

marquez.int.rhx



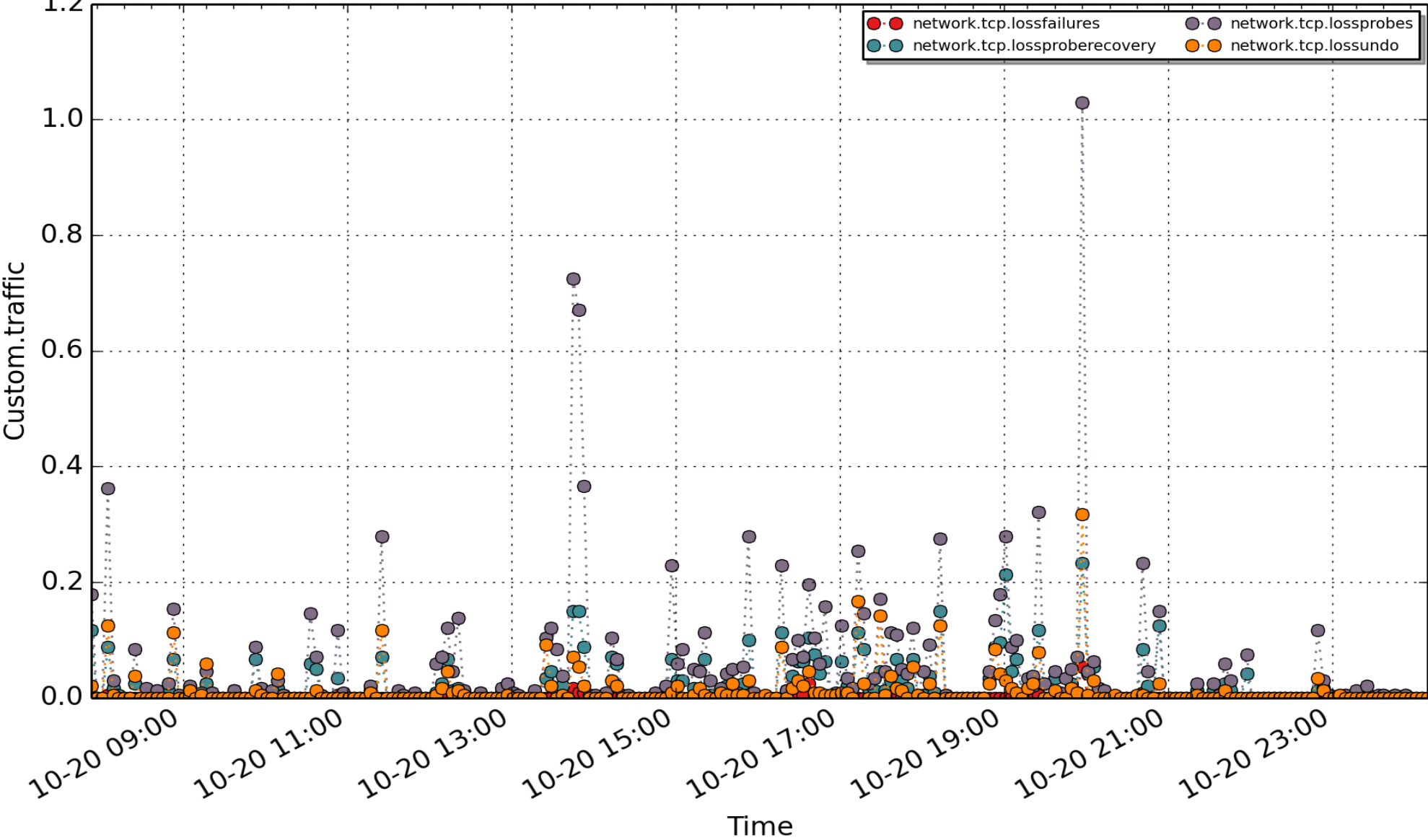
PCP Archive	20141020
Start	2014-10-20 07:45:08
End	2014-10-21 00:09:08
Interval	240.000 seconds
Created	2014-10-22 15:24:31

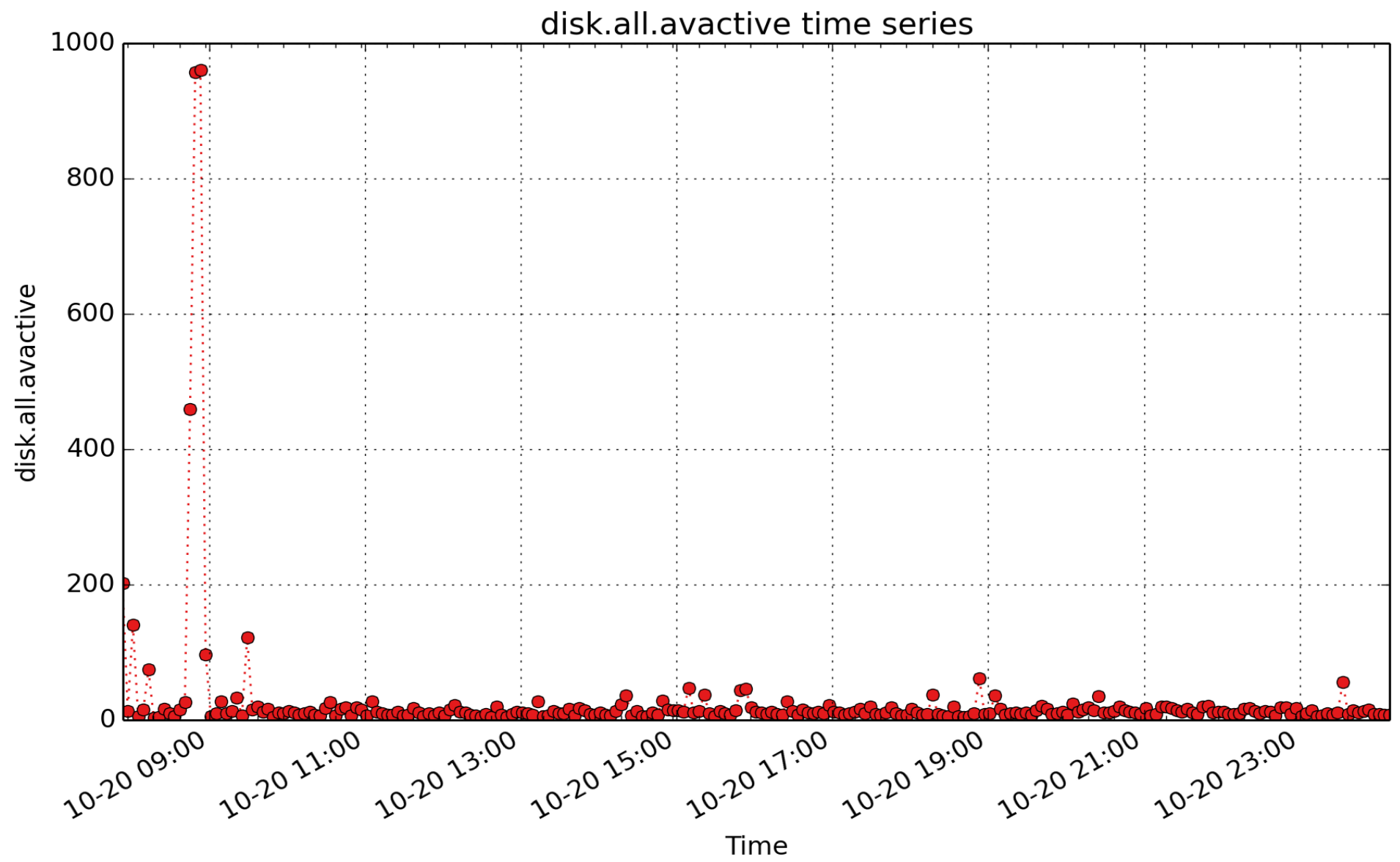
String Metrics

Metric	Timestamp	Value
filesystem.mountdir	2014-10-20 07:49:08	/home/michele
filesystem.mountdir	2014-10-20 07:49:08	/boot
filesystem.mountdir	2014-10-20 07:49:08	/
hinv.cpu.flags	2014-10-20 07:49:08	fpu vme de pse tsc msr pae mce cx8 apic sep mtrr .. ida dtherm tpr_shadow vnmi flexpriority ept vpid
hinv.cpu.model	2014-10-20 07:49:08	26
hinv.cpu.model_name	2014-10-20 07:49:08	Intel(R) Xeon(R) CPU W3550 @ 3.07GHz
hinv.cpu.stepping	2014-10-20 07:49:08	5
hinv.cpu.vendor	2014-10-20 07:49:08	GenuineIntel
hinv.machine	2014-10-20 07:49:08	linux
hinv.map.dmname	2014-10-20 07:49:08	dm-9
hinv.map.dmname	2014-10-20 07:49:08	dm-4
hinv.map.dmname	2014-10-20 07:49:08	dm-2
hinv.map.dmname	2014-10-20 07:49:08	dm-8
hinv.map.dmname	2014-10-20 07:49:08	dm-1
hinv.map.dmname	2014-10-20 07:49:08	dm-3
hinv.map.dmname	2014-10-20 07:49:08	dm-7
hinv.map.dmname	2014-10-20 07:49:08	dm-0
hinv.map.dmname	2014-10-20 07:49:08	dm-5
hinv.map.dmname	2014-10-20 07:49:08	dm-6
hinv.map.lvname	2014-10-20 07:49:08	backup-raw
hinv.map.lvname	2014-10-20 07:49:08	home_crypted
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-6c898455--d1f0--461a--b210--73d605357bb3
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-96bc870e--d7b0--4cb1--8435--cc6d09572301
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-outbox
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-master
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-ids
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-inbox
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-metadata

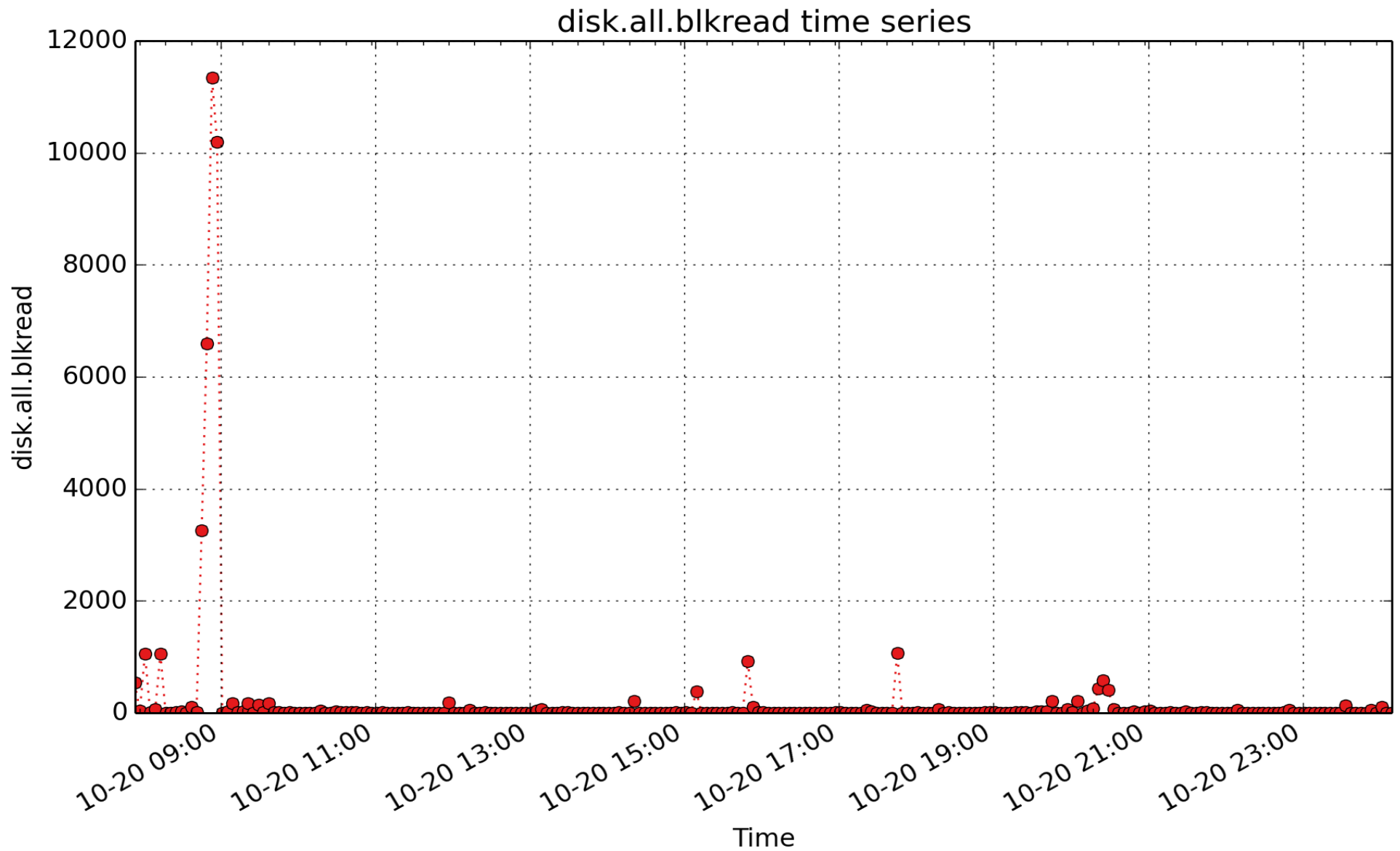
hinv.map.lvname	2014-10-20 07:49:08	127947d0--a4e1--4cd6--ad82--04c9e2ea48dd-leases
hinv.map.scsi	2014-10-20 07:49:08	sdc
hinv.map.scsi	2014-10-20 07:49:08	sdb
hinv.map.scsi	2014-10-20 07:49:08	sdd
hinv.map.scsi	2014-10-20 07:49:08	scd0
hinv.map.scsi	2014-10-20 07:49:08	sda
kernel.uname.machine	2014-10-20 07:49:08	x86_64
pmcd.build	2014-10-20 07:49:08	1
pmcd.pmlogger.archive	2014-10-20 07:45:08	/var/log/pcp/pmlogger/marquez.int.rhx/20141020.07.45
pmcd.pmlogger.host	2014-10-20 07:45:08	marquez.int.rhx
pmcd.pmlogger.pmcd_host	2014-10-20 07:49:08	marquez.int.rhx
pmcd.version	2014-10-20 07:49:08	3.9.10

Custom.traffic time series

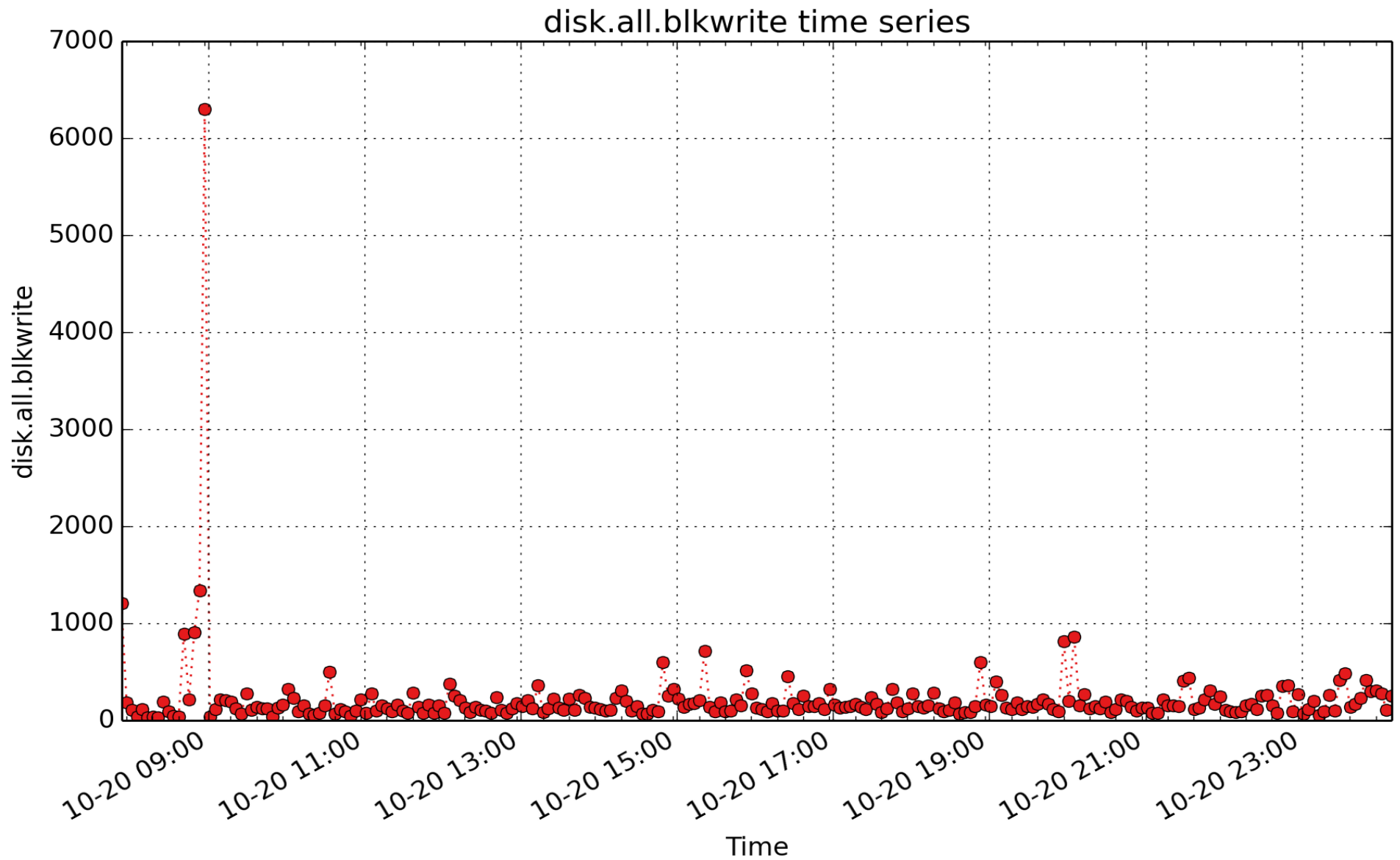




disk.all.avactive: When converted to a rate, this metric represents the average utilization of all disks during the sampling interval. A value of 0.25 (or 25%) means that on average every disk was active (i.e. busy) one quarter of the time. (millisec - U64) - *rate converted*



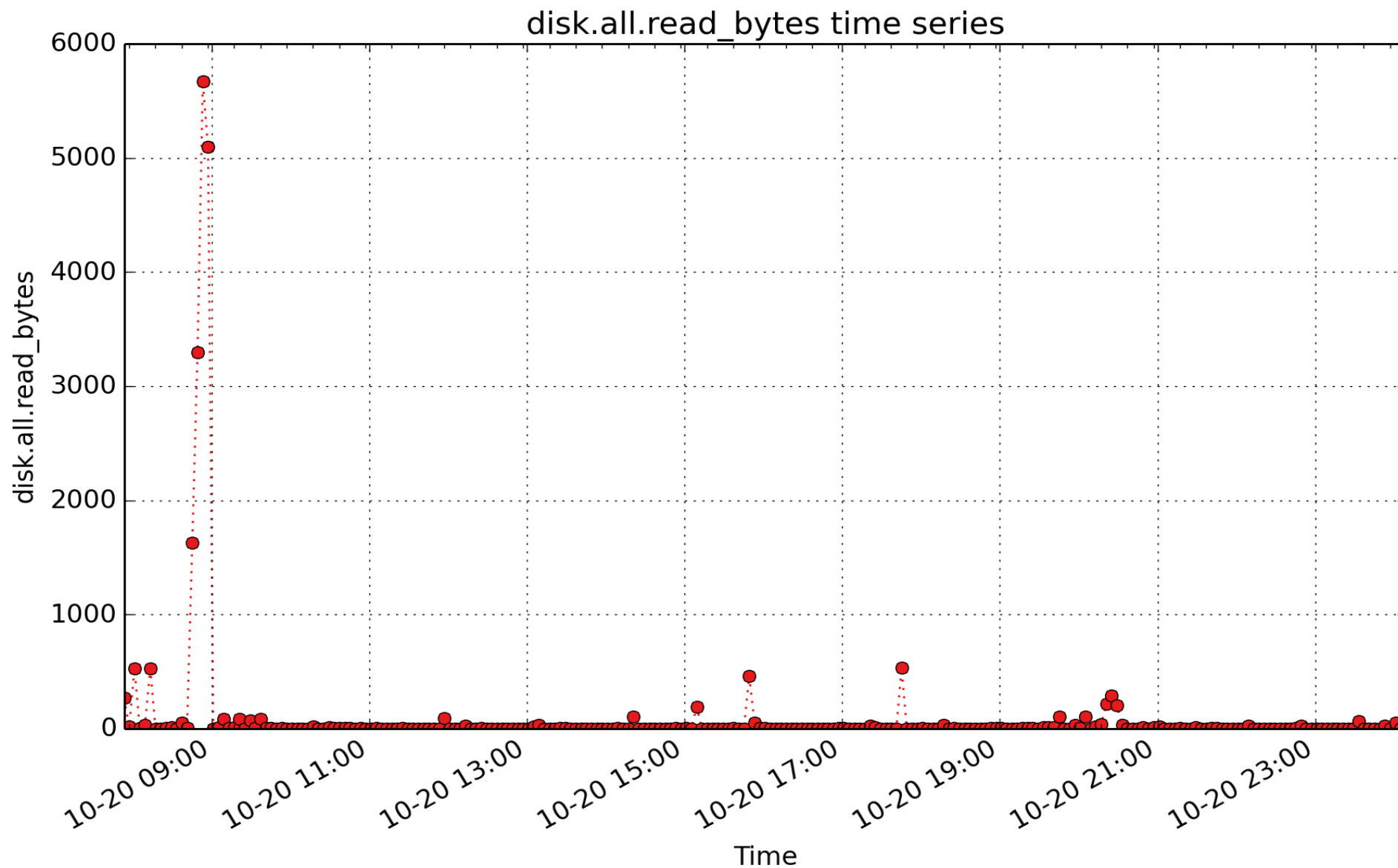
disk.all.blkread: Cumulative number of disk block read operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



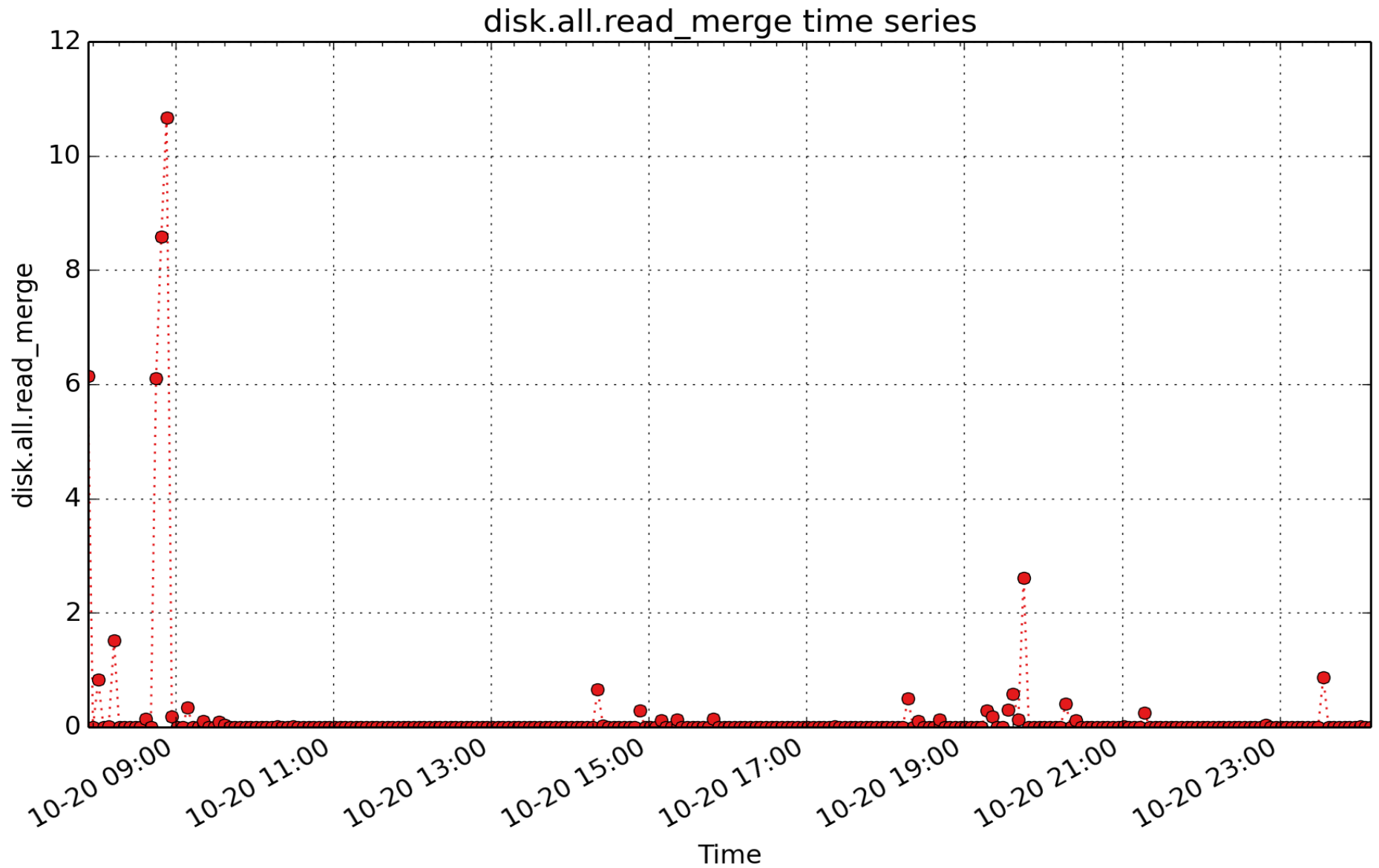
disk.all.blkwrite: Cumulative number of disk block write operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



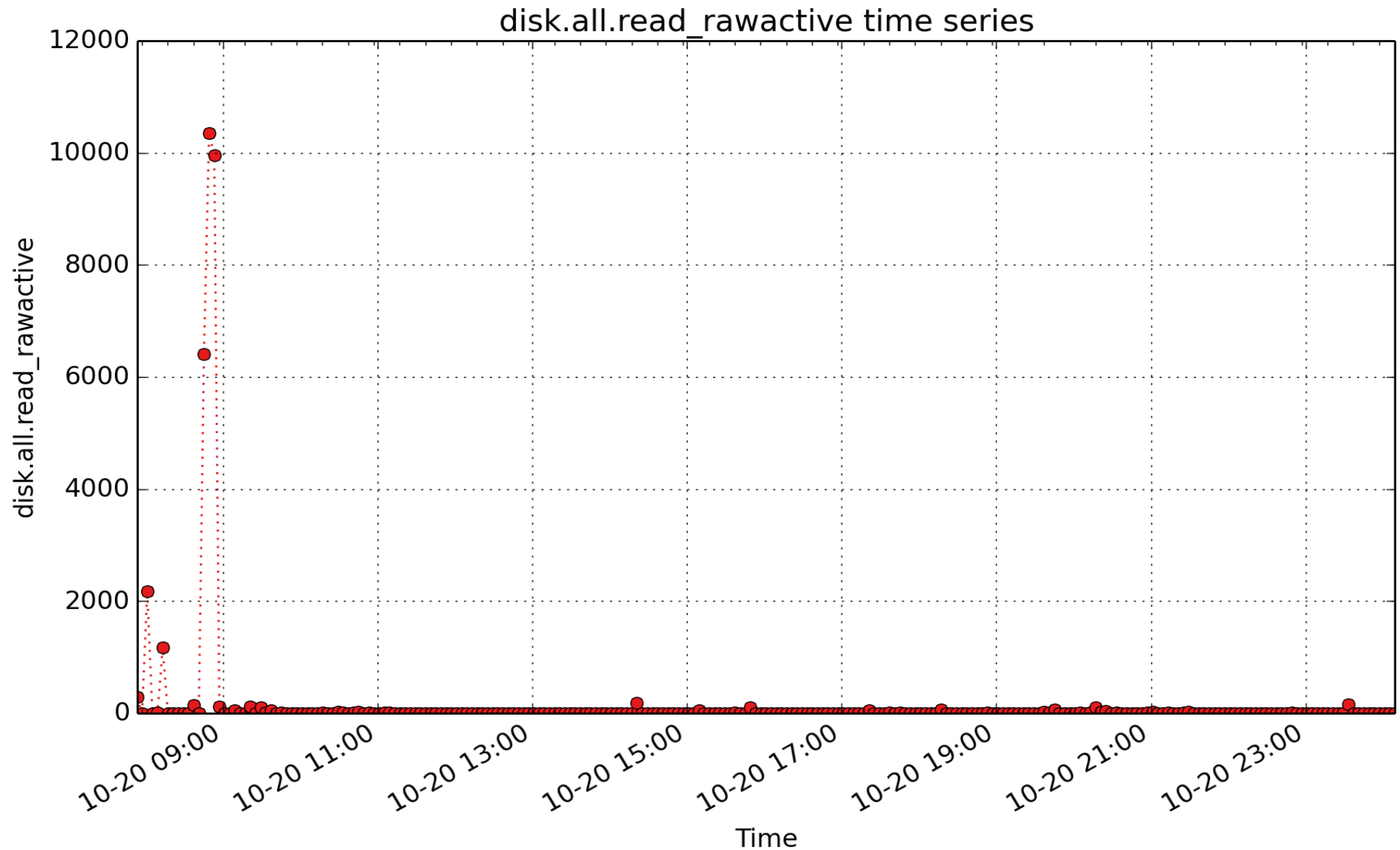
disk.all.read: Cumulative number of disk read operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



disk.all.read_bytes: count of bytes read for all disk devices (Kbyte - U32) - *rate converted*

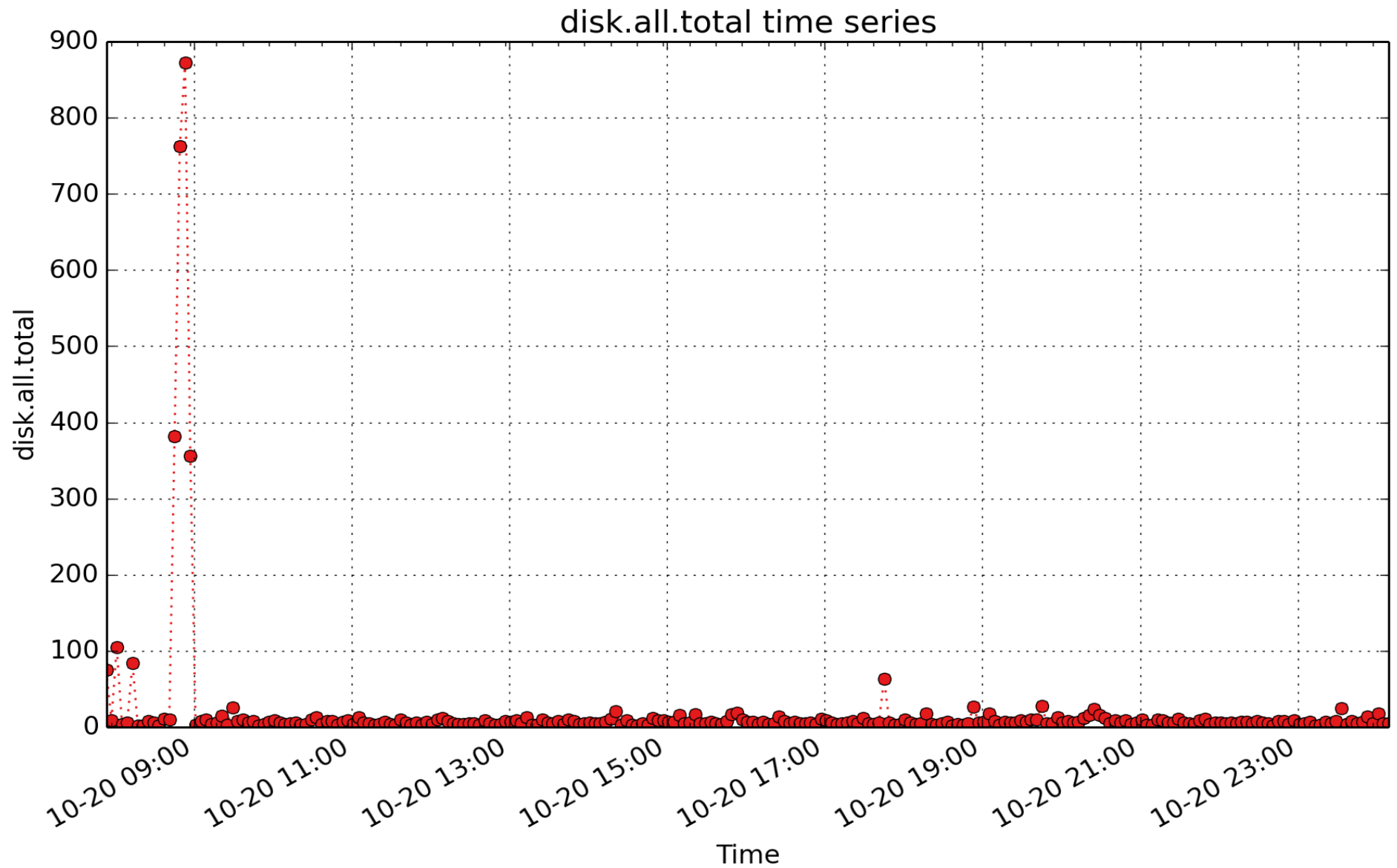


disk.all.read_merge: Total count of read requests that were merged with an already queued read request.
(count - U64) - *rate converted*

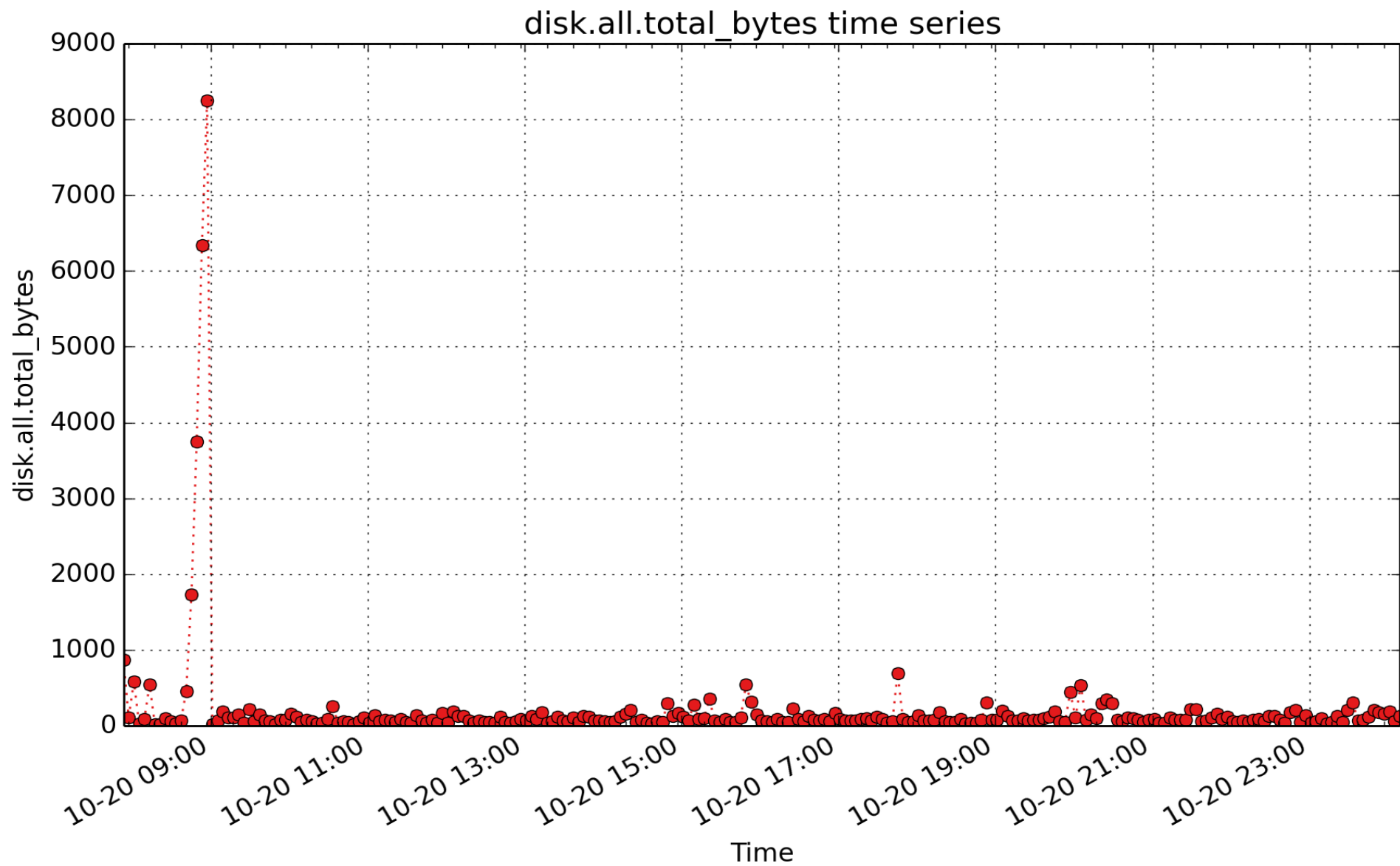


disk.all.read_rawactive: When converted to a rate, this metric represents the raw utilization of all disks during the sampling interval due to read requests. The accounting for this metric is only done on I/O completion and can thus result in more than a second's worth of IO being accounted for within any one second, leading to >100% utilisation. It is suitable mainly for use in calculations with other metrics, e.g.

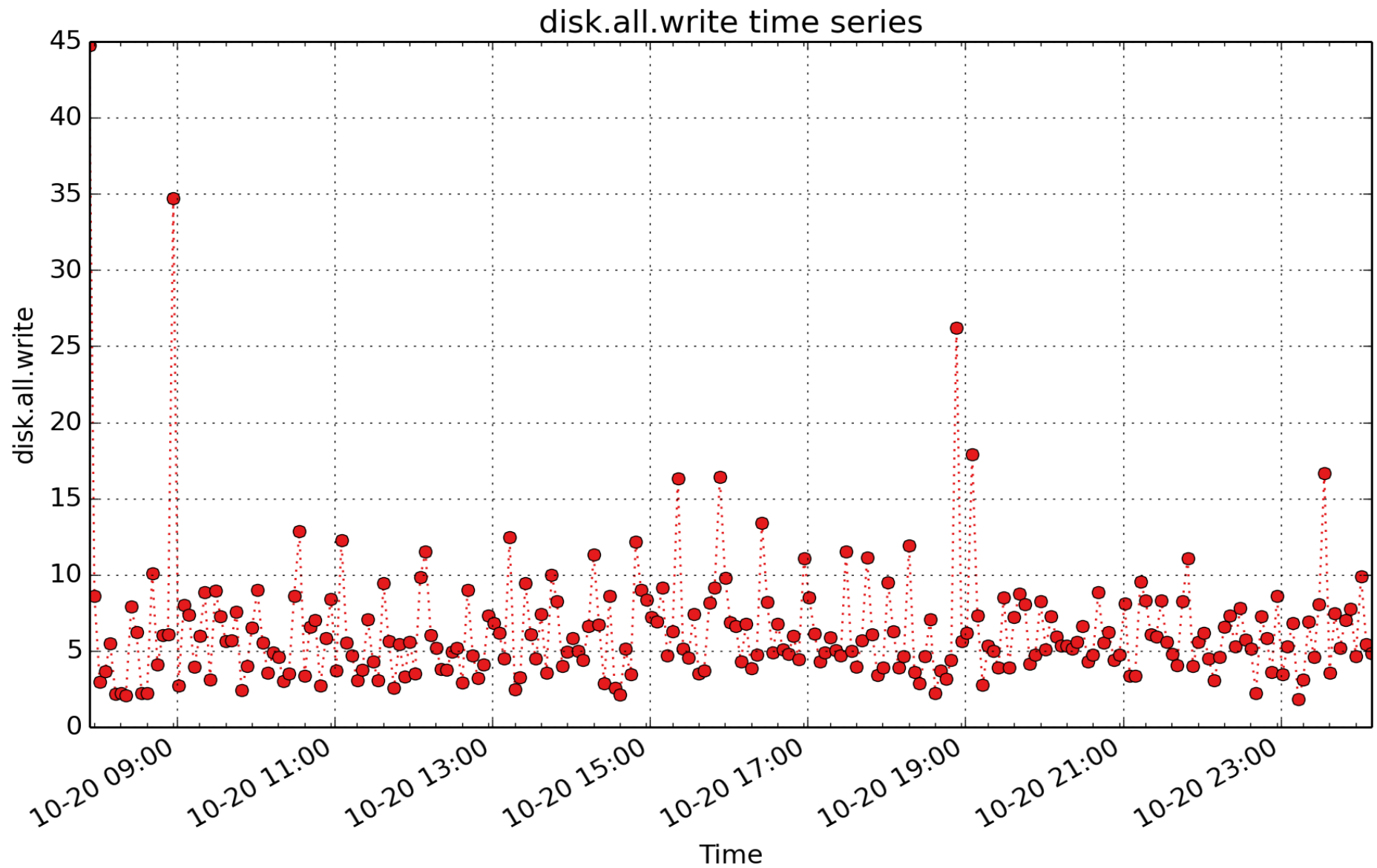
mirroring the results from existing performance tools: $\text{iostat.all.r_await} = \frac{\text{delta(disk.all.read_rawactive)}}{\text{delta(disk.all.read)}} \text{ (millisec - U64) - } \textit{rate converted}$



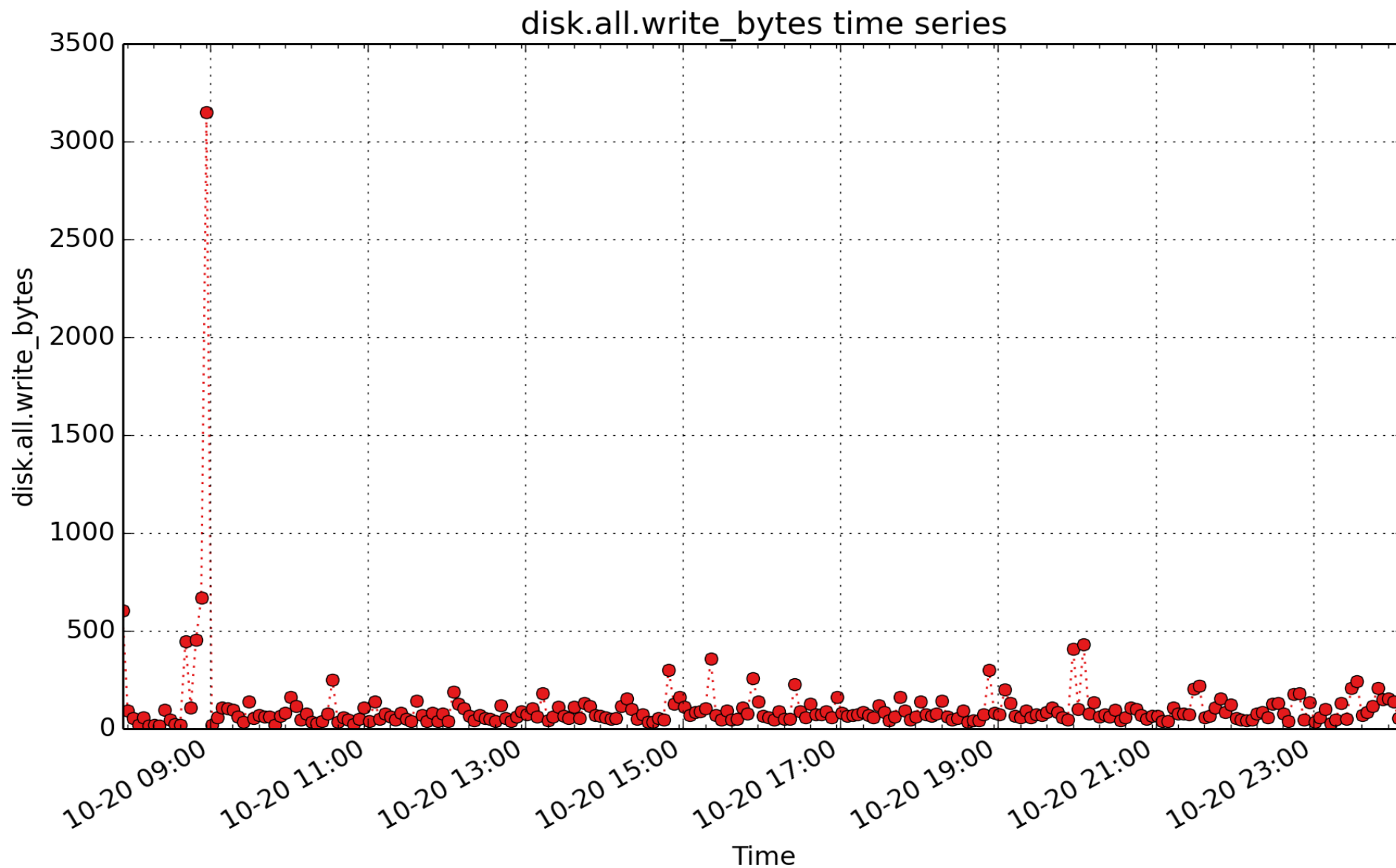
disk.all.total: Cumulative number of disk read and write operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



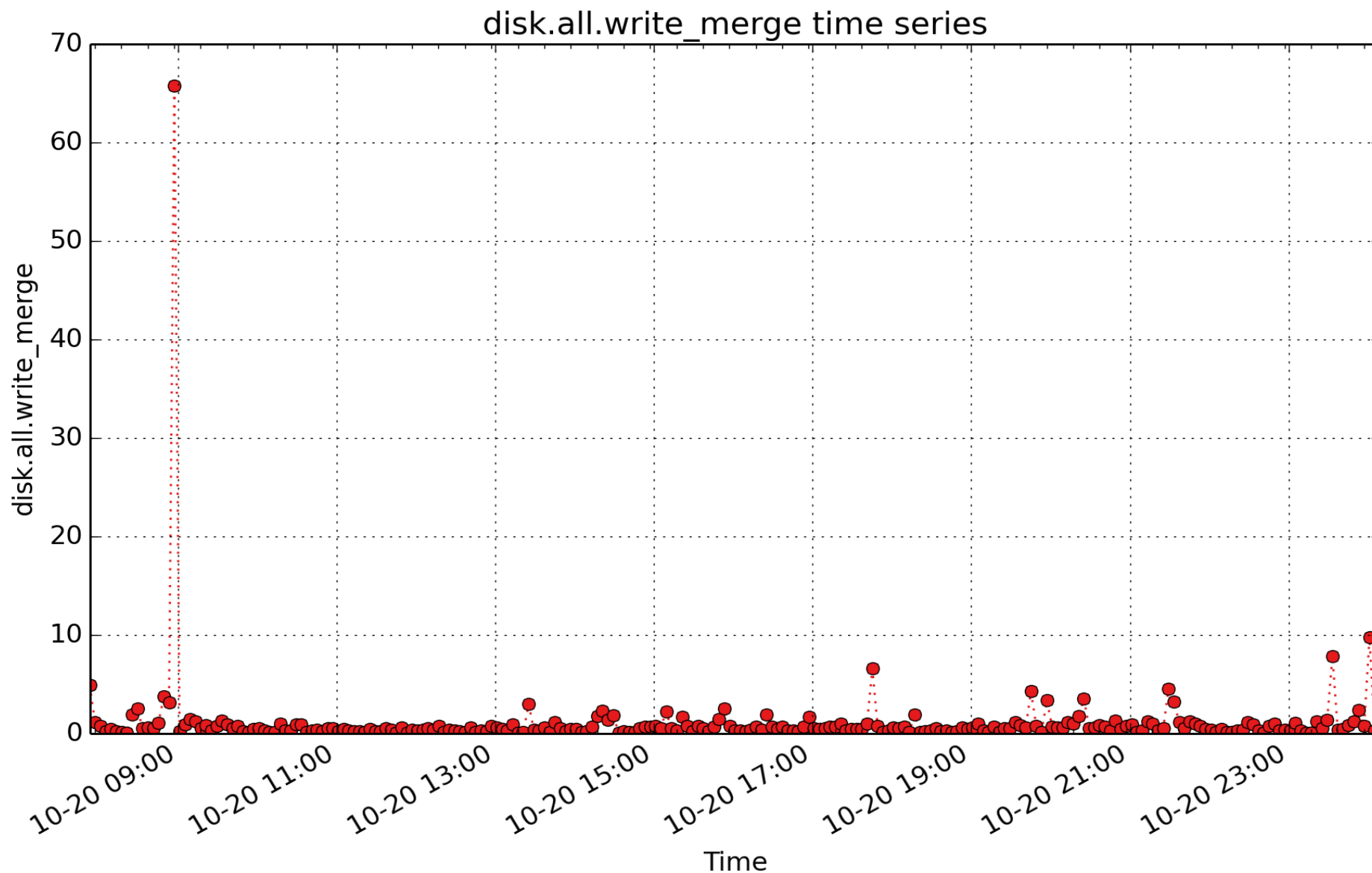
disk.all.total_bytes: total count of bytes read and written for all disk devices (Kbyte - U32) - *rate converted*



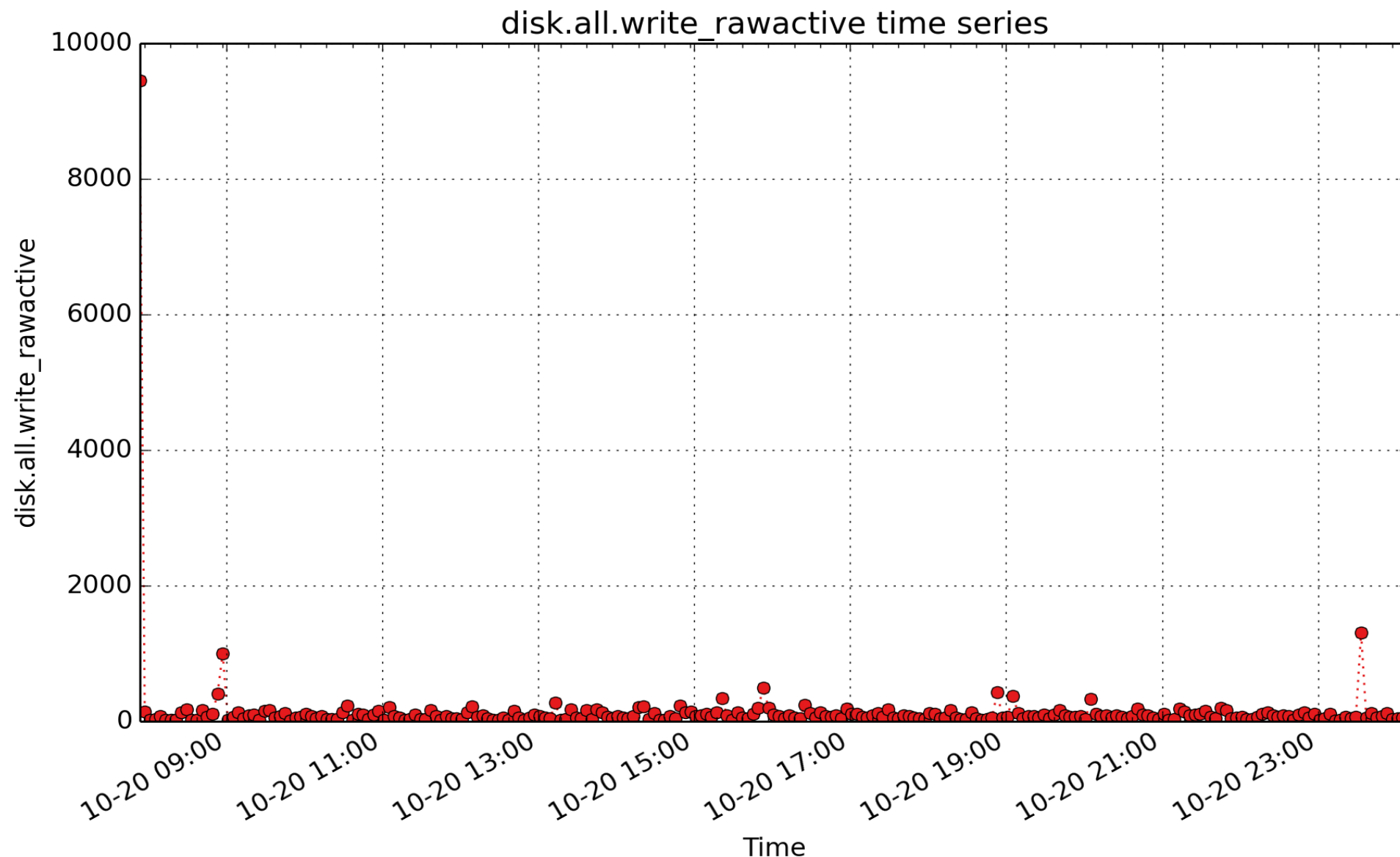
disk.all.write: Cumulative number of disk read operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



disk.all.write_bytes: count of bytes written for all disk devices (Kbyte - U32) - *rate converted*

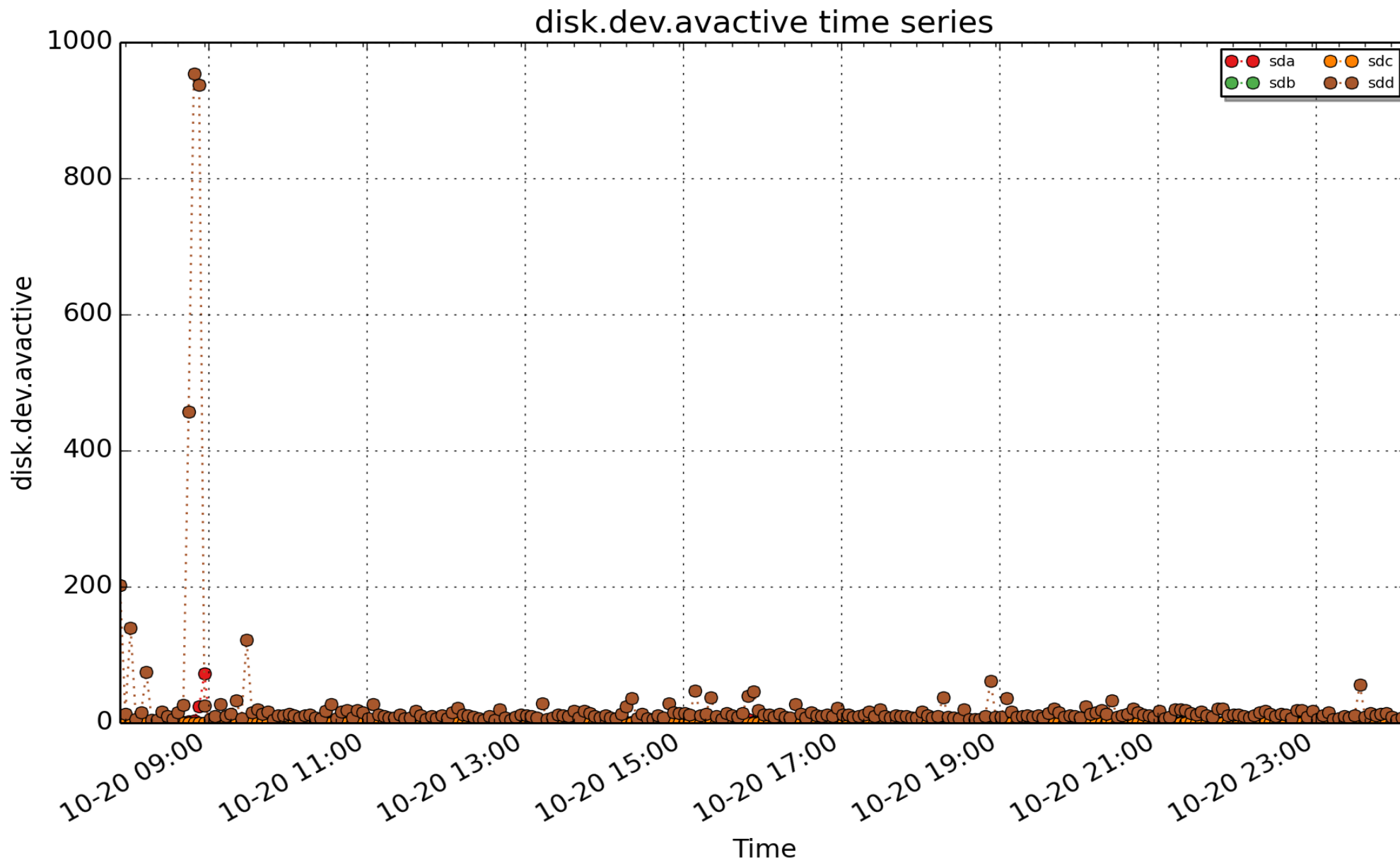


disk.all.write_merge: Total count of write requests that were merged with an already queued write request.
(count - U64) - *rate converted*

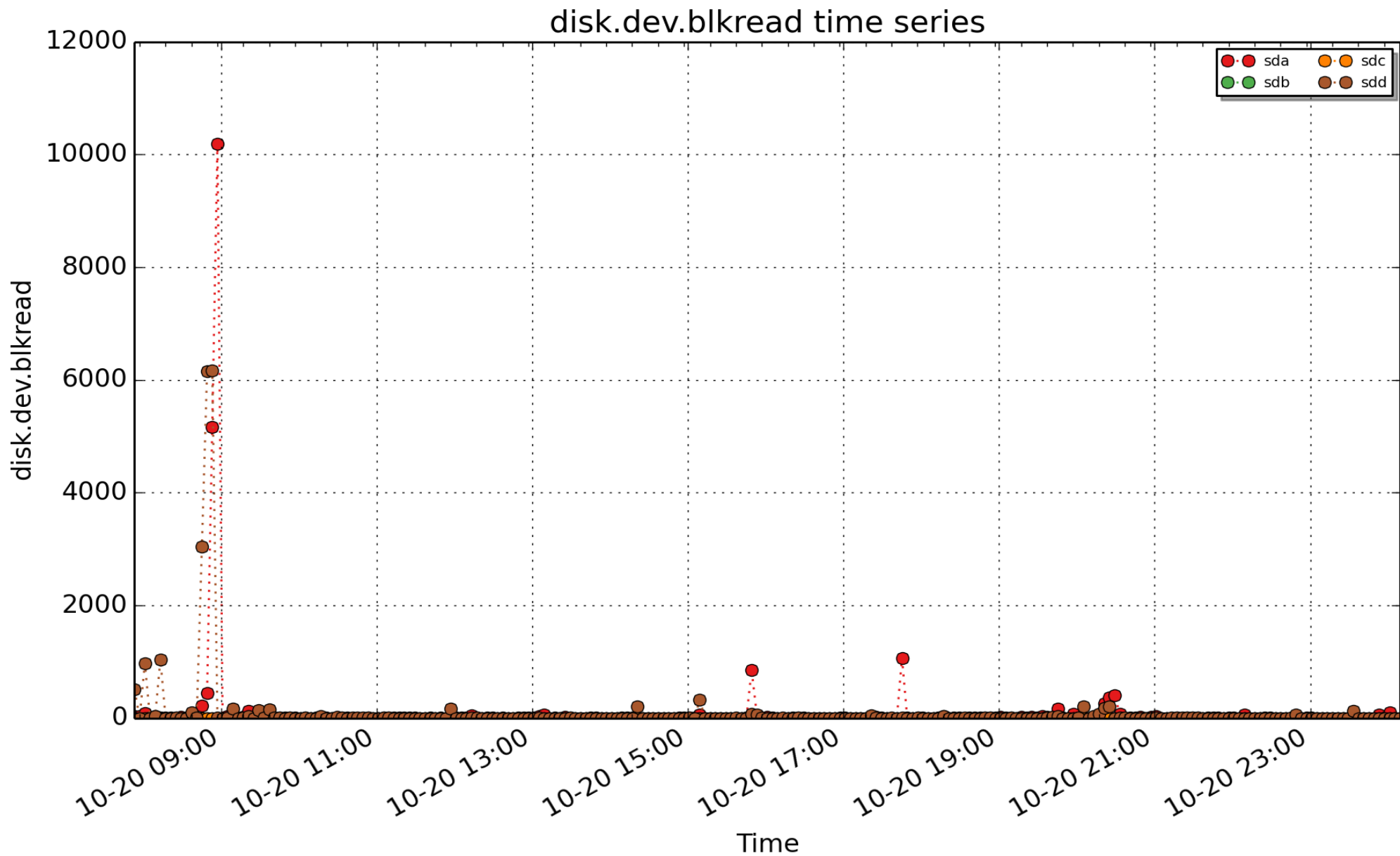


disk.all.write_rawactive: When converted to a rate, this metric represents the raw utilization of all disks during the sampling interval due to write requests. The accounting for this metric is only done on I/O completion and can thus result in more than a second's worth of IO being accounted for within any one second, leading to >100% utilisation. It is suitable mainly for use in calculations with other metrics, e.g.

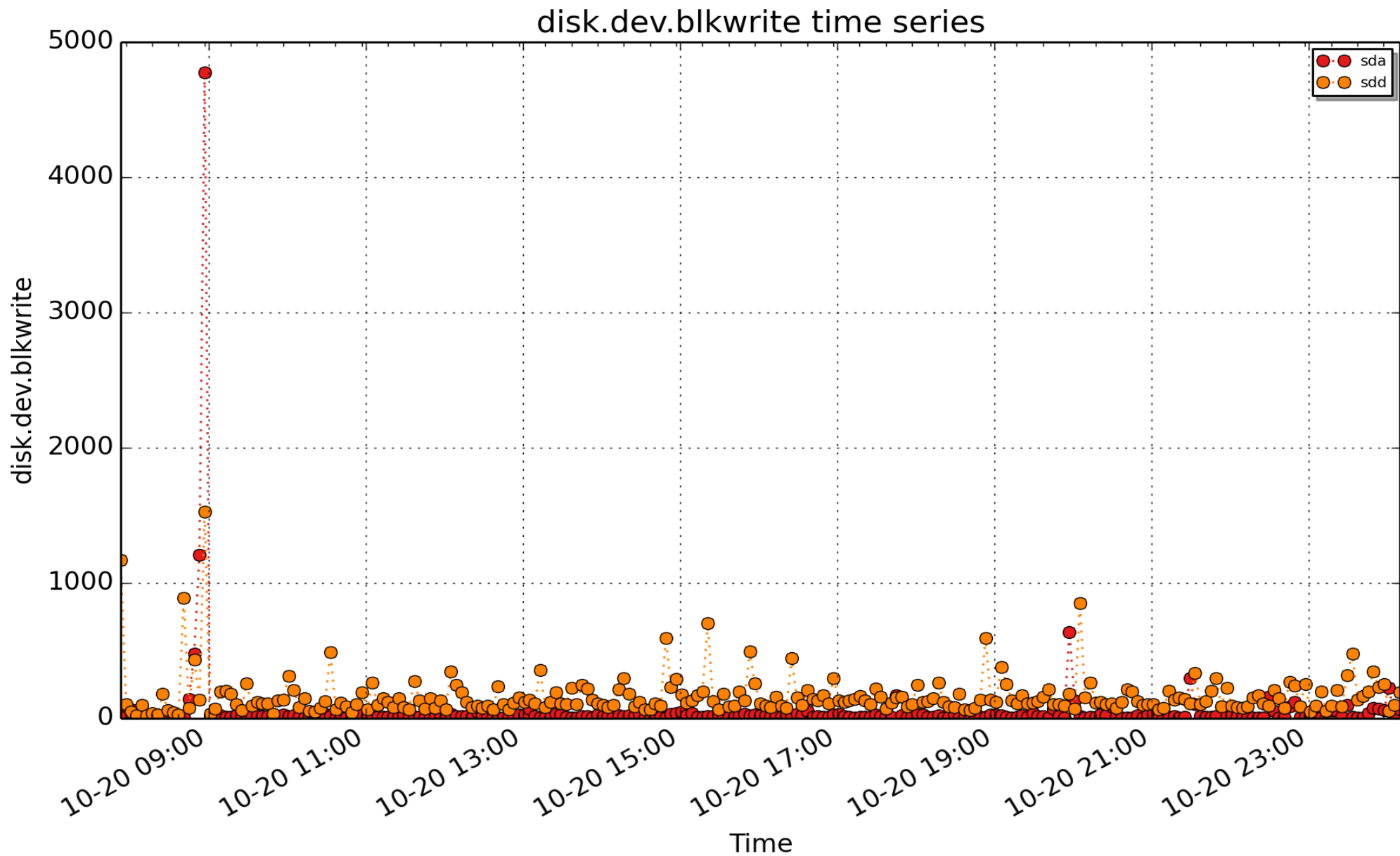
mirroring the result from existing performance tools: $\text{iostat.all.w_await} = \frac{\text{delta(disk.all.write_rawactive)}}{\text{delta(disk.all.write)}} \text{ (millisec - U64) - } \textit{rate converted}$



disk.dev.avactive: When converted to a rate, this metric represents the average utilization of the disk during the sampling interval. A value of 0.5 (or 50%) means the disk was active (i.e. busy) half the time. (millisec - U32) - *rate converted*



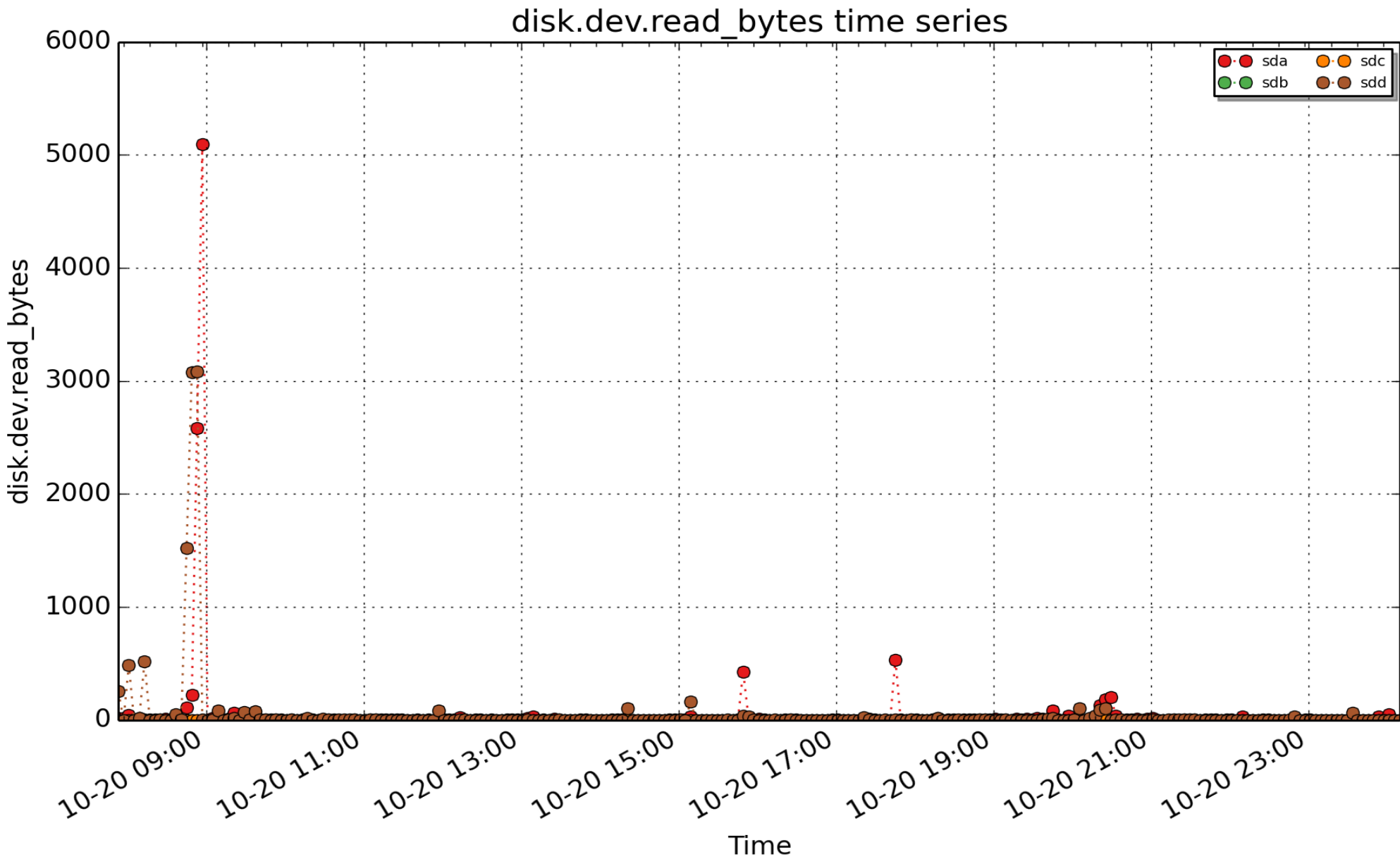
disk.dev.blkread: Cumulative number of disk block read operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*



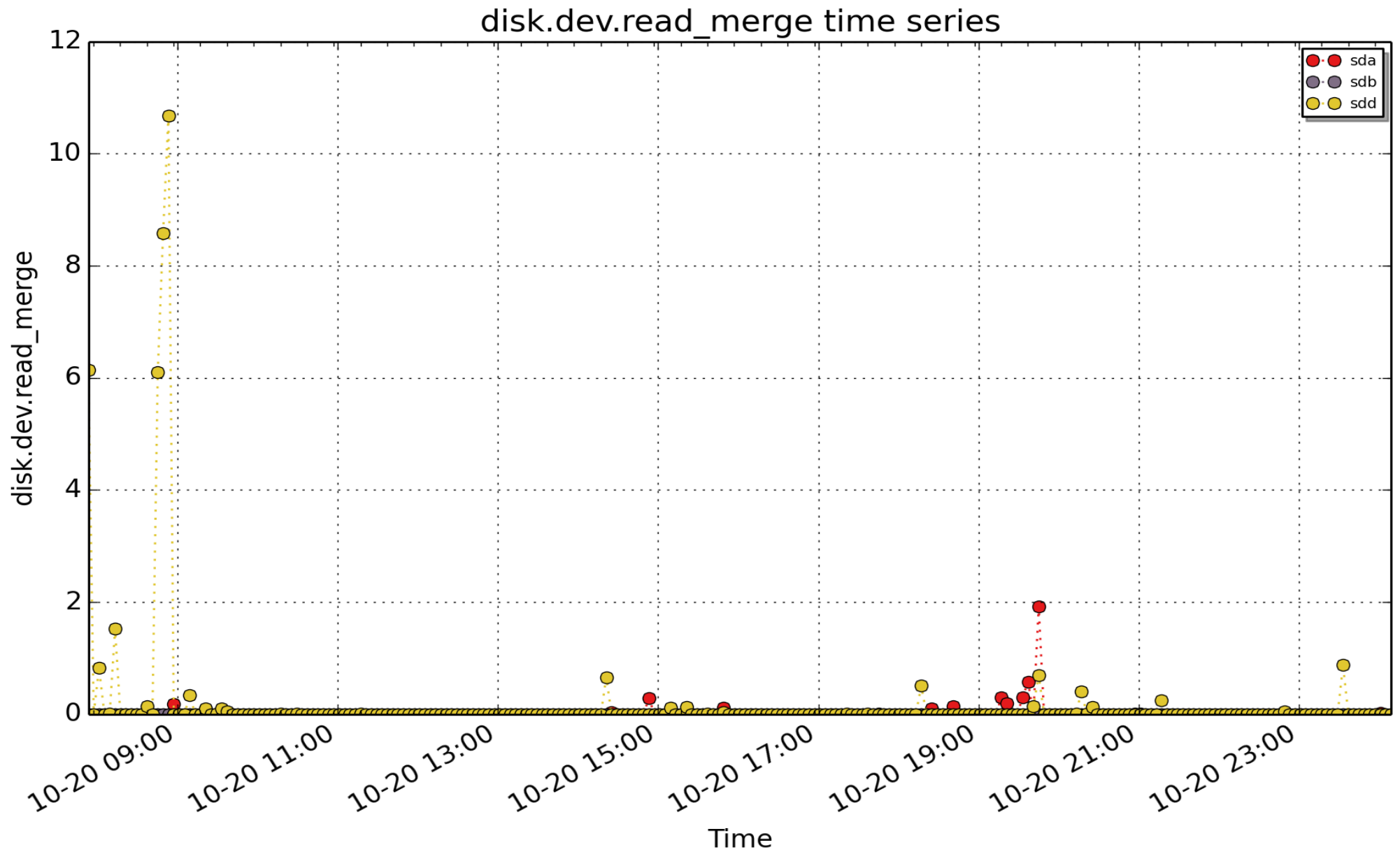
disk.dev.blkwrite: Cumulative number of disk block write operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*



disk.dev.read: Cumulative number of disk read operations since system boot time (subject to counter wrap).
(count - U64) - *rate converted*

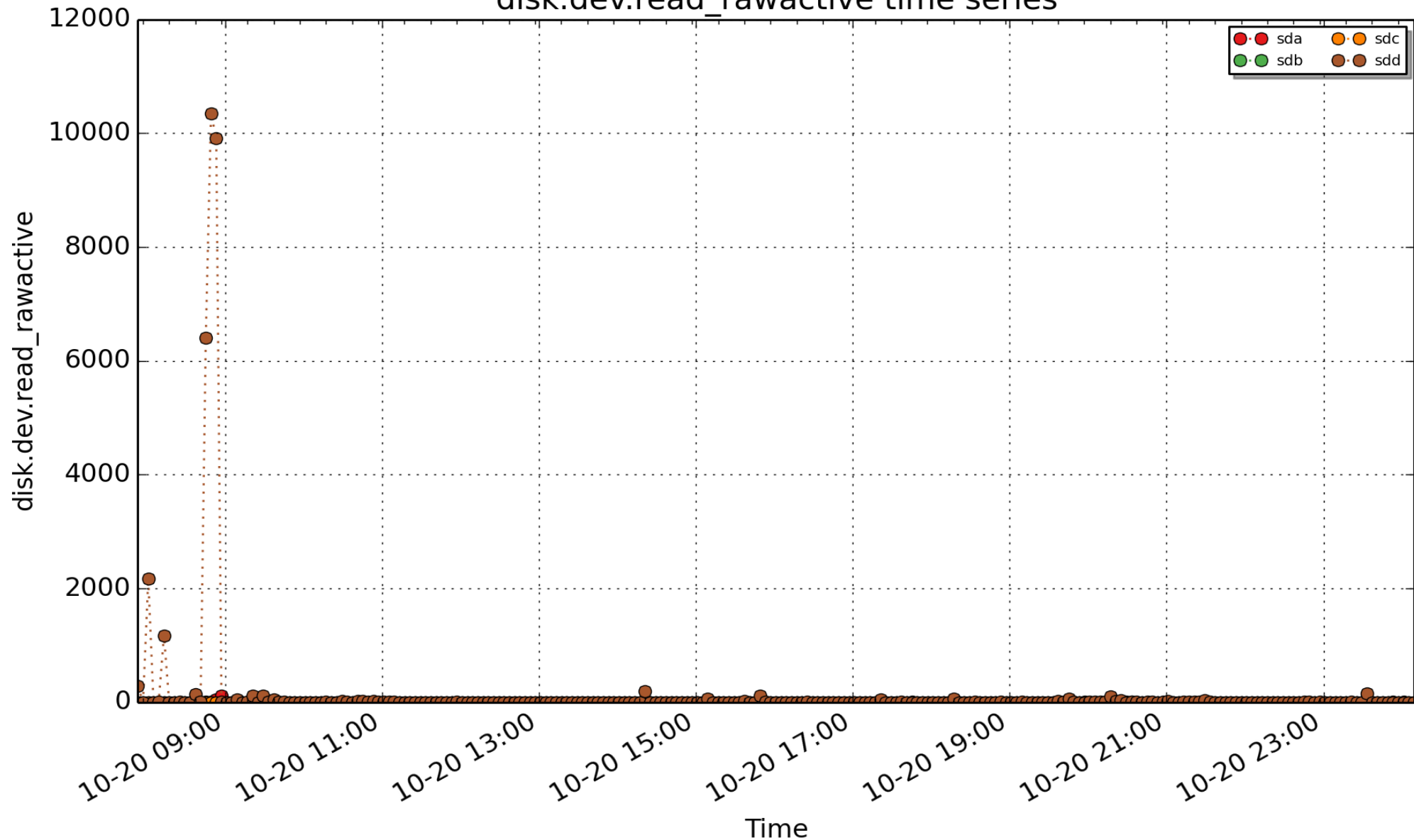


disk.dev.read_bytes: per-disk count of bytes read (Kbyte - U32) - *rate converted*



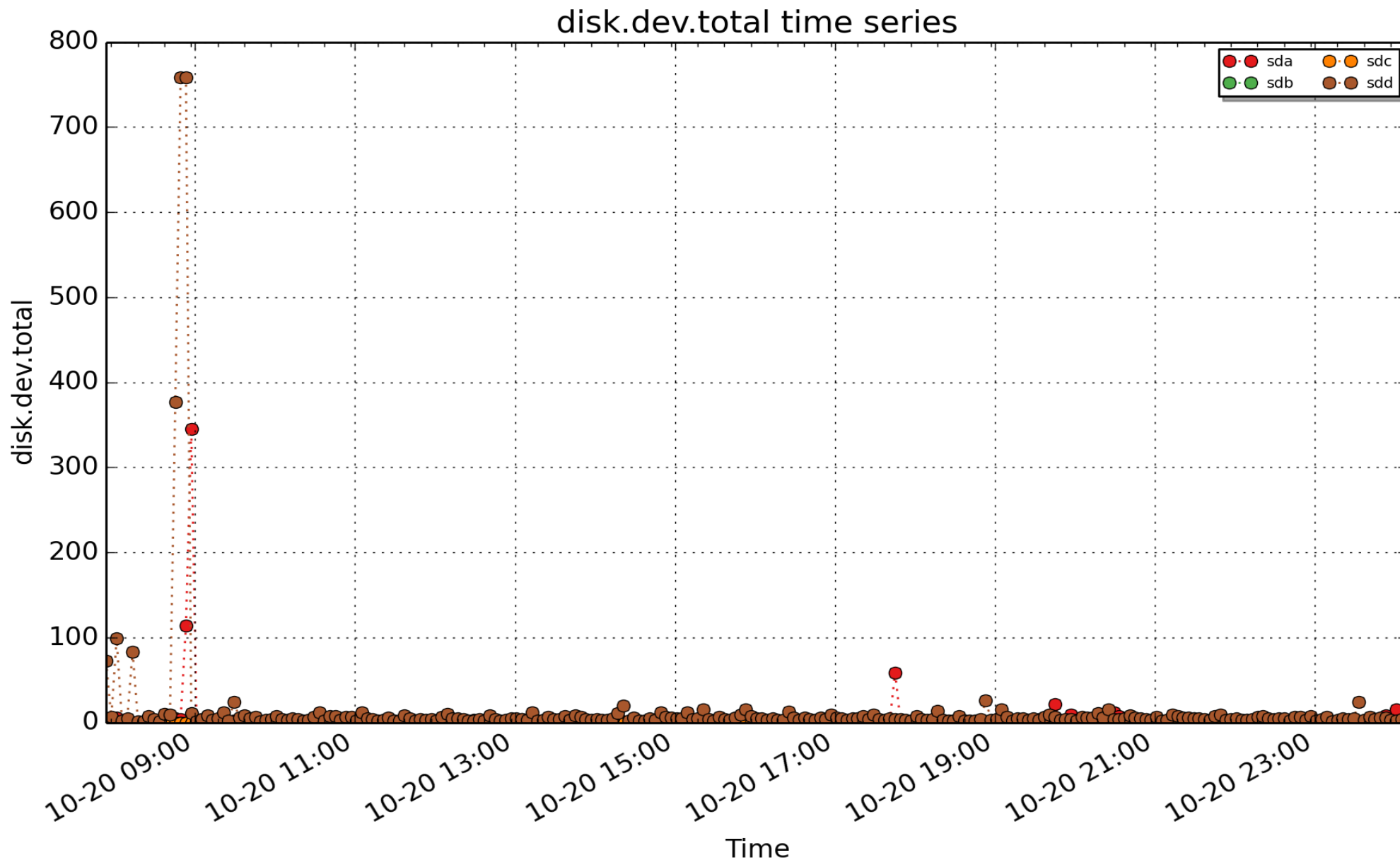
disk.dev.read_merge: Count of read requests that were merged with an already queued read request. (count - U64) - *rate converted*

disk.dev.read_rawactive time series



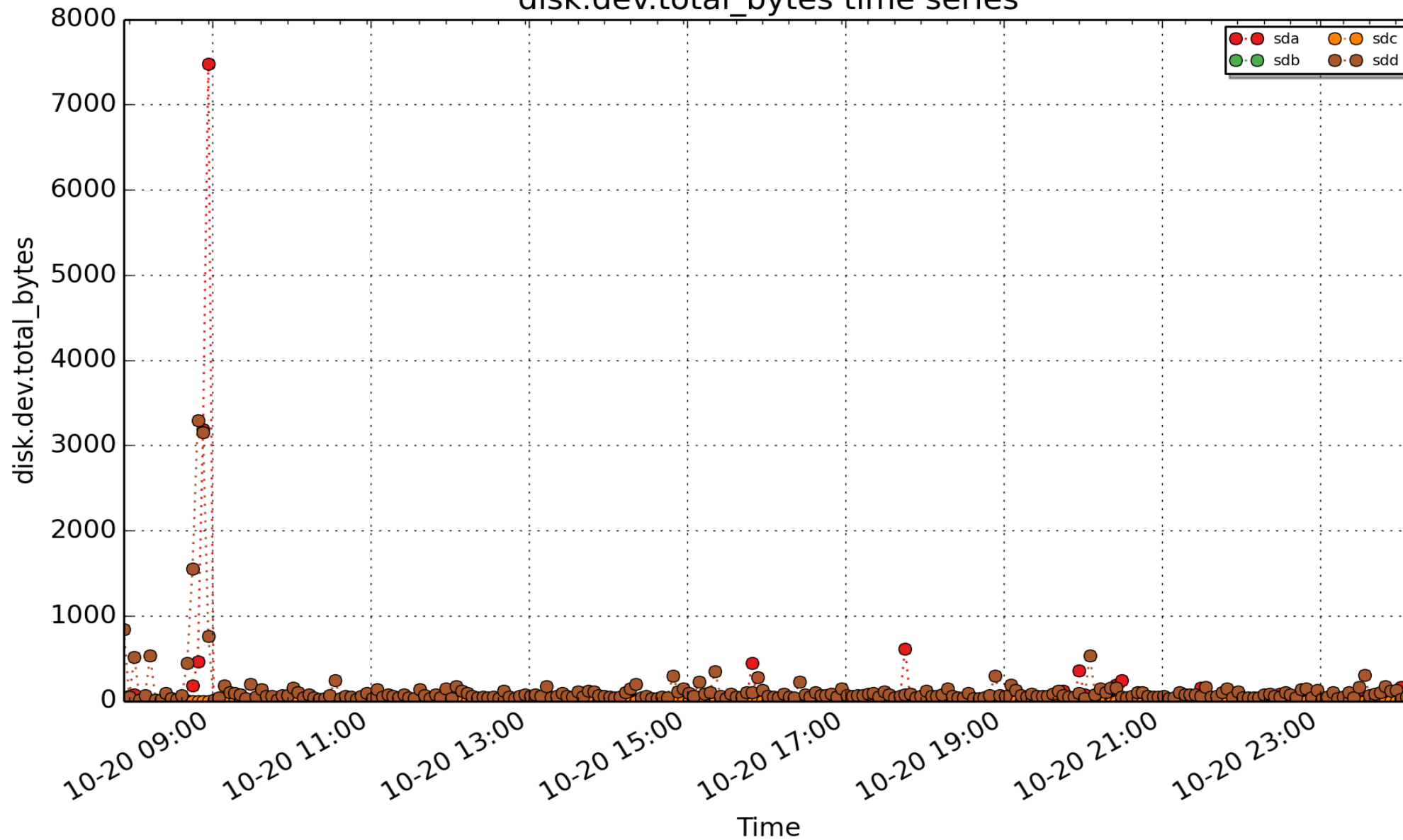
disk.dev.read_rawactive: When converted to a rate, this metric represents the raw utilization of the disk during the sampling interval as a result of reads. Accounting for this metric is only done on I/O completion and can thus result in more than a second's worth of IO being accounted for within any one second, leading to >100% utilisation. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from

existing performance tools: $\text{iostat.dev.r_await} = \text{delta}(\text{disk.dev.read_rawactive}) / \text{delta}(\text{disk.dev.read})$ (millisec
- U32) - *rate converted*

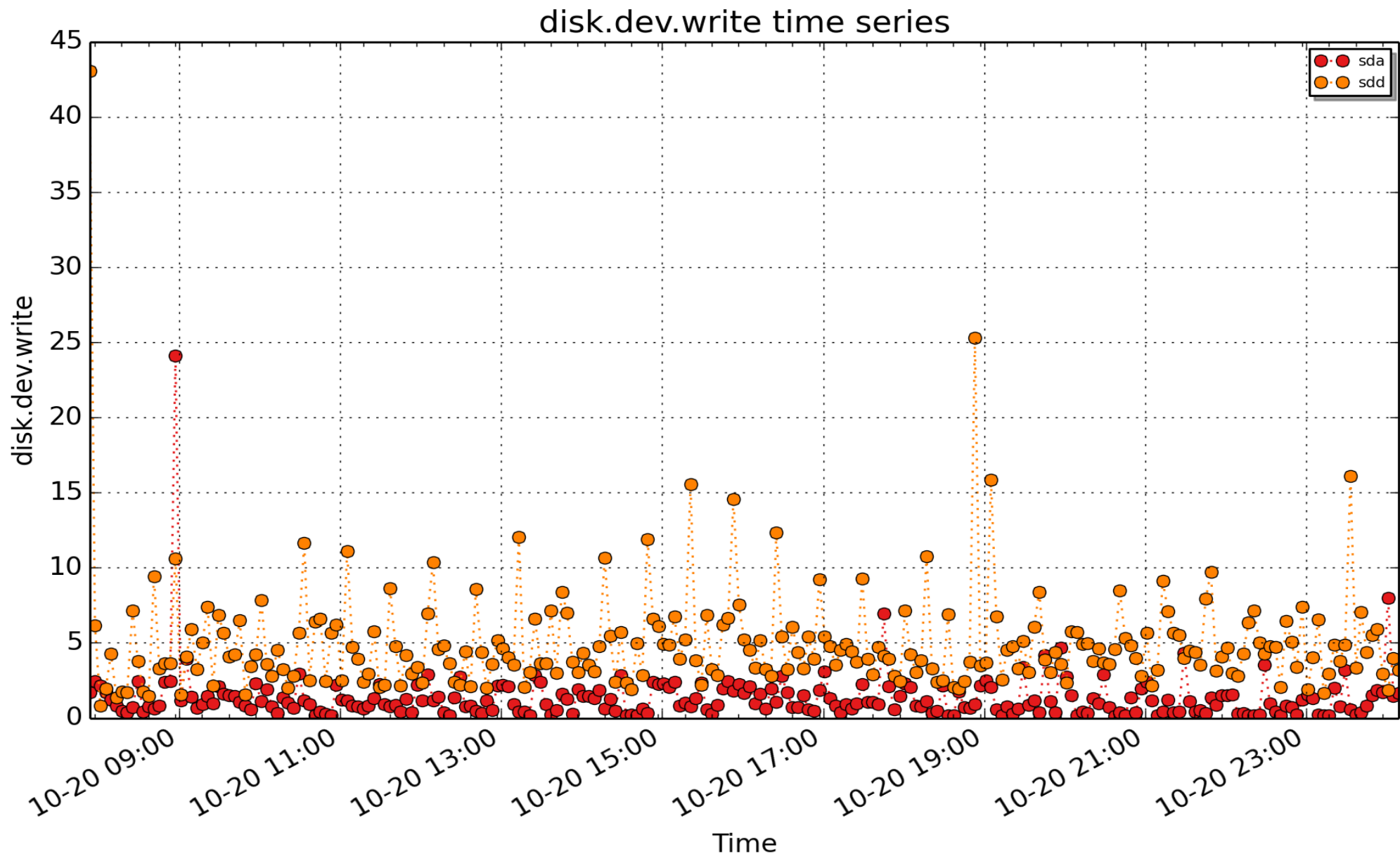


disk.dev.total: Cumulative number of disk read and write operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*

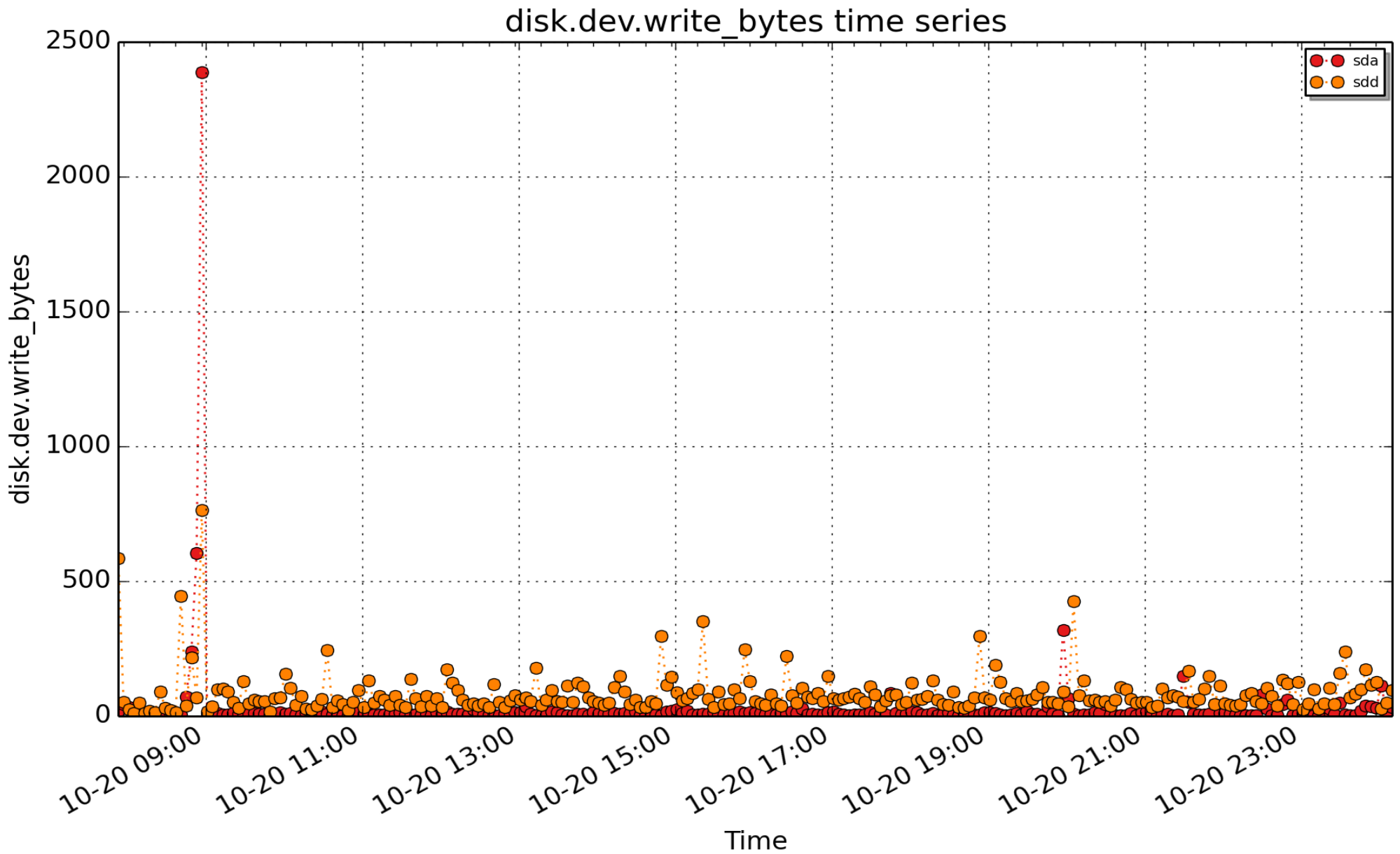
disk.dev.total_bytes time series



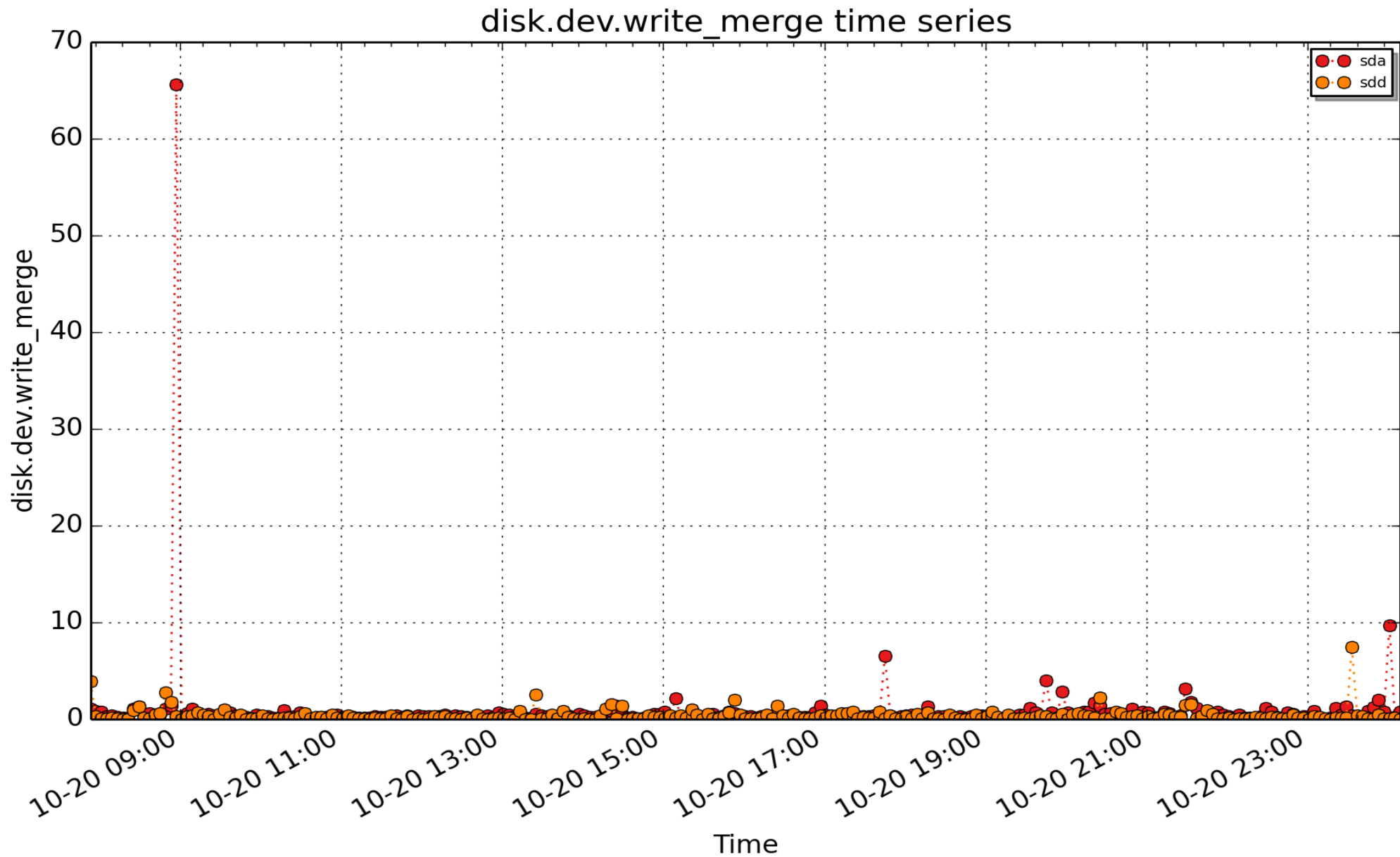
disk.dev.total_bytes: per-disk count of total bytes read and written (Kbyte - U32) - *rate converted*



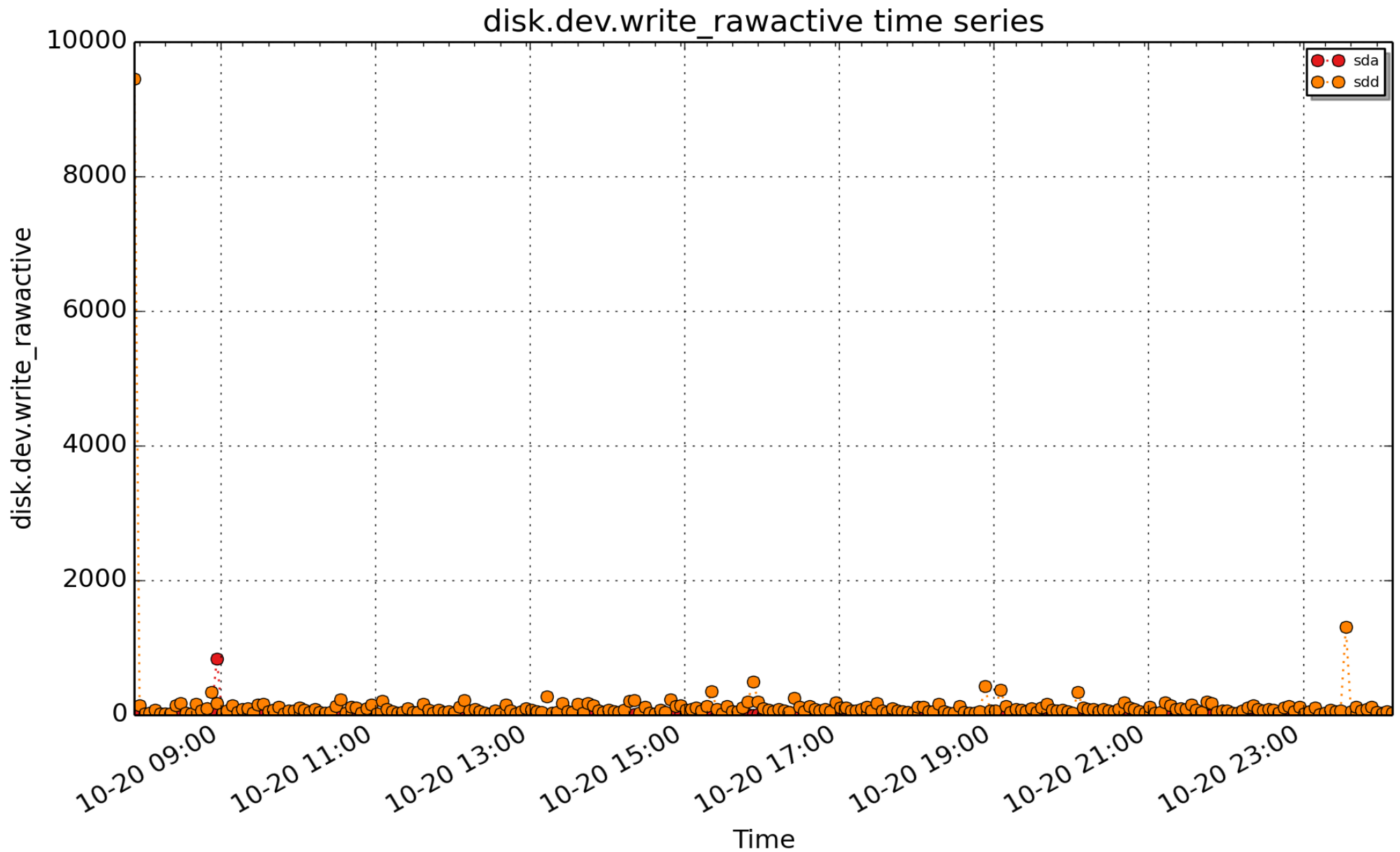
disk.dev.write: Cumulative number of disk write operations since system boot time (subject to counter wrap).
(count - U64) - *rate converted*



disk.dev.write_bytes: per-disk count of bytes written (Kbyte - U32) - *rate converted*

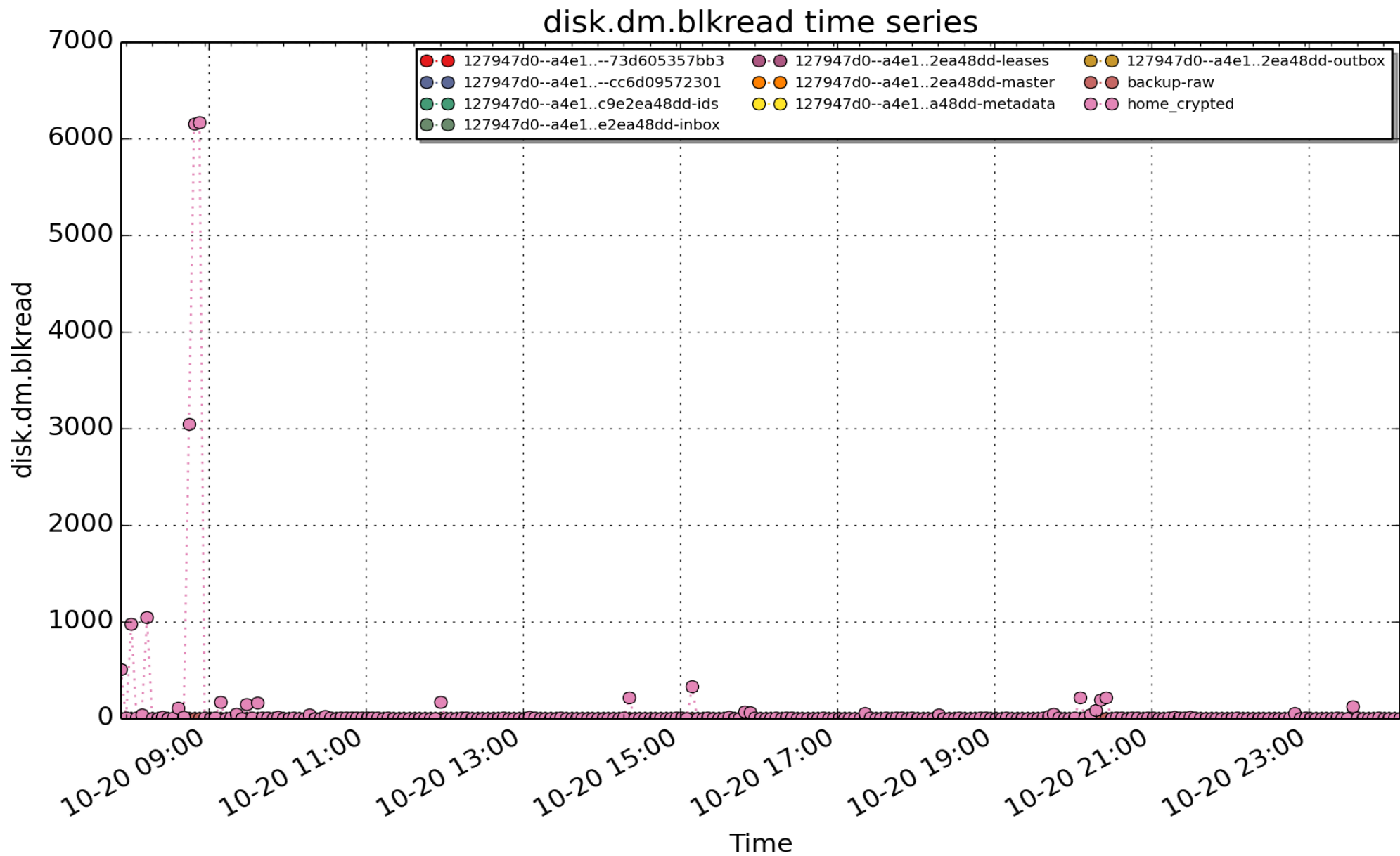


disk.dev.write_merge: Count of write requests that were merged with an already queued write request.
(count - U64) - *rate converted*

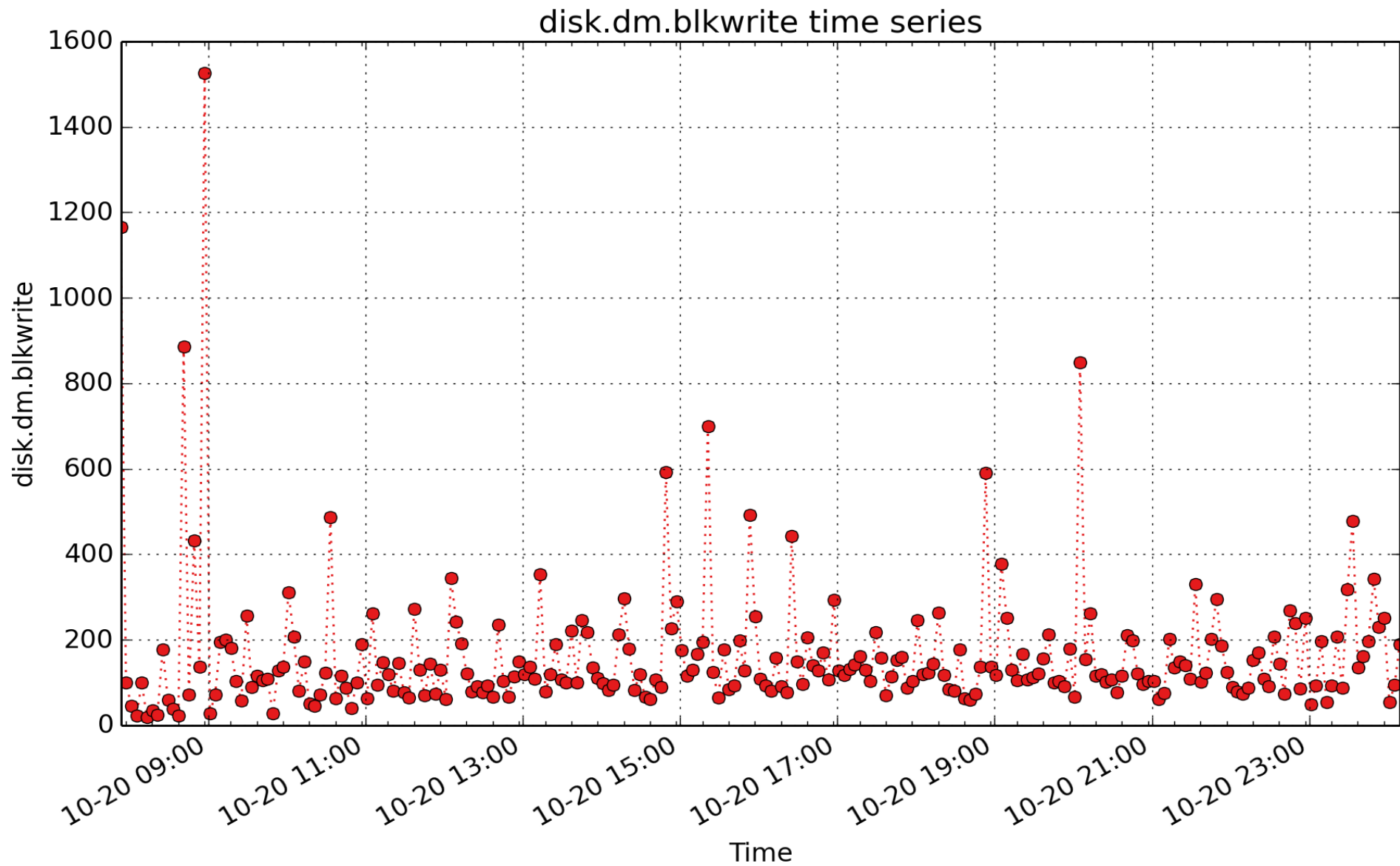


disk.dev.write_rawactive: When converted to a rate, this metric represents the raw utilization of the disk during the sampling interval as a result of writes. Accounting for this metric is only done on I/O completion and can thus result in more than a second's worth of IO being accounted for within any one second, leading to >100% utilisation. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from

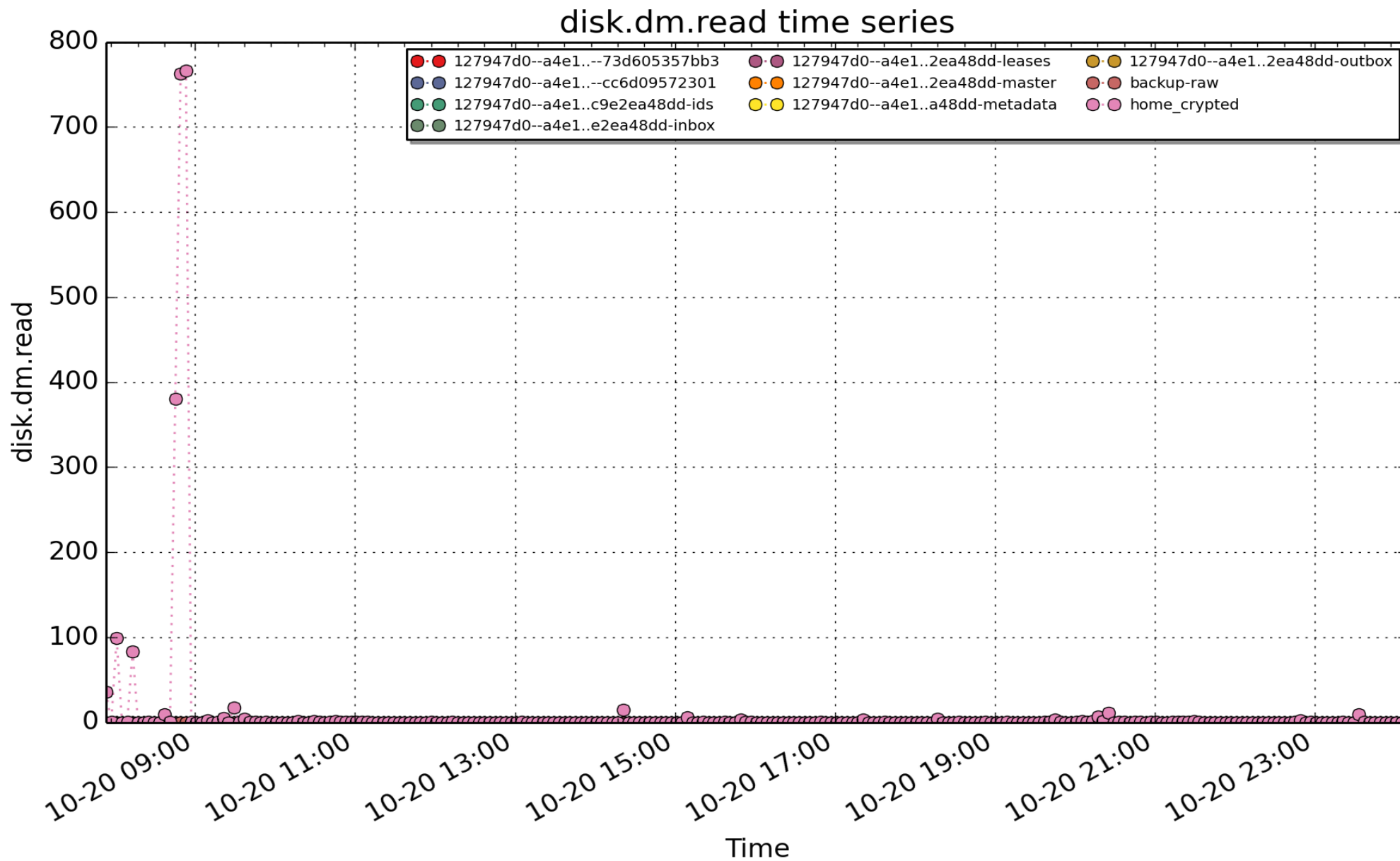
existing performance tools: $\text{iostat.dev.w_await} = \frac{\text{delta(disk.dev.write_rawactive)}}{\text{delta(disk.dev.write)}}$
(millisec - U32) - *rate converted*



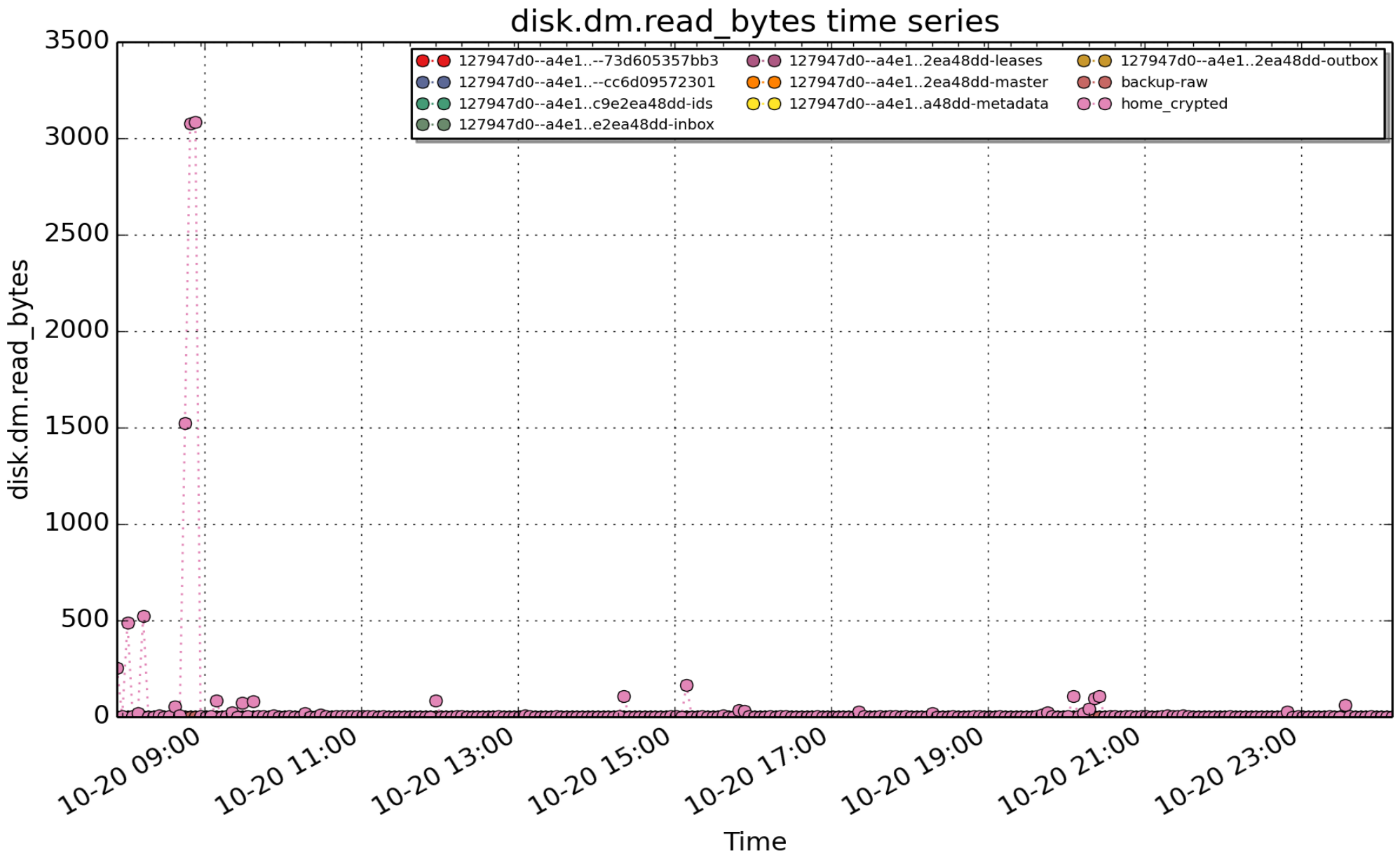
disk.dm.blkread: per-device-mapper device block read operations (count - U64) - *rate converted*



