ctrl-1-0.bgp.ftw

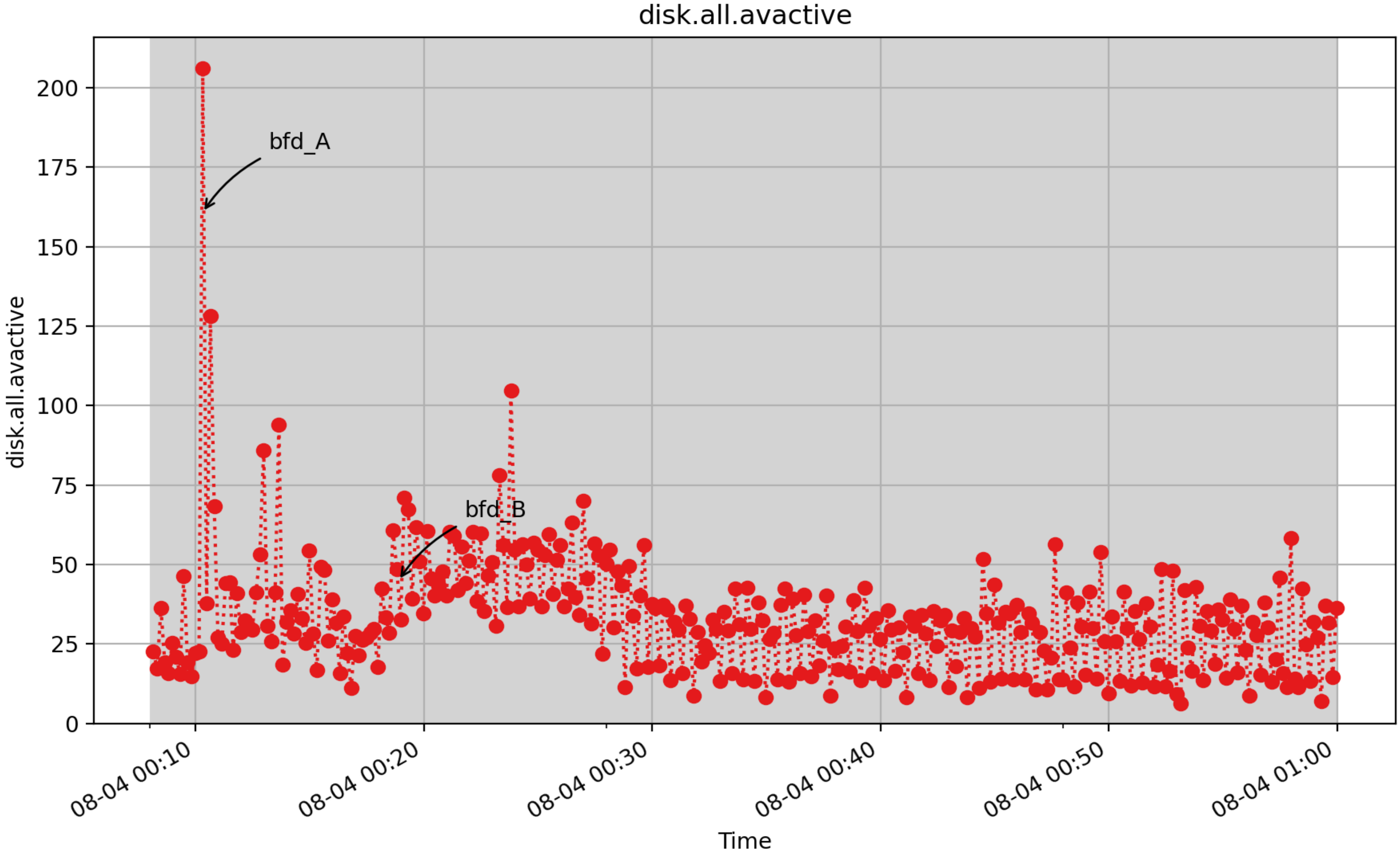
PCP Archive	shortened-ctrl-1-0
Start	2021-08-04 00:07:58
End	2021-08-04 00:59:58
Created	2021-08-04 19:10:08

String Metrics

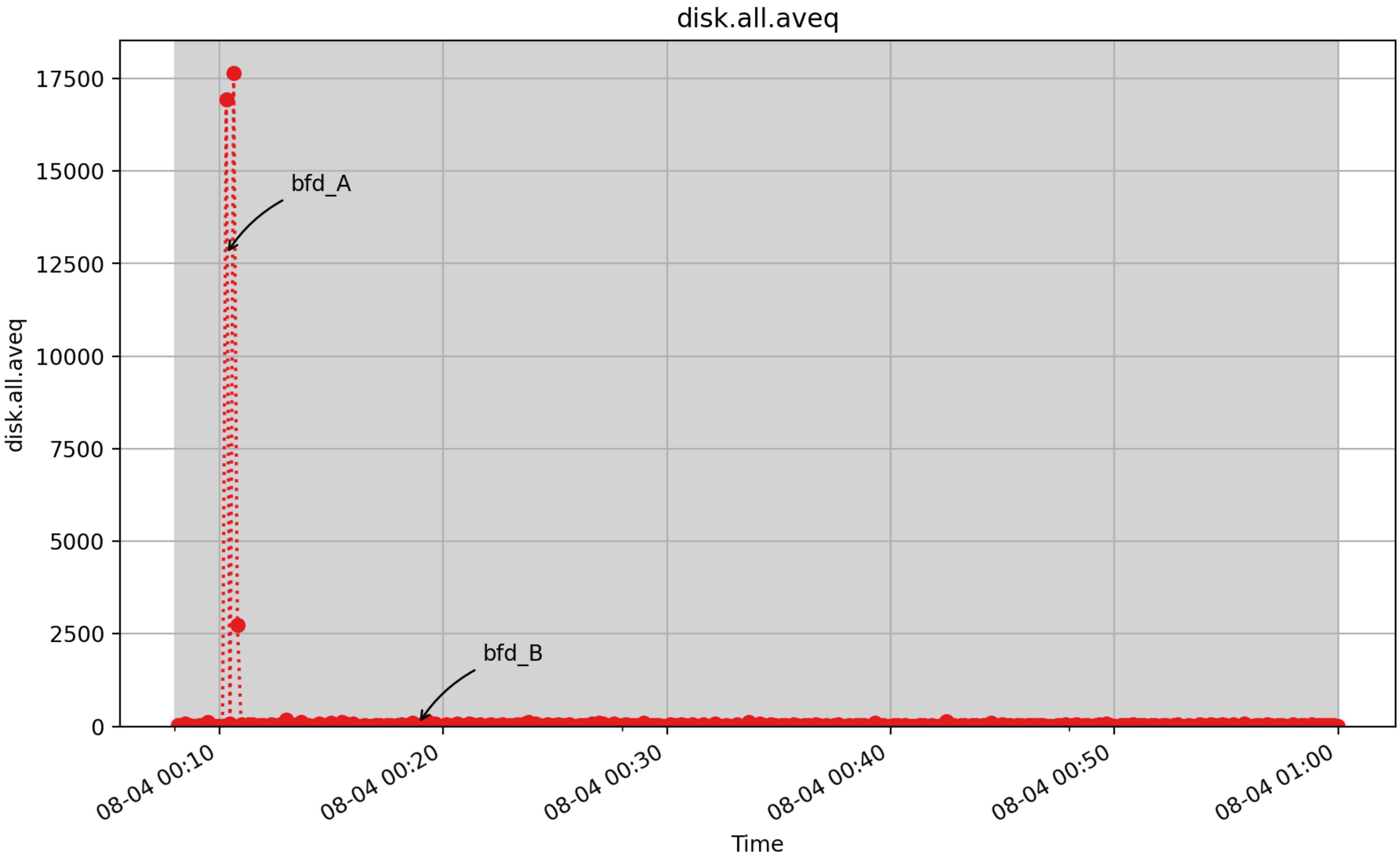
Metric	Timestamp	Value
disk.dev.scheduler	2021-08-04 00:07:58	mq-deadline
filesys.mountdir	2021-08-04 00:07:58	/
hinv.cpu.flags	2021-08-04 00:07:58	b'fpu vme de pse tsc msr pae mce cx8 apic sep mtrr '..b'cid xsaveopt arat umip md_clear arch_capabilities'
hinv.cpu.model	2021-08-04 00:07:58	63
hinv.cpu.model_name	2021-08-04 00:07:58	Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz
hinv.cpu.stepping	2021-08-04 00:07:58	2
hinv.cpu.vendor	2021-08-04 00:07:58	GenuineIntel
hinv.machine	2021-08-04 00:07:58	x86_64
kernel.uname.distro	2021-08-04 00:07:58	CentOS Stream release 8
kernel.uname.machine	2021-08-04 00:07:58	x86_64
kernel.uname.nodename	2021-08-04 00:07:58	ctrl-1-0.bgp.ftw
kernel.uname.release	2021-08-04 00:07:58	4.18.0-326.el8.x86_64
kernel.uname.sysname	2021-08-04 00:07:58	Linux
kernel.uname.version	2021-08-04 00:07:58	#1 SMP Wed Jul 28 21:21:05 UTC 2021
network.interface.hw_addr	2021-08-04 00:07:58	00:00:00:00:00:00
network.interface.hw_addr	2021-08-04 00:07:58	52:54:00:76:c7:07
network.interface.hw_addr	2021-08-04 00:07:58	52:54:00:98:49:3c
network.interface.hw_addr	2021-08-04 00:07:58	52:54:00:fa:bd:39
network.interface.hw_addr	2021-08-04 00:07:58	36:2d:3f:69:d5:4e
network.interface.hw_addr	2021-08-04 00:07:58	36:d7:a1:65:0f:4b
network.interface.hw_addr	2021-08-04 00:07:58	1e:75:e6:8a:d7:16
network.interface.hw_addr	2021-08-04 00:07:58	52:54:00:98:e2:94
network.interface.hw_addr	2021-08-04 00:07:58	12:7a:15:32:99:e3
network.interface.hw_addr	2021-08-04 00:07:58	fa:13:f4:4d:9e:e9
network.interface.inet_addr	2021-08-04 00:07:58	127.0.0.1
network.interface.inet_addr	2021-08-04 00:07:58	192.168.1.161
network.interface.inet_addr	2021-08-04 00:07:58	100.65.1.2
network.interface.inet_addr	2021-08-04 00:07:58	100.64.0.2

network.interface.inet_addr	2021-08-04 00:07:58	192.168.222.1
network.interface.inet_addr	2021-08-04 00:07:58	192.168.222.2
network.interface.ipv6_addr	2021-08-04 00:07:58	f00d:f00d:f00d:f00d:f00d:f00d:f00d:1/128
network.interface.ipv6_addr	2021-08-04 00:07:58	fe80::5054:ff:fe76:c707/64
network.interface.ipv6_addr	2021-08-04 00:07:58	fe80::5054:ff:fe98:493c/64
network.interface.ipv6_addr	2021-08-04 00:07:58	fe80::5054:ff:fefa:bd39/64
network.interface.ipv6_addr	2021-08-04 00:07:58	fd53:d91e:400:7f17::1/128
network.interface.ipv6_addr	2021-08-04 00:07:58	fe80::34d7:a1ff:fe65:f4b/64
network.interface.ipv6_addr	2021-08-04 00:07:58	fe80::f813:f4ff:fe4d:9ee9/64
network.interface.ipv6_scope	2021-08-04 00:07:58	Global
network.interface.ipv6_scope	2021-08-04 00:07:58	Link
network.interface.ipv6_scope	2021-08-04 00:07:58	Global
network.interface.ipv6_scope	2021-08-04 00:07:58	Link
pmda.uname	2021-08-04 00:07:58	Linux ctrl-1-0.bgp.ftw 4.18.0-326.el8.x86_64 #1 SMP Wed Jul 28 21:21:05 UTC 2021 x86_64

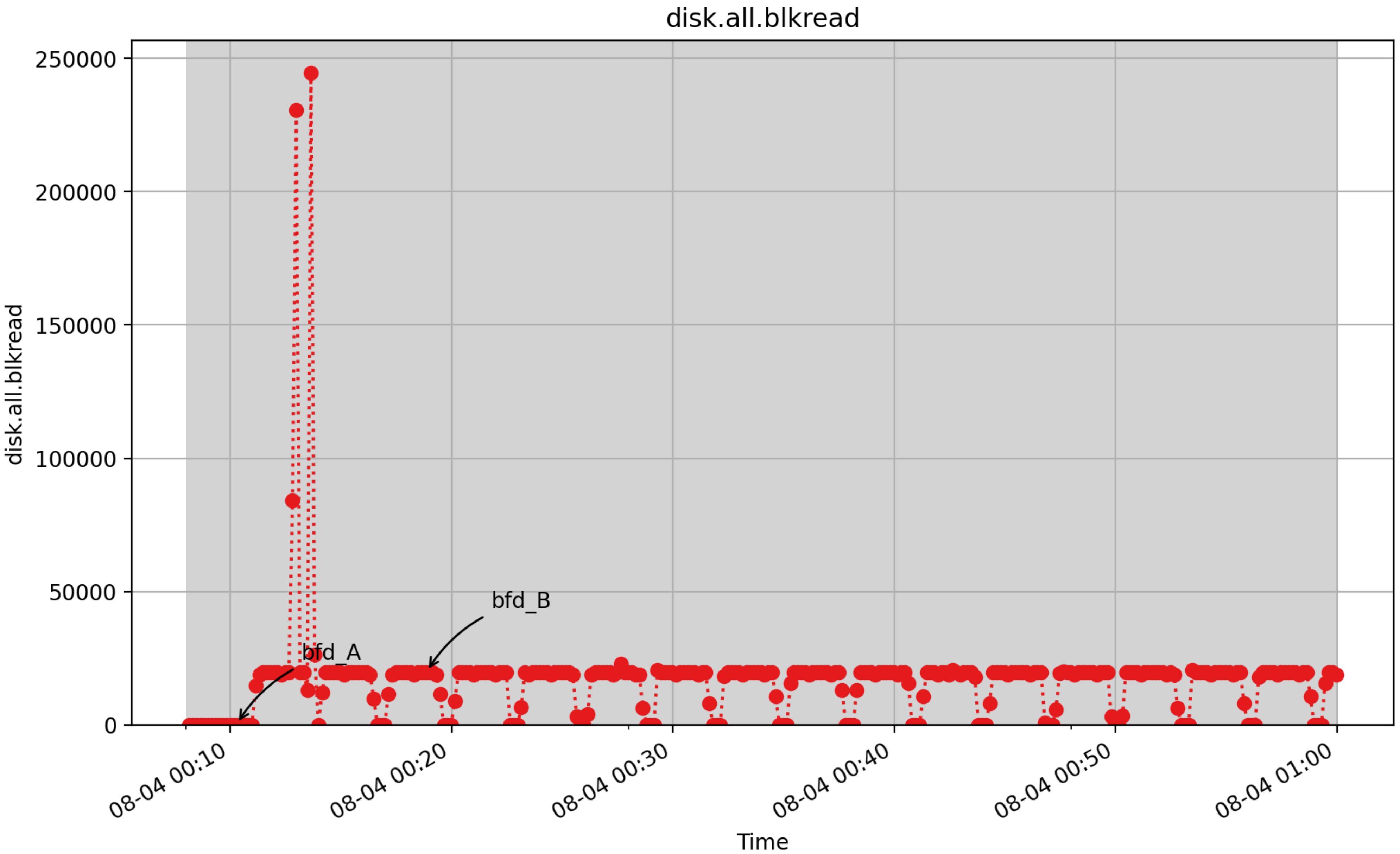
Disk



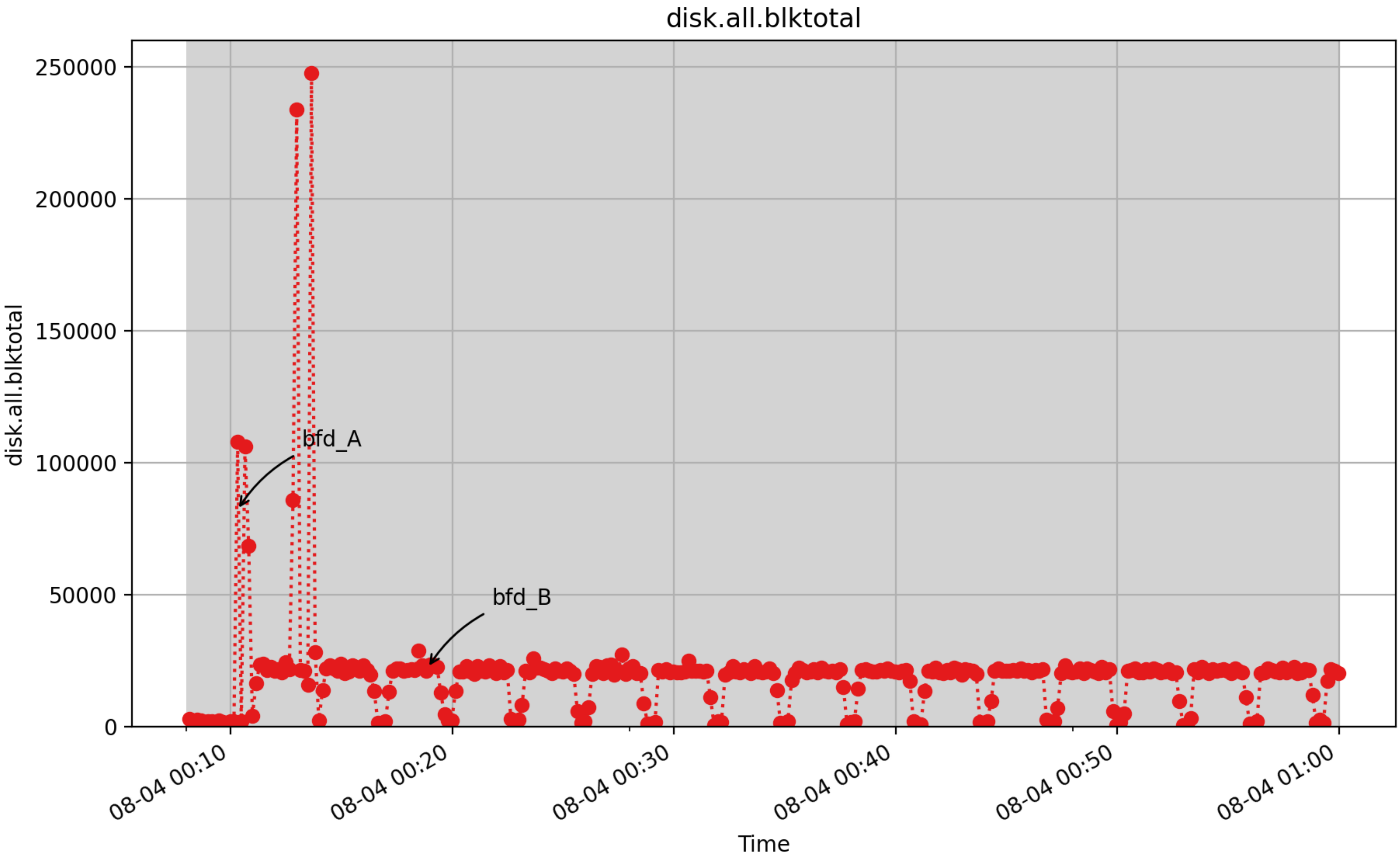
disk.all.avactive: Counts the number of milliseconds for which at least one I/O is in progress on each disk, summed across all disks. When converted to a rate and divided by the number of disks (hinv.ndisk), this metric represents the average utilization of all disks during the sampling interval. A value 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*



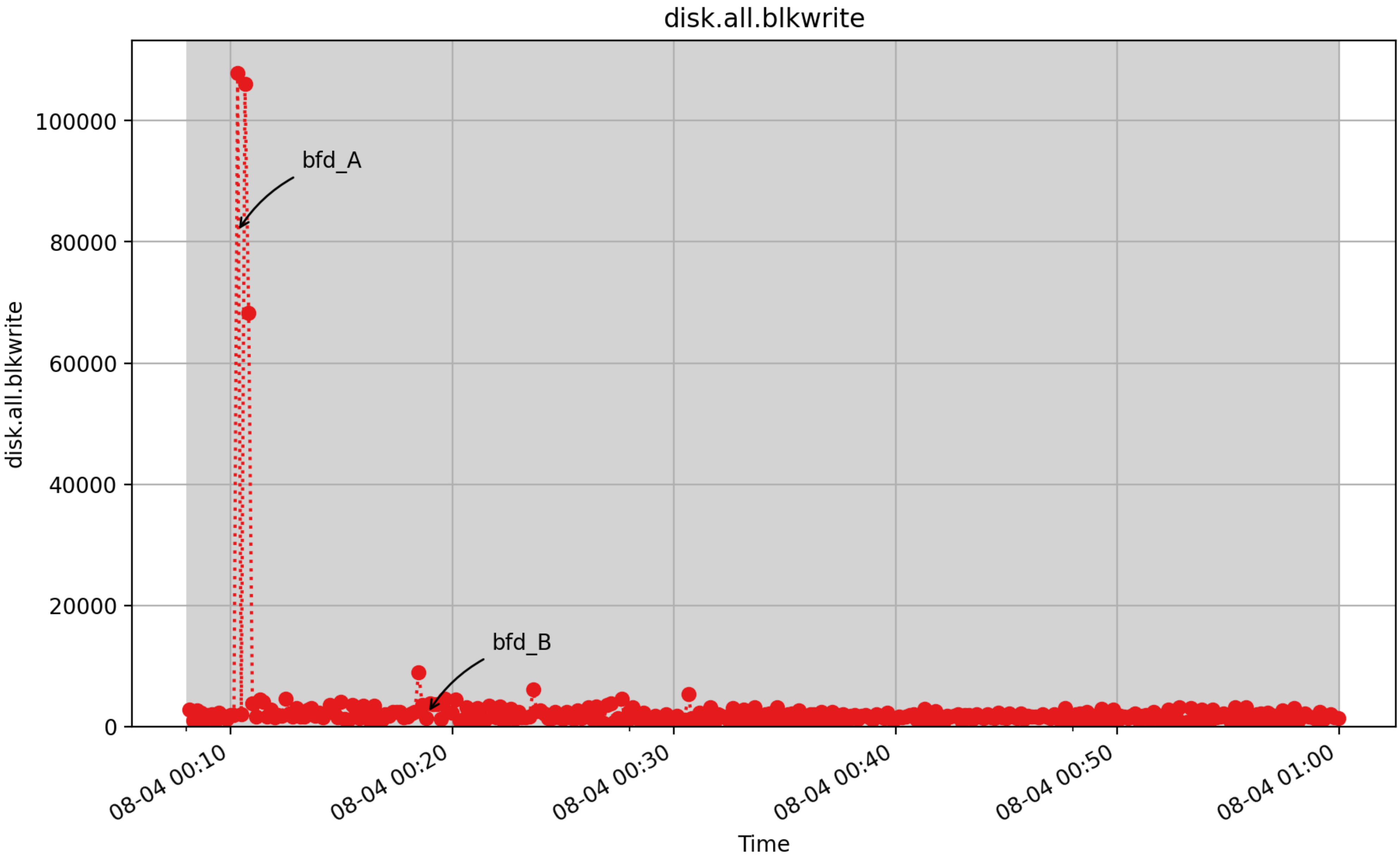
disk.all.aveq: When converted to a rate, this metric represents the average across all disks of the time averaged request queue length during the sampling interval. A value of 1.5 (or 150%) suggests that (on average) over all disks there was a time averaged queue length of 1.5 requests during the sampling interval. (millisec - U64) - *rate converted*



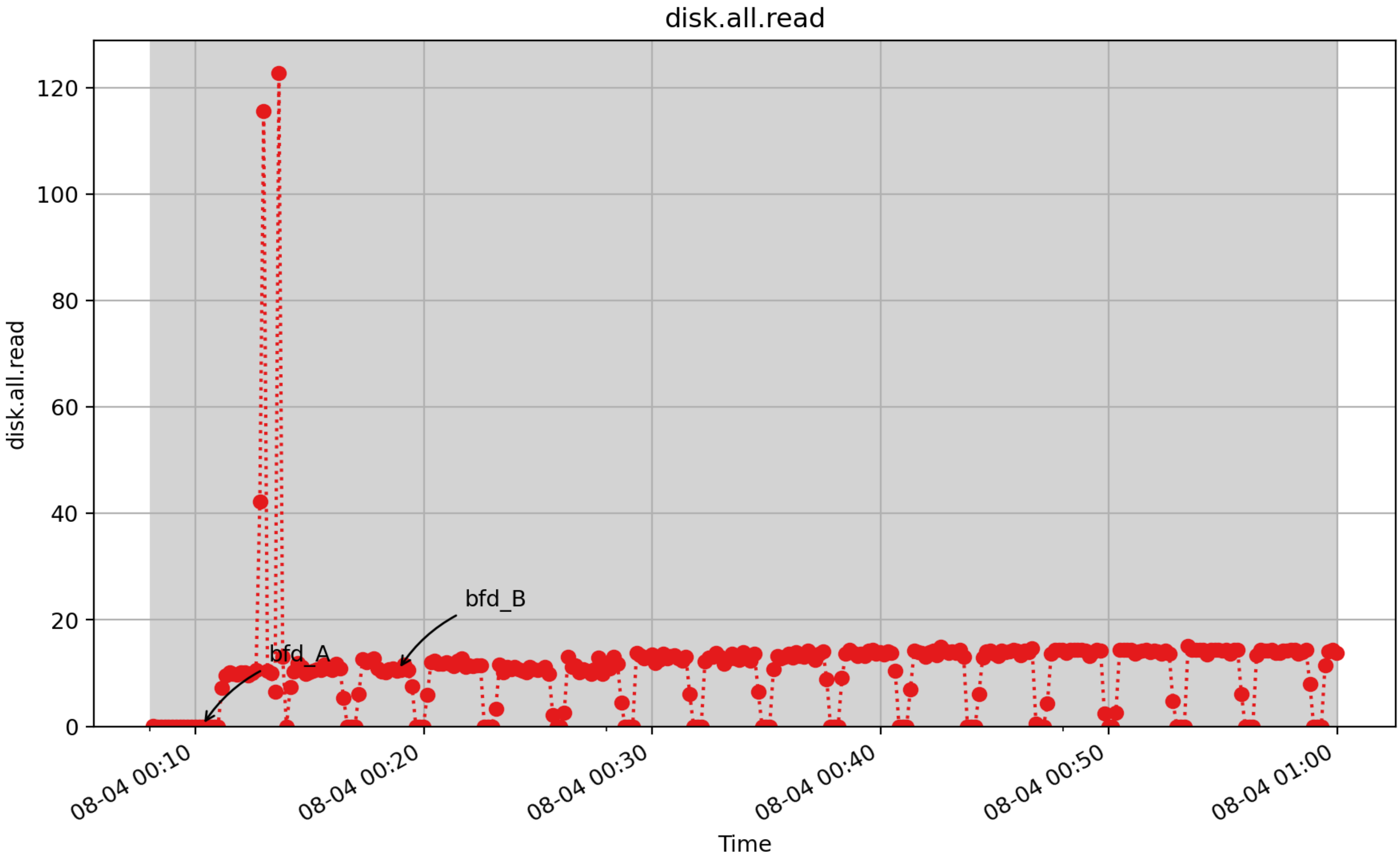
disk.all.blkread: Cumulative number of disk block read operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



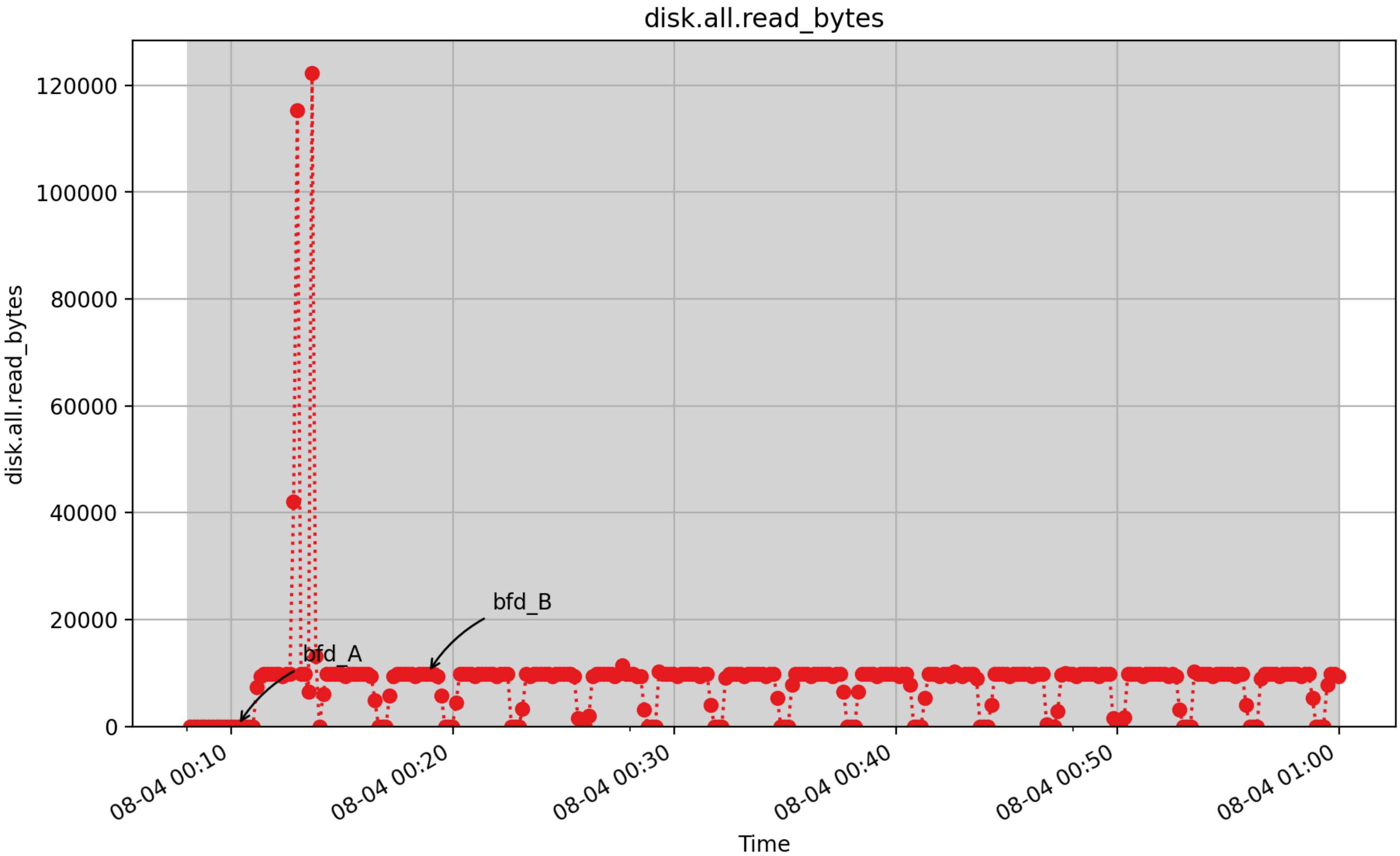
disk.all.blktotal: Cumulative number of disk block read and write operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



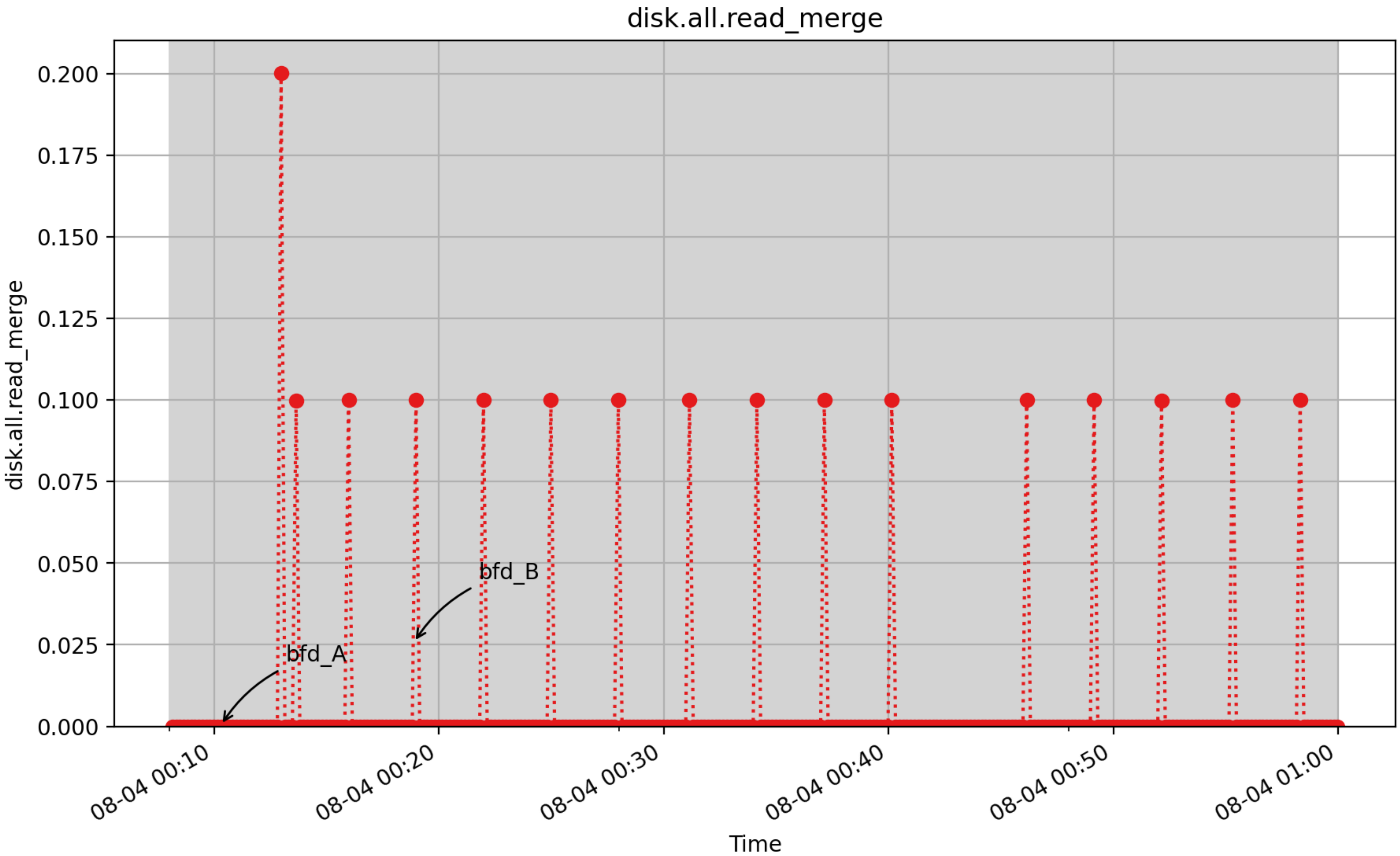
disk.all.blkwrite: Cumulative number of disk block write operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



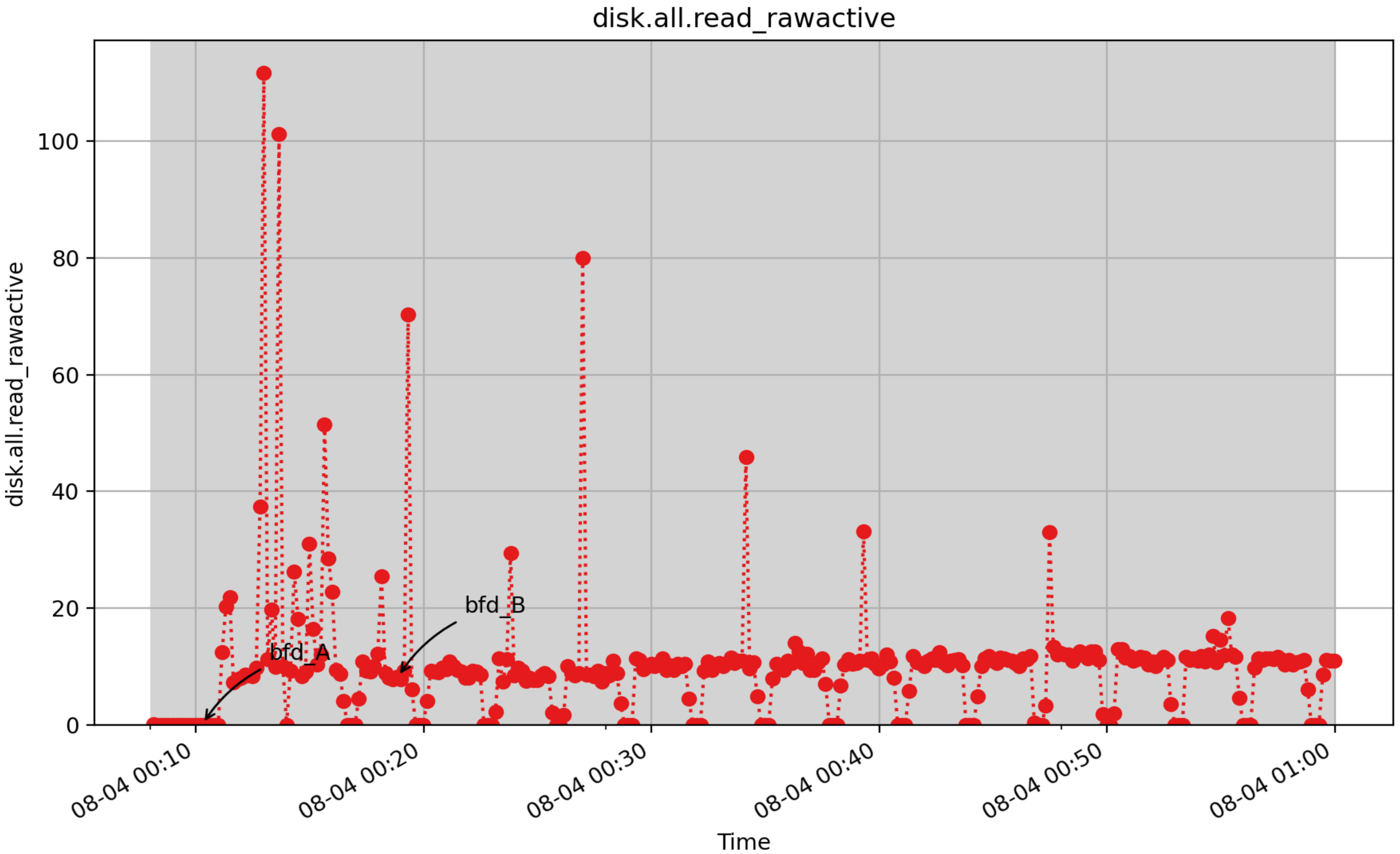
disk.all.read: Cumulative number of disk read operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



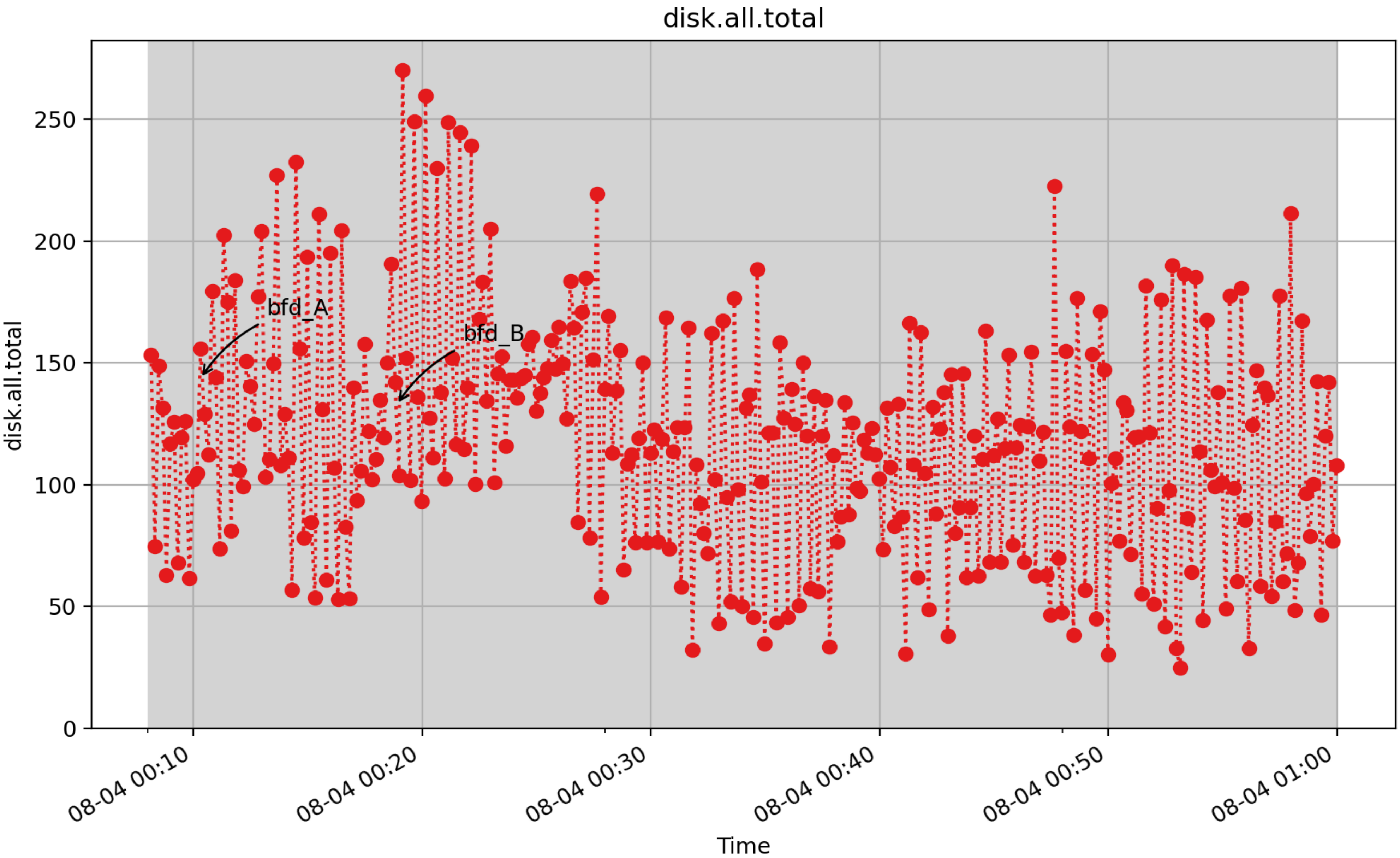
disk.all.read_bytes: count of bytes read for all disk devices (Kbyte - U64) - *rate converted*



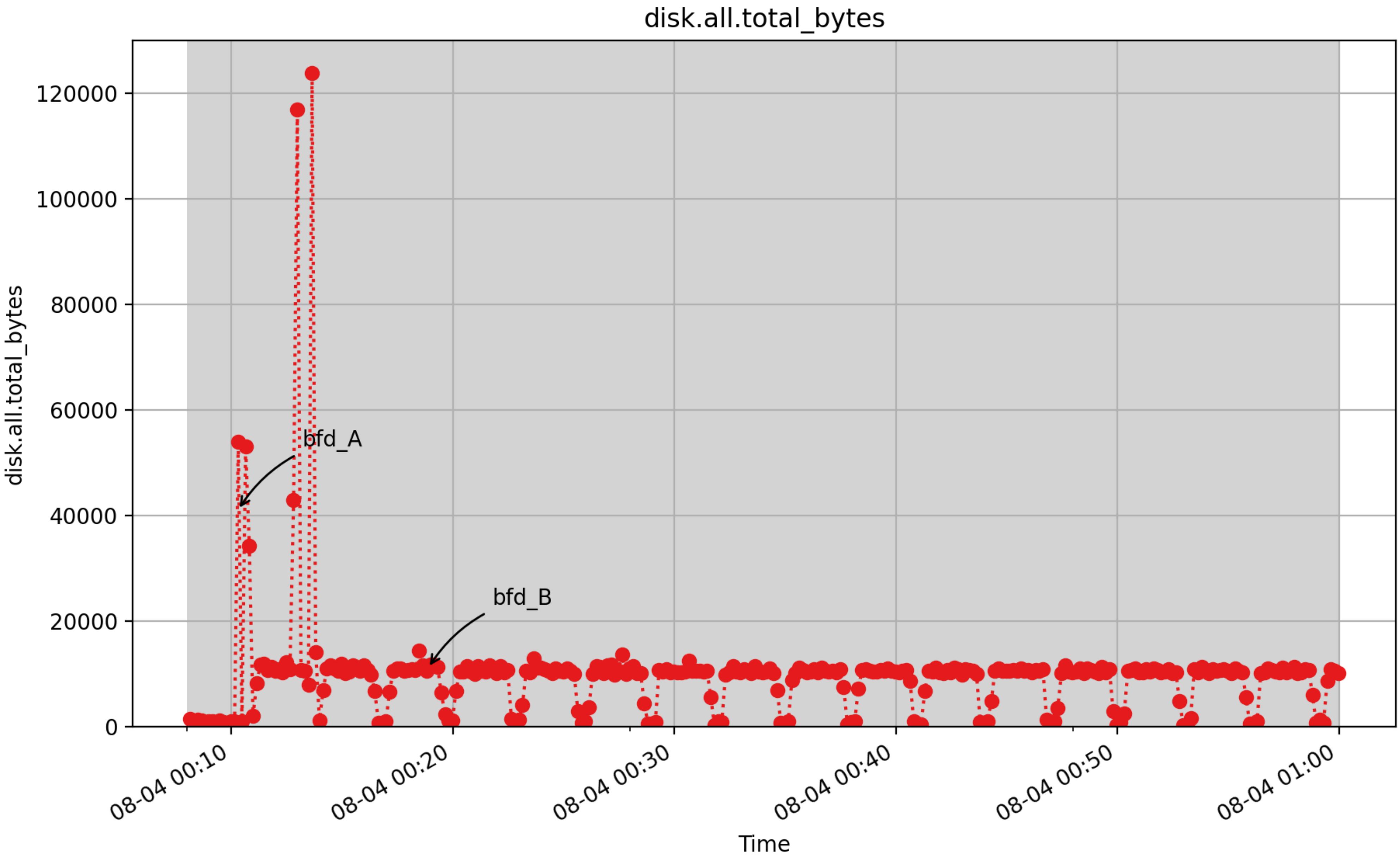
disk.all.read_merge: Total count of read requests that were merged with an already queued read request.
(count - U64) - *rate converted*



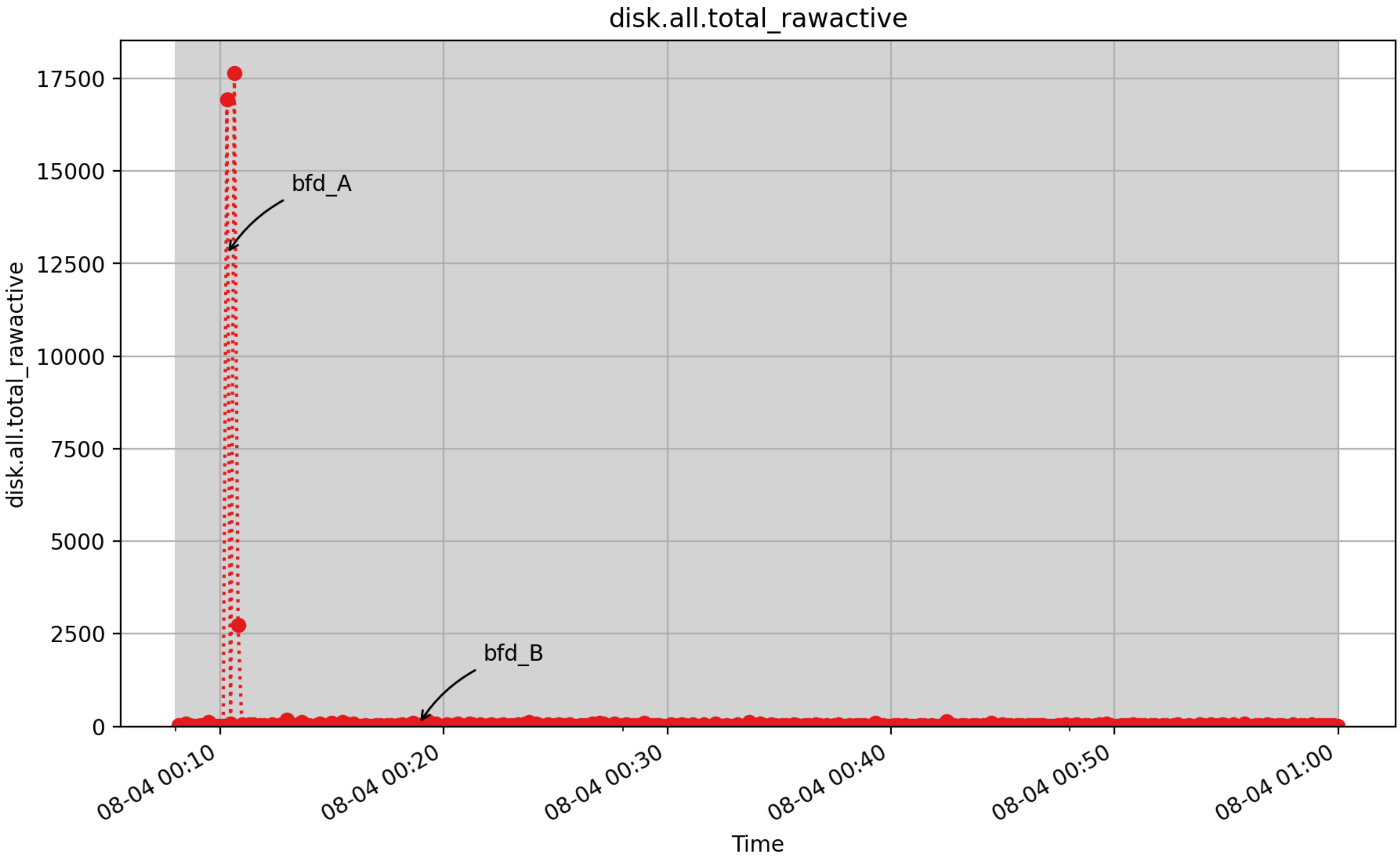
disk.all.read_rawactive: For each completed read on every disk the response time (queue time plus service time) in milliseconds is added to this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding reads across all disks. When divided by the number of completed reads for all disks (disk.all.read), value represents the stochastic average of the read response (or wait) time across all disks. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from existing performance tools: iostat.all.r_await = delta(disk.all.read_rawactive) / delta(disk.all.read) (millisec - U64) - *rate converted*



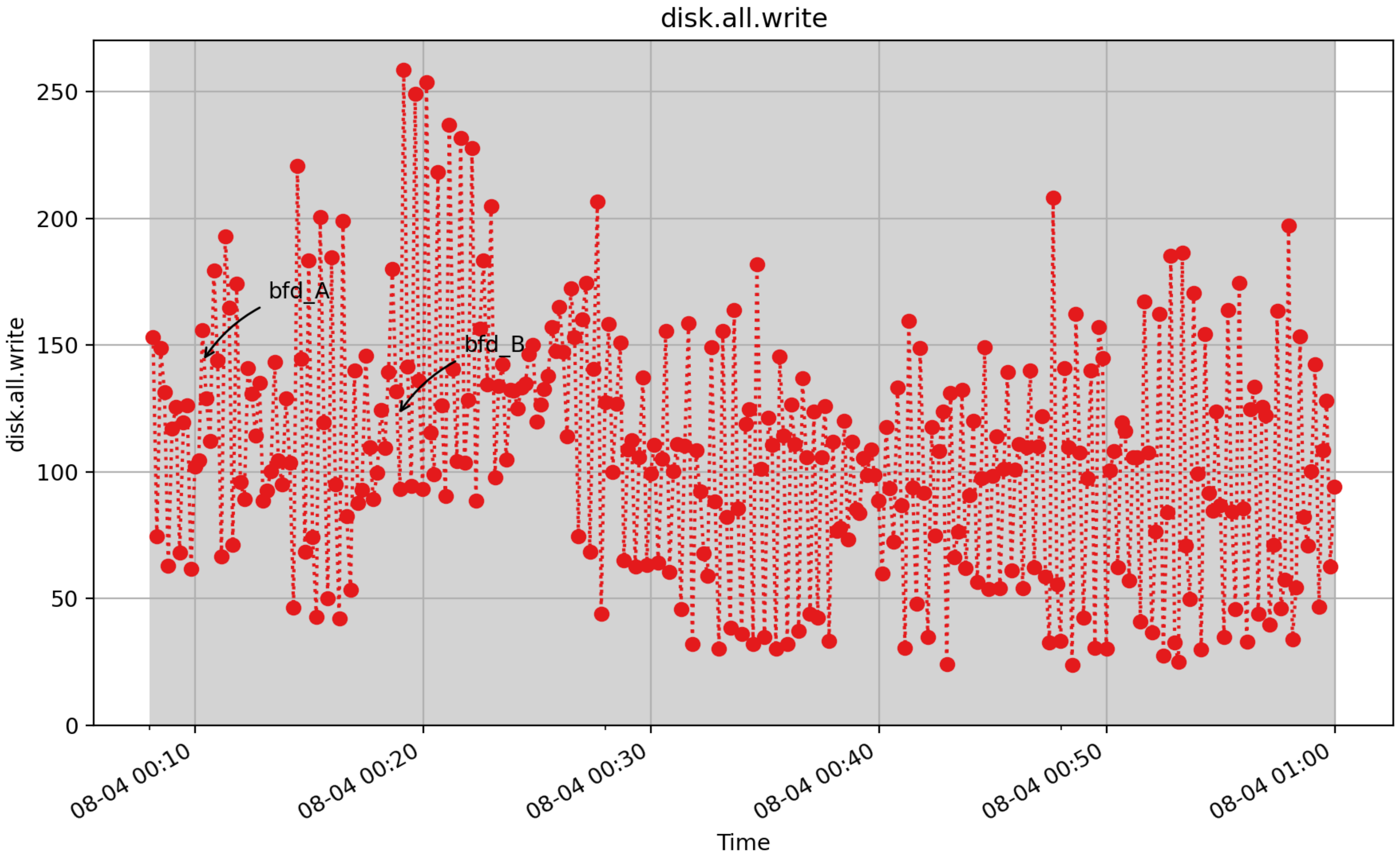
disk.all.total: Cumulative number of disk read and write operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



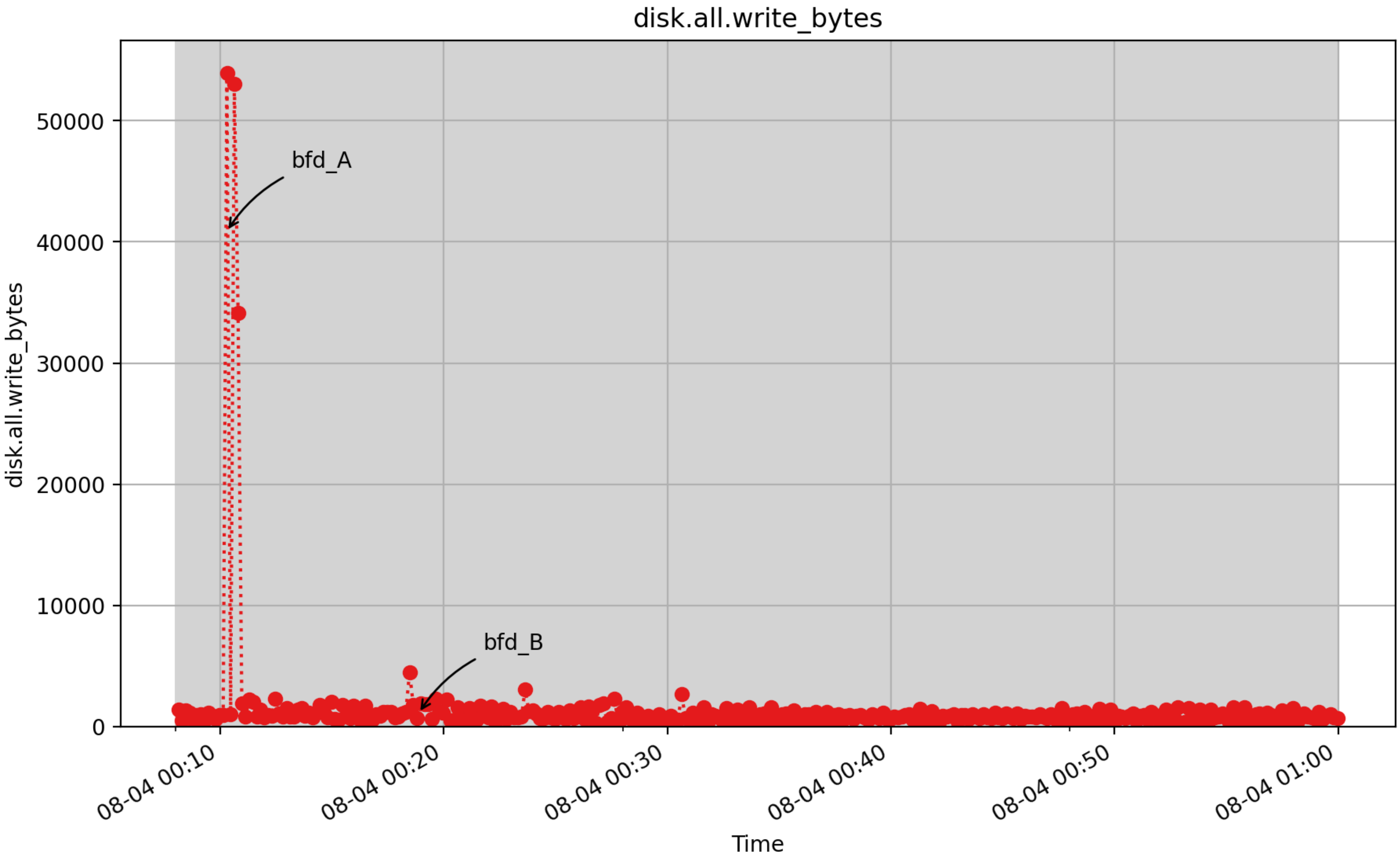
disk.all.total_bytes: total count of bytes read and written for all disk devices (Kbyte - U64) - *rate converted*



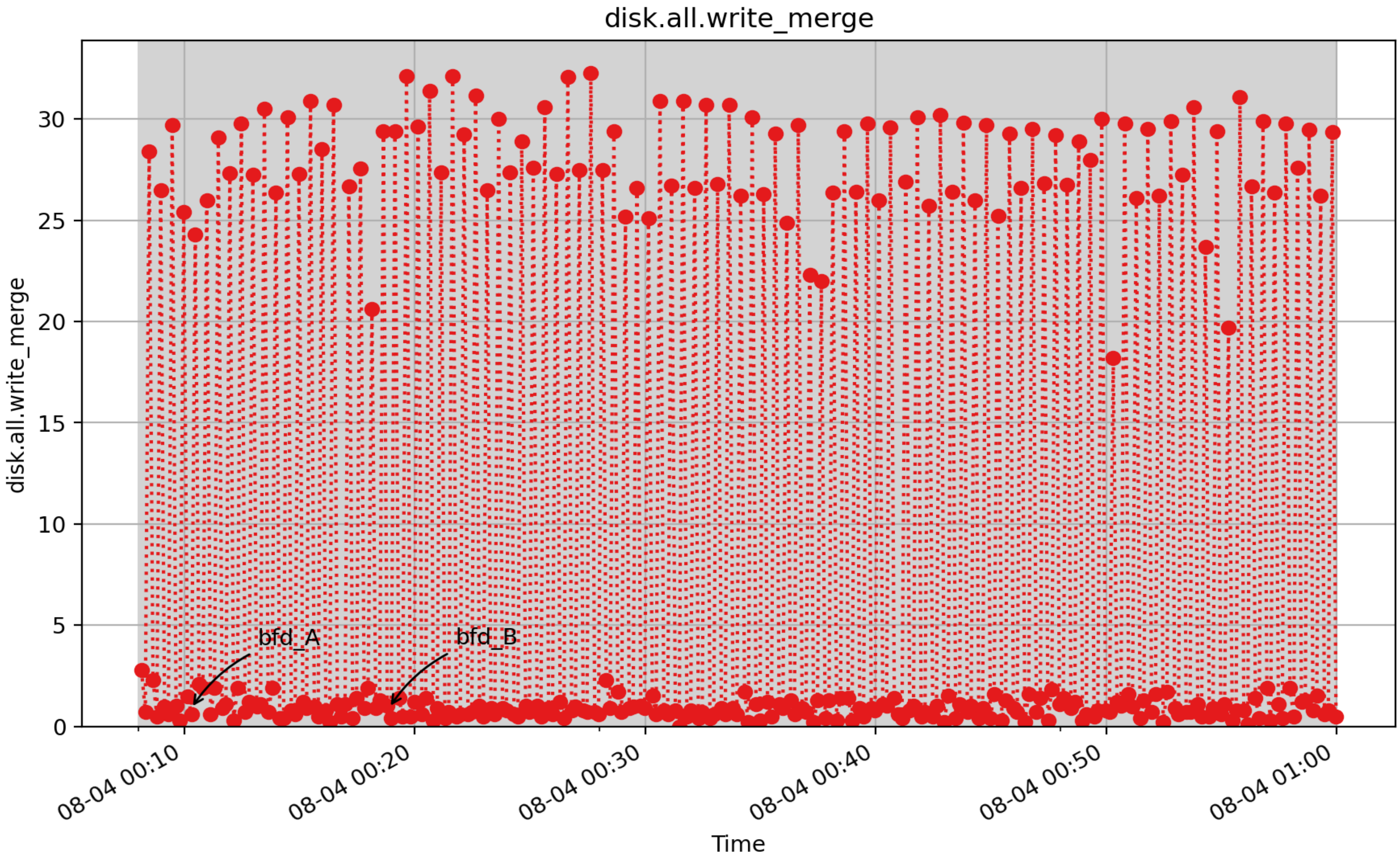
disk.all.total_rawactive: For each completed I/O on every disk the response time (queue time plus service time) in milliseconds is added to this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding I/Os across all disks. When divided by the number of completed I/Os for all disks (disk.all.total), value represents the stochastic average of the I/O response (or wait) time across all disks. (millisec - U64) - *rate converted*



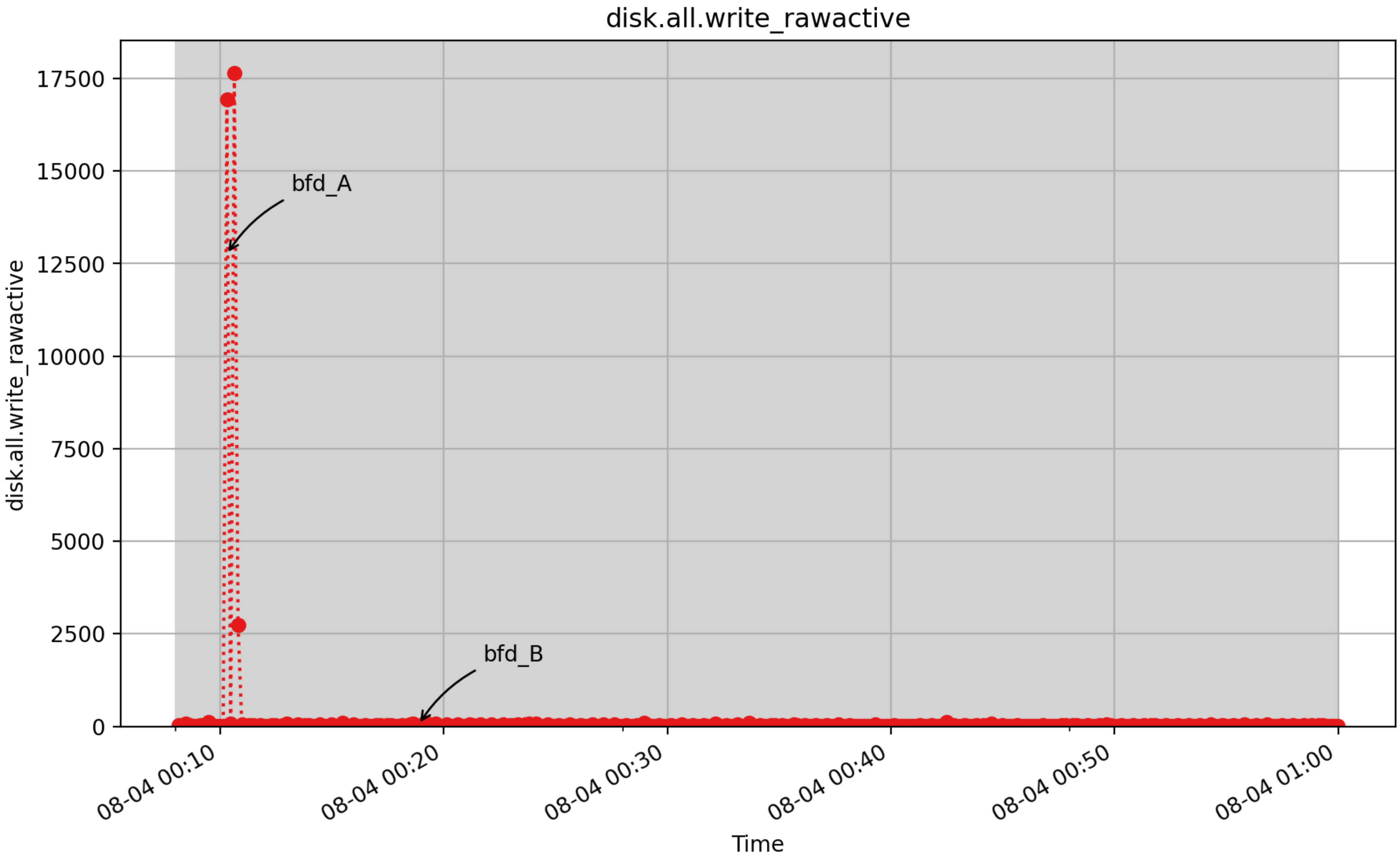
disk.all.write: Cumulative number of disk read operations since system boot time (subject to counter wrap), summed over all disk devices. (count - U64) - *rate converted*



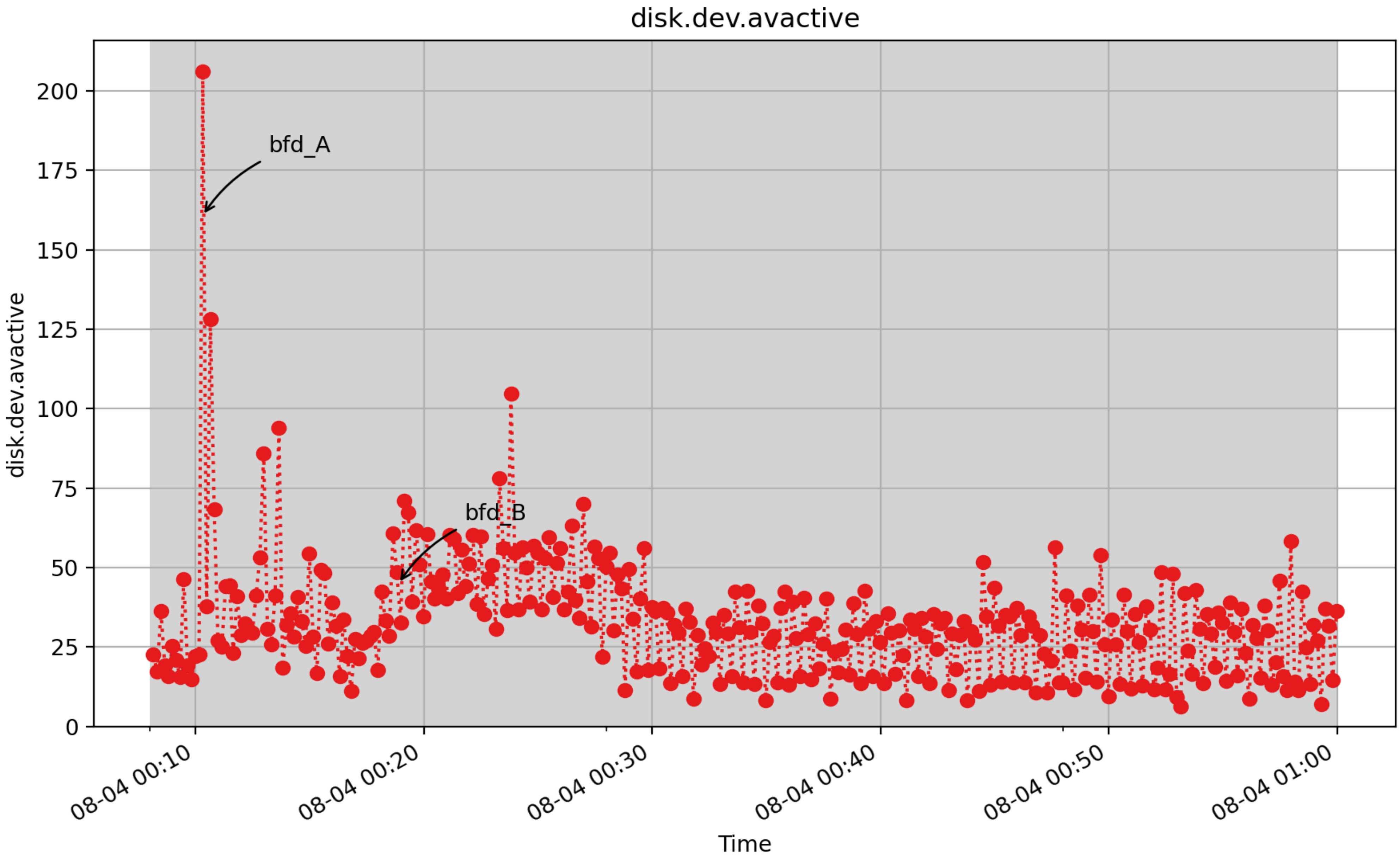
disk.all.write_bytes: count of bytes written for all disk devices (Kbyte - U64) - *rate converted*



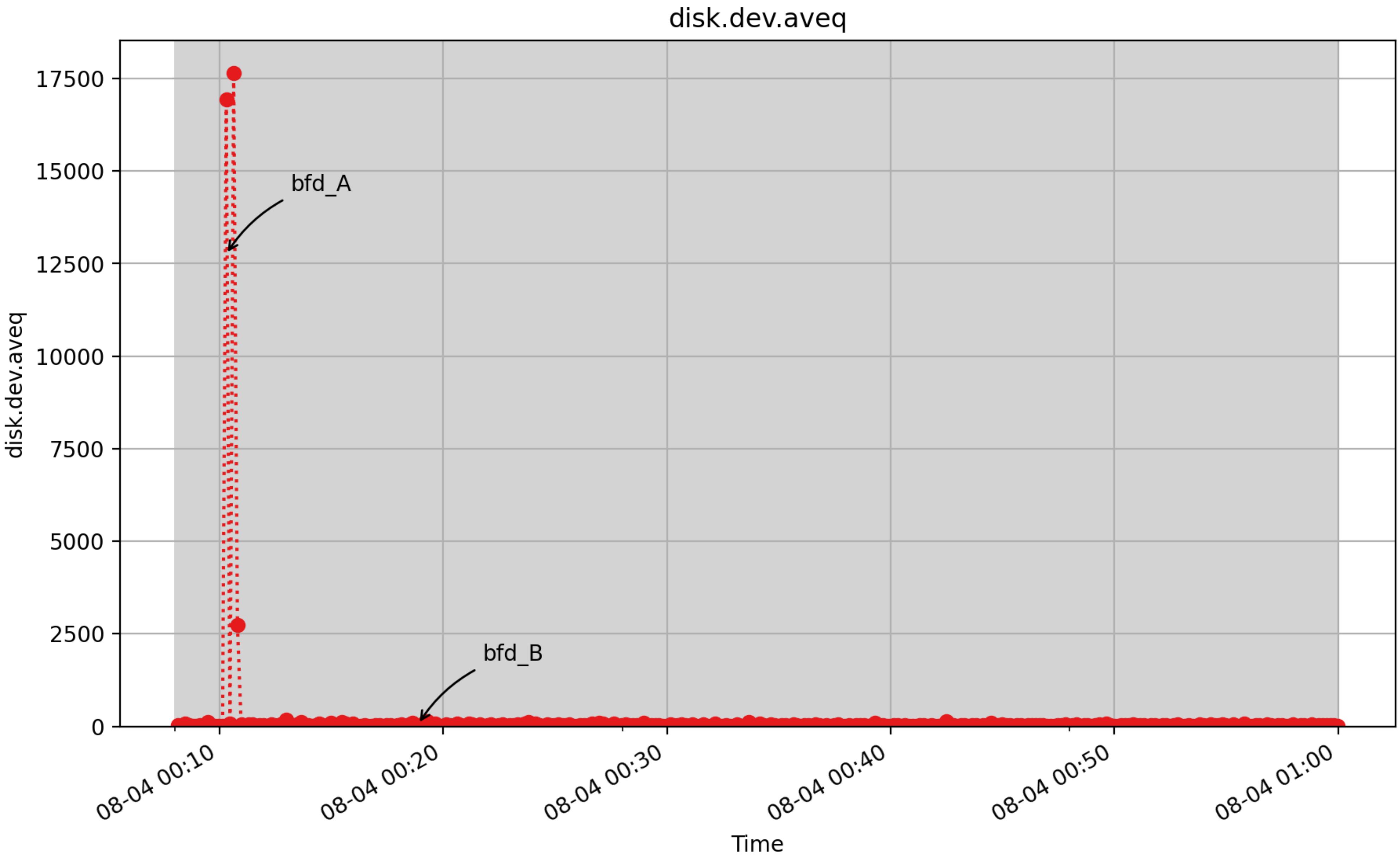
disk.all.write_merge: Total count of write requests that were merged with an already queued write request.
(count - U64) - *rate converted*



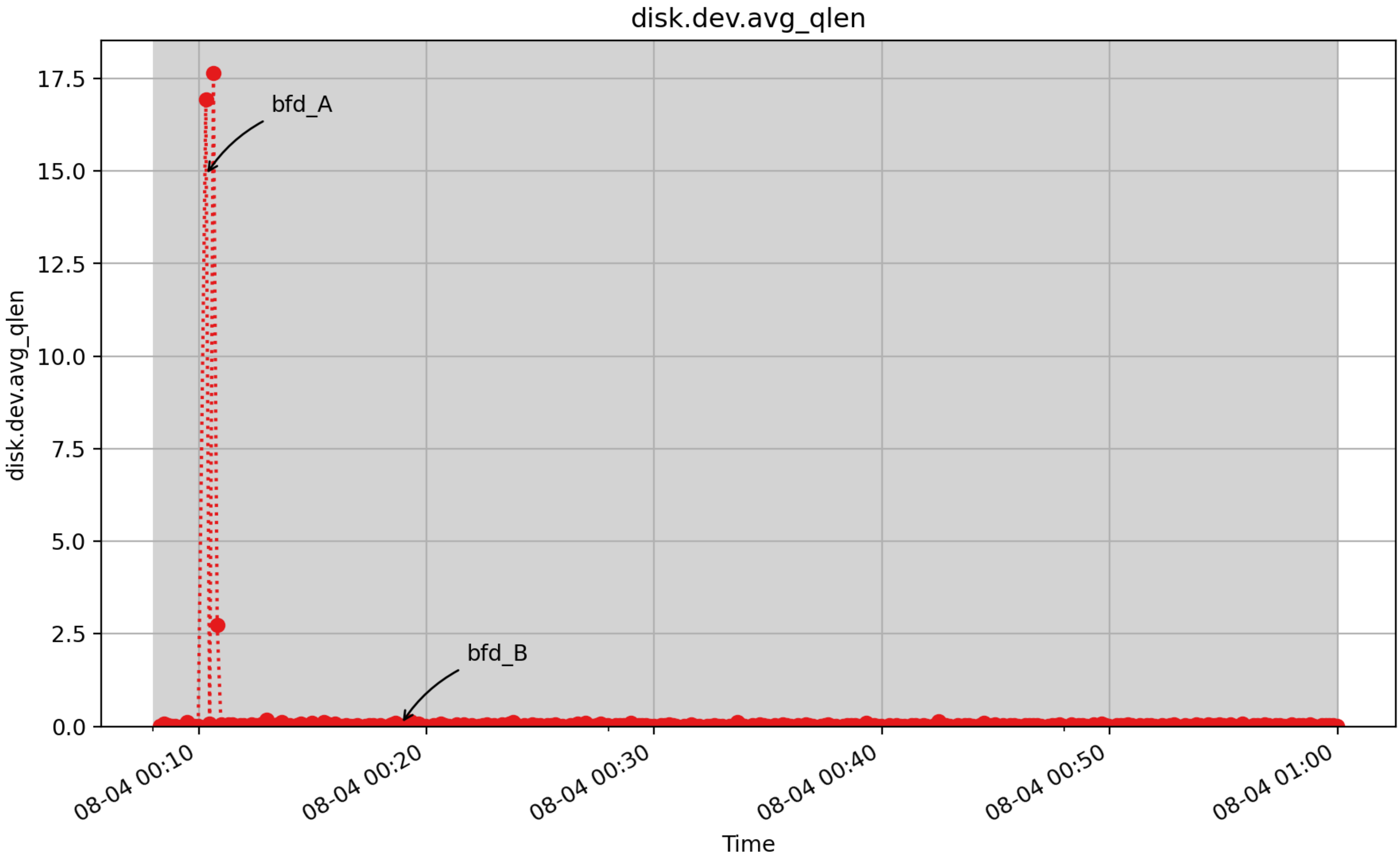
disk.all.write_rawactive: For each completed write on every disk the response time (queue time plus service time) in milliseconds is added to this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding writes across all disks. When divided by the number of completed writes for all disks (disk.all.write), value represents the stochastic average of the write response (or wait) time across all disks. (millisec - U64) - *rate converted*



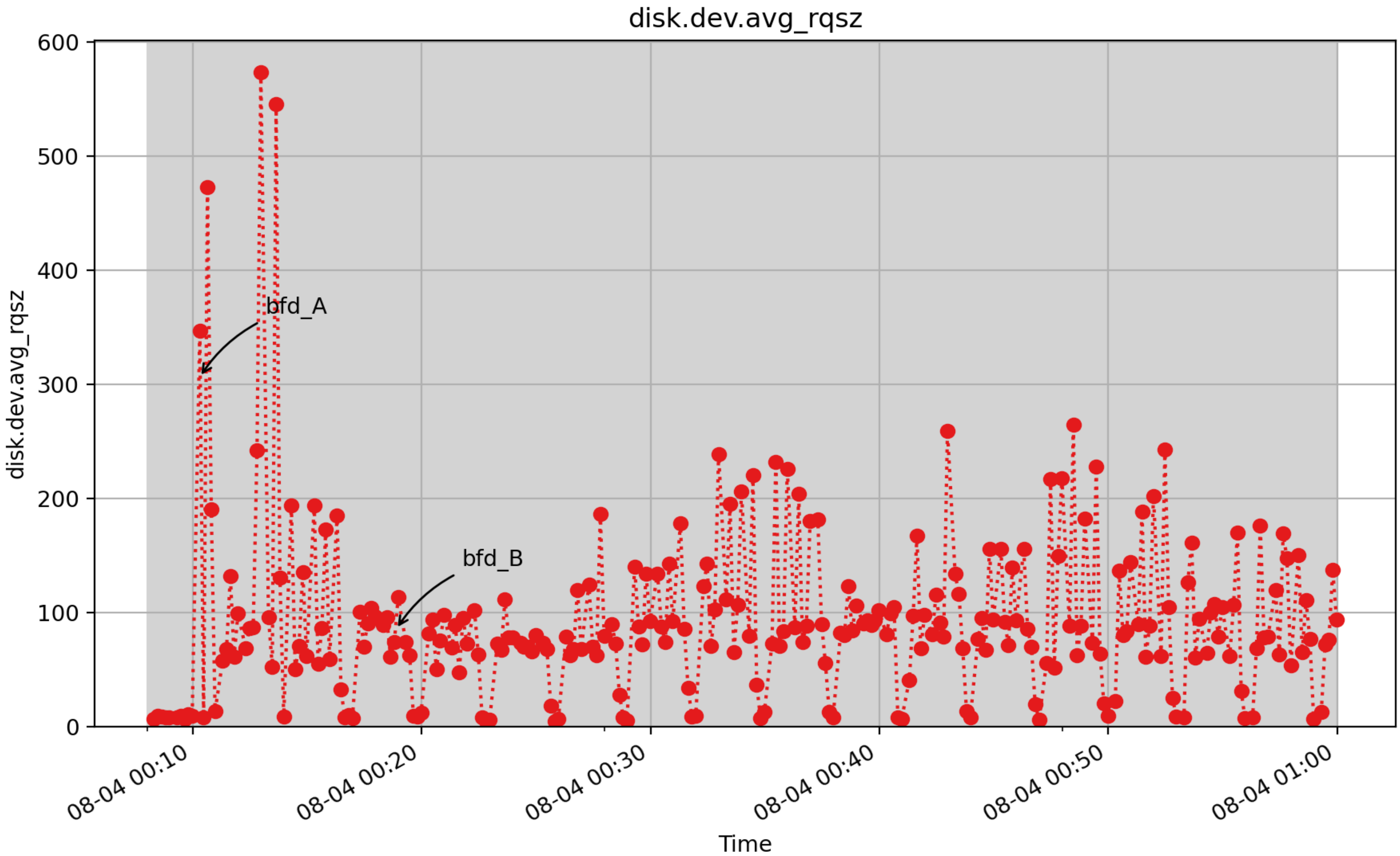
disk.dev.avactive: Counts the number of milliseconds for which at least one I/O is in progress for each device. When converted to a rate, this metric represents the average utilization of the disk during the sampling interval. A value of 0.5 (or 50%) means the disk was active (i.e. busy) half the time. (millisec - U32) - *rate converted*



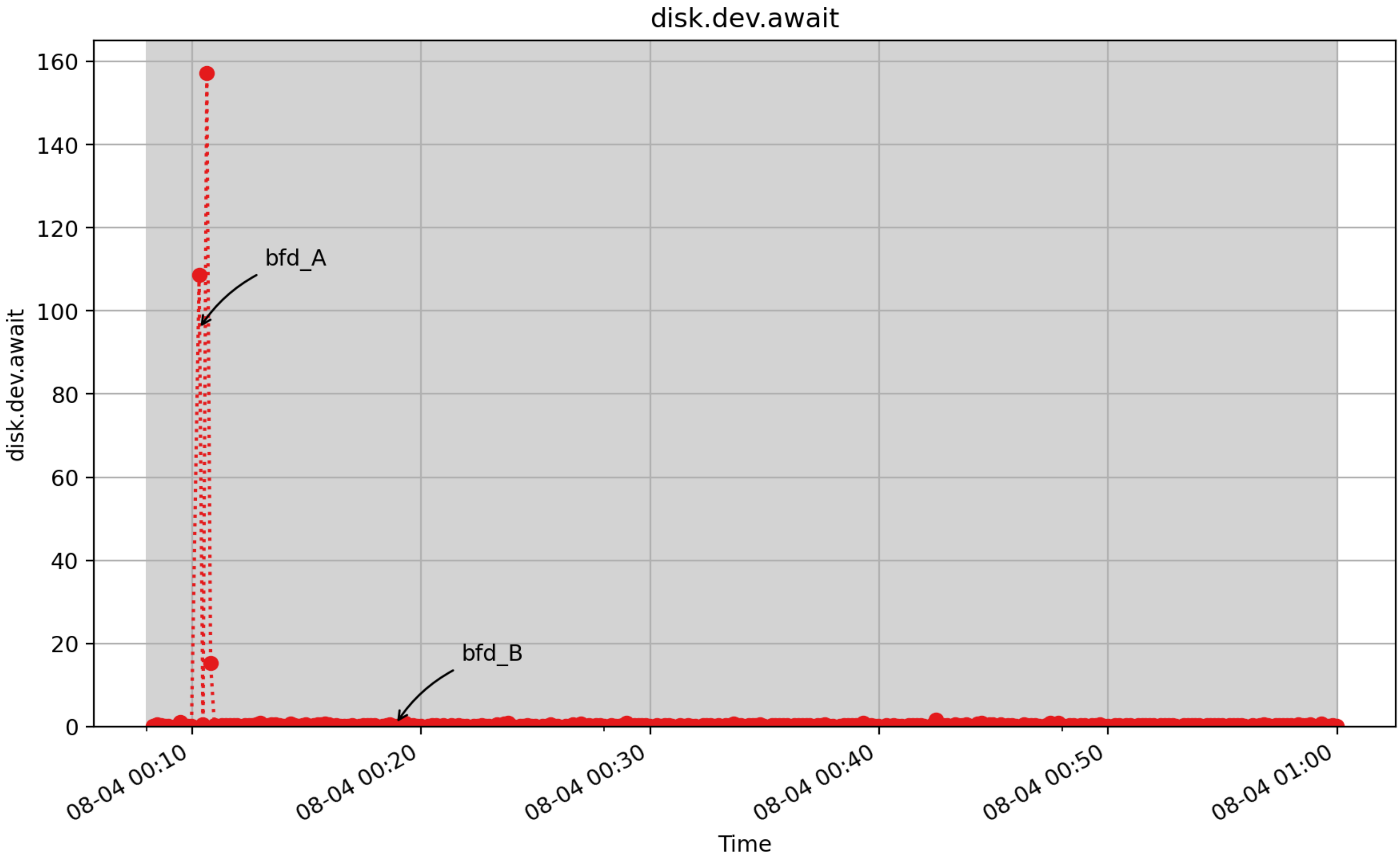
disk.dev.aveq: When converted to a rate, this metric represents the time averaged disk request queue length during the sampling interval. A value of 2.5 (or 250%) represents a time averaged queue length of 2.5 requests during the sampling interval. (millisec - U32) - *rate converted*



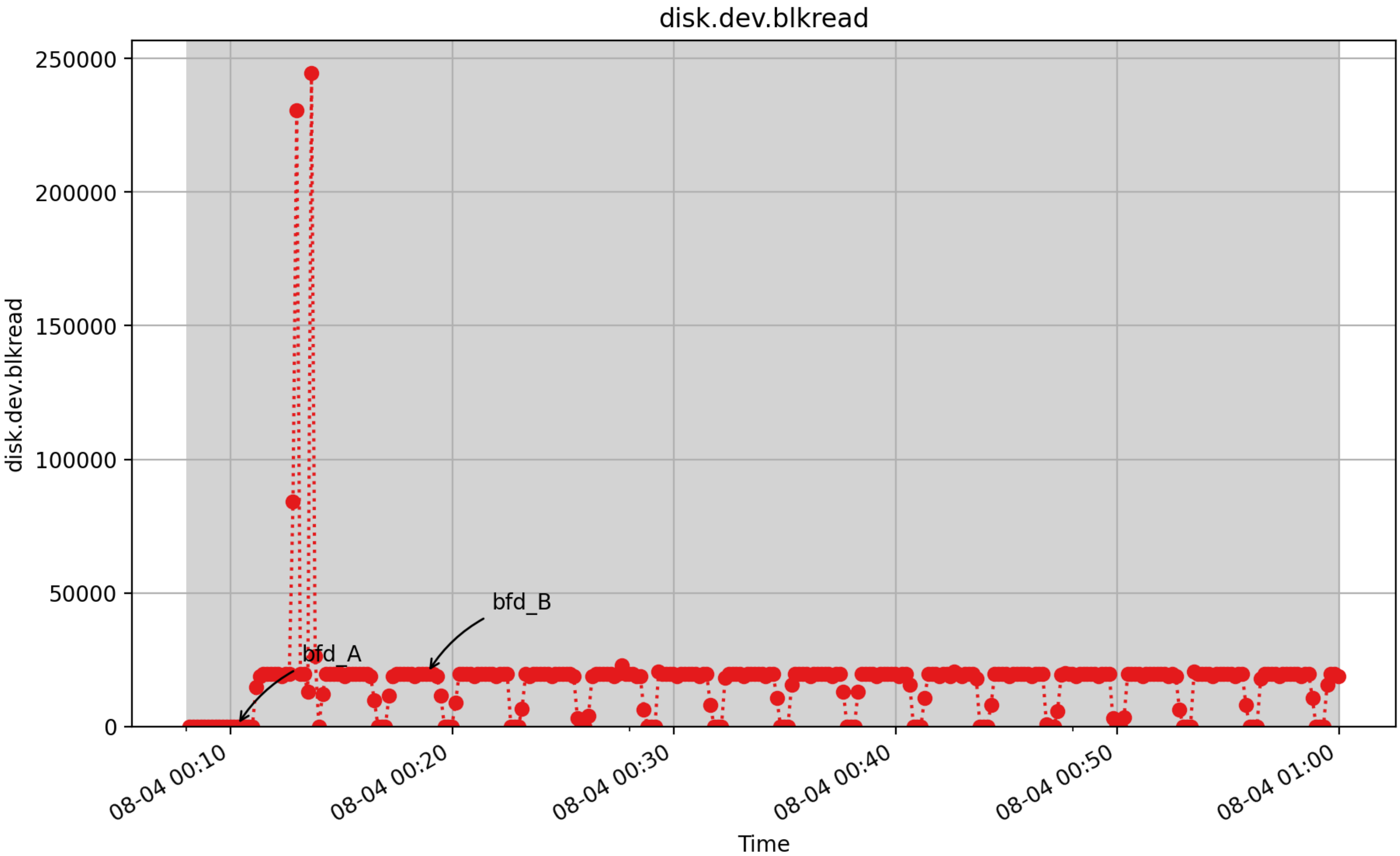
disk.dev.avg_qlen: average read and write I/O queue length to the device during the reporting interval. (- DOUBLE)



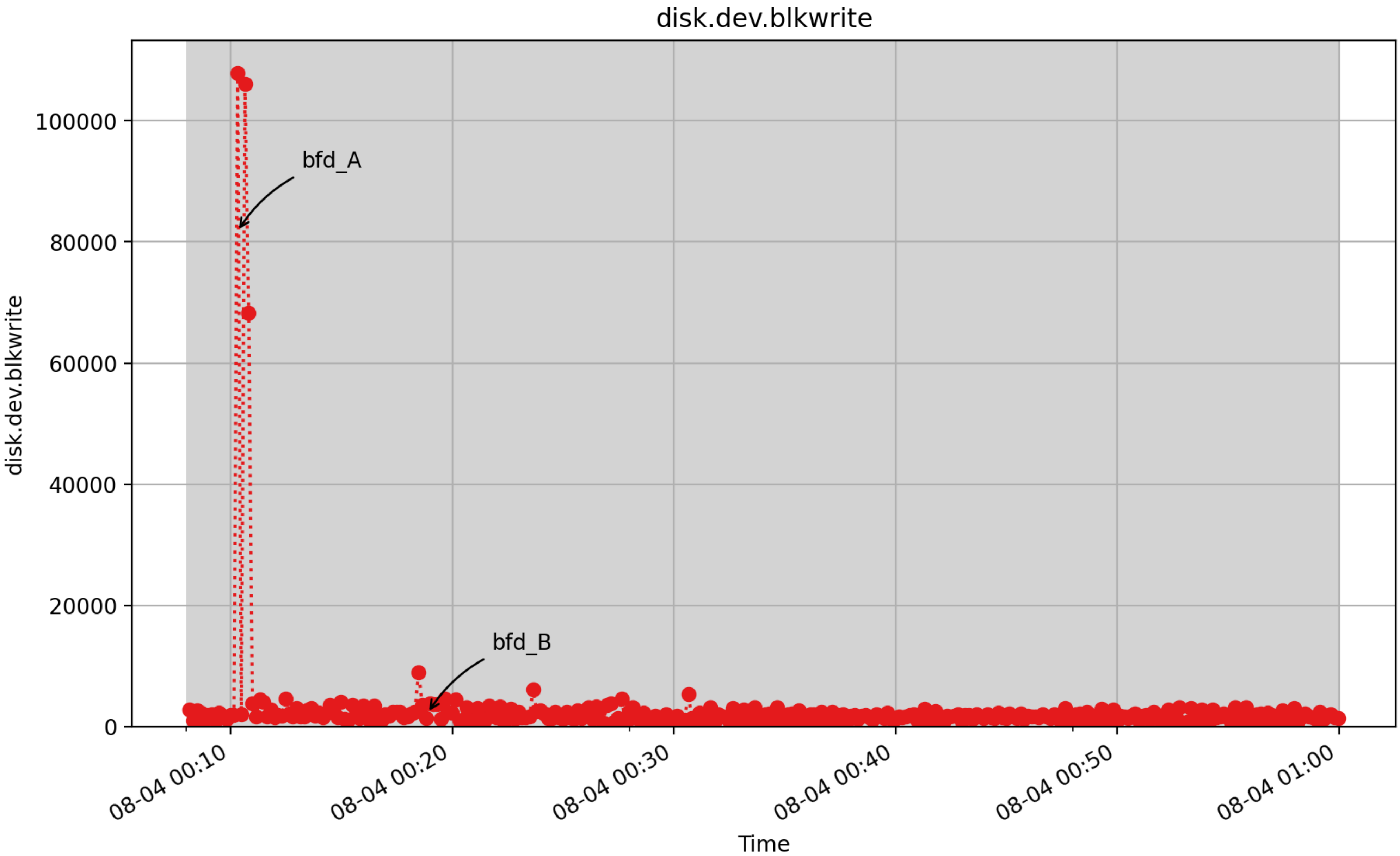
disk.dev.avg_rqsz: average I/O request size for both reads and writes during the reporting interval. (Kbyte / count - DOUBLE)



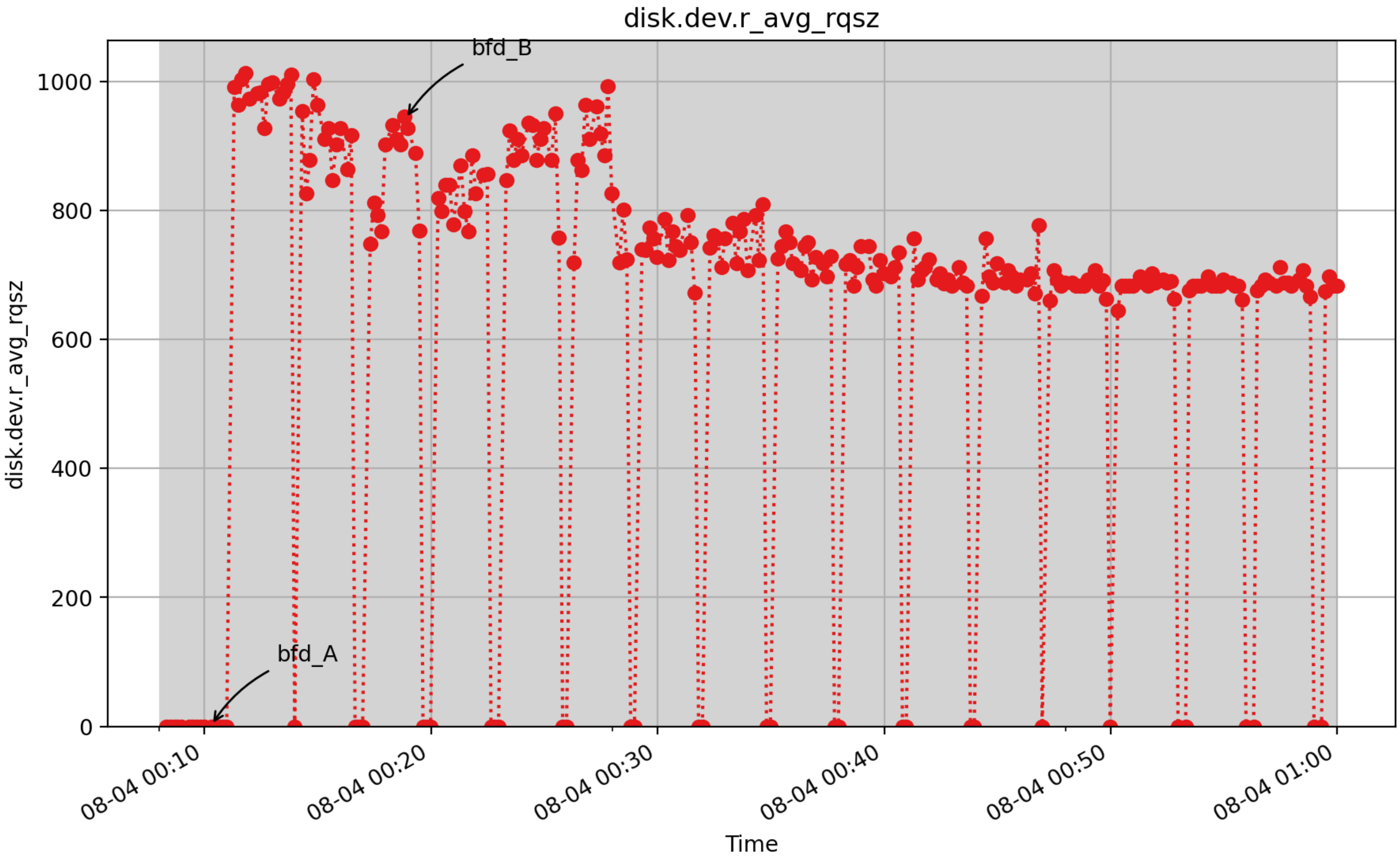
disk.dev.await: average time in milliseconds that read and write requests were queued (and serviced) during the reporting interval. (millisec / count - DOUBLE)



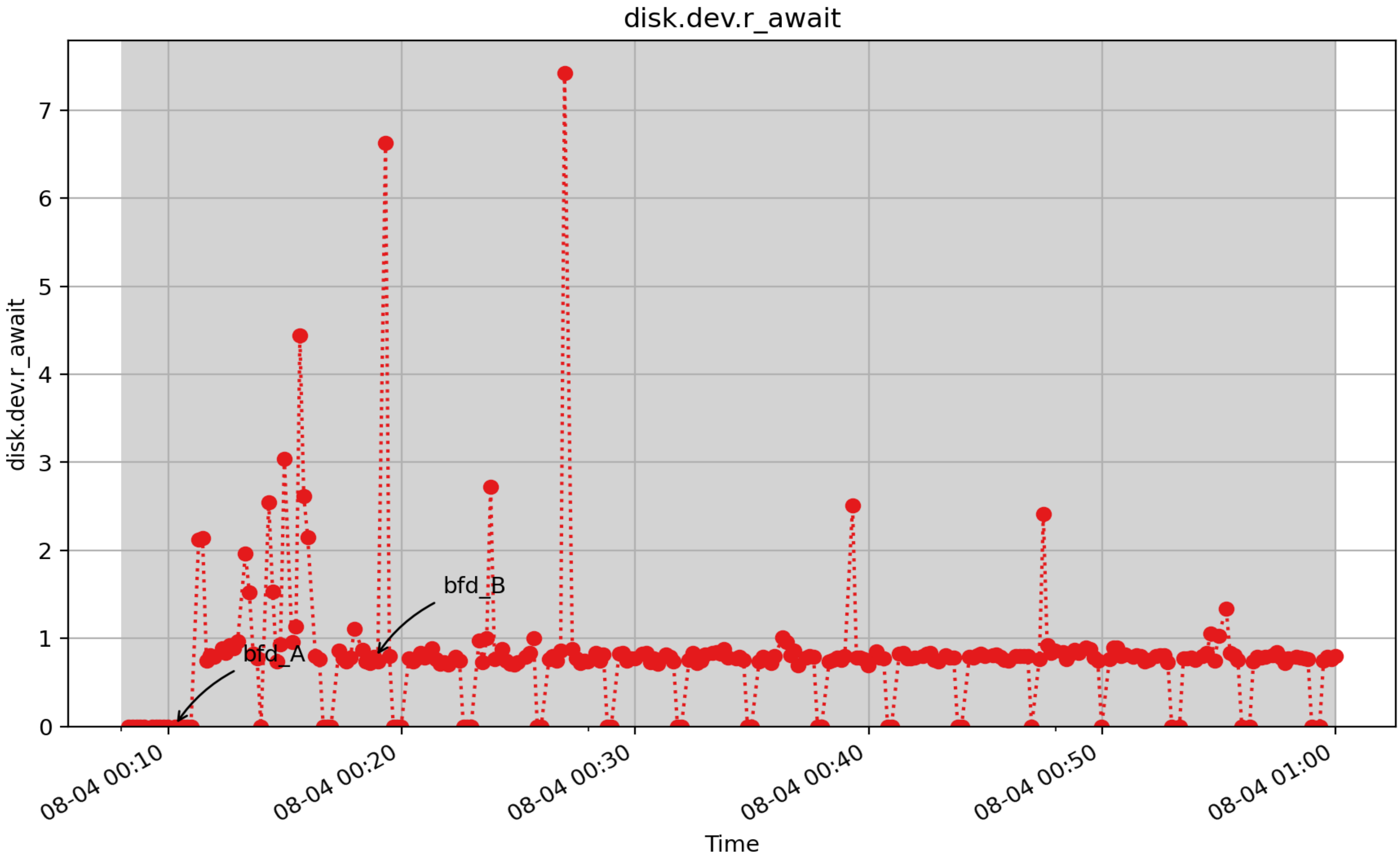
disk.dev.blkread: Cumulative number of disk block read operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*



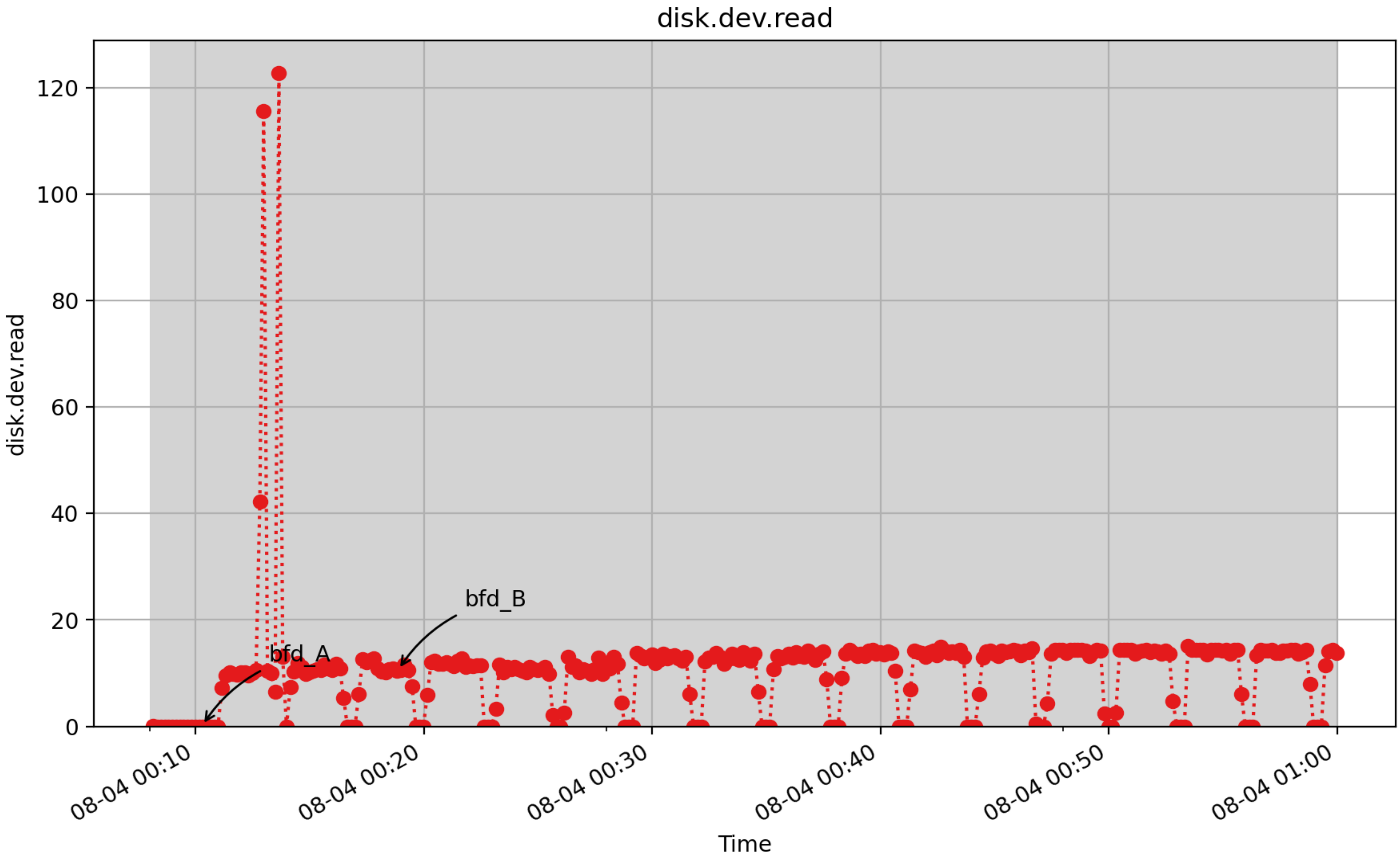
disk.dev.blkwrite: Cumulative number of disk block write operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*



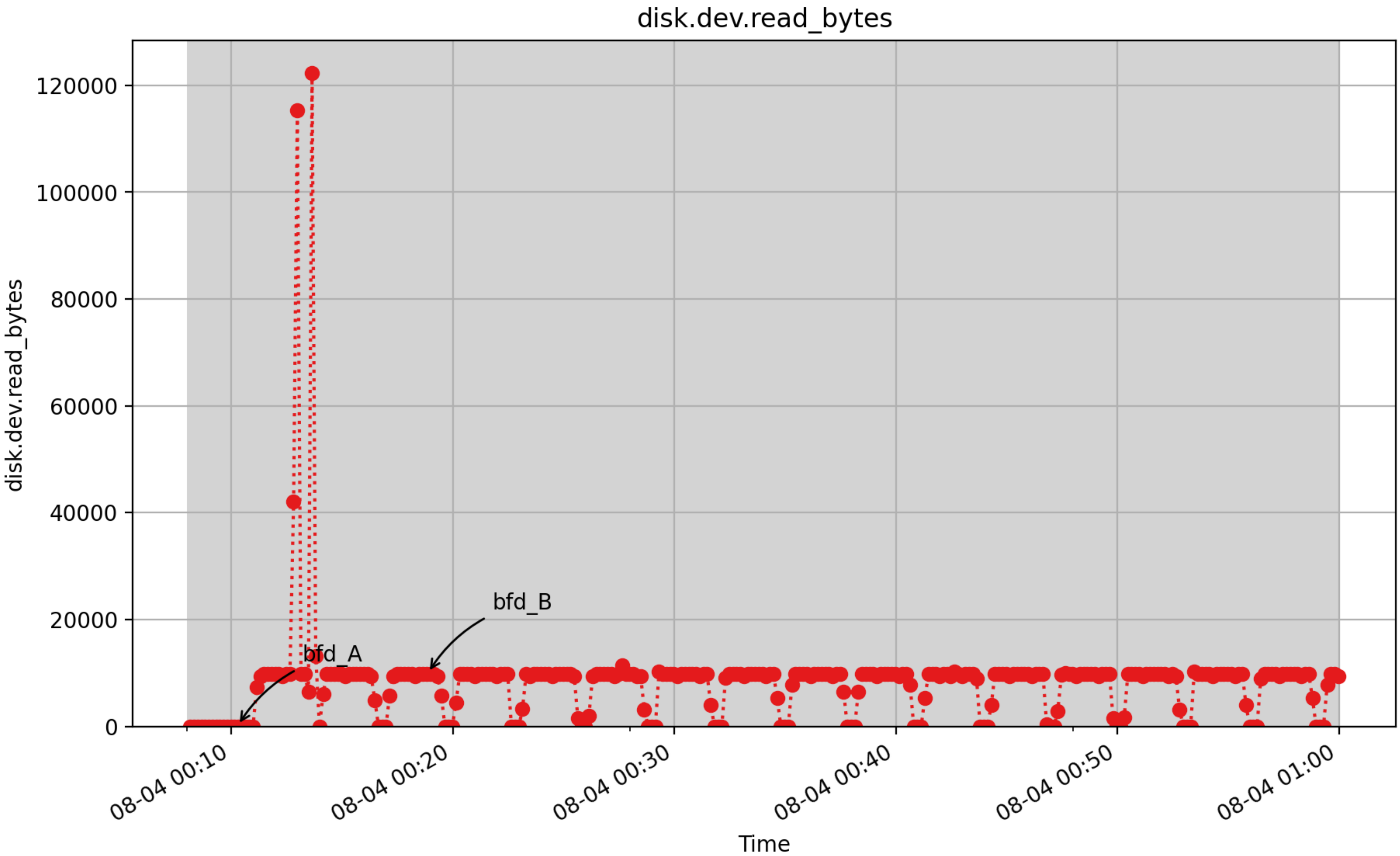
disk.dev.r_avg_rqsz: average I/O request size for reads to the device during the reporting interval. (Kbyte / count - DOUBLE)



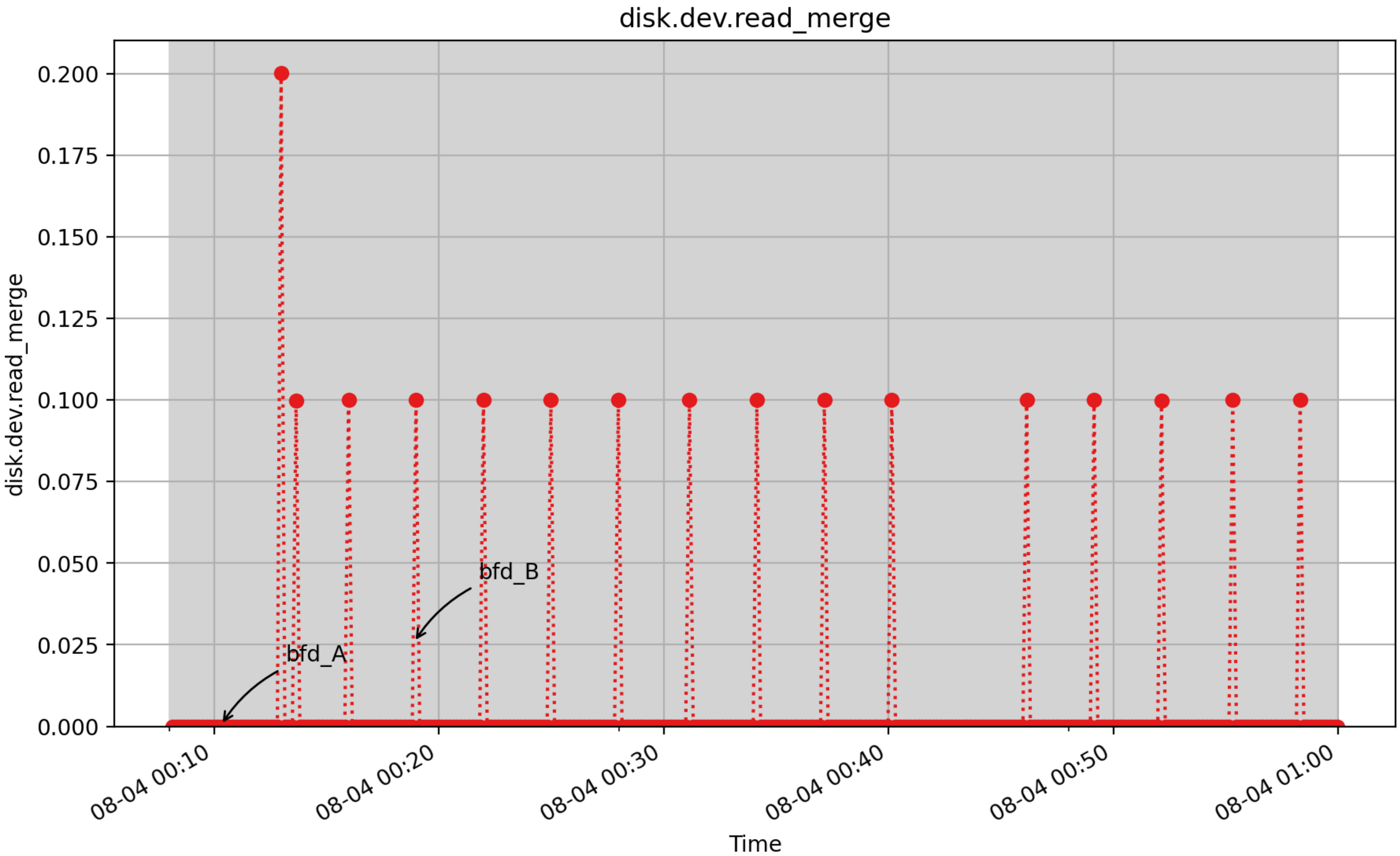
disk.dev.r_await: average time in milliseconds that read requests were queued (and serviced) during the reporting interval. (millisec / count - DOUBLE)



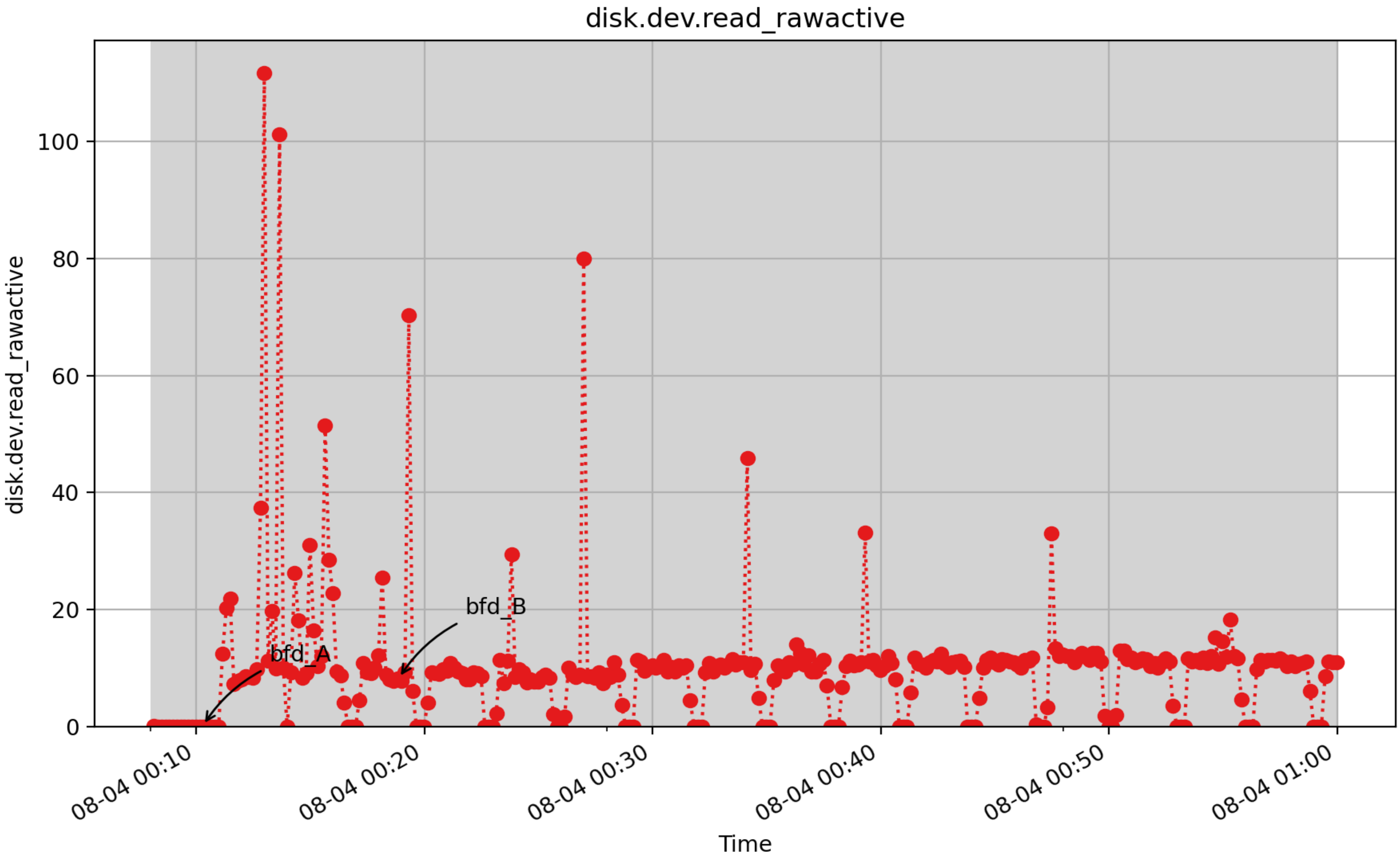
disk.dev.read: Cumulative number of disk read operations since system boot time (subject to counter wrap).
(count - U64) - *rate converted*



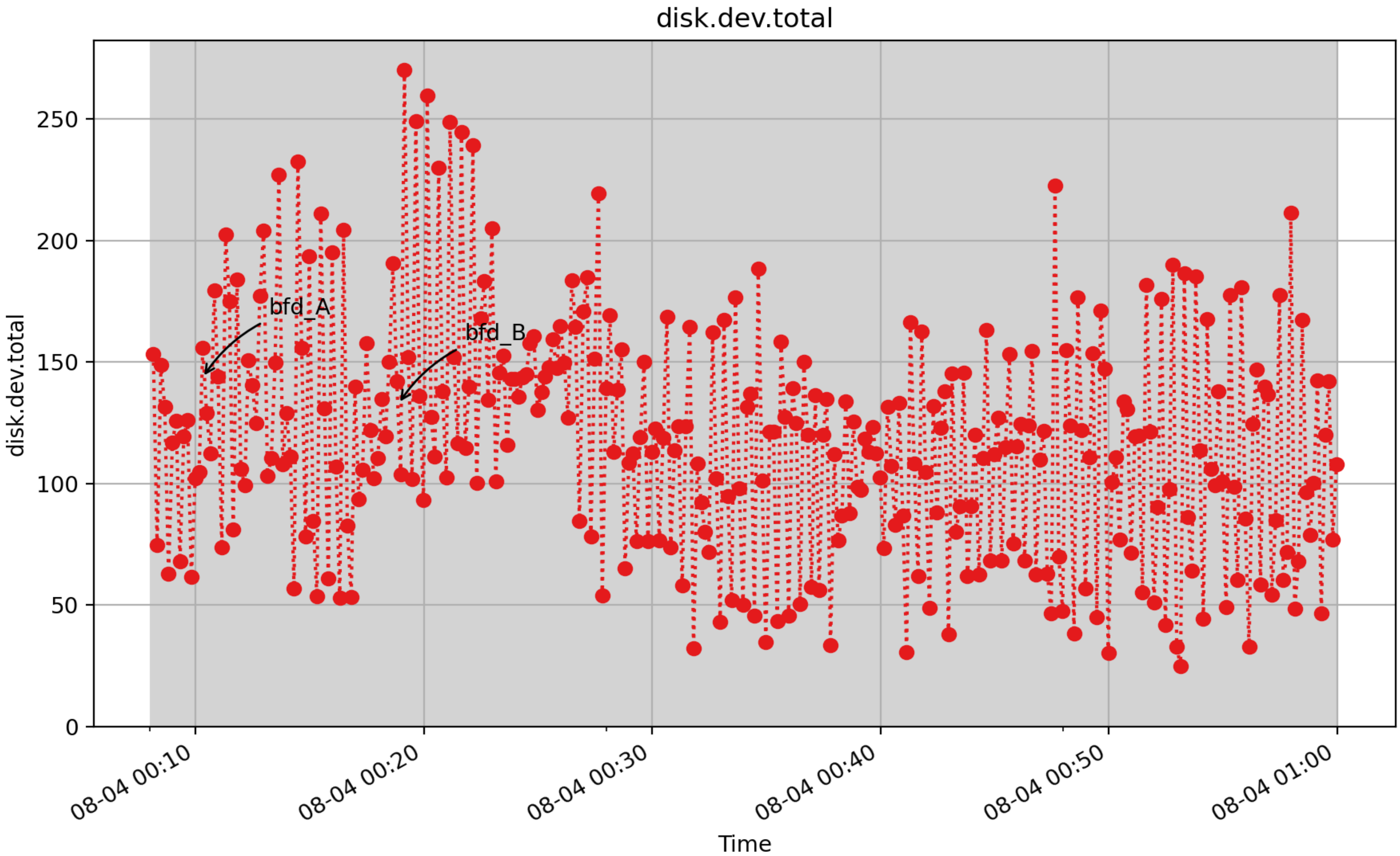
disk.dev.read_bytes: per-disk count of bytes read (Kbyte - U64) - *rate converted*



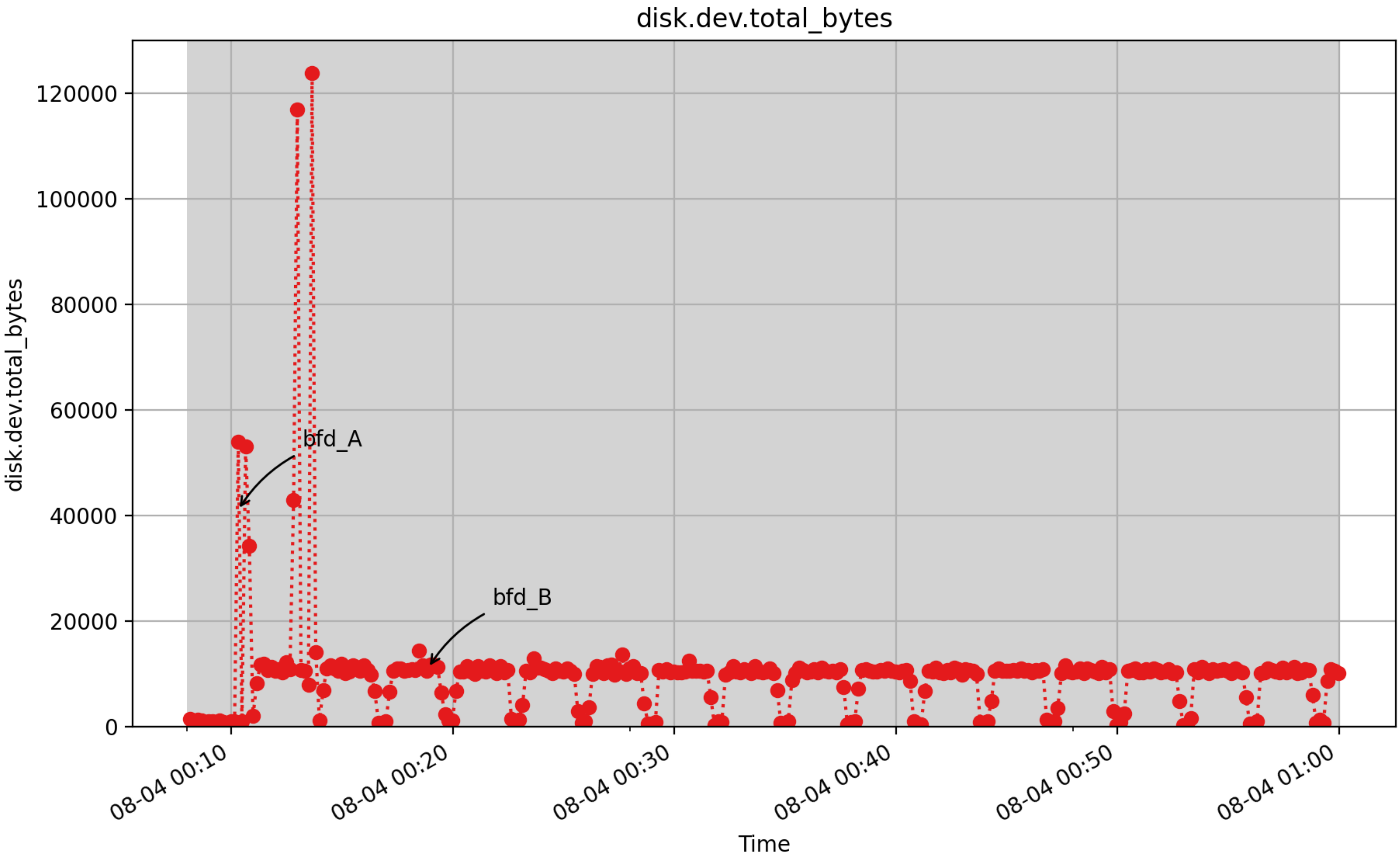
disk.dev.read_merge: Count of read requests that were merged with an already queued read request. (count - U64) - *rate converted*



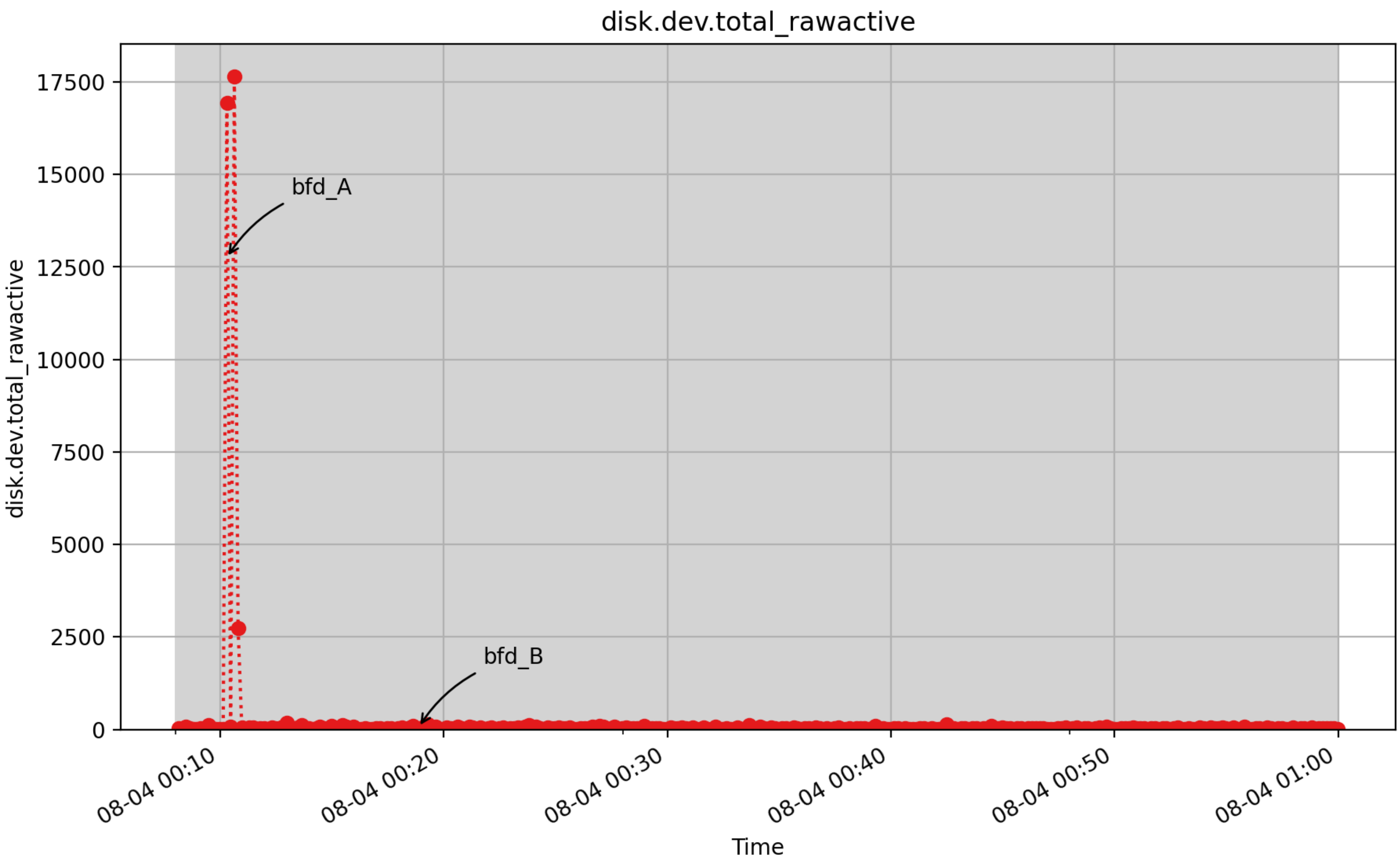
disk.dev.read_rawactive: For each completed read on each disk the response time (queue time plus service time) in milliseconds is added to the associated instance of this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding reads for a disk. When divided by the number of completed reads for a disk (disk.dev.read), the value represents the stochastic average of the read response (or wait) time for that disk. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from existing performance tools: iostat.dev.r_await = delta(disk.dev.read_rawactive) / delta(disk.dev.read) (millisec - U32) - *rate converted*



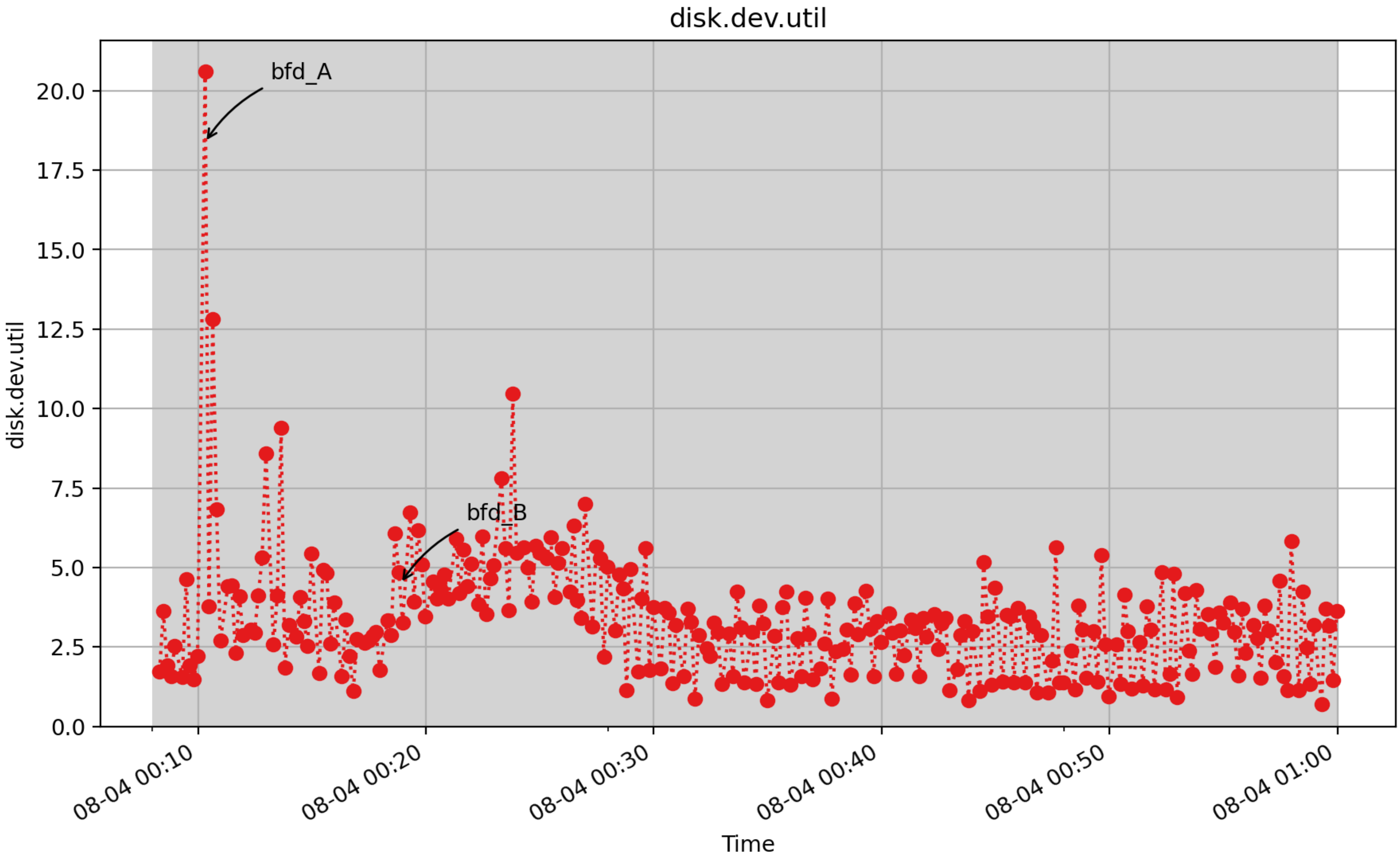
disk.dev.total: Cumulative number of disk read and write operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*



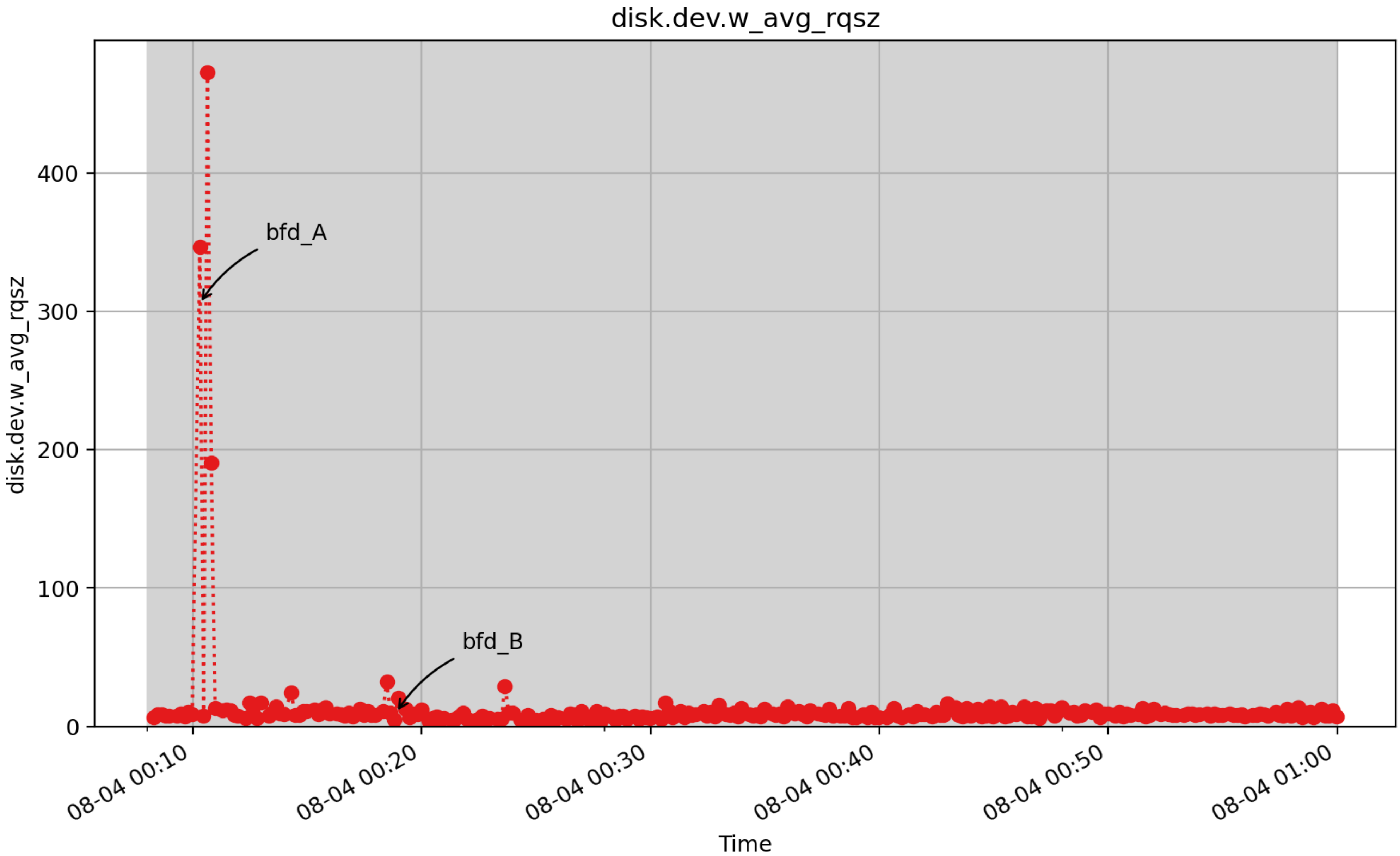
disk.dev.total_bytes: per-disk count of total bytes read and written (Kbyte - U64) - *rate converted*



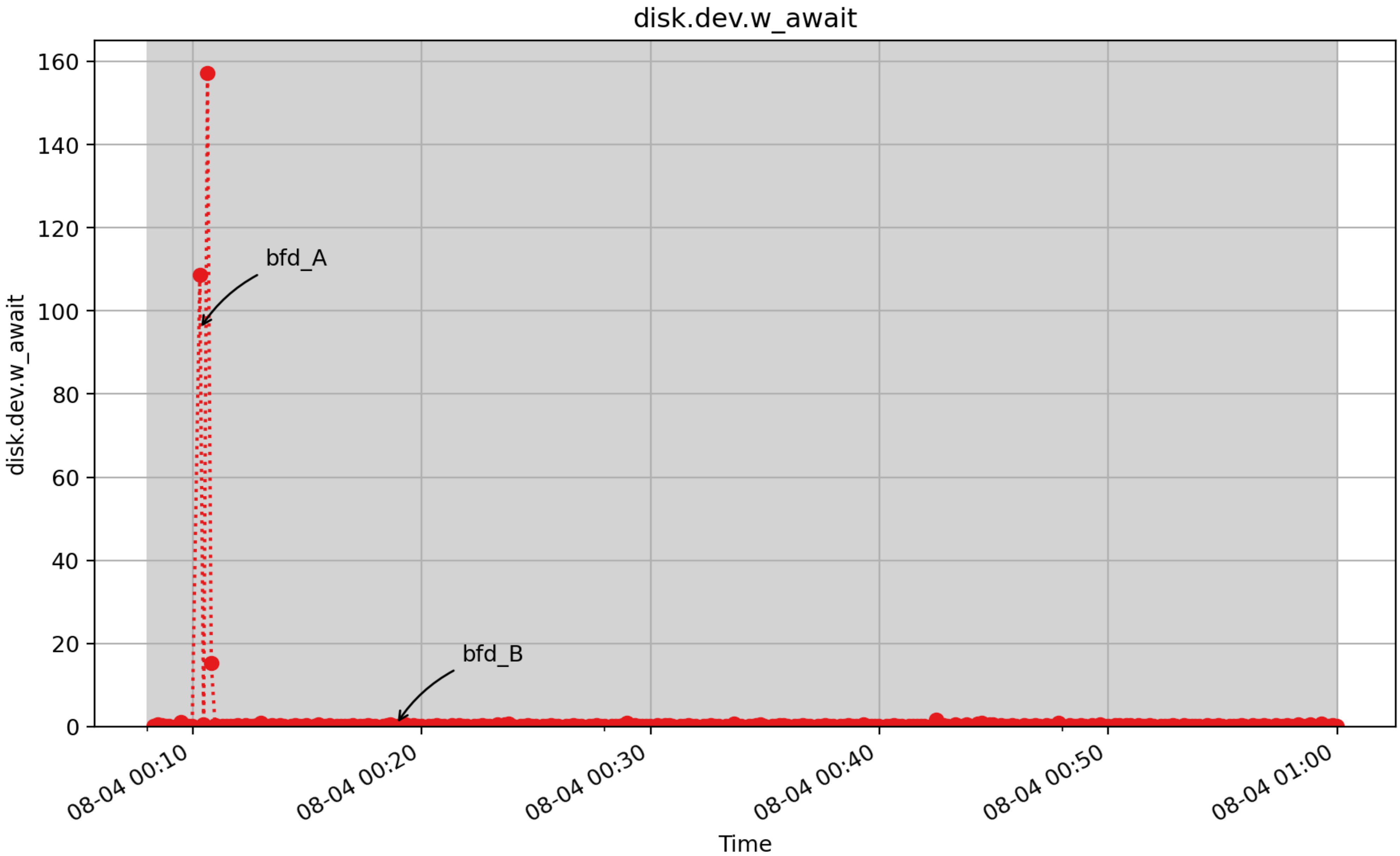
disk.dev.total_rawactive: For each completed I/O on each disk the response time (queue time plus service time) in milliseconds is added to the associated instance of this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding I/Os for a disk. When divided by the number of completed I/Os for a disk (disk.dev.total), the value represents the stochastic average of the I/O response (or wait) time for that disk. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from existing performance tools: iostat.dev.await = delta(disk.dev.total_rawactive) / delta(disk.dev.total) (millisec - U32) - *rate converted*



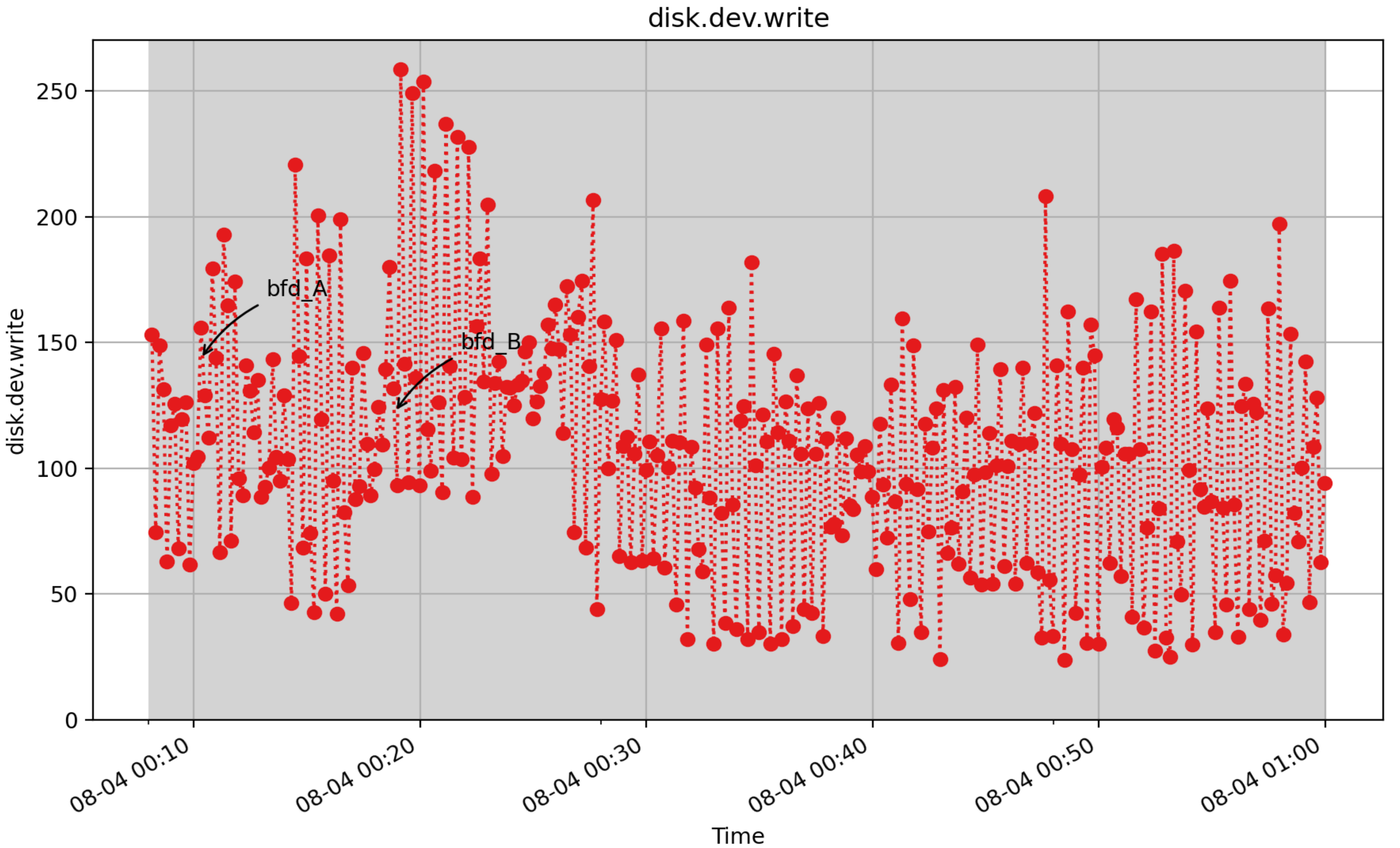
disk.dev.util: The percentage of time during the reporting interval that the device was busy processing requests (reads and writes). A value of 100% indicates device saturation. (- DOUBLE)



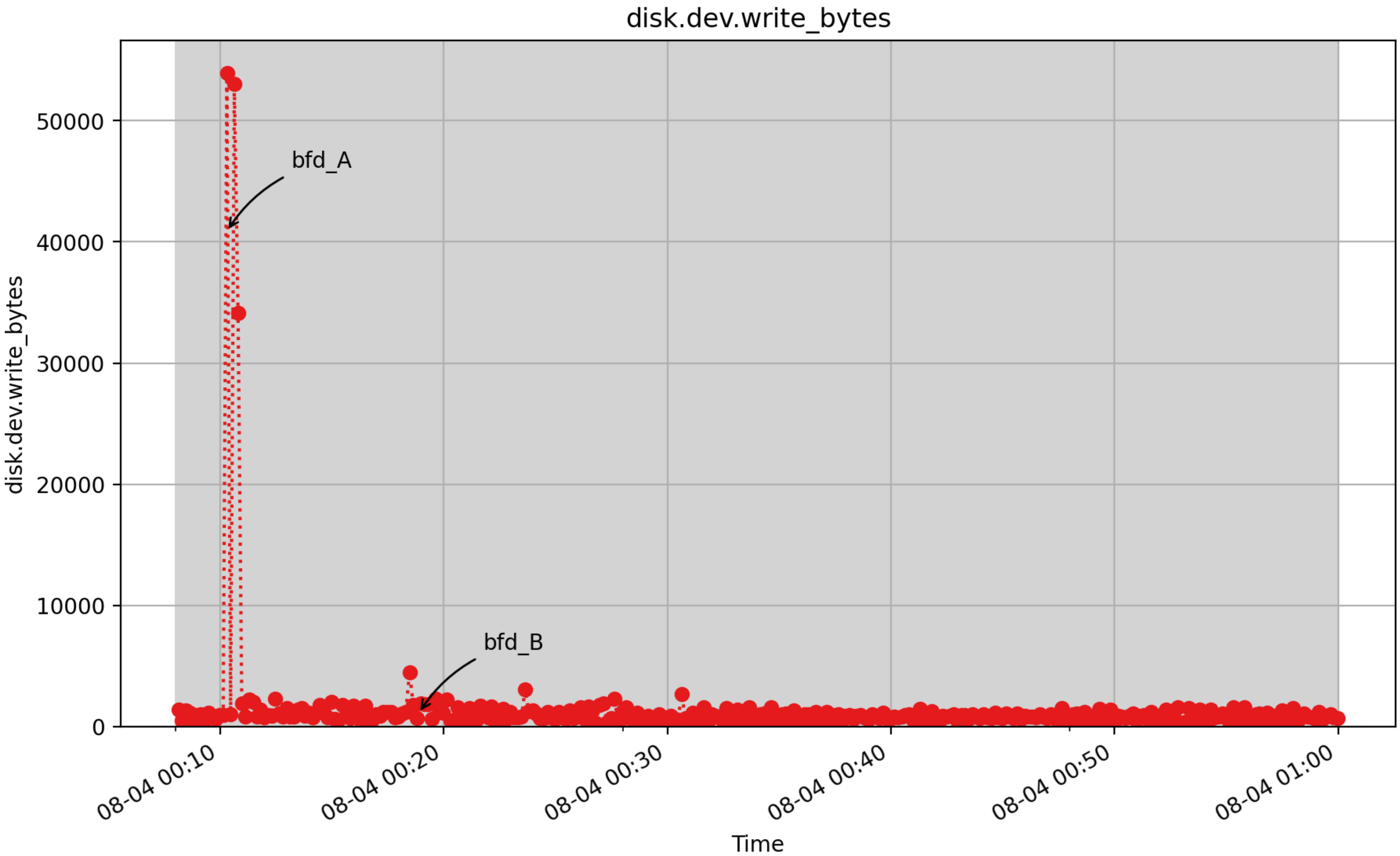
disk.dev.w_avg_rqsz: average I/O request size for writes to the device during the reporting interval. (Kbyte / count - DOUBLE)



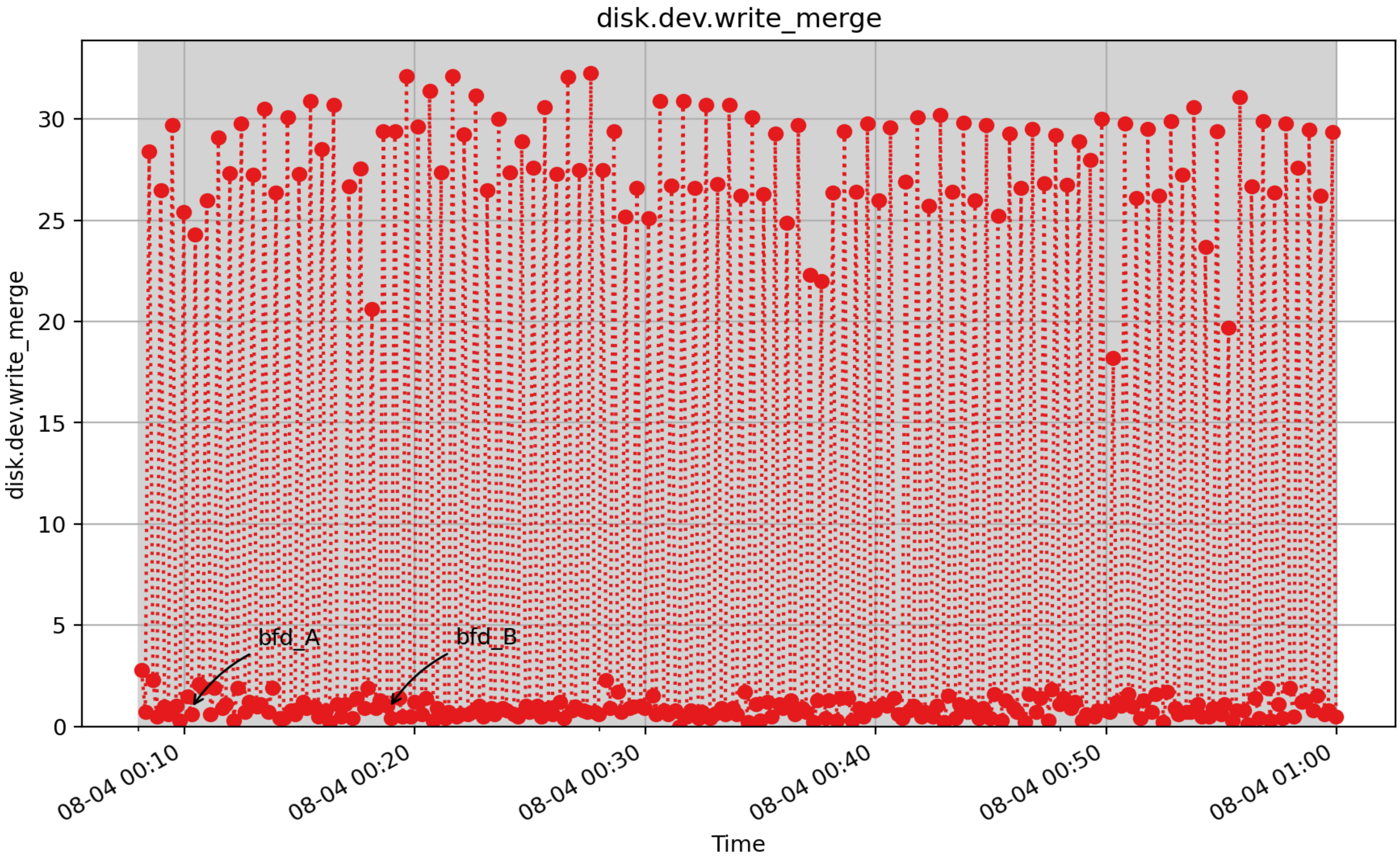
disk.dev.w_await: average time in milliseconds that write requests were queued (and serviced) during the reporting interval. (millisec / count - DOUBLE)



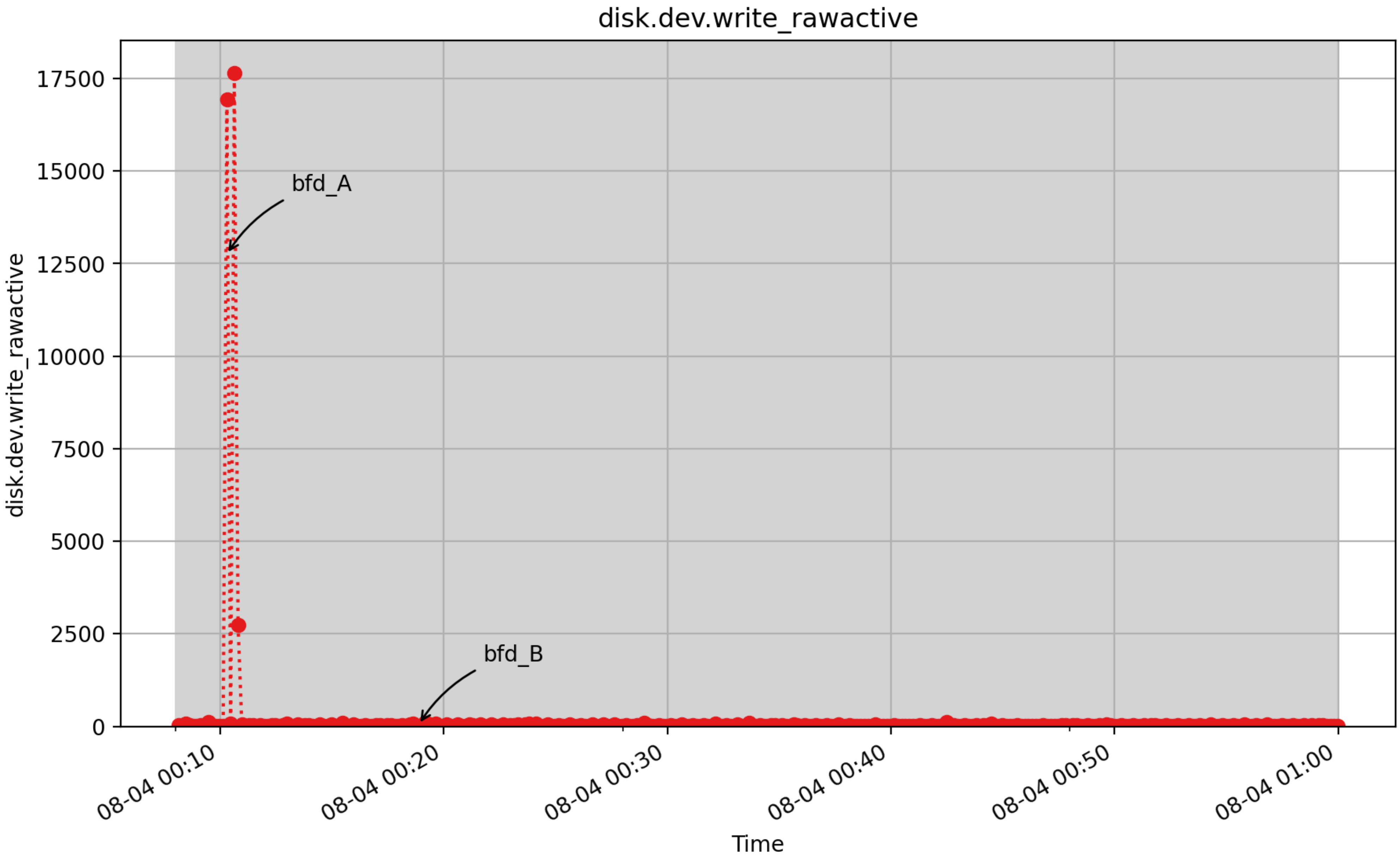
disk.dev.write: Cumulative number of disk write operations since system boot time (subject to counter wrap). (count - U64) - *rate converted*



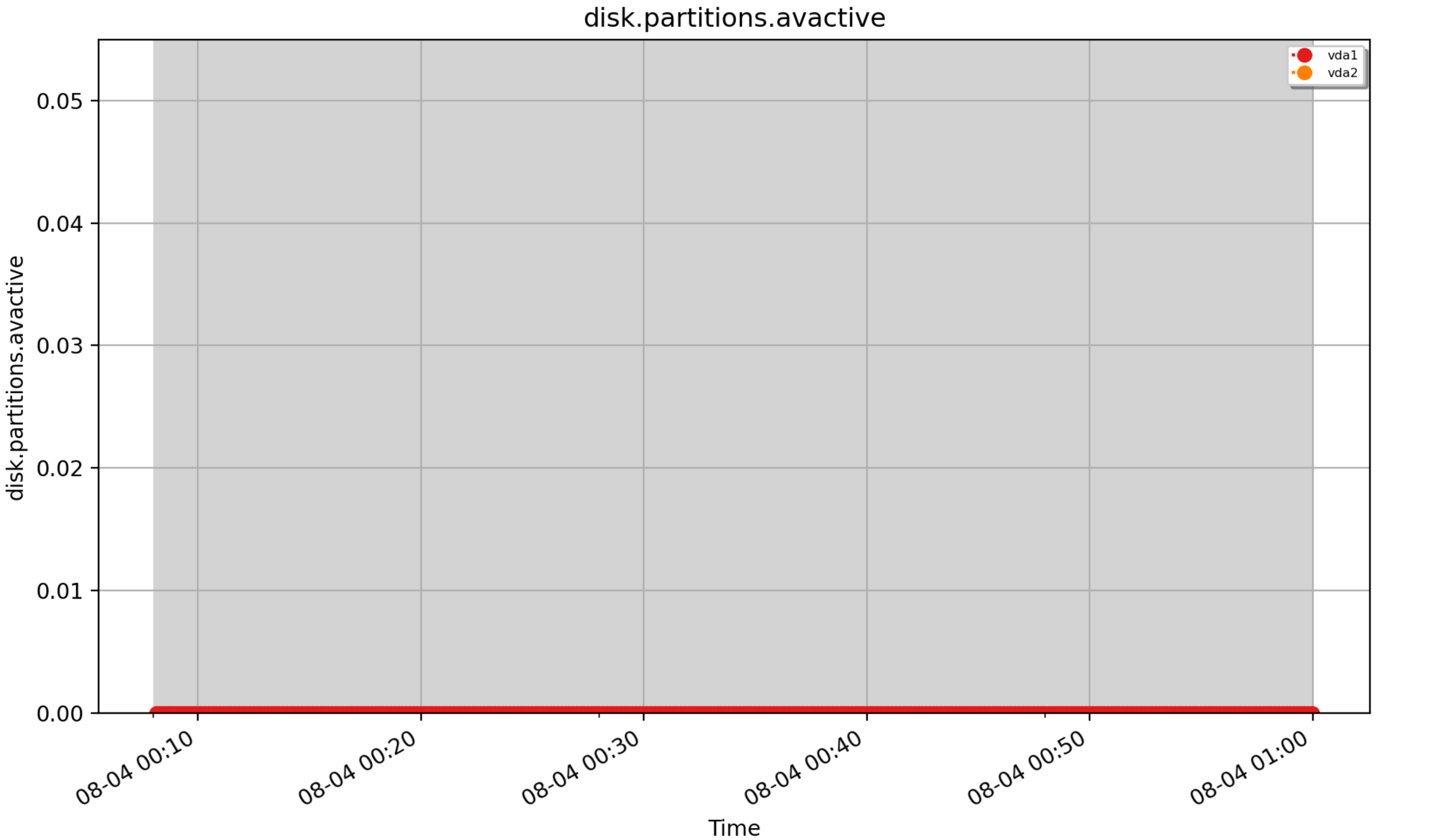
disk.dev.write_bytes: per-disk count of bytes written (Kbyte - U64) - *rate converted*



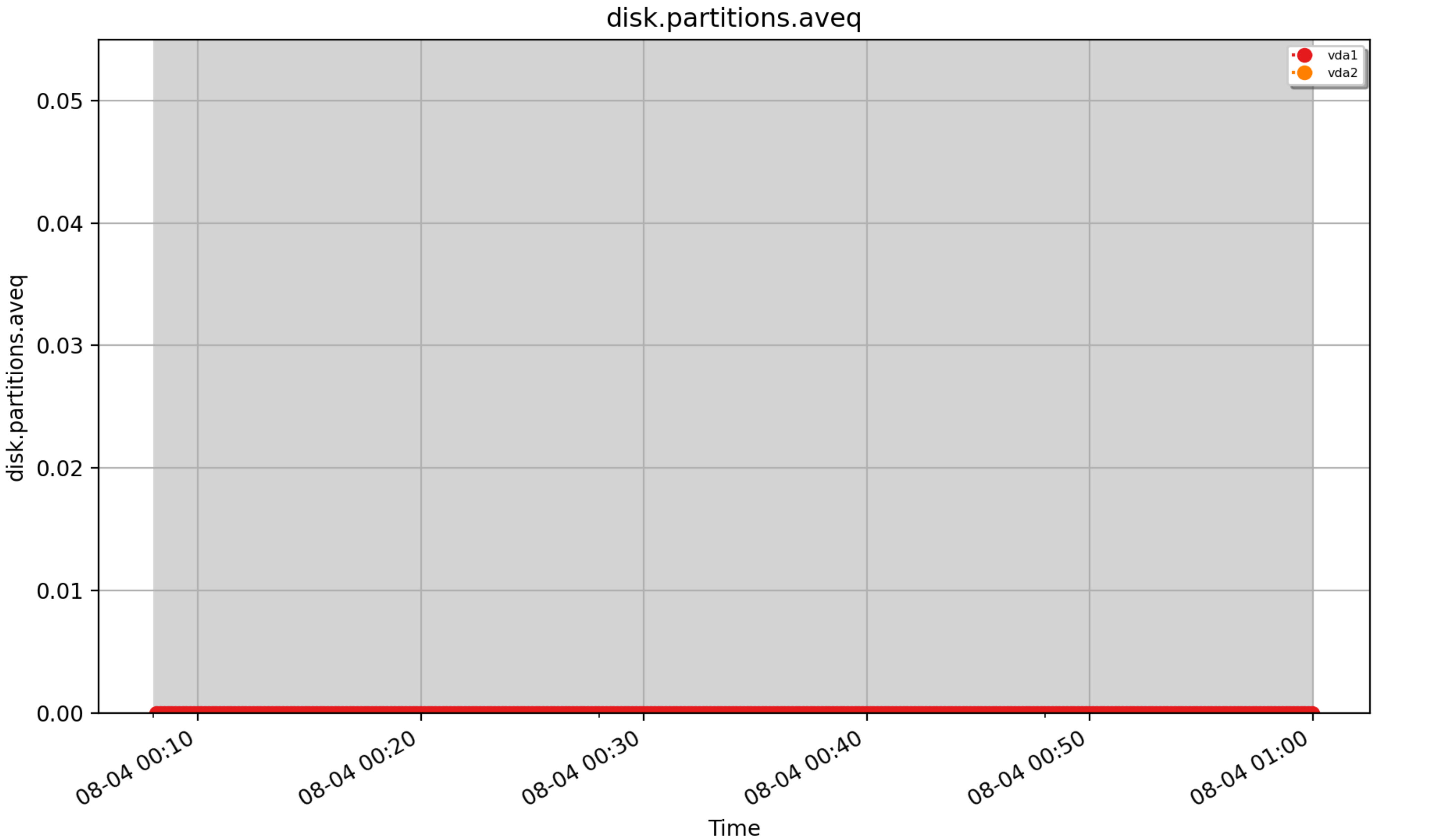
disk.dev.write_merge: Count of write requests that were merged with an already queued write request.
(count - U64) - *rate converted*



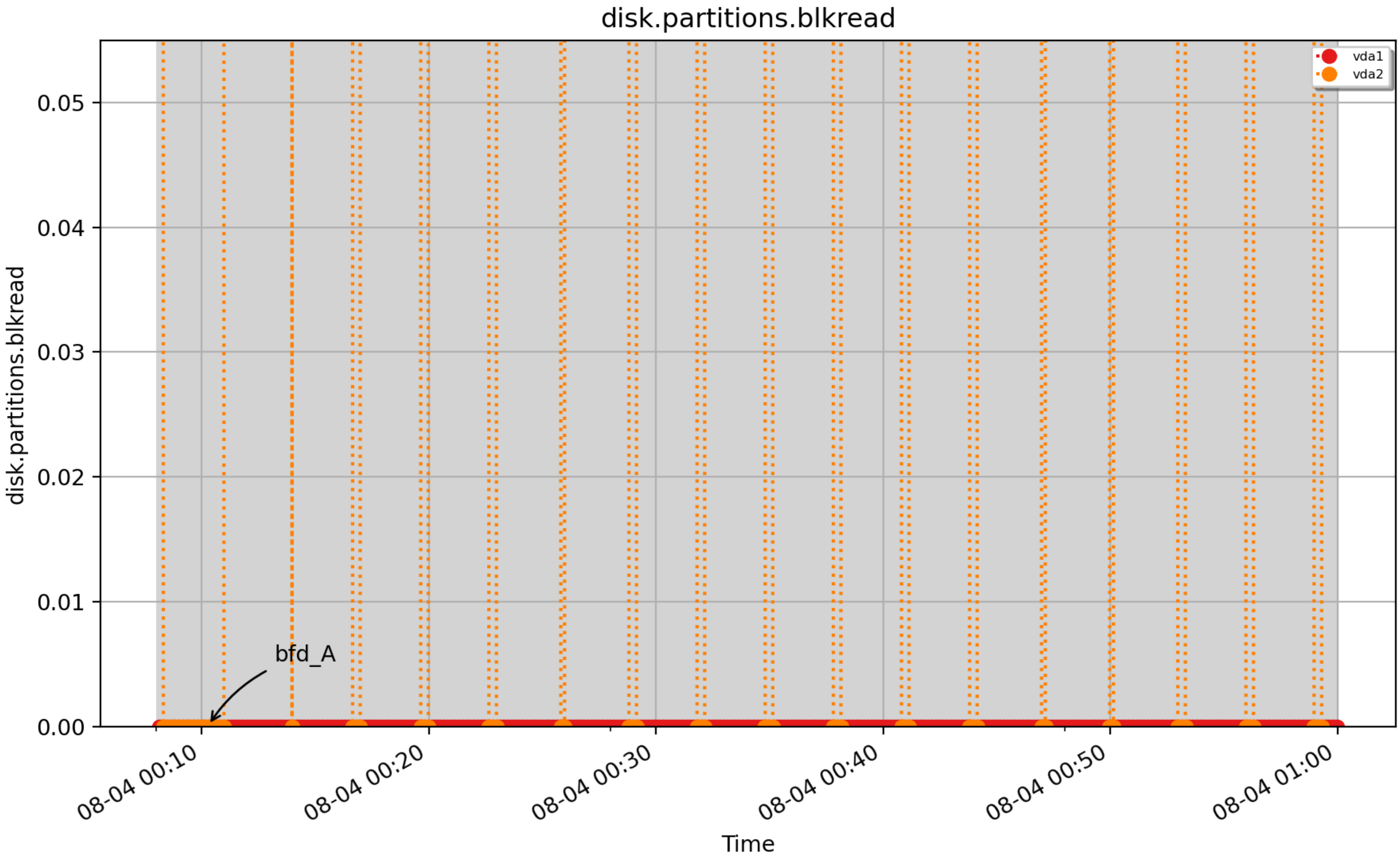
disk.dev.write_rawactive: For each completed write on each disk the response time (queue time plus service time) in milliseconds is added to the associated instance of this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding writes for a disk. When divided by the number of completed writes for a disk (disk.dev.write), the value represents the stochastic average of the write response (or wait) time for that disk. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from existing performance tools: iostat.dev.w_await = delta(disk.dev.write_rawactive) / delta(disk.dev.write) (millisec - U32) - *rate converted*



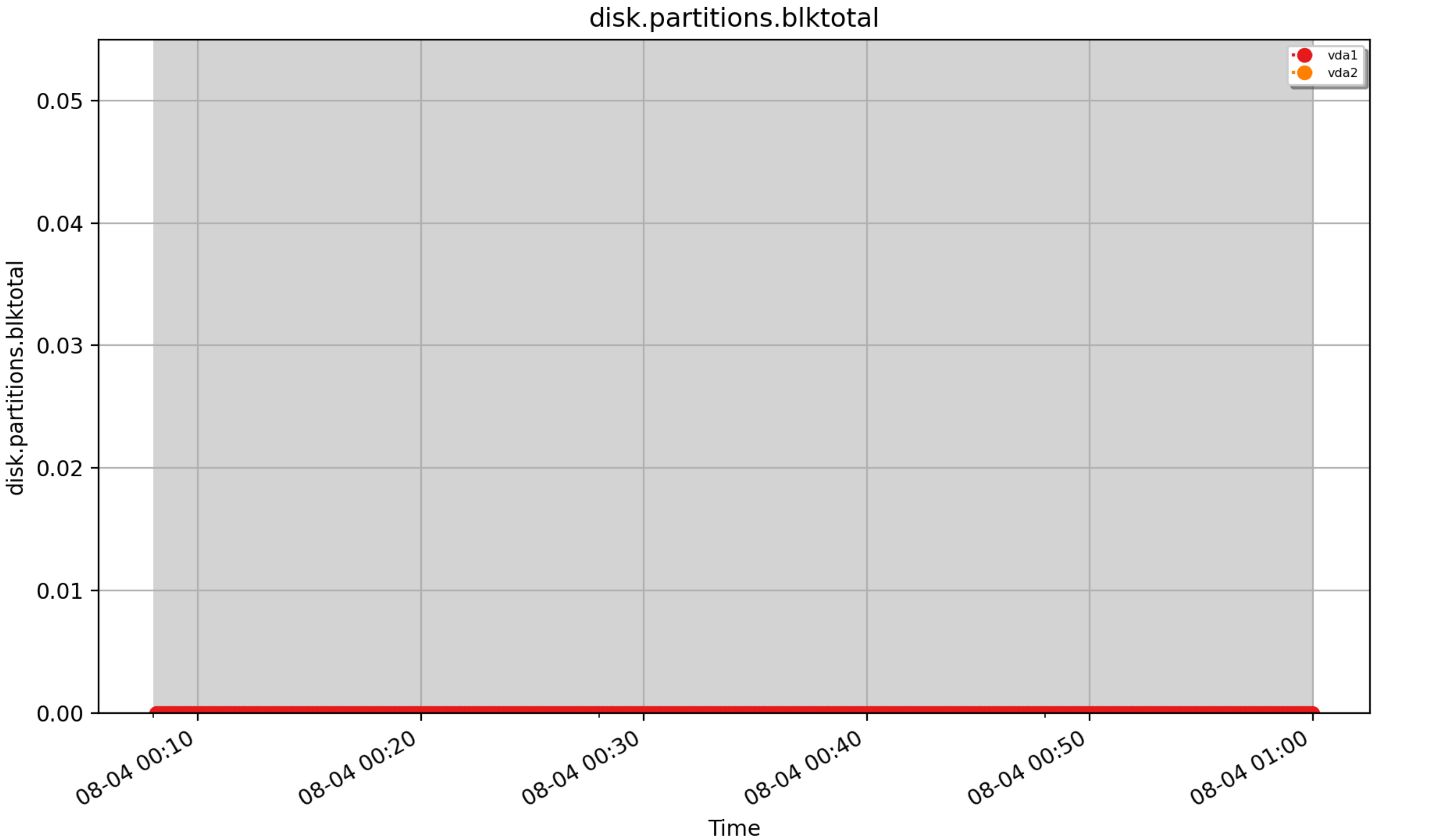
disk.partitions.avactive: Counts the number of milliseconds for which at least one I/O is in progress for each disk partition. When converted to a rate, this metric represents the average utilization of the disk partition during the sampling interval. A value of 0.5 (or 50%) means the disk partition was active (i.e. busy) half the time. (millisec - U32) - *rate converted*



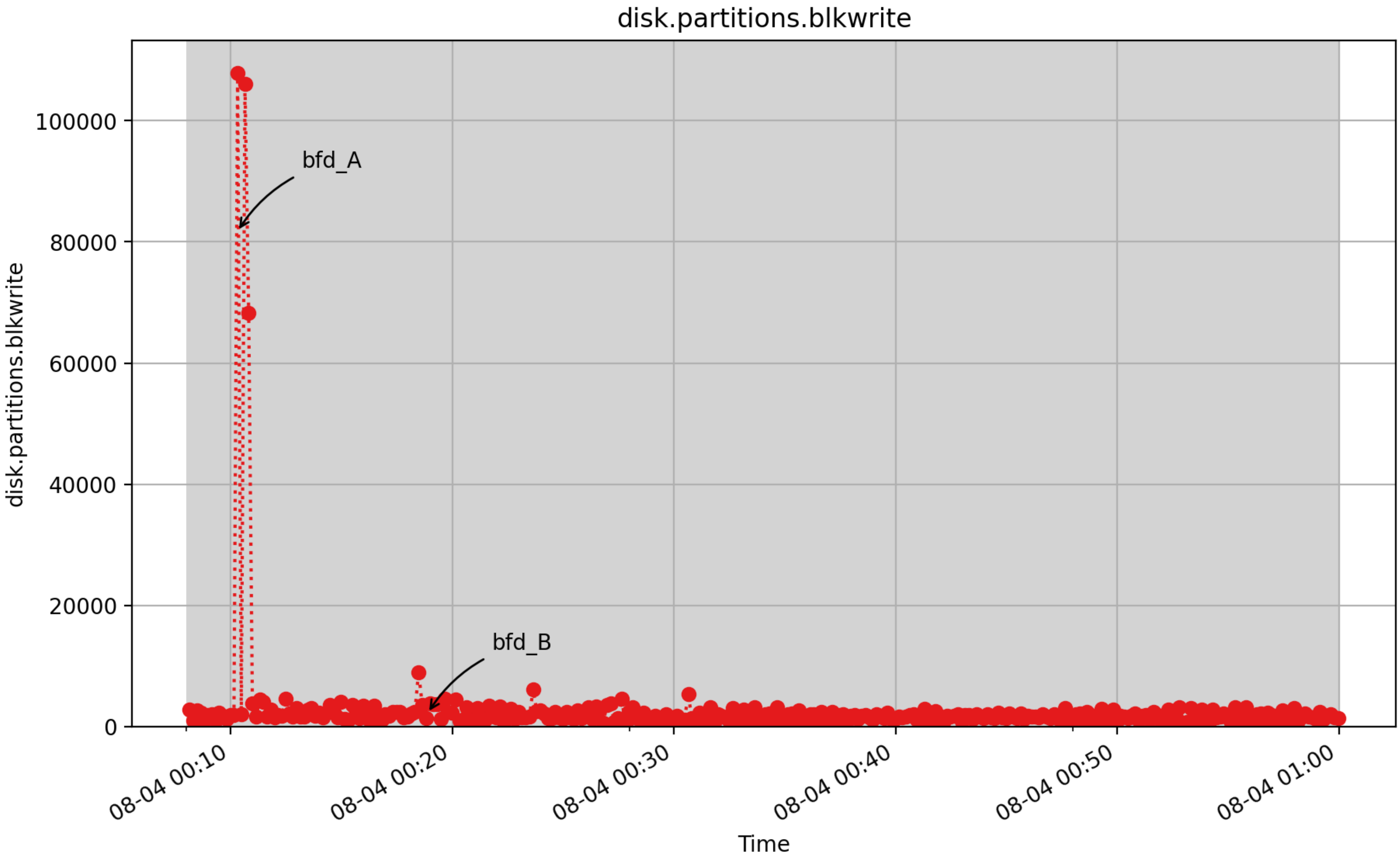
disk.partitions.aveq: per-disk-partition device time averaged count of request queue length (millisec - U32) - *rate converted*



