System Overview

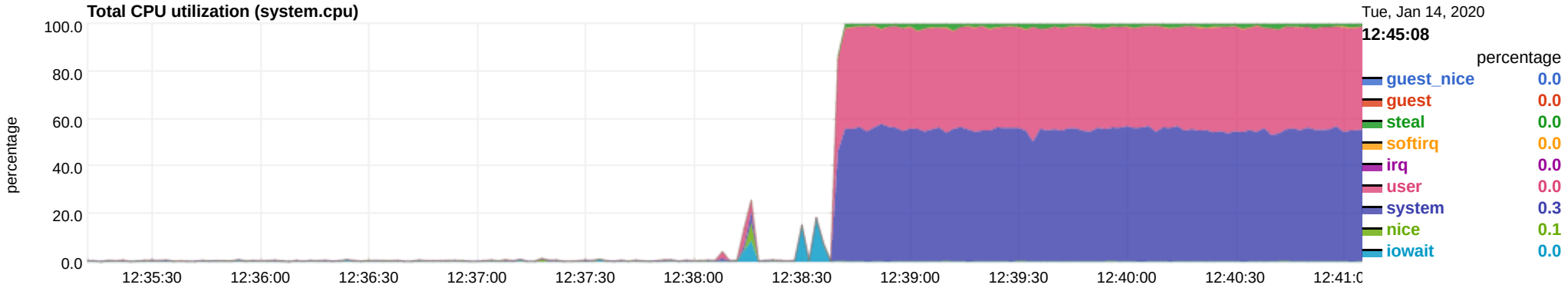
Overview of the key system metrics.

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.

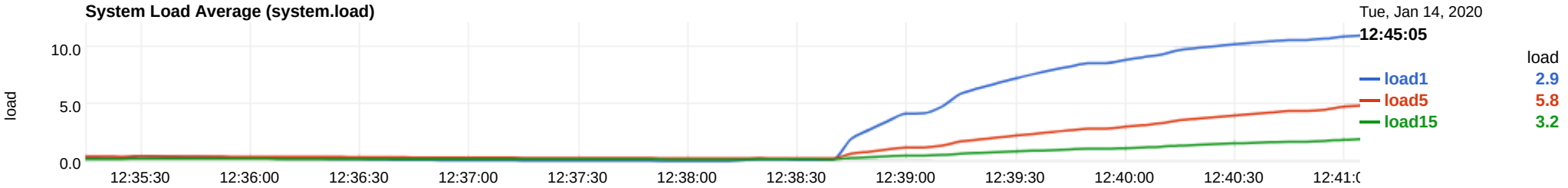
Keep an eye on **iowait** (0.5%). If it is constantly high, your disks are a bottleneck and they slow your system down.

An important metric worth monitoring, is **softirq** (0.00%). A constantly high percentage of softirq may indicate network driver issues.



load

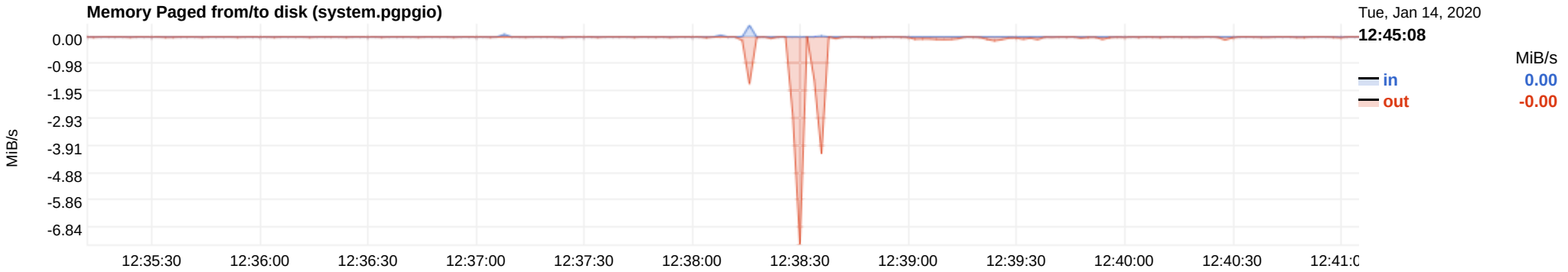
Current system load, i.e. the number of processes using CPU or waiting for system resources (usually CPU and disk). The 3 metrics refer to 1, 5 and 15 minute averages. The system calculates this once every 5 seconds. For more information check this wikipedia article (https://en.wikipedia.org/wiki/Load_(computing))



disk

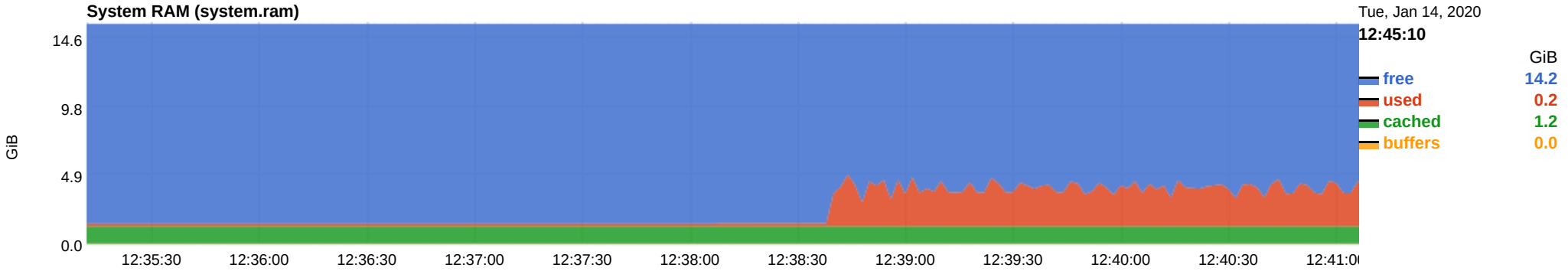
Total Disk I/O, for all physical disks. You can get detailed information about each disk at the Disks section and per application Disk usage at the Applications Monitoring section. Physical are all the disks that are listed in `/sys/block`, but do not exist in `/sys/devices/virtual/block`.

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



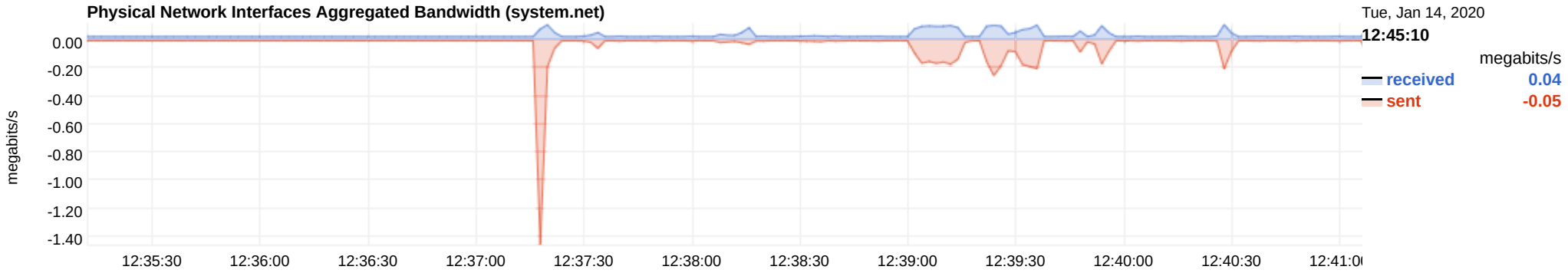
ram

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

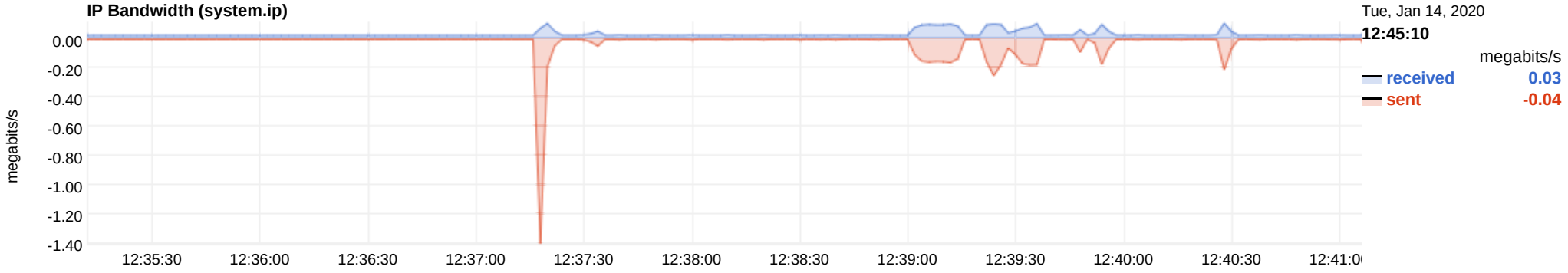


network

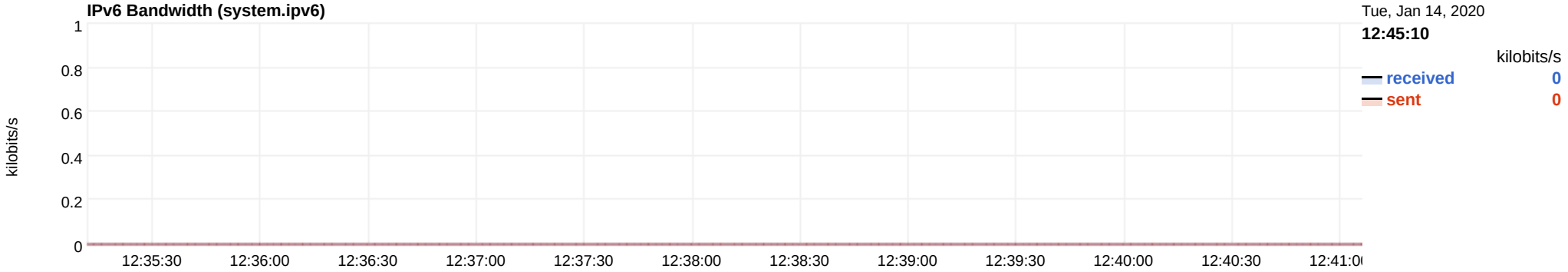
Total bandwidth of all physical network interfaces. This does not include `lo`, VPNs, network bridges, IFB devices, bond interfaces, etc. Only the bandwidth of physical network interfaces is aggregated. Physical are all the network interfaces that are listed in `/proc/net/dev`, but do not exist in `/sys/devices/virtual/net`.



Total IP traffic in the system.

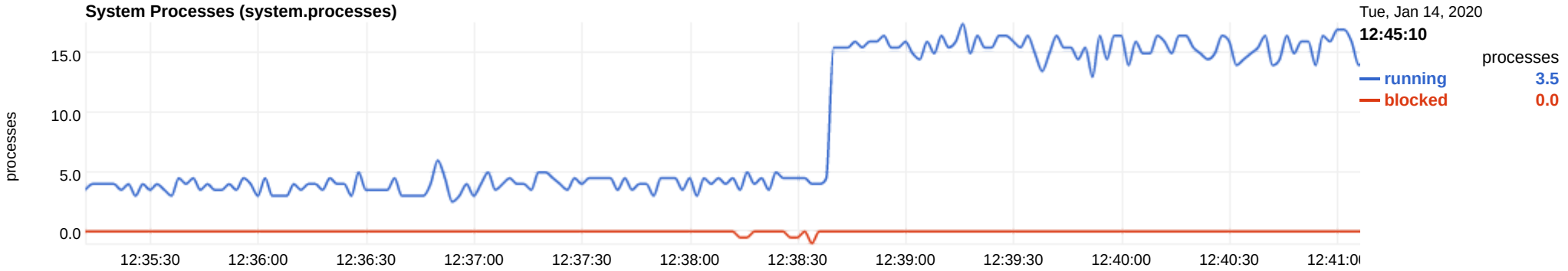


Total IPv6 Traffic.



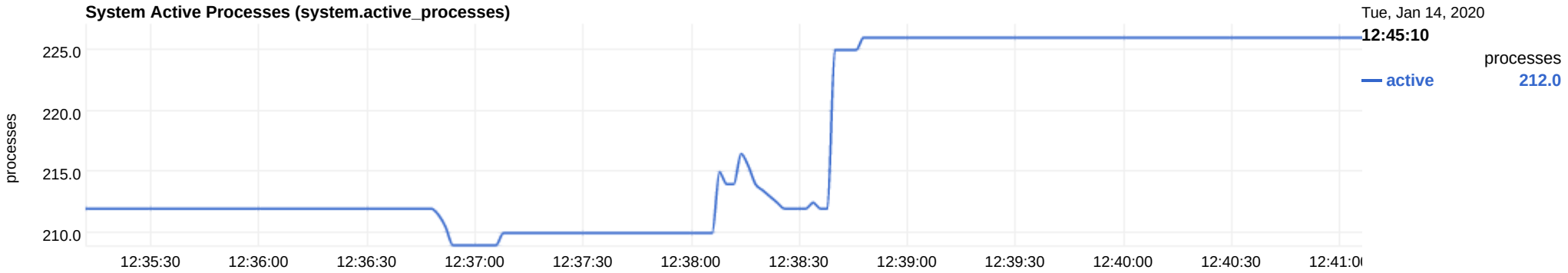
processes

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.



Number of new processes created.

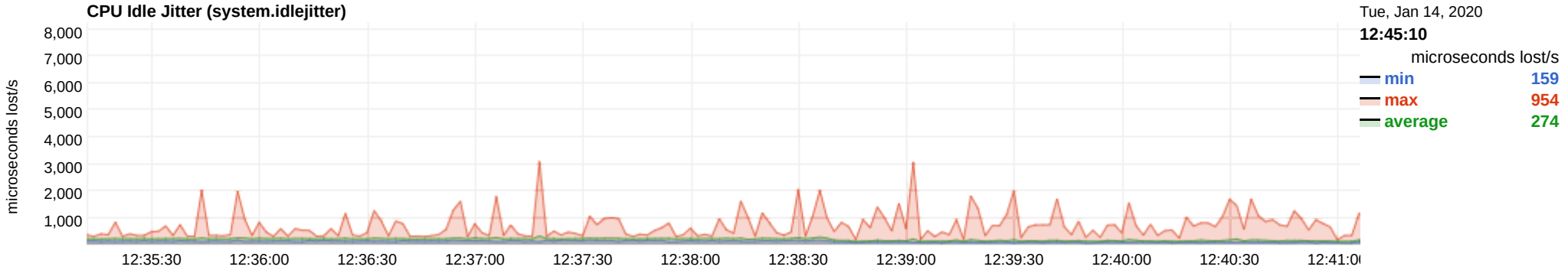
All system processes.



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.

idlejitter

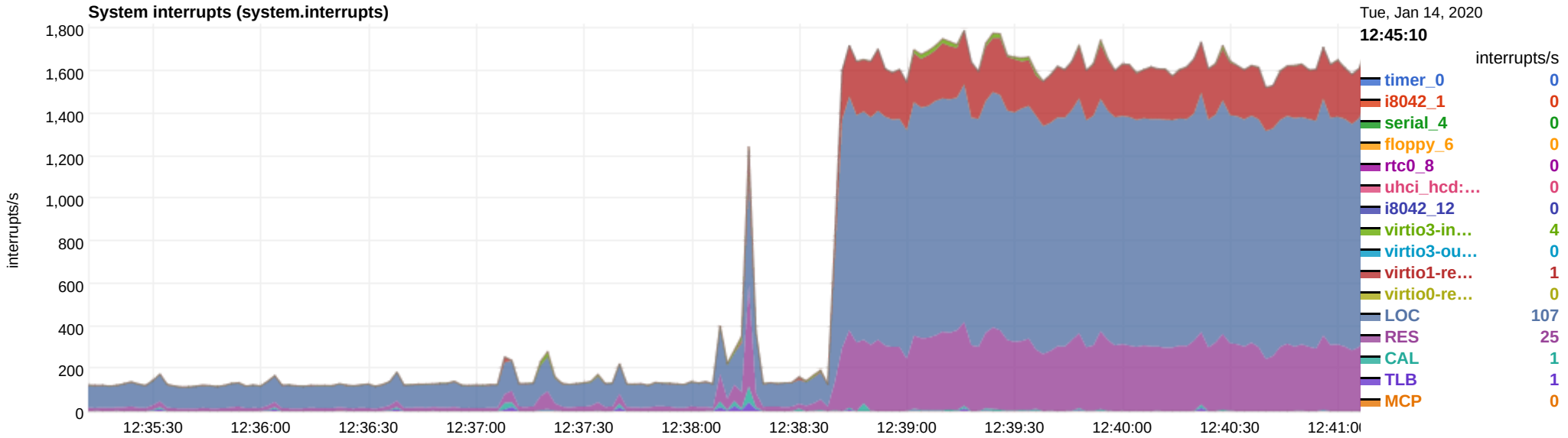
Idle jitter is calculated by netdata. A thread is spawned that requests to sleep for a few microseconds. When the system wakes it up, it measures how many microseconds have passed. The difference between the requested and the actual duration of the sleep, is the **idle jitter**. This number is useful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



interrupts

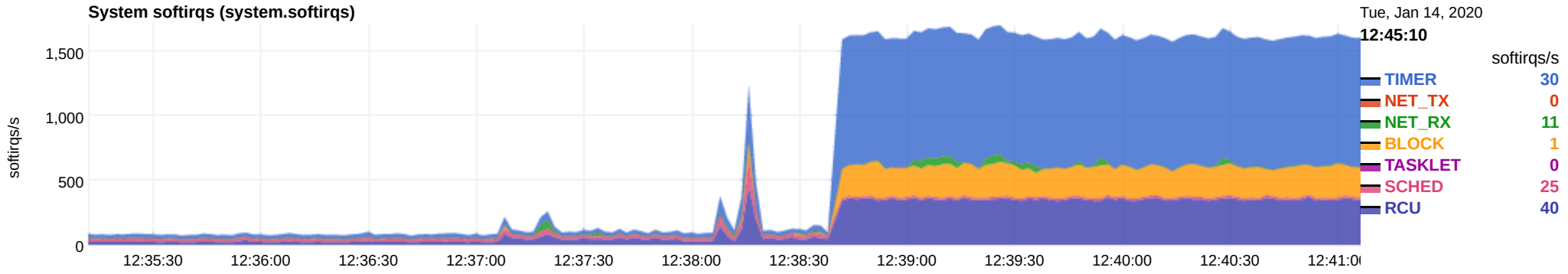
Total number of CPU interrupts. Check `system.interrupts` that gives more detail about each interrupt and also the CPUs section where interrupts are analyzed per CPU core.

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



softirqs

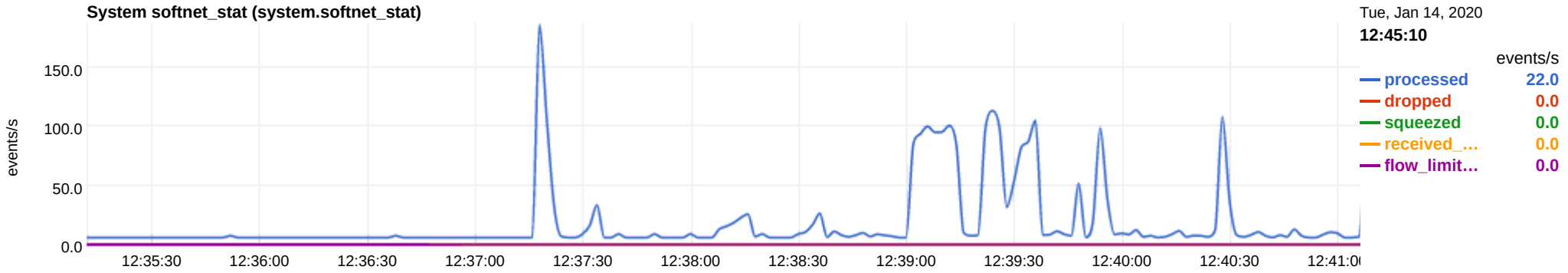
CPU softirqs in detail. At the CPUs section, softirqs are analyzed per CPU core.



softnet

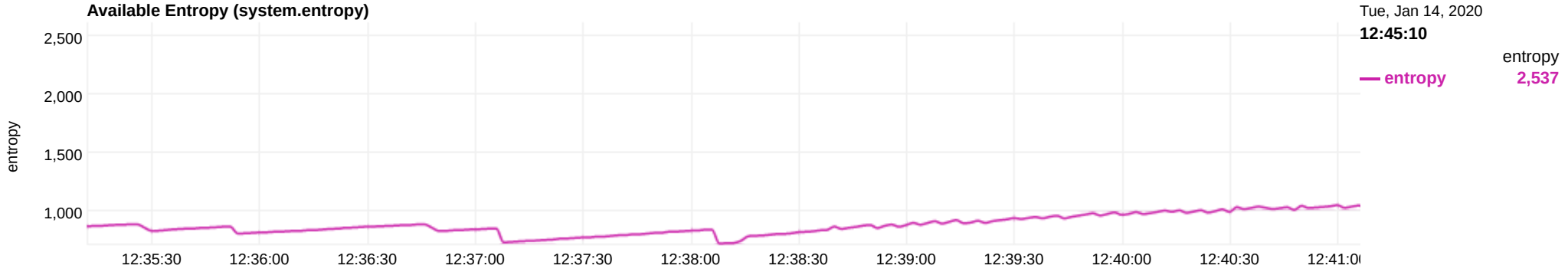
Statistics for CPUs SoftIRQs related to network receive work. Break down per CPU core can be found at CPU / softnet statistics. **processed** states the number of packets processed, **dropped** is the number packets dropped because the network device backlog was full (to fix them on Linux use `sysctl` 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).

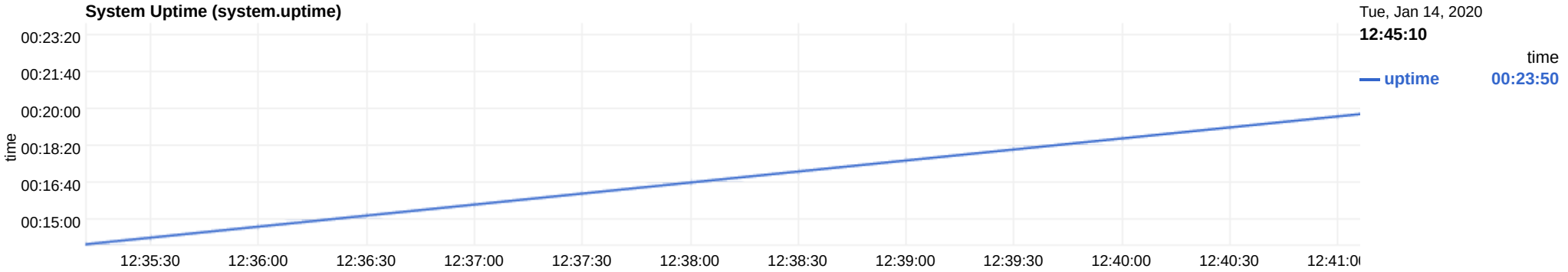


entropy

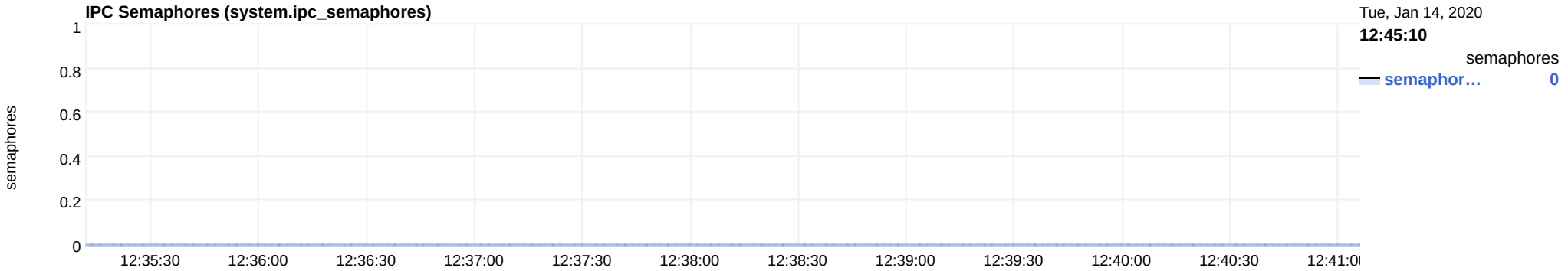
Entropy (https://en.wikipedia.org/wiki/Entropy_(computing)), is a pool of random numbers (/dev/random (https://en.wikipedia.org/wiki//dev/random)) that is mainly used in cryptography. If the pool of entropy gets empty, processes requiring random numbers may run a lot slower (it depends on the interface 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.

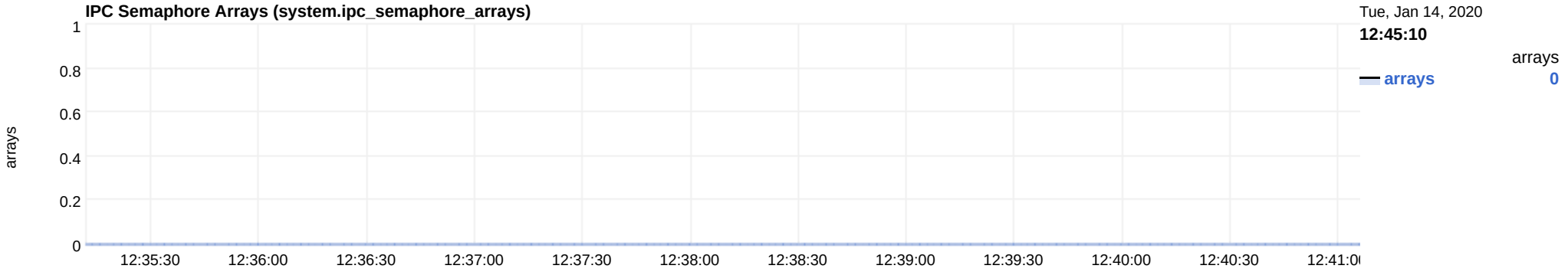


uptime

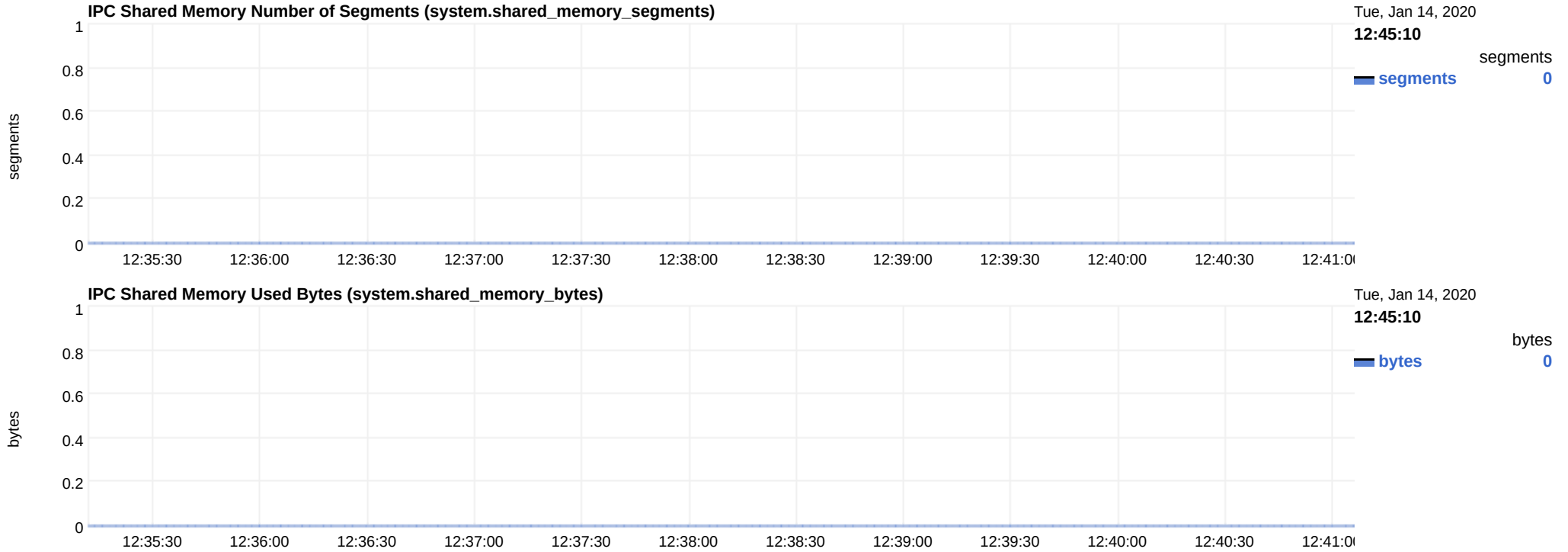


ipc semaphores





ipc shared memory

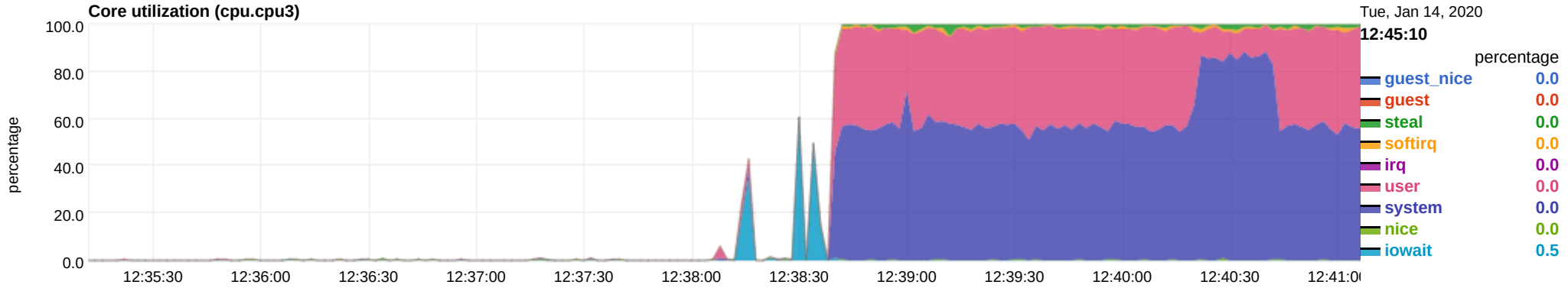


CPUs

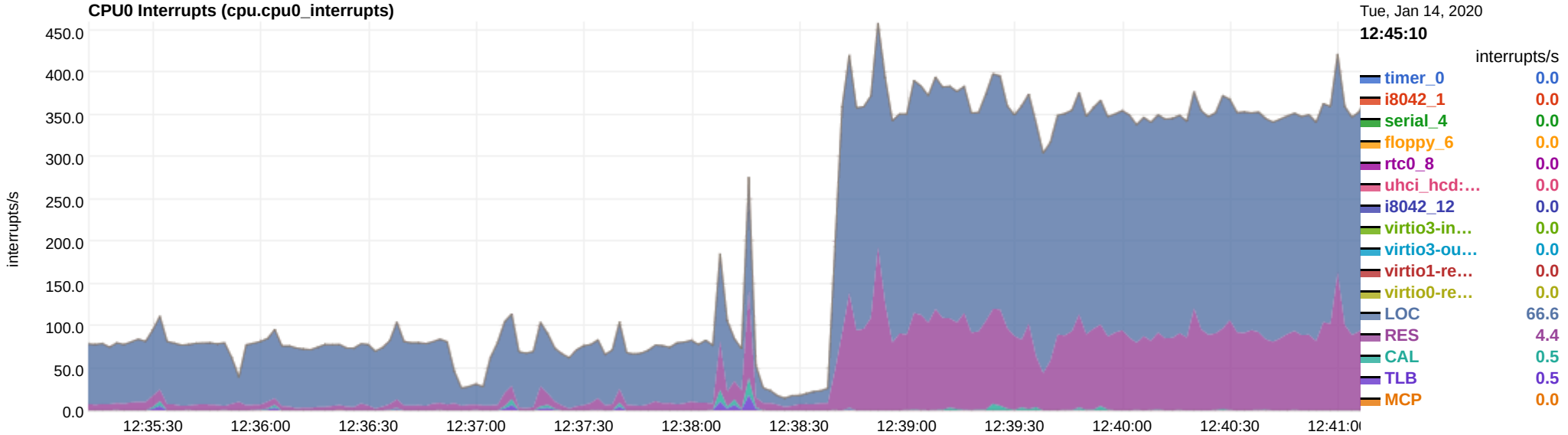
Detailed information for each CPU of the system. A summary of the system for all CPUs can be found at the System Overview section.

