

System Overview

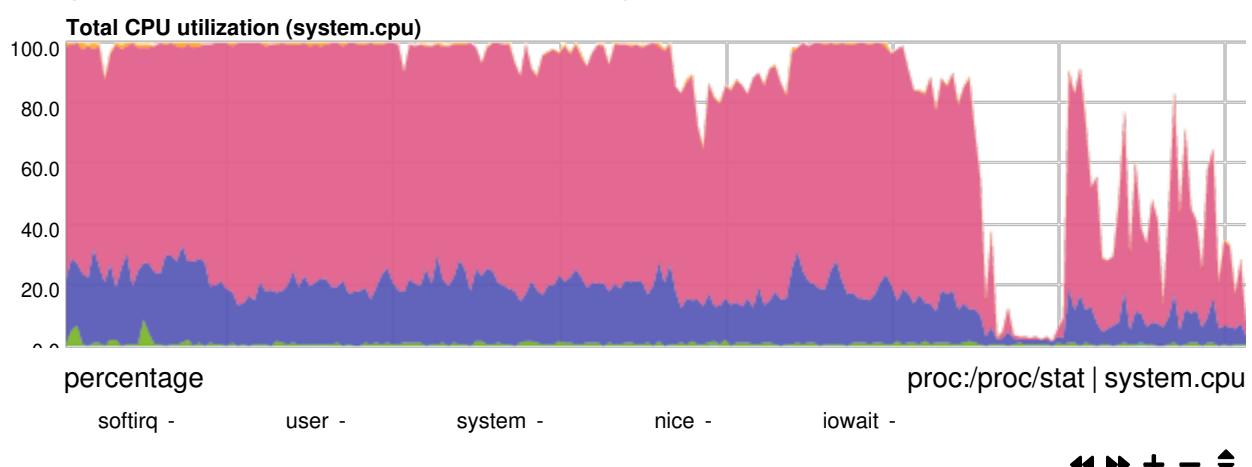
Overview of the key system metrics.

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.

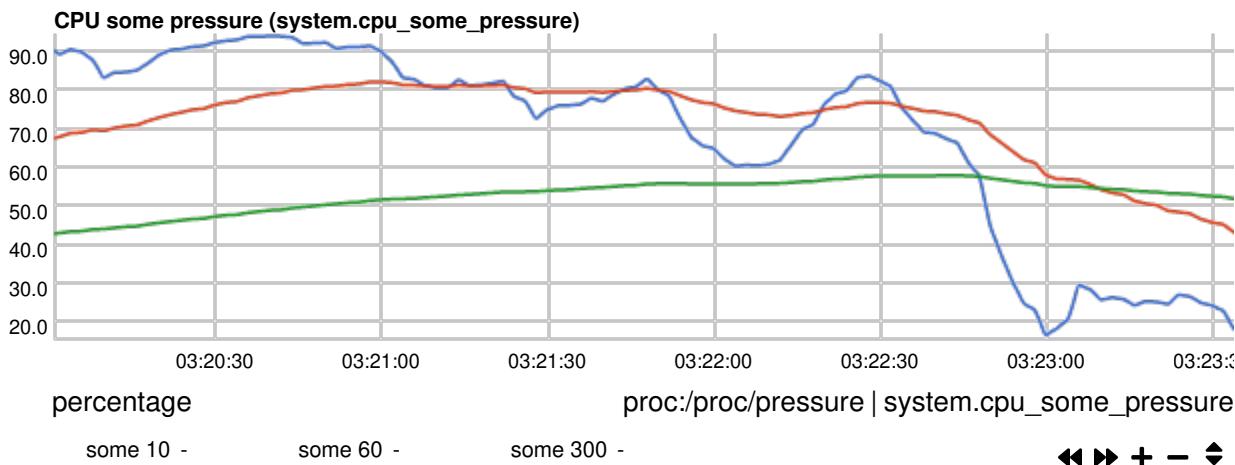
Keep an eye on **iowait**  (%). If it is constantly high, your disks are a bottleneck and they slow your system down.

An important metric worth monitoring, is **softirq**  (%). A constantly high percentage of softirq may indicate network driver issues. The individual metrics can be found in the kernel documentation (<https://www.kernel.org/doc/html/latest/filesystems/proc.html#miscellaneous-kernel-statistics-in-proc-stat>).

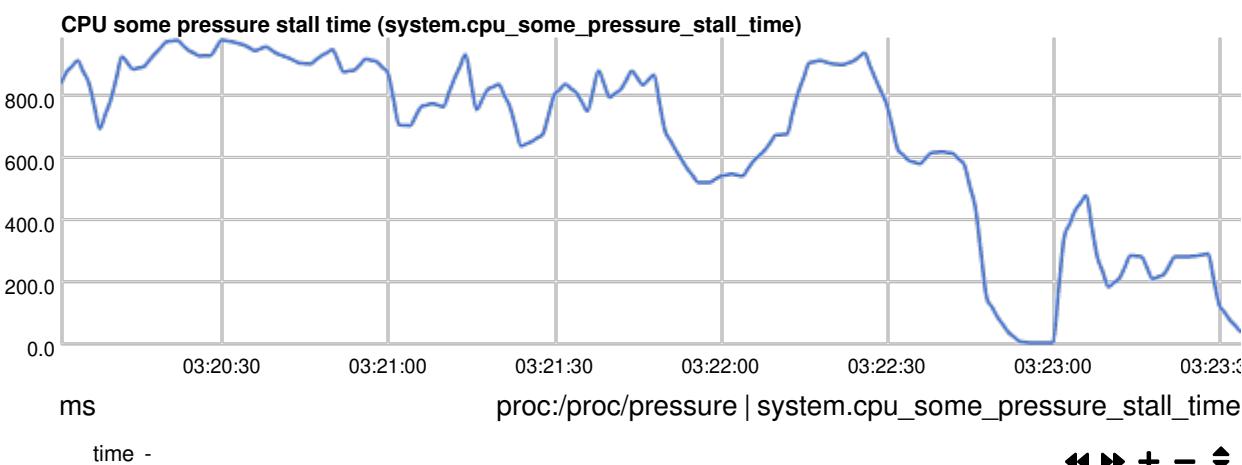


CPU Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

Some indicates the share of time in which at least **some tasks** are stalled on CPU. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.

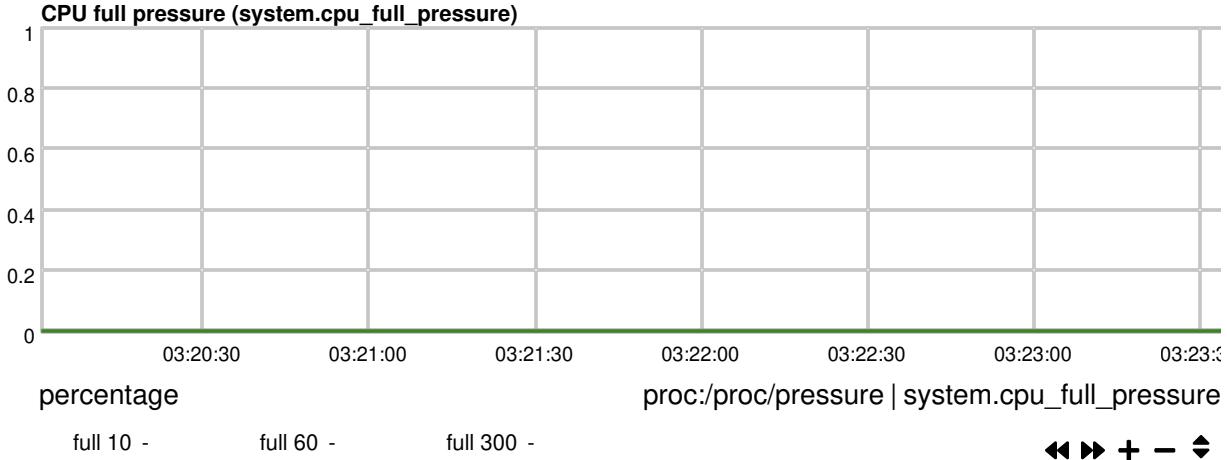


The amount of time some processes have been waiting for CPU time.

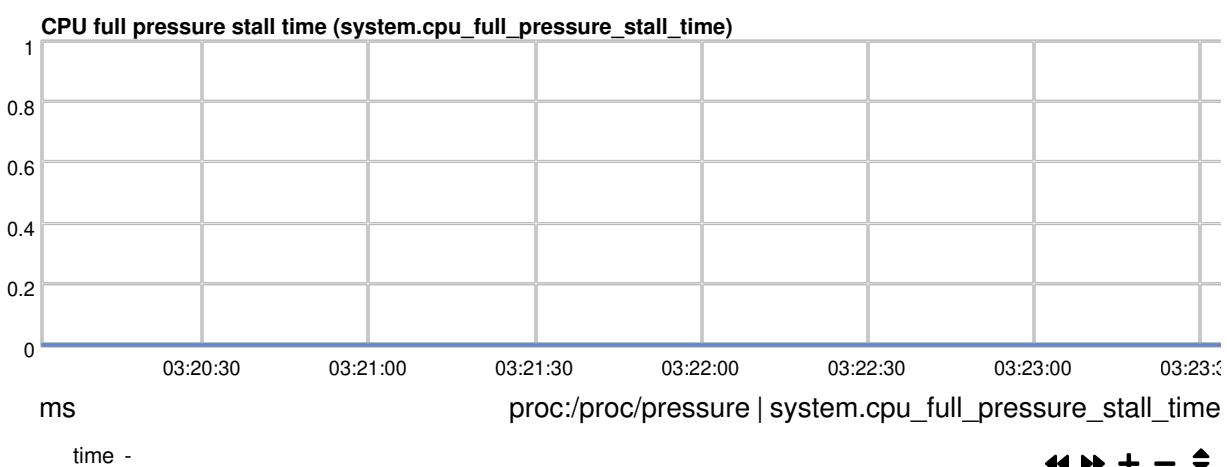


CPU Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

Full indicates the share of time in which **all non-idle tasks** are stalled on CPU resource simultaneously. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.

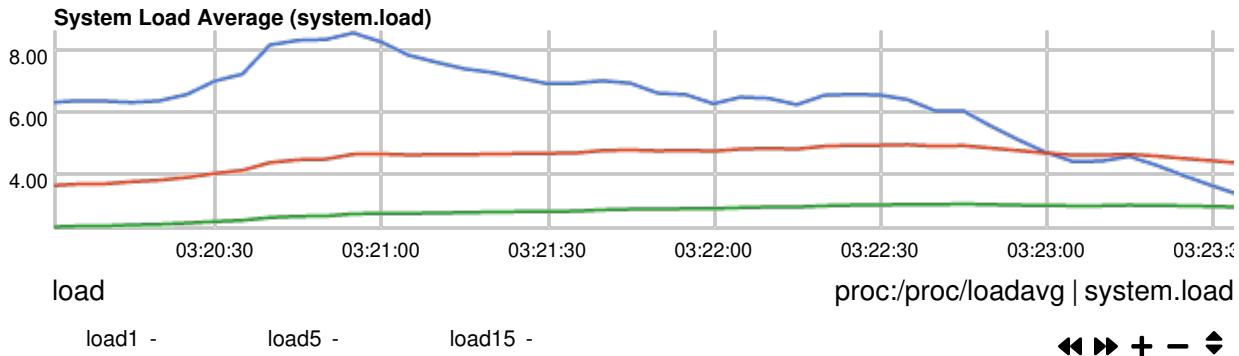


The amount of time all non-idle processes have been stalled due to CPU congestion.



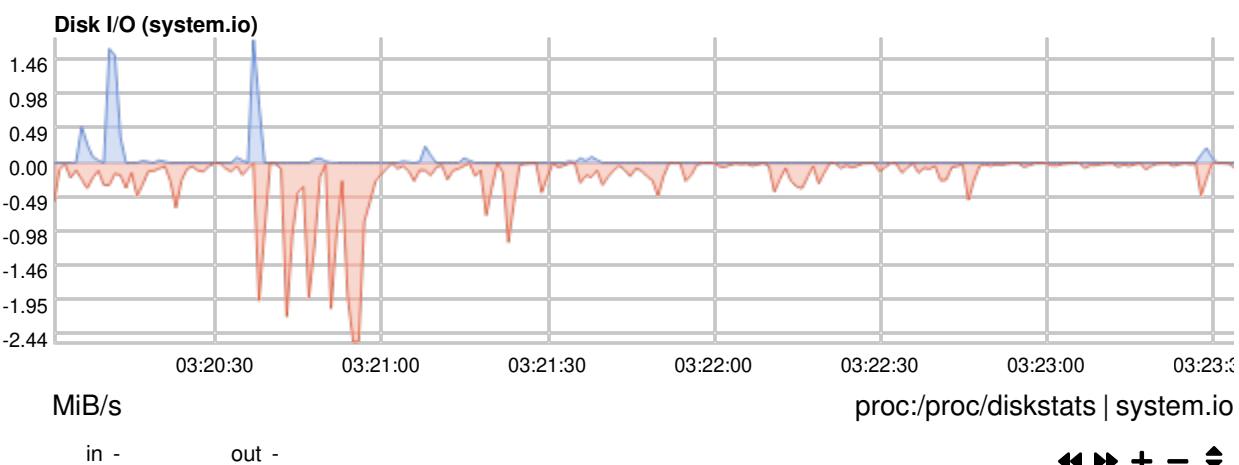
load

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_\(computing\)](https://en.wikipedia.org/wiki/Load_(computing))).

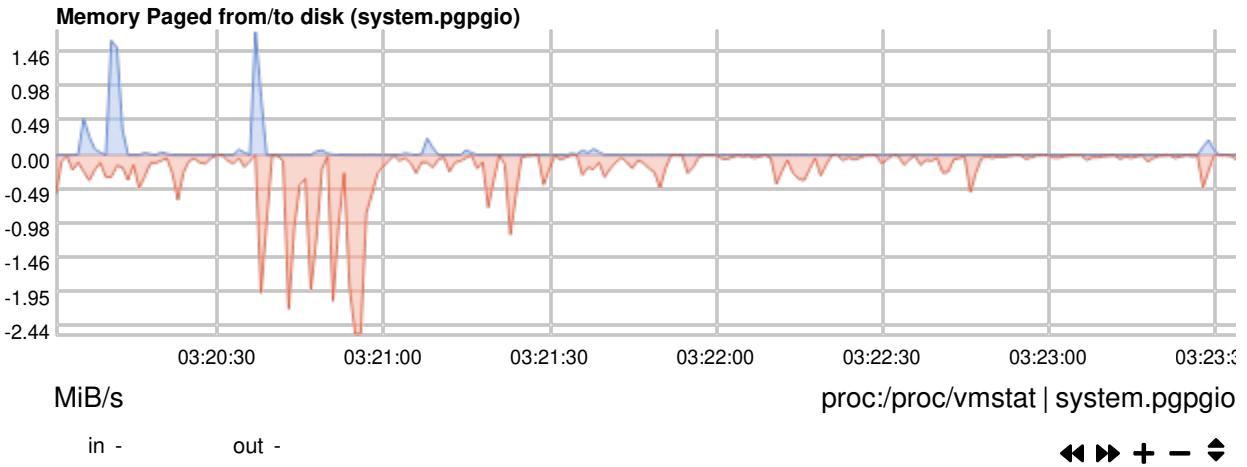


disk

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`.

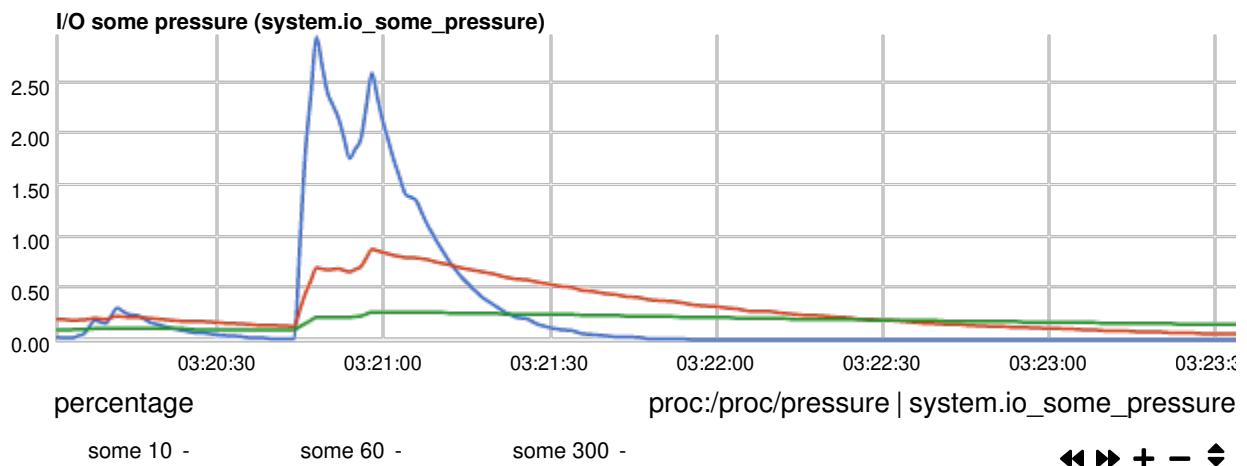


Memory paged from/to disk. This is usually the total disk I/O of the system.

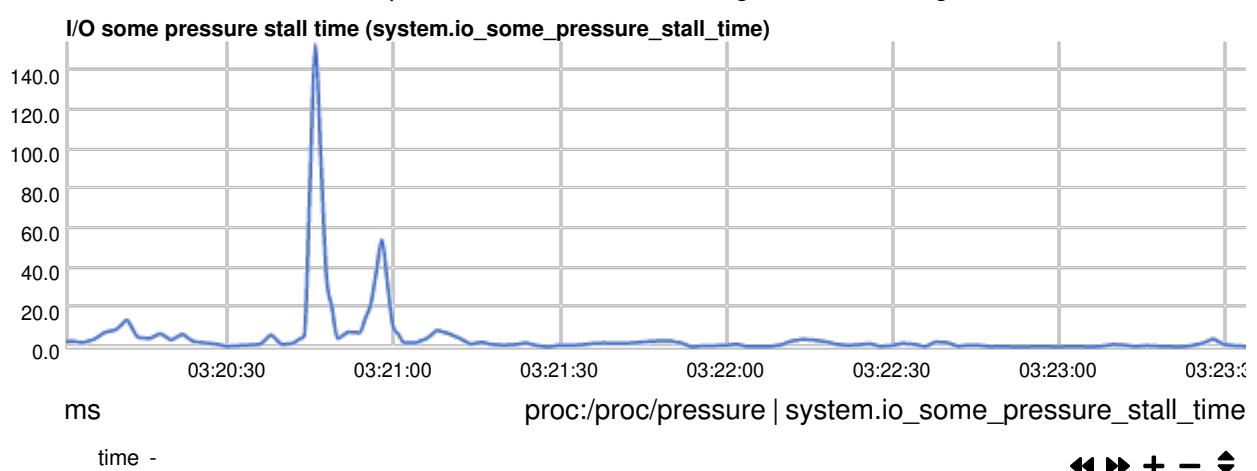


I/O Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

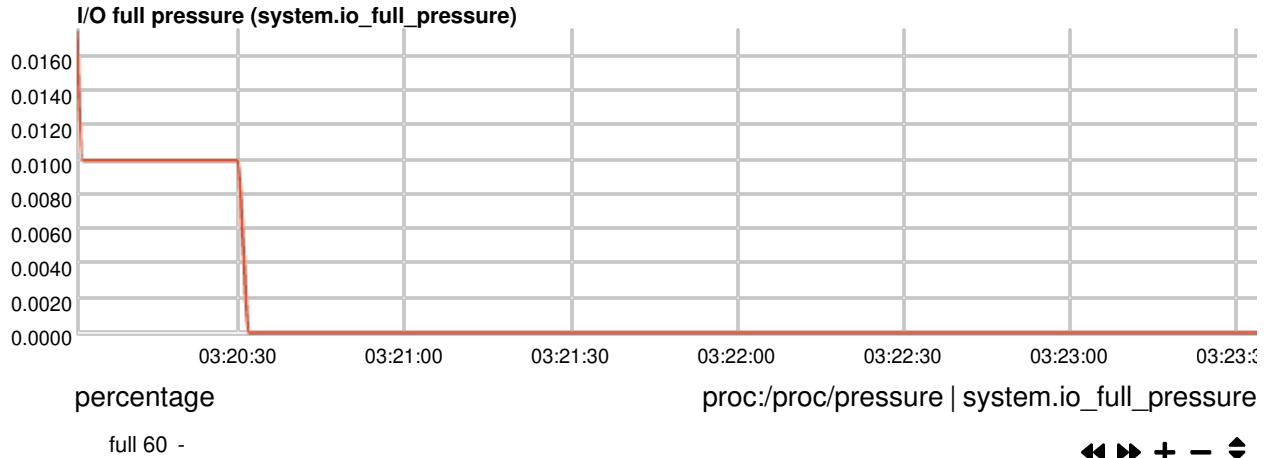
Some indicates the share of time in which at least **some tasks** are stalled on I/O. In this state the CPU is still doing productive work. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



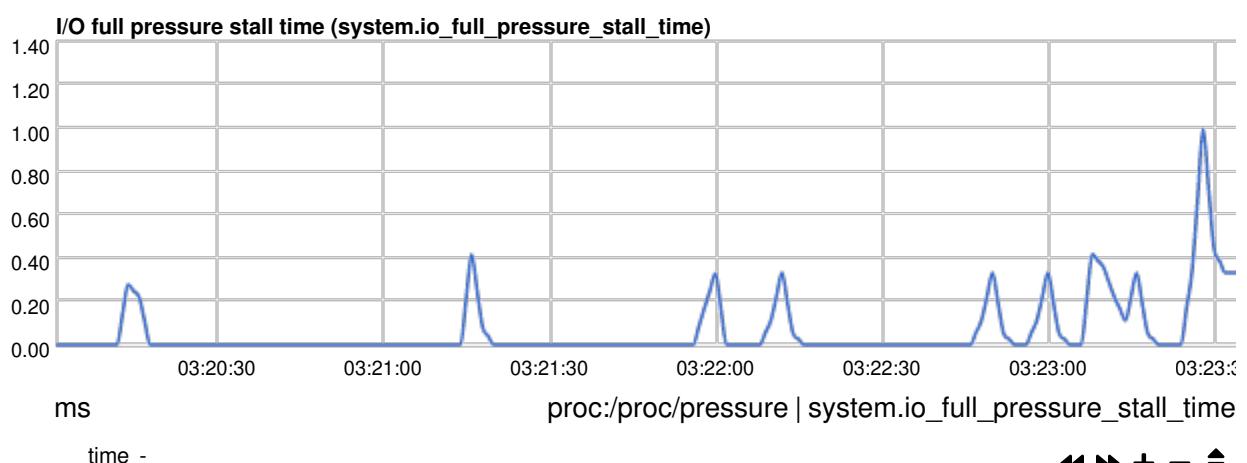
The amount of time some processes have been waiting due to I/O congestion.



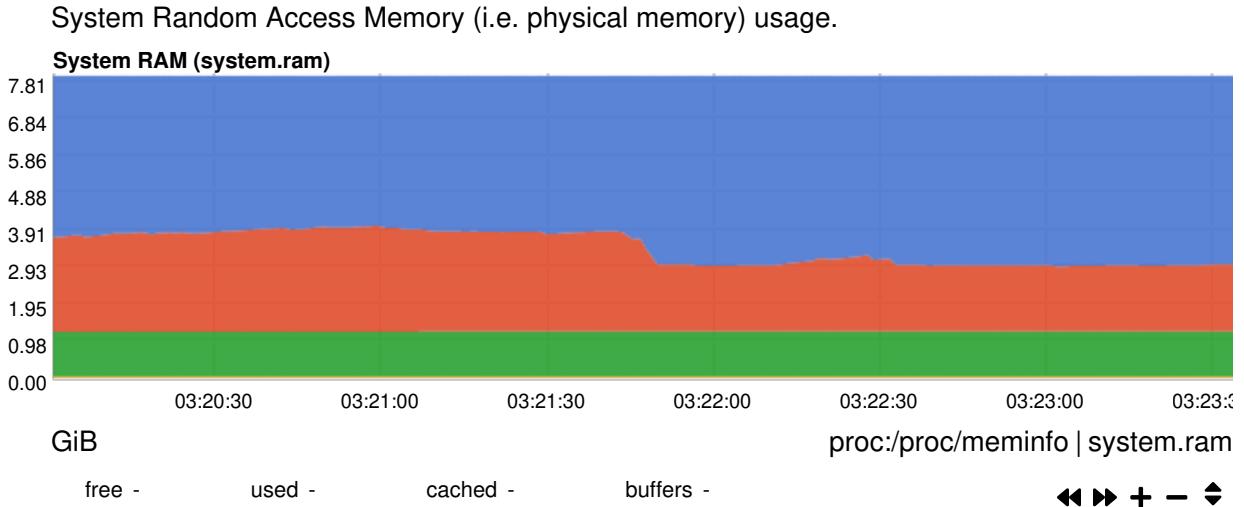
I/O Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>). **Full** line indicates the share of time in which **all non-idle tasks** are stalled on I/O resource simultaneously. In this state actual CPU cycles are going to waste, and a workload that spends extended time in this state is considered to be thrashing. This has severe impact on performance. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



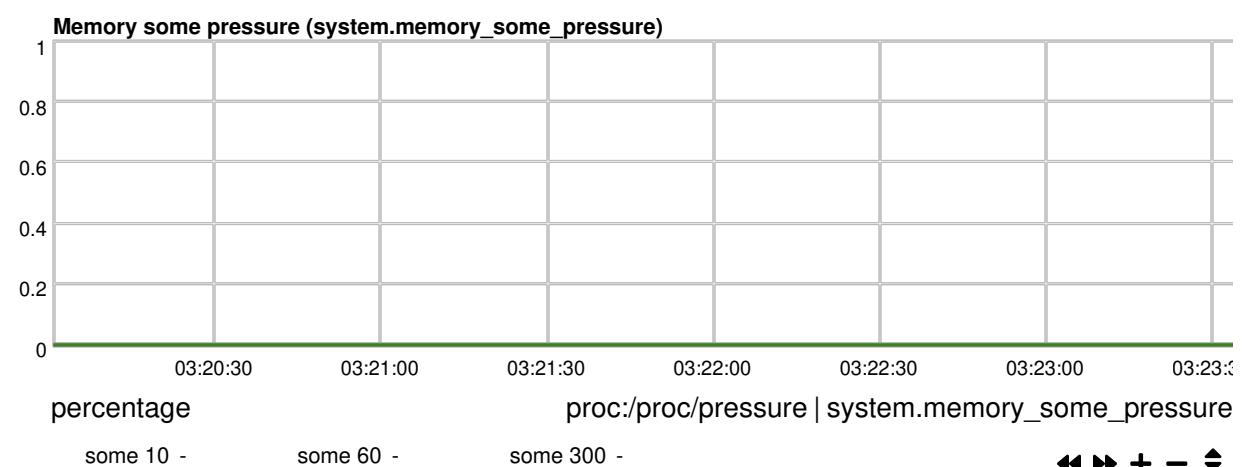
The amount of time all non-idle processes have been stalled due to I/O congestion.



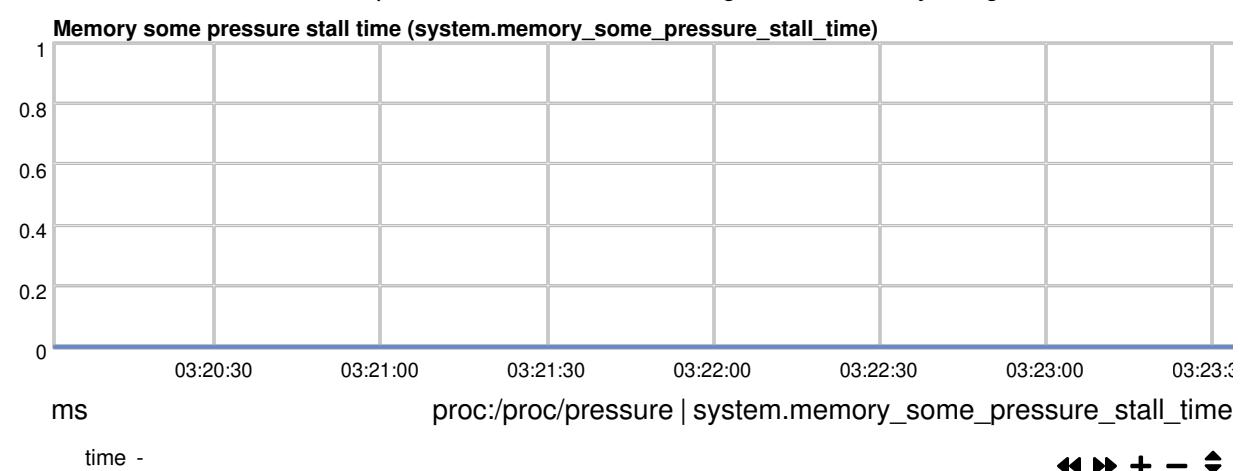
ram



Memory Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>). **Some** indicates the share of time in which at least **some tasks** are stalled on memory. In this state the CPU is still doing productive work. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



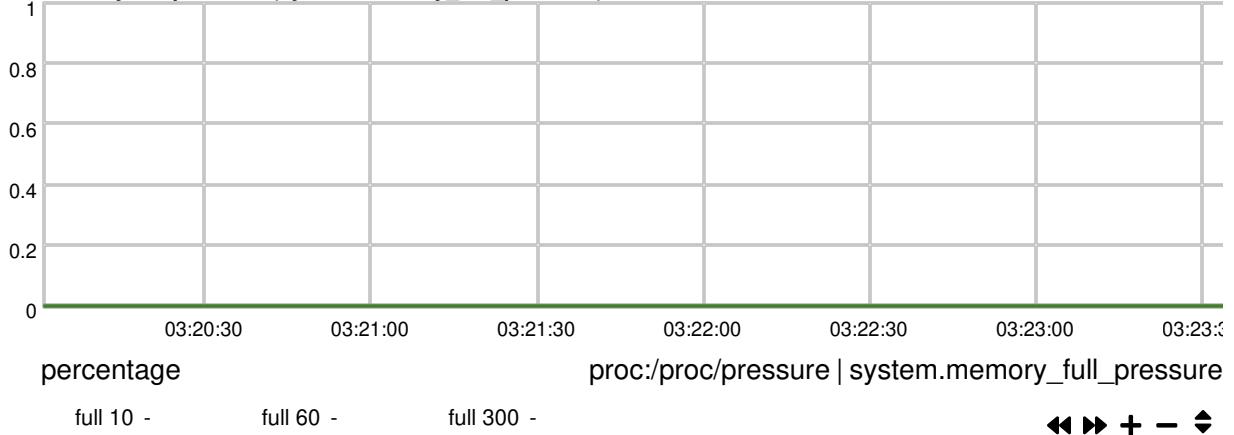
The amount of time some processes have been waiting due to memory congestion.



Memory Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

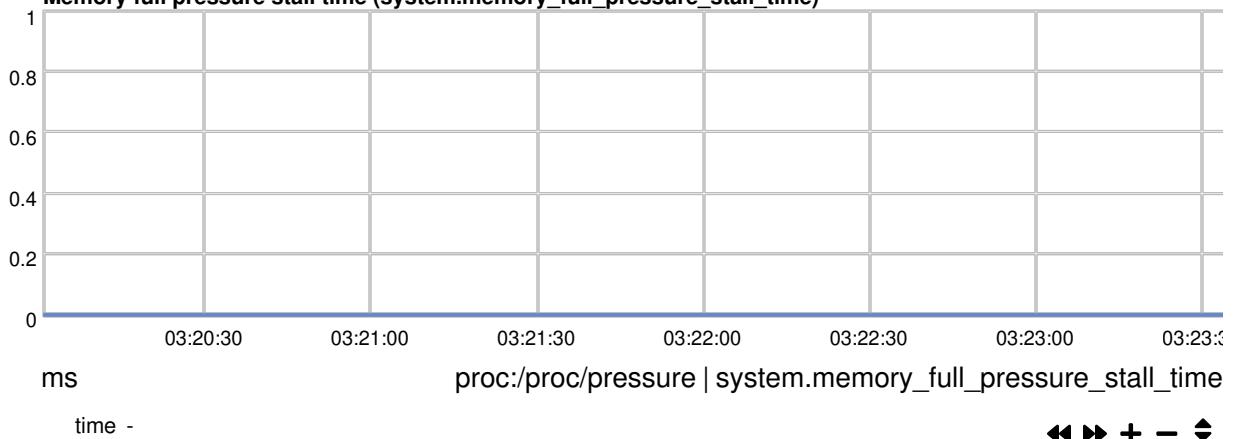
Full indicates the share of time in which **all non-idle tasks** are stalled on memory resource simultaneously. In this state actual CPU cycles are going to waste, and a workload that spends extended time in this state is considered to be thrashing. This has severe impact on performance. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.

Memory full pressure (system.memory_full_pressure)



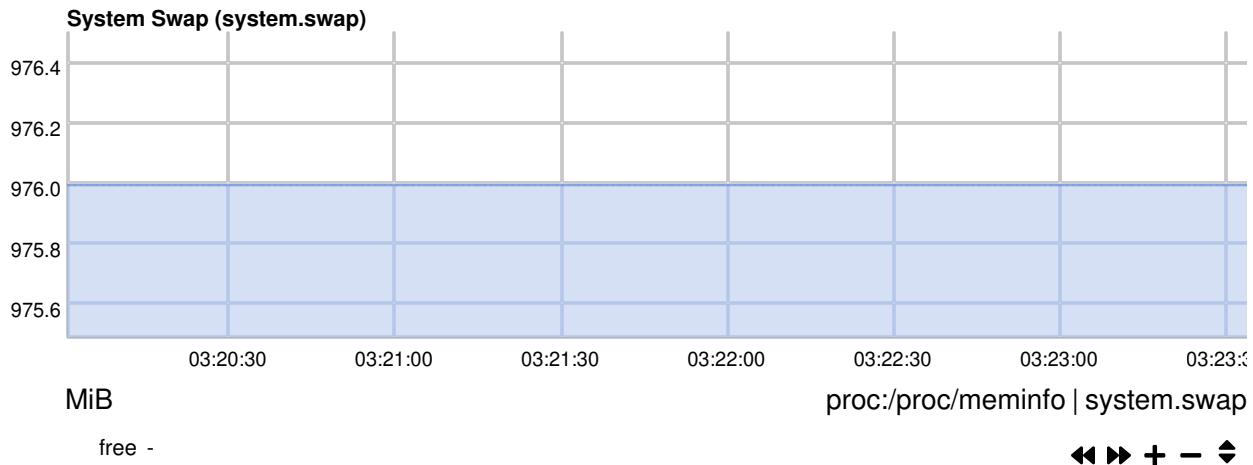
The amount of time all non-idle processes have been stalled due to memory congestion.

Memory full pressure stall time (system.memory_full_pressure_stall_time)



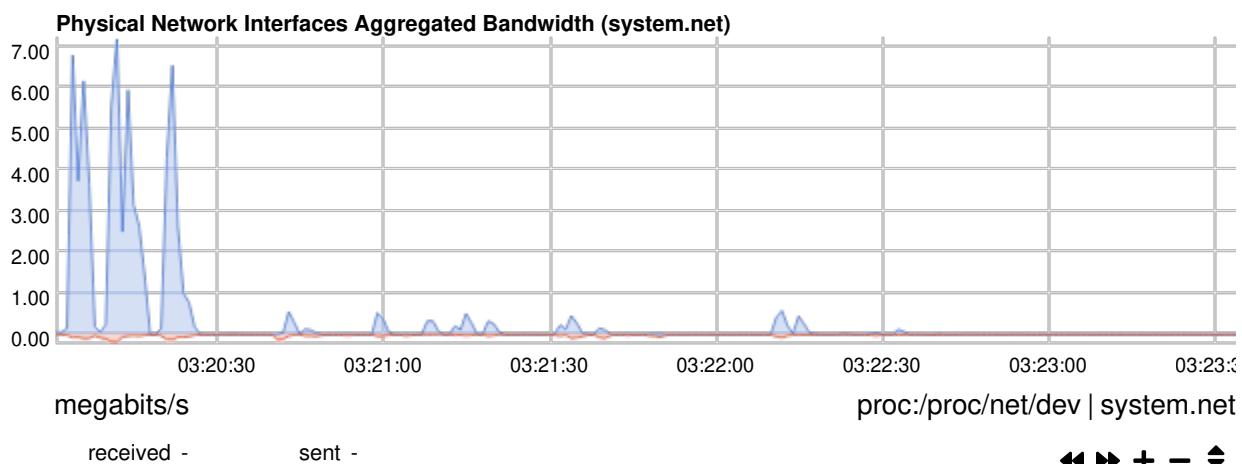
swap

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

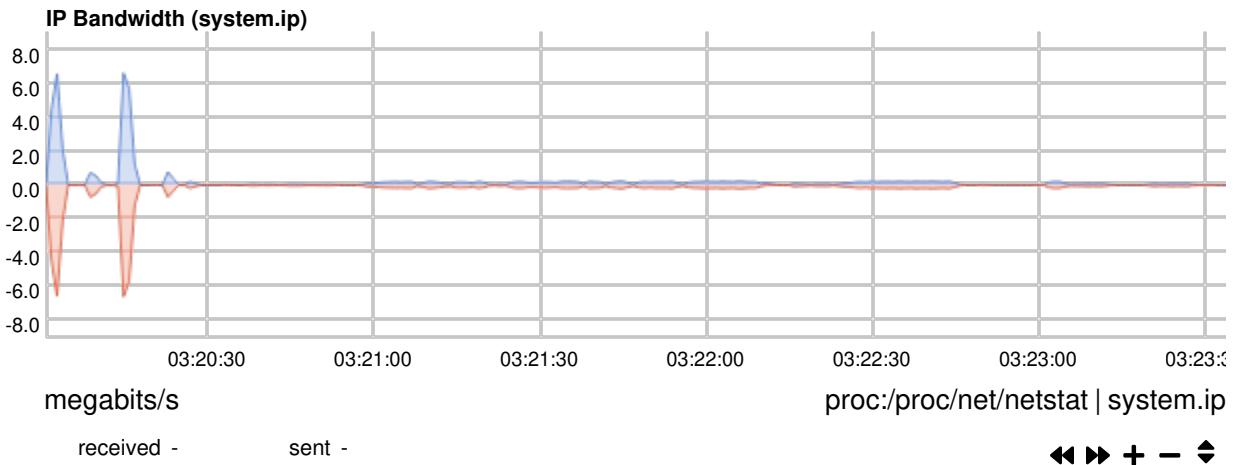


network

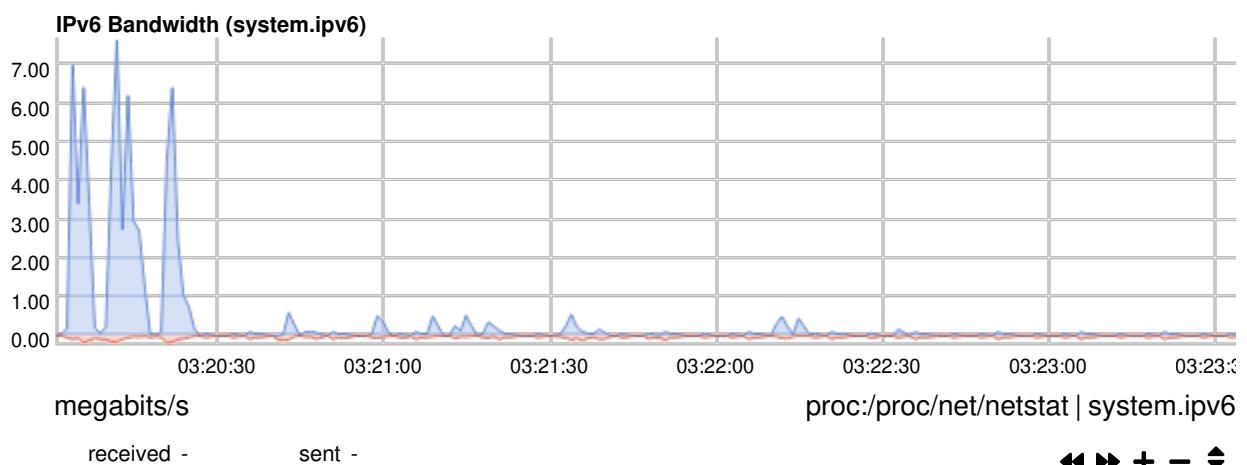
Total bandwidth of all physical network interfaces. This does not include `lo`, VPNs, network bridges, IFB devices, bond interfaces, etc. Only the bandwidth of physical network interfaces is aggregated. Physical are all the network interfaces that are listed in `/proc/net/dev`, but do not exist in `/sys/devices/virtual/net`.



Total IP traffic in the system.



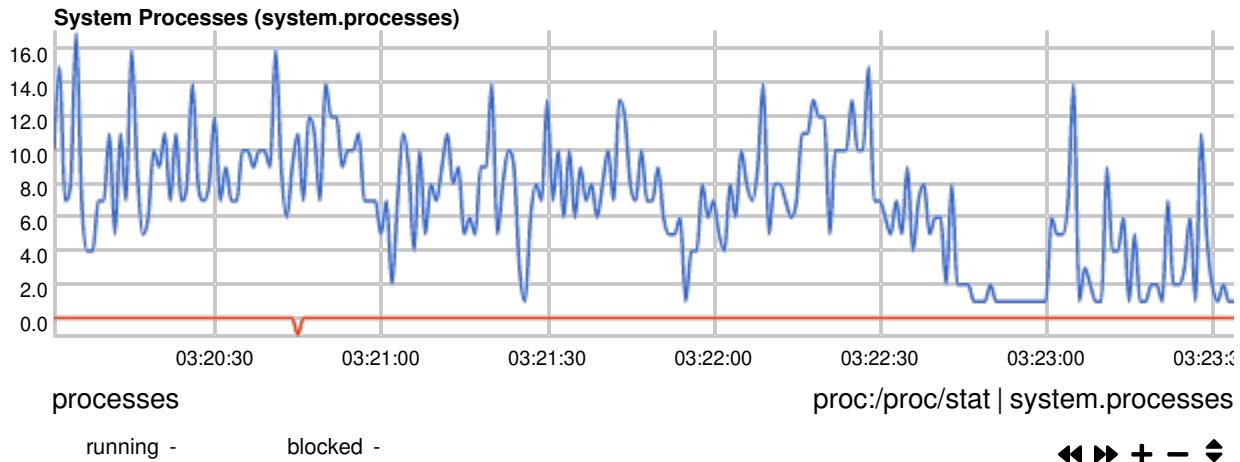
Total IPv6 Traffic.



processes

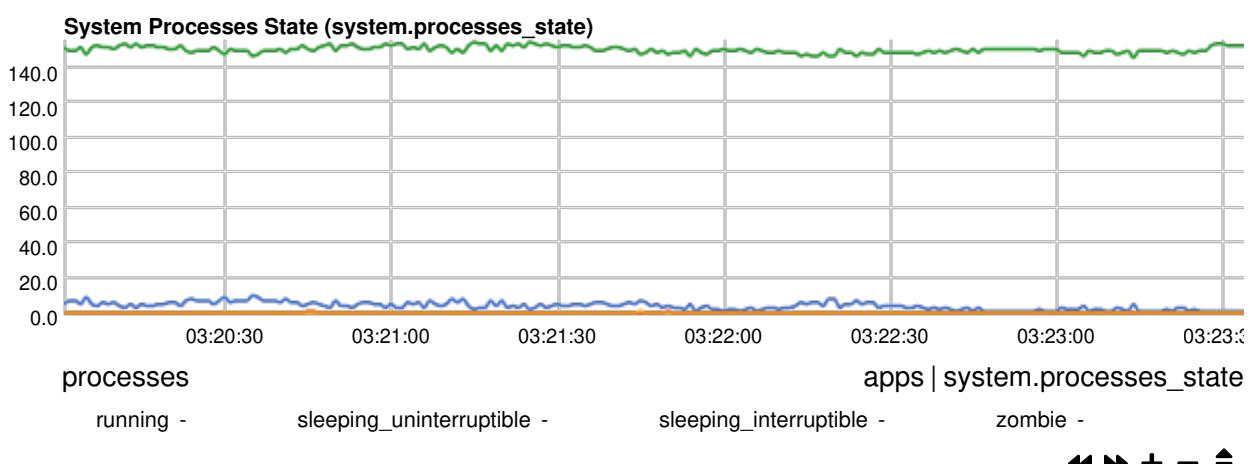
System processes.

Running - running or ready to run (runnable). **Blocked** - currently blocked, waiting for I/O to complete.

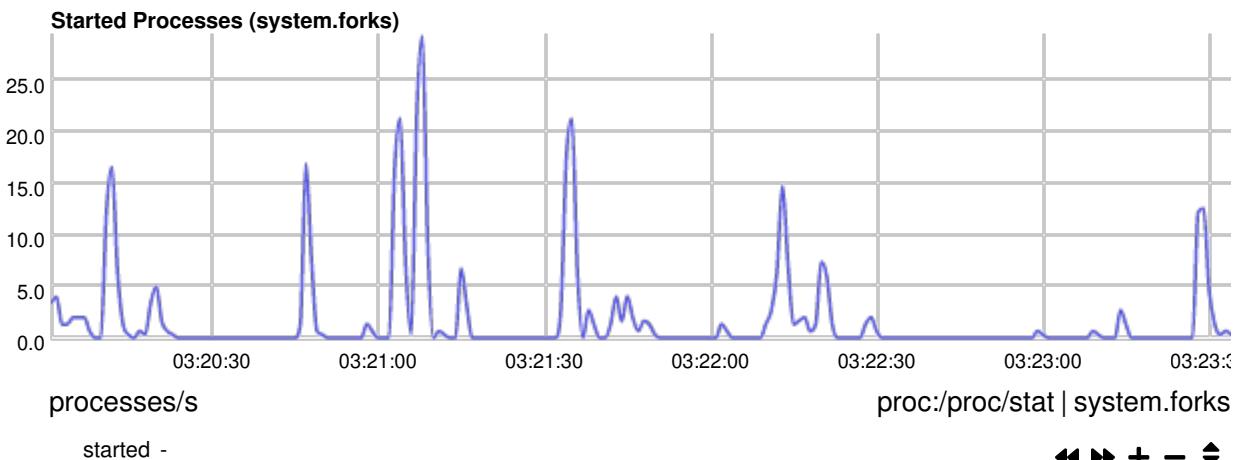


The number of processes in different states.

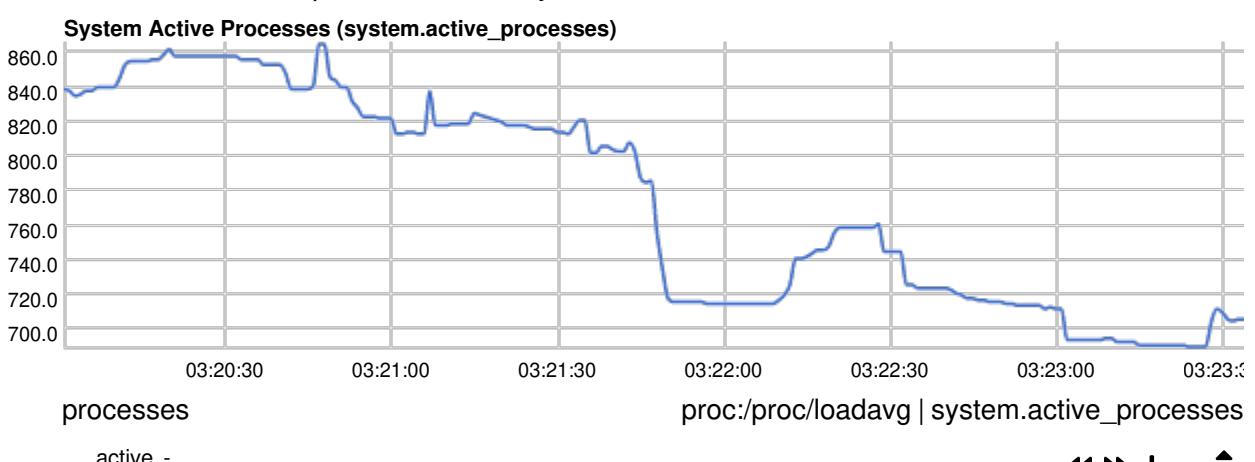
Running - Process using the CPU at a particular moment. **Sleeping (uninterruptible)** - Process will wake when a waited-upon resource becomes available or after a time-out occurs during that wait. Mostly used by device drivers waiting for disk or network I/O. **Sleeping (interruptible)** - Process is waiting either for a particular time slot or for a particular event to occur. **Zombie** - Process that has completed its execution, released the system resources, but its entry is not removed from the process table. Usually occurs in child processes when the parent process still needs to read its child's exit status. A process that stays a zombie for a long time is generally an error and causes system PID space leak. **Stopped** - Process is suspended from proceeding further due to STOP or TSTP signals. In this state, a process will not do anything (not even terminate) until it receives a CONT signal.



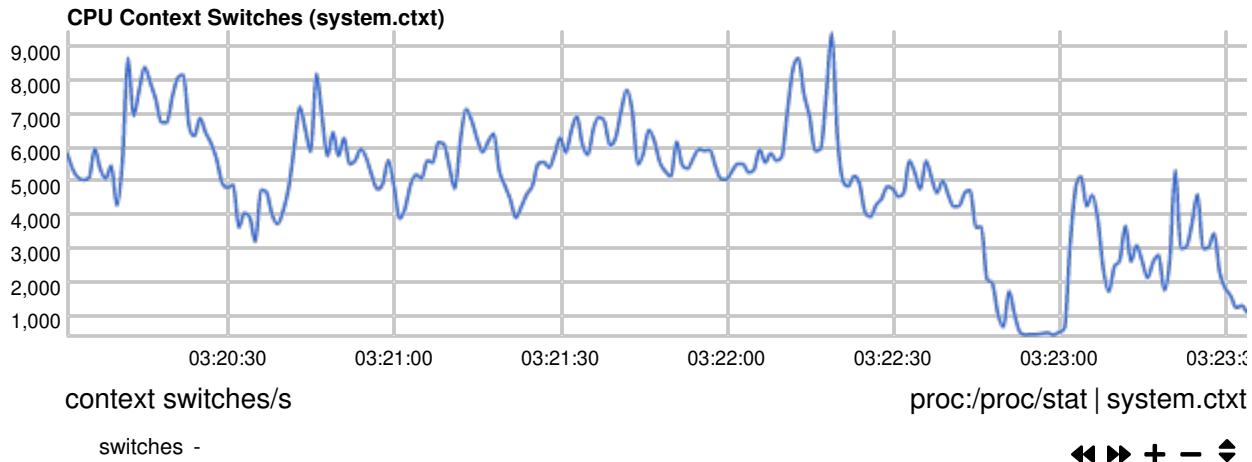
The number of new processes created.



The total number of processes in the system.

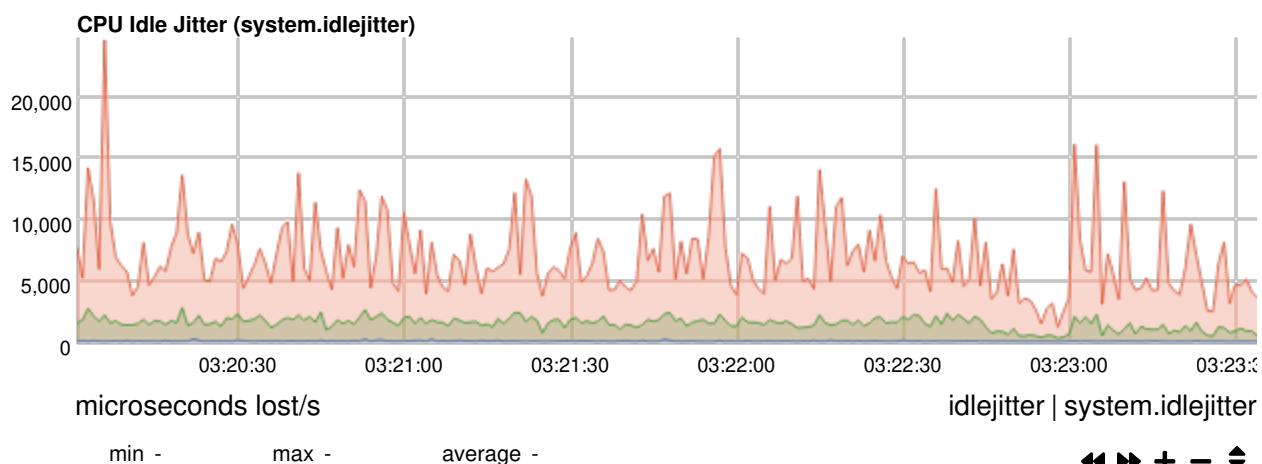


Context Switches (https://en.wikipedia.org/wiki/Context_switch), is the switching of the CPU from one process, task or thread to another. If there are many processes or threads willing to execute and very few CPU cores available to handle them, the system is making more context switching to balance the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

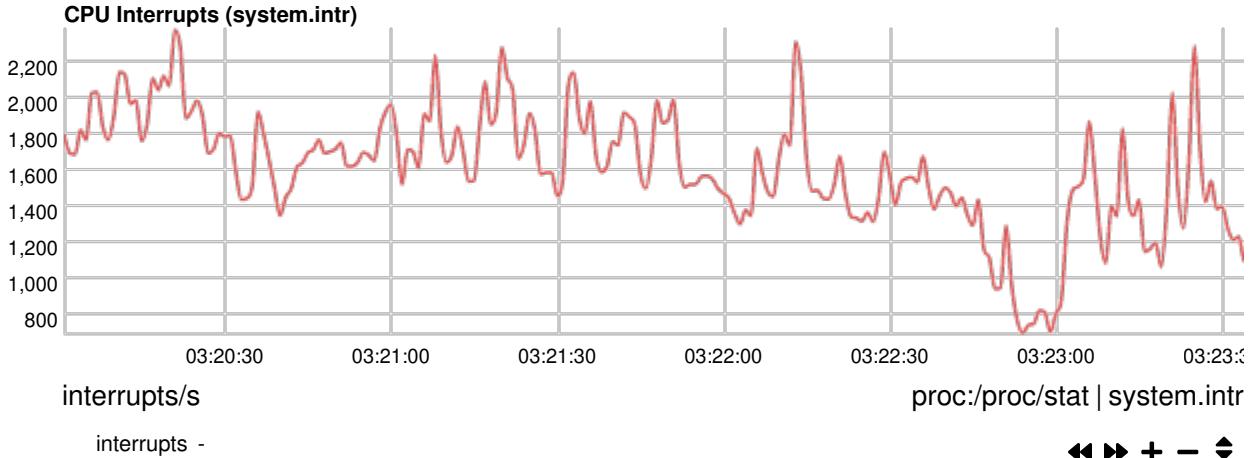


interrupts

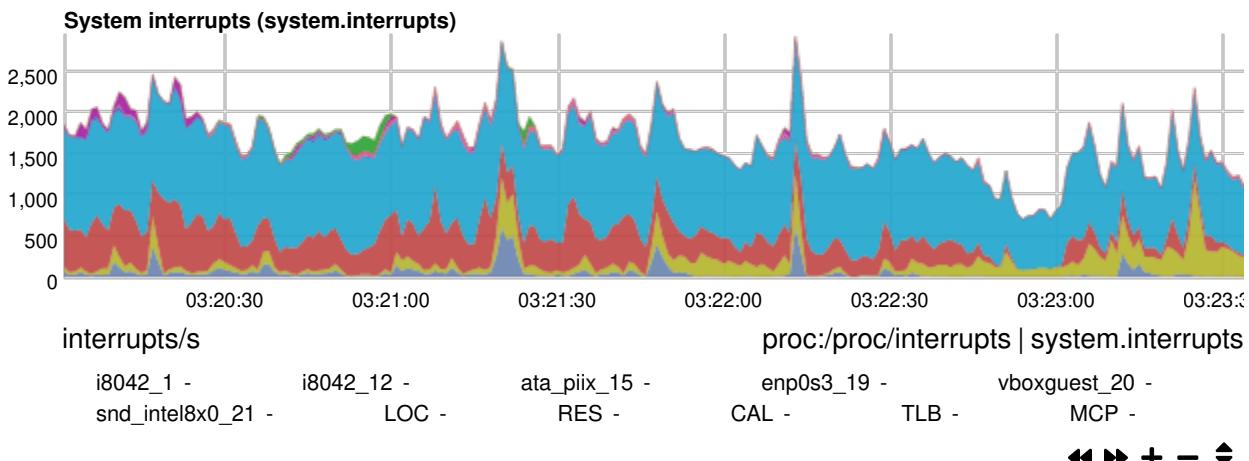
Interrupts (<https://en.wikipedia.org/wiki/Interrupt>) are signals sent to the CPU by external devices (normally I/O devices) or programs (running processes). They tell the CPU to stop its current activities and execute the appropriate part of the operating system. Interrupt types are

hardware (generated by hardware devices to signal that they need some attention from the OS), **software** (generated by programs when they want to request a system call to be performed by the operating system), and **traps** (generated by the CPU itself to indicate that some error or condition occurred for which assistance from the operating system is needed).

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. The last column in `/proc/interrupts` provides an interrupt description or the device name that registered the handler for that interrupt.

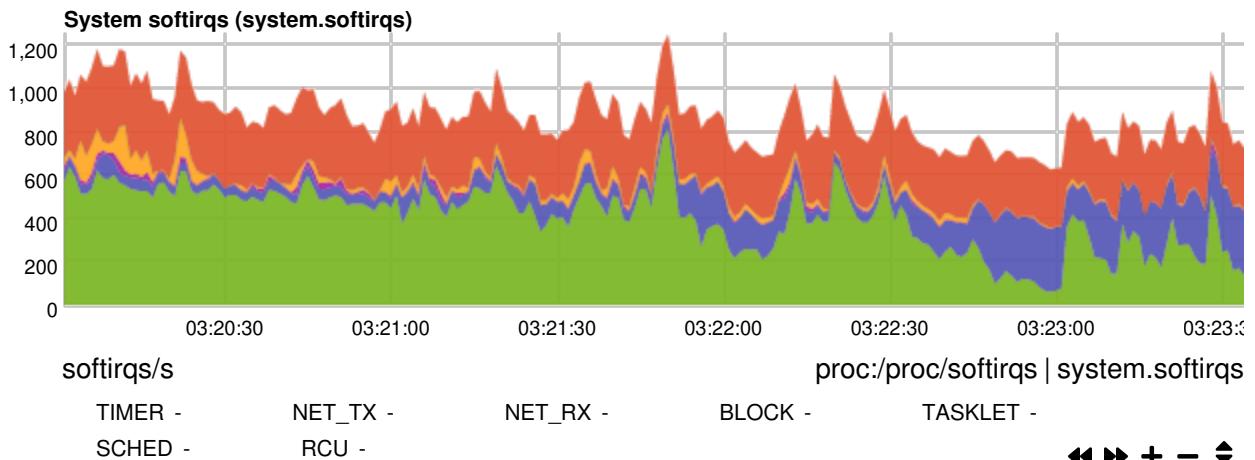


softirqs

Software interrupts (or "softirqs") are one of the oldest deferred-execution mechanisms in the kernel. Several tasks among those executed by the kernel are not critical: they can be deferred for a long period of time, if necessary. The deferrable tasks can execute with all interrupts enabled (softirqs are patterned after hardware interrupts). Taking them out of the interrupt handler helps keep kernel response time small.

Total number of software interrupts in the system. At the CPUs section, softirqs are analyzed per CPU core.

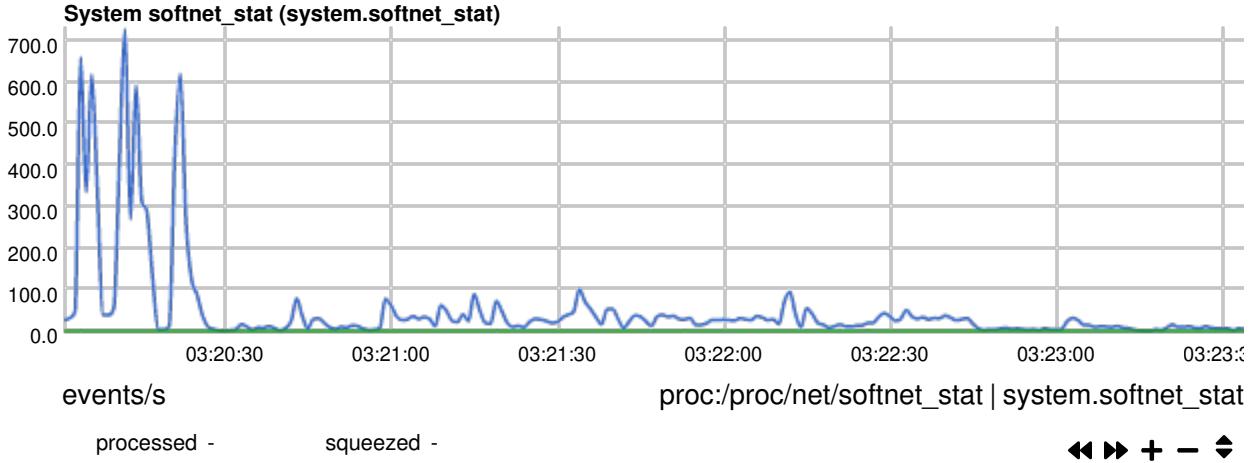
HI - high priority tasklets. **TIMER** - tasklets related to timer interrupts. **NET_TX**, **NET_RX** - used for network transmit and receive processing. **BLOCK** - handles block I/O completion events. **IRQ_POLL** - used by the IO subsystem to increase performance (a NAPI like approach for block devices). **TASKLET** - handles regular tasklets. **SCHED** - used by the scheduler to perform load-balancing and other scheduling tasks. **HRTIMER** - used for high-resolution timers. **RCU** - performs read-copy-update (RCU) processing.



softnet

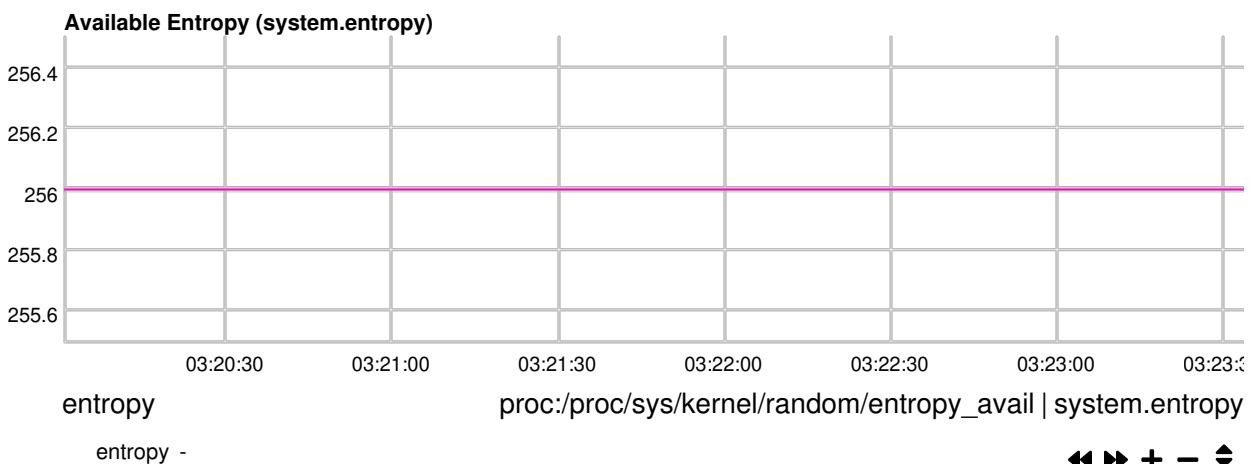
Statistics for CPUs SoftIRQs related to network receive work. Break down per CPU core can be found at CPU / softnet statistics. More information about identifying and troubleshooting network driver related issues can be found at Red Hat Enterprise Linux Network Performance Tuning Guide (https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf).

Processed - packets processed. **Dropped** - packets dropped because the network device backlog was full. **Squeezed** - number of times the network device budget was consumed or the time limit was reached, but more work was available. **ReceivedRPS** - number of times this CPU has been woken up to process packets via an Inter-processor Interrupt. **FlowLimitCount** - number of times the flow limit has been reached (flow limiting is an optional Receive Packet Steering feature).

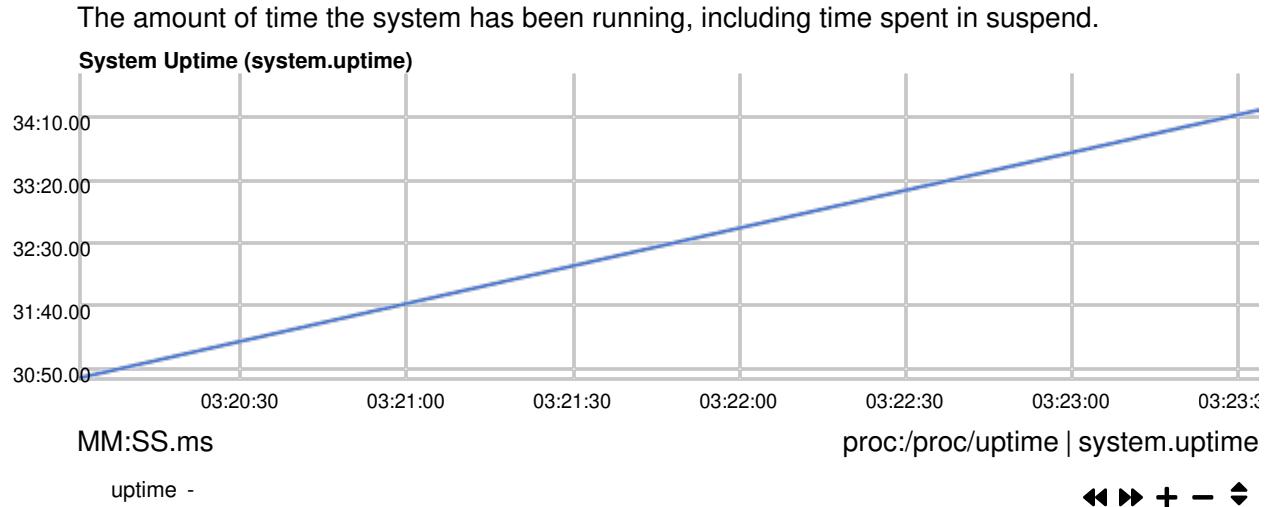


entropy

Entropy ([https://en.wikipedia.org/wiki/Entropy_\(computing\)](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

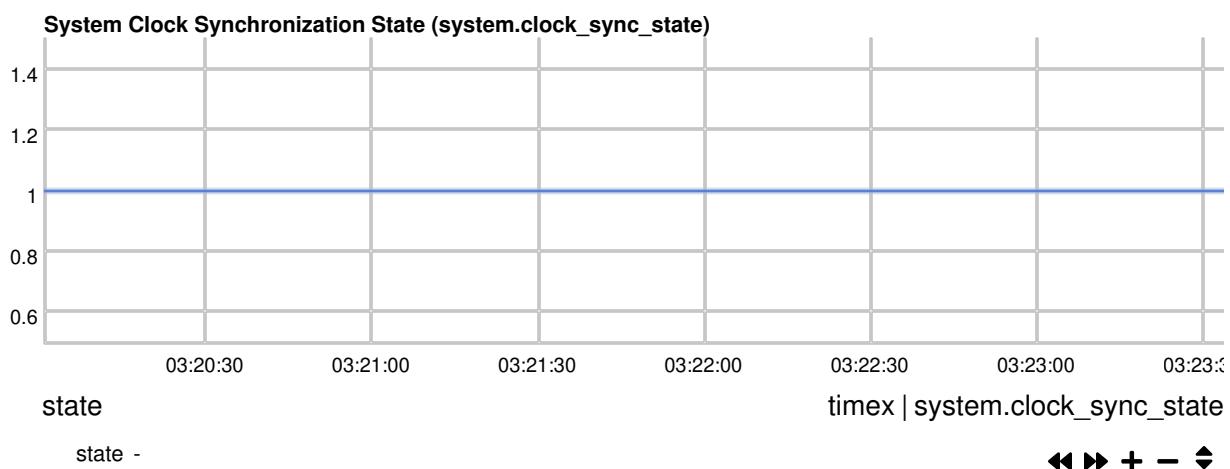


clock synchronization

NTP (https://en.wikipedia.org/wiki/Network_Time_Protocol) lets you automatically sync your system time with a remote server. This keeps your machine's time accurate by syncing with servers that are known to have accurate times.

The system clock synchronization state as provided by the `ntp_adjtime()` (<https://man7.org/linux/man-pages/man2/adjtimex.2.html>) system call. An unsynchronized clock may be the result of synchronization issues by the NTP daemon or a hardware clock fault. It can take several minutes (usually up to 17) before NTP daemon selects a server to synchronize with.

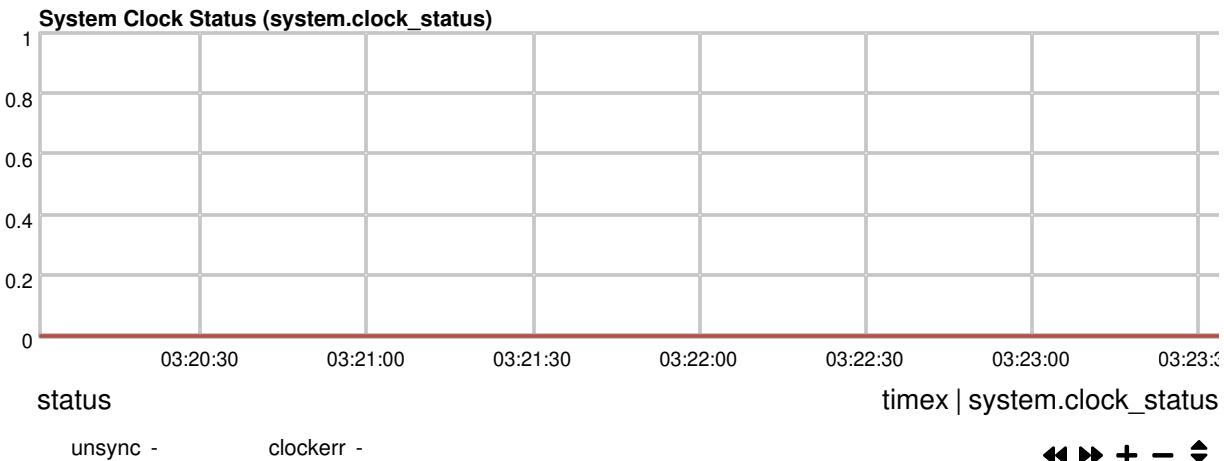
State map: 0 - not synchronized, 1 - synchronized.



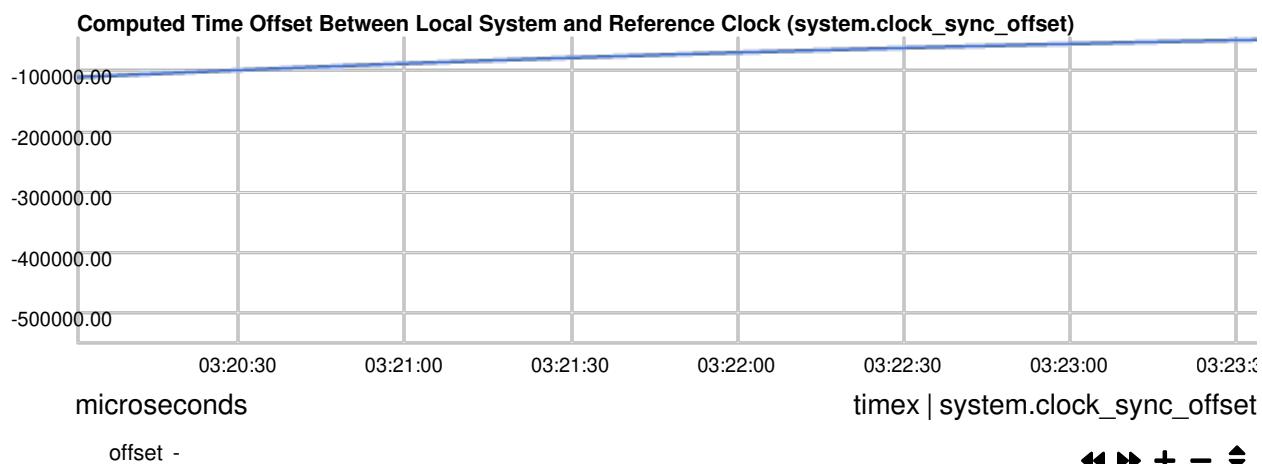
The kernel code can operate in various modes and with various features enabled or disabled, as selected by the `ntp_adjtime()` (<https://man7.org/linux/man-pages/man2/adjtimex.2.html>) system call. The system clock status shows the value of the `time_status` variable in the kernel. The bits of the variable are used to control these functions and record error conditions as they exist.

UNSYNC - set/cleared by the caller to indicate clock unsynchronized (e.g., when no peers are reachable). This flag is usually controlled by an application program, but the operating system may also set it. **CLOCKERR** - set/cleared by the external hardware clock driver to indicate hardware fault.

Status map: 0 - bit unset, 1 - bit set.



A typical NTP client regularly polls one or more NTP servers. The client must compute its time offset (https://en.wikipedia.org/wiki/Network_Time_Protocol#Clock_synchronization_algorithm) and round-trip delay. Time offset is the difference in absolute time between the two clocks.

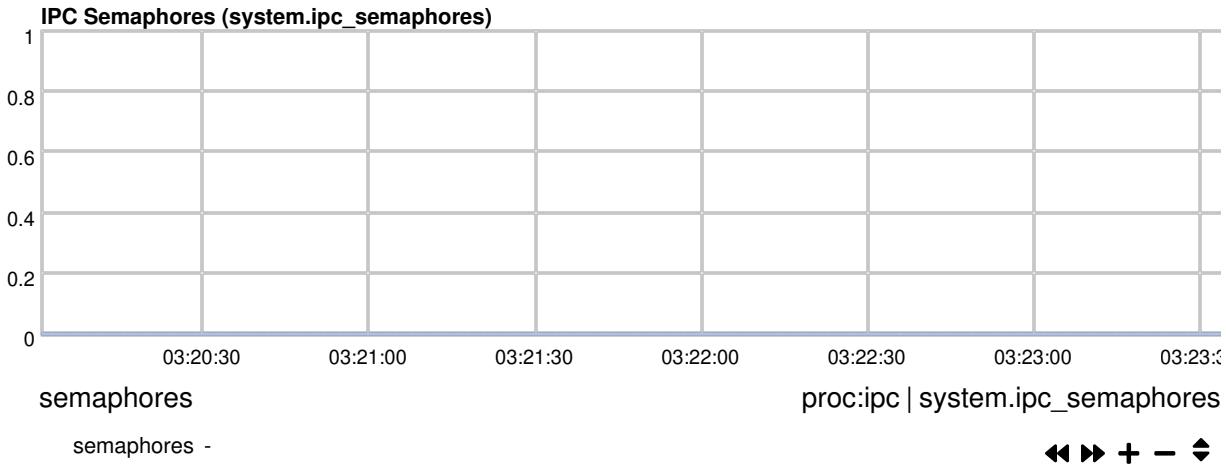


ipc semaphores

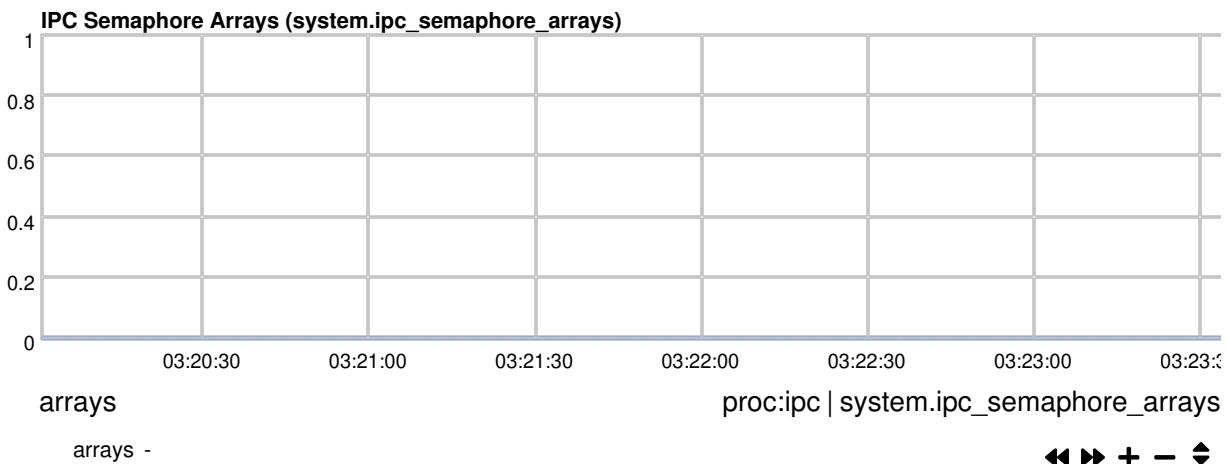
System V semaphores is an inter-process communication (IPC) mechanism. It allows processes or threads within a process to synchronize their actions. They are often used to

monitor and control the availability of system resources such as shared memory segments. For details, see `svipc(7)` (<https://man7.org/linux/man-pages/man7/sv ipc.7.html>). To see the host IPC semaphore information, run `ipcs -us`. For limits, run `ipcs -ls`.

Number of allocated System V IPC semaphores. The system-wide limit on the number of semaphores in all semaphore sets is specified in `/proc/sys/kernel/sem` file (2nd field).



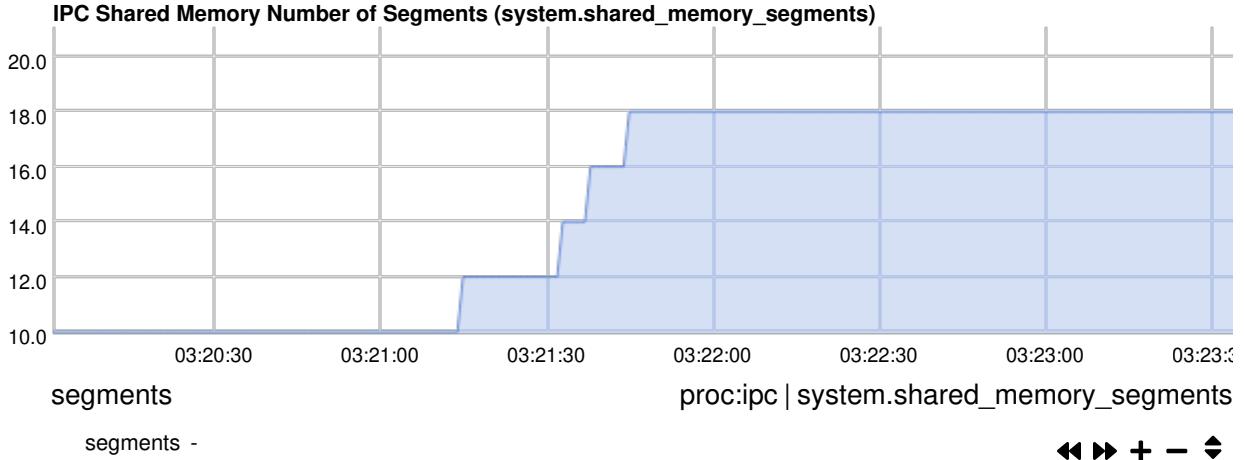
Number of used System V IPC semaphore arrays (sets). Semaphores support semaphore sets where each one is a counting semaphore. So when an application requests semaphores, the kernel releases them in sets. The system-wide limit on the maximum number of semaphore sets is specified in `/proc/sys/kernel/sem` file (4th field).



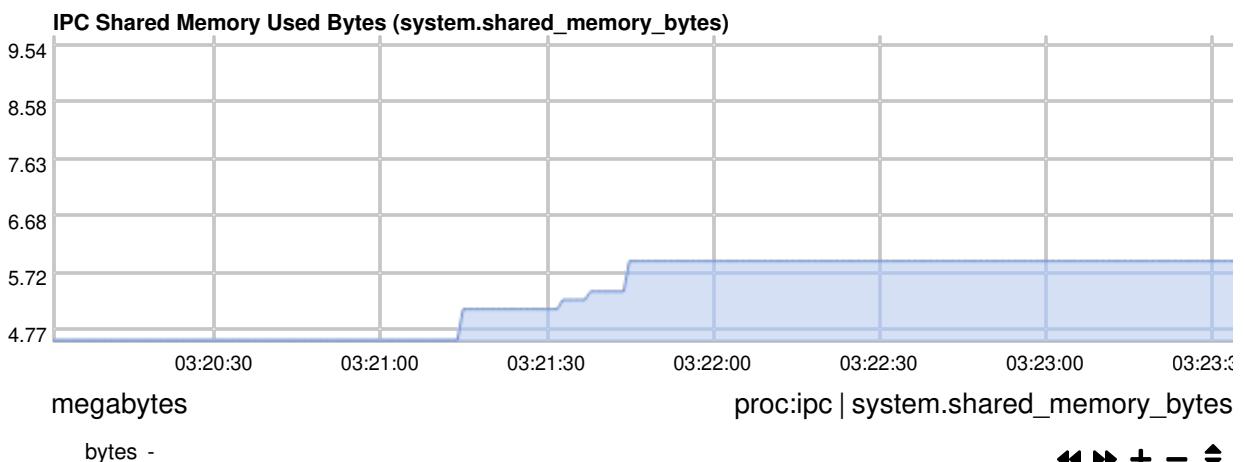
ipc shared memory

System V shared memory is an inter-process communication (IPC) mechanism. It allows processes to communicate information by sharing a region of memory. It is the fastest form of inter-process communication available since no kernel involvement occurs when data is passed between the processes (no copying). Typically, processes must synchronize their access to a shared memory object, using, for example, POSIX semaphores. For details, see `svipc(7)` (<https://man7.org/linux/man-pages/man7/sv ipc.7.html>). To see the host IPC shared memory information, run `ipcs -um`. For limits, run `ipcs -lm`.

Number of allocated System V IPC memory segments. The system-wide maximum number of shared memory segments that can be created is specified in `/proc/sys/kernel/shmmni` file.



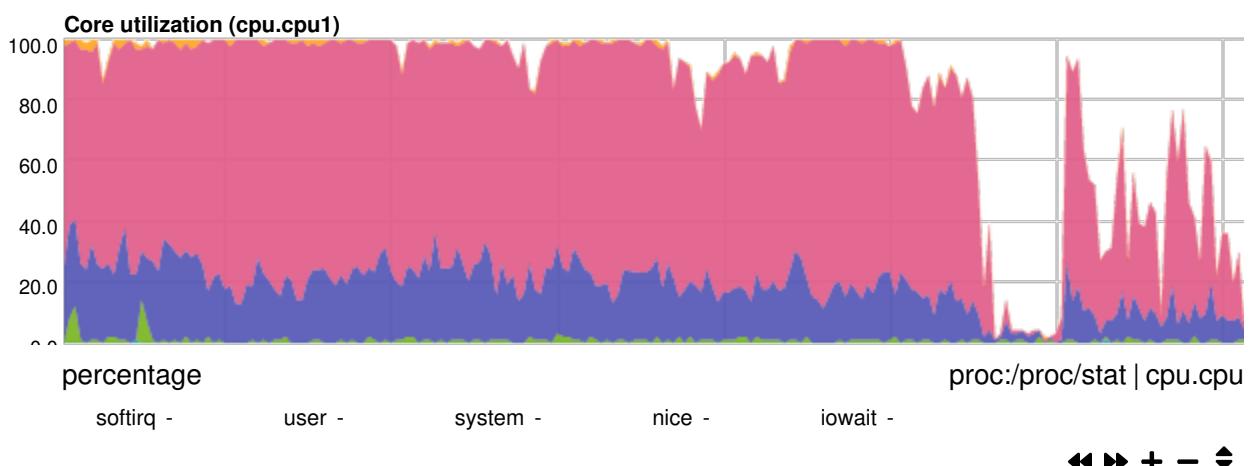
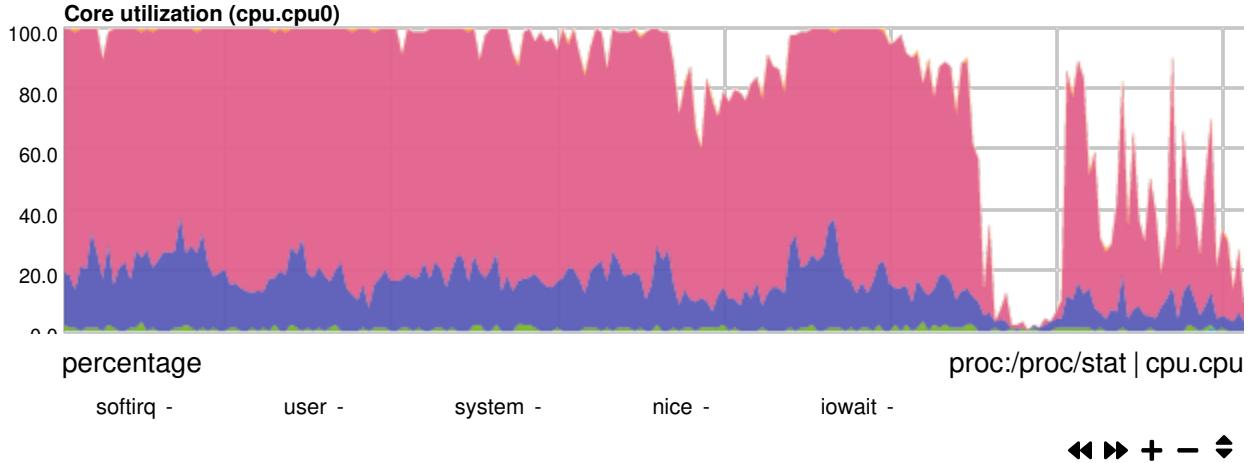
Amount of memory currently used by System V IPC memory segments. The run-time limit on the maximum shared memory segment size that can be created is specified in `/proc/sys/kernel/shmmax` file.