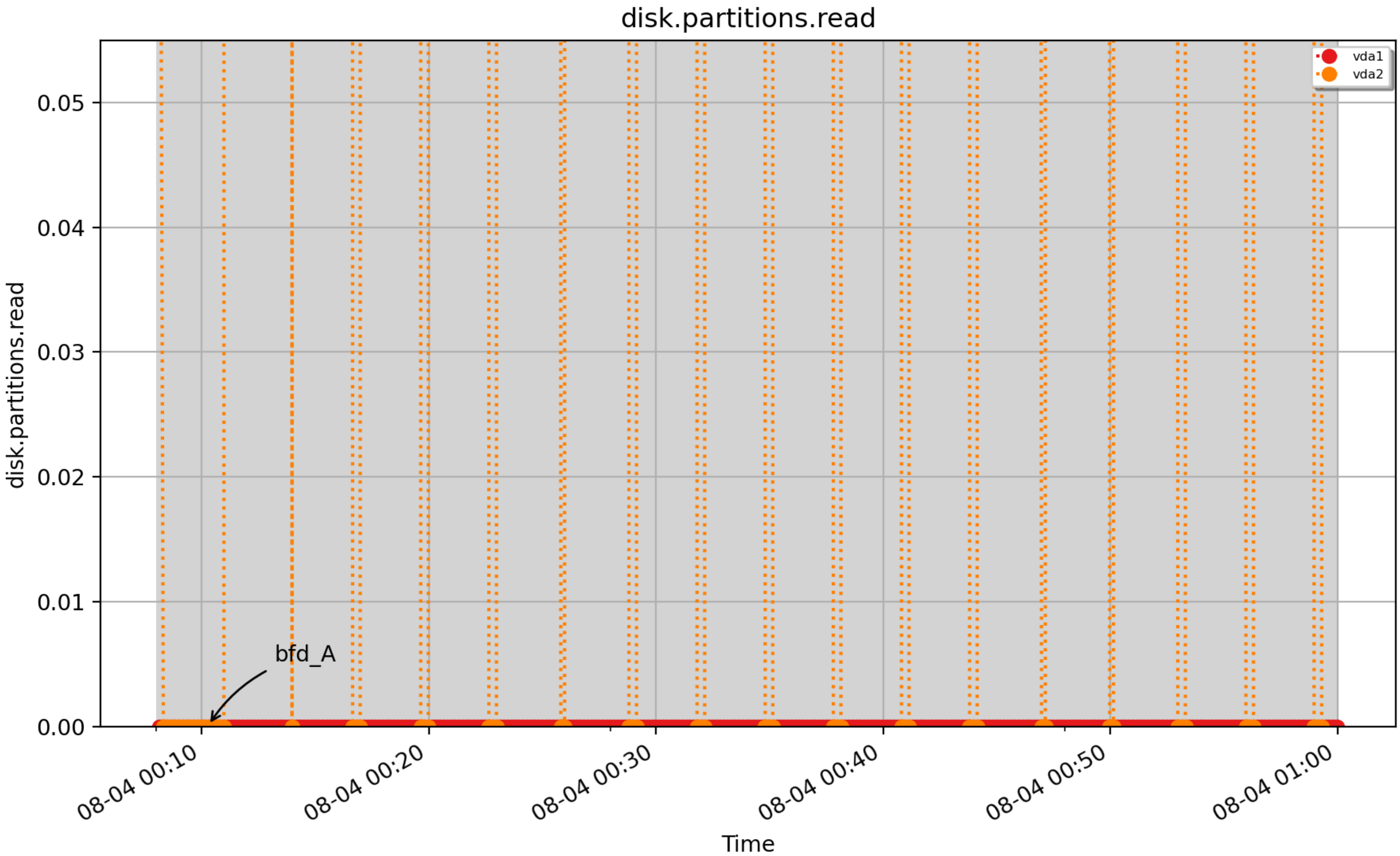
disk.partitions.blkread: Cumulative number of disk block read operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U64) - *rate converted*



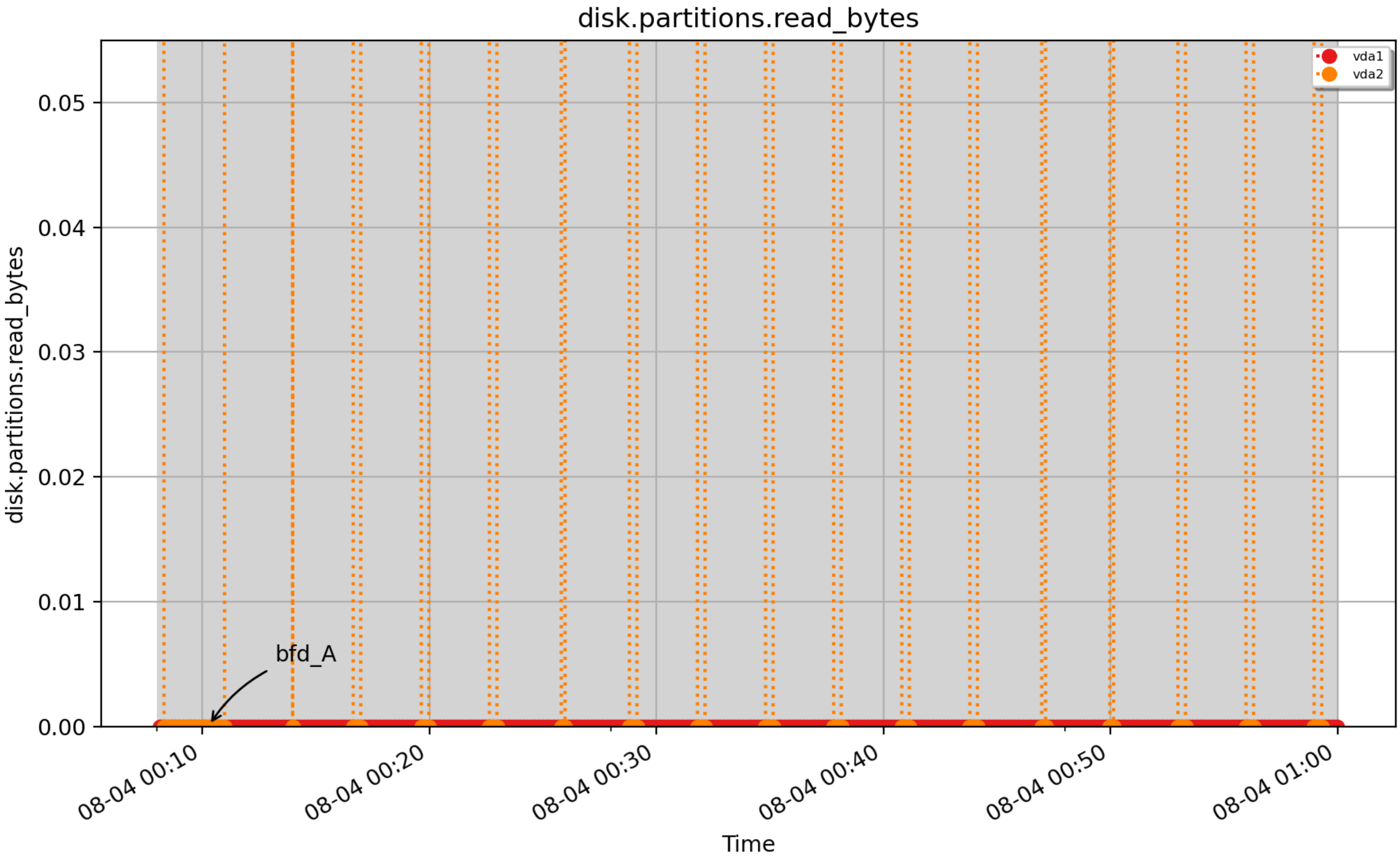
disk.partitions.blktotal: Cumulative number of disk block read and write operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U64) - *rate converted*



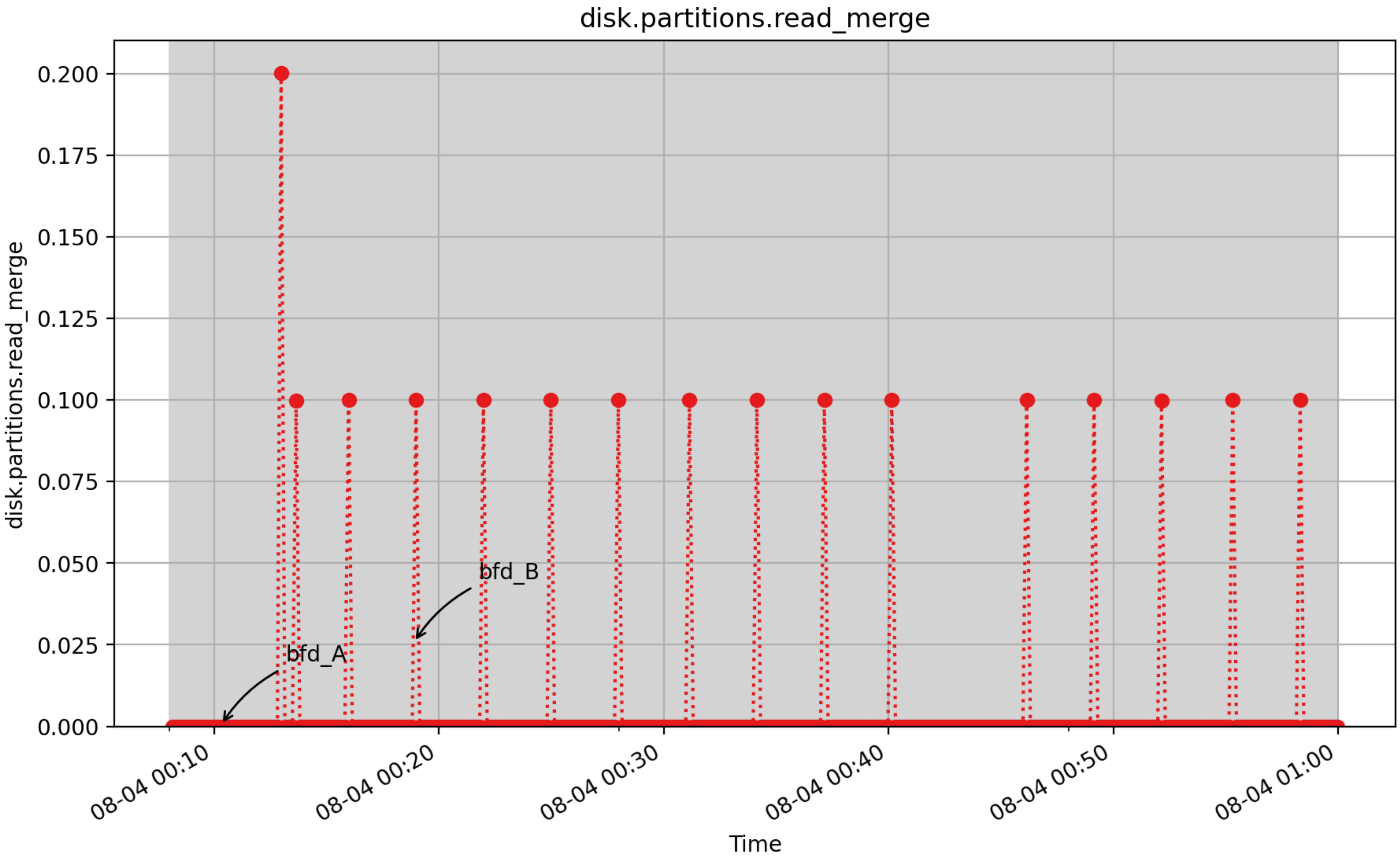
disk.partitions.blkwrite: Cumulative number of disk block write operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U64) - *rate converted*



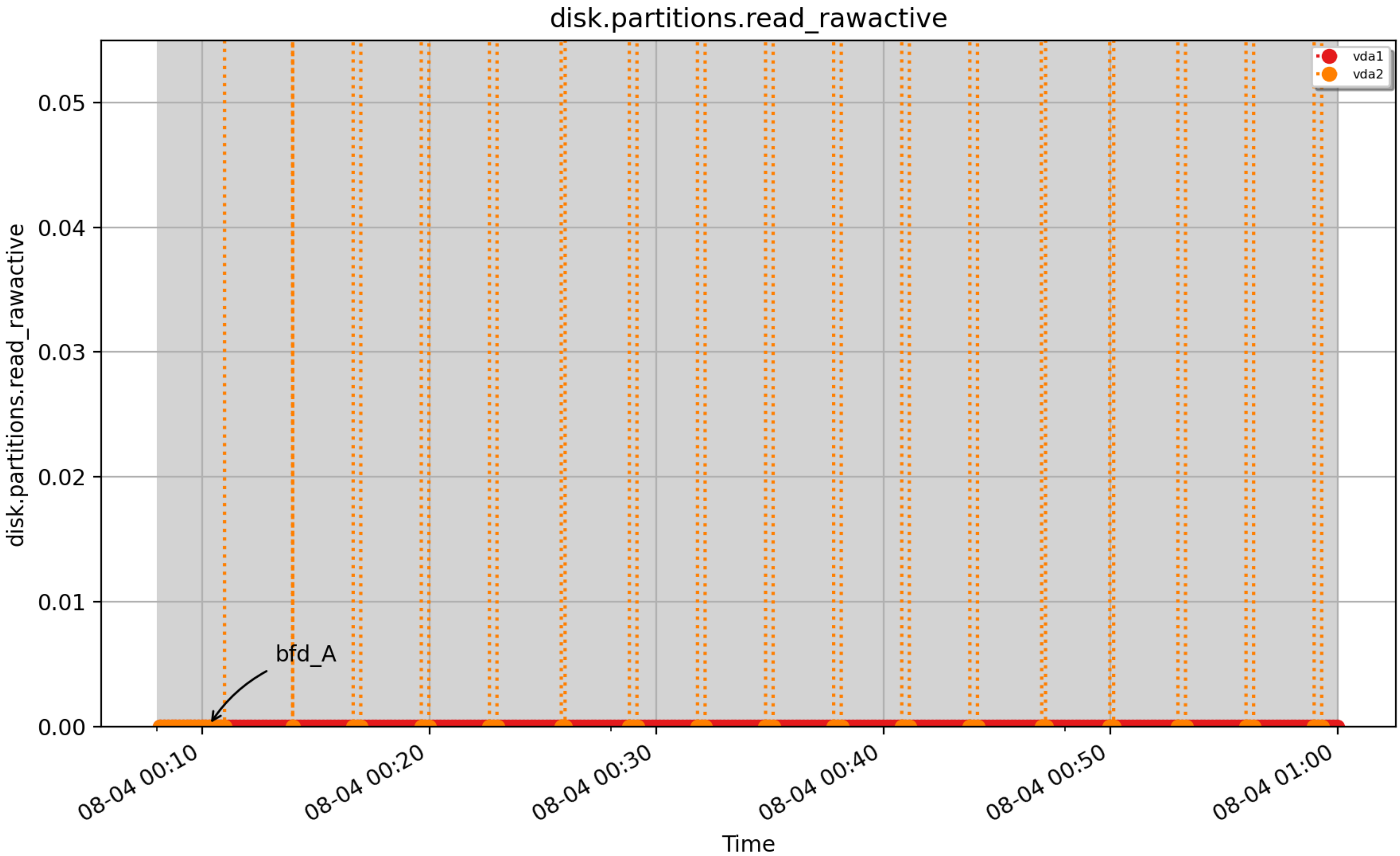
disk.partitions.read: Cumulative number of disk read operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U64) - *rate converted*



disk.partitions.read_bytes: Cumulative number of bytes read since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (Kbyte - U64) - *rate converted*

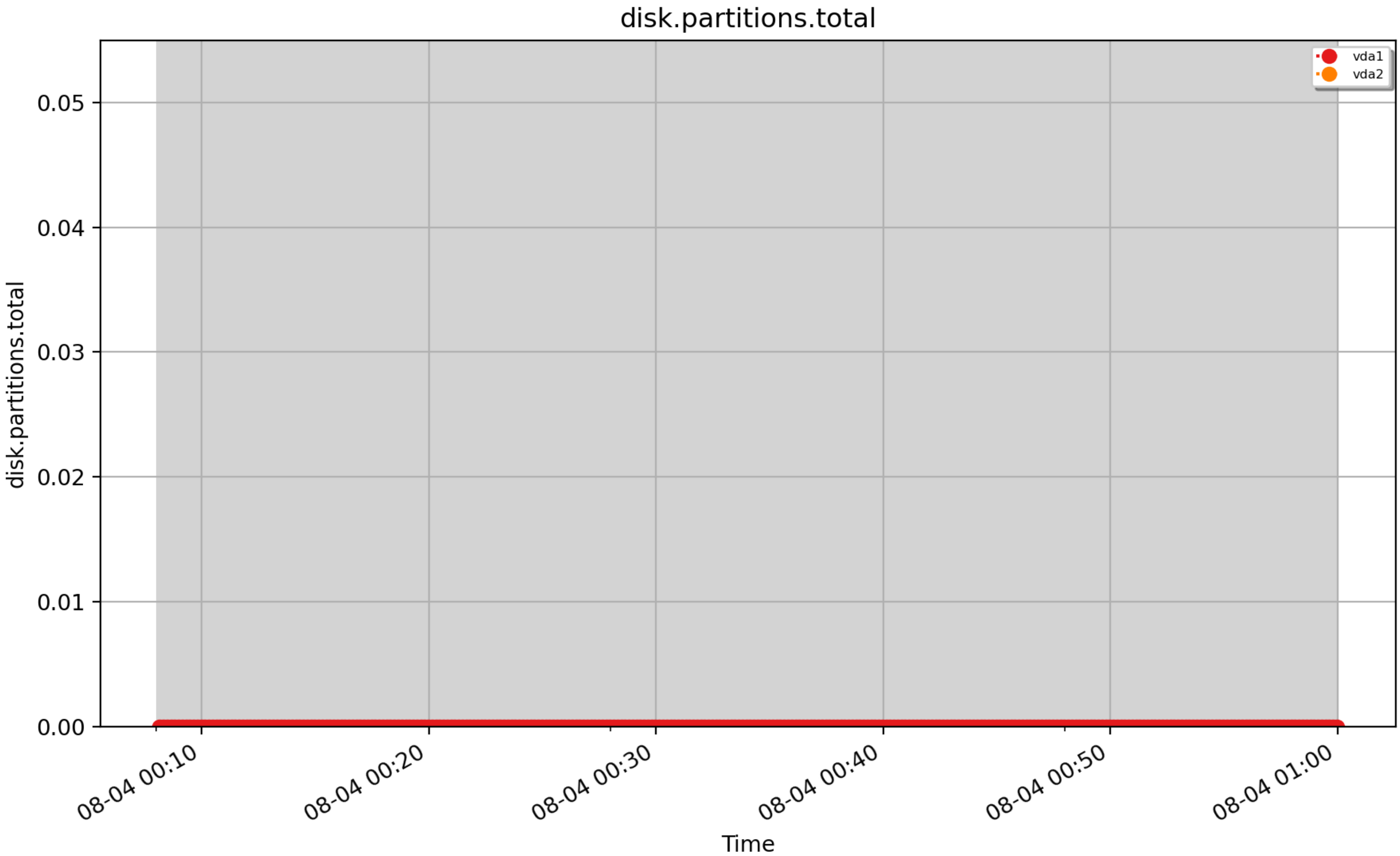


disk.partitions.read_merge: per-disk-partition count of merged read requests (count - U64) - *rate converted*

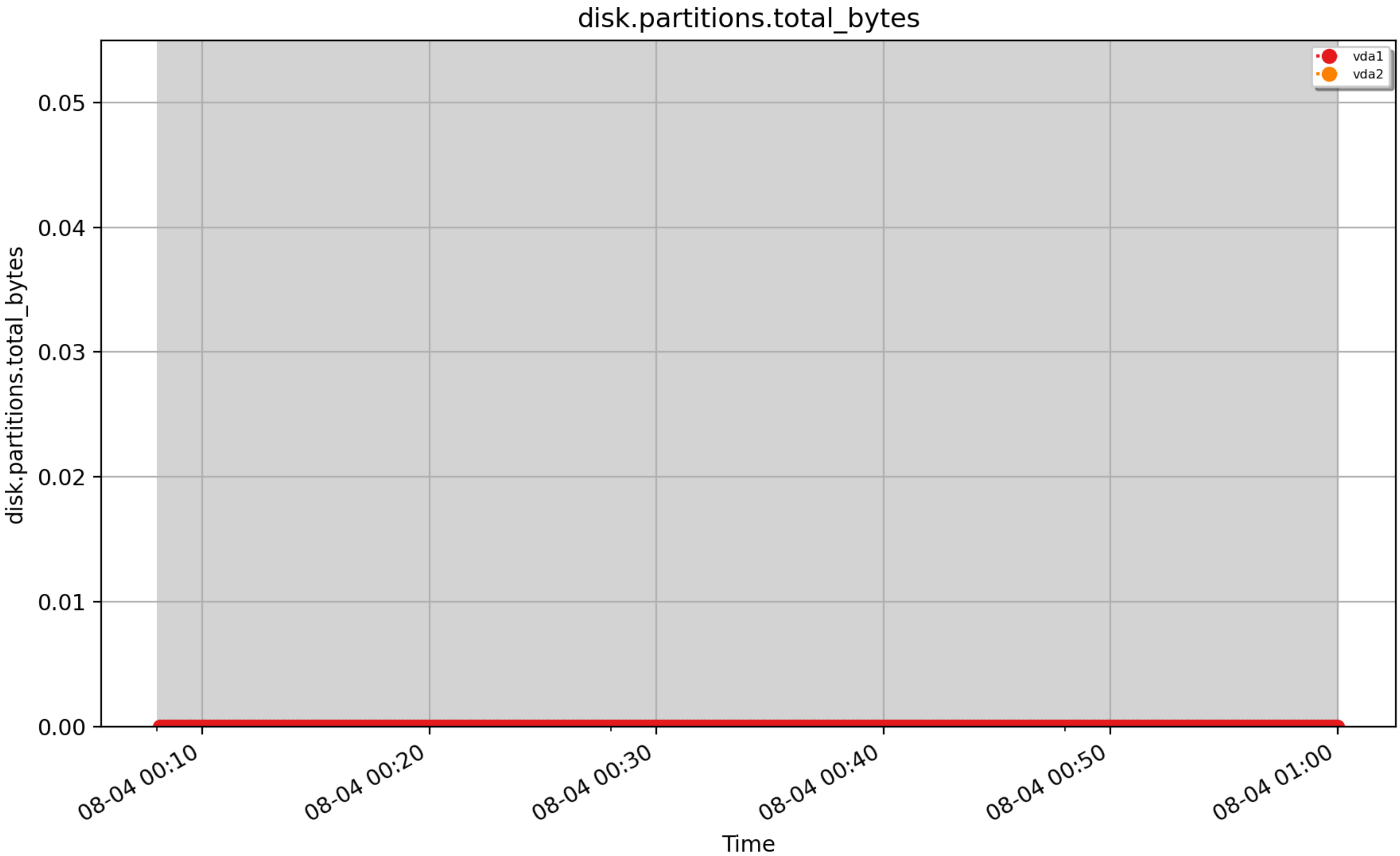


disk.partitions.read_rawactive: For each completed read on each disk partition the response time (queue time plus service time) in milliseconds is added to the associated instance of this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding reads for a disk partition. When divided by the number of completed reads for a disk partition (disk.partitions.read), the value represents

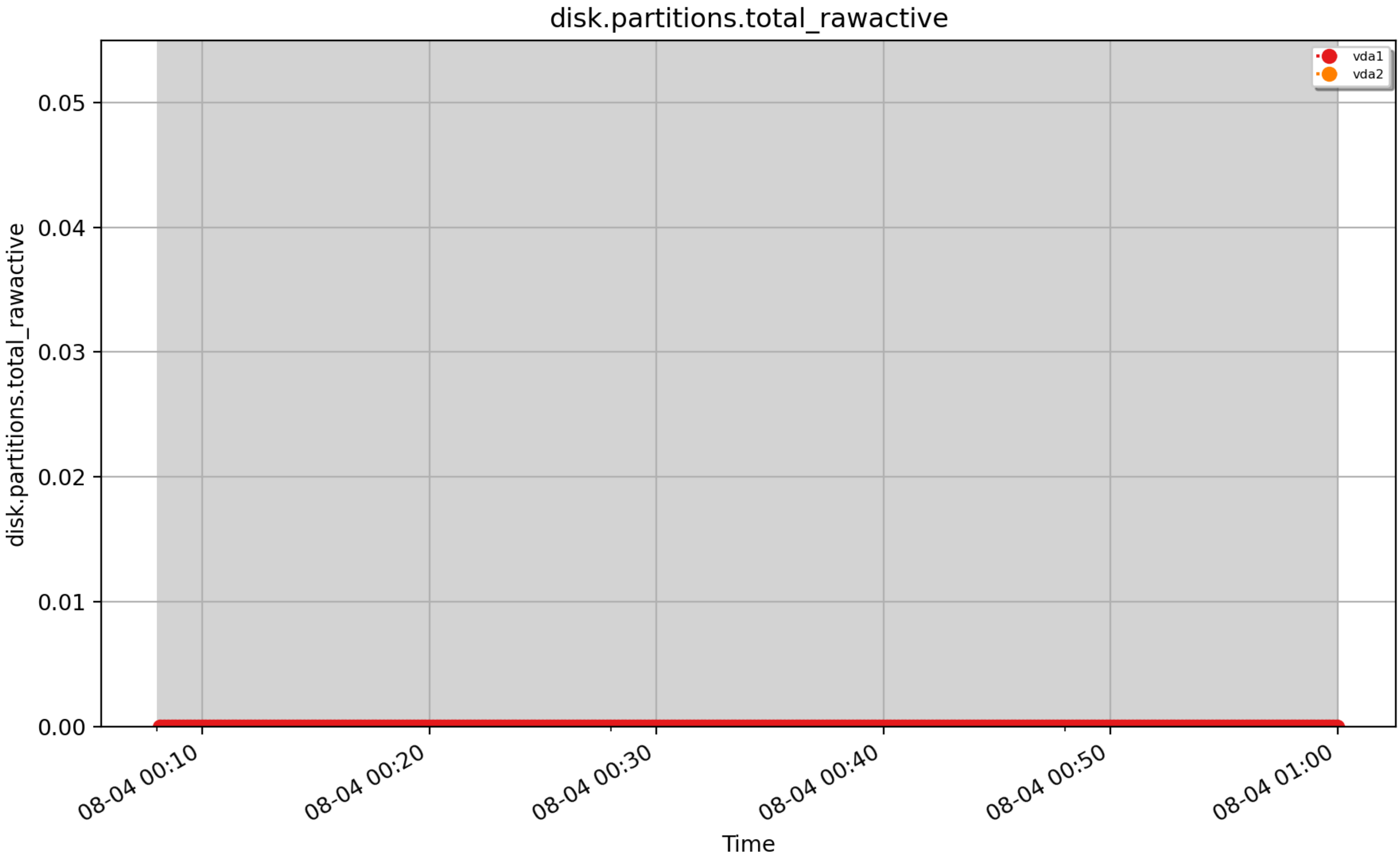
the stochastic average of the read response (or wait) time for that disk partition. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from existing performance tools:
iostat.partitions.r_await = delta(disk.partitions.read_rawactive) / delta(disk.partitions.read) (millisec - U32) - *rate converted*



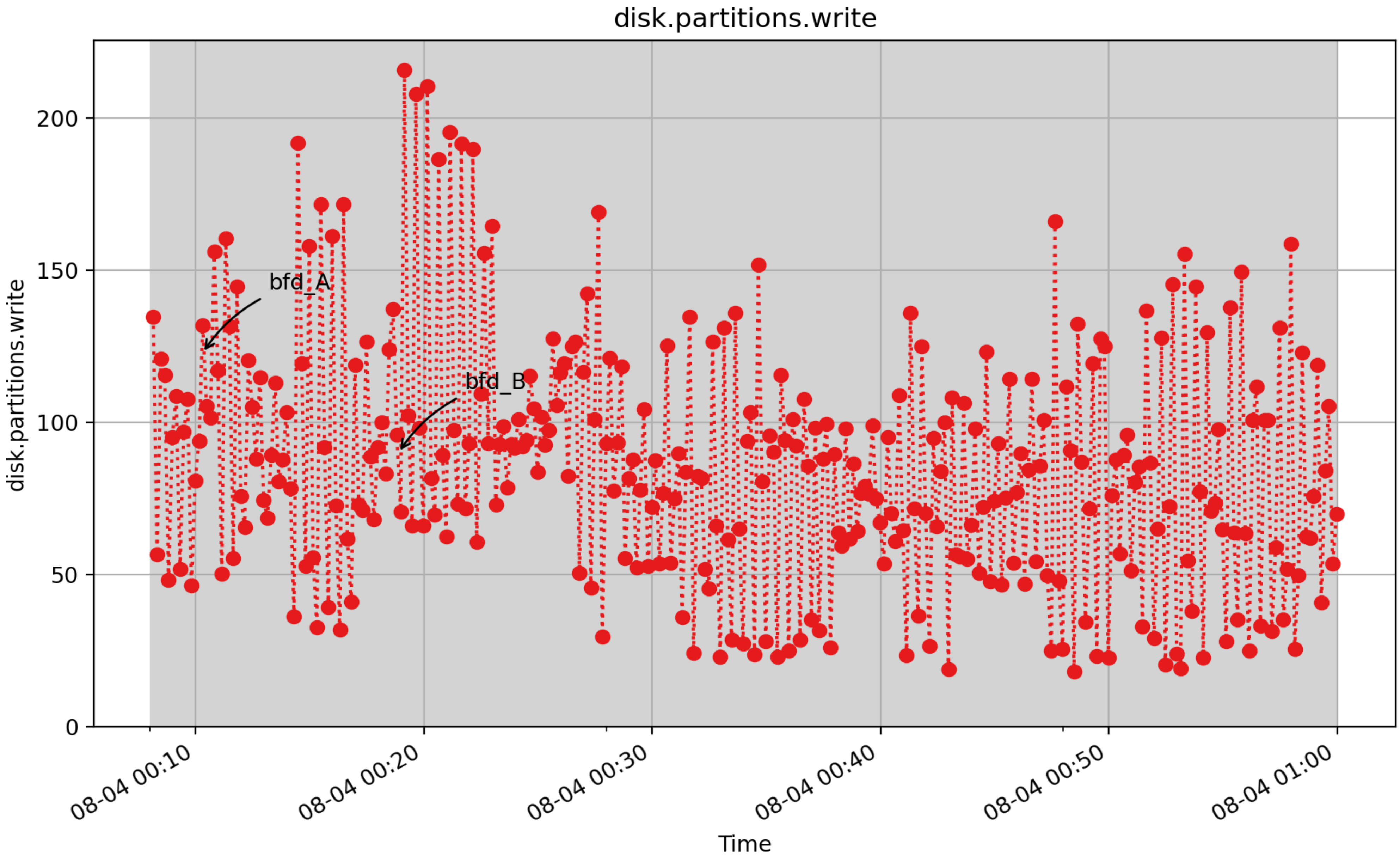
disk.partitions.total: Cumulative number of disk read and write operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U64) - *rate converted*



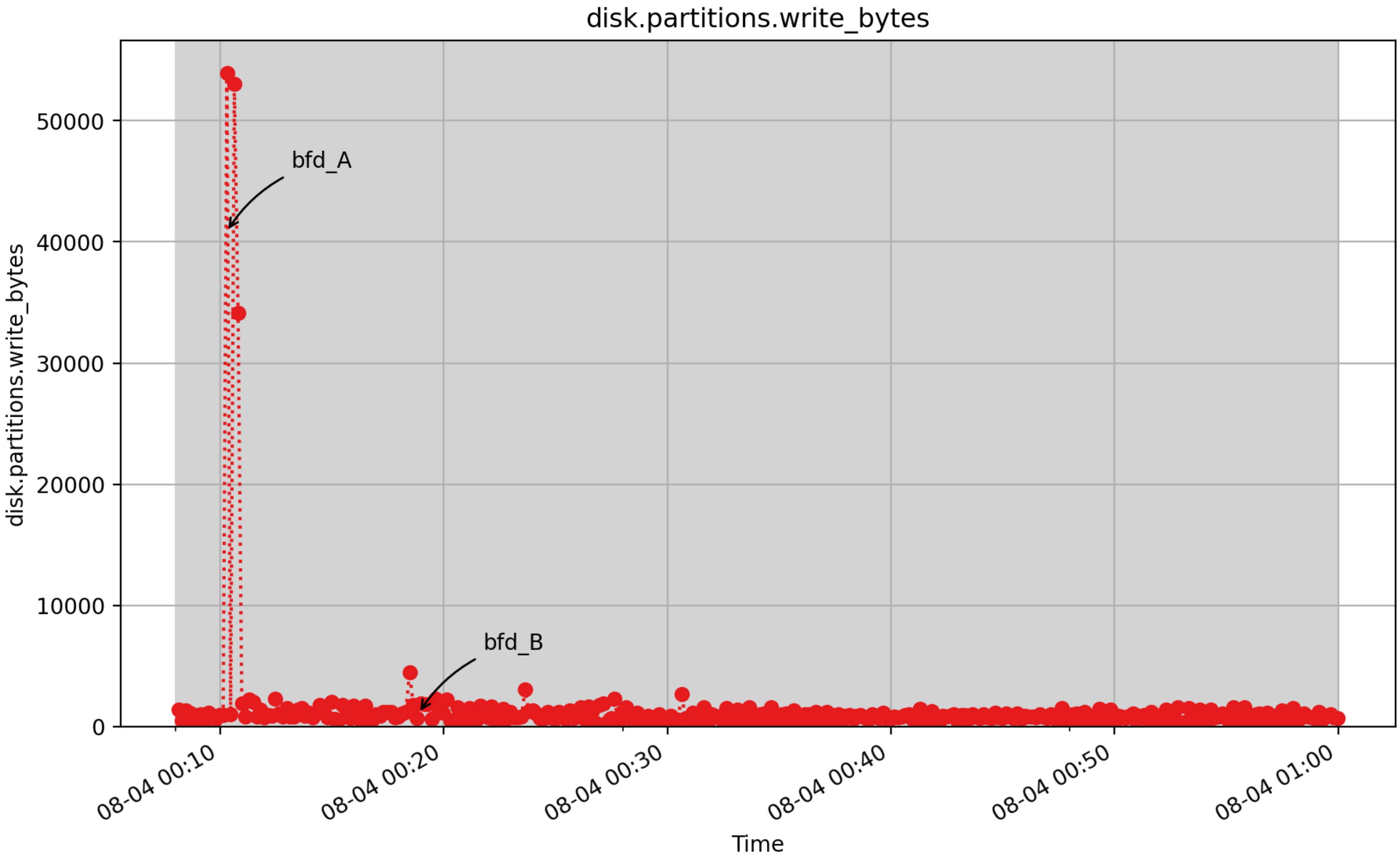
disk.partitions.total_bytes: Cumulative number of bytes read and written since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (Kbyte - U64) - *rate converted*



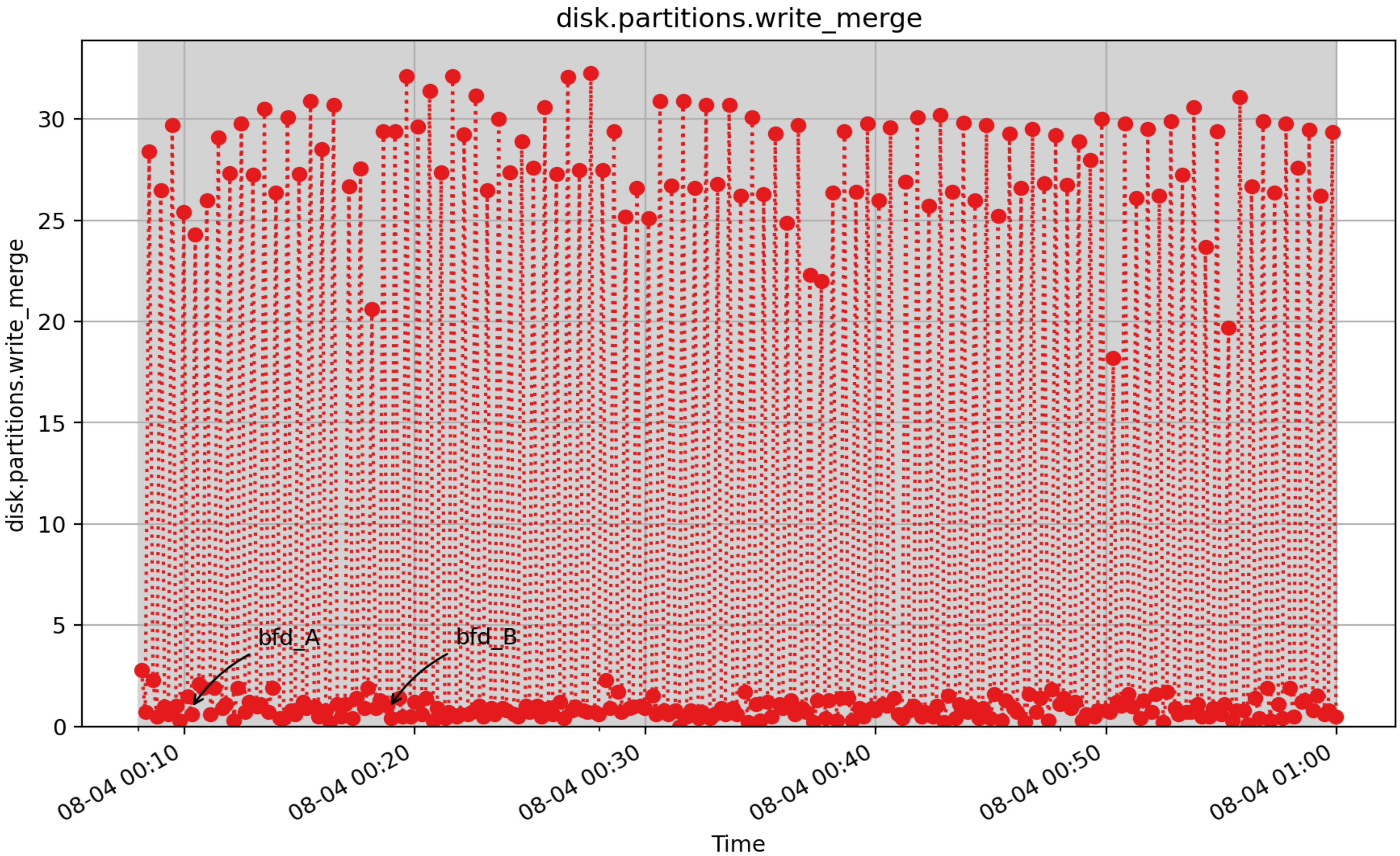
disk.partitions.total_rawactive: For each completed I/O on each disk partition the response time (queue time plus service time) in milliseconds is added to the associated instance of this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding I/Os for a disk partition. When divided by the number of completed I/Os for a disk partition (disk.partitions.total), the value represents the stochastic average of the I/O response (or wait) time for that disk partition. (millisec - U32) - *rate converted*



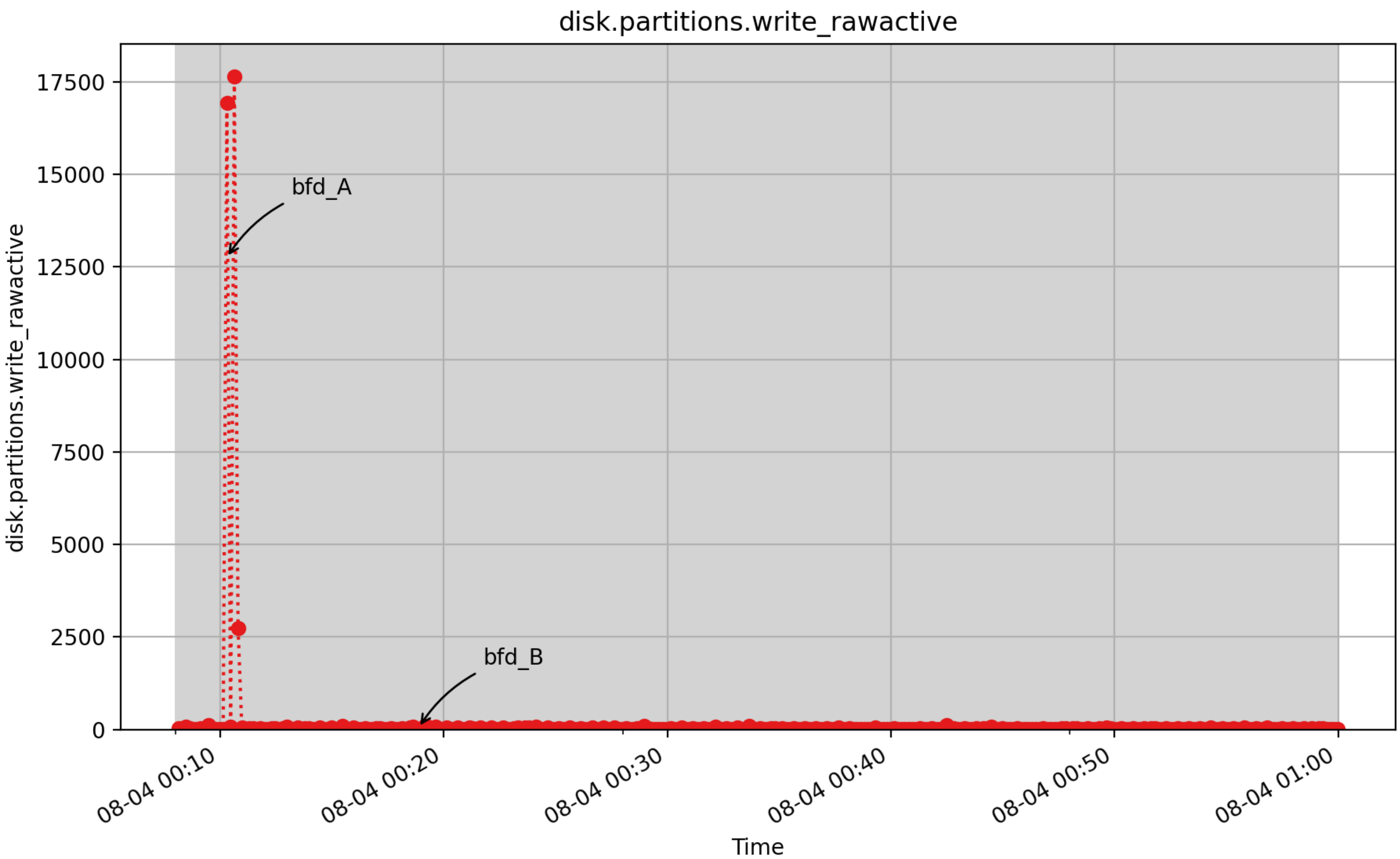
disk.partitions.write: Cumulative number of disk write operations since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (count - U64) - *rate converted*



disk.partitions.write_bytes: Cumulative number of bytes written since system boot time (subject to counter wrap) for individual disk partitions or logical volumes. (Kbyte - U64) - *rate converted*



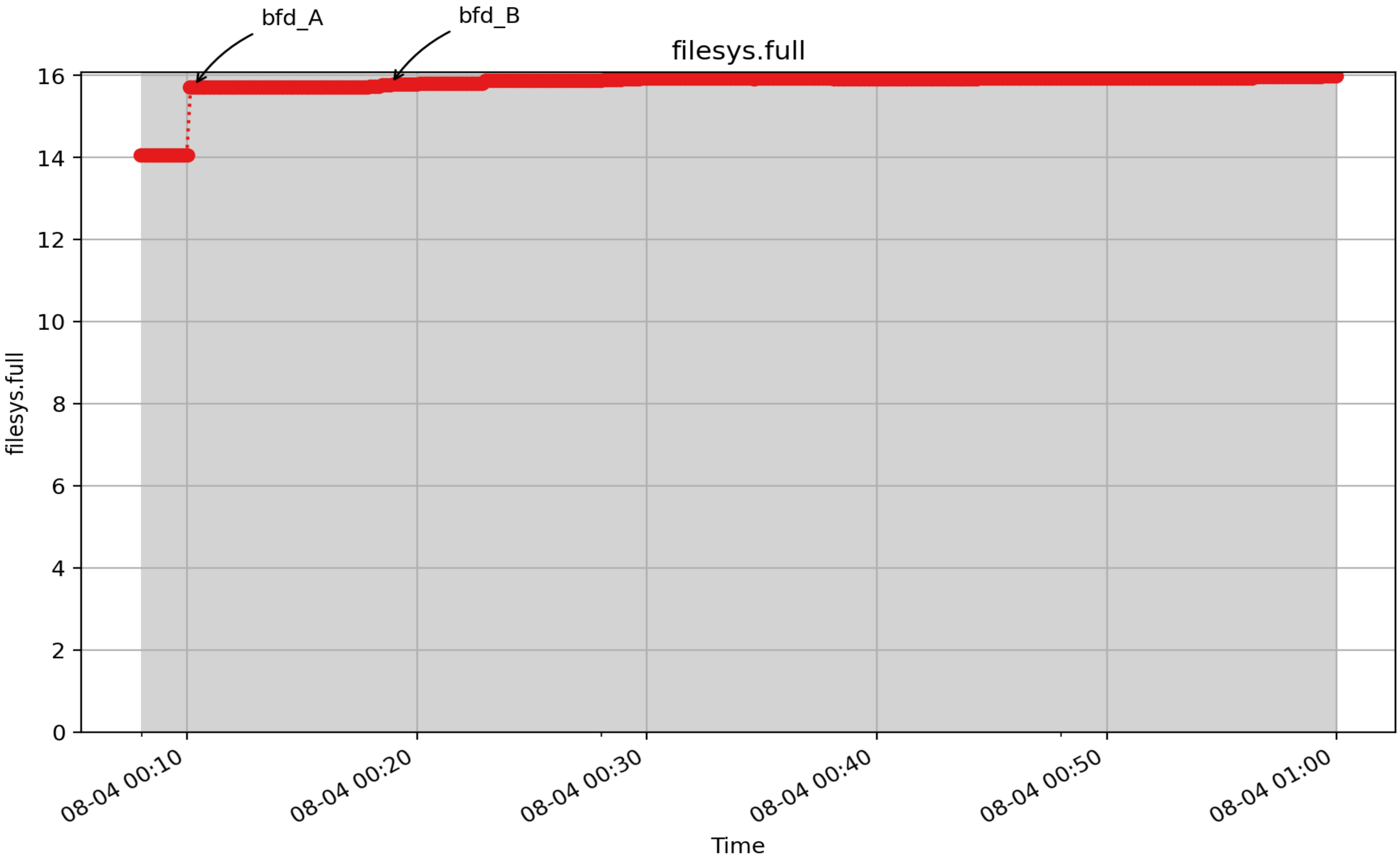
disk.partitions.write_merge: per-disk-partition count of merged write requests (count - U64) - *rate converted*



disk.partitions.write_rawactive: For each completed write on each disk partition the response time (queue time plus service time) in milliseconds is added to the associated instance of this metric. When converted to a normalized rate, the value represents the time average of the number of outstanding writes for a disk partition. When divided by the number of completed writes for a disk partition (disk.partitions.write), the value represents

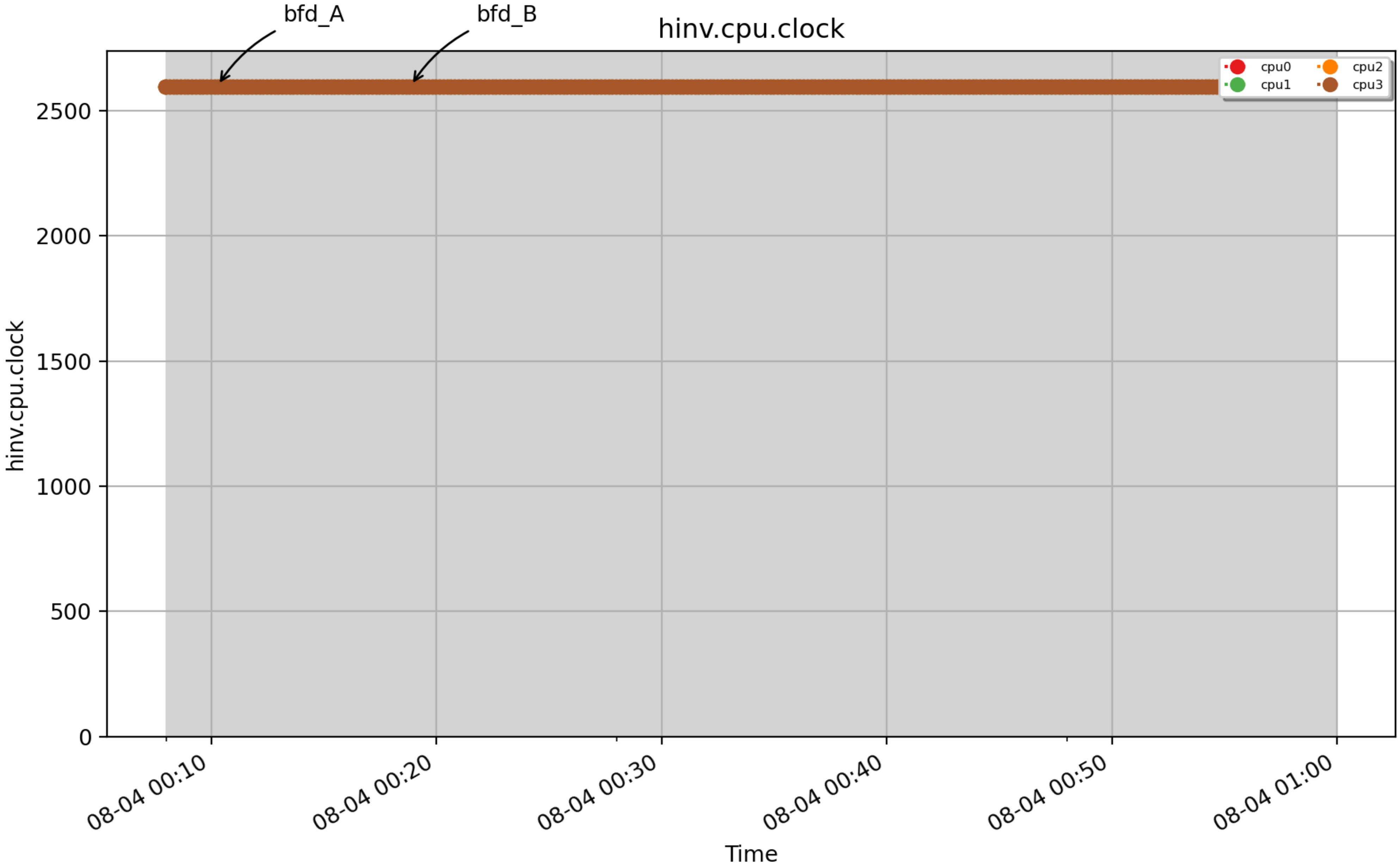
the stochastic average of the write response (or wait) time for that disk partition. It is suitable mainly for use in calculations with other metrics, e.g. mirroring the results from existing performance tools:
iostat.partitions.w_await = delta(disk.partitions.write_rawactive) / delta(disk.partitions.write) (millisec - U32) - *rate converted*

Filesys

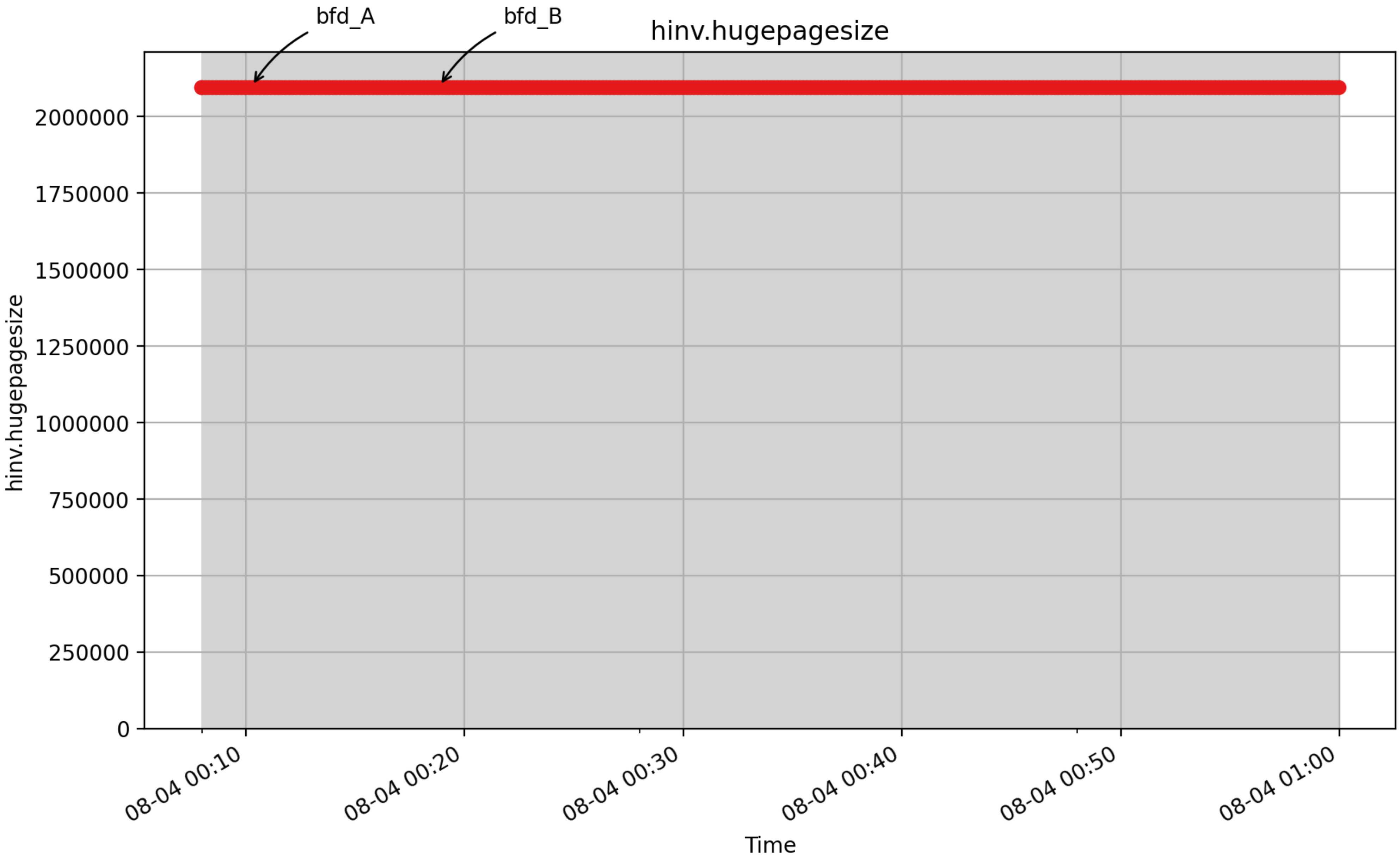


filesys.full: Percentage of filesystem in use (- DOUBLE)

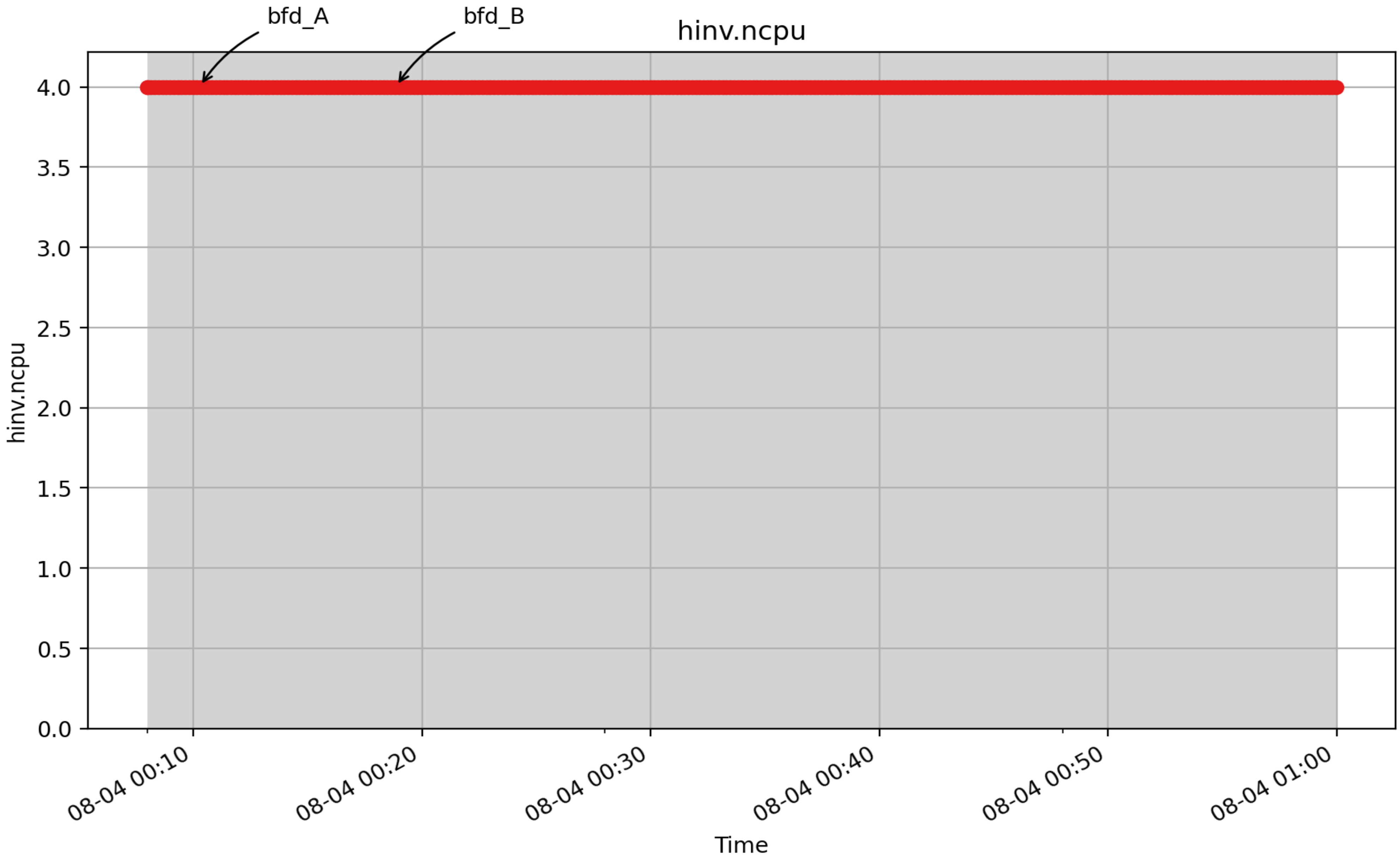
Hinv



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

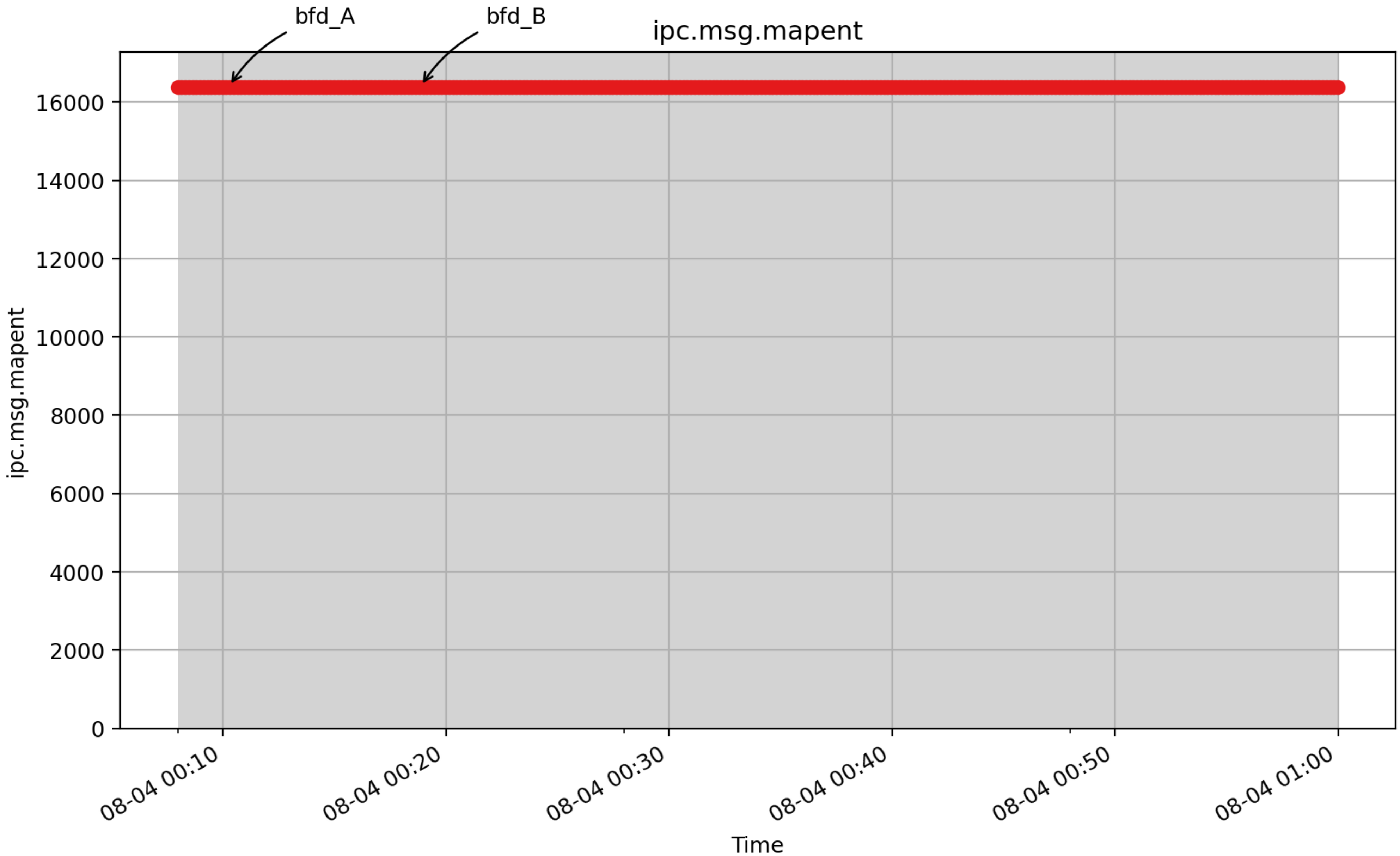


hinv.hugepagesize: The memory huge page size of the running kernel in bytes. (byte - U32)

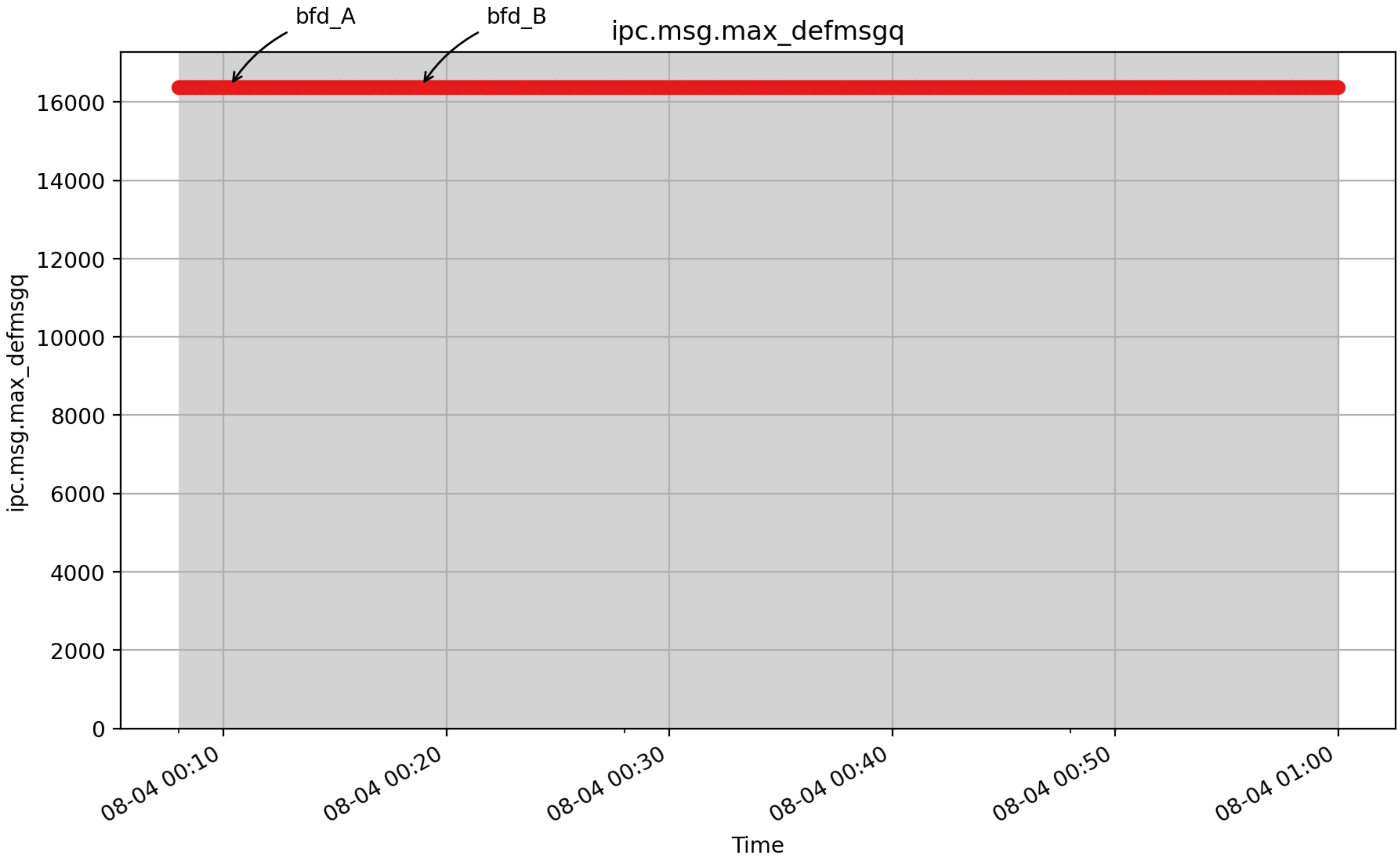


hinv.ncpu: number of CPUs in the system (- U32)

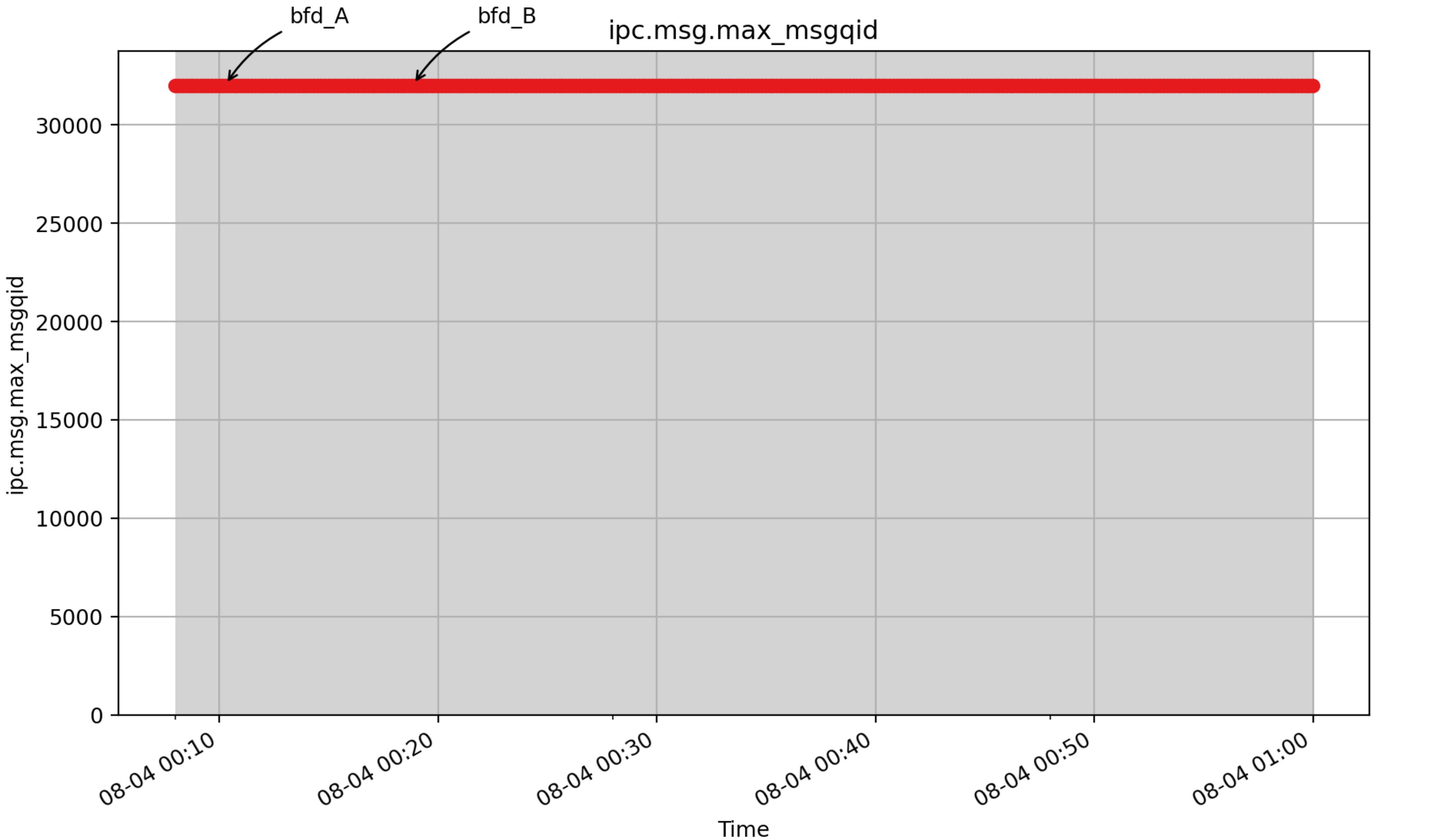
Ipc



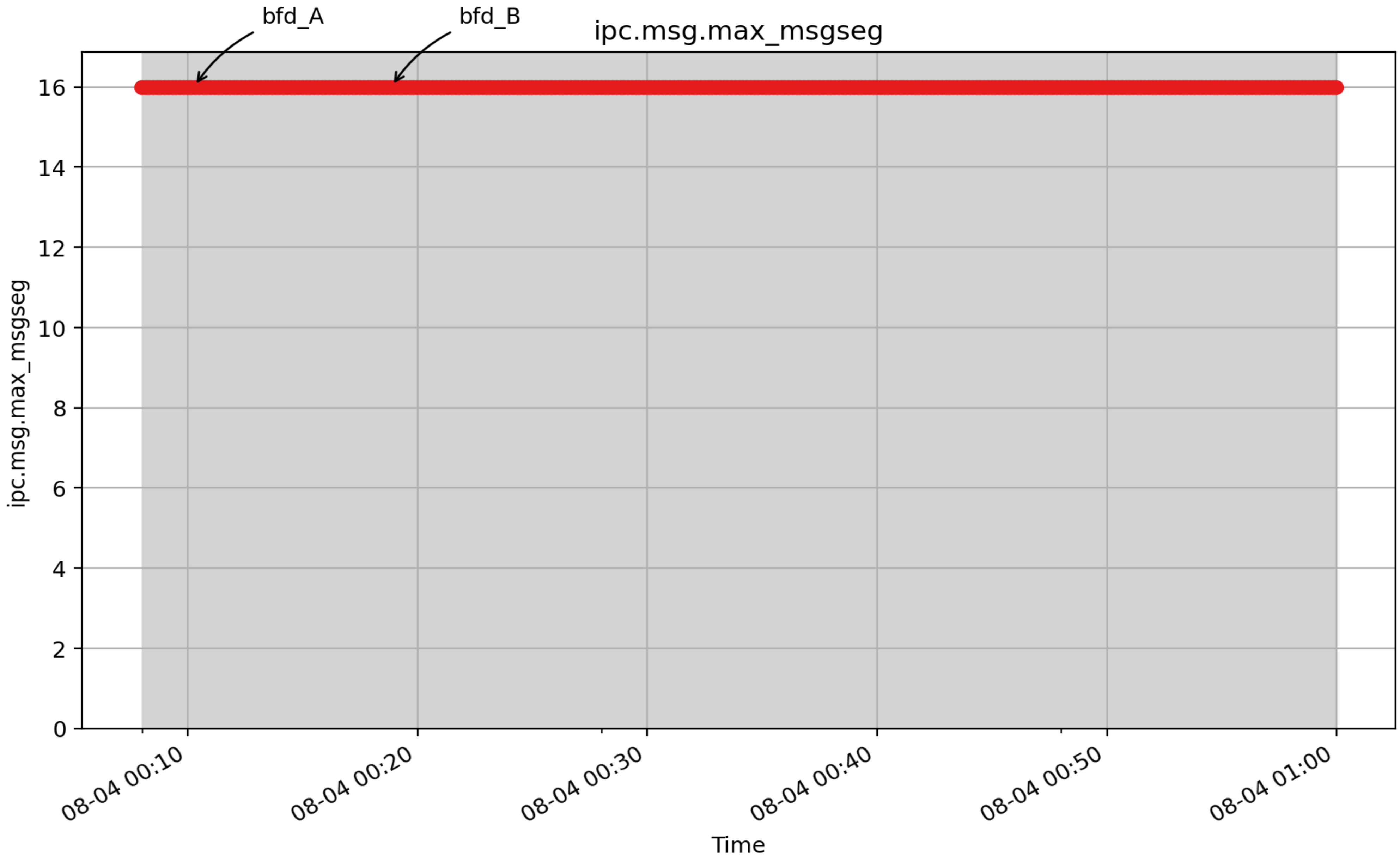
ipc.msg.mapent: number of entries in a message map (from msgctl(..,IPC_INFO,..)) (- U32)



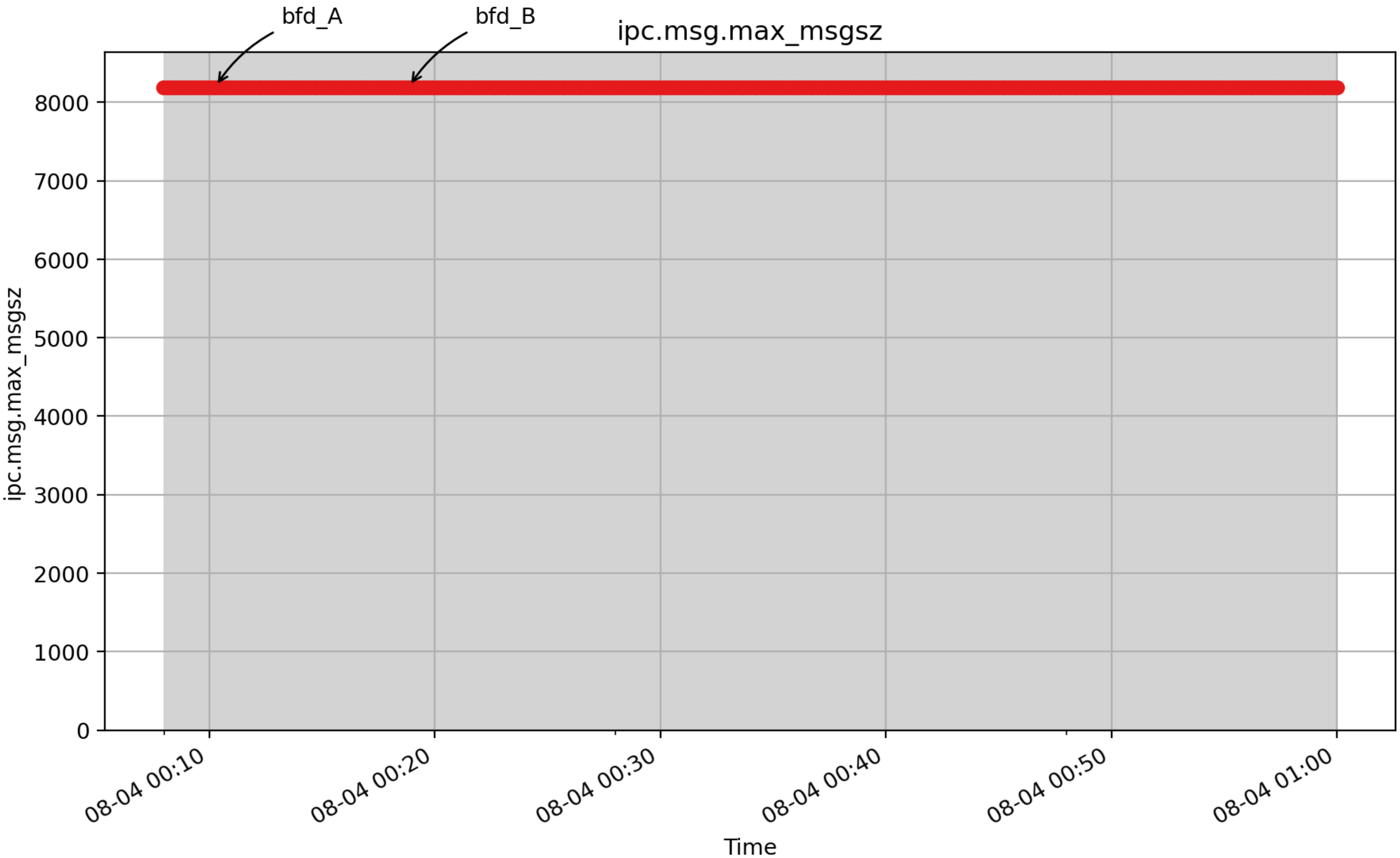
ipc.msg.max_defmsgq: default maximum size of a message queue (from msgctl(..,IPC_INFO,..)) (- U32)



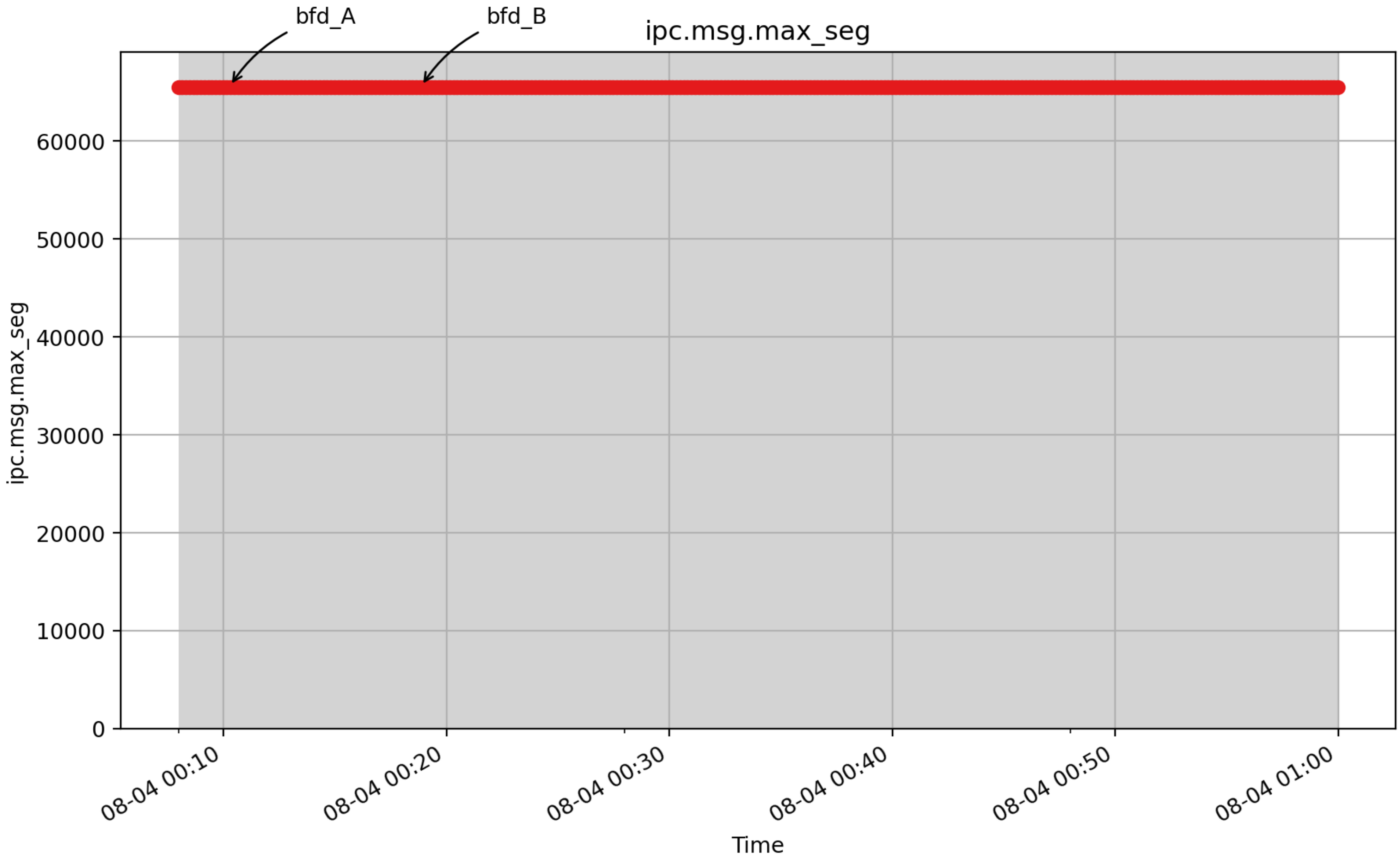
ipc.msg.max_msgqid: maximum number of message queue identifiers (from msgctl(..,IPC_INFO,..)) (- U32)



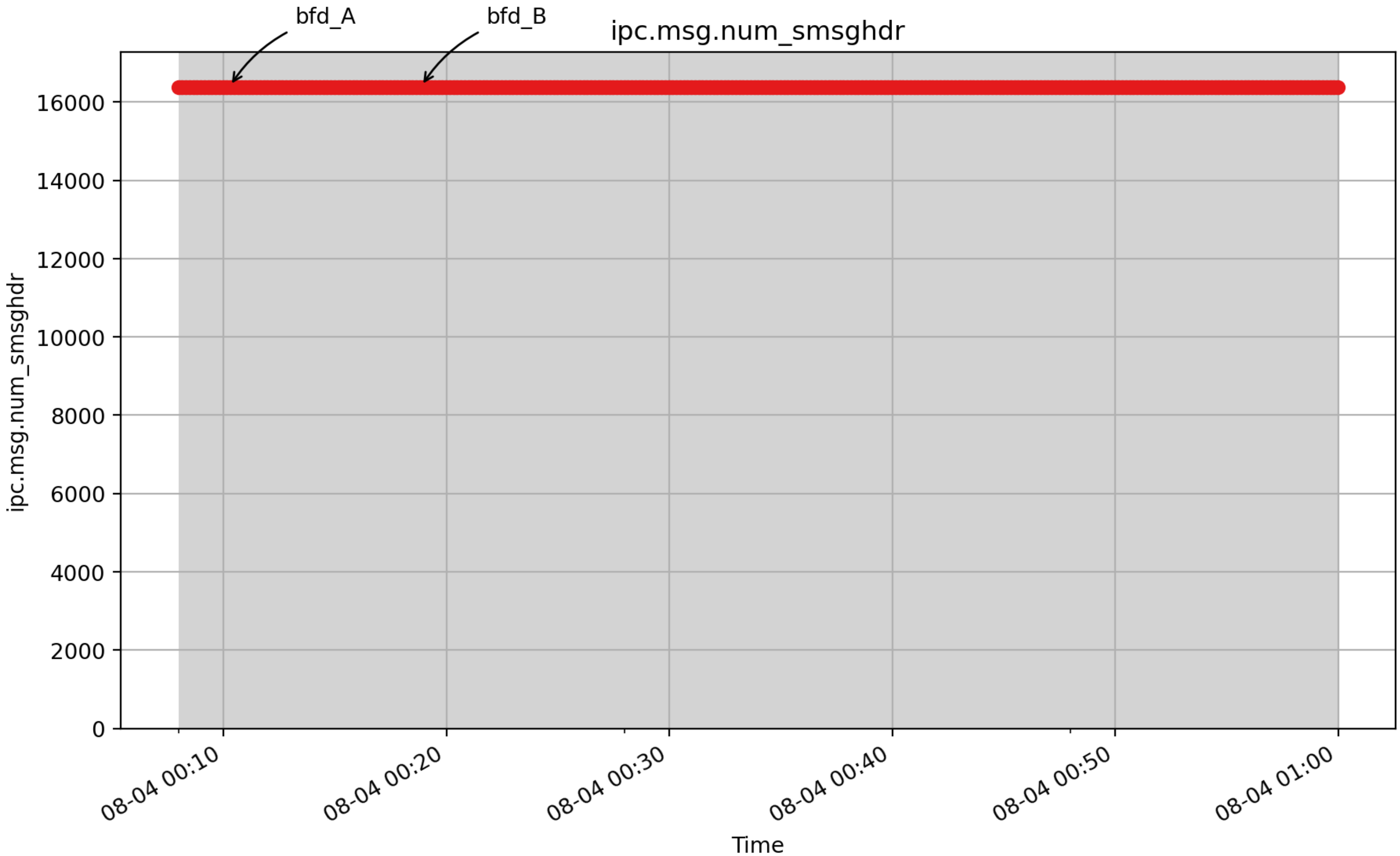
ipc.msg.max_msgseg: message segment size (from msgctl(..,IPC_INFO,..)) (- U32)



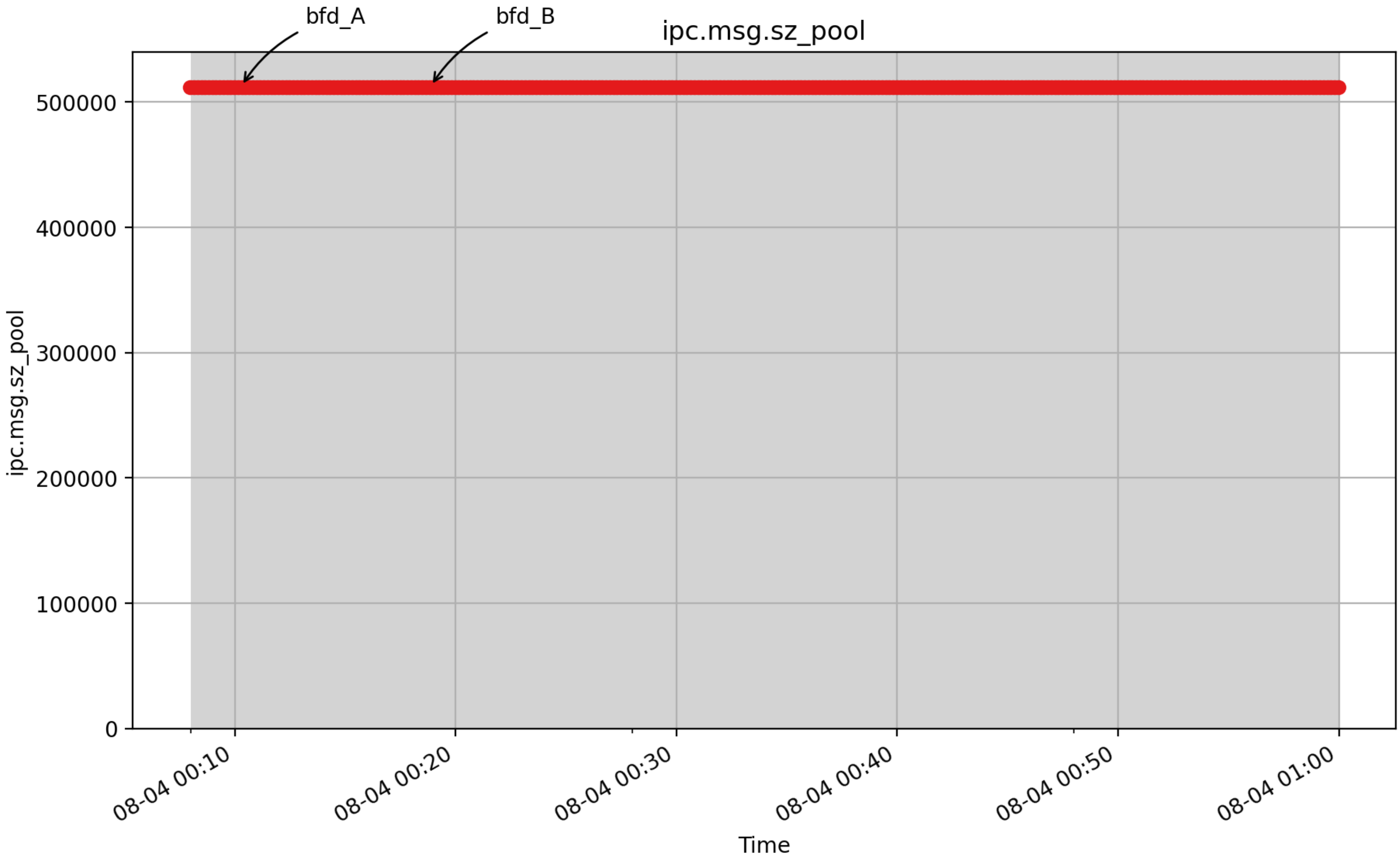
ipc.msg.max_msgsz: maximum size of a message in bytes (from msgctl(..,IPC_INFO,..)) (- U32)



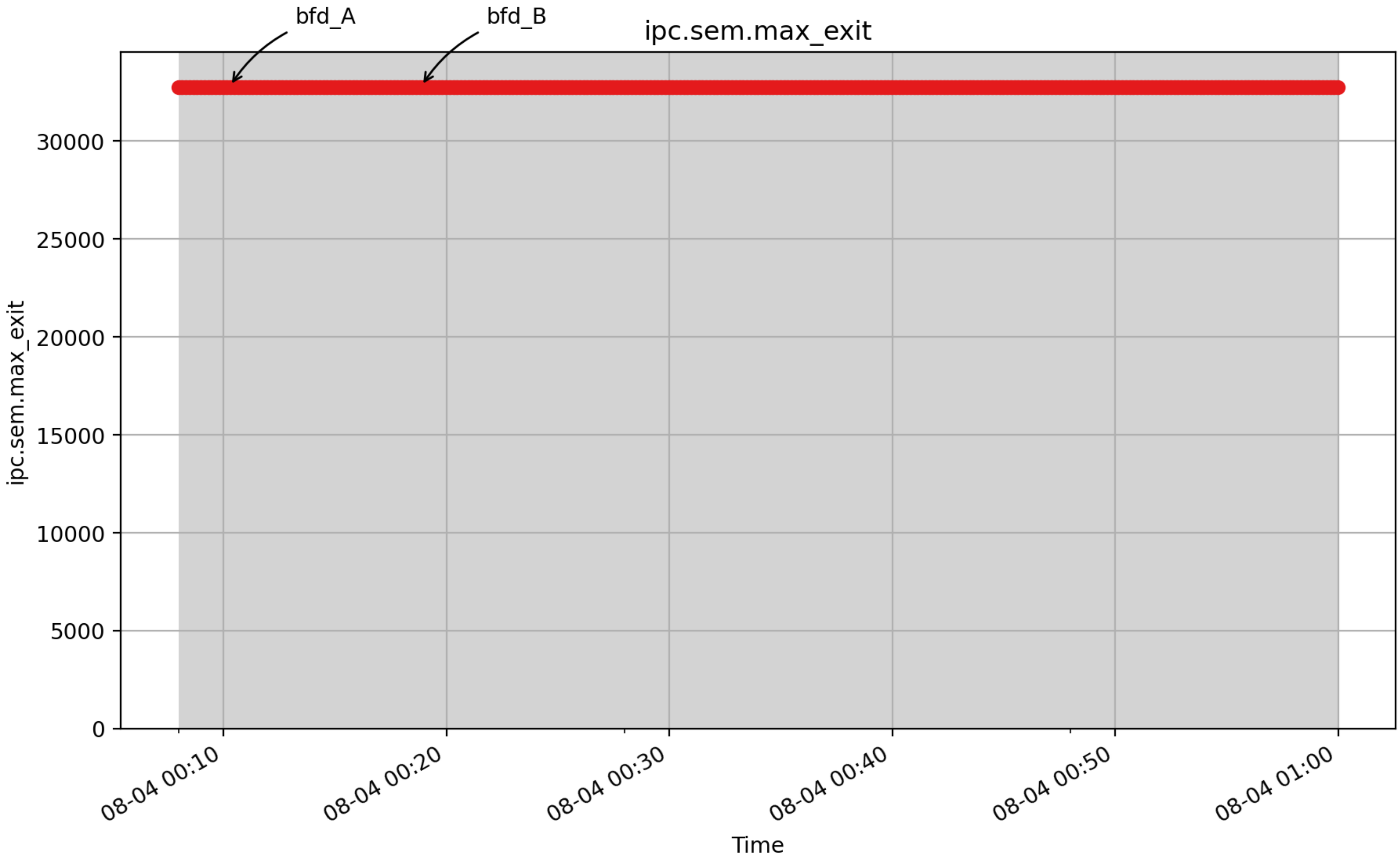
ipc.msg.max_seg: maximum number of message segments (from msgctl(..,IPC_INFO,..)) (- U32)



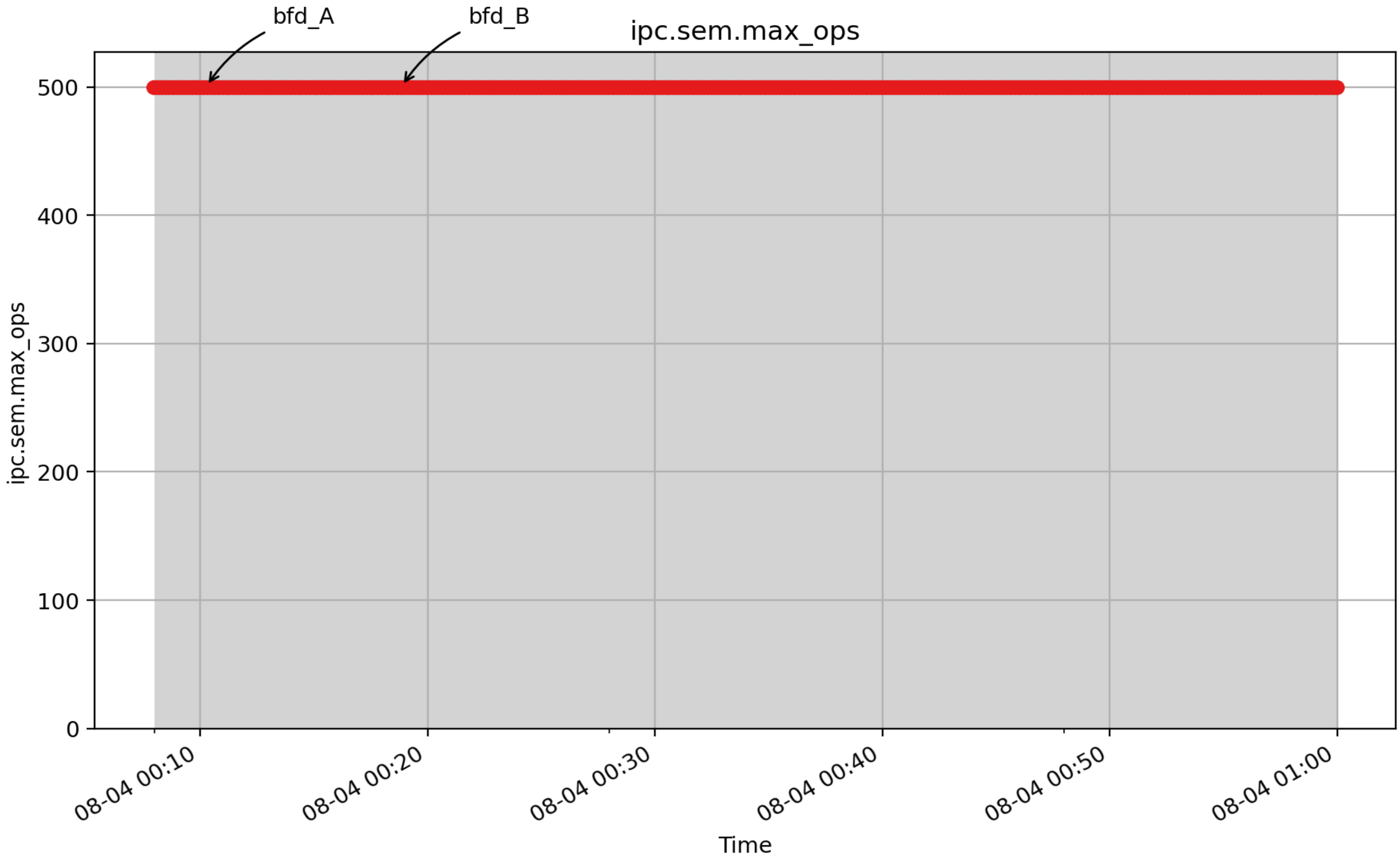
ipc.msg.num_smsghdr: number of system message headers (from msgctl(..,IPC_INFO,..)) (- U32)



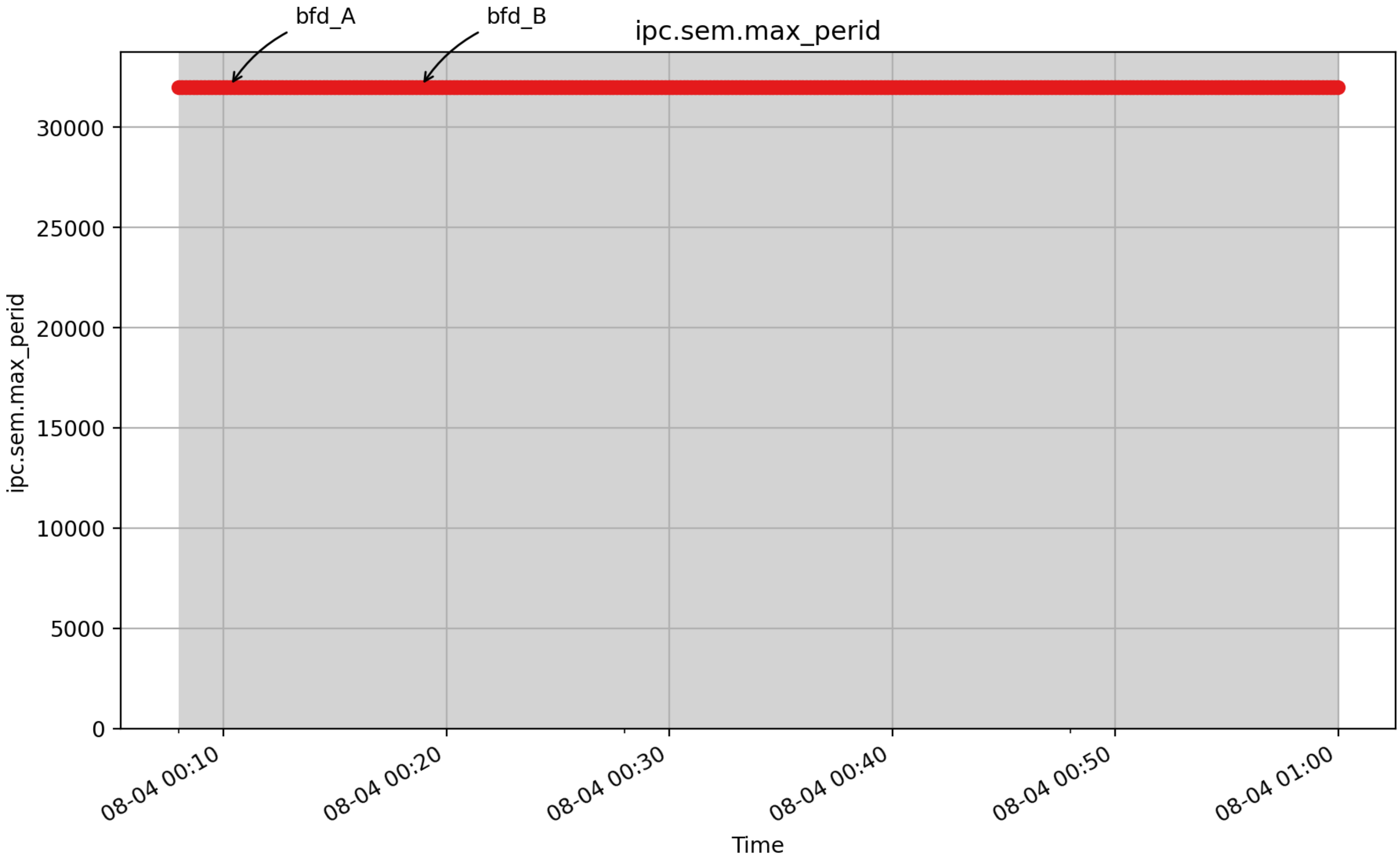
ipc.msg.sz_pool: size of message pool in kilobytes (from msgctl(..,IPC_INFO,..)) (Kbyte - U32)



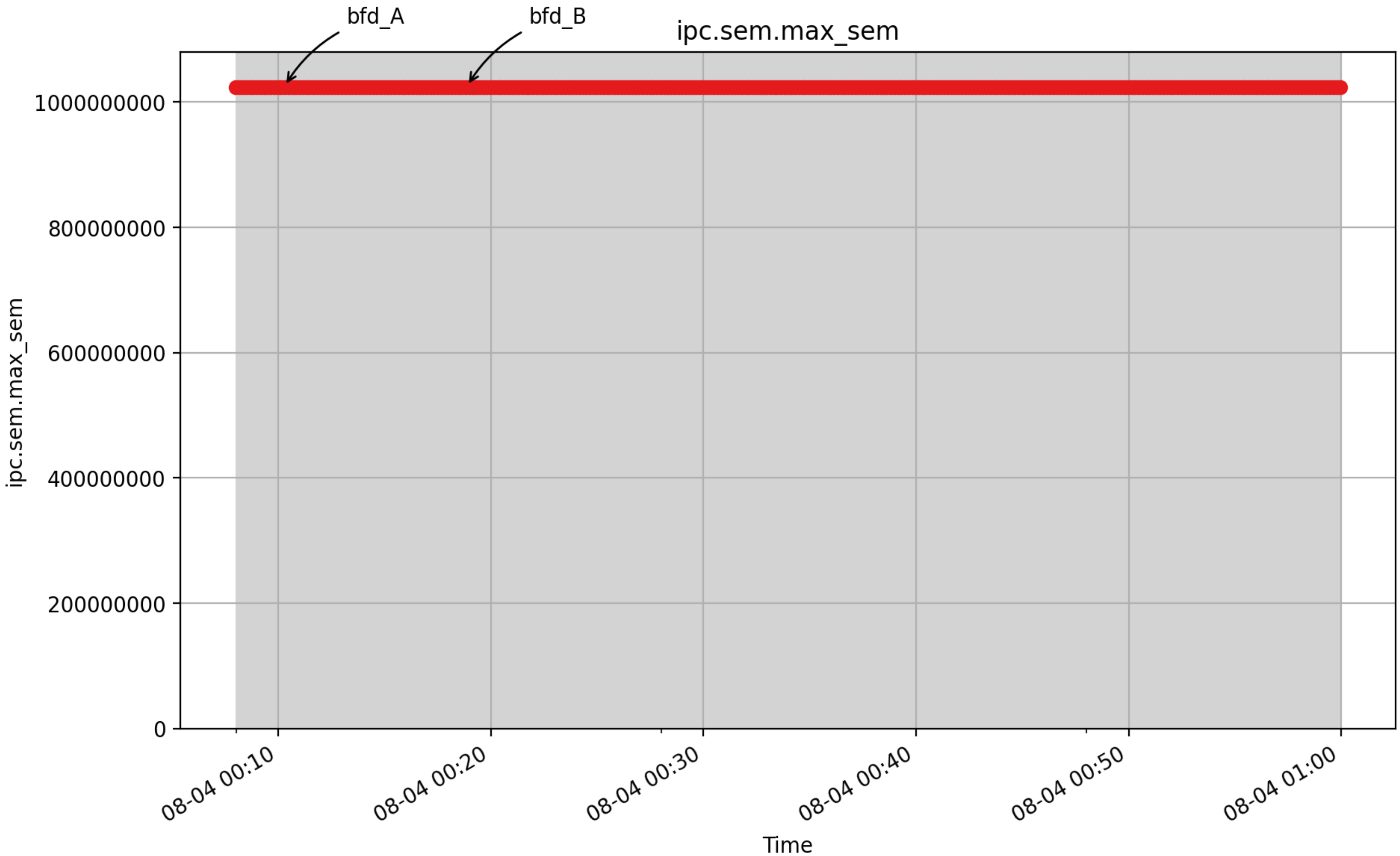
ipc.sem.max_exit: adjust on exit maximum value (from semctl(..,IPC_INFO,..)) (- U32)



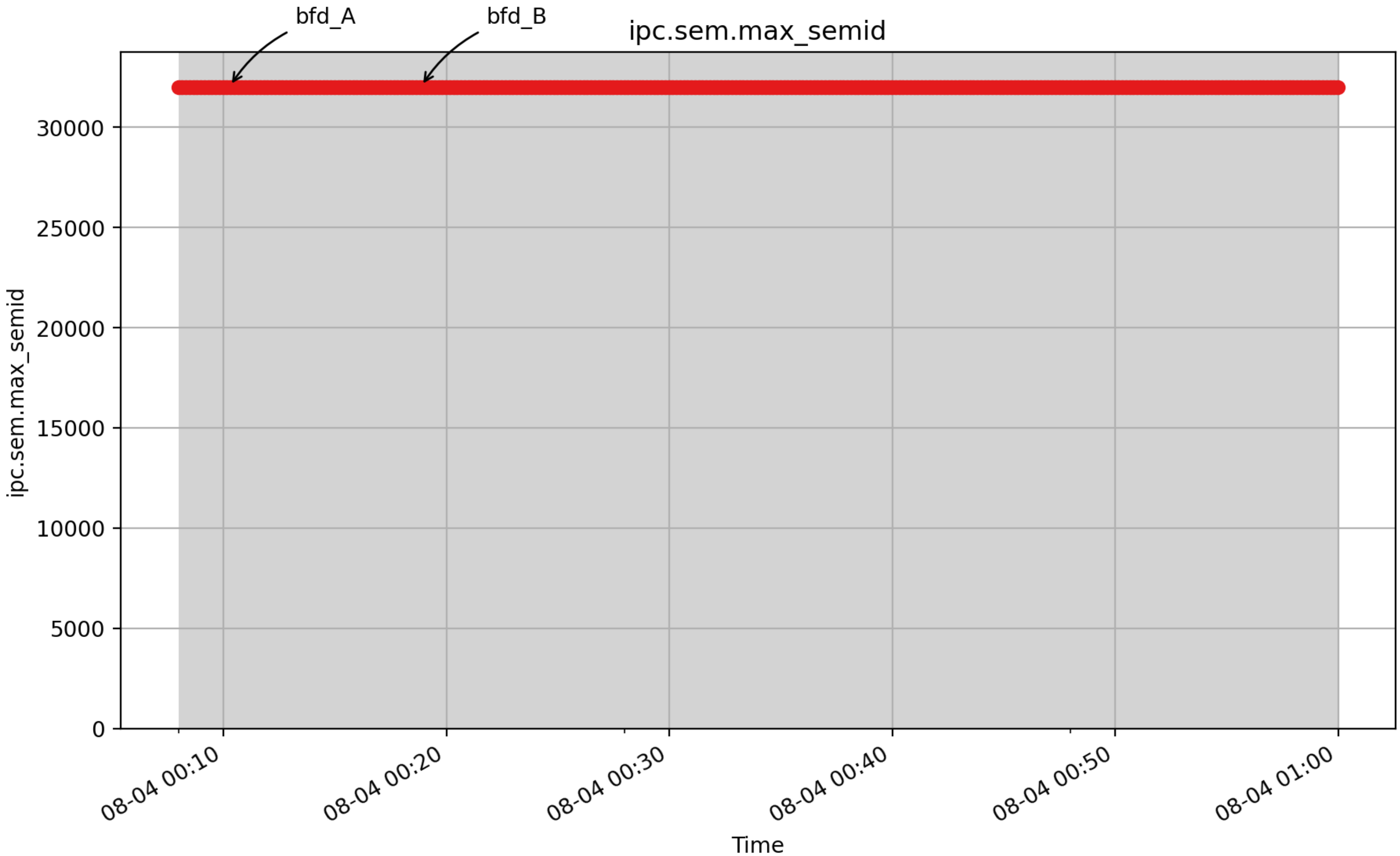
ipc.sem.max_ops: maximum number of operations per semop call (from semctl(..,IPC_INFO,..)) (- U32)



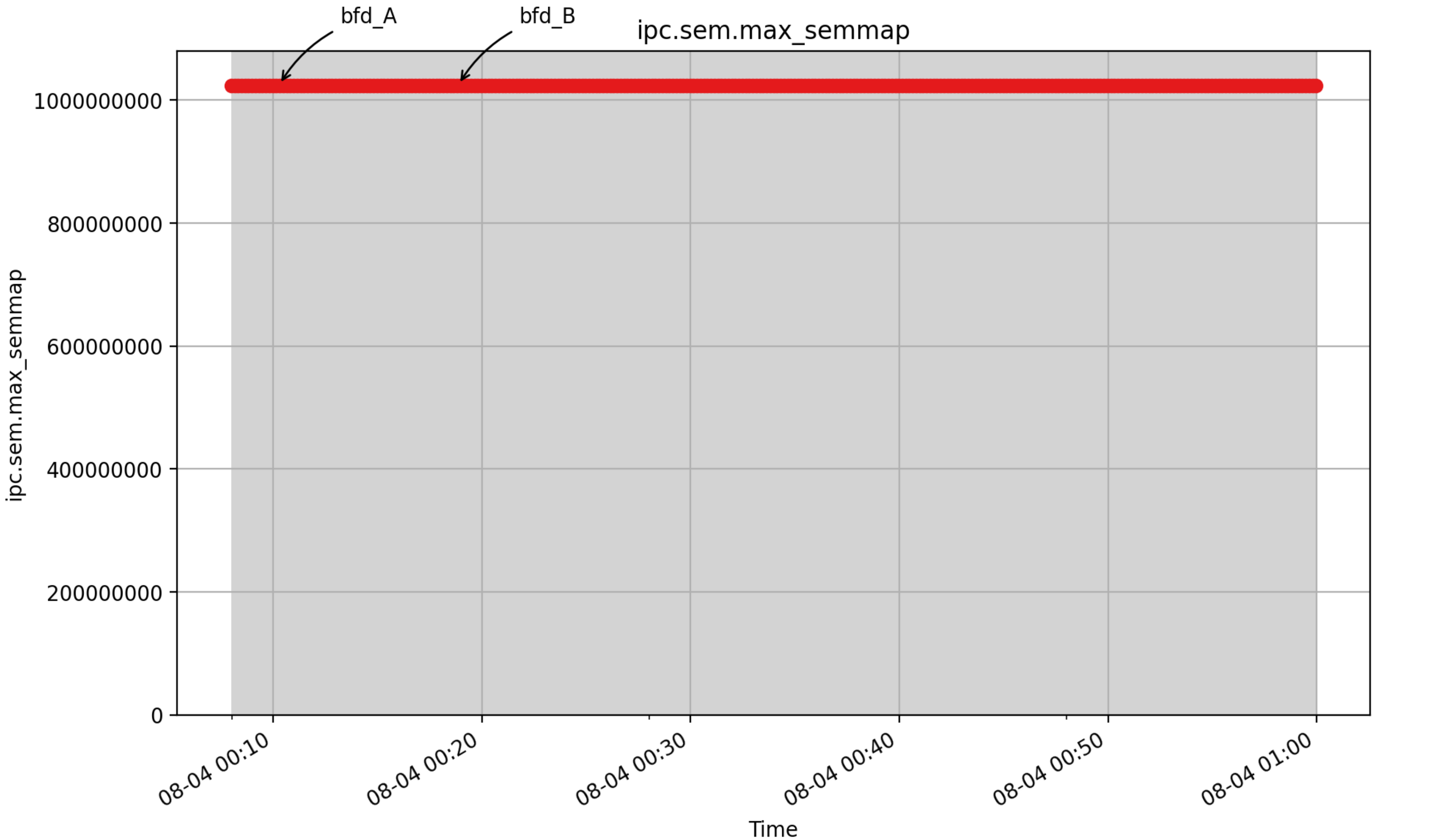
ipc.sem.max_perid: maximum number of semaphores per identifier (from semctl(..,IPC_INFO,..)) (- U32)



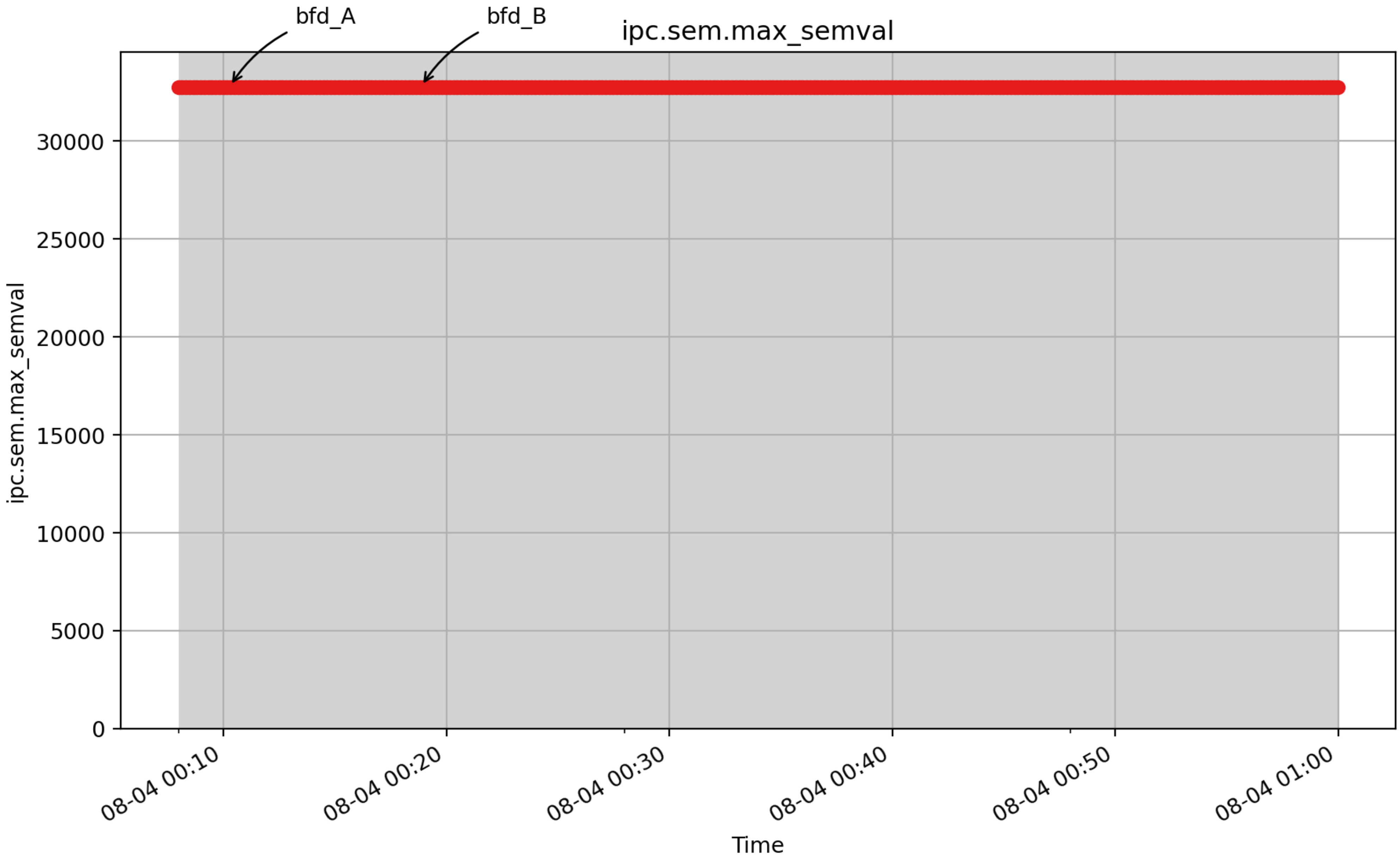
ipc.sem.max_sem: maximum number of semaphores in system (from semctl(..,IPC_INFO,..)) (- U32)



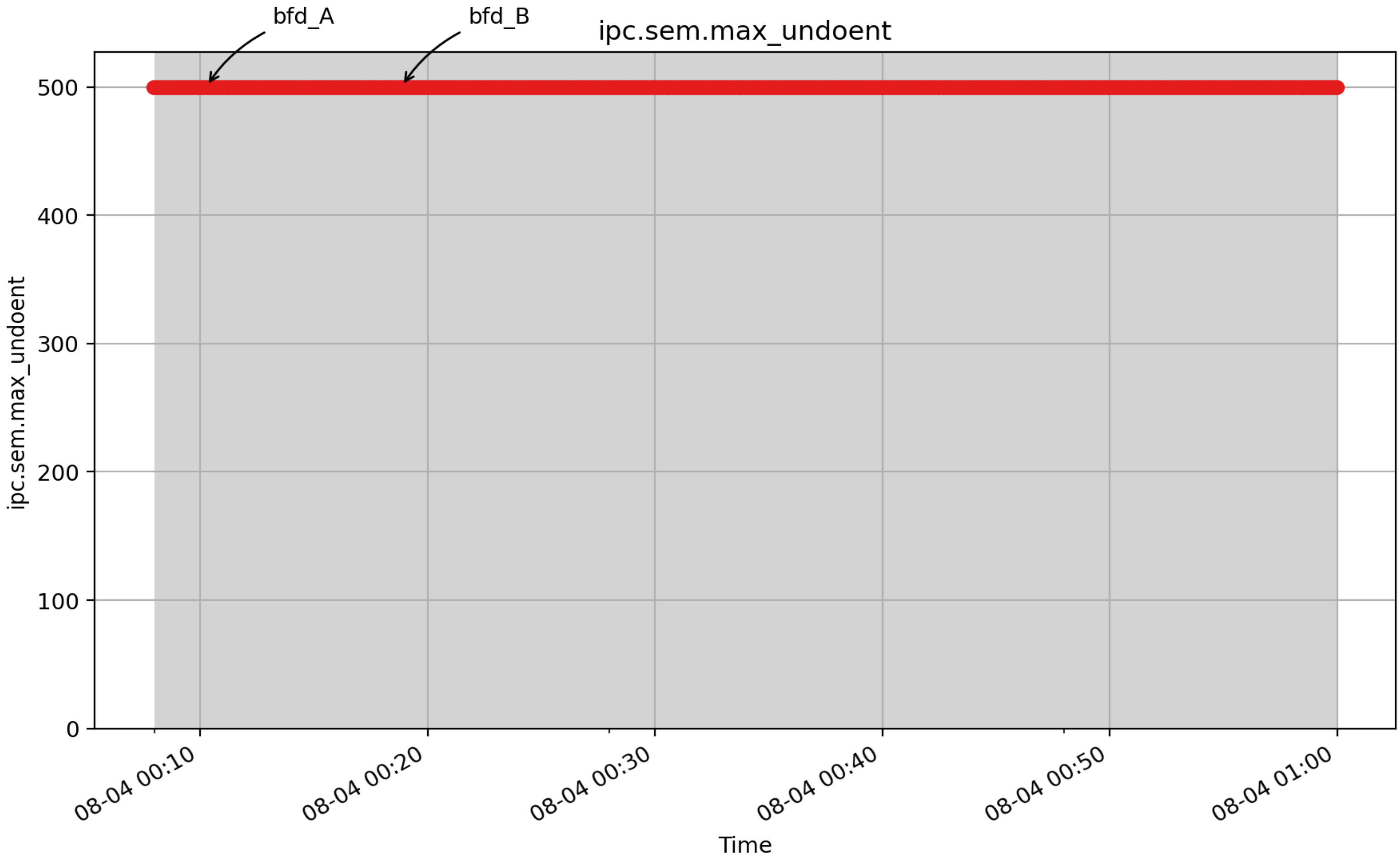
ipc.sem.max_semid: maximum number of semaphore identifiers (from semctl(..,IPC_INFO,..)) (- U32)



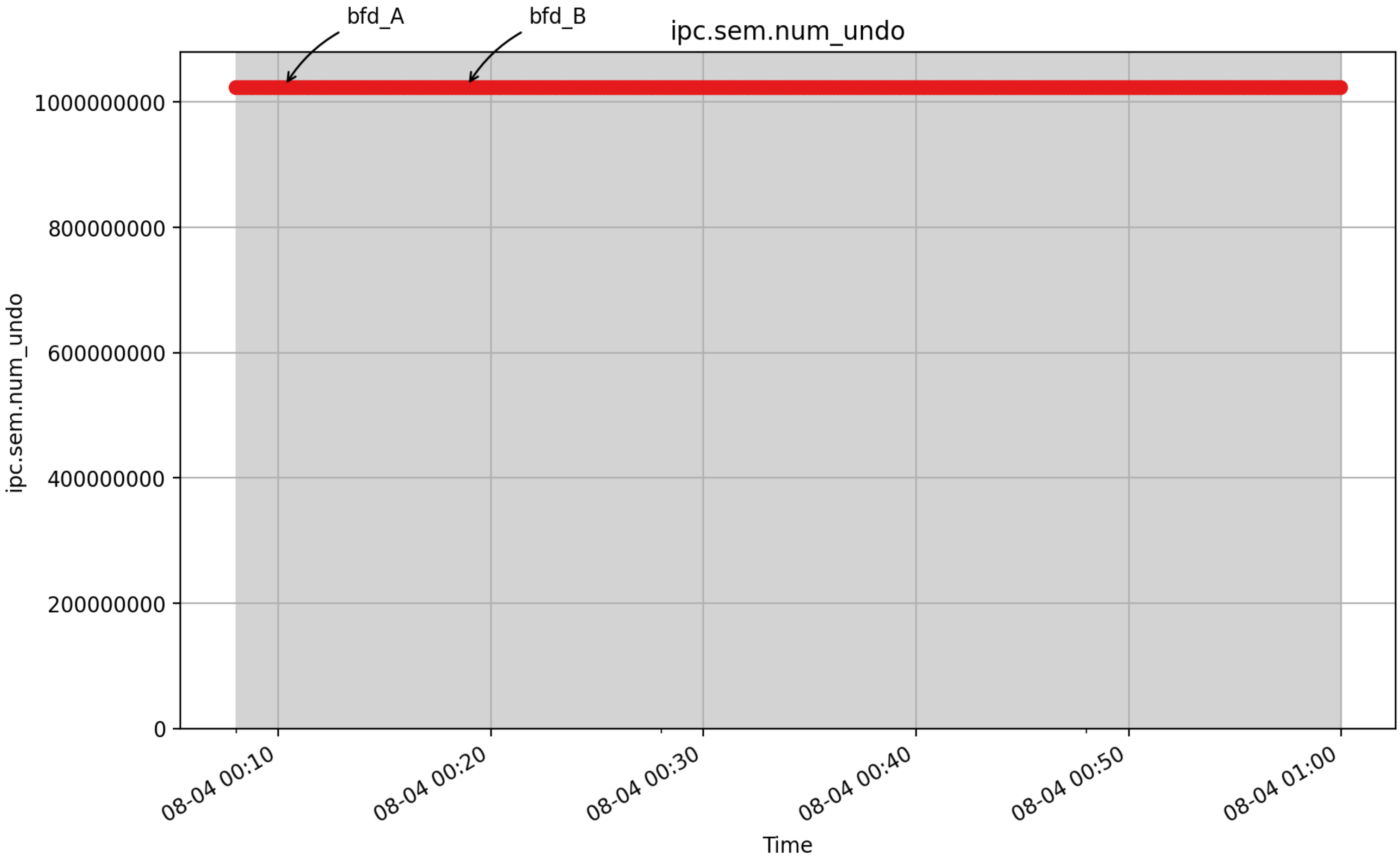
ipc.sem.max_semmap: maximum number of entries in a semaphore map (from semctl(..,IPC_INFO,..)) (- U32)



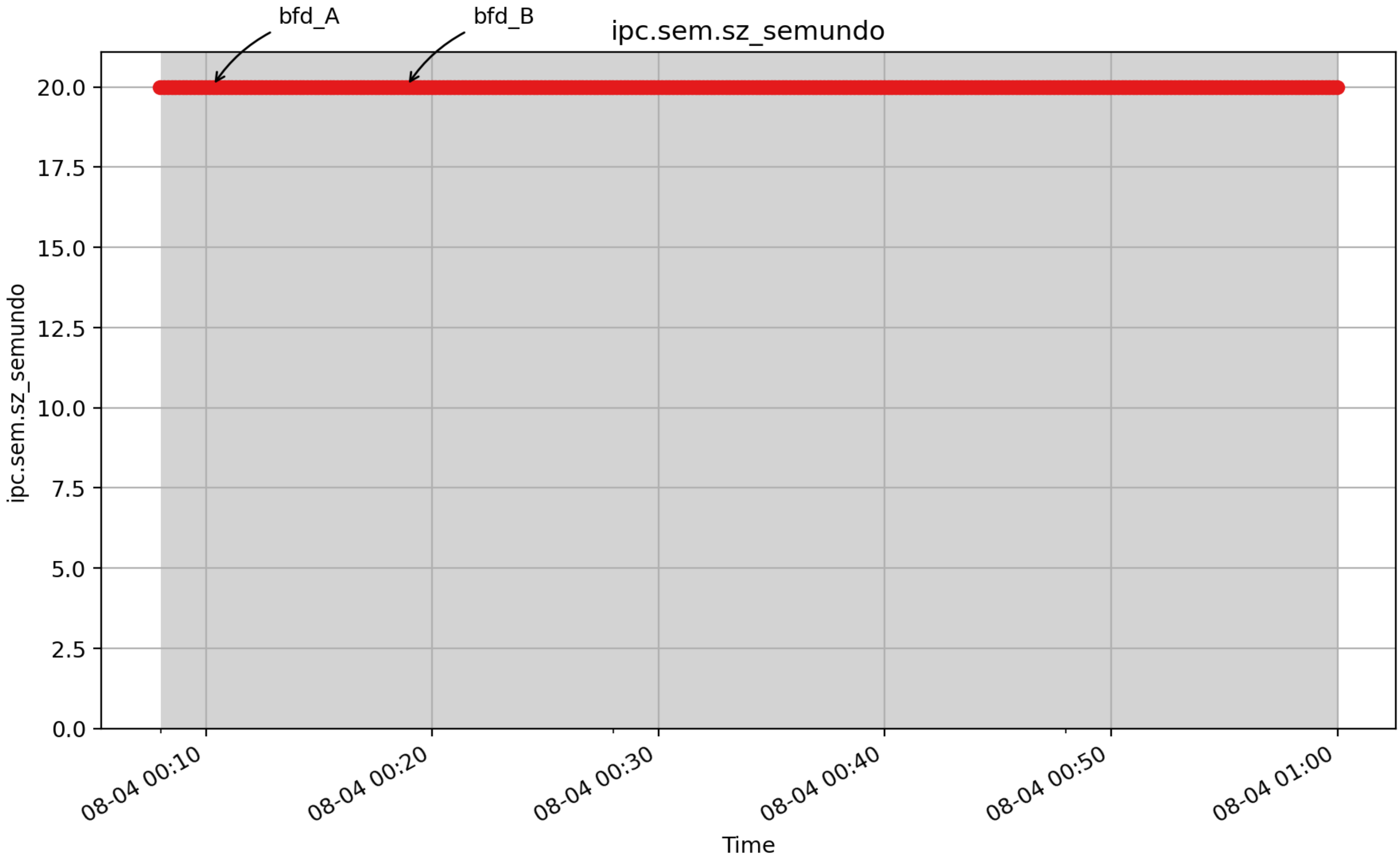
ipc.sem.max_semval: semaphore maximum value (from semctl(..,IPC_INFO,..)) (- U32)



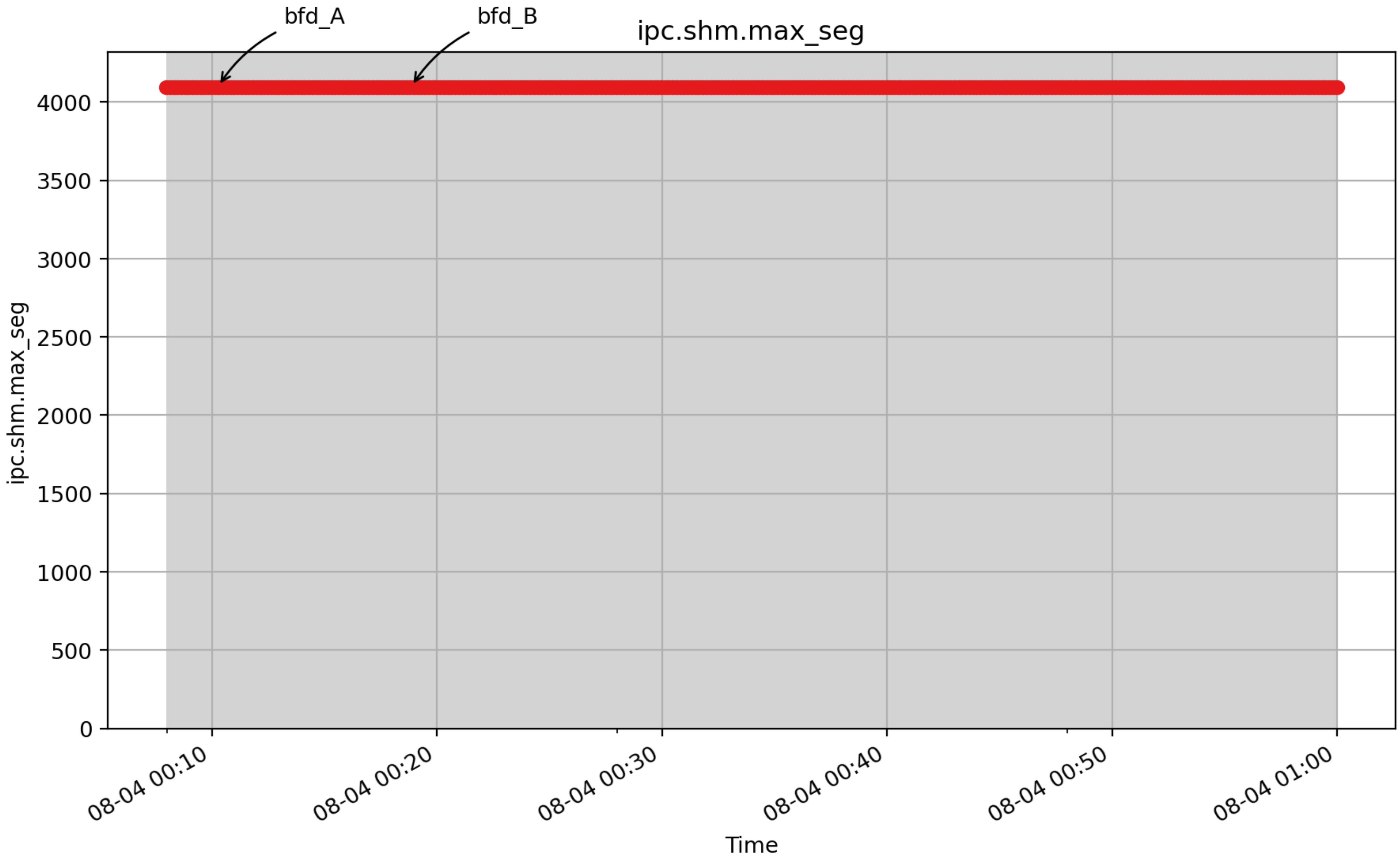
ipc.sem.max_undoent: maximum number of undo entries per process (from semctl(..,IPC_INFO,..)) (- U32)



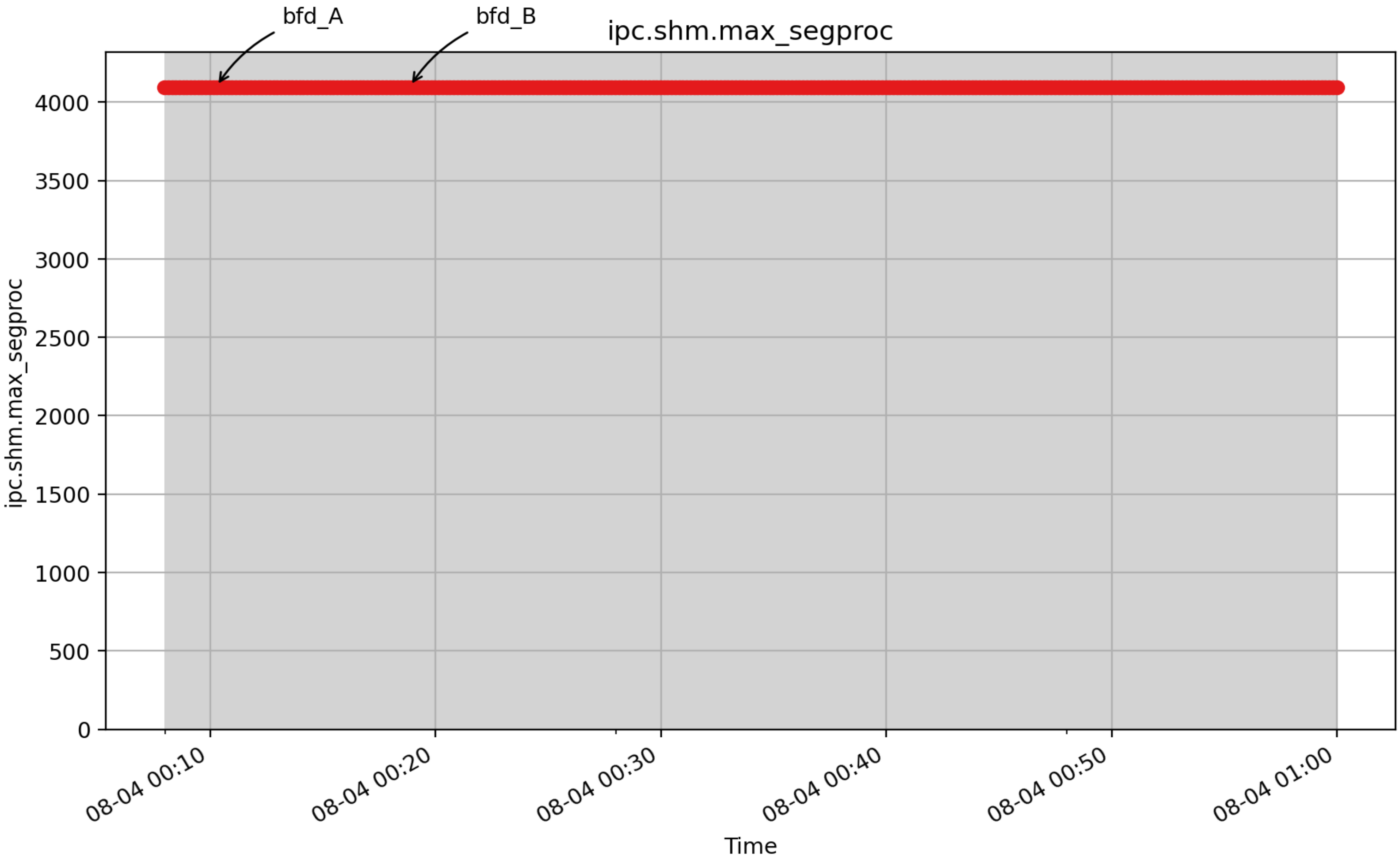
ipc.sem.num_undo: number of undo structures in system (from semctl(..,IPC_INFO,..)) (- U32)



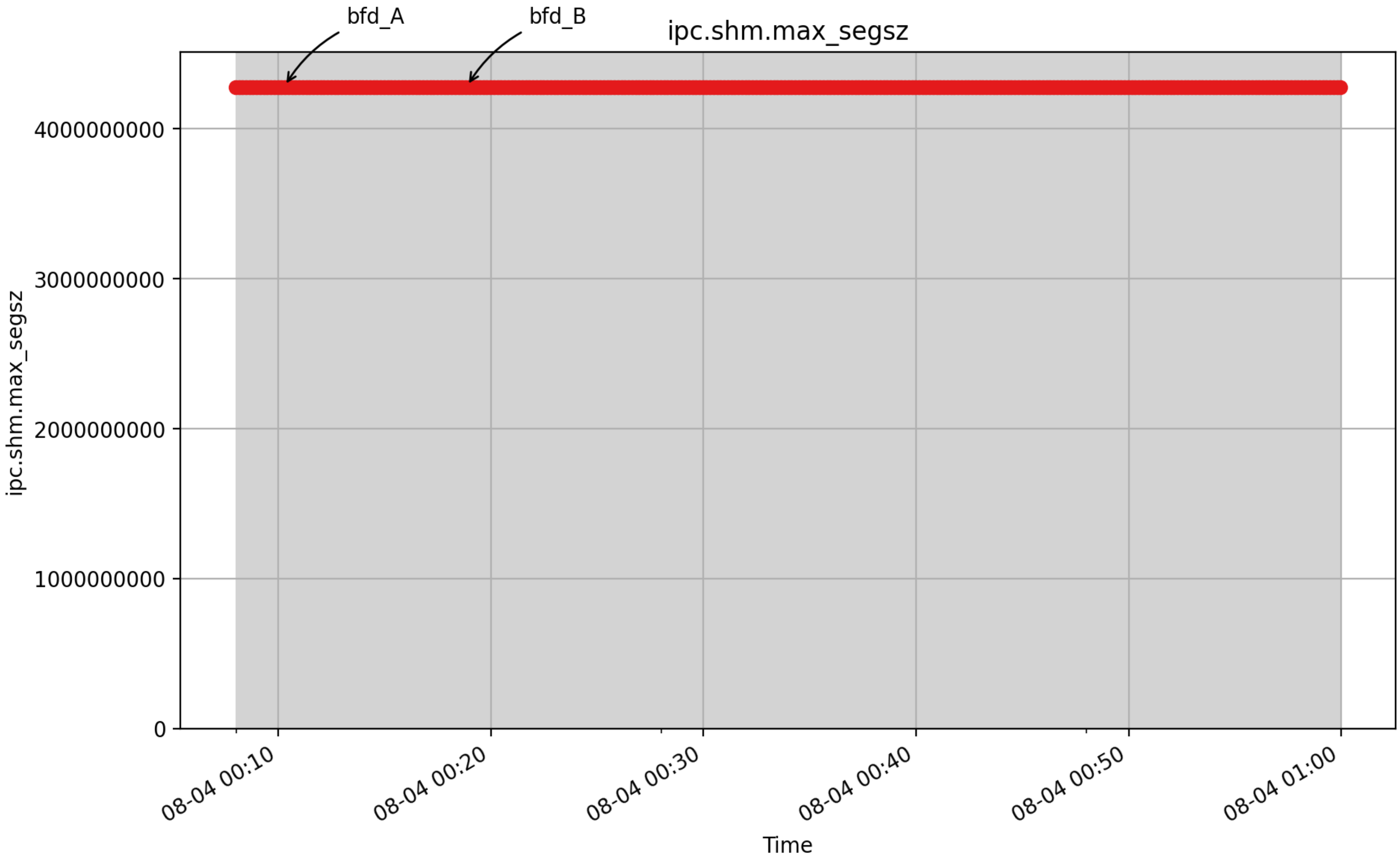
ipc.sem.sz_semundo: size of struct sem_undo (from semctl(..,IPC_INFO,..)) (- U32)



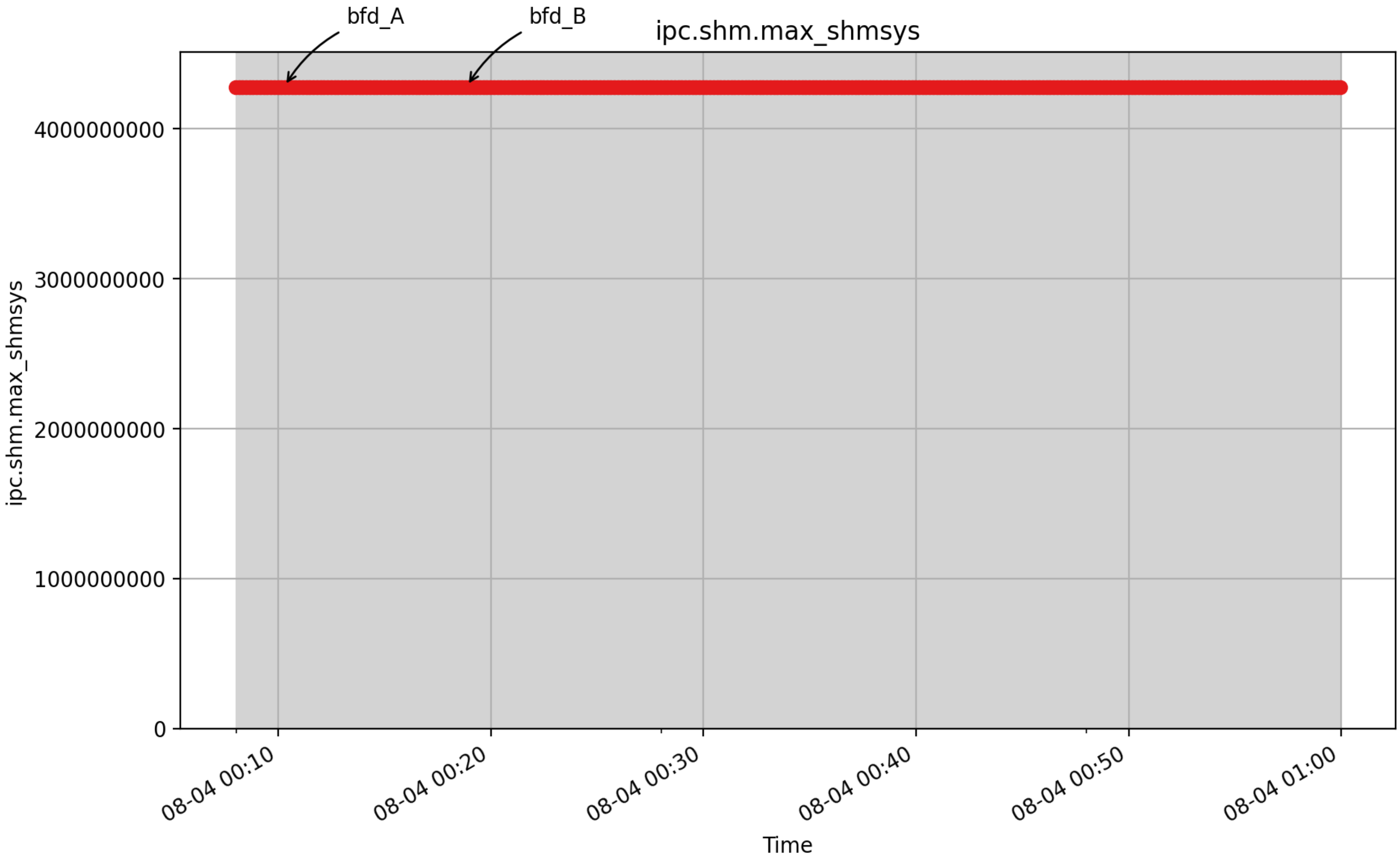
ipc.shm.max_seg: maximum number of shared segments in system (from shmctl(..,IPC_INFO,..)) (- U32)



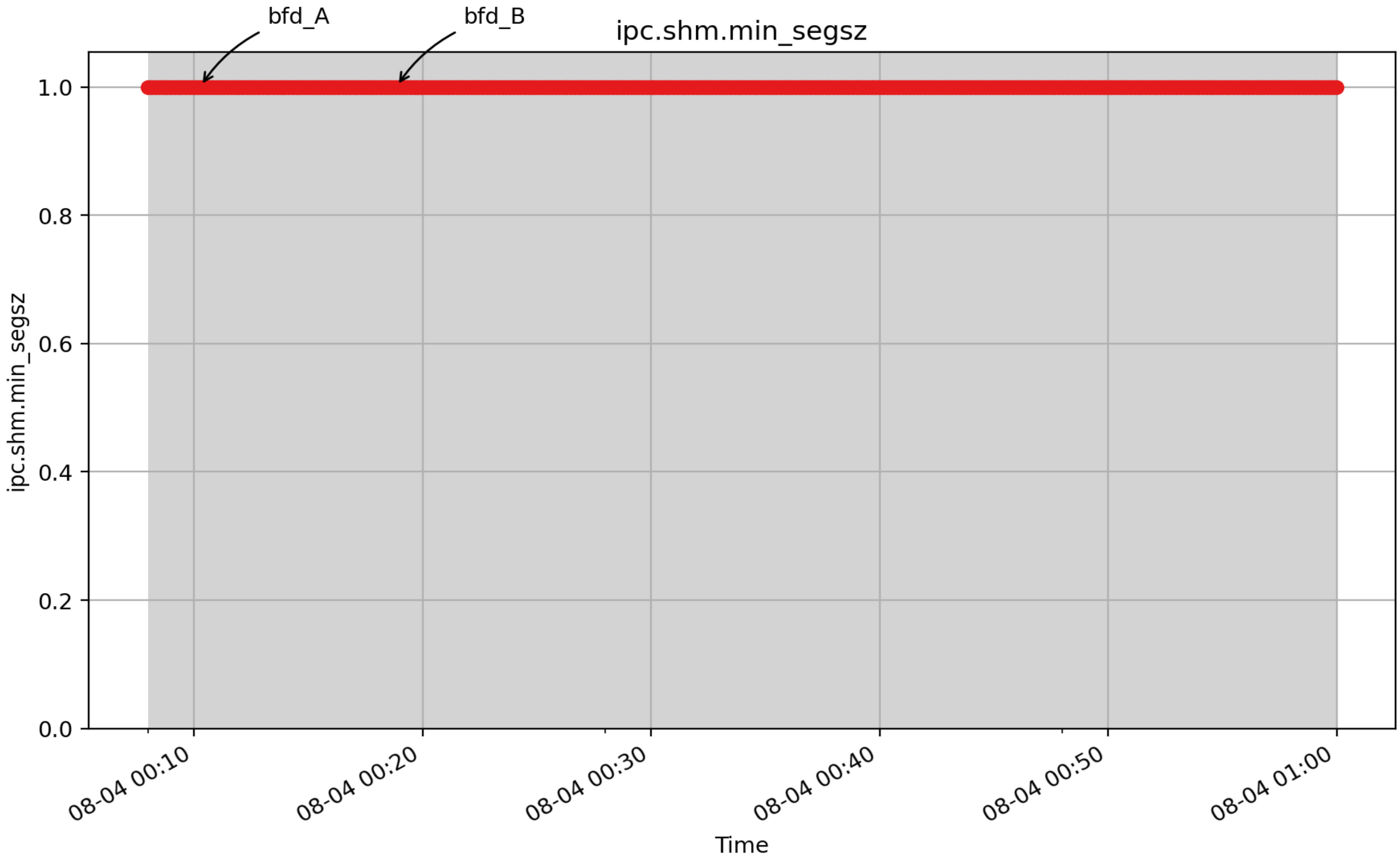
ipc.shm.max_segproc: maximum number of shared segments per process (from shmctl(..,IPC_INFO,..)) (- U32)



ipc.shm.max_segsz: maximum shared segment size in bytes (from shmctl(..,IPC_INFO,..)) (byte - U32)

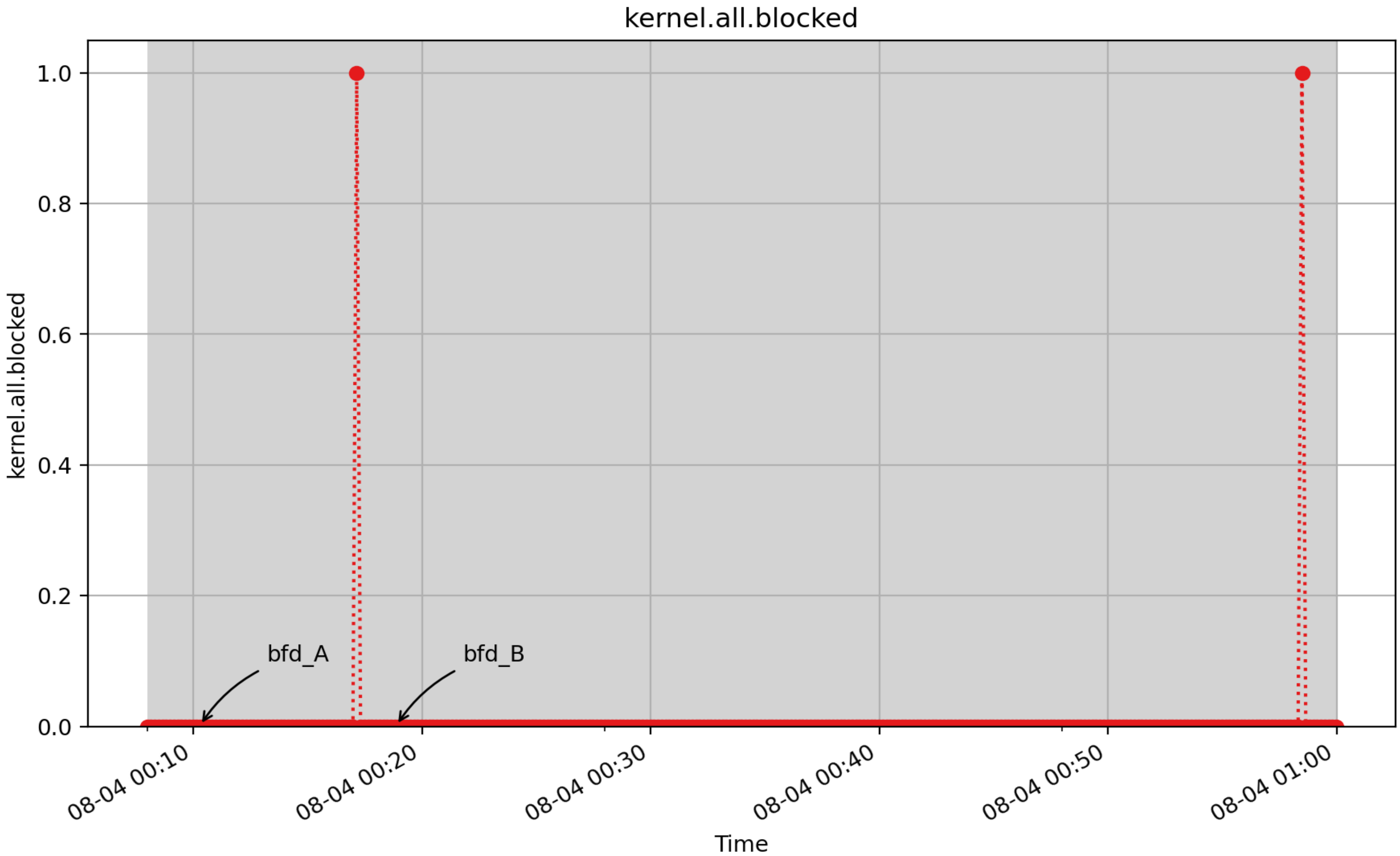


ipc.shm.max_shmsys: maximum amount of shared memory in system in pages (from shmctl(..,IPC_INFO,..))
(- U32)

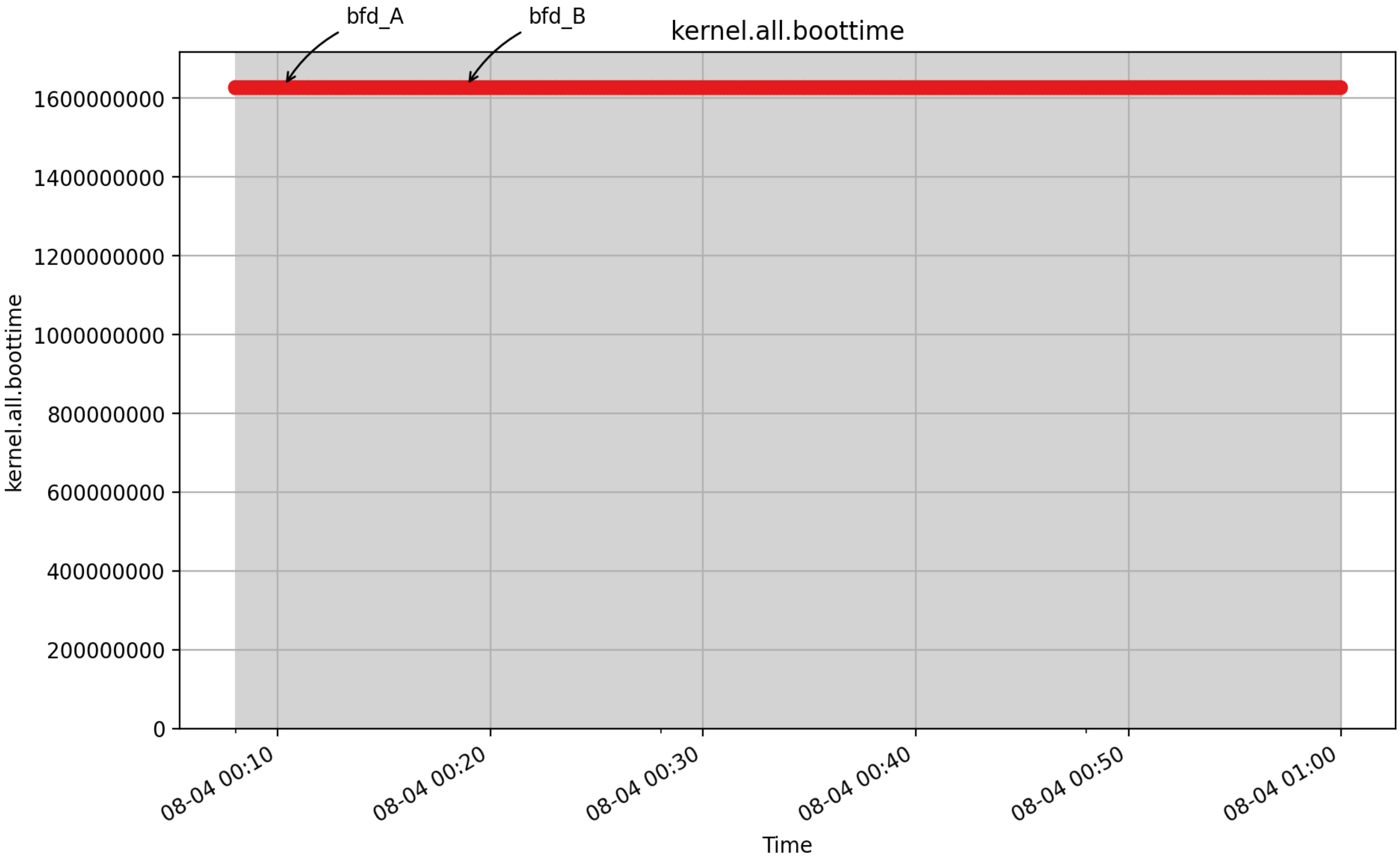


ipc.shm.min_segsz: minimum shared segment size in bytes (from shmctl(..,IPC_INFO,..)) (byte - U32)

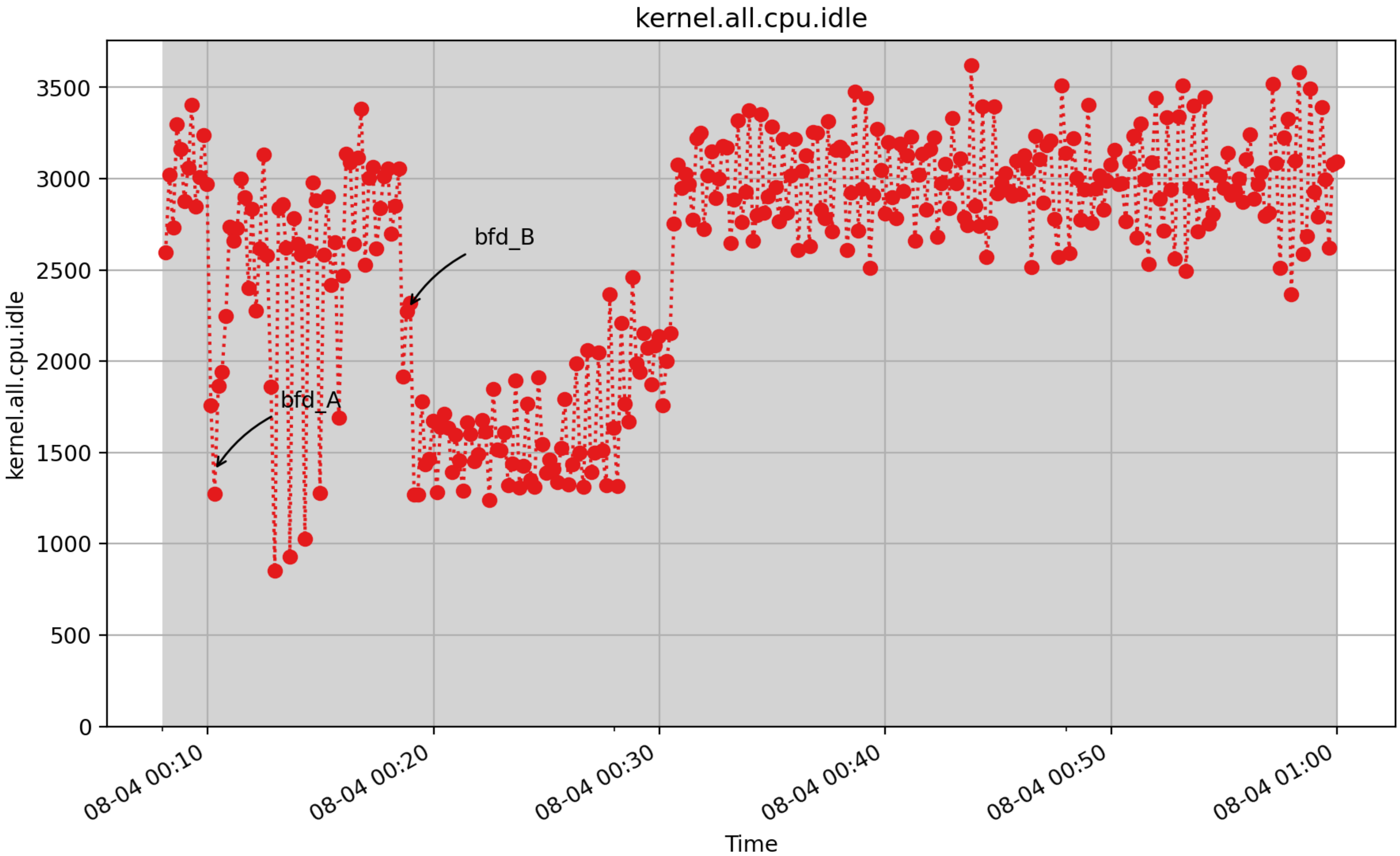
Kernel



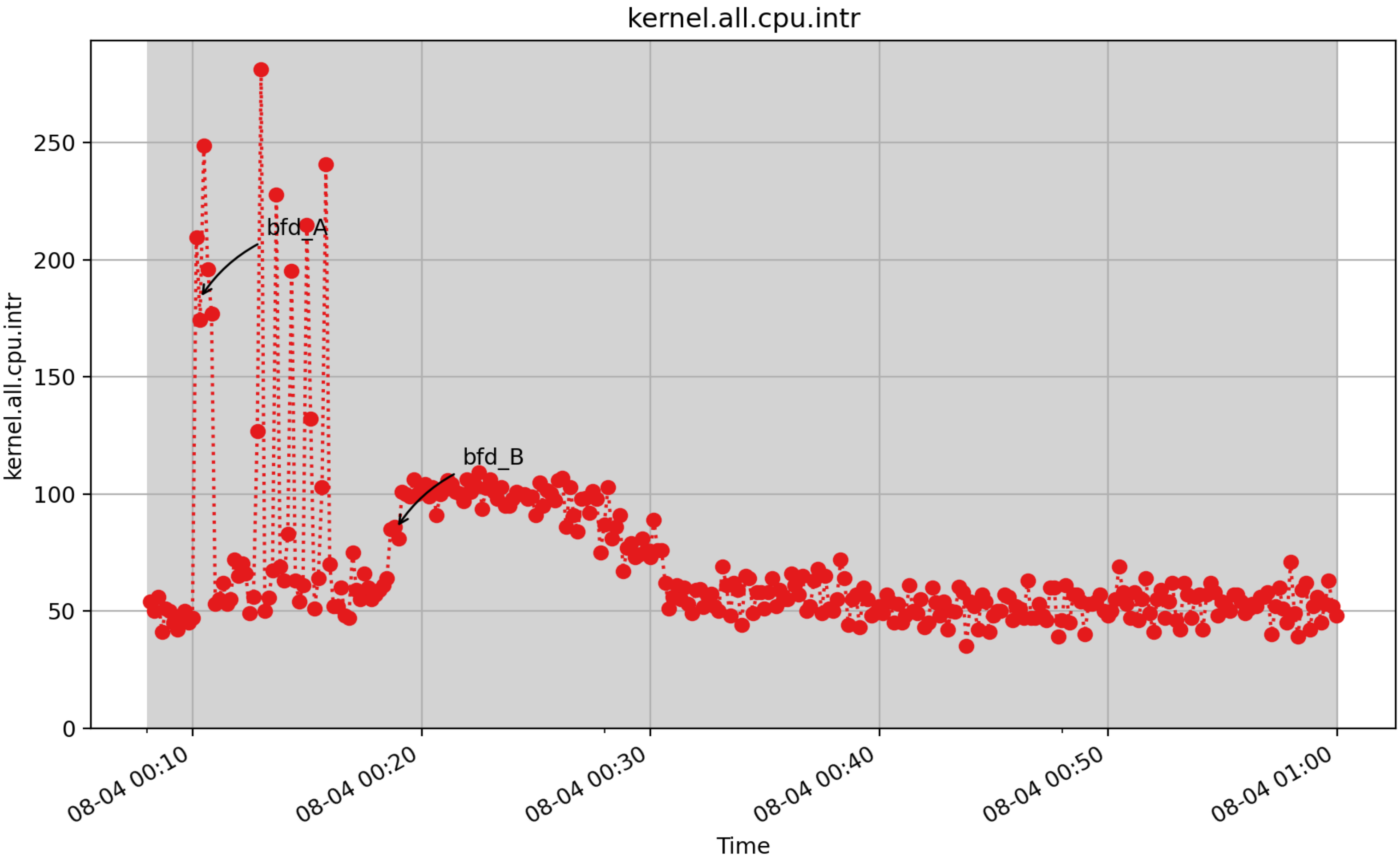
kernel.all.blocked: number of currently blocked processes from /proc/stat (- U64)



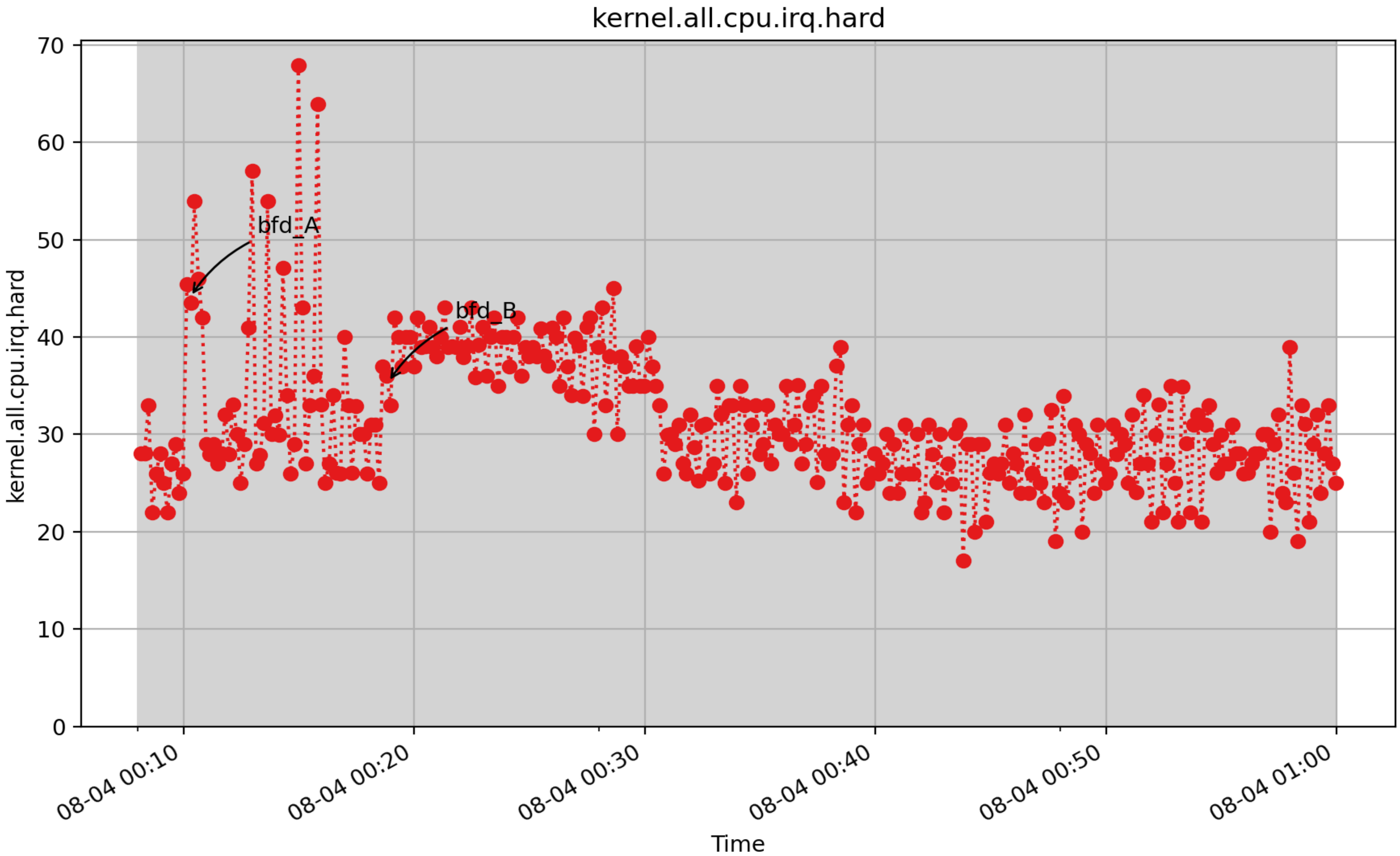
kernel.all.boottime: boot time from /proc/stat (sec - 64)



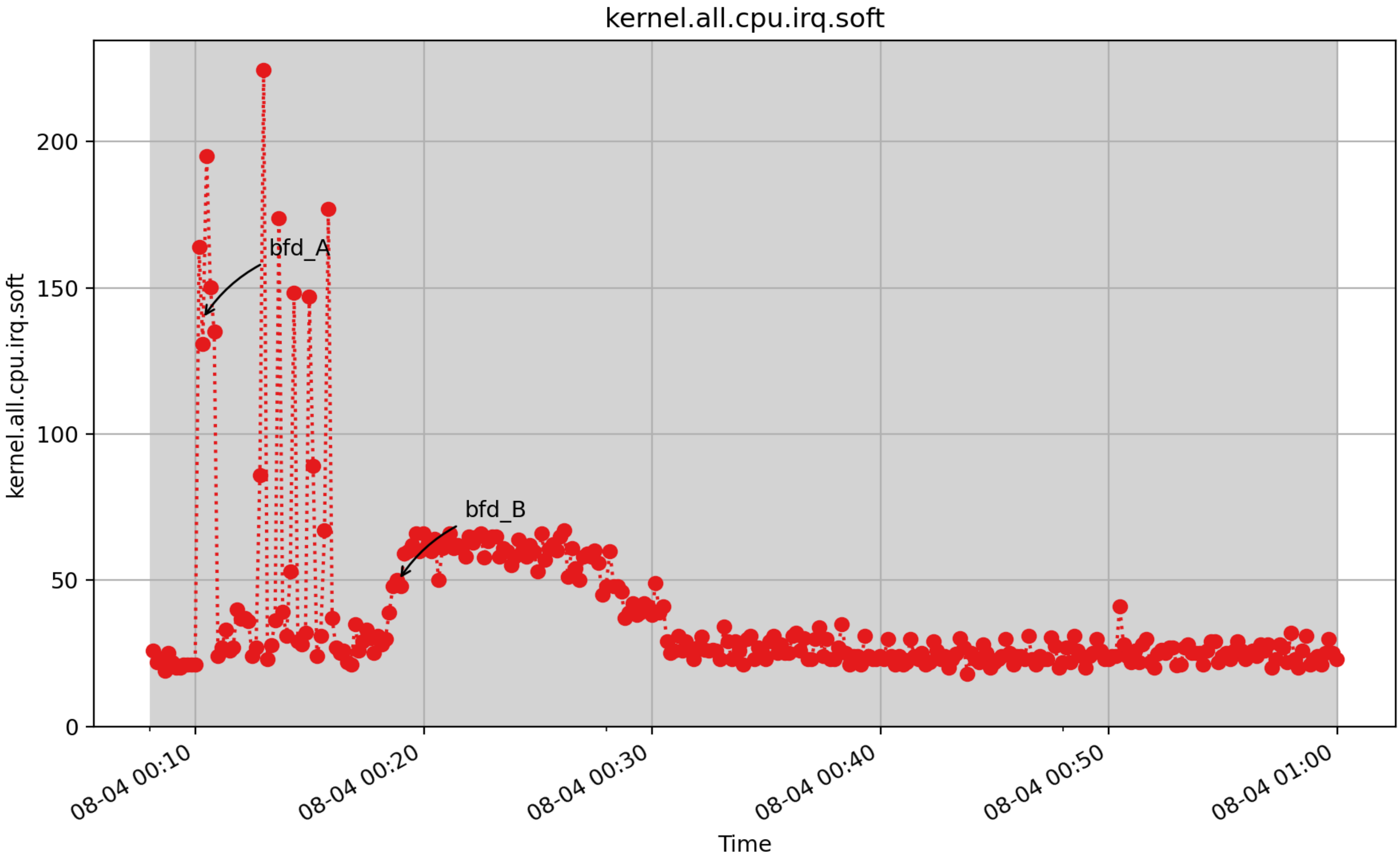
kernel.all.cpu.idle: total idle CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



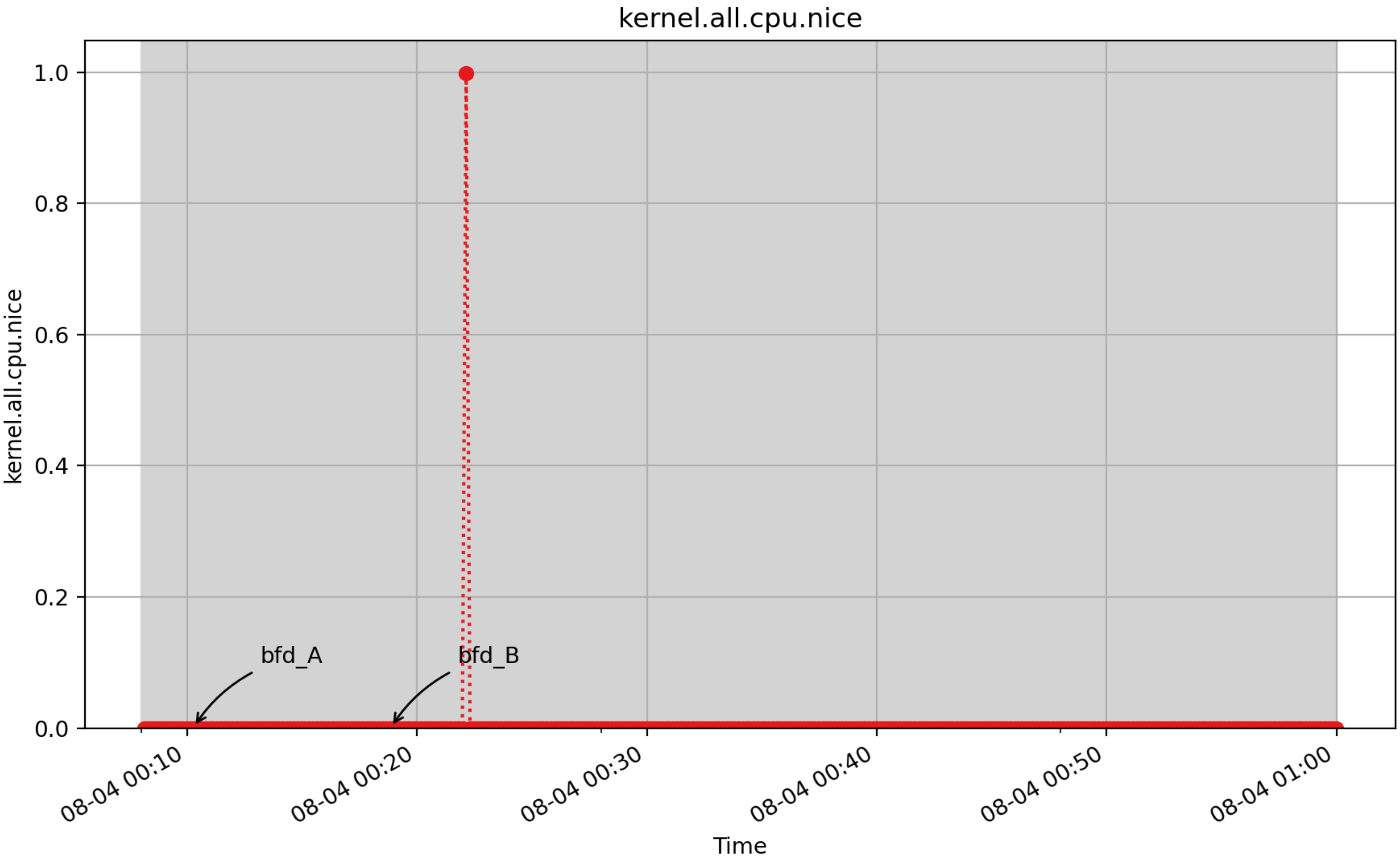
kernel.all.cpu.intr: Total time spent processing interrupts on all CPUs. This value includes both soft and hard interrupt processing time. (millisec - U64) - *rate converted*