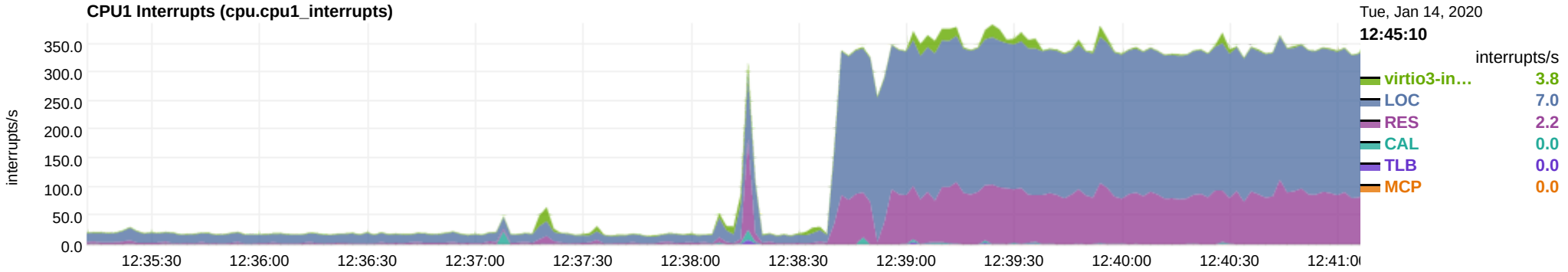
utilization

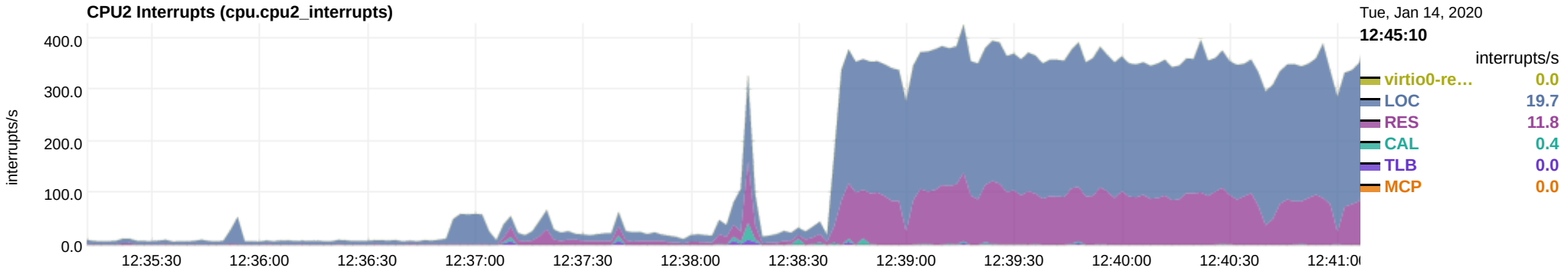

Core utilization (cpu.cpu0)
Tue, Jan 14, 2020
12:45:10
percentage
guest_nice 0.0
guest 0.0
steal 0.0
softirq 0.0
irq 0.0
user 0.0
system 1.0
nice 0.0
iowait 0.0
Core utilization (cpu.cpu1)
Tue, Jan 14, 2020
12:45:10
percentage
guest_nice 0.0
guest 0.0
steal 0.0
softirq 0.0
irq 0.0
user 0.0
system 0.0
nice 0.5
iowait 0.0
Core utilization (cpu.cpu2)
Tue, Jan 14, 2020
12:45:10
percentage
guest_nice 0.0
guest 0.0
steal 0.0
softirq 0.0
irq 0.0
user 0.0
system 0.0
nice 0.5
iowait 0.0
percentage
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30

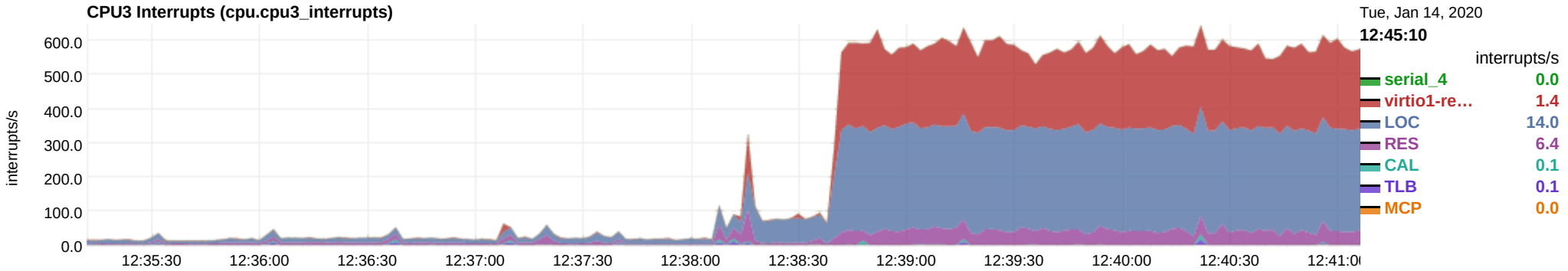

Core utilization (cpu.cpu3)
Tue, Jan 14, 2020
12:45:10
percentage
guest_nice 0.0
guest 0.0
steal 0.0
softirq 0.0
irq 0.0
user 0.0
system 0.0
nice 0.0
iowait 0.5
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30

interrupts

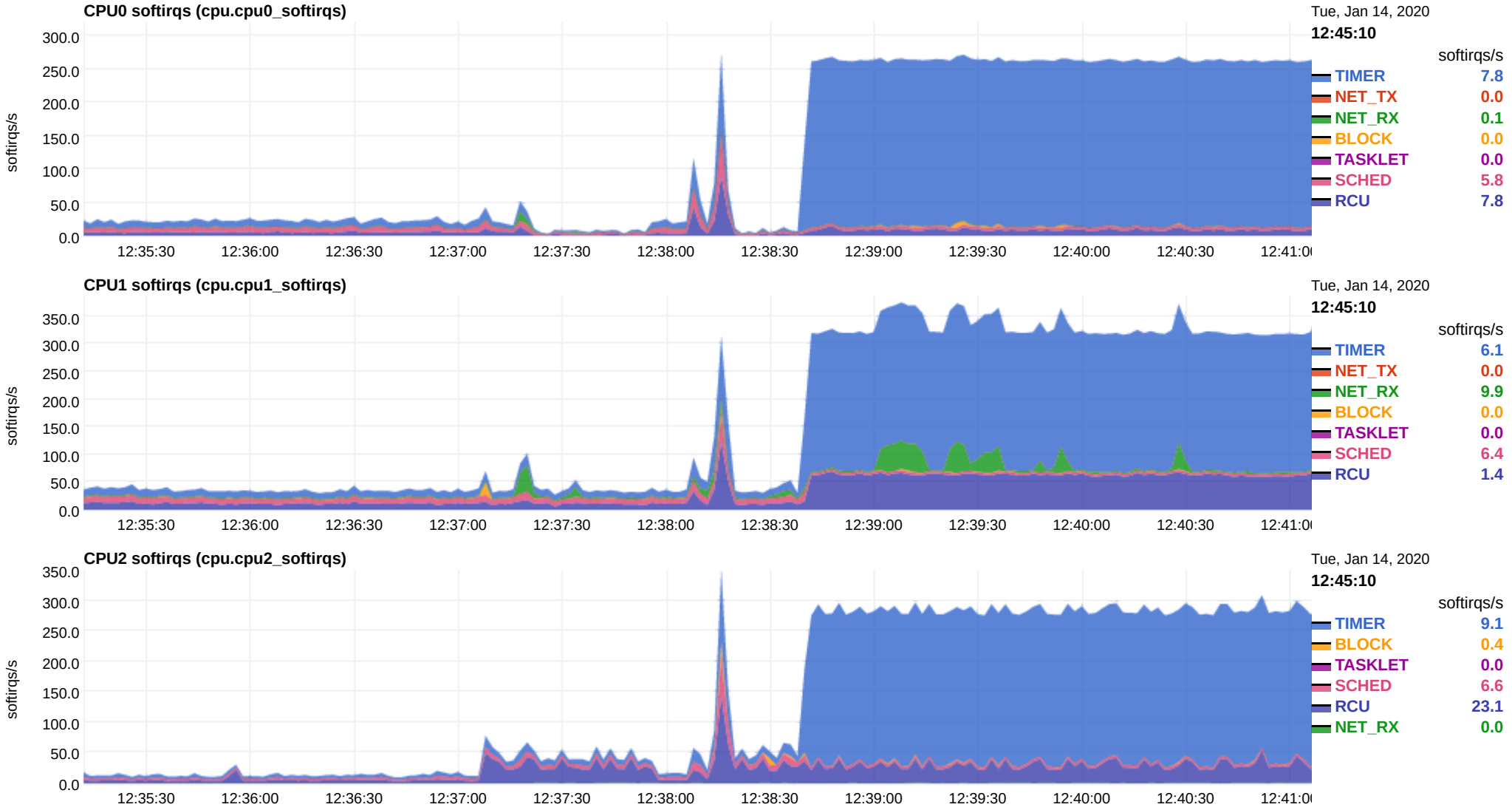

CPU0 Interrupts (cpu.cpu0_interrupts)
Tue, Jan 14, 2020
12:45:10
interrupts/s
timer_0 0.0
i8042_1 0.0
serial_4 0.0
floppy_6 0.0
rtc0_8 0.0
uhci_hcd:... 0.0
i8042_12 0.0
virtio3-in... 0.0
virtio3-ou... 0.0
virtio1-re... 0.0
virtio0-re... 0.0
LOC 66.6
RES 4.4
CAL 0.5
TLB 0.5
MCP 0.0
450.0
400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30

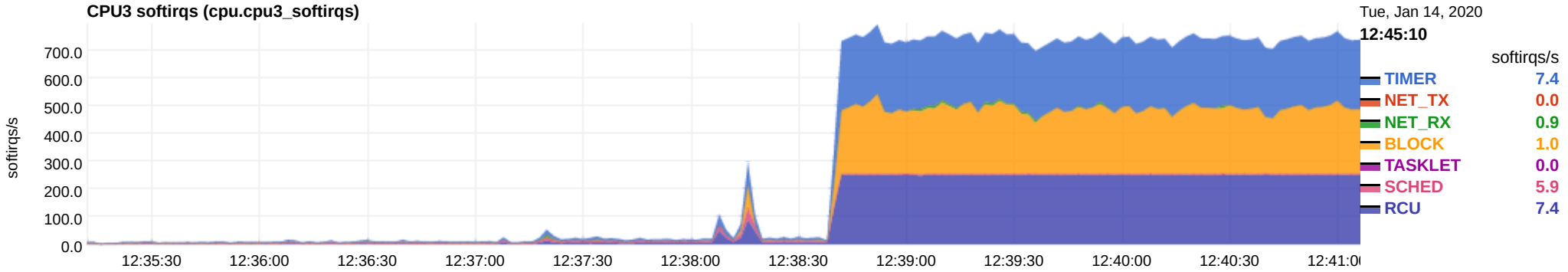

CPU1 Interrupts (cpu.cpu1_interrupts)
interrupts/s
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
Tue, Jan 14, 2020
12:45:10
interrupts/s
virtio3-in… 3.8
LOC 7.0
RES 2.2
CAL 0.0
TLB 0.0
MCP 0.0


CPU2 Interrupts (cpu.cpu2_interrupts)
interrupts/s
400.0
300.0
200.0
100.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
Tue, Jan 14, 2020
12:45:10
interrupts/s
virtio0-re… 0.0
LOC 19.7
RES 11.8
CAL 0.4
TLB 0.0
MCP 0.0


CPU3 Interrupts (cpu.cpu3_interrupts)
interrupts/s
600.0
500.0
400.0
300.0
200.0
100.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
Tue, Jan 14, 2020
12:45:10
interrupts/s
serial_4 0.0
virtio1-re… 1.4
LOC 14.0
RES 6.4
CAL 0.1
TLB 0.1
MCP 0.0

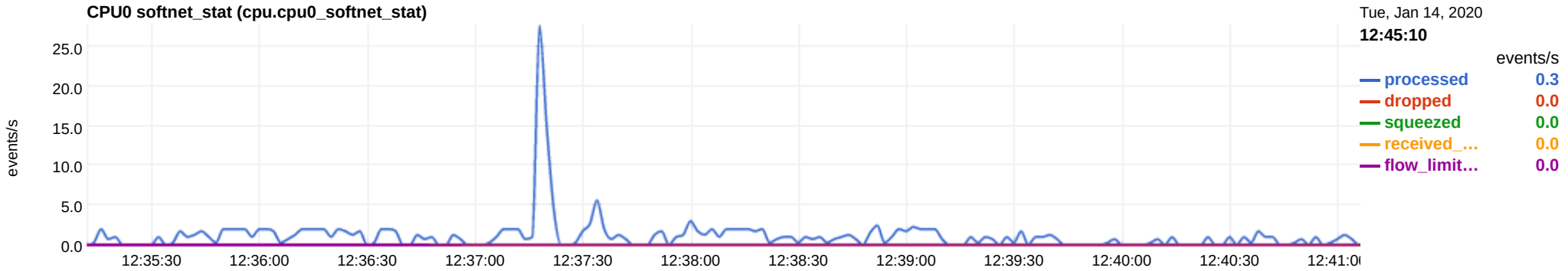
softirqs


CPU0 softirqs (cpu.cpu0_softirqs)
Tue, Jan 14, 2020
12:45:10
softirqs/s
TIMER 7.8
NET_TX 0.0
NET_RX 0.1
BLOCK 0.0
TASKLET 0.0
SCHED 5.8
RCU 7.8
softirqs/s
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
CPU1 softirqs (cpu.cpu1_softirqs)
Tue, Jan 14, 2020
12:45:10
softirqs/s
TIMER 6.1
NET_TX 0.0
NET_RX 9.9
BLOCK 0.0
TASKLET 0.0
SCHED 6.4
RCU 1.4
softirqs/s
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
CPU2 softirqs (cpu.cpu2_softirqs)
Tue, Jan 14, 2020
12:45:10
softirqs/s
TIMER 9.1
BLOCK 0.4
TASKLET 0.0
SCHED 6.6
RCU 23.1
NET_RX 0.0
softirqs/s
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30



softnet

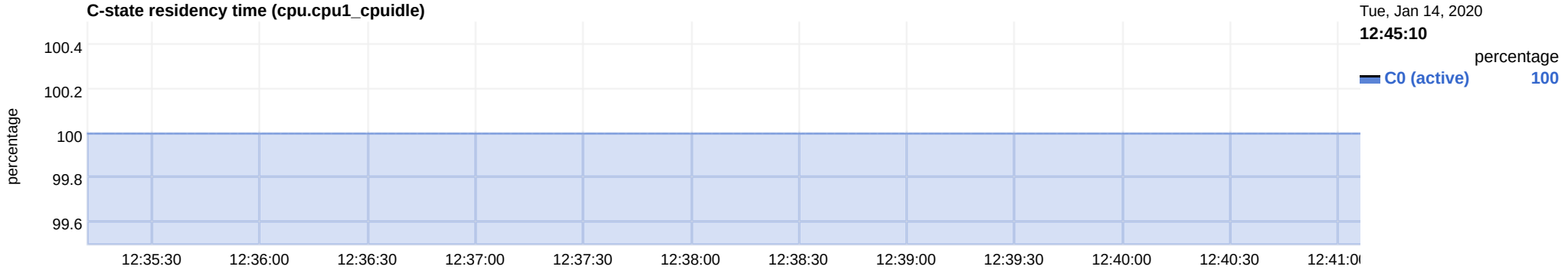
Statistics for per CPUs core SoftIRQs related to network receive work. Total for all CPU cores can be found at System / softnet statistics. **processed** states the number of packets processed, **dropped** is the number packets dropped because the network device backlog was full (to fix them on Linux 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).

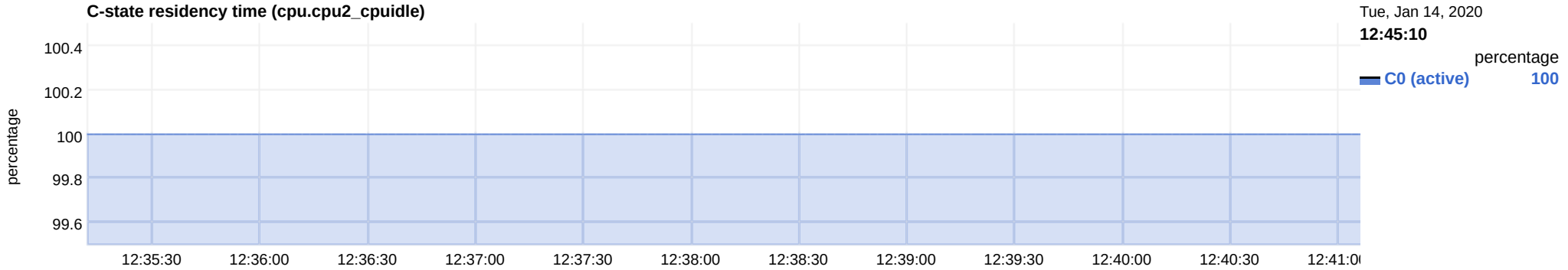


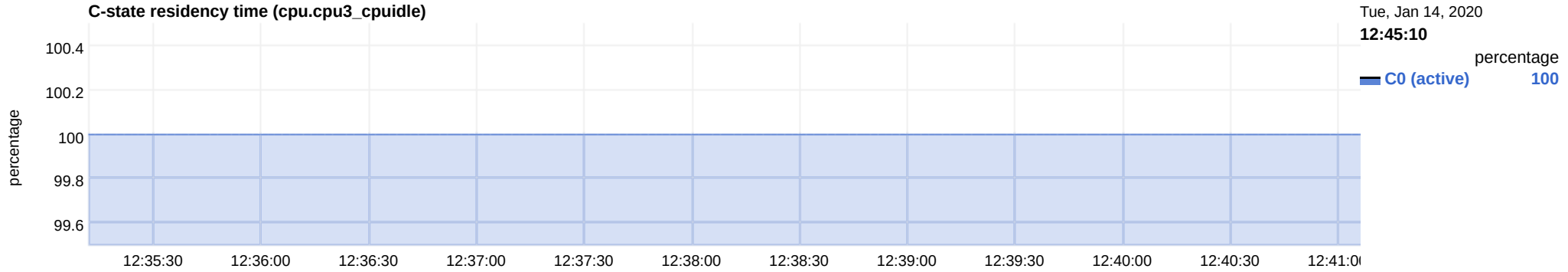

CPU1 softnet_stat (cpu.cpu1_softnet_stat)
Tue, Jan 14, 2020
12:45:10
events/s
processed 20.0
dropped 0.0
squeezed 0.0
received_... 0.0
flow_limit... 0.0
events/s
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30 12:36:00 12:36:30 12:37:00 12:37:30 12:38:00 12:38:30 12:39:00 12:39:30 12:40:00 12:40:30 12:41:0
CPU2 softnet_stat (cpu.cpu2_softnet_stat)
Tue, Jan 14, 2020
12:45:10
events/s
processed 0.00
dropped 0.00
squeezed 0.00
received_... 0.00
flow_limit... 0.00
events/s
2.00
1.50
1.00
0.50
0.00
12:35:30 12:36:00 12:36:30 12:37:00 12:37:30 12:38:00 12:38:30 12:39:00 12:39:30 12:40:00 12:40:30 12:41:0
CPU3 softnet_stat (cpu.cpu3_softnet_stat)
Tue, Jan 14, 2020
12:45:10
events/s
processed 1.7
dropped 0.0
squeezed 0.0
received_... 0.0
flow_limit... 0.0
events/s
15.0
10.0
5.0
0.0
12:35:30 12:36:00 12:36:30 12:37:00 12:37:30 12:38:00 12:38:30 12:39:00 12:39:30 12:40:00 12:40:30 12:41:0

cpuidle


C-state residency time (cpu.cpu0_cpuidle)
Tue, Jan 14, 2020
12:45:10
percentage
C0 (active) 100
percentage
100.4
100.2
100
99.8
99.6
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30


C-state residency time (cpu.cpu1_cpuidle)
Tue, Jan 14, 2020
12:45:10
percentage
C0 (active) 100
percentage
100.4
100.2
100
99.8
99.6
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30


C-state residency time (cpu.cpu2_cpuidle)
Tue, Jan 14, 2020
12:45:10
percentage
C0 (active) 100
percentage
100.4
100.2
100
99.8
99.6
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30

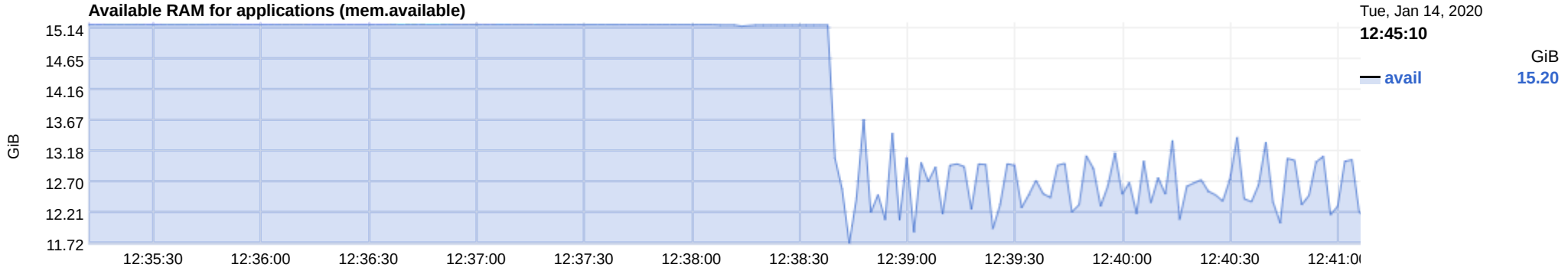


Memory

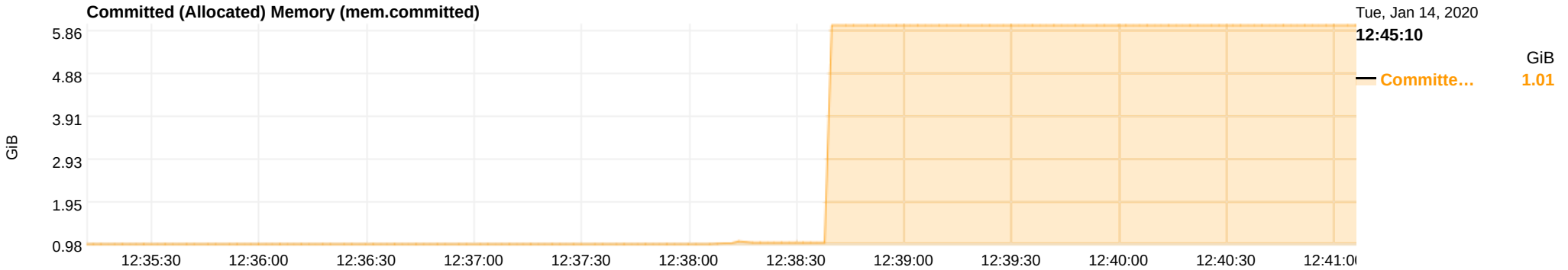
Detailed information about the memory management of the system.

system

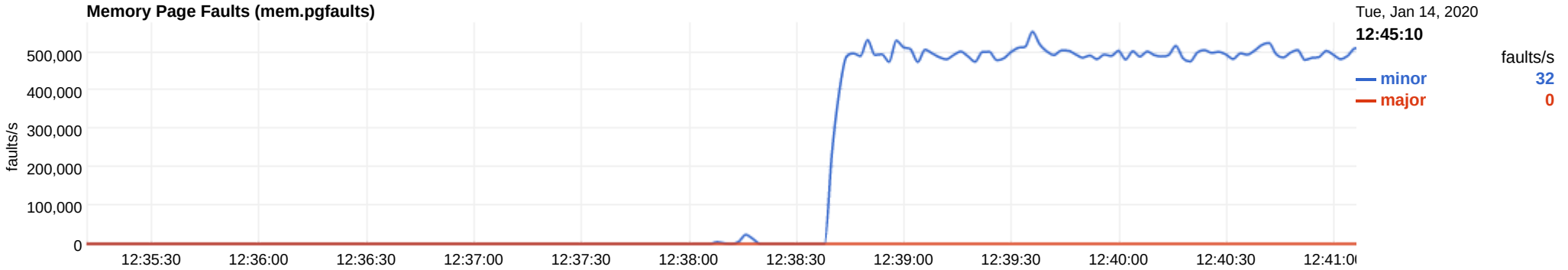
Available Memory is estimated by the kernel, as the amount of RAM that can be used by userspace processes, without causing swapping.



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

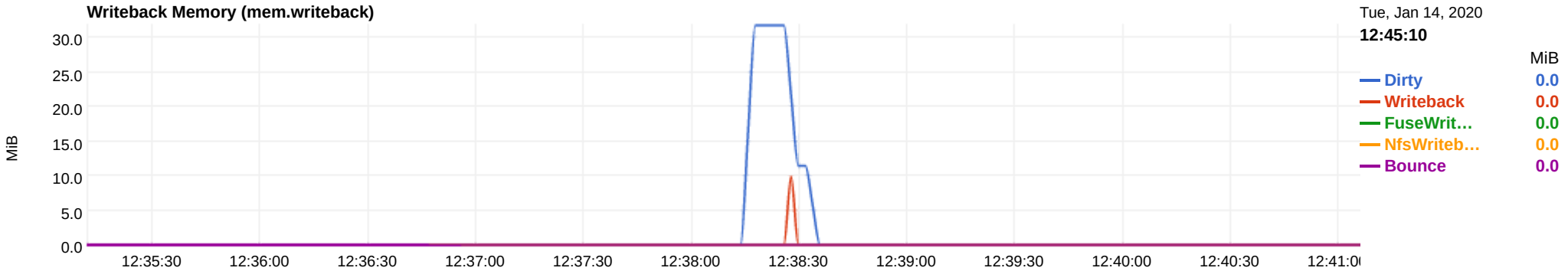


A page fault (https://en.wikipedia.org/wiki/Page_fault) is a type of interrupt, called trap, raised by computer hardware when a running program accesses a memory page that is mapped into the virtual address space, but not actually loaded into main memory. If the page is loaded in memory at the time the 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.

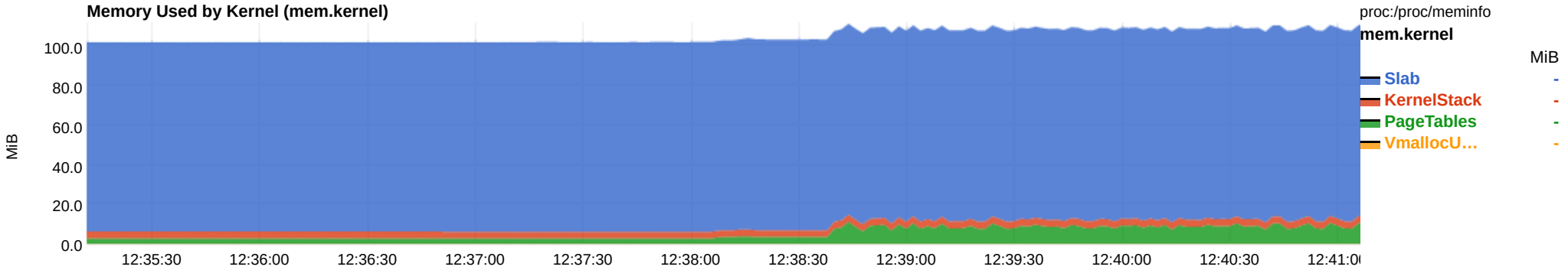


kernel

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

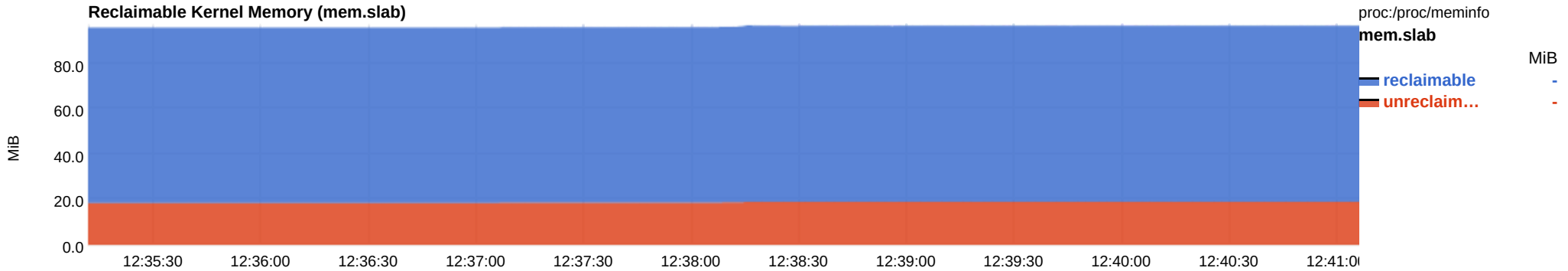


The total amount of memory being used by the kernel. **Slab** is the amount of memory used by the kernel to cache data structures for its own use. **KernelStack** is the amount of memory allocated for each task done by the kernel. **PageTables** is the amount of memory decicated to the lowest level of 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.



slab

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



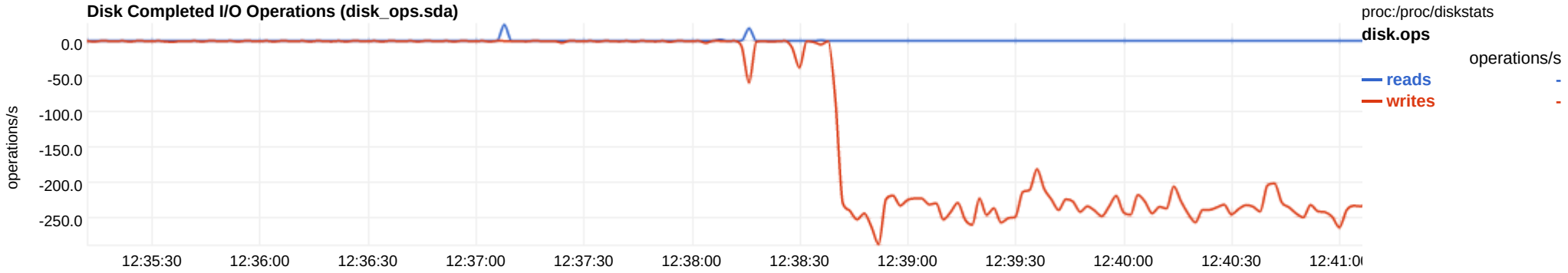
Disks

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

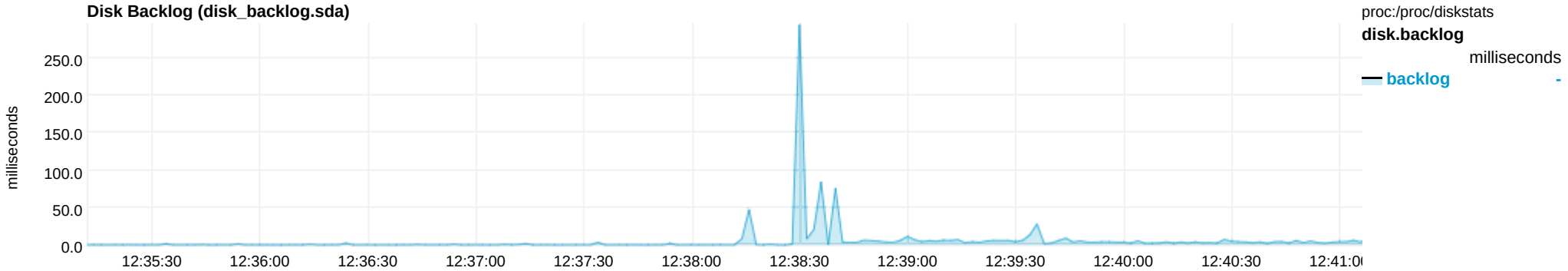
Amount of data transferred to and from disk.

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).

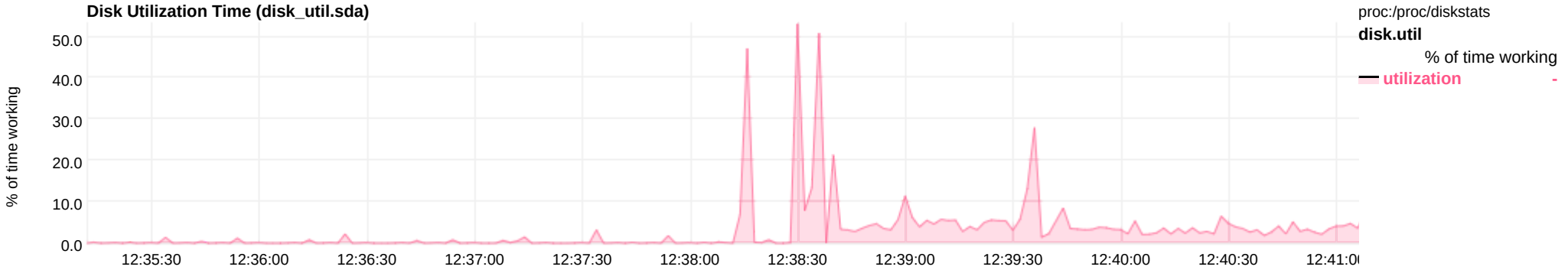


I/O operations currently in progress. This metric is a snapshot - it is not an average over the last interval.

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.

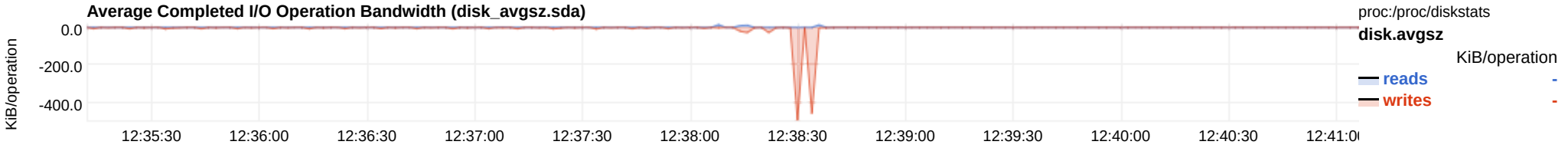


Disk Utilization measures the amount of time the disk was busy with something. This is not related to its performance. 100% means that the system always had an outstanding operation on the disk. Keep in mind that depending on the underlying technology of the disk, 100% here may or may not be an indication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



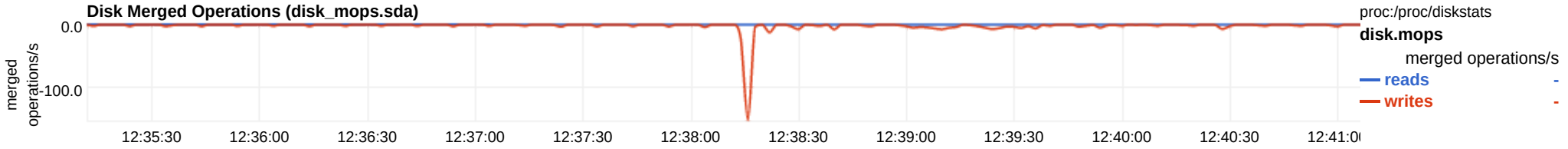
The average time for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

The average I/O operation size.

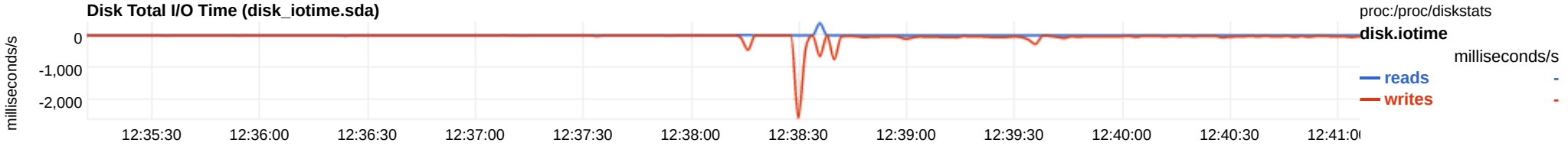


The average service time for completed I/O operations. This metric is calculated using the total busy time of the disk and the number of completed operations. If the disk is able to execute multiple parallel operations the reporting average service time will be misleading.

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.

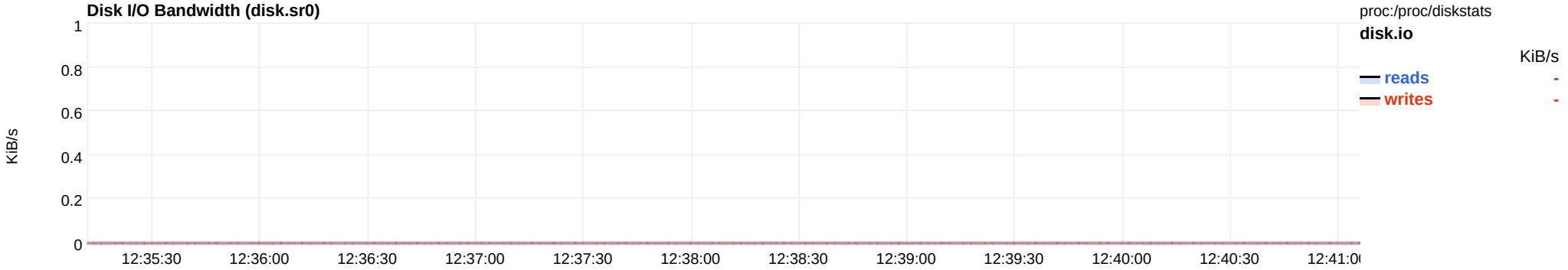


The sum of the duration of all completed I/O operations. This number can exceed the interval if the disk is able to execute I/O operations in parallel.

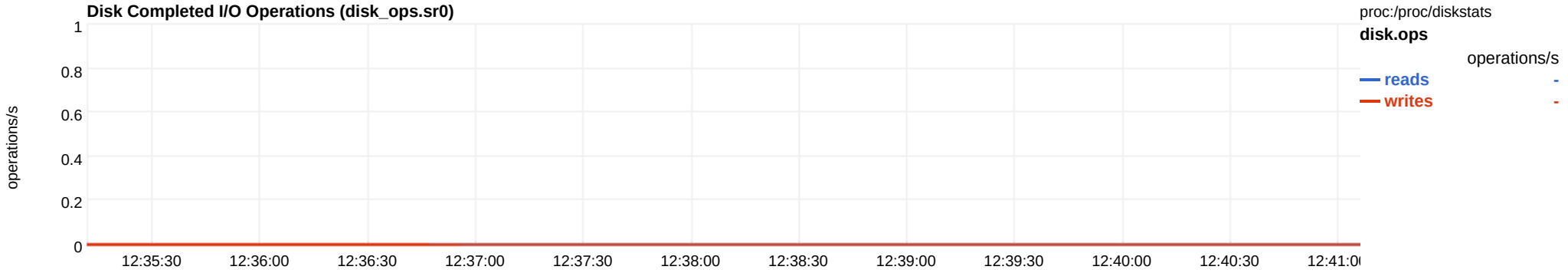


sr0

Amount of data transferred to and from disk.