⚡ CPUs

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

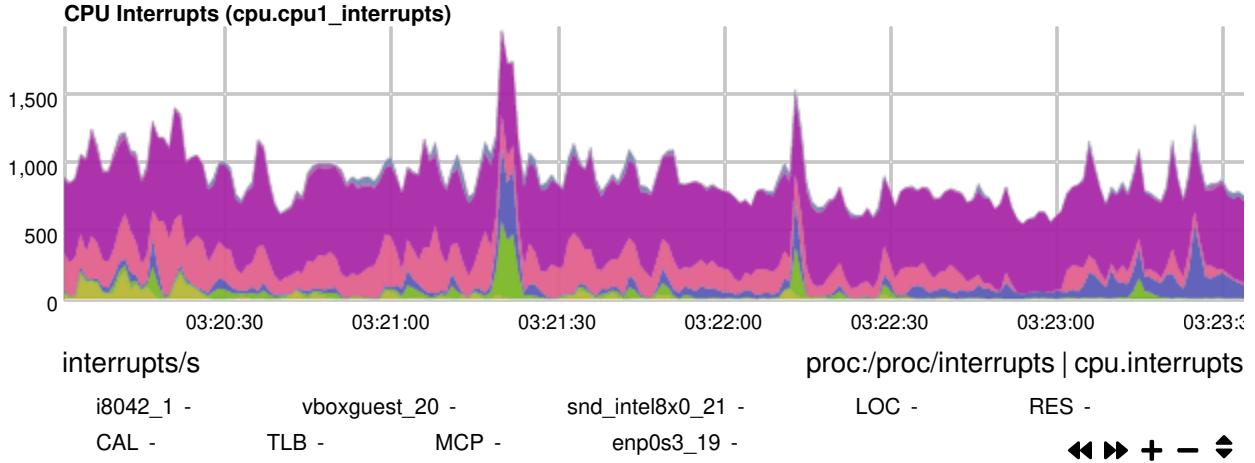
utilization



interrupts

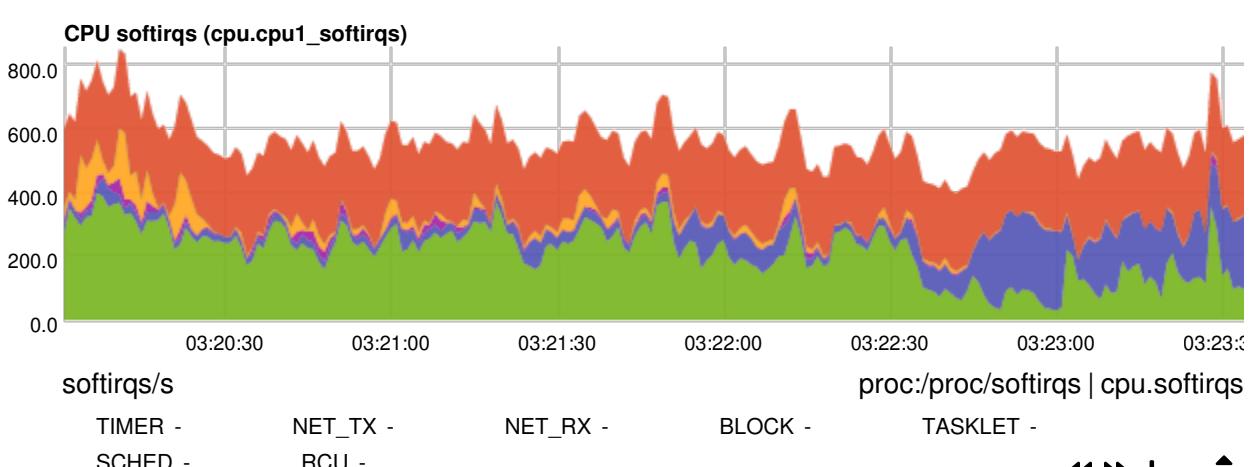
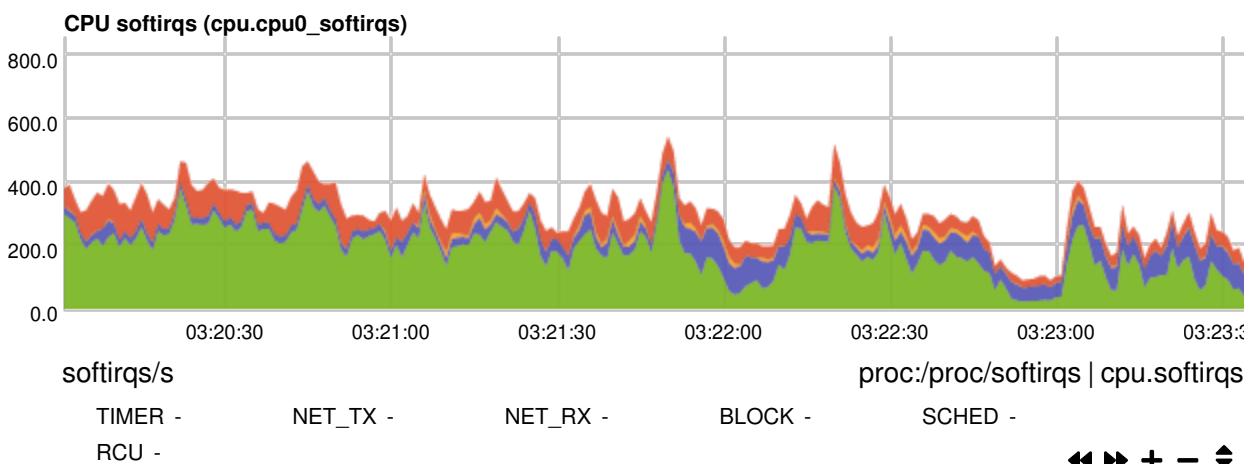
Total number of interrupts per CPU. To see the total number for the system check the interrupts section. The last column in `/proc/interrupts` provides an interrupt description or the device name that registered the handler for that interrupt.





softirqs

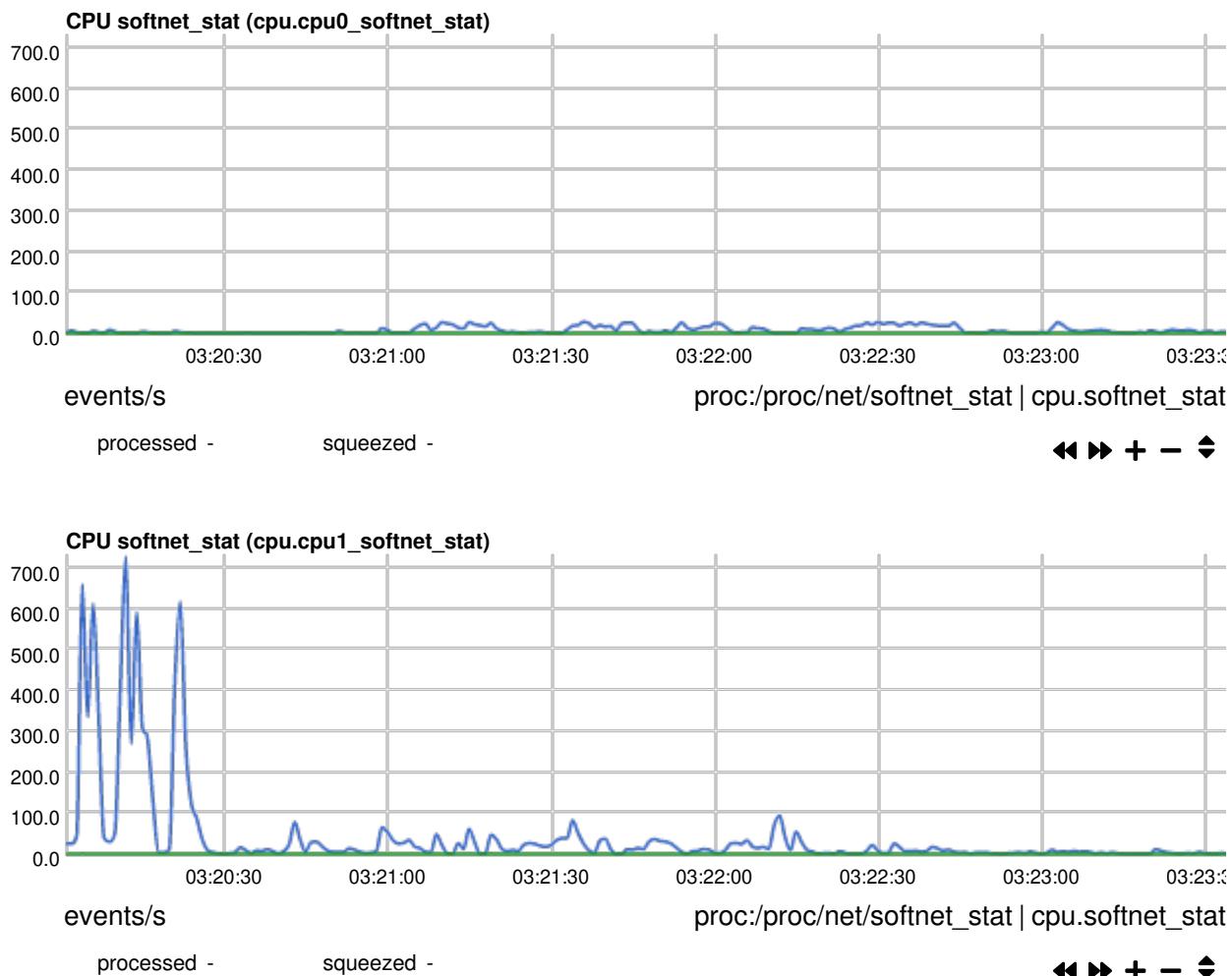
Total number of software interrupts per CPU. To see the total number for the system check the softirqs section.



softnet

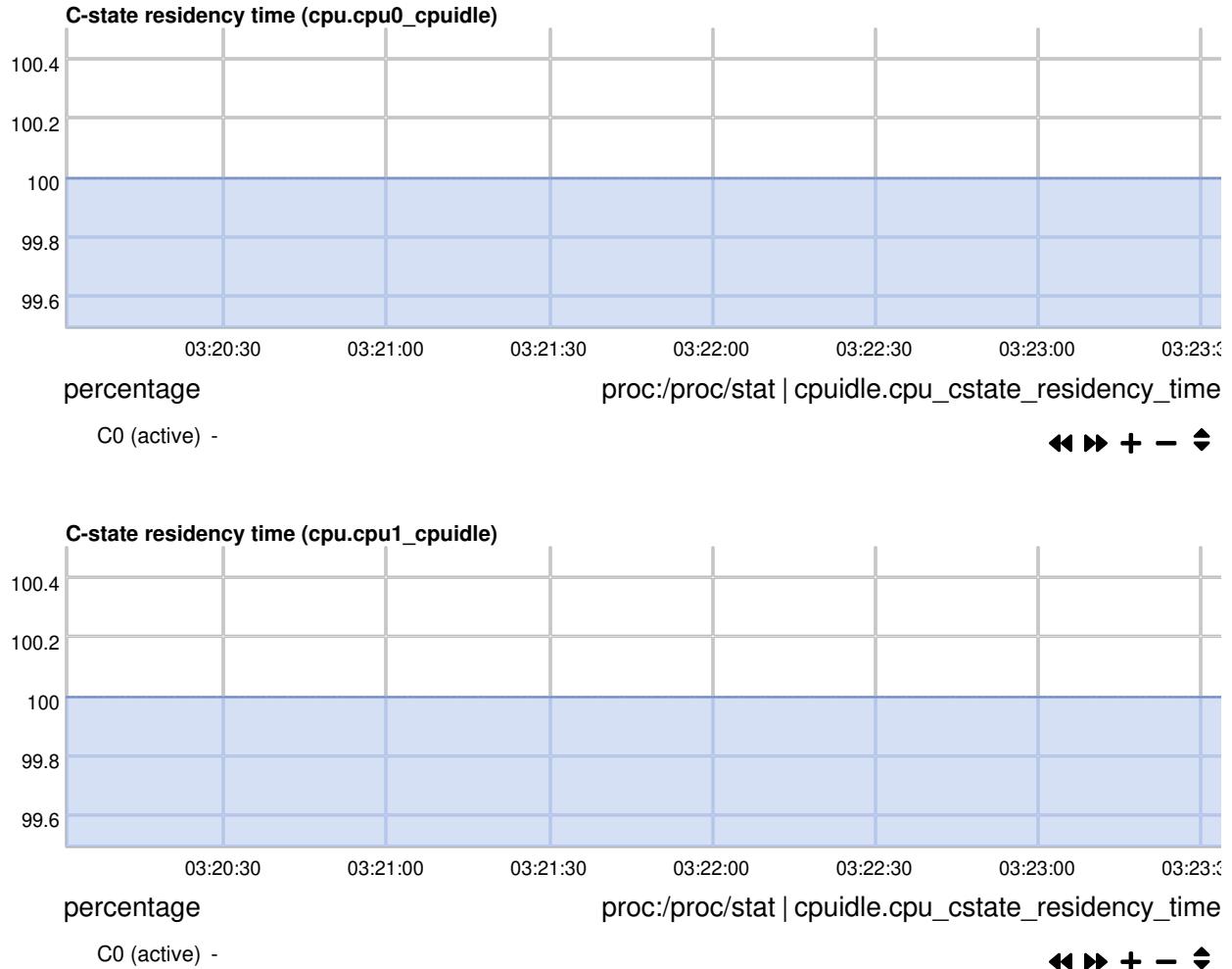
Statistics for CPUs SoftIRQs related to network receive work. Total for all CPU cores can be found at System / softnet statistics. More information about identifying and troubleshooting network driver related issues can be found at Red Hat Enterprise Linux Network Performance Tuning Guide (https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf).

Processed - packets processed. **Dropped** - packets dropped because the network device backlog was full. **Squeezed** - number of times the network device budget was consumed or the time limit was reached, but more work was available. **ReceivedRPS** - number of times this CPU has been woken up to process packets via an Inter-processor Interrupt. **FlowLimitCount** - number of times the flow limit has been reached (flow limiting is an optional Receive Packet Steering feature).



cpuidle

Idle States (C-states) (https://en.wikipedia.org/wiki/Advanced_Configuration_and_Power_Interface#Processor_states) are used to save power when the processor is idle.



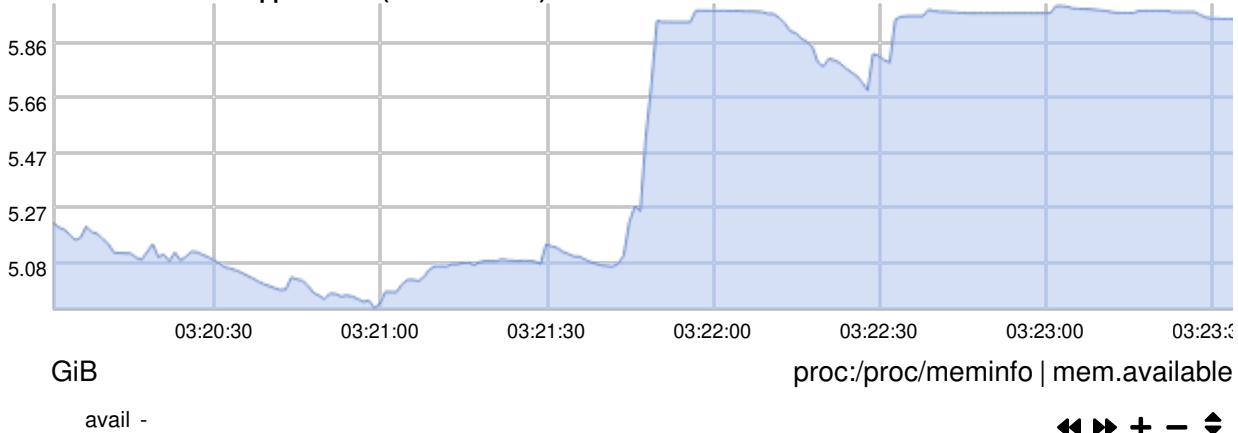
Memory

Detailed information about the memory management of the system.

system

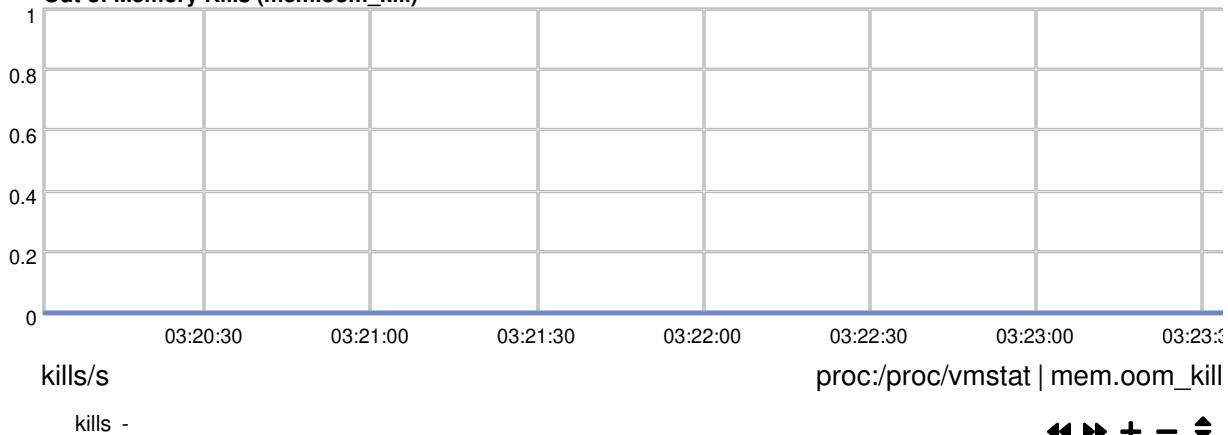
Available Memory is estimated by the kernel, as the amount of RAM that can be used by userspace processes, without causing swapping.

Available RAM for applications (mem.available)

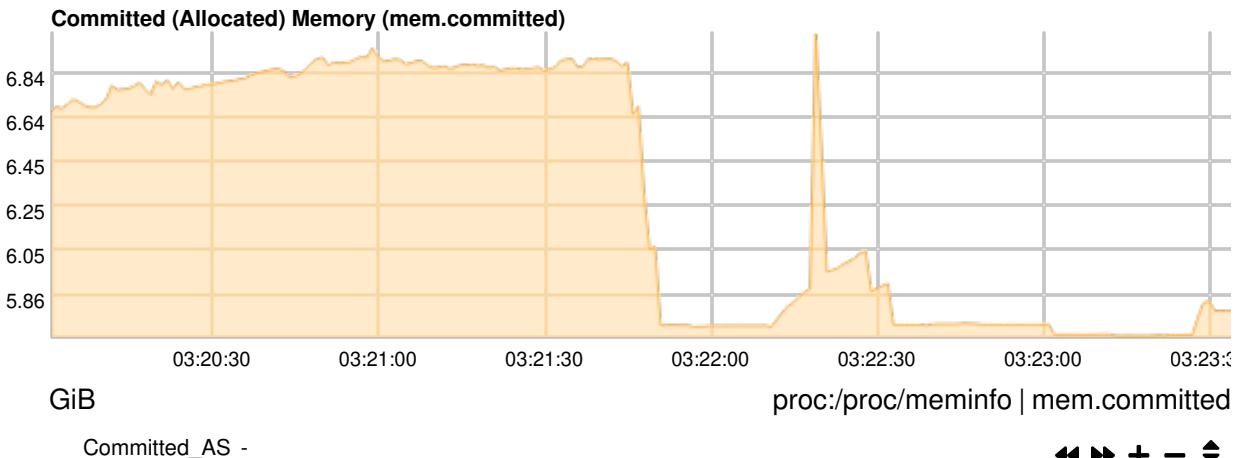


The number of processes killed by Out of Memory (https://en.wikipedia.org/wiki/Out_of_memory) Killer. The kernel's OOM killer is summoned when the system runs short of free memory and is unable to proceed without killing one or more processes. It tries to pick the process whose demise will free the most memory while causing the least misery for users of the system. This counter also includes processes within containers that have exceeded the memory limit.

Out of Memory Kills (mem.oom_kill)

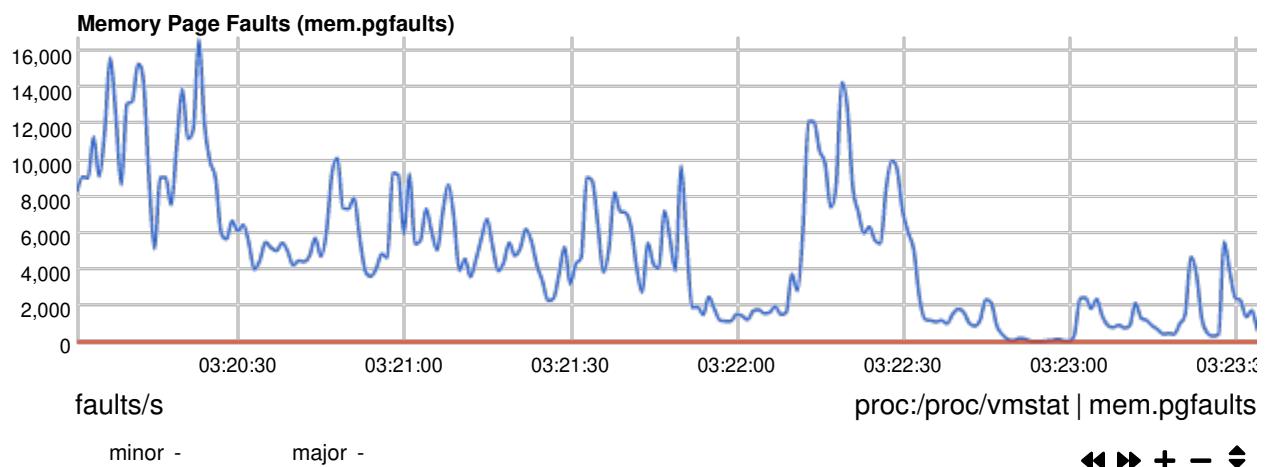


Committed Memory, is the sum of all memory which has been allocated by processes.



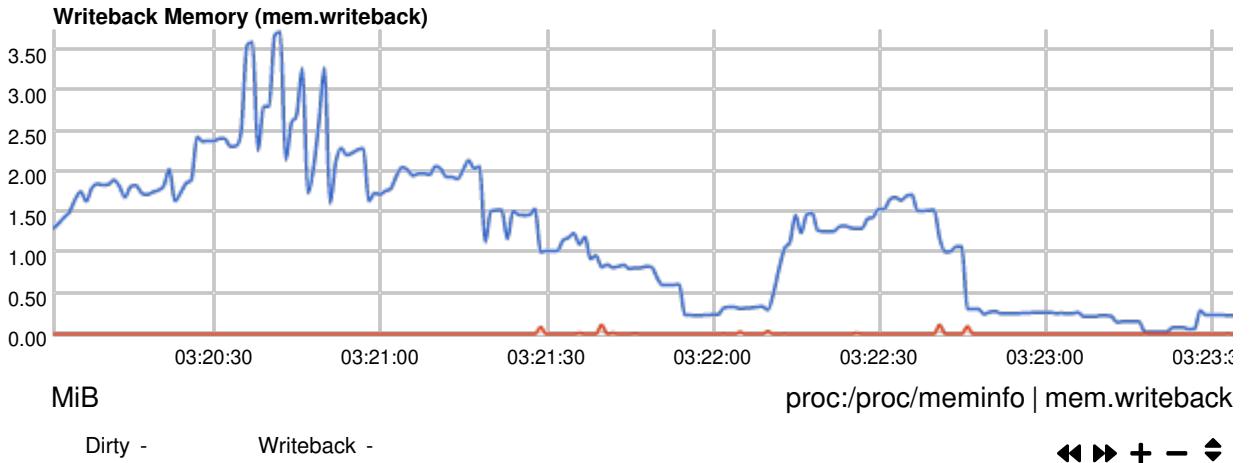
A page fault (https://en.wikipedia.org/wiki/Page_fault) is a type of interrupt, called trap, raised by computer hardware when a running program accesses a memory page that is mapped into the virtual address space, but not actually loaded into main memory.

Minor - 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. **Major** - generated when the system needs to load the memory page from disk or swap memory.



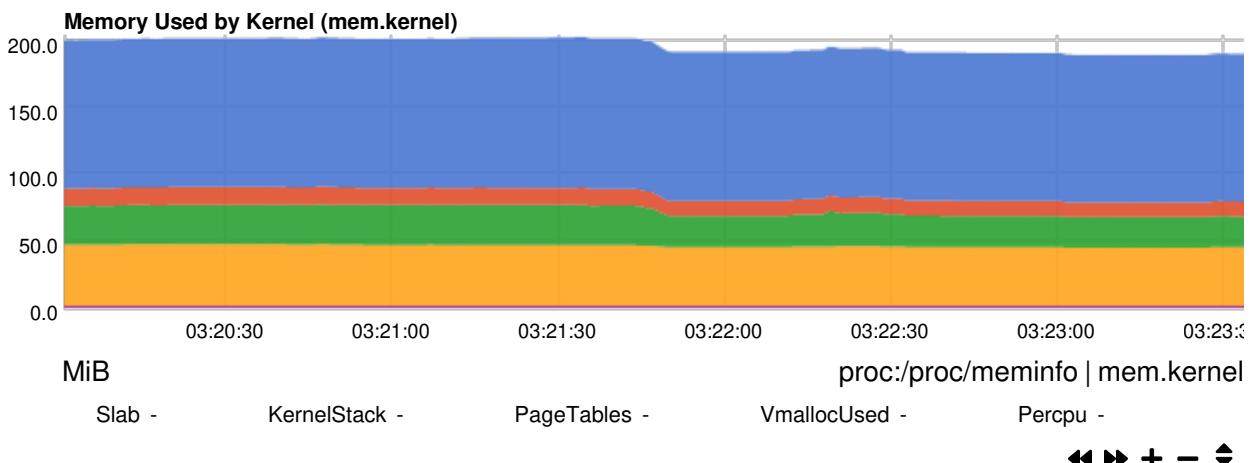
kernel

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.

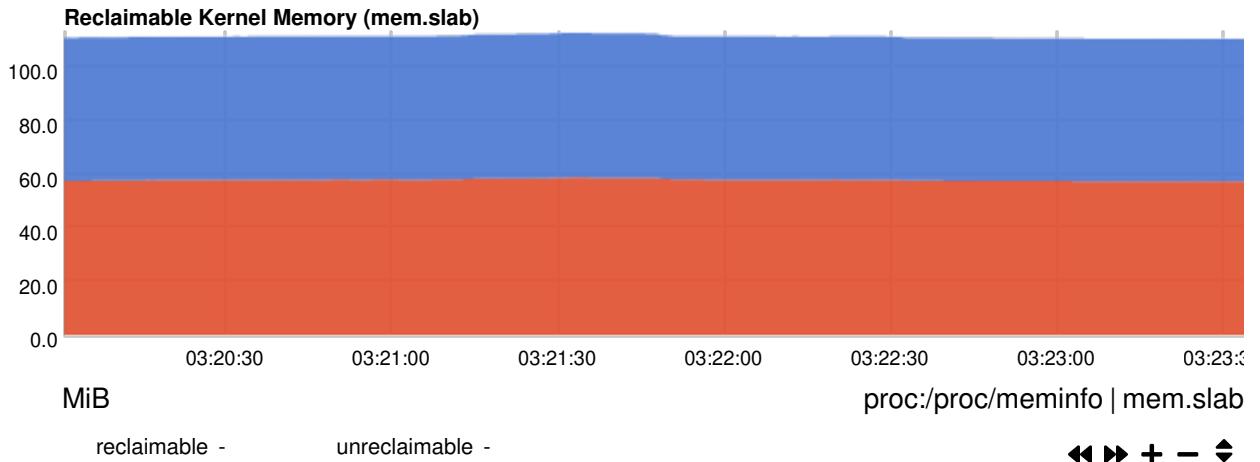
Slab - used by the kernel to cache data structures for its own use. **KernelStack** - allocated for each task done by the kernel. **PageTables** - dedicated to the lowest level of page tables (A page table is used to turn a virtual address into a physical memory address). **VmallocUsed** - being used as virtual address space. **Percpu** - allocated to the per-CPU allocator used to back per-CPU allocations (excludes the cost of metadata). When you create a per-CPU variable, each processor on the system gets its own copy of that variable.



slab

Slab memory (https://en.wikipedia.org/wiki/Slab_allocation) statistics.

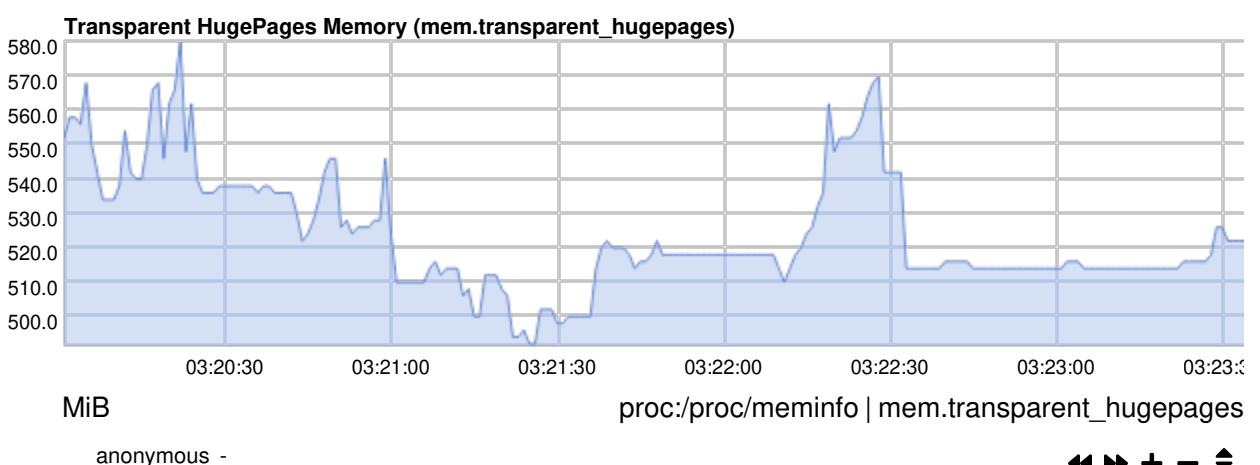
Reclaimable - amount of memory which the kernel can reuse. **Unreclaimable** - can not be reused even when the kernel is lacking memory.



hugepages

Hugepages is a feature that allows the kernel to utilize the multiple page size capabilities of modern hardware architectures. The kernel creates multiple pages of virtual memory, mapped from both physical RAM and swap. There is a mechanism in the CPU architecture called "Translation Lookaside Buffers" (TLB) to manage the mapping of virtual memory pages to actual physical memory addresses. The TLB is a limited hardware resource, so utilizing a large amount of physical memory with the default page size consumes the TLB and adds processing overhead. By utilizing Huge Pages, the kernel is able to create pages of much larger sizes, each page consuming a single resource in the TLB. Huge Pages are pinned to physical RAM and cannot be swapped/paged out.

Transparent HugePages (THP) is backing virtual memory with huge pages, supporting automatic promotion and demotion of page sizes. It works for all applications for anonymous memory mappings and tmpfs/shmem.

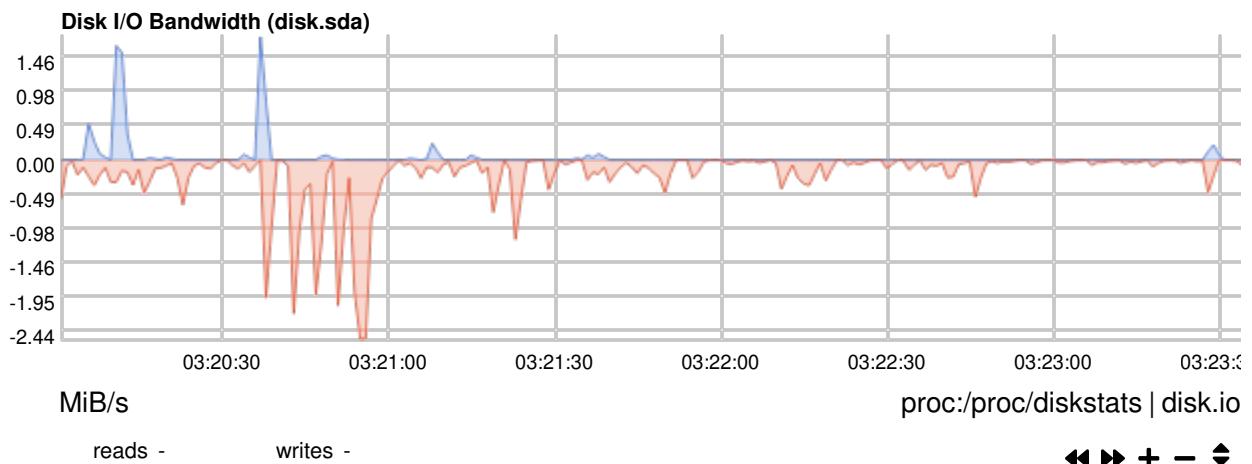


Disks

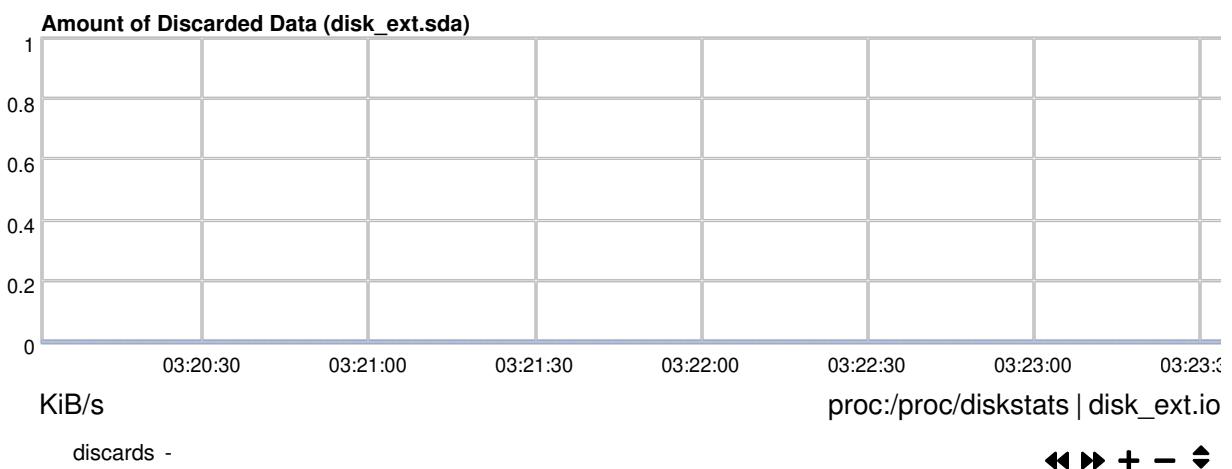
Charts with performance information for all the system disks. Special care has been given to present disk performance metrics in a way compatible with `iostat -x`. netdata by default prevents rendering performance charts for individual partitions and unmounted virtual disks. Disabled charts can still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

sda

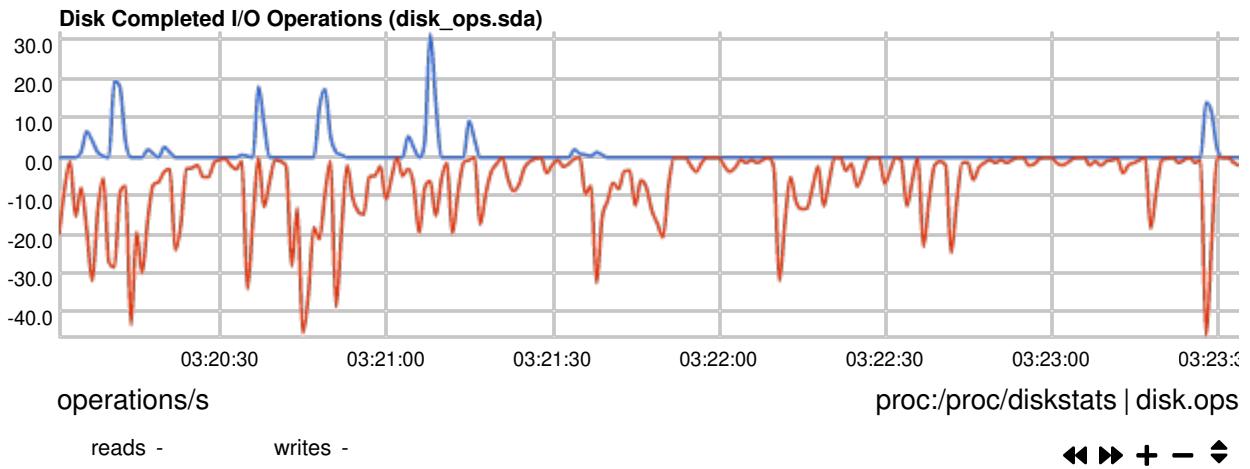
The amount of data transferred to and from disk.



The amount of discarded data that are no longer in use by a mounted file system.



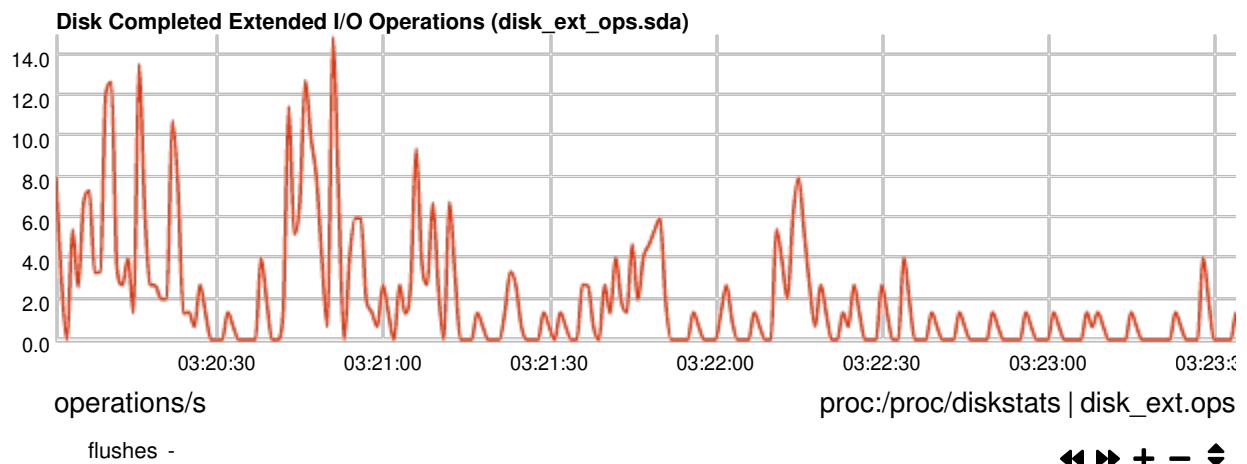
Completed disk I/O operations. Keep in mind the number of operations requested might be higher, since the system is able to merge adjacent to each other (see merged operations chart).



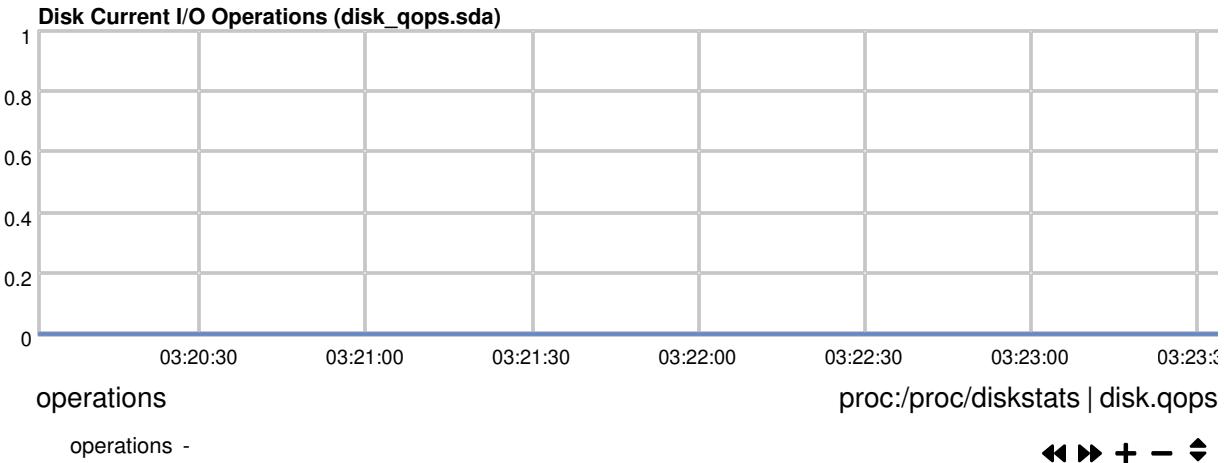
The number (after merges) of completed discard/flush requests.

Discard commands inform disks which blocks of data are no longer considered to be in use and therefore can be erased internally. They are useful for solid-state drivers (SSDs) and thinly-provisioned storage. Discarding/trimming enables the SSD to handle garbage collection more efficiently, which would otherwise slow future write operations to the involved blocks down.

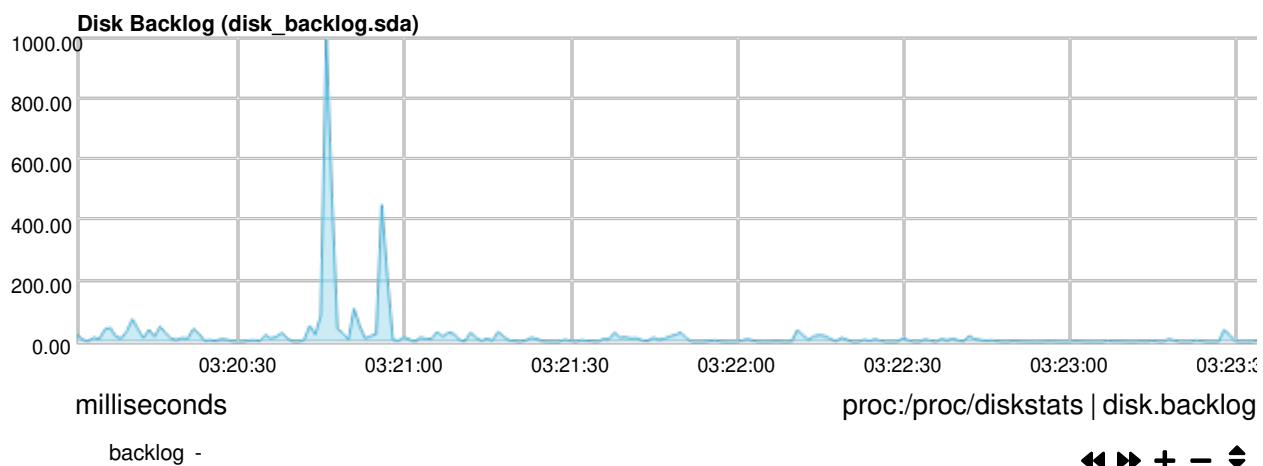
Flush operations transfer all modified in-core data (i.e., modified buffer cache pages) to the disk device so that all changed information can be retrieved even if the system crashes or is rebooted. Flush requests are executed by disks. Flush requests are not tracked for partitions. Before being merged, flush operations are counted as writes.



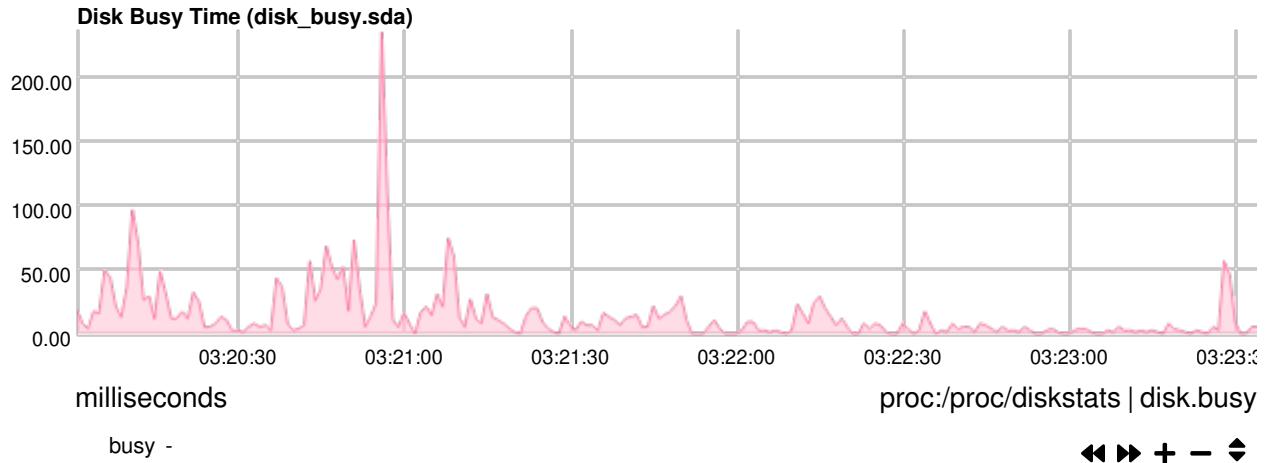
I/O operations currently in progress. This metric is a snapshot - it is not an average over the last interval.



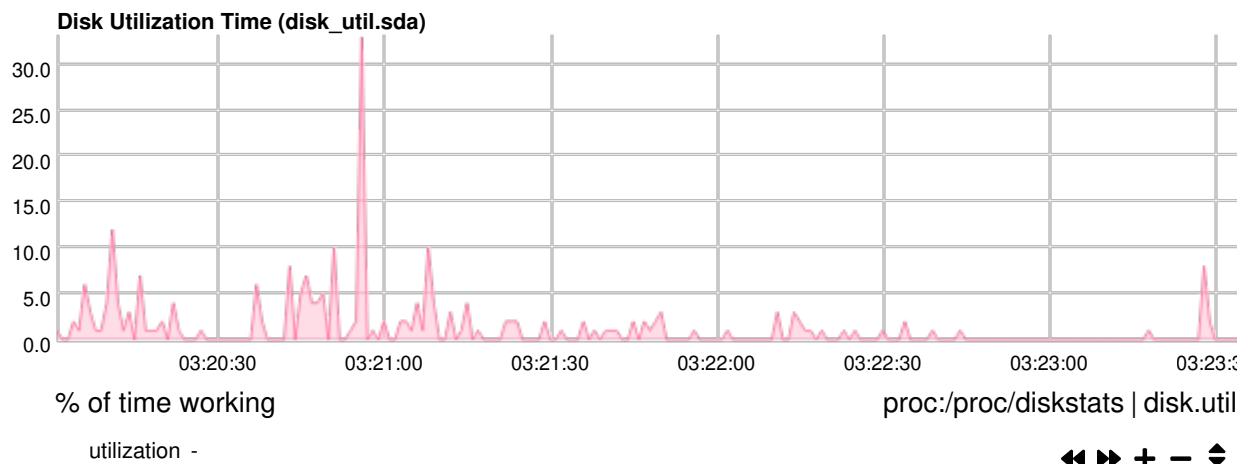
Backlog is an indication of the duration of pending disk operations. On every I/O event the system is multiplying the time spent doing I/O since the last update of this field with the number of pending operations. While not accurate, this metric can provide an indication of the expected completion time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



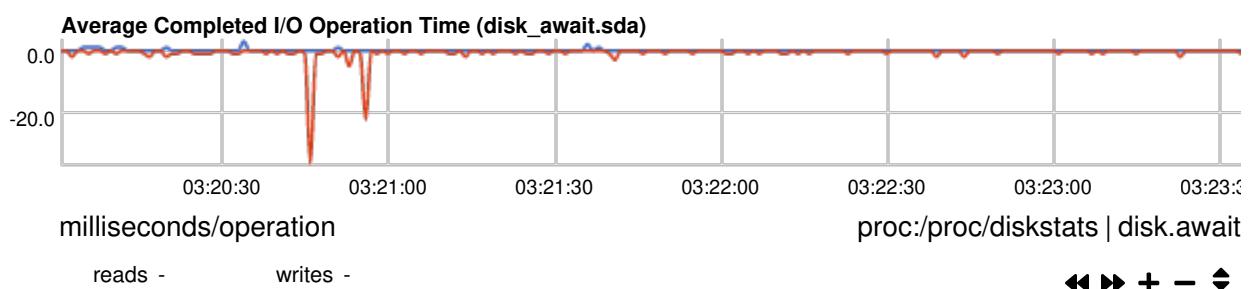
Disk Busy Time measures the amount of time the disk was busy with something.



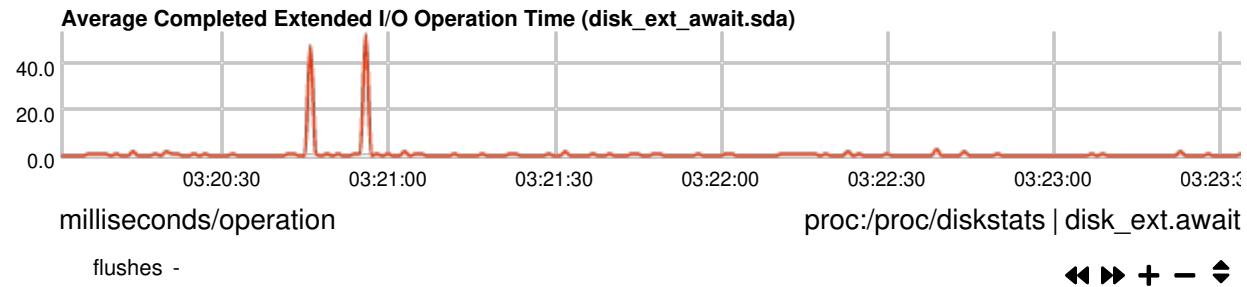
Disk Utilization measures the amount of time the disk was busy with something. This is not related to its performance. 100% means that the system always had an outstanding operation on the disk. Keep in mind that depending on the underlying technology of the disk, 100% here may or may not be an indication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



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



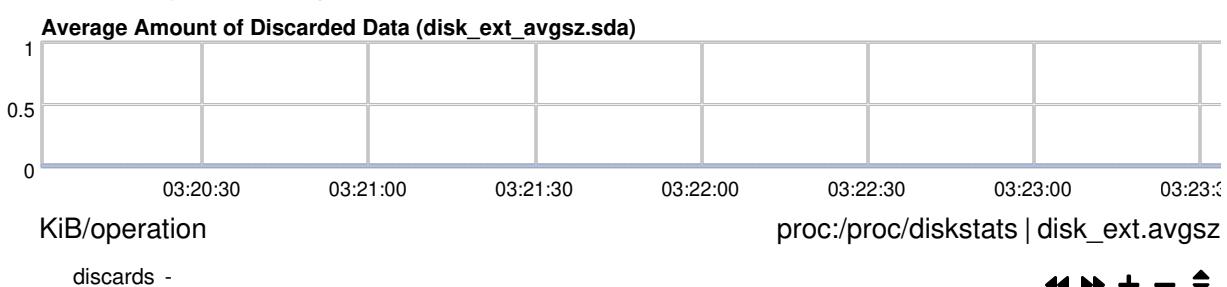
The average time for discard/flush requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.



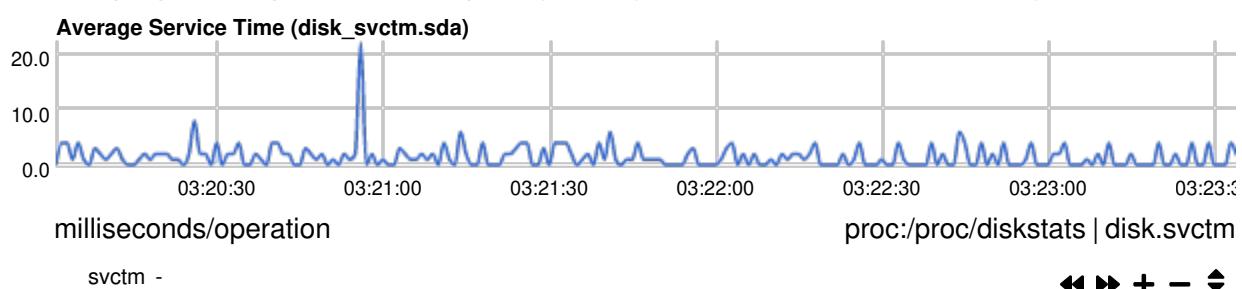
The average I/O operation size.



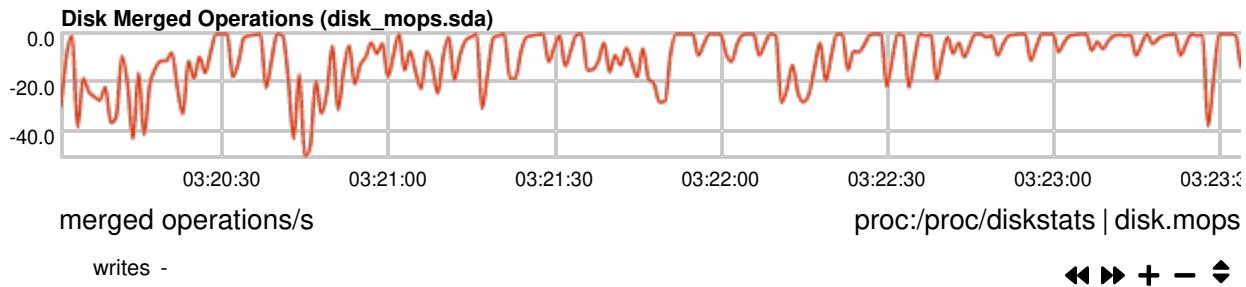
The average discard operation size.



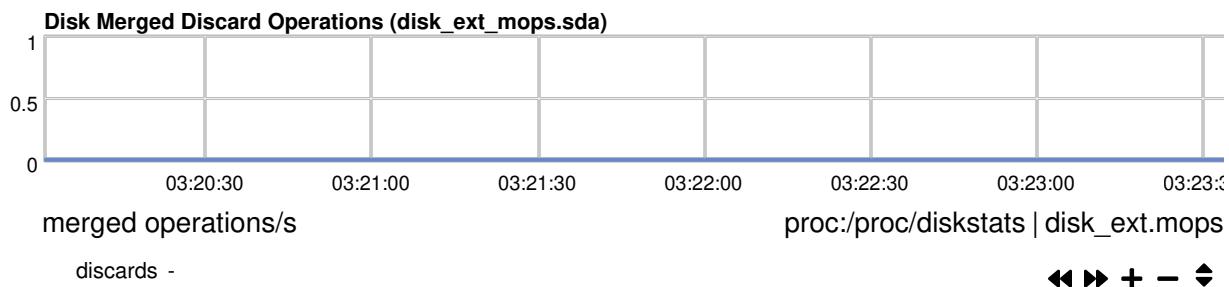
The average service time for completed I/O operations. This metric is calculated using the total busy time of the disk and the number of completed operations. If the disk is able to execute multiple parallel operations the reporting average service time will be misleading.



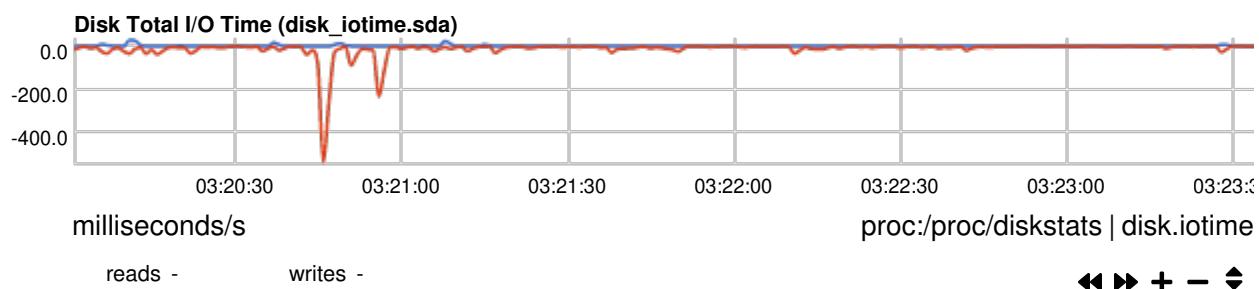
The number of merged disk operations. The system is able to merge adjacent I/O operations, for example two 4KB reads can become one 8KB read before given to disk.



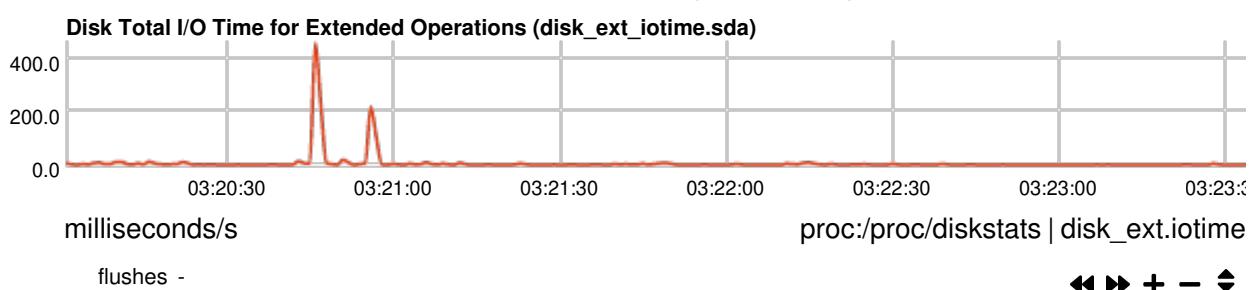
The number of merged discard disk operations. Discard operations which are adjacent to each other may be merged for efficiency.



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.

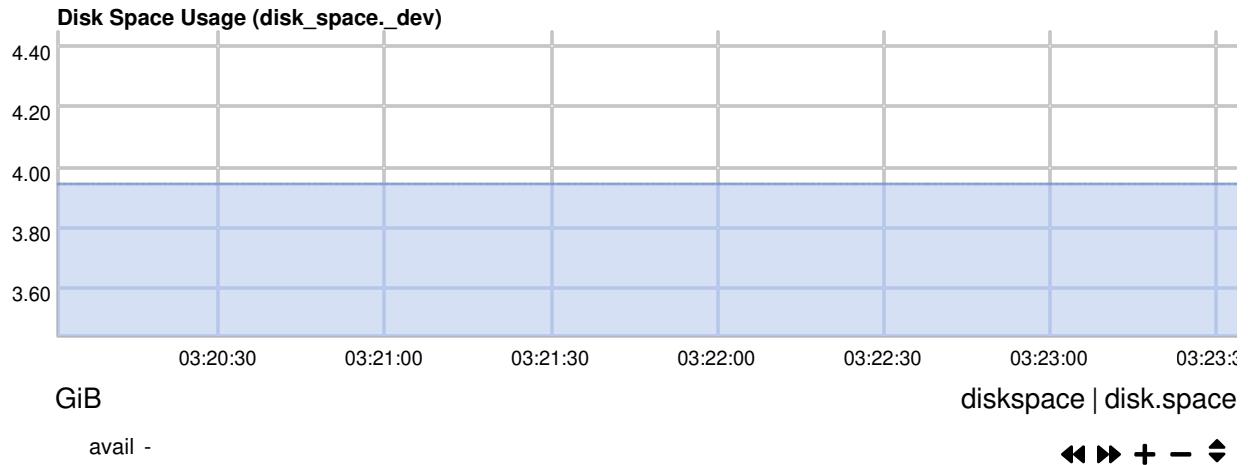


The sum of the duration of all completed discard/flush operations. This number can exceed the interval if the disk is able to execute discard/flush operations in parallel.

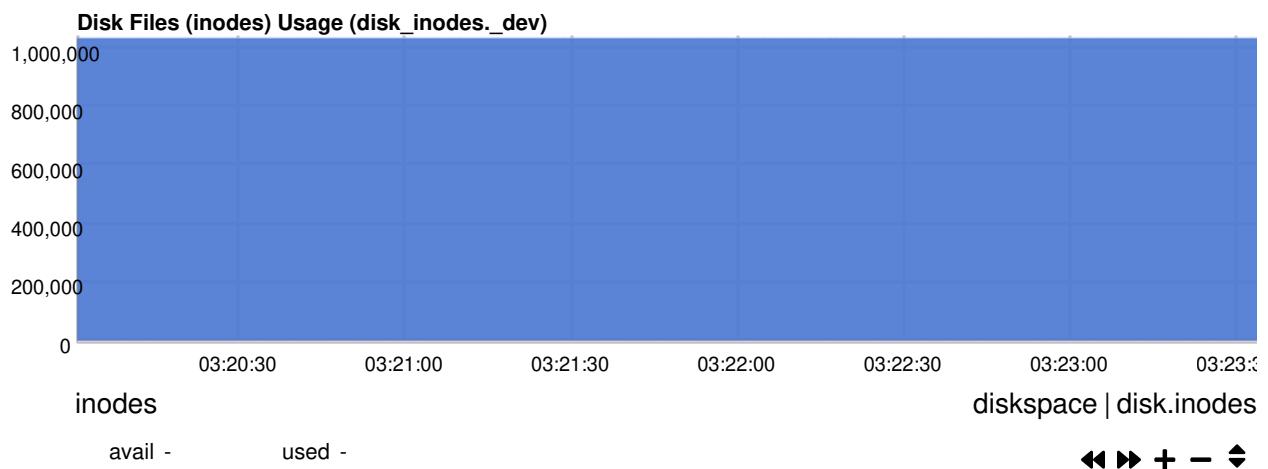


/dev

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

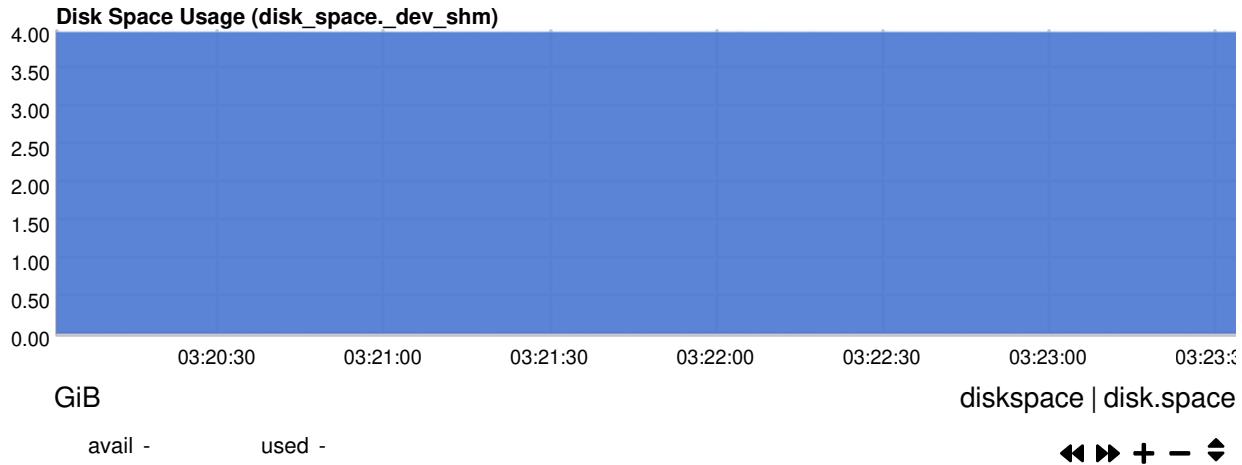


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

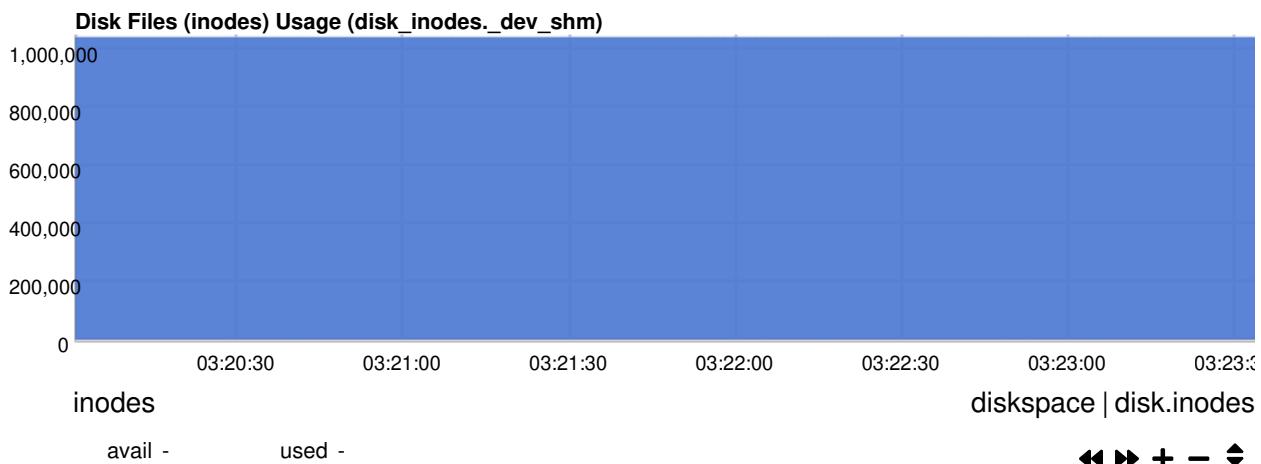


/dev/shm

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

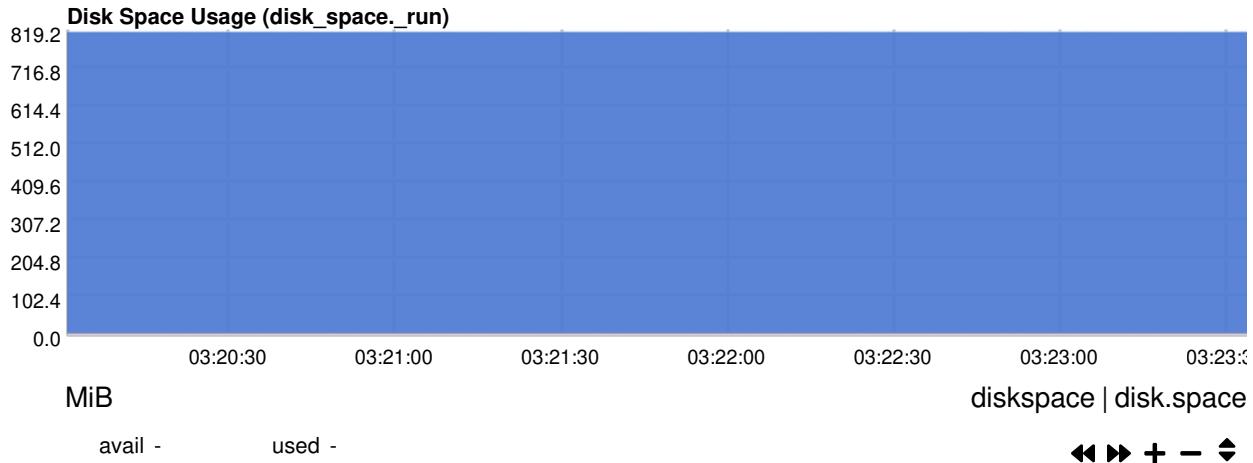


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

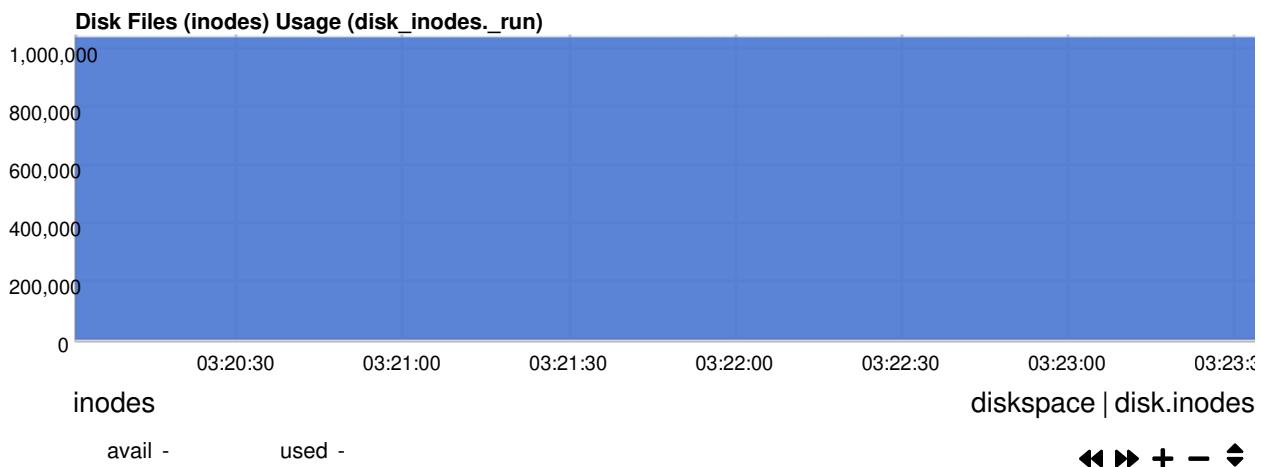


/run

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

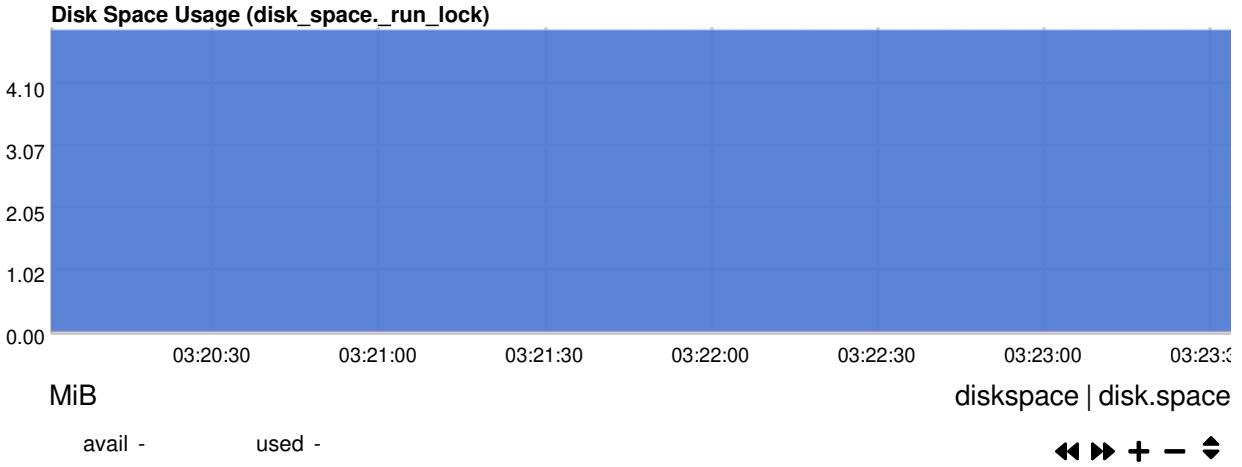


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

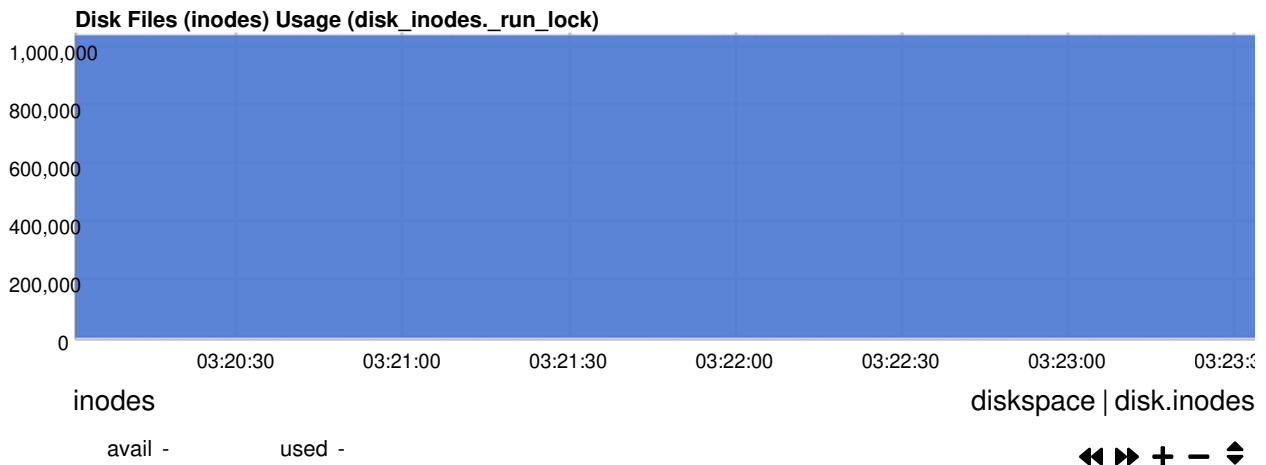


/run/lock

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

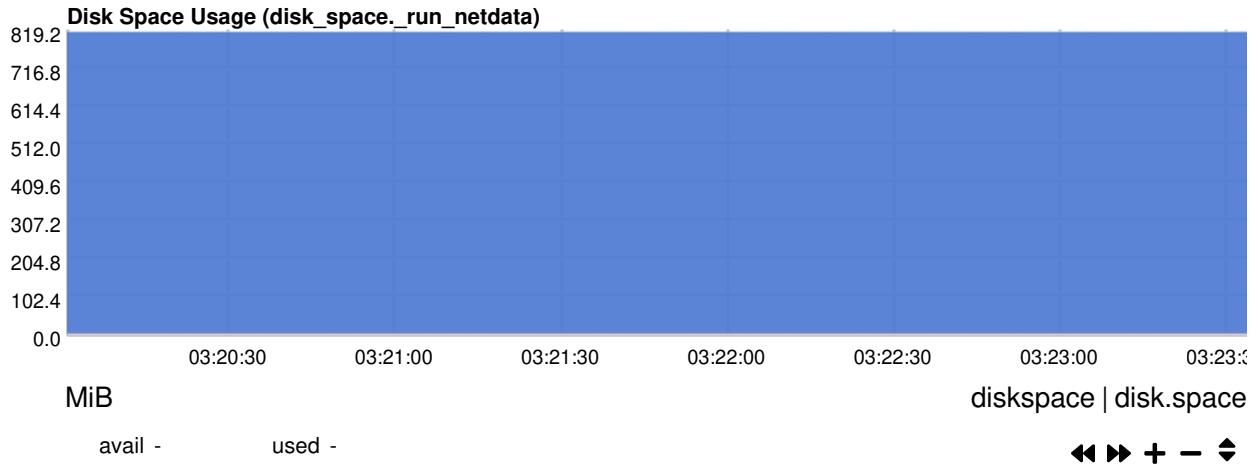


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

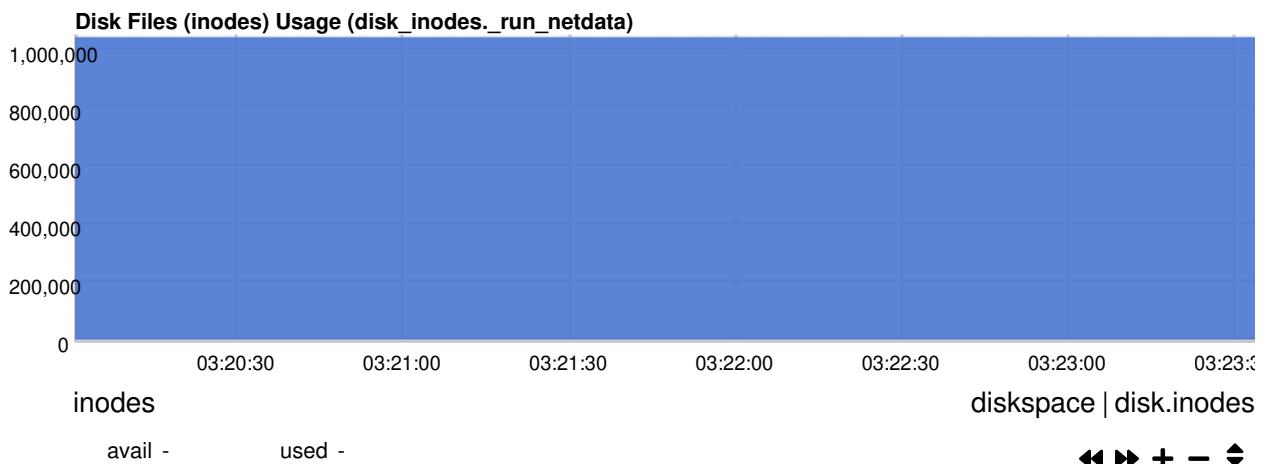


/run/netdata

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

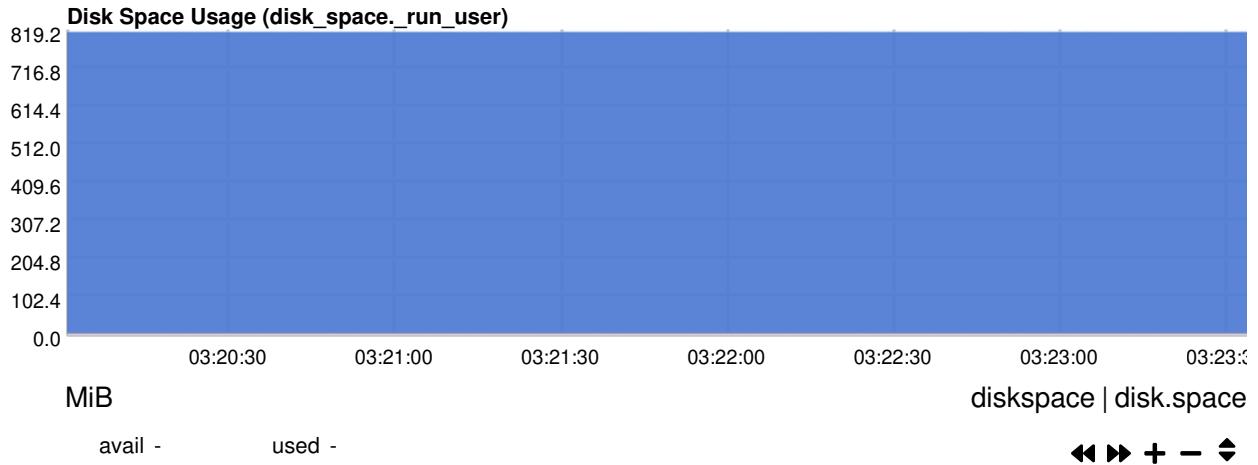


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

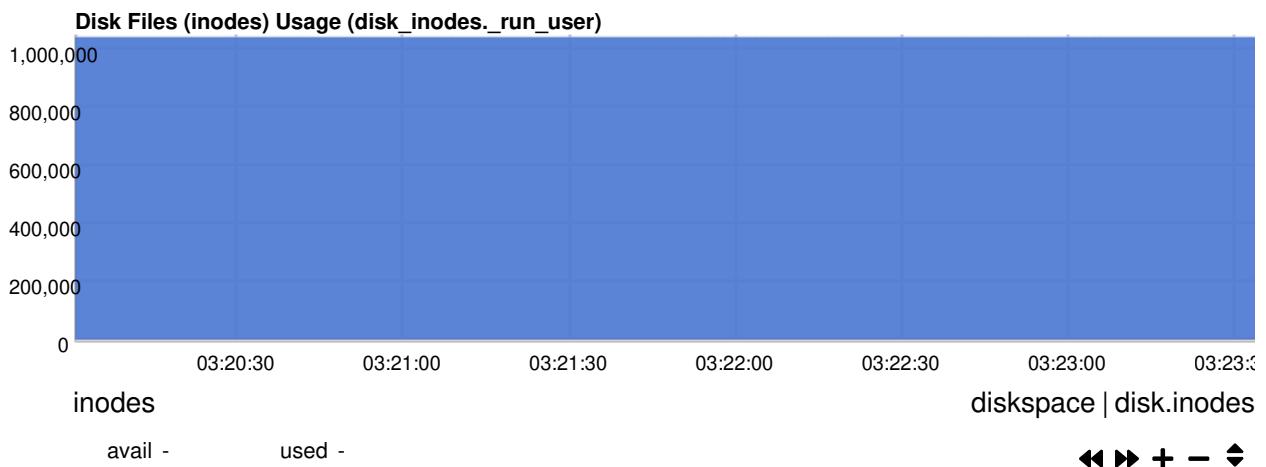


/run/user

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

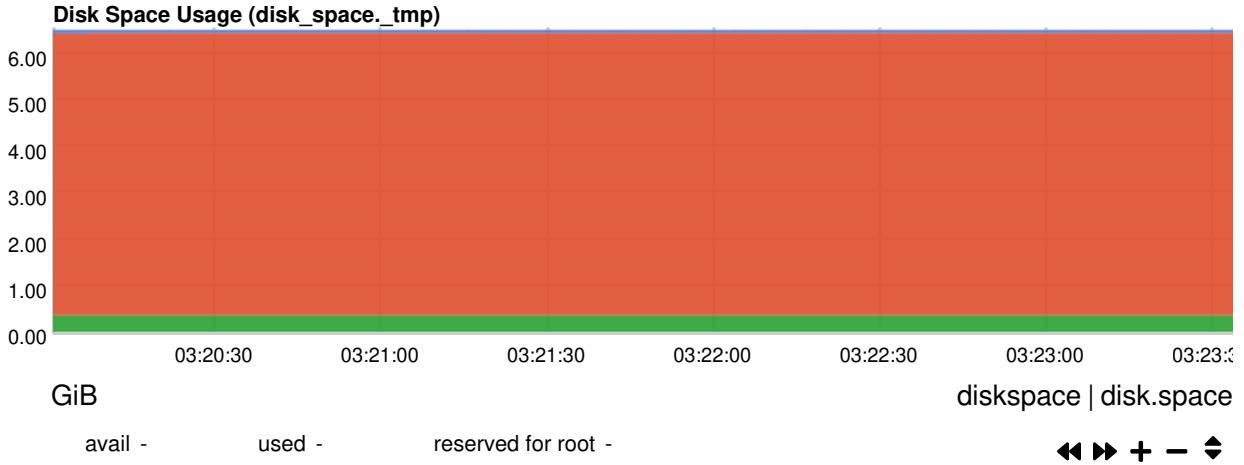


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

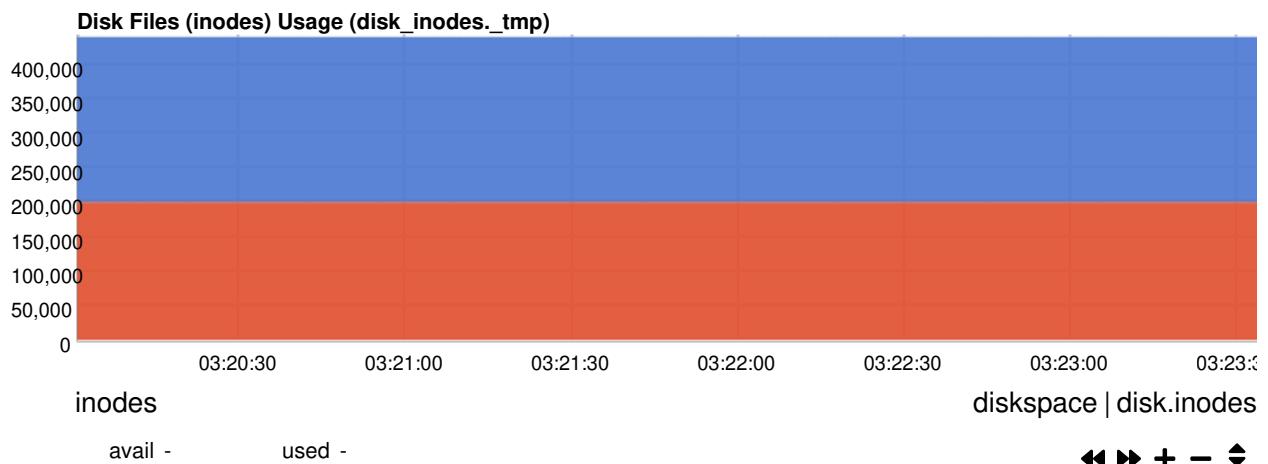


/tmp

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

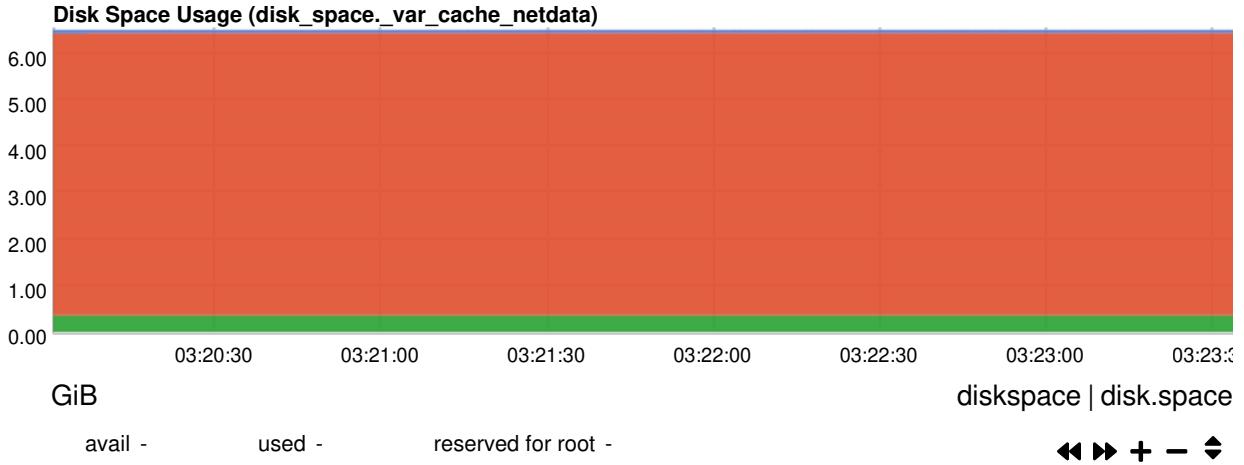


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

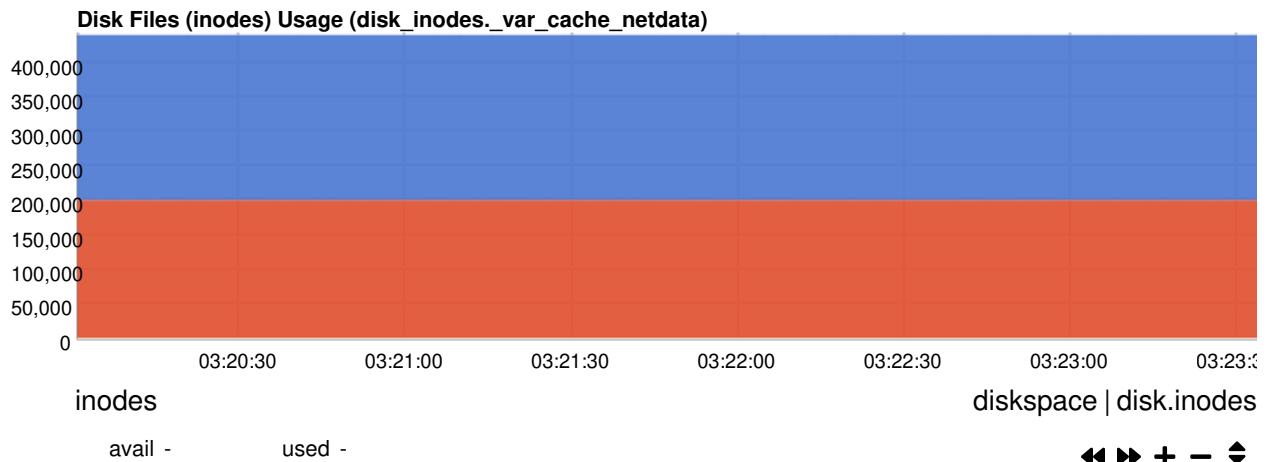


/var/cache/netdata

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

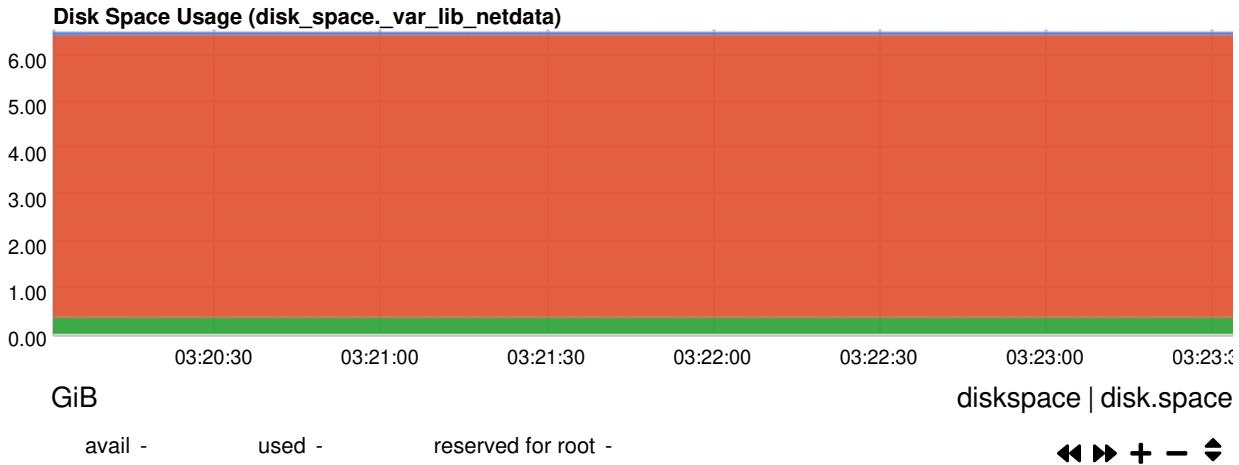


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

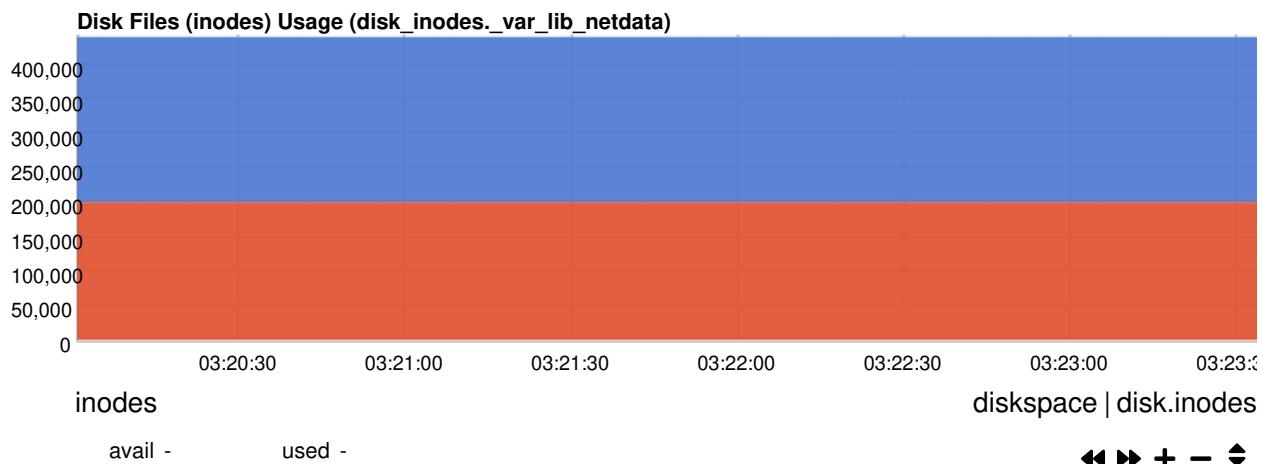


/var/lib/netdata

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

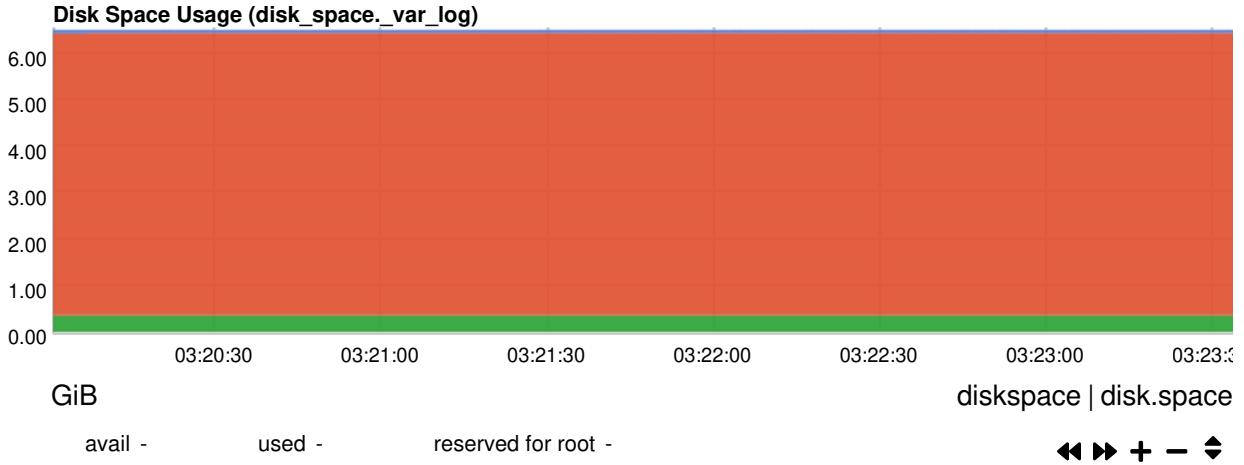


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

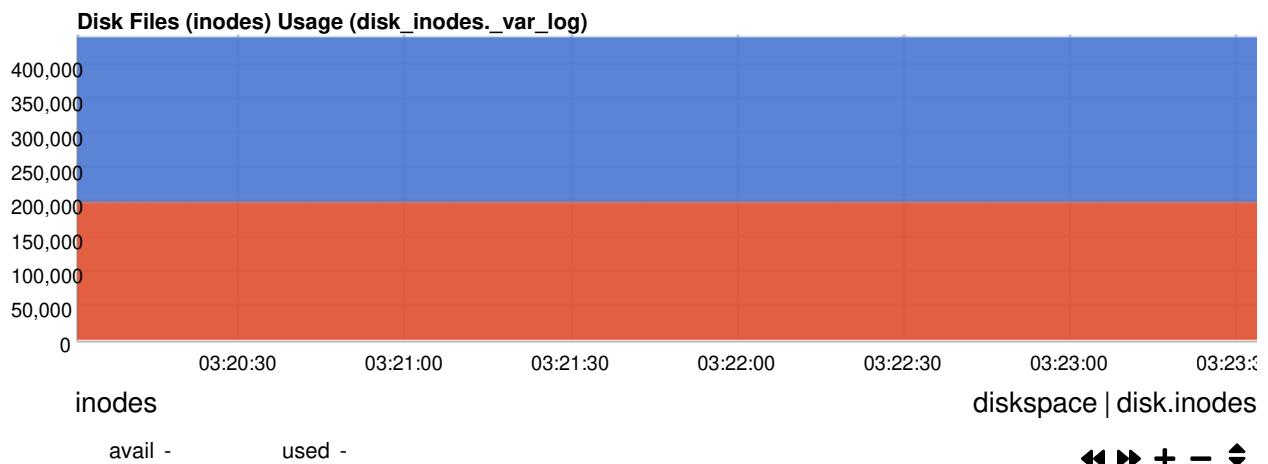


/var/log

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

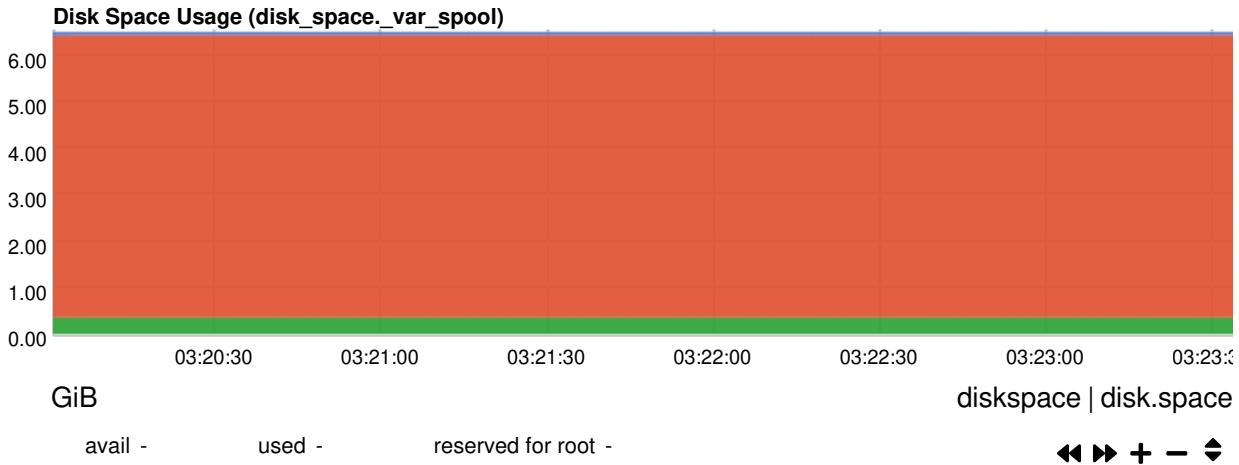


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

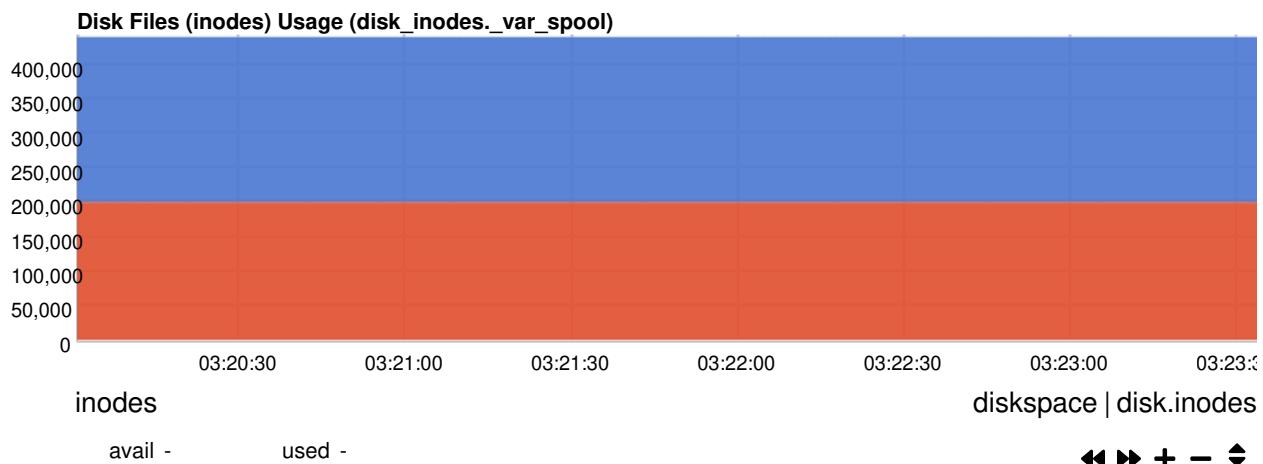


/var/spool

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

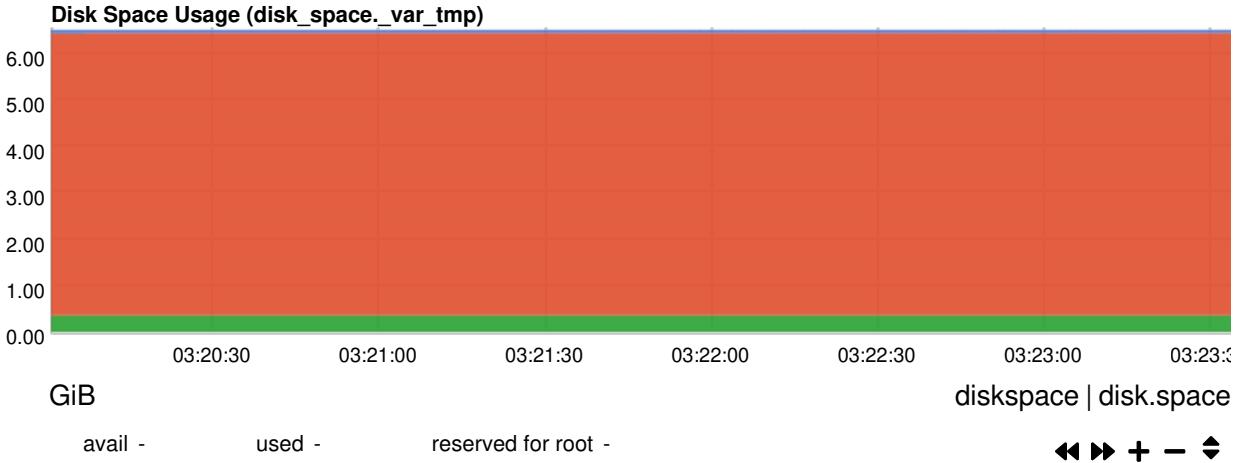


Inodes (or index nodes) are filesystem objects (e.g. files and directories). On many types of file system implementations, the maximum number of inodes is fixed at filesystem creation, limiting the maximum number of files the filesystem can hold. It is possible for a device to run out of inodes. When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

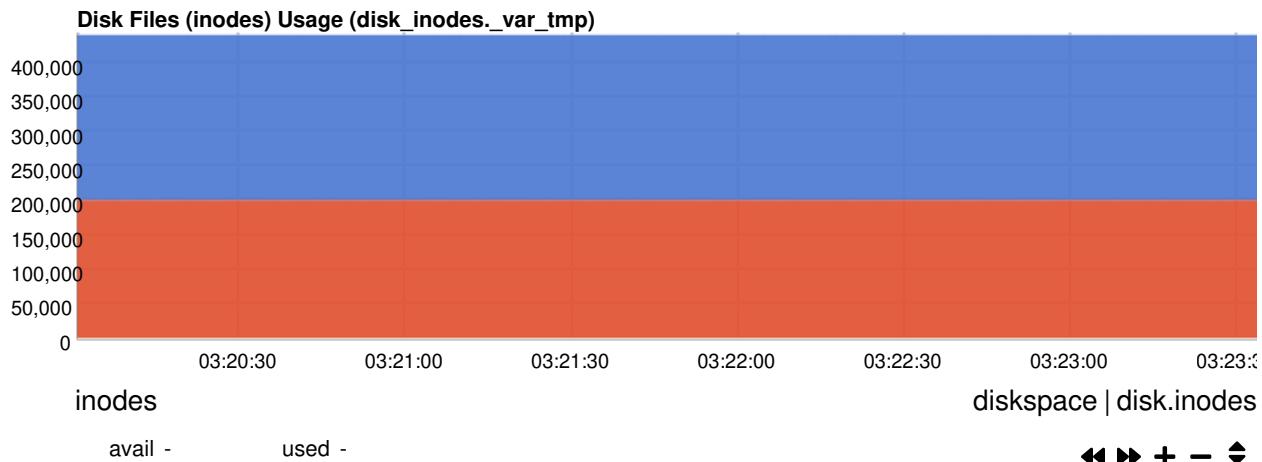


/var/tmp

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.