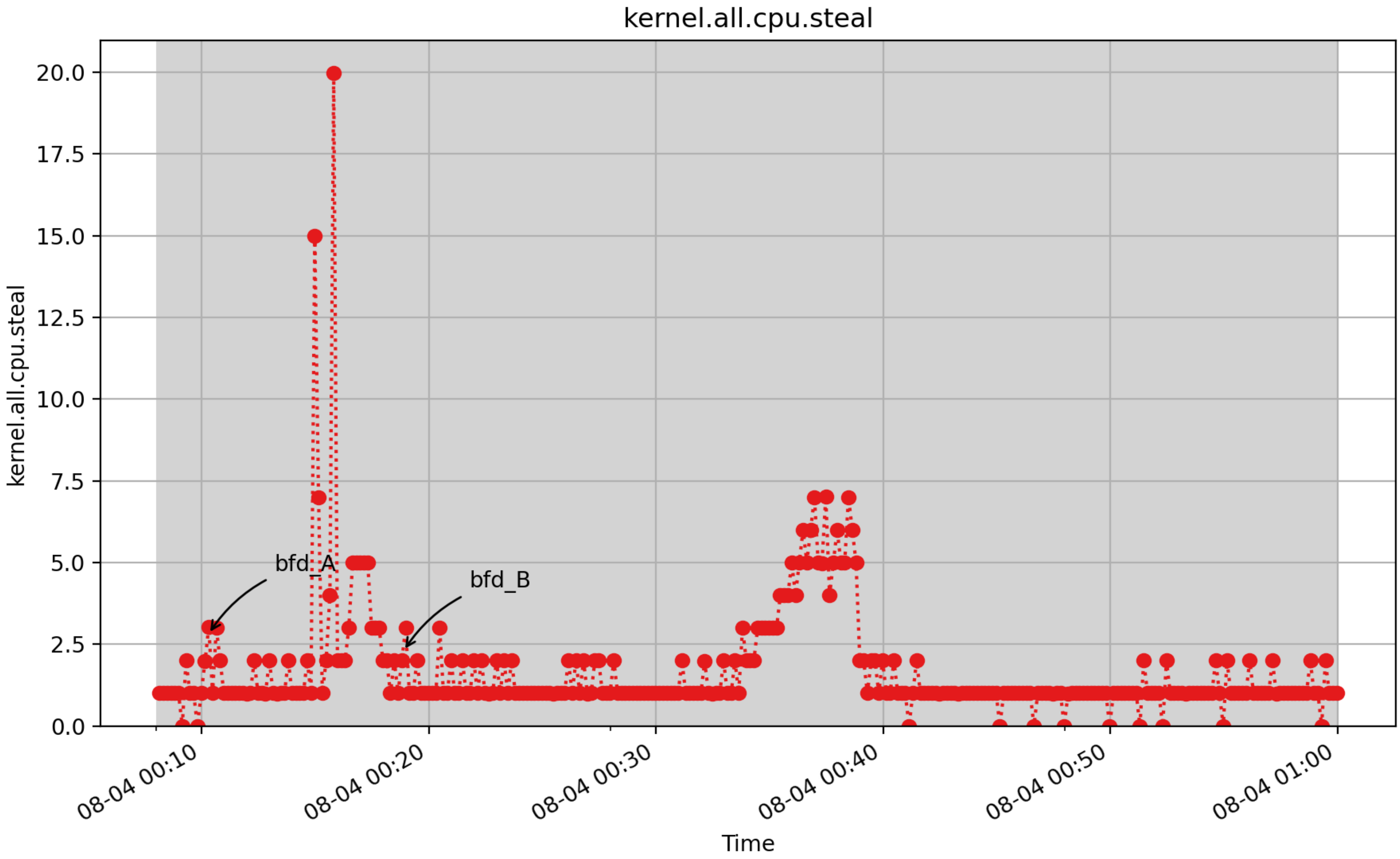
kernel.all.cpu.irq.hard: Total hard interrupt CPU time ("hard" interrupt handling code is the code run directly on receipt of the initial hardware interrupt, and does not include "soft" interrupt handling code which is deferred until later). (millisec - U64) - *rate converted*



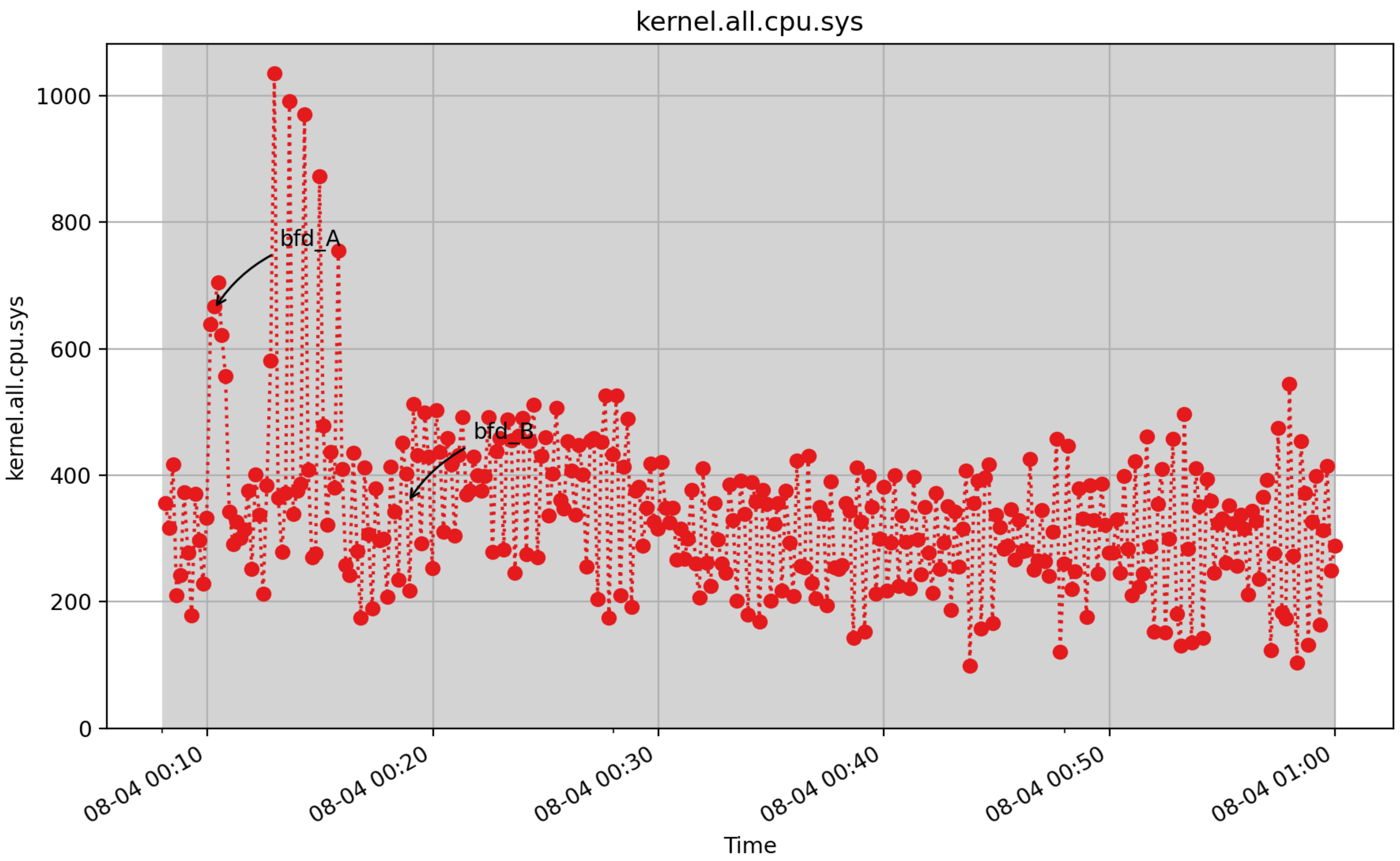
kernel.all.cpu.irq.soft: Total soft interrupt CPU time (deferred interrupt handling code, not run in the initial interrupt handler). (millisec - U64) - *rate converted*



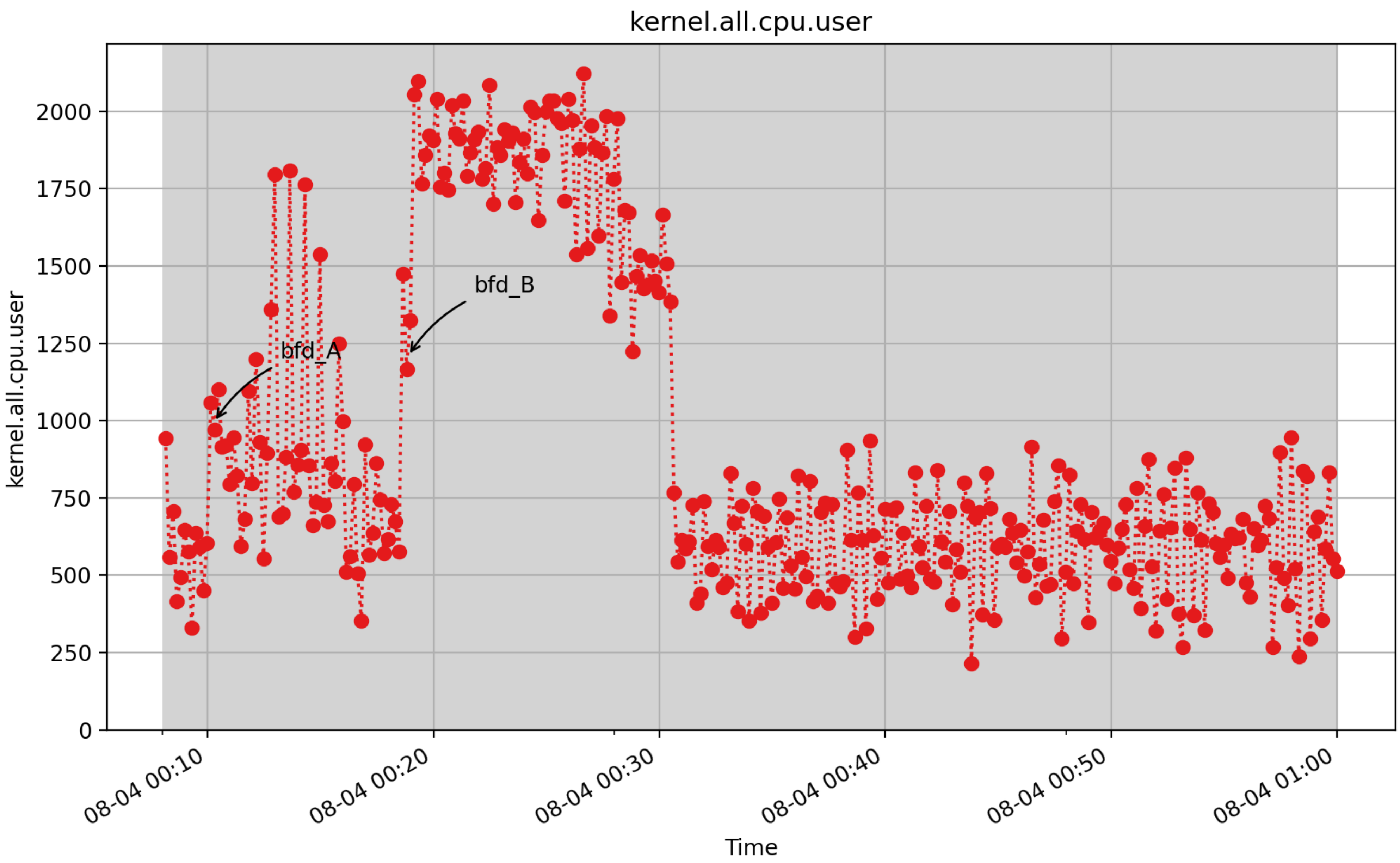
kernel.all.cpu.nice: total nice user CPU time from /proc/stat for all CPUs, including guest time (millisec - U64) - *rate converted*



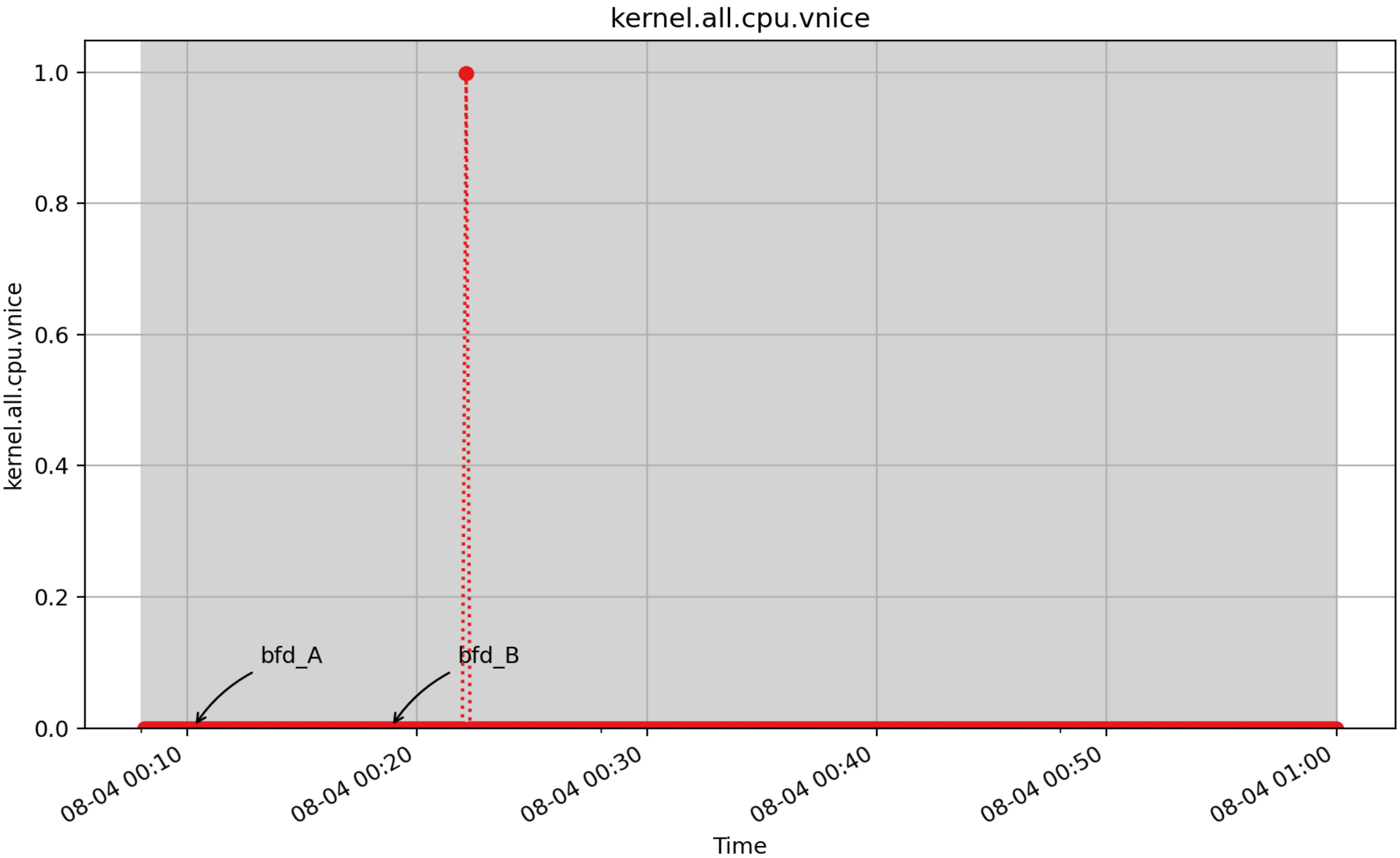
kernel.all.cpu.steal: Total CPU time when a CPU had a runnable process, but the hypervisor (virtualisation layer) chose to run something else instead. (millisec - U64) - *rate converted*



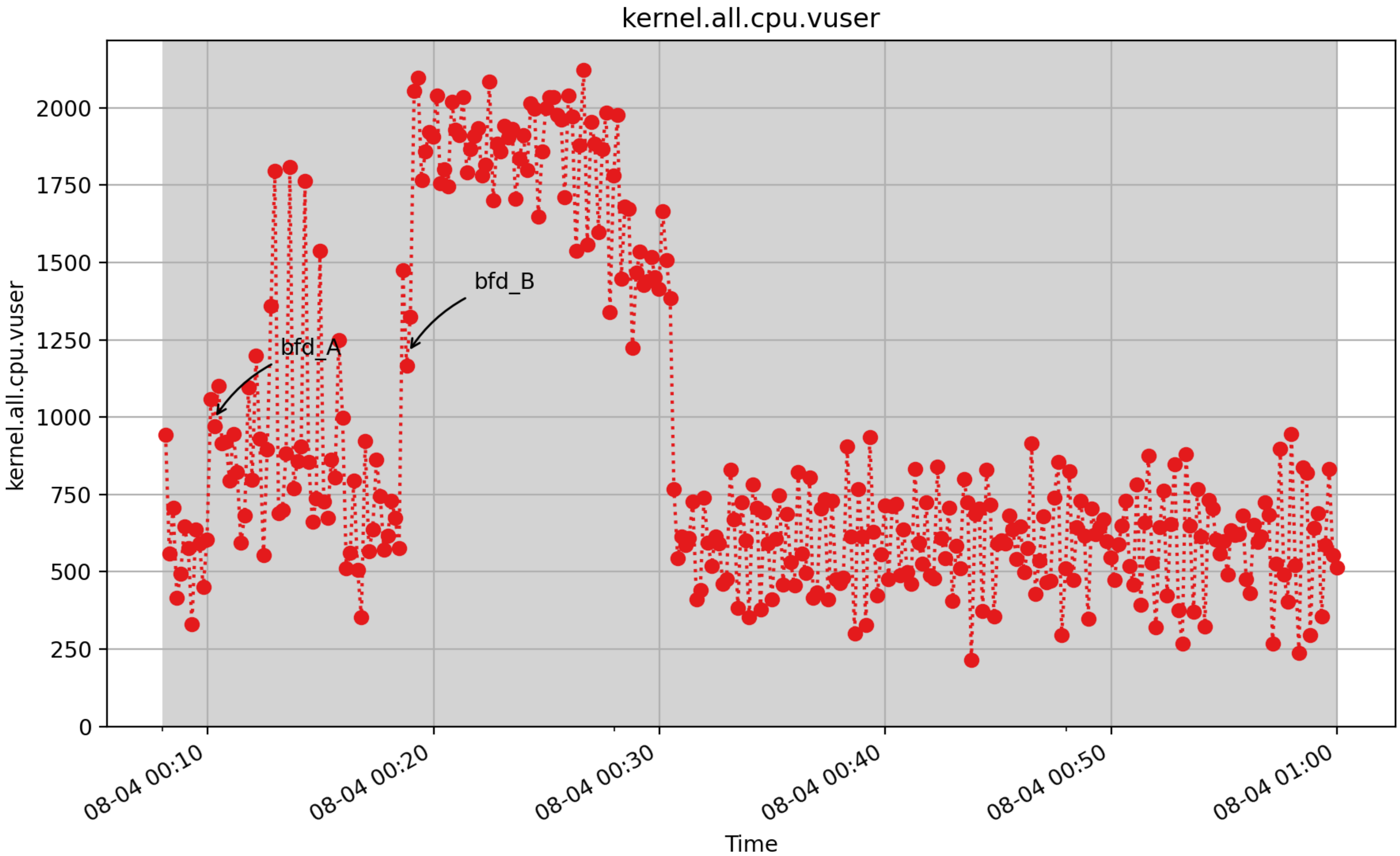
kernel.all.cpu.sys: total sys CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



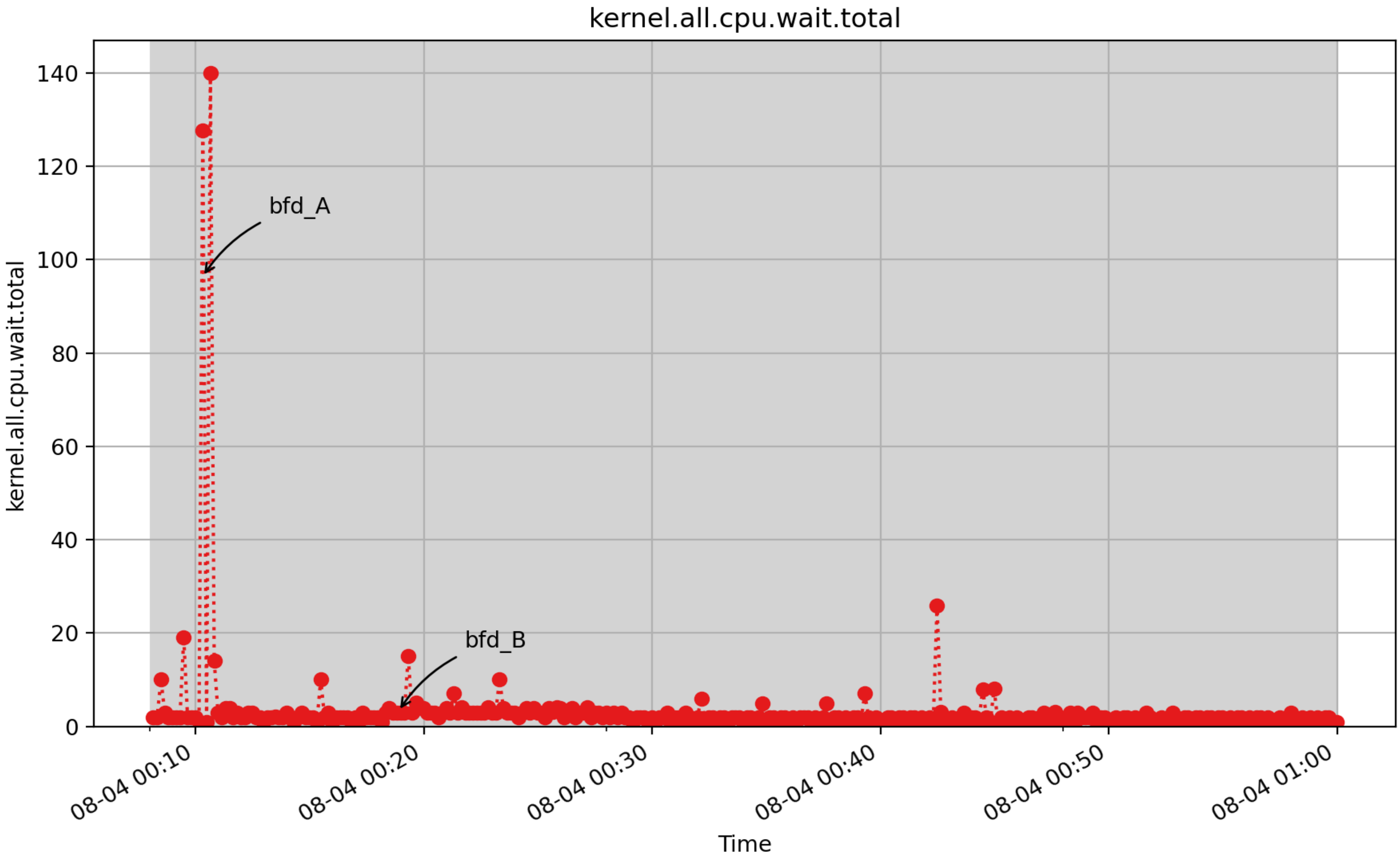
kernel.all.cpu.user: total user CPU time from /proc/stat for all CPUs, including guest CPU time (millisec - U64) - *rate converted*



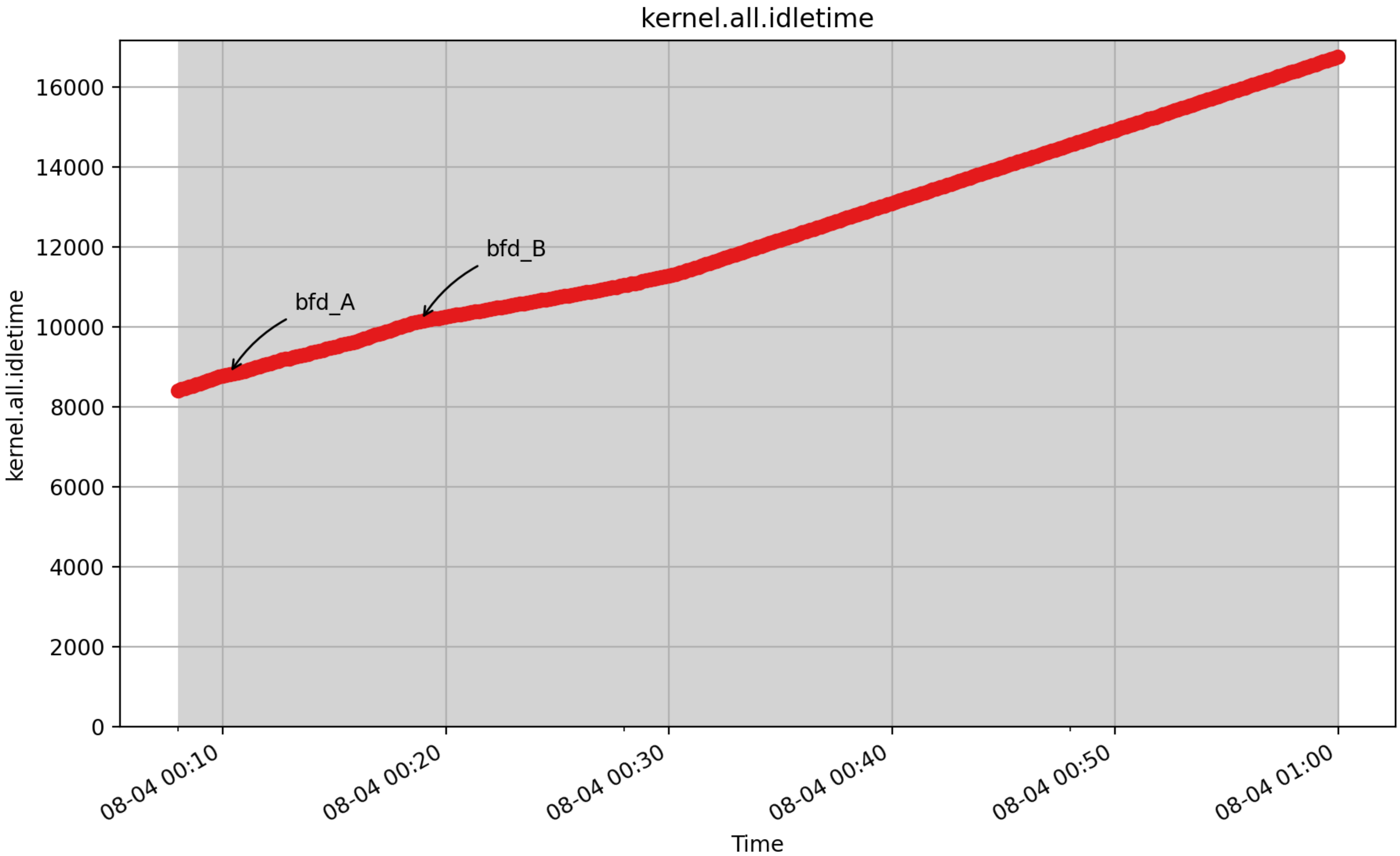
kernel.all.cpu.vnice: total nice user CPU time from /proc/stat for all CPUs, excluding guest time (millisec - U64) - *rate converted*



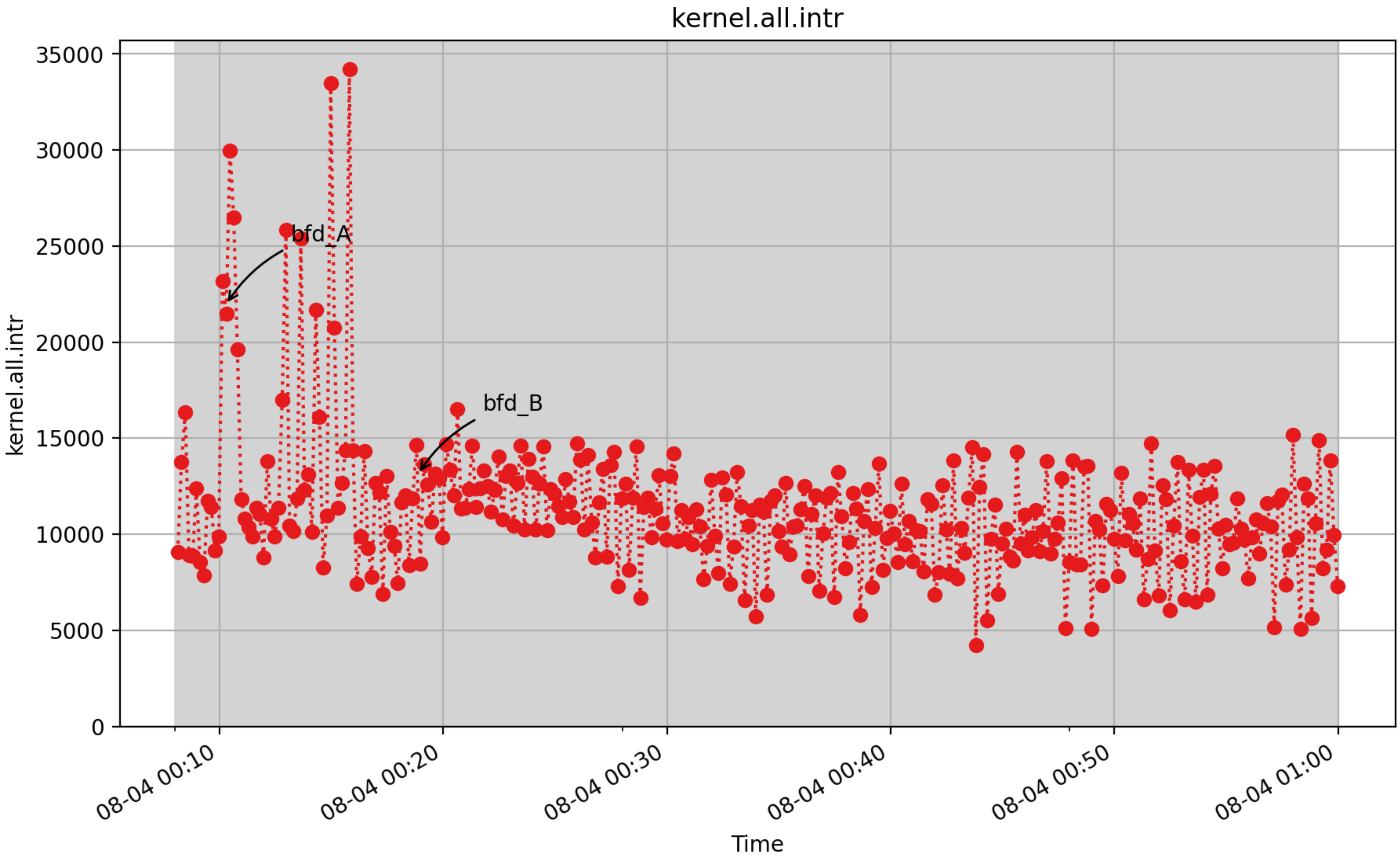
kernel.all.cpu.vuser: total user CPU time from /proc/stat for all CPUs, excluding guest CPU time (millisec - U64) - *rate converted*



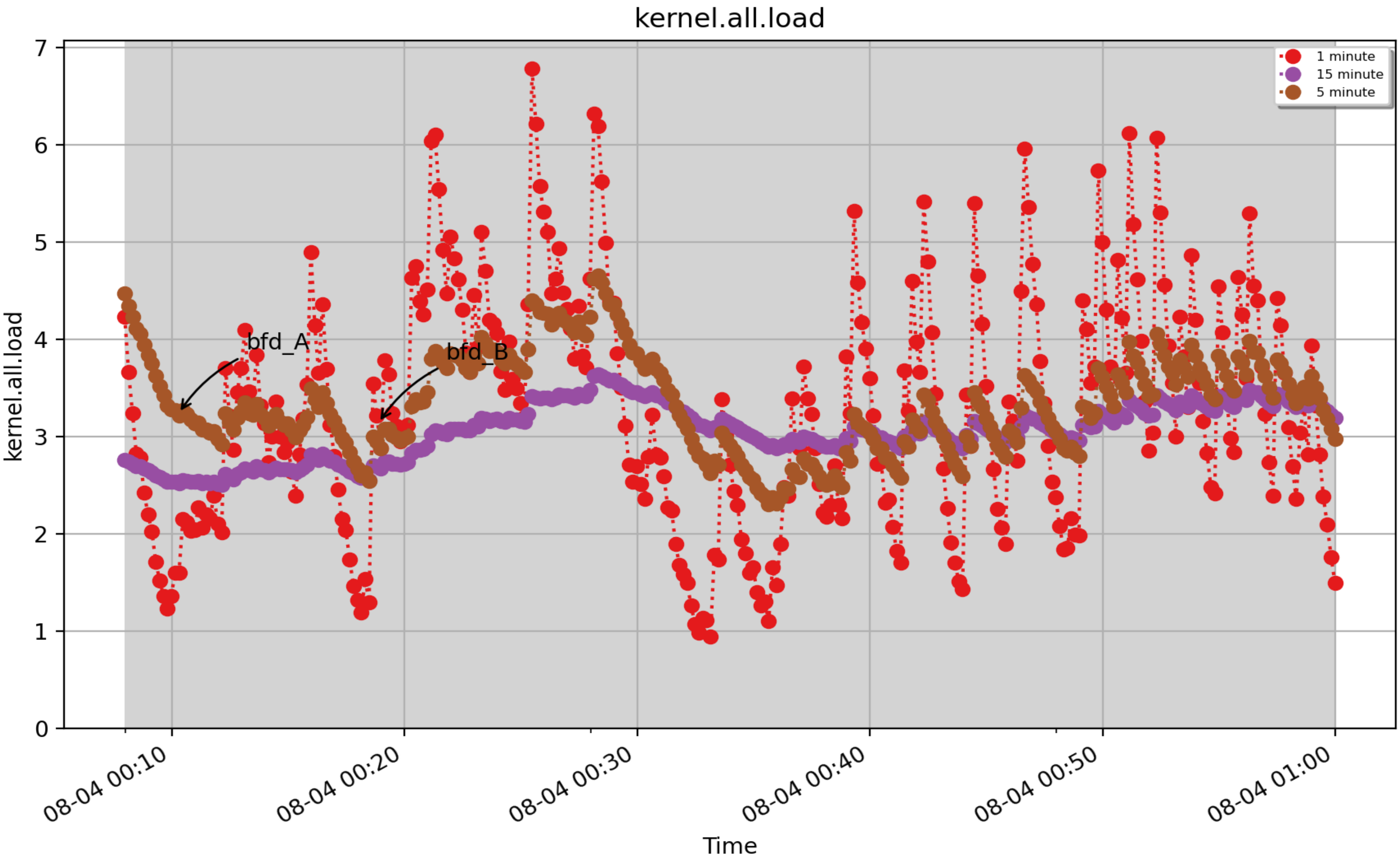
kernel.all.cpu.wait.total: total wait CPU time from /proc/stat for all CPUs (millisec - U64) - *rate converted*



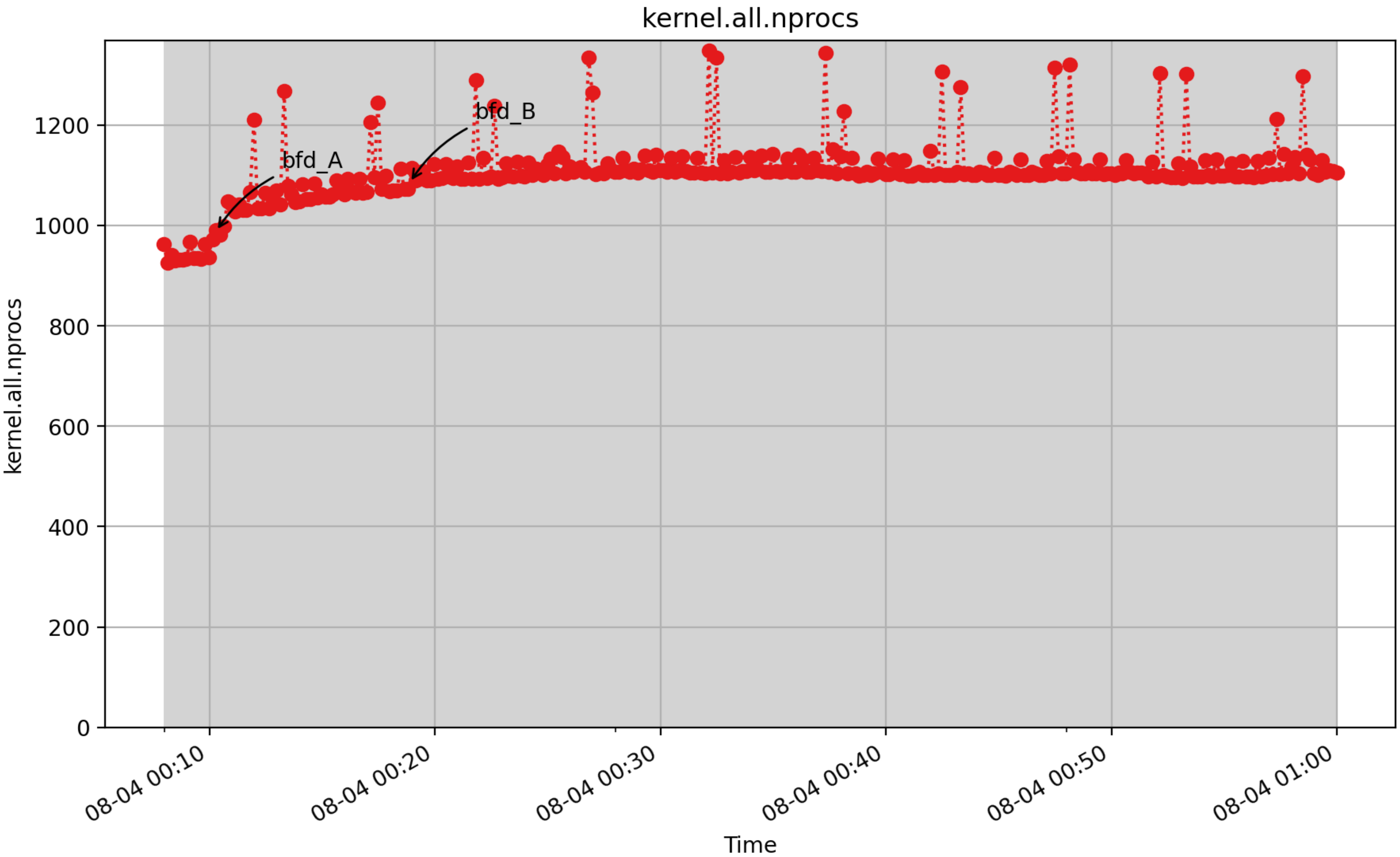
kernel.all.idletime: time the current kernel has been idle since boot (sec - DOUBLE)



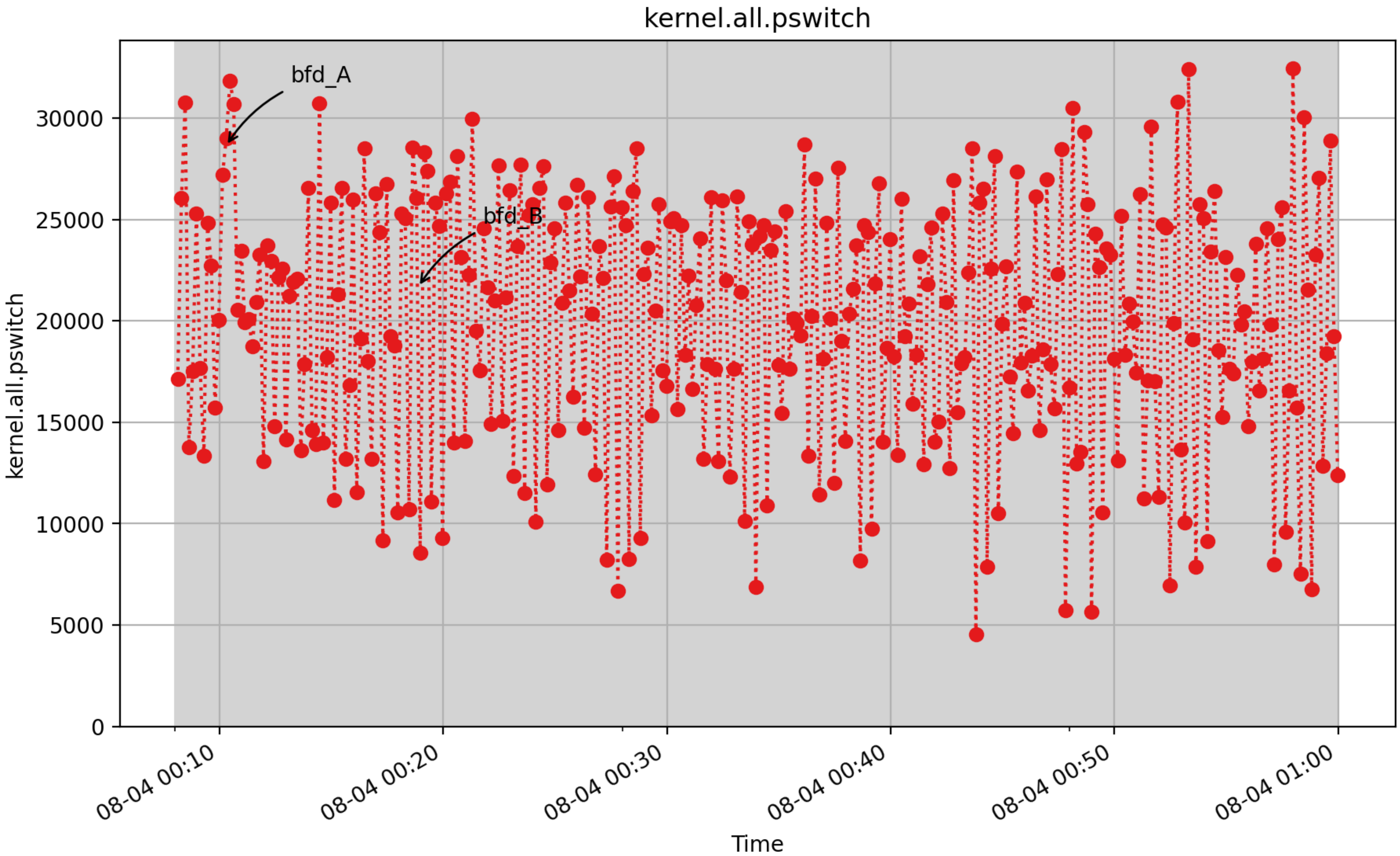
kernel.all.intr: The value is the first value from the intr field in /proc/stat, which is a counter of the total number of interrupts processed. The value is normally converted to a rate (count/second). This counter usually increases by at least HZ/second, i.e. the clock interrupt rate, wehich is usually 100/second. See also kernel.percpu.intr and kernel.percpu.interrupts to get the breakdown of interrupt count by interrupt type and which CPU processed each one. (count - U64) - *rate converted*



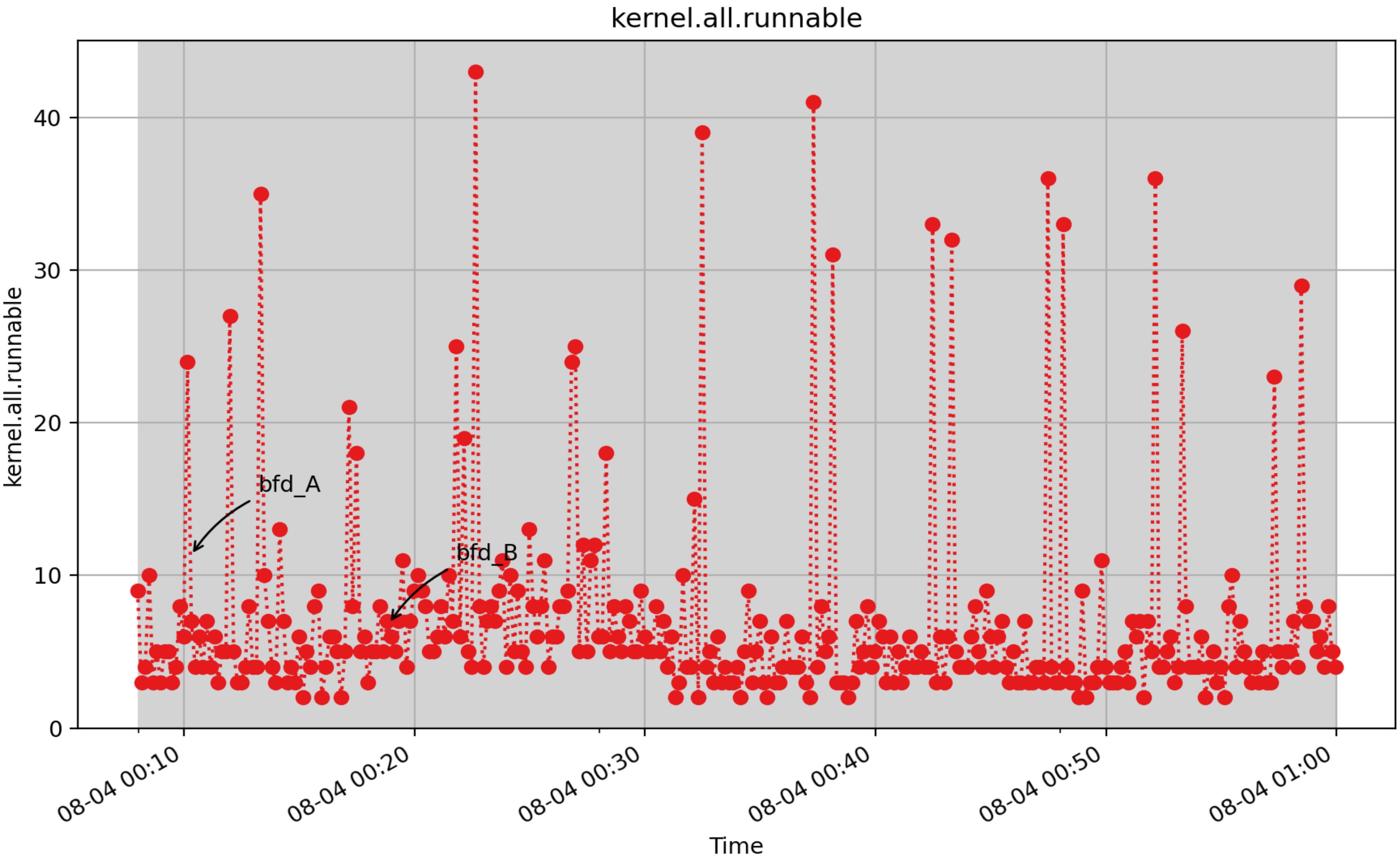
kernel.all.load: 1, 5 and 15 minute load average (- FLOAT)



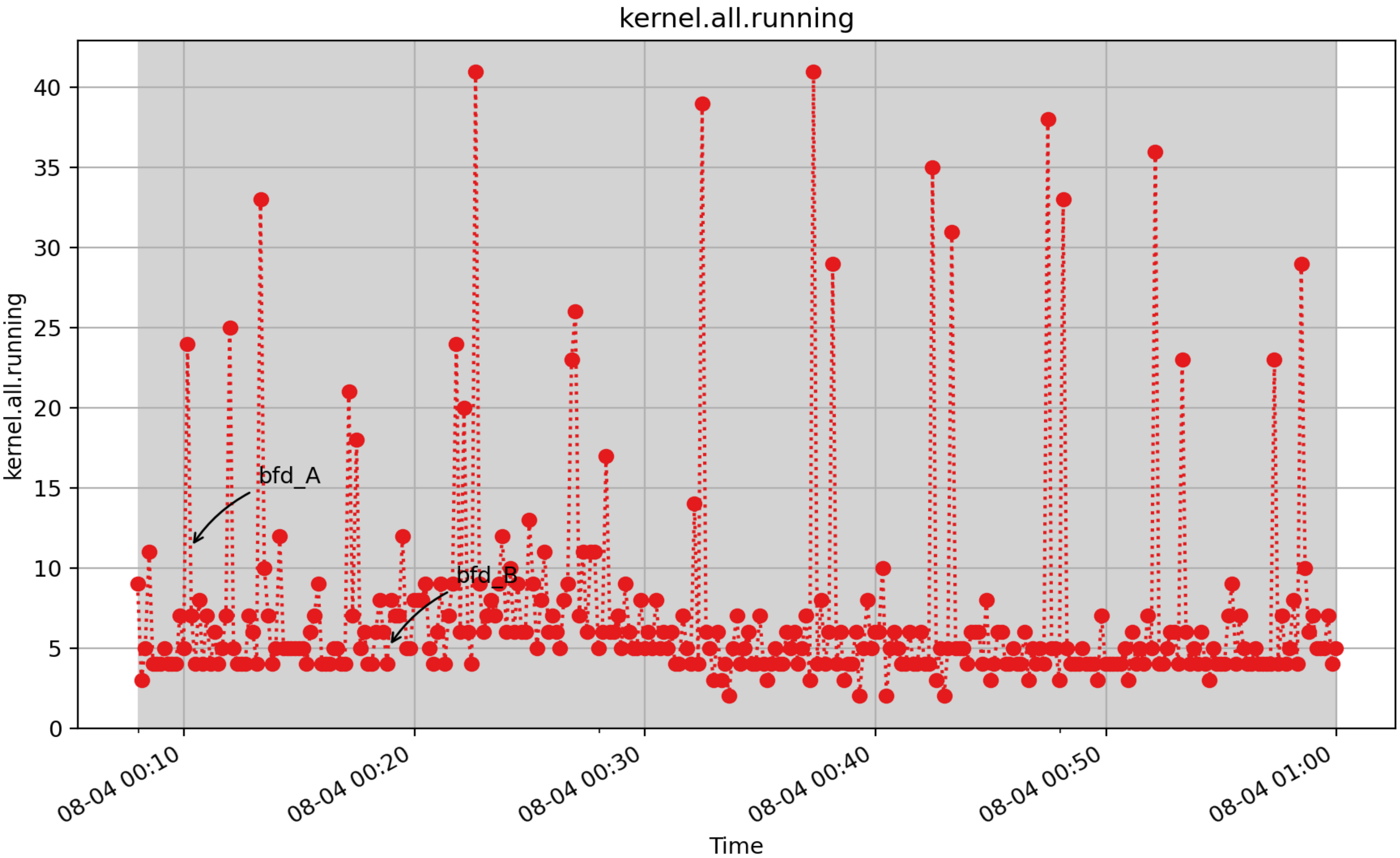
kernel.all.nprocs: total number of processes (lightweight) (- U32)



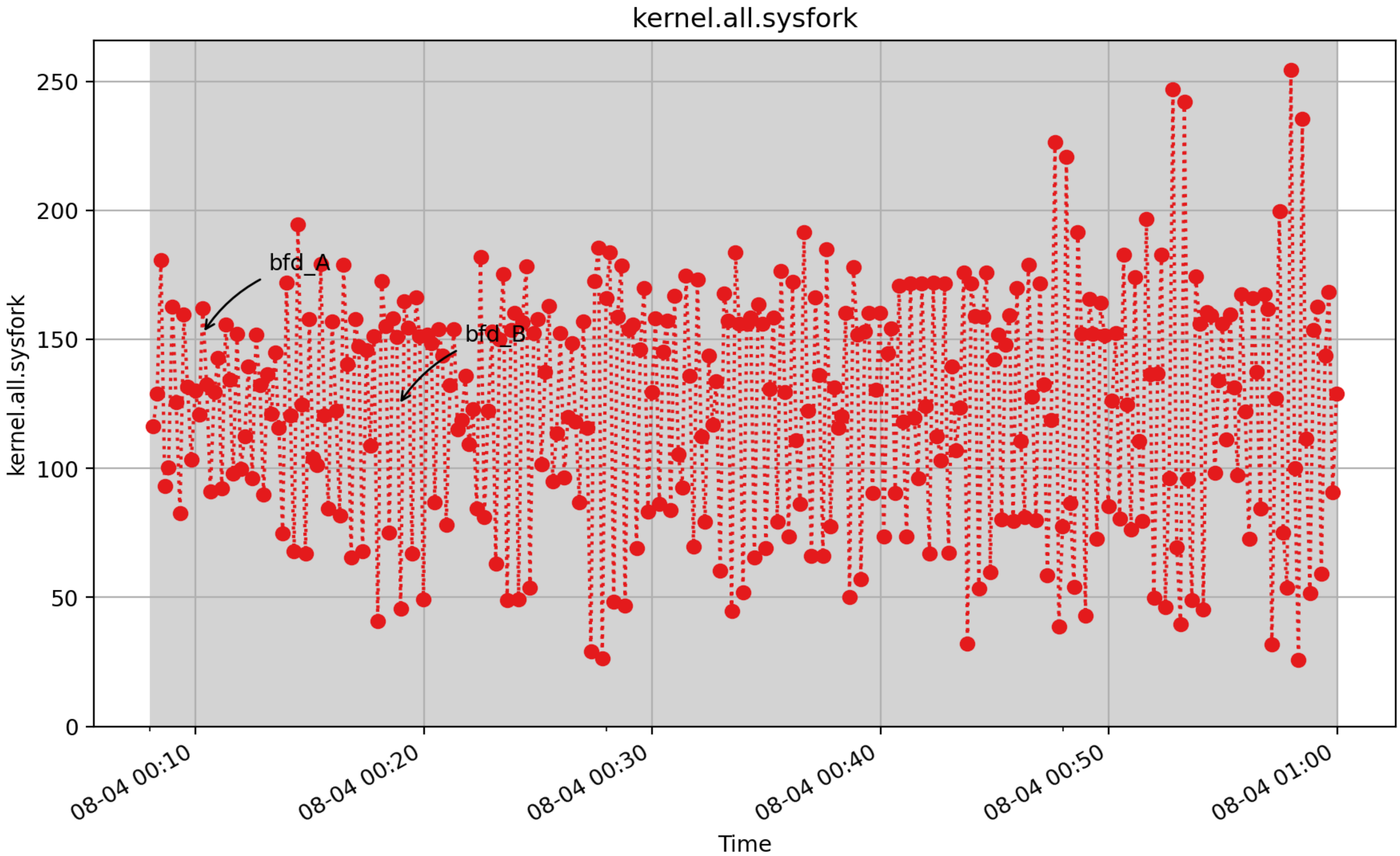
kernel.all.pswitch: context switches metric from /proc/stat (count - U64) - *rate converted*



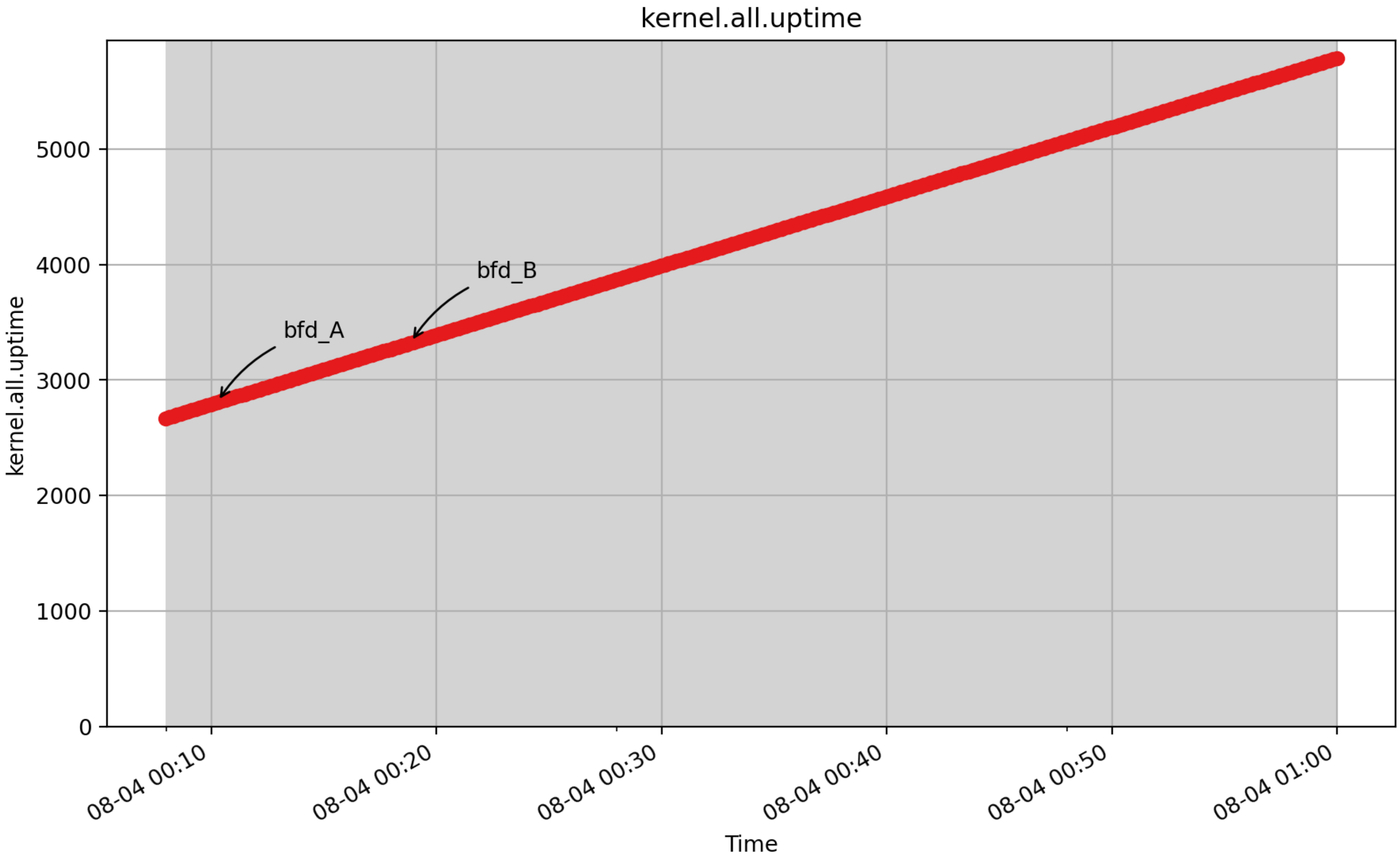
kernel.all.runnable: total number of processes in the (per-CPU) run queues (- U32)



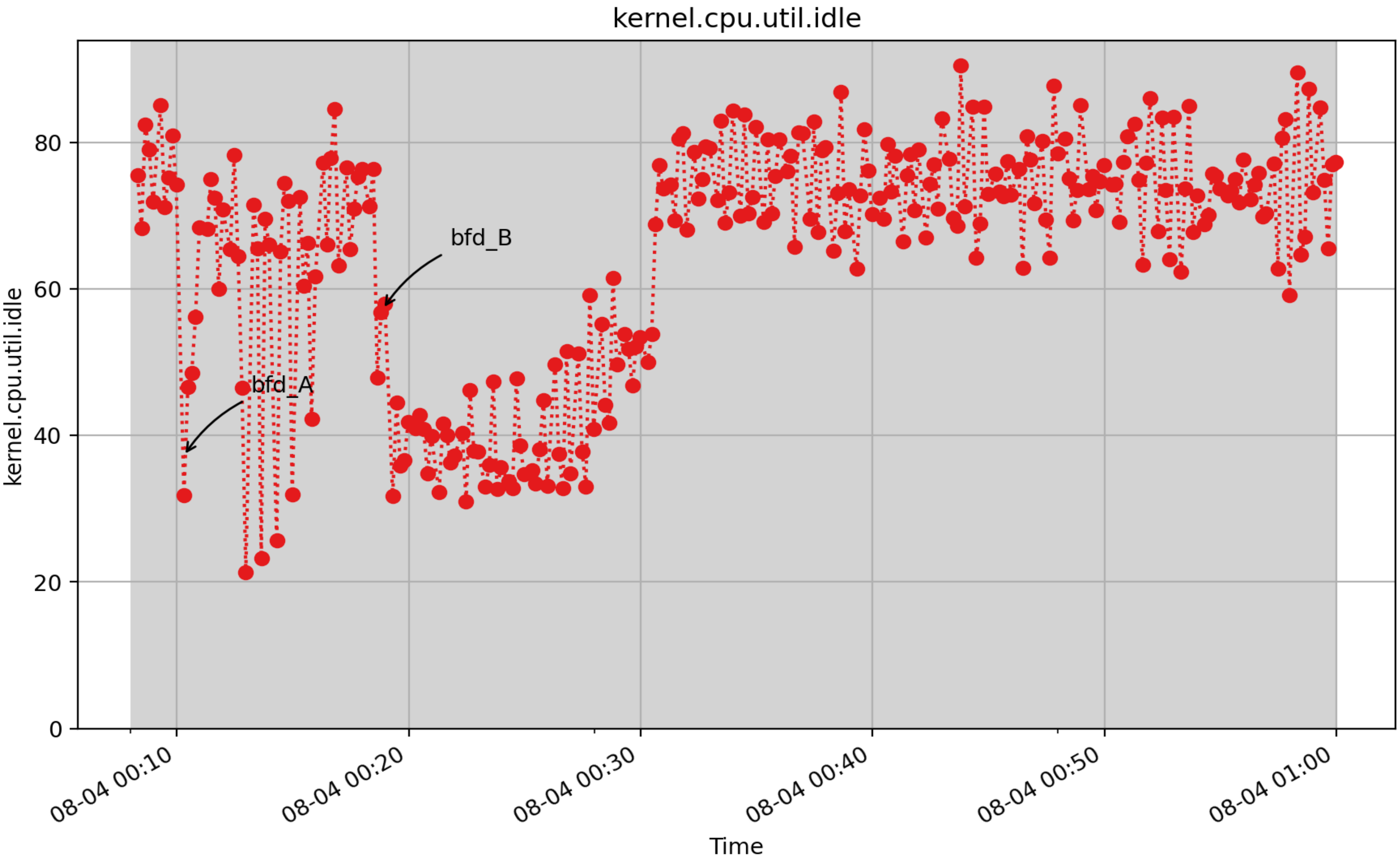
kernel.all.running: number of currently running processes from /proc/stat (- U64)



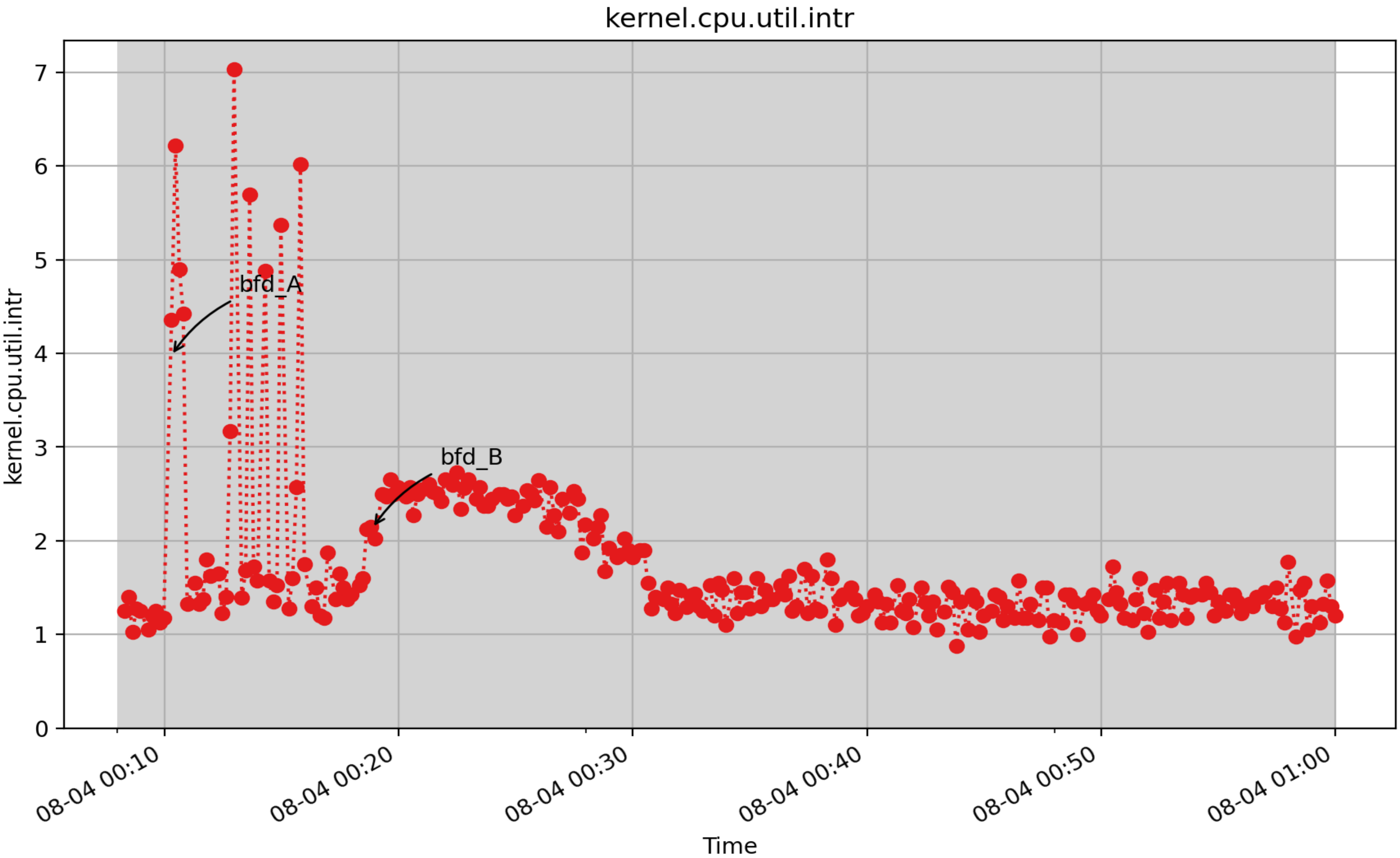
kernel.all.sysfork: fork rate metric from /proc/stat (count - U64) - *rate converted*



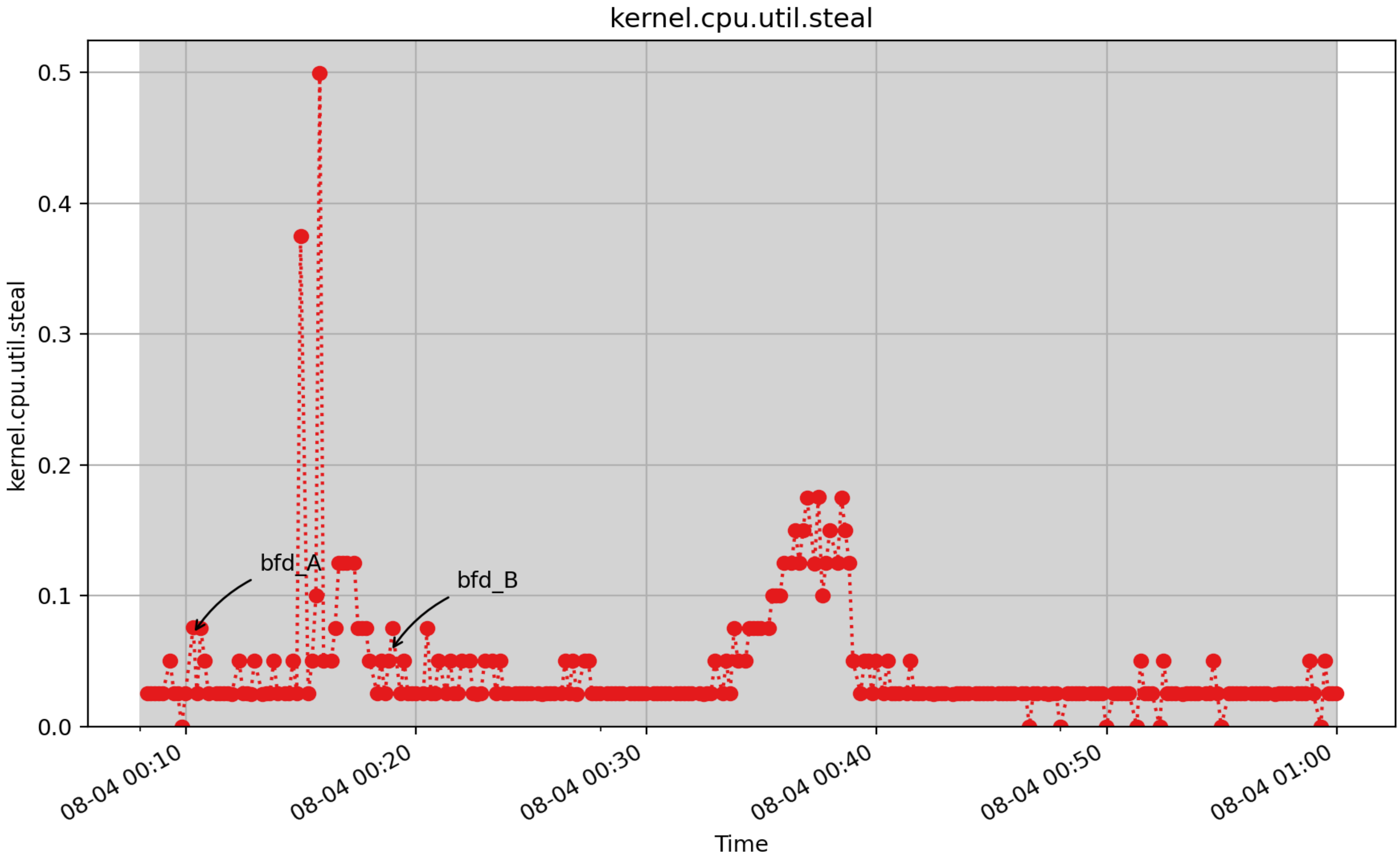
kernel.all.uptime: time the current kernel has been running (sec - DOUBLE)



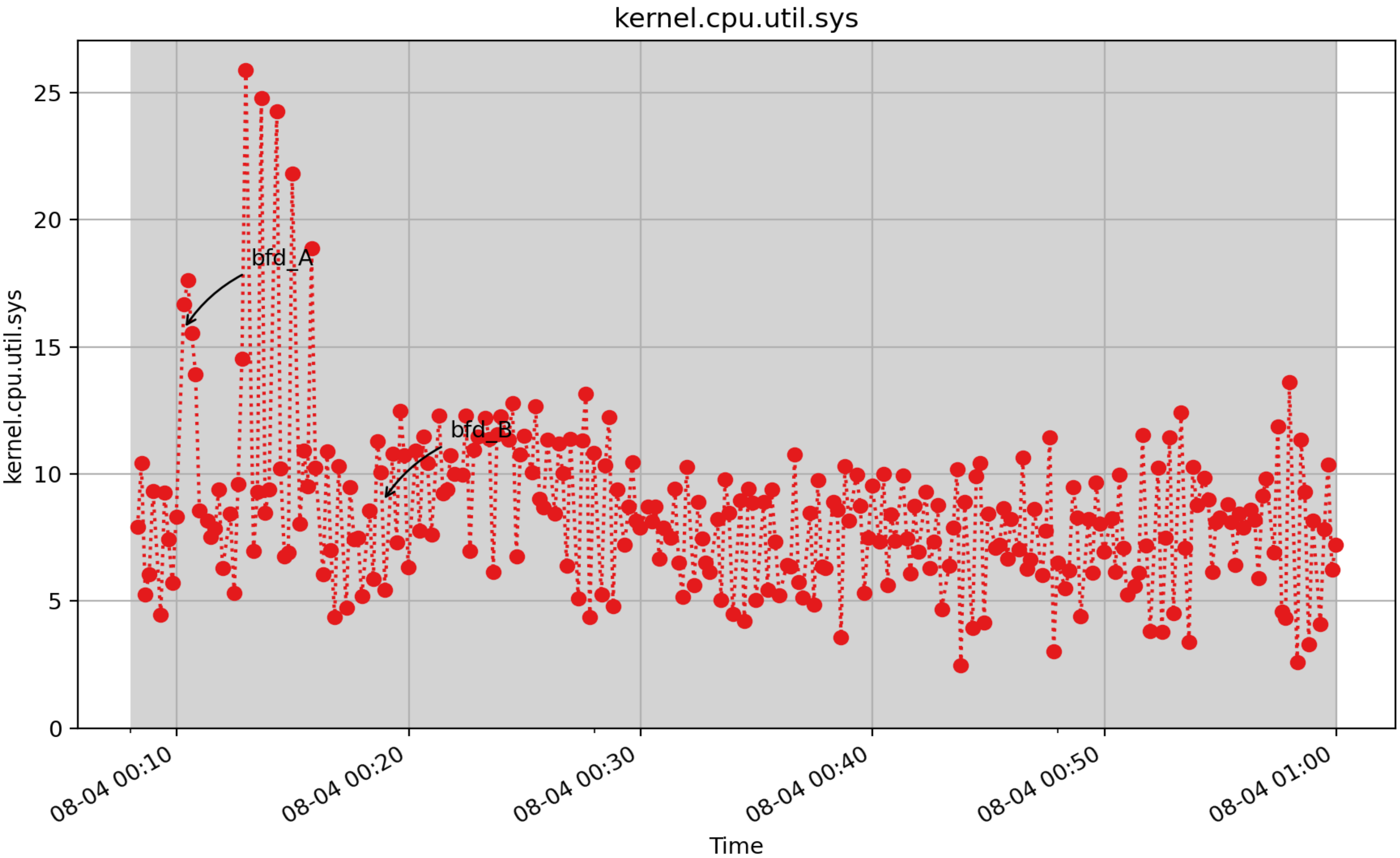
kernel.cpu.util.idle: percentage of idle time across all CPUs (- DOUBLE)



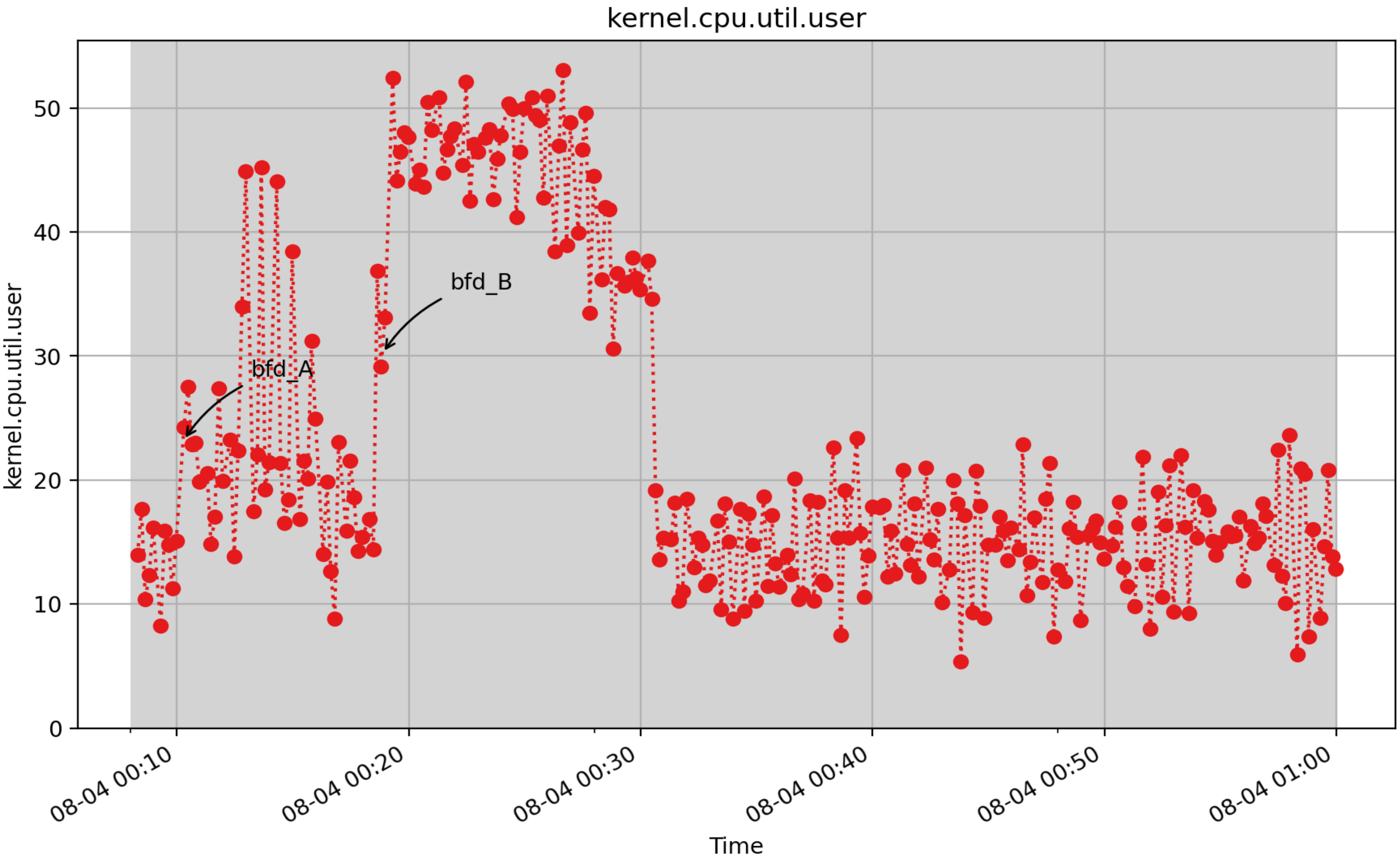
kernel.cpu.util.intr: Percentage of time spent processing interrupts across all CPUs. This value includes both soft and hard interrupt processing time. (- DOUBLE)



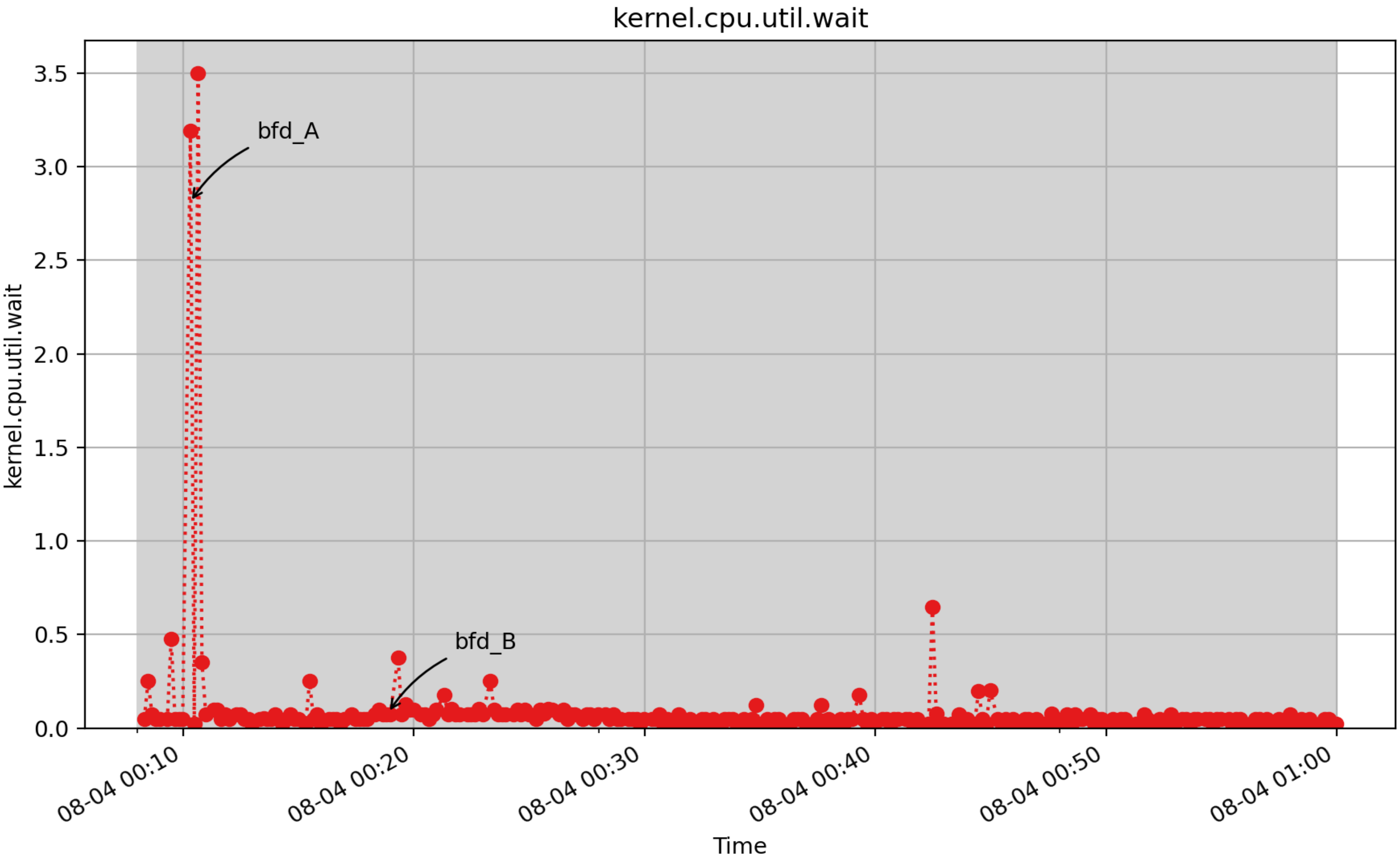
kernel.cpu.util.steal: Percentage of time across all CPUs when a CPU had a runnable process, but the hypervisor (virtualisation layer) chose to run something else instead. (- DOUBLE)



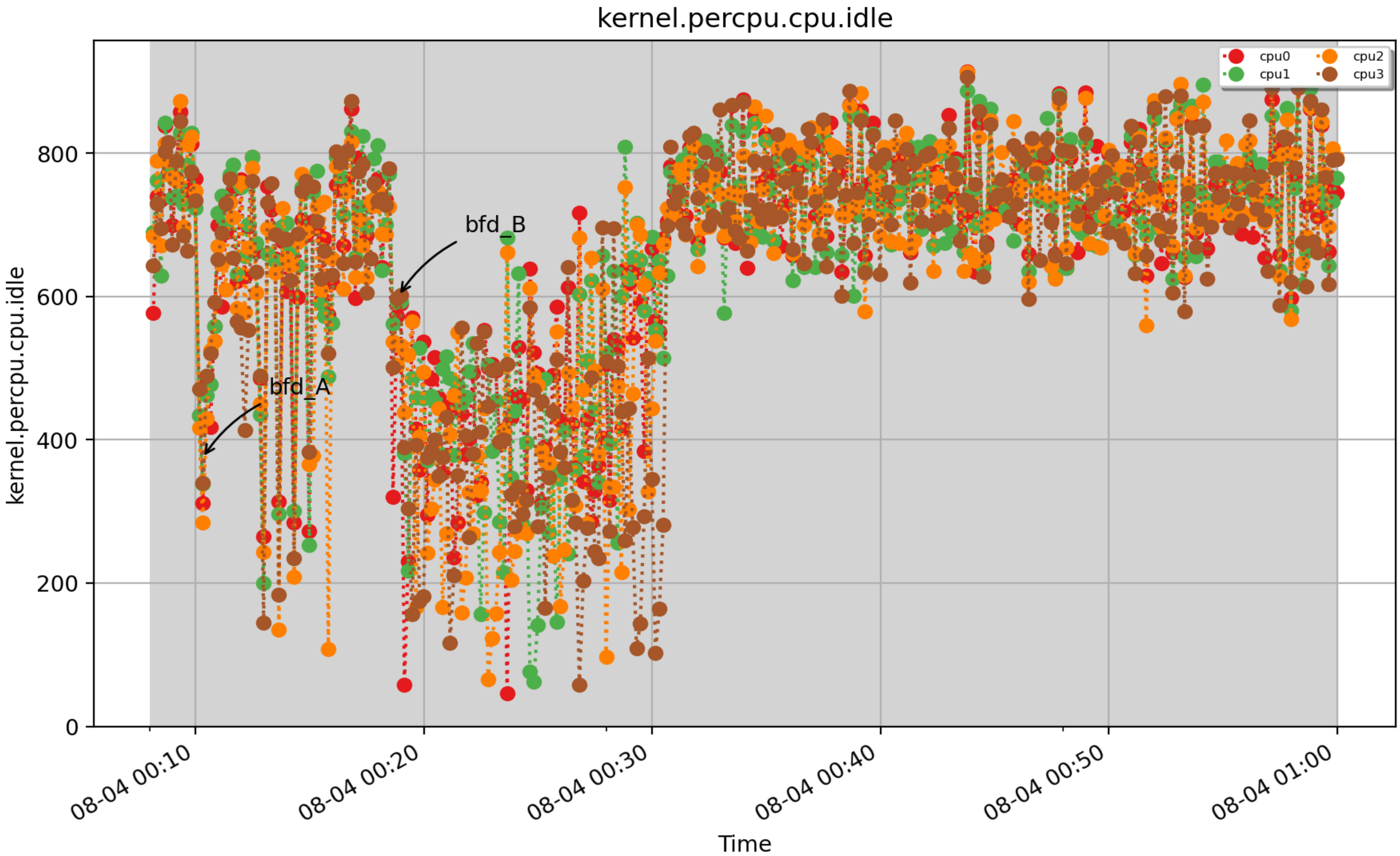
kernel.cpu.util.sys: percentage of sys time across all CPUs (- DOUBLE)



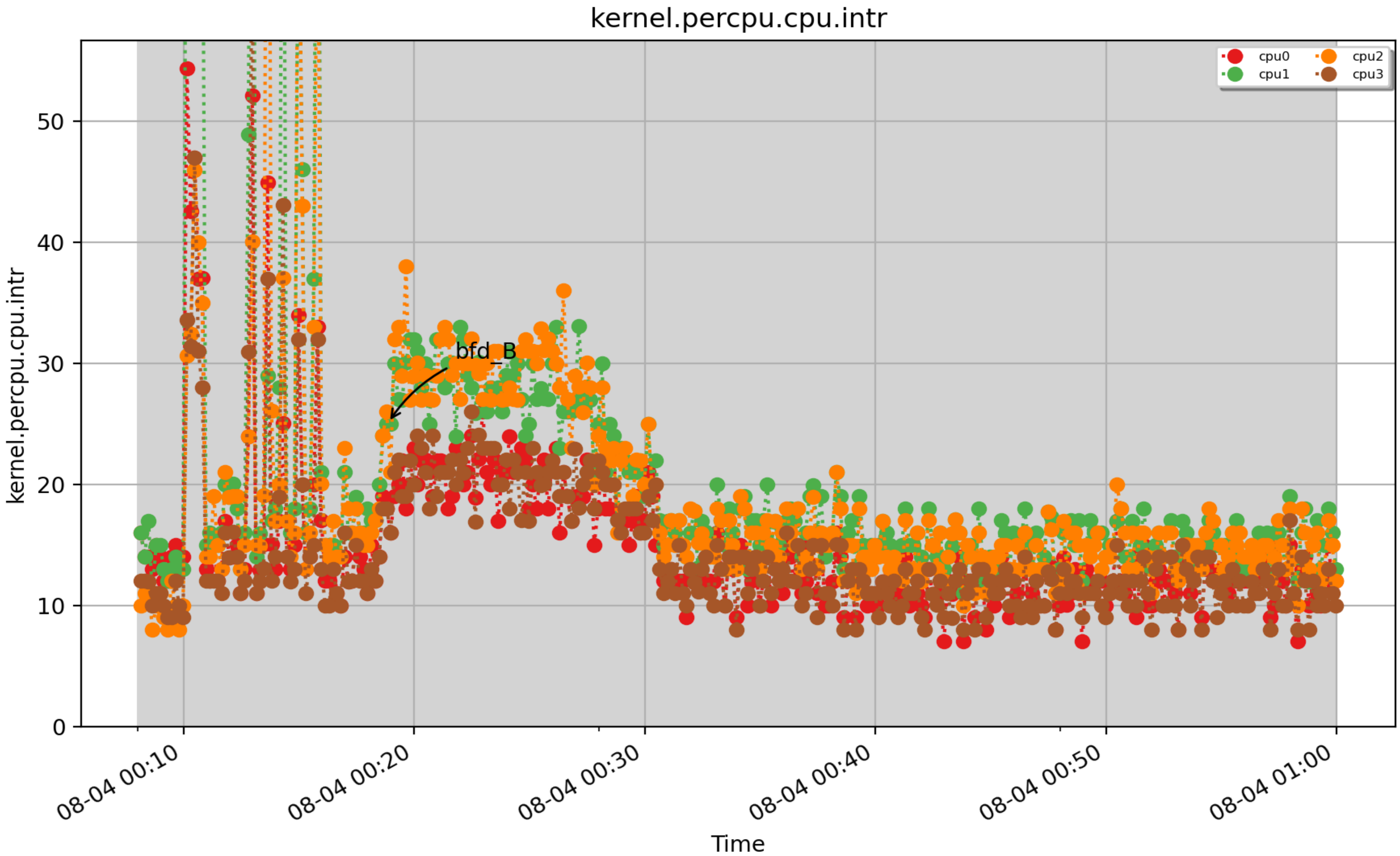
kernel.cpu.util.user: percentage of user time across all CPUs, including guest CPU time (- DOUBLE)



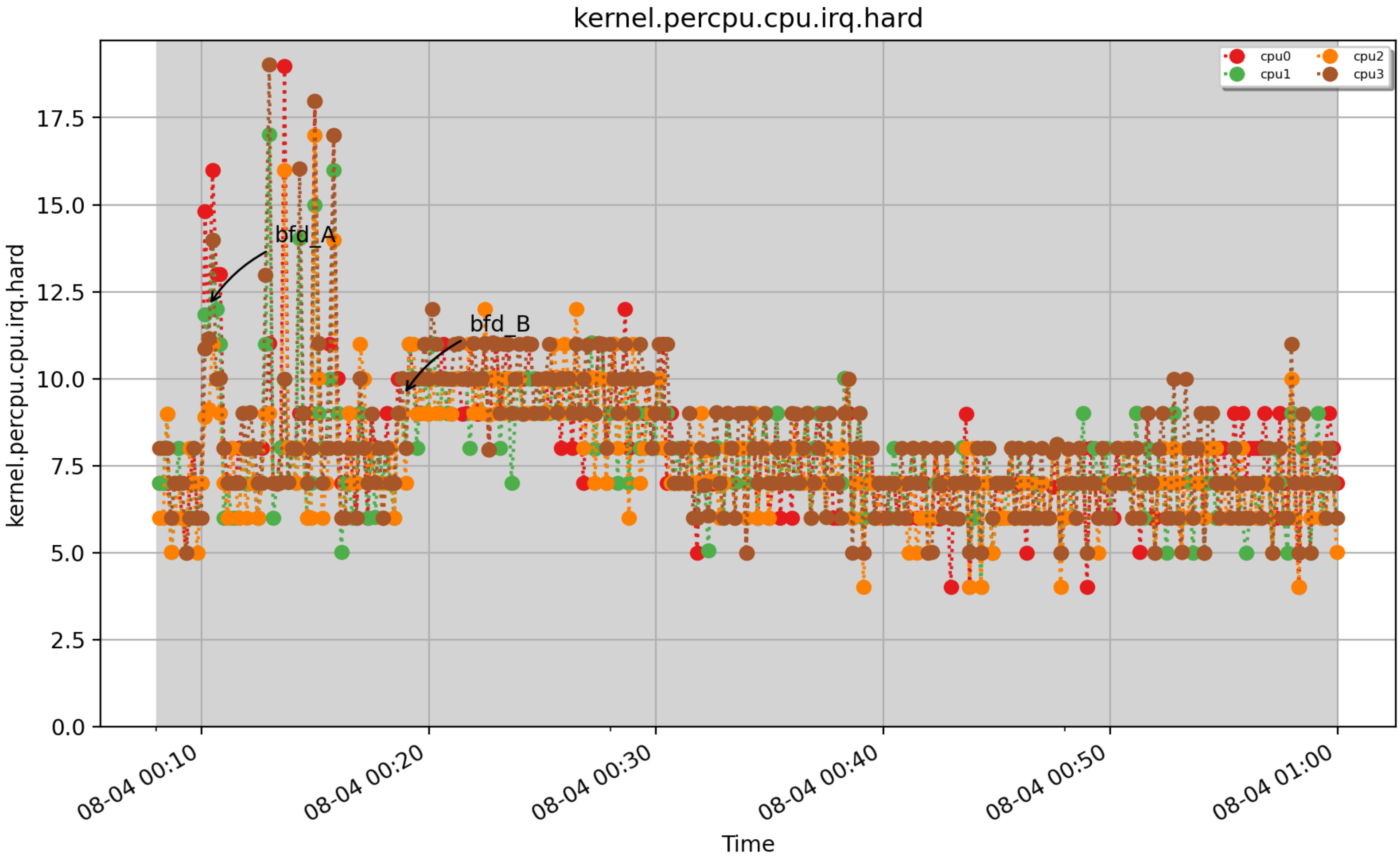
kernel.cpu.util.wait: percentage of wait time across all CPUs (- DOUBLE)



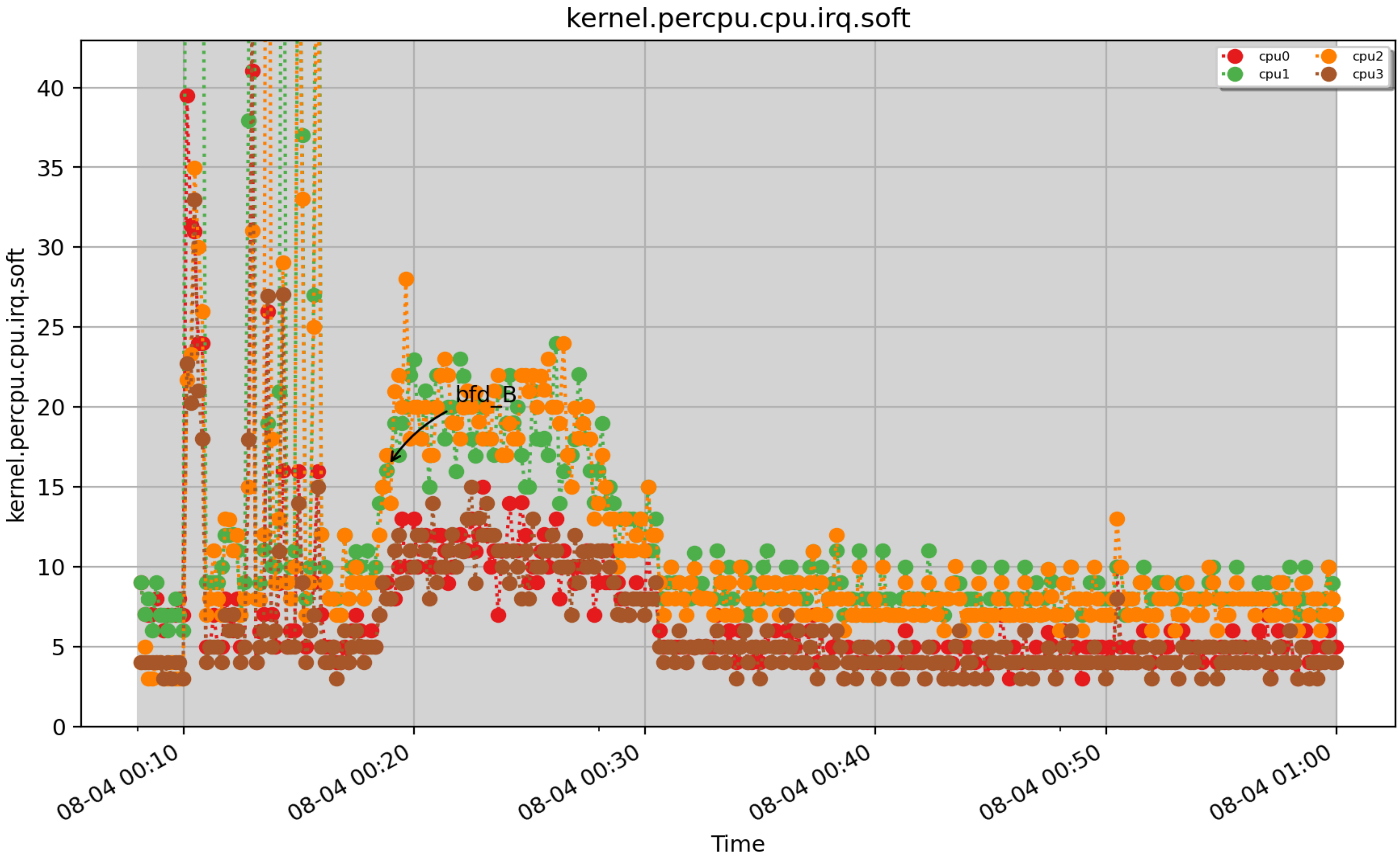
kernel.percpu.cpu.idle: percpu idle CPU time metric from /proc/stat (millisec - U64) - *rate converted*



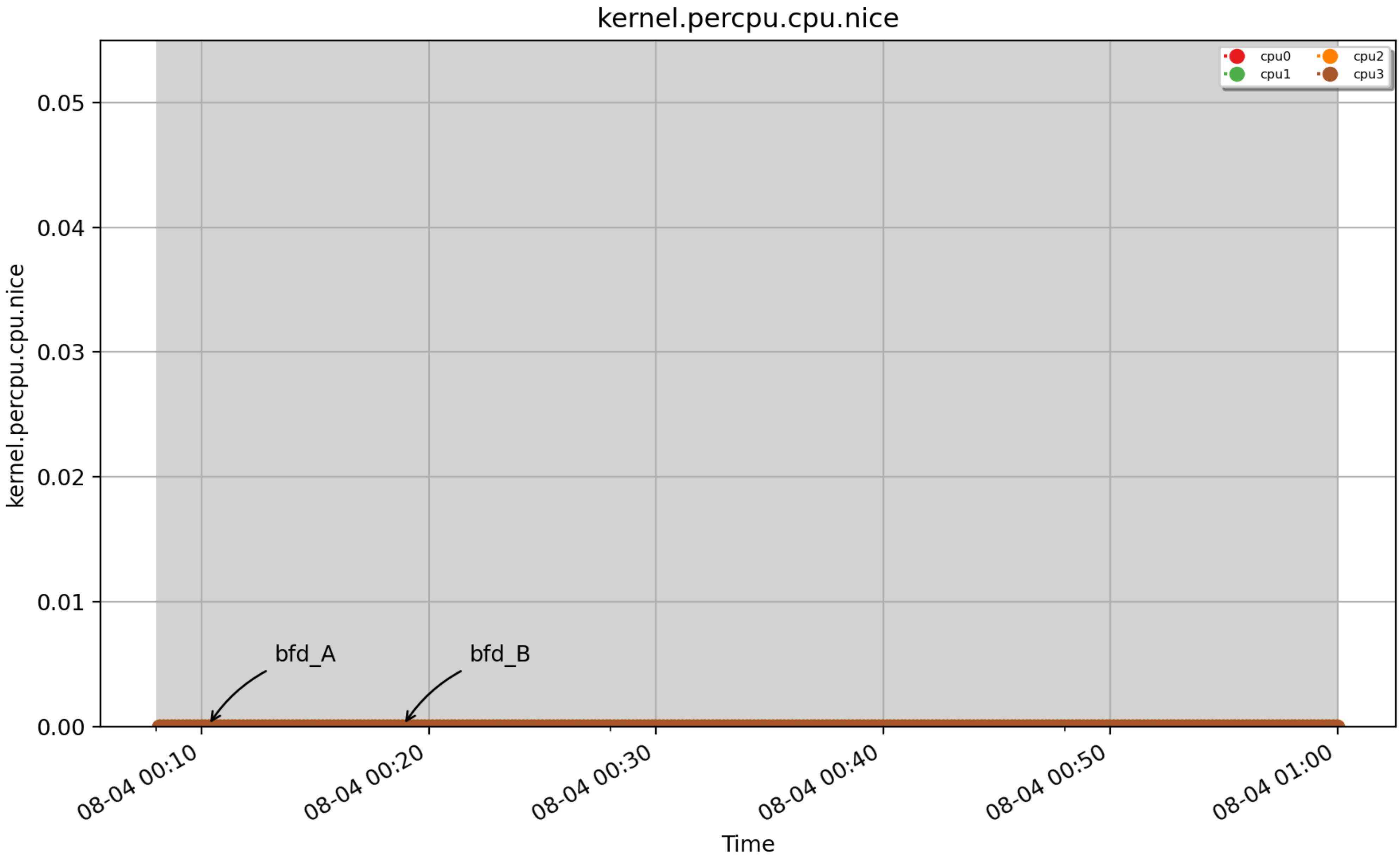
kernel.percpu.cpu.intr: Total time spent processing interrupts on each CPU (this includes both soft and hard interrupt processing time). (millisec - U64) - *rate converted*



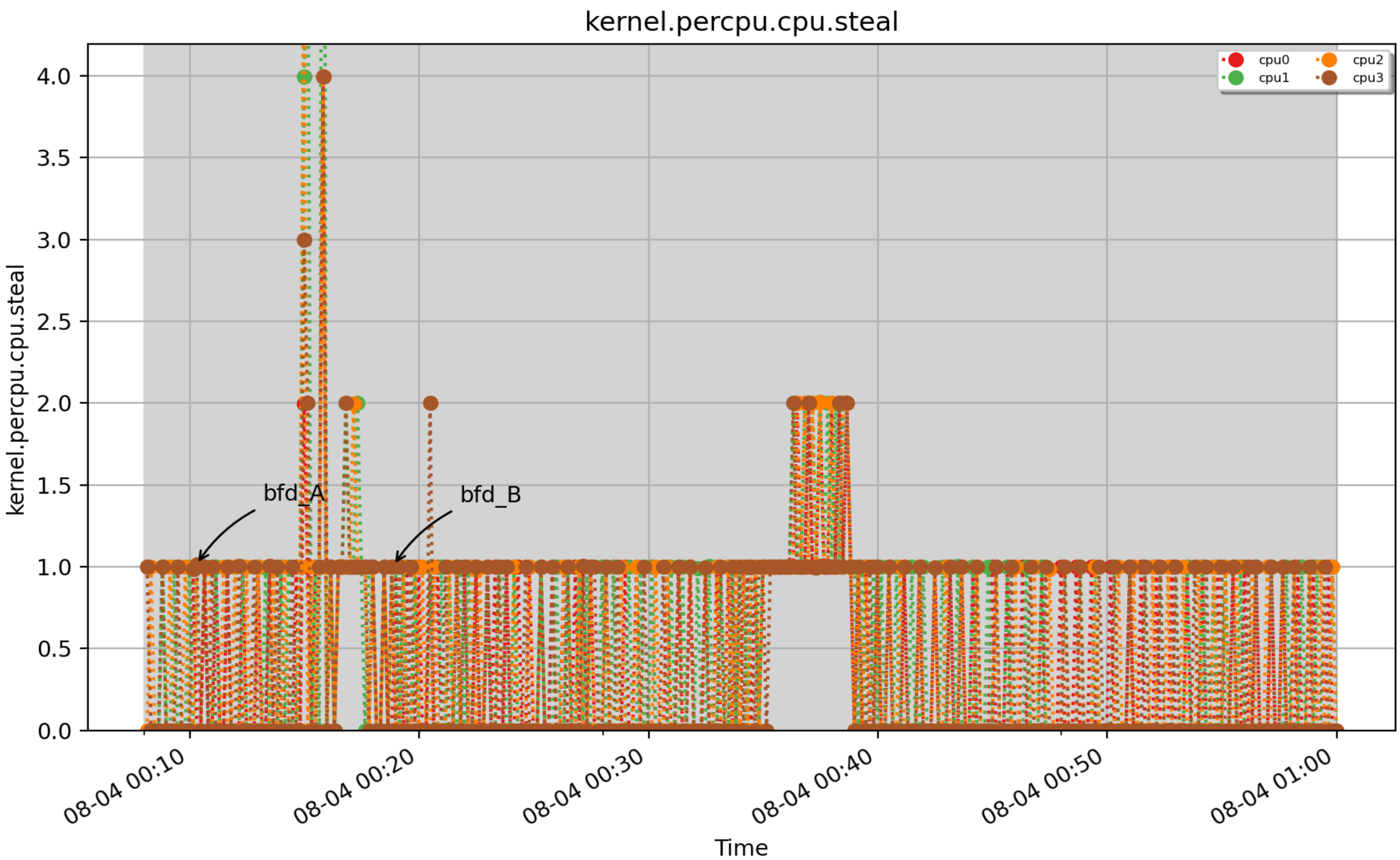
kernel.percpu.cpu.irq.hard: Per-CPU hard interrupt CPU time ("hard" interrupt handling code is the code run directly on receipt of the initial hardware interrupt, and does not include "soft" interrupt handling code which is deferred until later). (millisec - U64) - *rate converted*



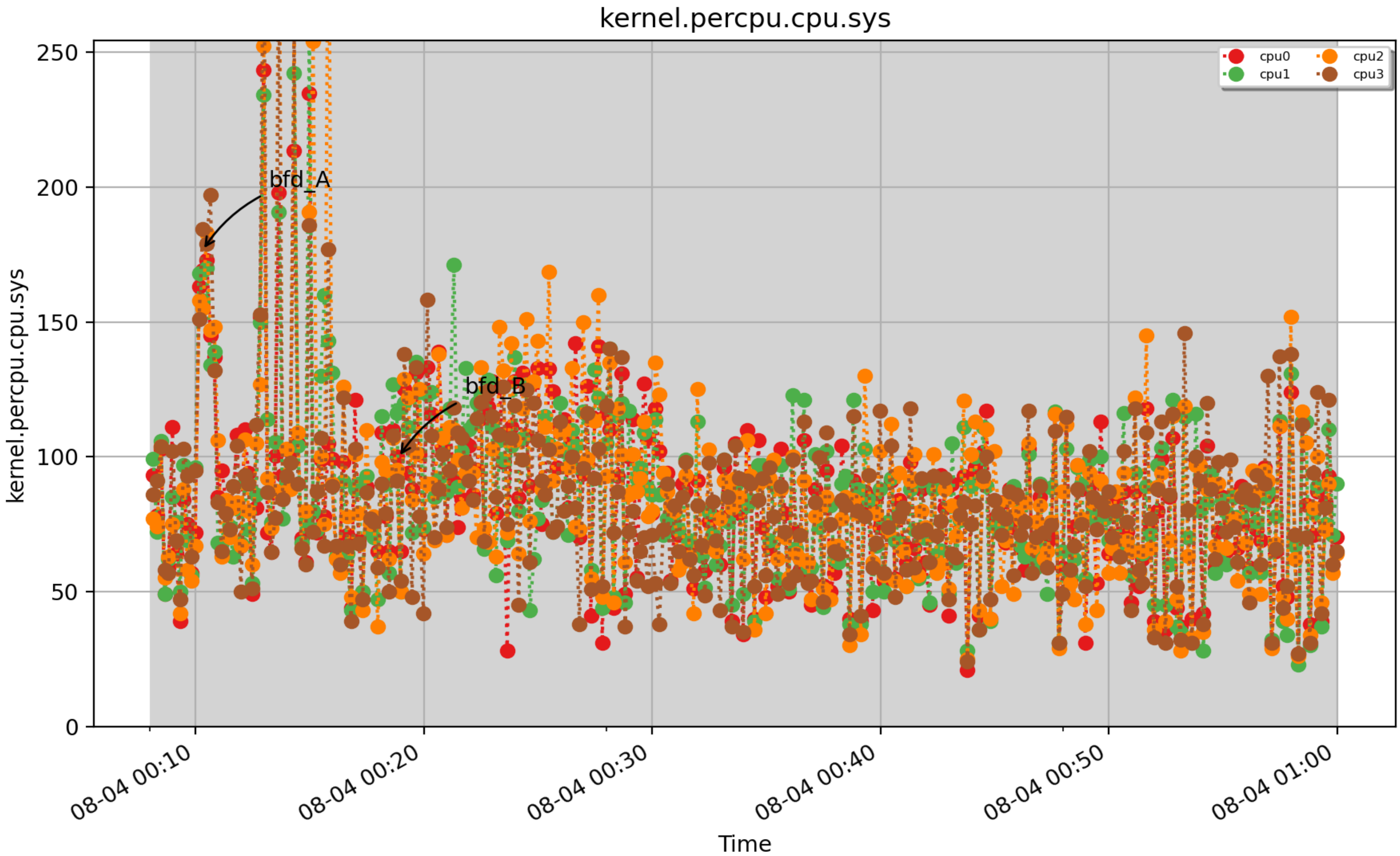
kernel.percpu.cpu.irq.soft: Per-CPU soft interrupt CPU time (deferred interrupt handling code, not run in the initial interrupt handler). (millisec - U64) - *rate converted*



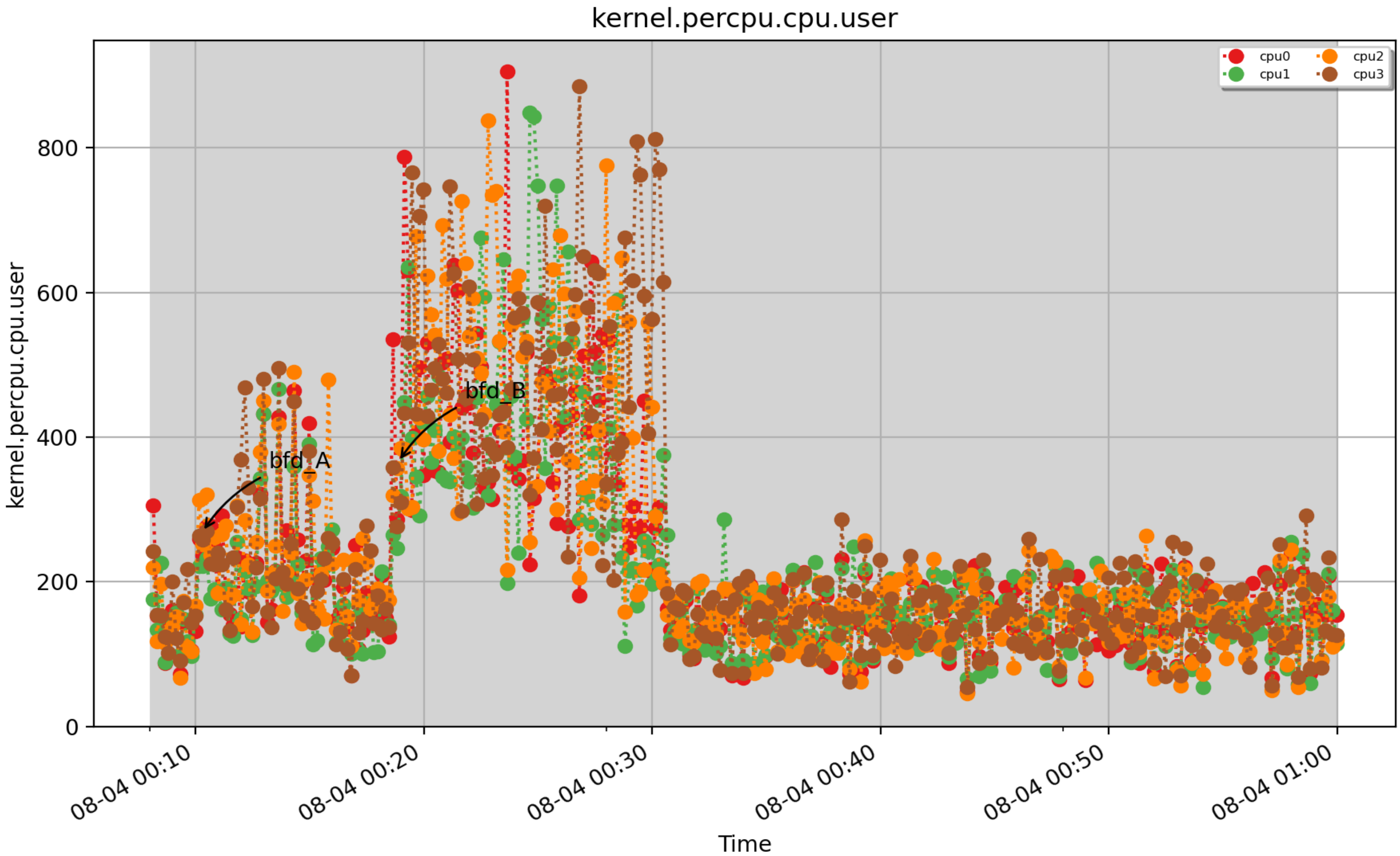
kernel.percpu.cpu.nice: percpu nice user CPU time metric from /proc/stat, including guest CPU time (millisec - U64) - *rate converted*



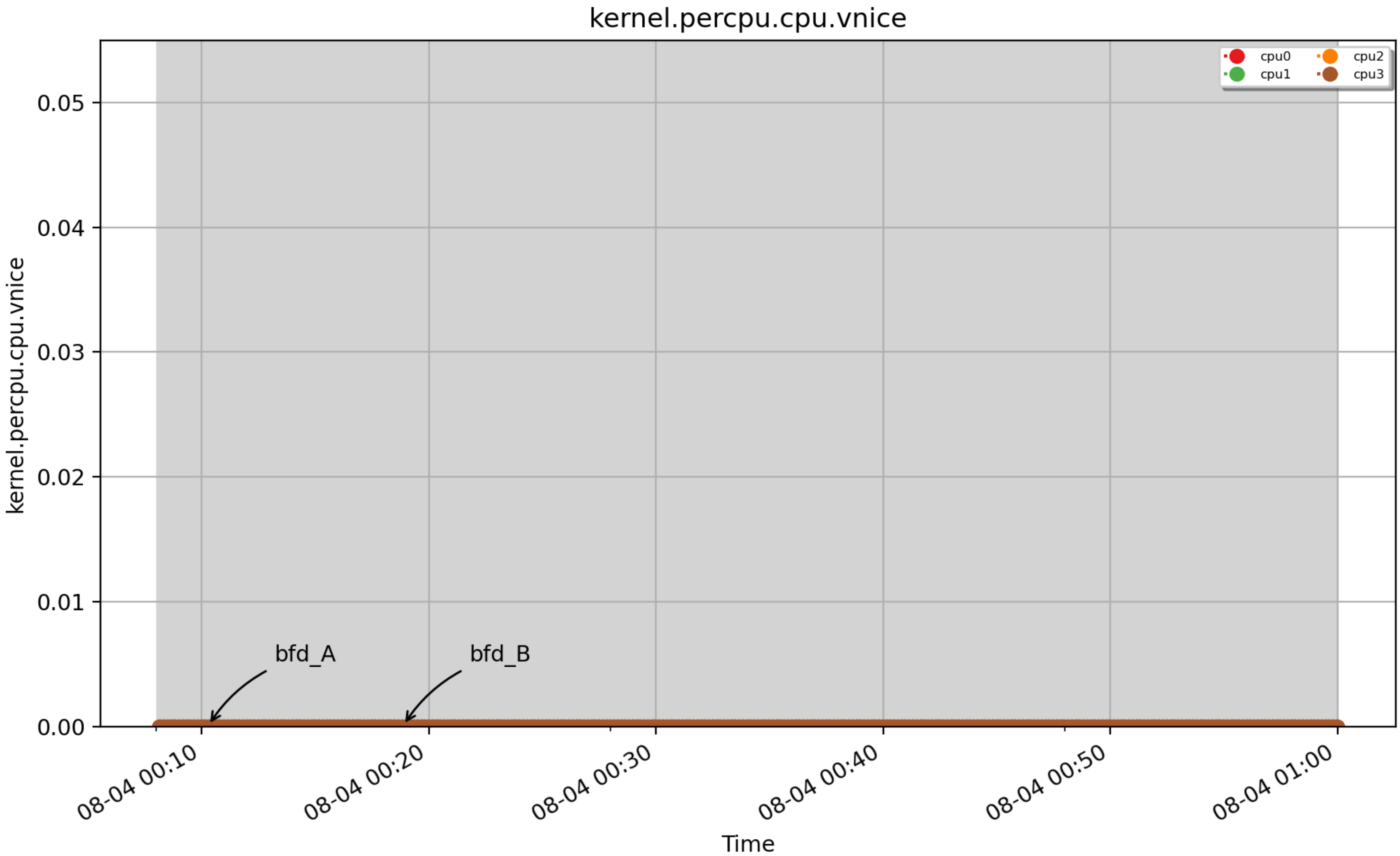
kernel.percpu.cpu.steal: Per-CPU time when the CPU had a runnable process, but the hypervisor (virtualisation layer) chose to run something else instead. (millisec - U64) - *rate converted*



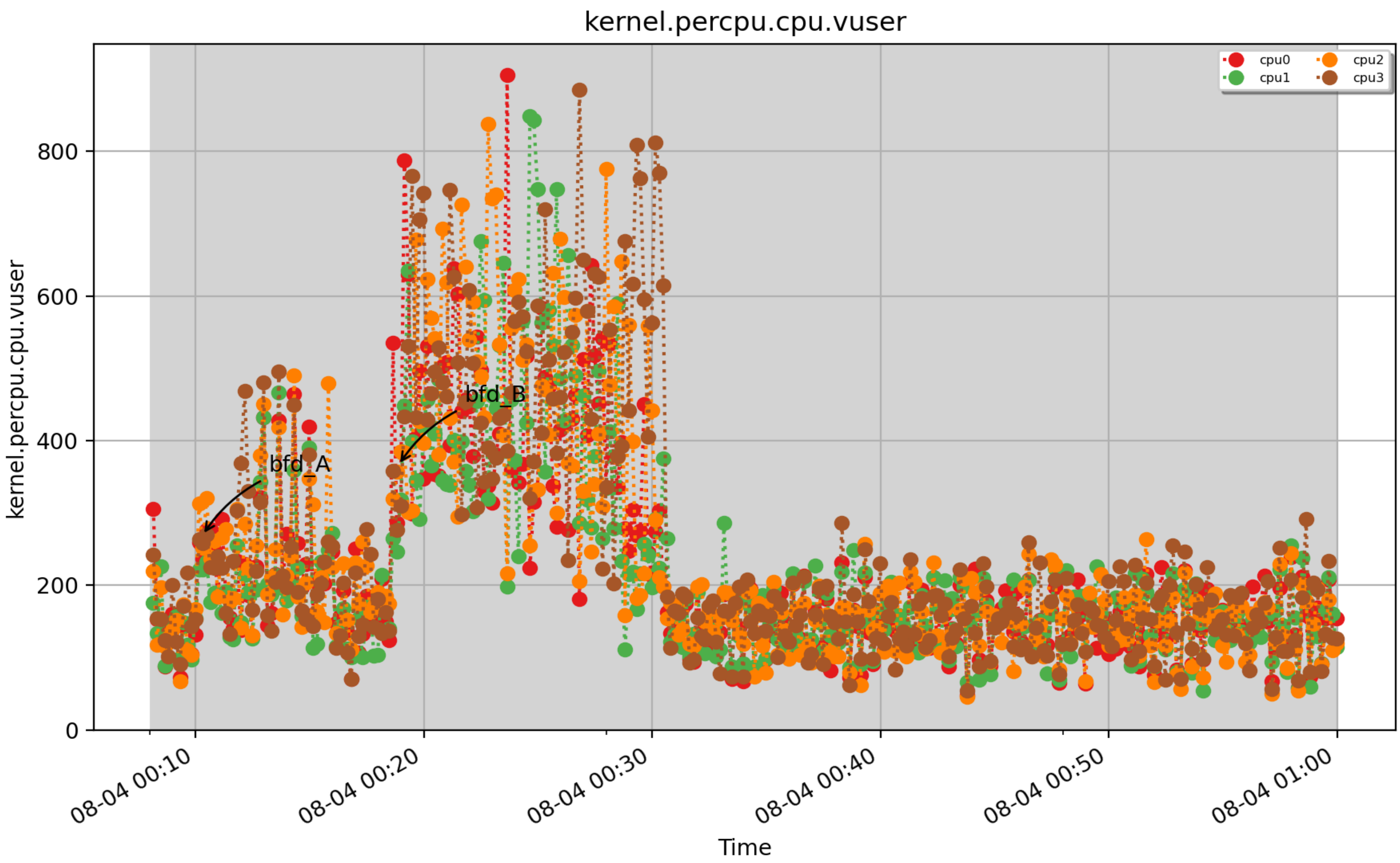
kernel.percpu.cpu.sys: percpu sys CPU time metric from /proc/stat (millisec - U64) - *rate converted*



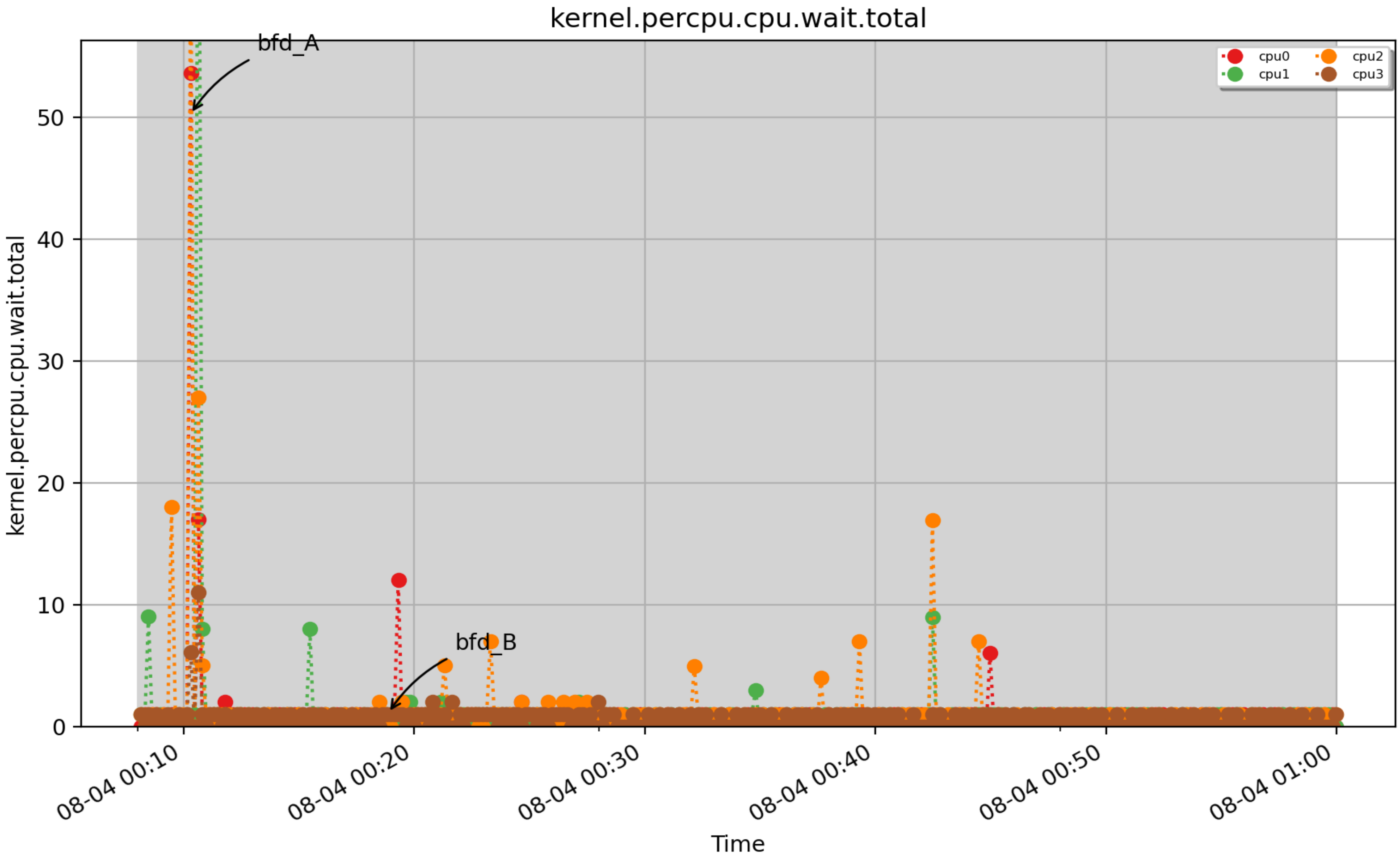
kernel.percpu.cpu.user: percpu user CPU time metric from /proc/stat, including guest CPU time (millisec - U64) - *rate converted*



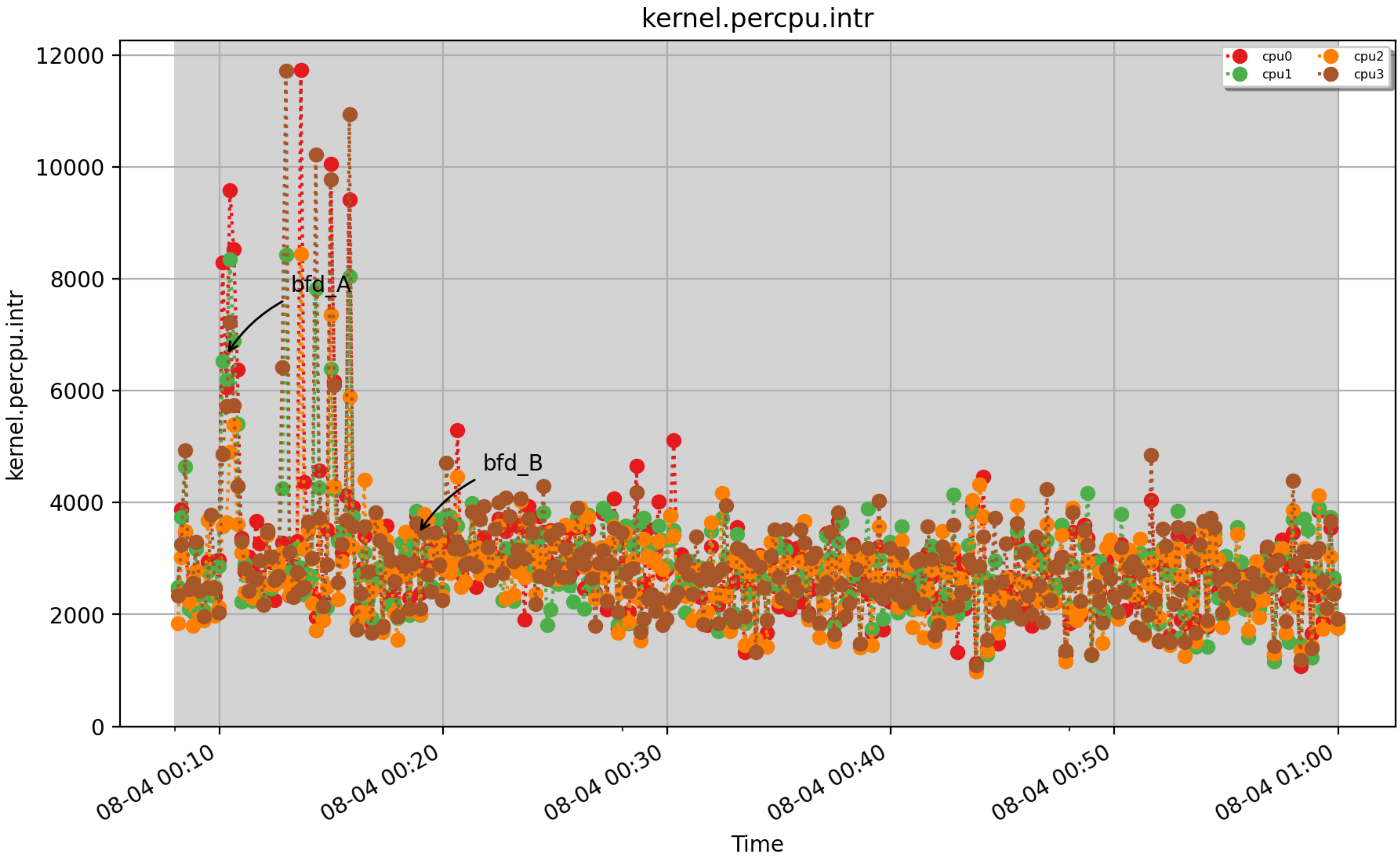
kernel.percpu.cpu.vnice: percpu nice user CPU time metric from /proc/stat, excluding guest CPU time (millisec - U64) - *rate converted*



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

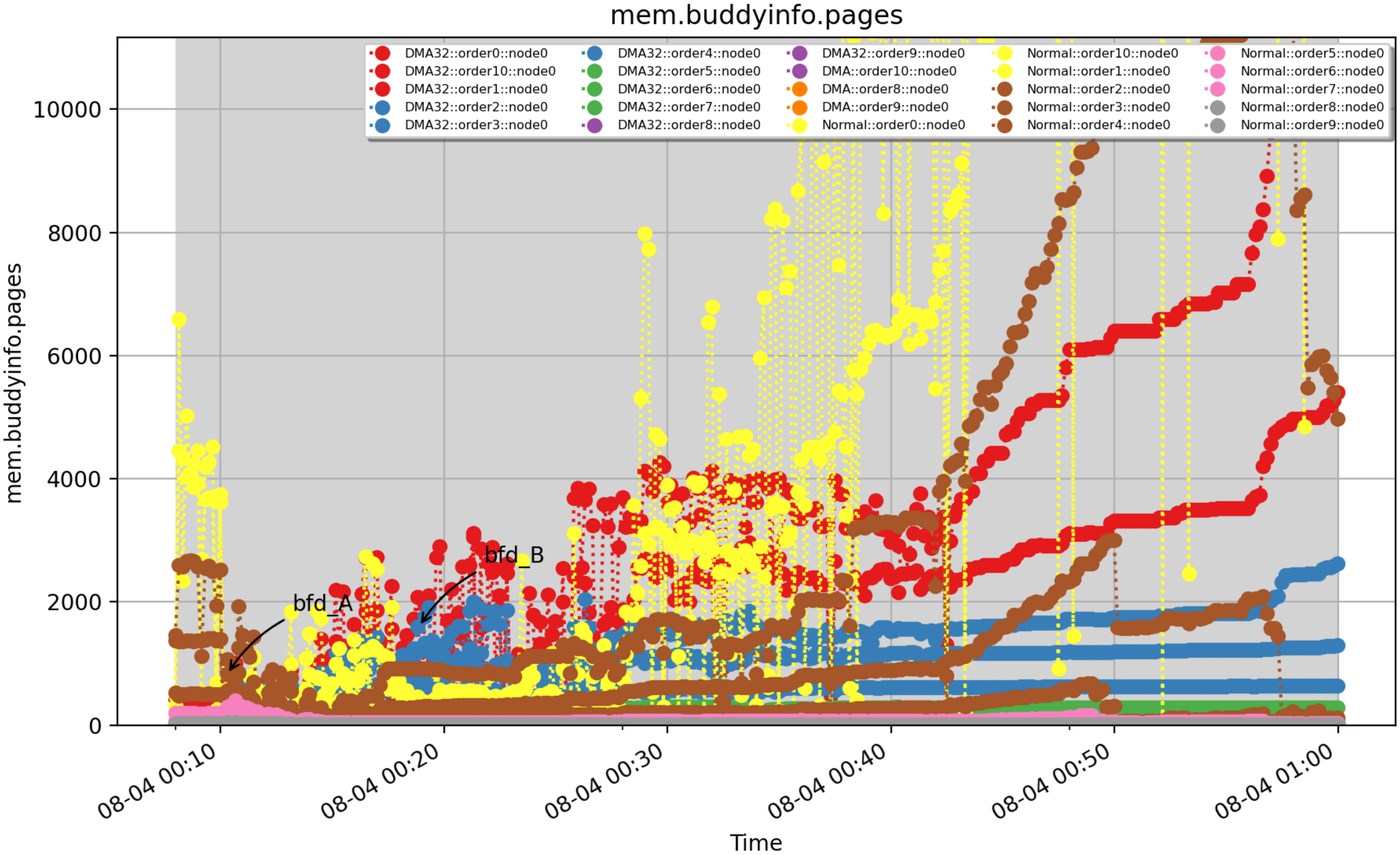


kernel.percpu.cpu.wait.total: Per-CPU I/O wait CPU time - time spent with outstanding I/O requests. (millisec - U64) - *rate converted*

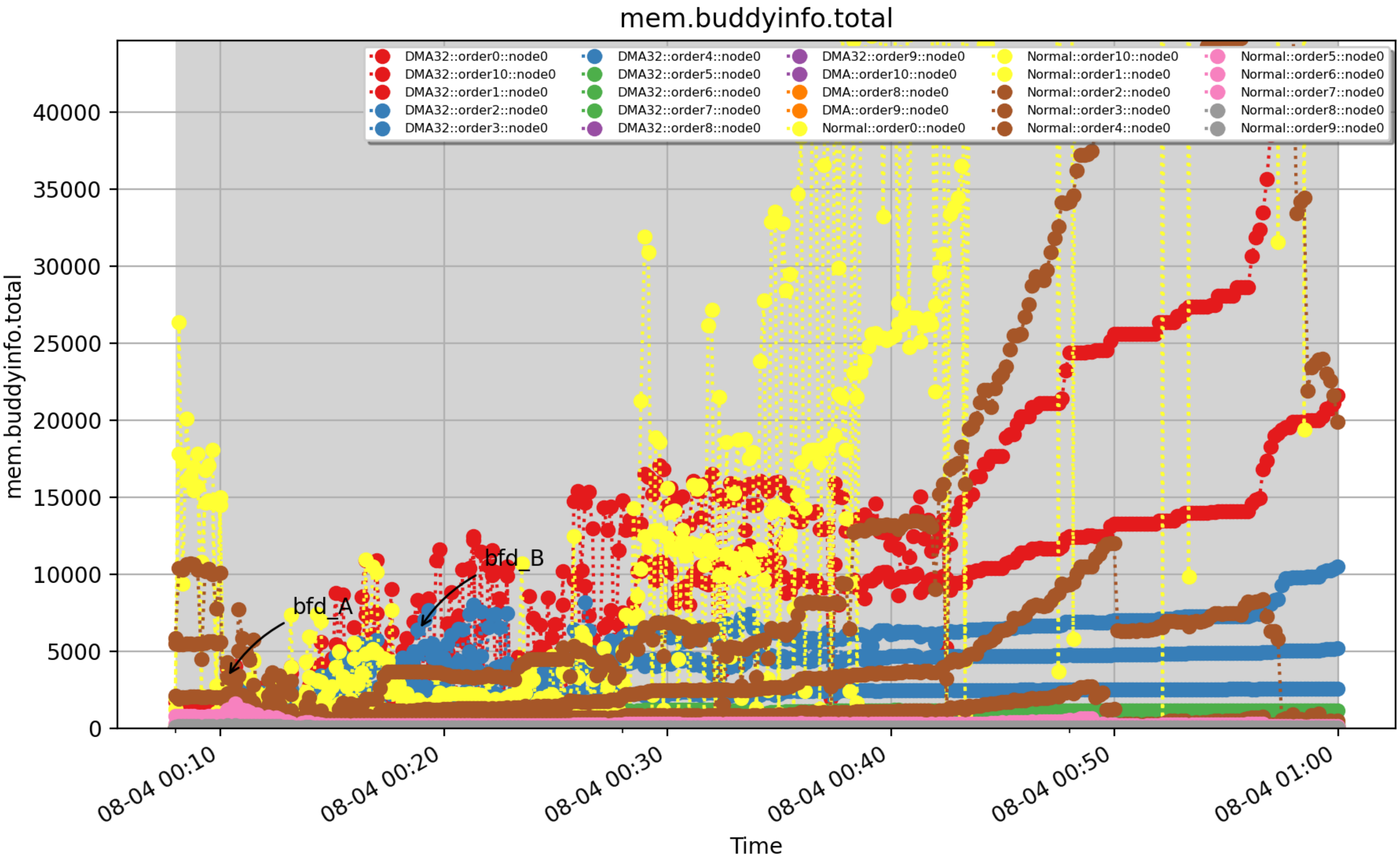


kernel.percpu.intr: Aggregate count of each CPUs interrupt processing count, calculated as the sum of all interrupt types in /proc/interrupts for each CPU. (count - U64) - *rate converted*

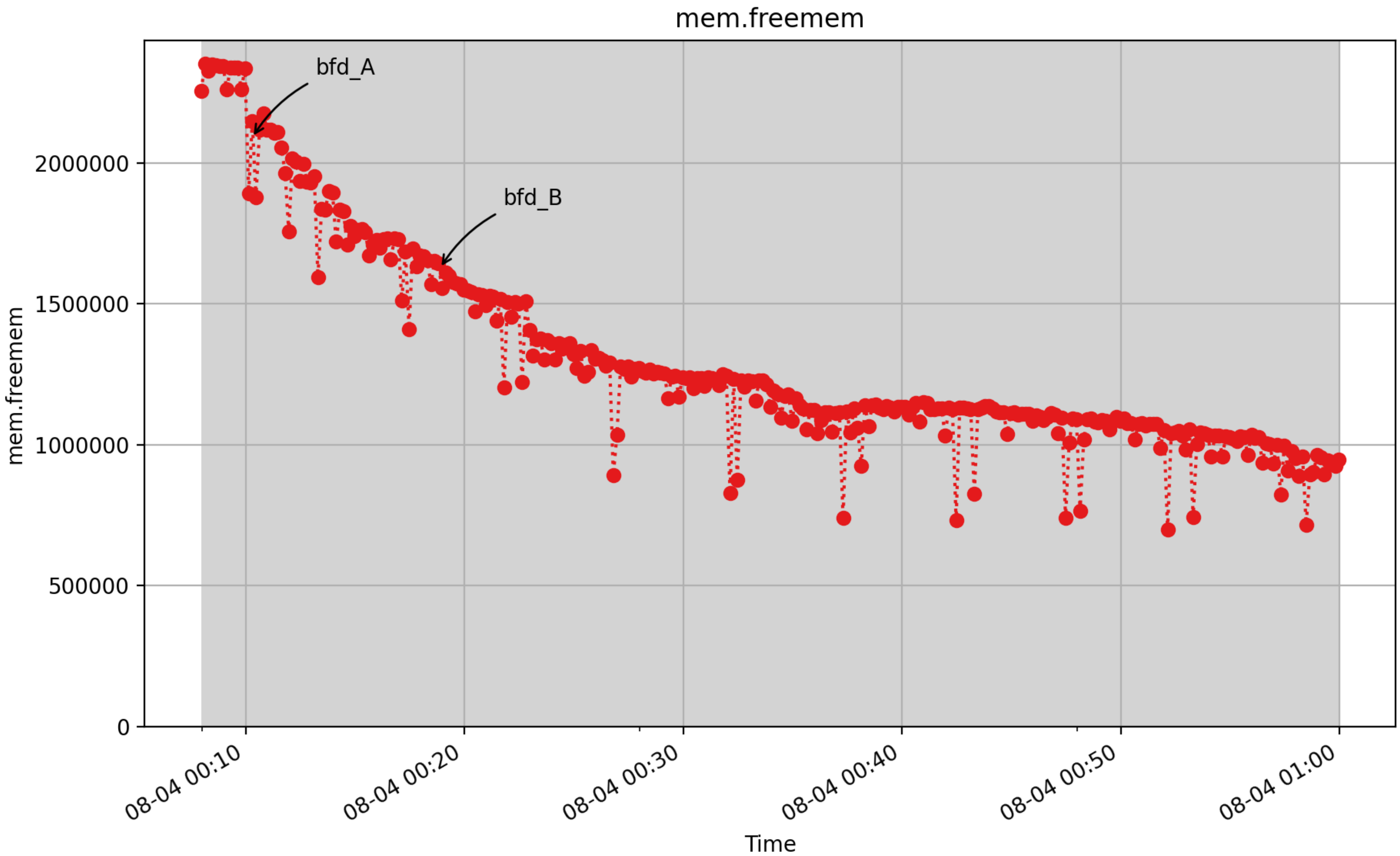
Mem



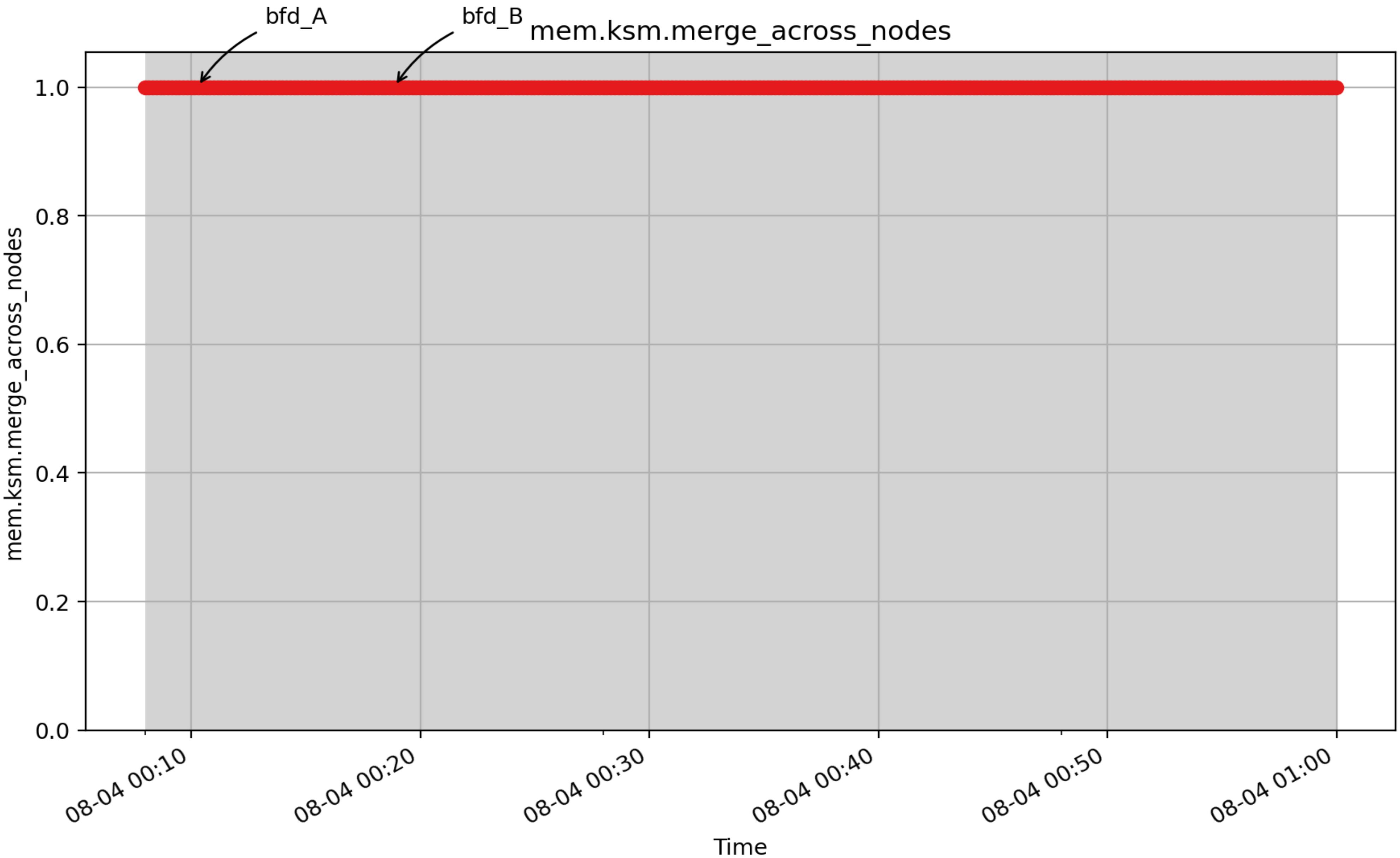
mem.buddyinfo.pages: fragmented page count from /proc/buddyinfo (- U64)



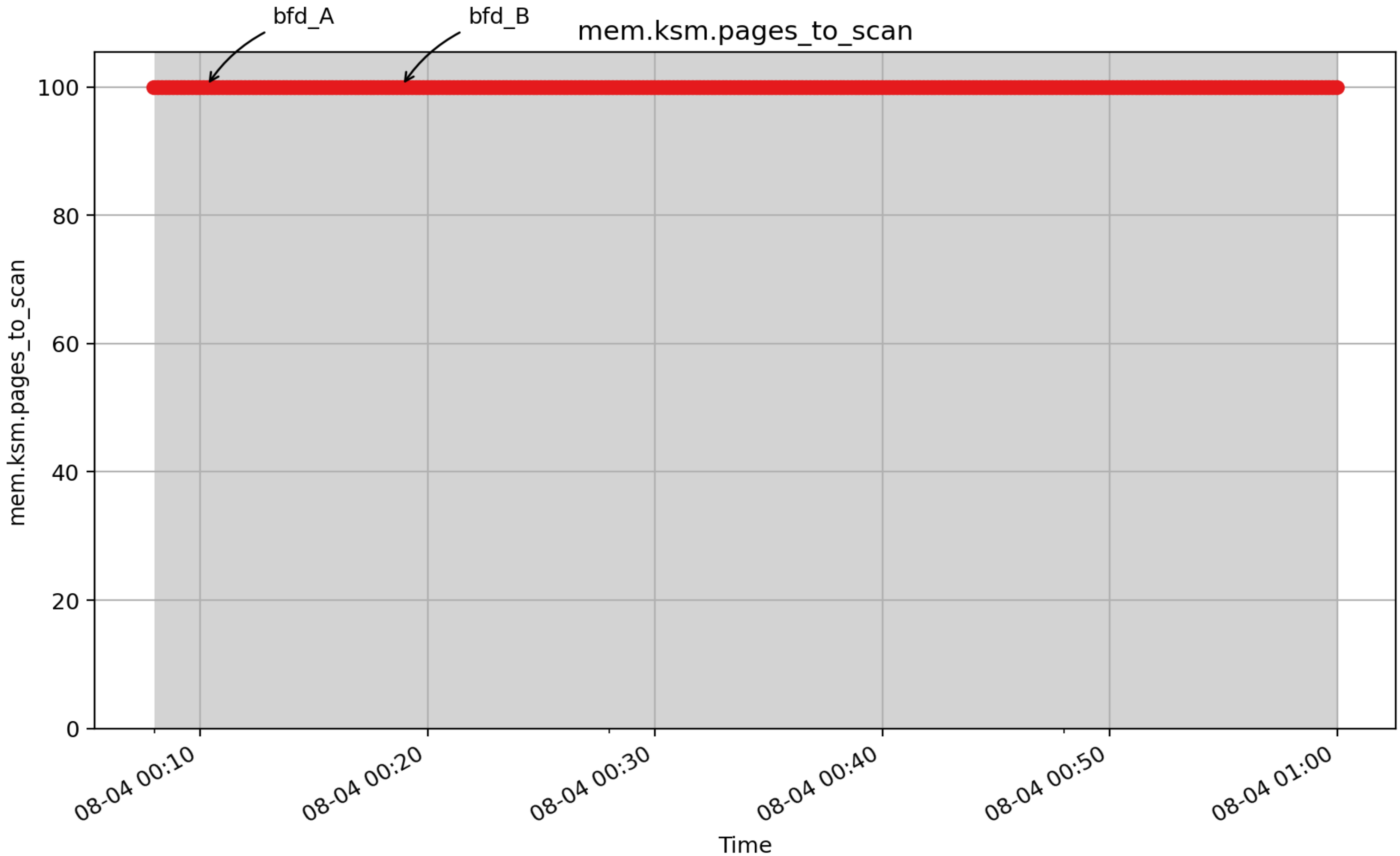
mem.buddyinfo.total: page fragmentation size from /proc/buddyinfo (Kbyte - U64)



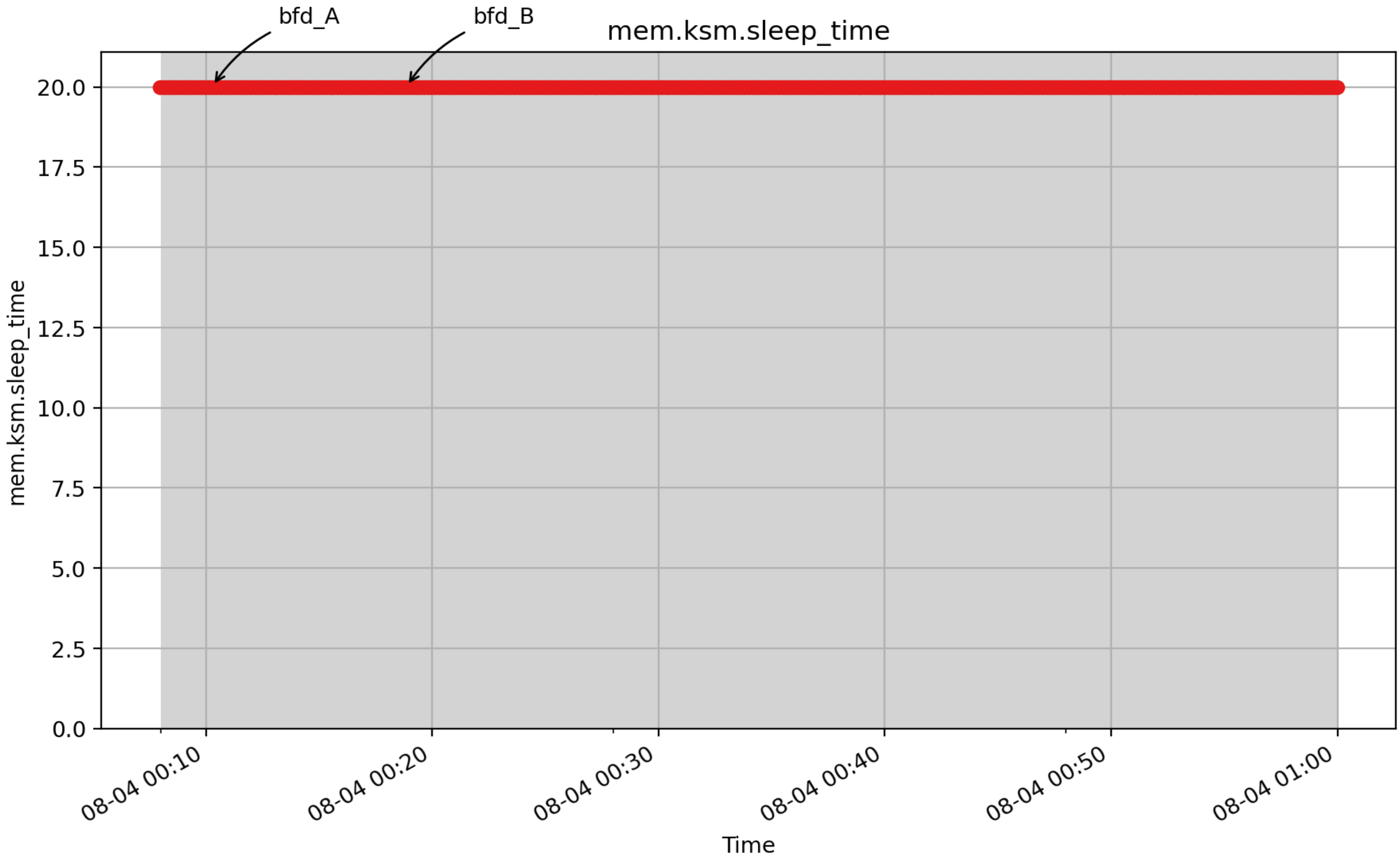
mem.freemem: free system memory metric from /proc/meminfo (Kbyte - U64)



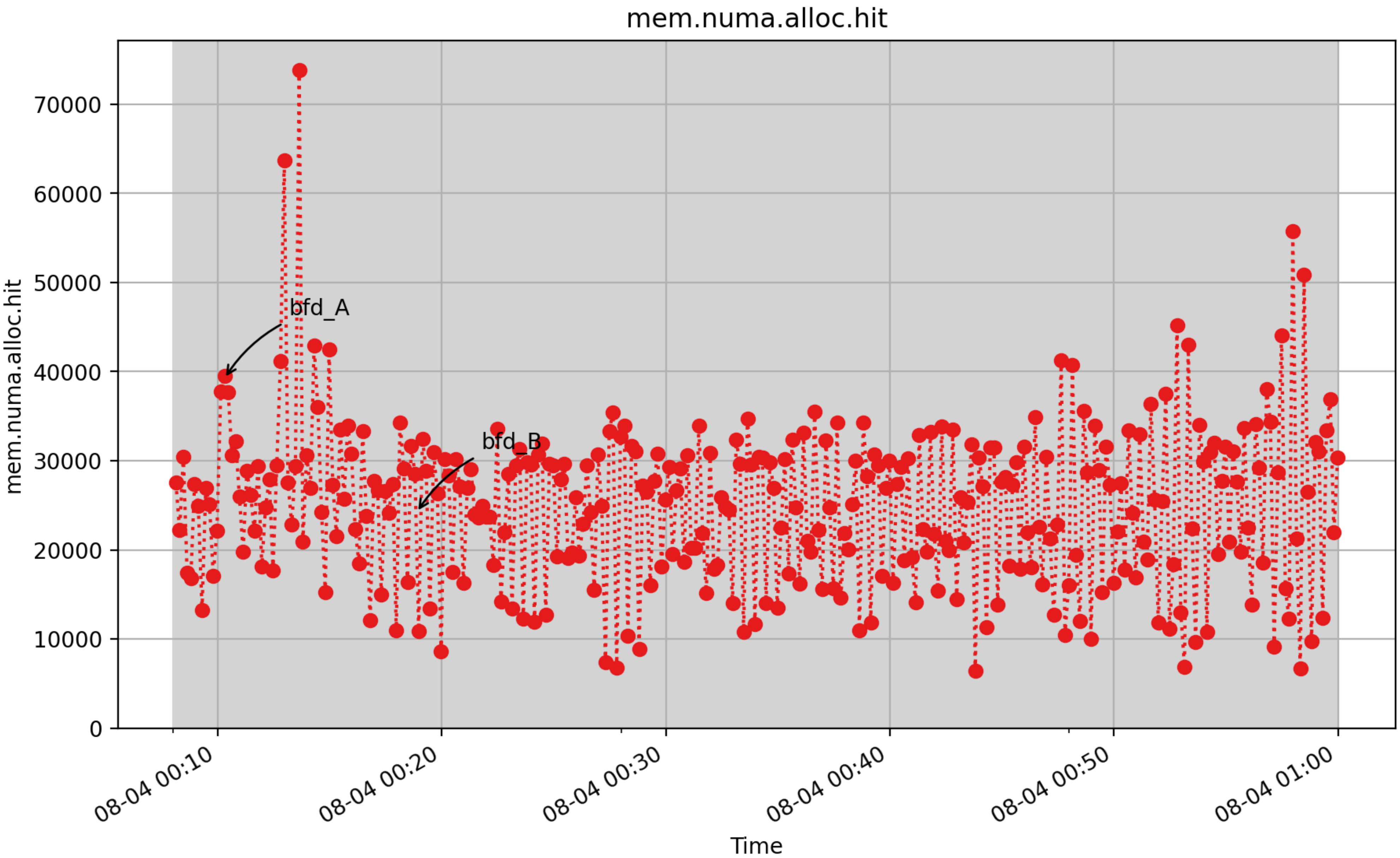
mem.ksm.merge_across_nodes: Kernel allows merging across NUMA nodes (- U64)



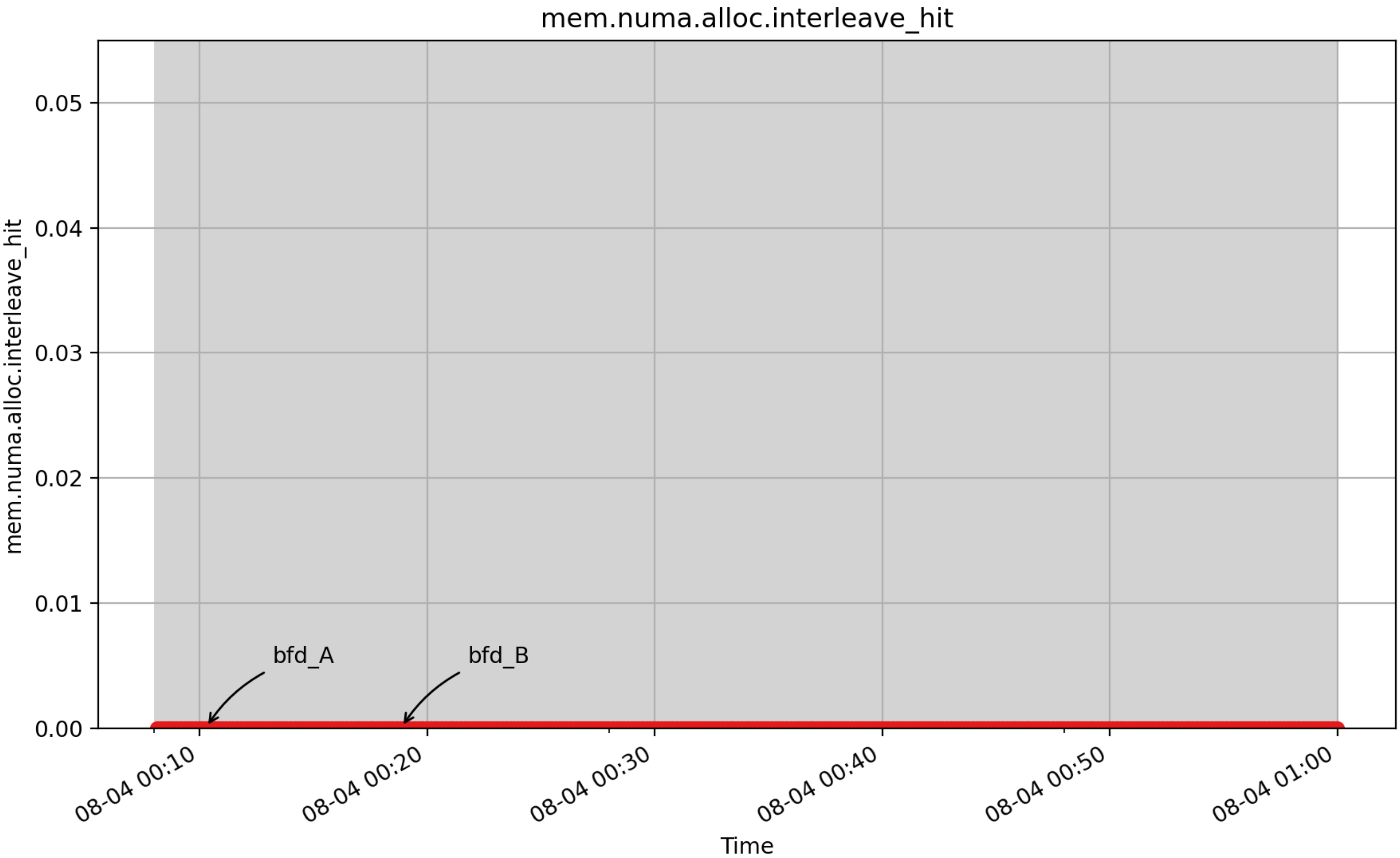
mem.ksm.pages_to_scan: Number of pages to scan at a time (- U64)



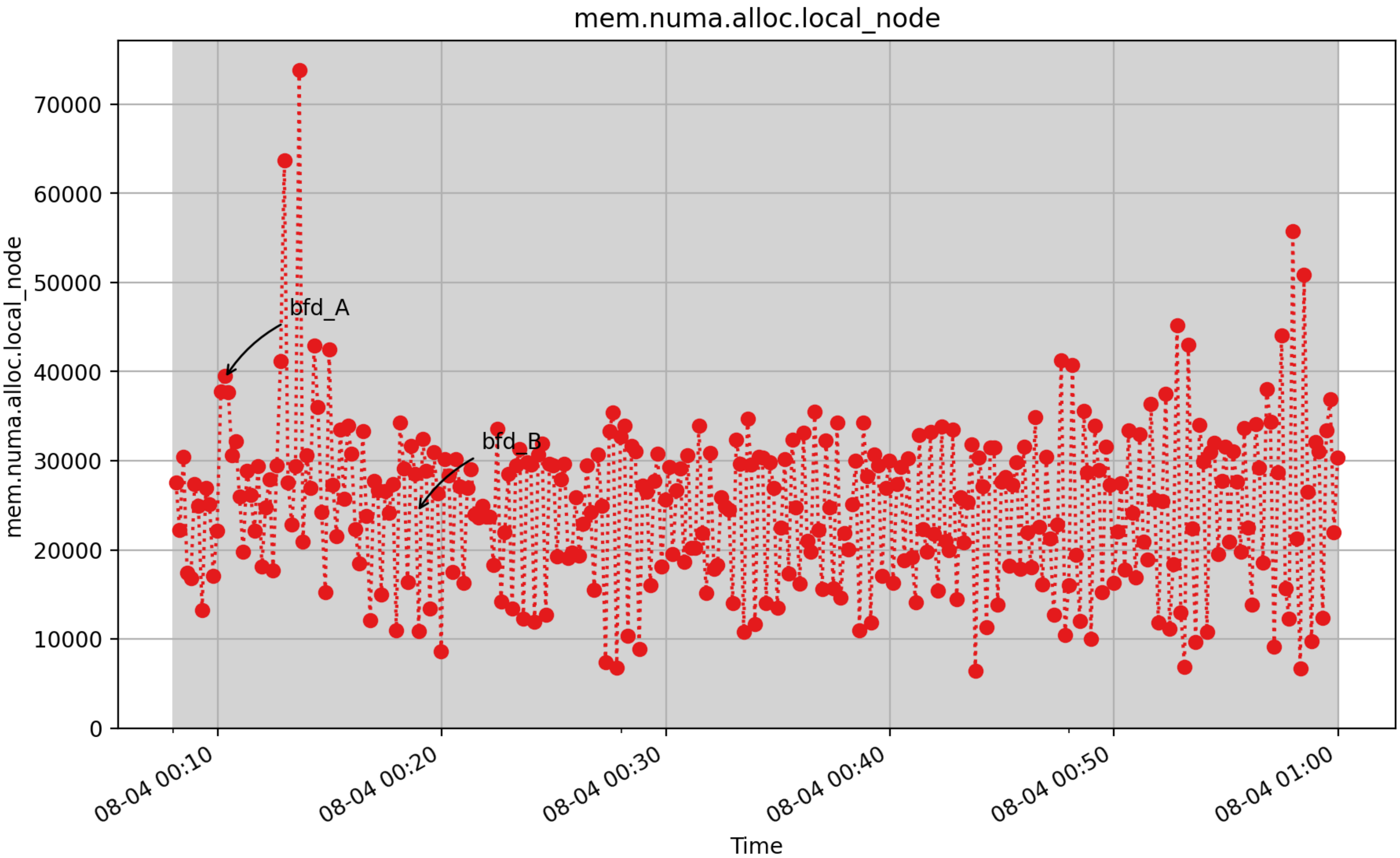
mem.ksm.sleep_time: Time ksmd should sleep between batches (millisec - U64)



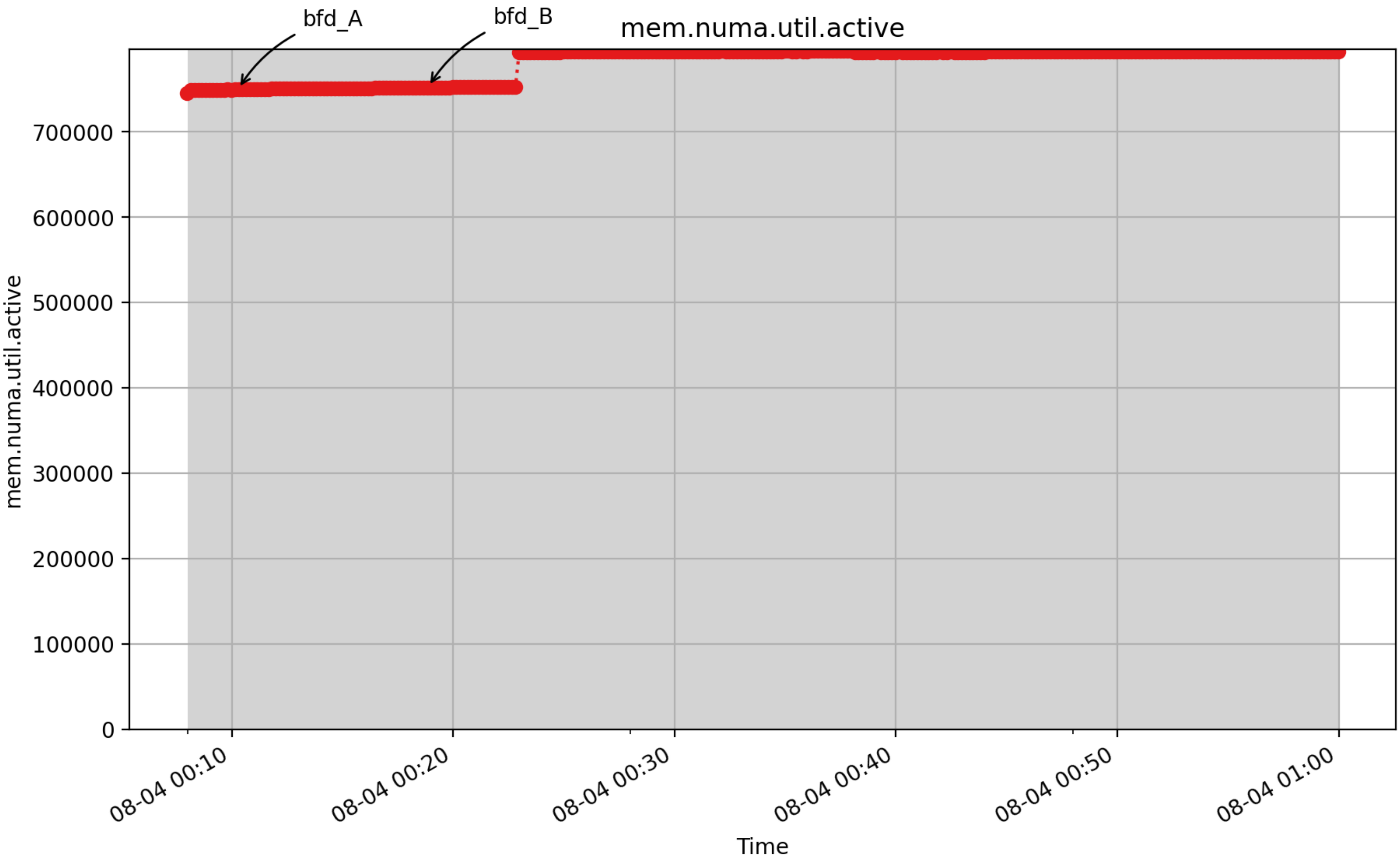
mem.numa.alloc.hit: per-node count of times a task wanted alloc on local node and succeeded (count - U64) - *rate converted*



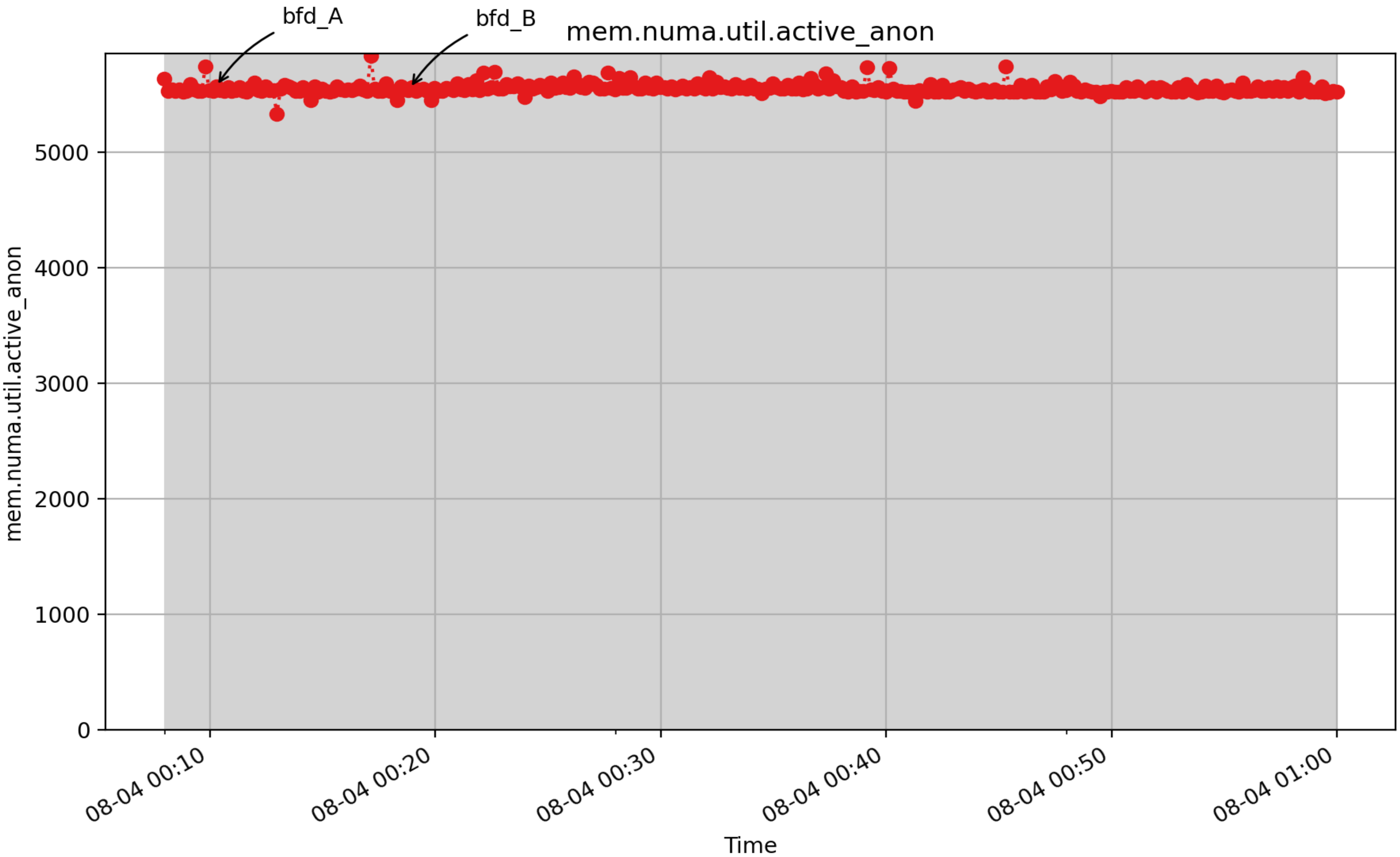
mem.numa.alloc.interleave_hit: count of times interleaving wanted to allocate on this node and succeeded (count - U64) - *rate converted*



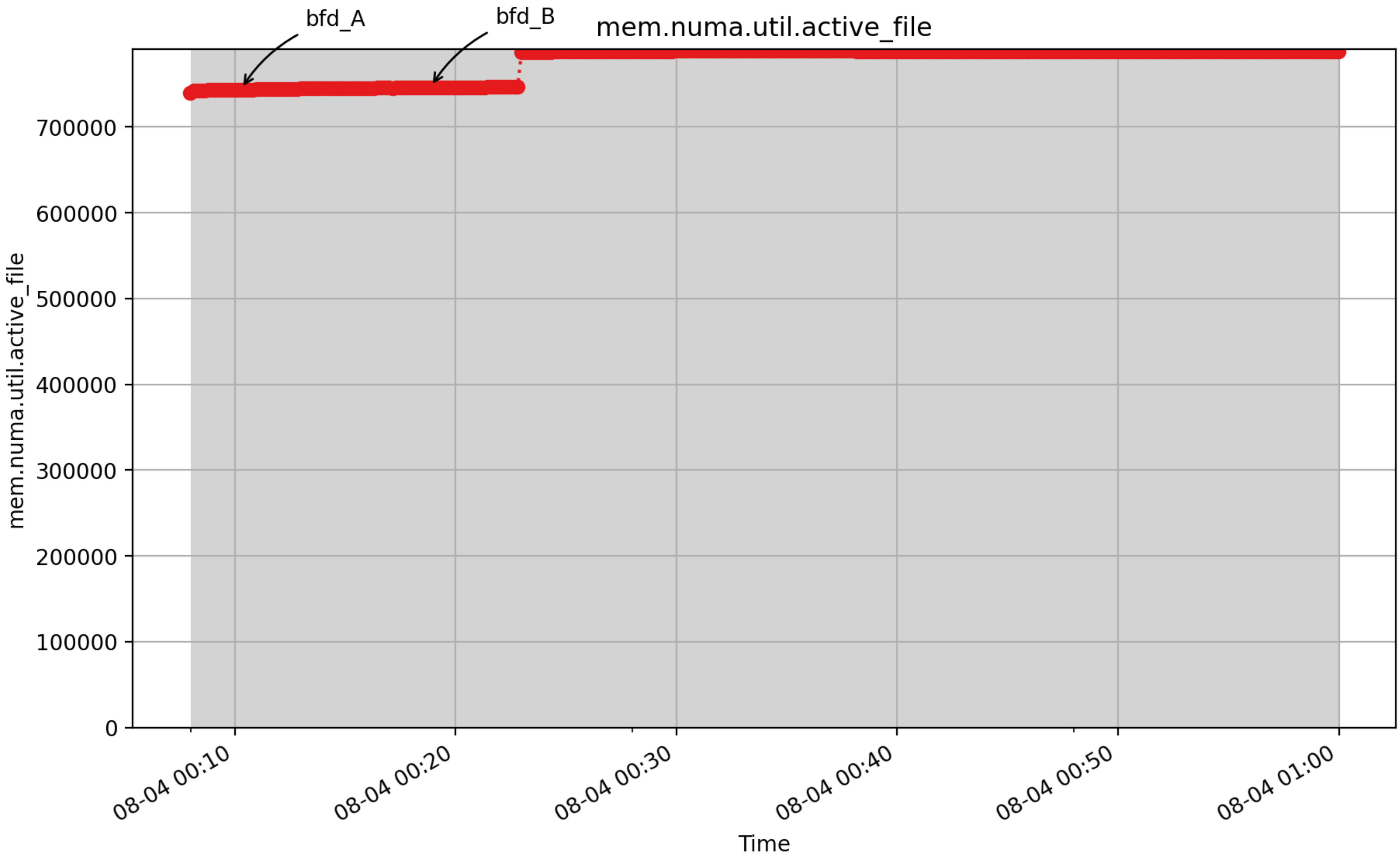
mem.numa.alloc.local_node: count of times a process ran on this node and got memory on this node (count - U64) - *rate converted*



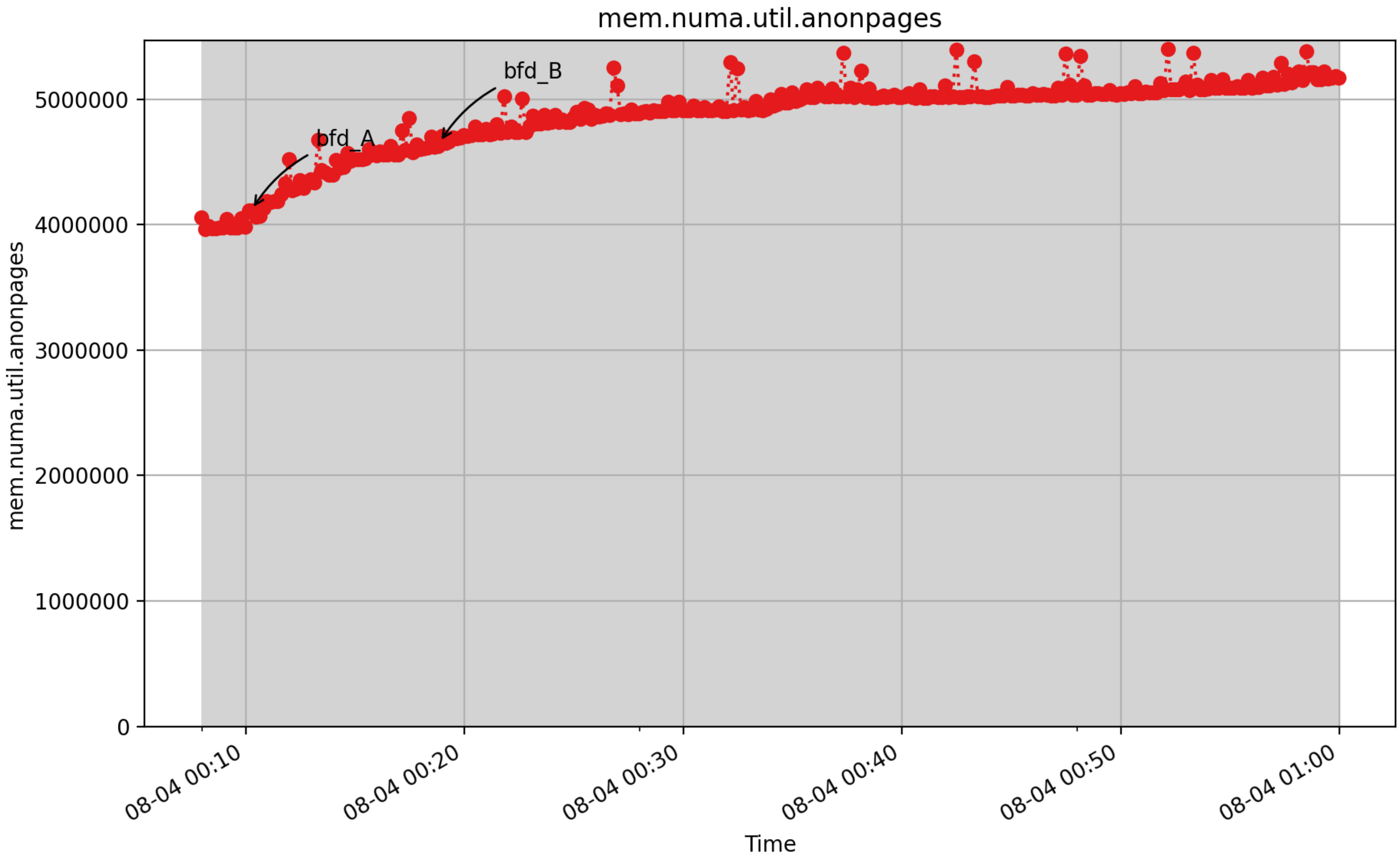
mem.numa.util.active: per-node Active list LRU memory (Kbyte - U64)