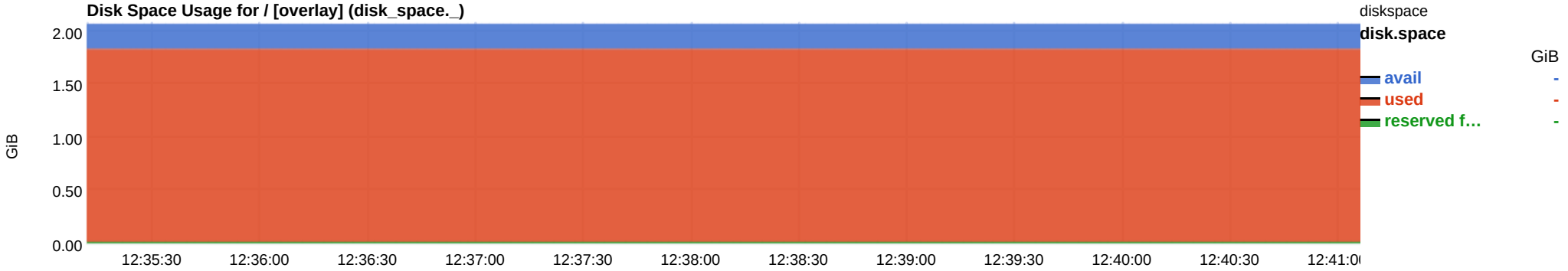
Completed disk I/O operations. Keep in mind the number of operations requested might be higher, since the system is able to merge adjacent to each other (see merged operations chart).



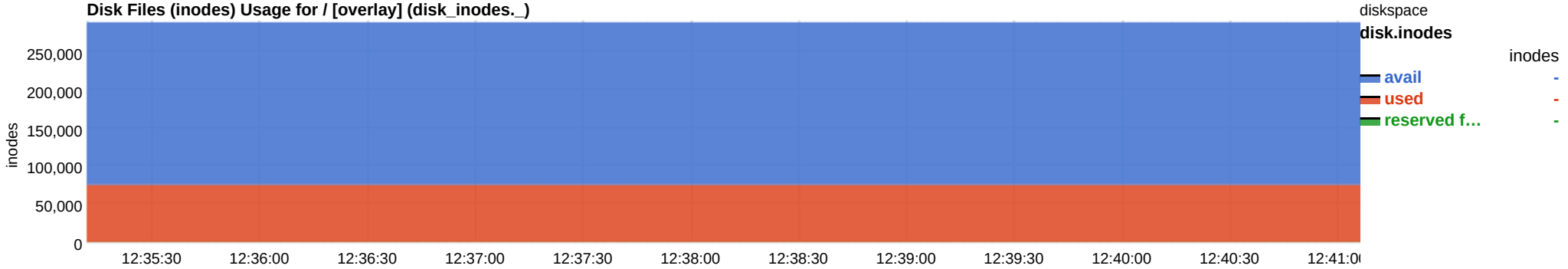
The average I/O operation size.

/

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

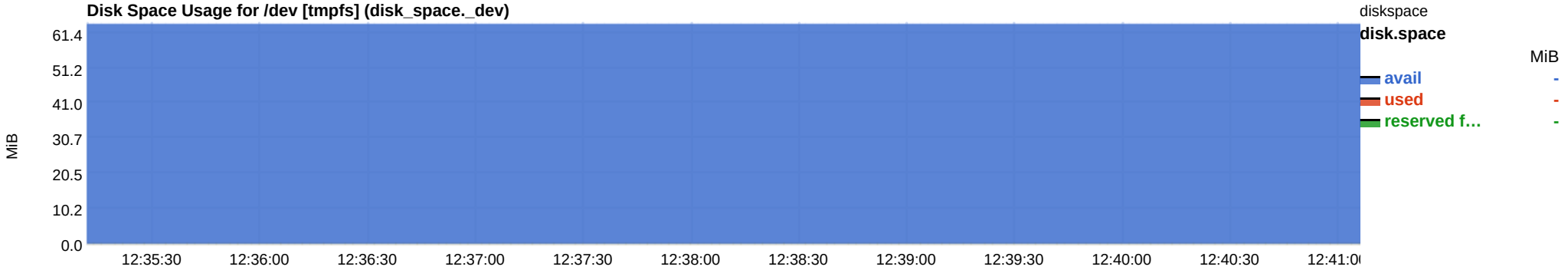


inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

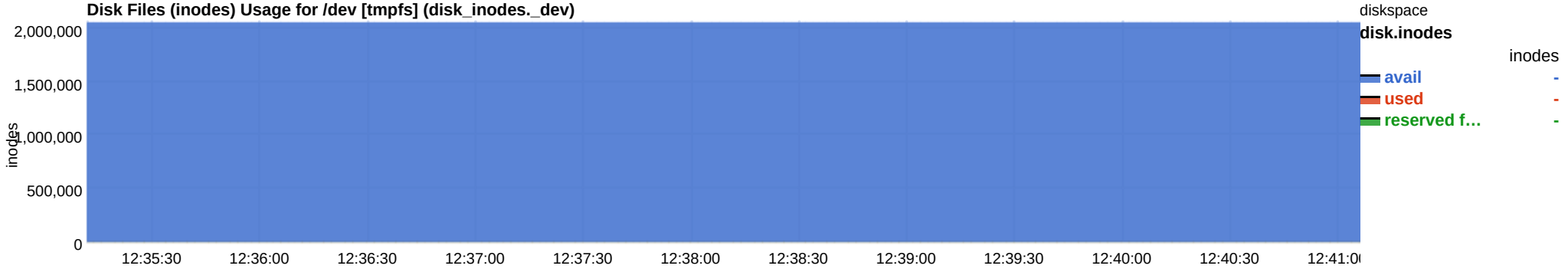


/dev

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

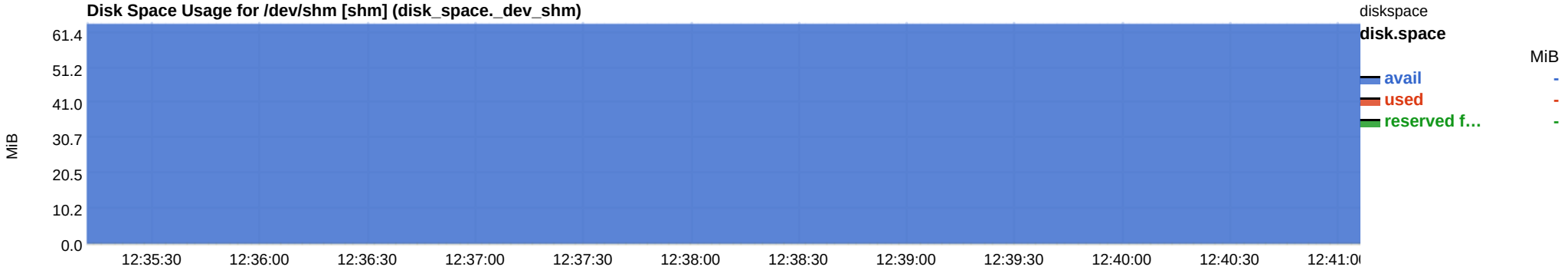


inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

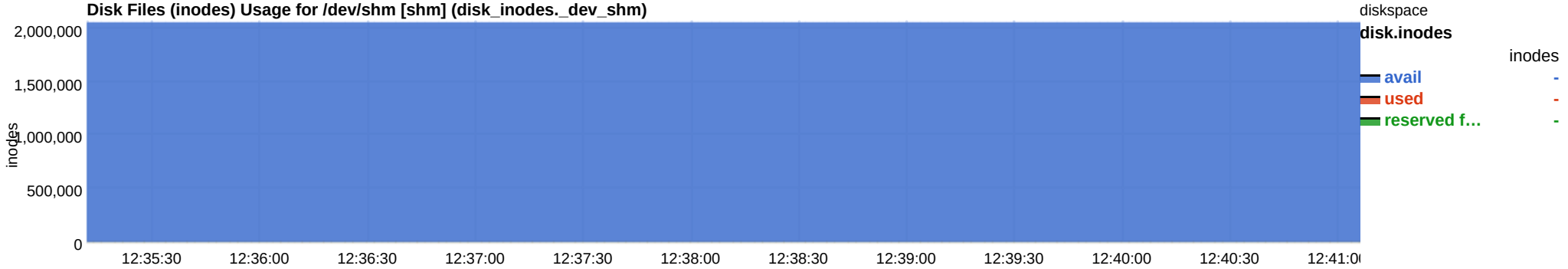


/dev/shm

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.

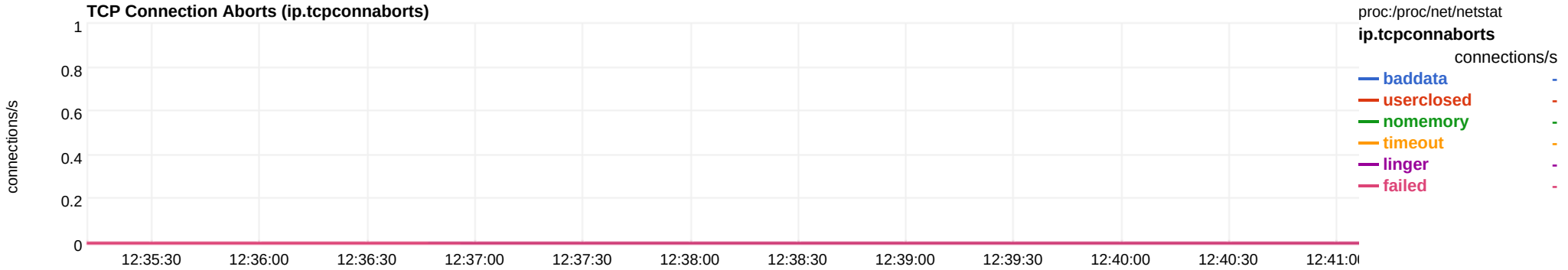


Networking Stack

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.

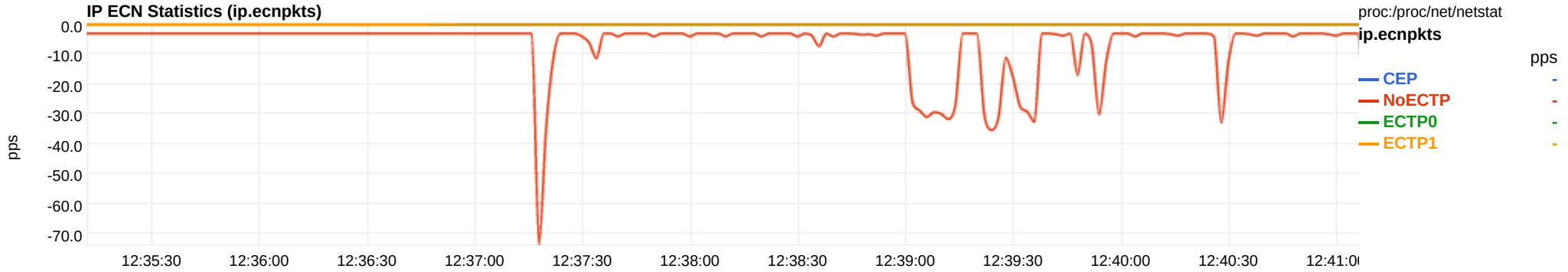
tcp

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.



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.

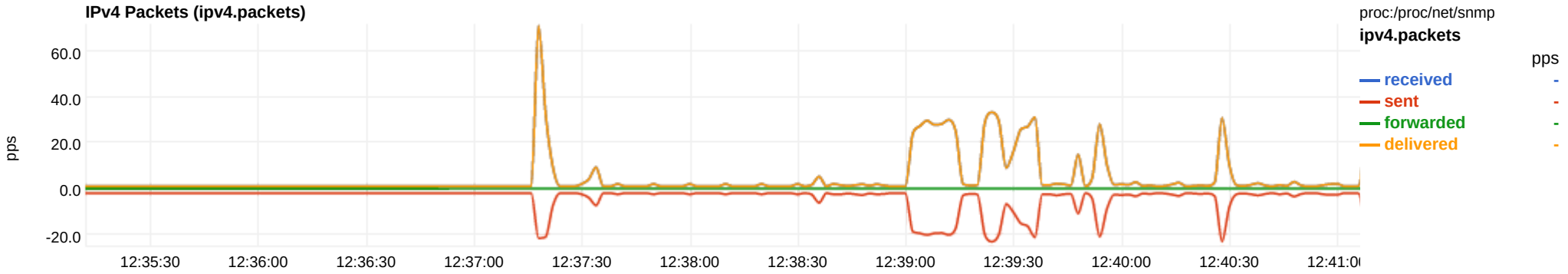


IPv4 Networking

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

packets

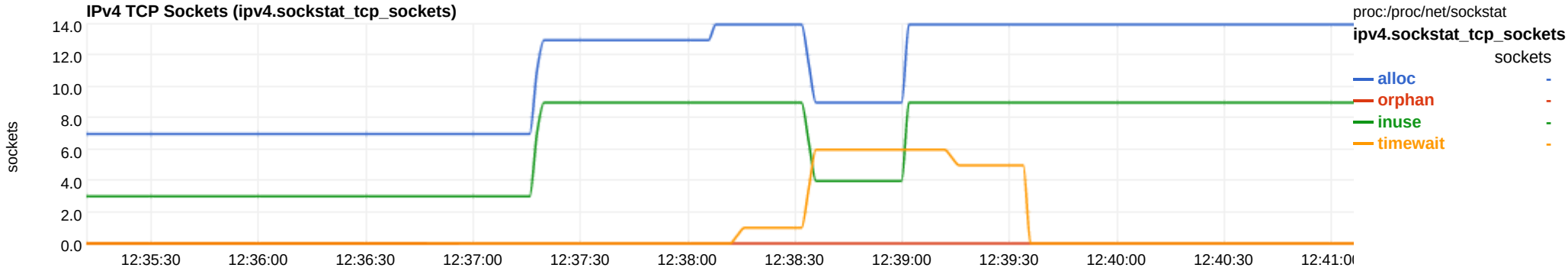


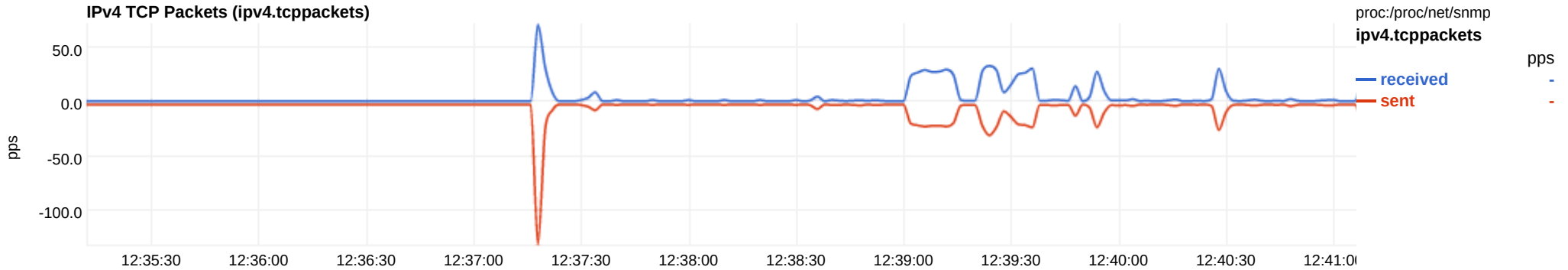
icmp


IPv4 ICMP Packets (ipv4.icmp)
proc:/proc/net/snmp
ipv4.icmp
pps
received -
sent -
pps
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
IPv4 ICMP Errors (ipv4.icmp_errors)
proc:/proc/net/snmp
ipv4.icmp_errors
pps
InErrors -
OutErrors -
InCsumEr… -
pps
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30

tcp

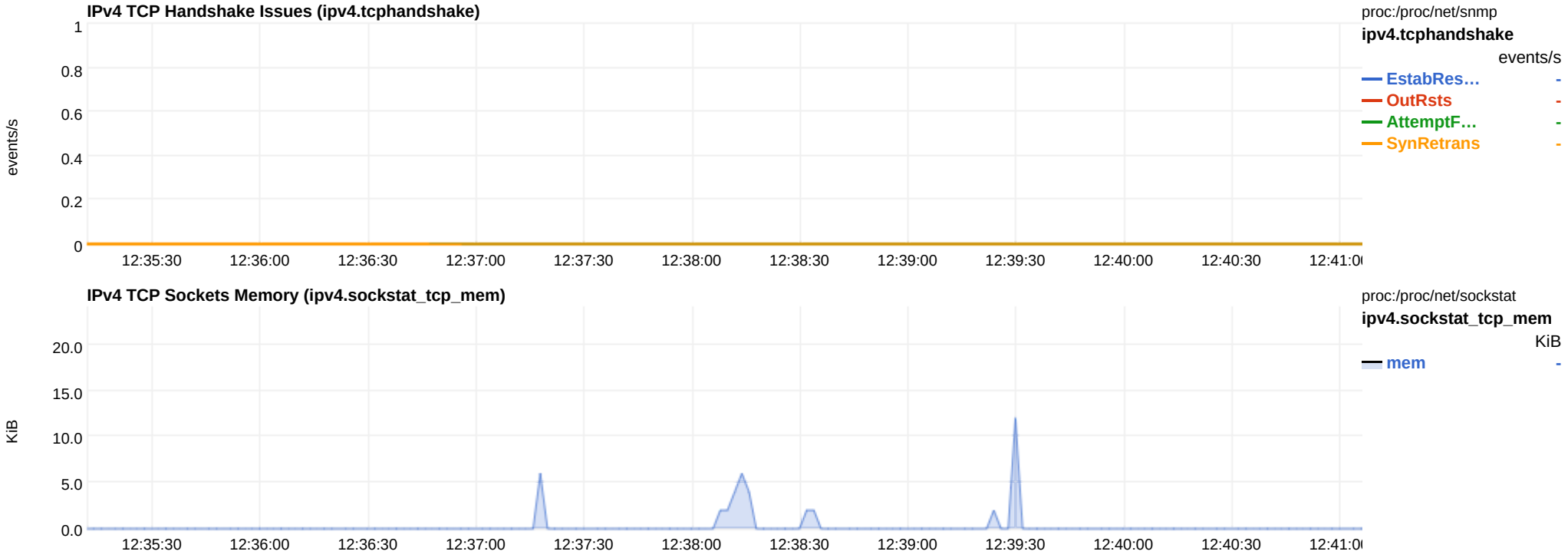
The number of established TCP connections (known as `CurrEstab`). This is a snapshot of the established connections at the time of measurement (i.e. a connection established and a connection disconnected within the same iteration will not affect this metric).





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.

`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.

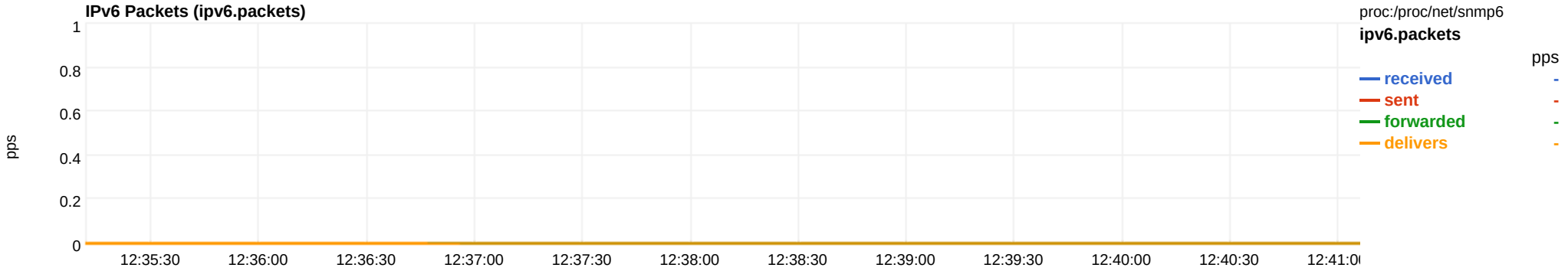


udp

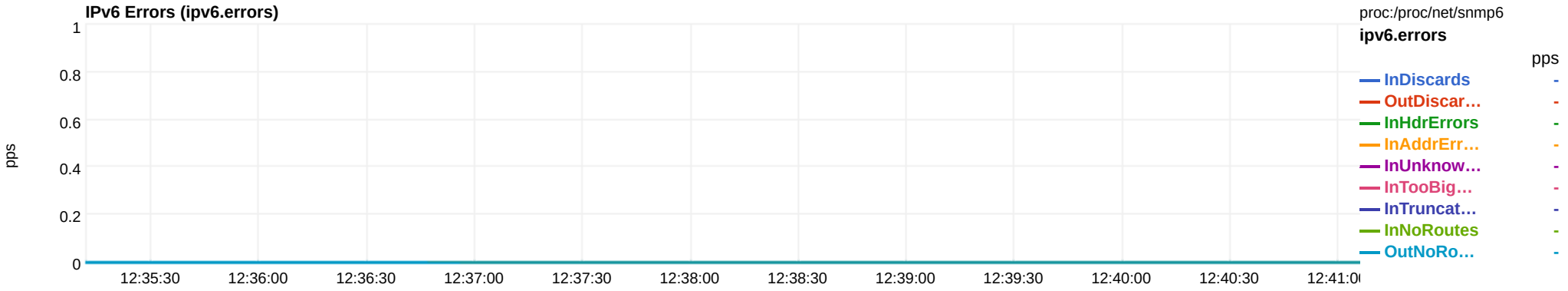
IPv6 Networking

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.

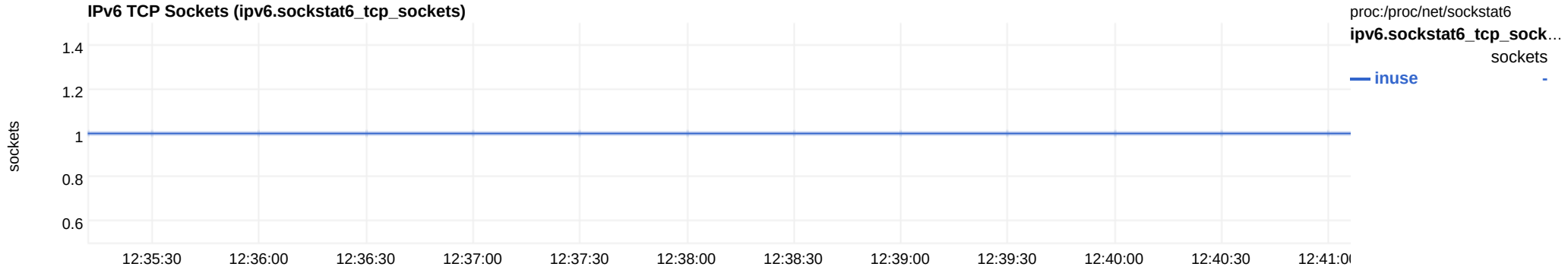
packets



errors



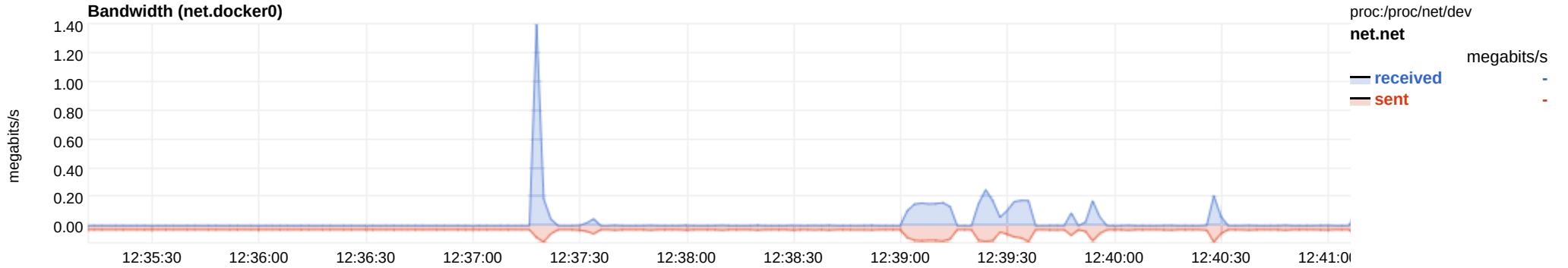
tcp6

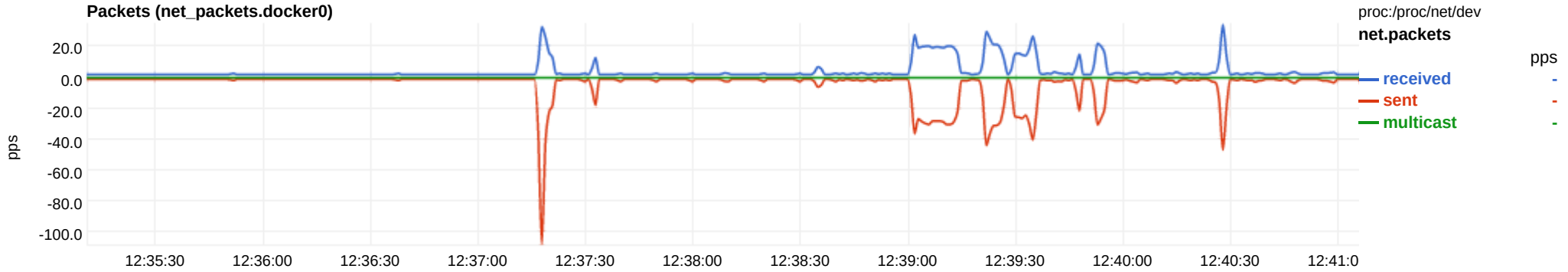


Network Interfaces

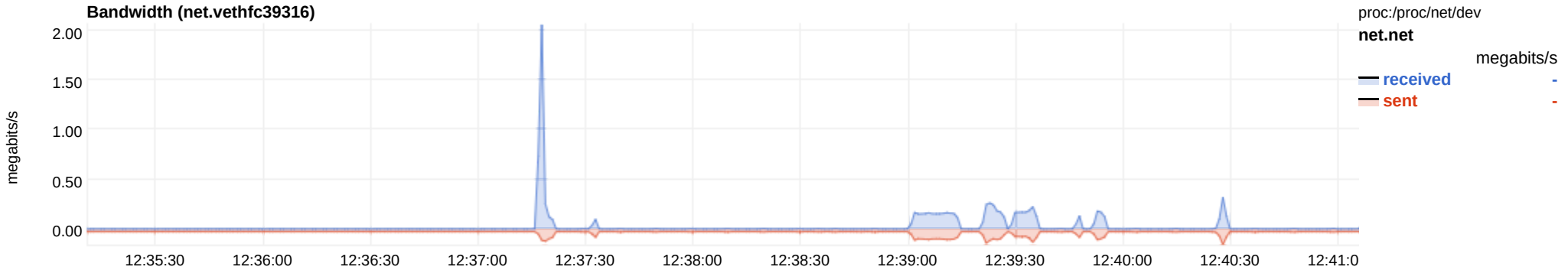
Performance metrics for network interfaces.

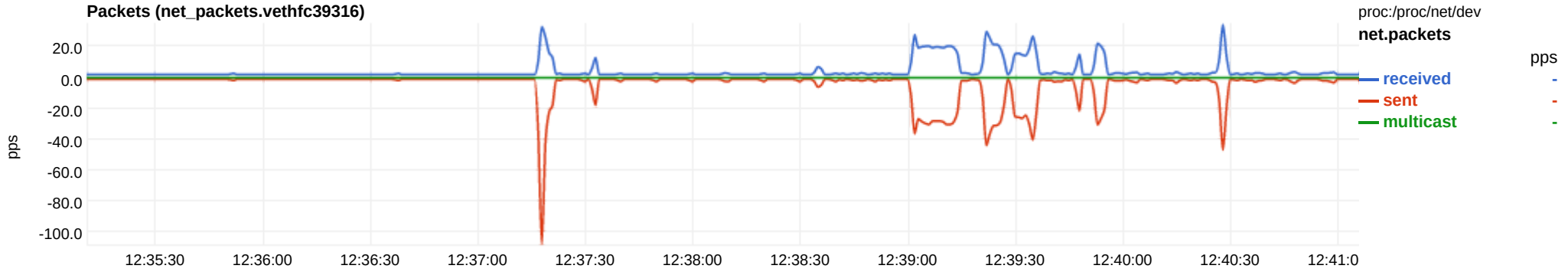
docker0





vethfc39316





virtlet-eth0

Firewall (netfilter)

Performance metrics of the netfilter components.

connection tracker

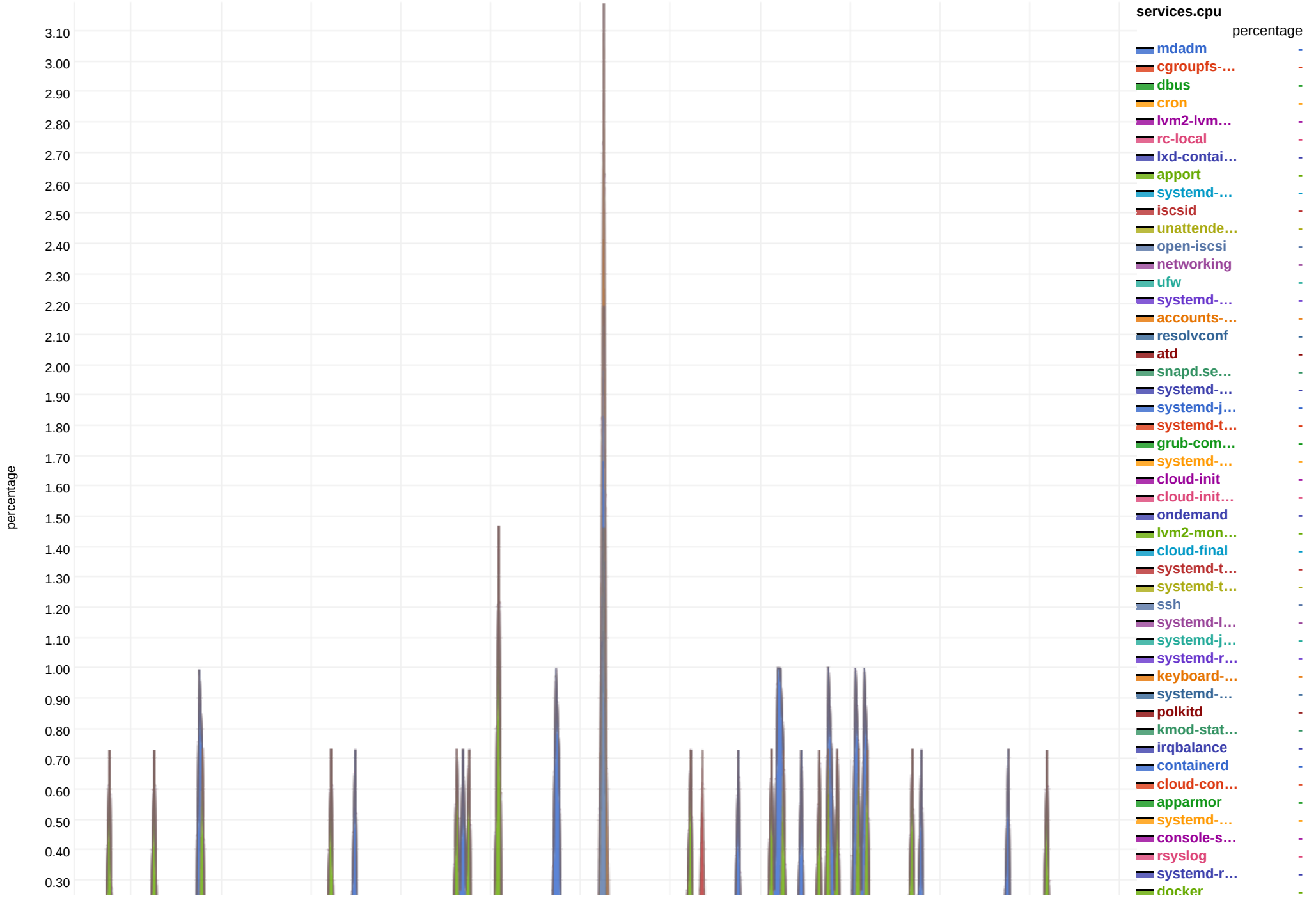
Netfilter Connection Tracker performance metrics. The connection tracker keeps track of all connections of the machine, inbound and outbound. It works by keeping a database with all open connections, tracking network and address translation and connection expectations.