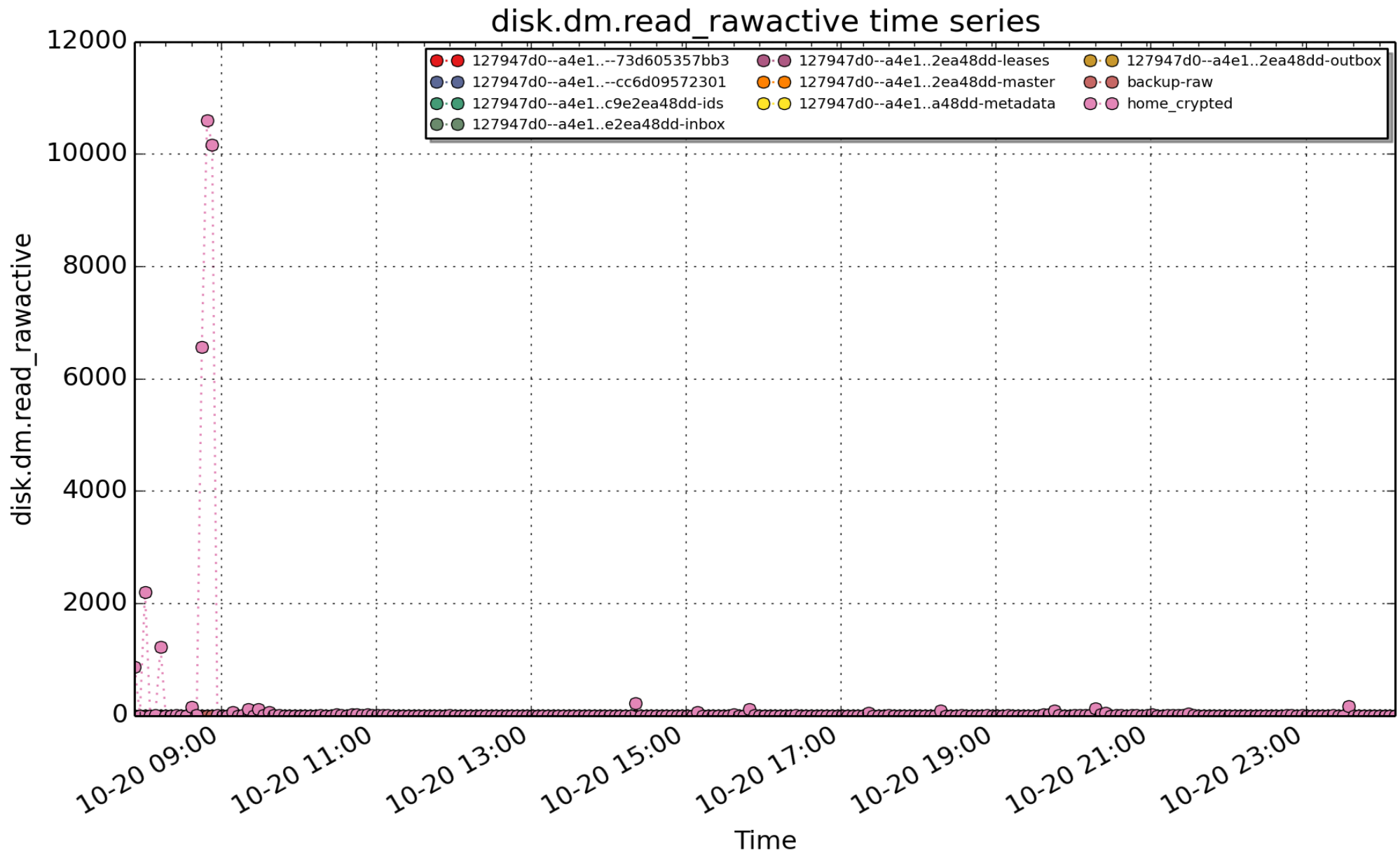
disk.dm.blkwrite: per-device-mapper device block write operations (count - U64) - *rate converted*



disk.dm.read: per-device-mapper device read operations (count - U64) - *rate converted*

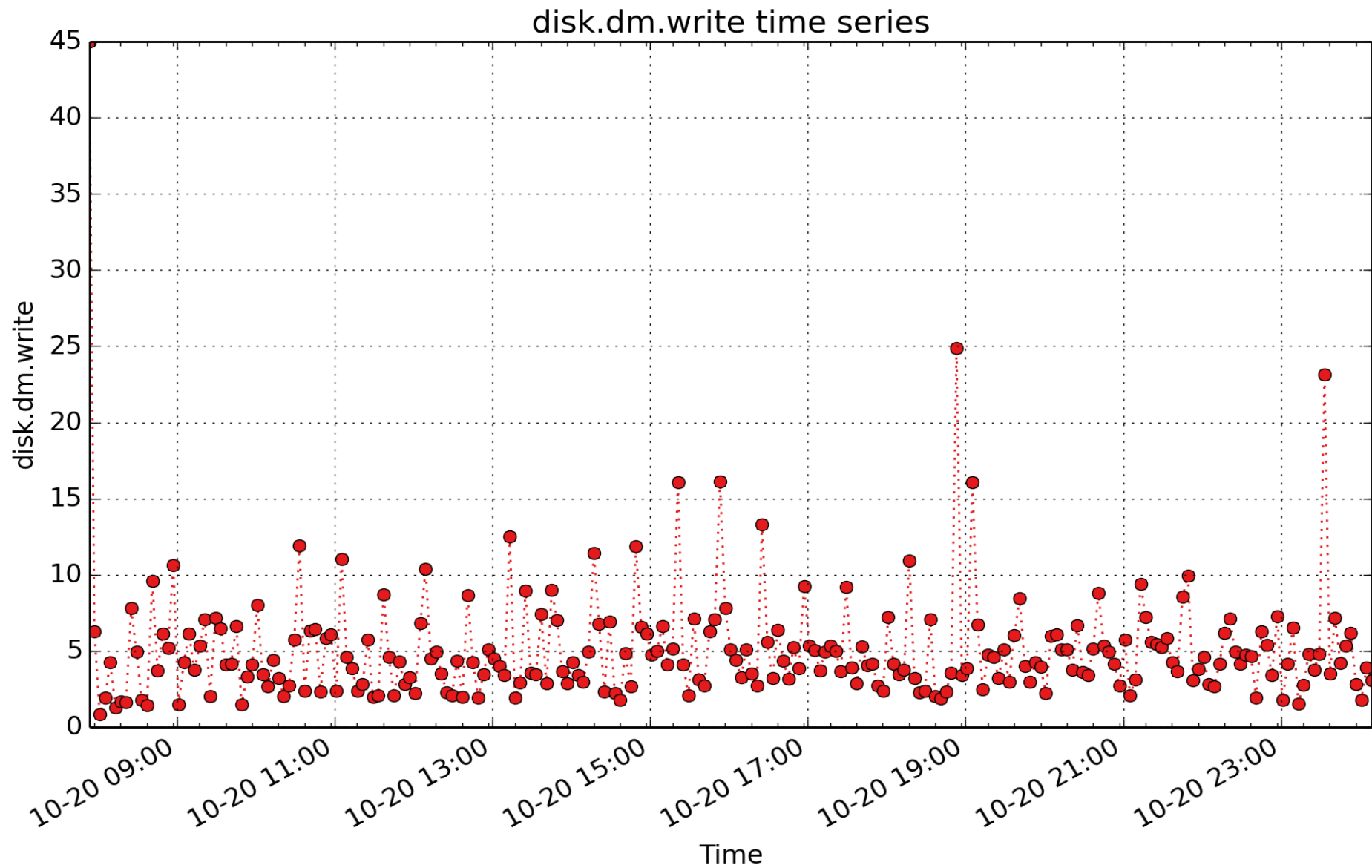


disk.dm.read_bytes: per-device-mapper device count of bytes read (Kbyte - U32) - *rate converted*

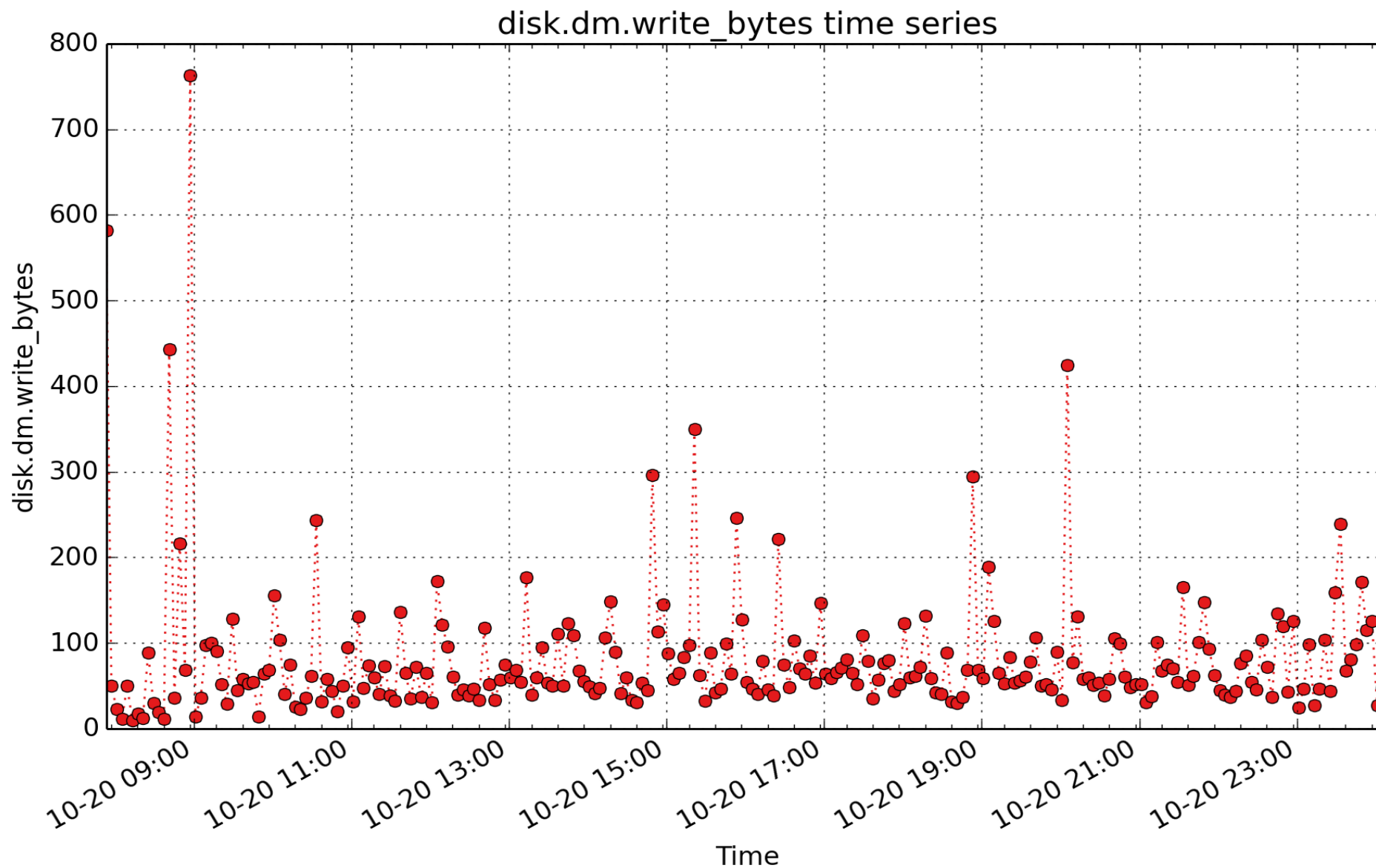


disk.dm.read_rawactive: When converted to a rate, this metric represents the raw utilization of the device during the sampling interval as a result of reads. Accounting for this metric is only done on I/O completion and can thus result in more than a second's worth of IO being accounted for within any one second, leading to >100% utilisation. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from

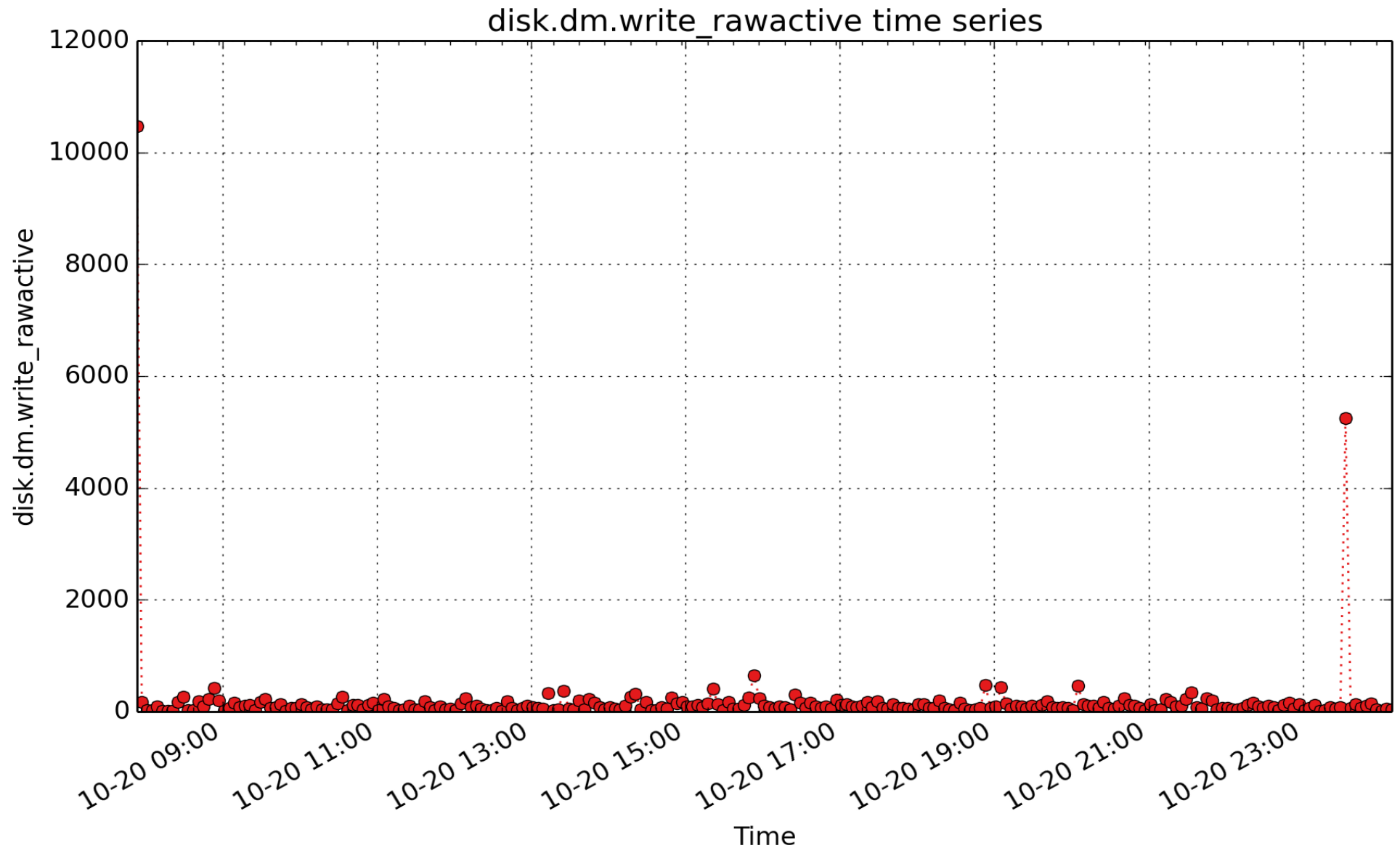
existing performance tools: $\text{iostat.dm.r_await} = \frac{\text{delta(disk.dm.read_rawactive)}}{\text{delta(disk.dm.read)}} \text{ (millisec - U32)}$ - *rate converted*



disk.dm.write: per-device-mapper device write operations (count - U64) - *rate converted*

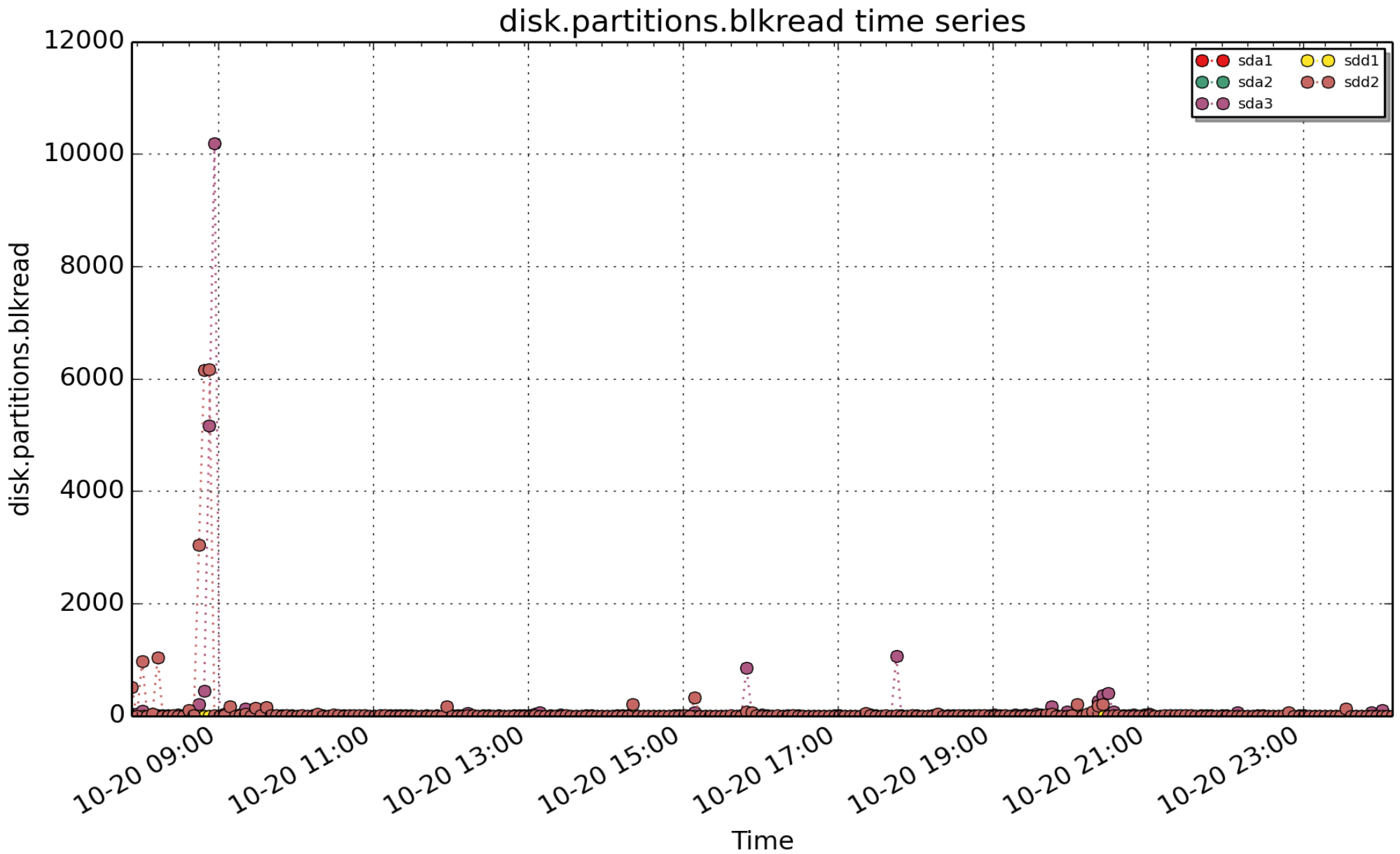


disk.dm.write_bytes: per-device-mapper device count of bytes written (Kbyte - U32) - *rate converted*



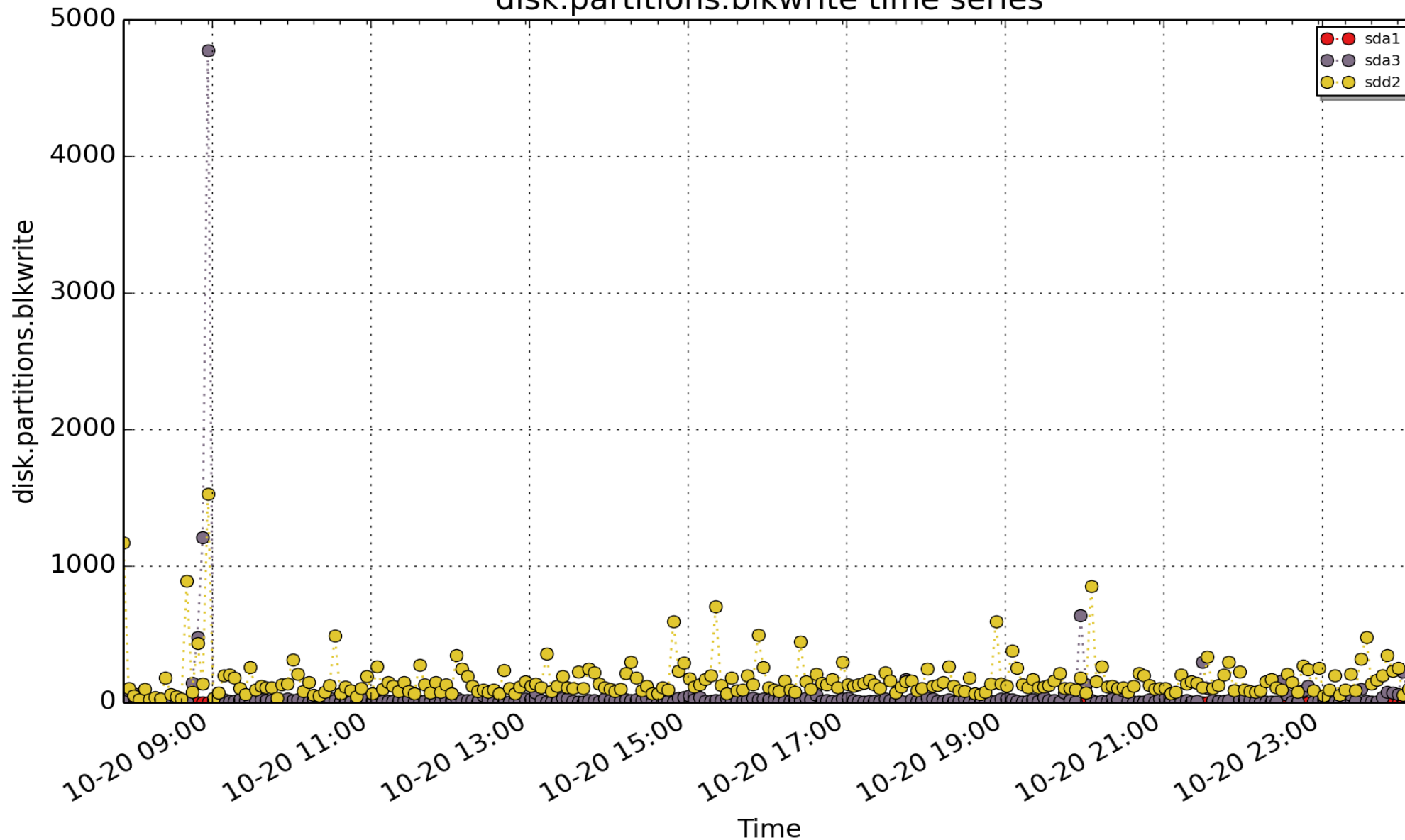
disk.dm.write_rawactive: When converted to a rate, this metric represents the raw utilization of the device during the sampling interval as a result of writes. Accounting for this metric is only done on I/O completion and can thus result in more than a second's worth of I/O being accounted for within any one second, leading to >100% utilisation. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from

existing performance tools: $\text{iostat.dm.w_await} = \text{delta}(\text{disk.dm.write_rawactive}) / \text{delta}(\text{disk.dm.write})$ (millisec
- U32) - *rate converted*

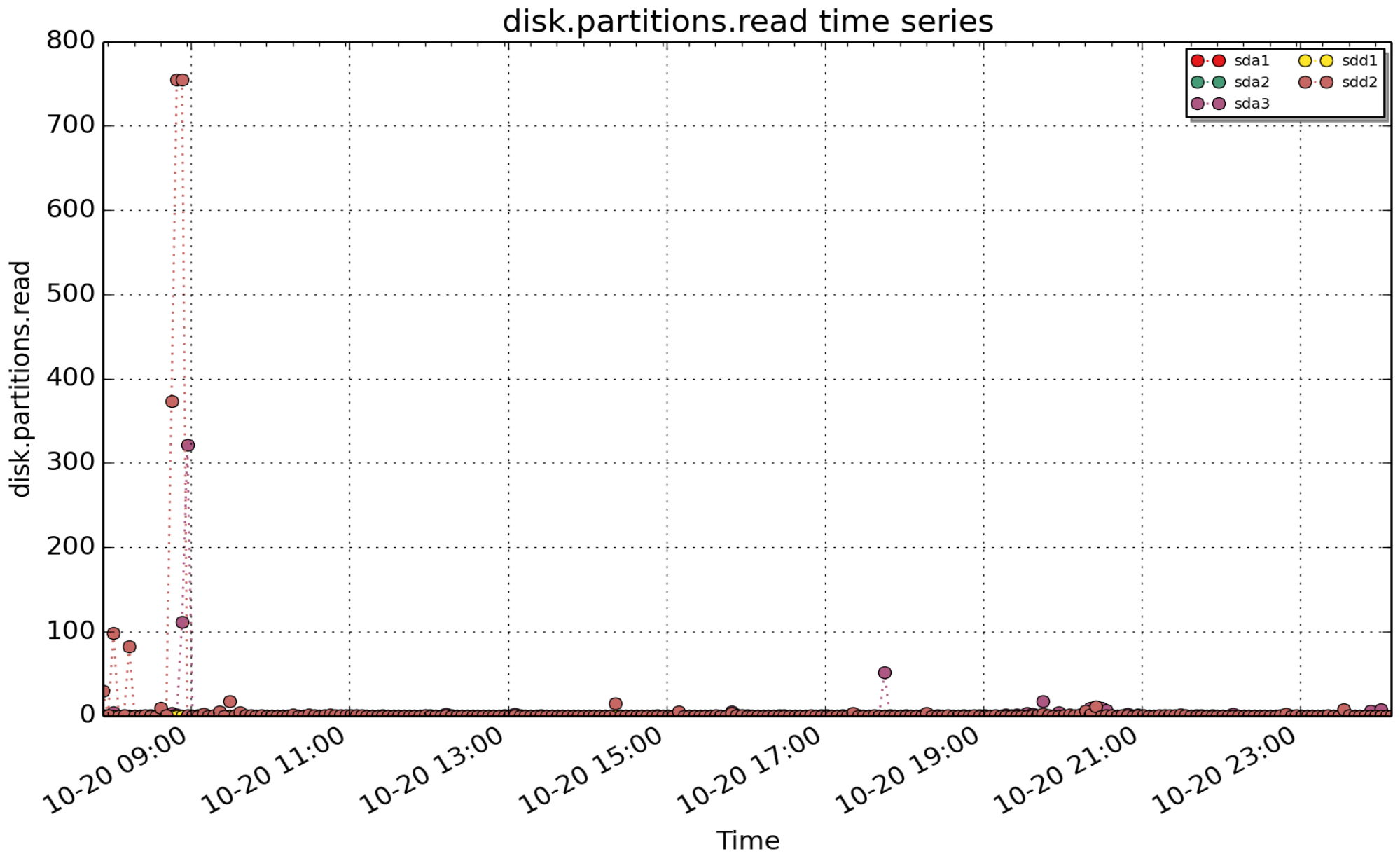


disk.partitions.blkread: Cumulative number of disk block read operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U32) - *rate converted*

disk.partitions.blkwrite time series

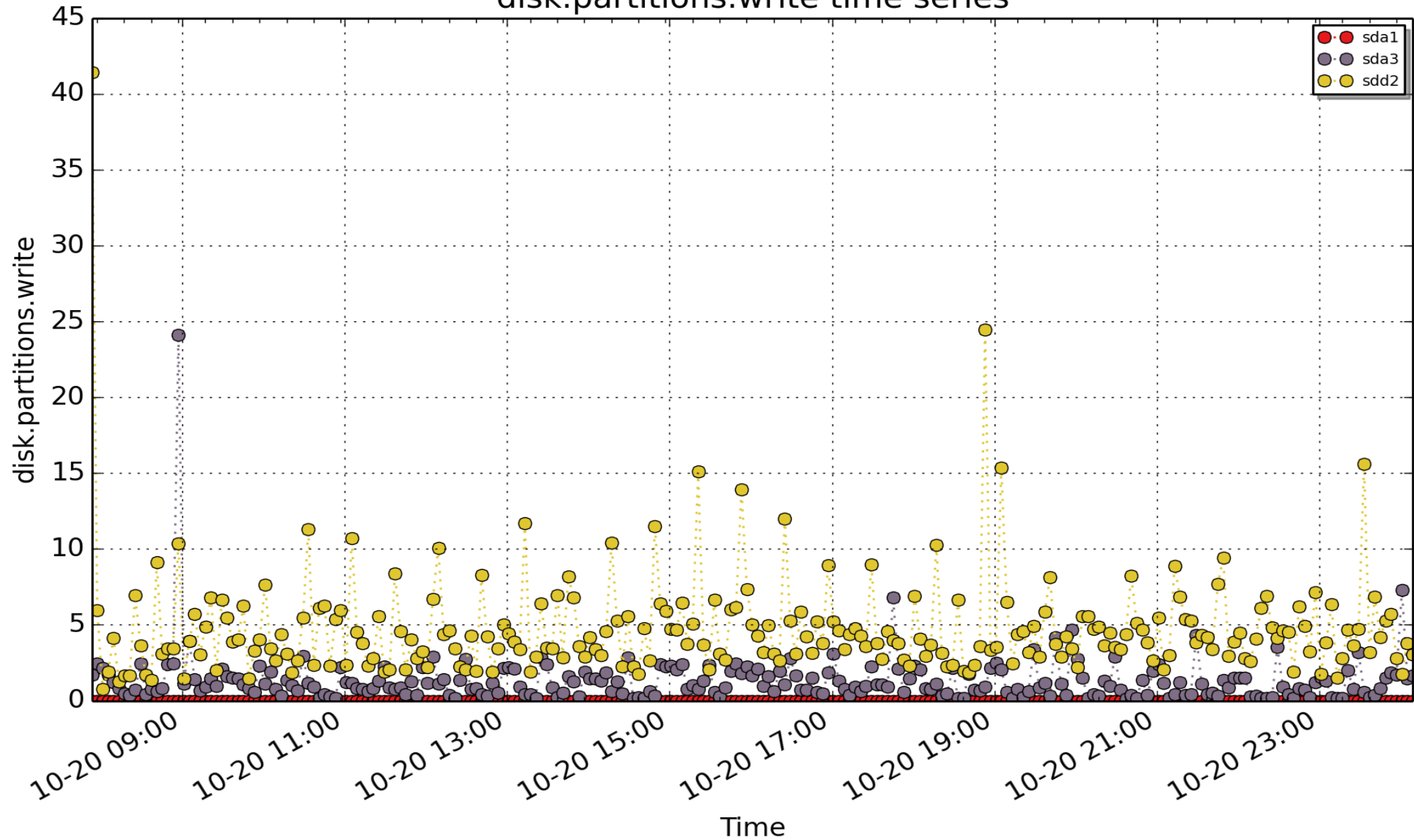


disk.partitions.blkwrite: Cumulative number of disk block write operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U32) - *rate converted*

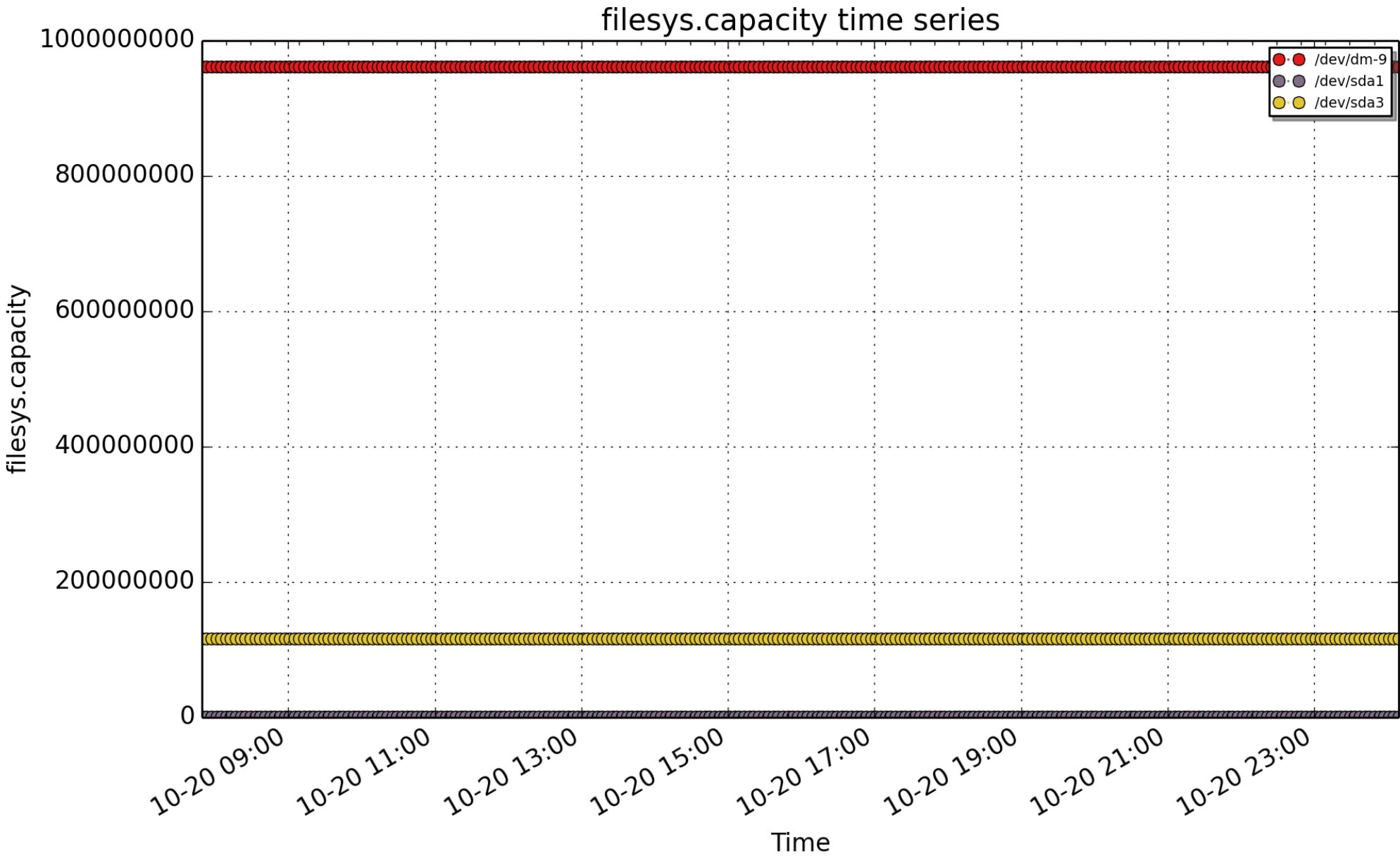


disk.partitions.read: Cumulative number of disk read operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U32) - *rate converted*

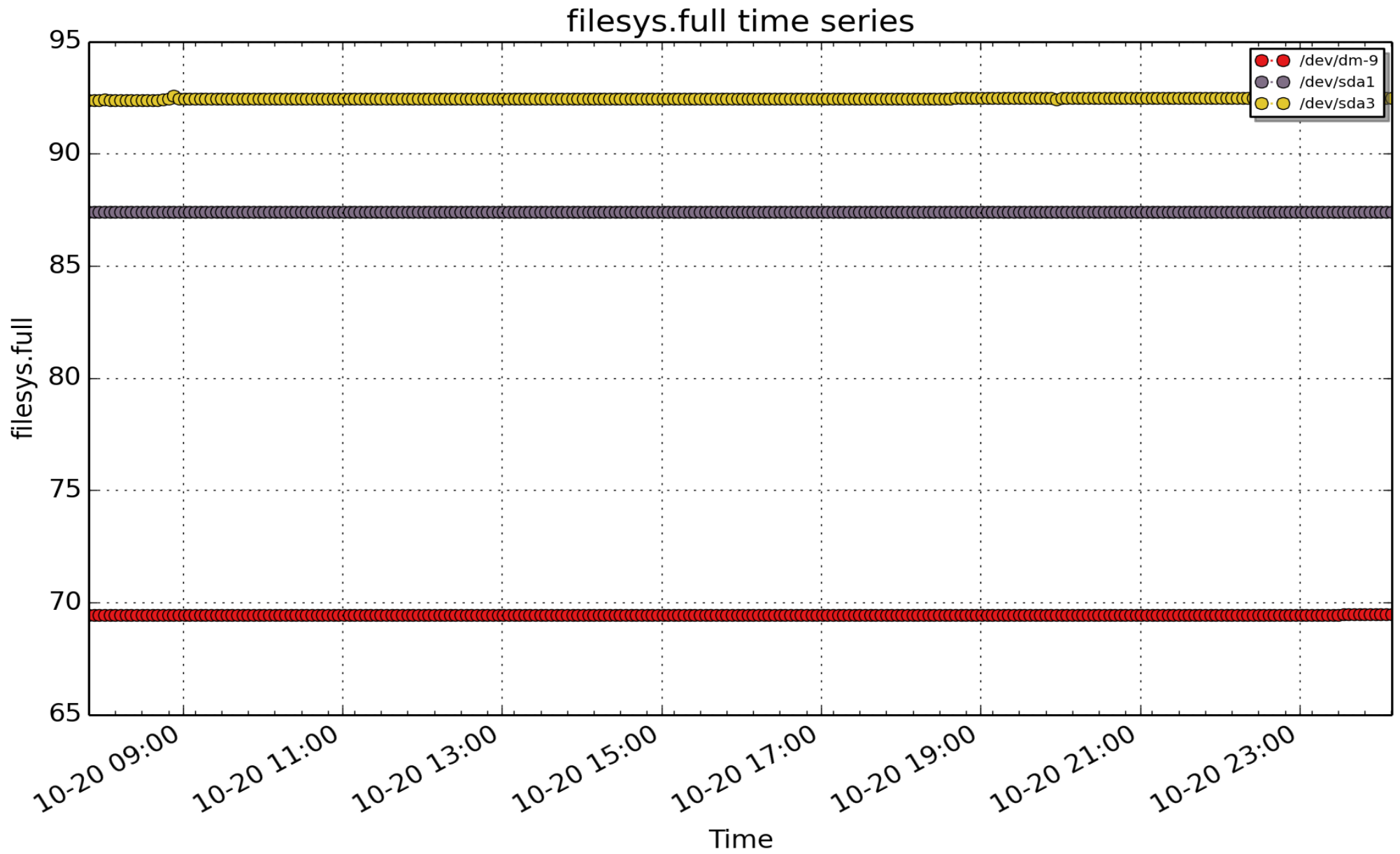
disk.partitions.write time series



disk.partitions.write: Cumulative number of disk write operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U32) - *rate converted*

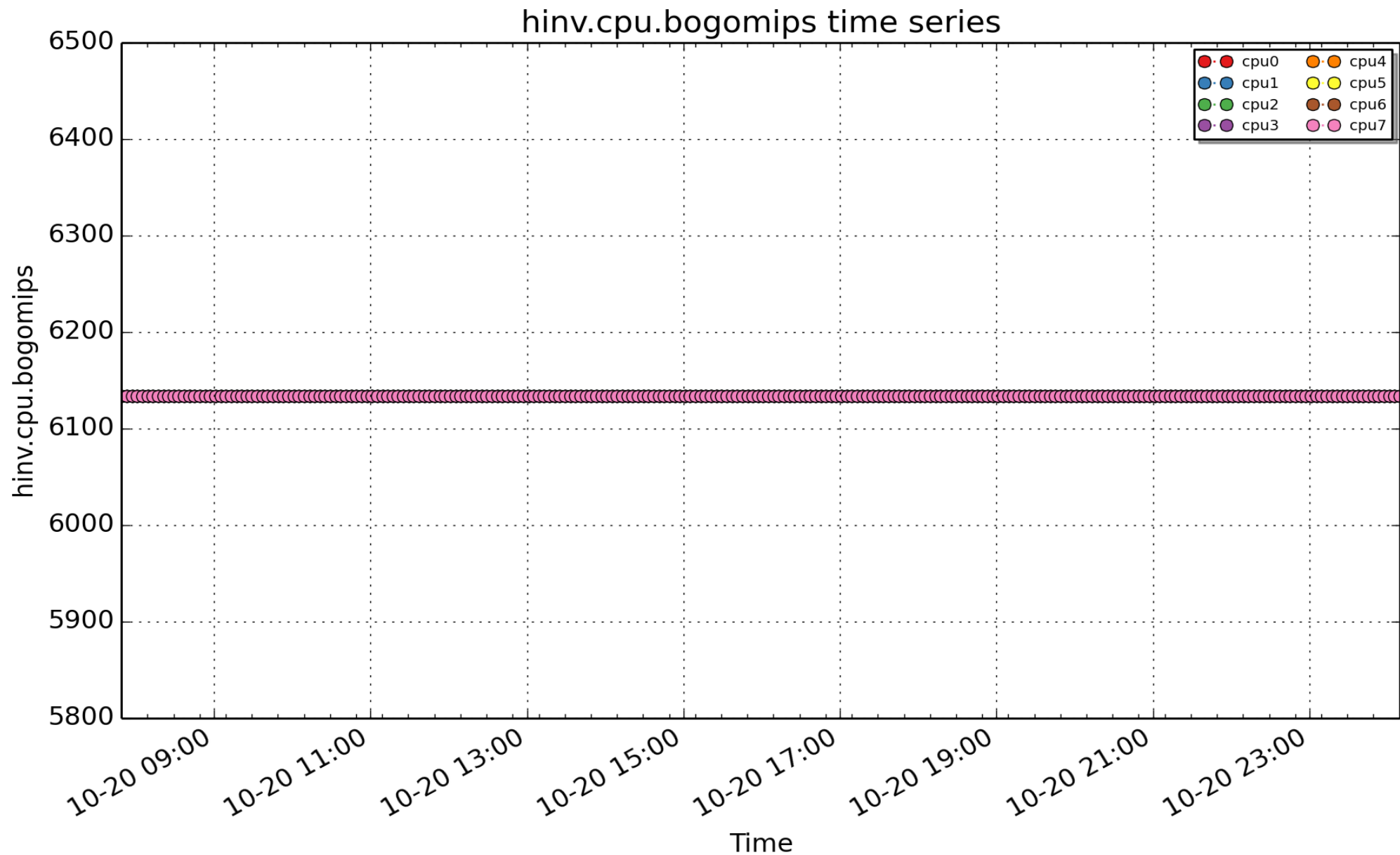


filesystem.capacity: Total capacity of mounted filesystem (Kbytes) (Kbyte - U64)



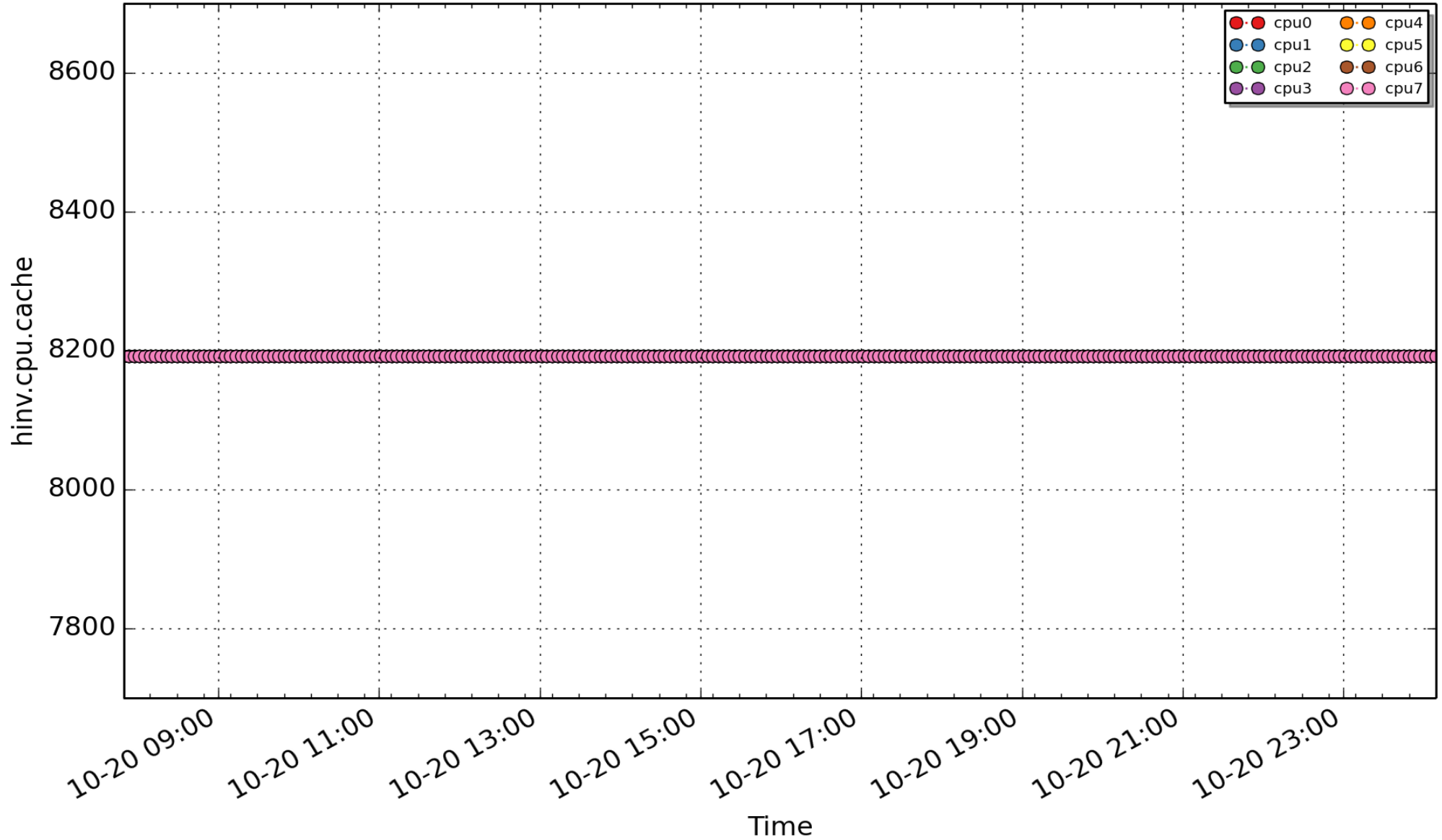
filesystem.full: Percentage of filesystem in use (- DOUBLE)

Hinv



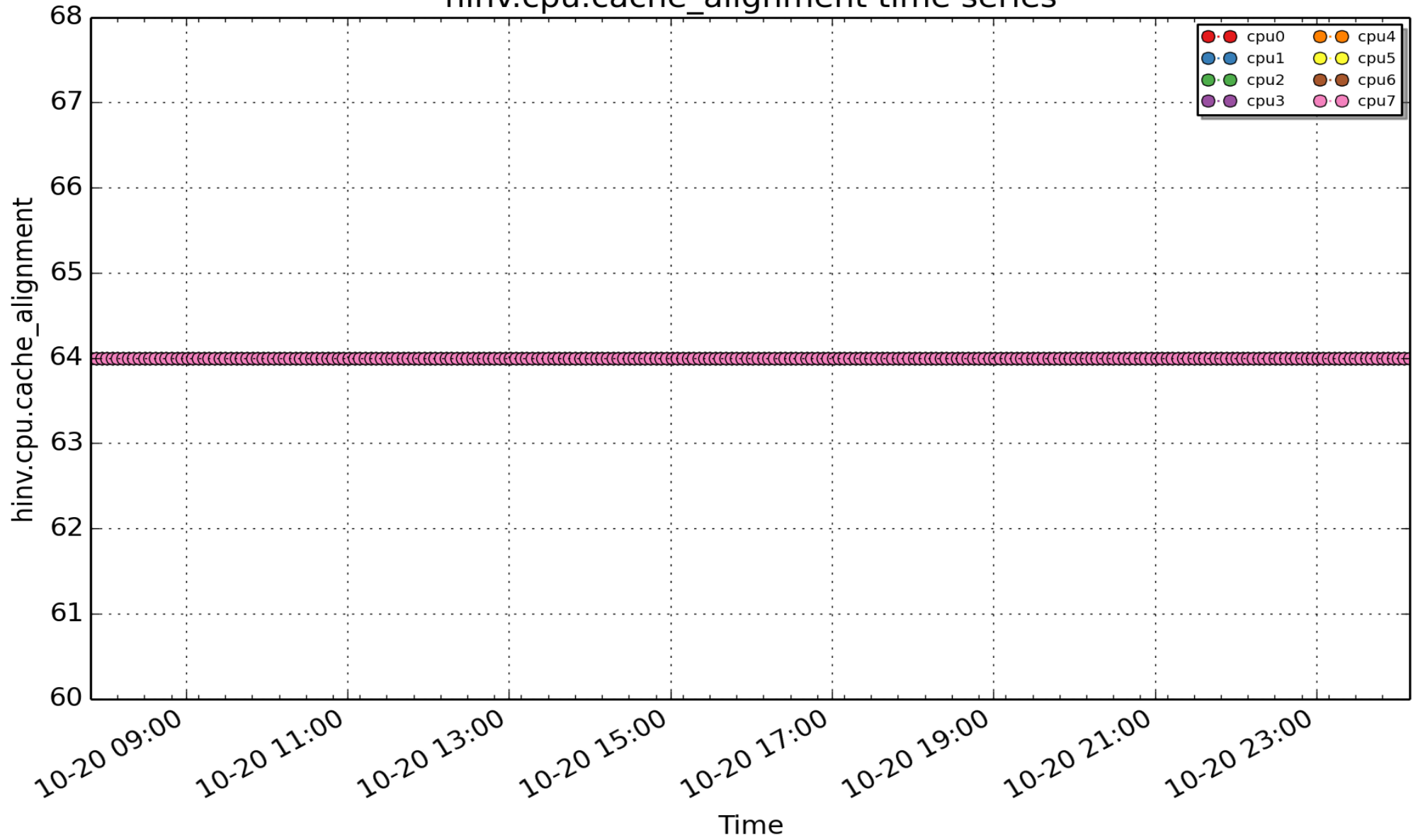
hinv.cpu.bogomips: bogo mips rating for each CPU as reported by /proc/cpuinfo (- FLOAT)

hinv.cpu.cache time series

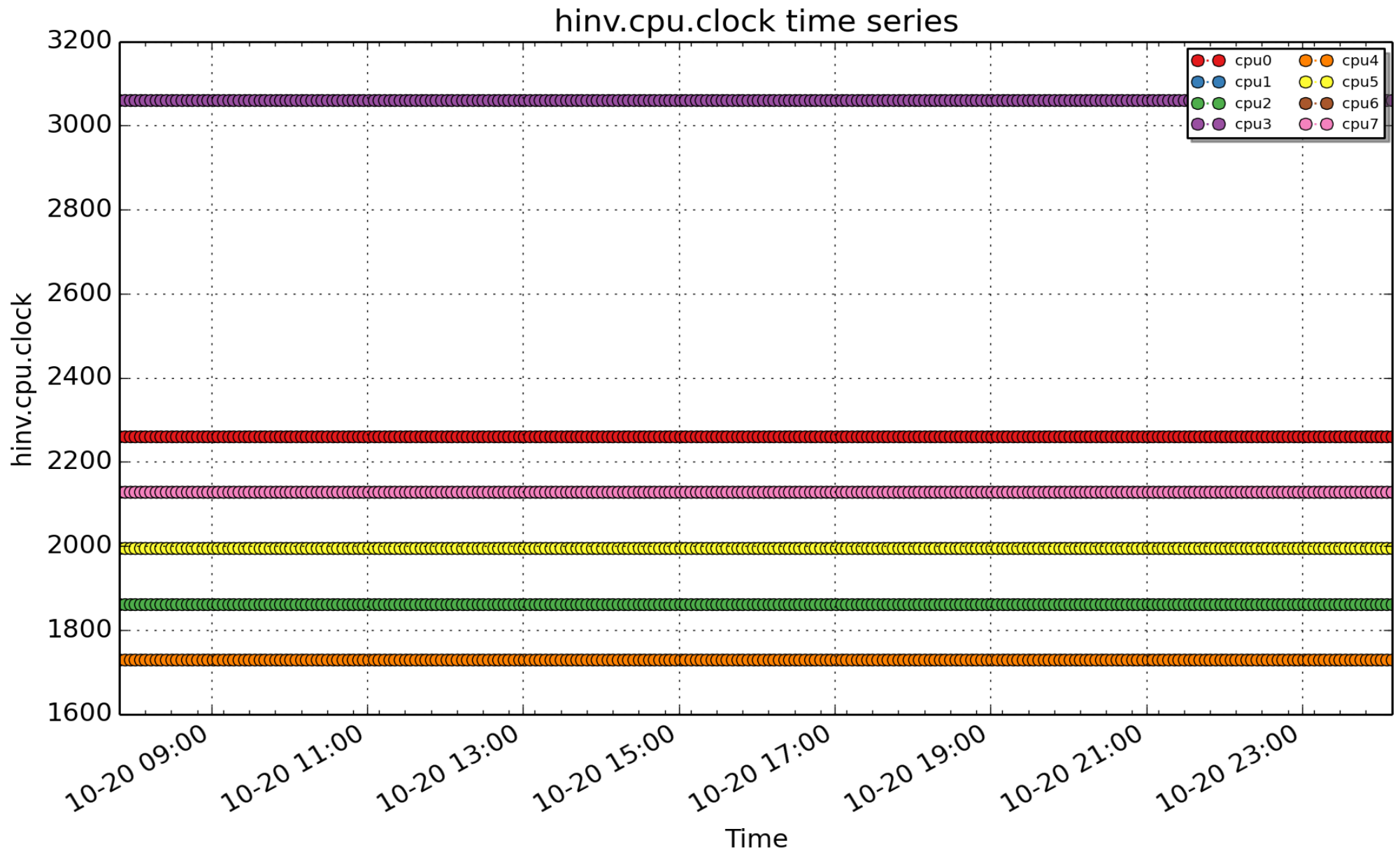


hinv.cpu.cache: primary cache size of each CPU as reported by /proc/cpuinfo (- U32)

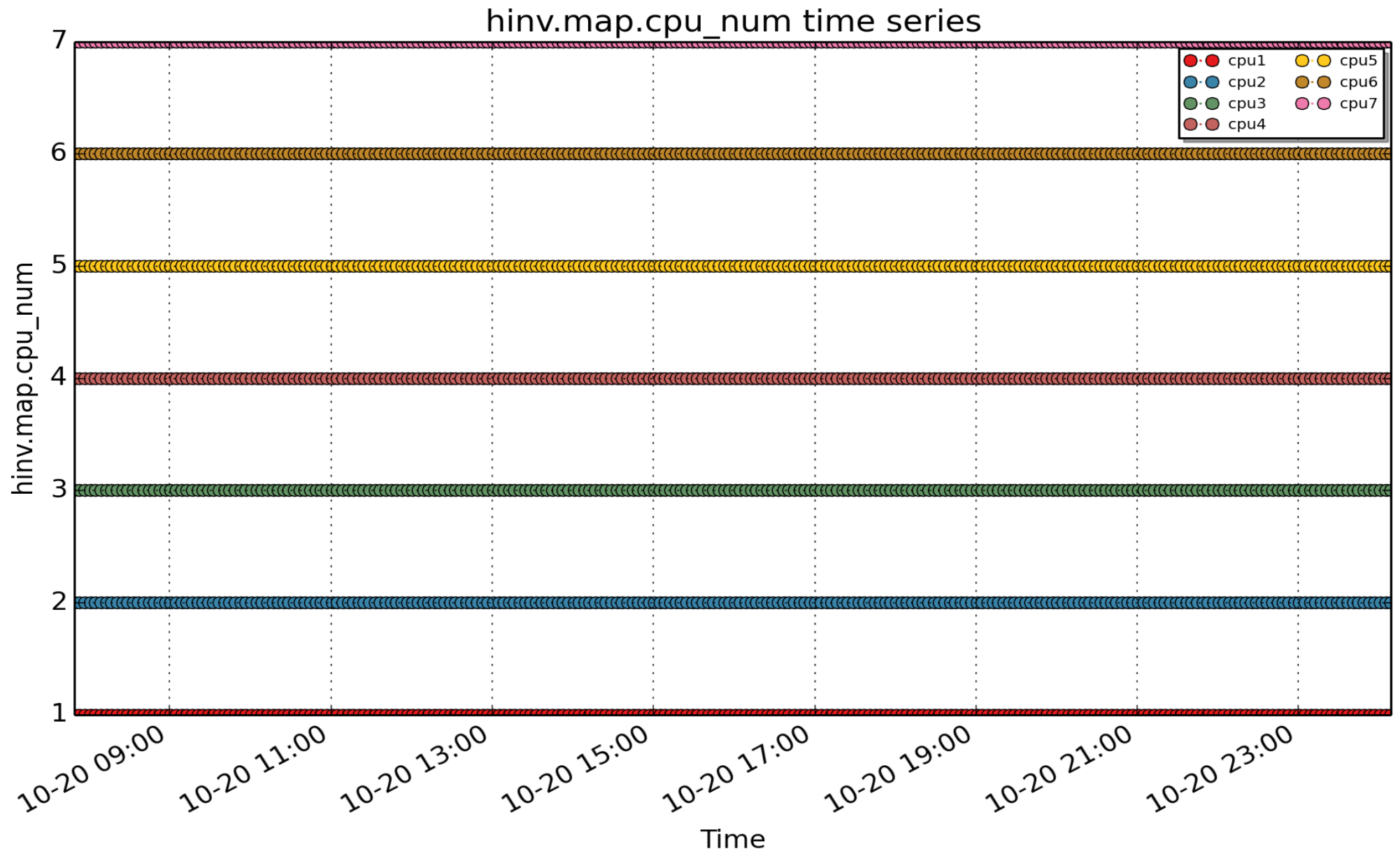
hinv.cpu.cache_alignment time series



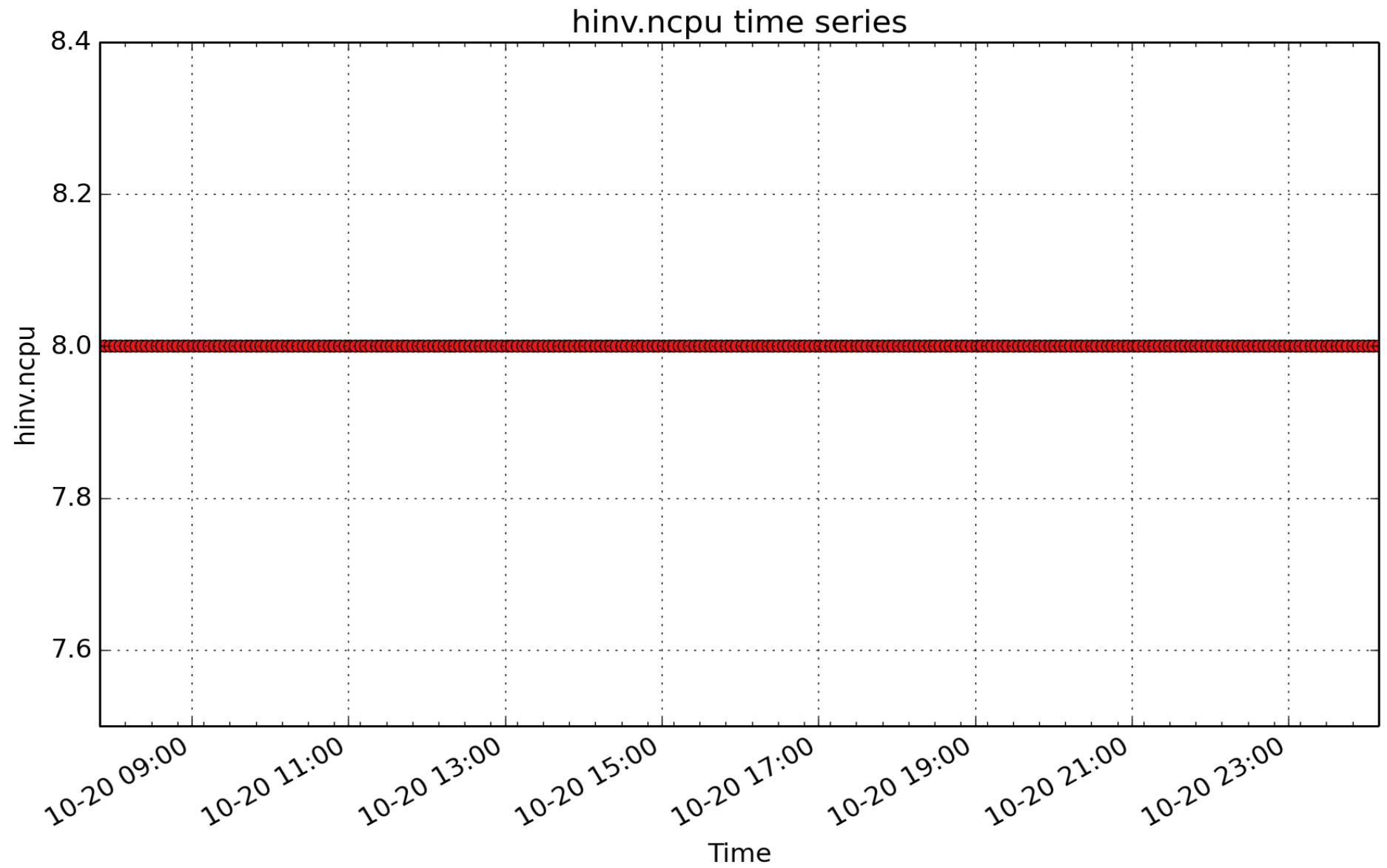
hinv.cpu.cache_alignment: ... (- U32)



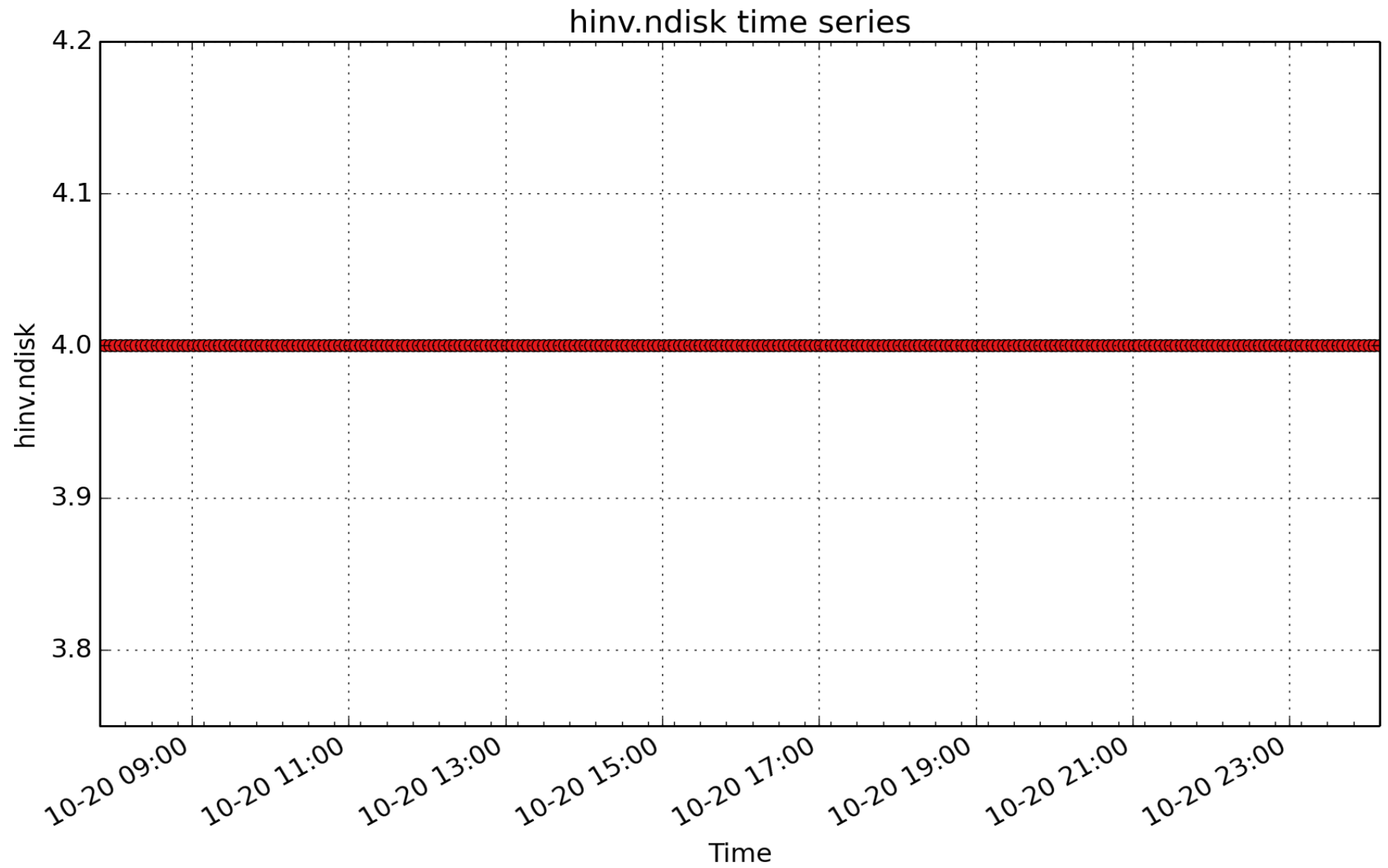
hinv.cpu.clock: clock rate in Mhz for each CPU as reported by /proc/cpuinfo (count - FLOAT)



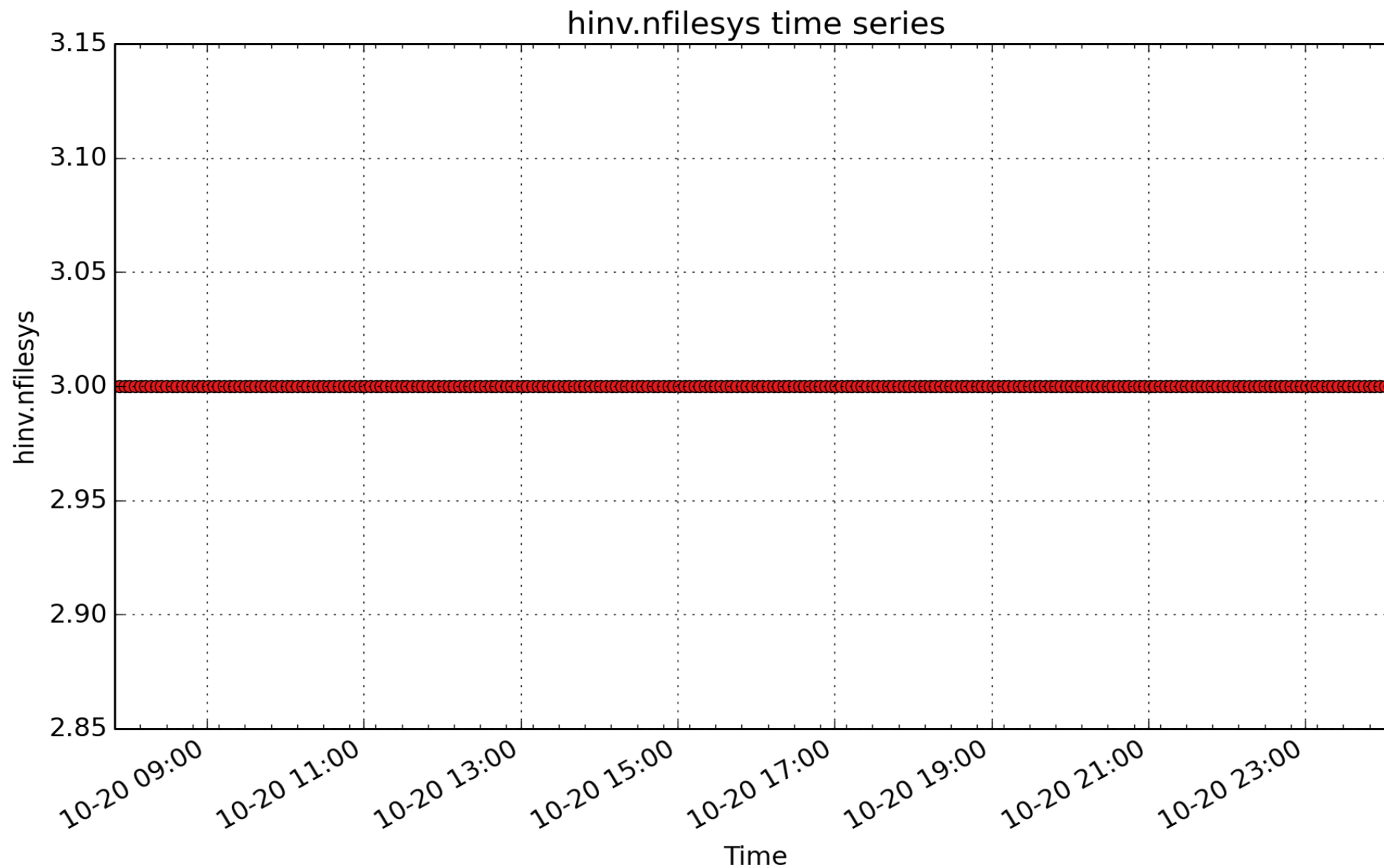
hinv.map.cpu_num: logical to physical CPU mapping for each CPU (- U32)



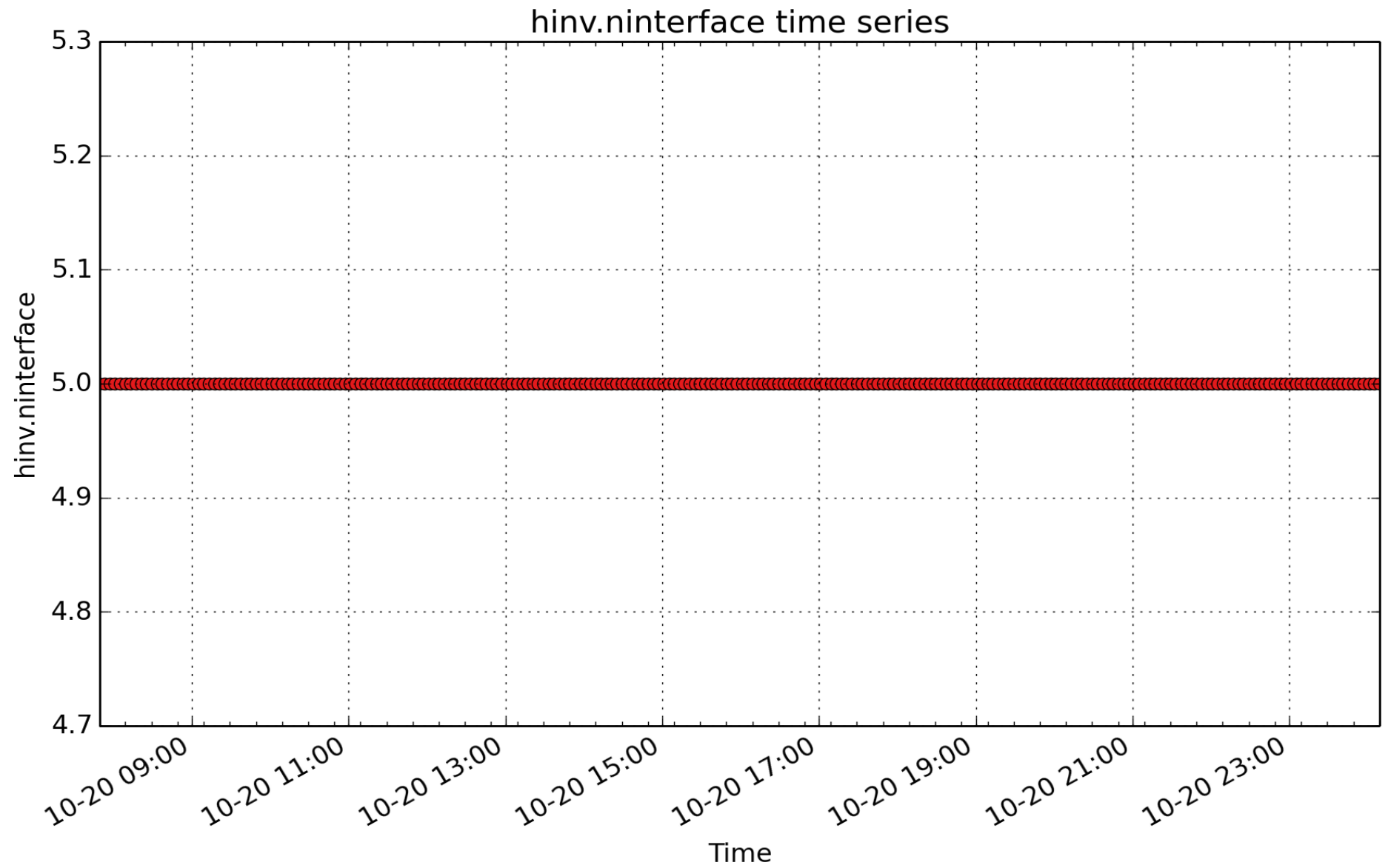
hinv.ncpu: number of CPUs in the system (- U32)



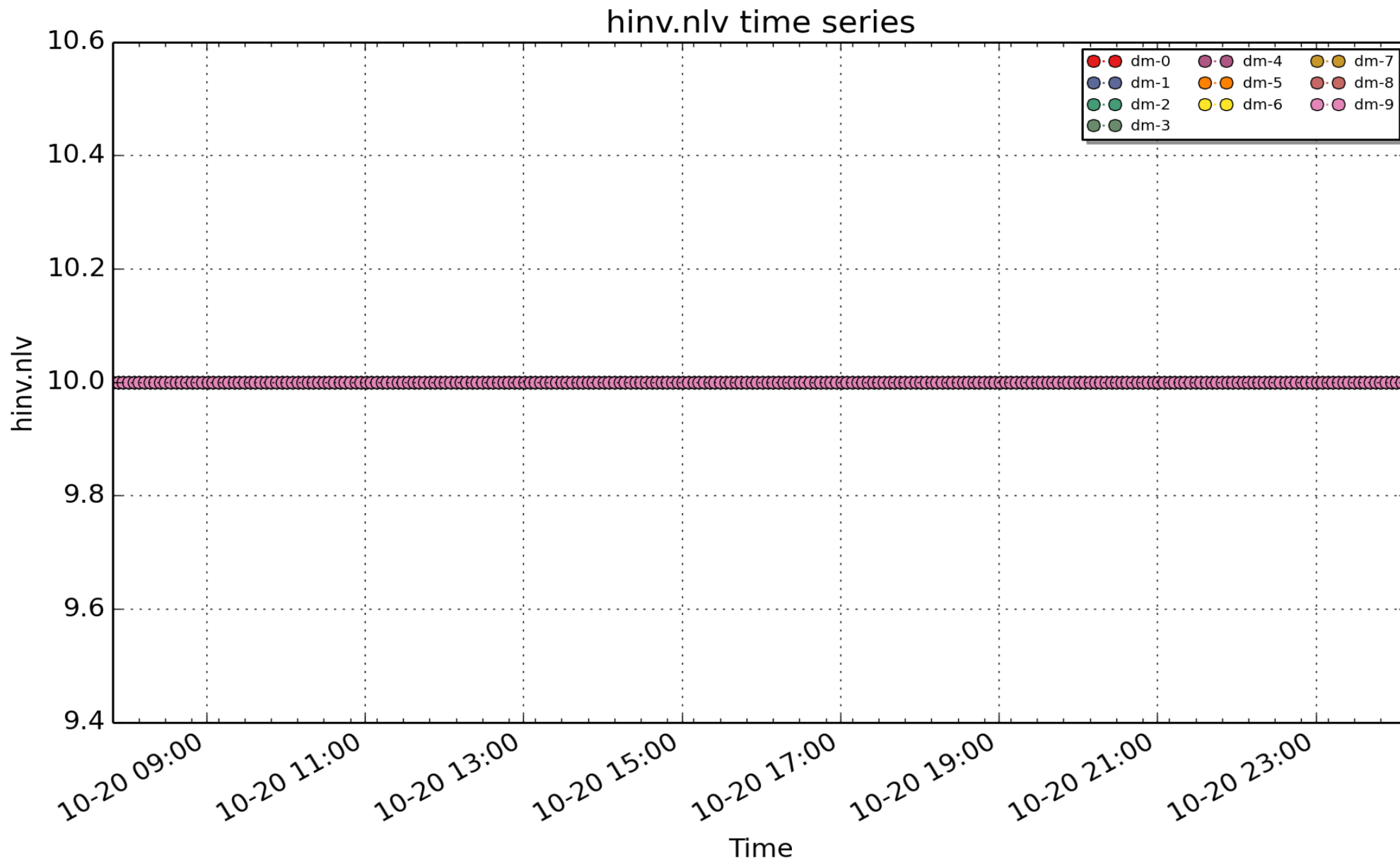
hinv.ndisk: number of disks in the system (- U32)



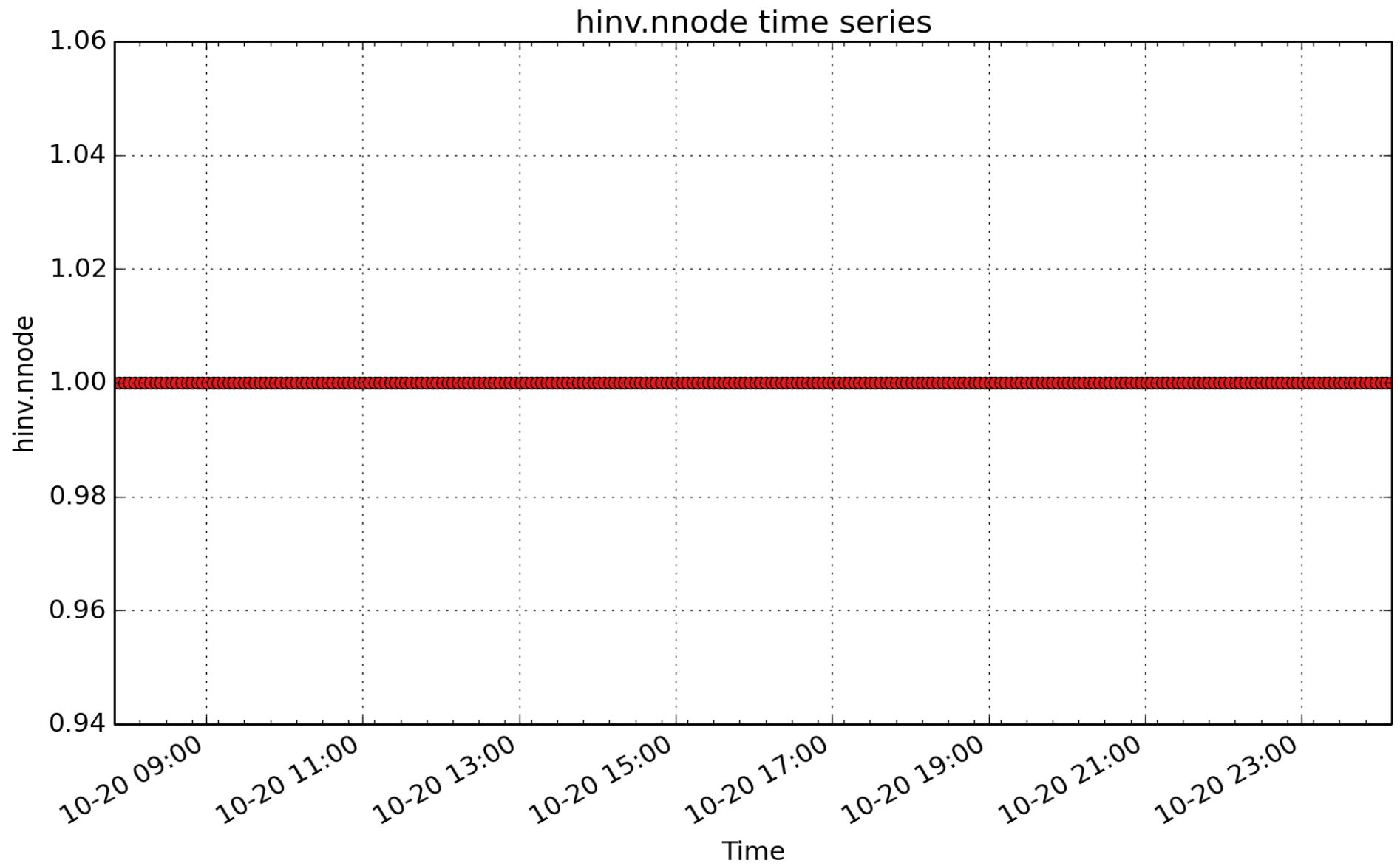
hinv.nfilesys: number of (local) file systems currently mounted (count - U32)



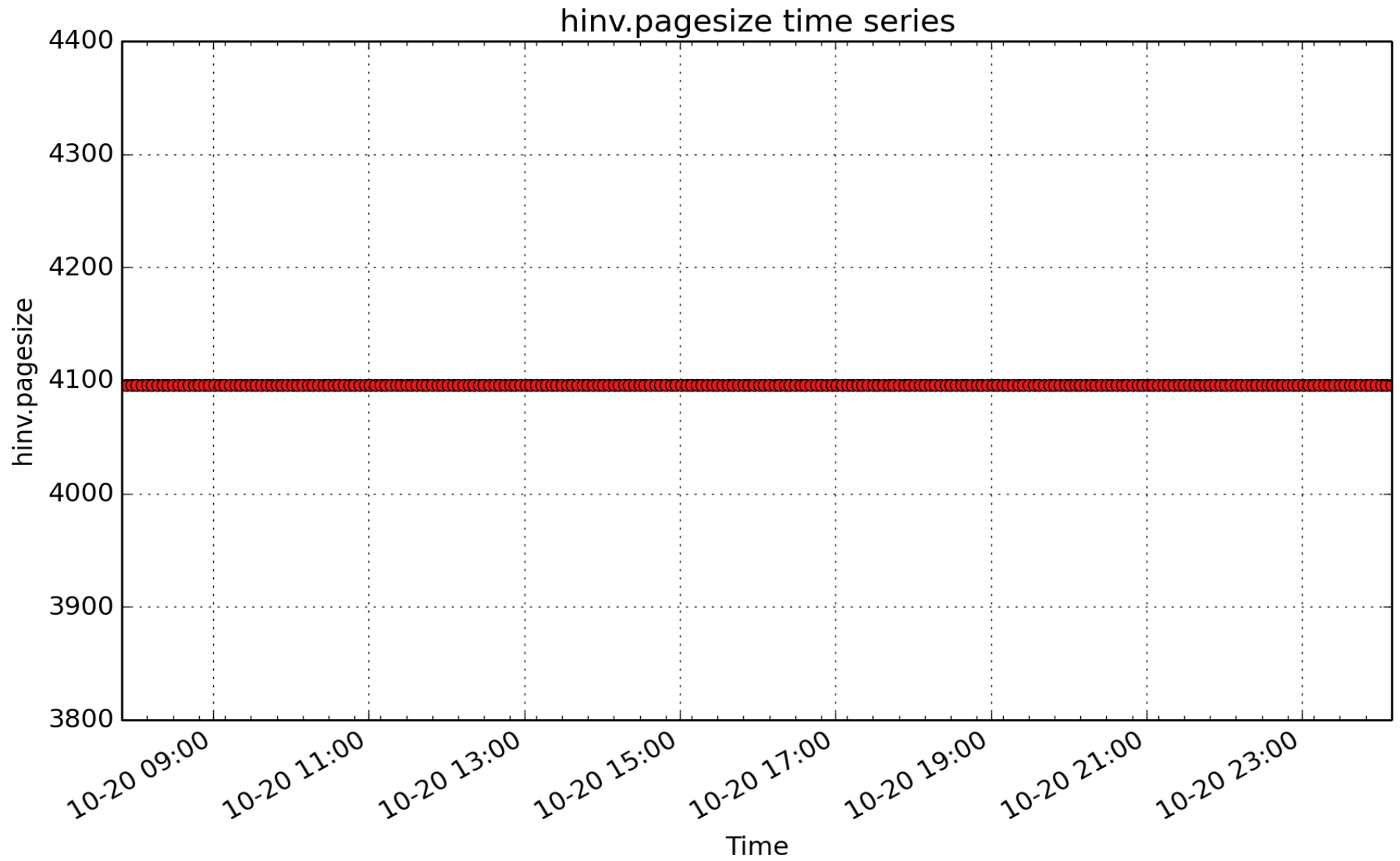
hinv.ninterface: number of active (up) network interfaces (- U32)



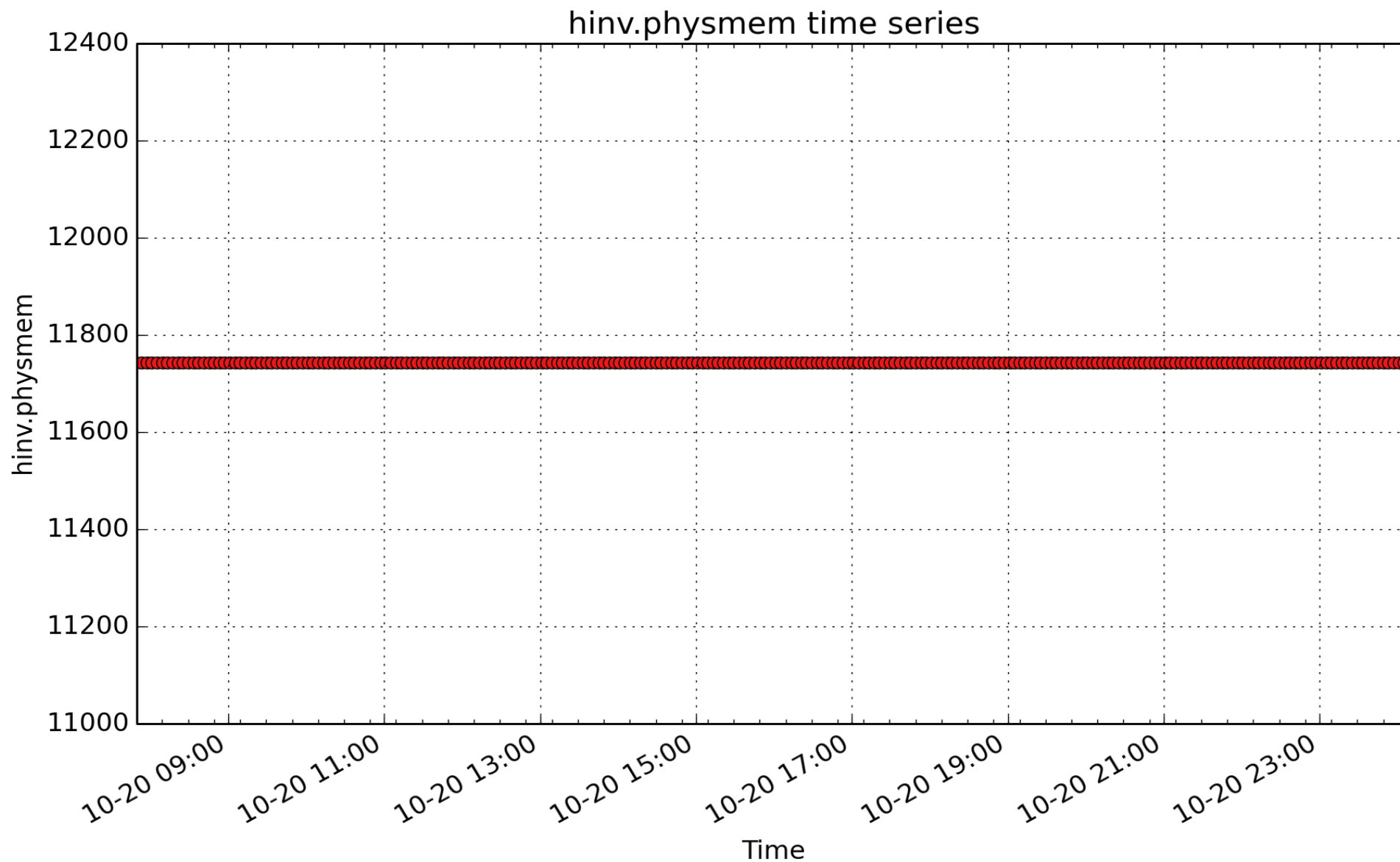
hinv.nlv: number of logical volumes (- U32)



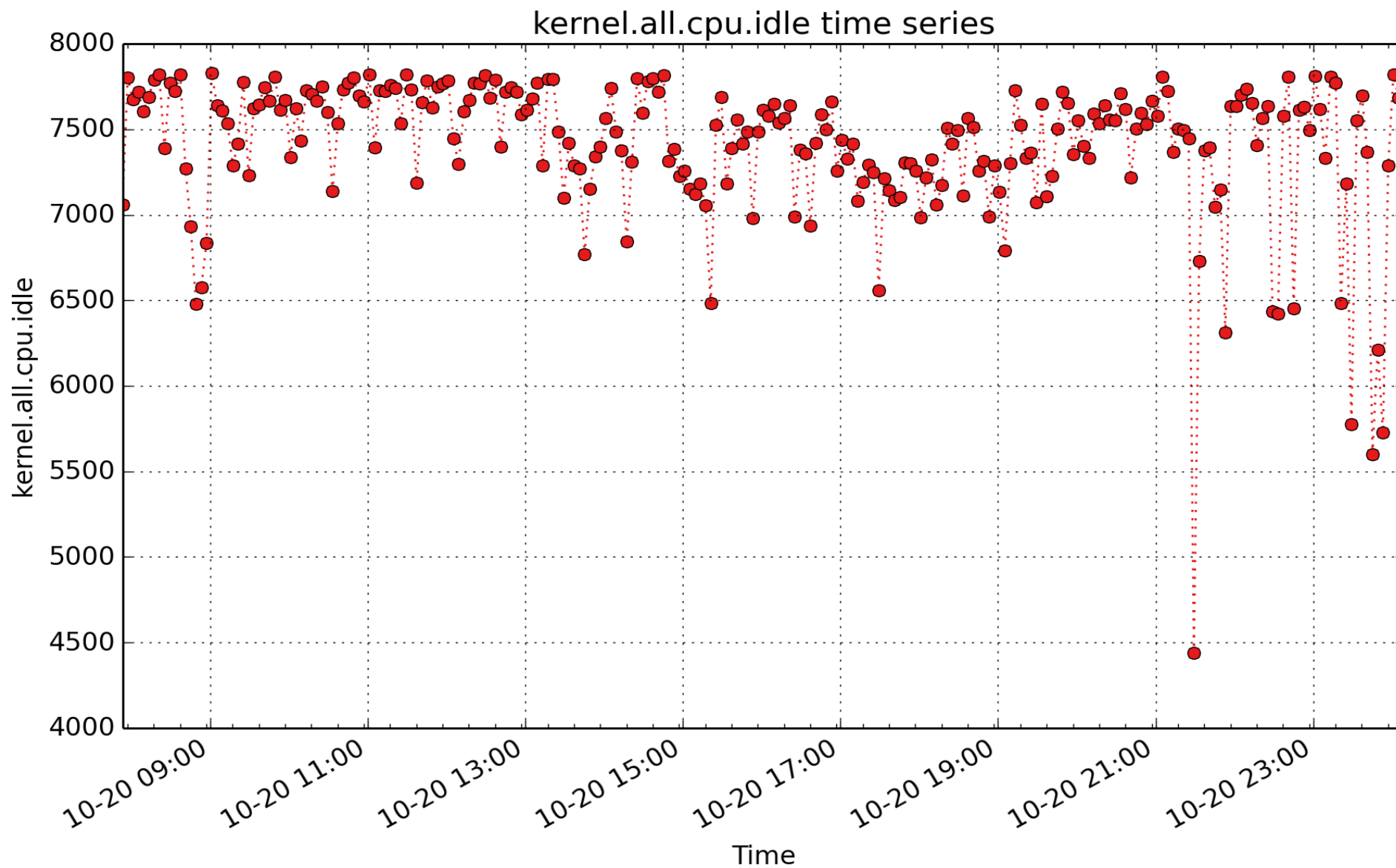
hinv.nnode: number of NUMA nodes in the system (- U32)



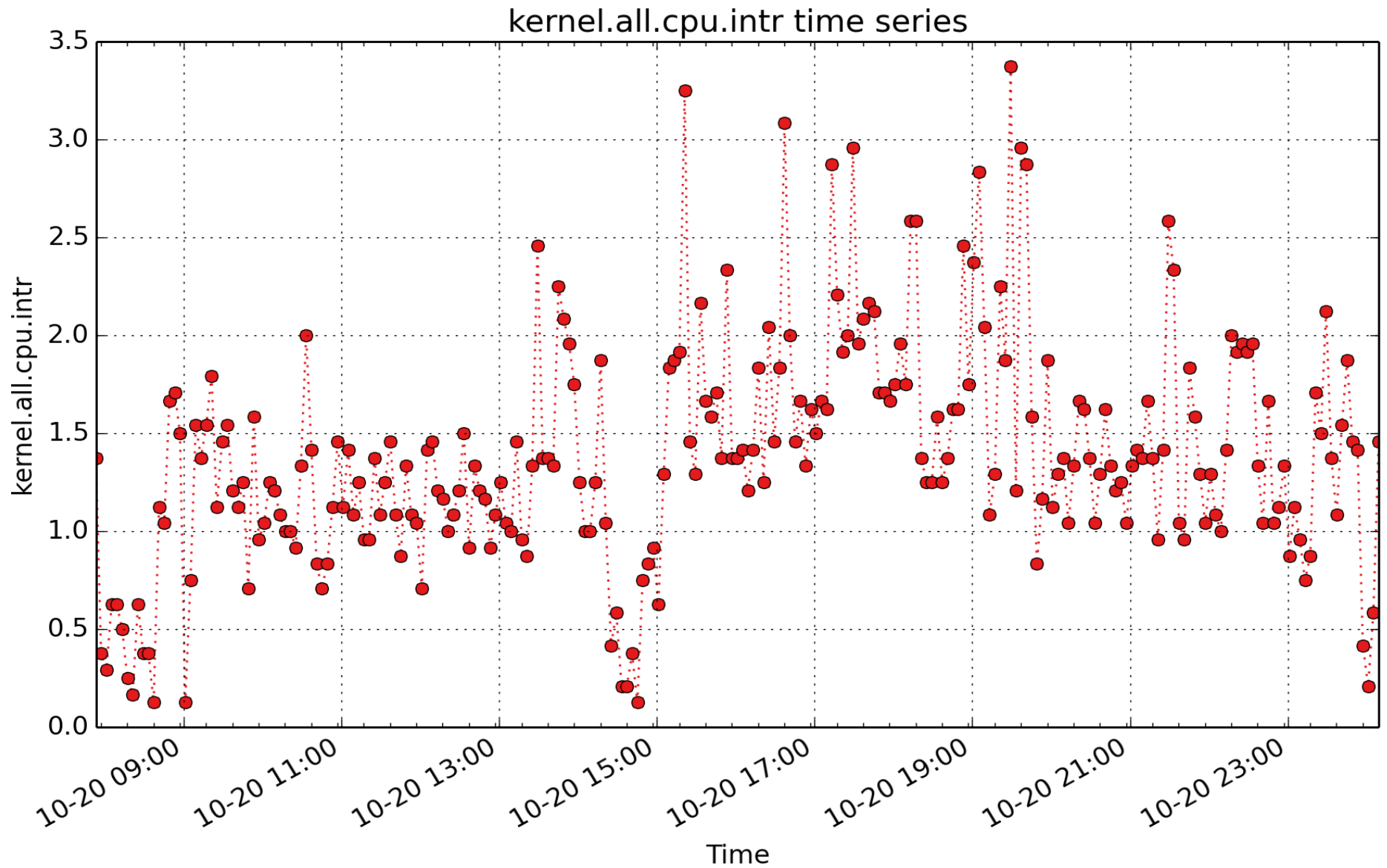
hinv.pagesize: The memory page size of the running kernel in bytes. (byte - U32)



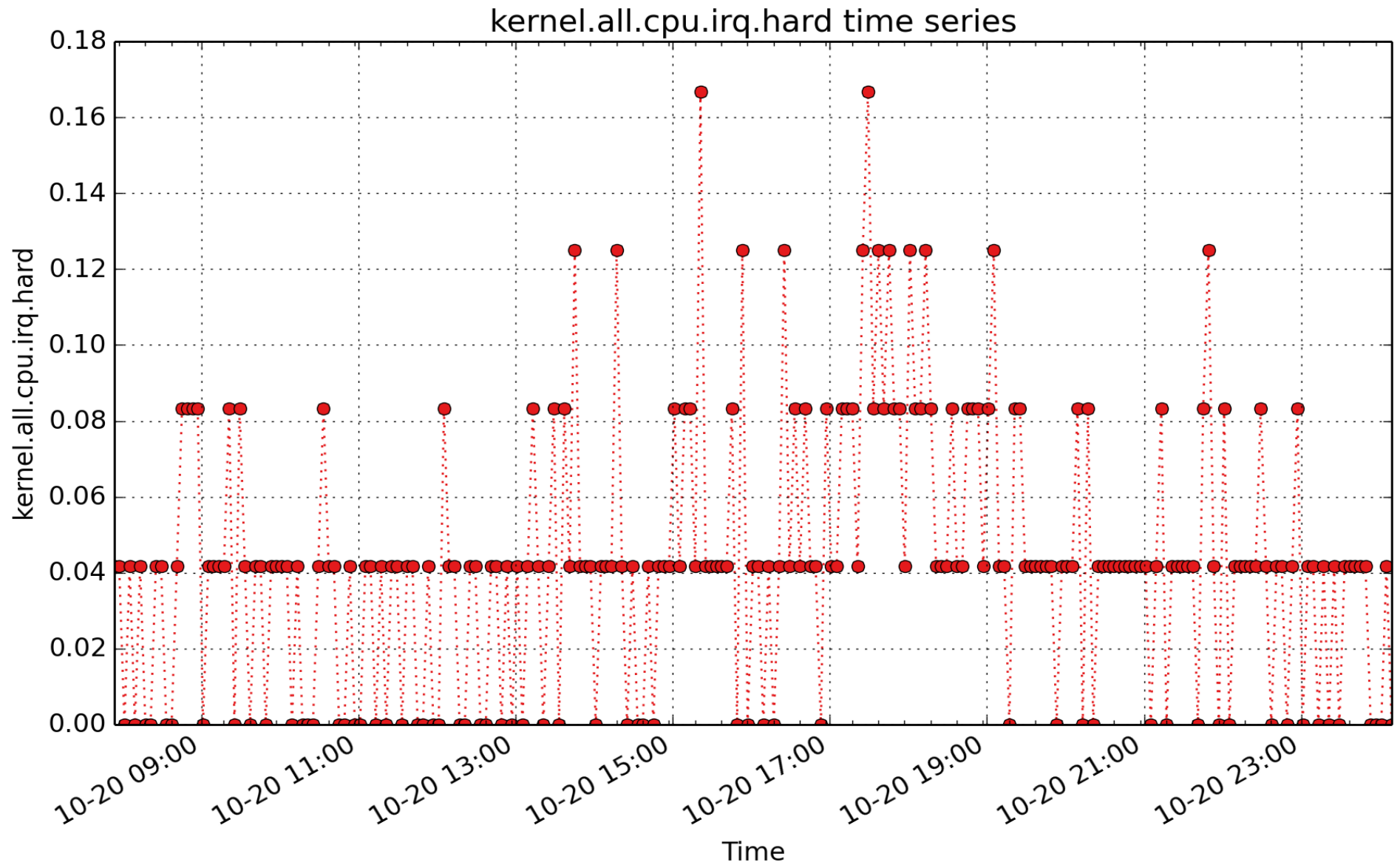
hinv.physmem: total system memory metric from /proc/meminfo (Mbyte - U32)



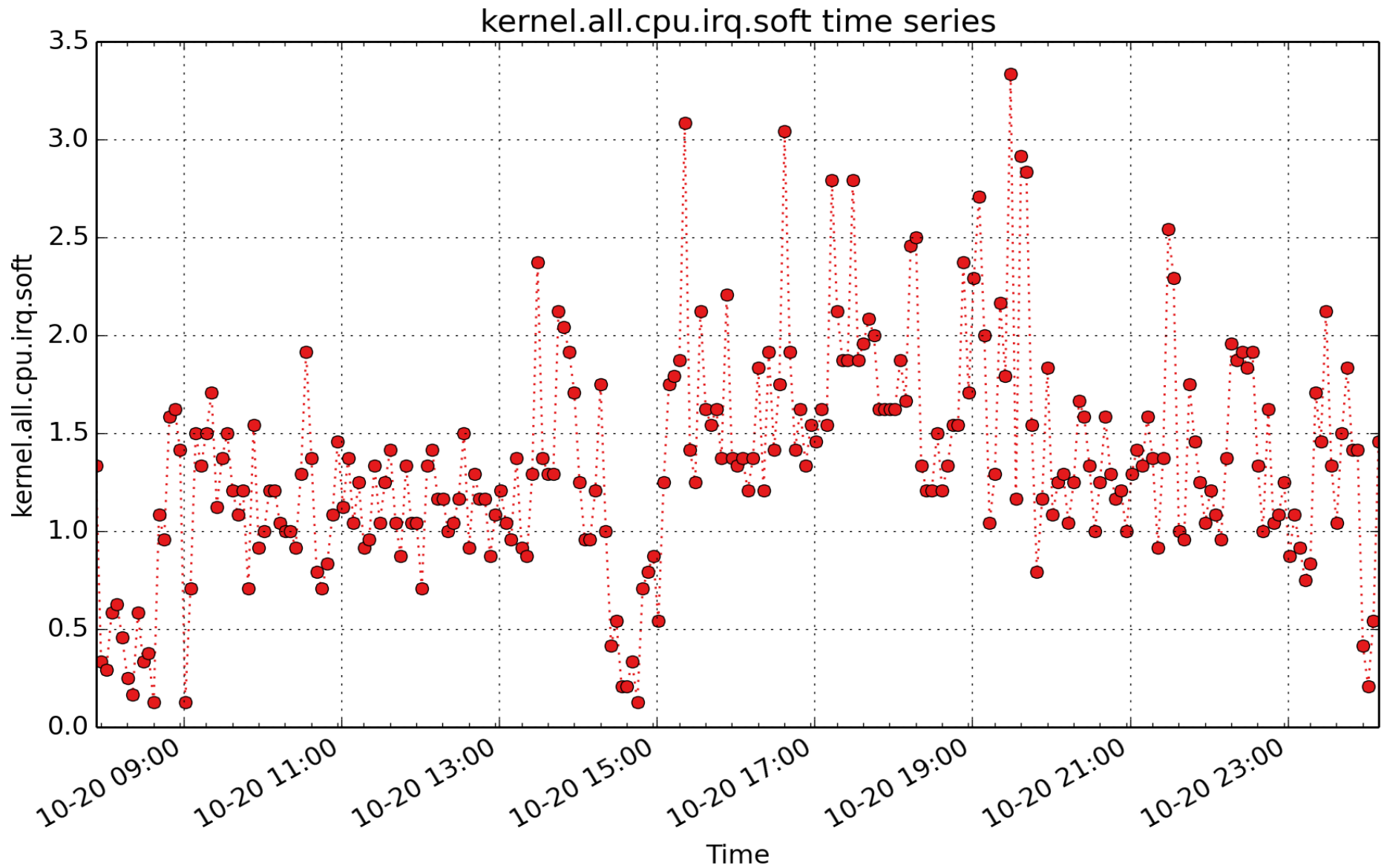
kernel.all.cpu.idle: total idle CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



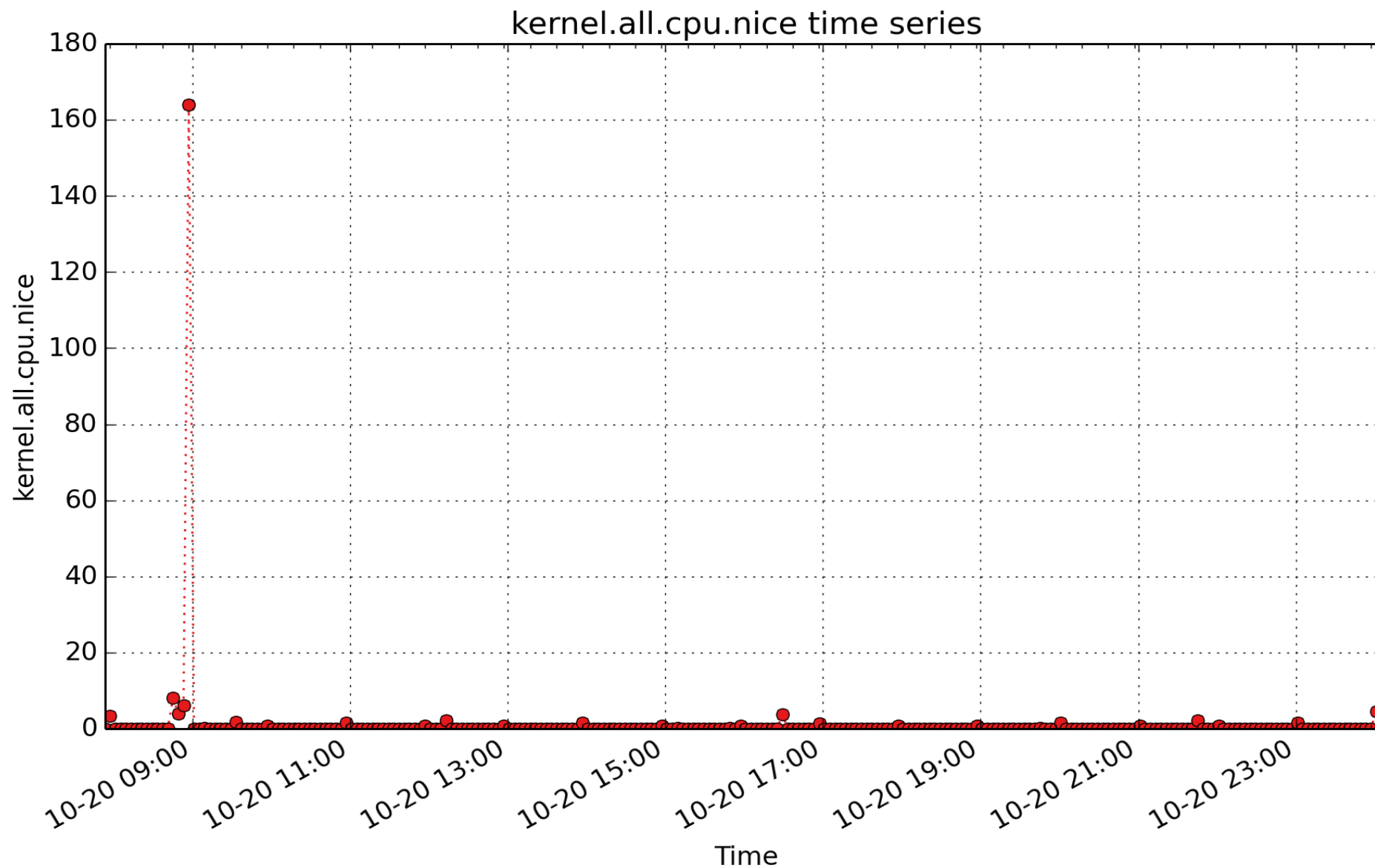
kernel.all.cpu.intr: Total time spent processing interrupts on all CPUs. This value includes both soft and hard interrupt processing time. (millisec - U64) - *rate converted*



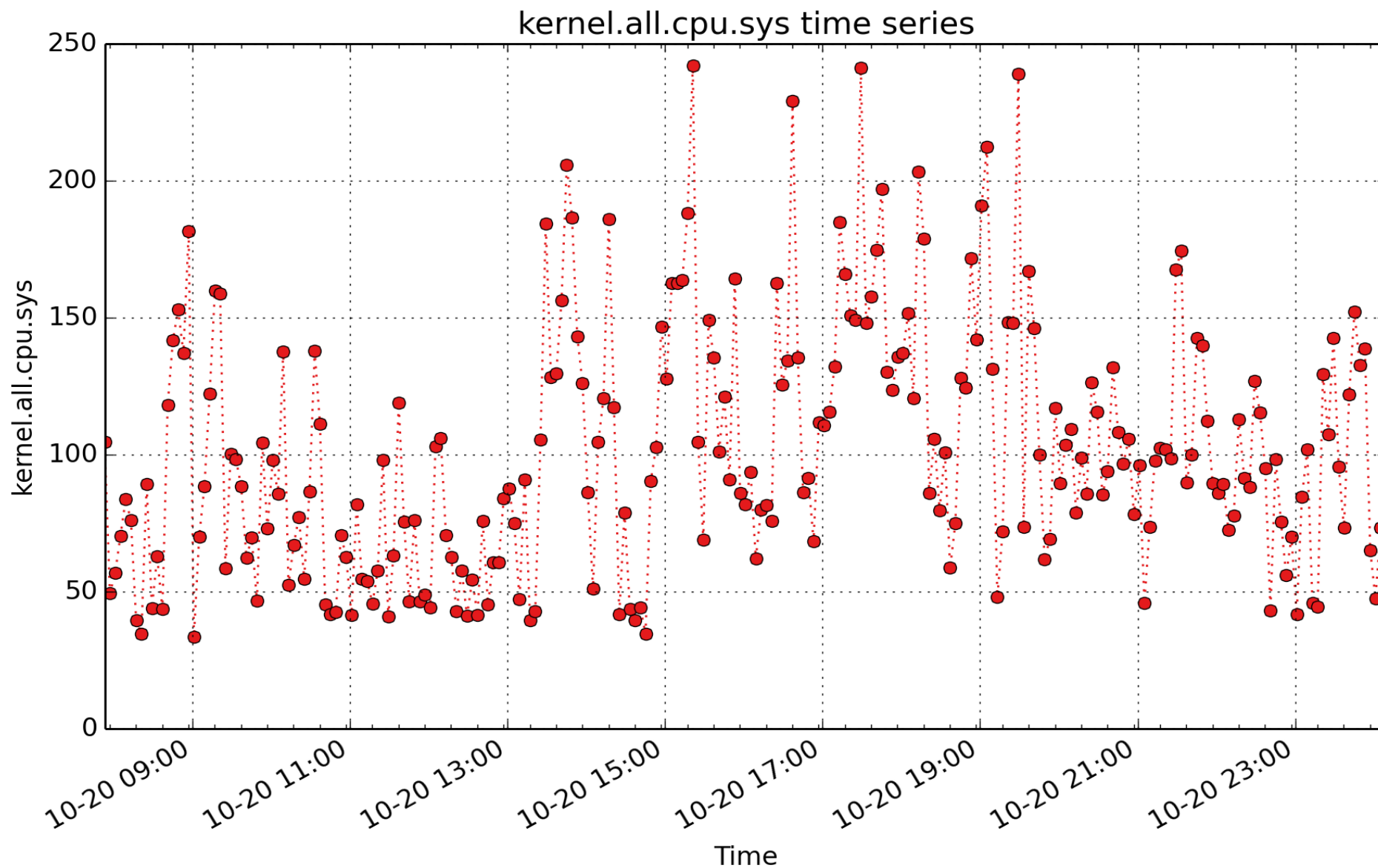
kernel.all.cpu.irq.hard: Total hard interrupt CPU time ("hard" interrupt handling code is the code run directly on receipt of the initial hardware interrupt, and does not include "soft" interrupt handling code which is deferred until later). (millisec - U64) - *rate converted*



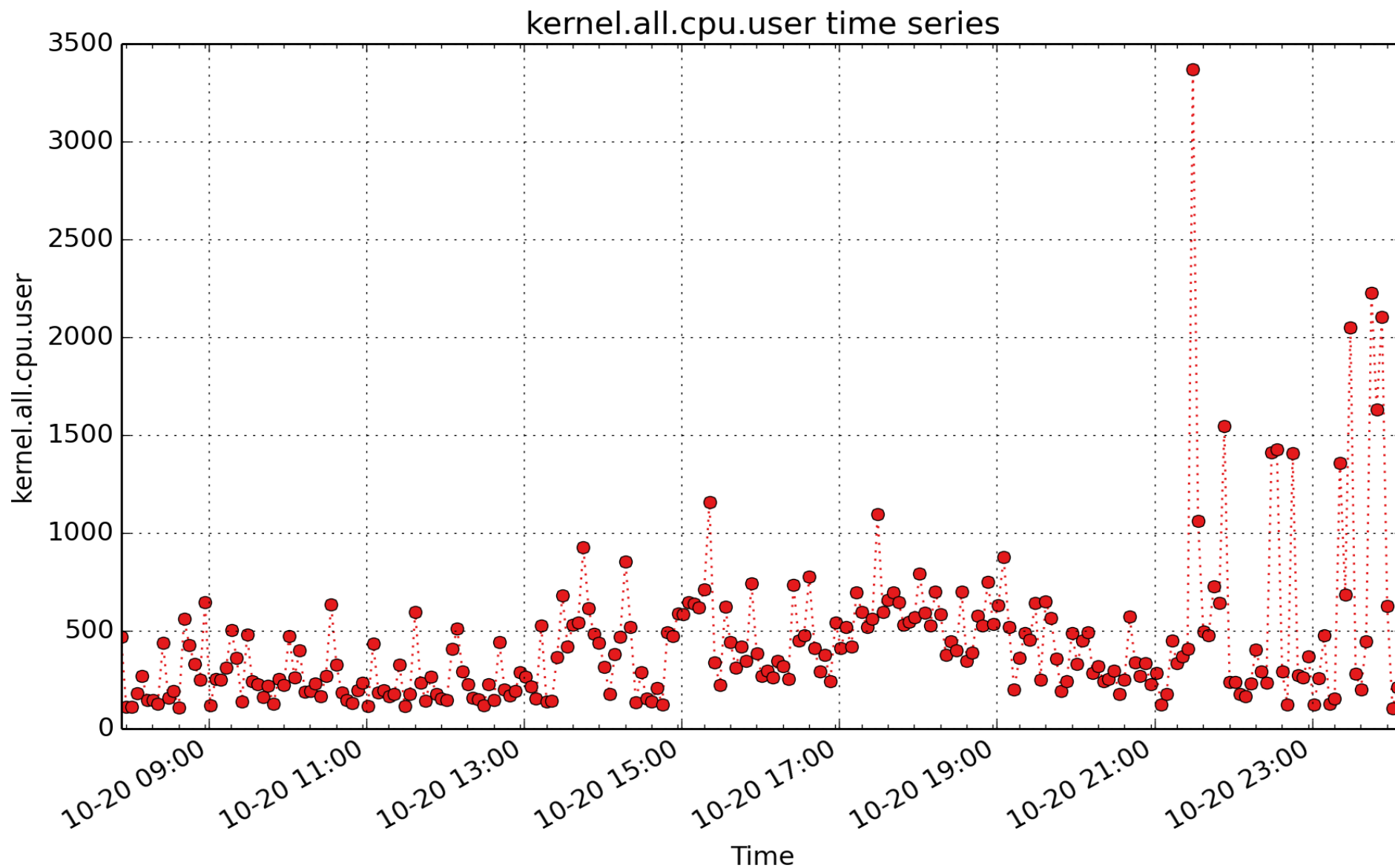
kernel.all.cpu irq.soft: Total soft interrupt CPU time (deferred interrupt handling code, not run in the initial interrupt handler). (millisec - U64) - *rate converted*



