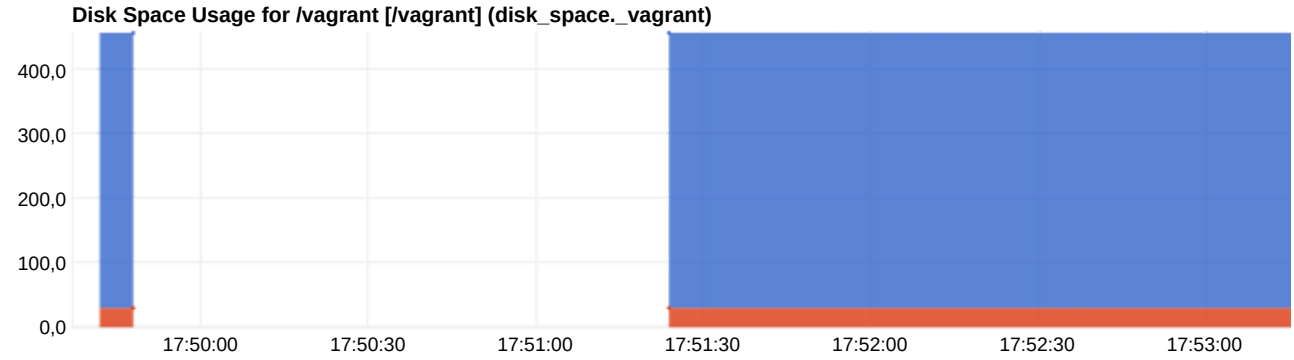


inodes сб, 24 апр. 2021 г. | 17:54:36

avail 503 150 used 686

/vagrant

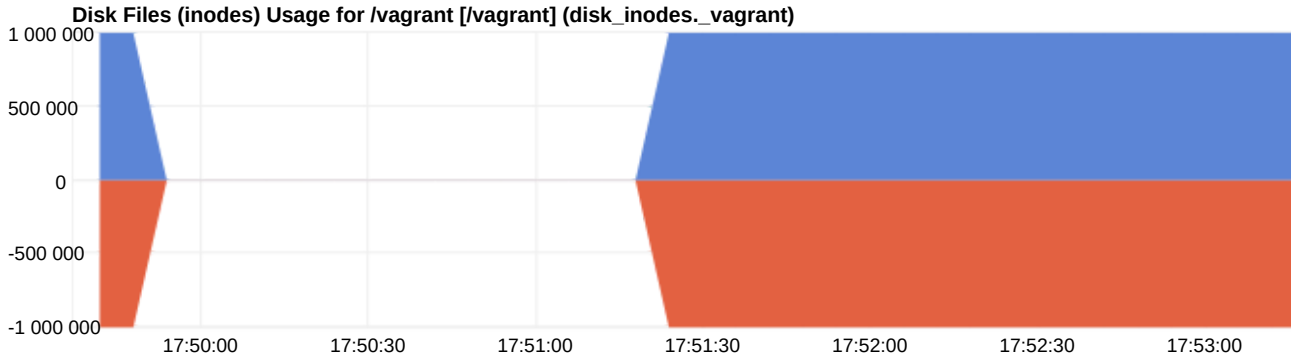
Disk space utilization. reserved for root is automatically reserved by the system to prevent the root user from getting out of space.



GiB сб, 24 апр. 2021 г. | 17:54:36

avail 427,4 used 30,0

inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available.



inodes сб, 24 апр. 2021 г. | 17:54:36

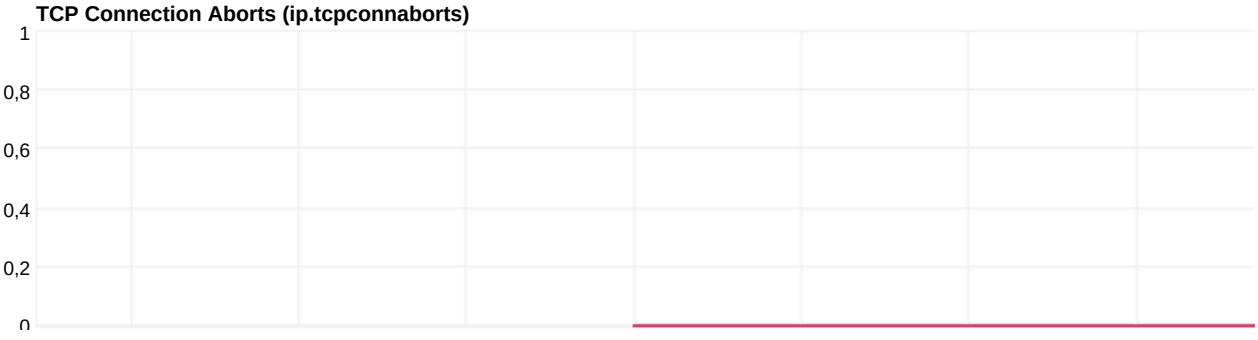
avail 1 000 000 used -999 000

Networking Stack

Metrics for the networking stack of the system. These metrics are collected from `/proc/net/netstat`, apply to both IPv4 and IPv6 traffic and are related to operation of the kernel networking stack.

tcp

TCP connection aborts. **baddata** (`TCPAbortOnData`) happens while the connection is on `FIN_WAIT1` and the kernel receives a packet with a sequence number beyond the last one for this connection - the kernel responds with `RST` (closes the connection). **userclosed** (`TCPAbortOnClose`) happens when the kernel receives data on an already closed connection and responds with `RST` . **nomemory** (`TCPAbortOnMemory` happens when there are too many orphaned sockets (not attached to an fd) and the kernel has to drop a connection - sometimes it will send an `RST` , sometimes it won't. **timeout** (`TCPAbortOnTimeout`) happens when a connection times out. **linger** (`TCPAbortOnLinger`) happens when the kernel killed a socket that was already closed by the application and lingered around for long enough. **failed** (`TCPAbortFailed`) happens when the kernel attempted to send an `RST` but failed because there was no memory available.

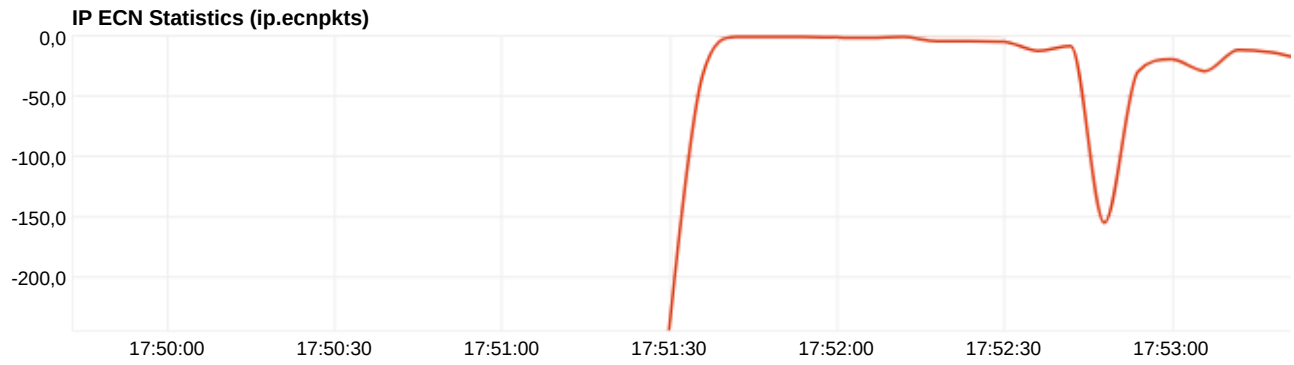


connections/s сб, 24 апр. 2021 г. | 17:54:42

baddata 0 userclosed 0 nomemory 0 timeout 0 linger 0 failed 0

ecn

Explicit Congestion Notification (ECN) (https://en.wikipedia.org/wiki/Explicit_Congestion_Notification) is a TCP extension that allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature that may be used between two ECN-enabled endpoints when the underlying network infrastructure also supports it.



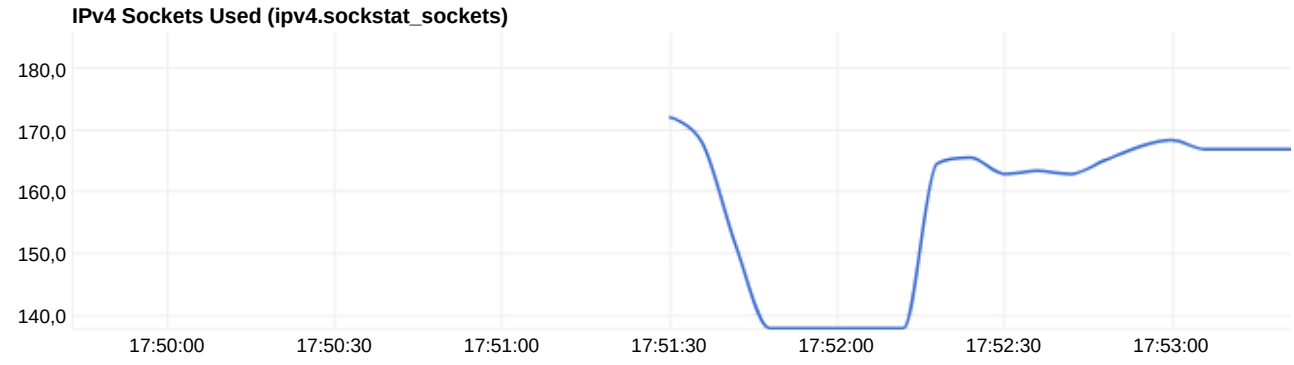
pps сб, 24 апр. 2021 г. | 17:54:42

NoECTP -15,7

IPv4 Networking

Metrics for the IPv4 stack of the system. Internet Protocol version 4 (IPv4) (https://en.wikipedia.org/wiki/IPv4) is the fourth version of the Internet Protocol (IP). It is one of the core protocols of standards-based internetworking methods in the Internet. IPv4 is a connectionless protocol for use on packet-switched networks. It operates on a best effort delivery model, in that it does not guarantee delivery, nor does it assure proper sequencing or avoidance of duplicate delivery. These aspects, including data integrity, are addressed by an upper layer transport protocol, such as the Transmission Control Protocol (TCP).

sockets

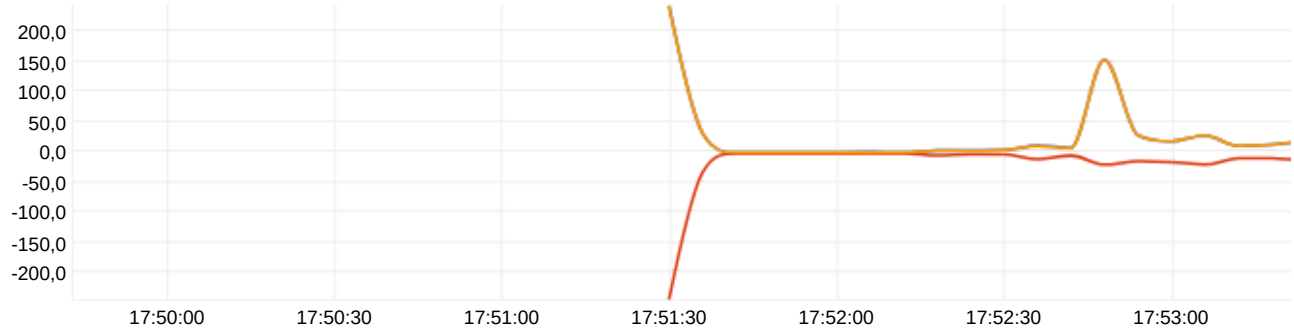


sockets сб, 24 апр. 2021 г. | 17:54:42

used 164,0

packets

IPv4 Packets (ipv4.packets)

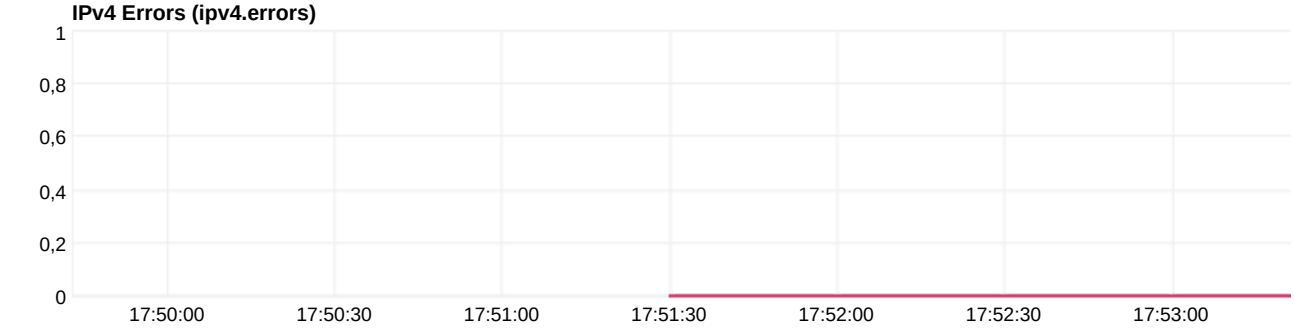


pps сб, 24 апр. 2021 г. | 17:54:42

received 15,7 sent -9,0 delivered 15,7

errors

IPv4 Errors (ipv4.errors)



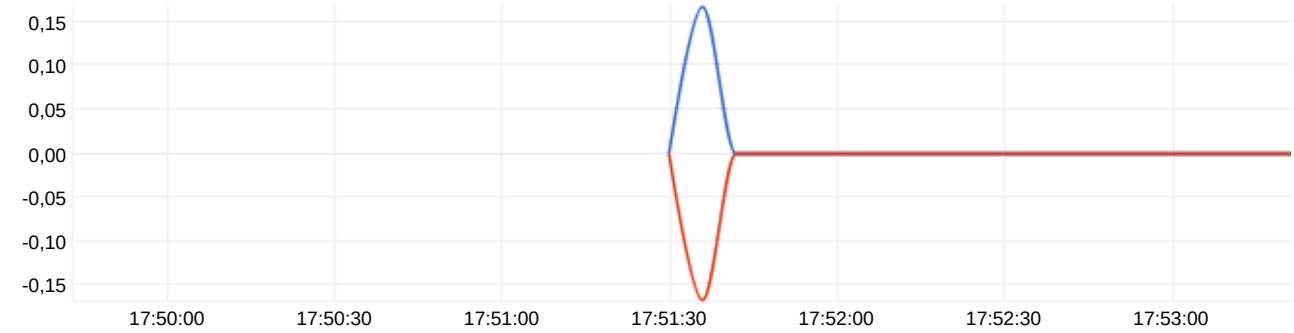
pps сб, 24 апр. 2021 г. | 17:54:42

InDiscards 0 OutDiscards 0 InHdrErrors 0 OutNoRoutes 0 InAddrErrors 0

InUnknownProtos 0

icmp

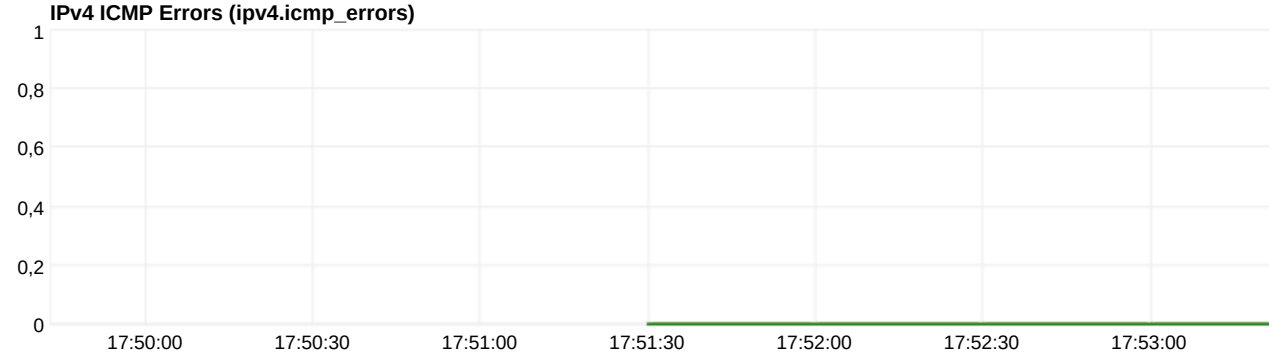
IPv4 ICMP Packets (ipv4.icmp)



packets/s сб, 24 апр. 2021 г. | 17:54:42

received 0,00 sent 0,00

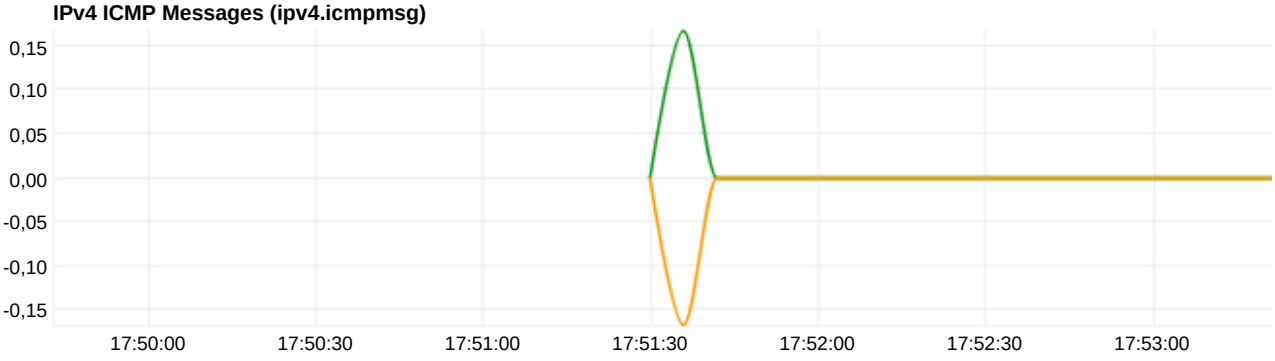
IPv4 ICMP Errors (ipv4.icmp_errors)



pps сб, 24 апр. 2021 г. | 17:54:42

InErrors 0 OutErrors 0 InCsumErrors 0

IPv4 ICMP Messages (ipv4.icmpmsg)



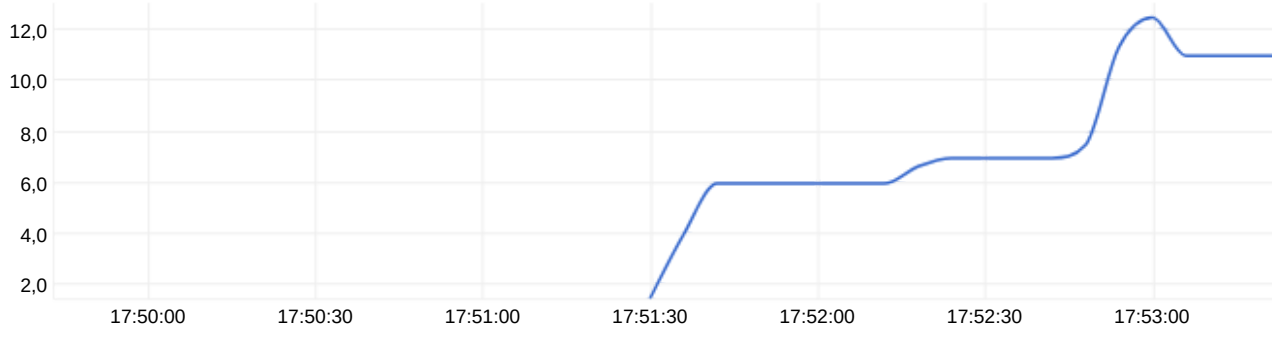
packets/s сб, 24 апр. 2021 г. | 17:54:42

InDestUnreachs 0,00 OutDestUnreachs 0,00

tcp

The number of established TCP connections (known as `CurrEstab`). This is a snapshot of the established connections at the time of measurement (i.e. a connection established and a connection disconnected within the same iteration will not affect this metric).

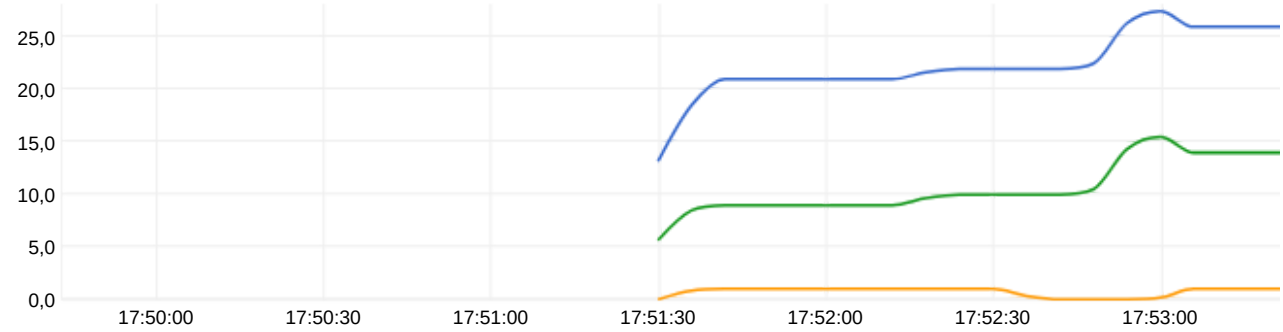
IPv4 TCP Connections (ipv4.tcpsock)



active connections сб, 24 апр. 2021 г. | 17:54:42

connections 8,0

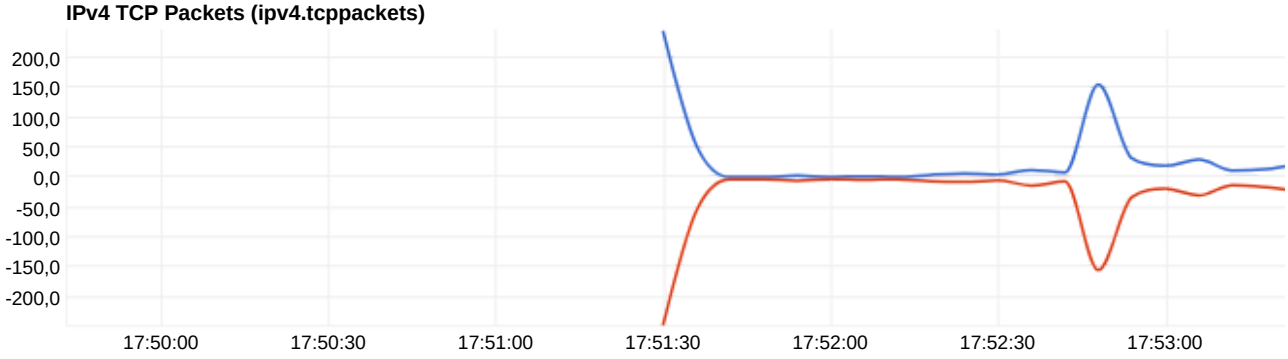
IPv4 TCP Sockets (ipv4.sockstat_tcp_sockets)



sockets сб, 24 апр. 2021 г. | 17:54:42

alloc 23,0 inuse 11,0 timewait 3,0

IPv4 TCP Packets (ipv4.tcppackets)

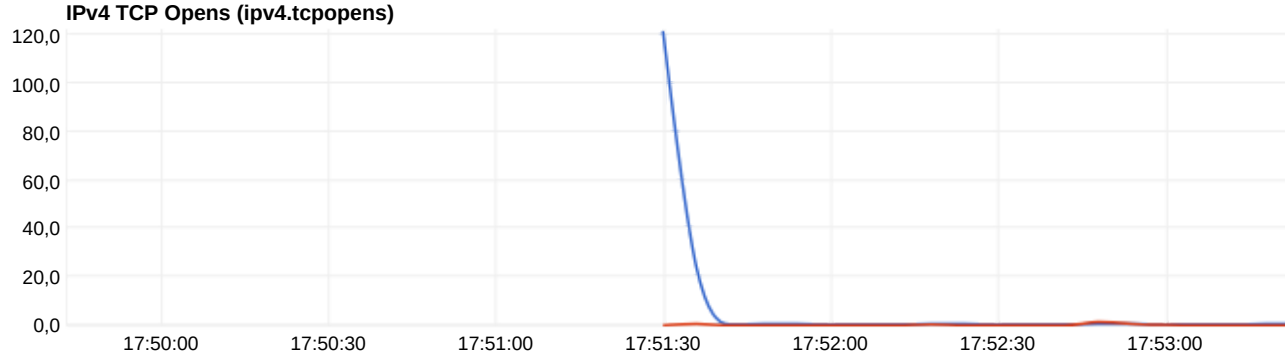


pps сб, 24 апр. 2021 г. | 17:54:42

received 17,3 sent -9,8

active or `ActiveOpens` is the number of outgoing TCP **connections attempted** by this host.
passive or `PassiveOpens` is the number of incoming TCP **connections accepted** by this host.

IPv4 TCP Opens (ipv4.tcpopens)



connections/s сб, 24 апр. 2021 г. | 17:54:42

active 0,3 passive 0,0

`InErrs` is the number of TCP segments received in error (including header too small, checksum errors, sequence errors, bad packets - for both IPv4 and IPv6). `InCsumErrors` is the number of TCP segments received with checksum errors (for both IPv4 and IPv6). `RetransSegs` is the number of TCP segments retransmitted.

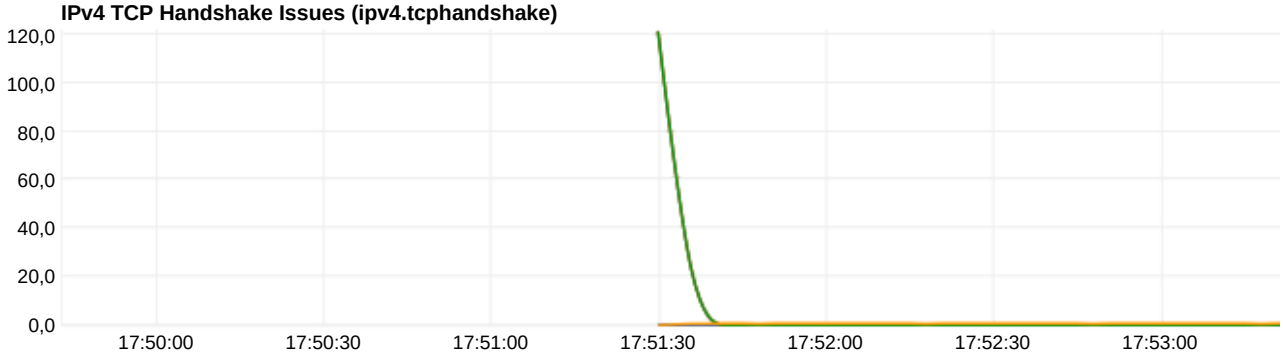
IPv4 TCP Errors (ipv4.tcperrors)

packets/s сб, 24 апр. 2021 г. | 17:54:42

RetransSegs -0,83

`EstabResets` is the number of established connections resets (i.e. connections that made a direct transition from `ESTABLISHED` or `CLOSE_WAIT` to `CLOSED`). `OutRsts` is the number of TCP segments sent, with the `RST` flag set (for both IPv4 and IPv6). `AttemptFails` is the number of times TCP connections made a direct transition from either `SYN_SENT` or `SYN_RECV` to `CLOSED`, plus the number of times TCP connections made a direct transition from the `SYN_RECV` to `LISTEN`. `TCPSynRetrans` shows retries for new outbound TCP connections, which can indicate general connectivity issues or backlog on the remote host.

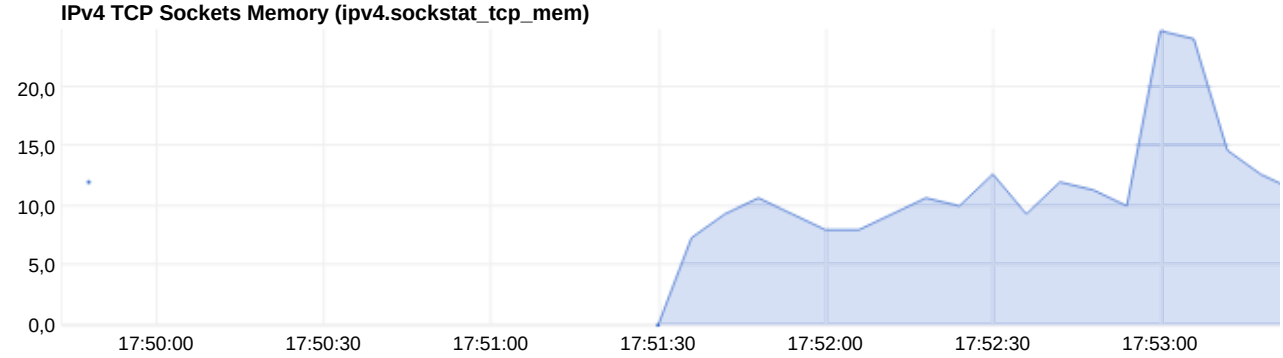
IPv4 TCP Handshake Issues (ipv4.tcphandshake)



events/s сб, 24 апр. 2021 г. | 17:54:42

EstabResets 0,0 OutRsts 0,0 AttemptFails 0,0 SynRetrans 0,8

IPv4 TCP Sockets Memory (ipv4.sockstat_tcp_mem)

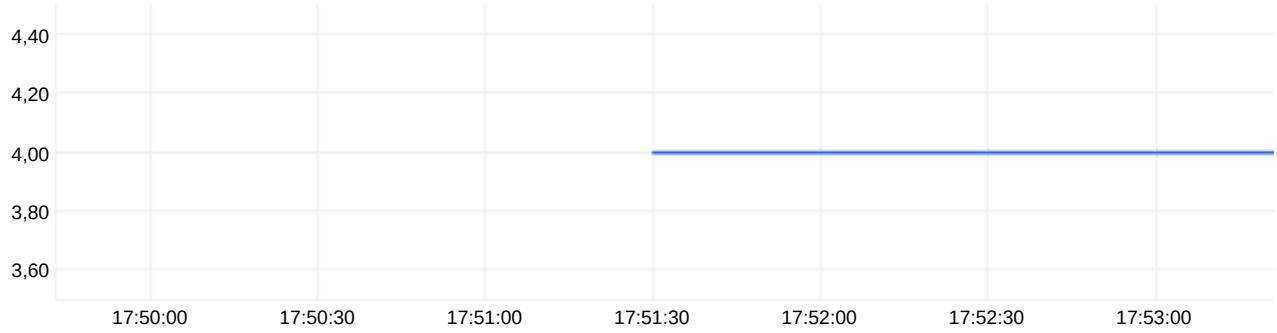


KiB сб, 24 апр. 2021 г. | 17:54:42

mem 12,7

udp

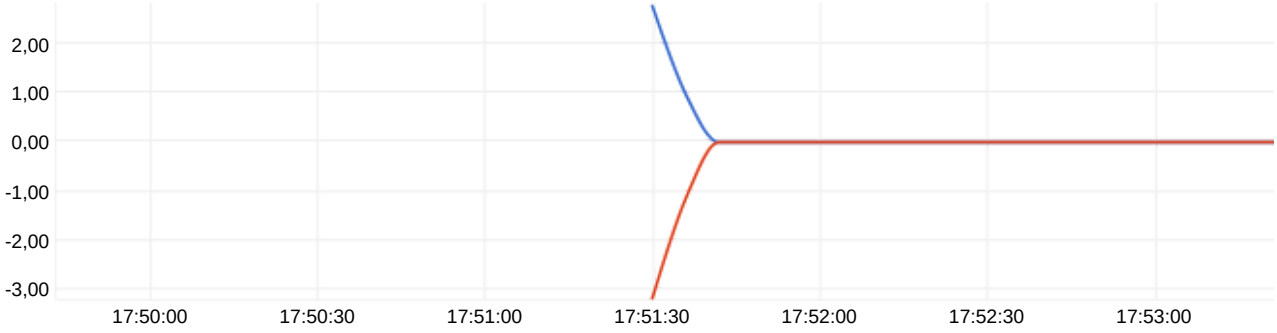
IPv4 UDP Sockets (ipv4.sockstat_udp_sockets)



sockets сб, 24 апр. 2021 г. | 17:54:42

inuse 4,00

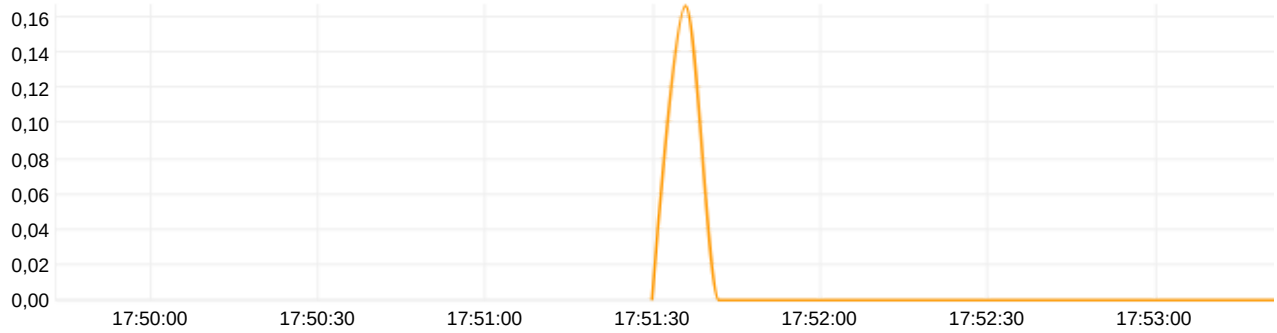
IPv4 UDP Packets (ipv4.udppackets)



pps сб, 24 апр. 2021 г. | 17:54:42

received 0,00 sent 0,00

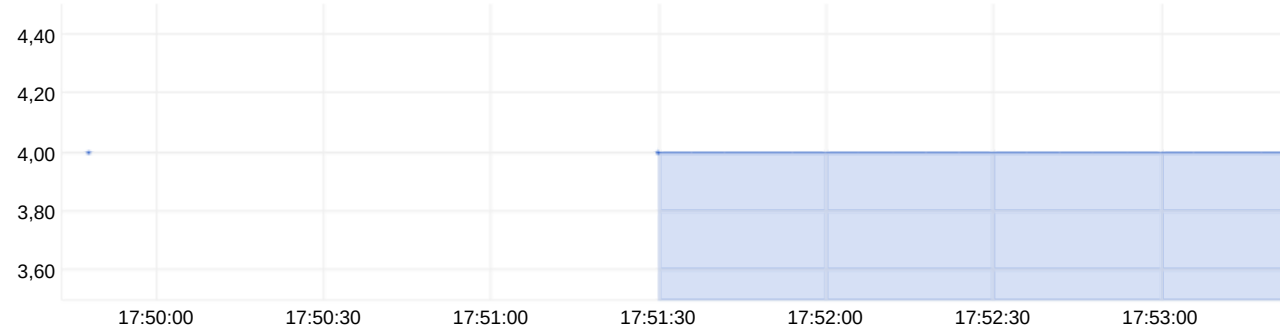
IPv4 UDP Errors (ipv4.udperrors)



events/s сб, 24 апр. 2021 г. | 17:54:42

NoPorts 0,00

IPv4 UDP Sockets Memory (ipv4.sockstat_udp_mem)



KiB сб, 24 апр. 2021 г. | 17:54:42

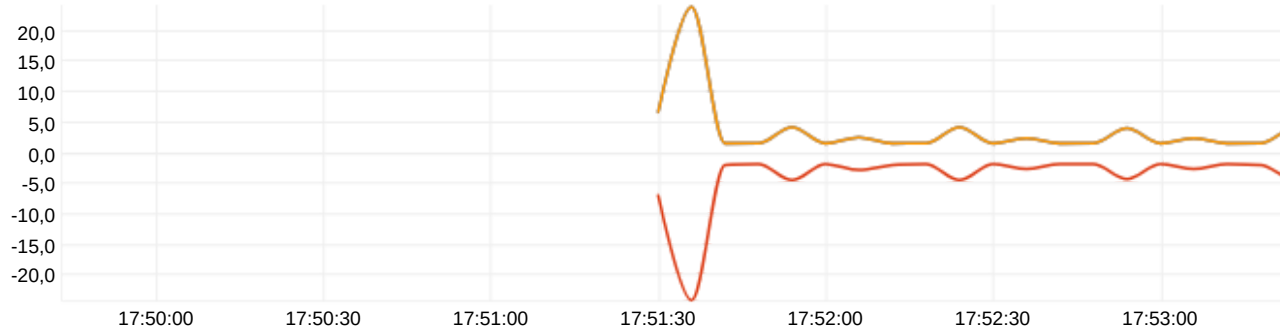
mem 4,00

IPv6 Networking

Metrics for the IPv6 stack of the system. Internet Protocol version 6 (IPv6) (https://en.wikipedia.org/wiki/IPv6) is the most recent version of the Internet Protocol (IP), the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet. IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion. IPv6 is intended to replace IPv4.

packets

IPv6 Packets (ipv6.packets)



pps сб, 24 апр. 2021 г. | 17:54:42

received 1,7 sent -1,7 delivers 1,7

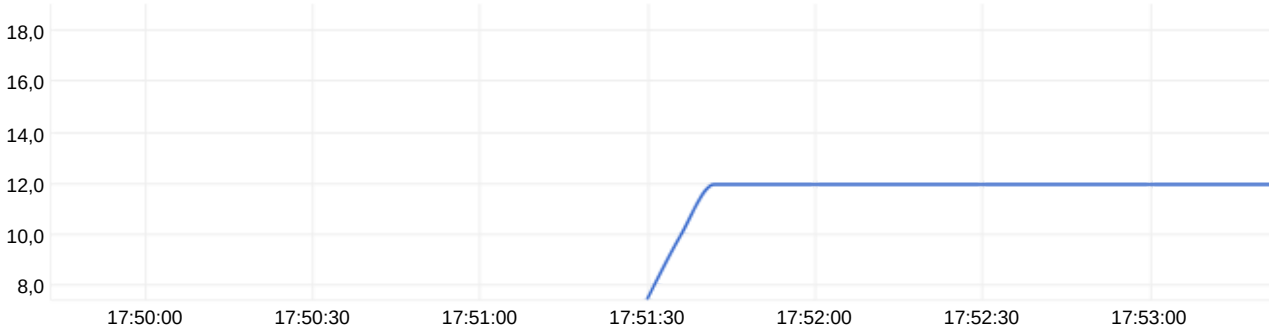
IPv6 ECT Packets (ipv6.ect)

pps сб, 24 апр. 2021 г. | 17:54:42

InNoECTPkts 1,7

tcp6

IPv6 TCP Sockets (ipv6.sockstat6_tcp_sockets)