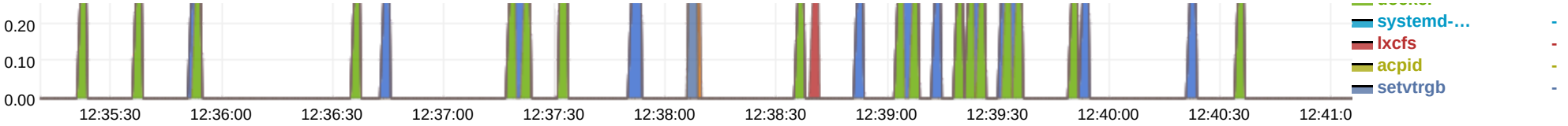
systemd Services

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

cpu

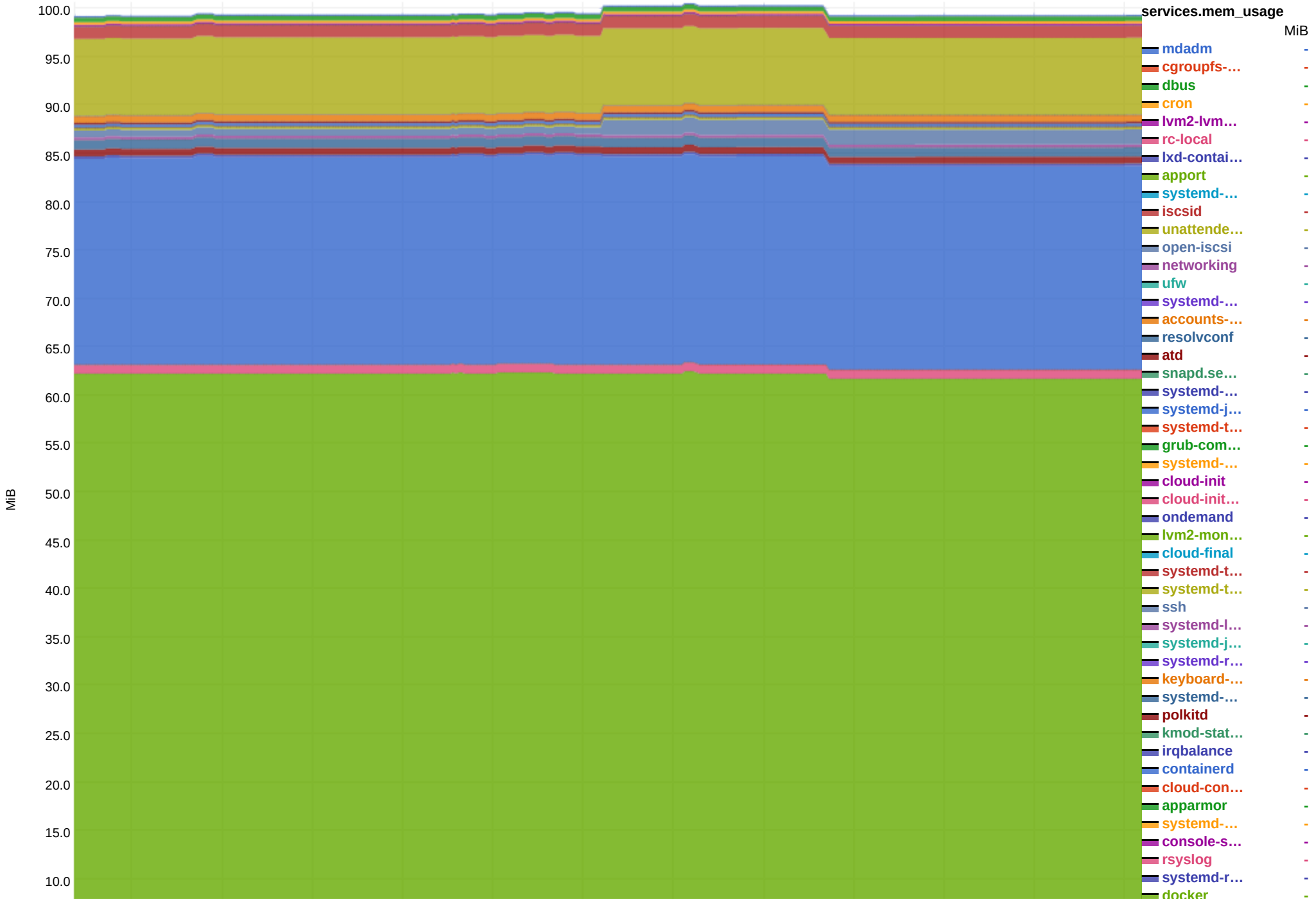
Systemd Services CPU utilization (400% = 4 cores) (services.cpu)

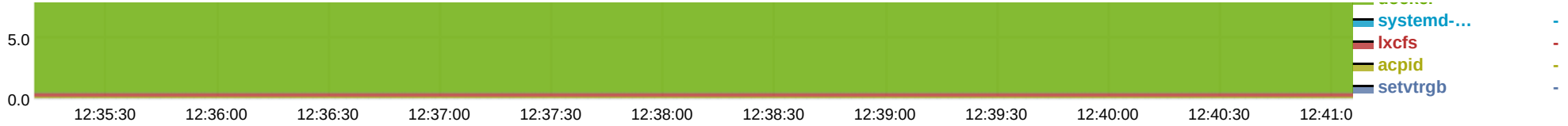



0.20
0.10
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
systemd-...
lxcfs
acpid
setvtrgb

mem

Systemd Services Used Memory without Cache (services.mem_usage)




5.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
systemd-…
lxcfs
acpid
setvtrgb

disk

Systemd Services Throttle Disk Read Bandwidth (services.throttle_io_read)


0.05
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
systemd-…
lxcfs
acpid
setvtrgb

Systemd Services Throttle Disk Write Bandwidth (services.throttle_io_write)


0.05
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
systemd-…
lxcfs
acpid
setvtrgb

Systemd Services Throttle Disk Read Operations (services.throttle_io_ops_read)

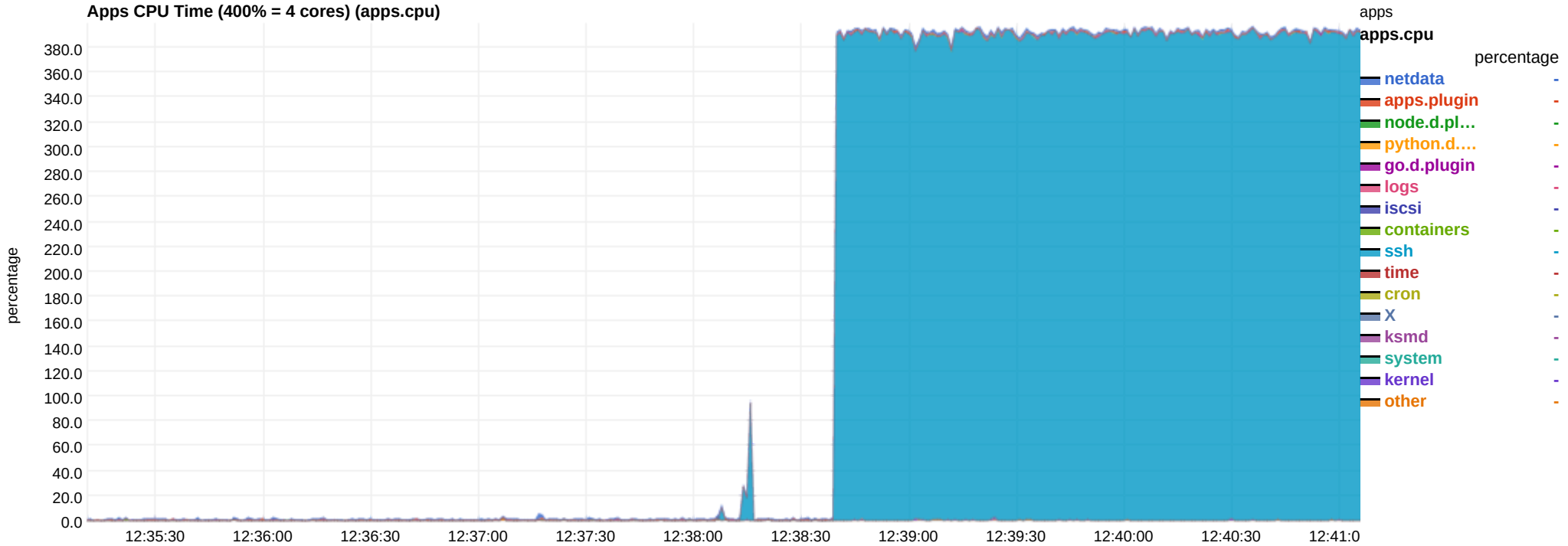

0.05
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
systemd-…
lxcfs
acpid
setvtrgb

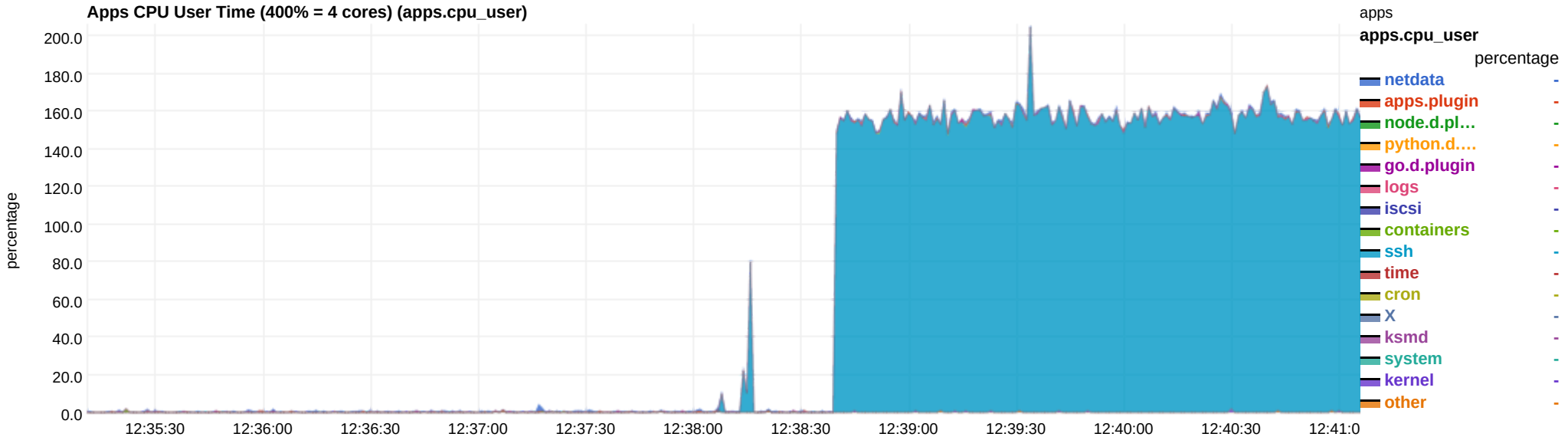
Systemd Services Throttle Disk Write Operations (services.throttle_io_ops_write)

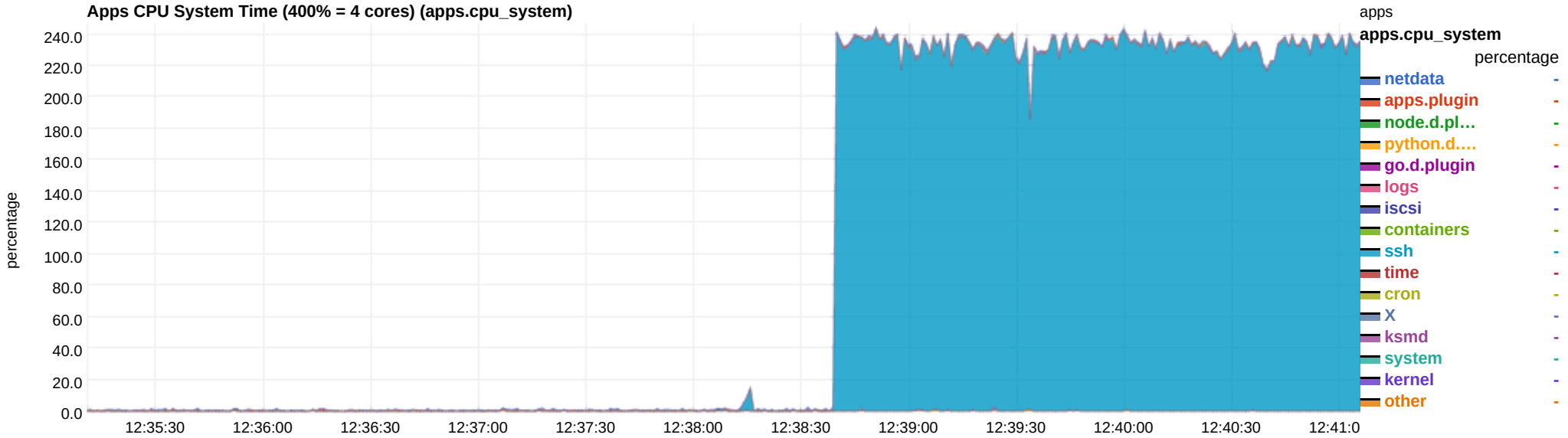
Applications

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.

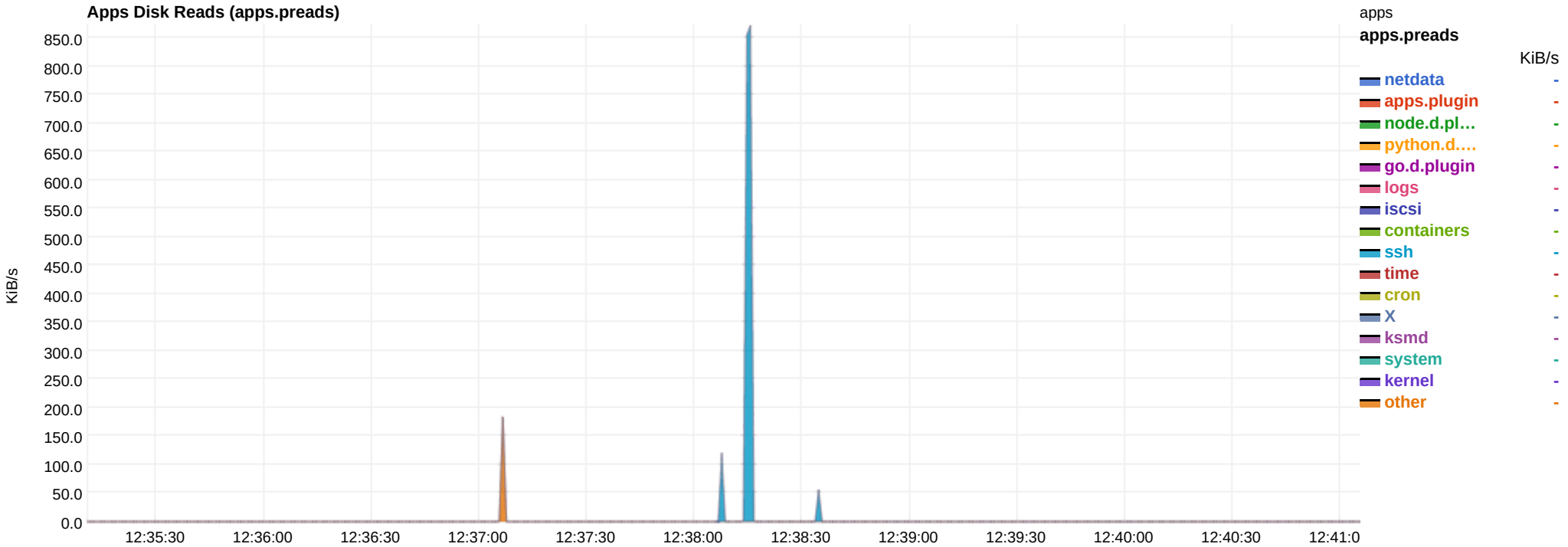
cpu


Apps CPU Time (400% = 4 cores) (apps.cpu)
percentage
380.0
360.0
340.0
320.0
300.0
280.0
260.0
240.0
220.0
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.cpu
percentage
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other

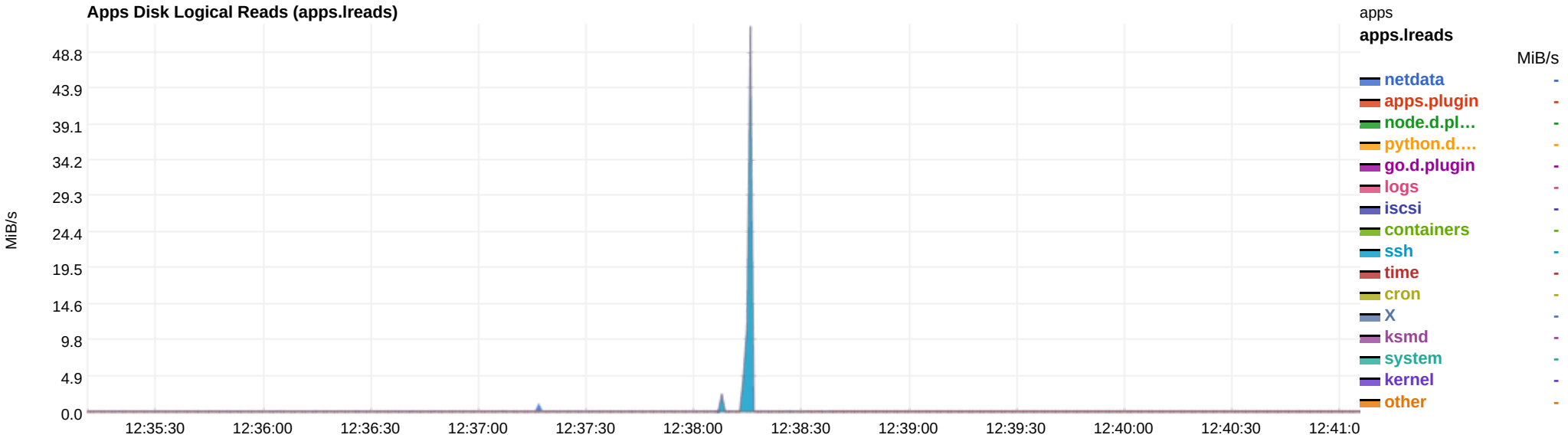

Apps CPU User Time (400% = 4 cores) (apps.cpu_user)
percentage
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.cpu_user
percentage
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other

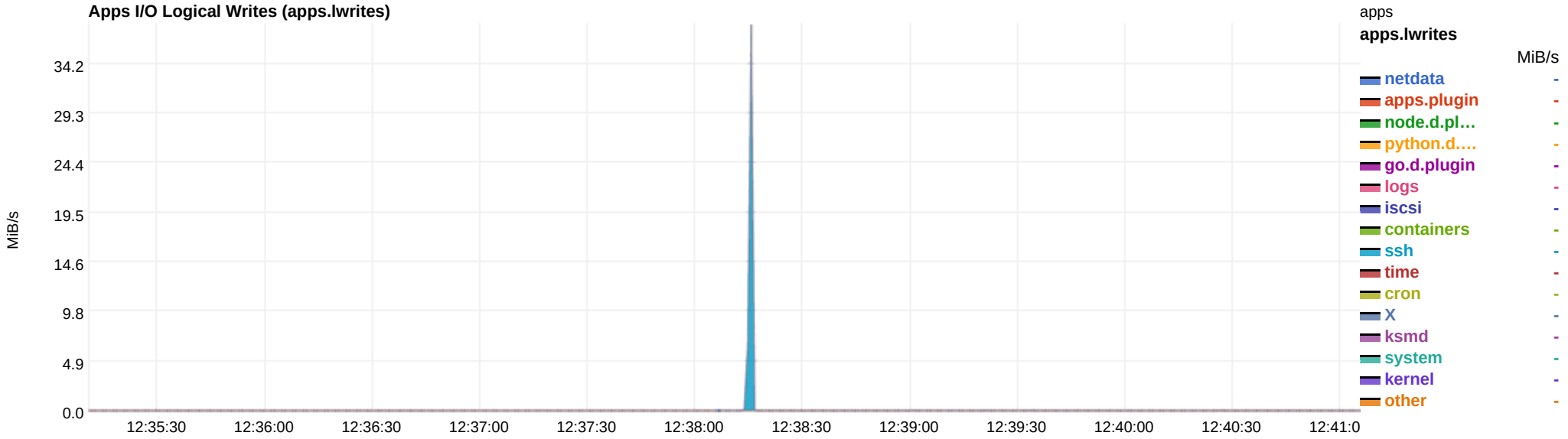

Apps CPU System Time (400% = 4 cores) (apps.cpu_system)
percentage
240.0
220.0
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.cpu_system
percentage
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other

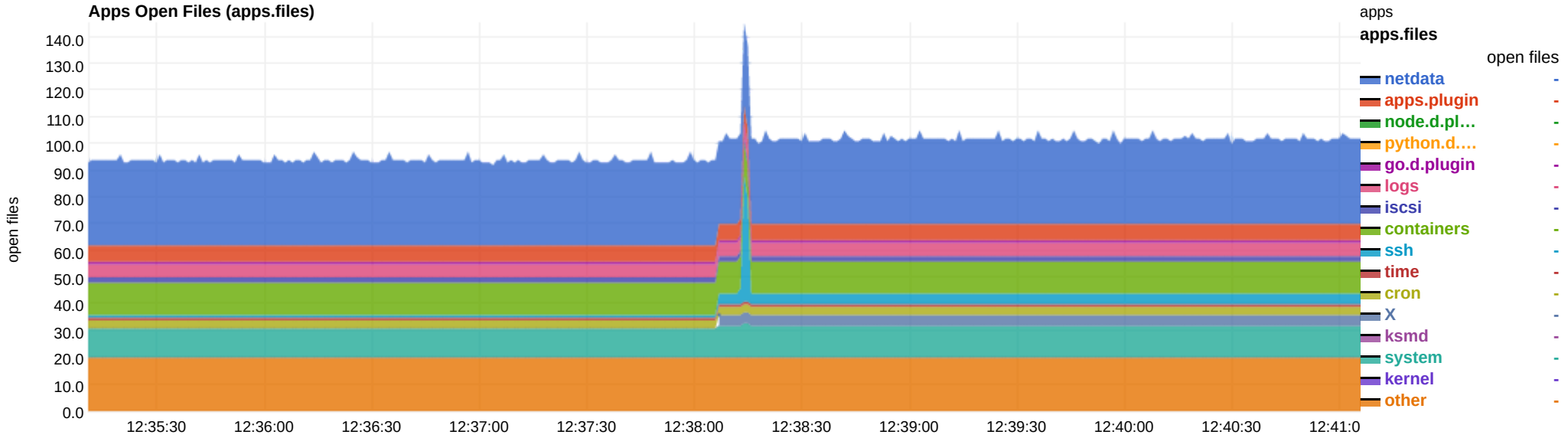
disk


Apps Disk Reads (apps.preads)
KiB/s
850.0
800.0
750.0
700.0
650.0
600.0
550.0
500.0
450.0
400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.preads
KiB/s
netdata -
apps.plugin -
node.d.pl… -
python.d.… -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -


Apps Disk Writes (apps.pwrites)
MiB/s
33.2
31.3
29.3
27.3
25.4
23.4
21.5
19.5
17.6
15.6
13.7
11.7
9.8
7.8
5.9
3.9
2.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.pwrites
MiB/s
netdata -
apps.plugin -
node.d.pl... -
python.d.... -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -

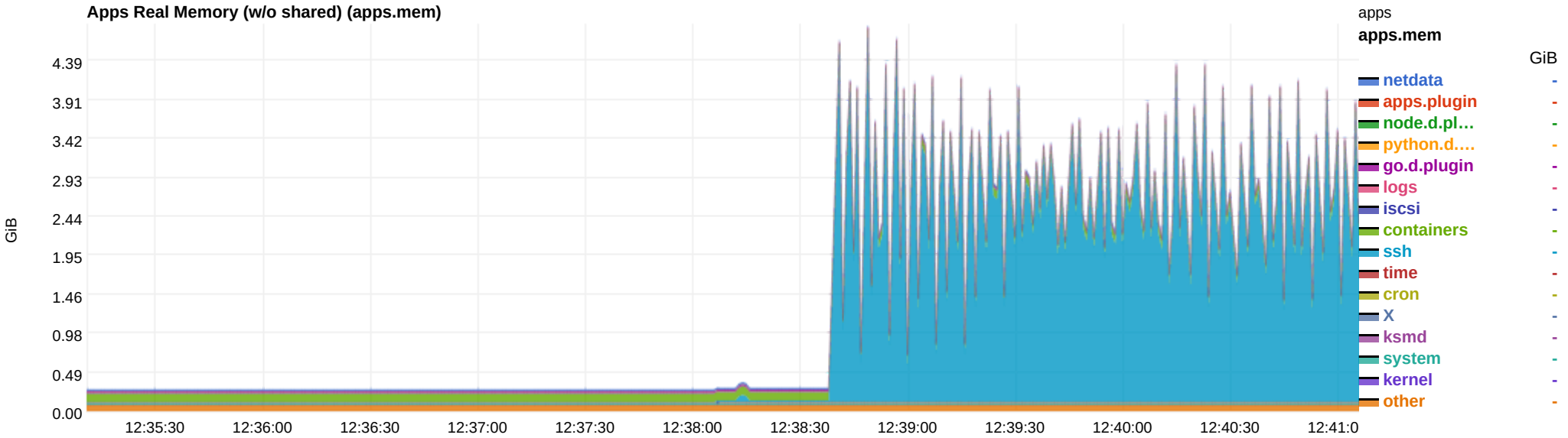

Apps Disk Logical Reads (apps.lreads)
MiB/s
48.8
43.9
39.1
34.2
29.3
24.4
19.5
14.6
9.8
4.9
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.lreads
MiB/s
netdata -
apps.plugin -
node.d.pl... -
python.d.... -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -


Apps I/O Logical Writes (apps.lwrites)
MiB/s
34.2
29.3
24.4
19.5
14.6
9.8
4.9
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.lwrites
MiB/s
netdata -
apps.plugin -
node.d.pl... -
python.d.... -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -

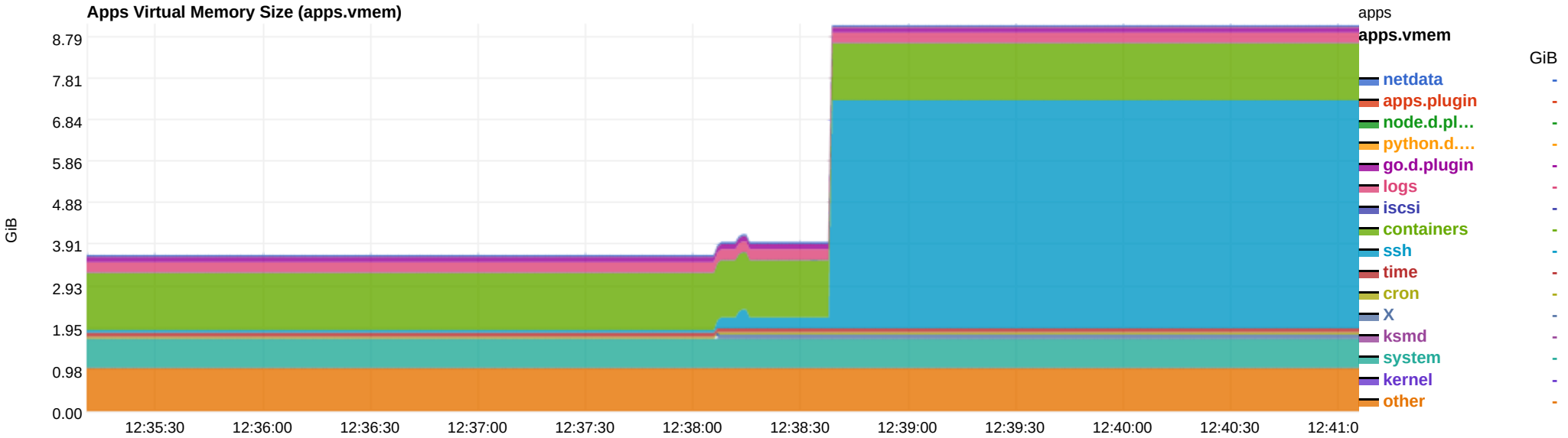

Apps Open Files (apps.files)
open files
140.0
130.0
120.0
110.0
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.files
open files
netdata -
apps.plugin -
node.d.pl... -
python.d.... -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -

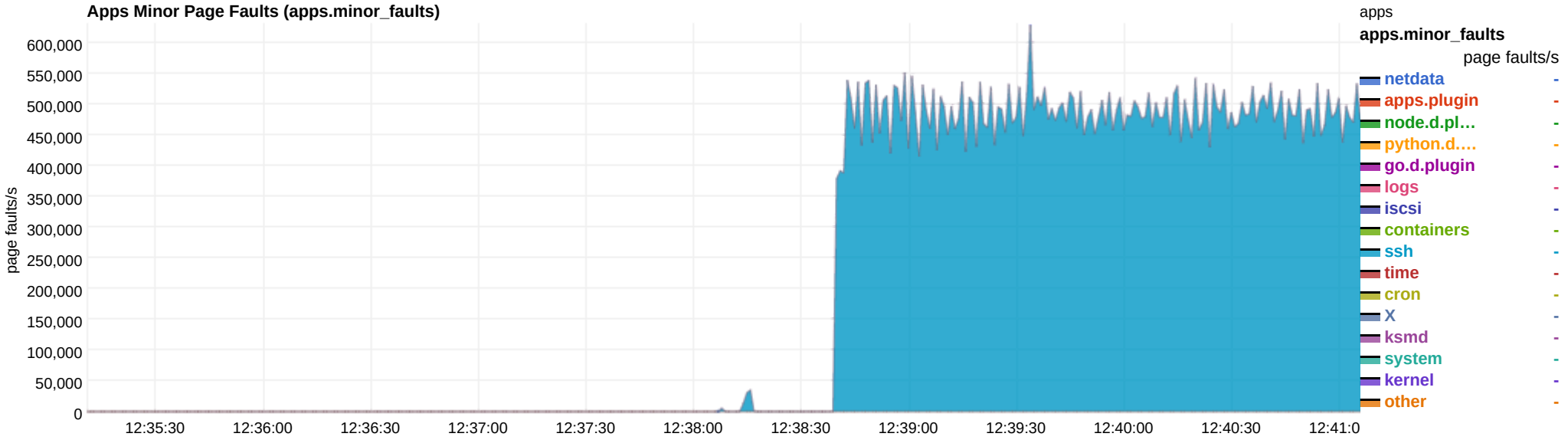
mem

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

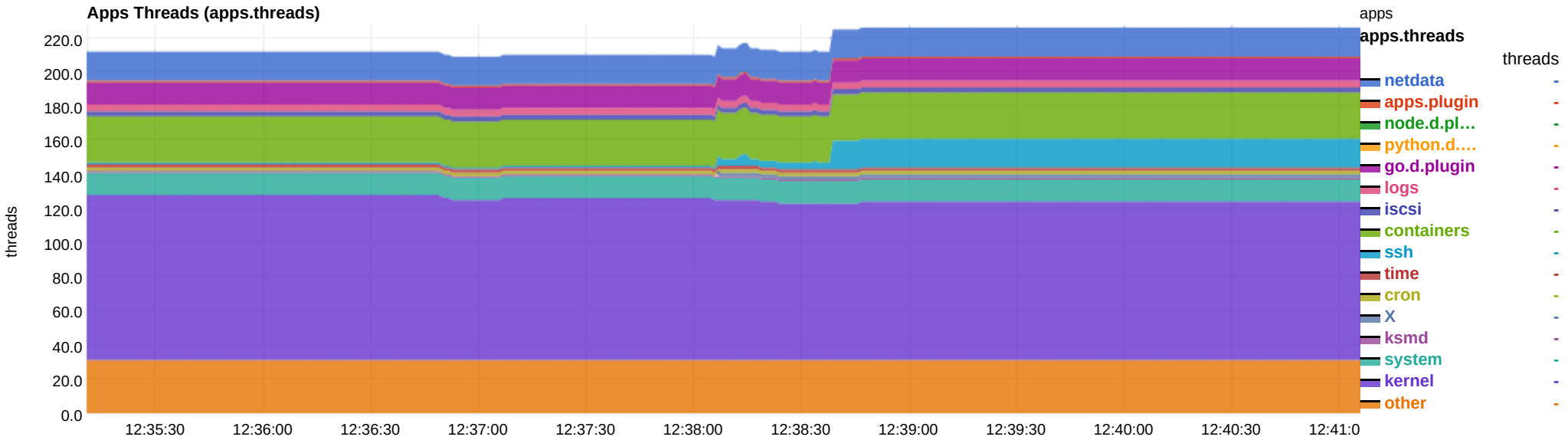


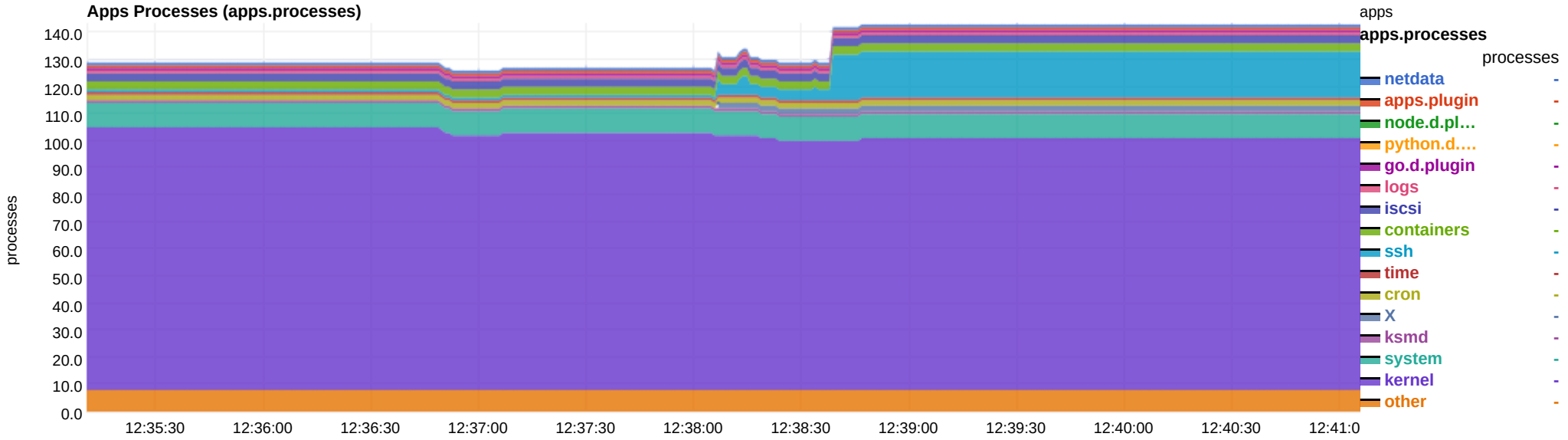
Virtual memory allocated by applications. Please check this article (https://github.com/netdata/netdata/tree/master/daemon#virtual-memory) for more information.