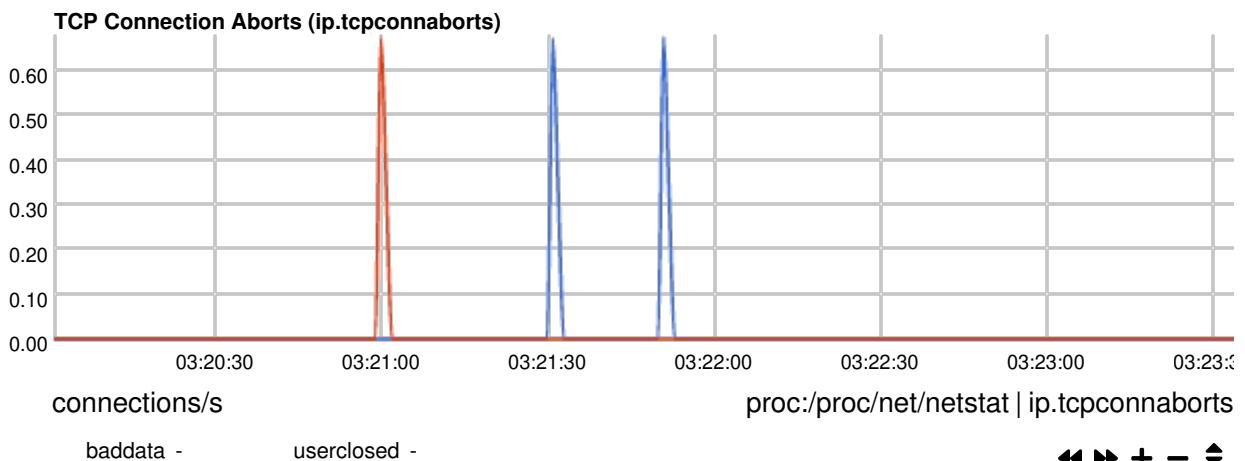
Networking Stack

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

tcp

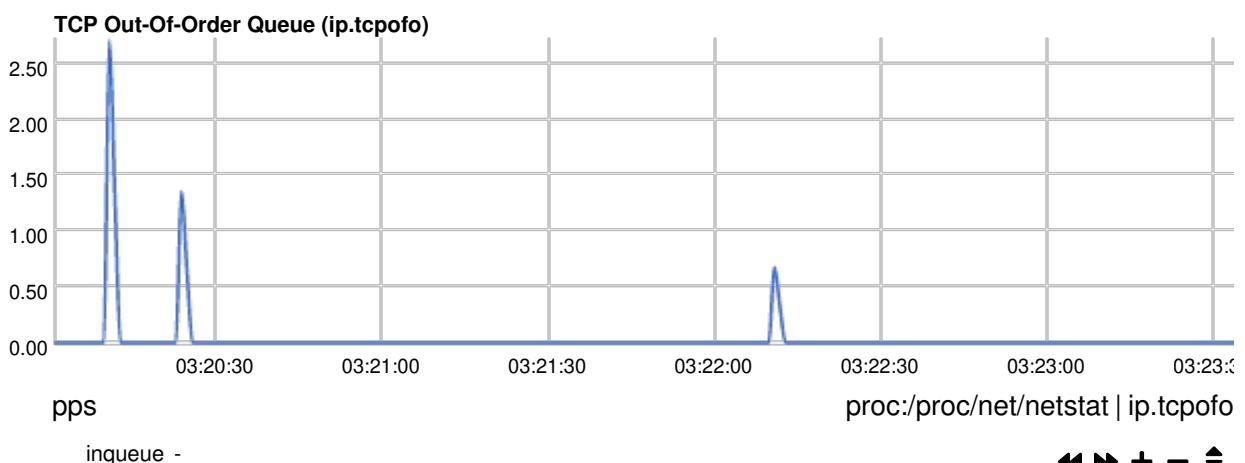
TCP connection aborts.

BadData - 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** - happens when the kernel receives data on an already closed connection and responds with RST. **NoMemory** - 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** - happens when a connection times out. **Linger** - happens when the kernel killed a socket that was already closed by the application and lingered around for long enough. **Failed** - happens when the kernel attempted to send an RST but failed because there was no memory available.



TCP maintains an out-of-order queue to keep the out-of-order packets in the TCP communication.

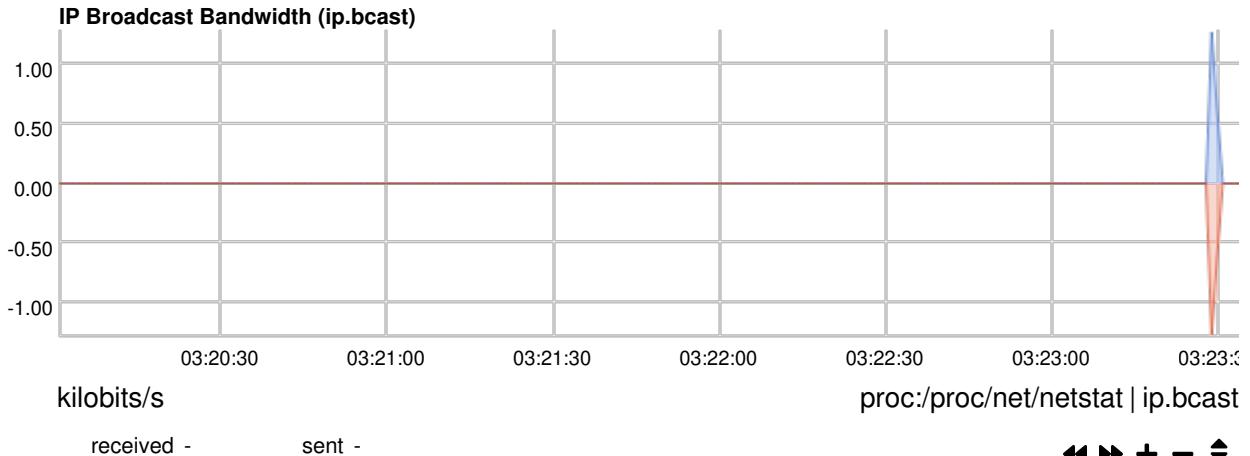
InQueue - the TCP layer receives an out-of-order packet and has enough memory to queue it. **Dropped** - the TCP layer receives an out-of-order packet but does not have enough memory, so drops it. **Merged** - the received out-of-order packet has an overlay with the previous packet. The overlay part will be dropped. All these packets will also be counted into **InQueue**. **Pruned** - packets dropped from out-of-order queue because of socket buffer overrun.



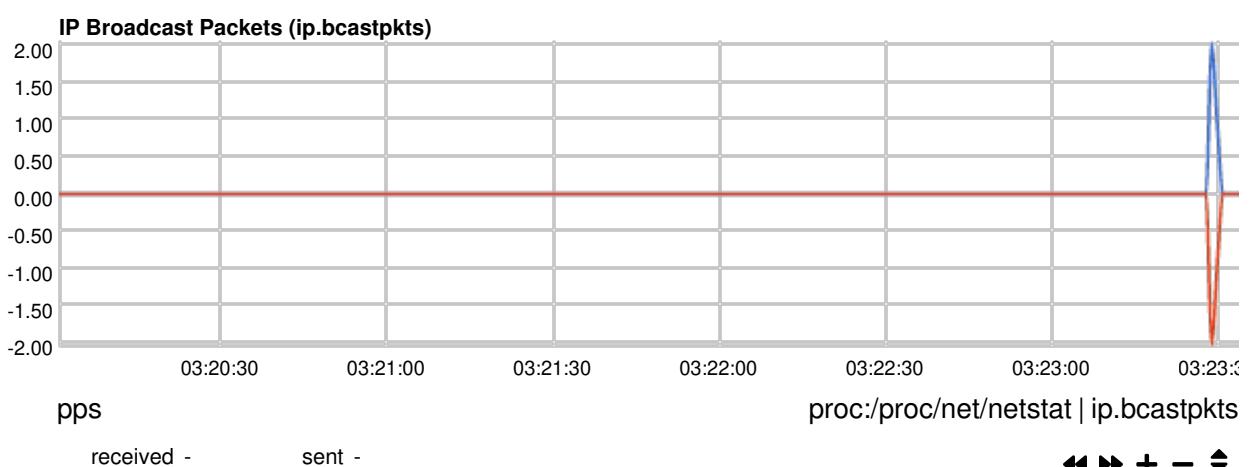
broadcast

In computer networking, broadcasting ([https://en.wikipedia.org/wiki/Broadcasting_\(networking\)](https://en.wikipedia.org/wiki/Broadcasting_(networking))) refers to transmitting a packet that will be received by every device on the network. In practice, the scope of the broadcast is limited to a broadcast domain.

Total broadcast traffic in the system.

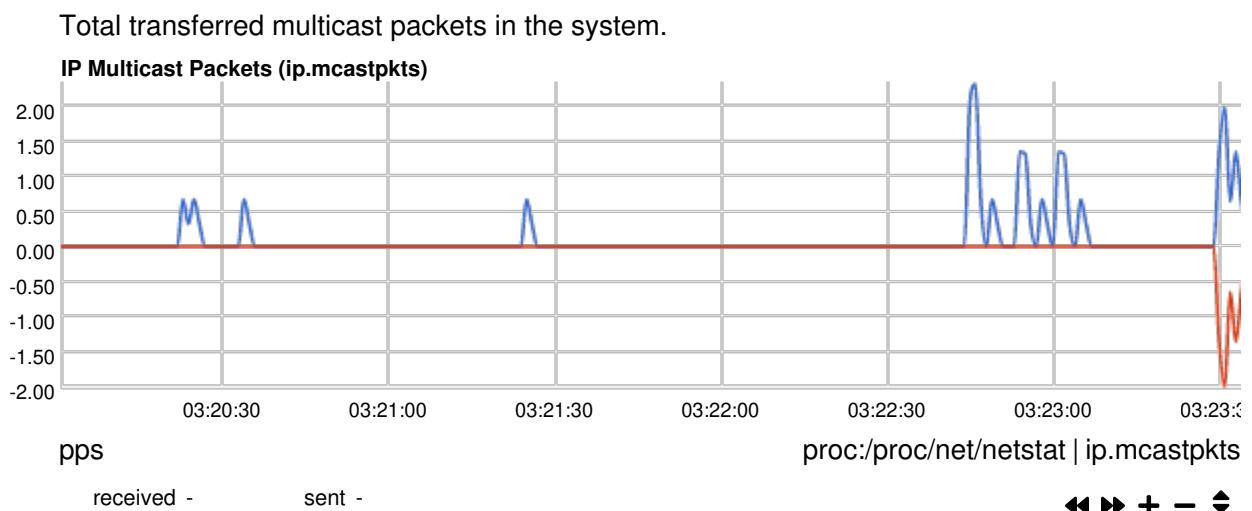
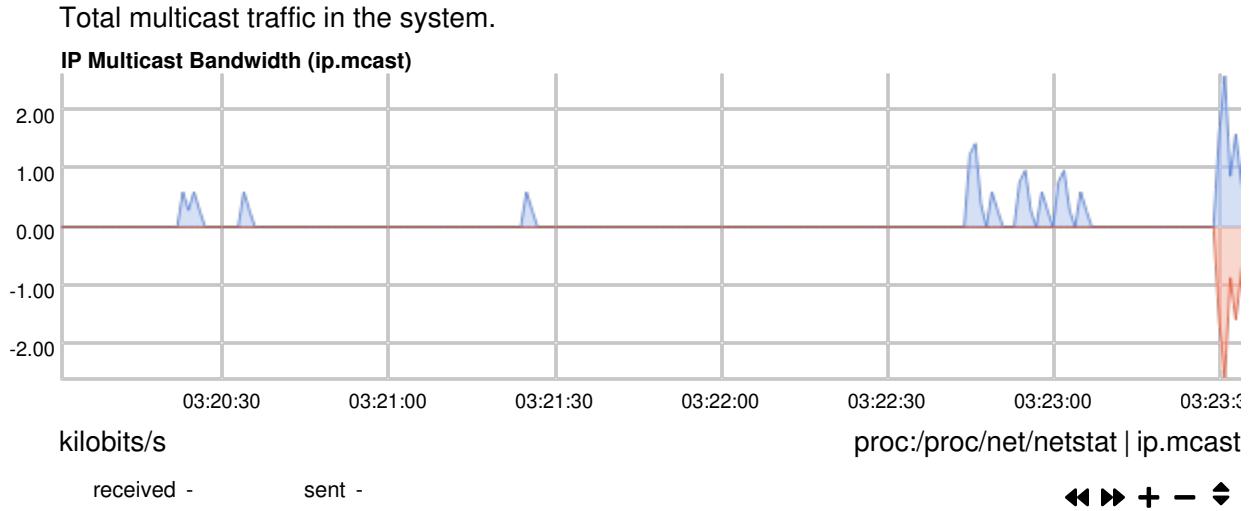


Total transferred broadcast packets in the system.



multicast

IP multicast (<https://en.wikipedia.org/wiki/Multicast>) is a technique for one-to-many communication over an IP network. Multicast uses network infrastructure efficiently by requiring the source to send a packet only once, even if it needs to be delivered to a large number of receivers. The nodes in the network take care of replicating the packet to reach multiple receivers only when necessary.

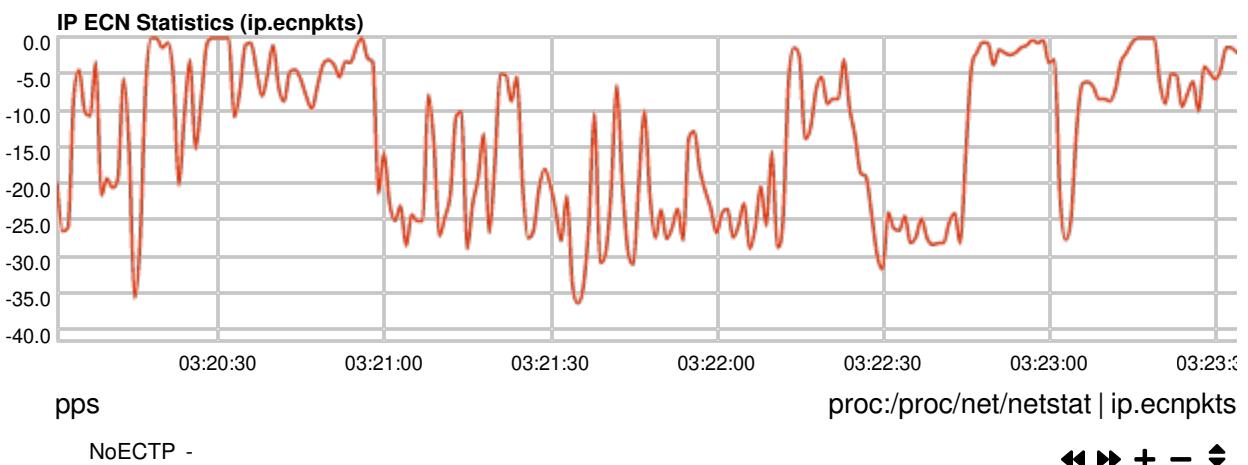


ecn

Explicit Congestion Notification (ECN) (https://en.wikipedia.org/wiki/Explicit_Congestion_Notation) is an extension to the IP and to the TCP 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.

Total number of received IP packets with ECN bits set in the system.

CEP - congestion encountered. **NoECTP** - non ECN-capable transport. **ECTP0** and **ECTP1** - ECN capable transport.

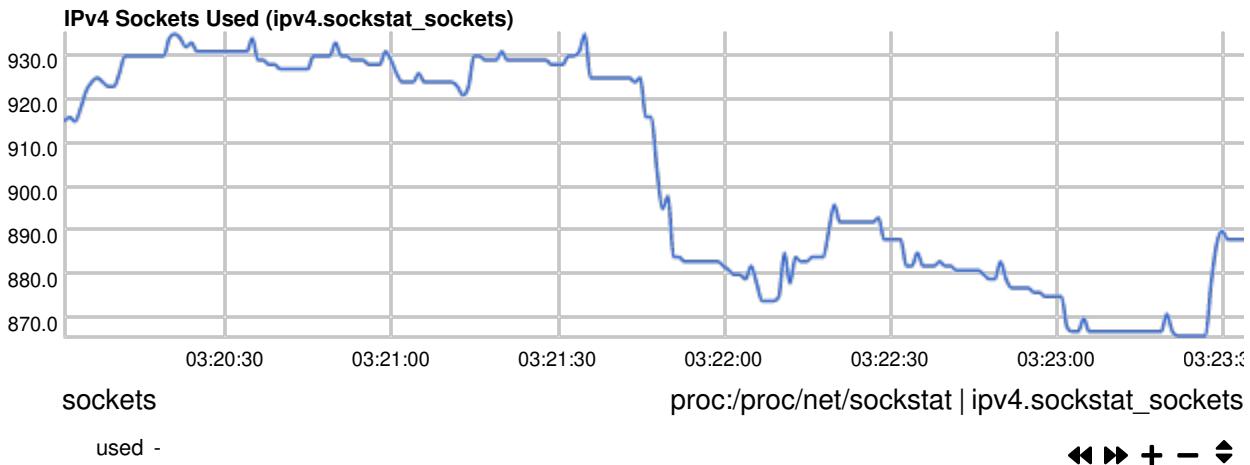


IPv4 Networking

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

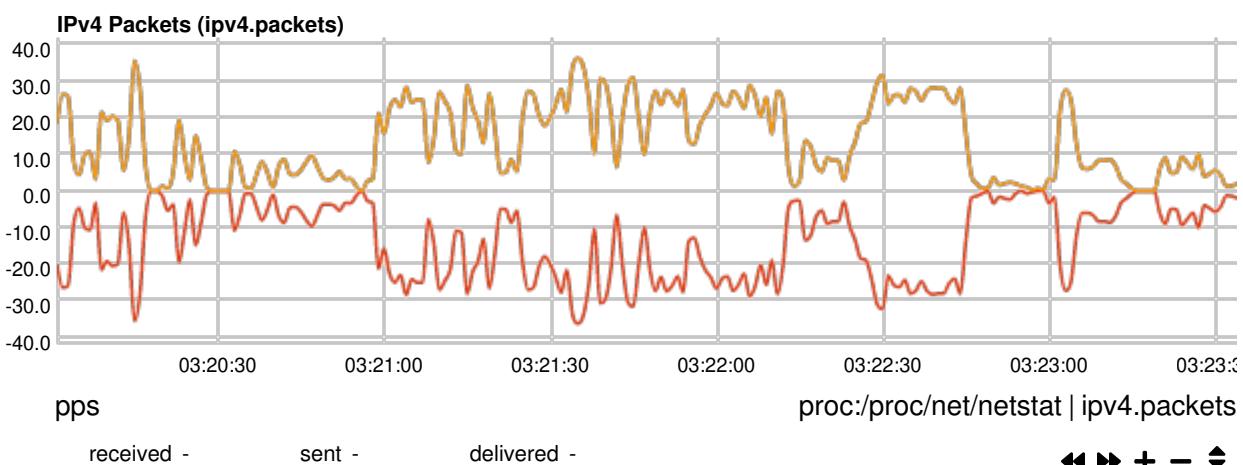
The total number of used sockets for all address families (https://man7.org/linux/man-pages/man7/address_families.7.html) in this system.



packets

IPv4 packets statistics for this host.

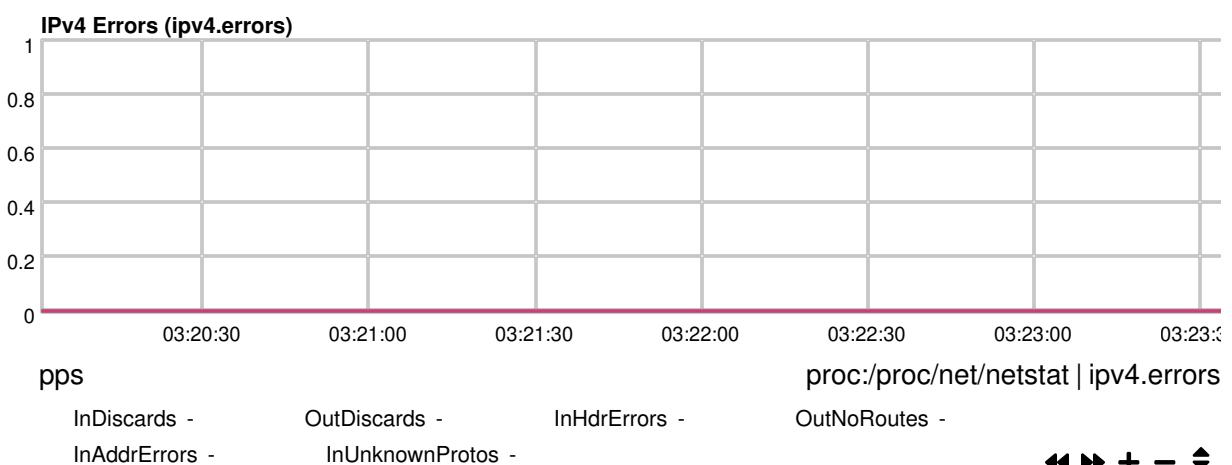
Received - packets received by the IP layer. This counter will be increased even if the packet is dropped later. **Sent** - packets sent via IP layer, for both single cast and multicast packets. This counter does not include any packets counted in **Forwarded**. **Forwarded** - input packets for which this host was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In hosts which do not act as IP Gateways, this counter will include only those packets which were Source-Routed (https://en.wikipedia.org/wiki/Source_routing) and the Source-Route option processing was successful. **Delivered** - packets delivered to the upper layer protocols, e.g. TCP, UDP, ICMP, and so on.



errors

The number of discarded IPv4 packets.

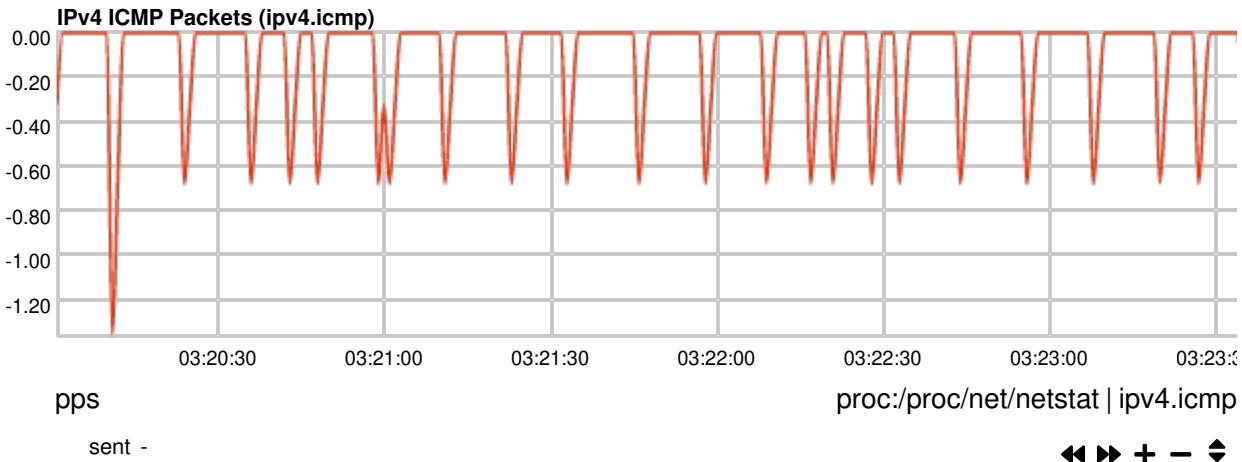
InDiscards, **OutDiscards** - inbound and outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. **InHdrErrors** - input packets that have been discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc. **OutNoRoutes** - packets that have been discarded because no route could be found to transmit them to their destination. This includes any packets which a host cannot route because all of its default gateways are down. **InAddrErrors** - input packets that have been discarded due to invalid IP address or the destination IP address is not a local address and IP forwarding is not enabled. **InUnknownProtos** - input packets which were discarded because of an unknown or unsupported protocol.



icmp

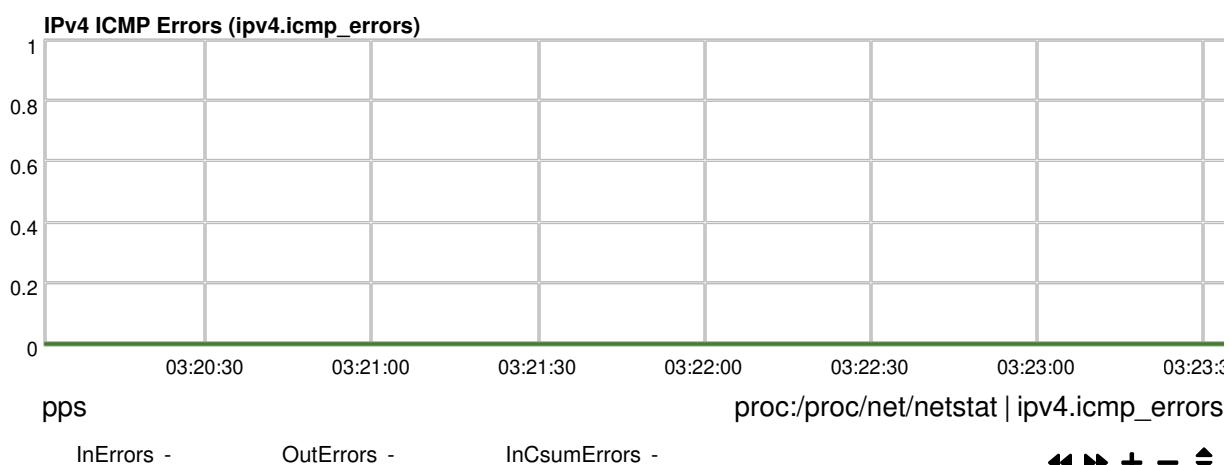
The number of transferred IPv4 ICMP messages.

Received, Sent - ICMP messages which the host received and attempted to send. Both these counters include errors.

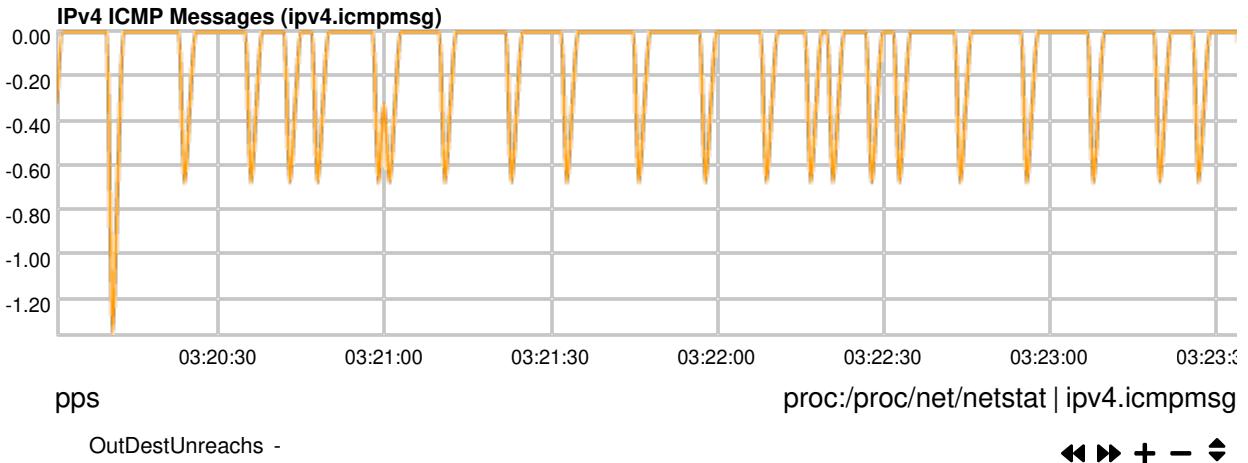


The number of IPv4 ICMP errors.

InErrors - received ICMP messages but determined as having ICMP-specific errors, e.g. bad ICMP checksums, bad length, etc. **OutErrors** - ICMP messages which this host did not send due to problems discovered within ICMP such as a lack of buffers. This counter does not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. **InCsumErrors** - received ICMP messages with bad checksum.

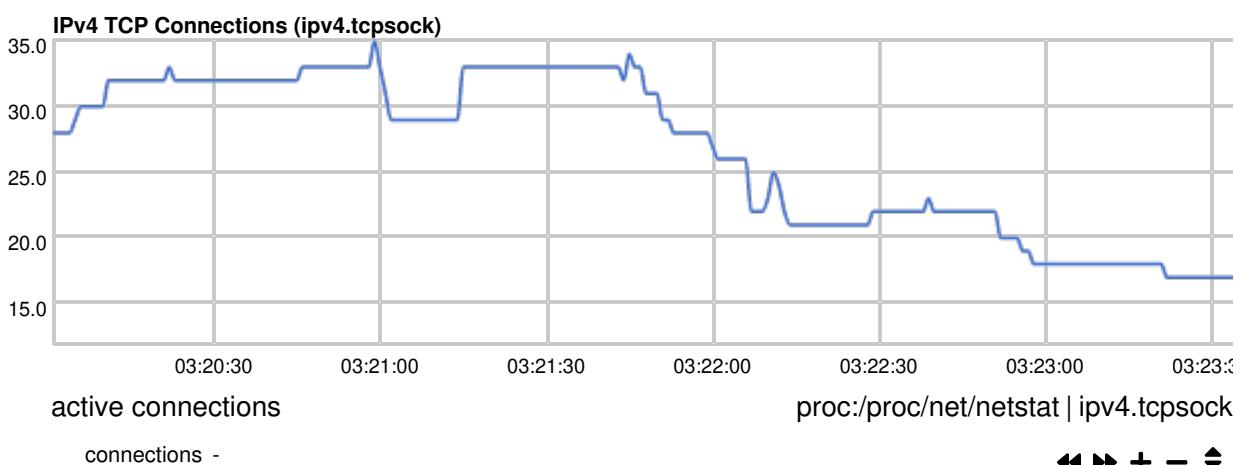


The number of transferred IPv4 ICMP control messages (<https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>).



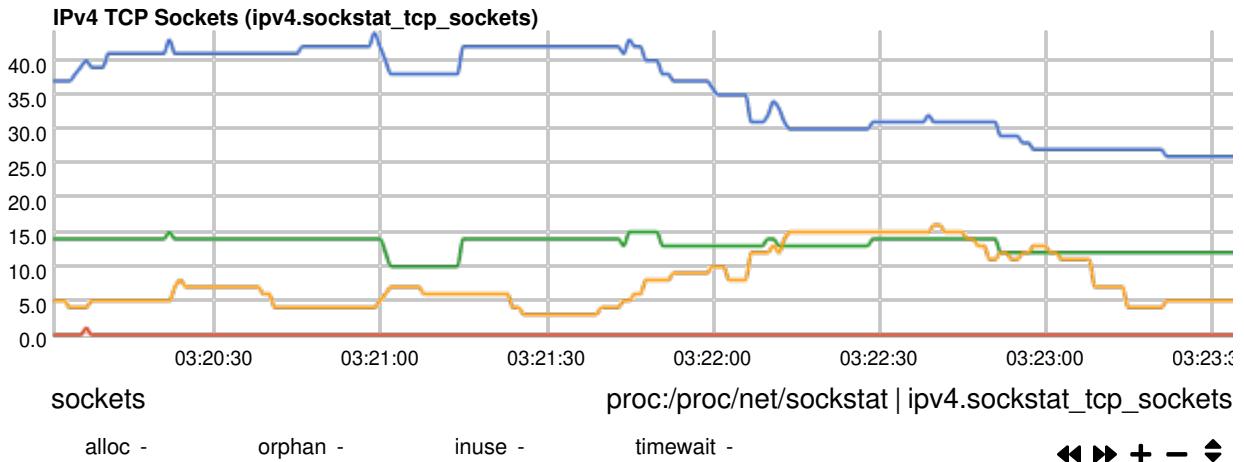
tcp

The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT. 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).



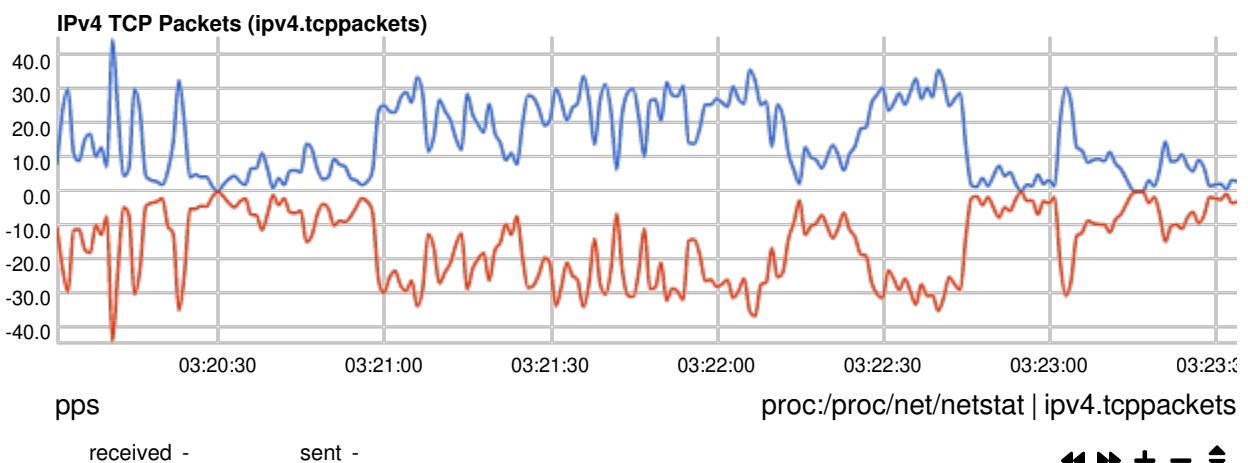
The number of TCP sockets in the system in certain states (https://en.wikipedia.org/wiki/Transmission_Control_Protocol#Protocol_operation).

Alloc - in any TCP state. **Orphan** - no longer attached to a socket descriptor in any user processes, but for which the kernel is still required to maintain state in order to complete the transport protocol. **InUse** - in any TCP state, excluding TIME-WAIT and CLOSED. **TimeWait** - in the TIME-WAIT state.



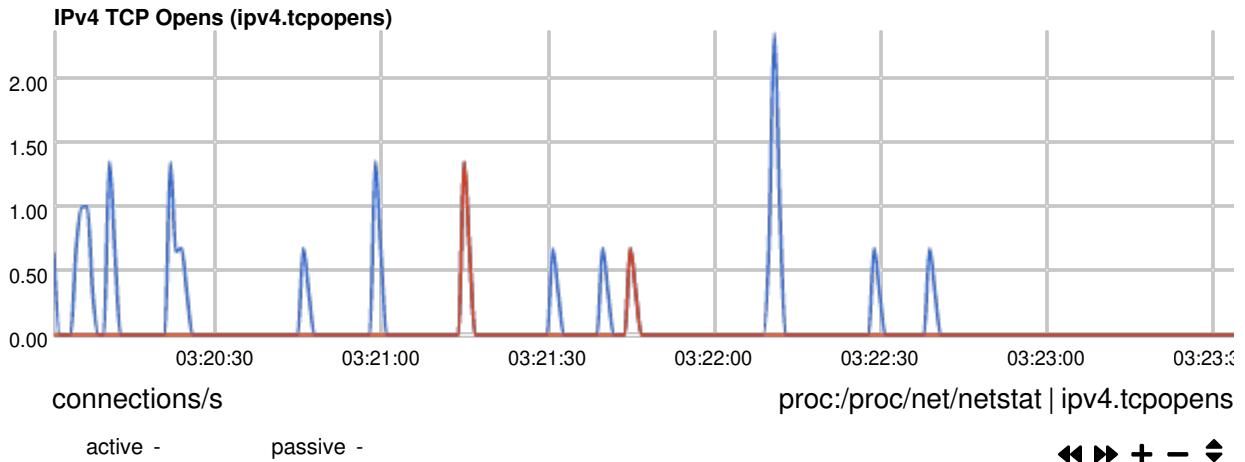
The number of packets transferred by the TCP layer.

Received - received packets, including those received in error, such as checksum error, invalid TCP header, and so on. **Sent** - sent packets, excluding the retransmitted packets. But it includes the SYN, ACK, and RST packets.



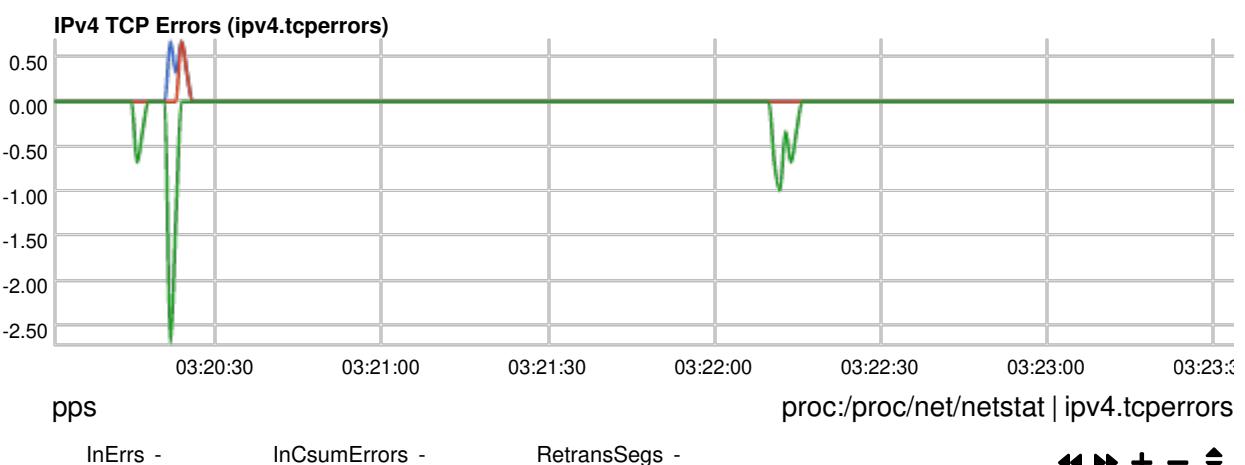
TCP connection statistics.

Active - number of outgoing TCP connections attempted by this host. **Passive** - number of incoming TCP connections accepted by this host.



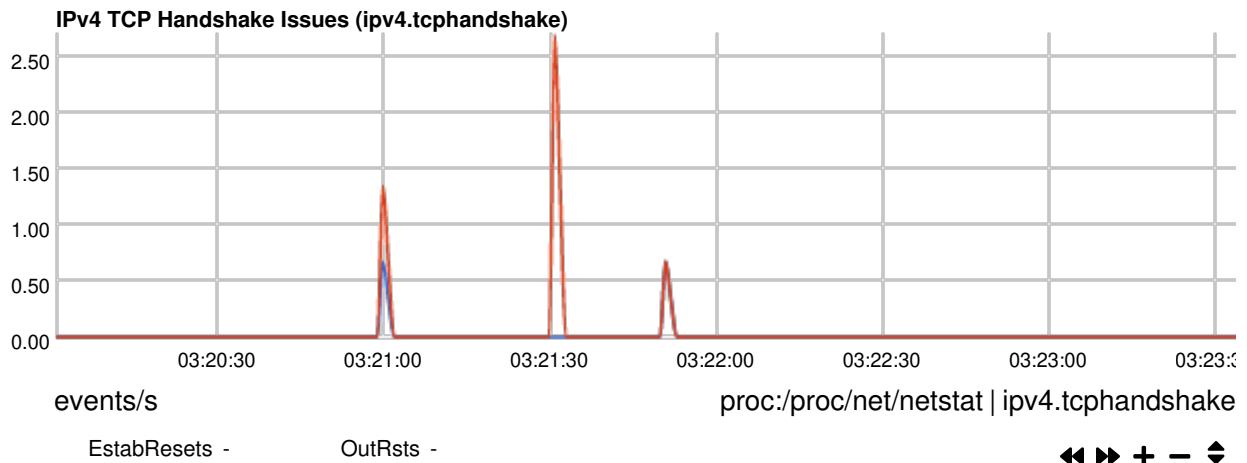
TCP errors.

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

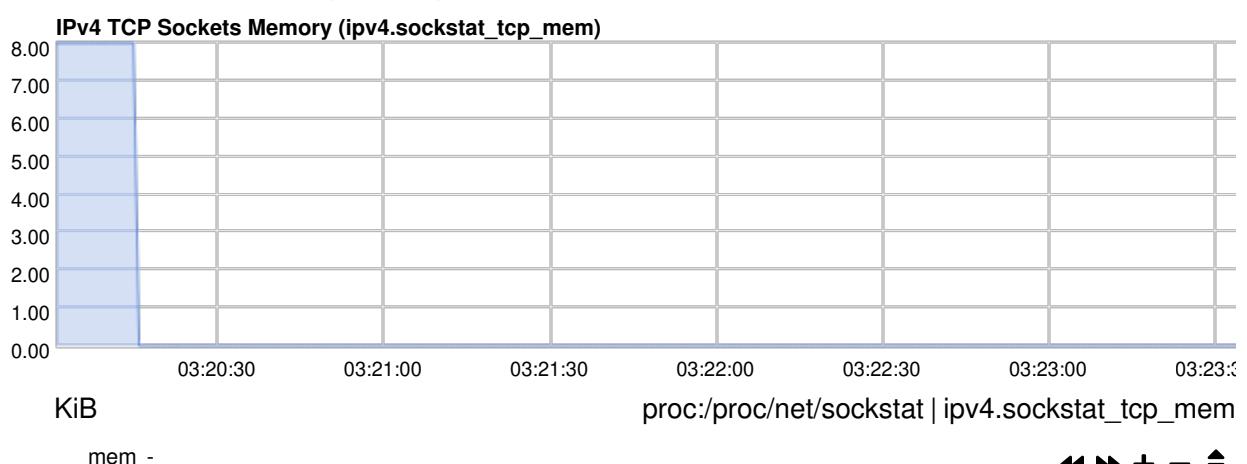


TCP handshake statistics.

EstabResets - established connections resets (i.e. connections that made a direct transition from ESTABLISHED or CLOSE_WAIT to CLOSED). **OutRsts** - TCP segments sent, with the RST flag set (for both IPv4 and IPv6). **AttemptFails** - 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. **SynRetrans** - shows retries for new outbound TCP connections, which can indicate general connectivity issues or backlog on the remote host.

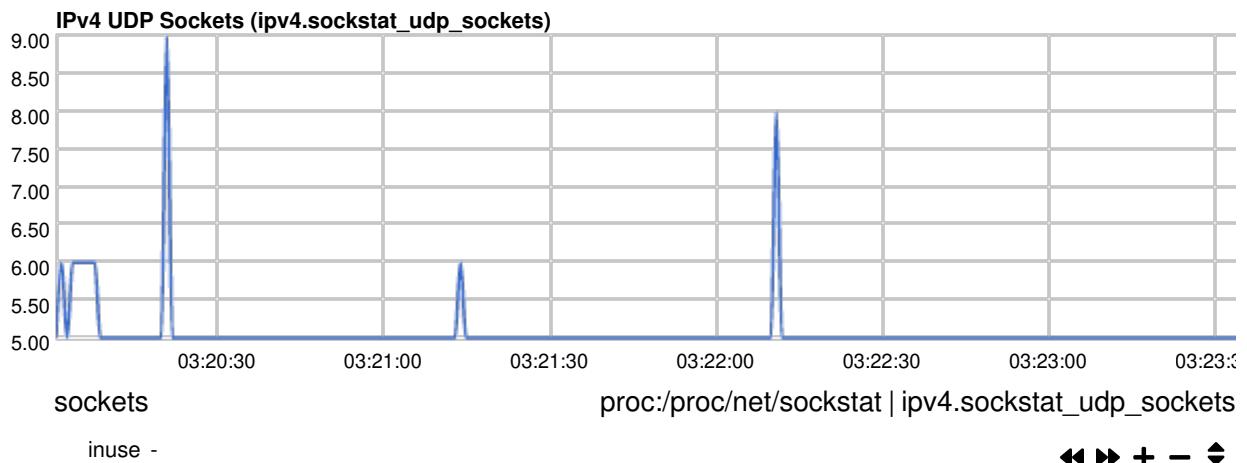


The amount of memory used by allocated TCP sockets.

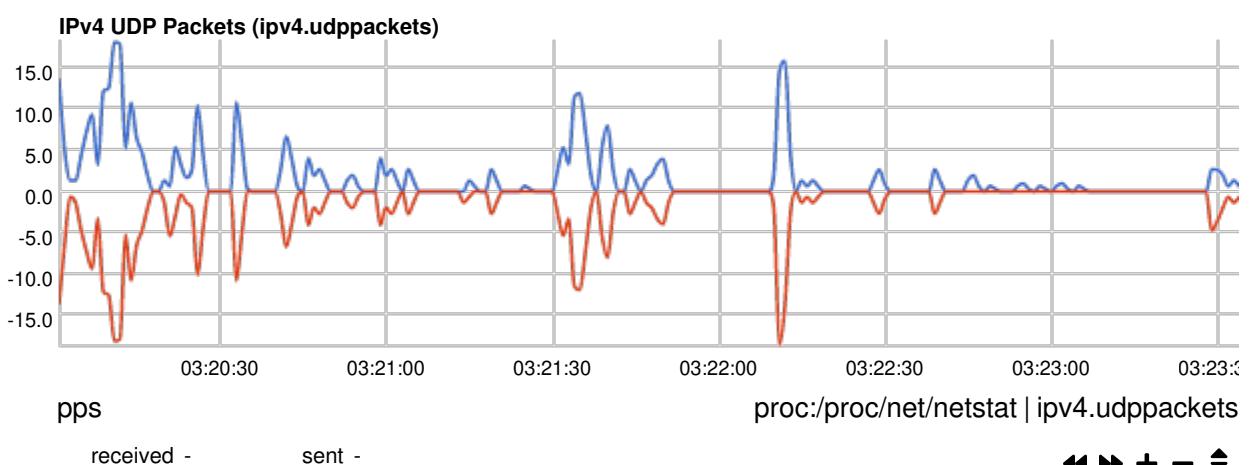


udp

The number of used UDP sockets.



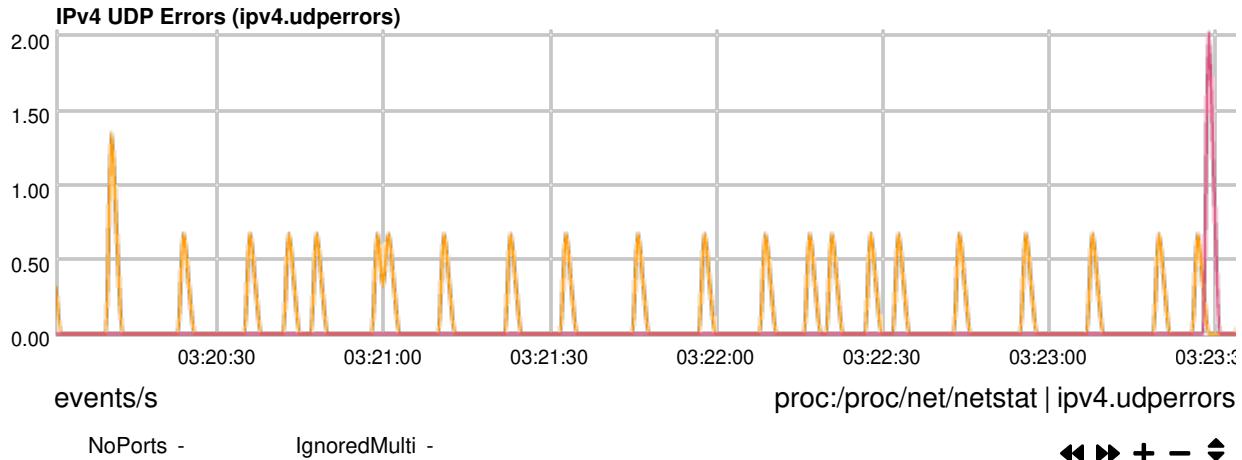
The number of transferred UDP packets.



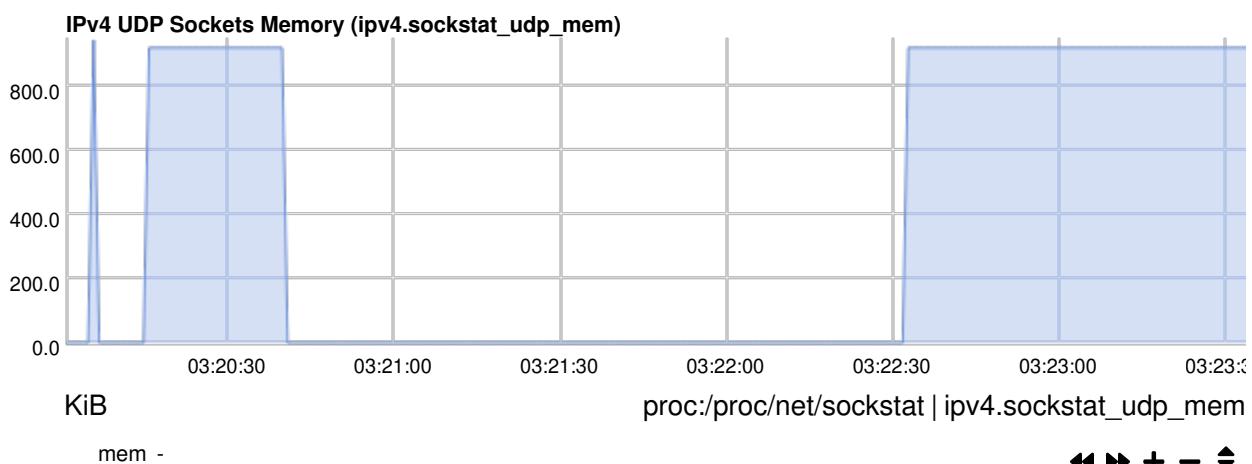
The number of errors encountered during transferring UDP packets.

RcvbufErrors - receive buffer is full. **SndbufErrors** - send buffer is full, no kernel memory available, or the IP layer reported an error when trying to send the packet and no error queue has been setup. **InErrors** - that is an aggregated counter for all errors, excluding **NoPorts**.

NoPorts - no application is listening at the destination port. **InCsumErrors** - a UDP checksum failure is detected. **IgnoredMulti** - ignored multicast packets.



The amount of memory used by allocated UDP sockets.



IPv6 Networking

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

packets

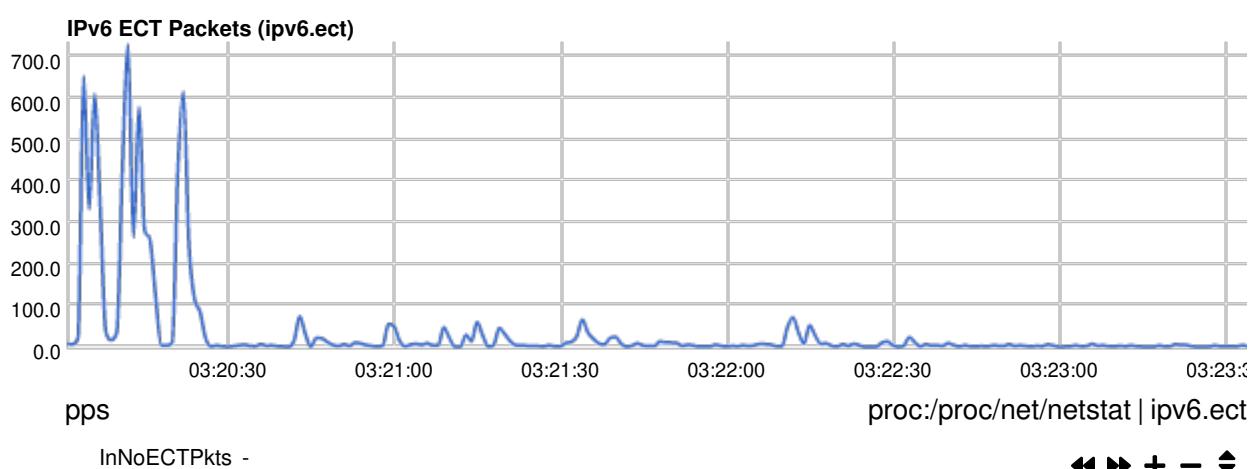
IPv6 packet statistics for this host.

Received - packets received by the IP layer. This counter will be increased even if the packet is dropped later. **Sent** - packets sent via IP layer, for both single cast and multicast packets. This counter does not include any packets counted in **Forwarded**. **Forwarded** - input packets for which this host was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In hosts which do not act as IP Gateways, this counter will include only those packets which were Source-Routed (https://en.wikipedia.org/wiki/Source_routing) and the Source-Route option processing was successful. **Delivers** - packets delivered to the upper layer protocols, e.g. TCP, UDP, ICMP, and so on.



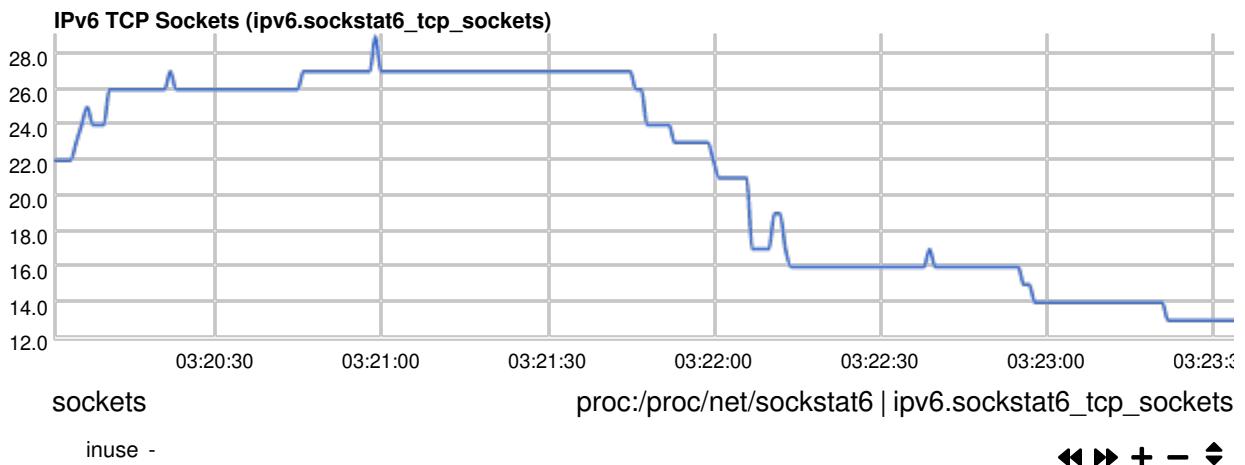
Total number of received IPv6 packets with ECN bits set in the system.

CEP - congestion encountered. **NoECTP** - non ECN-capable transport. **ECTP0** and **ECTP1** - ECN capable transport.



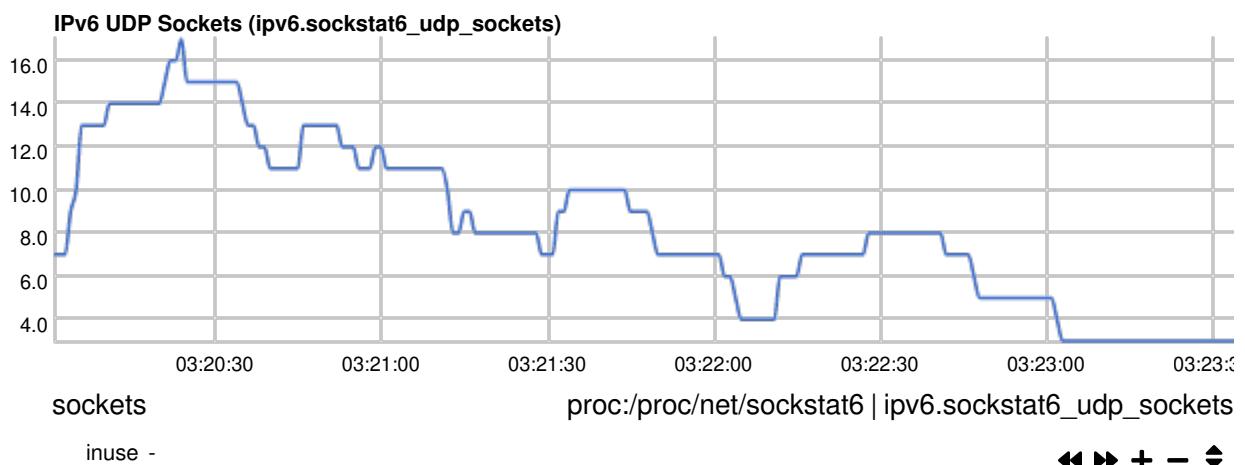
tcp6

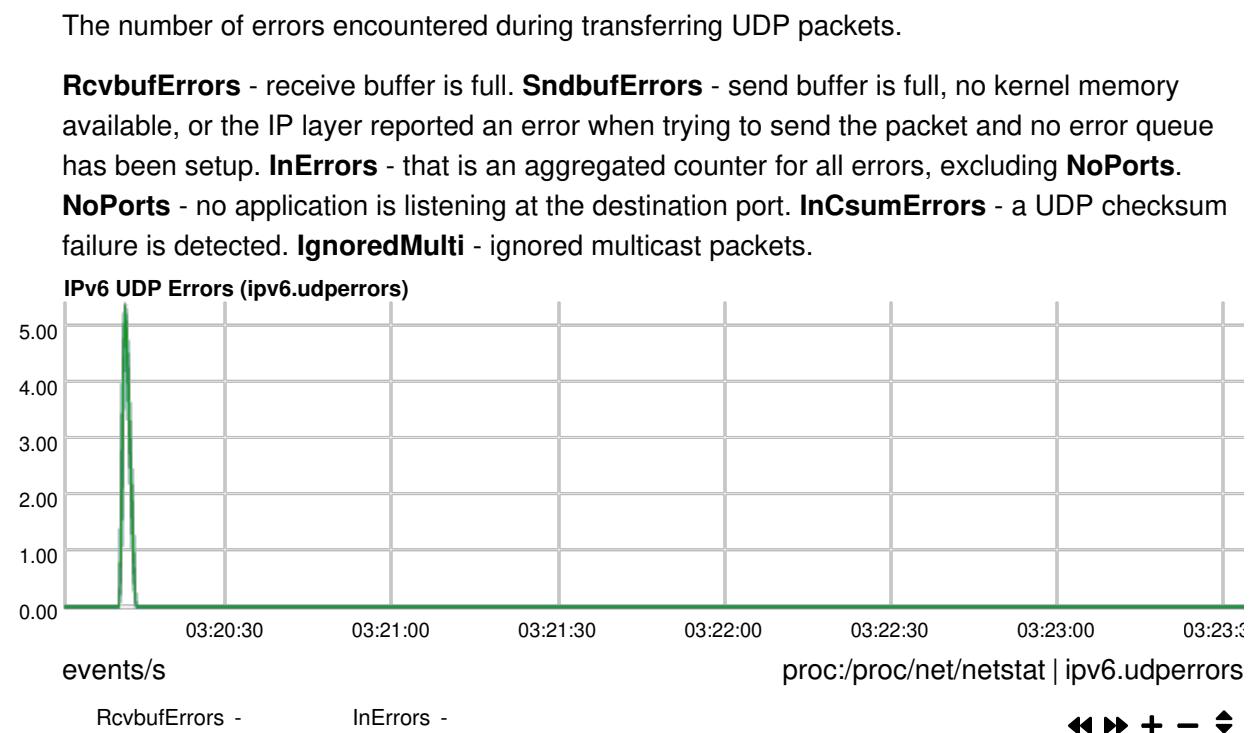
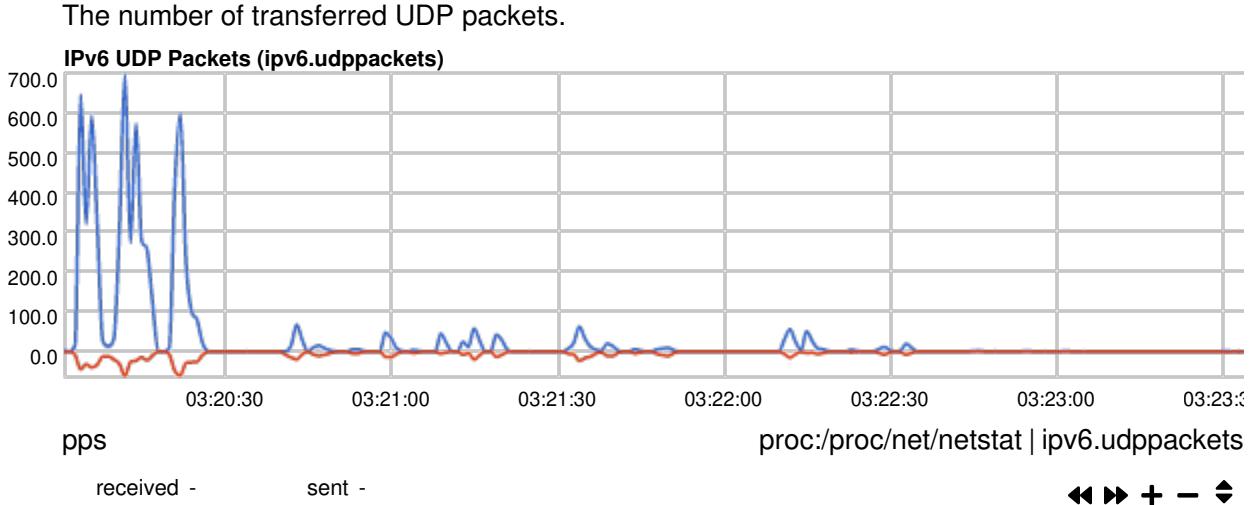
The number of TCP sockets in any state (https://en.wikipedia.org/wiki/Transmission_Control_Protocol#Protocol_operation), excluding TIME-WAIT and CLOSED.



udp6

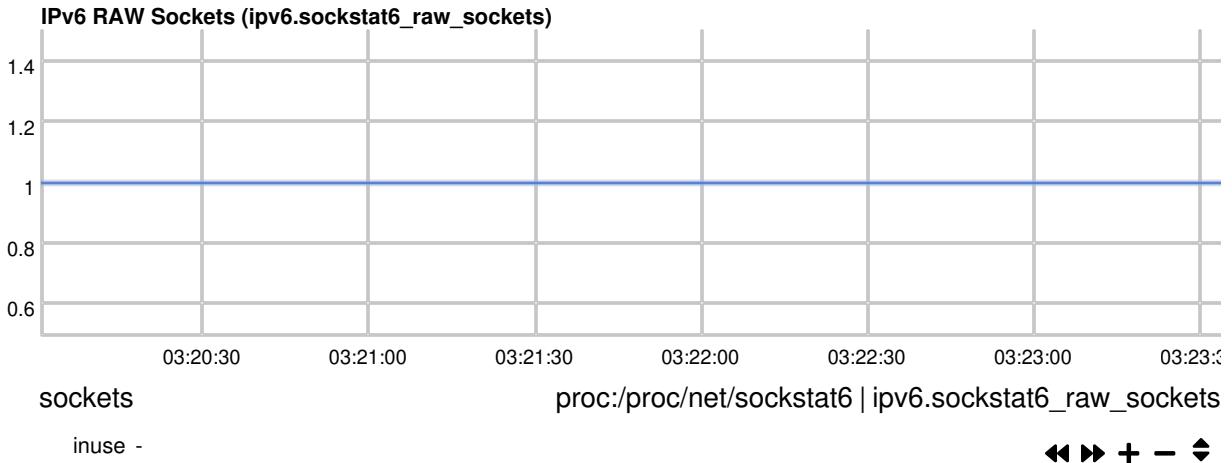
The number of used UDP sockets.





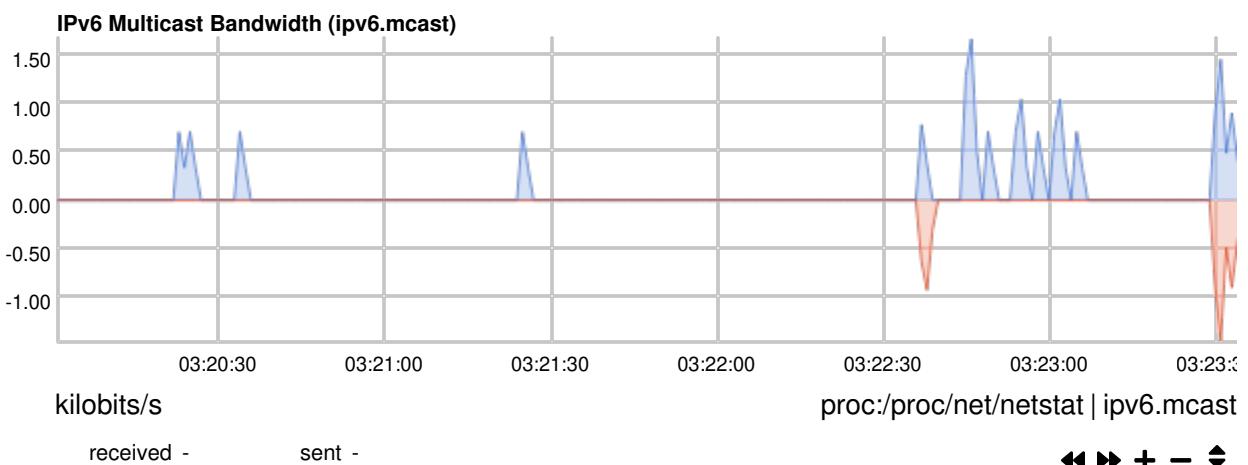
raw6

The number of used raw sockets (https://en.wikipedia.org/wiki/Network_socket#Types).

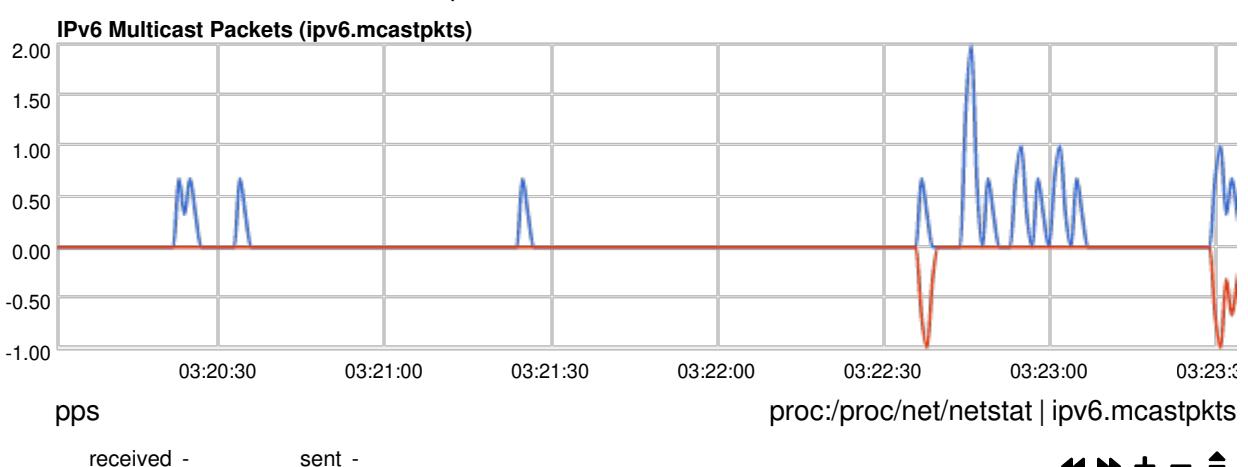


multicast6

Total IPv6 multicast traffic.



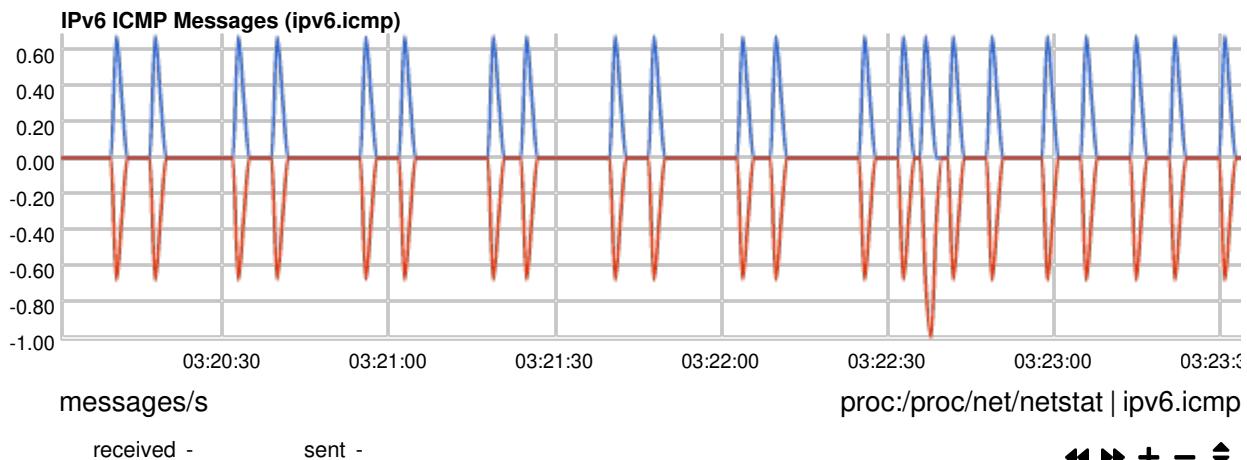
Total transferred IPv6 multicast packets.



icmp6

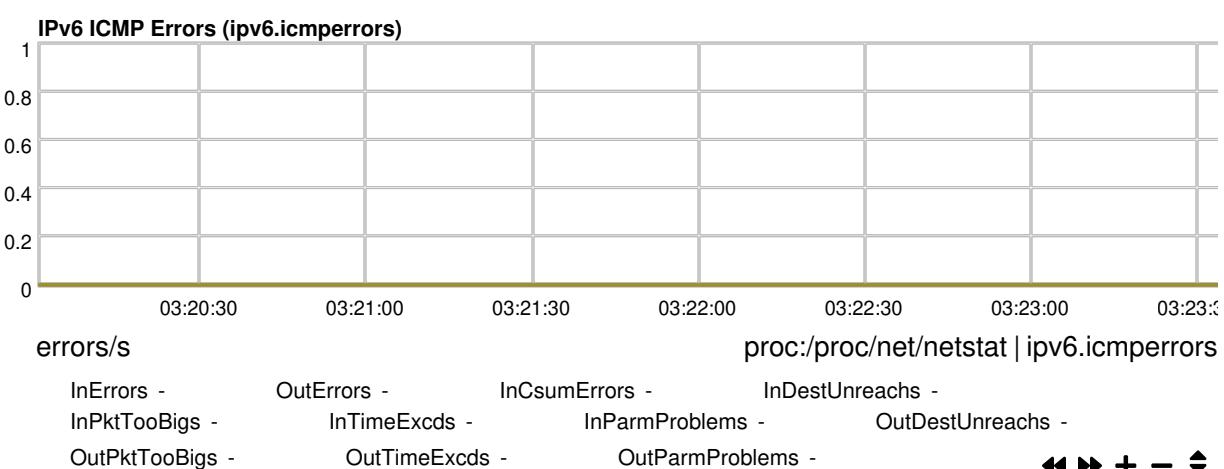
The number of transferred ICMPv6 messages.

Received, Sent - ICMP messages which the host received and attempted to send. Both these counters include errors.



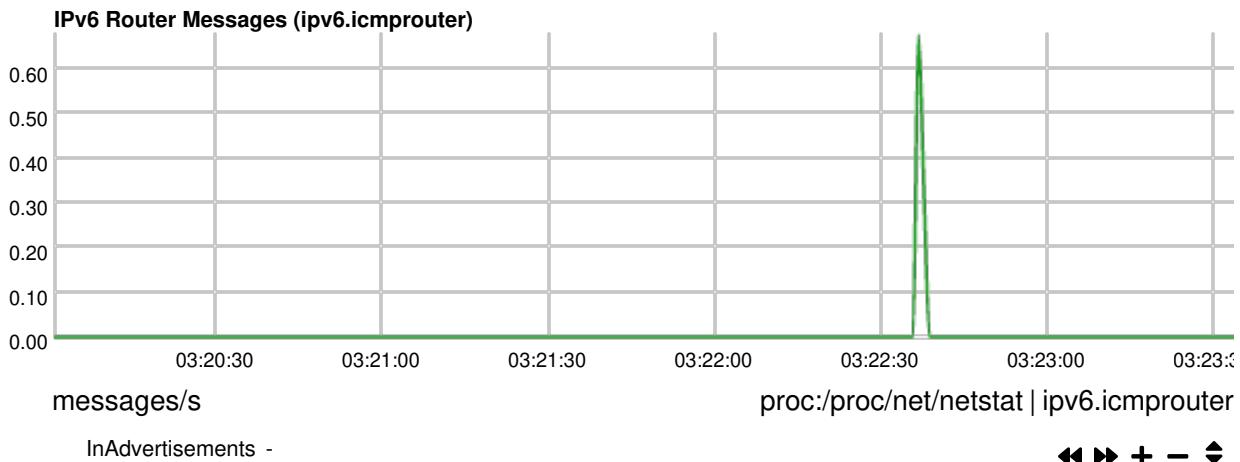
The number of ICMPv6 errors and error messages (<https://www.rfc-editor.org/rfc/rfc4443.html#section-3>).

InErrors, OutErrors - bad ICMP messages (bad ICMP checksums, bad length, etc.).
InCsumErrors - wrong checksum.



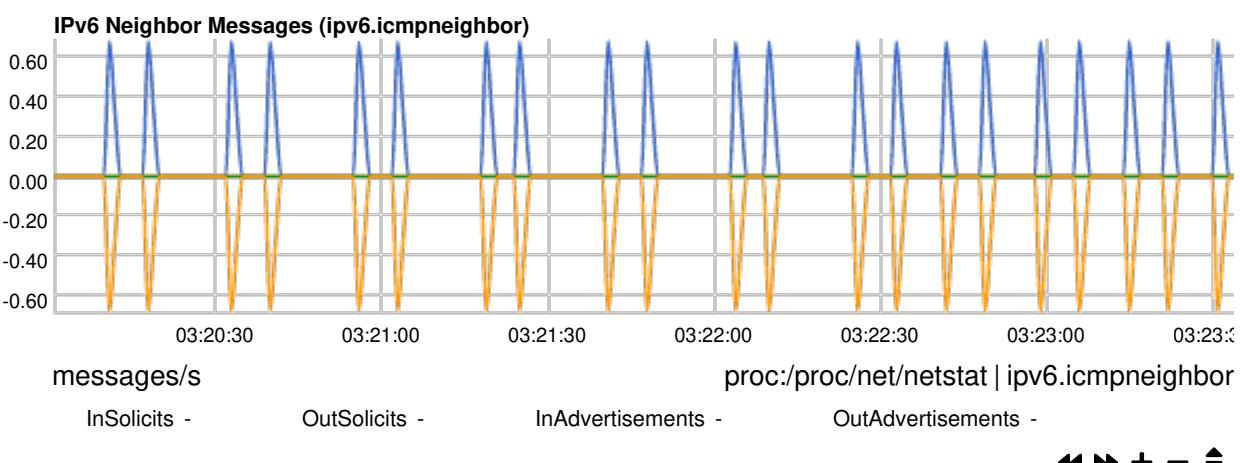
The number of transferred ICMPv6 Router Discovery (https://en.wikipedia.org/wiki/Neighbor_Discovery_Protocol) messages.

Router **Solicitations** message is sent from a computer host to any routers on the local area network to request that they advertise their presence on the network. Router **Advertisement** message is sent by a router on the local area network to announce its IP address as available for routing.

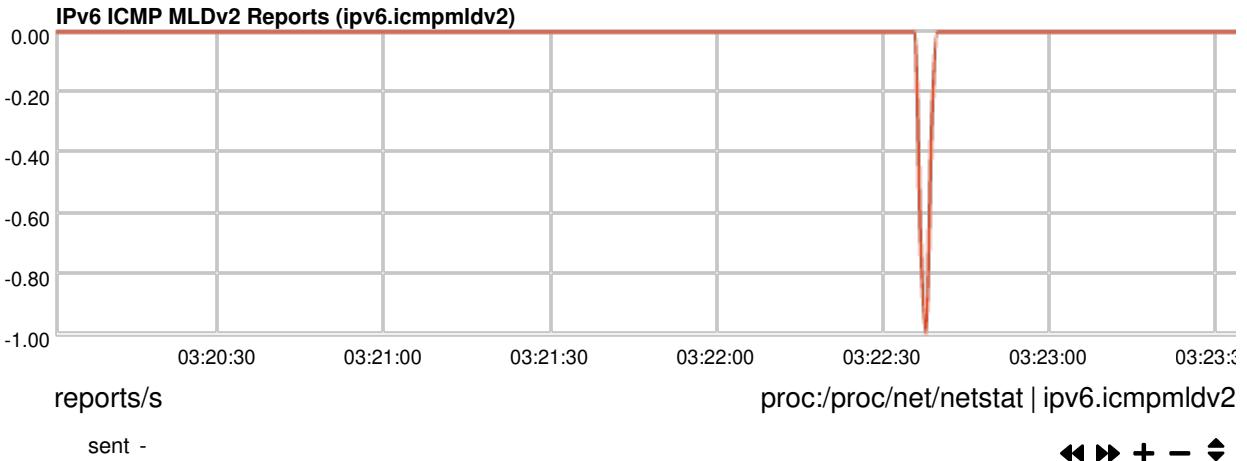


The number of transferred ICMPv6 Neighbour Discovery (https://en.wikipedia.org/wiki/Neighbor_Discovery_Protocol) messages.