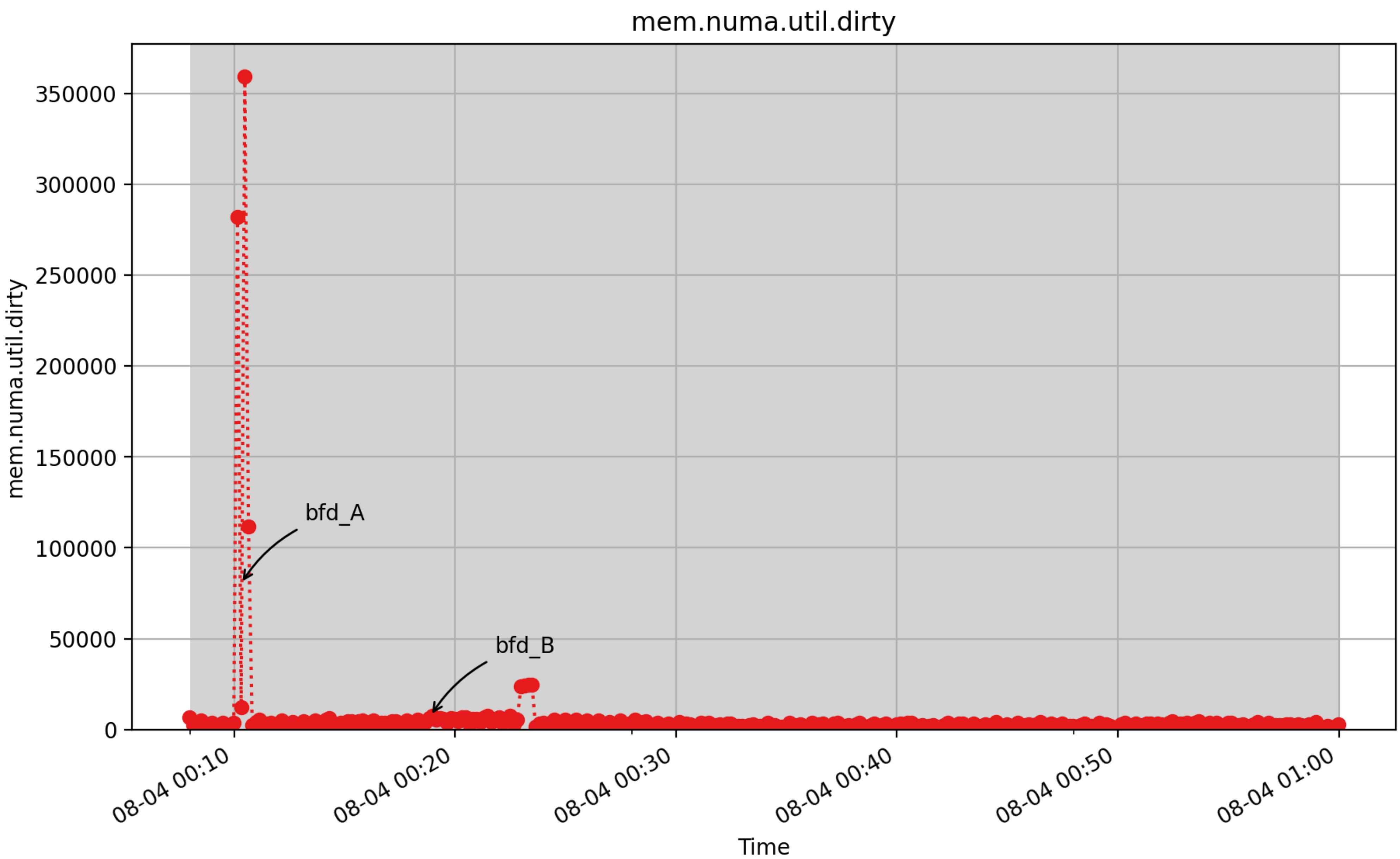
mem.numa.util.active_anon: per-node anonymous Active list LRU memory (Kbyte - U64)



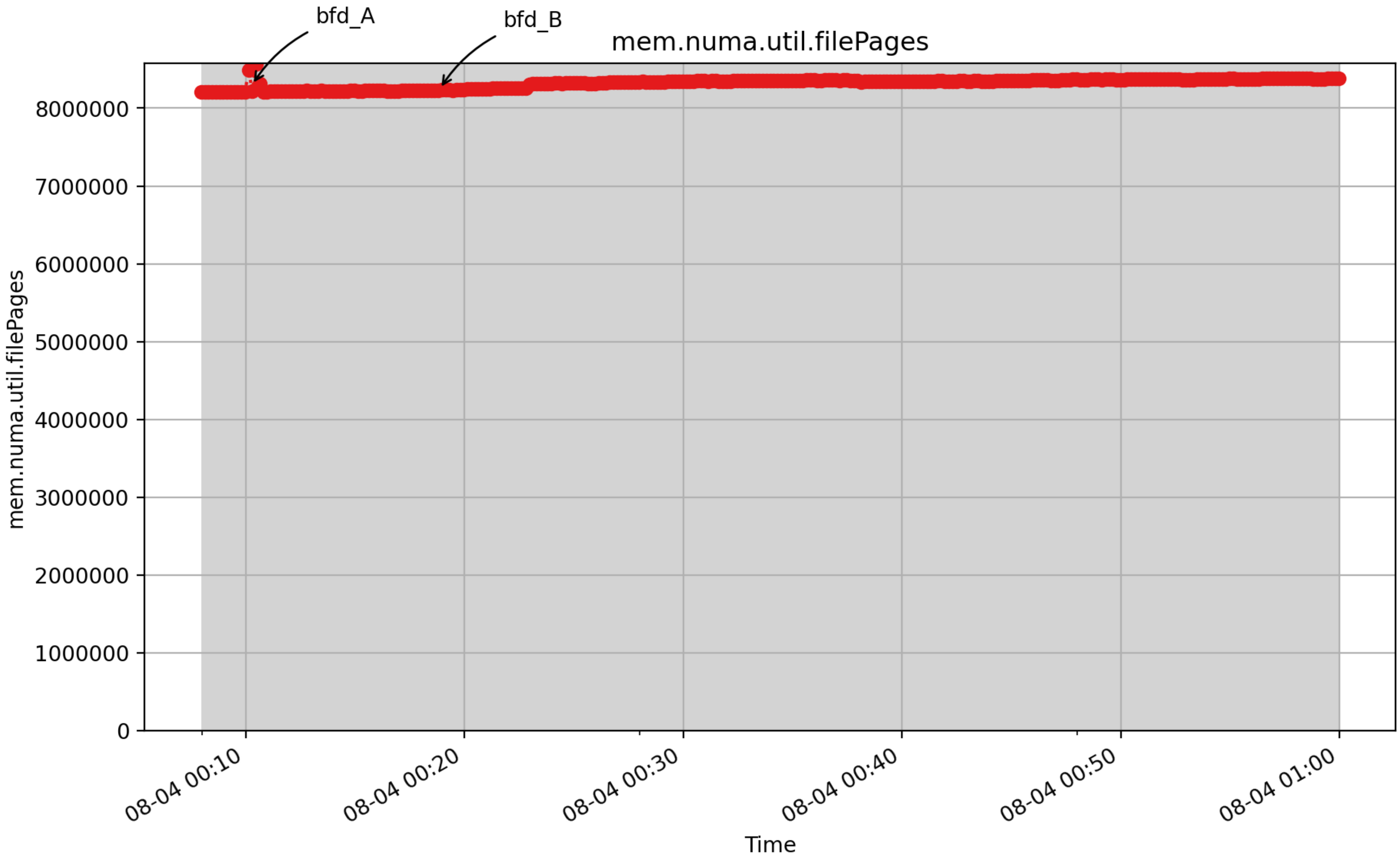
mem.numa.util.active_file: per-node file-backed Active list LRU memory (Kbyte - U64)



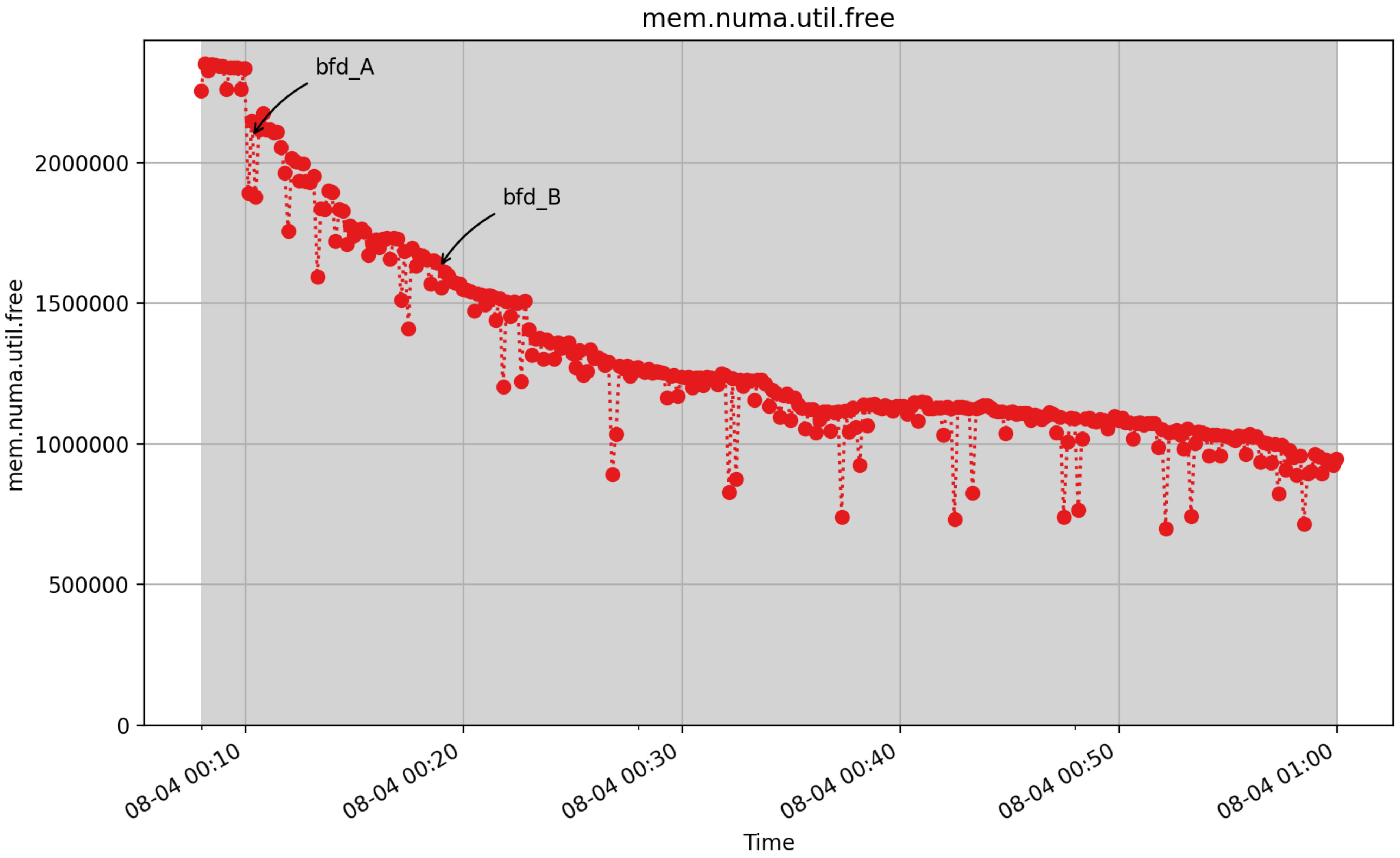
mem.numa.util.anonpages: per-node anonymous memory (Kbyte - U64)



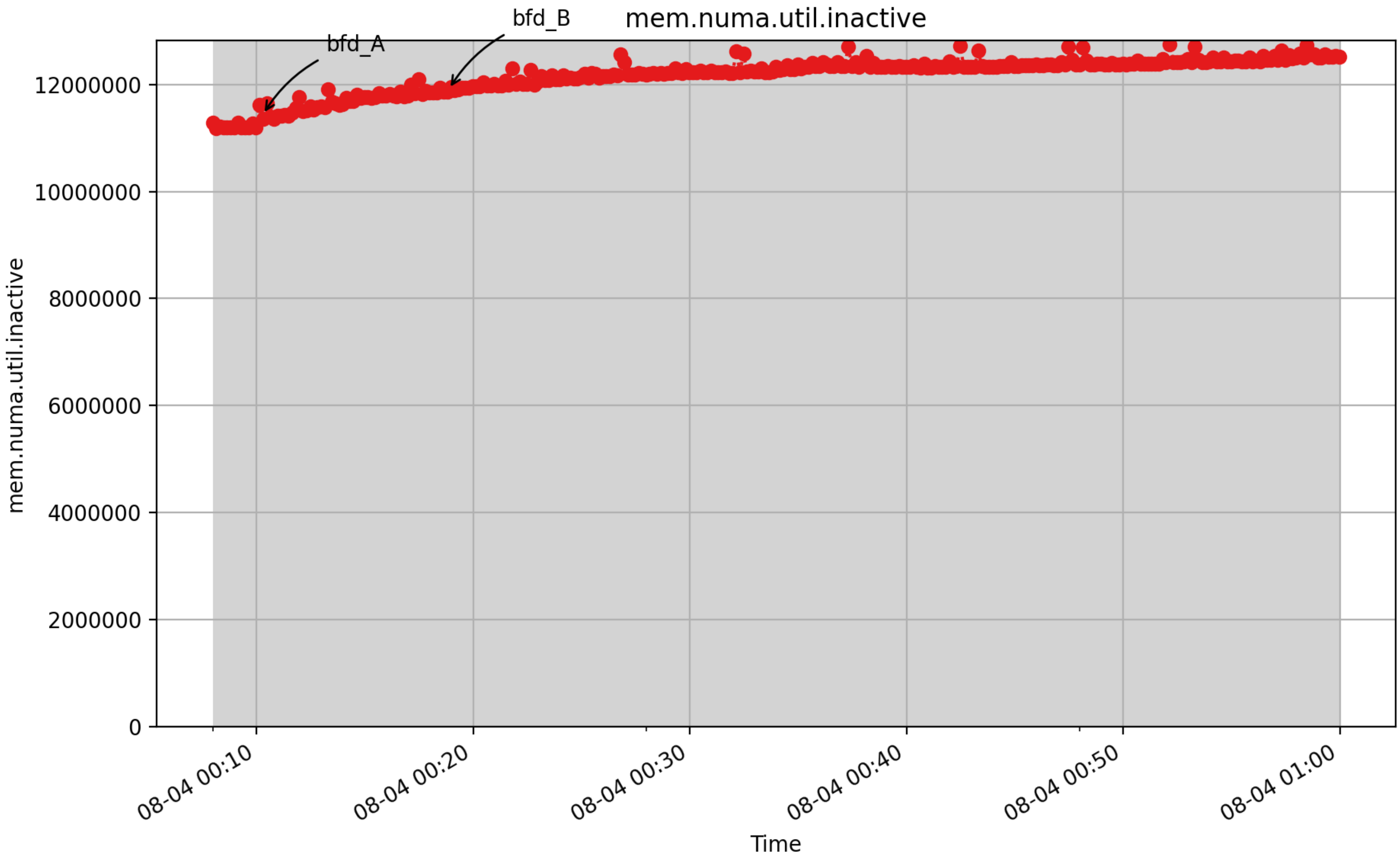
mem.numa.util.dirty: per-node dirty memory (Kbyte - U64)



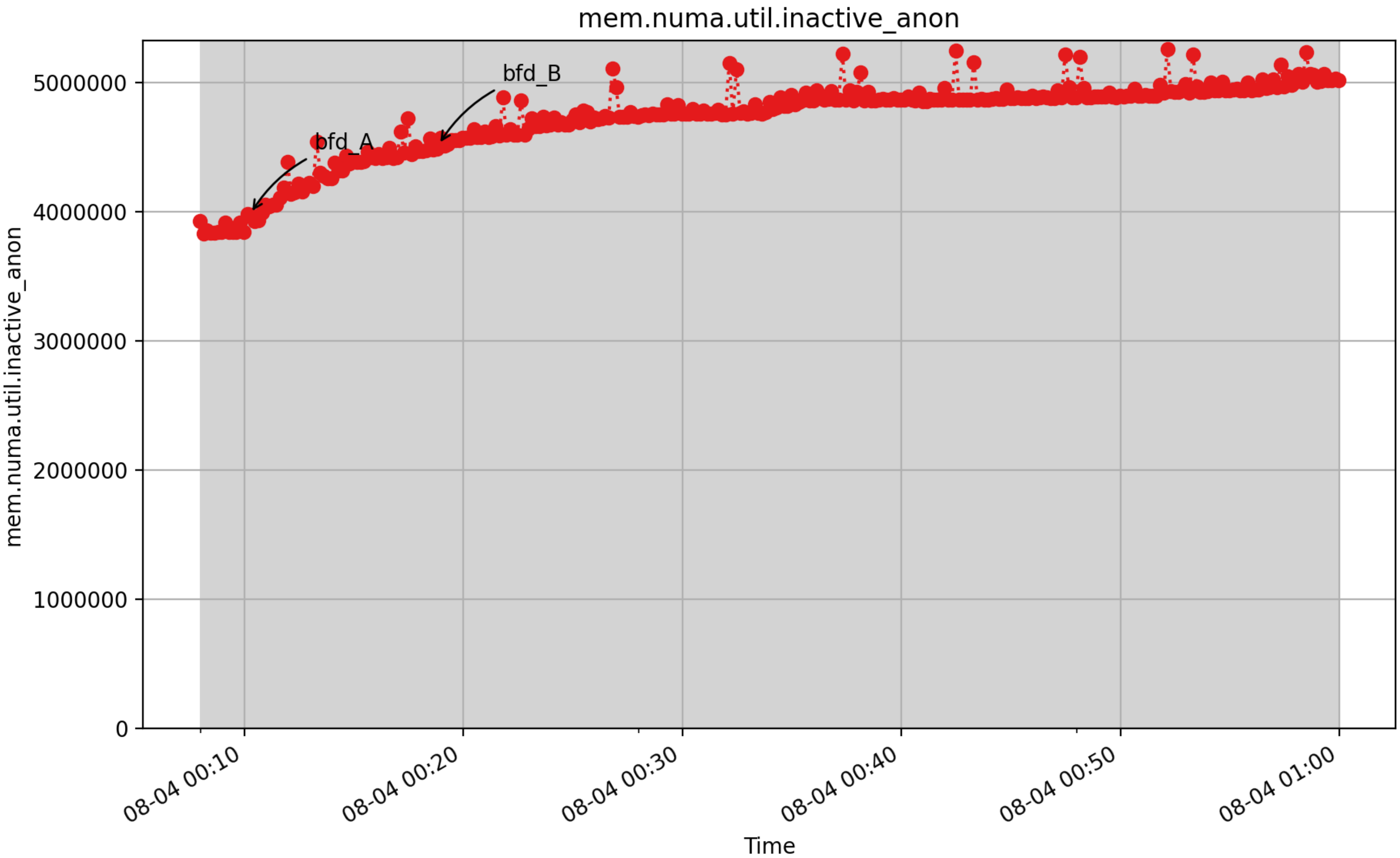
mem.numa.util.filePages: per-node count of memory backed by files (Kbyte - U64)



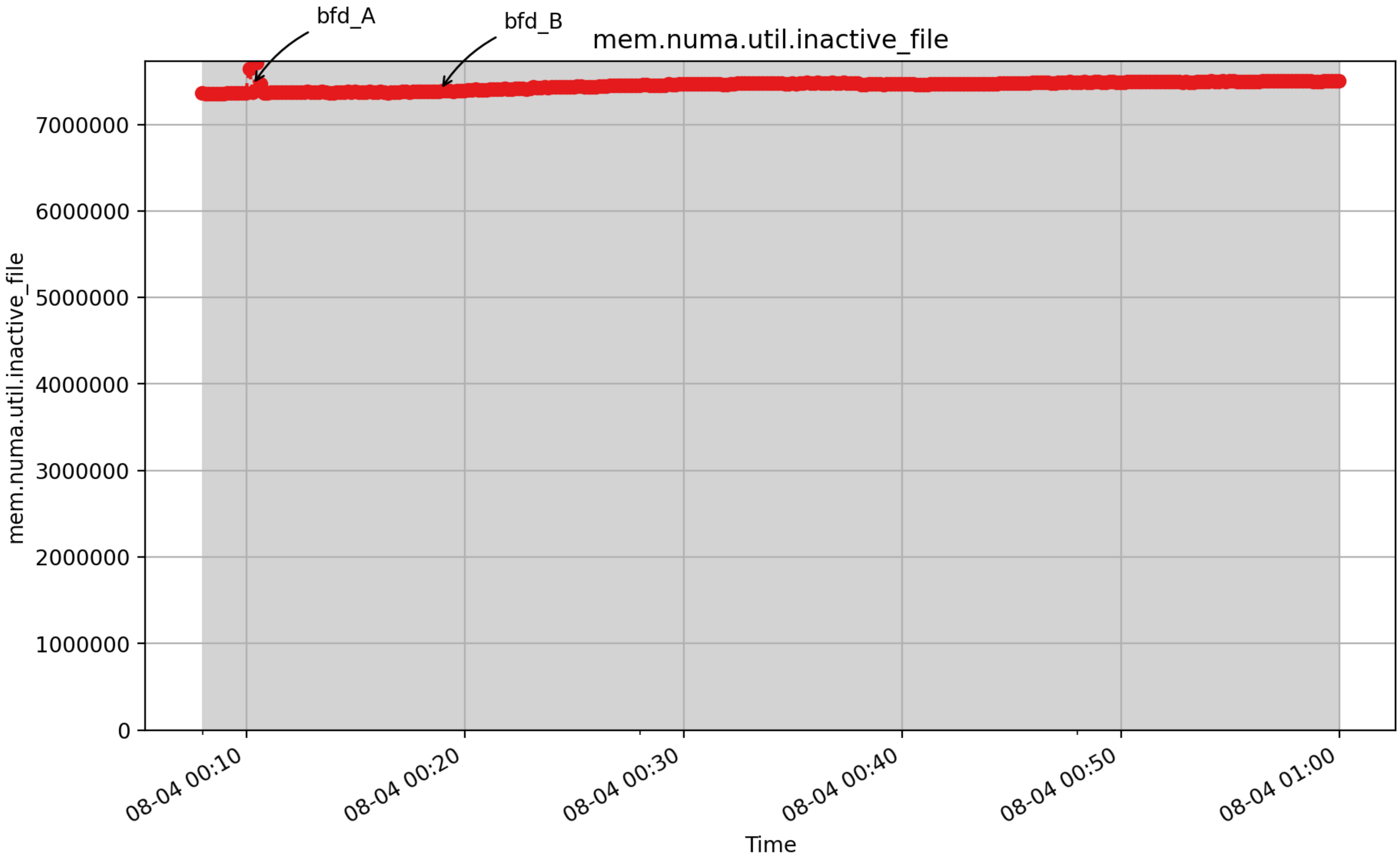
mem.numa.util.free: per-node free memory (Kbyte - U64)



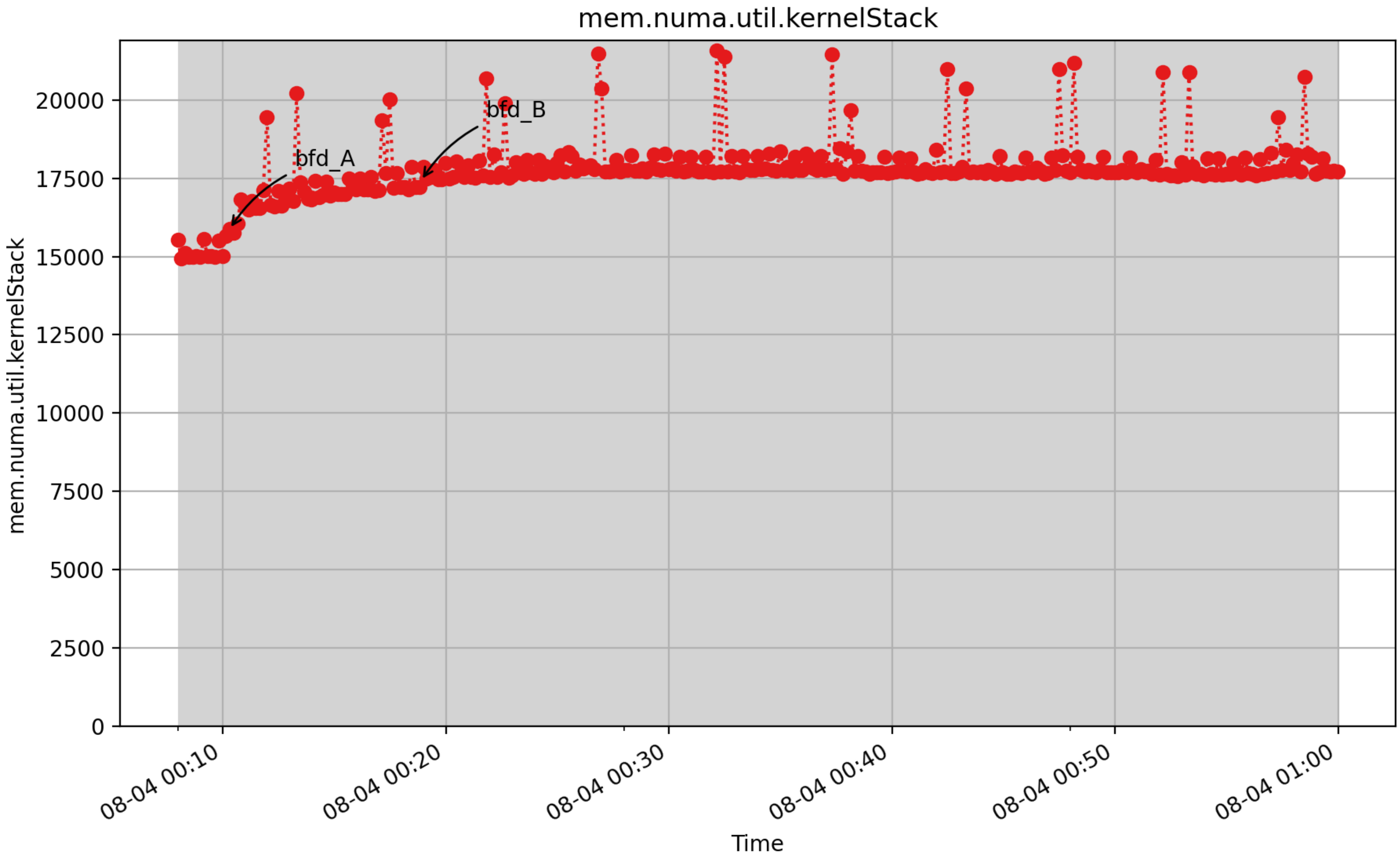
mem.numa.util.inactive: per-node Inactive list LRU memory (Kbyte - U64)



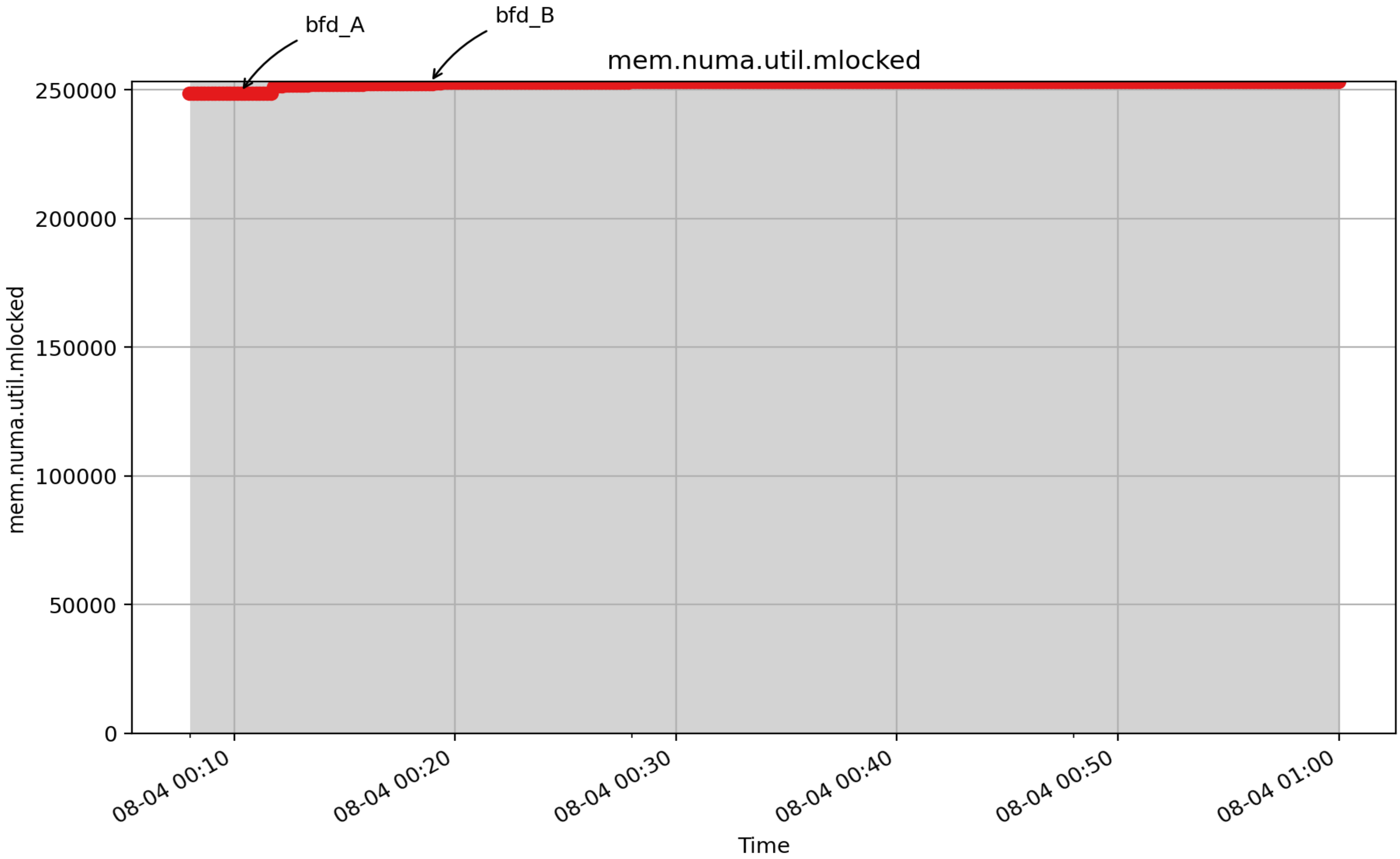
mem.numa.util.inactive_anon: per-node anonymous Inactive list LRU memory (Kbyte - U64)



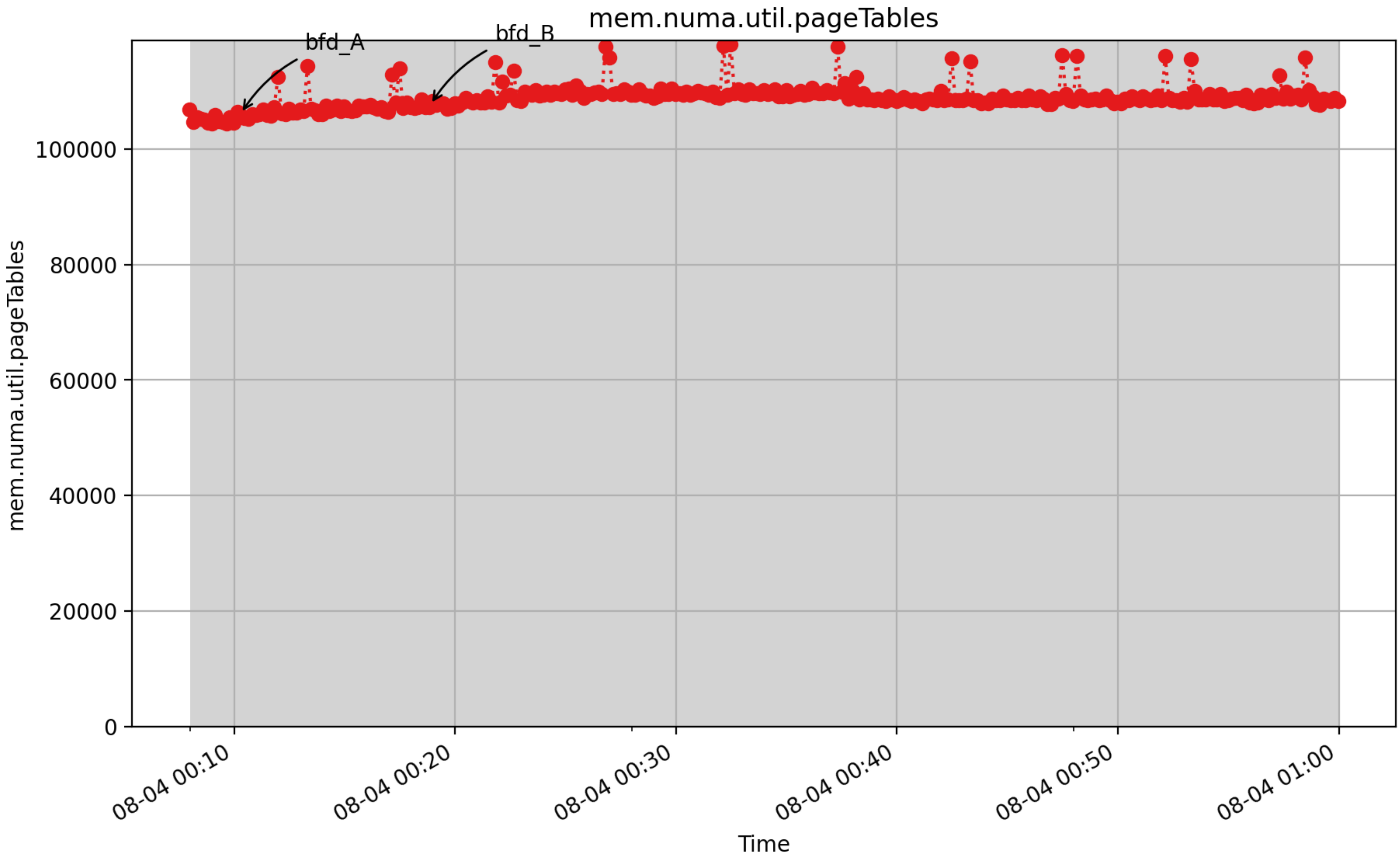
mem.numa.util.inactive_file: per-node file-backed Inactive list LRU memory (Kbyte - U64)



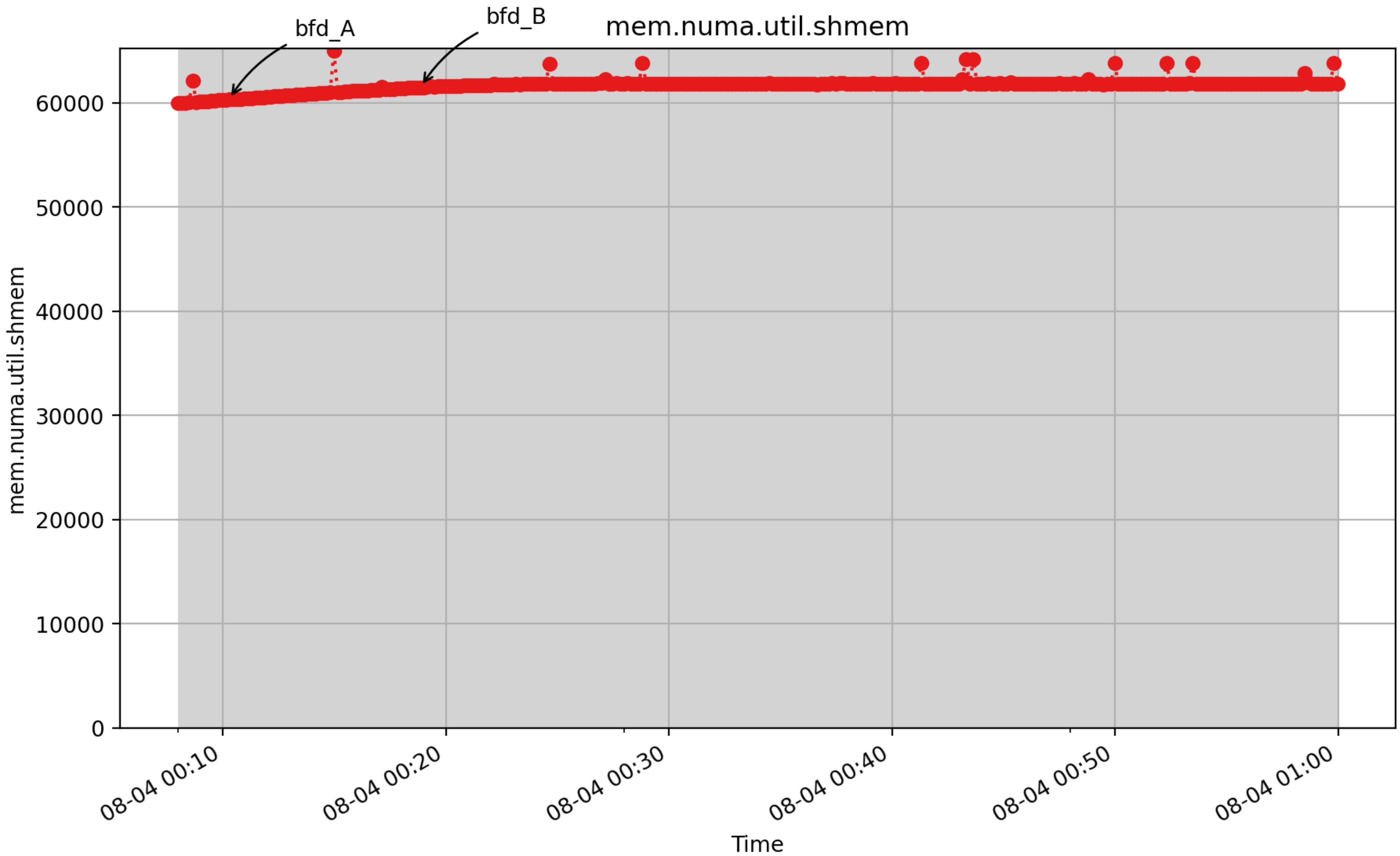
mem.numa.util.kernelStack: per-node memory used as kernel stacks (Kbyte - U64)



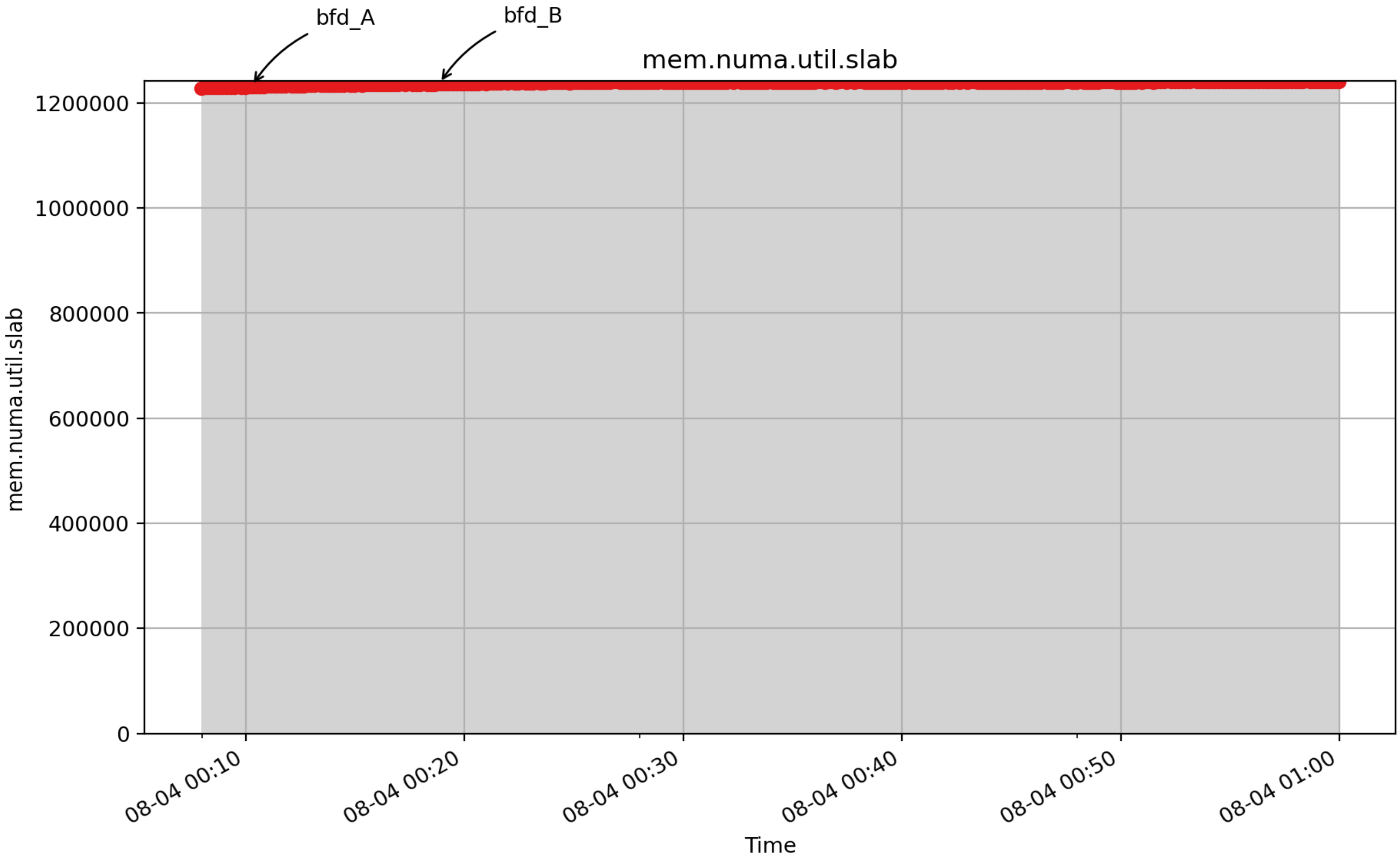
mem.numa.util.mlocked: per-node count of Mlocked memory (Kbyte - U64)



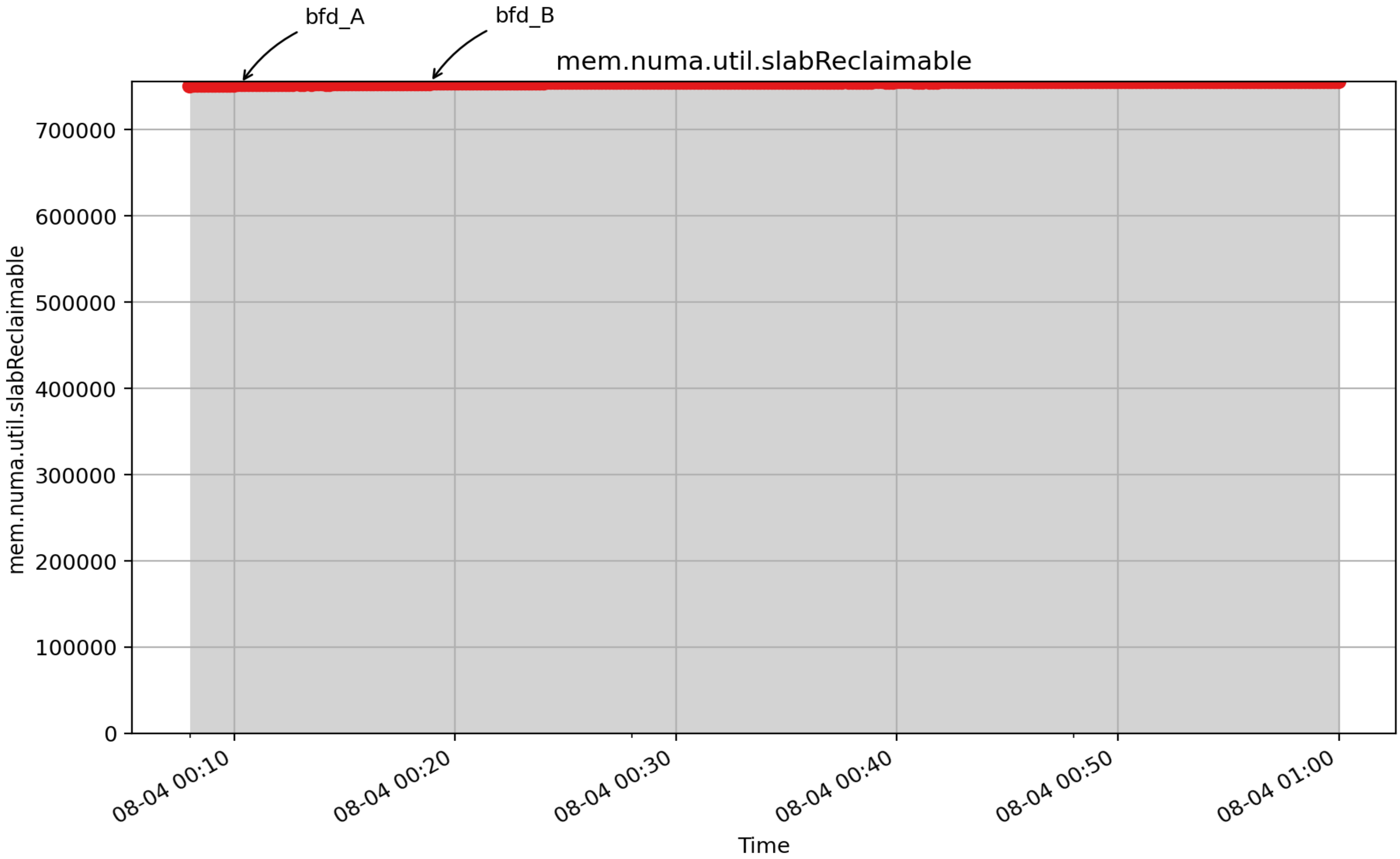
mem.numa.util.pageTables: per-node memory used for pagetables (Kbyte - U64)



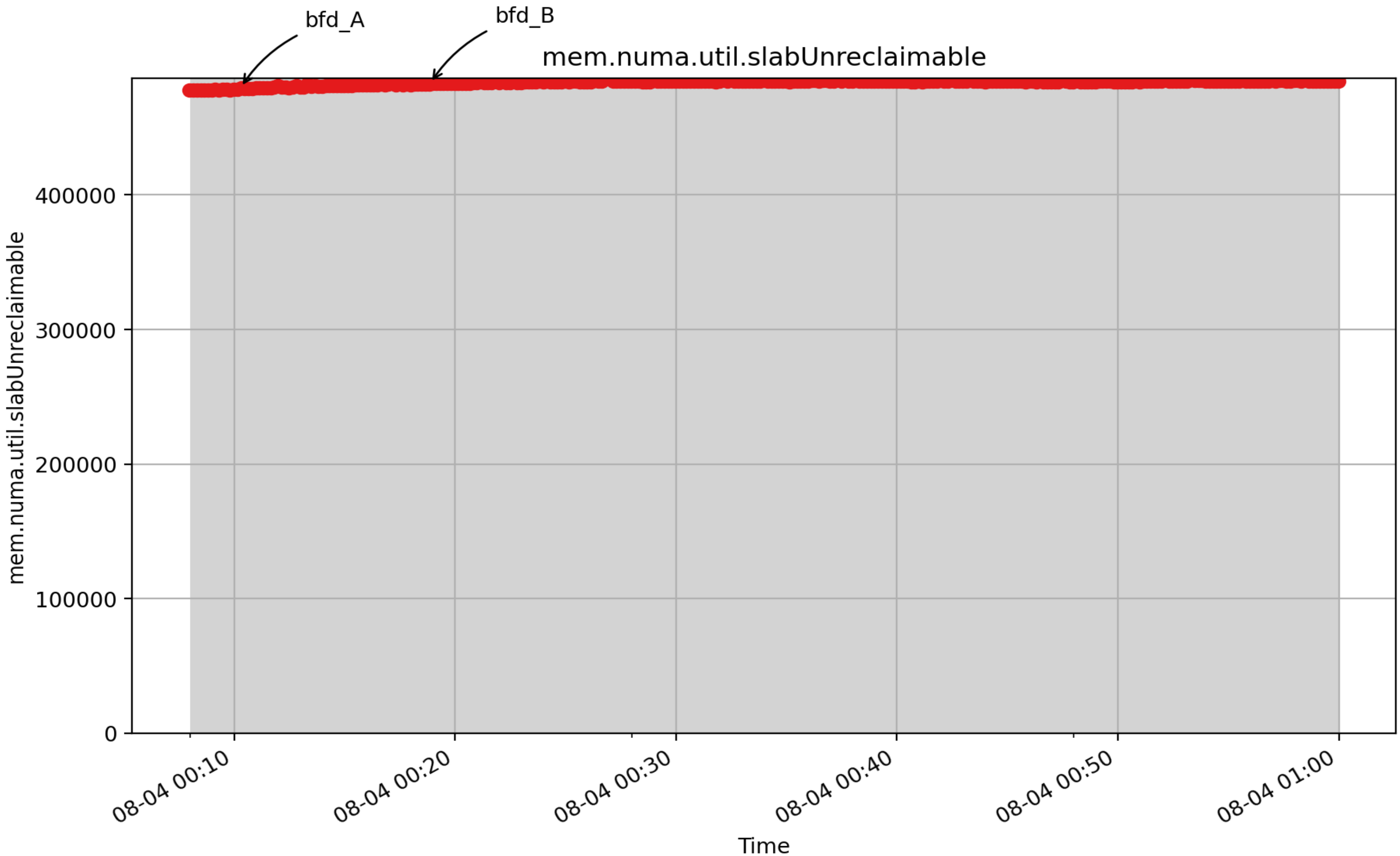
mem.numa.util.shmem: per-node amount of shared memory (Kbyte - U64)



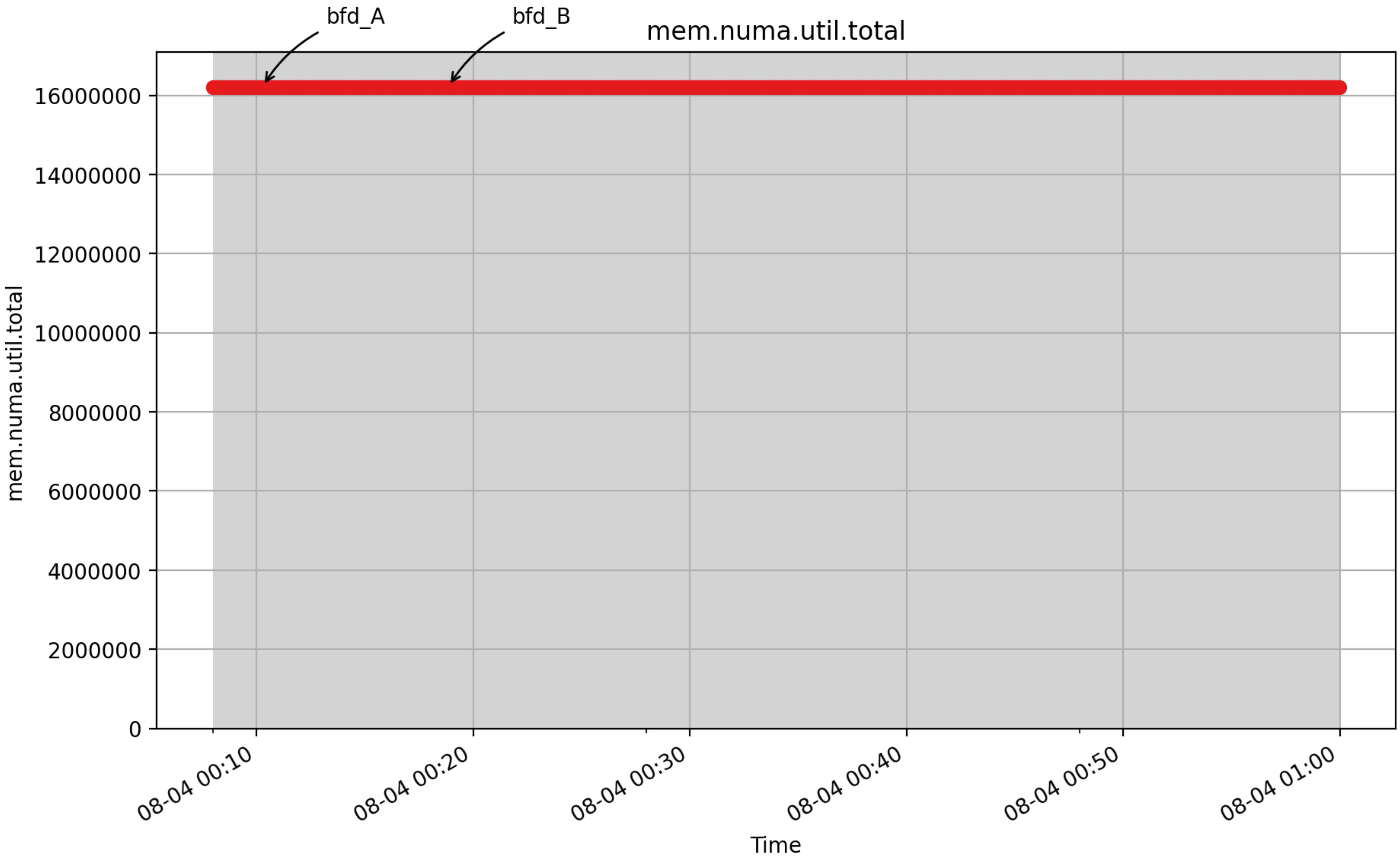
mem.numa.util.slab: per-node memory used for slab objects (Kbyte - U64)



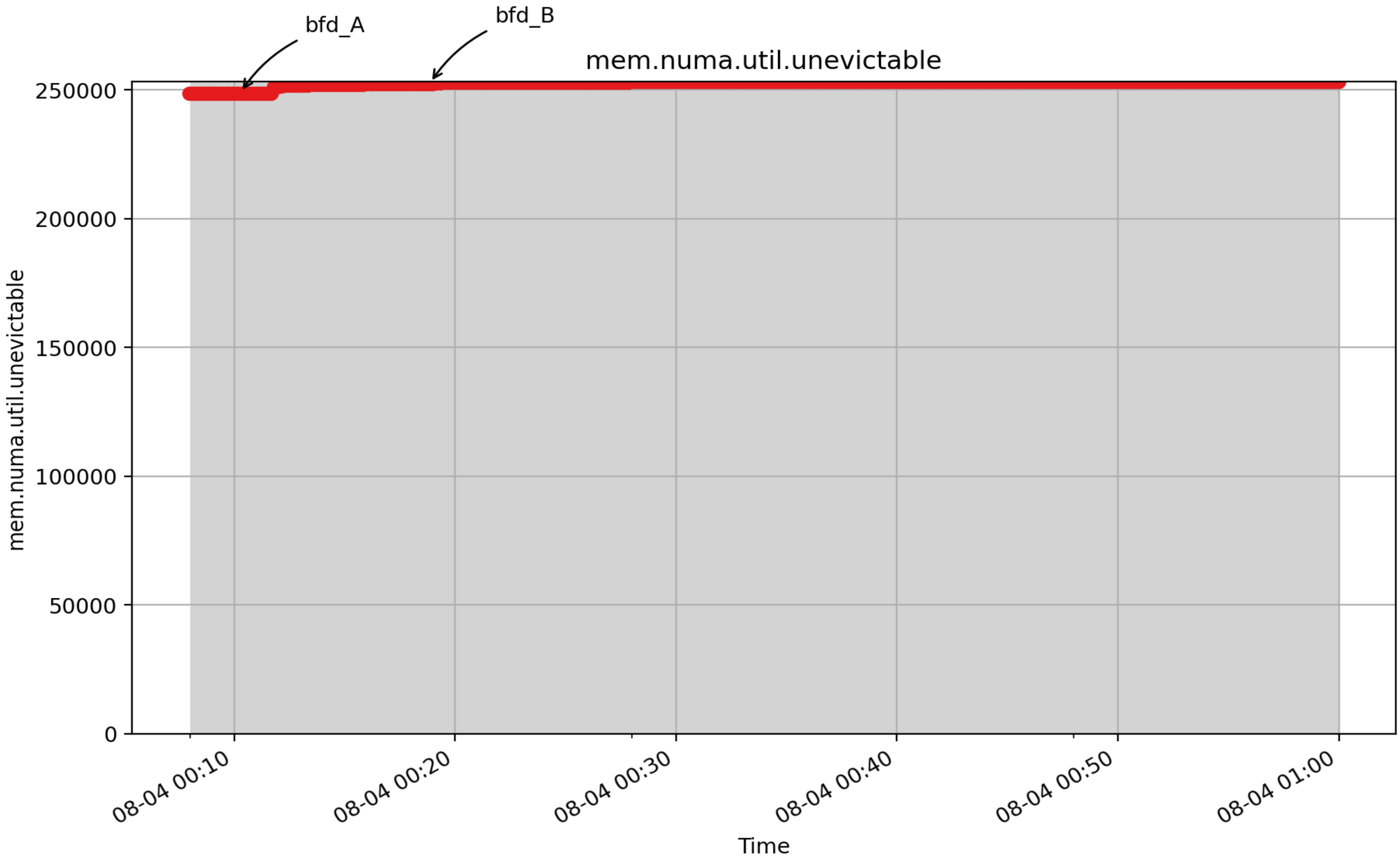
mem.numa.util.slabReclaimable: per-node memory used for slab objects that can be reclaimed (Kbyte - U64)



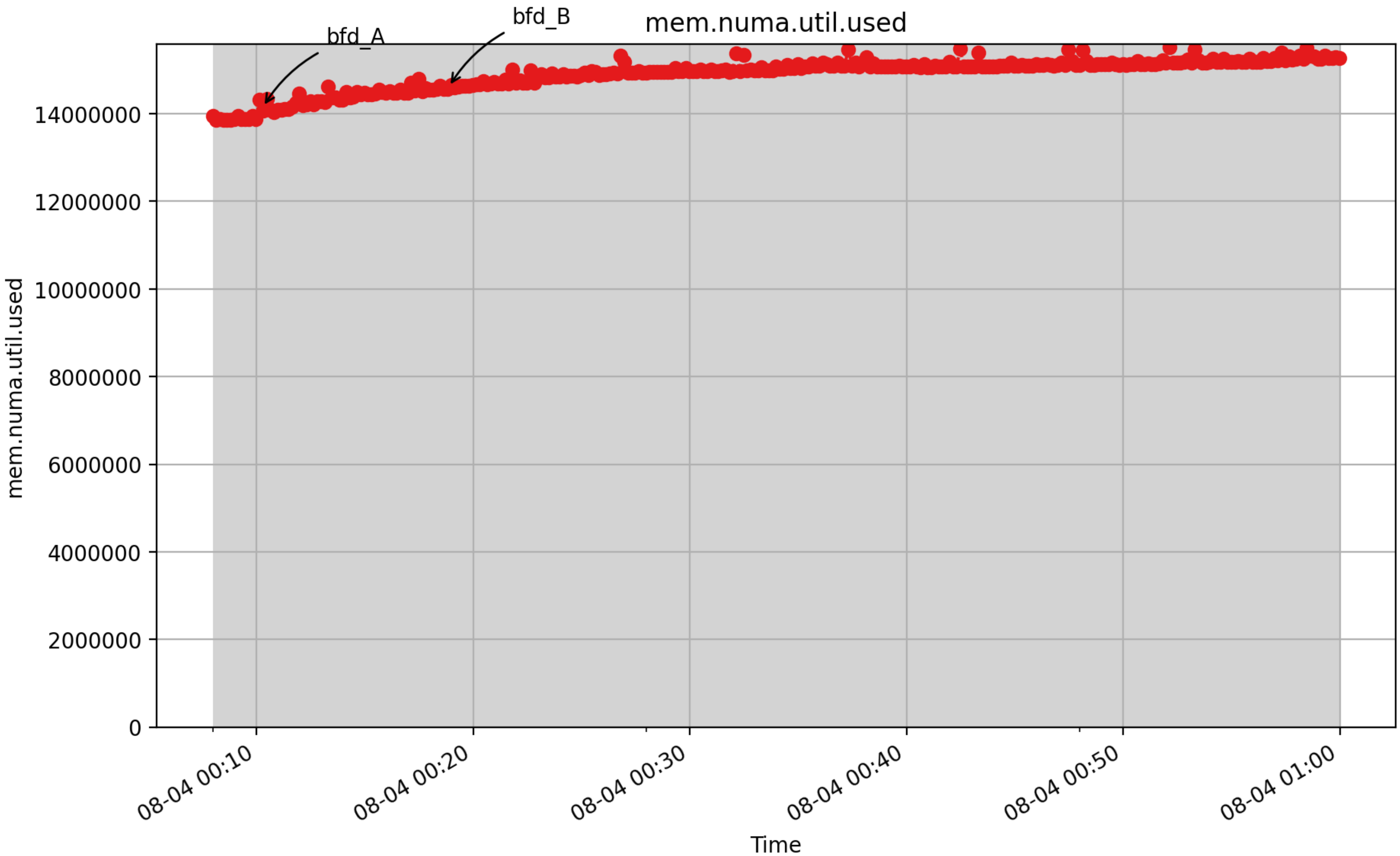
mem.numa.util.slabUnreclaimable: per-node memory used for slab objects that is unreclaimable (Kbyte - U64)



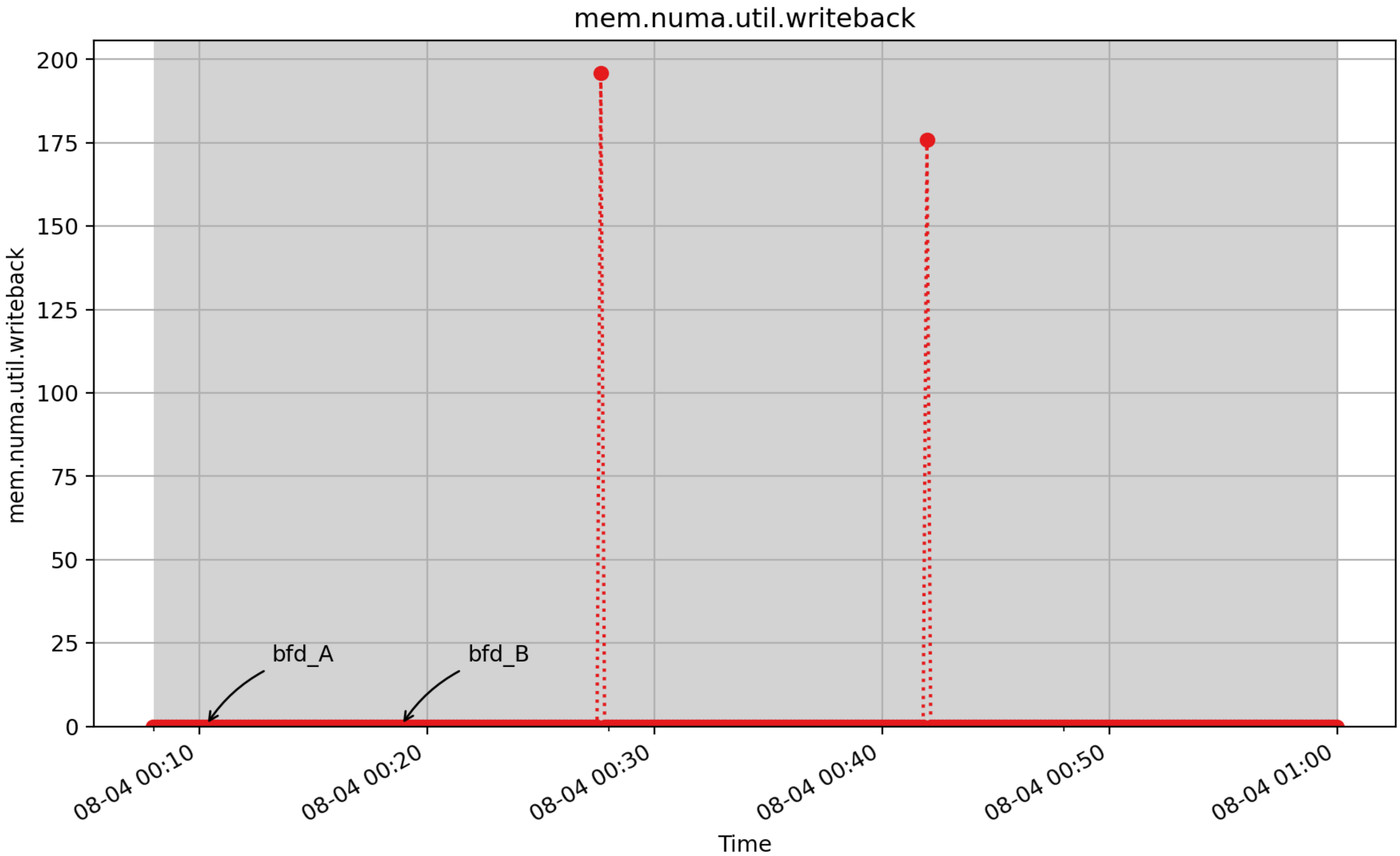
mem.numa.util.total: per-node total memory (Kbyte - U64)



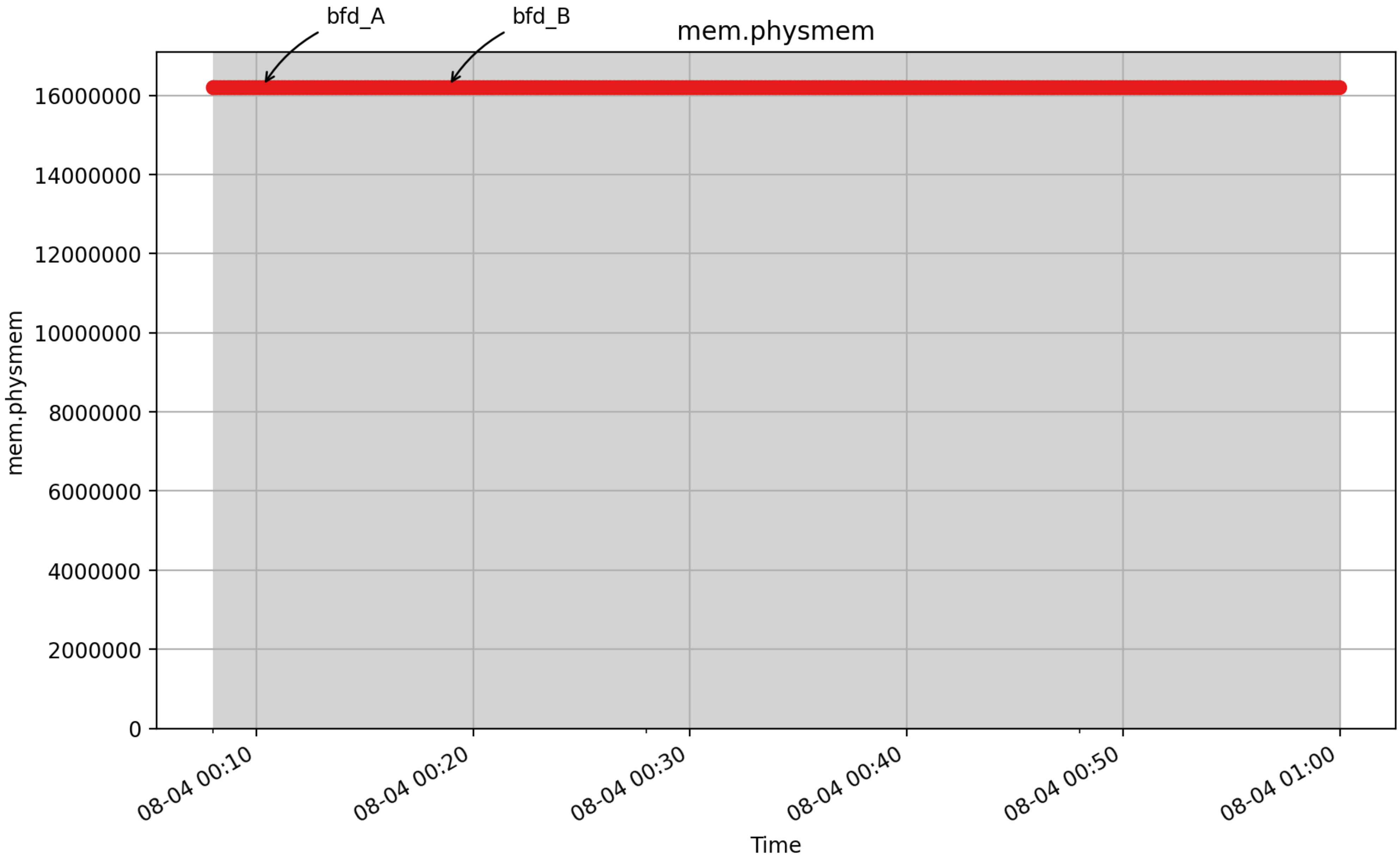
mem.numa.util.unevictable: per-node Unevictable memory (Kbyte - U64)



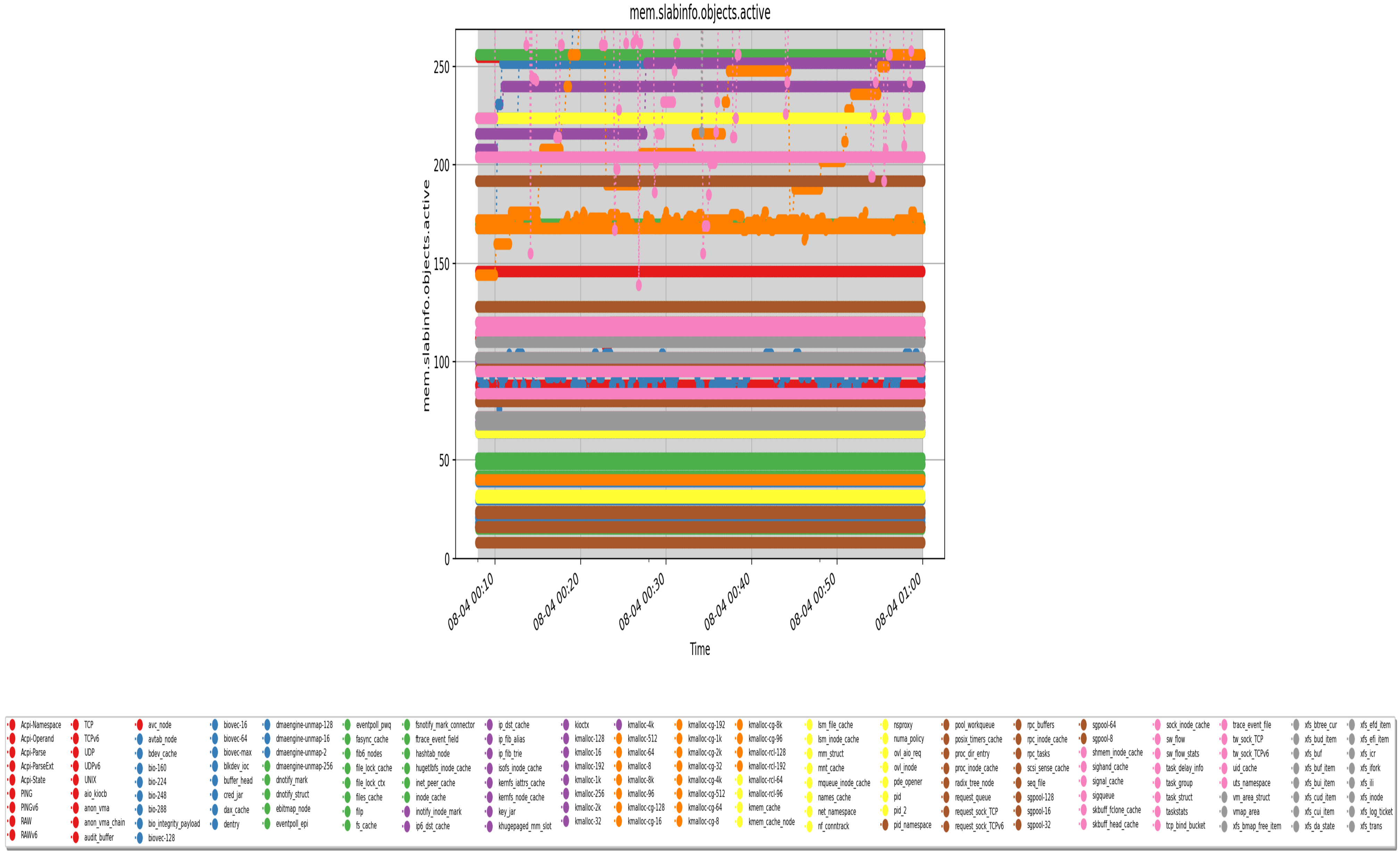
mem.numa.util.used: per-node used memory (Kbyte - U64)



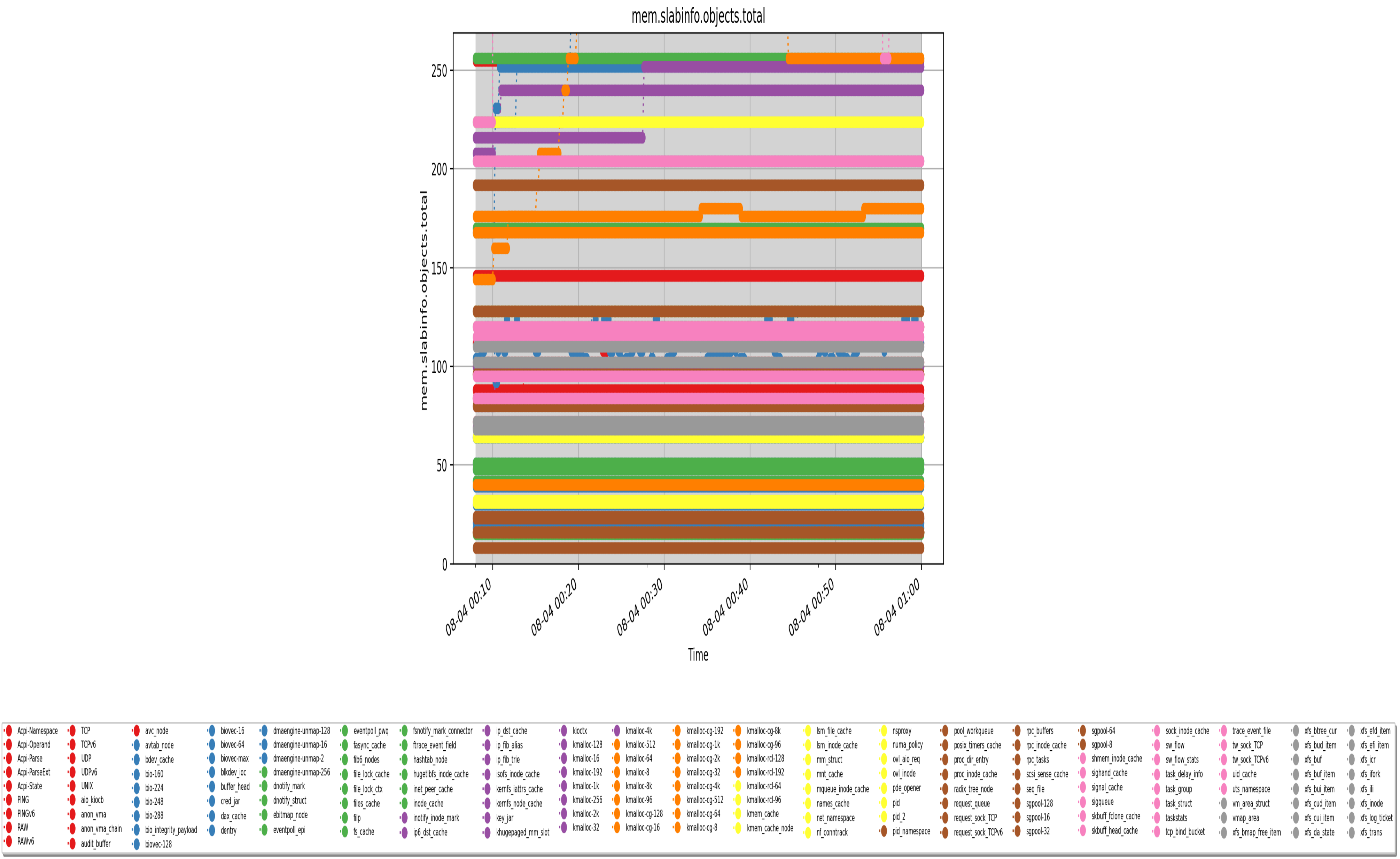
mem.numa.util.writeback: per-node count of memory locked for writeback to stable storage (Kbyte - U64)



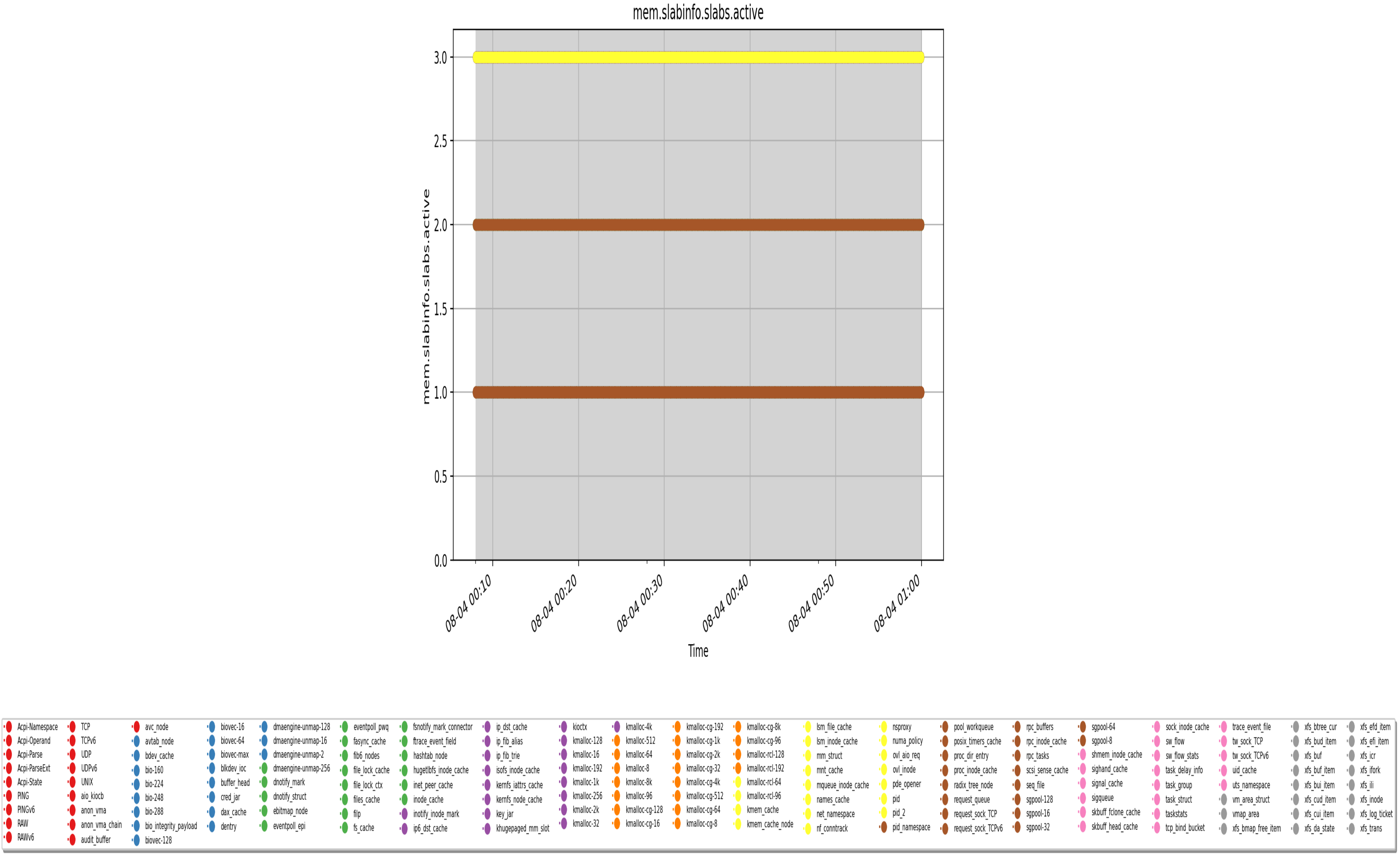
mem.physmem: The value of this metric corresponds to the "MemTotal" field reported by /proc/meminfo. Note that this does not necessarily correspond to actual installed physical memory - there may be areas of the physical address space mapped as ROM in various peripheral devices and the bios may be mirroring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



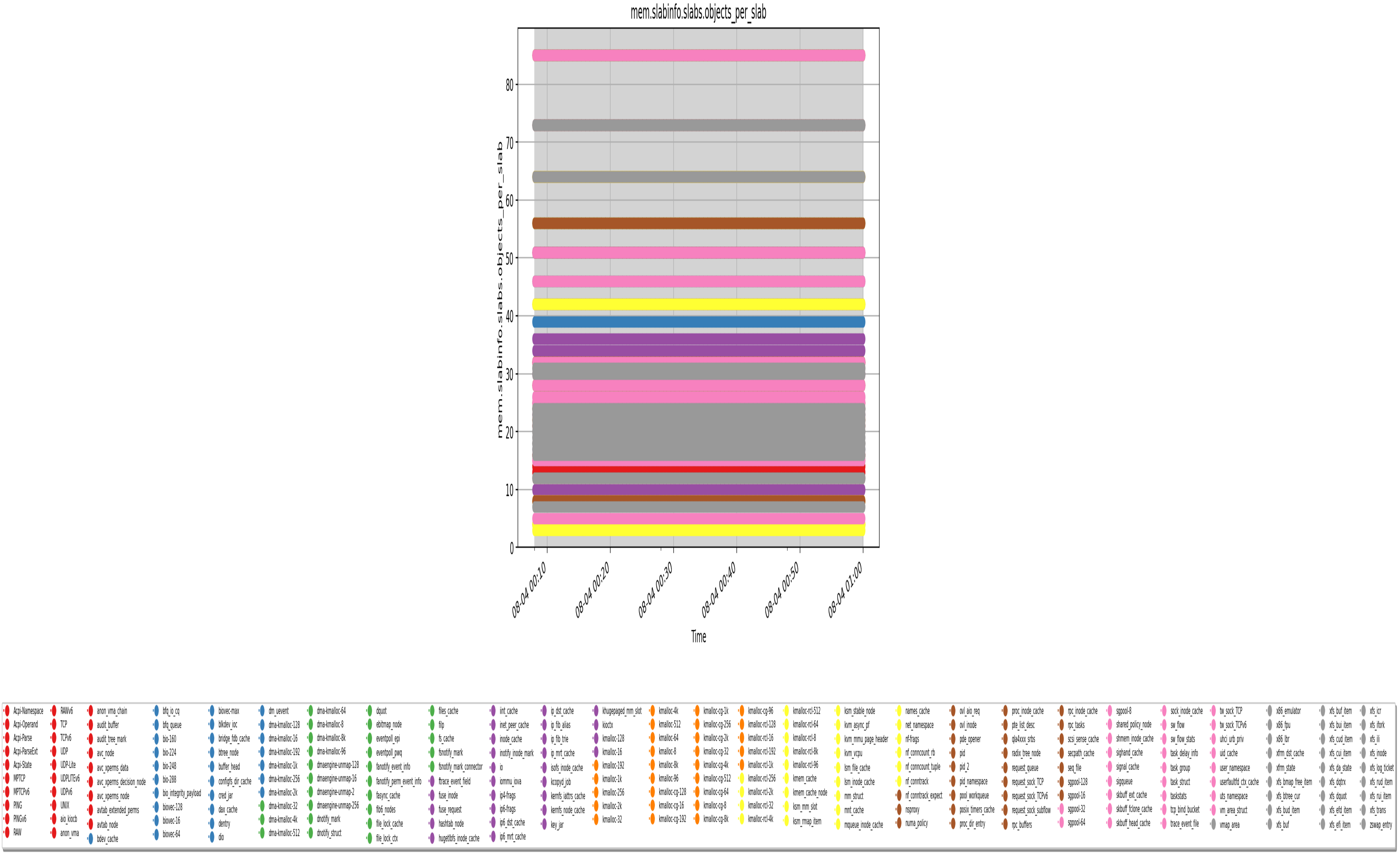
mem.slabinfo.objects.active: number of active objects in each cache (- U64)



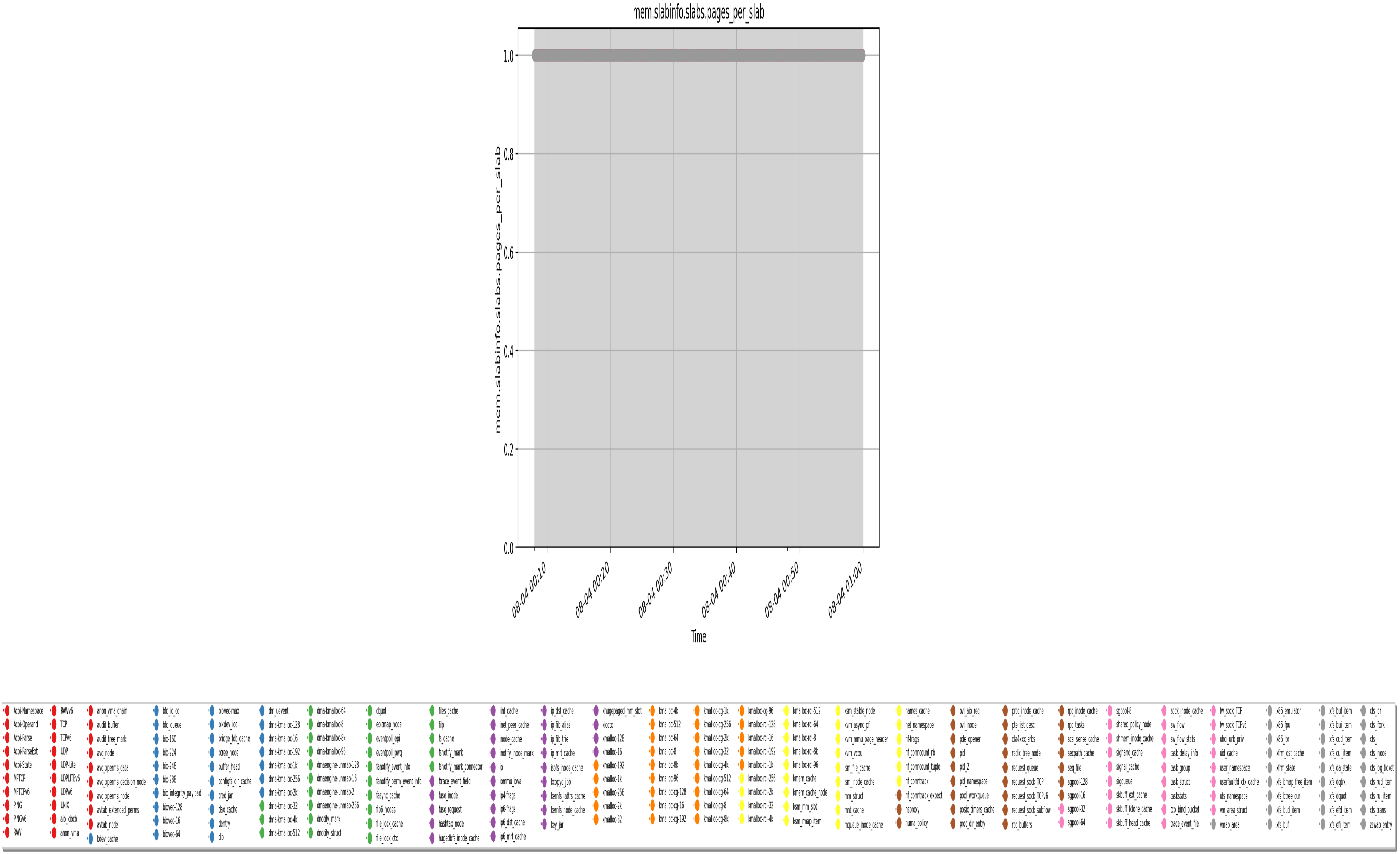
mem.slabinfo.objects.total: total number of objects in each cache (- U64)



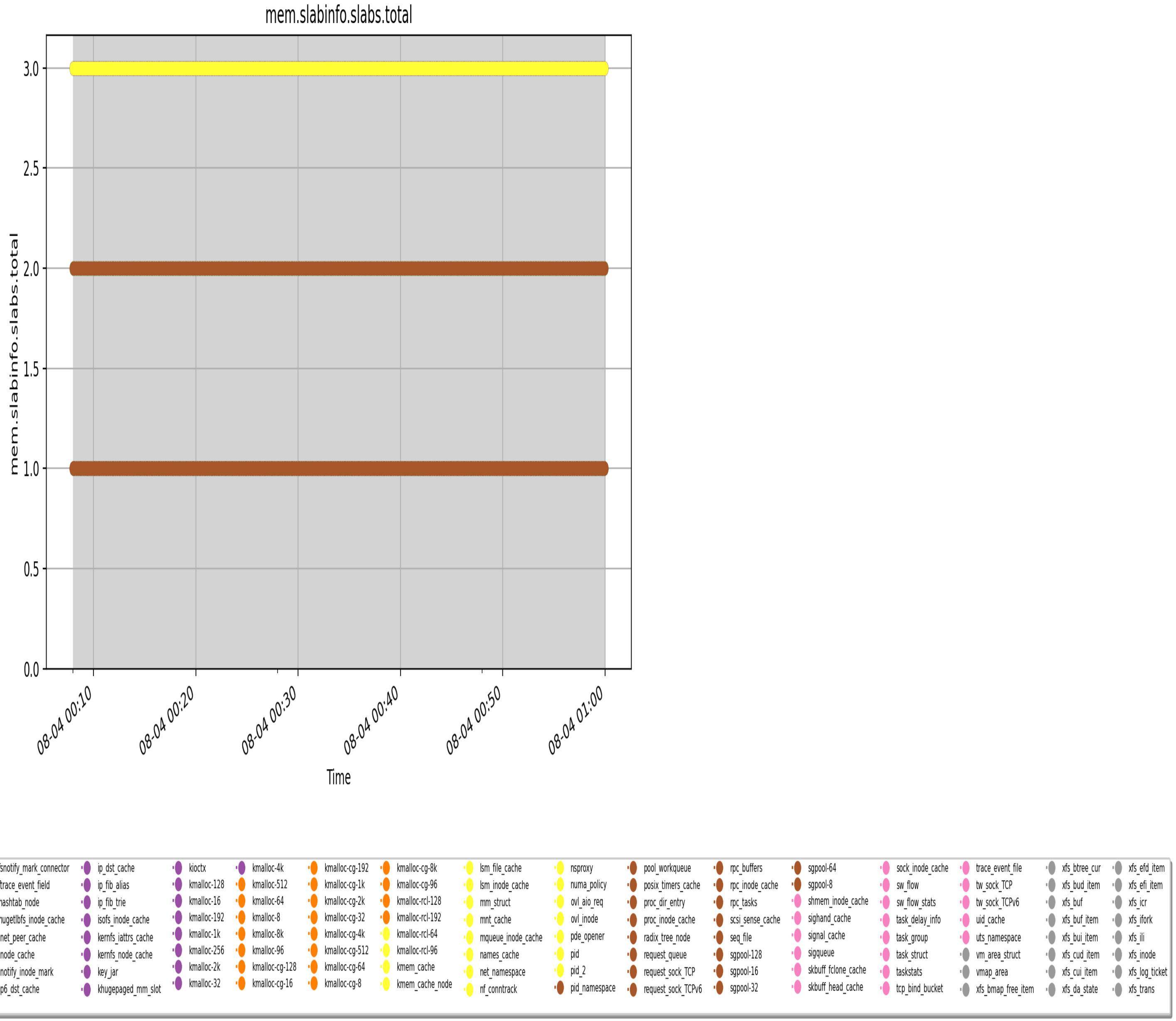
mem.slabinfo.slabs.active: number of active slabs comprising each cache (- U32)



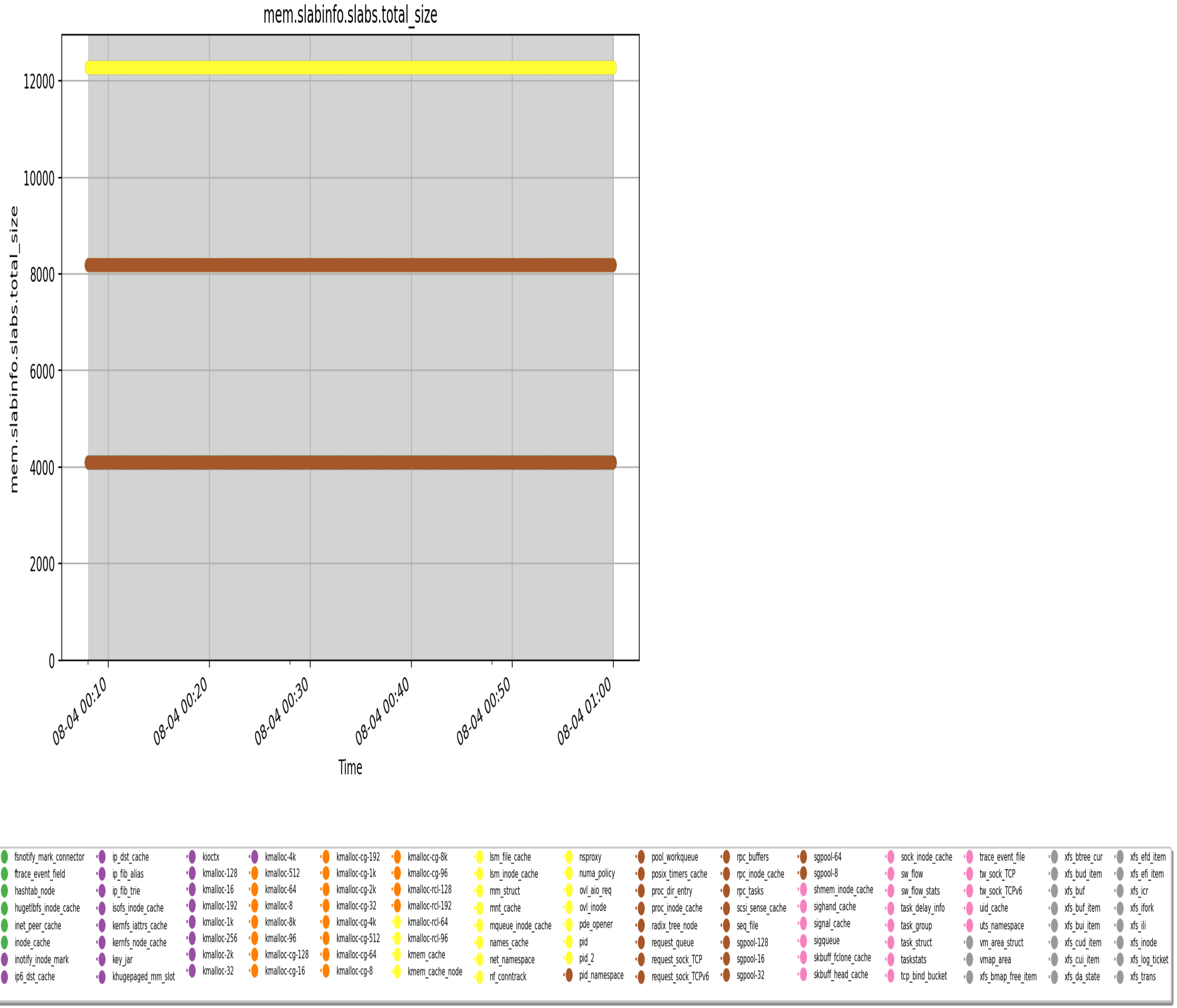
mem.slabinfo.slabs.objects_per_slab: number of objects in each slab (- U32)



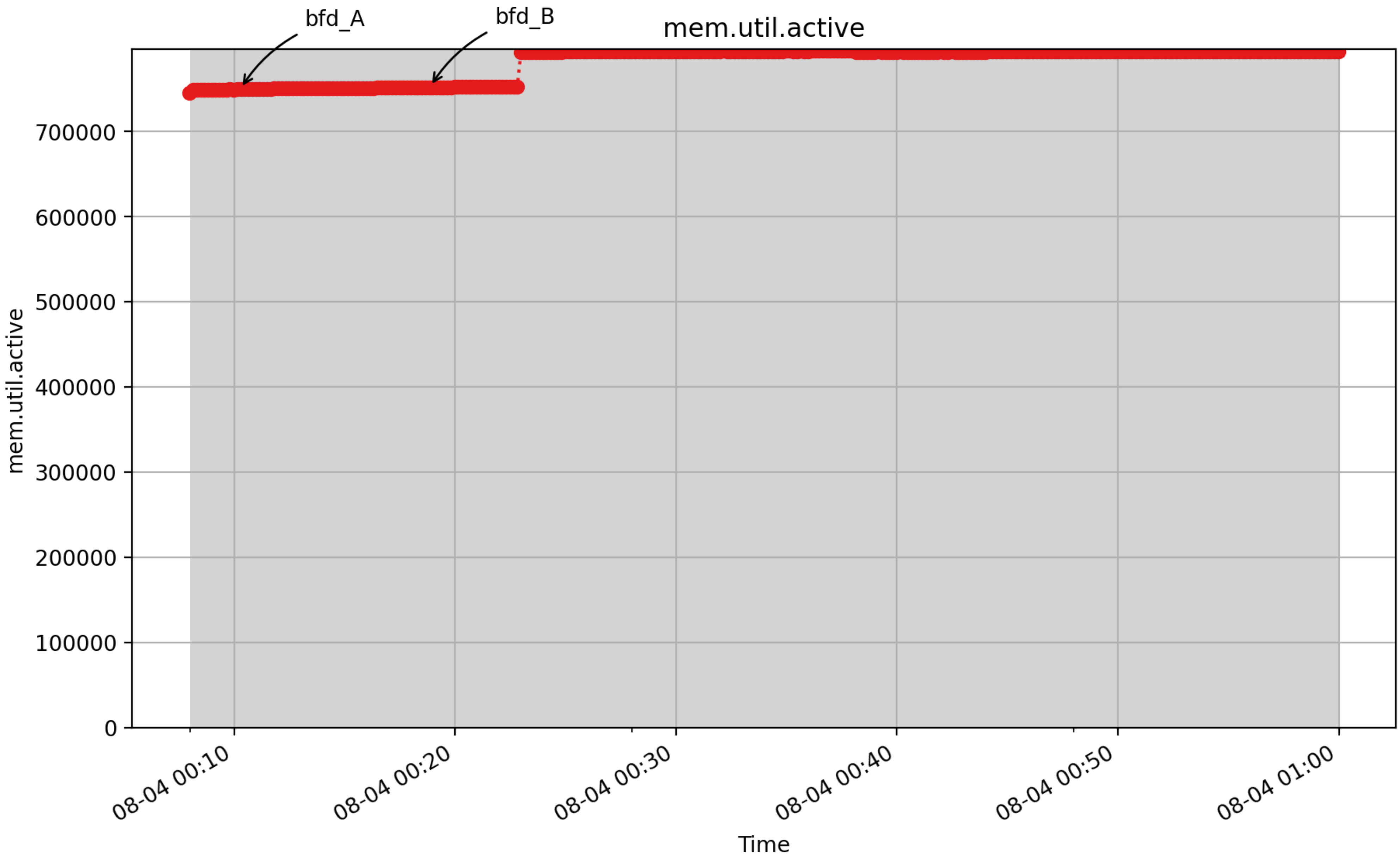
mem.slabinfo.slabs.pages_per_slab: number of pages in each slab (- U32)



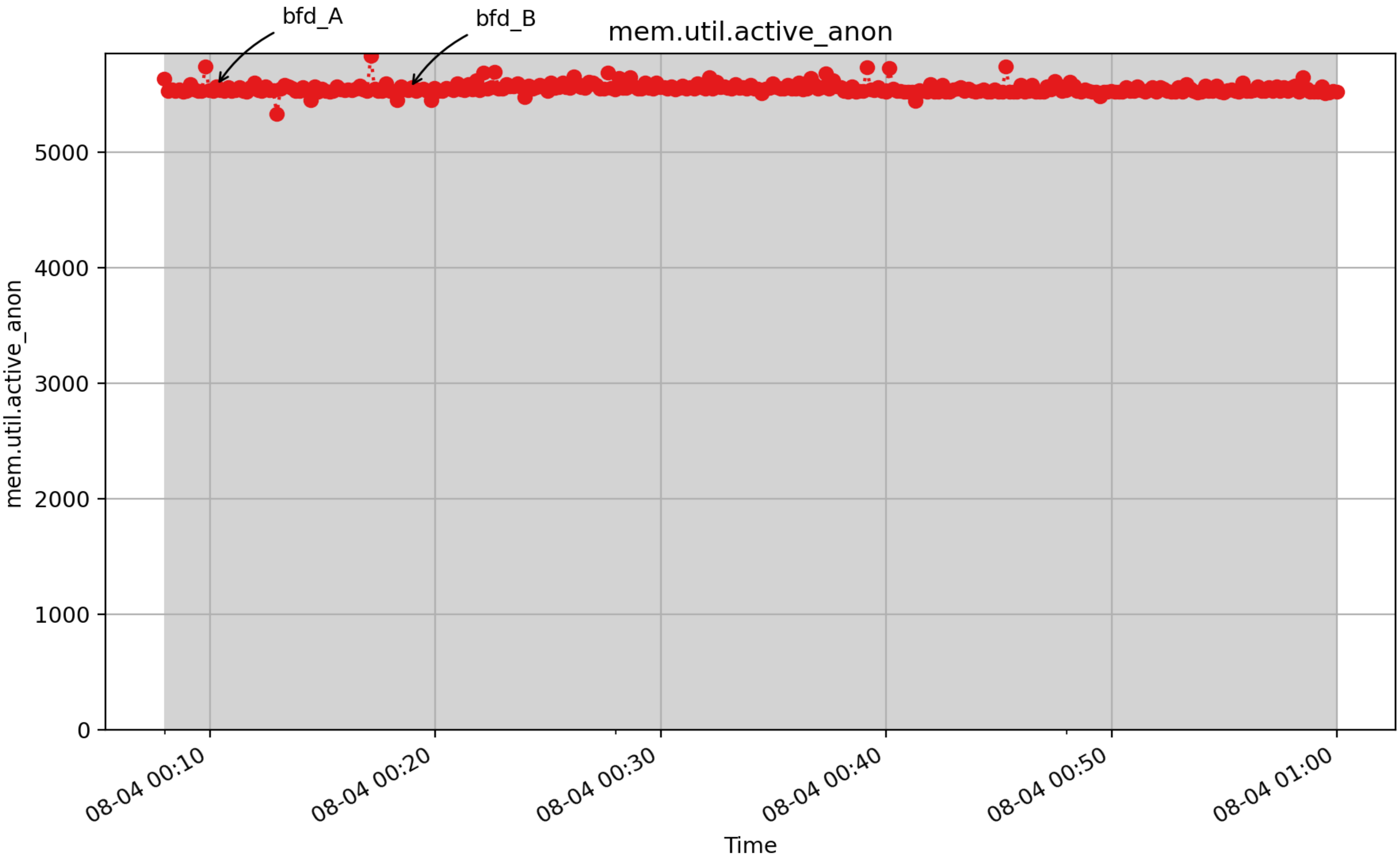
mem.slabinfo.slabs.total: total number of slabs comprising each cache (- U32)



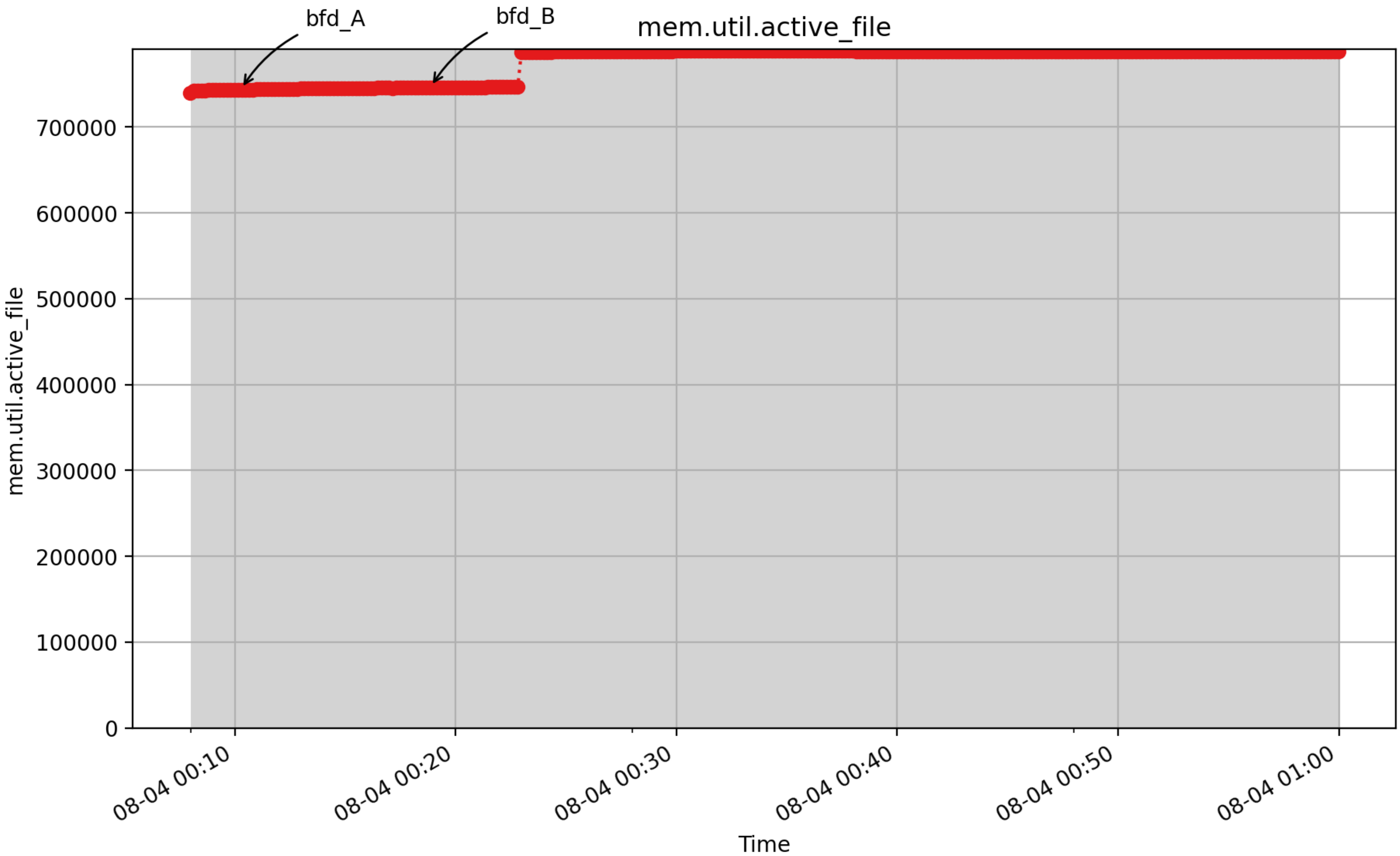
mem.slabinfo.slabs.total_size: total number of bytes allocated for active objects in each slab (byte - U64)



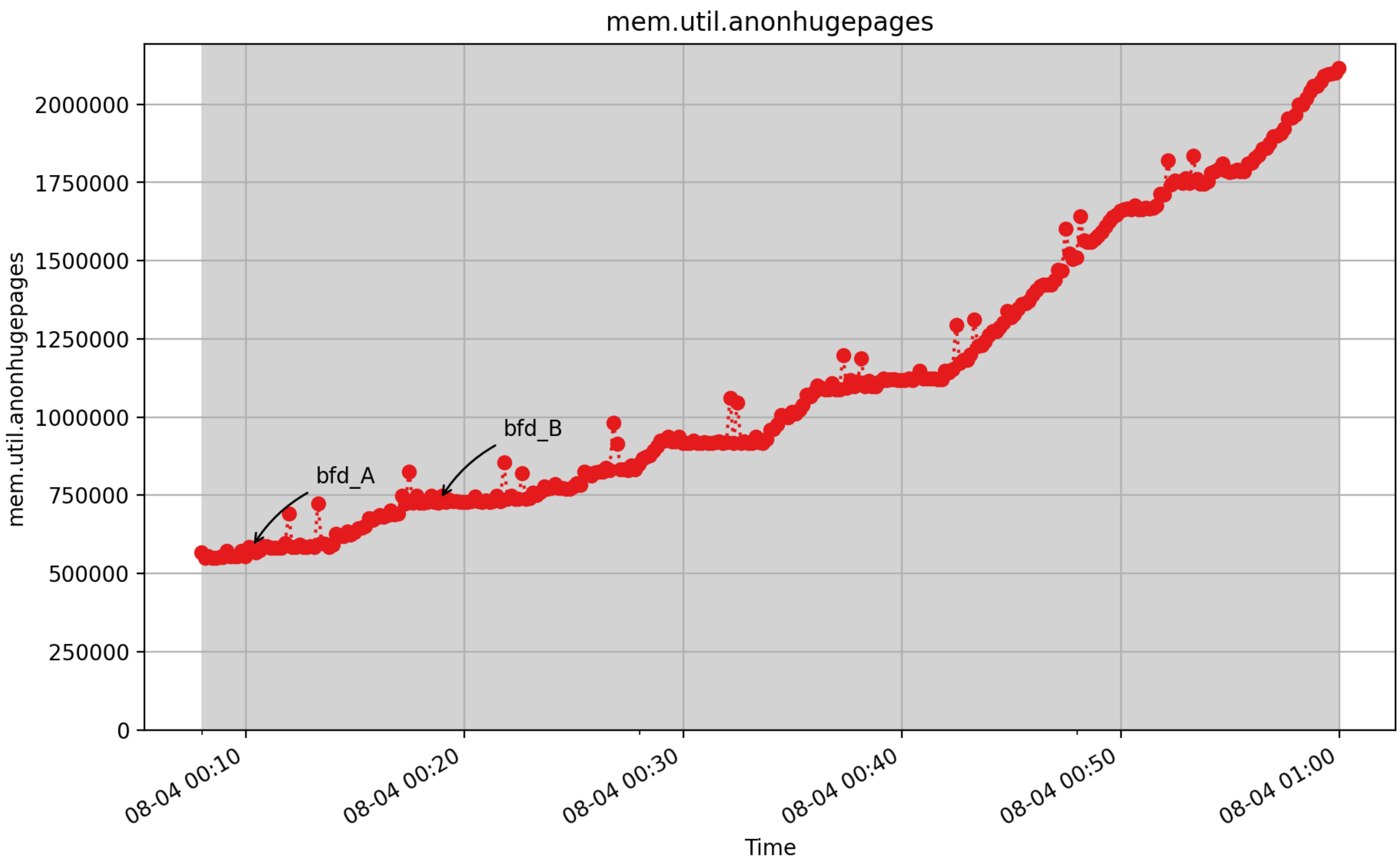
mem.util.active: Memory that has been used more recently and usually not reclaimed unless absolutely necessary. (Kbyte - U64)



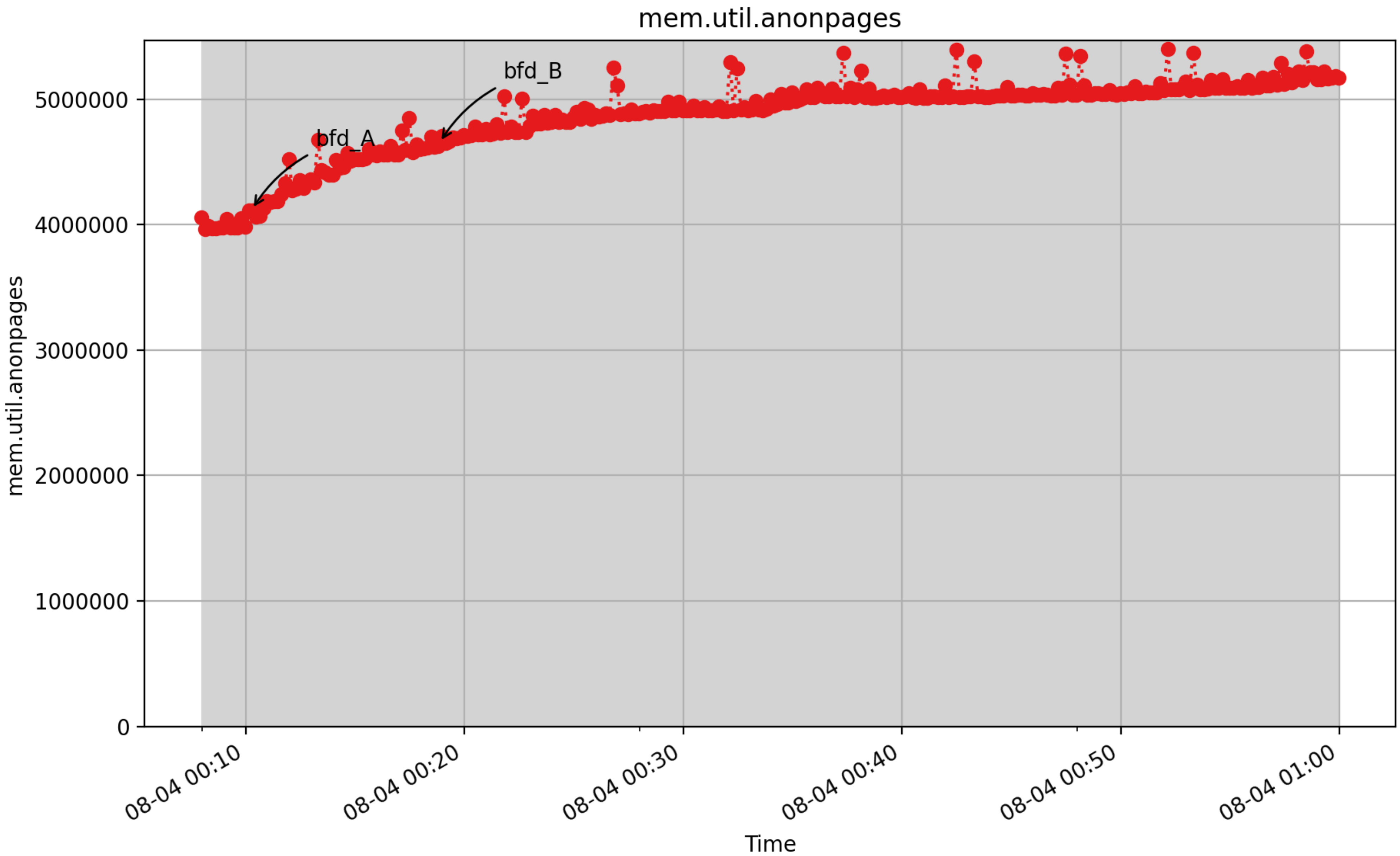
mem.util.active_anon: anonymous Active list LRU memory (Kbyte - U64)



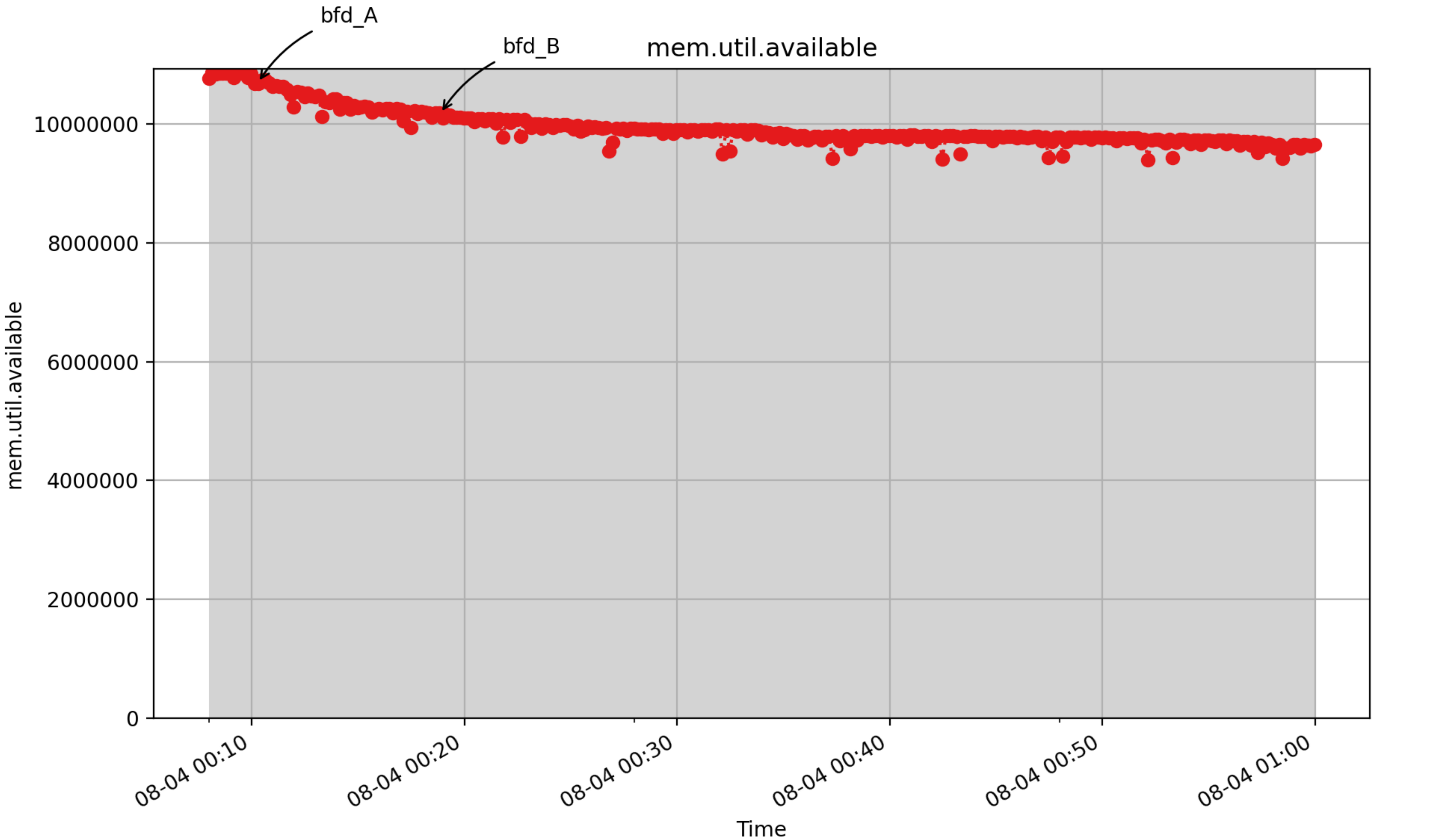
mem.util.active_file: file-backed Active list LRU memory (Kbyte - U64)



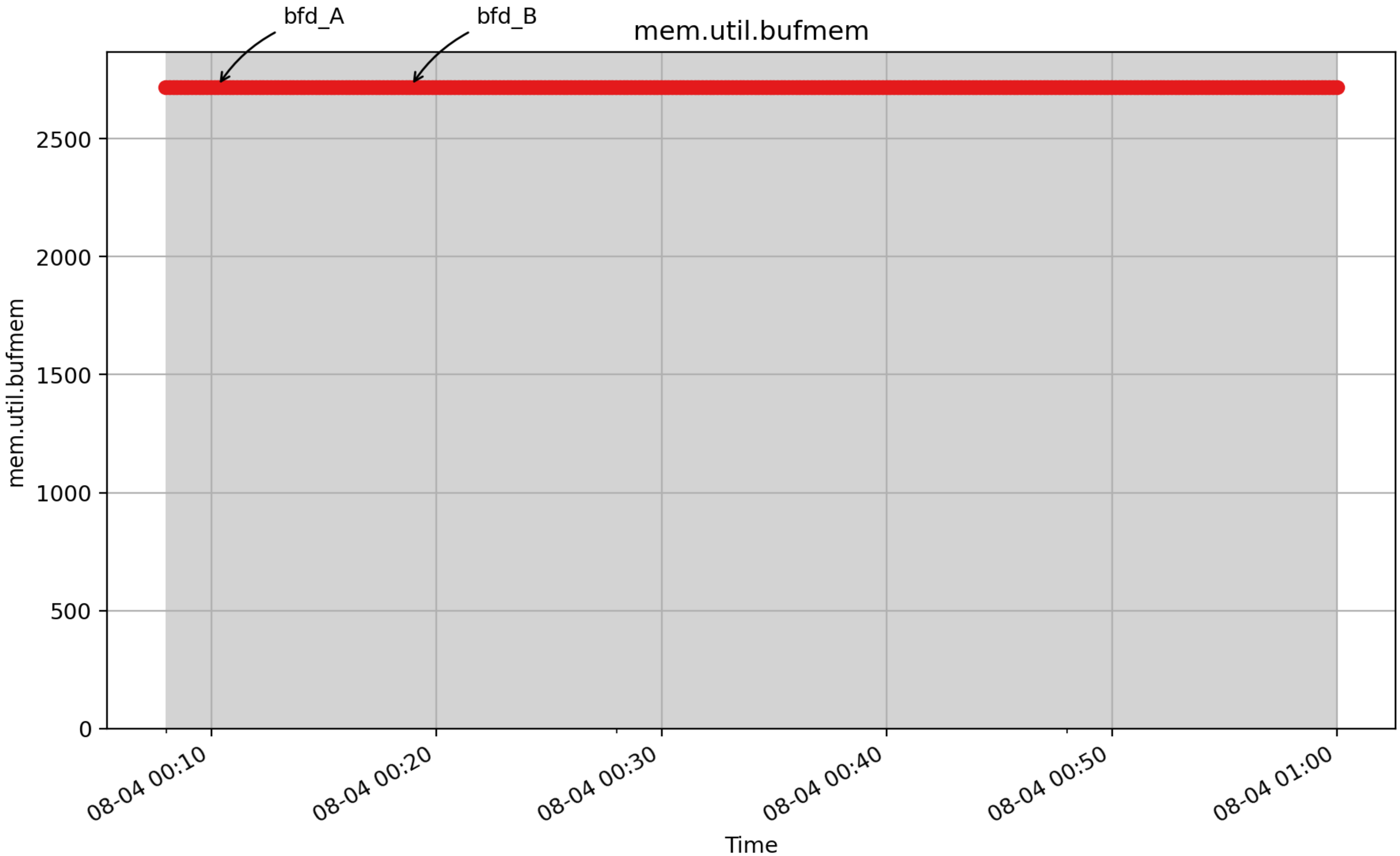
mem.util.anonhugepages: amount of memory in anonymous huge pages (Kbyte - U64)



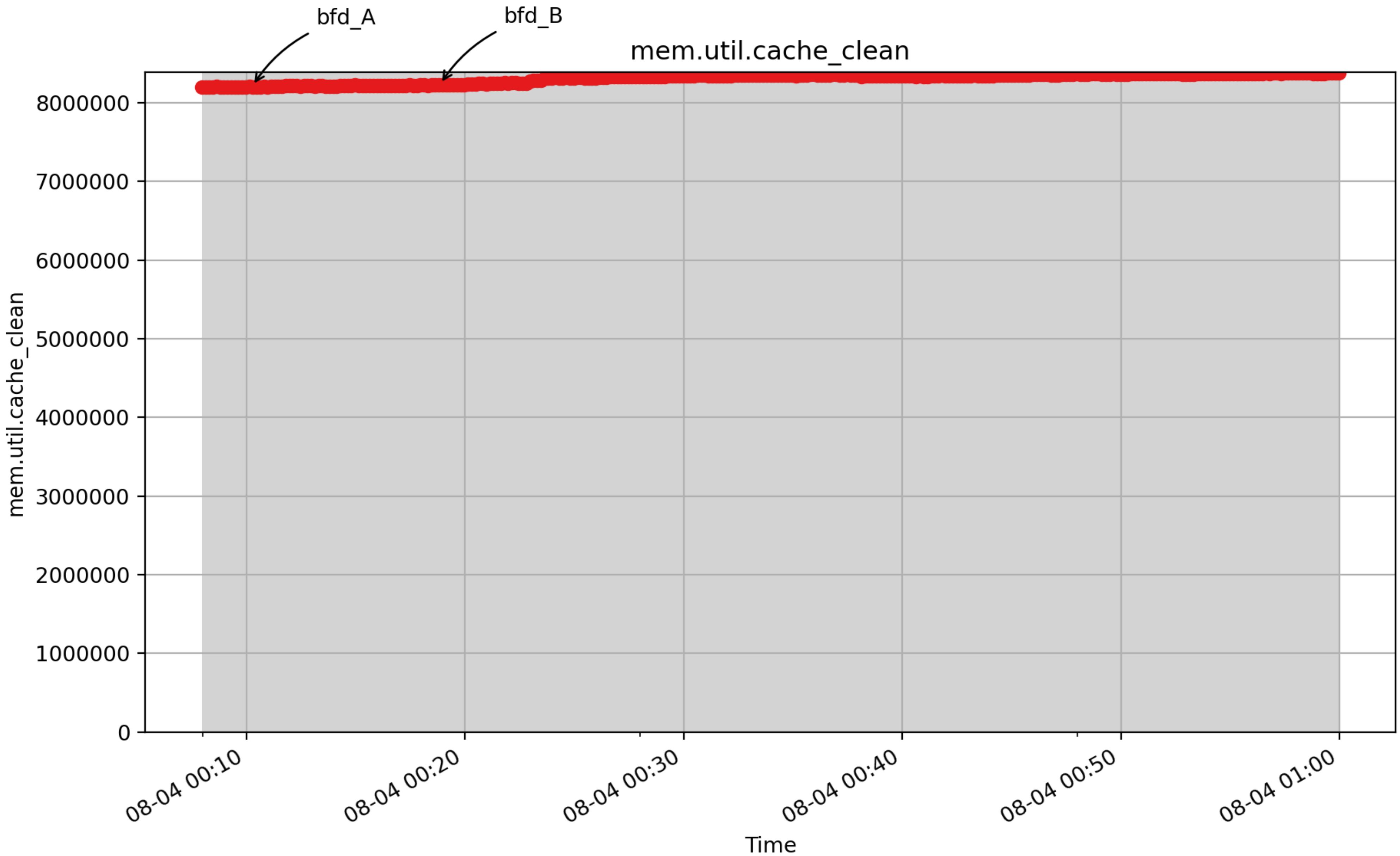
mem.util.anonpages: Kbytes in user pages not backed by files, from /proc/meminfo (Kbyte - U64)



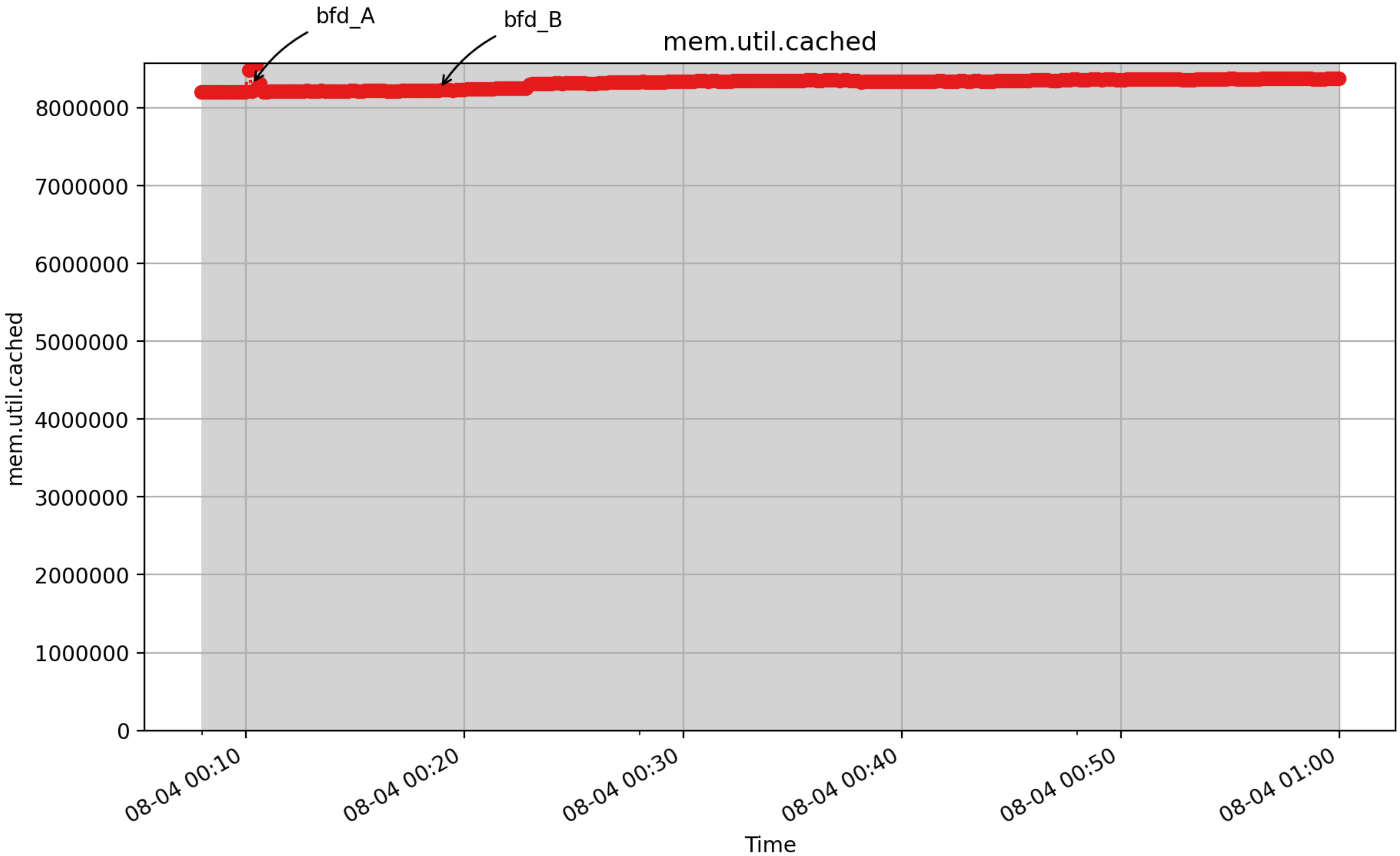
mem.util.available: The amount of memory that is available for a new workload, without pushing the system into swap. Estimated from MemFree, Active(file), Inactive(file), and SReclaimable, as well as the "low" watermarks from /proc/zoneinfo. (Kbyte - U64)



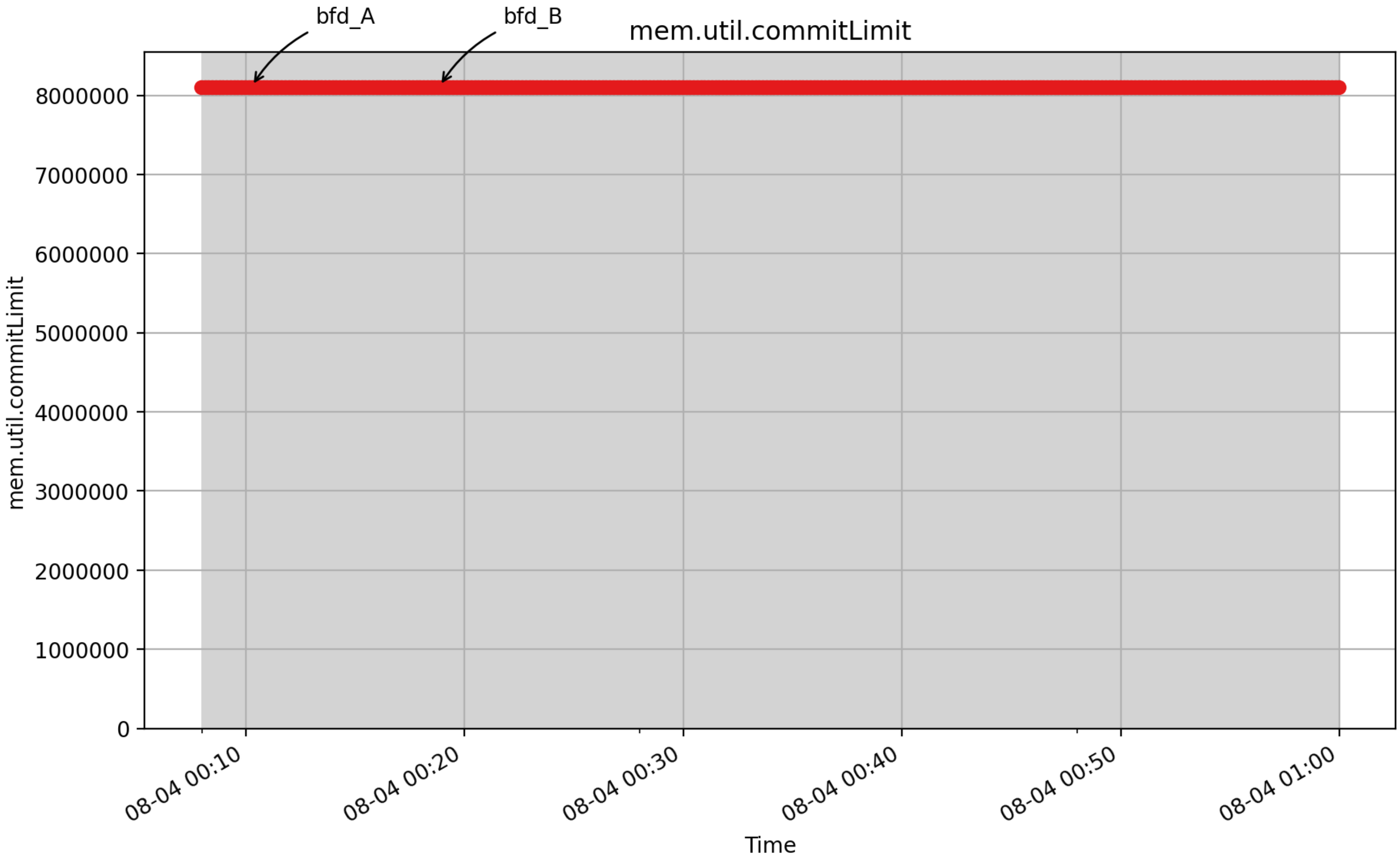
mem.util.bufmem: Memory allocated for buffer_heads. (Kbyte - U64)



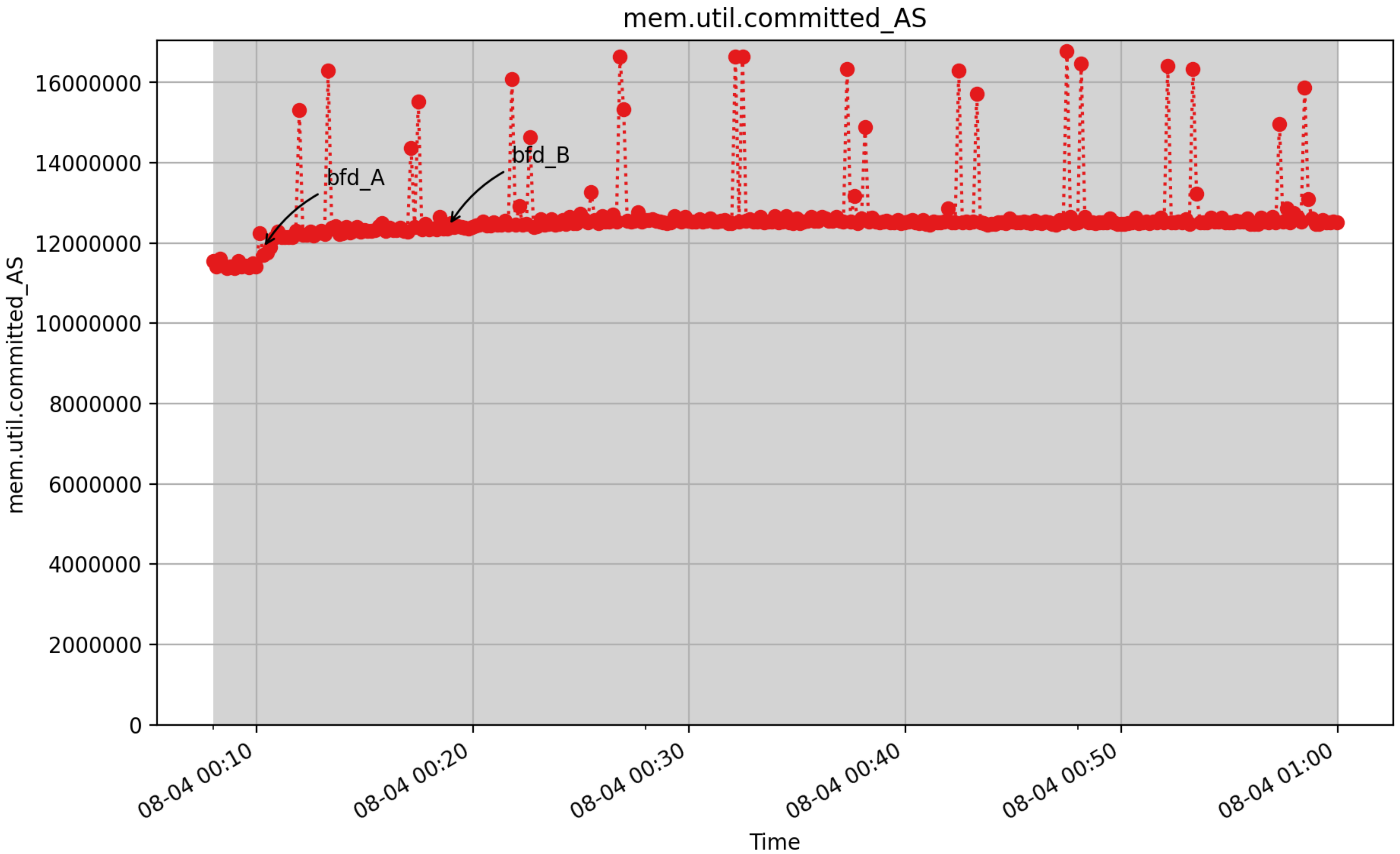
mem.util.cache_clean: Kbytes cached and not dirty or writeback, derived from /proc/meminfo (Kbyte - U64)



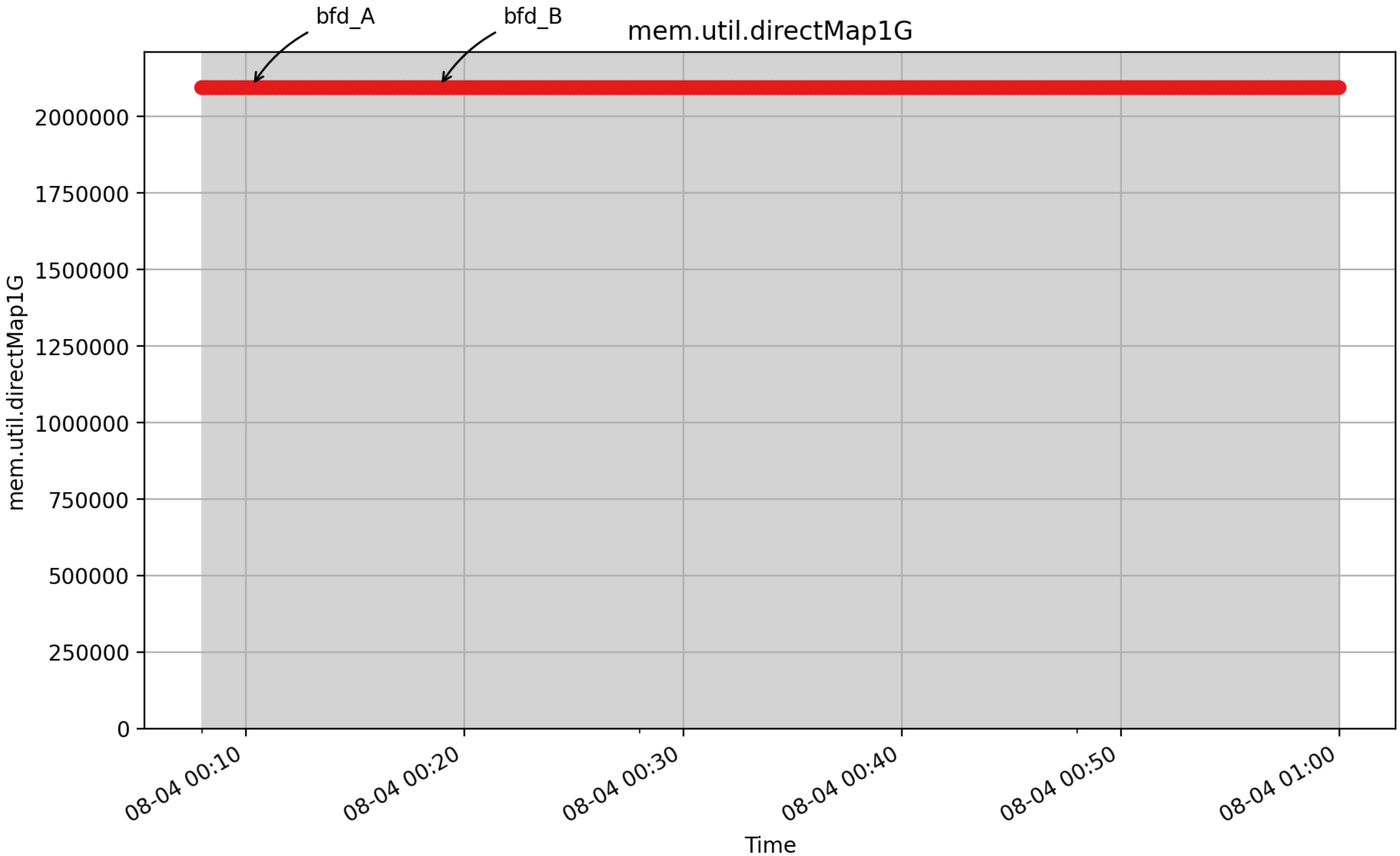
mem.util.cached: Memory used by the page cache, including buffered file data. This is in-memory cache for files read from the disk (the pagecache) but doesn't include SwapCached. (Kbyte - U64)



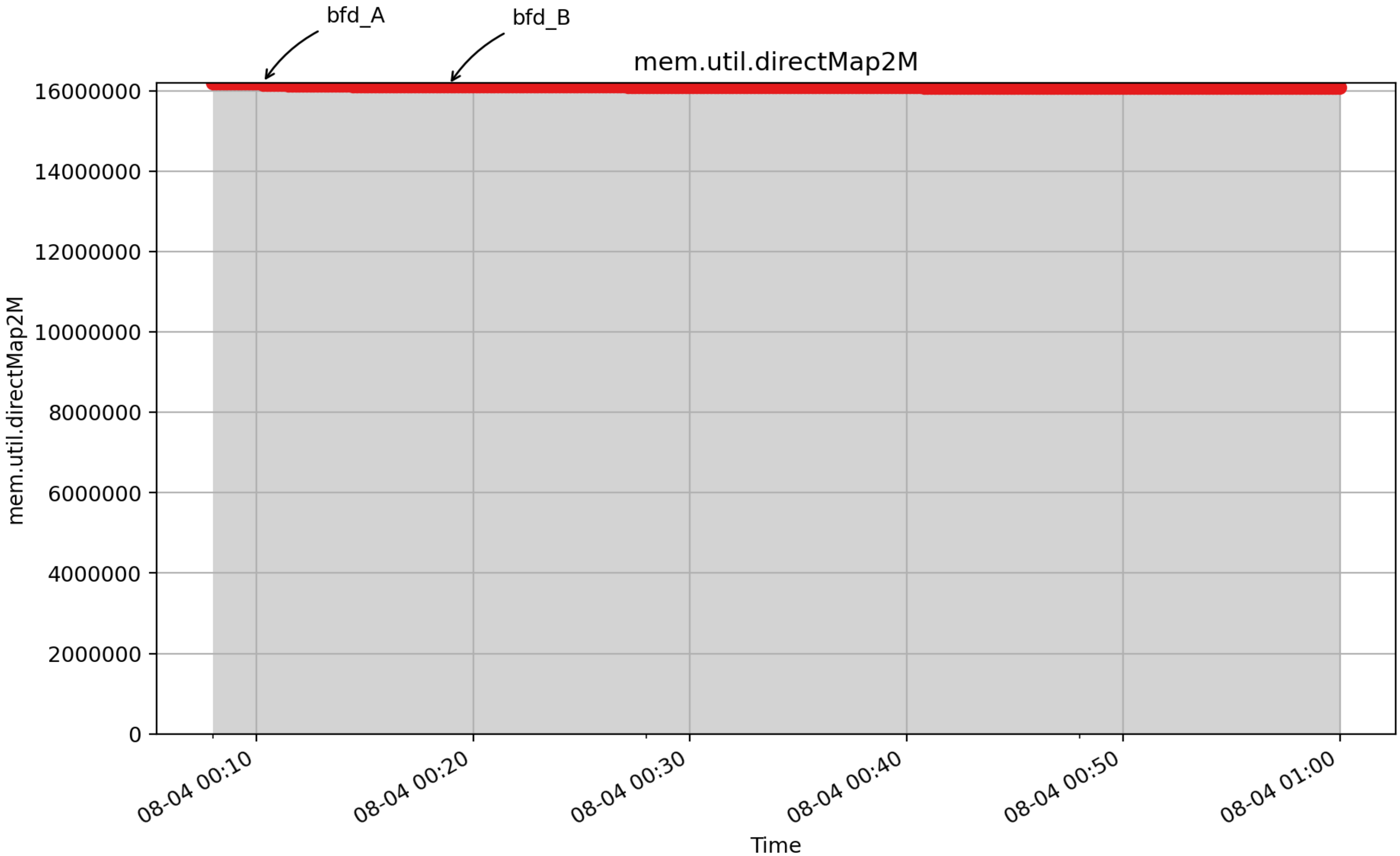
mem.util.commitLimit: The static total, in Kbytes, available for commitment to address spaces. Thus, mem.util.committed_AS may range up to this total. Normally the kernel overcommits memory, so this value may exceed mem.physmem (Kbyte - U64)



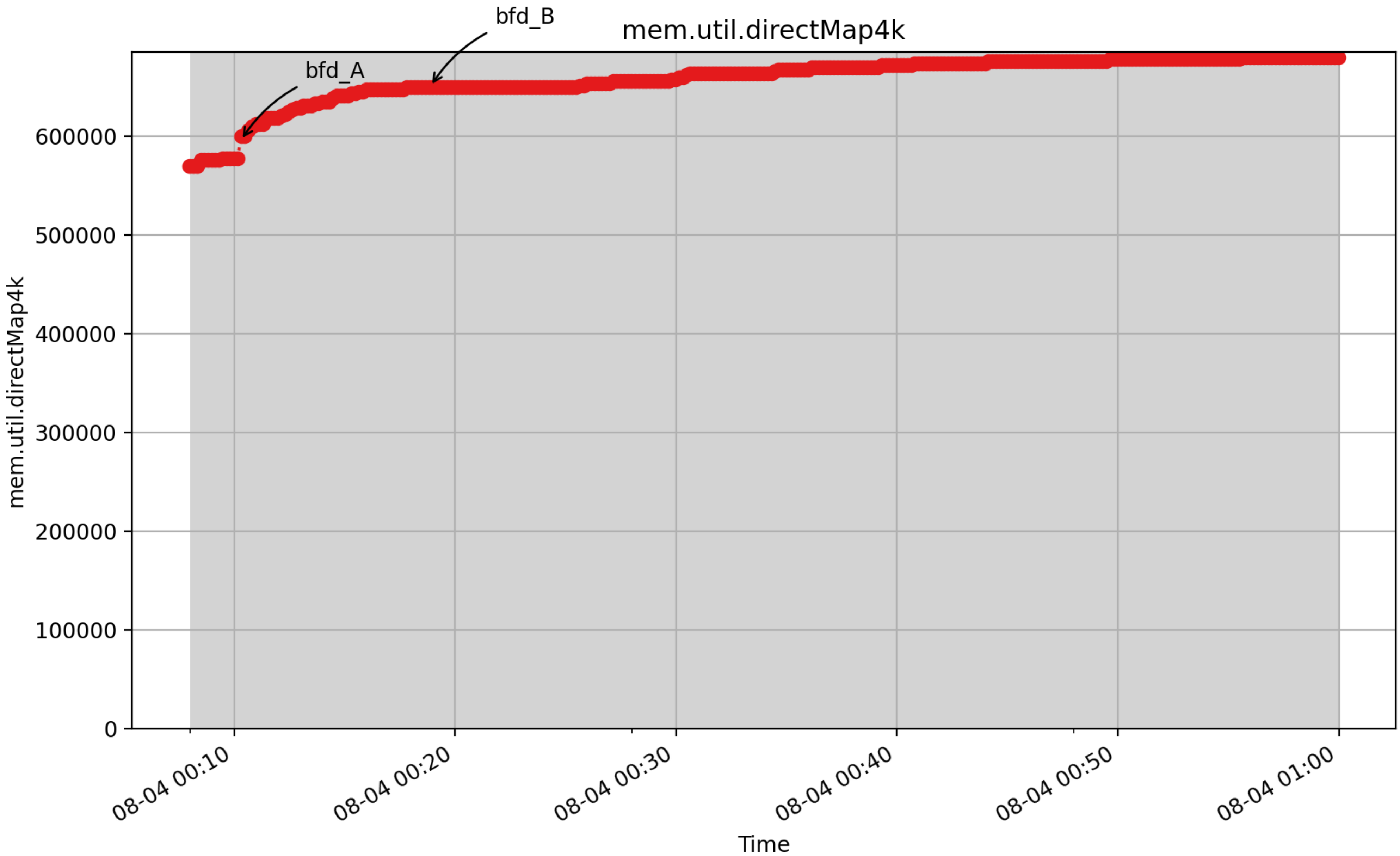
mem.util.committed_AS: An estimate of how much RAM you would need to make a 99.99% guarantee that there never is OOM (out of memory) for this workload. Normally the kernel will overcommit memory. That means, say you do a 1GB malloc, nothing happens, really. Only when you start USING that malloc memory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



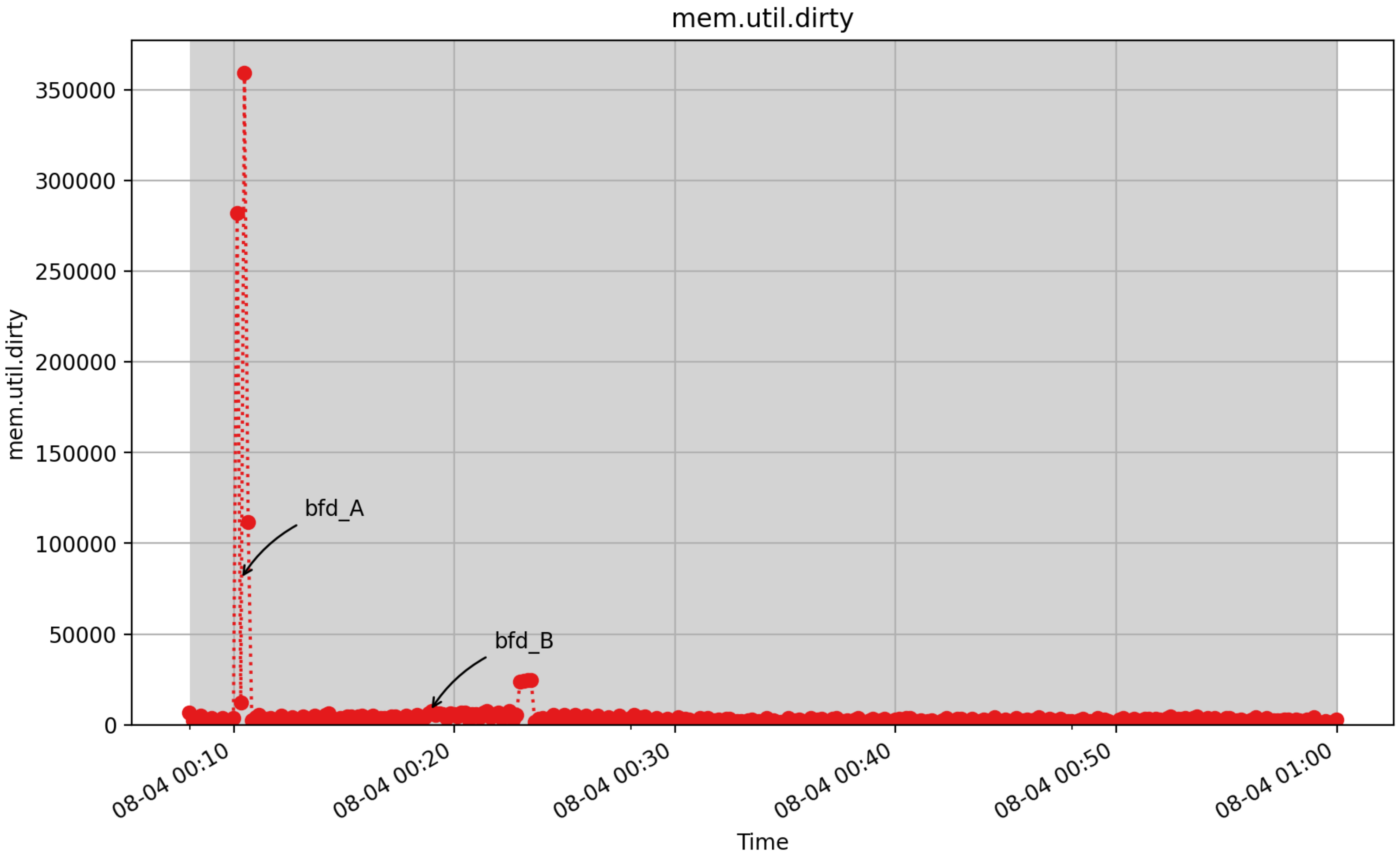
mem.util.directMap1G: amount of memory that is directly mapped in 1GB pages (Kbyte - U64)



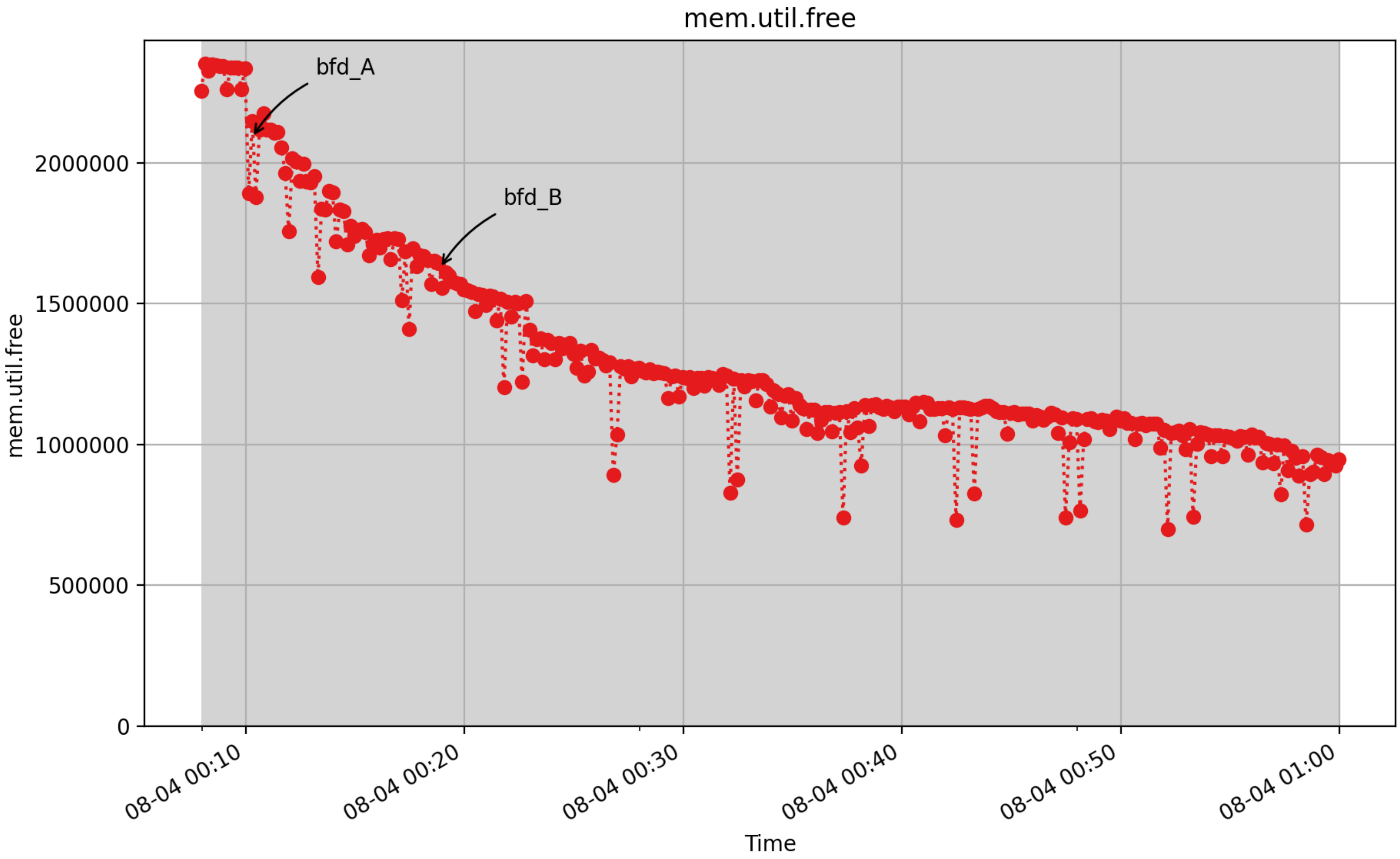
mem.util.directMap2M: amount of memory that is directly mapped in 2MB pages (Kbyte - U64)



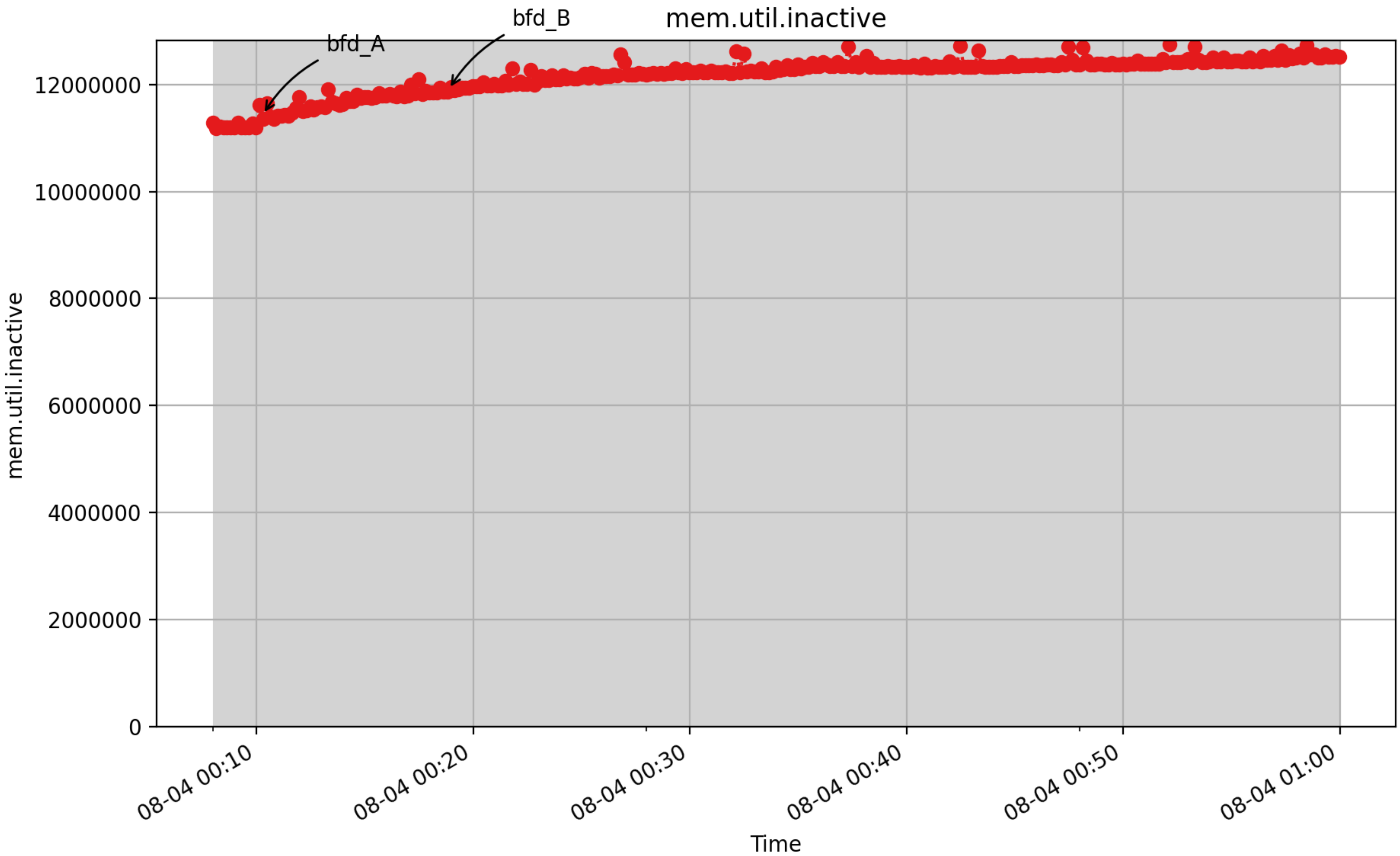
mem.util.directMap4k: amount of memory that is directly mapped in 4kB pages (Kbyte - U64)



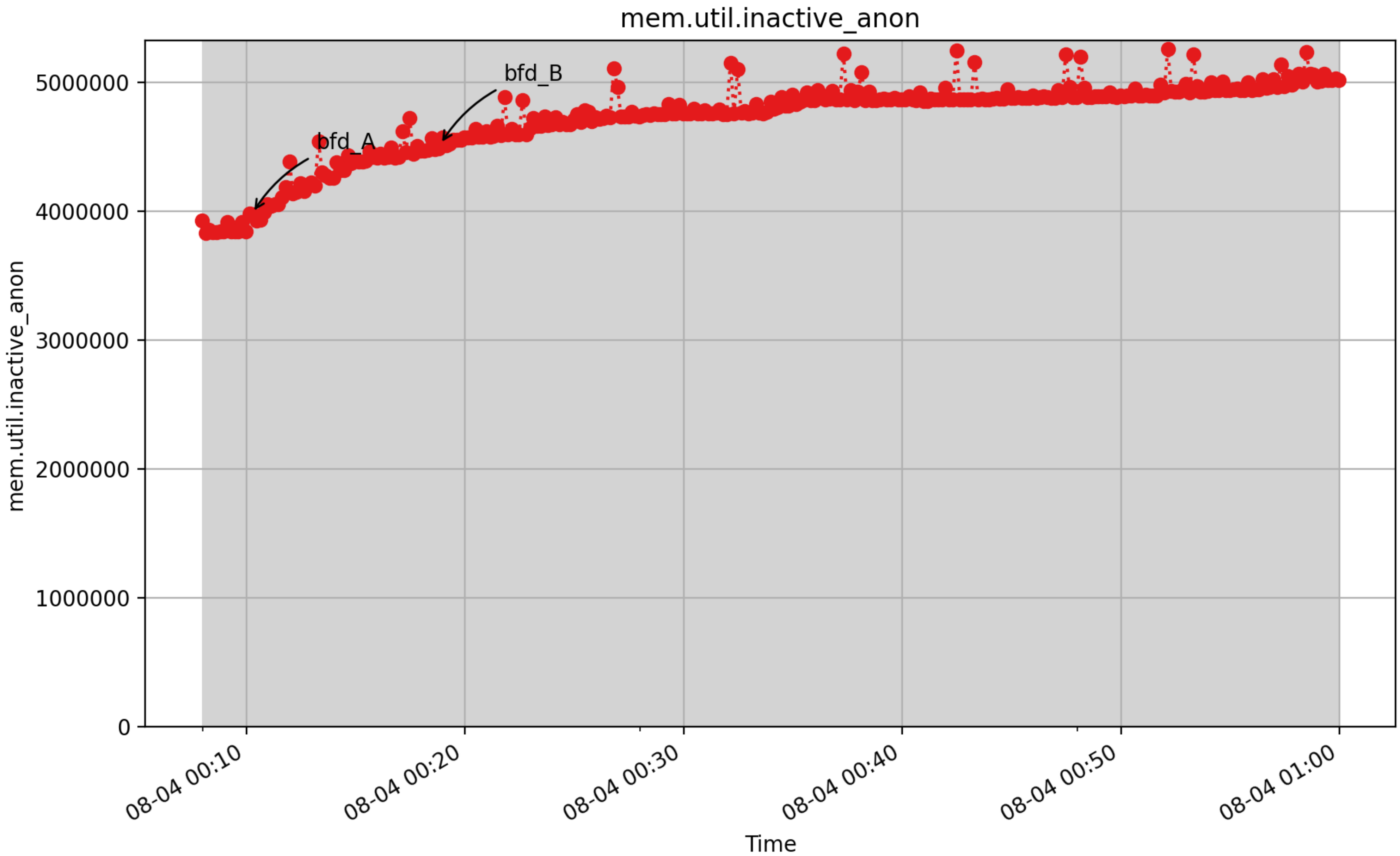
mem.util.dirty: Memory which is waiting to get written back to the disk (Kbyte - U64)



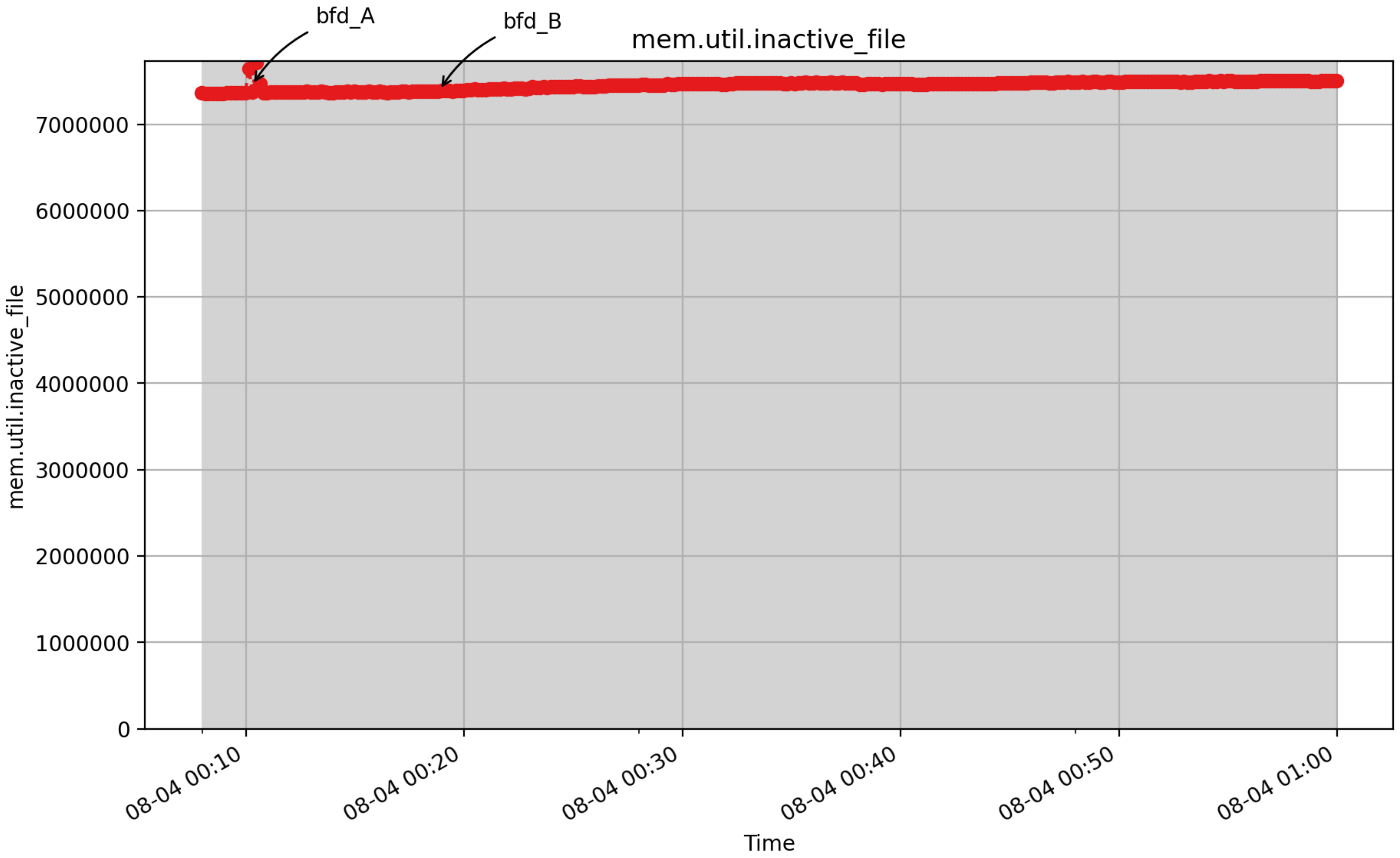
mem.util.free: Alias for mem.freemem. (Kbyte - U64)



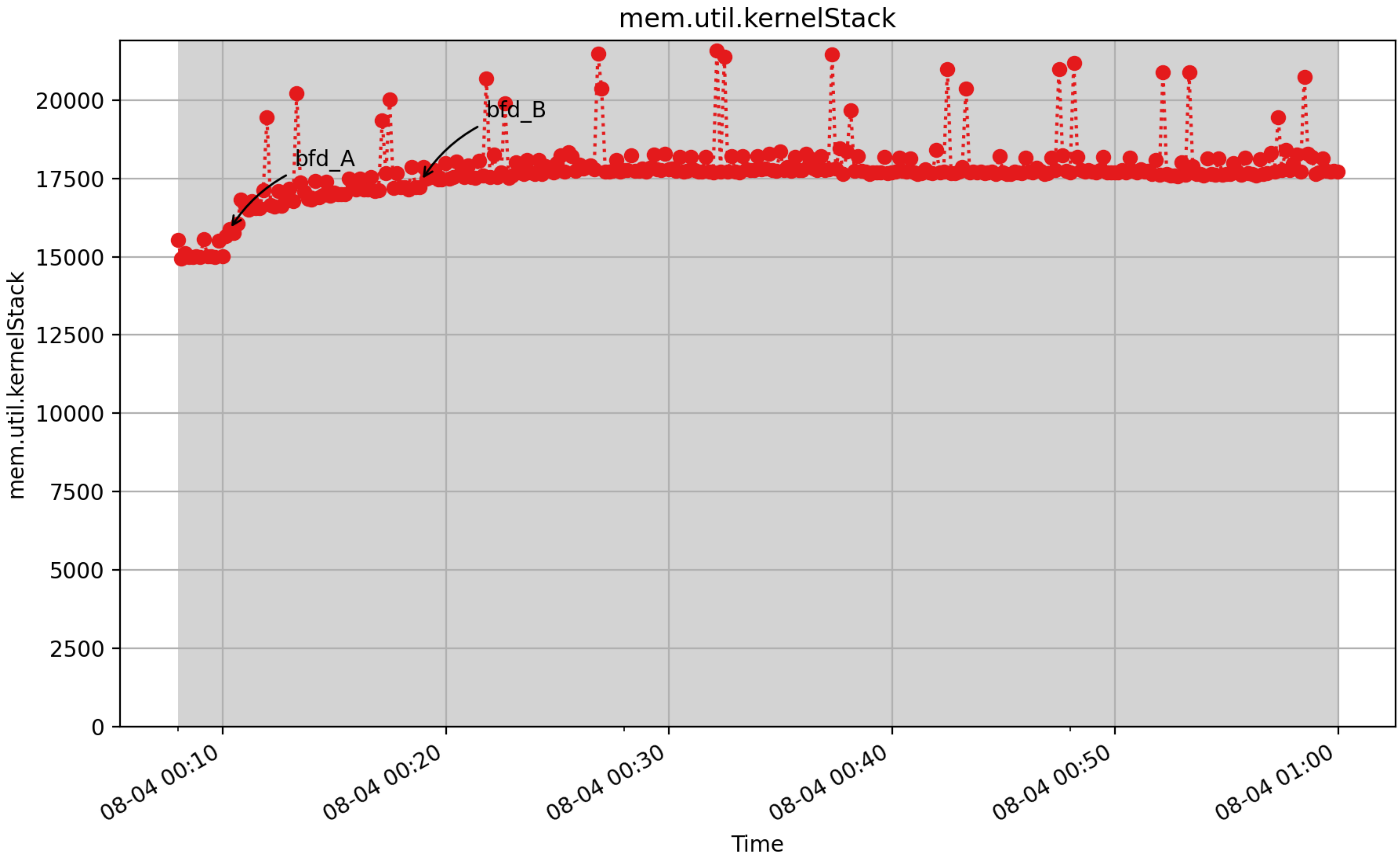
mem.util.inactive: Memory which has been less recently used. It is more eligible to be reclaimed for other purposes (Kbyte - U64)