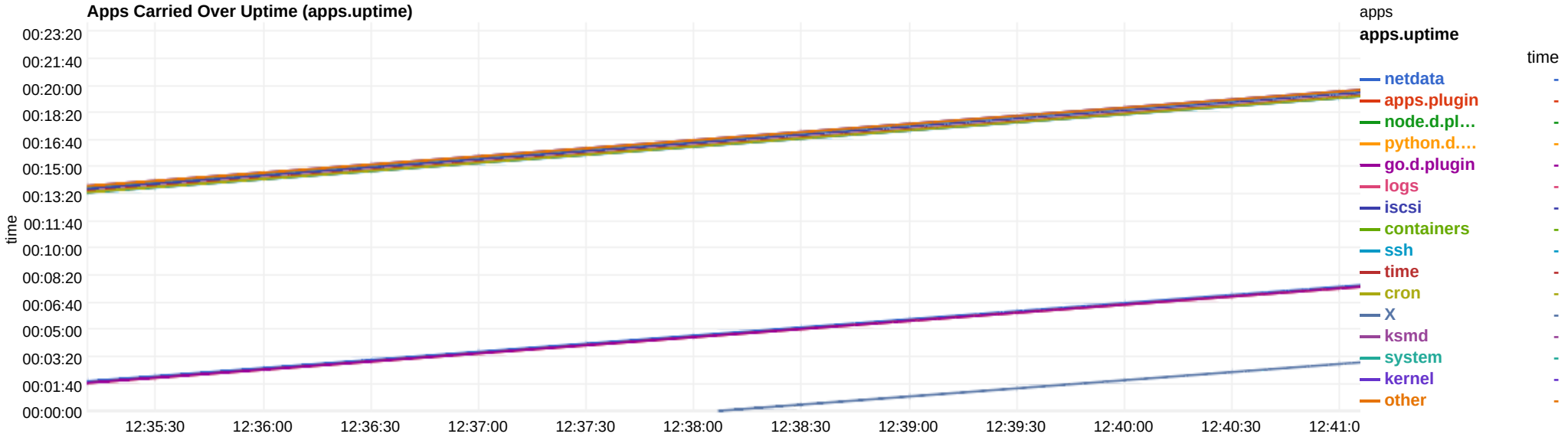

Apps Minor Page Faults (apps.minor_faults)
page faults/s
600,000
550,000
500,000
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.minor_faults
page faults/s
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other

processes

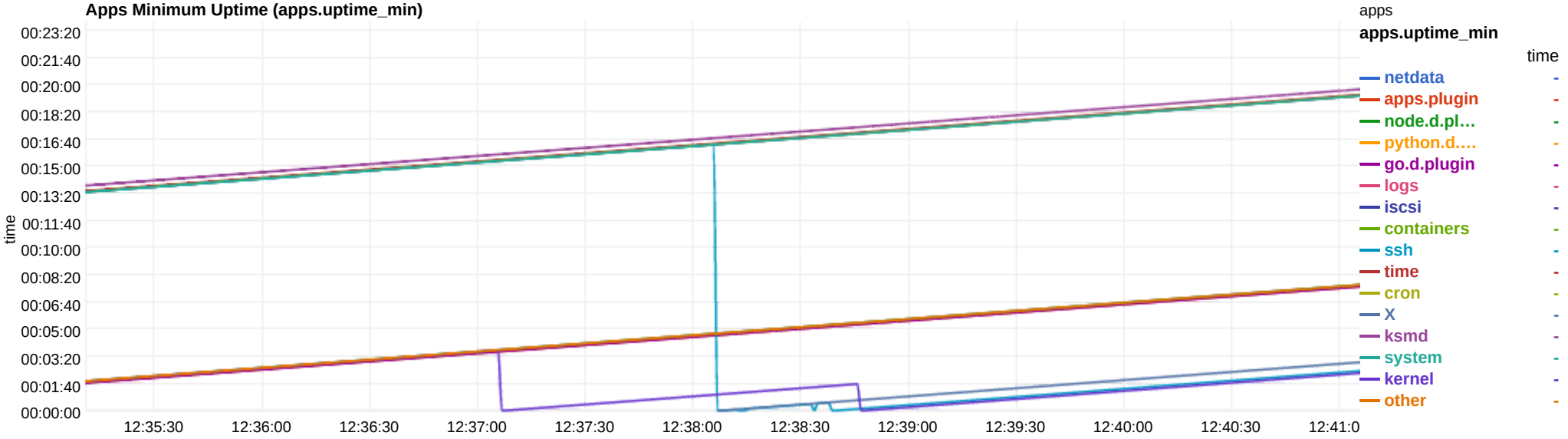

Apps Threads (apps.threads)
threads
220.0
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.threads
threads
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other



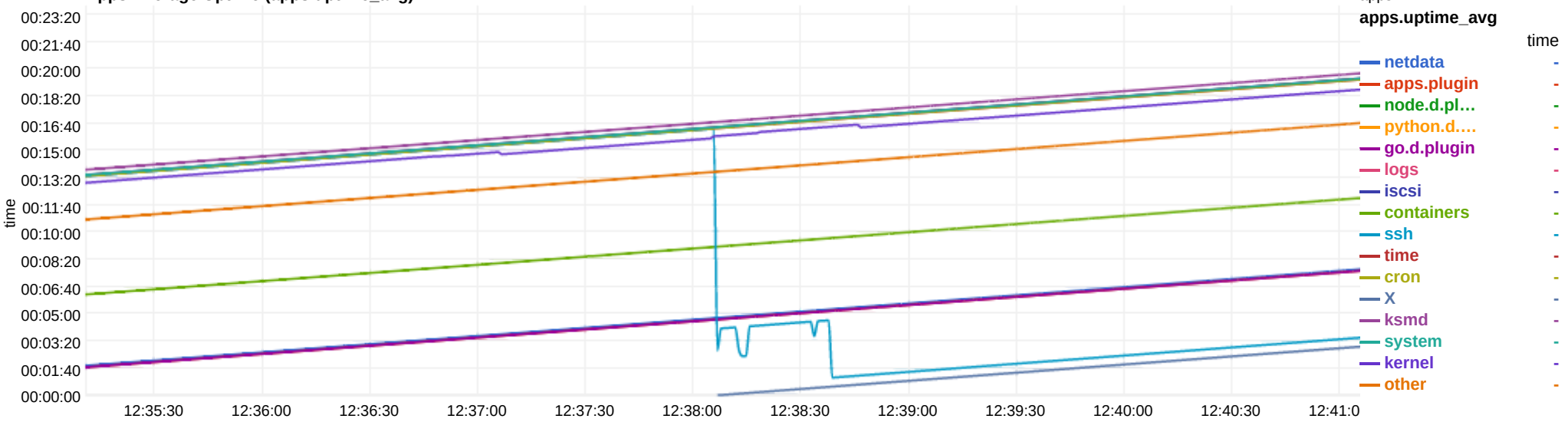
Carried over process group uptime since the Netdata restart. The period of time within which at least one process in the group was running.

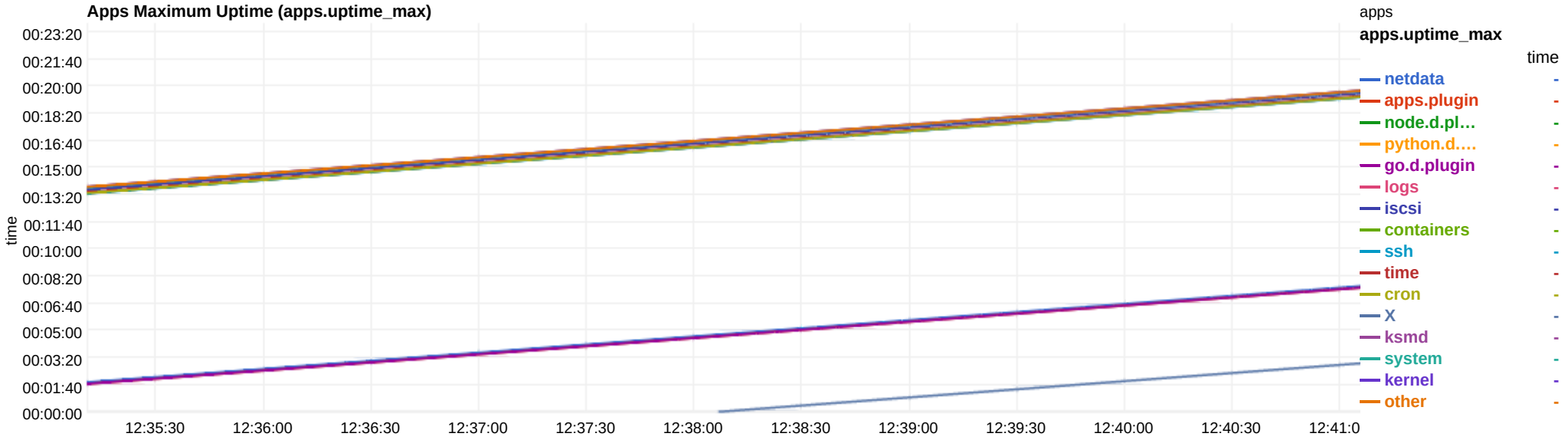


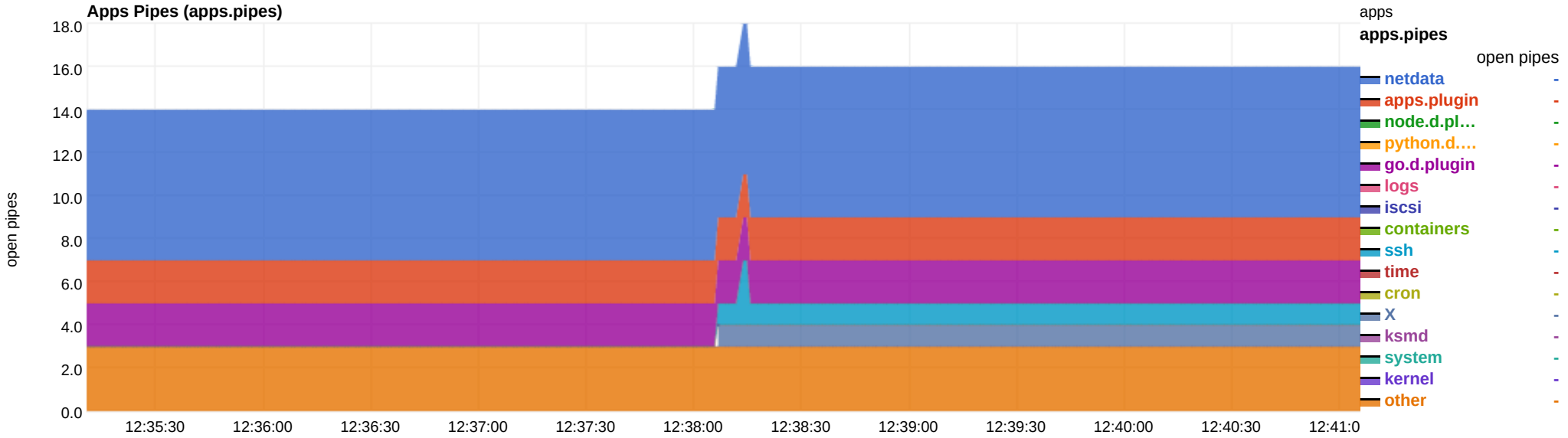
Apps Minimum Uptime (apps.uptime_min)



Apps Average Uptime (apps.uptime_avg)

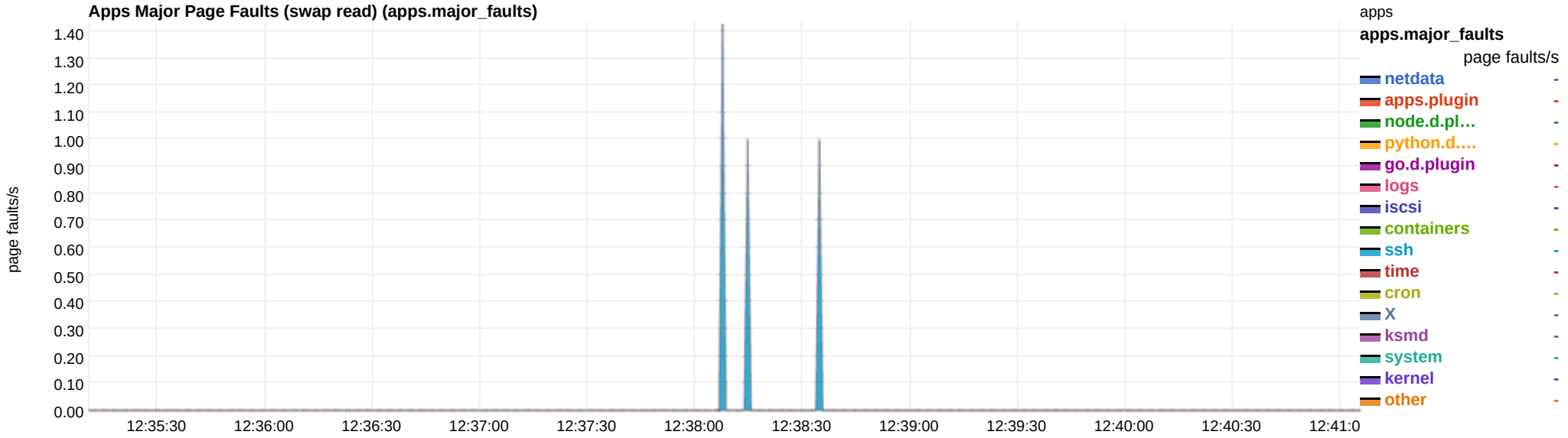



Apps Maximum Uptime (apps.uptime_max)
time
00:23:20
00:21:40
00:20:00
00:18:20
00:16:40
00:15:00
00:13:20
00:11:40
00:10:00
00:08:20
00:06:40
00:05:00
00:03:20
00:01:40
00:00:00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.uptime_max
time
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other

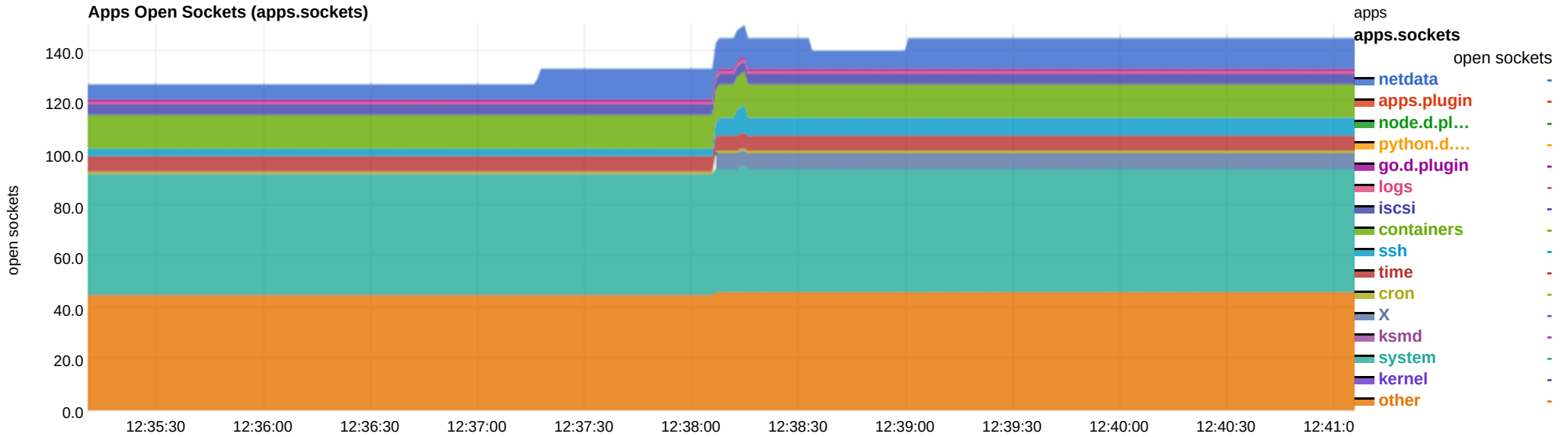

Apps Pipes (apps.pipes)
open pipes
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.pipes
open pipes
netdata
apps.plugin
node.d.pl…
python.d.…
go.d.plugin
logs
iscsi
containers
ssh
time
cron
X
ksmd
system
kernel
other

swap


Apps Swap Memory (apps.swap)
MiB
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.swap
MiB
netdata -
apps.plugin -
node.d.pl… -
python.d.… -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -


Apps Major Page Faults (swap read) (apps.major_faults)
page faults/s
1.40
1.30
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
apps.major_faults
page faults/s
netdata -
apps.plugin -
node.d.pl… -
python.d.… -
go.d.plugin -
logs -
iscsi -
containers -
ssh -
time -
cron -
X -
ksmd -
system -
kernel -
other -

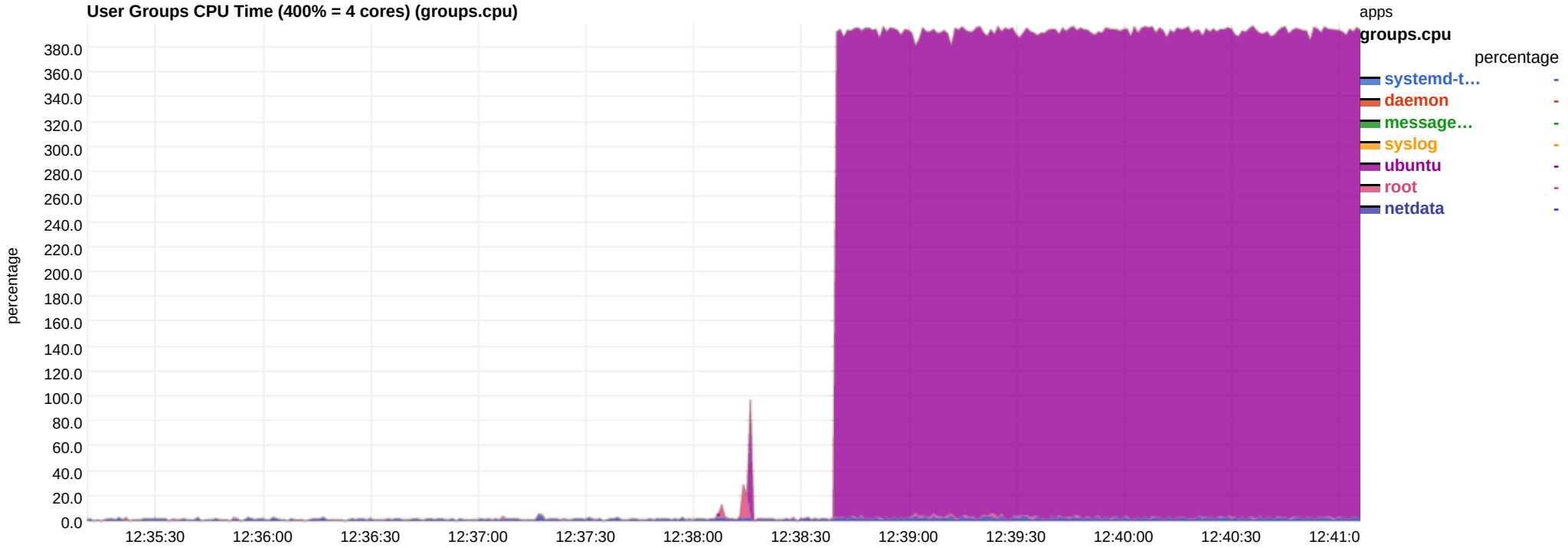
net

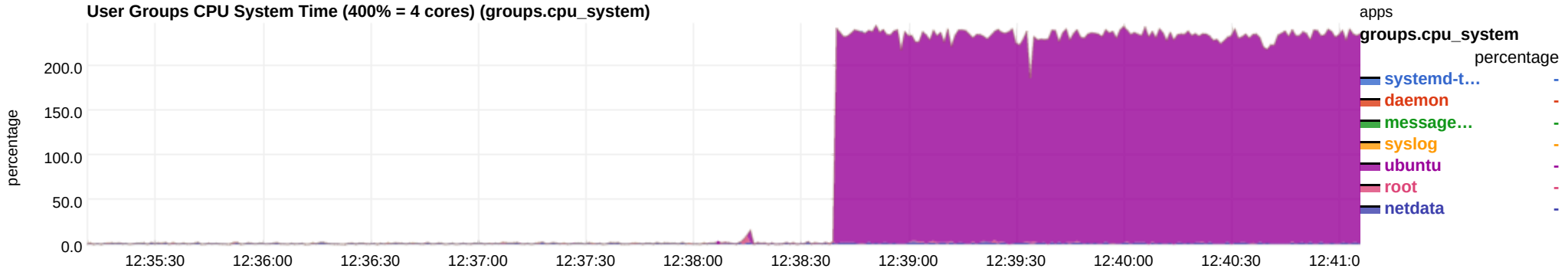


User Groups

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

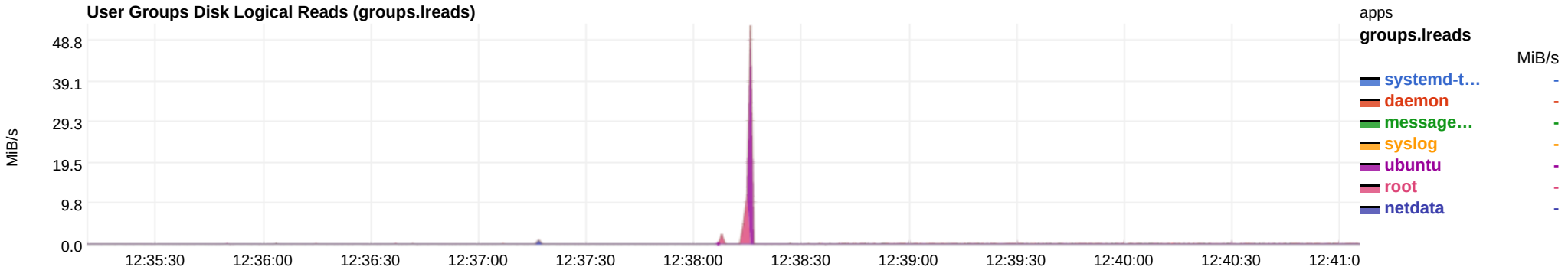

User Groups CPU Time (400% = 4 cores) (groups.cpu)
percentage
380.0
360.0
340.0
320.0
300.0
280.0
260.0
240.0
220.0
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.cpu
percentage
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -
User Groups CPU User Time (400% = 4 cores) (groups.cpu_user)
percentage
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.cpu_user
percentage
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

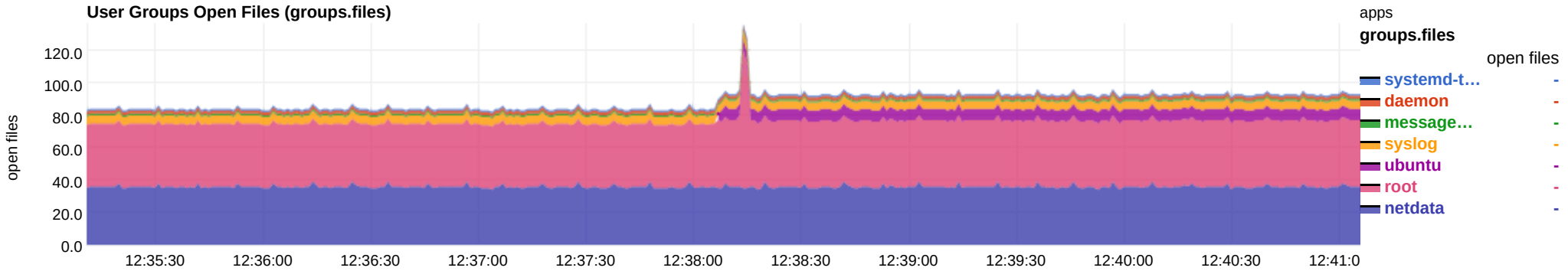

User Groups CPU System Time (400% = 4 cores) (groups.cpu_system)
percentage
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.cpu_system
percentage
systemd-t…
daemon
message…
syslog
ubuntu
root
netdata

disk


User Groups Disk Reads (groups.preads)
KiB/s
850.0
800.0
750.0
700.0
650.0
600.0
550.0
500.0
450.0
400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.preads
KiB/s
systemd-t…
daemon
message…
syslog
ubuntu
root
netdata

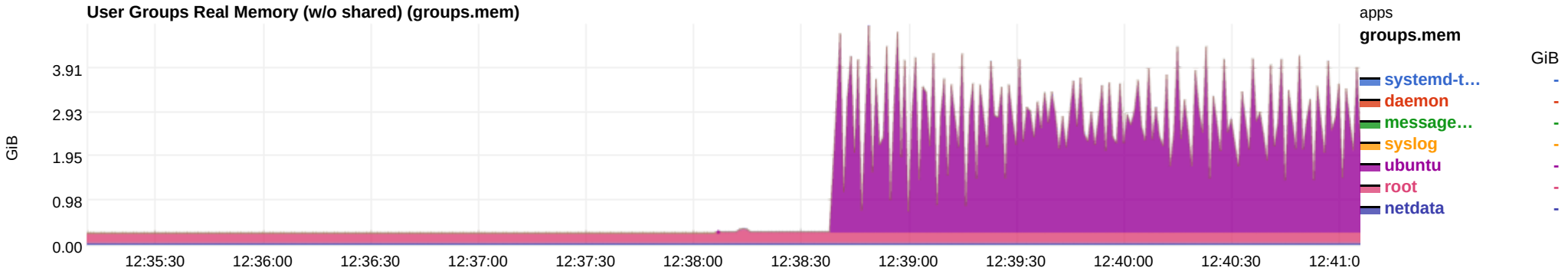

User Groups Disk Writes (groups.pwrites)
MiB/s
33.2
31.3
29.3
27.3
25.4
23.4
21.5
19.5
17.6
15.6
13.7
11.7
9.8
7.8
5.9
3.9
2.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.pwrites
MiB/s
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -


User Groups Disk Logical Reads (groups.lreads)
MiB/s
48.8
39.1
29.3
19.5
9.8
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.lreads
MiB/s
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

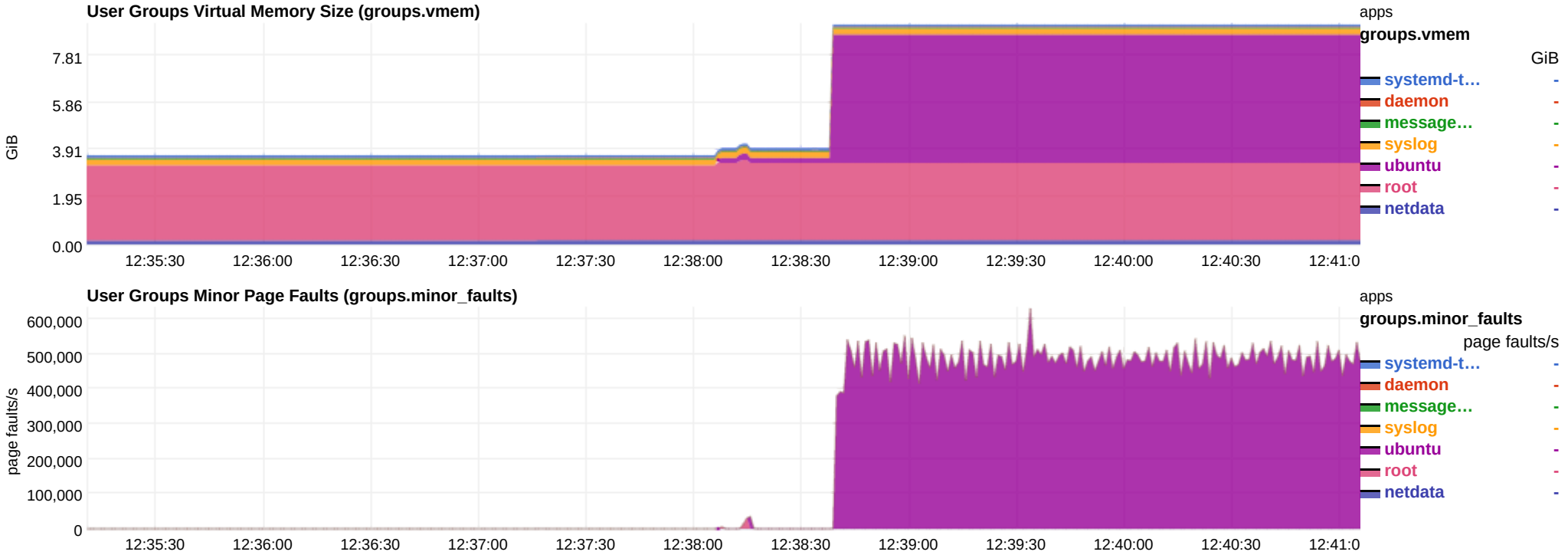


mem

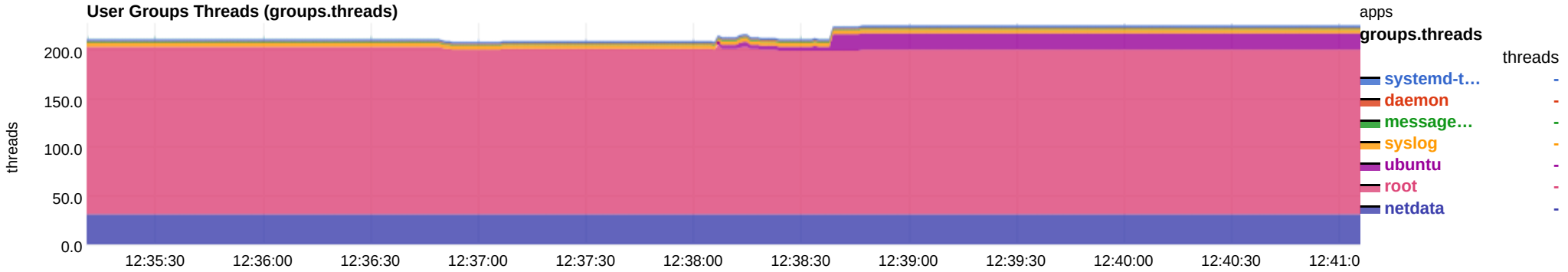
Real memory (RAM) used per user group. This does not include shared memory.



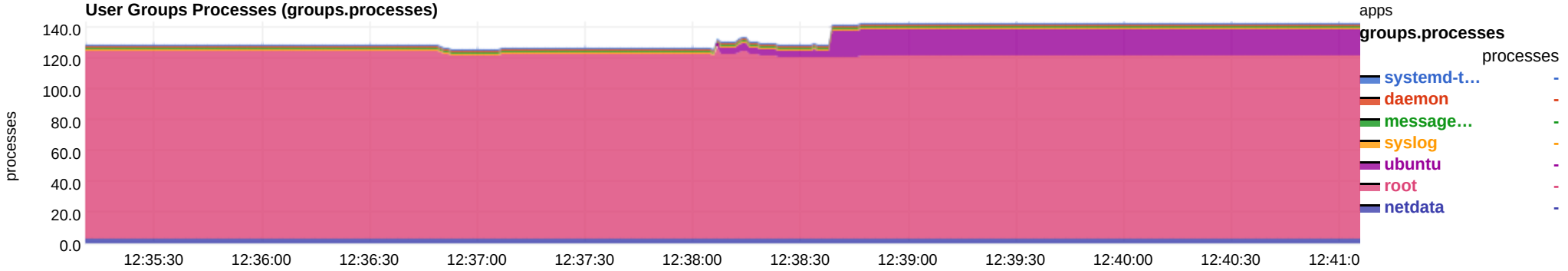
Virtual memory allocated per user group since the Netdata restart. Please check this article (https://github.com/netdata/netdata/tree/master/daemon#virtual-memory) for more information.



processes

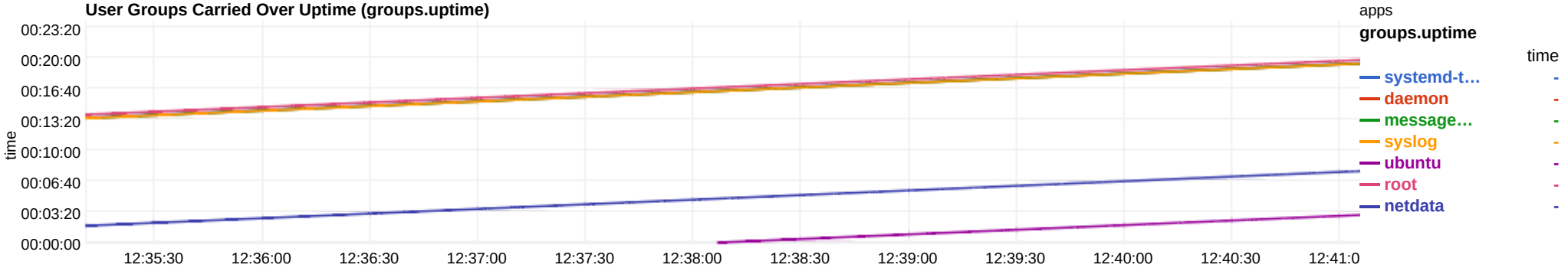


User Groups Processes (groups.processes)

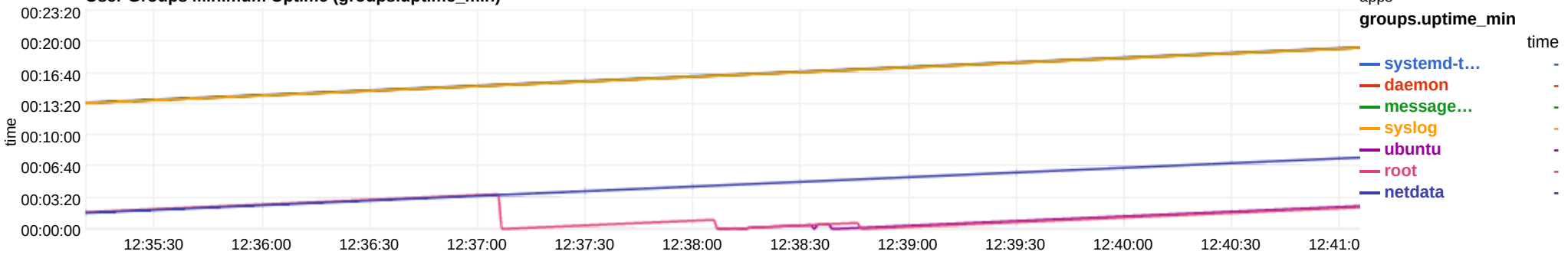


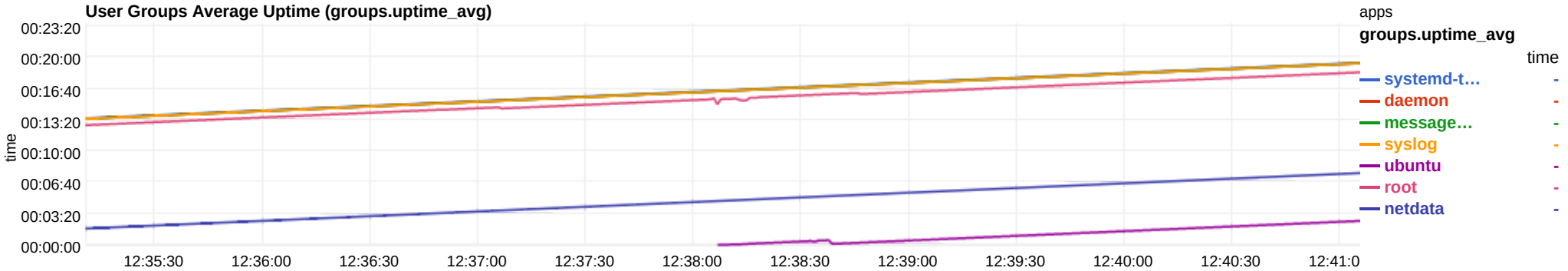
Carried over process group uptime. The period of time within which at least one process in the group was running.