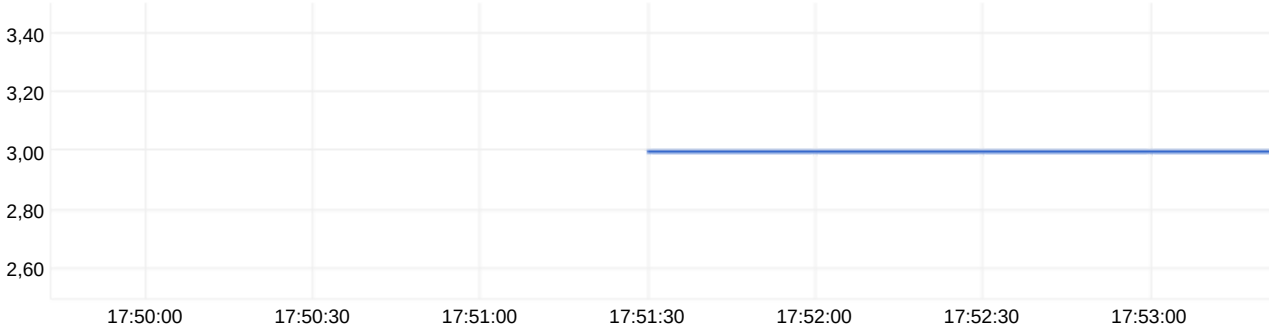


sockets сб, 24 апр. 2021 г. | 17:54:42

inuse 12,0

udp6

IPv6 UDP Sockets (ipv6.sockstat6_udp_sockets)



sockets сб, 24 апр. 2021 г. | 17:54:42

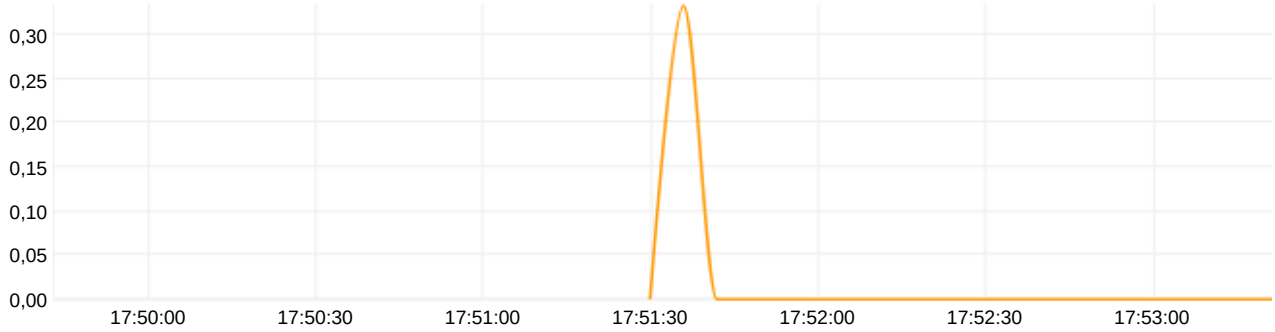
inuse 3,00

IPv6 UDP Packets (ipv6.udppackets)

packets/s сб, 24 апр. 2021 г. | 17:54:42

sent 0,00

IPv6 UDP Errors (ipv6.udperrors)

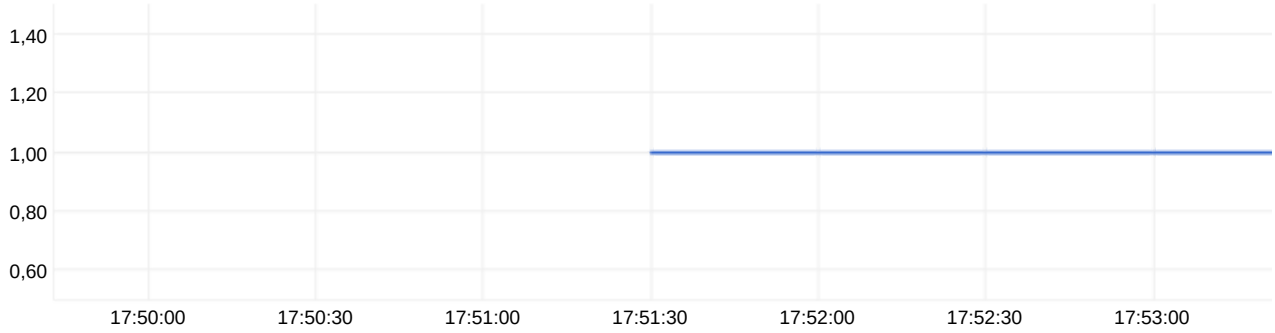


events/s сб, 24 апр. 2021 г. | 17:54:42

NoPorts 0,00

raw6

IPv6 RAW Sockets (ipv6.sockstat6_raw_sockets)



sockets сб, 24 апр. 2021 г. | 17:54:42

inuse 1,00

multicast6

IPv6 Multicast Bandwidth (ipv6.mcast)

bits/s сб, 24 апр. 2021 г. | 17:54:42

sent 0,0

IPv6 Multicast Packets (ipv6.mcastpkts)

packets/s сб, 24 апр. 2021 г. | 17:54:42

sent 0,00

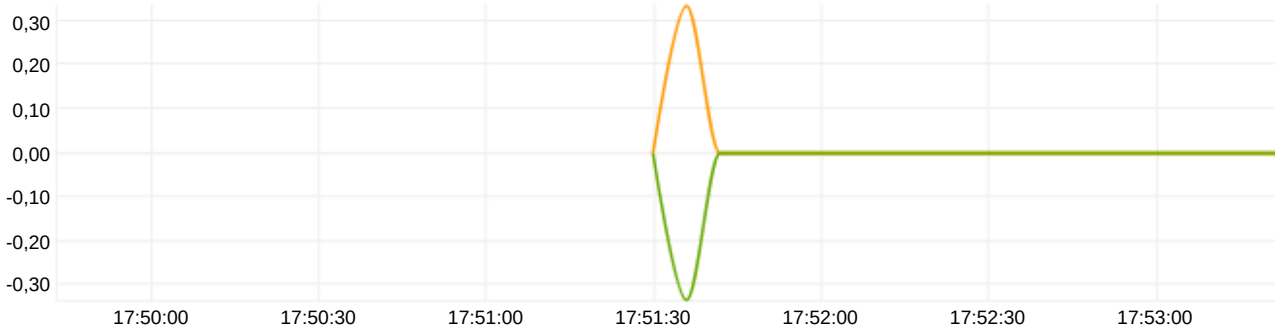
icmp6

IPv6 ICMP Messages (ipv6.icmp)

messages/s сб, 24 апр. 2021 г. | 17:54:42

received 0,00 sent 0,00

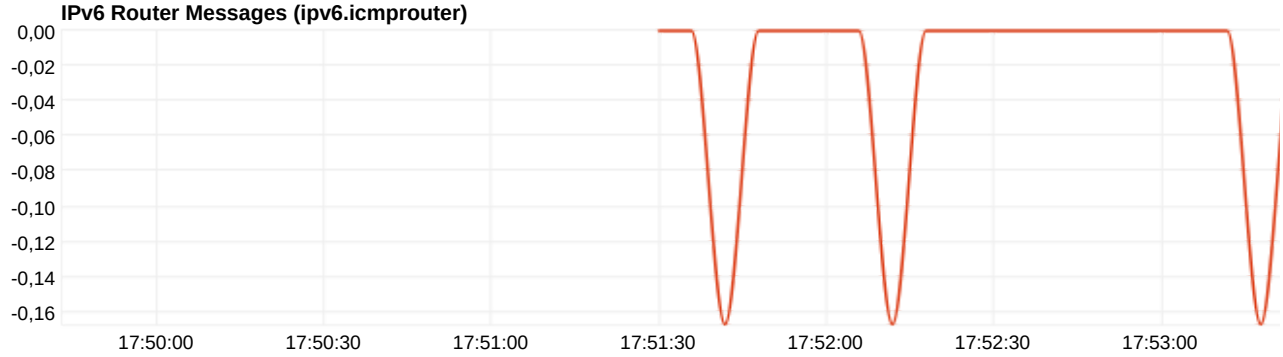
IPv6 ICMP Errors (ipv6.icmperrors)



errors/s сб, 24 апр. 2021 г. | 17:54:42

InDestUnreachs 0,00 OutDestUnreachs 0,00

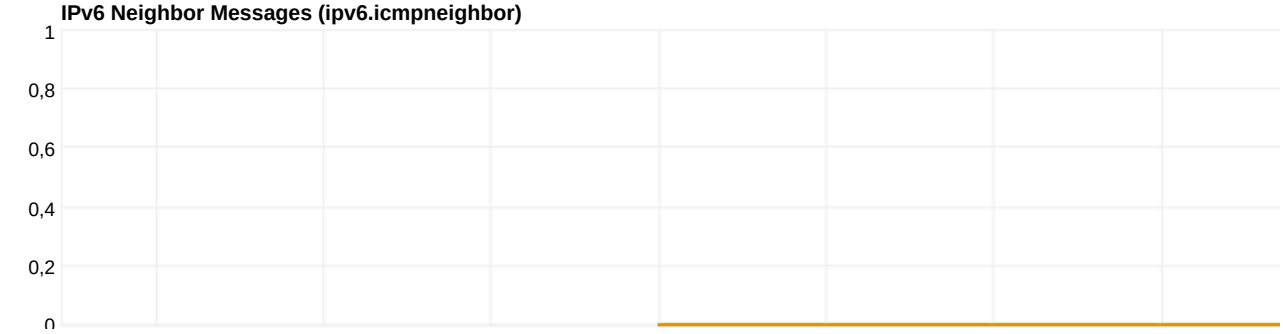
IPv6 Router Messages (ipv6.icmprouter)



messages/s сб, 24 апр. 2021 г. | 17:54:42

OutSolicits 0,00

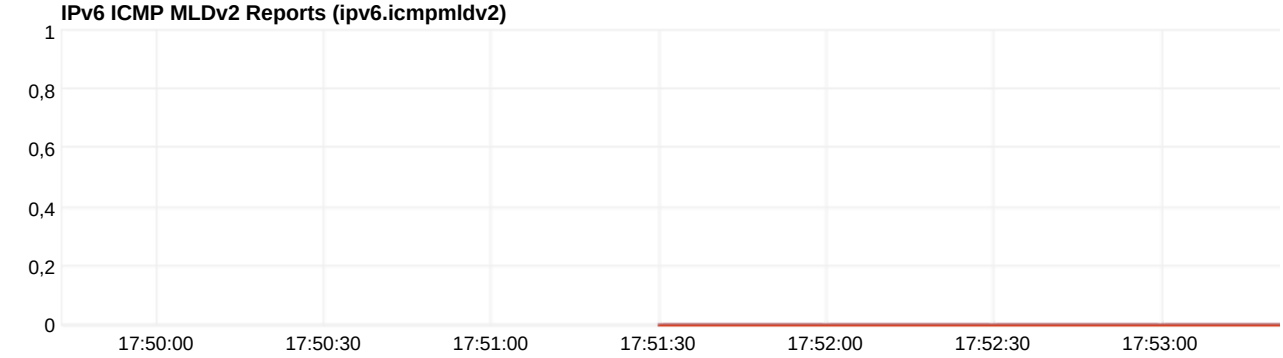
IPv6 Neighbor Messages (ipv6.icmpneighbor)



messages/s сб, 24 апр. 2021 г. | 17:54:42

InSolicits 0 OutSolicits 0 InAdvertisements 0 OutAdvertisements 0

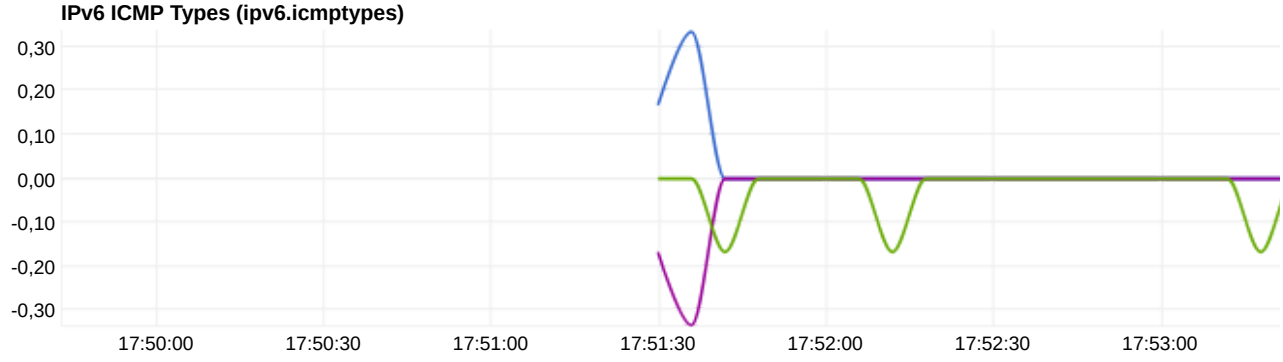
IPv6 ICMP MLDv2 Reports (ipv6.icmpmldv2)



reports/s сб, 24 апр. 2021 г. | 17:54:42

received 0 sent 0

IPv6 ICMP Types (ipv6.icmptypes)



messages/s сб, 24 апр. 2021 г. | 17:54:42

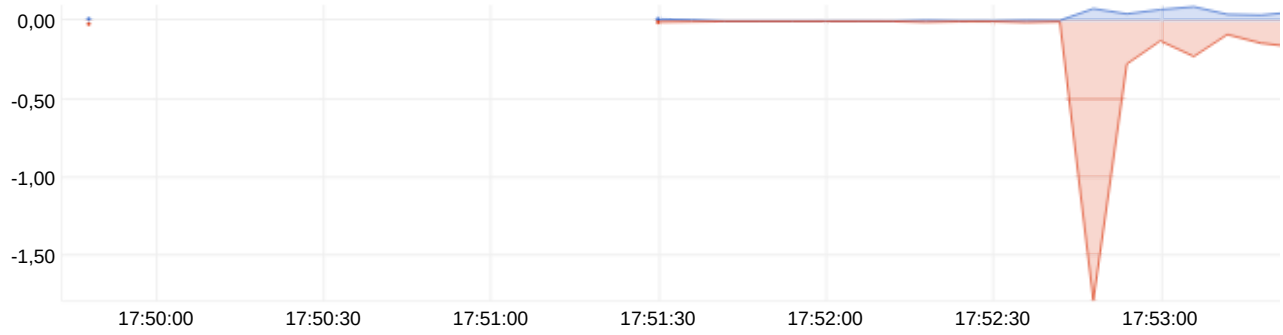
InType1 0,00 OutType1 0,00 OutType133 0,00

Network Interfaces

Performance metrics for network interfaces.

eth0

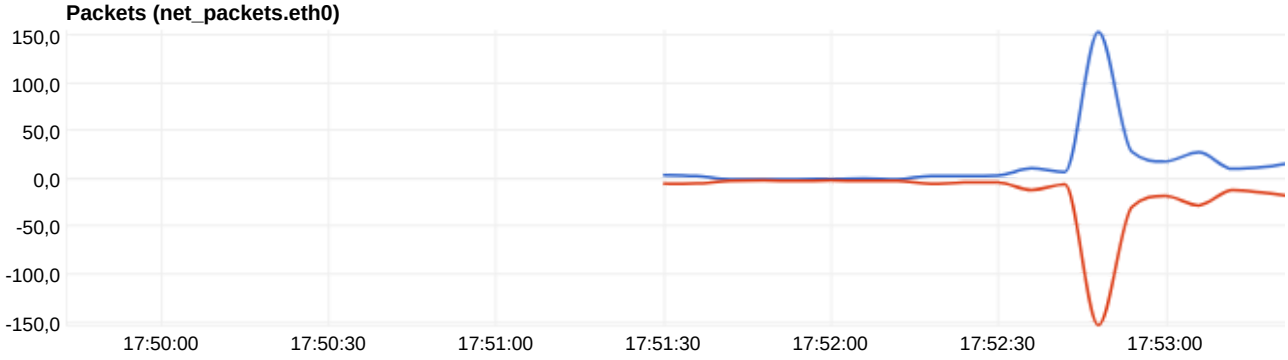
Bandwidth (net.eth0)



megabits/s сб, 24 апр. 2021 г. | 17:54:42

received 0,07 sent -0,06

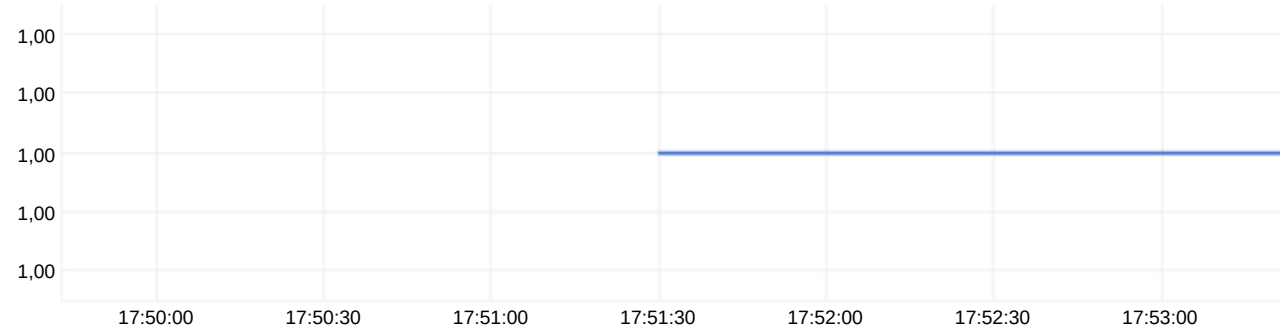
Packets (net_packets.eth0)



pps сб, 24 апр. 2021 г. | 17:54:42

received 15,7 sent -9,0

Interface Speed (net_speed.eth0)

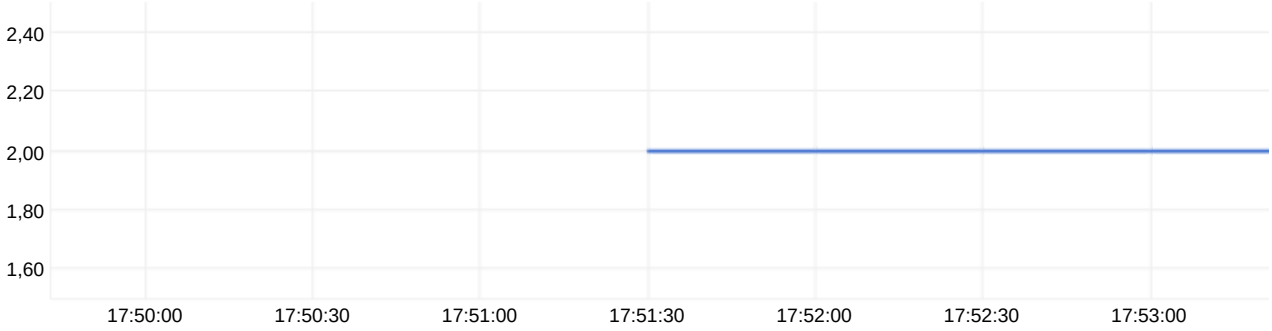


gigabits/s сб, 24 апр. 2021 г. | 17:54:42

speed 1,00

State map: 0 - unknown, 1 - half duplex, 2 - full duplex

Interface Duplex State (net_duplex.eth0)

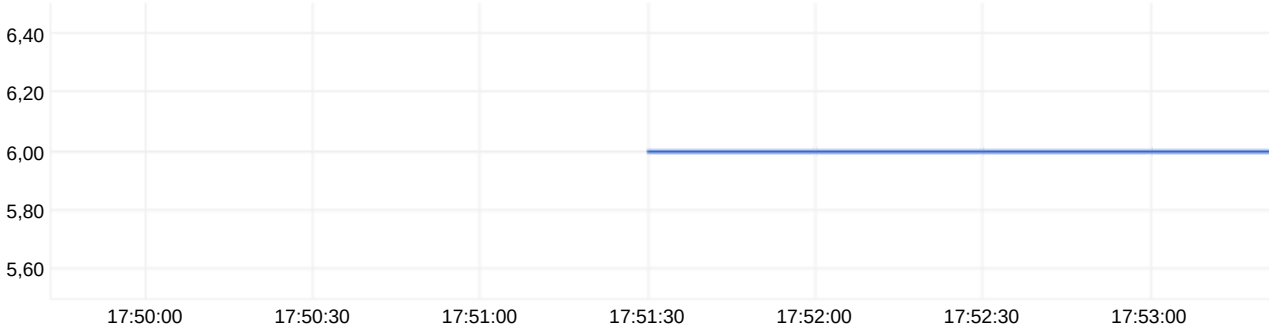


state сб, 24 апр. 2021 г. | 17:54:42

duplex 2,00

State map: 0 - unknown, 1 - notpresent, 2 - down, 3 - lowerlayerdown, 4 - testing, 5 - dormant, 6 - up

Interface Operational State (net_operstate.eth0)

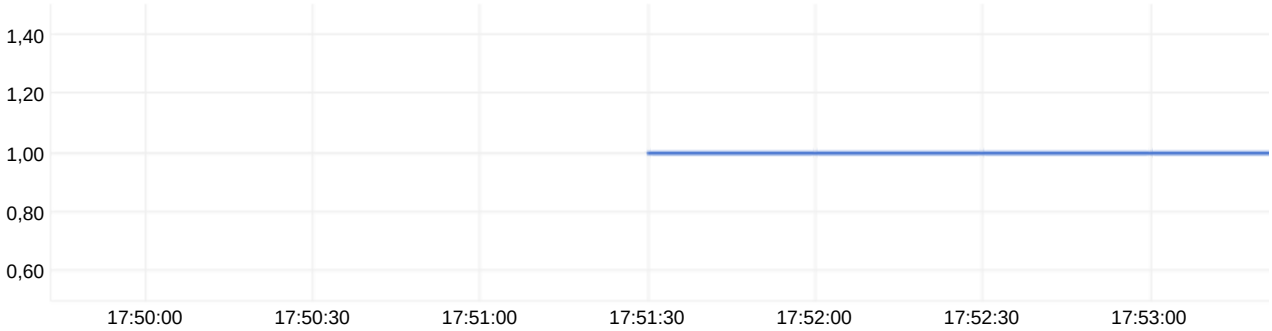


state сб, 24 апр. 2021 г. | 17:54:42

state 6,00

State map: 0 - down, 1 - up

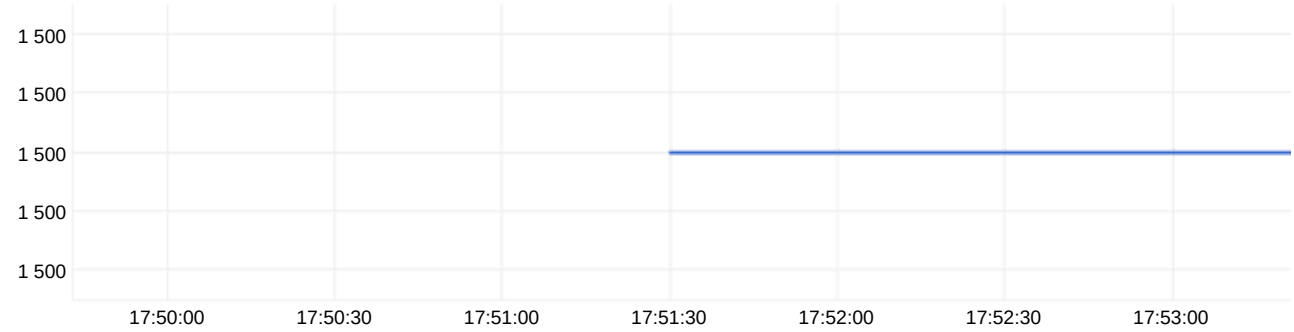
Inteface Physical Link State (net_carrier.eth0)



state сб, 24 апр. 2021 г. | 17:54:42

carrier 1,00

Interface MTU (net_mtu.eth0)



octets сб, 24 апр. 2021 г. | 17:54:42

mtu 1 500

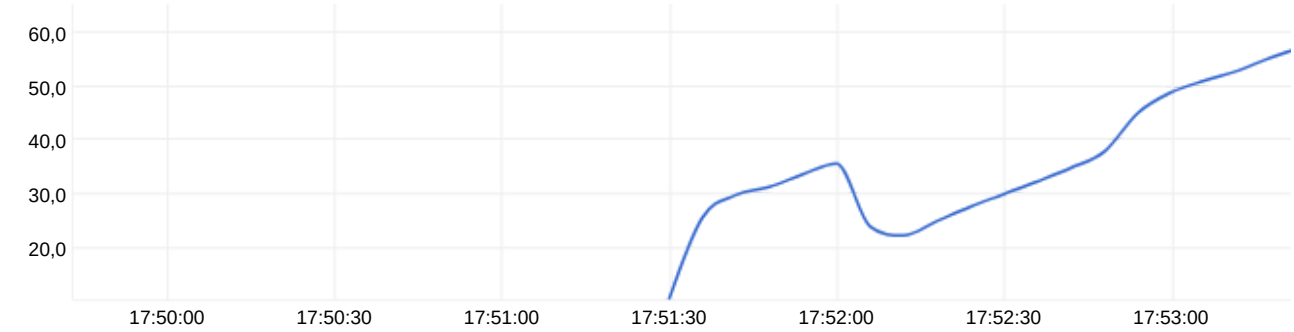
Firewall (netfilter)

Performance metrics of the netfilter components.

connection tracker

Netfilter Connection Tracker performance metrics. The connection tracker keeps track of all connections of the machine, inbound and outbound. It works by keeping a database with all open connections, tracking network and address translation and connection expectations.

Connection Tracker Connections (netfilter.conntrack_sockets)



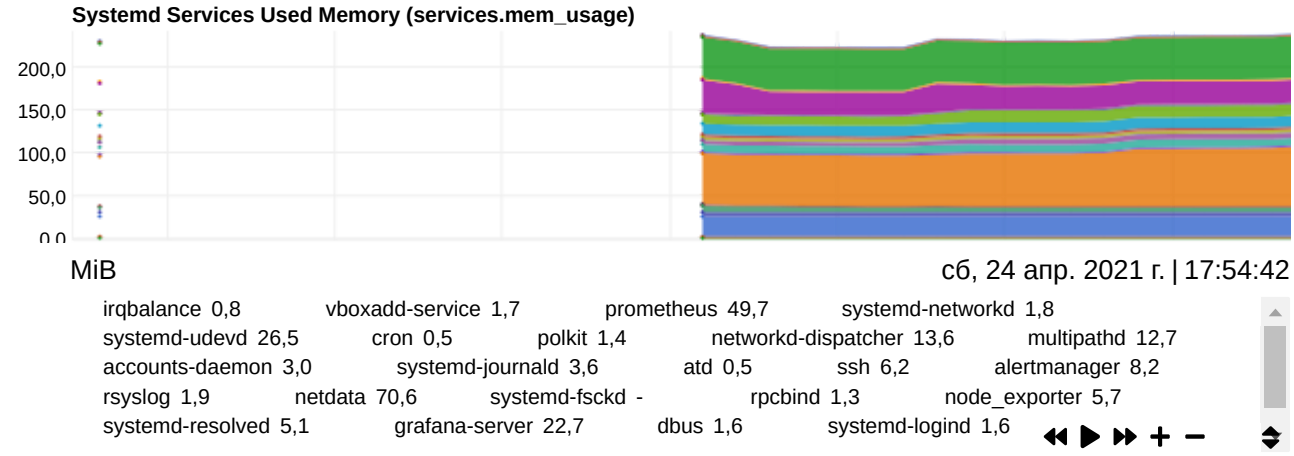
active connections сб, 24 апр. 2021 г. | 17:54:42

connections 61,2

systemd Services

Resources utilization of systemd services. netdata monitors all systemd services via CGROUPS (the resources accounting used by containers).

mem

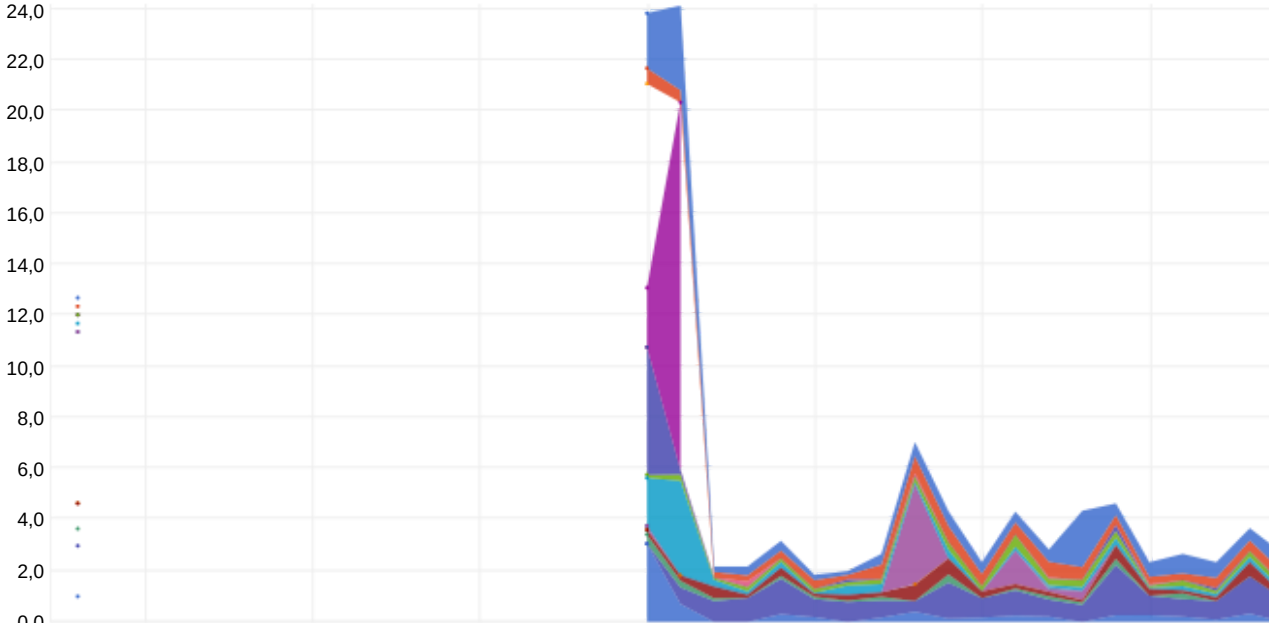


Applications

Per application statistics are collected using netdata's `apps.plugin` . This plugin walks through all processes and aggregates statistics for applications of interest, defined in `/etc/netdata/apps_groups.conf` , which can be edited by running `$ /etc/netdata/edit-config apps_groups.conf` (the default is here (https://github.com/netdata/netdata/blob/master/collectors/apps.plugin/apps_groups.conf)). The plugin internally builds a process tree (much like `ps fax` does), and groups processes together (evaluating both child and parent processes) so that the result is always a chart with a predefined set of dimensions (of course, only application groups found running are reported). The reported values are compatible with `top` , although the netdata plugin counts also the resources of exited children (unlike `top` which shows only the resources of the currently running processes). So for processes like shell scripts, the reported values include the resources used by the commands these scripts run within each timeframe.

cpu

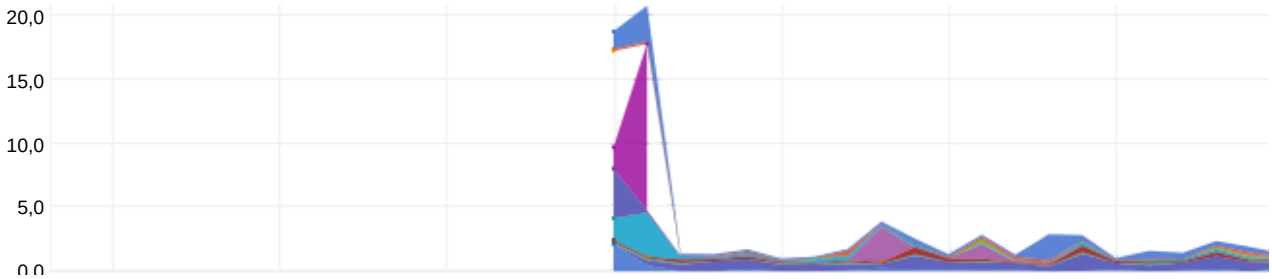
Apps CPU Time (100% = 1 core) (apps.cpu)



percentage сб, 24 апр. 2021 г. | 17:54:42

netdata 0,3 apps.plugin 0,2 node.d.plugin - python.d.plugin - tc-qos-helper 0,1
go.d.plugin 0,0 ebpf.plugin 0,1 timedb 0,0 VMs 0,0 ssh 0,0 X 0,0
system 0,4 kernel 0,0 node 0,7 other 0,1

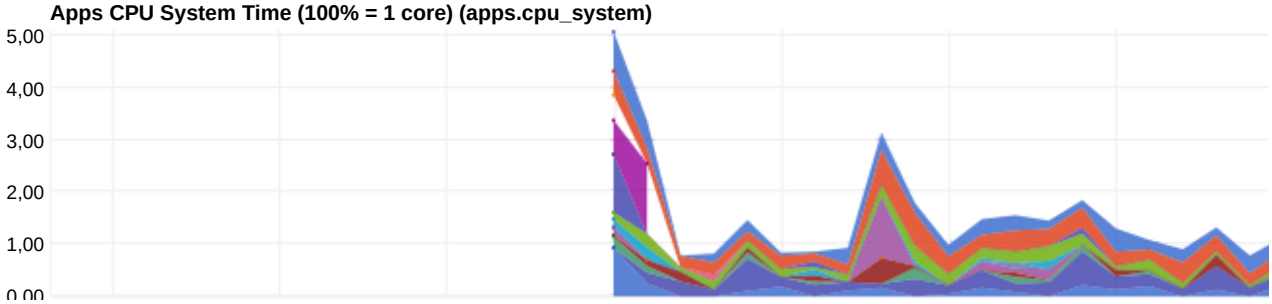
Apps CPU User Time (100% = 1 core) (apps.cpu_user)



percentage сб, 24 апр. 2021 г. | 17:54:48

netdata 0,6 apps.plugin 0,0 node.d.plugin - python.d.plugin - tc-qos-helper 0,0
go.d.plugin 0,0 ebpf.plugin 0,0 timedb 0,0 ssh 0,0 X 0,0 system 0,2
kernel 0,0 node 0,9 other 0,2

Apps CPU System Time (100% = 1 core) (apps.cpu_system)

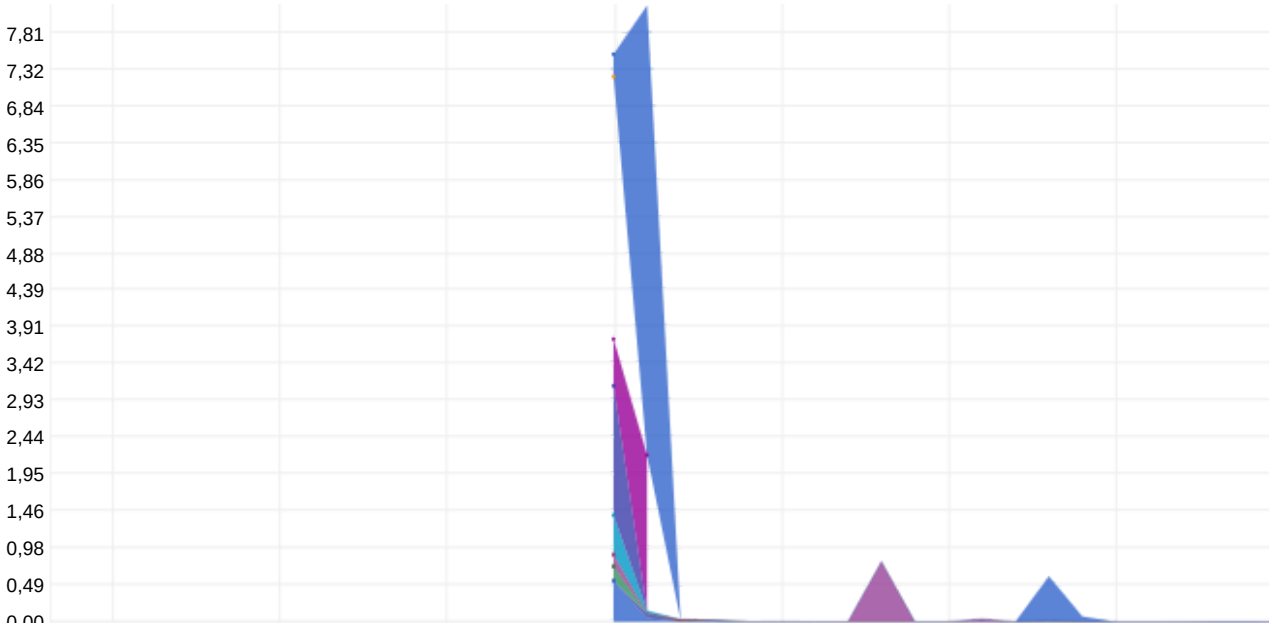


percentage сб, 24 апр. 2021 г. | 17:54:48

netdata 0,00 apps.plugin 0,58 node.d.plugin - python.d.plugin - tc-qos-helper 0,00
go.d.plugin 0,00 ebpf.plugin 0,20 timedb 0,18 VMs 0,00 ssh 0,00 X 0,00
system 0,00 kernel 0,00 node 0,38 other 0,00

disk

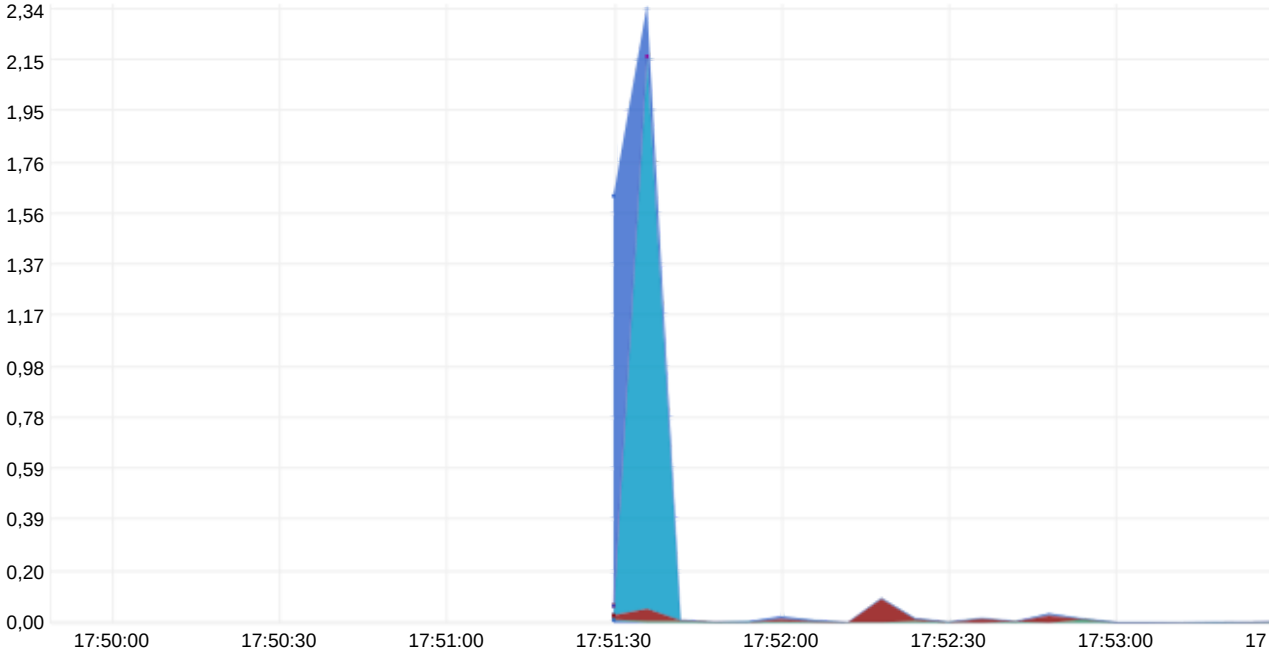
Apps Disk Reads (apps.preads)