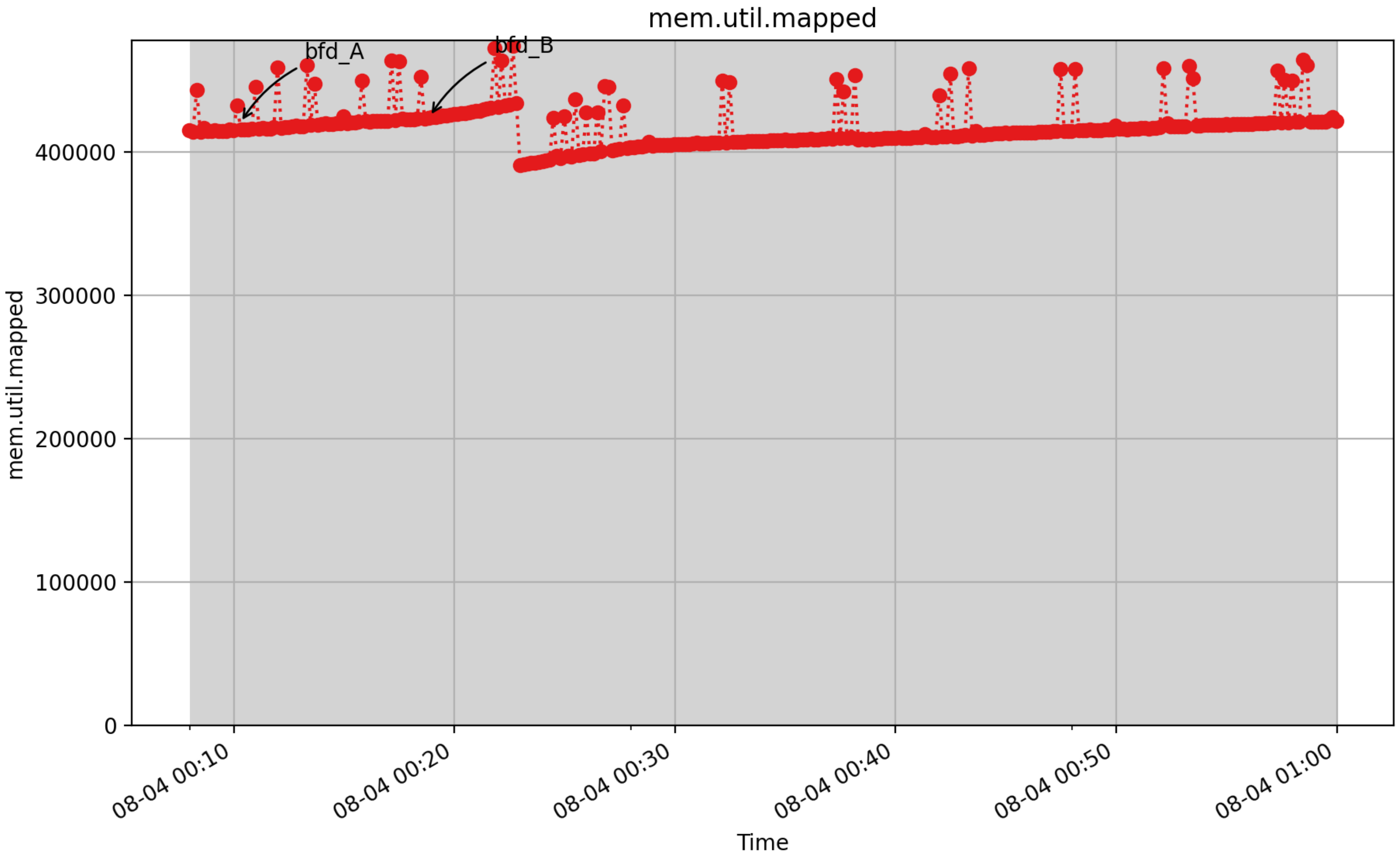
mem.util.inactive_anon: anonymous Inactive list LRU memory (Kbyte - U64)



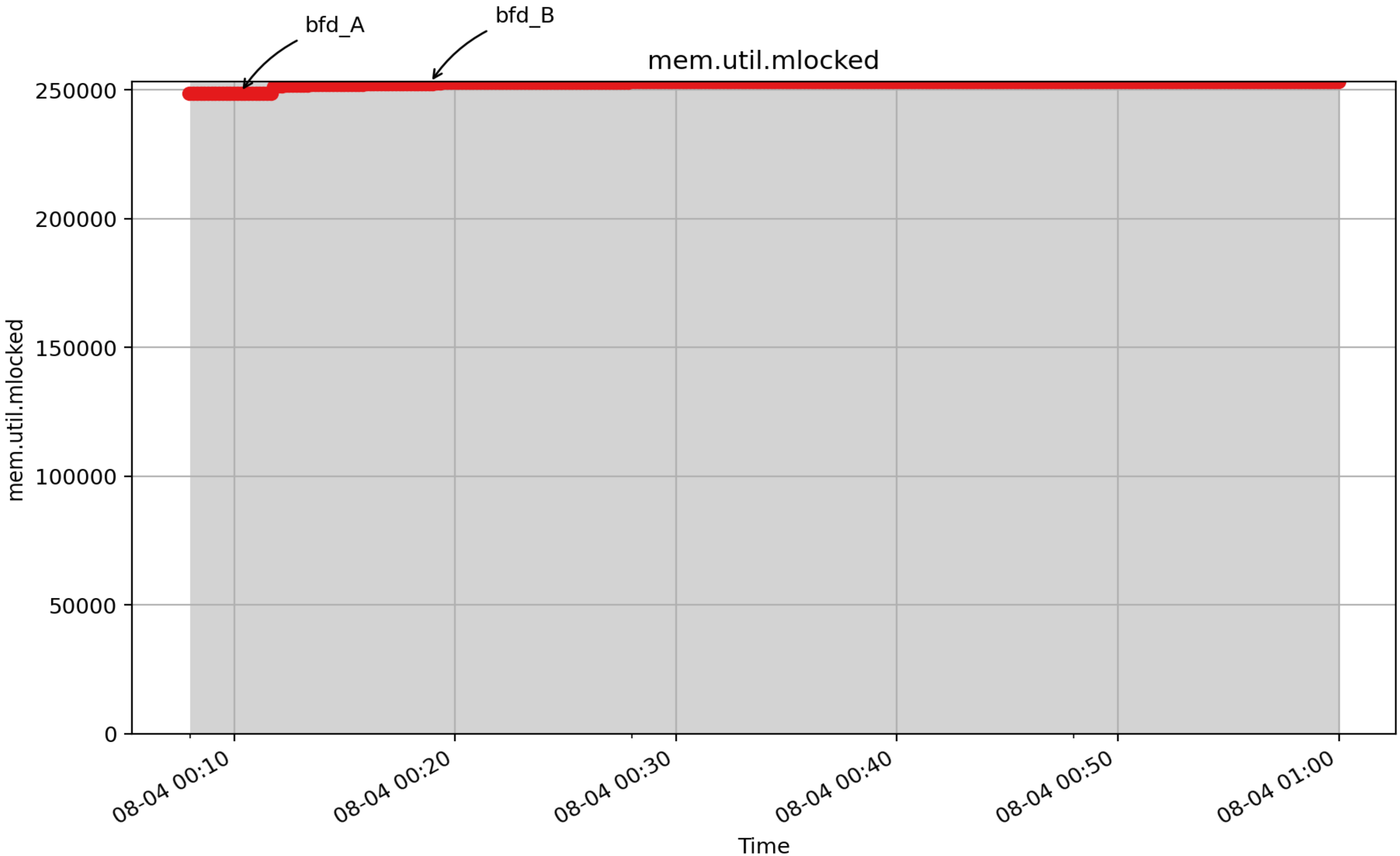
mem.util.inactive_file: file-backed Inactive list LRU memory (Kbyte - U64)



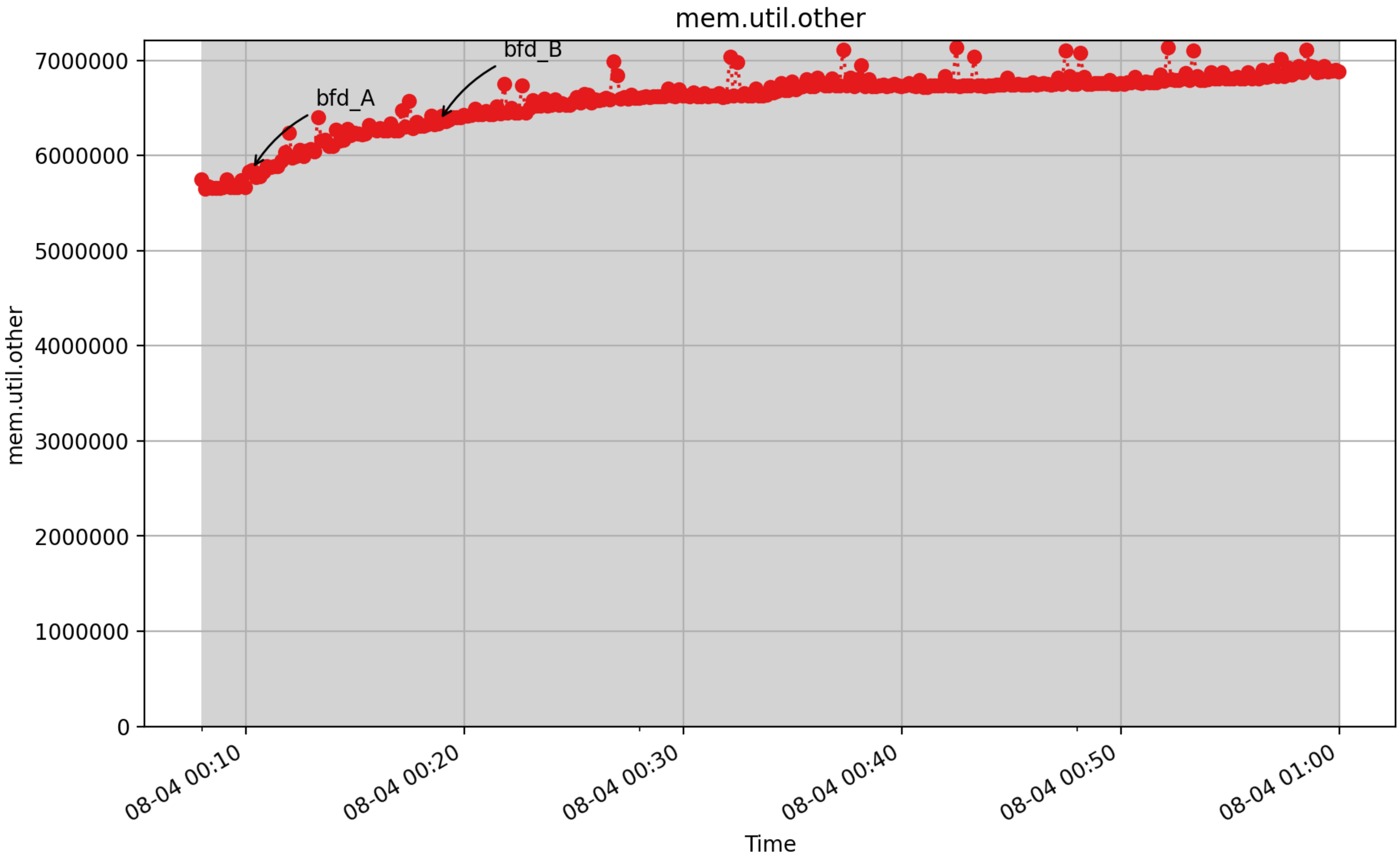
mem.util.kernelStack: kbytes of memory used for kernel stacks (Kbyte - U64)



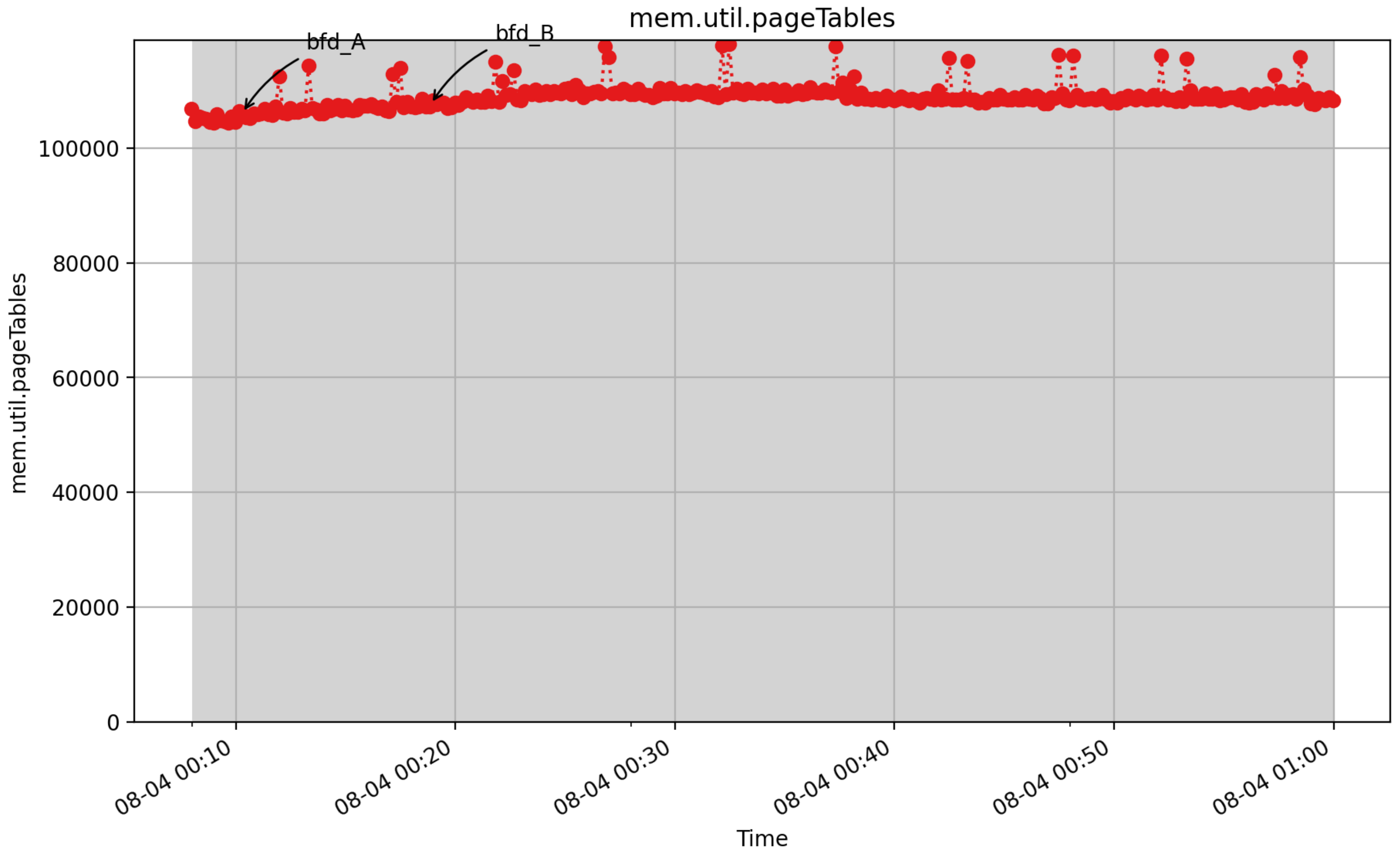
mem.util.mapped: files which have been mmaped, such as libraries (Kbyte - U64)



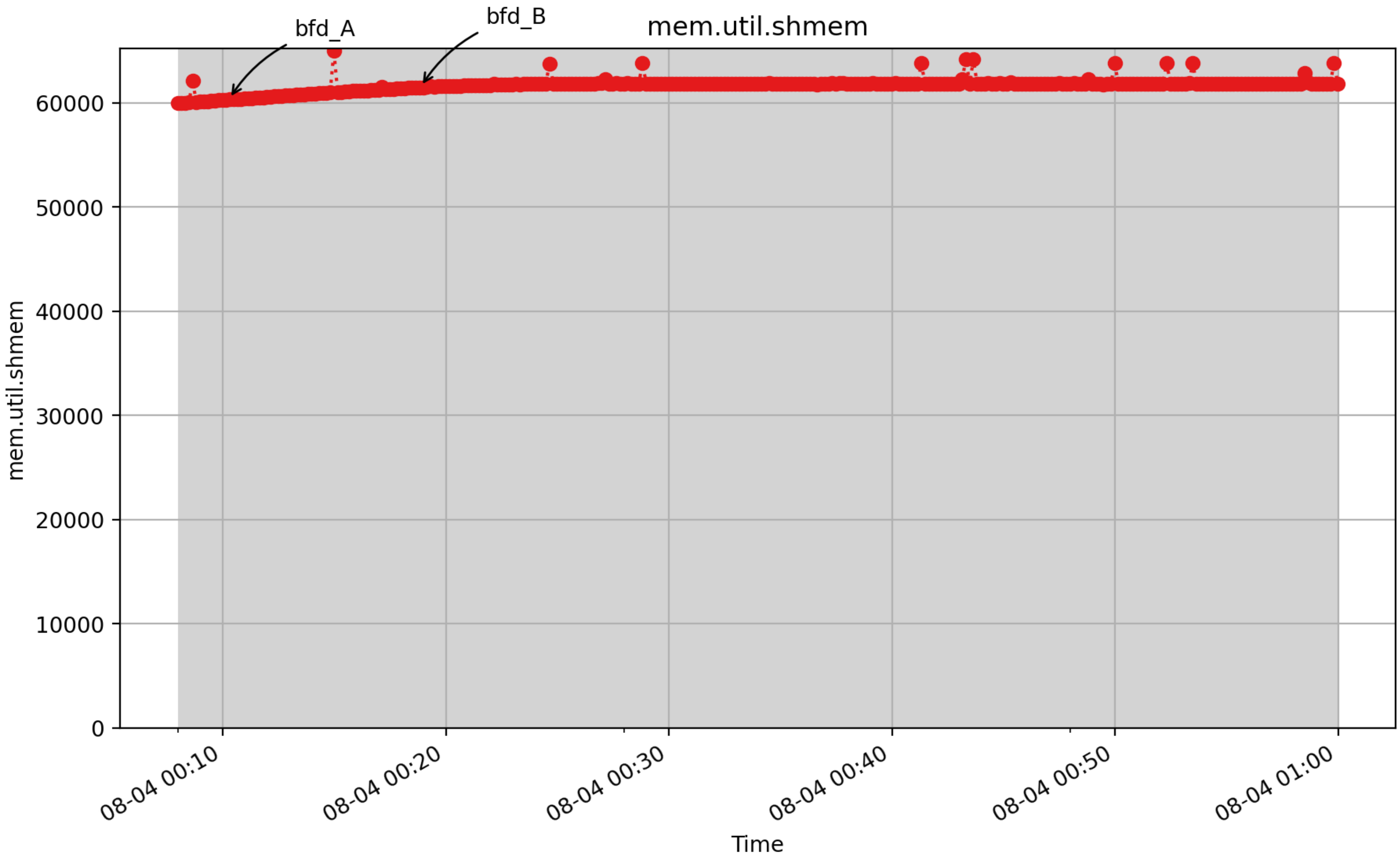
mem.util.mlocked: kbytes of memory that is pinned via mlock() (Kbyte - U64)



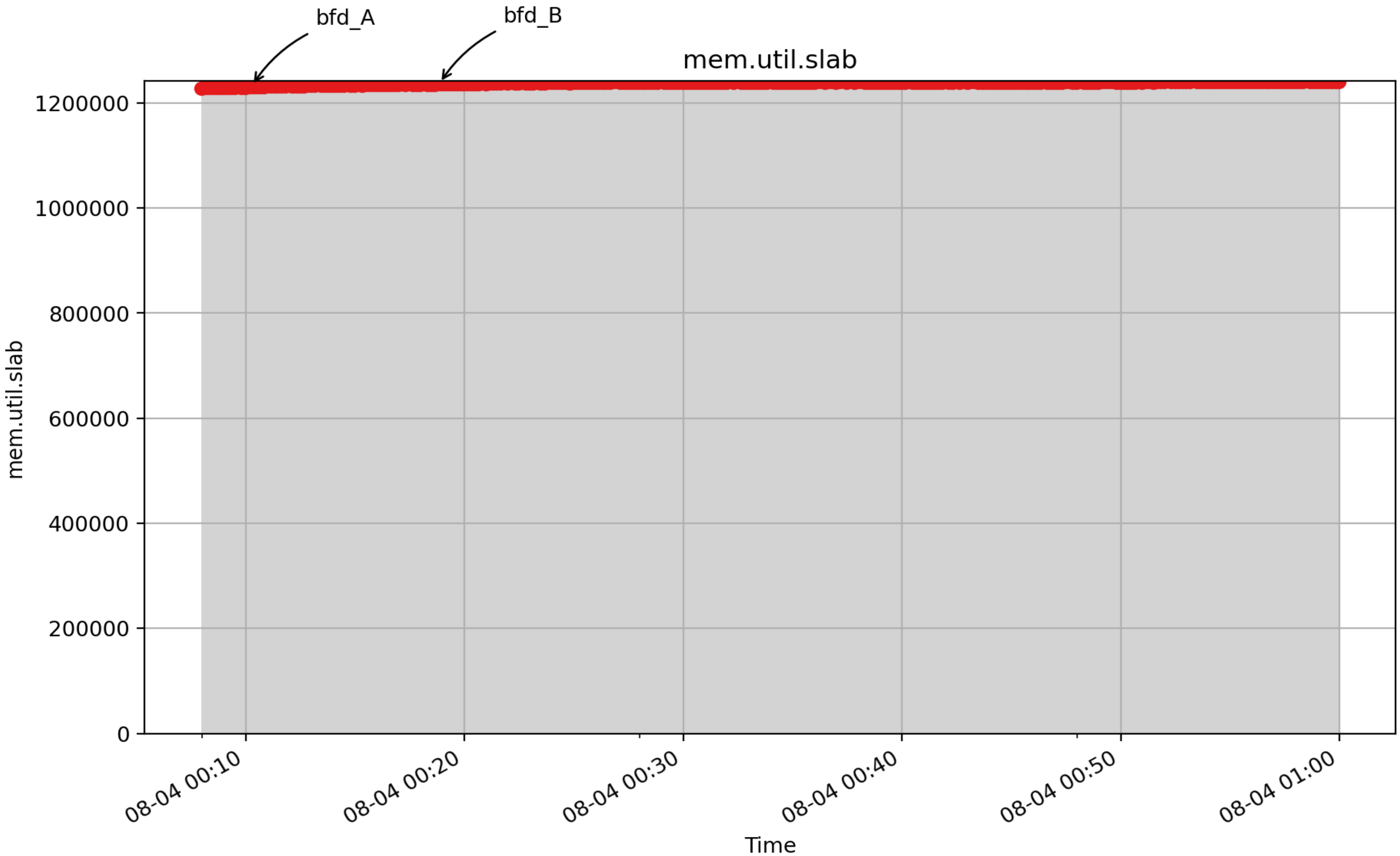
mem.util.other: Memory that is not free (i.e. has been referenced) and is not cached. mem.physmem - mem.util.free - mem.util.cached - mem.util.buffers (Kbyte - U64)



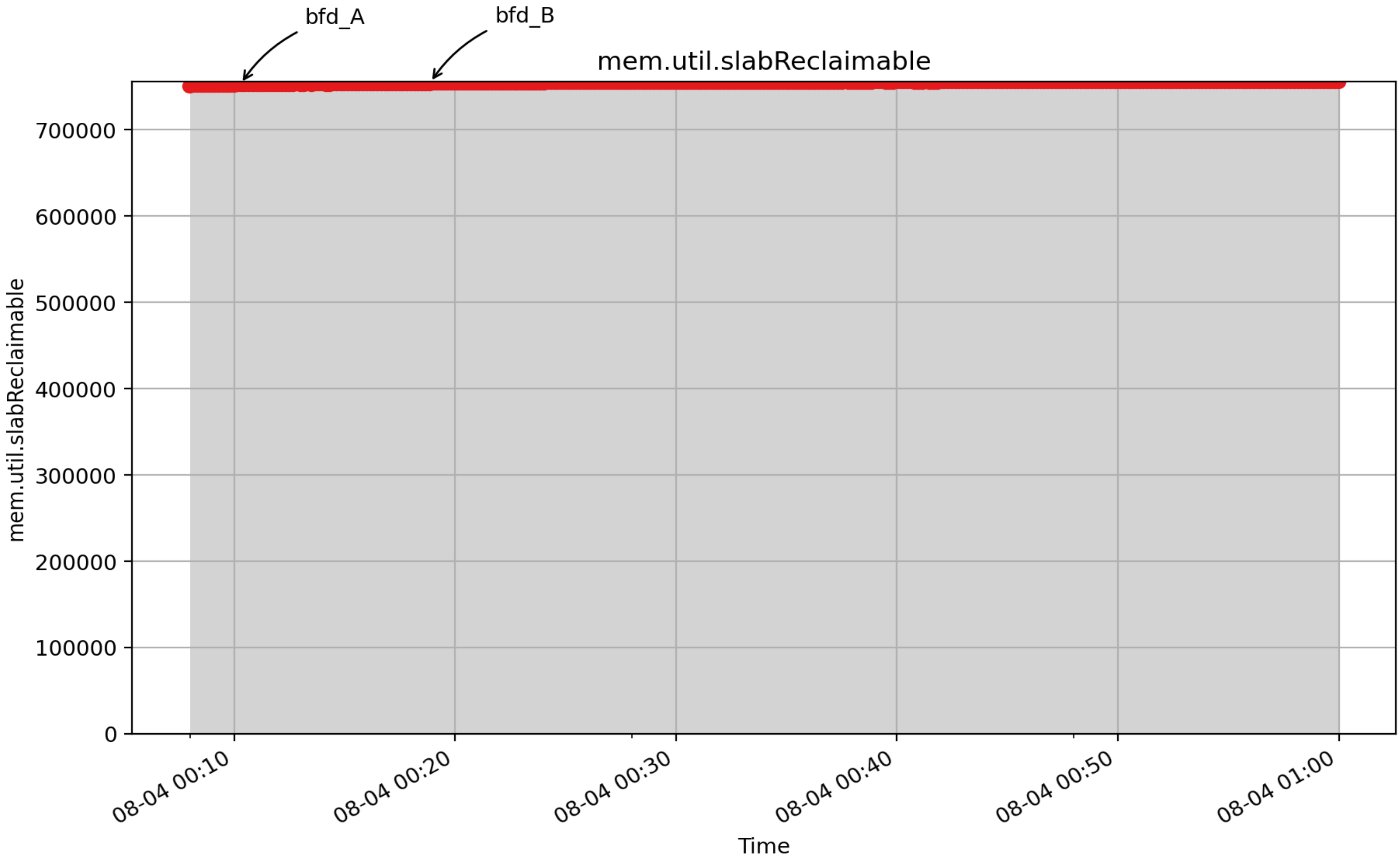
mem.util.pageTables: Kbytes in kernel page tables, from /proc/meminfo (Kbyte - U64)



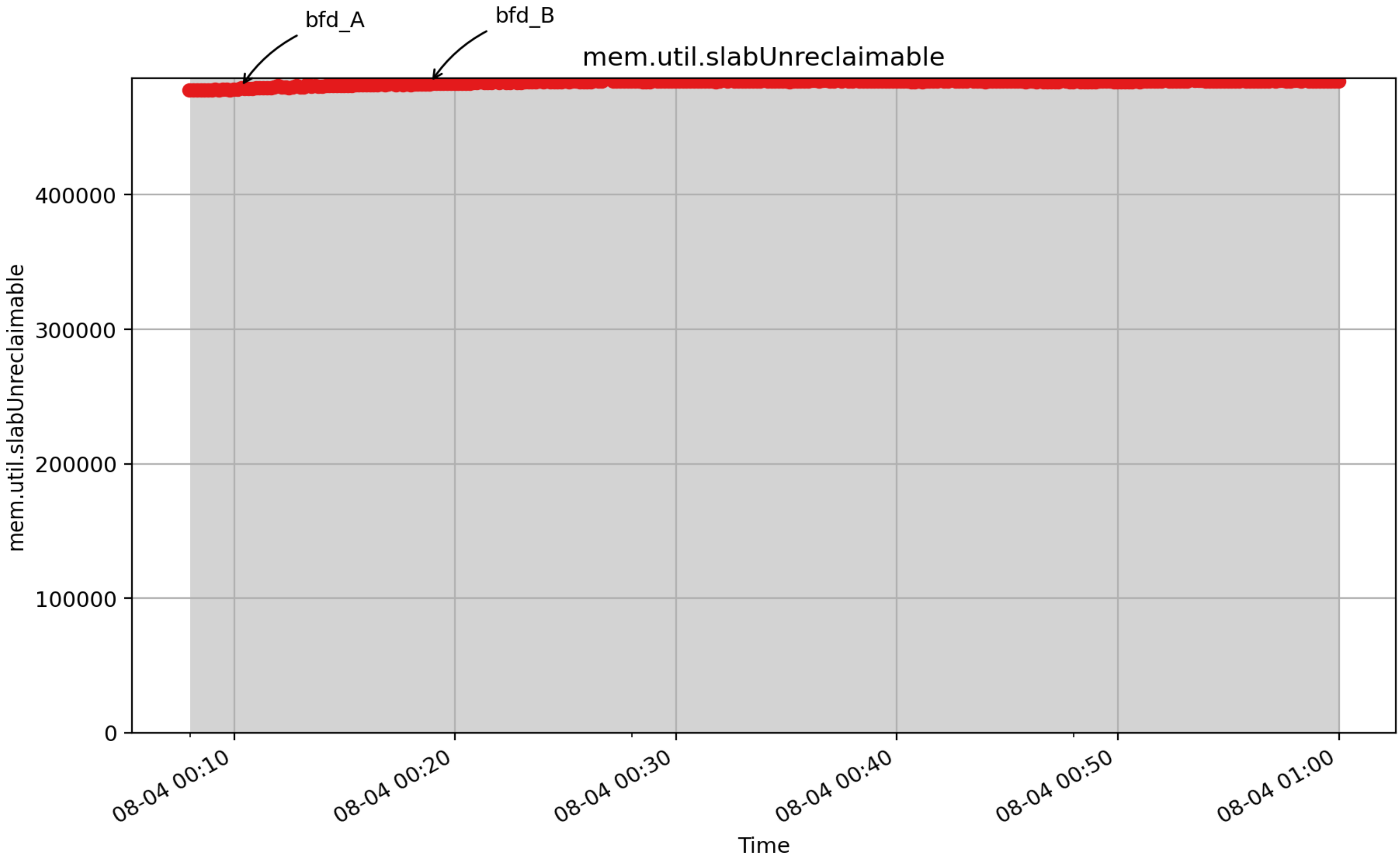
mem.util.shmem: kbytes of shmem (Kbyte - U64)



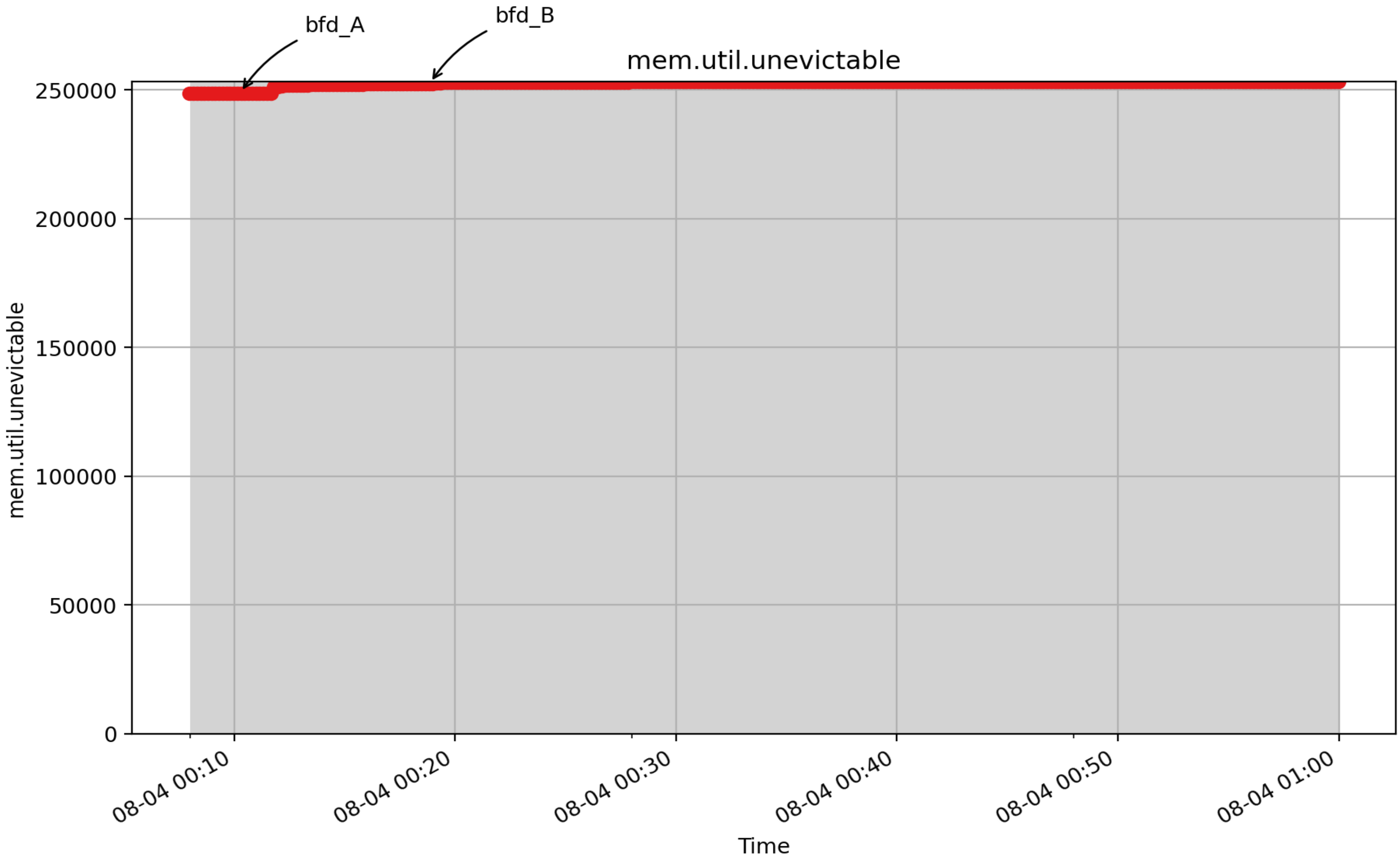
mem.util.slab: in-kernel data structures cache (Kbyte - U64)



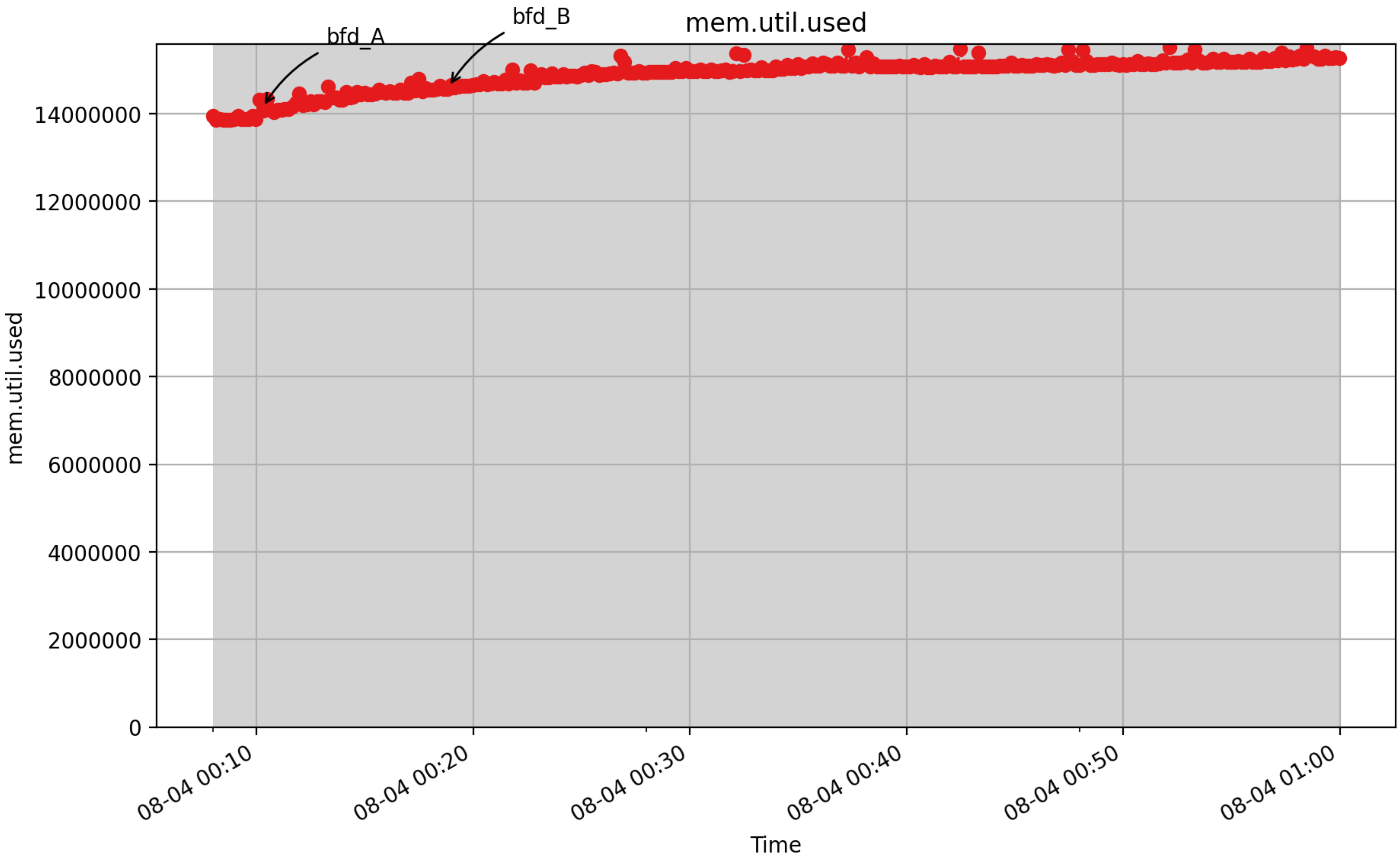
mem.util.slabReclaimable: Kbytes in reclaimable slab pages, from /proc/meminfo (Kbyte - U64)



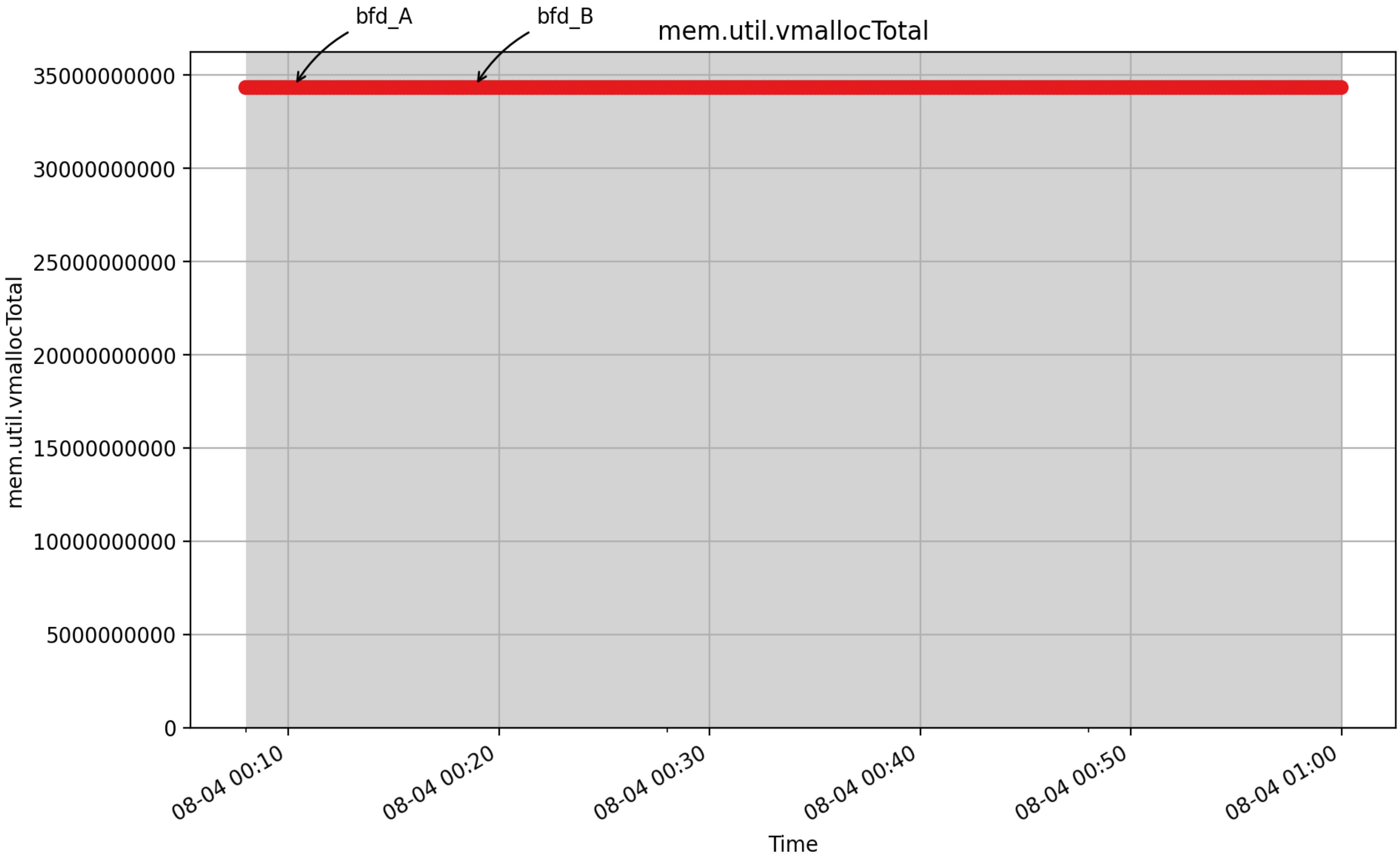
mem.util.slabUnreclaimable: Kbytes in unreclaimable slab pages, from /proc/meminfo (Kbyte - U64)



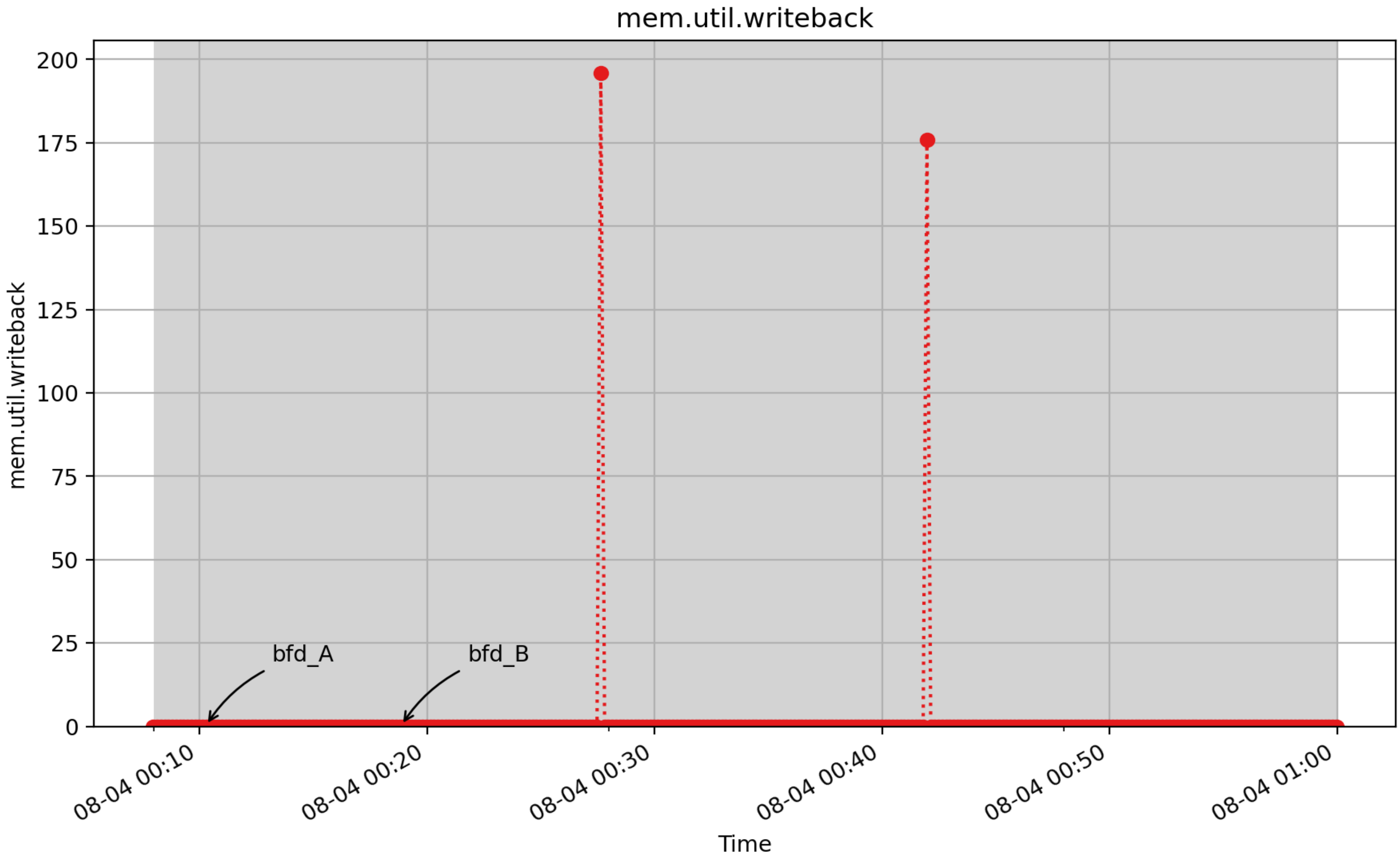
mem.util.unevictable: kbytes of memory that is unevictable (Kbyte - U64)



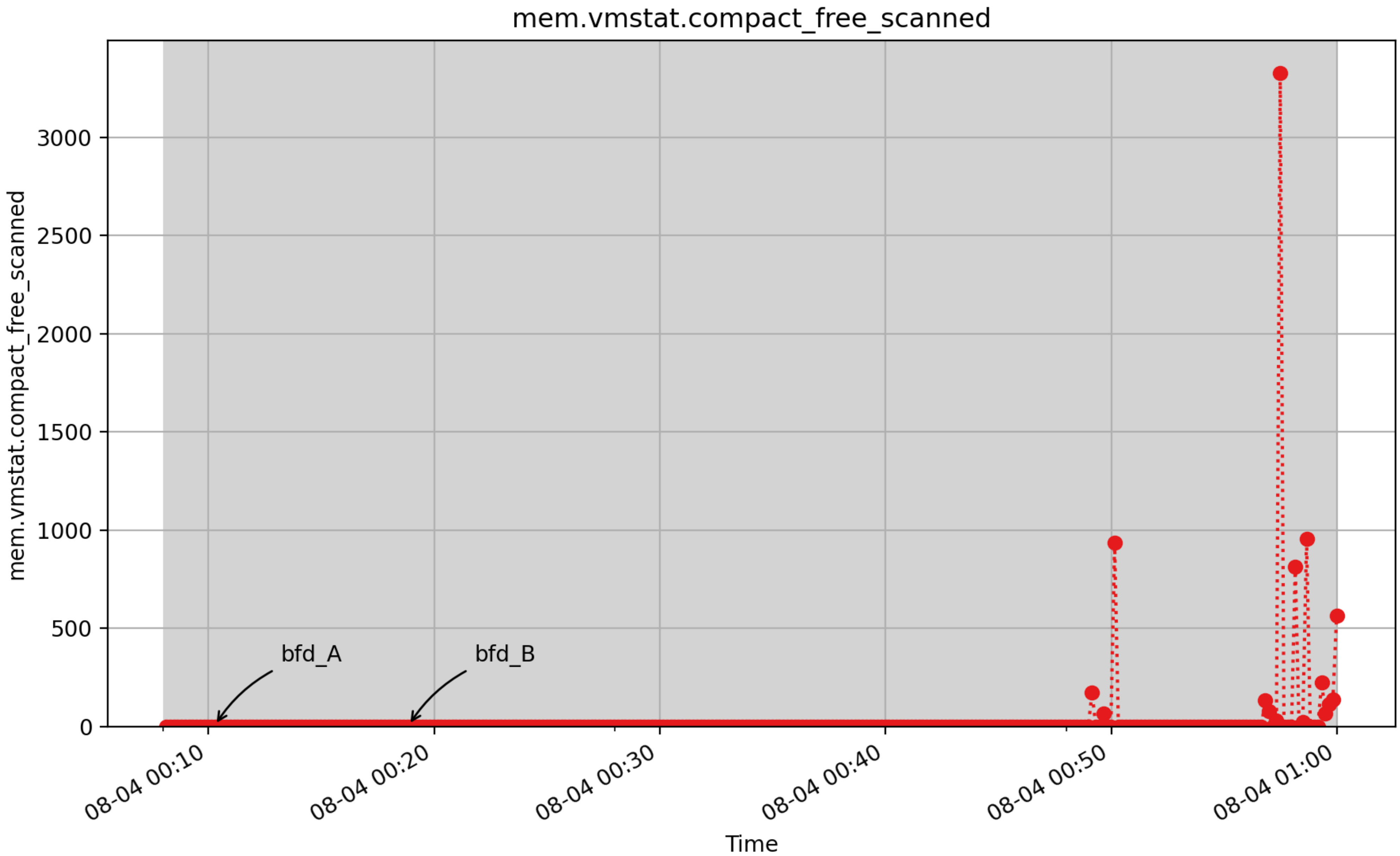
mem.util.used: Used memory is the difference between mem.physmem and mem.freemem. (Kbyte - U64)



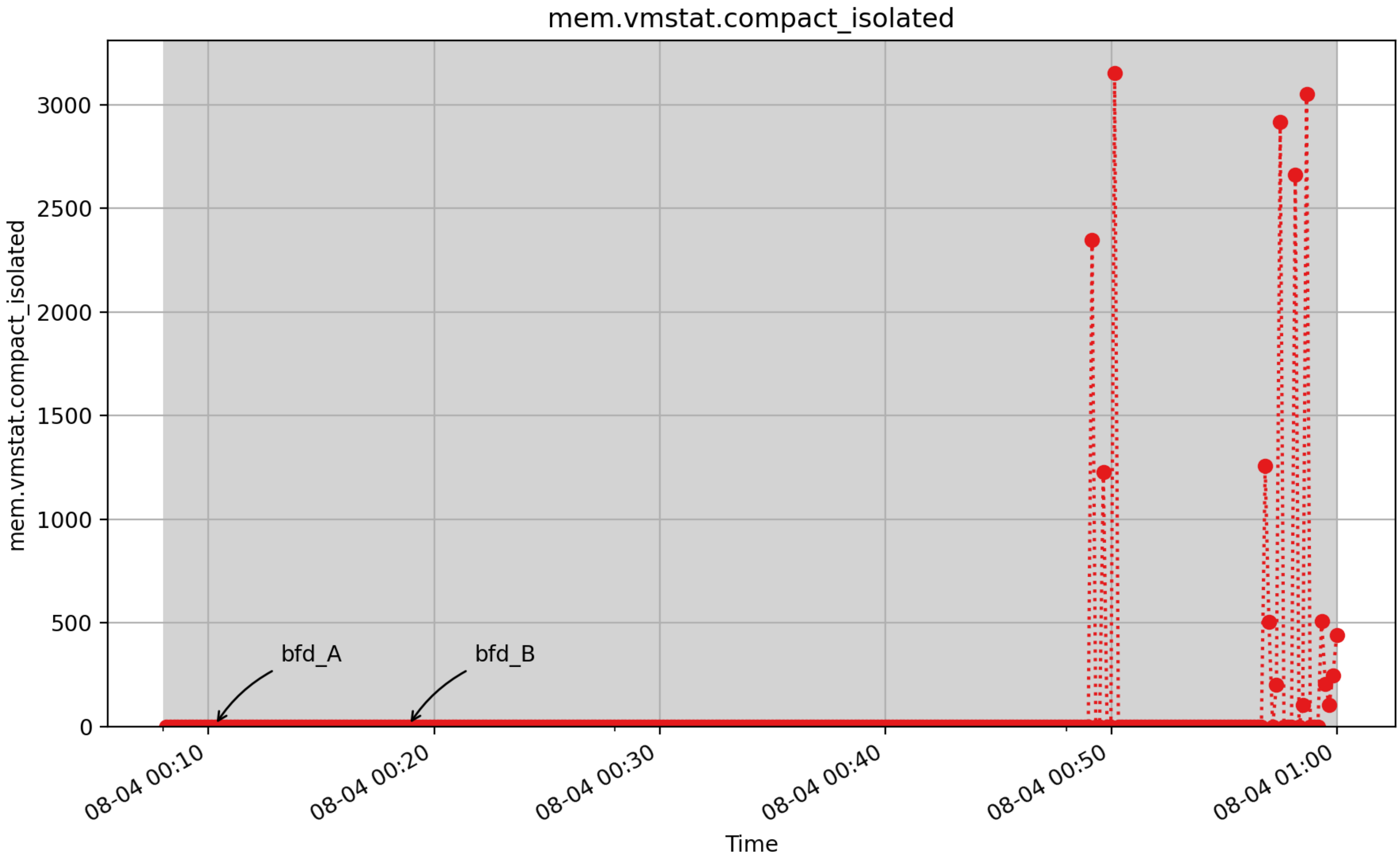
mem.util.vmallocTotal: amount of kernel memory allocated via vmalloc (Kbyte - U64)



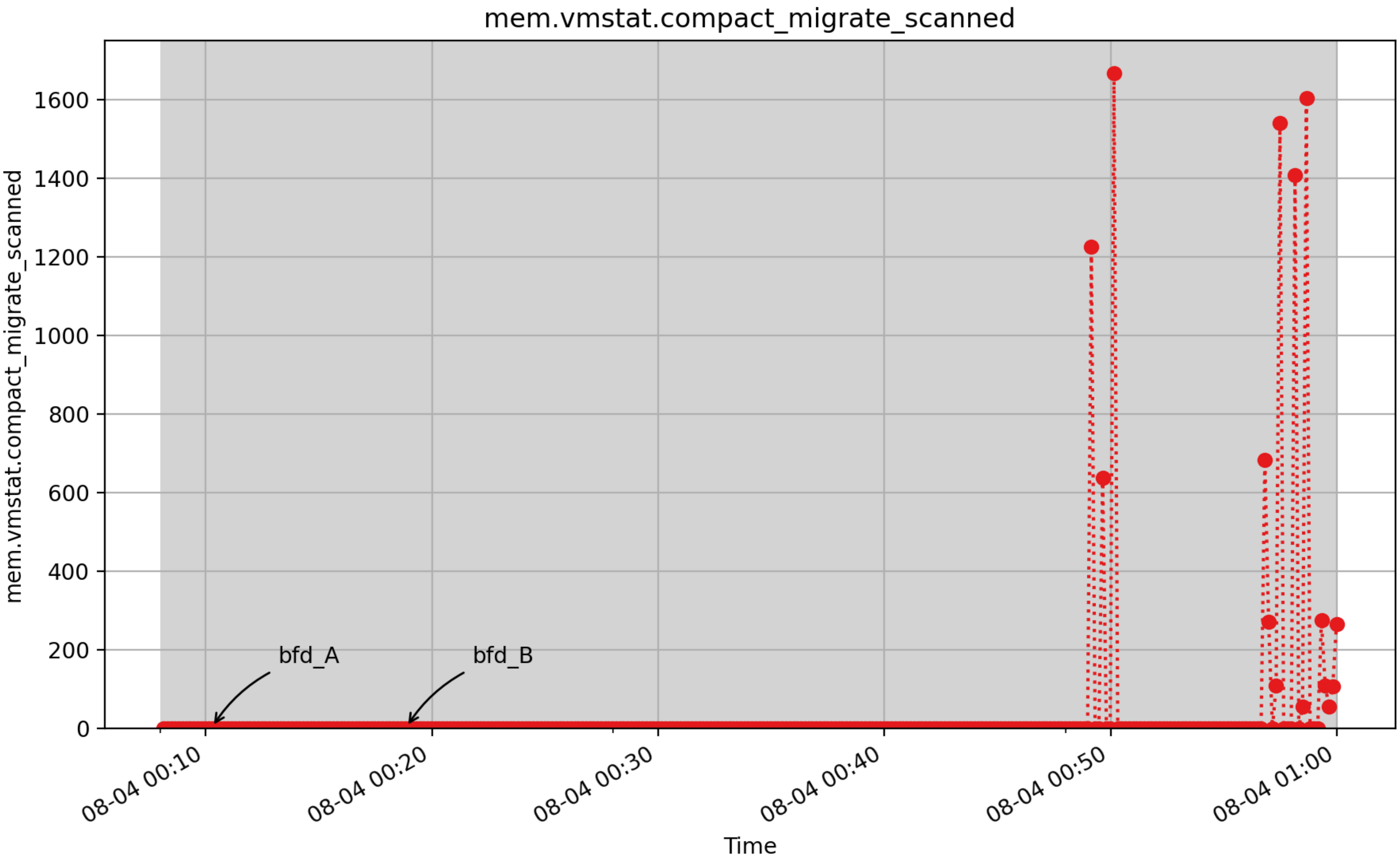
mem.util.writeback: Memory which is actively being written back to the disk (Kbyte - U64)



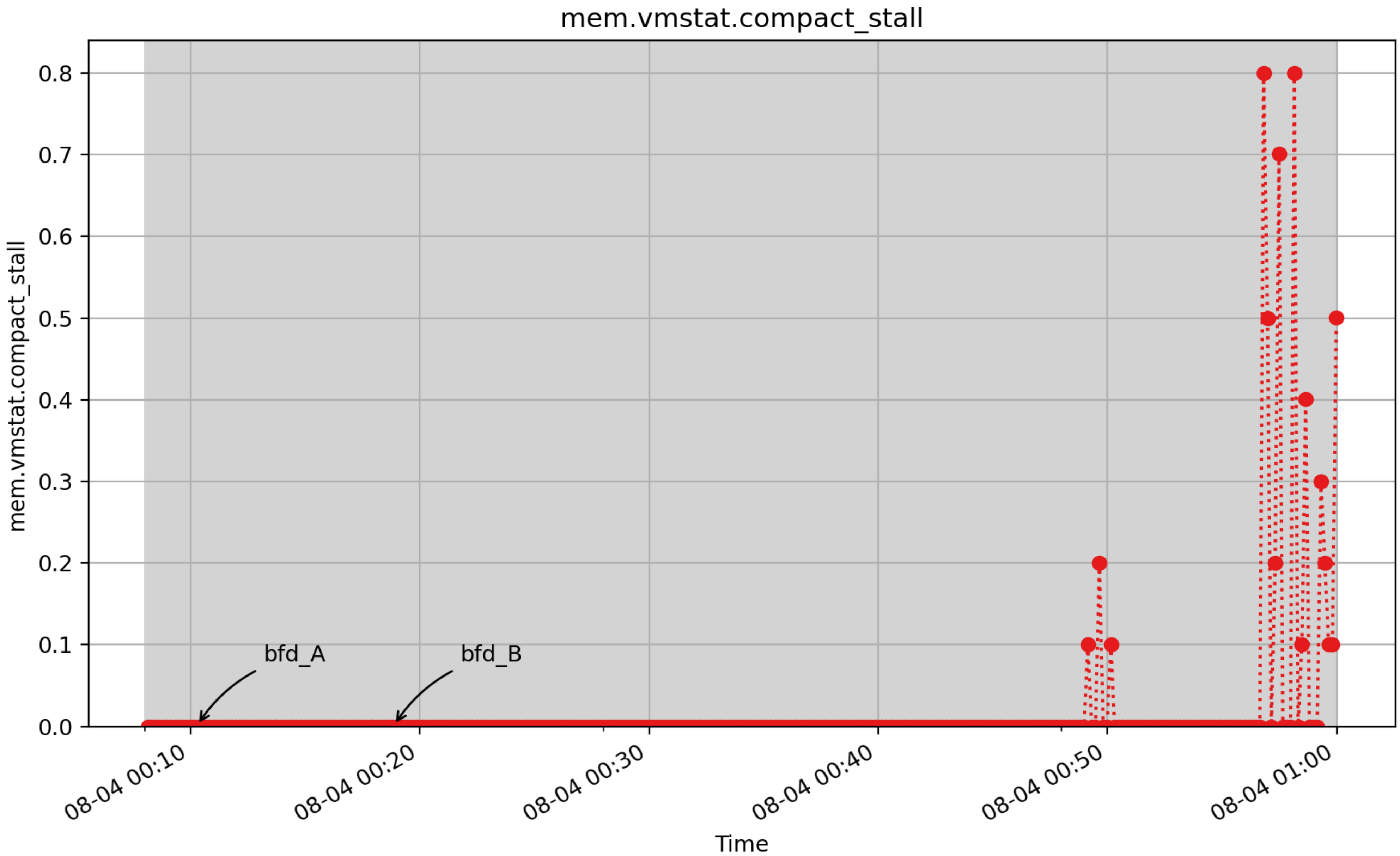
mem.vmstat.compact_free_scanned: count of pages scanned for freeing (count - U64) - *rate converted*



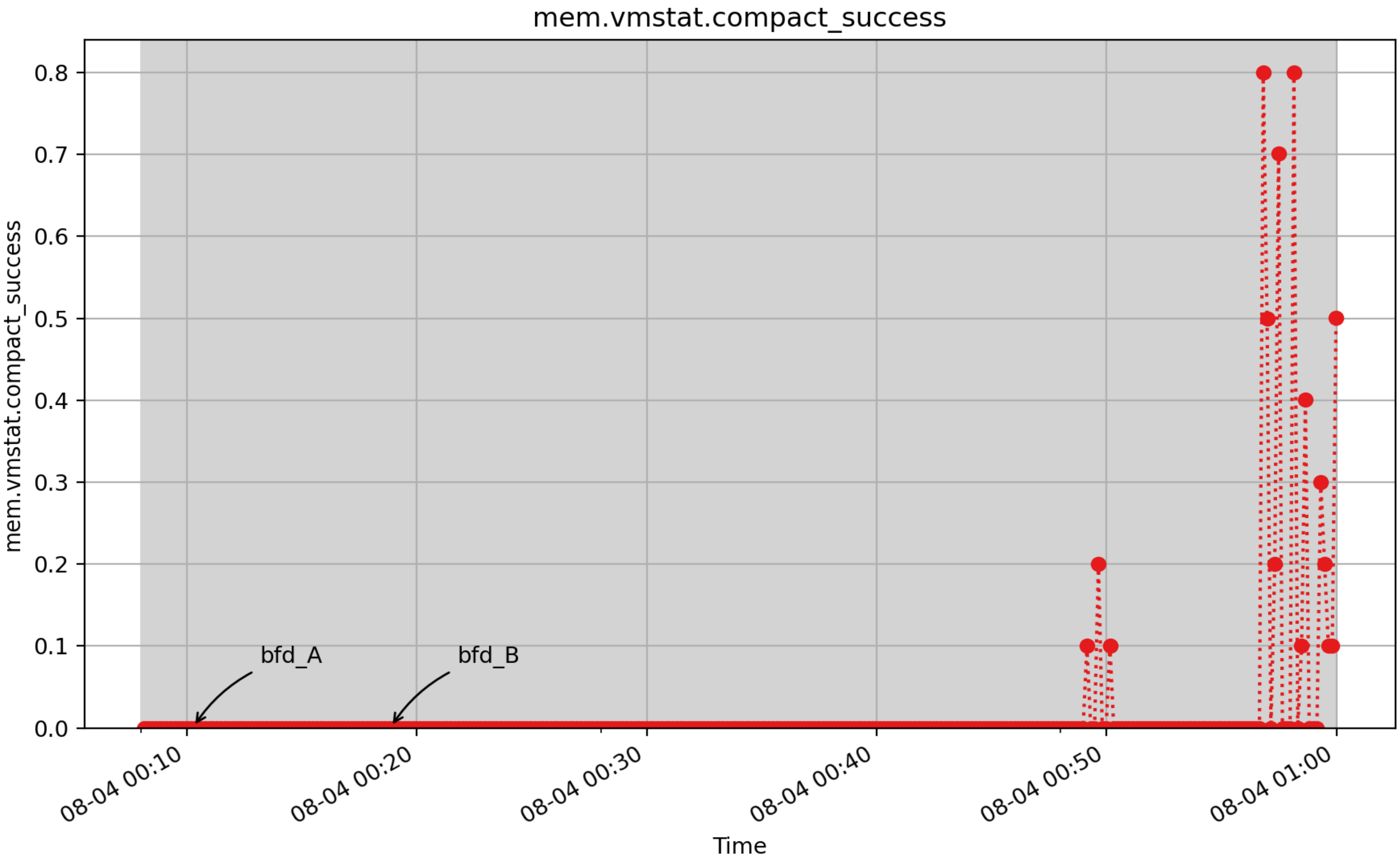
mem.vmstat.compact_isolated: count of isolated compaction pages (count - U64) - *rate converted*



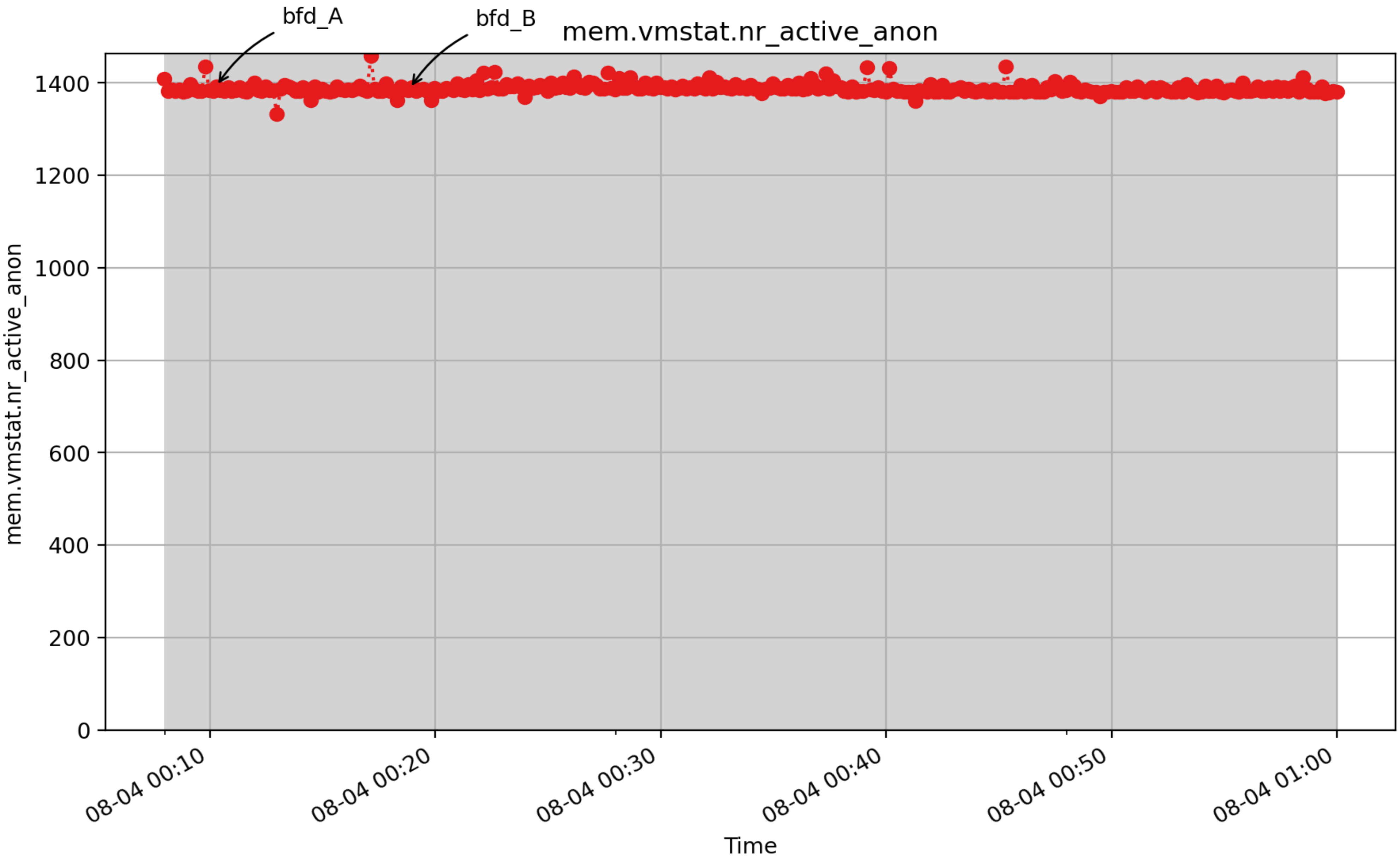
mem.vmstat.compact_migrate_scanned: count of pages scanned for migration (count - U64) - *rate converted*



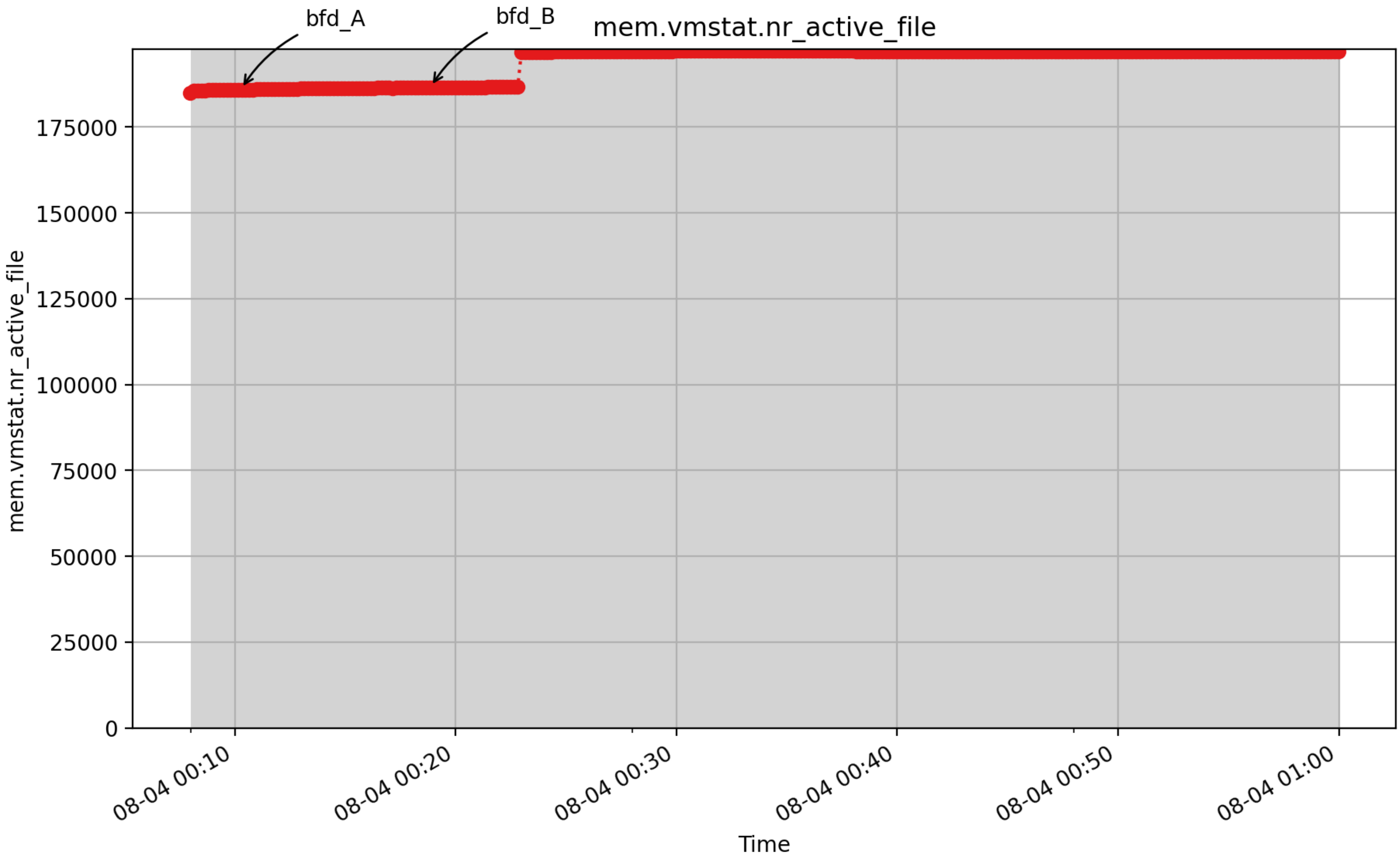
mem.vmstat.compact_stall: count of failures to even start compacting (count - U64) - *rate converted*



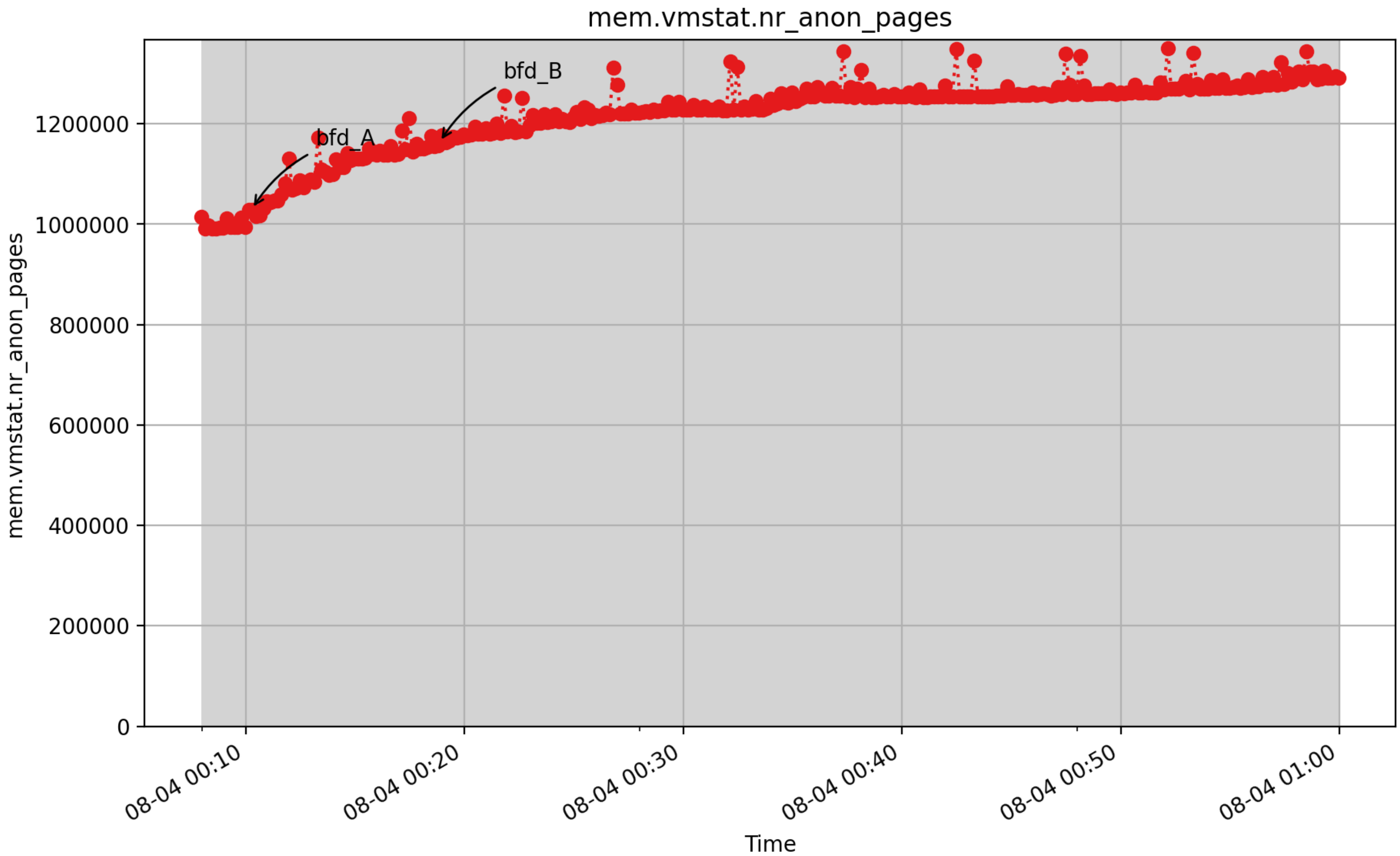
mem.vmstat.compact_success: count of successful compactions for high order allocations (count - U64) - *rate converted*



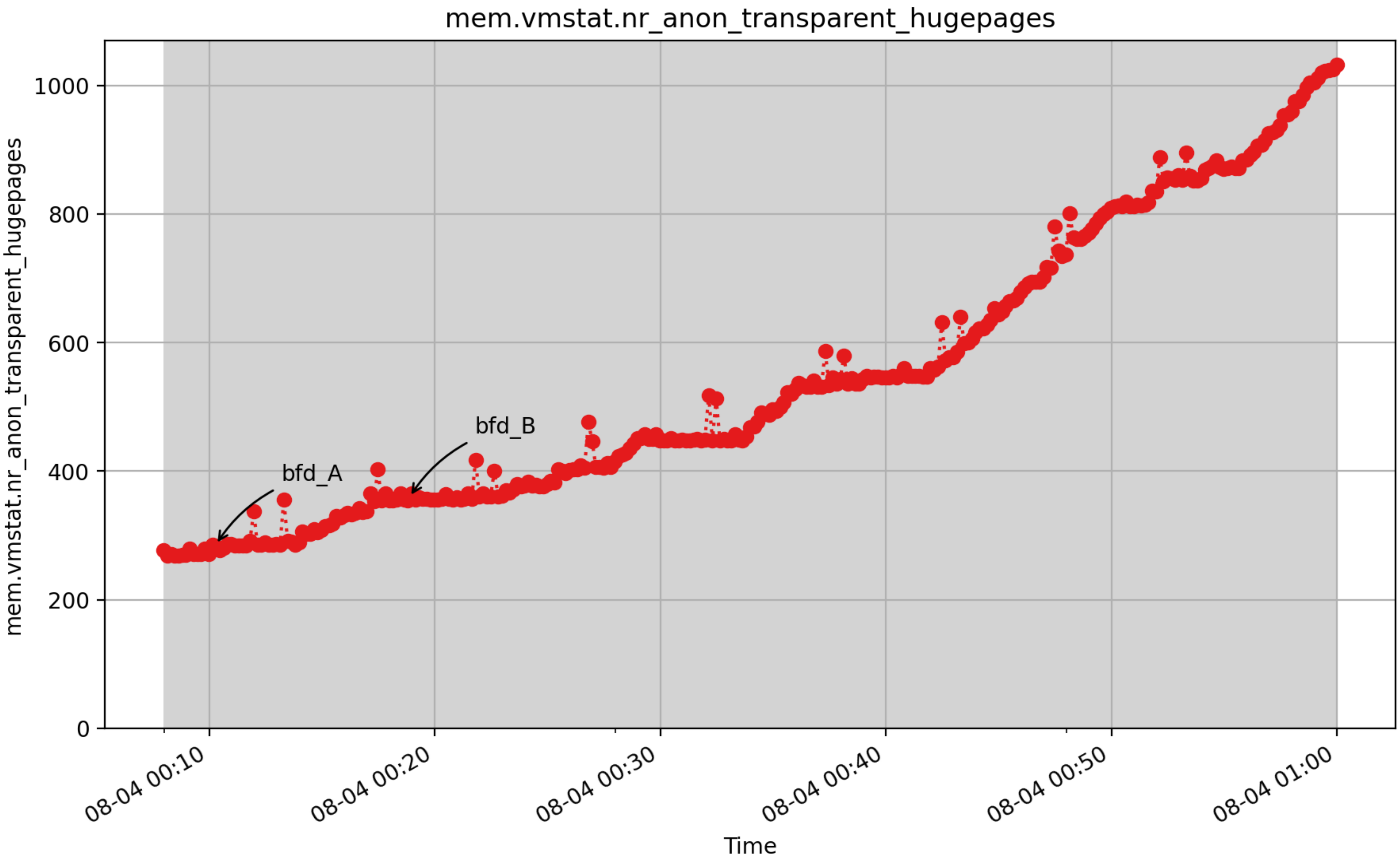
mem.vmstat.nr_active_anon: number of active anonymous memory pages (count - U64)



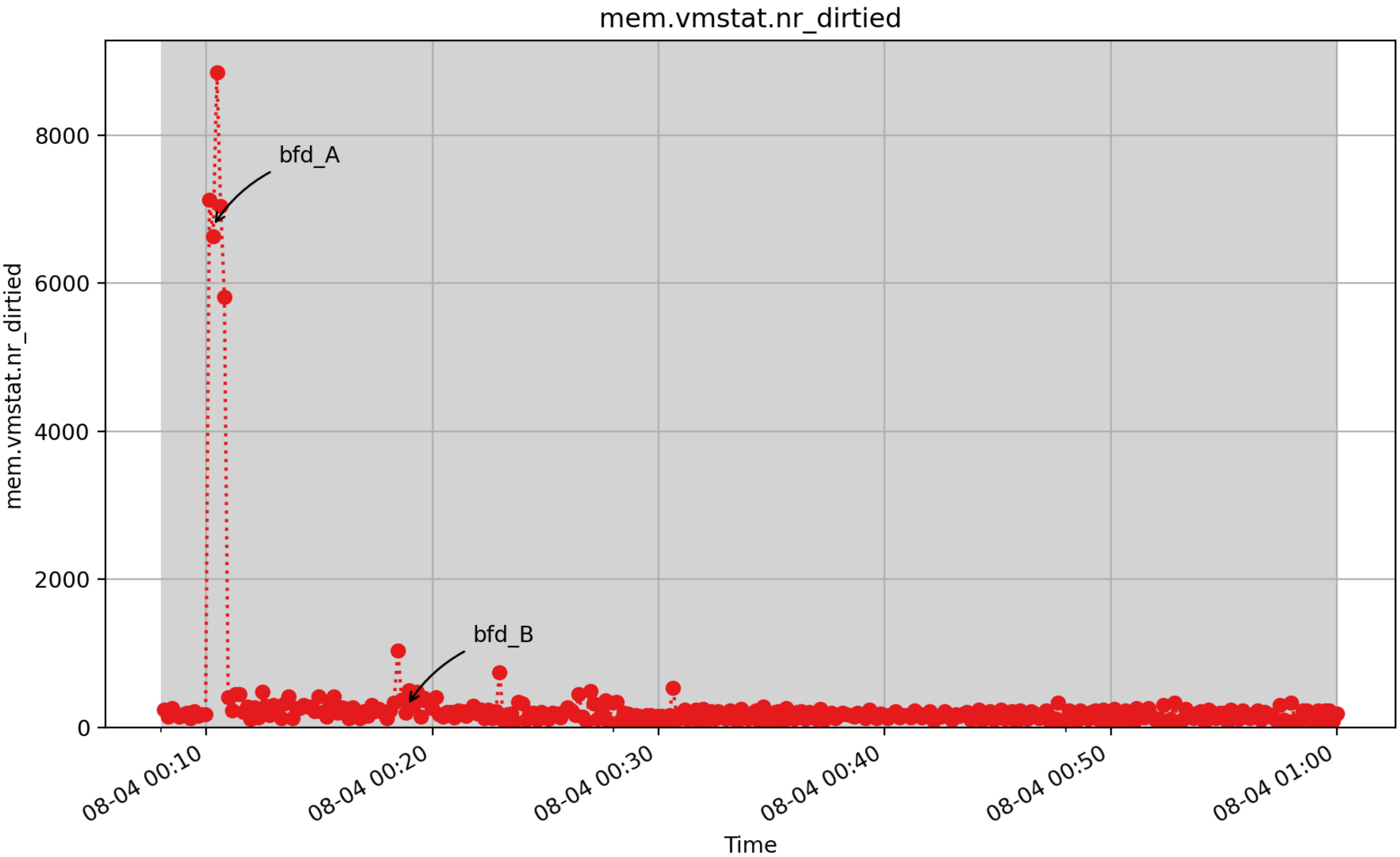
mem.vmstat.nr_active_file: number of active file memory memory pages (count - U64)



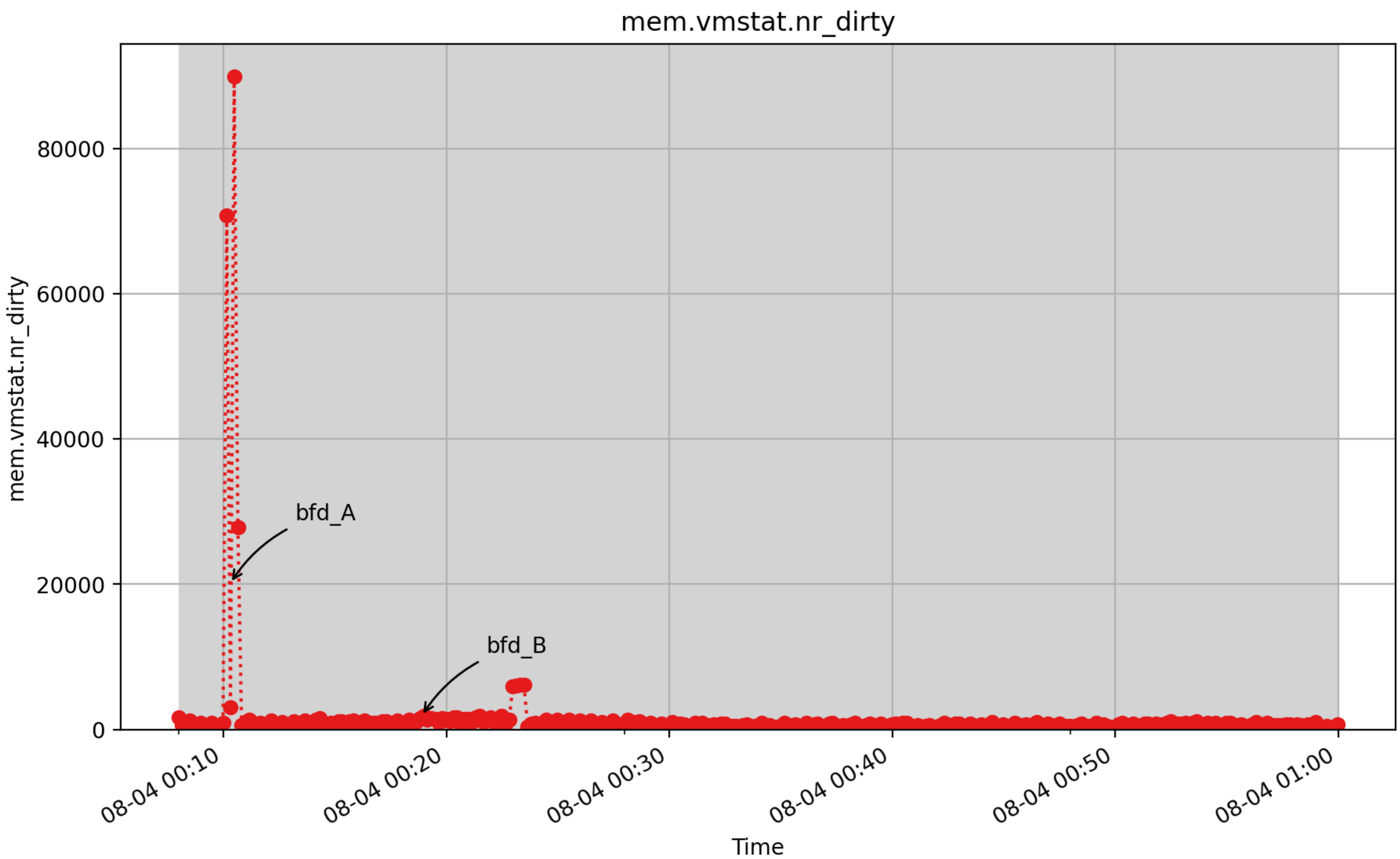
mem.vmstat.nr_anon_pages: Instantaneous number of anonymous mapped pagecache pages, from /proc/vmstat See also mem.vmstat.mapped for other mapped pages. (- U64)



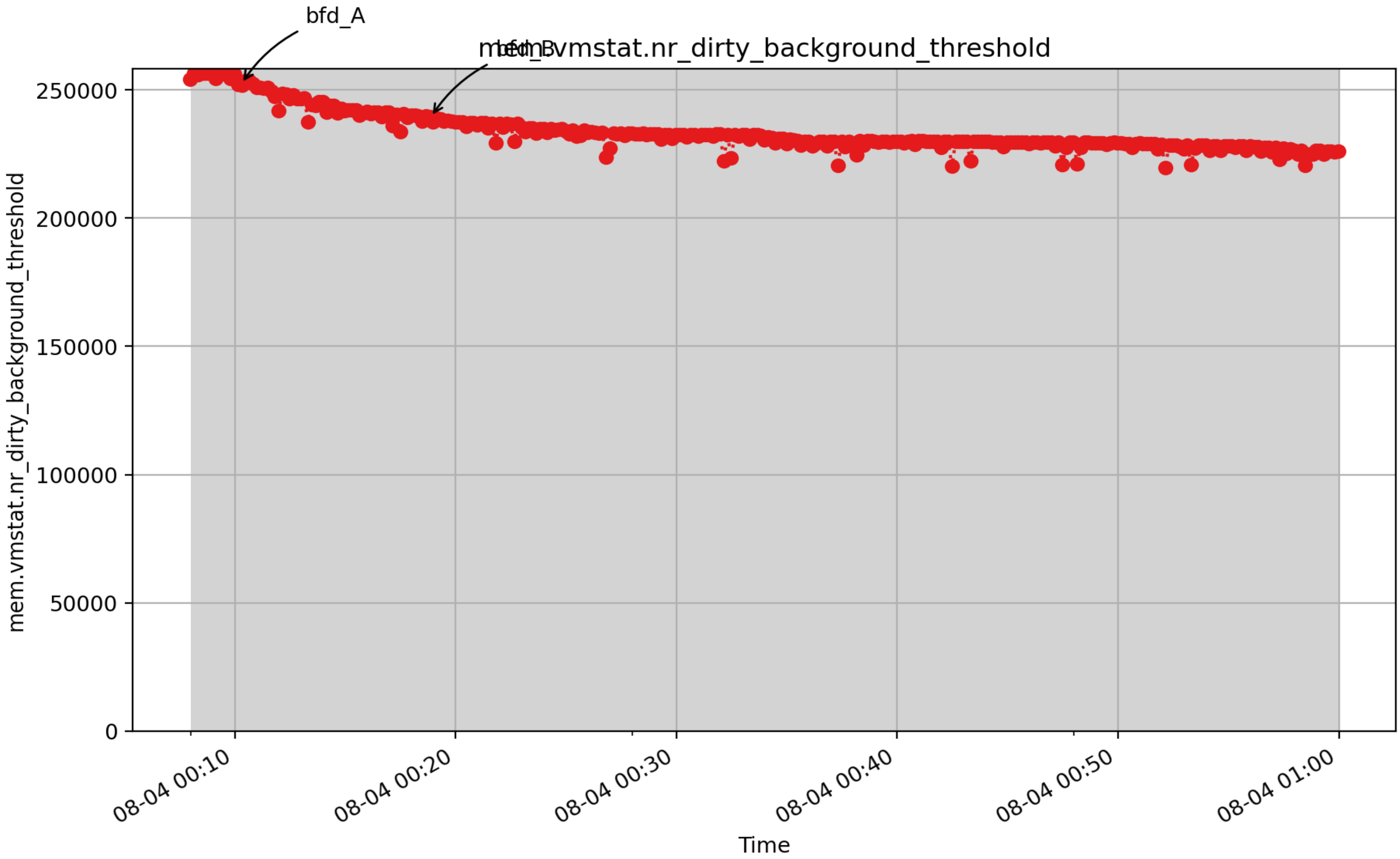
mem.vmstat.nr_anon_transparent_hugepages: Instantaneous number of anonymous transparent huge pages, from /proc/vmstat (count - U64)



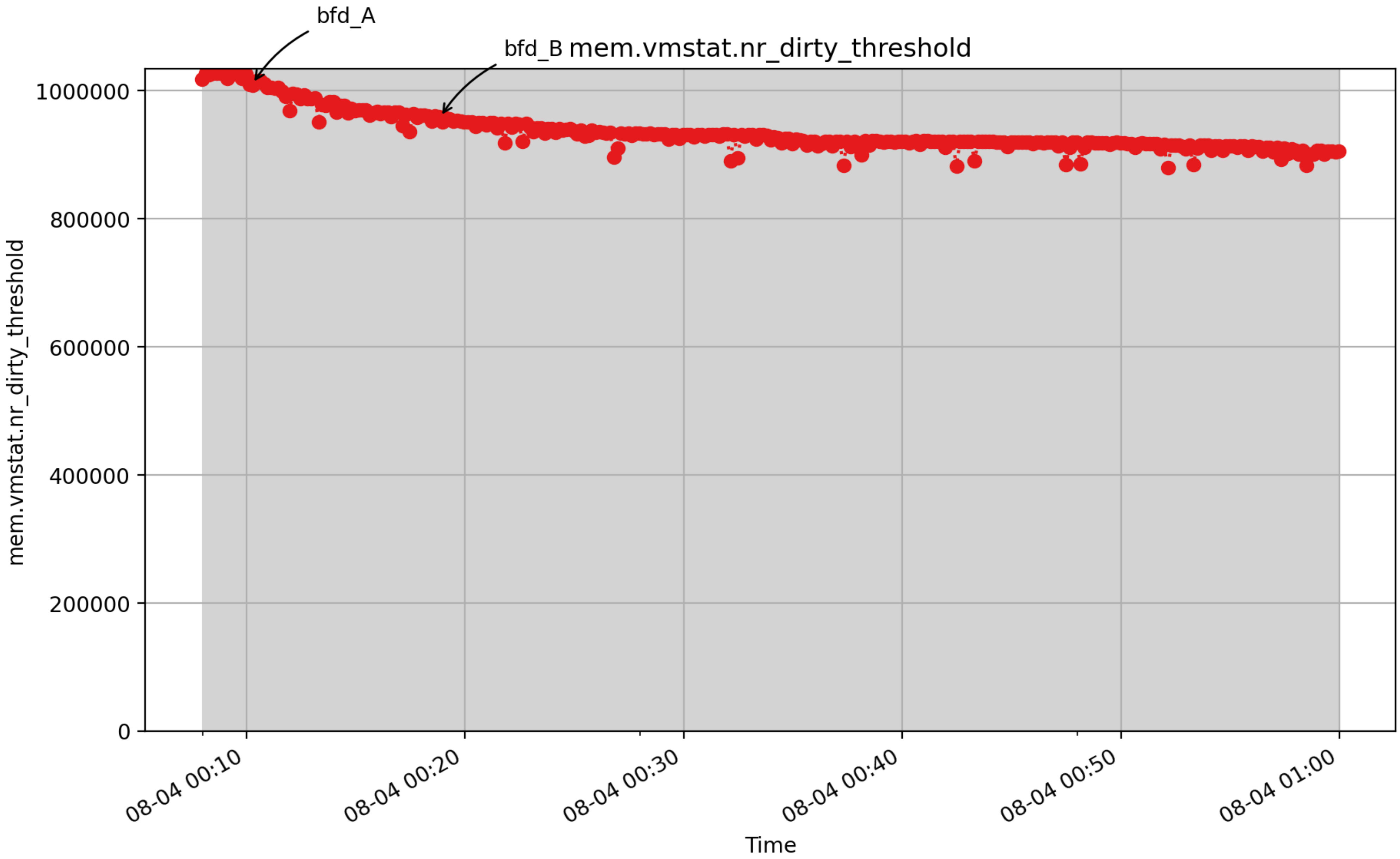
mem.vmstat.nr_dirtied: Count of pages entering dirty state, from /proc/vmstat (count - U64) - *rate converted*



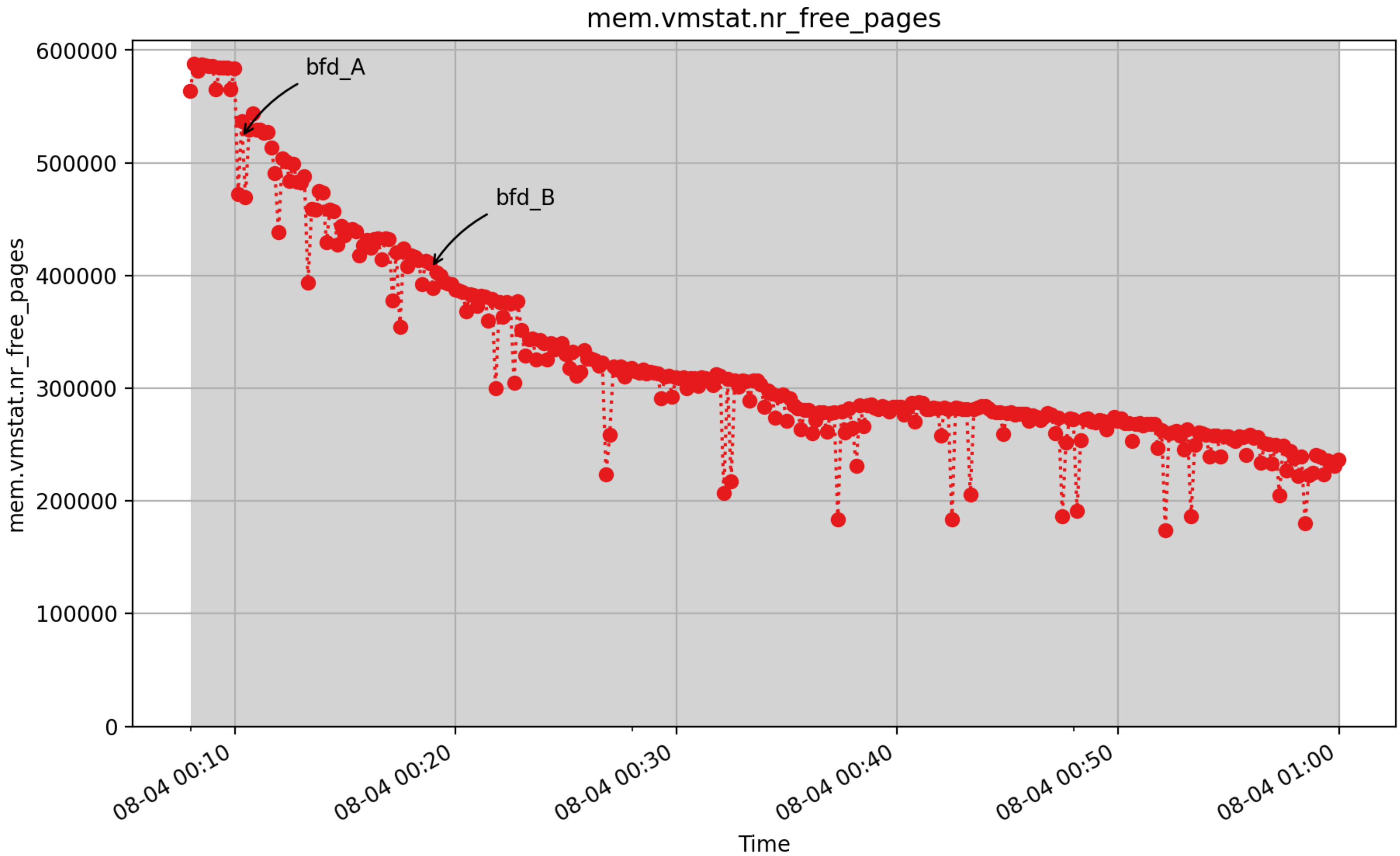
mem.vmstat.nr_dirty: Instantaneous number of pages in dirty state, from /proc/vmstat (- U64)



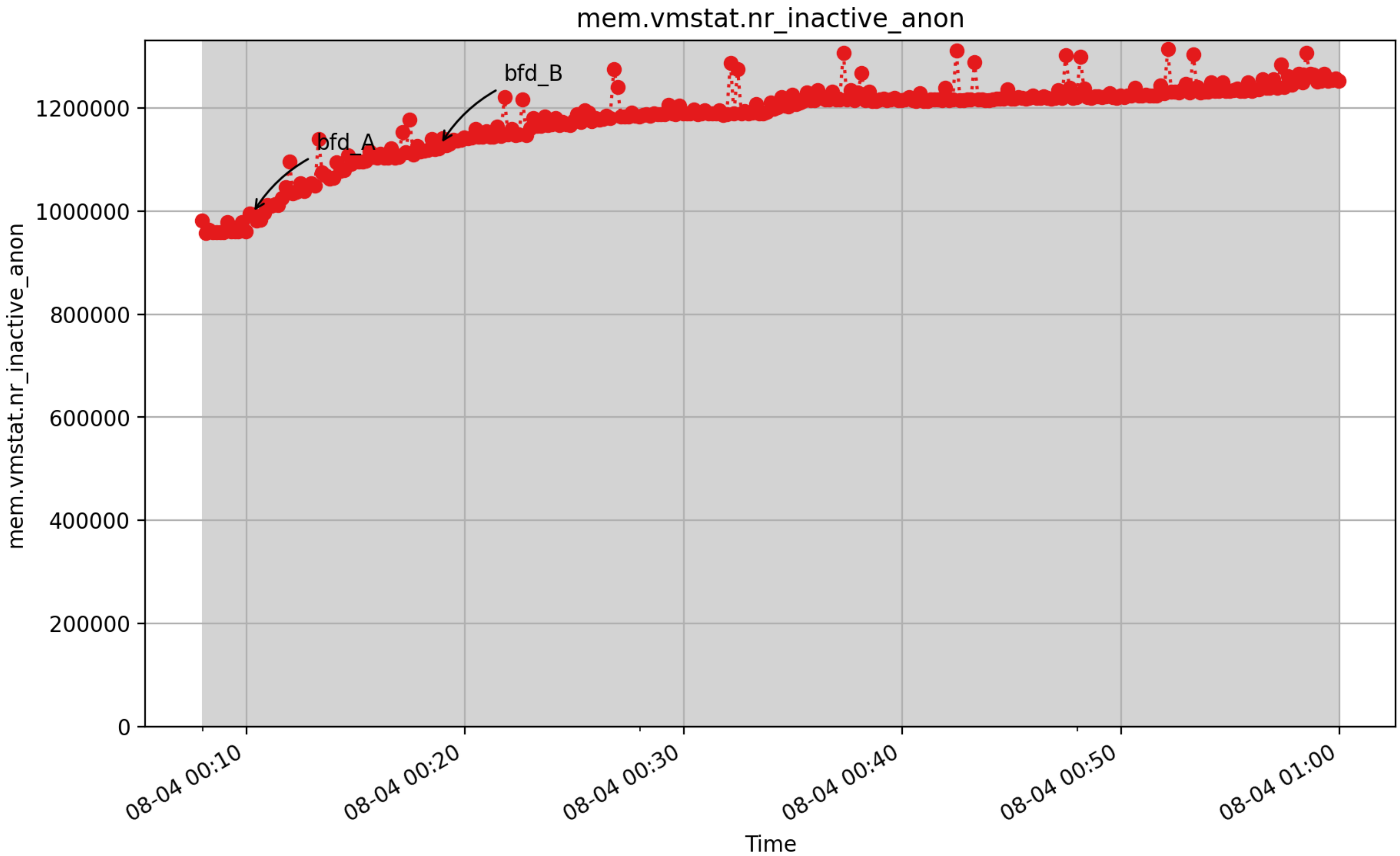
mem.vmstat.nr_dirty_background_threshold: background writeback threshold (count - U64)



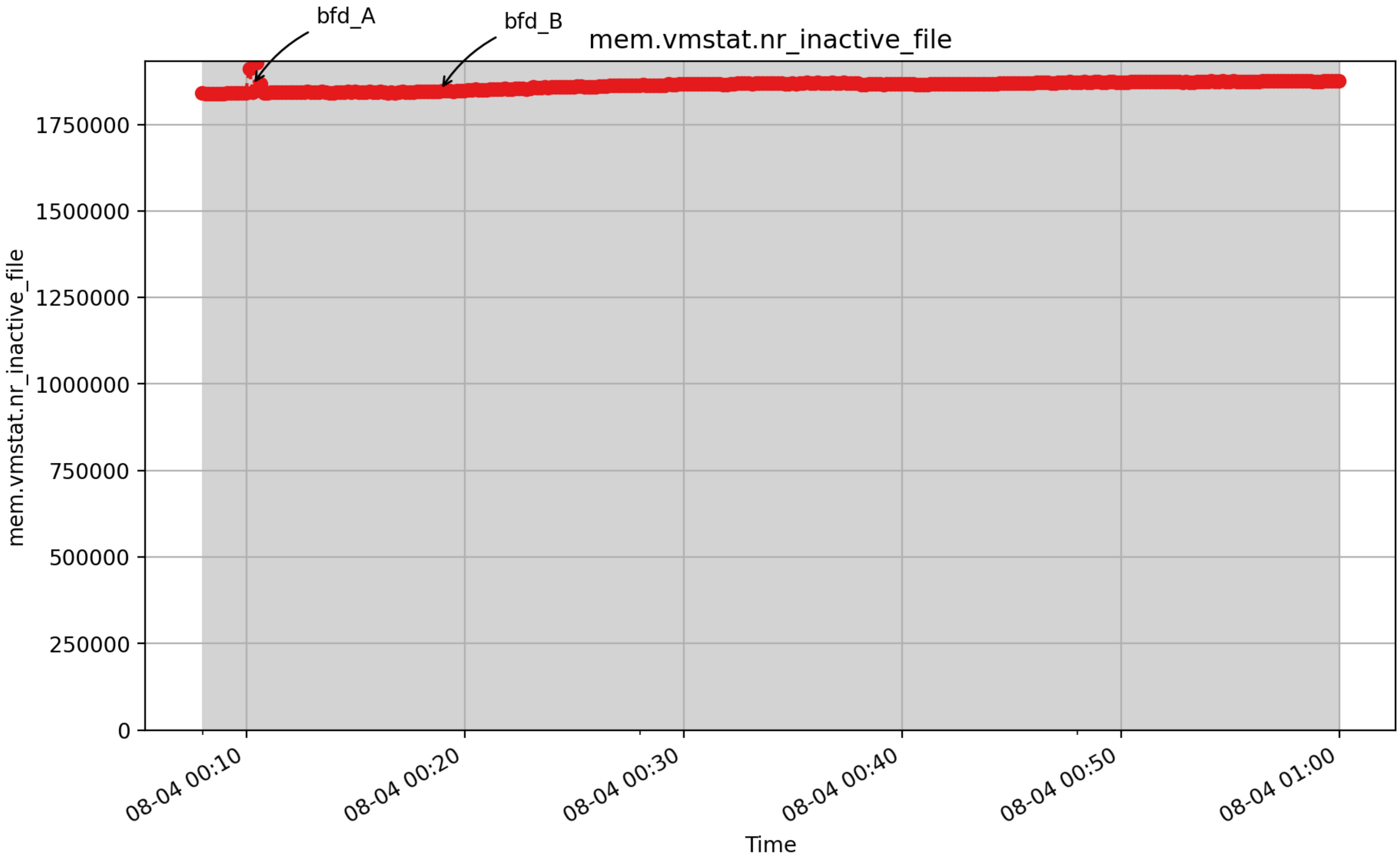
mem.vmstat.nr_dirty_threshold: dirty throttling threshold (count - U64)



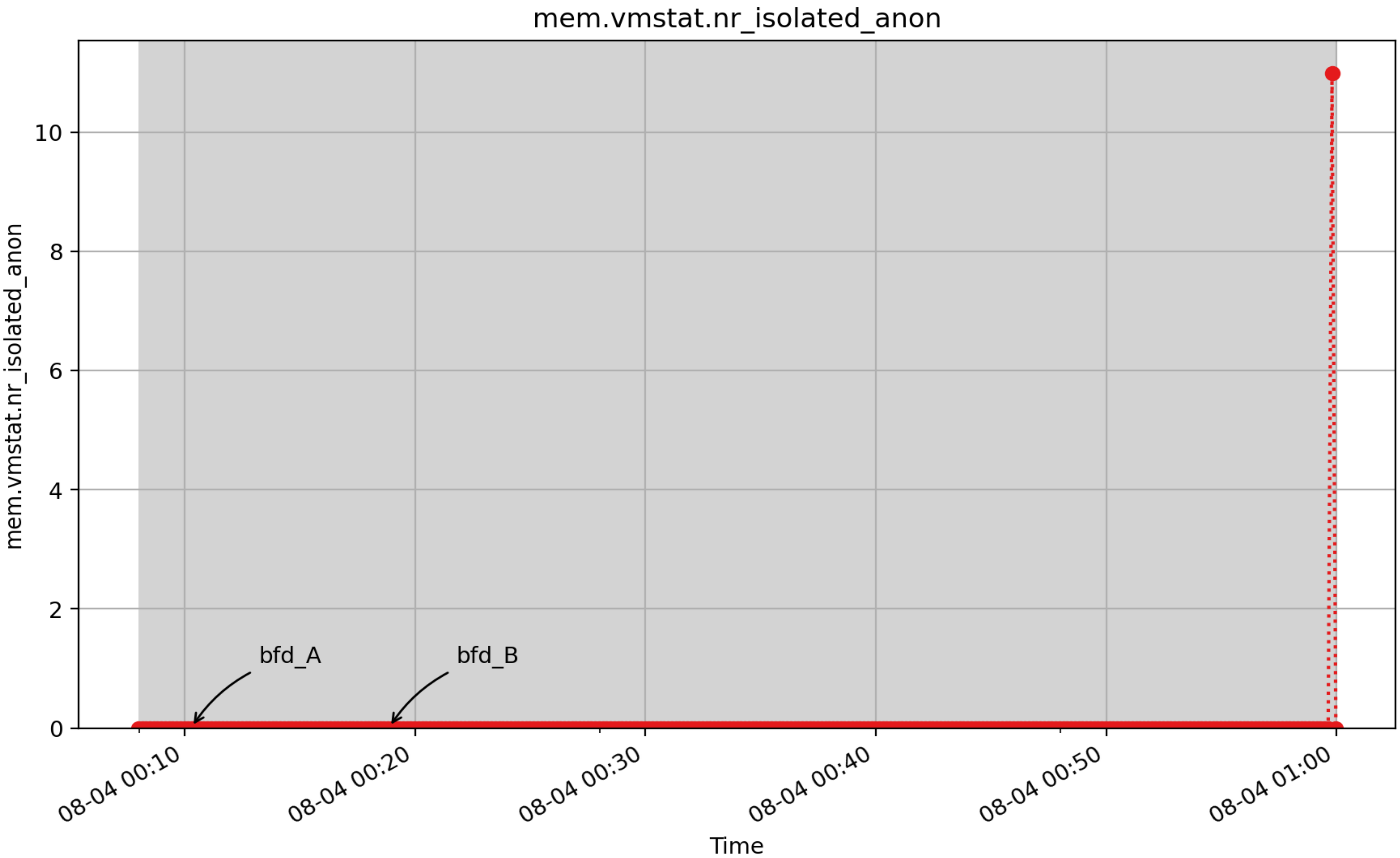
mem.vmstat.nr_free_pages: number of free pages (count - U64)



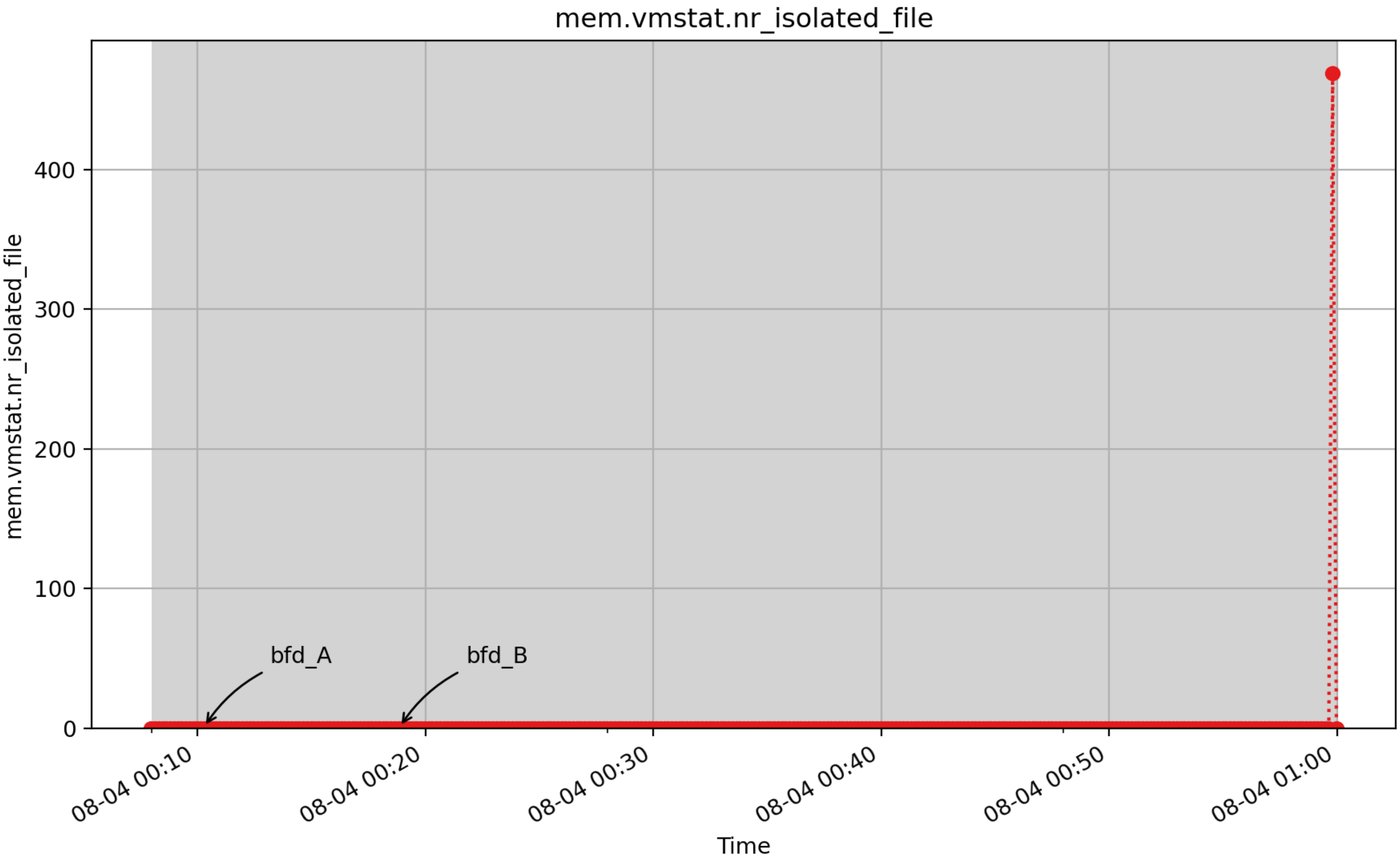
mem.vmstat.nr_inactive_anon: number of inactive anonymous memory pages (count - U64)



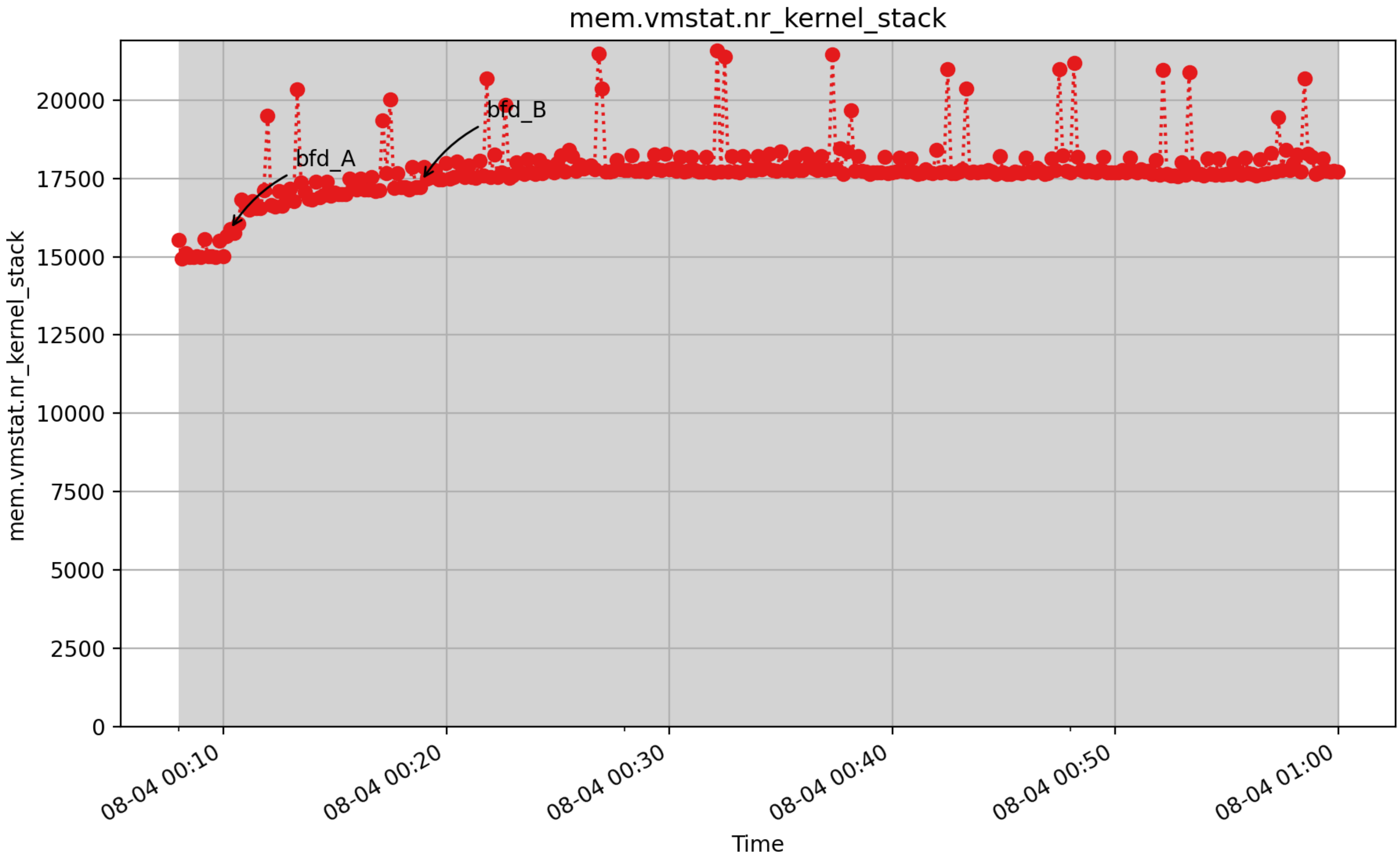
mem.vmstat.nr_inactive_file: number of inactive file memory pages (count - U64)



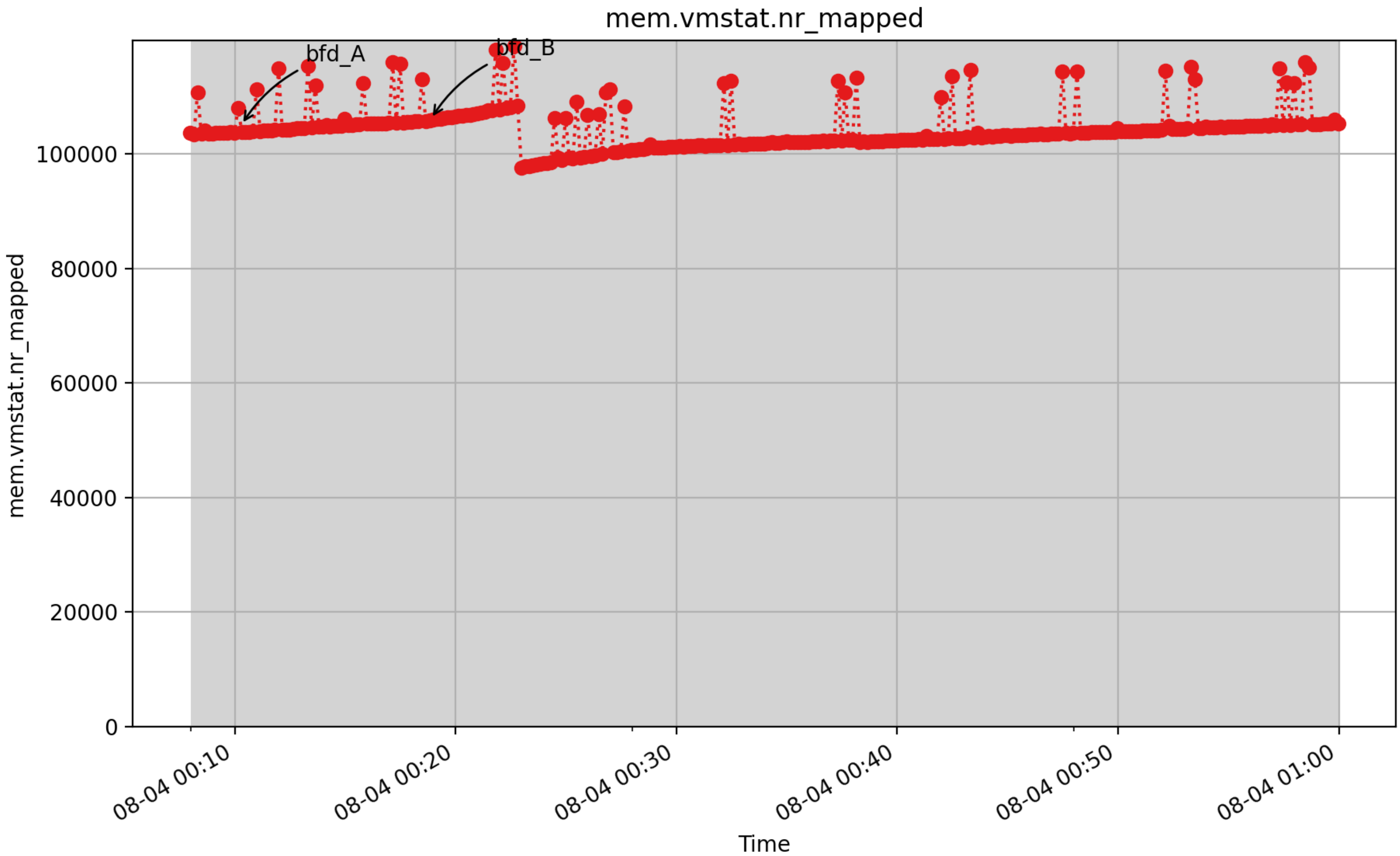
mem.vmstat.nr_isolated_anon: number of isolated anonymous memory pages (count - U64)



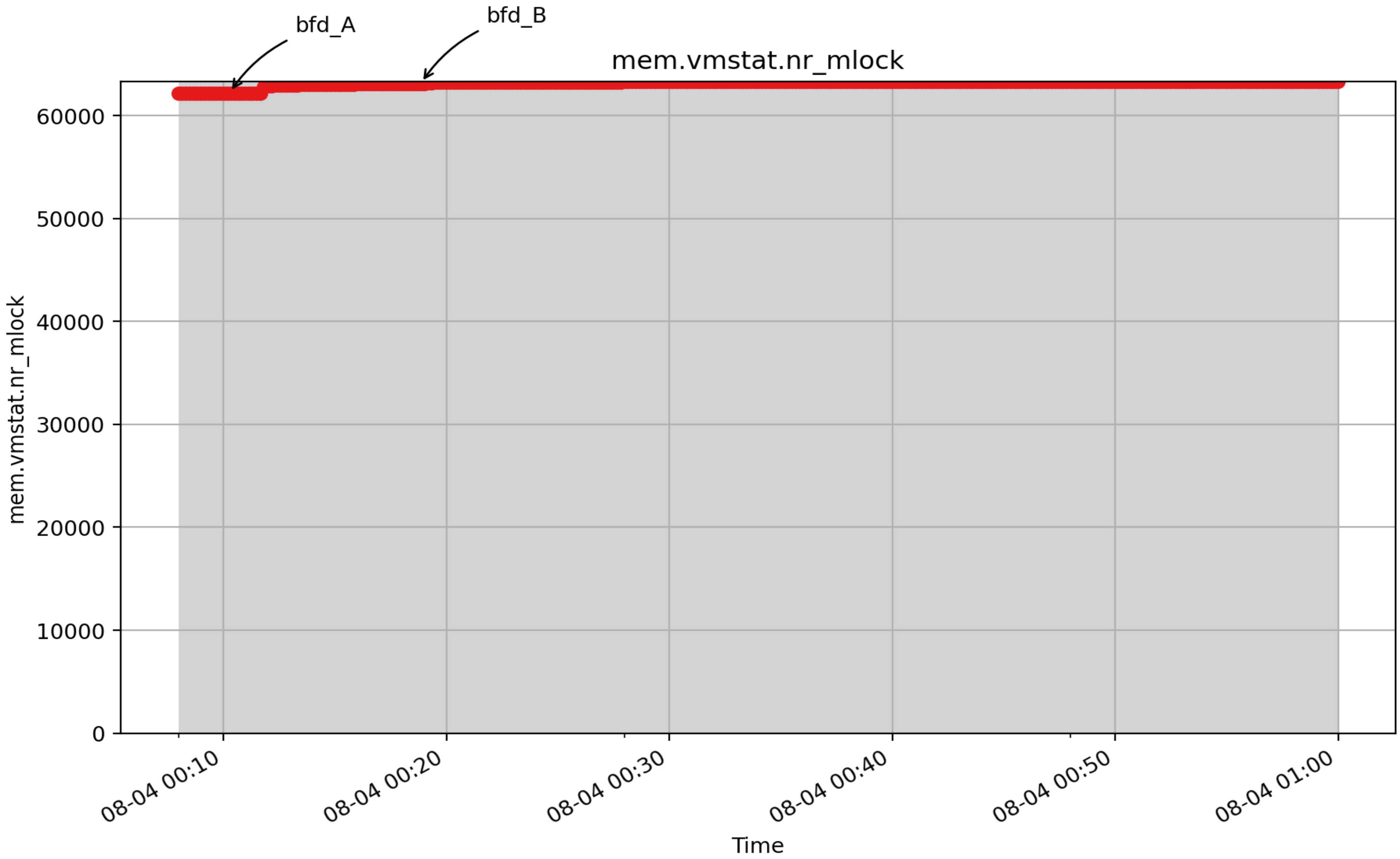
mem.vmstat.nr_isolated_file: number of isolated file memory pages (count - U64)



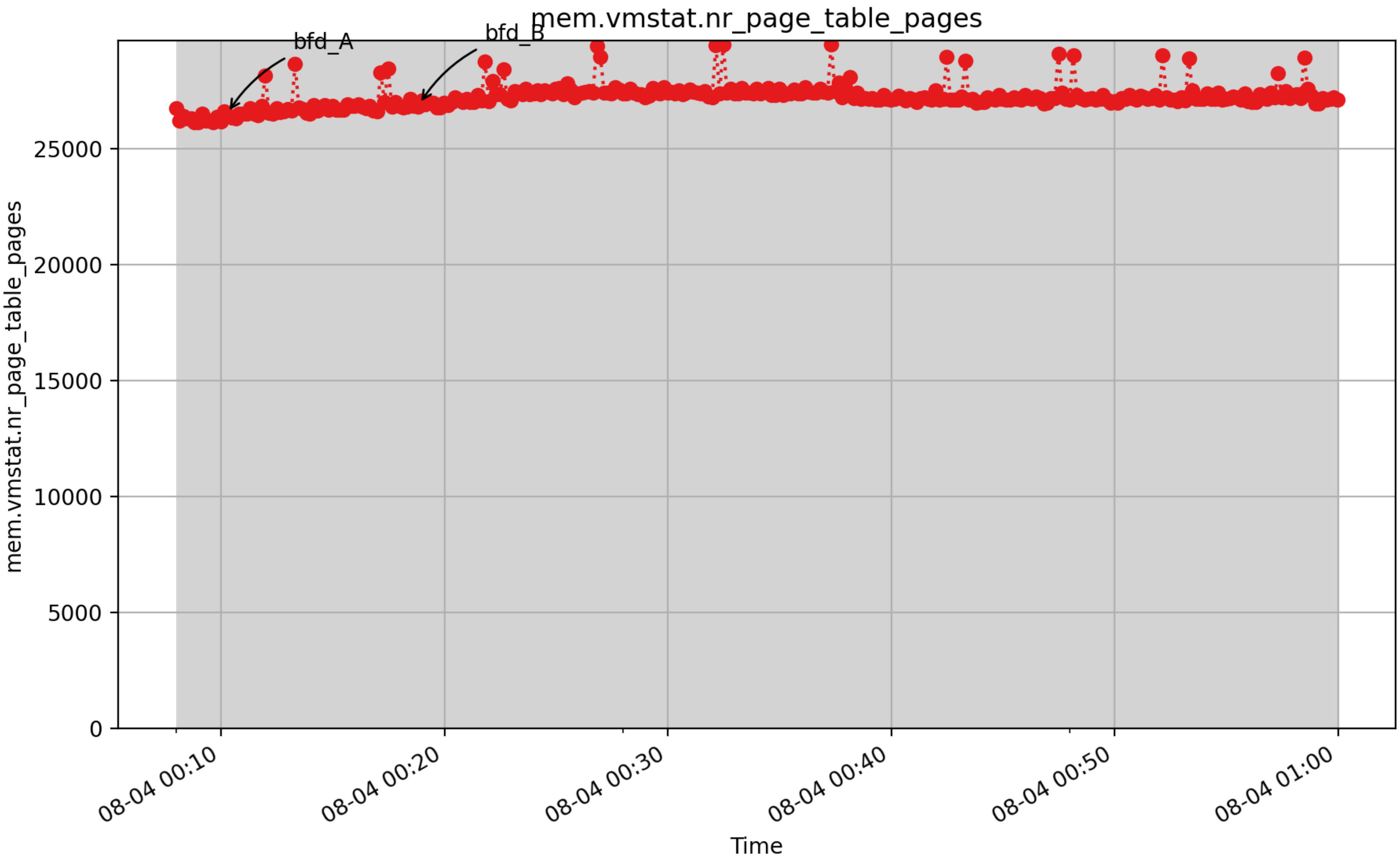
mem.vmstat.nr_kernel_stack: number of pages of kernel stack (count - U64)



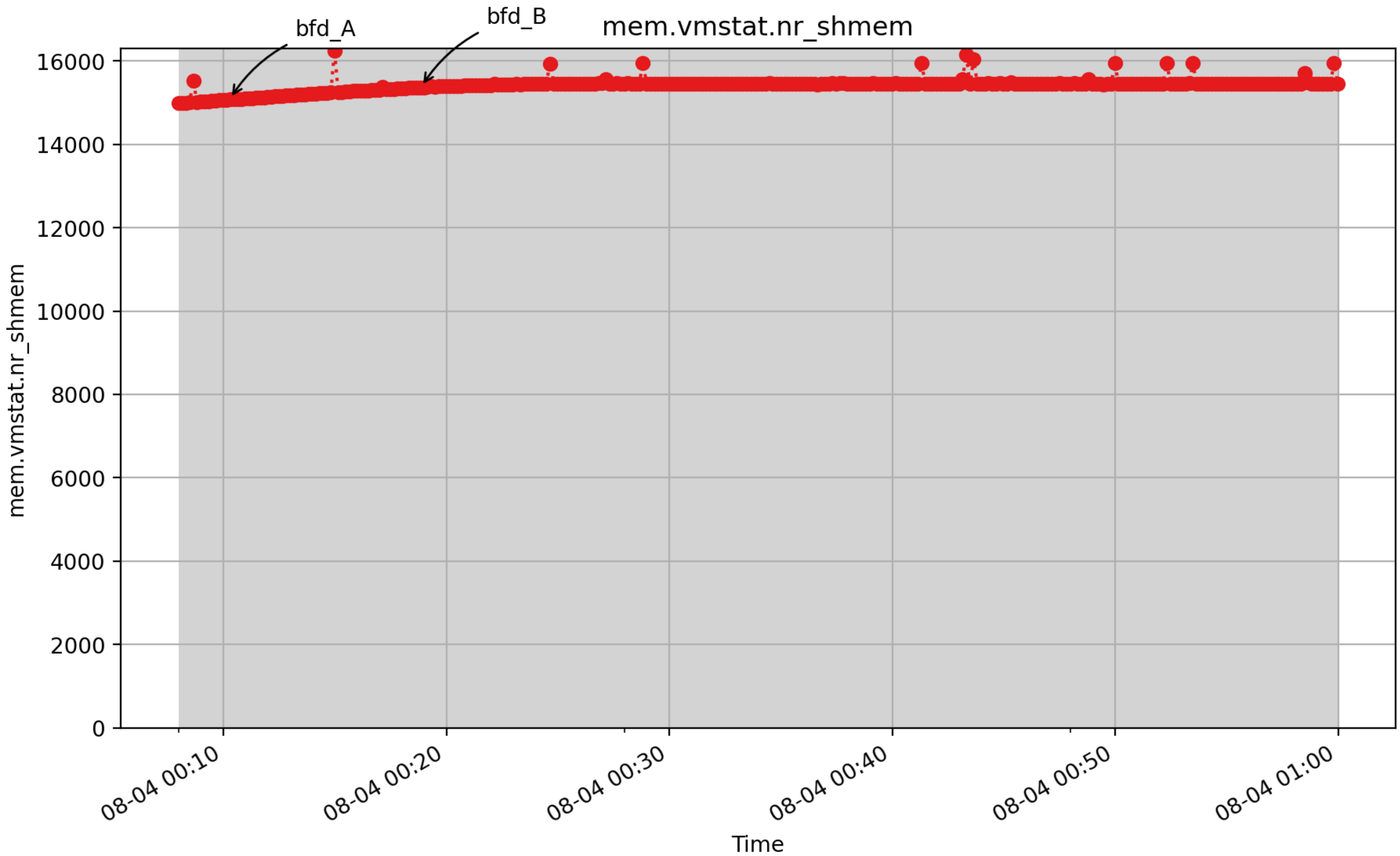
mem.vmstat.nr_mapped: Instantaneous number of mapped pagecache pages, from /proc/vmstat See also mem.vmstat.nr_anon for anonymous mapped pages. (- U64)



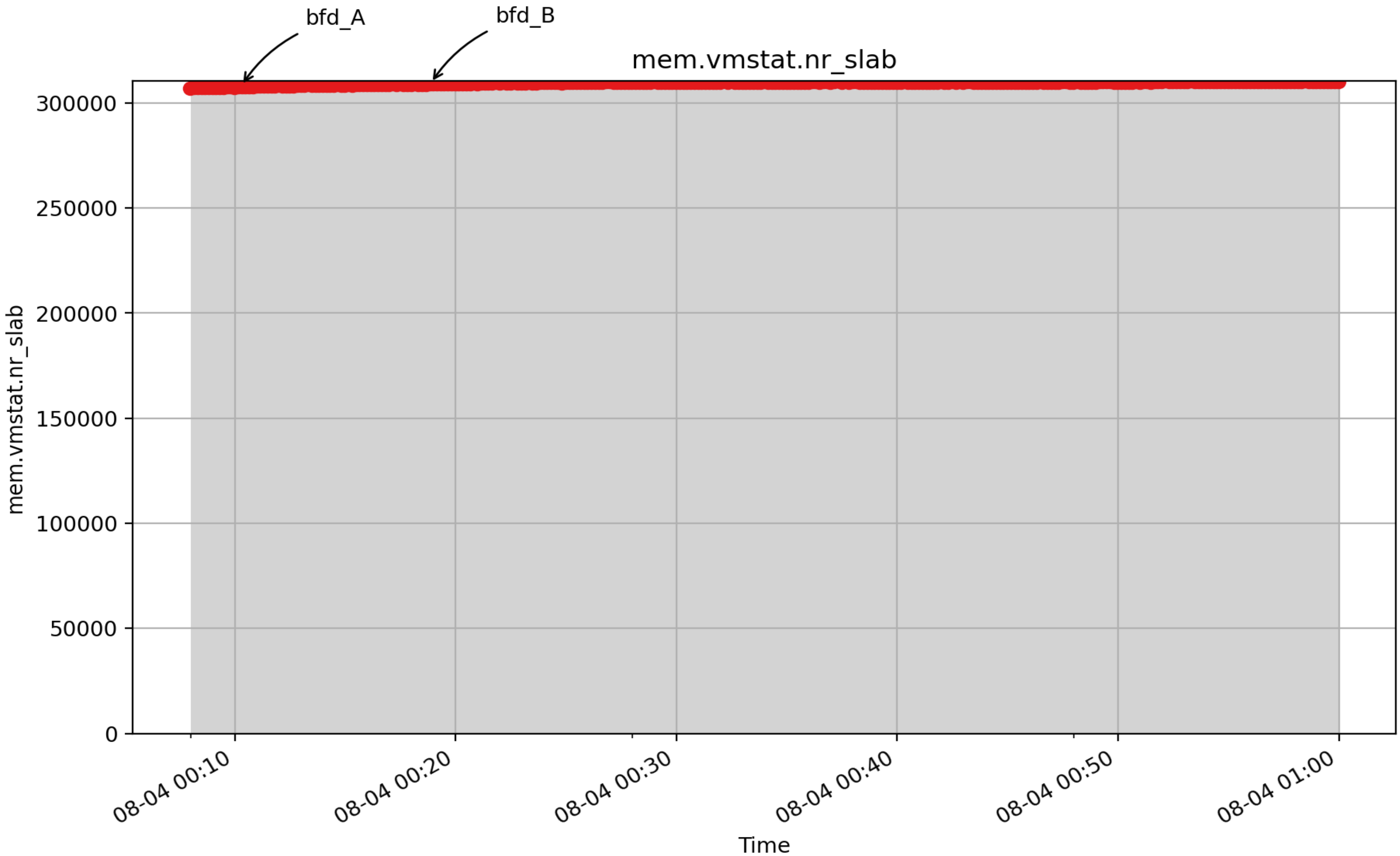
mem.vmstat.nr_mlock: number of pages under mlock (count - U64)



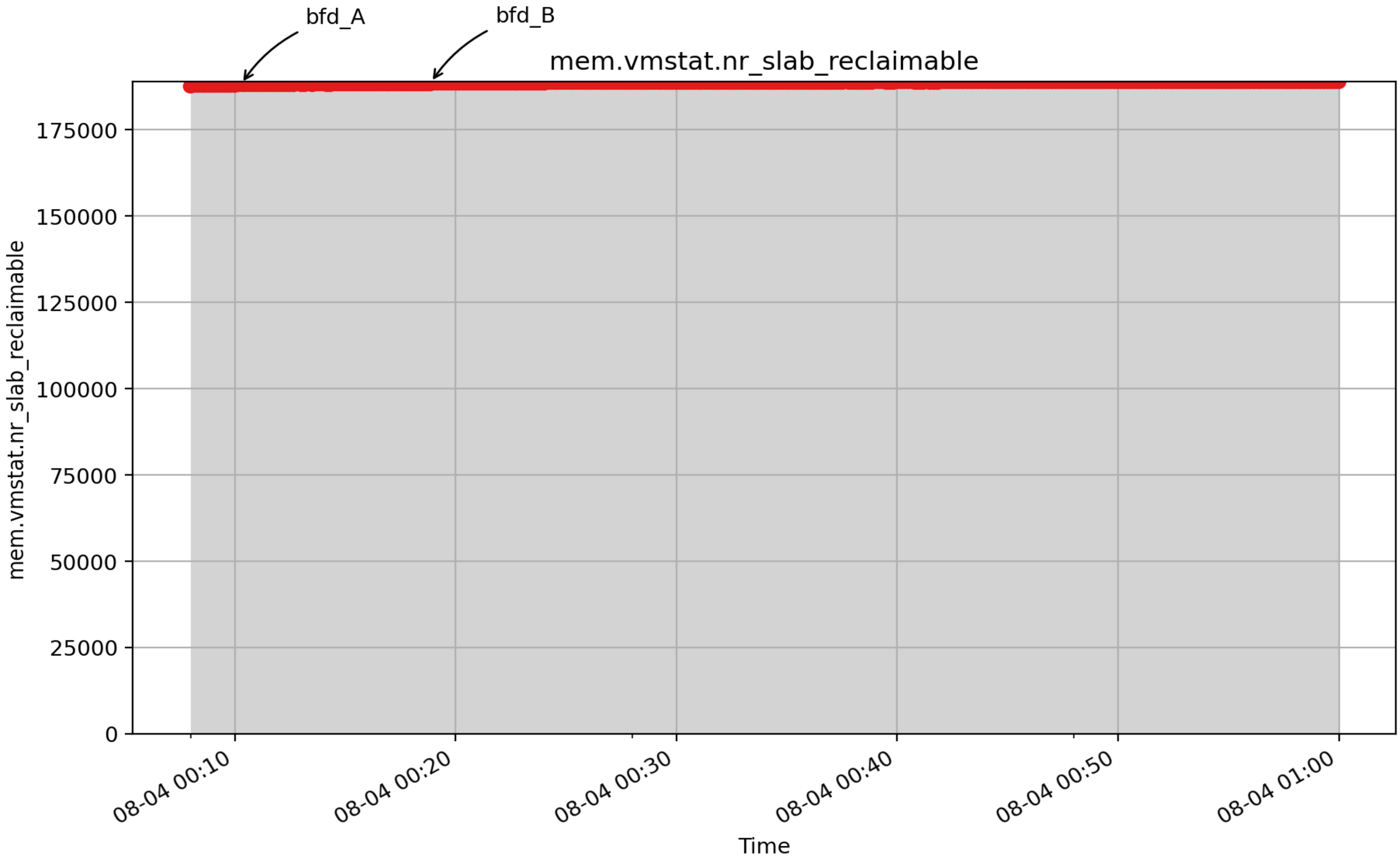
mem.vmstat.nr_page_table_pages: Instantaneous number of page table pages, from /proc/vmstat (- U64)



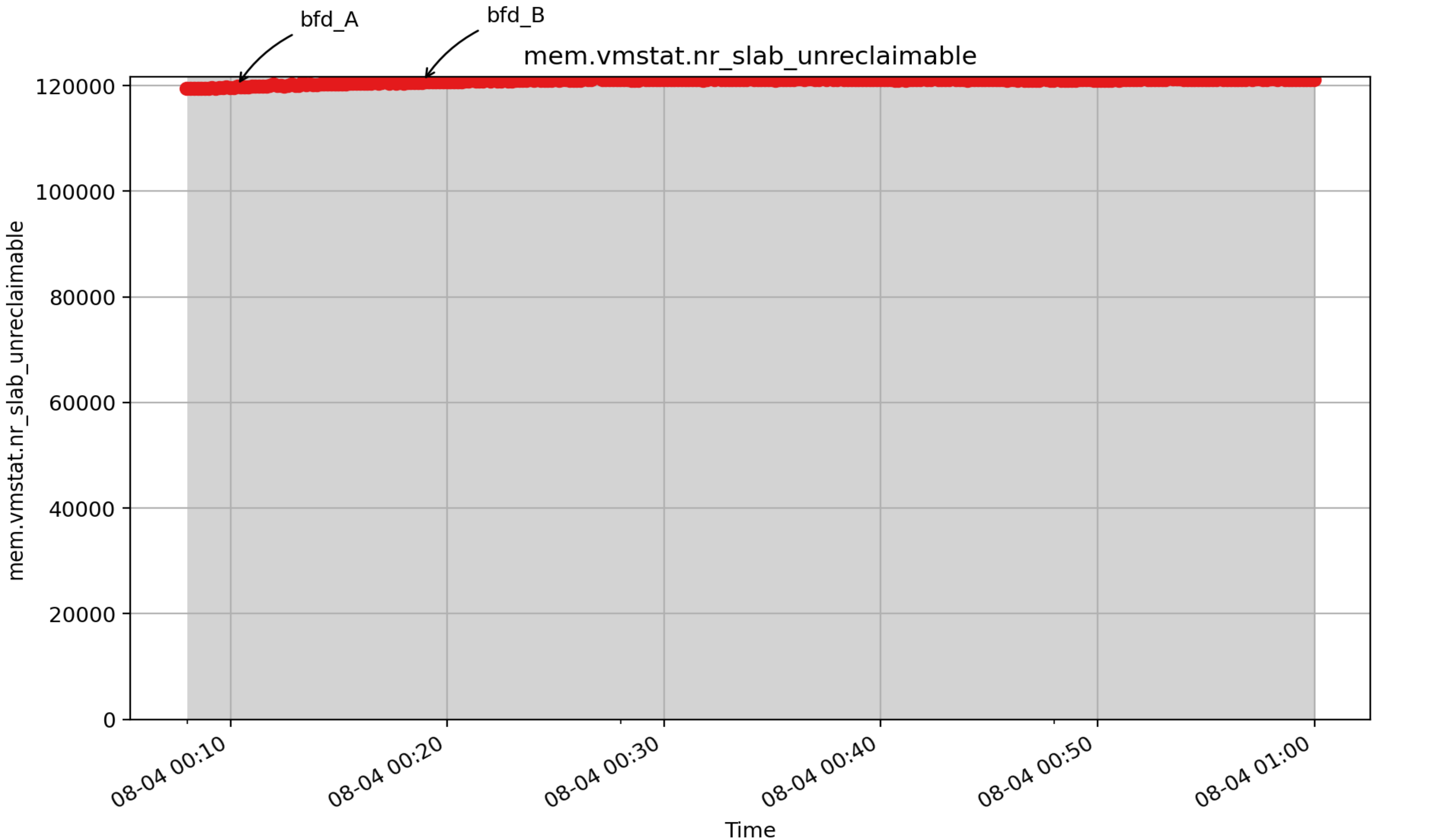
mem.vmstat.nr_shmem: number of shared memory pages (count - U64)



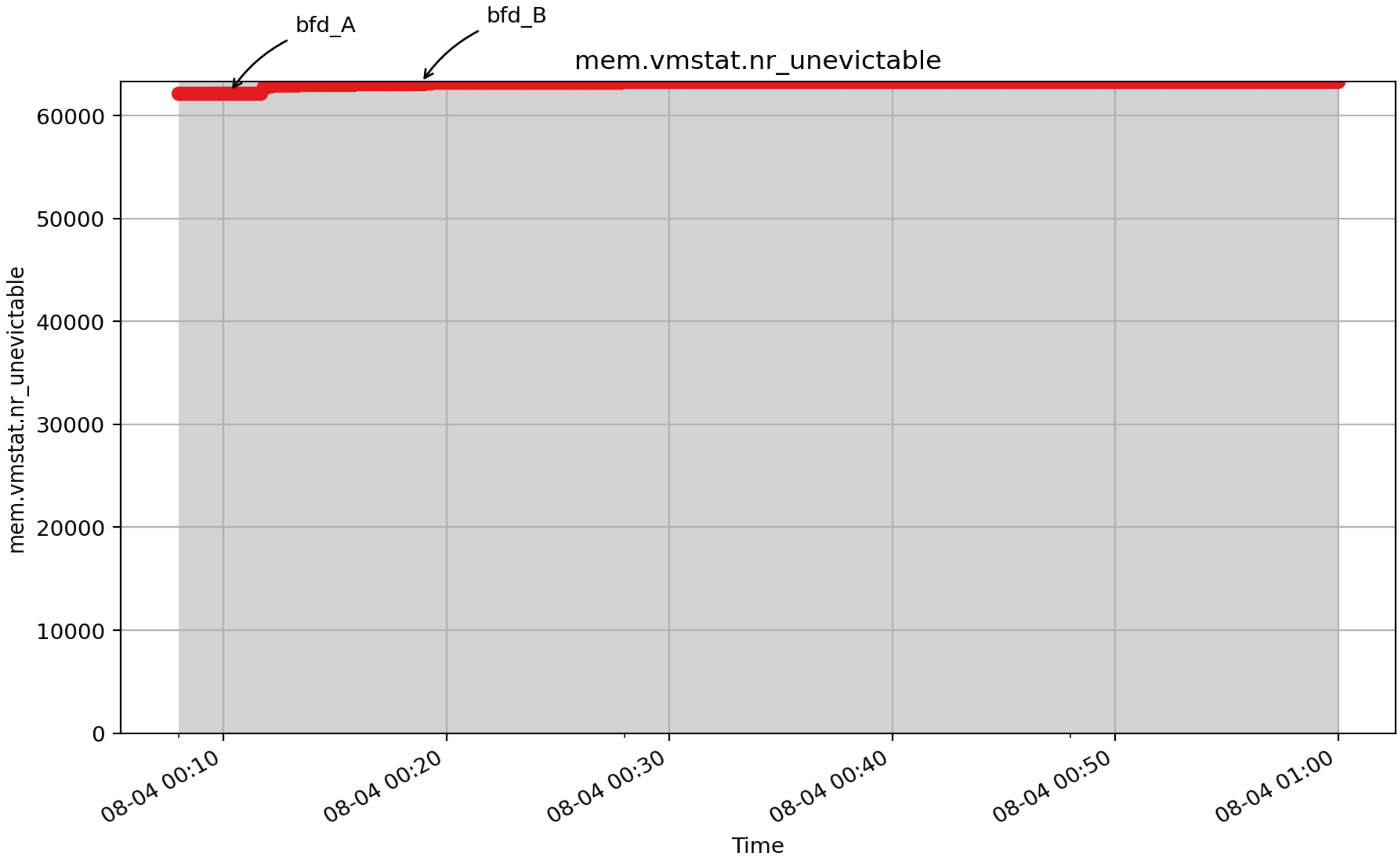
mem.vmstat.nr_slab: Instantaneous number of slab pages, from /proc/vmstat This counter was retired in 2.6.18 kernels, and is now the sum of mem.vmstat.nr_slab_reclaimable and mem.vmstat.nr_slab_unreclaimable. (- U64)



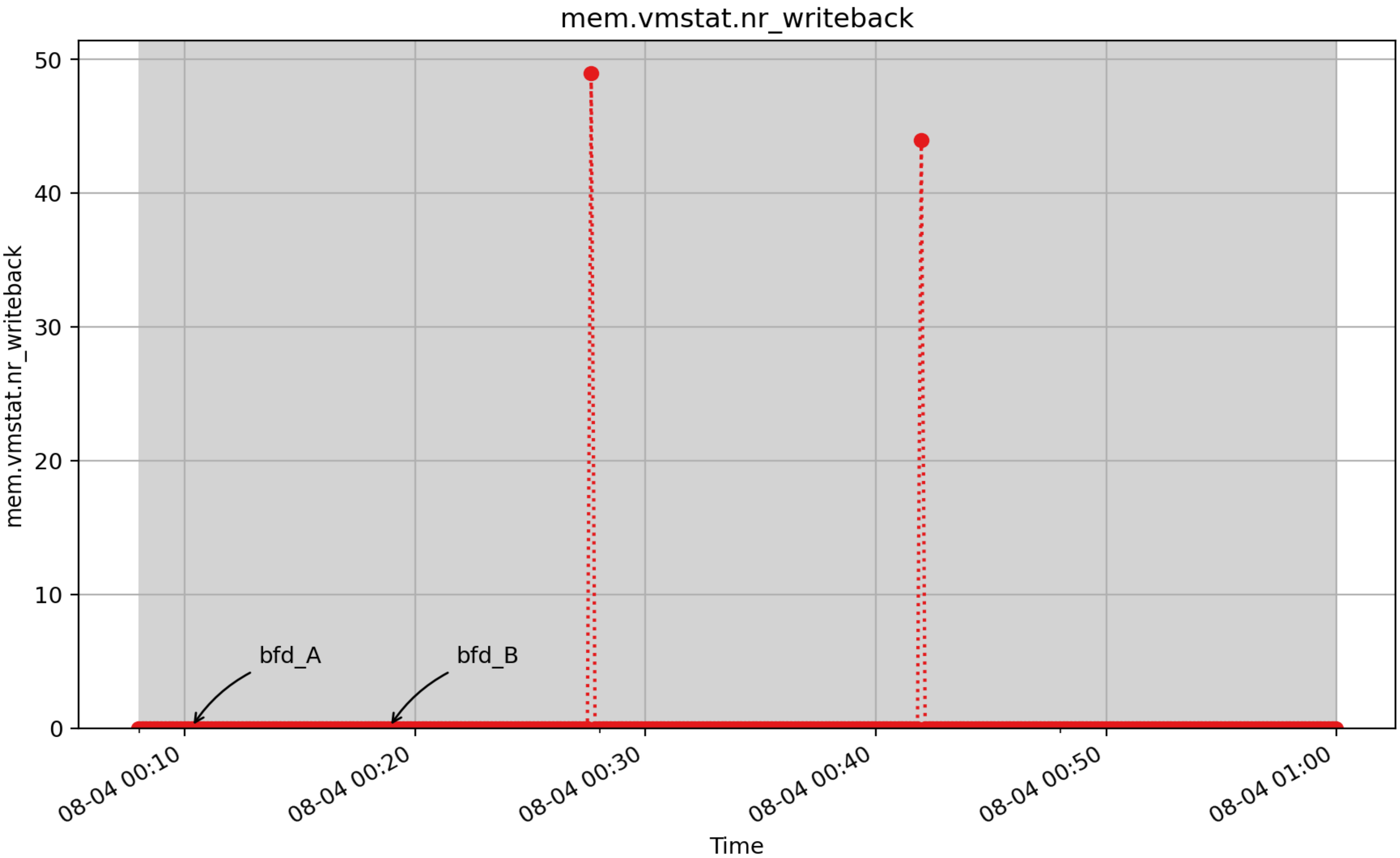
mem.vmstat.nr_slab_reclaimable: Instantaneous number of reclaimable slab pages, from /proc/vmstat. (- U64)



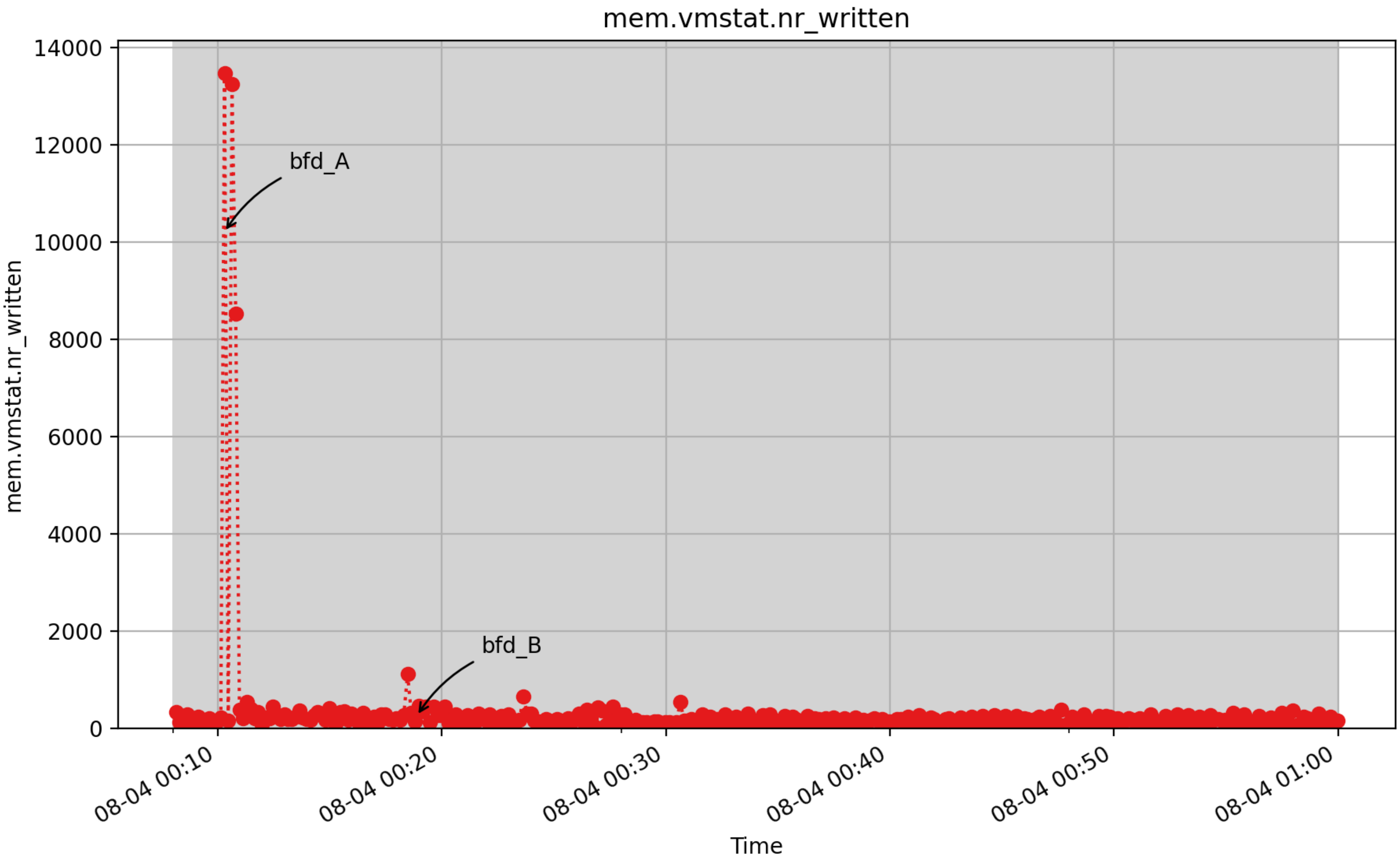
mem.vmstat.nr_slab_unreclaimable: Instantaneous number of unreclaimable slab pages, from /proc/vmstat.
(- U64)



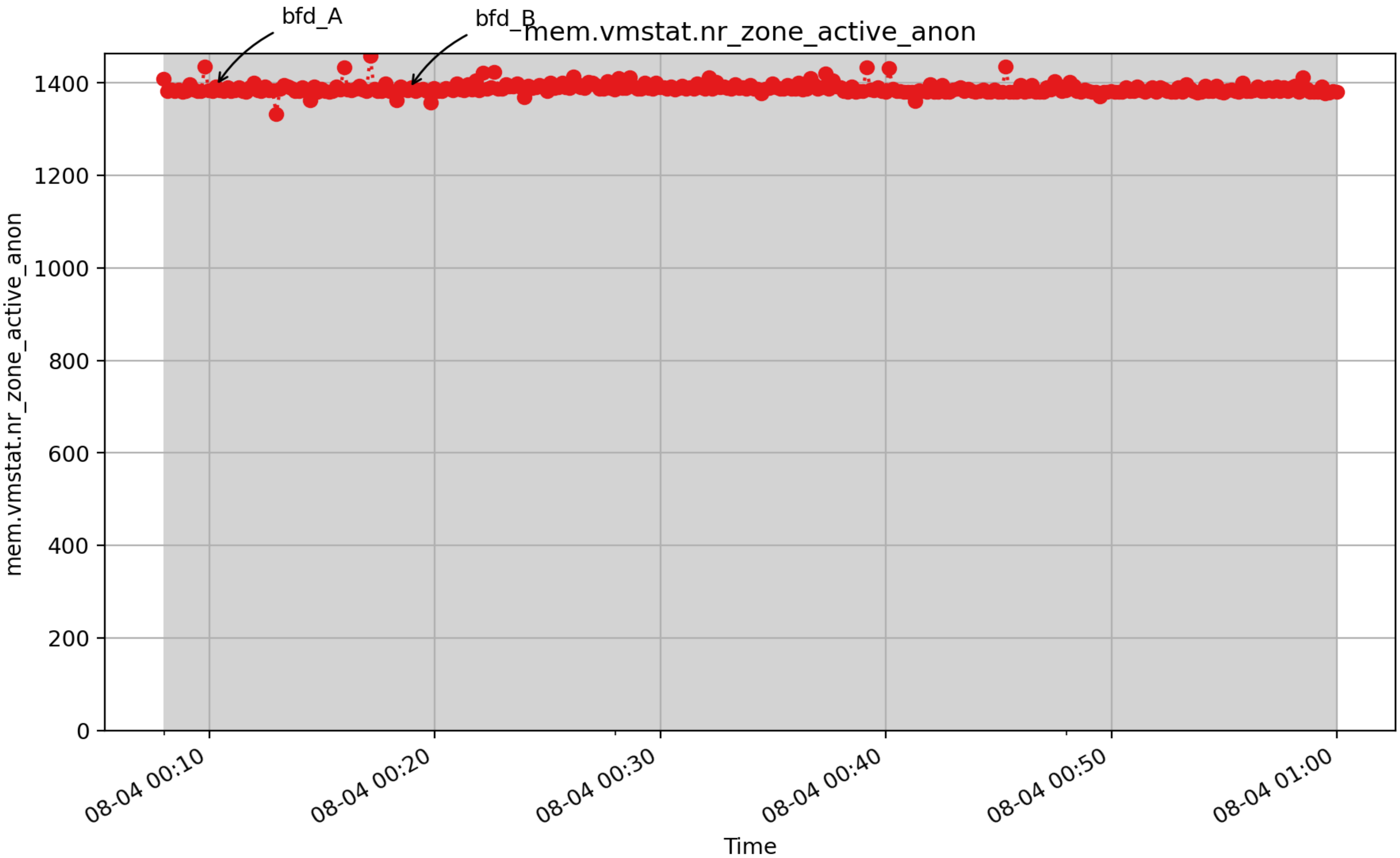
mem.vmstat.nr_unevictable: number of unevictable pages (count - U64)



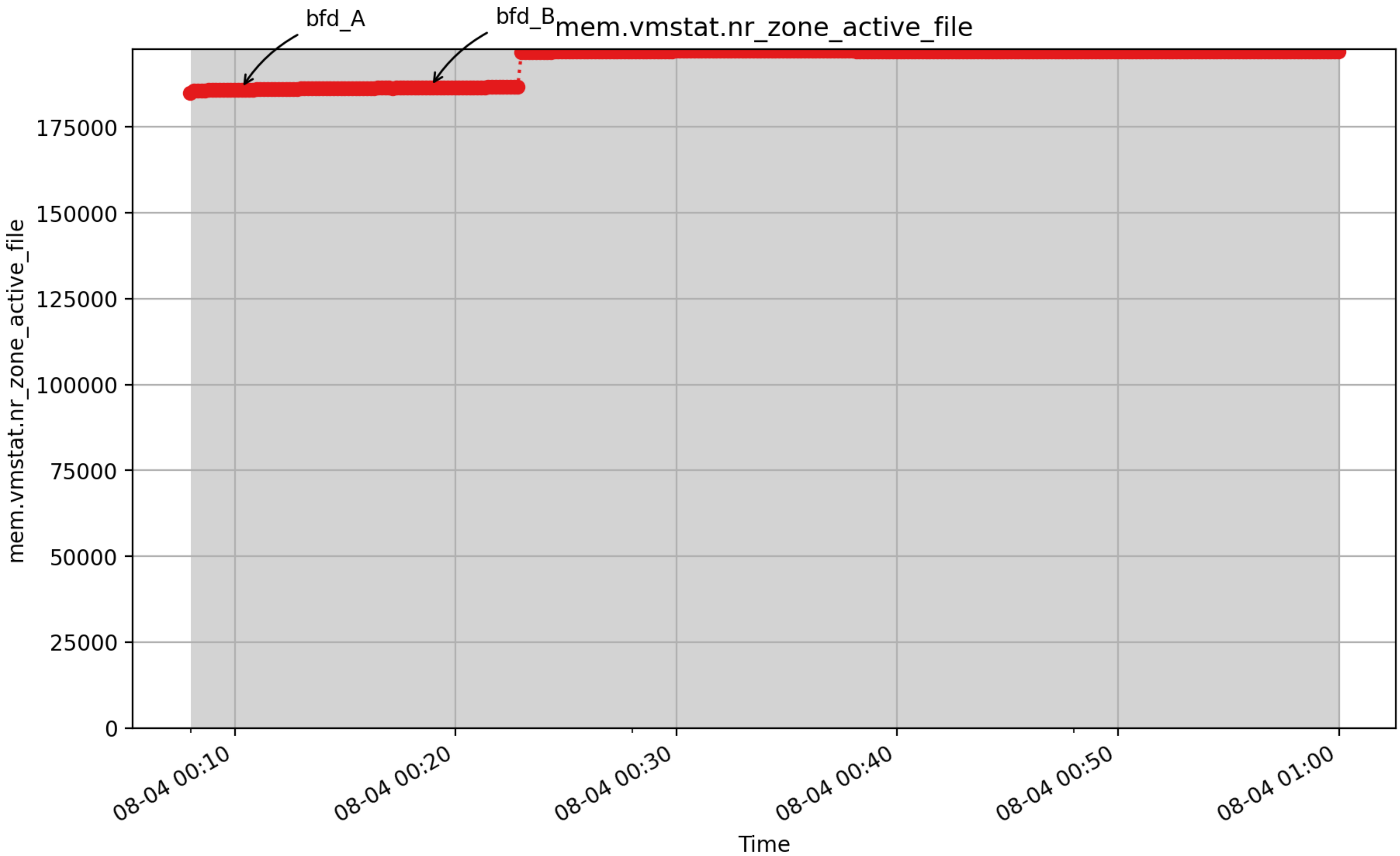
mem.vmstat.nr_writeback: Instantaneous number of pages in writeback state, from /proc/vmstat (- U64)



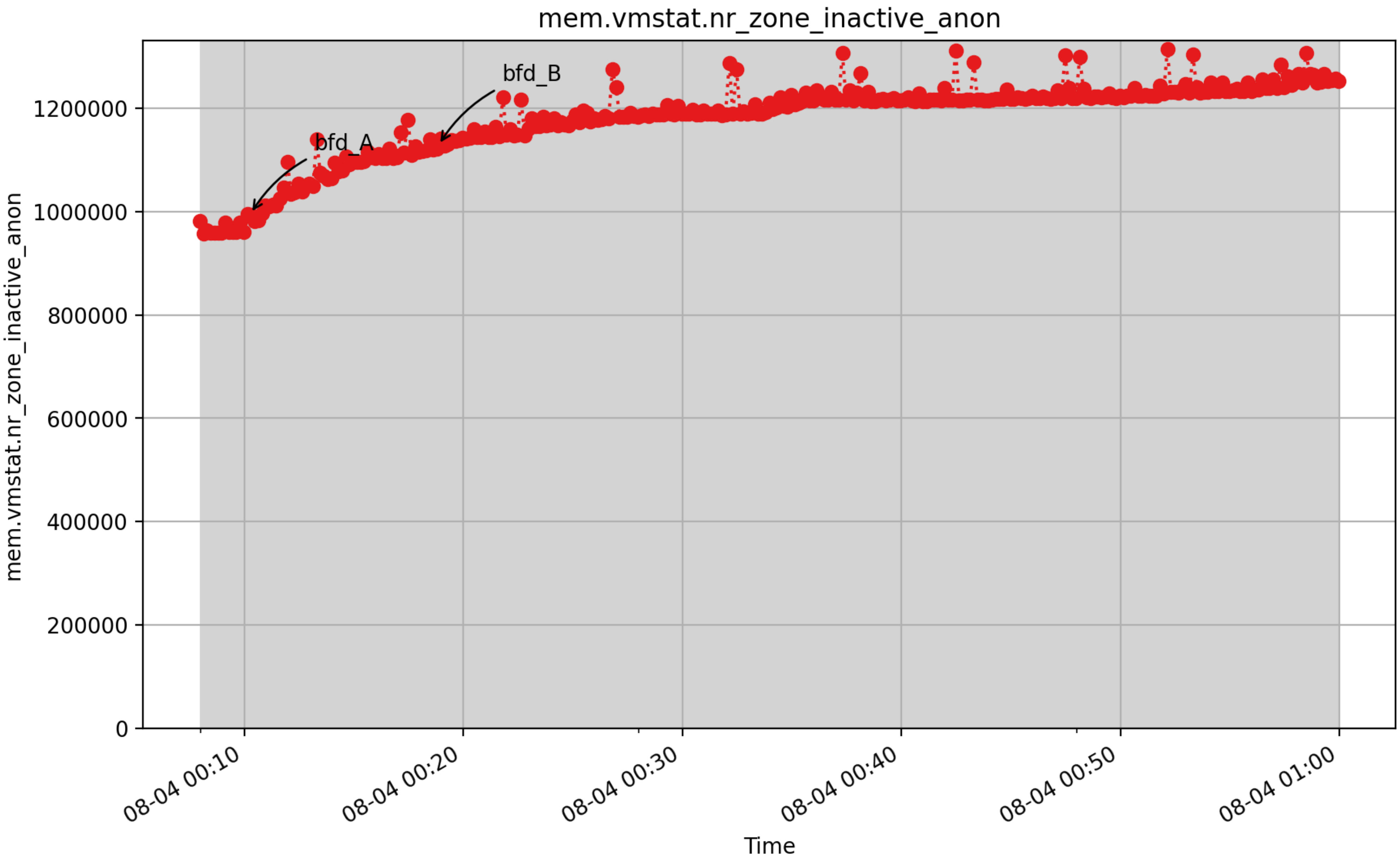
mem.vmstat.nr_written: Count of pages written out, from /proc/vmstat (count - U64) - *rate converted*



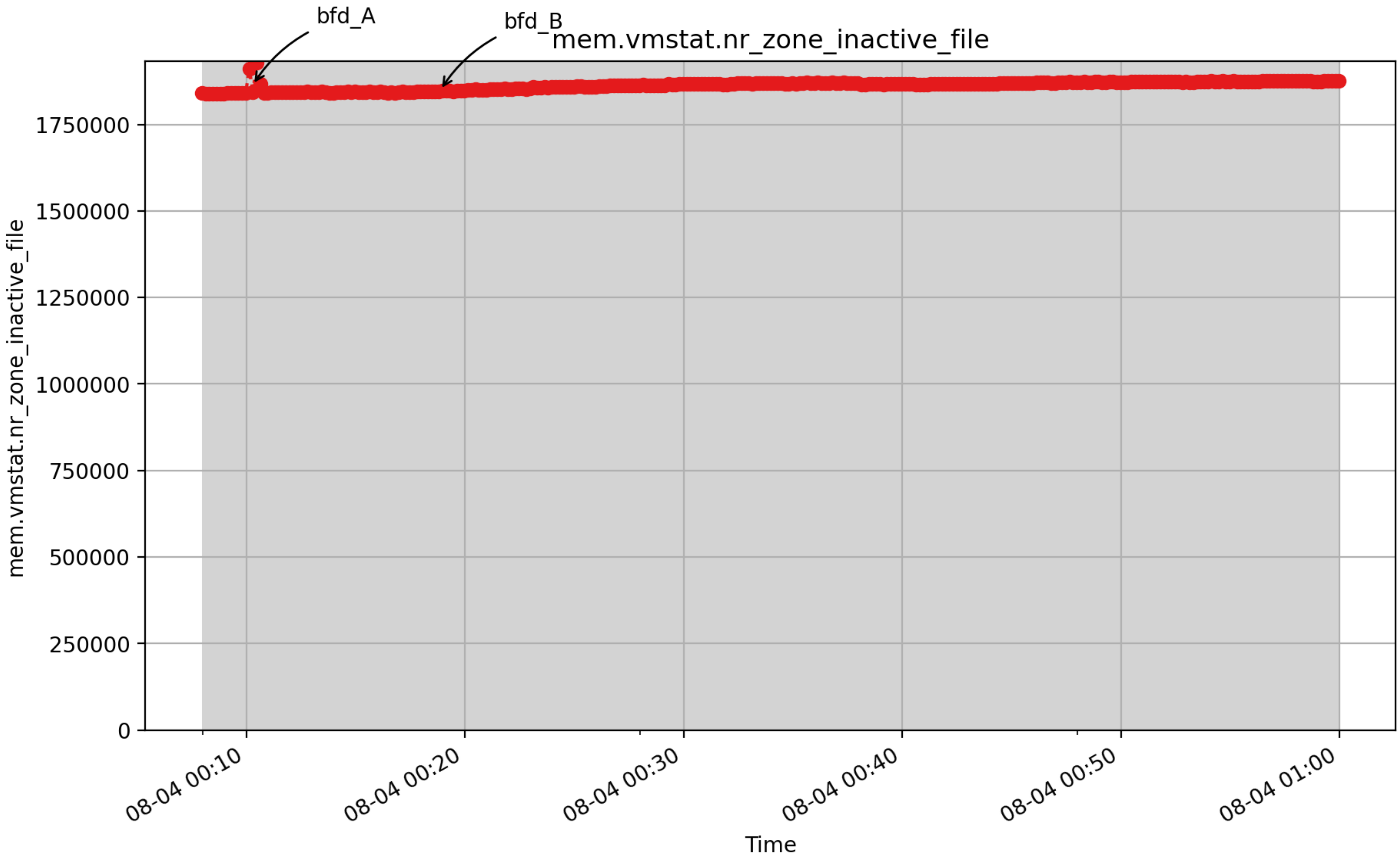
mem.vmstat.nr_zone_active_anon: number of inactive file memory pages in zones (count - U64)



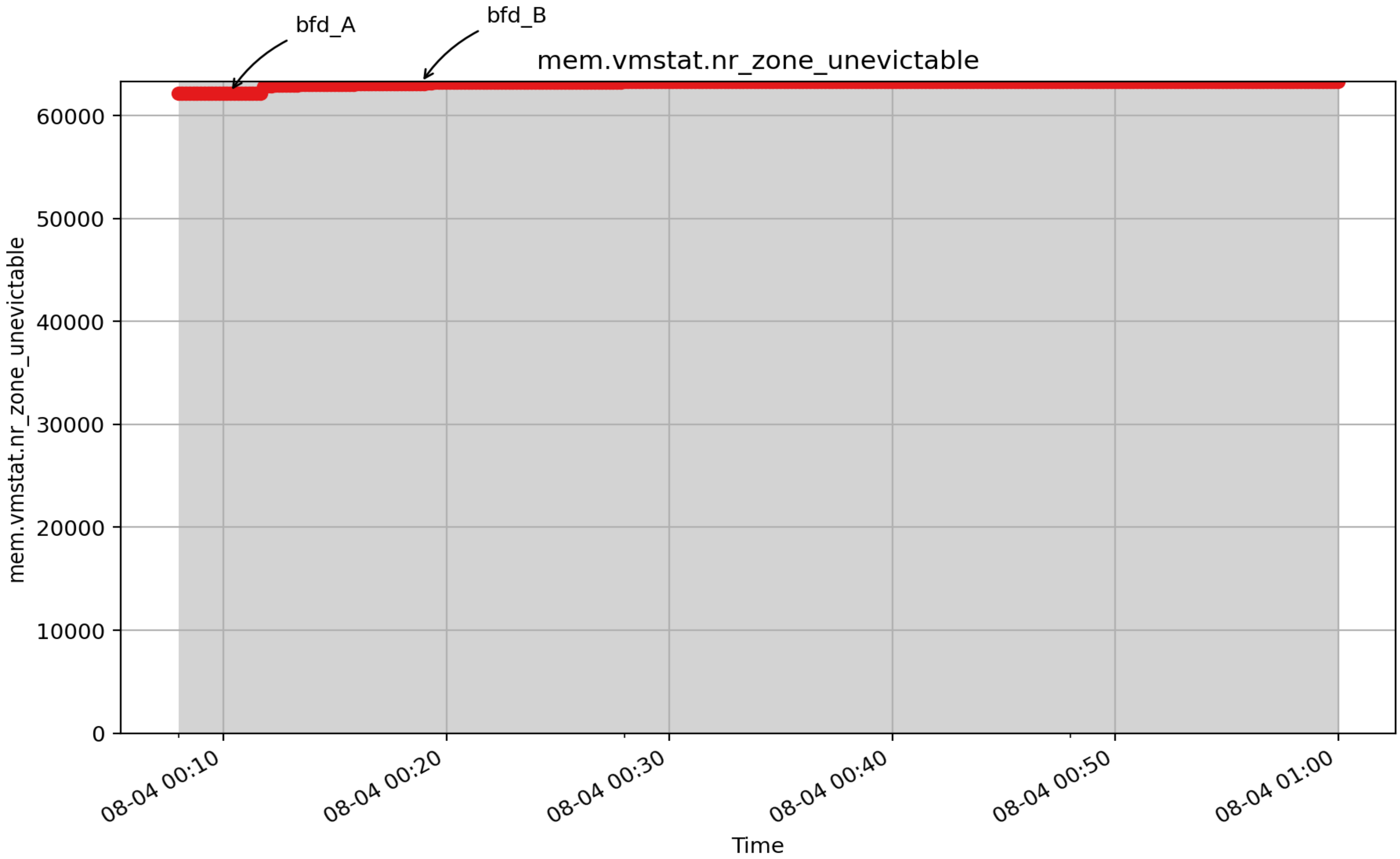
mem.vmstat.nr_zone_active_file: number of isolated file memory pages in zones (count - U64)