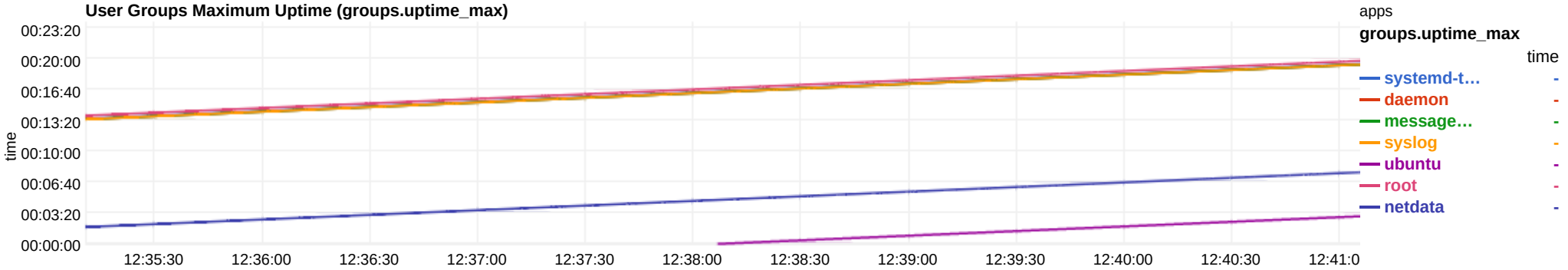
User Groups Carried Over Uptime (groups.uptime)

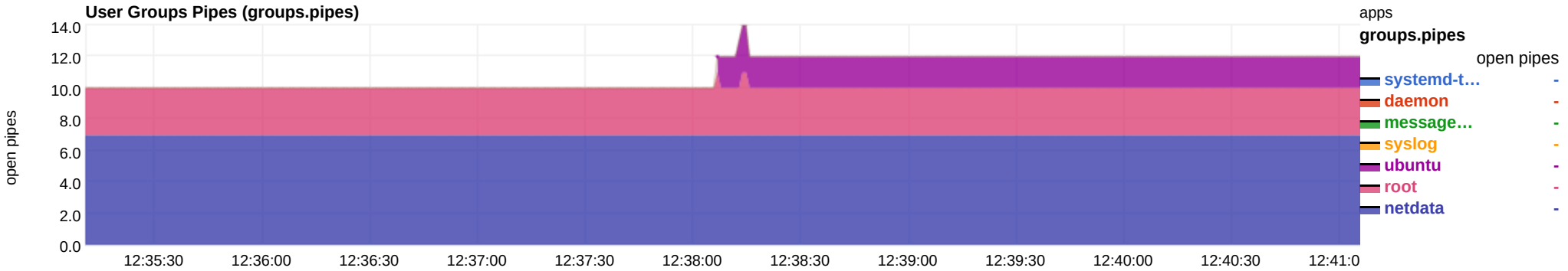


User Groups Minimum Uptime (groups.uptime_min)

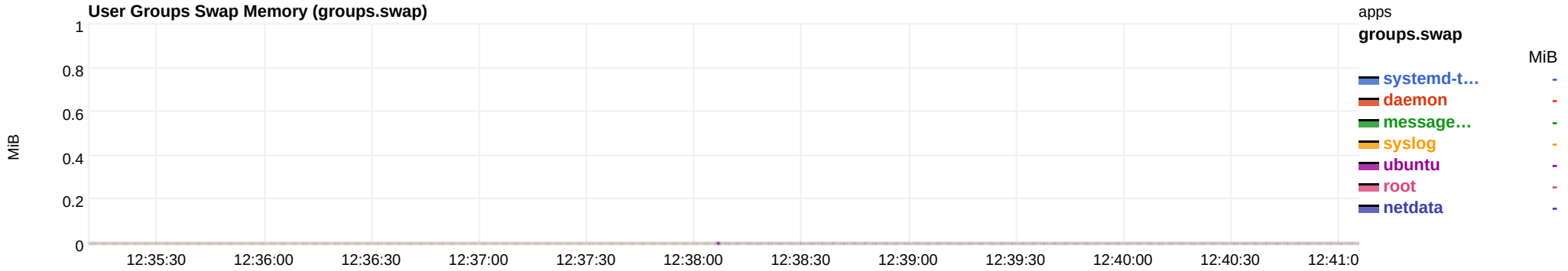



User Groups Average Uptime (groups.uptime_avg)
time
00:23:20
00:20:00
00:16:40
00:13:20
00:10:00
00:06:40
00:03:20
00:00:00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.uptime_avg
time
systemd-t…
daemon
message…
syslog
ubuntu
root
netdata


User Groups Maximum Uptime (groups.uptime_max)
time
00:23:20
00:20:00
00:16:40
00:13:20
00:10:00
00:06:40
00:03:20
00:00:00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.uptime_max
time
systemd-t…
daemon
message…
syslog
ubuntu
root
netdata

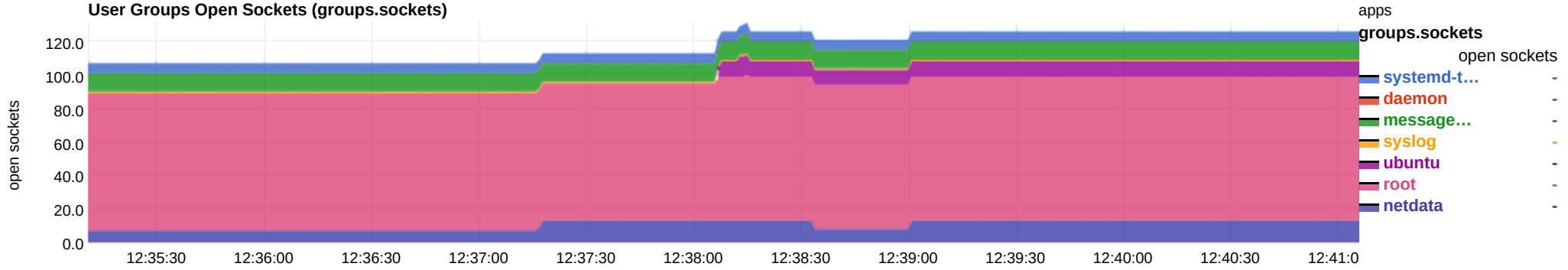

User Groups Pipes (groups.pipes)
open pipes
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.pipes
open pipes
systemd-t…
daemon
message…
syslog
ubuntu
root
netdata

swap


User Groups Swap Memory (groups.swap)
MiB
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.swap
MiB
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -


User Groups Major Page Faults (swap read) (groups.major_faults)
page faults/s
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.major_faults
page faults/s
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

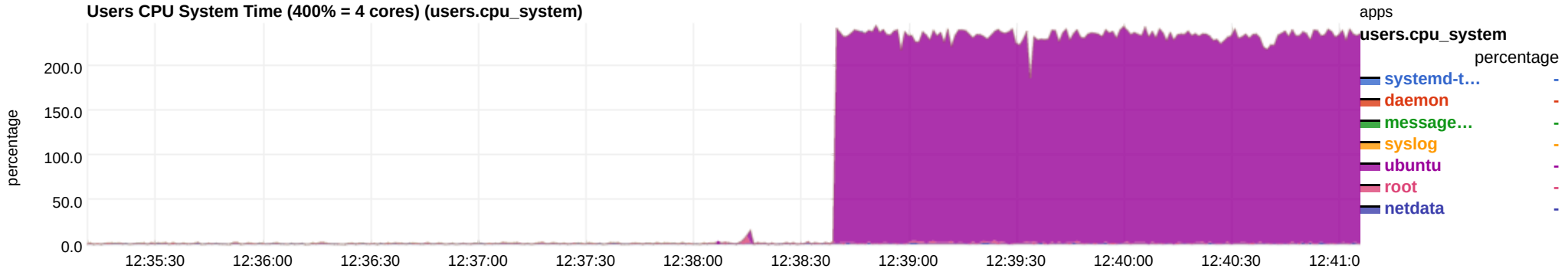
net


User Groups Open Sockets (groups.sockets)
open sockets
120.0
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
groups.sockets
open sockets
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

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.

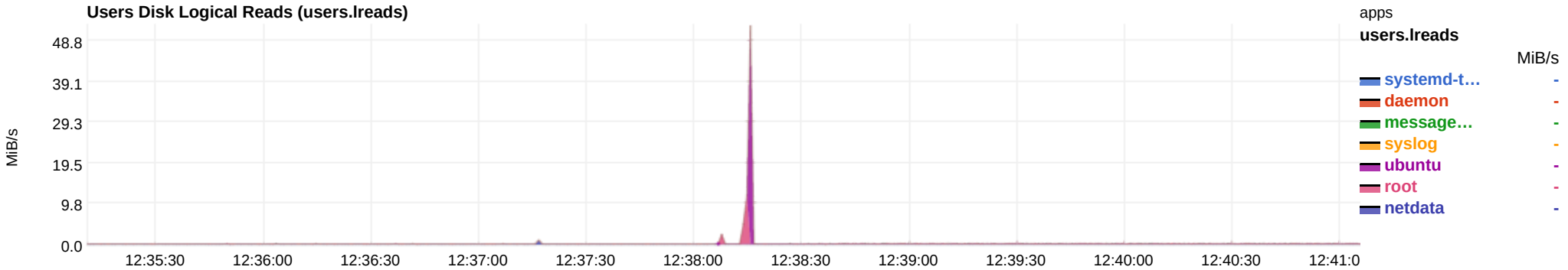
cpu

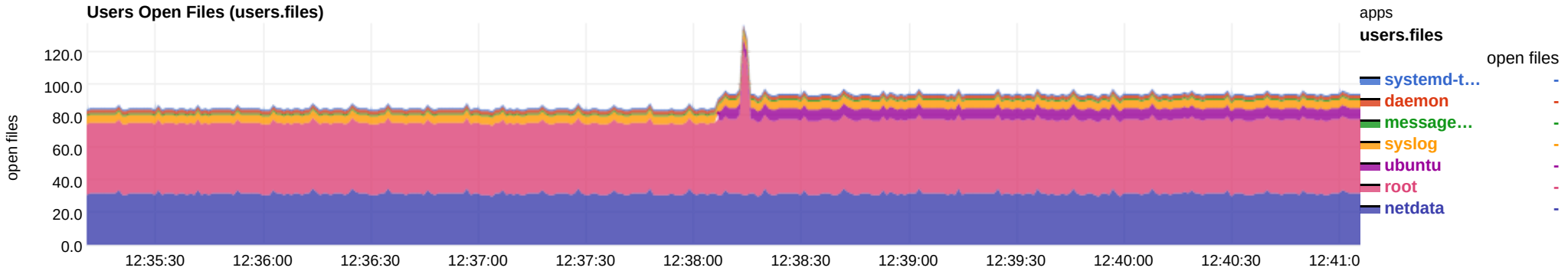

Users CPU System Time (400% = 4 cores) (users.cpu_system)
percentage
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.cpu_system
percentage
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

disk


Users Disk Reads (users.preads)
KiB/s
850.0
800.0
750.0
700.0
650.0
600.0
550.0
500.0
450.0
400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.preads
KiB/s
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

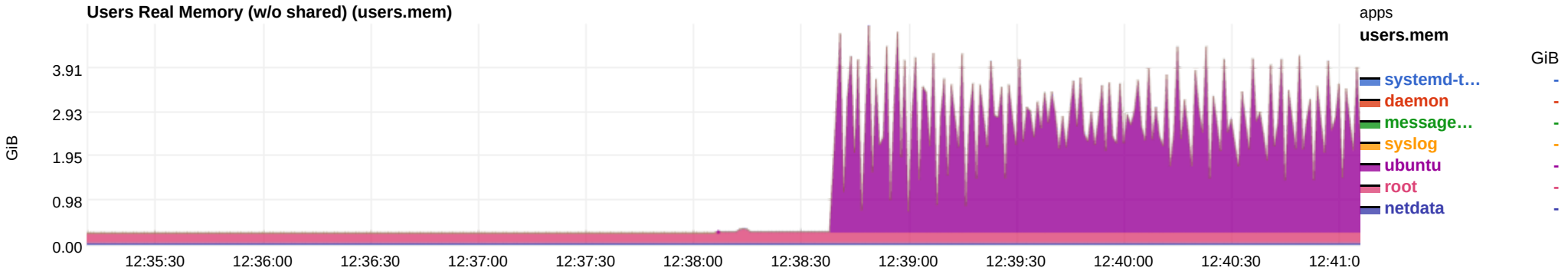

Users Disk Writes (users.pwrites)
MiB/s
33.2
31.3
29.3
27.3
25.4
23.4
21.5
19.5
17.6
15.6
13.7
11.7
9.8
7.8
5.9
3.9
2.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.pwrites
MiB/s
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -


Users Disk Logical Reads (users.lreads)
MiB/s
48.8
39.1
29.3
19.5
9.8
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.lreads
MiB/s
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

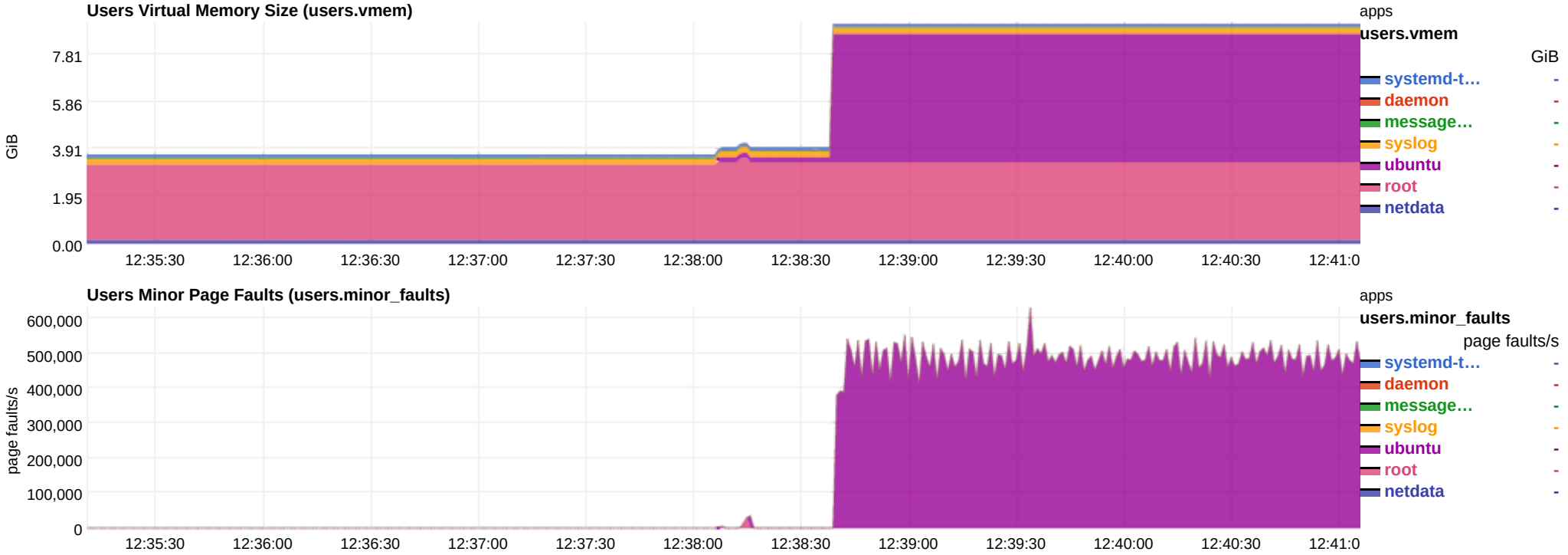


mem

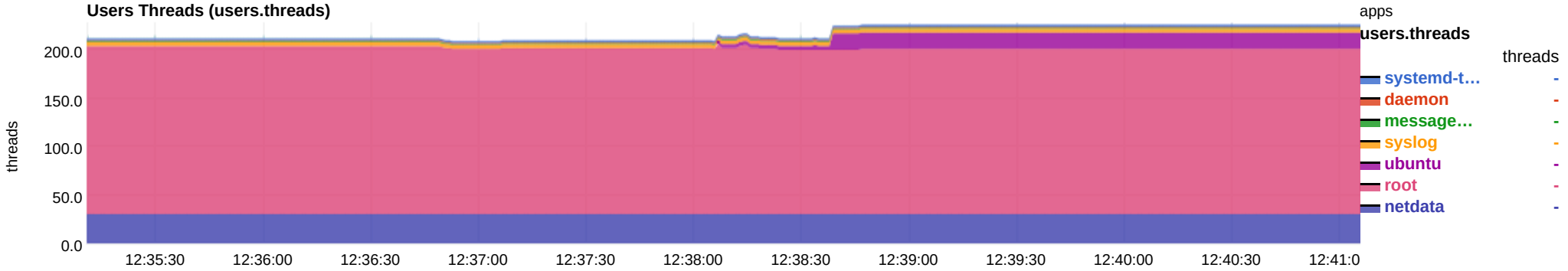
Real memory (RAM) used per user. This does not include shared memory.

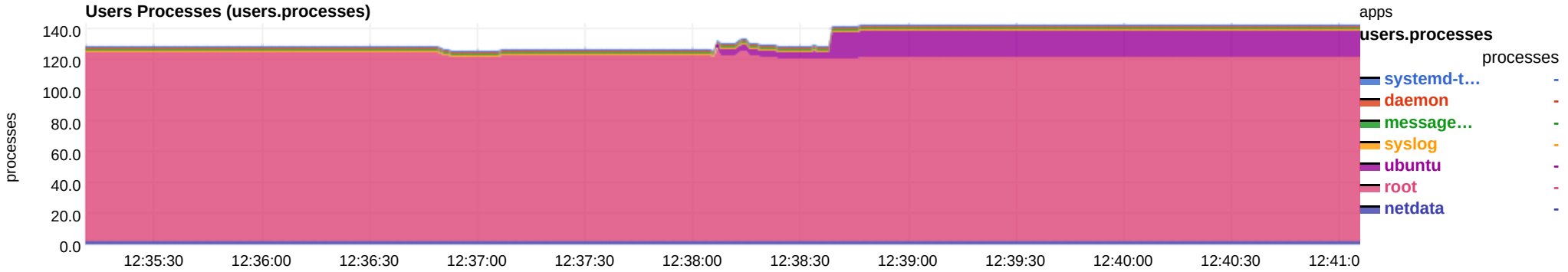


Virtual memory allocated per user. Please check this article (https://github.com/netdata/netdata/tree/master/daemon#virtual-memory) for more information.

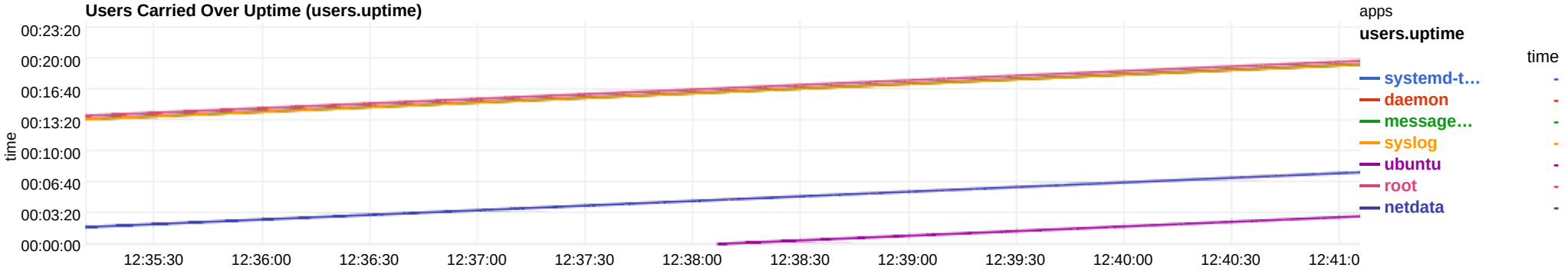


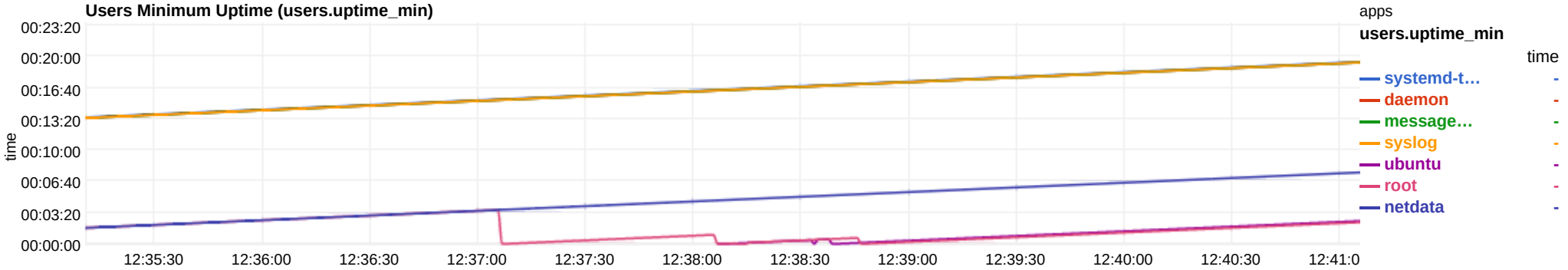
processes

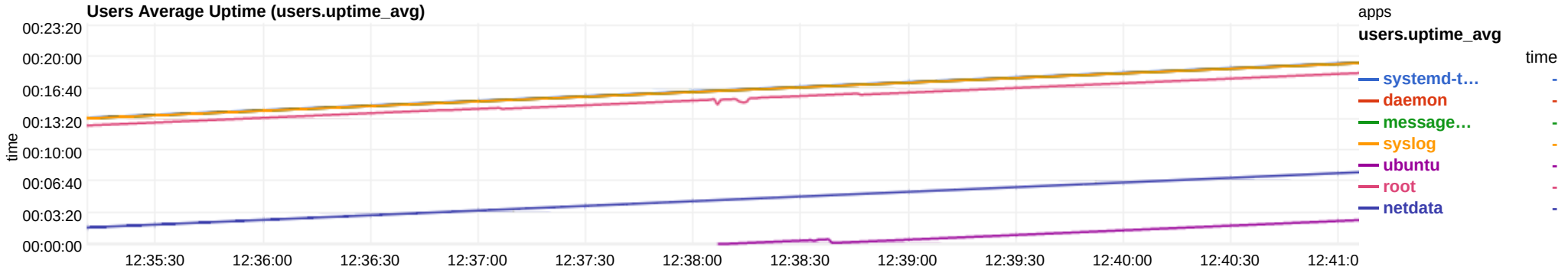


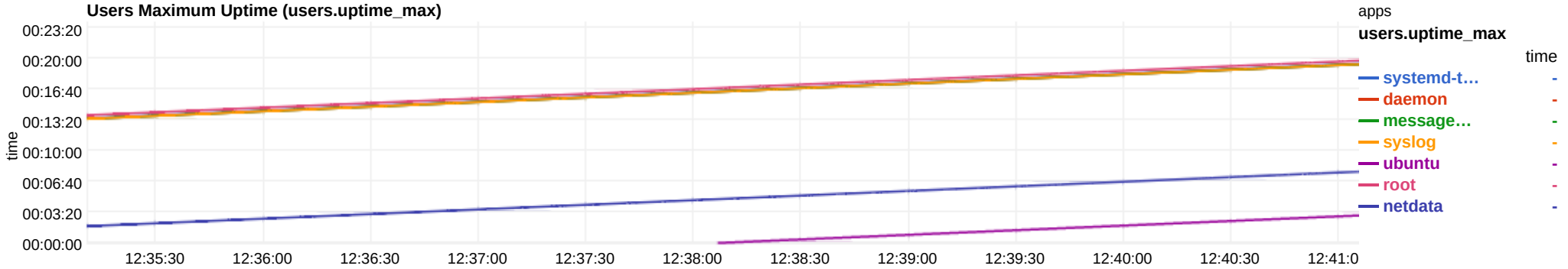


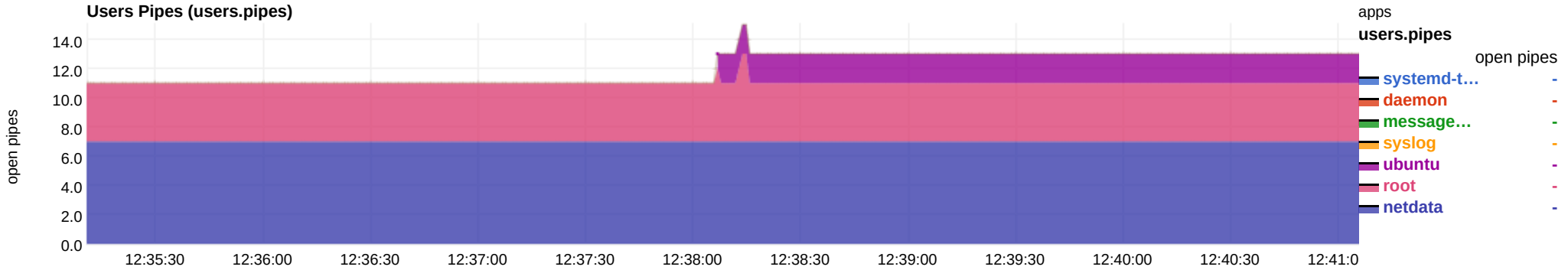
Carried over process group uptime since the Netdata restart. The period of time within which at least one process in the group was running.



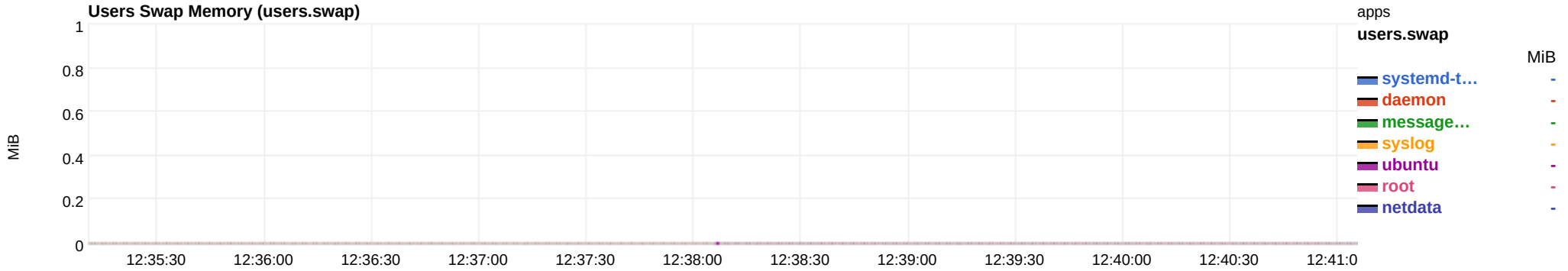


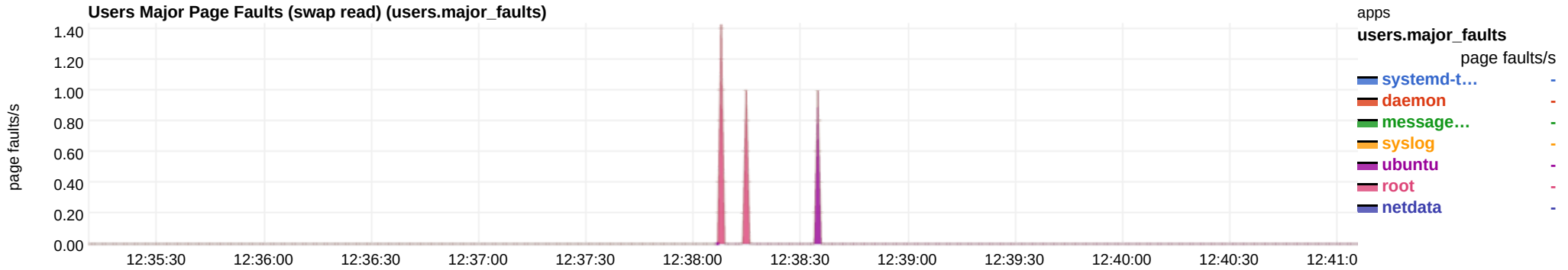

Users Average Uptime (users.uptime_avg)
time
00:23:20
00:20:00
00:16:40
00:13:20
00:10:00
00:06:40
00:03:20
00:00:00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.uptime_avg
time
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -


Users Maximum Uptime (users.uptime_max)
time
00:23:20
00:20:00
00:16:40
00:13:20
00:10:00
00:06:40
00:03:20
00:00:00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.uptime_max
time
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

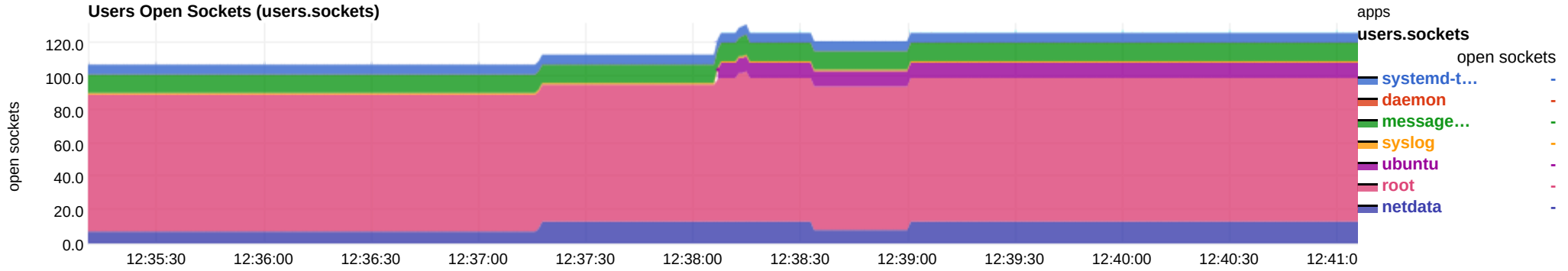

Users Pipes (users.pipes)
open pipes
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
users.pipes
open pipes
systemd-t… -
daemon -
message… -
syslog -
ubuntu -
root -
netdata -

swap





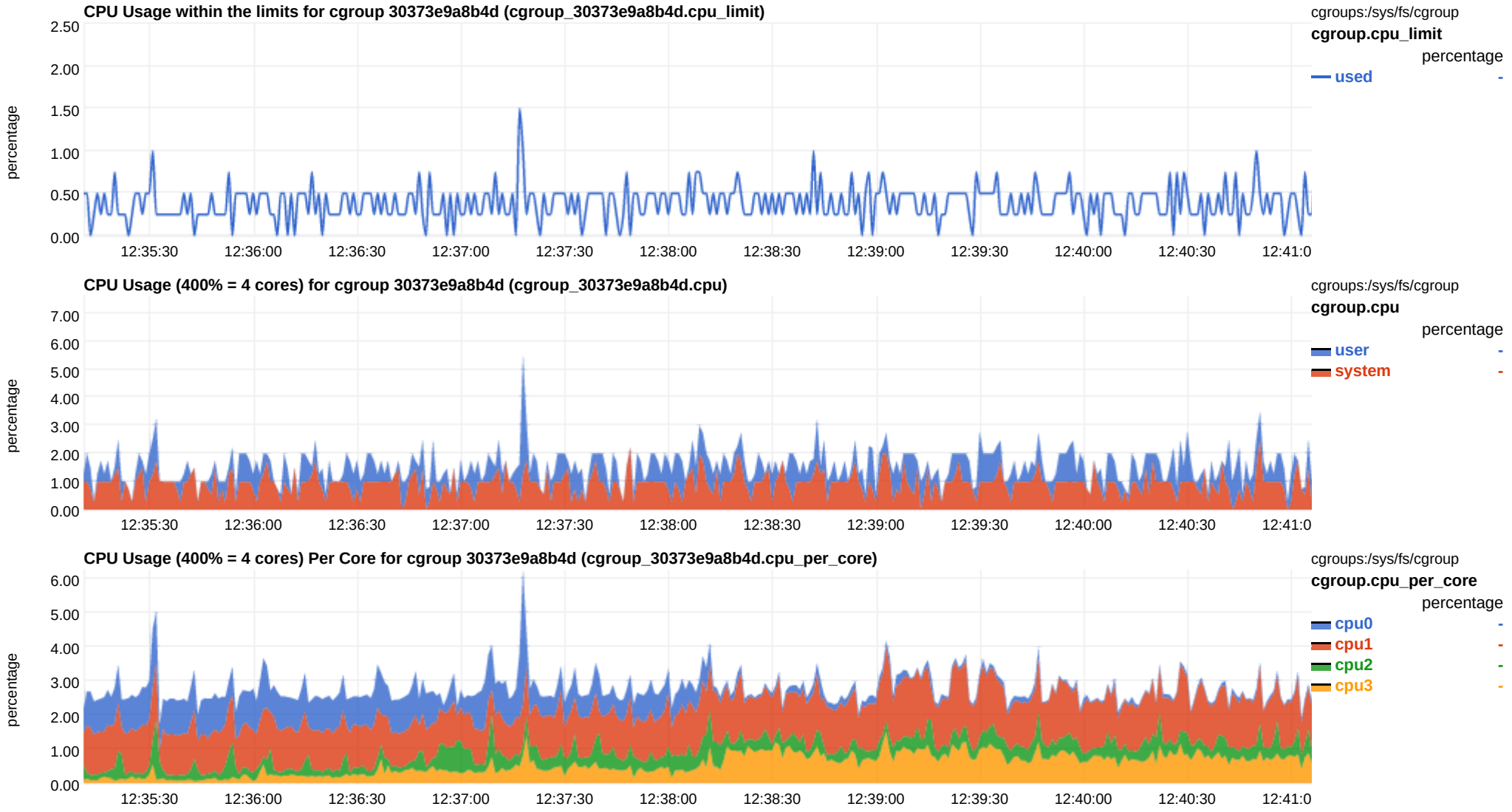
net



30373e9a8b4d

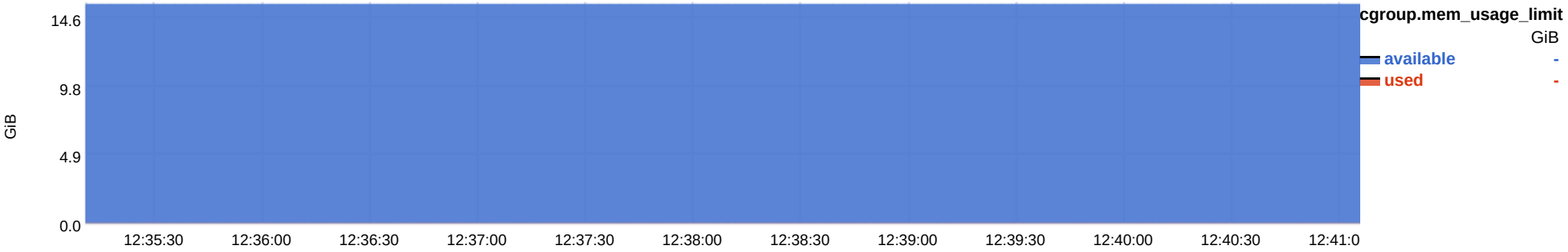
Container resource utilization metrics. Netdata reads this information from **cgroups** (abbreviated from **control groups**), a Linux kernel feature that limits and accounts resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes. **cgroups** together with **namespaces** (that offer isolation between processes) provide what we usually call: **containers**.

cpu

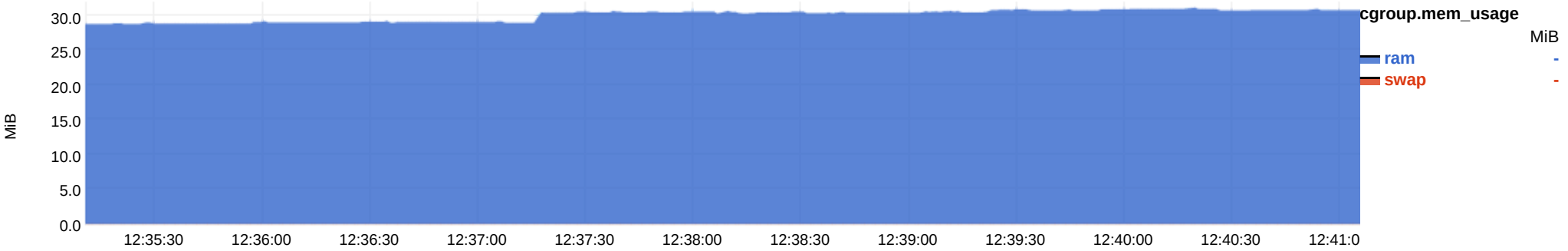


mem

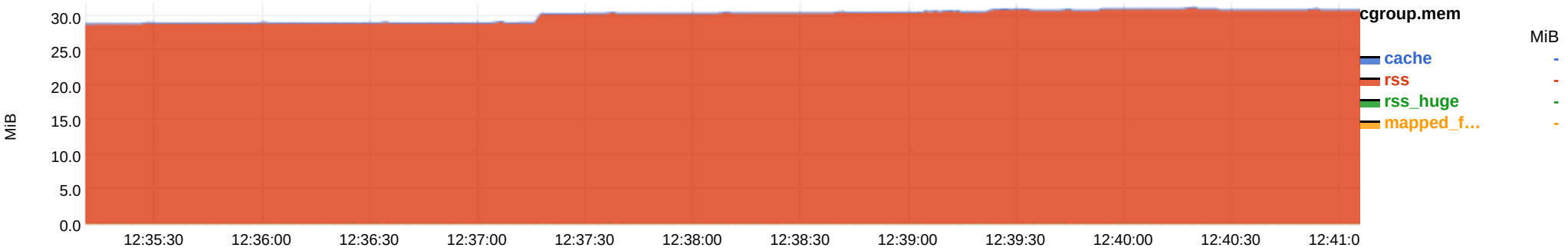
Used RAM without Cache within the limits for cgroup 30373e9a8b4d (cgroup_30373e9a8b4d.mem_usage_limit)



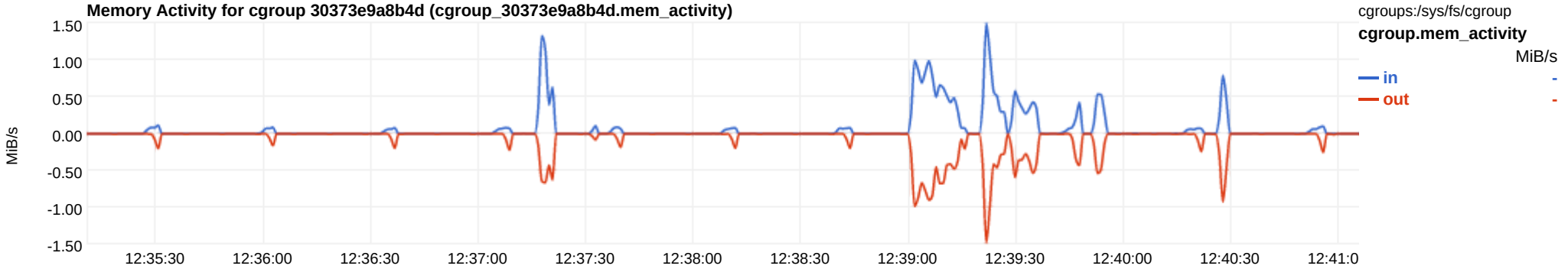
Used Memory without Cache for cgroup 30373e9a8b4d (cgroup_30373e9a8b4d.mem_usage)