MiB/s сб, 24 апр. 2021 г. | 17:54:48

netdata 0,00 node.d.plugin - python.d.plugin - go.d.plugin 0,00 timedb 0,00
logs 0,00 ssh 0,00 system 0,00 kernel 0,00 node 0,00 other 0,00

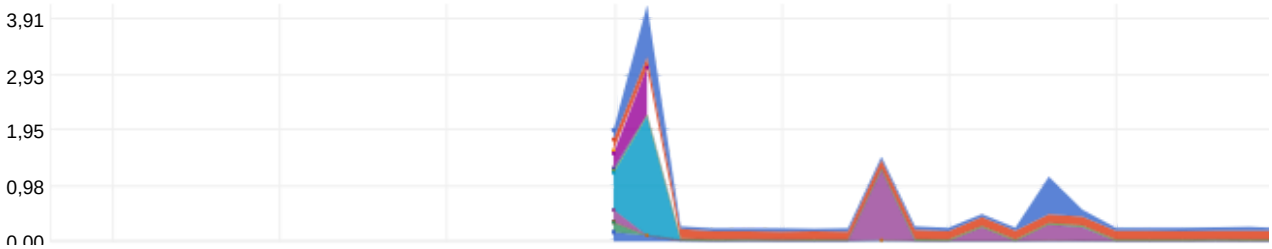
Apps Disk Writes (apps.pwrites)



MiB/s сб, 24 апр. 2021 г. | 17:54:48

netdata 0,00 node.d.plugin - python.d.plugin - go.d.plugin 0,00 timedb 0,00
logs 0,00 ssh 0,00 system 0,00 kernel 0,00 other 0,00

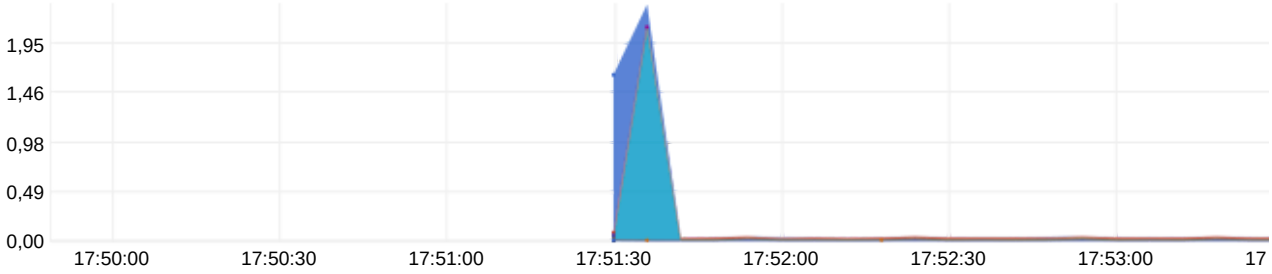
Apps Disk Logical Reads (apps.lreads)



MiB/s сб, 24 апр. 2021 г. | 17:54:48

netdata 0,06 apps.plugin 0,15 node.d.plugin - python.d.plugin - tc-qos-helper 0,00
go.d.plugin 0,00 ebpf.plugin 0,02 timedb 0,00 logs 0,00 VMs 0,00 ssh 0,00
dns 0,00 X 0,00 system 0,00 kernel 0,00 node 0,00 other 0,00

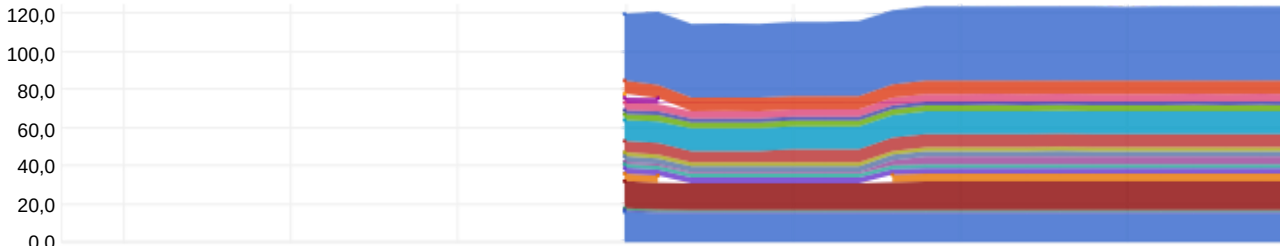
Apps I/O Logical Writes (apps.lwrites)



MiB/s сб, 24 апр. 2021 г. | 17:54:48

netdata 0,00 apps.plugin 0,02 node.d.plugin - python.d.plugin - tc-qos-helper 0,00
go.d.plugin 0,00 ebpf.plugin 0,01 timedb 0,00 logs 0,00 ssh 0,00 dns 0,00
X 0,00 system 0,00 kernel 0,00 node 0,02 other 0,00

Apps Open Files (apps.files)



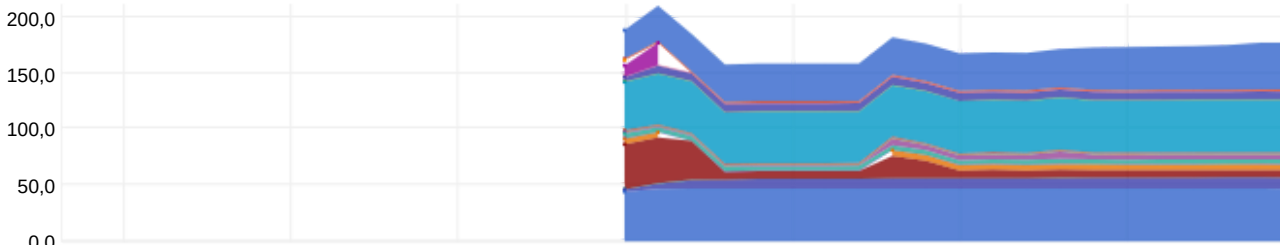
open files сб, 24 апр. 2021 г. | 17:54:48

netdata 39,0 apps.plugin 7,0 node.d.plugin - python.d.plugin - tc-qos-helper 4,0
go.d.plugin 2,0 ebpf.plugin 3,0 timedb 12,0 logs 7,0 nfs 2,0 VMs 3,0
ssh 4,0 dns 2,0 cron 3,0 X 4,0 system 15,0 kernel 1,0 node 1,0
other 15,0

mem

Real memory (RAM) used by applications. This does not include shared memory.

Apps Real Memory (w/o shared) (apps.mem)

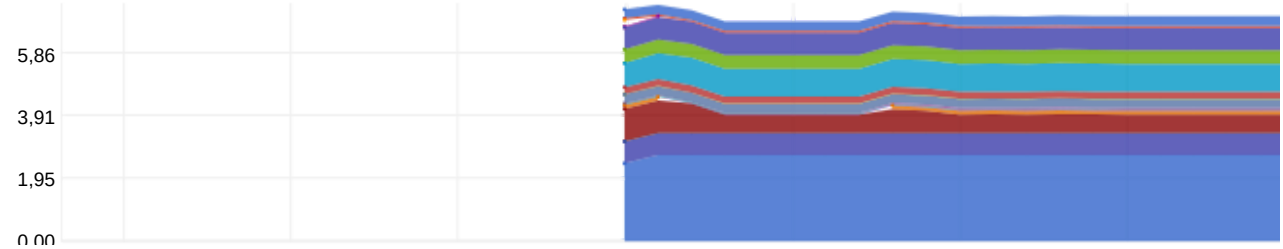


MiB сб, 24 апр. 2021 г. | 17:54:48

netdata 42,2 apps.plugin 1,4 node.d.plugin - python.d.plugin - tc-qos-helper 0,5
go.d.plugin 7,1 ebpf.plugin 0,4 timedb 47,2 logs 1,0 nfs 0,4 VMs 0,4
ssh 4,4 dns 4,1 cron 0,4 X 4,8 system 6,8 kernel 0,1 node 10,3
other 46,4

Virtual memory allocated by applications. Please check this article (https://github.com/netdata/netdata/tree/master/daemon#virtual-memory) for more information.

Apps Virtual Memory Size (apps.vmem)



GiB сб, 24 апр. 2021 г. | 17:54:48

netdata 0,31 apps.plugin 0,05 node.d.plugin - python.d.plugin - tc-qos-helper 0,00
go.d.plugin 0,69 ebpf.plugin 0,43 timedb 0,87 logs 0,21 nfs 0,01 VMs 0,28
ssh 0,05 dns 0,02 cron 0,01 X 0,12 system 0,58 kernel 0,00 node 0,69
other 2,68

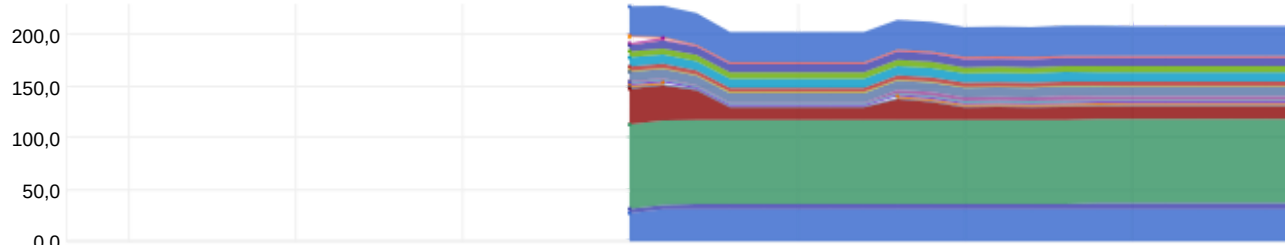
Apps Minor Page Faults (apps.minor_faults)

page faults/s сб, 24 апр. 2021 г. | 17:54:48

netdata 30 apps.plugin 0 node.d.plugin - python.d.plugin - tc-qos-helper 0
go.d.plugin 0 ebpf.plugin 0 timedb 0 logs 0 VMs 0 ssh 0
dns 0 X 0 system 0 kernel 0 node 3 other 0

processes

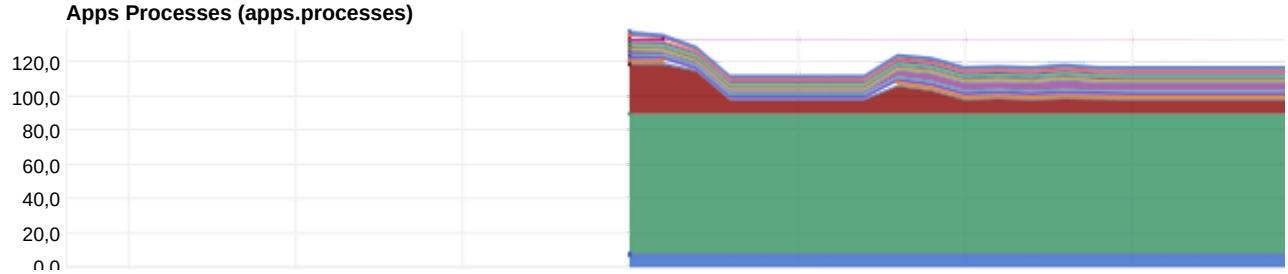
Apps Threads (apps.threads)



threads сб, 24 апр. 2021 г. | 17:54:48

netdata 29,0 apps.plugin 1,0 node.d.plugin - python.d.plugin - tc-qos-helper 1,0
go.d.plugin 8,0 ebpf.plugin 6,0 timedb 9,0 logs 4,0 nfs 1,0 VMs 9,0
ssh 4,0 dns 1,0 cron 2,0 X 2,0 ksmd 1,0 system 12,0 kernel 82,0
node 6,0 other 32,0

Apps Processes (apps.processes)



processes сб, 24 апр. 2021 г. | 17:54:48

netdata 2,0 apps.plugin 1,0 node.d.plugin - python.d.plugin - tc-qos-helper 1,0
go.d.plugin 1,0 ebpf.plugin 1,0 timedb 1,0 logs 1,0 nfs 1,0 VMs 1,0
ssh 4,0 dns 1,0 cron 2,0 X 2,0 ksmd 1,0 system 8,0 kernel 82,0
node 1,0 other 7,0

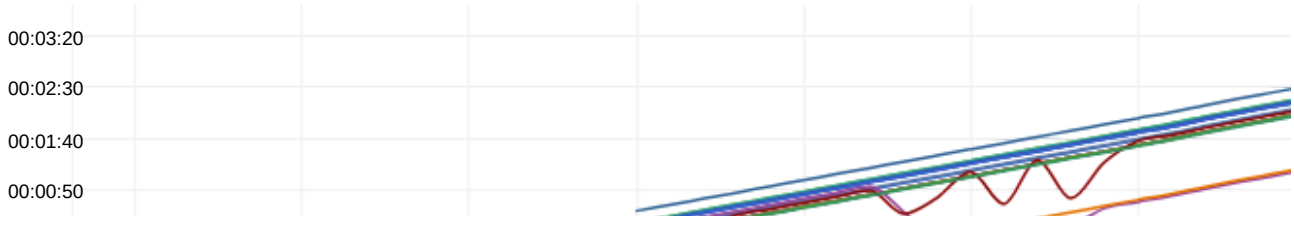
Carried over process group uptime since the Netdata restart. The period of time within which at least one process in the group was running.

Apps Carried Over Uptime (apps.uptime)

time сб, 24 апр. 2021 г. | 17:54:48

netdata 00:03:36 apps.plugin 00:03:25 node.d.plugin - python.d.plugin -
tc-qos-helper 00:03:25 go.d.plugin 00:03:25 ebpf.plugin 00:03:25 timedb 00:03:37 logs 00:03:38
nfs 00:03:40 VMs 00:03:31 ssh 00:03:33 dns 00:03:40 cron 00:03:37 X 00:02:32
ksmd 00:03:51 system 00:03:51 kernel 00:03:51 node 00:03:38 other 00:03:51

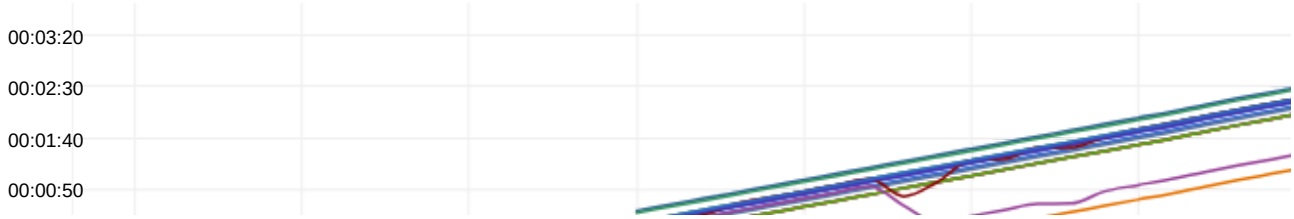
Apps Minimum Uptime (apps.uptime_min)



time сб, 24 апр. 2021 г. | 17:54:48

netdata 00:03:29 apps.plugin 00:03:24 node.d.plugin - python.d.plugin -
tc-qos-helper 00:03:24 go.d.plugin 00:03:24 ebpf.plugin 00:03:24 timedb 00:03:36 logs 00:03:37
nfs 00:03:39 VMs 00:03:30 ssh 00:02:29 dns 00:03:39 cron 00:03:36 X 00:02:31
ksmd 00:03:50 system 00:03:28 kernel 00:03:23 node 00:03:37 other 00:03:36

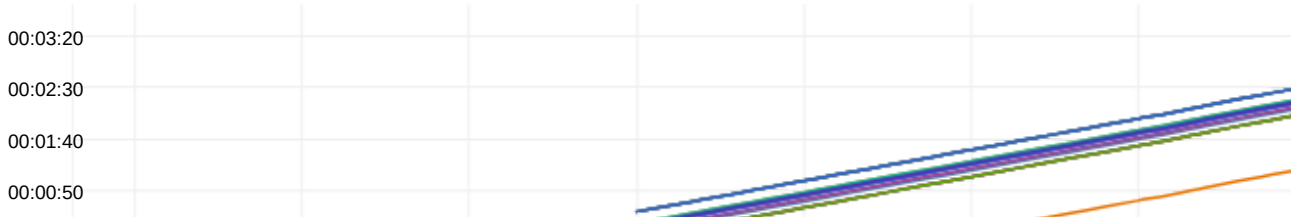
Apps Average Uptime (apps.uptime_avg)



time сб, 24 апр. 2021 г. | 17:54:48

netdata 00:03:32 apps.plugin 00:03:24 node.d.plugin - python.d.plugin -
tc-qos-helper 00:03:24 go.d.plugin 00:03:24 ebpf.plugin 00:03:24 timedb 00:03:36 logs 00:03:37
nfs 00:03:39 VMs 00:03:30 ssh 00:02:45 dns 00:03:39 cron 00:03:36 X 00:02:31
ksmd 00:03:50 system 00:03:39 kernel 00:03:48 node 00:03:37 other 00:03:39

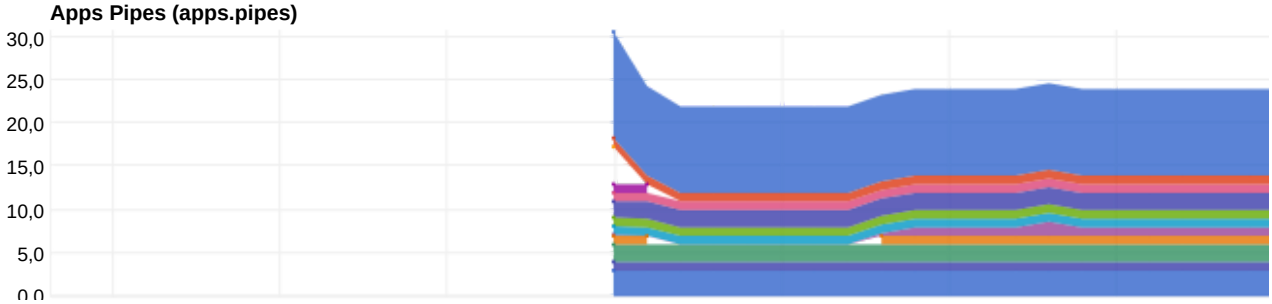
Apps Maximum Uptime (apps.uptime_max)



time сб, 24 апр. 2021 г. | 17:54:48

netdata 00:03:35 apps.plugin 00:03:24 node.d.plugin - python.d.plugin -
tc-qos-helper 00:03:24 go.d.plugin 00:03:24 ebpf.plugin 00:03:24 timedb 00:03:36 logs 00:03:37
nfs 00:03:39 VMs 00:03:30 ssh 00:03:32 dns 00:03:39 cron 00:03:36 X 00:02:31
ksmd 00:03:50 system 00:03:50 kernel 00:03:50 node 00:03:37 other 00:03:50

Apps Pipes (apps.pipes)



open pipes сб, 24 апр. 2021 г. | 17:54:48

netdata 10,0 apps.plugin 1,0 node.d.plugin - python.d.plugin - tc-qos-helper 1,0
go.d.plugin 2,0 ebpf.plugin 1,0 timedb 1,0 ssh 1,0 X 1,0 kernel 2,0
node 1,0 other 3,0

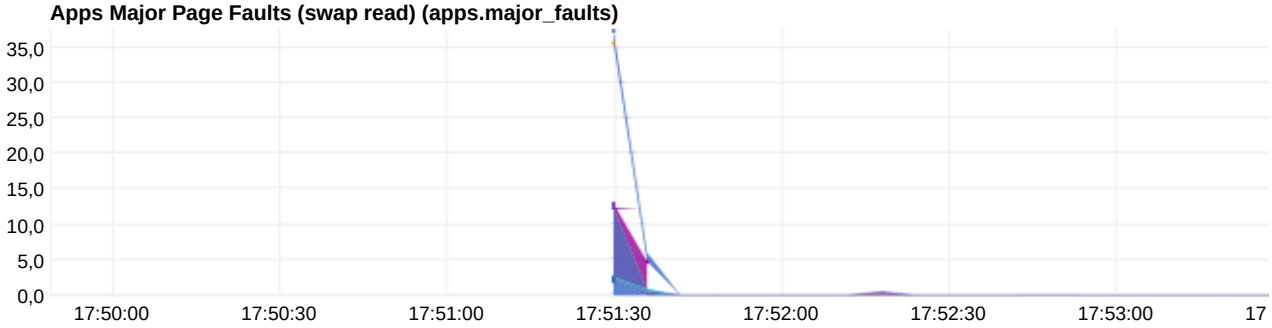
swap

Apps Swap Memory (apps.swap)

MiB сб, 24 апр. 2021 г. | 17:54:48

netdata 0 apps.plugin 0 charts.d.plugin - node.d.plugin - python.d.plugin -
tc-qos-helper 0 go.d.plugin 0 ebpf.plugin 0 timedb 0 logs 0 nfs 0
VMs 0 ssh 0 dns 0 cron 0 X 0 ksmd 0 system 0
kernel 0 node 0 other 0

Apps Major Page Faults (swap read) (apps.major_faults)

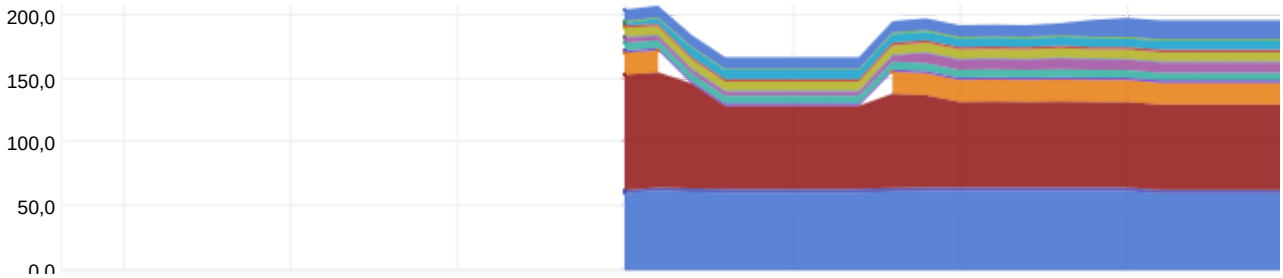


page faults/s сб, 24 апр. 2021 г. | 17:54:48

netdata 0,0 node.d.plugin - python.d.plugin - go.d.plugin 0,0 timedb 0,0
ssh 0,0 system 0,0 node 0,0 other 0,0

net

Apps Open Sockets (apps.sockets)



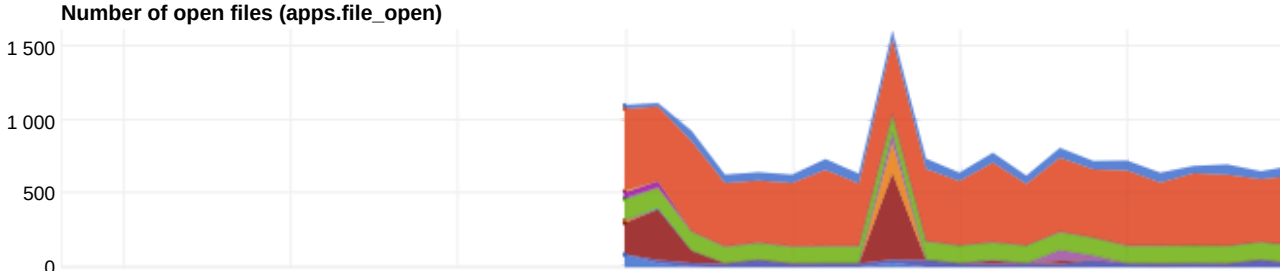
open sockets сб, 24 апр. 2021 г. | 17:54:48

netdata 12,0 ebpf.plugin 1,0 timedb 7,0 logs 2,0 nfs 7,0 VMs 1,0
ssh 8,0 dns 6,0 cron 2,0 X 17,0 system 67,0 node 3,0 other 60,0

file (eBPF)

Calls to the internal function `do_sys_open` (For kernels newer than `5.5.19` we add a kprobe to `do_sys_openat2`), which is the common function called from open(2) (https://www.man7.org/linux/man-pages/man2/open.2.html) and openat(2) (https://www.man7.org/linux/man-pages/man2/openat.2.html).

Number of open files (apps.file_open)

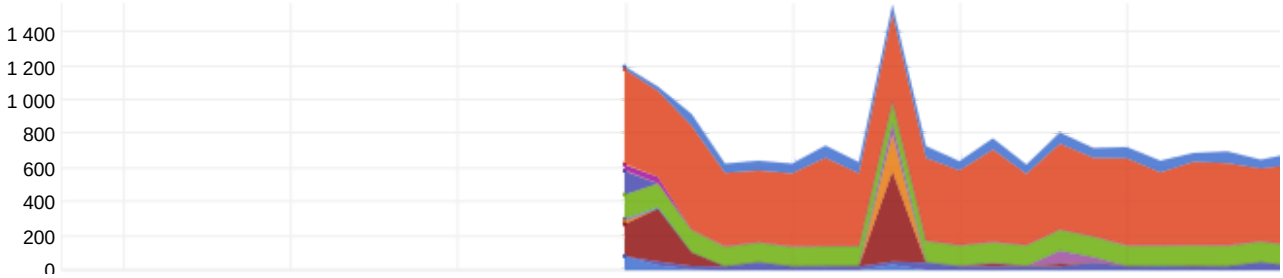


calls/s сб, 24 апр. 2021 г. | 17:54:48

netdata 44 apps.plugin 464 charts.d.plugin 0 node.d.plugin 0 python.d.plugin 0
tc-qos-helper 2 go.d.plugin 0 ebpf.plugin 120 timedb 0 logs 0
VMs 0 ssh 0 dns 0 X 0 system 1 node 23 other 0

Calls to the internal function __close_fd (https://elixir.bootlin.com/linux/v5.10/source/fs/file.c#L665) or close_fd (https://elixir.bootlin.com/linux/v5.11/source/fs/file.c#L617) according to your kernel version, which is called from close(2) (https://www.man7.org/linux/man-pages/man2/close.2.html).

Files closed (apps.file_closed)



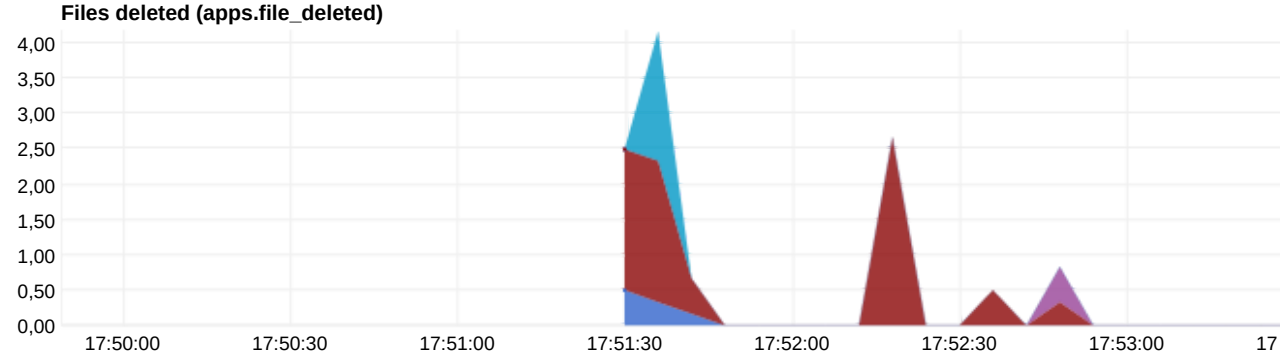
calls/s сб, 24 апр. 2021 г. | 17:54:48

netdata 44 apps.plugin 464 node.d.plugin 0 python.d.plugin 0 tc-qos-helper 4
go.d.plugin 0 ebpf.plugin 120 timedb 0 VMs 0 ssh 0 dns 0
X 0 system 1 node 22 other 0

vfs (eBPF)

Calls to the function vfs_unlink (https://www.kernel.org/doc/htmldocs/filesystems/API-vfs-unlink.html). This chart does not show all events that remove files from the filesystem, because filesystems can create their own functions to remove files.

Files deleted (apps.file_deleted)

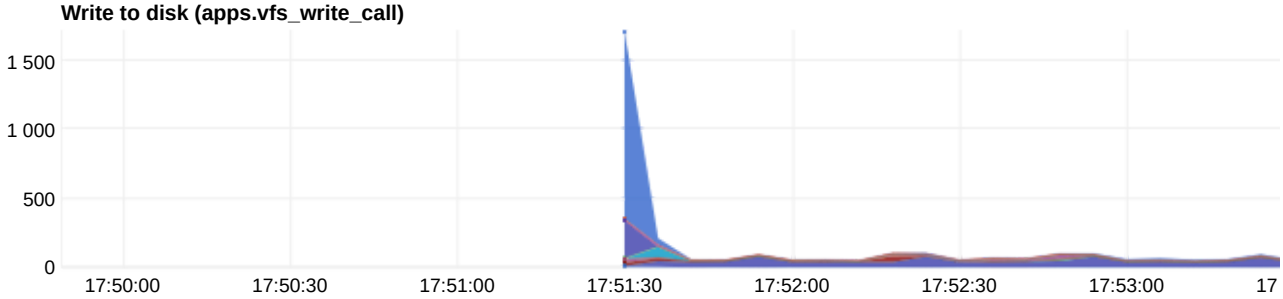


calls/s сб, 24 апр. 2021 г. | 17:54:48

timedb 0,00 ssh 0,00 system 0,00 other 0,00

Successful calls to the function vfs_write (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html). This chart may not show all filesystem events if it uses other functions to store data on disk.

Write to disk (apps.vfs_write_call)

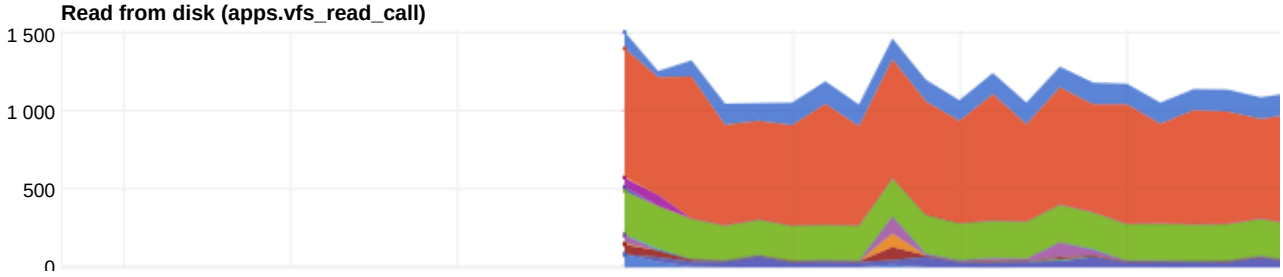


calls/s сб, 24 апр. 2021 г. | 17:54:48

netdata 8 apps.plugin 5 node.d.plugin 0 python.d.plugin 0 tc-qos-helper 1
go.d.plugin 1 ebpf.plugin 3 timedb 1 logs 0 ssh 0 dns 0
X 0 system 3 kernel 0 node 39 other 0

Successful calls to the function vfs_read (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html). This chart may not show all filesystem events if it uses other functions to store data on disk.

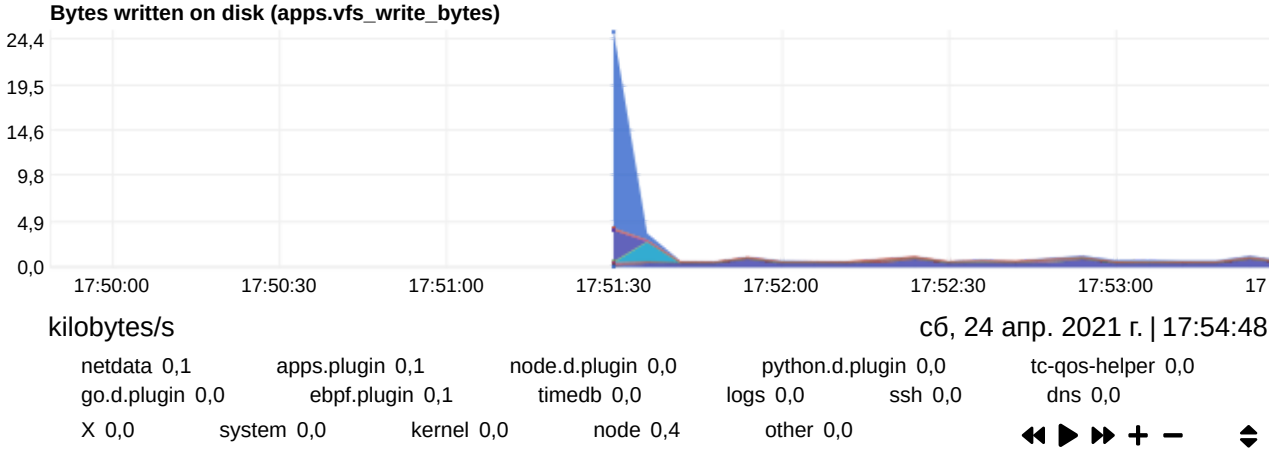
Read from disk (apps.vfs_read_call)



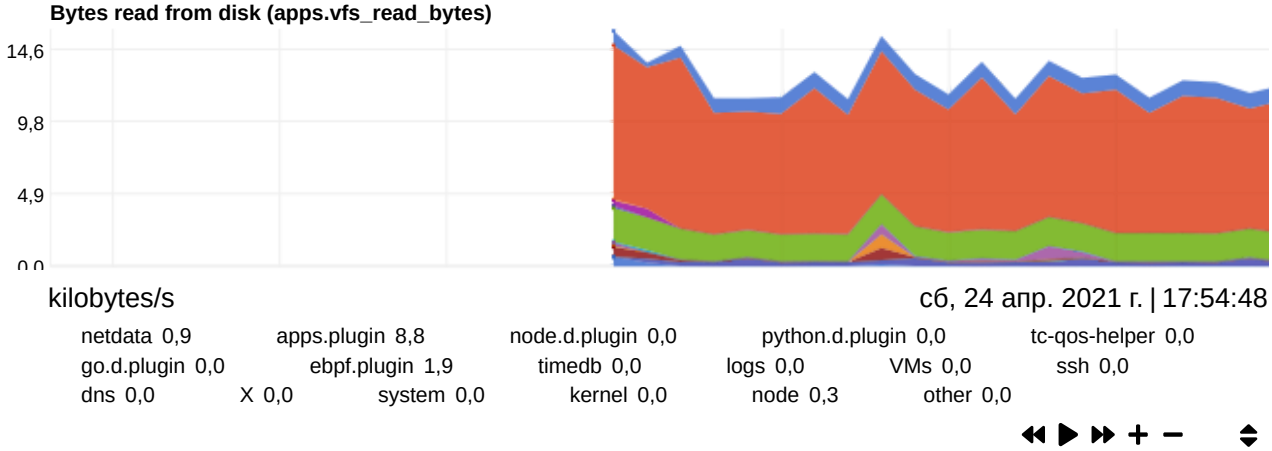
calls/s сб, 24 апр. 2021 г. | 17:54:48

netdata 121 apps.plugin 696 node.d.plugin 0 python.d.plugin 0 tc-qos-helper 2
go.d.plugin 0 ebpf.plugin 237 timedb 1 logs 0 VMs 1 ssh 0
dns 0 X 0 system 4 kernel 0 node 34 other 0

Total of bytes successfully written using the function vfs_write (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html).



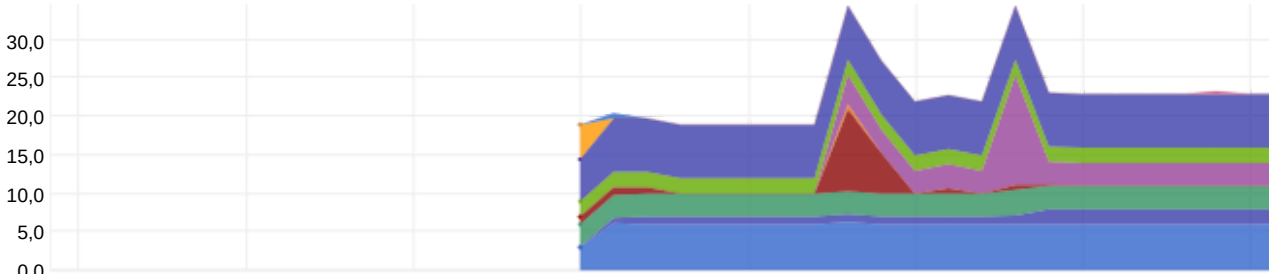
Total of bytes successfully read using the function vfs_read (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html).



process (eBPF)

Calls to either do_fork (https://www.ece.uic.edu/~yshi1/linux/lkse/node4.html#SECTION00421000000000000000), or `kernel_clone` if you are running kernel newer than 5.9.16, to create a new task, which is the common name used to define process and tasks inside the kernel. Netdata identifies the process by counting the number of calls to sys_clone (https://linux.die.net/man/2/clone) that do not have the flag `CLONE_THREAD` set.

Process started (apps.process_create)

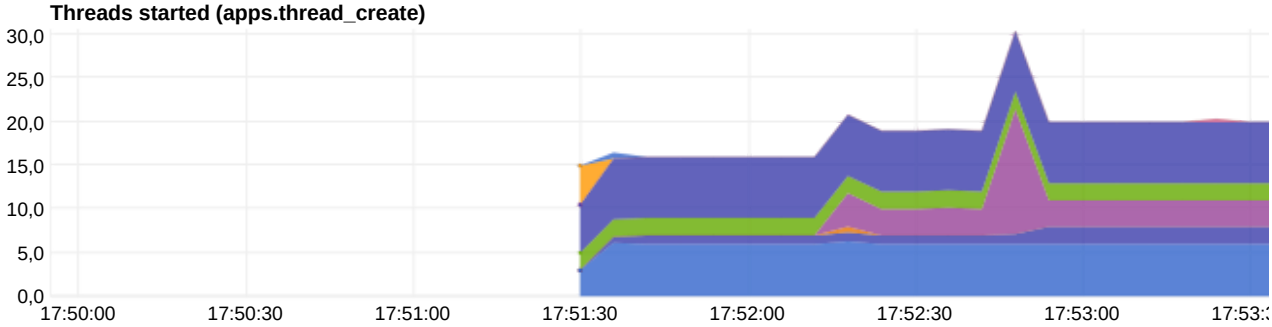


calls/s сб, 24 апр. 2021 г. | 17:54:54

netdata 0,0 node.d.plugin 0,0 tc-qos-helper 0,0 go.d.plugin 7,0 ebpf.plugin 2,0
ssh 3,0 X 0,0 system 0,0 kernel 3,0 node 2,0 other 7,0

Calls to either do_fork (https://www.ece.uic.edu/~yshi1/linux/lkse/node4.html#SECTION00421000000000000000), or `kernel_clone` if you are running kernel newer than 5.9.16, to create a new task, which is the common name used to define process and tasks inside the kernel. Netdata identifies the threads by counting the number of calls to sys_clone (https://linux.die.net/man/2/clone) that have the flag `CLONE_THREAD` set.