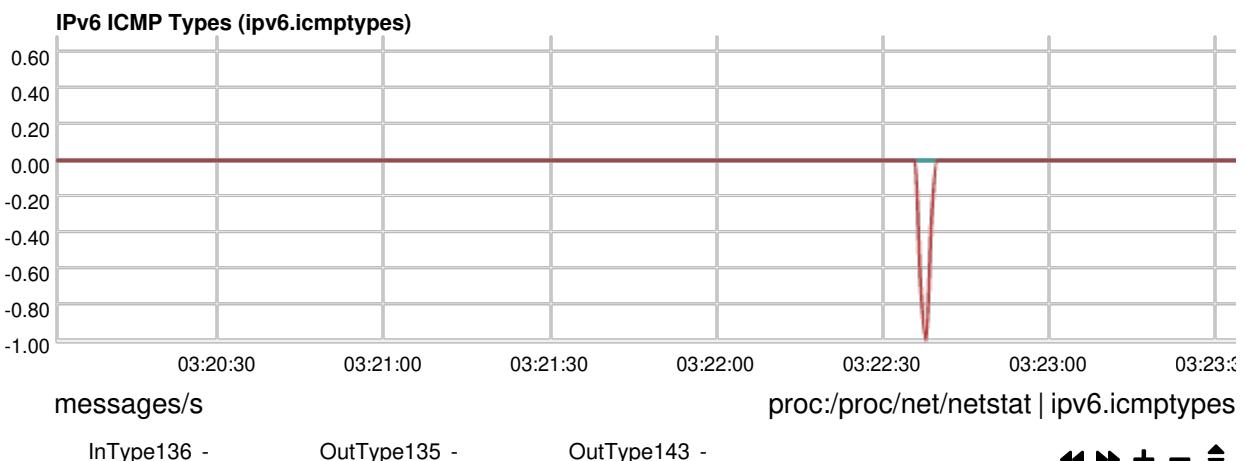
Neighbor **Solicitations** are used by nodes to determine the link layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link layer address. Neighbor **Advertisements** are used by nodes to respond to a Neighbor Solicitation message.



The number of transferred ICMPv6 Multicast Listener Discovery (https://en.wikipedia.org/wiki/Multicast_Listener_Discovery) (MLD) messages.



The number of transferred ICMPv6 messages of certain types (https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol_for_IPv6#Types).



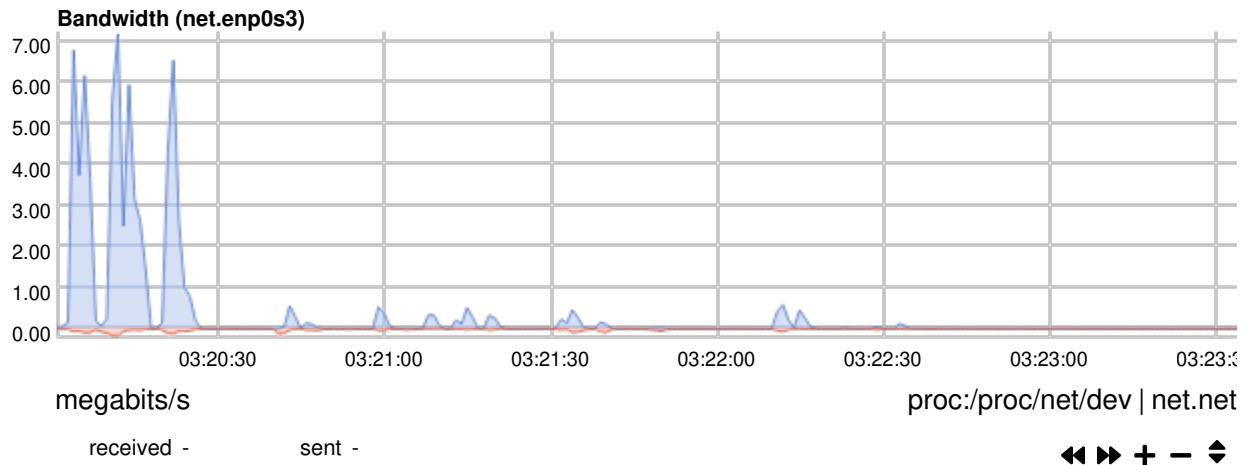
Network Interfaces

Performance metrics for network interfaces (<https://www.kernel.org/doc/html/latest/networking/statistics.html>).

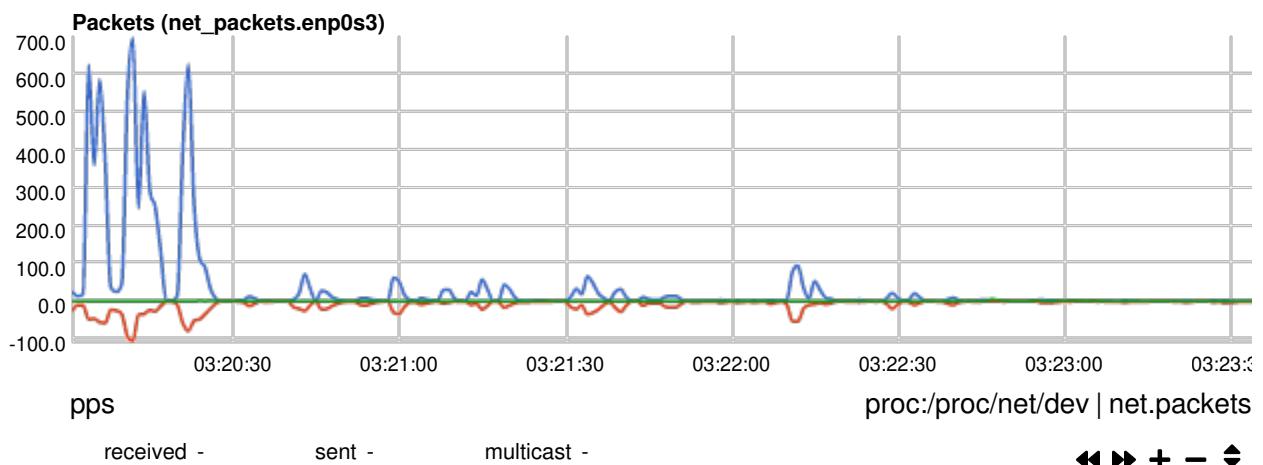
Netdata retrieves this data reading the `/proc/net/dev` file and `/sys/class/net/` directory.

enp0s3

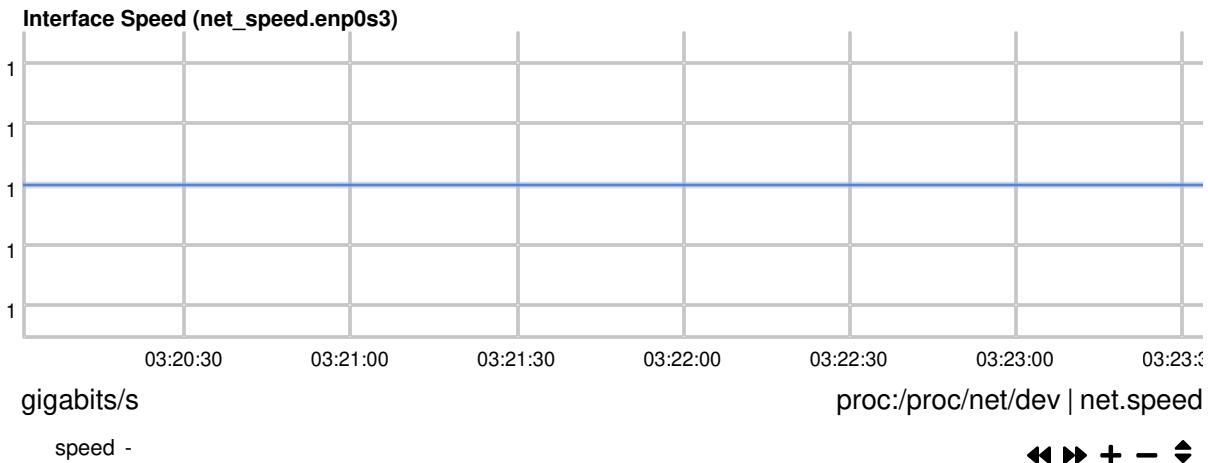
The amount of traffic transferred by the network interface.



The number of packets transferred by the network interface. Received multicast (<https://en.wikipedia.org/wiki/Multicast>) counter is commonly calculated at the device level (unlike **received**) and therefore may include packets which did not reach the host.

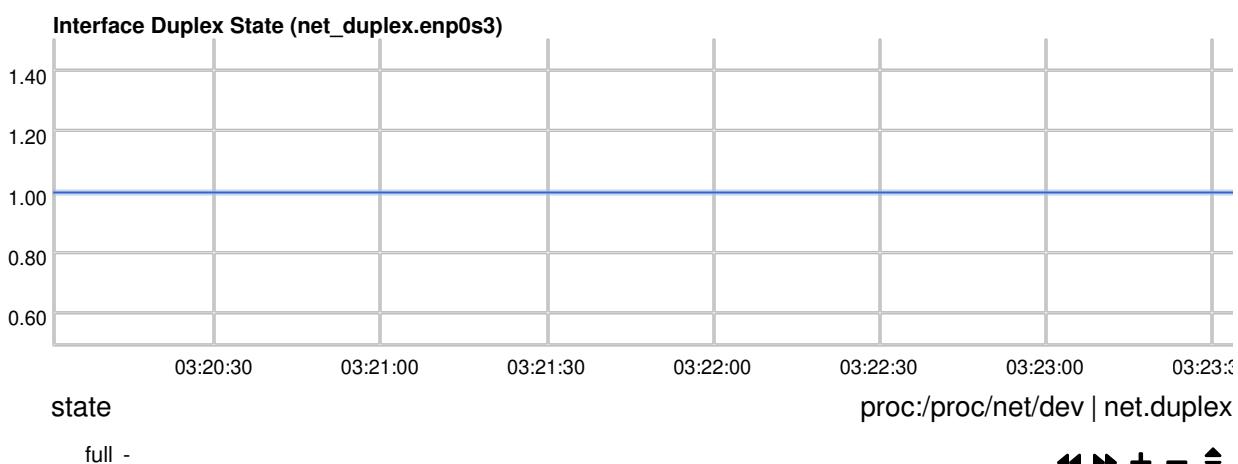


The interface's latest or current speed that the network adapter negotiated (<https://en.wikipedia.org/wiki/Autonegotiation>) with the device it is connected to. This does not give the max supported speed of the NIC.



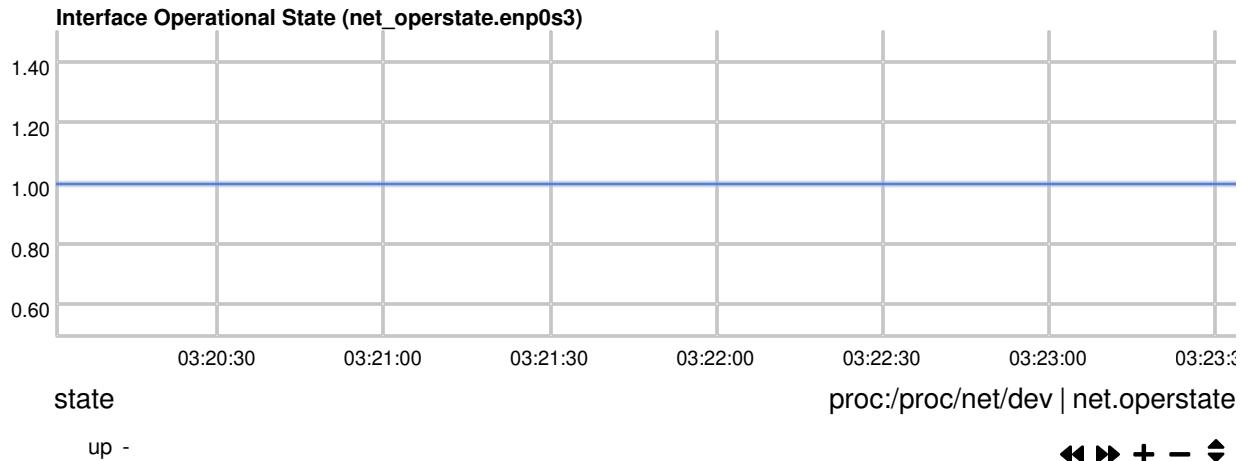
The interface's latest or current duplex ([https://en.wikipedia.org/wiki/Duplex_\(telecommunications\)](https://en.wikipedia.org/wiki/Duplex_(telecommunications))) that the network adapter negotiated (<https://en.wikipedia.org/wiki/Autonegotiation>) with the device it is connected to.

Unknown - the duplex mode can not be determined. **Half duplex** - the communication is one direction at a time. **Full duplex** - the interface is able to send and receive data simultaneously.

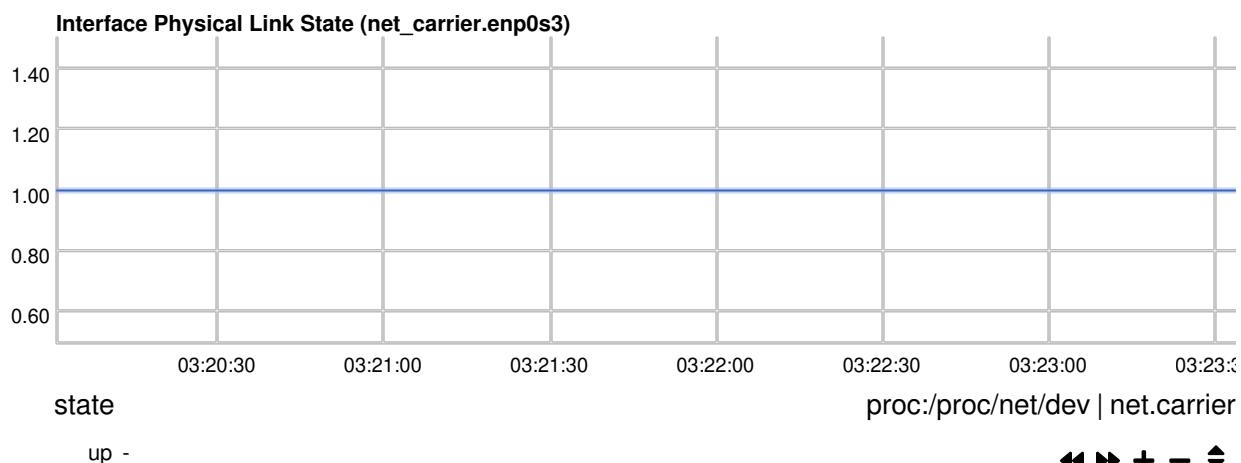


The current operational state (<https://datatracker.ietf.org/doc/html/rfc2863>) of the interface.

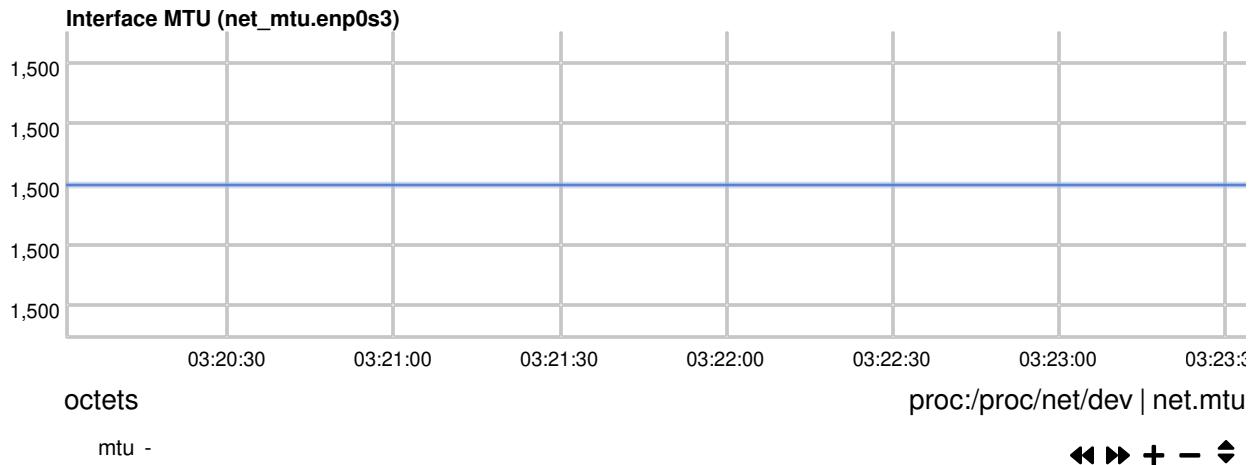
Unknown - the state can not be determined. **NotPresent** - the interface has missing (typically, hardware) components. **Down** - the interface is unable to transfer data on L1, e.g. ethernet is not plugged or interface is administratively down. **LowerLayerDown** - the interface is down due to state of lower-layer interface(s). **Testing** - the interface is in testing mode, e.g. cable test. It can't be used for normal traffic until tests complete. **Dormant** - the interface is L1 up, but waiting for an external event, e.g. for a protocol to establish. **Up** - the interface is ready to pass packets and can be used.



The current physical link state of the interface.



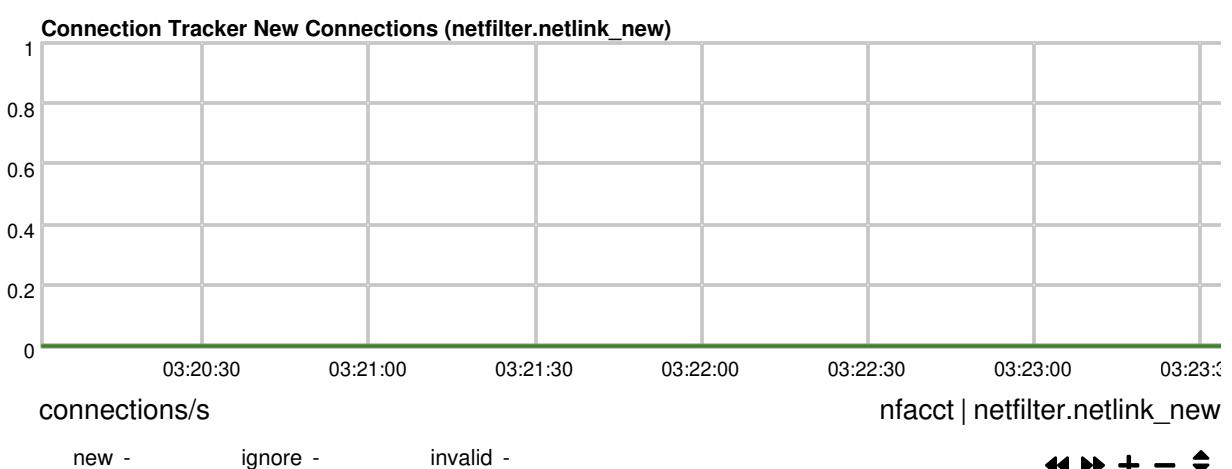
The interface's currently configured Maximum transmission unit (https://en.wikipedia.org/wiki/Maximum_transmission_unit) (MTU) value. MTU is the size of the largest protocol data unit that can be communicated in a single network layer transaction.

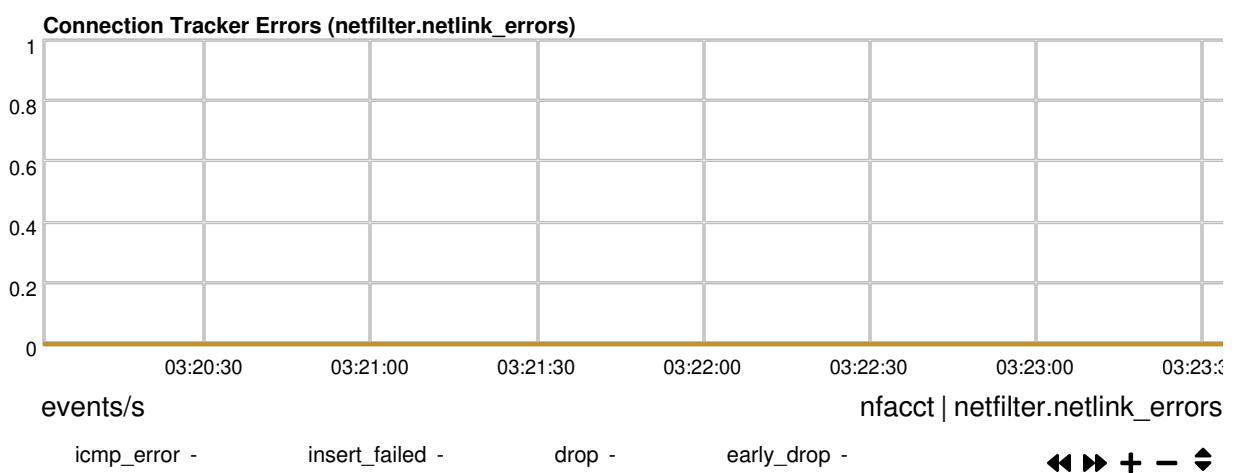
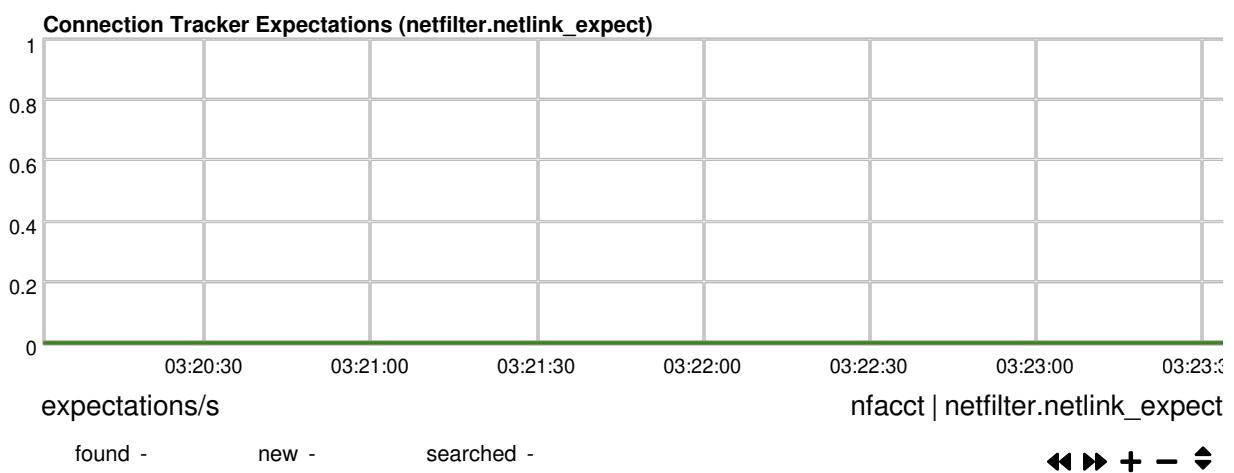
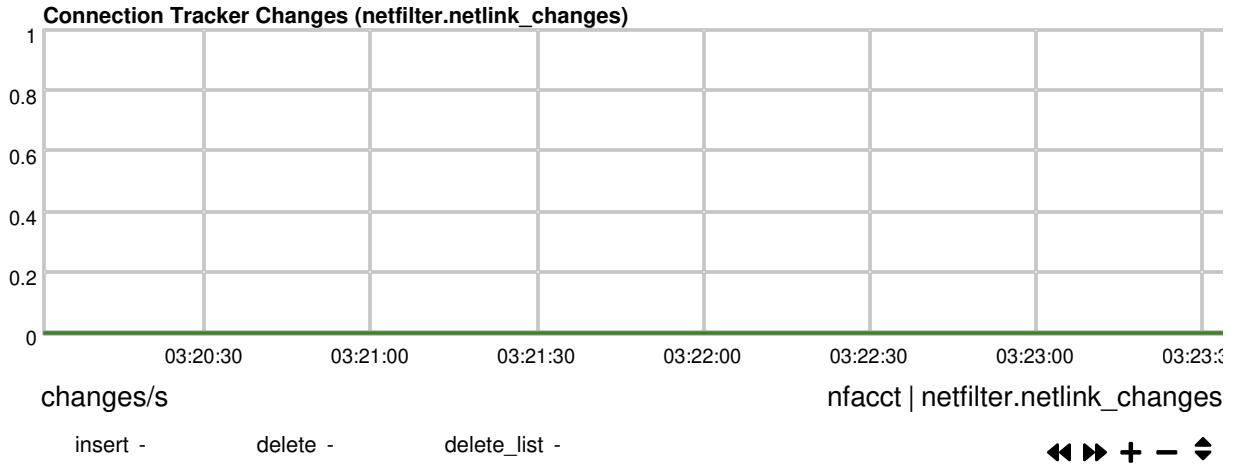


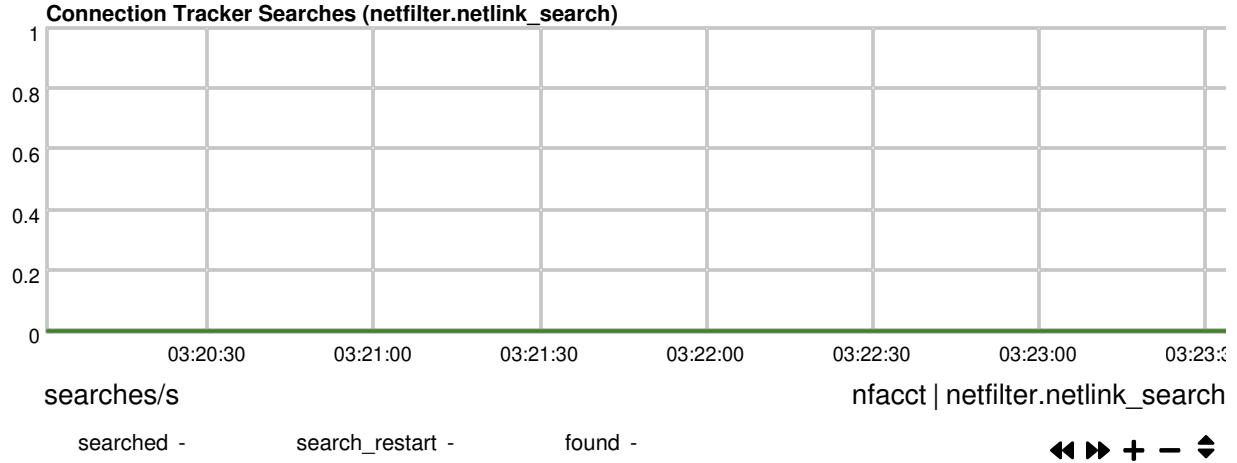
🛡 Firewall (netfilter)

Performance metrics of the netfilter components.

netlink





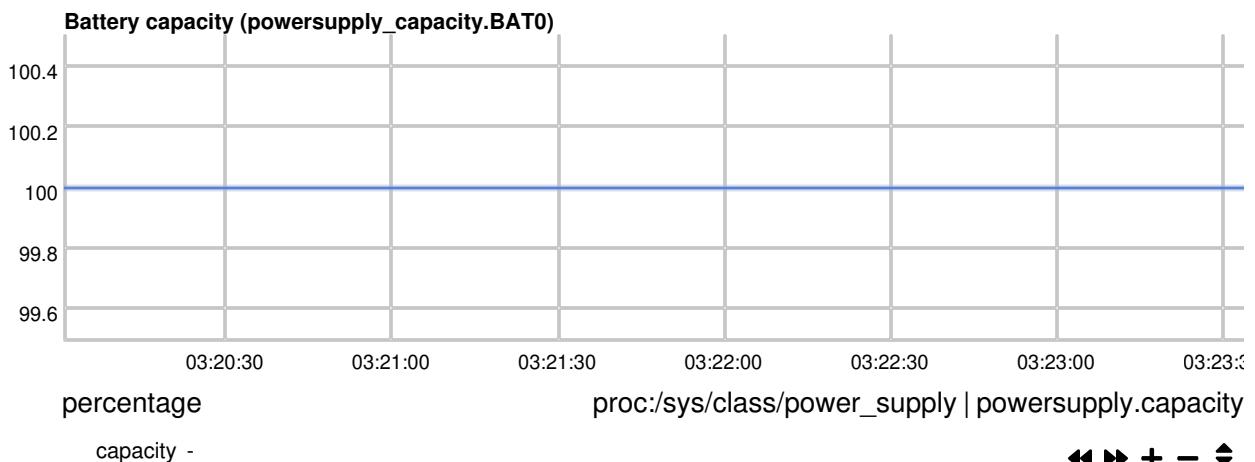


🔋 Power Supply

Statistics for the various system power supplies. Data collected from Linux power supply class (https://www.kernel.org/doc/Documentation/power/power_supply_class.txt).

BAT0

The current battery charge.

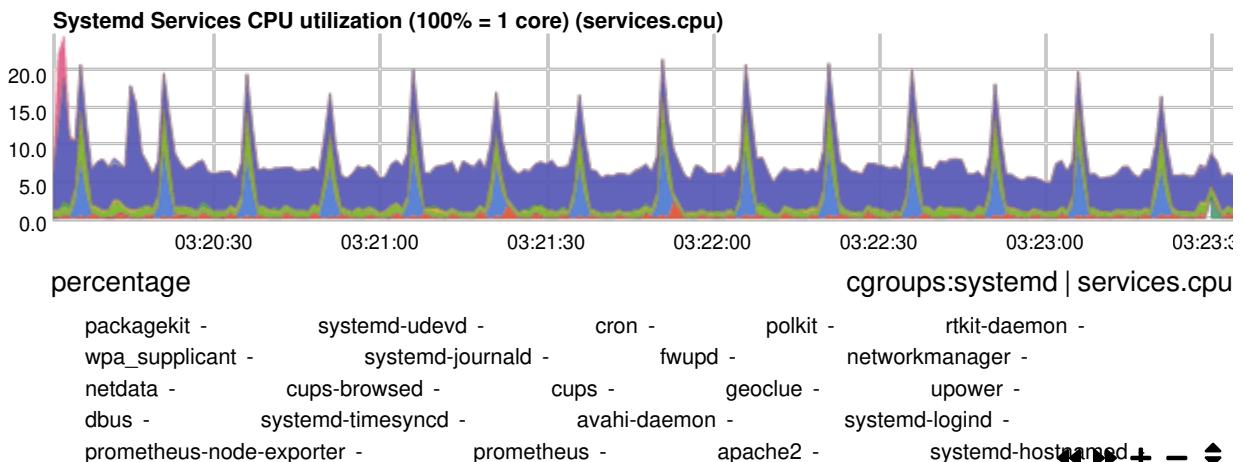


⚙️ systemd Services

Resources utilization of systemd services. Netdata monitors all systemd services via cgroups (<https://en.wikipedia.org/wiki/Cgroups>) (the resources accounting used by containers).

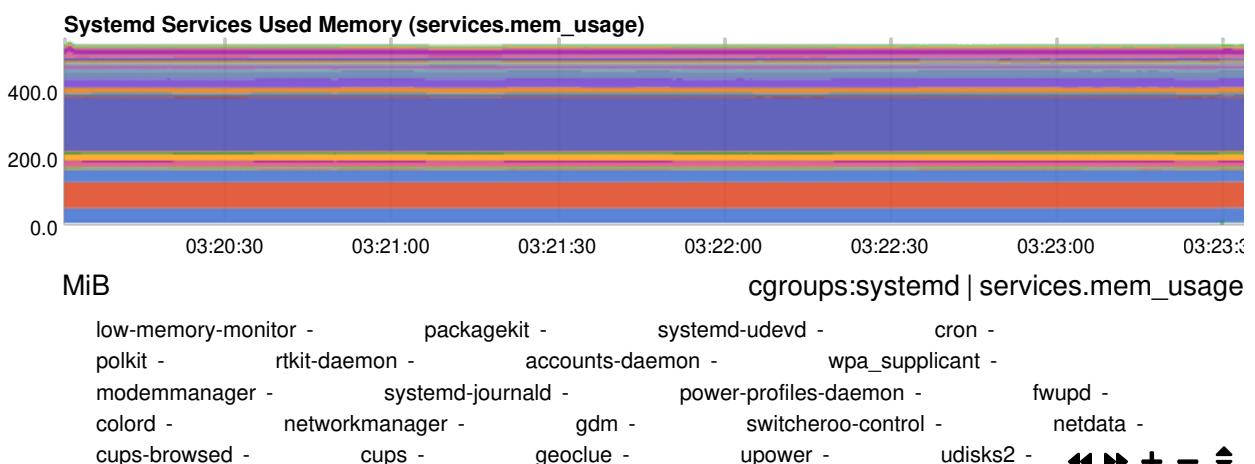
cpu

Total CPU utilization within the system-wide CPU resources (all cores). The amount of time spent by tasks of the cgroup in user and kernel (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) modes.



mem

The amount of used RAM.

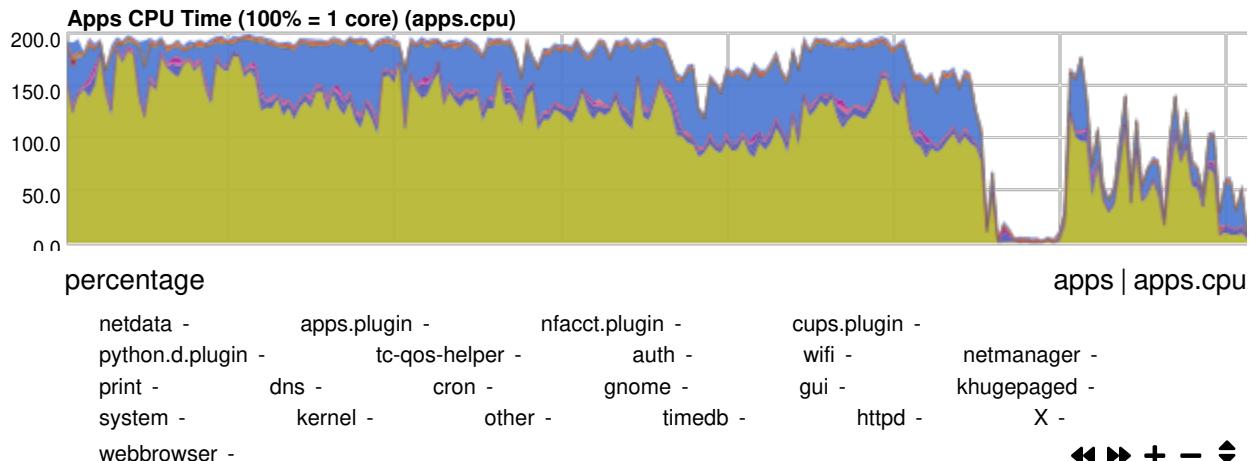


❤️ Applications

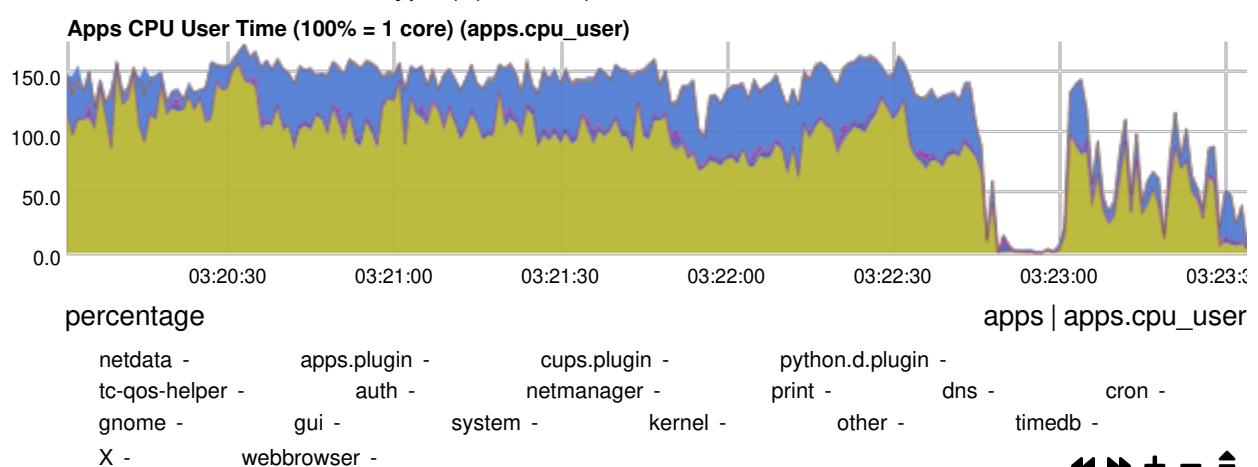
Per application statistics are collected using apps.plugin (<https://learn.netdata.cloud/docs/agent/collectors/apps.plugin>). This plugin walks through all processes and aggregates statistics for application groups (<https://learn.netdata.cloud/docs/agent/collectors/apps.plugin#configuration>). The plugin also counts the resources of exited children. So for processes like shell scripts, the reported values include the resources used by the commands these scripts run within each timeframe.

cpu

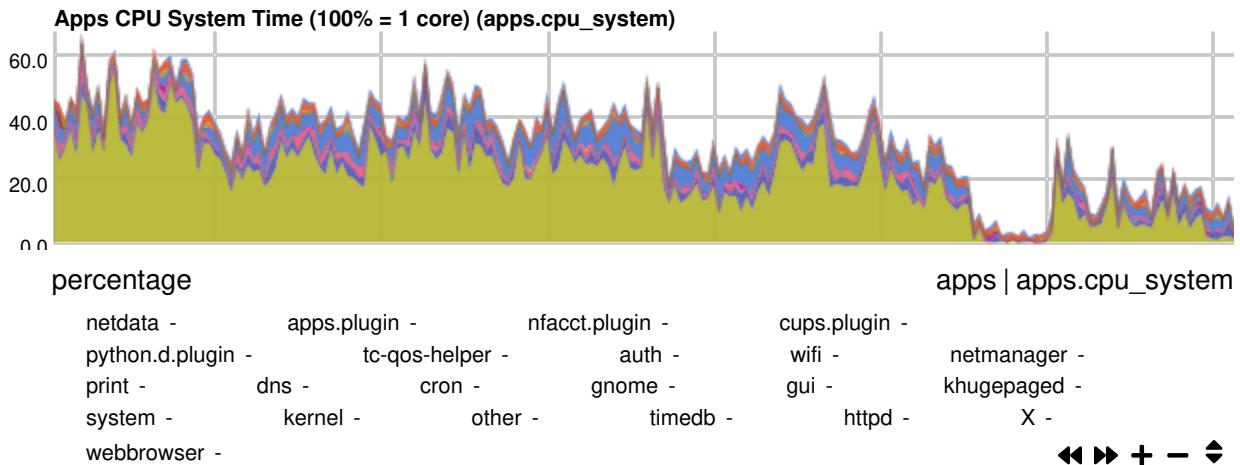
Total CPU utilization (all cores). It includes user, system and guest time.



The amount of time the CPU was busy executing code in user mode (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) (all cores).

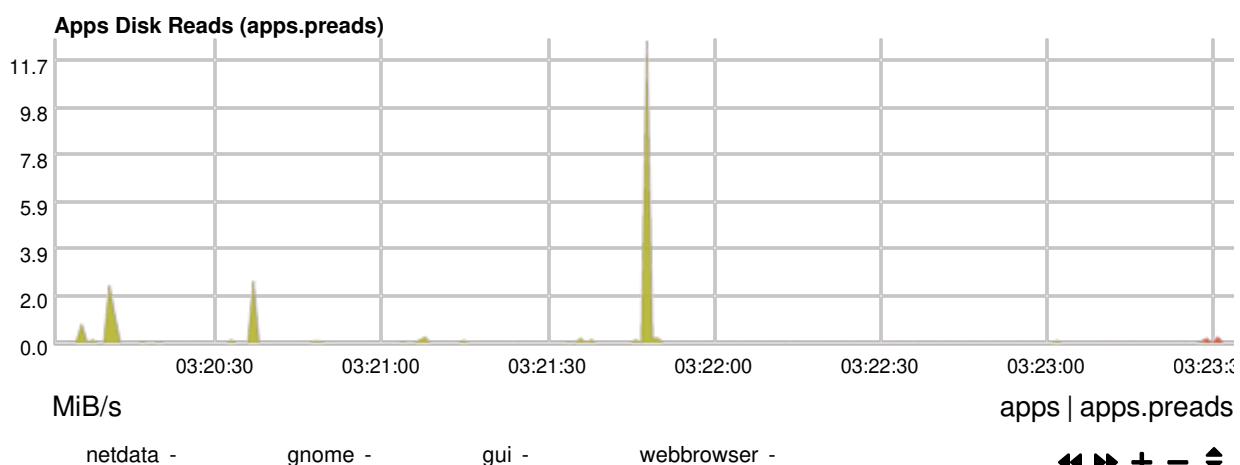


The amount of time the CPU was busy executing code in kernel mode (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) (all cores).

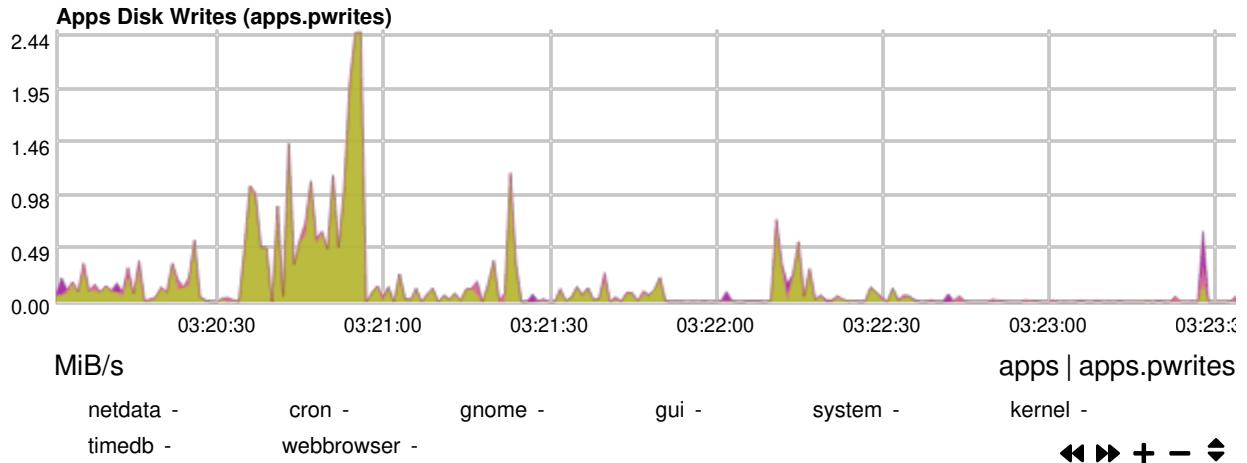


disk

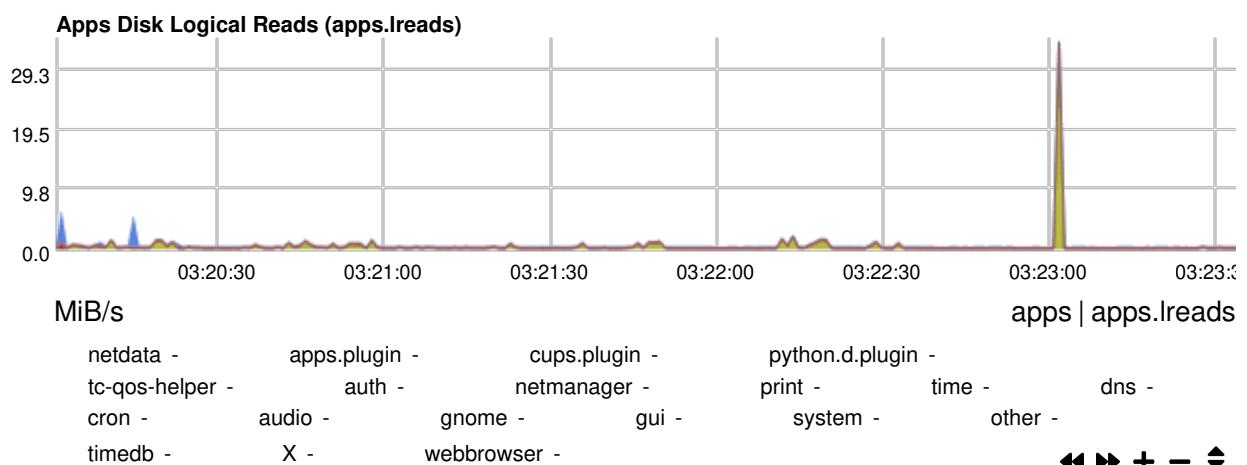
The amount of data that has been read from the storage layer. Actual physical disk I/O was required.



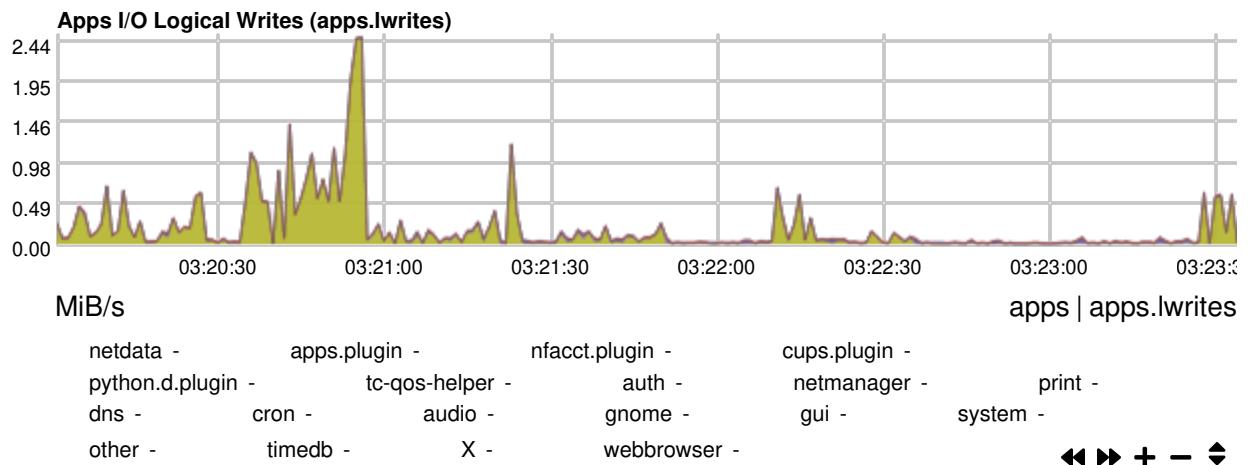
The amount of data that has been written to the storage layer. Actual physical disk I/O was required.



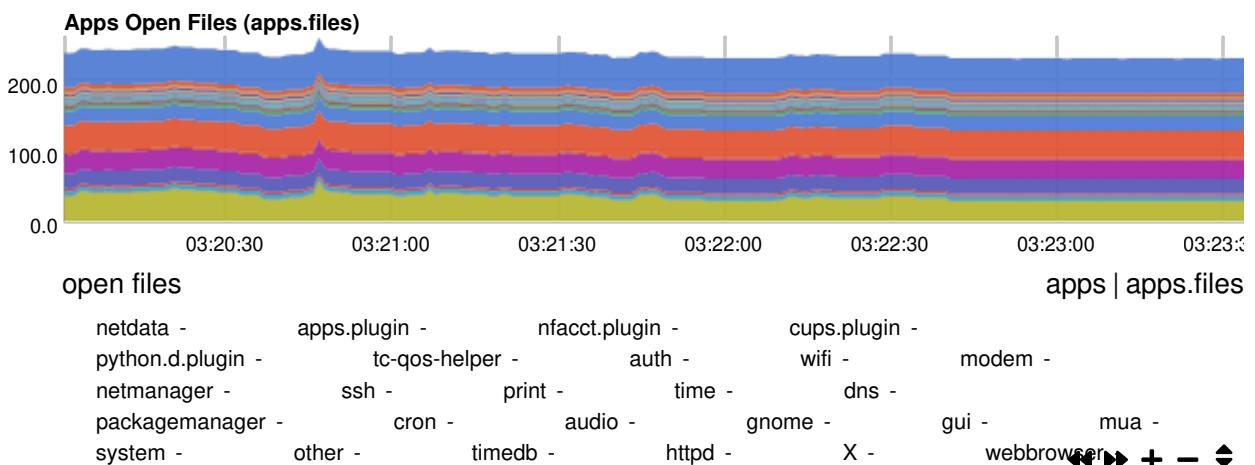
The amount of data that has been read from the storage layer. It includes things such as terminal I/O and is unaffected by whether or not actual physical disk I/O was required (the read might have been satisfied from pagecache).



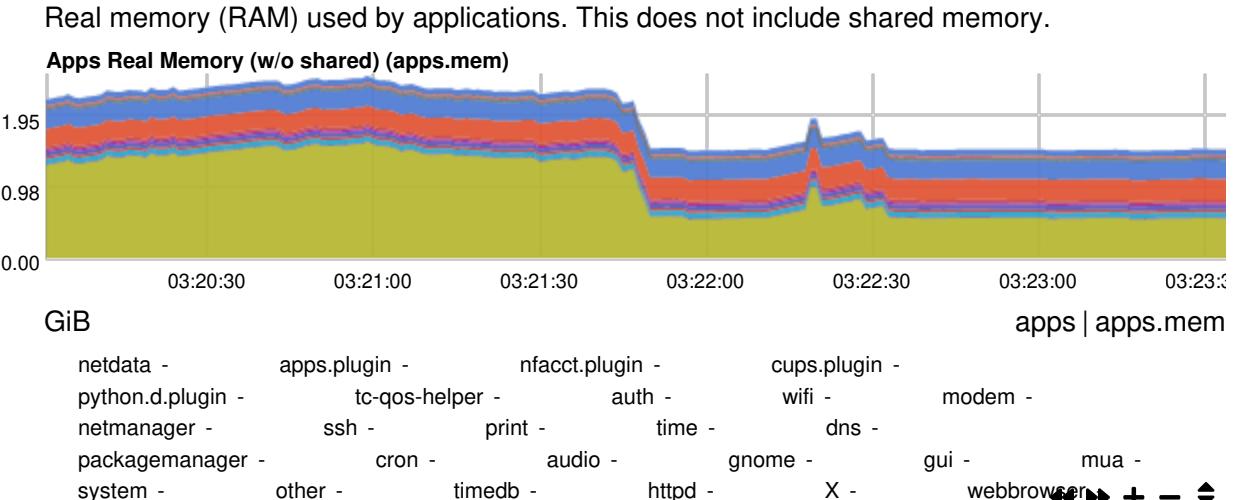
The amount of data that has been written or shall be written to the storage layer. It includes things such as terminal I/O and is unaffected by whether or not actual physical disk I/O was required.



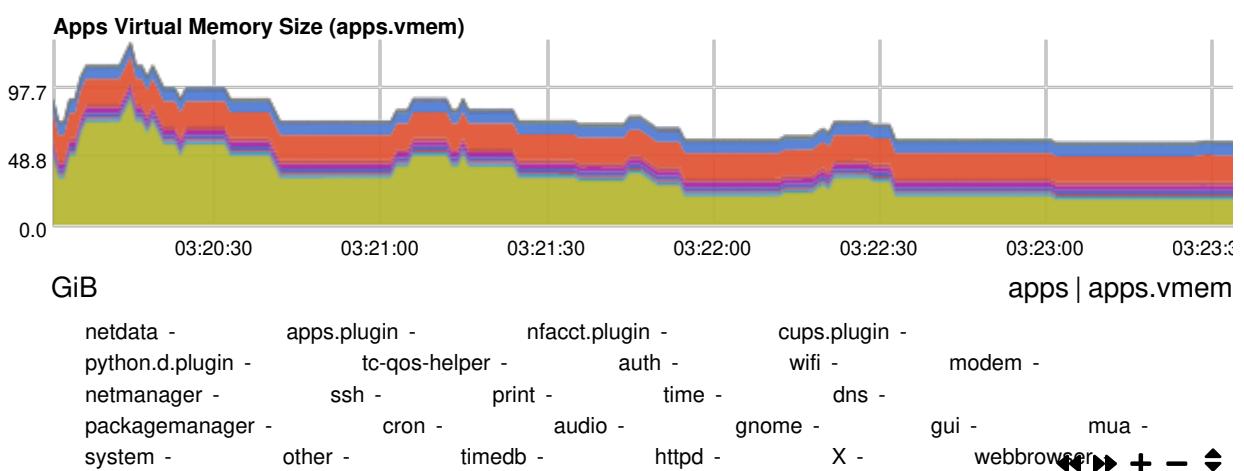
The number of open files and directories.



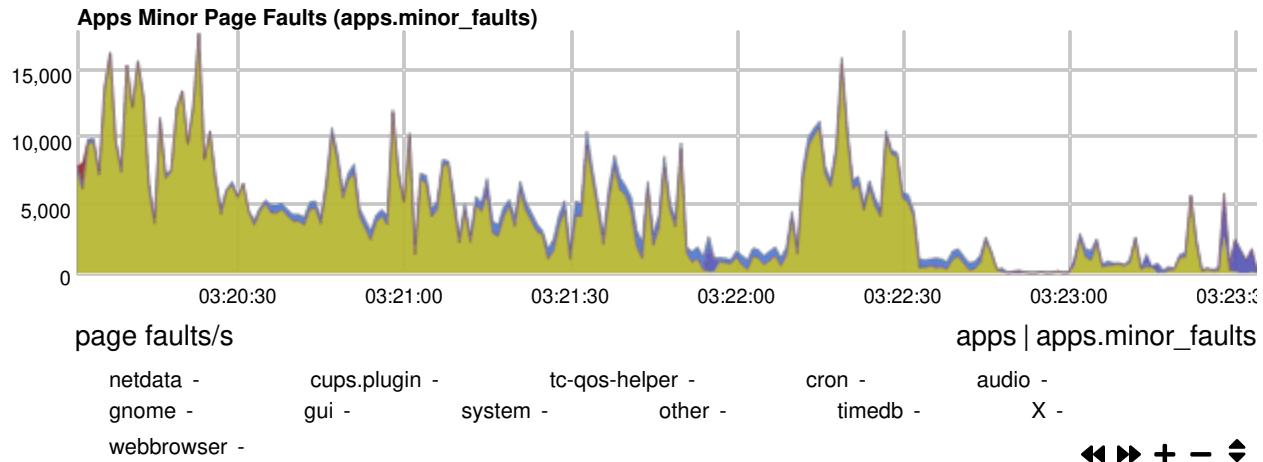
mem



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

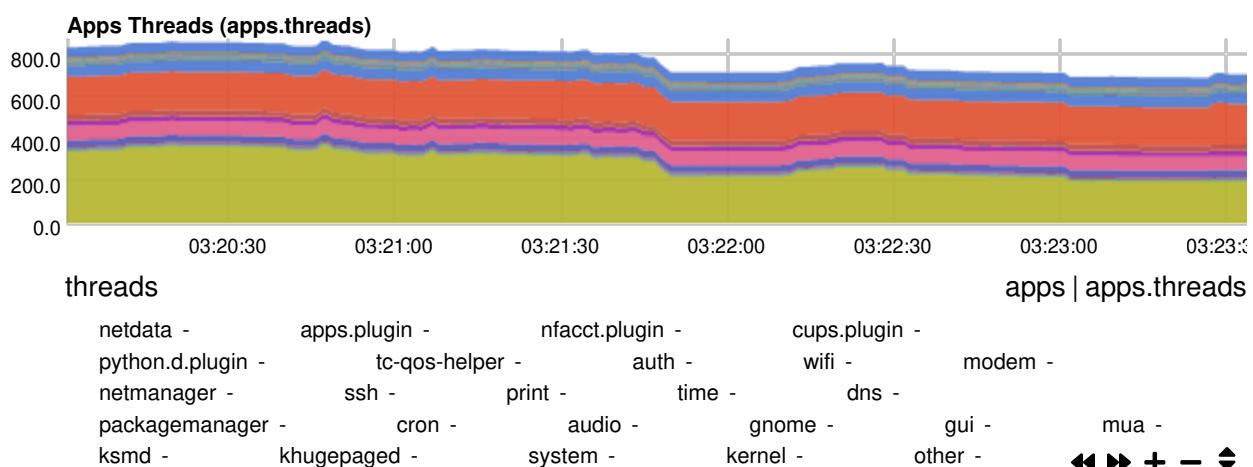


The number of minor faults (https://en.wikipedia.org/wiki/Page_fault#Minor) which have not required loading a memory page from the disk. Minor page faults occur when a process needs data that is in memory and is assigned to another process. They share memory pages between multiple processes – no additional data needs to be read from disk to memory.

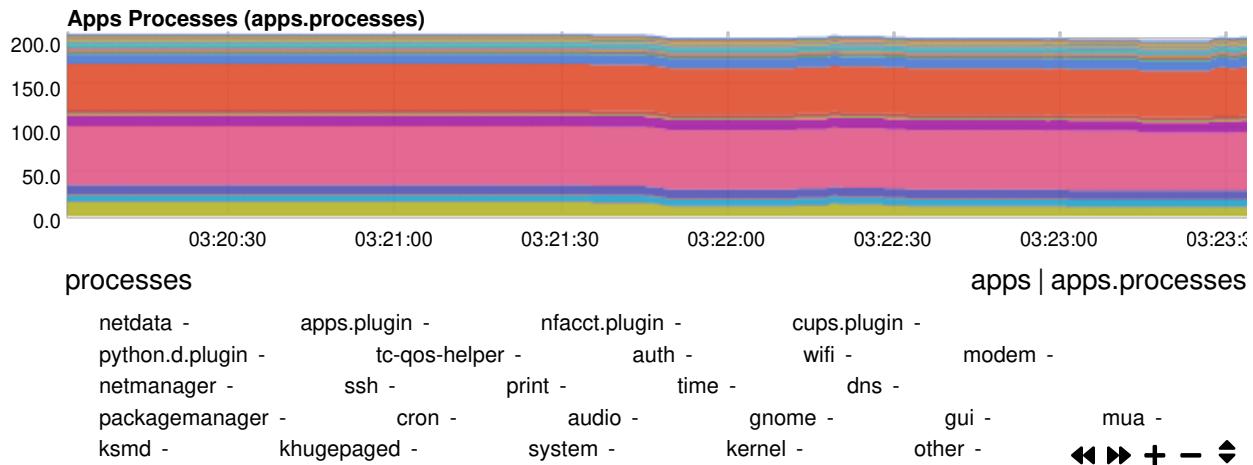


processes

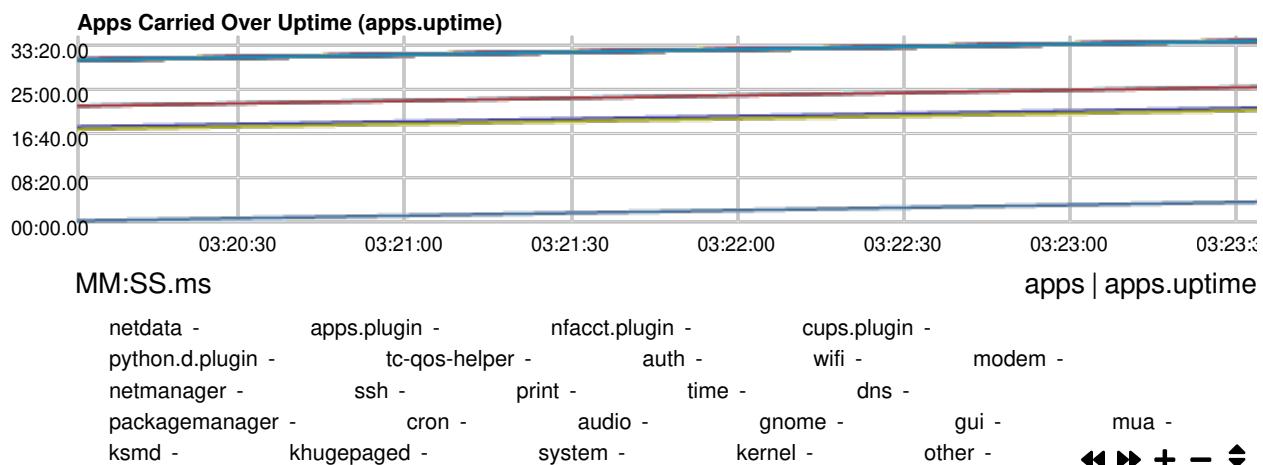
The number of threads ([https://en.wikipedia.org/wiki/Thread_\(computing\)](https://en.wikipedia.org/wiki/Thread_(computing))).



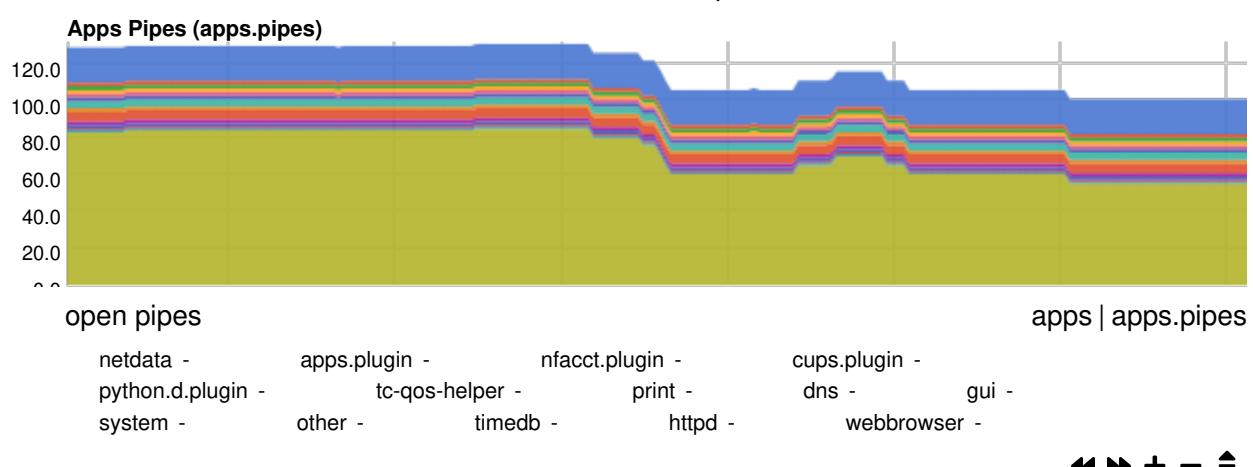
The number of processes ([https://en.wikipedia.org/wiki/Process_\(computing\)](https://en.wikipedia.org/wiki/Process_(computing))).



The period of time within which at least one process in the group has been running.

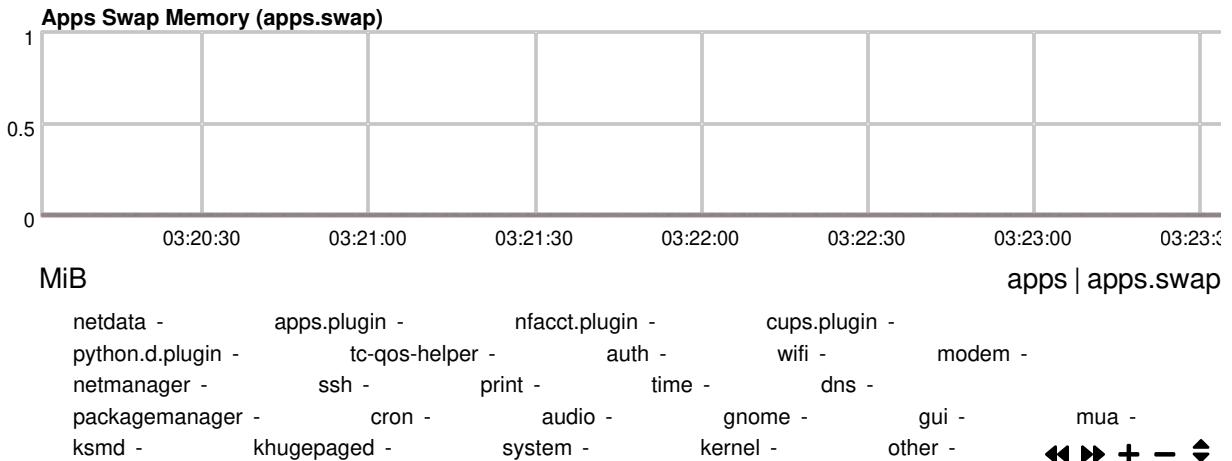


The number of open pipes (https://en.wikipedia.org/wiki/Anonymous_pipe#Unix). A pipe is a unidirectional data channel that can be used for interprocess communication.

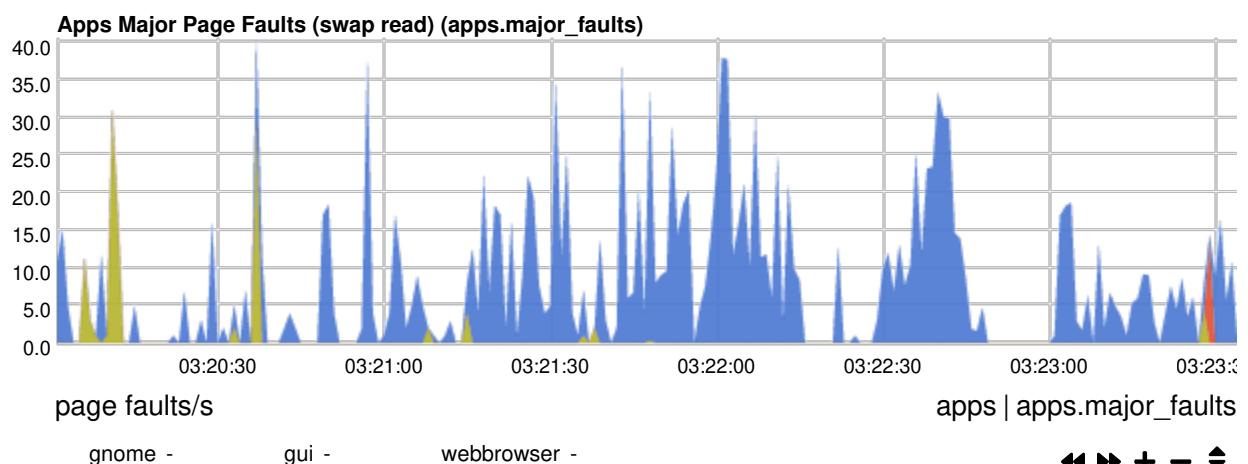


Swap

The amount of swapped-out virtual memory by anonymous private pages. This does not include shared swap memory.



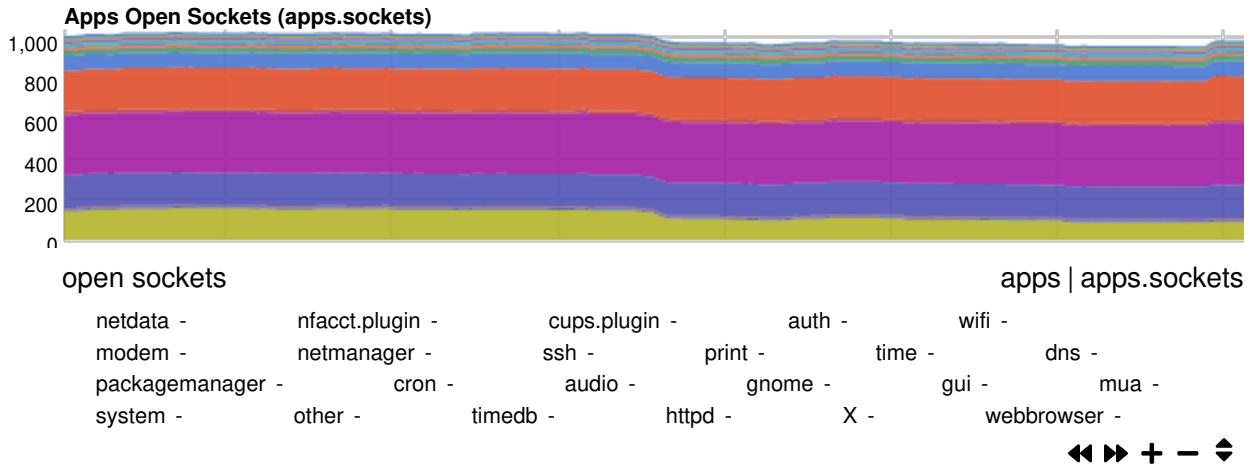
The number of major faults (https://en.wikipedia.org/wiki/Page_fault#Major) which have required loading a memory page from the disk. Major page faults occur because of the absence of the required page from the RAM. They are expected when a process starts or needs to read in additional data and in these cases do not indicate a problem condition. However, a major page fault can also be the result of reading memory pages that have been written out to the swap file, which could indicate a memory shortage.



network

Netdata also gives a summary for eBPF charts in Networking Stack submenu.

The number of open sockets. Sockets are a way to enable inter-process communication between programs running on a server, or between programs running on separate servers. This includes both network and UNIX sockets.

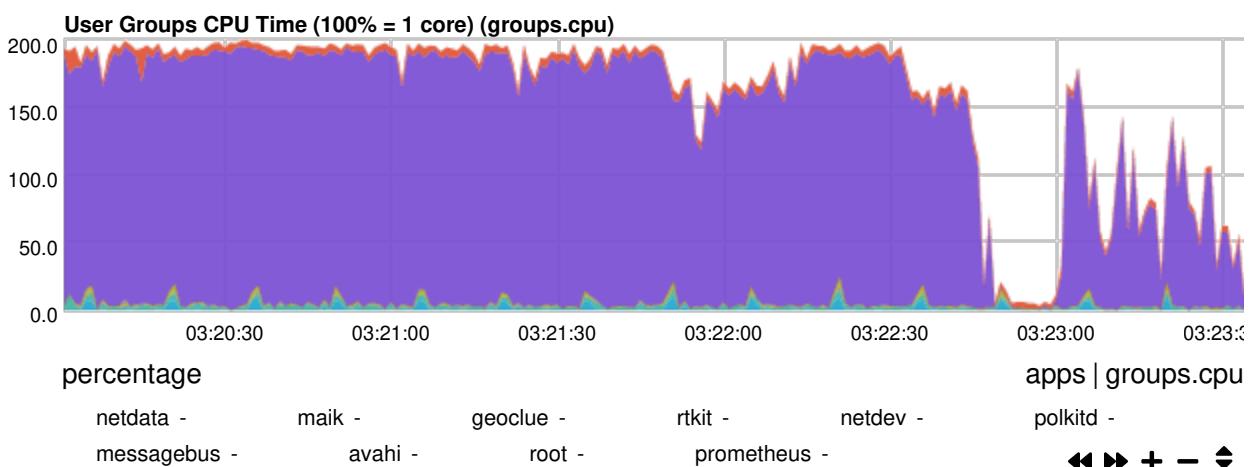


User Groups

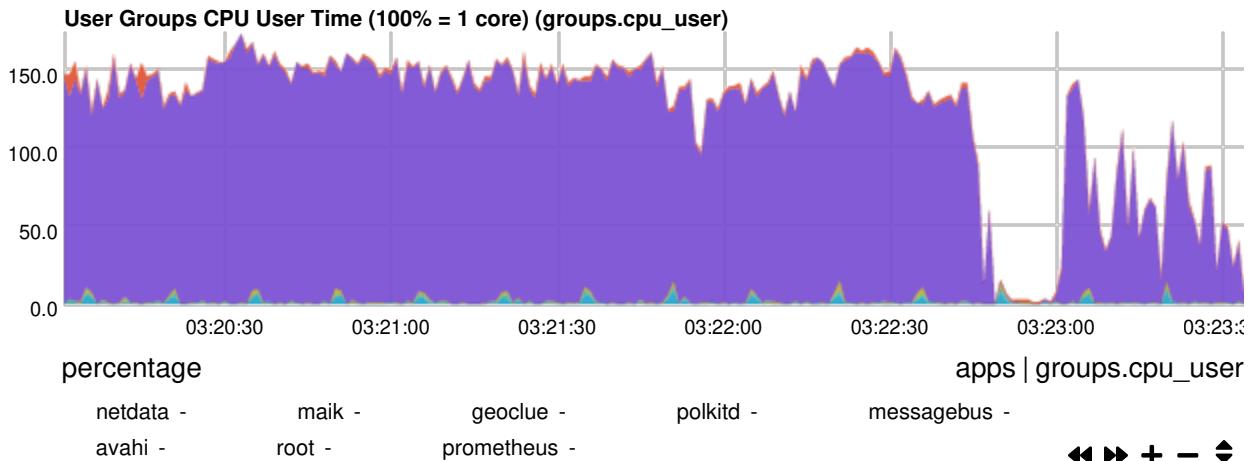
Per user group statistics are collected using `apps.plugin` (<https://learn.netdata.cloud/docs/agent/collectors/apps.plugin>). This plugin walks through all processes and aggregates statistics per user group. The plugin also counts the resources of exited children. So for processes like shell scripts, the reported values include the resources used by the commands these scripts run within each timeframe.

CPU

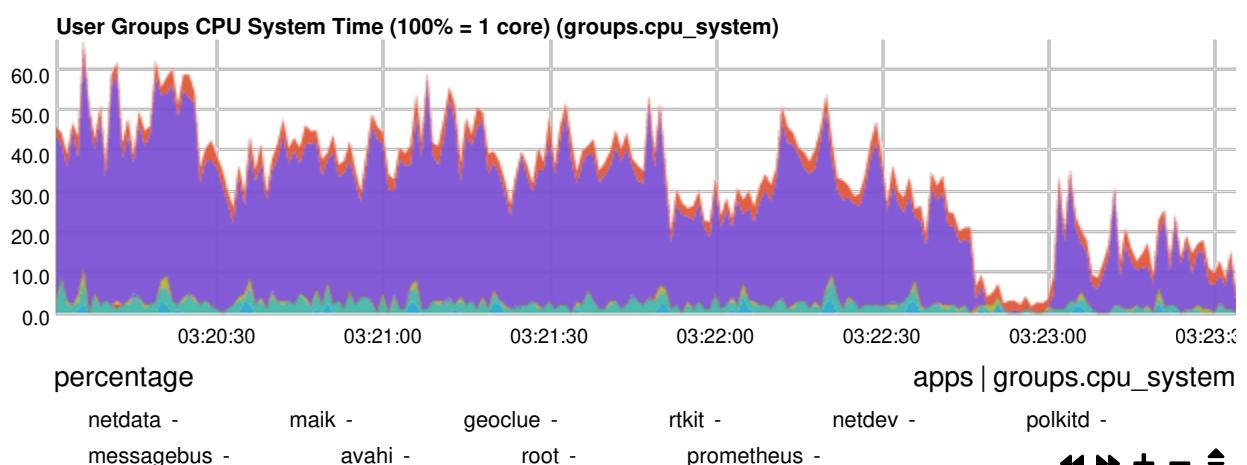
Total CPU utilization (all cores). It includes user, system and guest time.



The amount of time the CPU was busy executing code in user mode (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) (all cores).

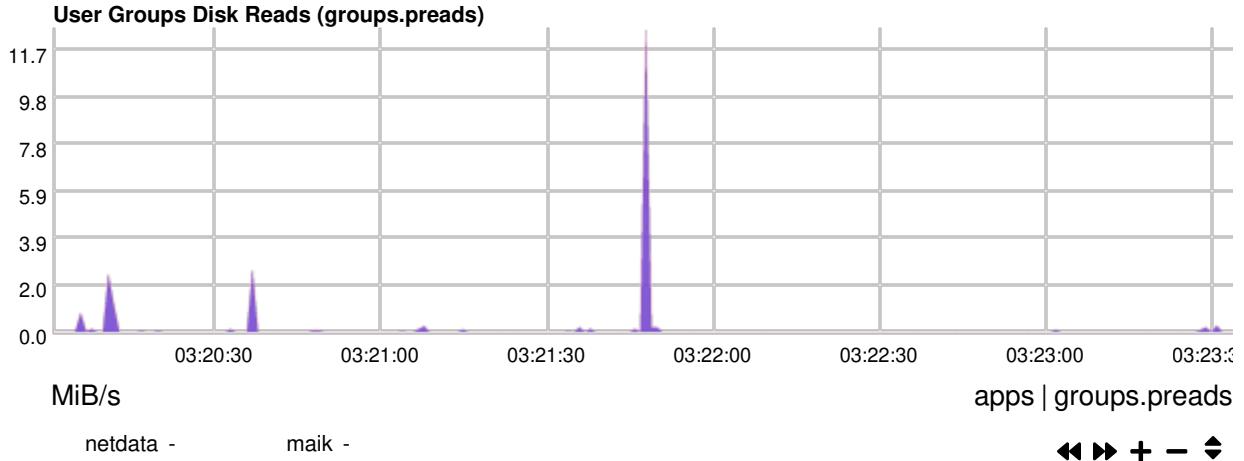


The amount of time the CPU was busy executing code in kernel mode (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) (all cores).

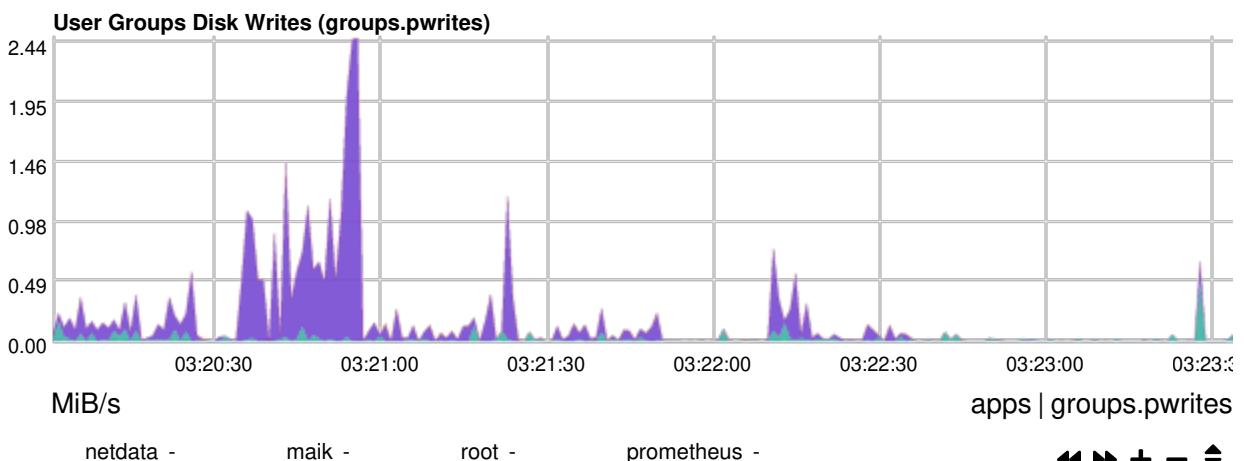


disk

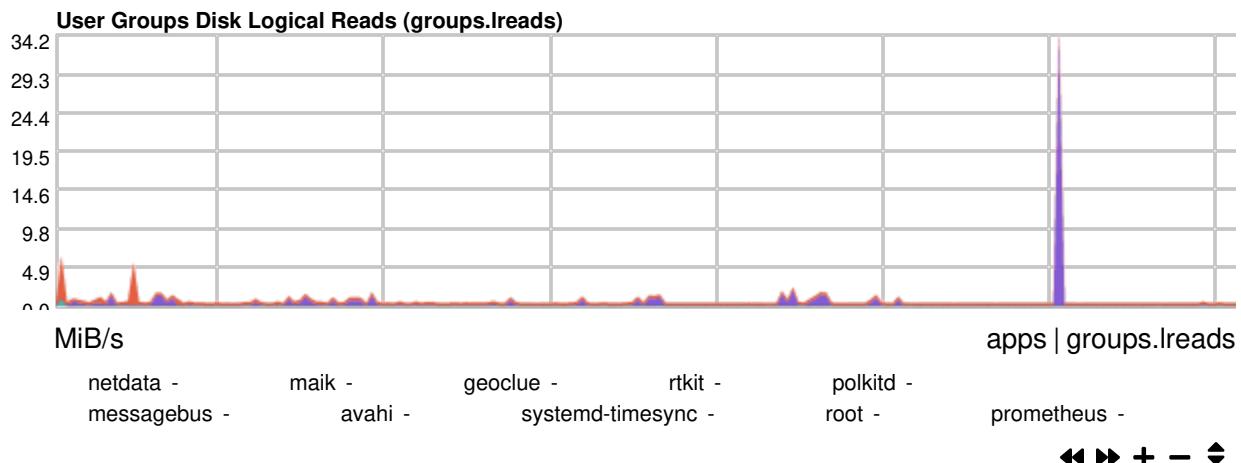
The amount of data that has been read from the storage layer. Actual physical disk I/O was required.



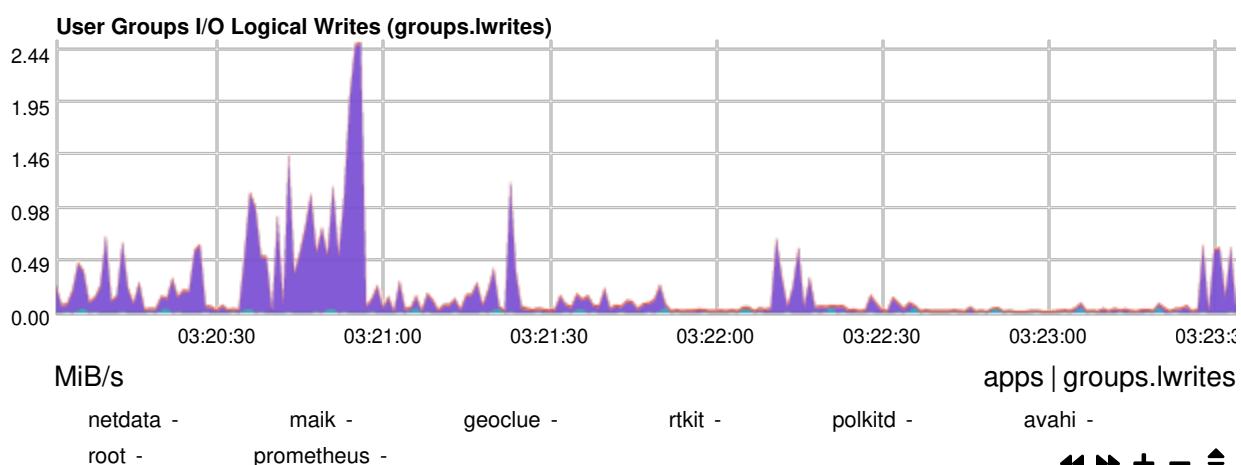
The amount of data that has been written to the storage layer. Actual physical disk I/O was required.



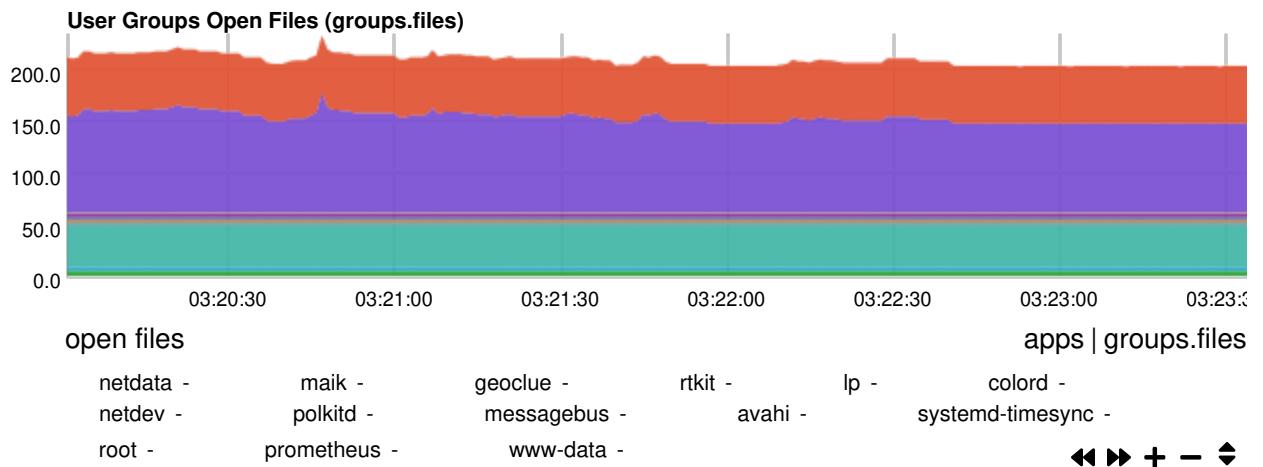
The amount of data that has been read from the storage layer. It includes things such as terminal I/O and is unaffected by whether or not actual physical disk I/O was required (the read might have been satisfied from pagecache).



The amount of data that has been written or shall be written to the storage layer. It includes things such as terminal I/O and is unaffected by whether or not actual physical disk I/O was required.

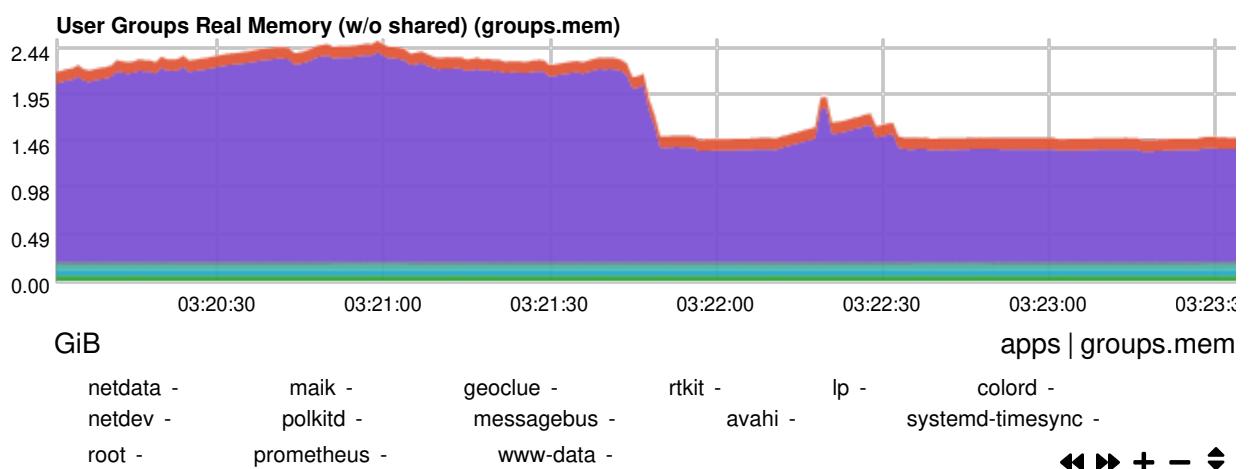


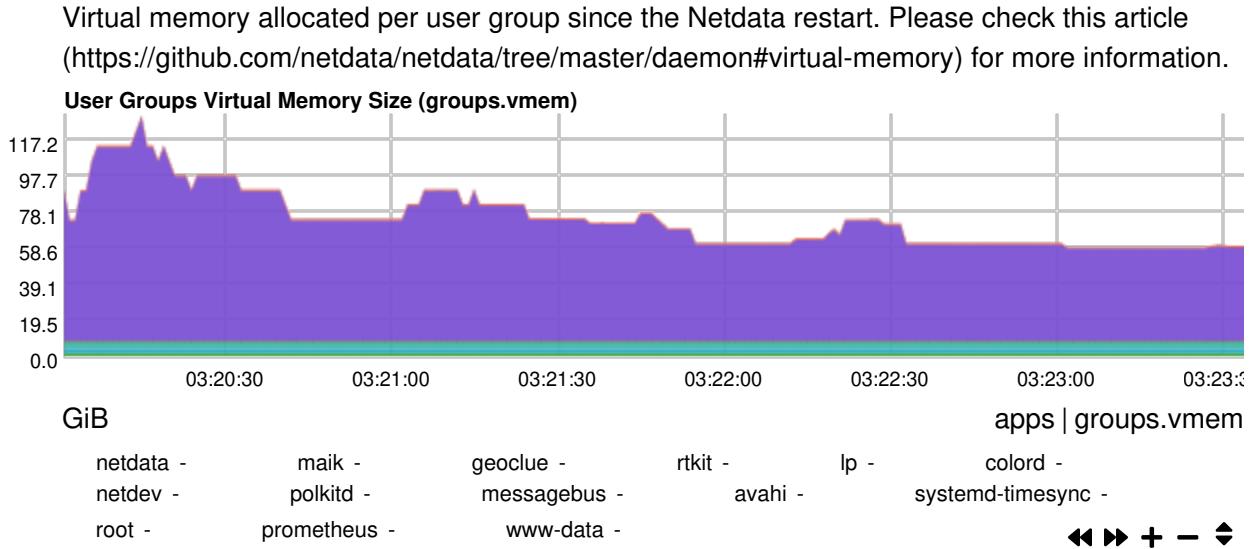
The number of open files and directories.