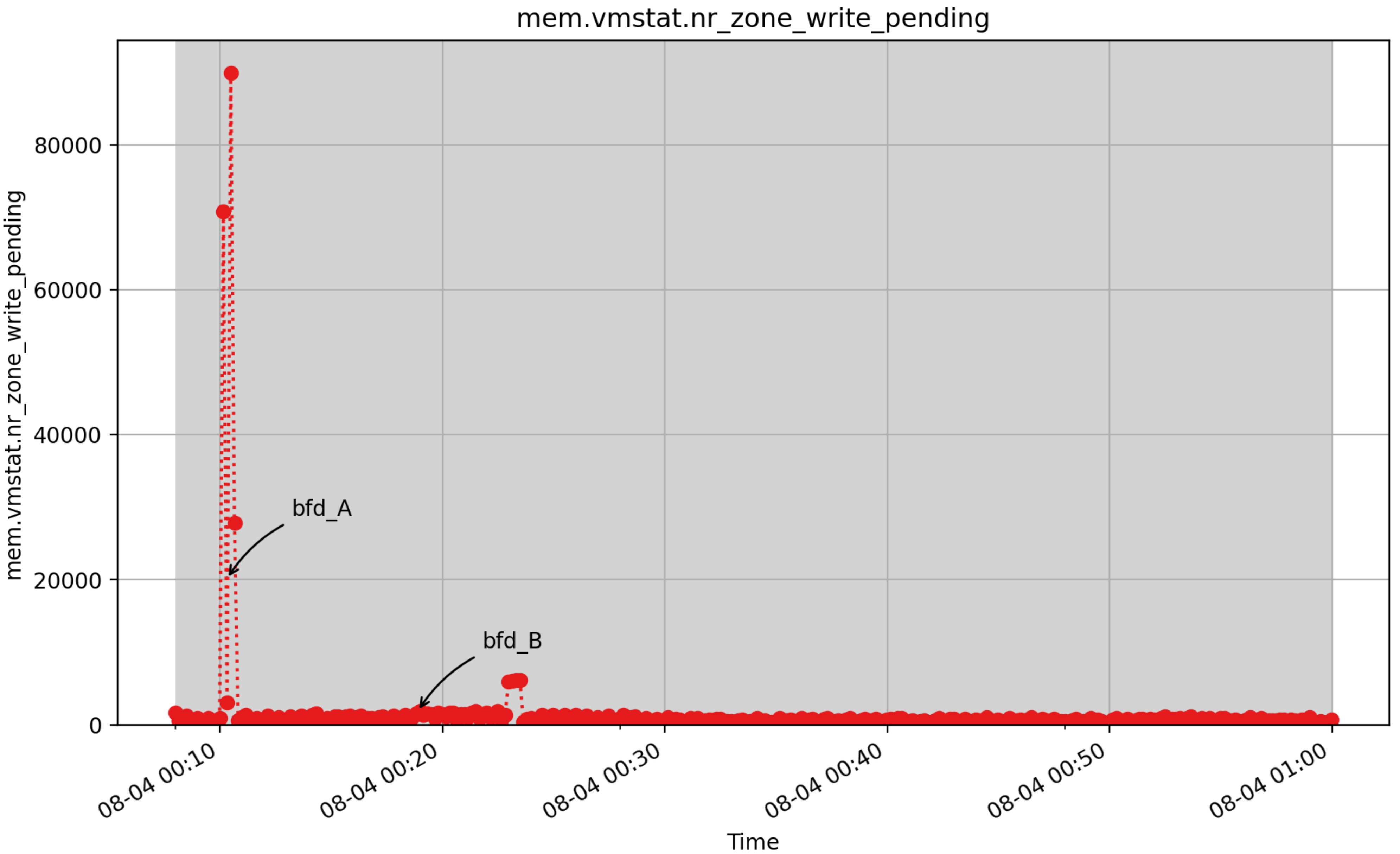
mem.vmstat.nr_zone_inactive_anon: number of inactive anonymous memory pages in zones (count - U64)



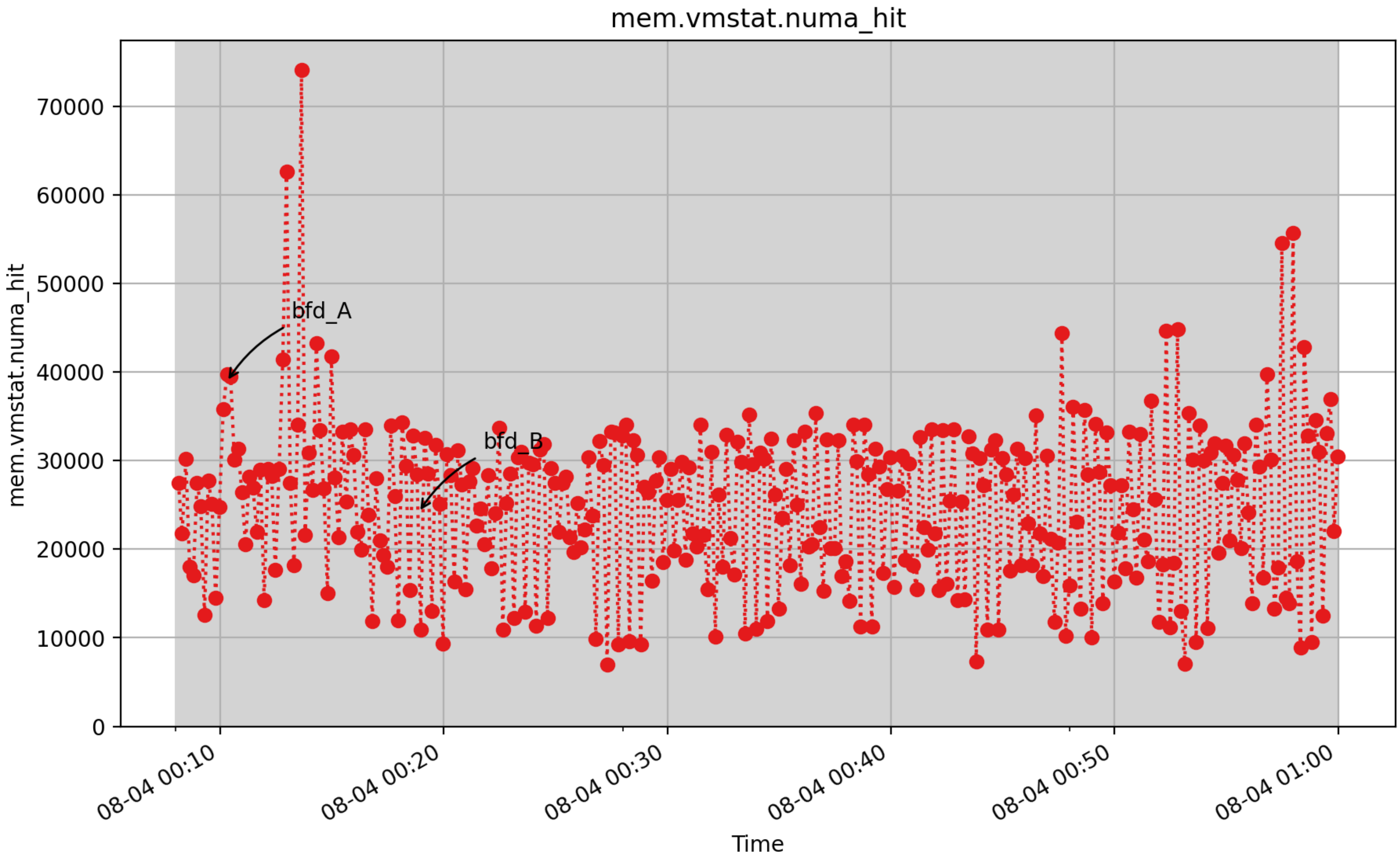
mem.vmstat.nr_zone_inactive_file: number of isolated anonymous memory pages in zones (count - U64)



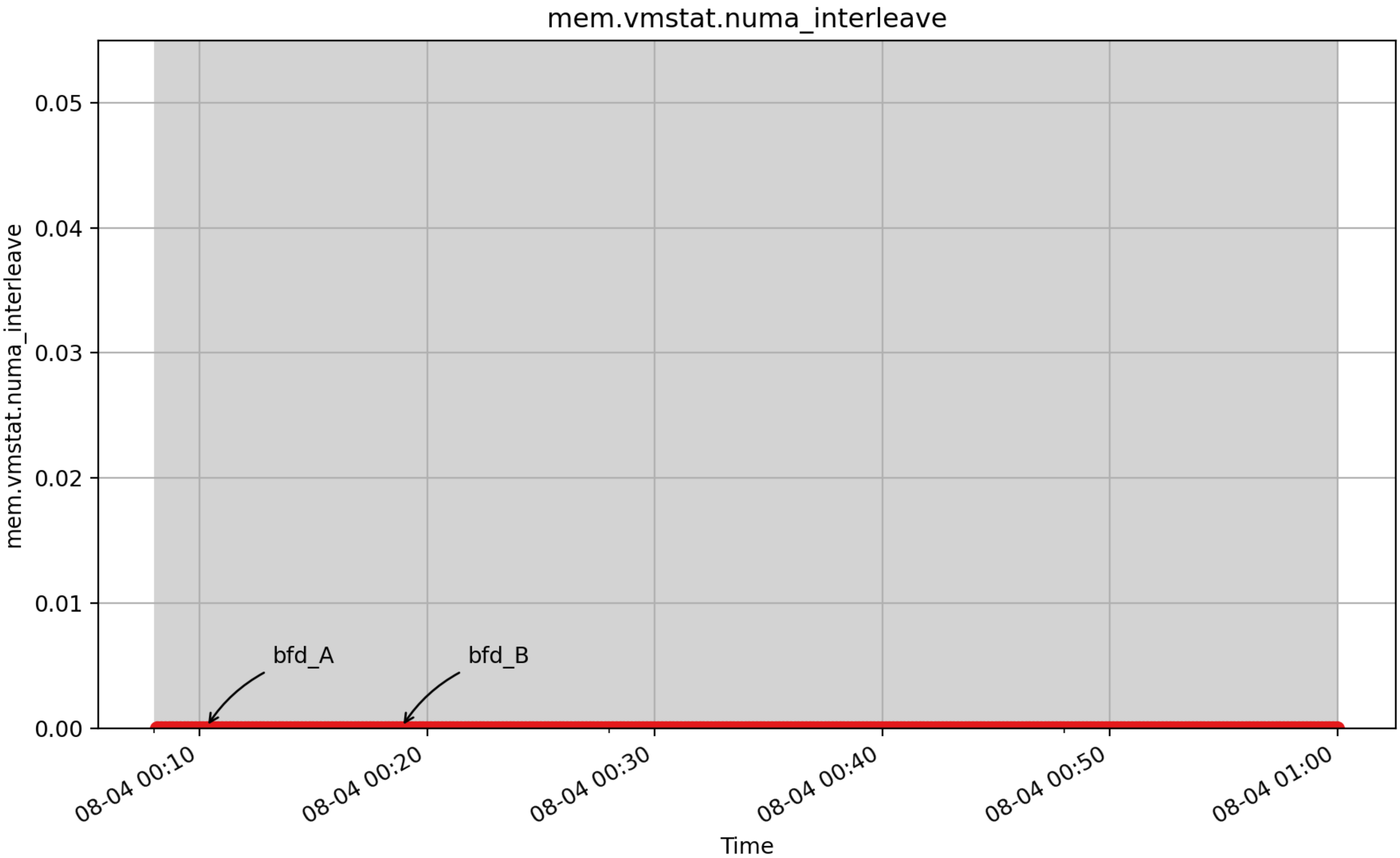
mem.vmstat.nr_zone_unevictable: number of unevictable memory pages in zones (count - U64)



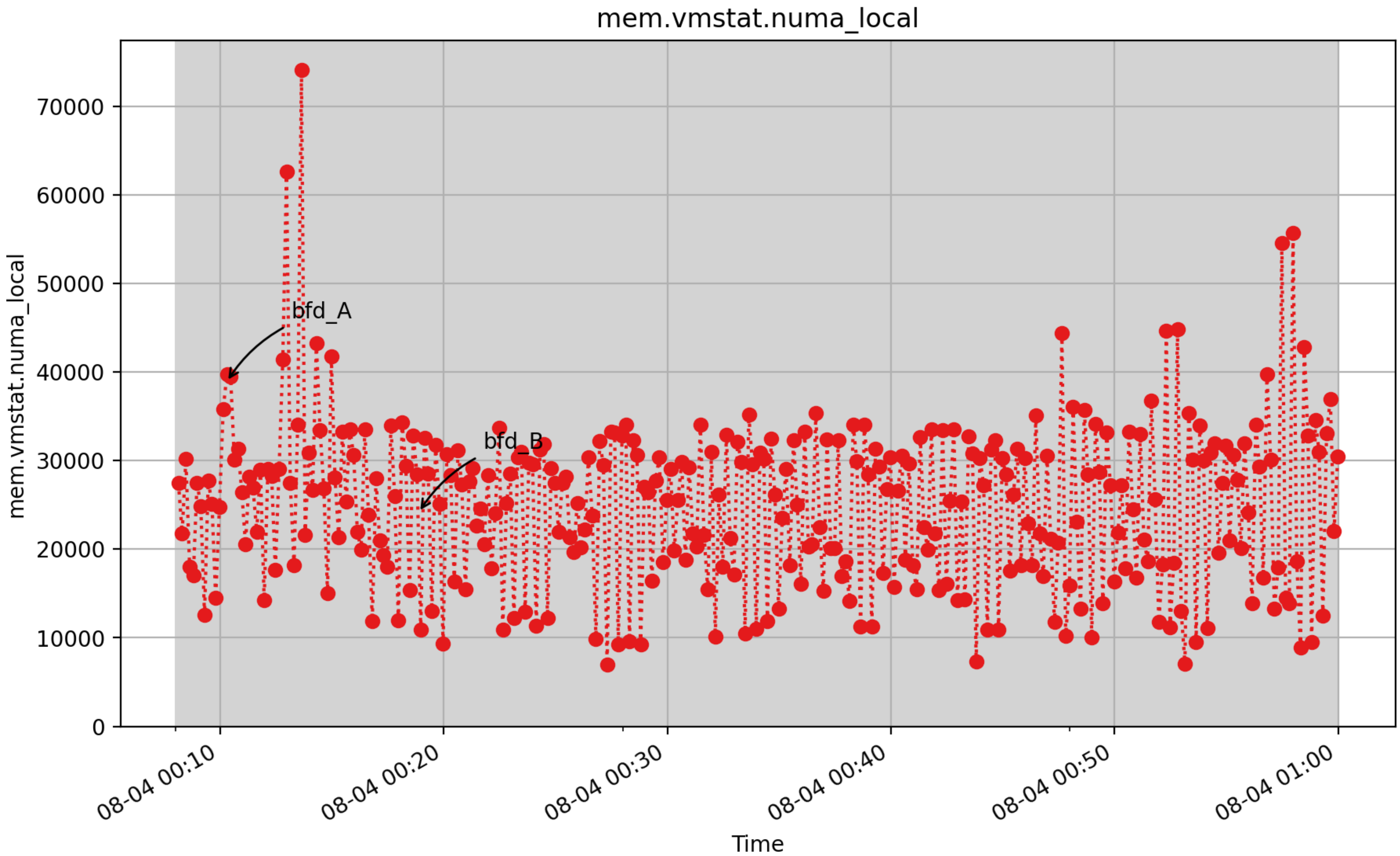
mem.vmstat.nr_zone_write_pending: count of dirty, writeback and unstable pages (count - U64)



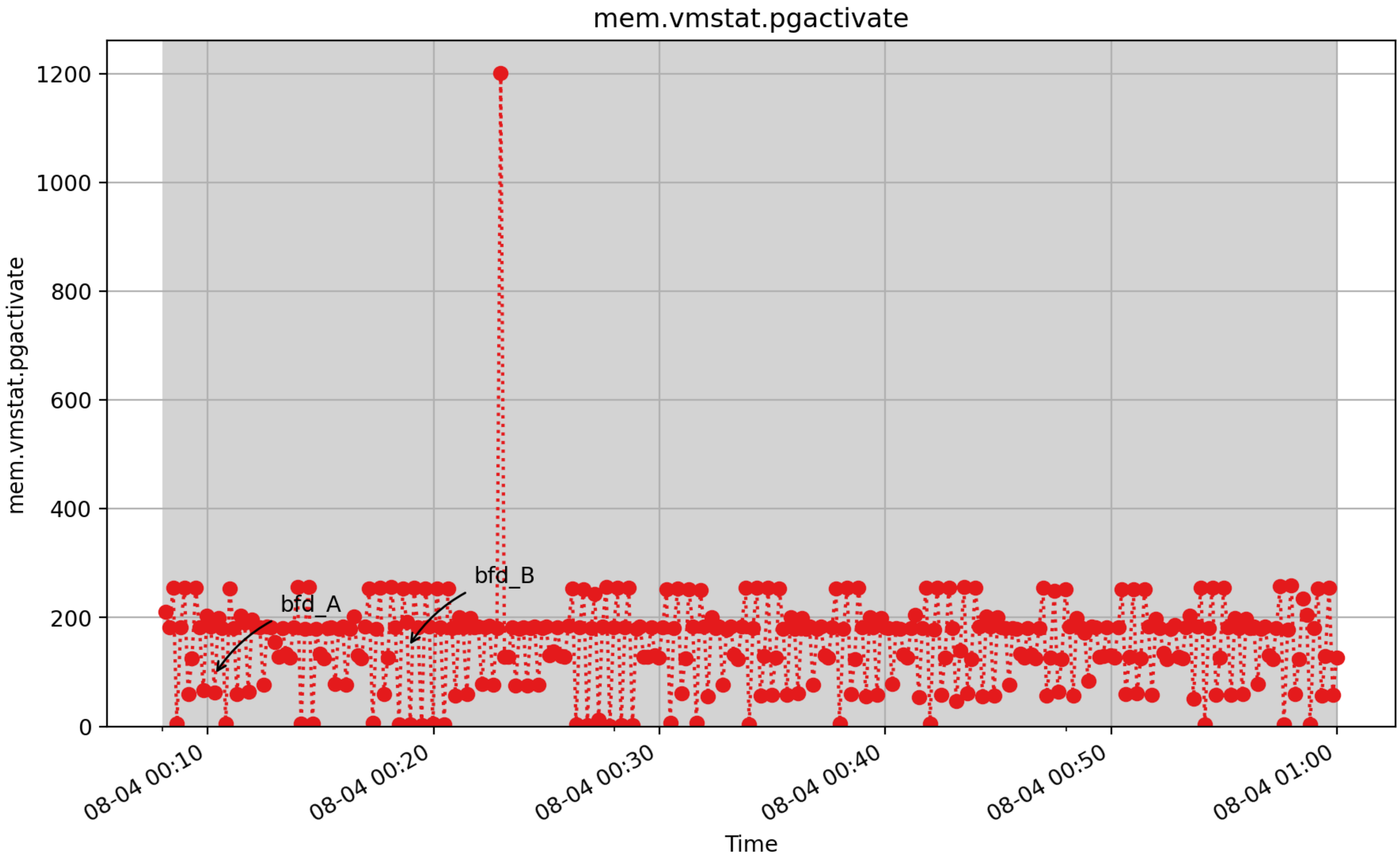
mem.vmstat.numa_hit: count of successful allocations from preferred NUMA zone (count - U64) - *rate converted*



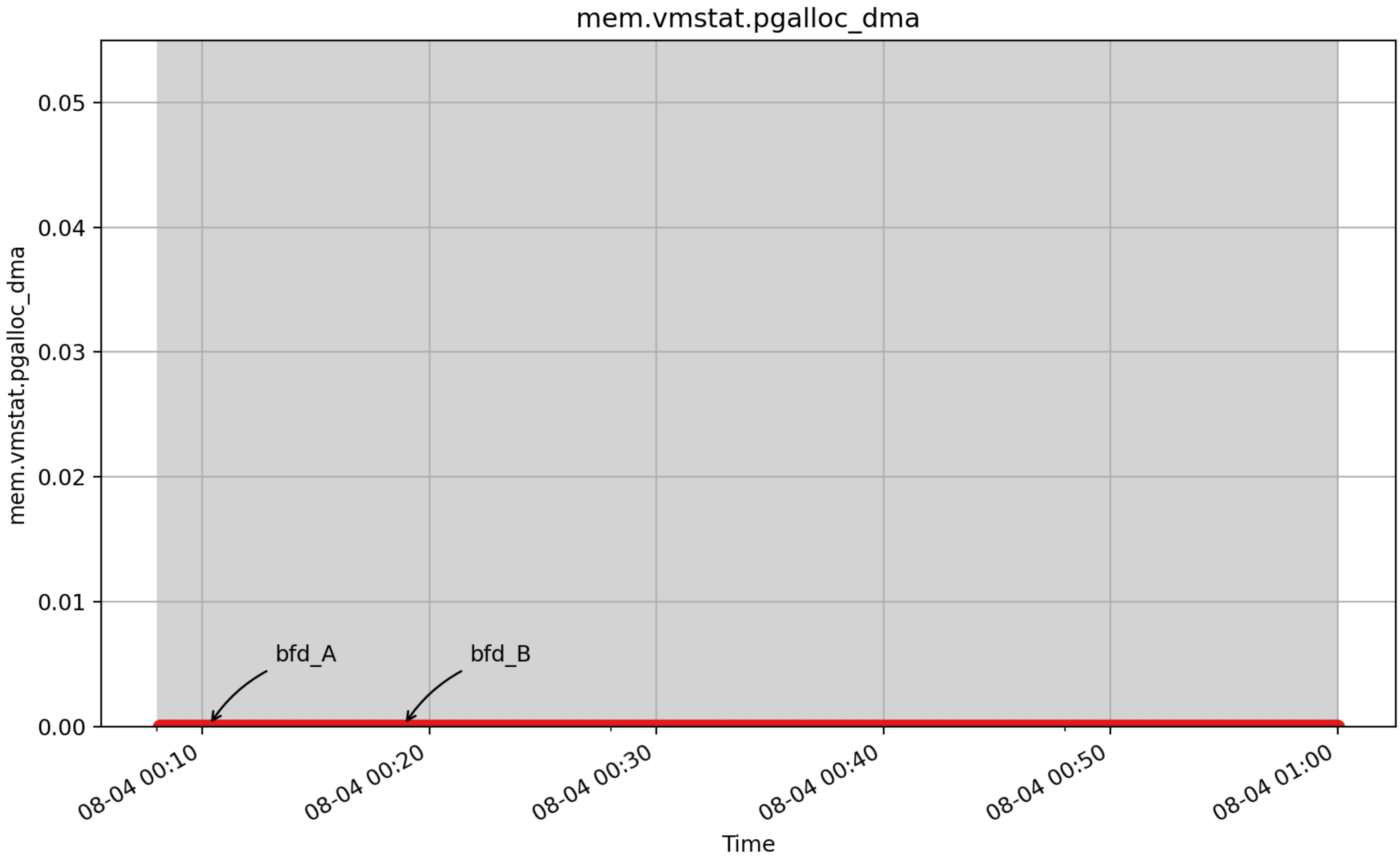
mem.vmstat.numa_interleave: count of interleaved NUMA allocations (count - U64) - *rate converted*



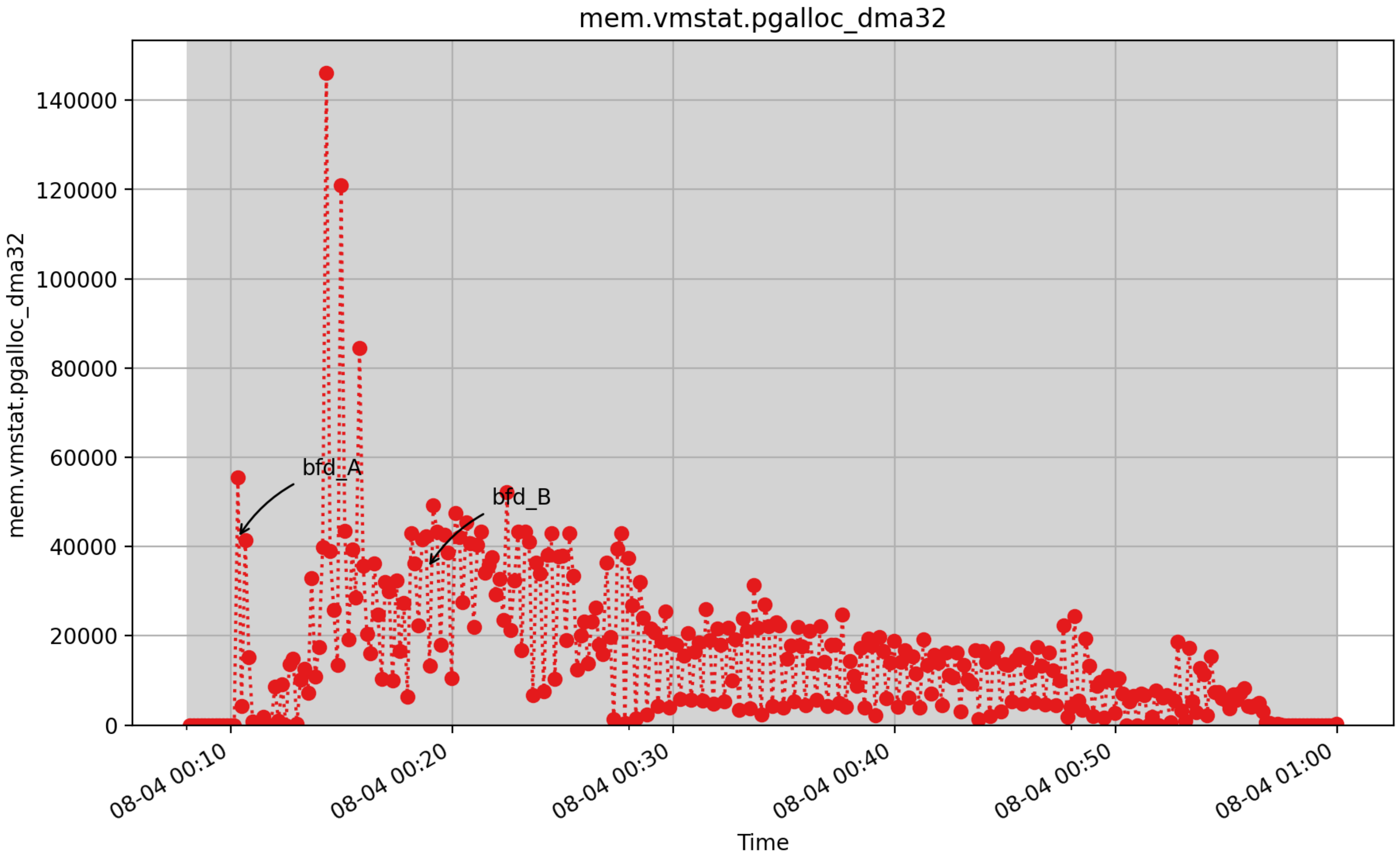
mem.vmstat.numa_local: count of successful allocations from local NUMA zone (count - U64) - *rate converted*



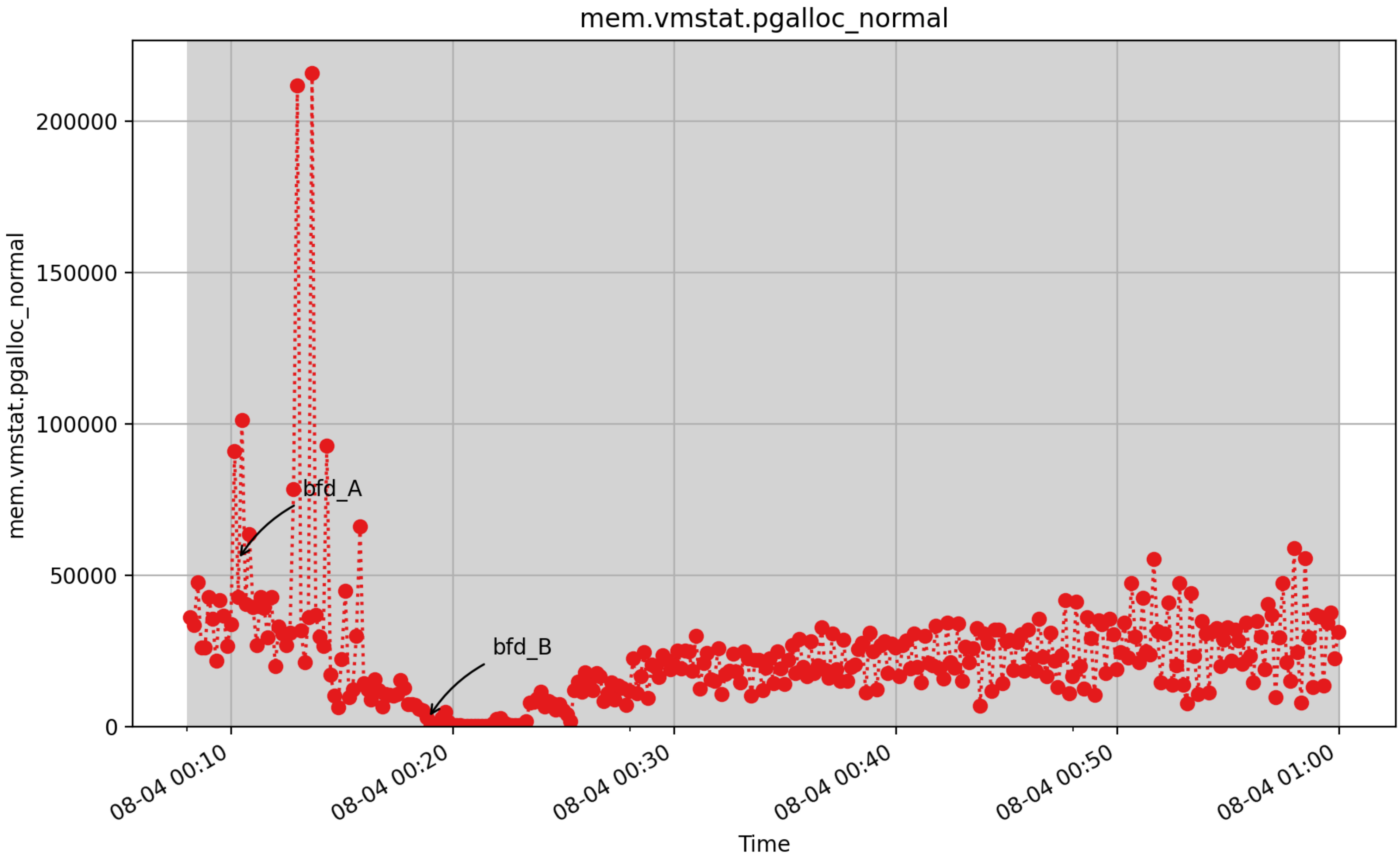
mem.vmstat.pgactivate: Count of pages moved from inactive to active since boot, from /proc/vmstat (count - U64) - *rate converted*



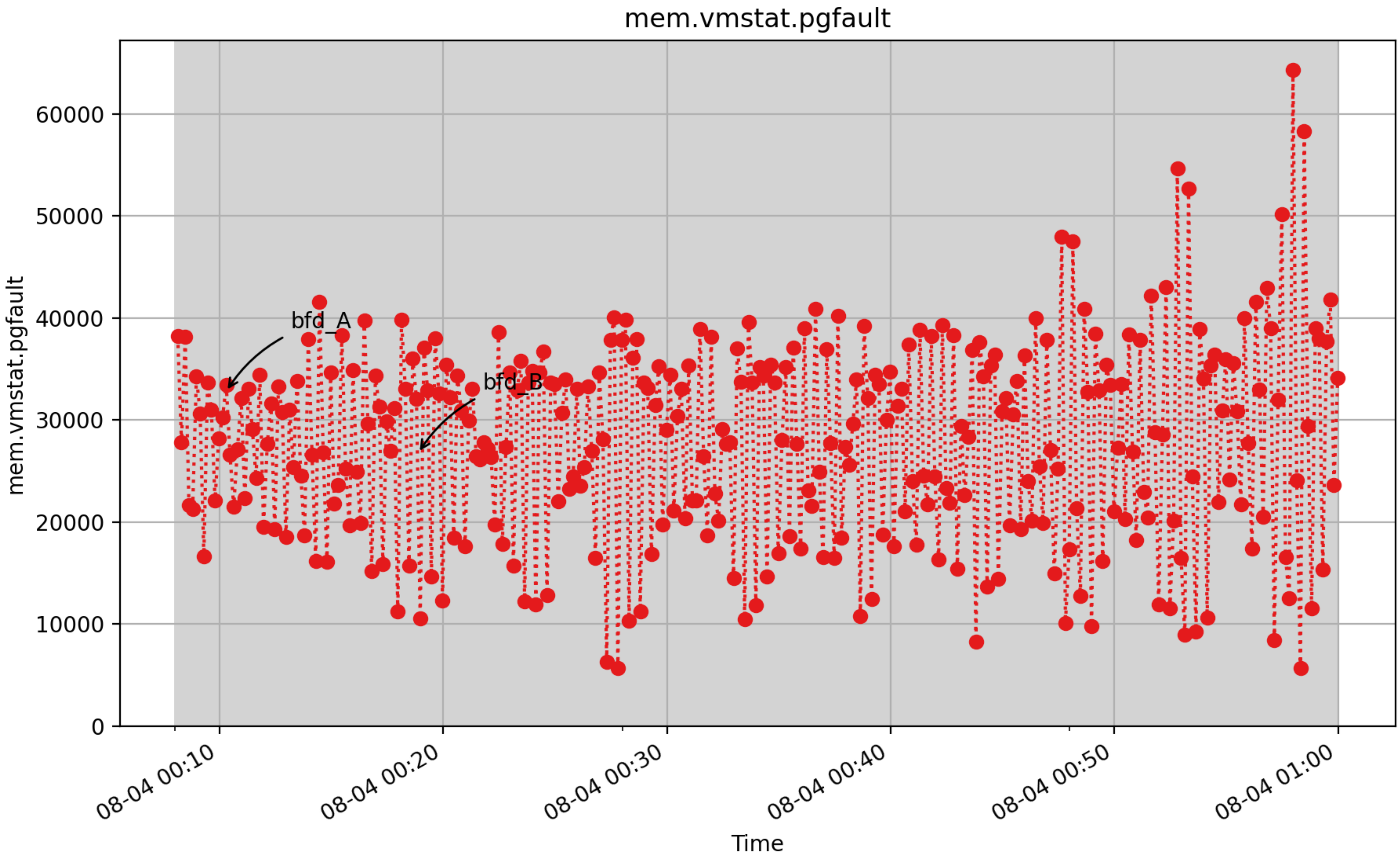
mem.vmstat.pgalloc_dma: Count of dma mem page allocations since boot, from /proc/vmstat (count - U64) - *rate converted*



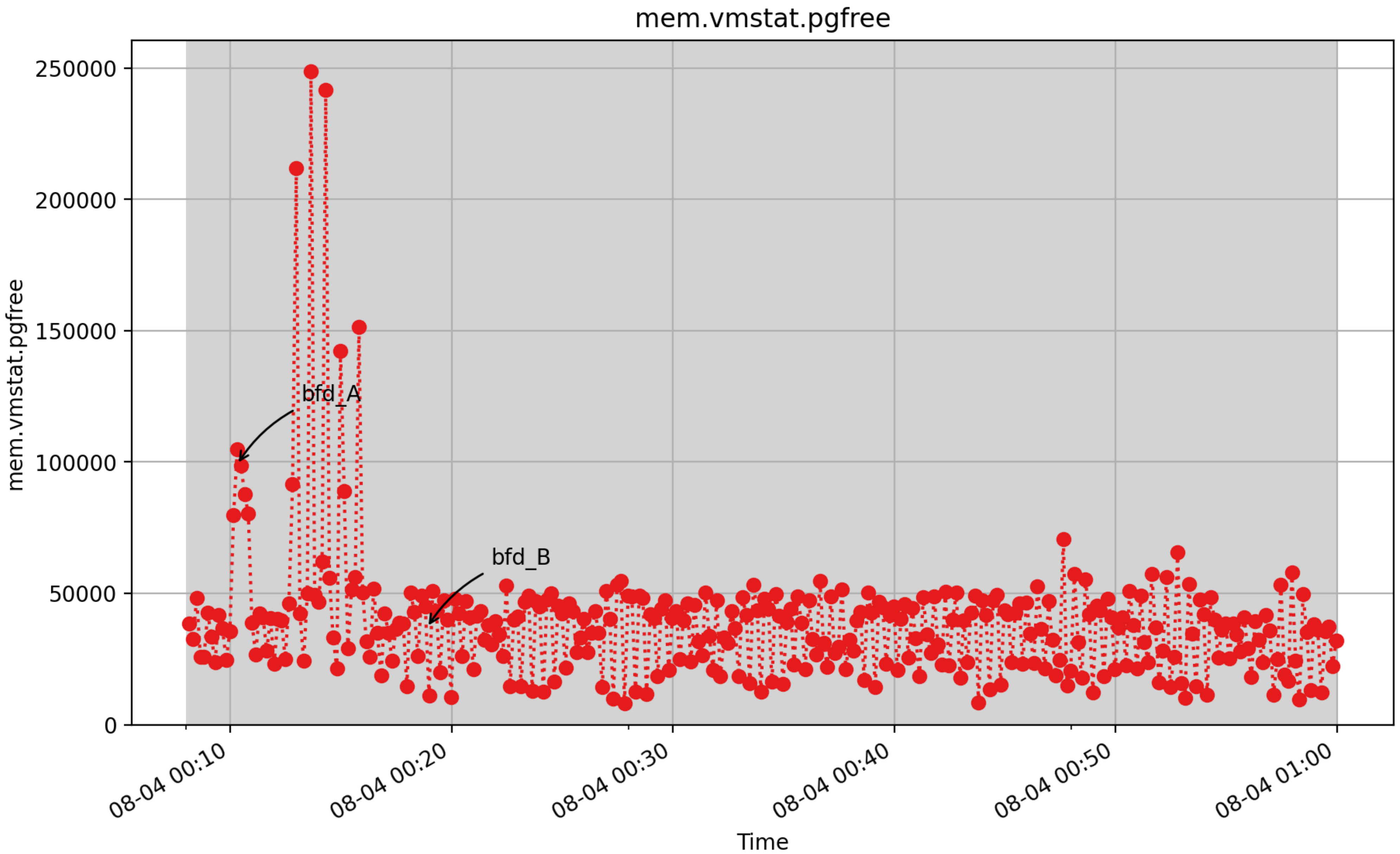
mem.vmstat.pgalloc_dma32: Count of dma32 mem page allocations since boot, from /proc/vmstat (count - U64) - *rate converted*



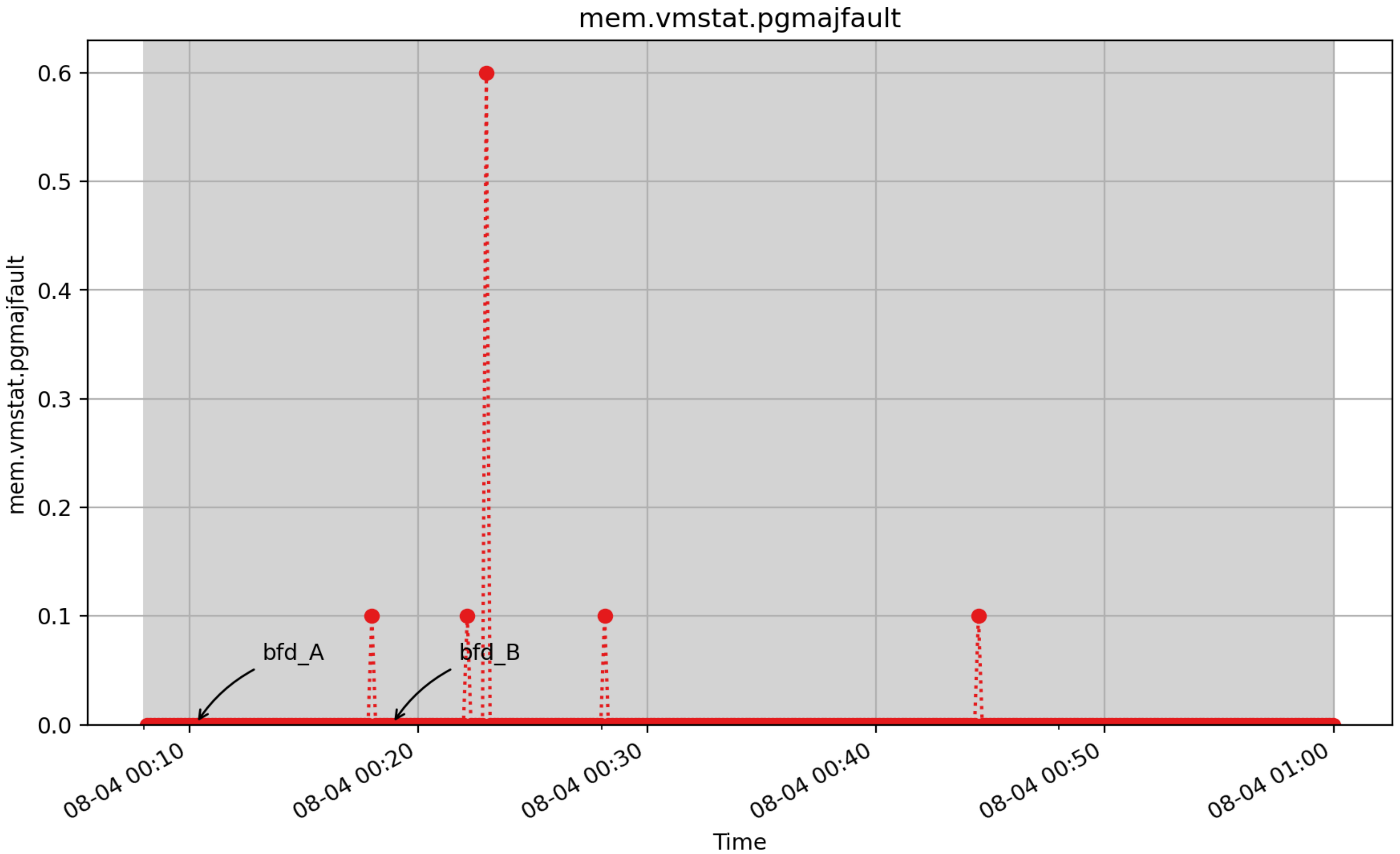
mem.vmstat.pgalloc_normal: Count of normal mem page allocations since boot, from /proc/vmstat (count - U64) - *rate converted*



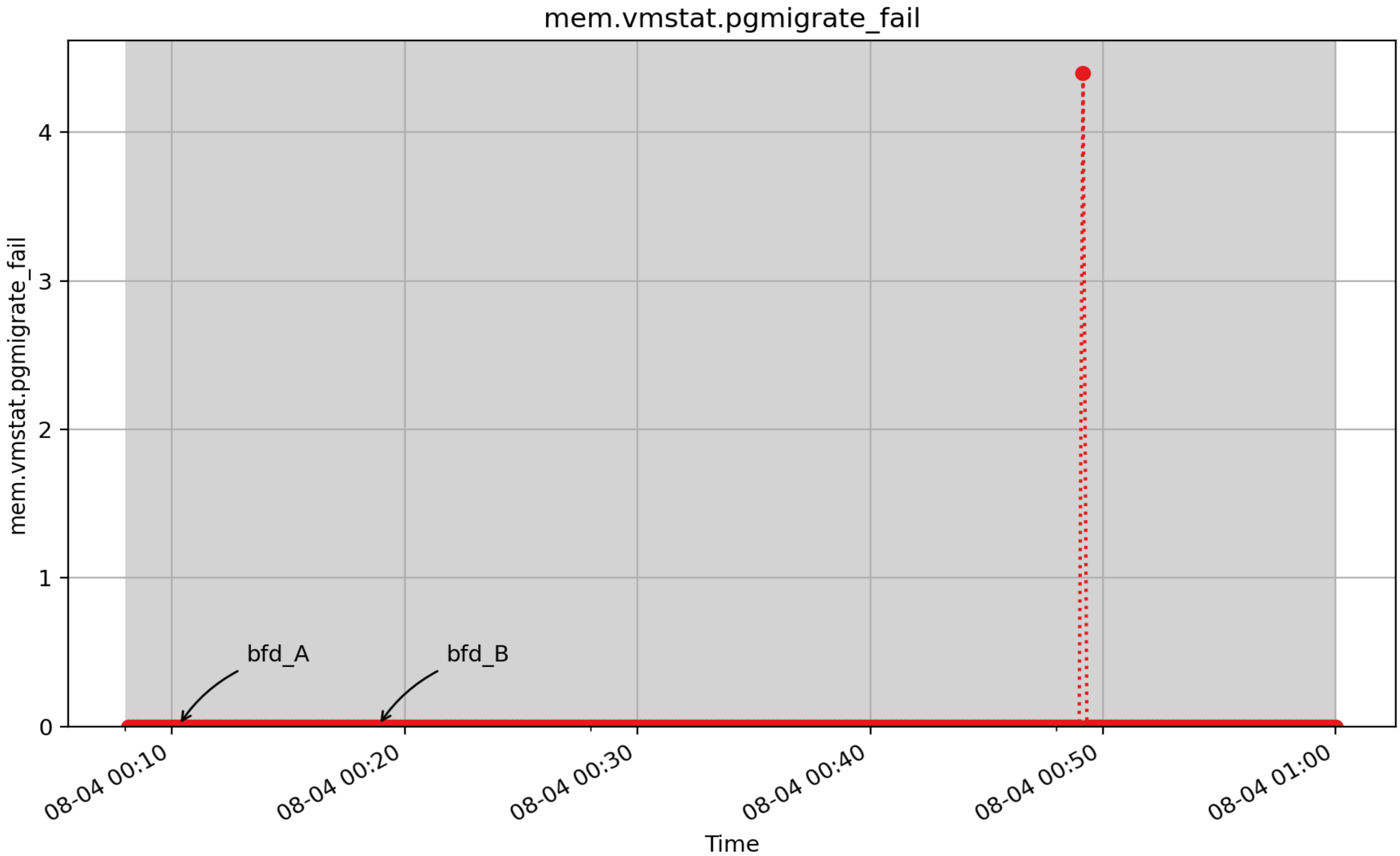
mem.vmstat.pgfault: Count of page major and minor fault operations since boot, from /proc/vmstat (count - U64) - *rate converted*



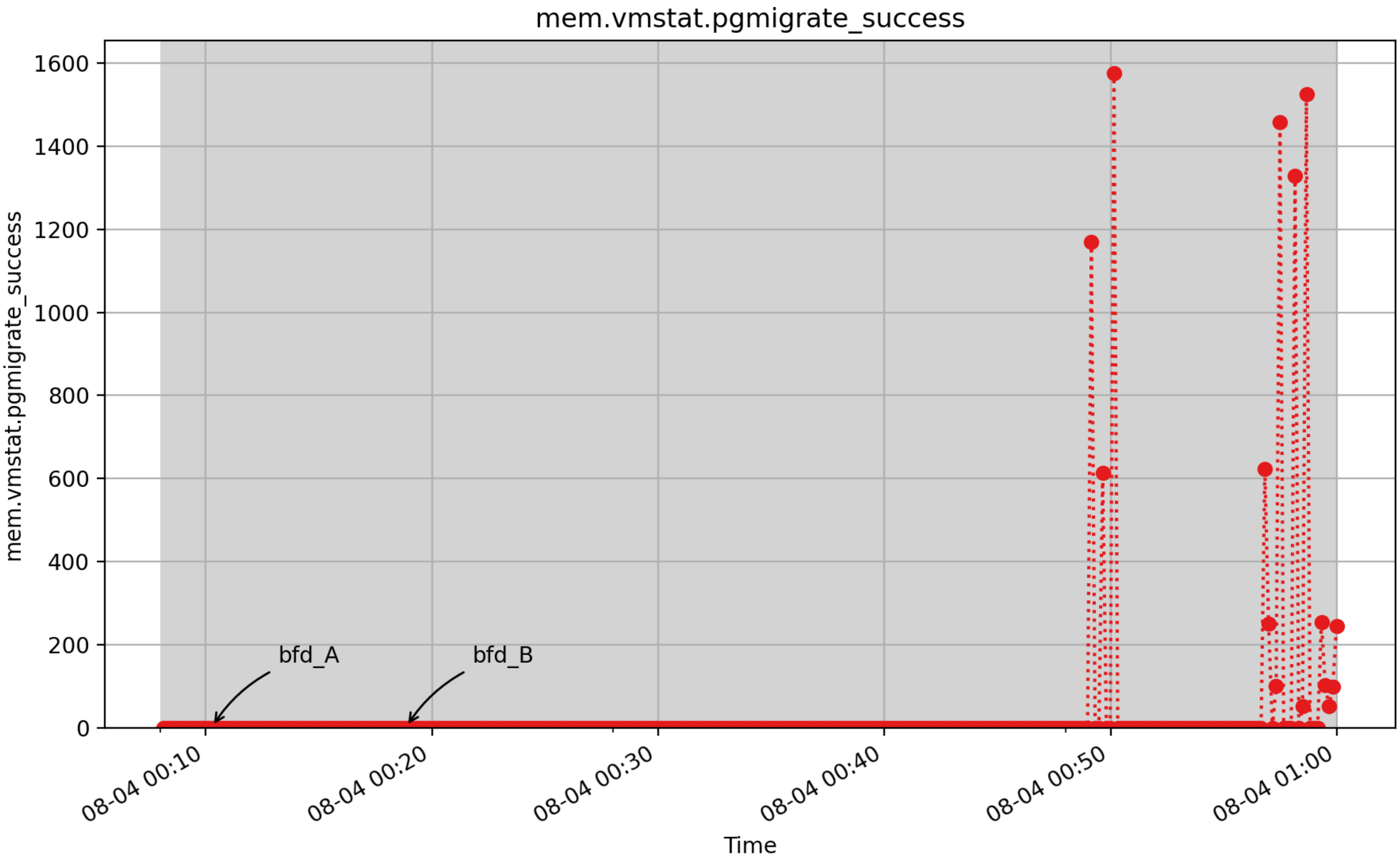
mem.vmstat.pgfree: Count of page free operations since boot, from /proc/vmstat (count - U64) - *rate converted*



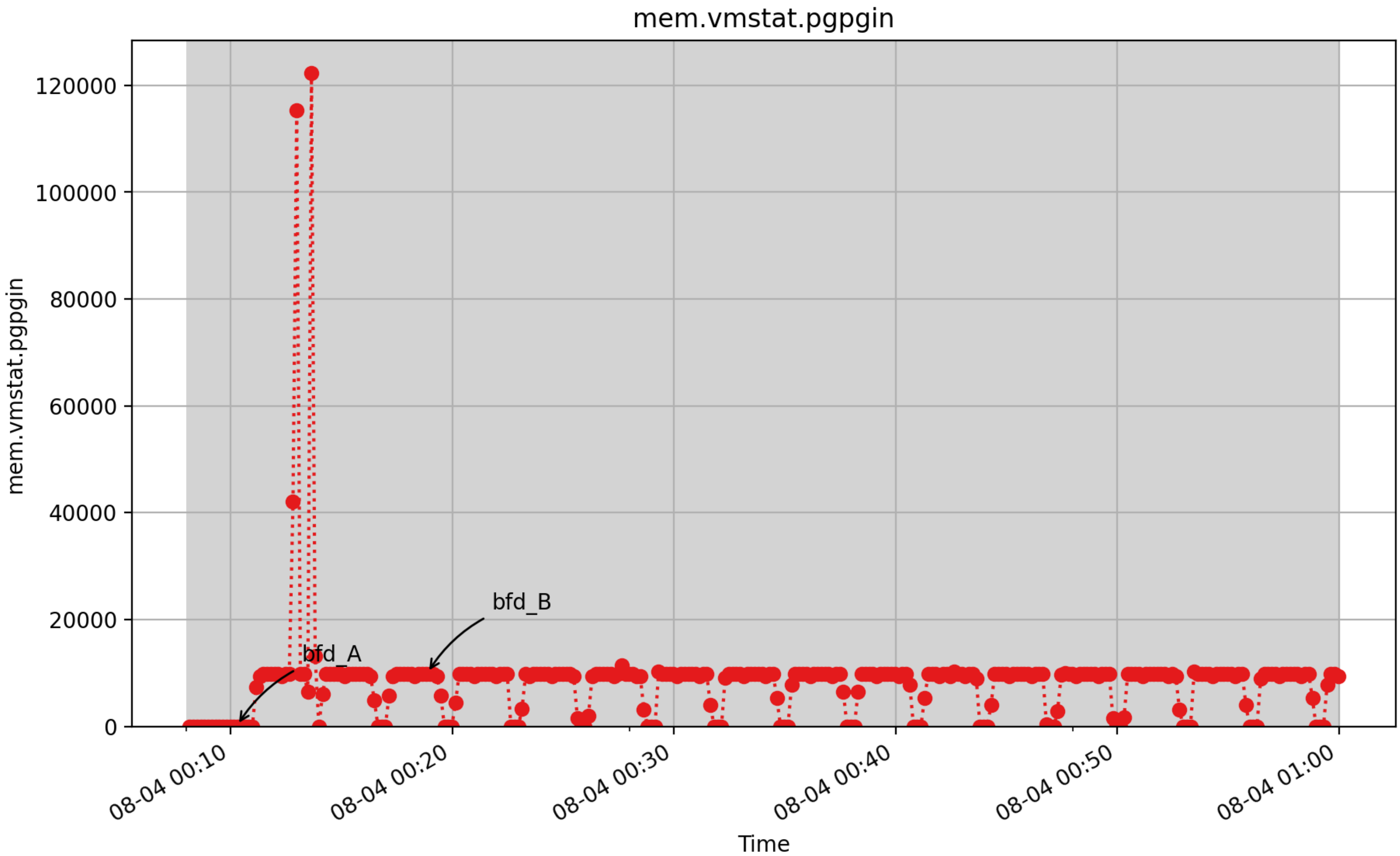
mem.vmstat.pgmajfault: Count of major page fault operations since boot, from /proc/vmstat (count - U64) - *rate converted*



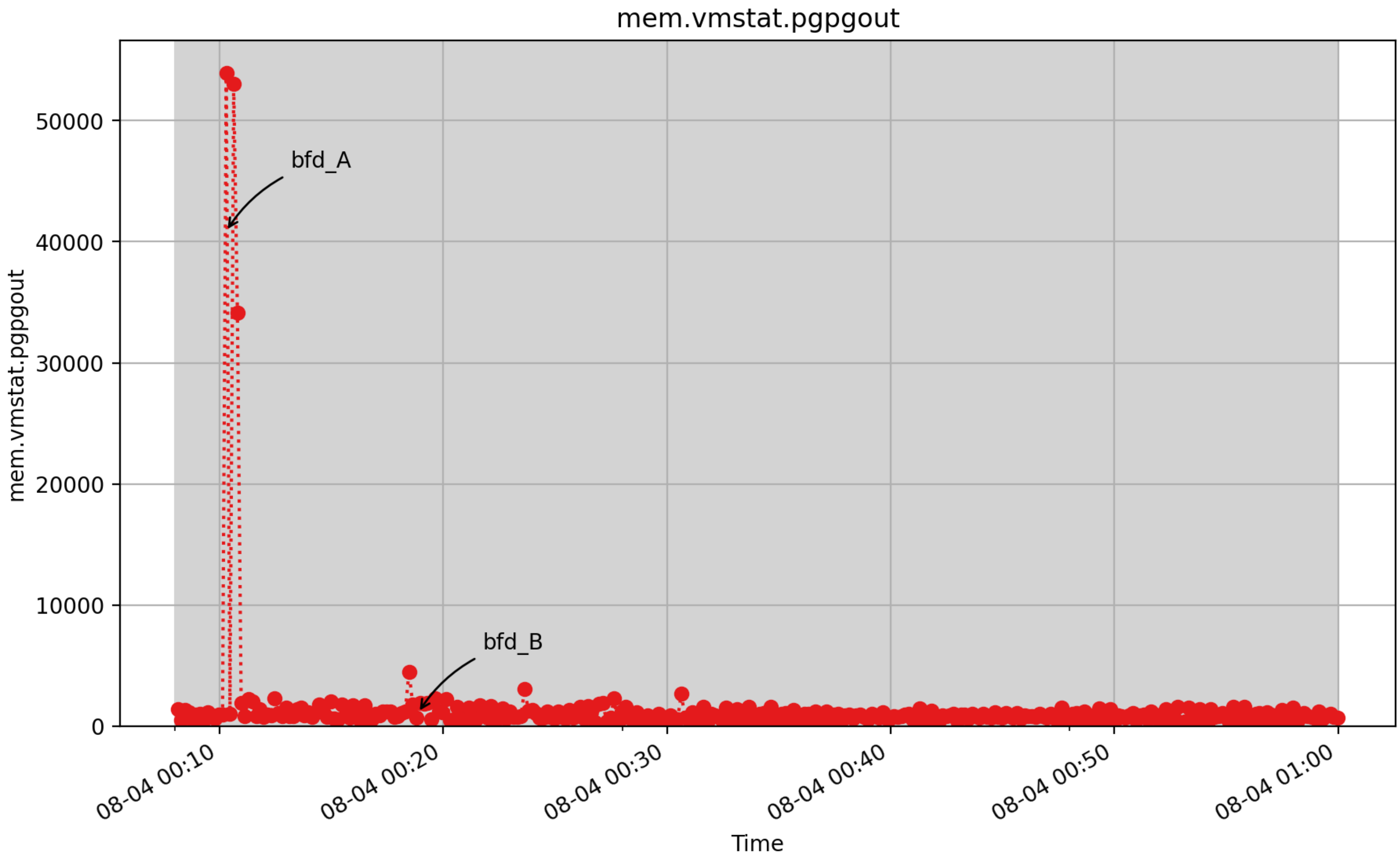
mem.vmstat.pgmigrate_fail: count of unsuccessful NUMA page migrations (count - U64) - *rate converted*



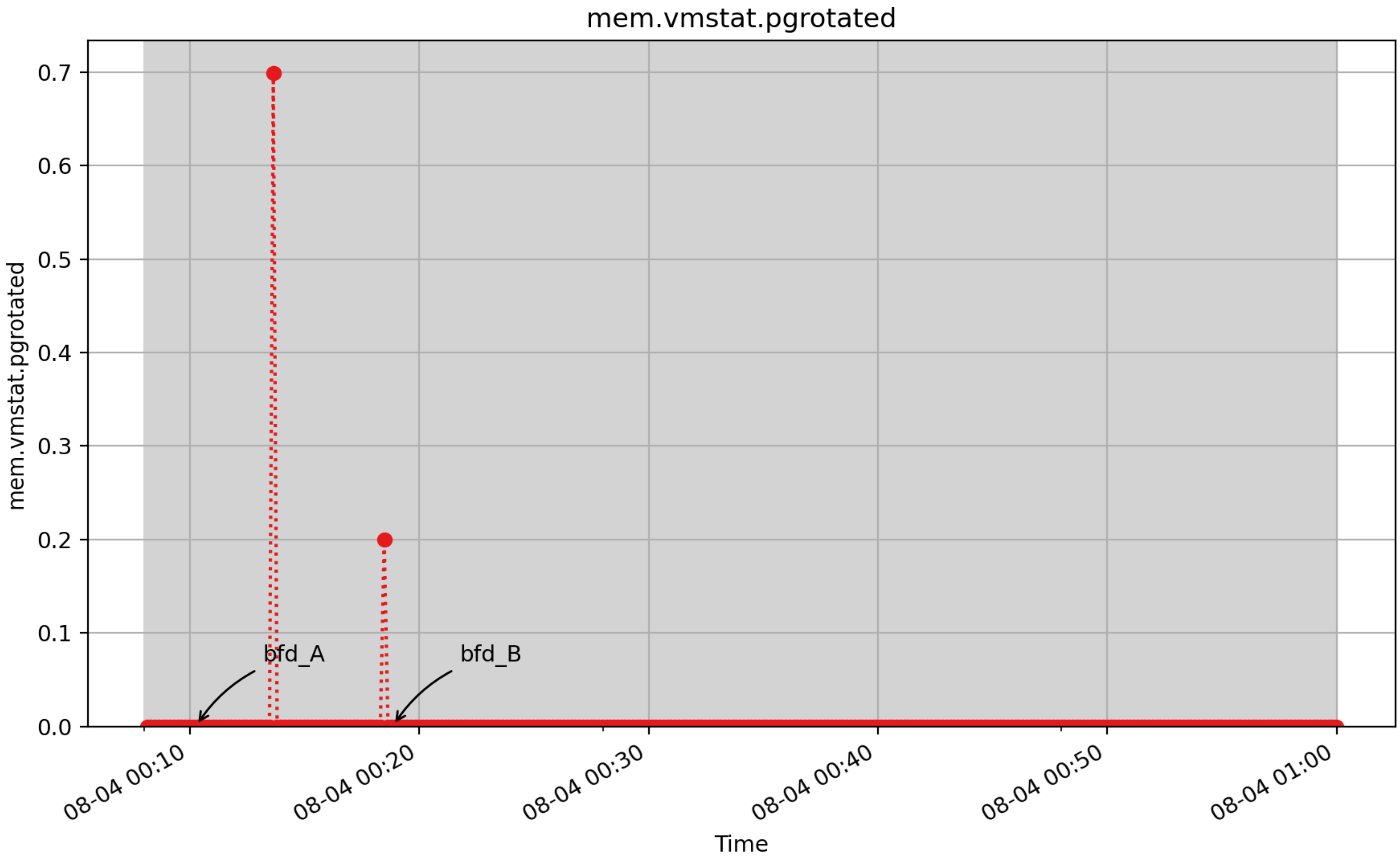
mem.vmstat.pgmigrate_success: count of successful NUMA page migrations (count - U64) - *rate converted*



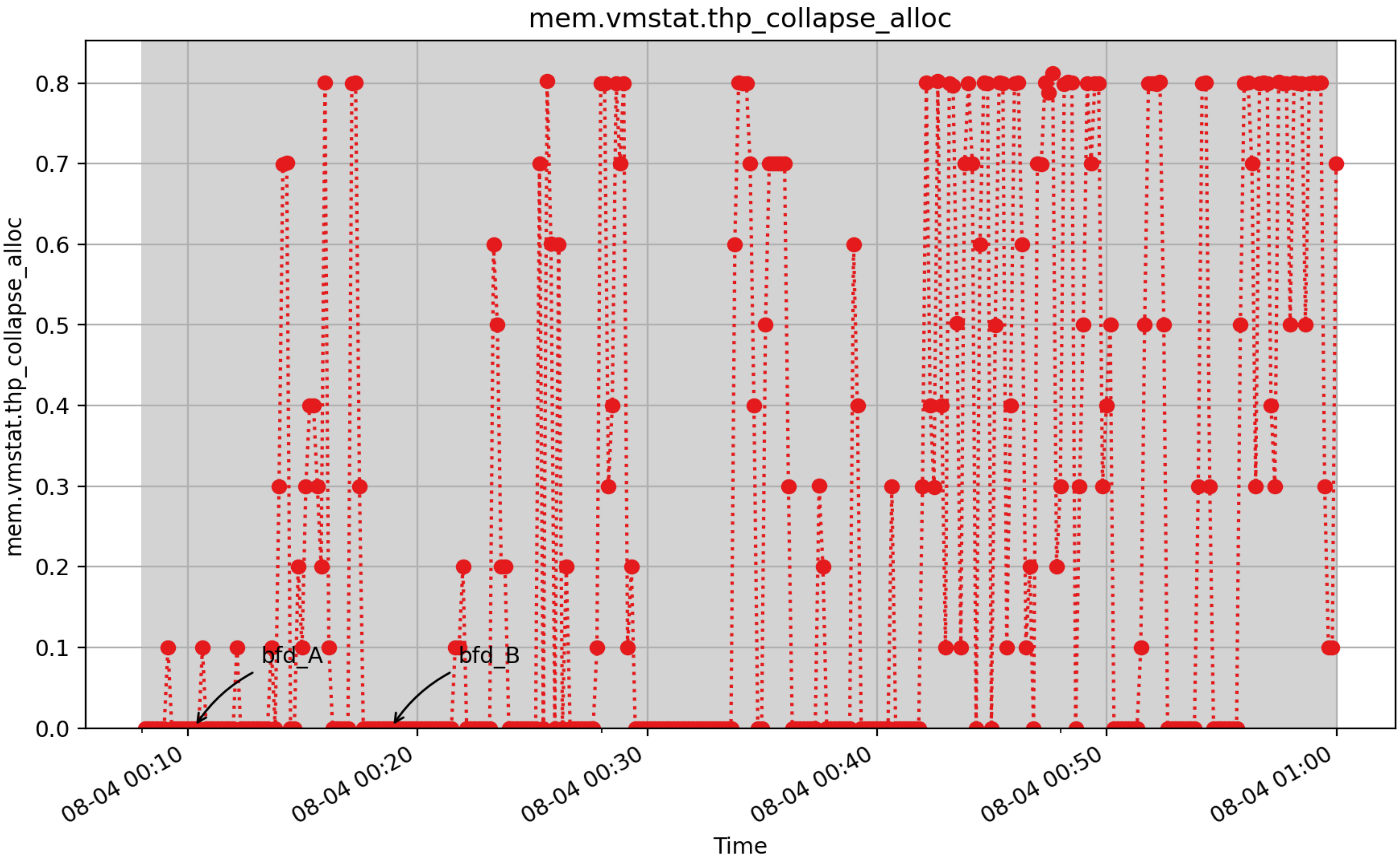
mem.vmstat.pgpgin: Count of page in operations since boot, from /proc/vmstat (count - U64) - *rate converted*



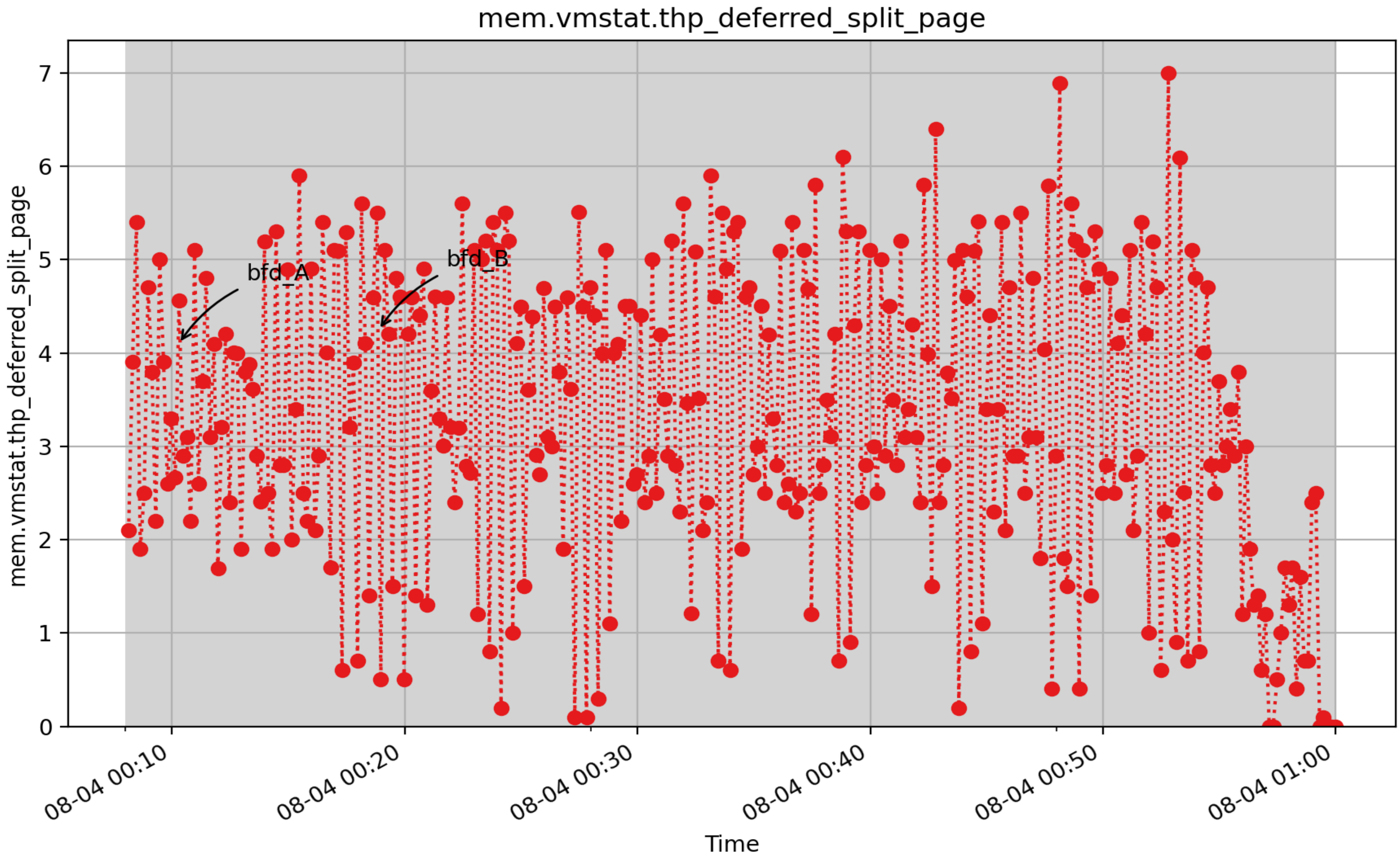
mem.vmstat.pgpgout: Count of page out operations since boot, from /proc/vmstat (count - U64) - *rate converted*



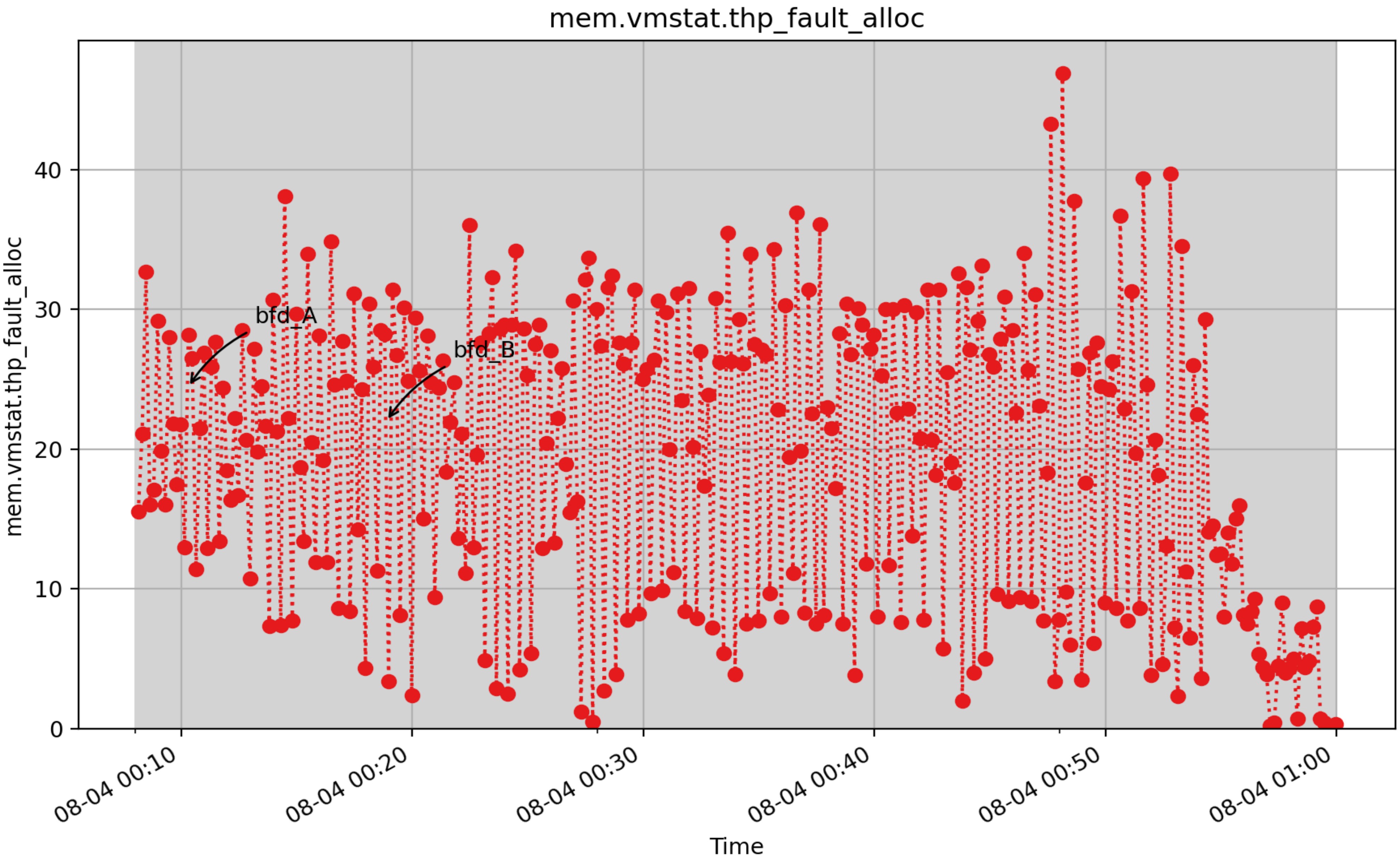
mem.vmstat.pgrotated: Count of pages rotated to tail of the LRU since boot, from /proc/vmstat (count - U64) - *rate converted*



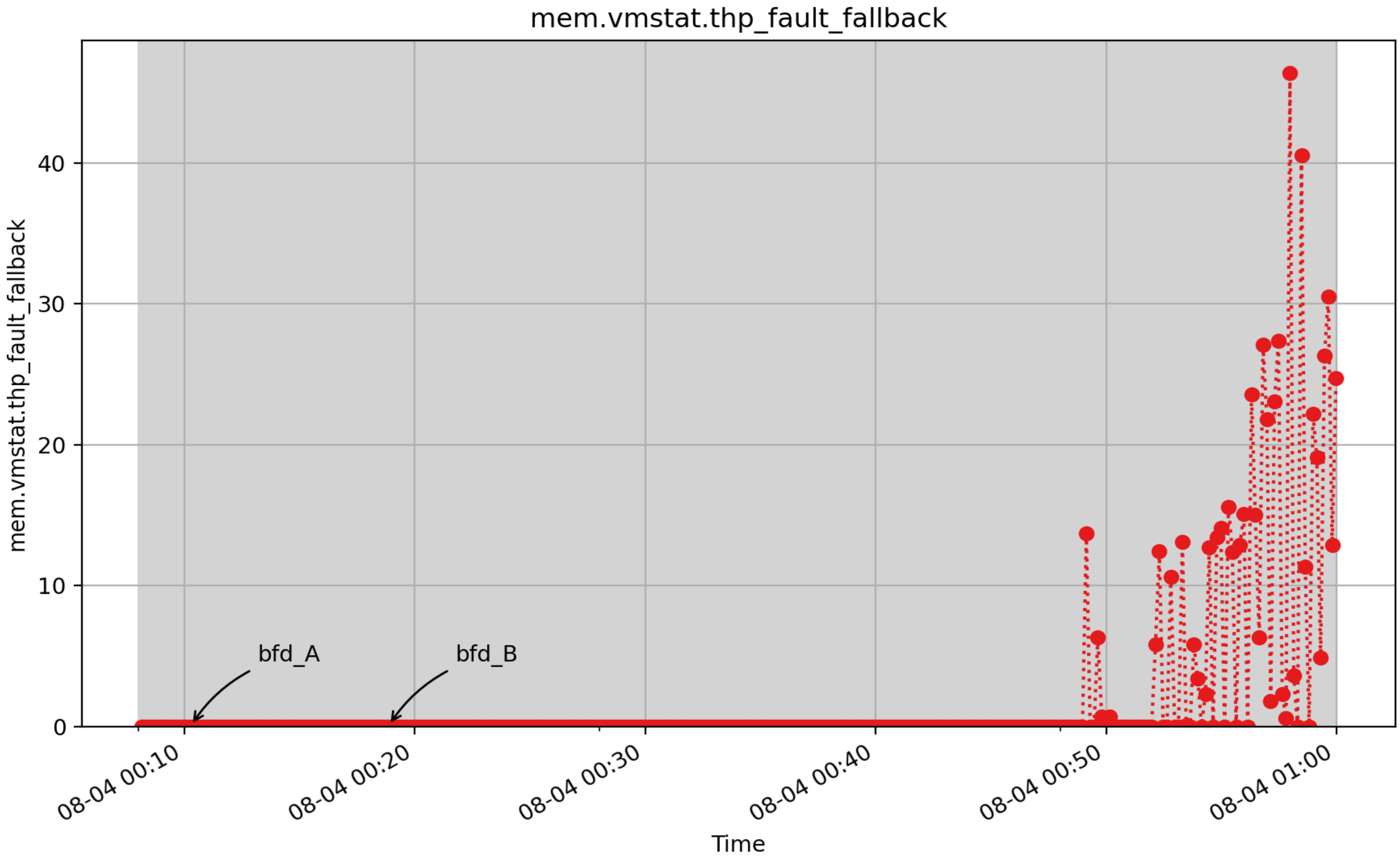
mem.vmstat.thp_collapse_alloc: transparent huge page collapse allocations (count - U64) - *rate converted*



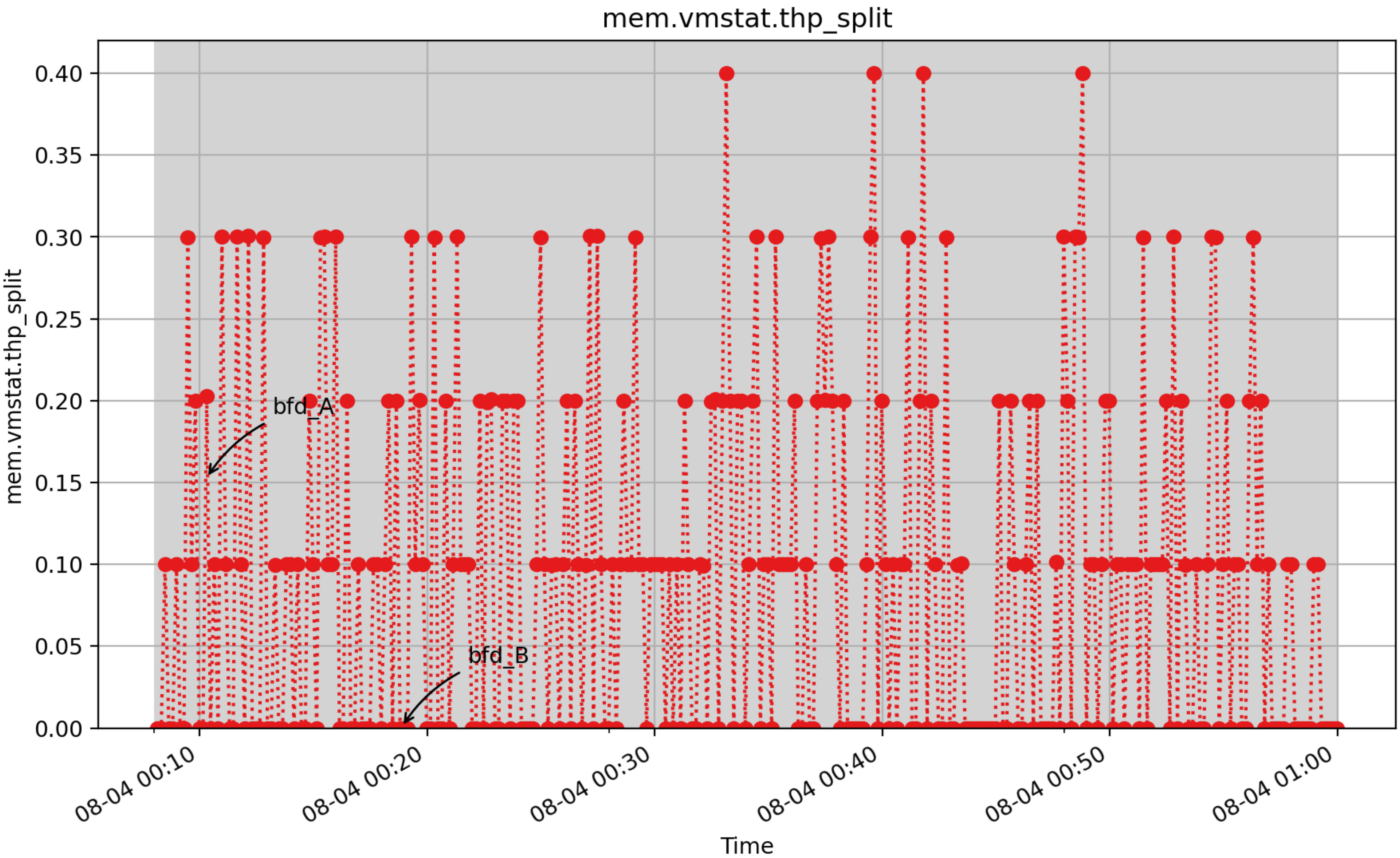
mem.vmstat.thp_deferred_split_page: count of huge page enqueues for splitting (count - U64) - *rate converted*



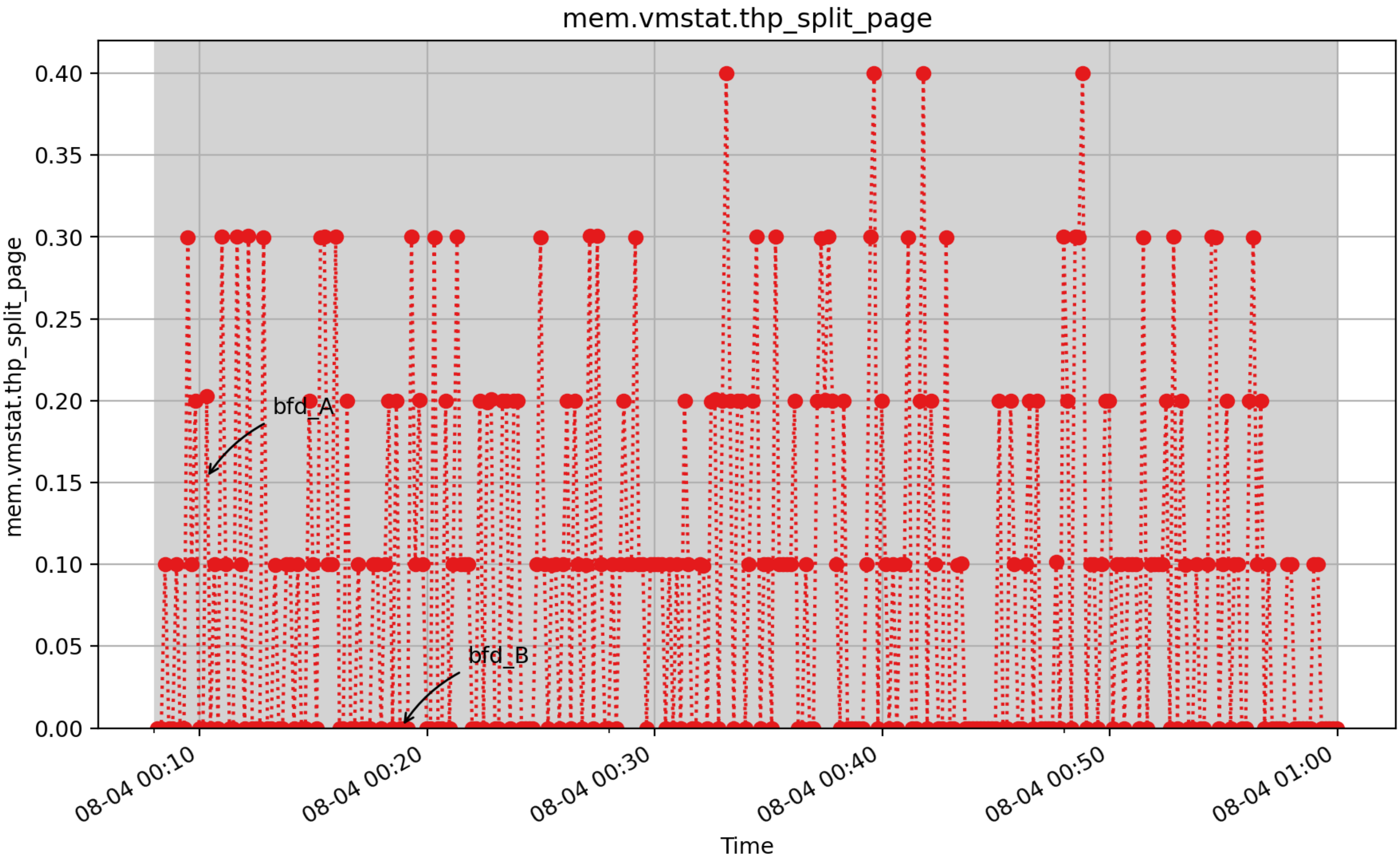
mem.vmstat.thp_fault_alloc: transparent huge page fault allocations (count - U64) - *rate converted*



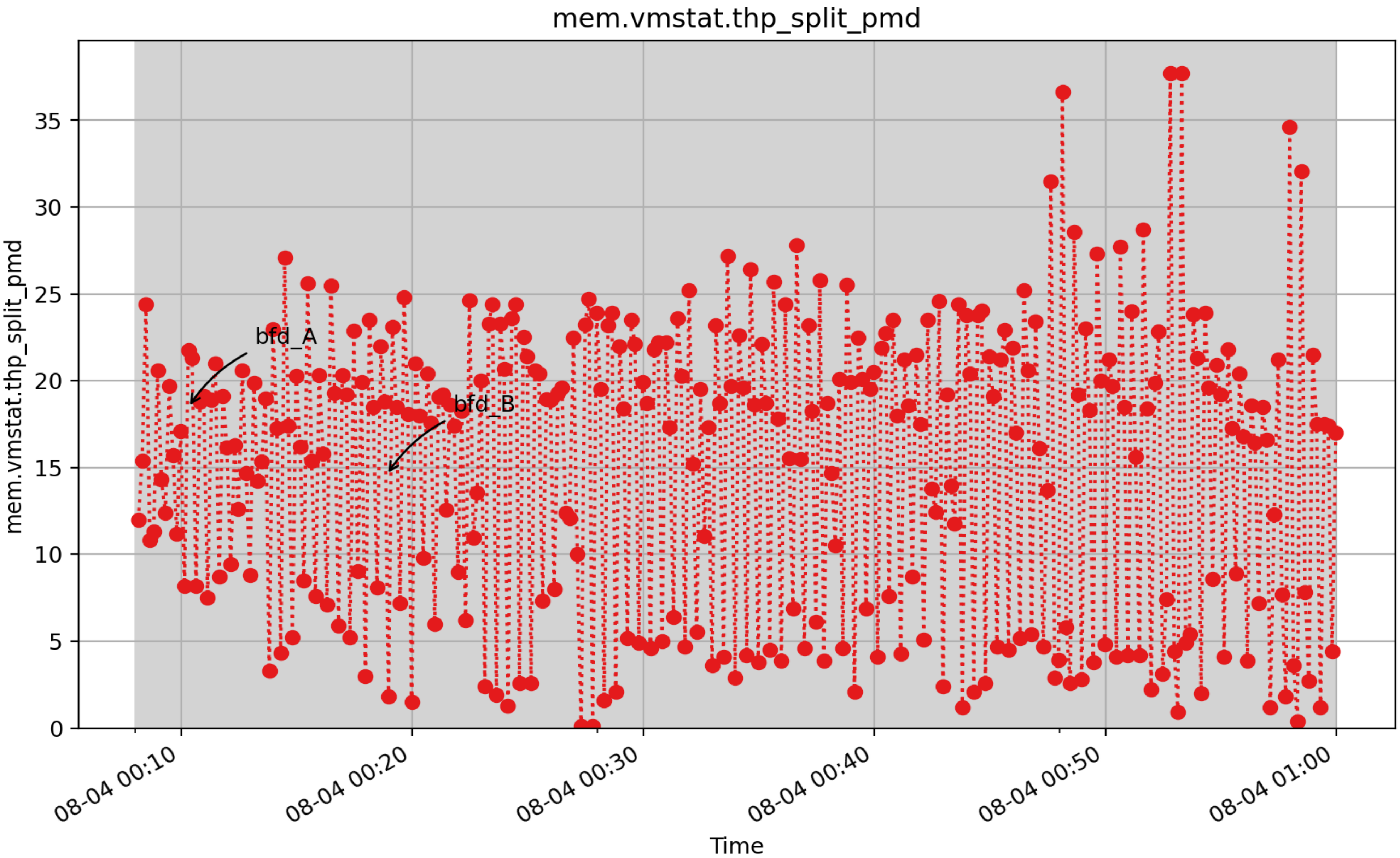
mem.vmstat.thp_fault_fallback: transparent huge page fault fallbacks (count - U64) - *rate converted*



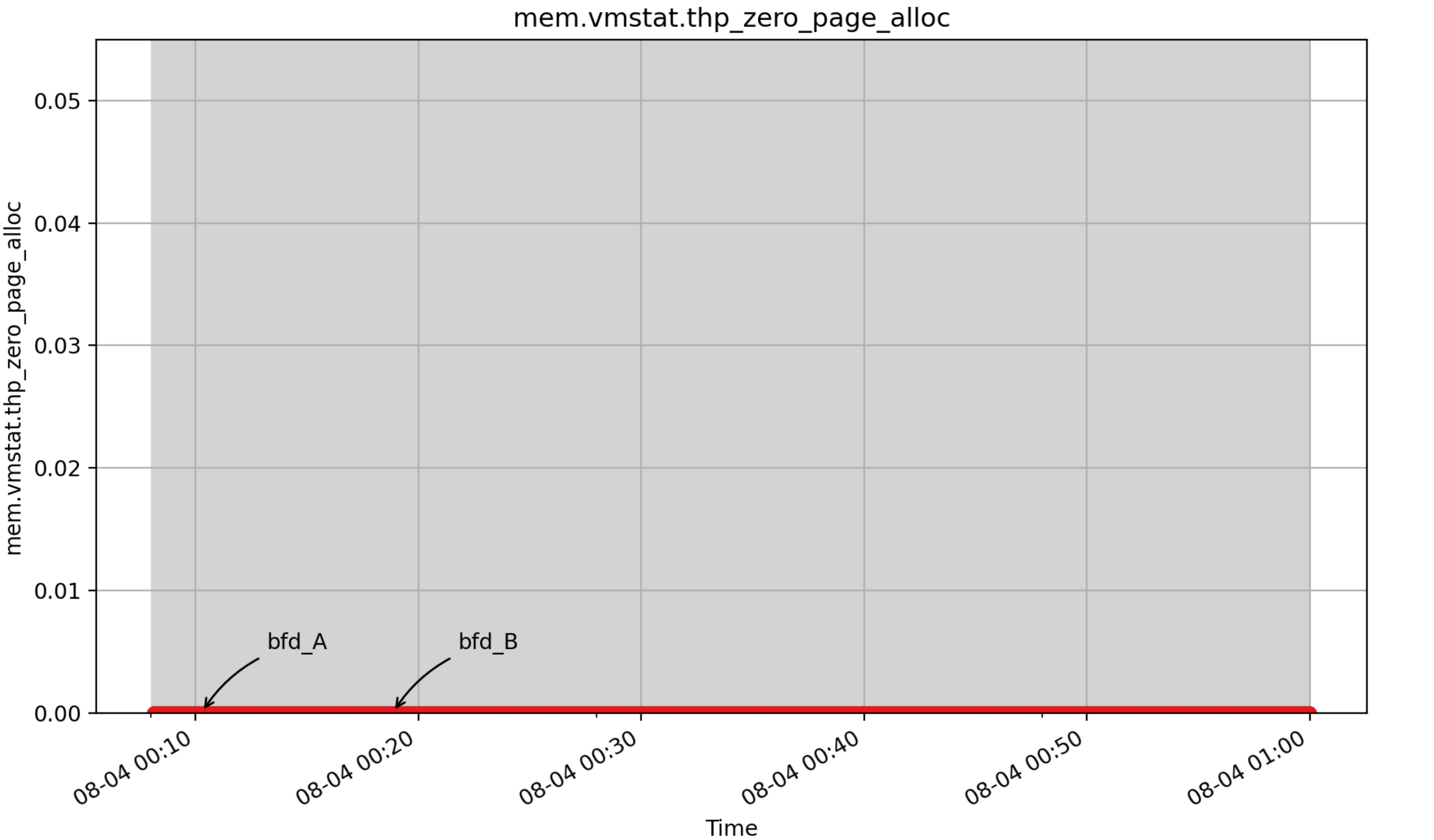
mem.vmstat.thp_split: count of transparent huge page splits (count - U64) - *rate converted*



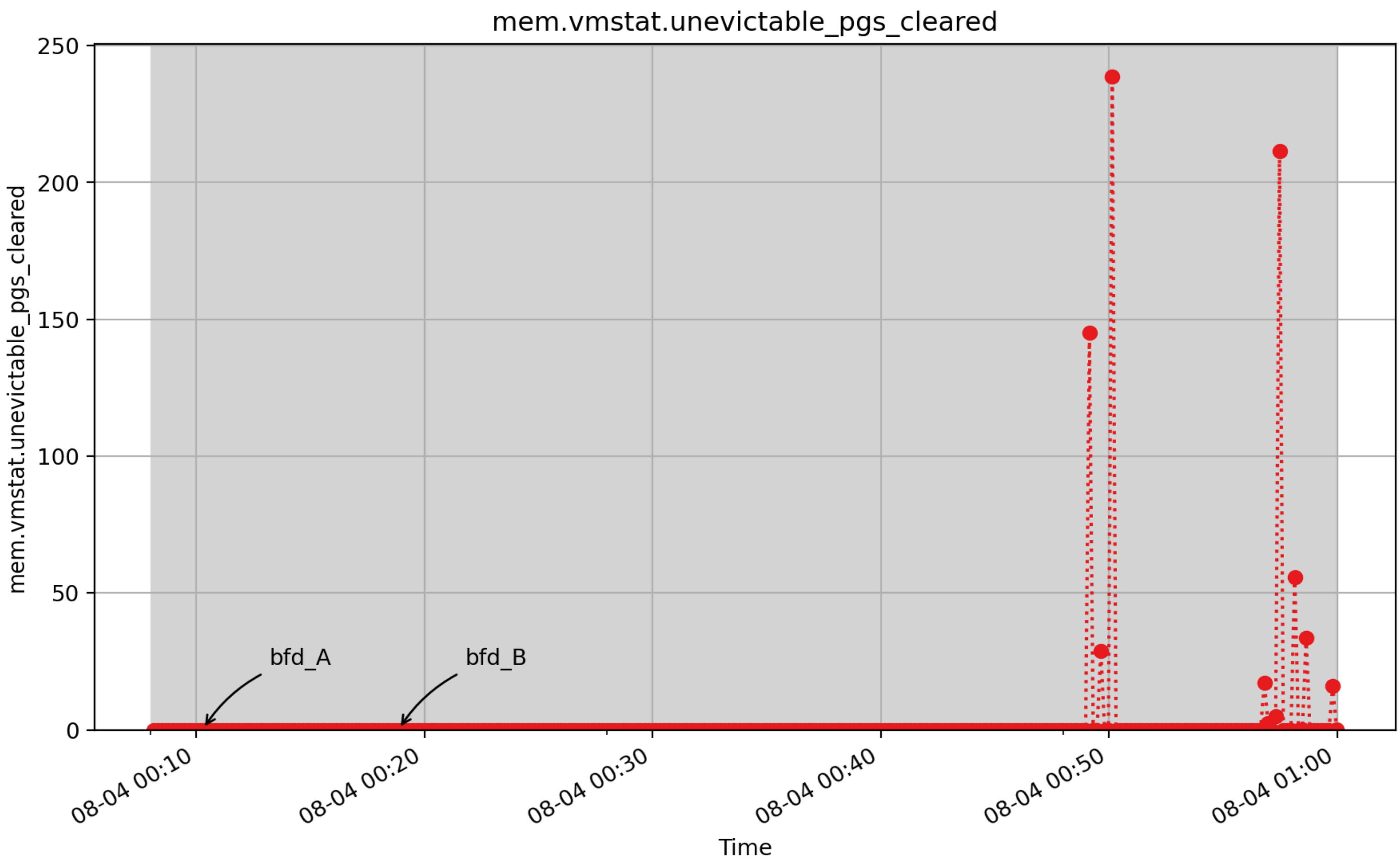
mem.vmstat.thp_split_page: count of huge page splits into base pages (count - U64) - *rate converted*



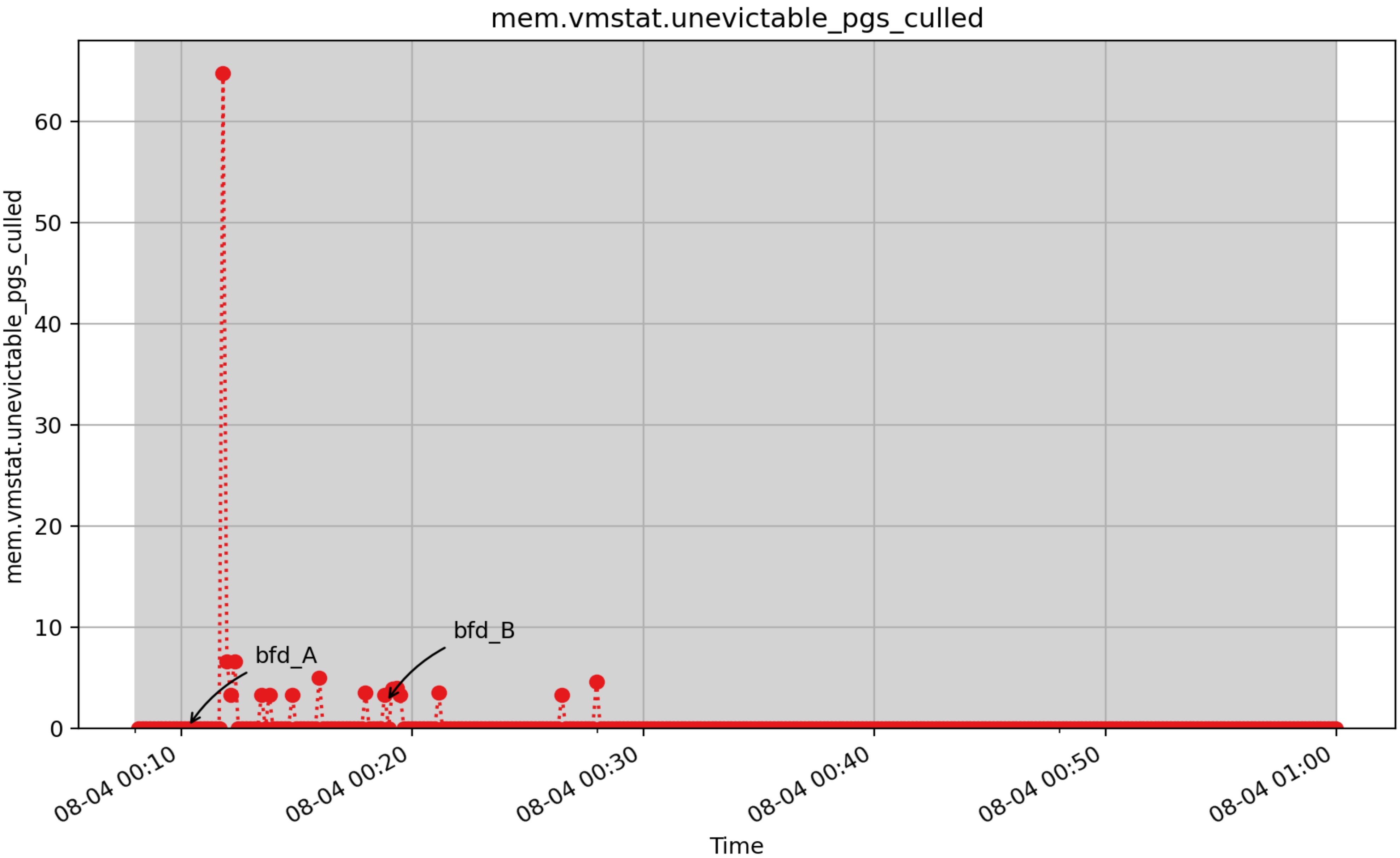
mem.vmstat.thp_split_pmd: This can happen, for instance, when an application calls mprotect() or munmap() on part of huge page. It doesn't split the huge page, only the page table entry. (count - U64) - *rate converted*



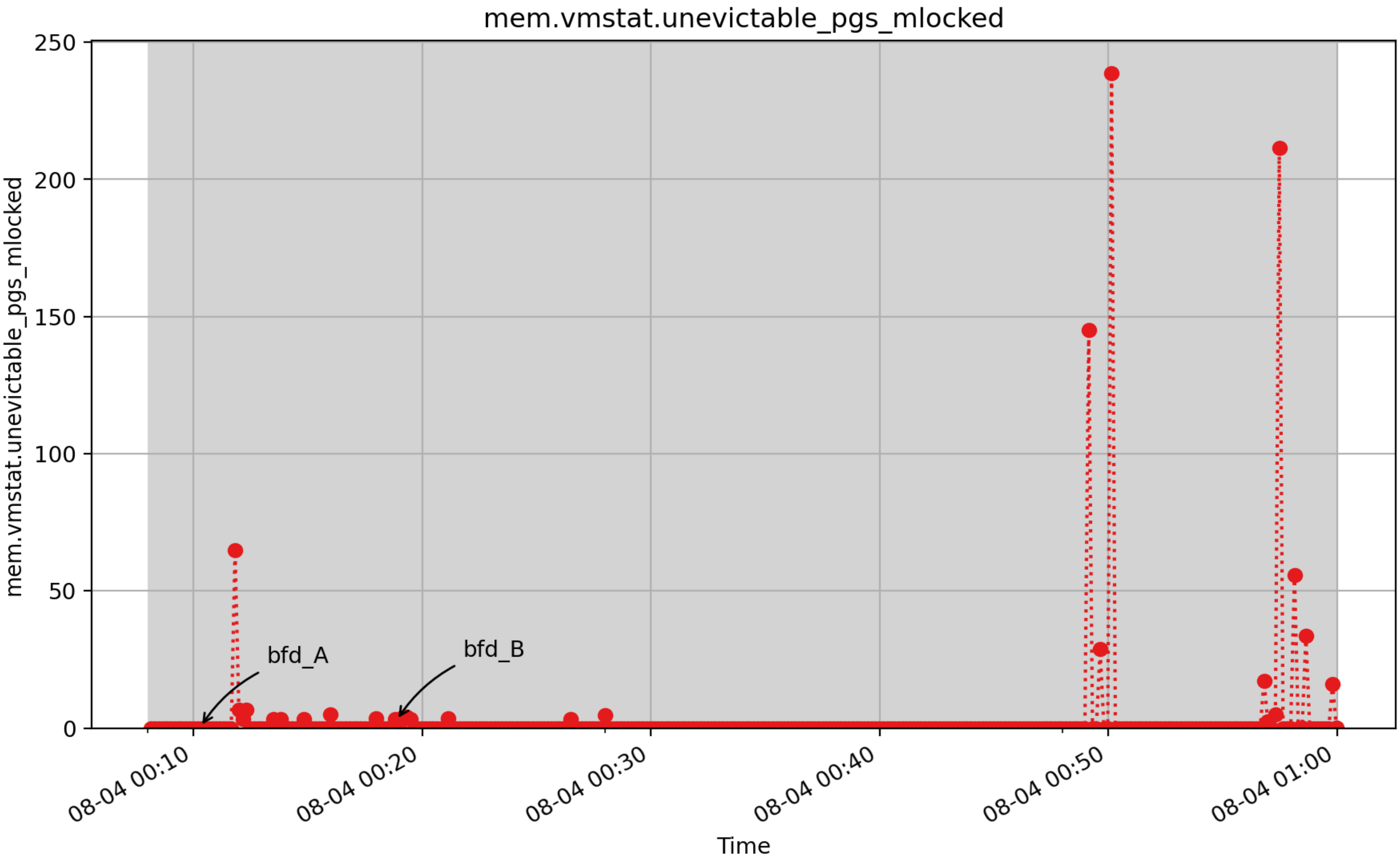
mem.vmstat.thp_zero_page_alloc: count of transparent huge page zeroed page allocations (count - U64) - *rate converted*



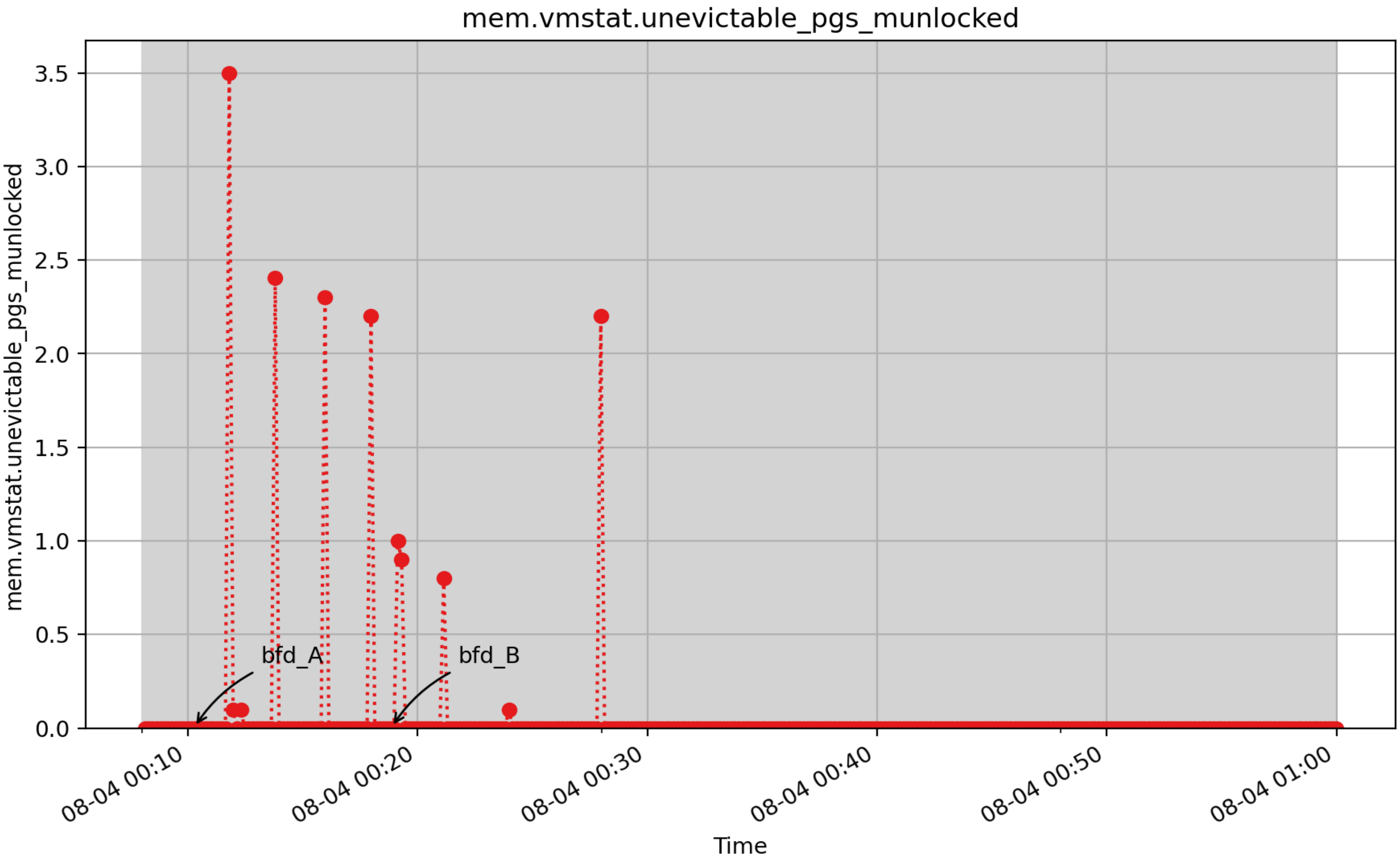
mem.vmstat.unevictable_pgs_cleared: count of unevictable pages cleared (count - U64) - *rate converted*



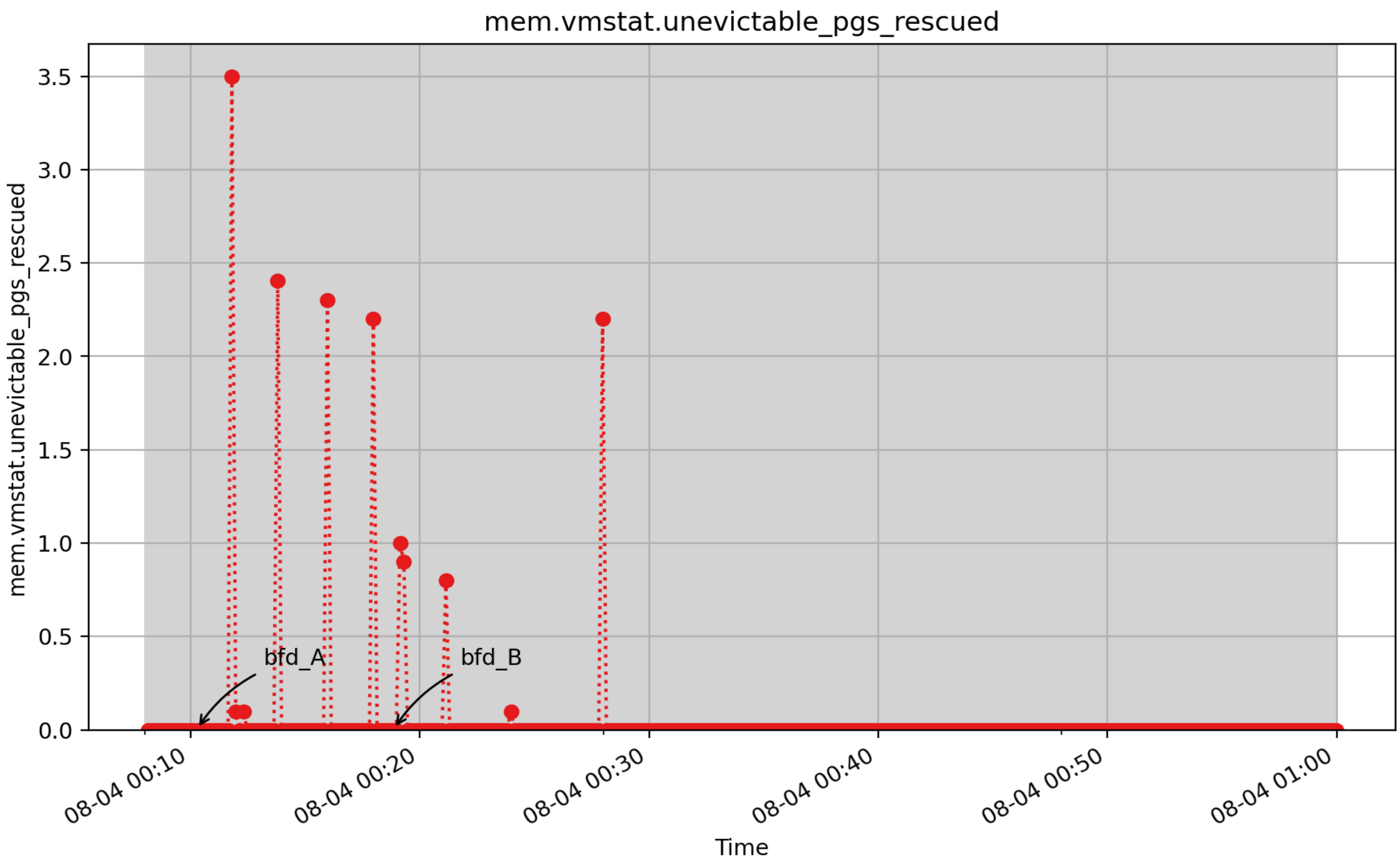
mem.vmstat.unevictable_pgs_culled: count of unevictable pages culled (count - U64) - *rate converted*



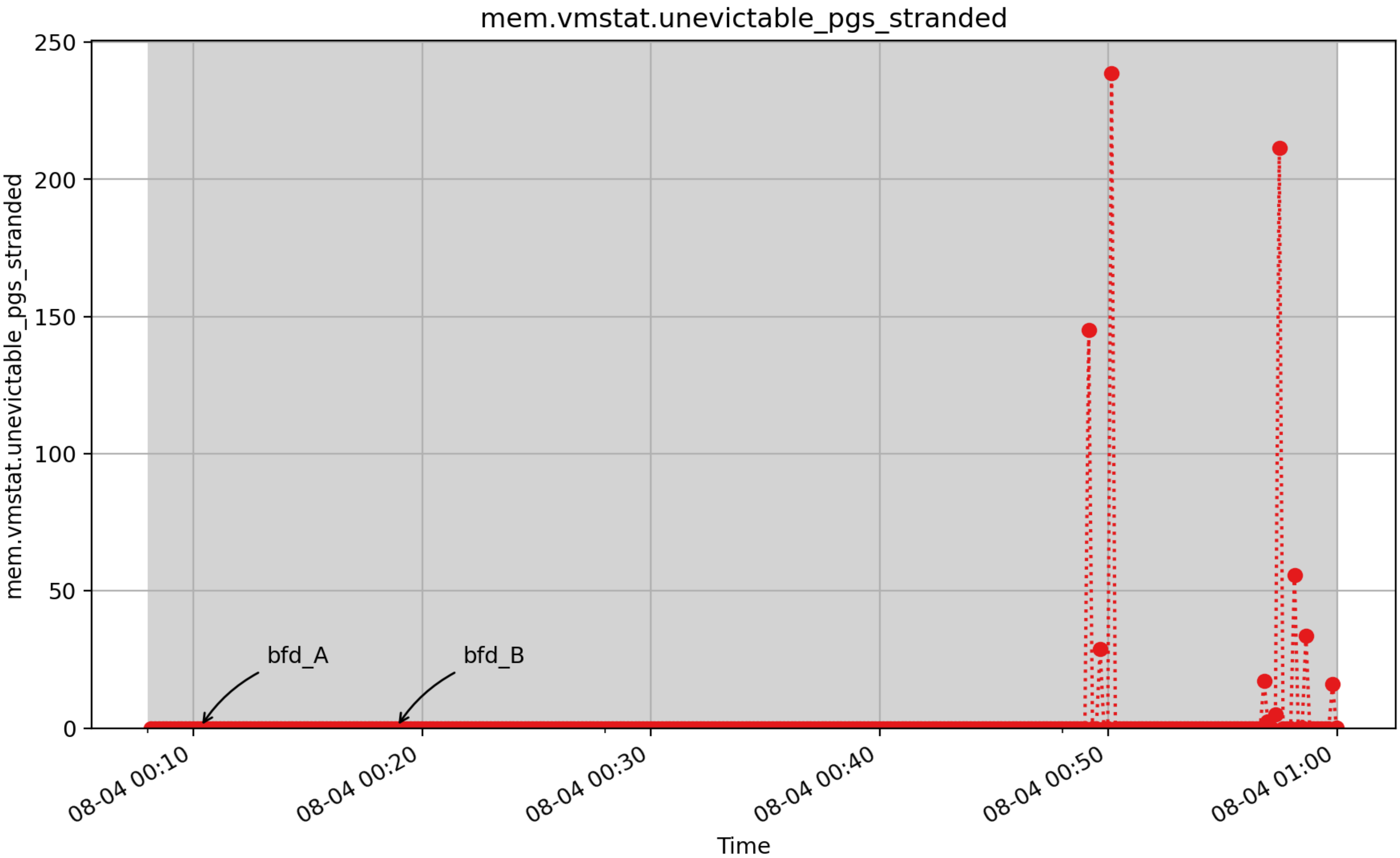
mem.vmstat.unevictable_pgs_mlocked: count of mlocked unevictable pages (count - U64) - *rate converted*



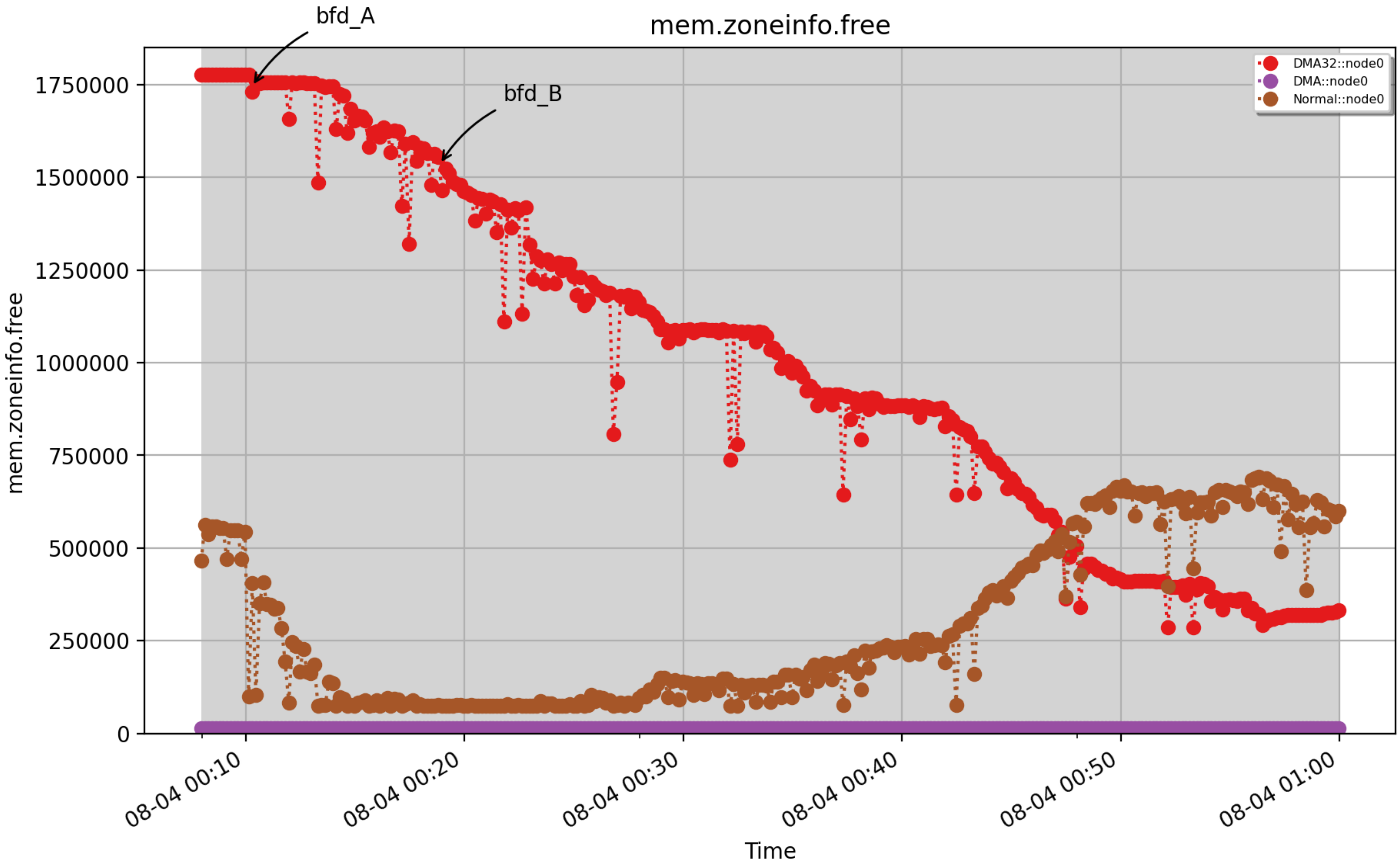
mem.vmstat.unevictable_pgs_munlocked: count of unevictable pages munlocked (count - U64) - *rate converted*



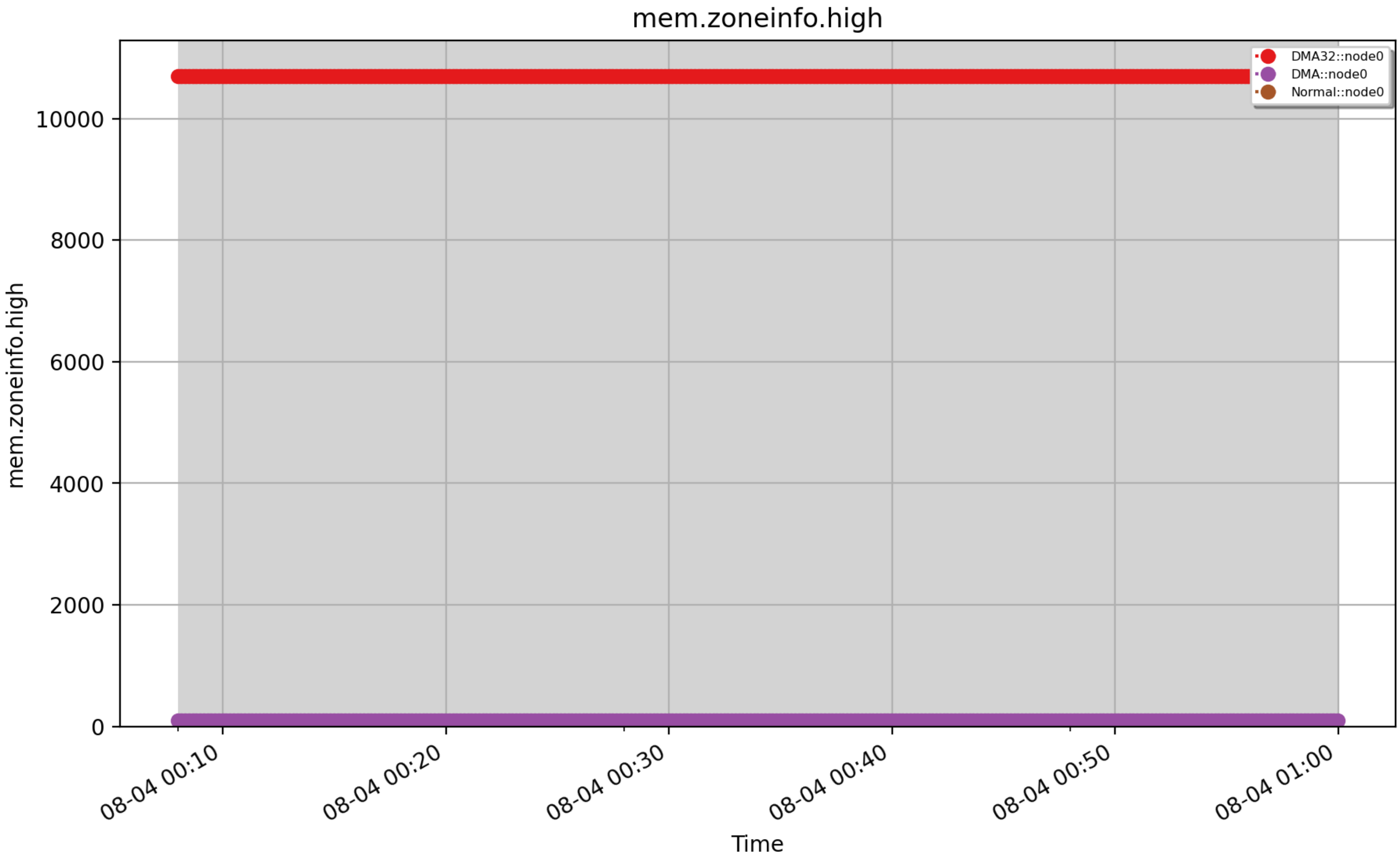
mem.vmstat.unevictable_pgs_rescued: count of unevictable pages rescued (count - U64) - *rate converted*



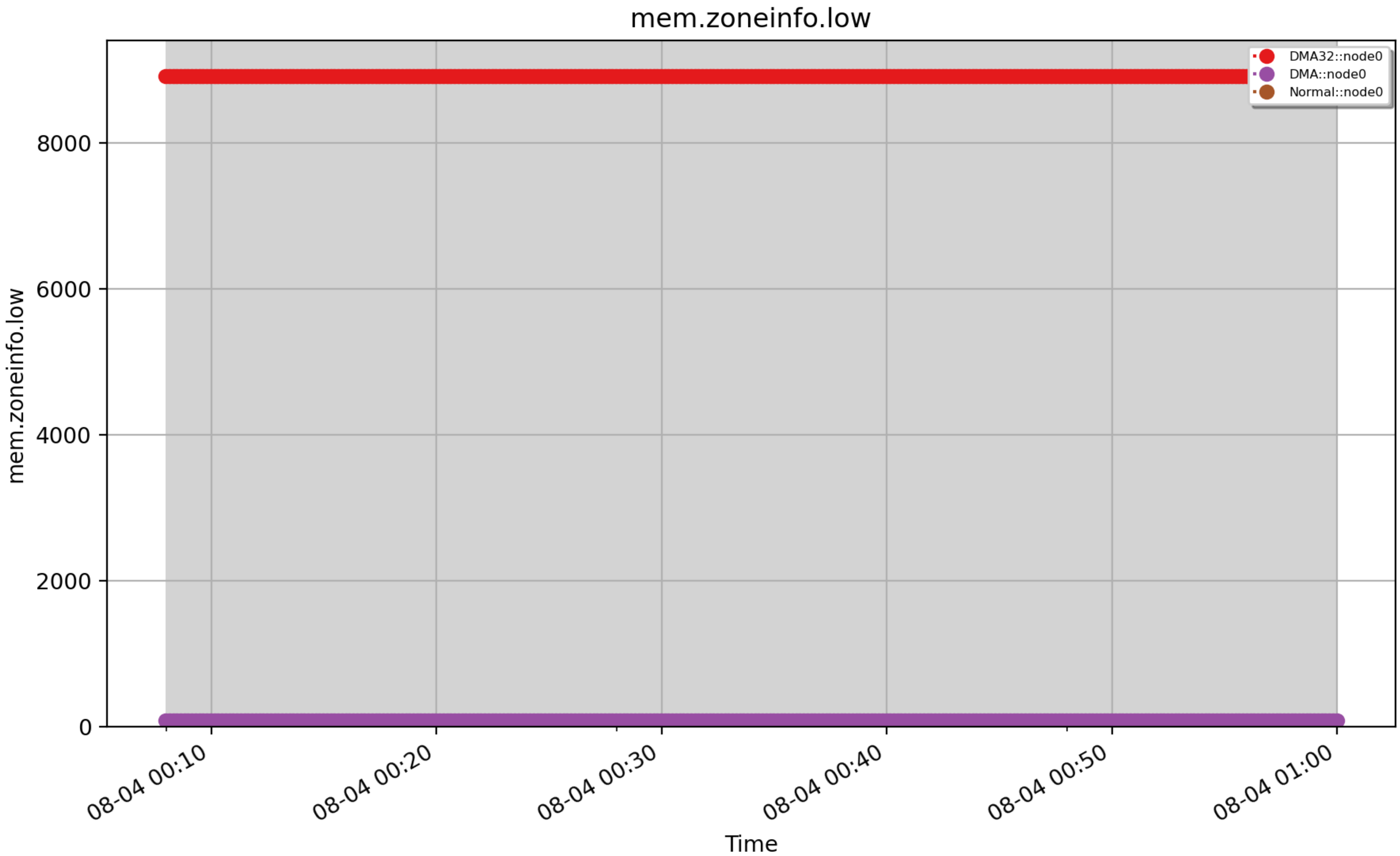
mem.vmstat.unevictable_pgs_stranded: count of unevictable pages stranded (count - U64) - *rate converted*



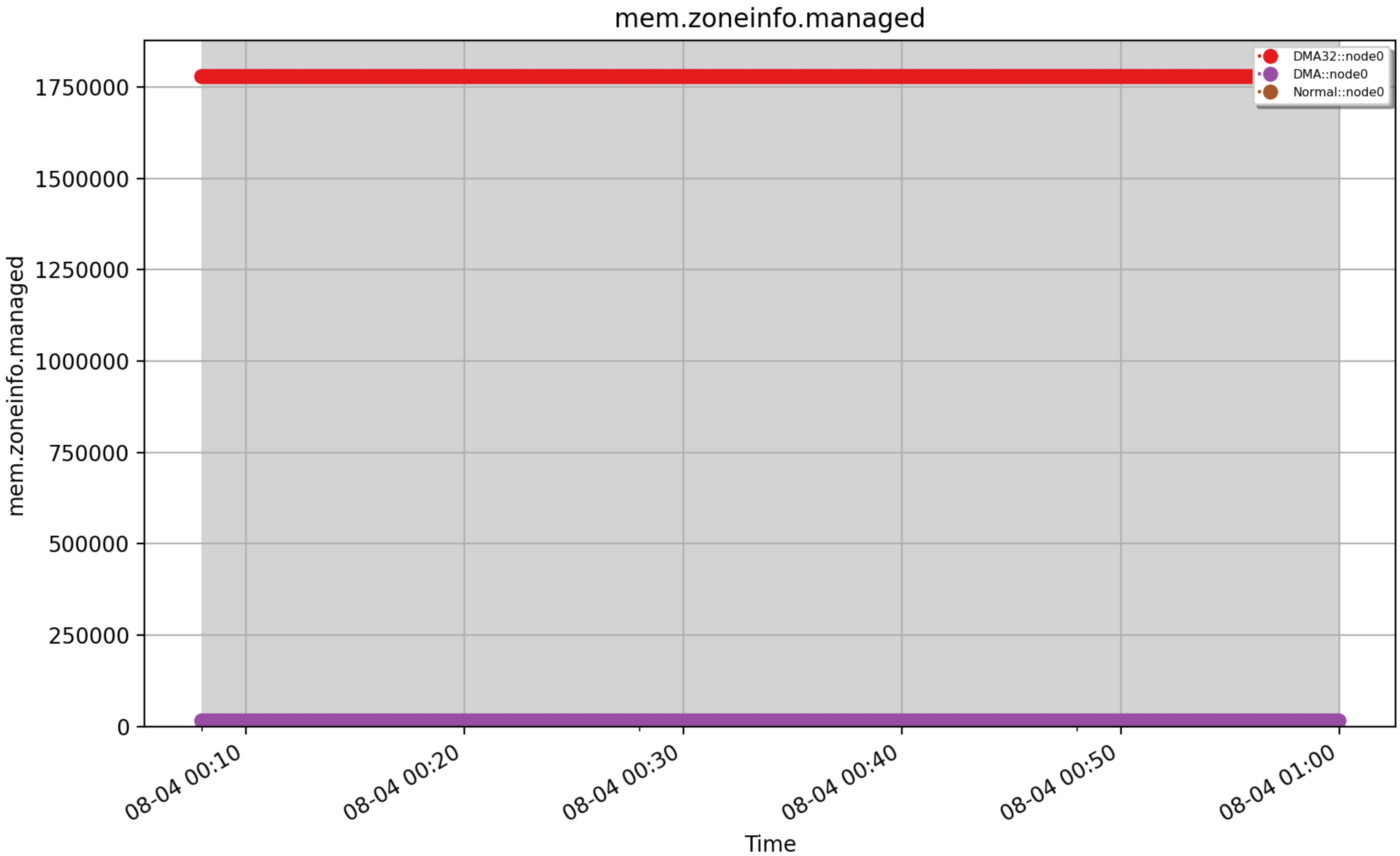
mem.zoneinfo.free: free space in each zone for each NUMA node (Kbyte - U64)



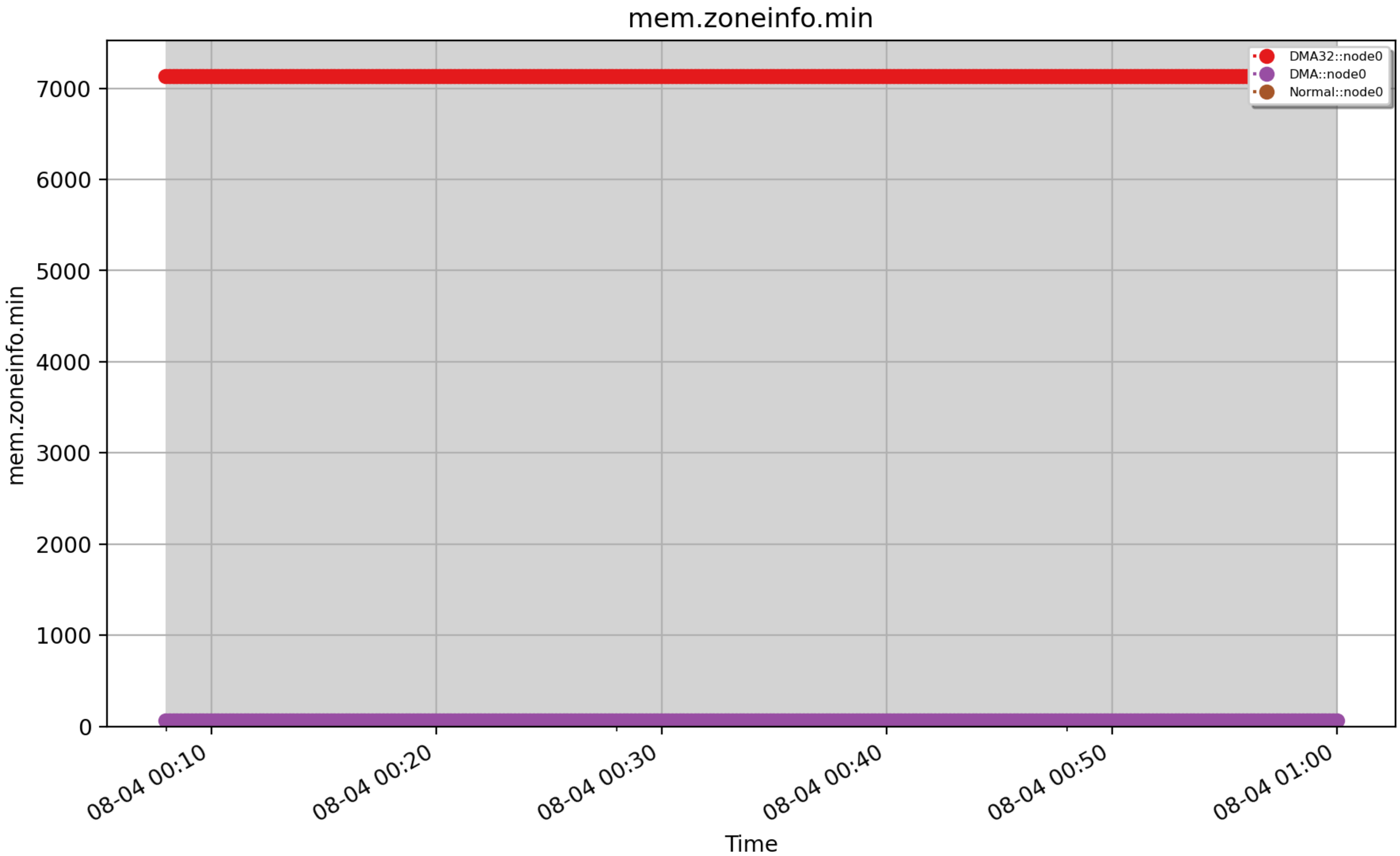
mem.zoneinfo.high: high space in each zone for each NUMA node (Kbyte - U64)



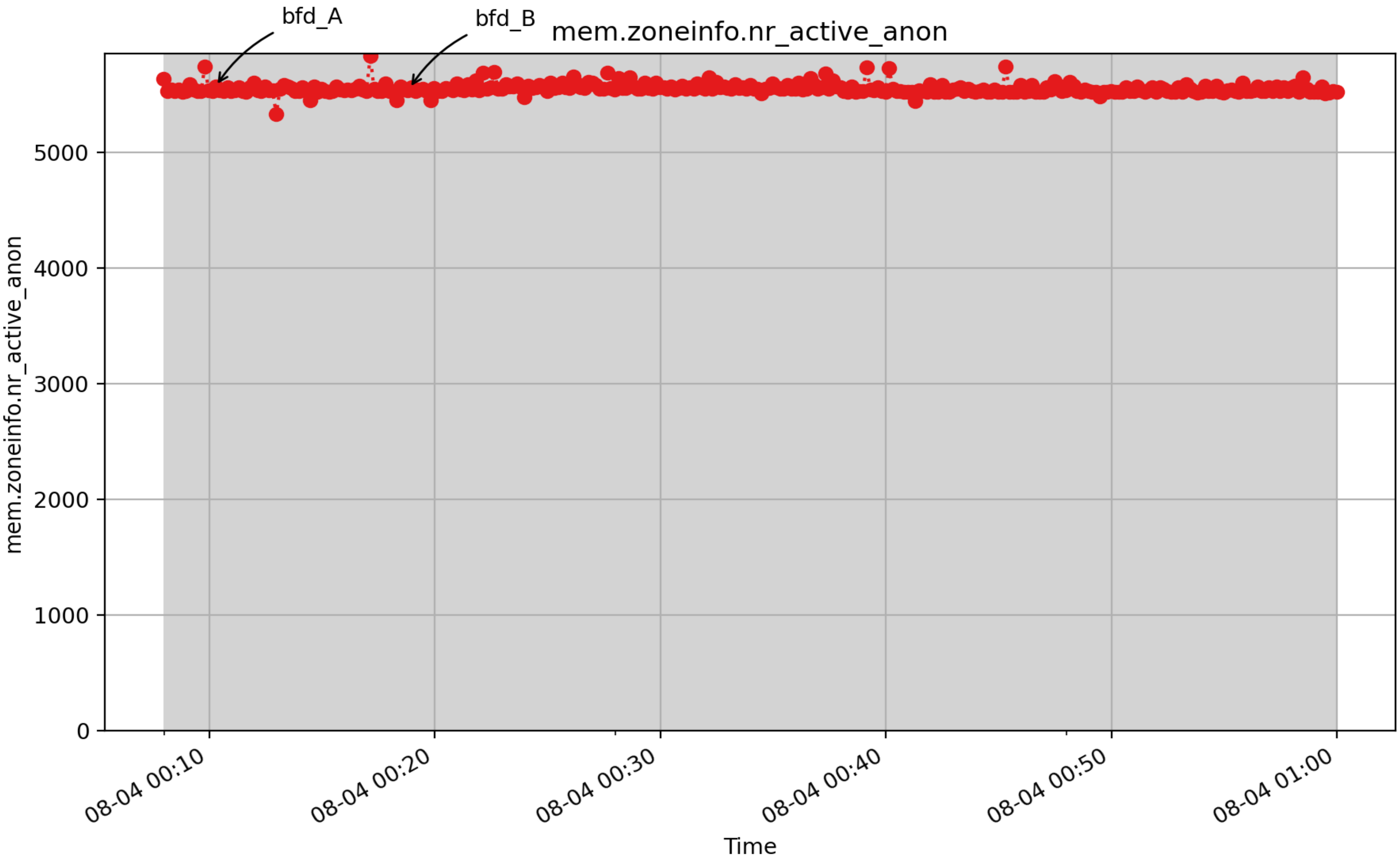
mem.zoneinfo.low: low space in each zone for each NUMA node (Kbyte - U64)



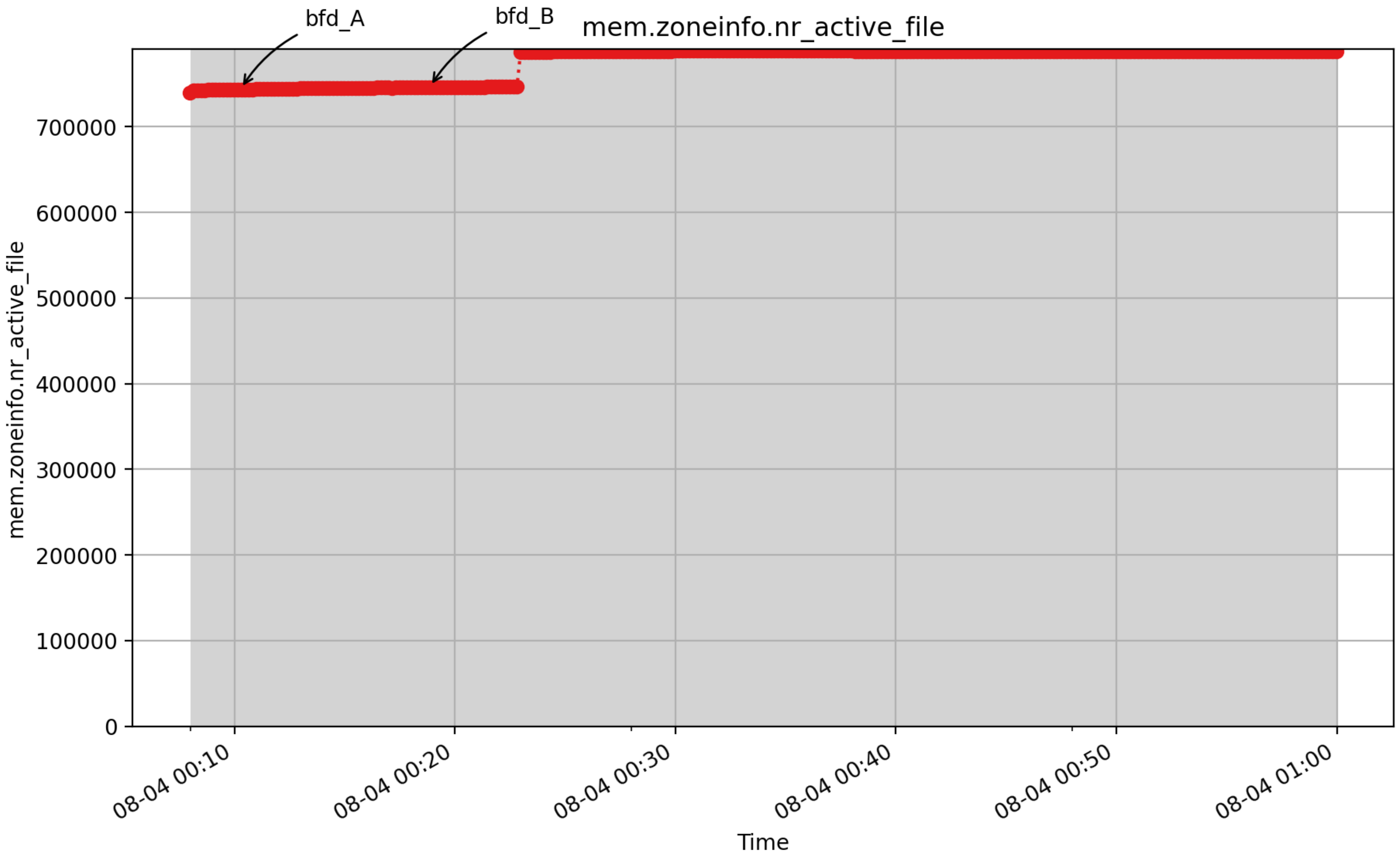
mem.zoneinfo.managed: managed space in each zone for each NUMA node (Kbyte - U64)



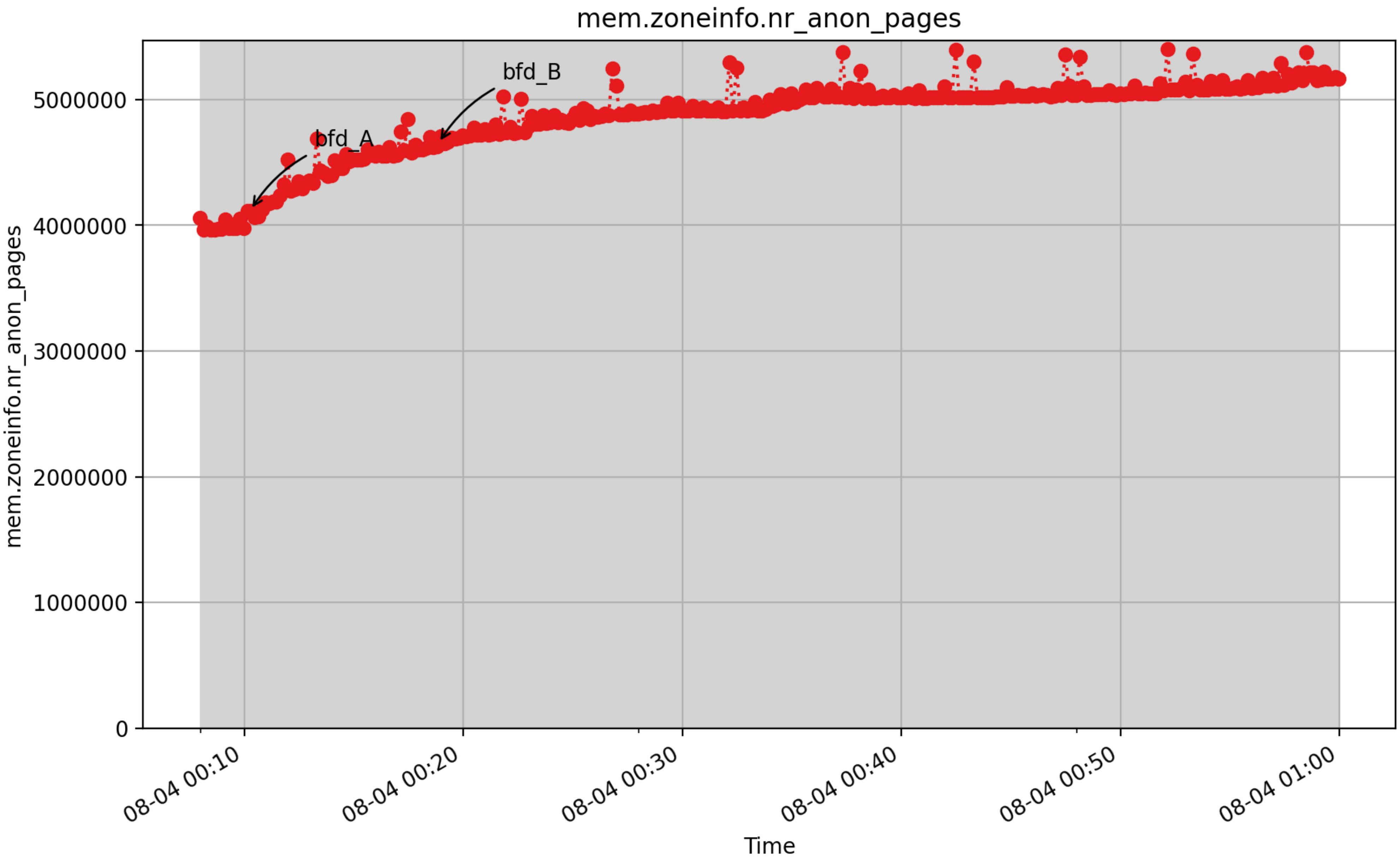
mem.zoneinfo.min: min space in each zone for each NUMA node (Kbyte - U64)



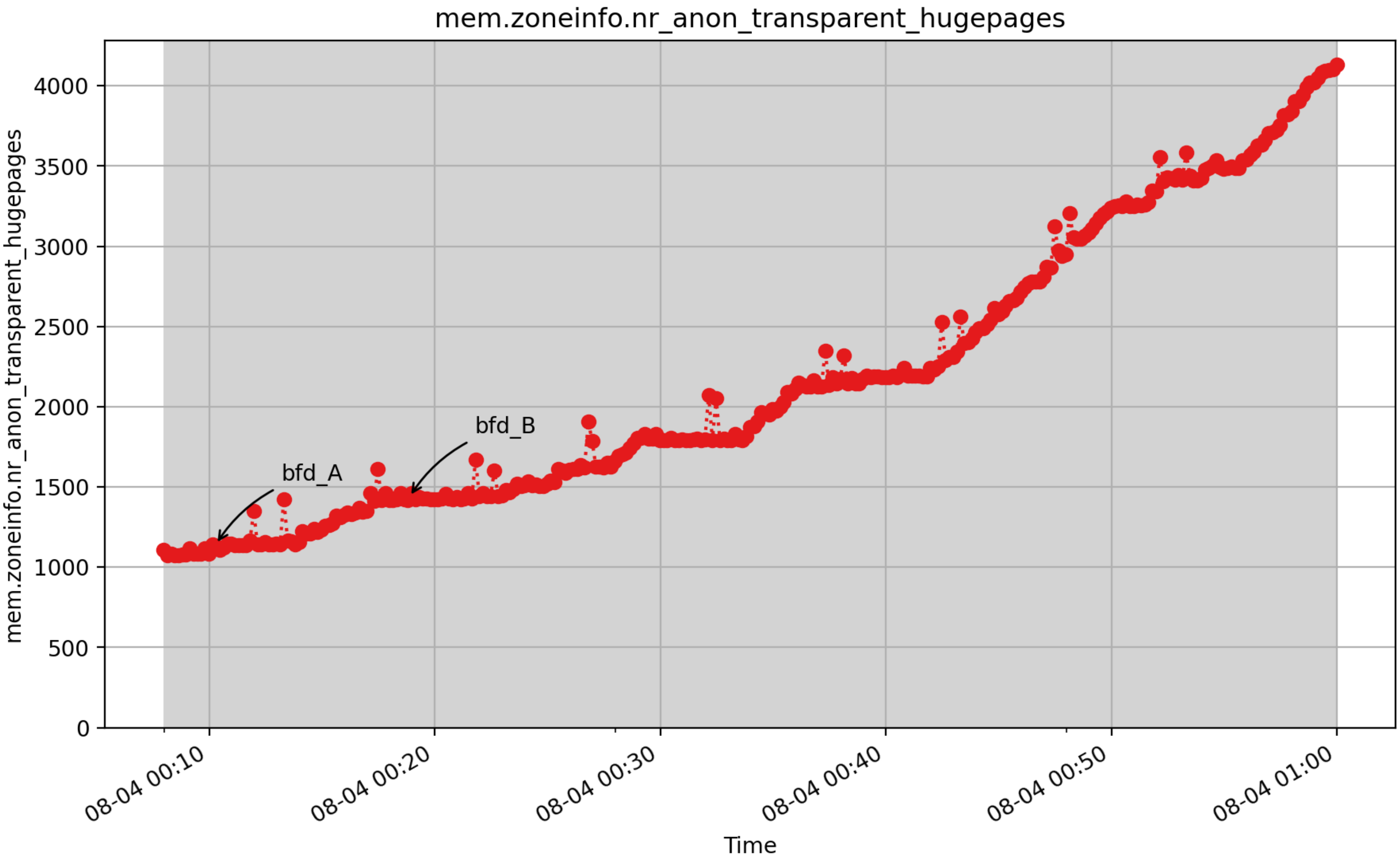
mem.zoneinfo.nr_active_anon: number of active anonymous memory pages in each zone for each NUMA node (Kbyte - U64)



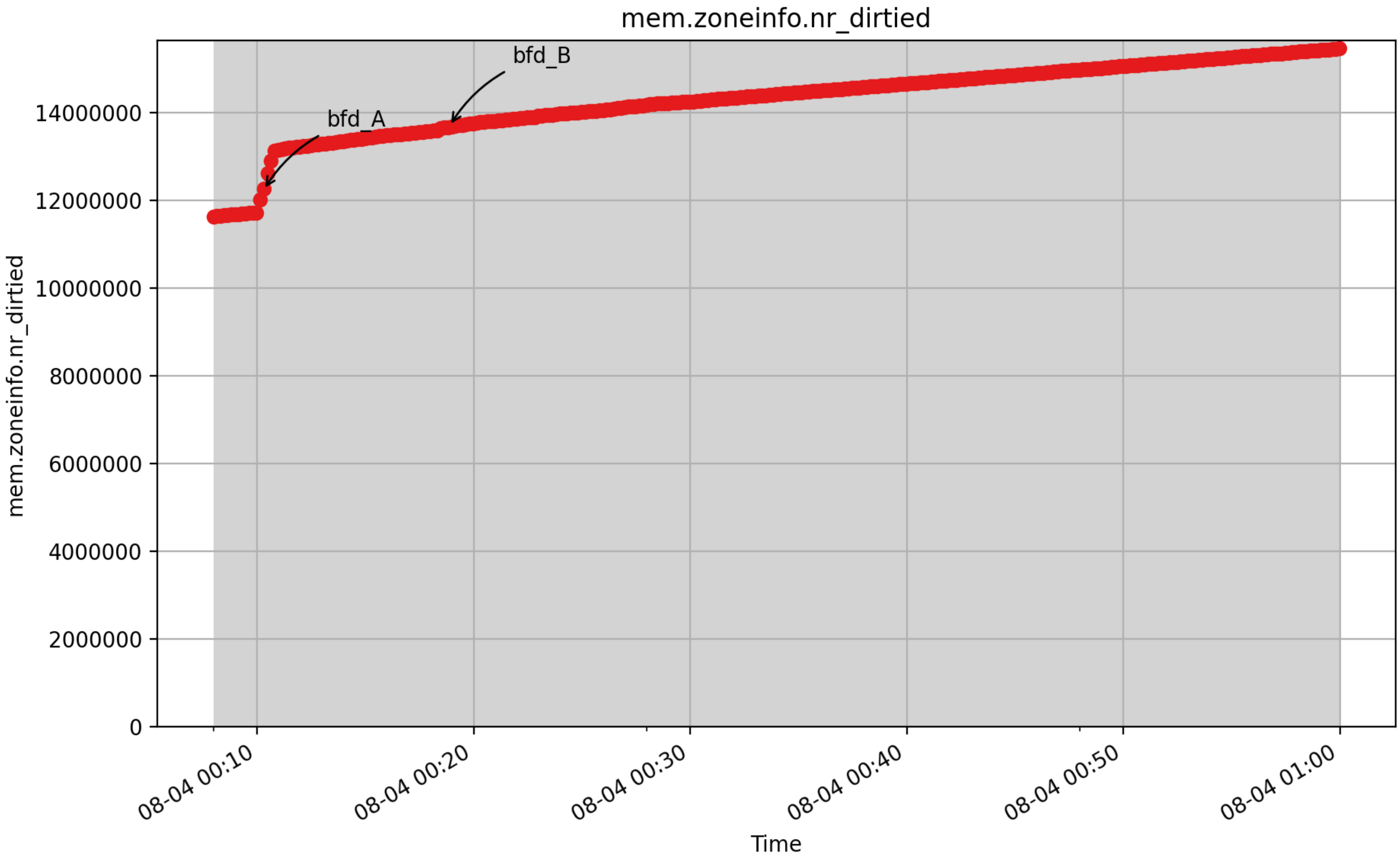
mem.zoneinfo.nr_active_file: number of active file memory memory pages in each zone for each NUMA node (Kbyte - U64)



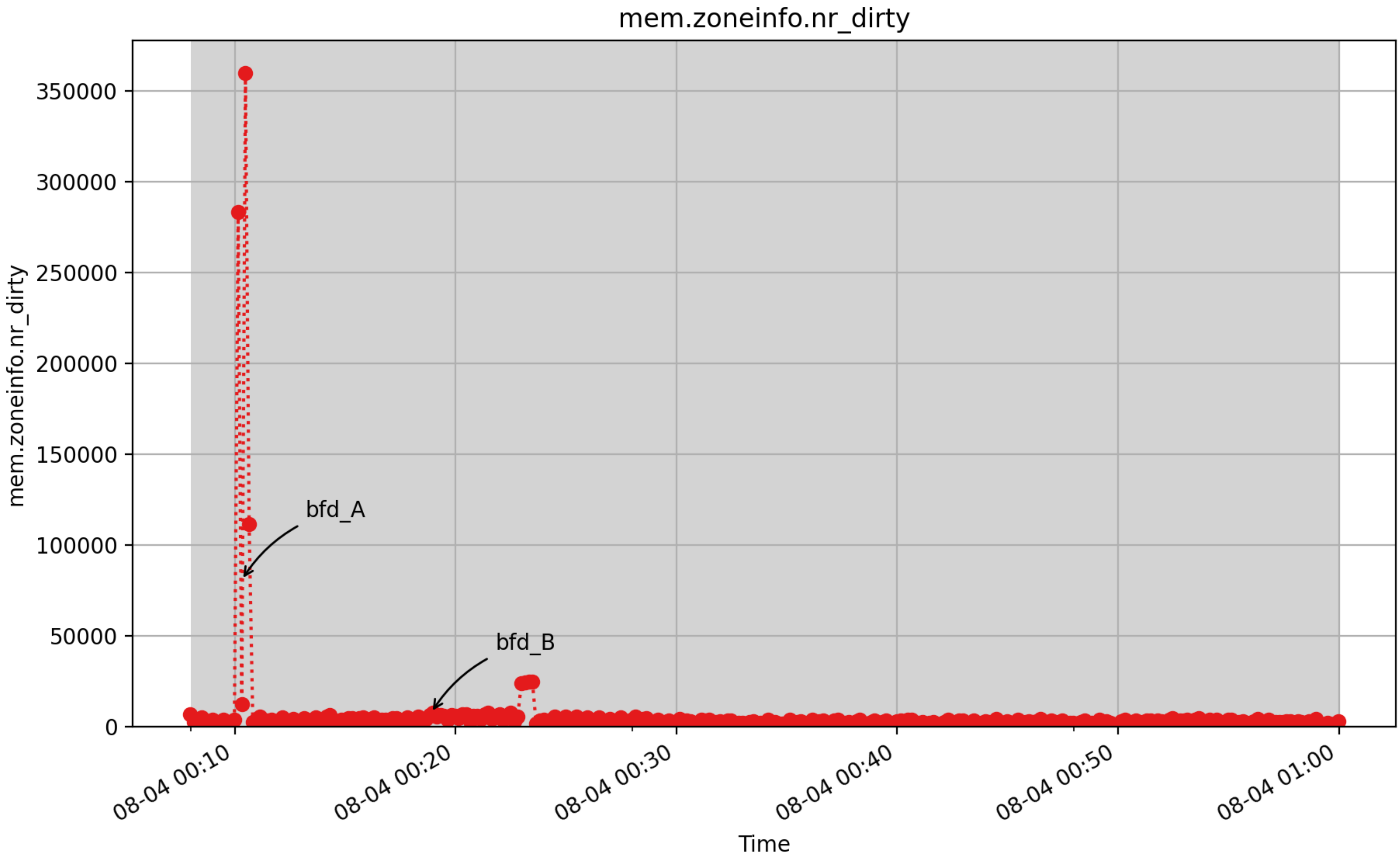
mem.zoneinfo.nr_anon_pages: number of anonymous mapped pagecache pages in each zone for each NUMA node (Kbyte - U64)



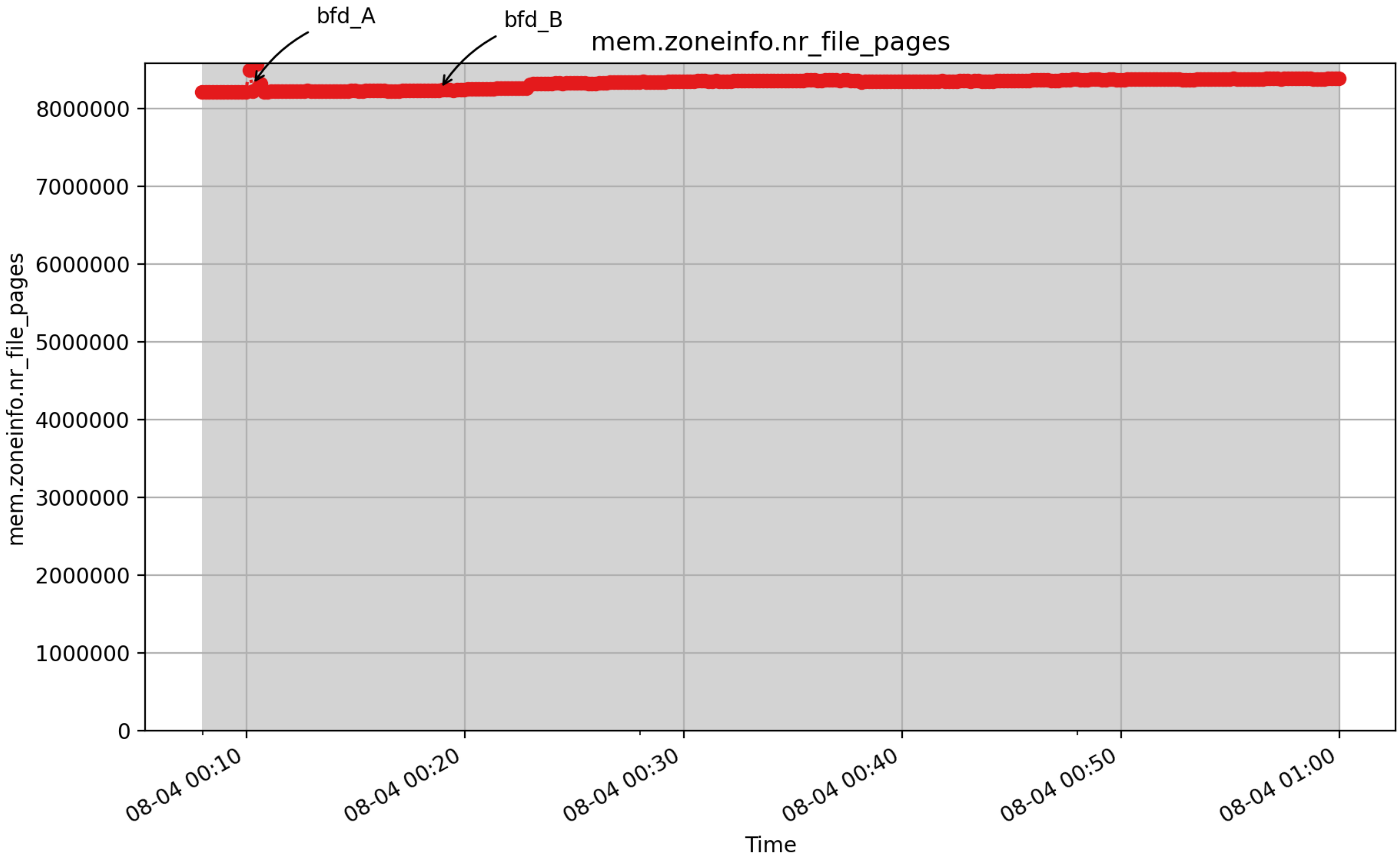
mem.zoneinfo.nr_anon_transparent_hugepages: number of anonymous transparent huge pages in each zone for each NUMA node (Kbyte - U64)



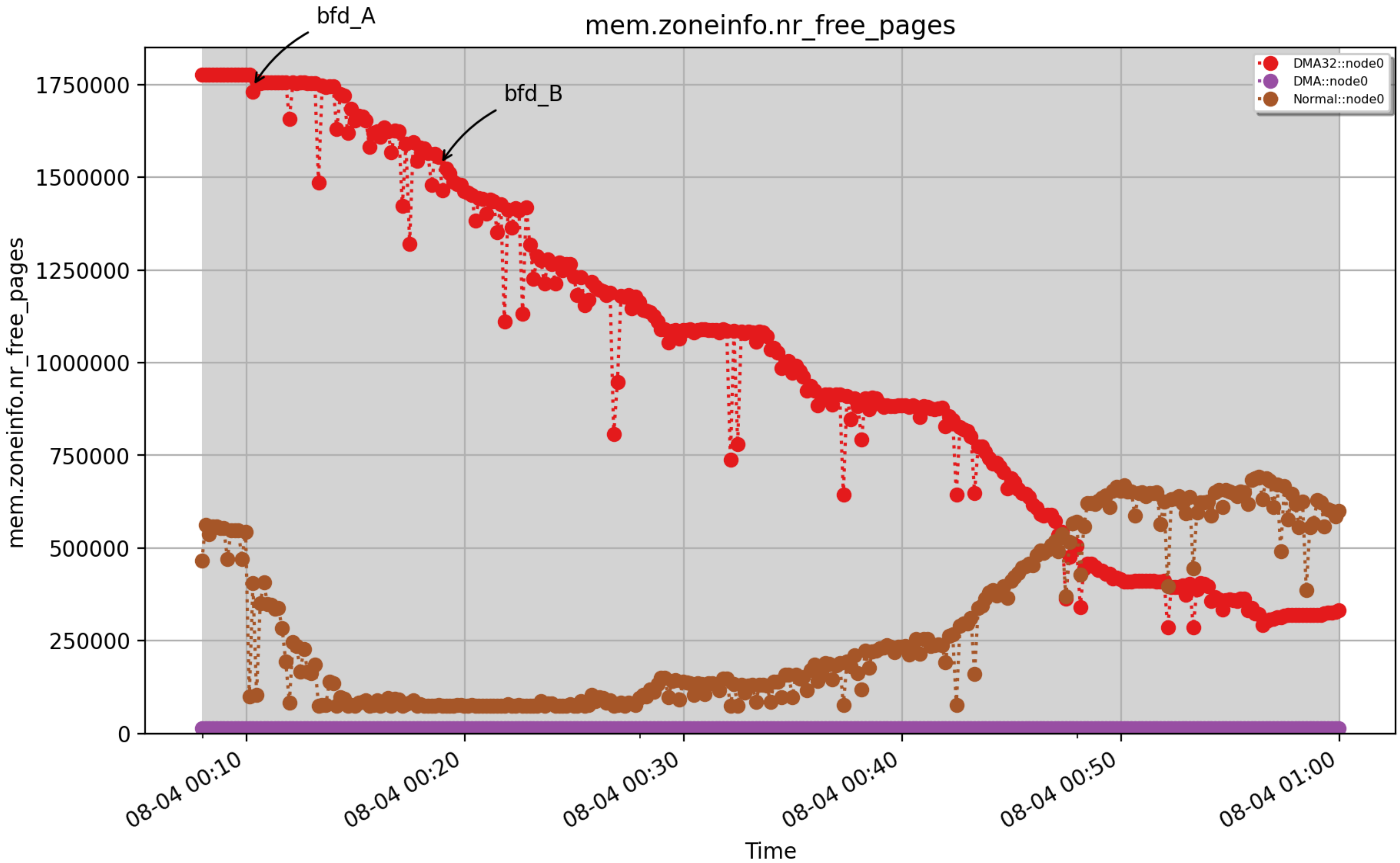
mem.zoneinfo.nr_dirtied: count of pages entering dirty state in each zone for each NUMA node (Kbyte - U64)



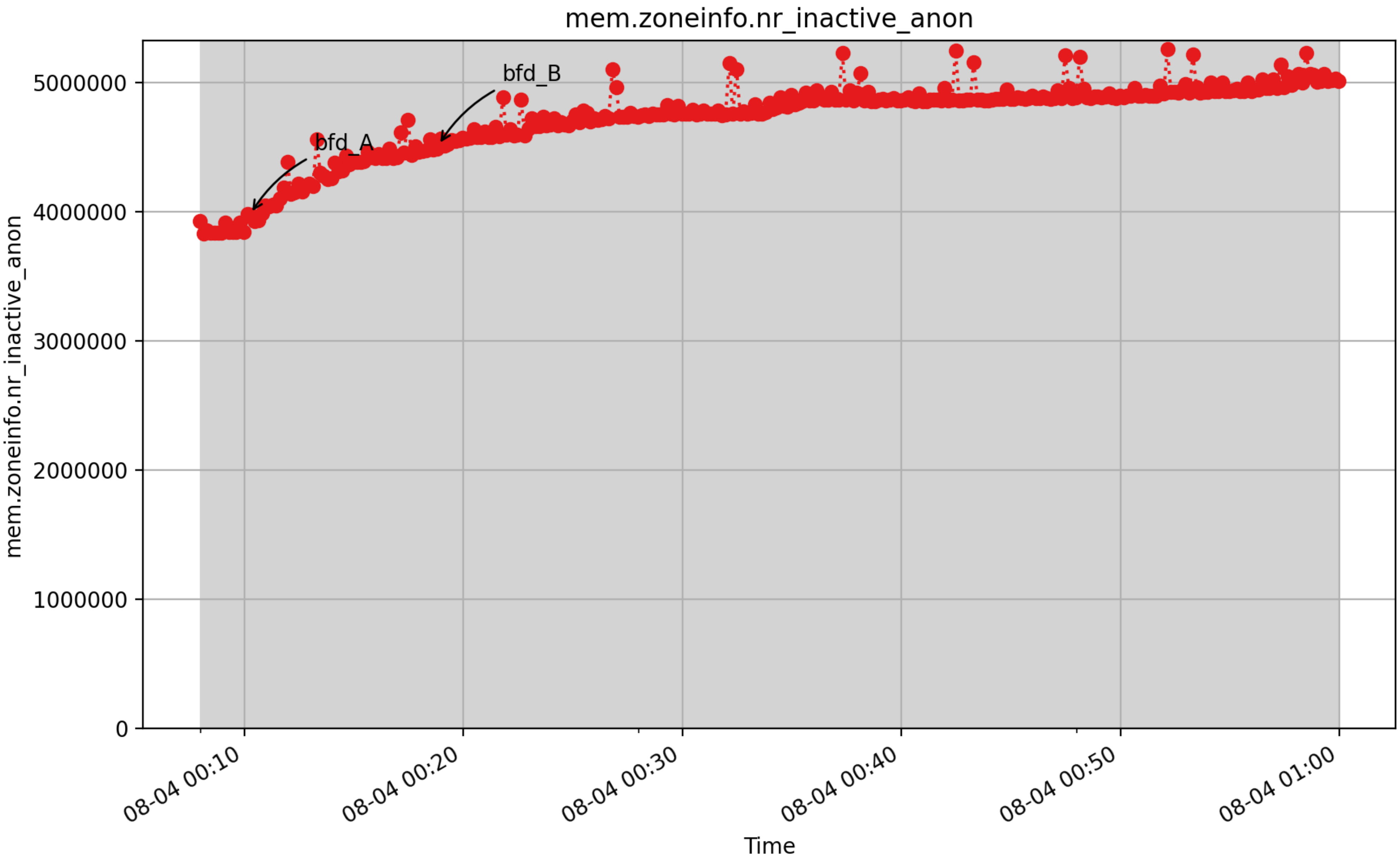
mem.zoneinfo.nr_dirty: number of pages dirty state in each zone for each NUMA node (Kbyte - U64)



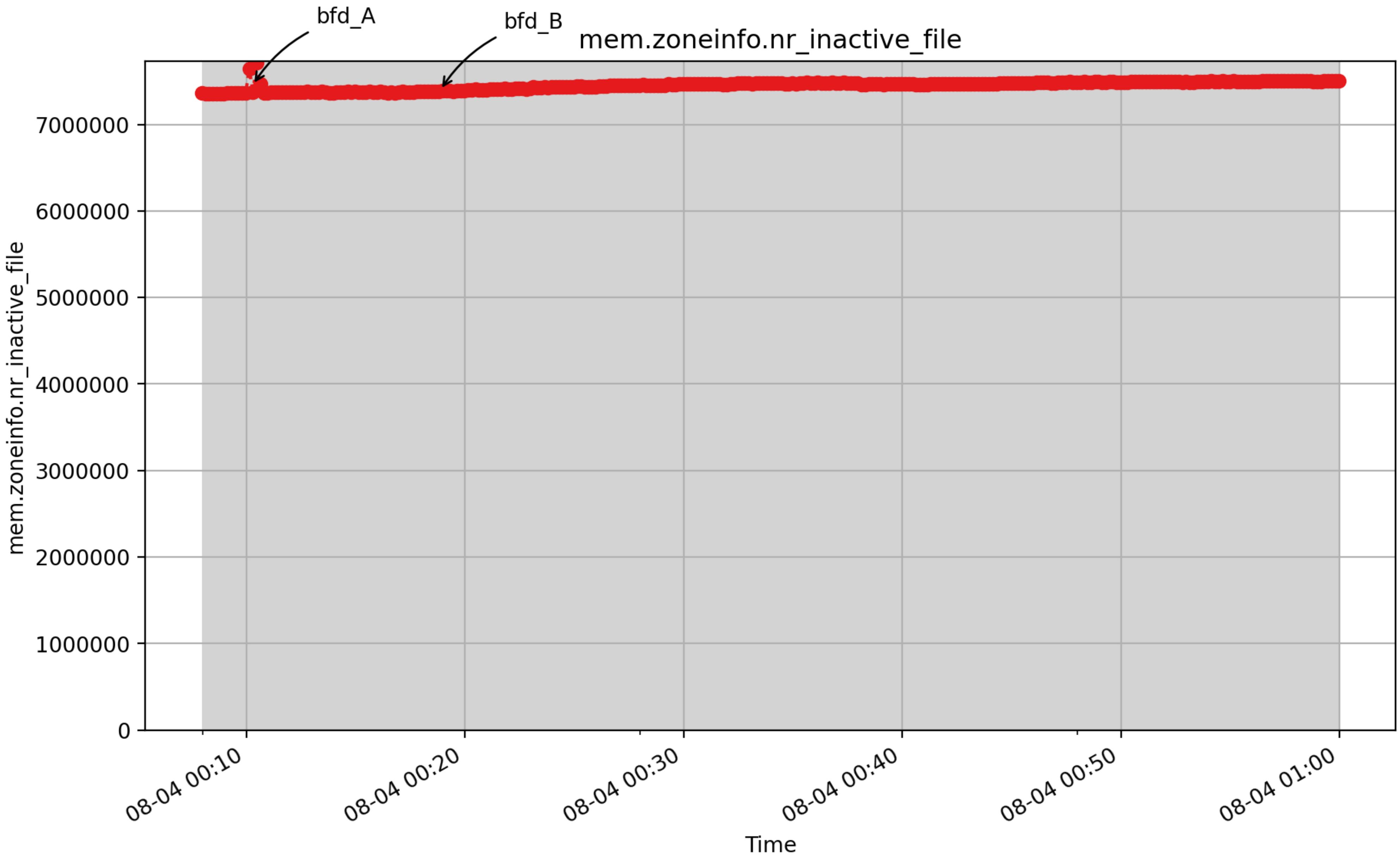
mem.zoneinfo.nr_file_pages: number of file pagecache pages in each zone for each NUMA node (Kbyte - U64)