Threads started (apps.thread_create)

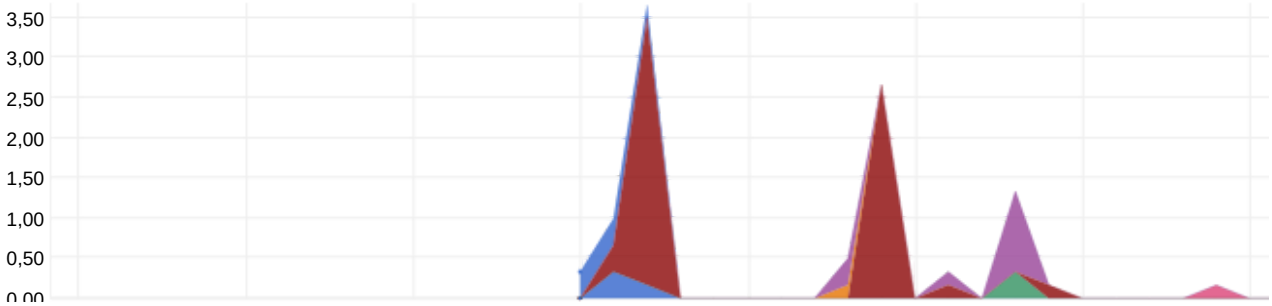


calls/s сб, 24 апр. 2021 г. | 17:54:54

netdata 0,0 node.d.plugin 0,0 tc-qos-helper 0,0 go.d.plugin 7,0 ebpf.plugin 2,0
ssh 3,0 X 0,0 node 2,0 other 7,0

Calls to the functions responsible for closing (do_exit (https://www.informit.com/articles/article.aspx?p=370047&seqNum=4)) and releasing (release_task (https://www.informit.com/articles/article.aspx?p=370047&seqNum=4)) tasks.

Tasks closed (apps.task_close)



calls/s сб, 24 апр. 2021 г. | 17:54:54

netdata 0,00 tc-qos-helper 0,00 ssh 0,00 X 0,00 system 0,00 kernel 0,00

other 0,00

net (eBPF)

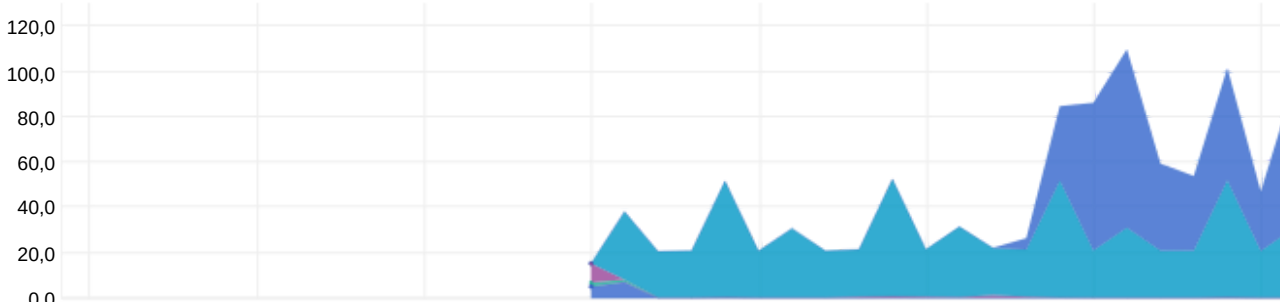
Bytes sent (apps.total_bandwidth_sent)

megabits/s сб, 24 апр. 2021 г. | 17:54:54

netdata 0,07 go.d.plugin 0,00 timedb 0,01 ssh 0,00 dns 0,00 node 0,04

other 0,00

bytes received (apps.total_bandwidth_recv)



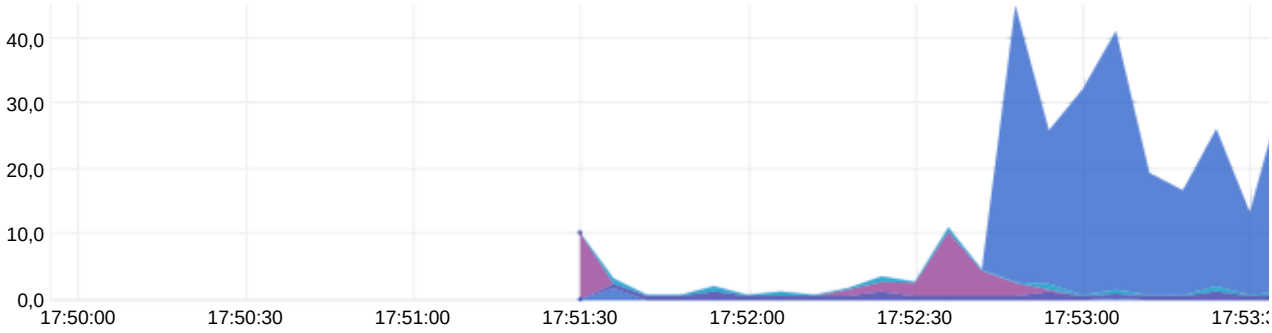
kilobits/s сб, 24 апр. 2021 г. | 17:54:54

netdata 43,3 go.d.plugin 0,0 timedb 51,6 ssh 0,0 dns 0,0 node 0,7

other 0,0

Calls for function `tcp_sendmsg` .

Calls for tcp_sendmsg (apps.bandwidth_tcp_send)

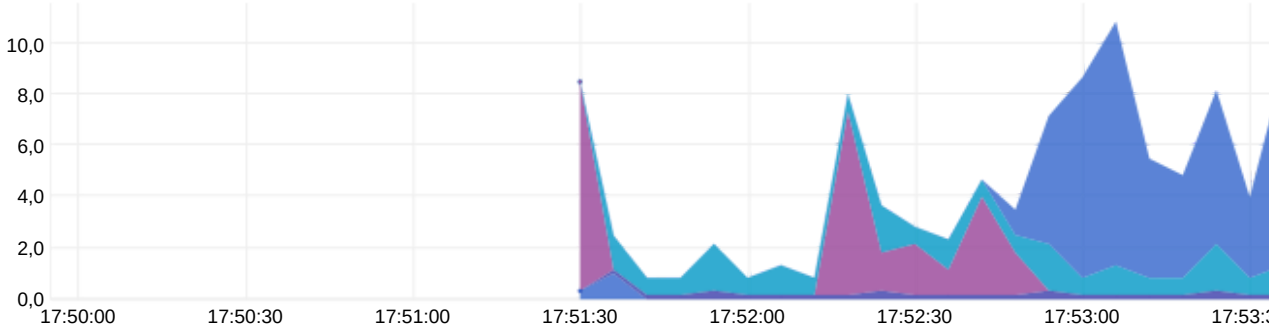


calls/s сб, 24 апр. 2021 г. | 17:54:54

netdata 21,3 timedb 0,8 ssh 0,0 node 1,3 other 0,0

Calls for functions `tcp_cleanup_rbuf` . We use `tcp_cleanup_rbuf` instead `tcp_recvmsg` , because this last misses `tcp_read_sock()` traffic and we would also need to have more probes to get the socket and package size.

Calls for tcp_cleanup_rbuf (apps.bandwidth_tcp_recv)

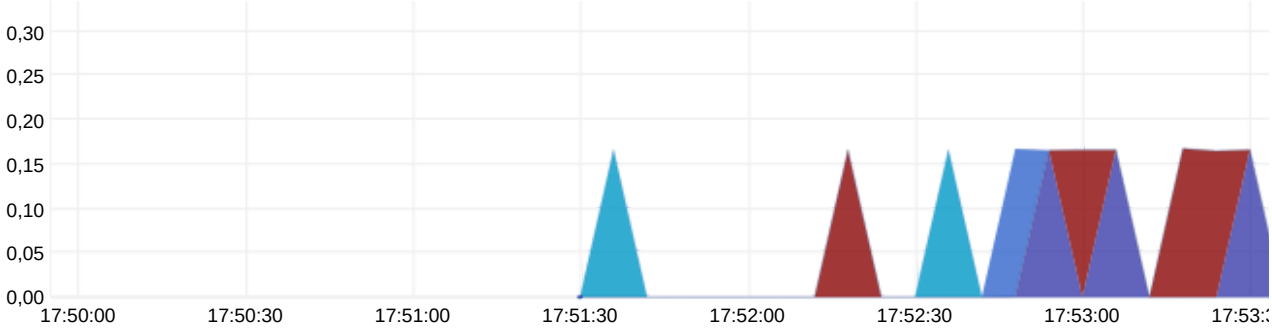


calls/s сб, 24 апр. 2021 г. | 17:54:54

netdata 5,3 timedb 1,8 ssh 0,0 node 0,3 other 0,0

Calls for functions `tcp_retransmit_skb` .

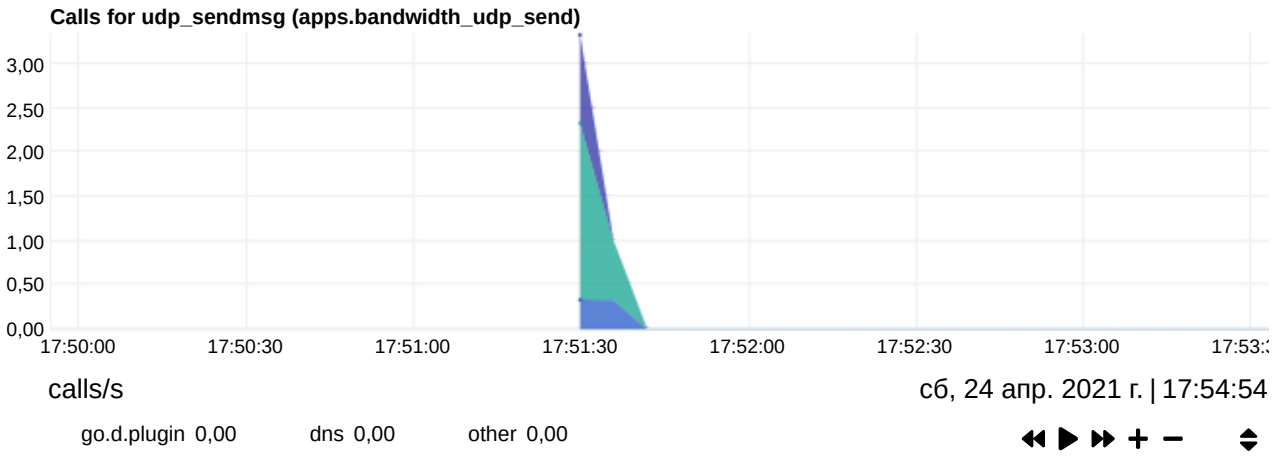
Calls for tcp_retransmit (apps.bandwidth_tcp_retransmit)



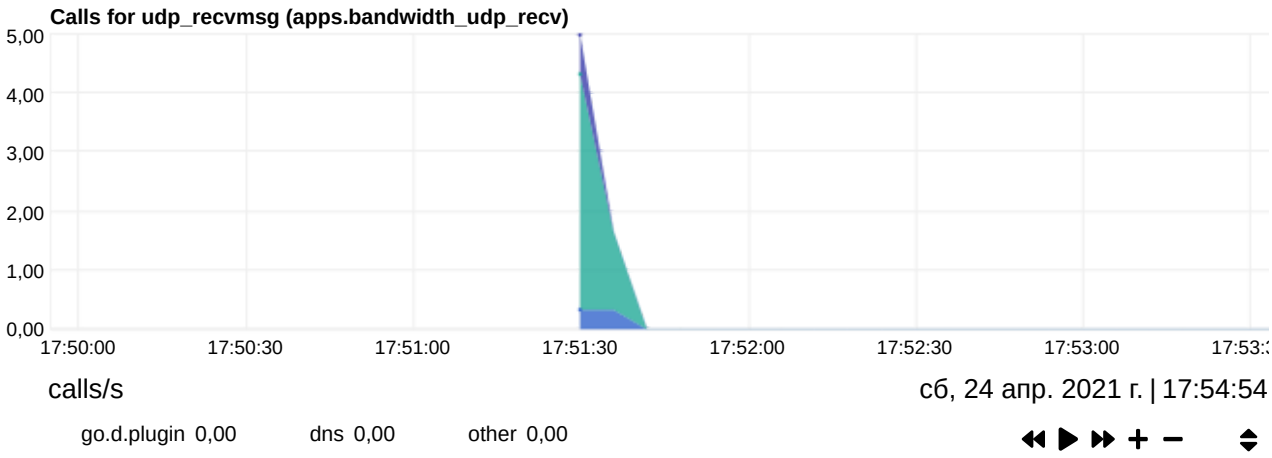
calls/s сб, 24 апр. 2021 г. | 17:54:54

netdata 0,00 timedb 0,00 system 0,00 node 0,00 other 0,00

Calls for function `udp_sendmsg` .



Calls for function `udp_recvmsg` .

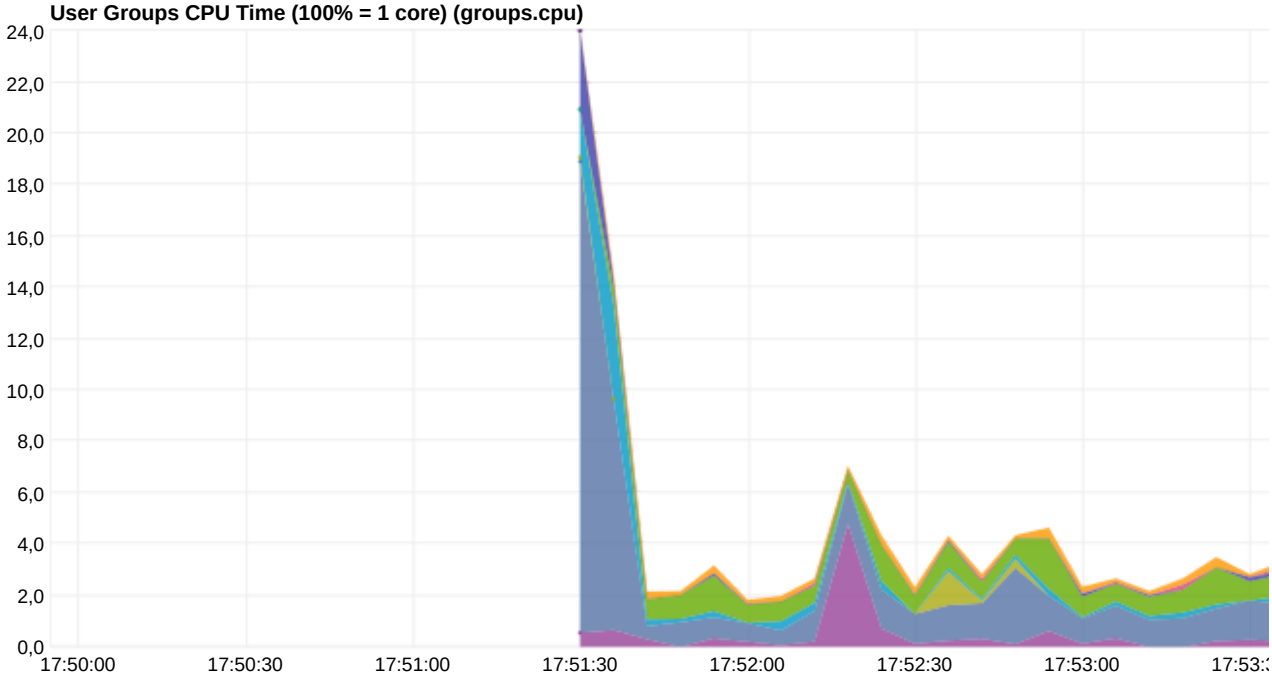


User Groups

Per user group statistics are collected using netdata's `apps.plugin` . This plugin walks through all processes and aggregates statistics per user group. The reported values are compatible with `top` , although the netdata plugin counts also the resources of exited children (unlike `top` which shows only the resources of the currently running processes). So for processes like shell scripts, the reported values include the resources used by the commands these scripts run within each timeframe.

cpu

User Groups CPU Time (100% = 1 core) (groups.cpu)

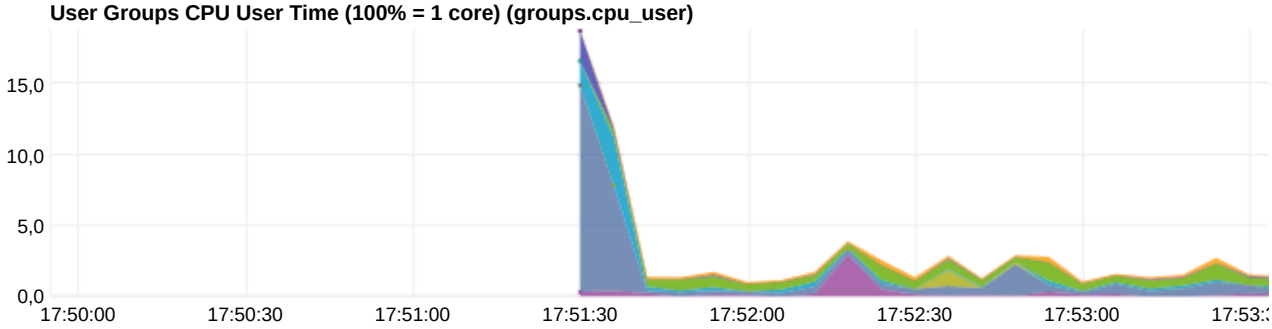


percentage сб, 24 апр. 2021 г. | 17:54:54

messagebus 0,5 alertmanager 0,0 grafana 0,0 nodeusr 1,7 prometheus 0,2

vagrant 0,0 netdata 0,8 root 0,0

User Groups CPU User Time (100% = 1 core) (groups.cpu_user)

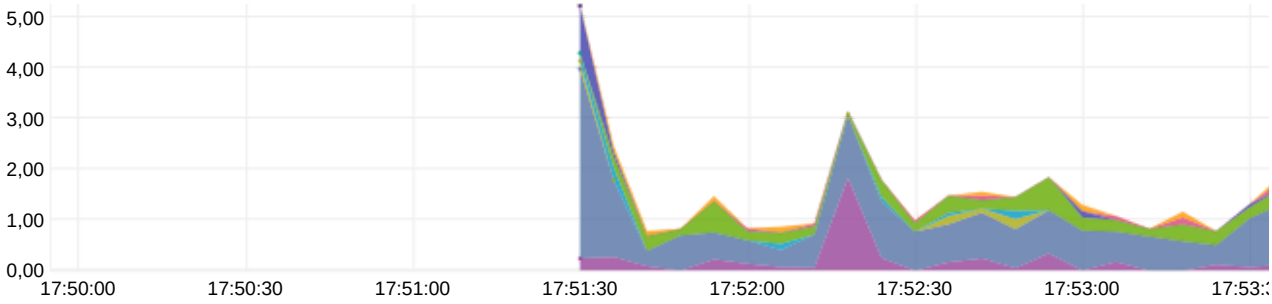


percentage сб, 24 апр. 2021 г. | 17:54:54

messagebus 0,2 alertmanager 0,0 grafana 0,0 nodeusr 1,1 prometheus 0,2

vagrant 0,0 netdata 0,5 root 0,0

User Groups CPU System Time (100% = 1 core) (groups.cpu_system)



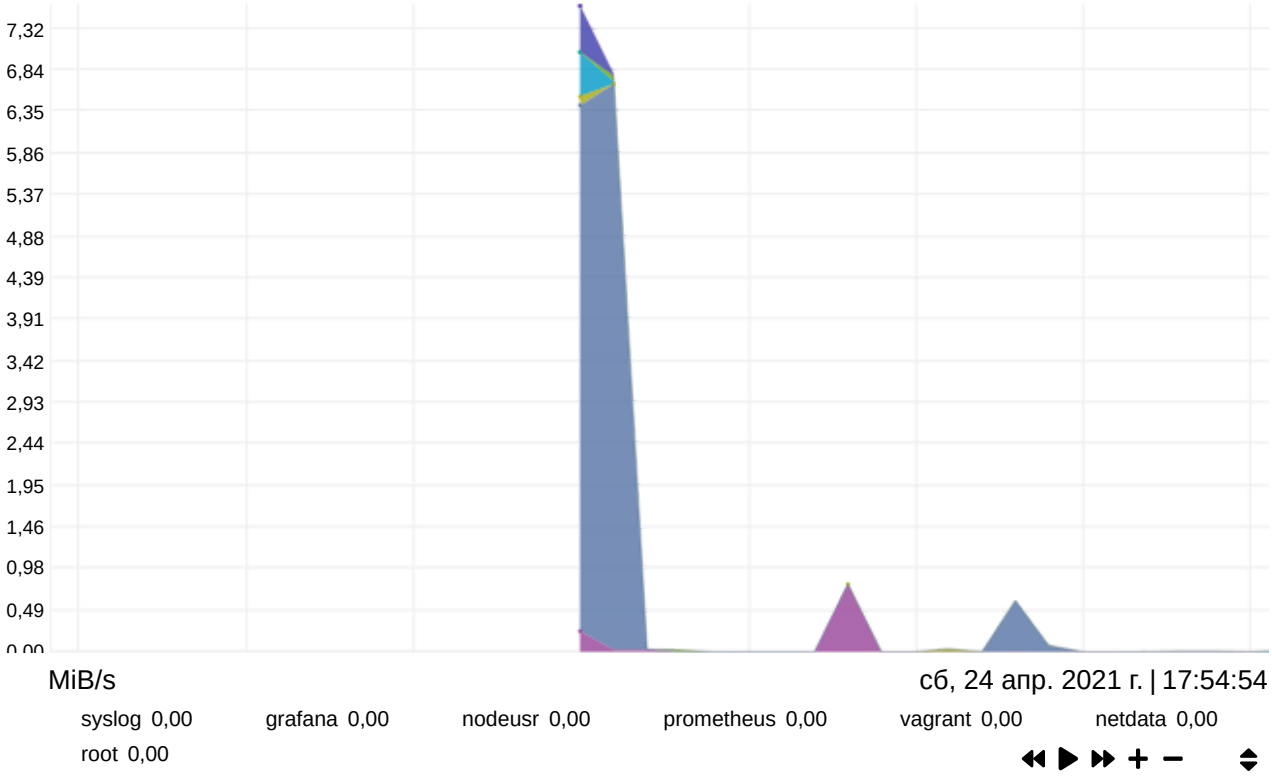
percentage сб, 24 апр. 2021 г. | 17:54:54

messagebus 0,28 alertmanager 0,00 grafana 0,00 nodeusr 0,64 prometheus 0,00

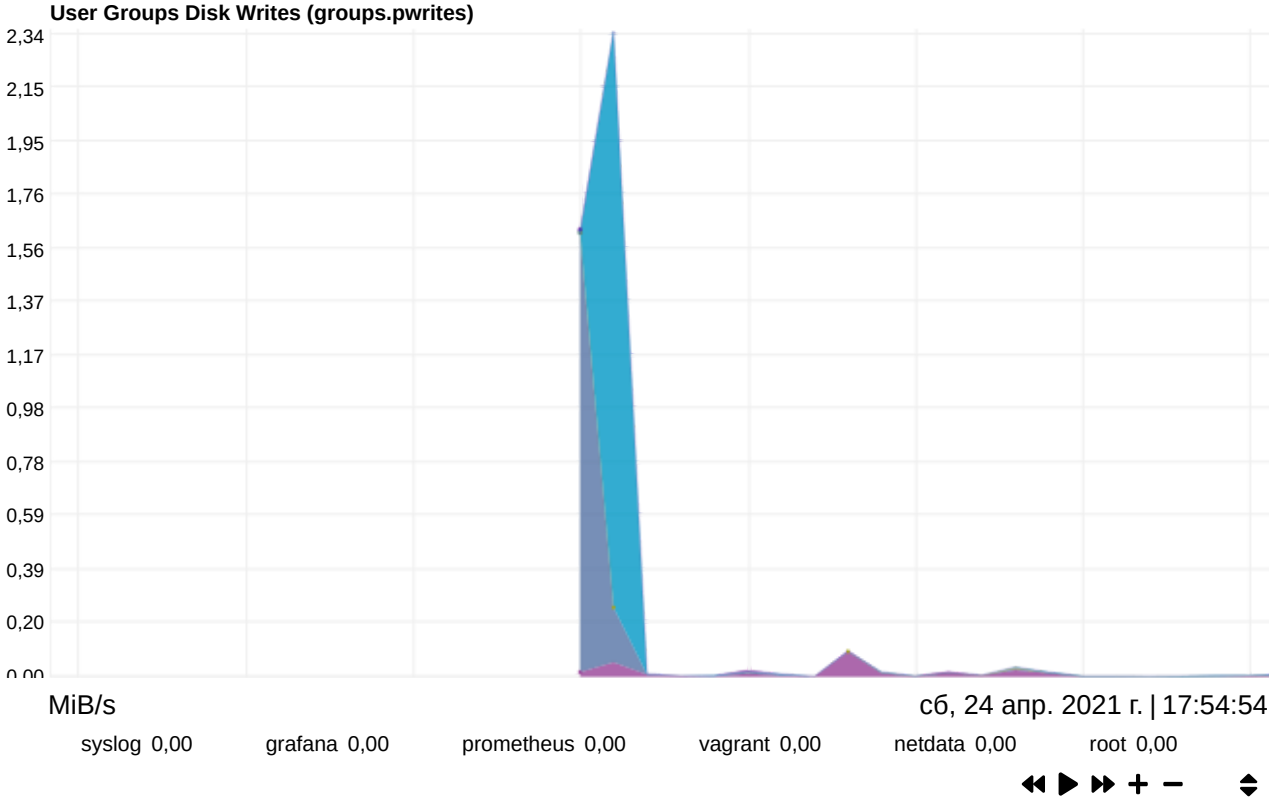
vagrant 0,00 netdata 0,31 root 0,00

disk

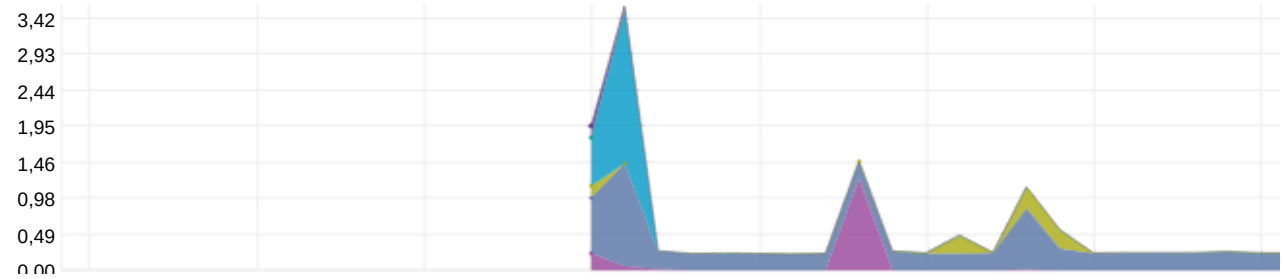
User Groups Disk Reads (groups.preads)



User Groups Disk Writes (groups.pwrites)



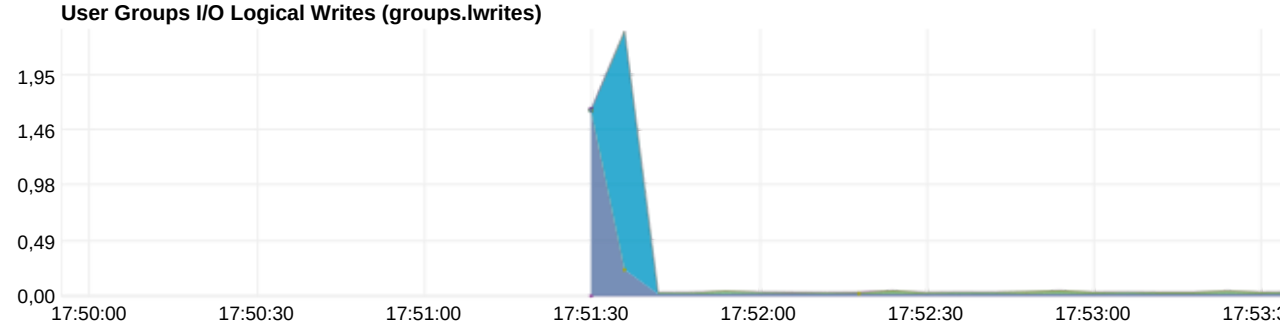
User Groups Disk Logical Reads (groups.lreads)



MiB/s сб, 24 апр. 2021 г. | 17:54:54

systemd-network 0,00 systemd-resolve 0,00 messagebus 0,00 syslog 0,00
alertmanager 0,00 grafana 0,00 nodeusr 0,01 prometheus 0,01 vagrant 0,00
netdata 0,23 root 0,00

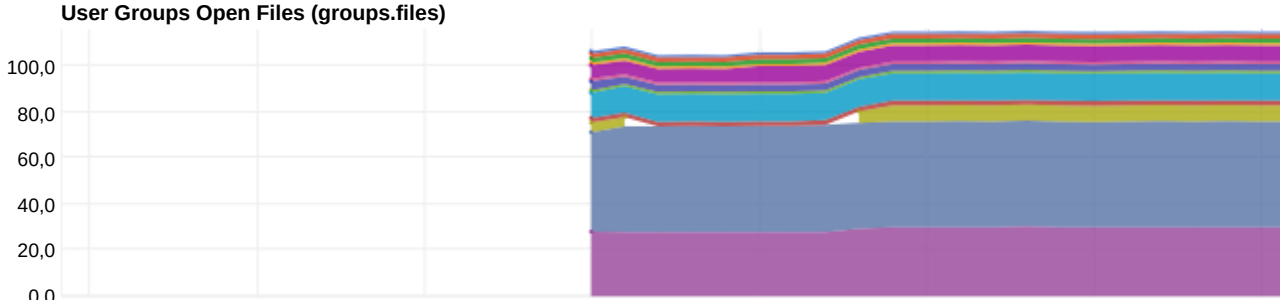
User Groups I/O Logical Writes (groups.lwrites)



MiB/s сб, 24 апр. 2021 г. | 17:54:54

systemd-resolve 0,00 syslog 0,00 alertmanager 0,00 grafana 0,00 nodeusr 0,02
prometheus 0,00 vagrant 0,00 netdata 0,03 root 0,00

User Groups Open Files (groups.files)



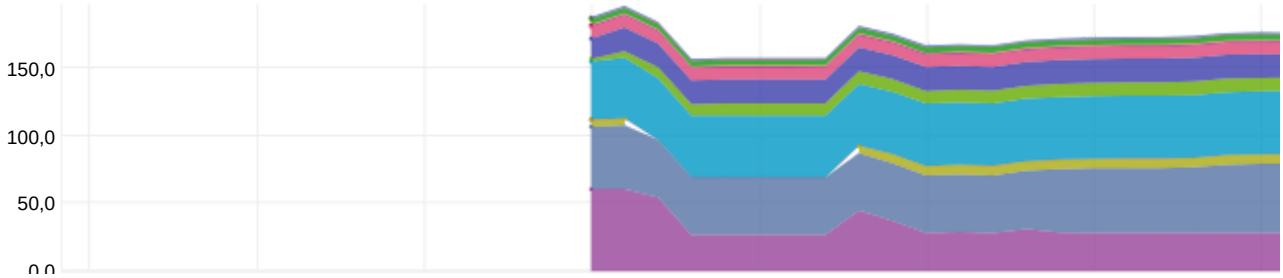
open files сб, 24 апр. 2021 г. | 17:54:54

systemd-network 1,0 nogroup 2,0 systemd-resolve 2,0 messagebus 1,0 syslog 7,0
alertmanager 1,0 grafana 3,0 nodeusr 1,0 prometheus 12,0 daemon 2,0
vagrant 7,0 netdata 46,0 root 30,0

mem

Real memory (RAM) used per user group. This does not include shared memory.

User Groups Real Memory (w/o shared) (groups.mem)

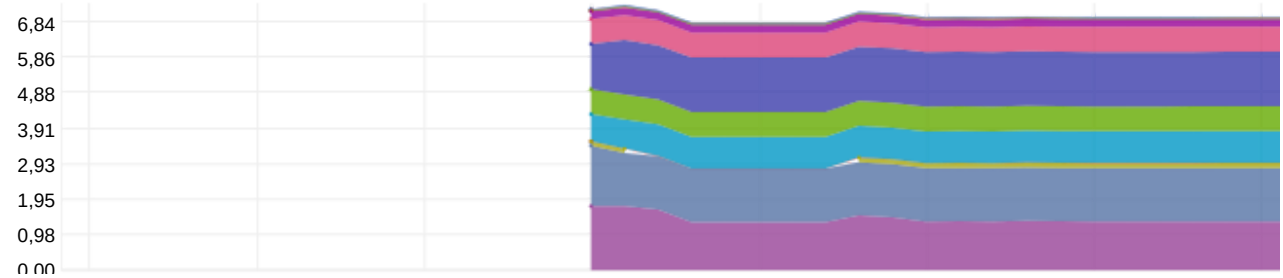


MiB сб, 24 апр. 2021 г. | 17:54:54

systemd-network 0,9 nogroup 0,4 systemd-resolve 4,1 messagebus 0,7 syslog 1,0
alertmanager 8,9 grafana 17,1 nodeusr 10,4 prometheus 47,2 daemon 0,2
vagrant 6,9 netdata 51,9 root 28,6

Virtual memory allocated per user group since the Netdata restart. Please check this article (https://github.com/netdata/netdata/tree/master/daemon#virtual-memory) for more information.

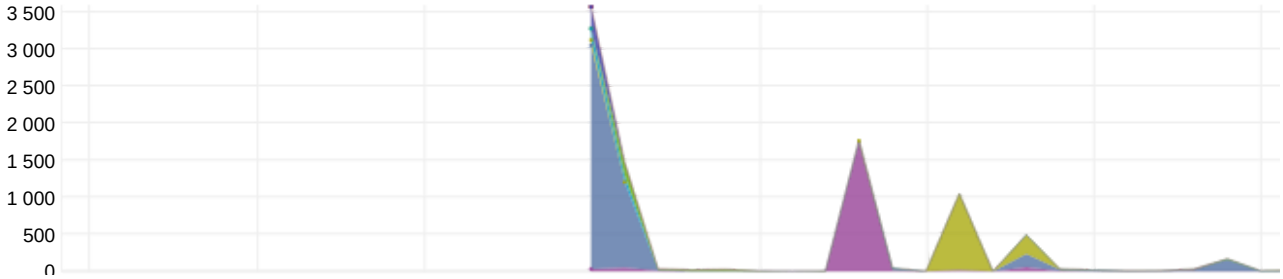
User Groups Virtual Memory Size (groups.vmem)



GiB сб, 24 апр. 2021 г. | 17:54:54

systemd-network 0,03 nogroup 0,01 systemd-resolve 0,02 messagebus 0,01 syslog 0,21
alertmanager 0,69 grafana 1,50 nodeusr 0,69 prometheus 0,87 daemon 0,00
vagrant 0,14 netdata 1,48 root 1,35

User Groups Minor Page Faults (groups.minor_faults)

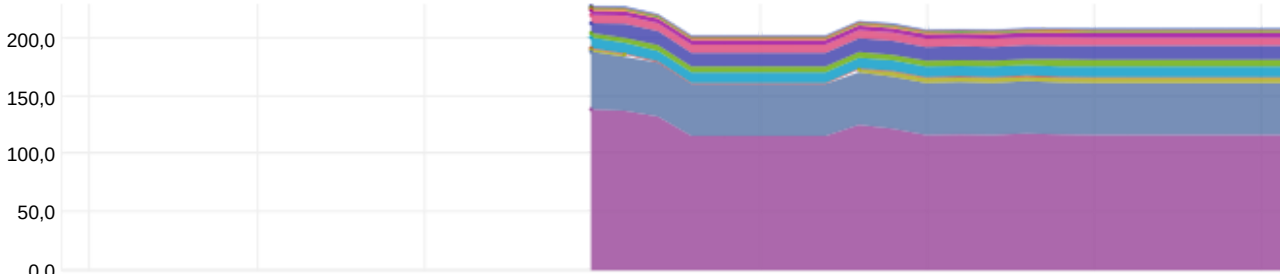


page faults/s сб, 24 апр. 2021 г. | 17:54:54

systemd-resolve 0 messagebus 0 syslog 0 alertmanager 0 grafana 0
nodeusr 2 prometheus 0 vagrant 0 netdata 12 root 0

processes

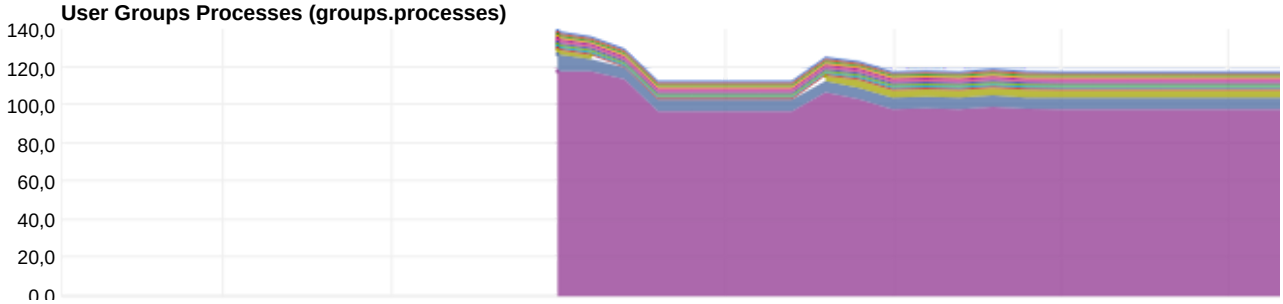
User Groups Threads (groups.threads)



threads сб, 24 апр. 2021 г. | 17:54:54

systemd-network 1,0 nogroup 1,0 systemd-resolve 1,0 messagebus 1,0 syslog 4,0
alertmanager 8,0 grafana 12,0 nodeusr 6,0 prometheus 9,0 daemon 1,0
vagrant 4,0 netdata 45,0 root 117,0

User Groups Processes (groups.processes)

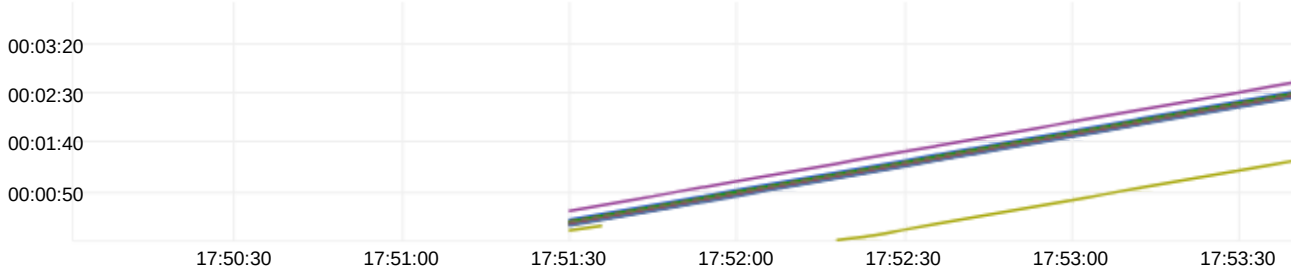


processes сб, 24 апр. 2021 г. | 17:55:00

systemd-network 1,0 nogroup 1,0 systemd-resolve 1,0 messagebus 1,0 syslog 1,0
alertmanager 1,0 grafana 1,0 nodeusr 1,0 prometheus 1,0 daemon 1,0
vagrant 4,0 netdata 6,0 root 98,0

Carried over process group uptime. The period of time within which at least one process in the group was running.