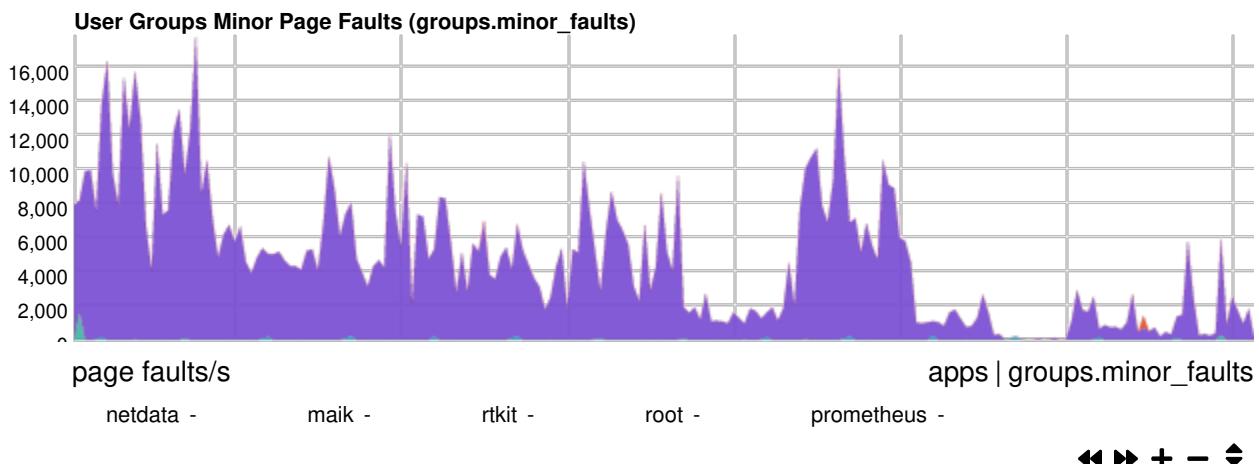
mem

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



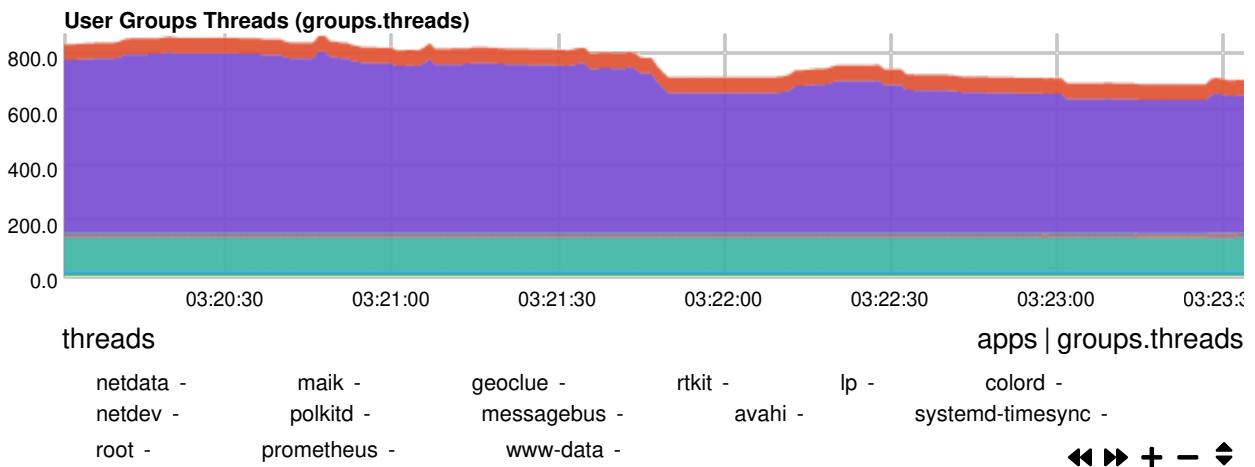


The number of minor faults (https://en.wikipedia.org/wiki/Page_fault#Minor) which have not required loading a memory page from the disk. Minor page faults occur when a process needs data that is in memory and is assigned to another process. They share memory pages between multiple processes – no additional data needs to be read from disk to memory.

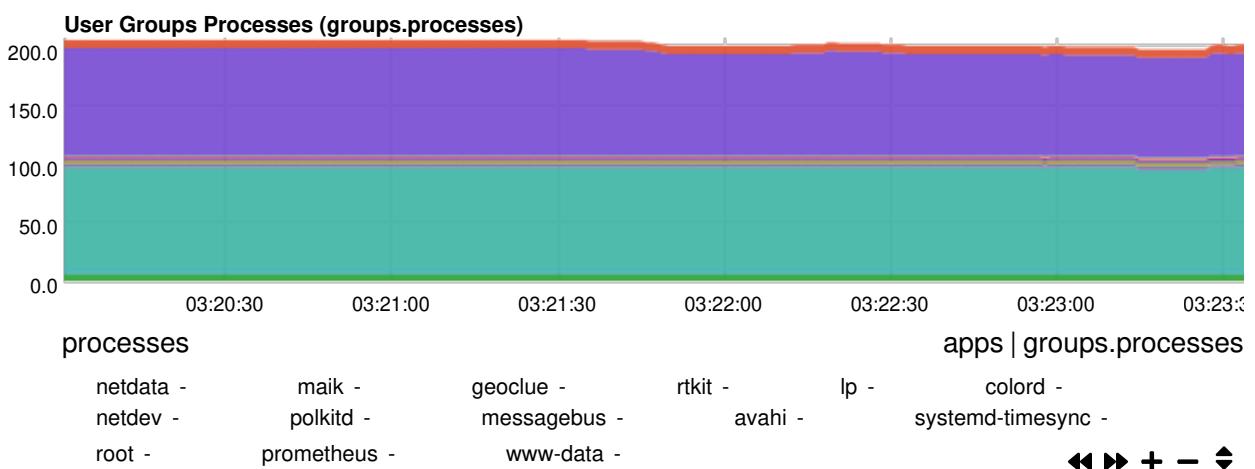


processes

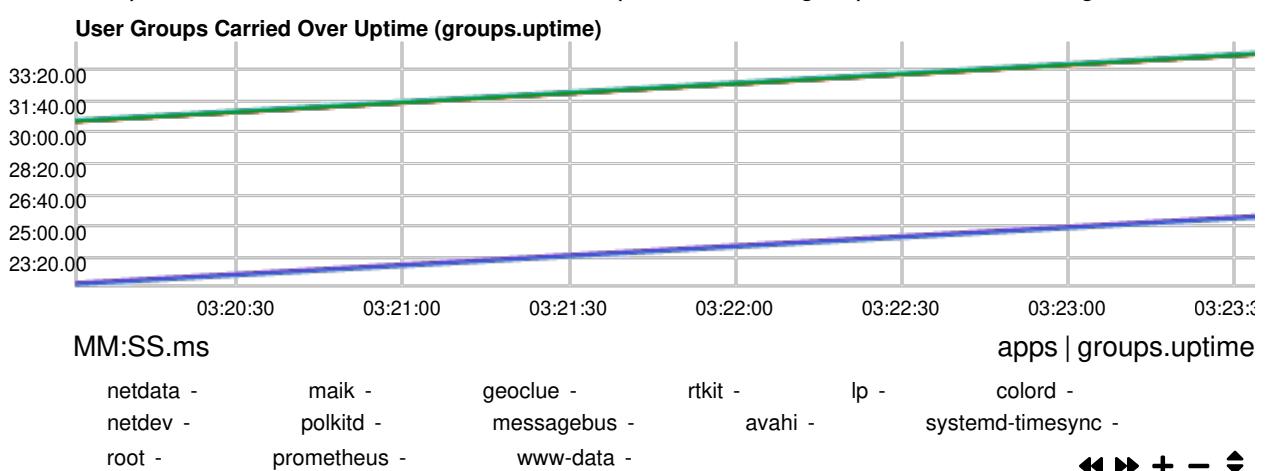
The number of threads ([https://en.wikipedia.org/wiki/Thread_\(computing\)](https://en.wikipedia.org/wiki/Thread_(computing))).



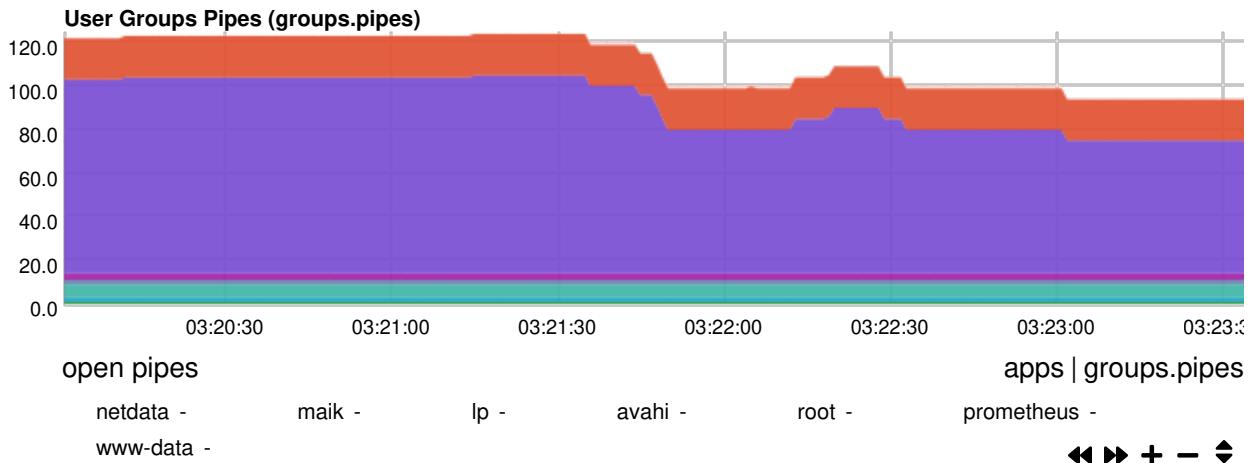
The number of processes ([https://en.wikipedia.org/wiki/Process_\(computing\)](https://en.wikipedia.org/wiki/Process_(computing))).



The period of time within which at least one process in the group has been running.

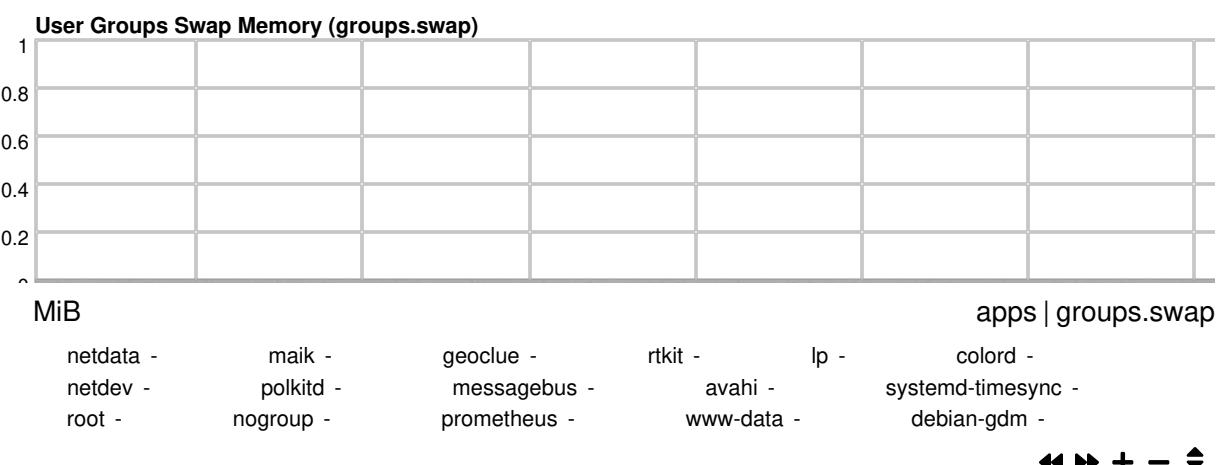


The number of open pipes (https://en.wikipedia.org/wiki/Anonymous_pipe#Unix). A pipe is a unidirectional data channel that can be used for interprocess communication.

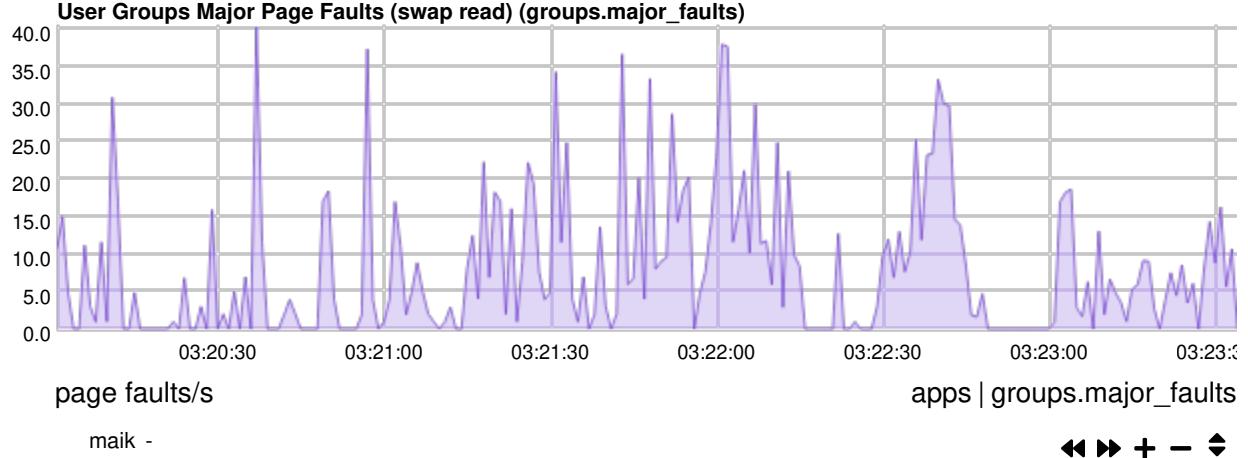


swap

The amount of swapped-out virtual memory by anonymous private pages. This does not include shared swap memory.

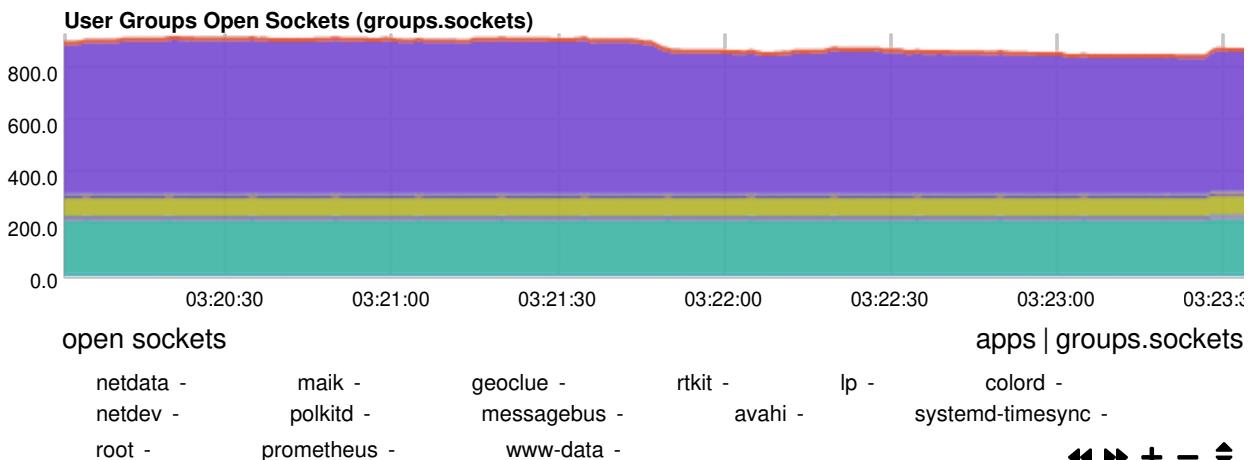


The number of major faults (https://en.wikipedia.org/wiki/Page_fault#Major) which have required loading a memory page from the disk. Major page faults occur because of the absence of the required page from the RAM. They are expected when a process starts or needs to read in additional data and in these cases do not indicate a problem condition. However, a major page fault can also be the result of reading memory pages that have been written out to the swap file, which could indicate a memory shortage.



net

The number of open sockets. Sockets are a way to enable inter-process communication between programs running on a server, or between programs running on separate servers. This includes both network and UNIX sockets.



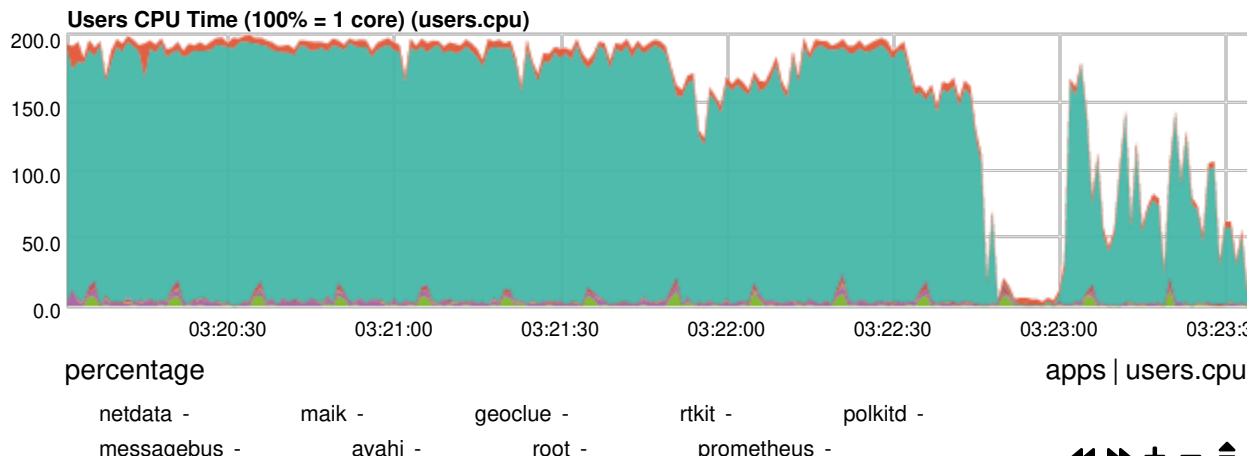
Users

Per user statistics are collected using apps.plugin (<https://learn.netdata.cloud/docs/agent/collectors/apps.plugin>). This plugin walks through all processes and aggregates statistics per user. The plugin also counts the resources of exited children. So for processes like shell scripts, the reported values

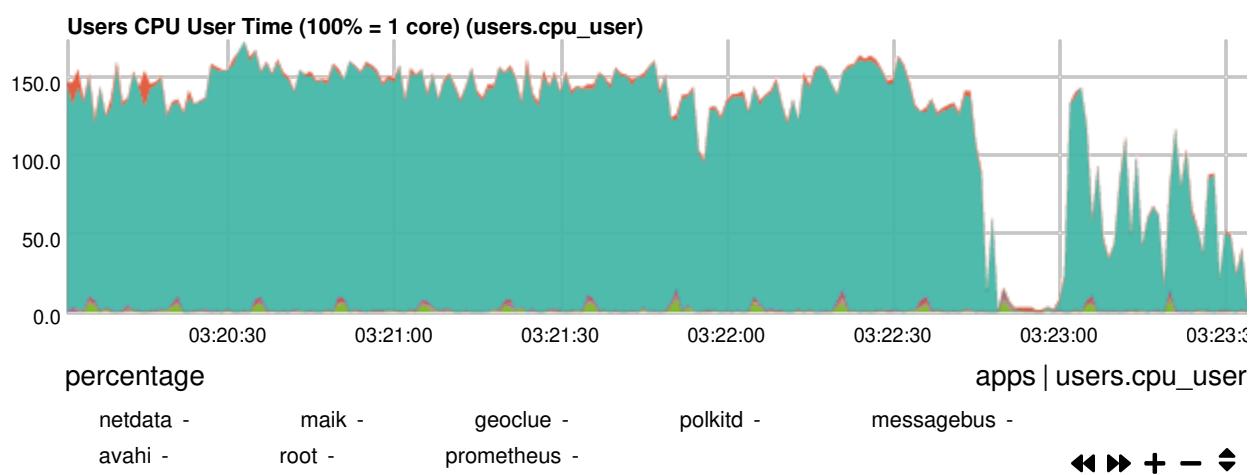
include the resources used by the commands these scripts run within each timeframe.

cpu

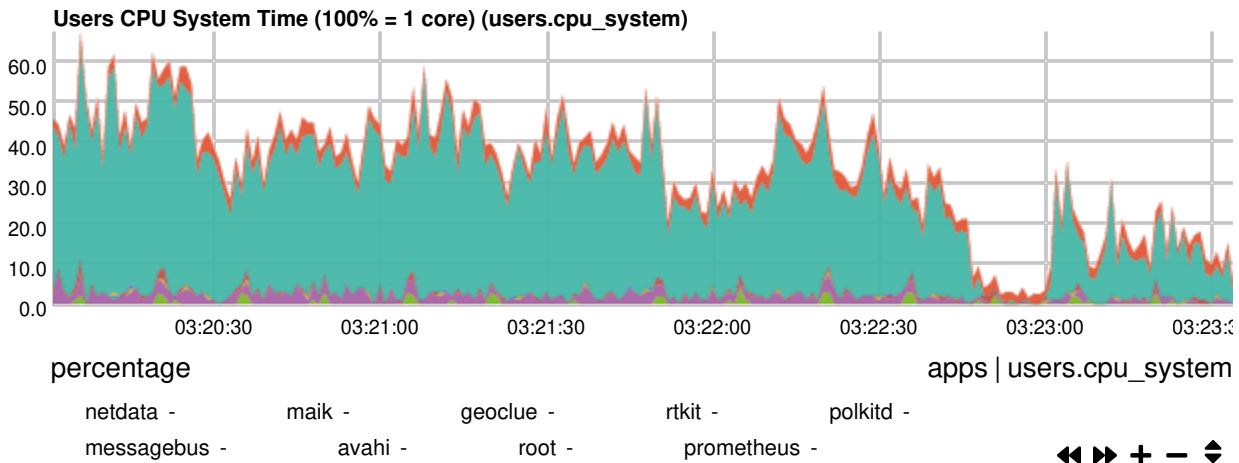
Total CPU utilization (all cores). It includes user, system and guest time.



The amount of time the CPU was busy executing code in user mode (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) (all cores).

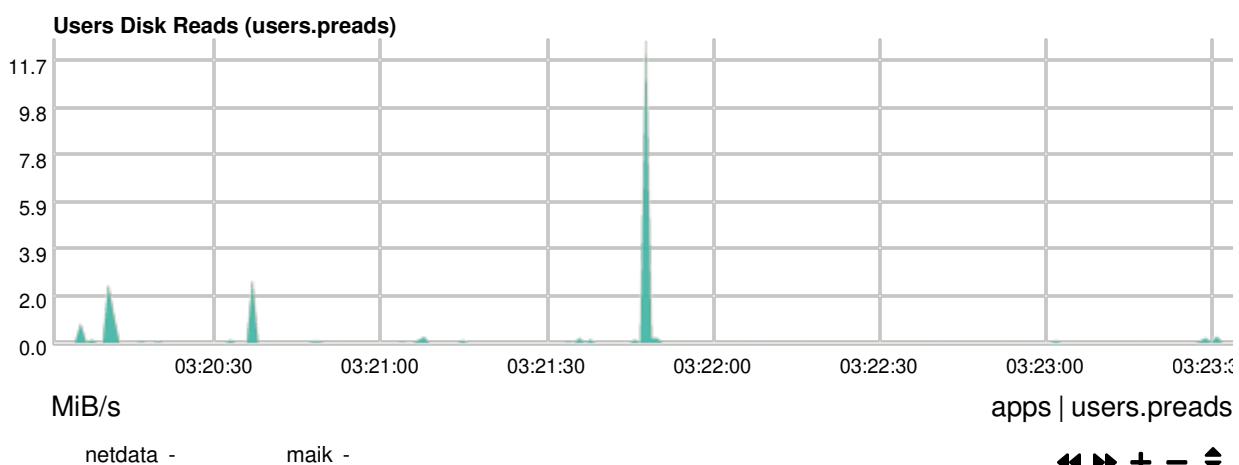


The amount of time the CPU was busy executing code in kernel mode (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) (all cores).

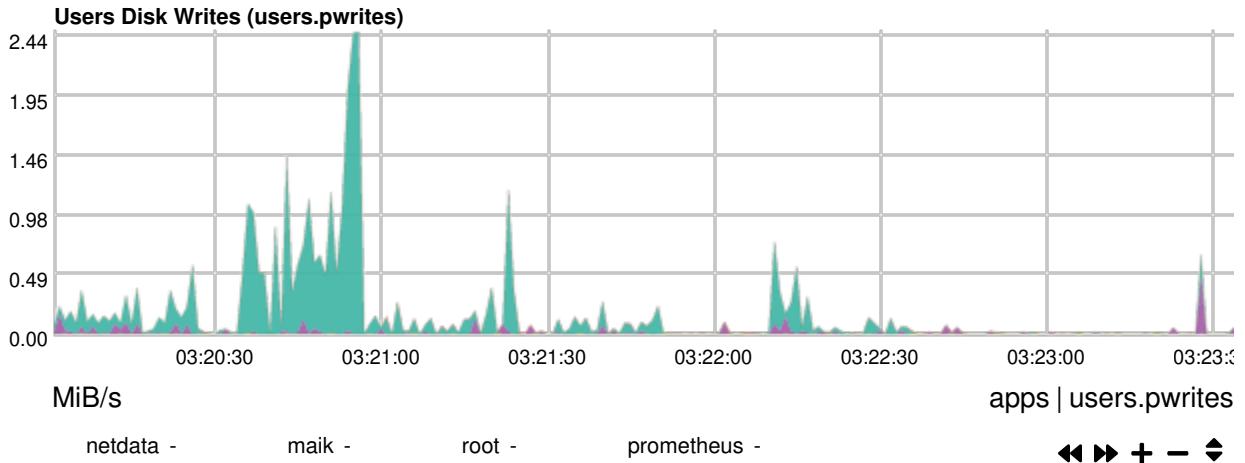


disk

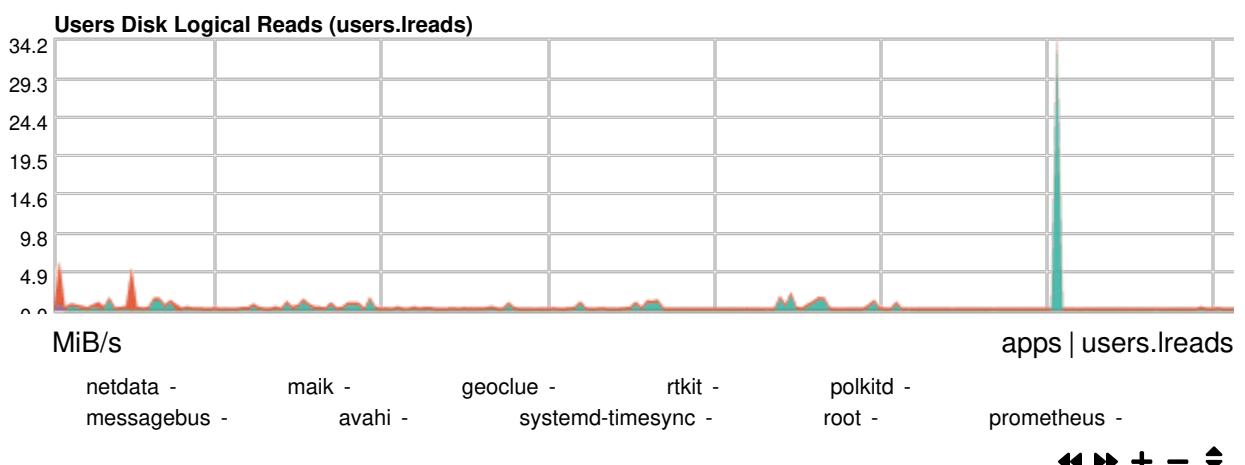
The amount of data that has been read from the storage layer. Actual physical disk I/O was required.



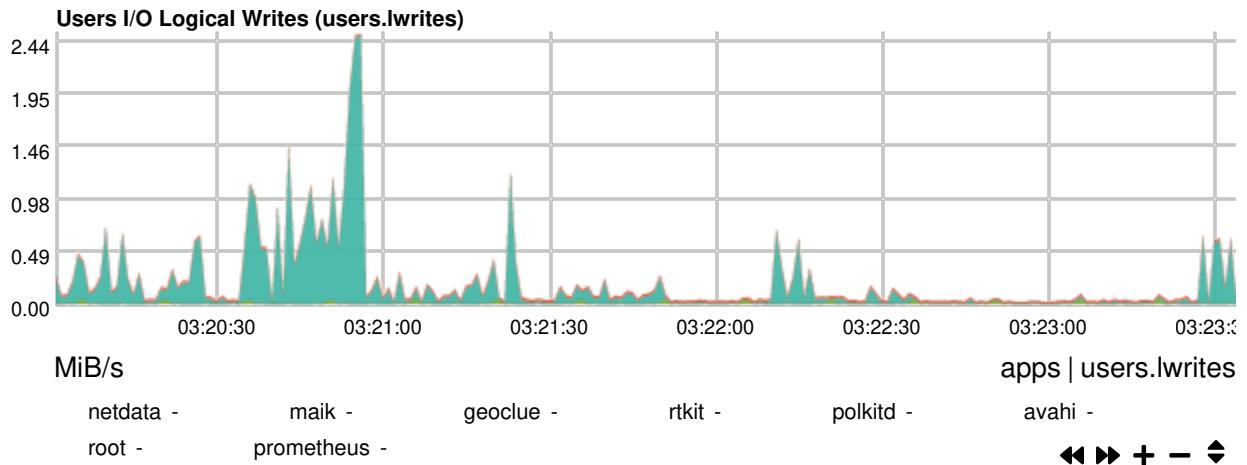
The amount of data that has been written to the storage layer. Actual physical disk I/O was required.



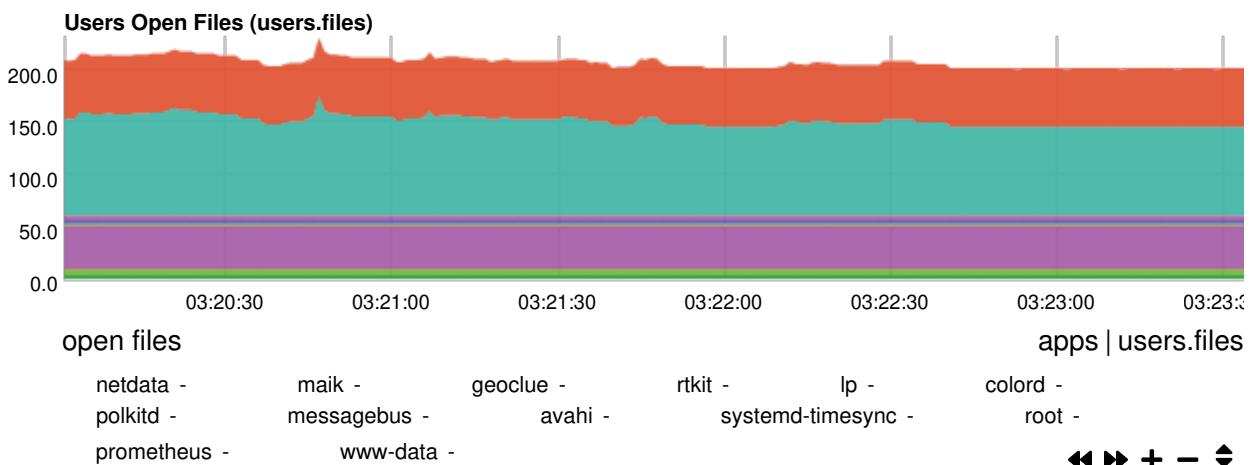
The amount of data that has been read from the storage layer. It includes things such as terminal I/O and is unaffected by whether or not actual physical disk I/O was required (the read might have been satisfied from pagecache).



The amount of data that has been written or shall be written to the storage layer. It includes things such as terminal I/O and is unaffected by whether or not actual physical disk I/O was required.

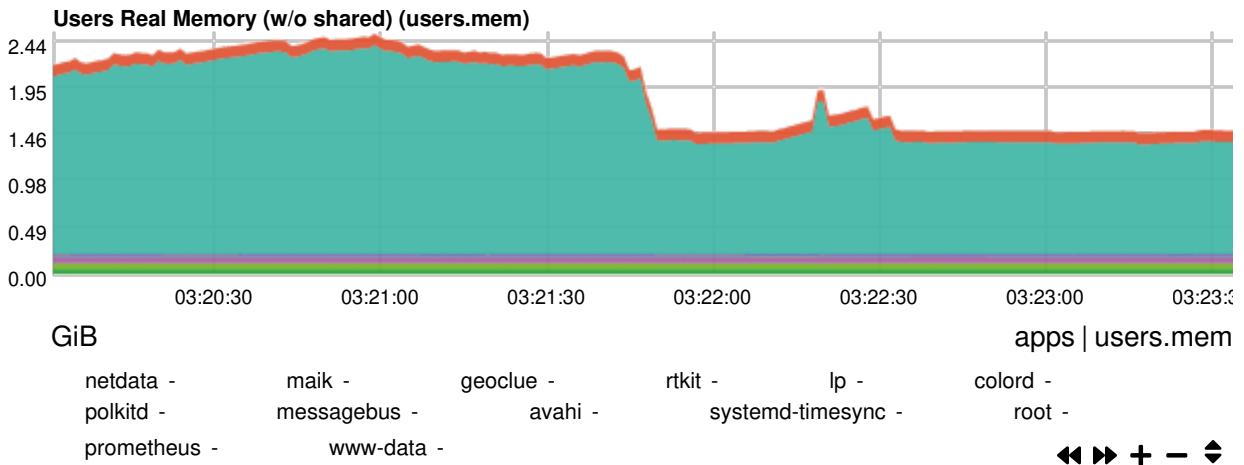


The number of open files and directories.



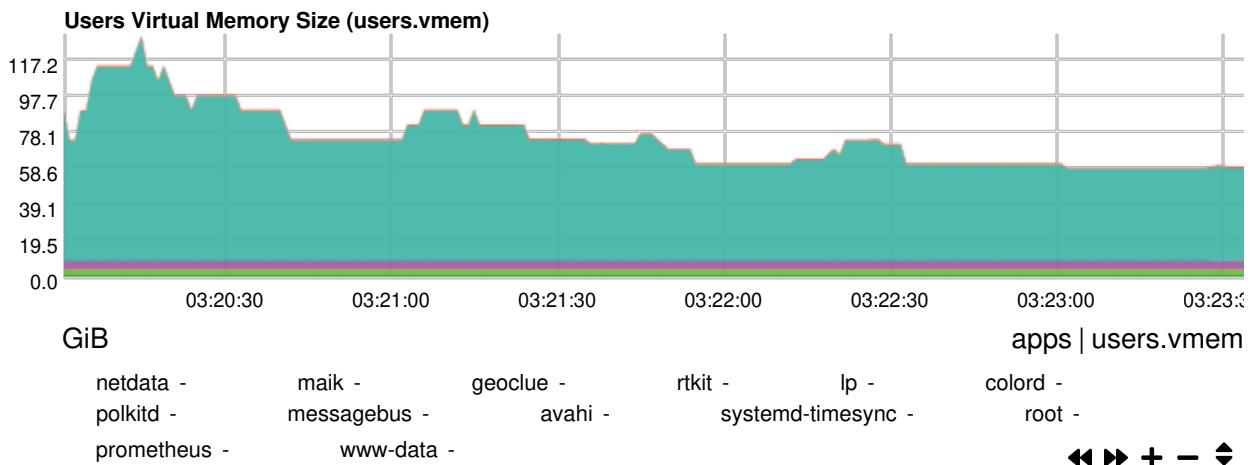
mem

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

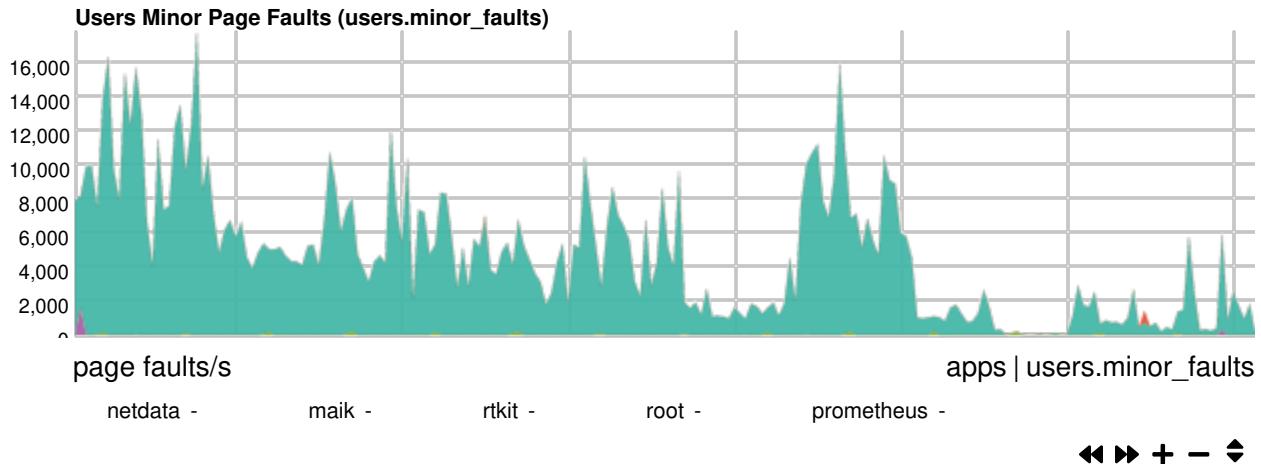


Virtual memory allocated per user since the Netdata restart. Please check this article

(<https://github.com/netdata/netdata/tree/master/daemon#virtual-memory>) for more information.

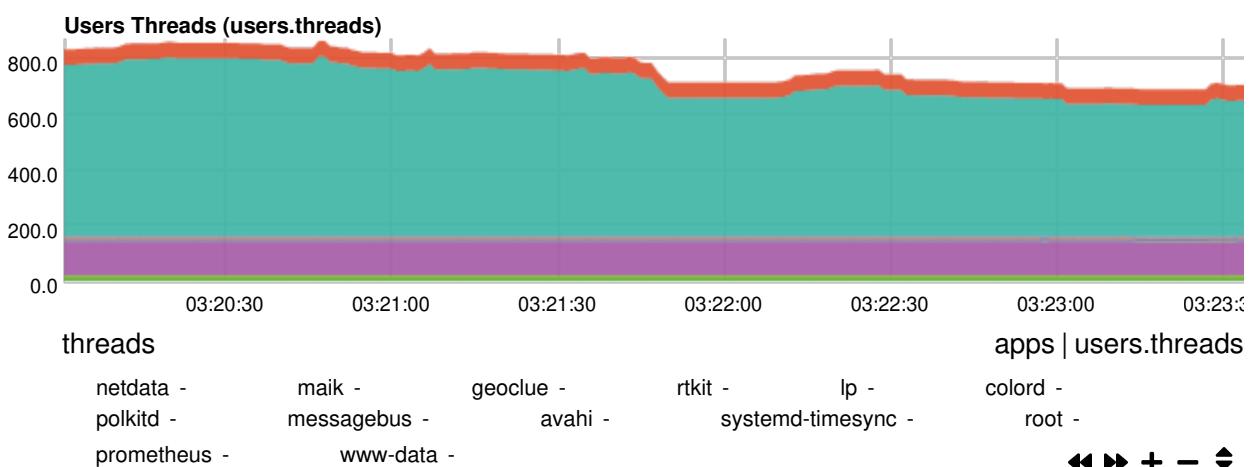


The number of minor faults (https://en.wikipedia.org/wiki/Page_fault#Minor) which have not required loading a memory page from the disk. Minor page faults occur when a process needs data that is in memory and is assigned to another process. They share memory pages between multiple processes – no additional data needs to be read from disk to memory.

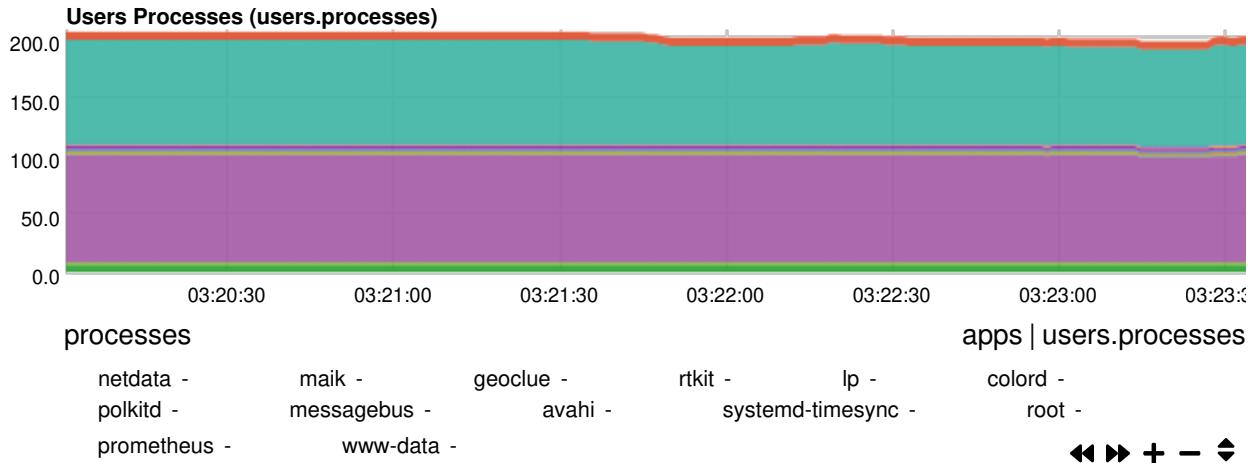


processes

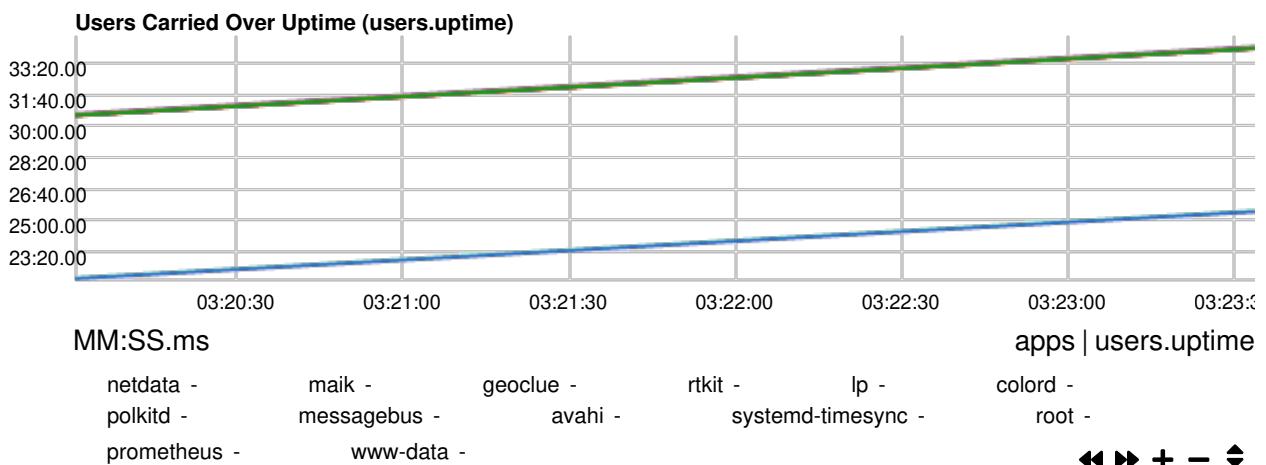
The number of threads ([https://en.wikipedia.org/wiki/Thread_\(computing\)](https://en.wikipedia.org/wiki/Thread_(computing))).



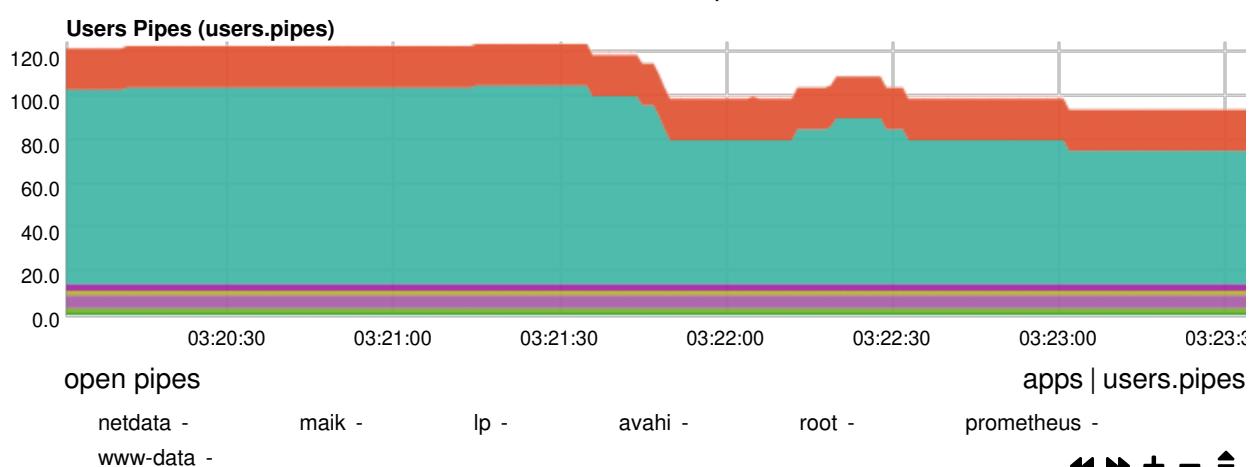
The number of processes ([https://en.wikipedia.org/wiki/Process_\(computing\)](https://en.wikipedia.org/wiki/Process_(computing))).



The period of time within which at least one process in the group has been running.

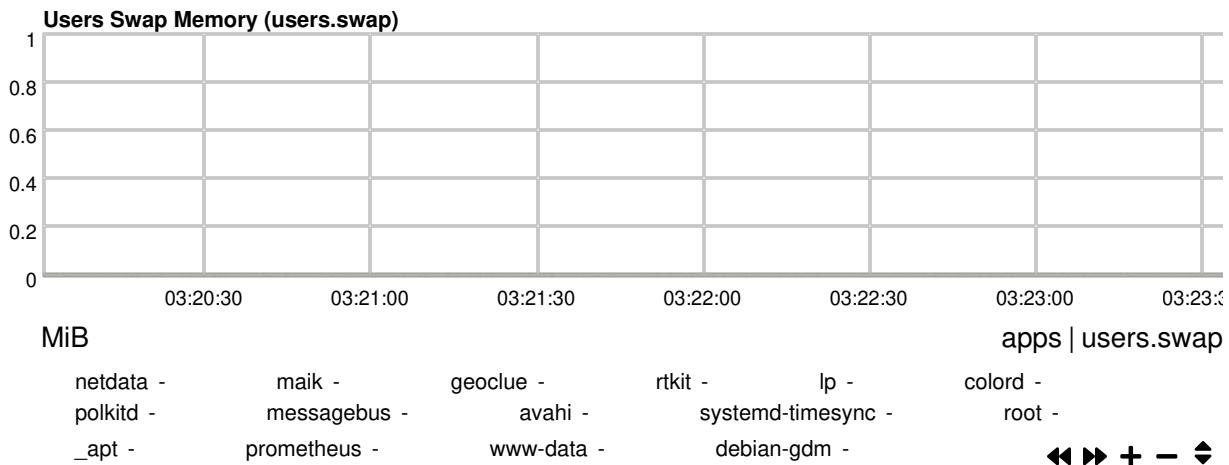


The number of open pipes (https://en.wikipedia.org/wiki/Anonymous_pipe#Unix). A pipe is a unidirectional data channel that can be used for interprocess communication.

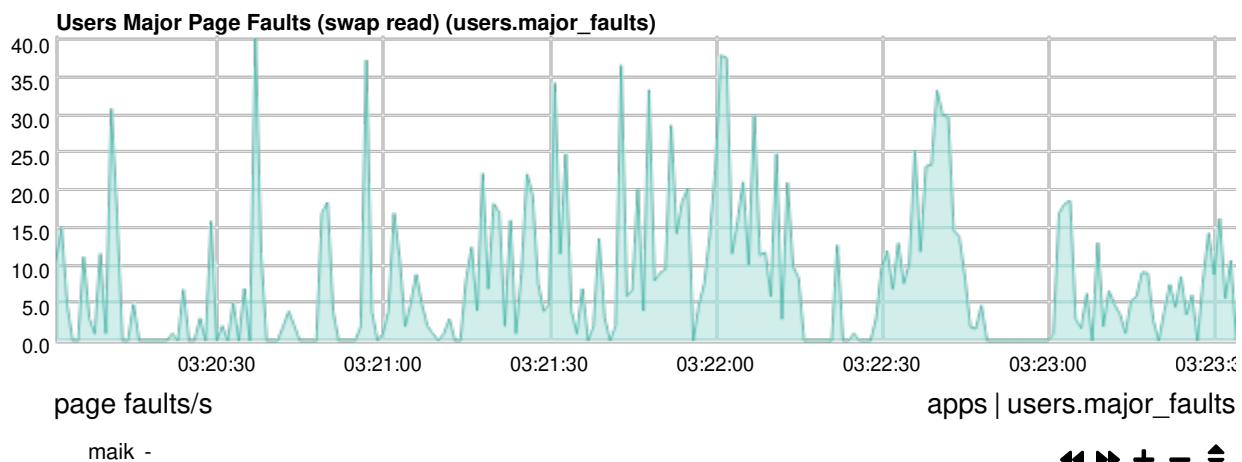


Swap

The amount of swapped-out virtual memory by anonymous private pages. This does not include shared swap memory.

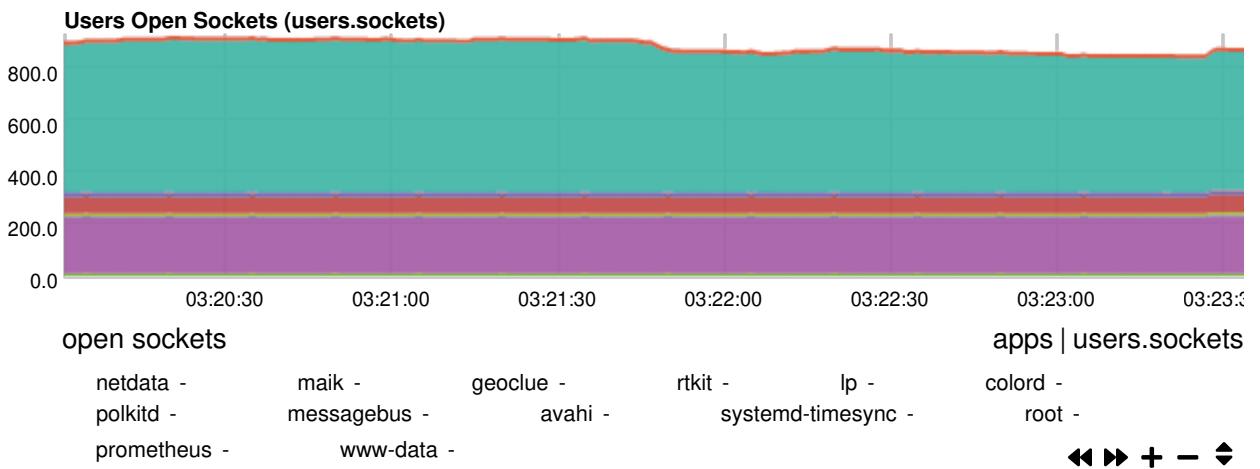


The number of major faults (https://en.wikipedia.org/wiki/Page_fault#Major) which have required loading a memory page from the disk. Major page faults occur because of the absence of the required page from the RAM. They are expected when a process starts or needs to read in additional data and in these cases do not indicate a problem condition. However, a major page fault can also be the result of reading memory pages that have been written out to the swap file, which could indicate a memory shortage.



net

The number of open sockets. Sockets are a way to enable inter-process communication between programs running on a server, or between programs running on separate servers. This includes both network and UNIX sockets.

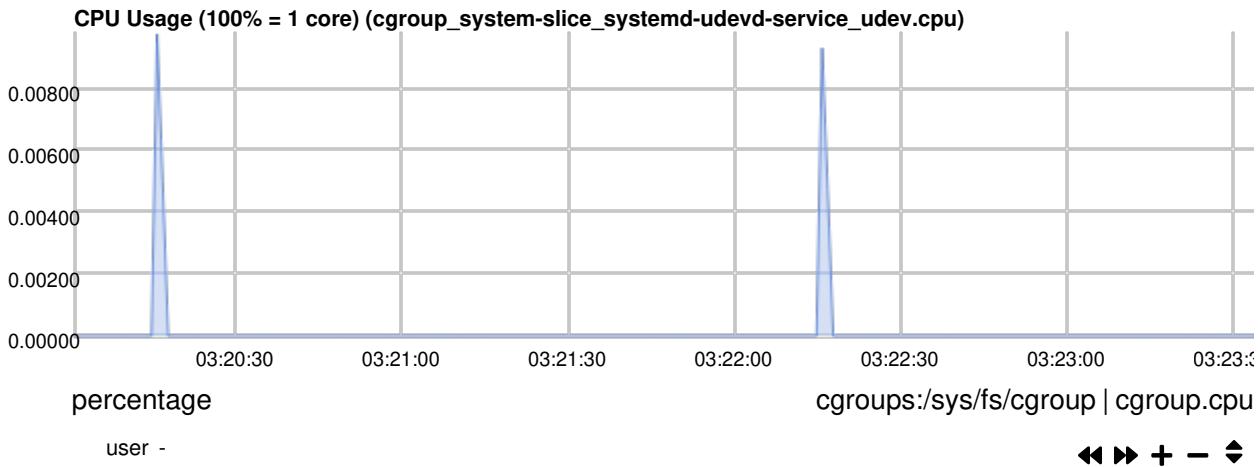


■■■ system-slice systemd-udevd-service udev

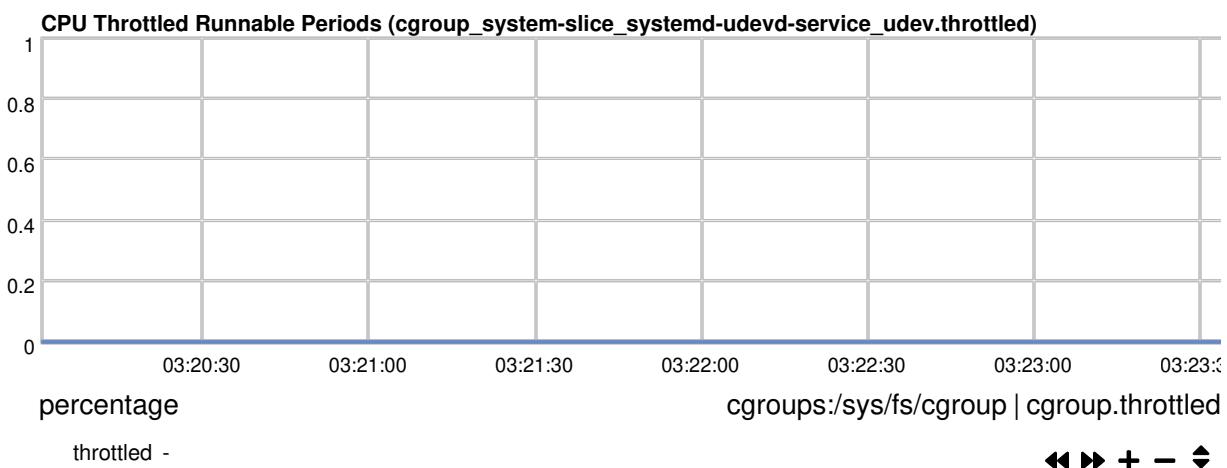
Container resource utilization metrics. Netdata reads this information from **cgroups** (abbreviated from **control groups**), a Linux kernel feature that limits and accounts resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes. **cgroups** together with **namespaces** (that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

cpu

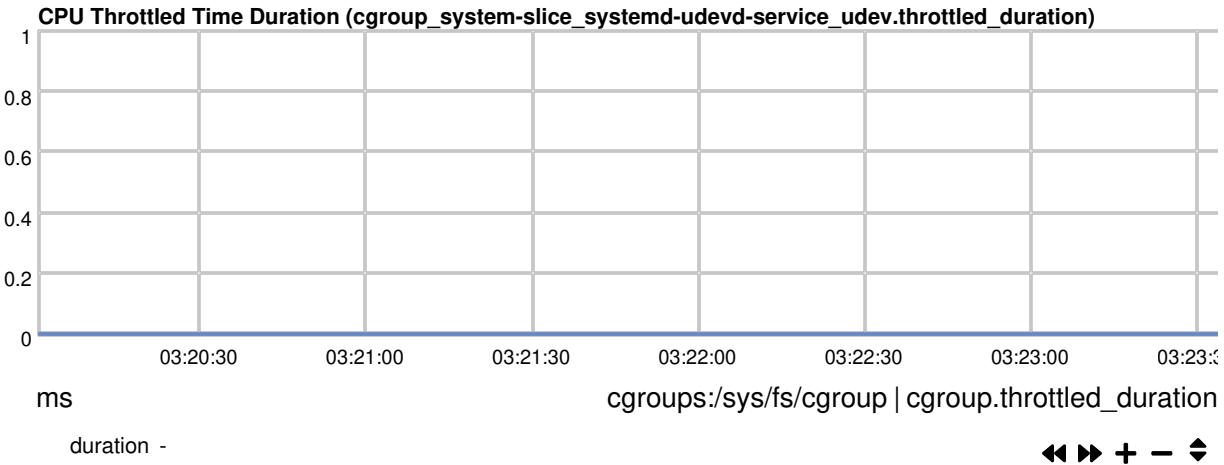
Total CPU utilization within the system-wide CPU resources (all cores). The amount of time spent by tasks of the cgroup in user and kernel (https://en.wikipedia.org/wiki/CPU_modes#Mode_types) modes.



The percentage of runnable periods when tasks in a cgroup have been throttled. The tasks have not been allowed to run because they have exhausted all of the available time as specified by their CPU quota.

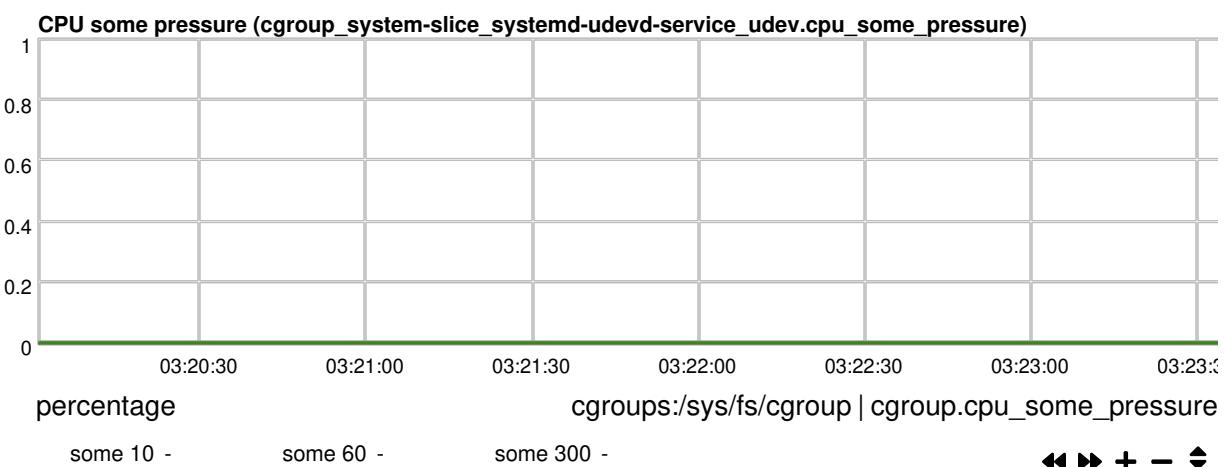


The total time duration for which tasks in a cgroup have been throttled. When an application has used its allotted CPU quota for a given period, it gets throttled until the next period.

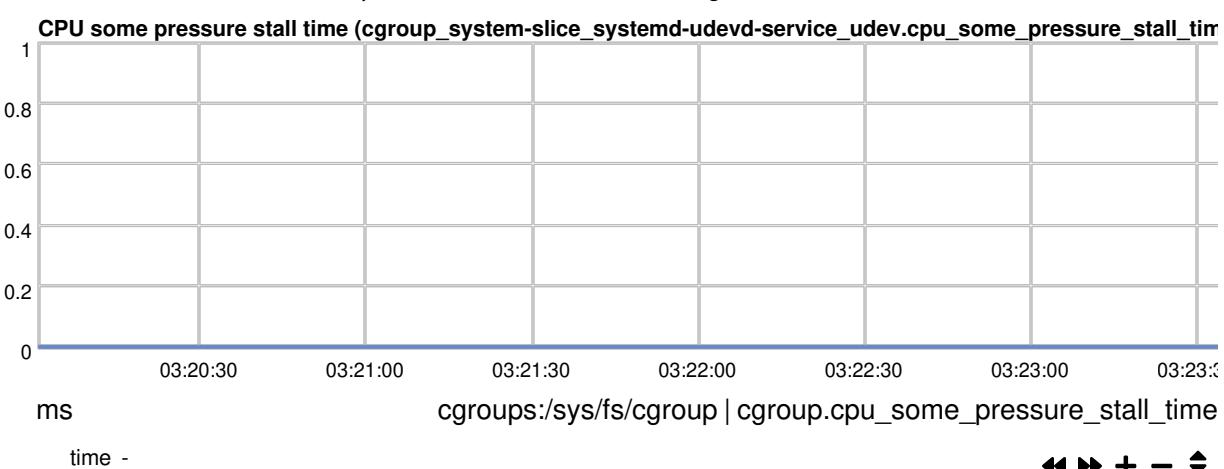


CPU Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

Some indicates the share of time in which at least **some tasks** are stalled on CPU. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.

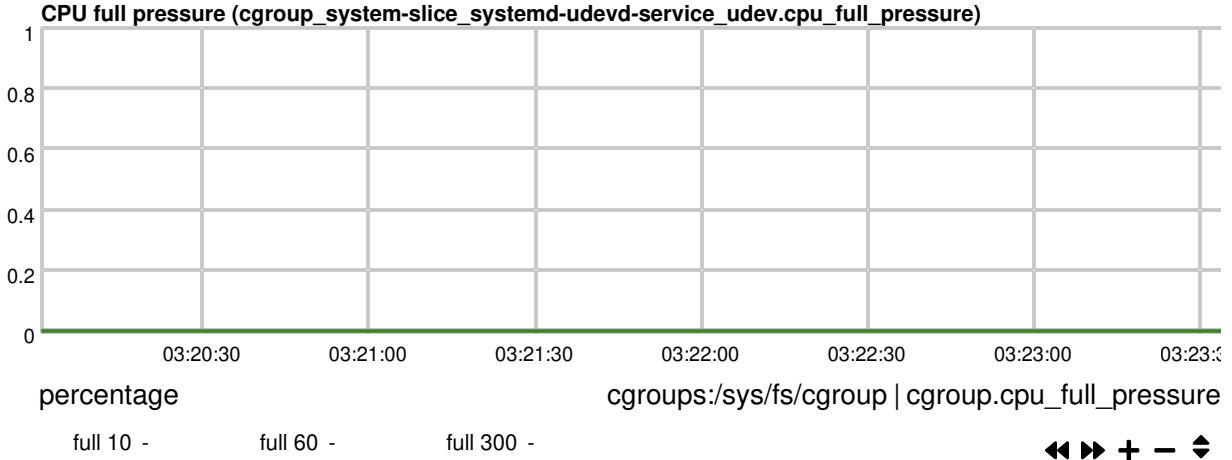


The amount of time some processes have been waiting for CPU time.

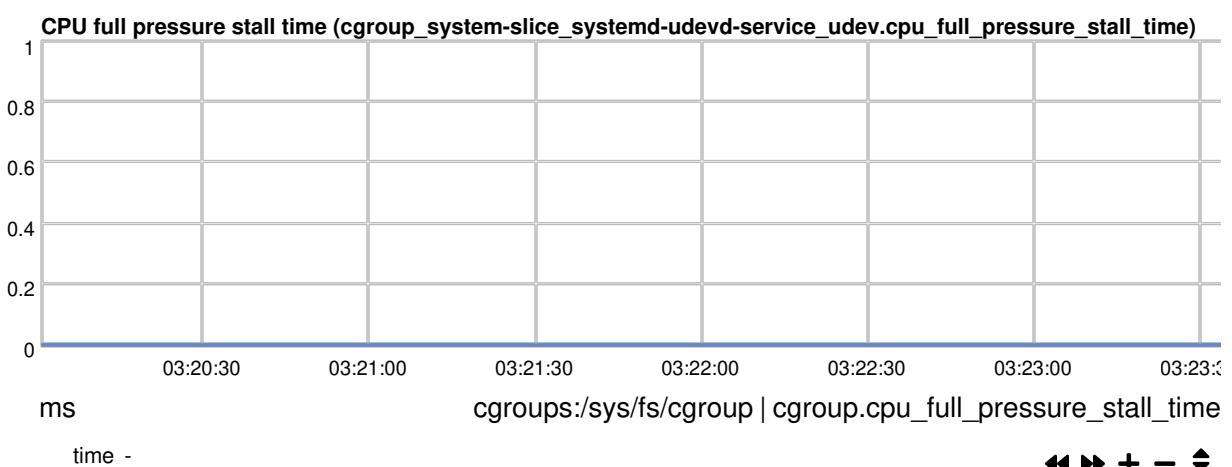


CPU Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

Full indicates the share of time in which **all non-idle tasks** are stalled on CPU resource simultaneously. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



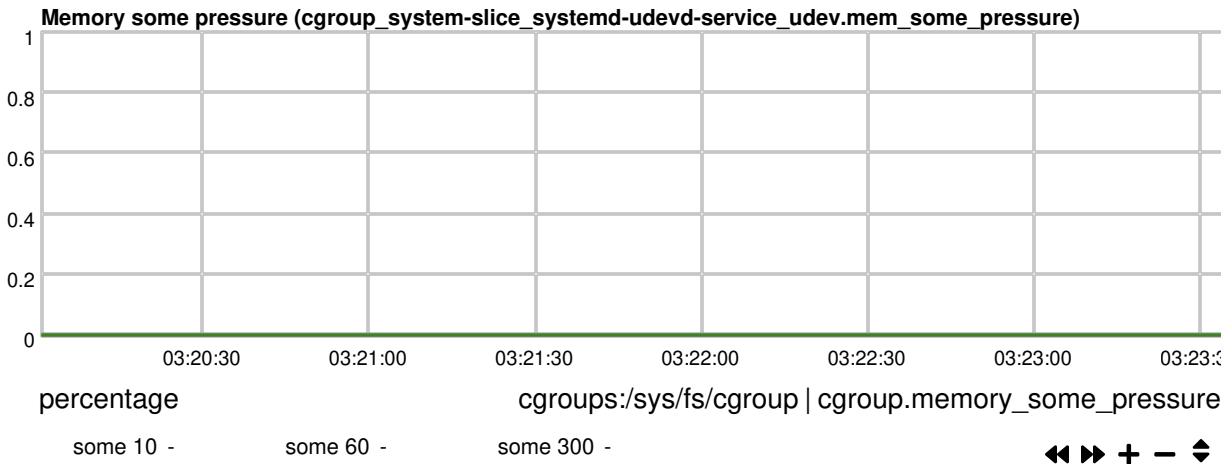
The amount of time all non-idle processes have been stalled due to CPU congestion.



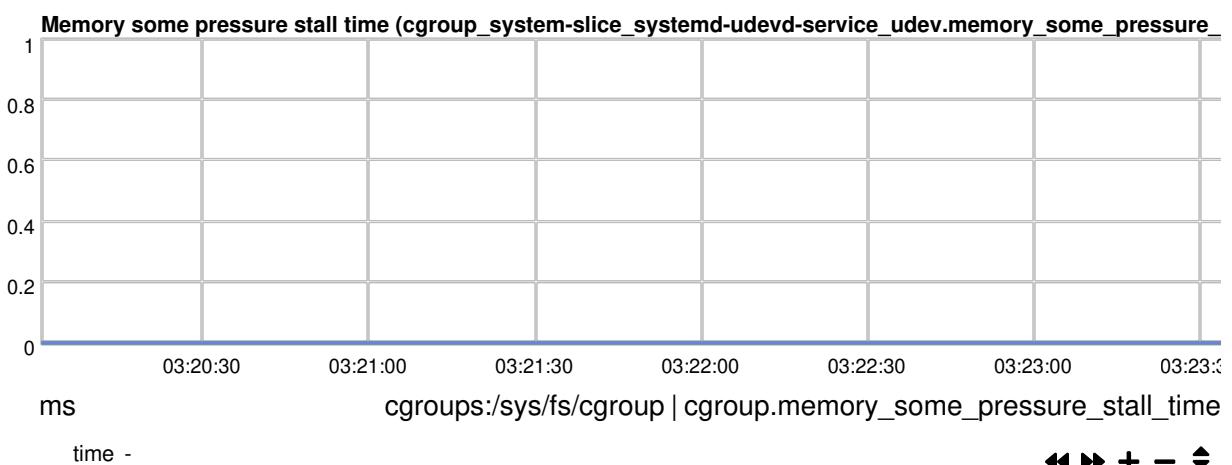
mem

Memory Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

Some indicates the share of time in which at least **some tasks** are stalled on memory. In this state the CPU is still doing productive work. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.

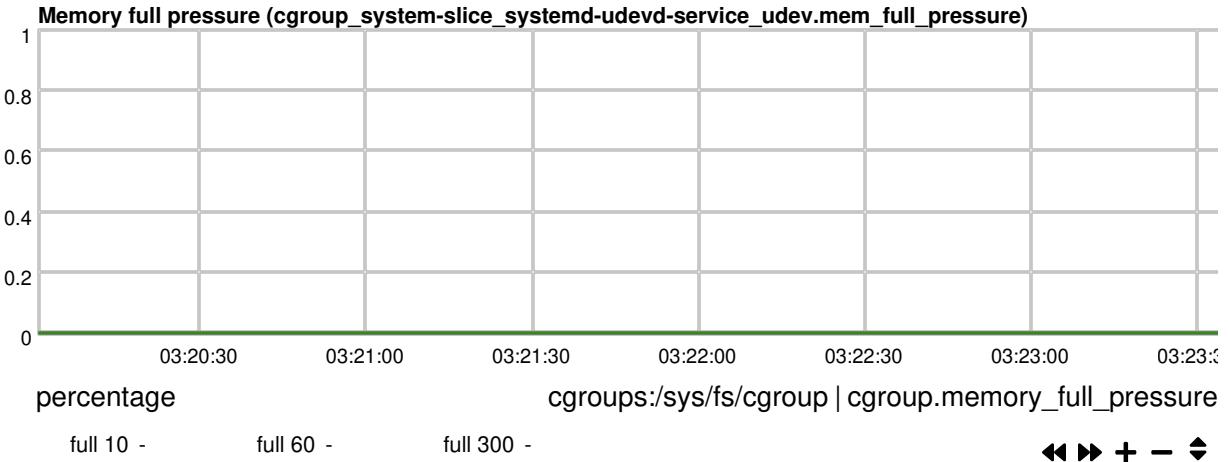


The amount of time some processes have been waiting due to memory congestion.

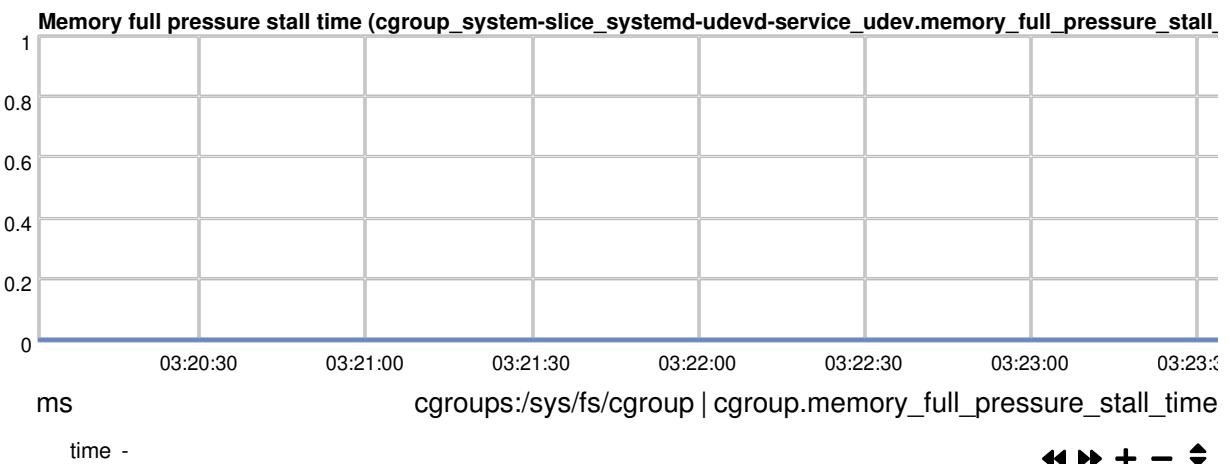


Memory Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

Full indicates the share of time in which **all non-idle tasks** are stalled on memory resource simultaneously. In this state actual CPU cycles are going to waste, and a workload that spends extended time in this state is considered to be thrashing. This has severe impact on performance. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



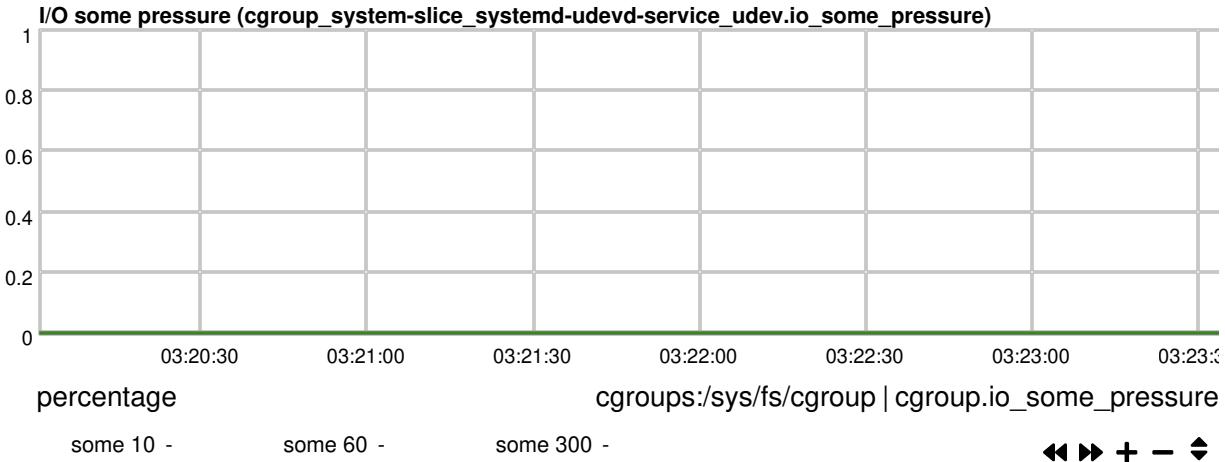
The amount of time all non-idle processes have been stalled due to memory congestion.



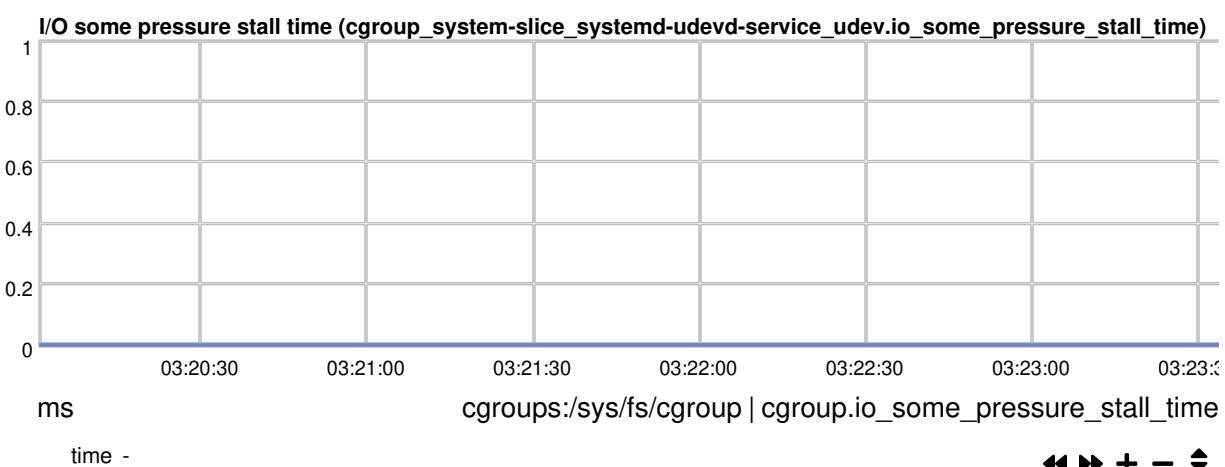
disk

I/O Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>).

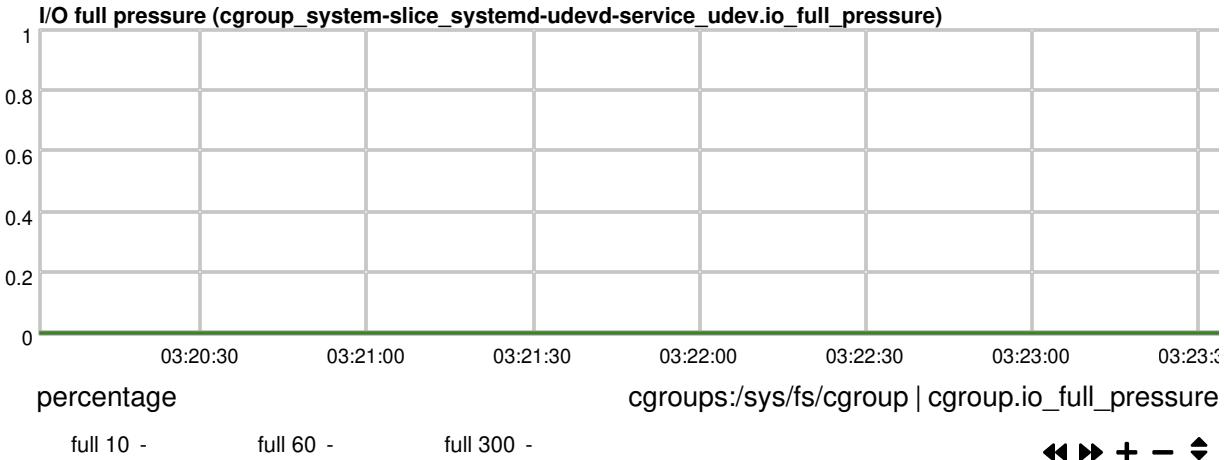
Some indicates the share of time in which at least **some tasks** are stalled on I/O. In this state the CPU is still doing productive work. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



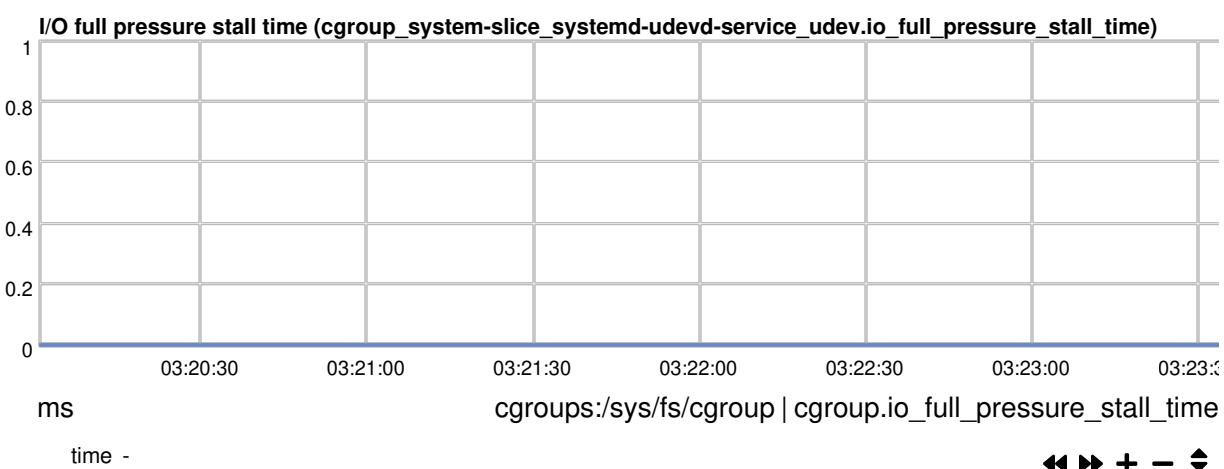
The amount of time some processes have been waiting due to I/O congestion.



I/O Pressure Stall Information (<https://www.kernel.org/doc/html/latest/accounting/psi.html>). **Full** line indicates the share of time in which **all non-idle tasks** are stalled on I/O resource simultaneously. In this state actual CPU cycles are going to waste, and a workload that spends extended time in this state is considered to be thrashing. This has severe impact on performance. The ratios are tracked as recent trends over 10-, 60-, and 300-second windows.



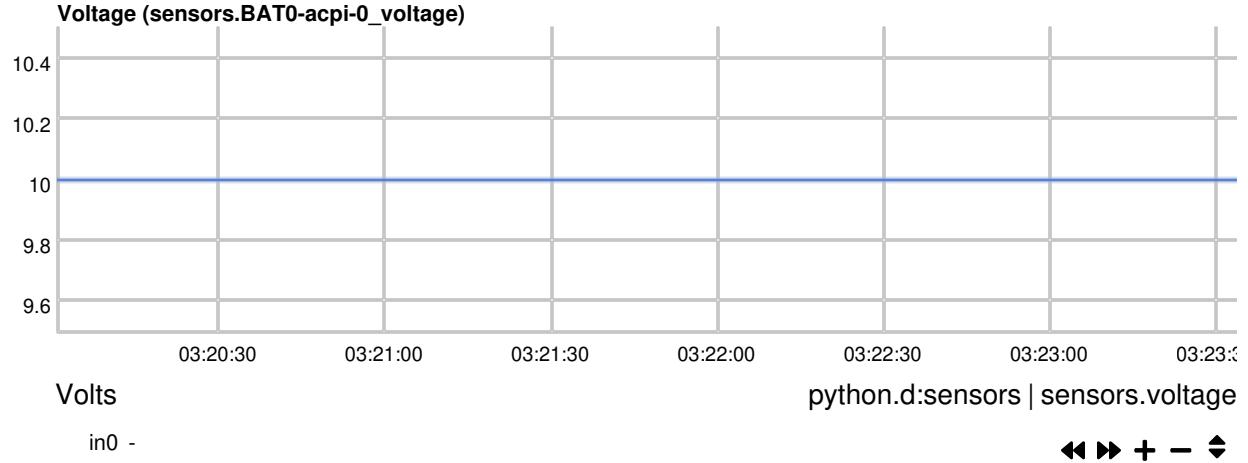
The amount of time all non-idle processes have been stalled due to I/O congestion.