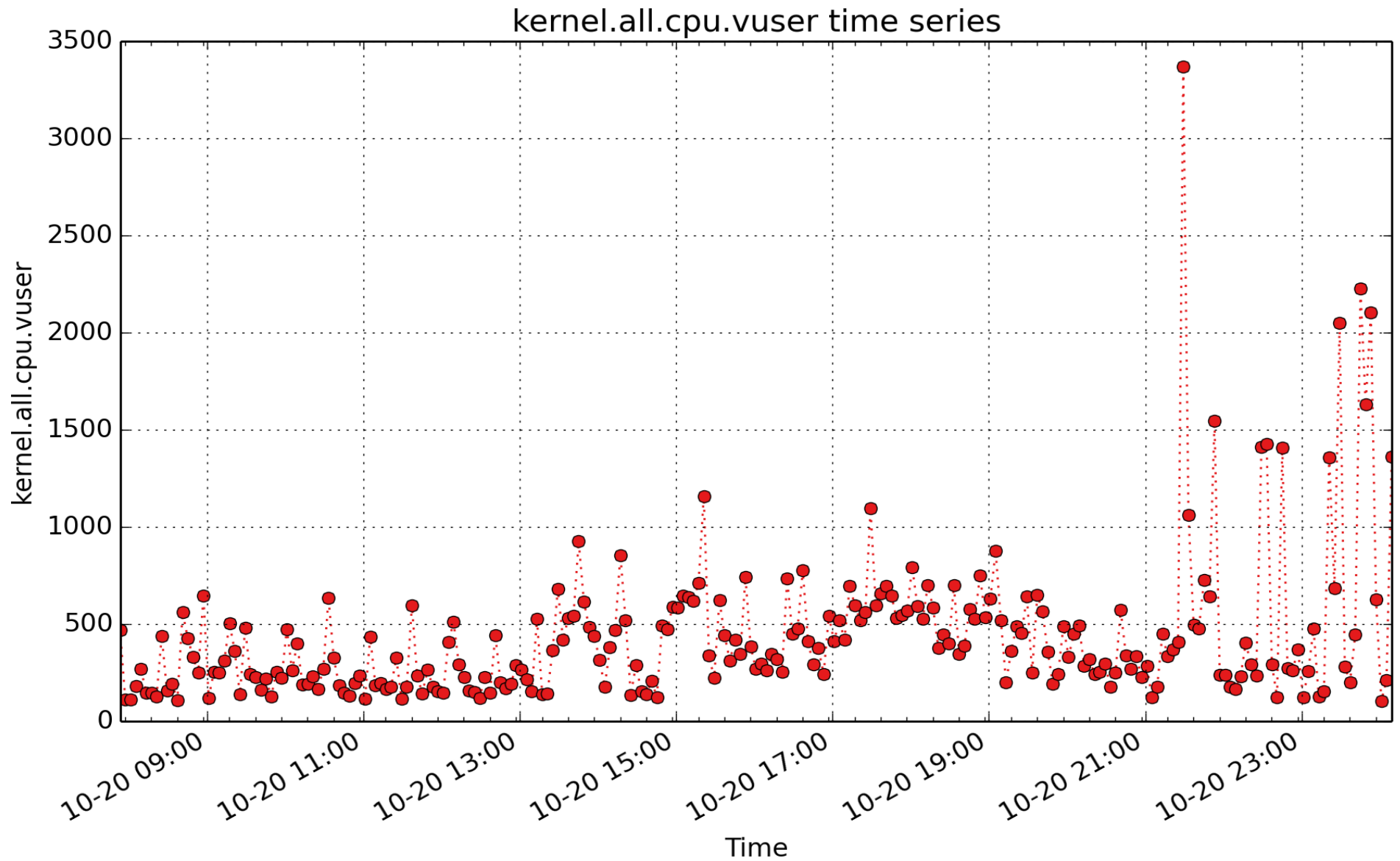
kernel.all.cpu.nice: total nice user CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



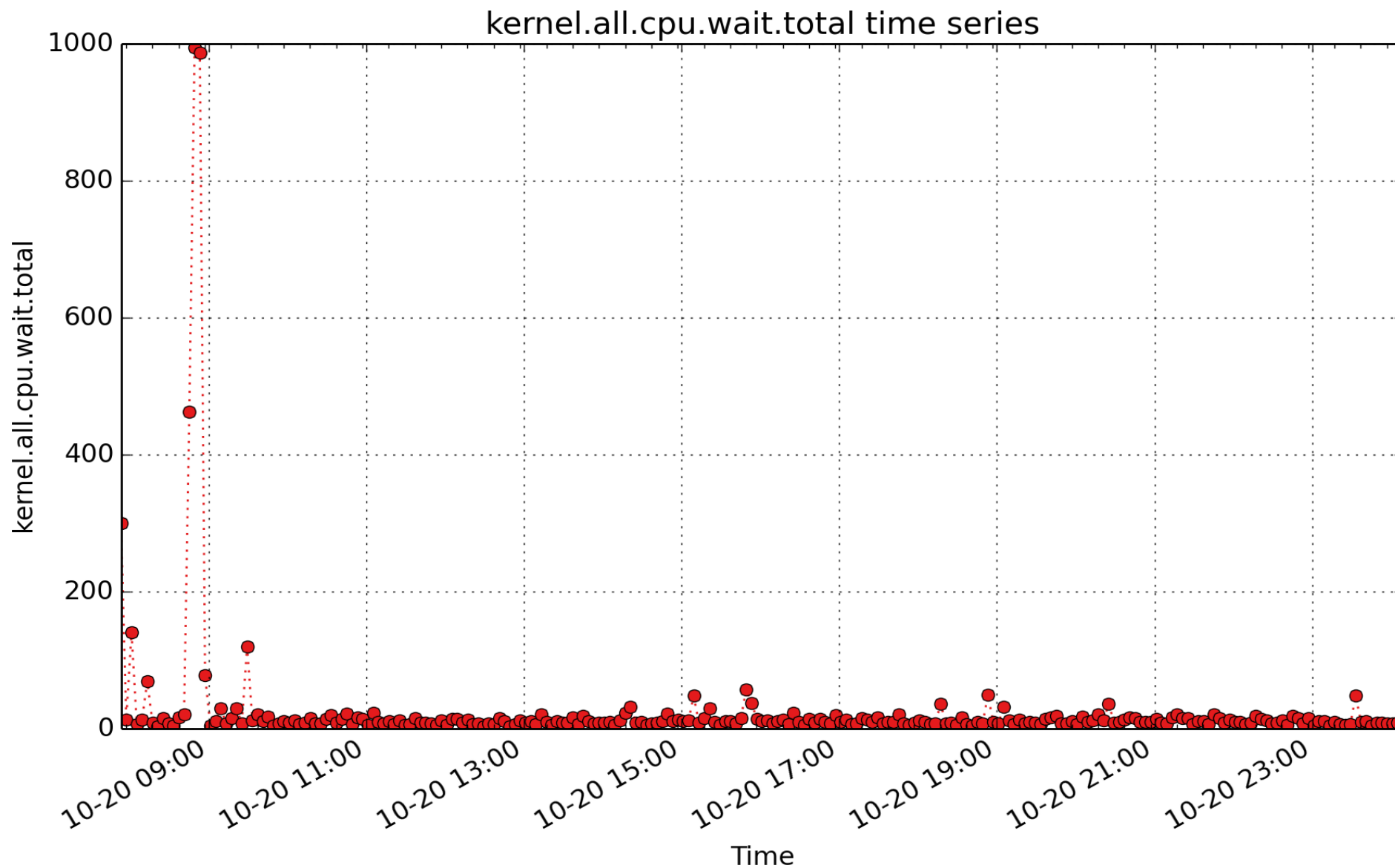
kernel.all.cpu.sys: total sys CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



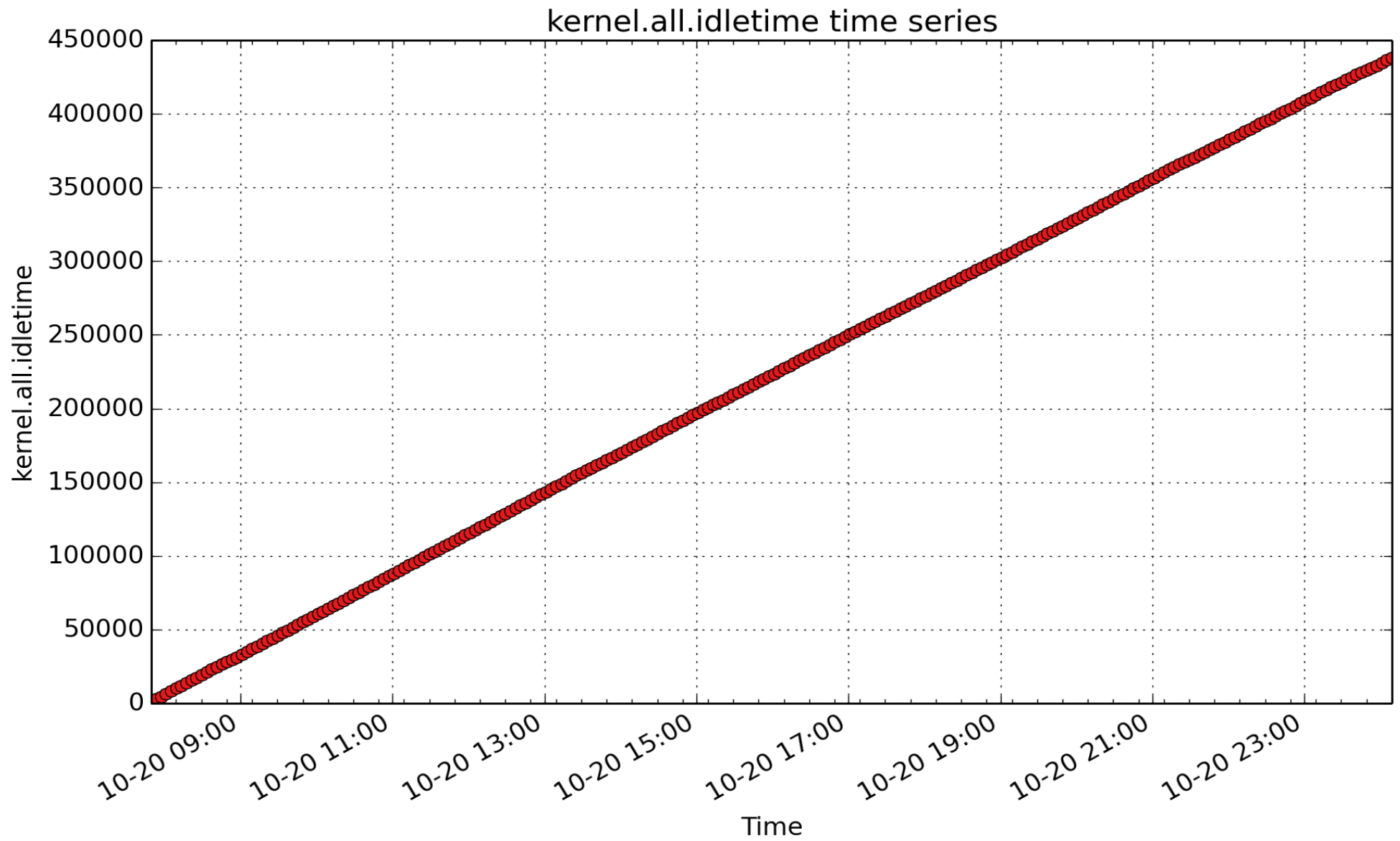
kernel.all.cpu.user: total user CPU time from /proc/stat for all CPUs, including guest CPU time (millisec - U64) - *rate converted*



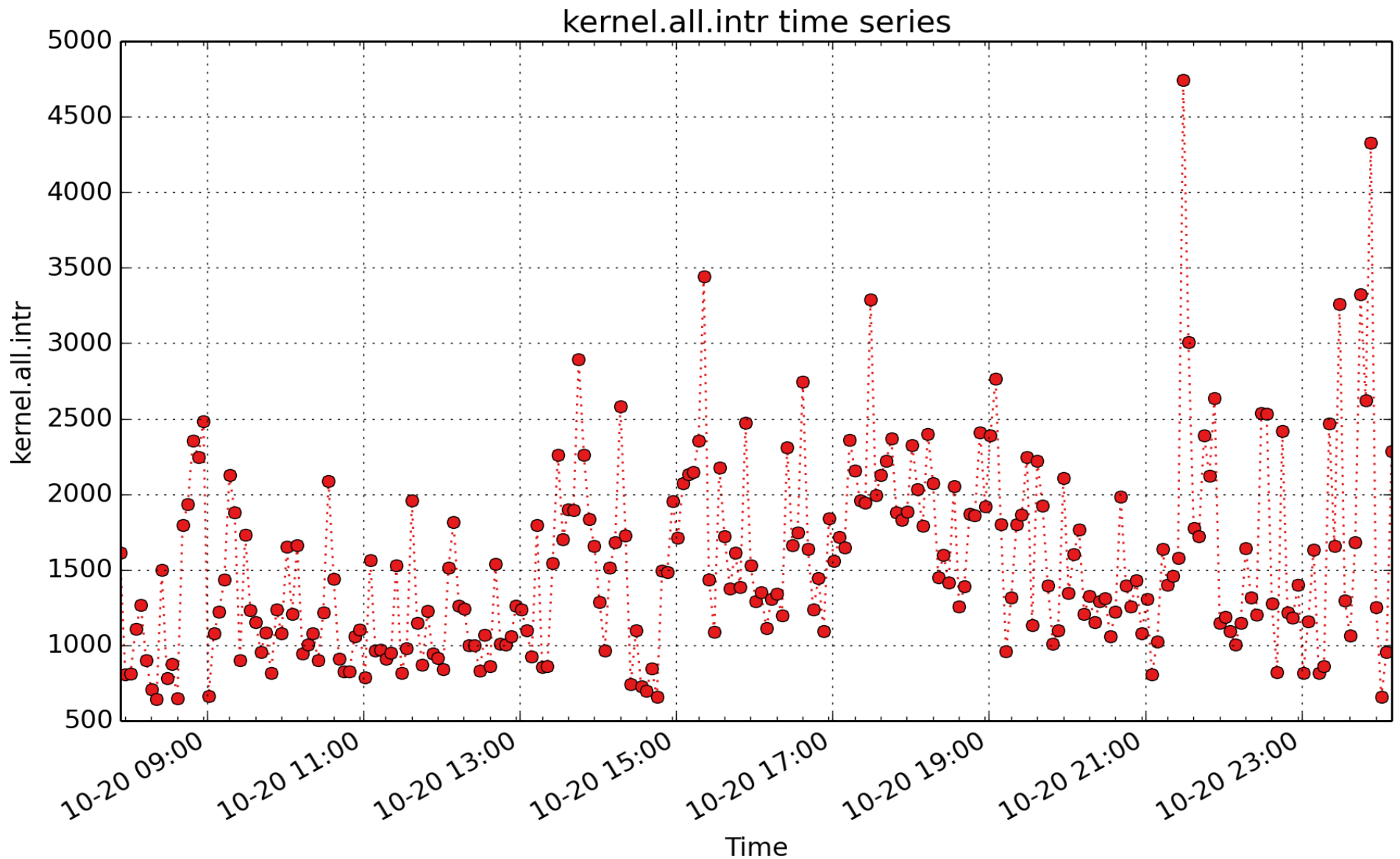
kernel.all.cpu.vuser: total user CPU time from /proc/stat for all CPUs, excluding guest CPU time (millisec - U64) - *rate converted*



kernel.all.cpu.wait.total: total wait CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



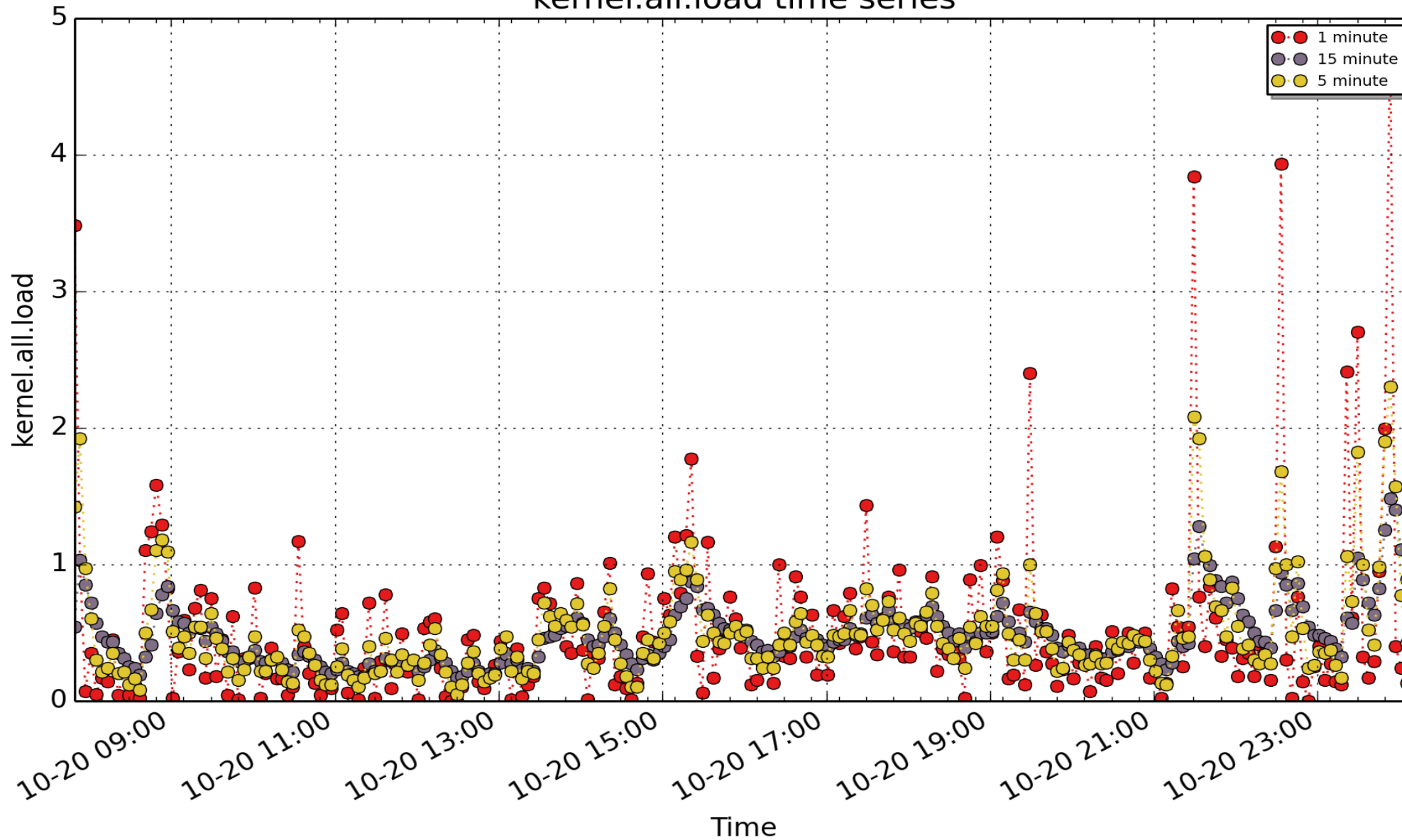
kernel.all.idletime: time the current kernel has been idle since boot (sec - U32)



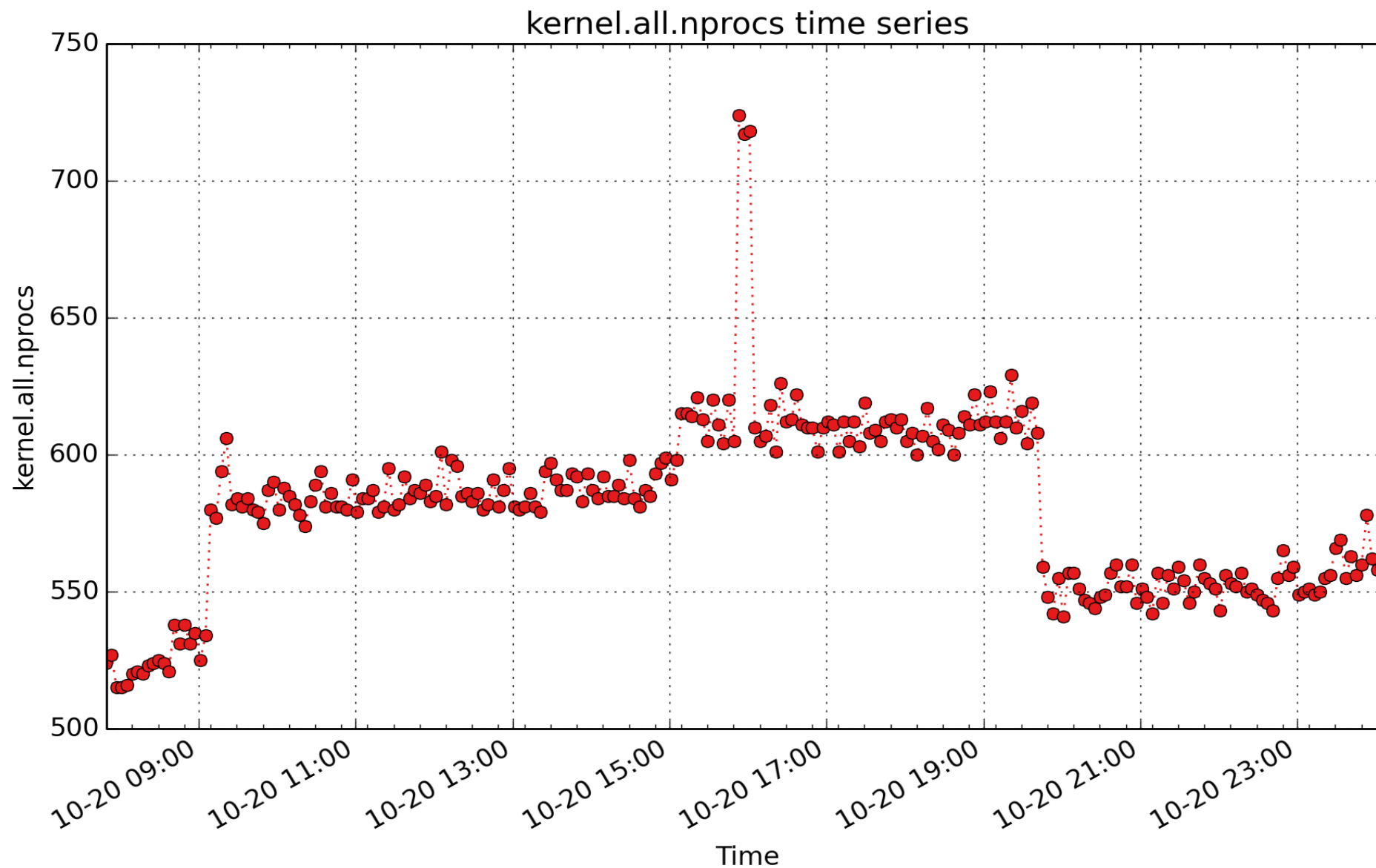
kernel.all.intr: The value is the first value from the intr field in /proc/stat, which is a counter of the total number of interrupts processed. The value is normally converted to a rate (count/second). This counter usually increases by at least HZ/second, i.e. the clock interrupt rate, which is usually 100/second. See also kernel.percpu.interrupts to get a breakdown of interrupt rates by interrupt type and which CPU processed

each one. (count - U64) - *rate converted*

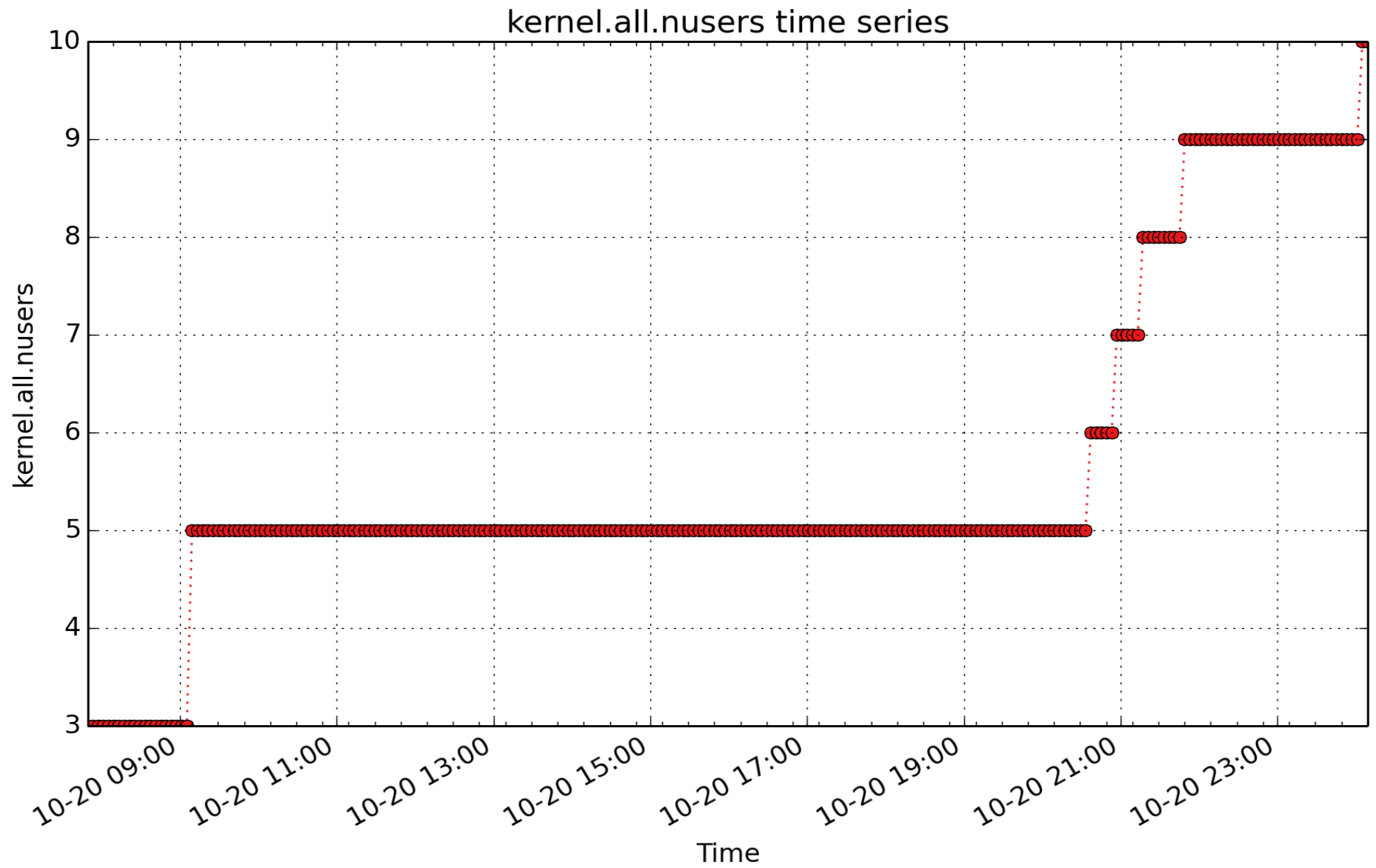
kernel.all.load time series



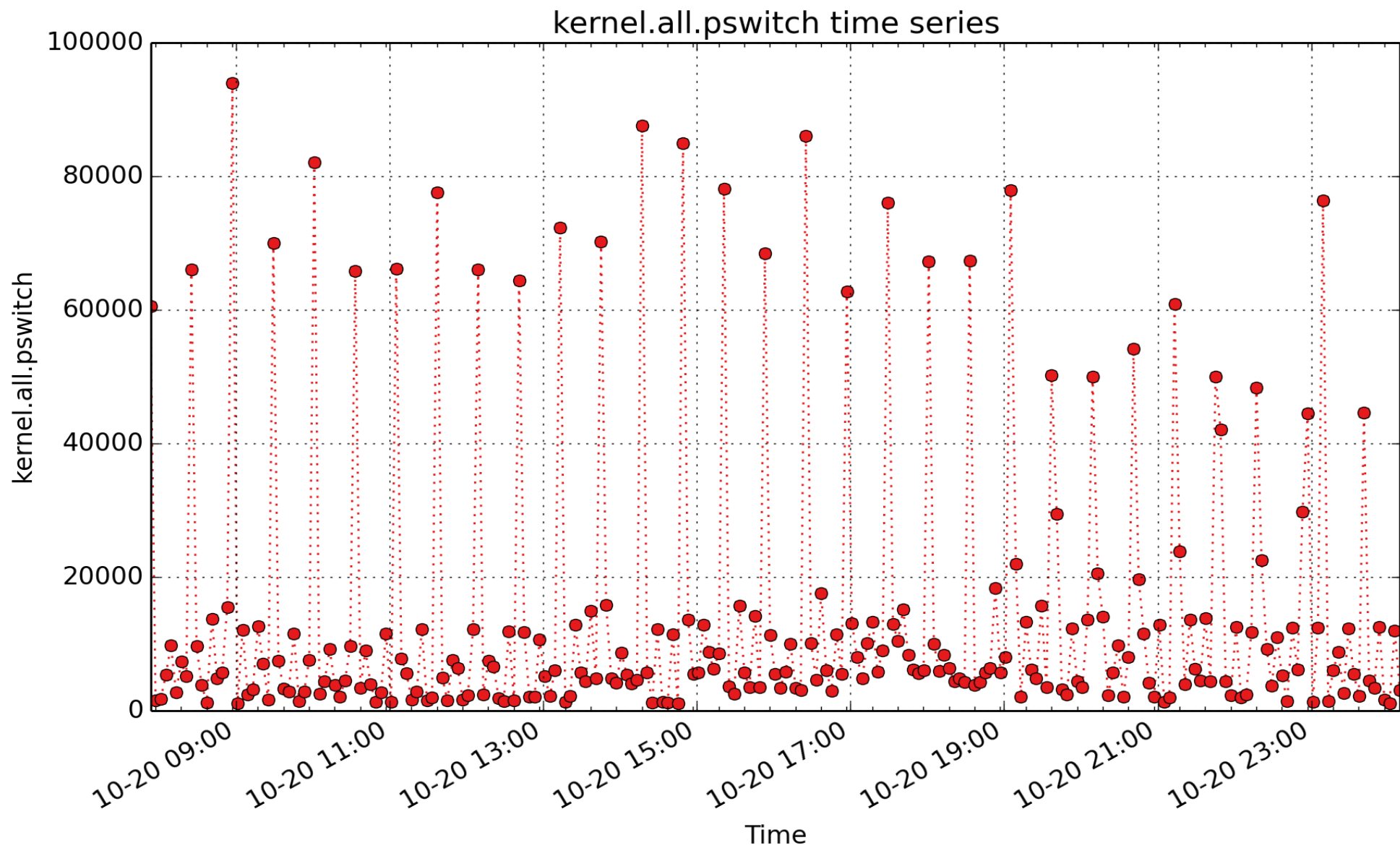
kernel.all.load: 1, 5 and 15 minute load average (- FLOAT)



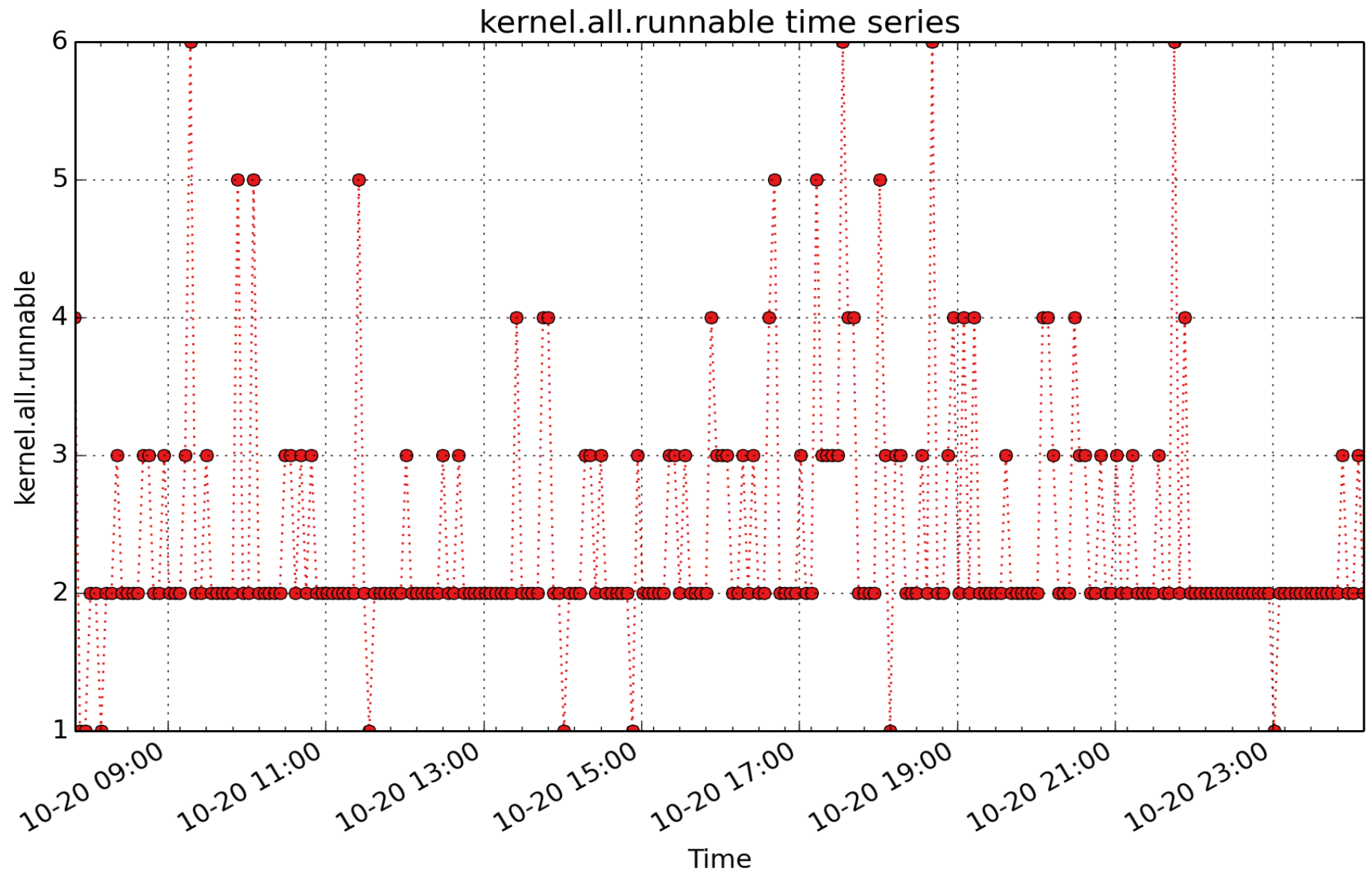
kernel.all.nprocs: total number of processes (lightweight) (- U32)



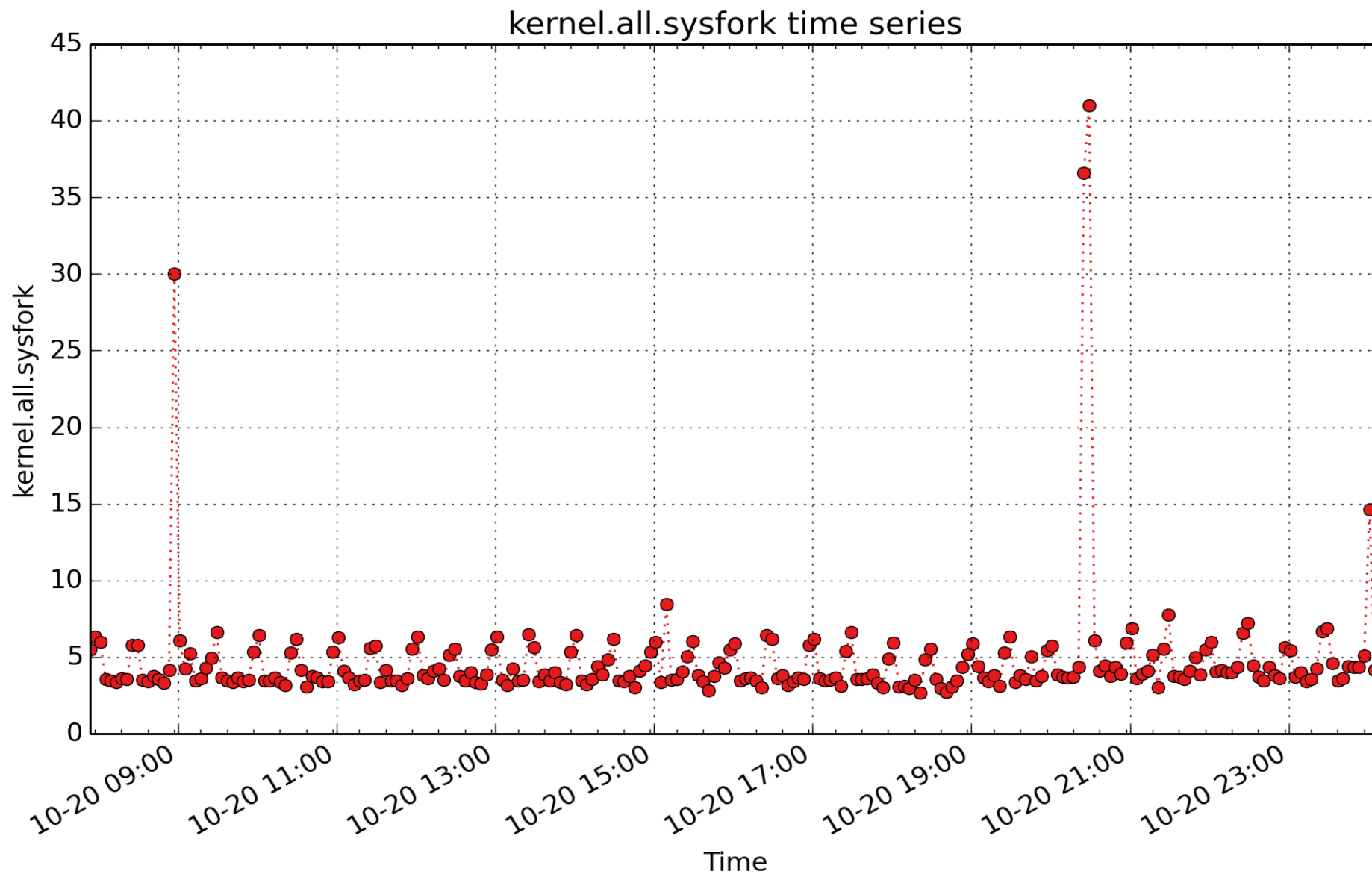
kernel.all.nusers: number of user sessions on system (- U32)



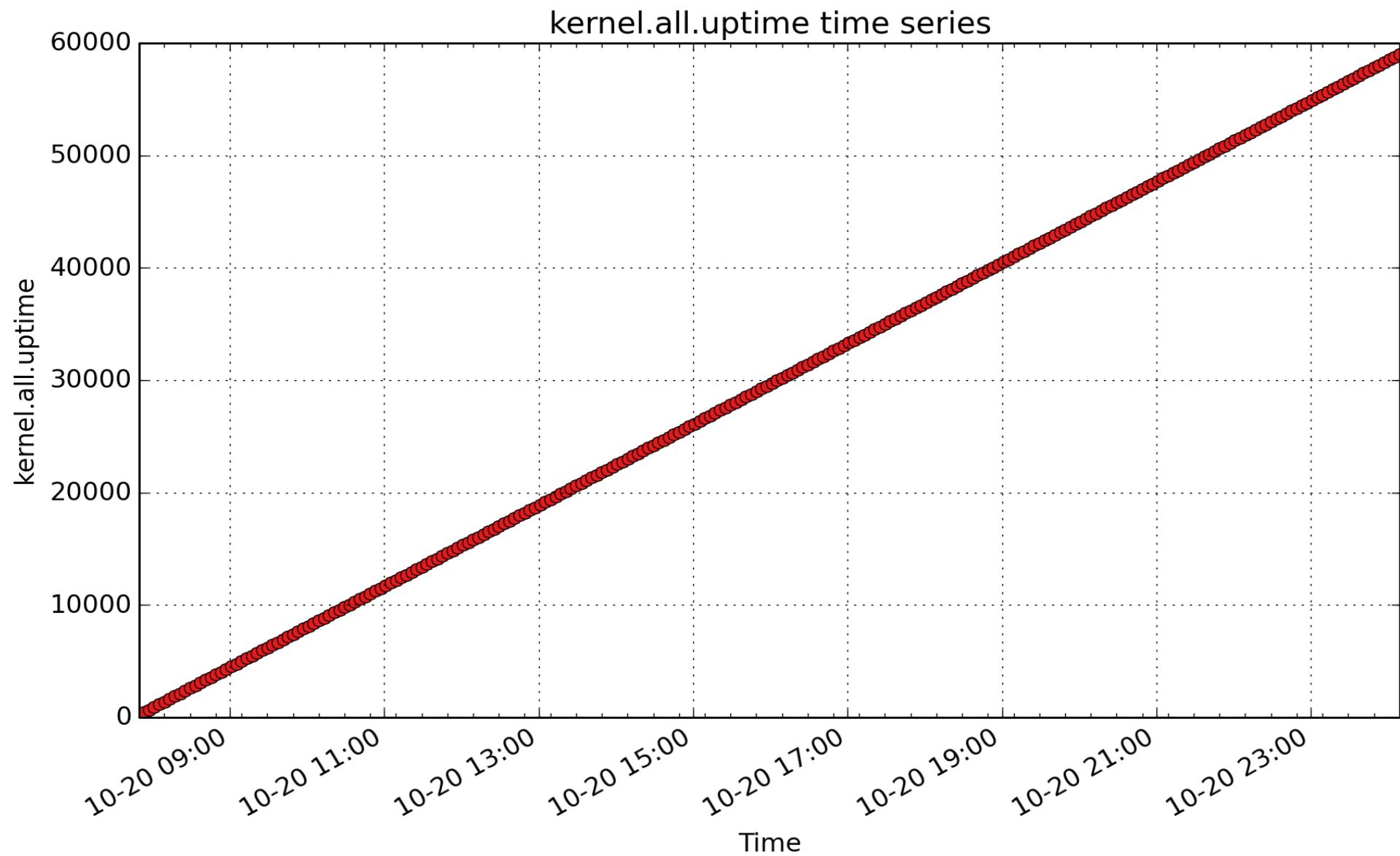
kernel.all.pswitch: context switches metric from /proc/stat (count - U64) - *rate converted*



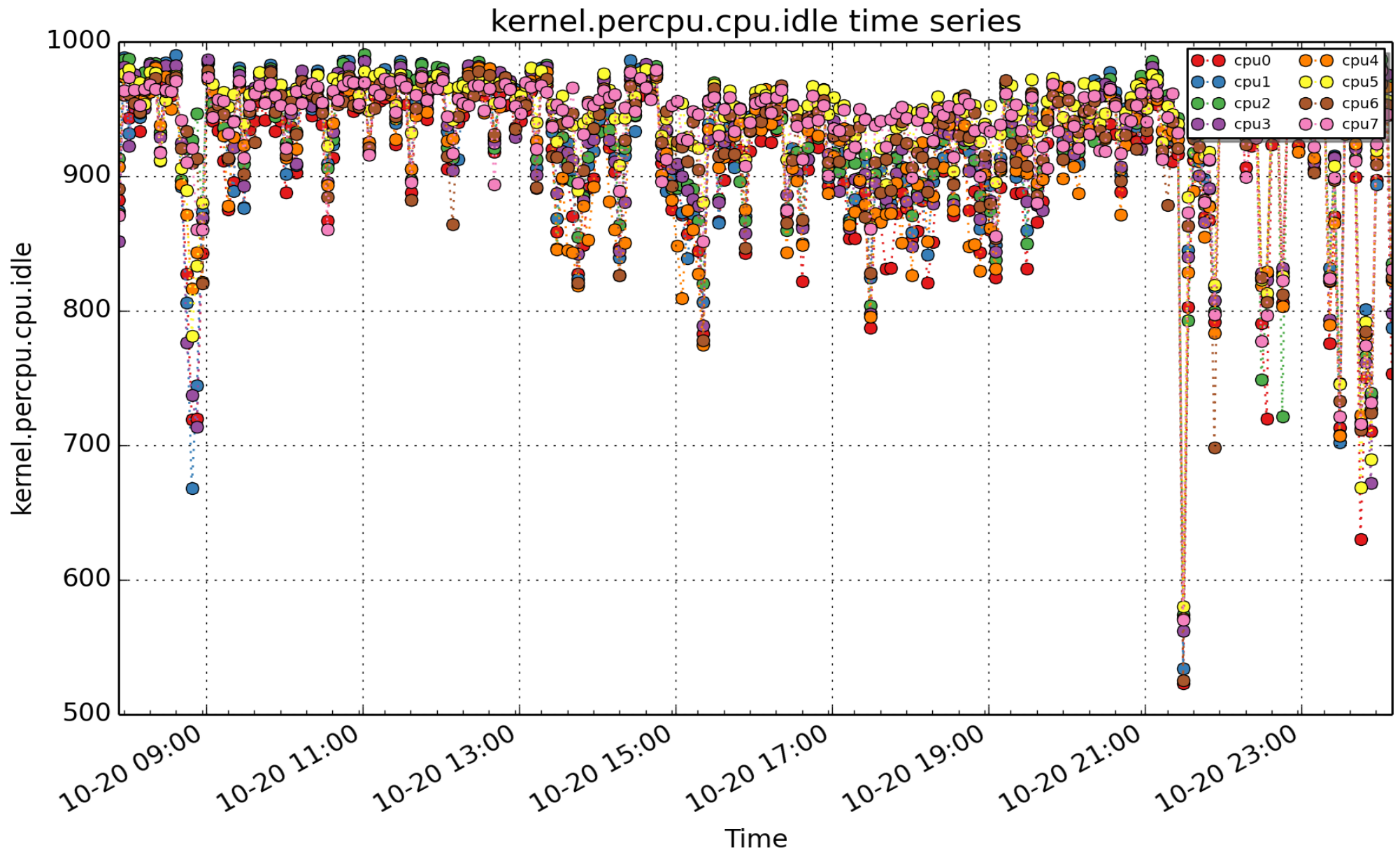
kernel.all.runnable: total number of processes in the (per-CPU) run queues (- U32)



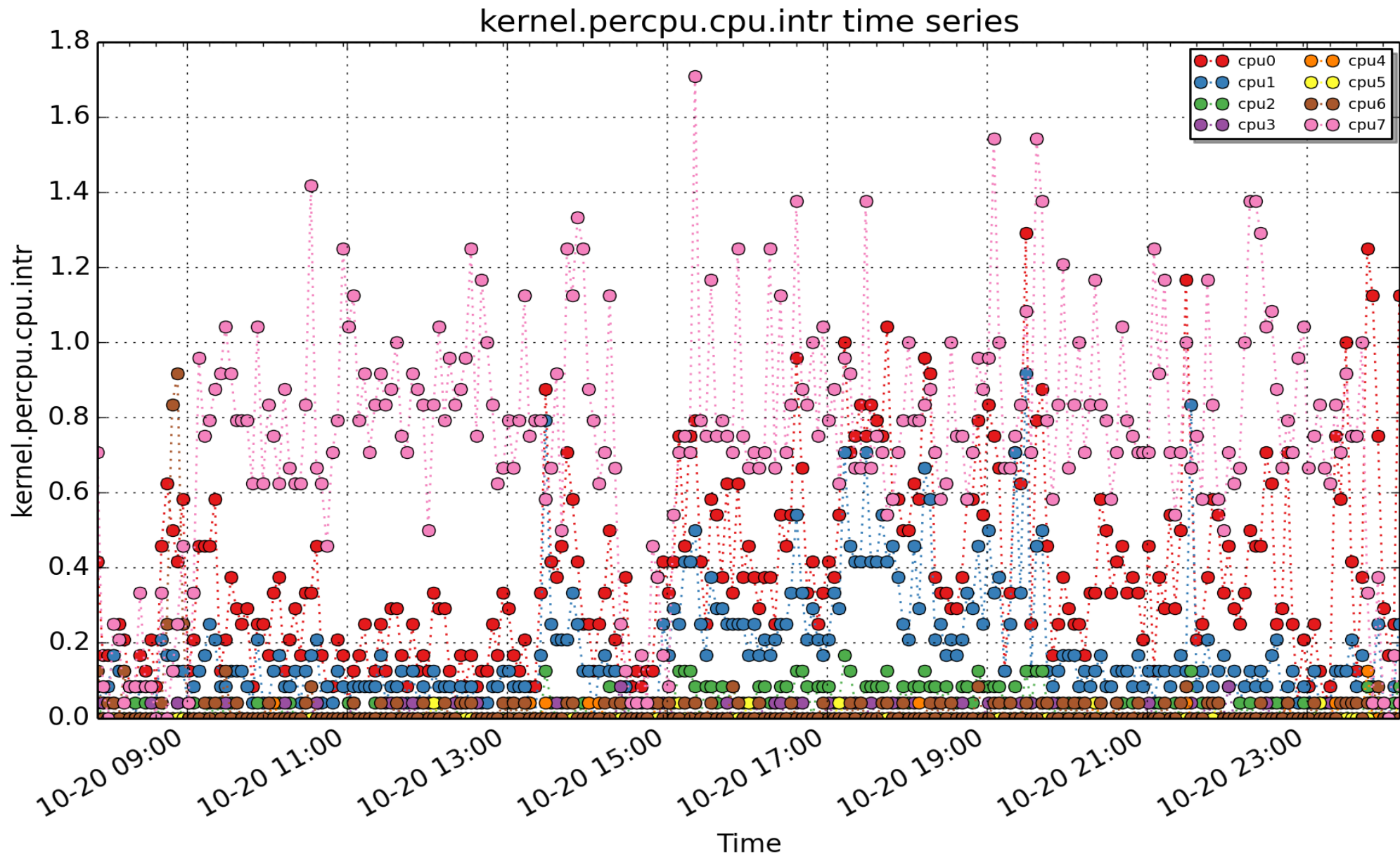
kernel.all.sysfork: fork rate metric from /proc/stat (count - U64) - *rate converted*



kernel.all.uptime: time the current kernel has been running (sec - U32)

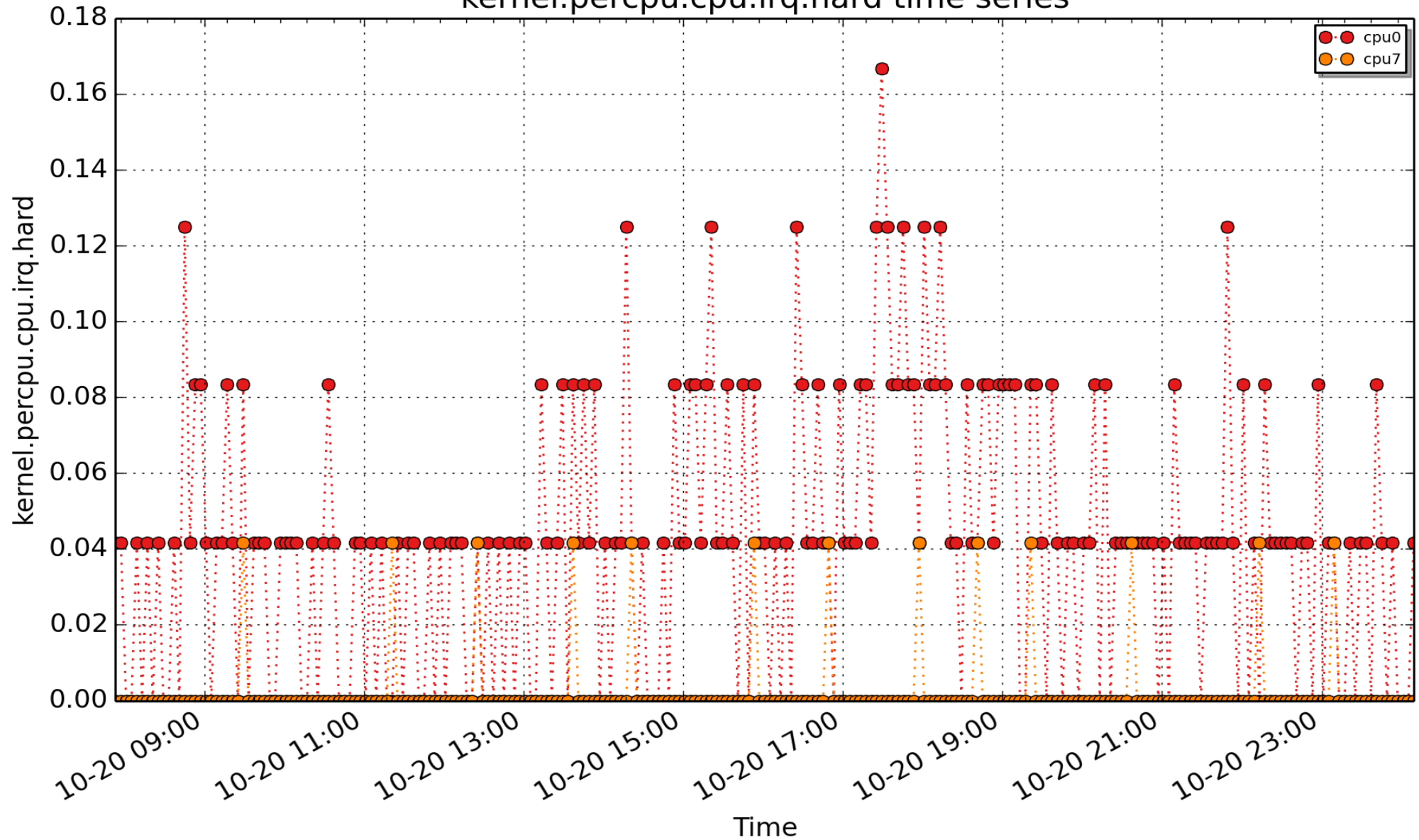


kernel.percpu.cpu.idle: percpu idle CPU time metric from /proc/stat (millisec - U64) - *rate converted*



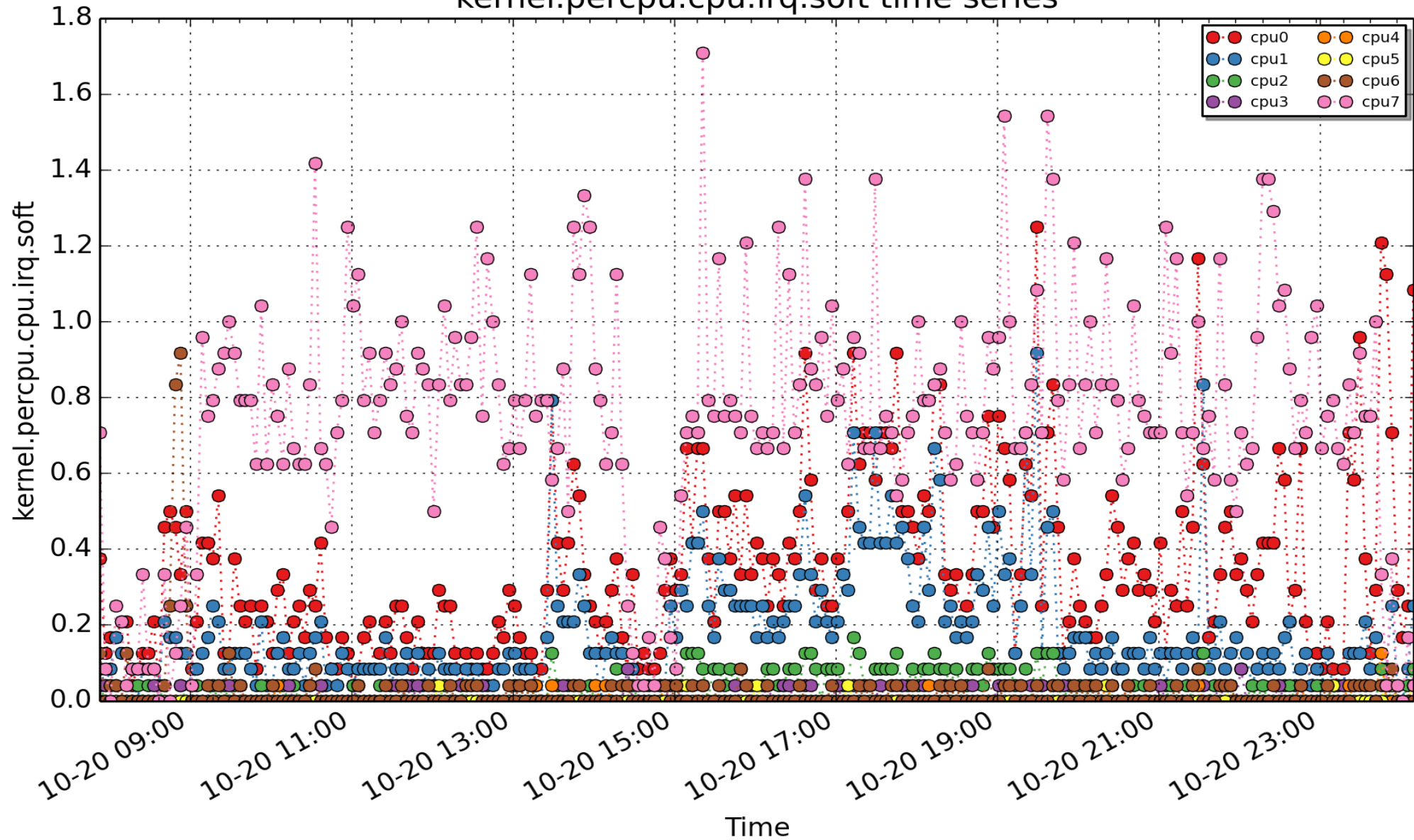
kernel.percpu.cpu.intr: Total time spent processing interrupts on each CPU (this includes both soft and hard interrupt processing time). (millisec - U64) - *rate converted*

kernel.percpu.cpu irq.hard time series

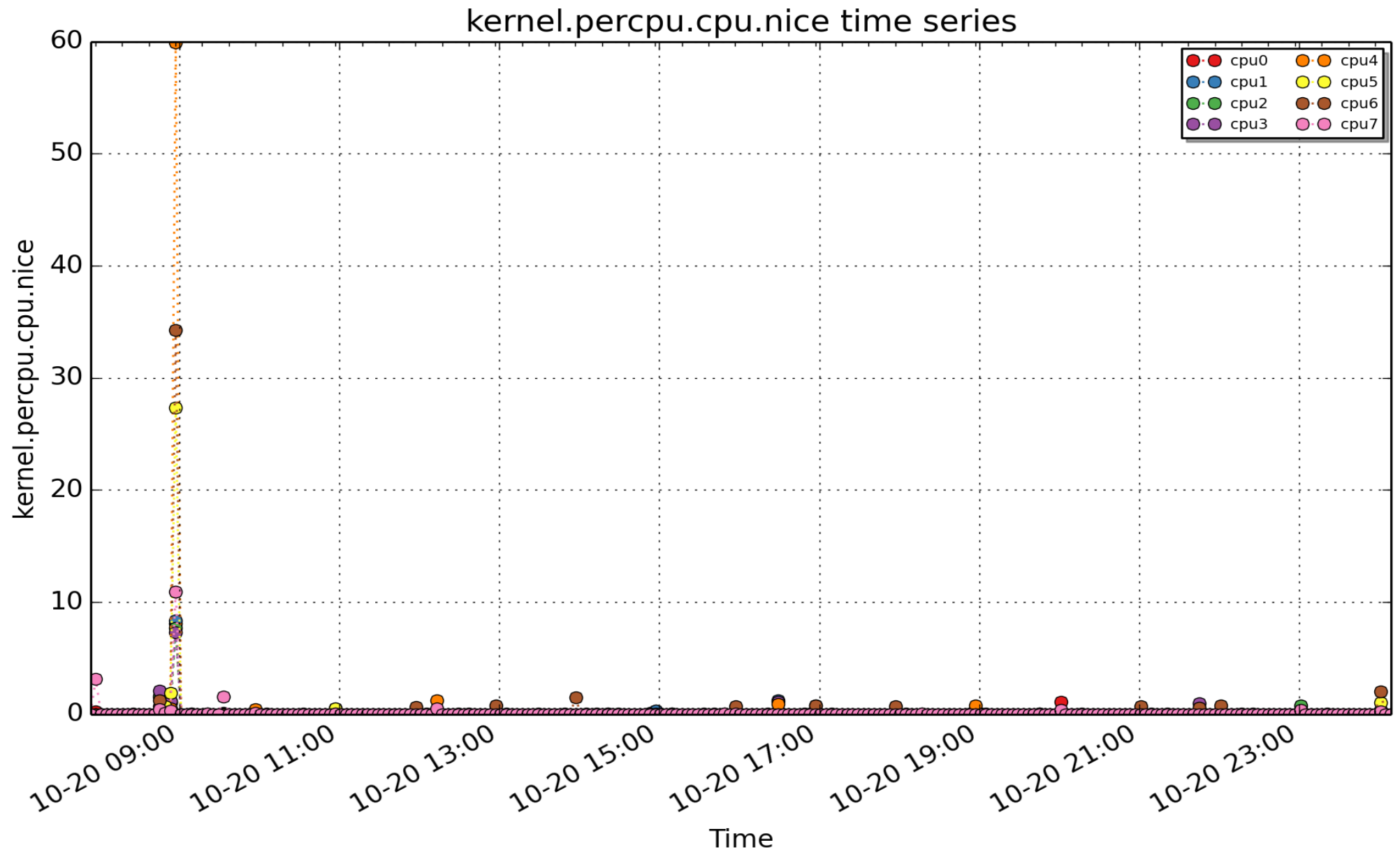


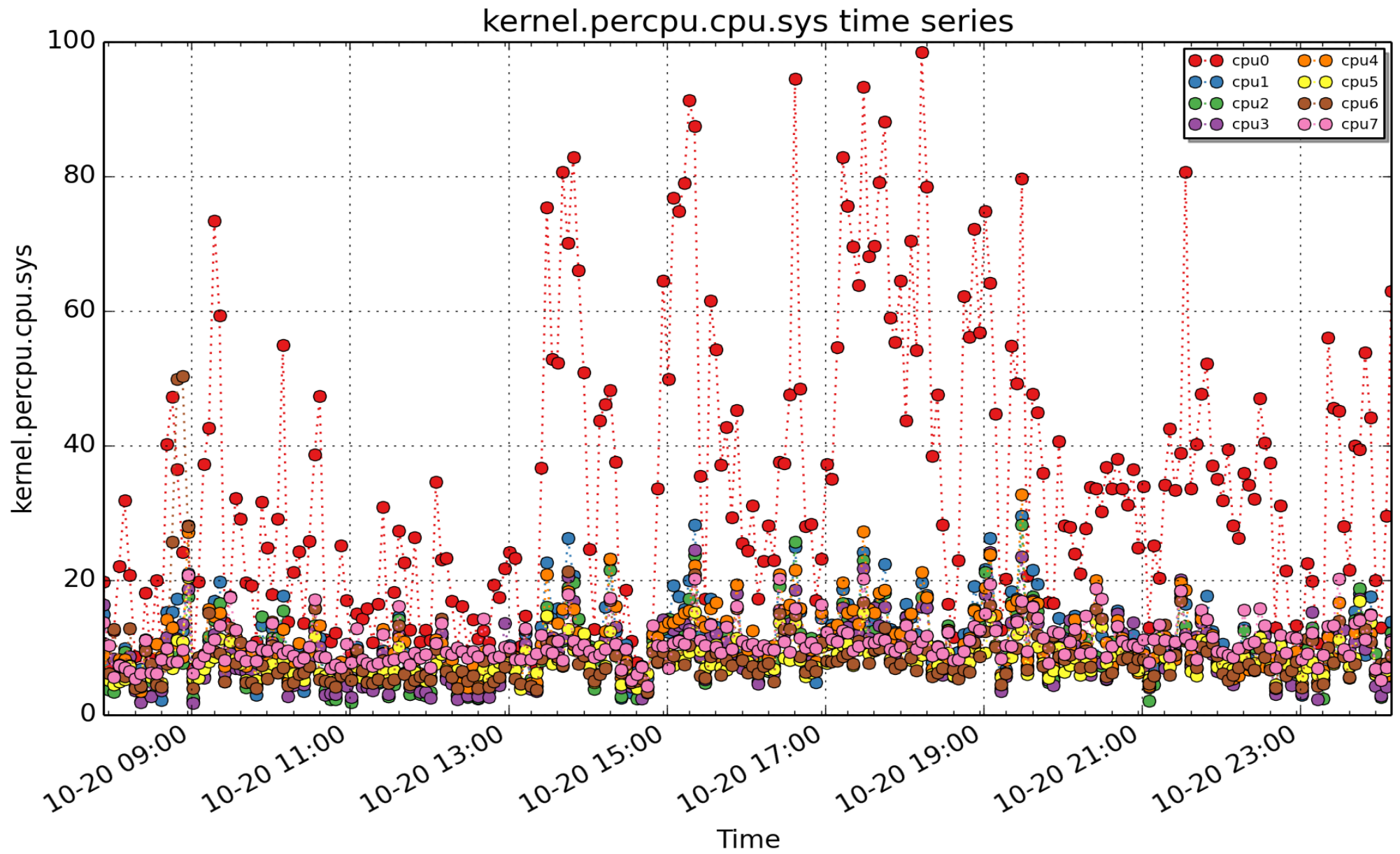
kernel.percpu.cpu.irq.hard: Per-CPU hard interrupt CPU time ("hard" interrupt handling code is the code run directly on receipt of the initial hardware interrupt, and does not include "soft" interrupt handling code which is deferred until later). (millisec - U64) - *rate converted*

kernel.percpu.cpu irq.soft time series



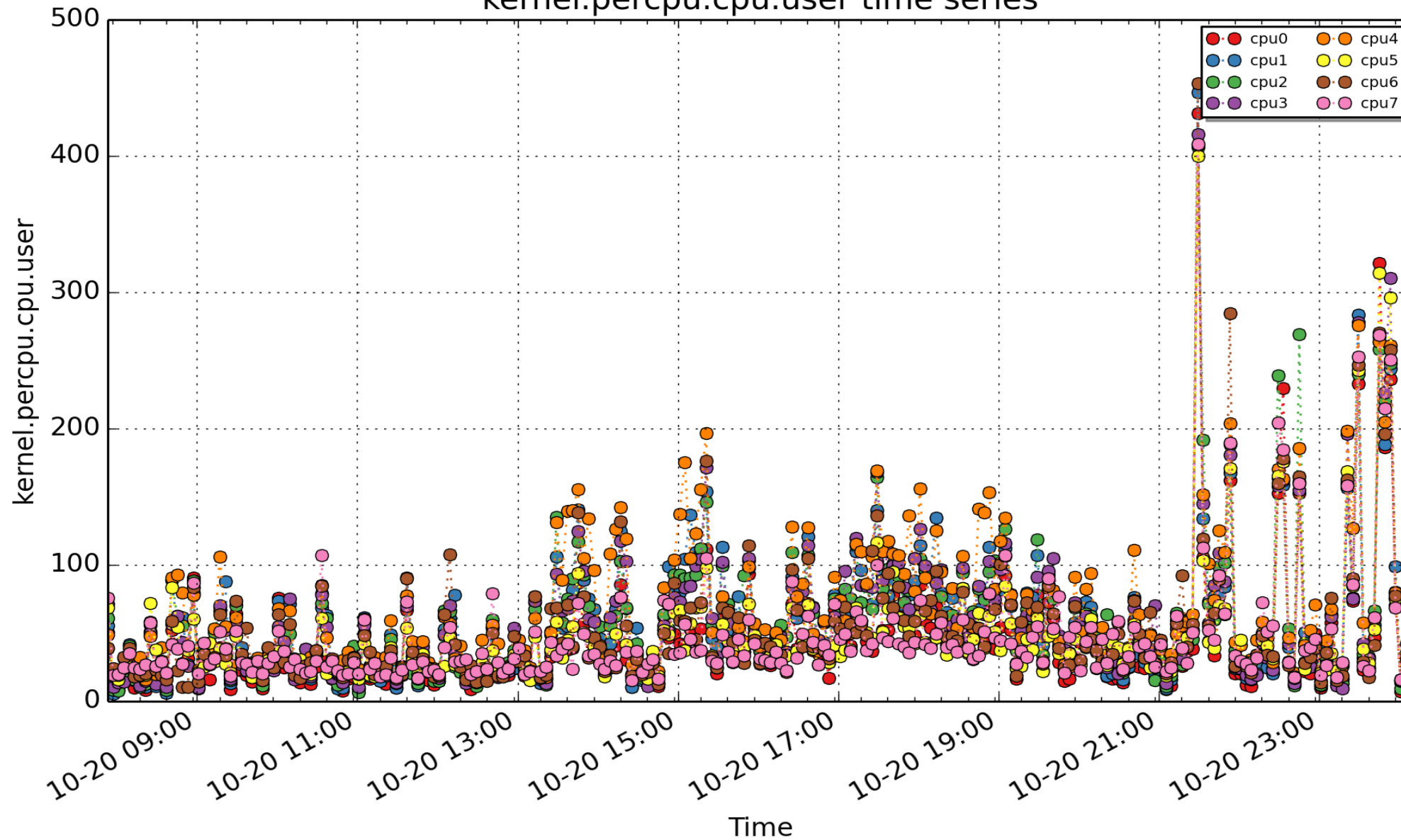
kernel.percpu.cpu irq.soft: Per-CPU soft interrupt CPU time (deferred interrupt handling code, not run in the initial interrupt handler). (millisec - U64) - *rate converted*





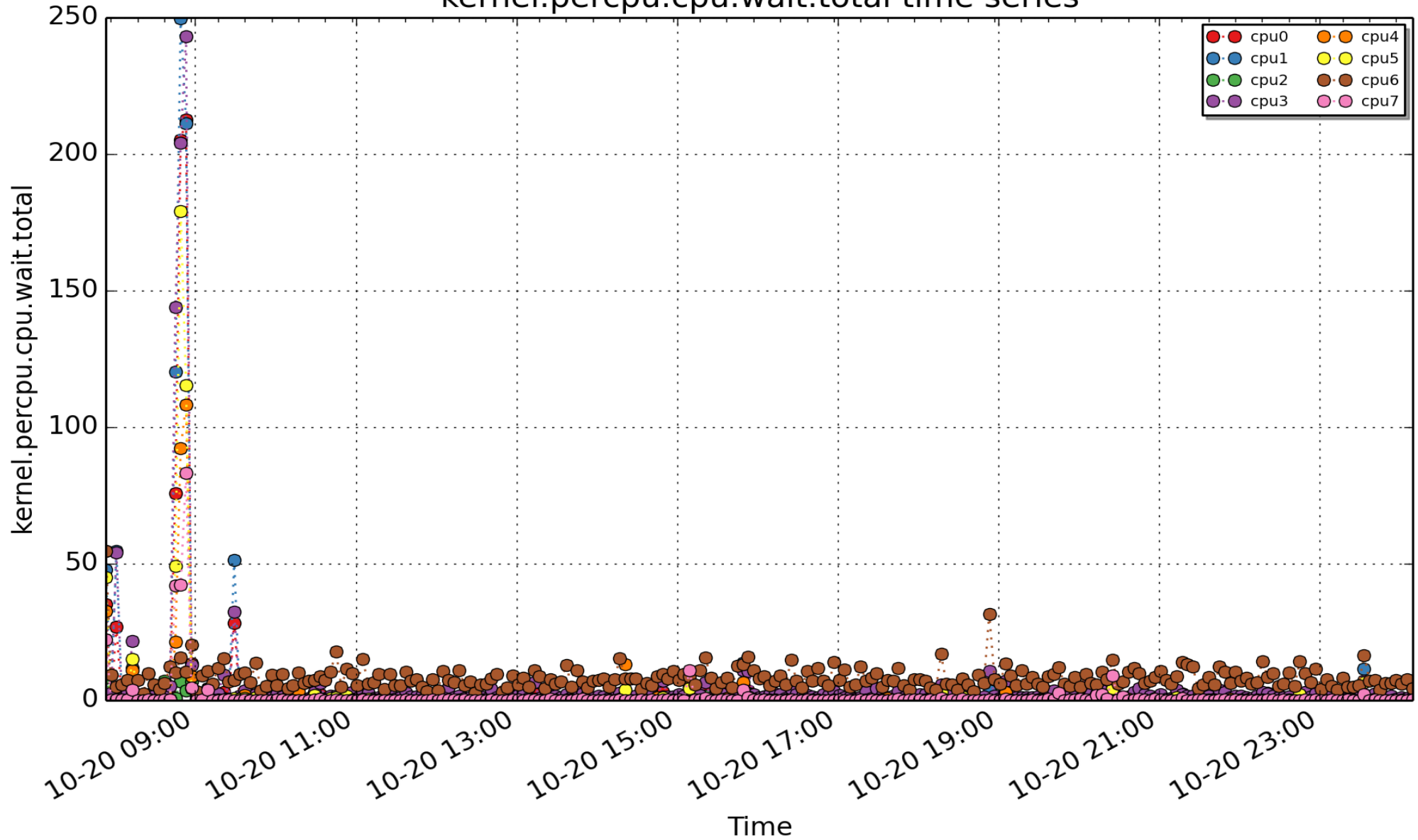
kernel.percpu.cpu.sys: percpu sys CPU time metric from /proc/stat (millisec - U64) - *rate converted*

kernel.percpu.cpu.user time series



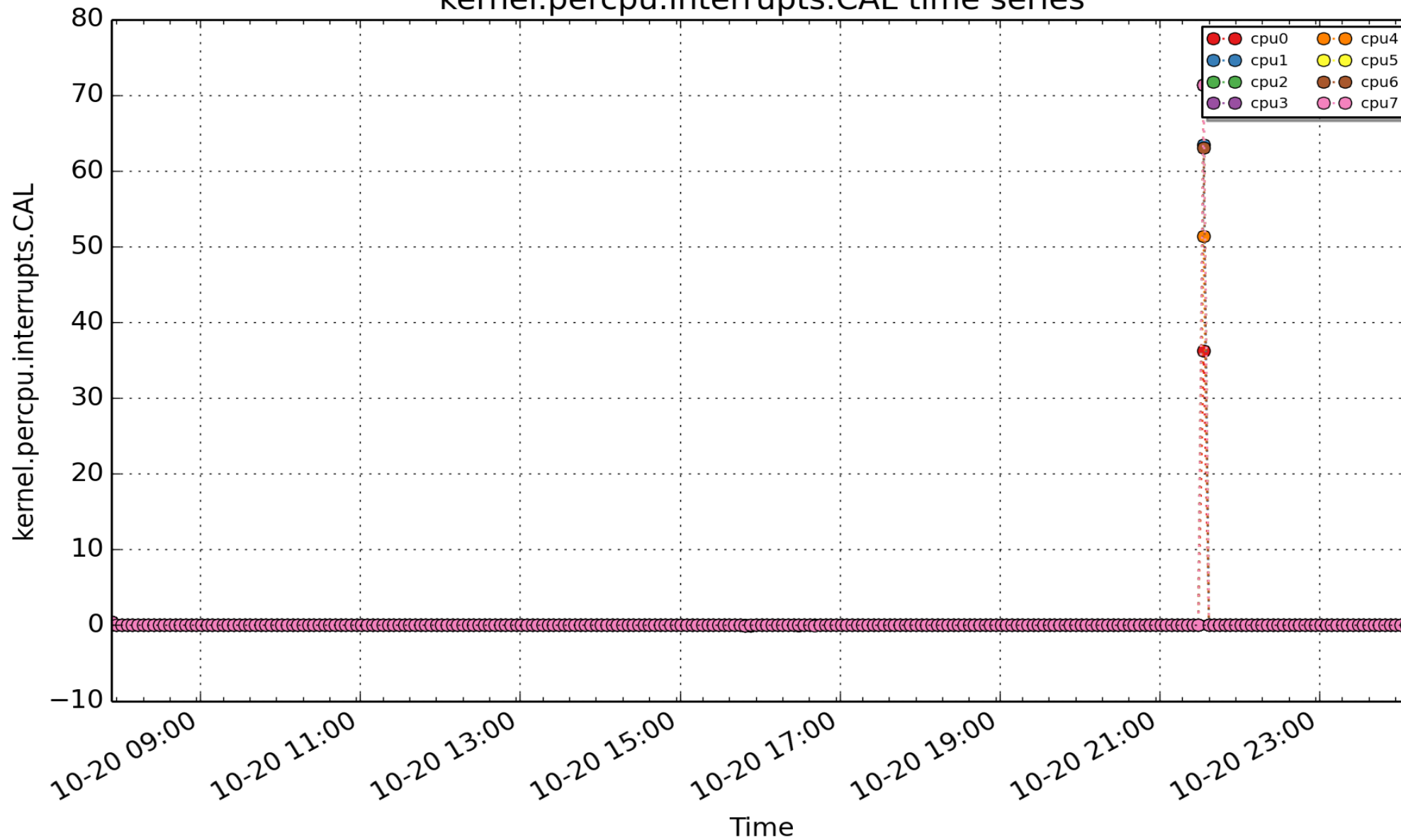
kernel.percpu.cpu.user: percpu user CPU time metric from /proc/stat, including guest CPU time (millisec - U64) - *rate converted*

kernel.percpu.cpu.wait.total time series

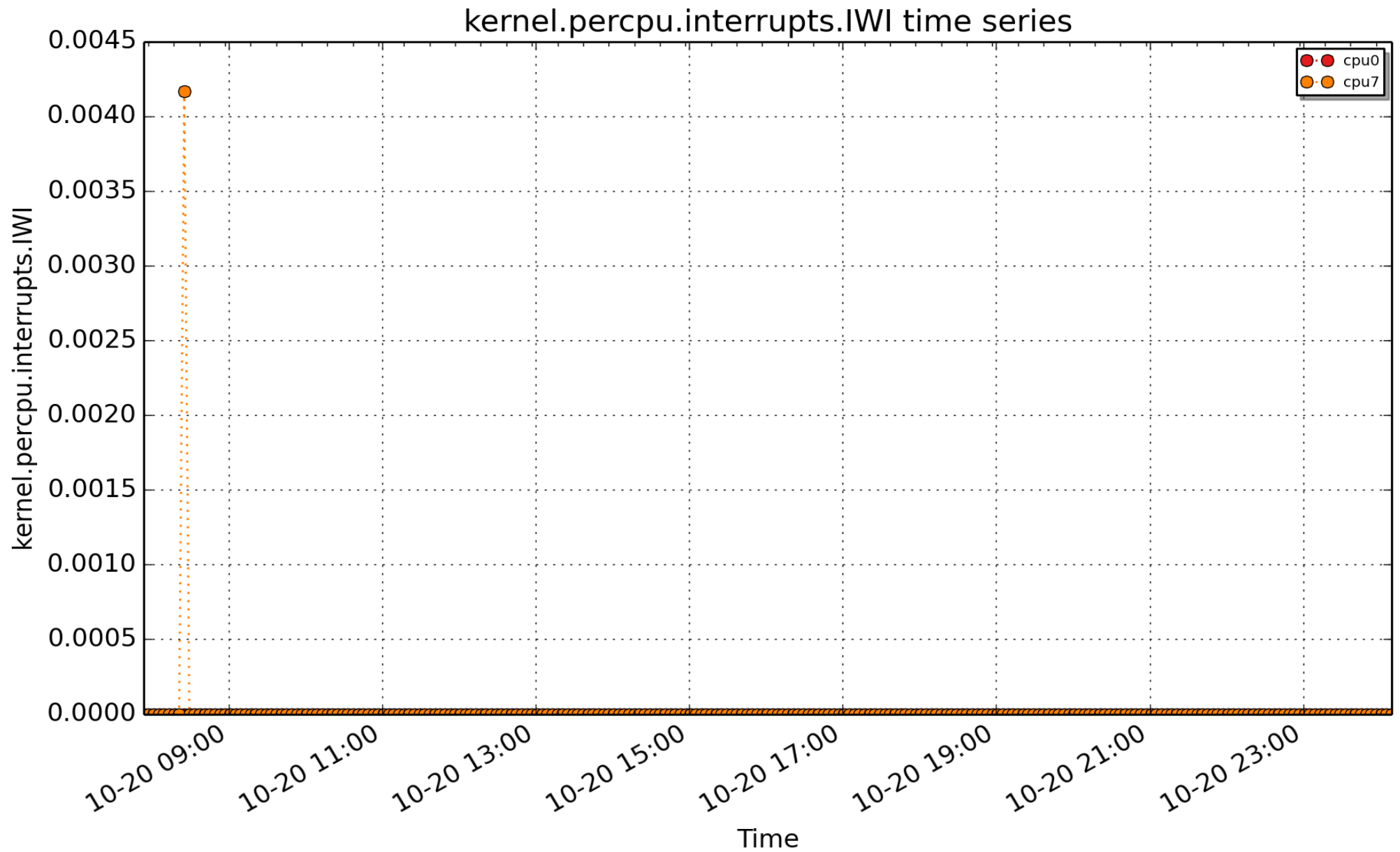


kernel.percpu.cpu.wait.total: Per-CPU I/O wait CPU time - time spent with outstanding I/O requests.
(millisec - U64) - *rate converted*

kernel.percpu.interrupts.CAL time series

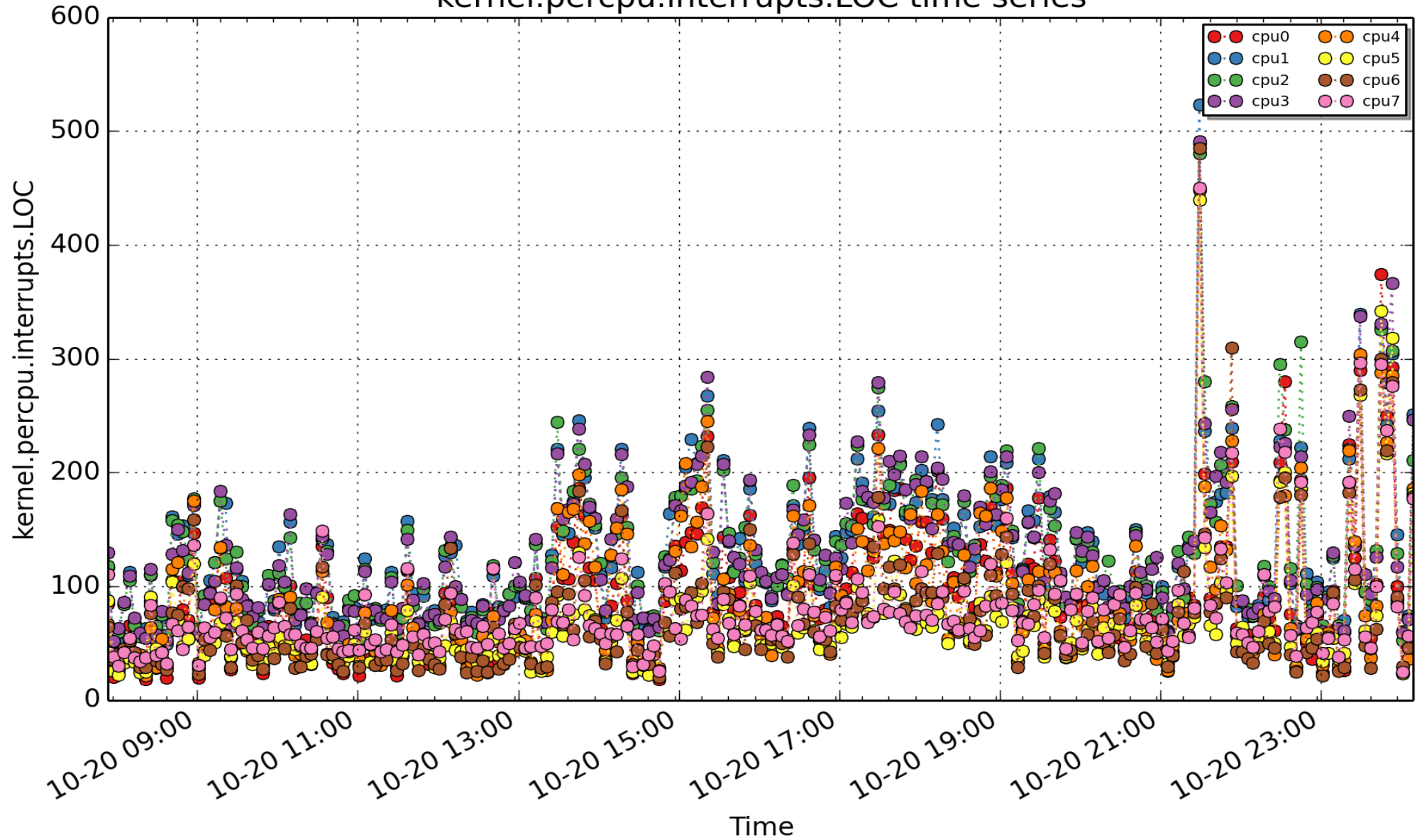


kernel.percpu.interrupts.CAL: Function call interrupts (count - U32) - *rate converted*



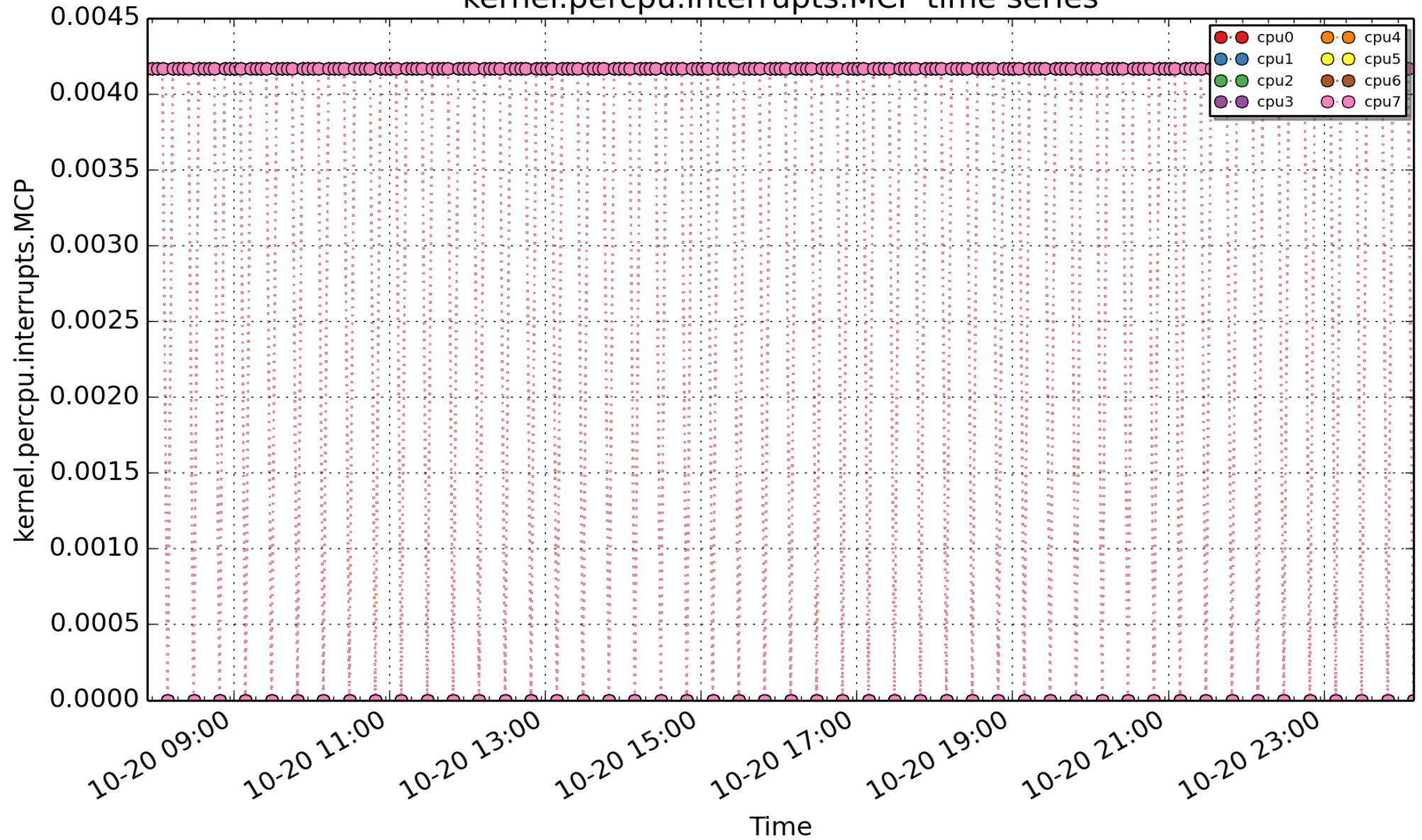
kernel.percpu.interrupts.IWI: IRQ work interrupts (count - U32) - *rate converted*

kernel.percpu.interrupts.LOC time series



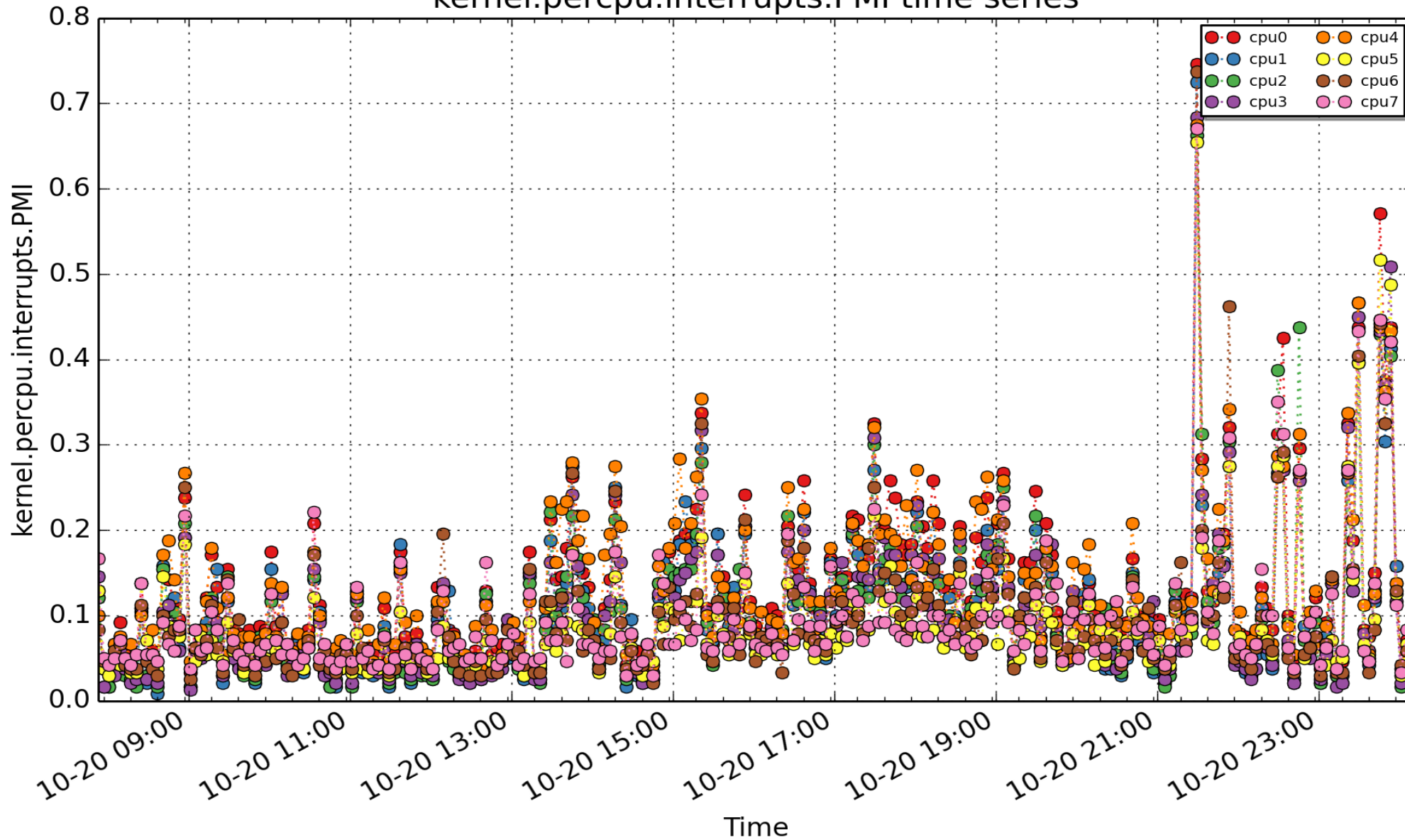
kernel.percpu.interrupts.LOC: Local timer interrupts (count - U32) - *rate converted*

kernel.percpu.interrupts.MCP time series



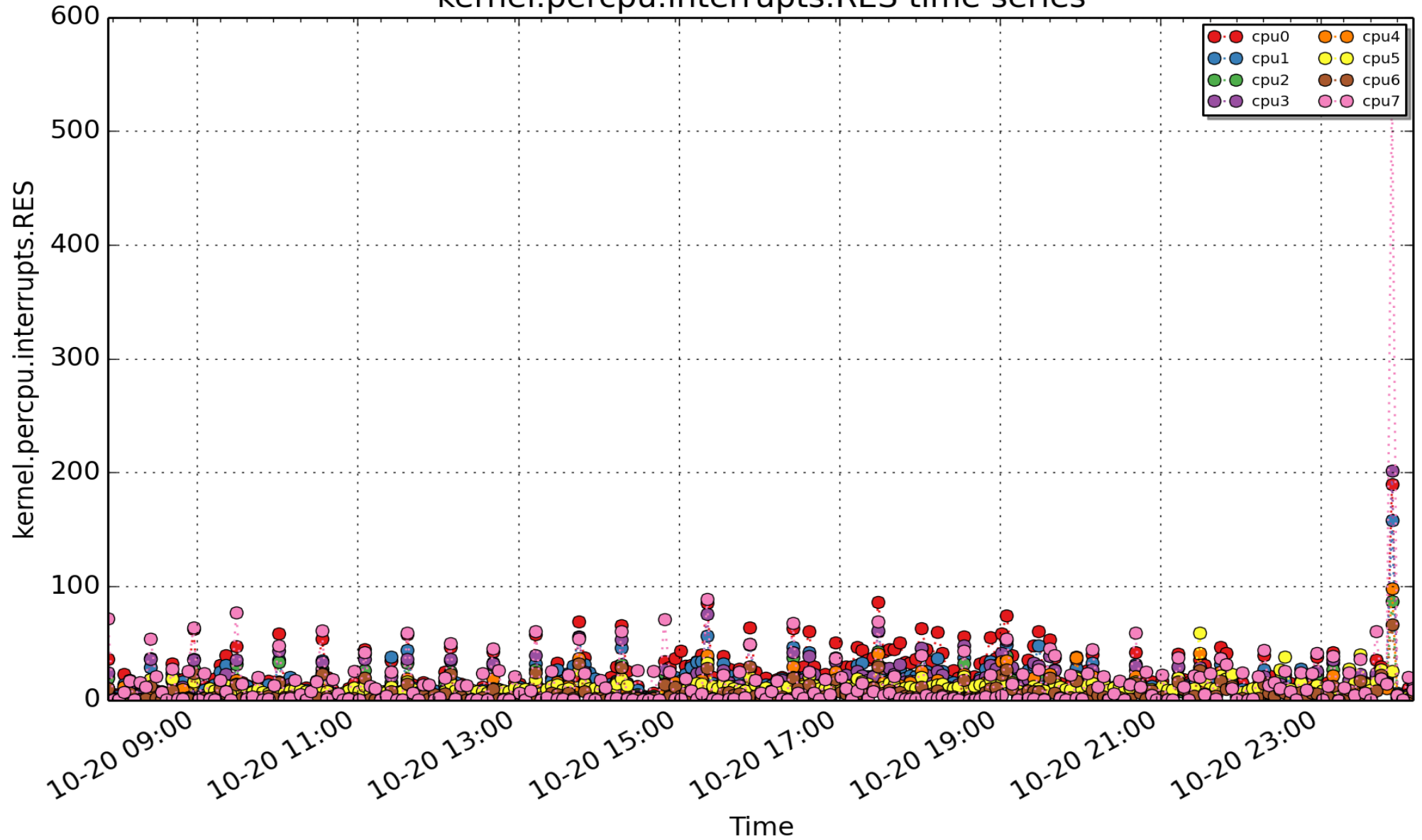
kernel.percpu.interrupts.MCP: Machine check polls (count - U32) - *rate converted*

kernel.percpu.interrupts.PMI time series

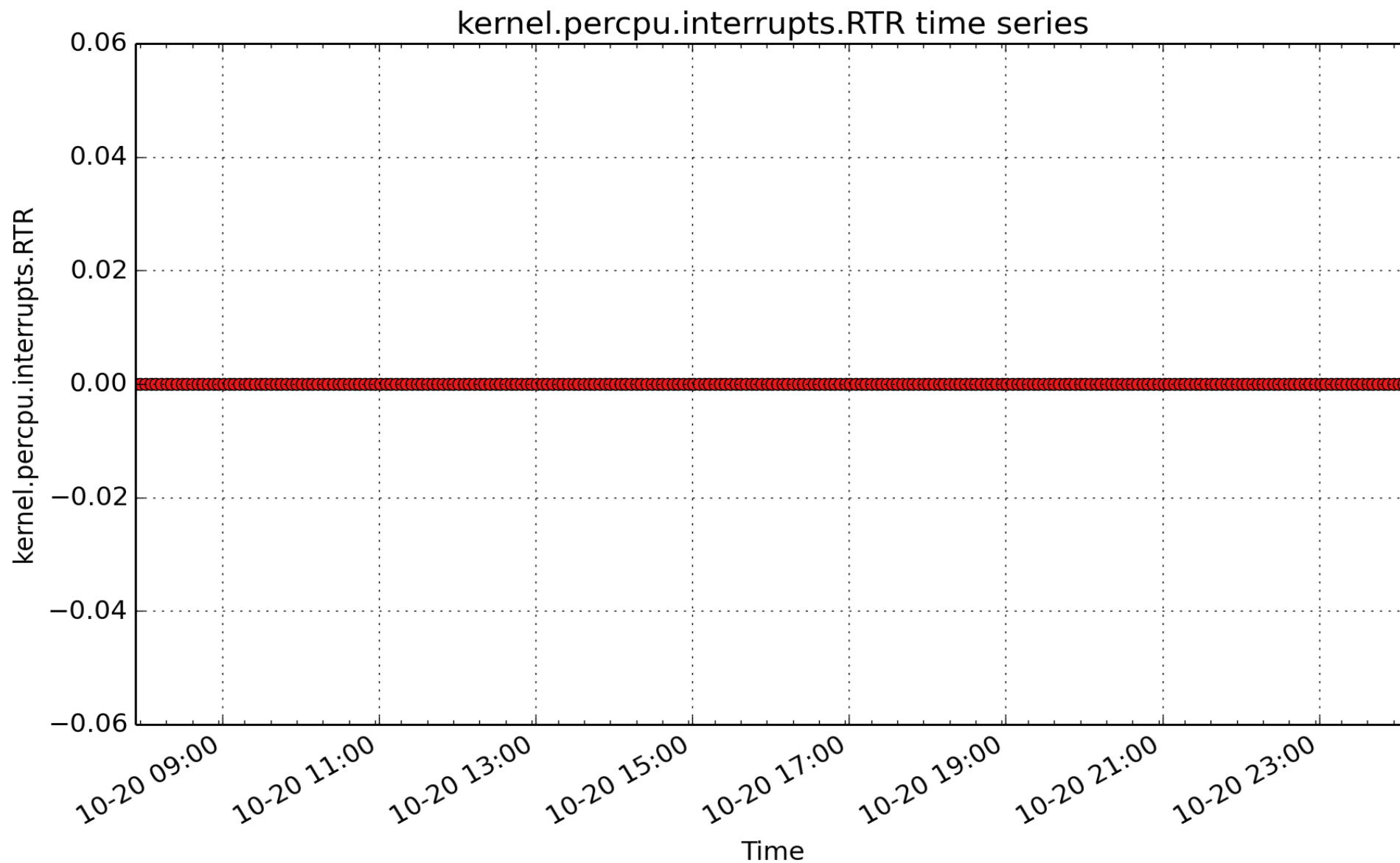


kernel.percpu.interrupts.PMI: Performance monitoring interrupts (count - U32) - *rate converted*

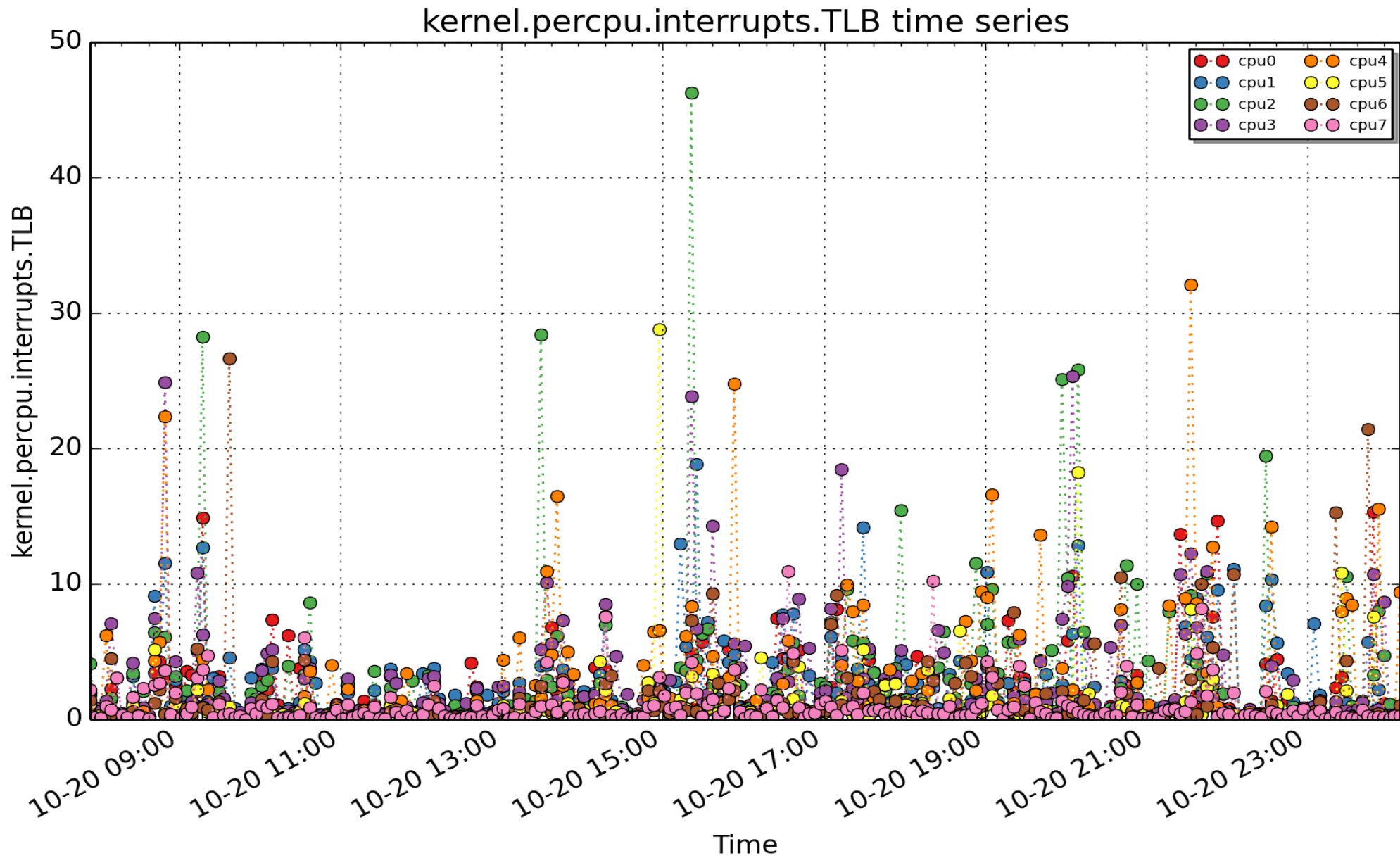
kernel.percpu.interrupts.RES time series



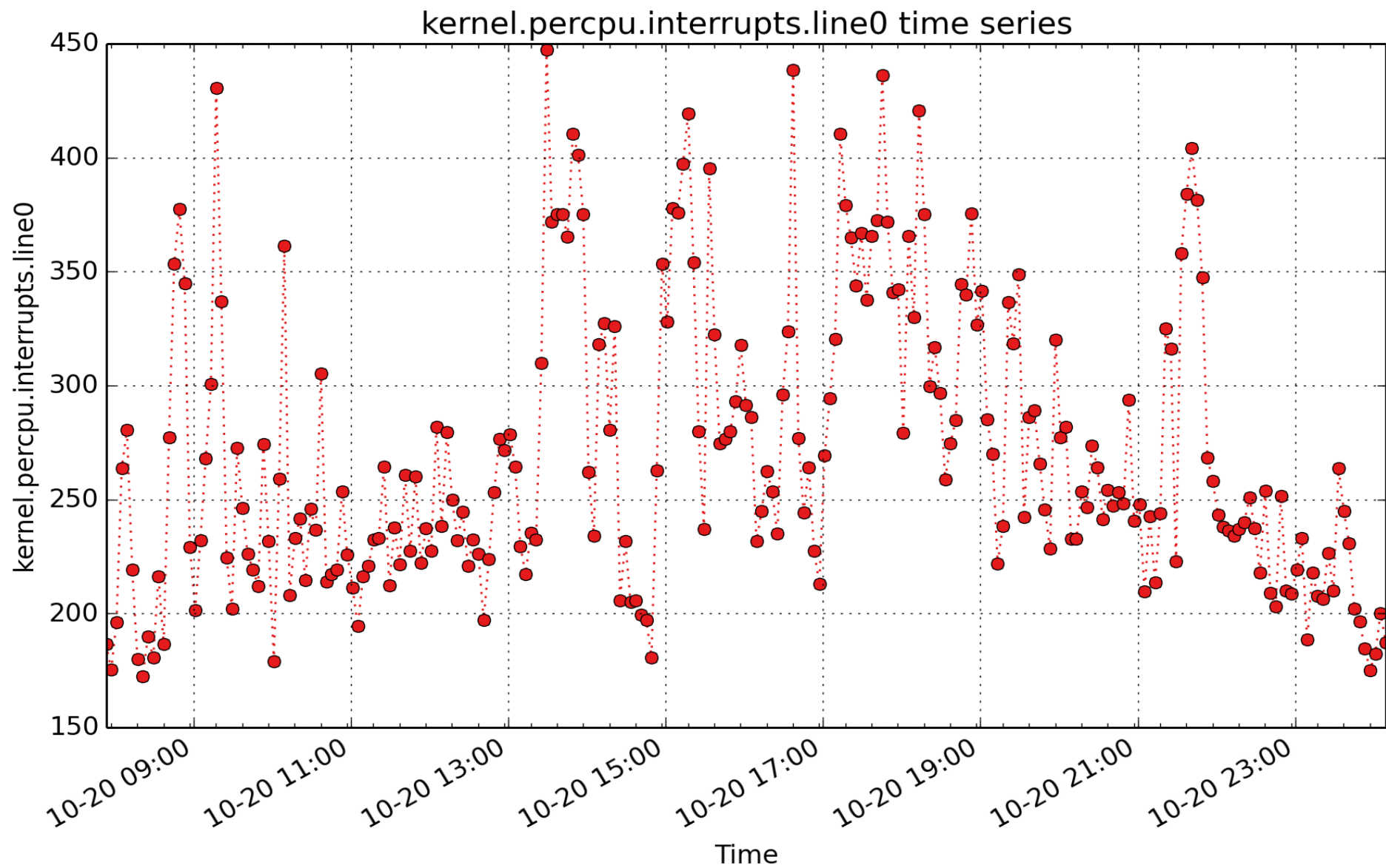
kernel.percpu.interrupts.RES: Rescheduling interrupts (count - U32) - *rate converted*



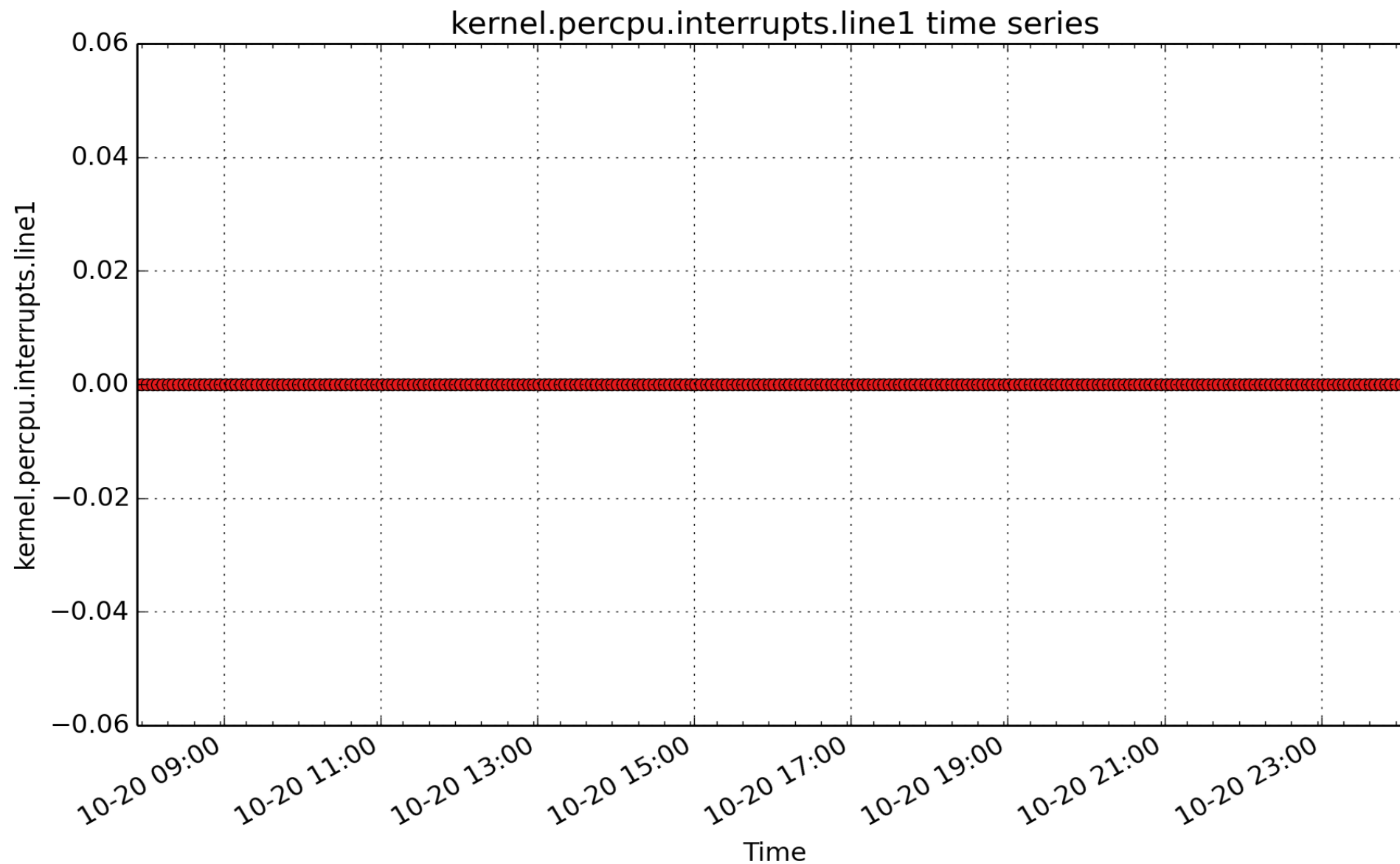
kernel.percpu.interrupts.RTR: APIC ICR read retries (count - U32) - *rate converted*



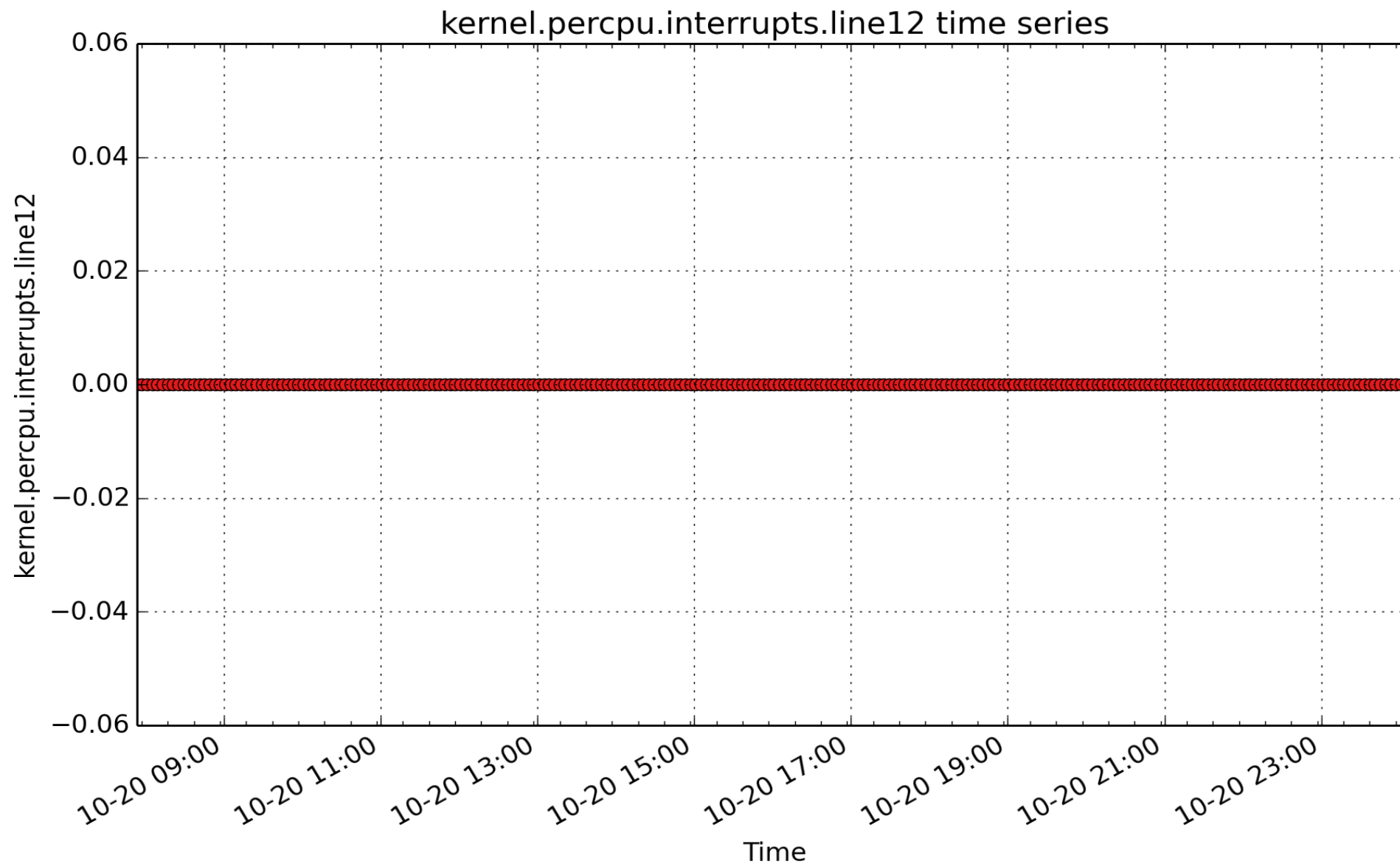
kernel.percpu.interrupts.TLB: TLB shutdowns (count - U32) - *rate converted*



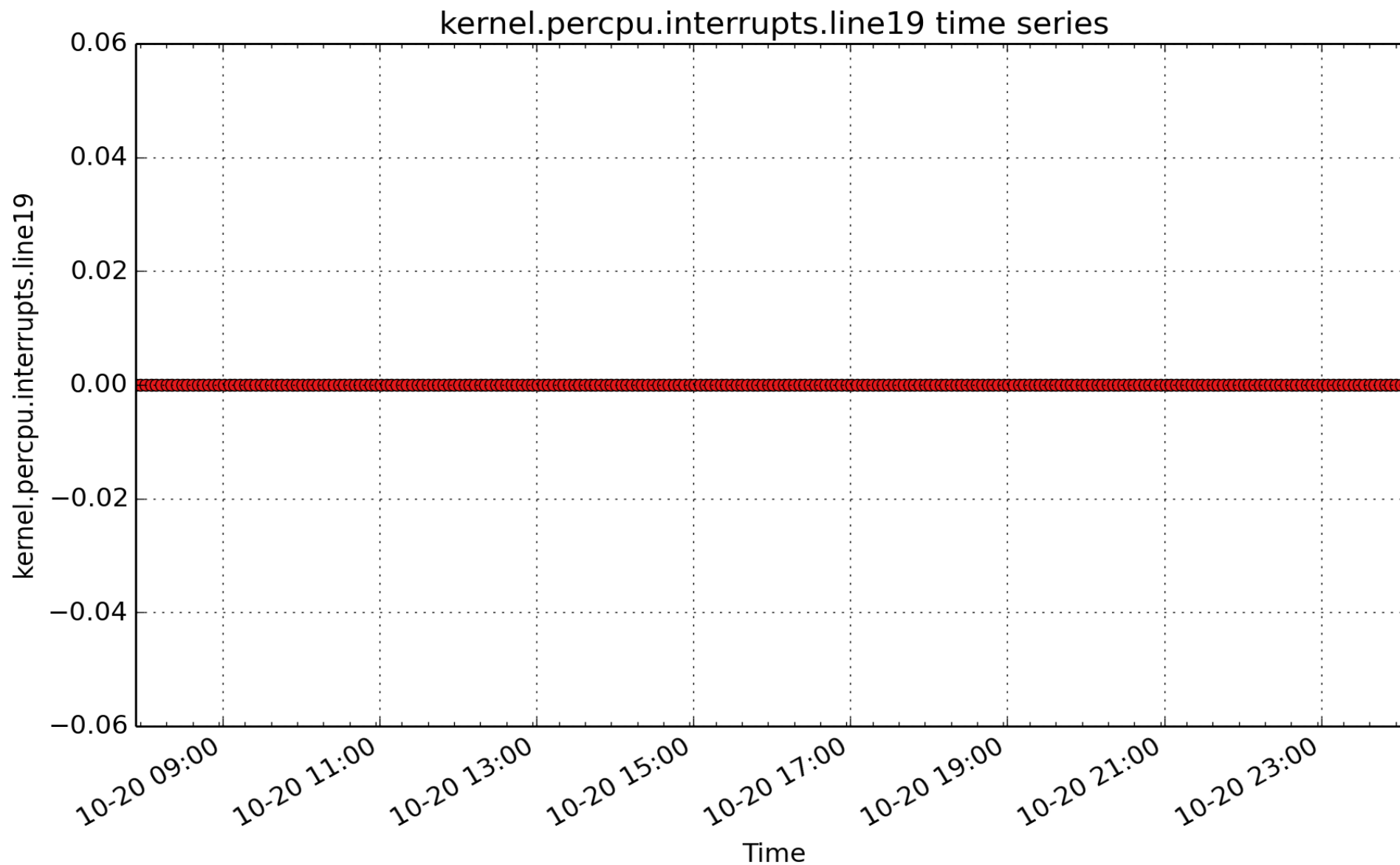
kernel.percpu.interrupts.line0: IO-APIC-edge timer (count - U32) - *rate converted*