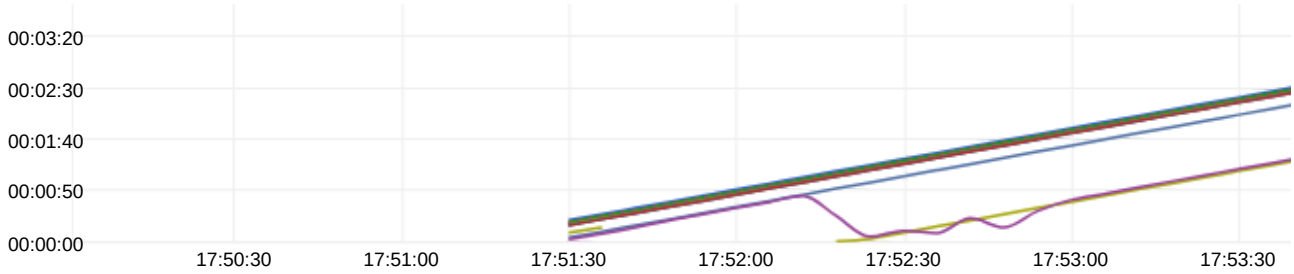
User Groups Carried Over Uptime (groups.uptime)



time сб, 24 апр. 2021 г. | 17:55:00

systemd-network 00:03:52 nogroup 00:03:50 systemd-resolve 00:03:50 messagebus 00:03:48
syslog 00:03:48 alertmanager 00:03:48 grafana 00:03:48 nodeusr 00:03:48 prometheus 00:03:47
daemon 00:03:47 vagrant 00:02:42 netdata 00:03:46 root 00:04:01

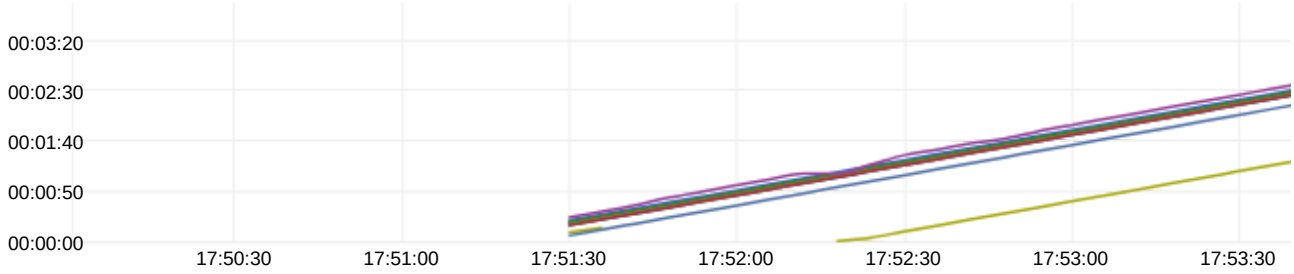
User Groups Minimum Uptime (groups.uptime_min)



time сб, 24 апр. 2021 г. | 17:55:00

systemd-network 00:03:51 nogroup 00:03:49 systemd-resolve 00:03:49 messagebus 00:03:47
syslog 00:03:47 alertmanager 00:03:47 grafana 00:03:47 nodeusr 00:03:47 prometheus 00:03:46
daemon 00:03:46 vagrant 00:02:39 netdata 00:03:34 root 00:02:41

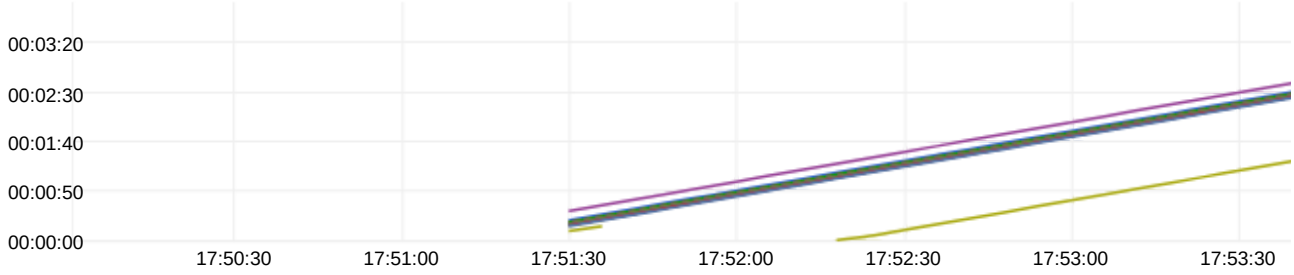
User Groups Average Uptime (groups.uptime_avg)



time сб, 24 апр. 2021 г. | 17:55:00

systemd-network 00:03:51 nogroup 00:03:49 systemd-resolve 00:03:49 messagebus 00:03:47
syslog 00:03:47 alertmanager 00:03:47 grafana 00:03:47 nodeusr 00:03:47 prometheus 00:03:46
daemon 00:03:46 vagrant 00:02:40 netdata 00:03:36 root 00:03:56

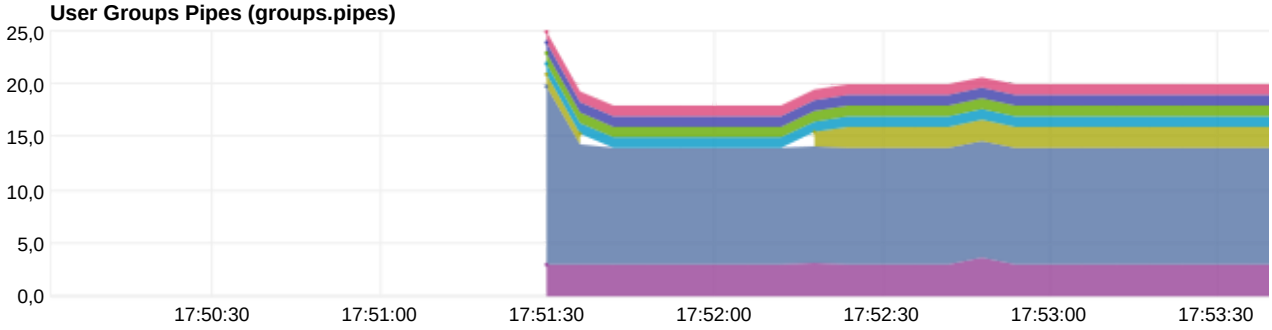
User Groups Maximum Uptime (groups.uptime_max)



time сб, 24 апр. 2021 г. | 17:55:00

systemd-network 00:03:51 nogroup 00:03:49 systemd-resolve 00:03:49 messagebus 00:03:47
syslog 00:03:47 alertmanager 00:03:47 grafana 00:03:47 nodeusr 00:03:47 prometheus 00:03:46
daemon 00:03:46 vagrant 00:02:41 netdata 00:03:45 root 00:04:00

User Groups Pipes (groups.pipes)

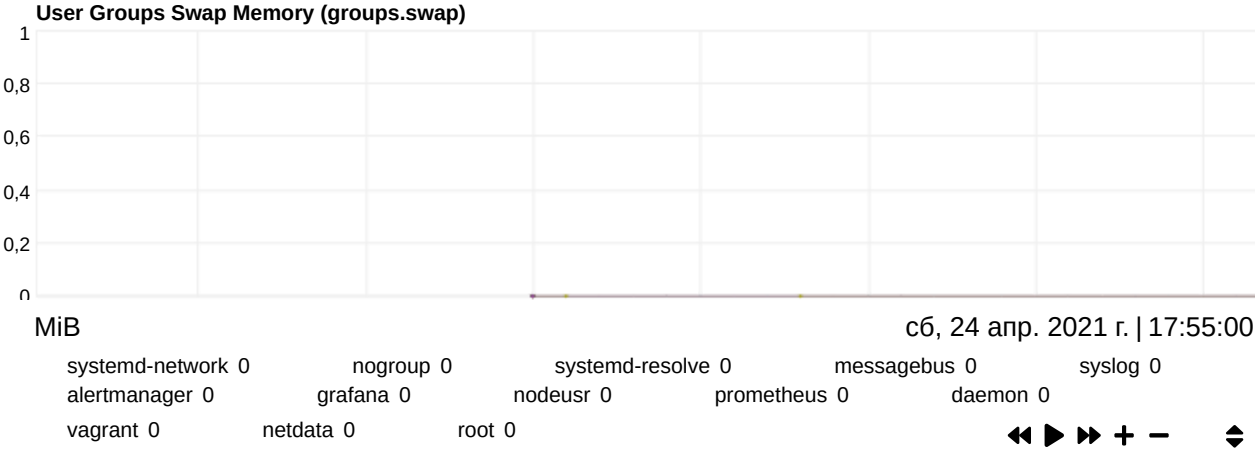


open pipes сб, 24 апр. 2021 г. | 17:55:00

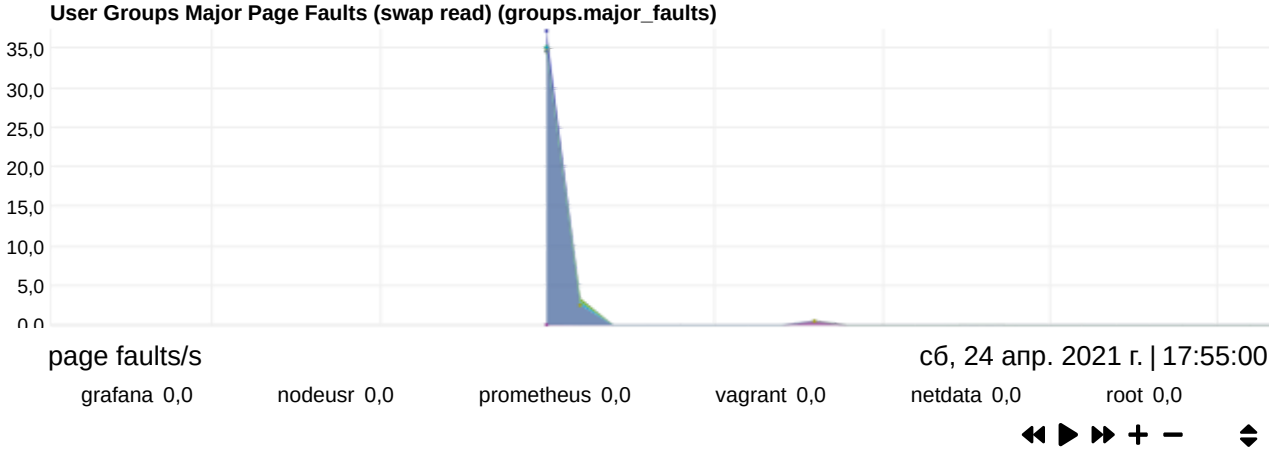
alertmanager 1,0 grafana 1,0 nodeusr 1,0 prometheus 1,0 vagrant 2,0
netdata 11,0 root 3,0

swap

User Groups Swap Memory (groups.swap)

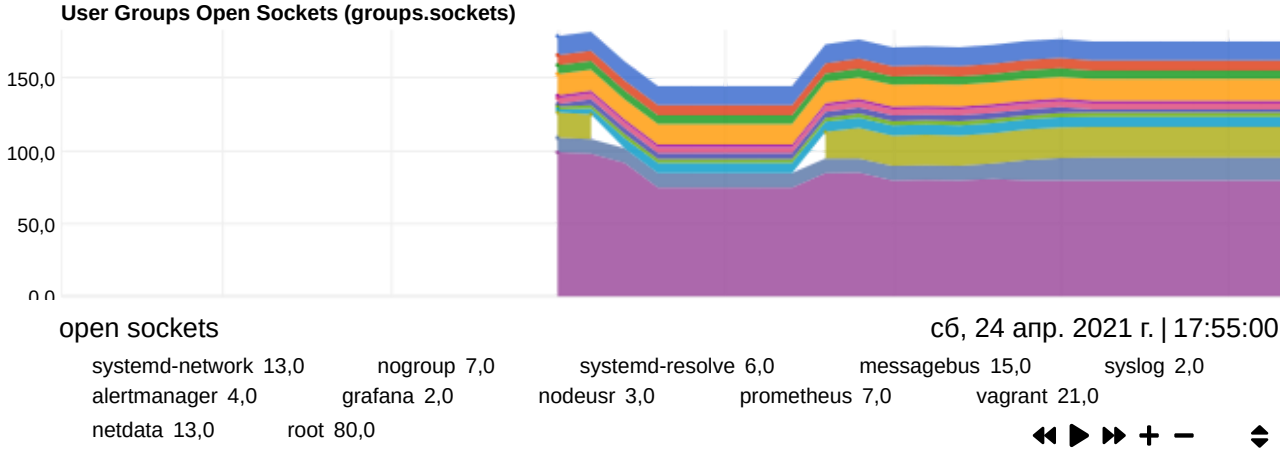


User Groups Major Page Faults (swap read) (groups.major_faults)



net

User Groups Open Sockets (groups.sockets)

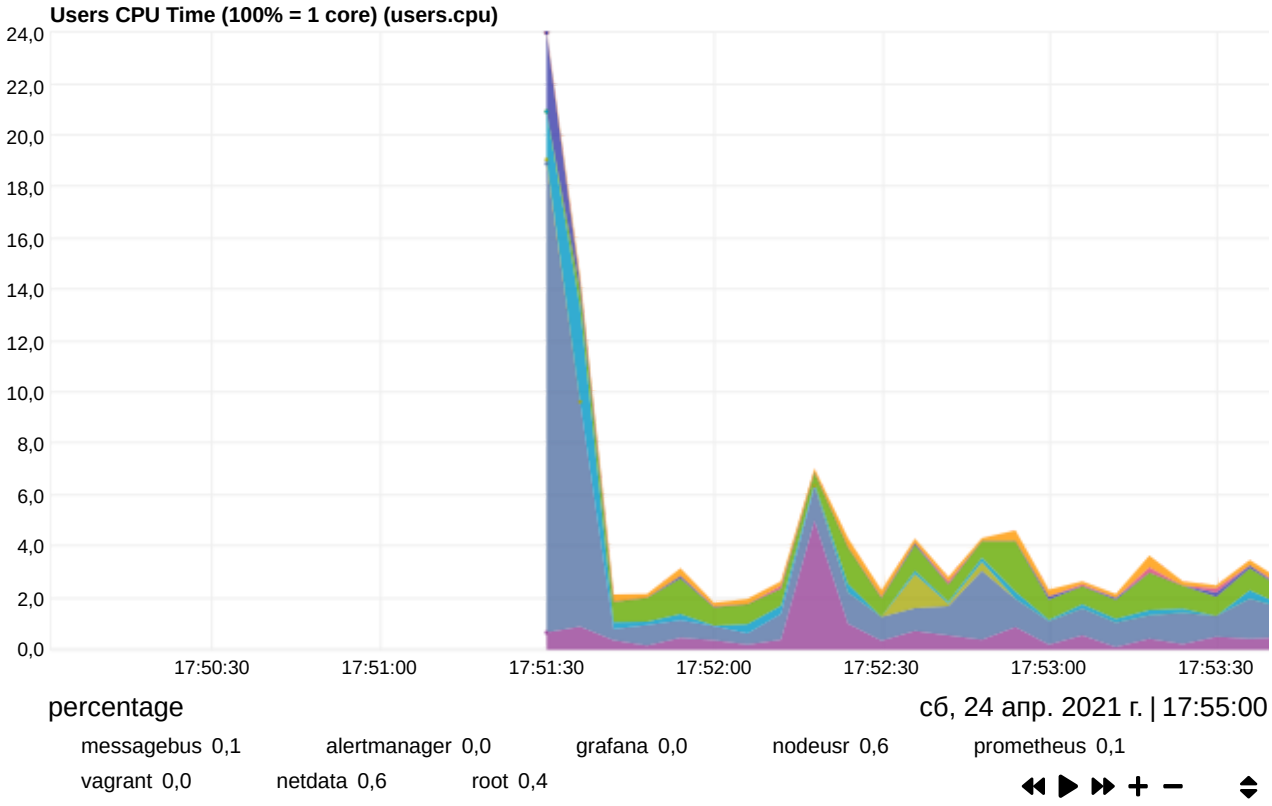


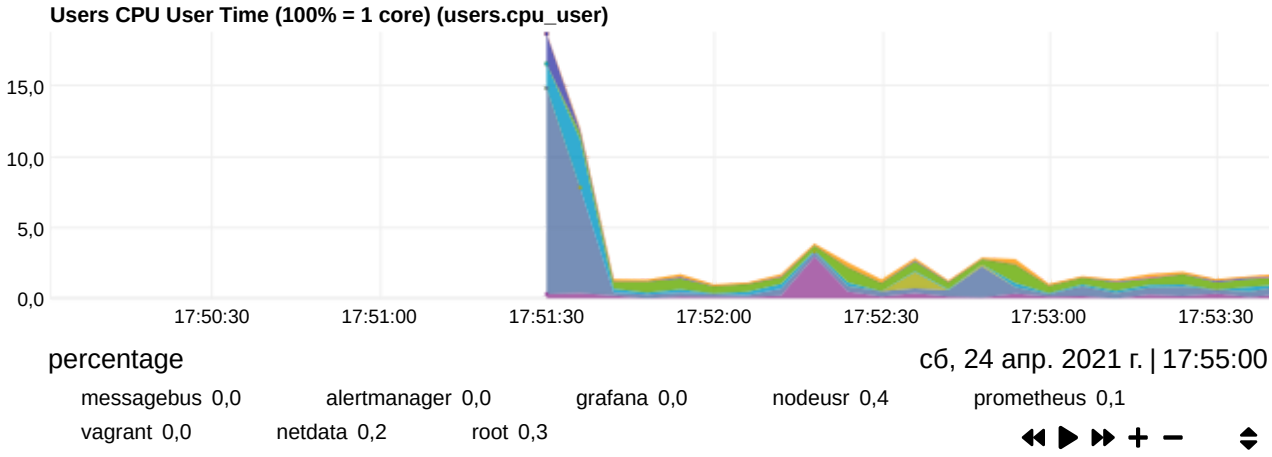
Users

Per user statistics are collected using netdata's `apps.plugin` . This plugin walks through all processes and aggregates statistics per user. The reported values are compatible with `top` , although the netdata plugin counts also the resources of exited children (unlike `top` which shows only the resources of the

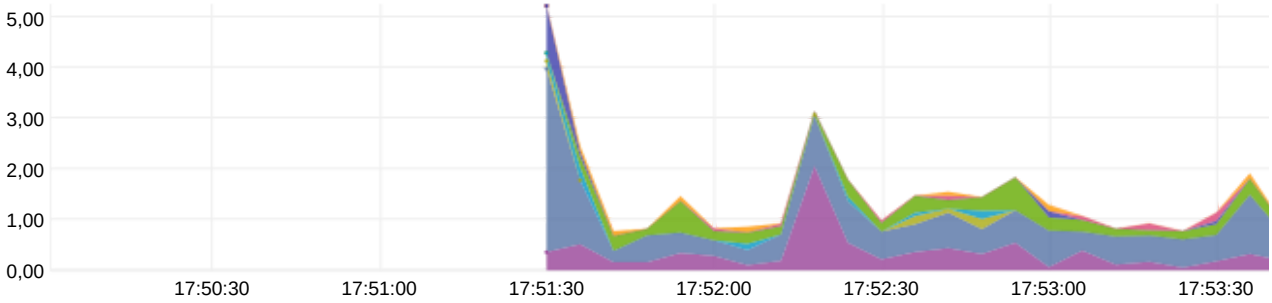
currently running processes). So for processes like shell scripts, the reported values include the resources used by the commands these scripts run within each timeframe.

cpu





Users CPU System Time (100% = 1 core) (users.cpu_system)

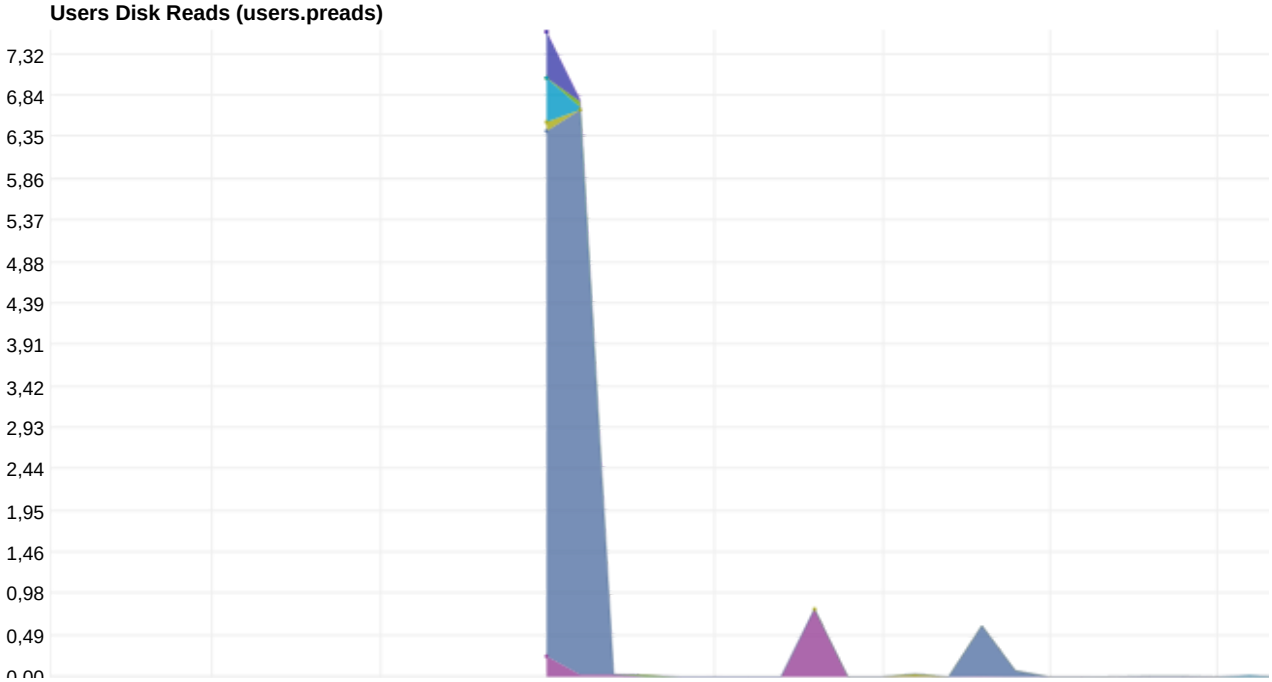


percentage сб, 24 апр. 2021 г. | 17:55:00

messagebus 0,09 alertmanager 0,00 grafana 0,00 nodeusr 0,09 prometheus 0,00
vagrant 0,00 netdata 0,45 root 0,14

disk

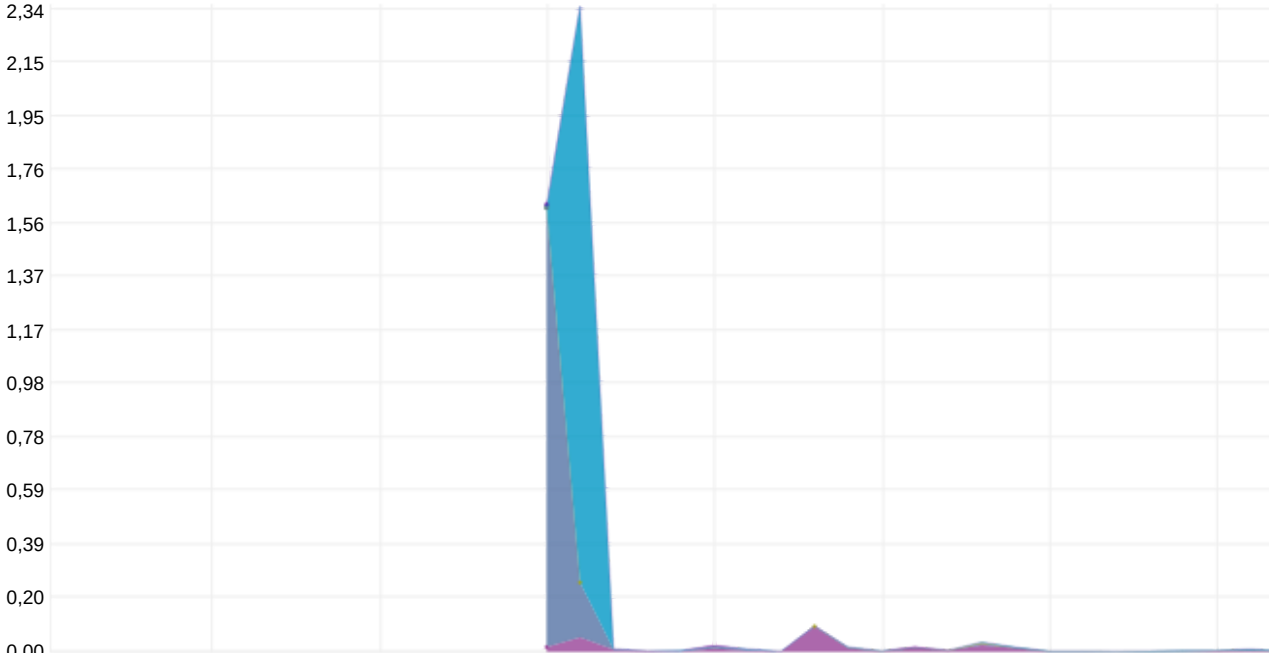
Users Disk Reads (users.preads)



MiB/s сб, 24 апр. 2021 г. | 17:55:00

syslog 0,00 grafana 0,00 nodeusr 0,00 prometheus 0,00 vagrant 0,00 netdata 0,00
root 0,00

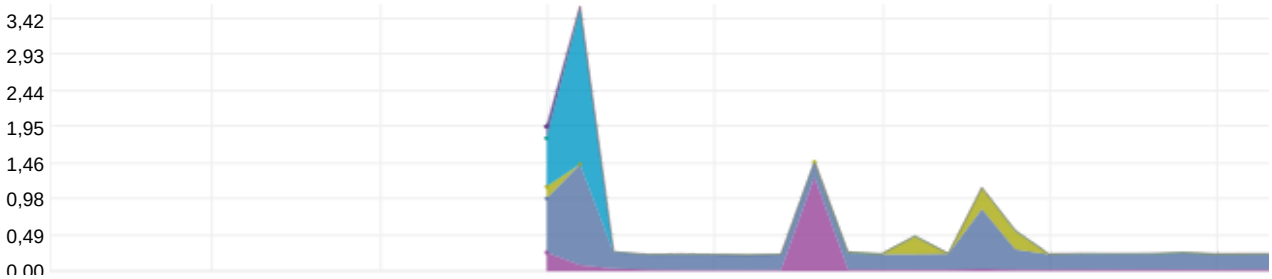
Users Disk Writes (users.pwrites)



MiB/s сб, 24 апр. 2021 г. | 17:55:00

syslog 0,00 grafana 0,00 prometheus 0,00 vagrant 0,00 netdata 0,00 root 0,00

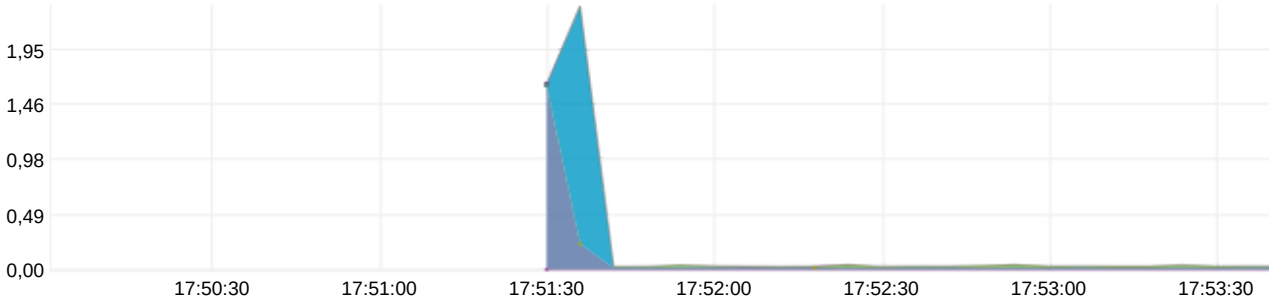
Users Disk Logical Reads (users.lreads)



MiB/s сб, 24 апр. 2021 г. | 17:55:00

systemd-network 0,00 systemd-resolve 0,00 messagebus 0,00 syslog 0,00
alertmanager 0,00 grafana 0,00 nodeusr 0,00 prometheus 0,00 vagrant 0,00
netdata 0,21 root 0,02

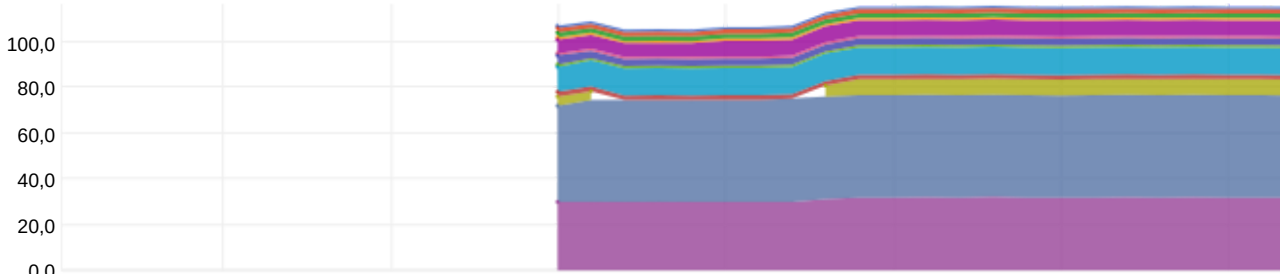
Users I/O Logical Writes (users.lwrites)



MiB/s сб, 24 апр. 2021 г. | 17:55:00

systemd-resolve 0,00 syslog 0,00 alertmanager 0,00 grafana 0,00 nodeusr 0,01
prometheus 0,00 vagrant 0,00 netdata 0,02 root 0,01

Users Open Files (users.files)



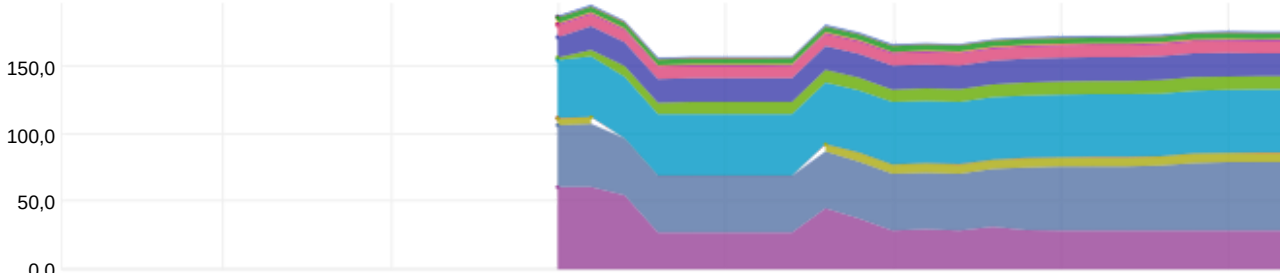
open files сб, 24 апр. 2021 г. | 17:55:00

systemd-network 1,0 _rpc 2,0 systemd-resolve 2,0 messagebus 1,0 syslog 7,0
alertmanager 1,0 grafana 3,0 nodeusr 1,0 prometheus 12,0 daemon 2,0
vagrant 7,0 netdata 44,8 root 32,0

mem

Real memory (RAM) used per user. This does not include shared memory.

Users Real Memory (w/o shared) (users.mem)

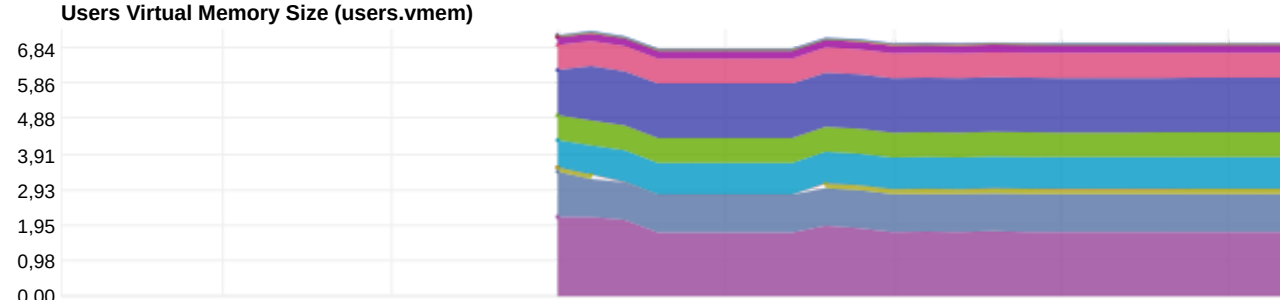


MiB сб, 24 апр. 2021 г. | 17:55:00

systemd-network 0,9 _rpc 0,4 systemd-resolve 4,1 messagebus 0,7 syslog 1,0
alertmanager 8,9 grafana 17,1 nodeusr 10,4 prometheus 47,2 daemon 0,2
vagrant 6,9 netdata 51,5 root 29,0

Virtual memory allocated per user. Please check this article (https://github.com/netdata/netdata/tree/master/daemon#virtual-memory) for more information.

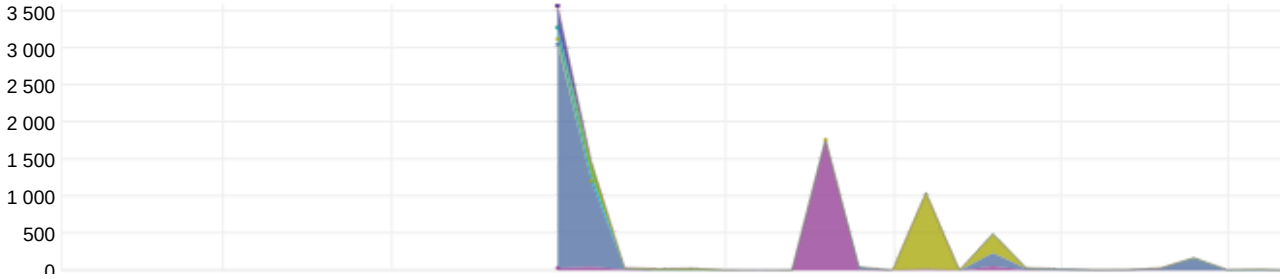
Users Virtual Memory Size (users.vmem)



GiB сб, 24 апр. 2021 г. | 17:55:00

systemd-network 0,03 _rpc 0,01 systemd-resolve 0,02 messagebus 0,01 syslog 0,21
alertmanager 0,69 grafana 1,50 nodeusr 0,69 prometheus 0,87 daemon 0,00
vagrant 0,14 netdata 1,05 root 1,77

Users Minor Page Faults (users.minor_faults)

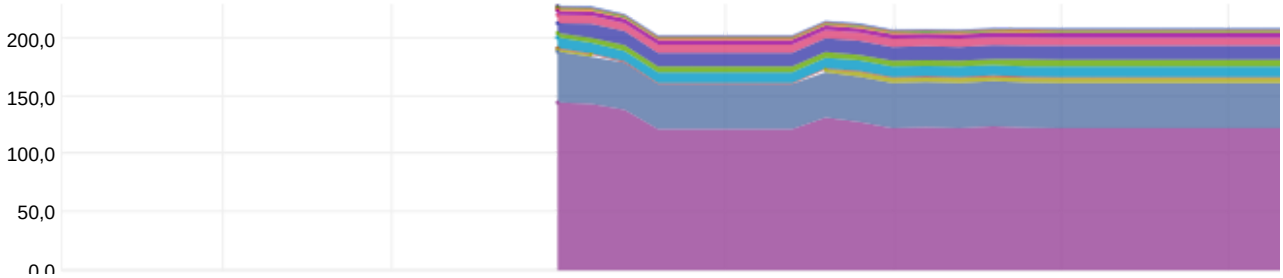


page faults/s сб, 24 апр. 2021 г. | 17:55:00

systemd-resolve 0 messagebus 0 syslog 0 alertmanager 0 grafana 0
nodeusr 1 prometheus 0 vagrant 0 netdata 0 root 0

processes

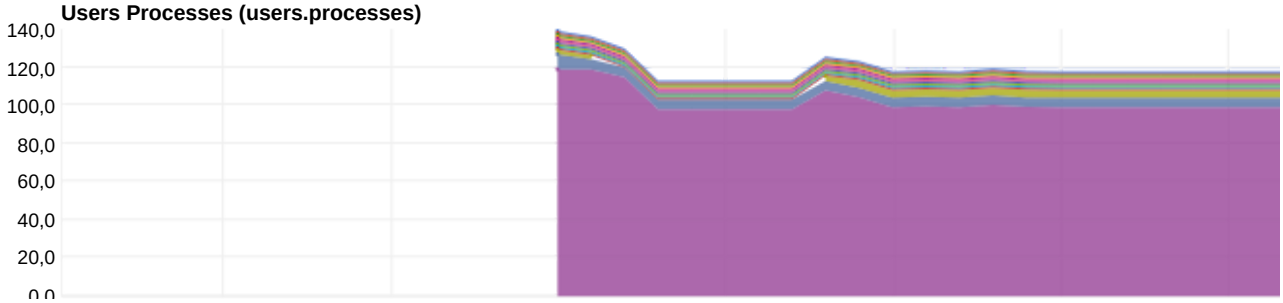
Users Threads (users.threads)



threads сб, 24 апр. 2021 г. | 17:55:00

systemd-network 1,0 _rpc 1,0 systemd-resolve 1,0 messagebus 1,0 syslog 4,0
alertmanager 8,0 grafana 12,0 nodeusr 6,0 prometheus 9,0 daemon 1,0
vagrant 4,0 netdata 39,0 root 123,0

Users Processes (users.processes)

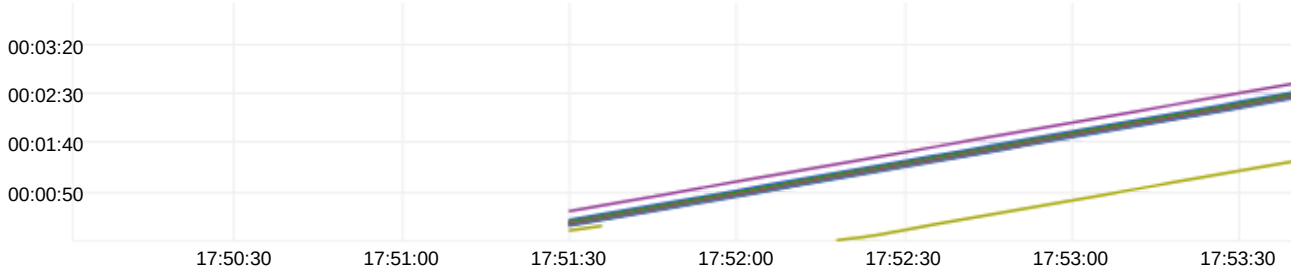


processes сб, 24 апр. 2021 г. | 17:55:00

systemd-network 1,0 _rpc 1,0 systemd-resolve 1,0 messagebus 1,0 syslog 1,0
alertmanager 1,0 grafana 1,0 nodeusr 1,0 prometheus 1,0 daemon 1,0
vagrant 4,0 netdata 5,0 root 99,0

Carried over process group uptime since the Netdata restart. The period of time within which at least one process in the group was running.