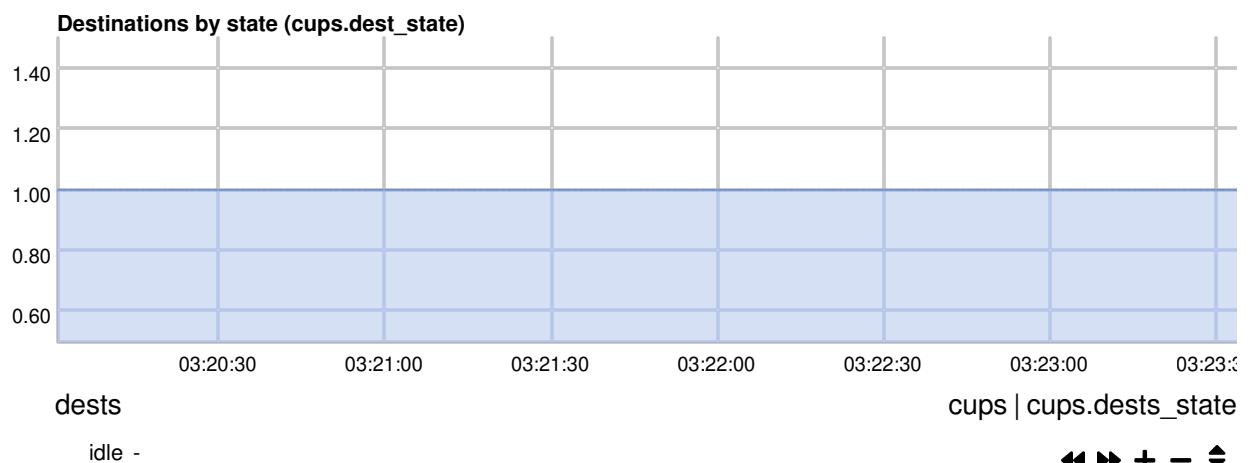
Sensors

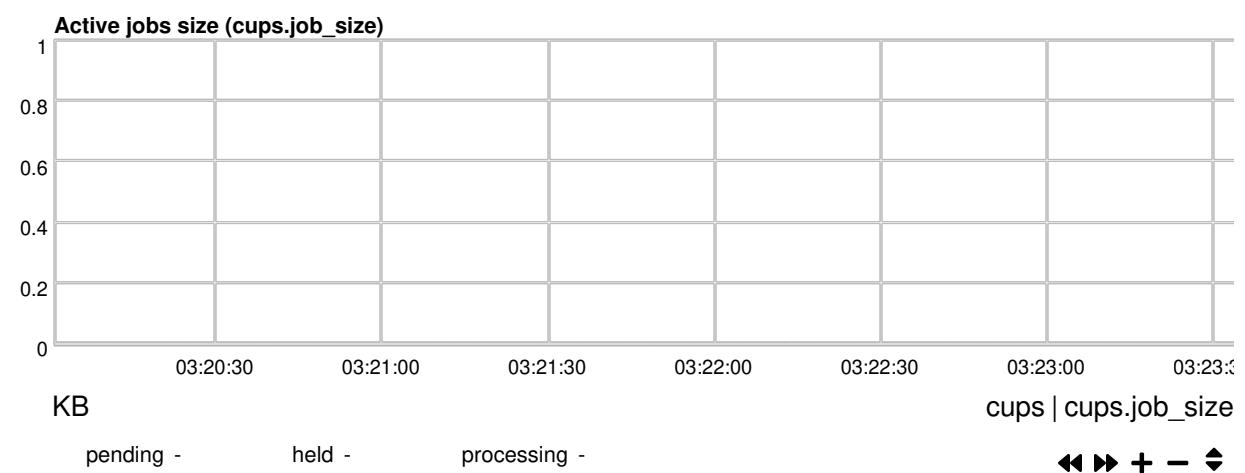
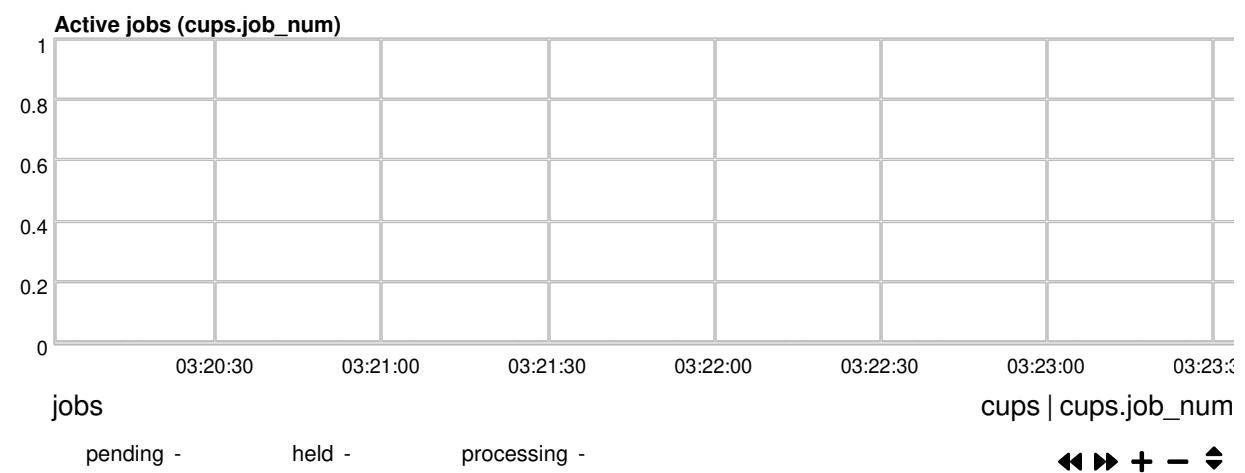
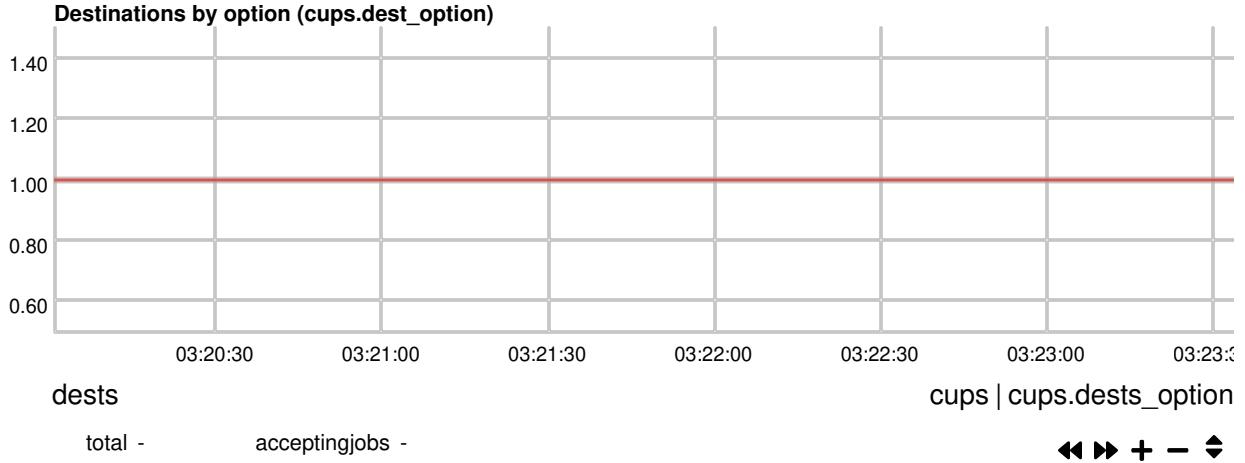
Readings of the configured system sensors.

voltage

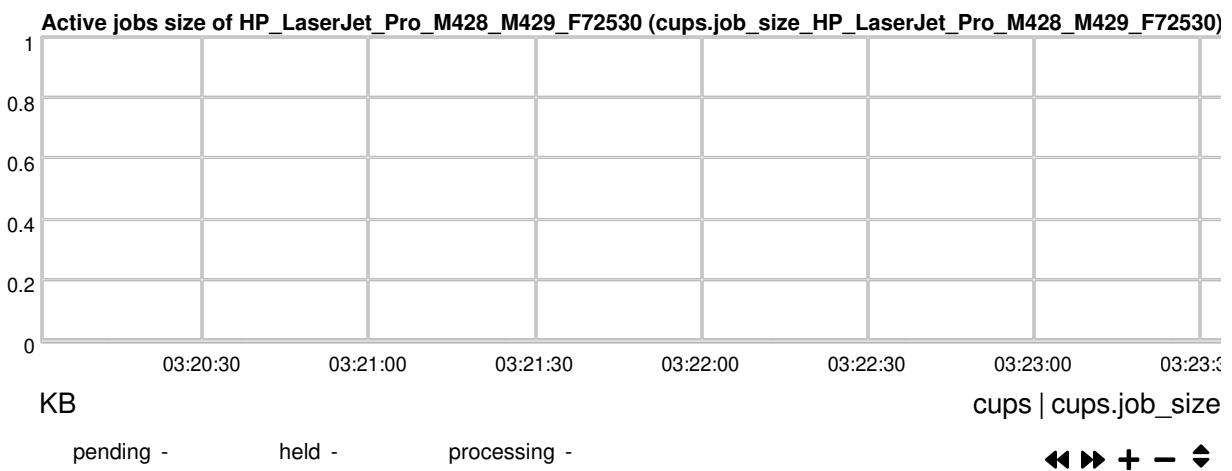
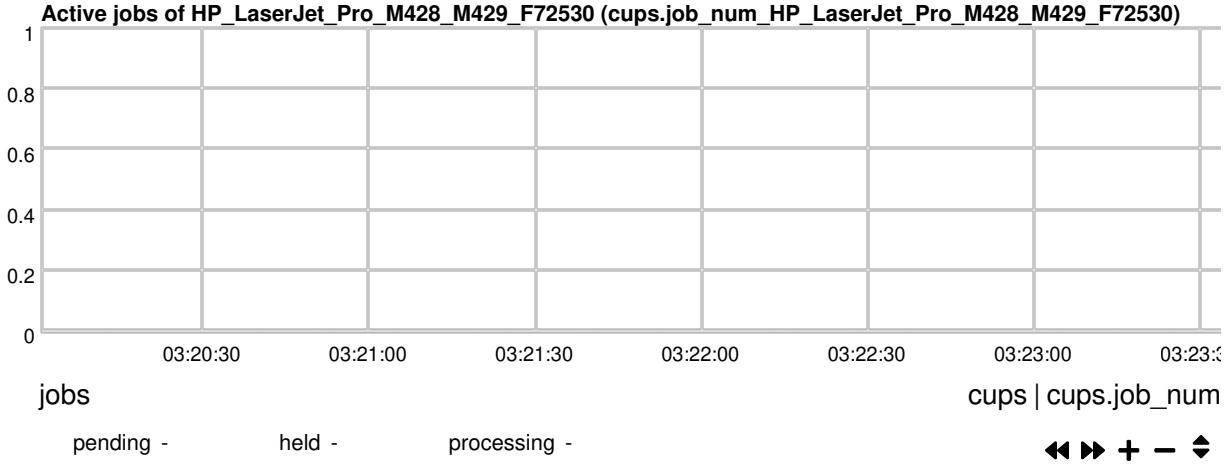


overview





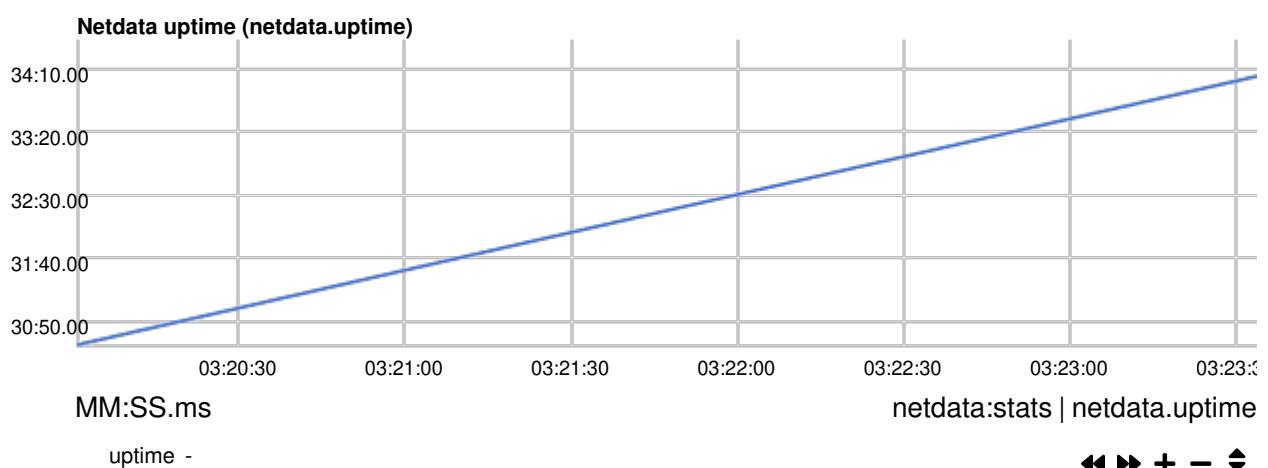
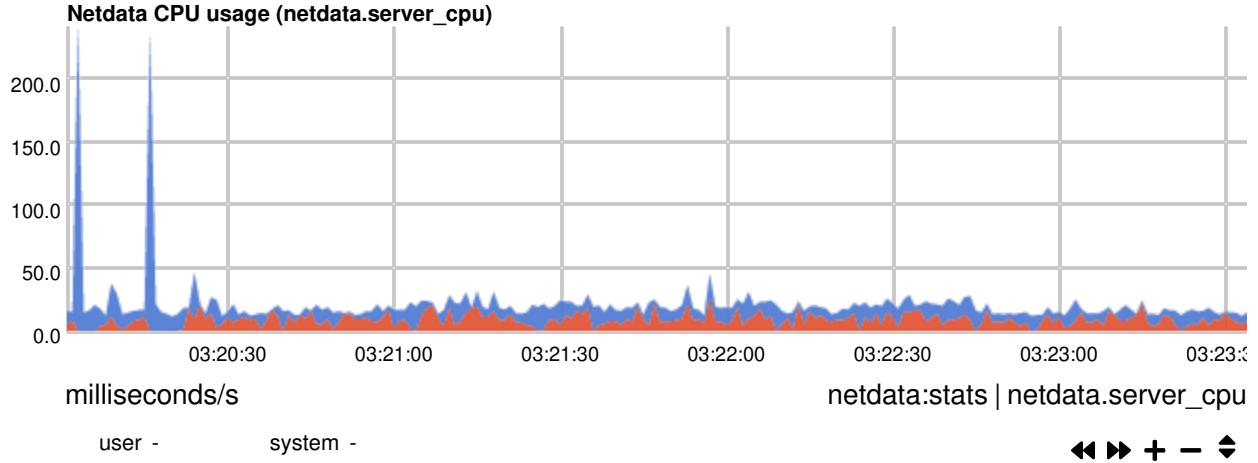
HP LaserJet P... M429 F72530



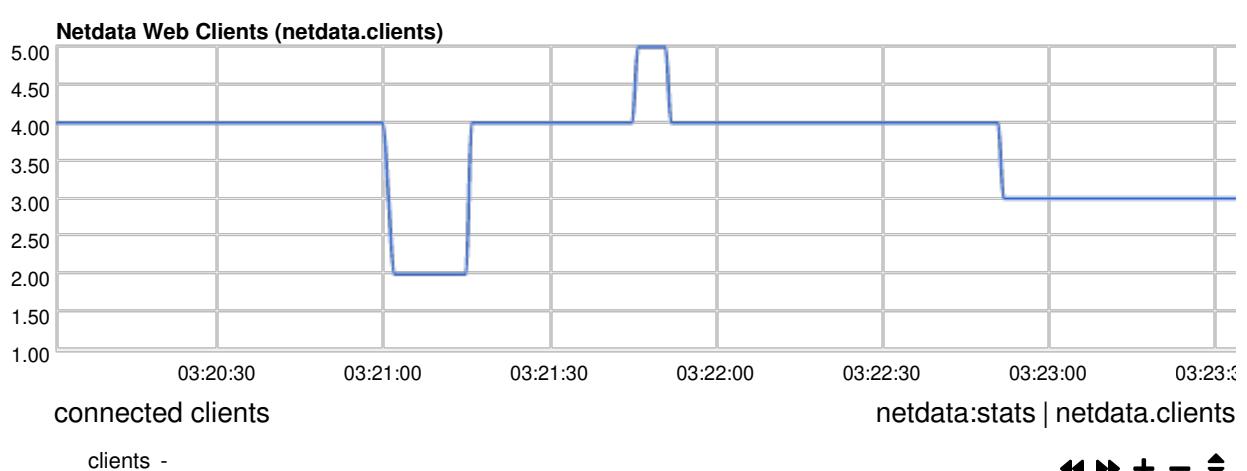
📶 Netdata Monitoring

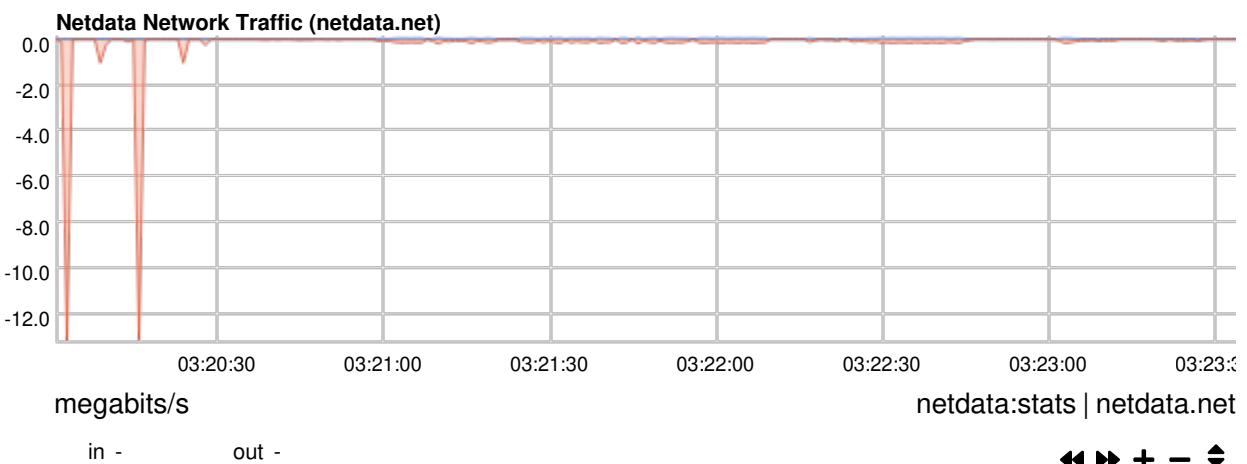
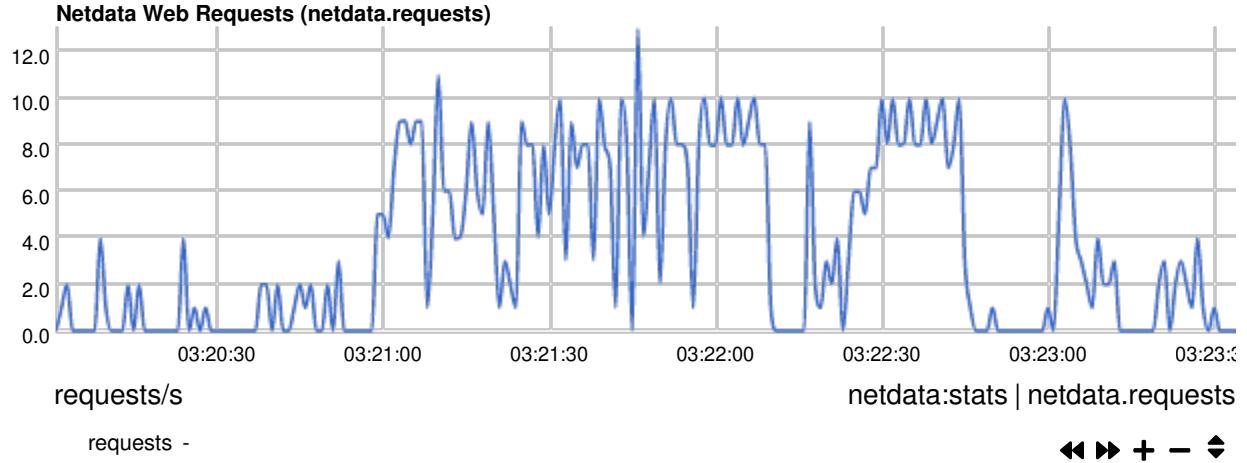
Performance metrics for the operation of netdata itself and its plugins.

netdata

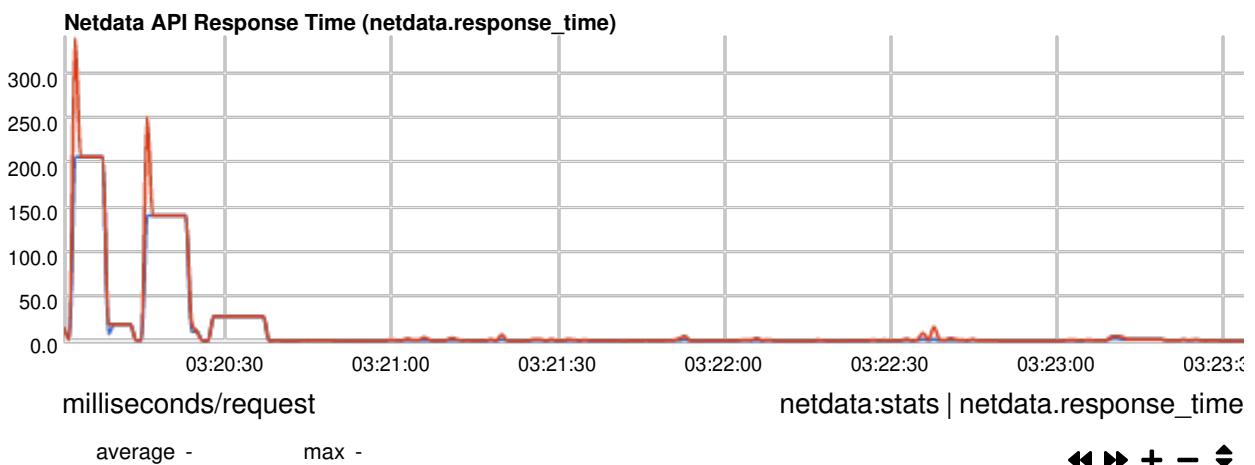


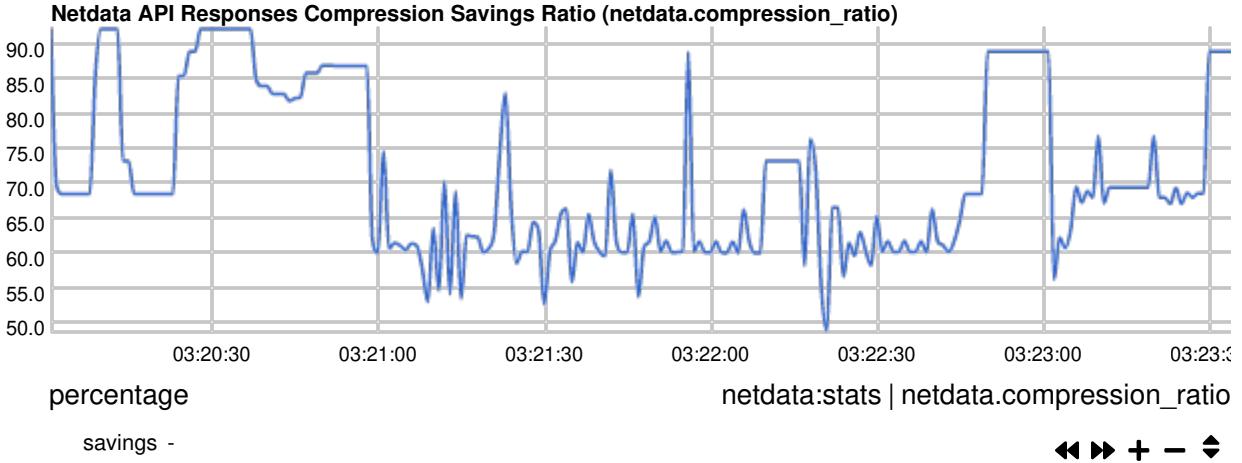
api



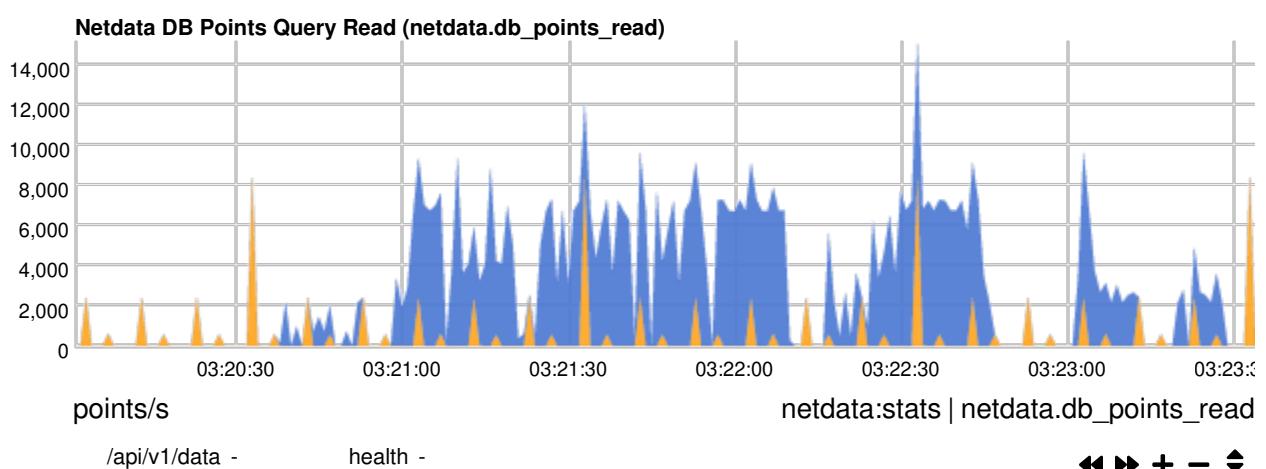
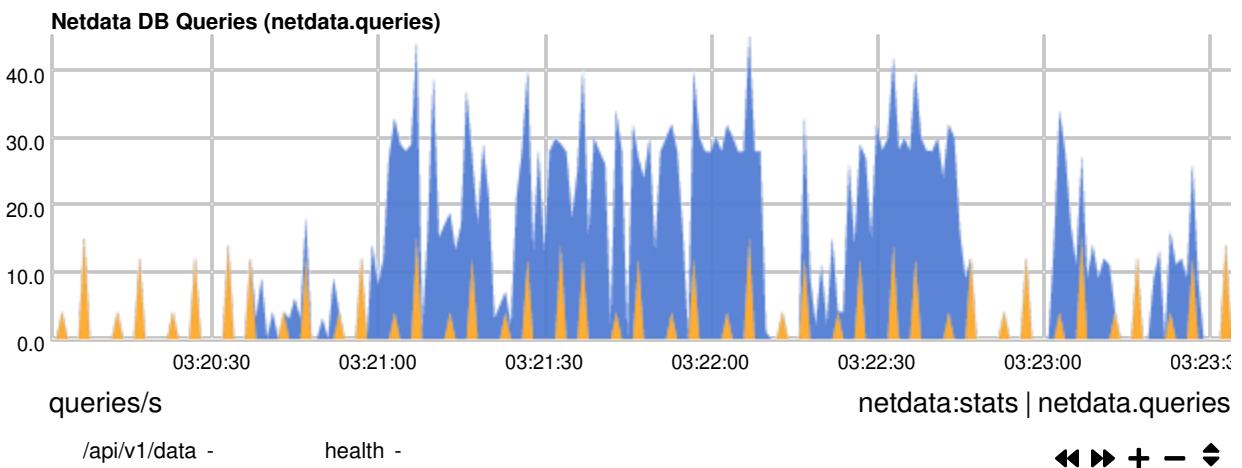


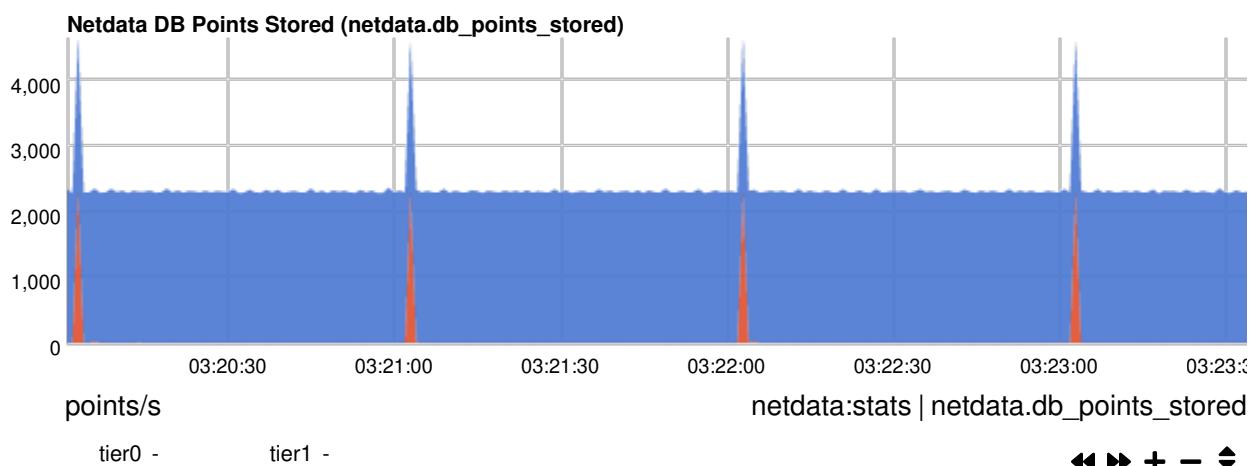
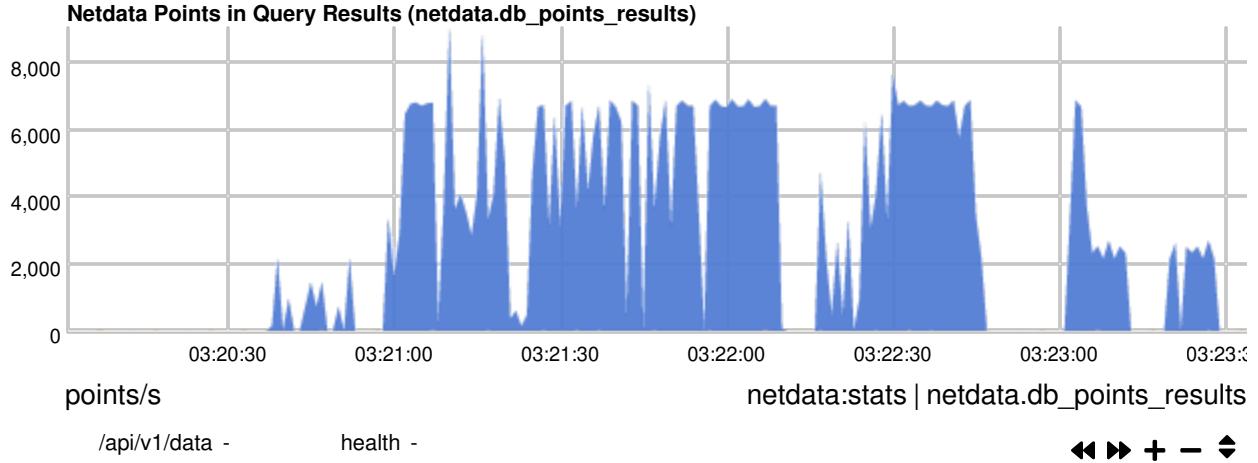
The netdata API response time measures the time netdata needed to serve requests. This time includes everything, from the reception of the first byte of a request, to the dispatch of the last byte of its reply, therefore it includes all network latencies involved (i.e. a client over a slow network will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



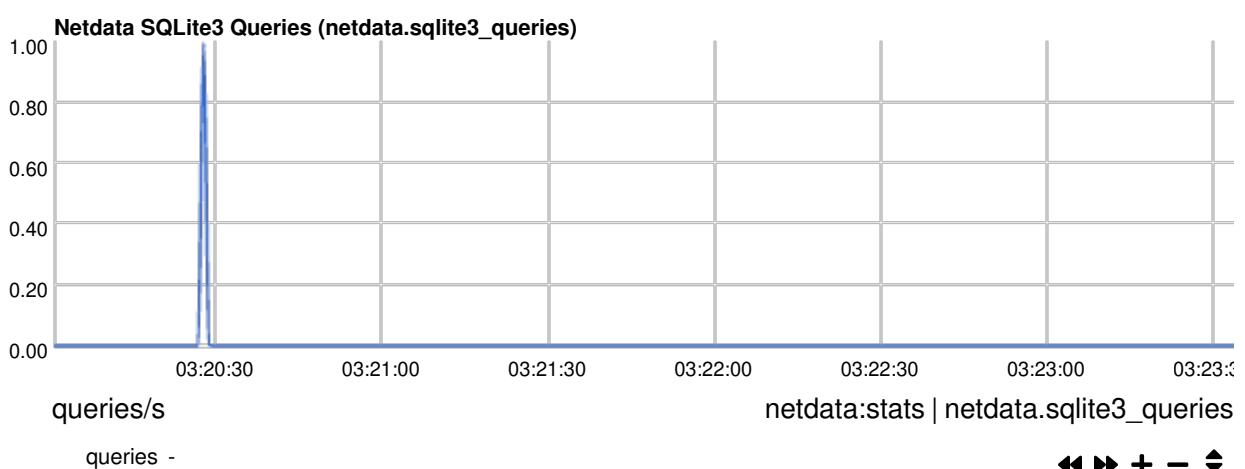


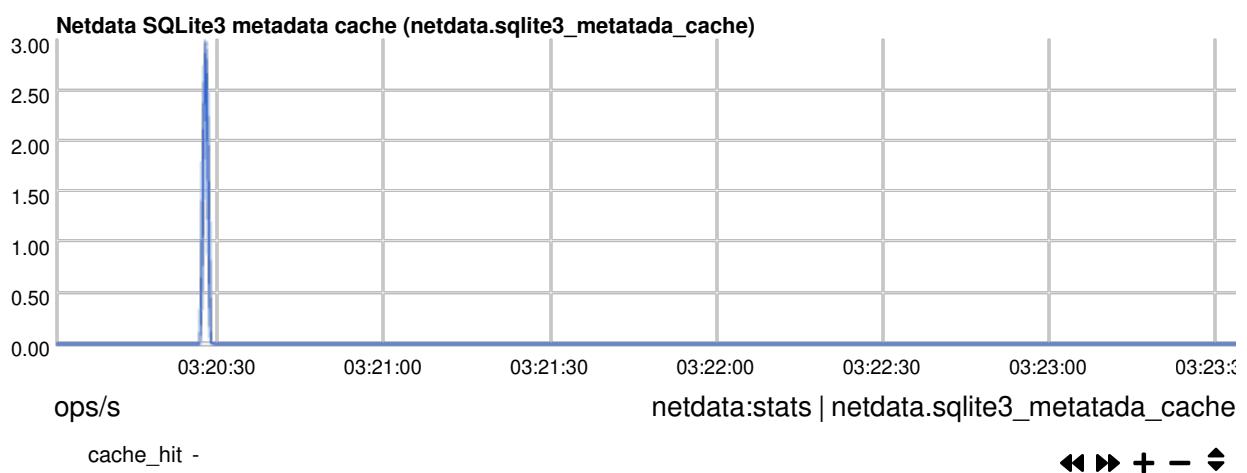
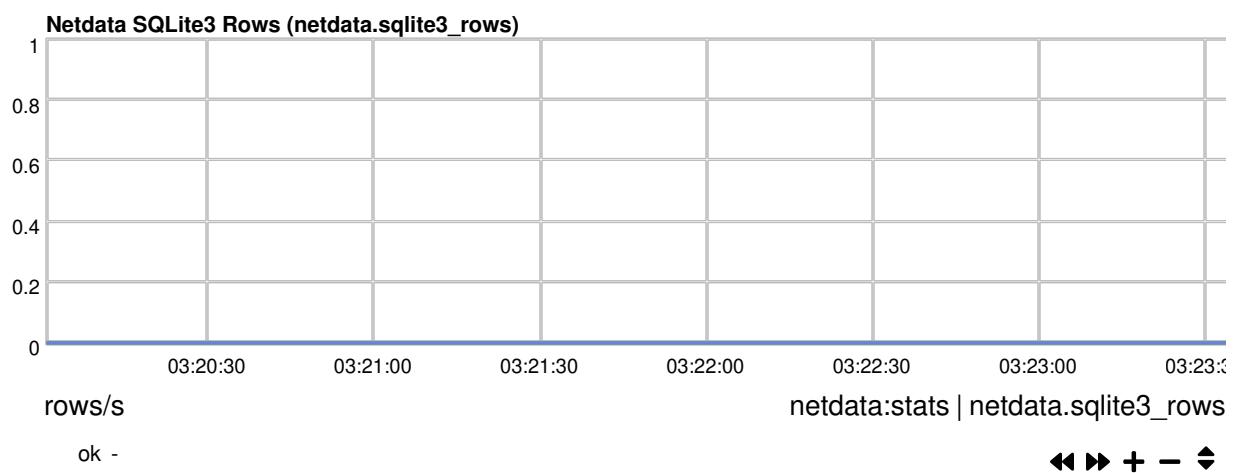
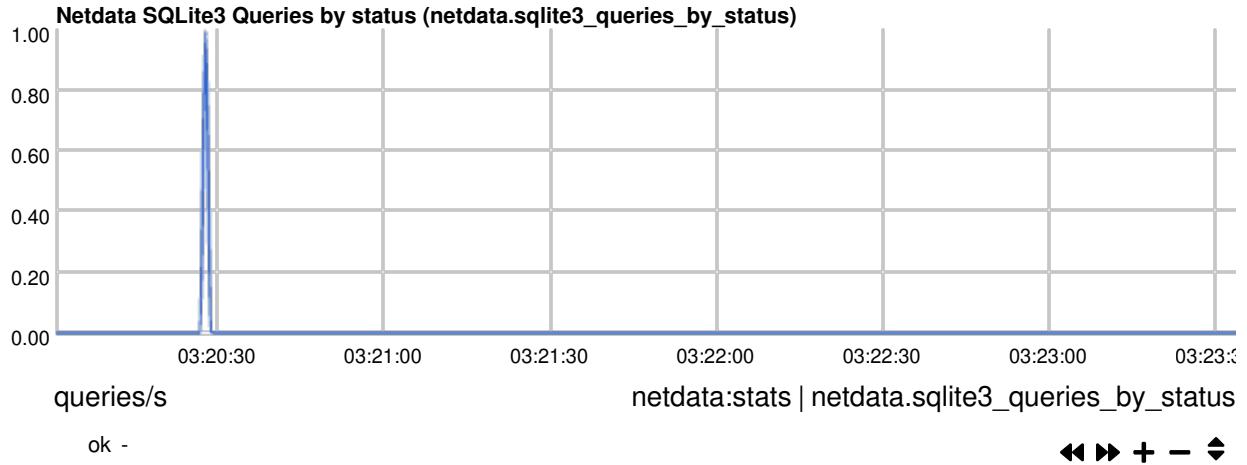
queries

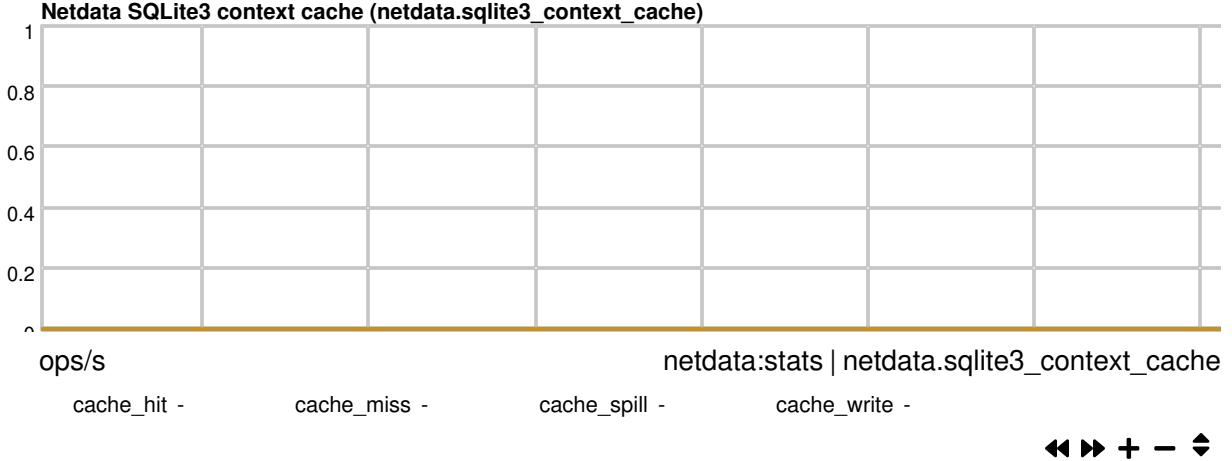




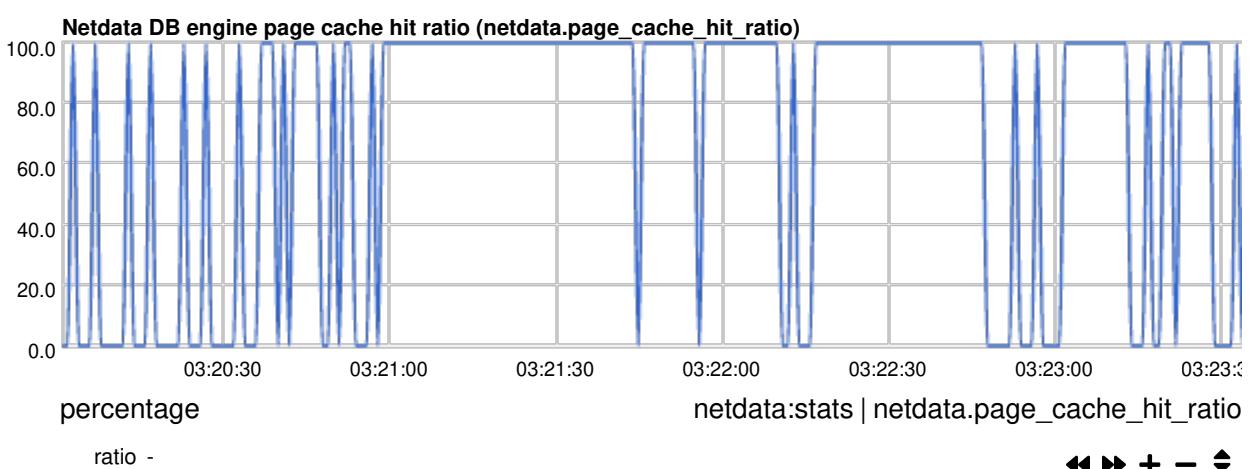
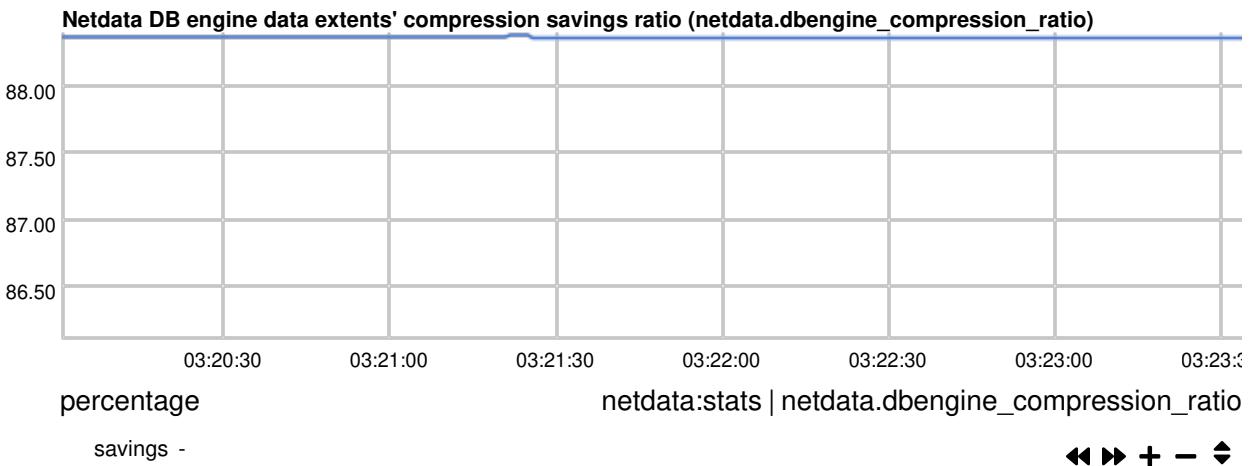
sqlite3

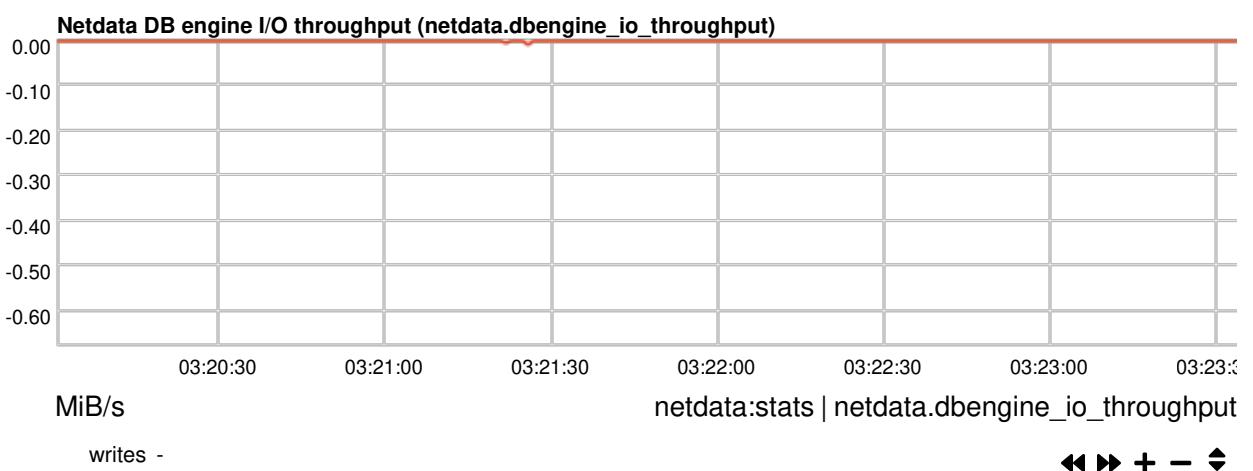
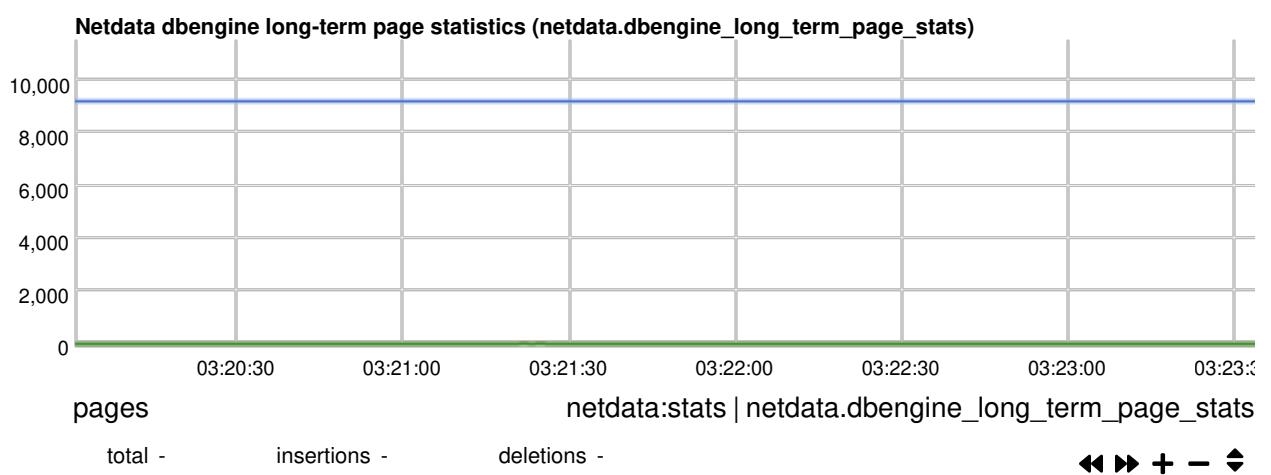
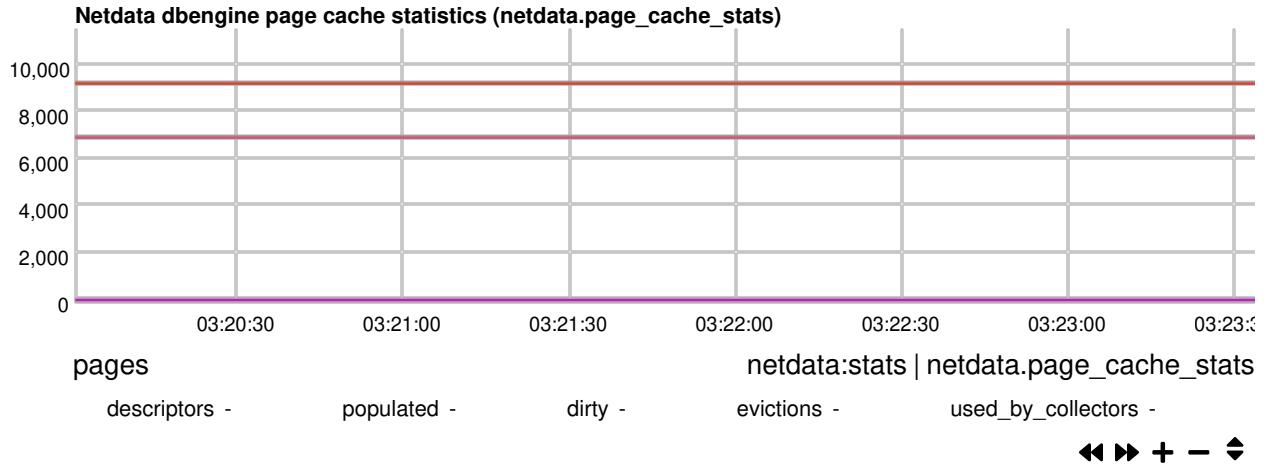


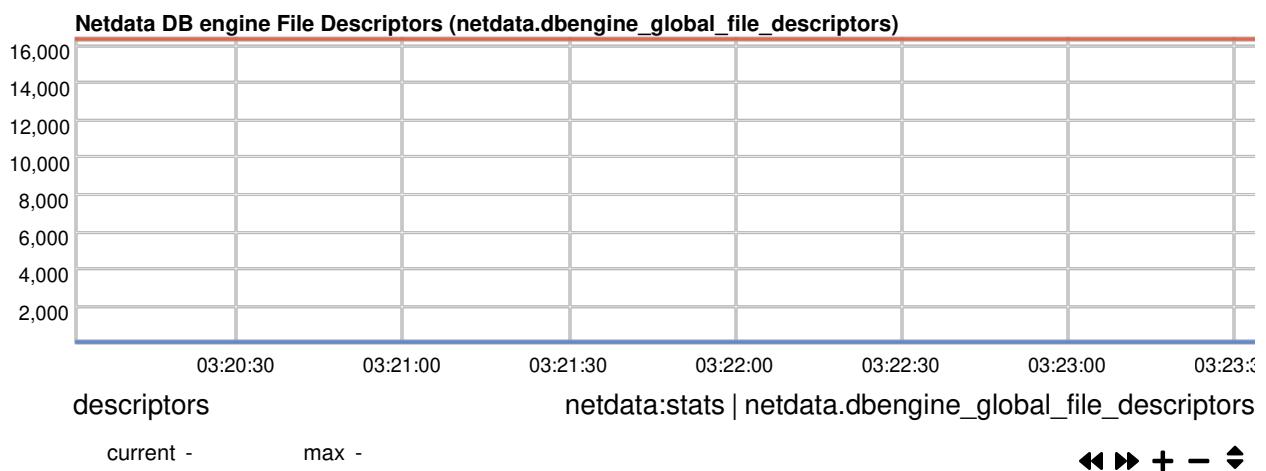
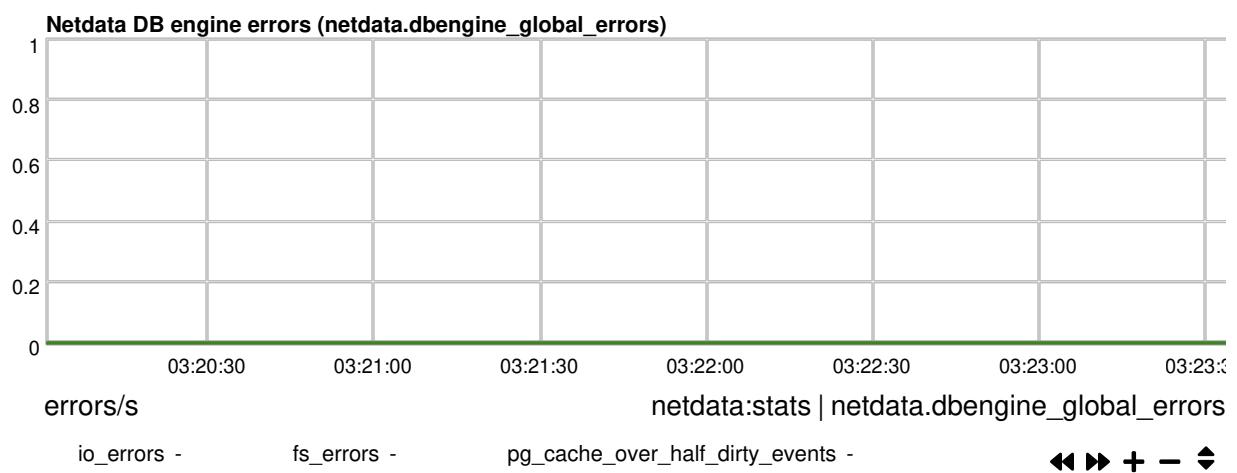
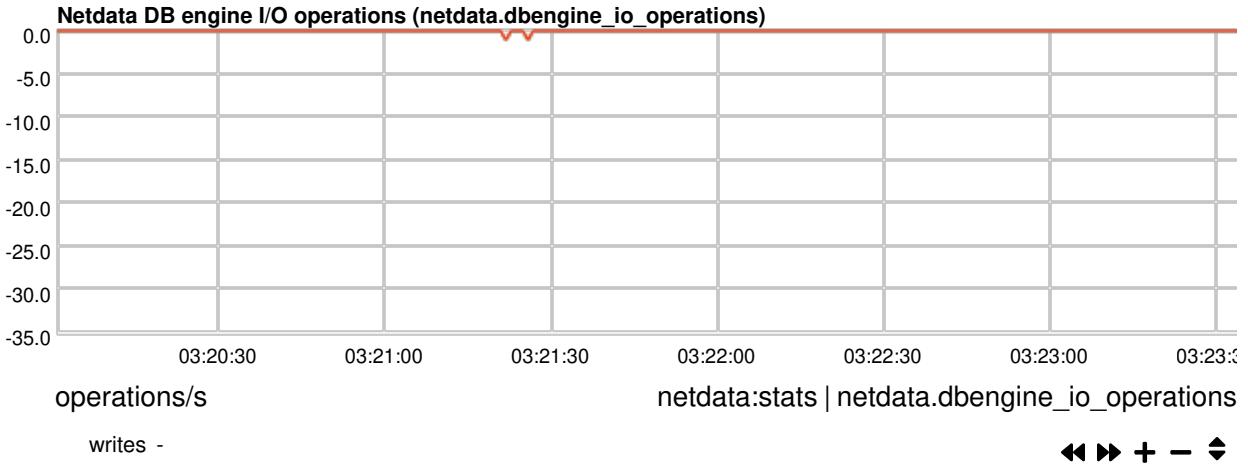


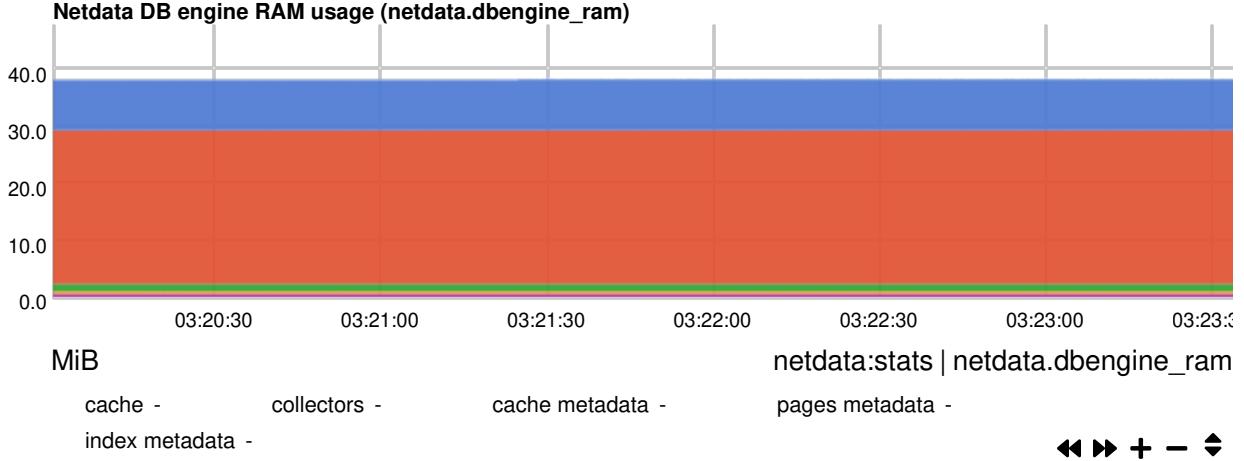


dbengine

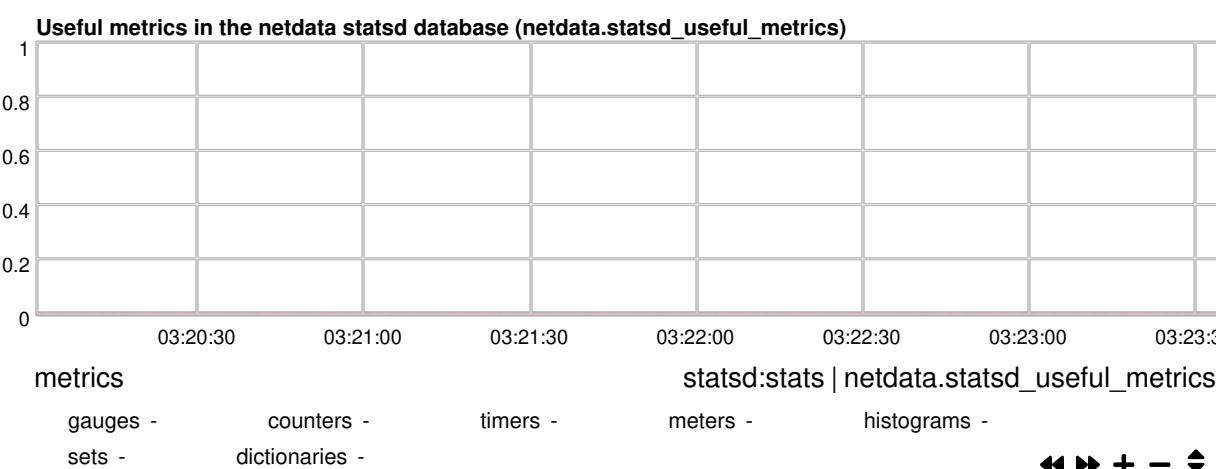
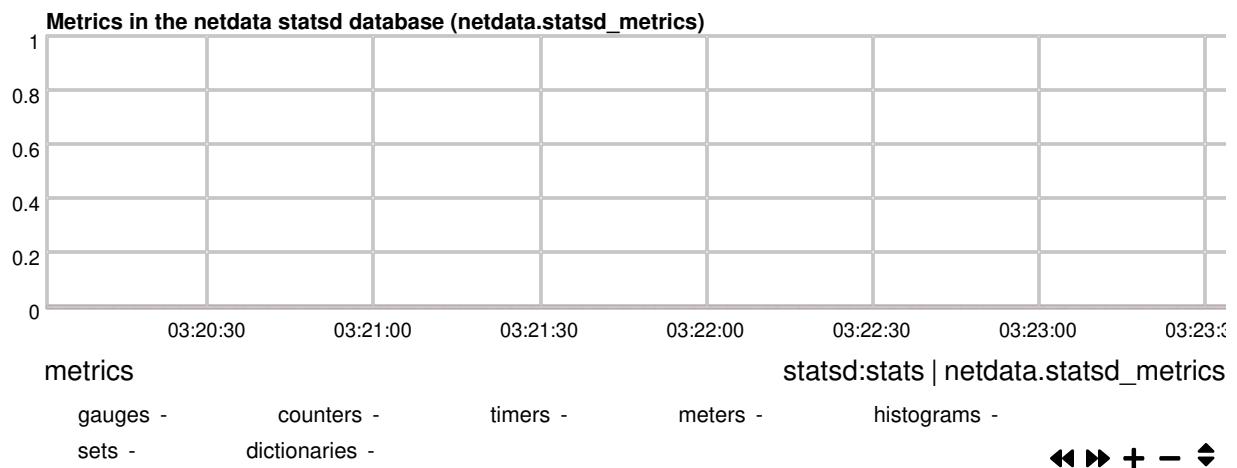


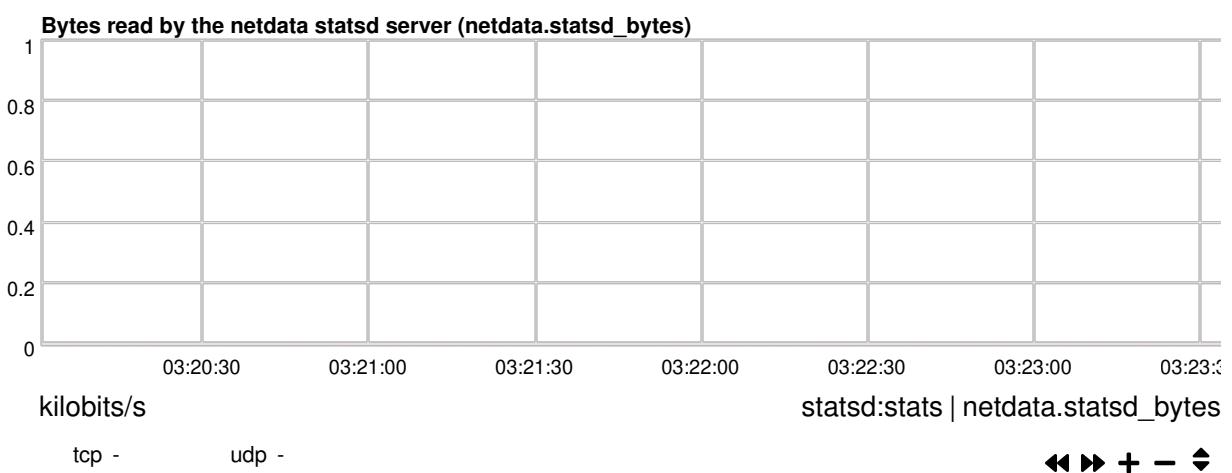
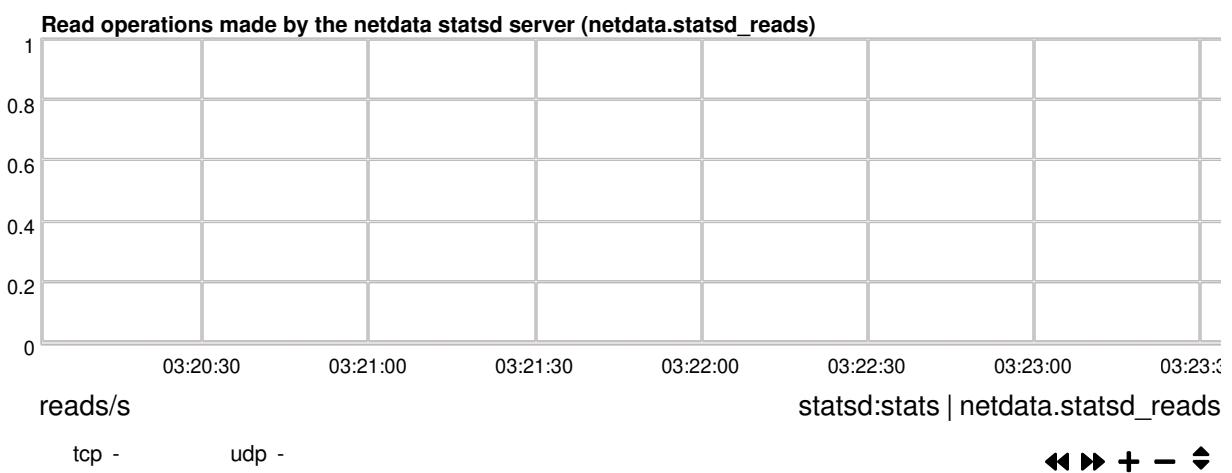
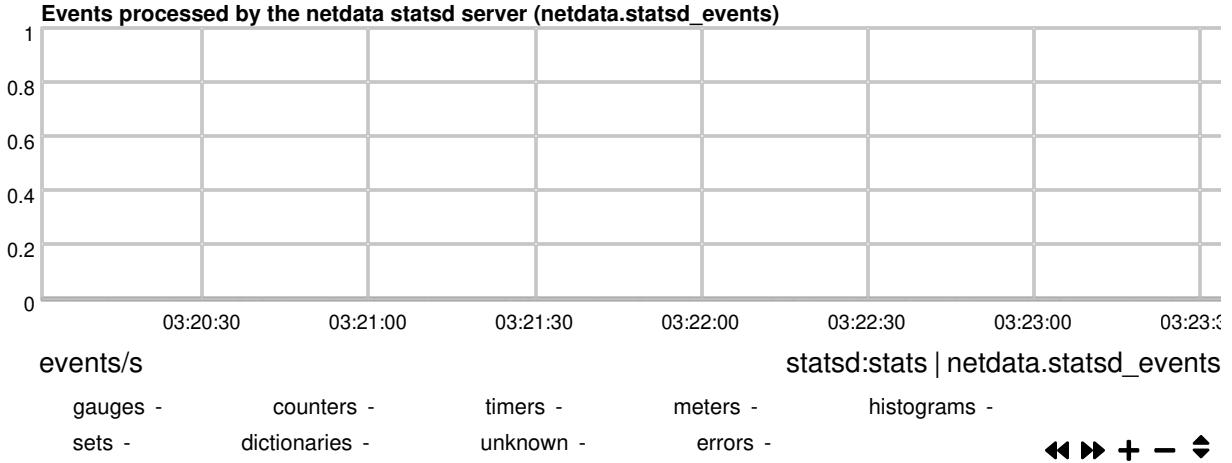


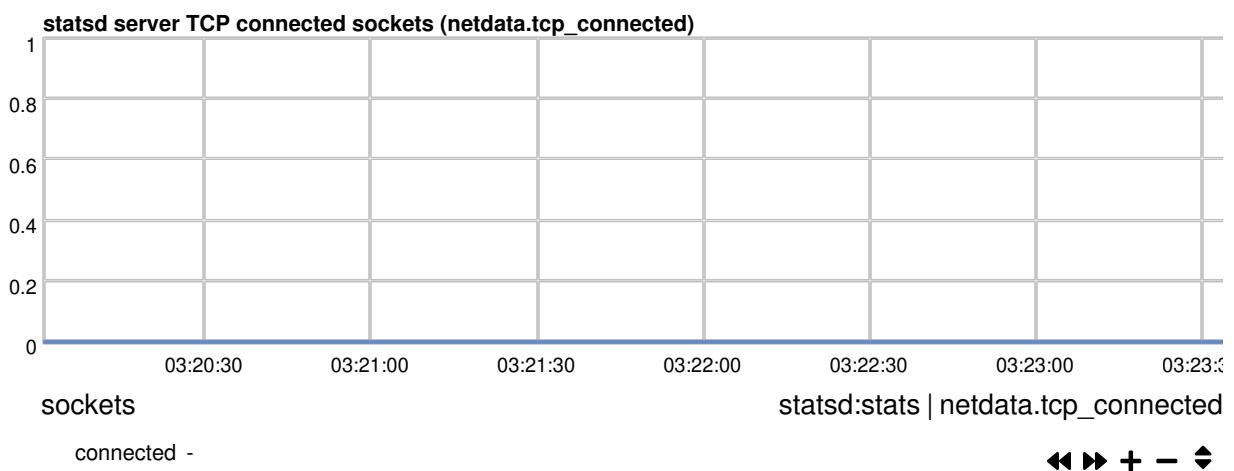
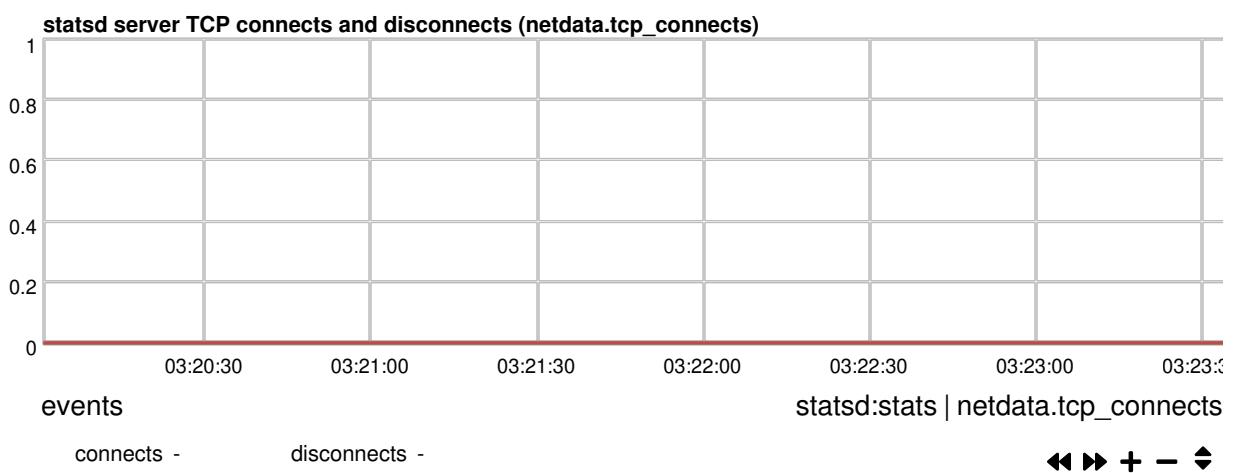
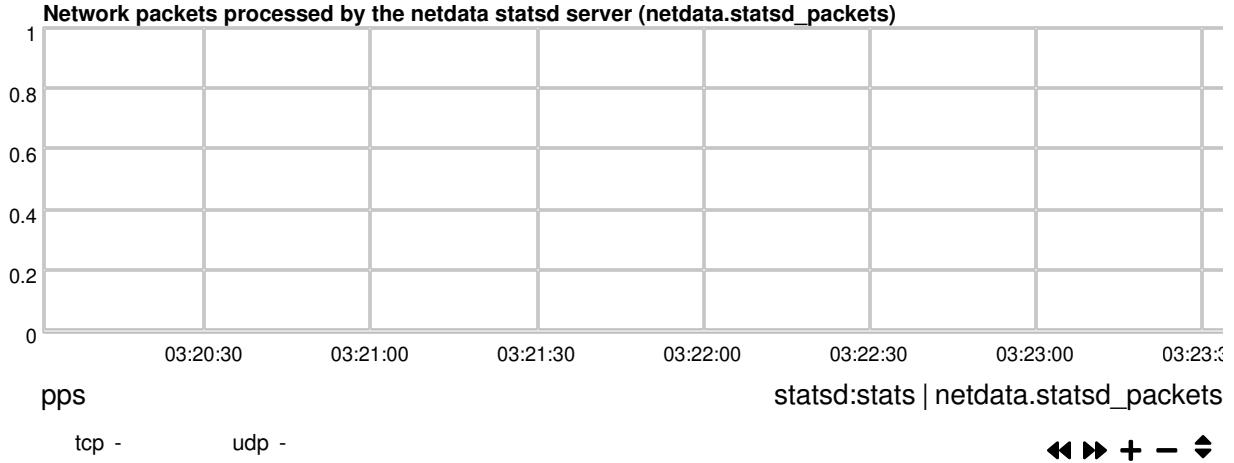


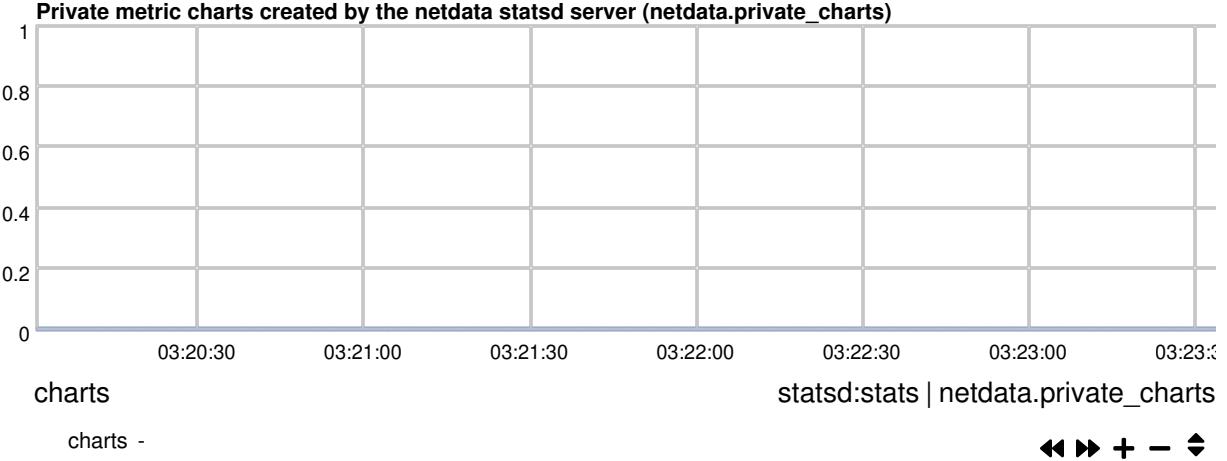


statsd

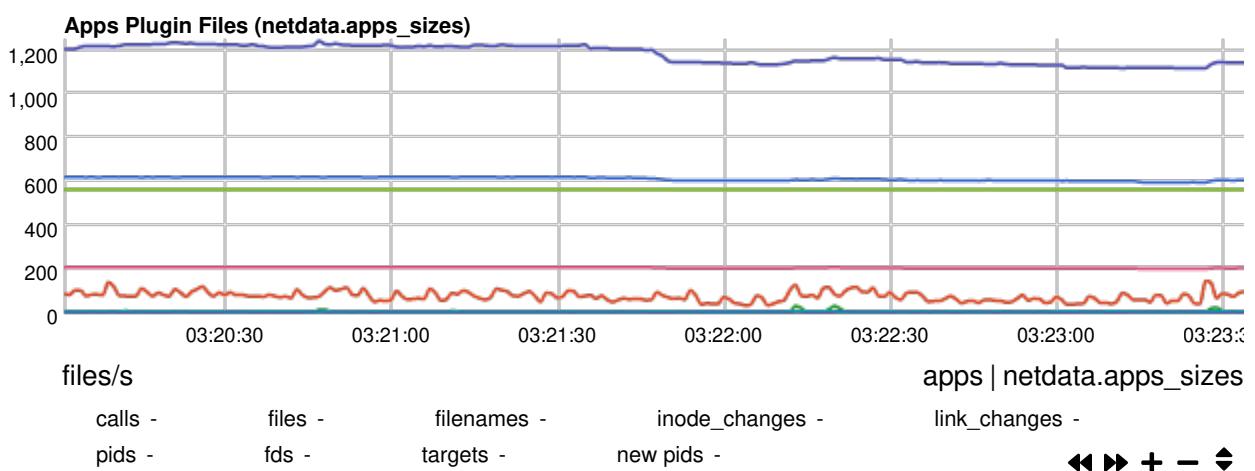
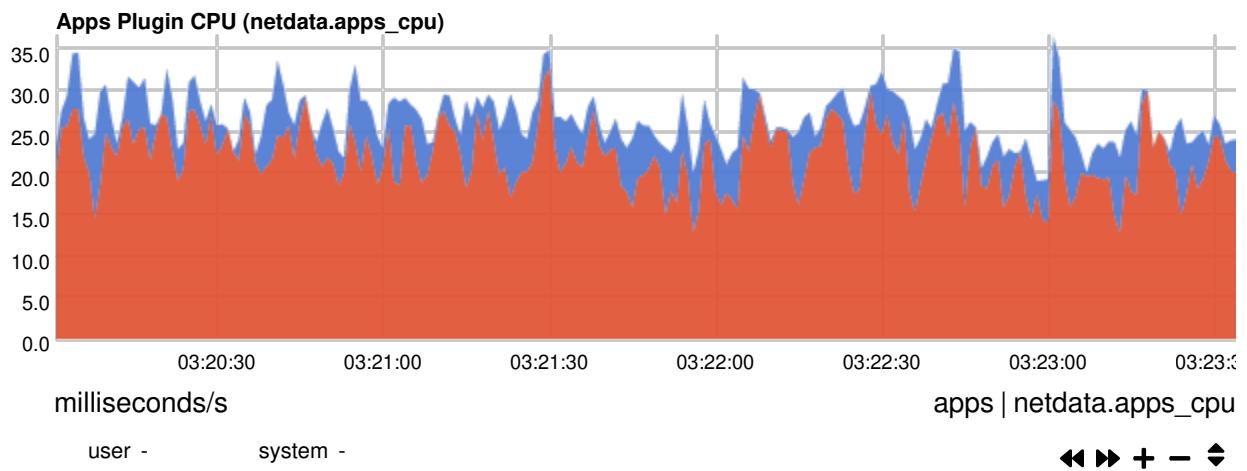


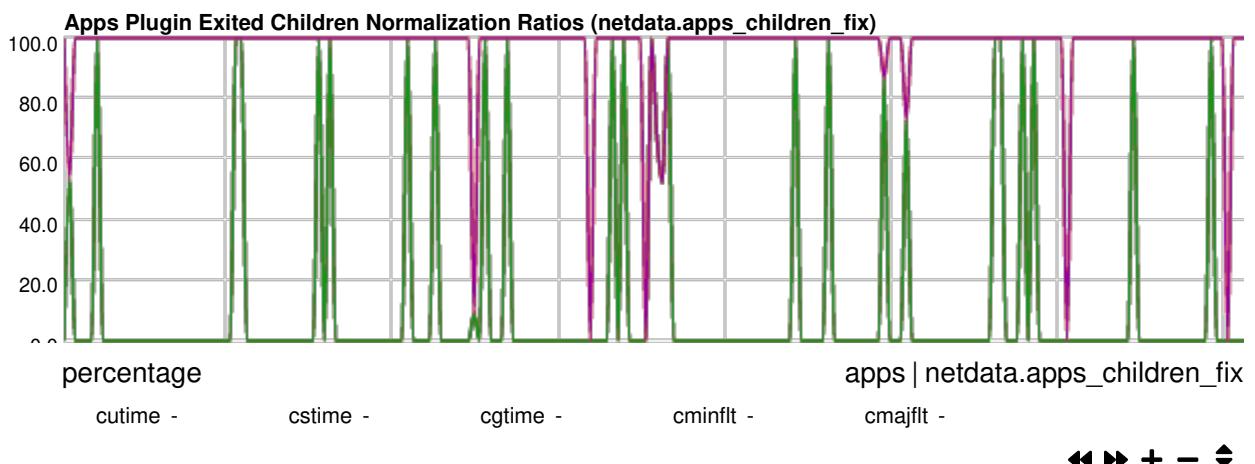
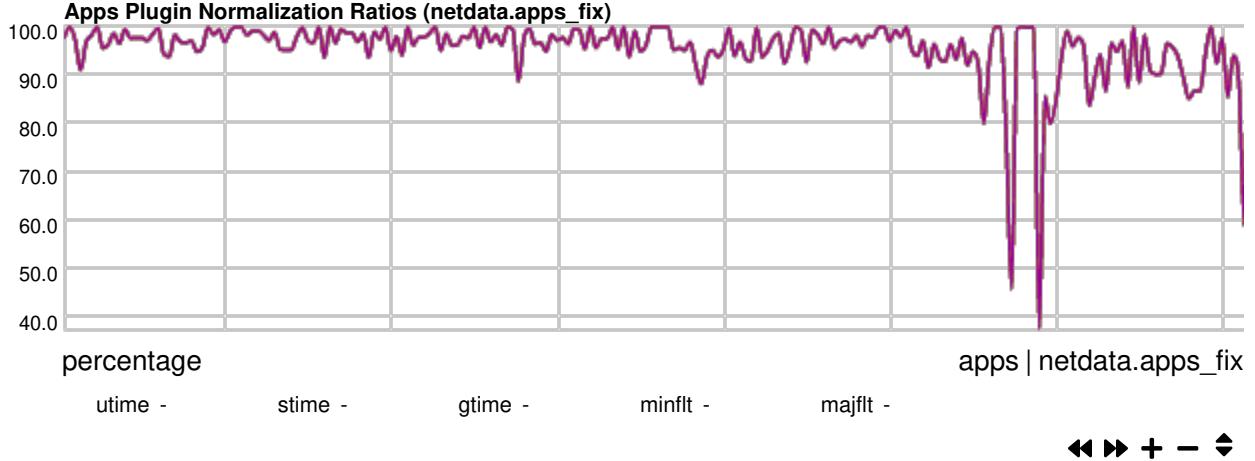




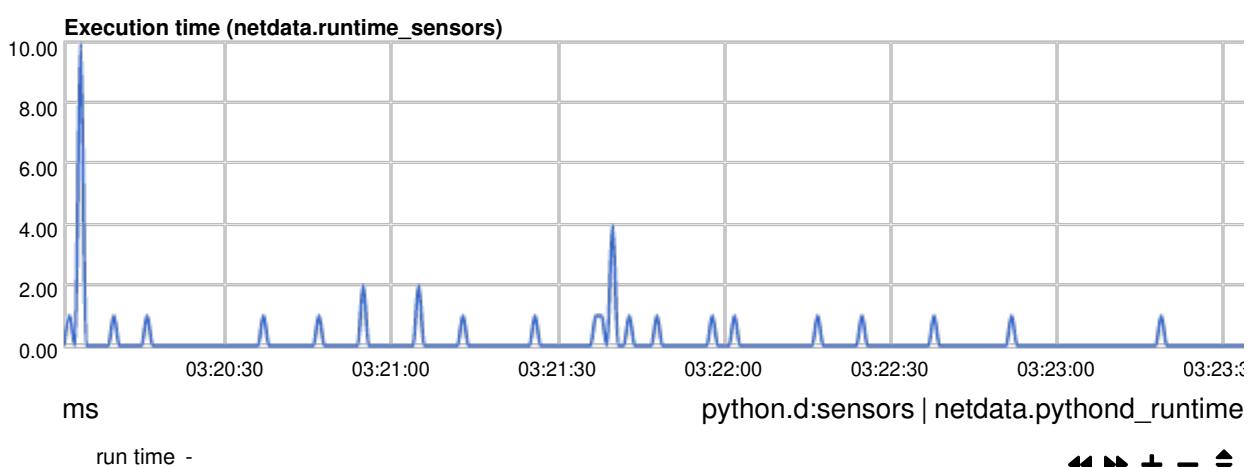


apps.plugin

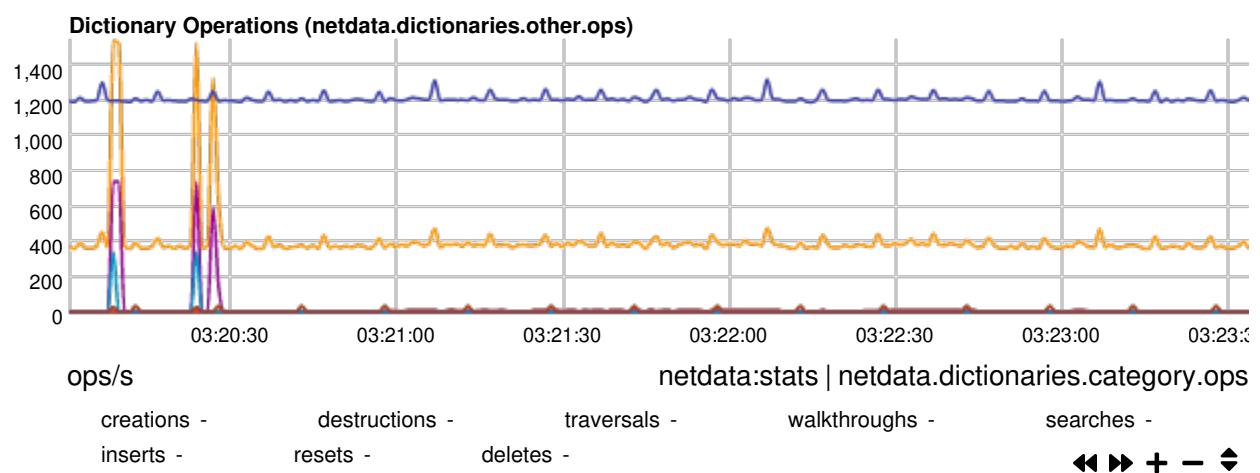
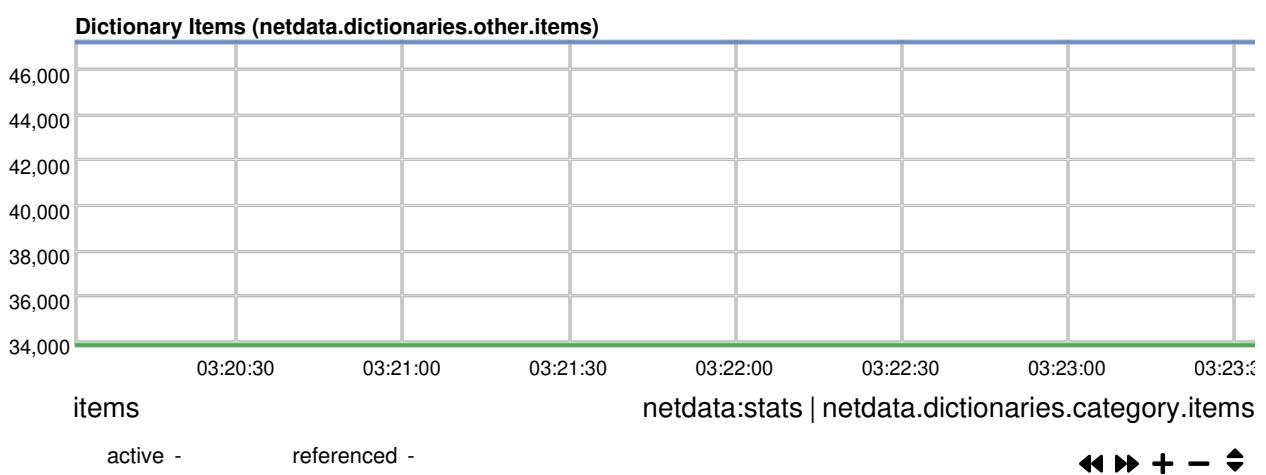
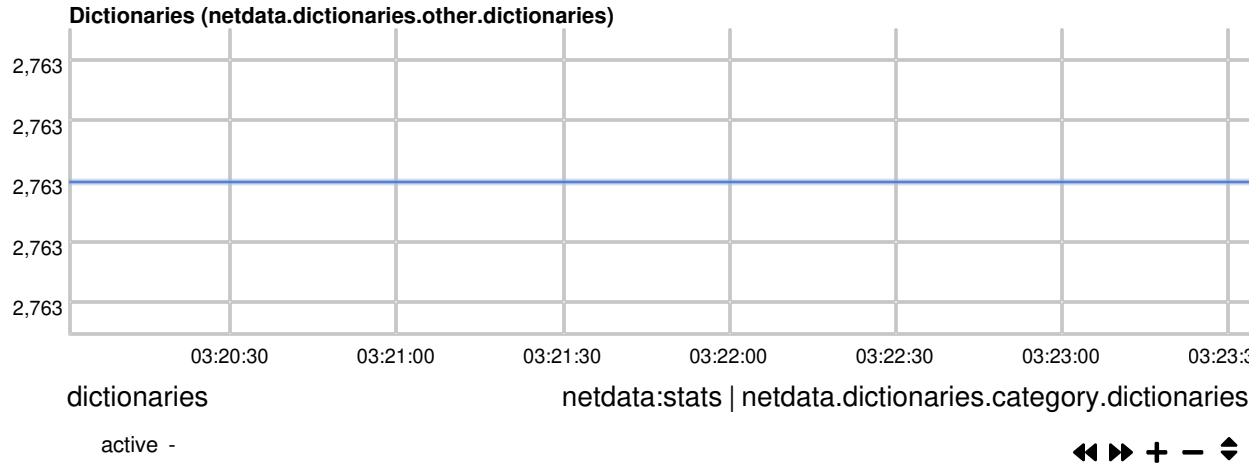


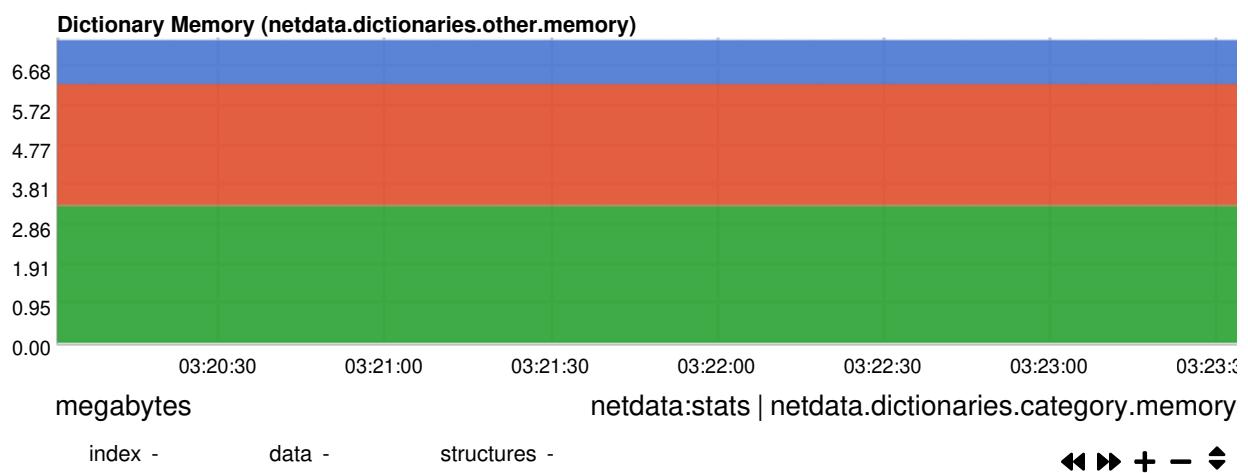
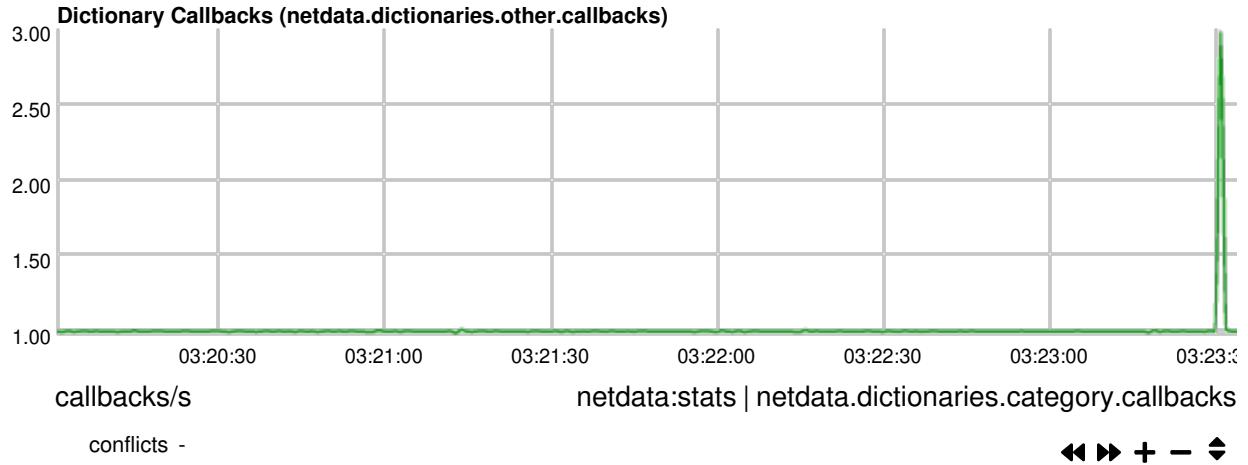


python.d

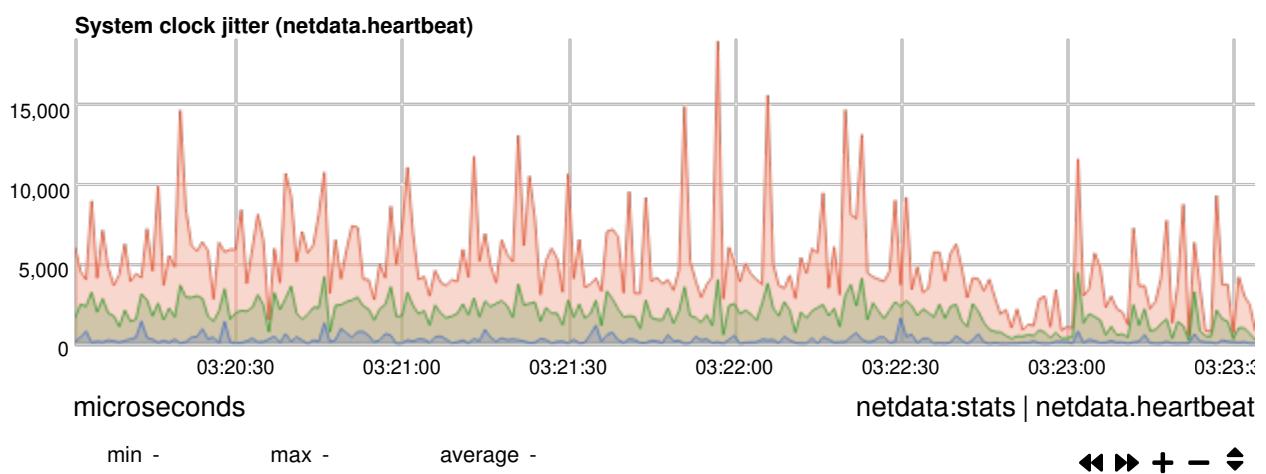


dictionaries

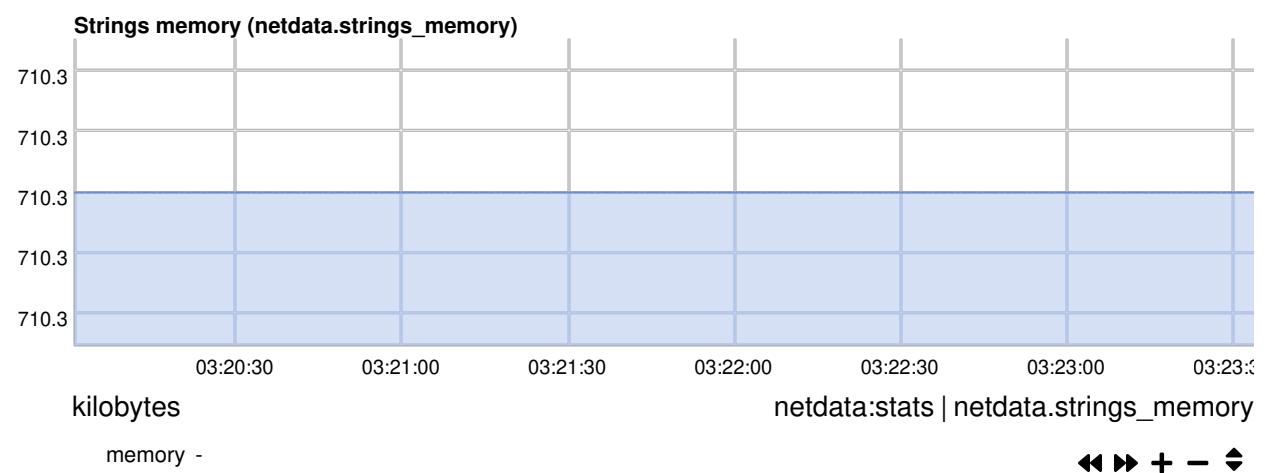
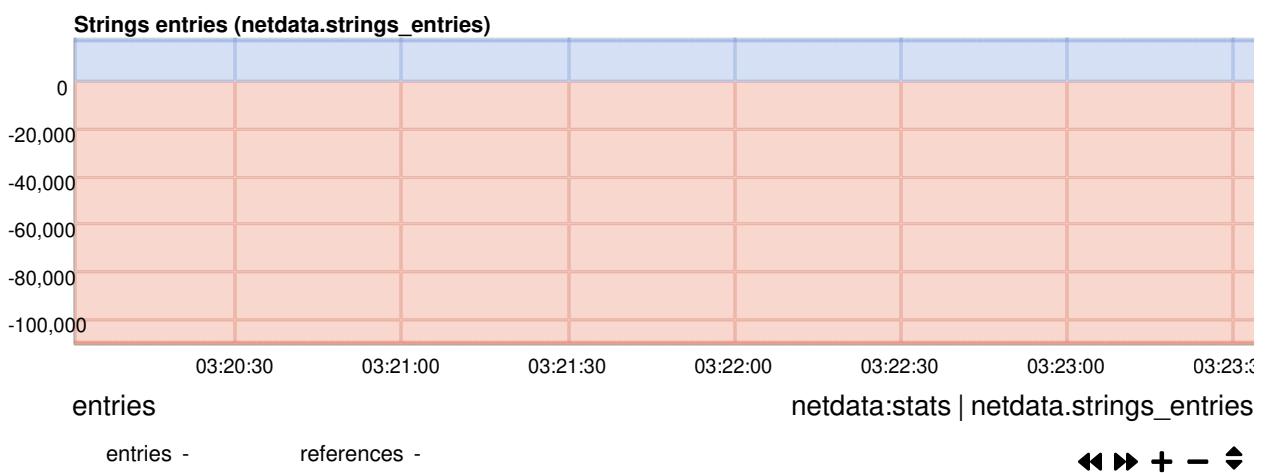
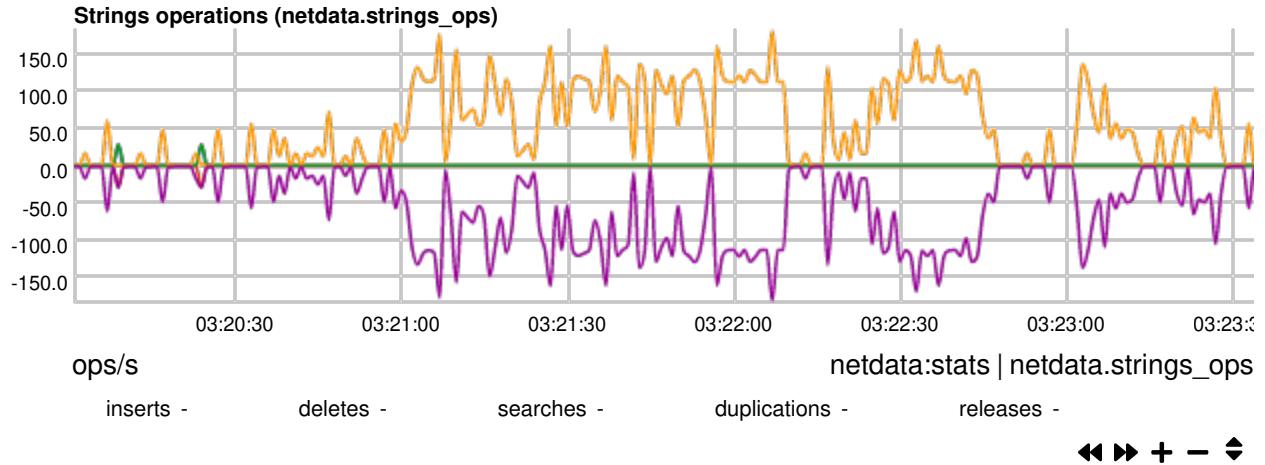