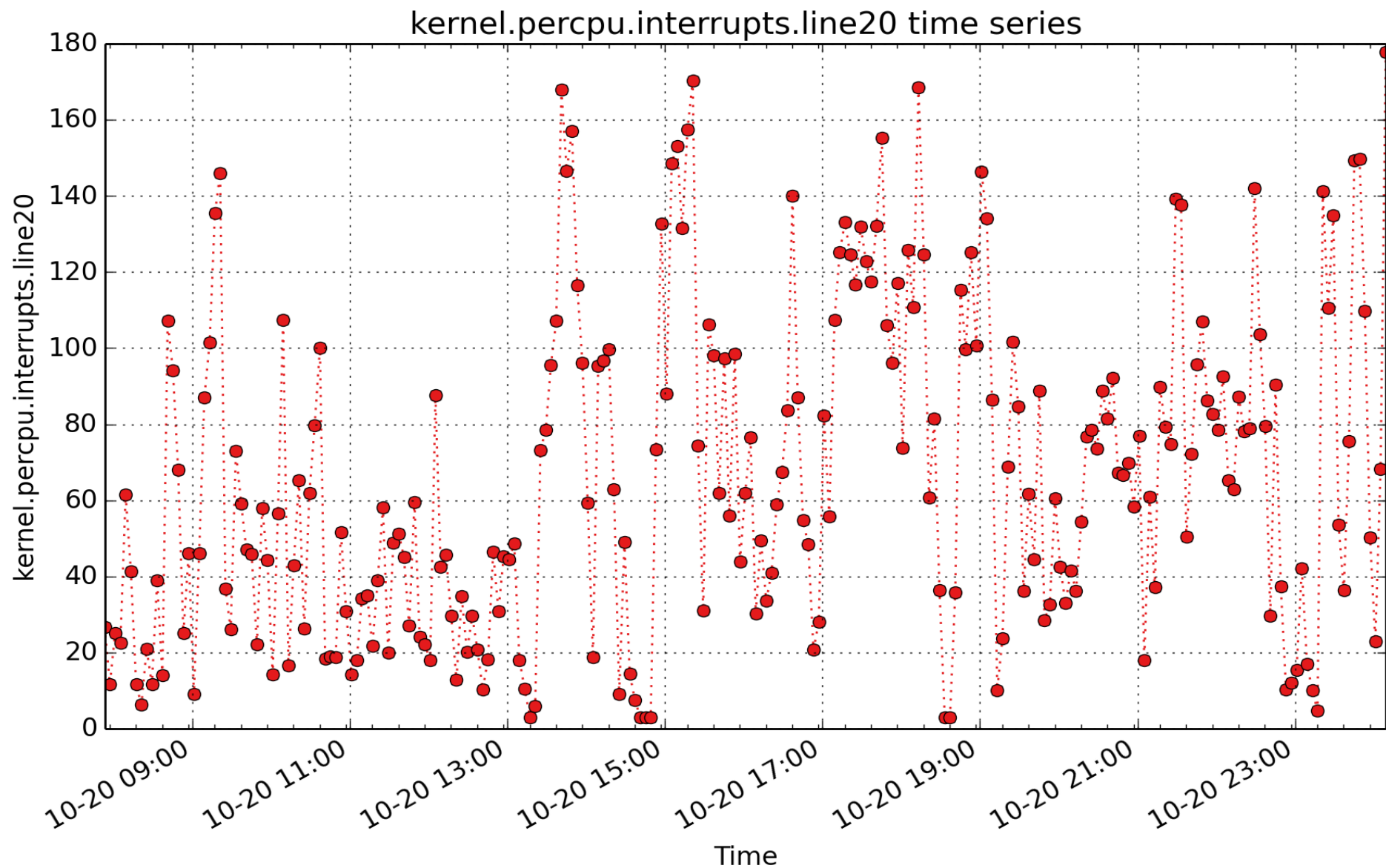
kernel.percpu.interrupts.line1: IO-APIC-edge i8042 (count - U32) - *rate converted*



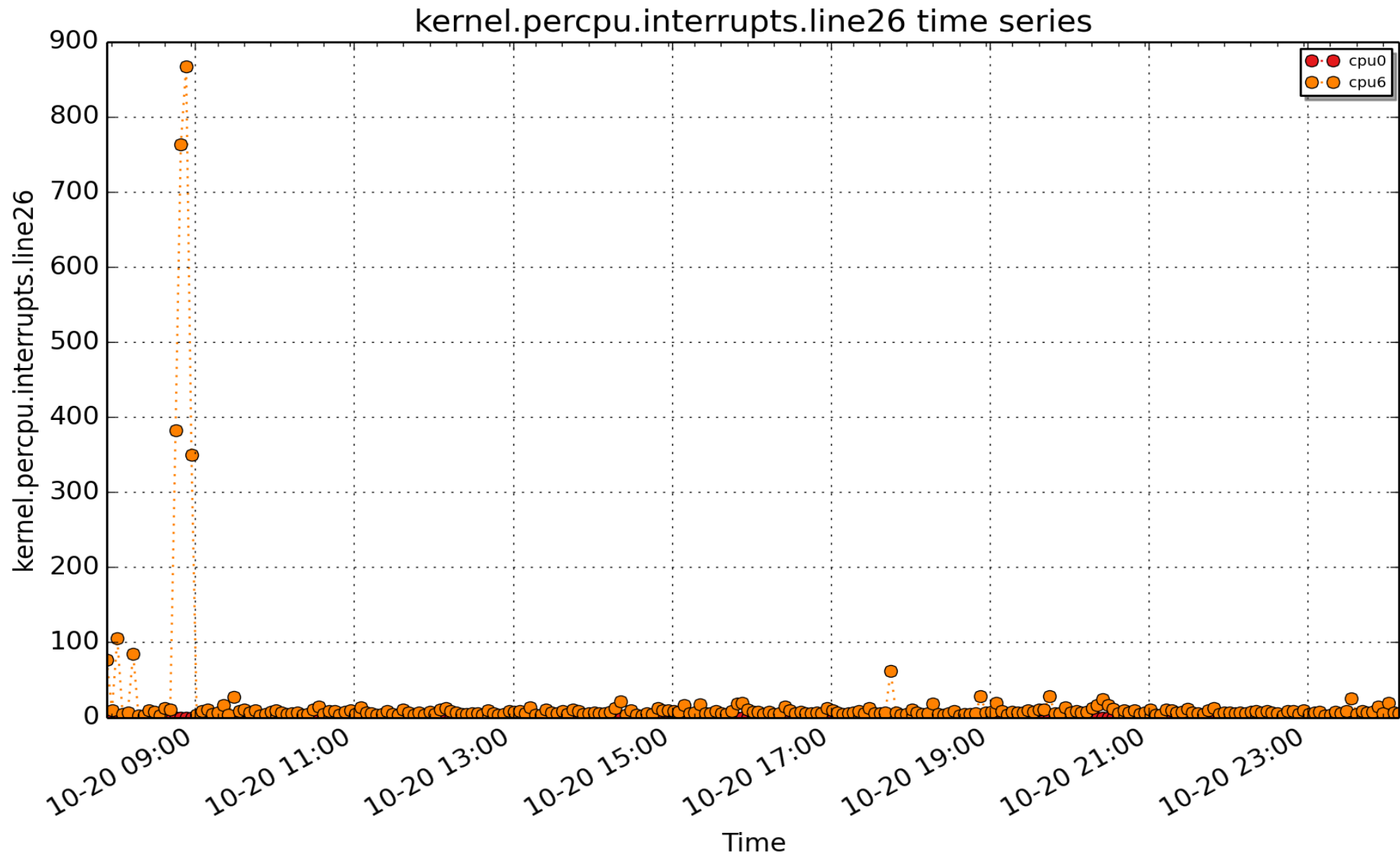
kernel.percpu.interrupts.line12: IO-APIC-edge i8042 (count - U32) - *rate converted*



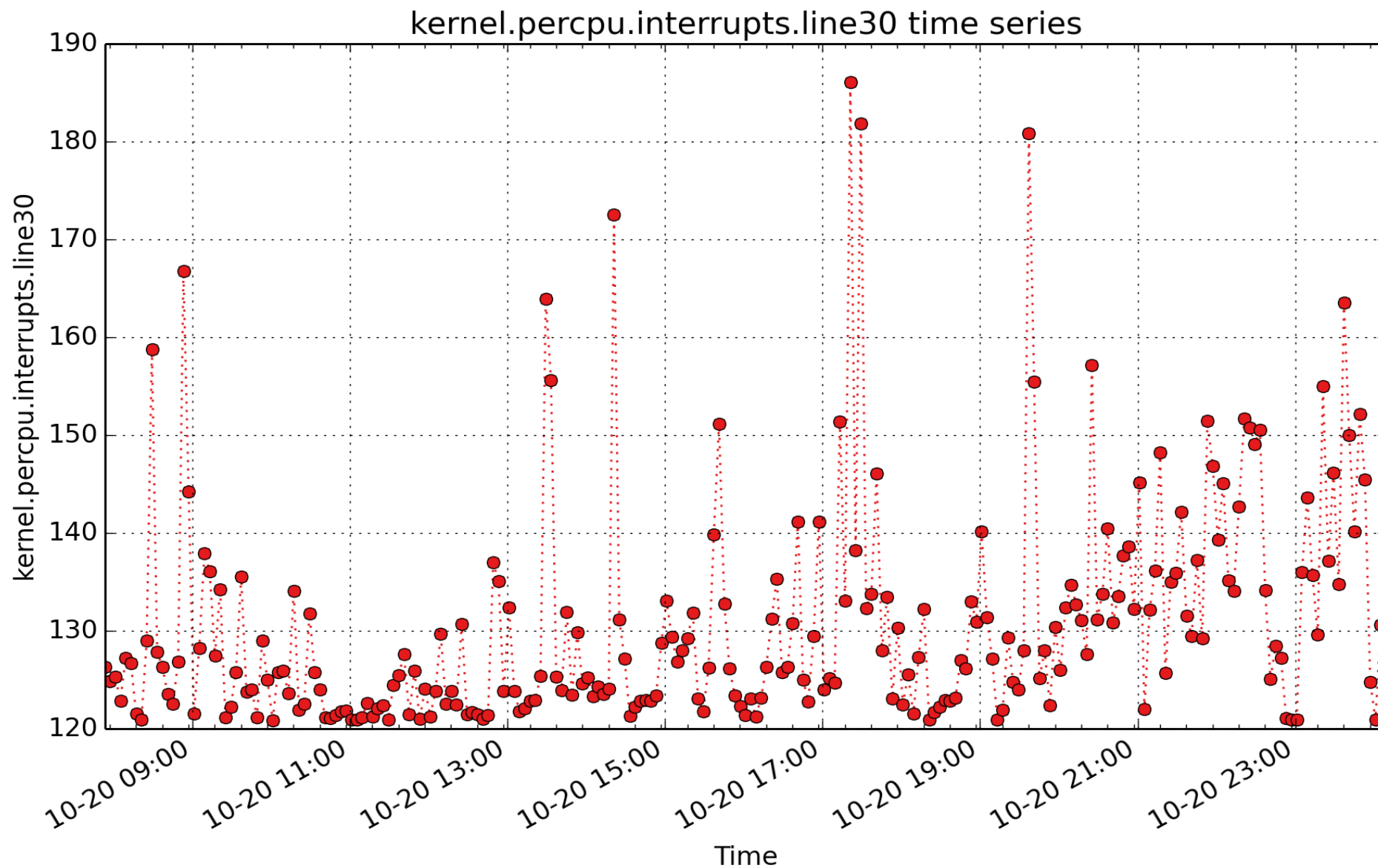
kernel.percpu.interrupts.line19: IO-APIC 19-fasteoi firewire_ohci (count - U32) - *rate converted*



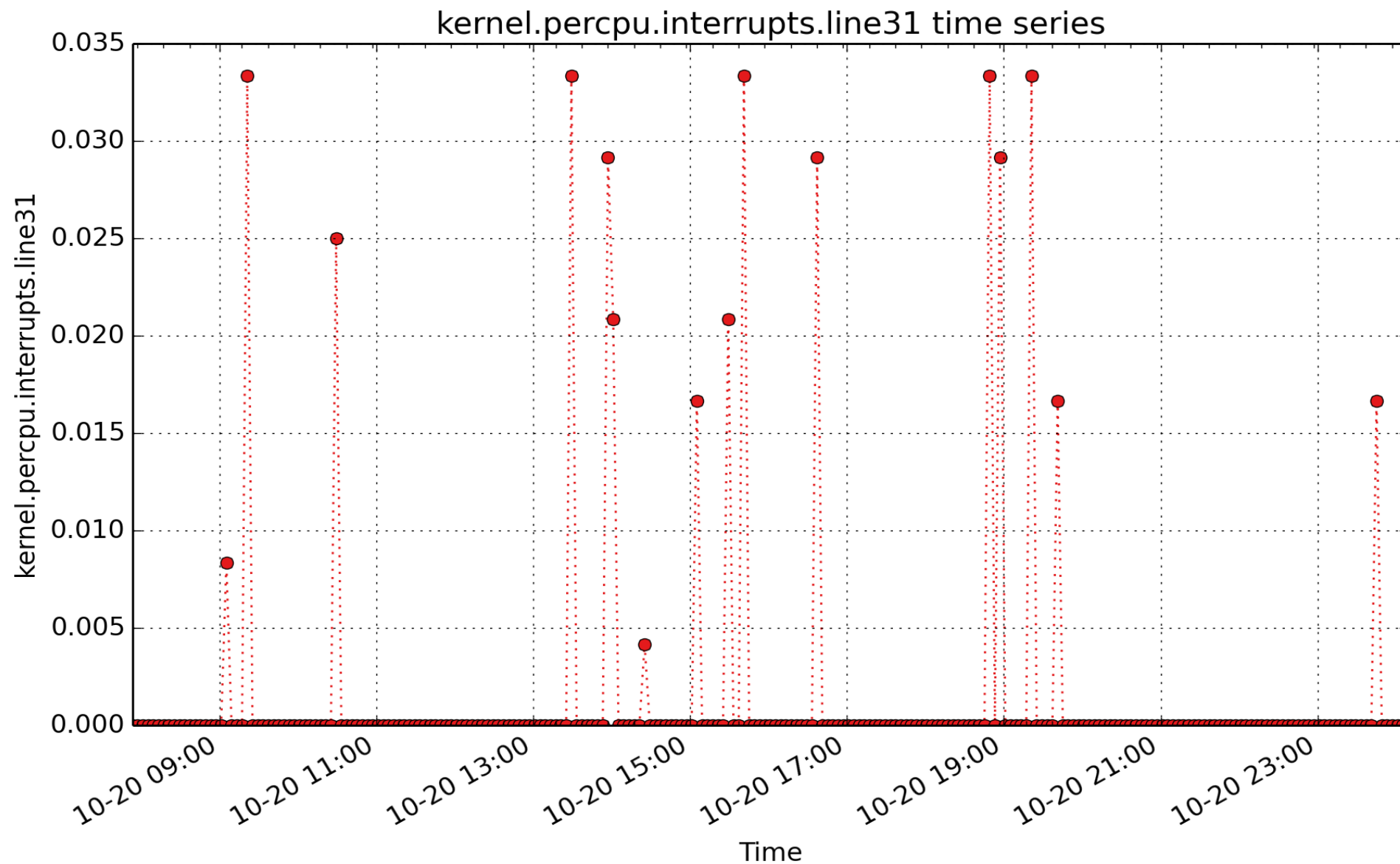
kernel.percpu.interrupts.line20: IO-APIC 20-fasteoi ehci_hcd:usb2, uhci_hcd:usb3, uhci_hcd:usb6 (count - U32) - *rate converted*



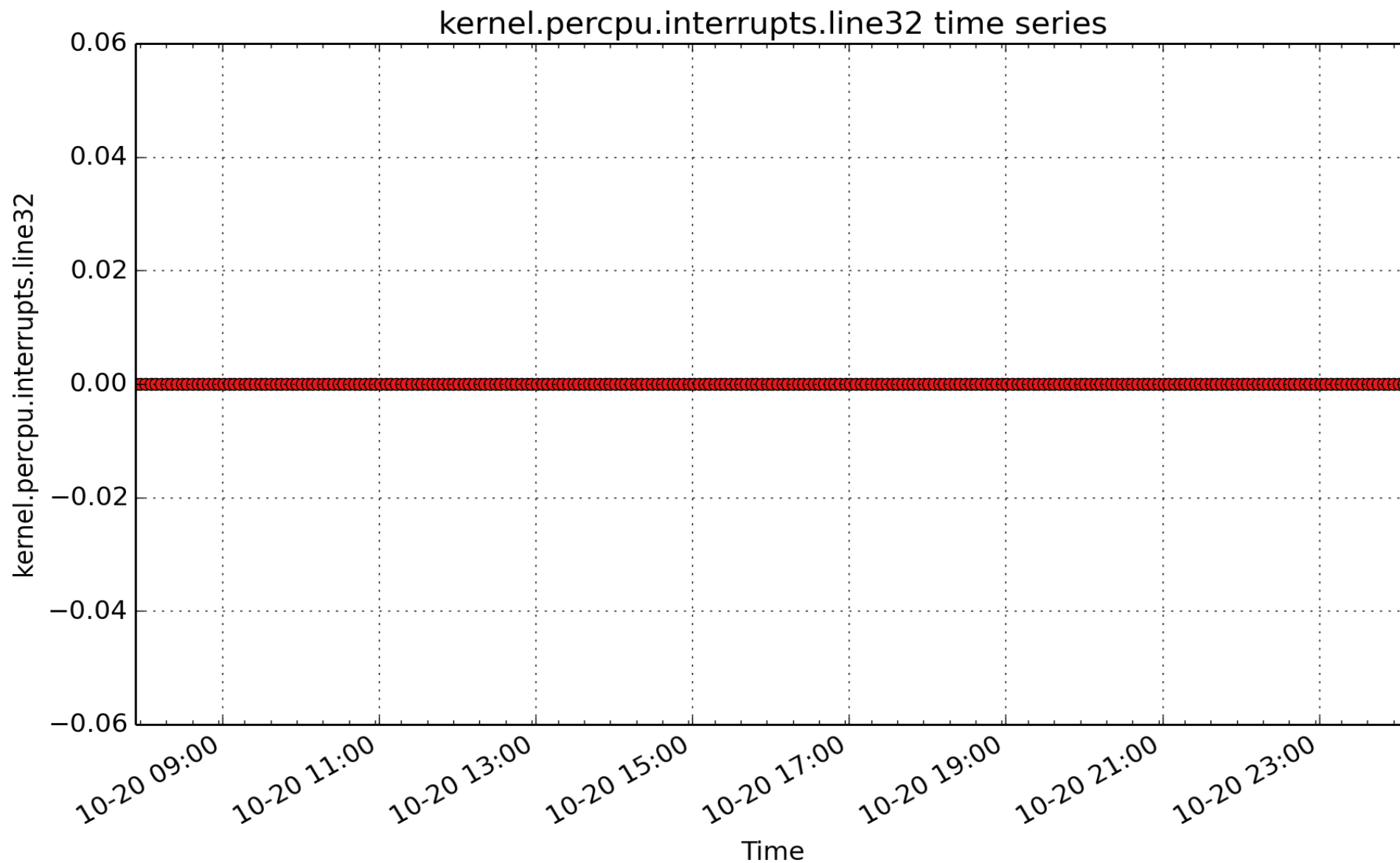
kernel.percpu.interrupts.line26: PCI-MSI-edge ahci (count - U32) - *rate converted*



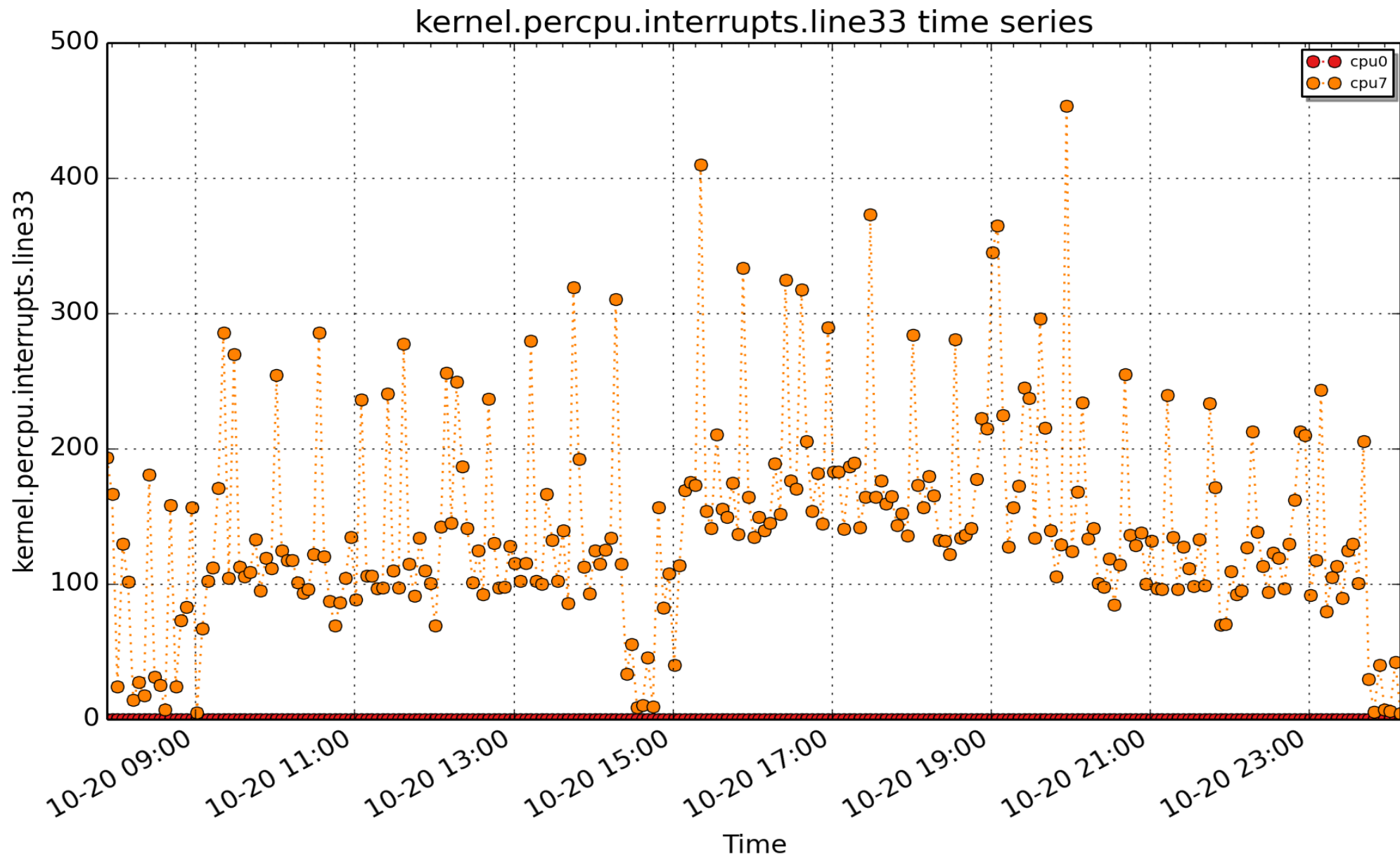
kernel.percpu.interrupts.line30: PCI-MSI-edge nouveau (count - U32) - *rate converted*



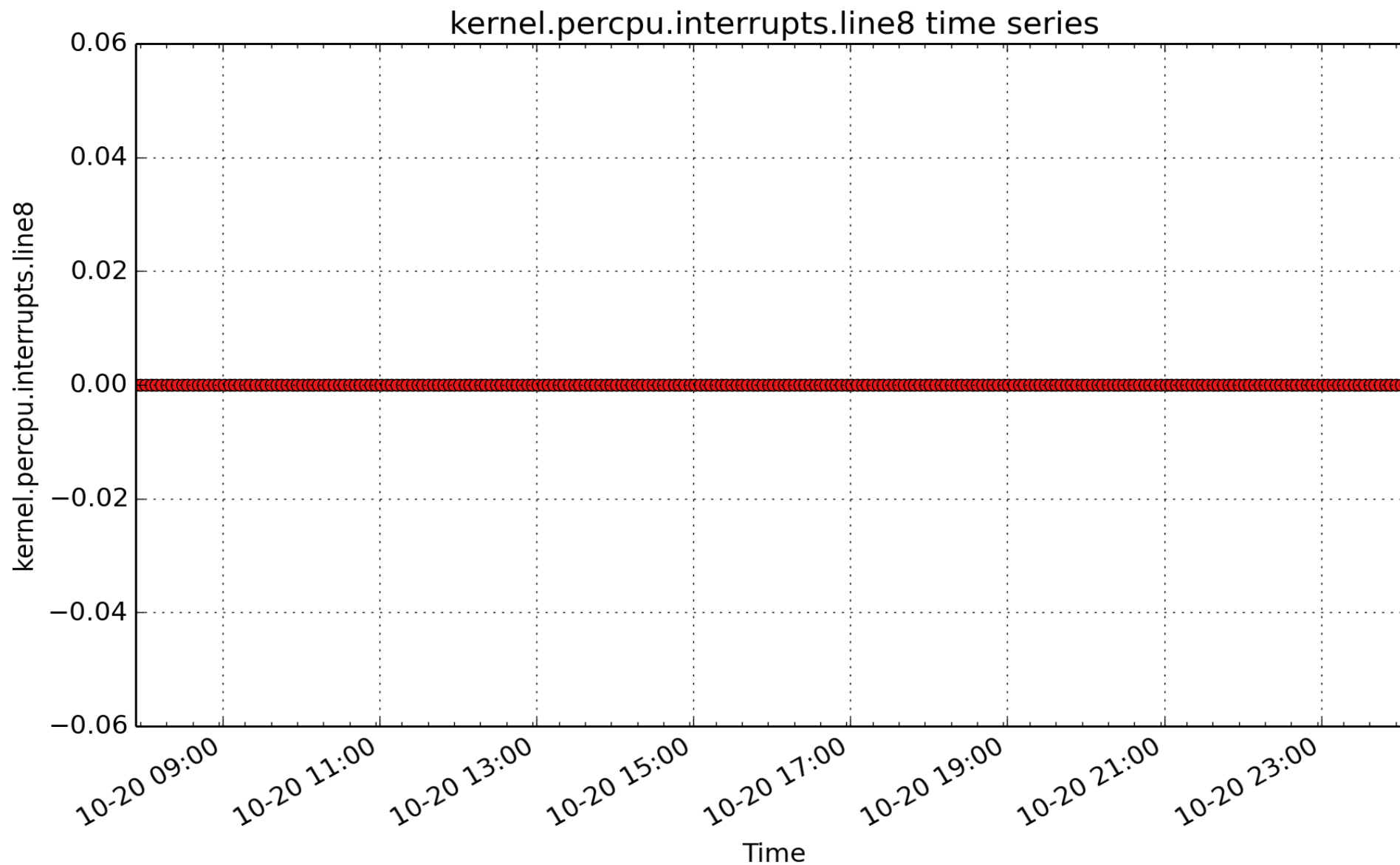
kernel.percpu.interrupts.line31: PCI-MSI-edge snd_hda_intel (count - U32) - *rate converted*



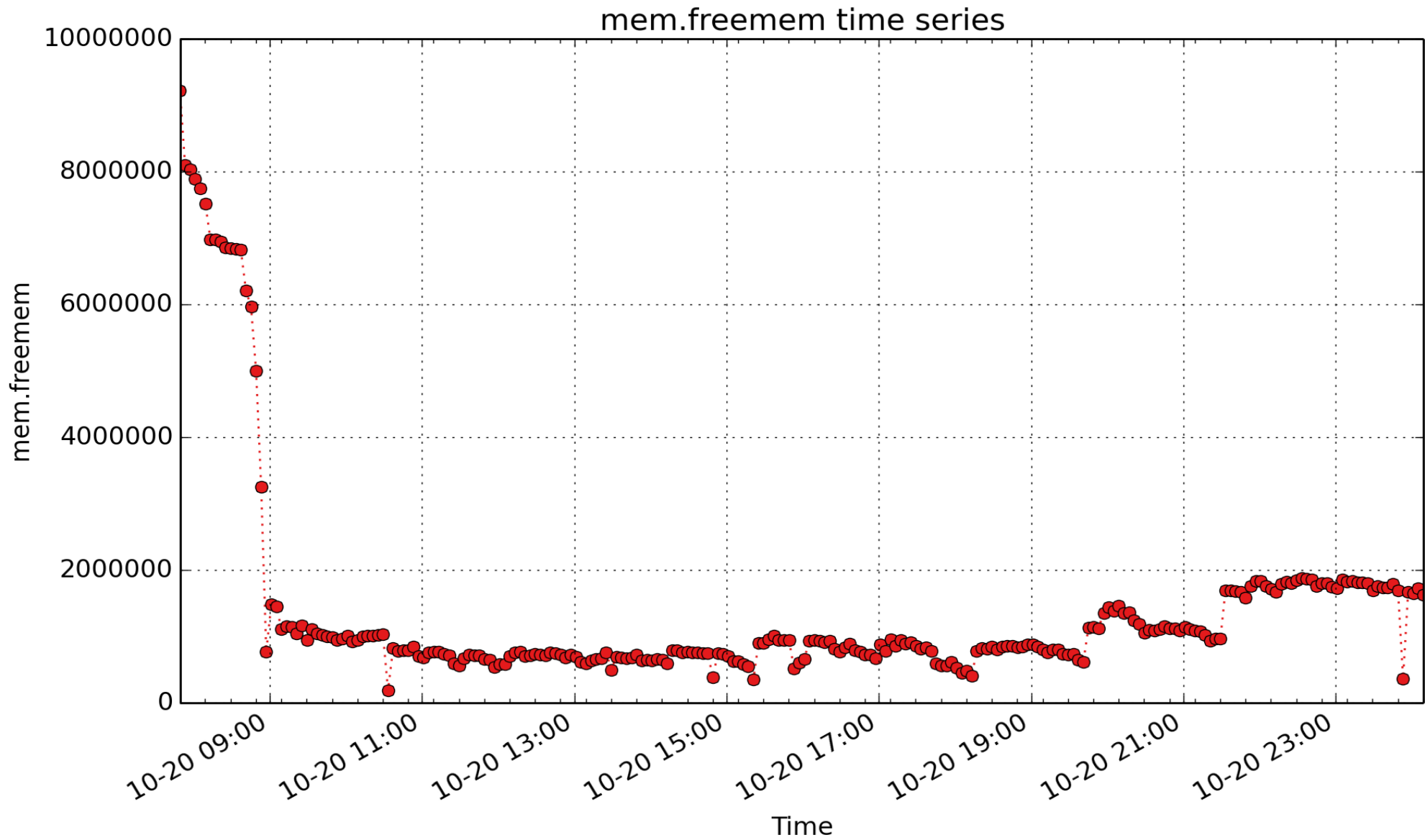
kernel.percpu.interrupts.line32: IO-APIC 10-fasteoi snd_hda_intel (count - U32) - *rate converted*



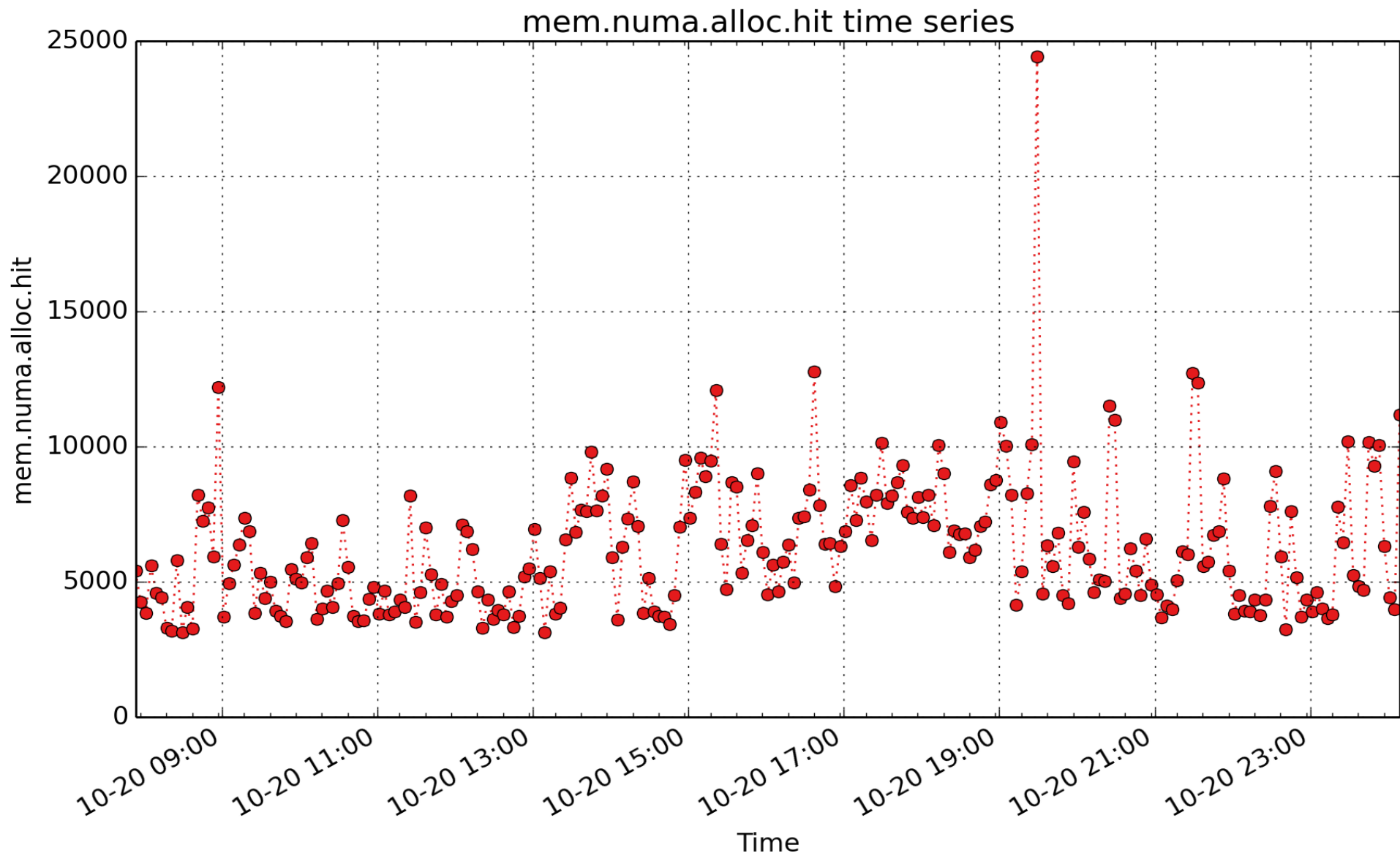
kernel.percpu.interrupts.line33: PCI-MSI-edge em1 (count - U32) - *rate converted*



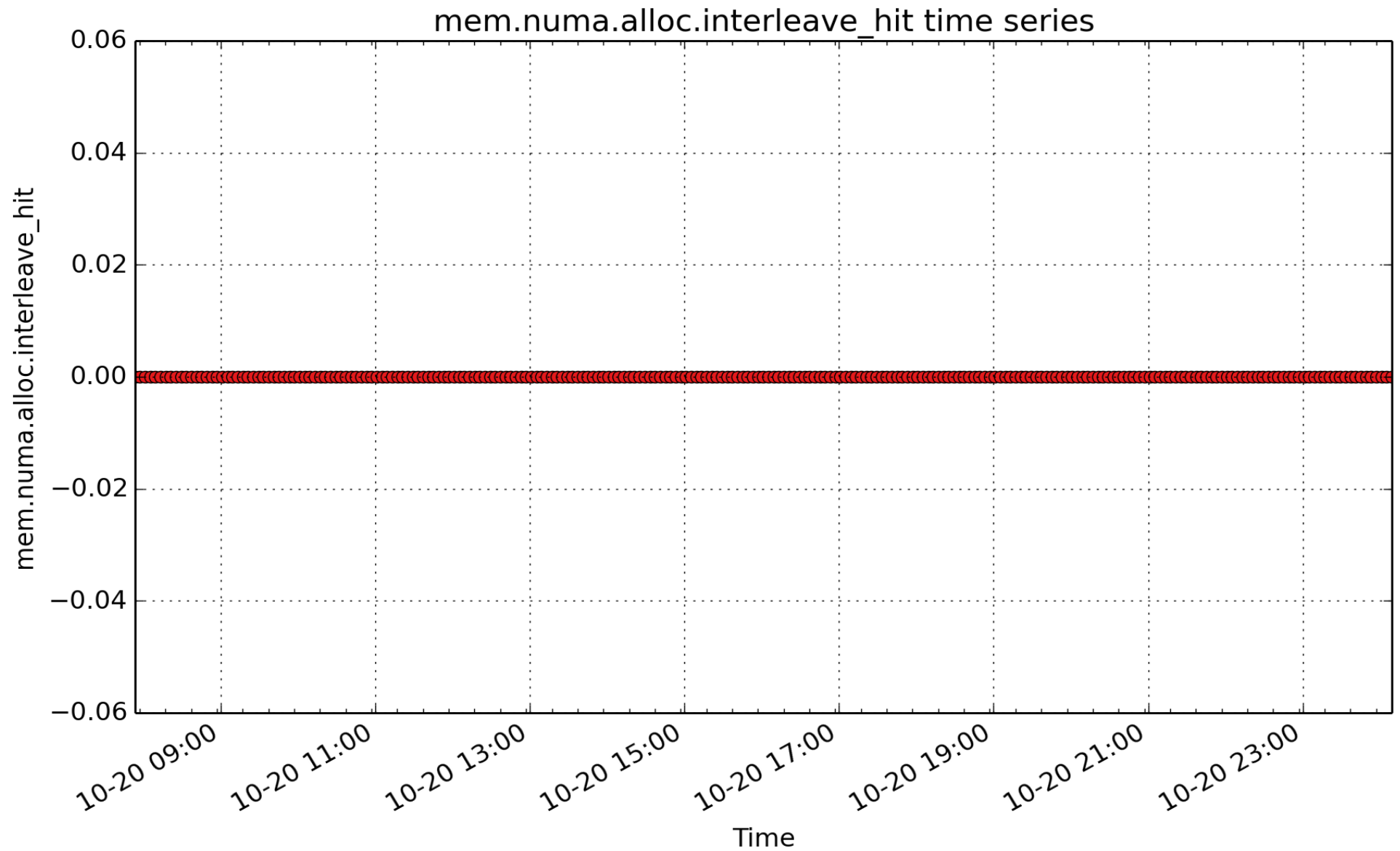
kernel.percpu.interrupts.line8: IO-APIC-edge rtc0 (count - U32) - *rate converted*



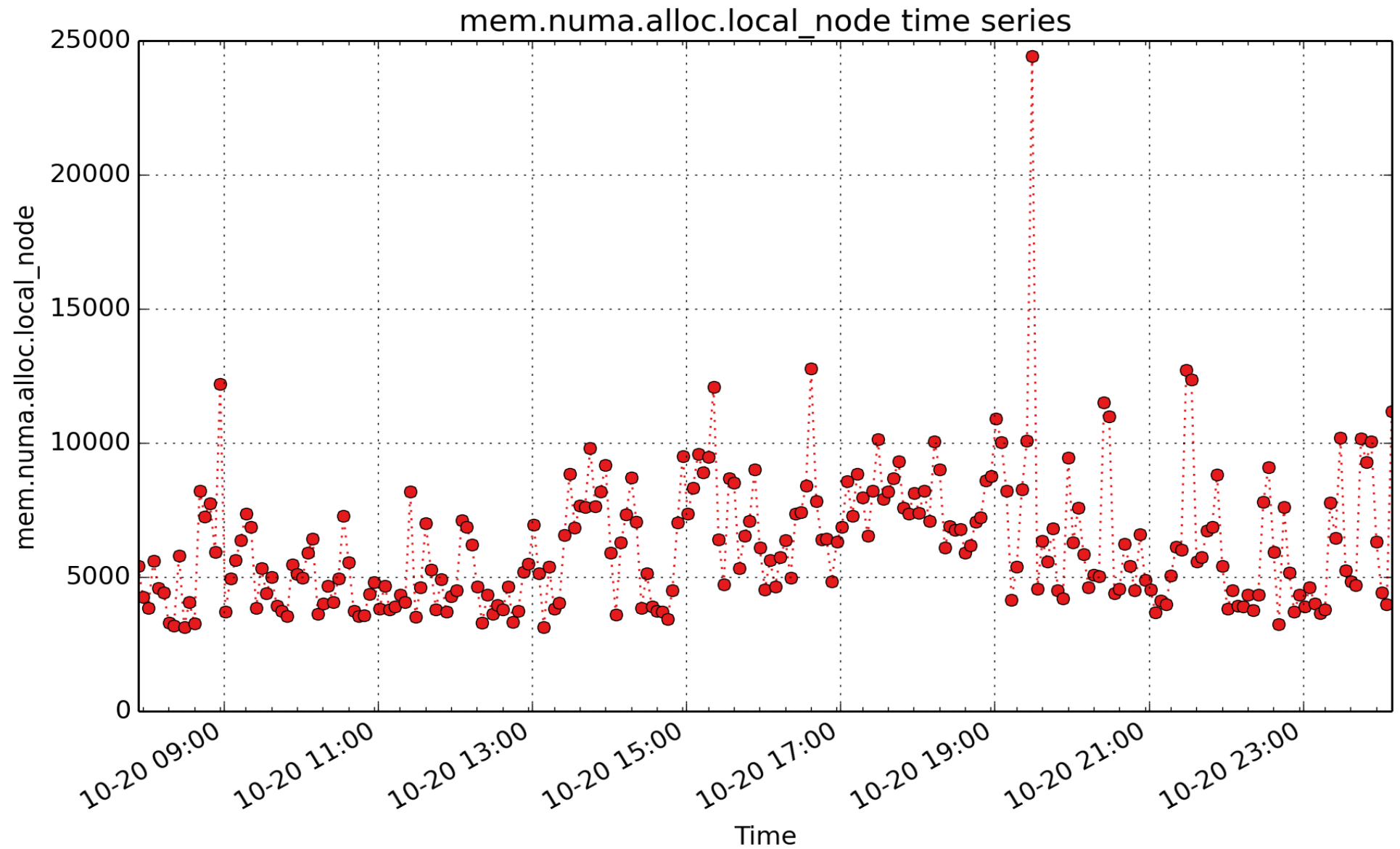
mem.freemem: free system memory metric from /proc/meminfo (Kbyte - U64)



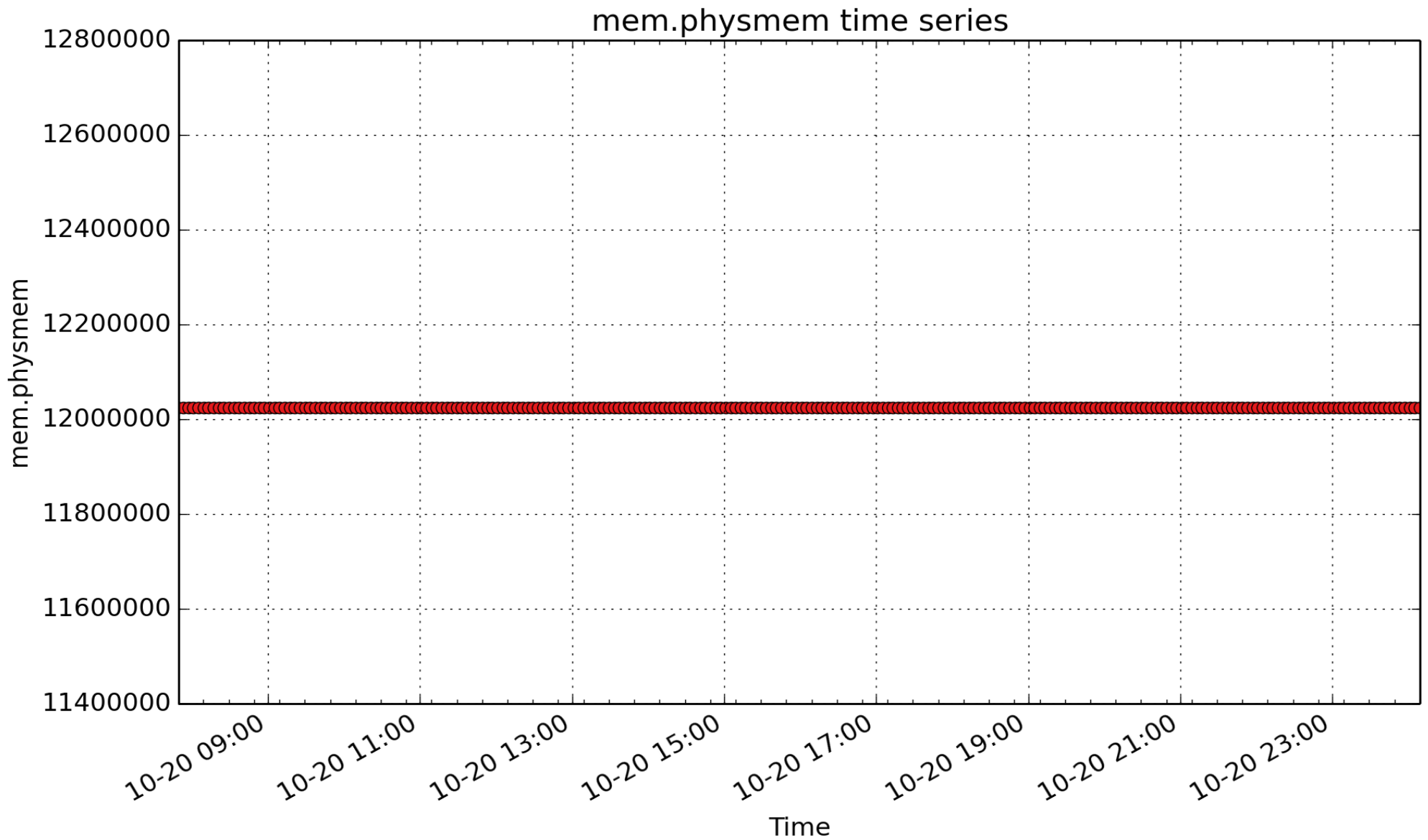
mem.numa.alloc.hit: per-node count of times a task wanted alloc on local node and succeeded (count - U64)
- *rate converted*



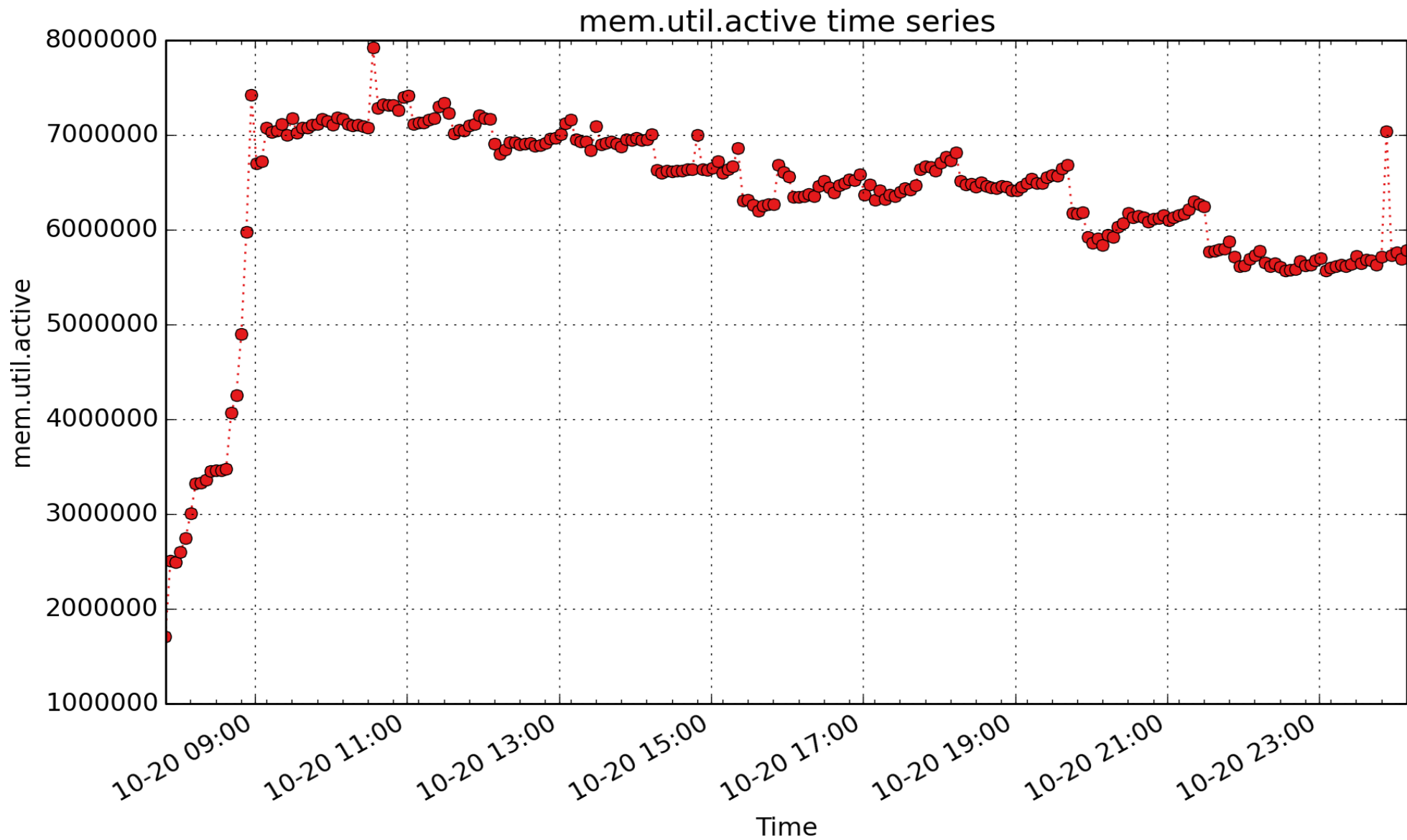
mem.numa.alloc.interleave_hit: count of times interleaving wanted to allocate on this node and succeeded
(count - U64) - *rate converted*



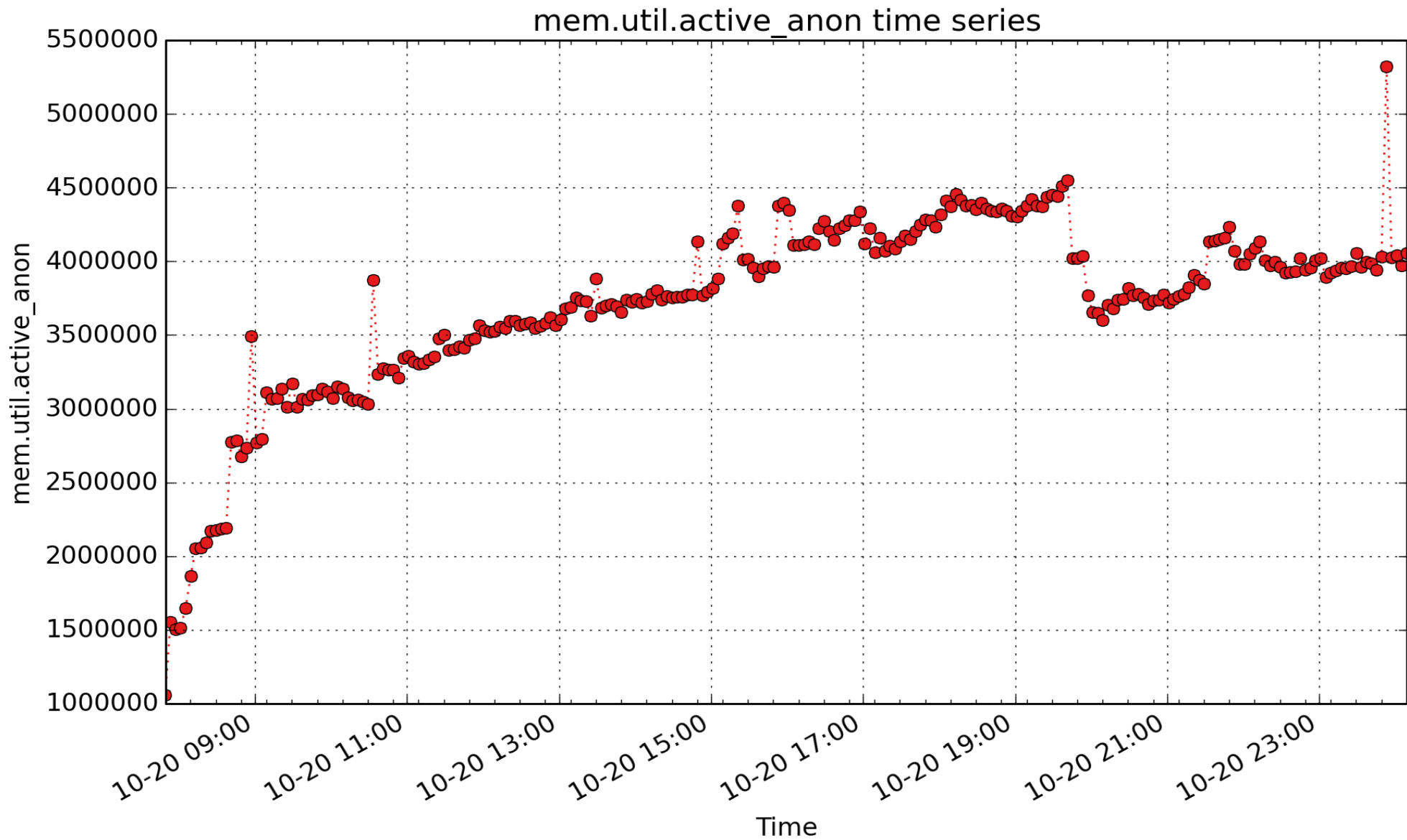
mem.numa.alloc.local_node: count of times a process ran on this node and got memory on this node (count - U64) - *rate converted*



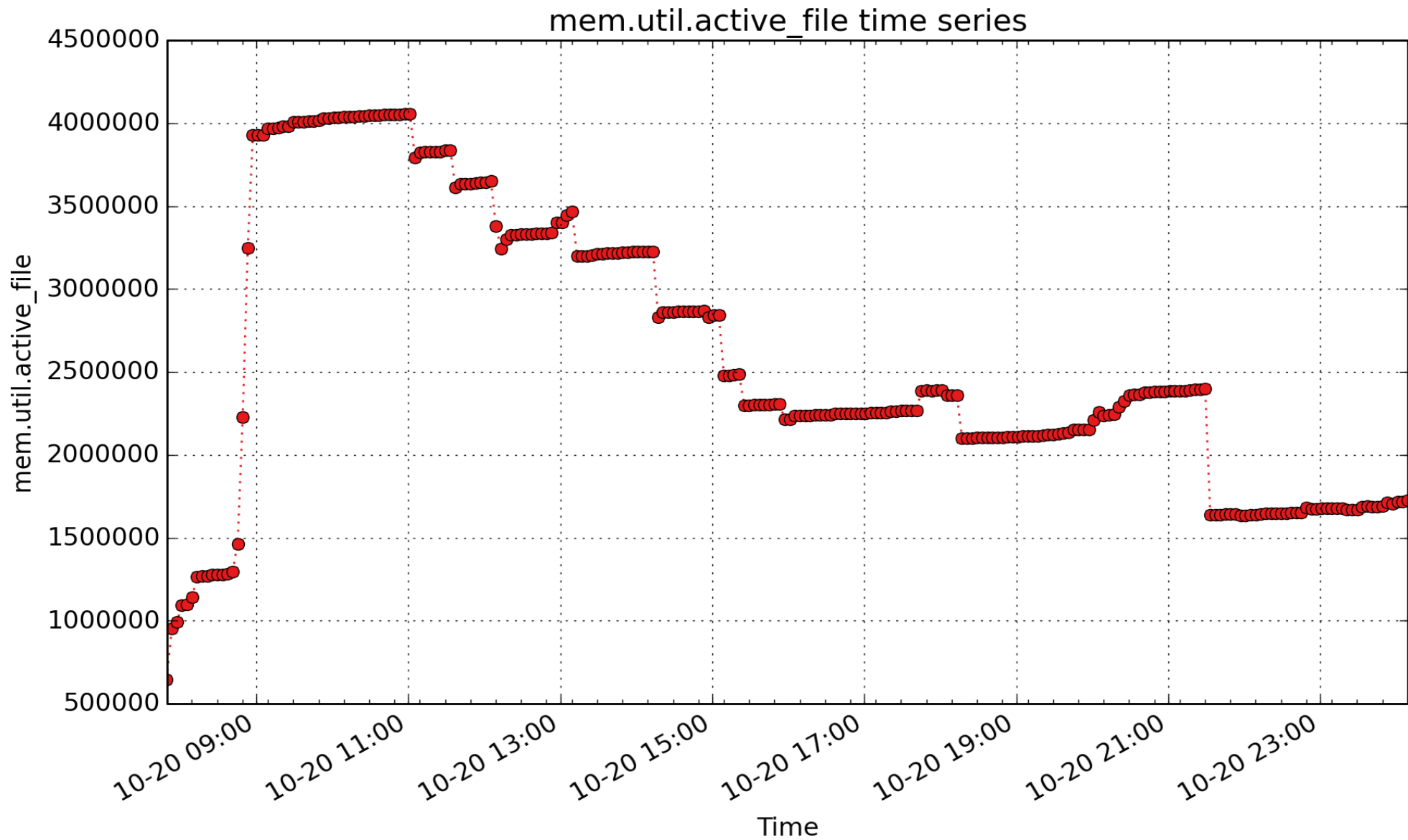
mem.physmem: The value of this metric corresponds to the "MemTotal" field reported by `/proc/meminfo`. Note that this does not necessarily correspond to actual installed physical memory - there may be areas of the physical address space mapped as ROM in various peripheral devices and the bios may be mirroring certain ROMs in RAM. (Kbyte - U64)



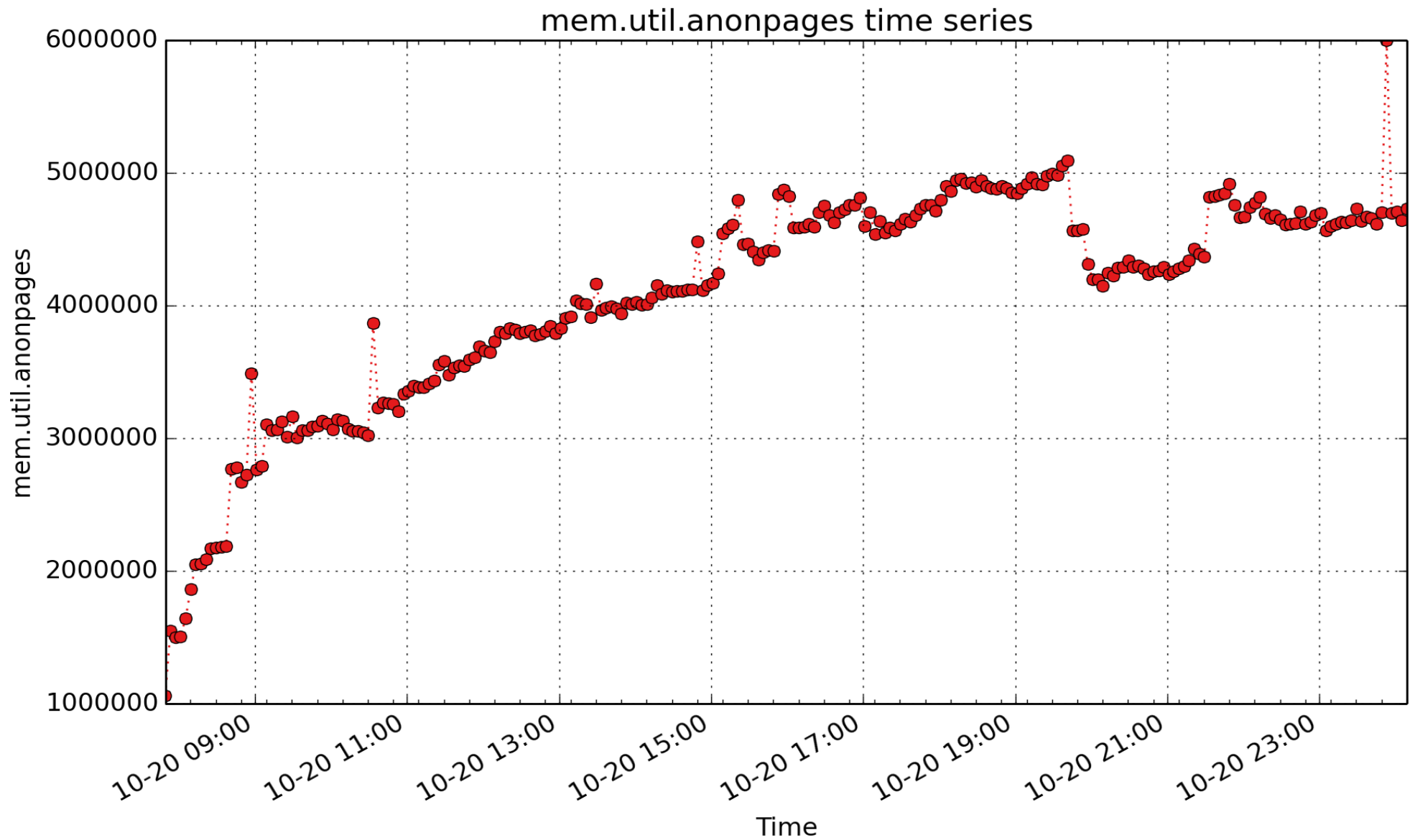
mem.util.active: Memory that has been used more recently and usually not reclaimed unless absolutely necessary. (Kbyte - U64)



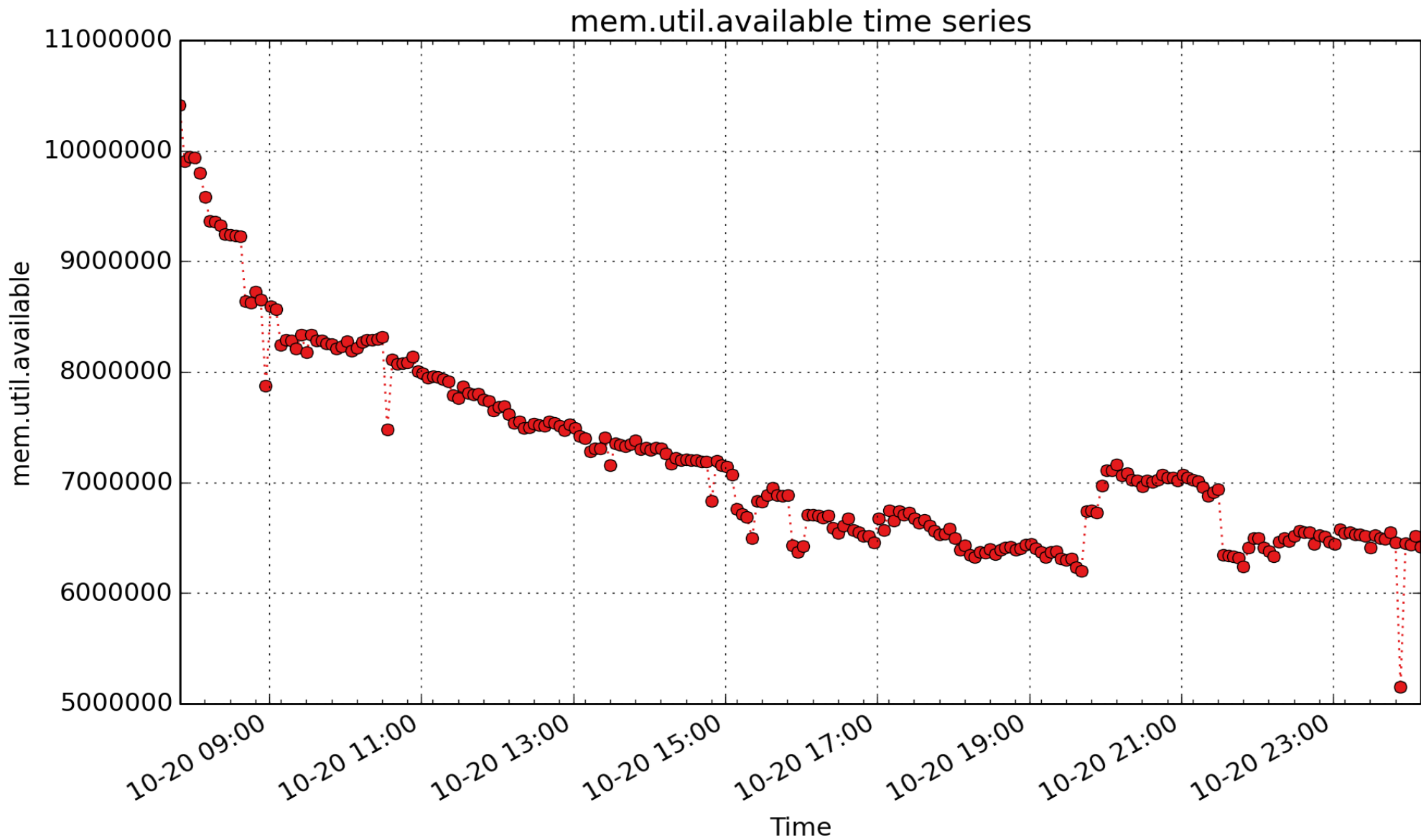
mem.util.active_anon: anonymous Active list LRU memory (Kbyte - U64)



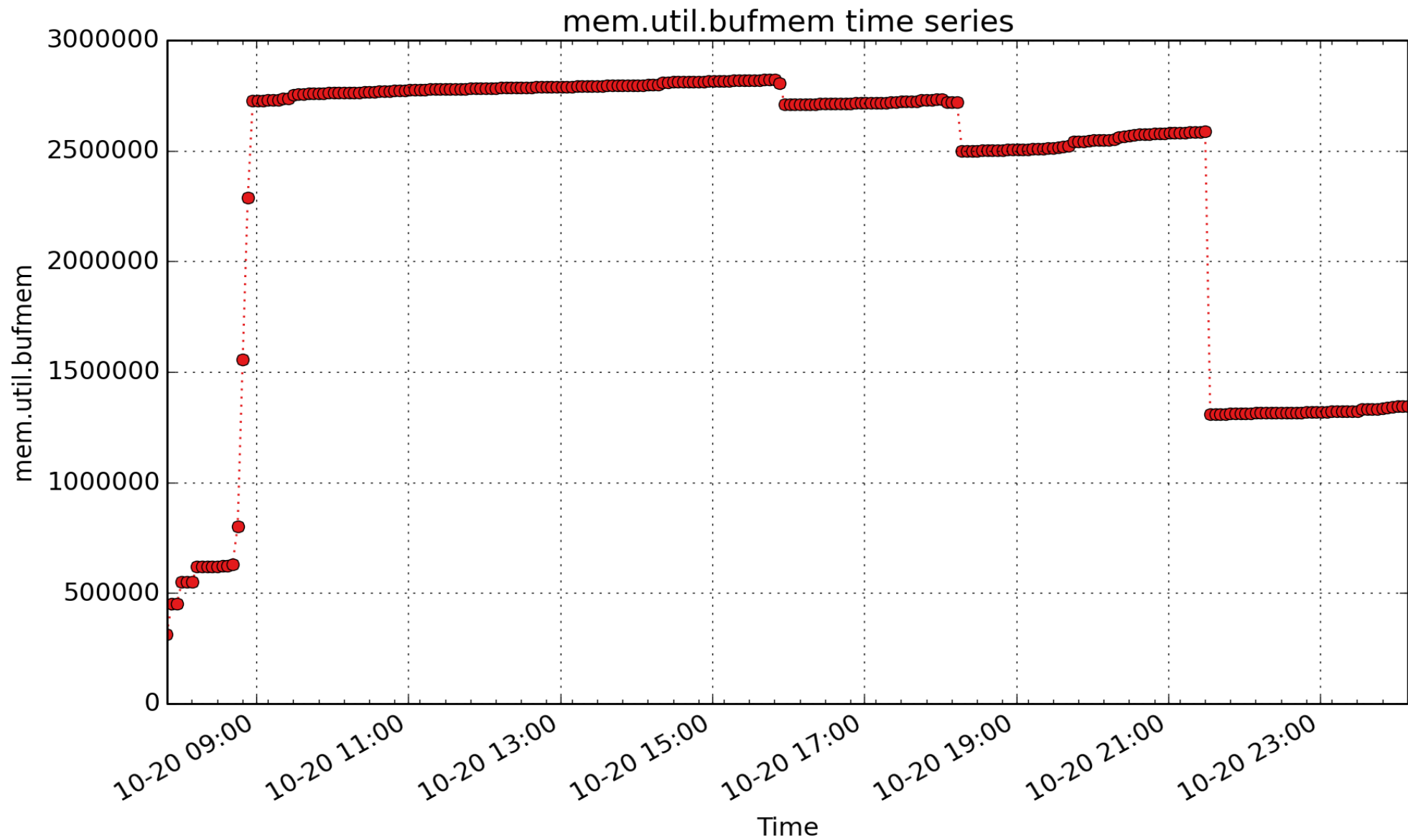
mem.util.active_file: file-backed Active list LRU memory (Kbyte - U64)



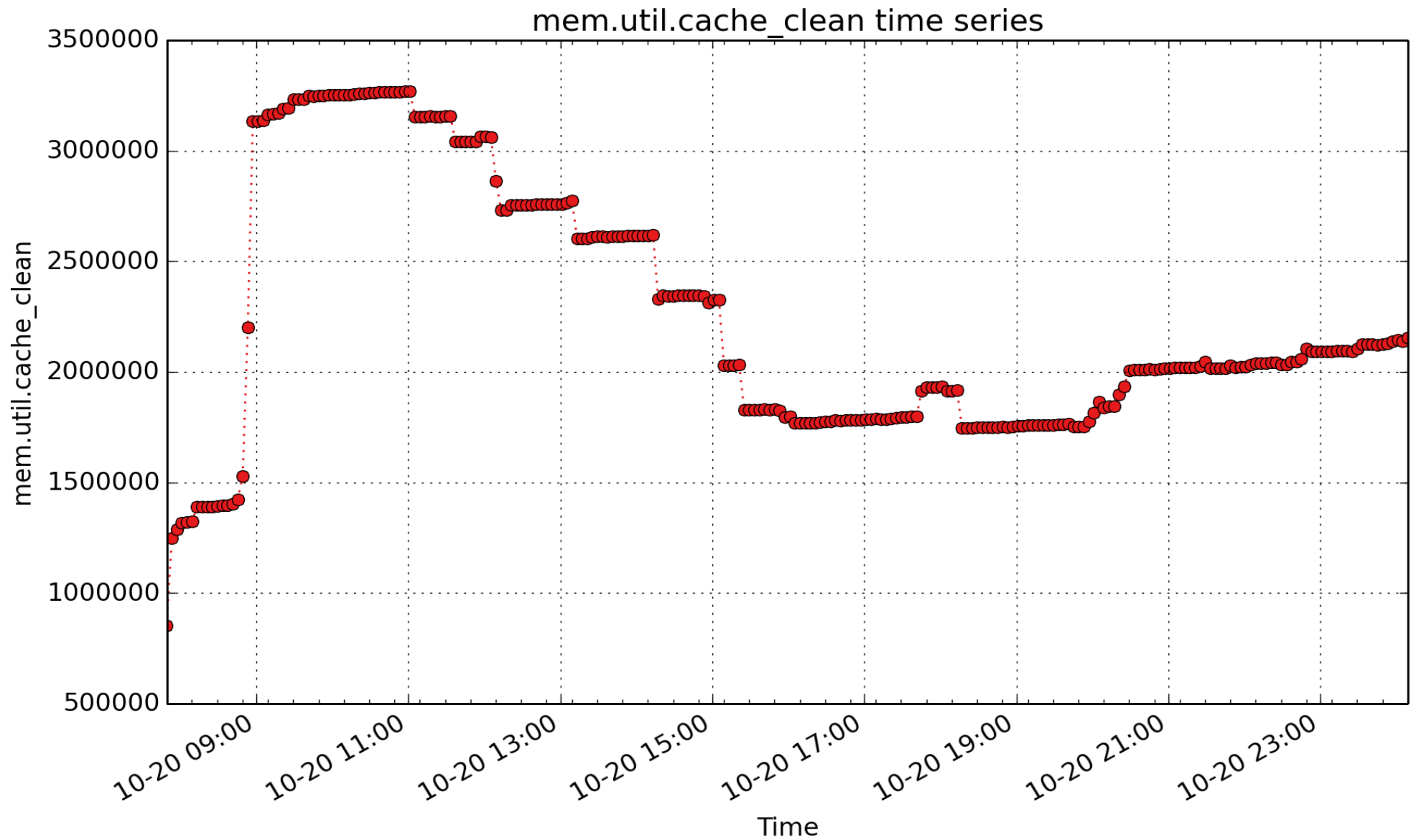
mem.util.anonpages: Kbytes in user pages not backed by files, from /proc/meminfo (Kbyte - U64)



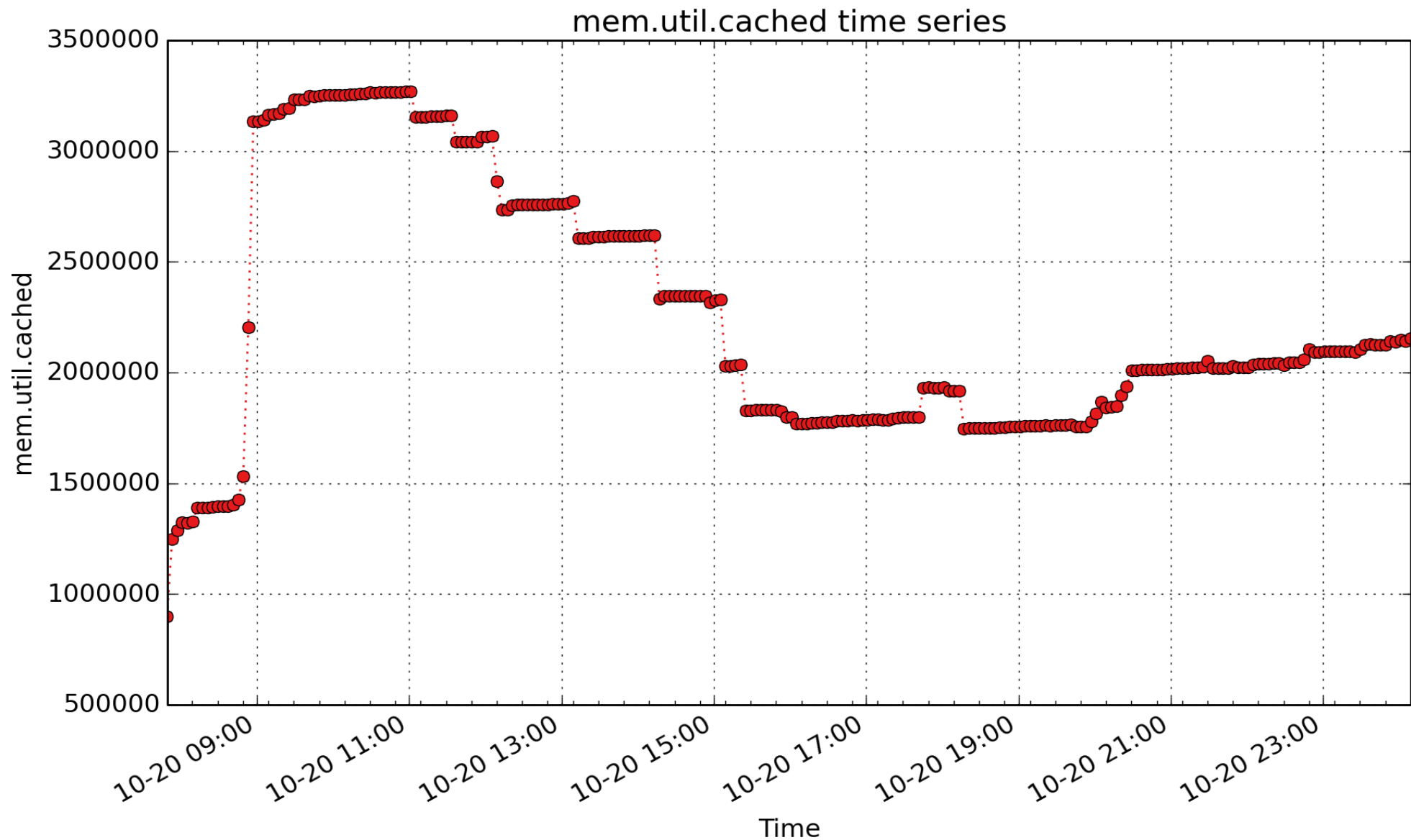
mem.util.available: The amount of memory that is available for a new workload, without pushing the system into swap. Estimated from MemFree, Active(file), Inactive(file), and SReclaimable, as well as the "low" watermarks from /proc/zoneinfo. (Kbyte - U64)



mem.util.bufmem: Memory allocated for buffer_heads. (Kbyte - U64)

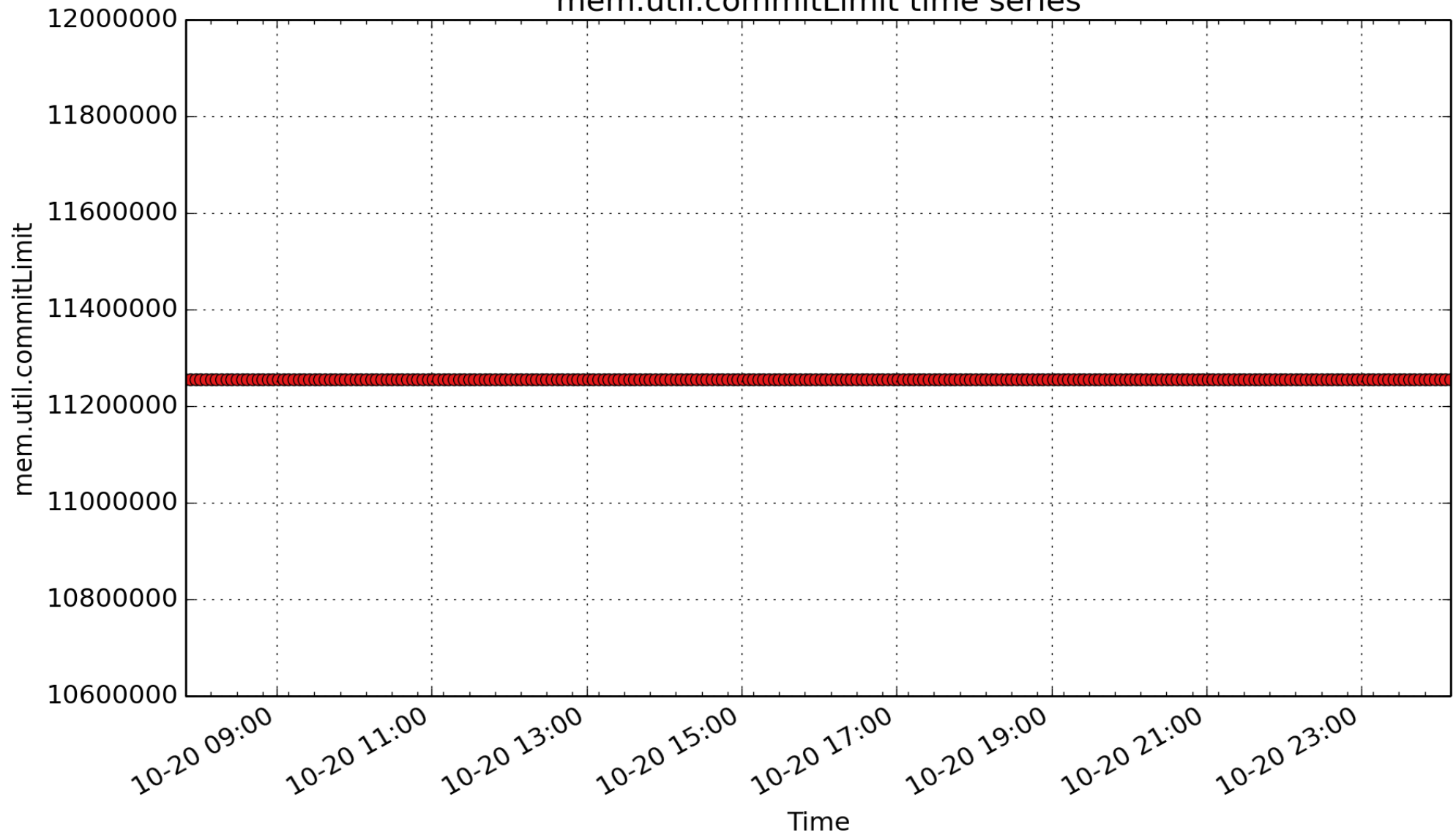


mem.util.cache_clean: Kbytes cached and not dirty or writeback, derived from /proc/meminfo (Kbyte - U64)

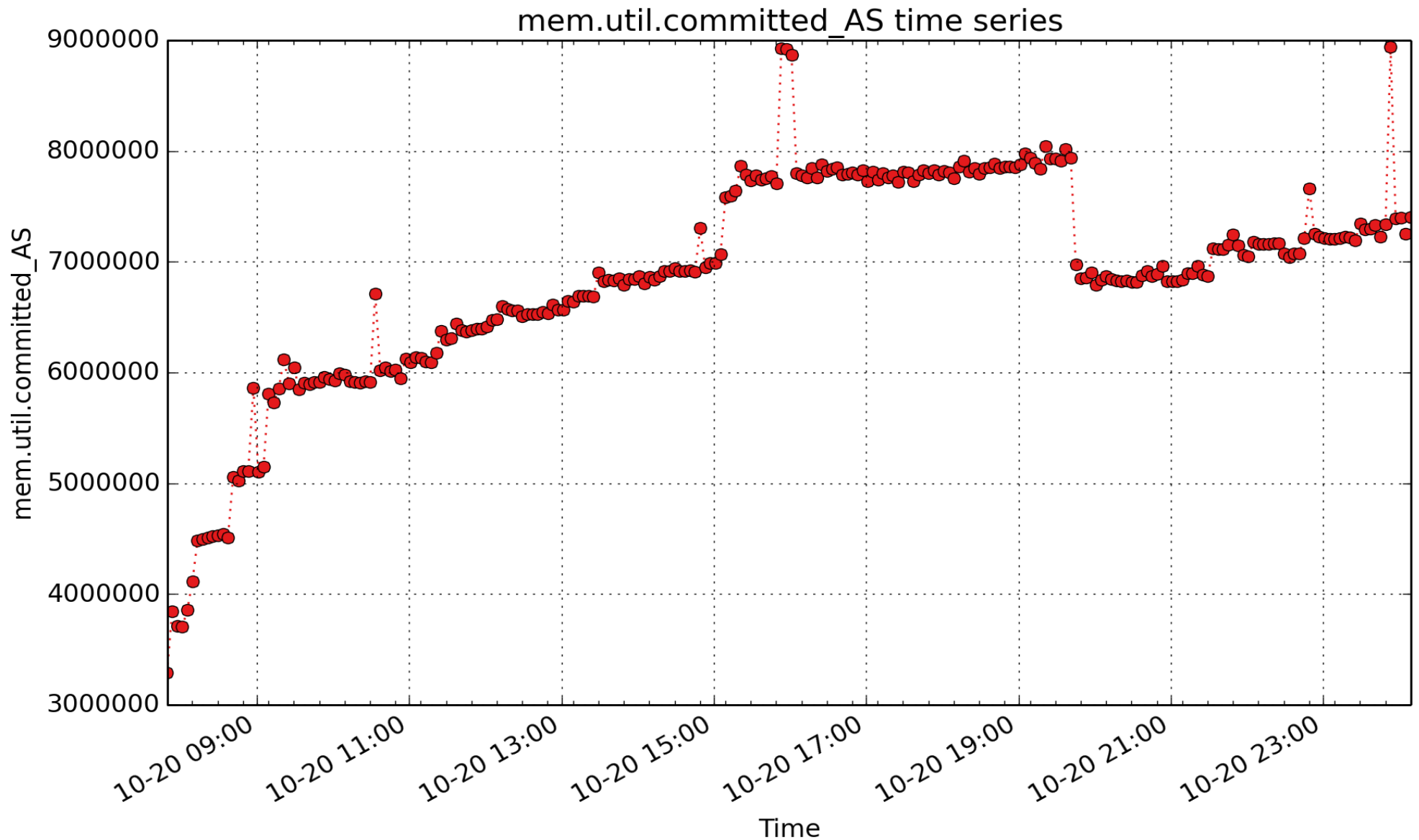


mem.util.cached: Memory used by the page cache, including buffered file data. This is in-memory cache for files read from the disk (the pagecache) but doesn't include SwapCached. (Kbyte - U64)

mem.util.commitLimit time series

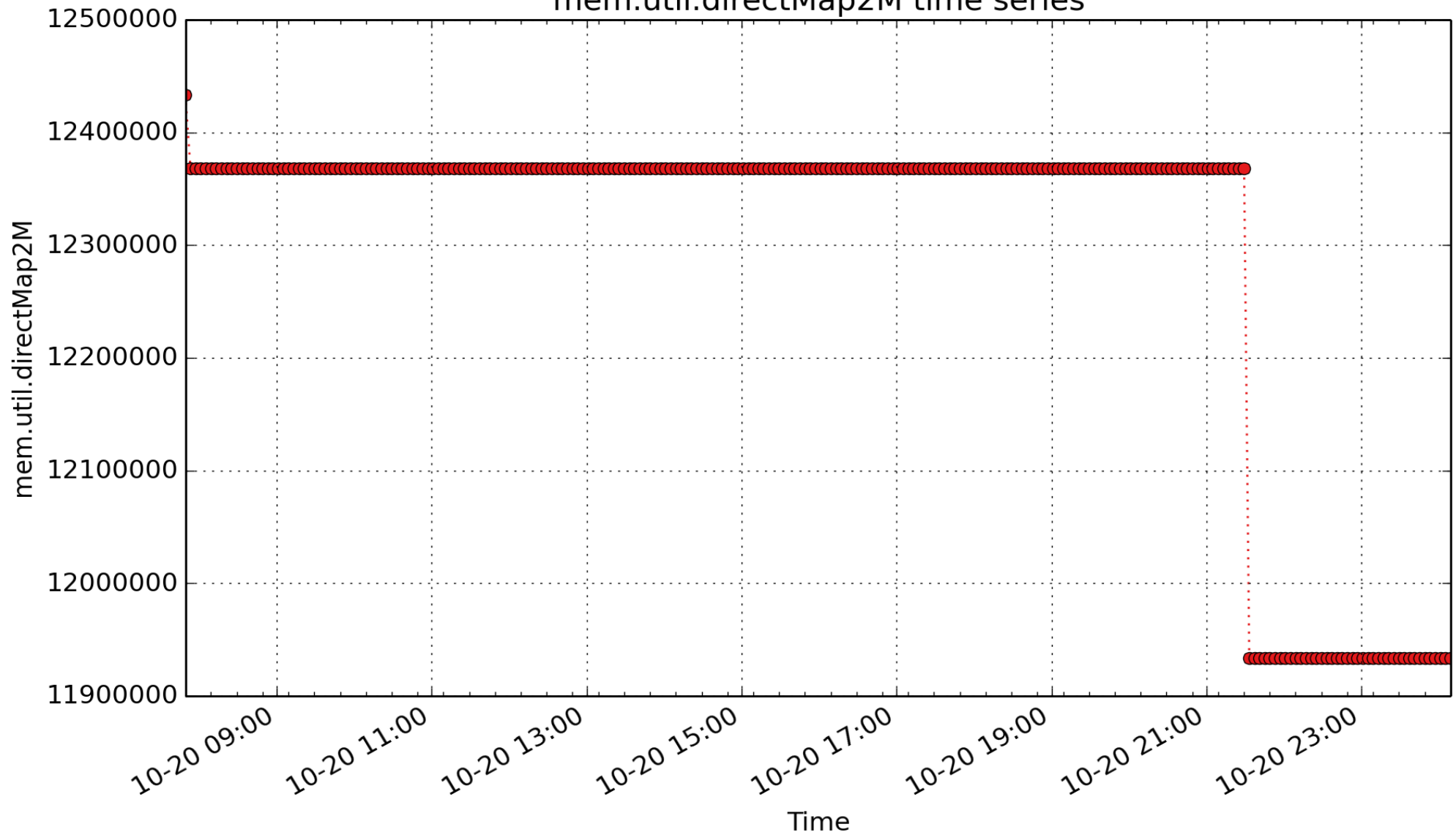


mem.util.commitLimit: The static total, in Kbytes, available for commitment to address spaces. Thus, mem.util.committed_AS may range up to this total. Normally the kernel overcommits memory, so this value may exceed mem.physmem (Kbyte - U64)

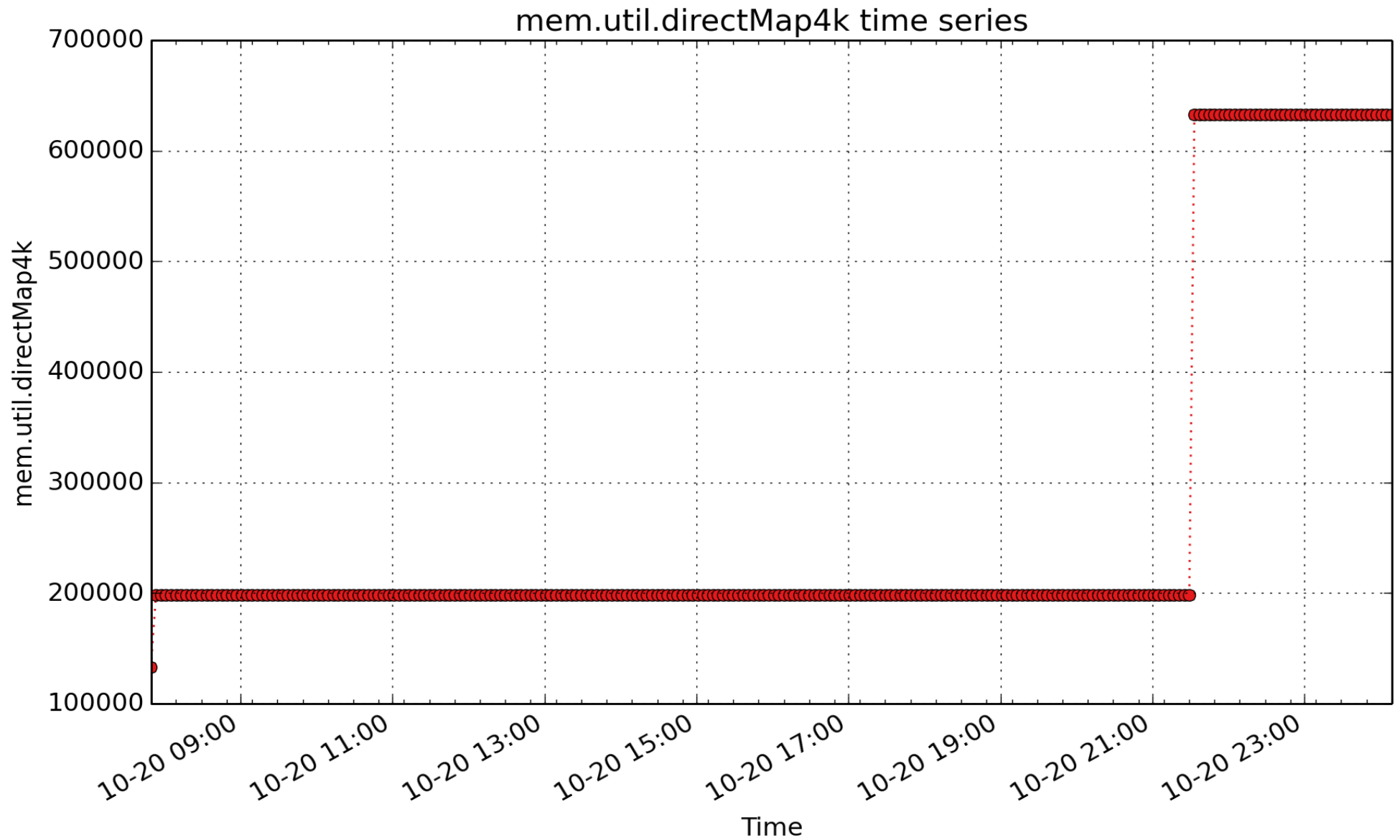


mem.util.committed_AS: An estimate of how much RAM you would need to make a 99.99% guarantee that there never is OOM (out of memory) for this workload. Normally the kernel will overcommit memory. That means, say you do a 1GB malloc, nothing happens, really. Only when you start USING that malloc memory you will get real memory on demand, and just as much as you use. (Kbyte - U64)

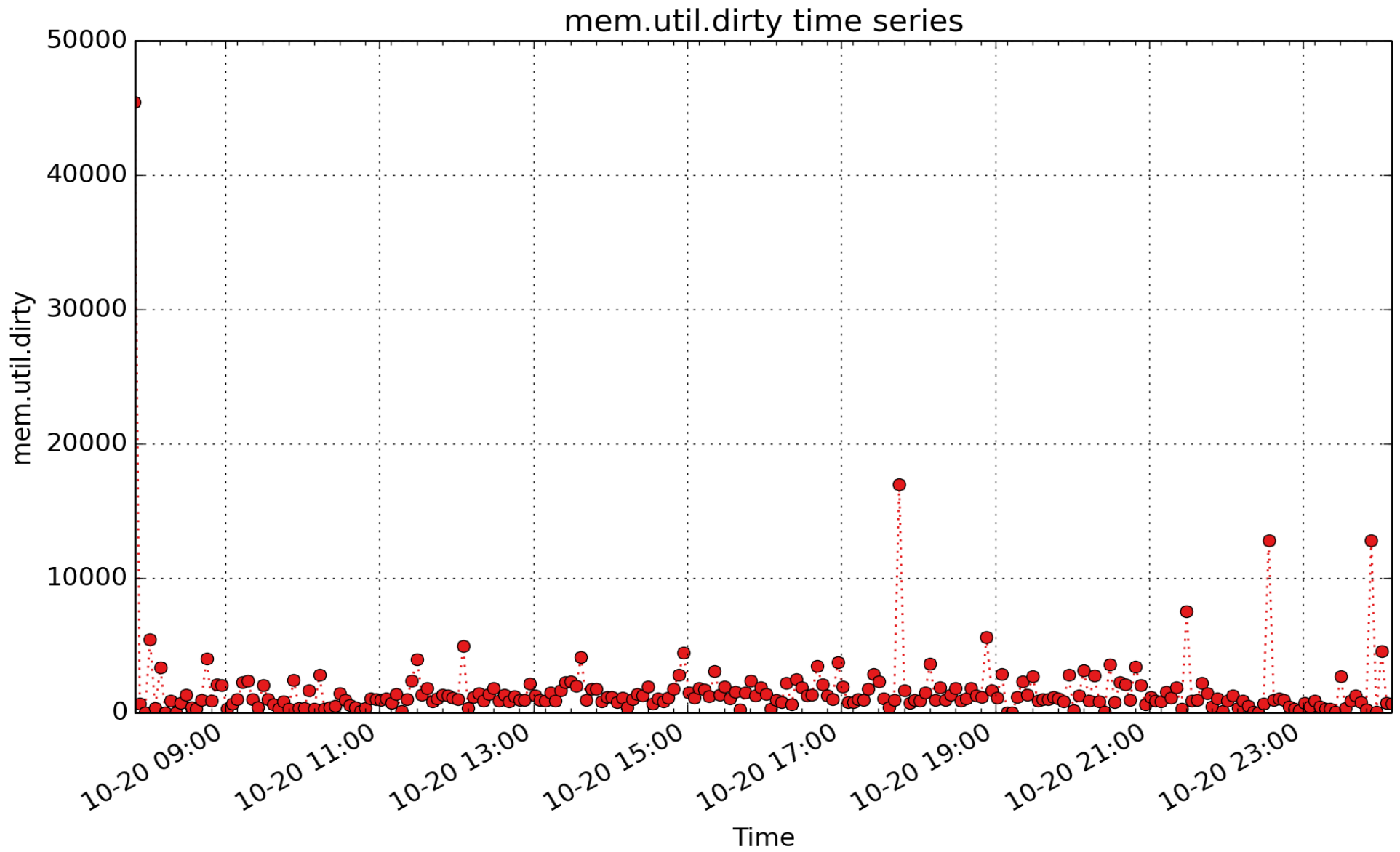
mem.util.directMap2M time series



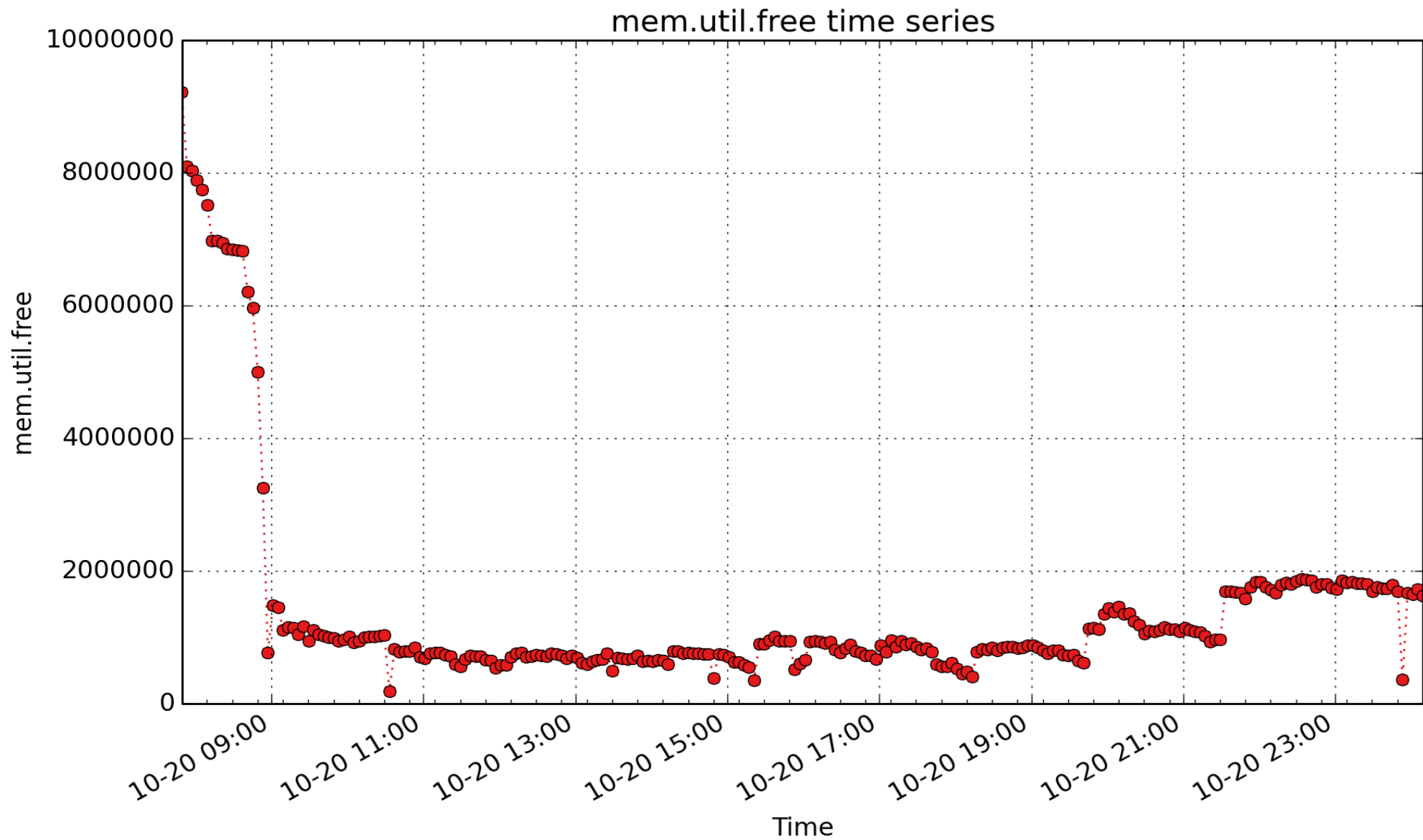
mem.util.directMap2M: amount of memory that is directly mapped in 2MB pages (Kbyte - U64)



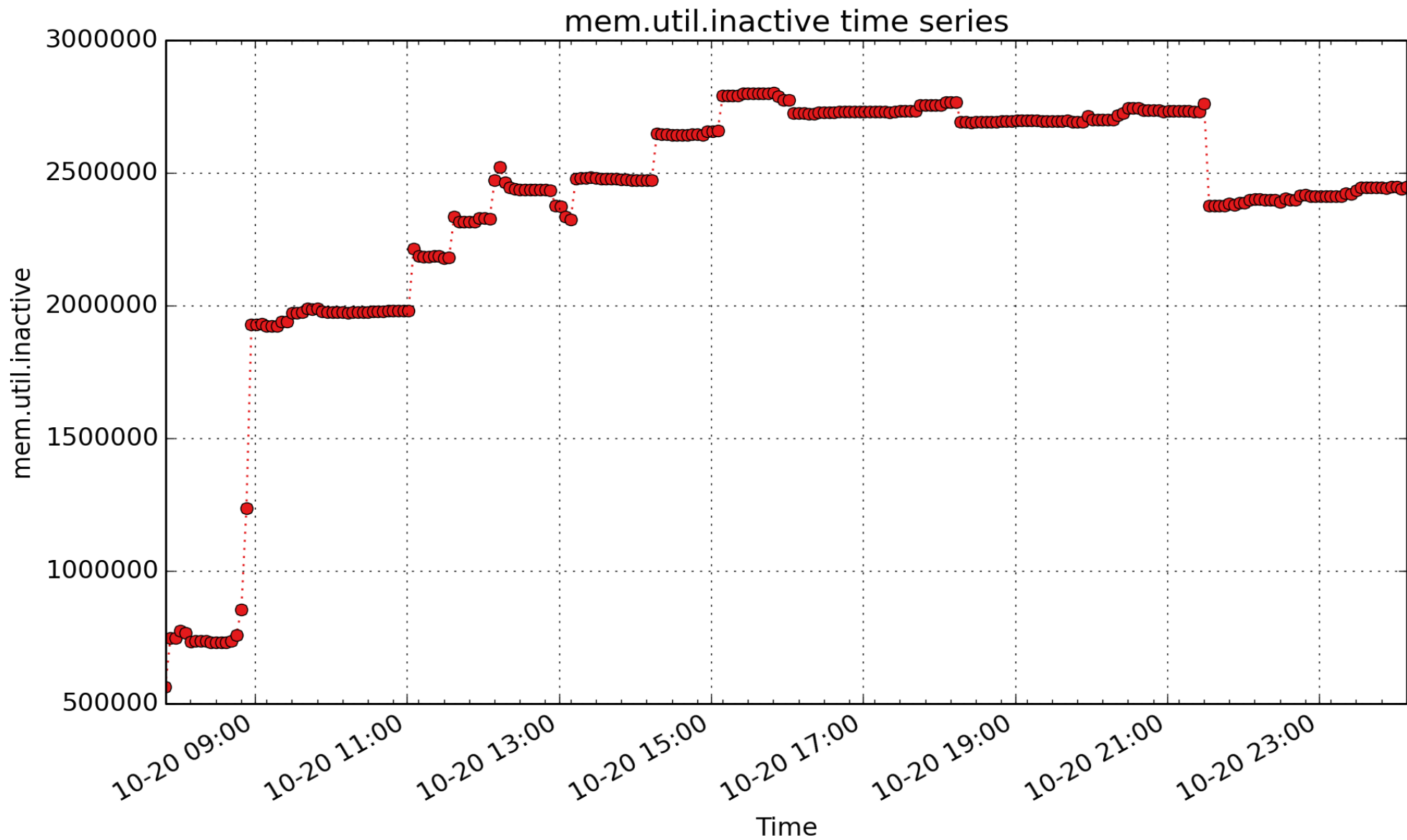
mem.util.directMap4k: amount of memory that is directly mapped in 4kB pages (Kbyte - U64)



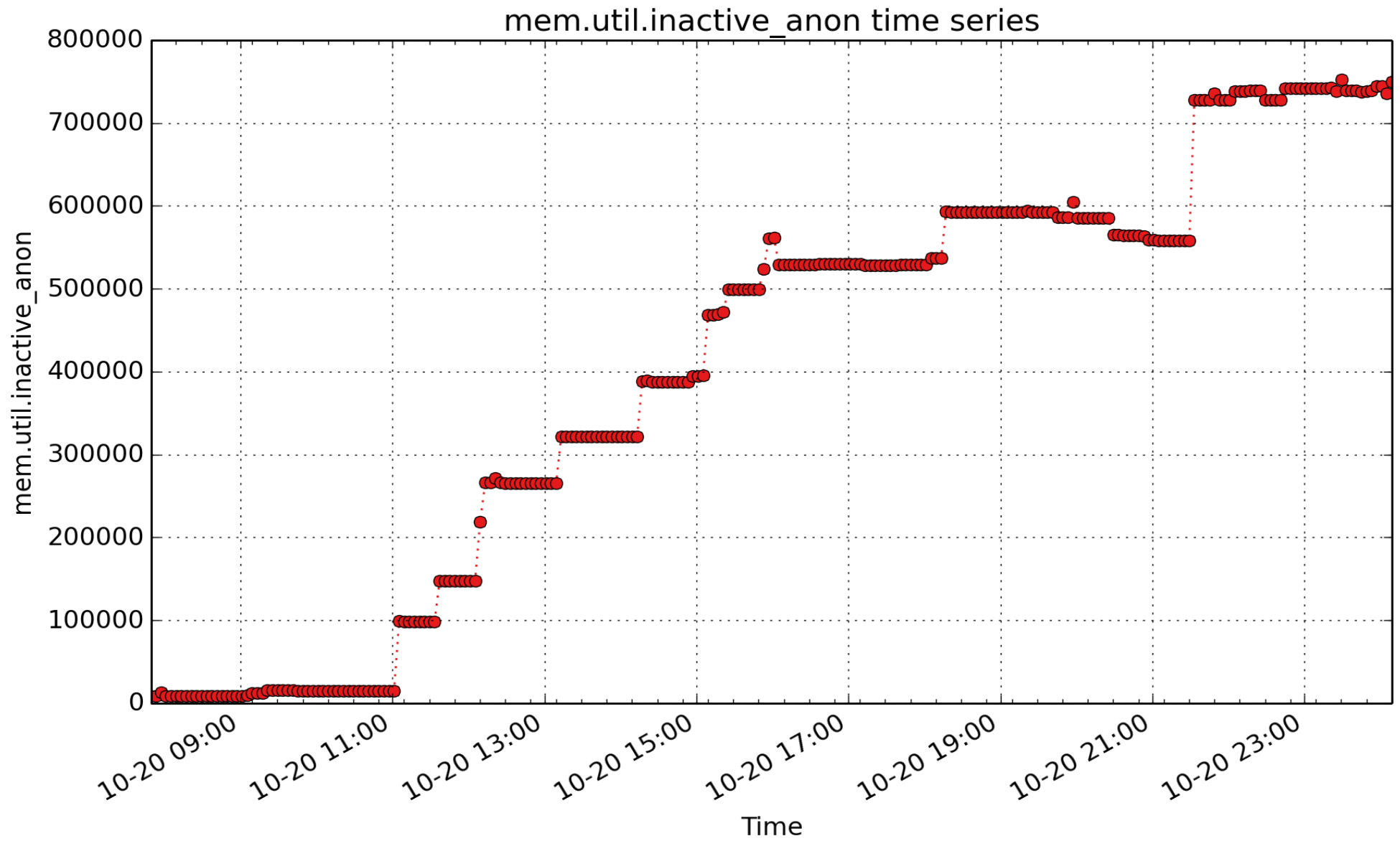
mem.util.dirty: Memory which is waiting to get written back to the disk (Kbyte - U64)



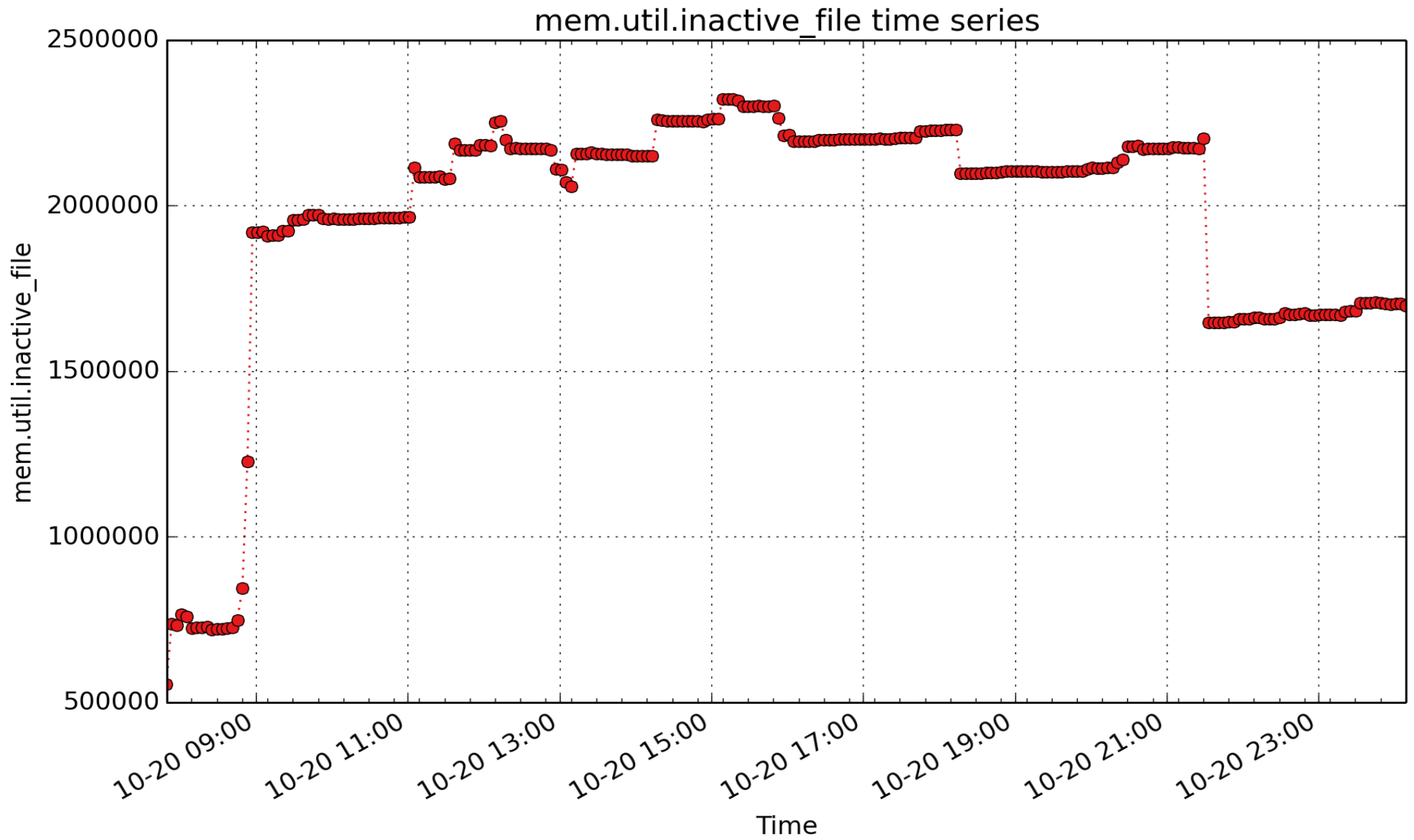
mem.util.free: Alias for mem.freemem. (Kbyte - U64)



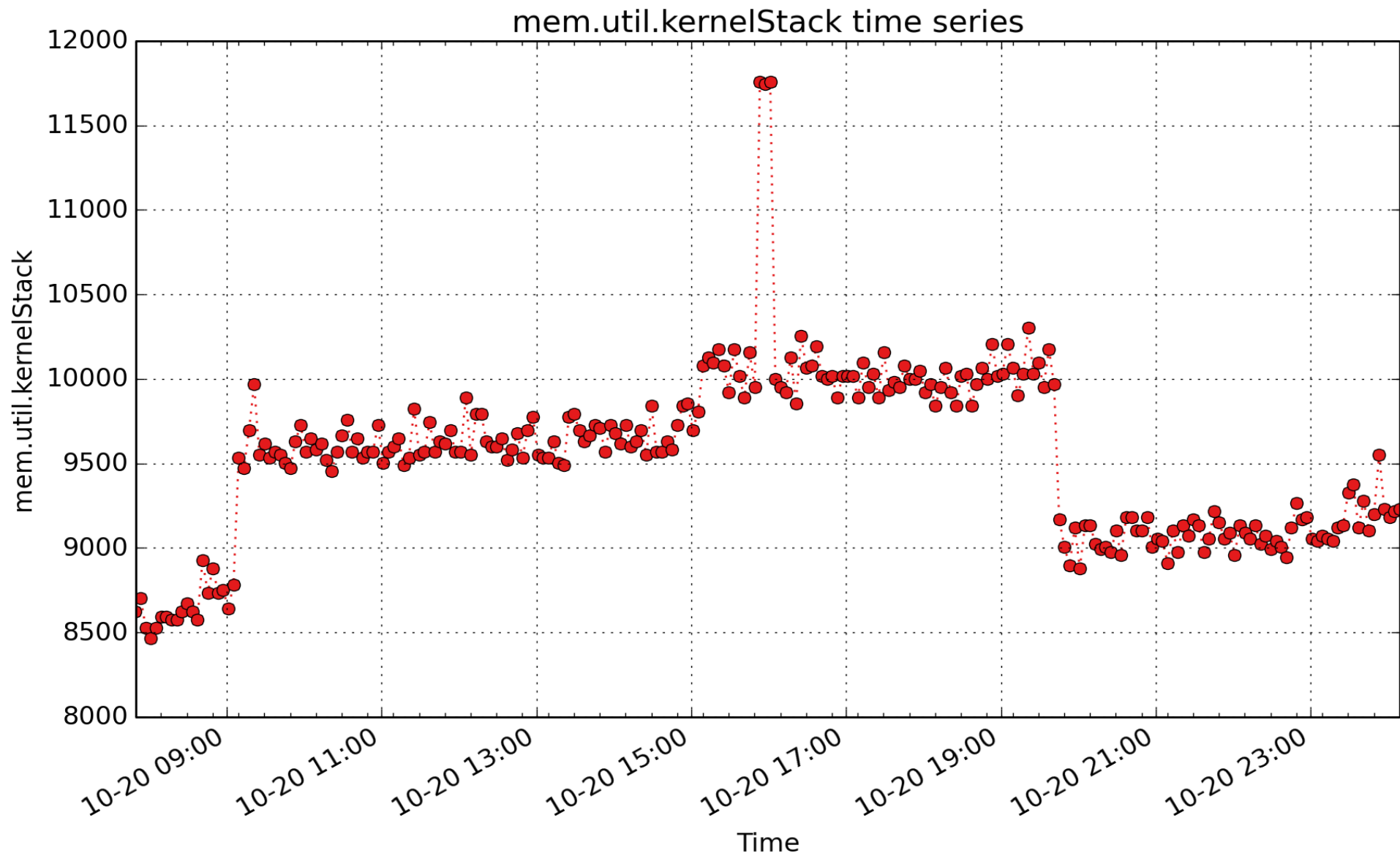
mem.util.inactive: Memory which has been less recently used. It is more eligible to be reclaimed for other purposes (Kbyte - U64)