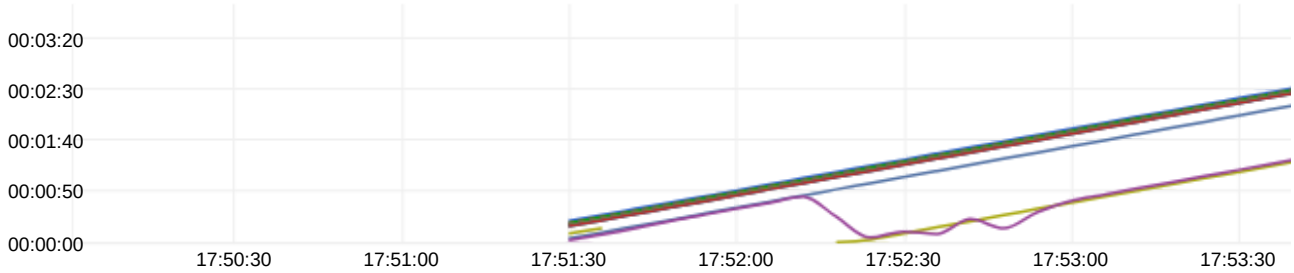
Users Carried Over Uptime (users.uptime)



time сб, 24 апр. 2021 г. | 17:55:00

systemd-network 00:03:53 _rpc 00:03:51 systemd-resolve 00:03:51 messagebus 00:03:49
syslog 00:03:49 alertmanager 00:03:49 grafana 00:03:49 nodeusr 00:03:49 prometheus 00:03:48
daemon 00:03:48 vagrant 00:02:43 netdata 00:03:47 root 00:04:02

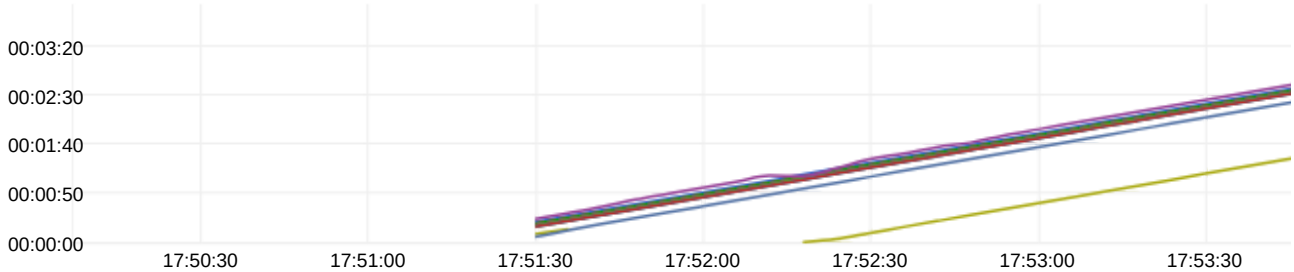
Users Minimum Uptime (users.uptime_min)



time сб, 24 апр. 2021 г. | 17:55:00

systemd-network 00:03:52 _rpc 00:03:50 systemd-resolve 00:03:50 messagebus 00:03:48
syslog 00:03:48 alertmanager 00:03:48 grafana 00:03:48 nodeusr 00:03:48 prometheus 00:03:47
daemon 00:03:47 vagrant 00:02:40 netdata 00:03:35 root 00:02:42

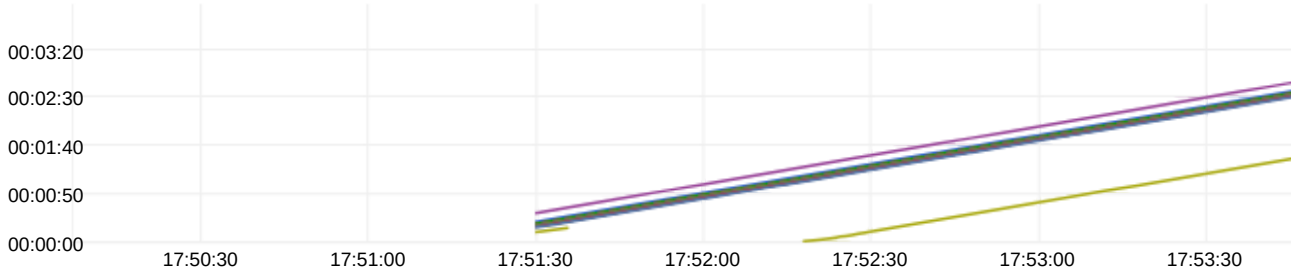
Users Average Uptime (users.uptime_avg)



time сб, 24 апр. 2021 г. | 17:55:06

systemd-network 00:03:57 _rpc 00:03:55 systemd-resolve 00:03:55 messagebus 00:03:53
syslog 00:03:53 alertmanager 00:03:53 grafana 00:03:53 nodeusr 00:03:53 prometheus 00:03:52
daemon 00:03:52 vagrant 00:02:46 netdata 00:03:43 root 00:04:01

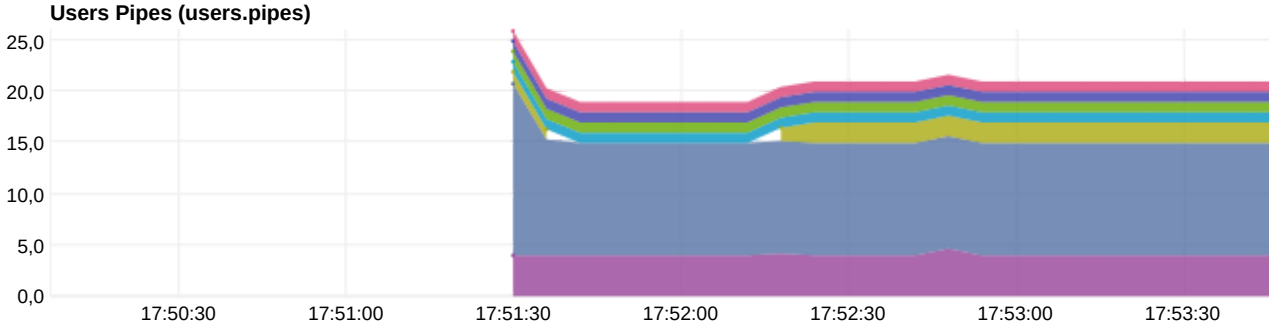
Users Maximum Uptime (users.uptime_max)



time сб, 24 апр. 2021 г. | 17:55:06

systemd-network 00:03:57 _rpc 00:03:55 systemd-resolve 00:03:55 messagebus 00:03:53
syslog 00:03:53 alertmanager 00:03:53 grafana 00:03:53 nodeusr 00:03:53 prometheus 00:03:52
daemon 00:03:52 vagrant 00:02:47 netdata 00:03:51 root 00:04:06

Users Pipes (users.pipes)



open pipes сб, 24 апр. 2021 г. | 17:55:06

alertmanager 1,0 grafana 1,0 nodeusr 1,0 prometheus 1,0 vagrant 2,0
netdata 11,0 root 4,0

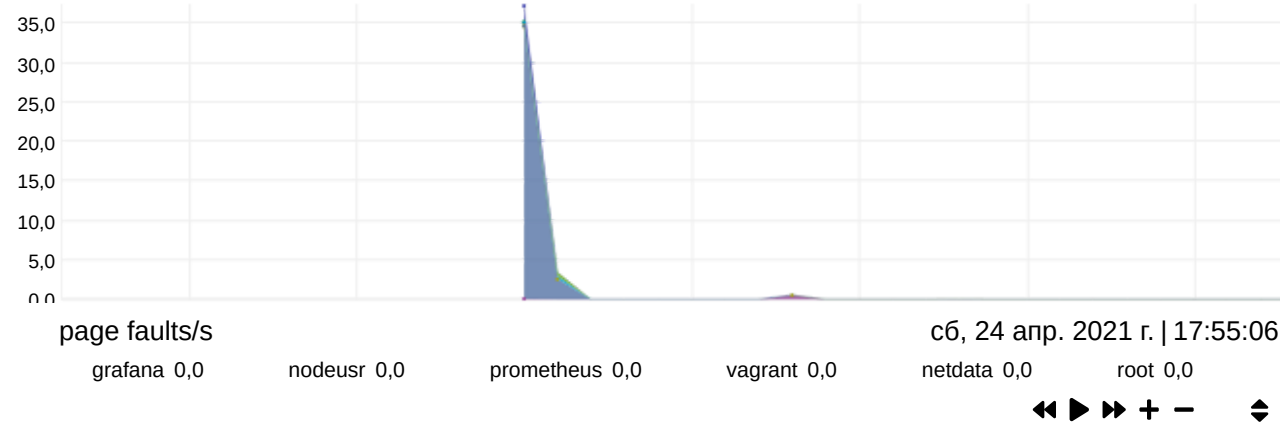
swap

Users Swap Memory (users.swap)

MiB сб, 24 апр. 2021 г. | 17:55:06

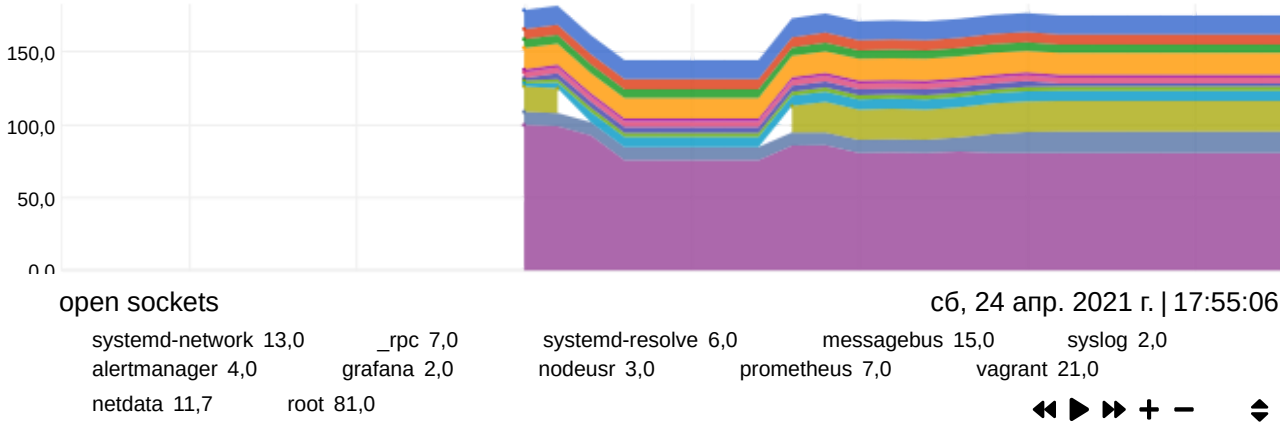
systemd-network 0 _rpc 0 systemd-resolve 0 messagebus 0 syslog 0
alertmanager 0 grafana 0 nodeusr 0 prometheus 0 daemon 0
vagrant 0 netdata 0 root 0

Users Major Page Faults (swap read) (users.major_faults)



net

Users Open Sockets (users.sockets)



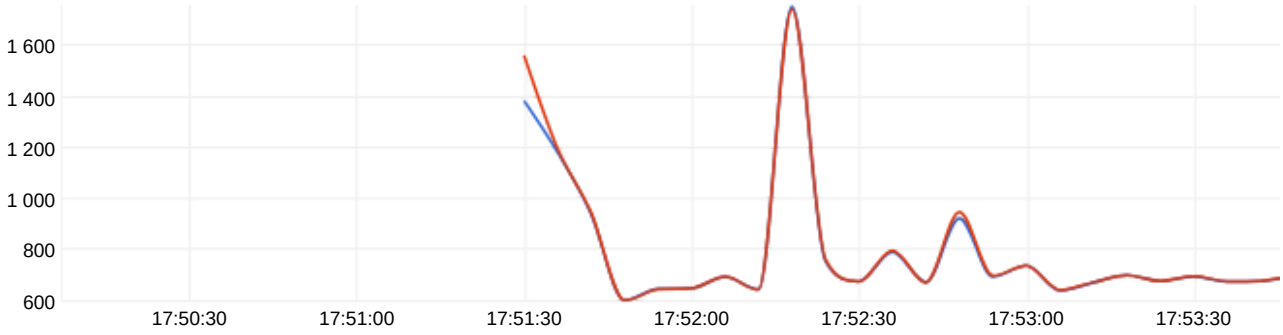
eBPF

Monitor system calls, internal functions, bytes read, bytes written and errors using `eBPF` .

File

Calls for internal functions on Linux kernel. The open dimension is attached to the kernel internal function `do_sys_open` (For kernels newer than `5.5.19` we add a kprobe to `do_sys_openat2`), which is the common function called from open(2) (https://www.man7.org/linux/man-pages/man2/open.2.html) and openat(2) (https://www.man7.org/linux/man-pages/man2/openat.2.html). The close dimension is attached to the function `__close_fd` or `close_fd` according to your kernel version, which is called from system call close(2) (https://www.man7.org/linux/man-pages/man2/close.2.html).

Open and close calls (ebpf.file_descriptor)



calls/s сб, 24 апр. 2021 г. | 17:55:06

open 633 close 634

VFS

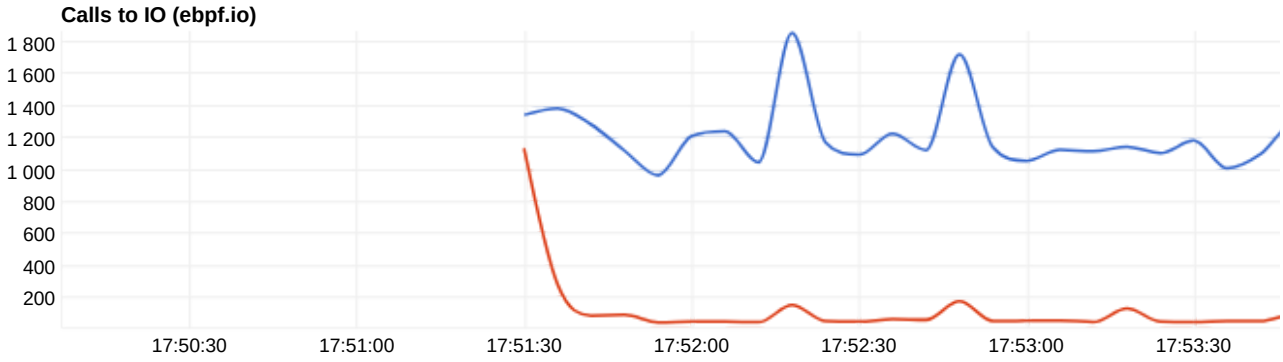
This chart does not show all events that remove files from the file system, because file systems can create their own functions to remove files, it shows calls for the function vfs_unlink (https://www.kernel.org/doc/htmldocs/filesystems/API-vfs-unlink.html).

Remove files (ebpf.deleted_objects)

calls/s сб, 24 апр. 2021 г. | 17:55:06

delete 0,00

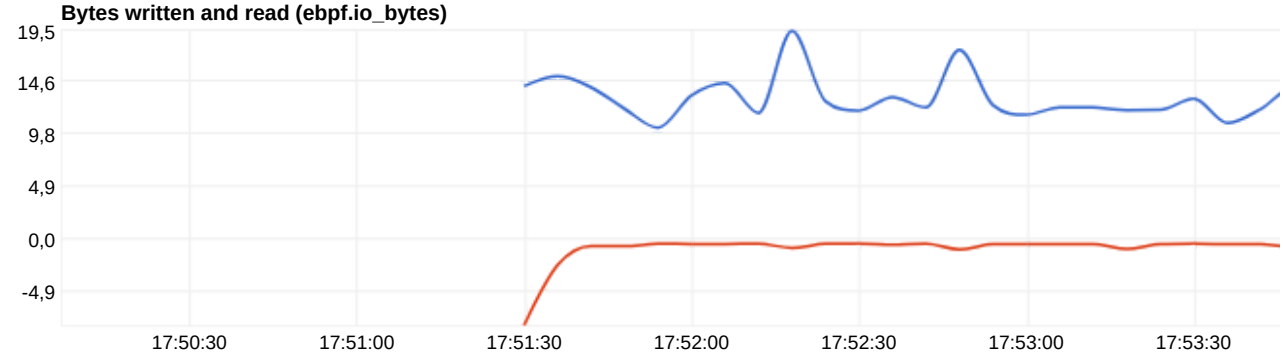
Successful or failed calls to functions vfs_read (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html) and vfs_write (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html). This chart may not show all file system events if it uses other functions to store data on disk.



calls/s сб, 24 апр. 2021 г. | 17:55:06

read 1 180 write 56

Total of bytes read or written with success using the functions vfs_read (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html) and vfs_write (https://topic.alibabacloud.com/a/kernel-state-file-operation-__-work-information-kernel_8_8_20287135.html).



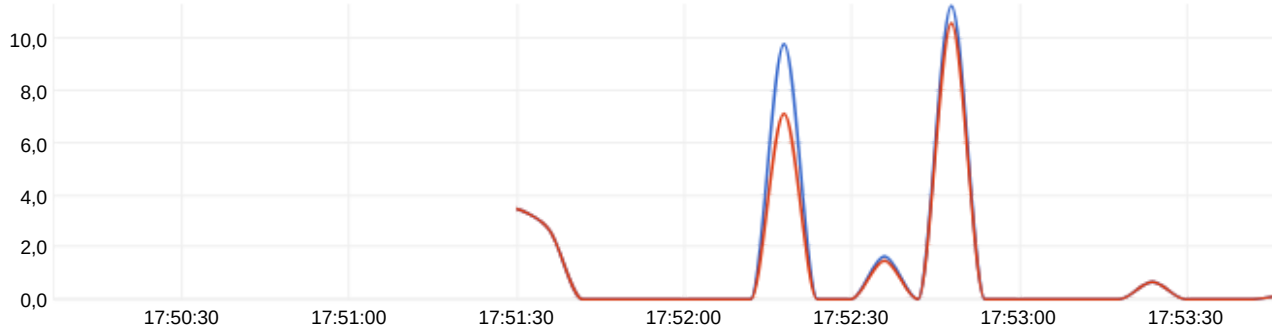
kilobytes/s сб, 24 апр. 2021 г. | 17:55:06

read 12,9 write -0,4

Process

Number of times that either do_fork (https://www.ece.uic.edu/~yshi1/linux/lkse/node4.html#SECTION00421000000000000000), or `kernel_clone` if you are running kernel newer than 5.9.16, is called to create a new task, which is the common name used to define process and tasks inside the kernel. Netdata identifies the threads by counting the number of calls for sys_clone (https://linux.die.net/man/2/clone) that has the flag `CLONE_THREAD` set.

Start process (ebpf.process_thread)

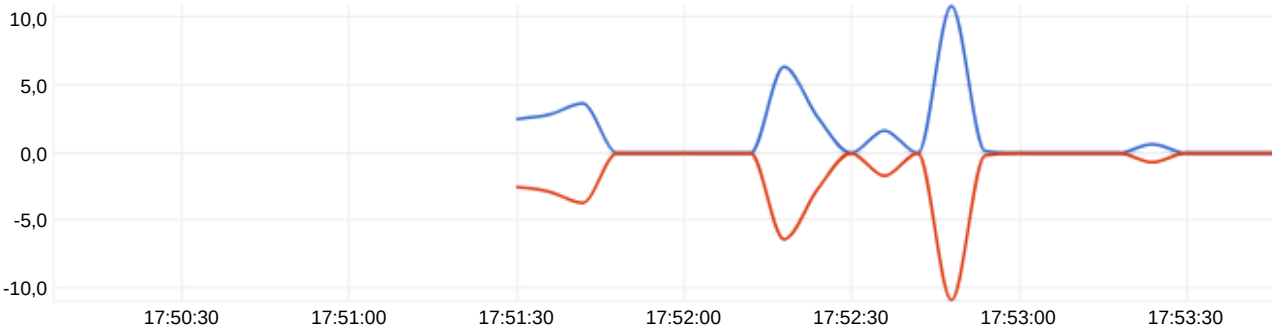


calls/s сб, 24 апр. 2021 г. | 17:55:06

process 0,0 thread 0,0

Calls for the functions responsible for closing (do_exit (https://www.informit.com/articles/article.aspx?p=370047&seqNum=4)) and releasing (release_task (https://www.informit.com/articles/article.aspx?p=370047&seqNum=4)) tasks.

Exit process (ebpf.exit)



calls/s сб, 24 апр. 2021 г. | 17:55:06

process 0,0 task 0,0

Difference between the number of process created and the number of threads created per period(`process` dimension), it also shows the number of possible zombie process running on system.

Process not closed (ebpf.process_status)

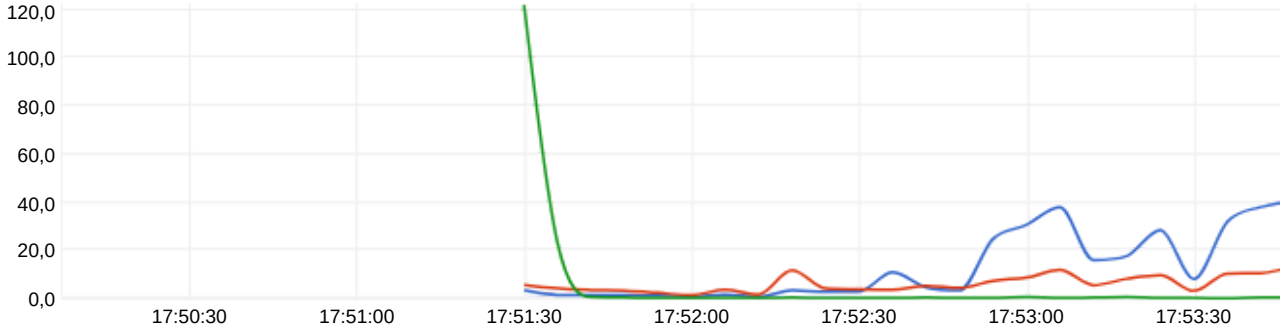
difference сб, 24 апр. 2021 г. | 17:55:06

process 0,00

Socket

Successful or failed calls to functions `tcp_sendmsg` , `tcp_cleanup_rbuf` and `tcp_close` .

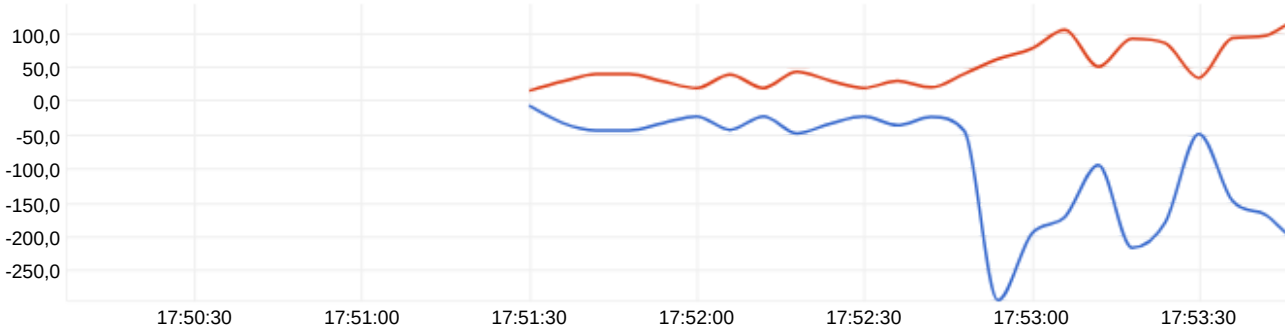
Calls to internal functions (ebpf.tcp_functions)



calls/s сб, 24 апр. 2021 г. | 17:55:06

sent 18,0 received 6,7 close 0,5

TCP bandwidth (ebpf.total_tcp_bandwidth)

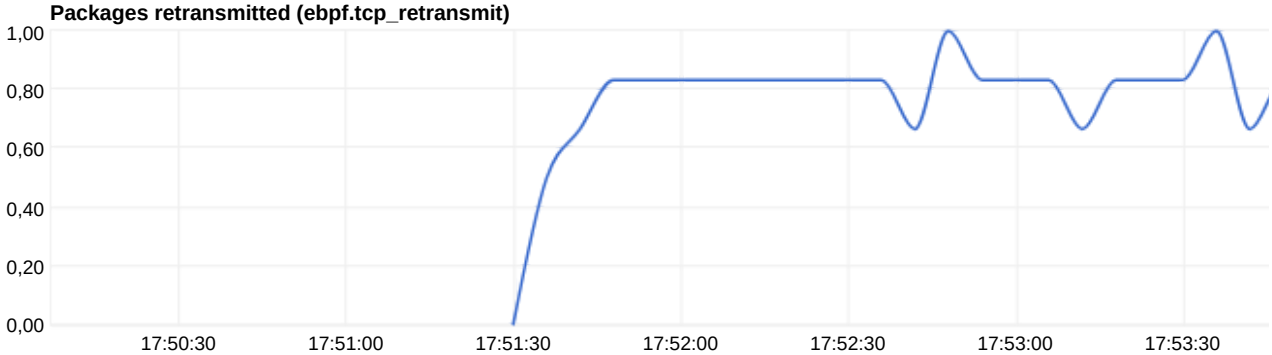


kilobits/s сб, 24 апр. 2021 г. | 17:55:06

sent -78,5 received 64,7

Number of packets retransmitted for function `tcp_retransmit_skb` .

Packages retransmitted (ebpf.tcp_retransmit)

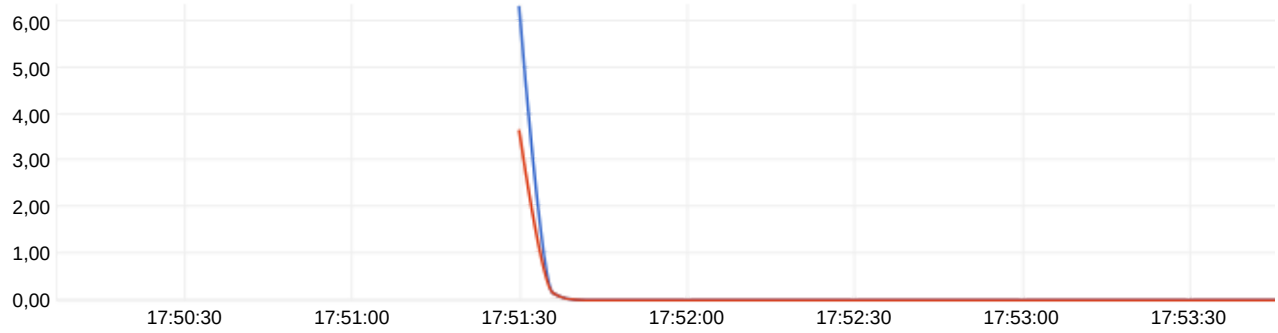


calls/s сб, 24 апр. 2021 г. | 17:55:06

retransmitted 0,83

Successful or failed calls to functions `udp_sendmsg` and `udp_recvmsg`.

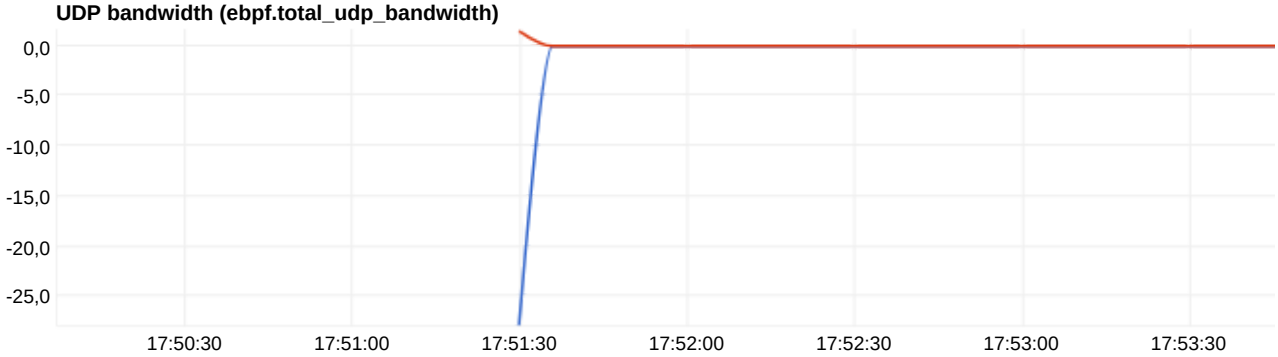
UDP calls (ebpf.udp_functions)



calls/s сб, 24 апр. 2021 г. | 17:55:06

sent 0,00 received 0,00

UDP bandwidth (ebpf.total_udp_bandwidth)



kilobits/s сб, 24 апр. 2021 г. | 17:55:06

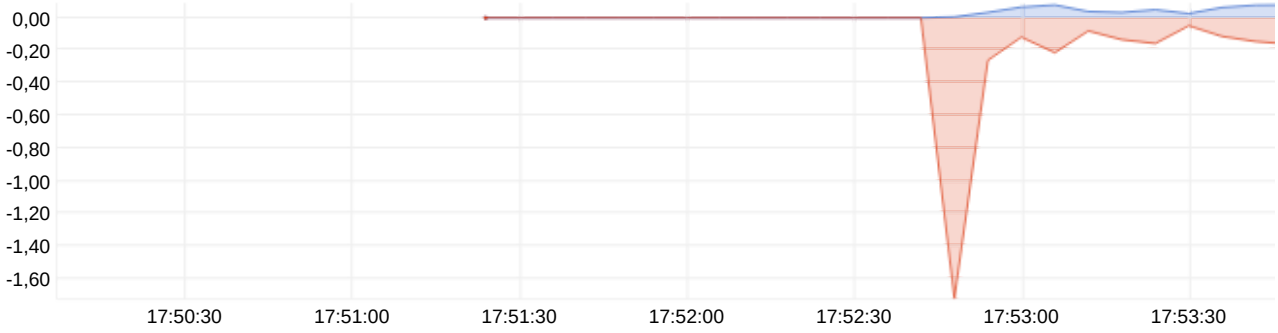
sent 0,0 received 0,0

Netdata Monitoring

Performance metrics for the operation of netdata itself and its plugins.

netdata

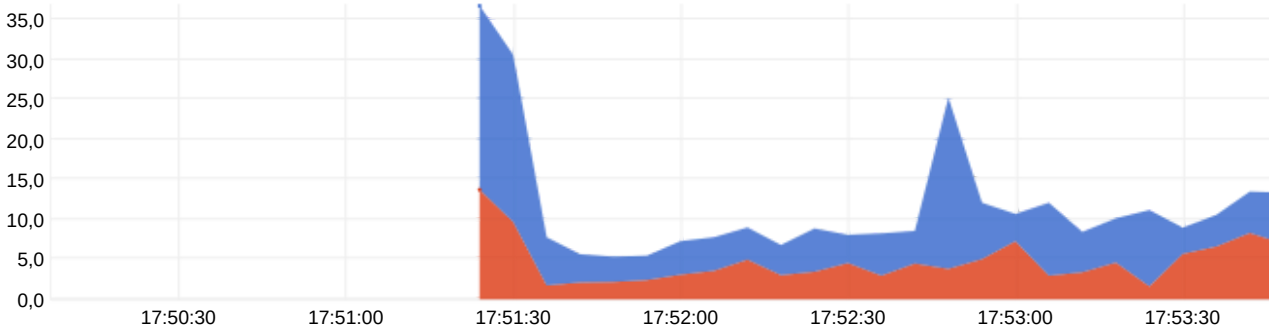
Netdata Network Traffic (netdata.net)



megabits/s сб, 24 апр. 2021 г. | 17:55:06

in 0,04 out -0,05

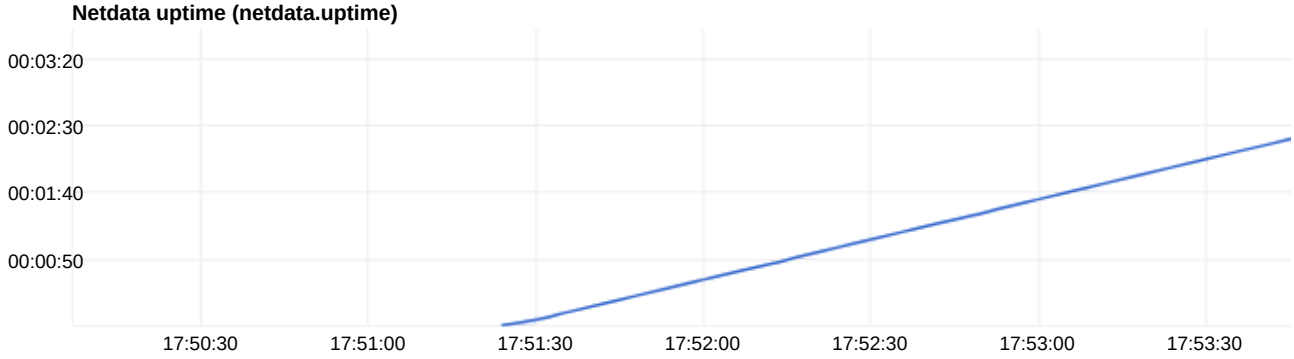
Netdata CPU usage (netdata.server_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:06

user 4,1 system 3,6

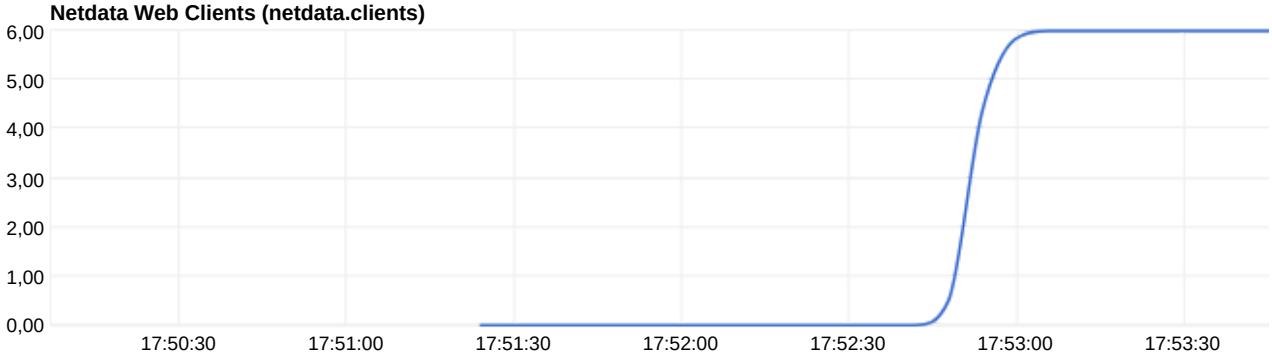
Netdata uptime (netdata.uptime)



time сб, 24 апр. 2021 г. | 17:55:06

uptime 00:03:42

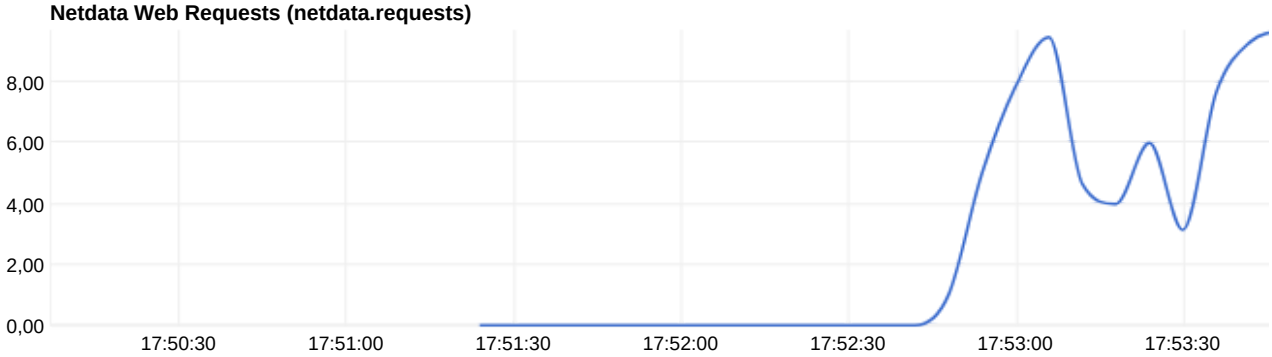
Netdata Web Clients (netdata.clients)



connected clients сб, 24 апр. 2021 г. | 17:55:06

clients 2,50

Netdata Web Requests (netdata.requests)

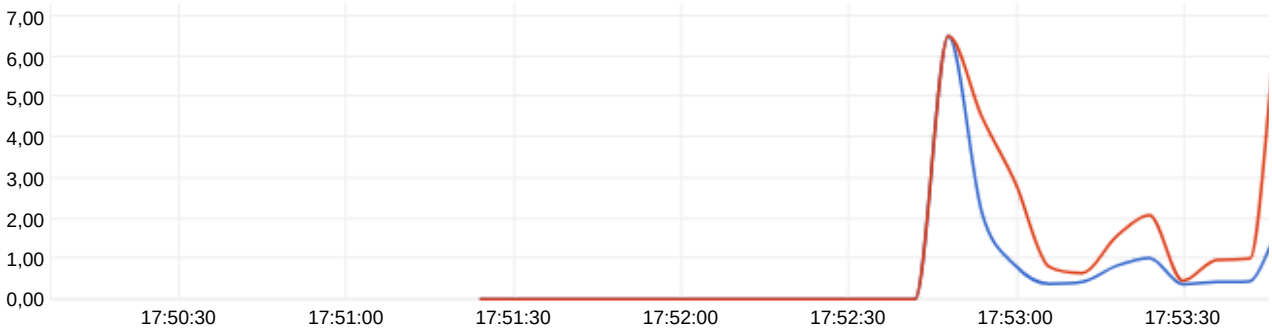


requests/s сб, 24 апр. 2021 г. | 17:55:06

requests 4,33

The netdata API response time measures the time netdata needed to serve requests. This time includes everything, from the reception of the first byte of a request, to the dispatch of the last byte of its reply, therefore it includes all network latencies involved (i.e. a client over a slow network will influence these metrics).