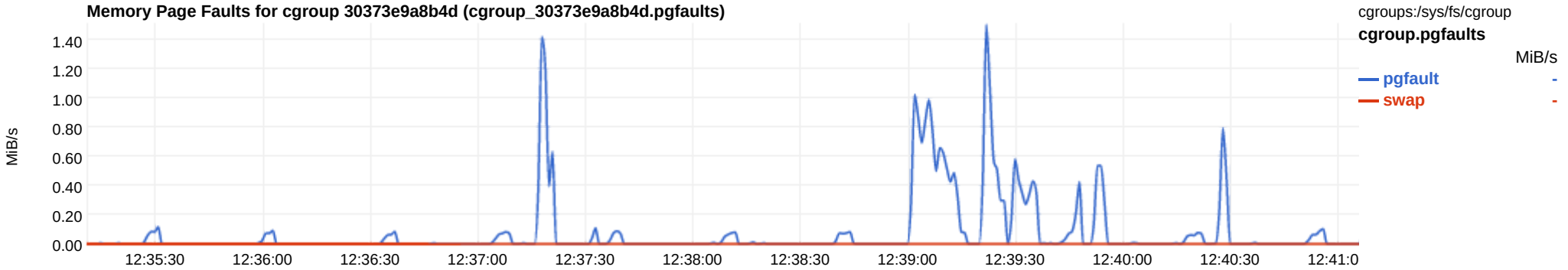


Memory Usage for cgroup 30373e9a8b4d (cgroup_30373e9a8b4d.mem)

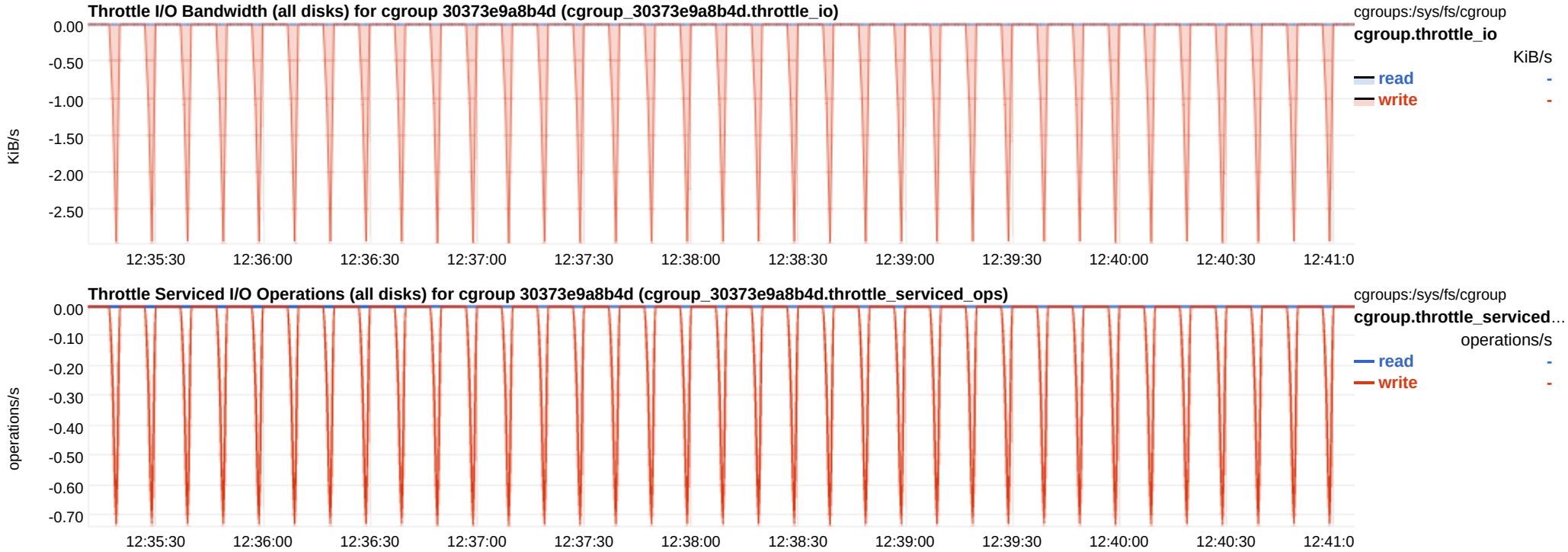



Writeback Memory for cgroup 30373e9a8b4d (cgroup_30373e9a8b4d.writeback)
cgroups:/sys/fs/cgroup
cgroup.writeback
KiB
dirty -
writeback -
KiB
7.17
6.14
5.12
4.10
3.07
2.05
1.02
0.00
12:35:30 12:36:00 12:36:30 12:37:00 12:37:30 12:38:00 12:38:30 12:39:00 12:39:30 12:40:00 12:40:30 12:41:0


Memory Activity for cgroup 30373e9a8b4d (cgroup_30373e9a8b4d.mem_activity)
cgroups:/sys/fs/cgroup
cgroup.mem_activity
MiB/s
in -
out -
MiB/s
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
12:35:30 12:36:00 12:36:30 12:37:00 12:37:30 12:38:00 12:38:30 12:39:00 12:39:30 12:40:00 12:40:30 12:41:0

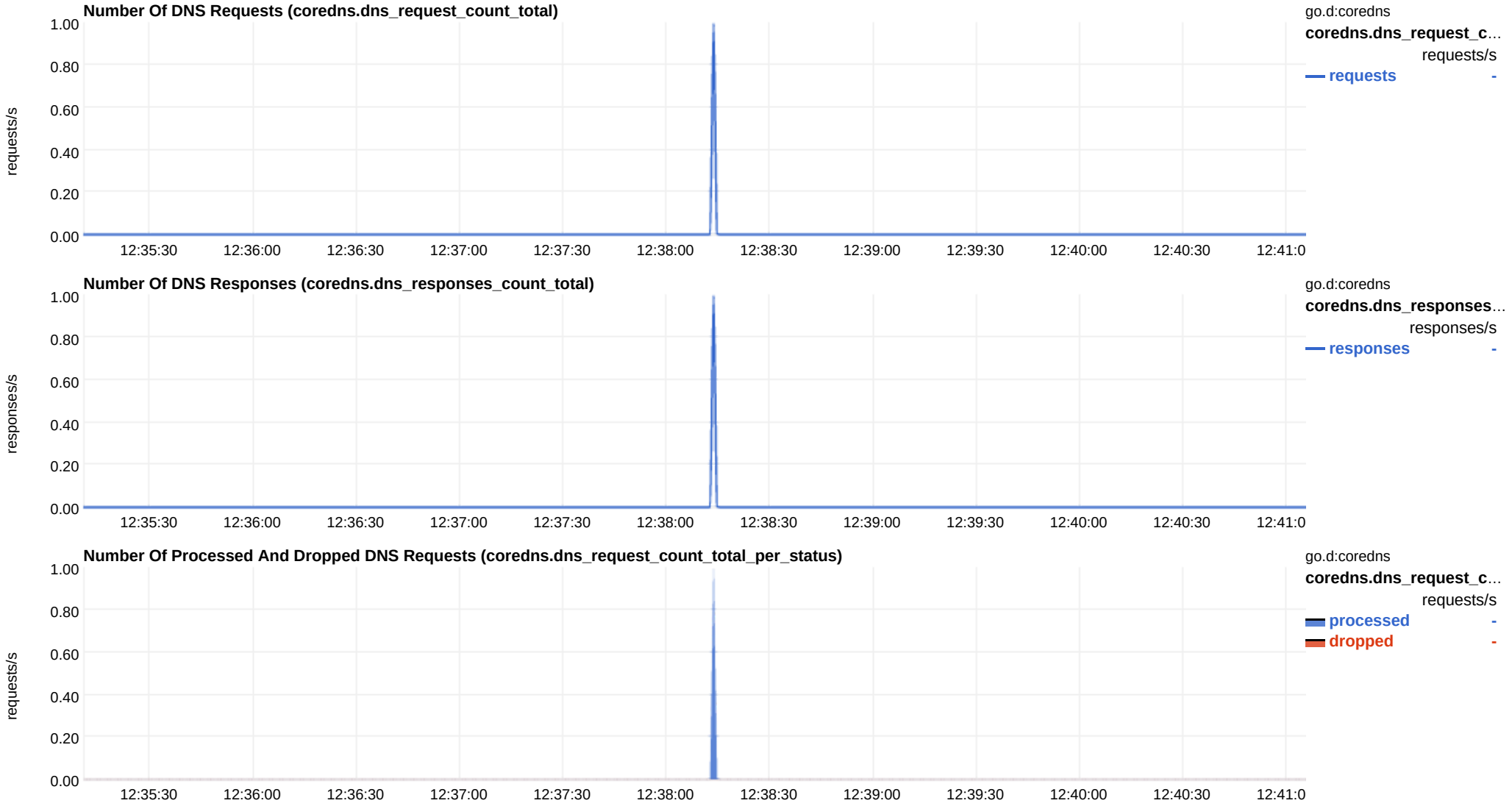

Memory Page Faults for cgroup 30373e9a8b4d (cgroup_30373e9a8b4d.pgfaults)
cgroups:/sys/fs/cgroup
cgroup.pgfaults
MiB/s
pgfault -
swap -
MiB/s
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
12:35:30 12:36:00 12:36:30 12:37:00 12:37:30 12:38:00 12:38:30 12:39:00 12:39:30 12:40:00 12:40:30 12:41:0

disk

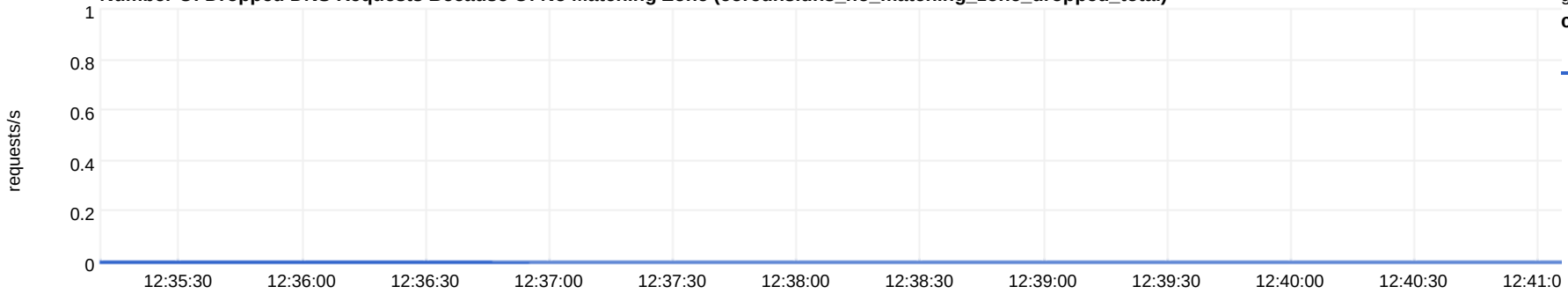


coredns

summary

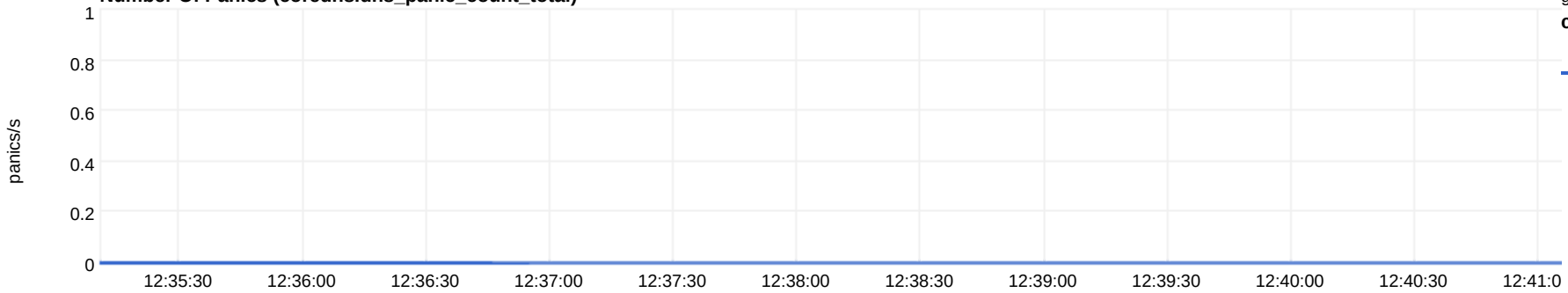

Number Of DNS Requests (coredns.dns_request_count_total)
requests/s
1.00
0.80
0.60
0.40
0.20
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
go.d:coredns
coredns.dns_request_c...
requests/s
requests -
Number Of DNS Responses (coredns.dns_responses_count_total)
responses/s
1.00
0.80
0.60
0.40
0.20
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
go.d:coredns
coredns.dns_responses...
responses/s
responses -
Number Of Processed And Dropped DNS Requests (coredns.dns_request_count_total_per_status)
requests/s
1.00
0.80
0.60
0.40
0.20
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
go.d:coredns
coredns.dns_request_c...
requests/s
processed -
dropped -

Number Of Dropped DNS Requests Because Of No Matching Zone (coredns.dns_no_matching_zone_dropped_total)



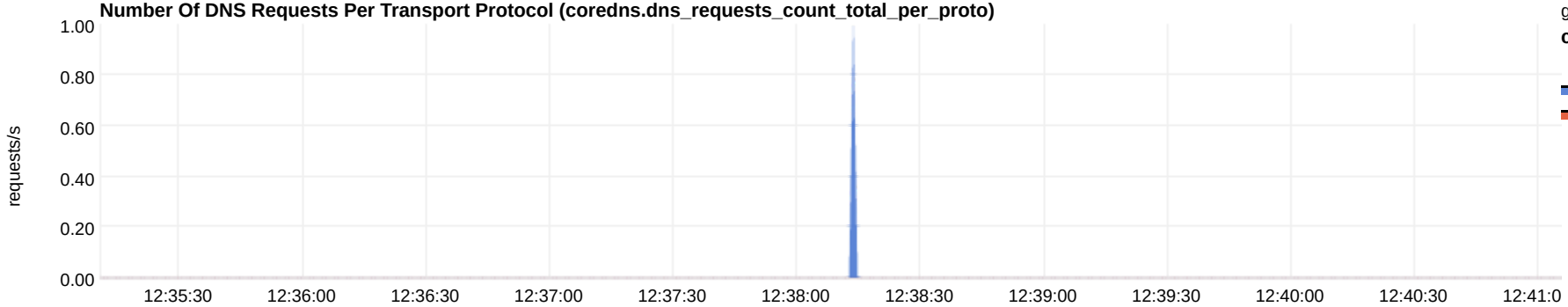
go.d:coredns
coredns.dns_no_matchi...
requests/s
dropped -

Number Of Panics (coredns.dns_panic_count_total)

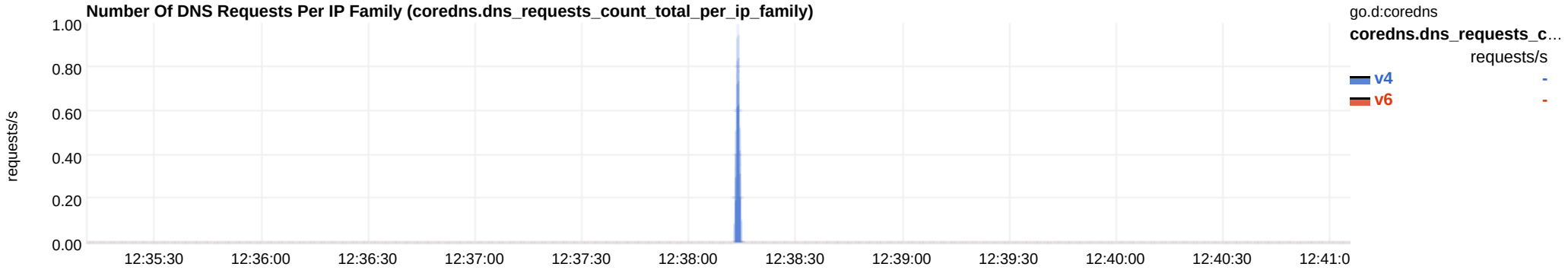


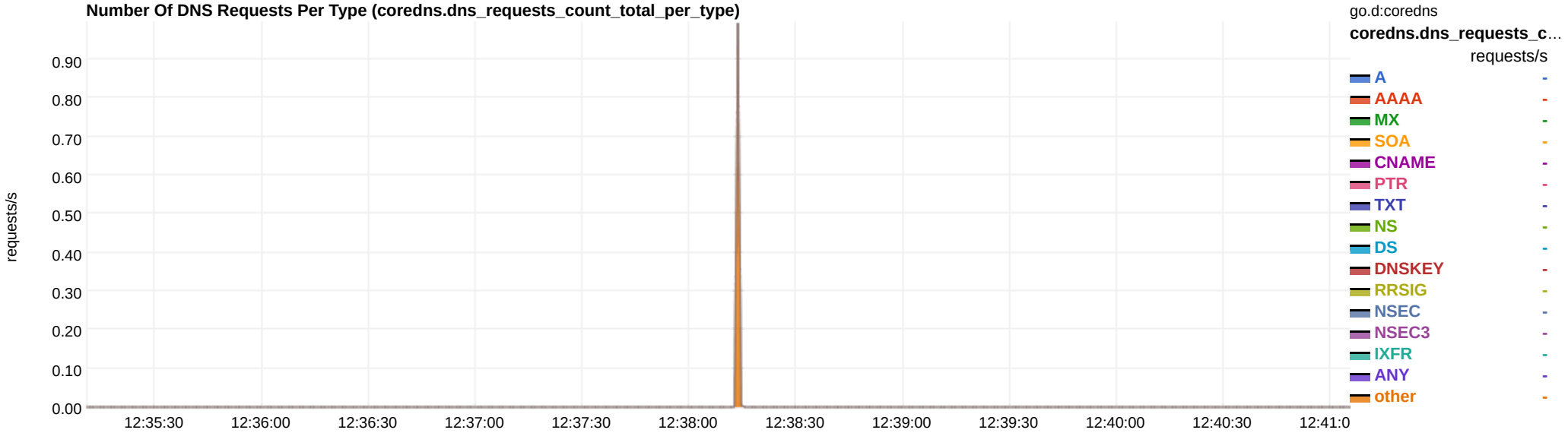
go.d:coredns
coredns.dns_panic_cou...
panics/s
panics -

Number Of DNS Requests Per Transport Protocol (coredns.dns_requests_count_total_per_proto)



go.d:coredns
coredns.dns_requests_c...
requests/s
udp -
tcp -


Number Of DNS Requests Per IP Family (coredns.dns_requests_count_total_per_ip_family)
requests/s
1.00
0.80
0.60
0.40
0.20
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
go.d:coredns
coredns.dns_requests_c...
requests/s
v4 -
v6 -


Number Of DNS Requests Per Type (coredns.dns_requests_count_total_per_type)
requests/s
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
go.d:coredns
coredns.dns_requests_c...
requests/s
A -
AAAA -
MX -
SOA -
CNAME -
PTR -
TXT -
NS -
DS -
DNSKEY -
RRSIG -
NSEC -
NSEC3 -
IXFR -
ANY -
other -

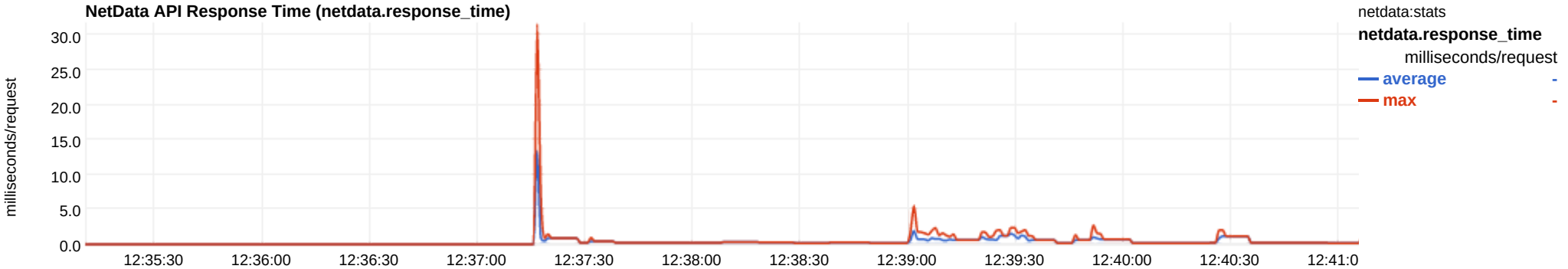
Netdata Monitoring

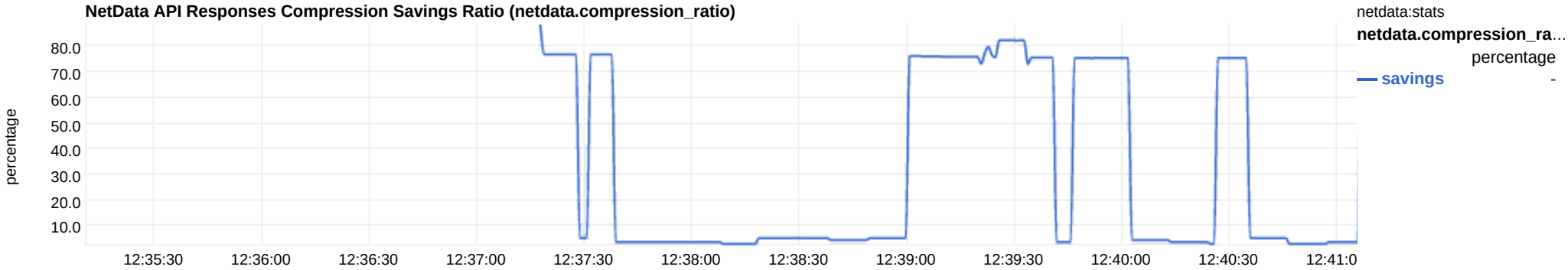
Performance metrics for the operation of netdata itself and its plugins.

netdata


NetData Network Traffic (netdata.net)
megabits/s
0.00
-0.50
-1.00
-1.50
-2.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.net
megabits/s
in -
out -
NetData CPU usage (netdata.server_cpu)
milliseconds/s
40.0
30.0
20.0
10.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.server_cpu
milliseconds/s
user -
system -
NetData Web Clients (netdata.clients)
connected clients
6.00
5.00
4.00
3.00
2.00
1.00
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.clients
connected clients
clients -

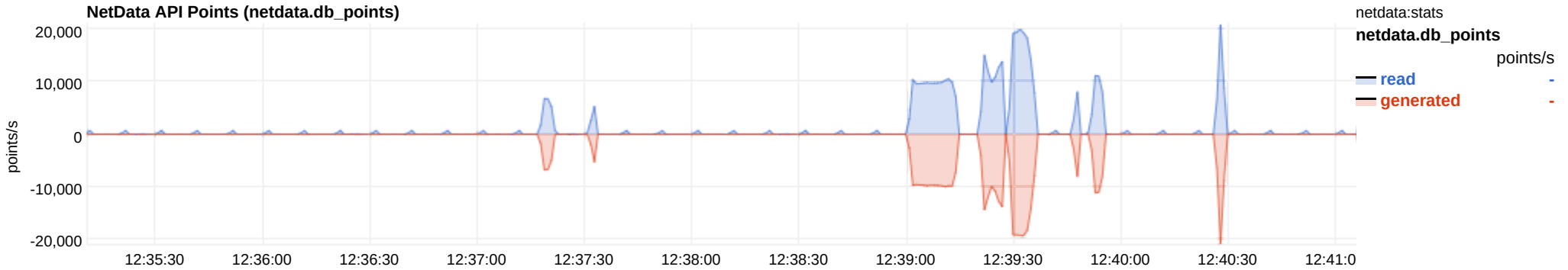
The netdata API response time measures the time netdata needed to serve requests. This time includes everything, from the reception of the first byte of a request, to the dispatch of the last byte of its reply, therefore it includes all network latencies involved (i.e. a client over a slow network will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



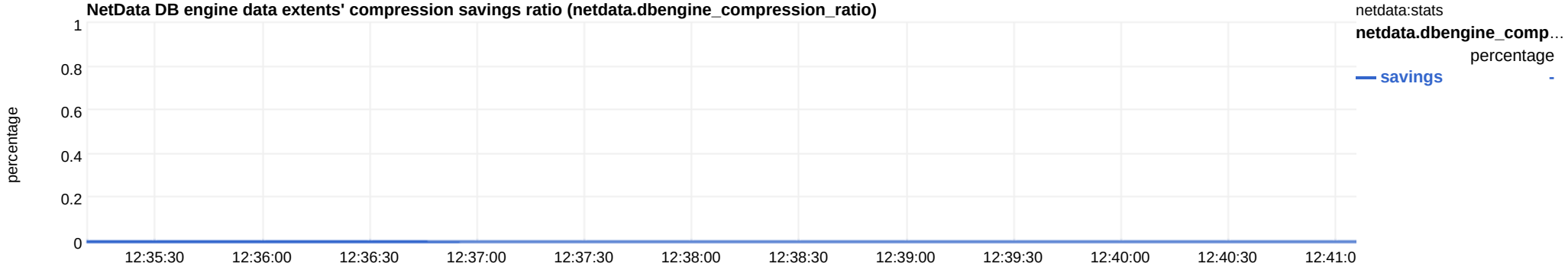


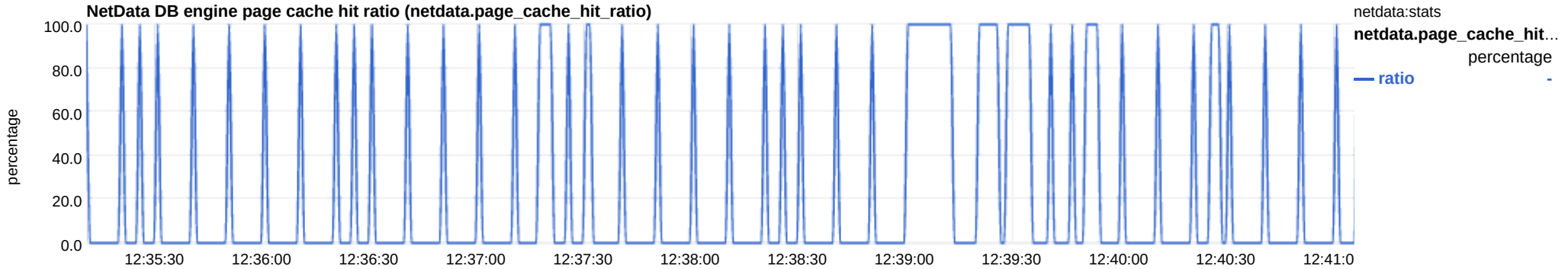
queries

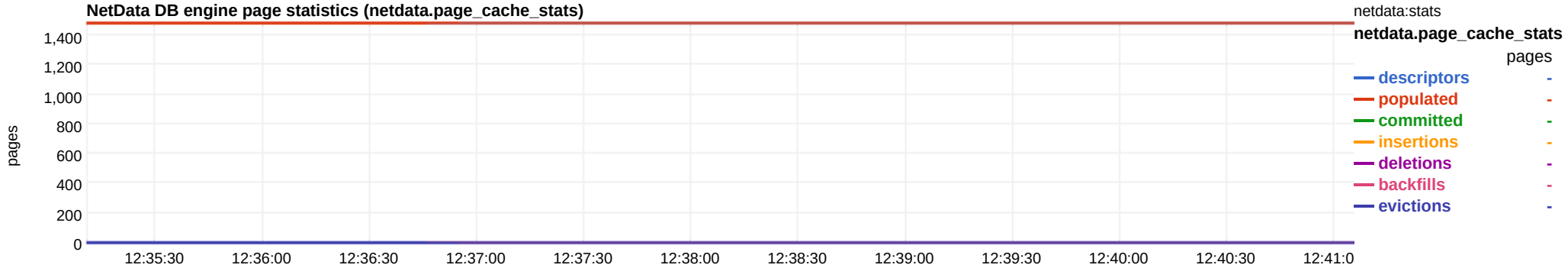

NetData API Queries (netdata.queries)
queries/s
30.0
25.0
20.0
15.0
10.0
5.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.queries
queries/s
queries -

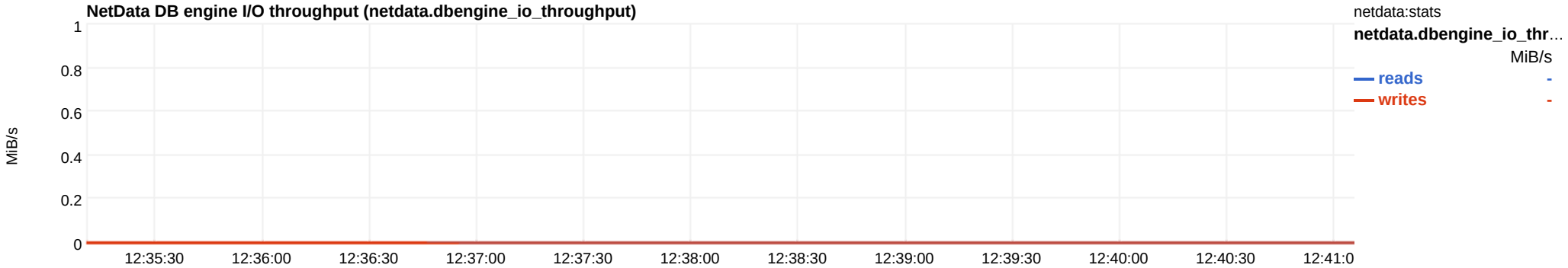

NetData API Points (netdata.db_points)
points/s
20,000
10,000
0
-10,000
-20,000
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.db_points
points/s
read -
generated -

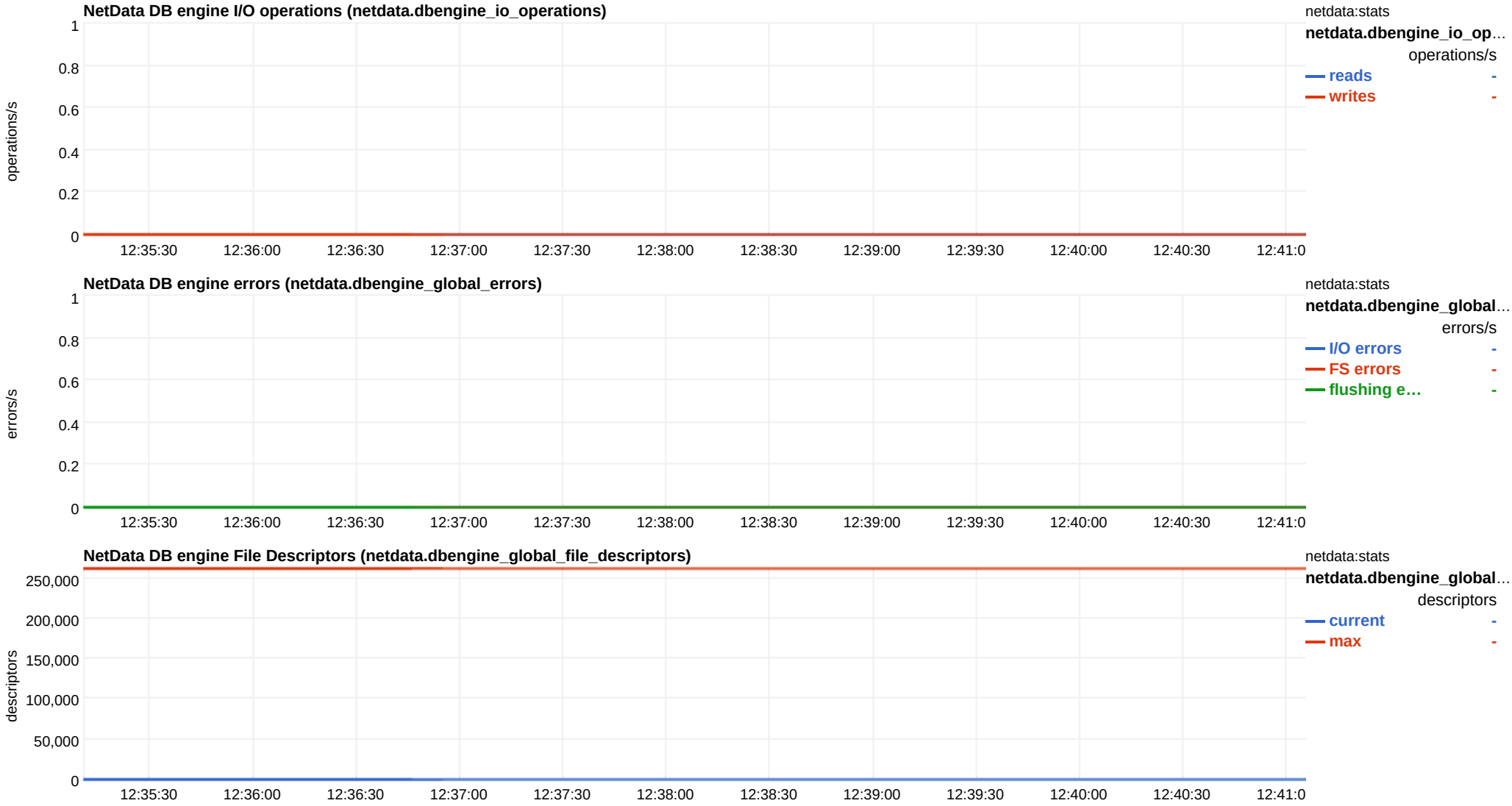
dbengine


NetData DB engine data extents' compression savings ratio (netdata.dbengine_compression_ratio)
percentage
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.dbengine_comp...
percentage
savings -

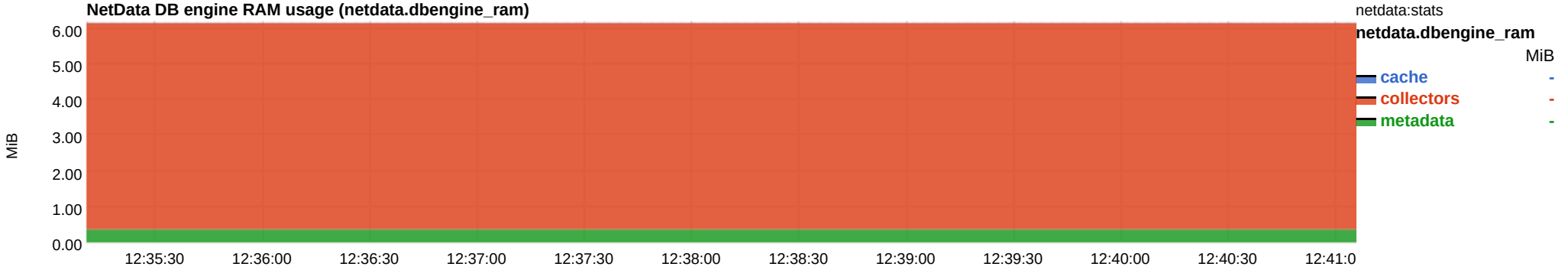

NetData DB engine page cache hit ratio (netdata.page_cache_hit_ratio)
percentage
100.0
80.0
60.0
40.0
20.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.page_cache_hit...
percentage
ratio -