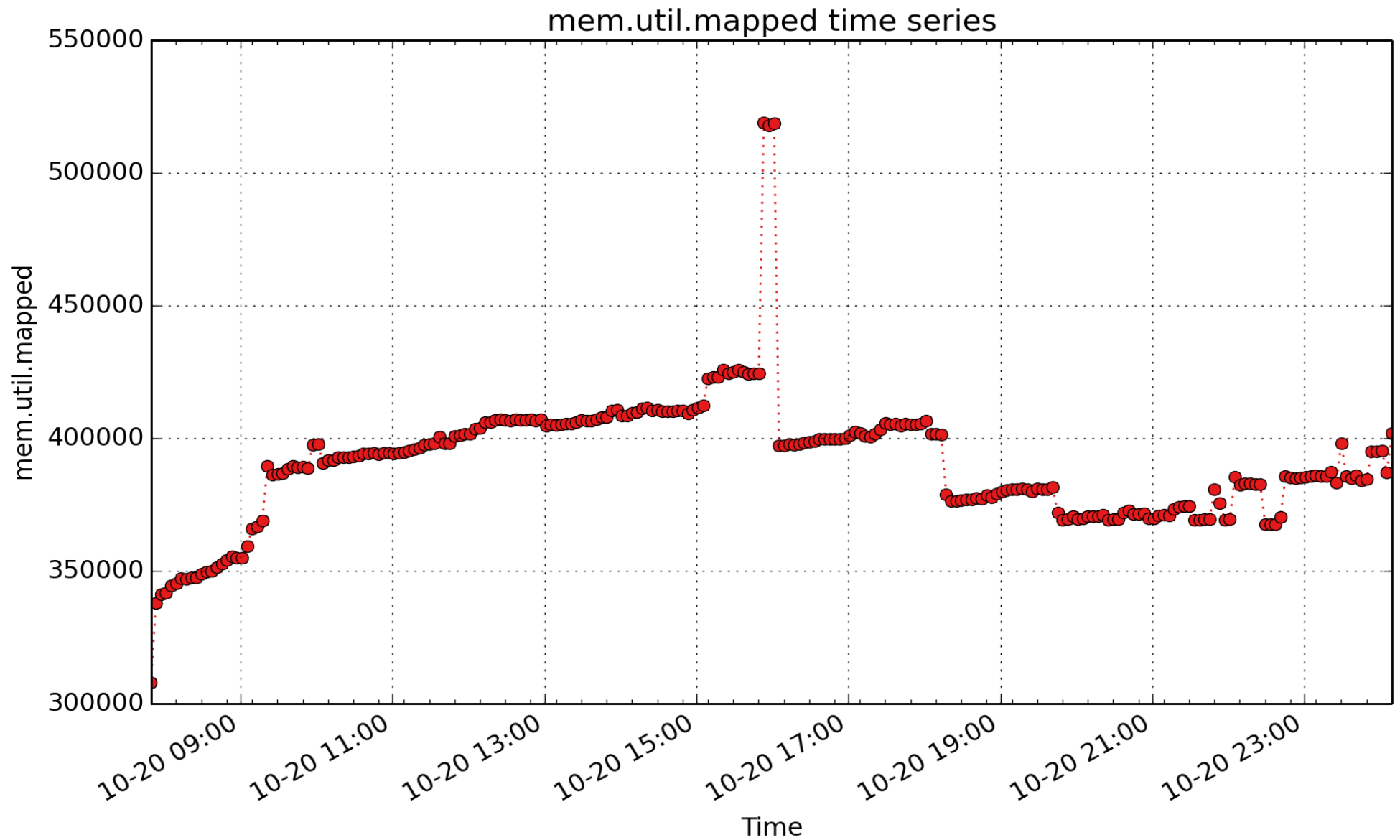
mem.util.inactive_anon: anonymous Inactive list LRU memory (Kbyte - U64)



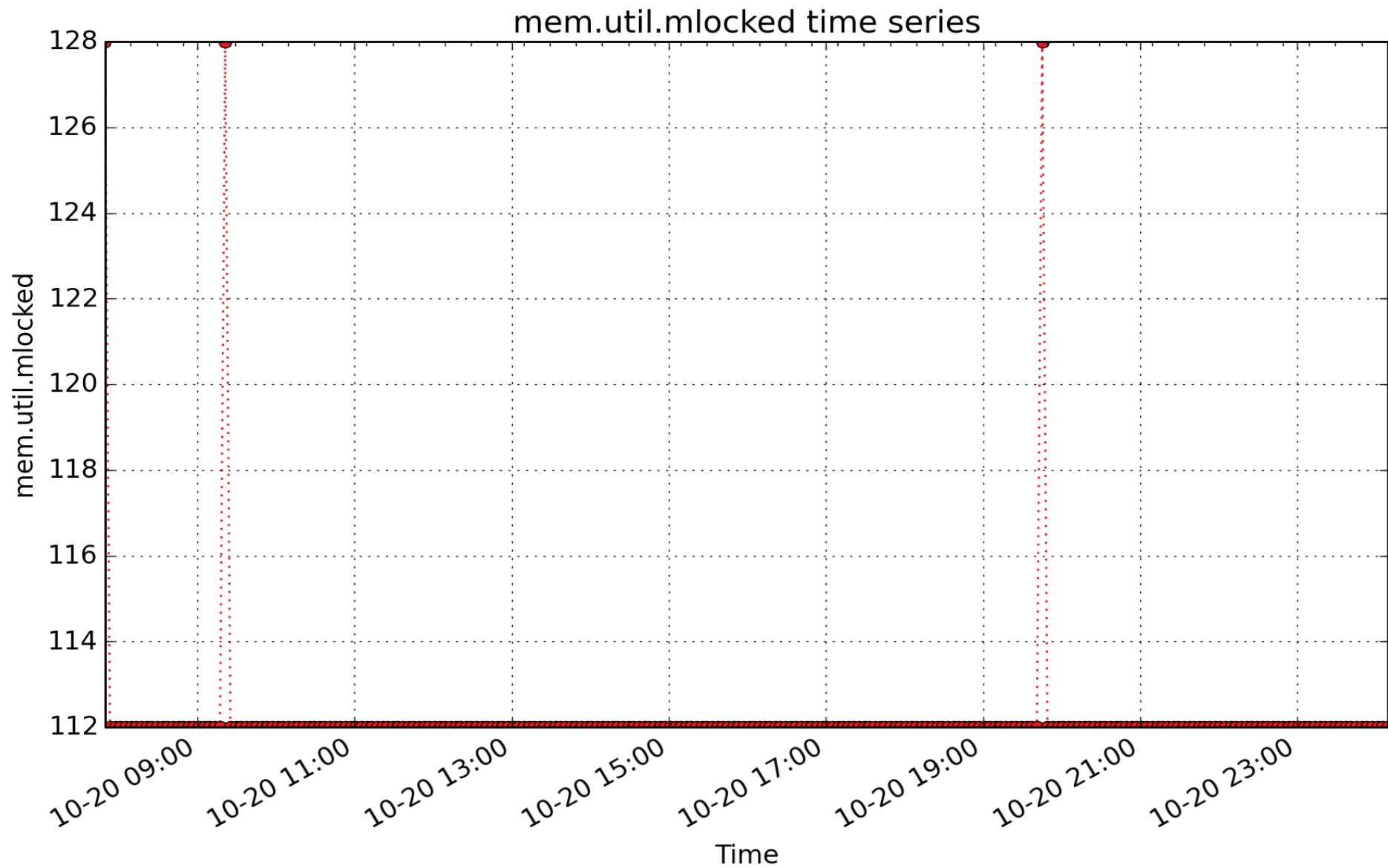
mem.util.inactive_file: file-backed Inactive list LRU memory (Kbyte - U64)



mem.util.kernelStack: kbytes of memory used for kernel stacks (Kbyte - U64)

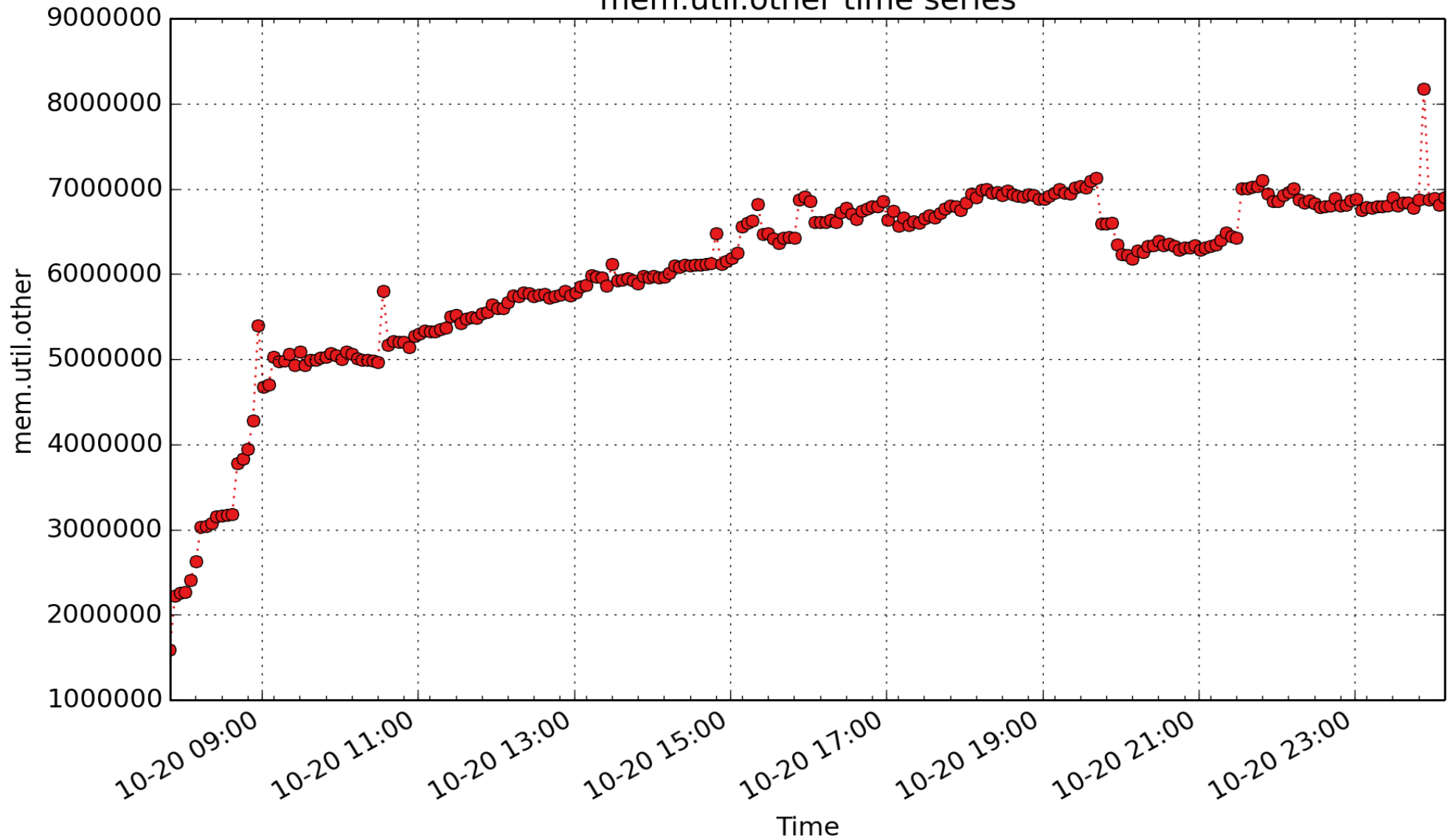


mem.util.mapped: files which have been mmaped, such as libraries (Kbyte - U64)

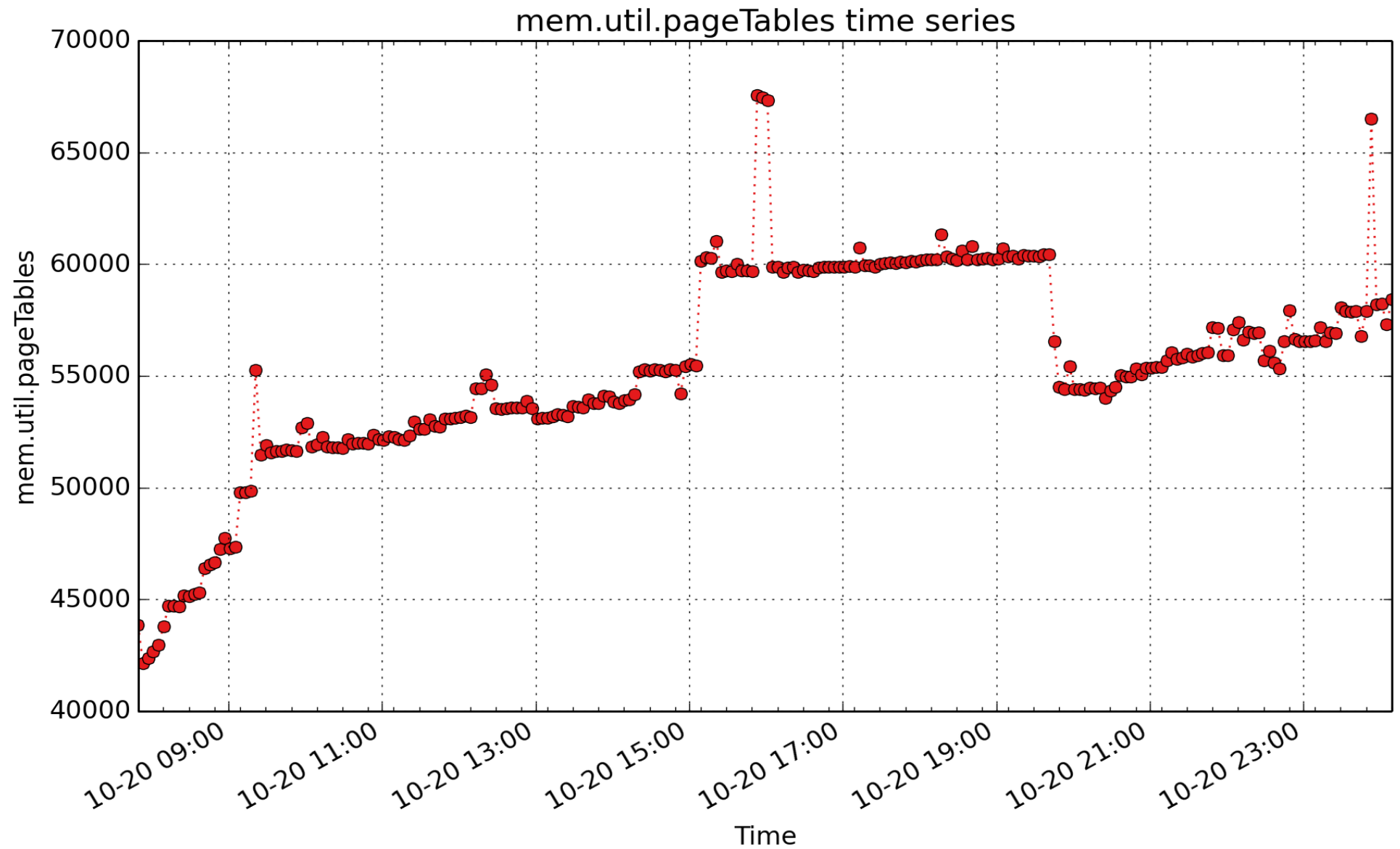


mem.util.mlocked: kbytes of memory that is pinned via mlock() (Kbyte - U64)

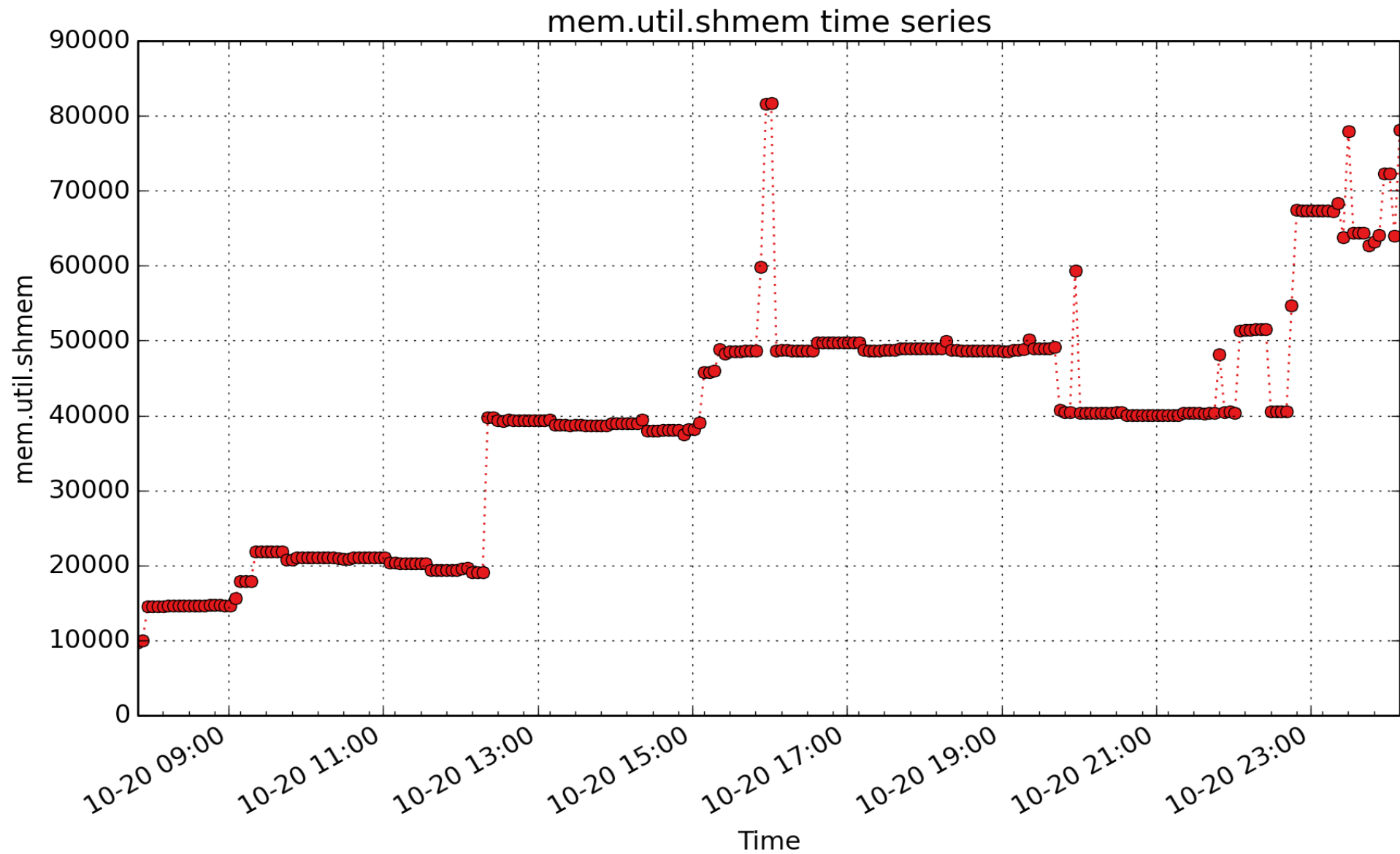
mem.util.other time series



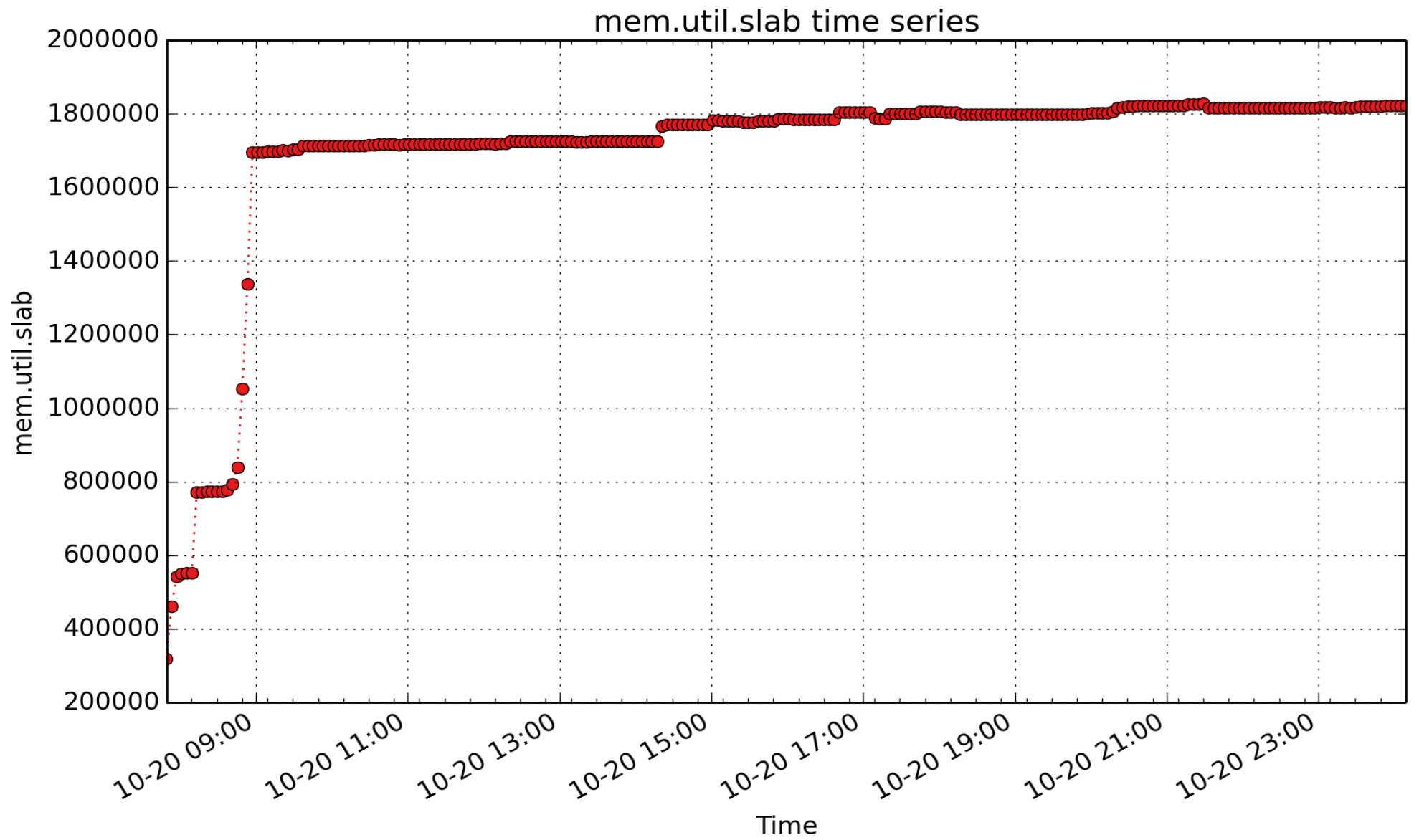
mem.util.other: Memory that is not free (i.e. has been referenced) and is not cached. $\text{mem.physmem} - \text{mem.util.free} - \text{mem.util.cached} - \text{mem.util.buffers}$ (Kbyte - U64)



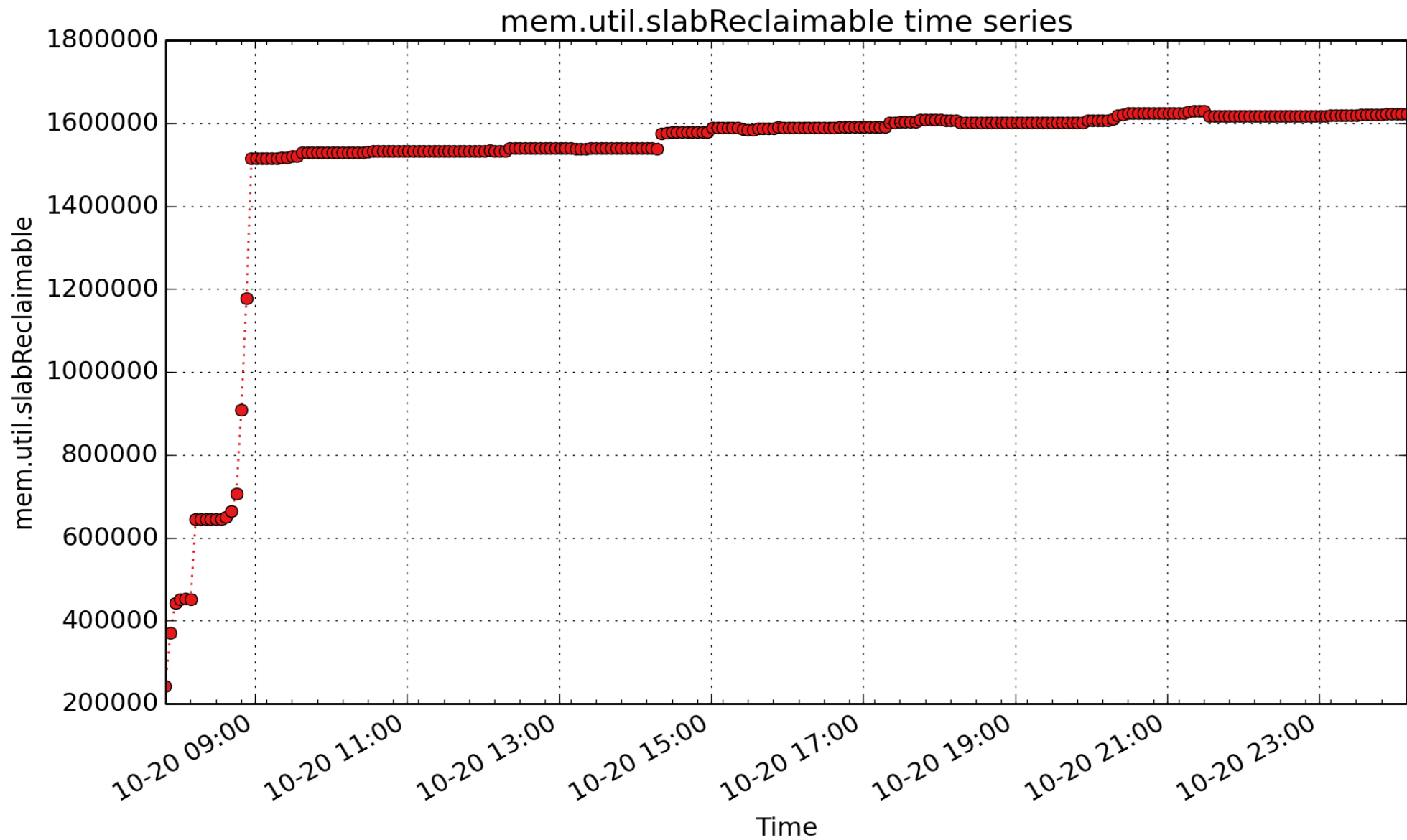
mem.util.pageTables: Kbytes in kernel page tables, from /proc/meminfo (Kbyte - U64)



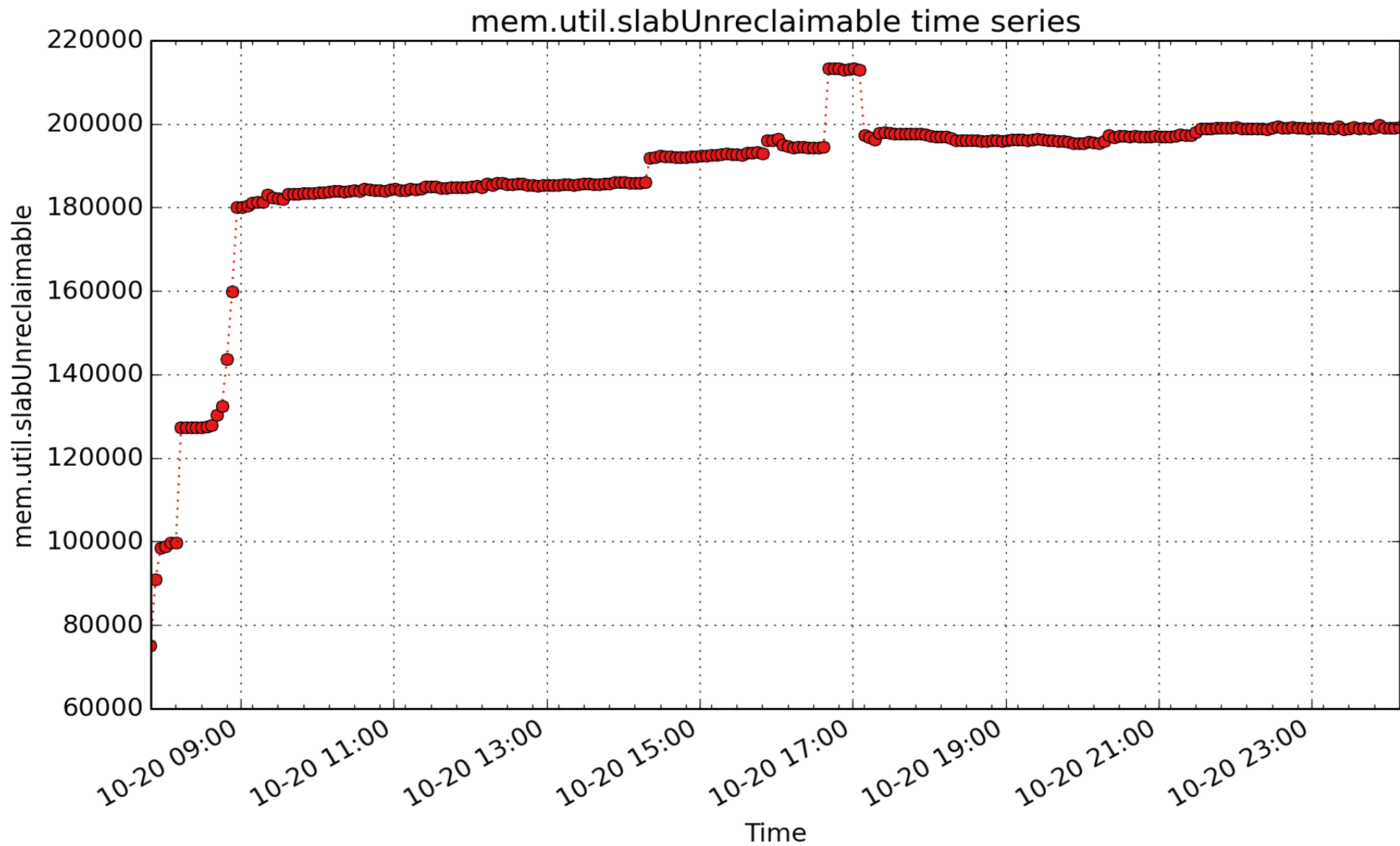
mem.util.shmem: kbytes of shmem (Kbyte - U64)



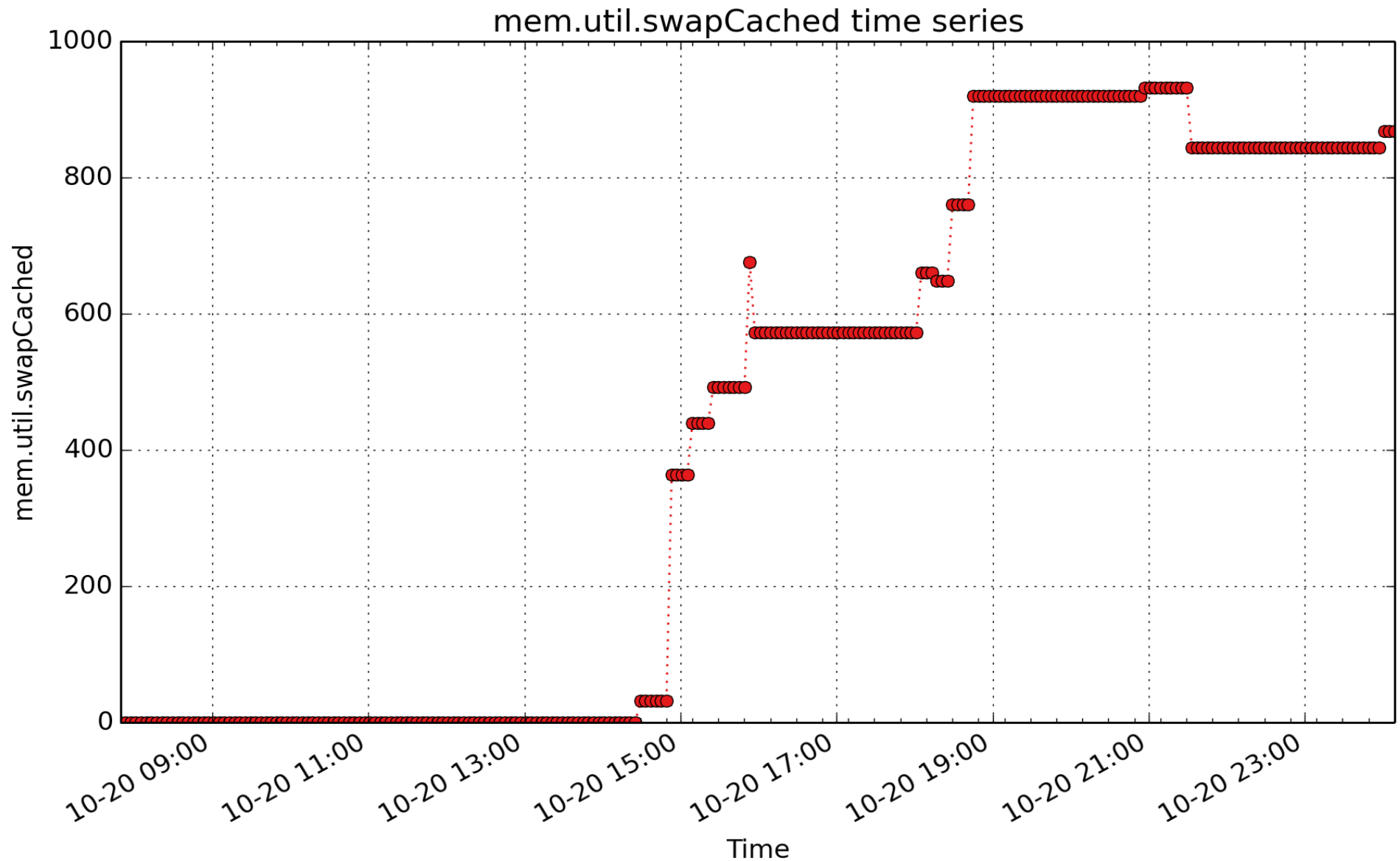
mem.util.slab: in-kernel data structures cache (Kbyte - U64)



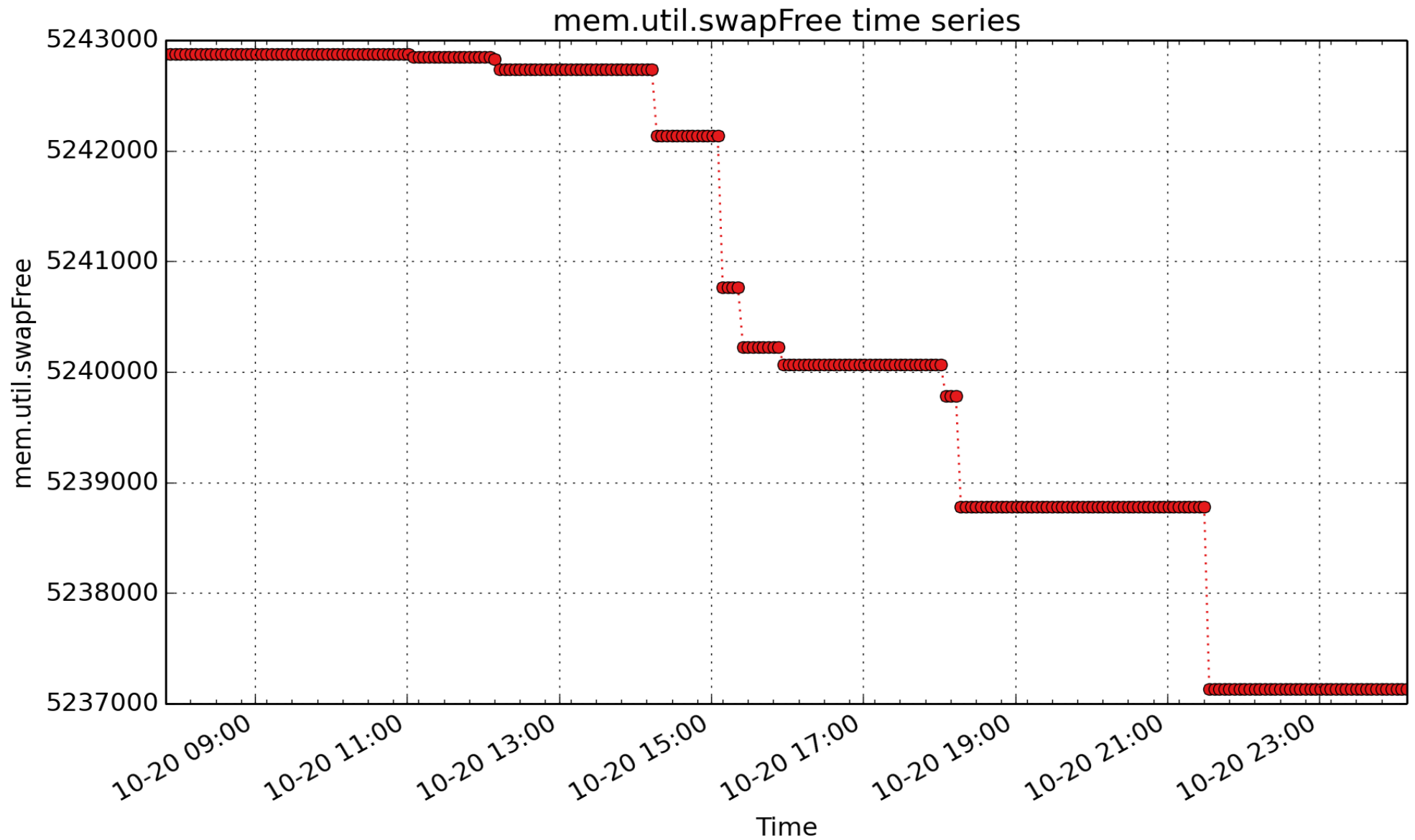
mem.util.slabReclaimable: Kbytes in reclaimable slab pages, from /proc/meminfo (Kbyte - U64)



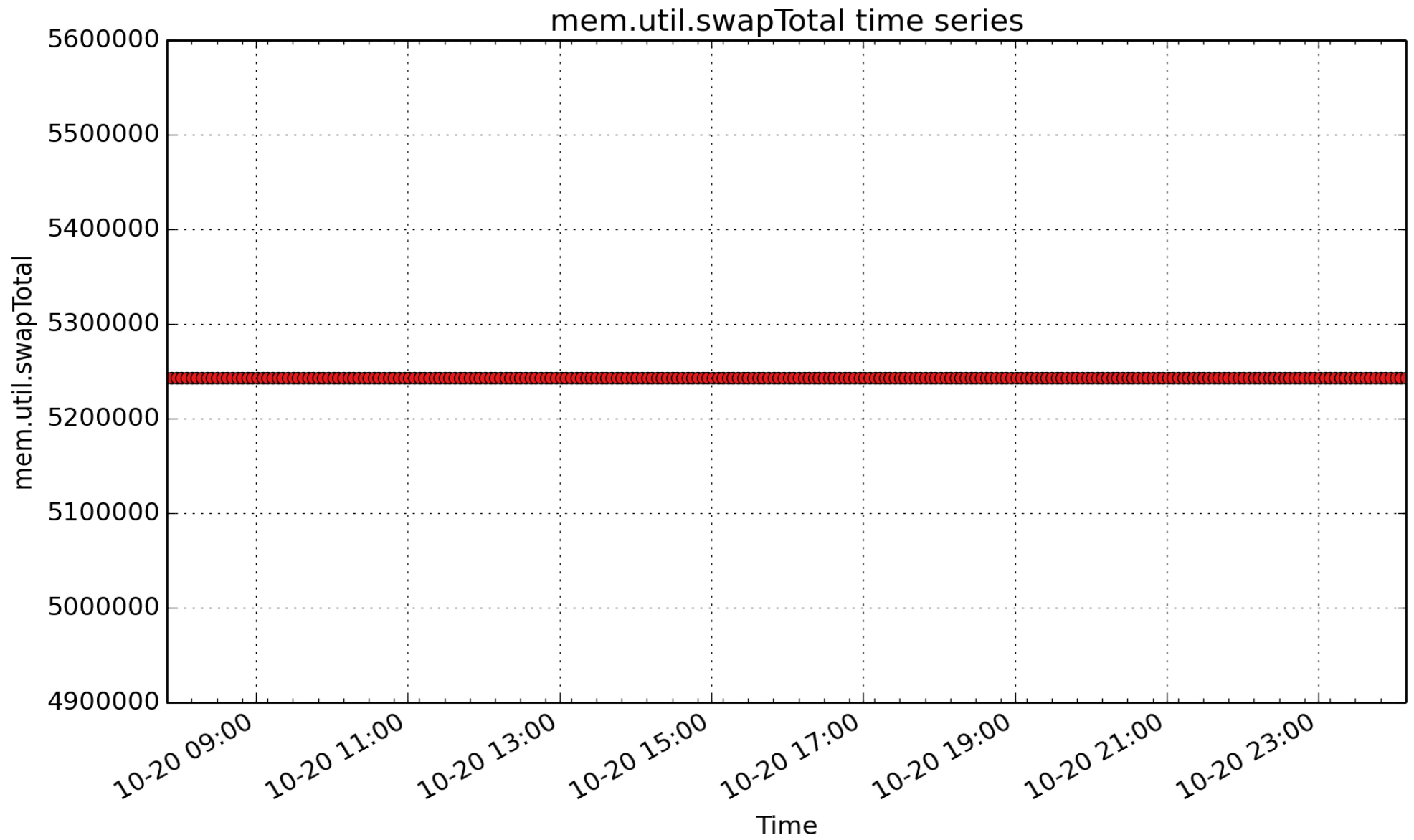
mem.util.slabUnreclaimable: Kbytes in unreclaimable slab pages, from /proc/meminfo (Kbyte - U64)



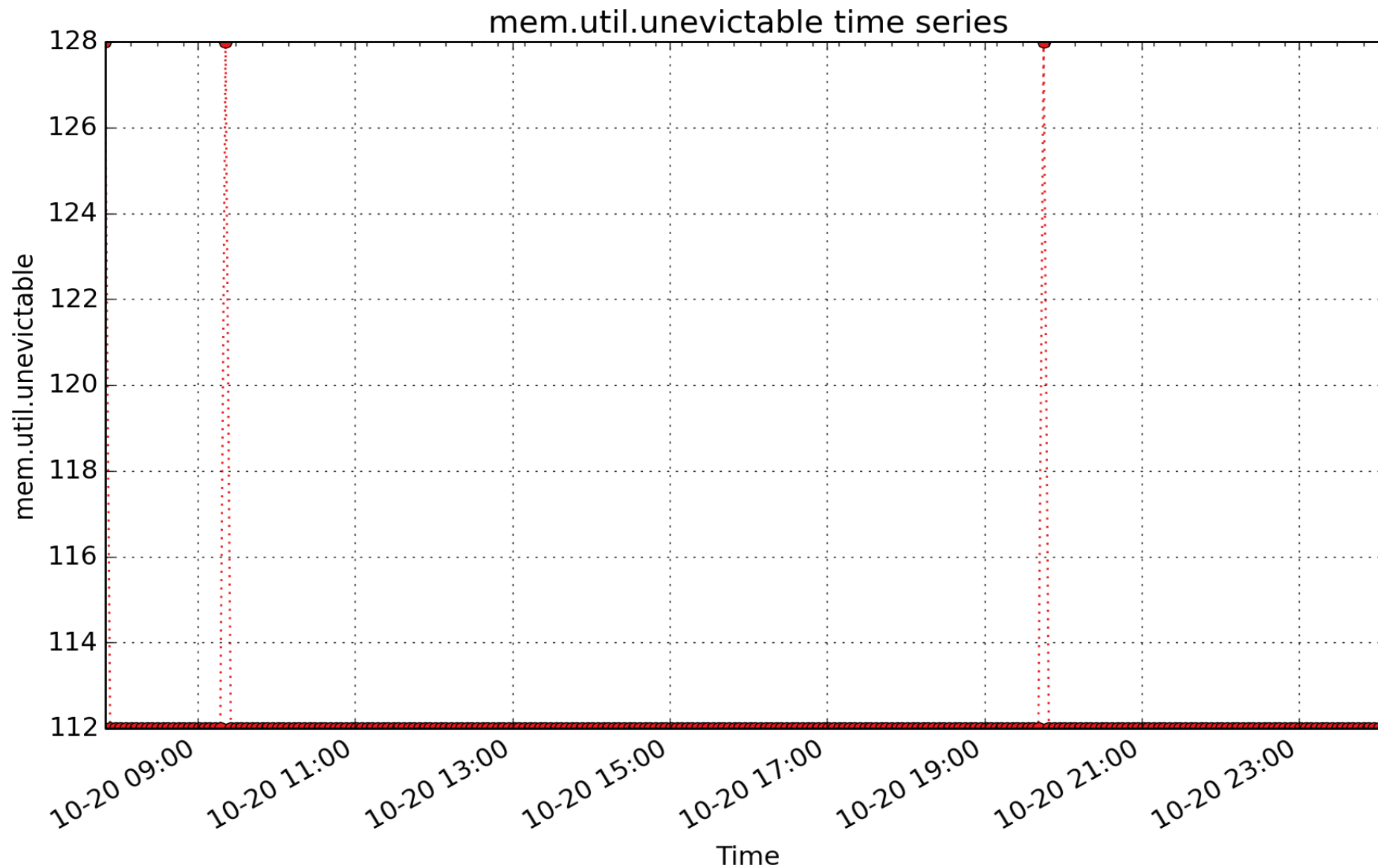
mem.util.swapCached: Memory that once was swapped out, is swapped back in but still also is in the swapfile (if memory is needed it doesn't need to be swapped out AGAIN because it is already in the swapfile. This saves I/O) (Kbyte - U64)



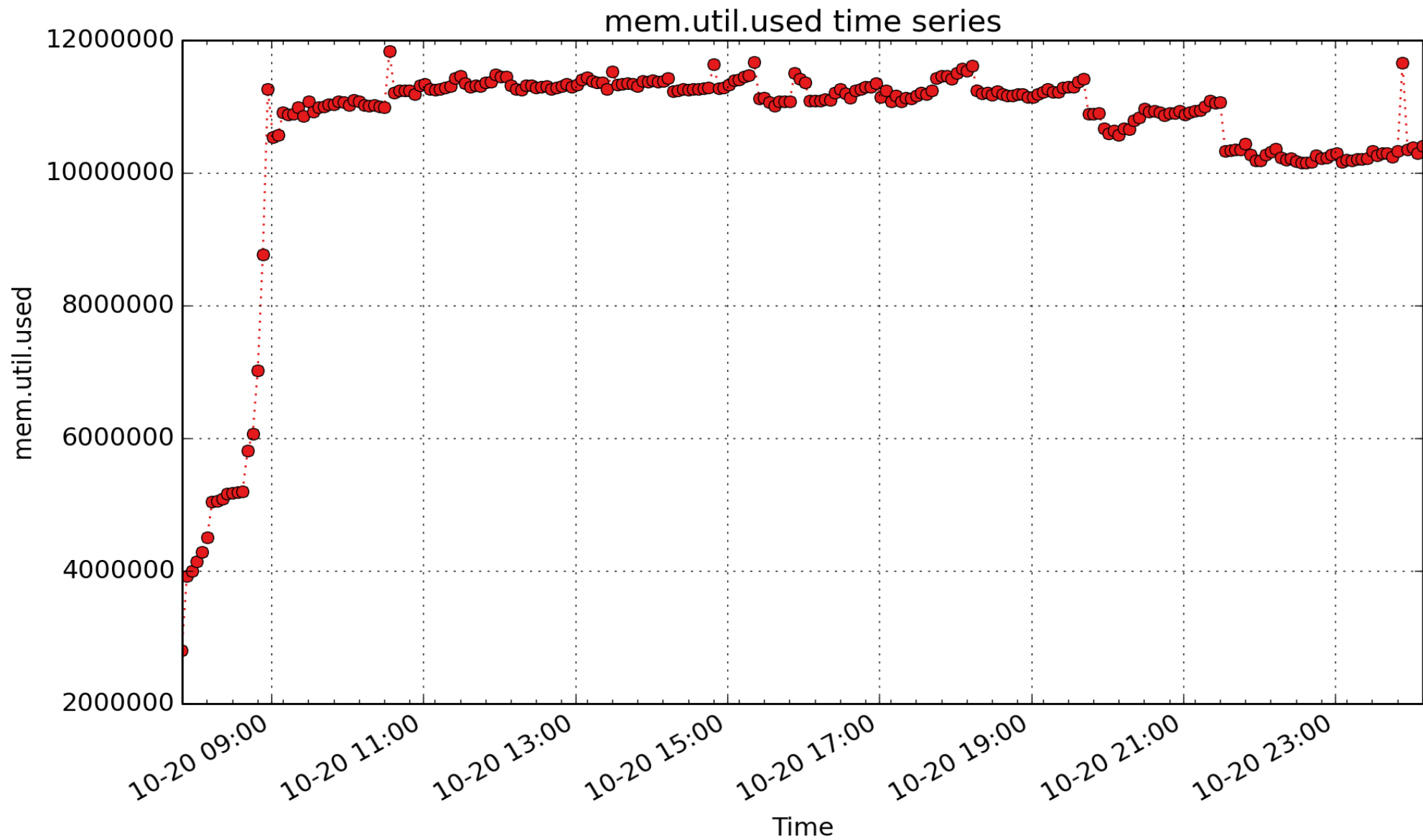
mem.util.swapFree: Memory which has been evicted from RAM, and is temporarily on the disk (Kbyte - U64)



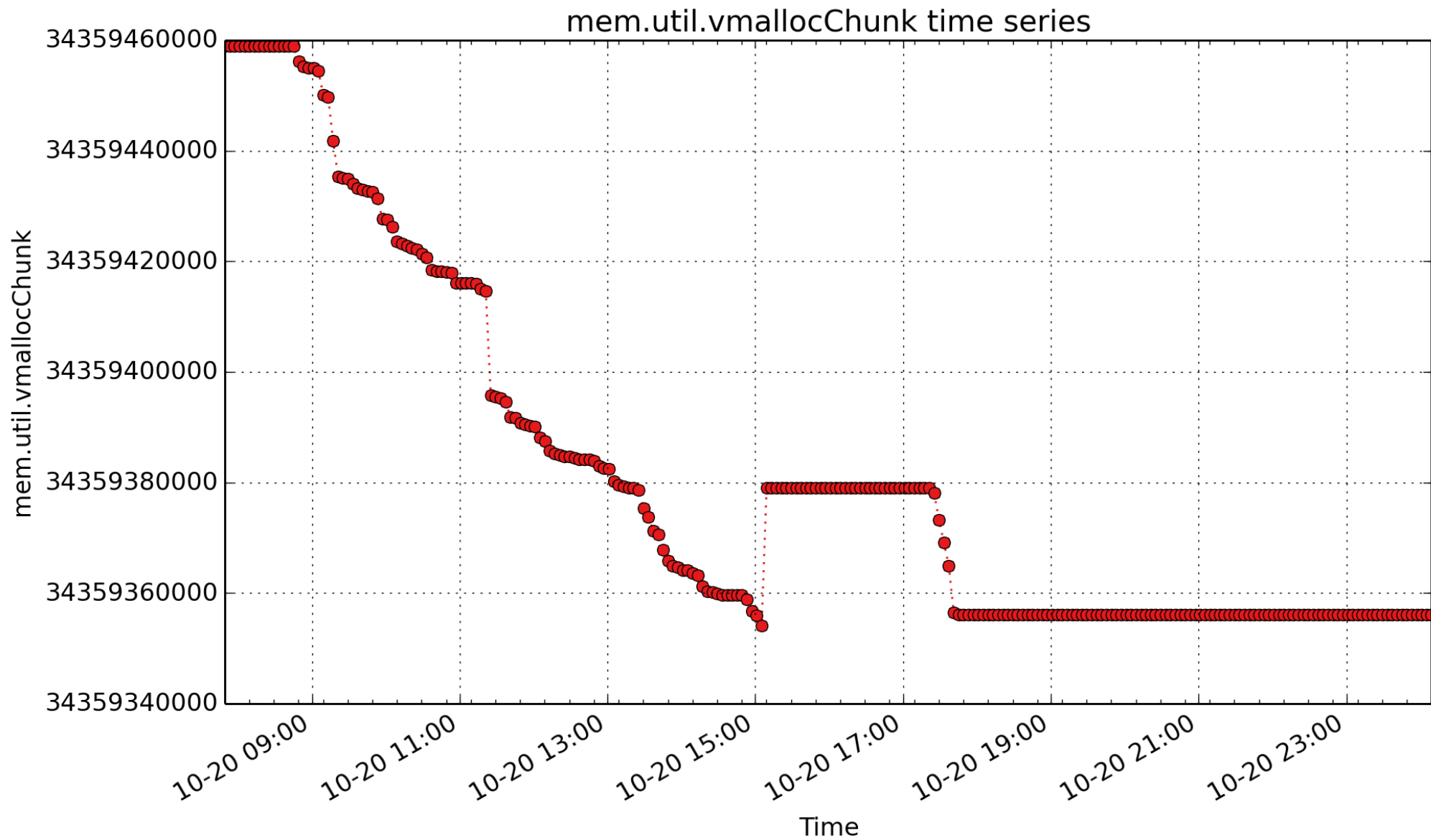
mem.util.swapTotal: total amount of swap space available (Kbyte - U64)



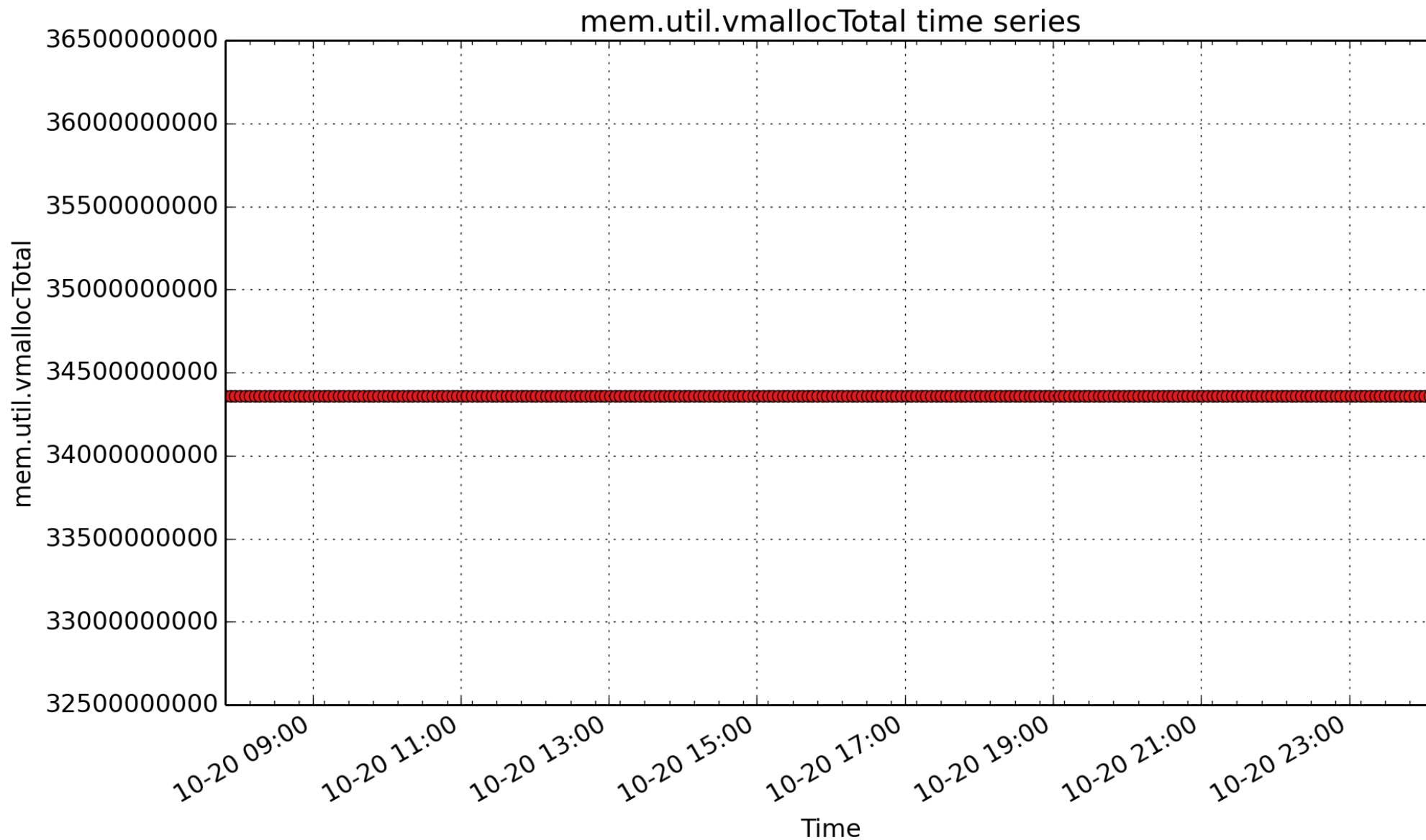
mem.util.unevictable: kbytes of memory that is unevictable (Kbyte - U64)



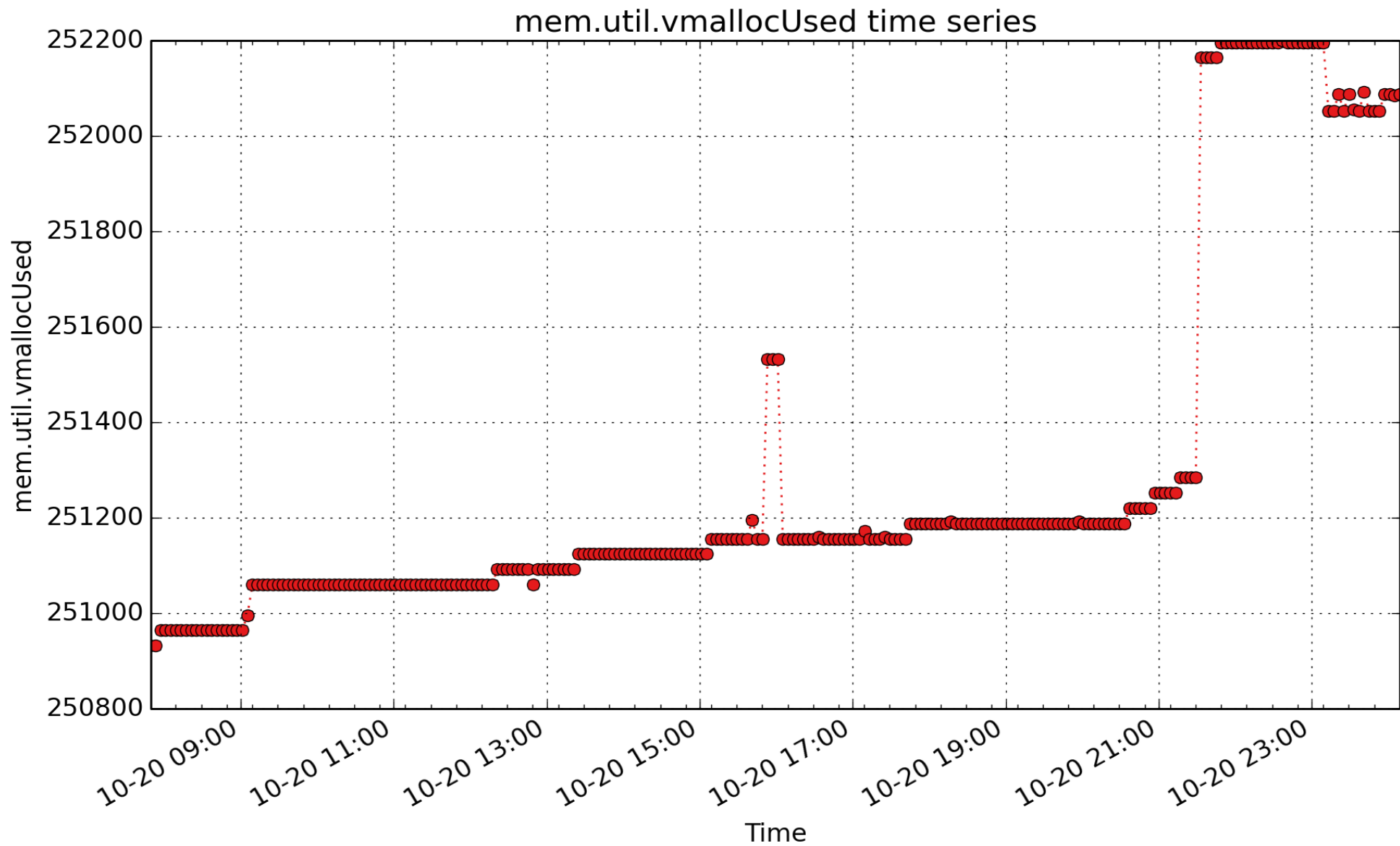
mem.util.used: Used memory is the difference between mem.physmem and mem.freemem. (Kbyte - U64)



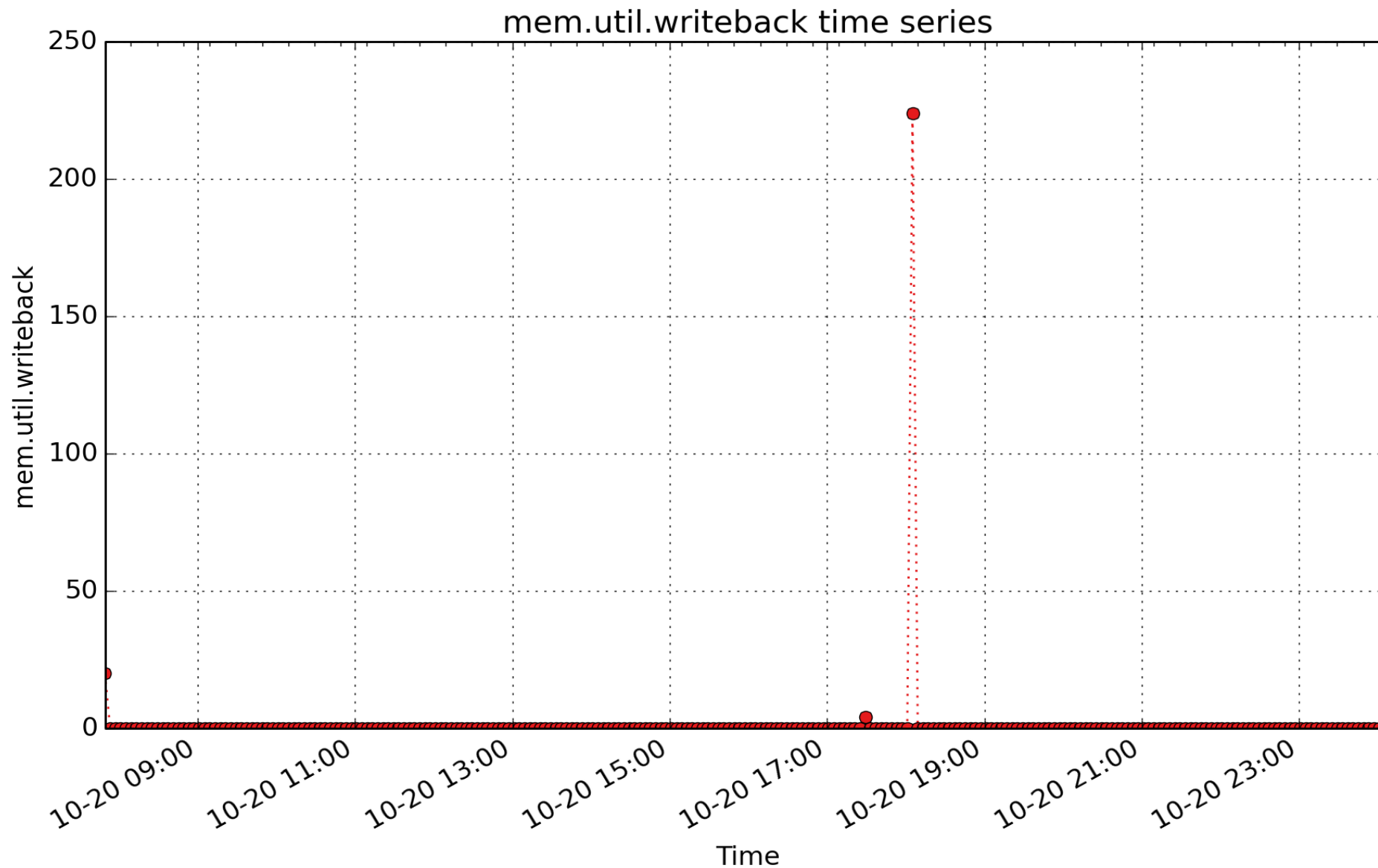
mem.util.vmallocChunk: amount of vmalloc chunk memory (Kbyte - U64)



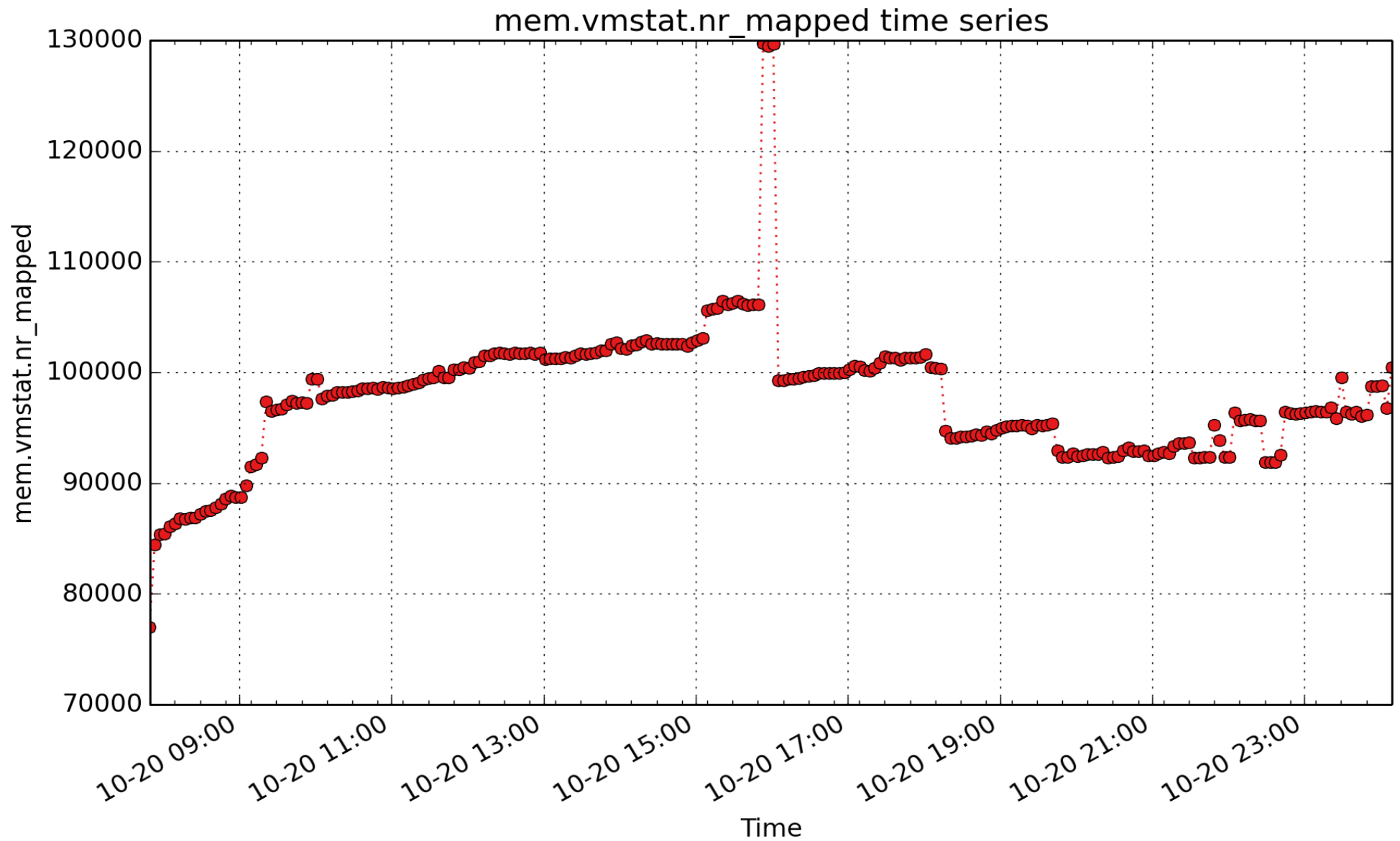
mem.util.vmallocTotal: amount of kernel memory allocated via vmalloc (Kbyte - U64)



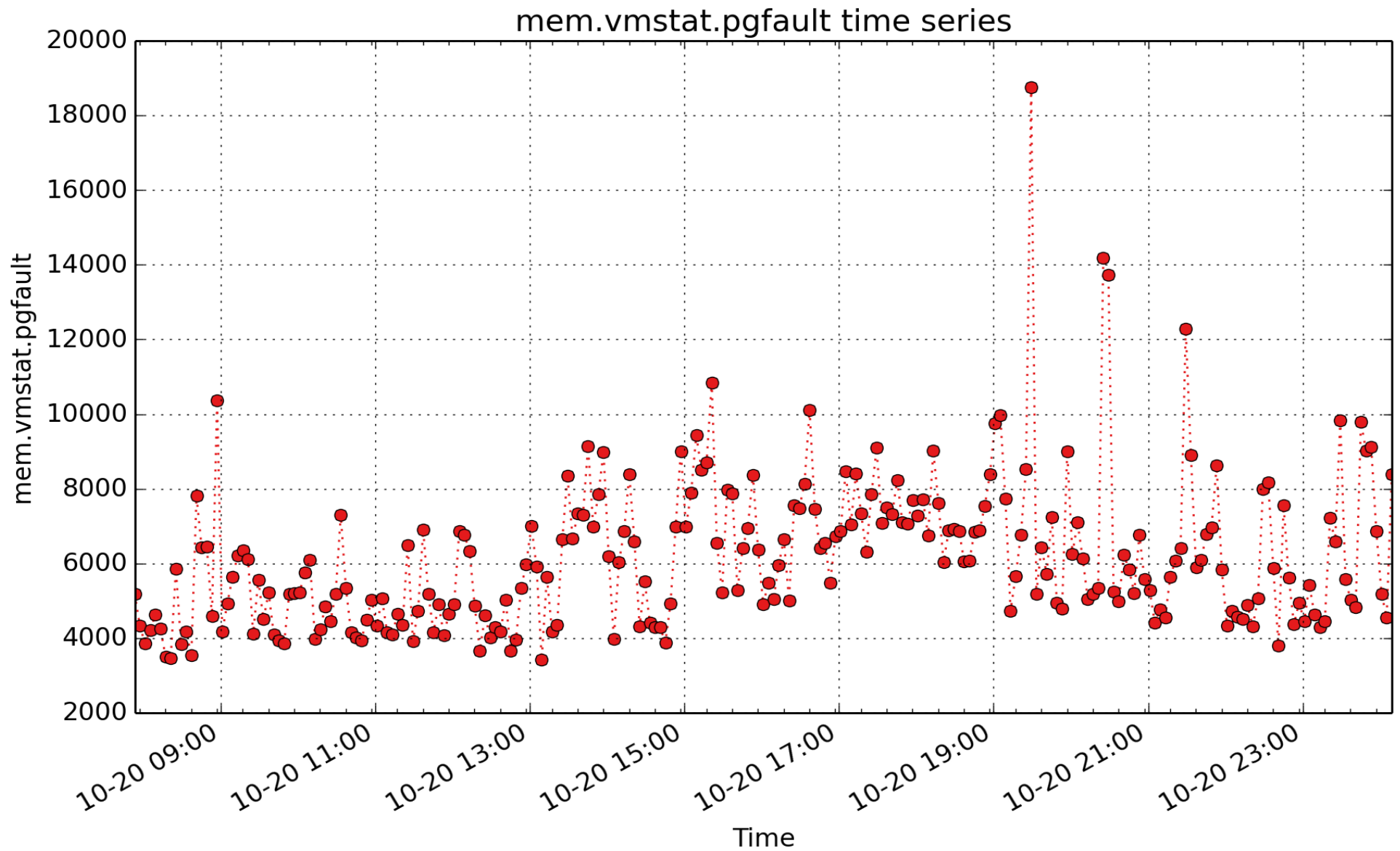
mem.util.vmallocUsed: amount of used vmalloc memory (Kbyte - U64)



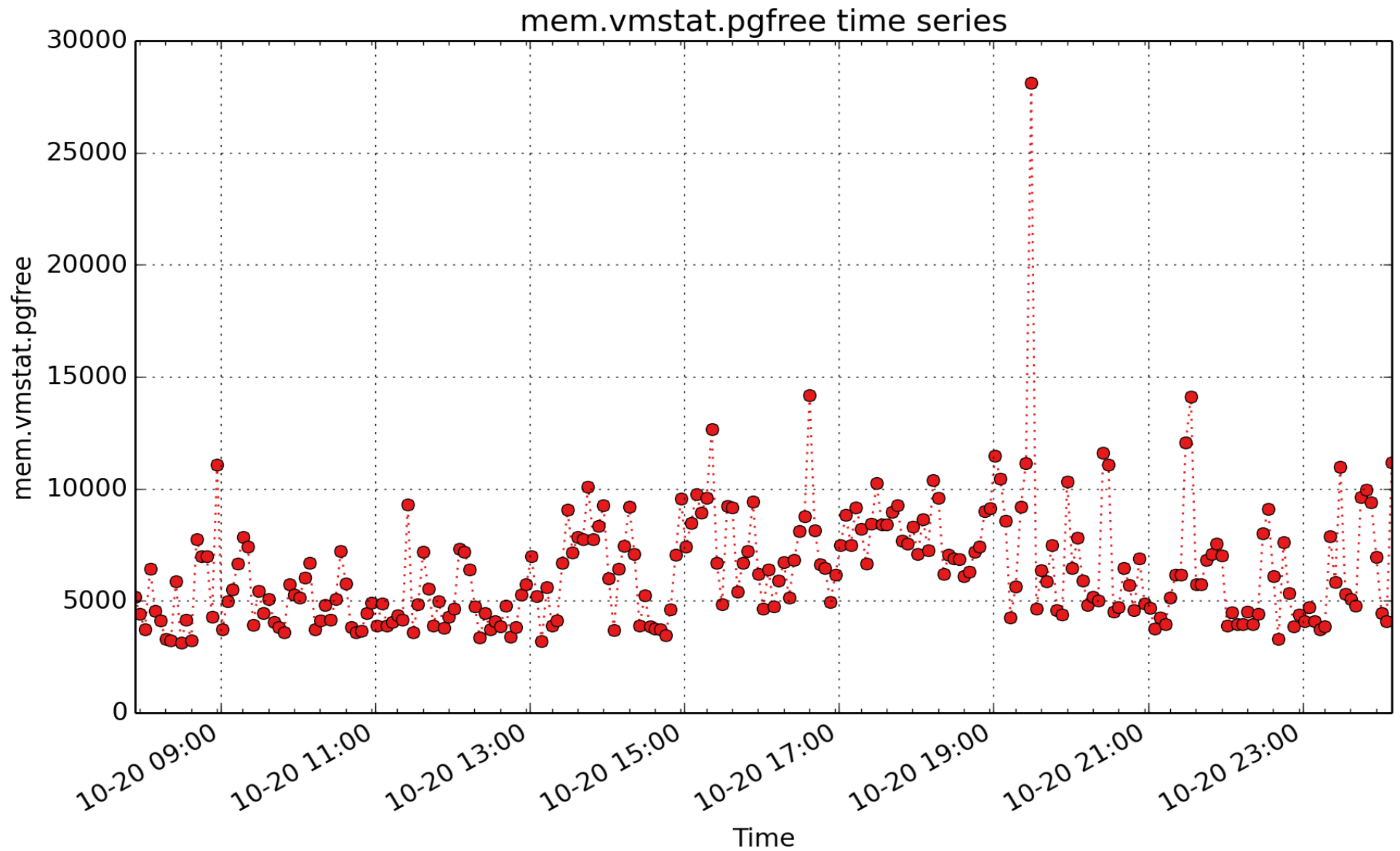
mem.util.writeback: Memory which is actively being written back to the disk (Kbyte - U64)



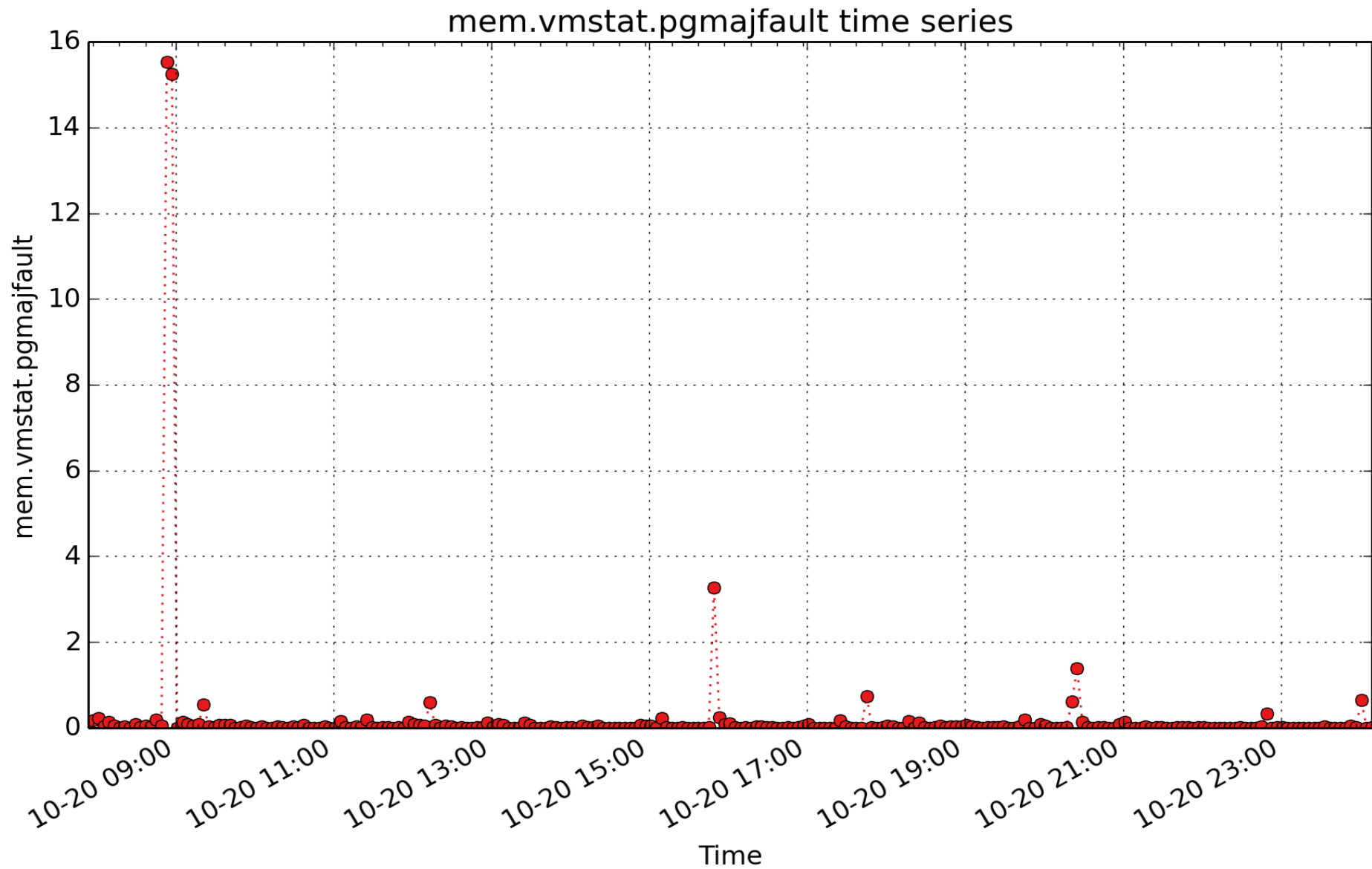
mem.vmstat.nr_mapped: Instantaneous number of mapped pagecache pages, from /proc/vmstat See also mem.vmstat.nr_anon for anonymous mapped pages. (- U64)



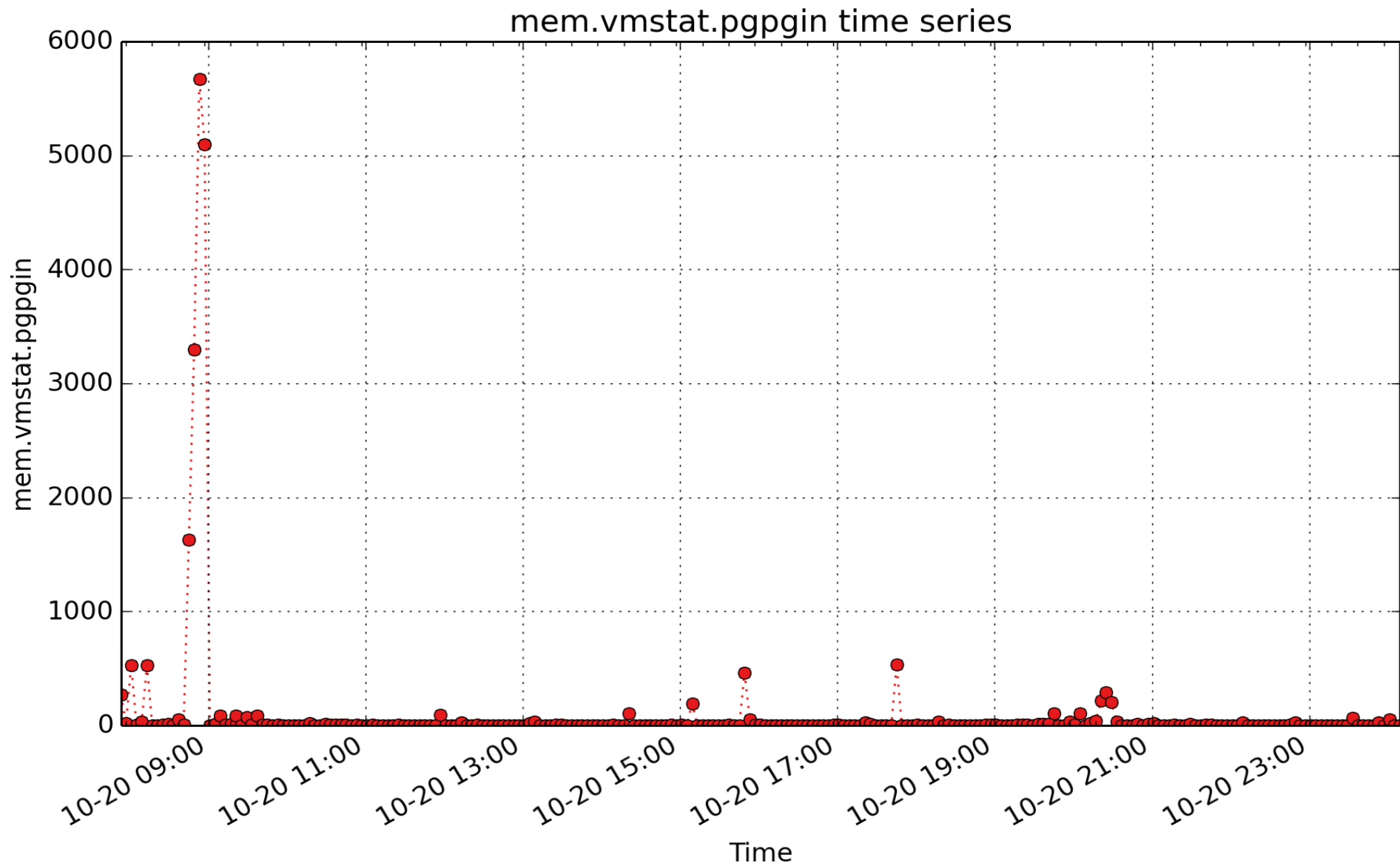
mem.vmstat.pgfault: Count of page major and minor fault operations since boot, from /proc/vmstat (count - U64) - *rate converted*



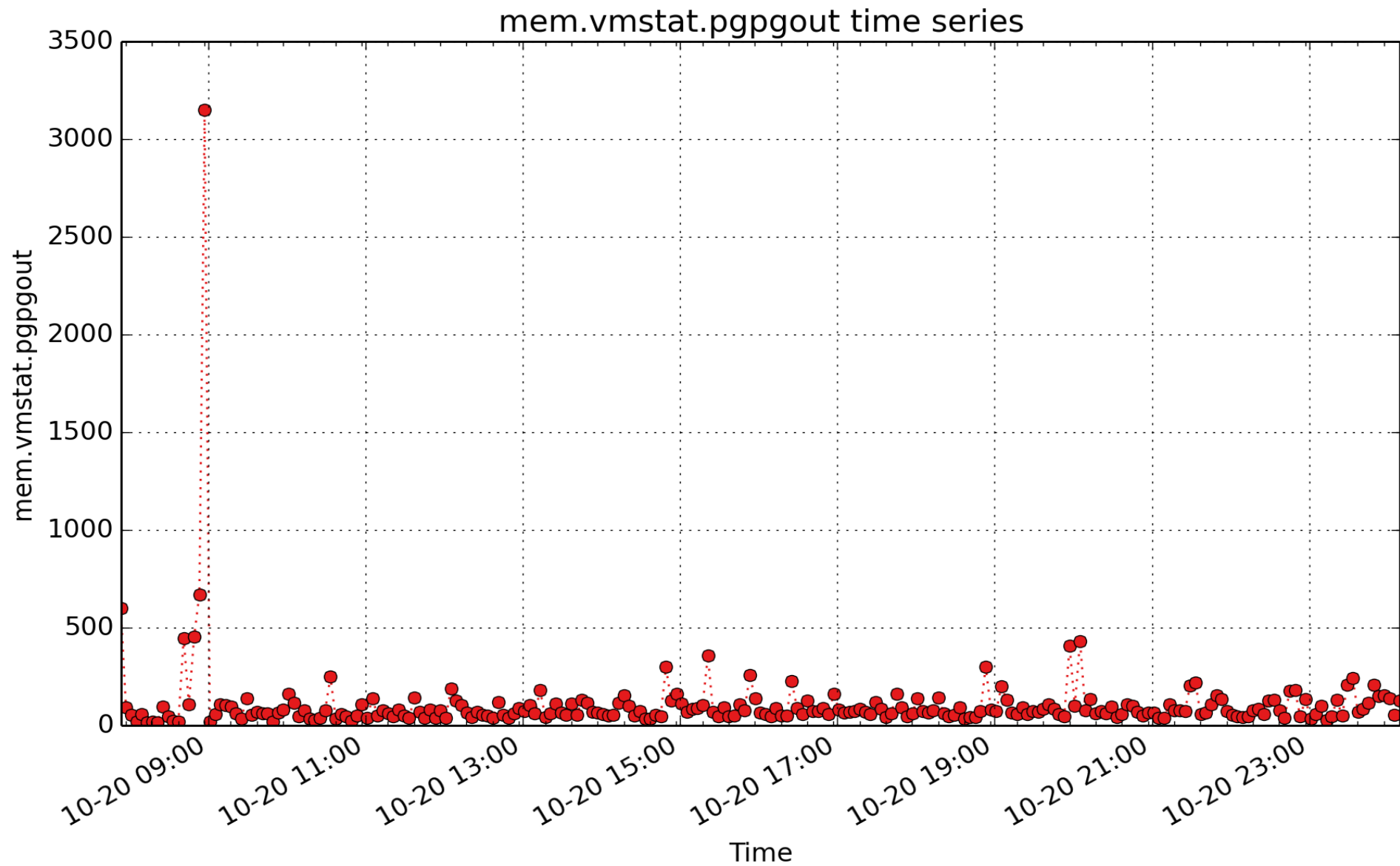
mem.vmstat.pgfree: Count of page free operations since boot, from /proc/vmstat (count - U64) - *rate converted*



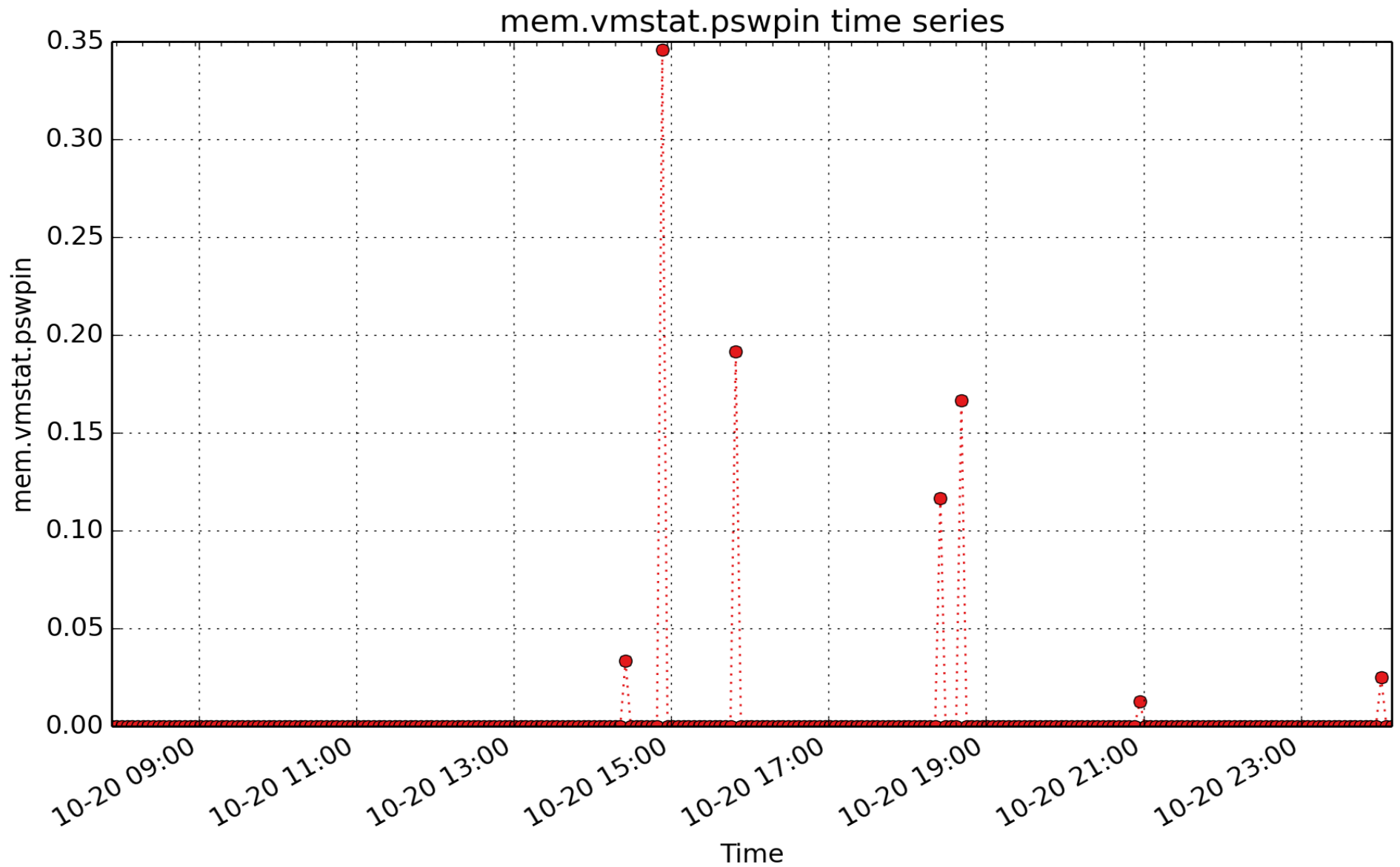
mem.vmstat.pgmajfault: Count of major page fault operations since boot, from /proc/vmstat (count - U64) -
rate converted



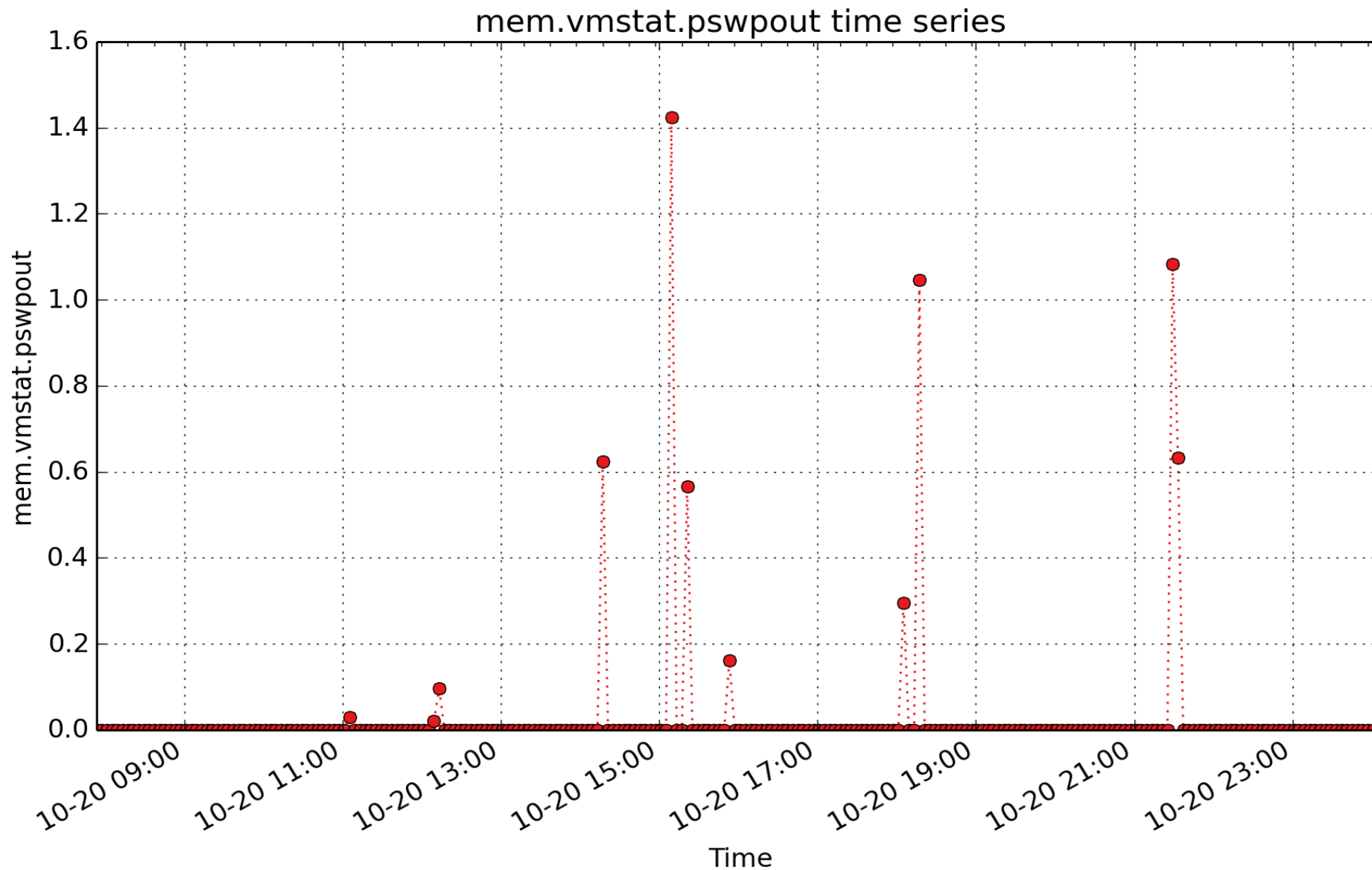
mem.vmstat.pgpgin: Count of page in operations since boot, from /proc/vmstat (count - U64) - *rate converted*



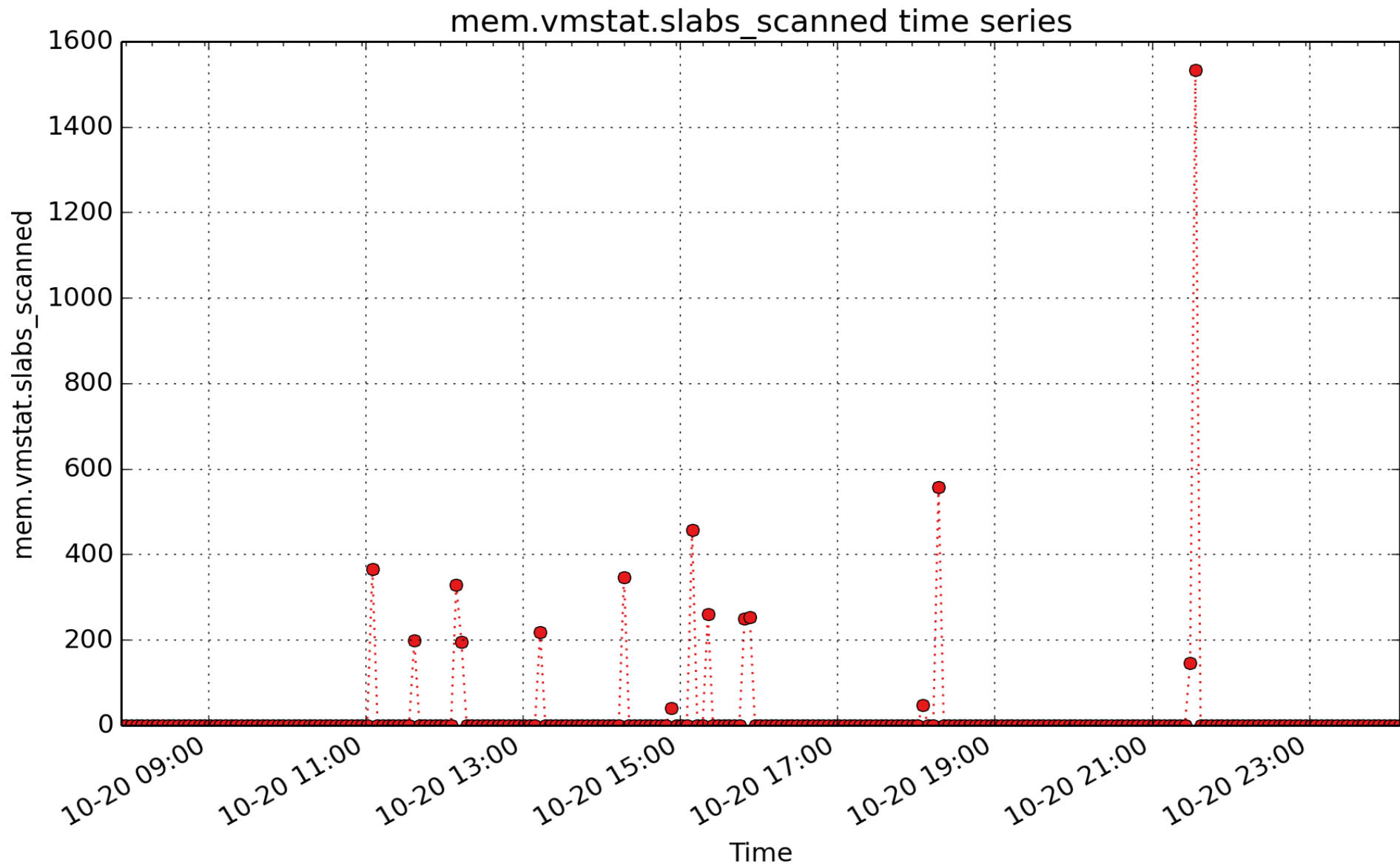
mem.vmstat.pgpgout: Count of page out operations since boot, from `/proc/vmstat` (count - U64) - *rate converted*



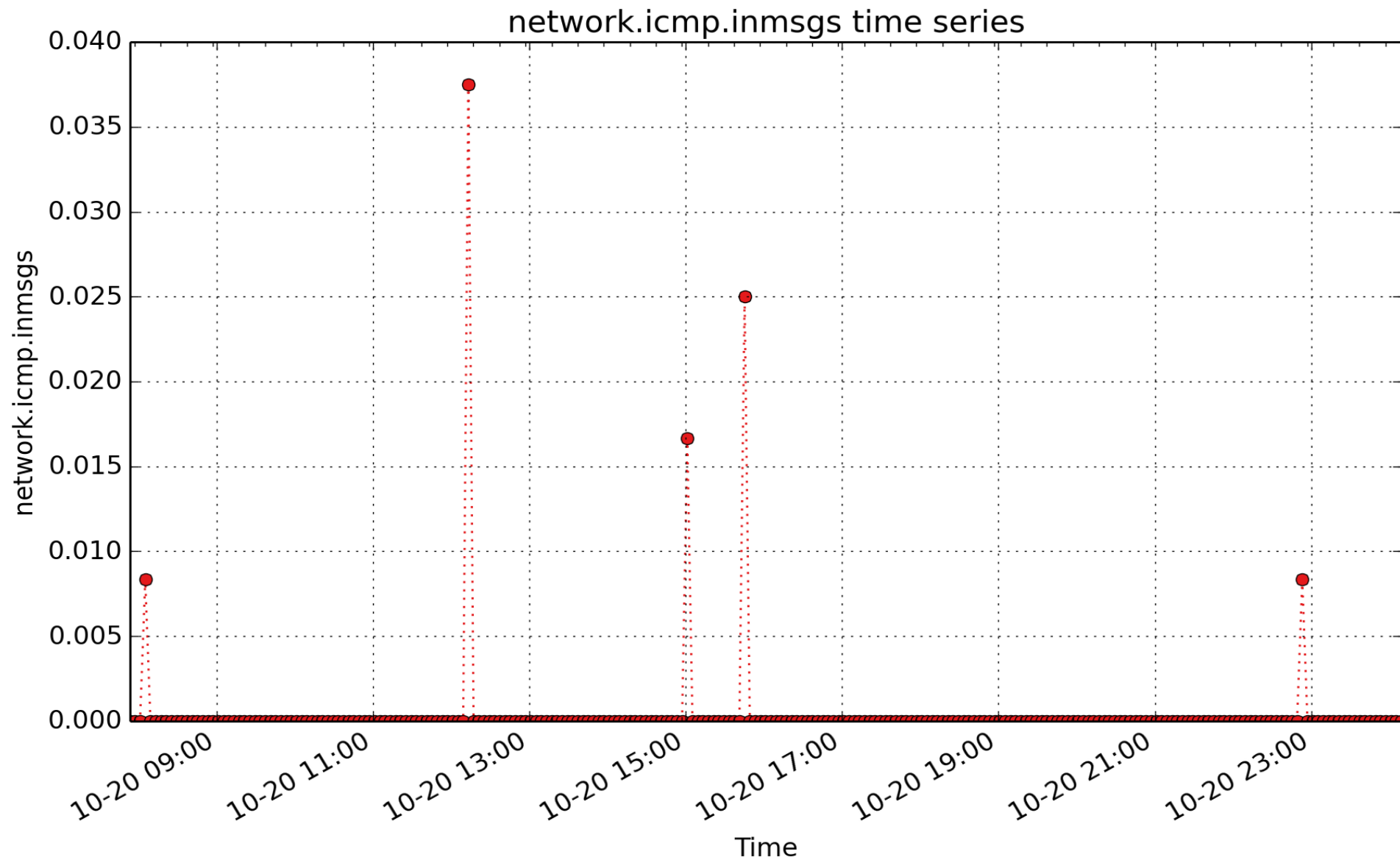
mem.vmstat.pswpin: Count of pages swapped in since boot, from /proc/vmstat (count - U64) - *rate converted*



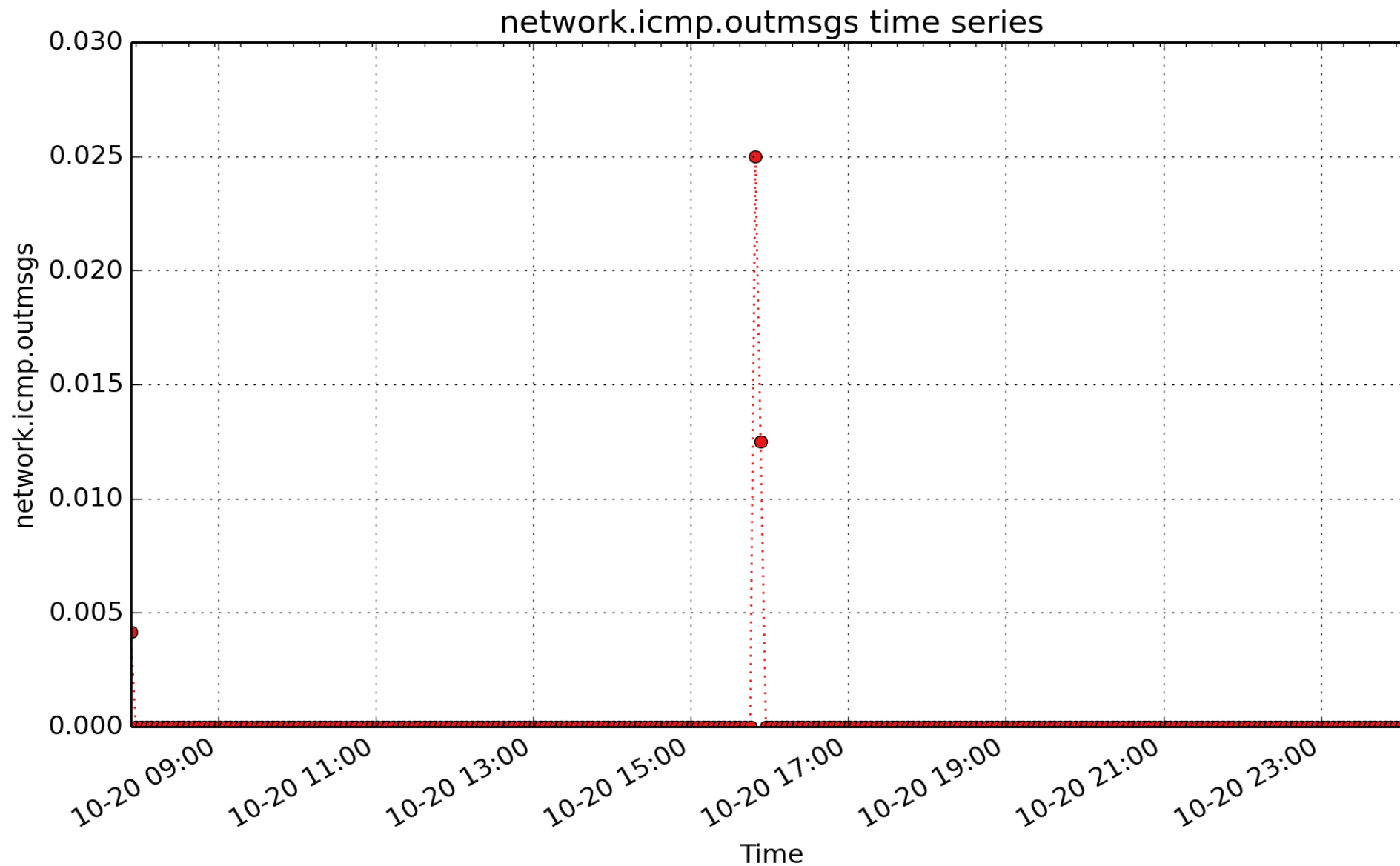
mem.vmstat.pswpout: Count of pages swapped out since boot, from /proc/vmstat (count - U64) - *rate converted*



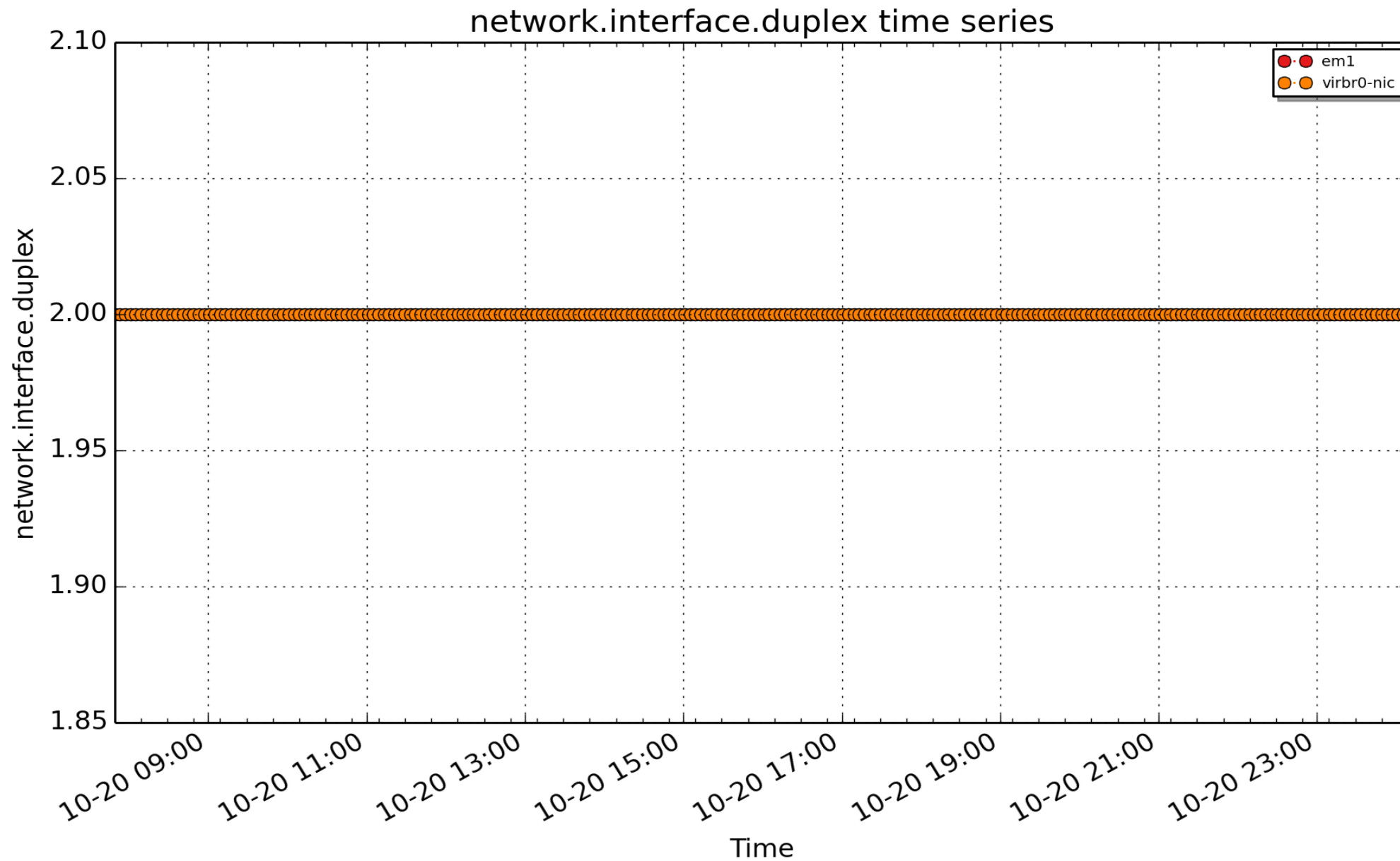
mem.vmstat.slabs_scanned: Count of slab pages scanned since boot, from /proc/vmstat (count - U64) - *rate converted*



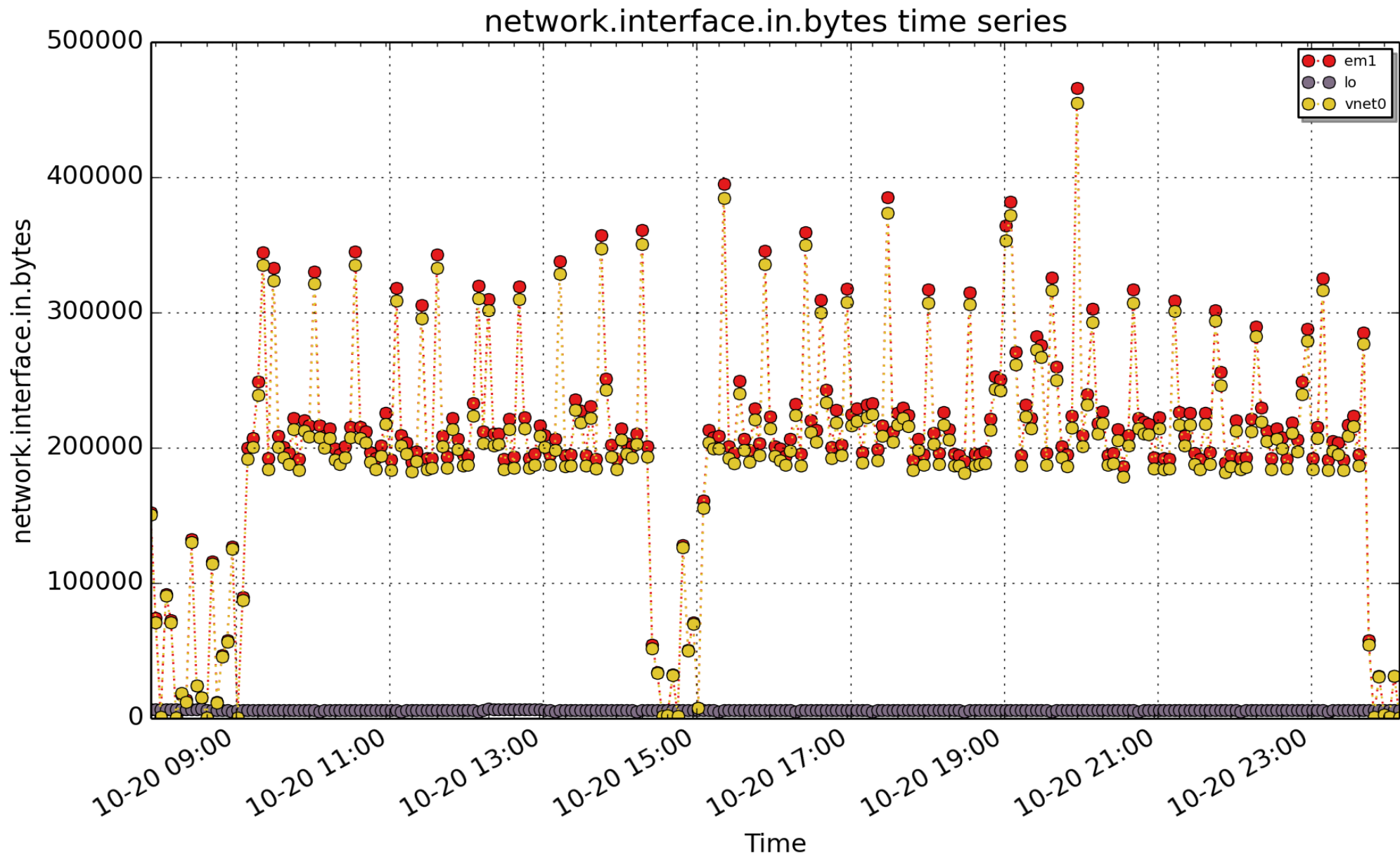
network.icmp.inmsgs: count of icmp inmsgs (count - U64) - *rate converted*