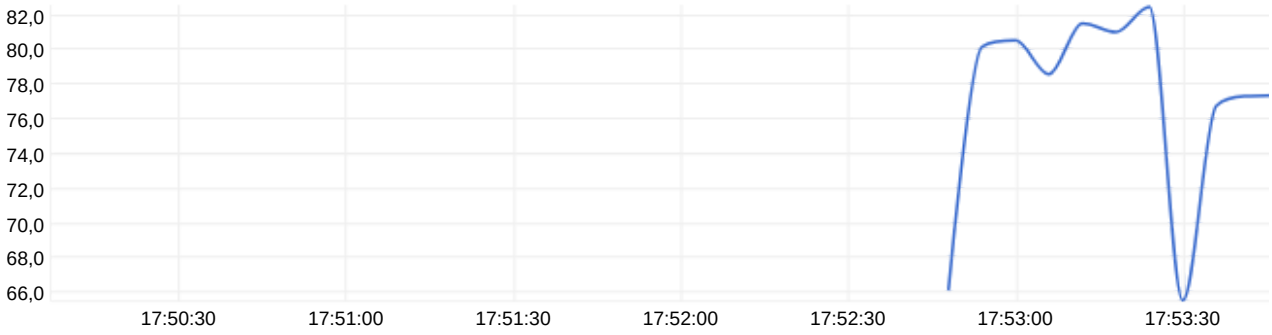
Netdata API Response Time (netdata.response_time)



milliseconds/request сб, 24 апр. 2021 г. | 17:55:06

average 0,51 max 0,64

Netdata API Responses Compression Savings Ratio (netdata.compression_ratio)

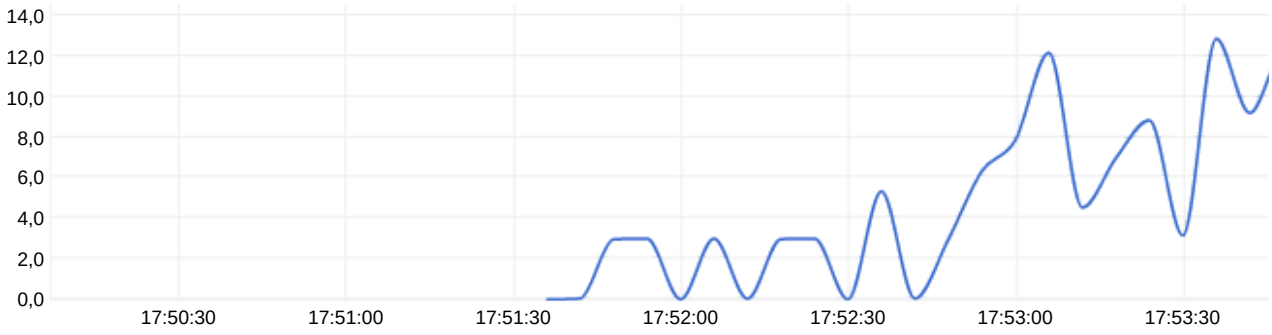


percentage сб, 24 апр. 2021 г. | 17:55:06

savings 78,9

queries

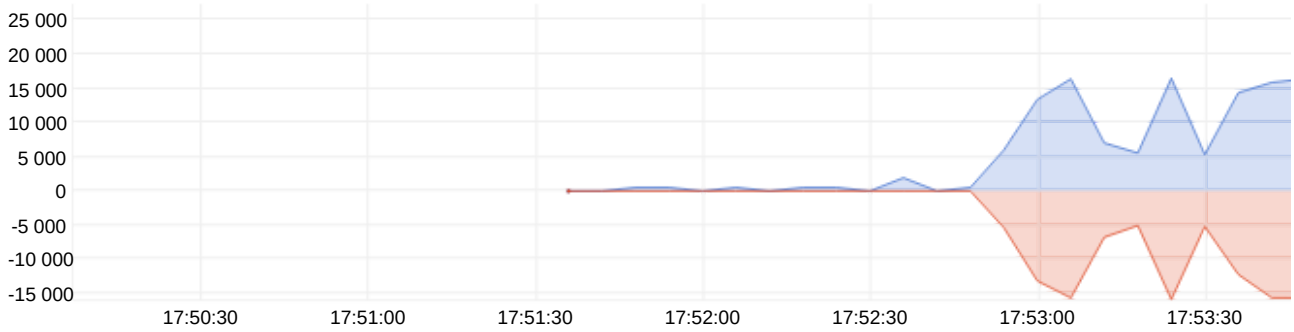
Netdata API Queries (netdata.queries)



queries/s сб, 24 апр. 2021 г. | 17:55:06

queries 7,0

Netdata API Points (netdata.db_points)

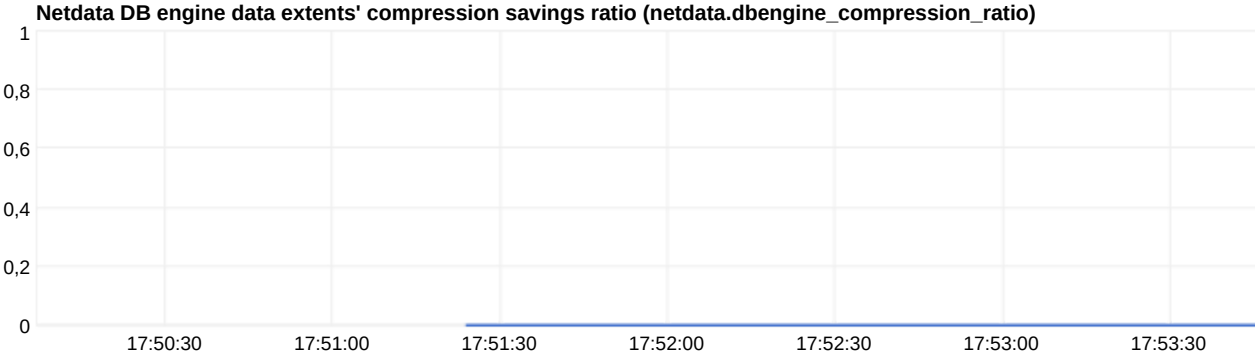


points/s сб, 24 апр. 2021 г. | 17:55:06

read 16 099 generated -2 602

dbengine

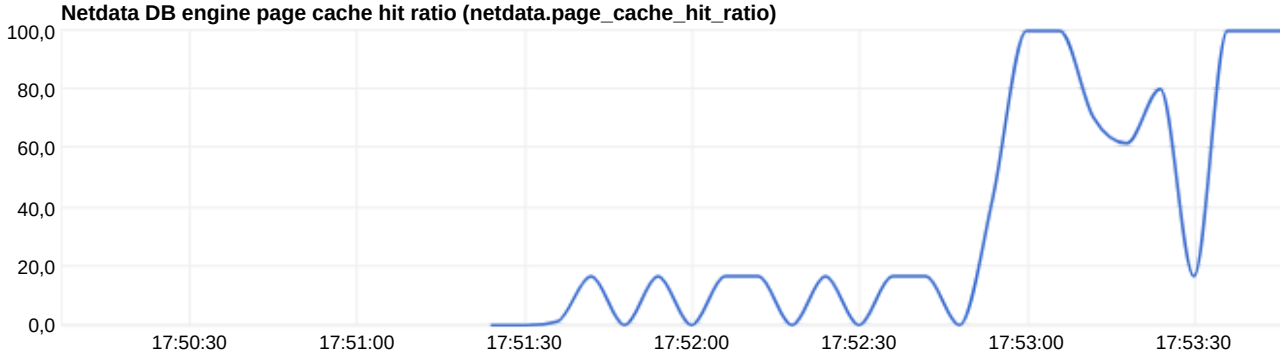
Netdata DB engine data extents' compression savings ratio (netdata.dbengine_compression_ratio)



percentage сб, 24 апр. 2021 г. | 17:55:06

savings 0

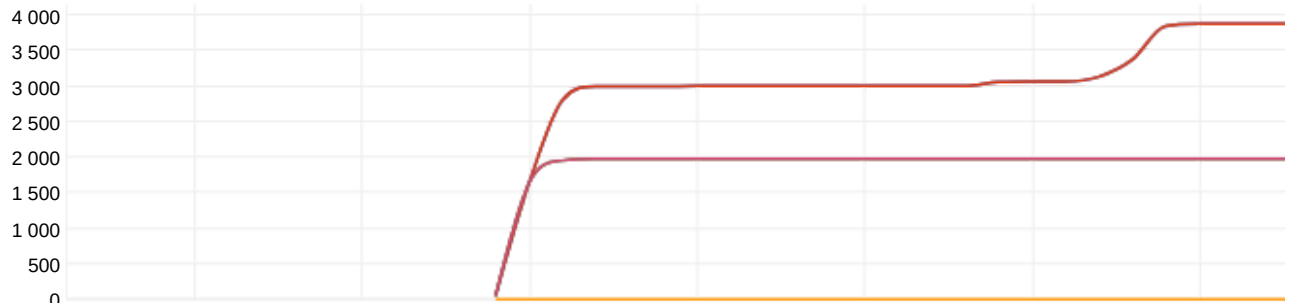
Netdata DB engine page cache hit ratio (netdata.page_cache_hit_ratio)



percentage сб, 24 апр. 2021 г. | 17:55:06

ratio 100,0

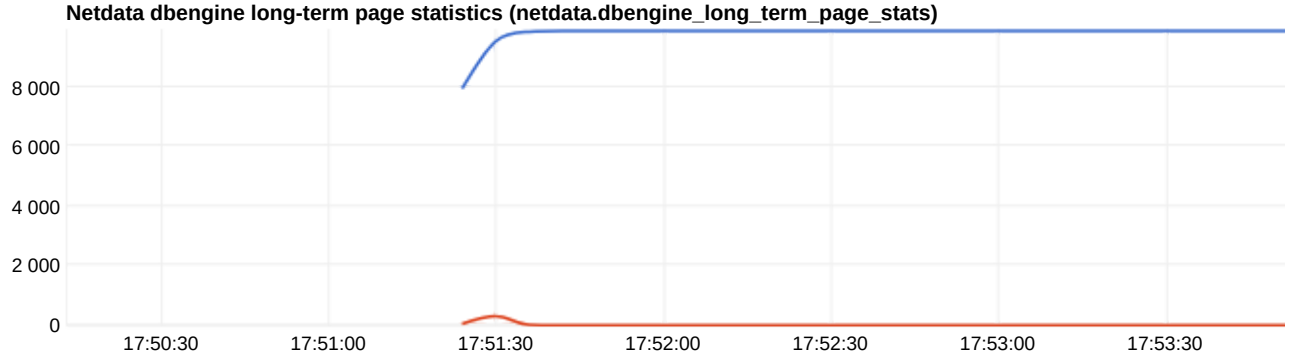
Netdata dbengine page cache statistics (netdata.page_cache_stats)



pages сб, 24 апр. 2021 г. | 17:55:06

descriptors 4 142 populated 4 142 dirty 1 988 backfills 0 used_by_collectors 1 988

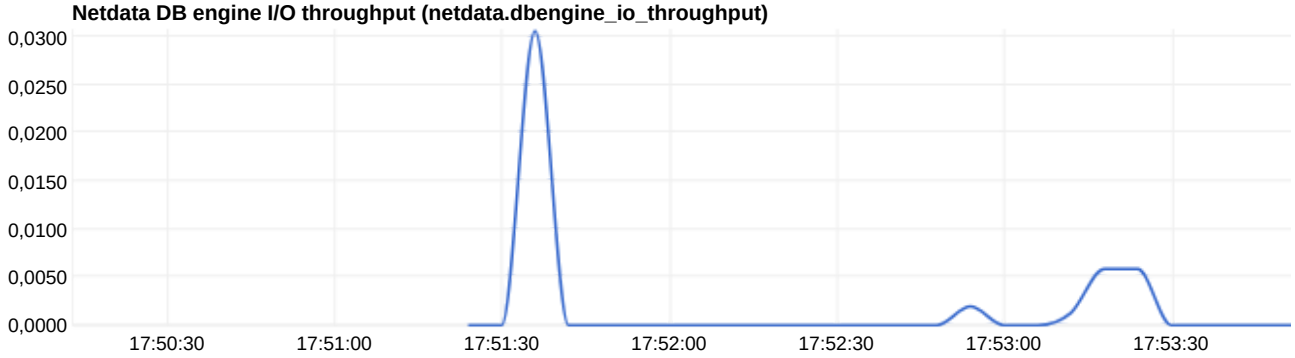
Netdata dbengine long-term page statistics (netdata.dbengine_long_term_page_stats)



pages сб, 24 апр. 2021 г. | 17:55:12

total 9 883 insertions 0

Netdata DB engine I/O throughput (netdata.dbengine_io_throughput)



MiB/s сб, 24 апр. 2021 г. | 17:55:12

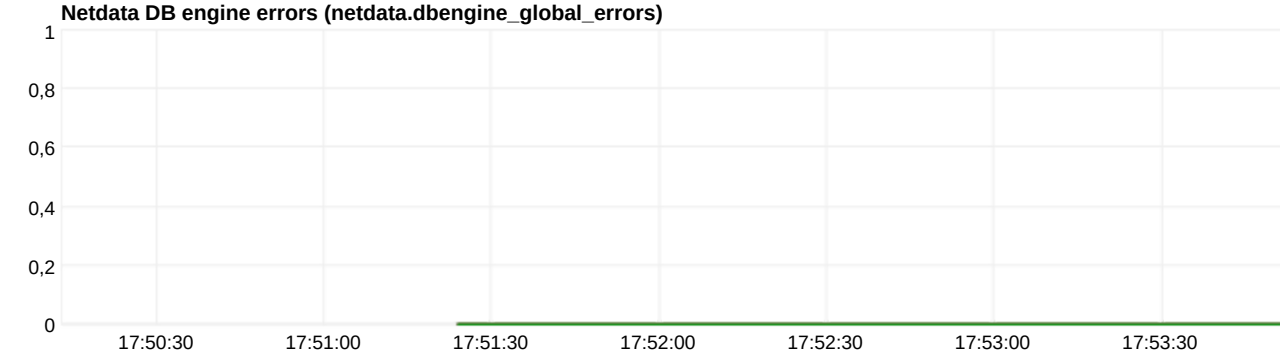
reads 0,0000

Netdata DB engine I/O operations (netdata.dbengine_io_operations)

operations/s сб, 24 апр. 2021 г. | 17:55:12

reads 0,00

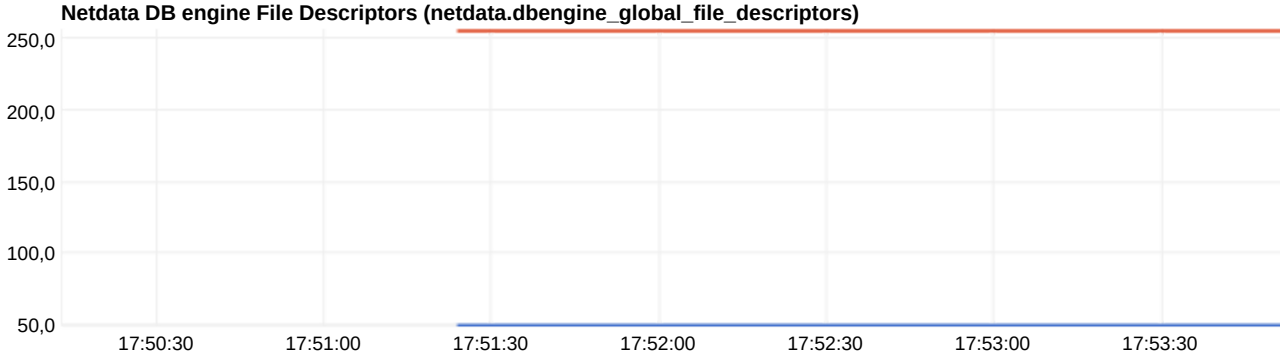
Netdata DB engine errors (netdata.dbengine_global_errors)



errors/s сб, 24 апр. 2021 г. | 17:55:12

io_errors 0 fs_errors 0 pg_cache_over_half_dirty_events 0

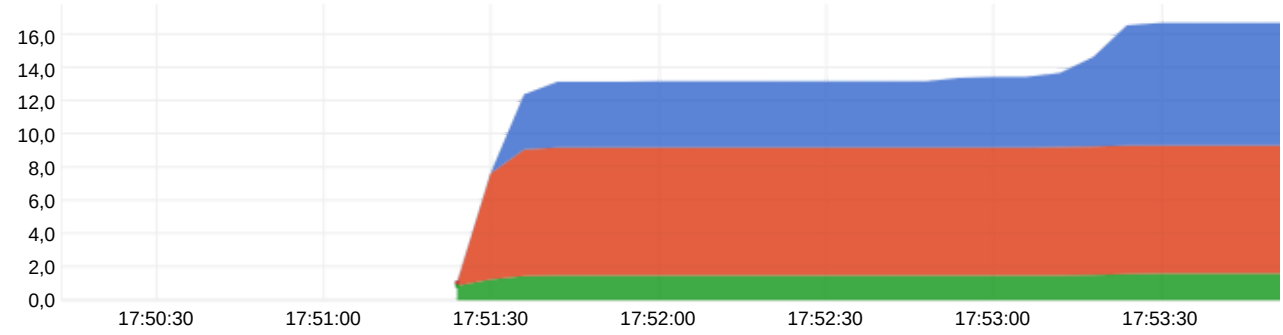
Netdata DB engine File Descriptors (netdata.dbengine_global_file_descriptors)



descriptors сб, 24 апр. 2021 г. | 17:55:12

current 50,0 max 256,0

Netdata DB engine RAM usage (netdata.dbengine_ram)

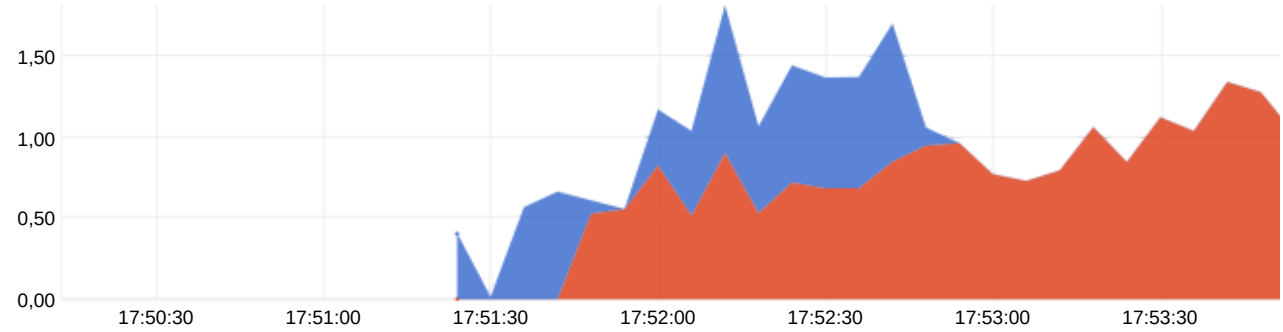


MiB сб, 24 апр. 2021 г. | 17:55:12

cache 0,6 collectors 15,5 metadata 1,6

cgroups

Netdata CGroups Plugin CPU usage (netdata.plugin_cgroups_cpu)

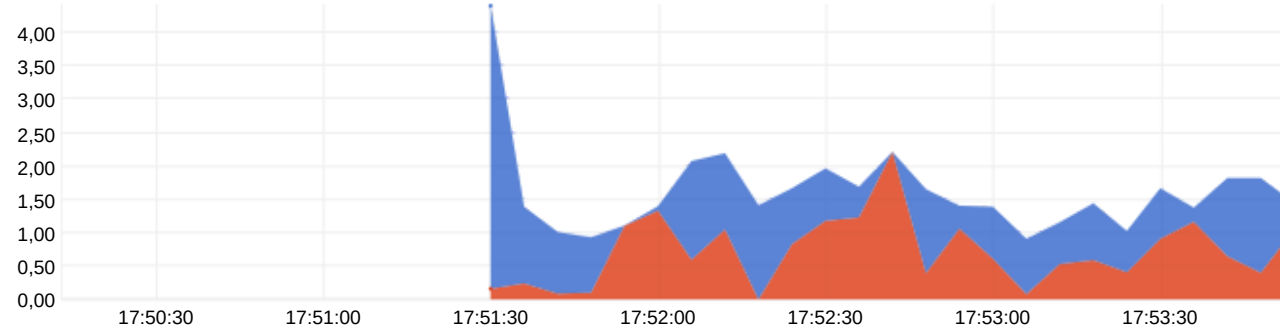


milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,16 system 0,37

proc

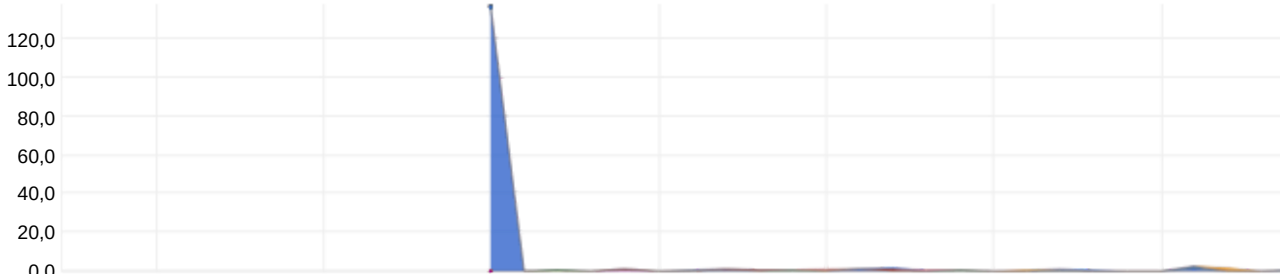
Netdata proc plugin CPU usage (netdata.plugin_proc_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,53 system 0,38

Netdata proc plugin modules durations (netdata.plugin_proc_modules)

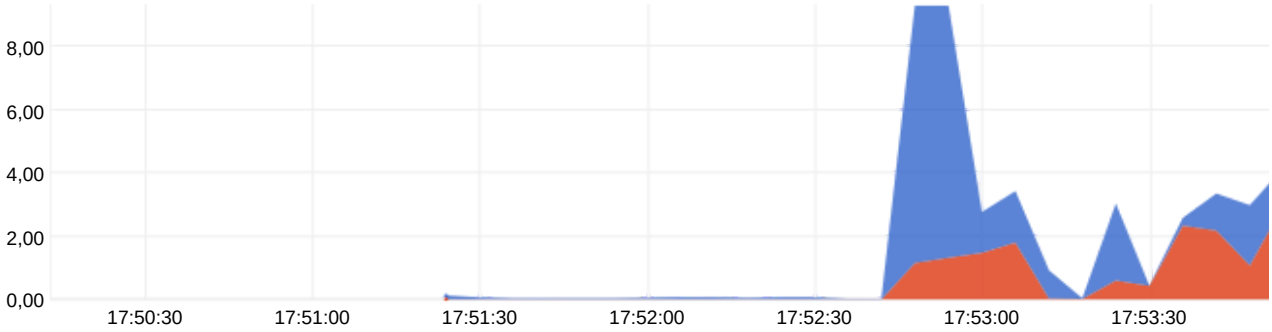


milliseconds/run сб, 24 апр. 2021 г. | 17:55:12

stat 0,0 entropy 0,0 interrupts 0,0 meminfo 0,0 netdev 0,0 netstat 0,7
snmp 0,0 snmp6 0,0 softnet 0,0 diskstats 0,0 btrfs 0,0 ipc 0,0
power_supply 0,0

web

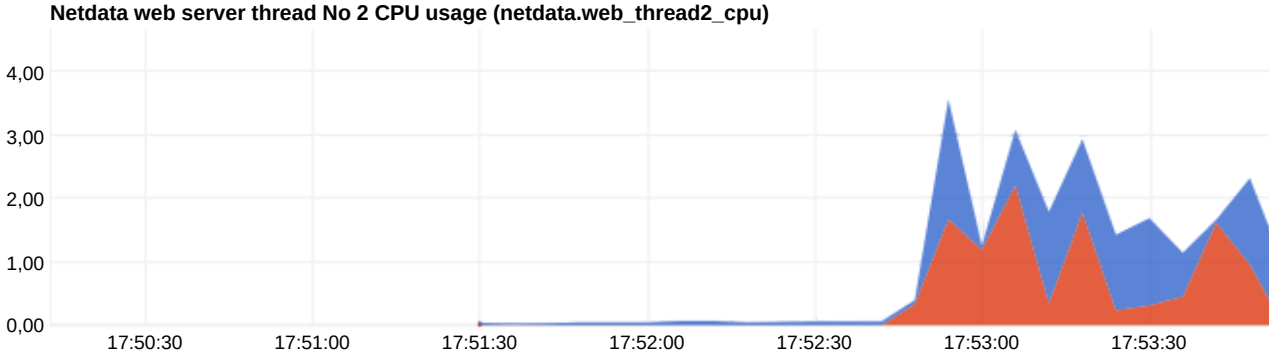
Netdata web server thread No 1 CPU usage (netdata.web_thread1_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,04 system 1,76

Netdata web server thread No 2 CPU usage (netdata.web_thread2_cpu)

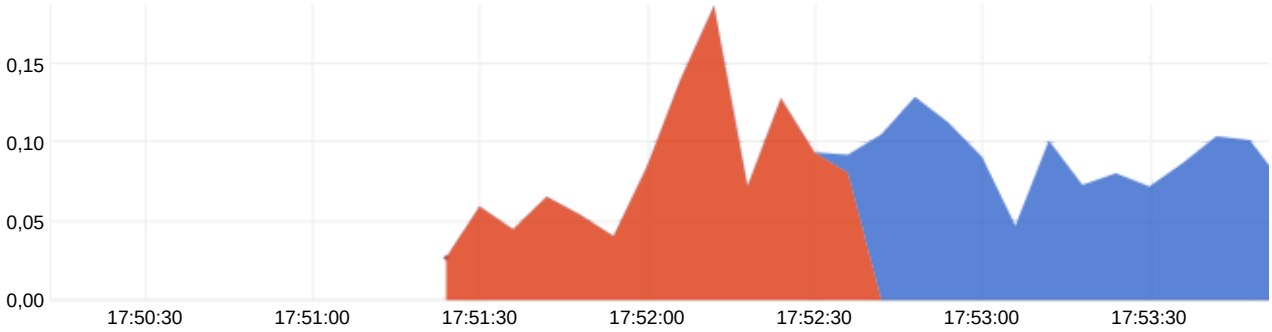


milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,02 system 0,02

statsd

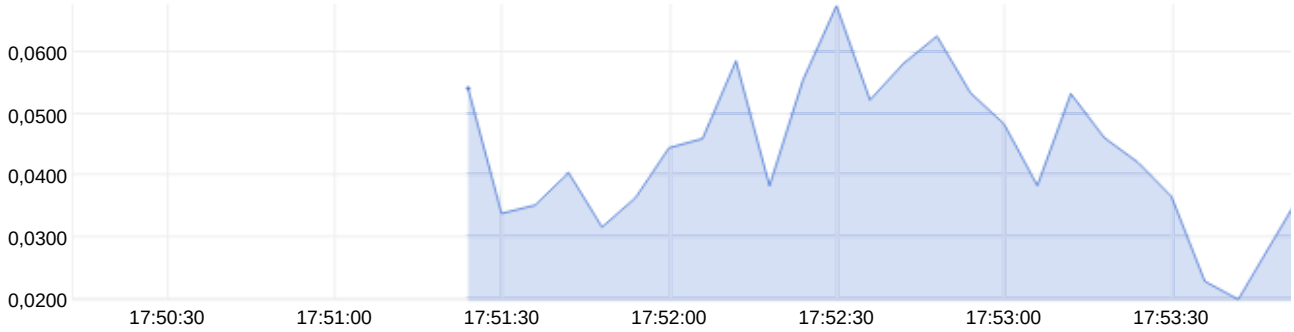
Netdata statsd charting thread CPU usage (netdata.plugin_statsd_charting_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,06 system 0,00

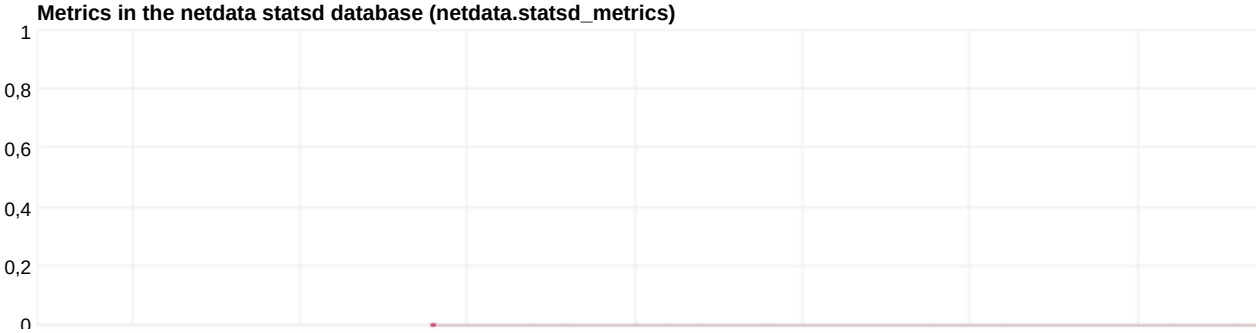
NetData statsd collector thread No 1 CPU usage (netdata.plugin_statsd_collector1_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,0260

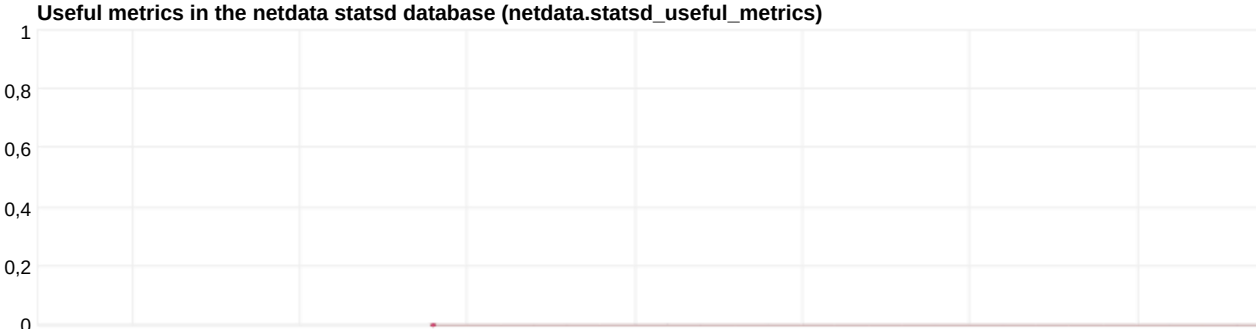
Metrics in the netdata statsd database (netdata.statsd_metrics)



metrics сб, 24 апр. 2021 г. | 17:55:12

gauges 0 counters 0 timers 0 meters 0 histograms 0 sets 0

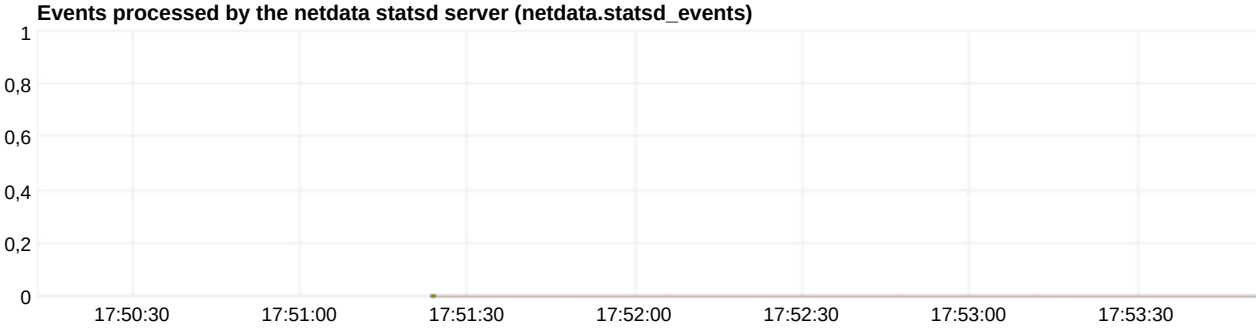
Useful metrics in the netdata statsd database (netdata.statsd_useful_metrics)



metrics сб, 24 апр. 2021 г. | 17:55:12

gauges 0 counters 0 timers 0 meters 0 histograms 0 sets 0

Events processed by the netdata statsd server (netdata.statsd_events)

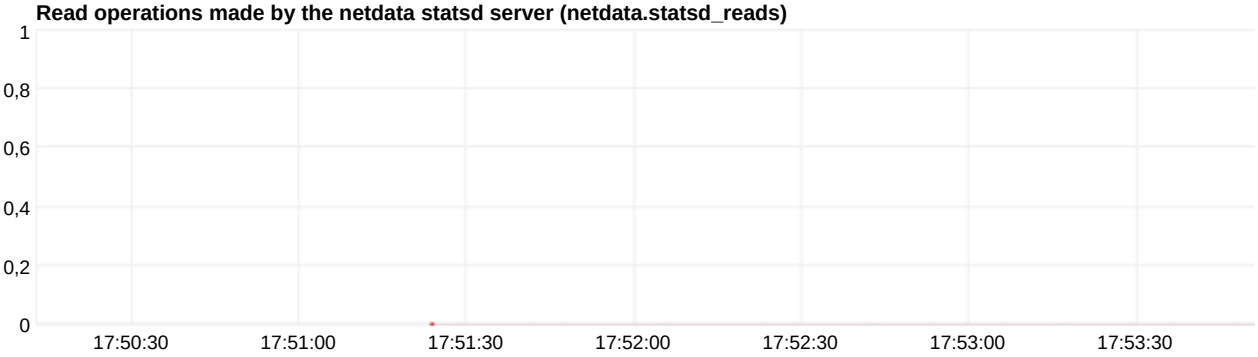


events/s сб, 24 апр. 2021 г. | 17:55:12

gauges 0 counters 0 timers 0 meters 0 histograms 0 sets 0

unknown 0 errors 0

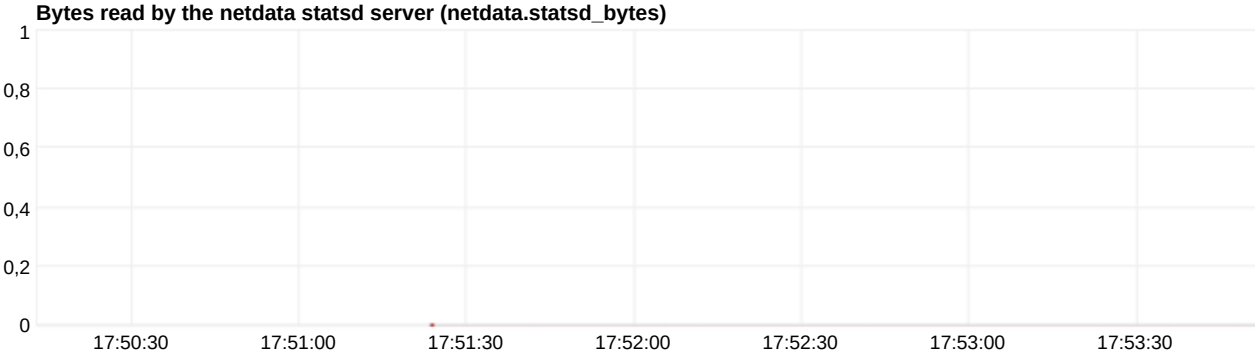
Read operations made by the netdata statsd server (netdata.statsd_reads)



reads/s сб, 24 апр. 2021 г. | 17:55:12

tcp 0 udp 0

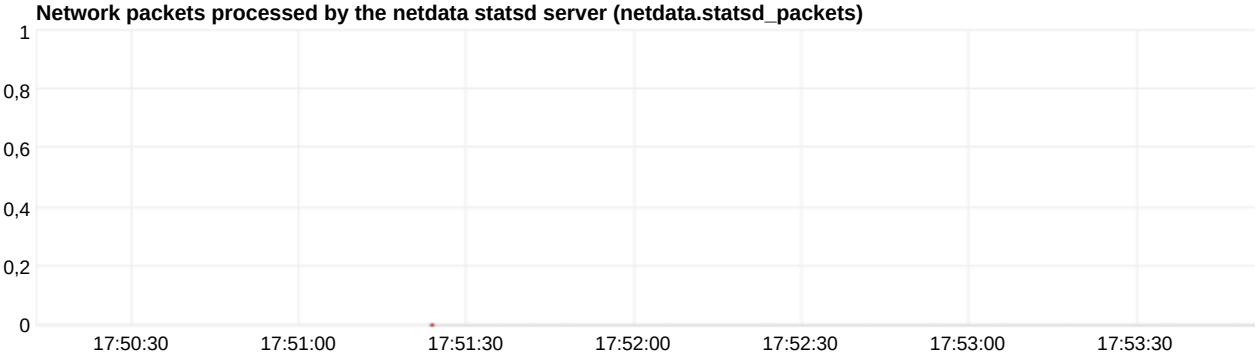
Bytes read by the netdata statsd server (netdata.statsd_bytes)



kilobits/s сб, 24 апр. 2021 г. | 17:55:12

tcp 0 udp 0

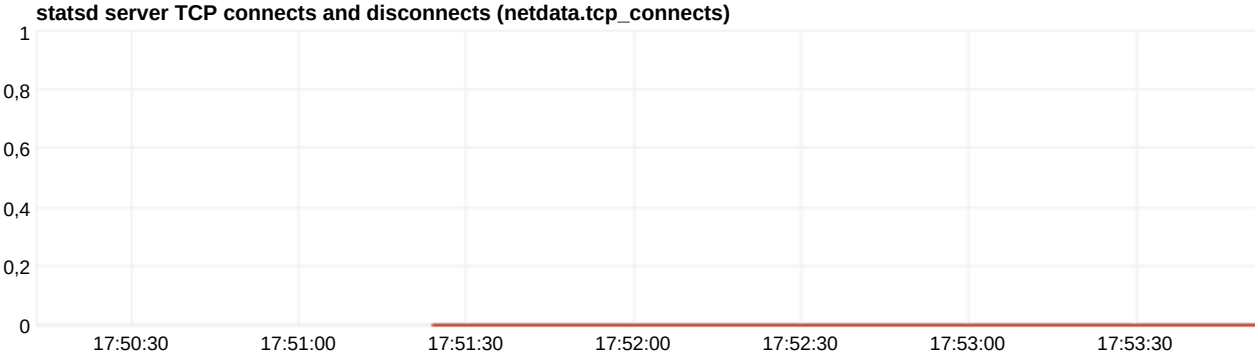
Network packets processed by the netdata statsd server (netdata.statsd_packets)



pps сб, 24 апр. 2021 г. | 17:55:12

tcp 0 udp 0

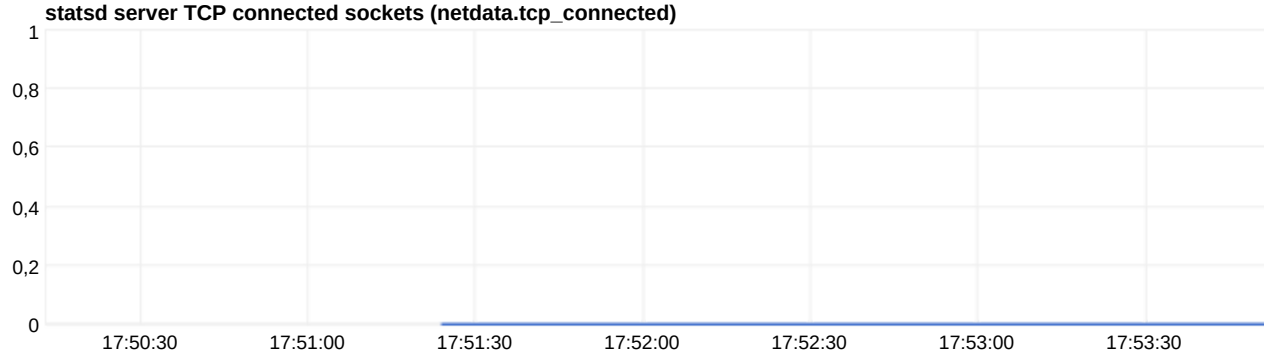
statsd server TCP connects and disconnects (netdata.tcp_connects)



events сб, 24 апр. 2021 г. | 17:55:12

connects 0 disconnects 0

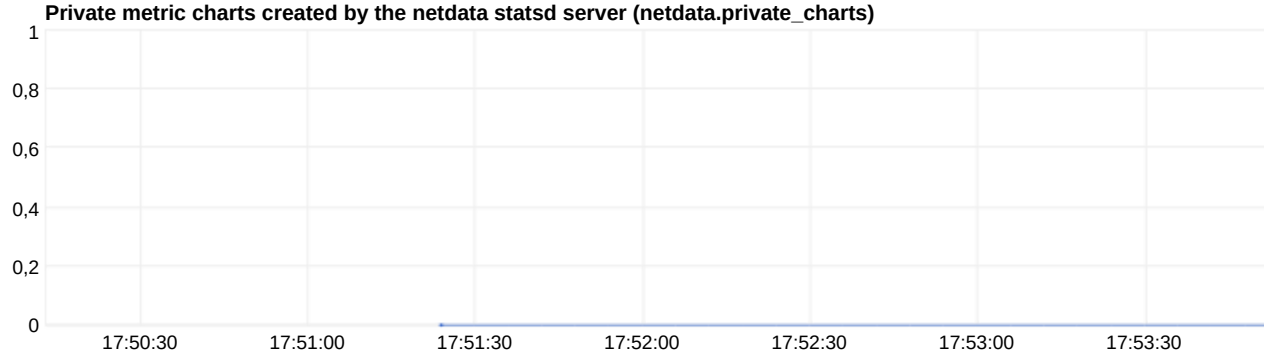
statsd server TCP connected sockets (netdata.tcp_connected)



sockets сб, 24 апр. 2021 г. | 17:55:12

connected 0

Private metric charts created by the netdata statsd server (netdata.private_charts)