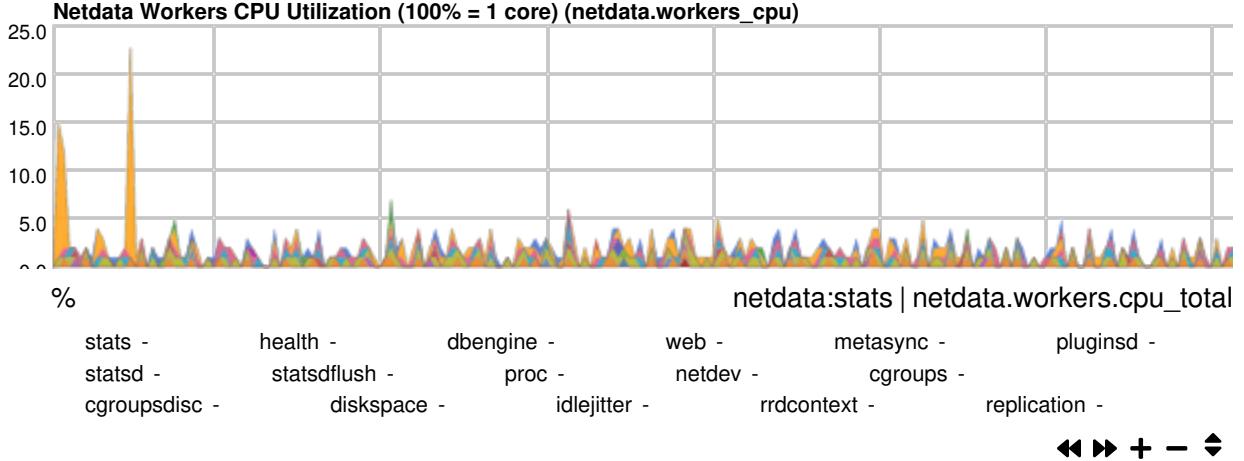
heartbeat



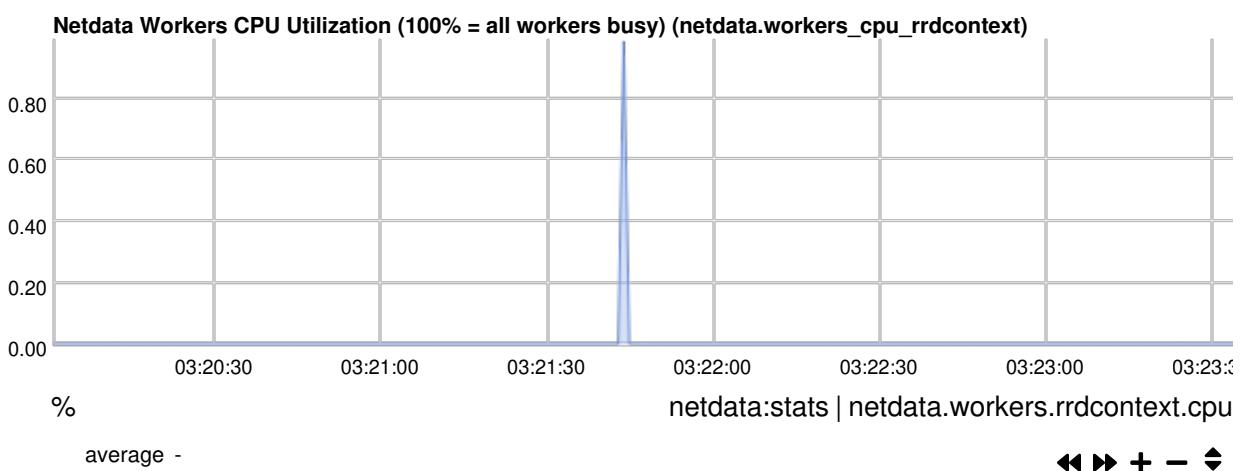
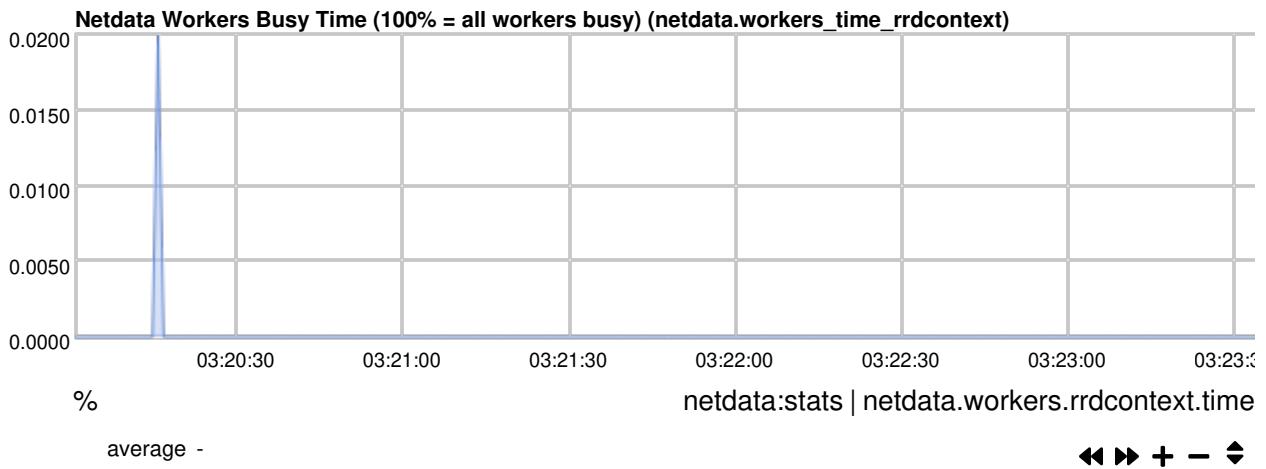
strings

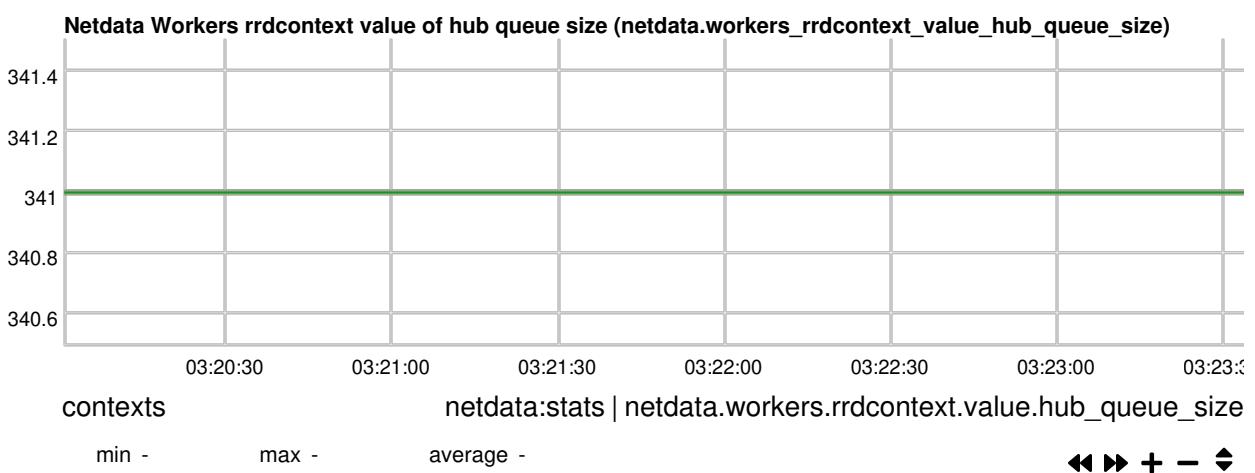
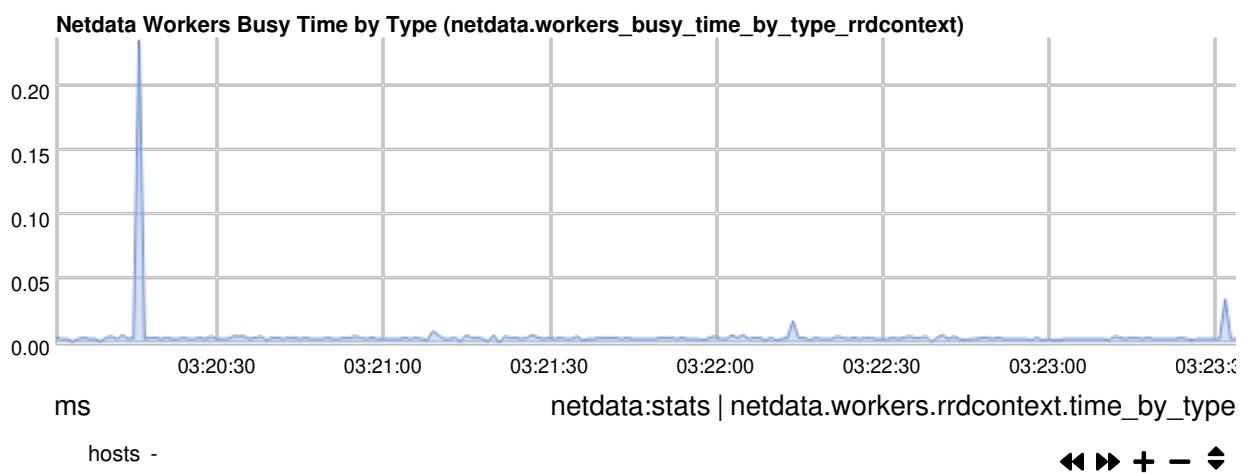
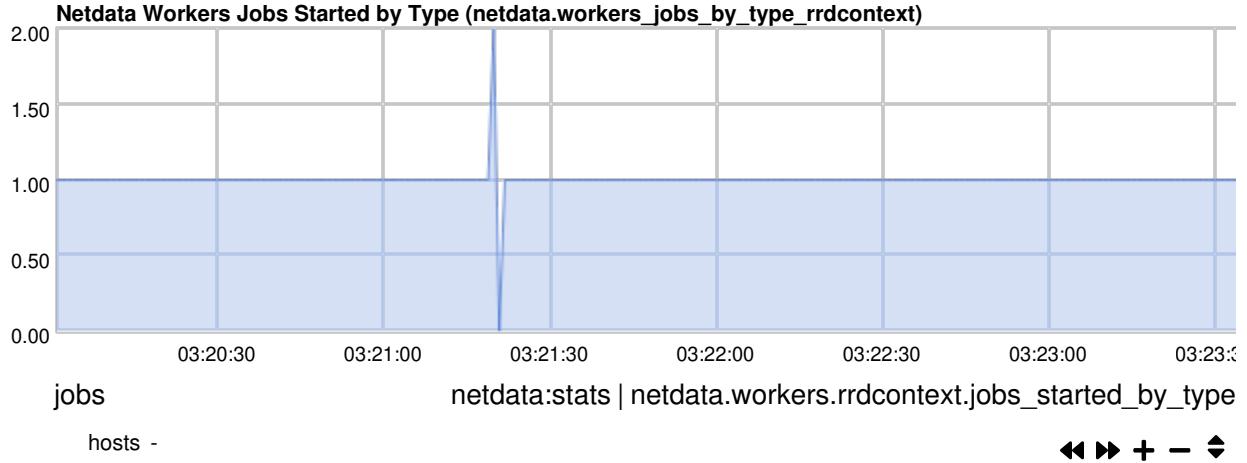


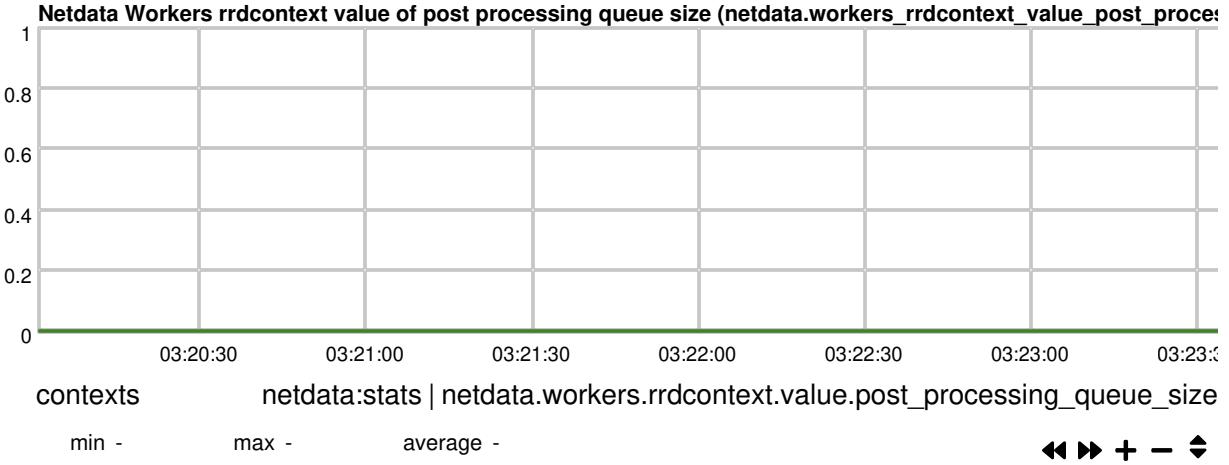
workers



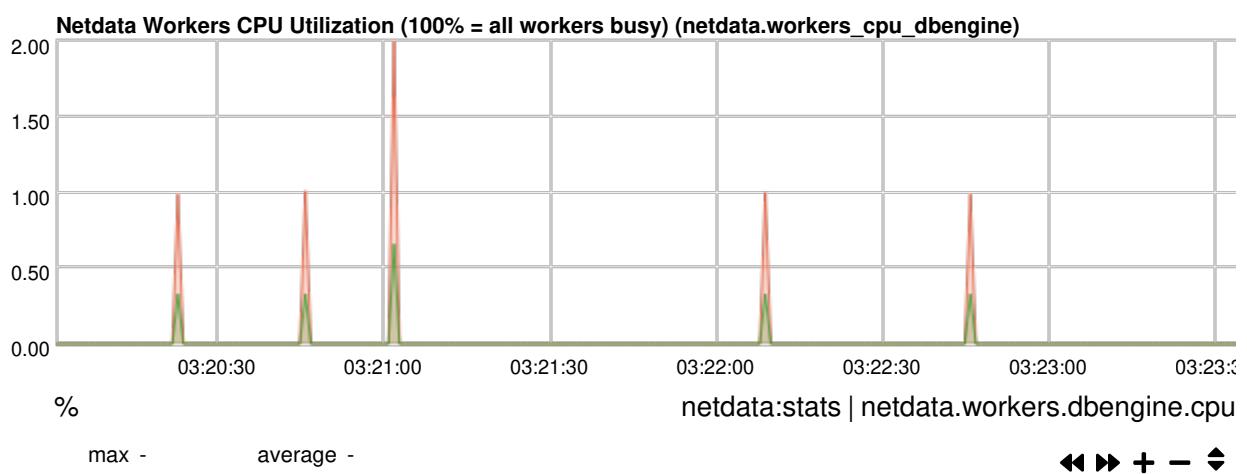
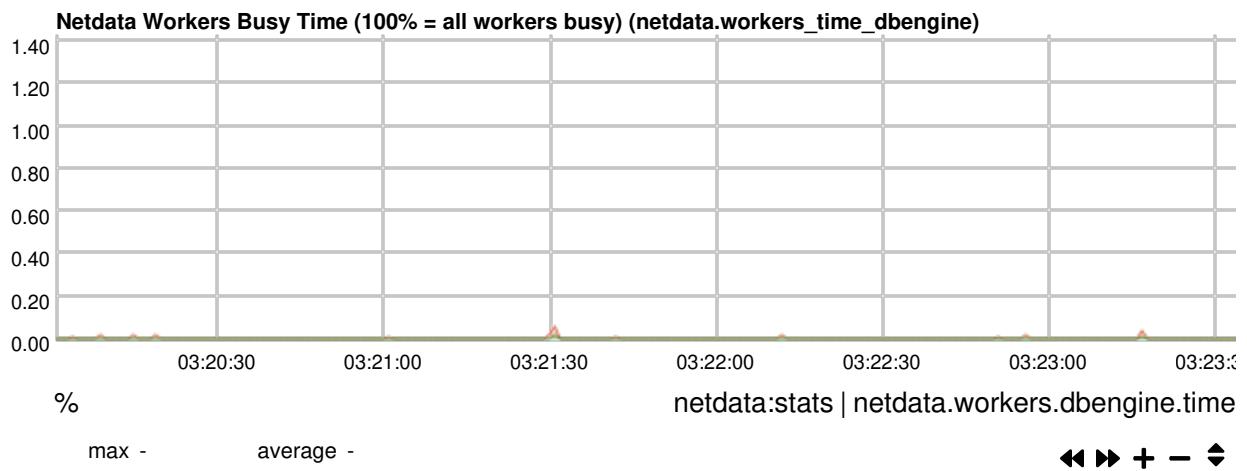
workers contexts

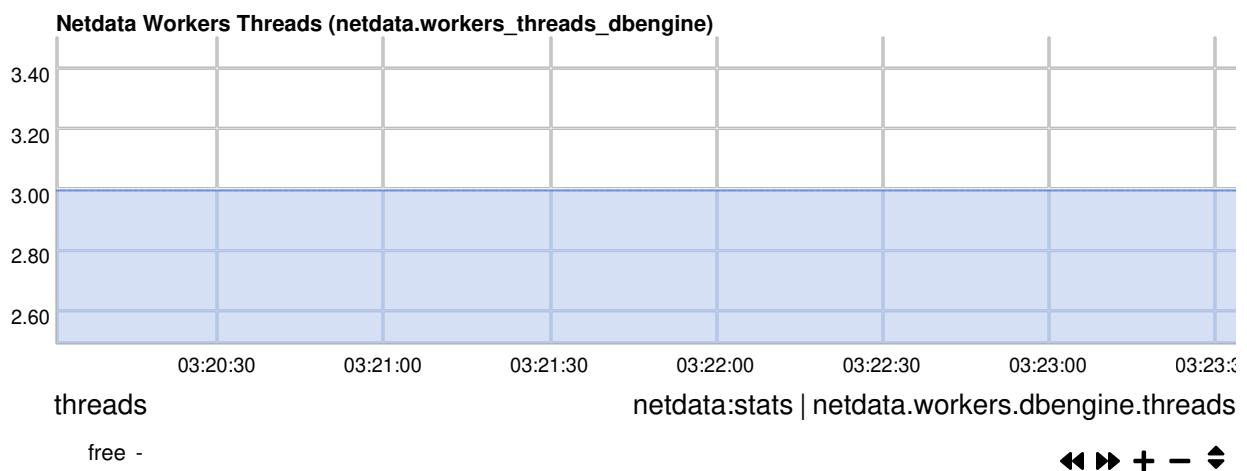
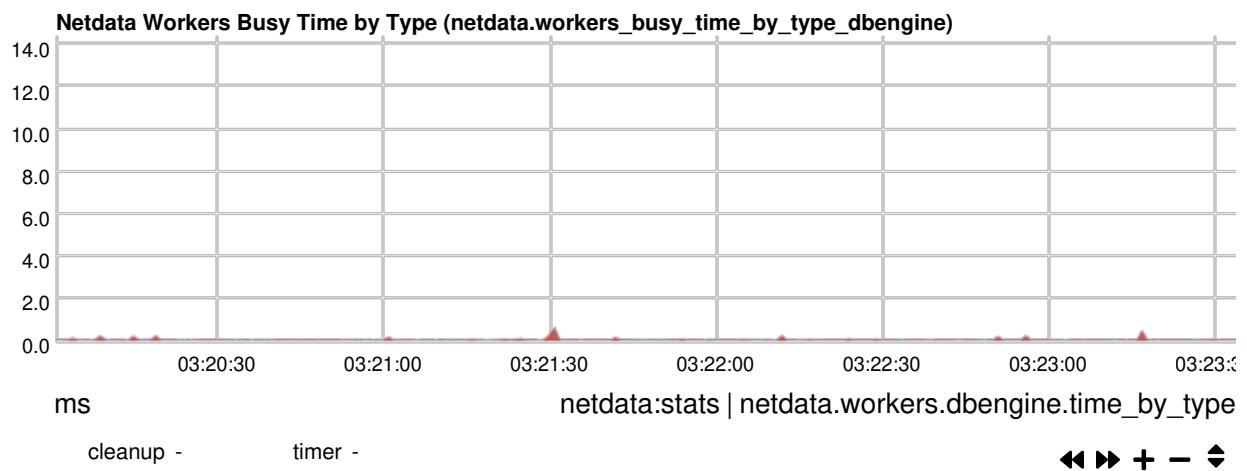
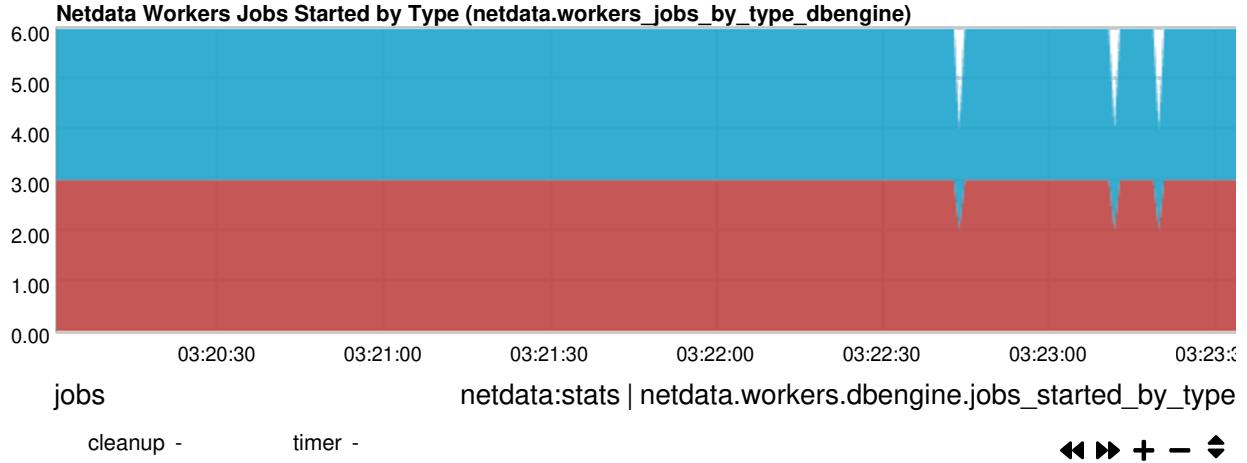




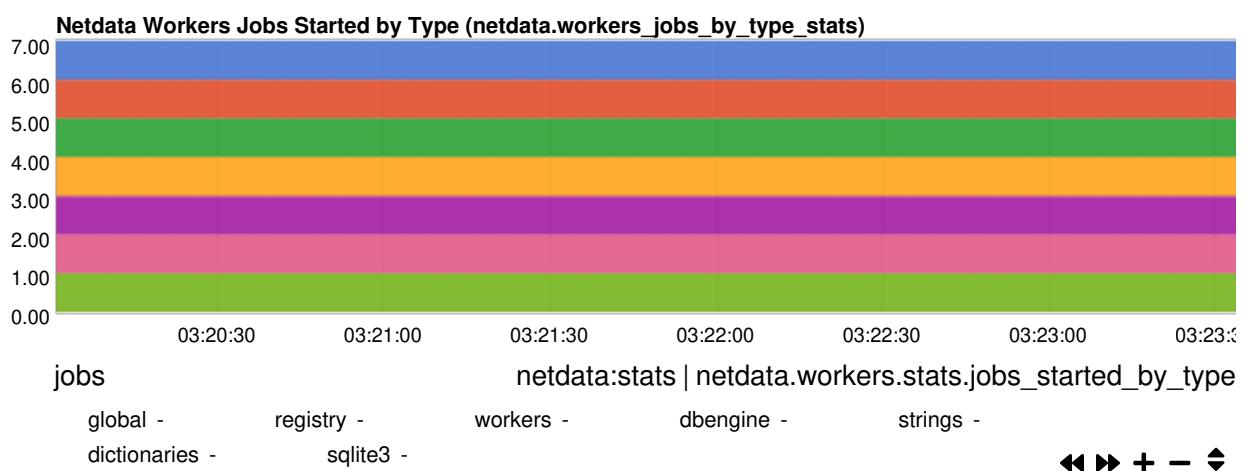
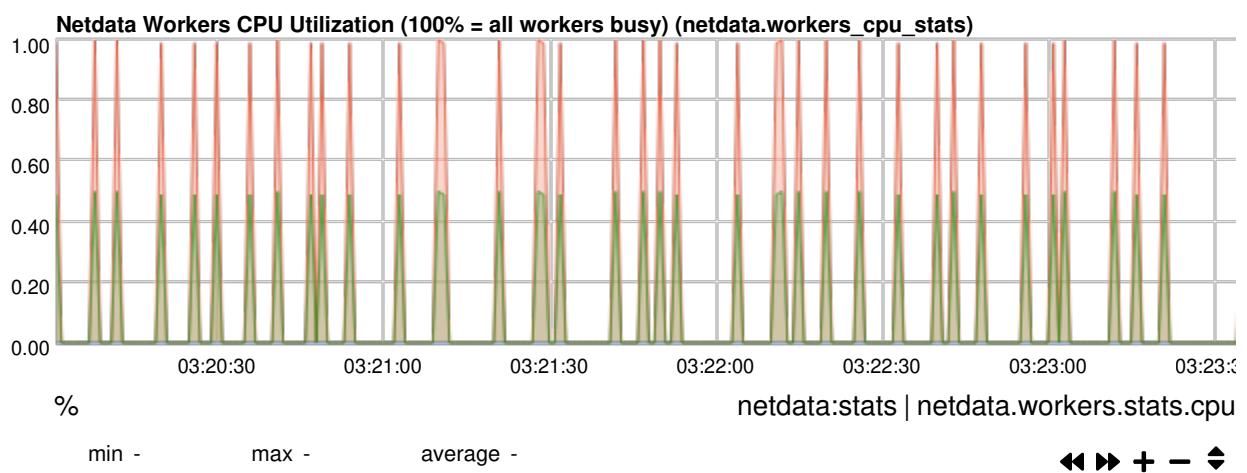
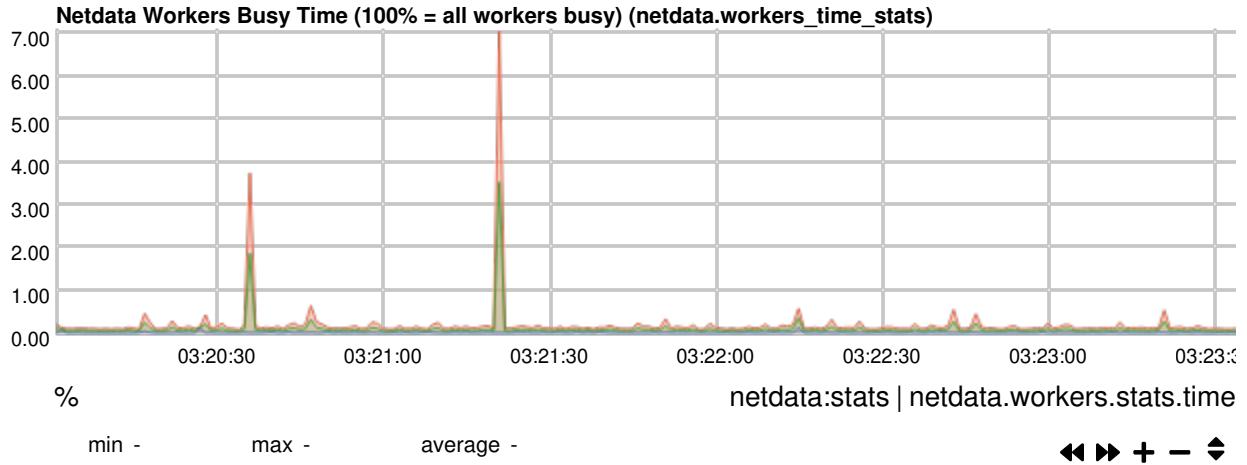


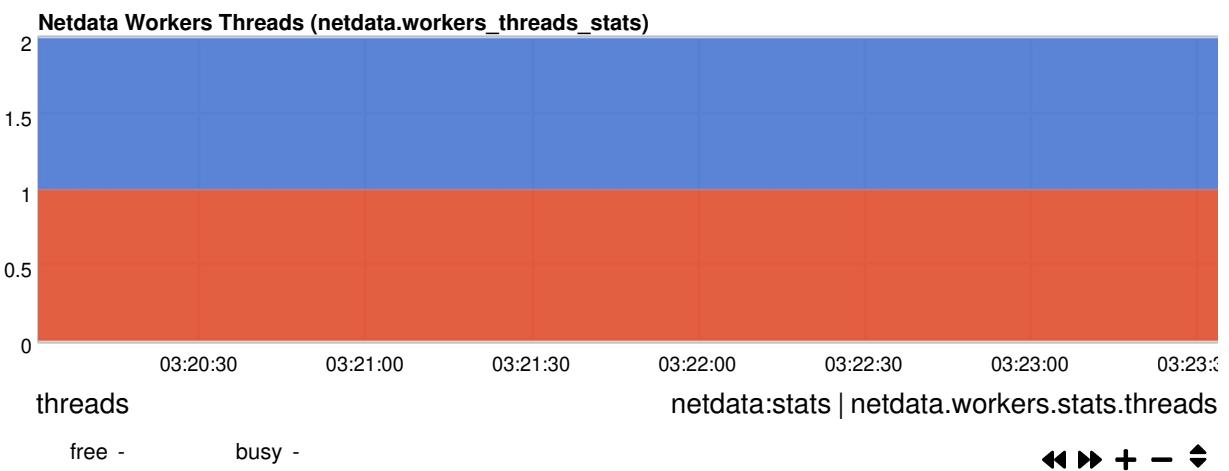
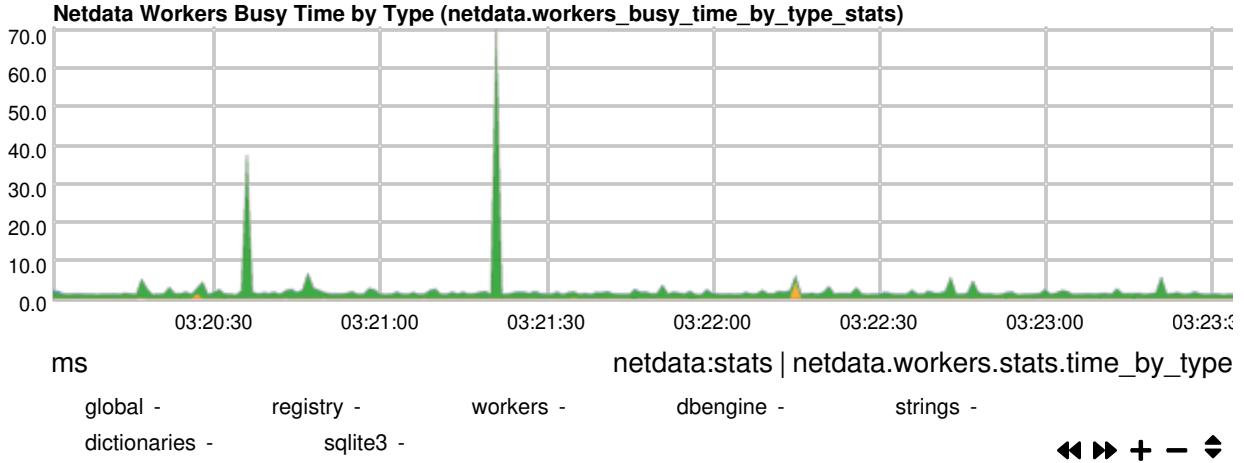
workers dbengine instances



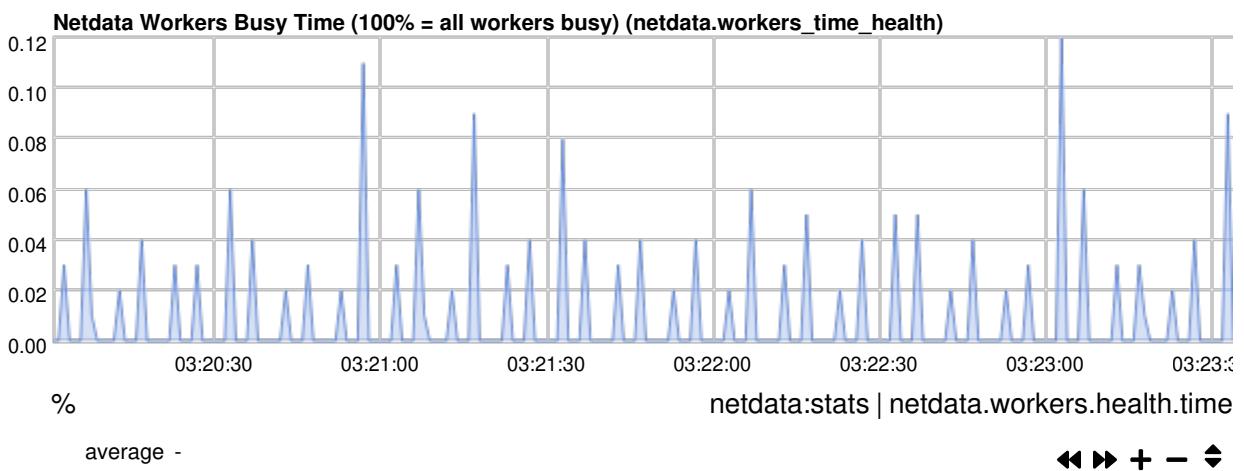


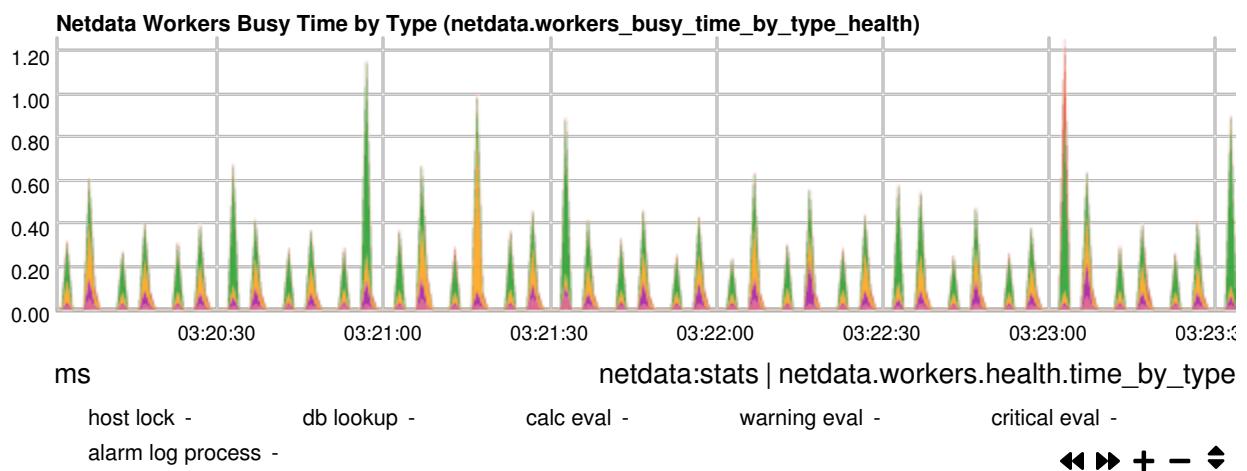
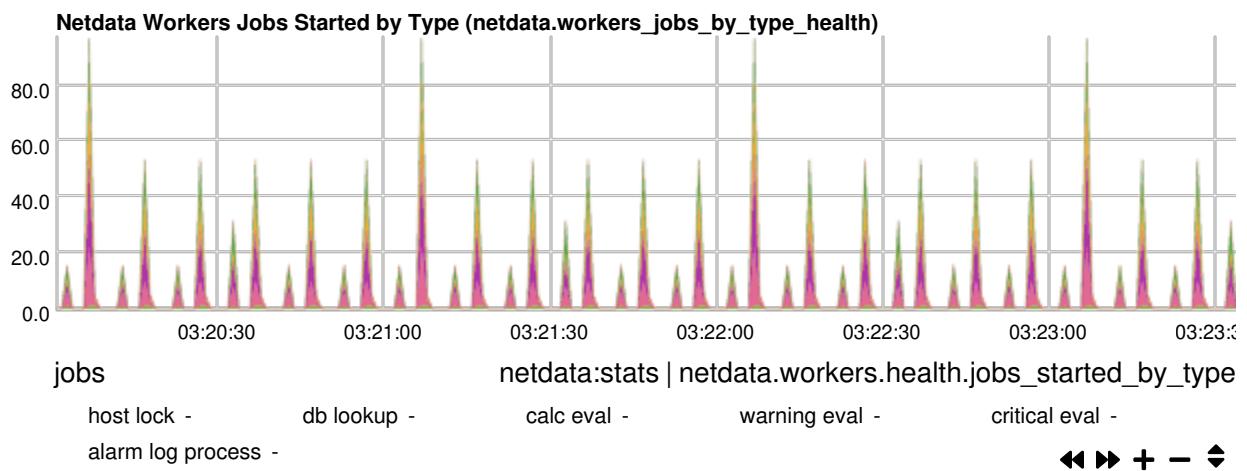
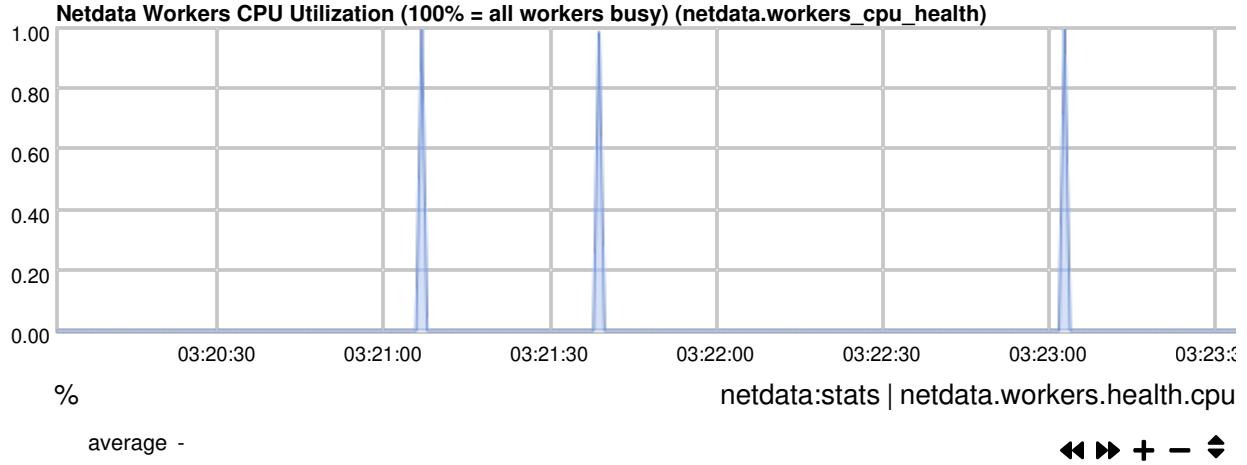
workers global statistics



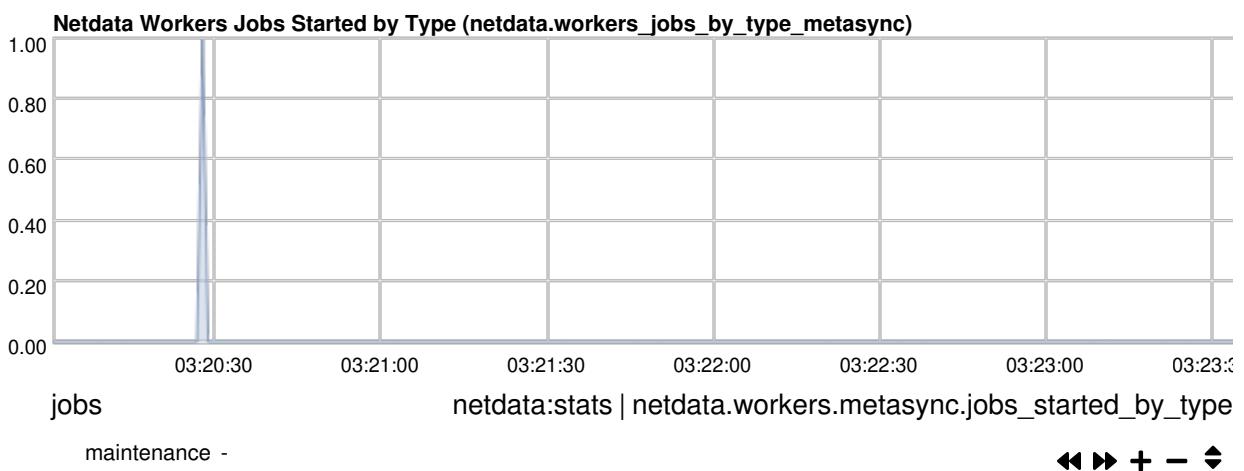
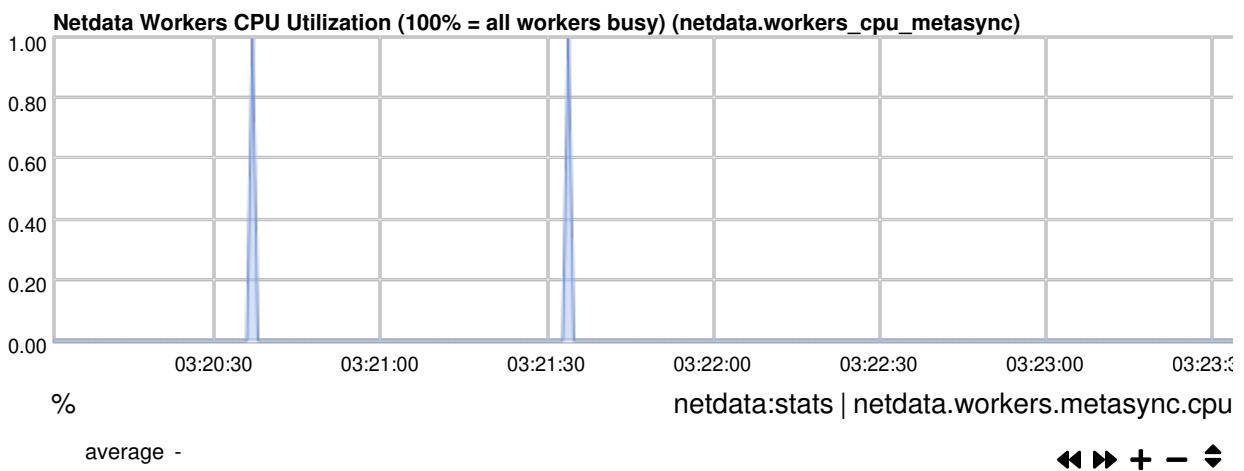
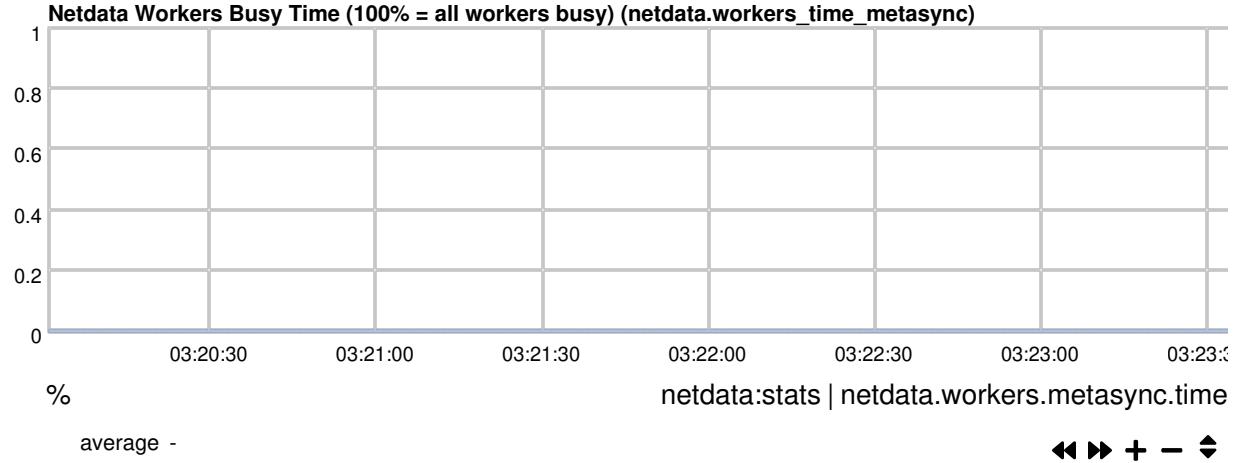


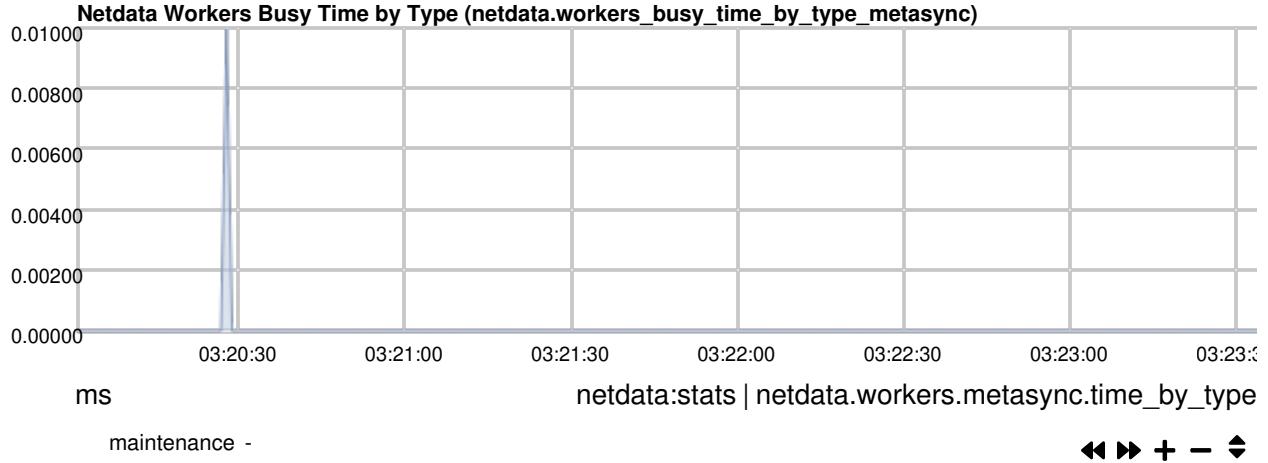
workers health alarms



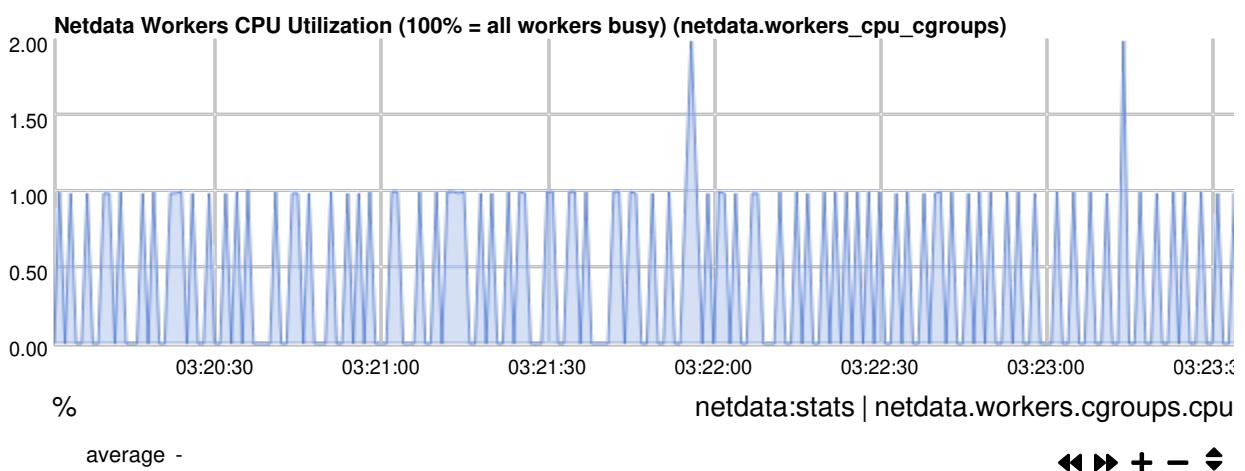
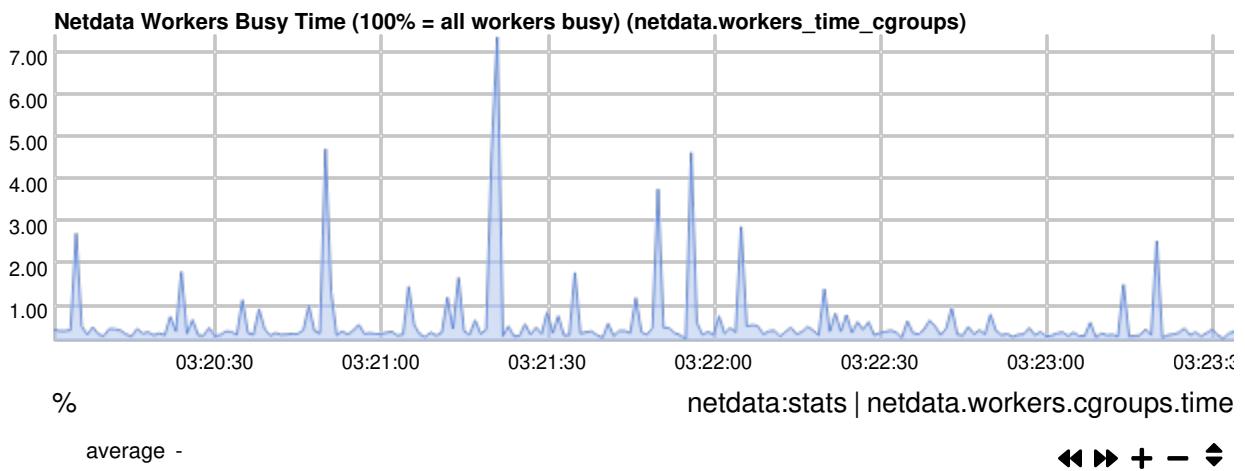


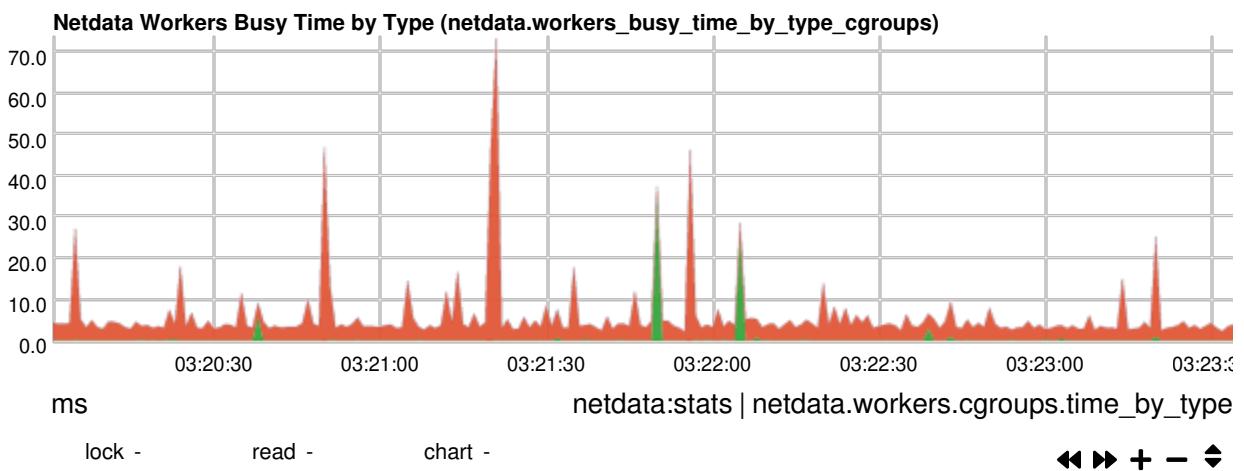
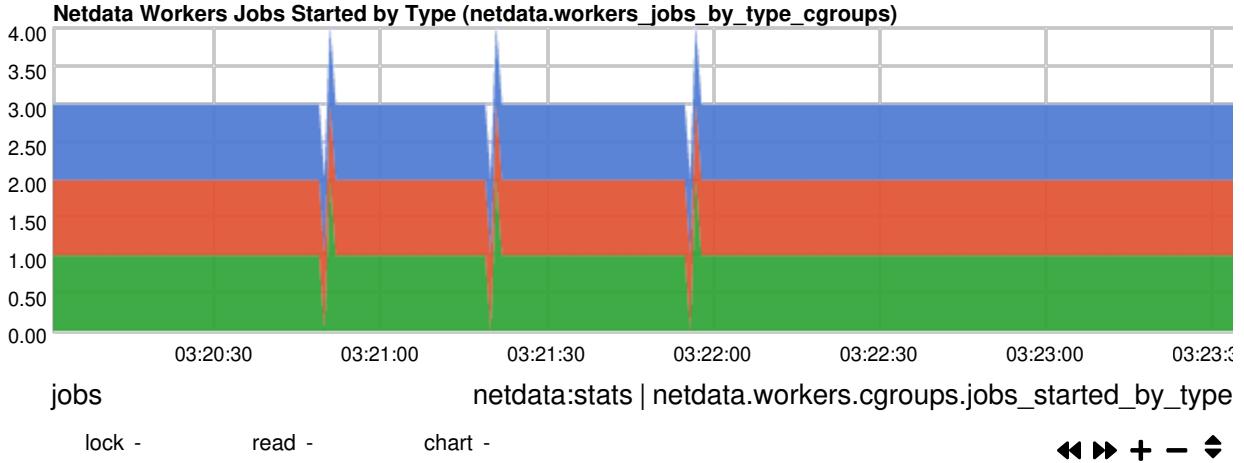
workers metadata sync



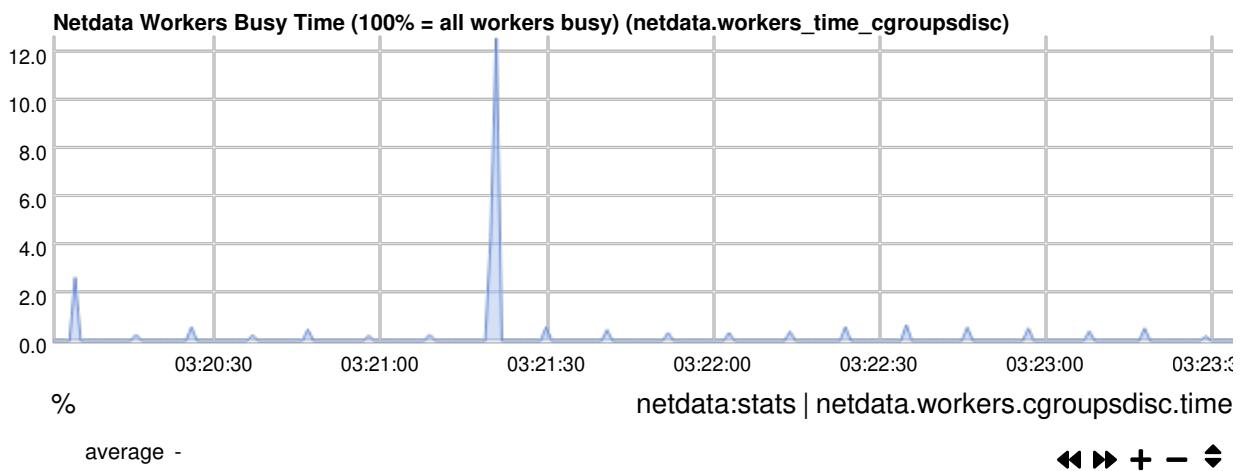


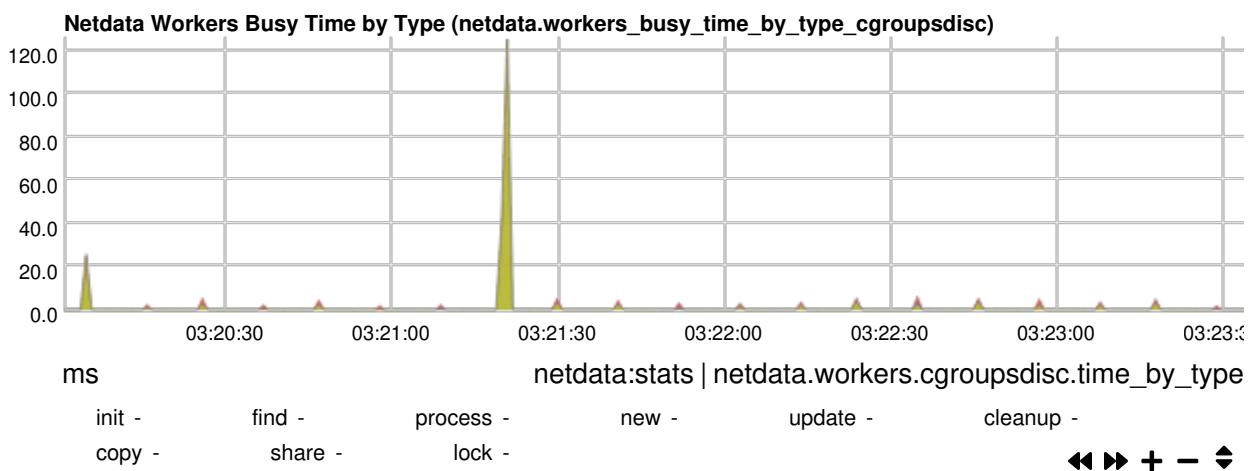
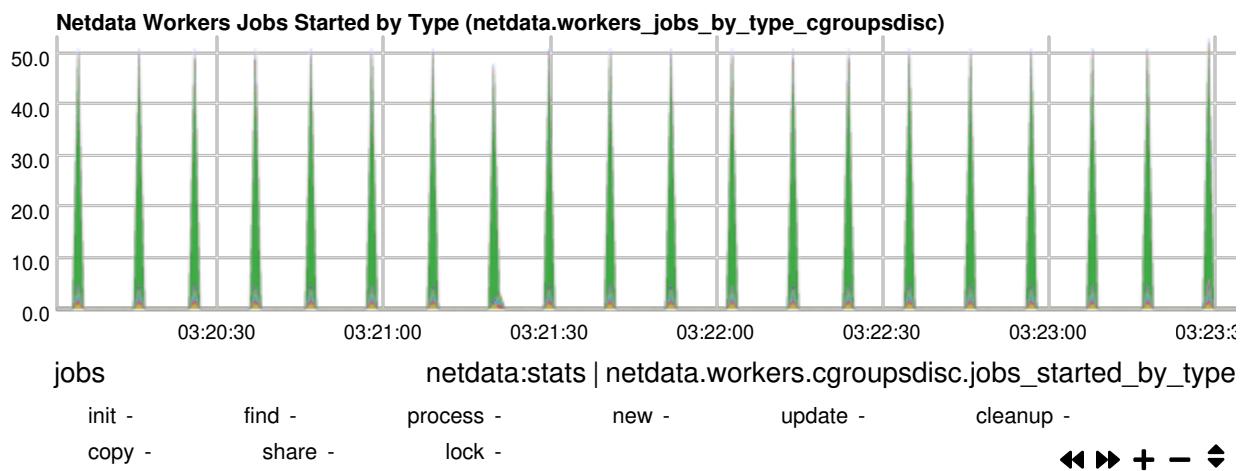
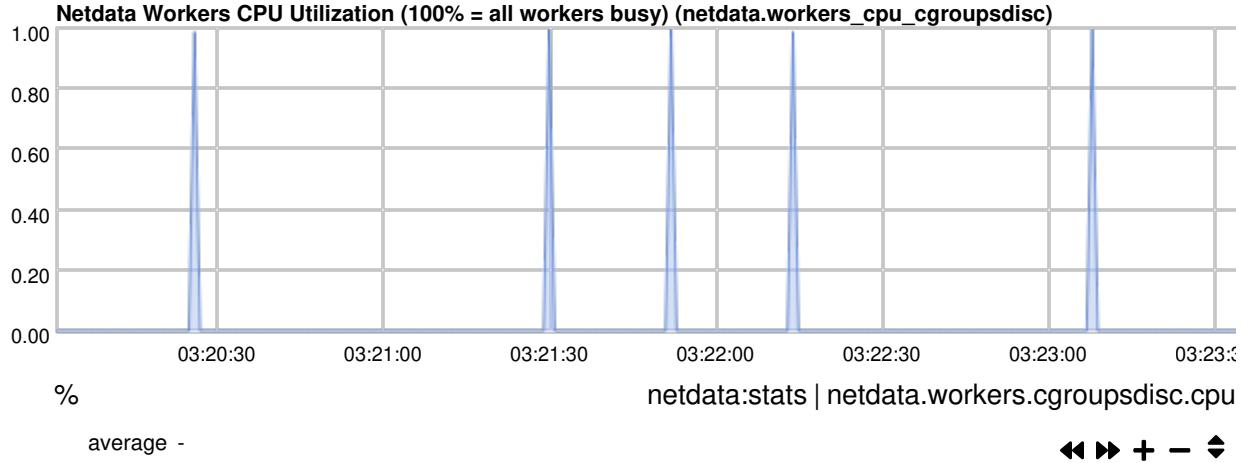
workers plugin cgroups



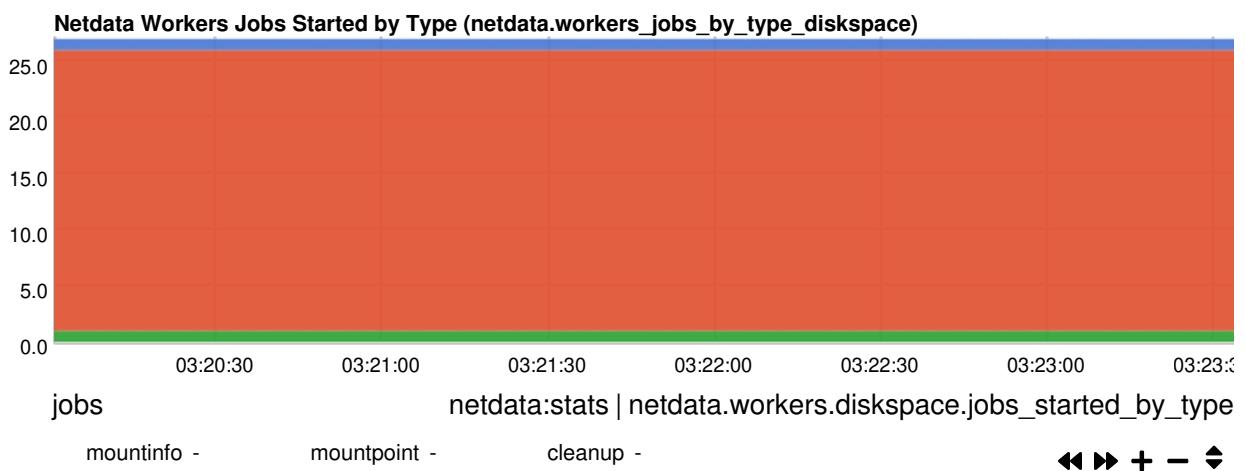
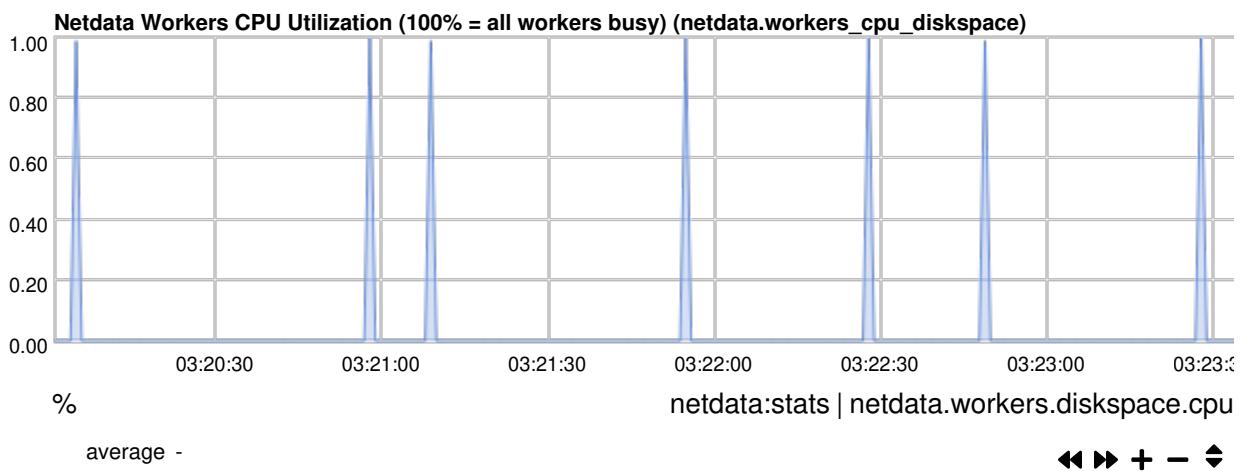
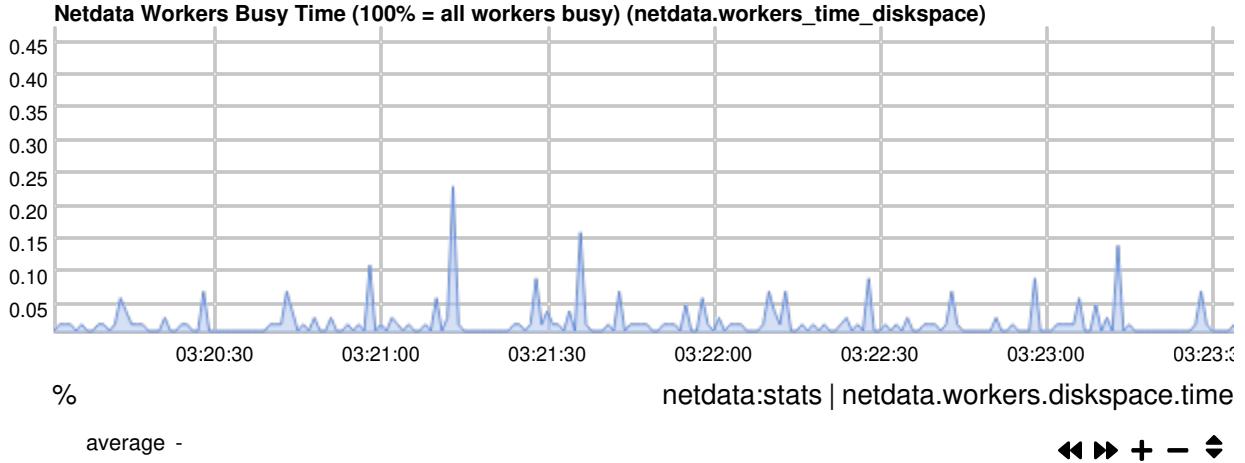


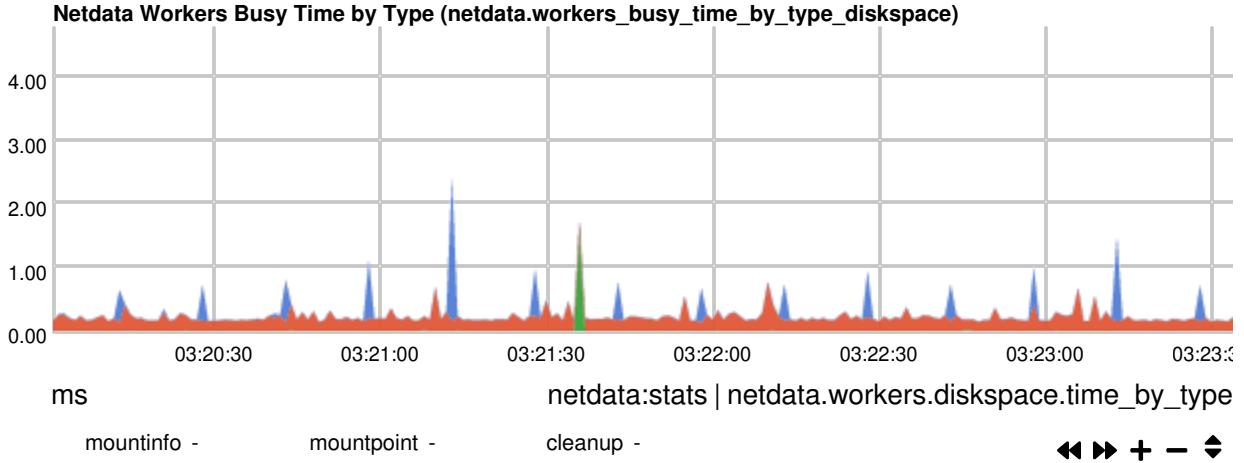
workers plugin cgroups find



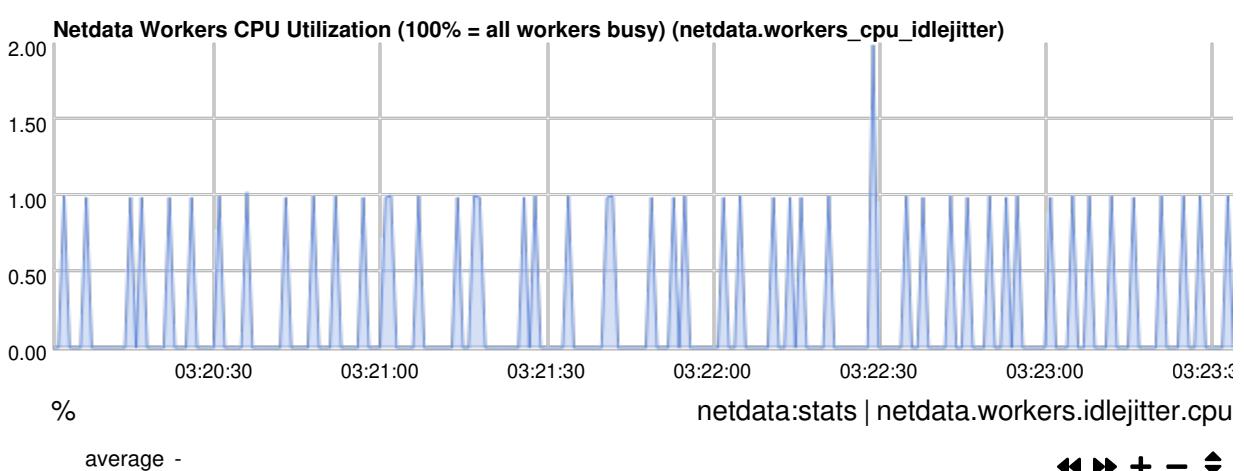
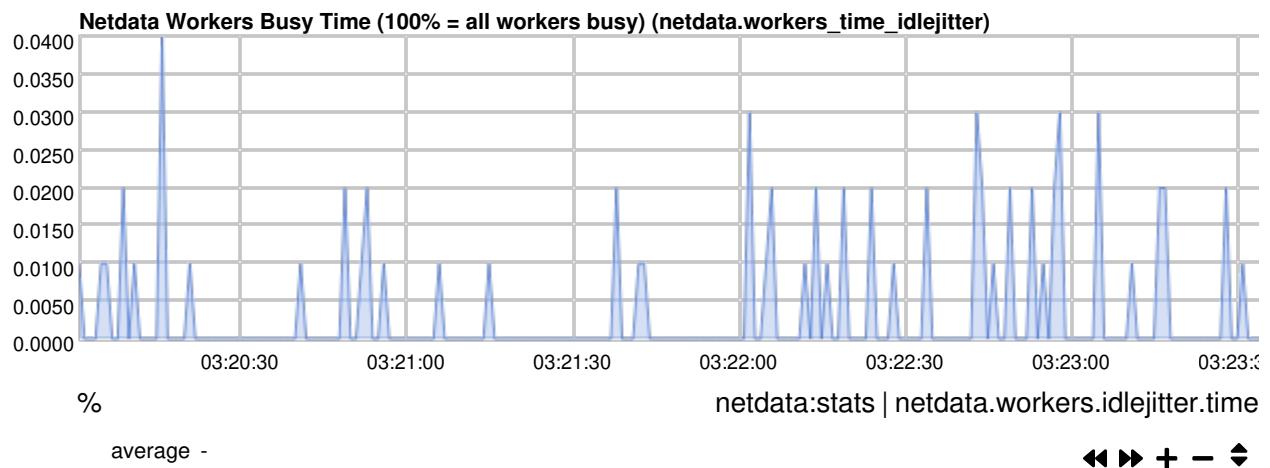


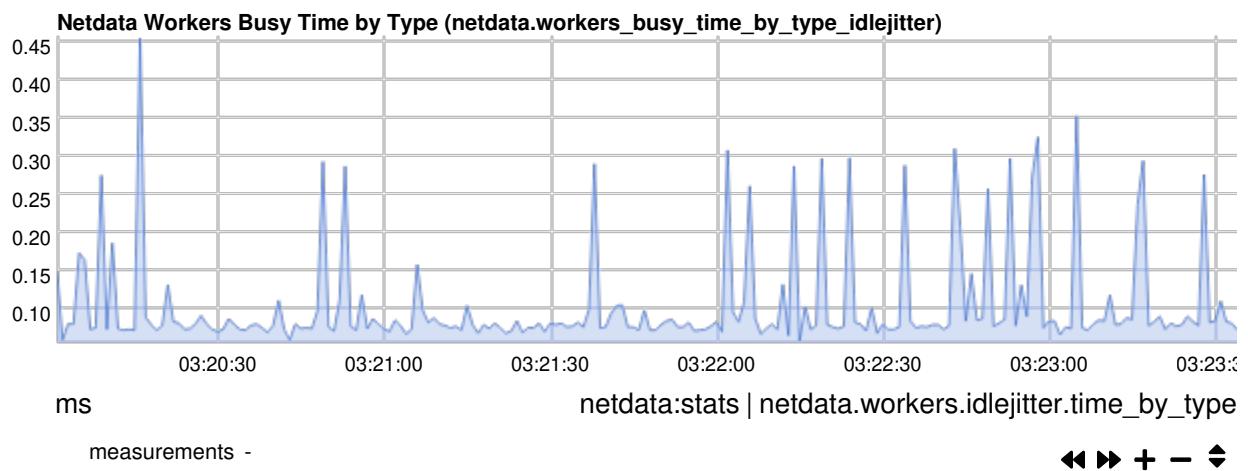
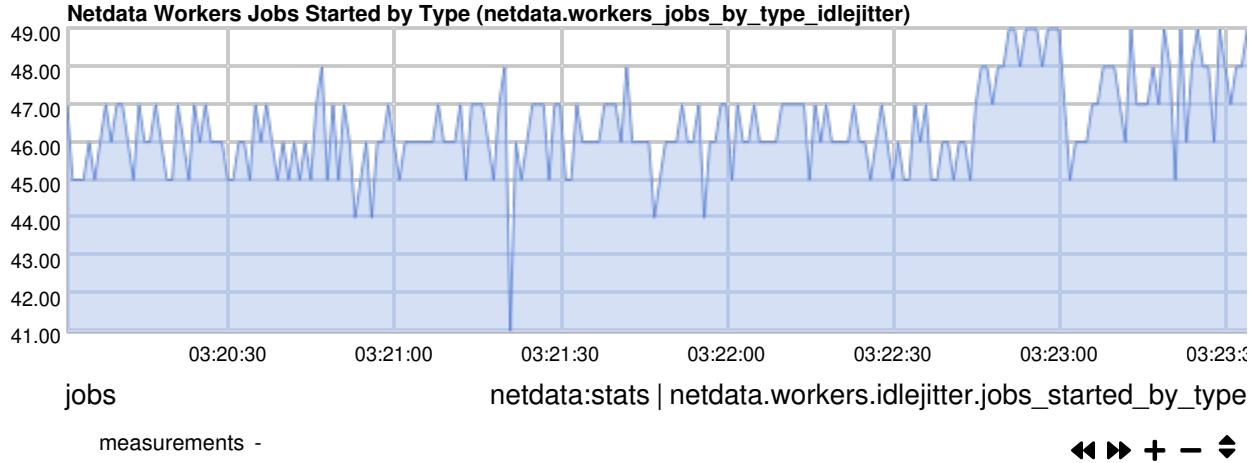
workers plugin diskspace



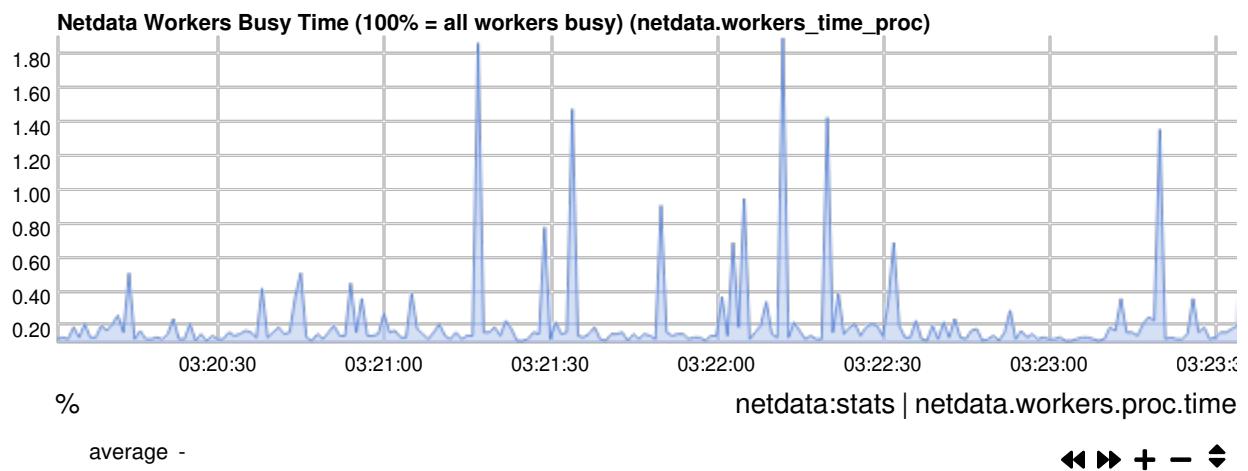


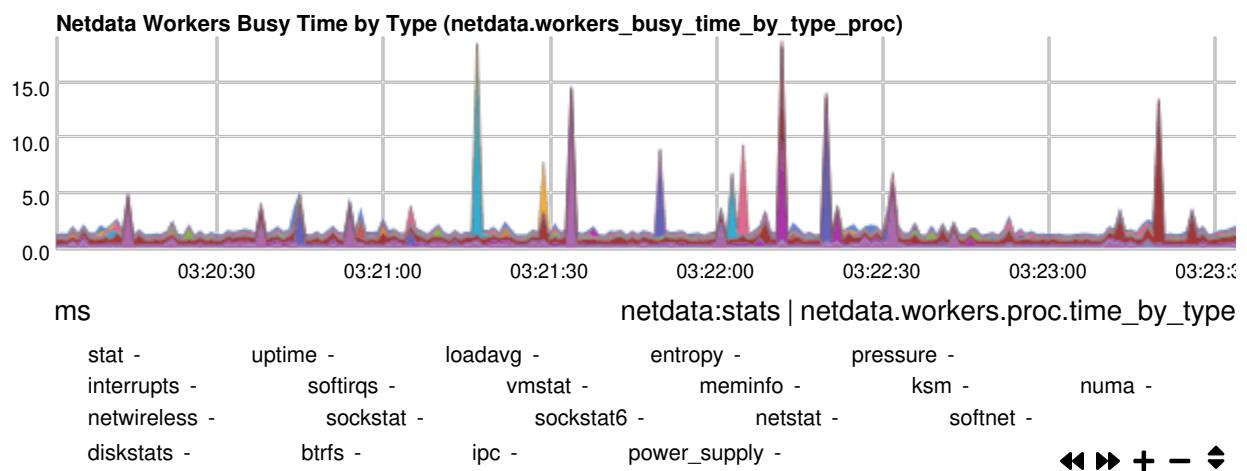
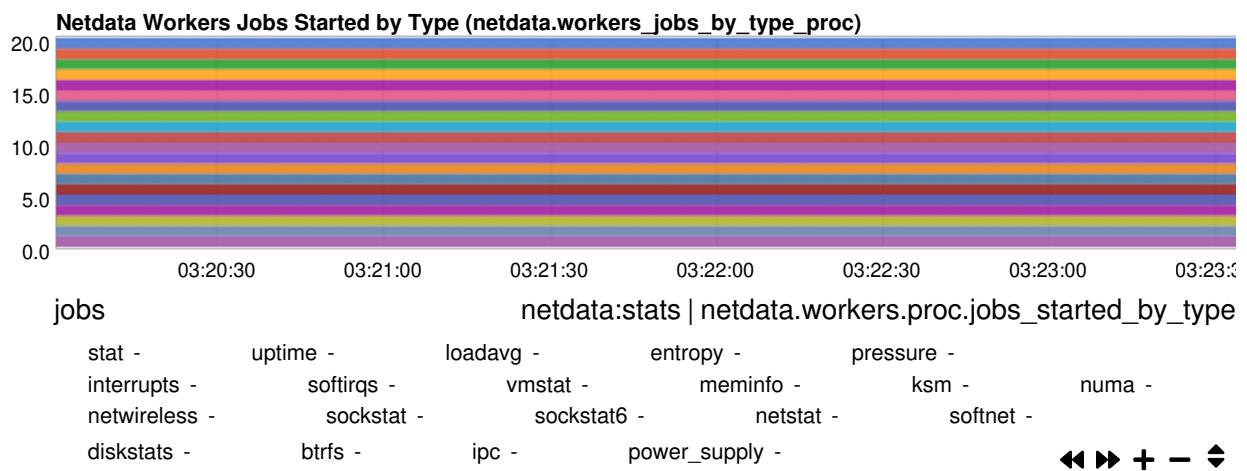
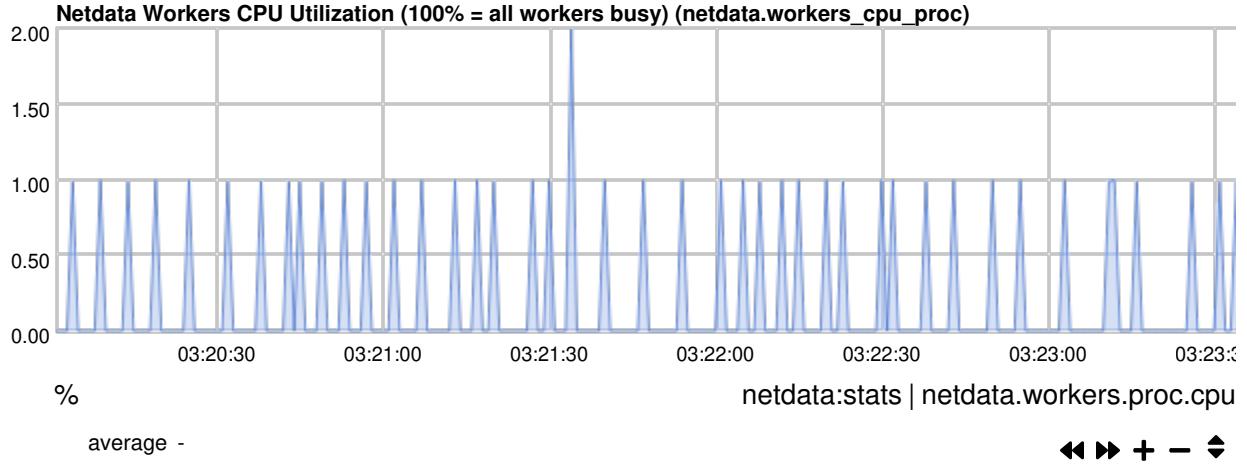
workers plugin idlejitter