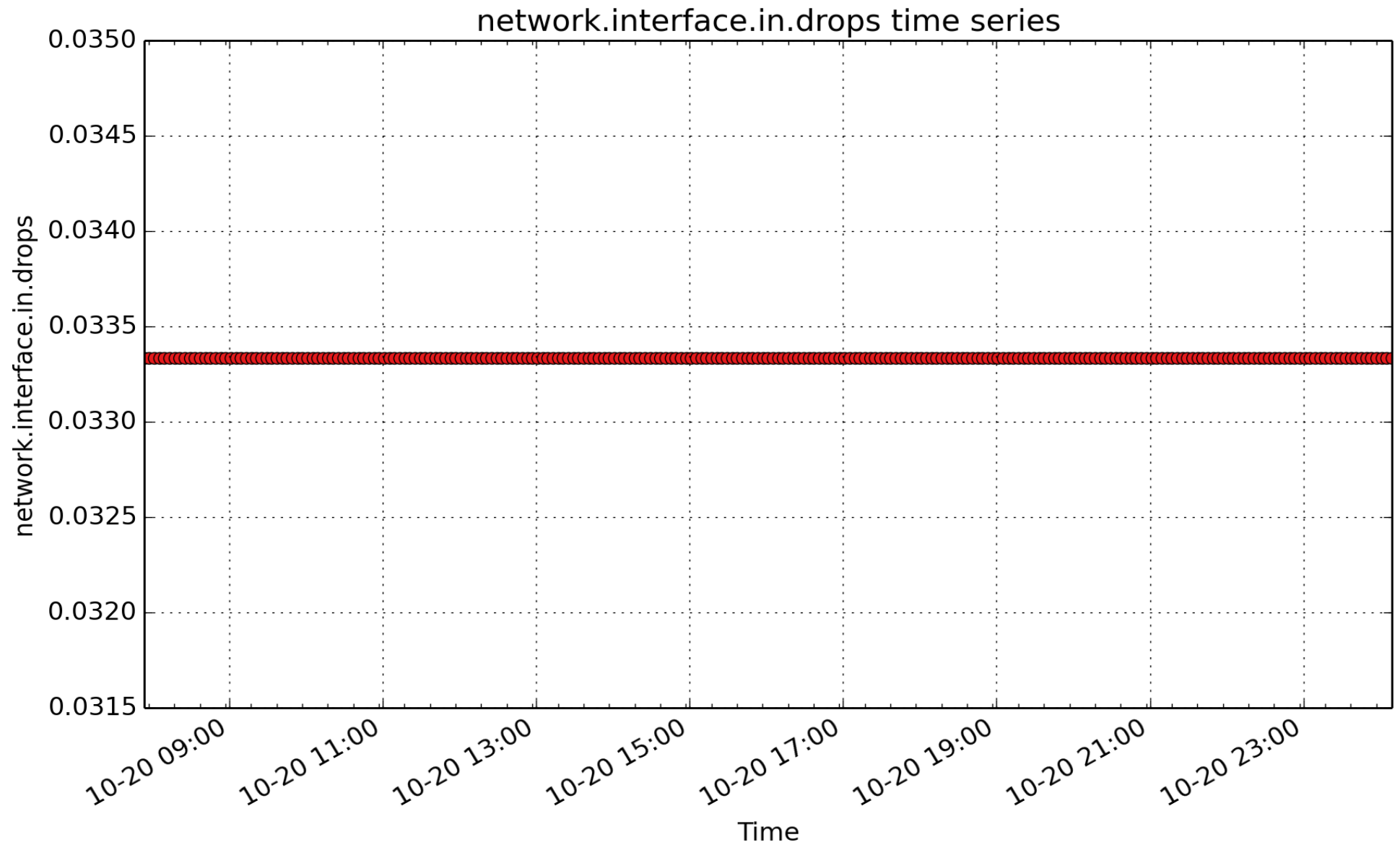
network.icmp.outmsgs: count of icmp outmsgs (count - U64) - *rate converted*



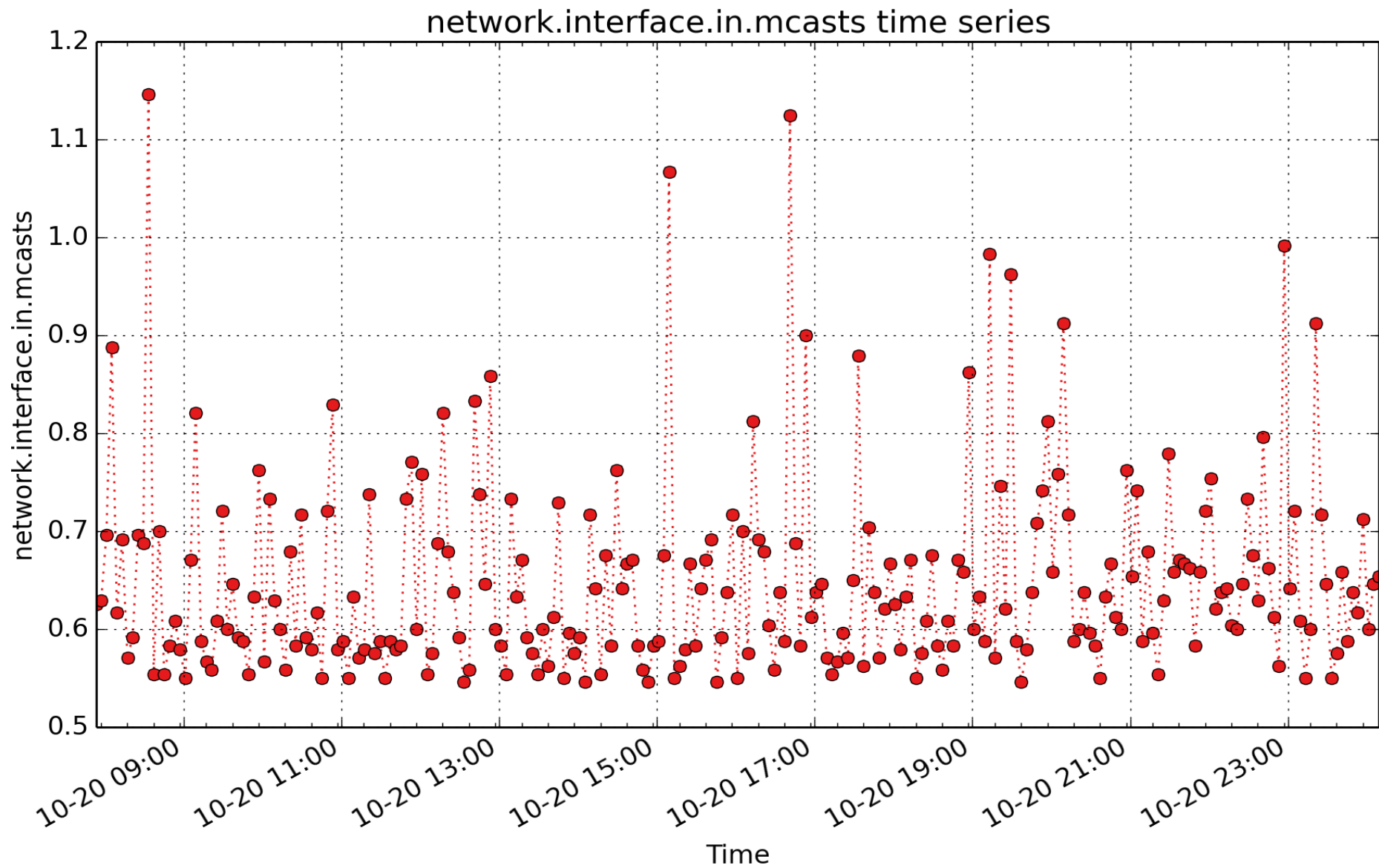
network.interface.duplex: value one for half or two for full duplex interface (- U32)



network.interface.in.bytes: network recv read bytes from /proc/net/dev per network interface (byte - U64) - rate converted

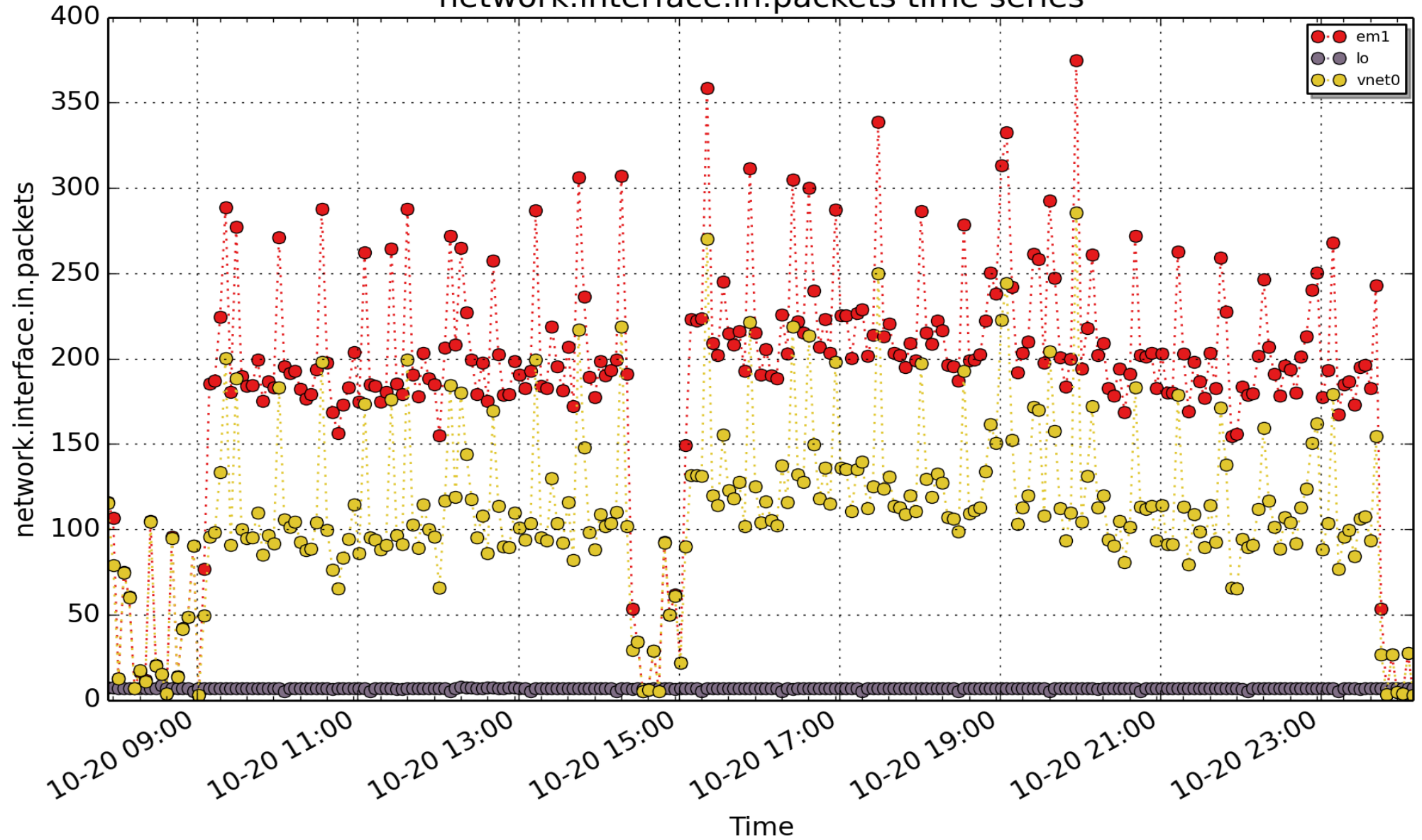


network.interface.in.drops: network recv read drops from /proc/net/dev per network interface (count - U64) -
rate converted

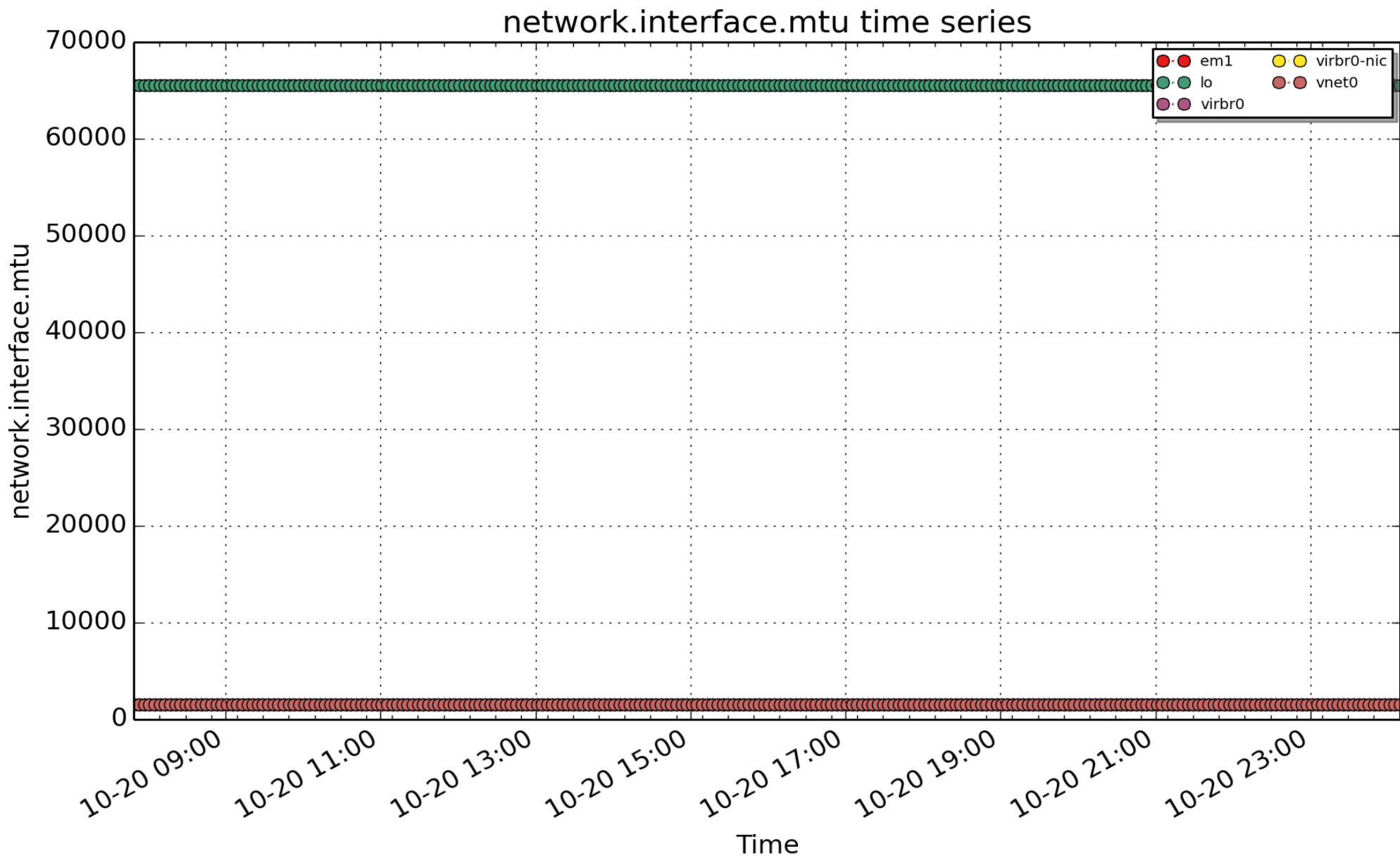


network.interface.in.mcasts: network recv compressed from /proc/net/dev per network interface (count - U64) - *rate converted*

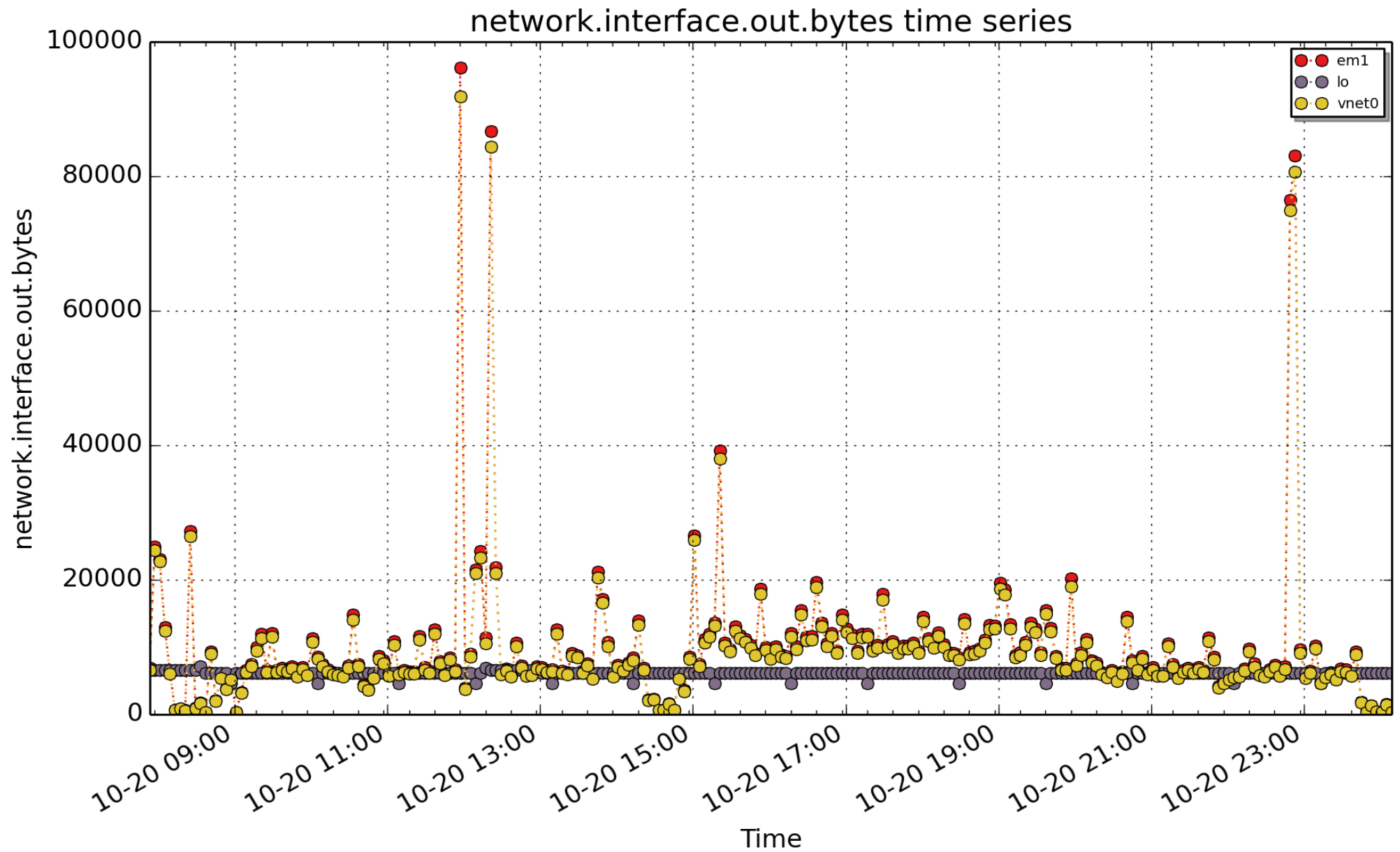
network.interface.in.packets time series



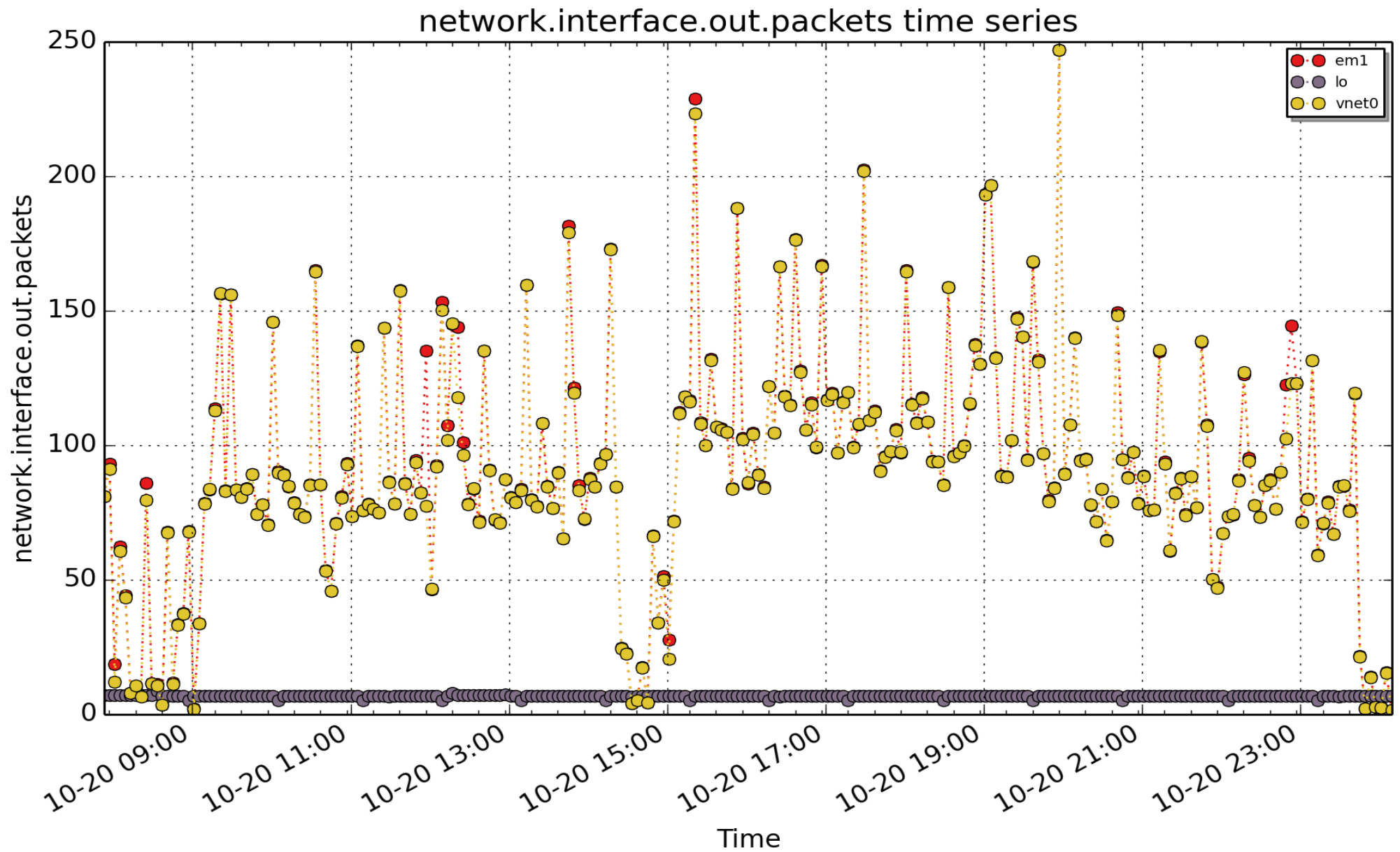
network.interface.in.packets: network recv read packets from /proc/net/dev per network interface (count - U64) - *rate converted*



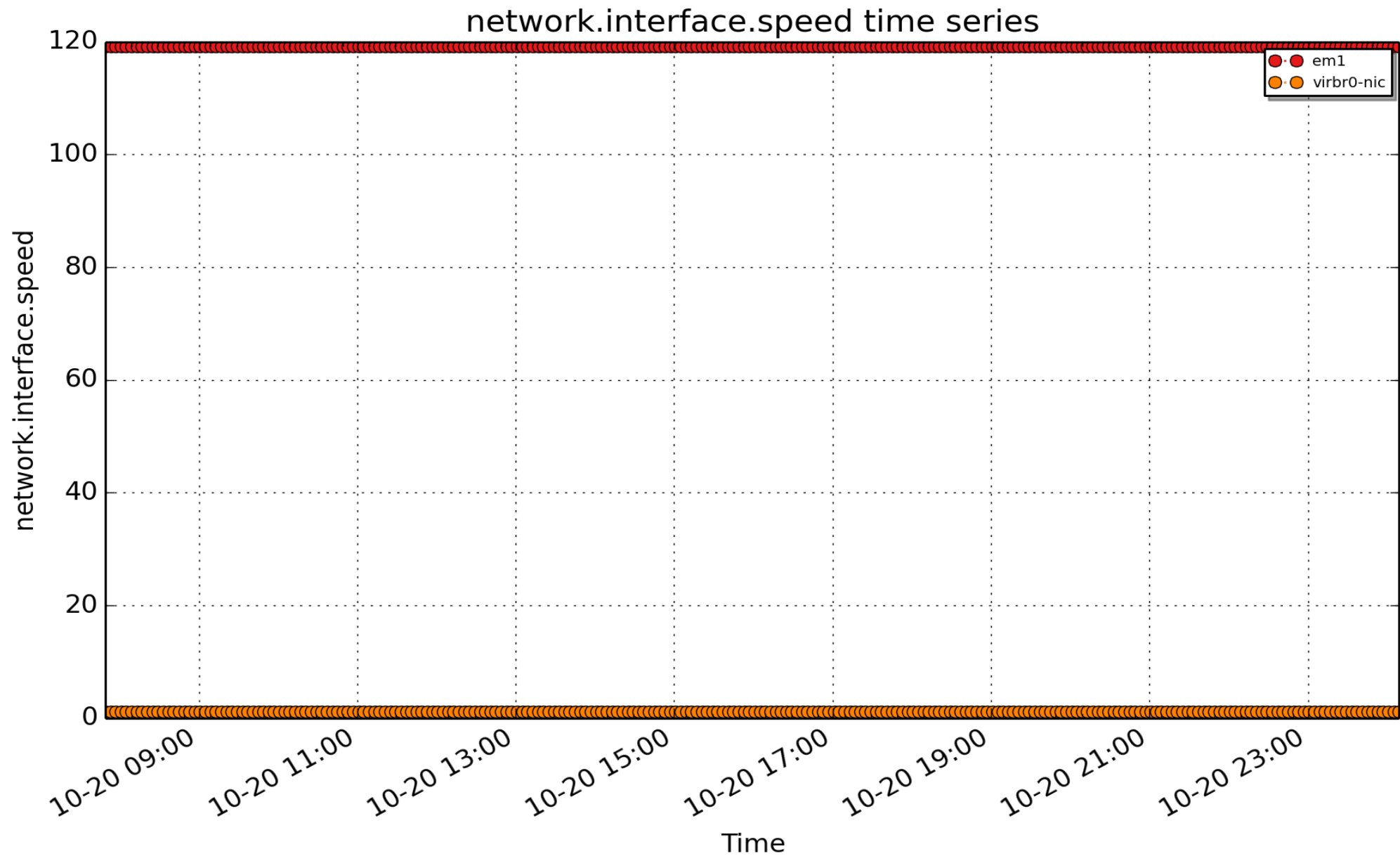
network.interface.mtu: maximum transmission unit on network interface (byte - U32)



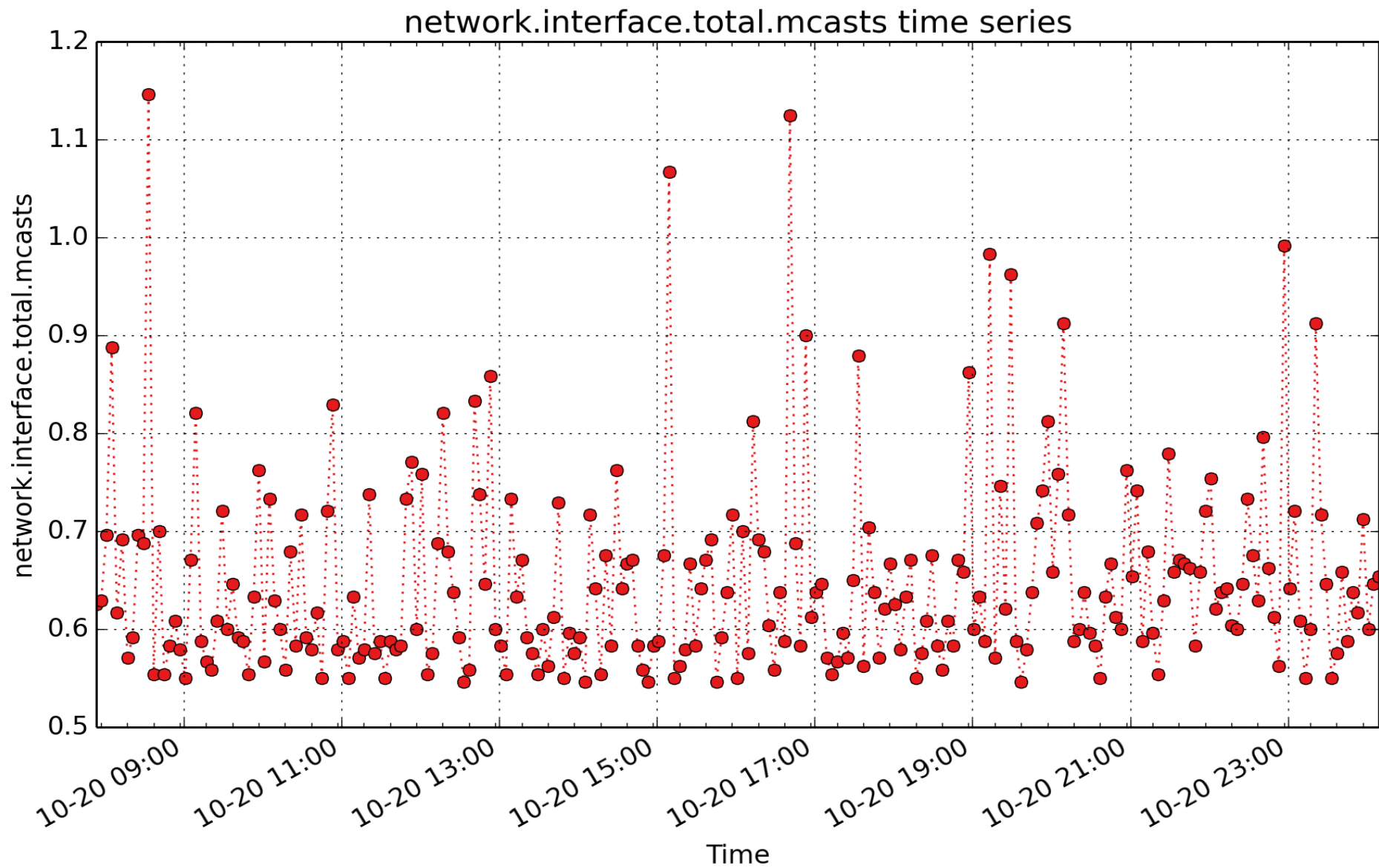
network.interface.out.bytes: network send bytes from /proc/net/dev per network interface (byte - U64) - *rate converted*



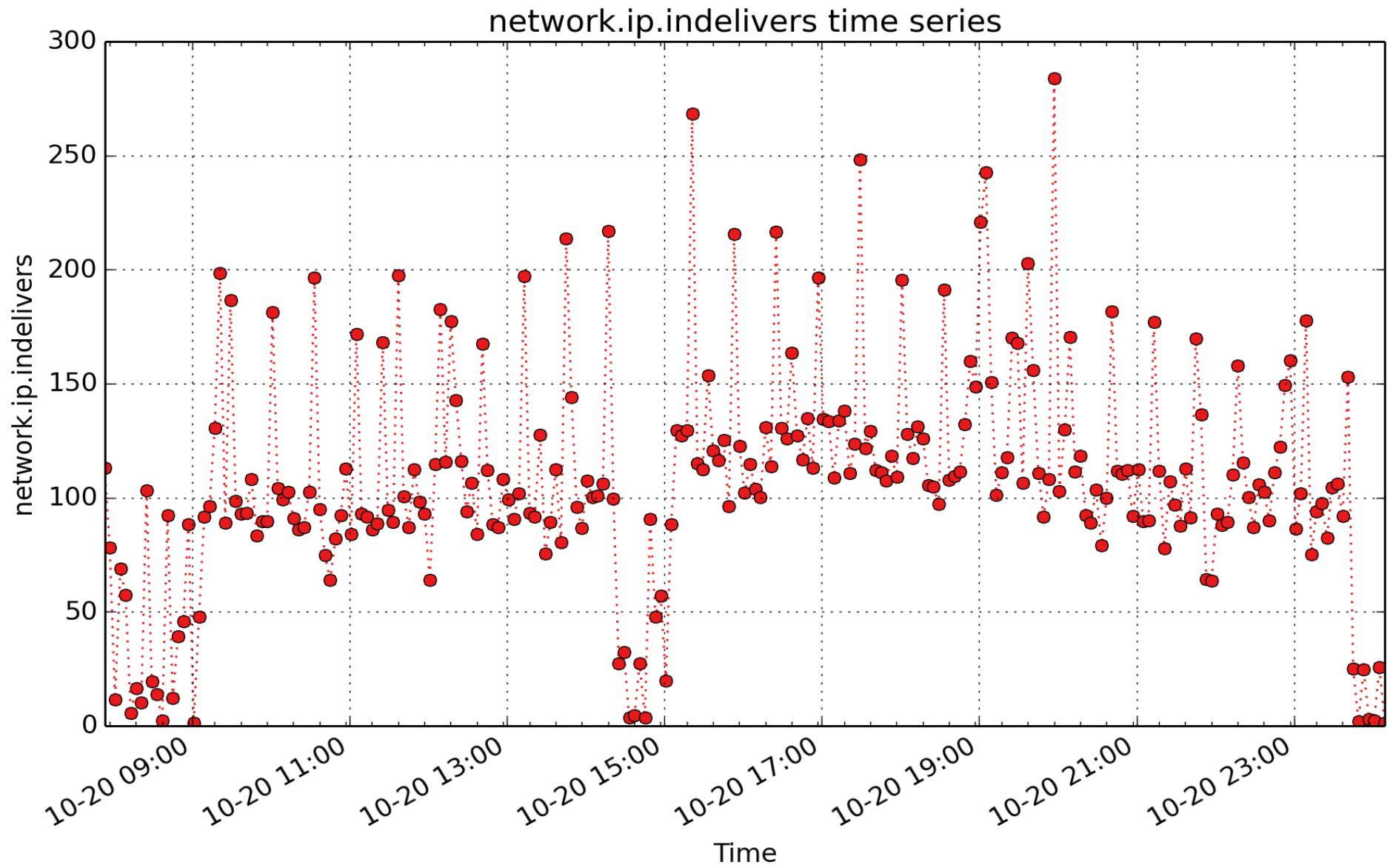
network.interface.out.packets: network send packets from /proc/net/dev per network interface (count - U64)
- rate converted



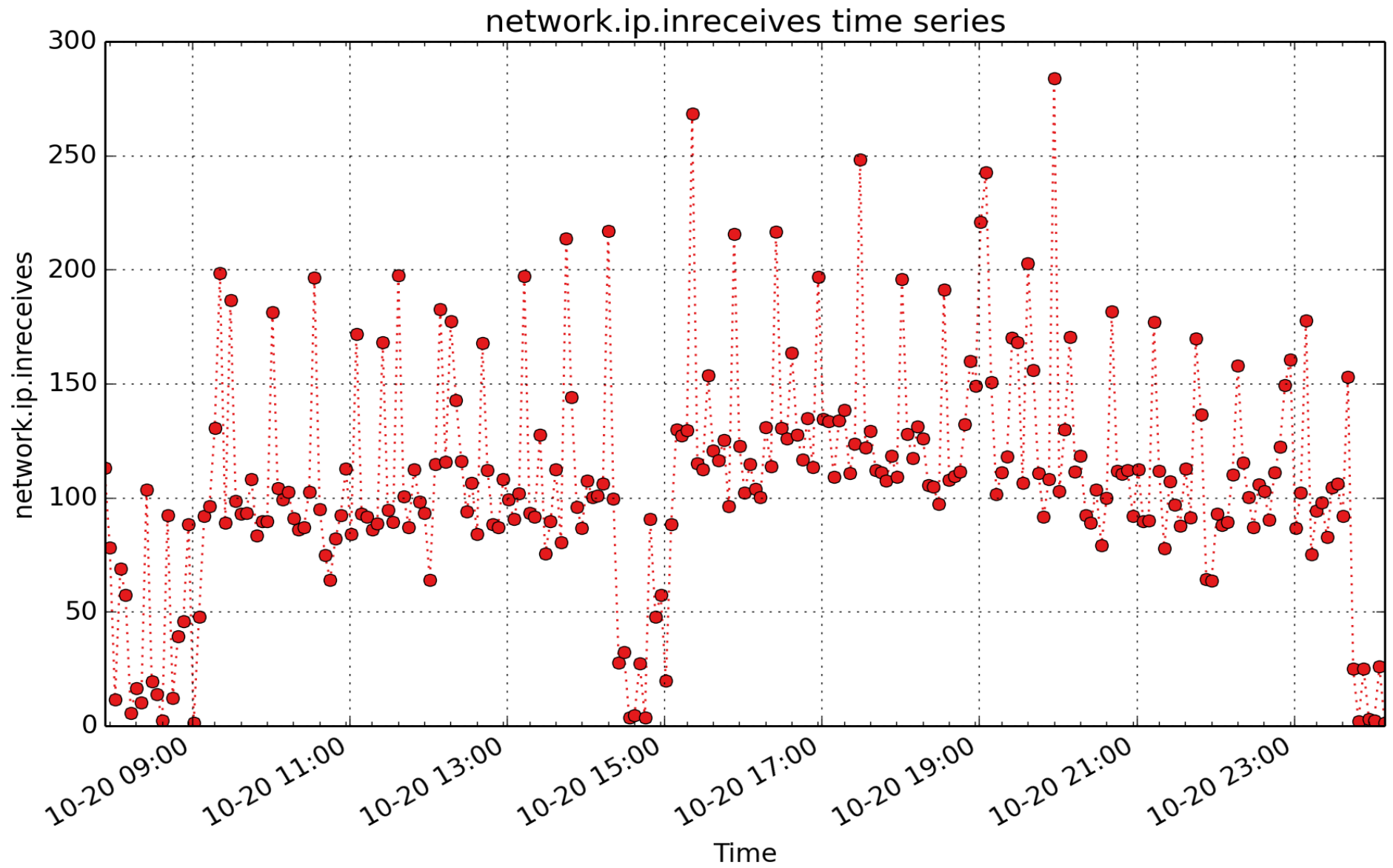
network.interface.speed: The linespeed on the network interface, as reported by the kernel, scaled from Megabits/second to Megabytes/second. See also `network.interface.baudrate` for the bytes/second value. (Mbyte / sec - FLOAT)



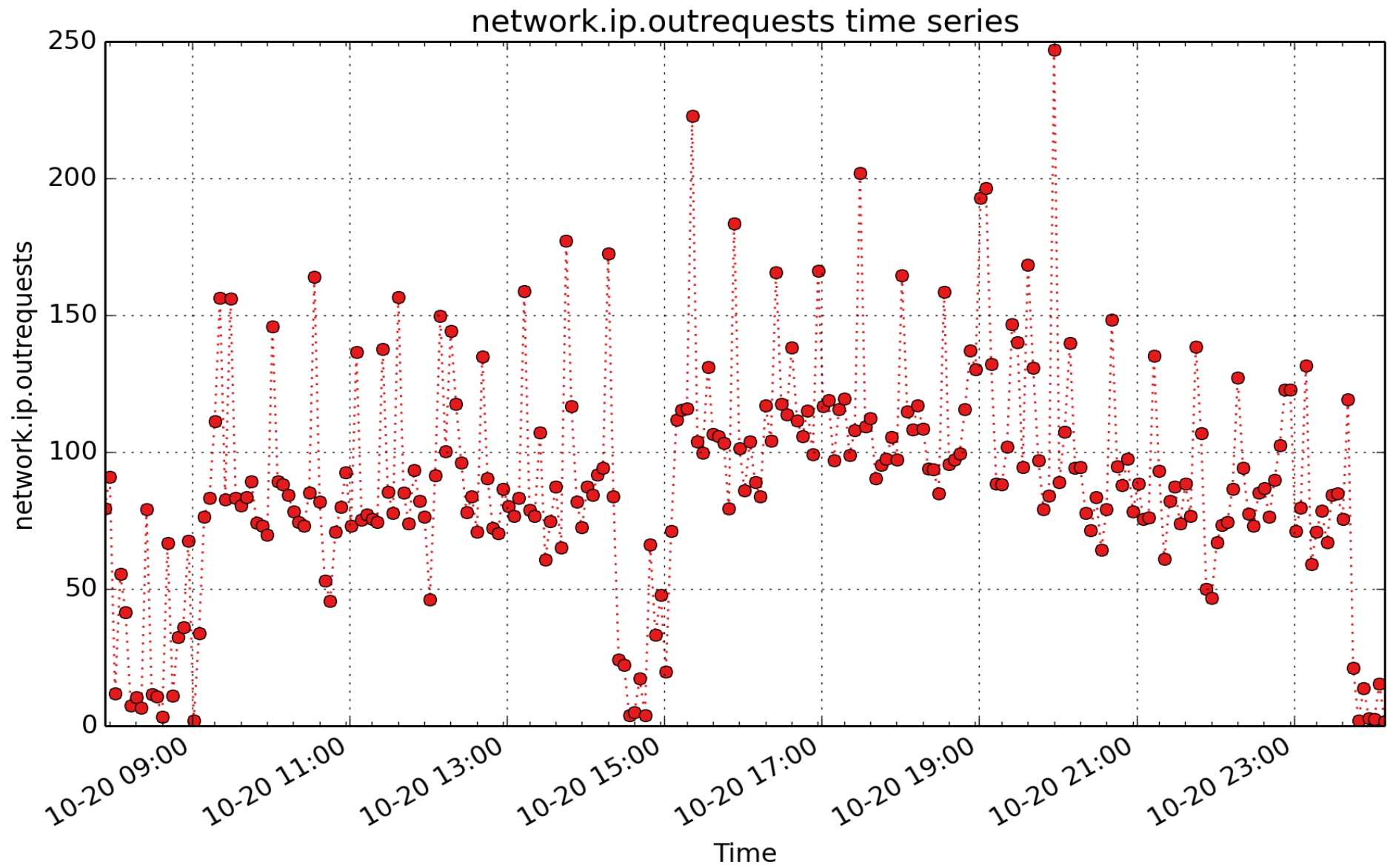
network.interface.total.mcasts: network total (in+out) mcasts from /proc/net/dev per network interface
(count - U64) - *rate converted*



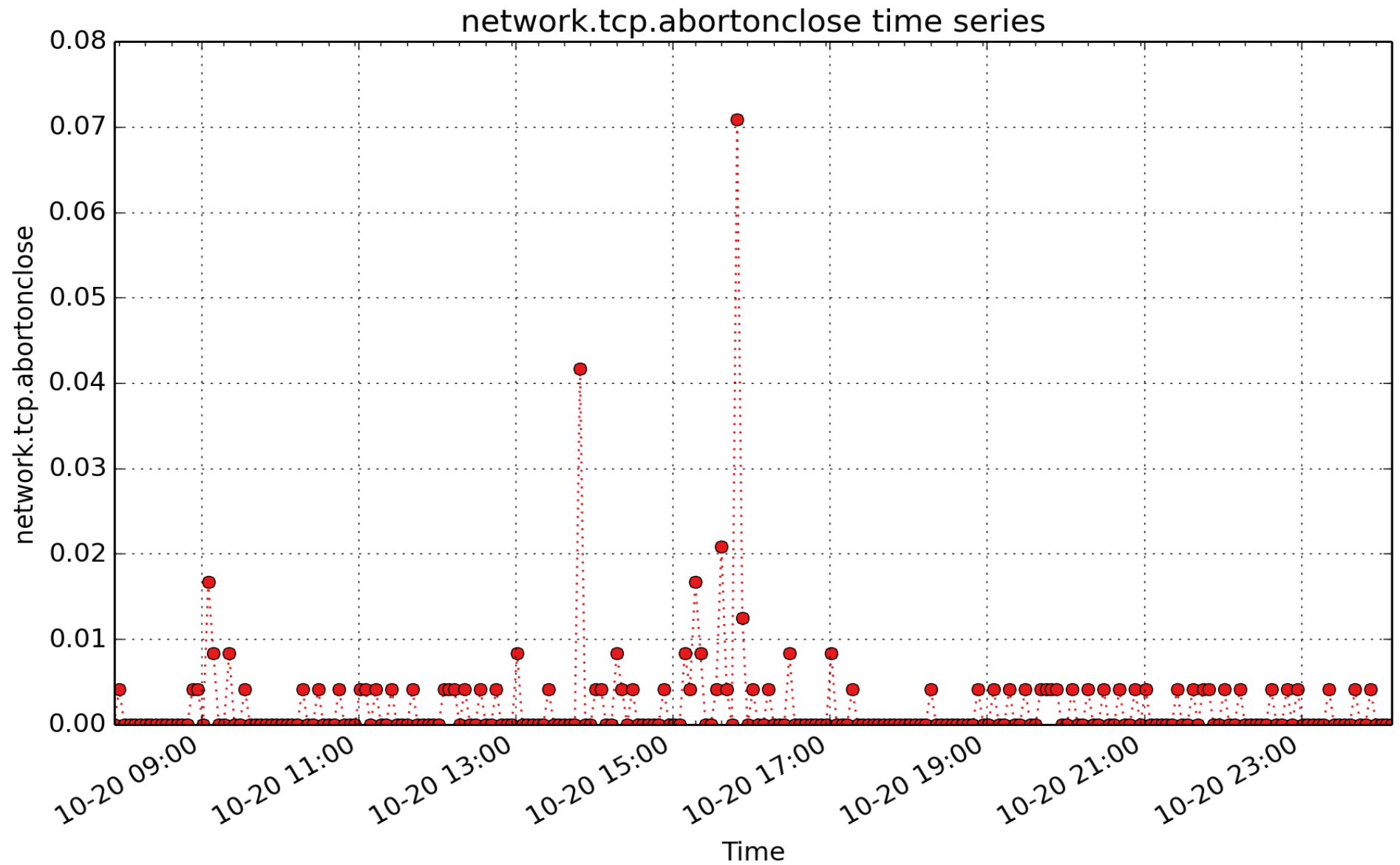
network.ip.indelivers: count of ip indelivers (count - U64) - *rate converted*



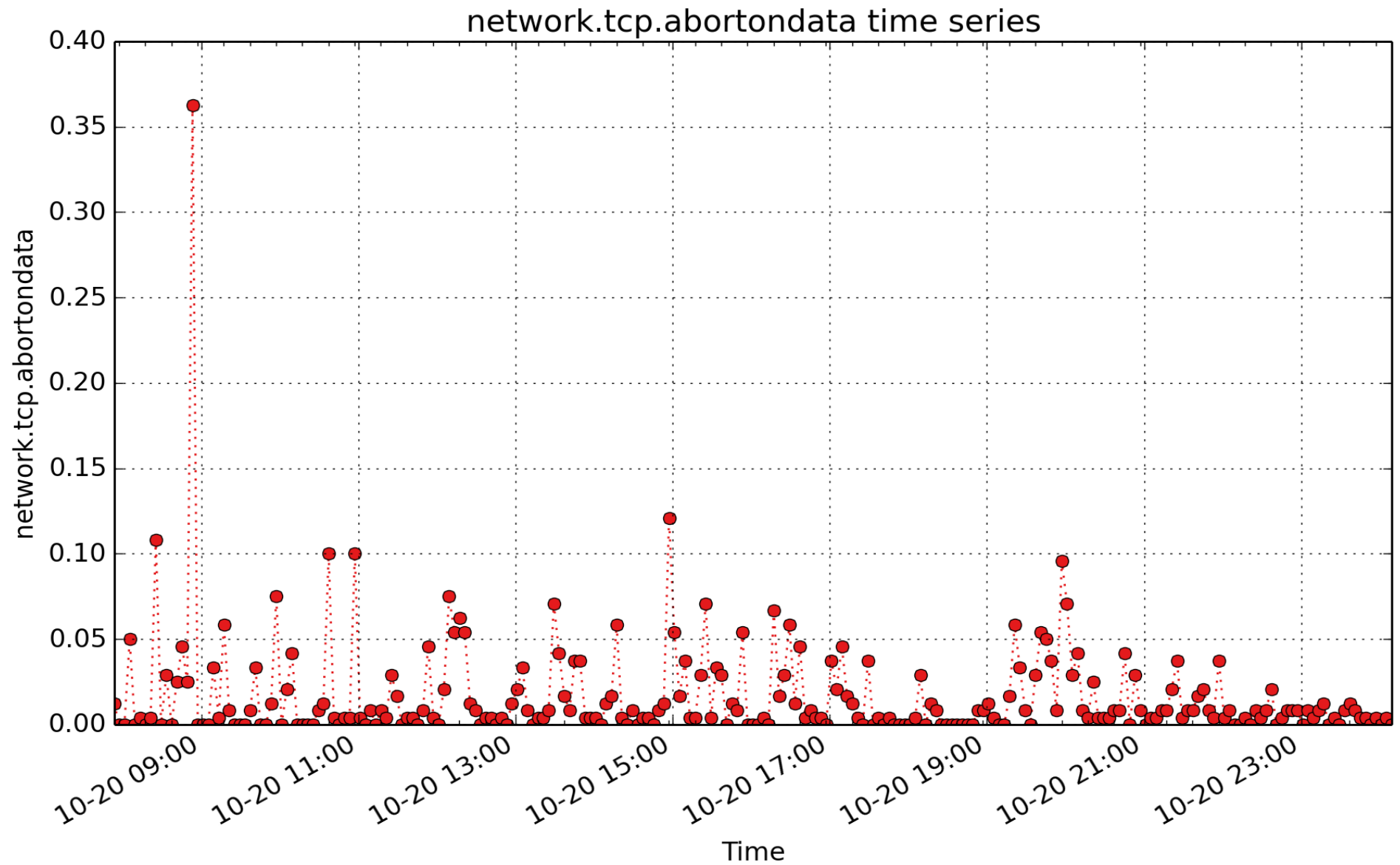
network.ip.inreceives: count of ip inreceives (count - U64) - *rate converted*



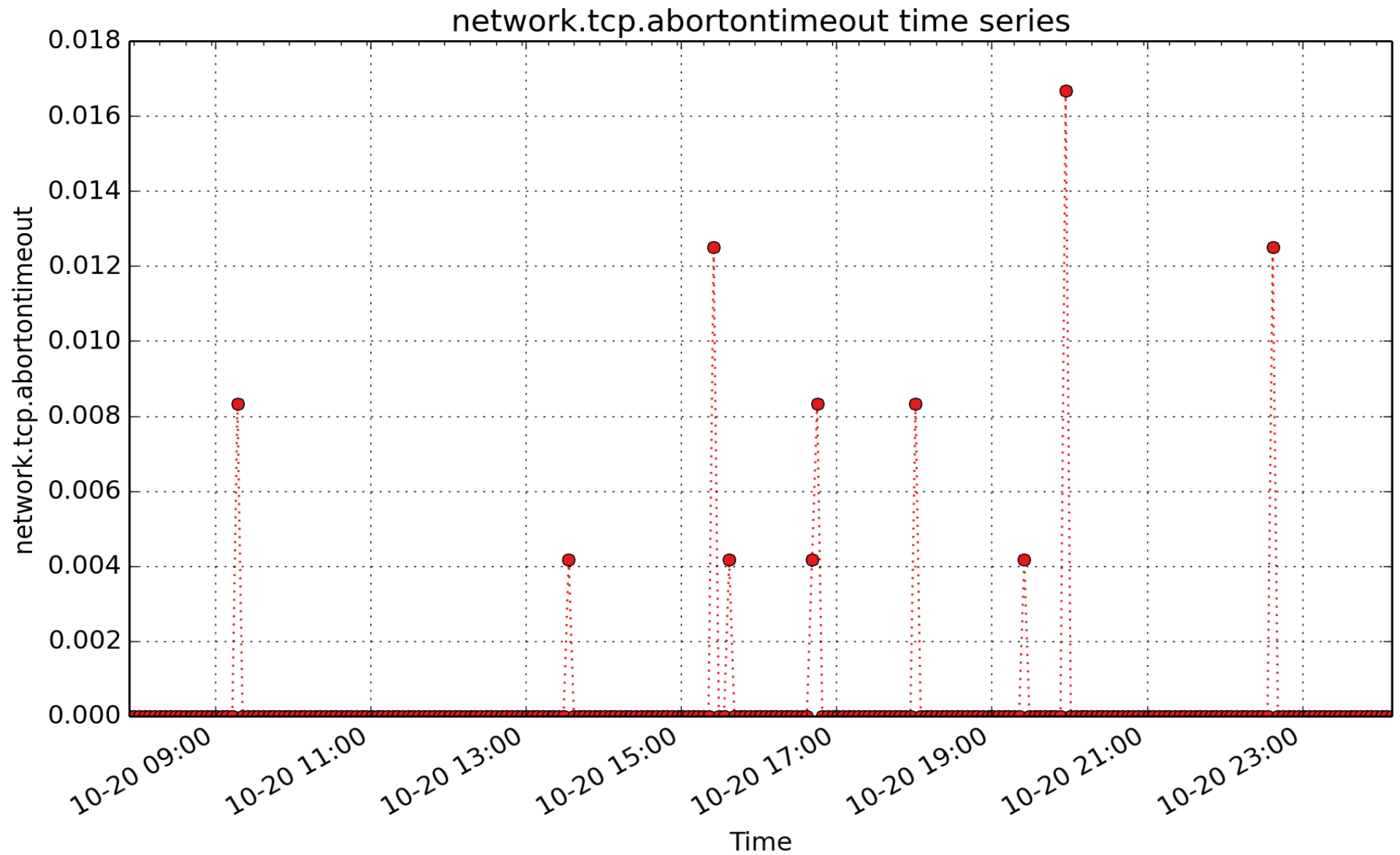
network.ip.outrequests: count of ip outrequests (count - U64) - *rate converted*



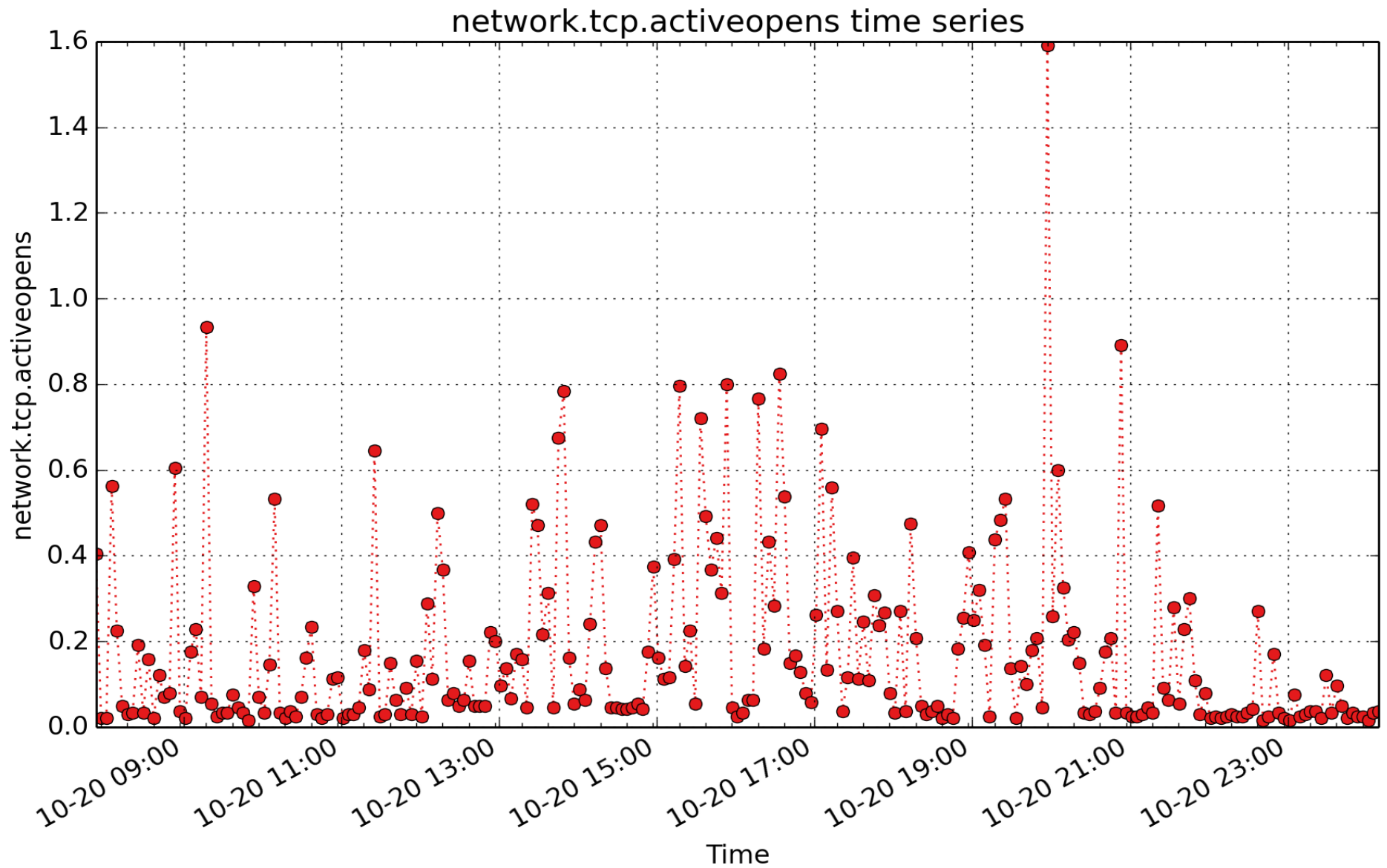
network.tcp.abortonclose: Number of connections reset due to early user close (count - U64) - *rate converted*



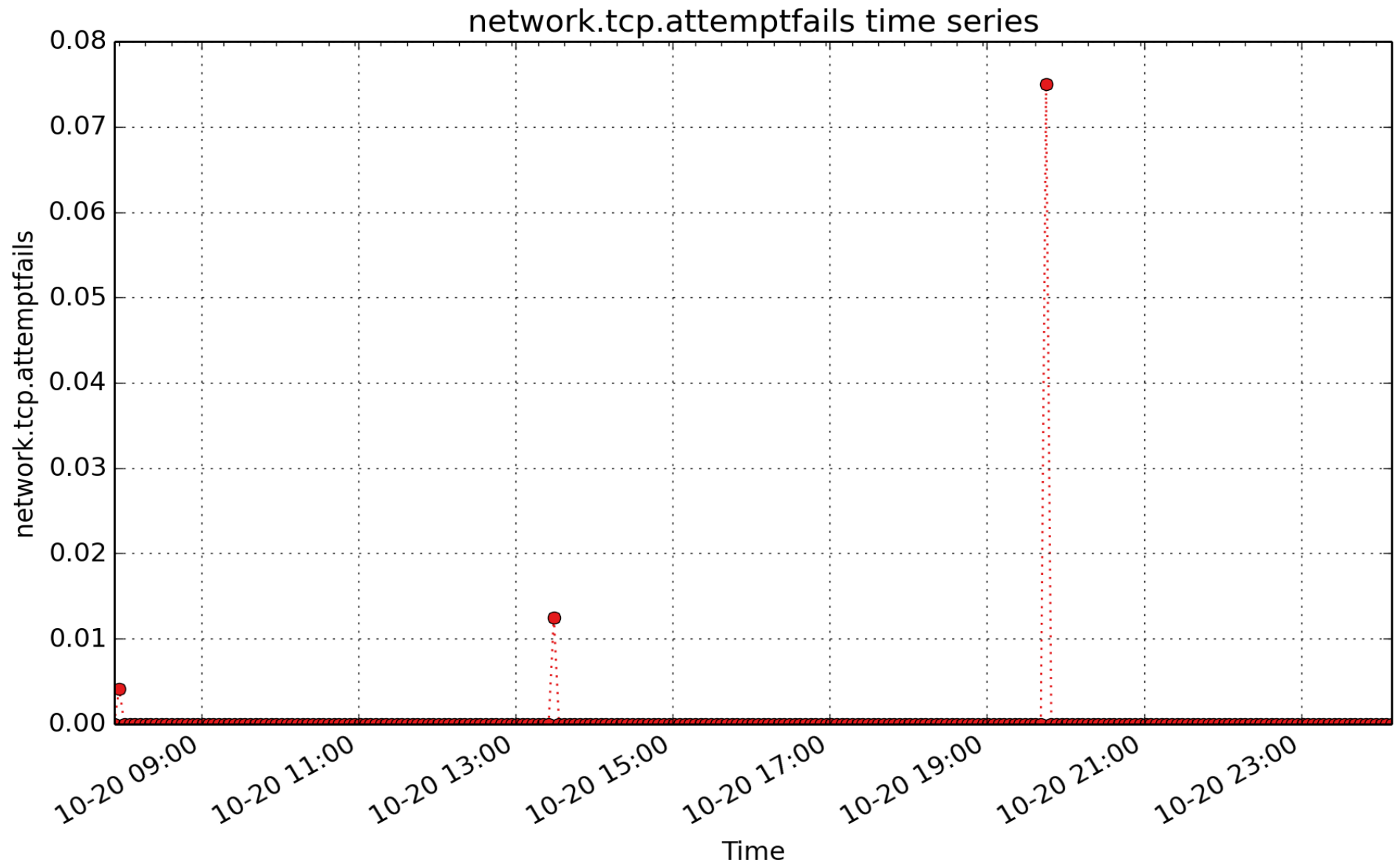
network.tcp.abortondata: Number of connections reset due to unexpected data (count - U64) - *rate converted*



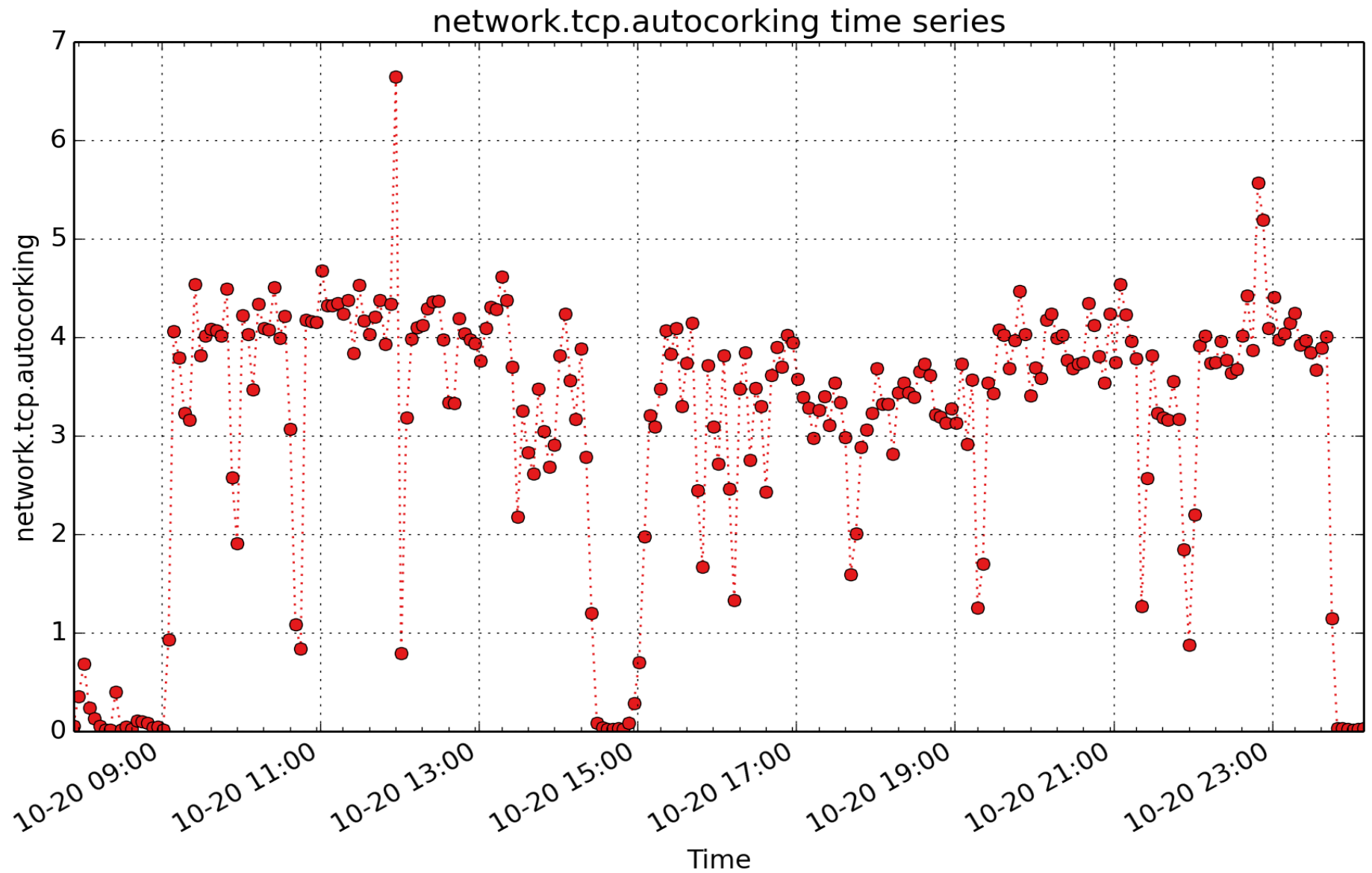
network.tcp.abortontimeout: Number of connections aborted due to timeout (count - U64) - *rate converted*



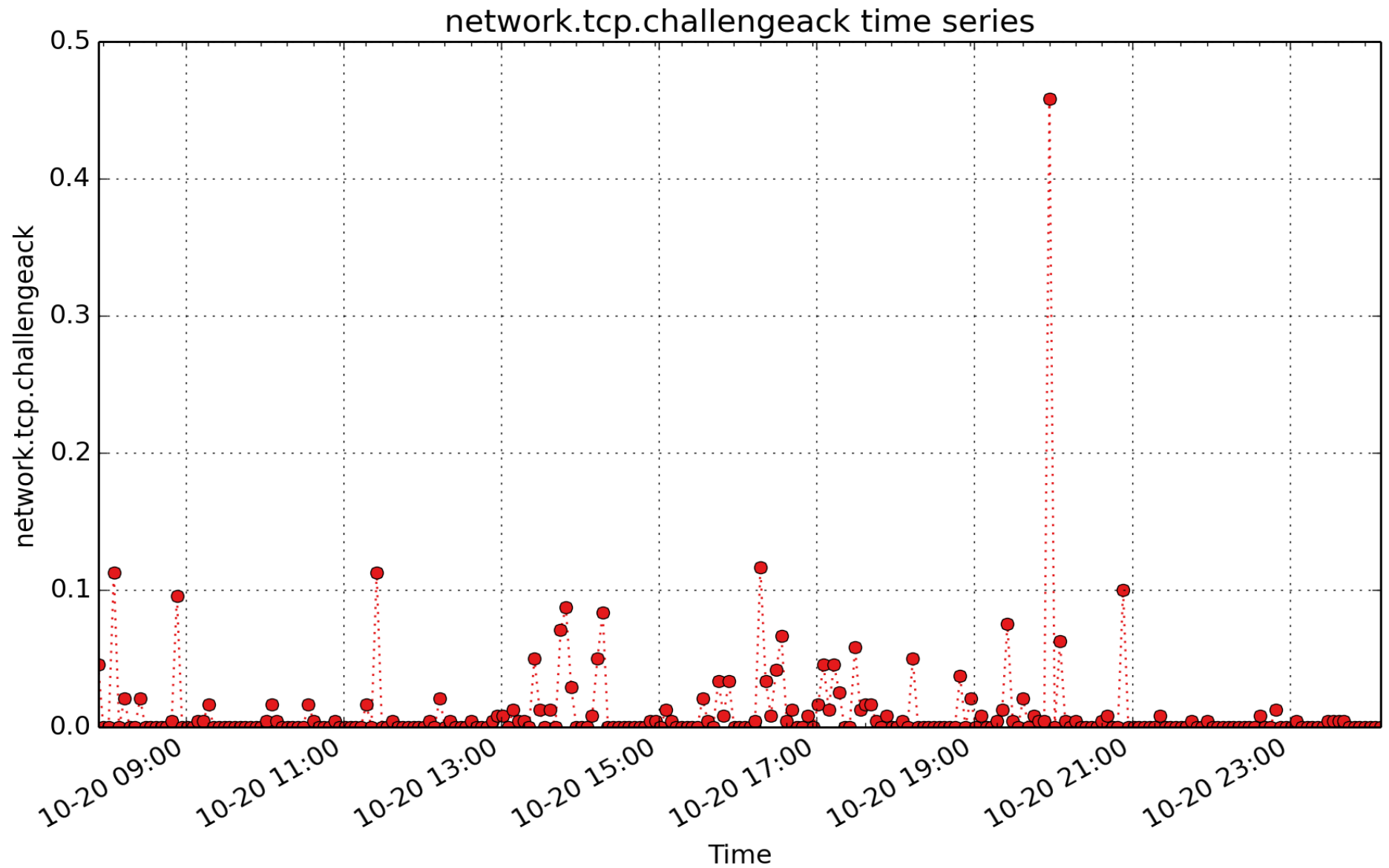
network.tcp.activeopens: count of tcp activeopens (count - U64) - *rate converted*



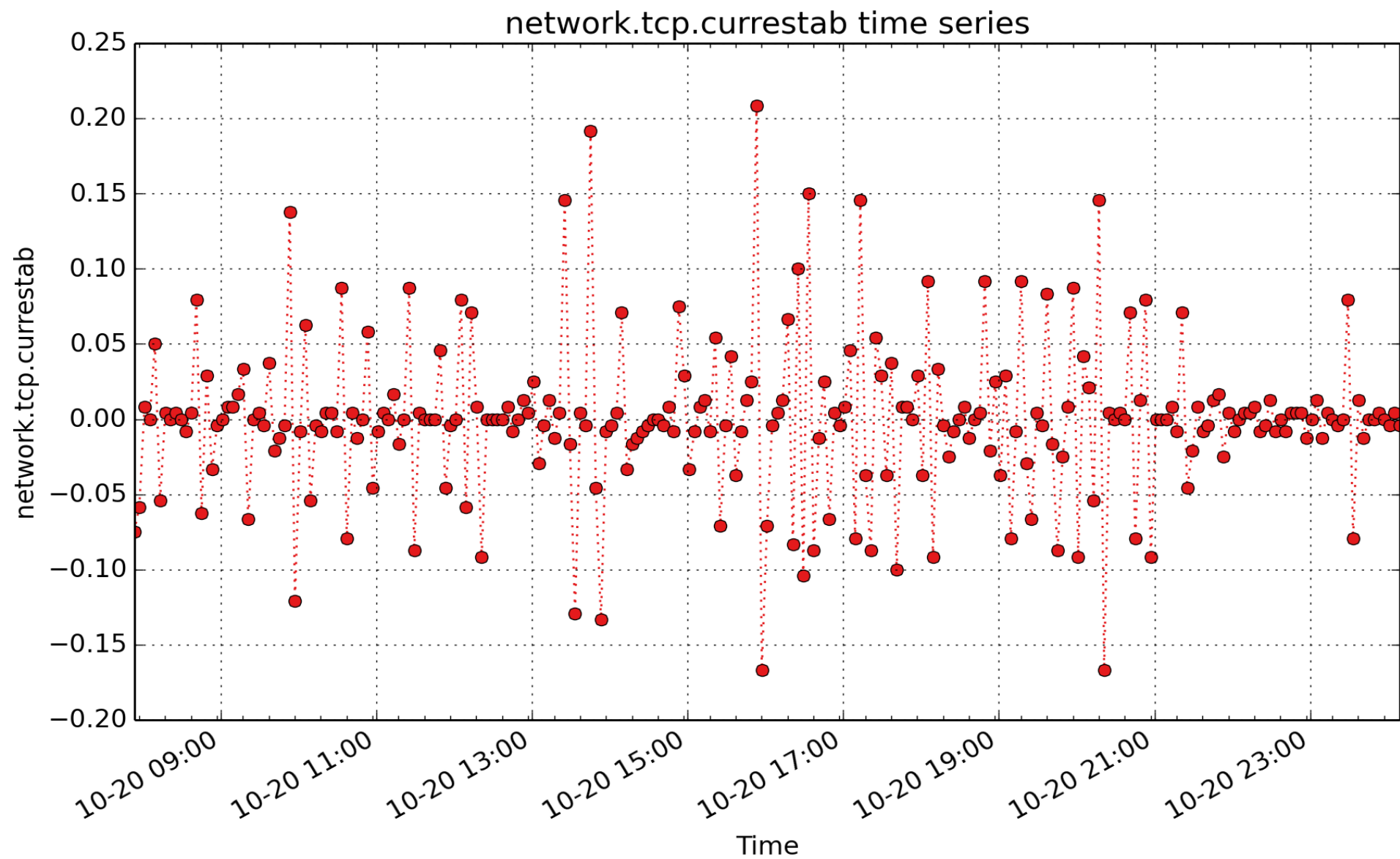
network.tcp.attemptfails: count of tcp attemptfails (count - U64) - *rate converted*



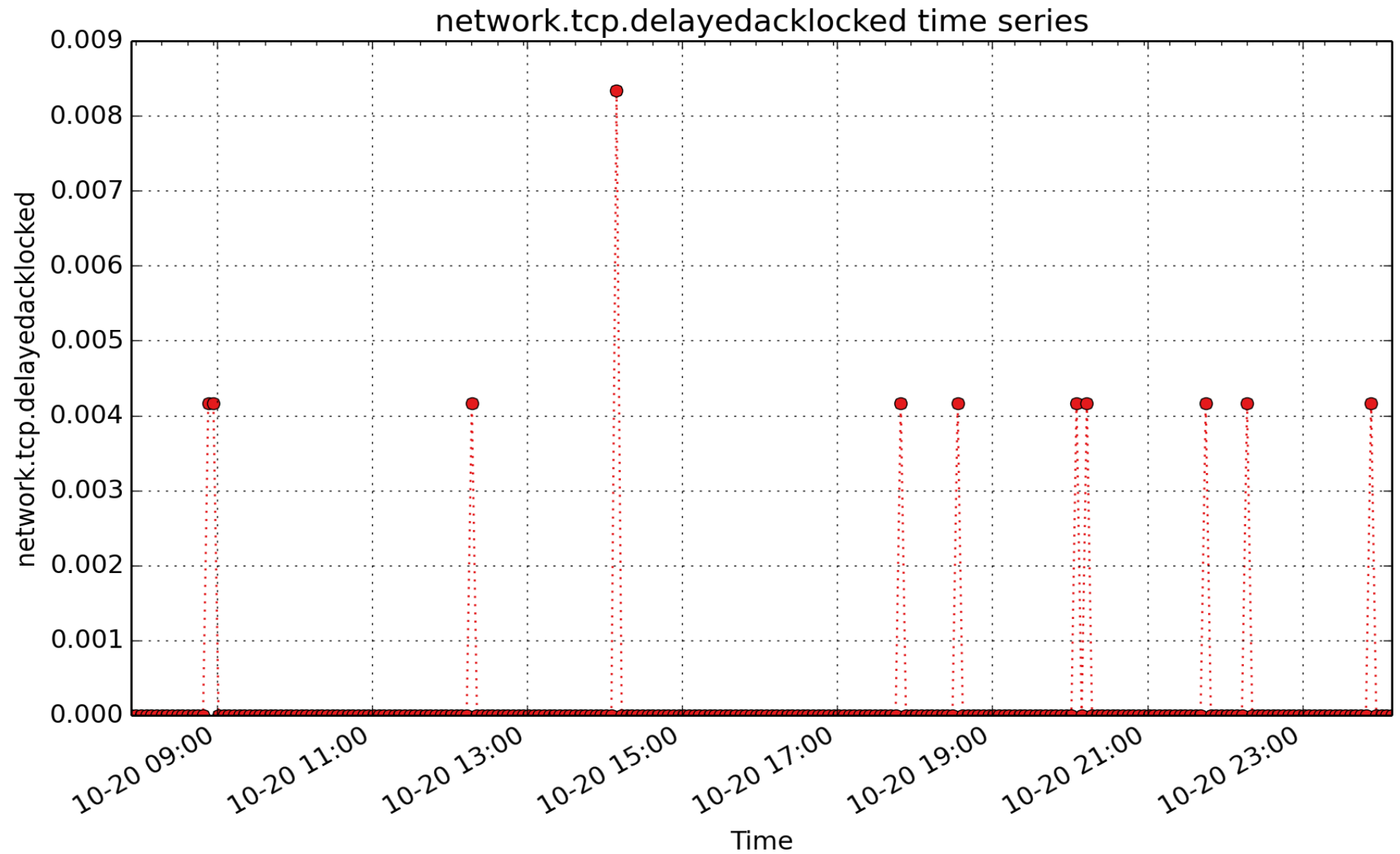
network.tcp.autocorking: Number of times stack detected skb was underused and its flush was deferred (count - U64) - *rate converted*



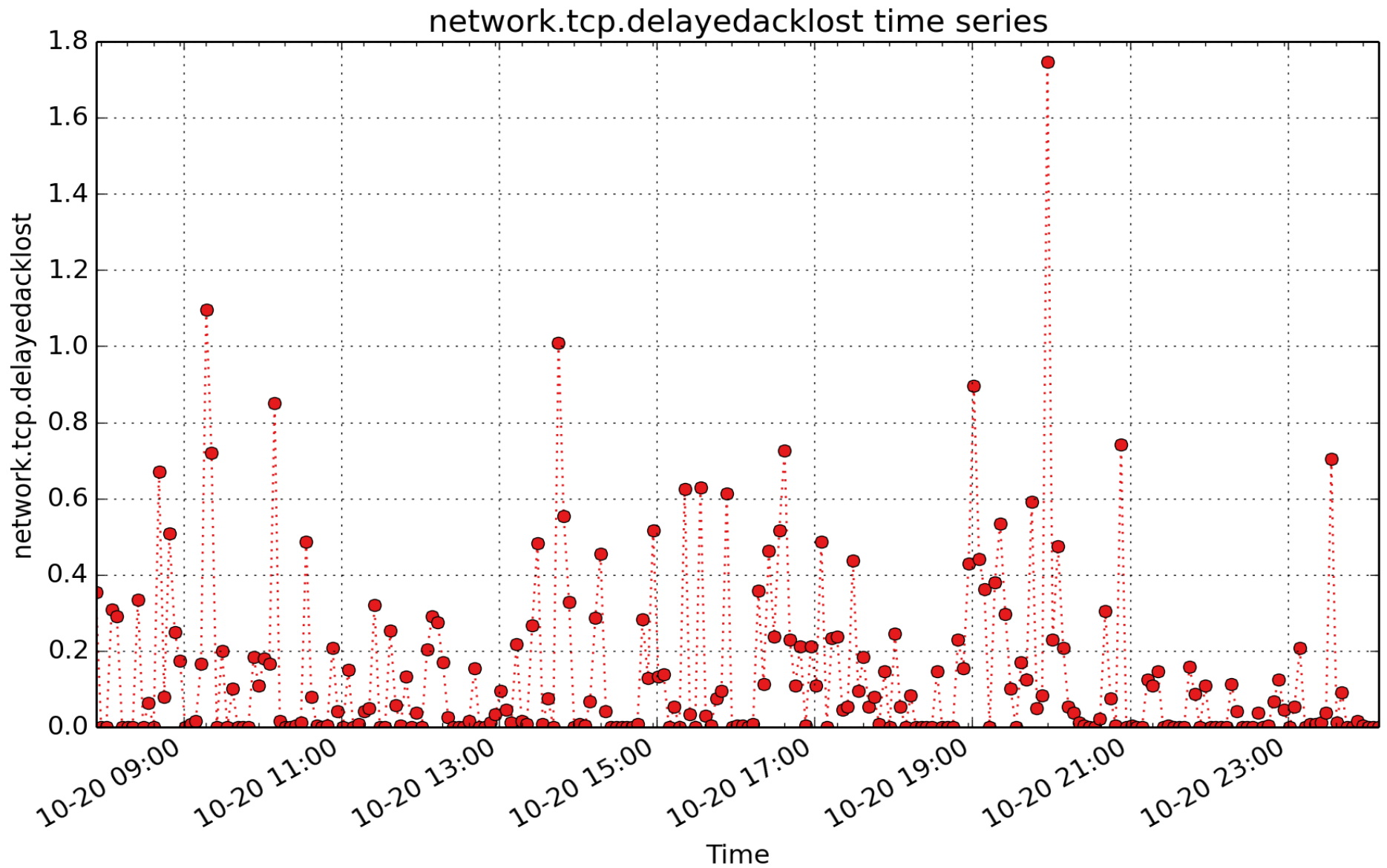
network.tcp.challengeack: Number of challenge ACKs sent (RFC 5961 3.2) (count - U64) - *rate converted*



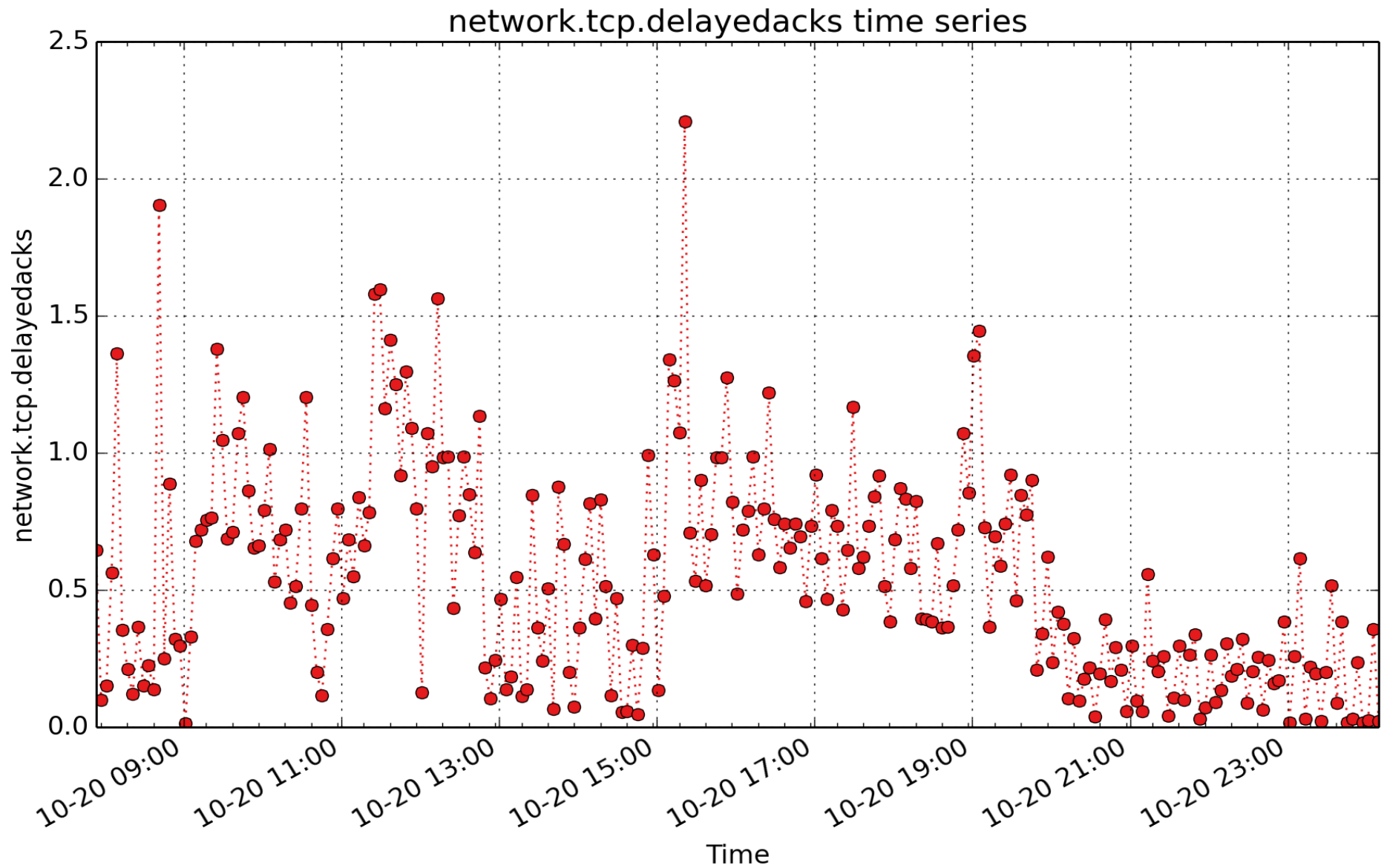
network.tcp.currestab: count of tcp currestab (count - U64) - *rate converted*



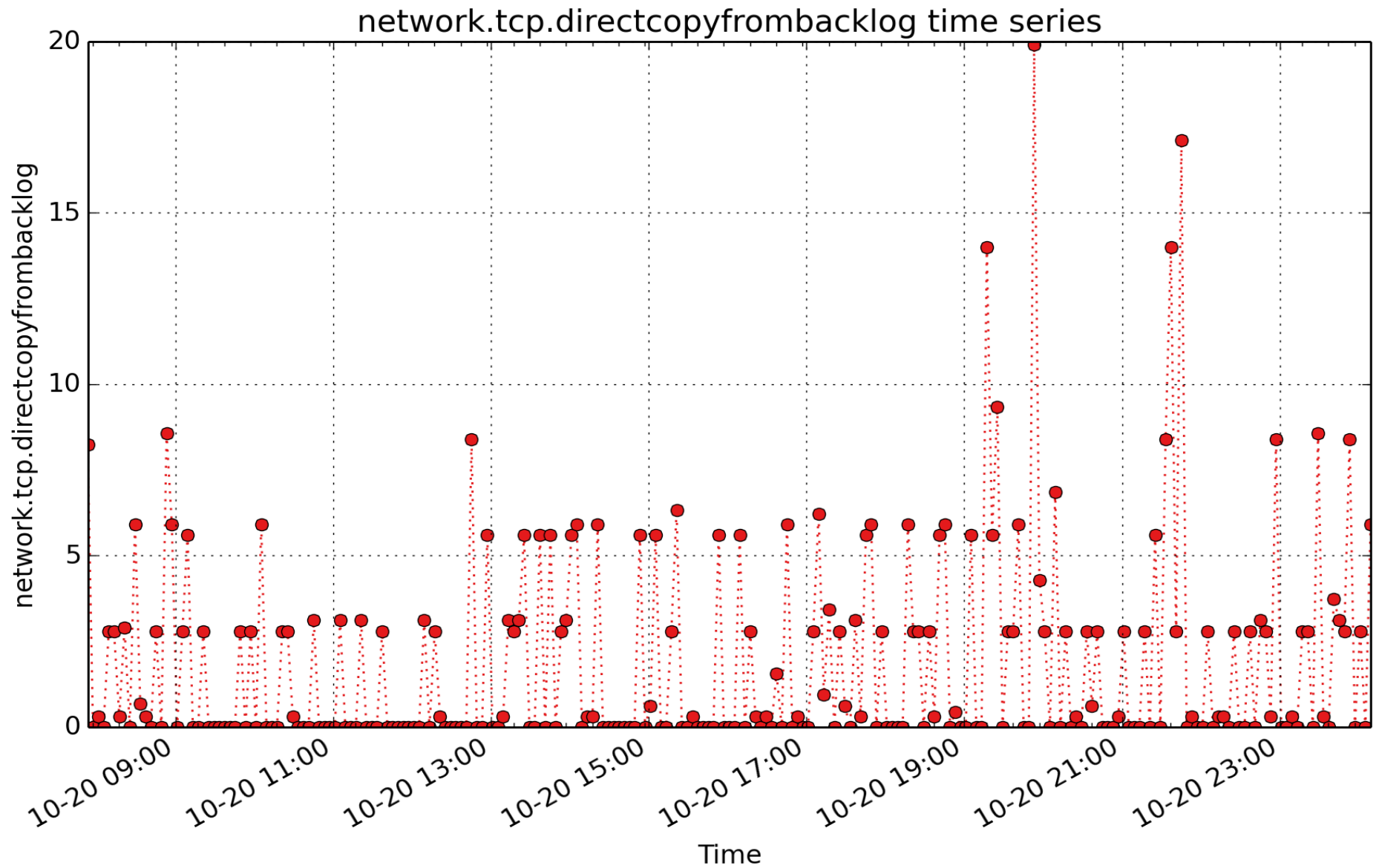
network.tcp.delayedacklocked: Number of delayed acks further delayed because of locked socket (count - U64) - *rate converted*



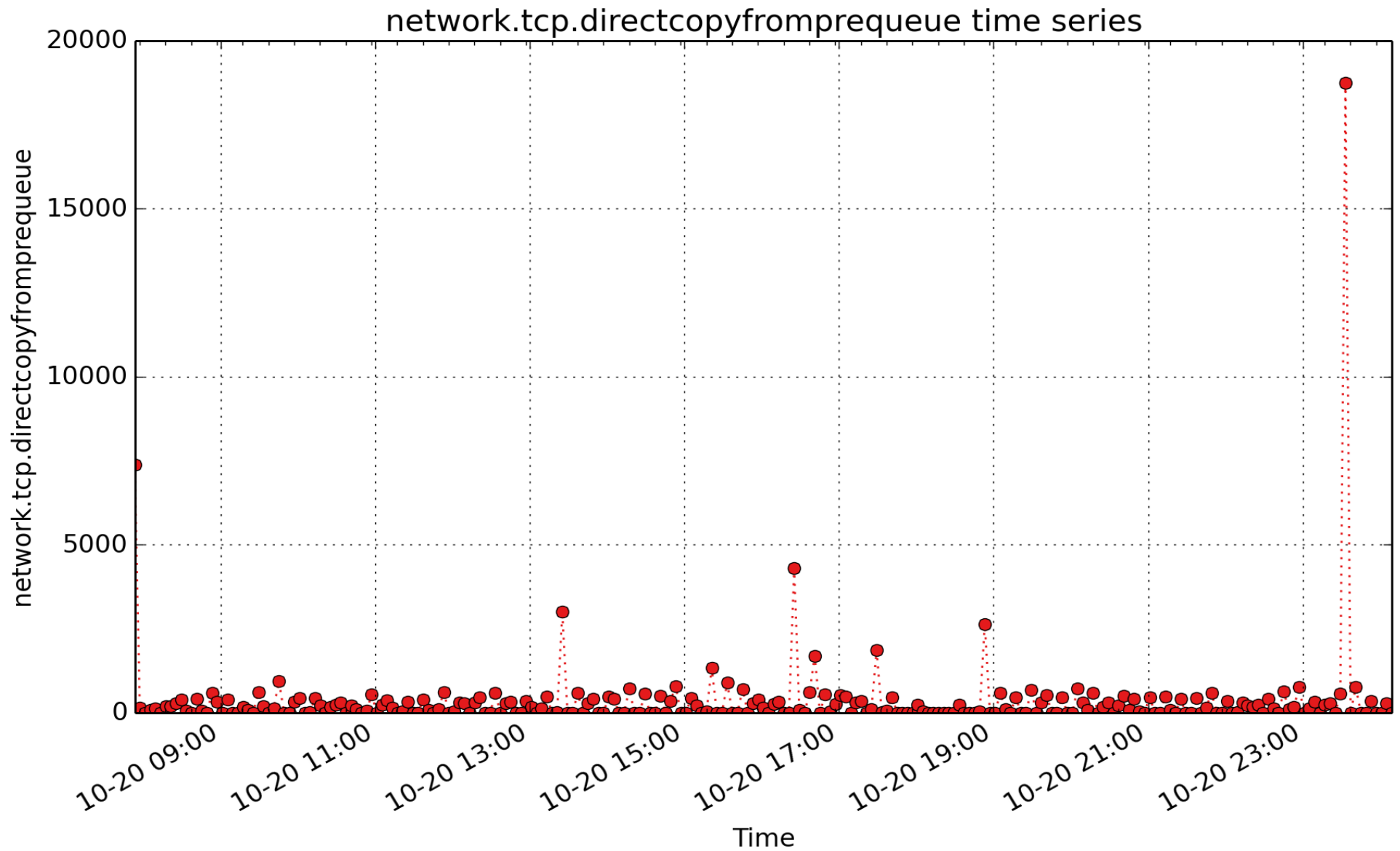
network.tcp.delayedacklost: Number of times quick ack mode was activated times (count - U64) - *rate converted*



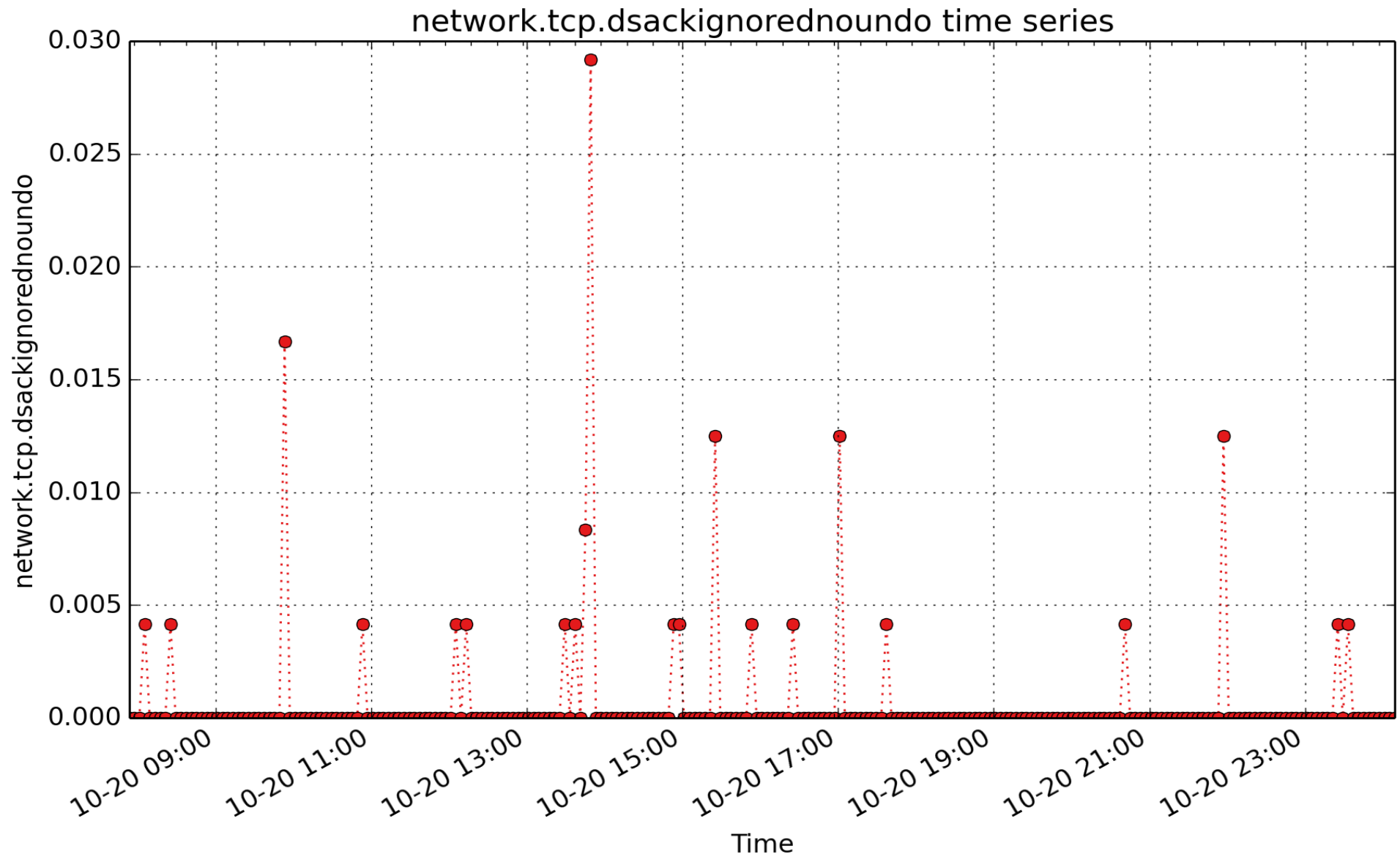
network.tcp.delayedacks: Number of delayed acks sent (count - U64) - *rate converted*



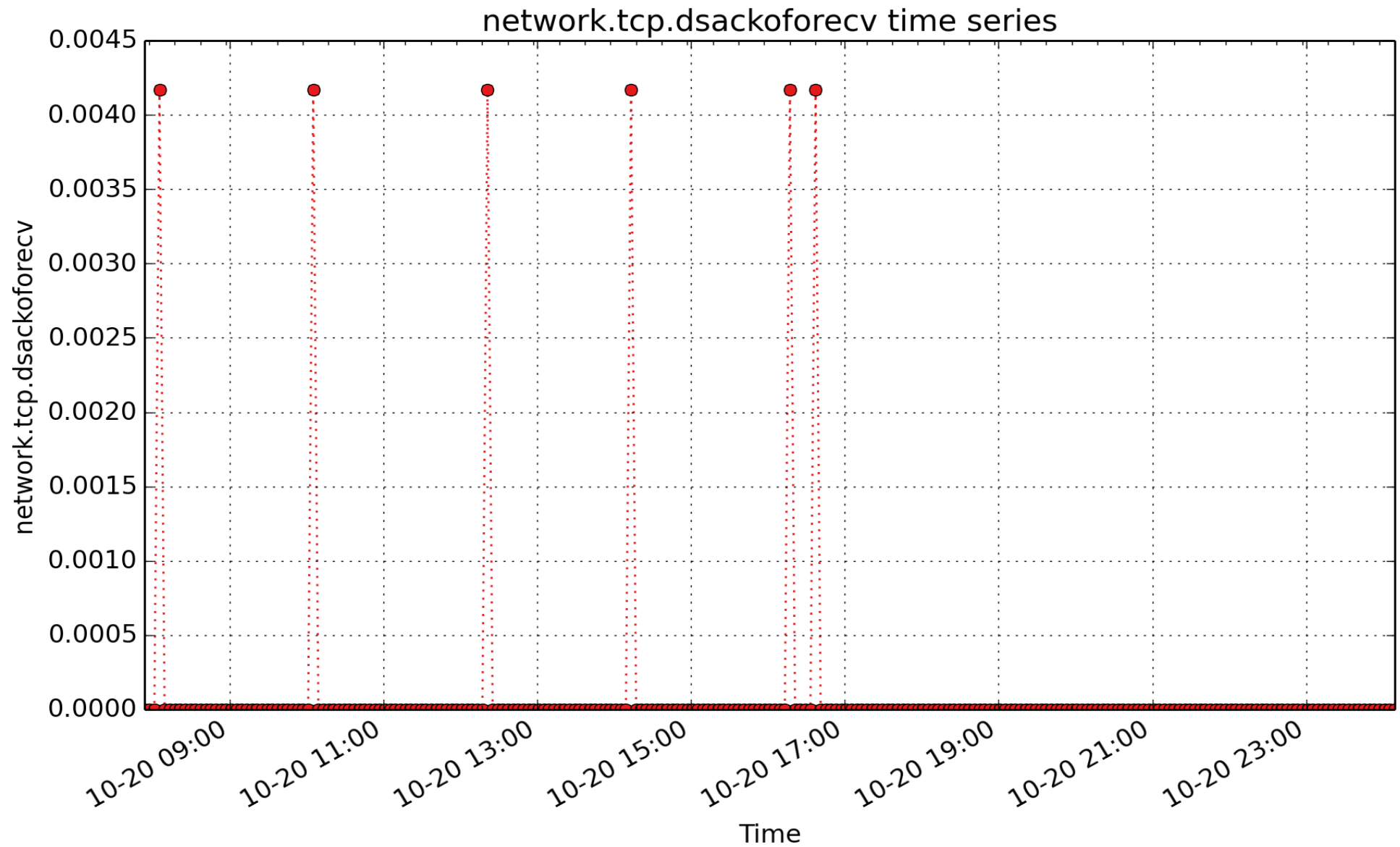
network.tcp.directcopyfrombacklog: Number of bytes directly in process context from backlog (count - U64)
- *rate converted*



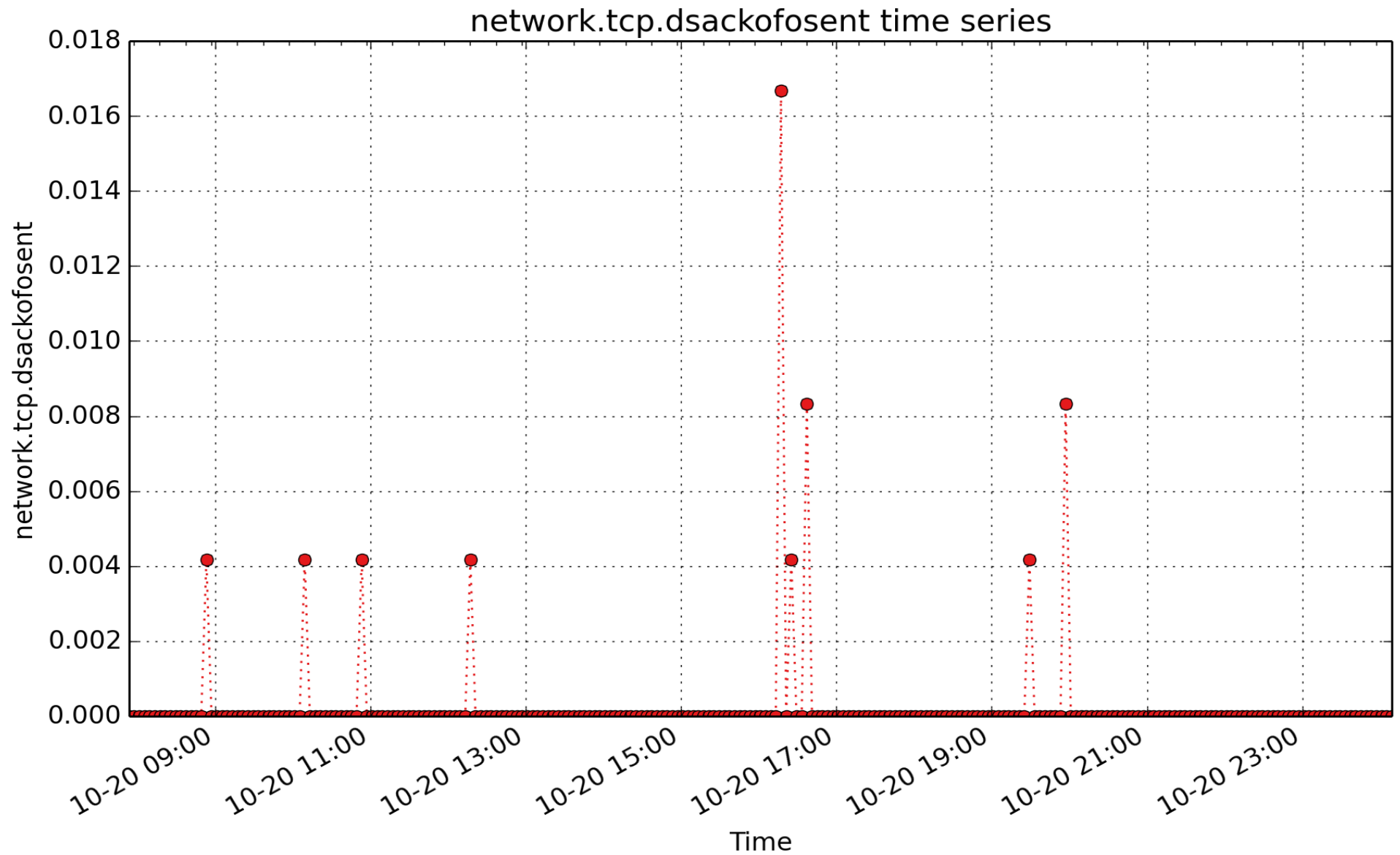
network.tcp.directcopyfromprequeue: Number of bytes directly received in process context from prequeue (count - U64) - *rate converted*



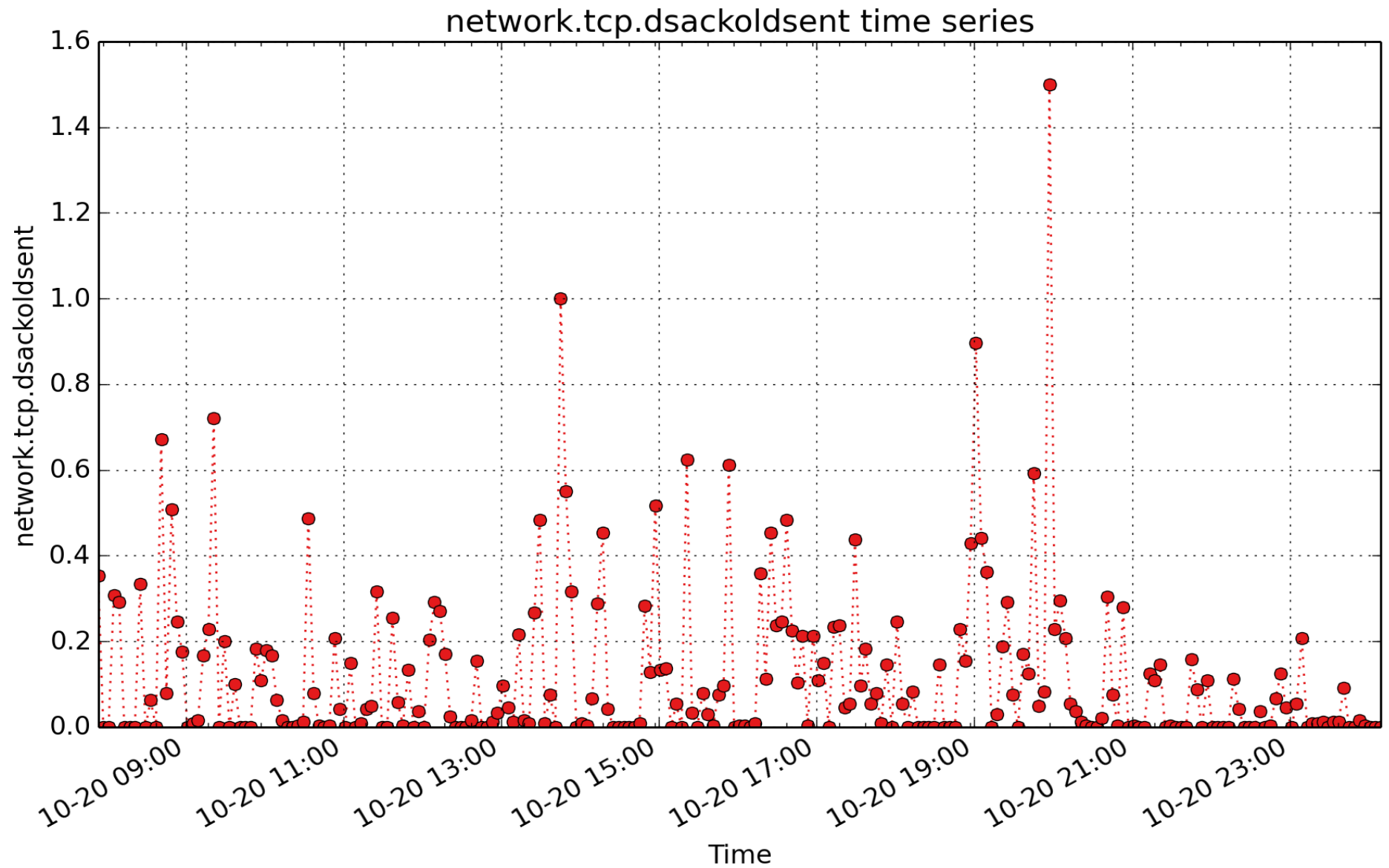
network.tcp.dsackignorednoundo: Number of ignored duplicate SACKs with undo_marker not set (count - U64) - *rate converted*



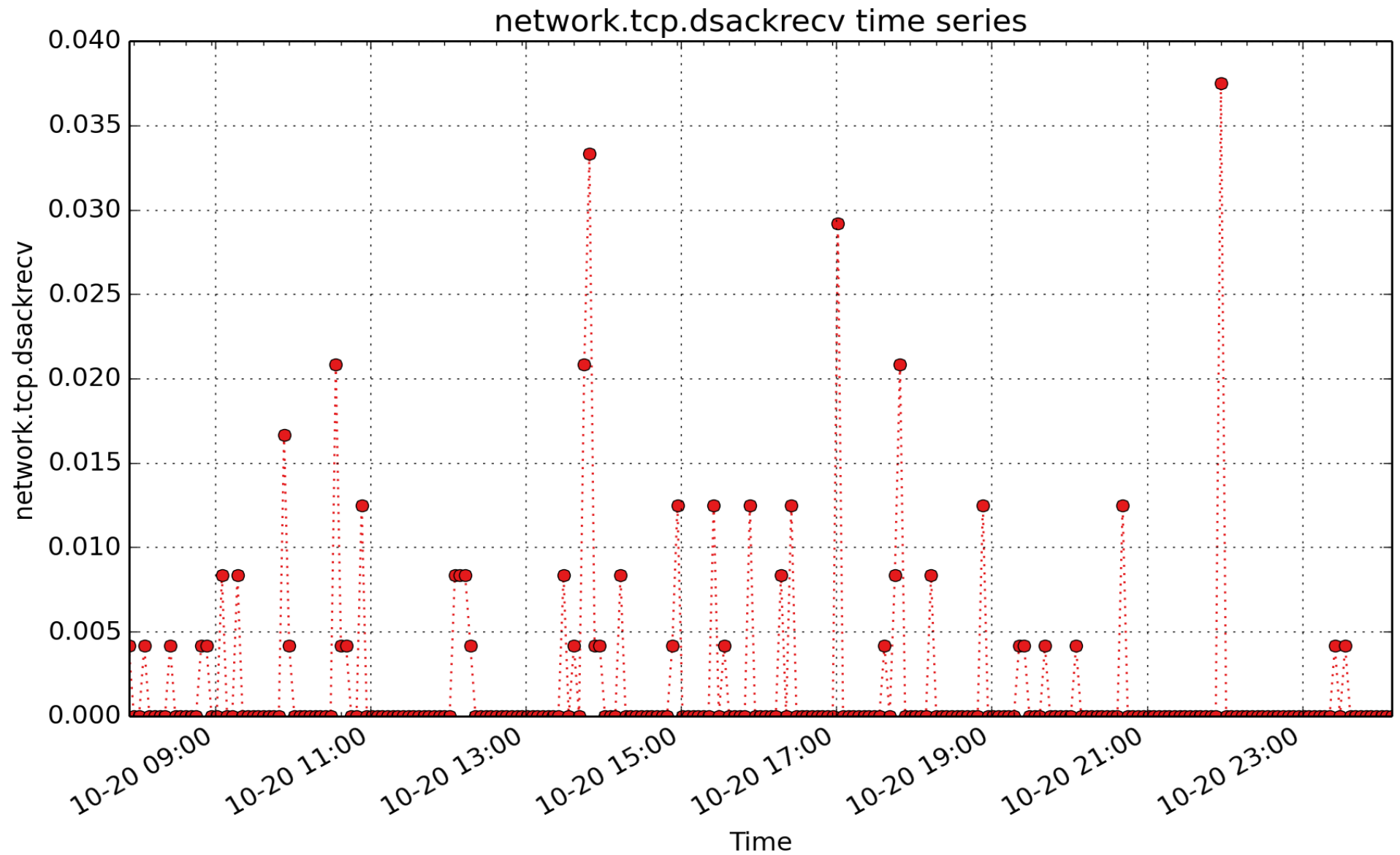
network.tcp.dsackoforecv: Number of DSACKs for out of order packets received (count - U64) - *rate converted*



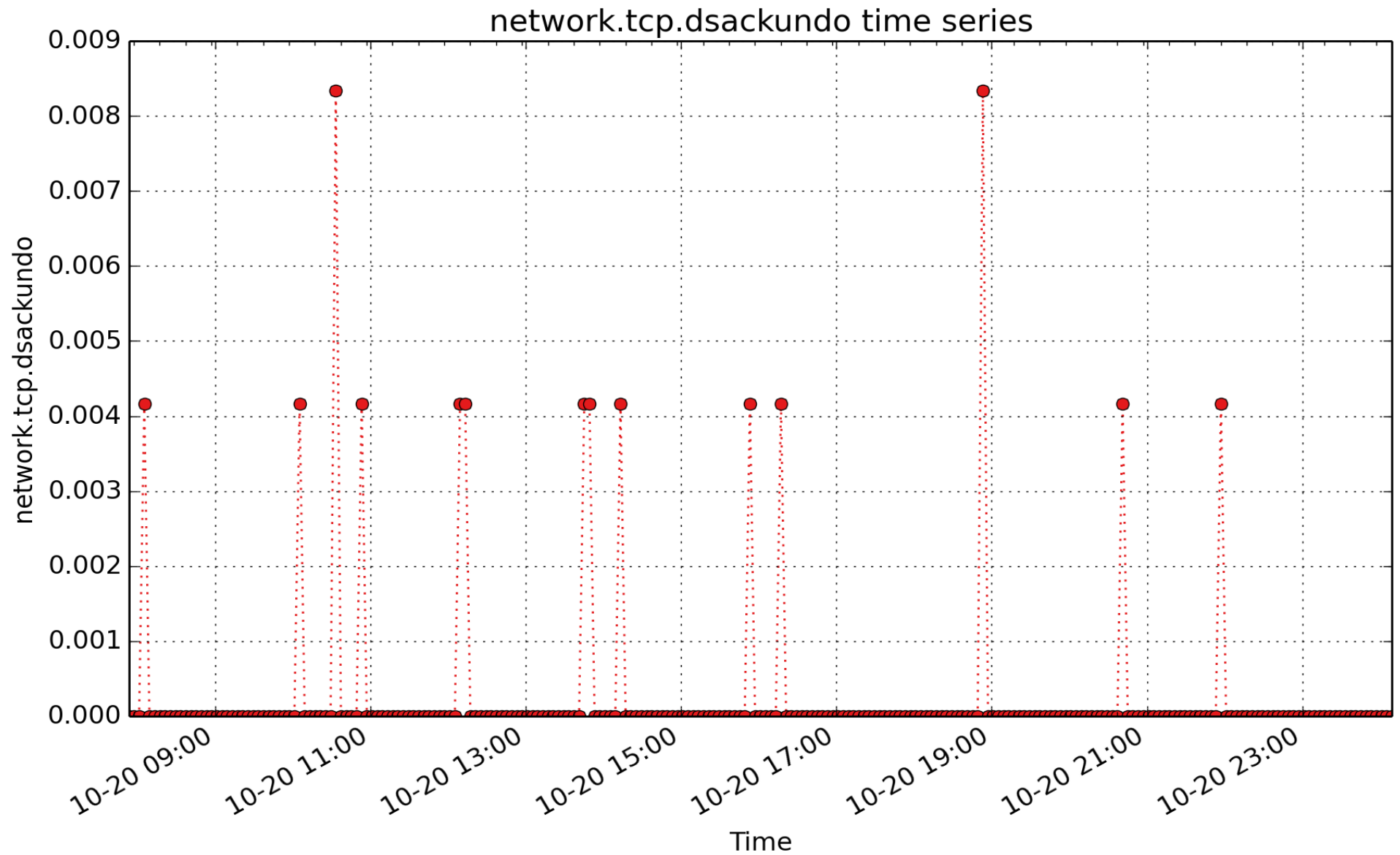
network.tcp.dsackofosent: Number of DSACKs sent for out of order packets (count - U64) - *rate converted*