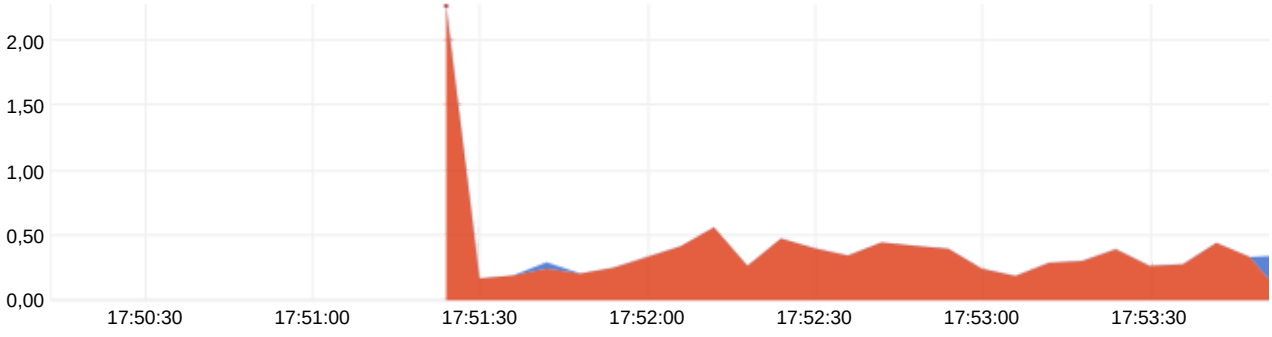


charts сб, 24 апр. 2021 г. | 17:55:12

charts 0

diskspace

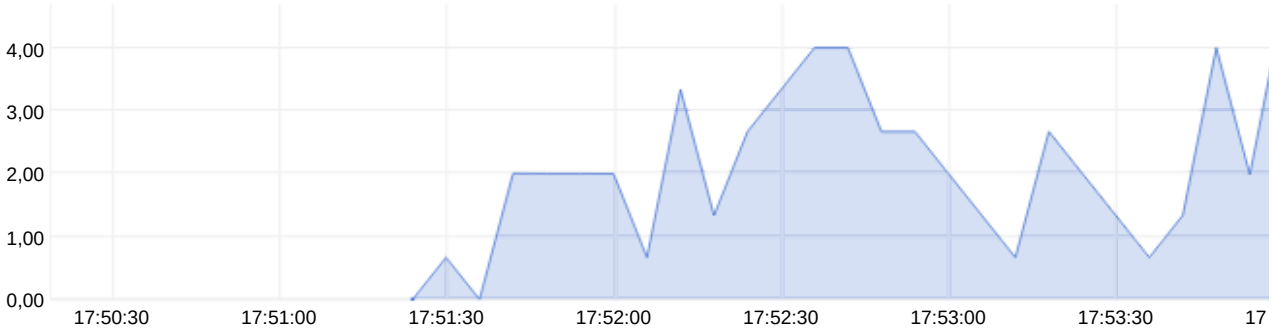
Netdata Disk Space Plugin CPU usage (netdata.plugin_diskspace)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:12

user 0,00 system 0,23

Netdata Disk Space Plugin Duration (netdata.plugin_diskspace_dt)



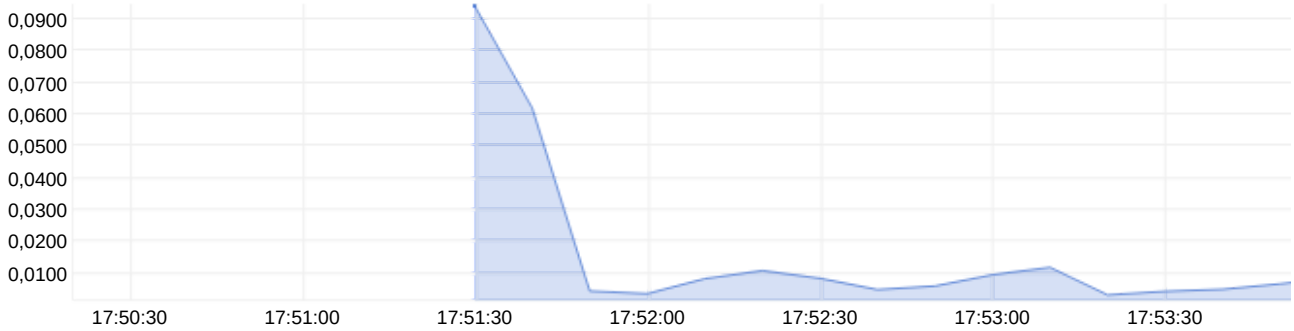
milliseconds/run

сб, 24 апр. 2021 г. | 17:55:18

duration 0,00

timex

Netdata Timex Plugin CPU usage (netdata.plugin_timex)

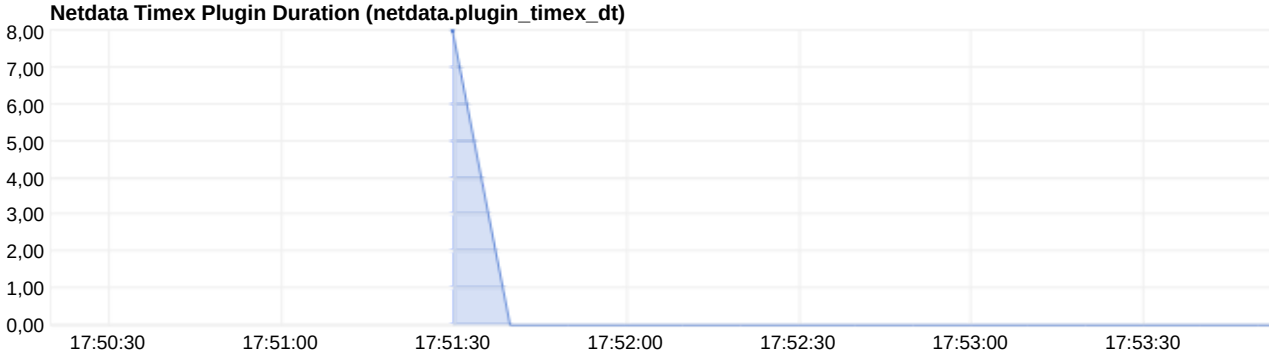


milliseconds/s

сб, 24 апр. 2021 г. | 17:55:10

user 0,0044

Netdata Timex Plugin Duration (netdata.plugin_timex_dt)



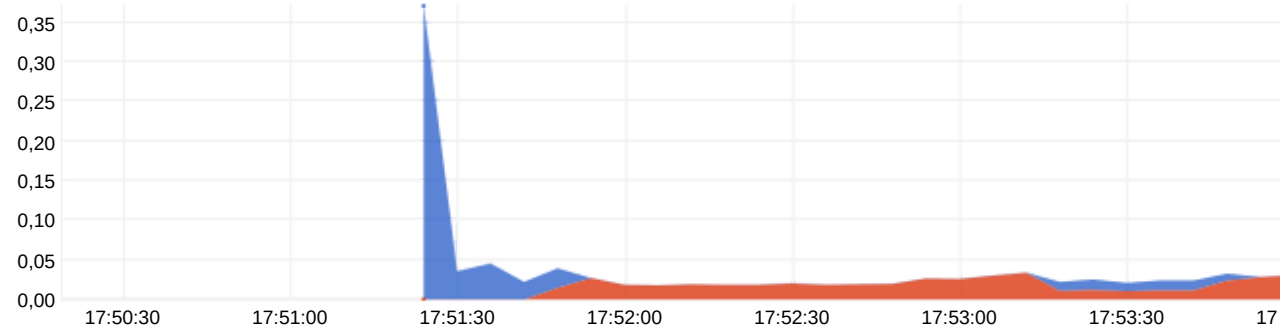
milliseconds/run

сб, 24 апр. 2021 г. | 17:55:10

duration 0,00

tc.helper

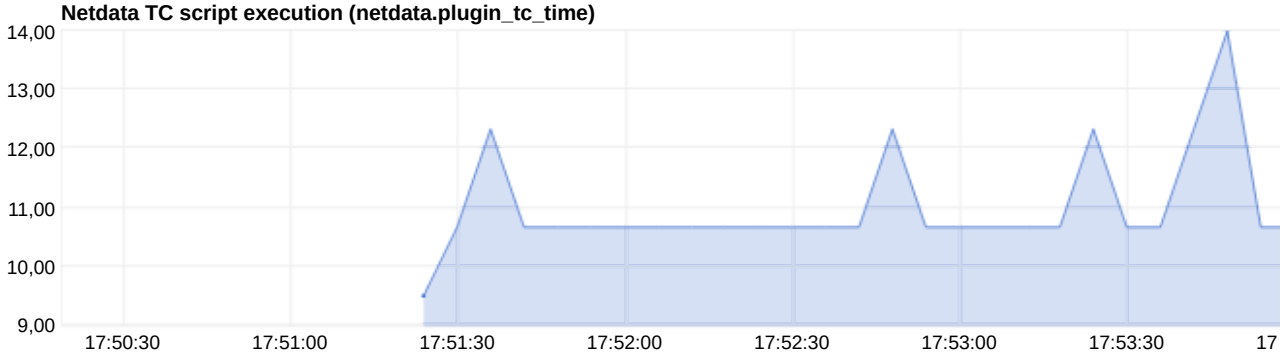
Netdata TC CPU usage (netdata.plugin_tc_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:18

user 0,00 system 0,02

Netdata TC script execution (netdata.plugin_tc_time)

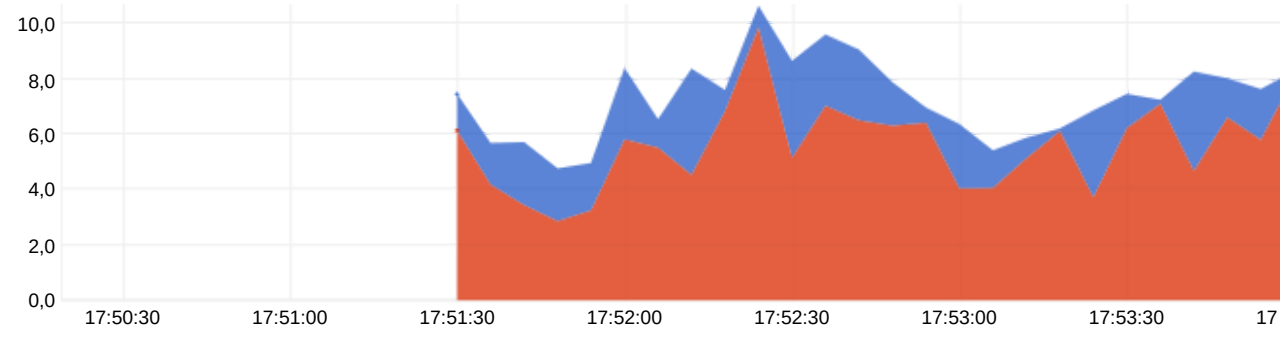


milliseconds/run сб, 24 апр. 2021 г. | 17:55:18

run time 10,67

apps.plugin

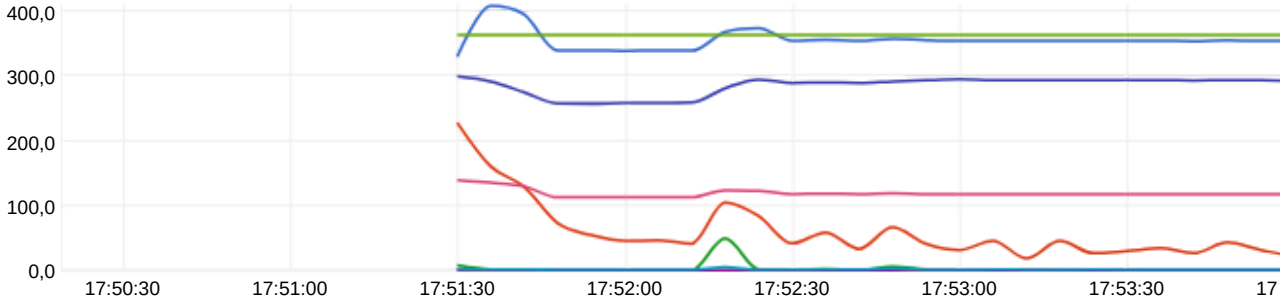
Apps Plugin CPU (netdata.apps_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:18

user 1,2 system 3,1

Apps Plugin Files (netdata.apps_sizes)

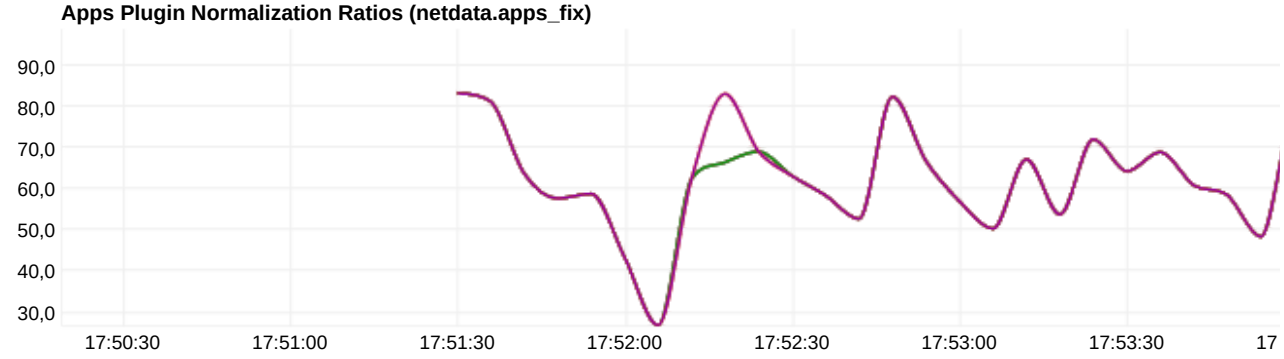


files/s сб, 24 апр. 2021 г. | 17:55:18

calls 354,8 files 29,9 filenames 0,0 inode_changes 0,0 link_changes 0,2

pids 118,0 fds 290,0 targets 364,0 new pids 0,9

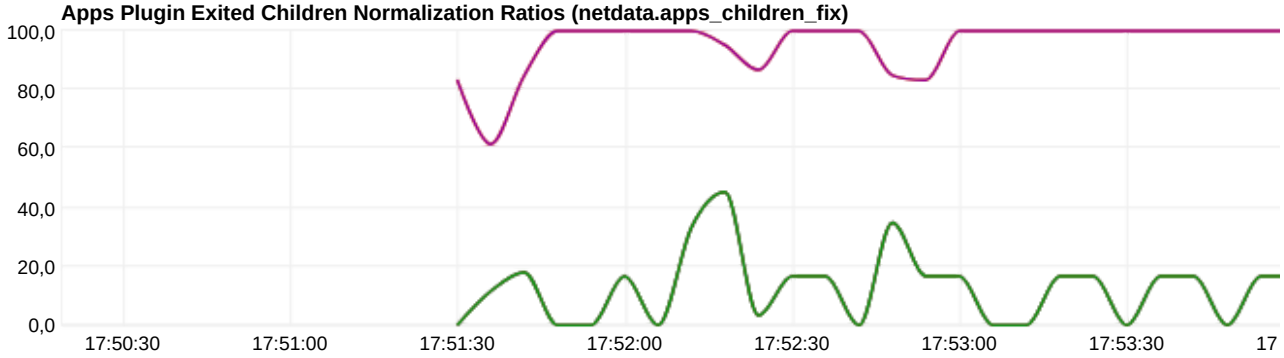
Apps Plugin Normalization Ratios (netdata.apps_fix)



percentage сб, 24 апр. 2021 г. | 17:55:18

utime 77,1 stime 77,1 gtime 77,1 minflt 77,1 majflt 77,1

Apps Plugin Exited Children Normalization Ratios (netdata.apps_children_fix)



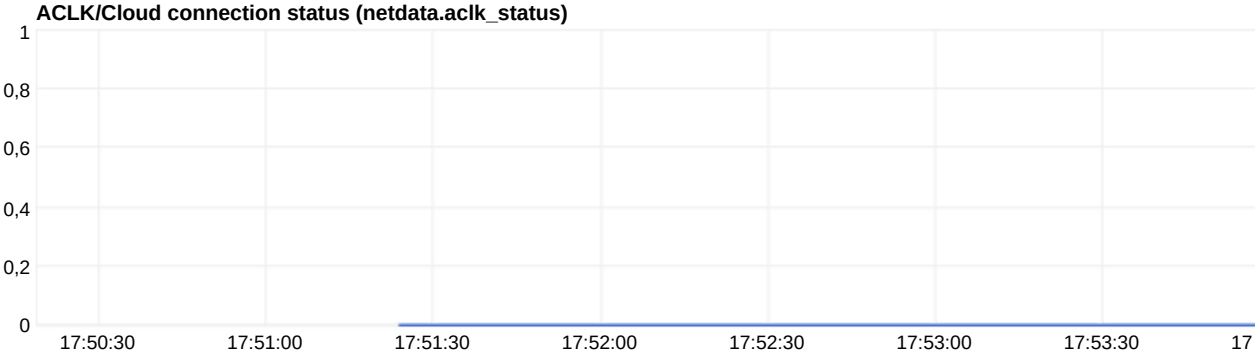
percentage сб, 24 апр. 2021 г. | 17:55:18

cutime 0,0 cstime 0,0 cgtime 0,0 cminflt 100,0 cmajflt 100,0

aclk

This chart shows if ACLK was online during entirety of the sample duration.

ACLK/Cloud connection status (netdata.aclk_status)

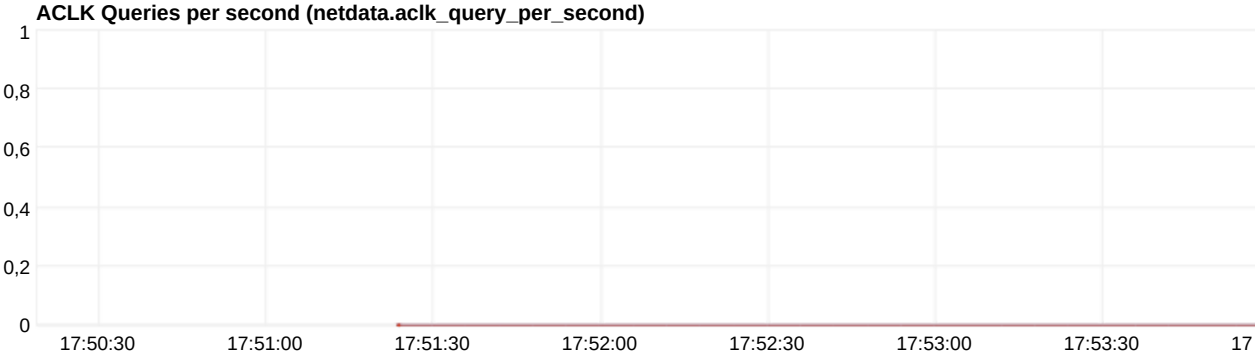


connected сб, 24 апр. 2021 г. | 17:55:18

online 0

This chart shows how many queries were added for ACLK_query thread to process and how many it was actually able to process.

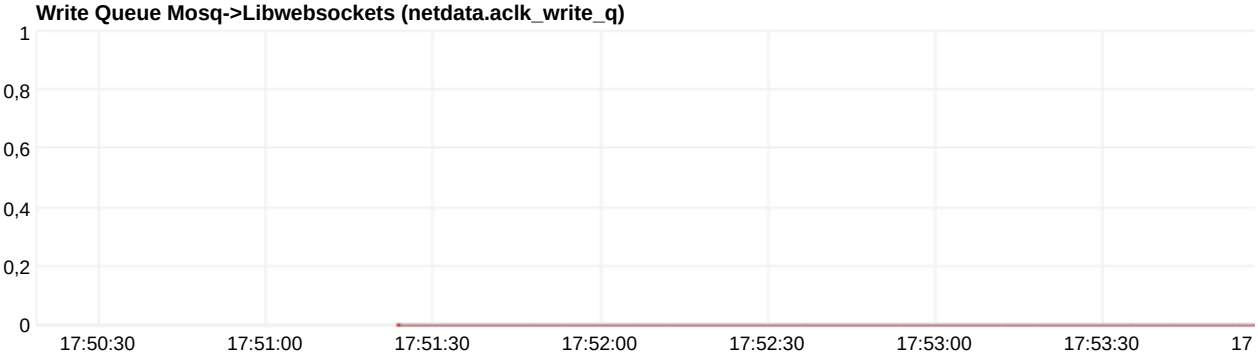
ACLK Queries per second (netdata.aclk_query_per_second)



queries/s сб, 24 апр. 2021 г. | 17:55:18

added 0 dispatched 0

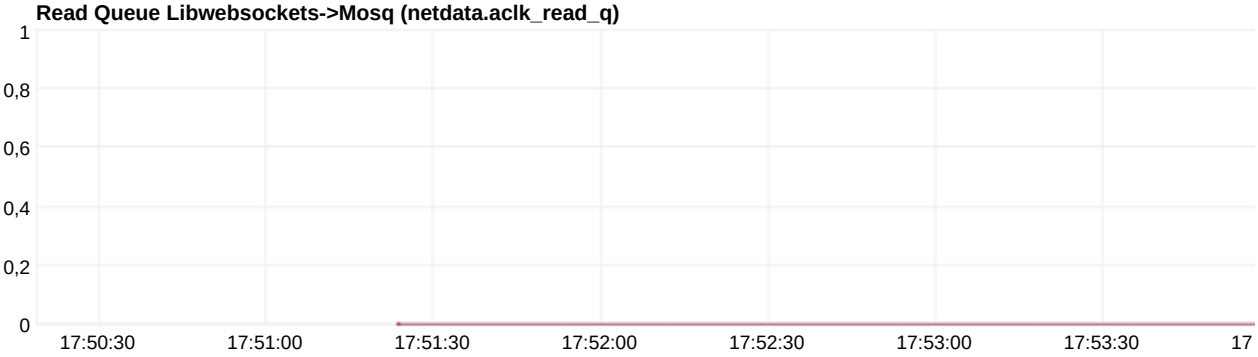
Write Queue Mosq->Libwebsockets (netdata.aclk_write_q)



kB/s сб, 24 апр. 2021 г. | 17:55:18

added 0 consumed 0

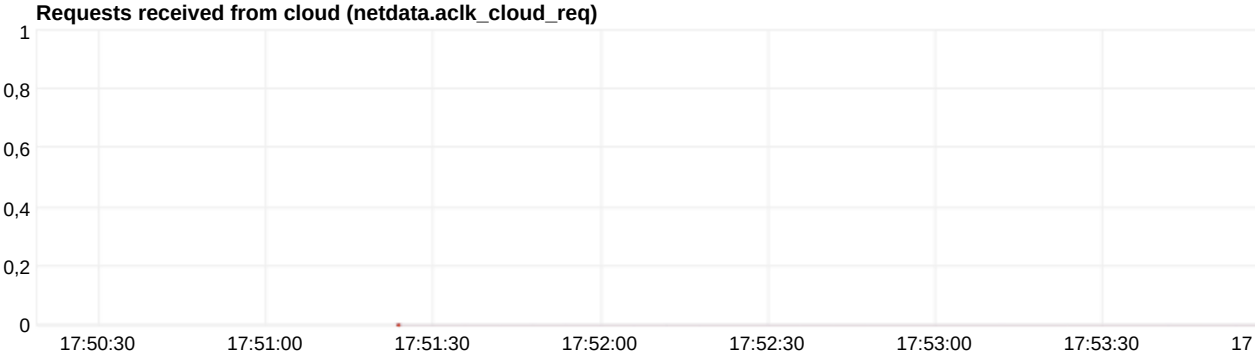
Read Queue Libwebsockets->Mosq (netdata.aclk_read_q)



kB/s сб, 24 апр. 2021 г. | 17:55:18

added 0 consumed 0

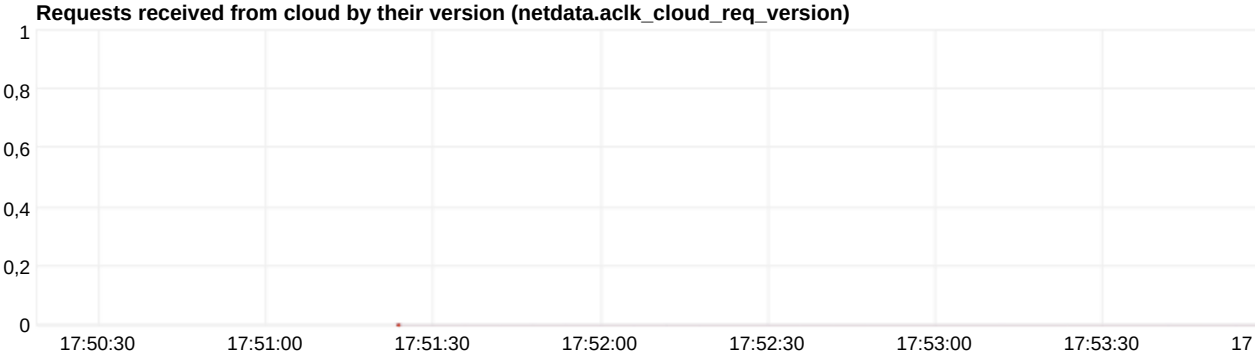
Requests received from cloud (netdata.aclk_cloud_req)



req/s сб, 24 апр. 2021 г. | 17:55:18

accepted 0 rejected 0

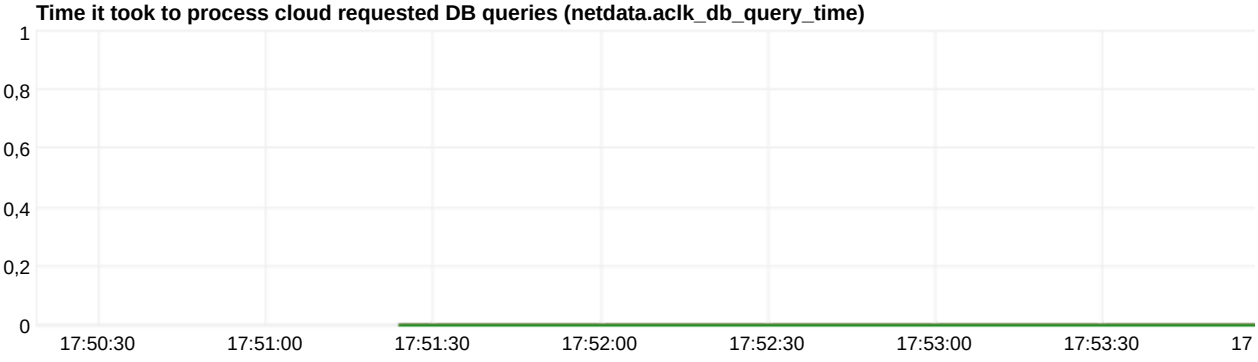
Requests received from cloud by their version (netdata.aclk_cloud_req_version)



req/s сб, 24 апр. 2021 г. | 17:55:18

v1 0 v2+ 0

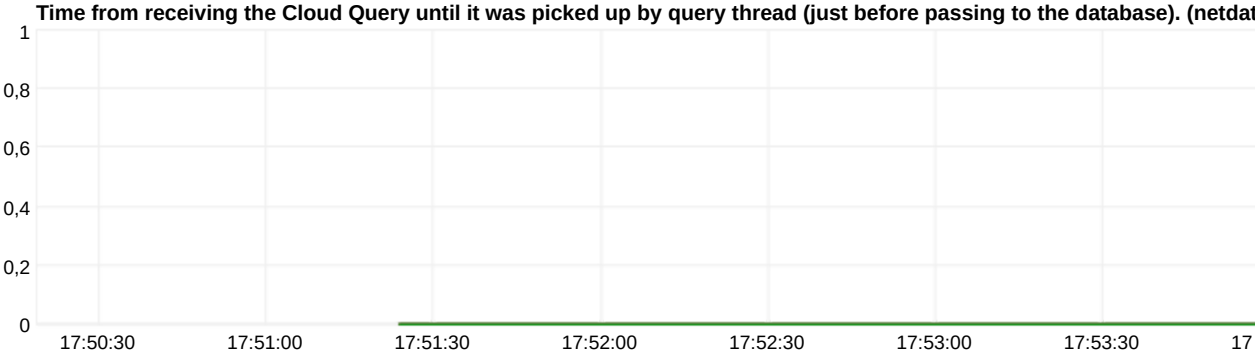
Time it took to process cloud requested DB queries (netdata.aclk_db_query_time)



us сб, 24 апр. 2021 г. | 17:55:18

avg 0 max 0 total 0

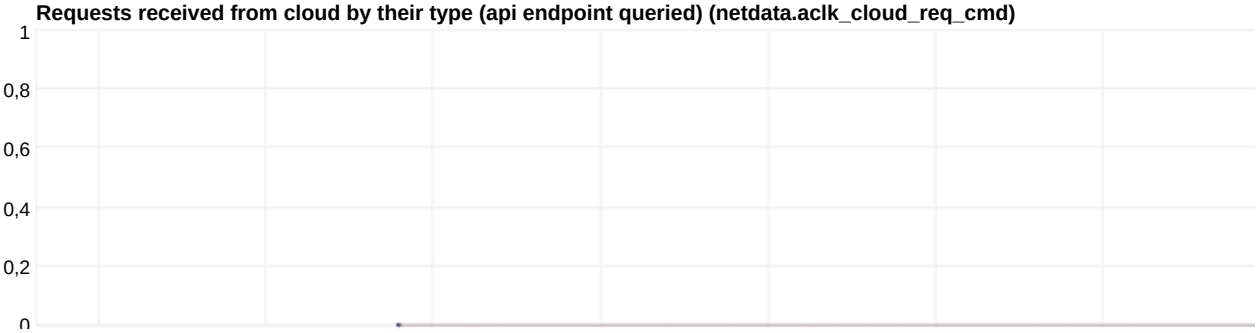
Time from receiving the Cloud Query until it was picked up by query thread (just before passing to the database). (netdat



us сб, 24 апр. 2021 г. | 17:55:18

avg 0 max 0 total 0

Requests received from cloud by their type (api endpoint queried) (netdata.aclk_cloud_req_cmd)

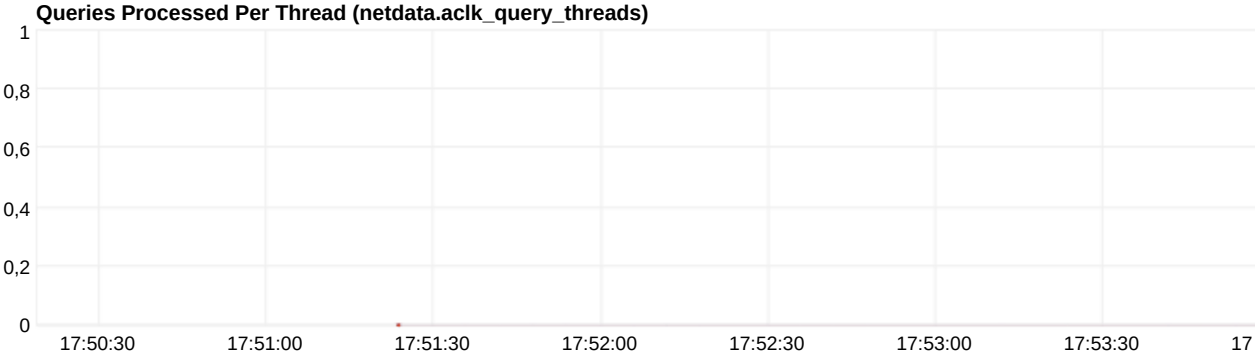


req/s сб, 24 апр. 2021 г. | 17:55:18

other 0 info 0 data 0 alarms 0 alarm_log 0 chart 0

charts 0

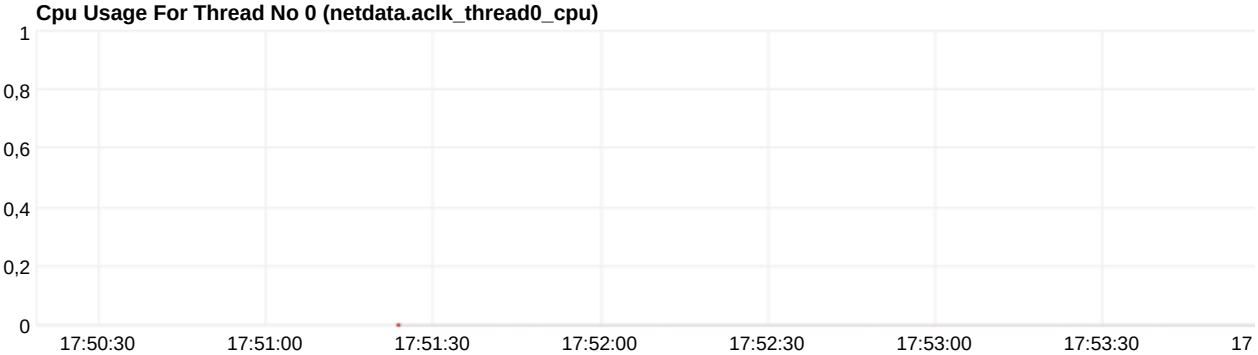
Queries Processed Per Thread (netdata.aclk_query_threads)



req/s сб, 24 апр. 2021 г. | 17:55:18

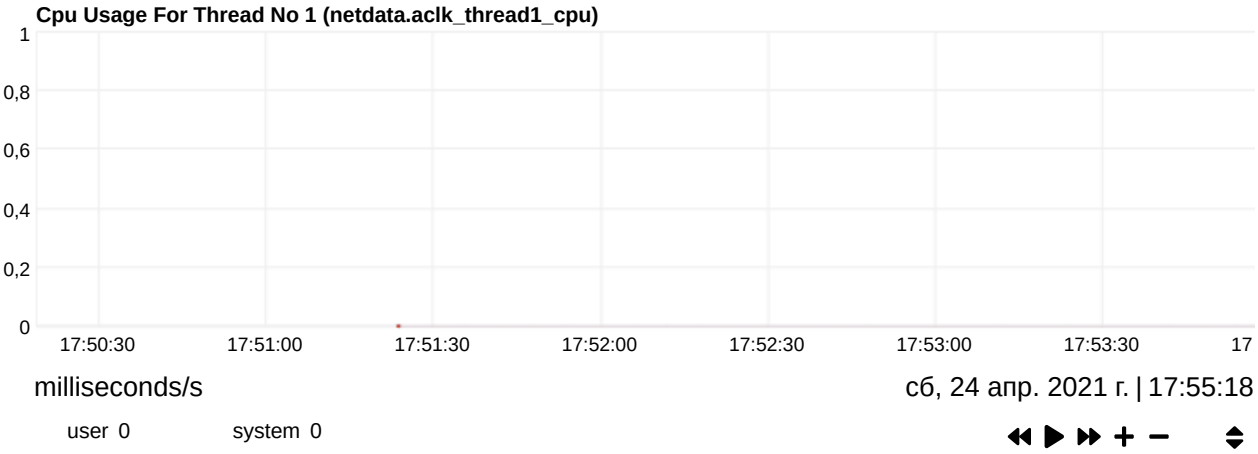
Query 0 0 Query 1 0

Cpu Usage For Thread No 0 (netdata.aclk_thread0_cpu)



milliseconds/s сб, 24 апр. 2021 г. | 17:55:18

user 0 system 0



Netdata (https://github.com/netdata/netdata/wiki)