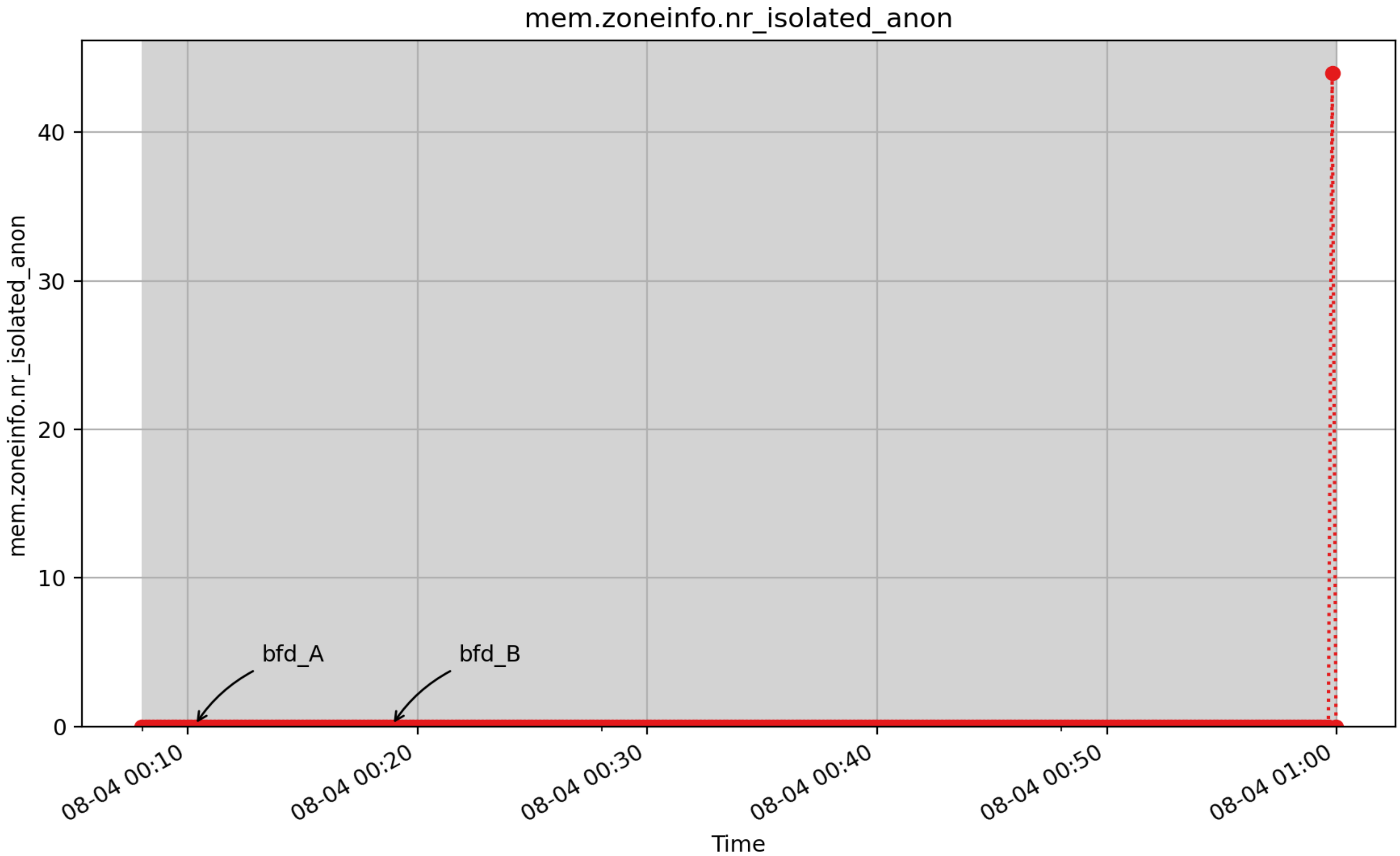
mem.zoneinfo.nr_free_pages: number of free pages in each zone for each NUMA node. (Kbyte - U64)



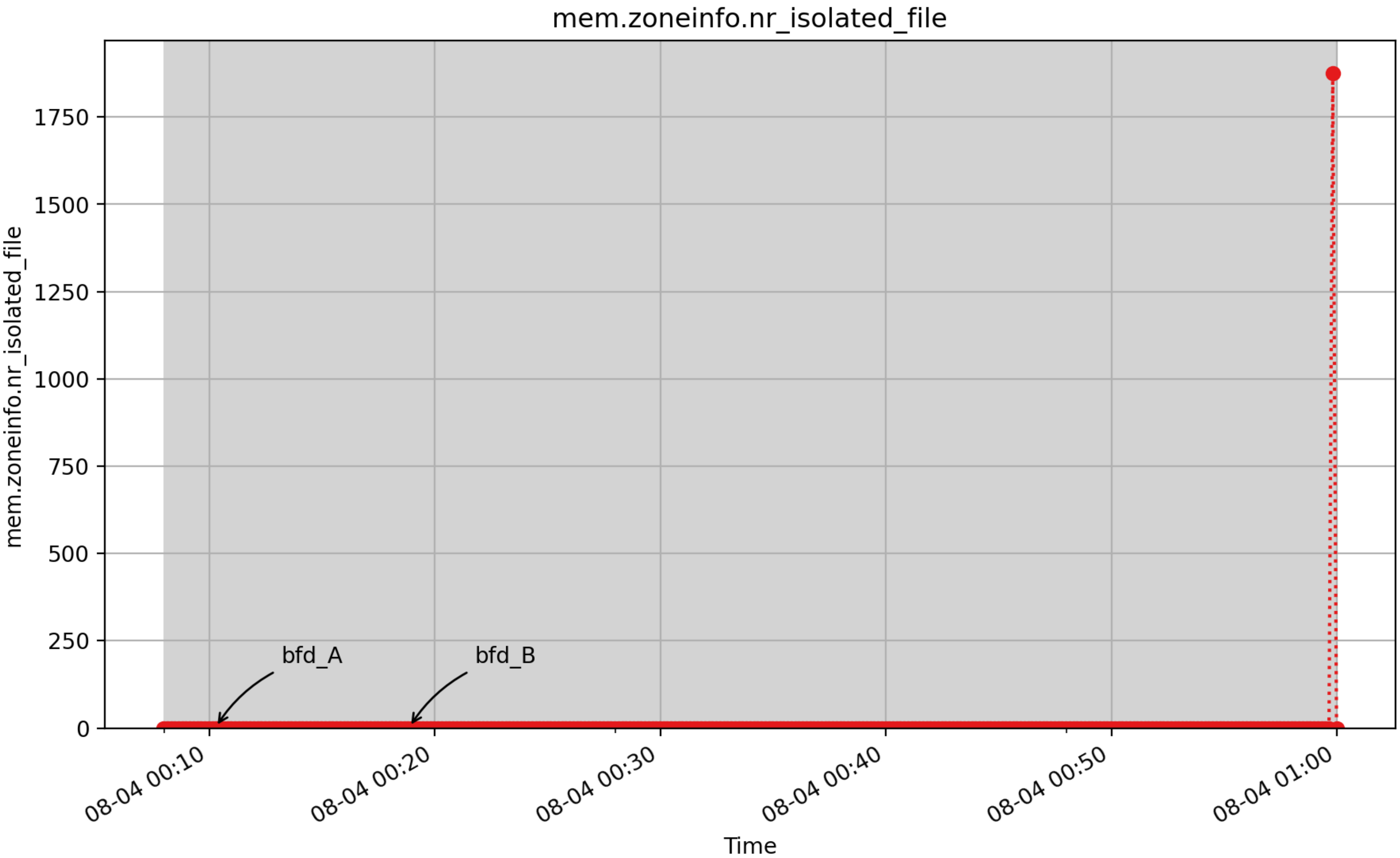
mem.zoneinfo.nr_inactive_anon: number of inactive anonymous memory pages in each zone for each NUMA node (Kbyte - U64)



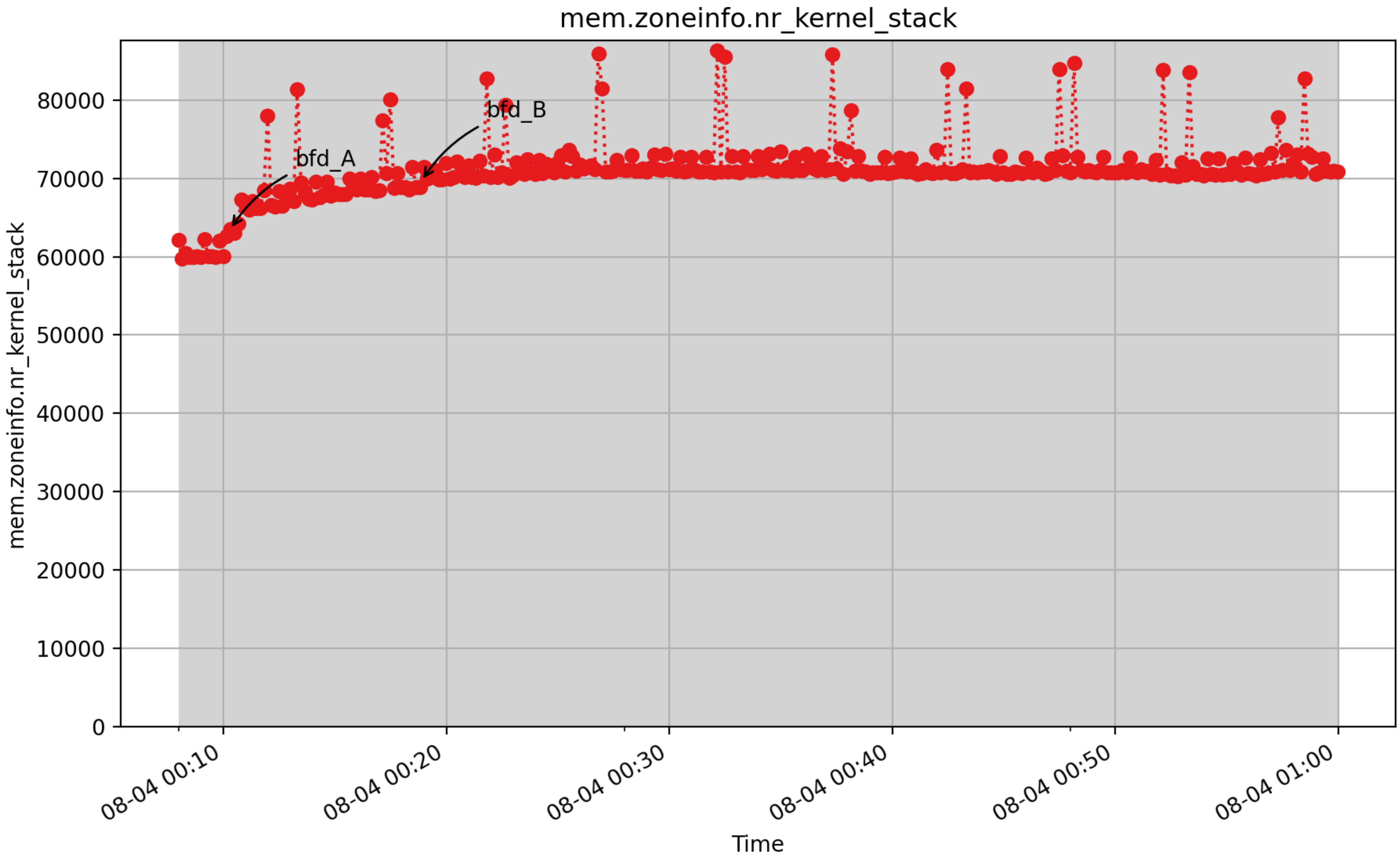
mem.zoneinfo.nr_inactive_file: number of inactive file memory pages in each zone for each NUMA node (Kbyte - U64)



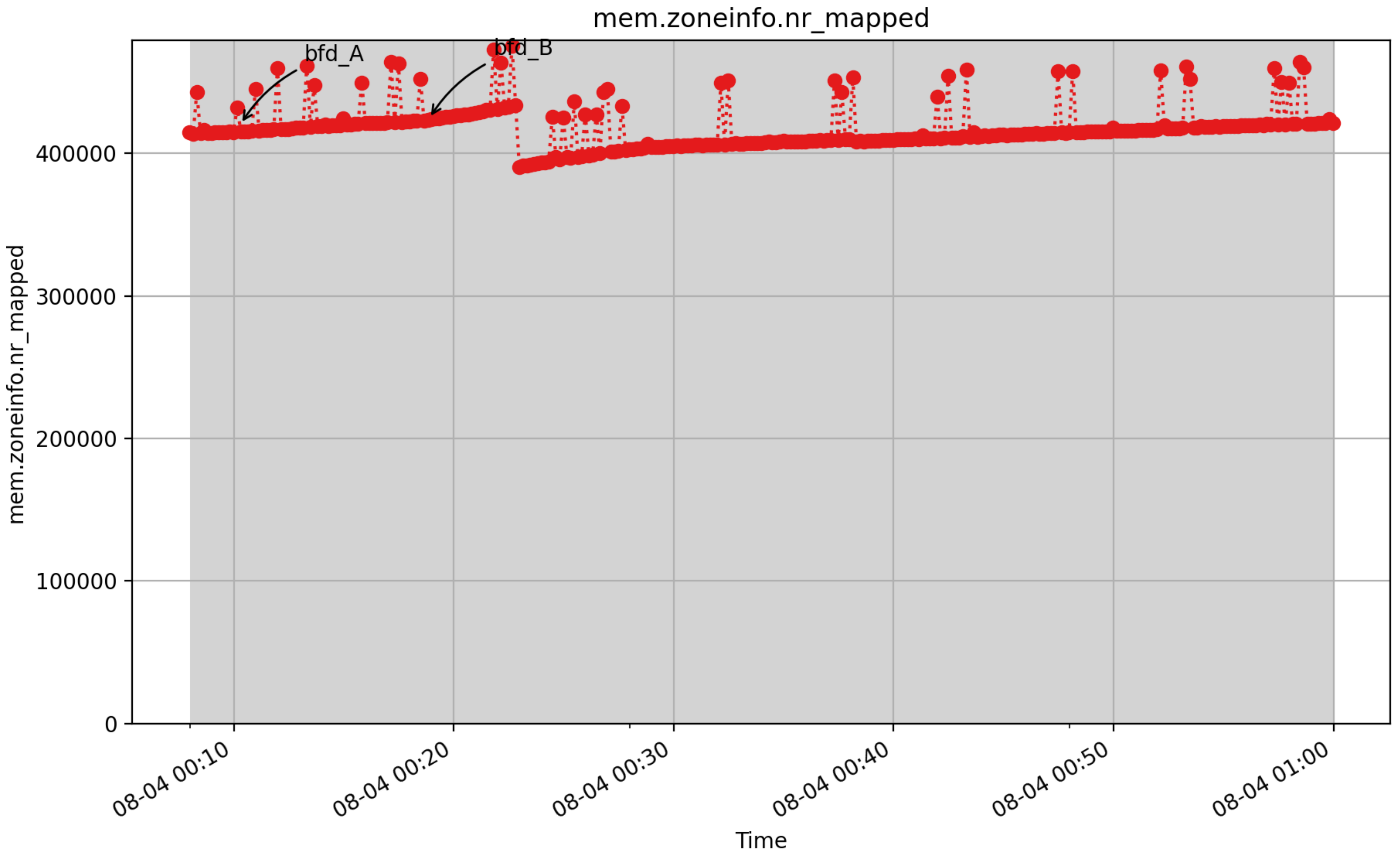
mem.zoneinfo.nr_isolated_anon: number of isolated anonymous memory pages in each zone for each NUMA node (Kbyte - U64)



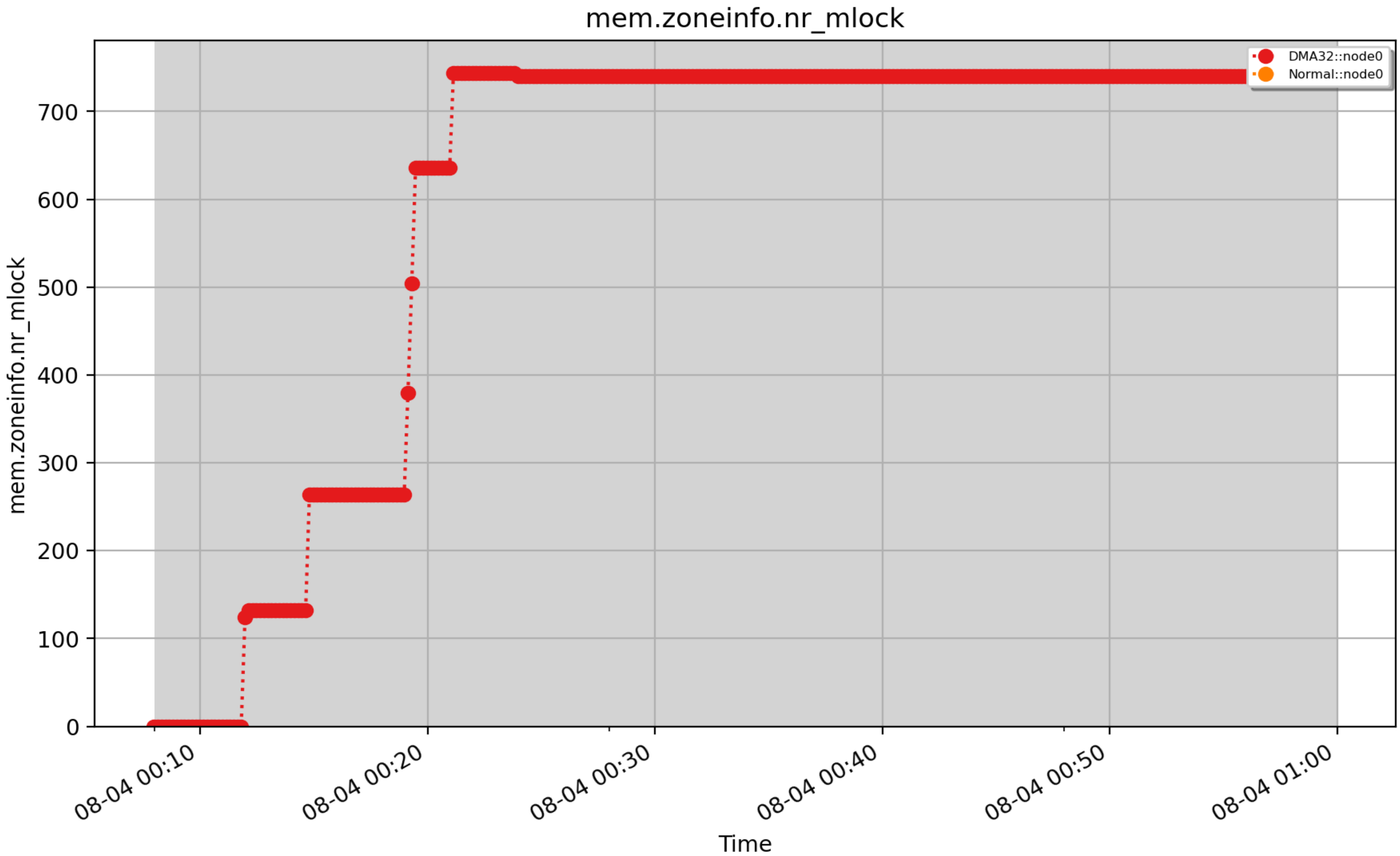
mem.zoneinfo.nr_isolated_file: number of isolated file memory pages in each zone for each NUMA node (Kbyte - U64)



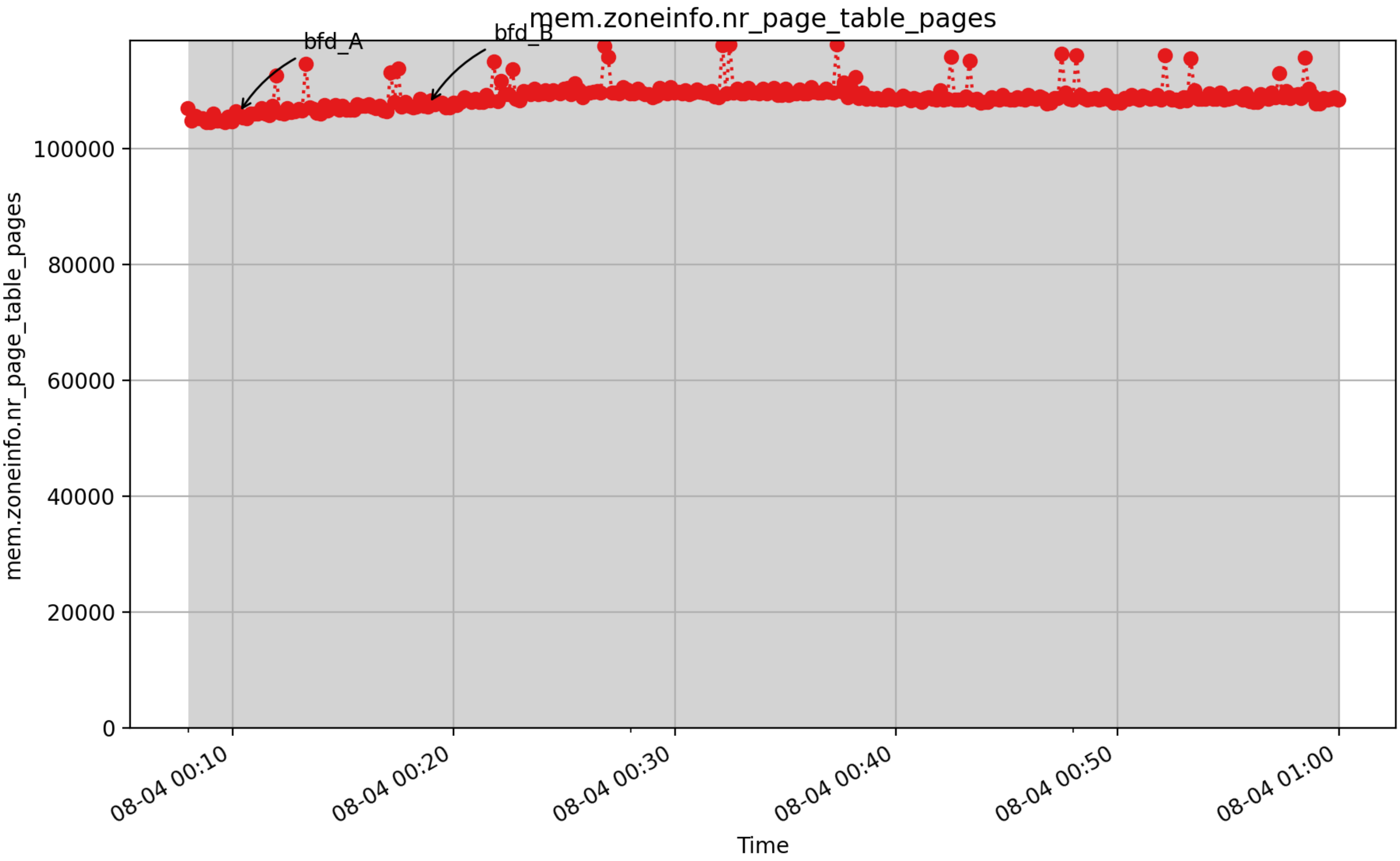
mem.zoneinfo.nr_kernel_stack: number of pages of kernel stack in each zone for each NUMA node (Kbyte - U64)



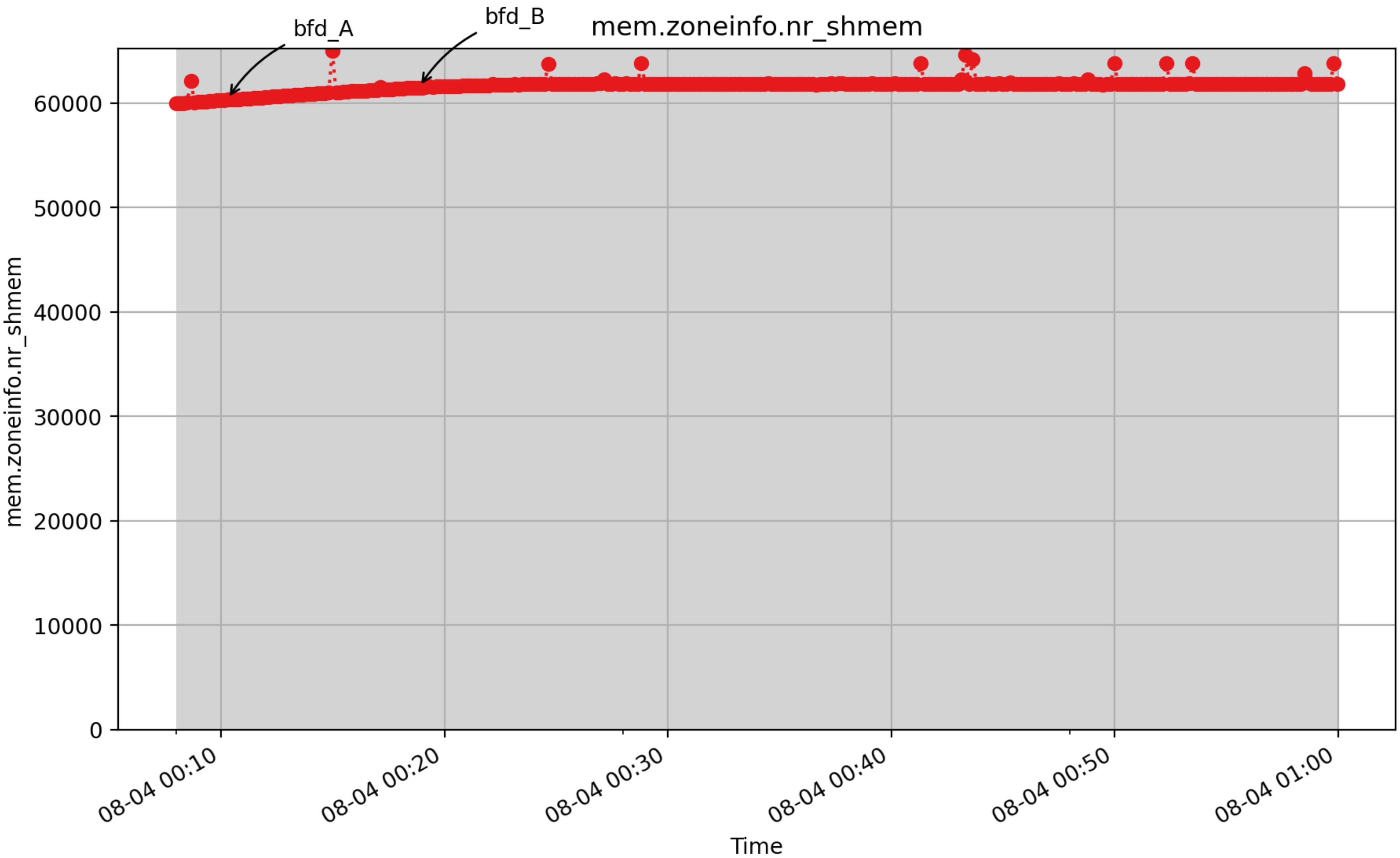
mem.zoneinfo.nr_mapped: number of mapped pagecache pages in each zone for each NUMA node (Kbyte - U64)



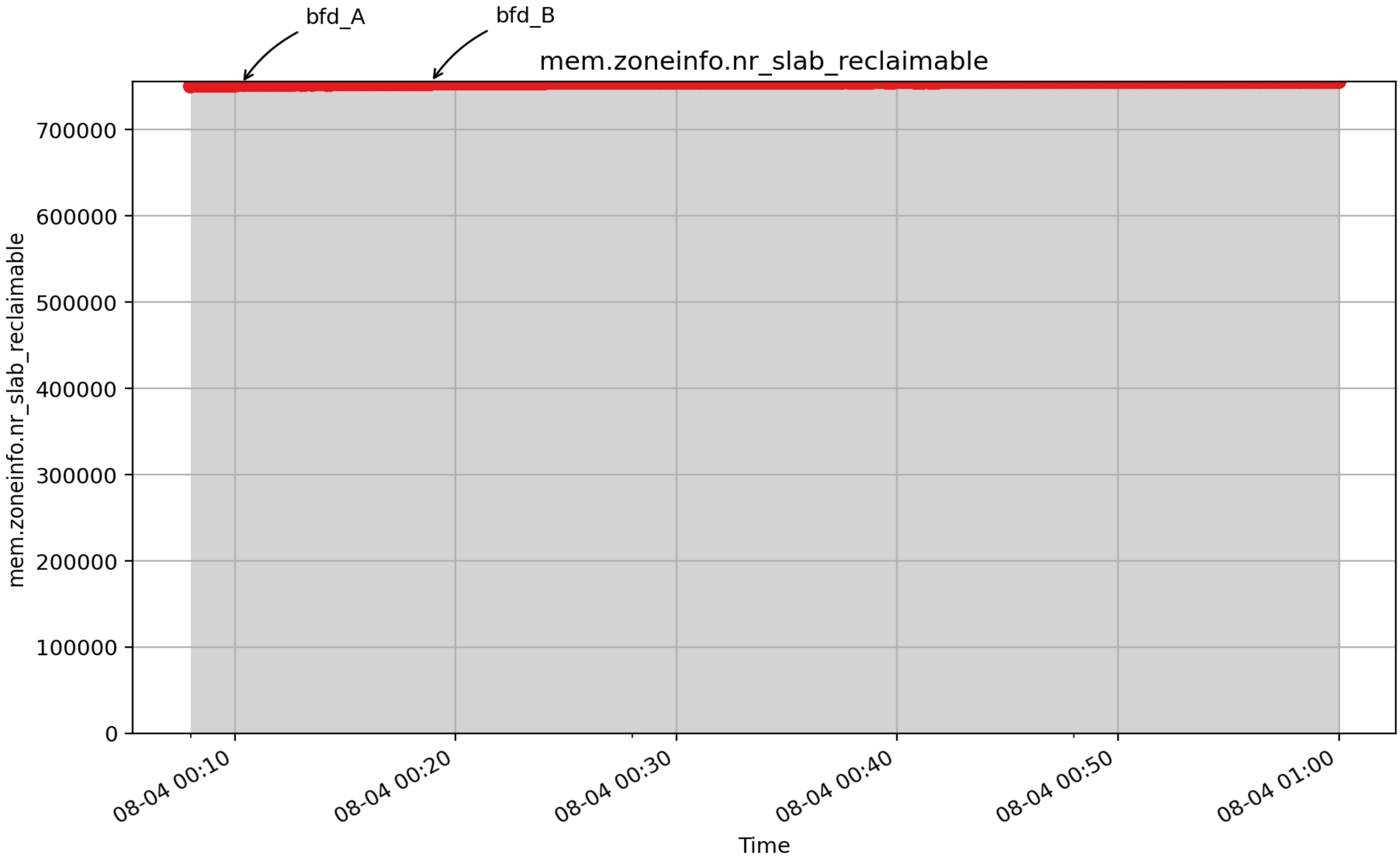
mem.zoneinfo.nr_mlock: number of pages under mlock in each zone for each NUMA node (Kbyte - U64)



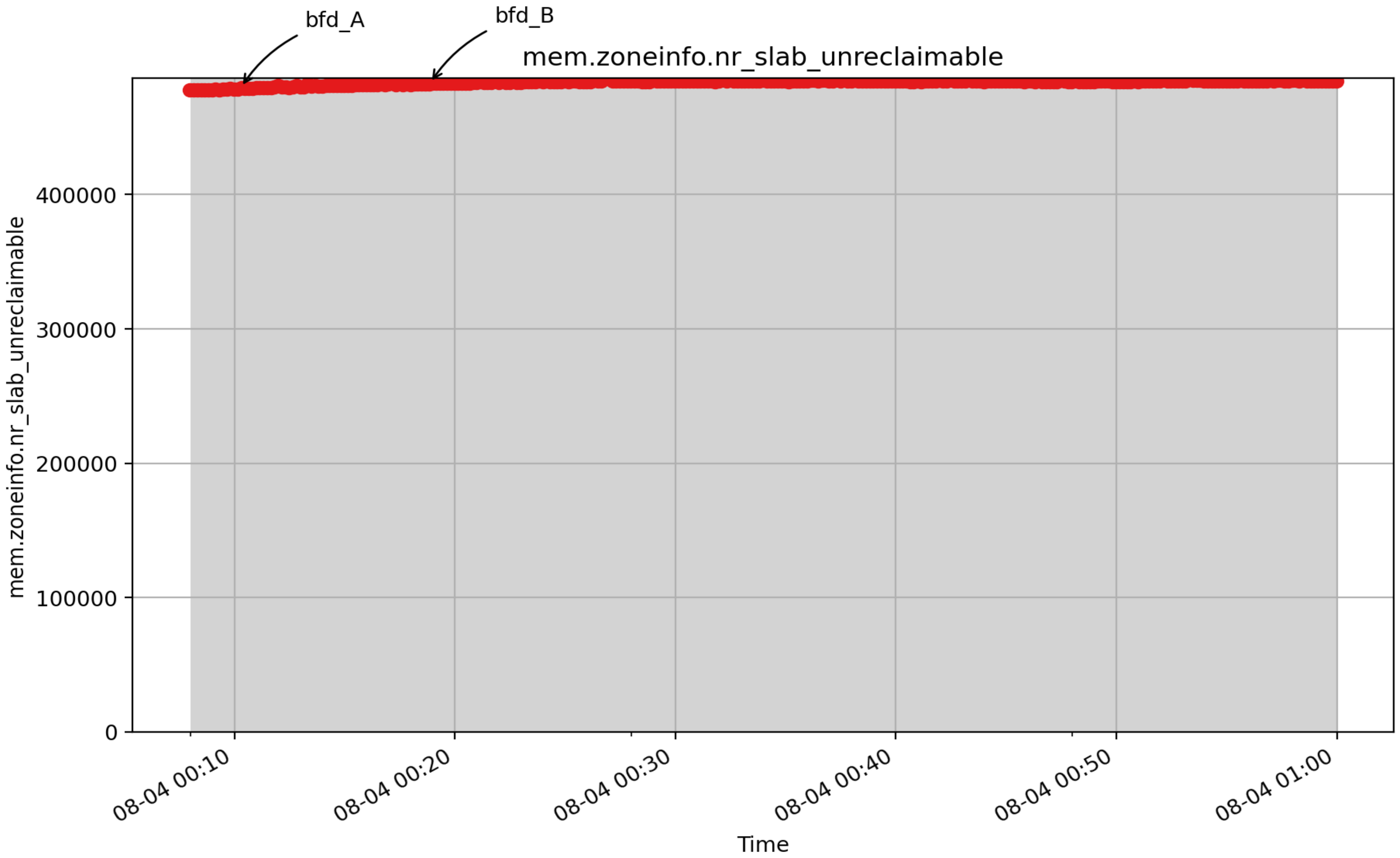
mem.zoneinfo.nr_page_table_pages: number of page table pages in each zone for each NUMA node (Kbyte - U64)



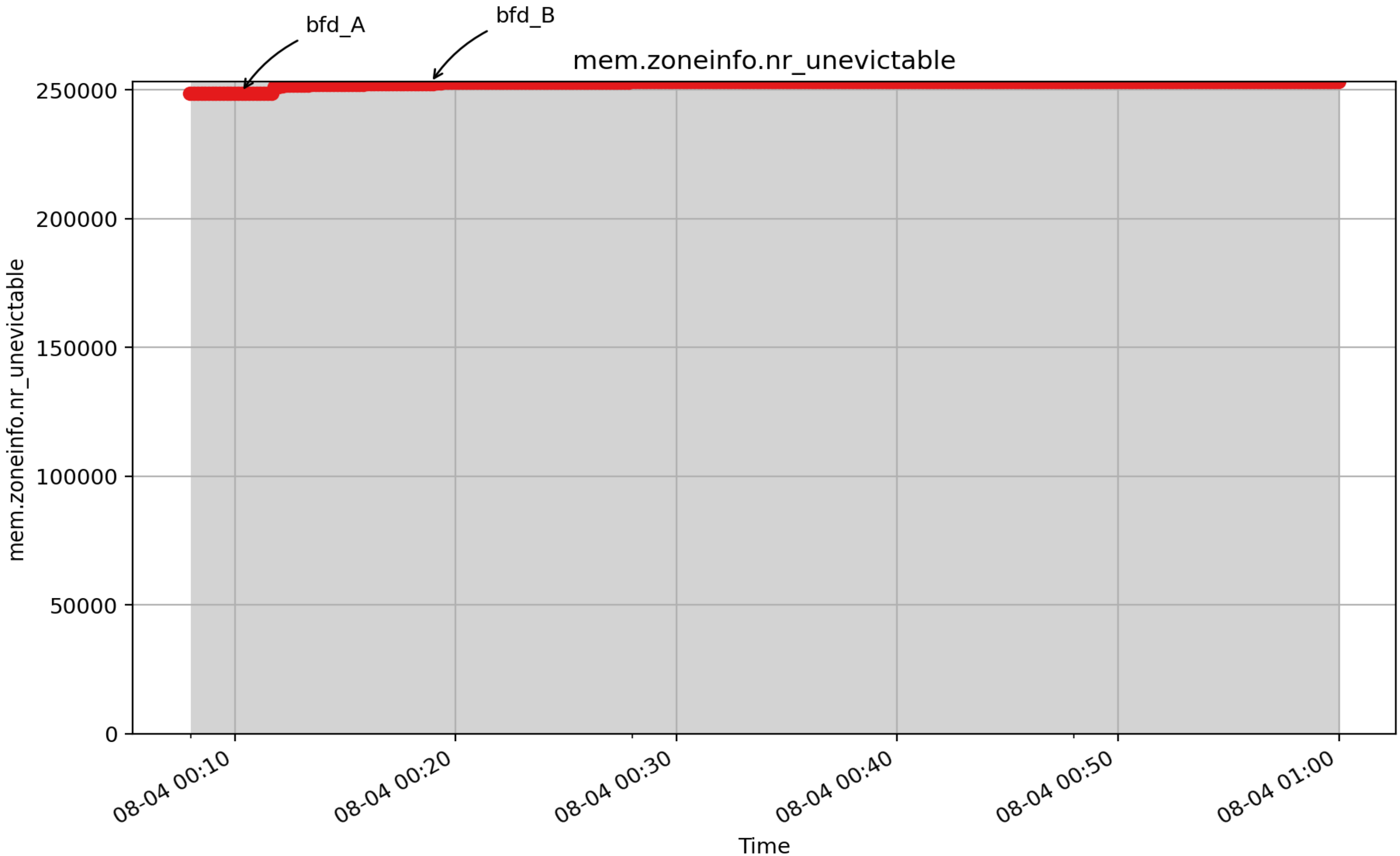
mem.zoneinfo.nr_shmem: number of shared memory pages in each zone for each NUMA node (Kbyte - U64)



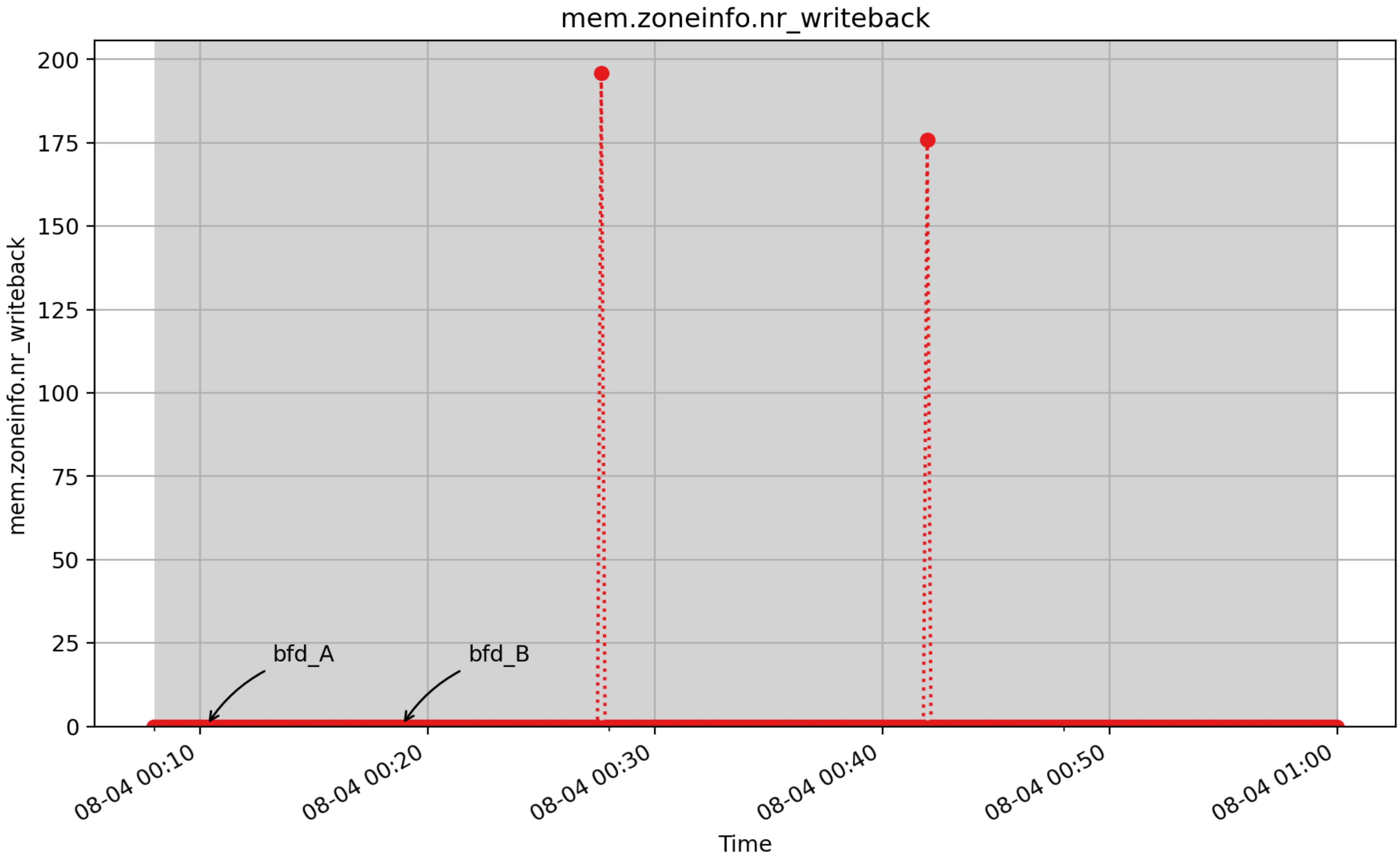
mem.zoneinfo.nr_slab_reclaimable: number of reclaimable slab pages in each zone for each NUMA node (Kbyte - U64)



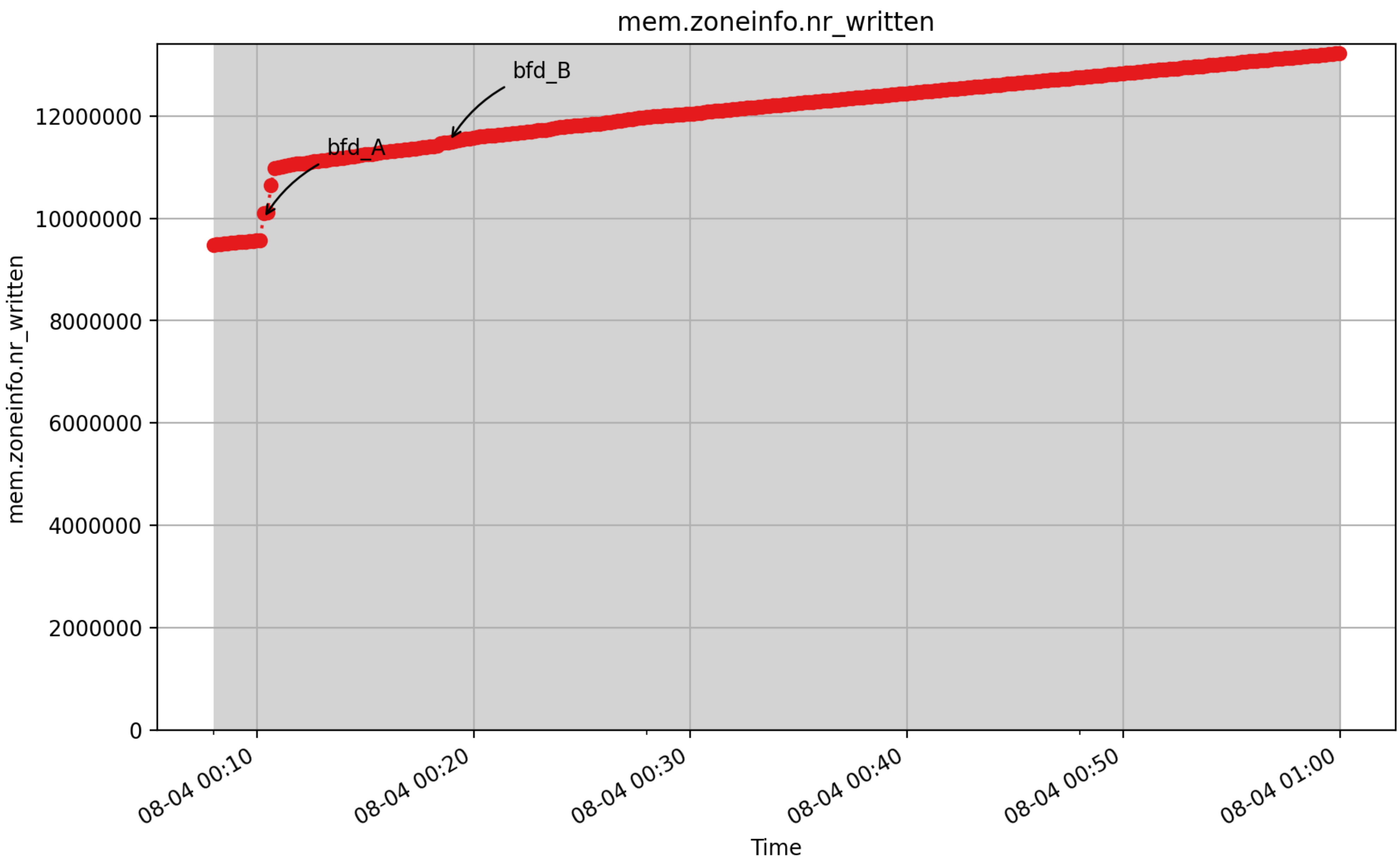
mem.zoneinfo.nr_slab_unreclaimable: number of unreclaimable slab pages in each zone for each NUMA node (Kbyte - U64)



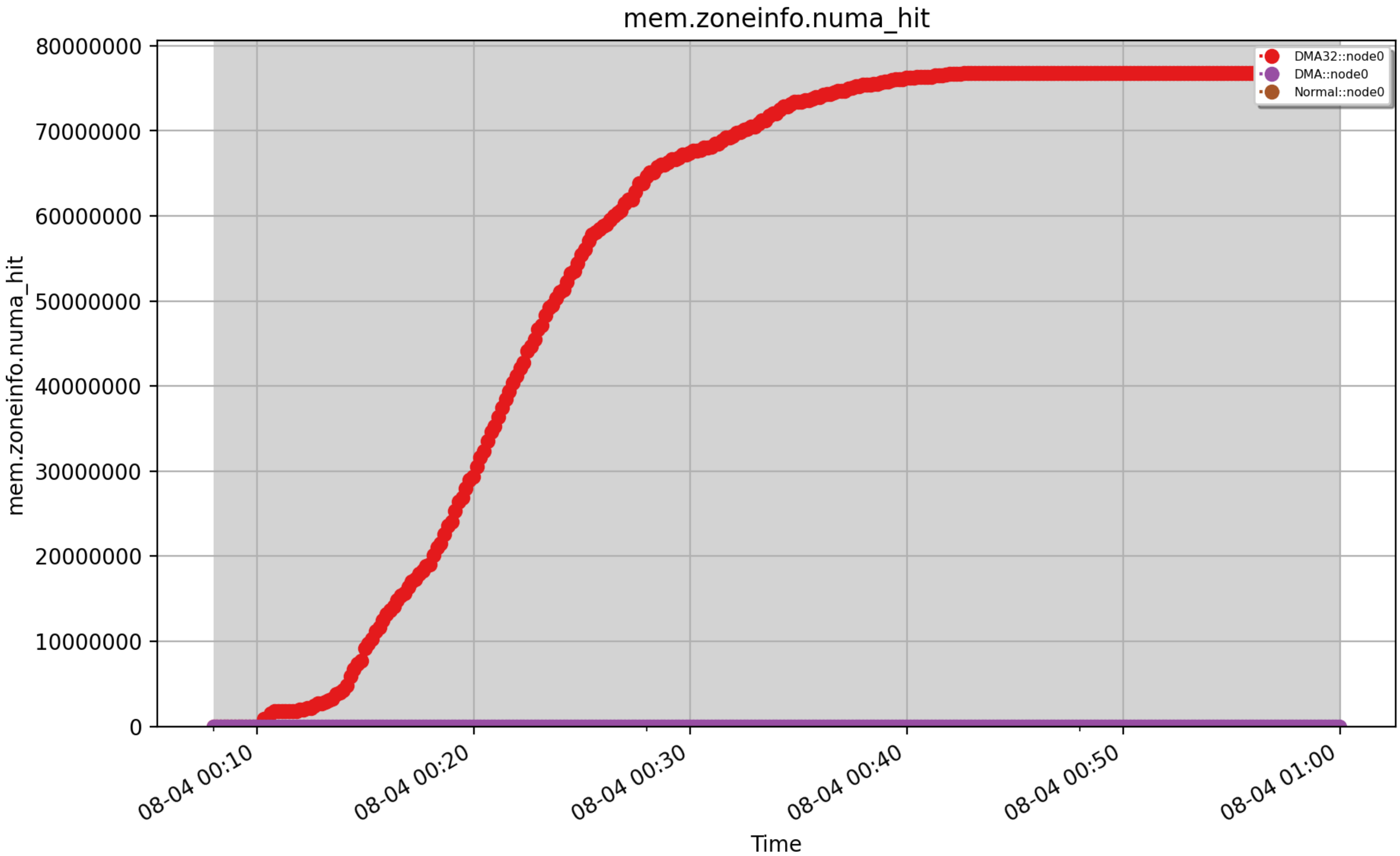
mem.zoneinfo.nr_unevictable: number of unevictable pages in each zone for each NUMA node (Kbyte - U64)



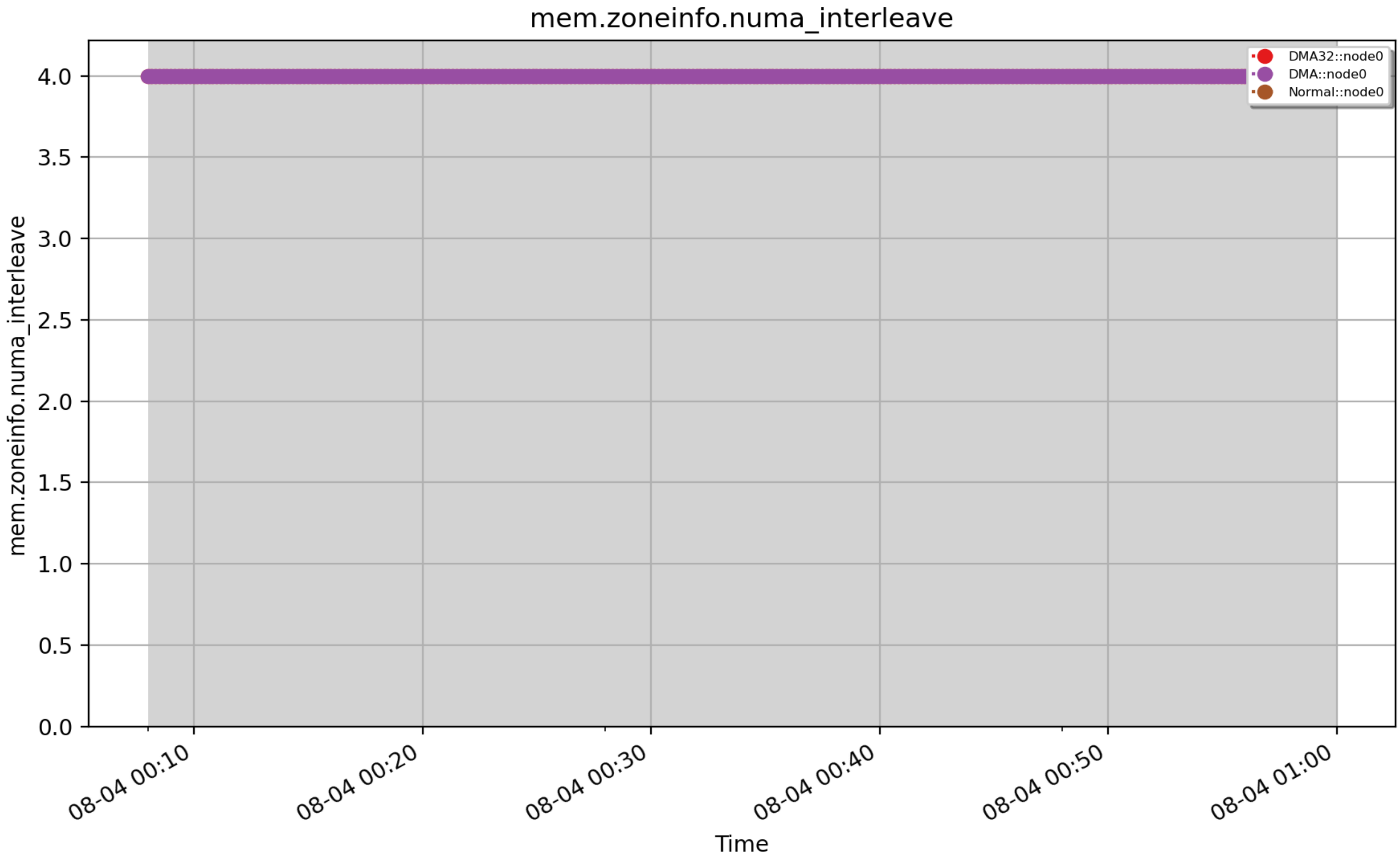
mem.zoneinfo.nr_writeback: number of pages writeback state in each zone for each NUMA node (Kbyte - U64)



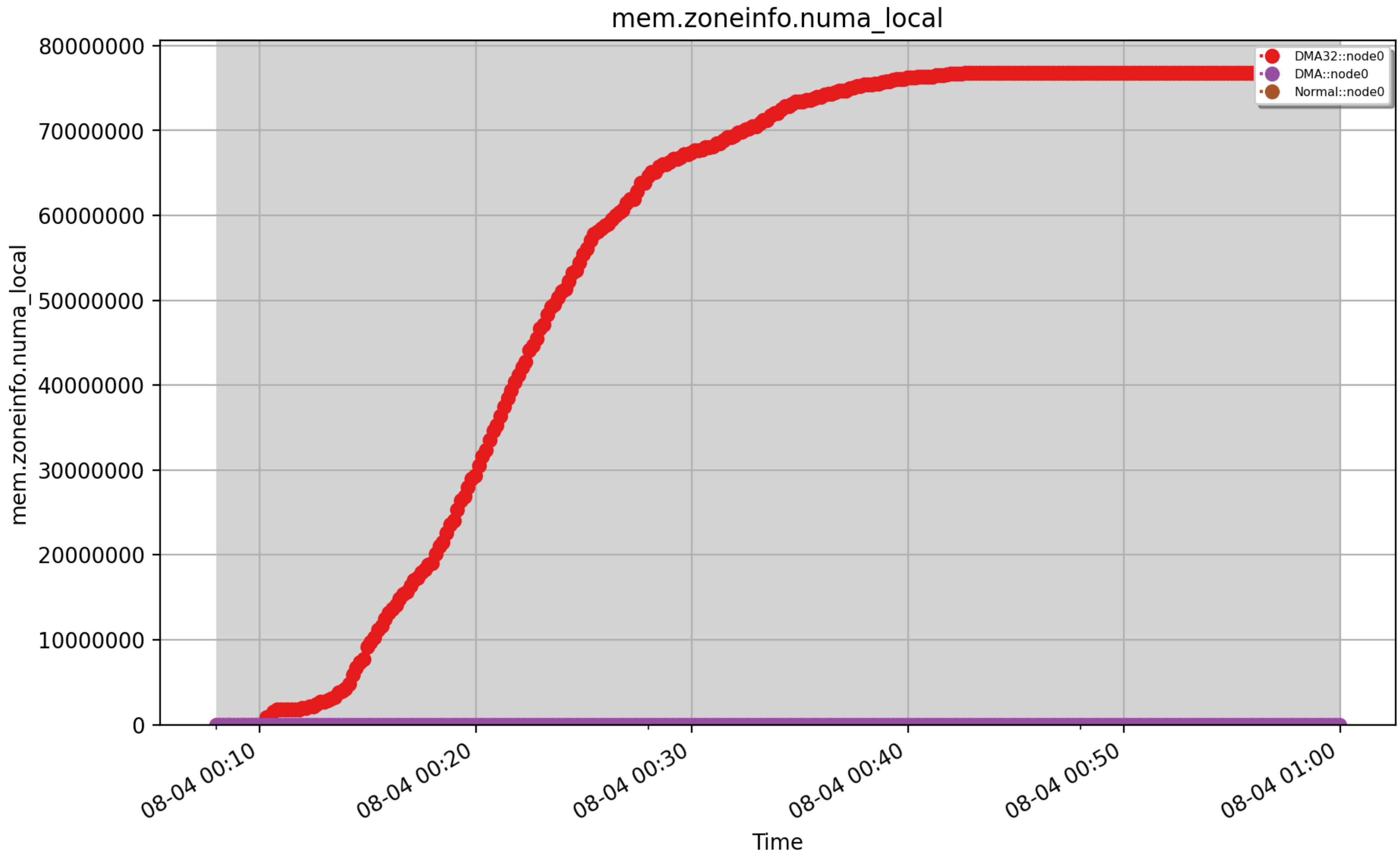
mem.zoneinfo.nr_written: count of pages written out in each zone for each NUMA node (Kbyte - U64)



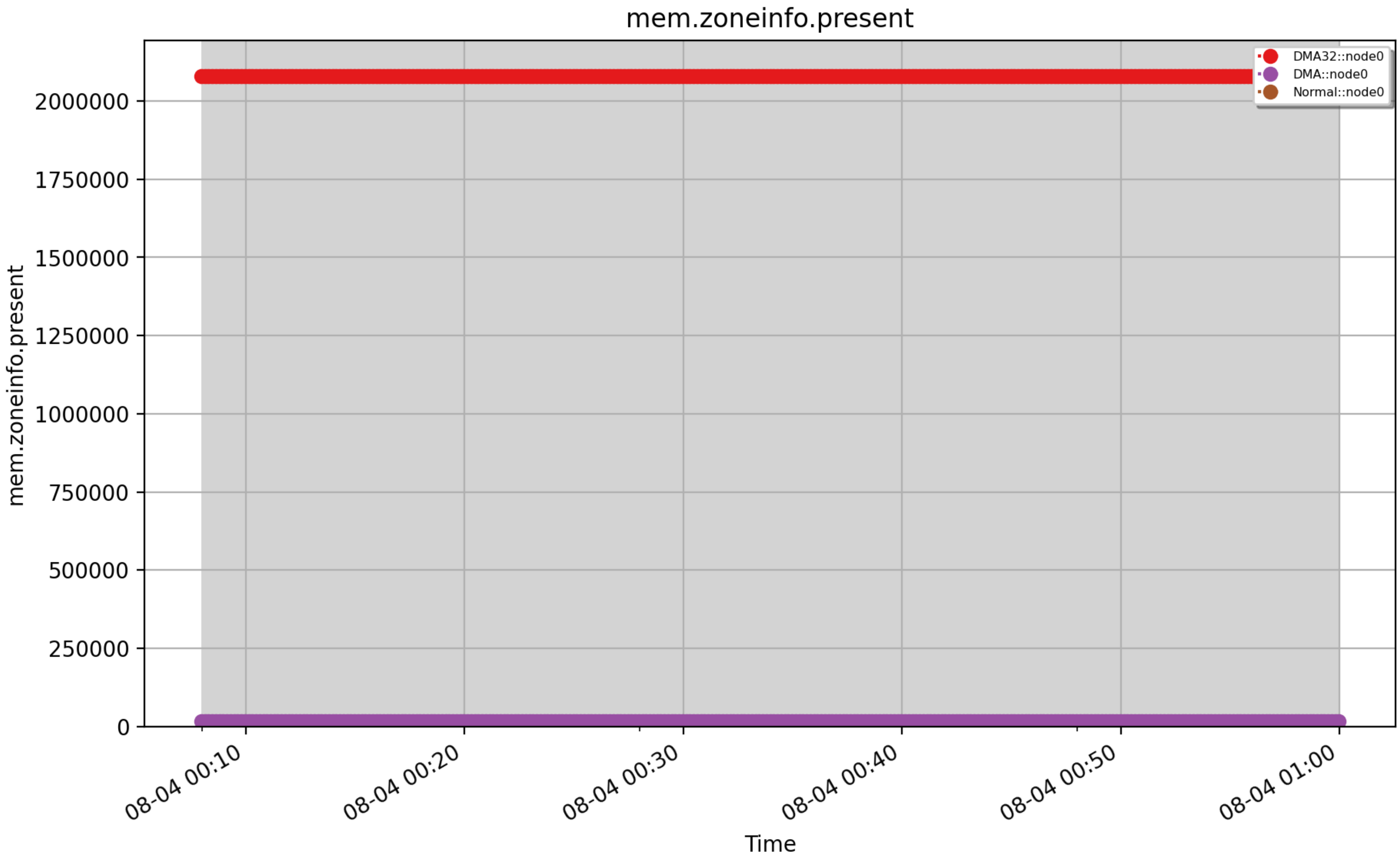
mem.zoneinfo.numa_hit: Count of successful allocations from preferred NUMA zone in each zone for each NUMA node. (Kbyte - U64)



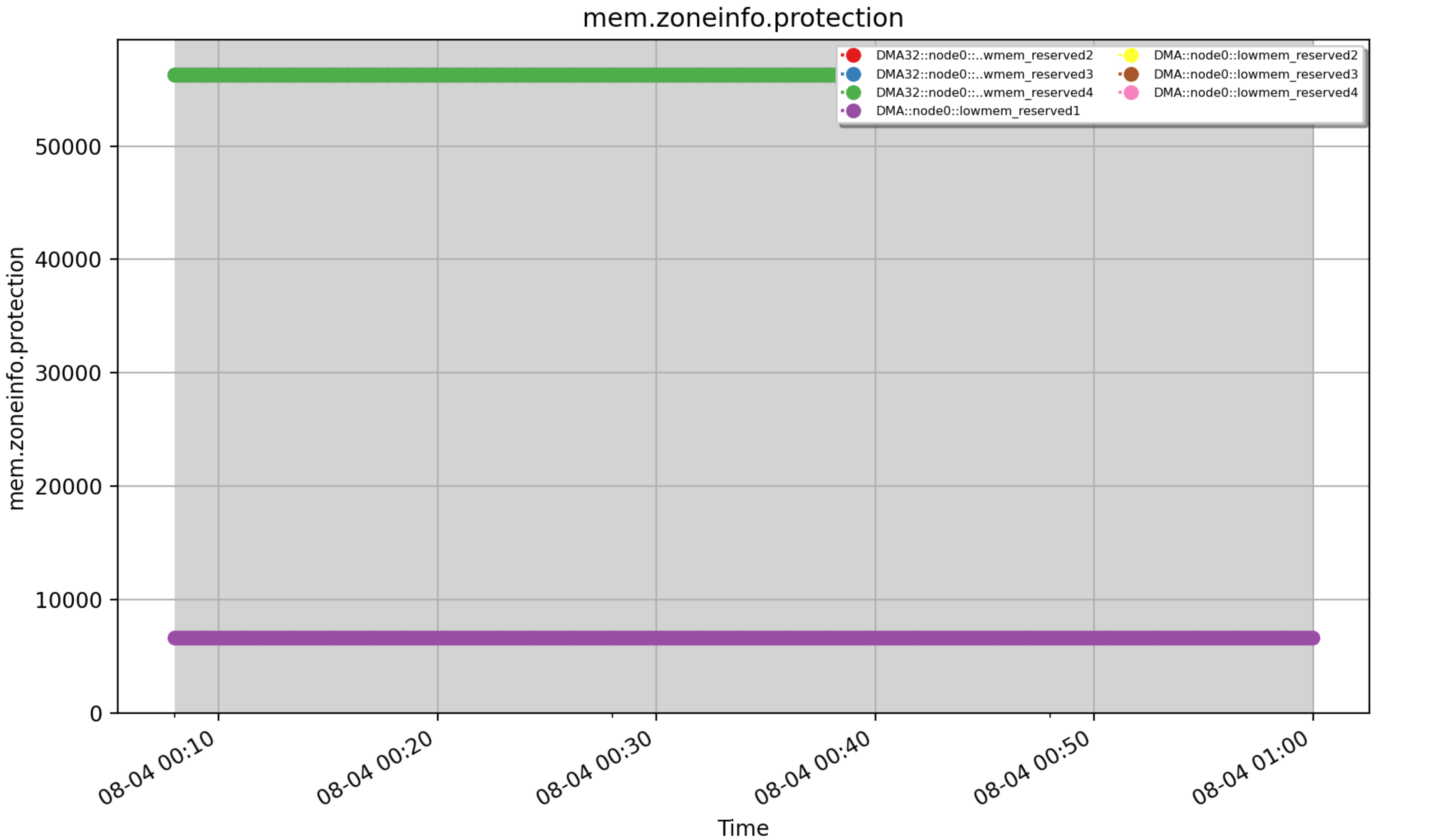
mem.zoneinfo.numa_interleave: count of interleaved NUMA allocations in each zone for each NUMA node (Kbyte - U64)



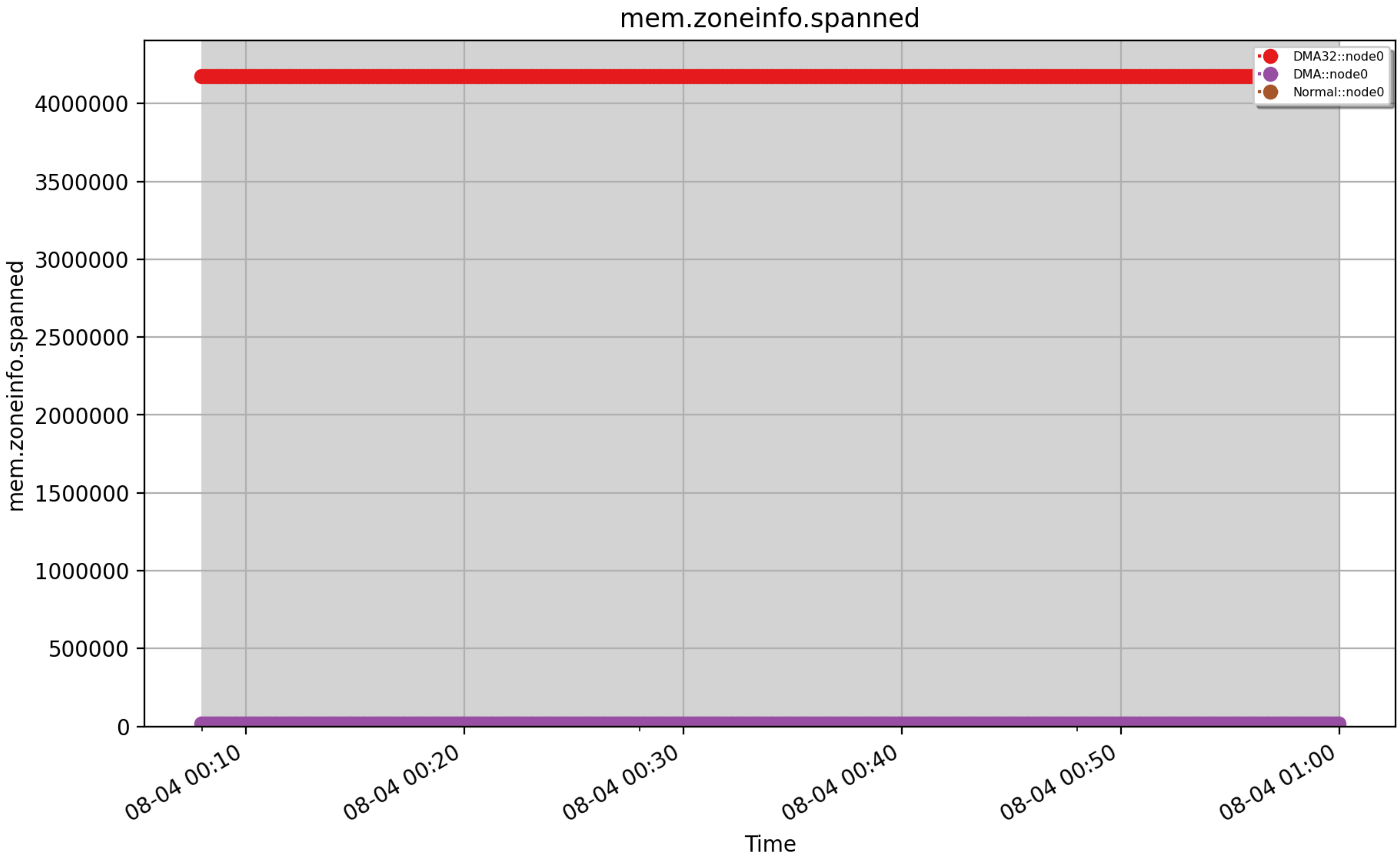
mem.zoneinfo.numa_local: Count of successful allocations from local NUMA zone in each zone for each NUMA node. (Kbyte - U64)



mem.zoneinfo.present: present space in each zone for each NUMA node (Kbyte - U64)

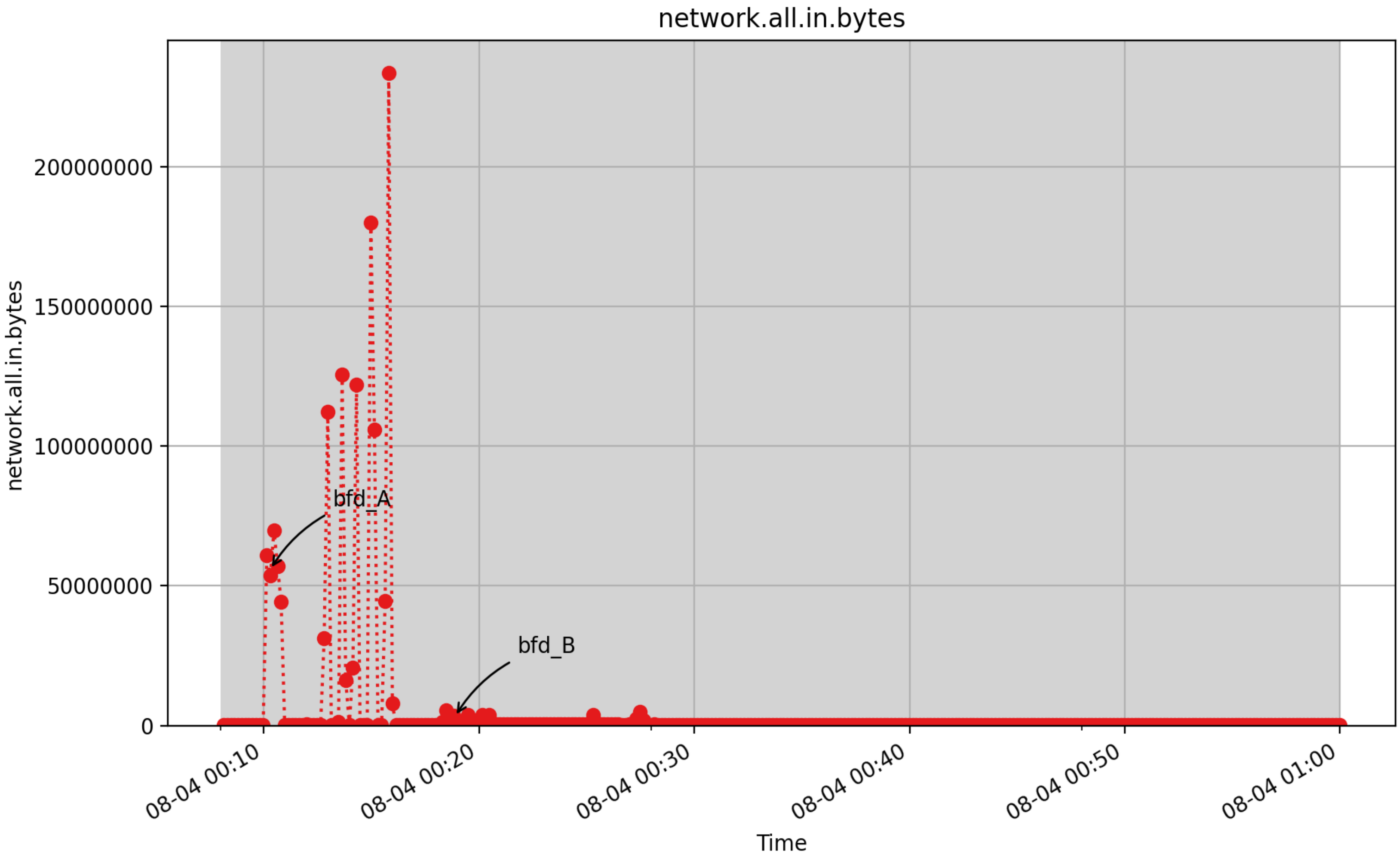


mem.zoneinfo.protection: protection space in each zone for each NUMA node (Kbyte - U64)

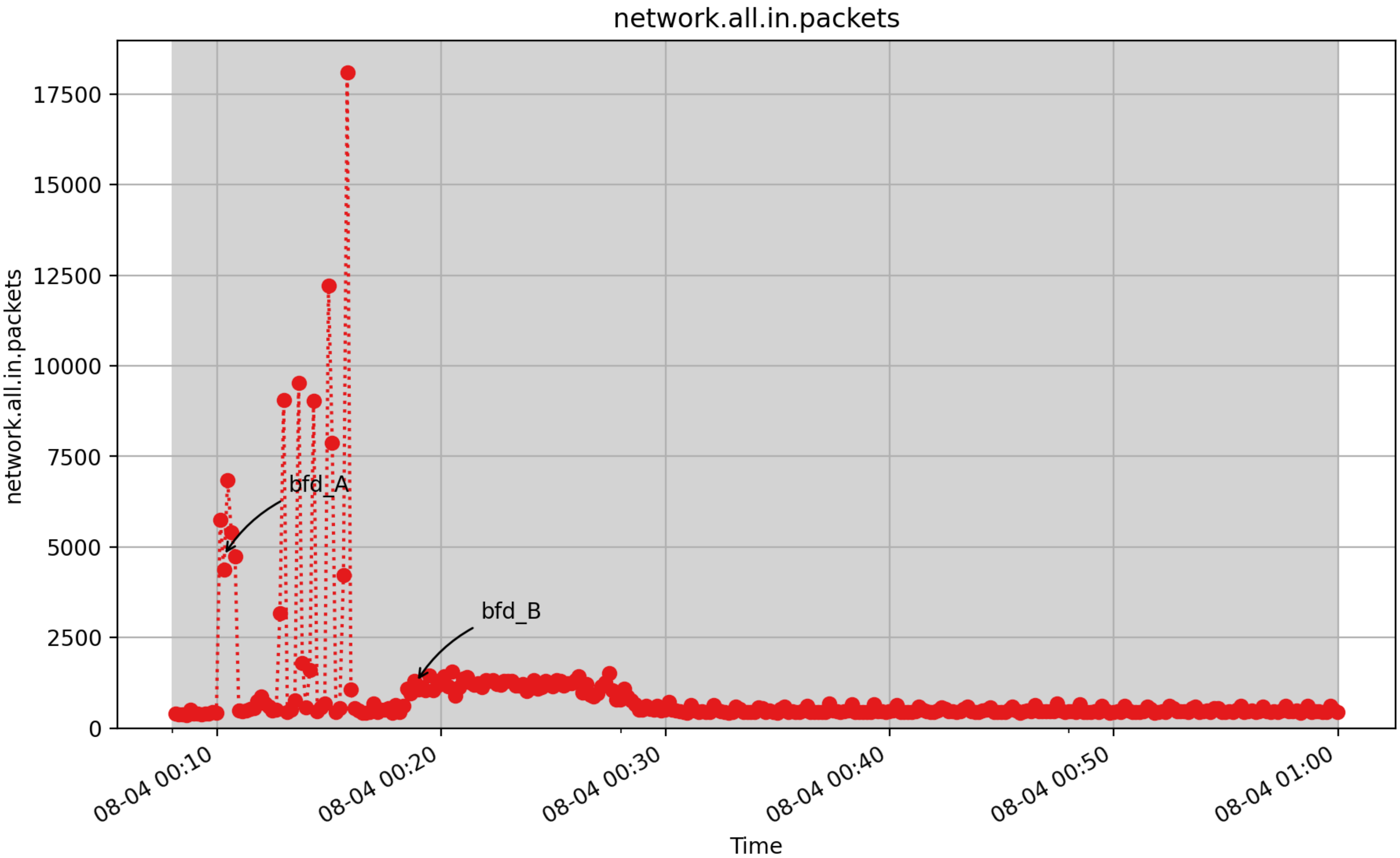


mem.zoneinfo.spanned: spanned space in each zone for each NUMA node (Kbyte - U64)

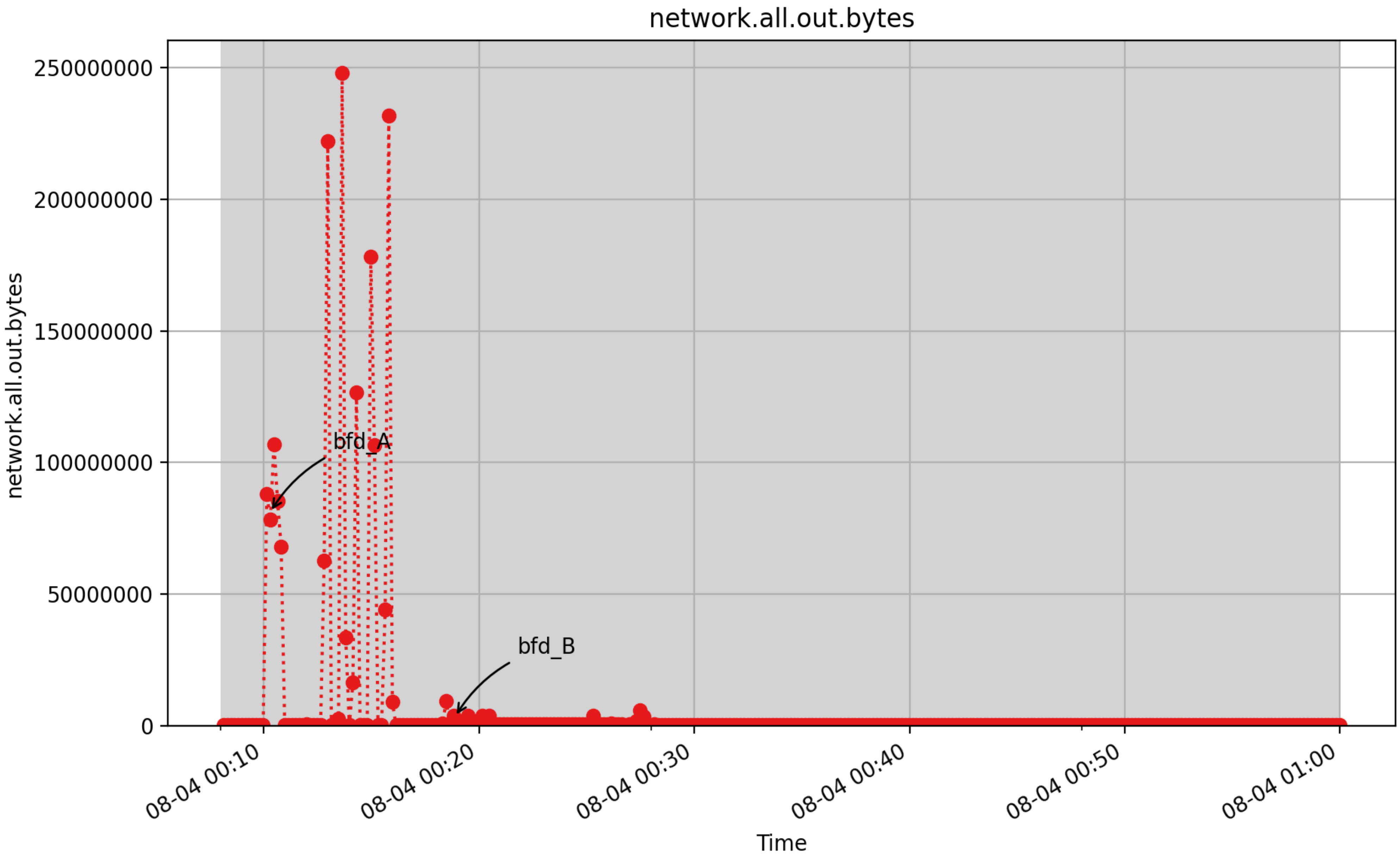
Network



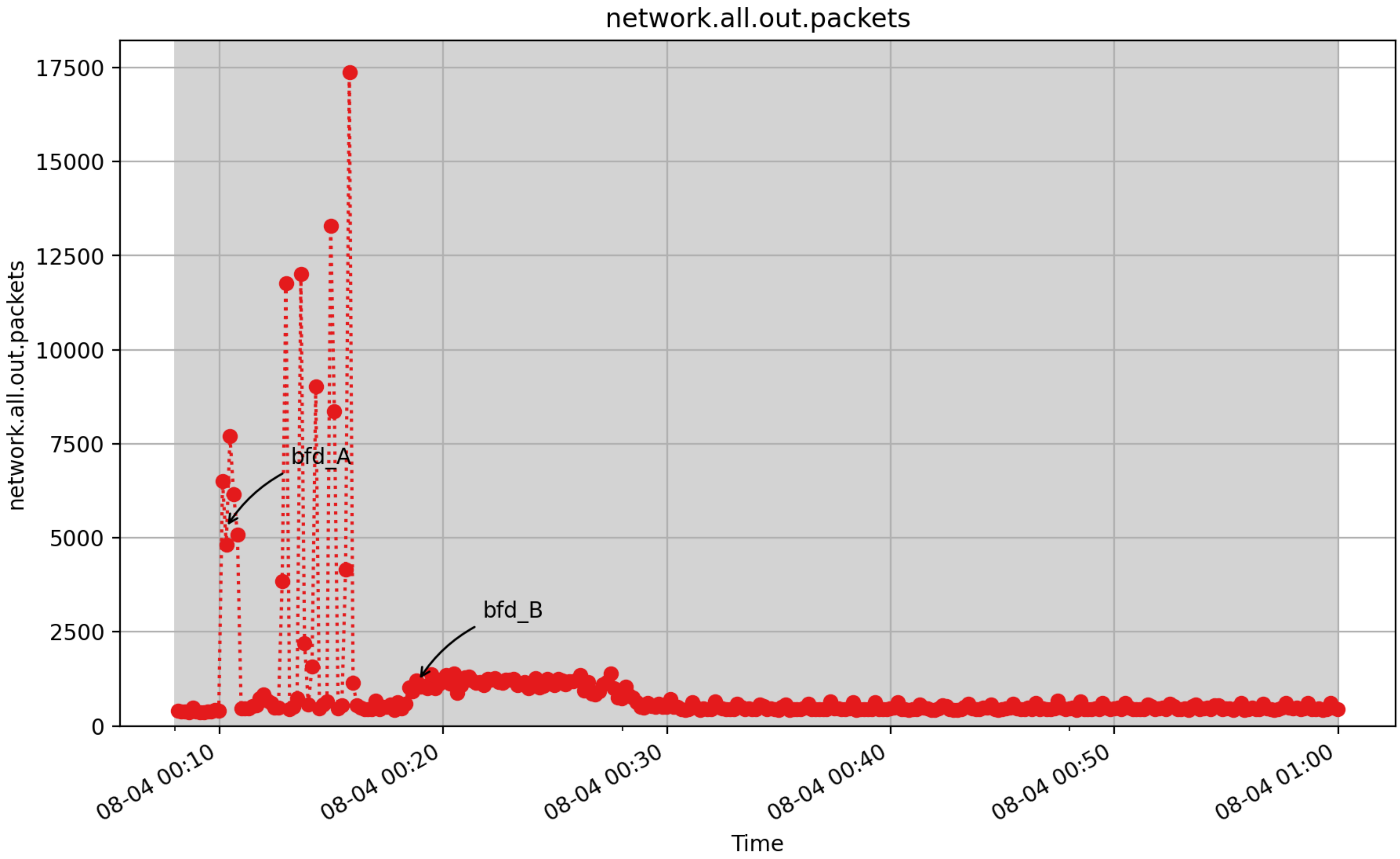
network.all.in.bytes: Sum of bytes column on the "Receive" side of /proc/net/dev for network interfaces deemed to be 'physical' interfaces, using regular expression pattern described in the $PCP_SYSCONF_DIR/linux/interfaces.conf file. (byte - U64) - *rate converted*



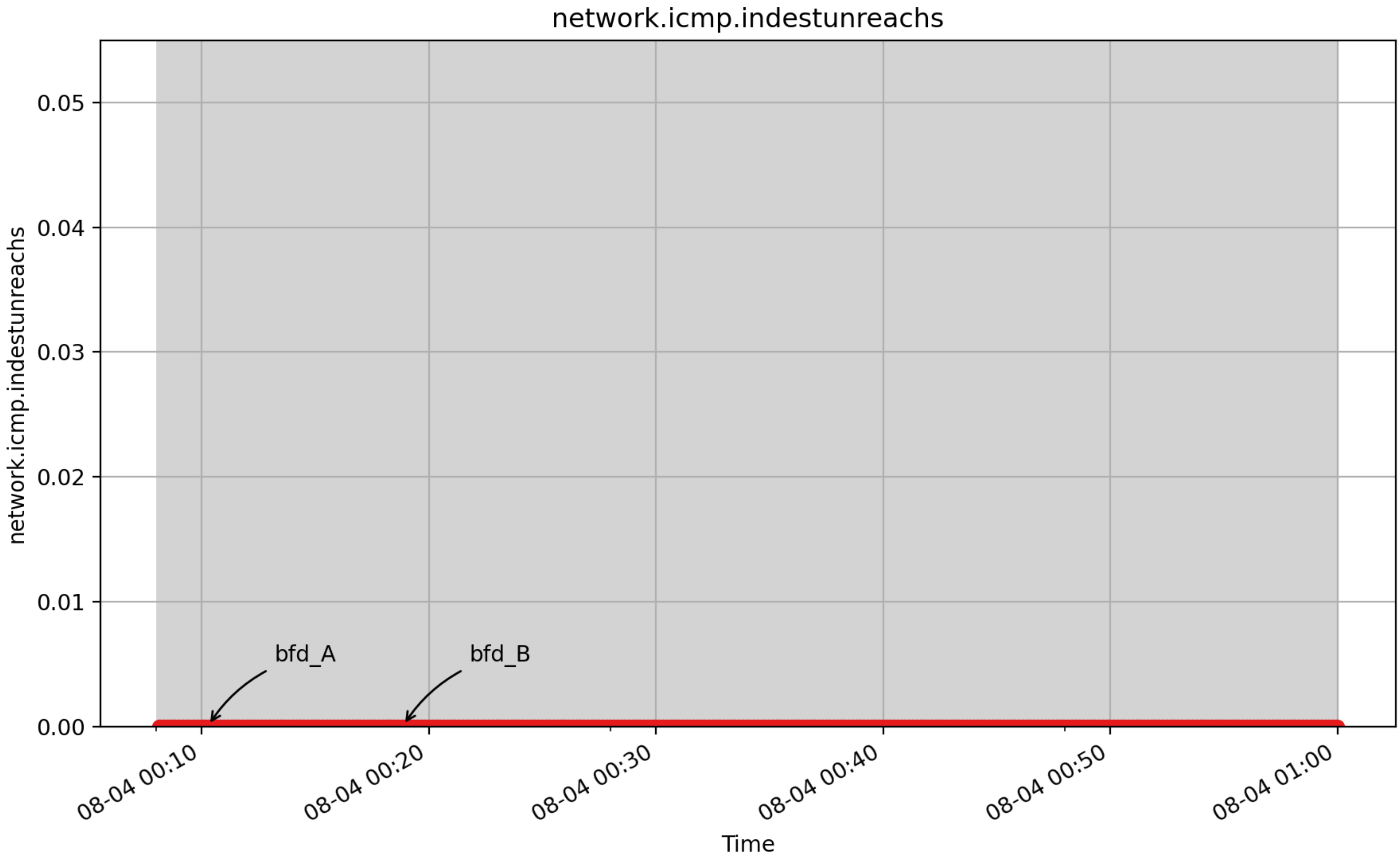
network.all.in.packets: Sum of packets column on the "Receive" side of /proc/net/dev for network interfaces deemed to be 'physical' interfaces, using regular expression pattern described in the $PCP_SYSCONF_DIR/linux/interfaces.conf file. (count - U64) - *rate converted*



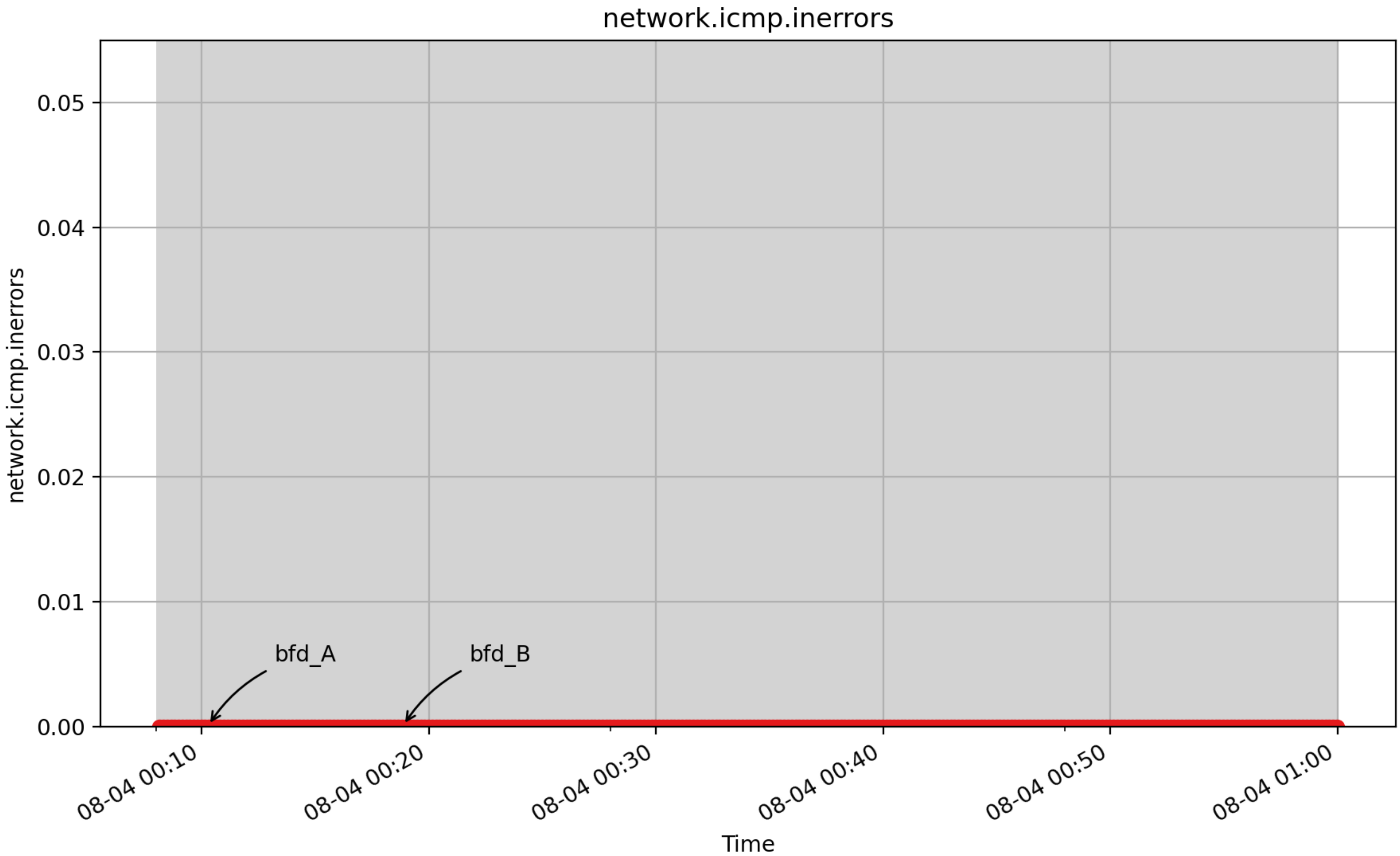
network.all.out.bytes: Sum of bytes column on the "Transmit" side of /proc/net/dev for network interfaces deemed to be 'physical' interfaces, using regular expression pattern described in the $PCP_SYSCONF_DIR/linux/interfaces.conf file. (byte - U64) - *rate converted*



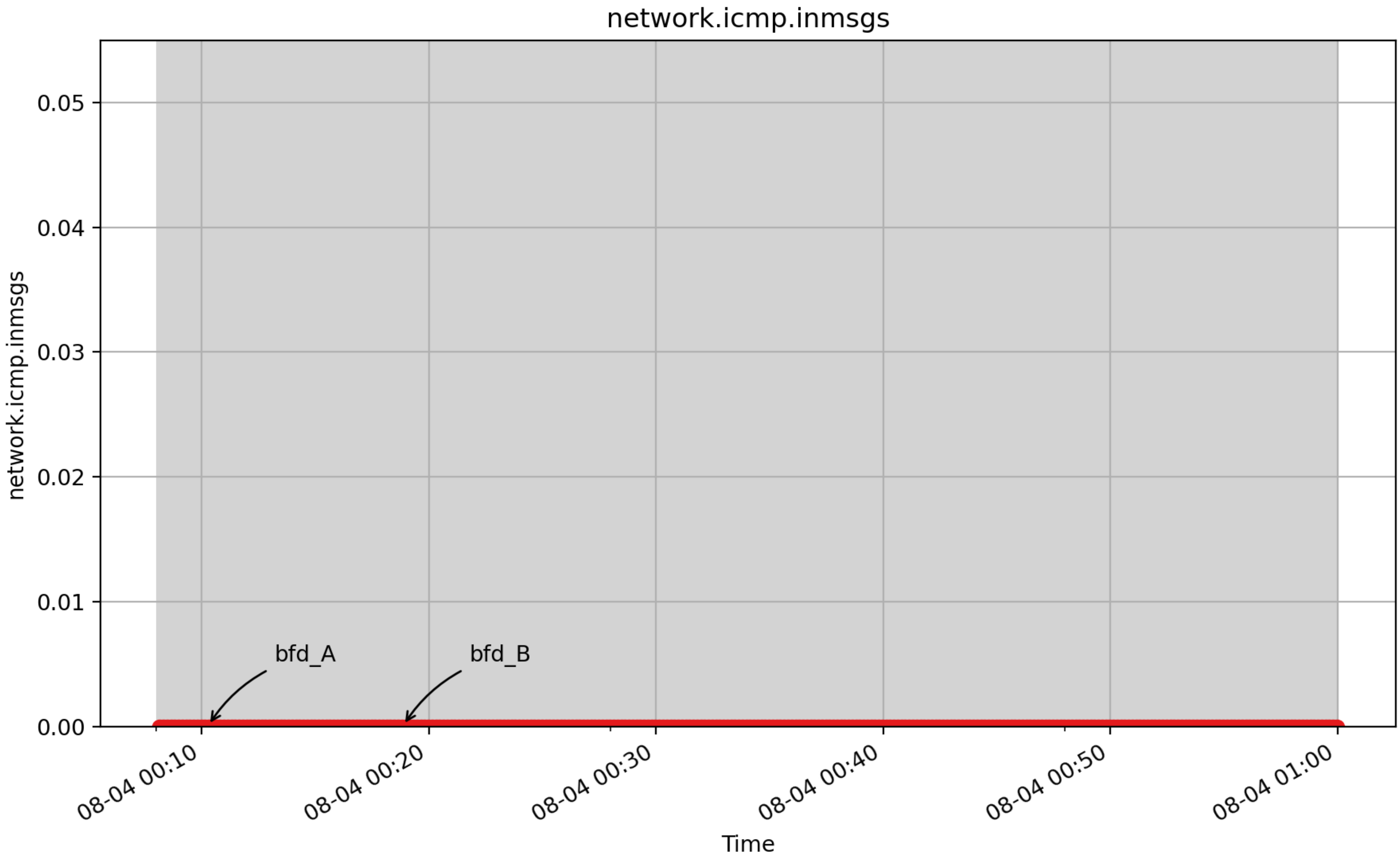
network.all.out.packets: Sum of packets column on the "Transmit" side of /proc/net/dev for network interfaces deemed to be 'physical' interfaces, using regular expression pattern described in the $PCP_SYSCONF_DIR/linux/interfaces.conf file. (count - U64) - *rate converted*



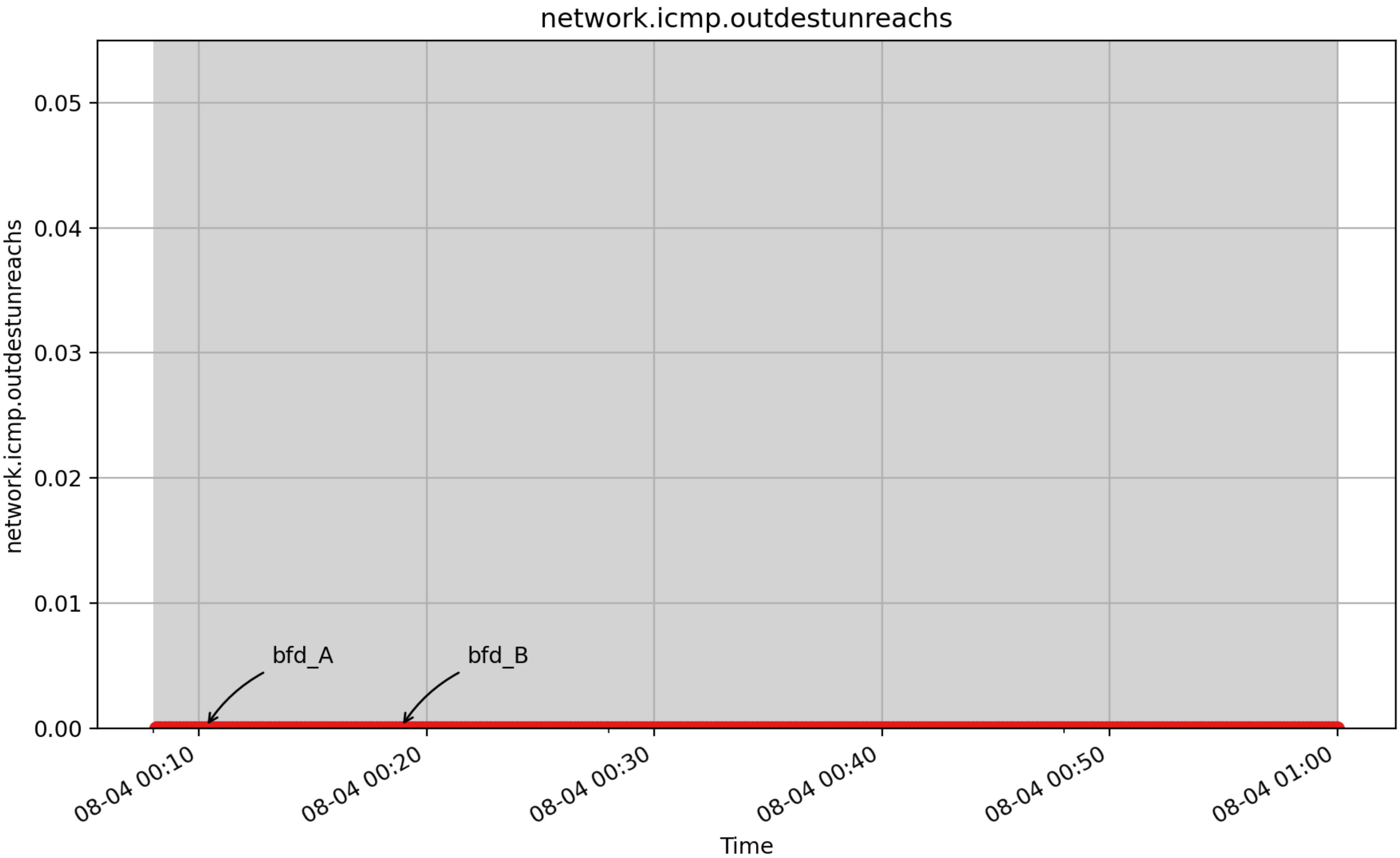
network.icmp.indestunreachs: count of icmp indestunreachs (count - U64) - *rate converted*



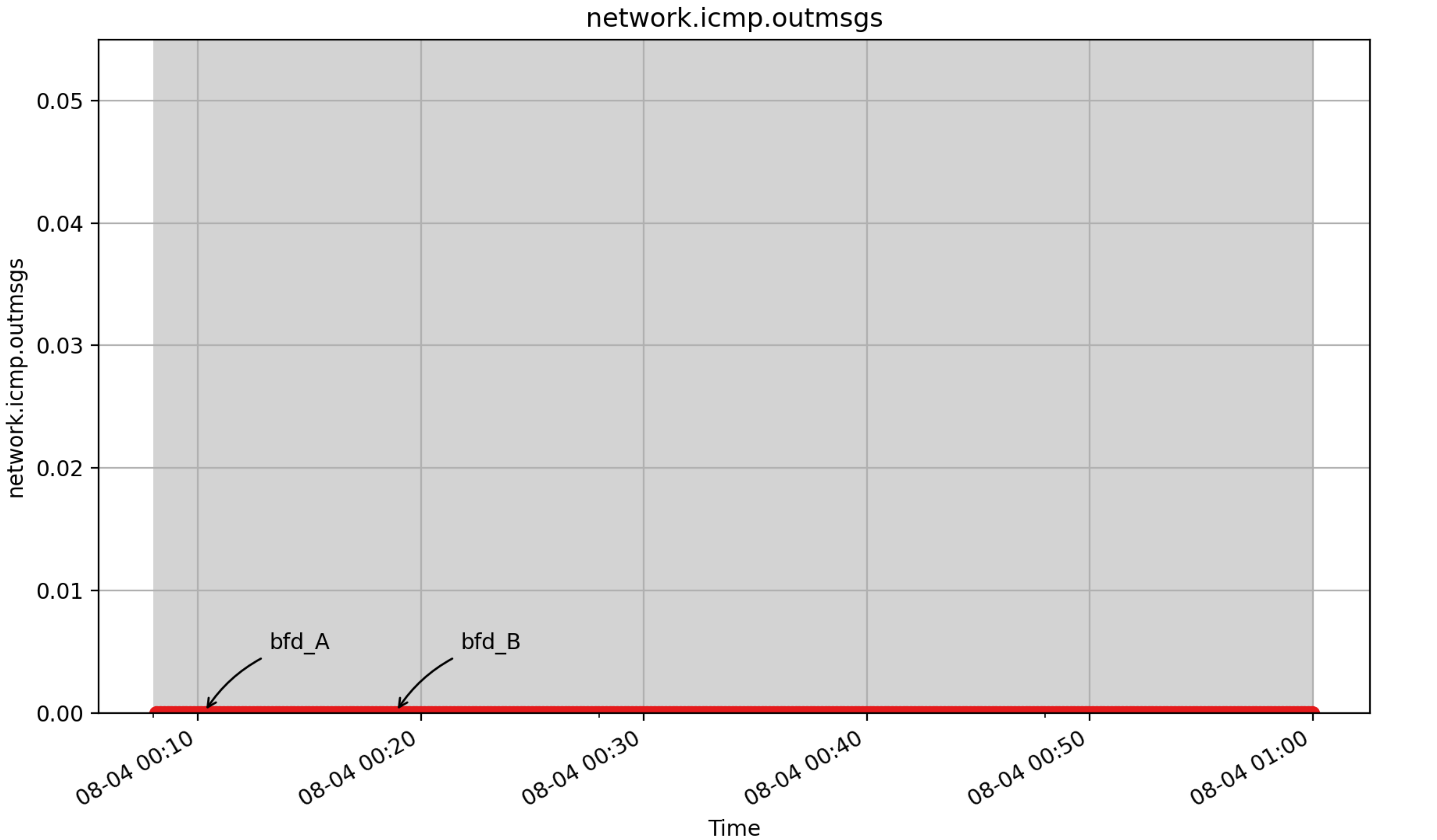
network.icmp.inerrors: count of icmp inerrors (count - U64) - *rate converted*



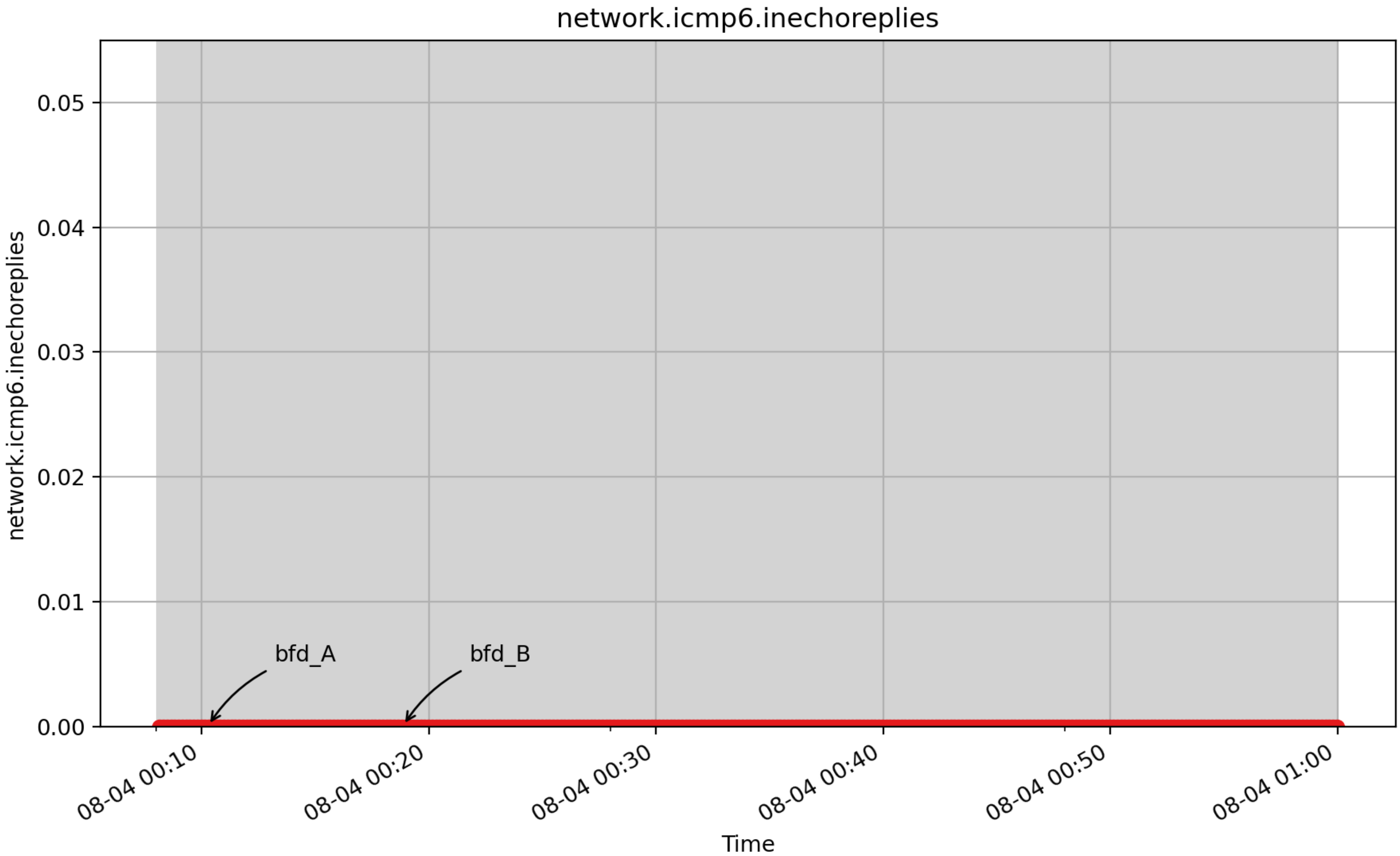
network.icmp.inmsgs: count of icmp inmsgs (count - U64) - *rate converted*



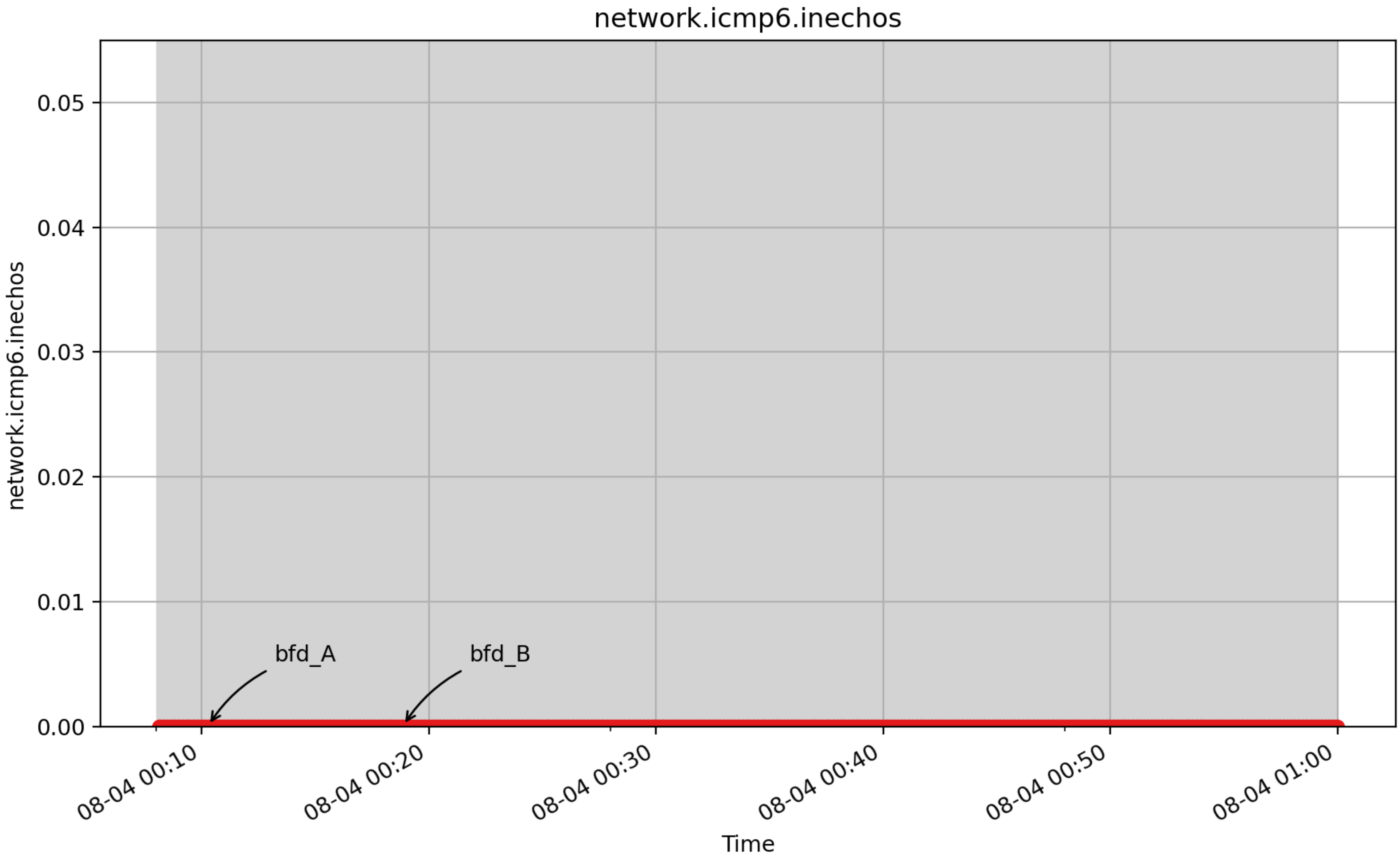
network.icmp.outdestunreachs: count of icmp outdestunreachs (count - U64) - *rate converted*



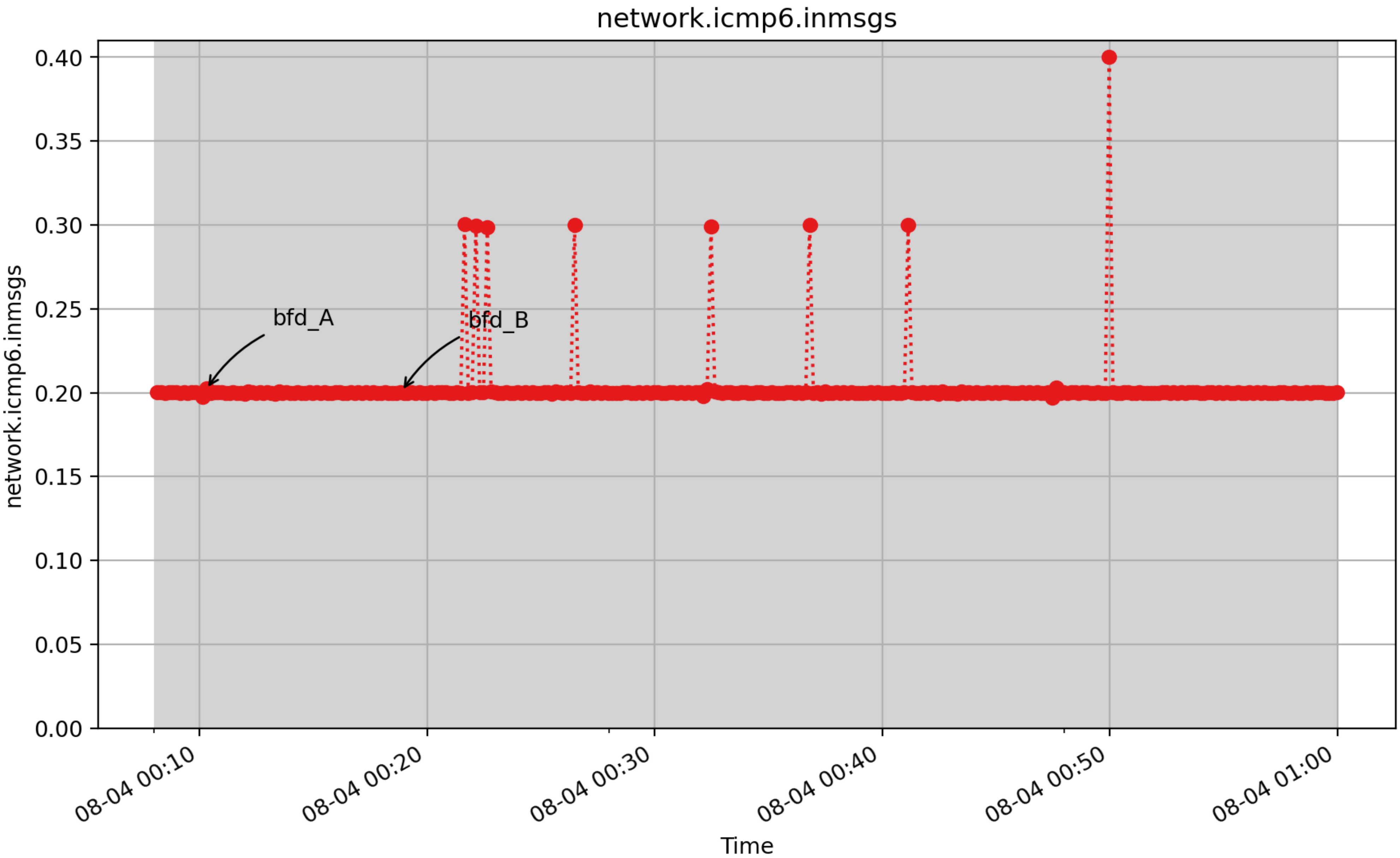
network.icmp.outmsgs: count of icmp outmsgs (count - U64) - *rate converted*



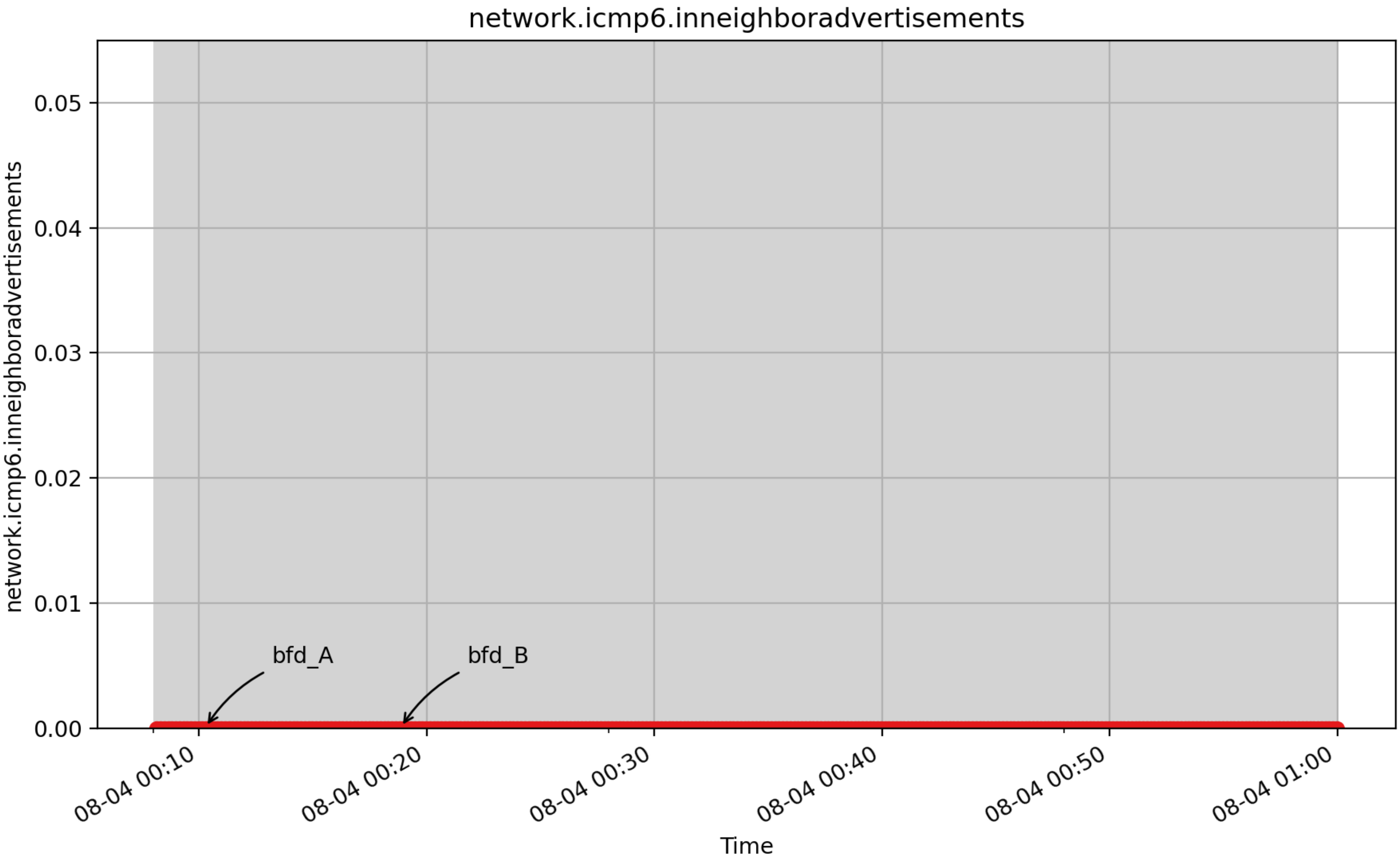
network.icmp6.inechoreplies: count of icmp6 inechoreplies (count - U64) - *rate converted*



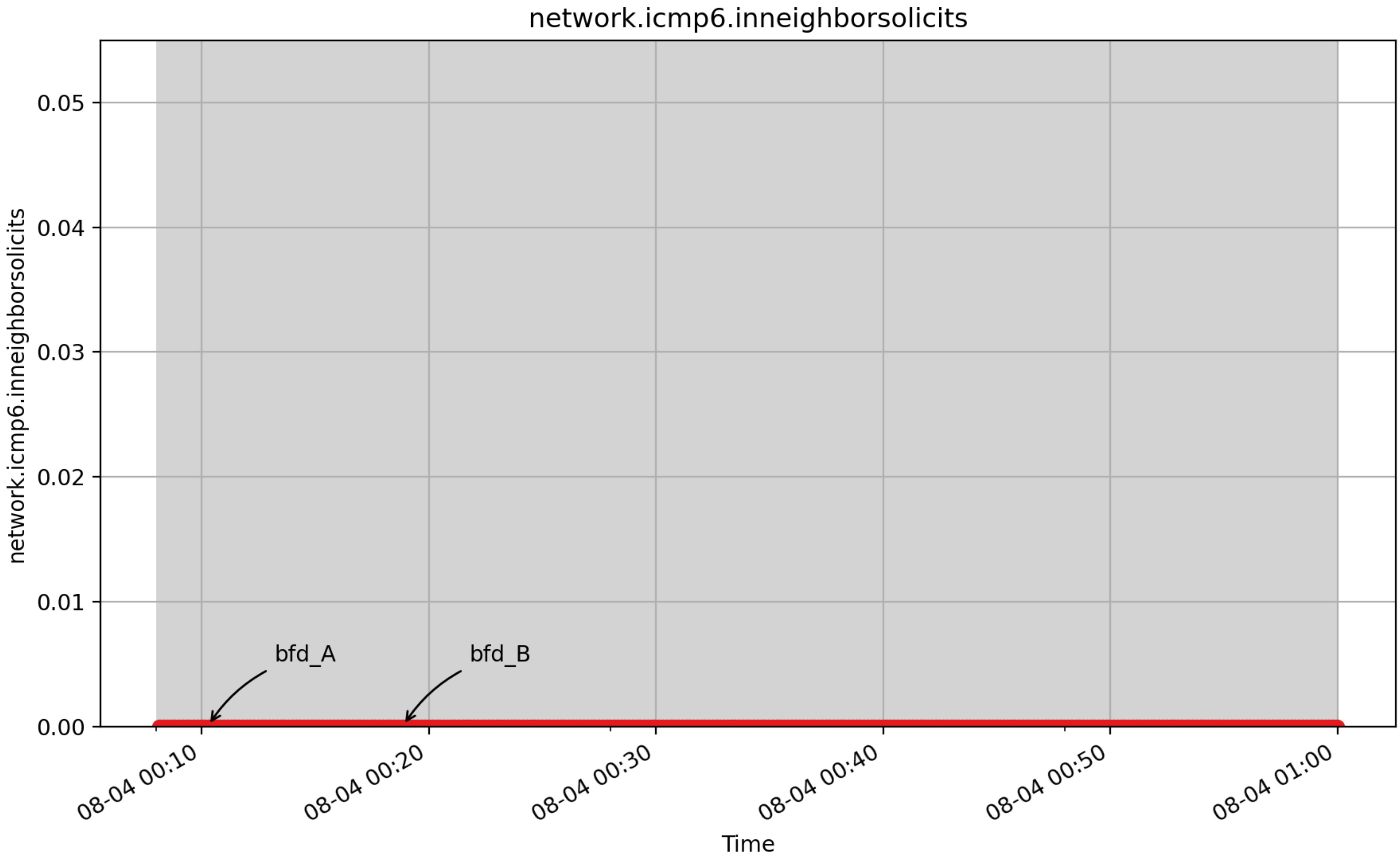
network.icmp6.inechos: count of icmp6 inechos (count - U64) - *rate converted*



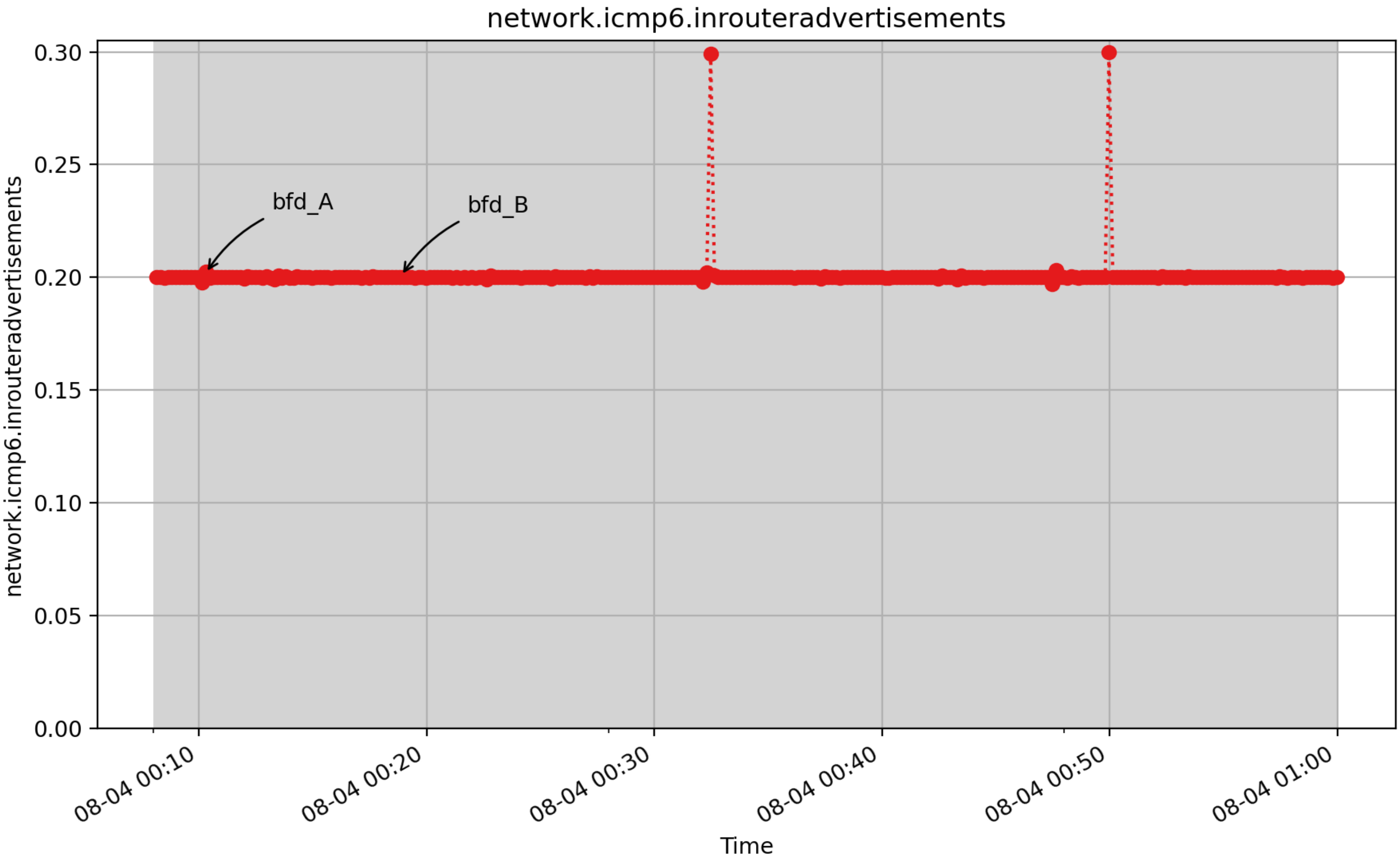
network.icmp6.inmsgs: count of icmp6 inmsgs (count - U64) - *rate converted*



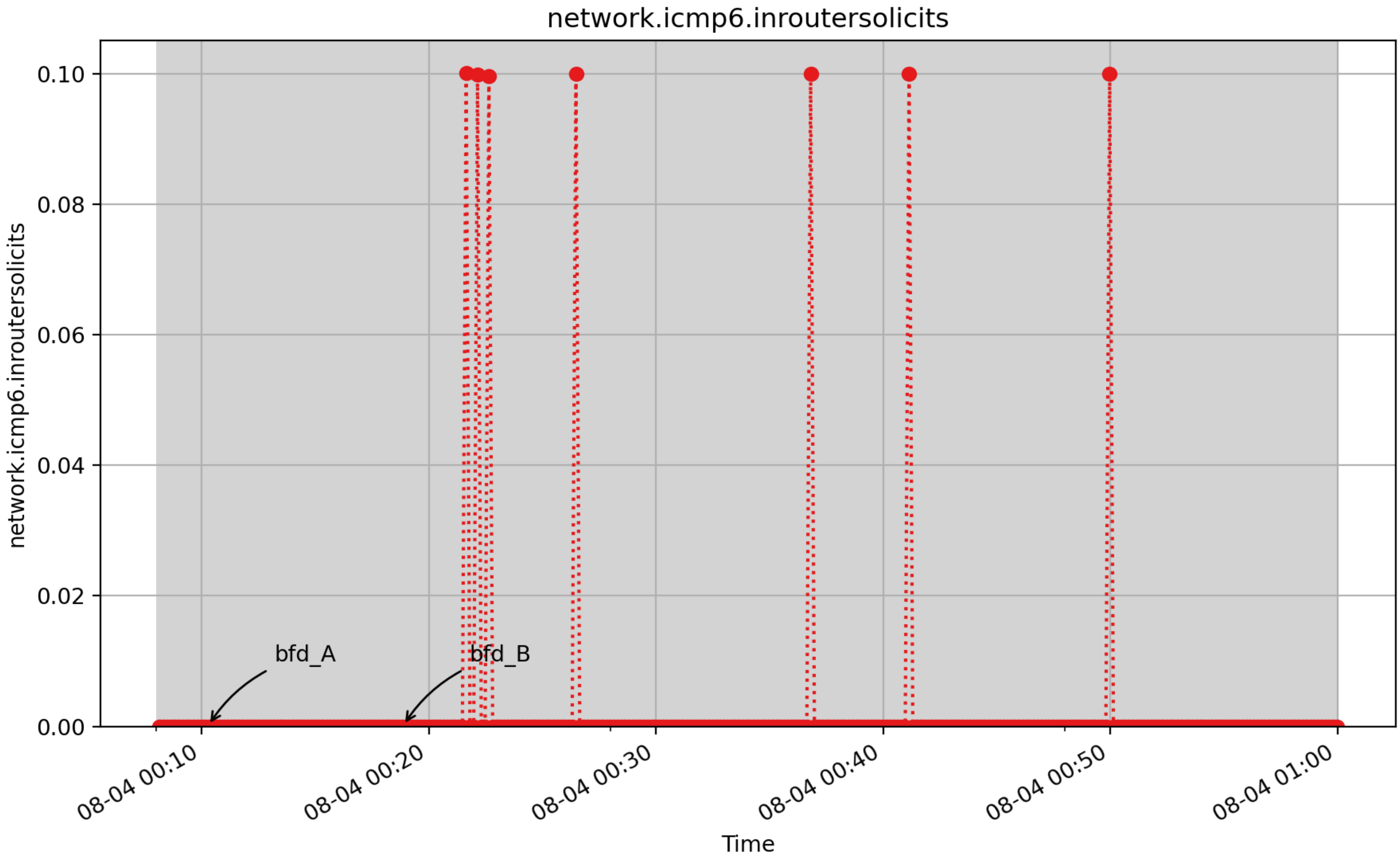
network.icmp6.inneighboradvertisements: count of icmp6 inneighboradvertisements (count - U64) - *rate converted*



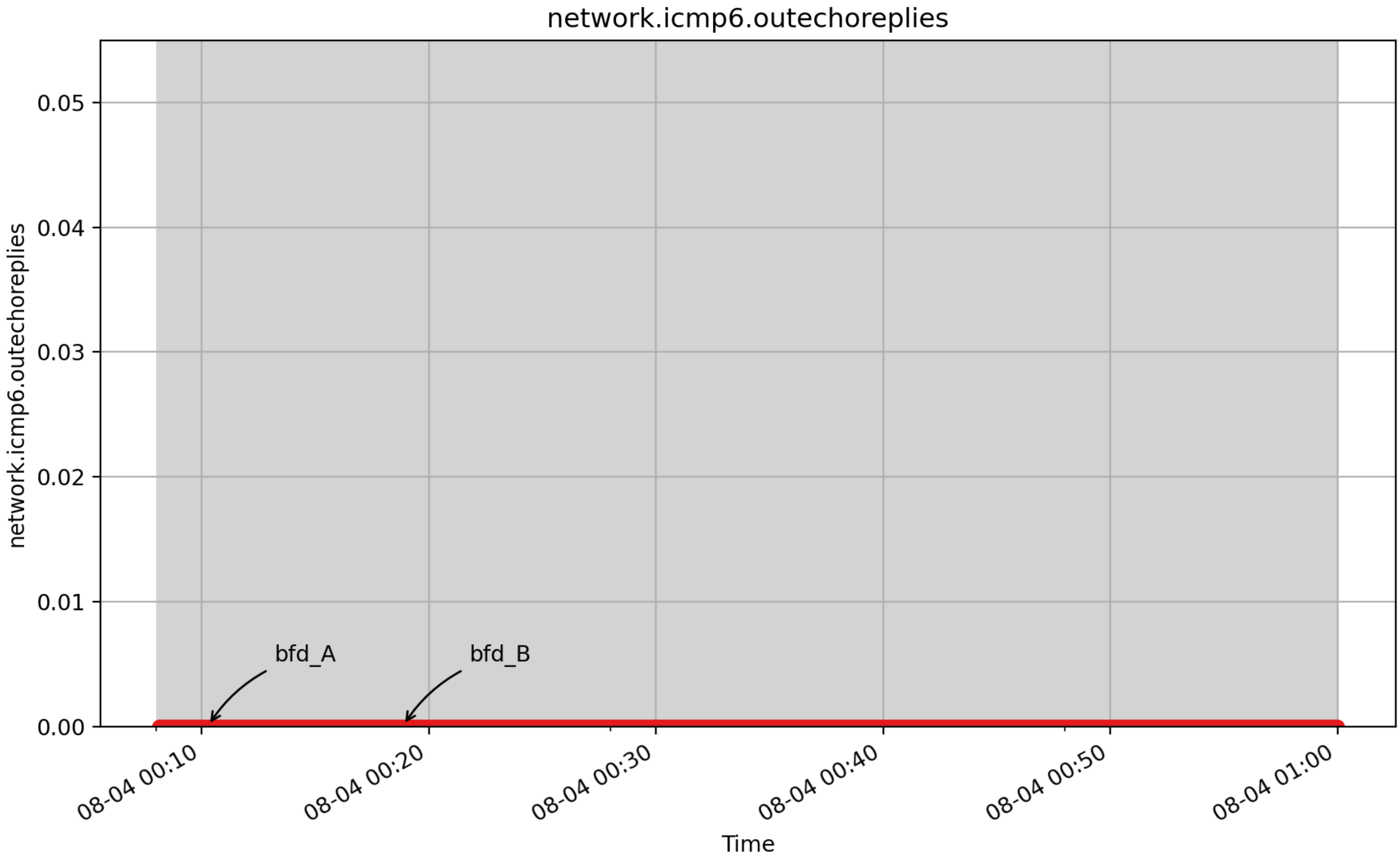
network.icmp6.inneighborsolicits: count of icmp6 inneighborsolicits (count - U64) - *rate converted*



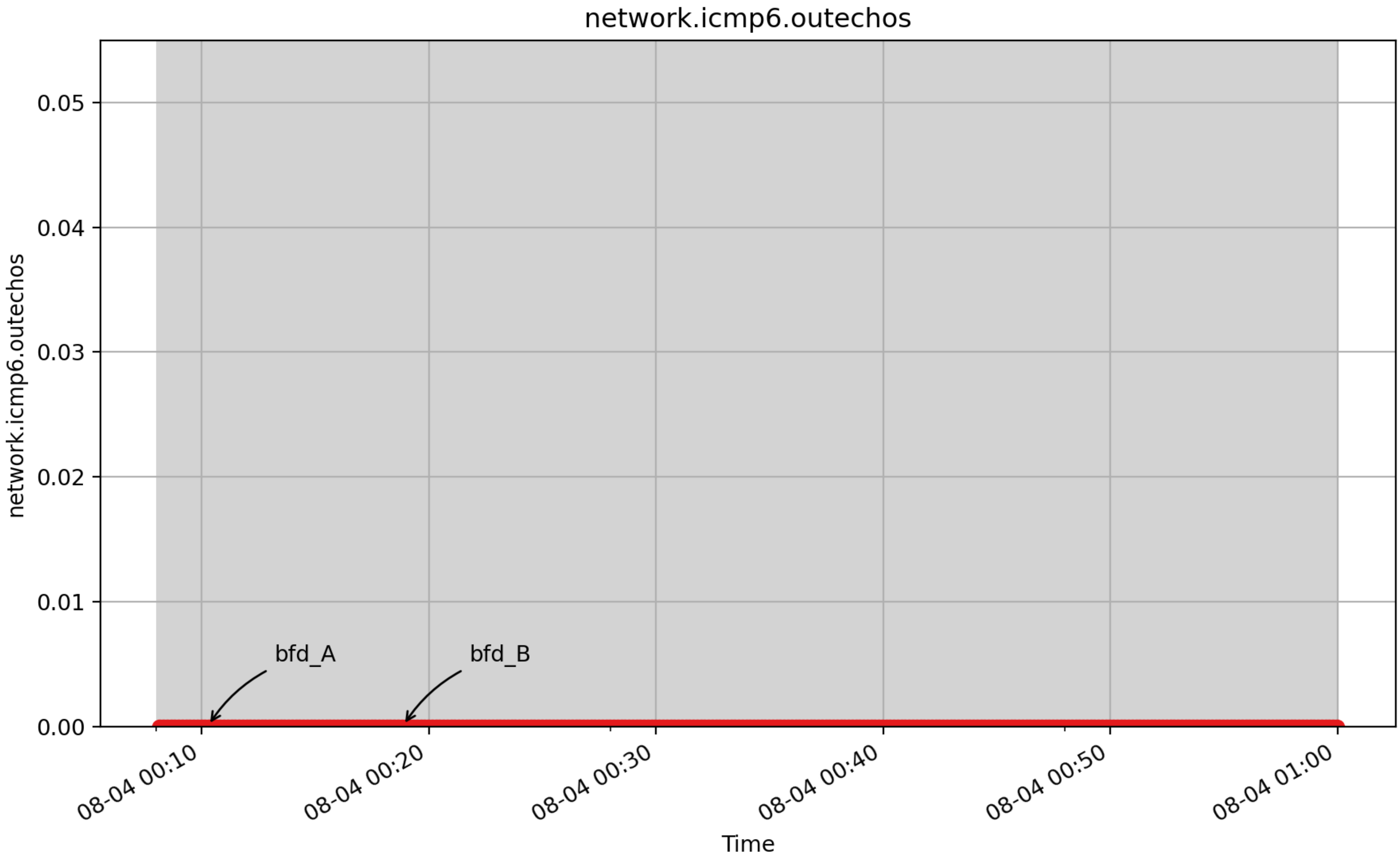
network.icmp6.inrouteradvertisements: count of icmp6 inrouteradvertisements (count - U64) - *rate converted*



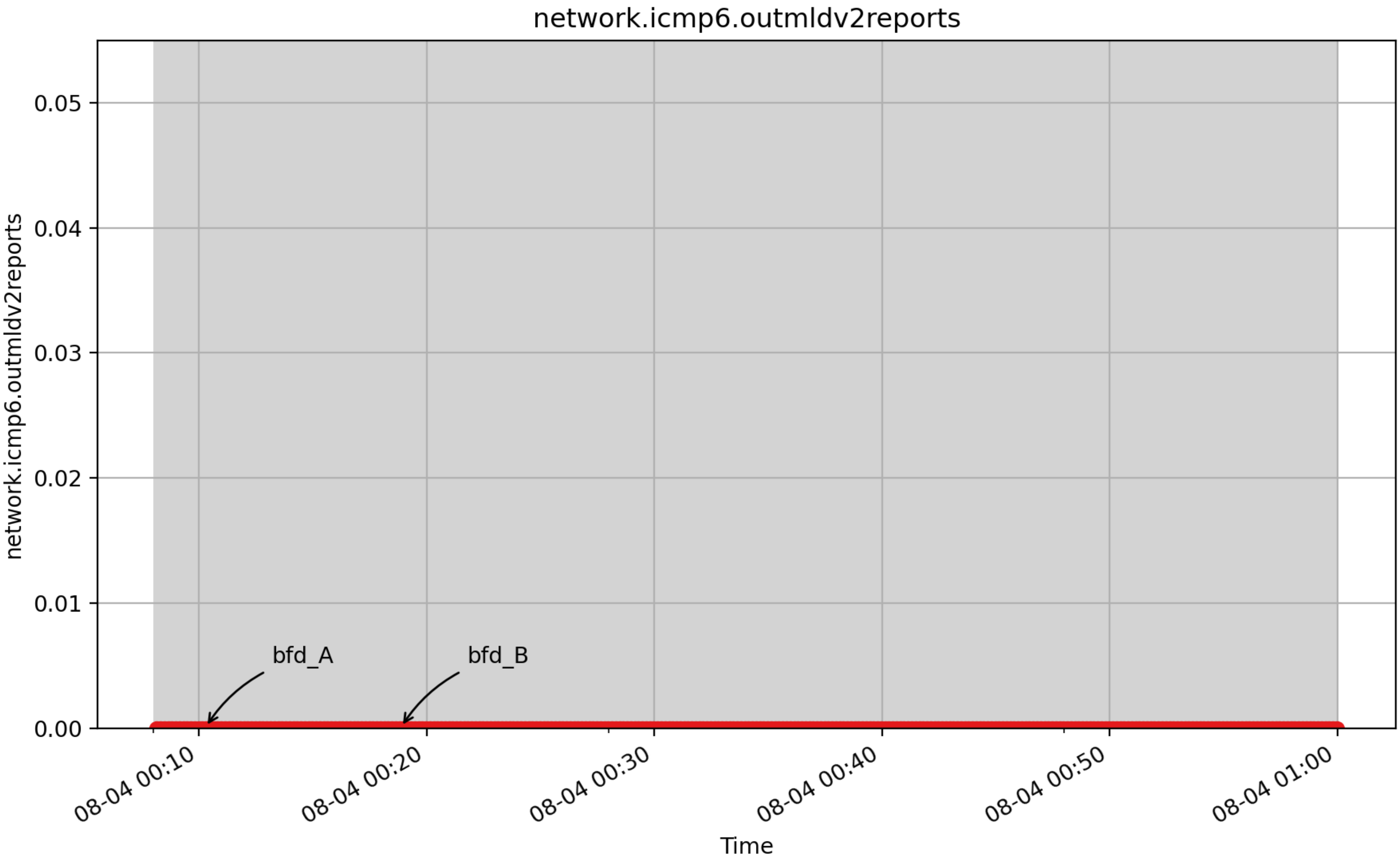
network.icmp6.inroutersolicits: count of icmp6 inroutersolicits (count - U64) - *rate converted*



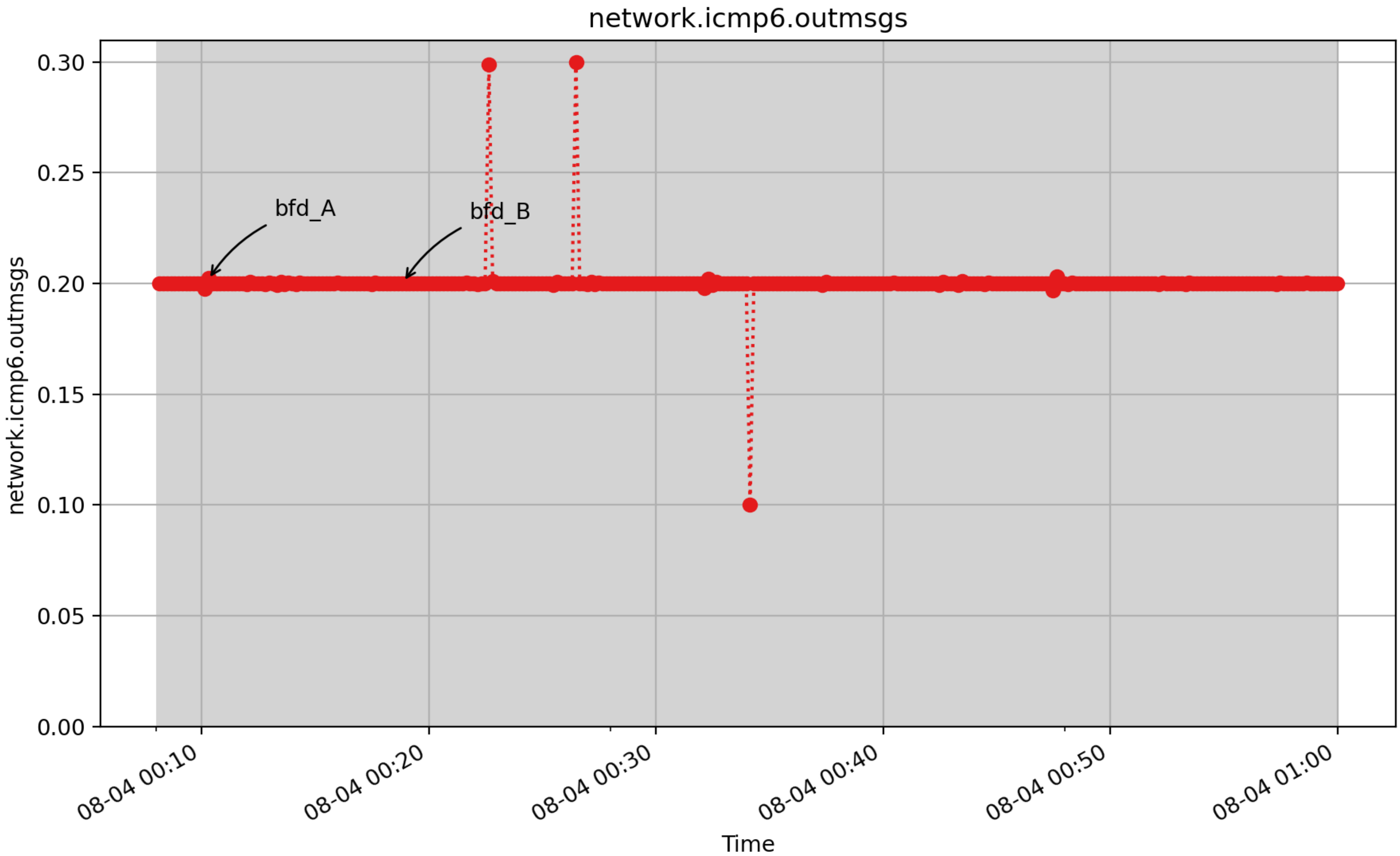
network.icmp6.outechoreplies: count of icmp6 outechoreplies (count - U64) - *rate converted*



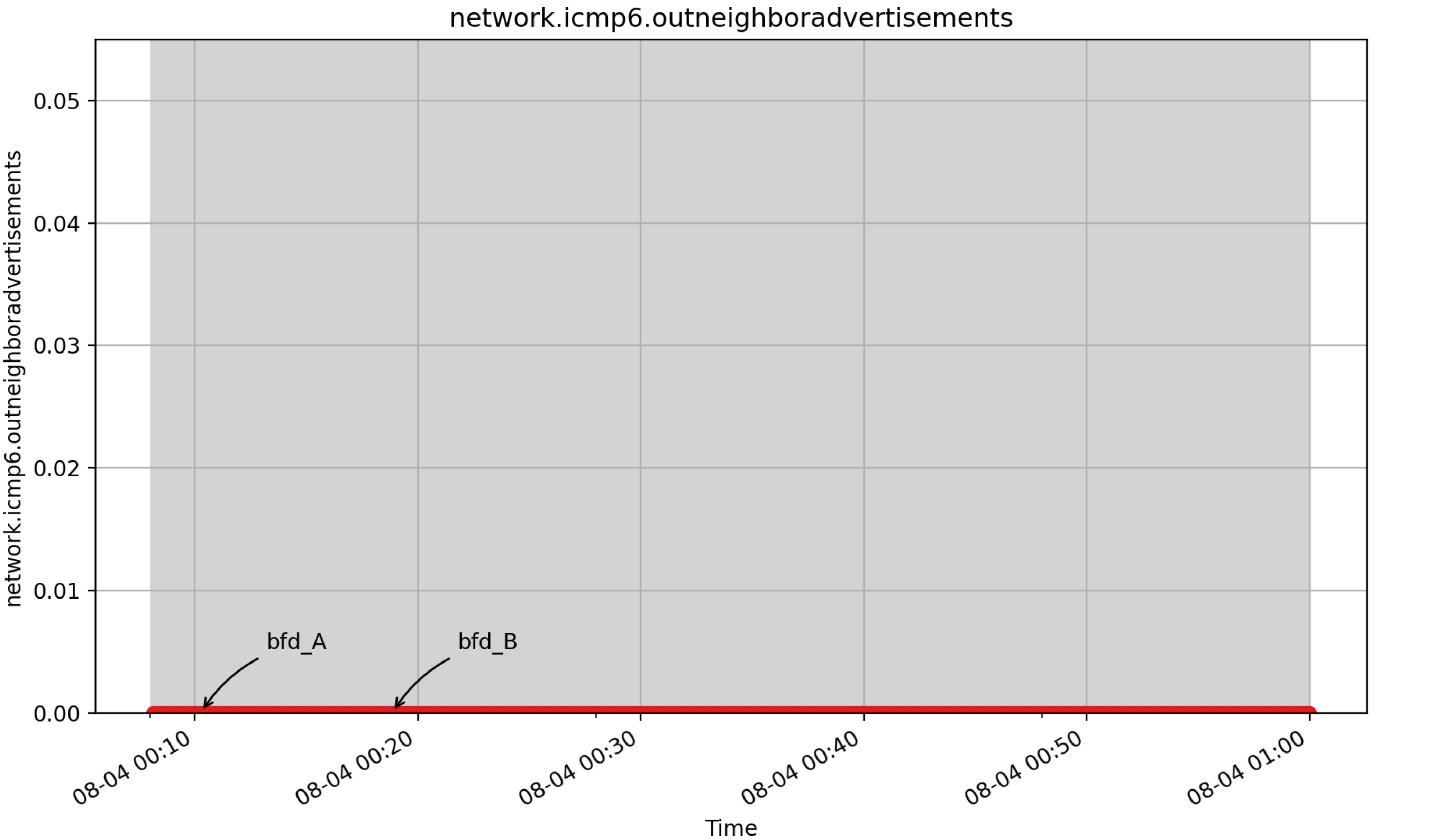
network.icmp6.outechos: count of icmp6 outechos (count - U64) - *rate converted*



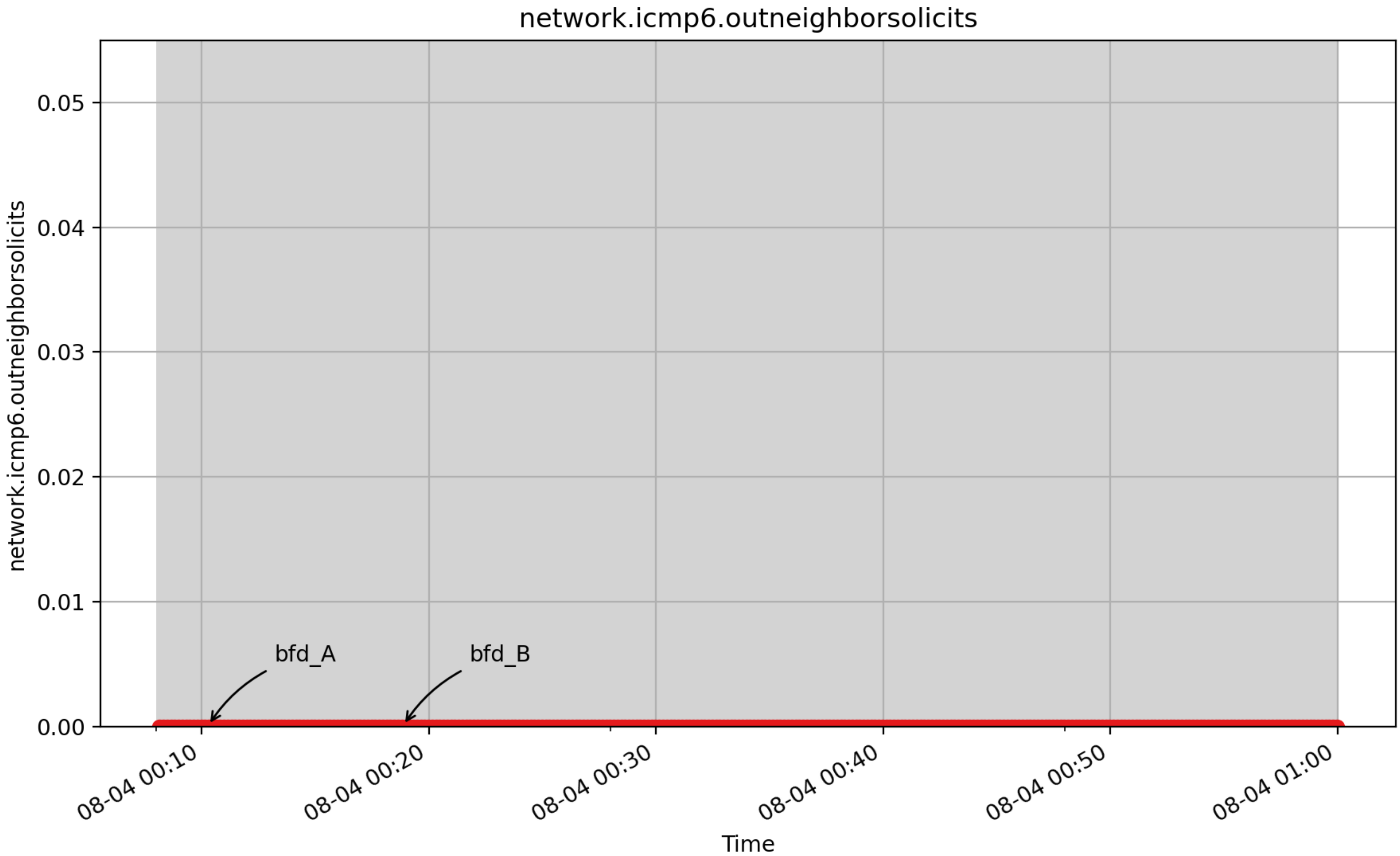
network.icmp6.outmldv2reports: count of icmp6 outmldv2reports (count - U64) - *rate converted*



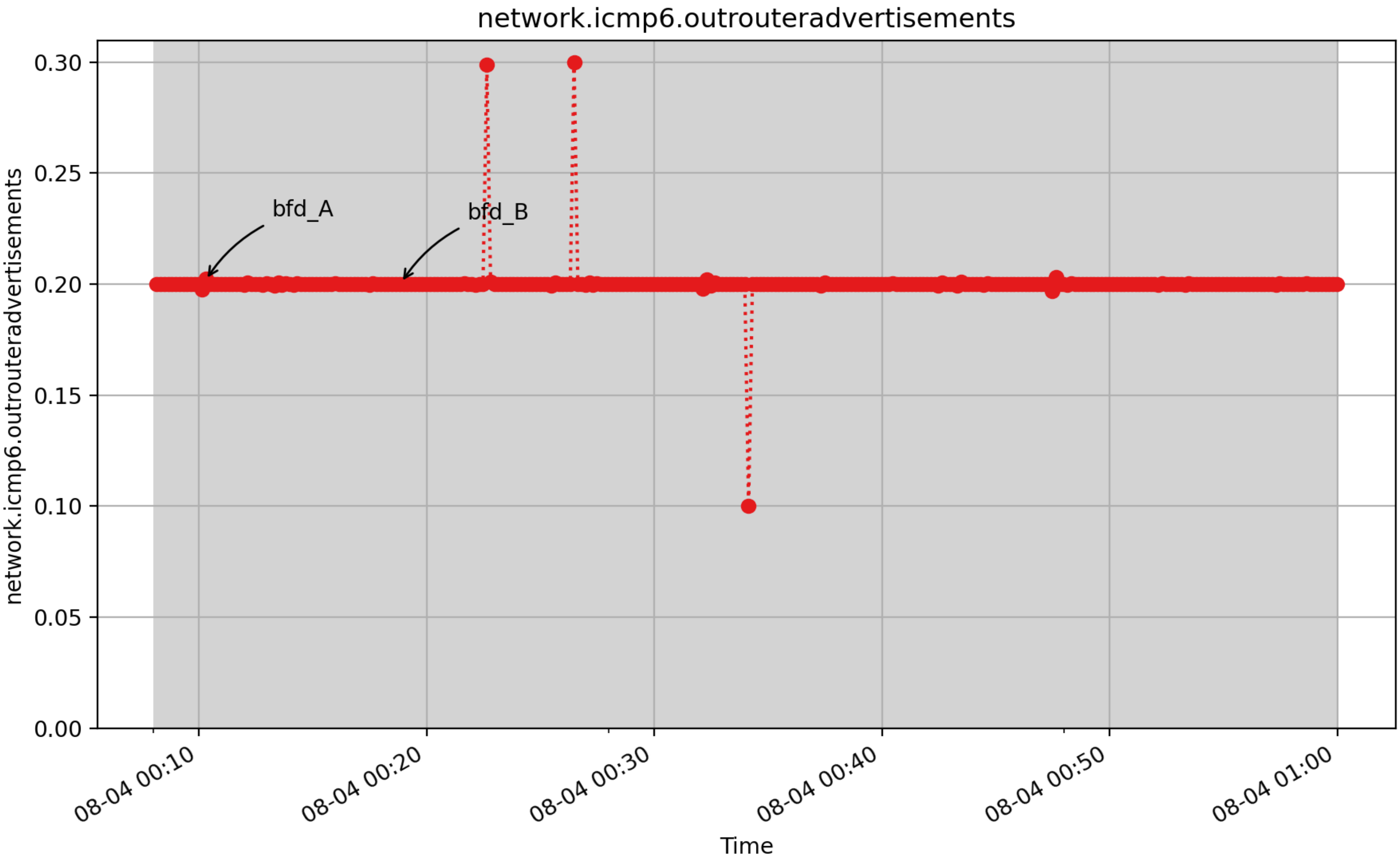
network.icmp6.outmsgs: count of icmp6 outmsgs (count - U64) - *rate converted*



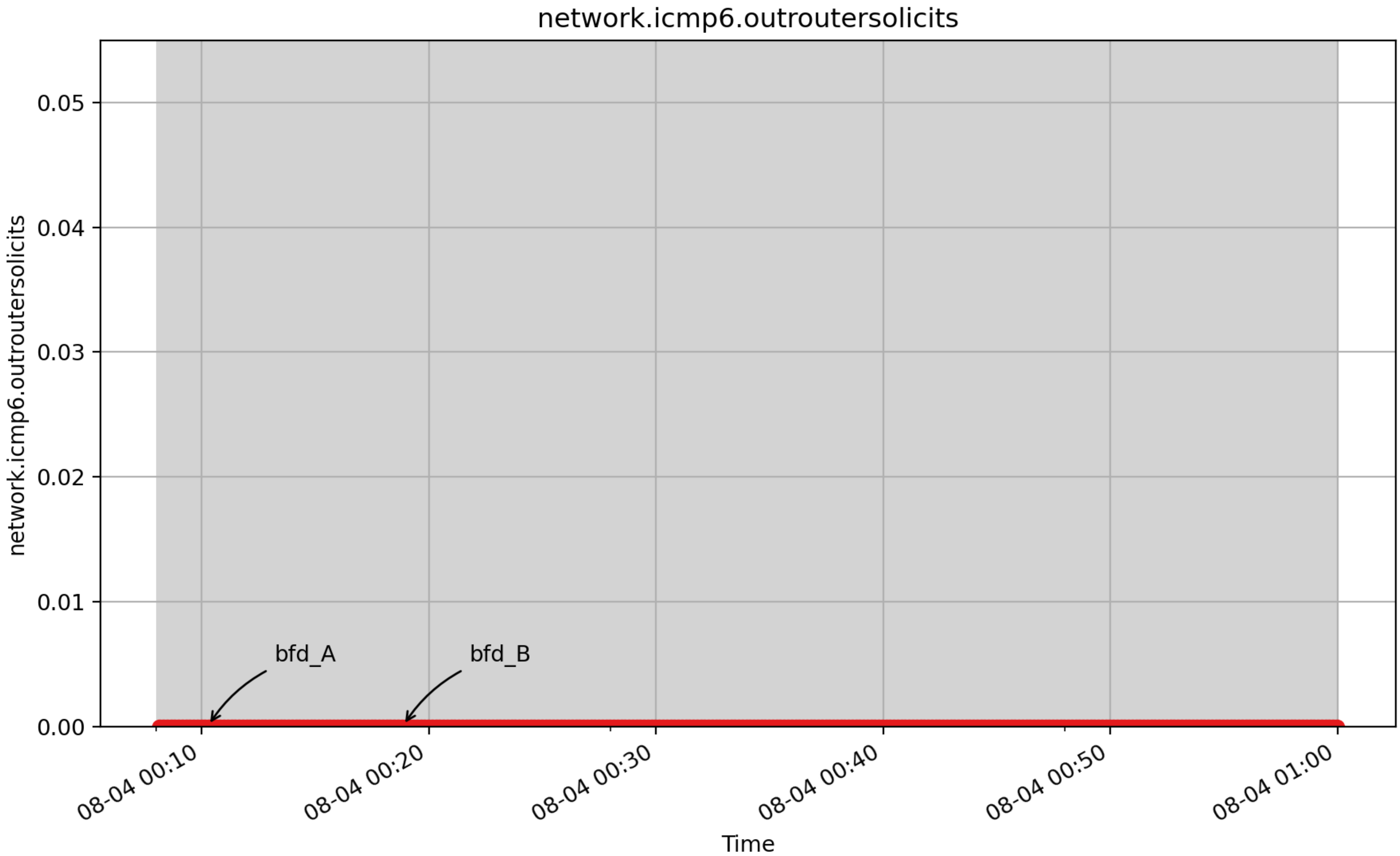
network.icmp6.outneighboradvertisements: count of icmp6 outneighboradvertisements (count - U64) - *rate converted*



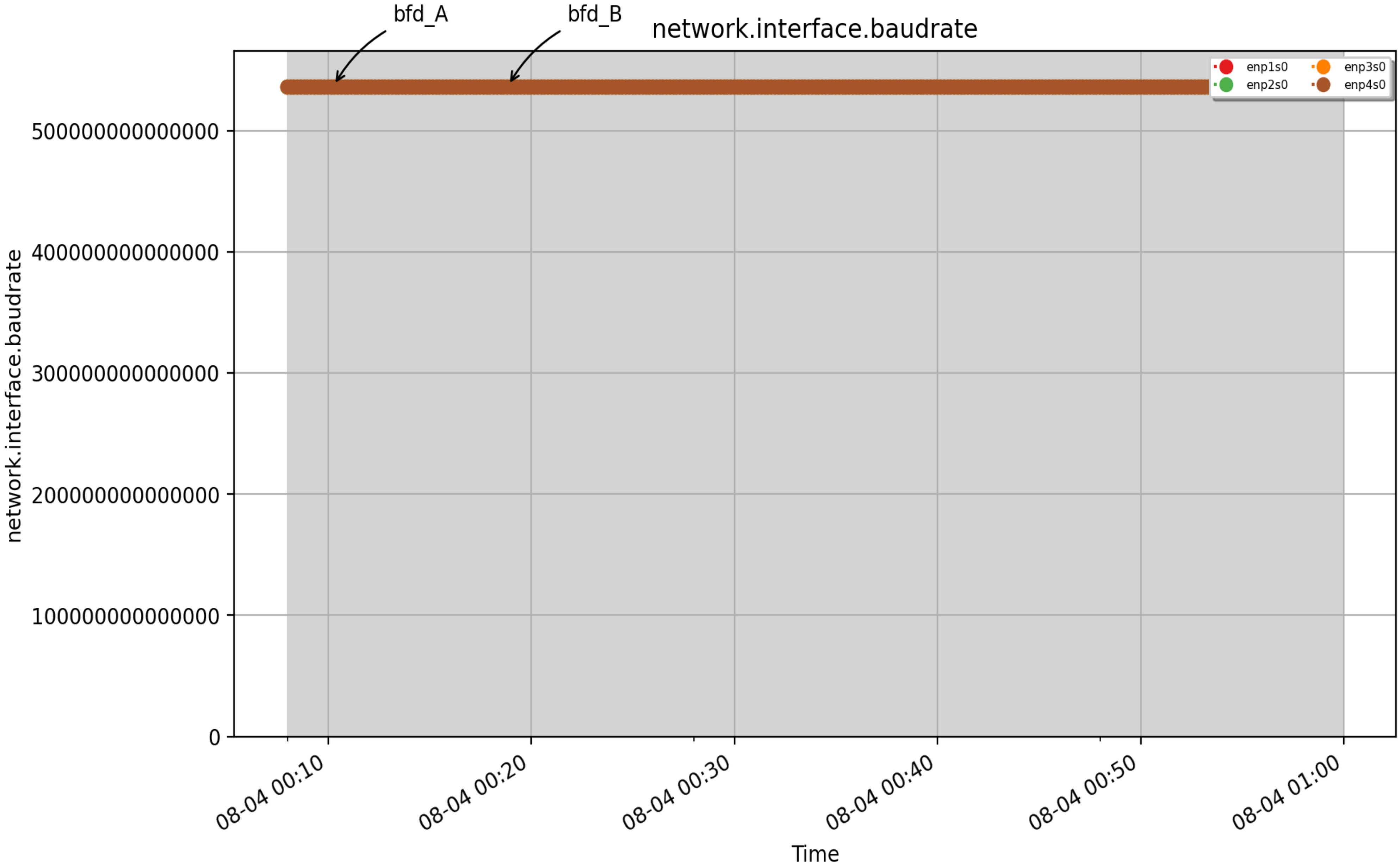
network.icmp6.outneighborsolicits: count of icmp6 outneighborsolicits (count - U64) - *rate converted*



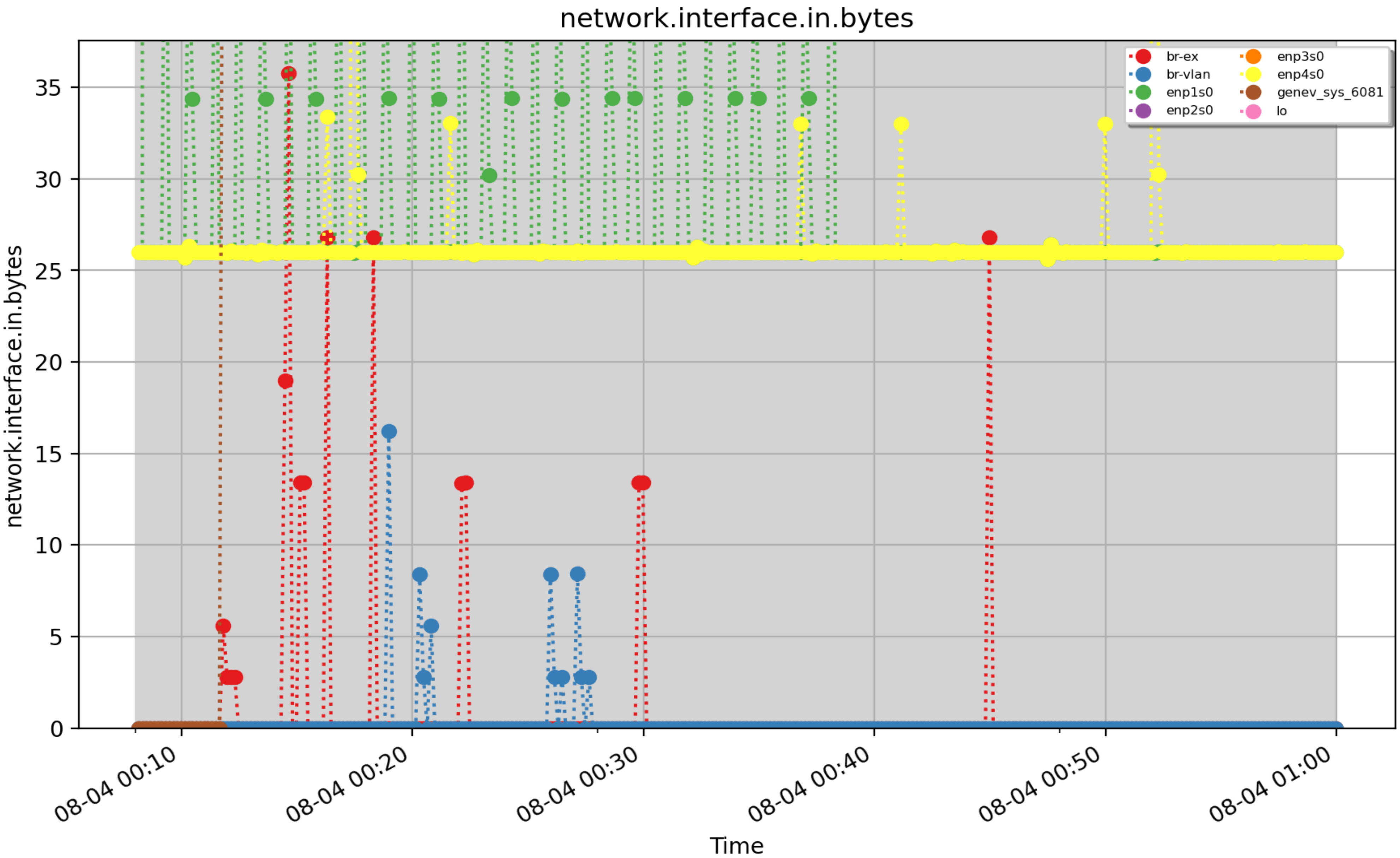
network.icmp6.outrouteradvertisements: count of icmp6 outrouteradvertisements (count - U64) - *rate converted*



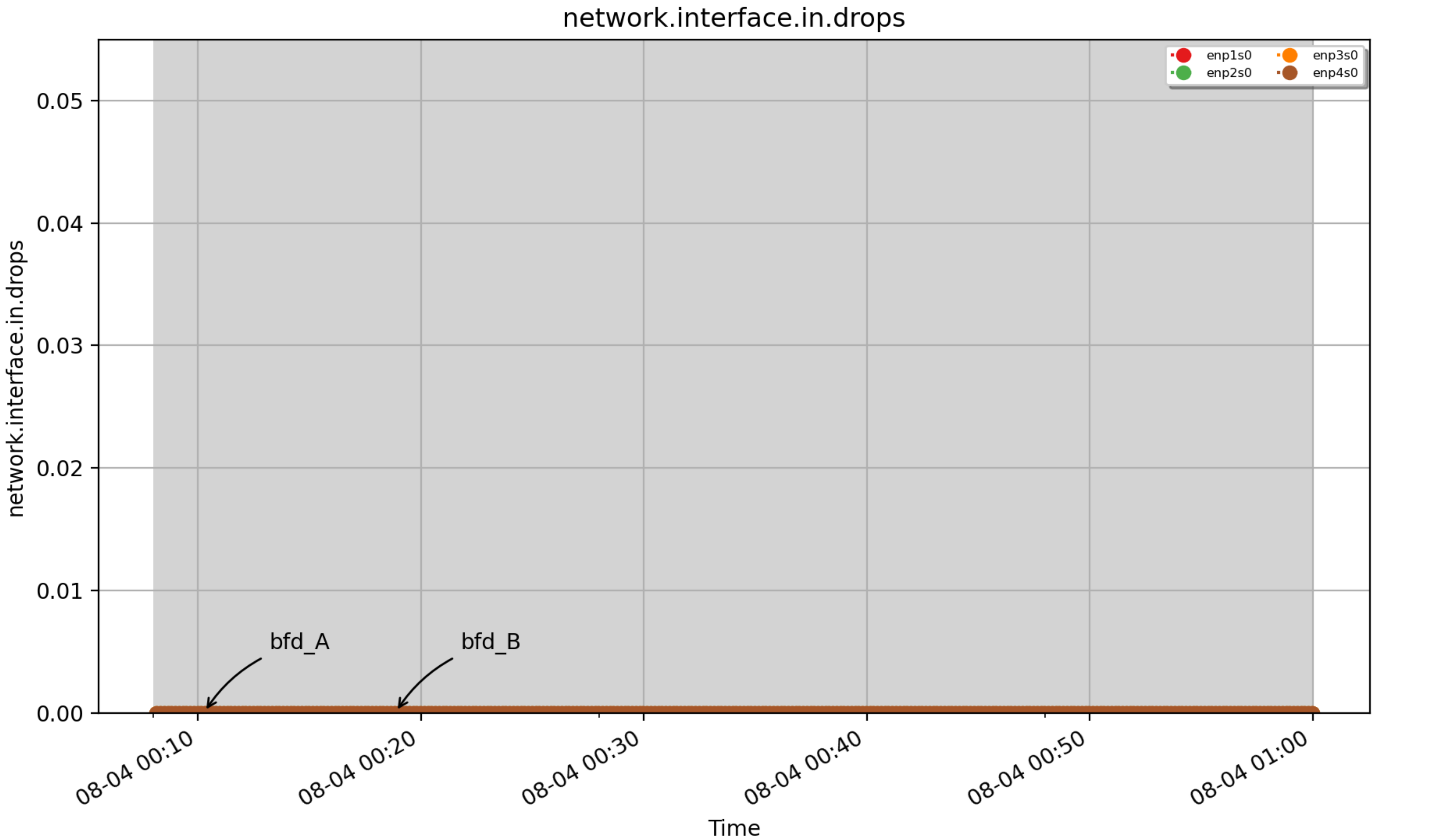
network.icmp6.outroutersolicits: count of icmp6 outroutersolicits (count - U64) - *rate converted*