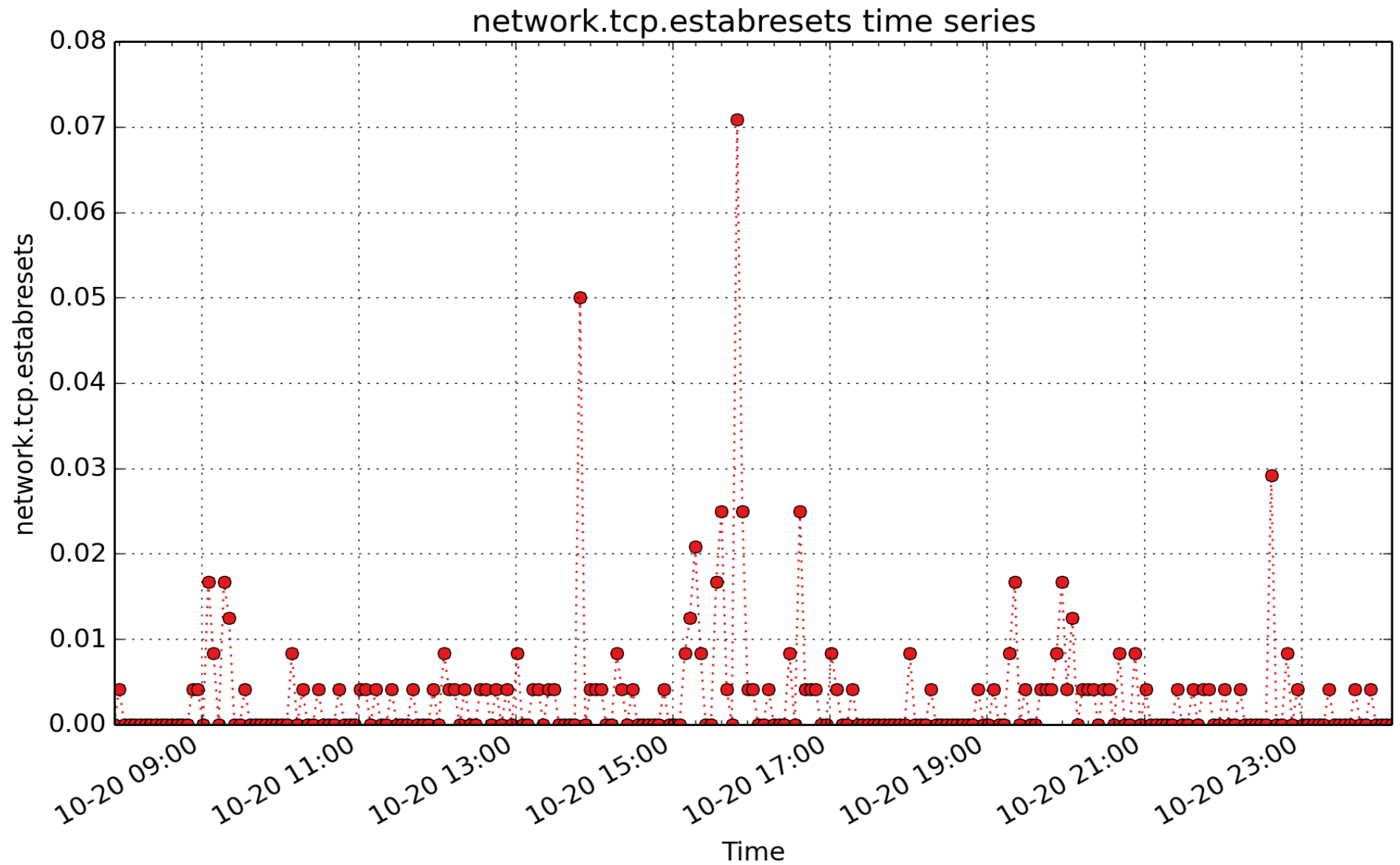
network.tcp.dsackoldsent: Number of DSACKs sent for old packets (count - U64) - *rate converted*



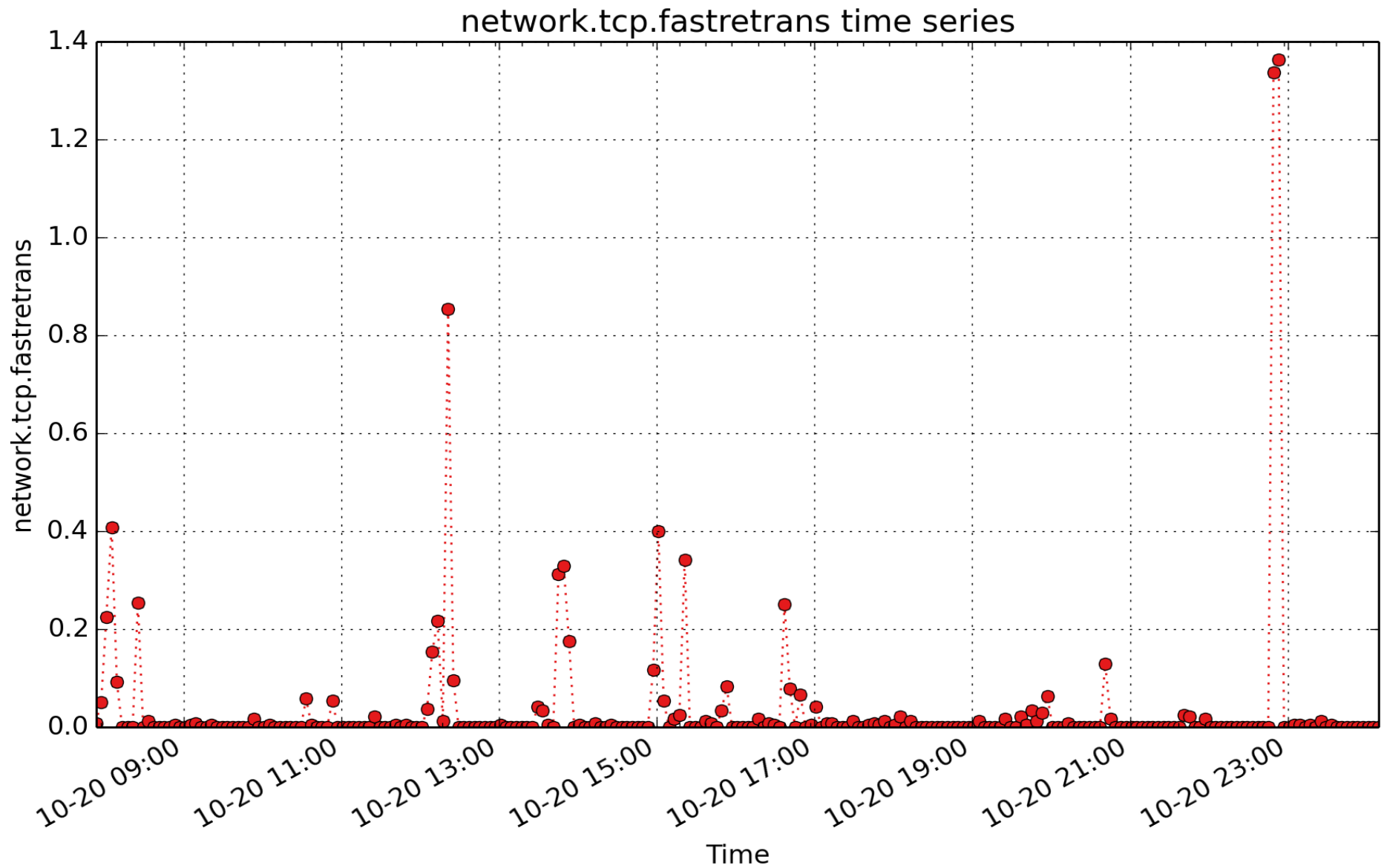
network.tcp.dsackrecv: Number of DSACKs received (count - U64) - *rate converted*



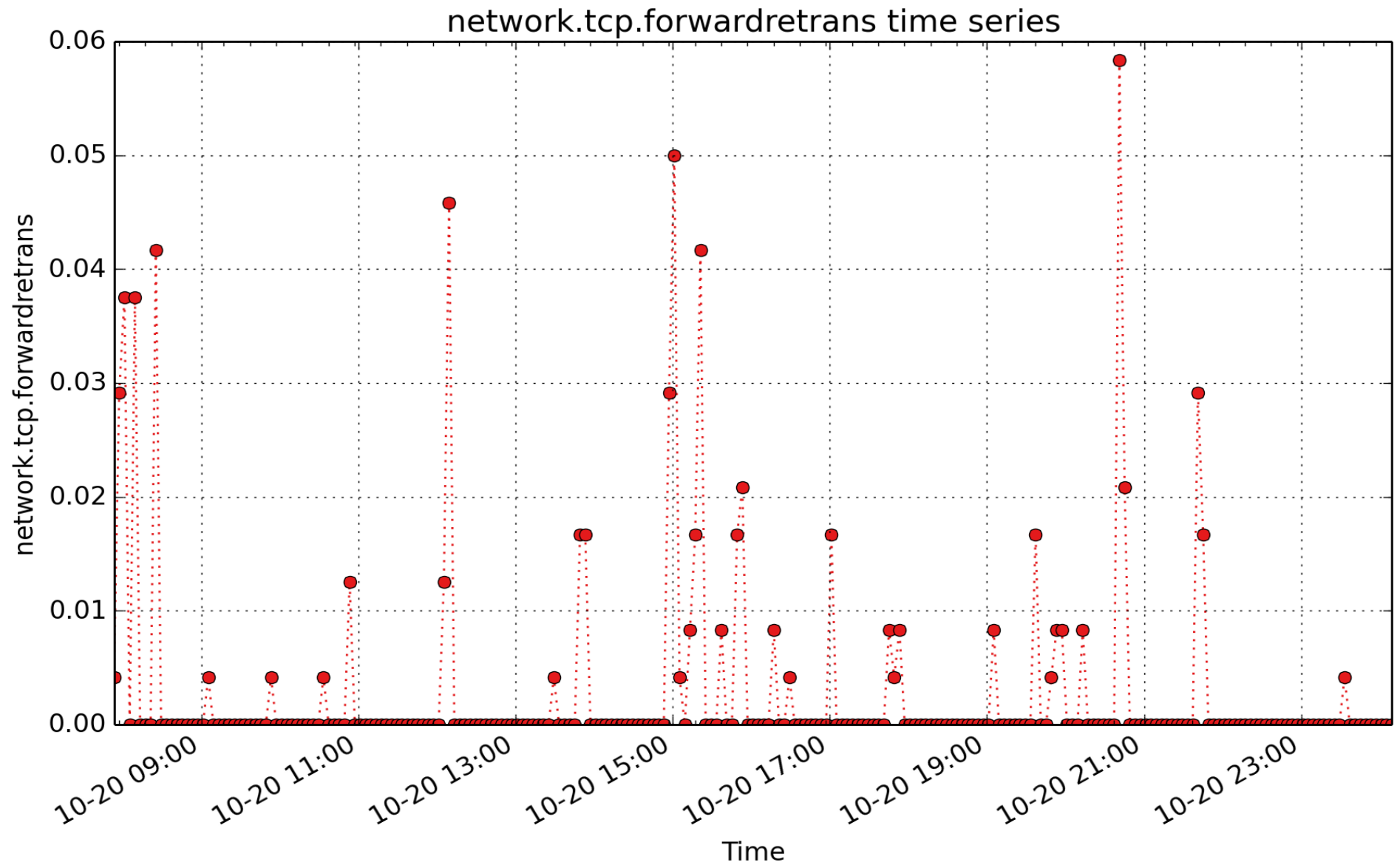
network.tcp.dsackundo: Number of congestion windows recovered without slow start using DSACK (count - U64) - *rate converted*



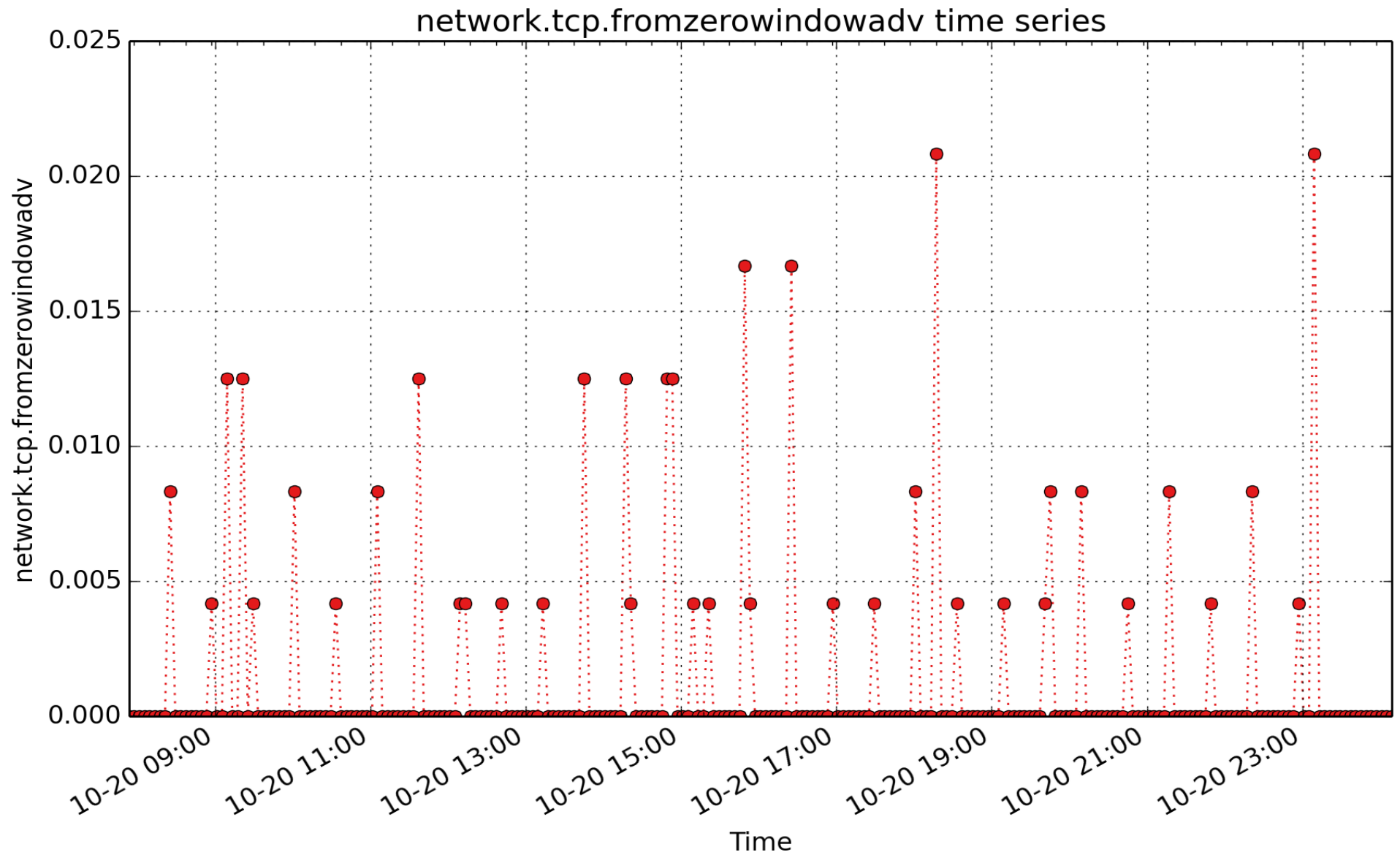
network.tcp.estabresets: count of tcp estabresets (count - U64) - *rate converted*



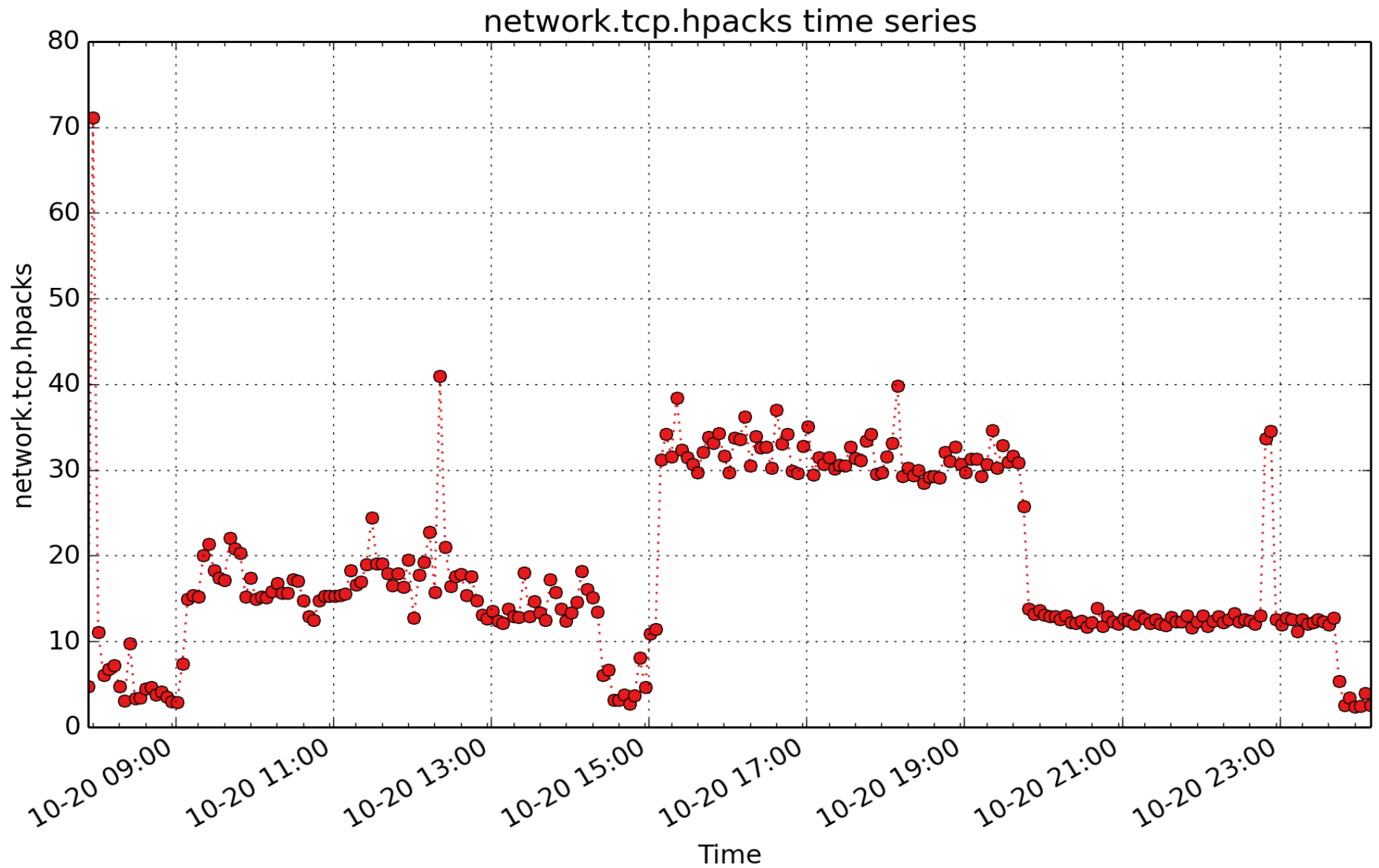
network.tcp.fastretrans: Number of fast retransmits (count - U64) - *rate converted*



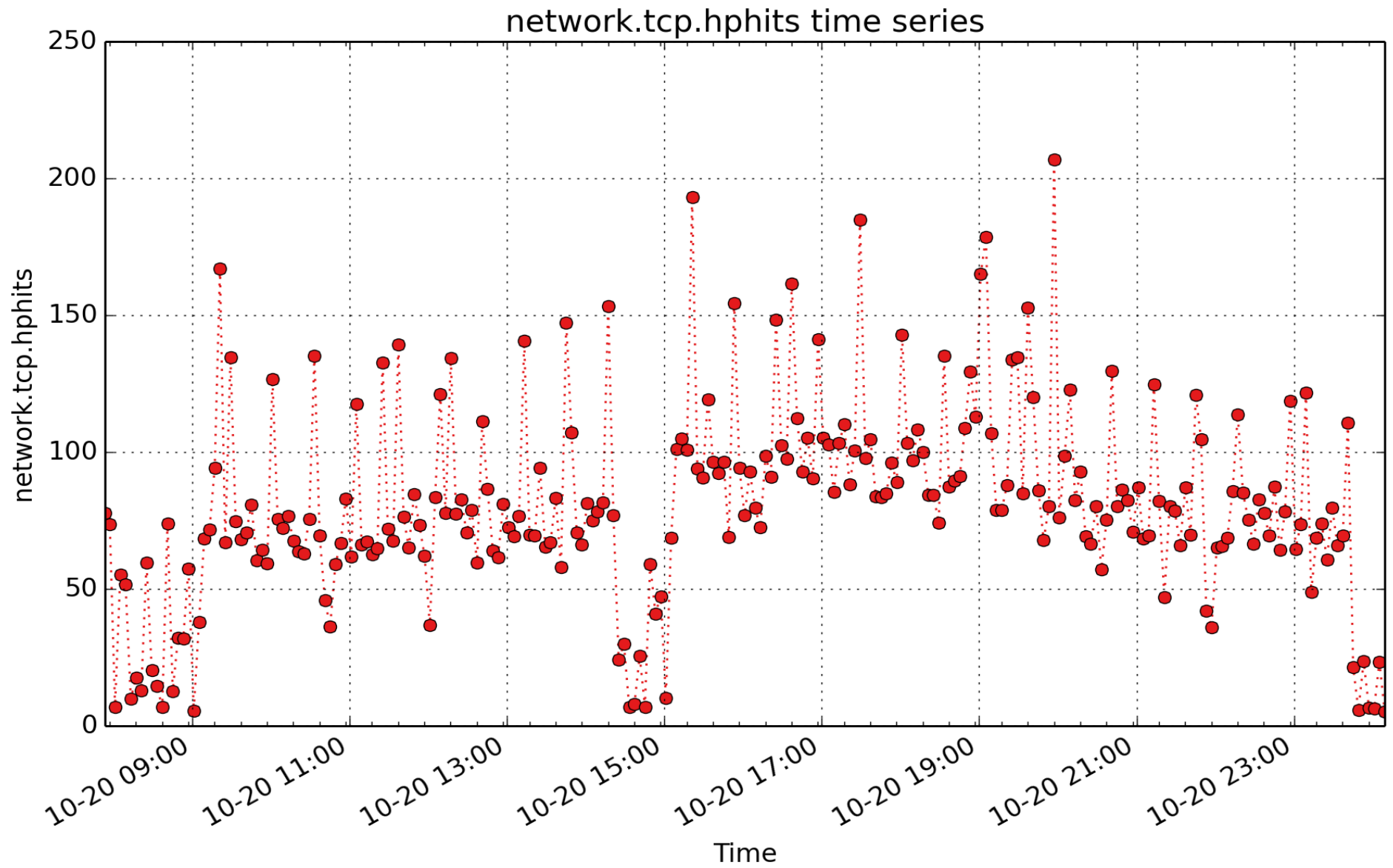
network.tcp.forwardretrans: Number of forward retransmits (count - U64) - *rate converted*



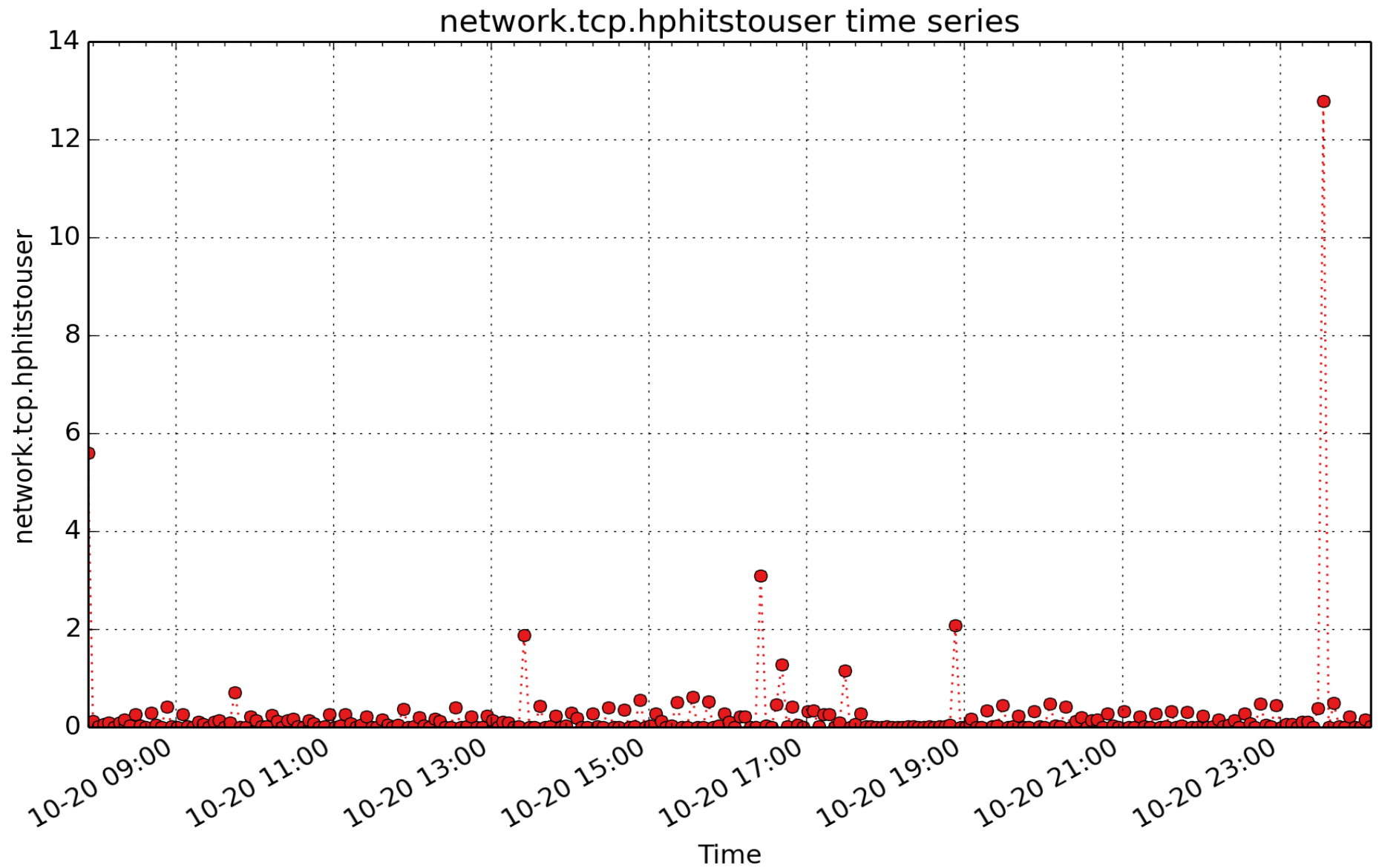
network.tcp.fromzerowindowadv: Number of times window went from zero to non-zero (count - U64) - *rate converted*



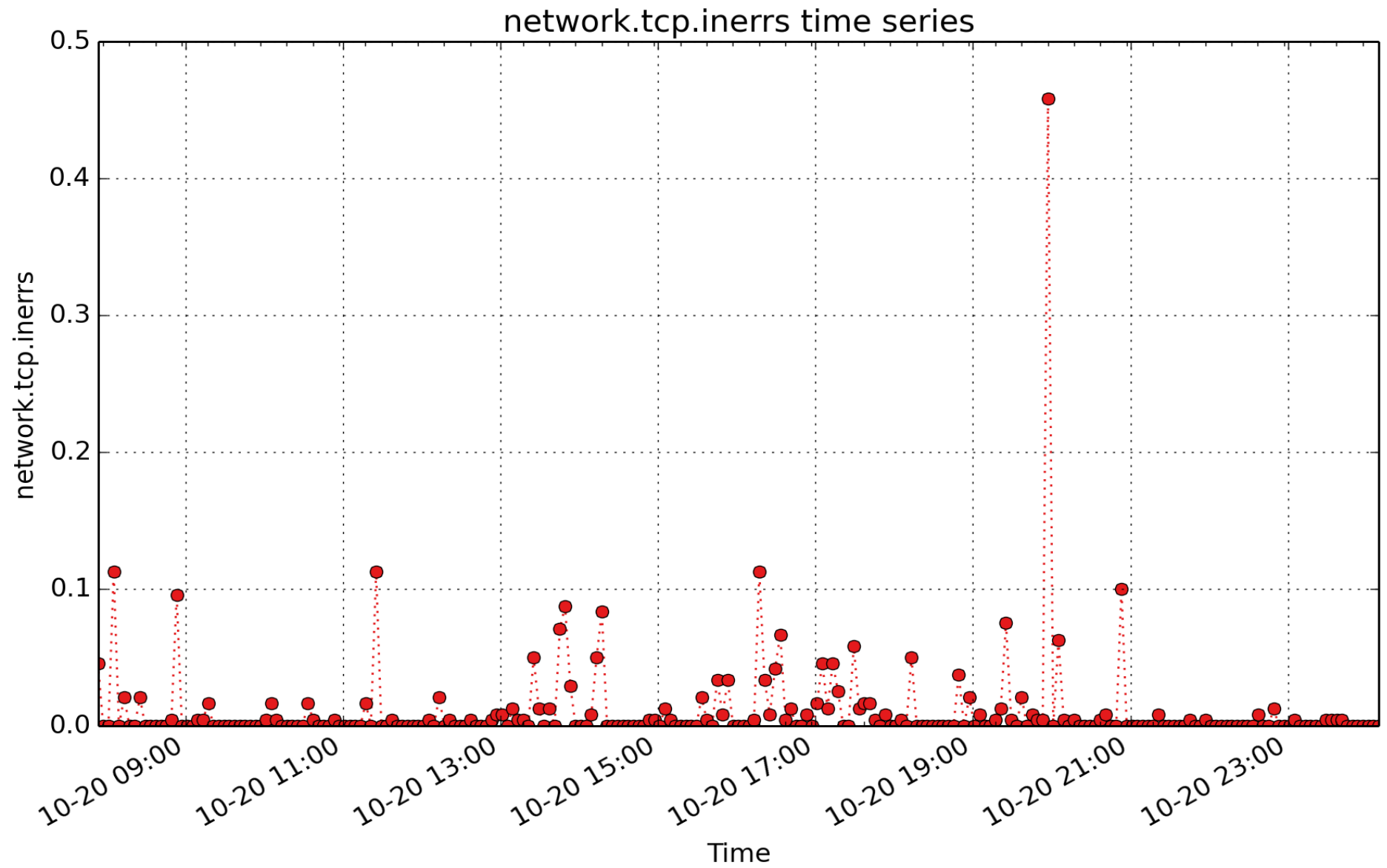
network.tcp.hpacks: Number of predicted acknowledgments (count - U64) - *rate converted*



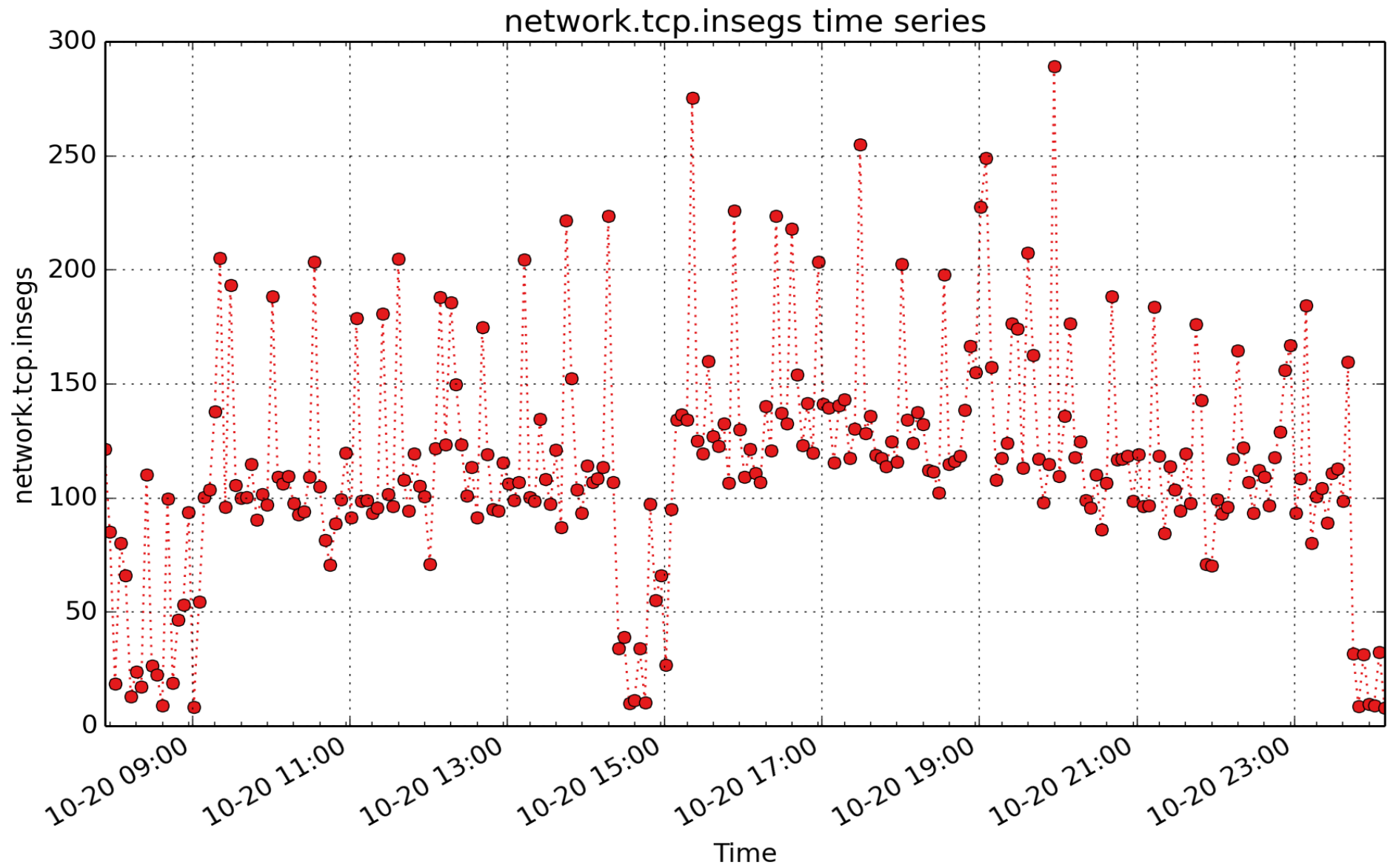
network.tcp.hphits: Number of packet headers predicted (count - U64) - *rate converted*



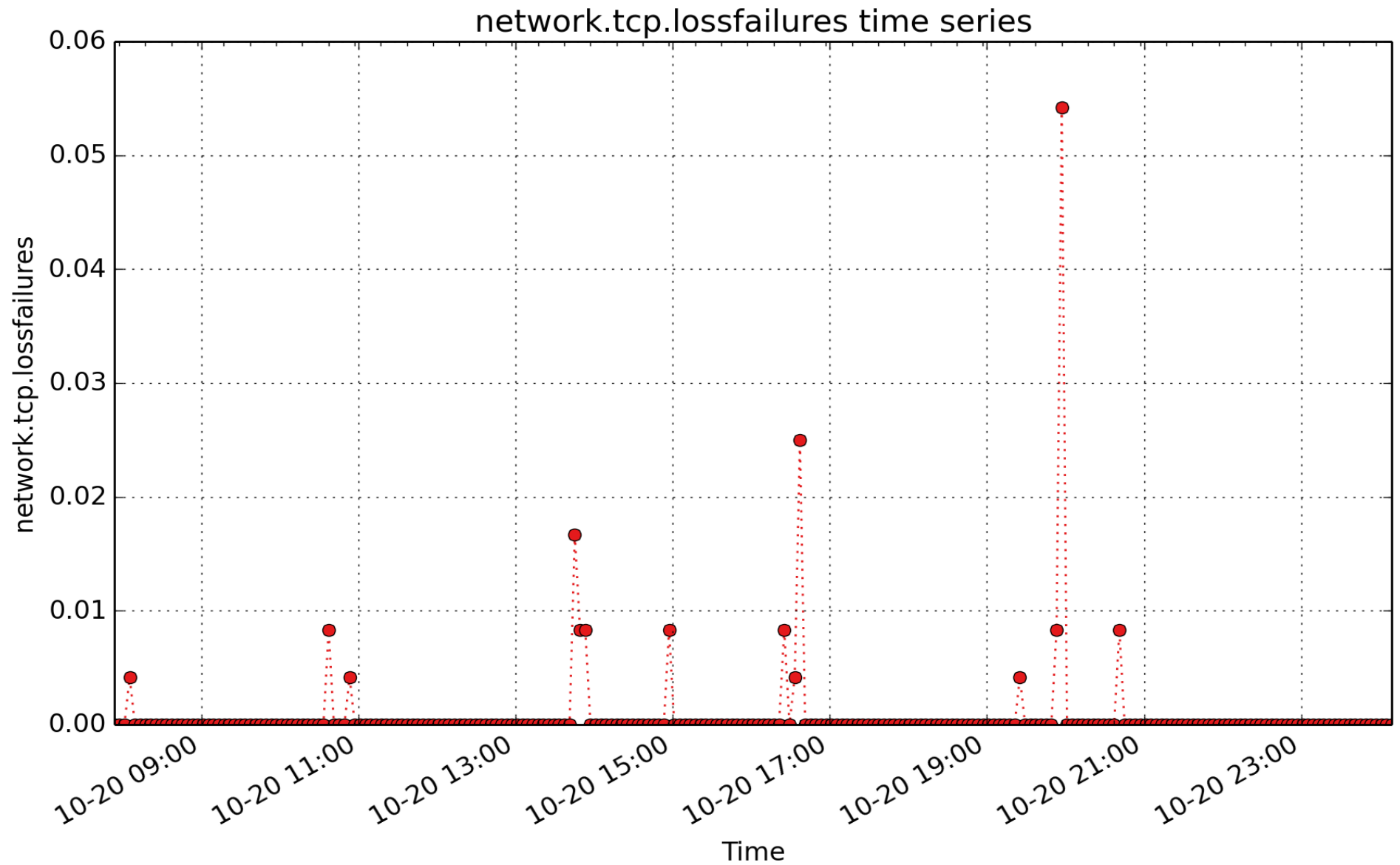
network.tcp.hphitstouser: Number of packets header predicted and directly queued to user (count - U64) -
rate converted



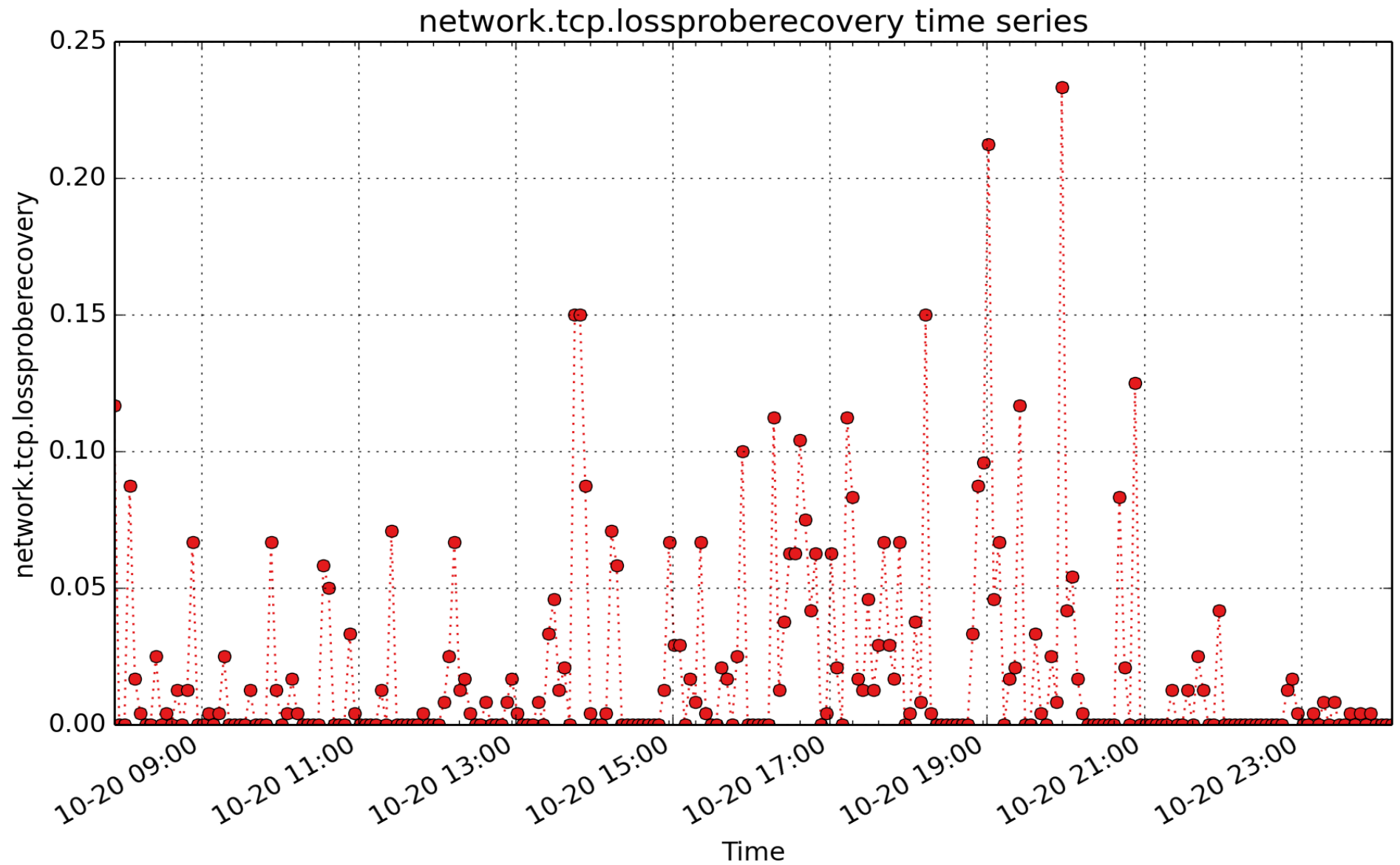
network.tcp.inerrs: count of tcp inerrs (count - U64) - *rate converted*



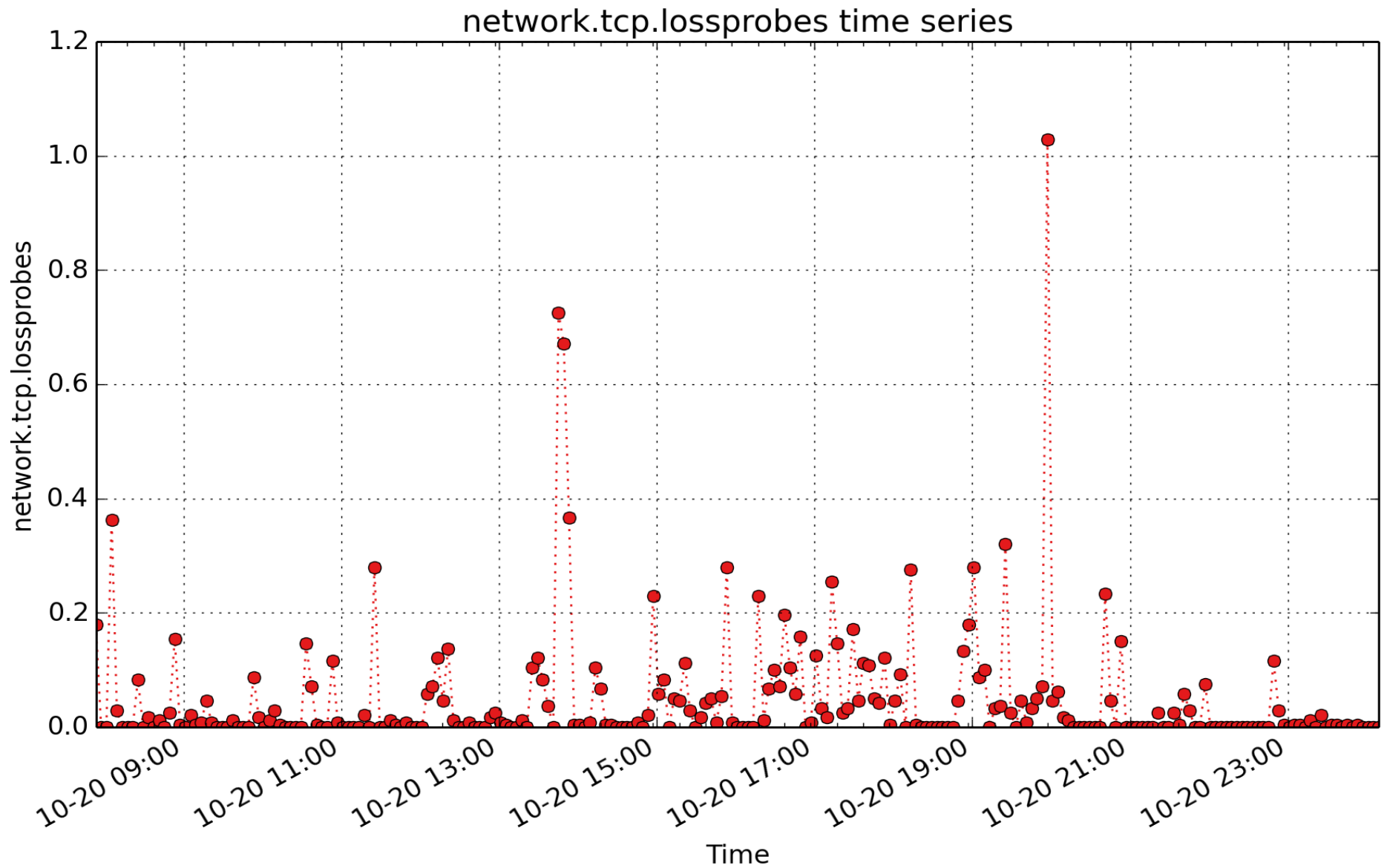
network.tcp.insegs: count of tcp insegs (count - U64) - *rate converted*



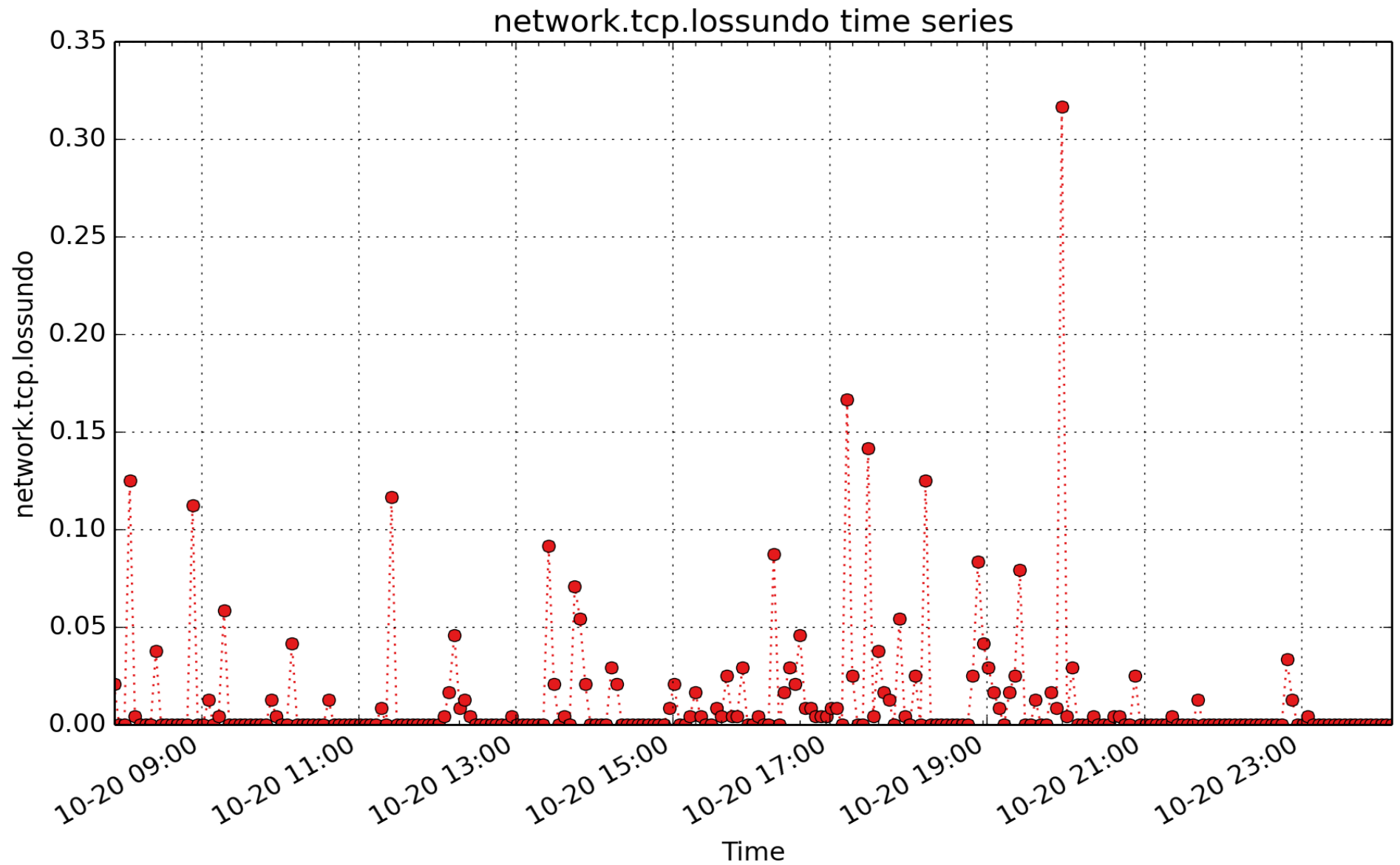
network.tcp.lossfailures: Number of timeouts in loss state (count - U64) - *rate converted*



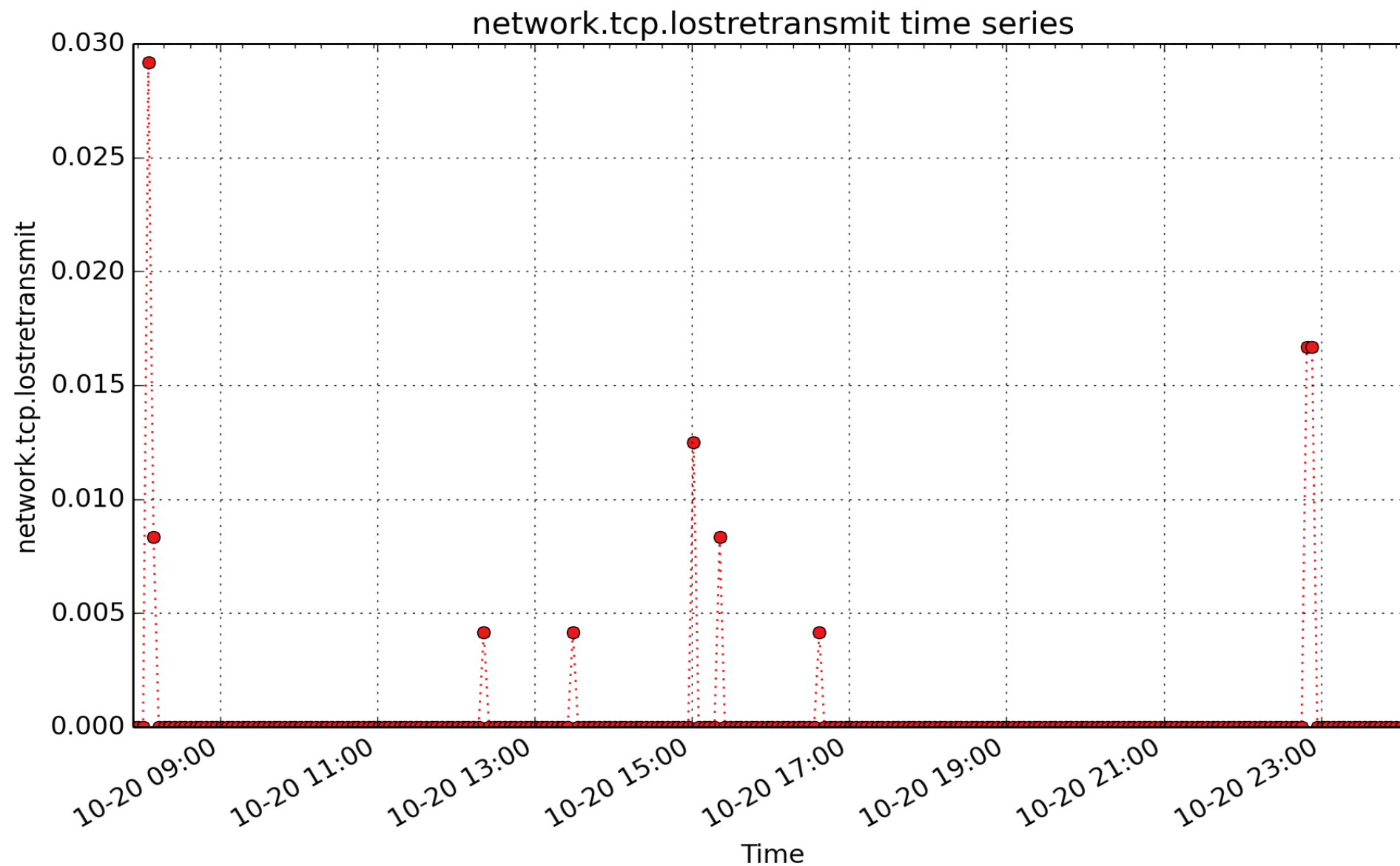
network.tcp.lossproberecovery: Number of TCP loss probe recoveries (count - U64) - *rate converted*



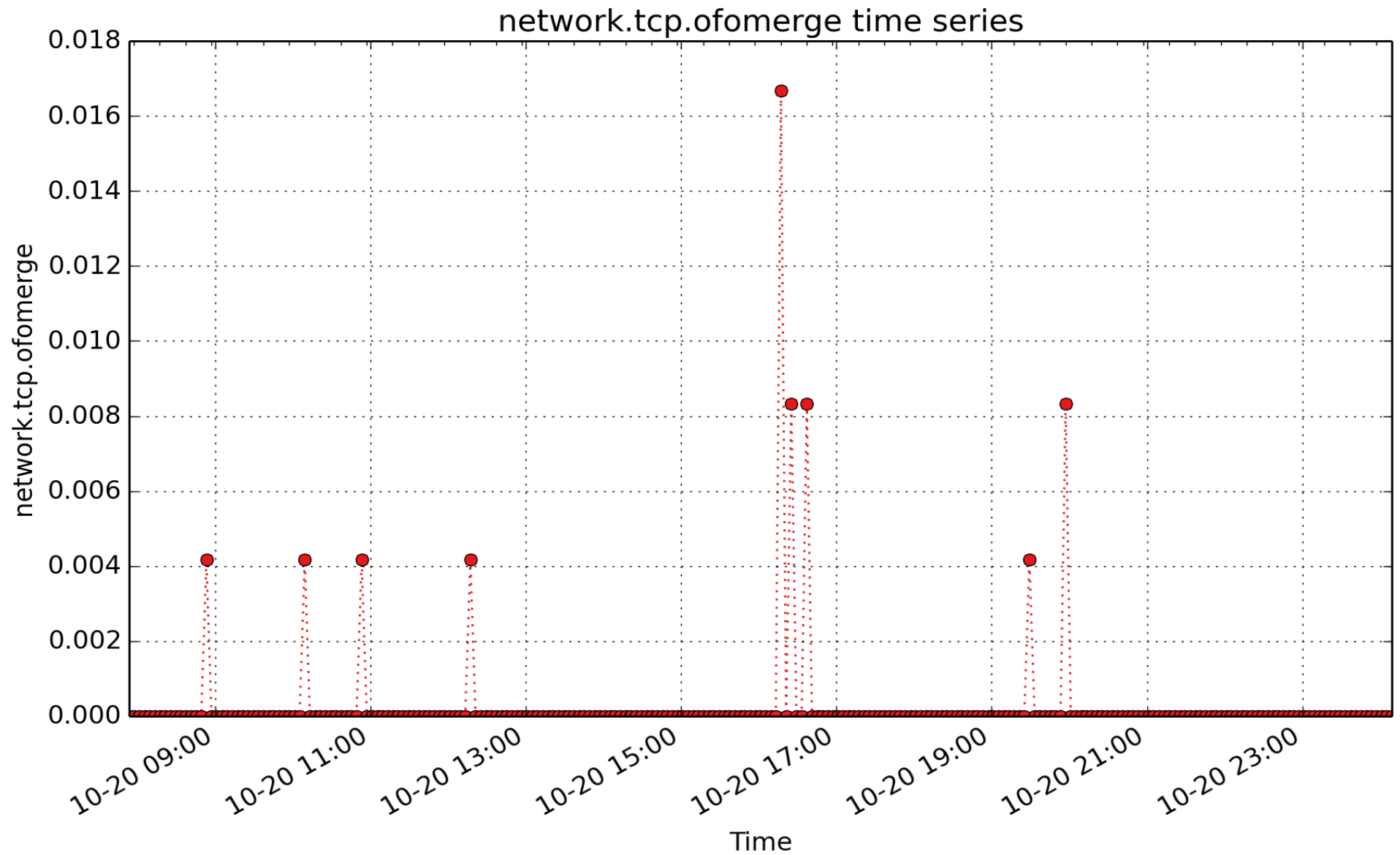
network.tcp.lossprobes: Number of sent TCP loss probes (count - U64) - *rate converted*



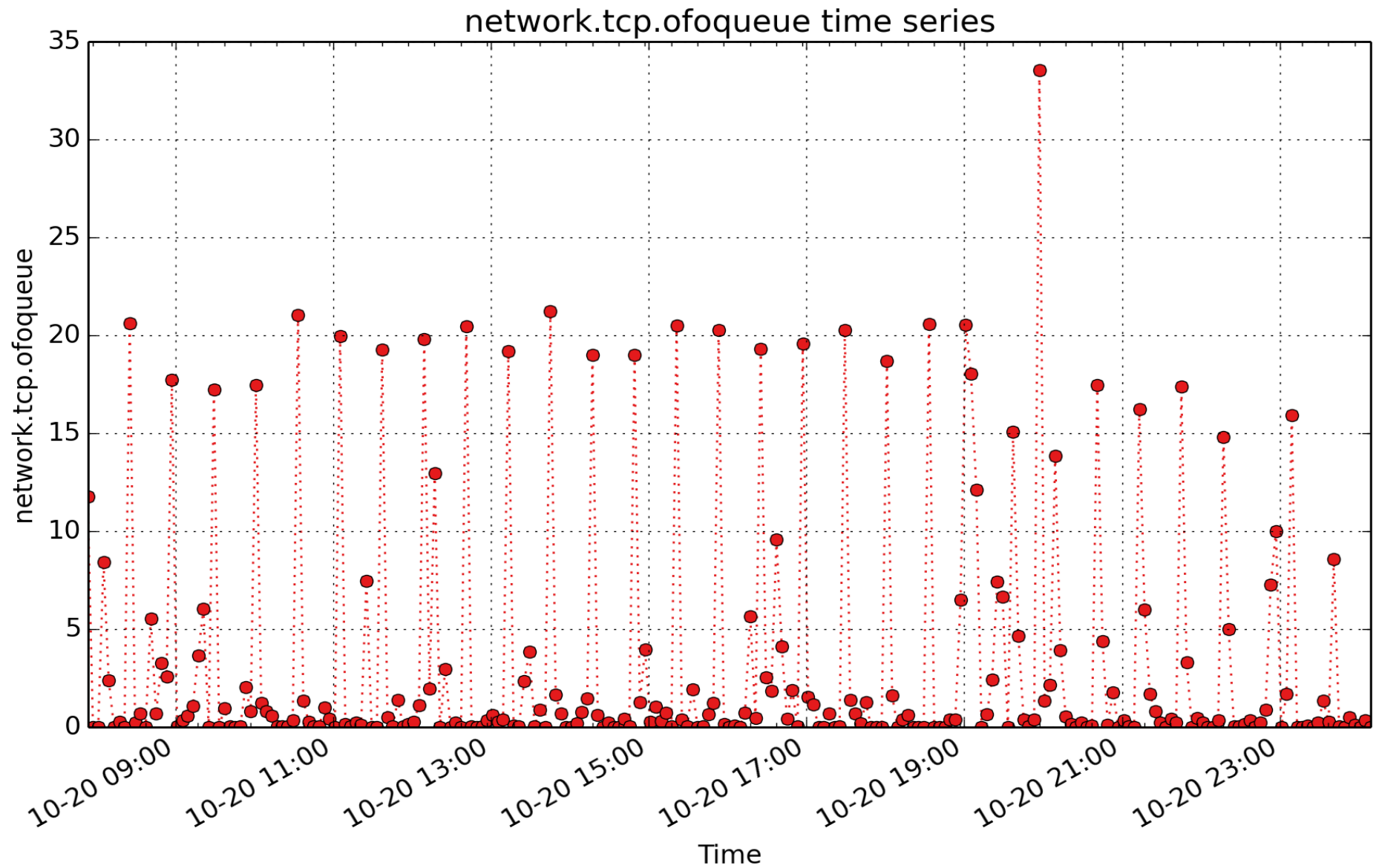
network.tcp.lossundo: Number of congestion windows recovered without slow start after partial ack (count - U64) - *rate converted*



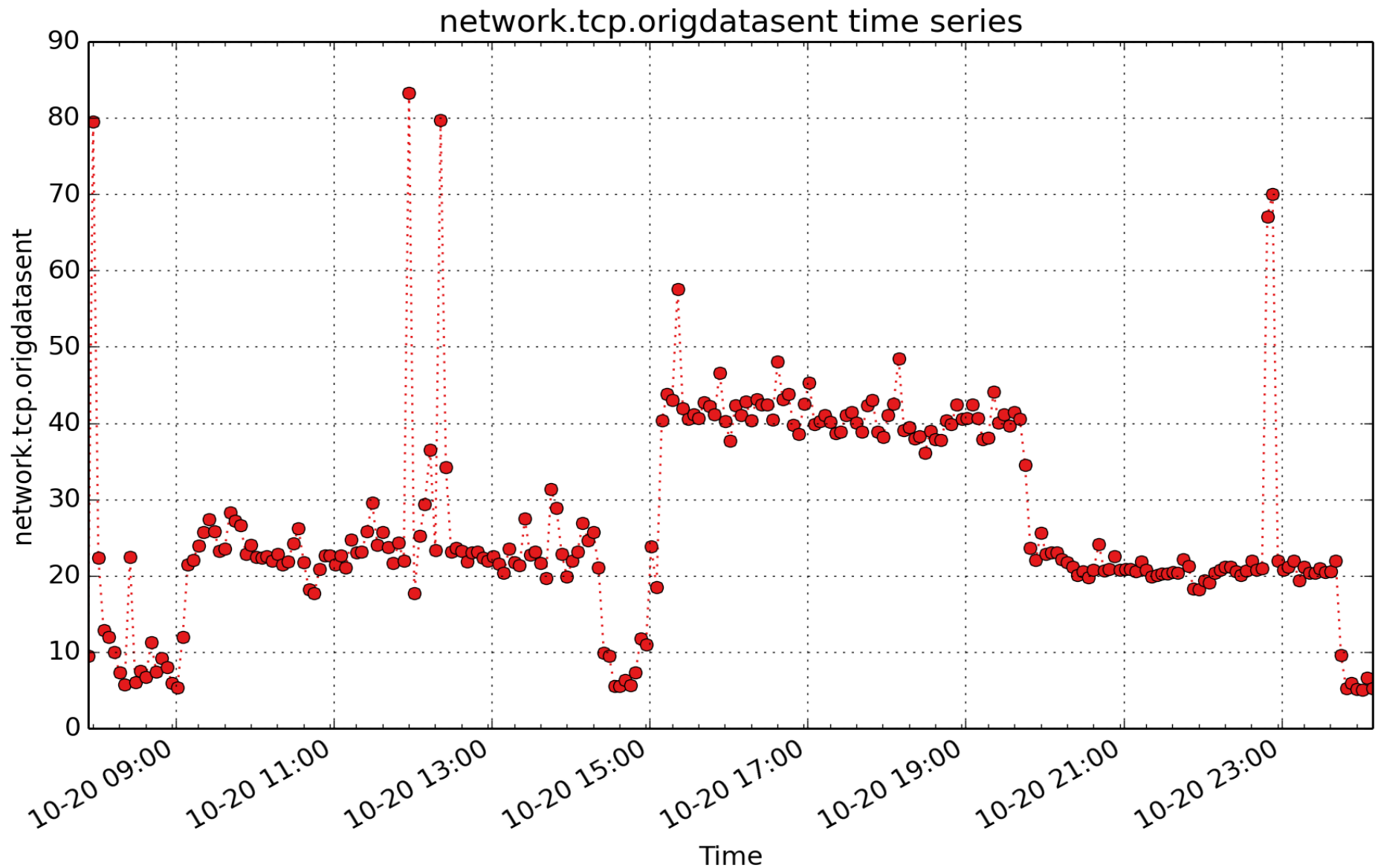
network.tcp.lostretransmit: Number of retransmits lost (count - U64) - *rate converted*



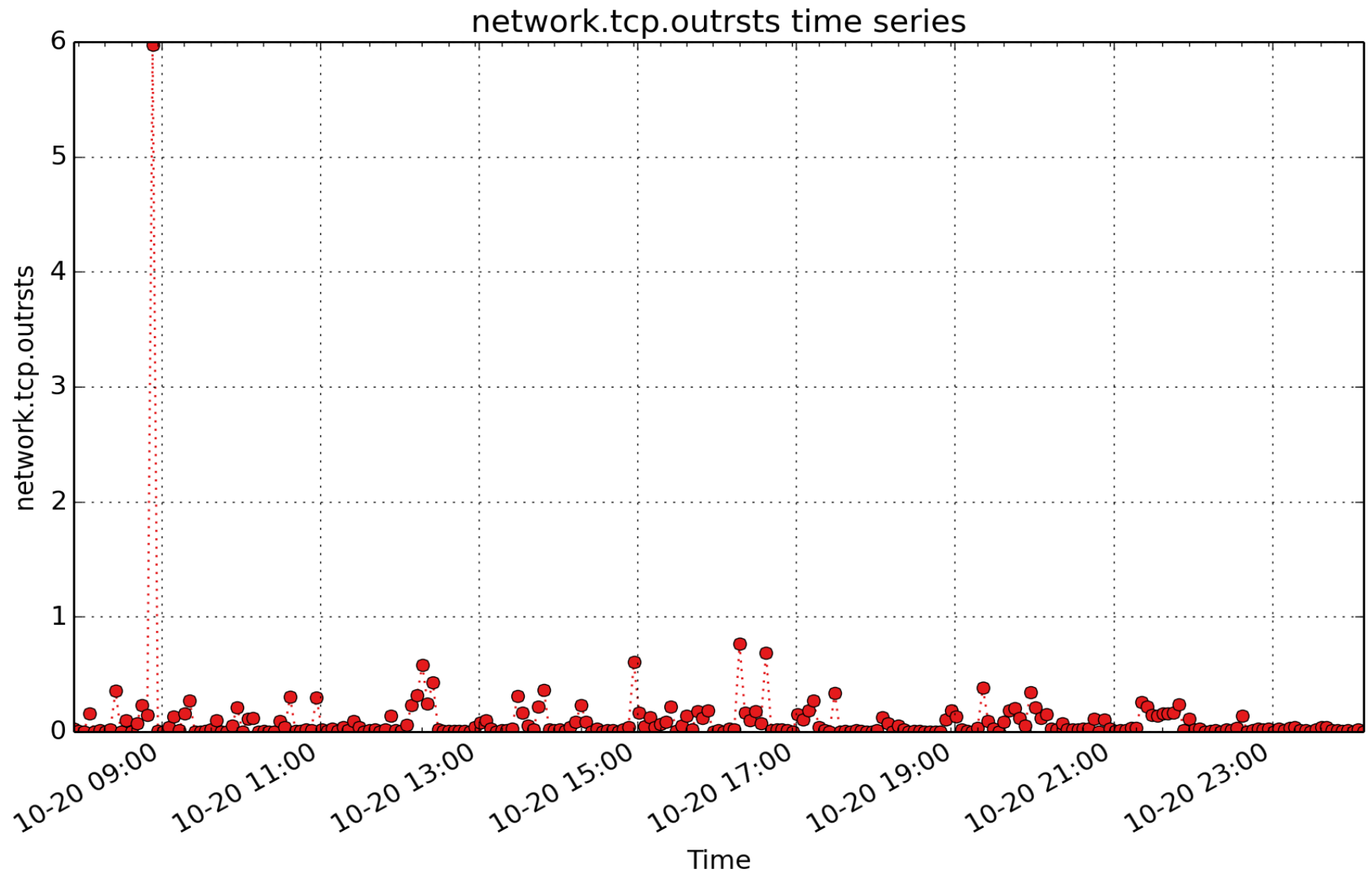
network.tcp.ofomerge: Number of packets in OFO that were merged with other packets (count - U64) - *rate converted*



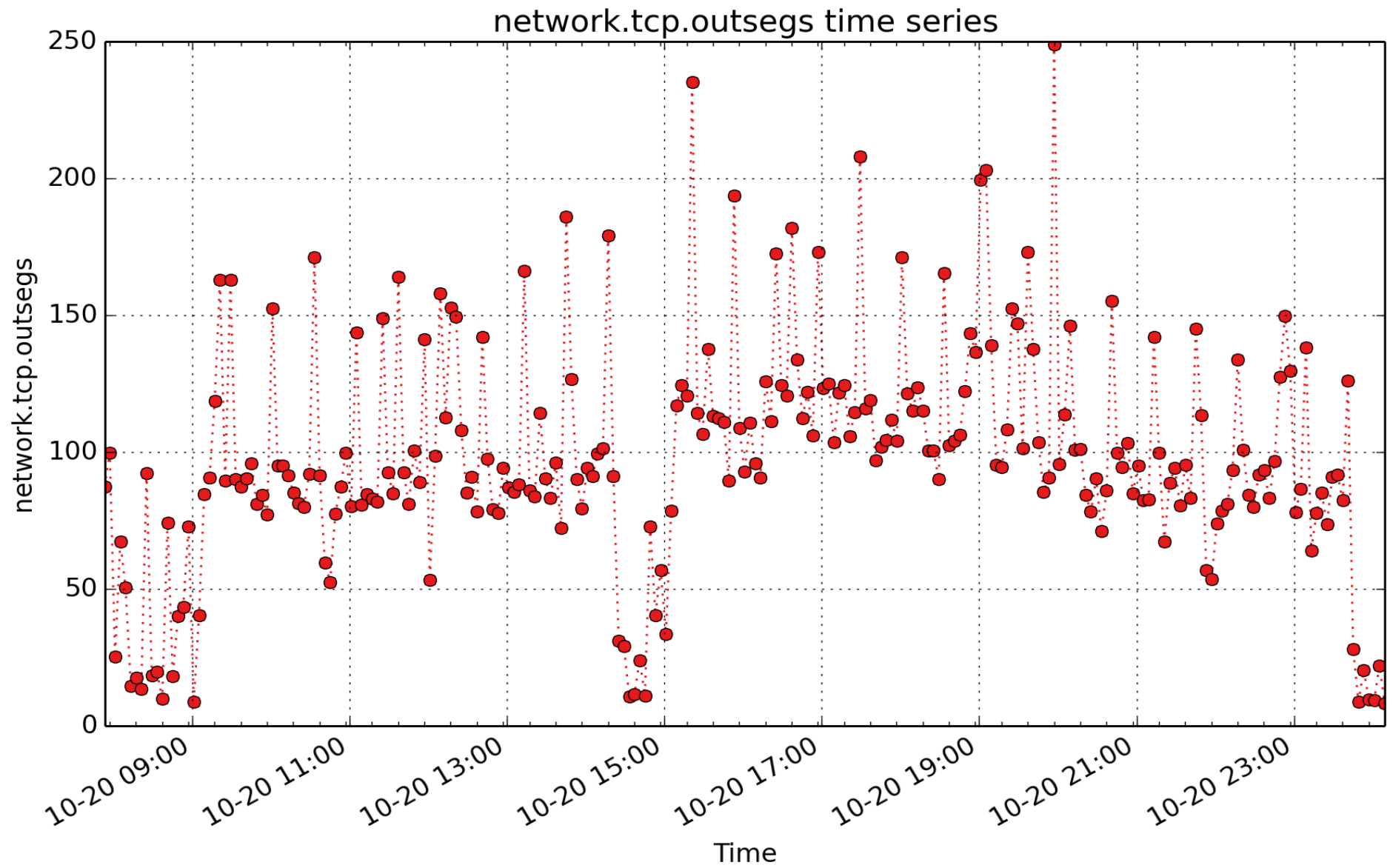
network.tcp.foqueue: Number of packets queued in OFO queue (count - U64) - *rate converted*



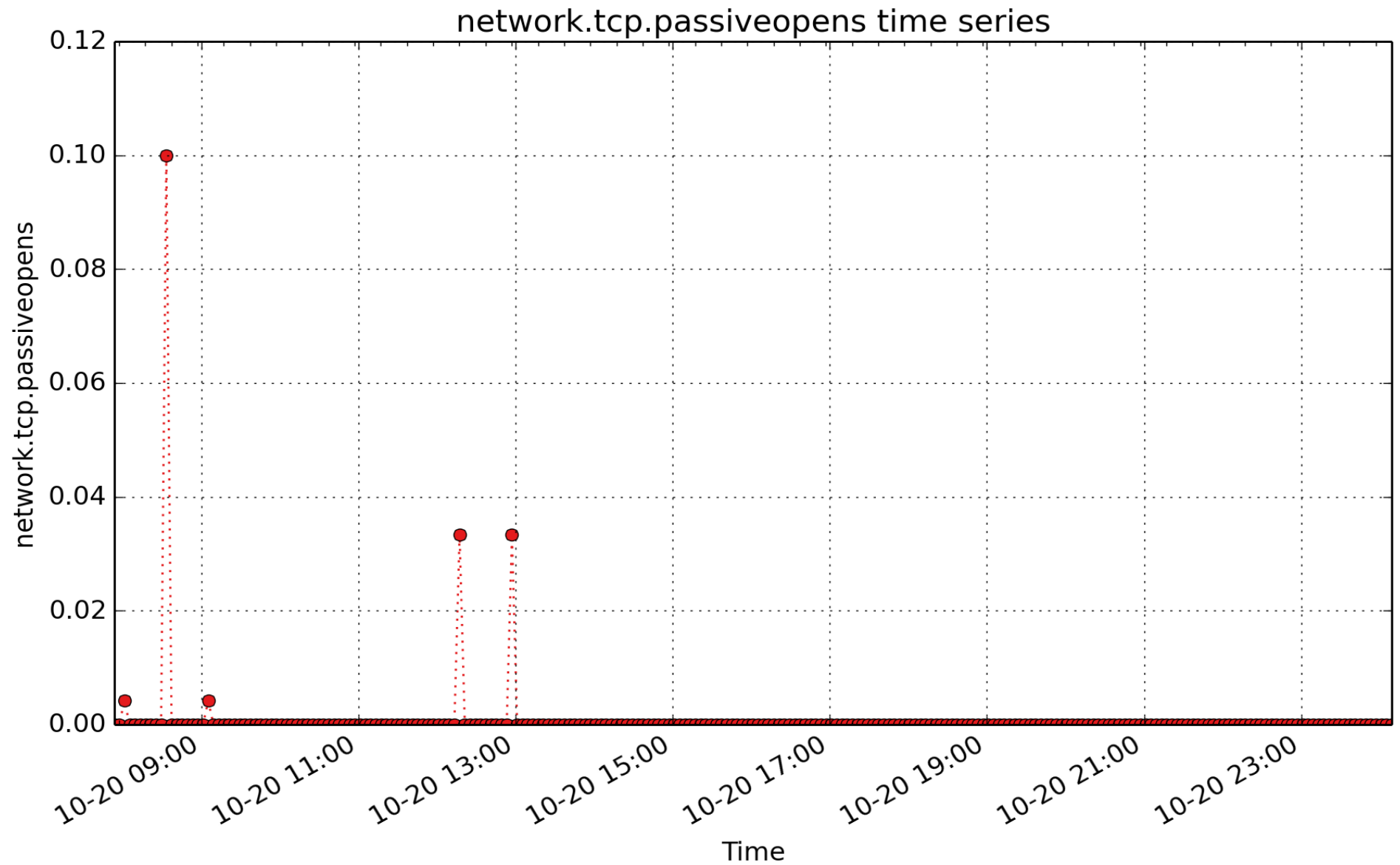
network.tcp.origdatasent: Excluding retransmission but including data-in-SYN). This counter is different from TcpOutSegs because TcpOutSegs also tracks pure ACKs. TCPOrigDataSent is more useful to track the TCP retransmission rate. (count - U64) - *rate converted*



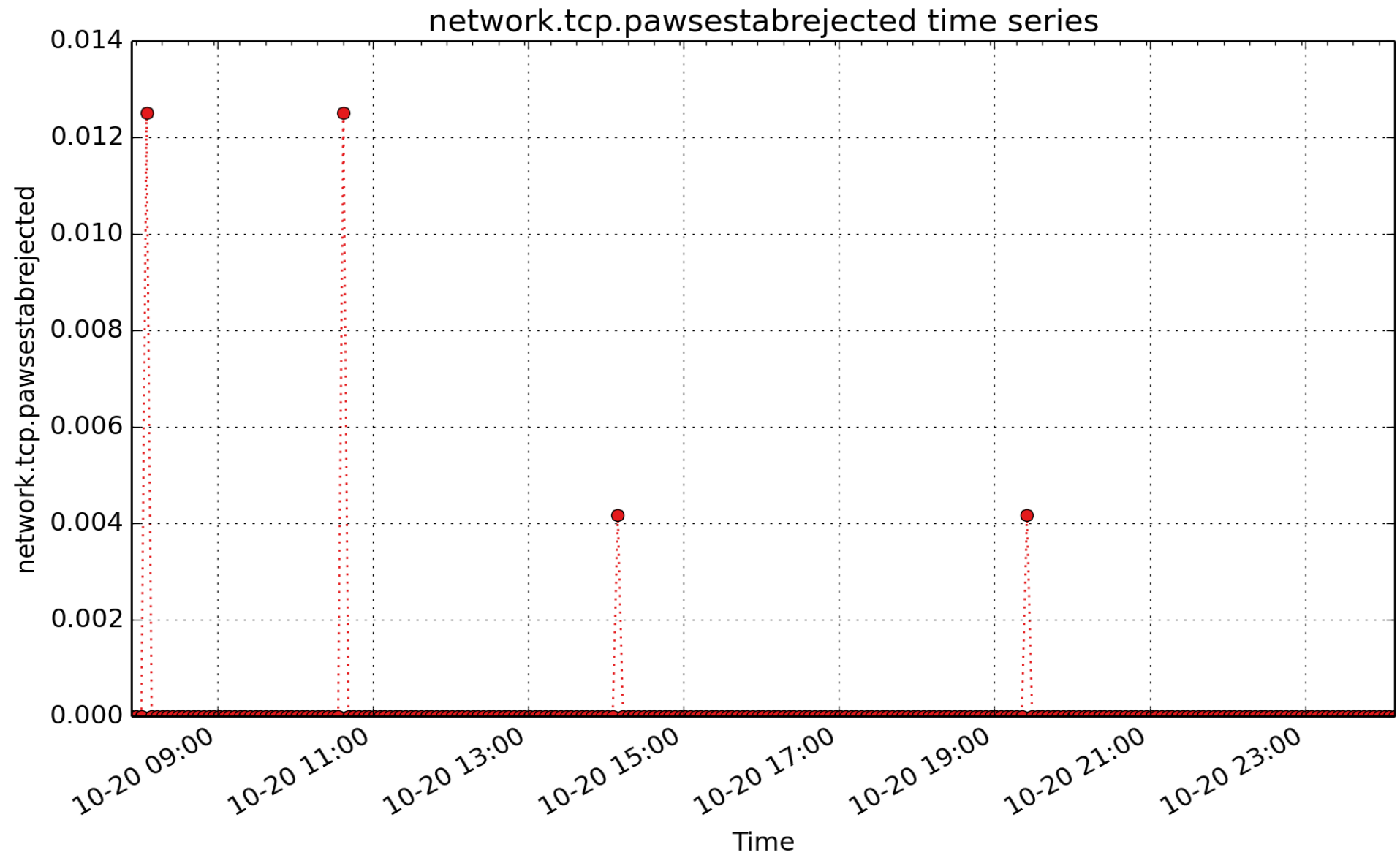
network.tcp.outrsts: count of tcp outrsts (count - U64) - *rate converted*



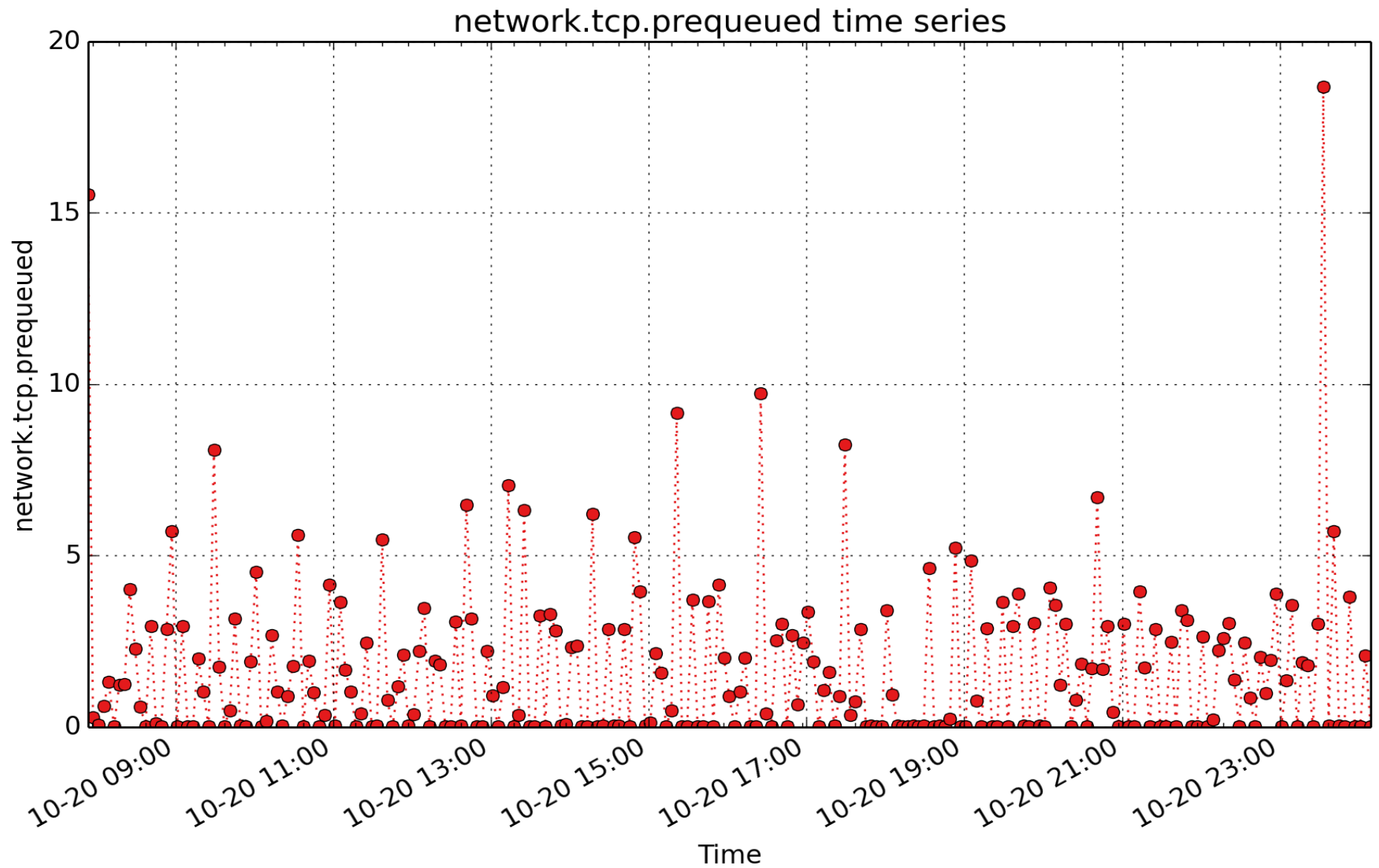
network.tcp.outsegs: count of tcp outsegs (count - U64) - *rate converted*



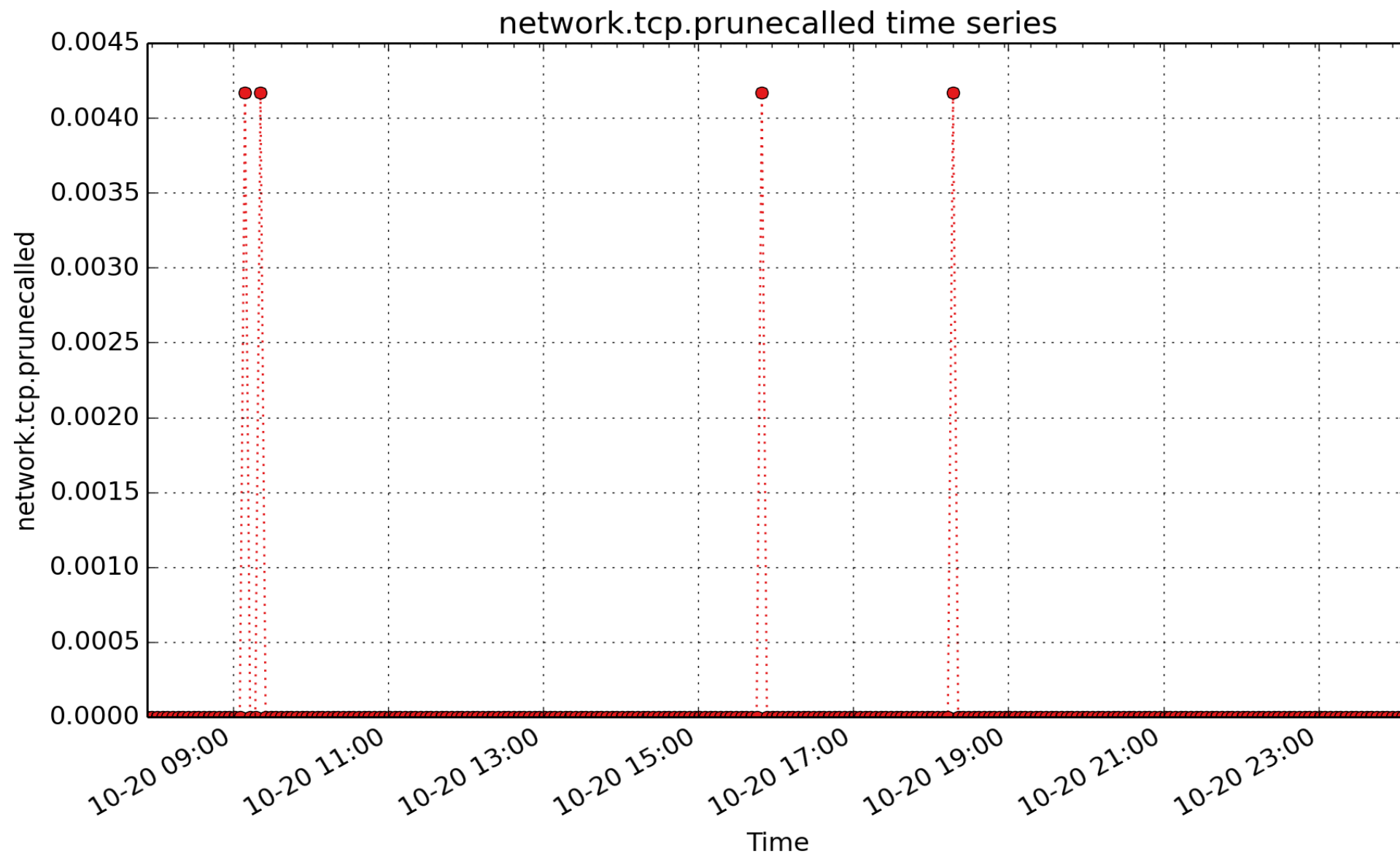
network.tcp.passiveopens: count of tcp passiveopens (count - U64) - *rate converted*



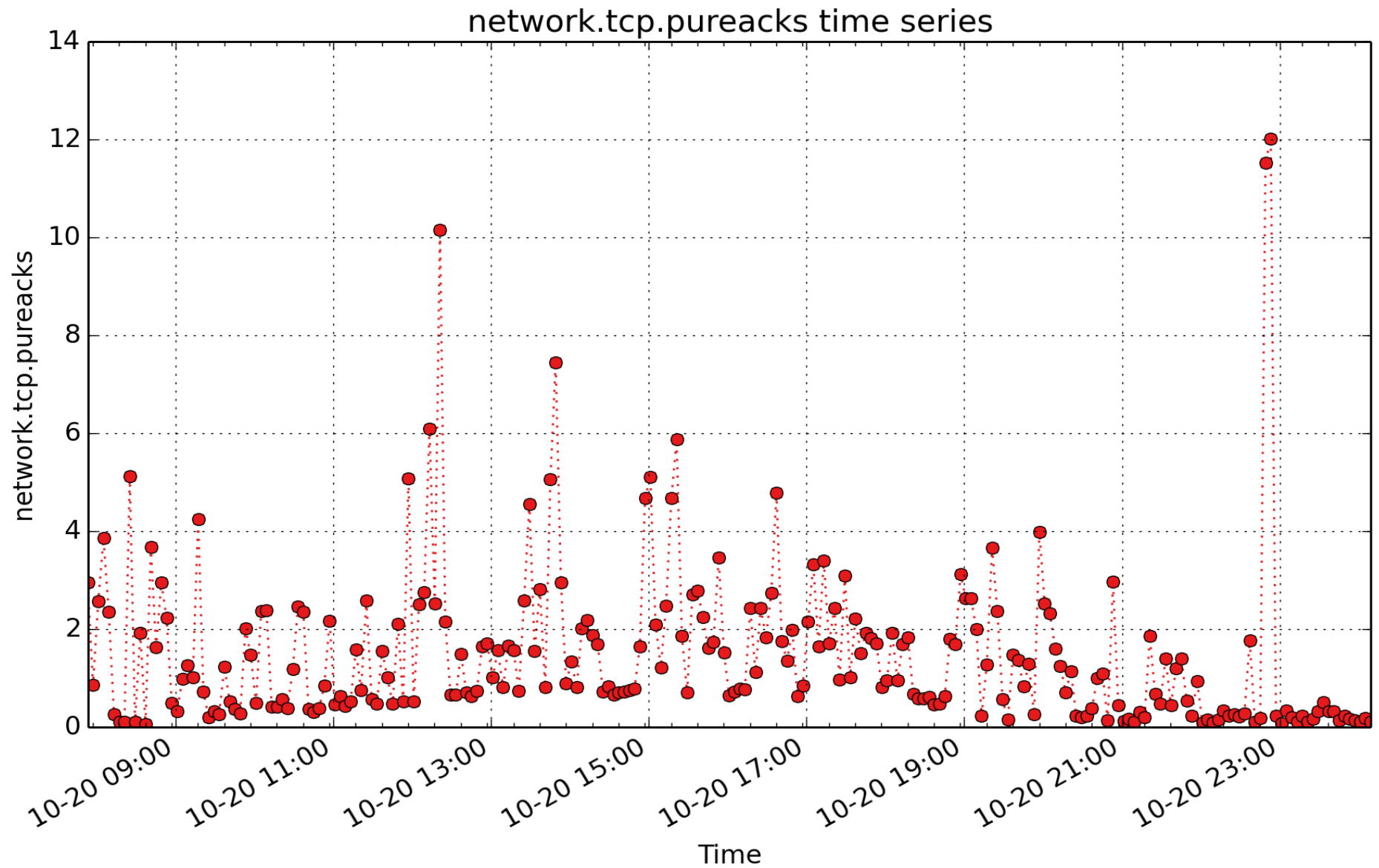
network.tcp.pawsestabrejected: Number of packets rejects in established connections because of timestamp (count - U64) - *rate converted*



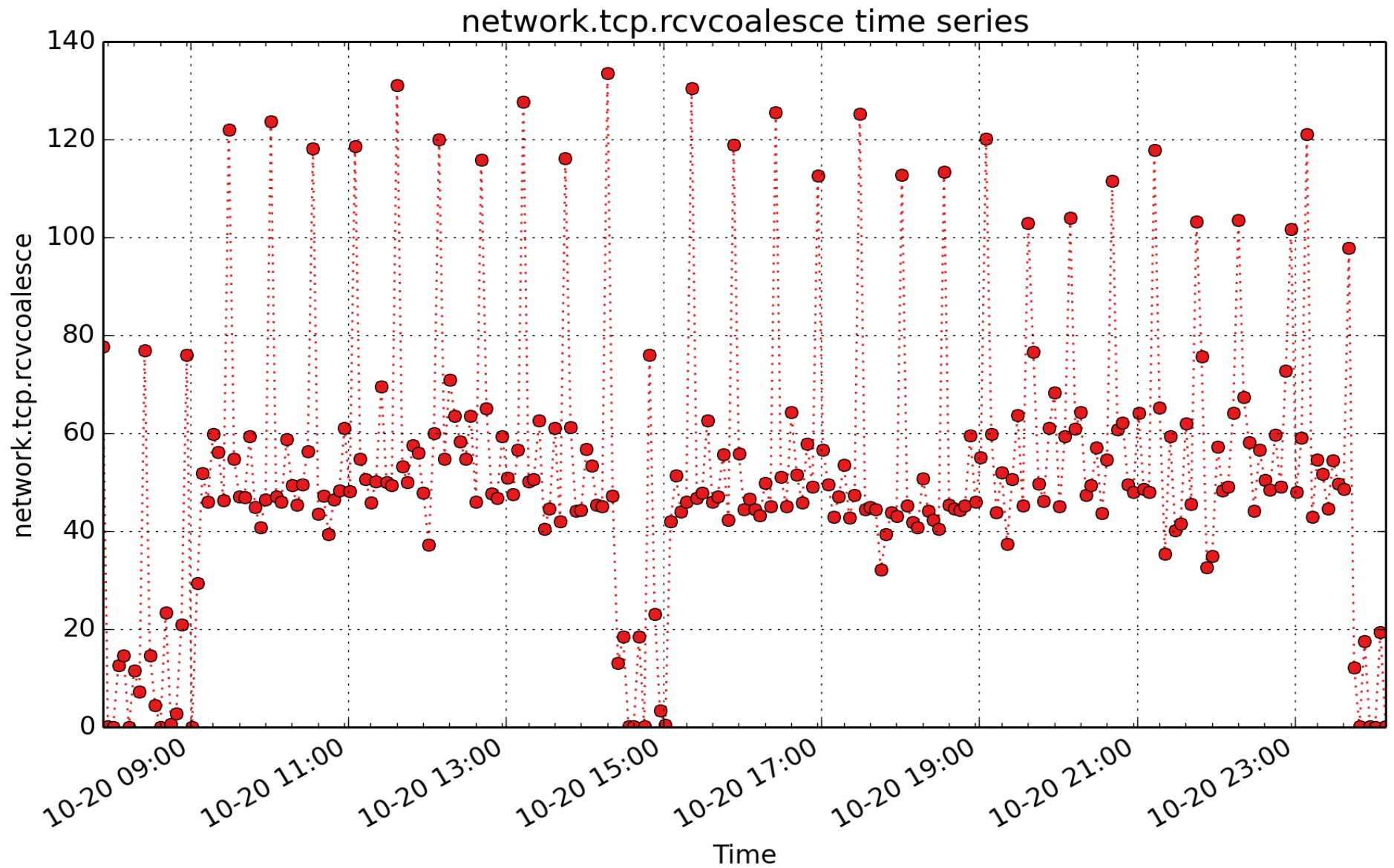
network.tcp.prequeued: Number of packets directly queued to recvmmsg prequeue (count - U64) - *rate converted*



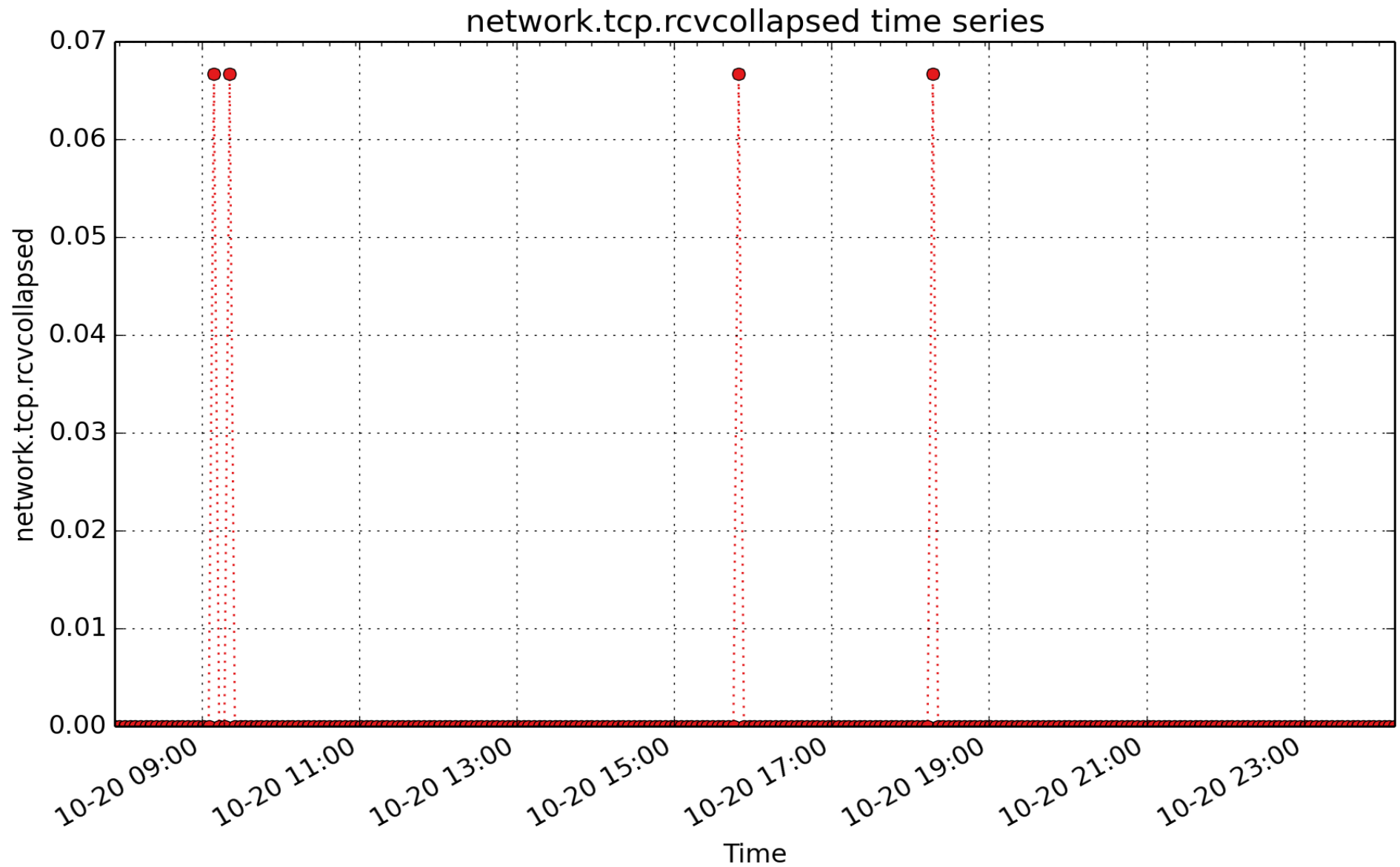
network.tcp.prunecalled: Number of packets pruned from receive queue because of socket buffer overrun (count - U64) - *rate converted*



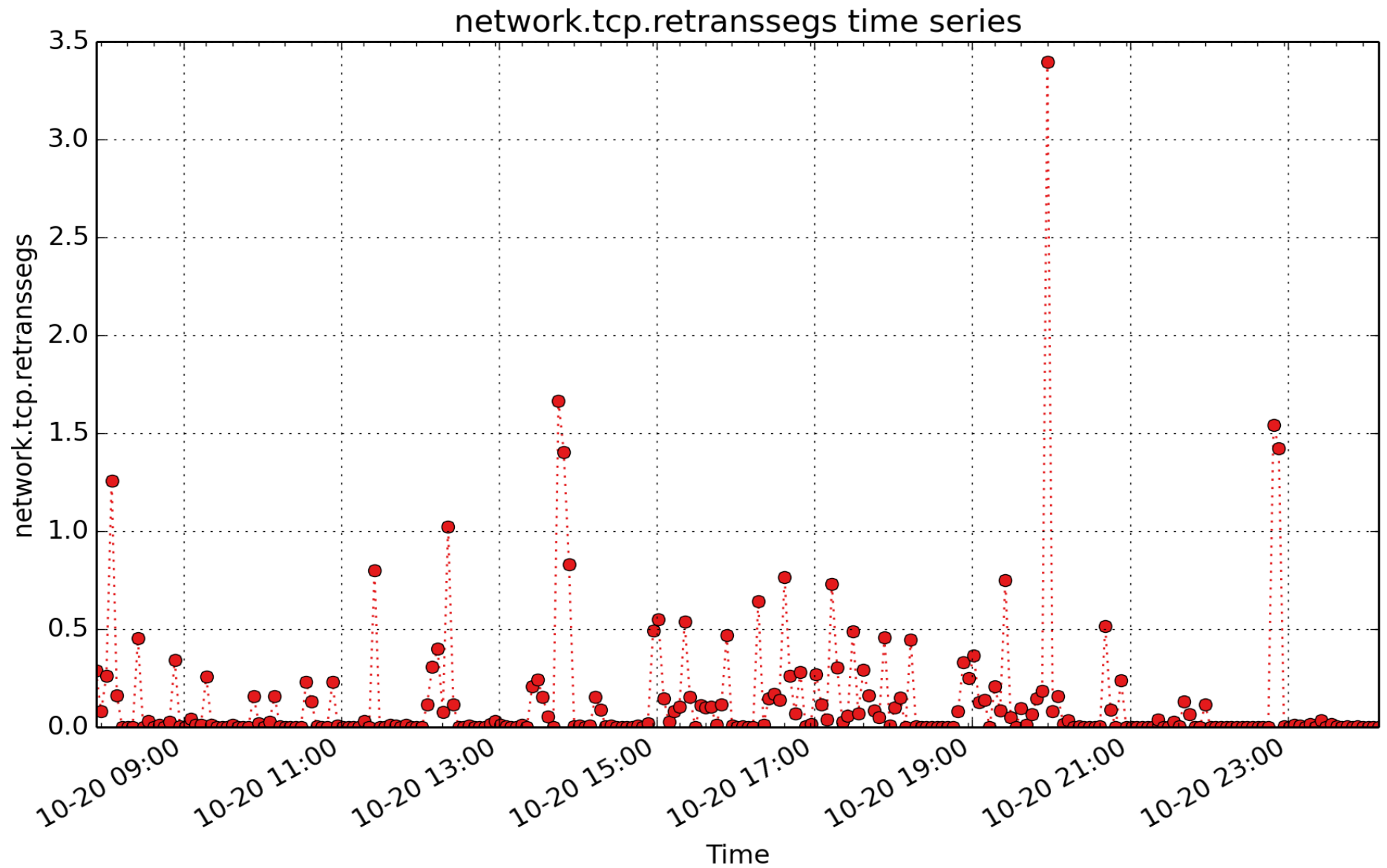
network.tcp.pureacks: Number of acknowledgments not containing data payload received (count - U64) -
rate converted



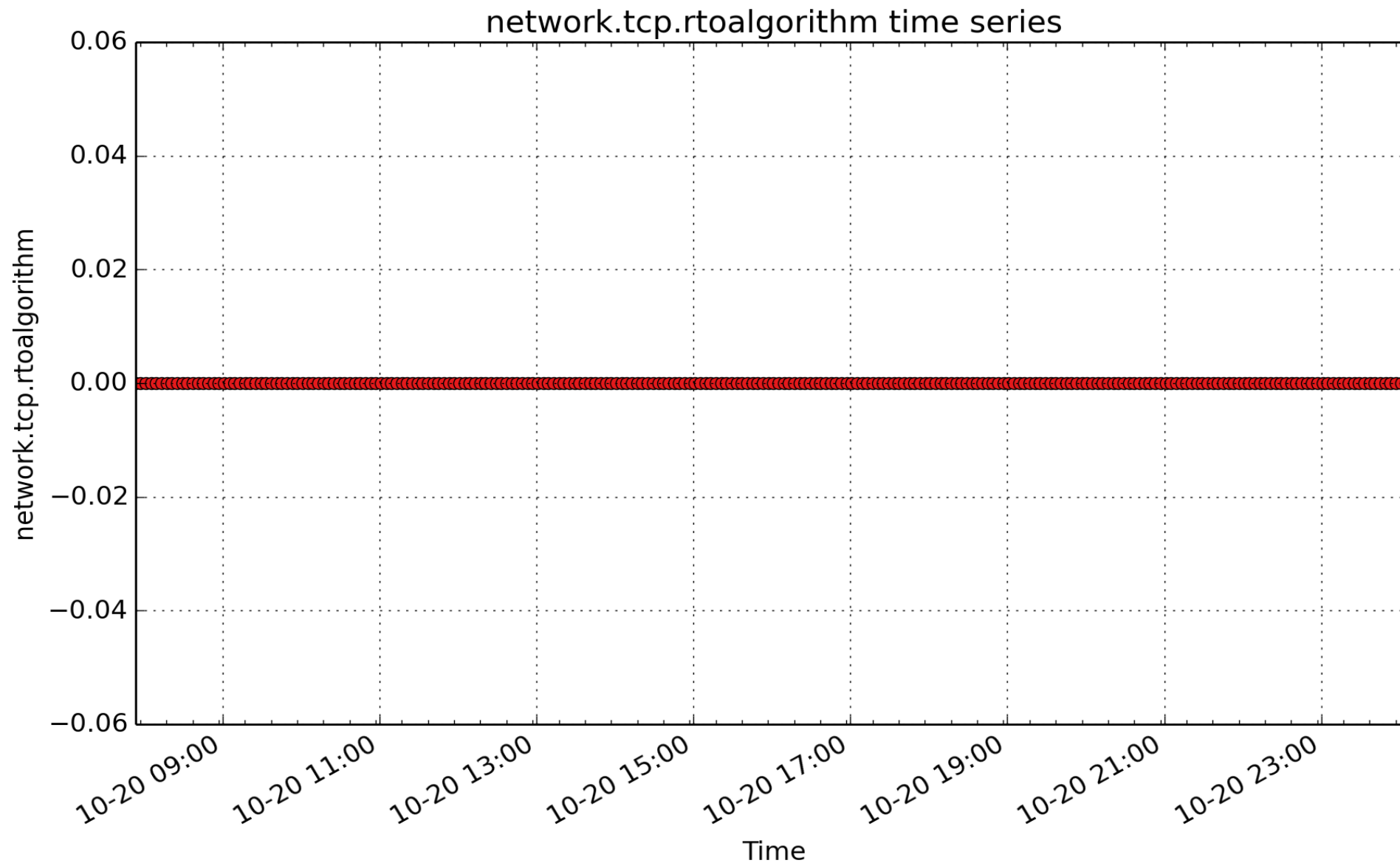
network.tcp.rcvcoalesce: Number of times tried to coalesce the receive queue (count - U64) - *rate converted*



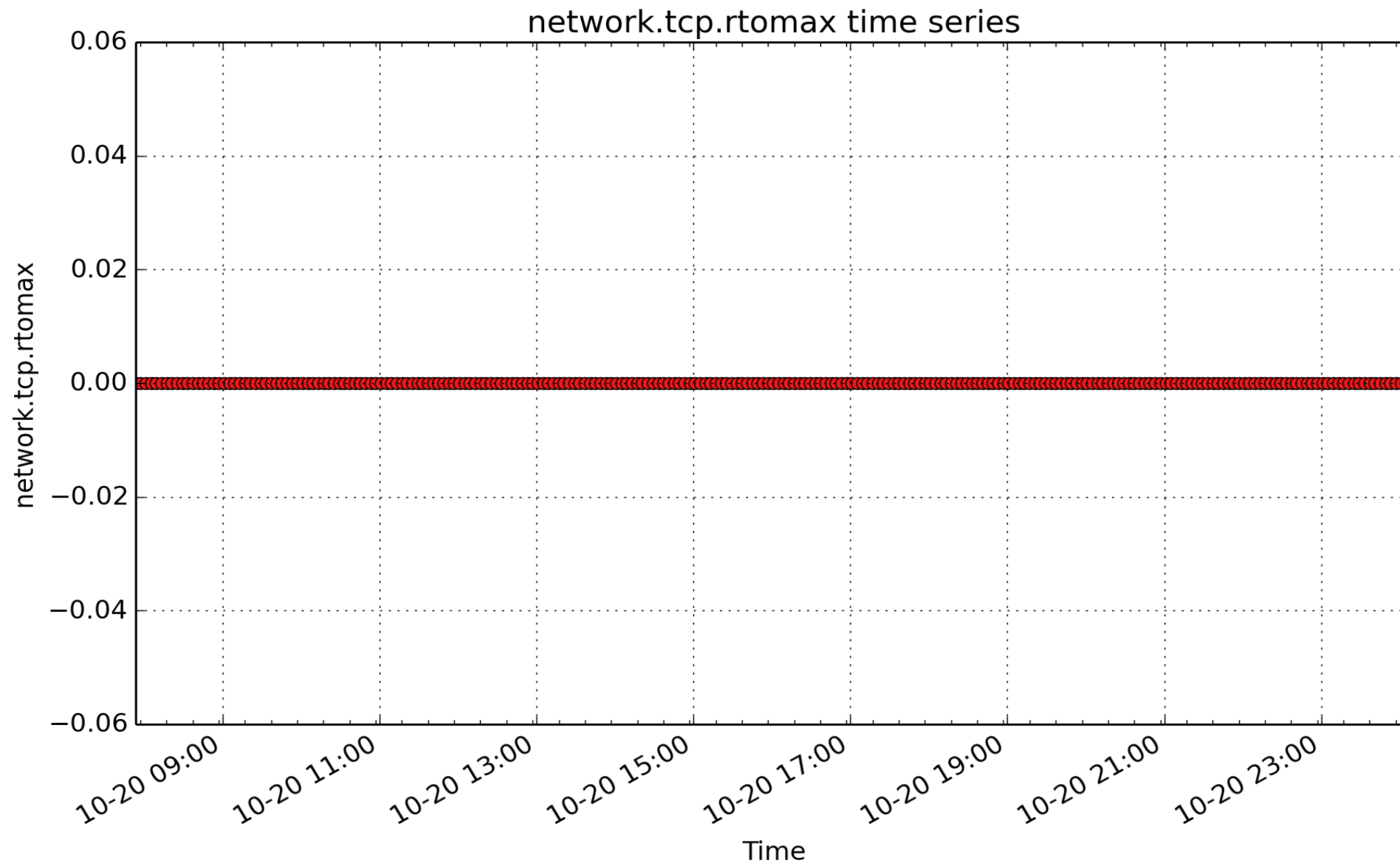
network.tcp.rcvcollapsed: Number of packets collapsed in receive queue due to low socket buffer (count - U64) - *rate converted*



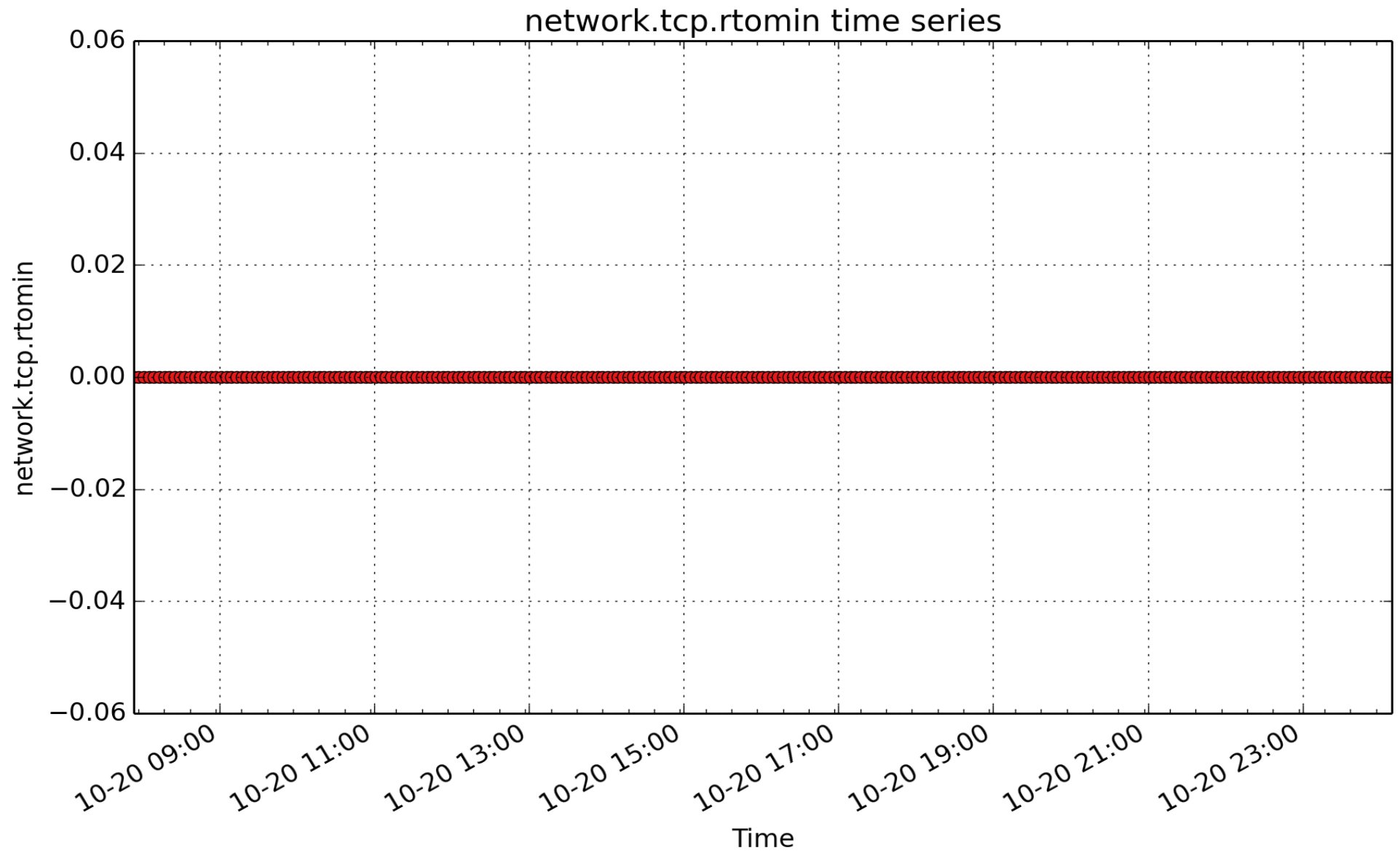
network.tcp.retranssegs: count of tcp retranssegs (count - U64) - *rate converted*



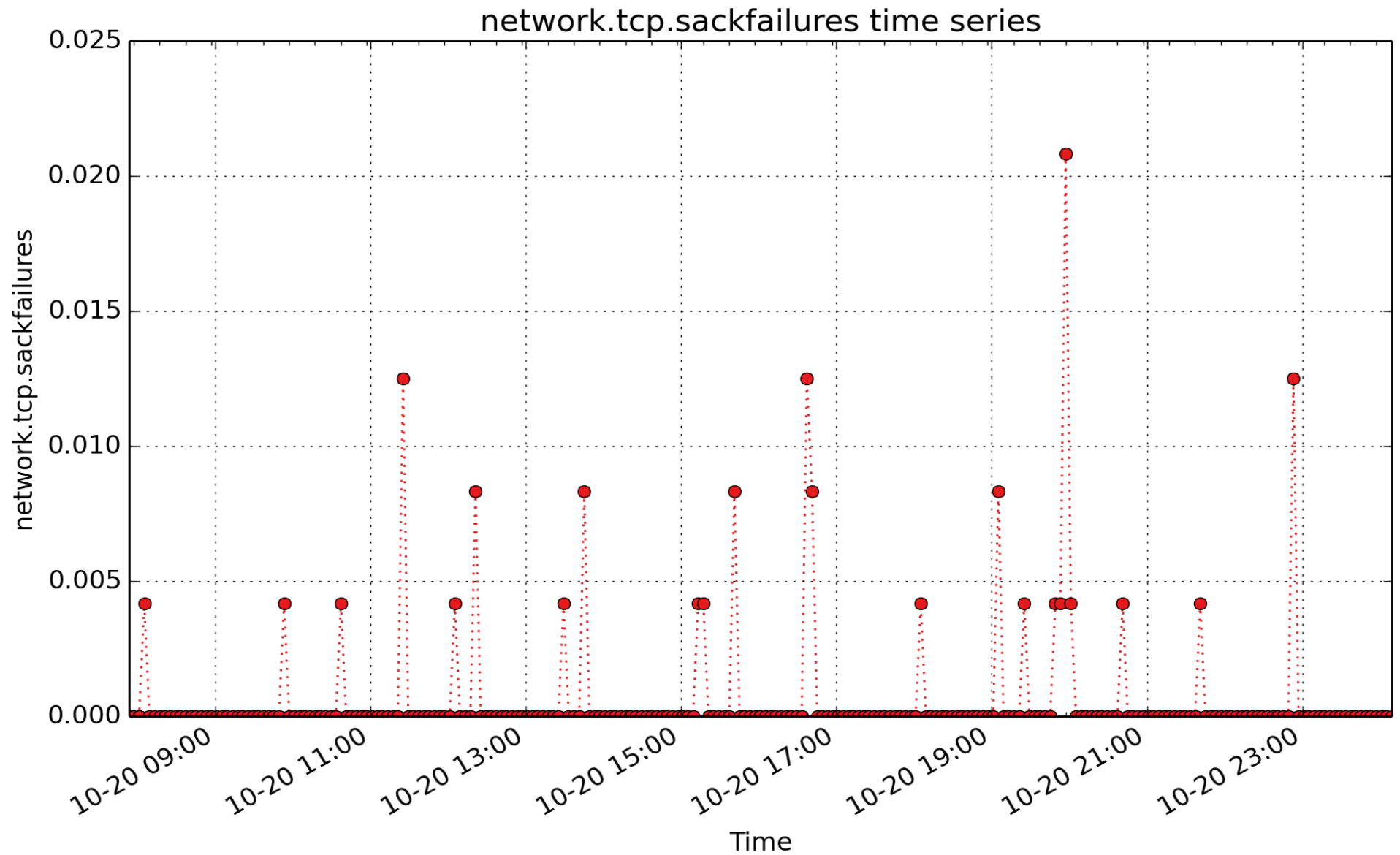
network.tcp.rtoalgorithm: count of tcp rtoalgorithm (count - U64) - *rate converted*