NetData DB engine page statistics (netdata.page_cache_stats)
pages
1,400
1,200
1,000
800
600
400
200
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.page_cache_stats
pages
descriptors -
populated -
committed -
insertions -
deletions -
backfills -
evictions -


NetData DB engine I/O throughput (netdata.dbengine_io_throughput)
MiB/s
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.dbengine_io_thr...
MiB/s
reads -
writes -

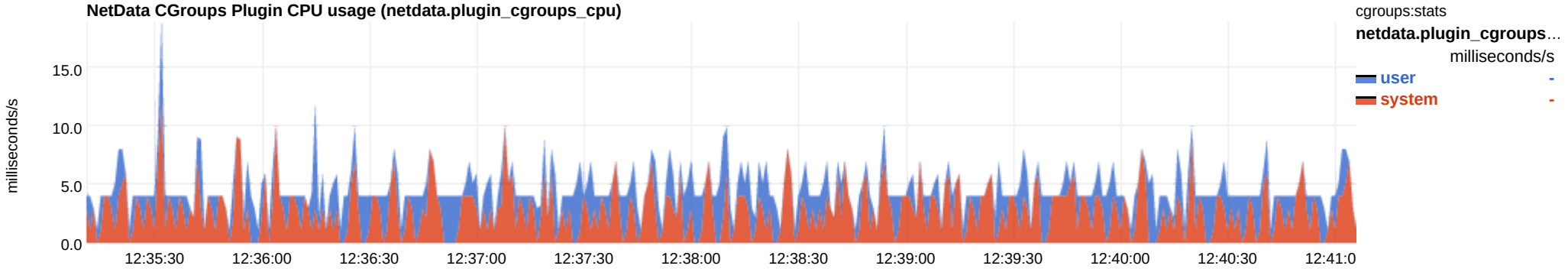

NetData DB engine I/O operations (netdata.dbengine_io_operations)
operations/s
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.dbengine_io_op...
operations/s
reads -
writes -
NetData DB engine errors (netdata.dbengine_global_errors)
errors/s
1
0.8
0.6
0.4
0.2
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.dbengine_global...
errors/s
I/O errors -
FS errors -
flushing e... -
NetData DB engine File Descriptors (netdata.dbengine_global_file_descriptors)
descriptors
250,000
200,000
150,000
100,000
50,000
0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
netdata:stats
netdata.dbengine_global...
descriptors
current -
max -

NetData DB engine RAM usage (netdata.dbengine_ram)



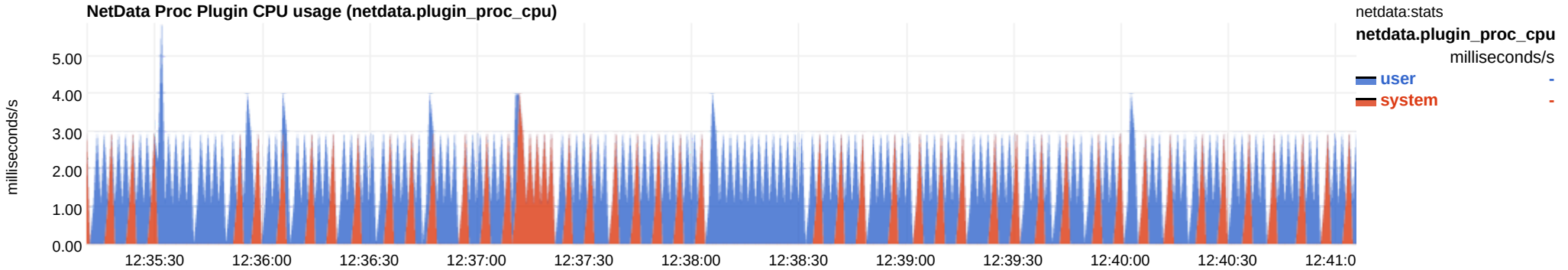
cgroups

NetData CGroups Plugin CPU usage (netdata.plugin_cgroups_cpu)

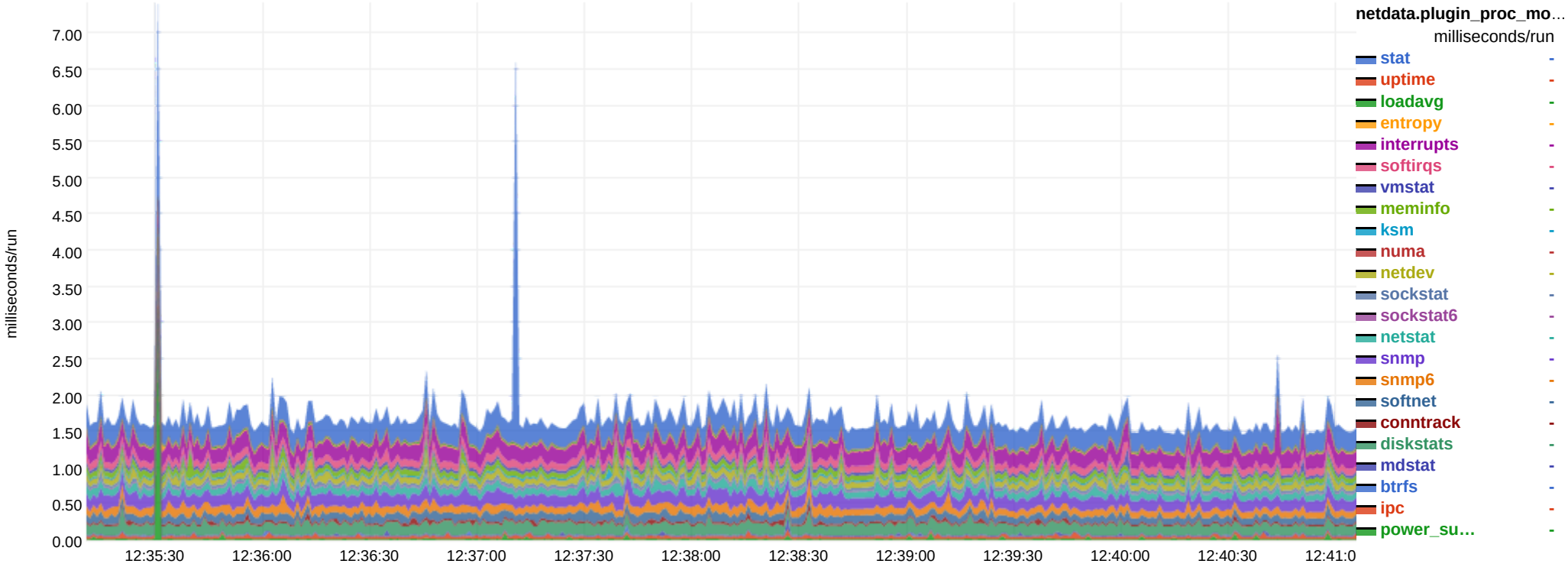


proc

NetData Proc Plugin CPU usage (netdata.plugin_proc_cpu)

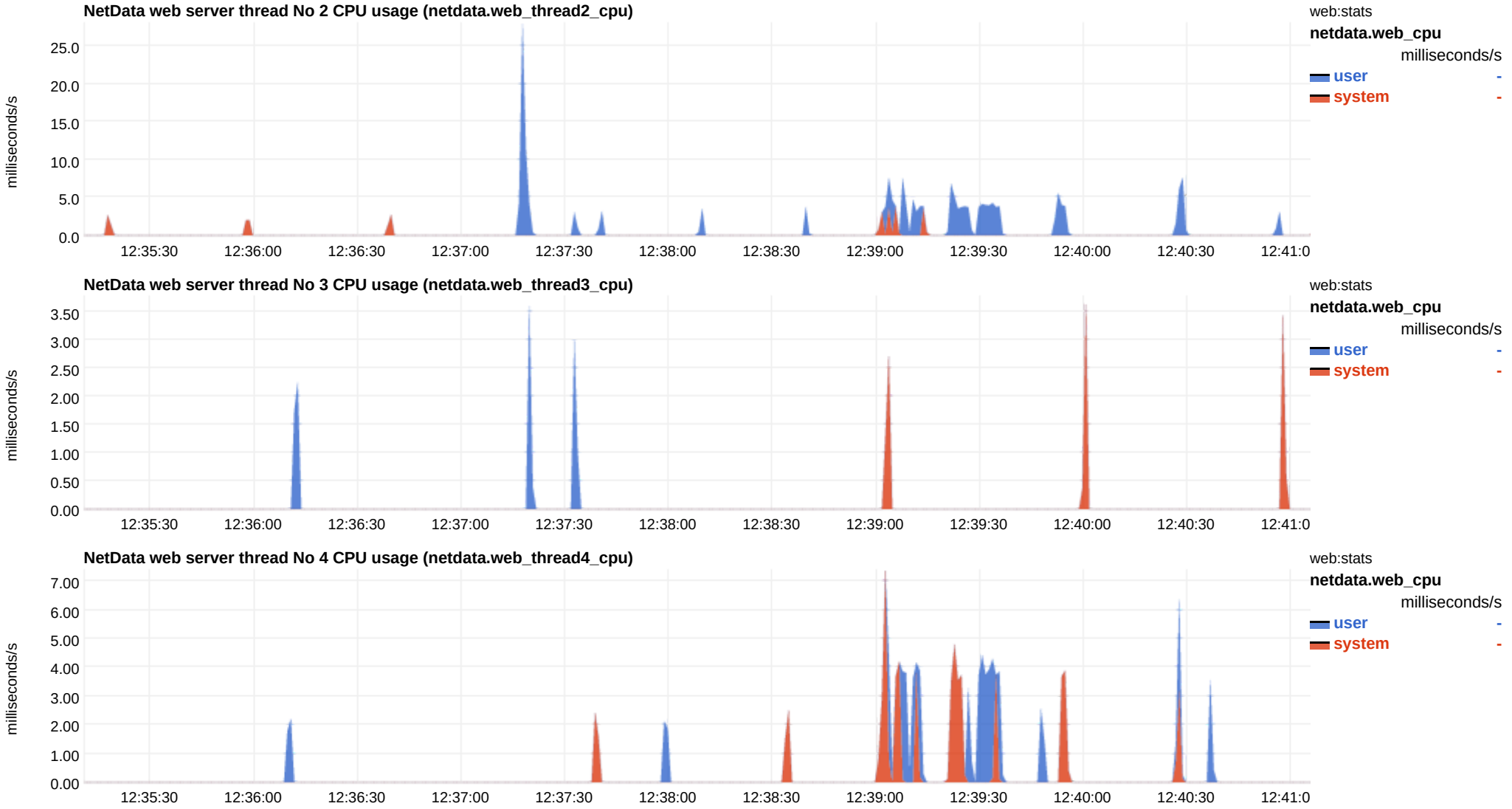


NetData Proc Plugin Modules Durations (netdata.plugin_proc_modules)



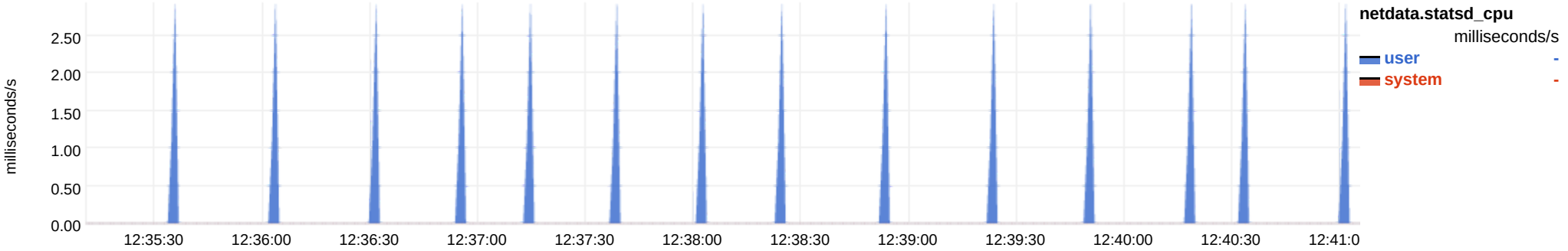
web

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


NetData web server thread No 2 CPU usage (netdata.web_thread2_cpu)
web:stats
netdata.web_cpu
milliseconds/s
user -
system -
milliseconds/s
25.0
20.0
15.0
10.0
5.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
NetData web server thread No 3 CPU usage (netdata.web_thread3_cpu)
web:stats
netdata.web_cpu
milliseconds/s
user -
system -
milliseconds/s
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
NetData web server thread No 4 CPU usage (netdata.web_thread4_cpu)
web:stats
netdata.web_cpu
milliseconds/s
user -
system -
milliseconds/s
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0

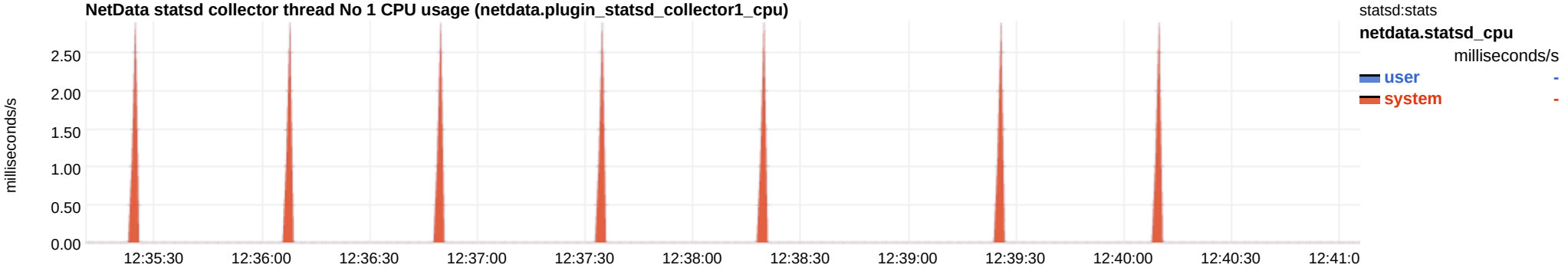
statsd

NetData statsd charting thread CPU usage (netdata.plugin_statsd_charting_cpu)



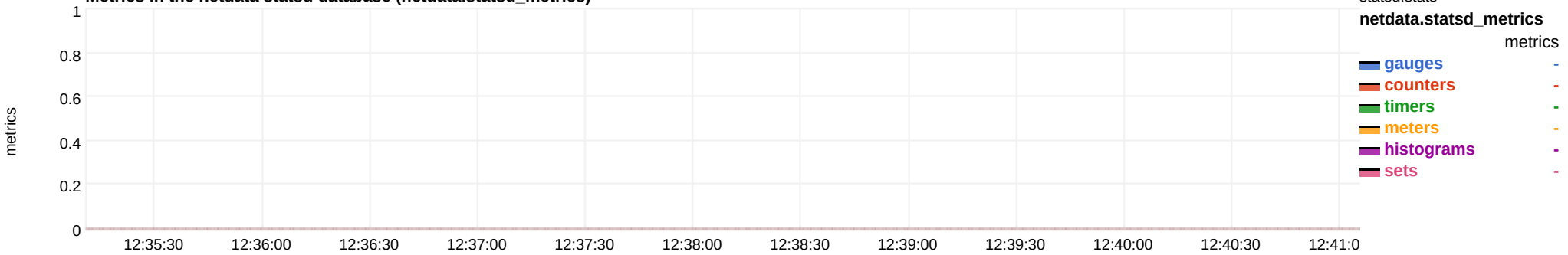
statsd:stats
netdata.statsd_cpu
milliseconds/s
user -
system -

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



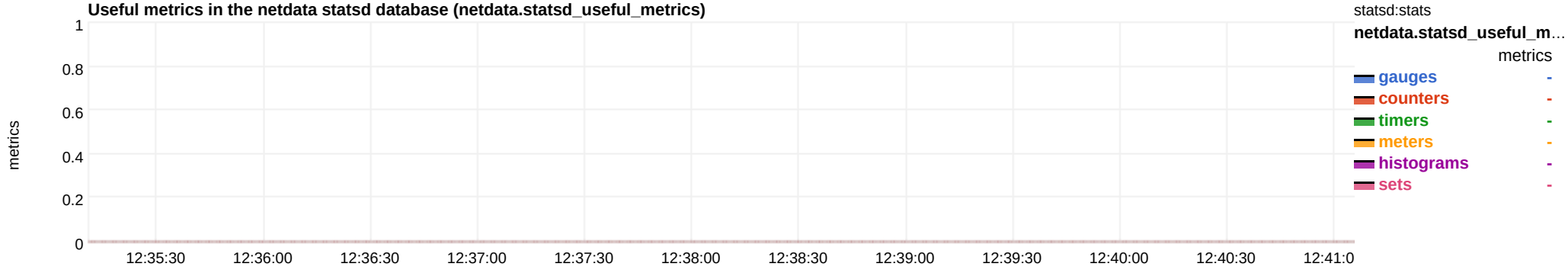
statsd:stats
netdata.statsd_cpu
milliseconds/s
user -
system -

Metrics in the netdata statsd database (netdata.statsd_metrics)



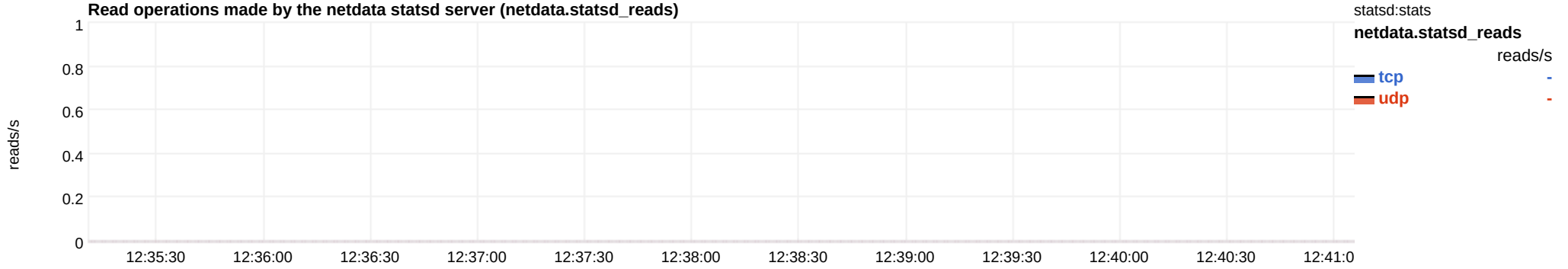
statsd:stats
netdata.statsd_metrics
metrics
gauges -
counters -
timers -
meters -
histograms -
sets -

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

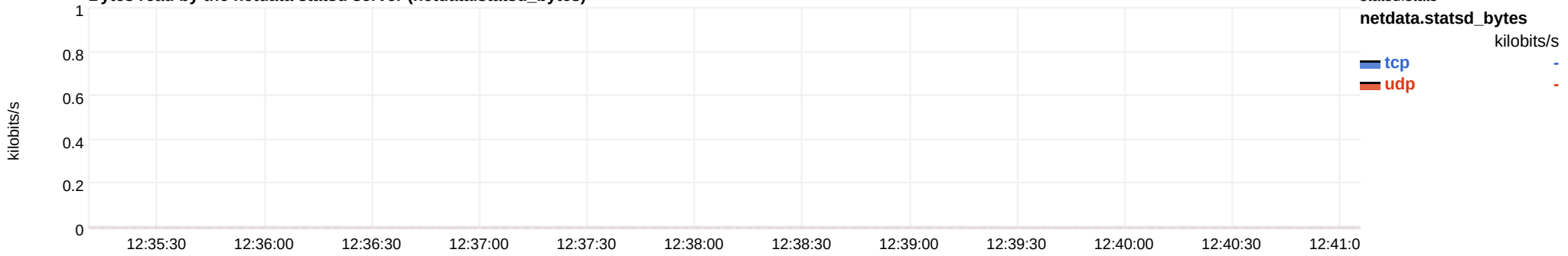


Events processed by the netdata statsd server (netdata.statsd_events)

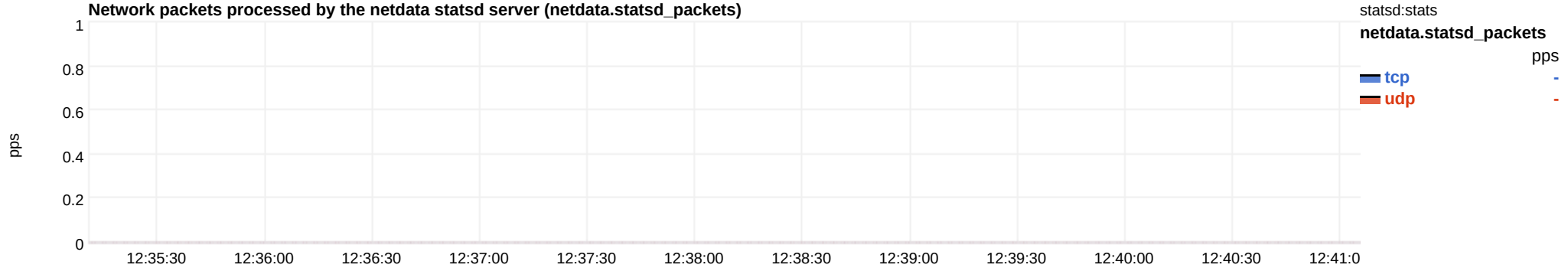
Read operations made by the netdata statsd server (netdata.statsd_reads)



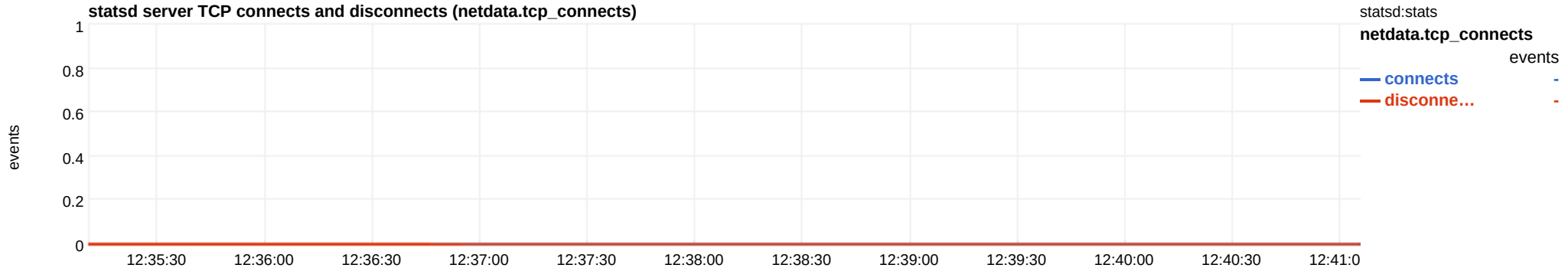
Bytes read by the netdata statsd server (netdata.statsd_bytes)



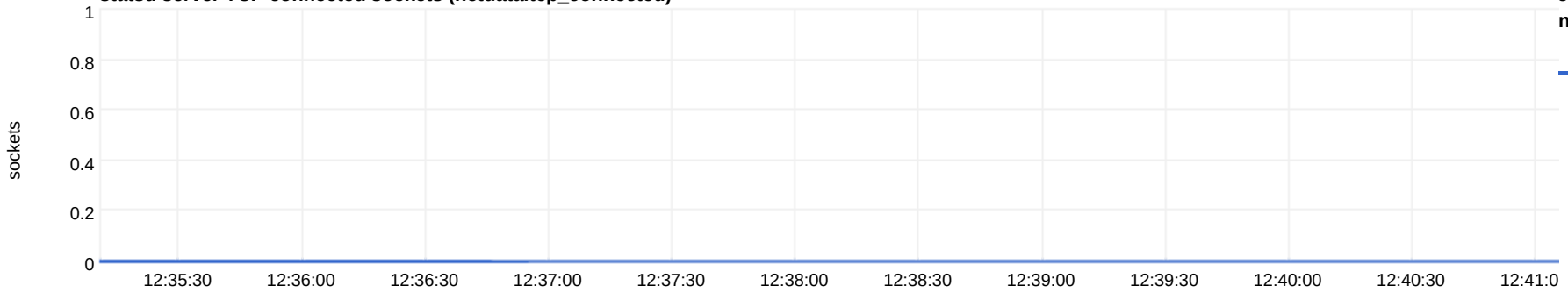
Network packets processed by the netdata statsd server (netdata.statsd_packets)



statsd server TCP connects and disconnects (netdata.tcp_connects)

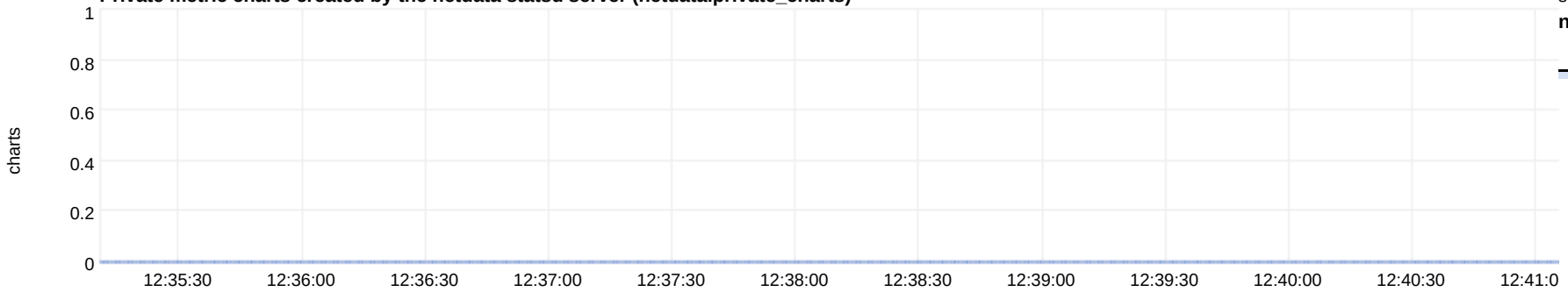


statsd server TCP connected sockets (netdata.tcp_connected)



statsd:stats
netdata.tcp_connected
sockets
connected -

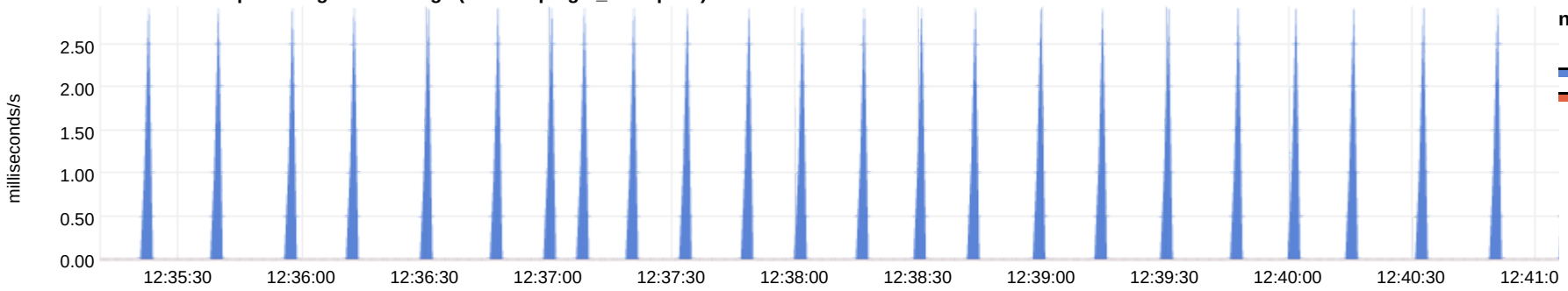
Private metric charts created by the netdata statsd server (netdata.private_charts)



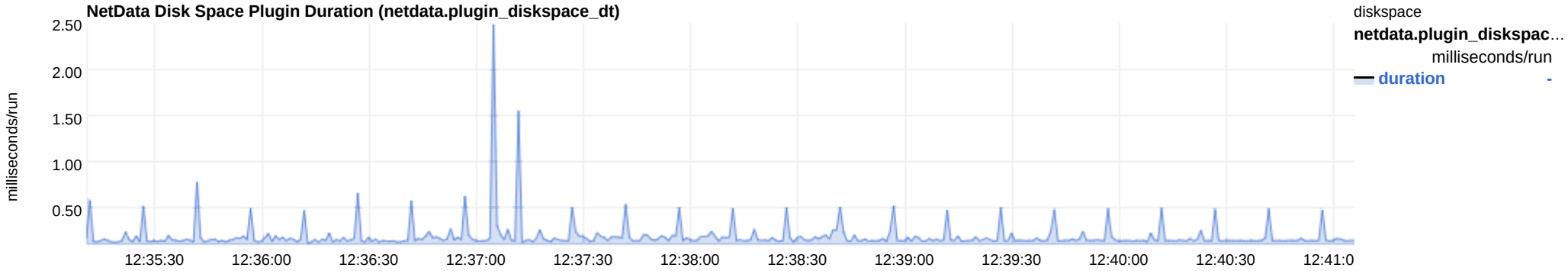
statsd:stats
netdata.private_charts
charts
charts -

diskspace

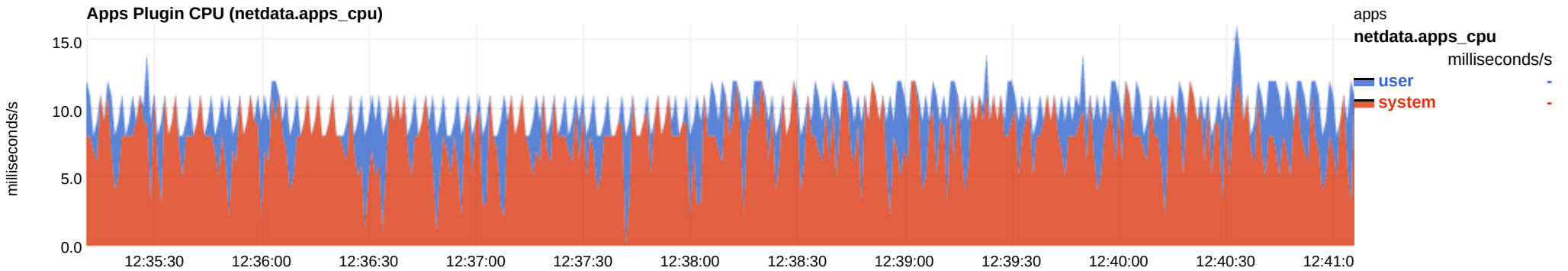
NetData Disk Space Plugin CPU usage (netdata.plugin_diskspace)

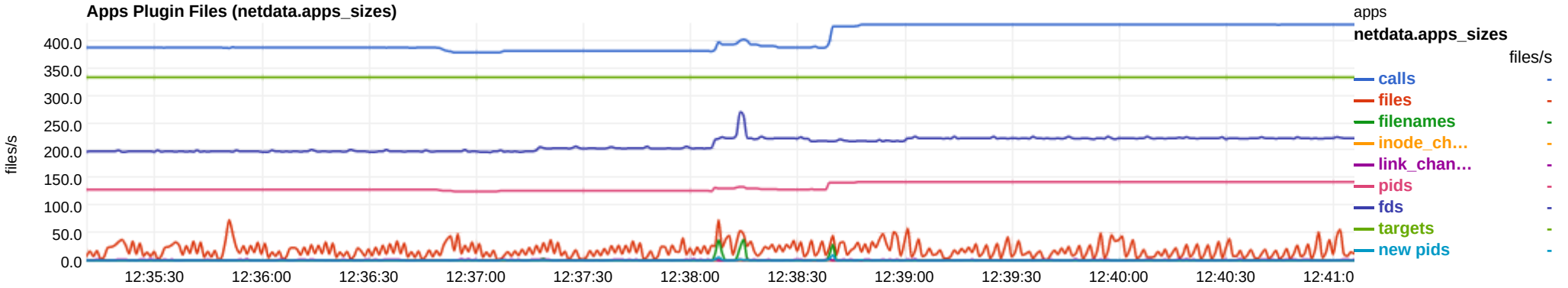


diskspace
netdata.plugin_diskspace
milliseconds/s
user -
system -

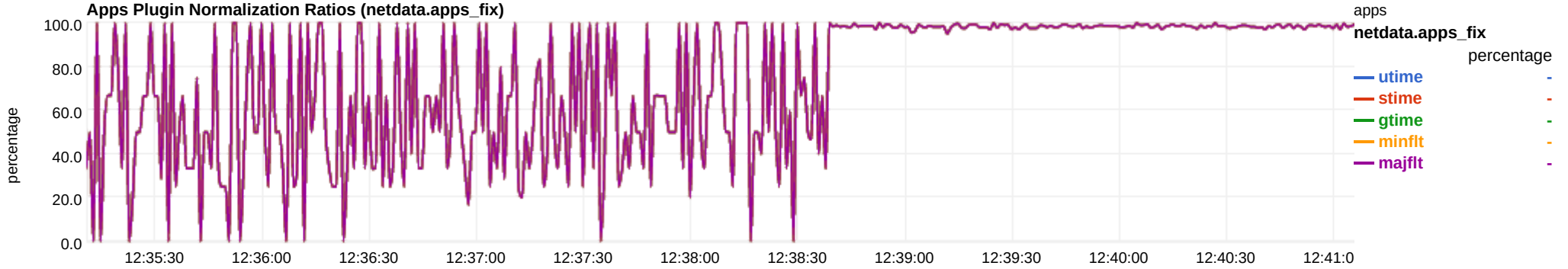

NetData Disk Space Plugin Duration (netdata.plugin_diskspace_dt)
milliseconds/run
2.50
2.00
1.50
1.00
0.50
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
diskspace
netdata.plugin_diskspac...
milliseconds/run
duration

apps.plugin


Apps Plugin CPU (netdata.apps_cpu)
milliseconds/s
15.0
10.0
5.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
netdata.apps_cpu
milliseconds/s
user
system


Apps Plugin Files (netdata.apps_sizes)
files/s
400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
12:35:30
12:36:00
12:36:30
12:37:00
12:37:30
12:38:00
12:38:30
12:39:00
12:39:30
12:40:00
12:40:30
12:41:0
apps
netdata.apps_sizes
files/s
calls
files
filenames
inode_ch...
link_chan...
pids
fds
targets
new pids

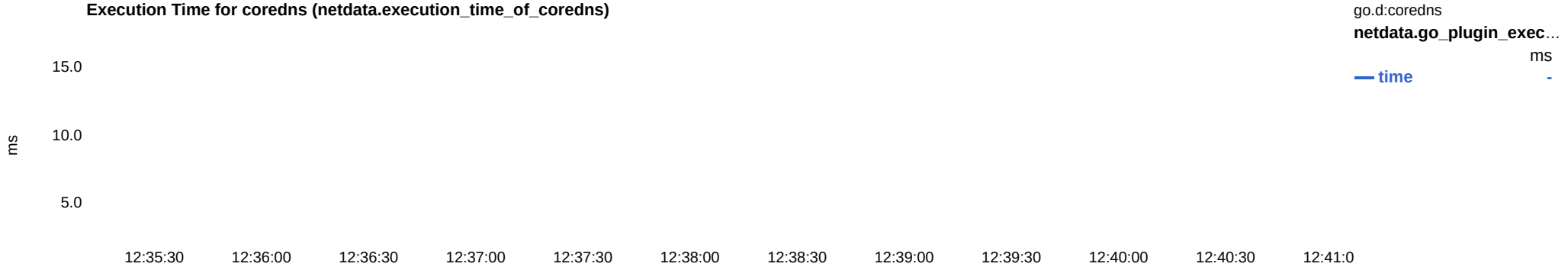
Apps Plugin Normalization Ratios (netdata.apps_fix)



Apps Plugin Exited Children Normalization Ratios (netdata.apps_children_fix)

go.d

Execution Time for coredns (netdata.execution_time_of_coredns)



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