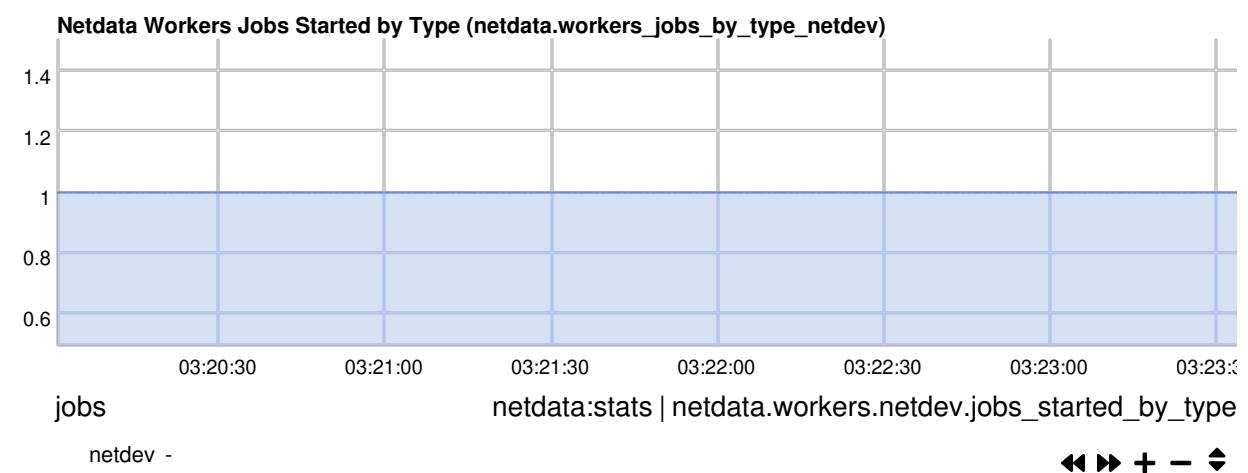
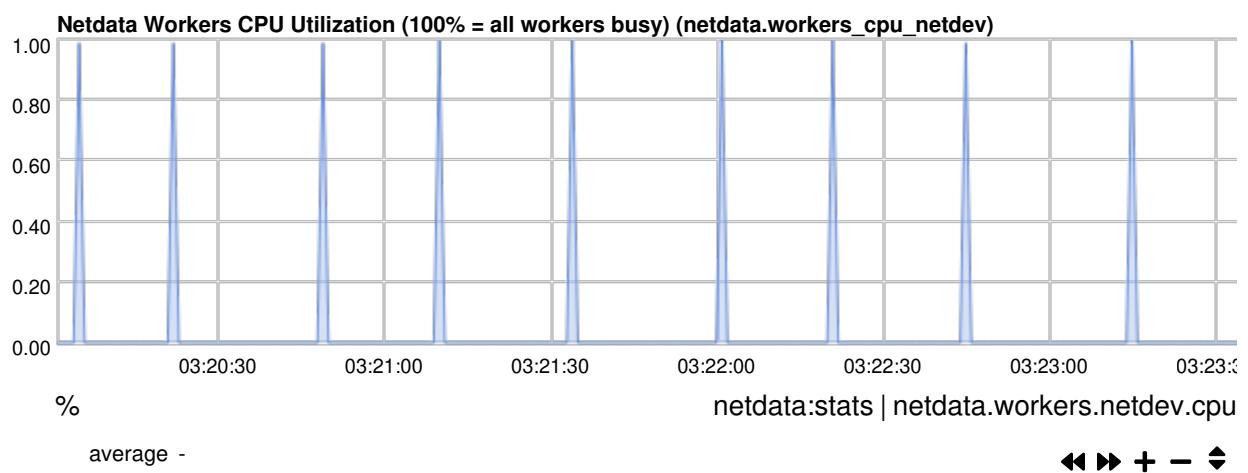
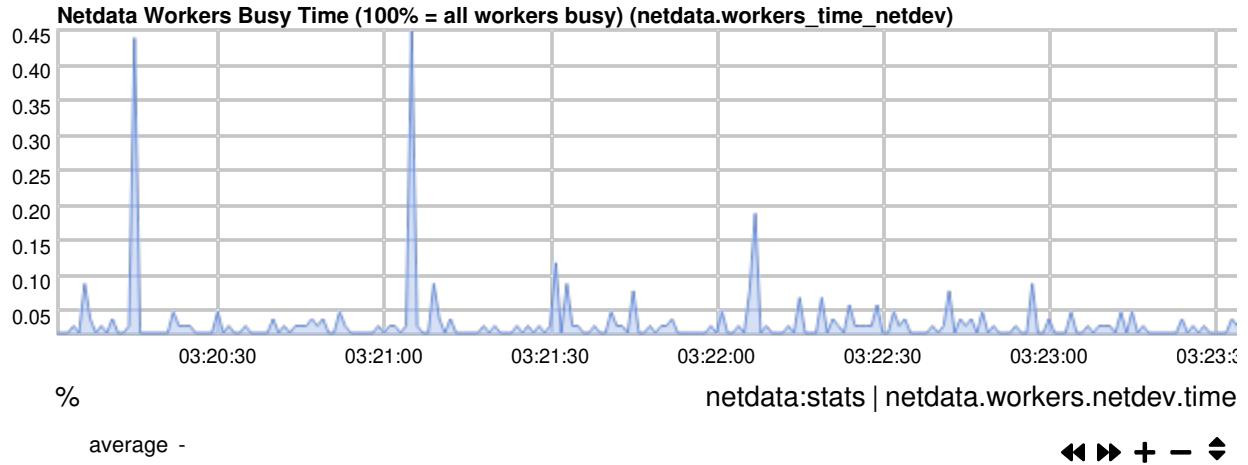


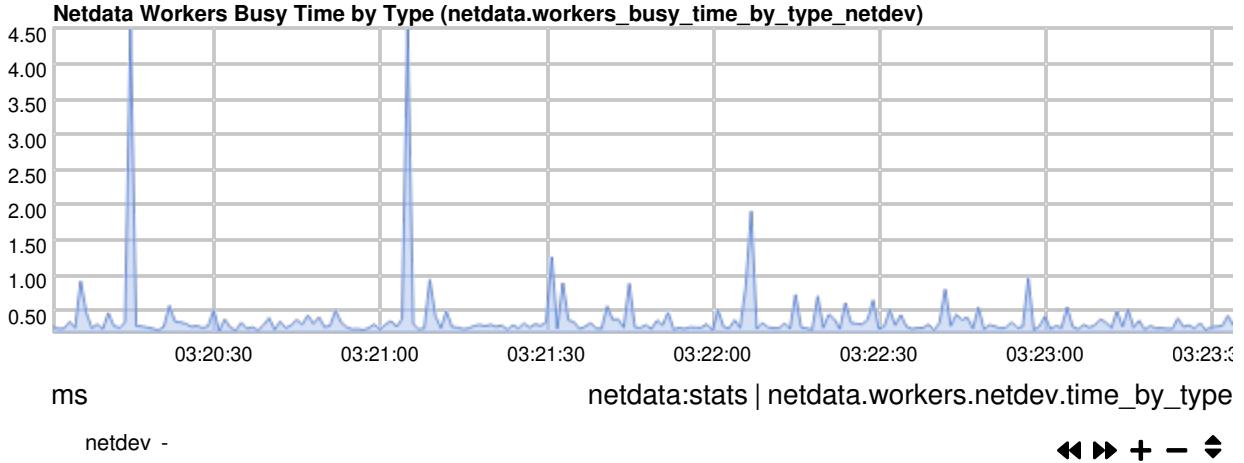
workers plugin proc



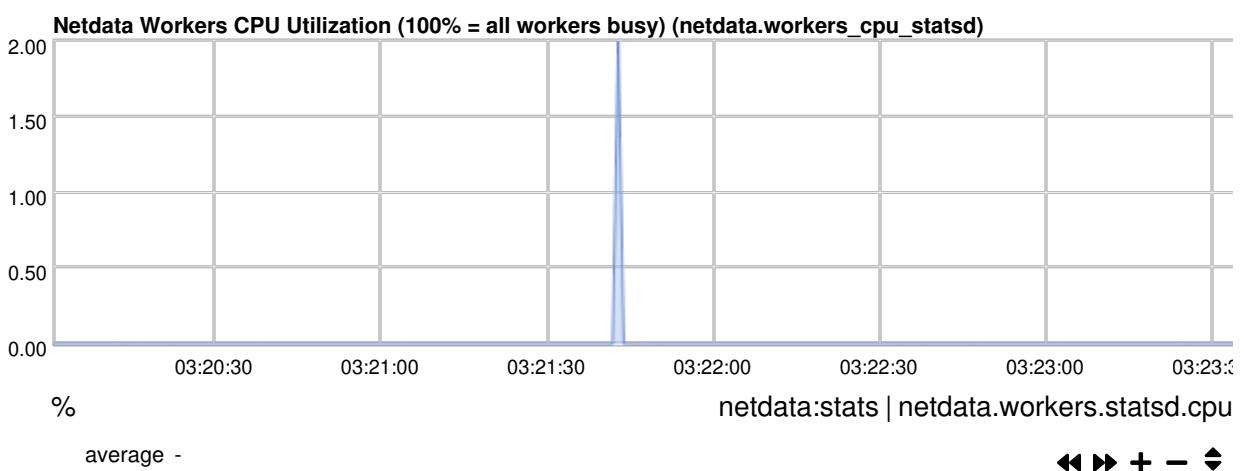
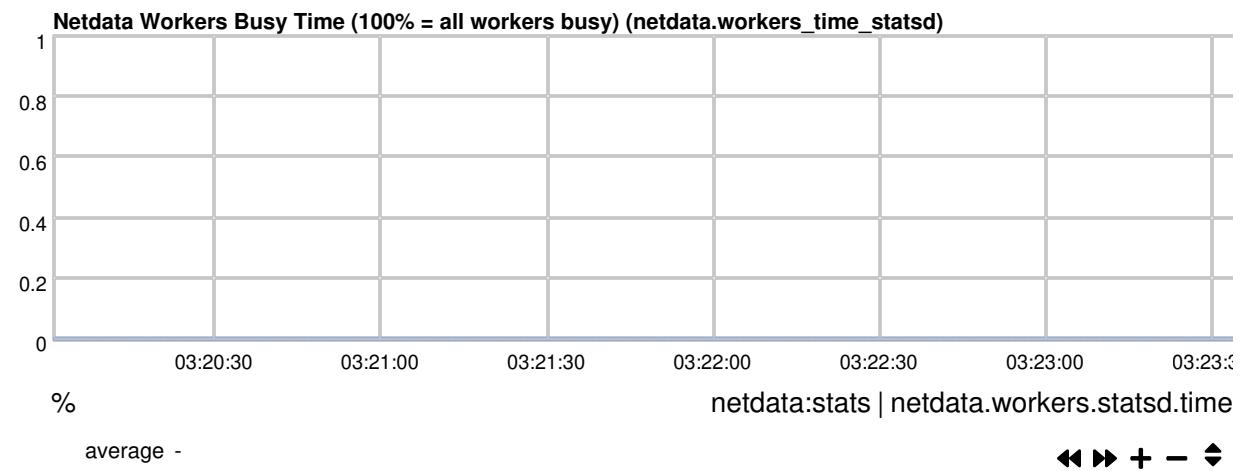


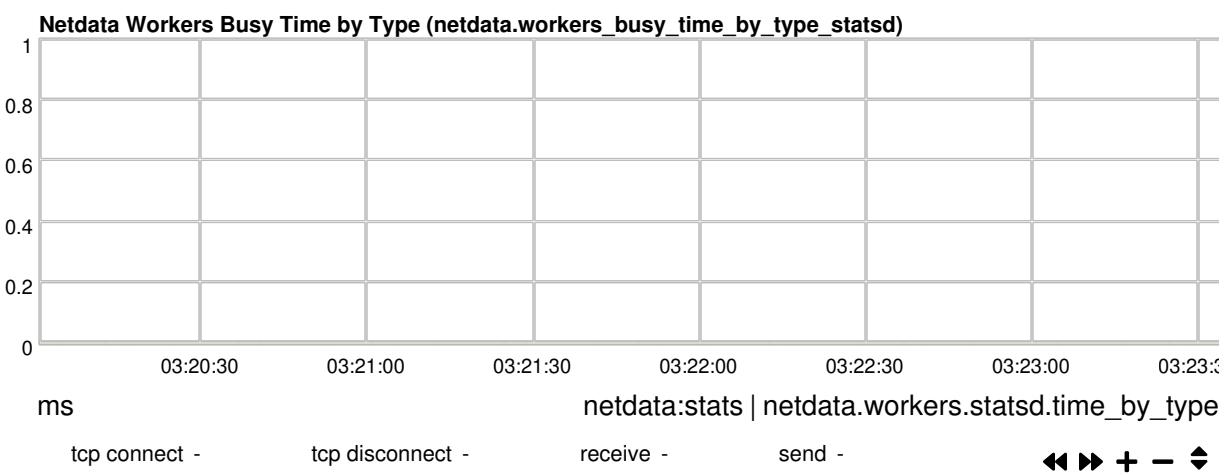
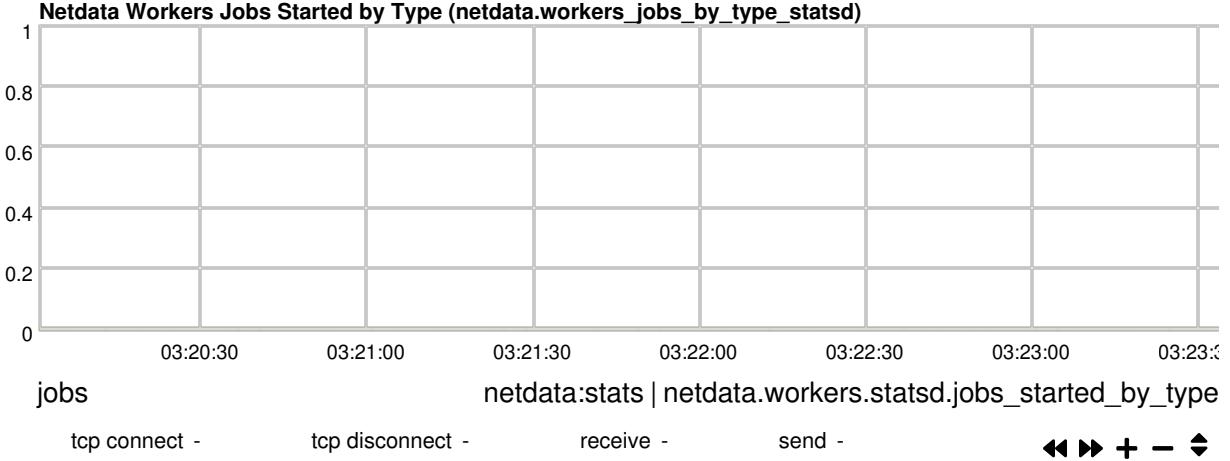
workers plugin proc netdev



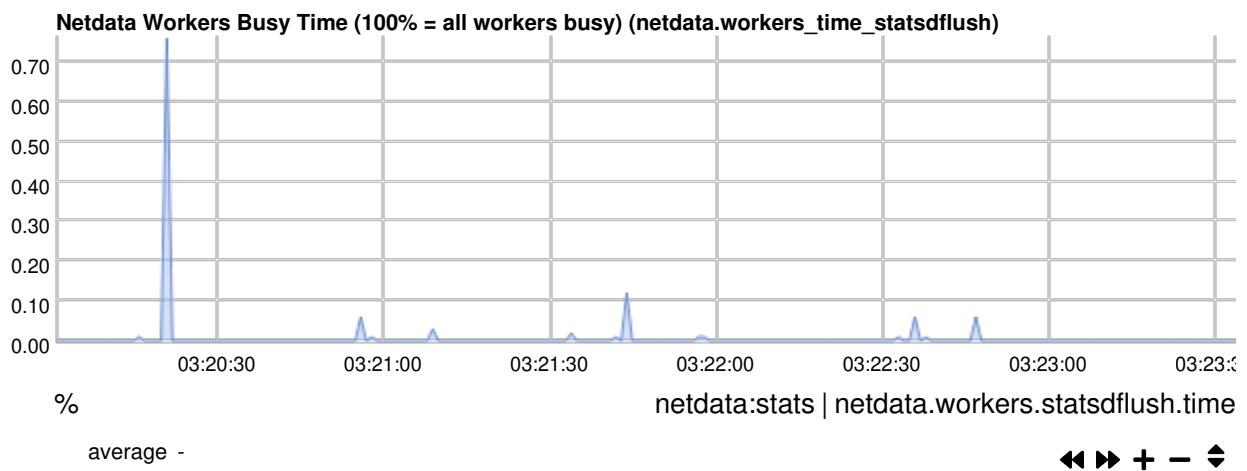


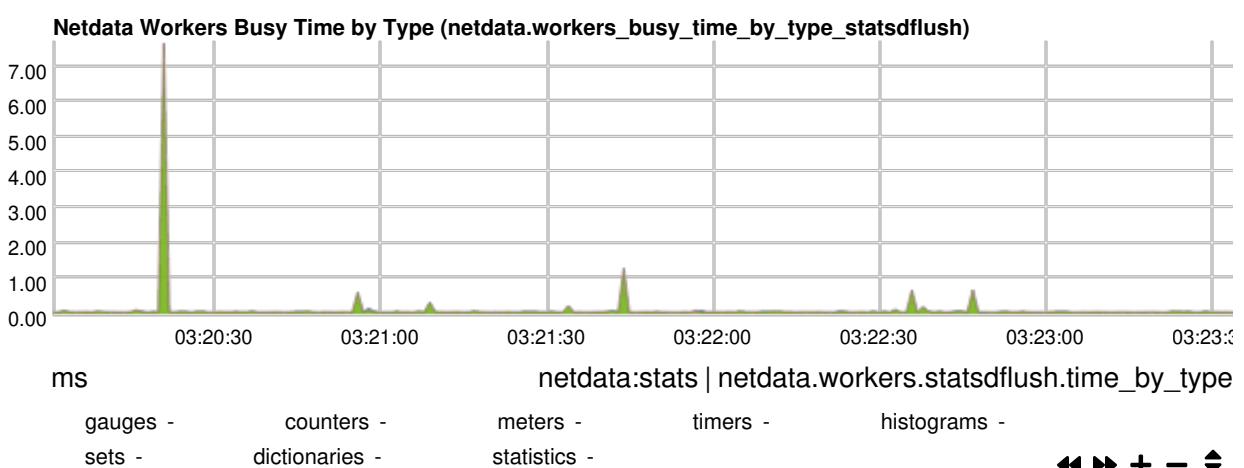
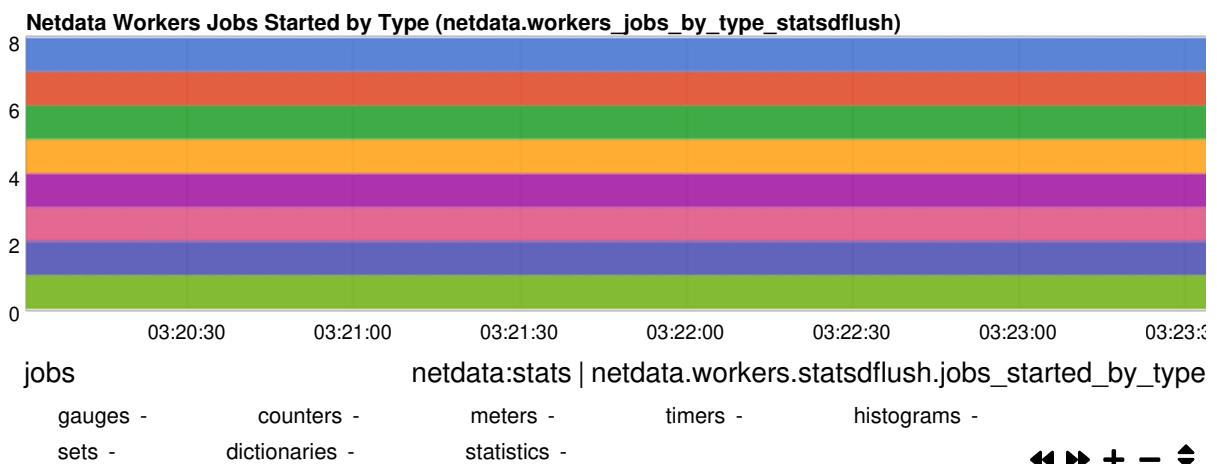
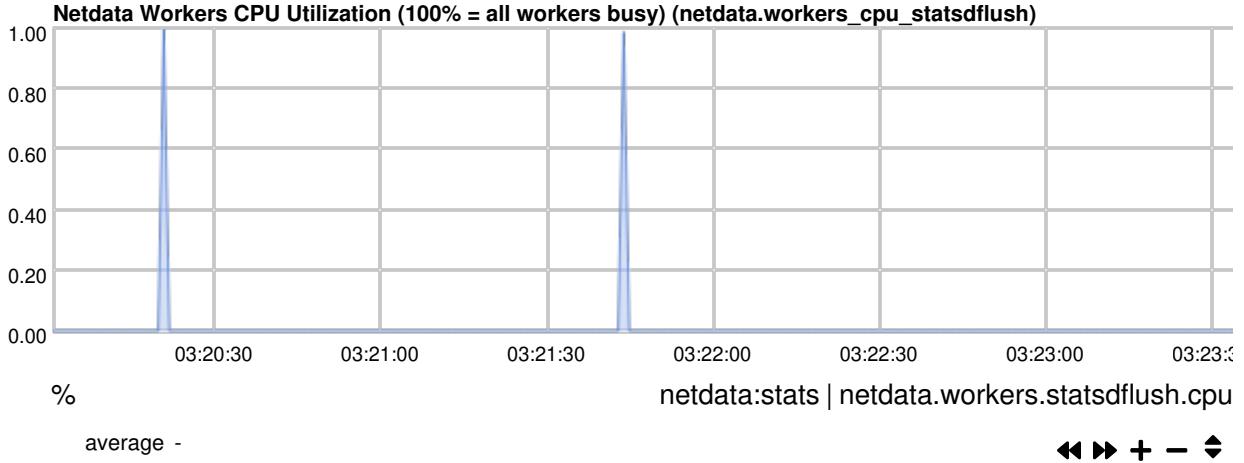
workers plugin statsd



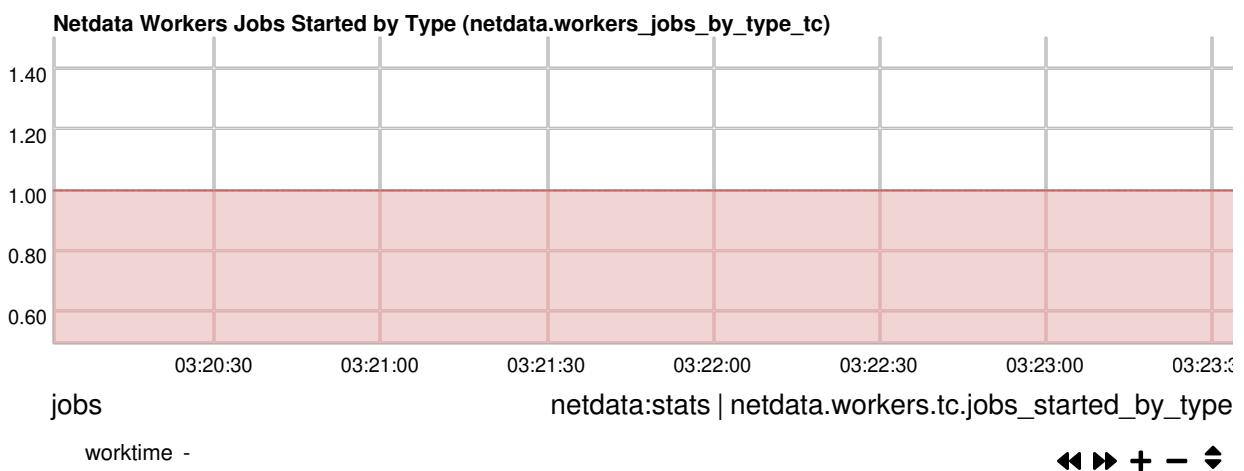
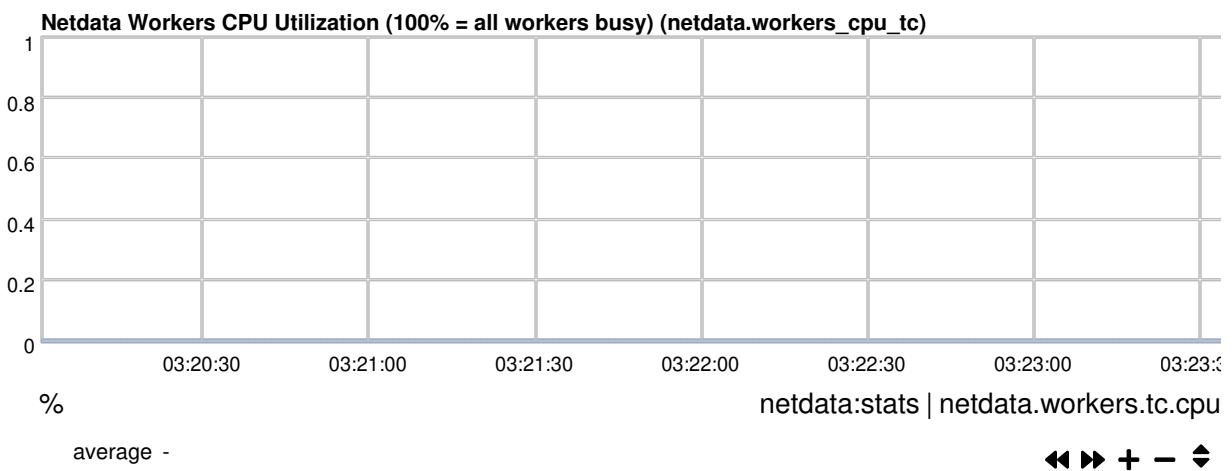
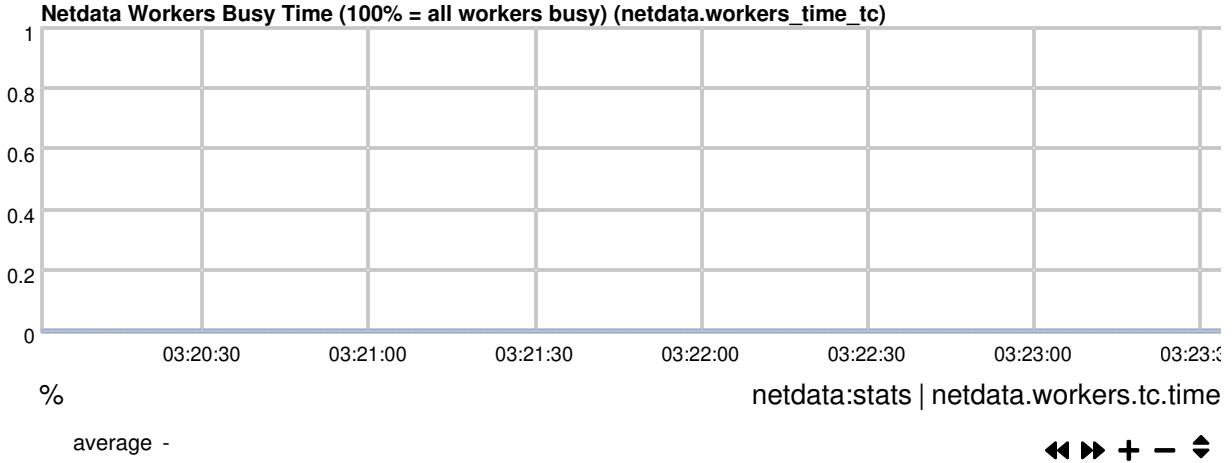


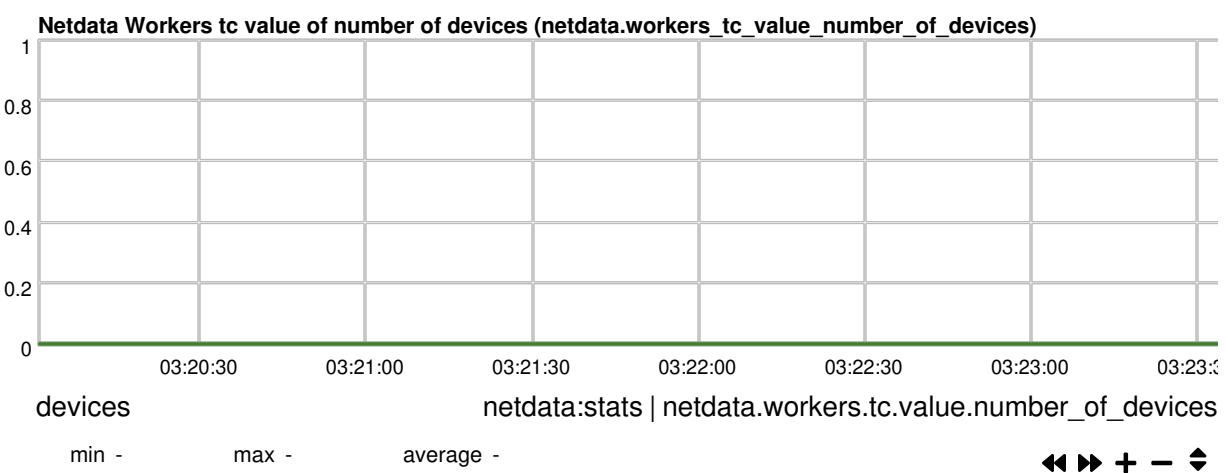
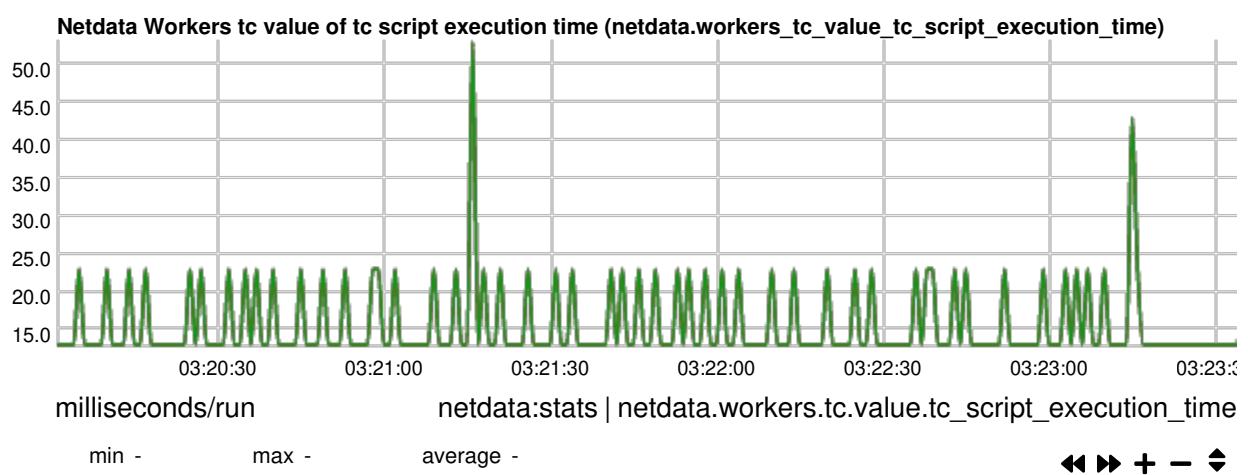
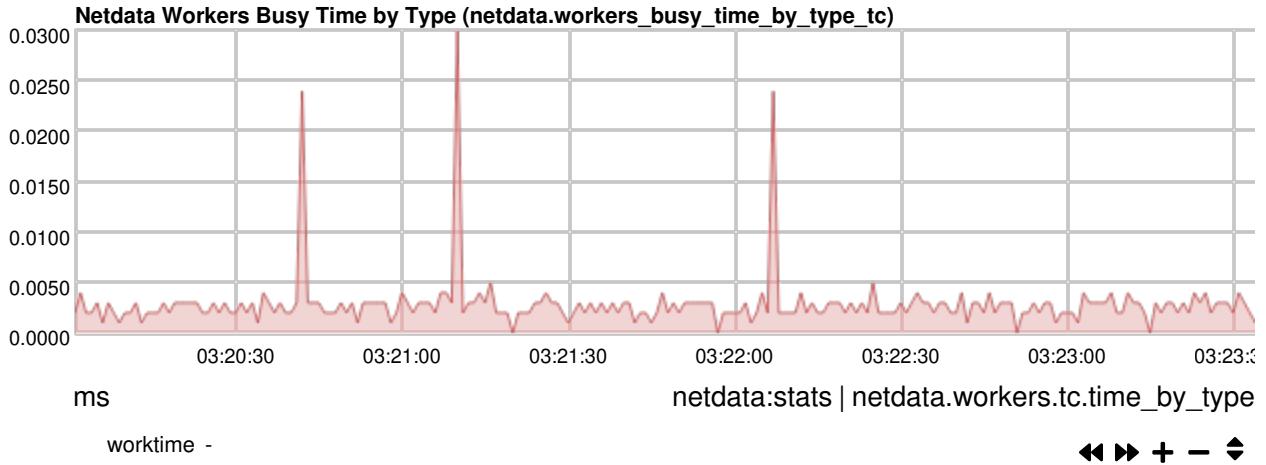
workers plugin statsd flush

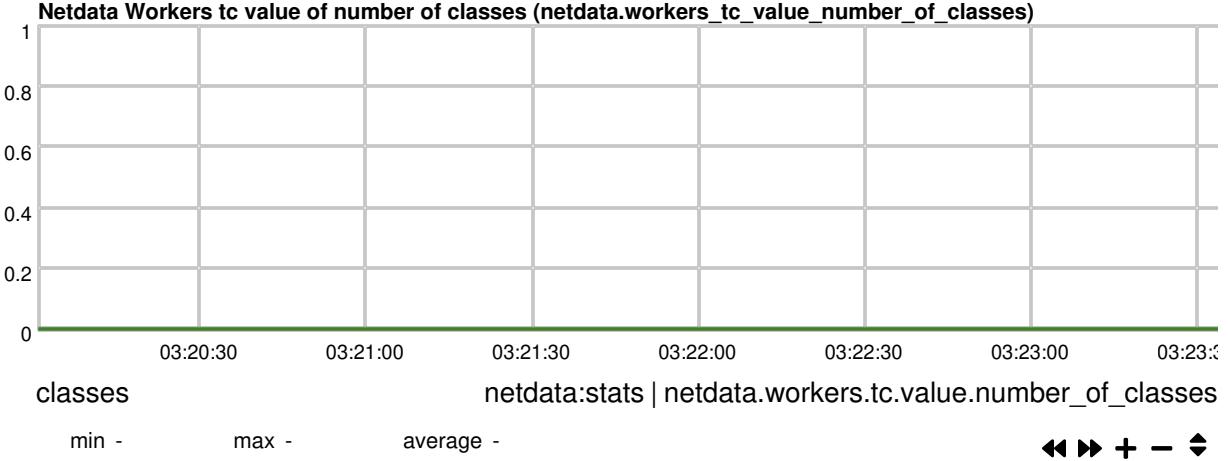




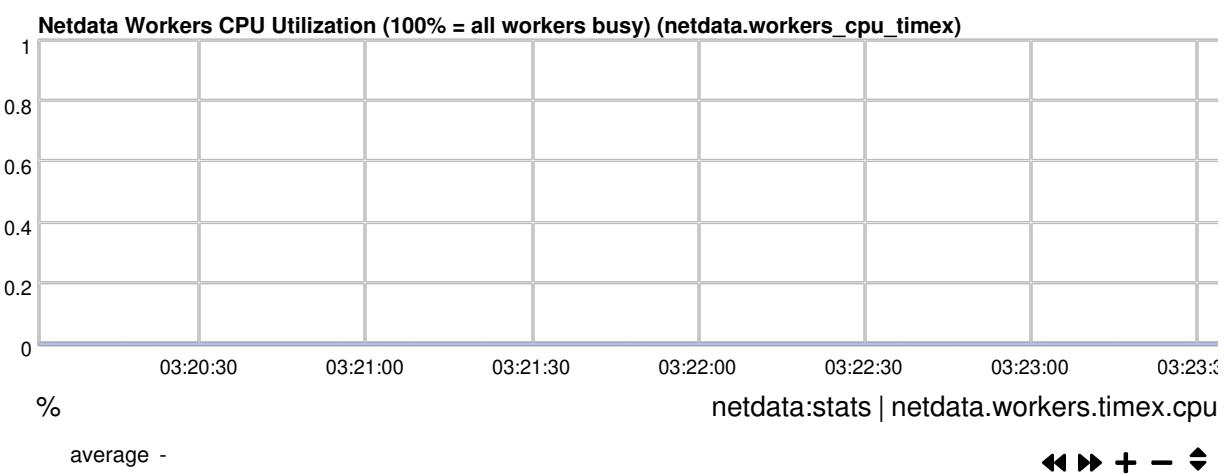
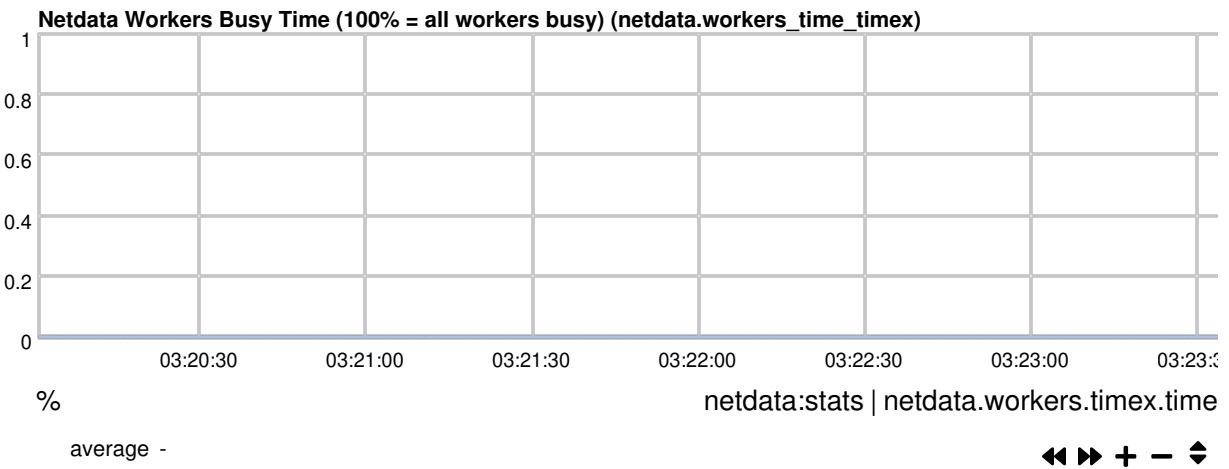
workers plugin tc

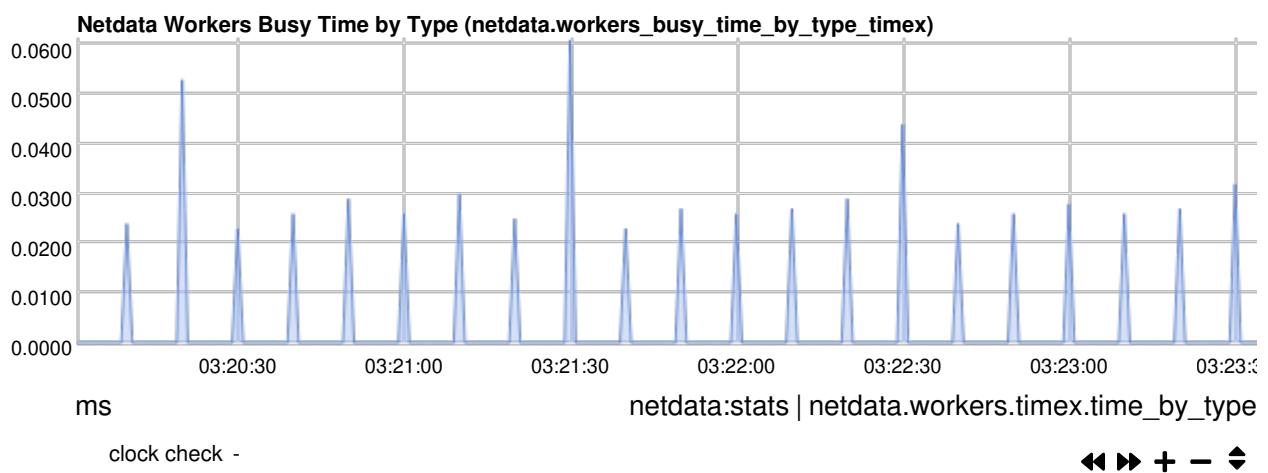
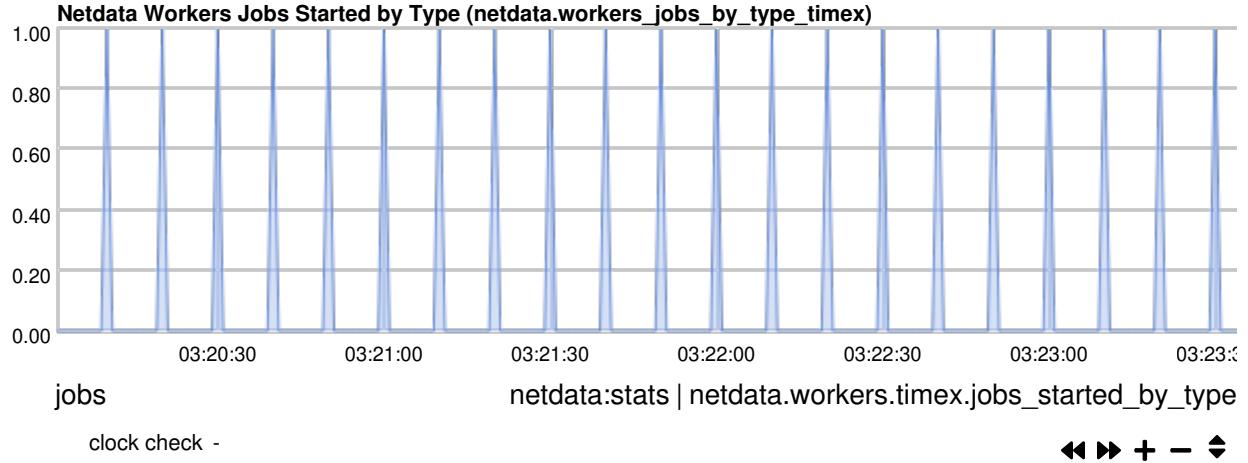




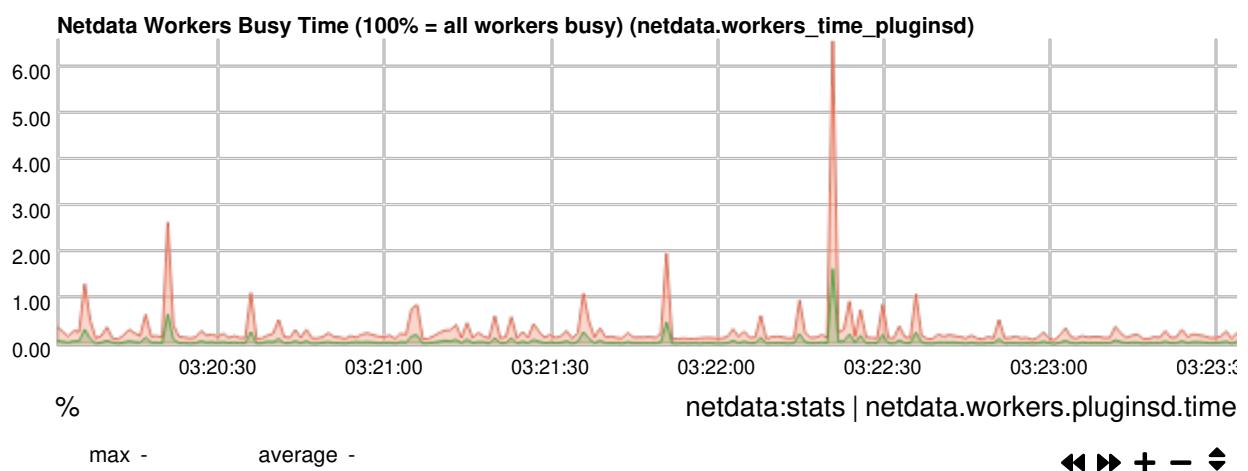


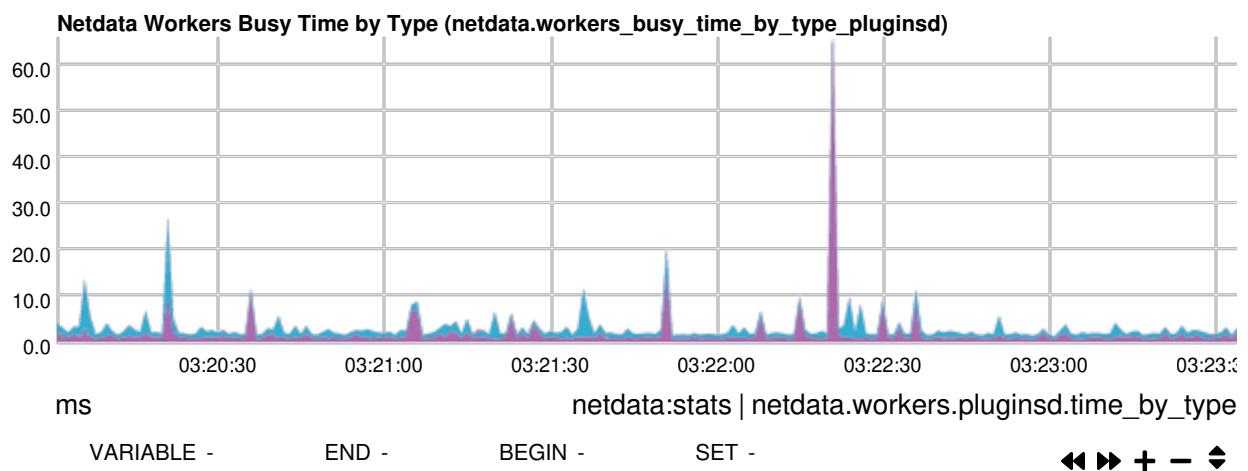
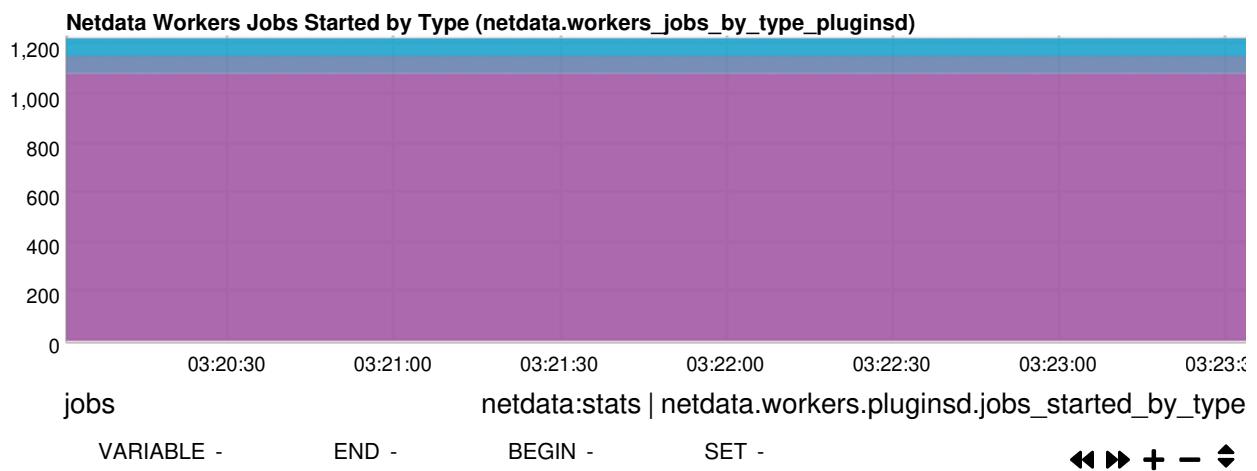
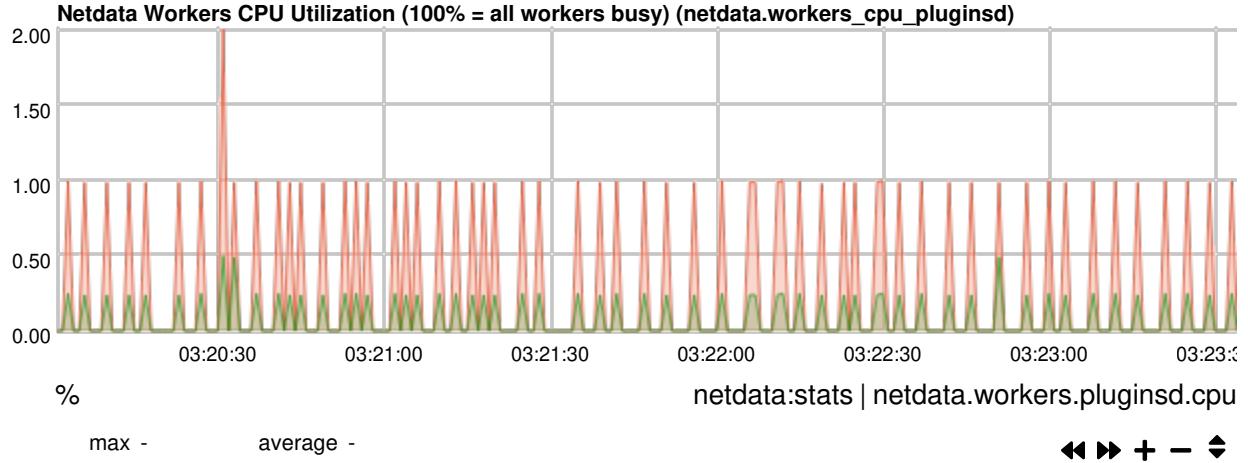
workers plugin timex

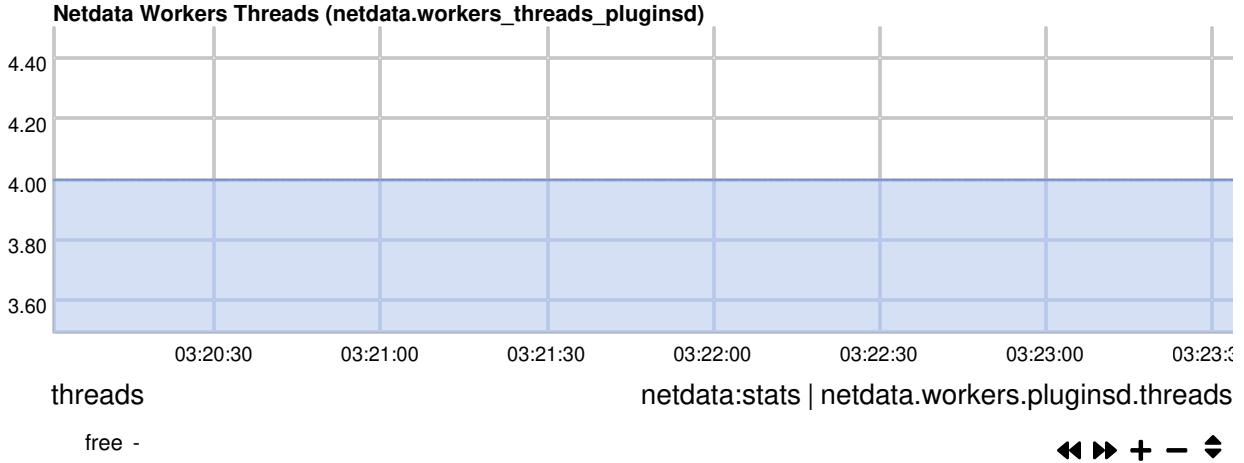




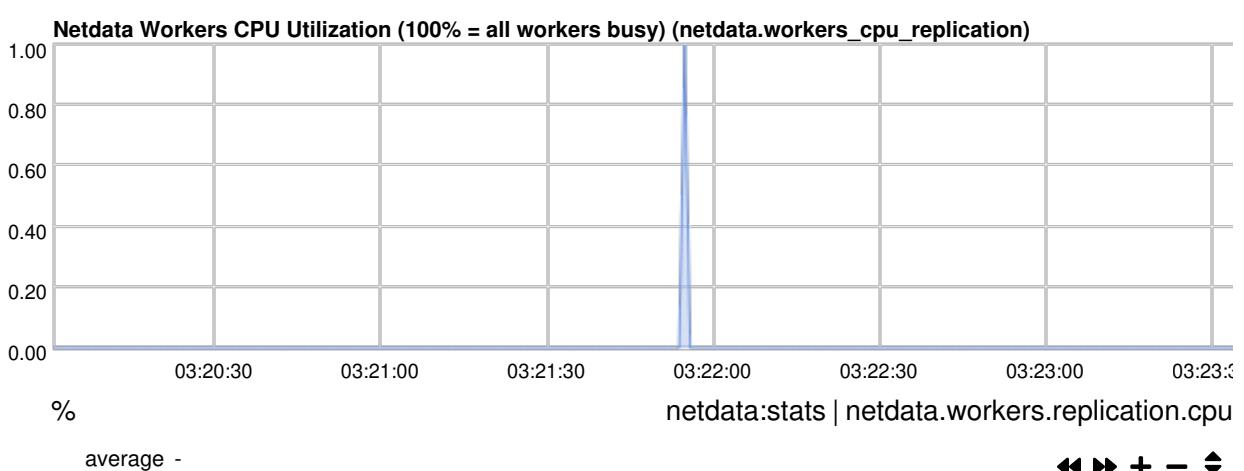
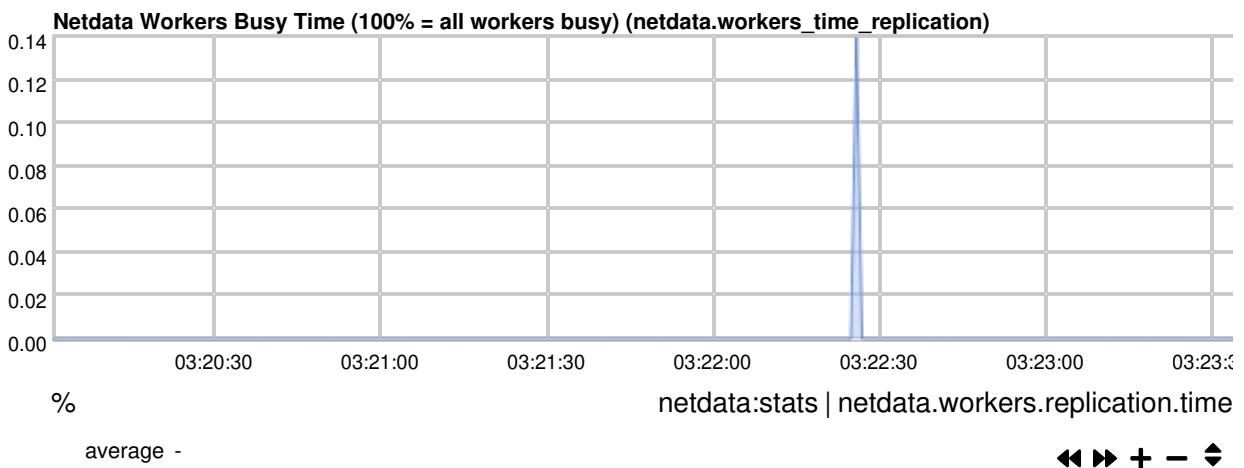
workers plugins.d

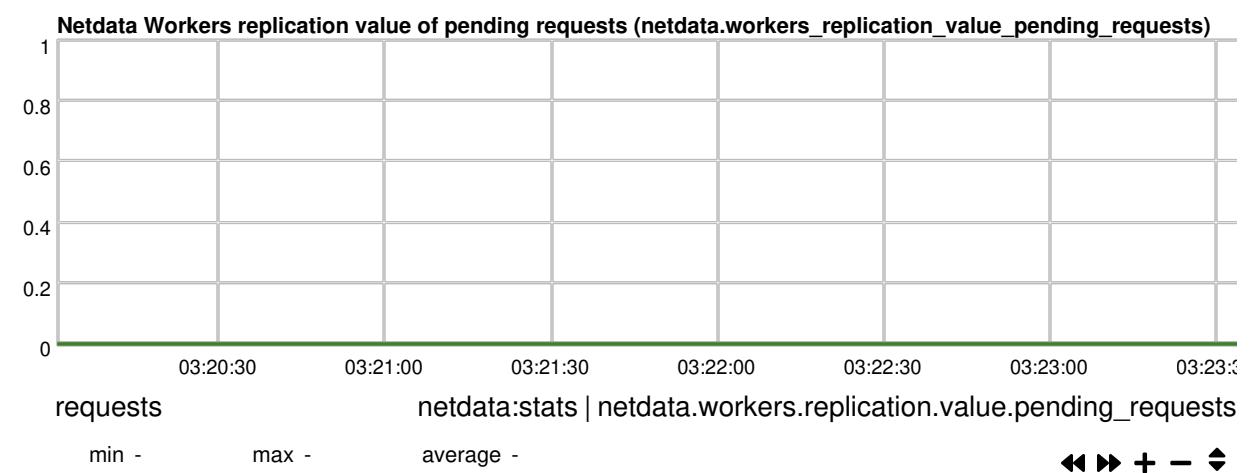
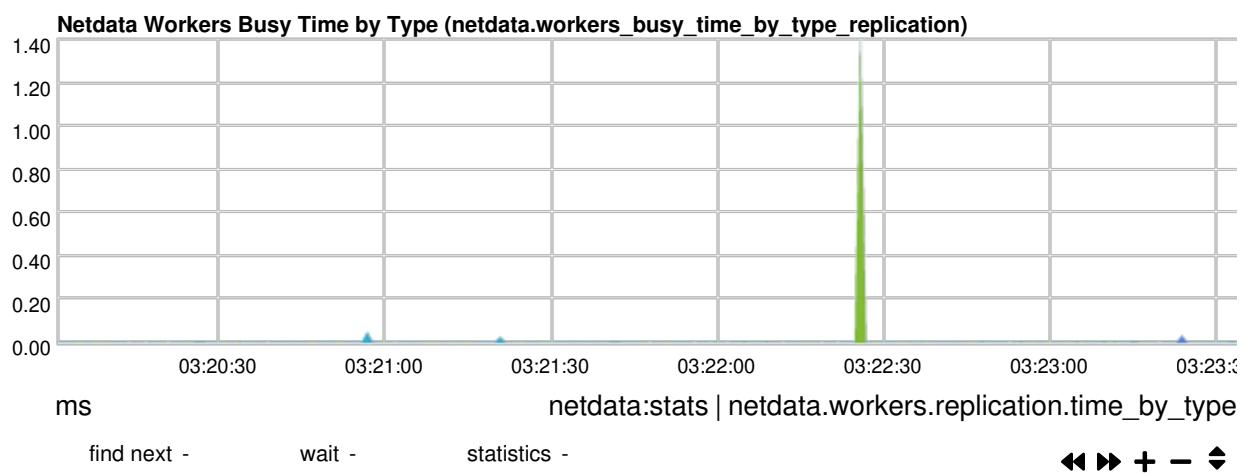
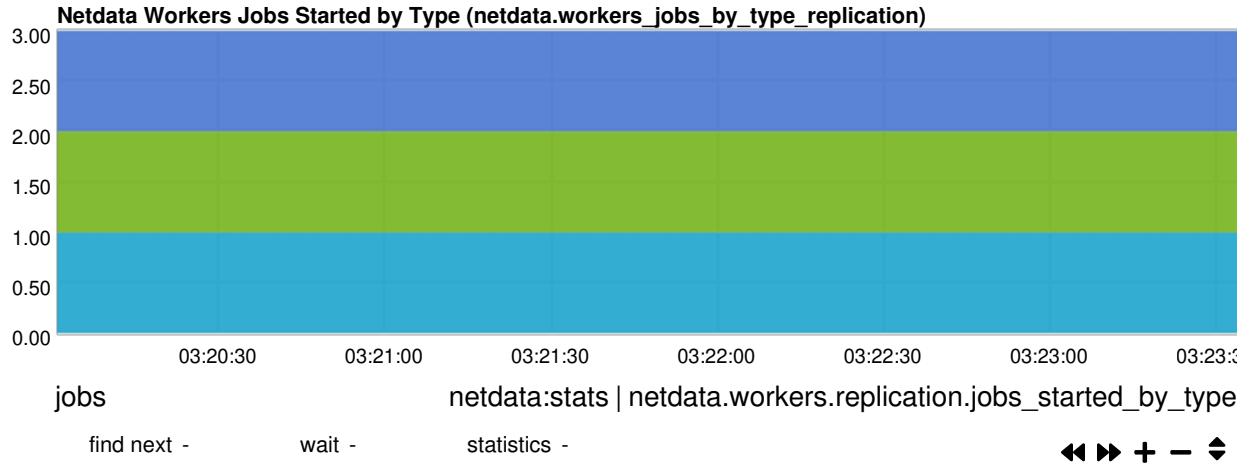


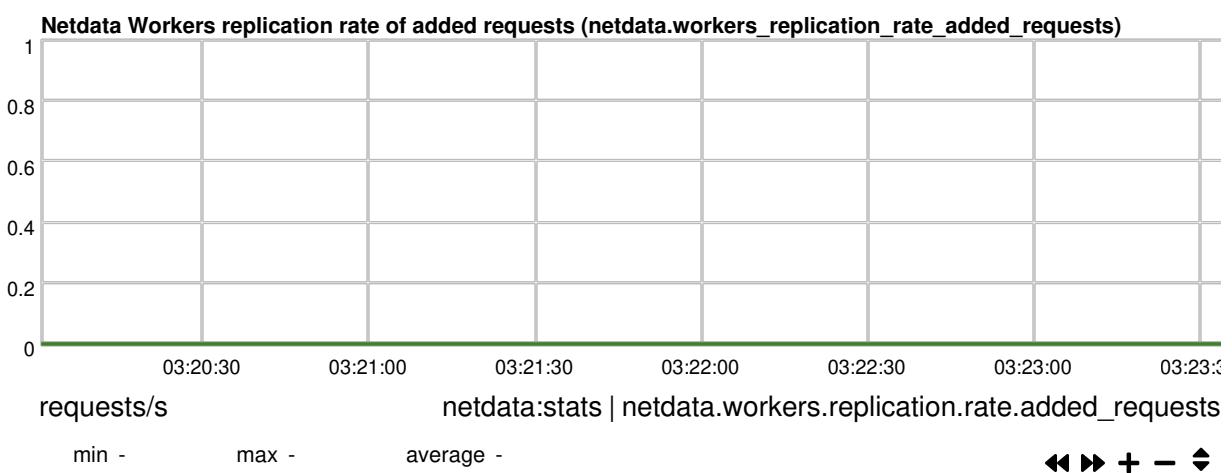
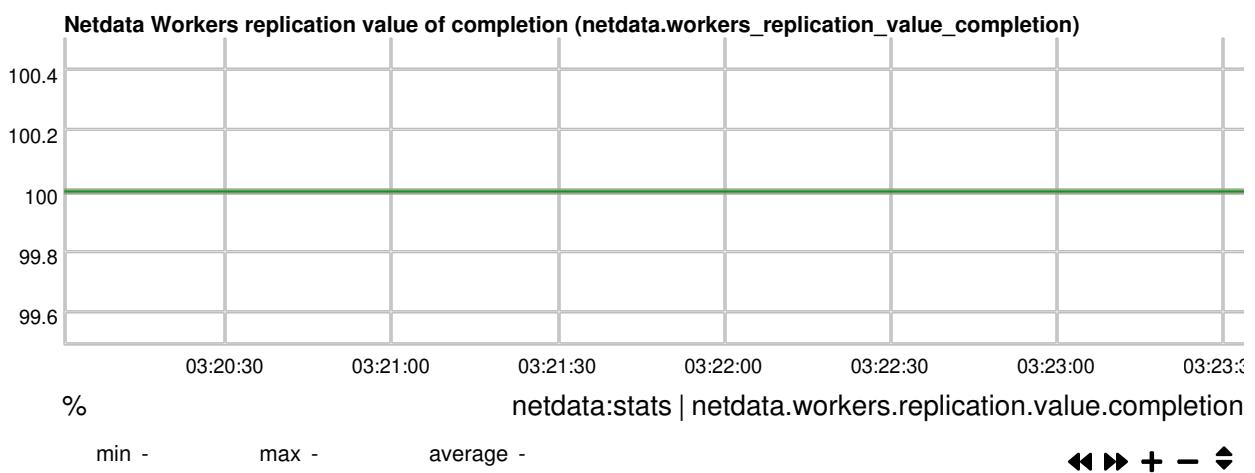
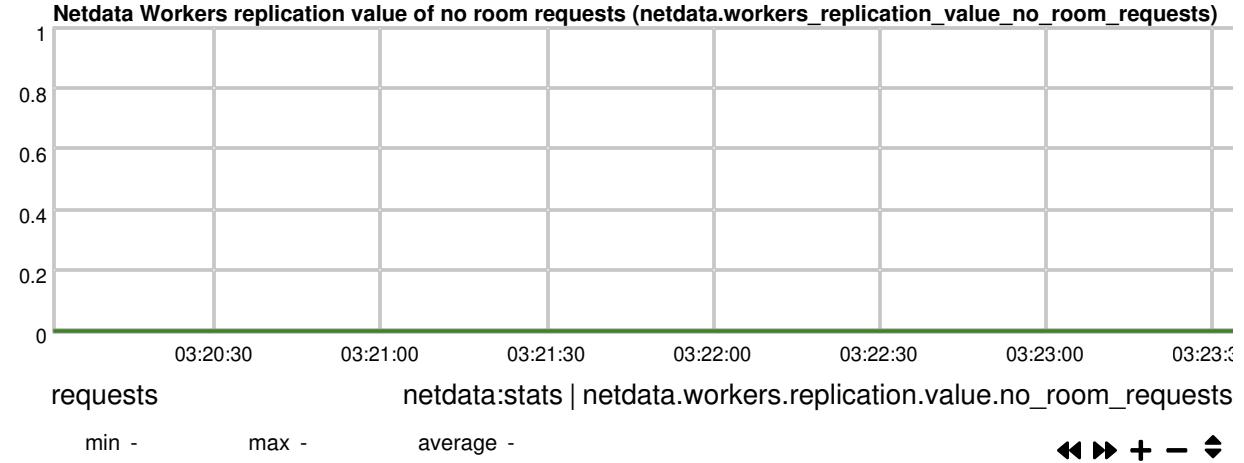


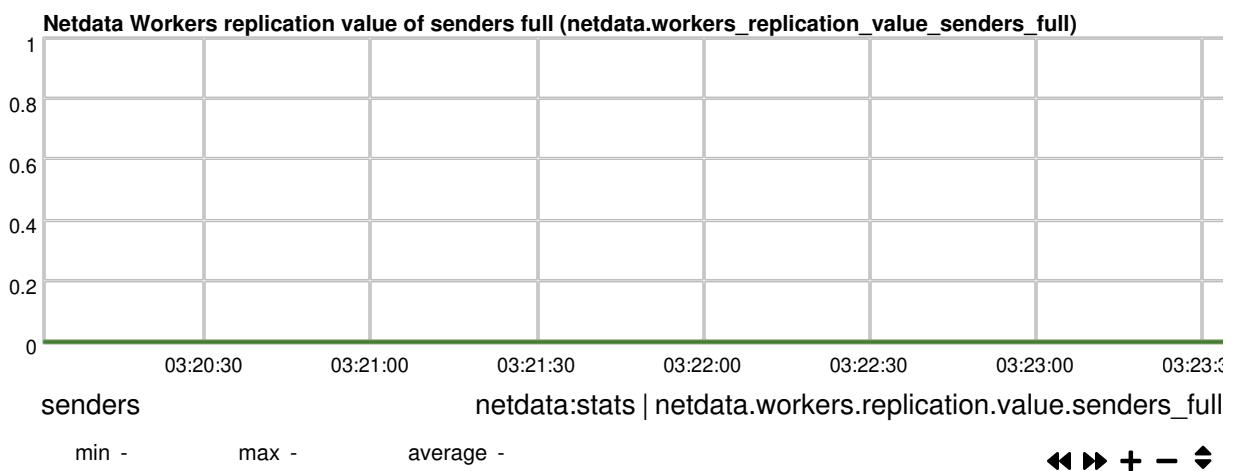
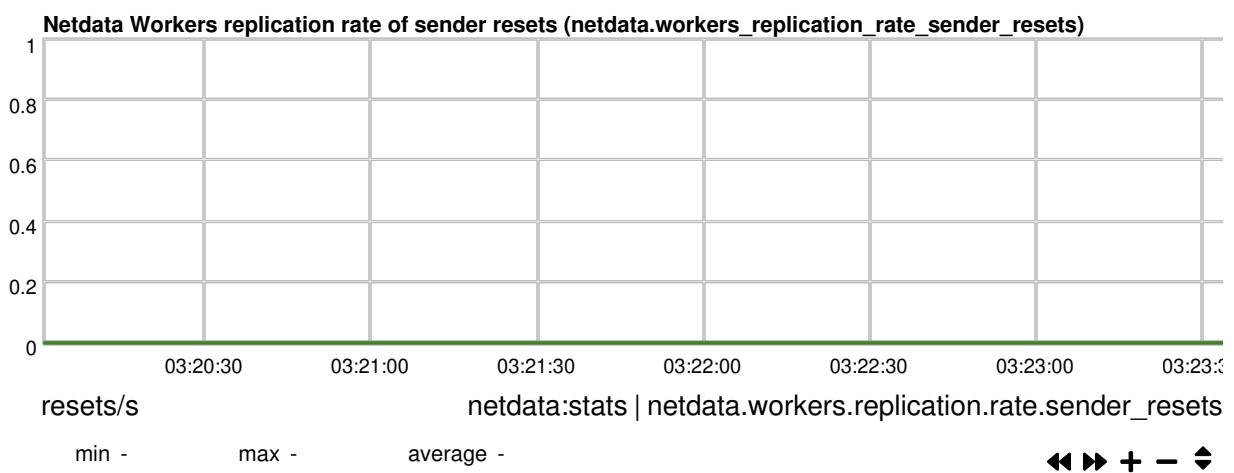
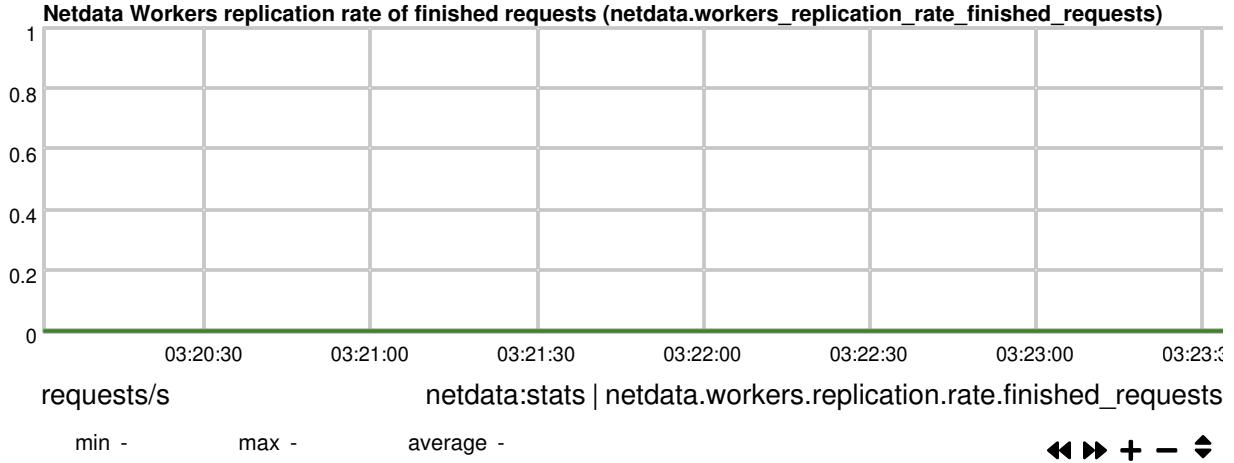


workers replication sender

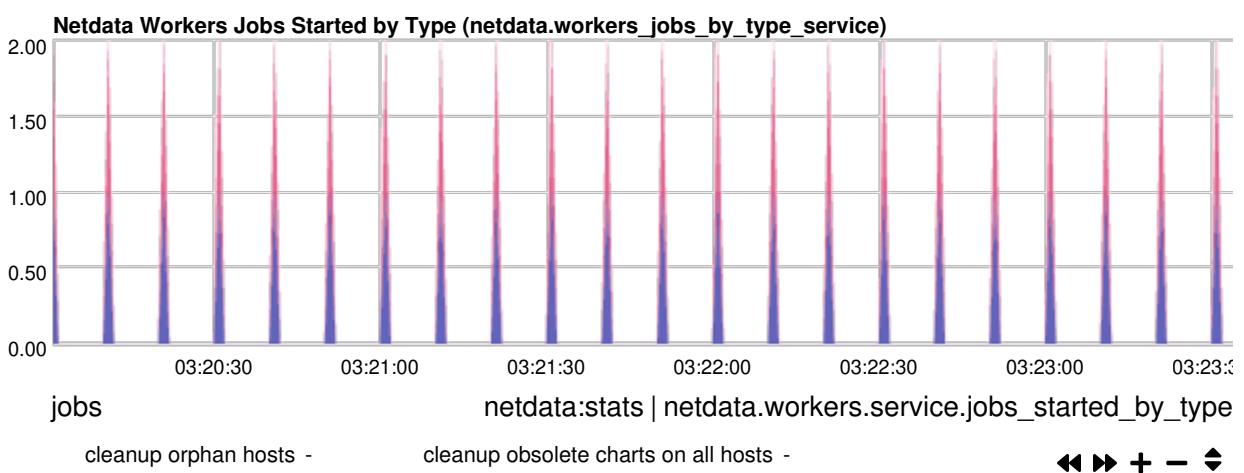
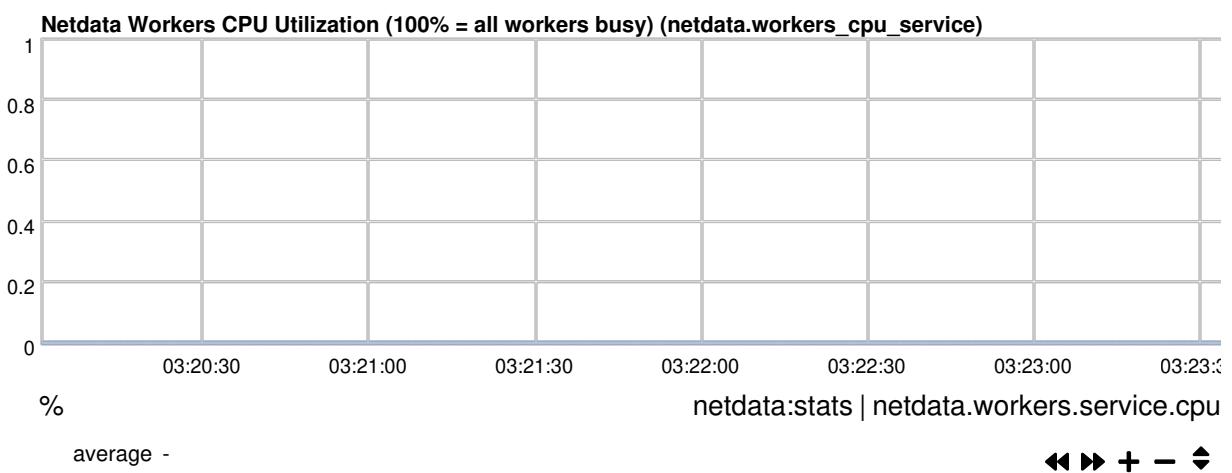
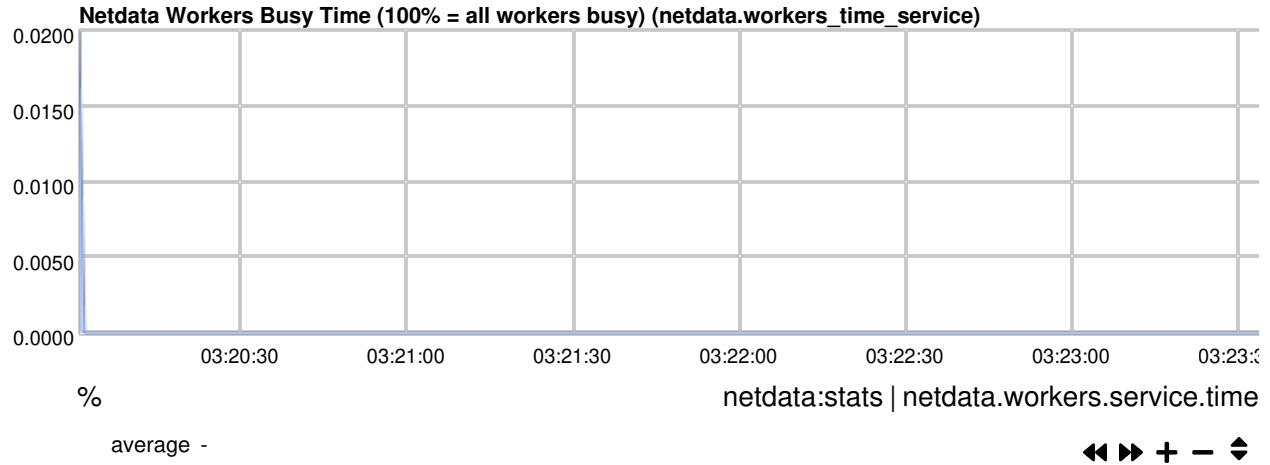


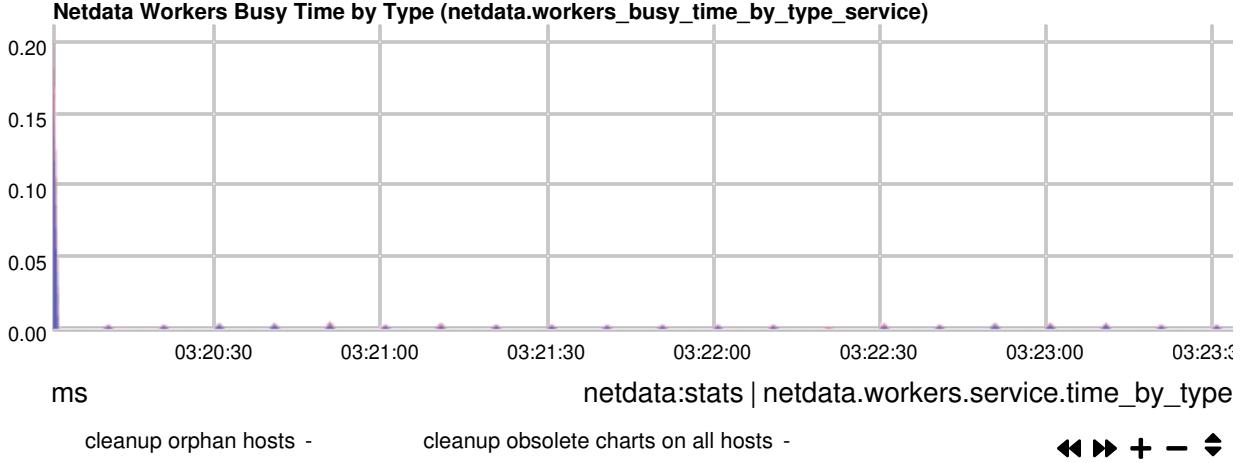




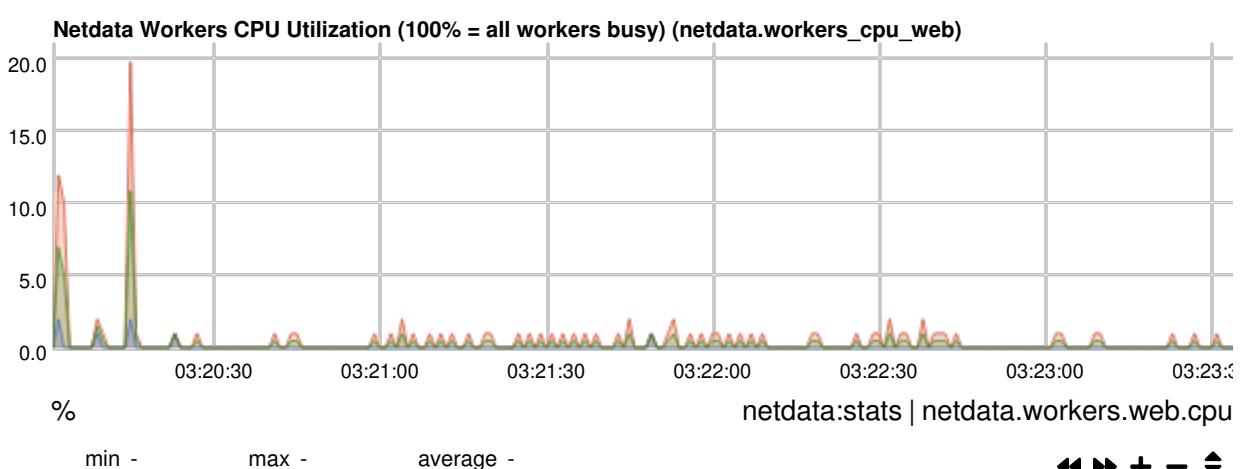
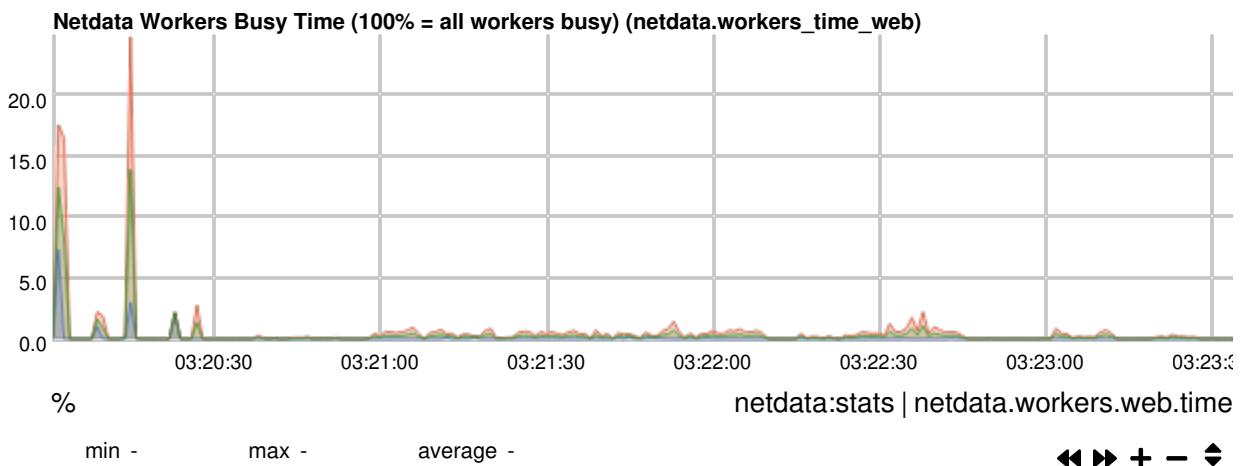


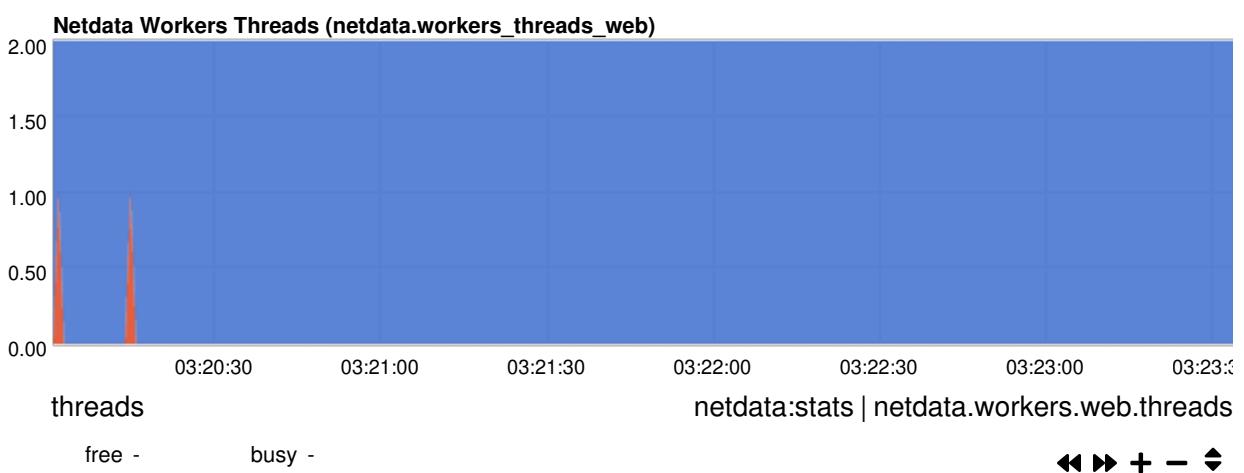
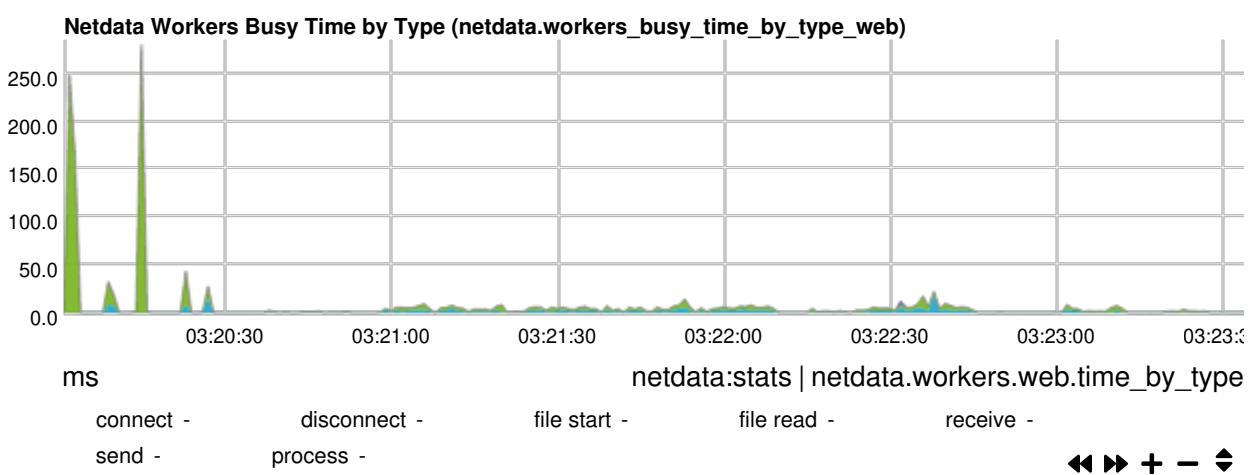
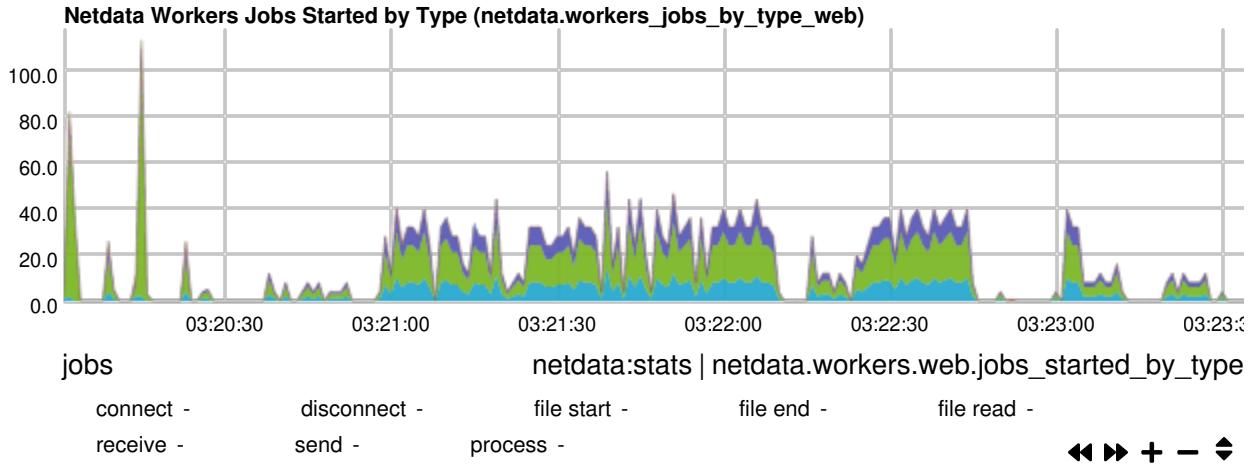
workers service





workers web server





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

© Copyright 2020, Netdata, Inc (<mailto:info@netdata.cloud>).

Terms and conditions (<https://www.netdata.cloud/terms/>)

[Privacy Policy \(https://www.netdata.cloud/privacy/\)](https://www.netdata.cloud/privacy/)

Released under GPL v3 or later (<http://www.gnu.org/licenses/gpl-3.0.en.html>). Netdata uses third party tools (<https://github.com/netdata/netdata/blob/master/REDISTRIBUTED.md>).