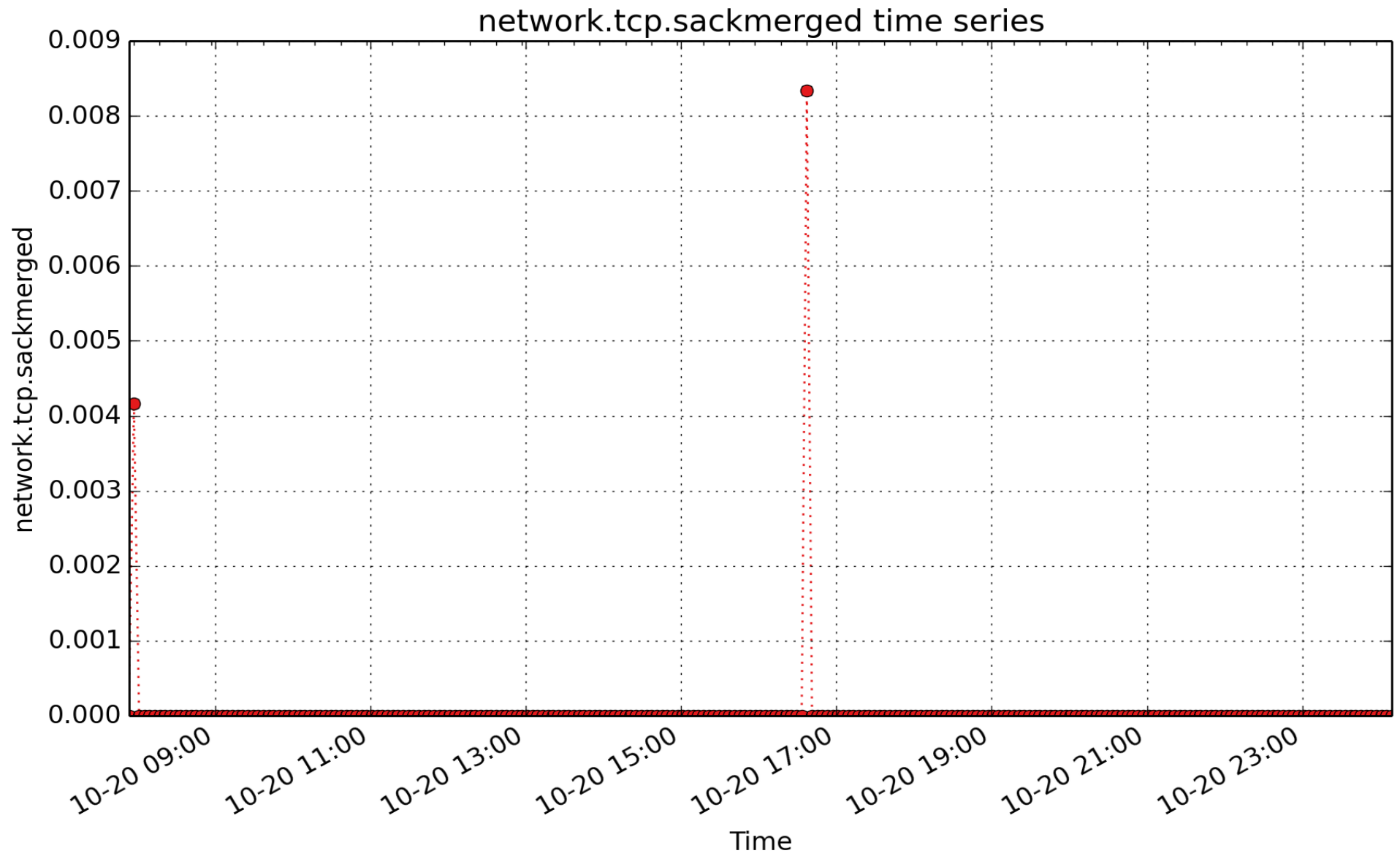
network.tcp.rtomax: count of tcp rtomax (count - U64) - *rate converted*



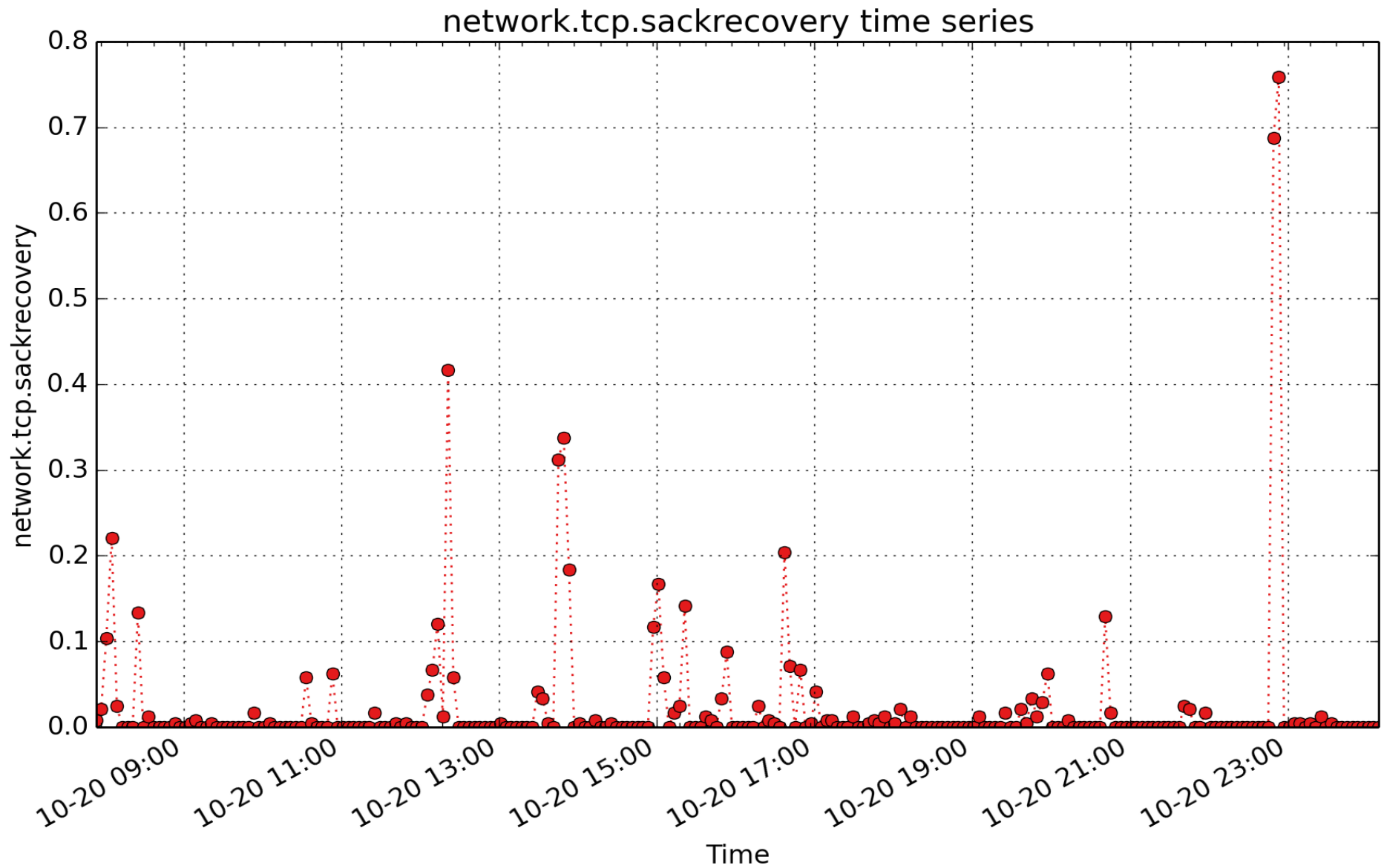
network.tcp.rtomin: count of tcp rtomin (count - U64) - *rate converted*



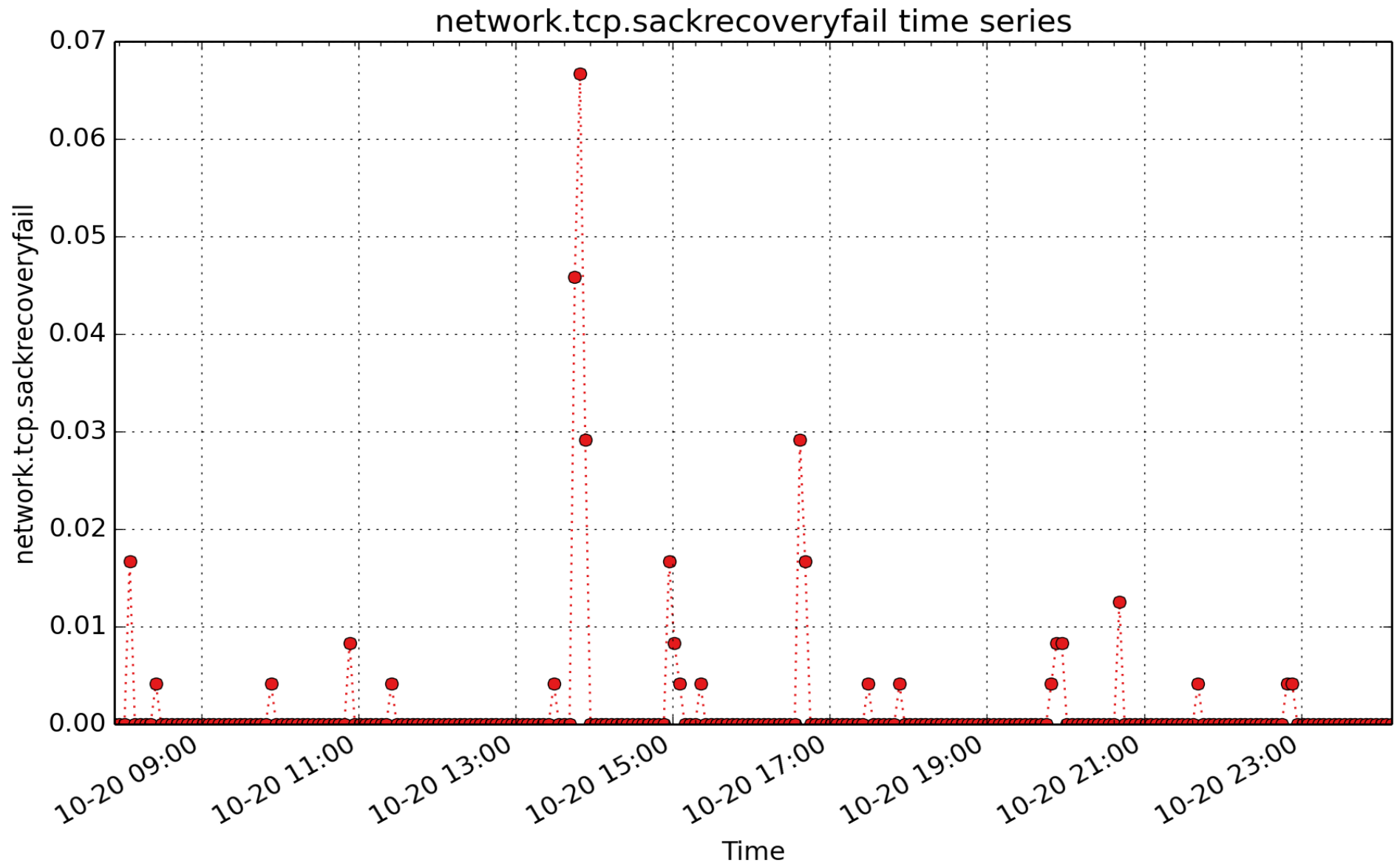
network.tcp.sackfailures: Number of timeouts after SACK recovery (count - U64) - *rate converted*



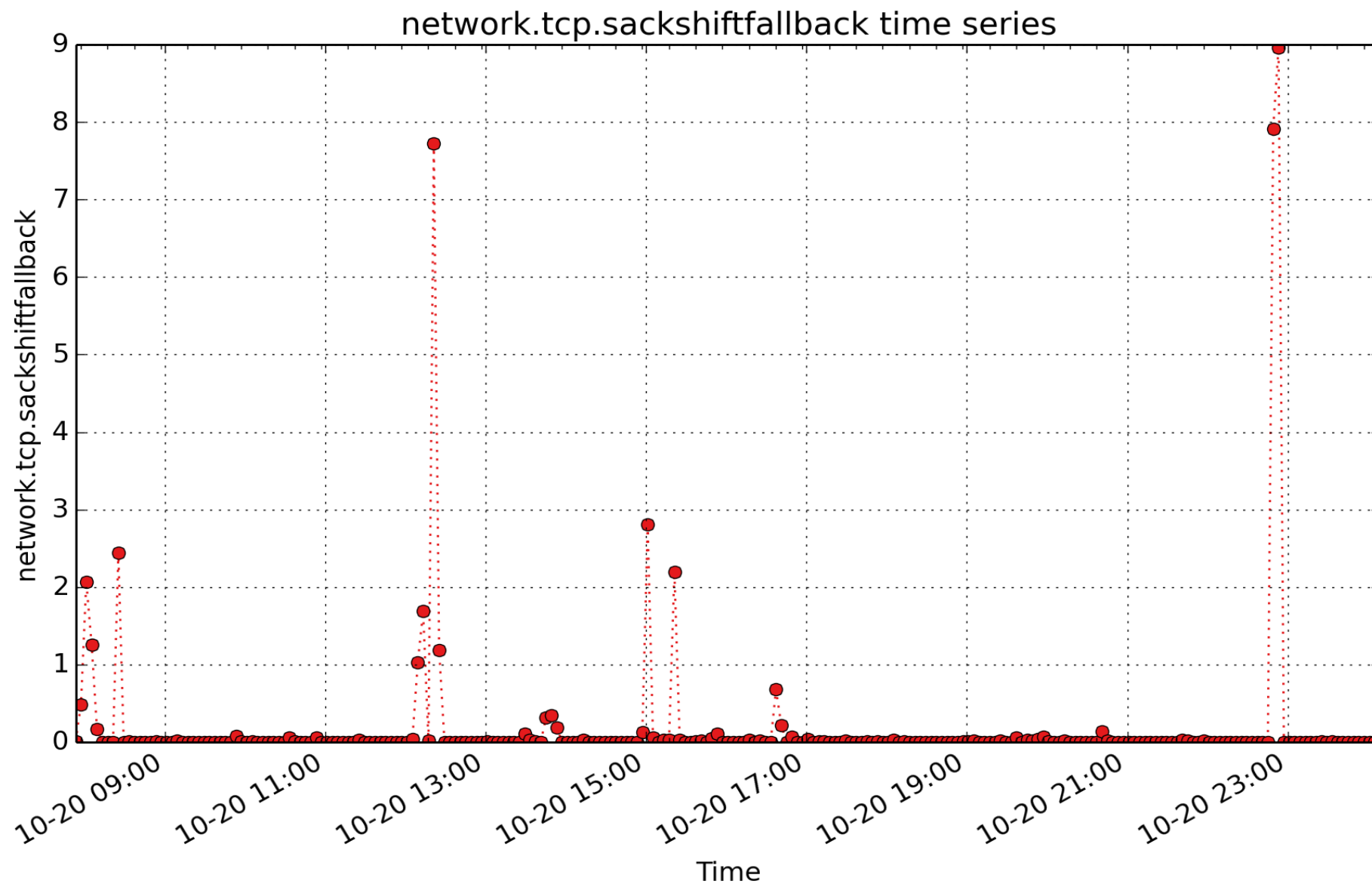
network.tcp.sackmerged: Number of SACKs merged (count - U64) - *rate converted*



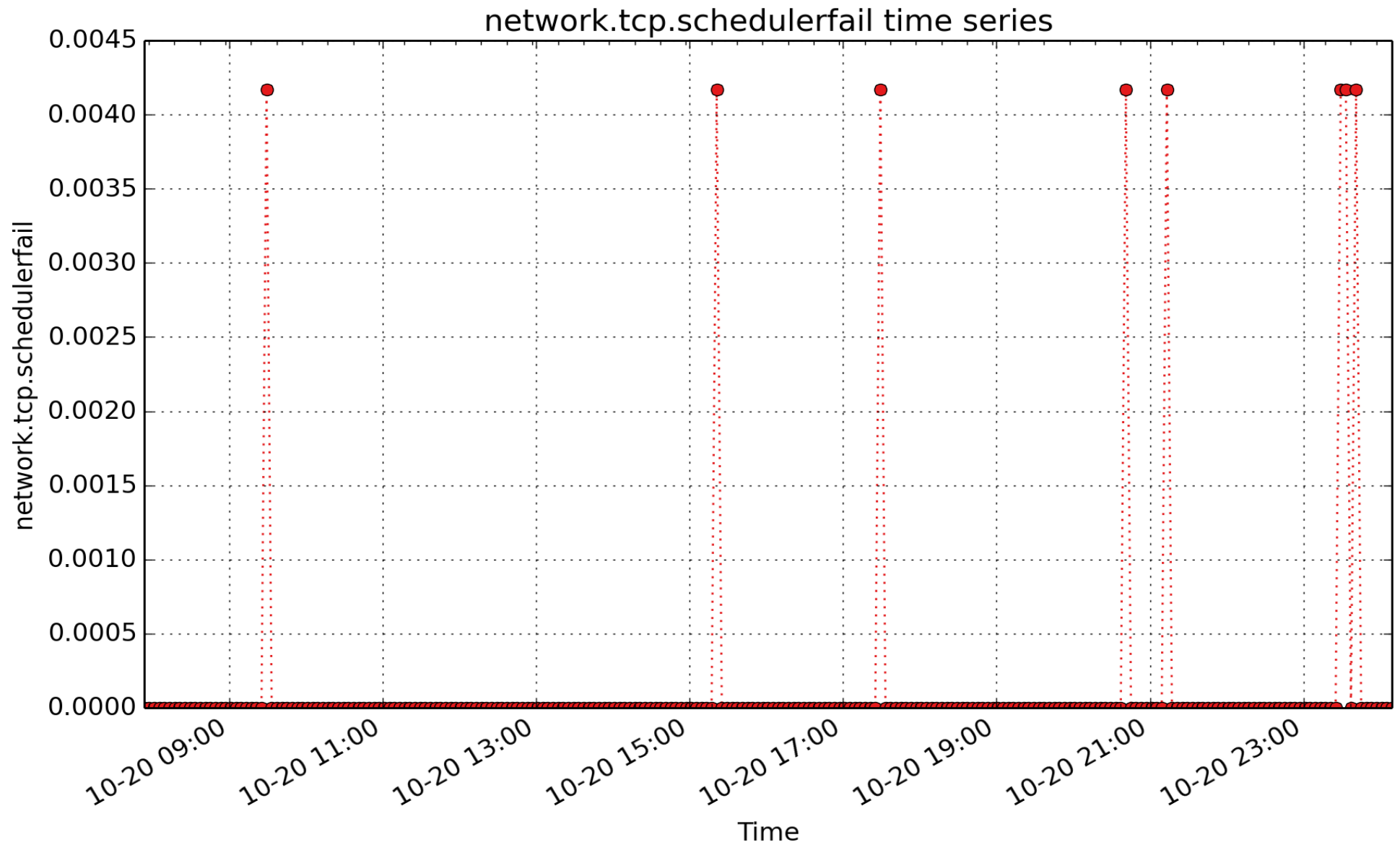
network.tcp.sackrecovery: Number of times recovered from packet loss by selective acknowledgements
(count - U64) - *rate converted*



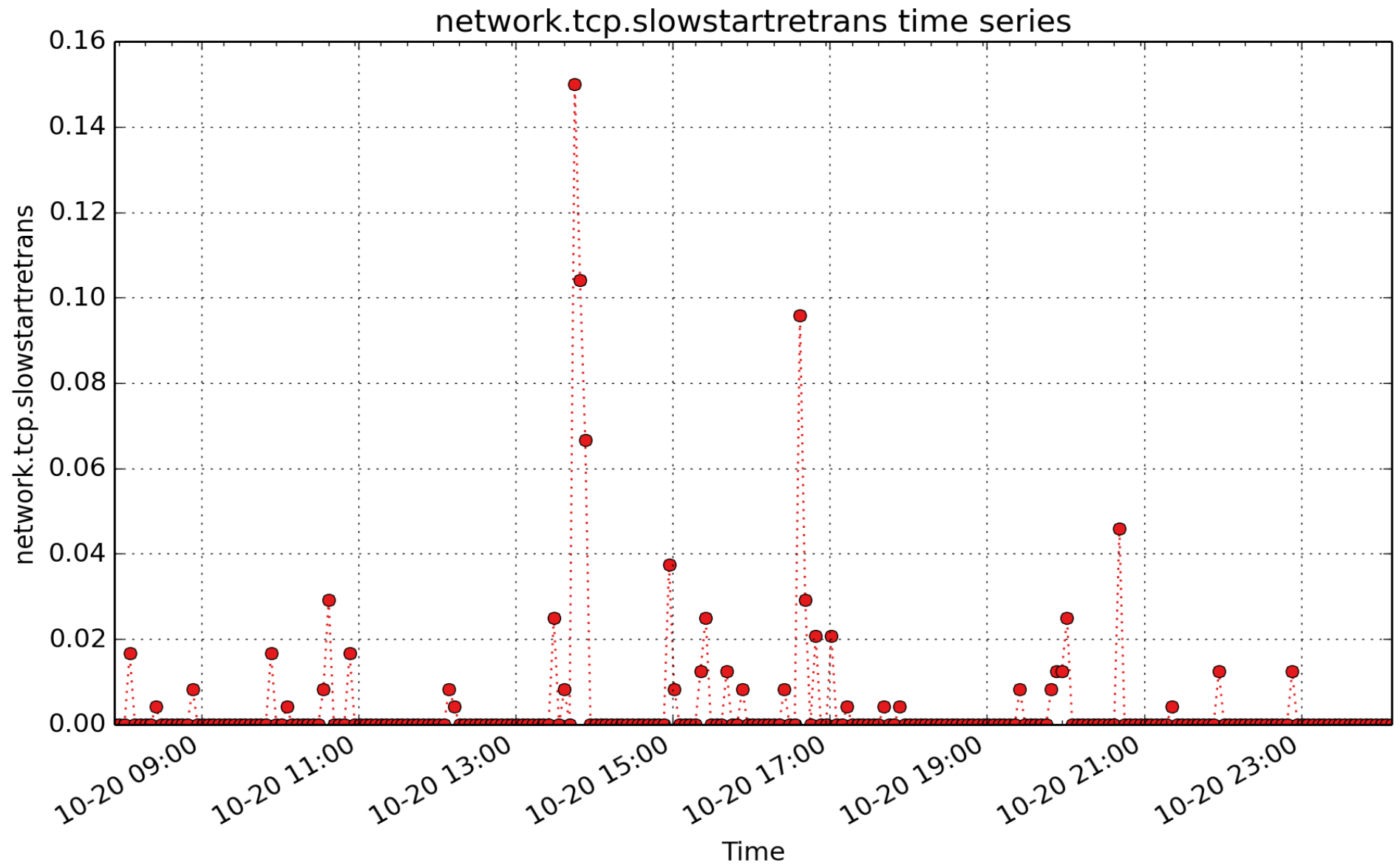
network.tcp.sackrecoveryfail: Number of SACK retransmits failed (count - U64) - *rate converted*



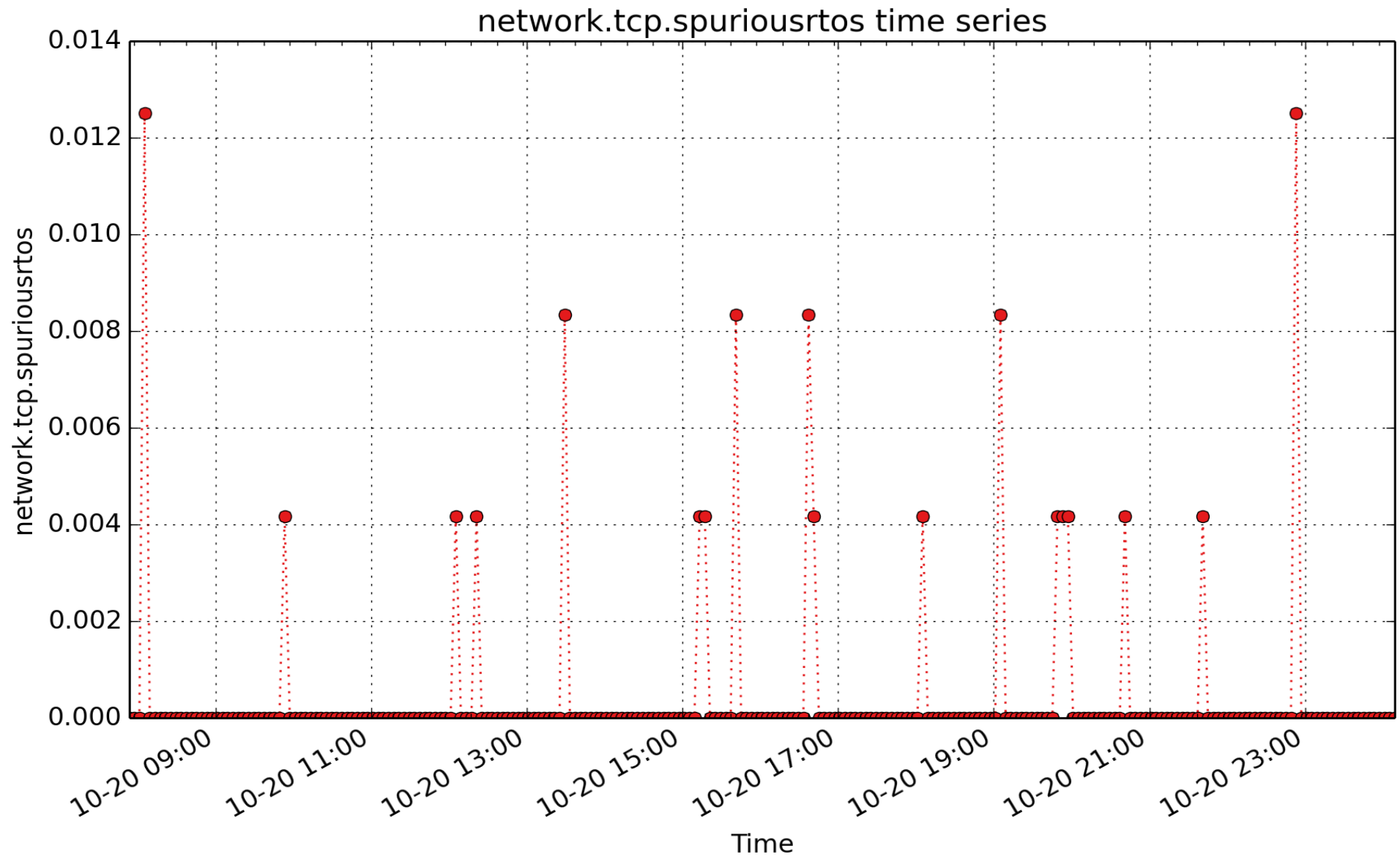
network.tcp.sackshiftfallback: Number of SACKs fallbacks (count - U64) - *rate converted*



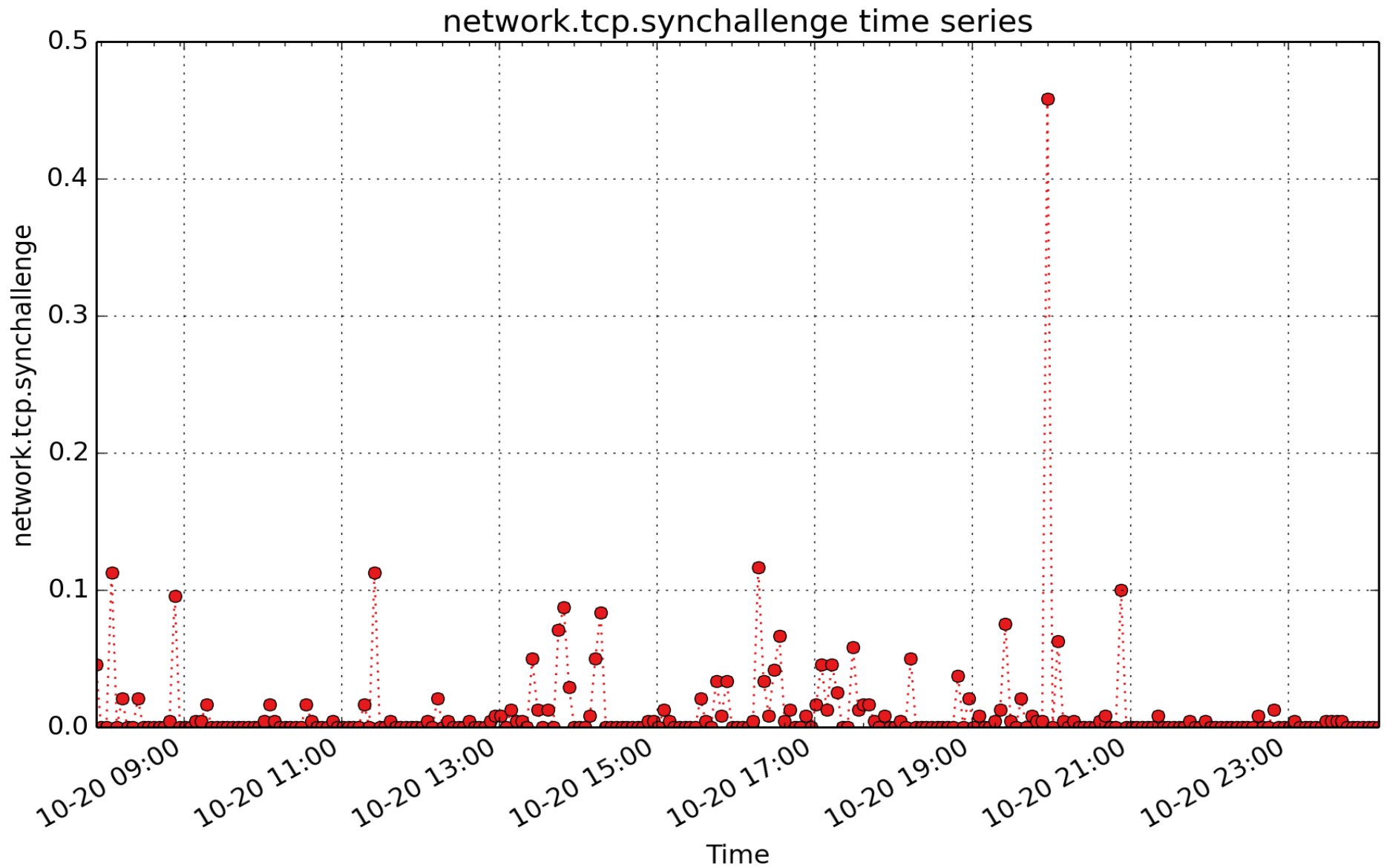
network.tcp.schedulerfail: Number of times receiver scheduled too late for direct processing (count - U64) -
rate converted



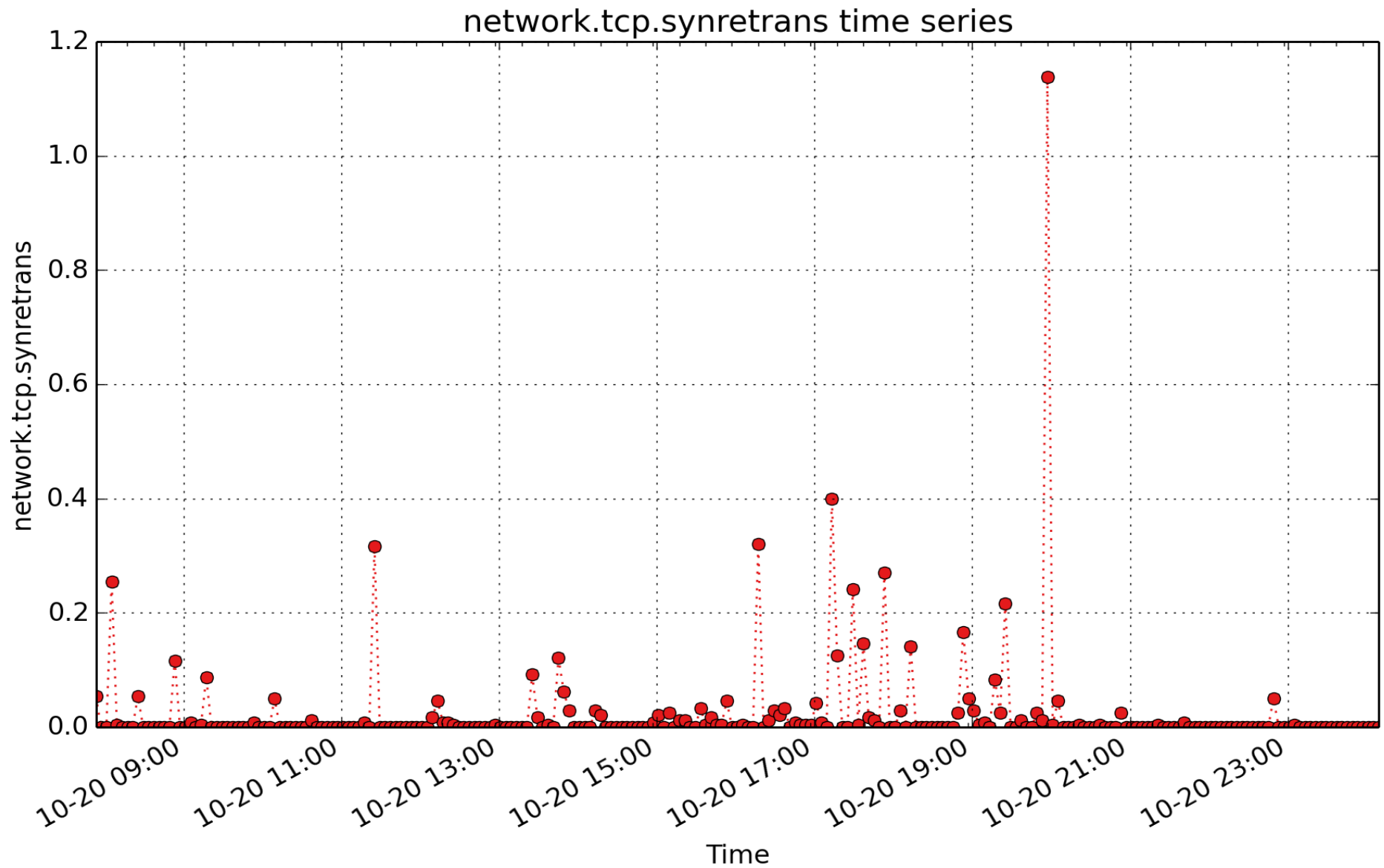
network.tcp.slowstartretrans: Number of retransmits in slow start (count - U64) - *rate converted*



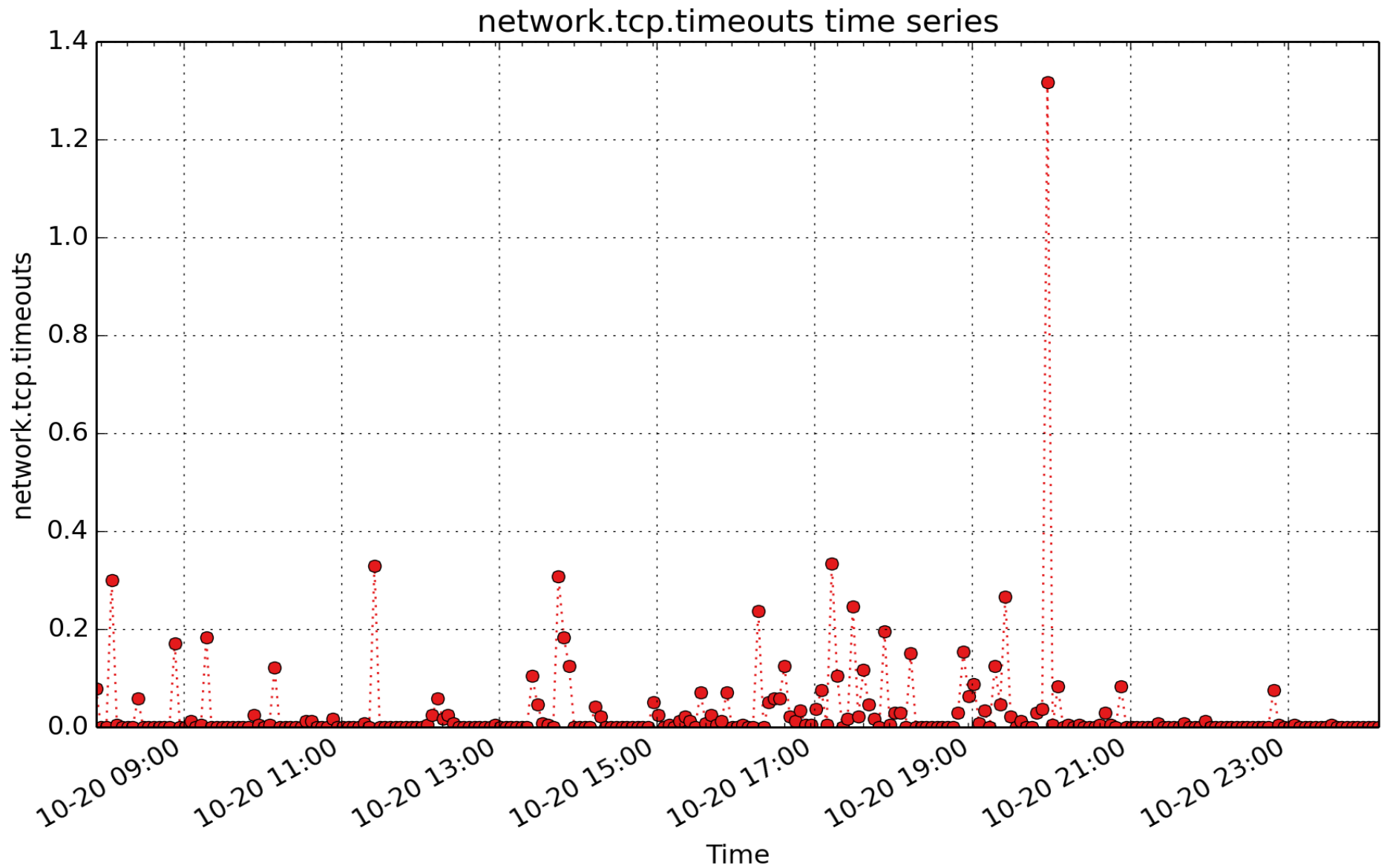
network.tcp.spuriousrtos: Number of FRTO's successfully detected spurious RTOs (count - U64) - *rate converted*



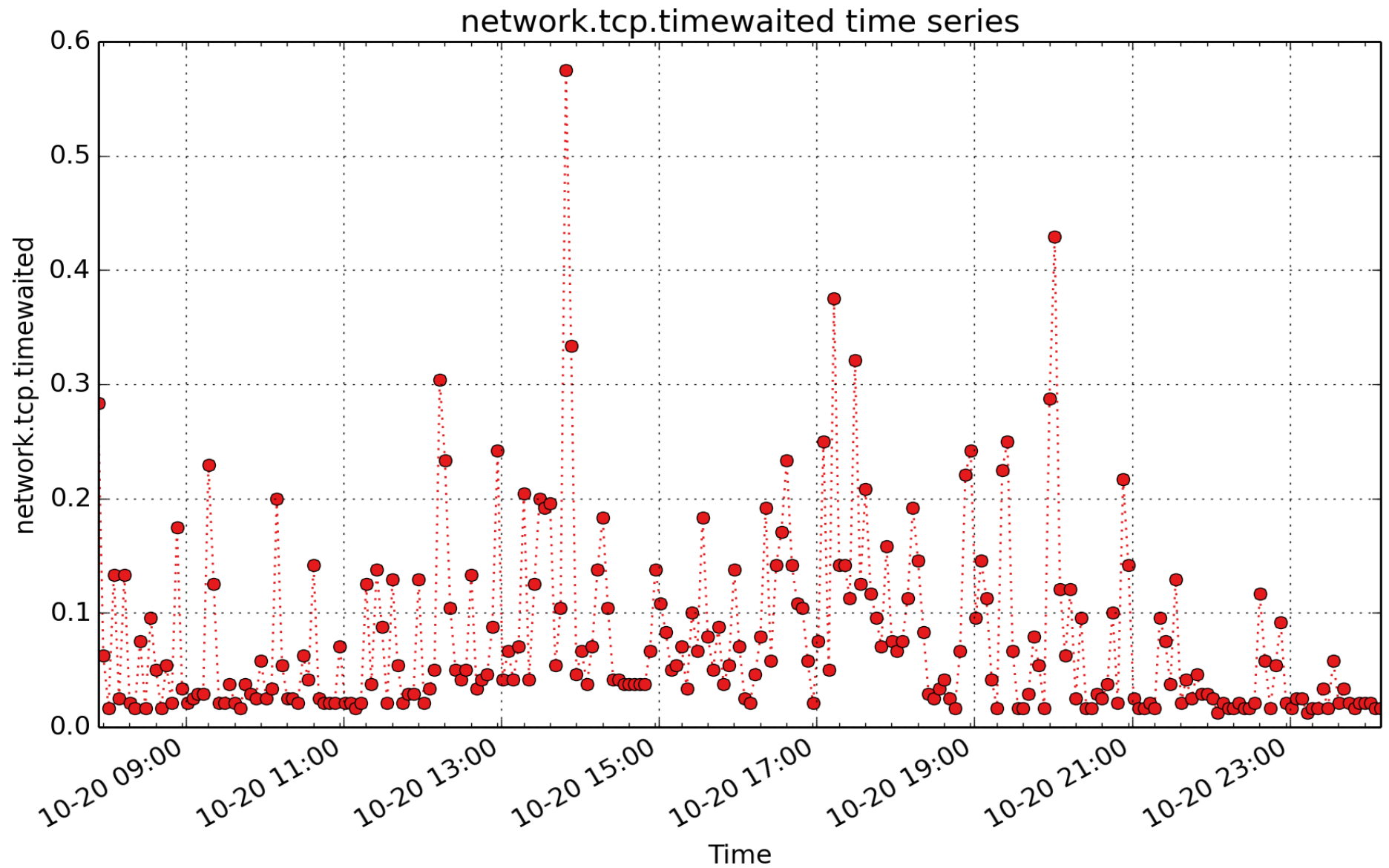
network.tcp.synchallenge: Number of challenge ACKs sent in response to SYN packets (count - U64) - *rate converted*



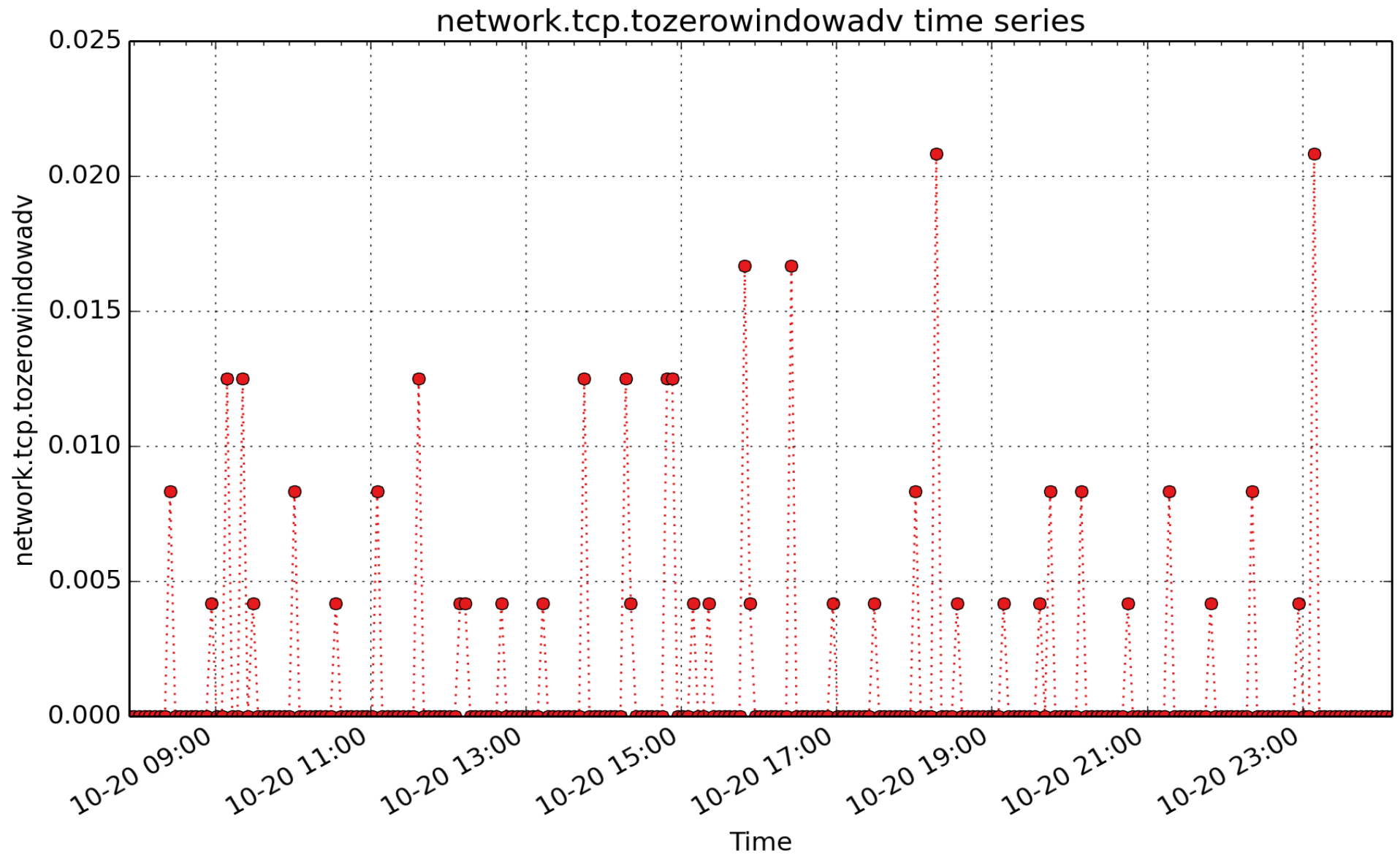
network.tcp.synretrans: Number of SYN-SYN/ACK retransmits to break down retransmissions in SYN, fast/timeout retransmits. (count - U64) - *rate converted*



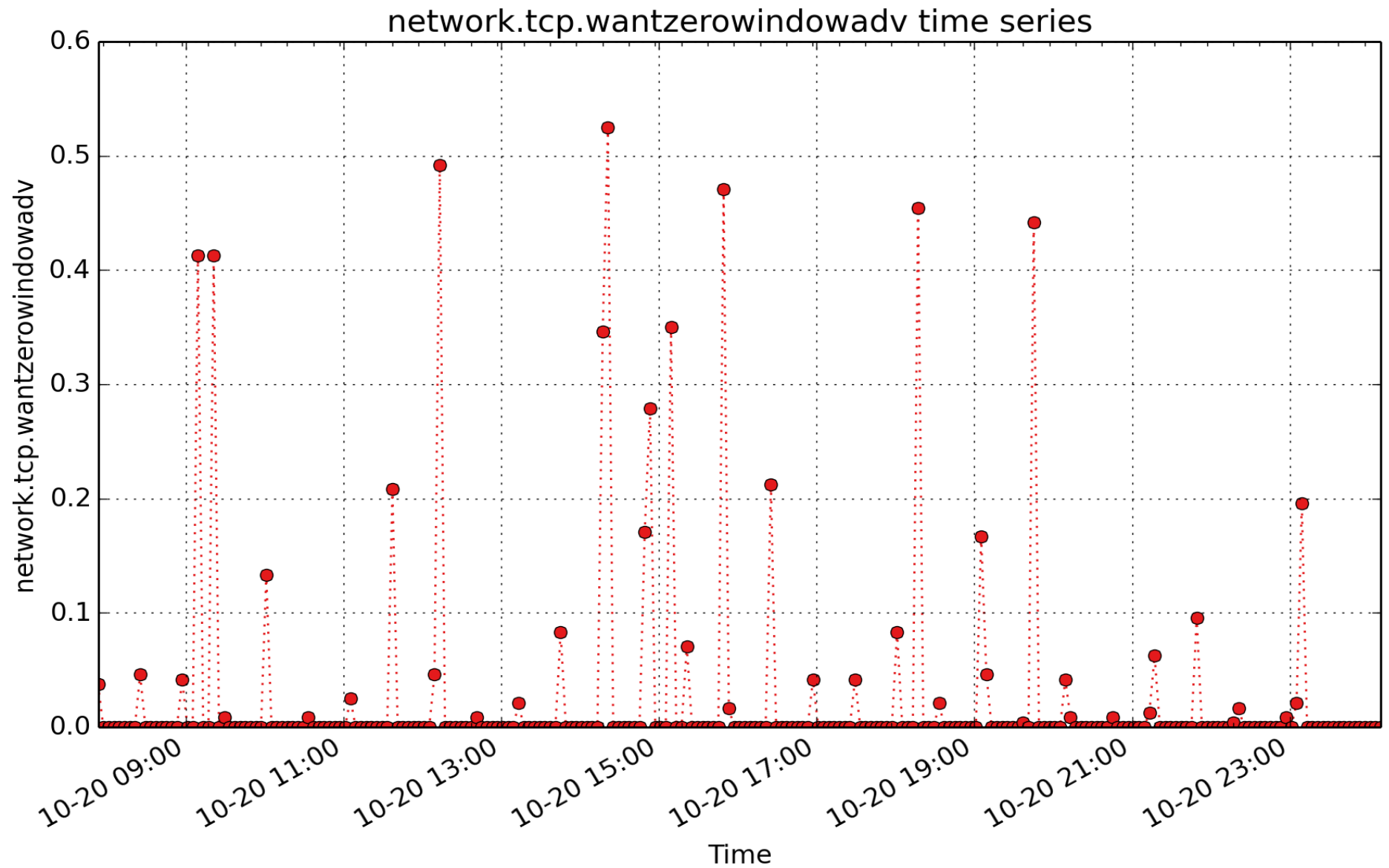
network.tcp.timeouts: Number of other TCP timeouts (count - U64) - *rate converted*



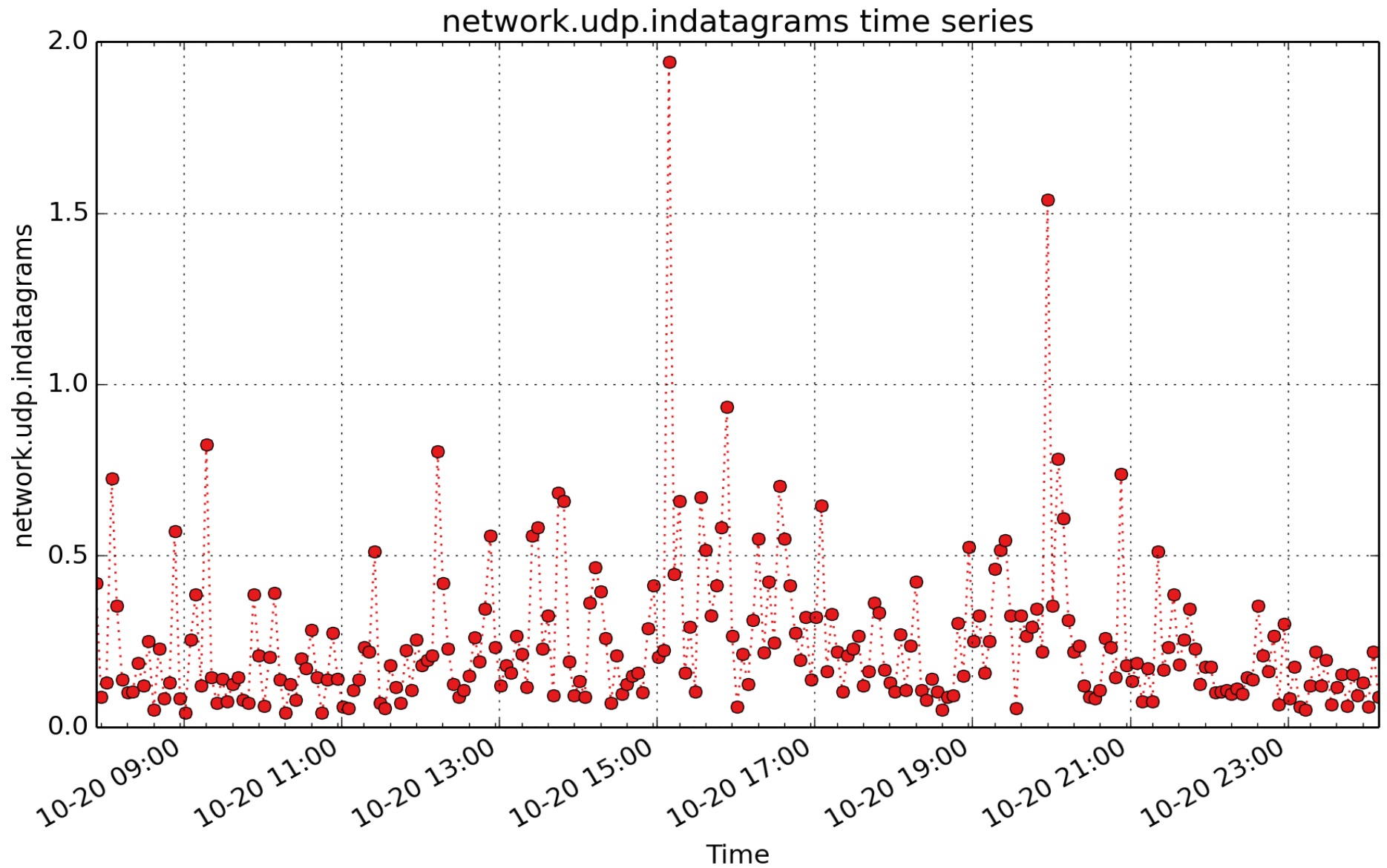
network.tcp.timewaited: Number of TCP sockets finished time wait in fast timer (count - U64) - *rate converted*



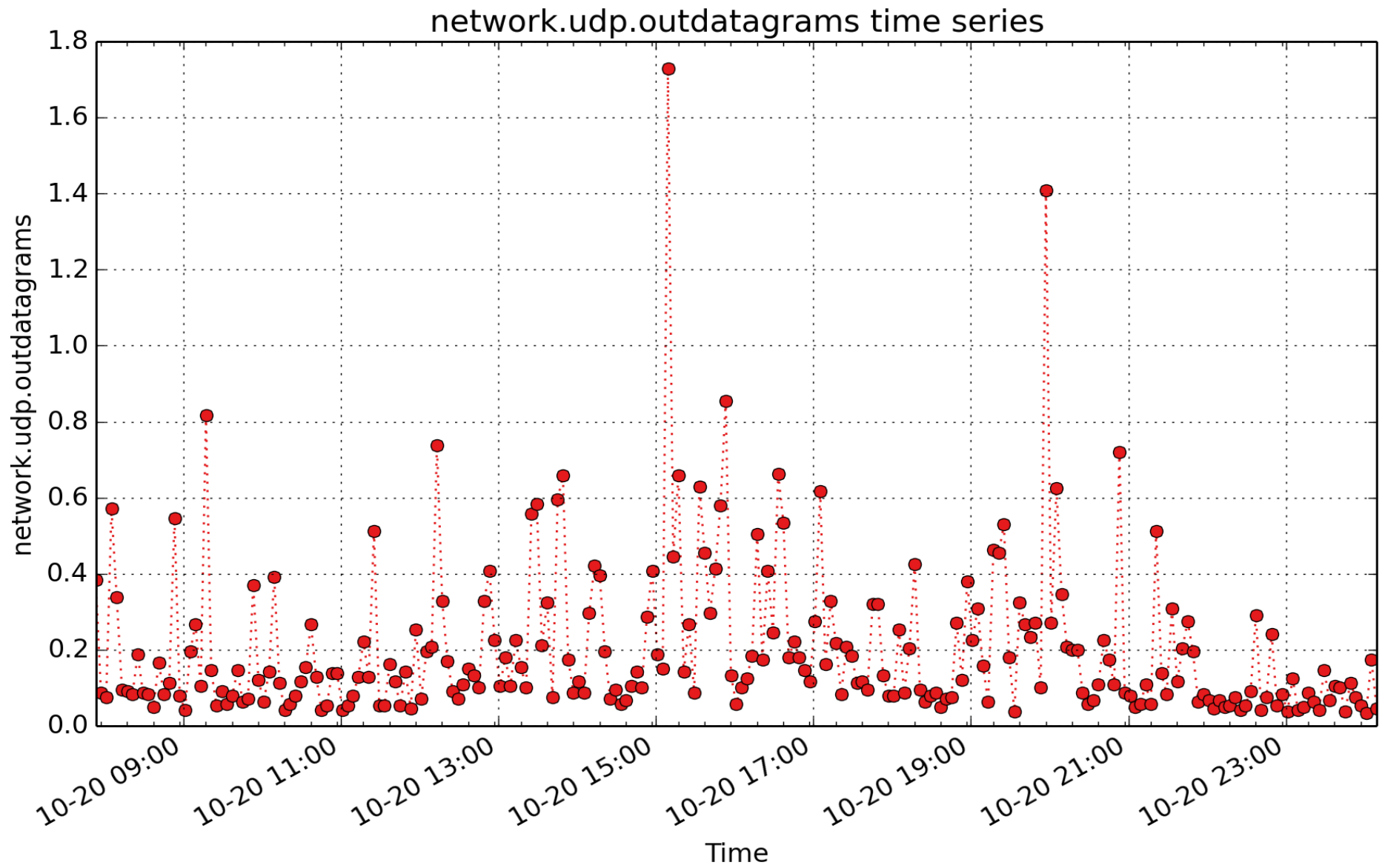
network.tcp.tozerowindowadv: Number of times window went from non-zero to zero (count - U64) - *rate converted*



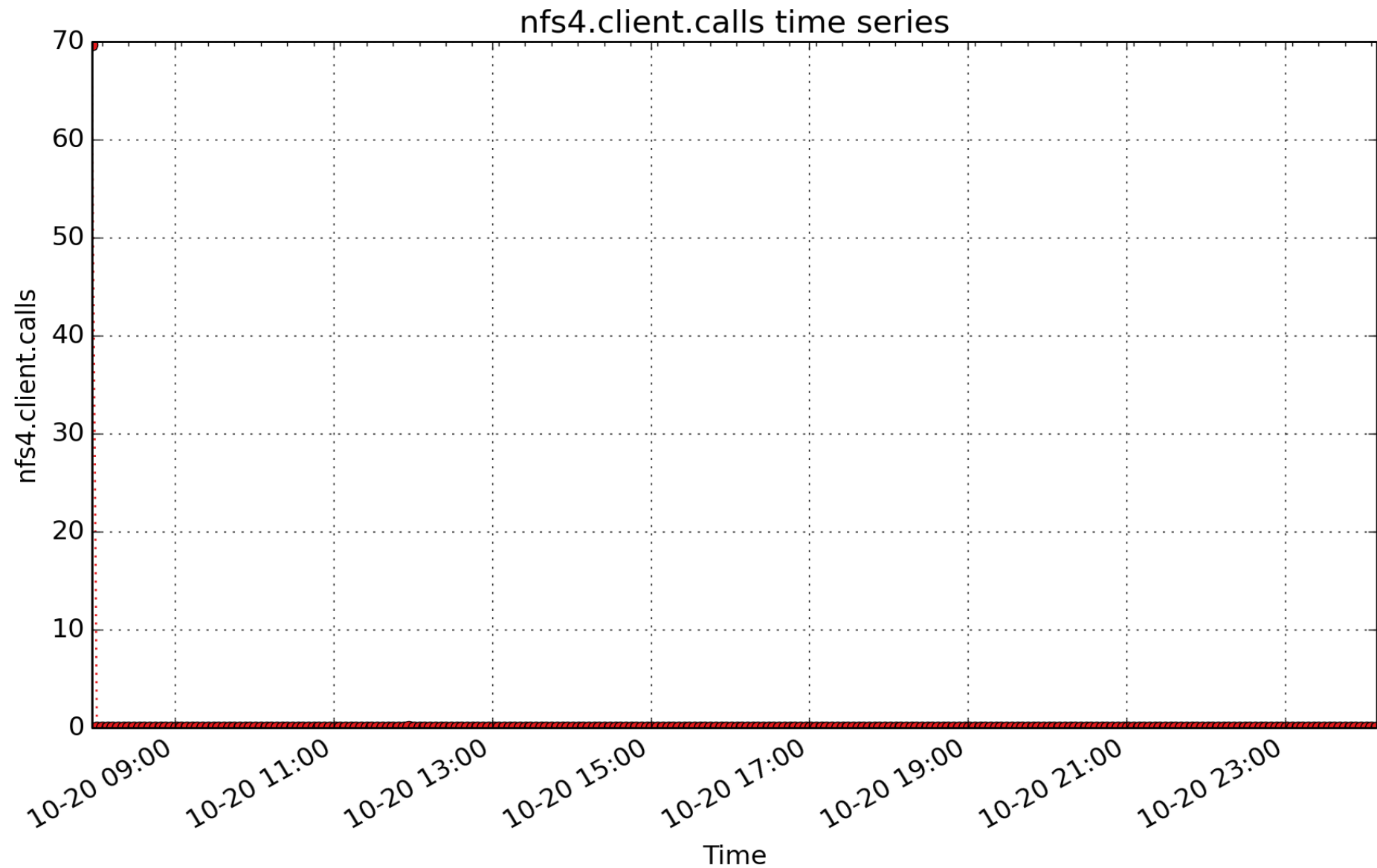
network.tcp.wantzerowindowadv: Number of times zero window announced (count - U64) - *rate converted*



network.udp.indatagrams: count of udp indatagrams (count - U64) - *rate converted*

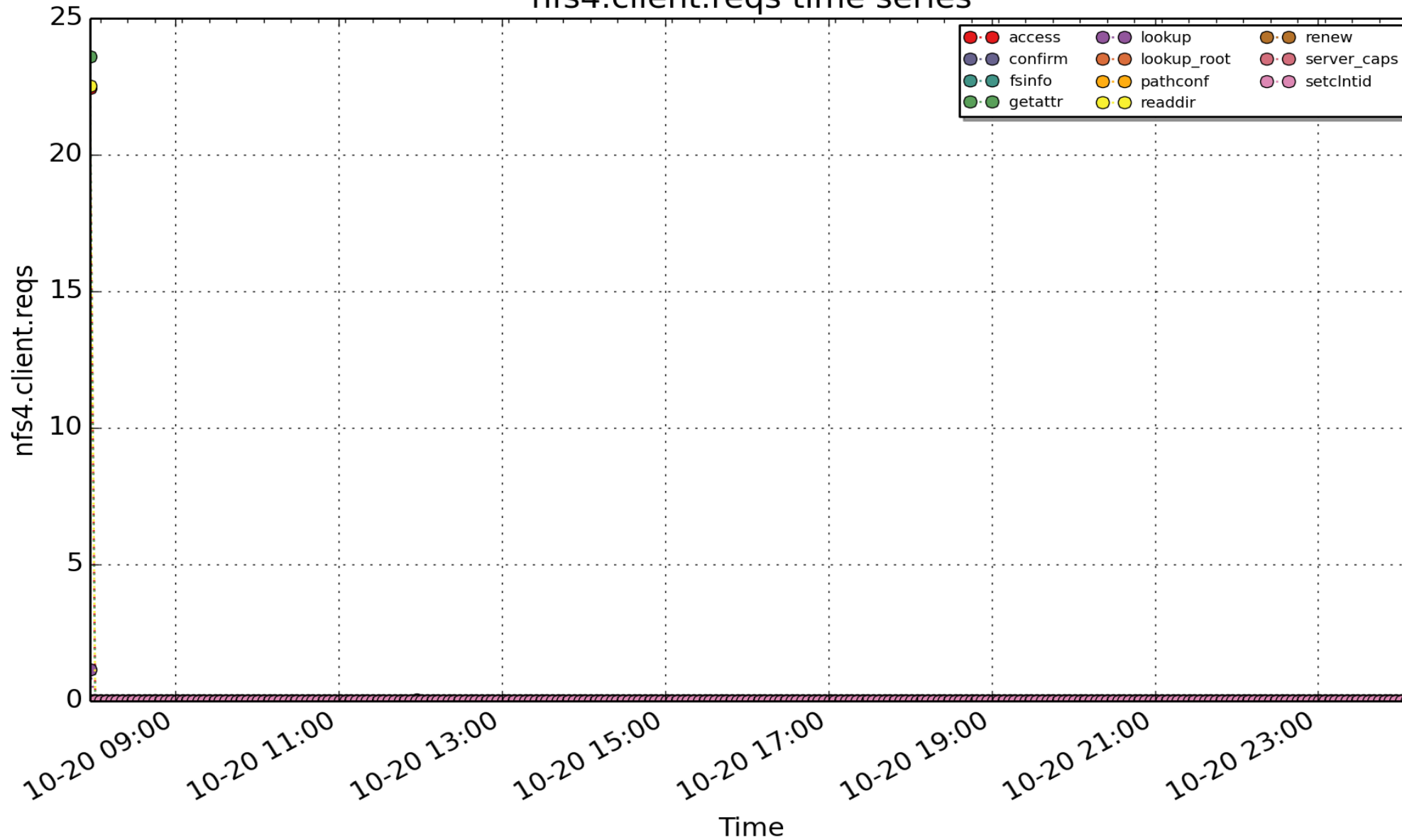


network.udp.outdatagrams: count of udp outdatagrams (count - U64) - *rate converted*

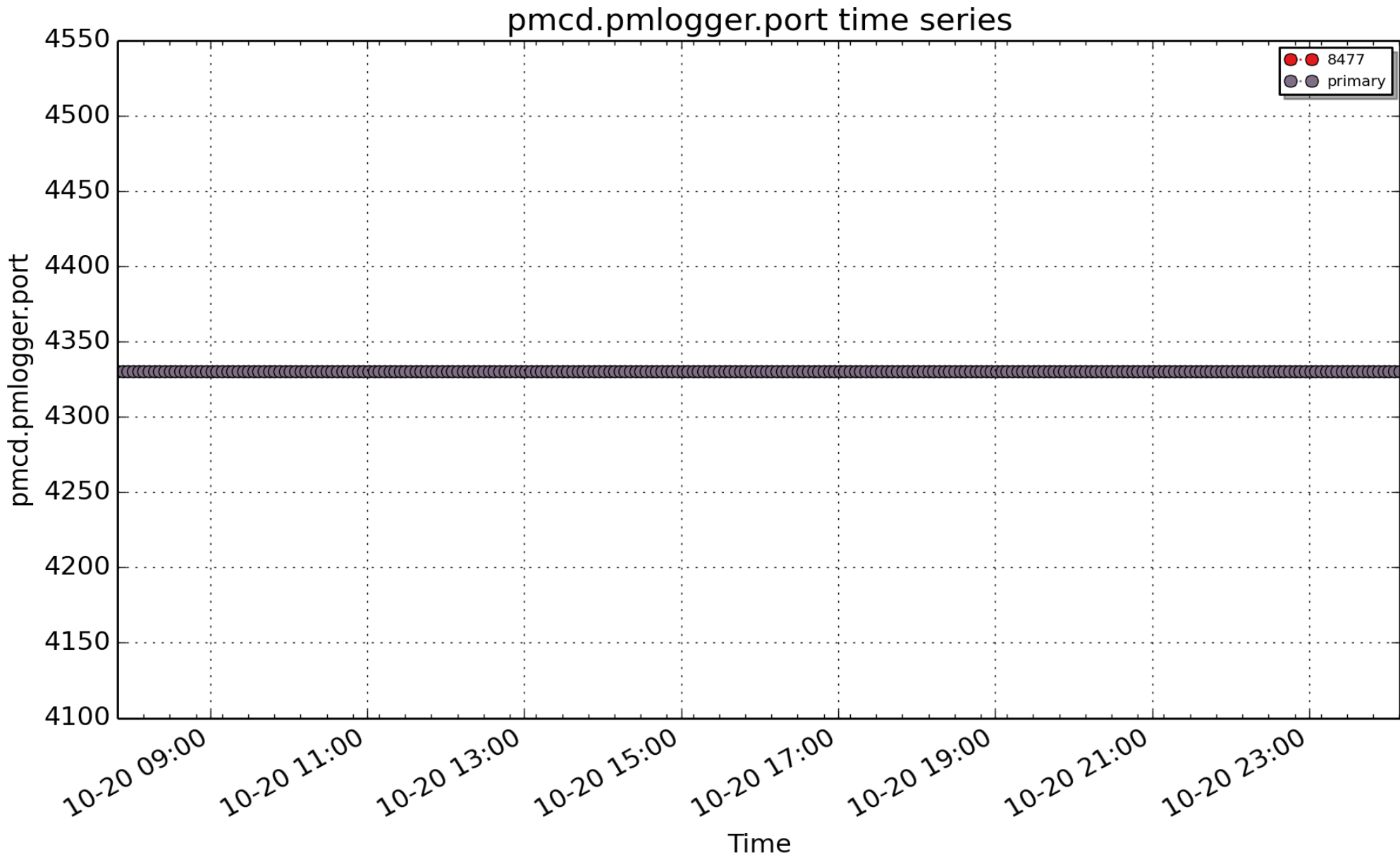


nfs4.client.calls: cumulative total of client NFSv4 requests (count - U32) - *rate converted*

nfs4.client.reqs time series

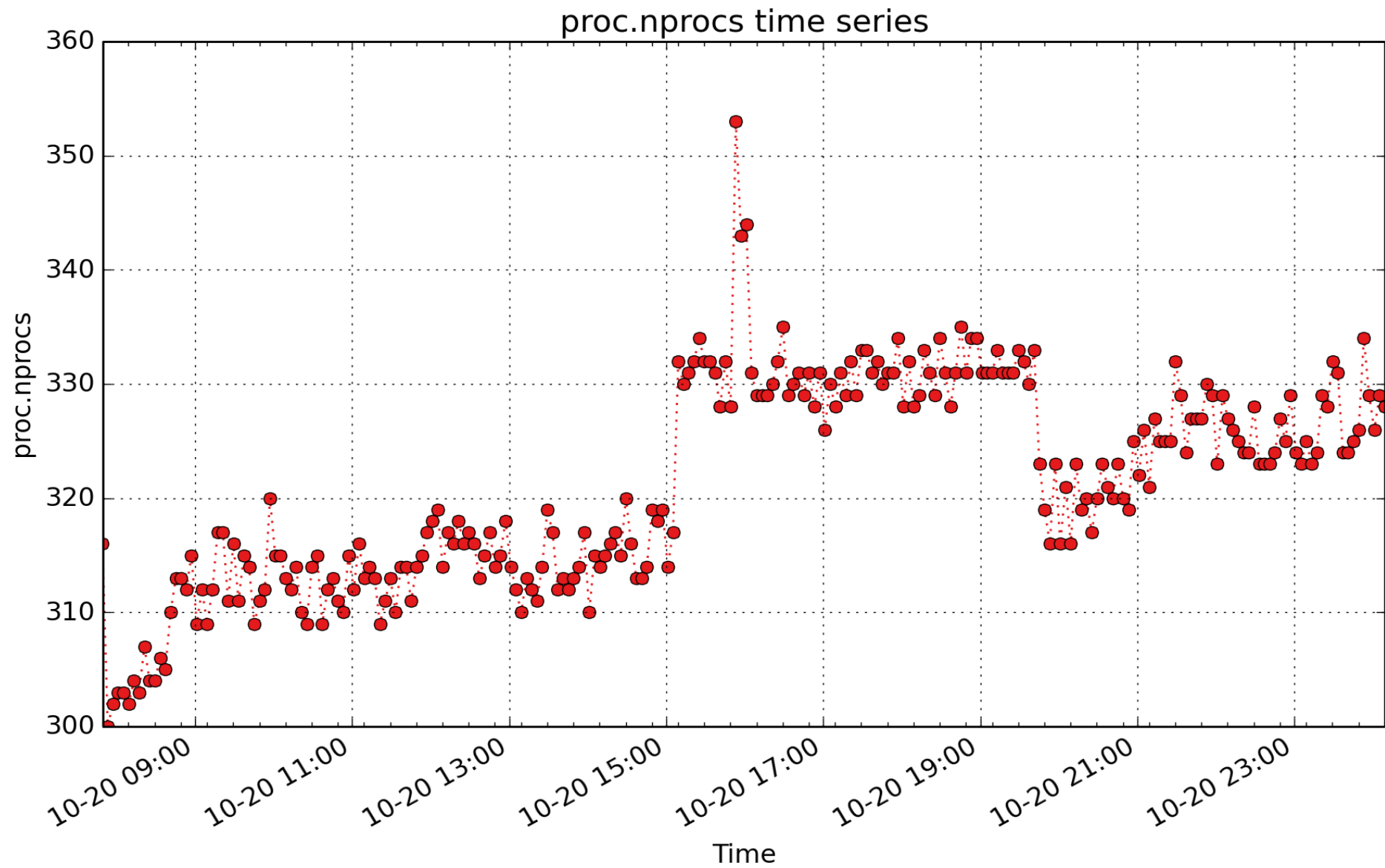


nfs4.client.reqs: cumulative total for each client NFSv4 request type (count - U32) - *rate converted*

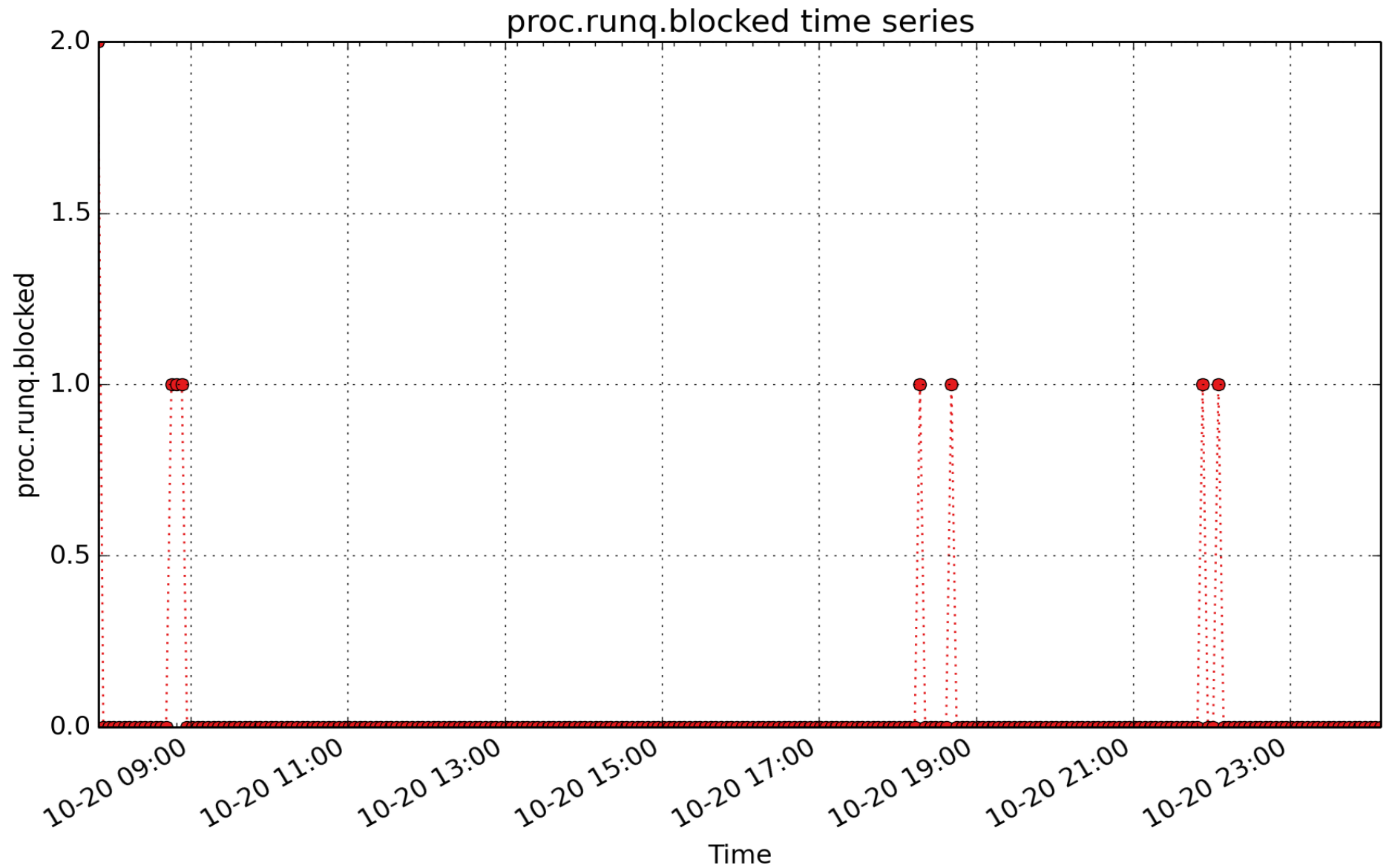


pmcd.pmlogger.port: Each pmlogger instance has a port for receiving log control information. This metric is a list of the active pmlogger control ports on the same machine as this PMCD (i.e. the host identified in the corresponding pmcd.pmlogger.host metric). The instance names are process ids of the active pmloggers. The

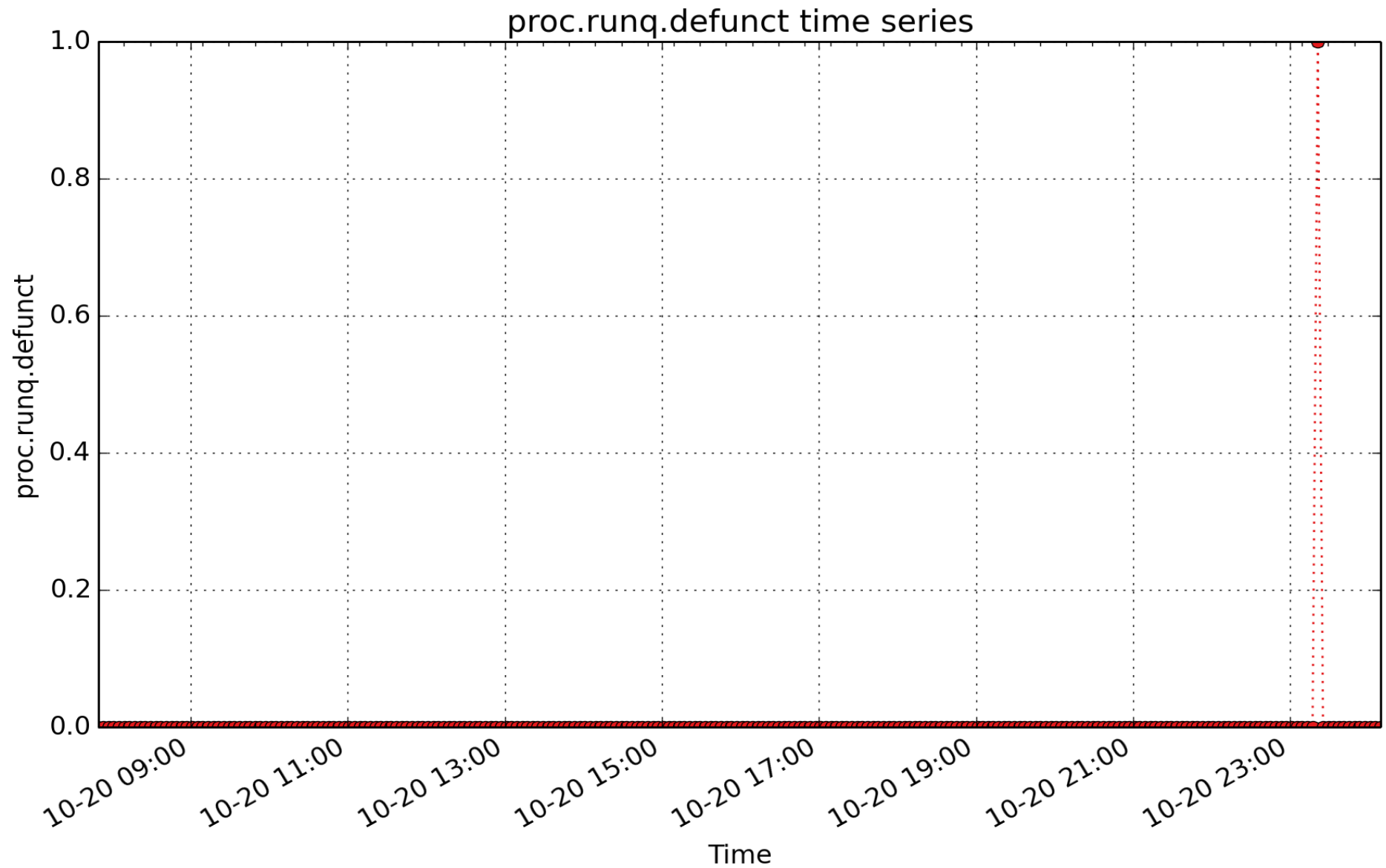
primary pmlogger has an extra instance with the instance name "primary" and an instance id of zero (in addition to its normal process id instance). (- U32)



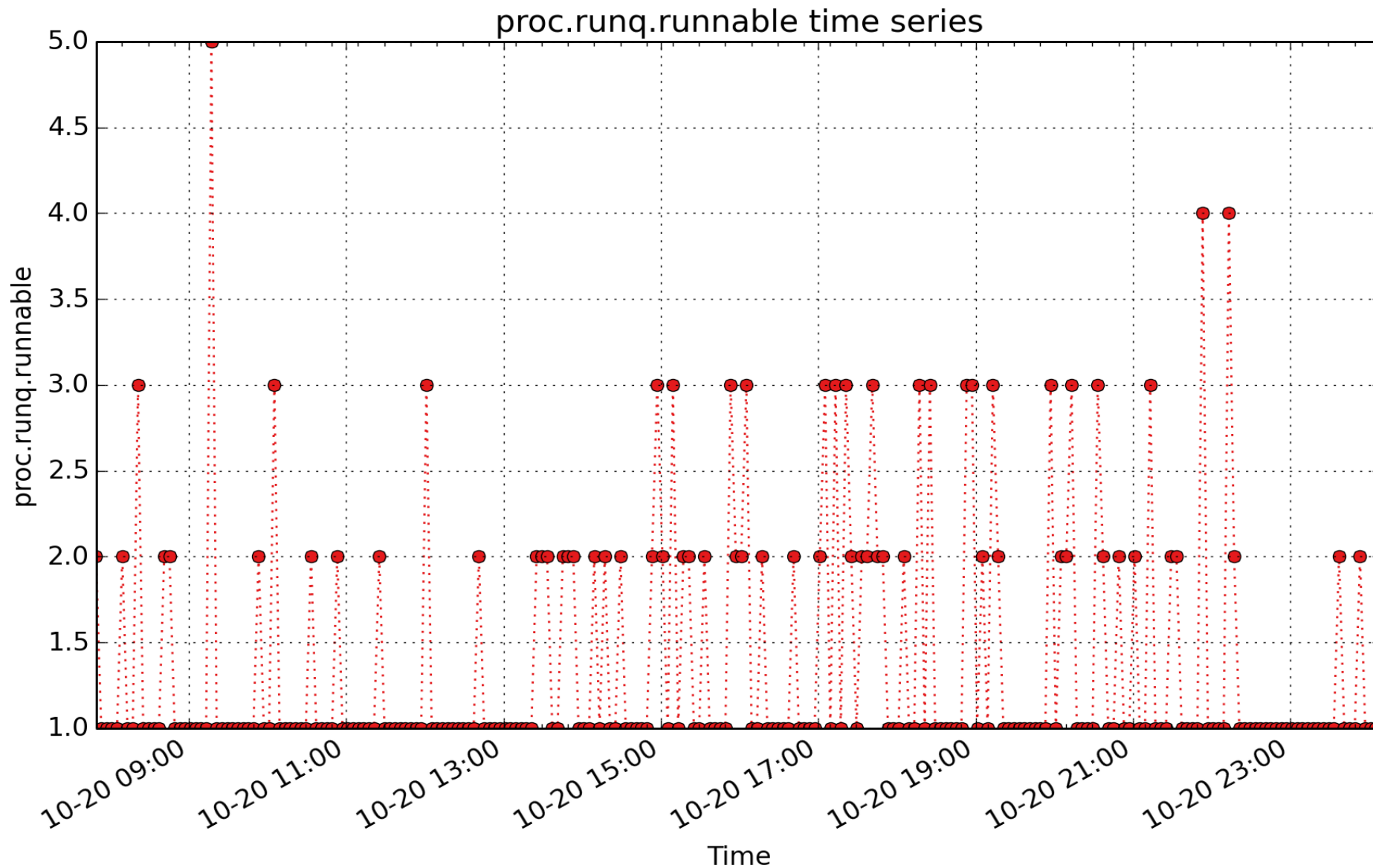
proc.nprocs: instantaneous number of processes (- U32)



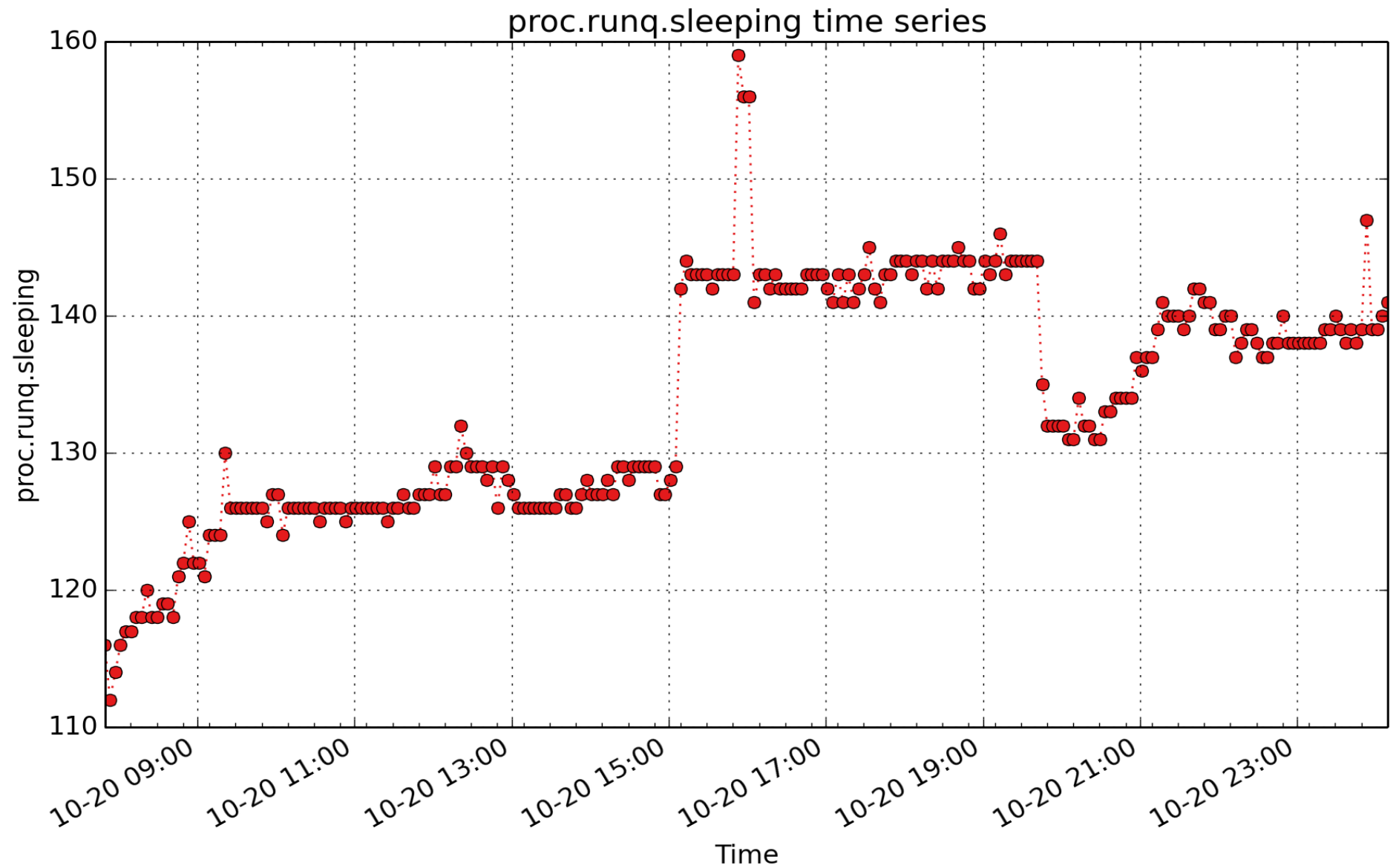
proc.runq.blocked: Instantaneous number of processes in uninterruptible sleep, state 'D' in ps (count - 32)



proc.runq.defunct: Instantaneous number of defunct/zombie processes, state 'Z' in ps (count - 32)

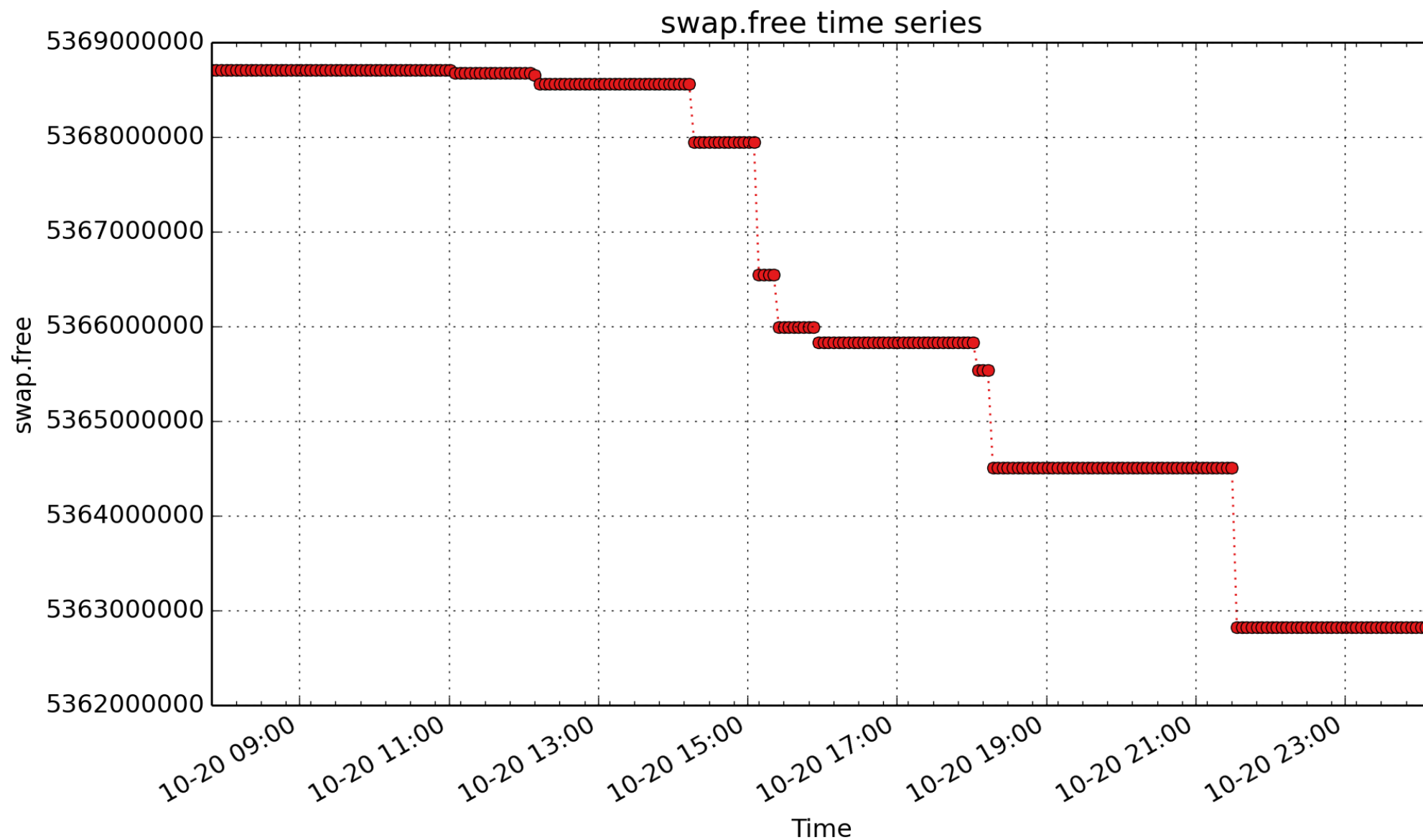


proc.runq.runnable: Instantaneous number of runnable (on run queue) processes, state 'R' in ps (count - 32)

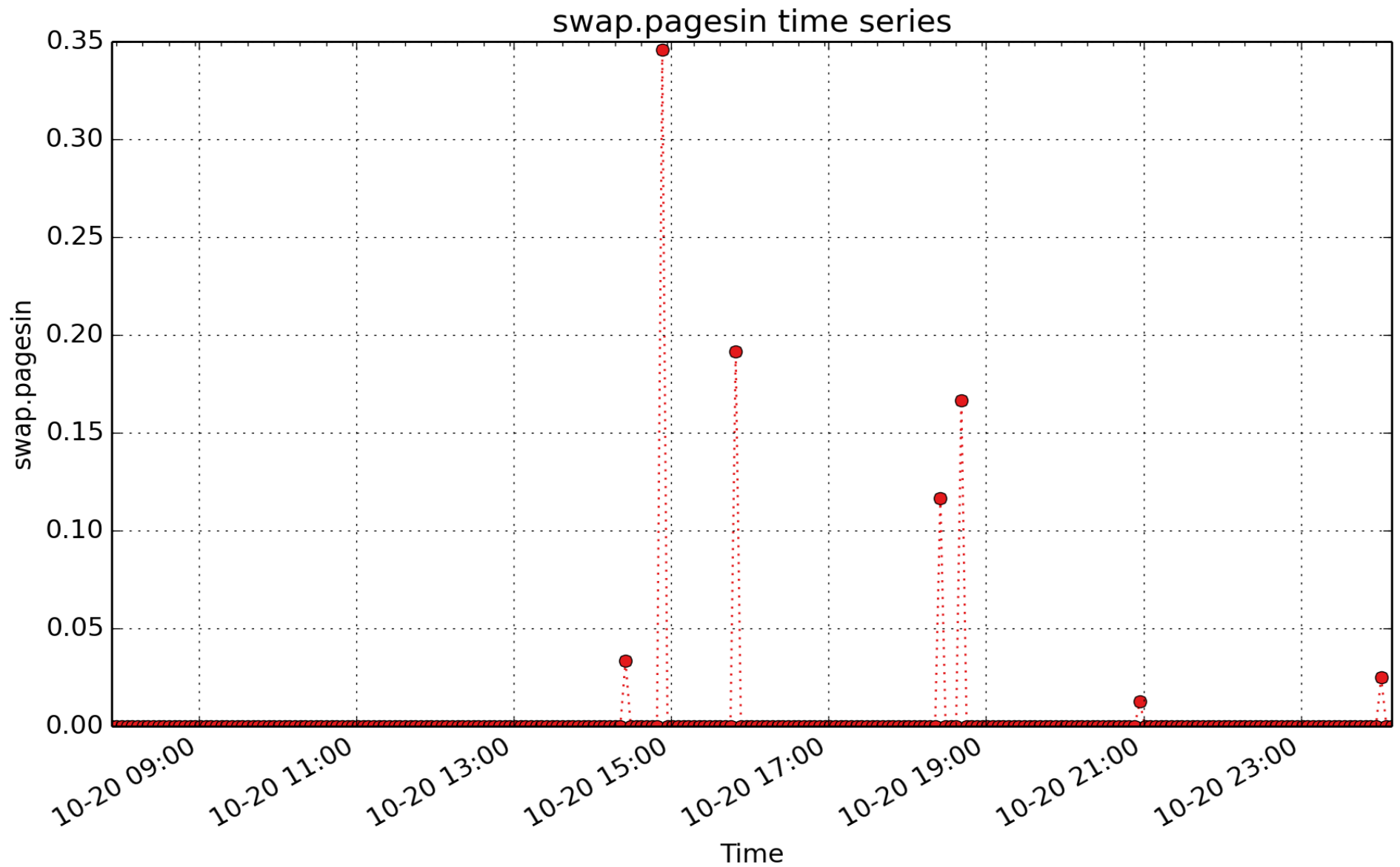


proc.runq.sleeping: Instantaneous number of processes sleeping, state 'S' in ps (count - 32)

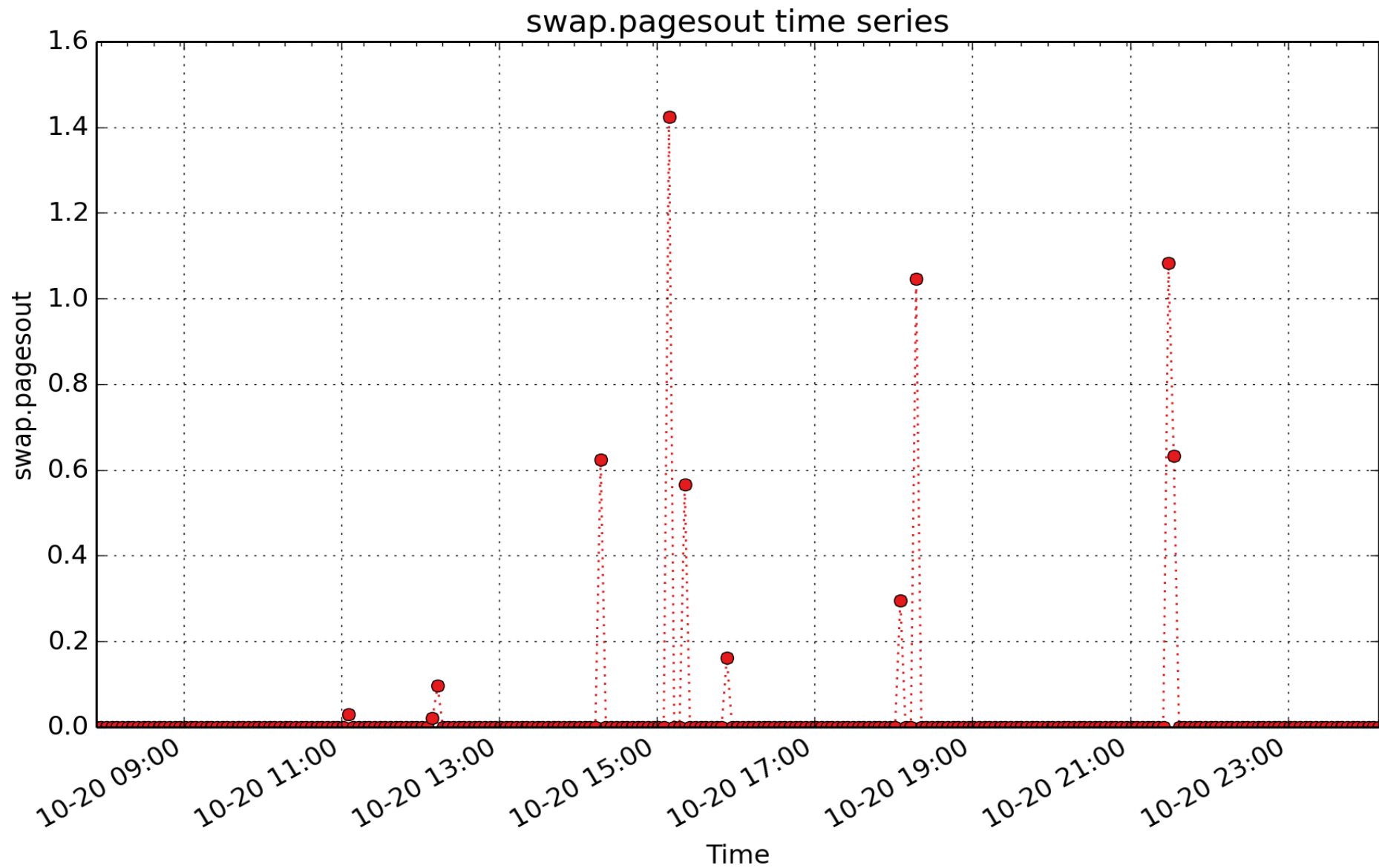
Swap



swap.free: swap free metric from /proc/meminfo (byte - U64)

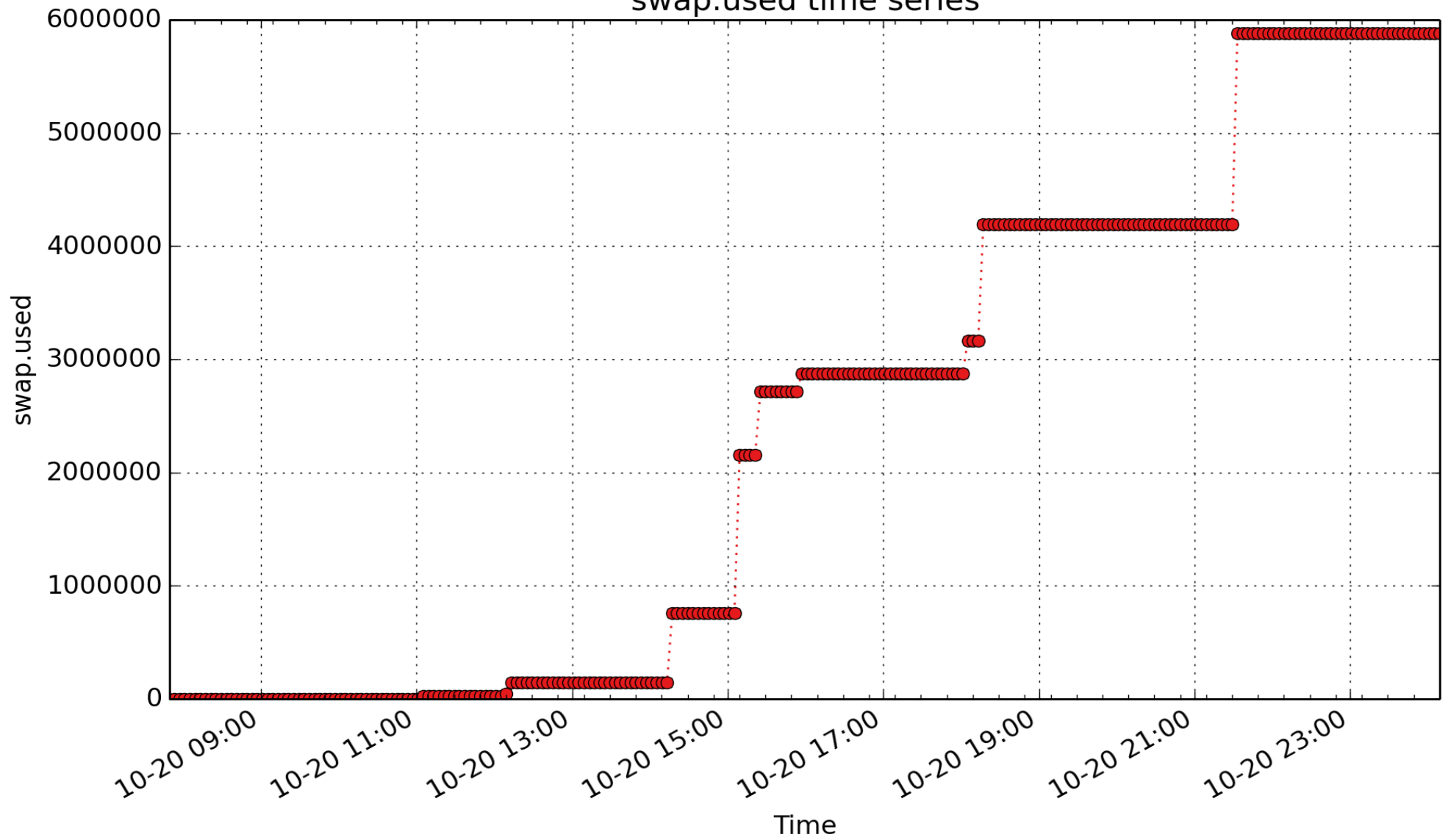


swap.pagesin: pages read from swap devices due to demand for physical memory (count - U32) - *rate converted*



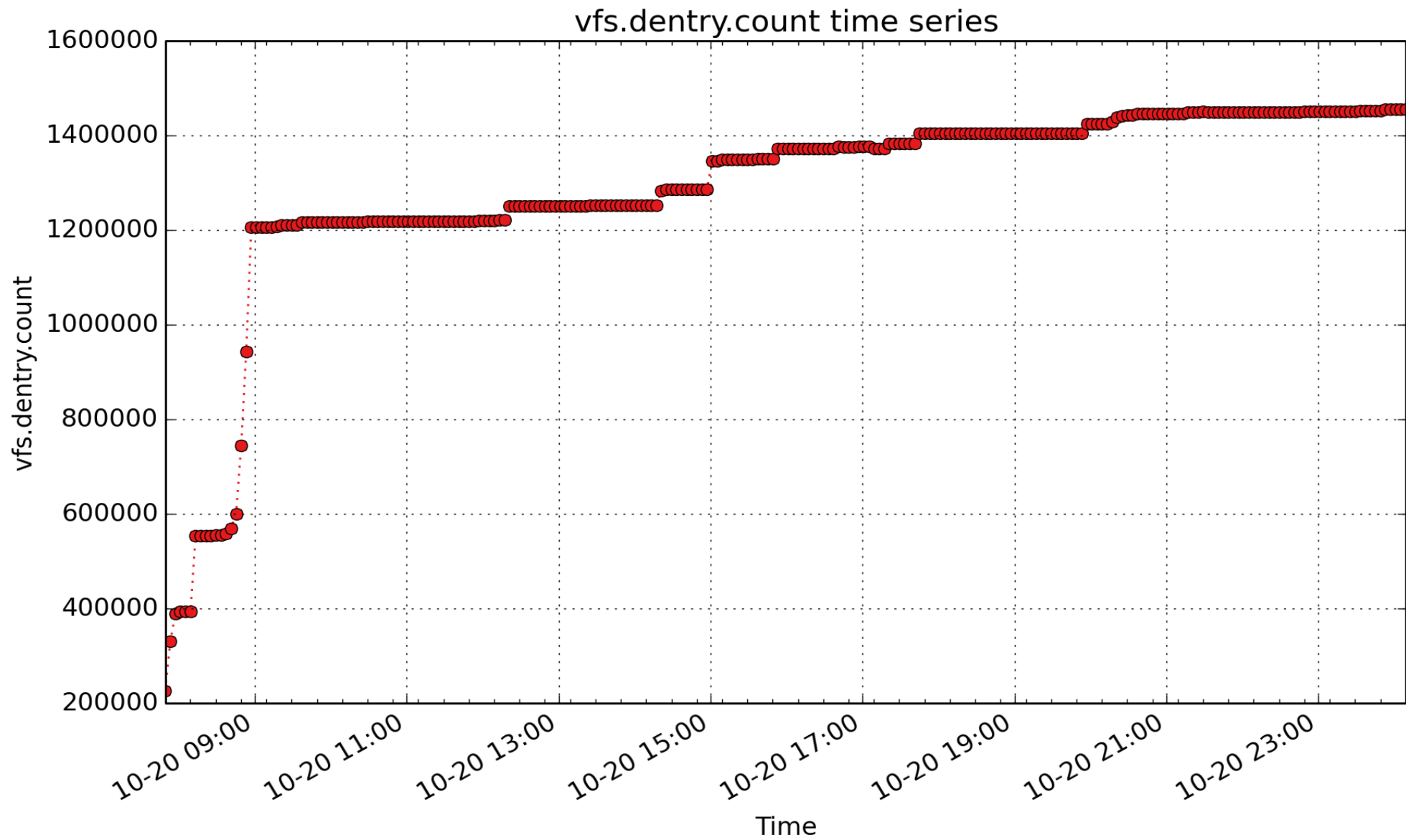
swap.pagesout: pages written to swap devices due to demand for physical memory (count - U32) - *rate converted*

swap.used time series

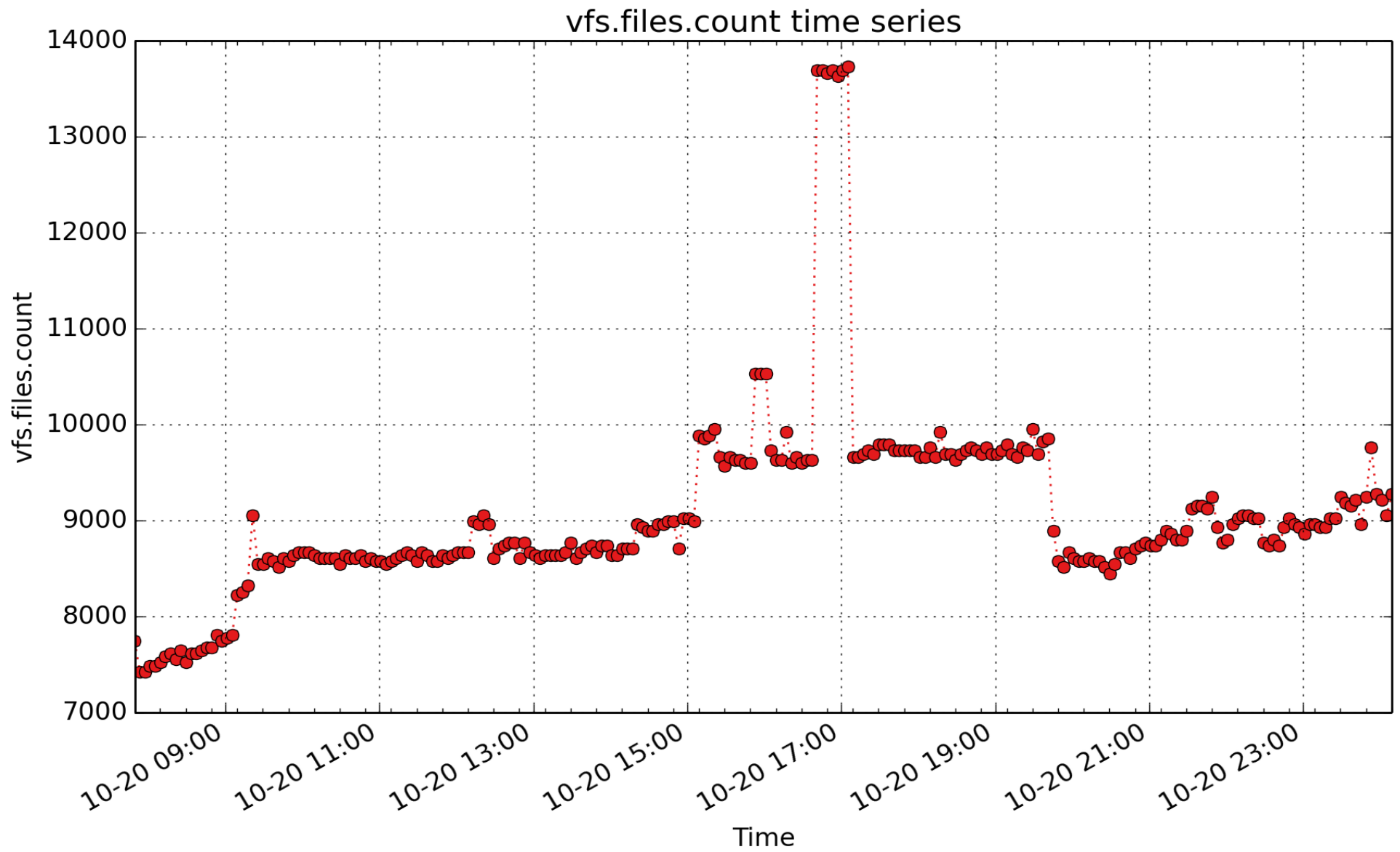


swap.used: swap used metric from /proc/meminfo (byte - U64)

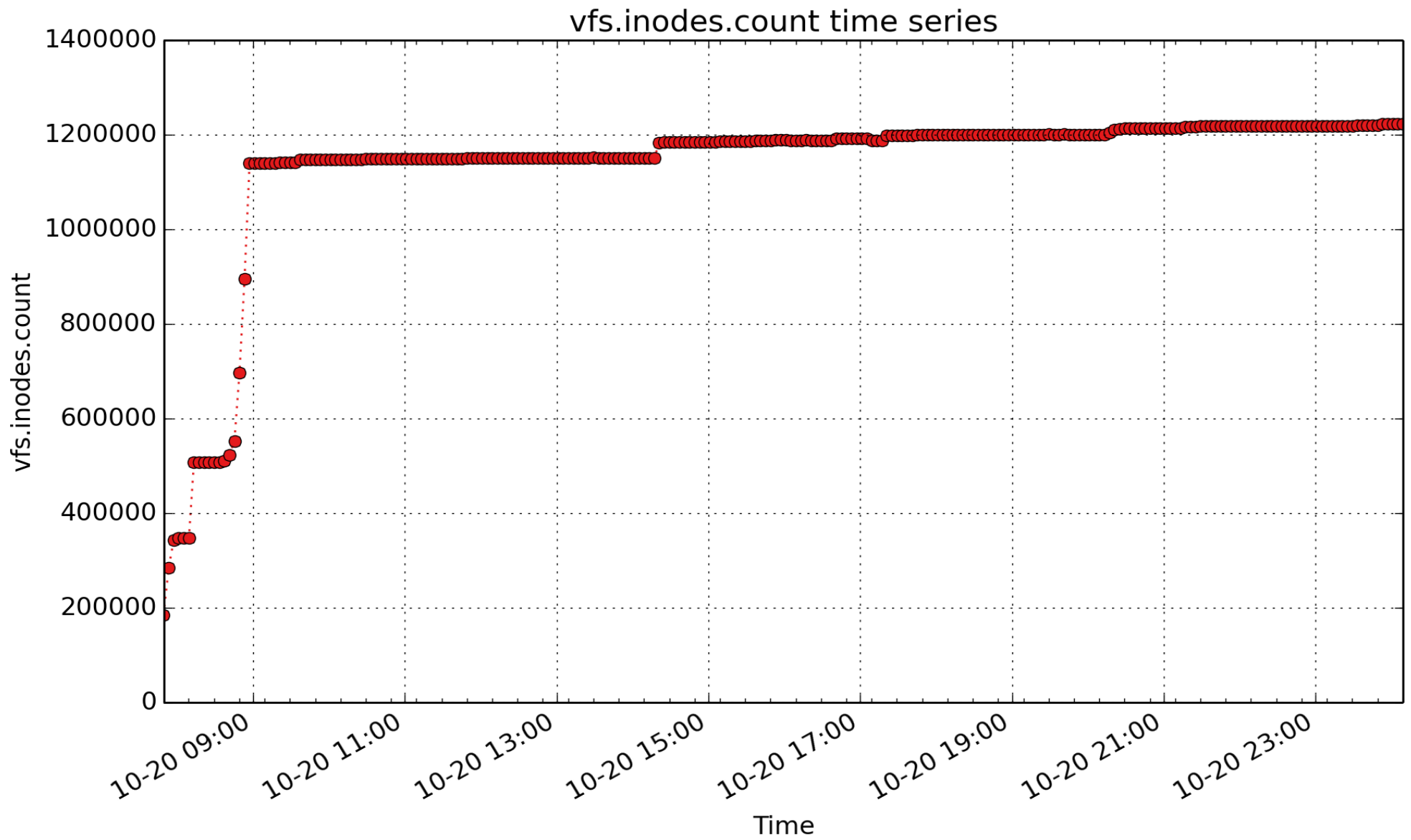
Vfs



vfs.dentry.count: number of in-use dentry structures (- 32)



vfs.files.count: number of in-use file structures (- 32)



vfs.inodes.count: number of in-use inode structures (- 32)