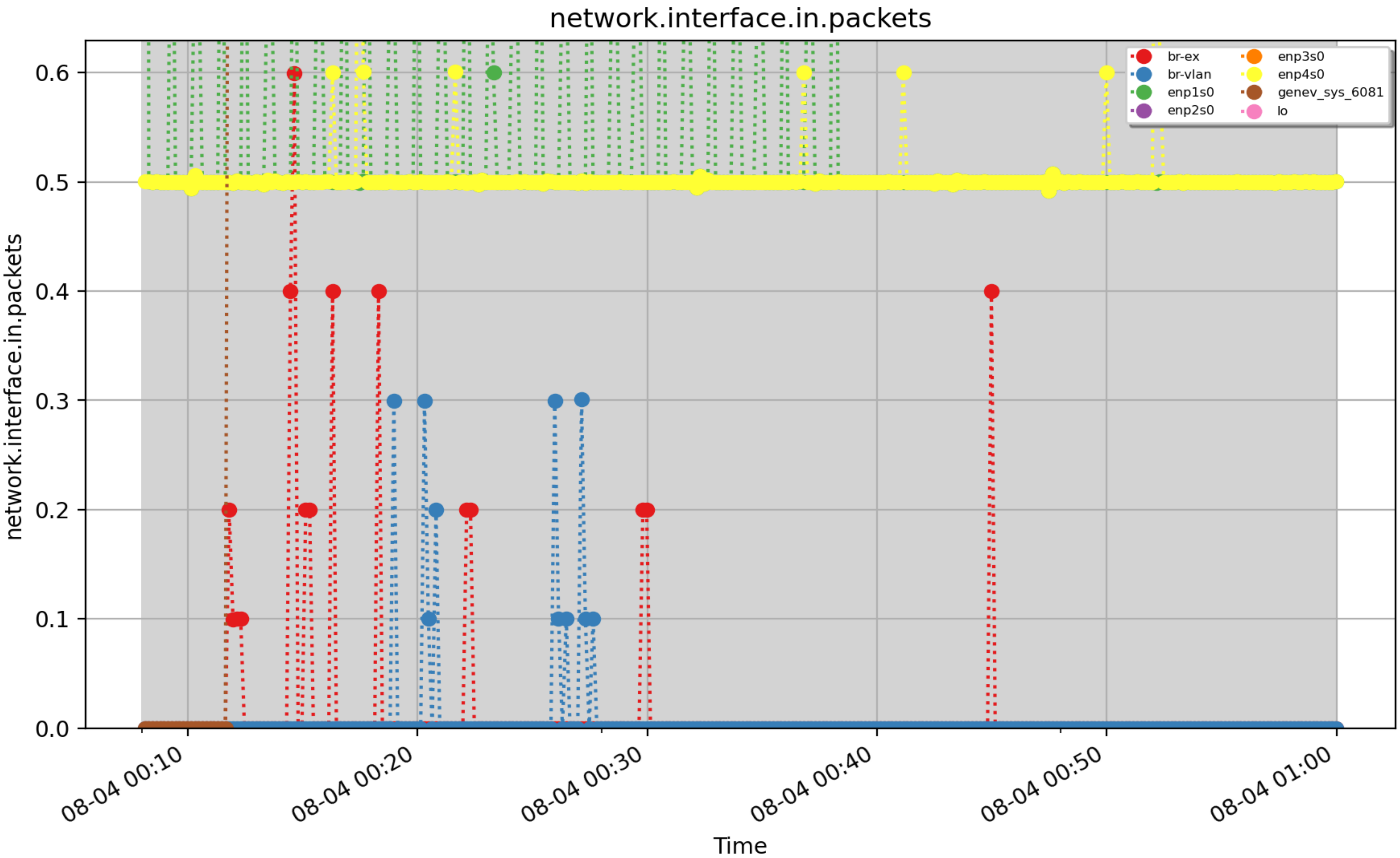
network.interface.baudrate: The linespeed on the network interface, as reported by the kernel, scaled up from Megabits/second to bits/second and divided by 8 to convert to bytes/second. See also network.interface.speed for the Megabytes/second value. (byte / sec - U64)



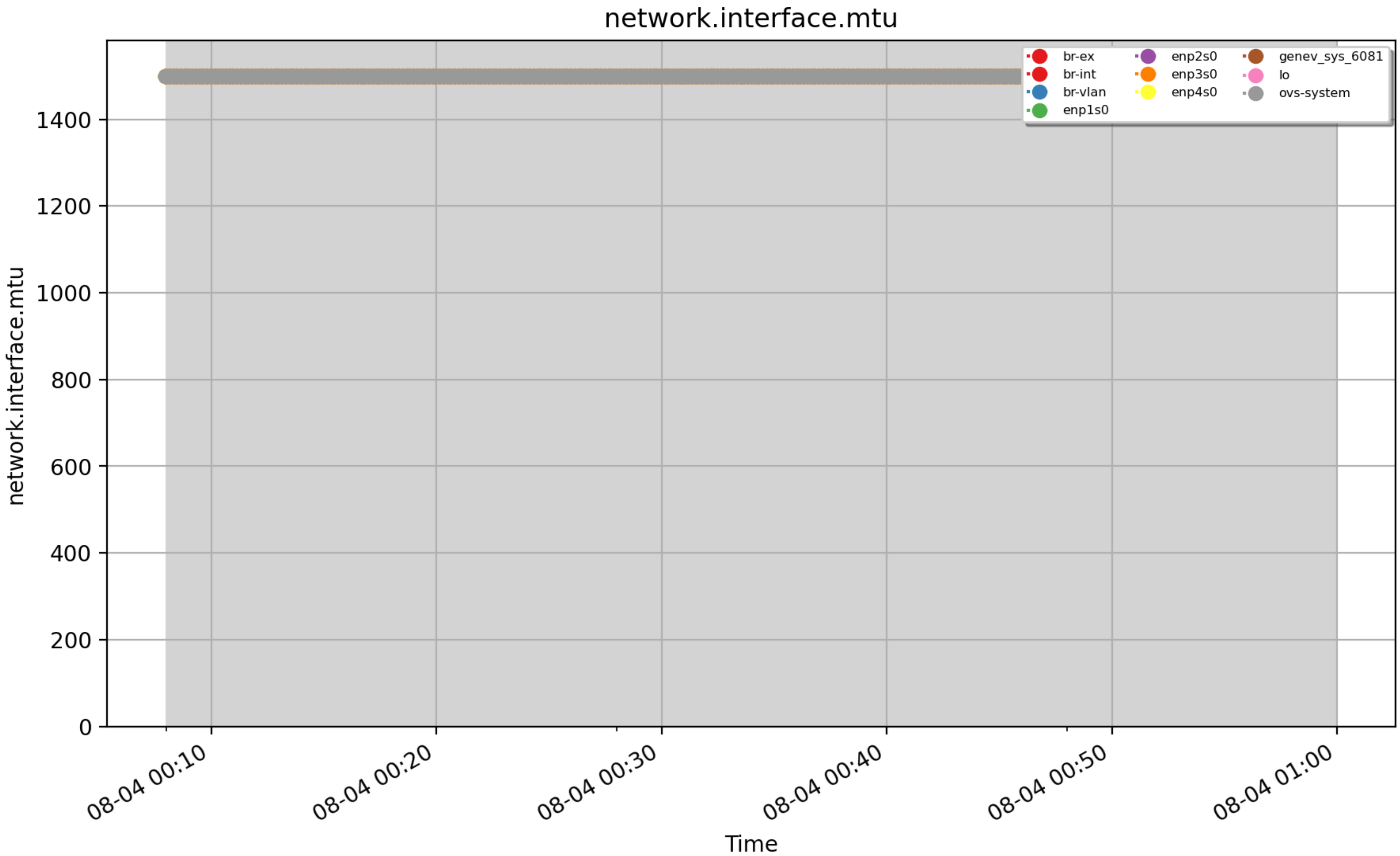
network.interface.in.bytes: bytes column on the "Receive" side of /proc/net/dev (stats->rx_bytes counter in rtnl_link_stats64) (byte - U64) - *rate converted*



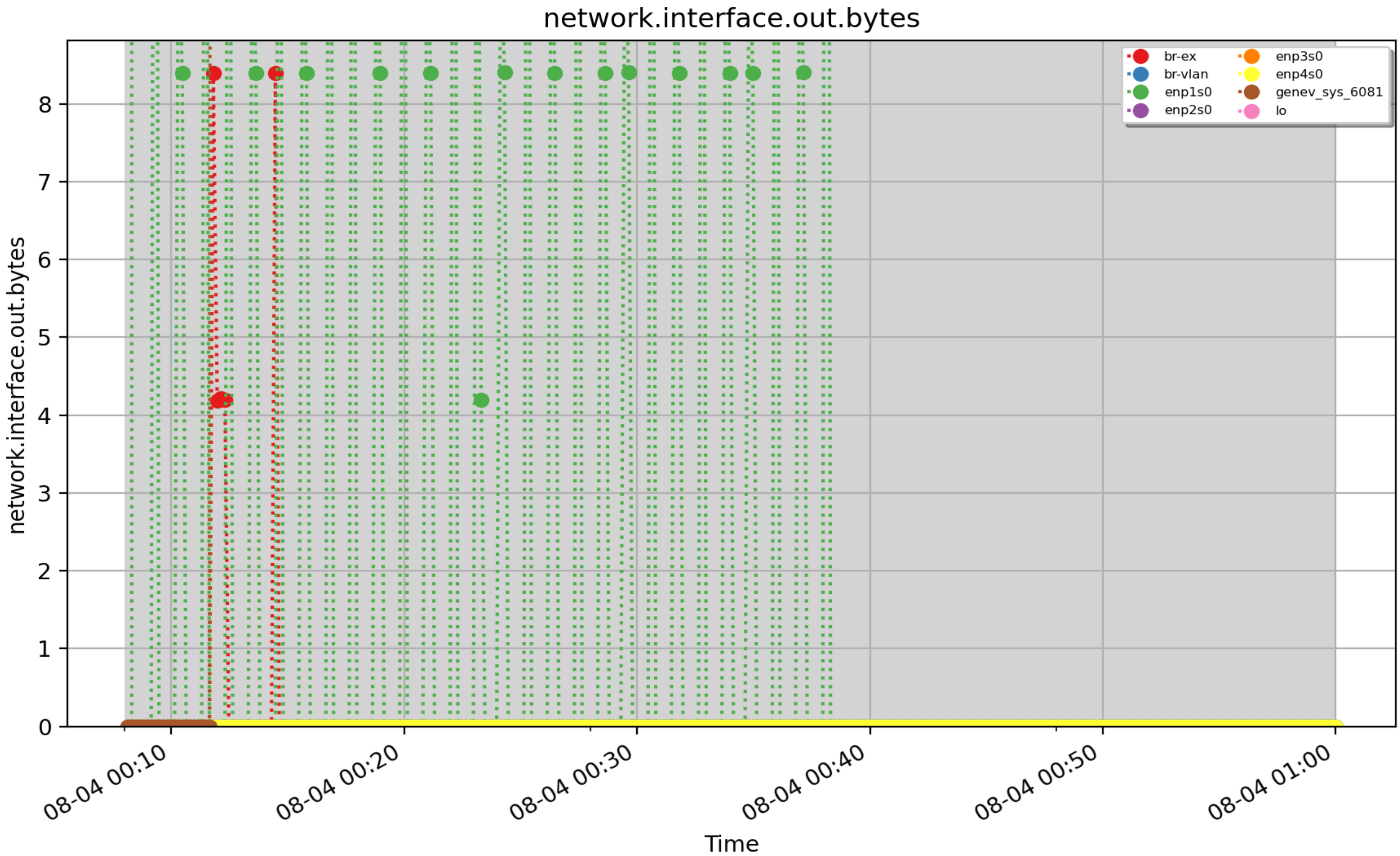
network.interface.in.drops: drop column on the "Receive" side of /proc/net/dev (stats->{rx_dropped + rx_missed_errors} counters in rtnl_link_stats64) rx_dropped are the dropped packets due to no space in linux buffers and rx_missed are due to the receiver NIC missing a packet. Not all NICS use the rx_missed_errors counter. (count - U64) - *rate converted*



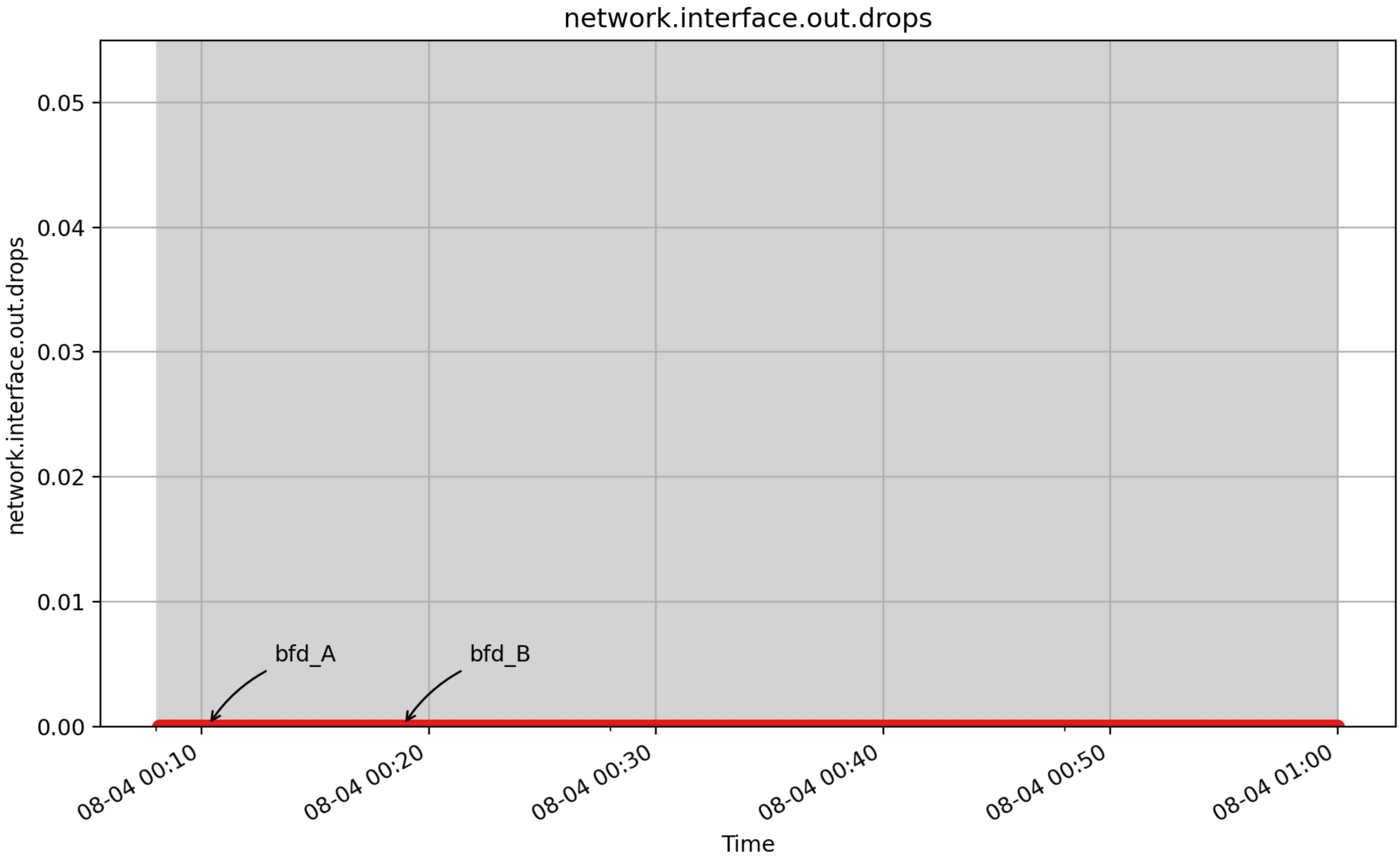
network.interface.in.packets: packets column on the "Receive" side of /proc/net/dev (stats->rx_packets counter in rtnl_link_stats64) (count - U64) - *rate converted*



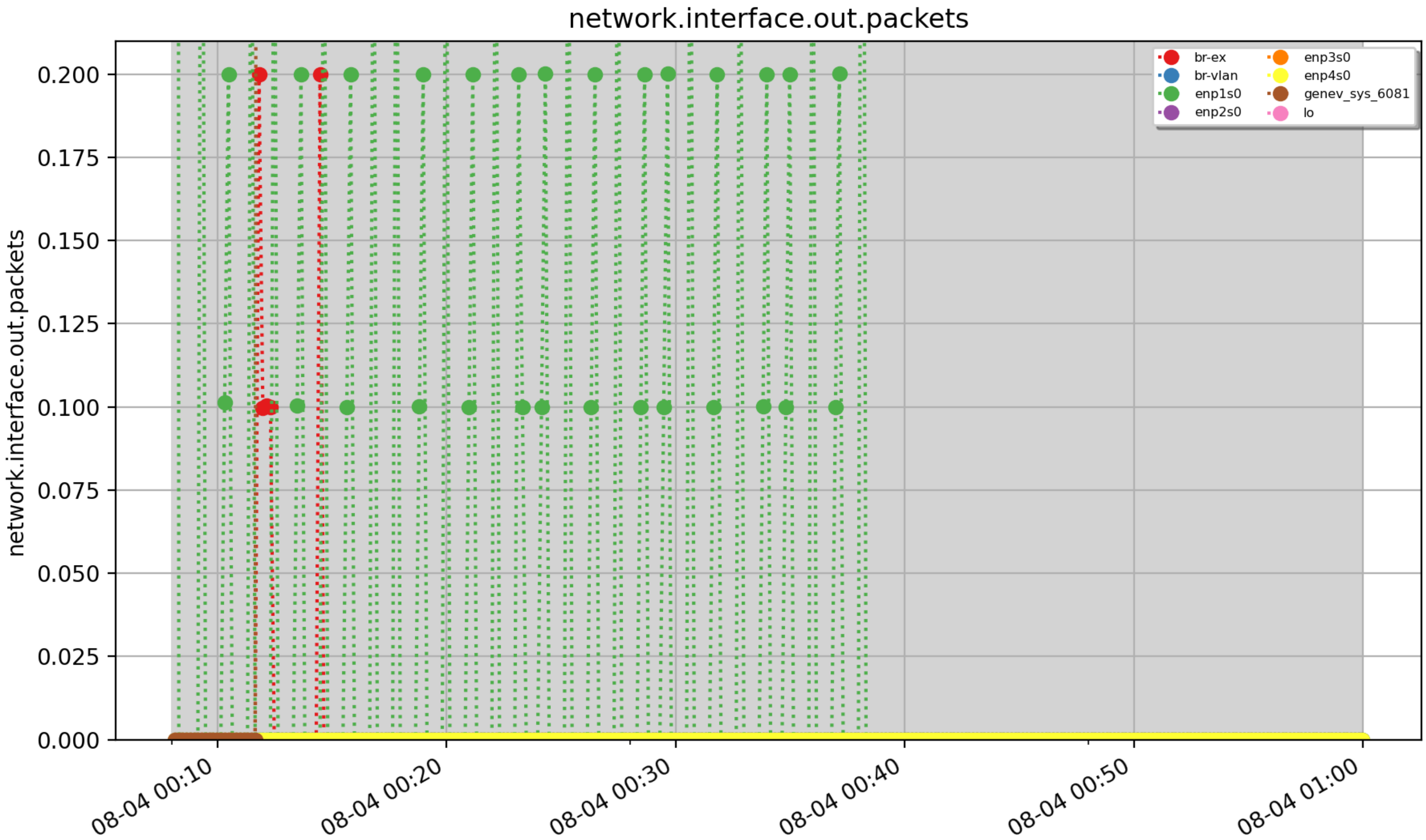
network.interface.mtu: maximum transmission unit on network interface (byte - U32)



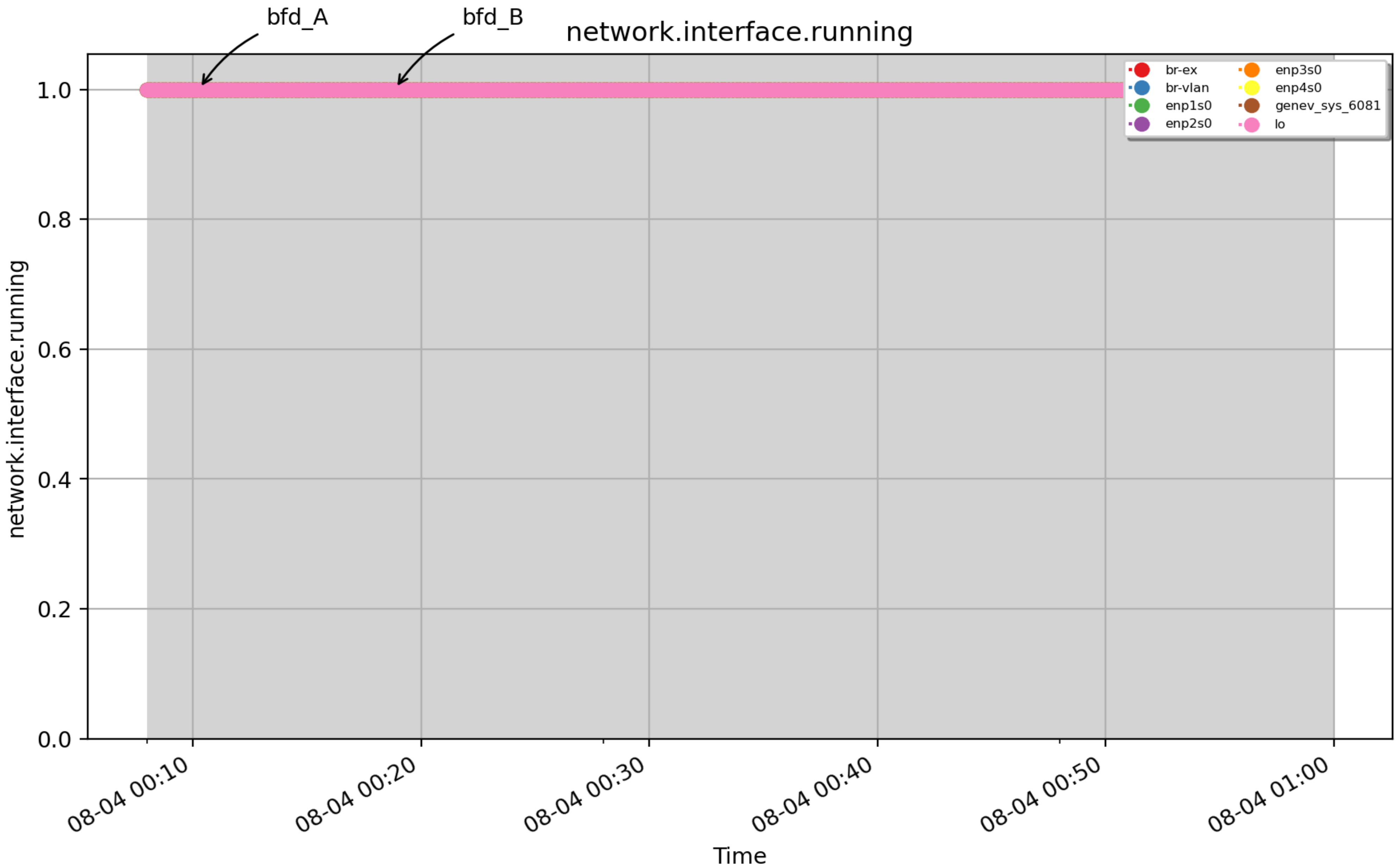
network.interface.out.bytes: bytes column on the "Transmit" side of /proc/net/dev (stats->tx_bytes counter in rtnl_link_stats64) (byte - U64) - *rate converted*



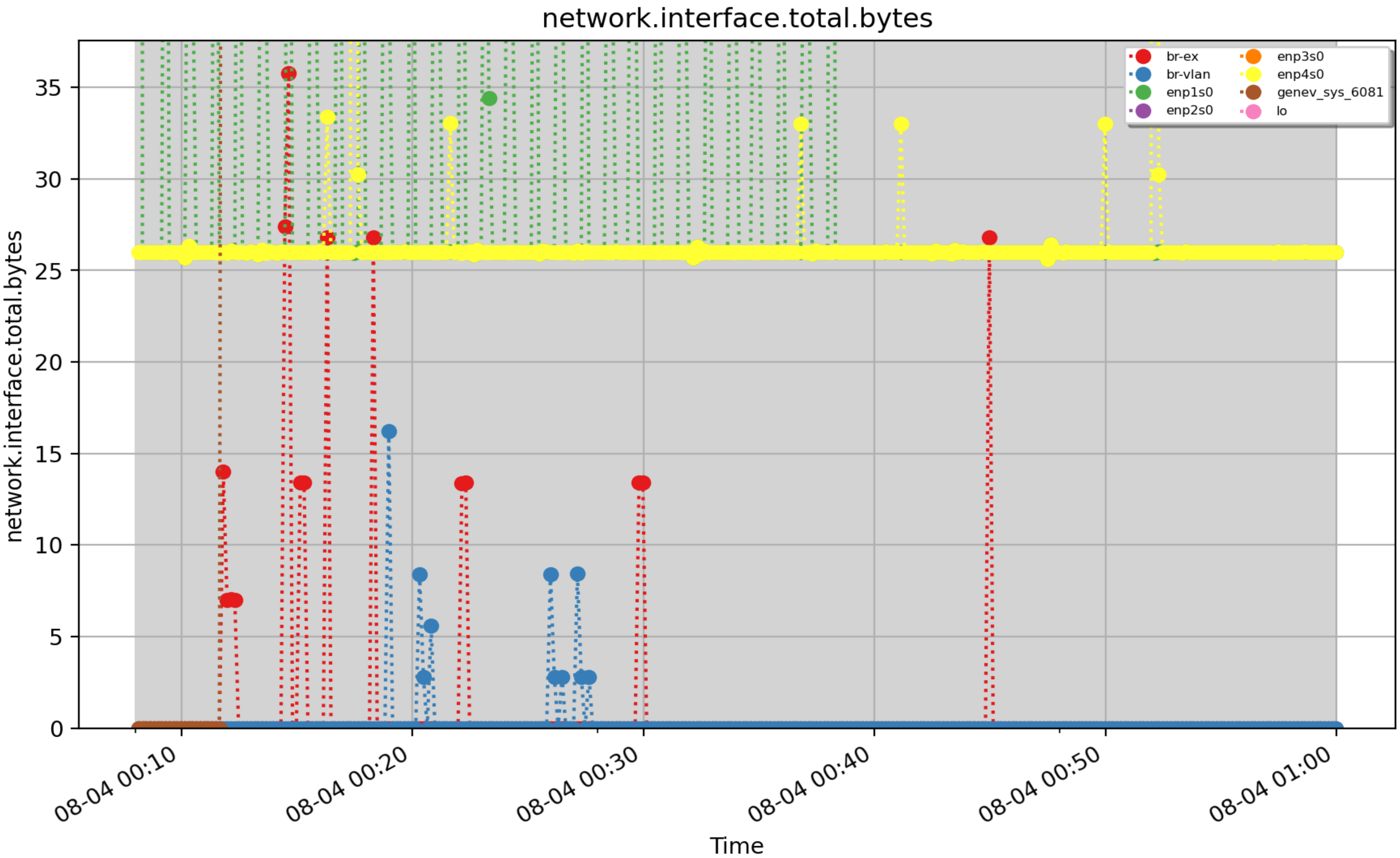
network.interface.out.drops: drop column on the "Transmit" side of /proc/net/dev (stats->tx_dropped counter in rtnl_link_stats64) (count - U64) - *rate converted*



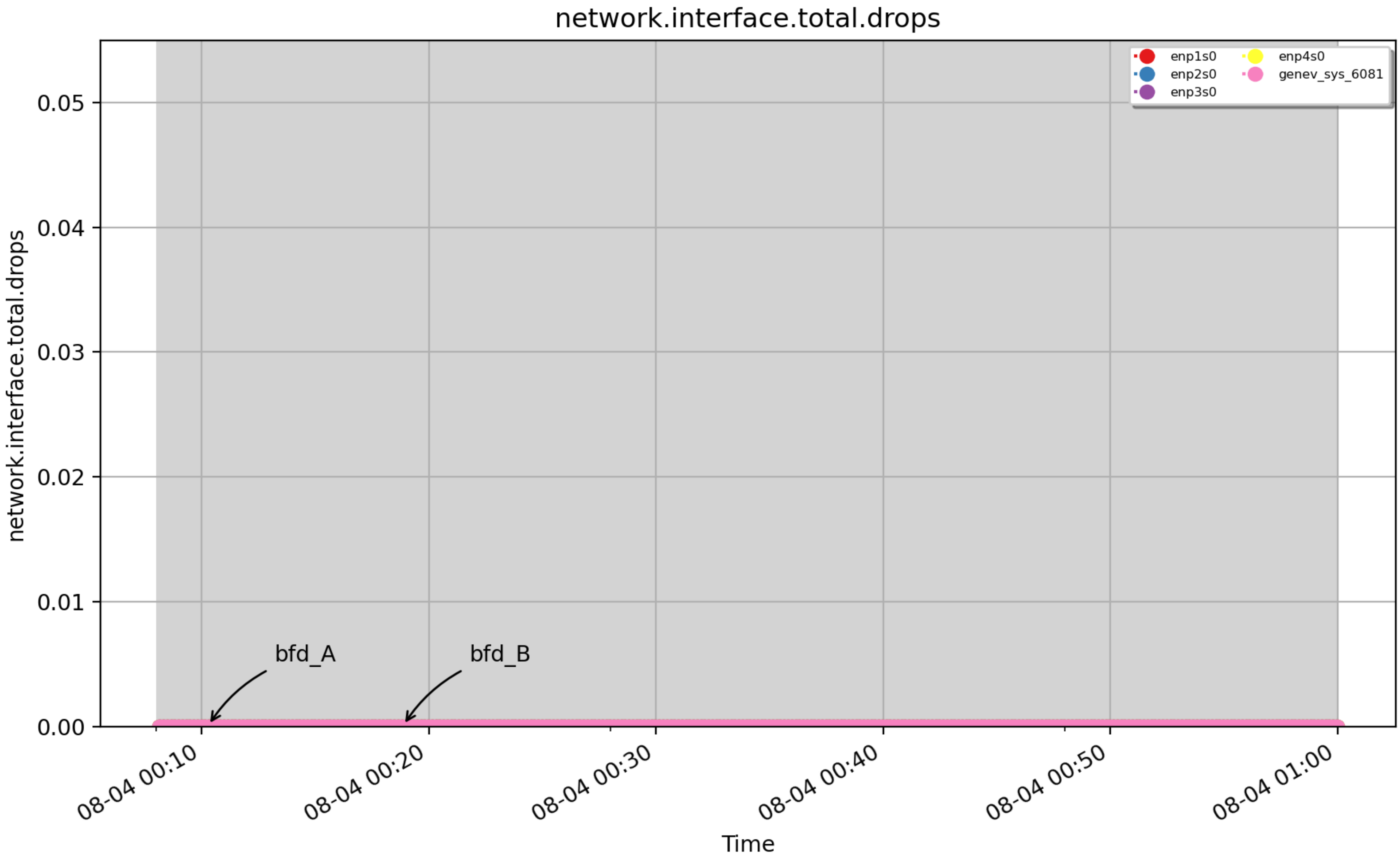
network.interface.out.packets: packets column on the "Transmit" side of /proc/net/dev (stats->tx_packets counter in rtnl_link_stats64) (count - U64) - *rate converted*



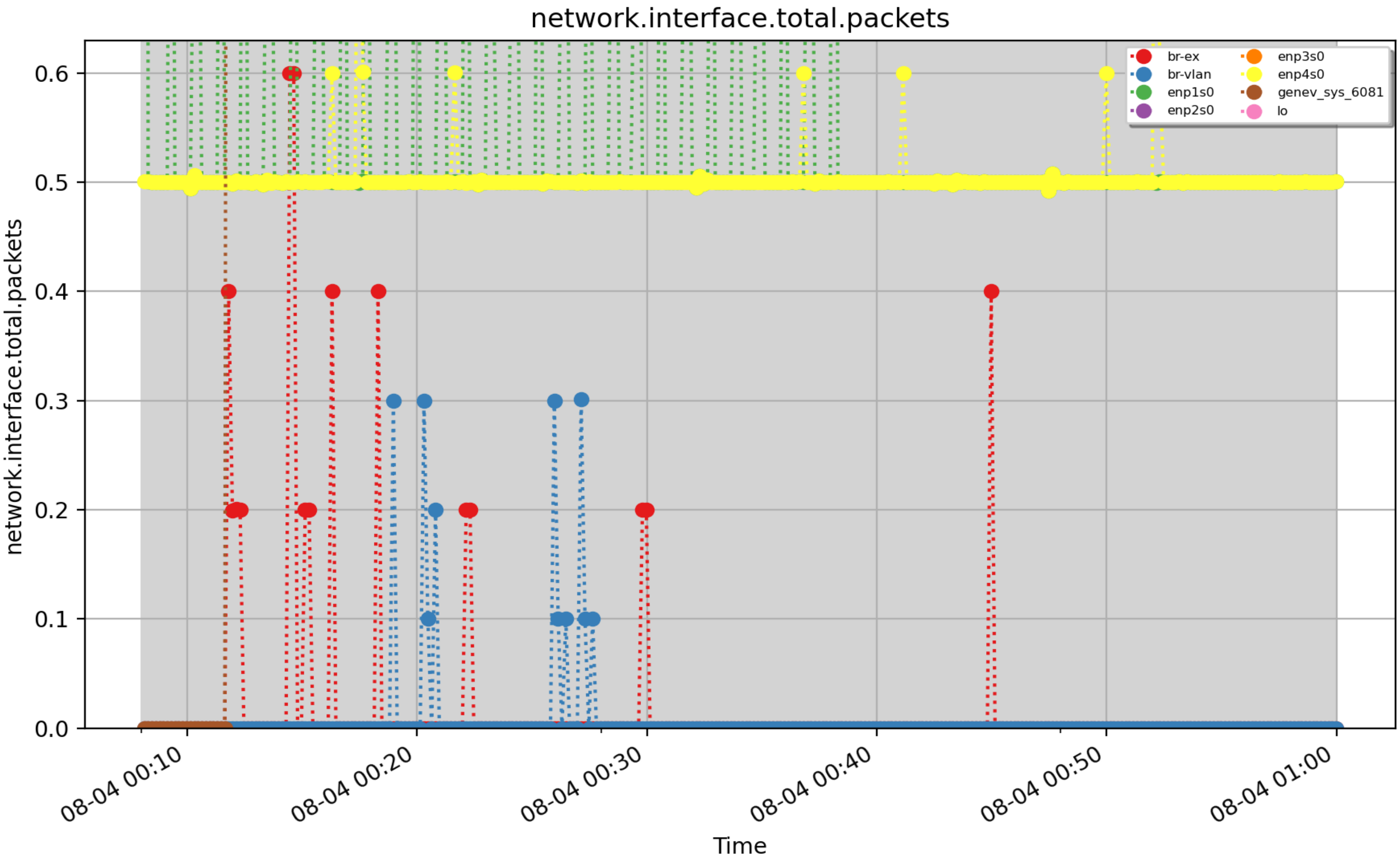
network.interface.running: boolean for whether interface has resources allocated (- U32)



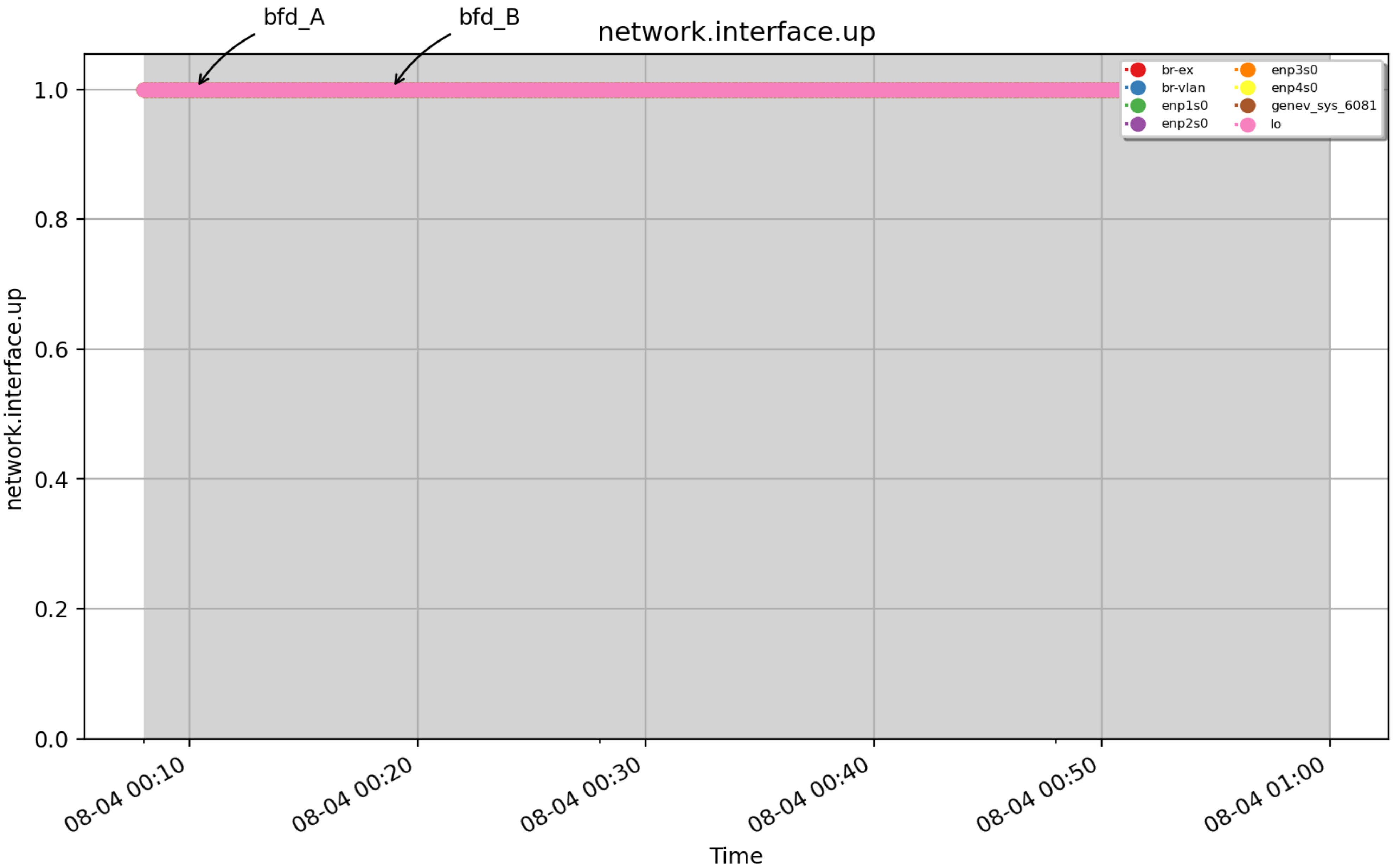
network.interface.total.bytes: network total (in+out) bytes from /proc/net/dev per network interface (byte - U64) - *rate converted*



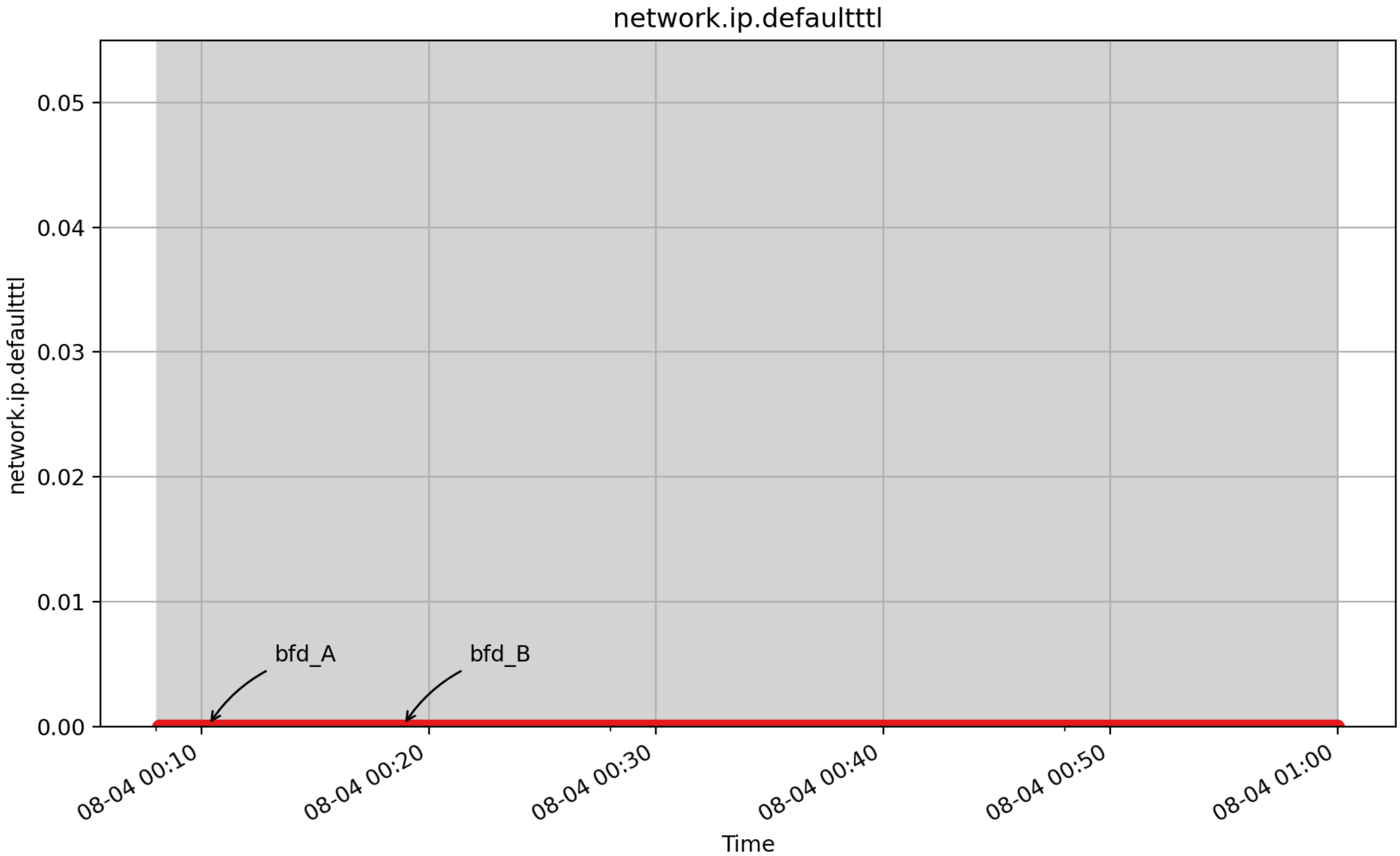
network.interface.total.drops: network total (in+out) drops from /proc/net/dev per network interface (count - U64) - *rate converted*



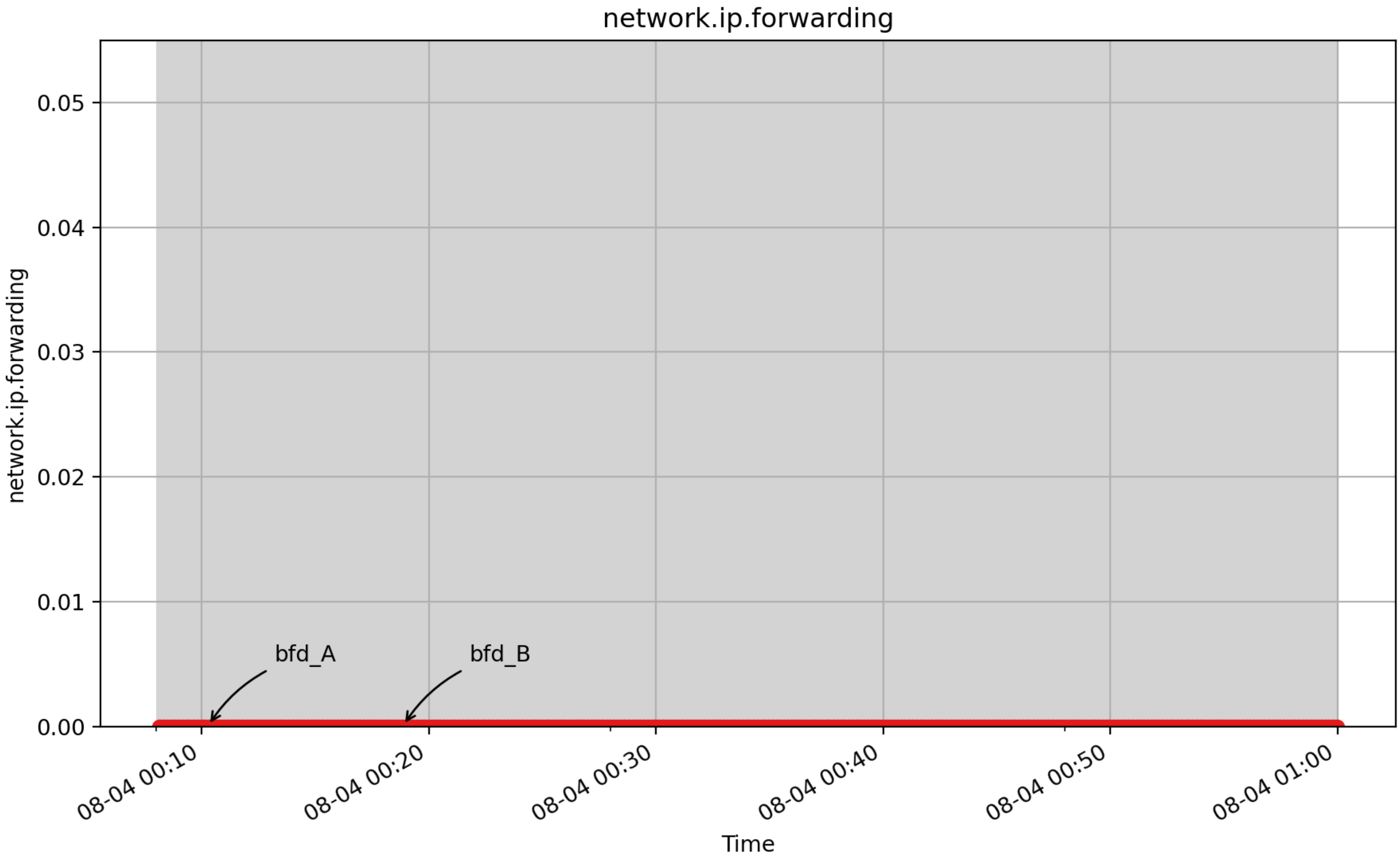
network.interface.total.packets: network total (in+out) packets from /proc/net/dev per network interface (count - U64) - *rate converted*



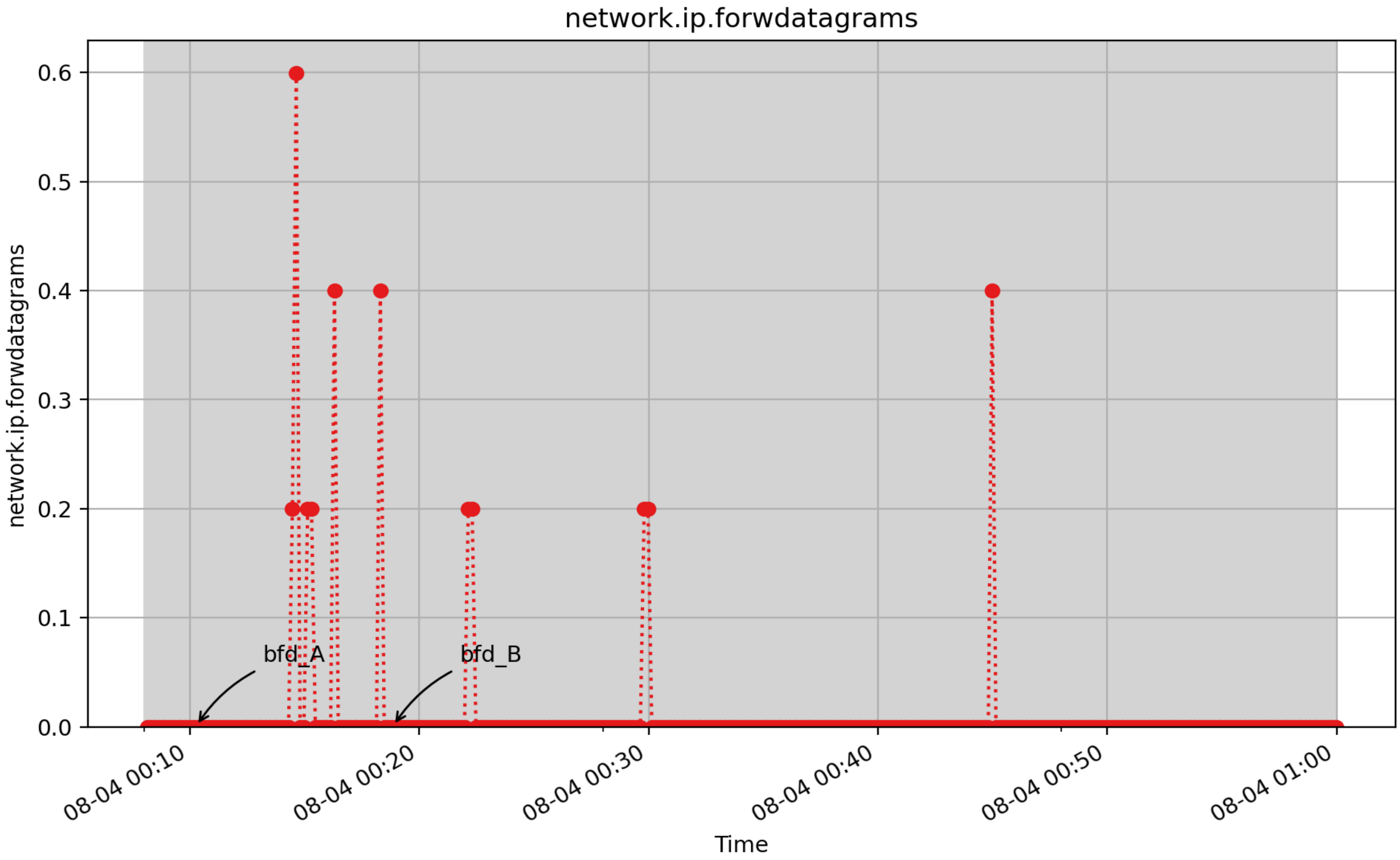
network.interface.up: boolean for whether interface is currently up or down (- U32)



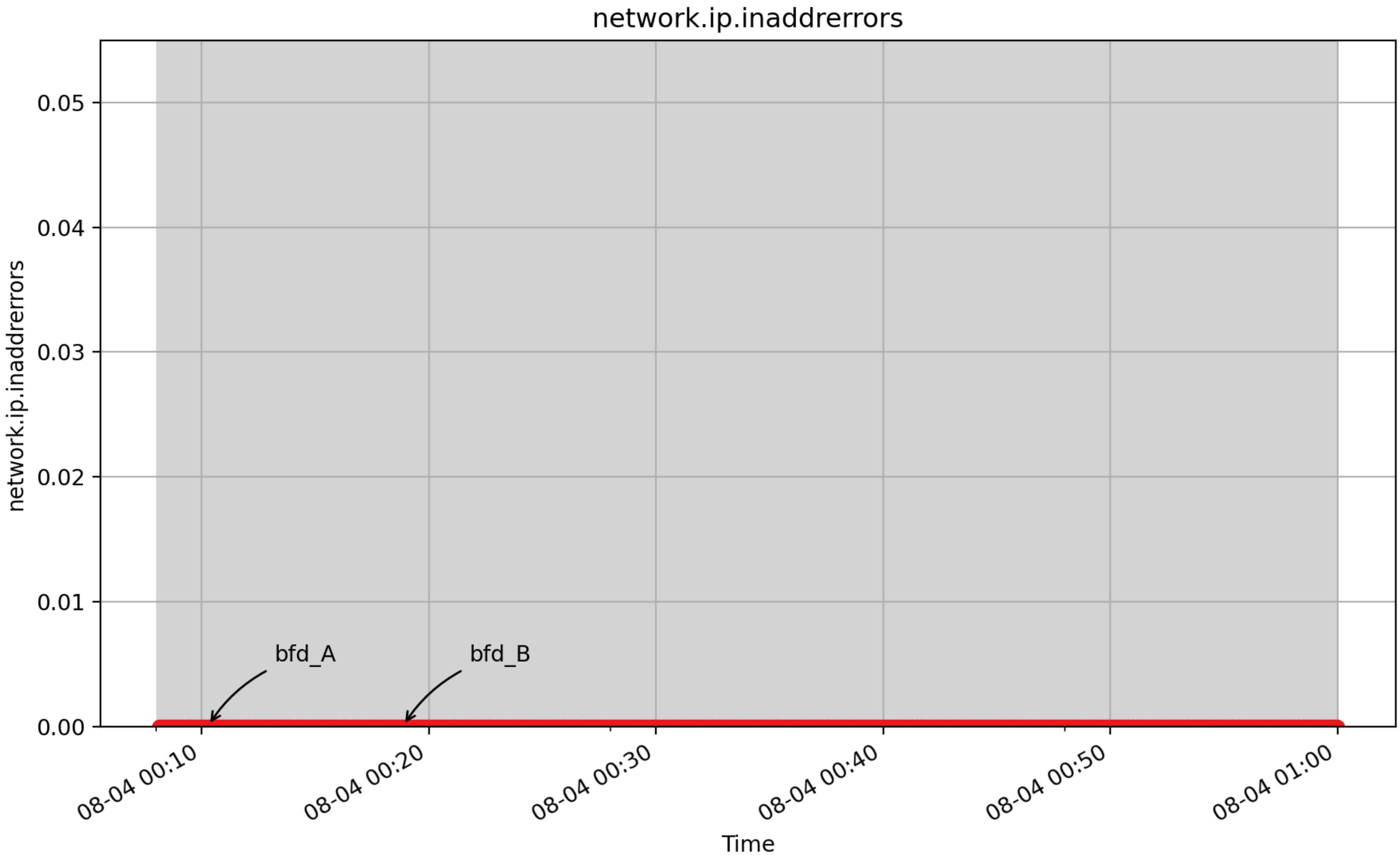
network.ip.defaultttl: count of ip defaultttl (count - U64) - *rate converted*



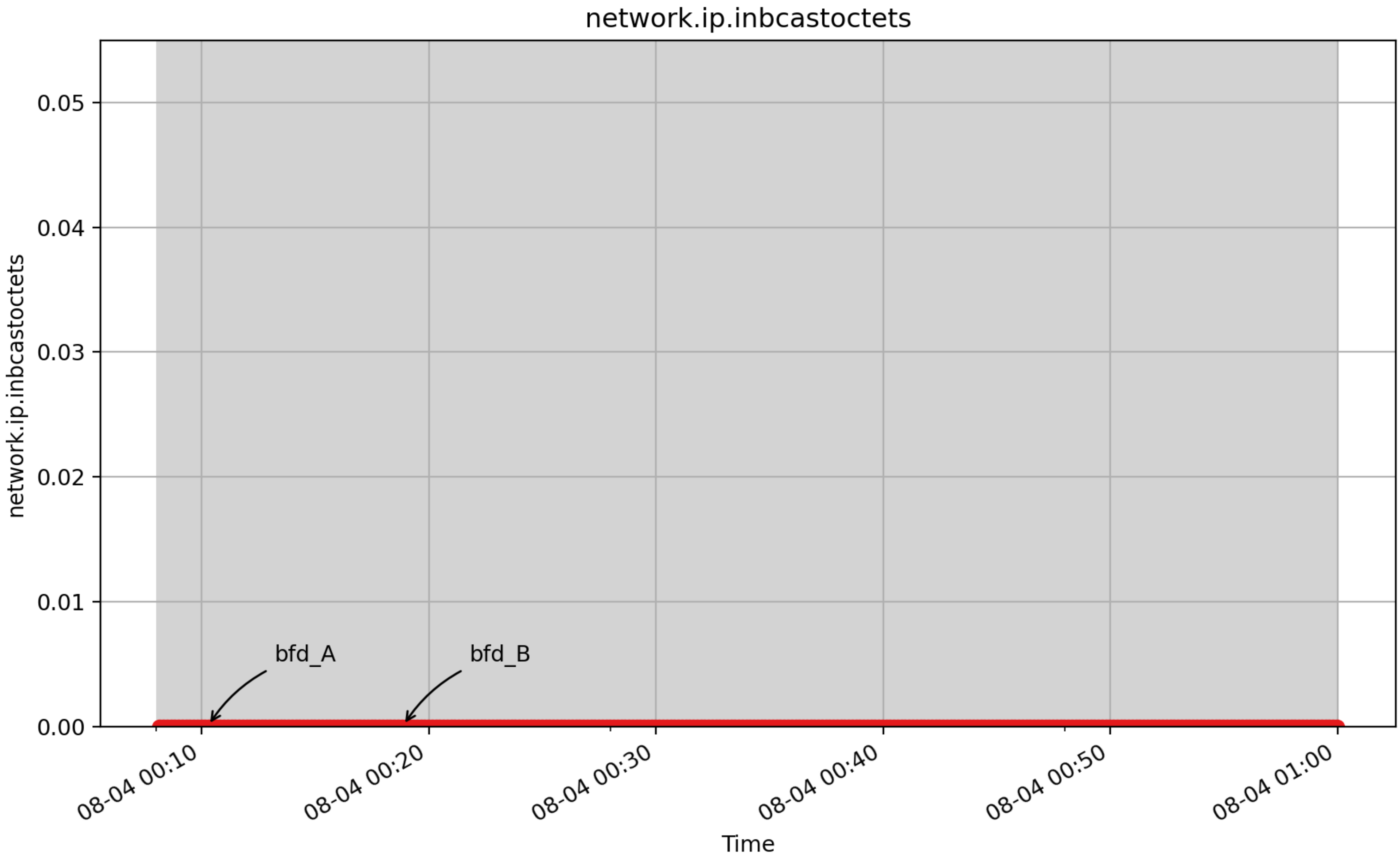
network.ip.forwarding: count of ip forwarding (count - U64) - *rate converted*



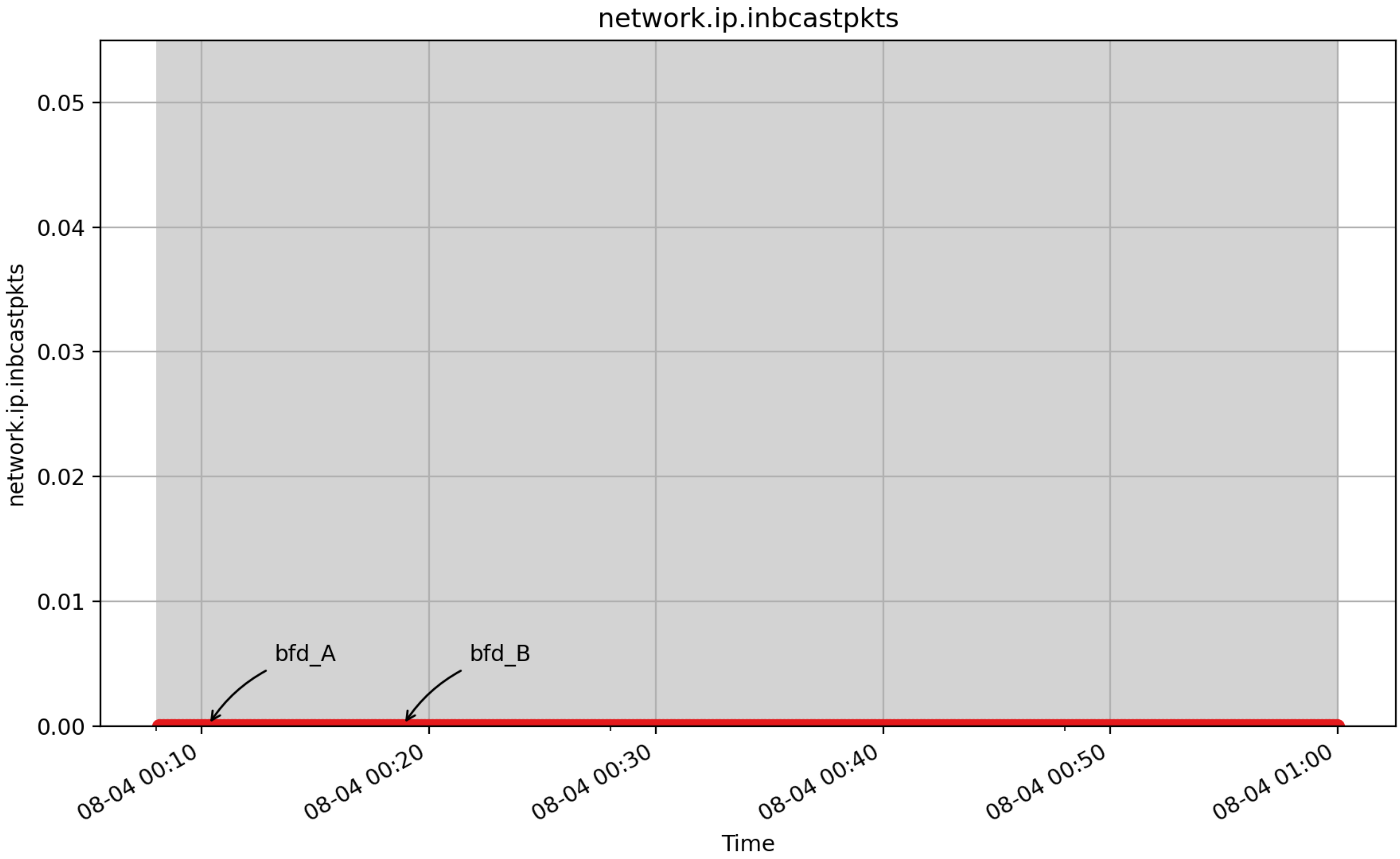
network.ip.forwdatagrams: count of ip forwdatagrams (count - U64) - *rate converted*



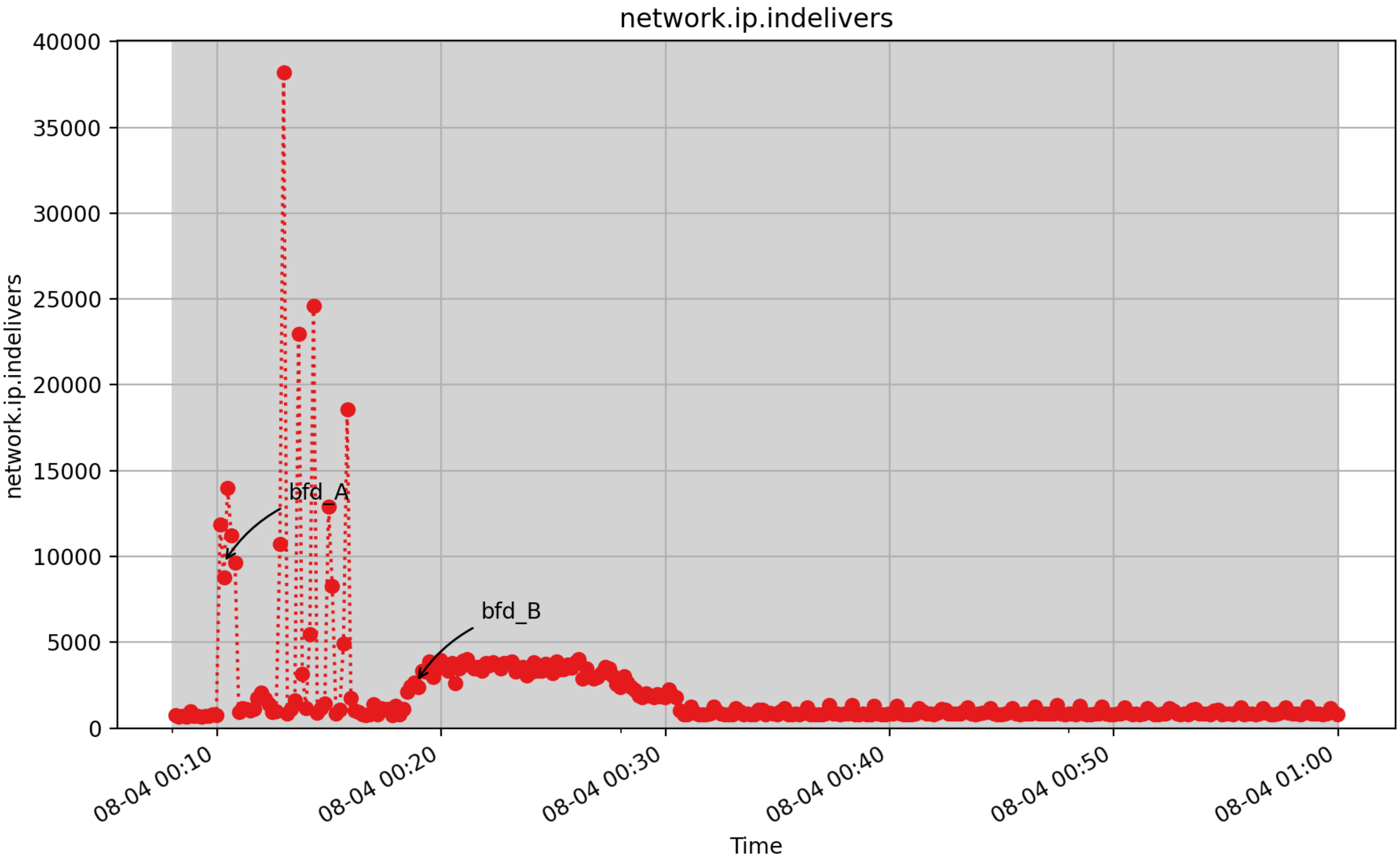
network.ip.inaddrerrors: count of ip inaddrerrors (count - U64) - *rate converted*



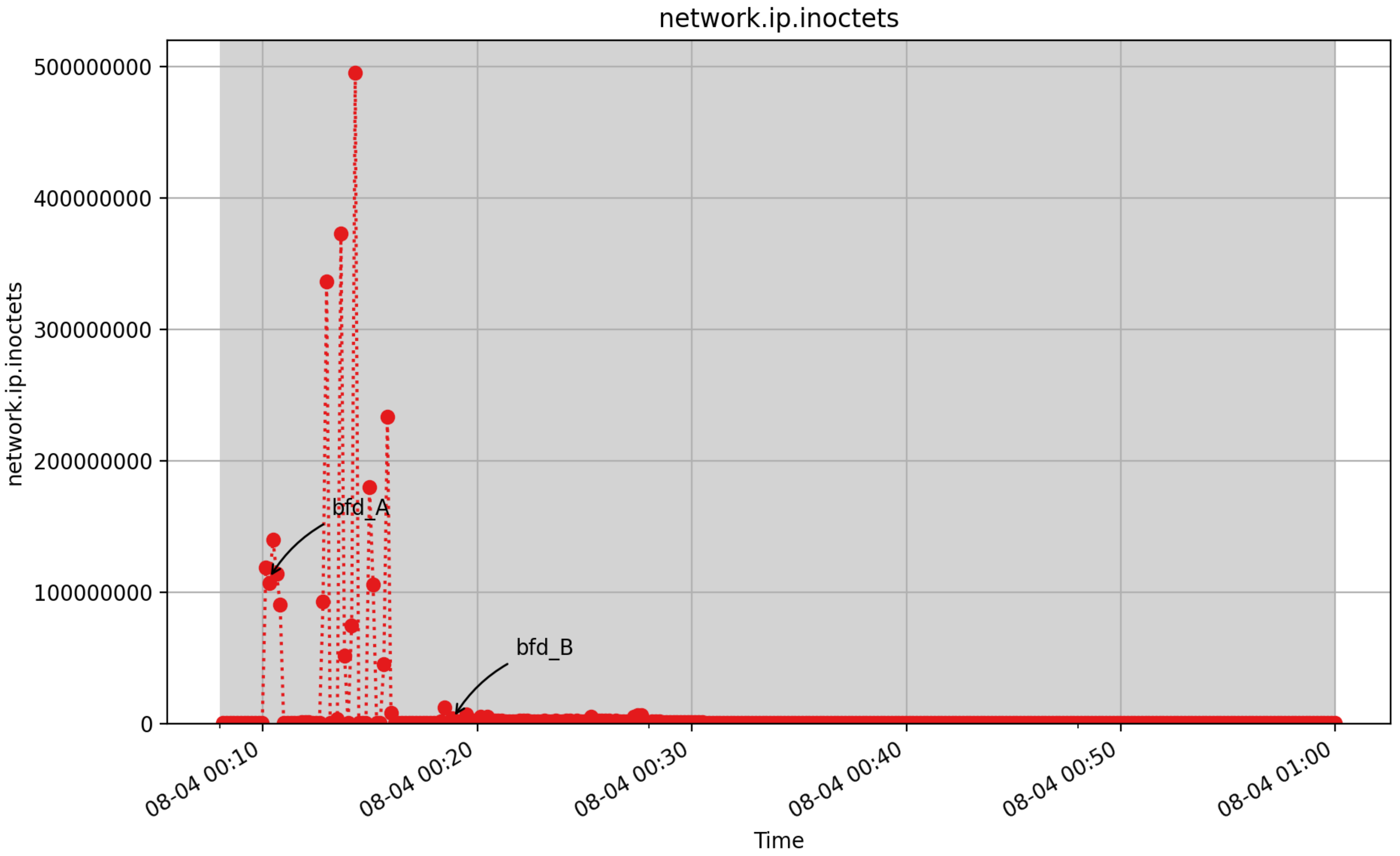
network.ip.inbcastoctets: Number of received IP broadcast octets (count - U64) - *rate converted*



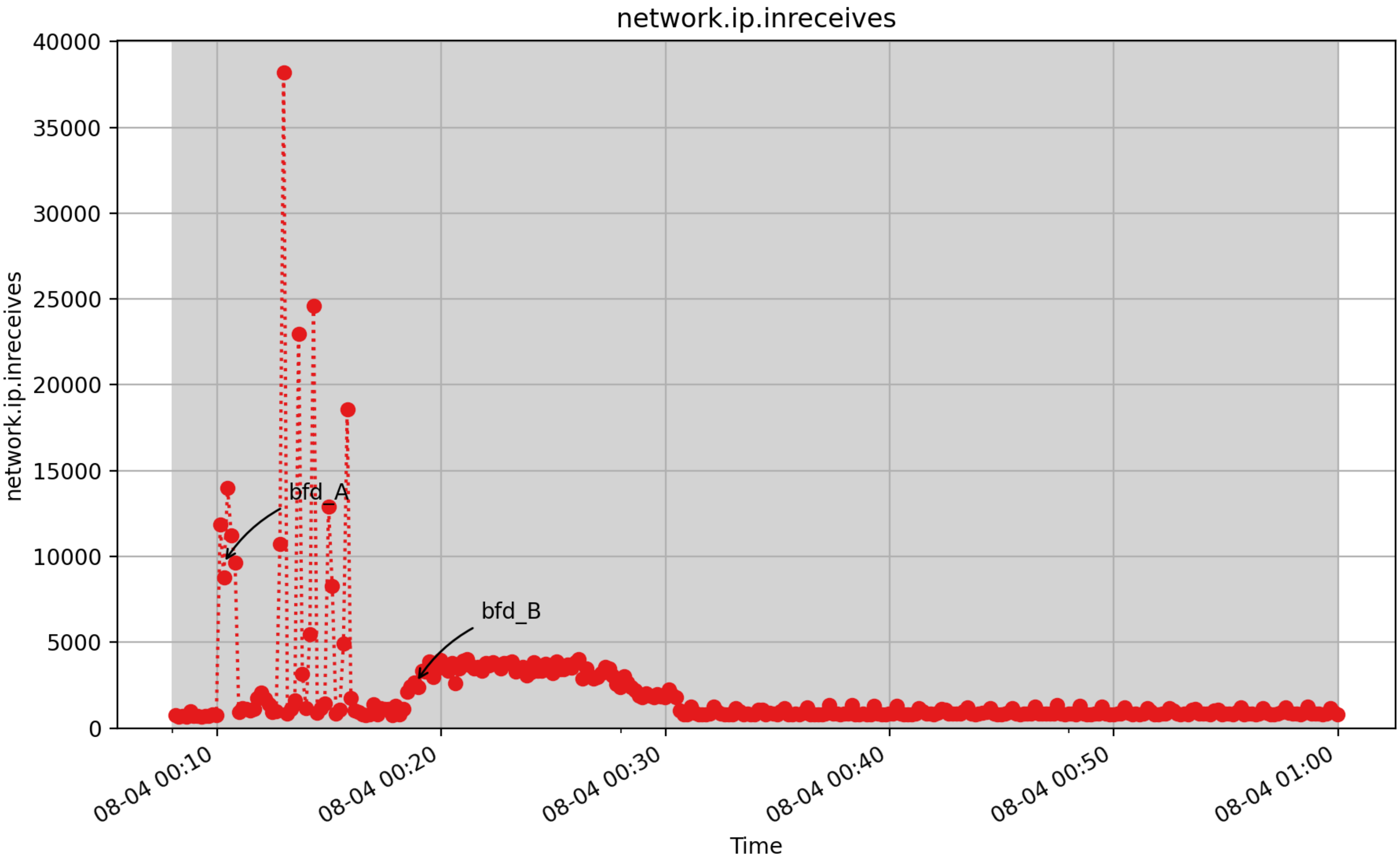
network.ip.inbcastpkts: Number of received IP broadcast datagrams (count - U64) - *rate converted*



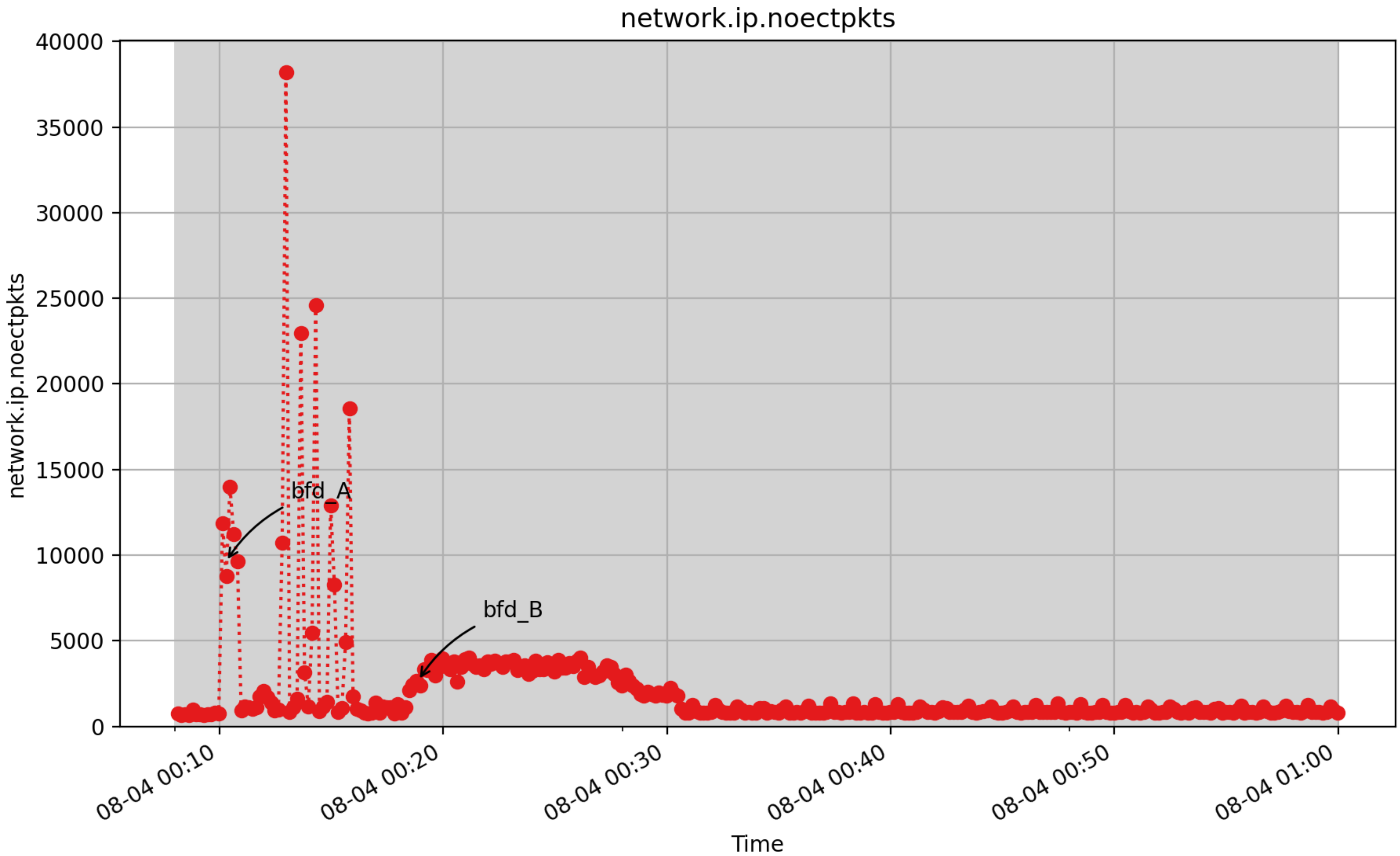
network.ip.indelivers: count of ip indelivers (count - U64) - *rate converted*



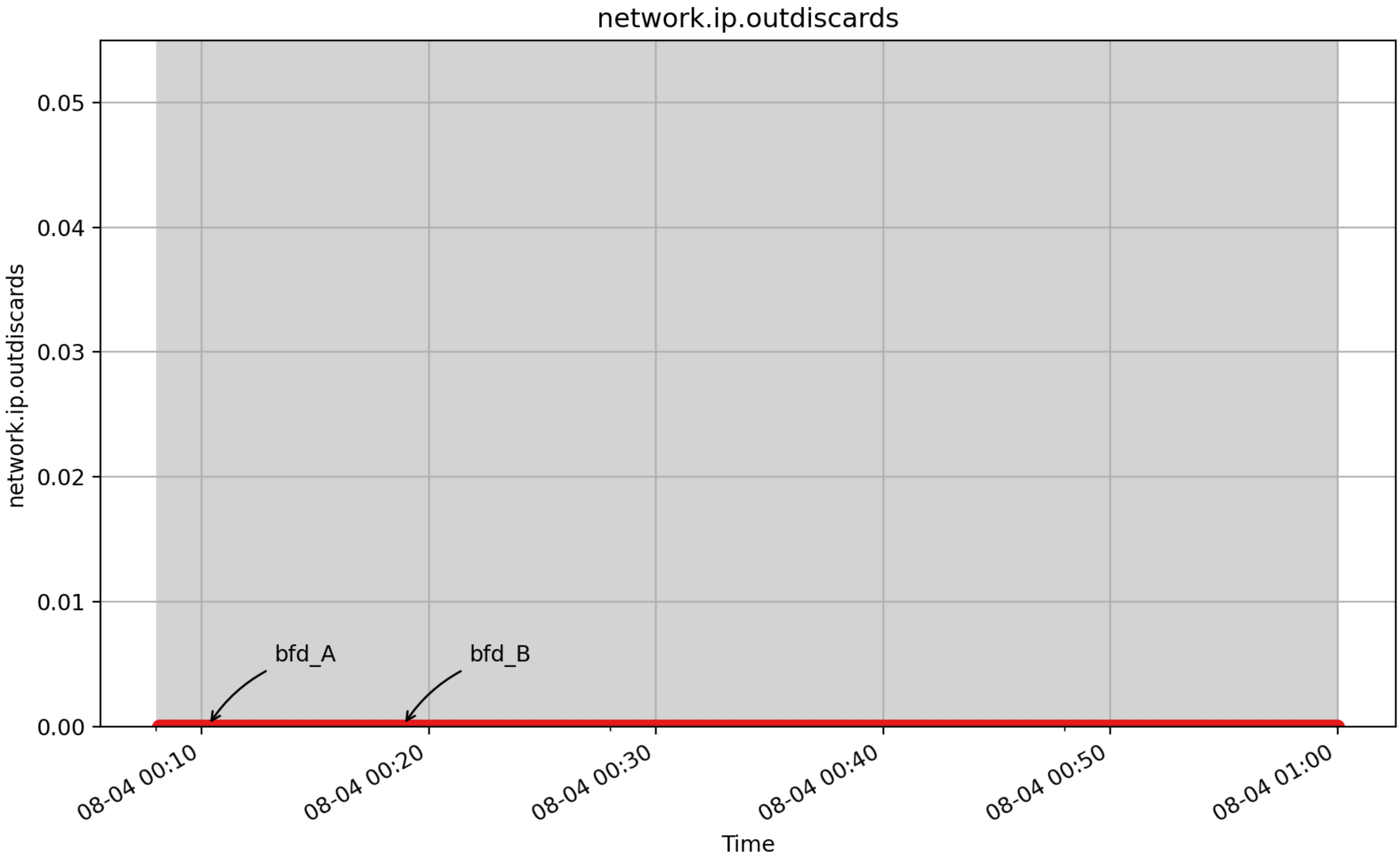
network.ip.inoctets: Number of received octets (count - U64) - *rate converted*



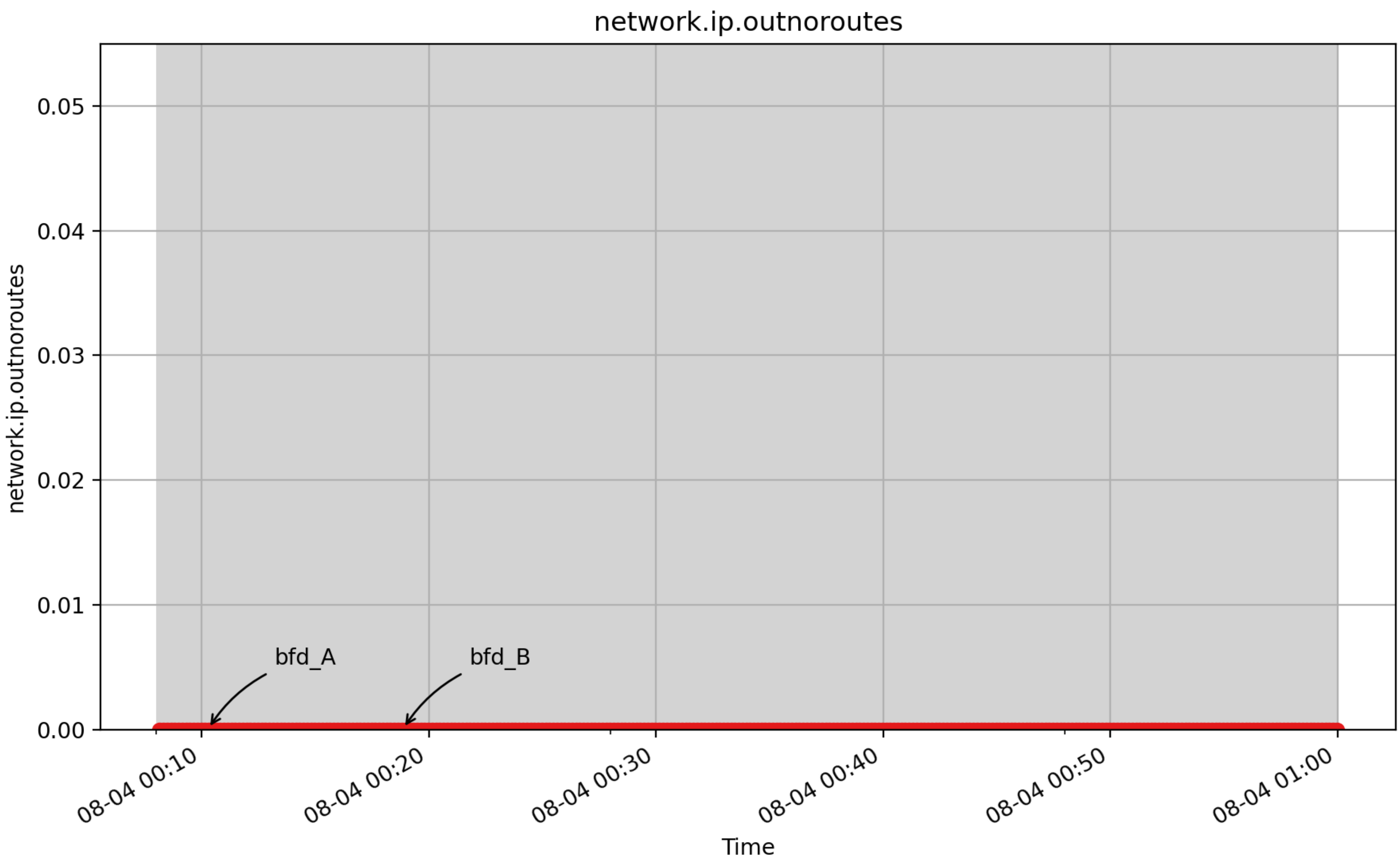
network.ip.inreceives: count of ip inreceives (count - U64) - *rate converted*



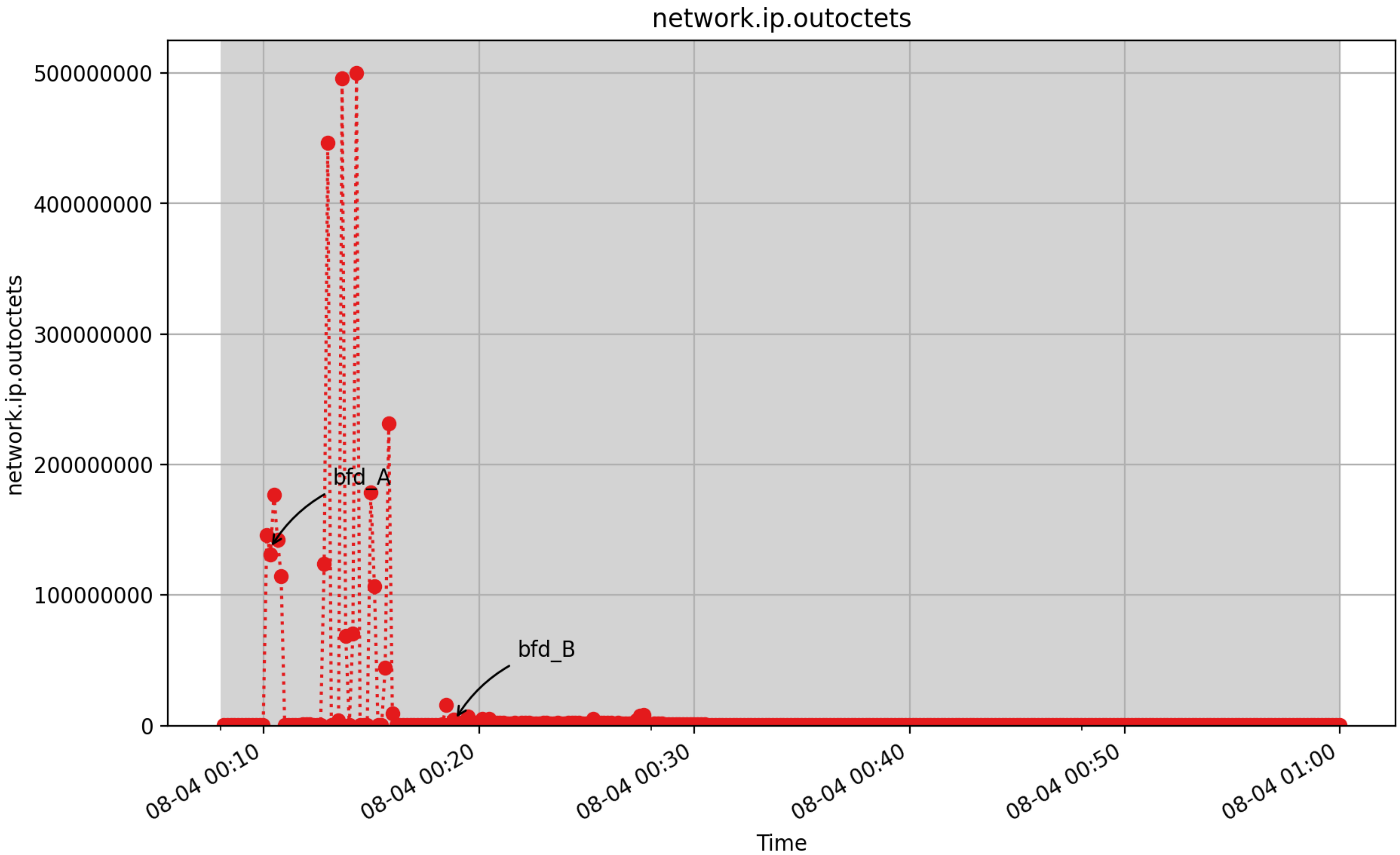
network.ip.noectpkts: Number of packets received with NOECT (count - U64) - *rate converted*



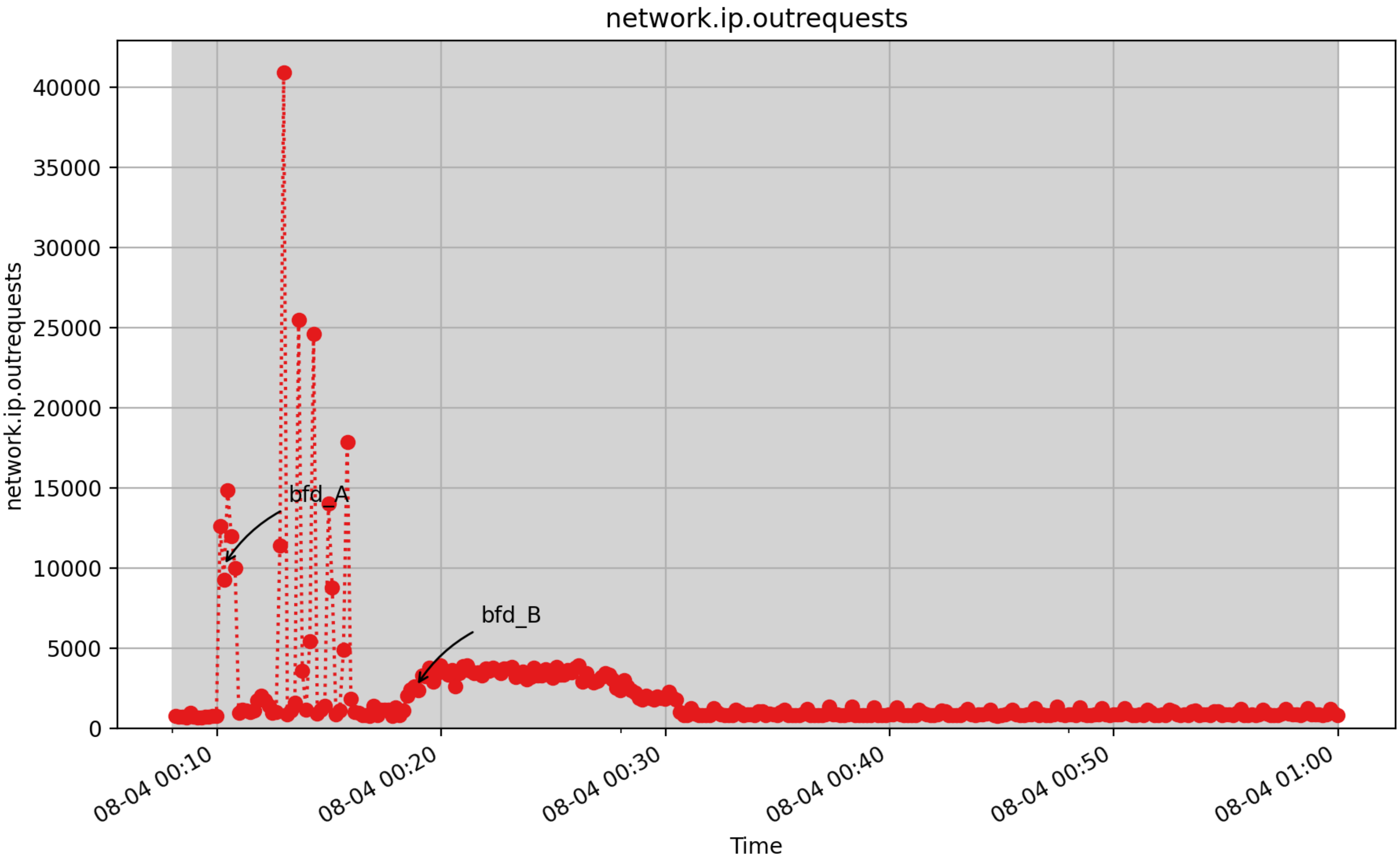
network.ip.outdiscards: count of ip outdiscards (count - U64) - *rate converted*



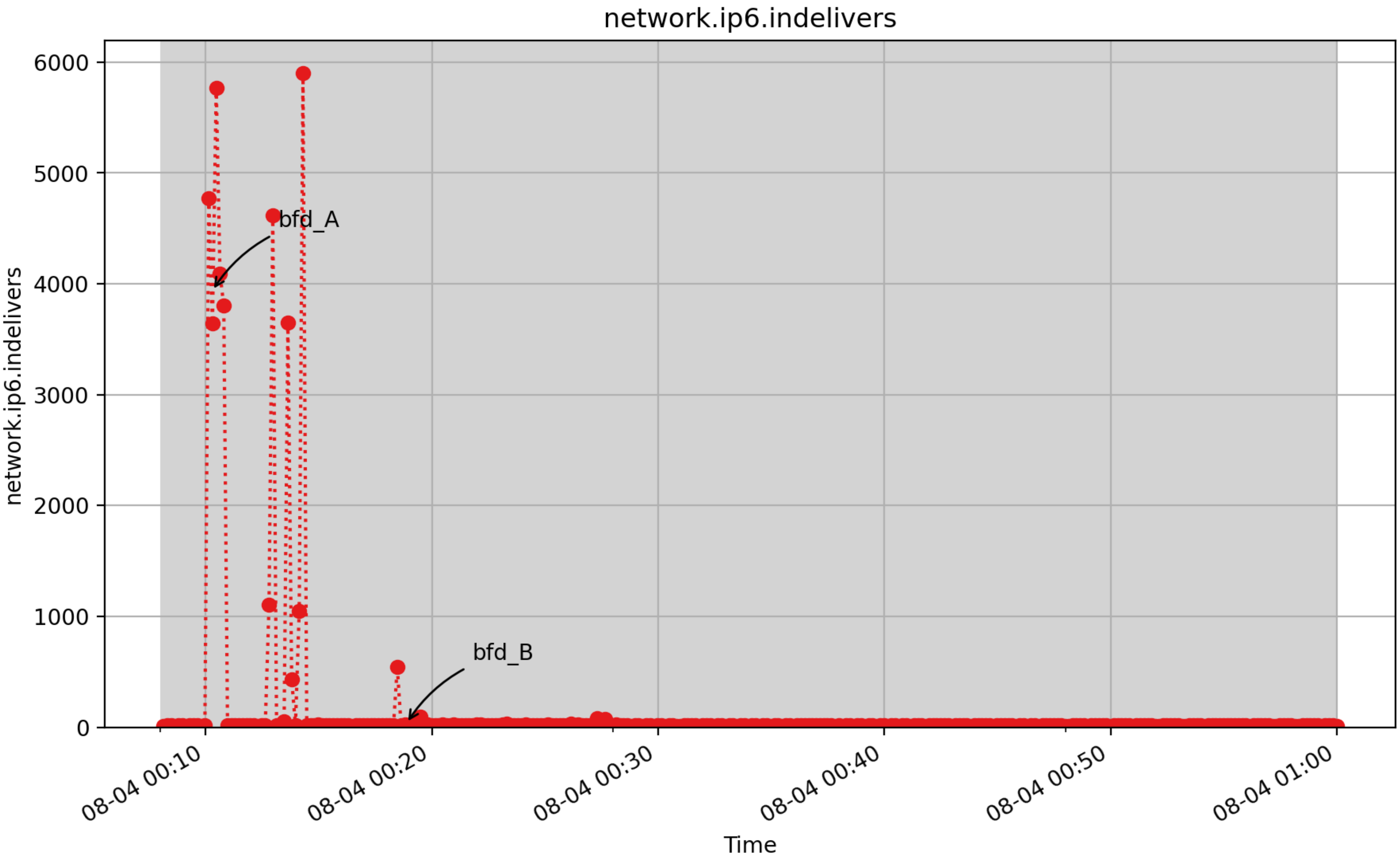
network.ip.outnoroutes: count of ip outnoroutes (count - U64) - *rate converted*



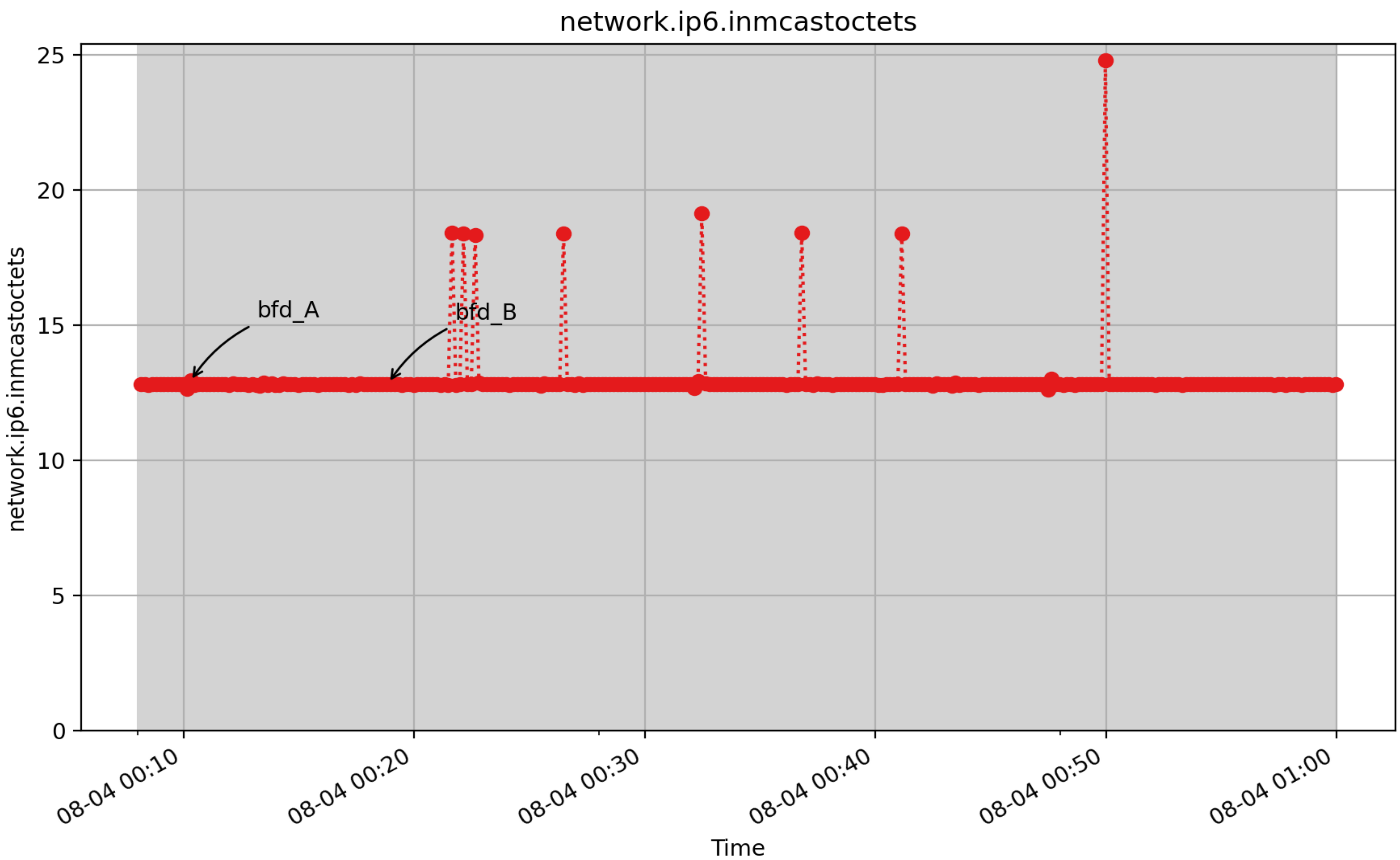
network.ip.outoctets: Number of sent octets (count - U64) - *rate converted*



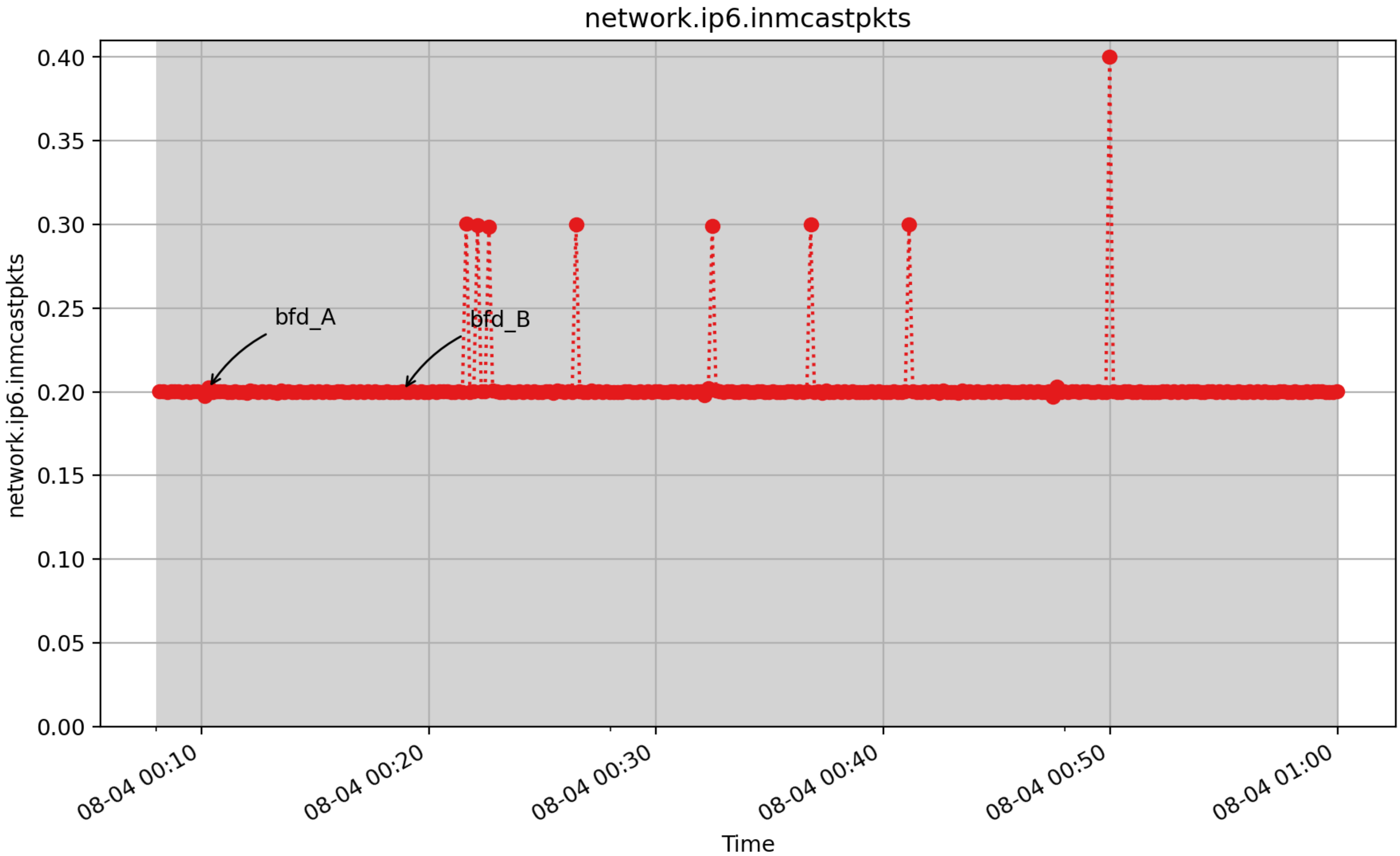
network.ip.outrequests: count of ip outrequests (count - U64) - *rate converted*



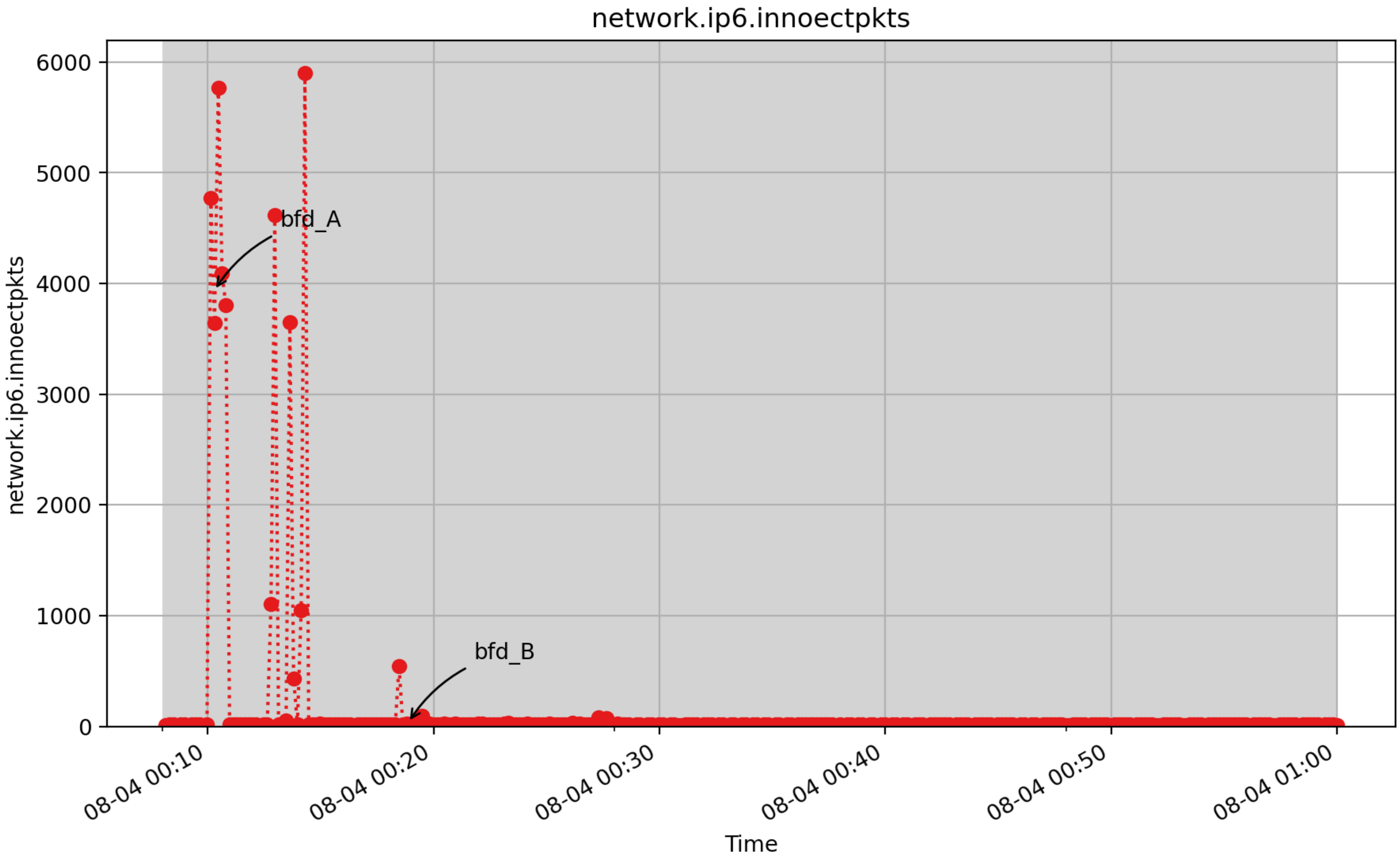
network.ip6.indelivers: count of ip6 indelivers (count - U64) - *rate converted*



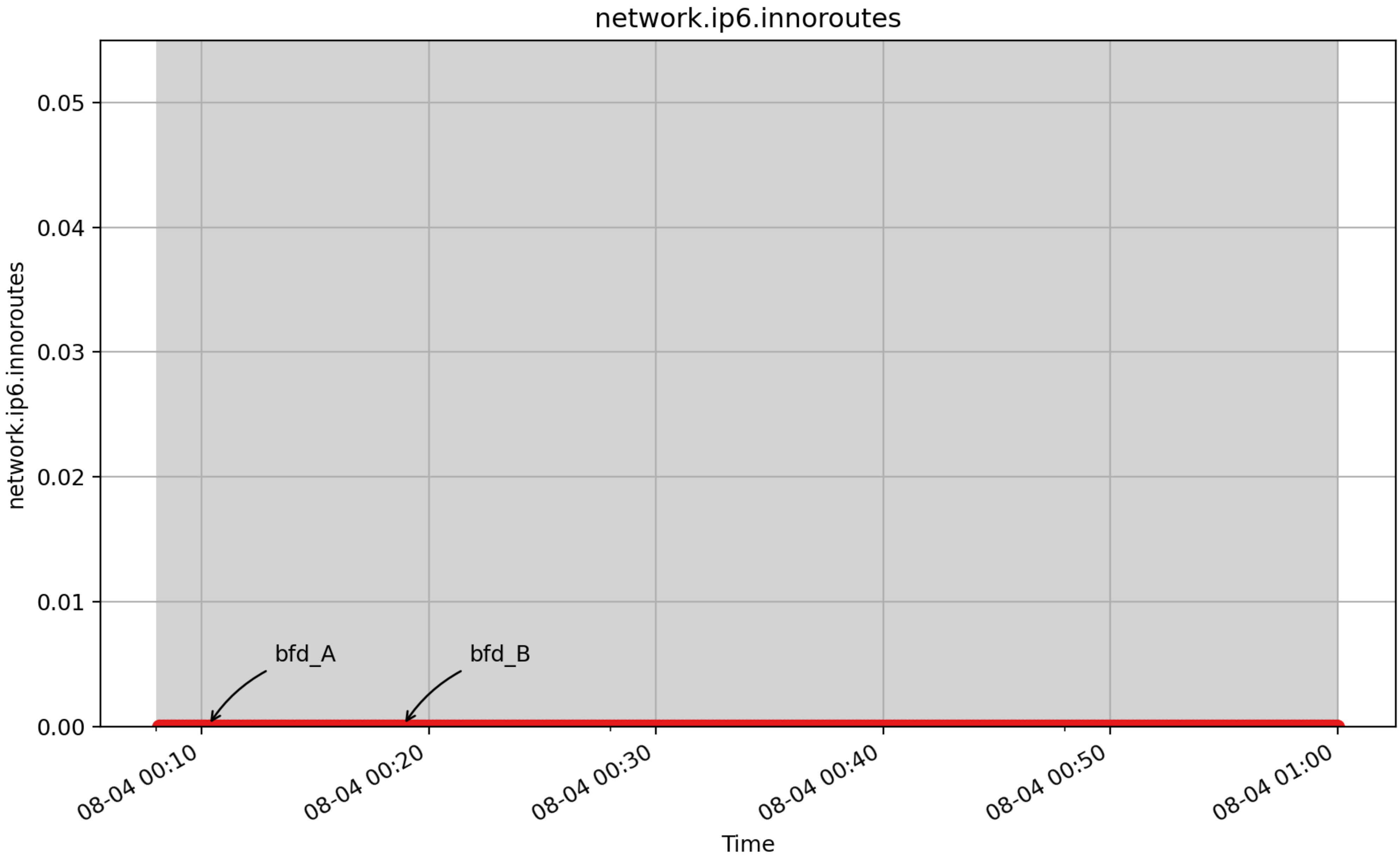
network.ip6.inmcastoctets: count of ip6 multicast octets in (count - U64) - *rate converted*



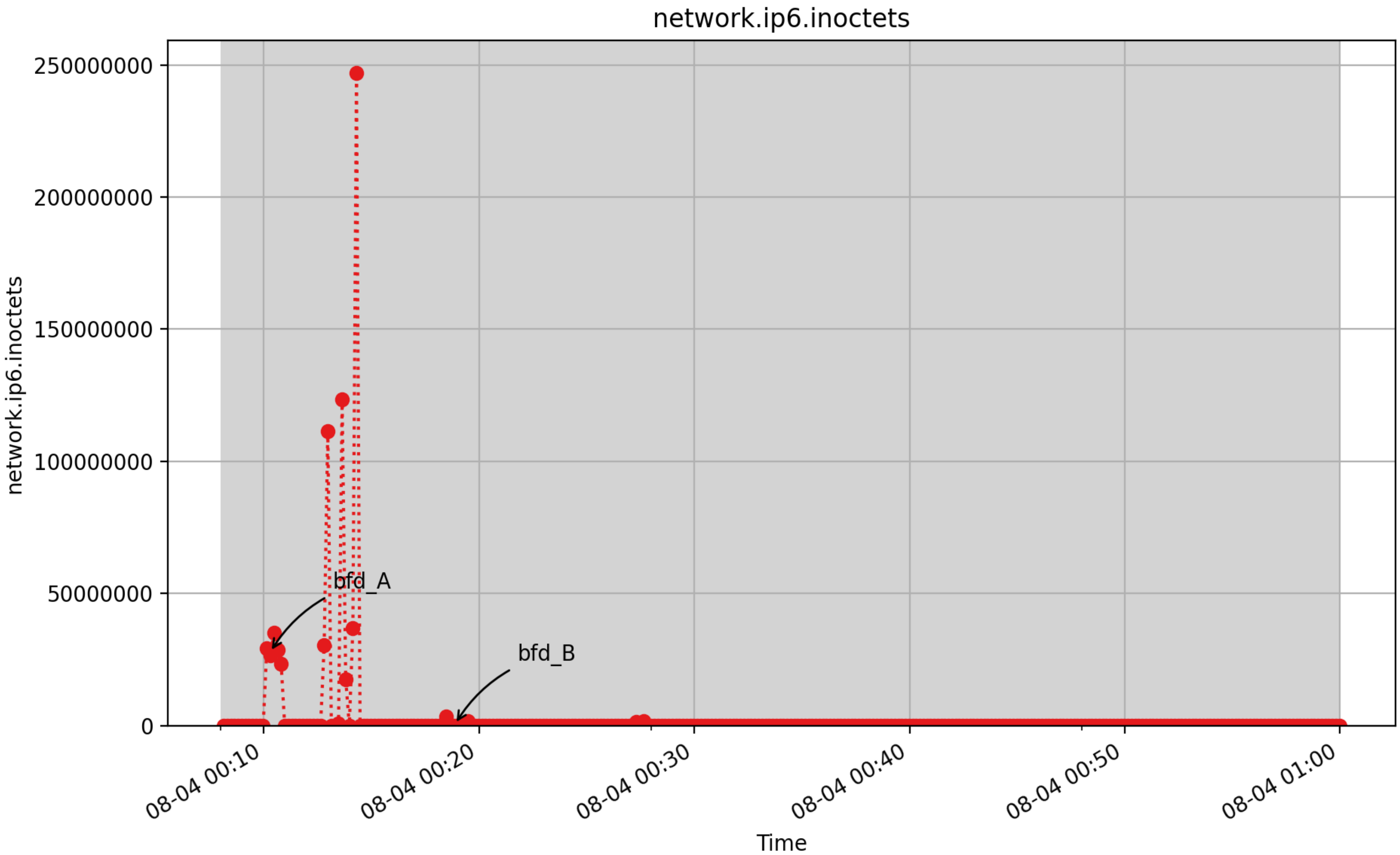
network.ip6.inmcastpkts: count of ip6 multicast packets in (count - U64) - *rate converted*



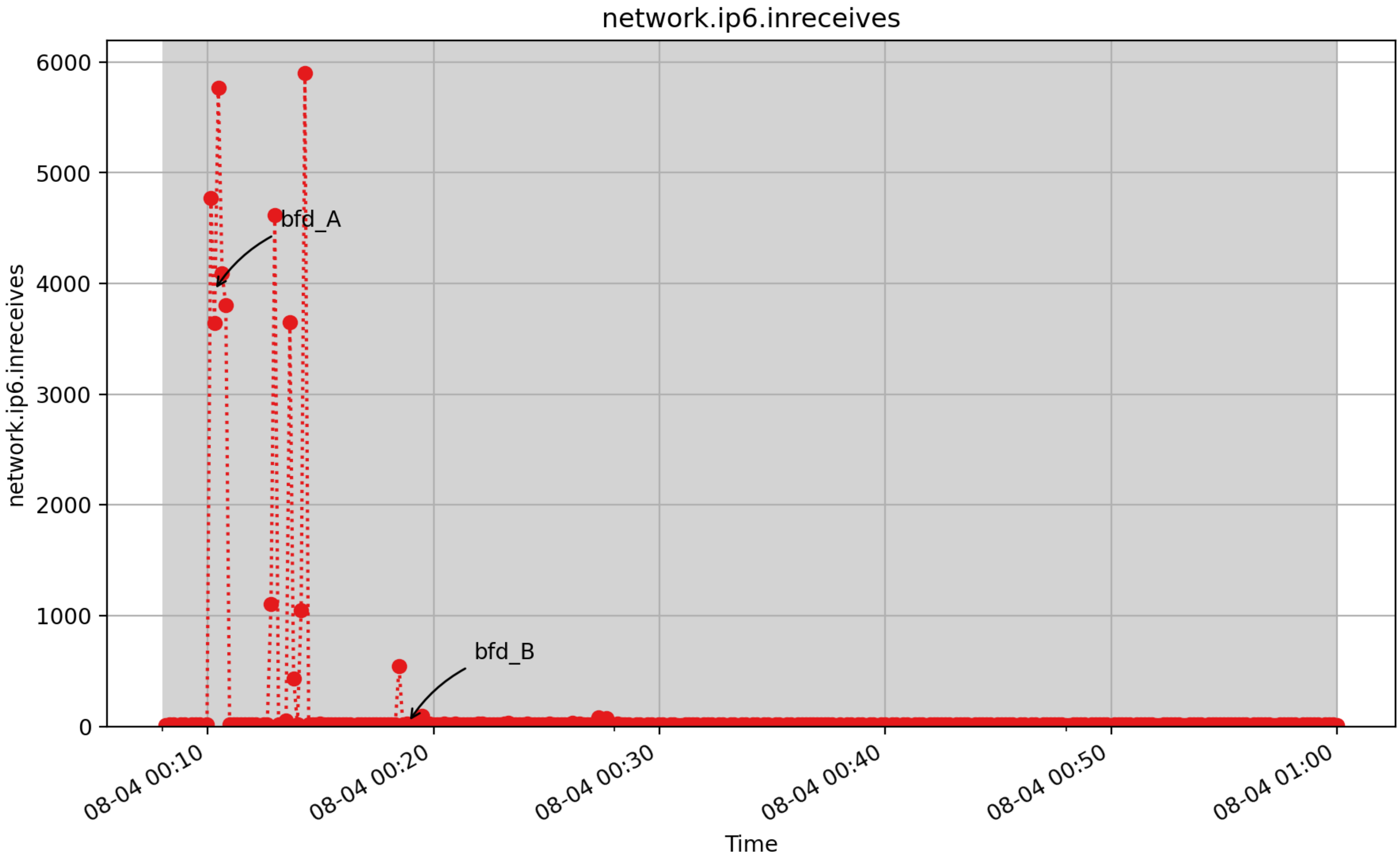
network.ip6.innoectpkts: count of ip6 packets received with NOECT (count - U64) - *rate converted*



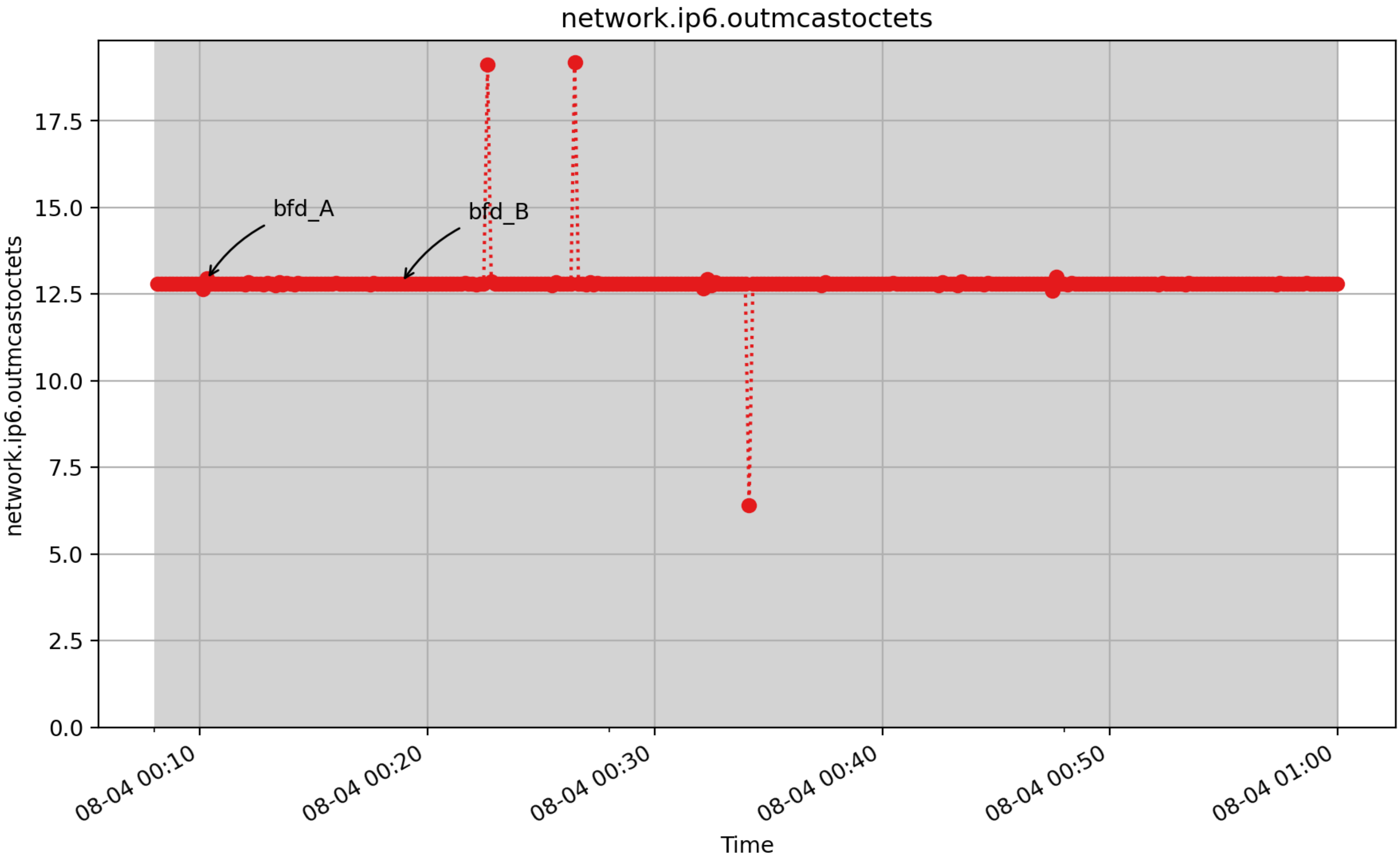
network.ip6.innoroutes: count of ip6 innoroutes (count - U64) - *rate converted*



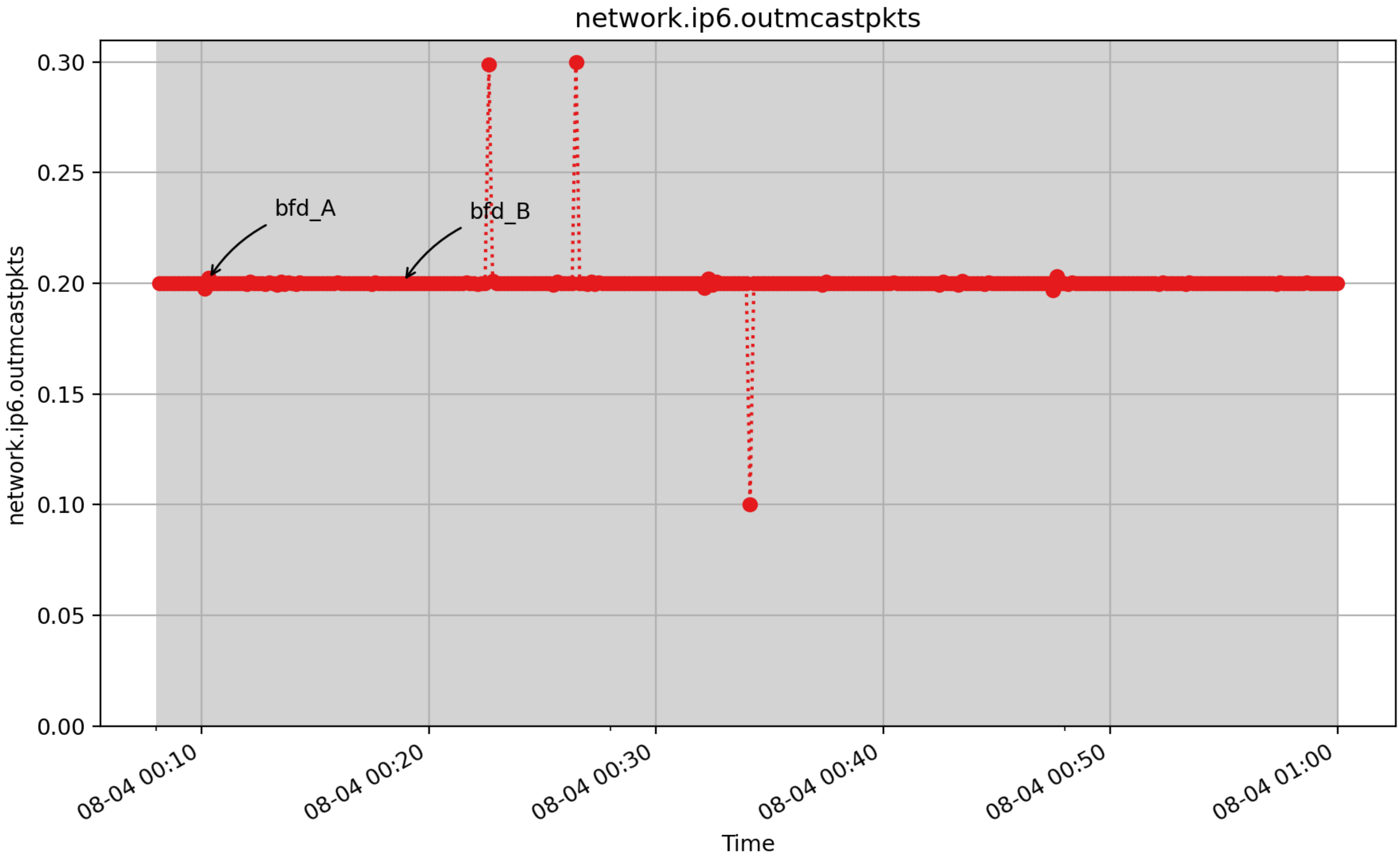
network.ip6.inoctets: count of ip6 octets in (count - U64) - *rate converted*



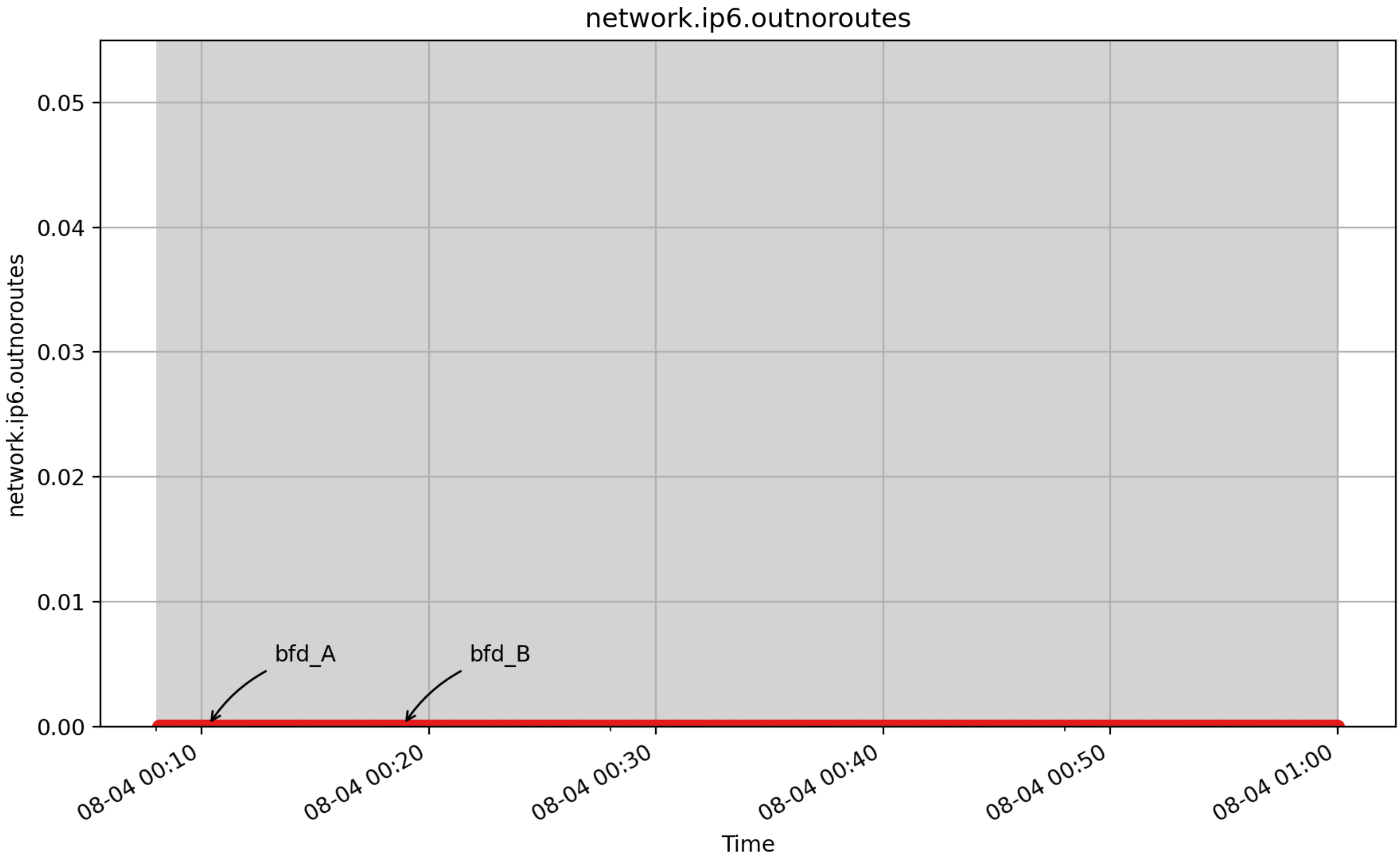
network.ip6.inreceives: count of ip6 inreceives (count - U64) - *rate converted*



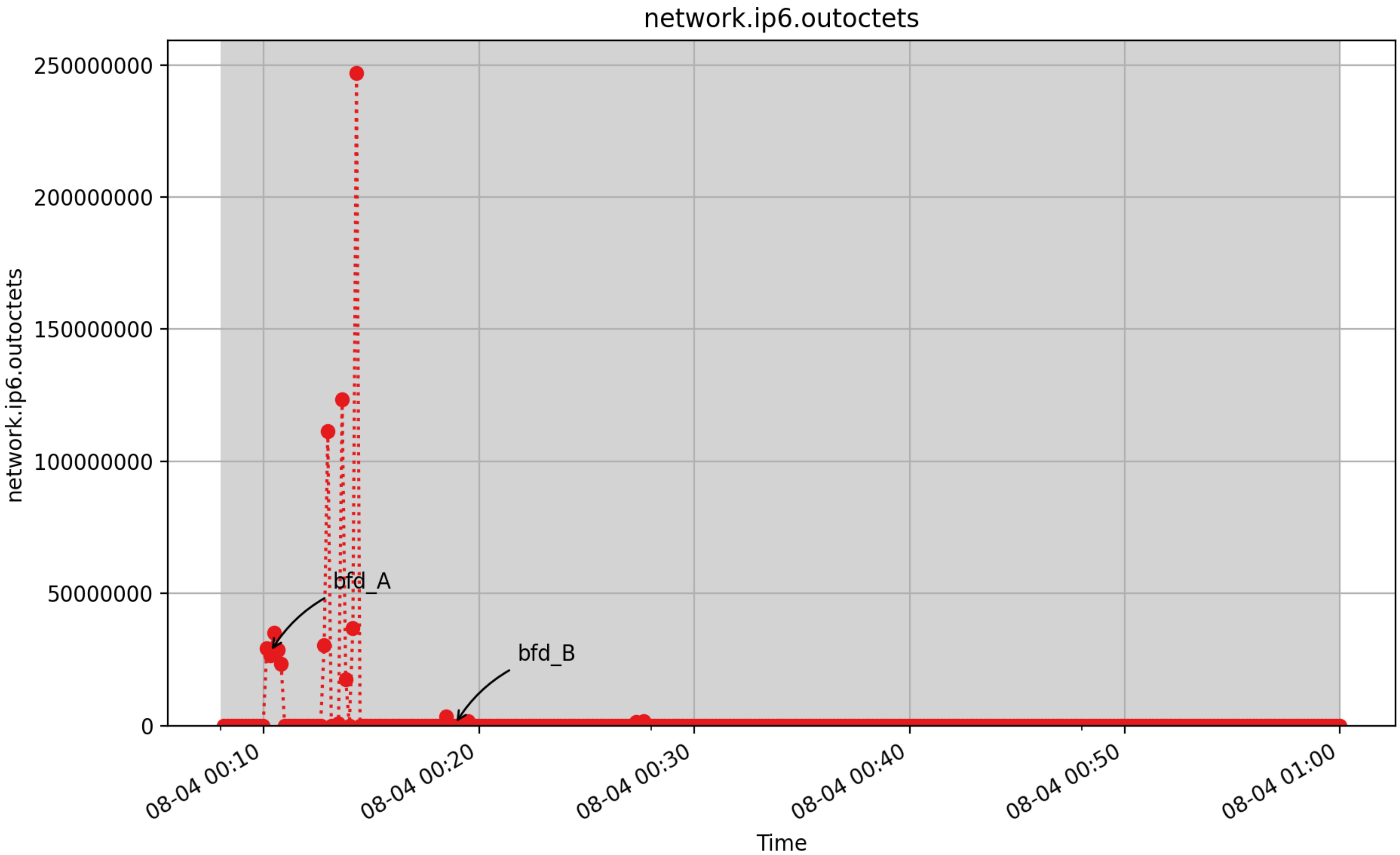
network.ip6.outmcastoctets: count of ip6 multicast octets out (count - U64) - *rate converted*



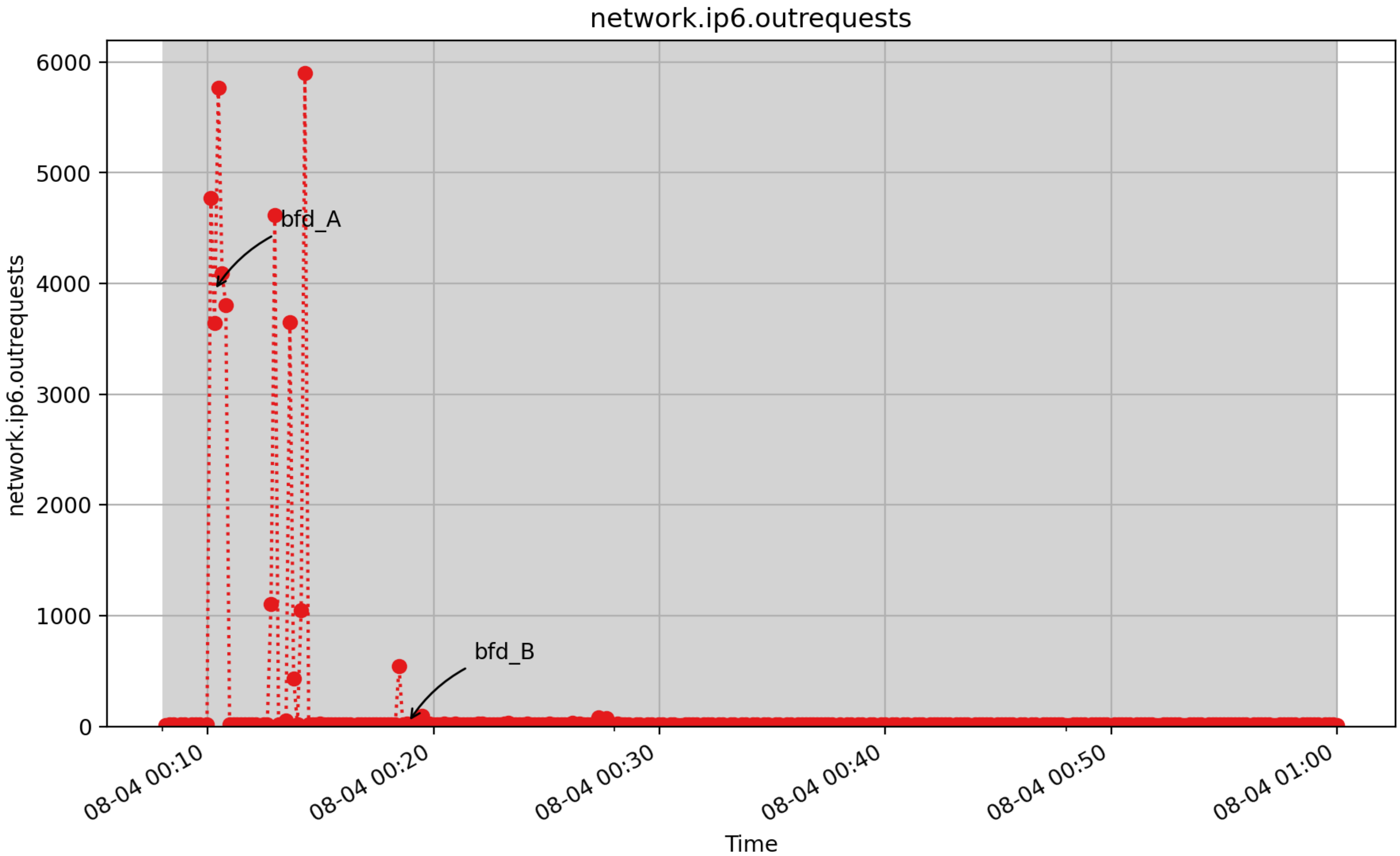
network.ip6.outmcastpkts: count of ip6 multicast packets out (count - U64) - *rate converted*



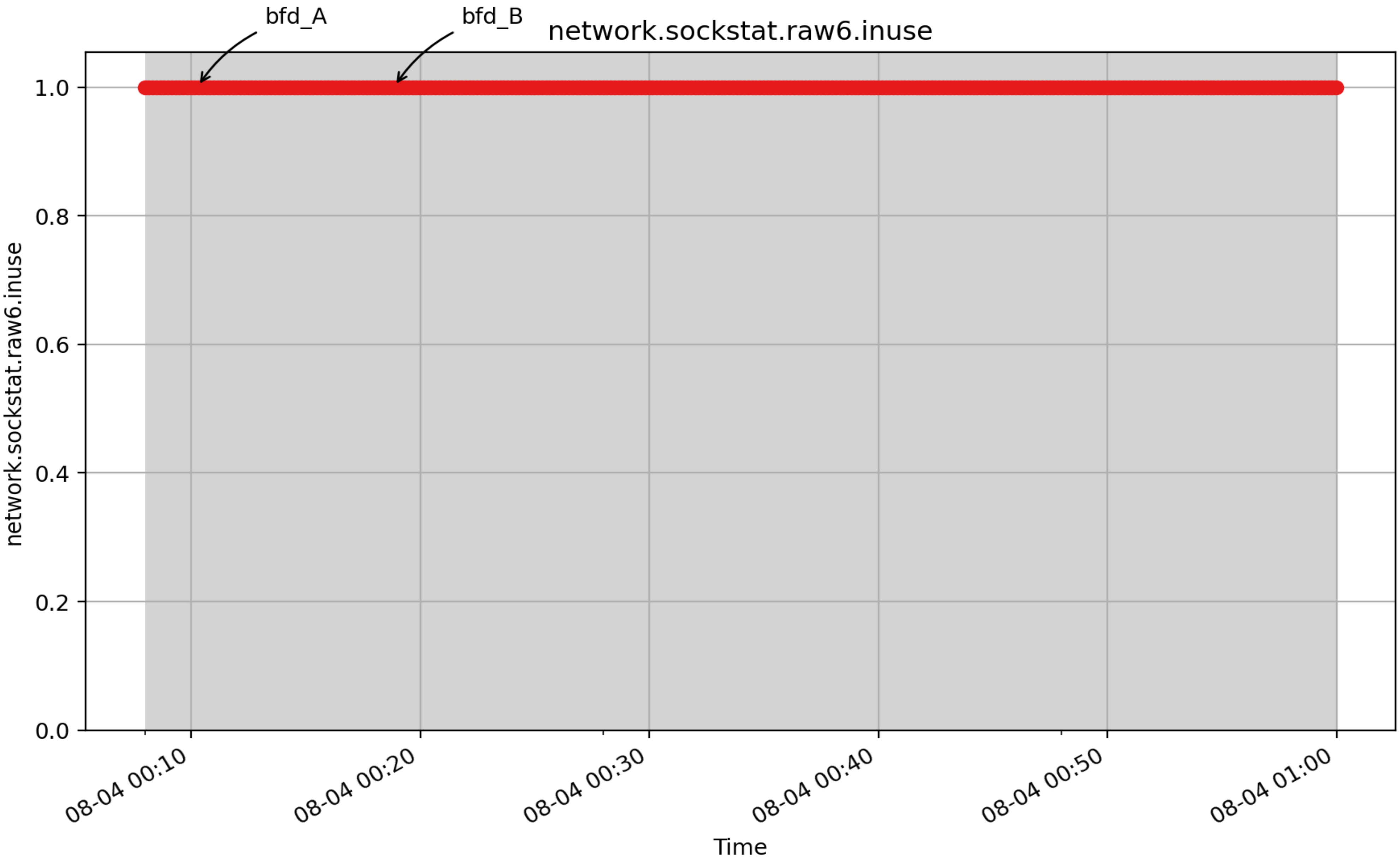
network.ip6.outnoroutes: count of ip6 outnoroutes (count - U64) - *rate converted*



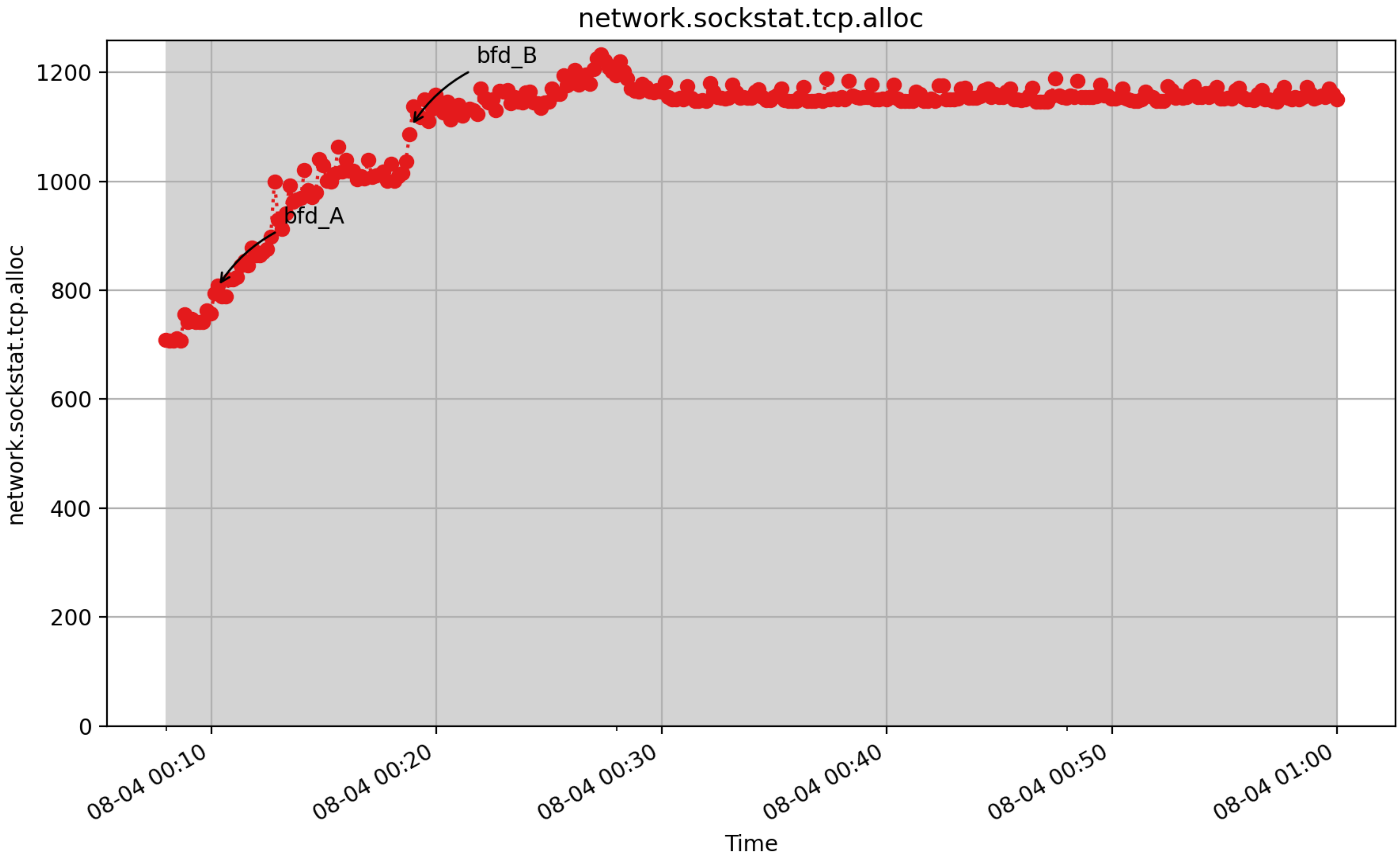
network.ip6.outoctets: count of ip6 octets out (count - U64) - *rate converted*



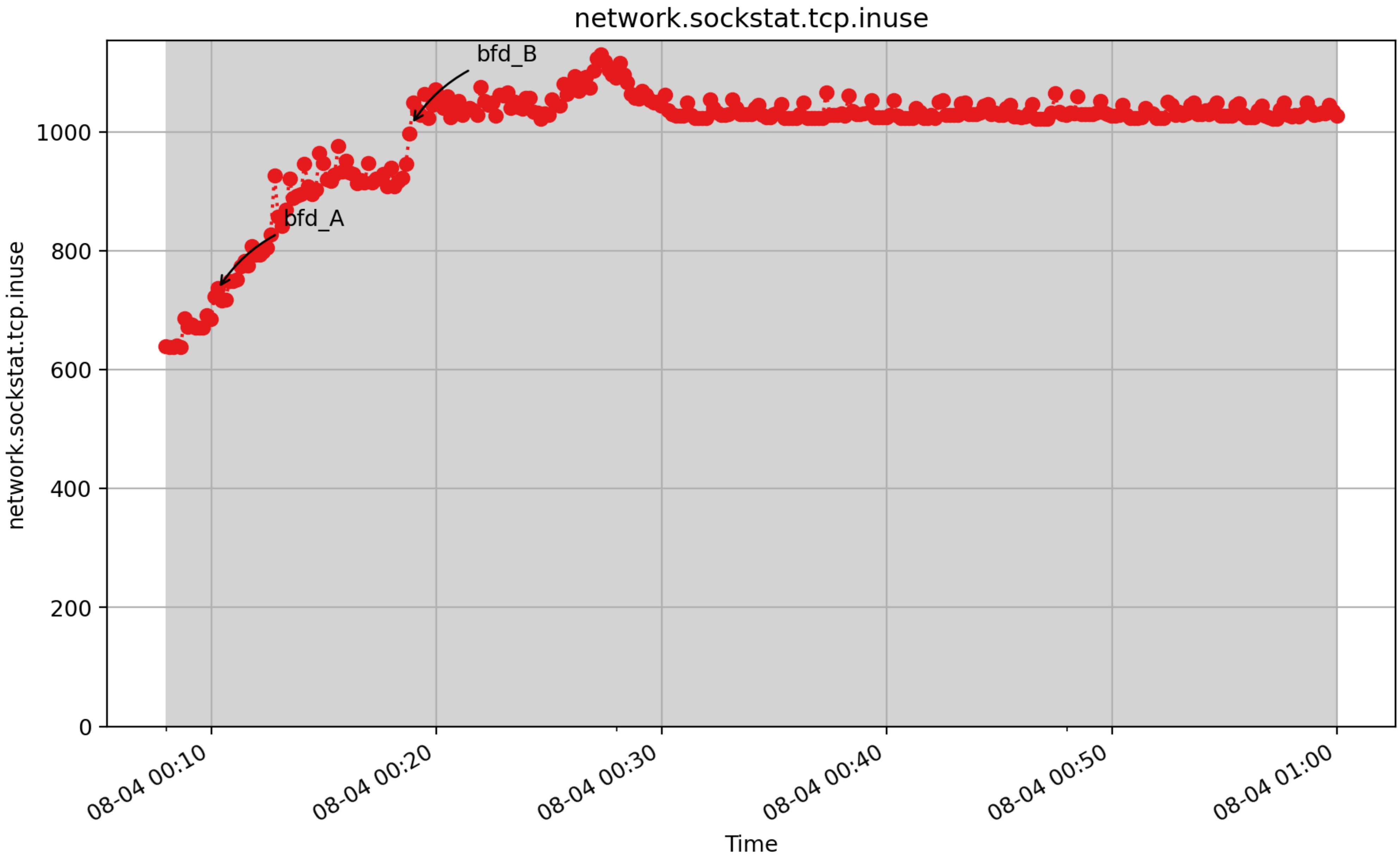
network.ip6.outrequests: count of ip6 outrequests (count - U64) - *rate converted*



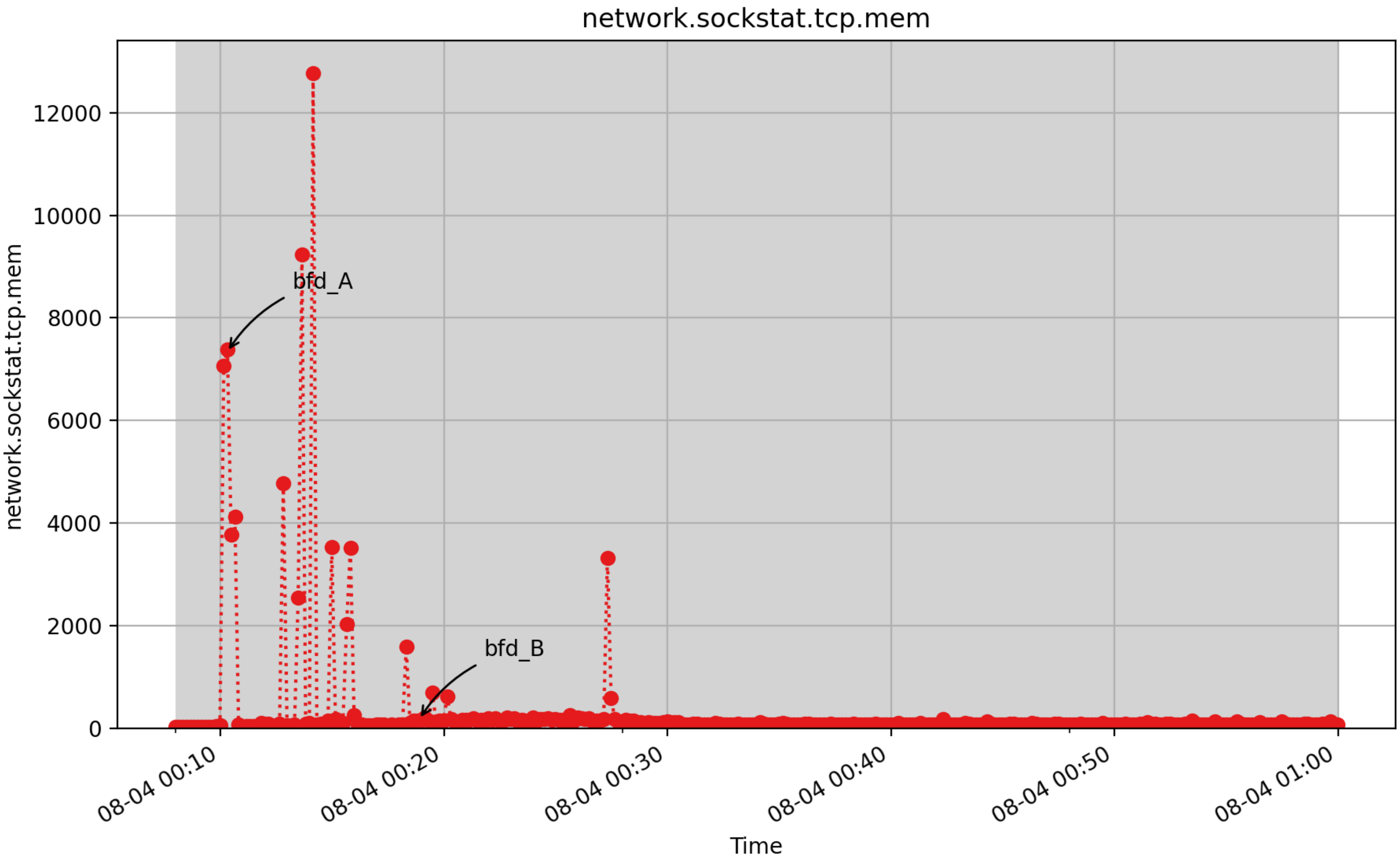
network.sockstat.raw6.inuse: instantaneous number of raw6 sockets currently in use (count - 32)



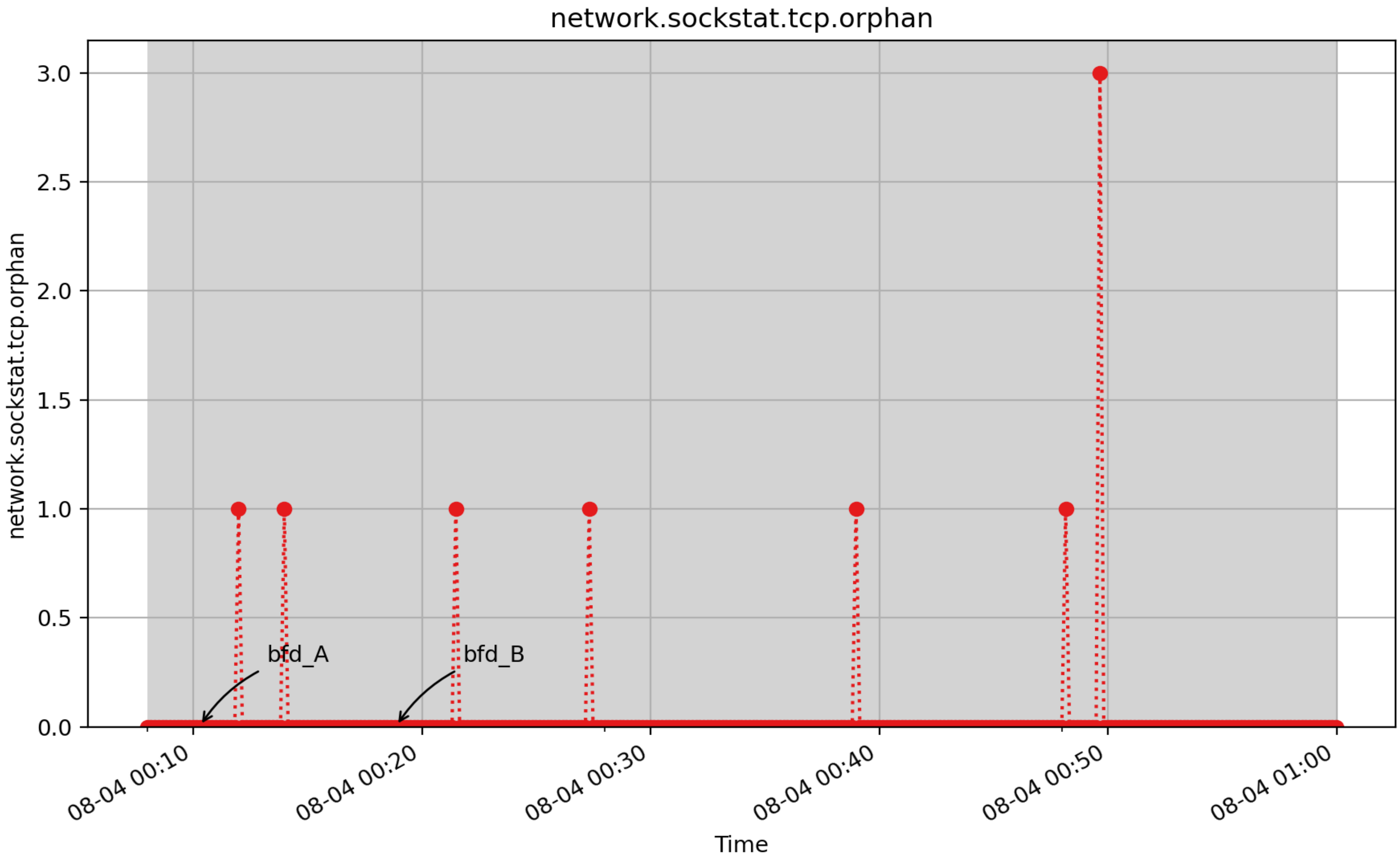
network.sockstat.tcp.alloc: instantaneous number of allocated sockets (count - 32)



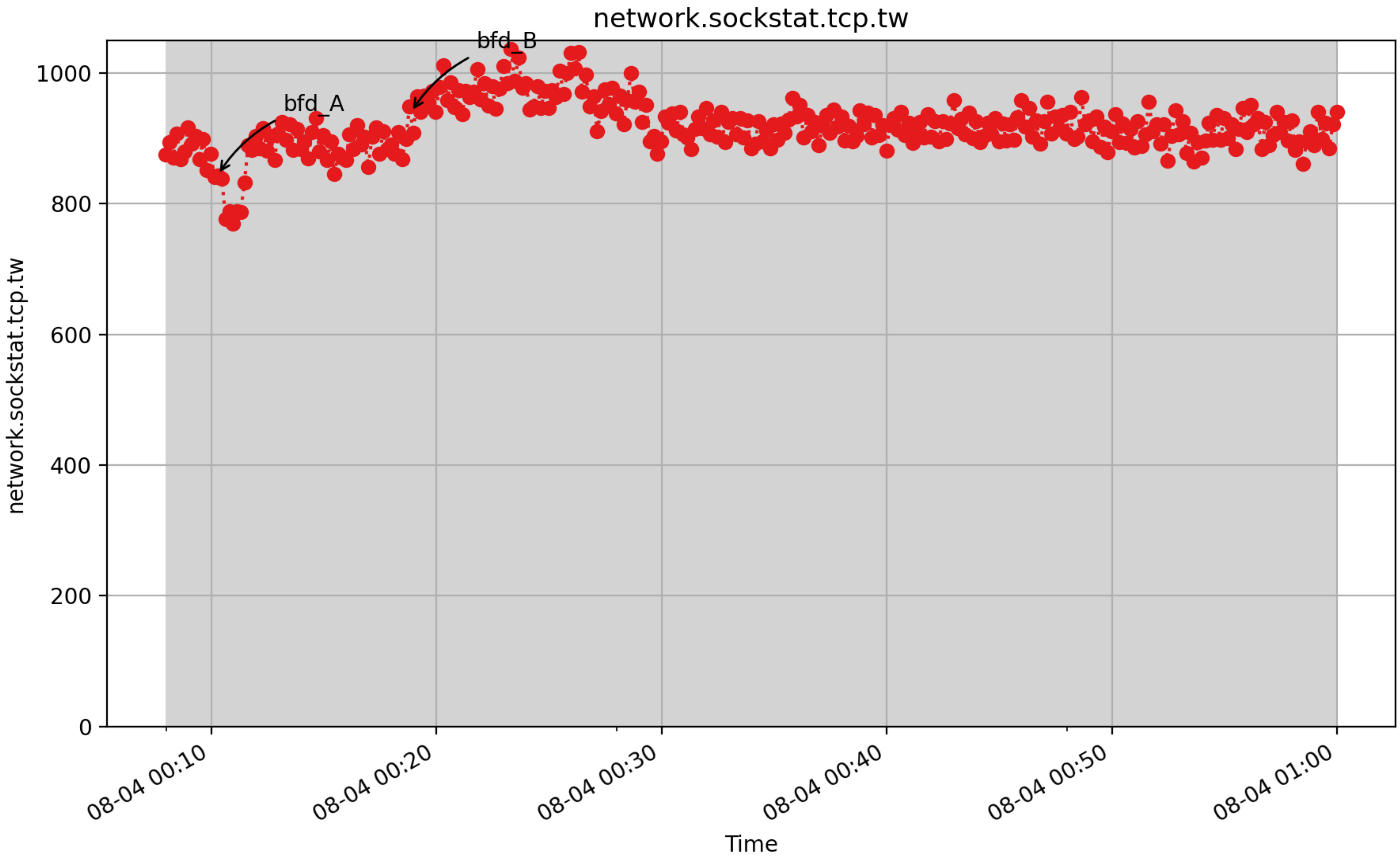
network.sockstat.tcp.inuse: instantaneous number of tcp sockets currently in use (count - 32)



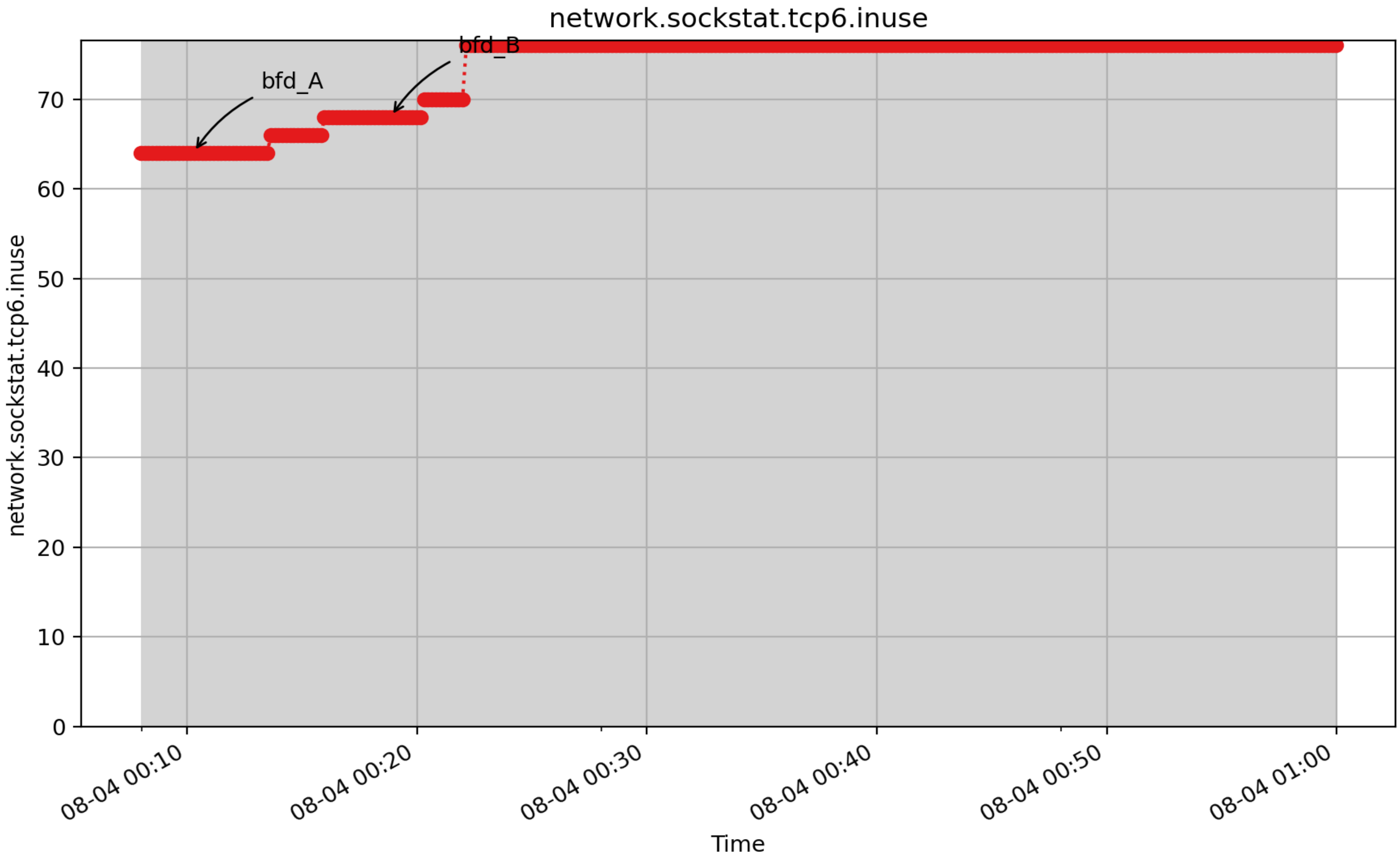
network.sockstat.tcp.mem: instantaneous number of used memory for tcp (count - 32)



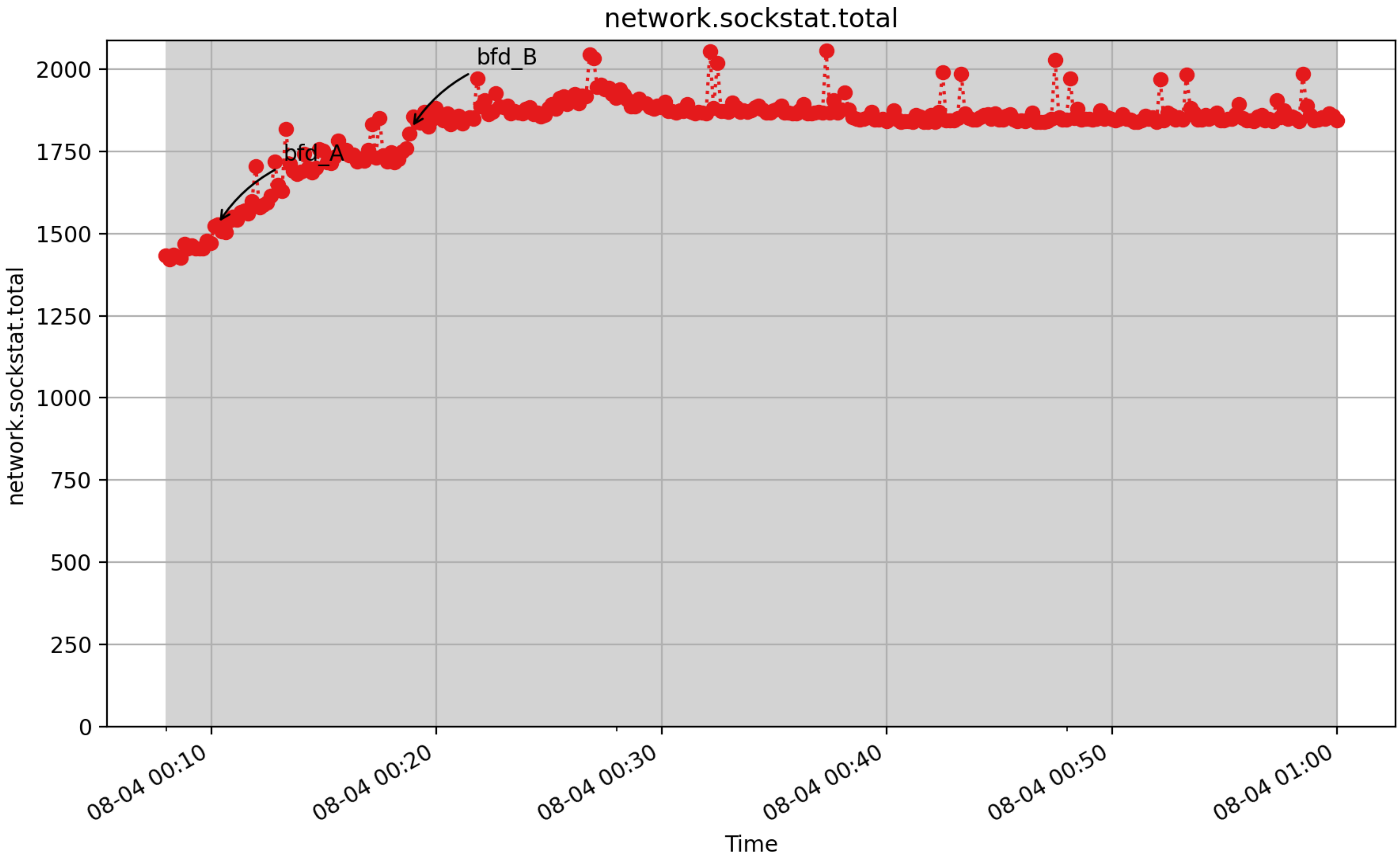
network.sockstat.tcp.orphan: instantaneous number of orphan sockets (count - 32)



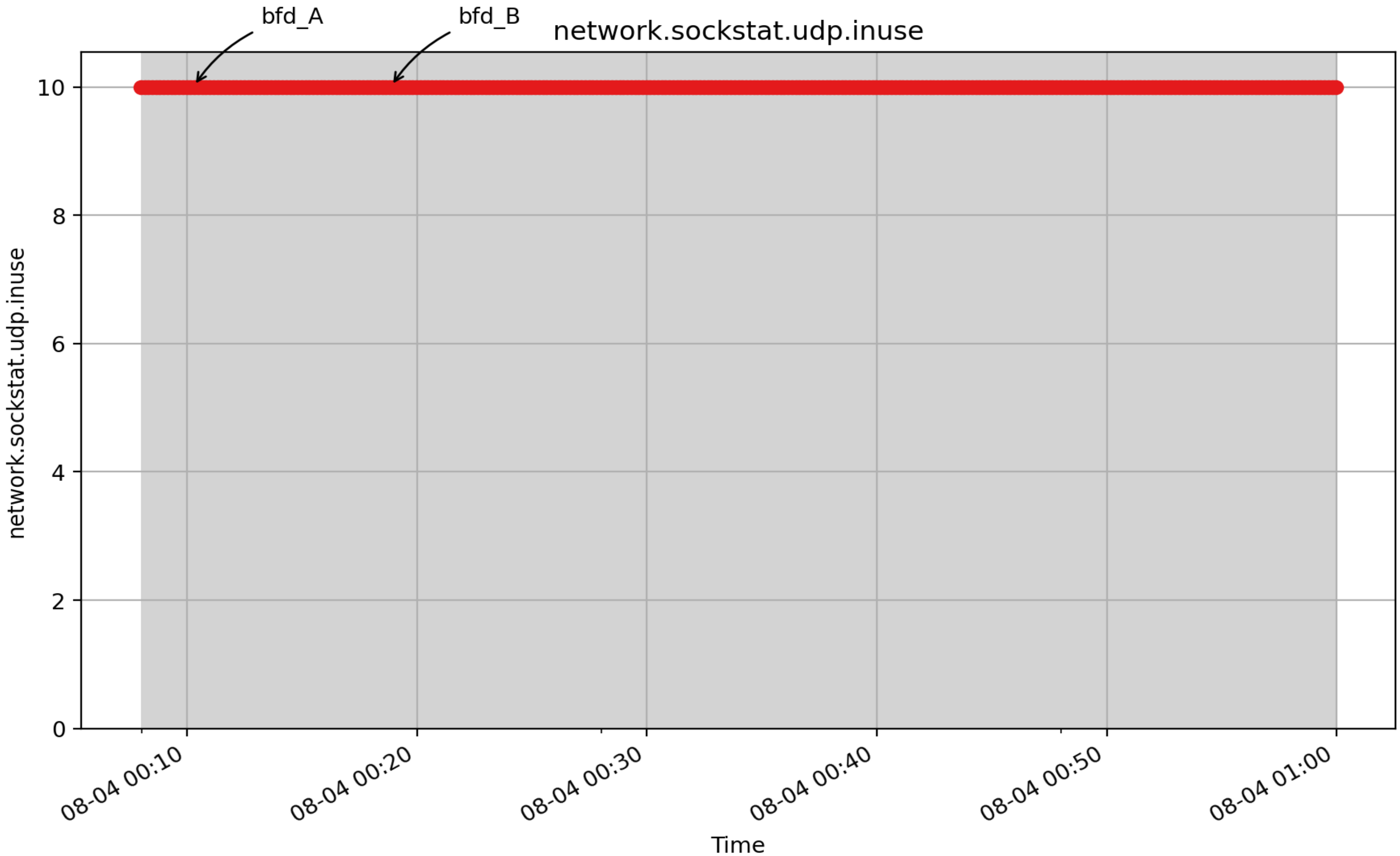
network.sockstat.tcp.tw: instantaneous number of sockets waiting close (count - 32)



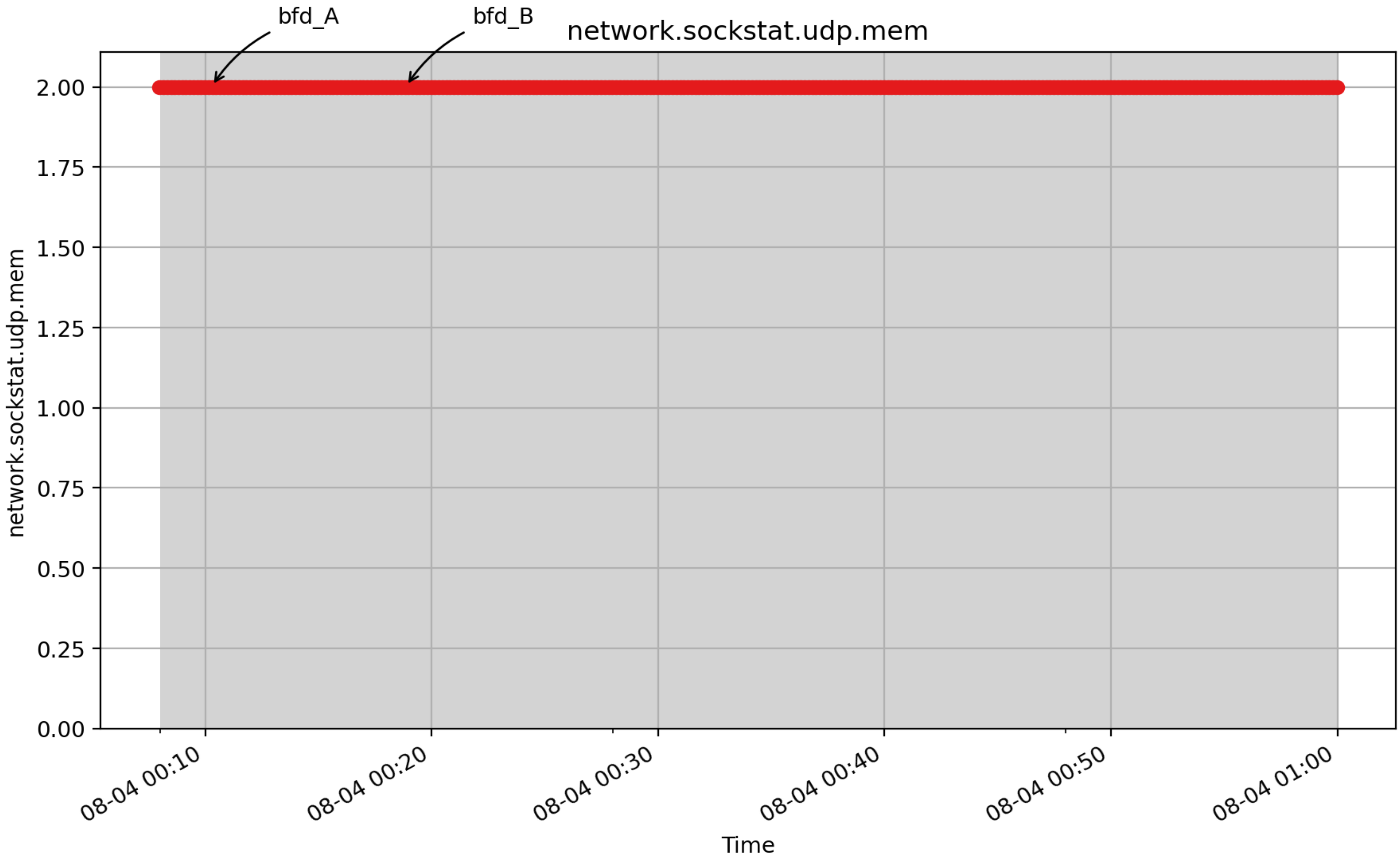
network.sockstat.tcp6.inuse: instantaneous number of tcp6 sockets currently in use (count - 32)



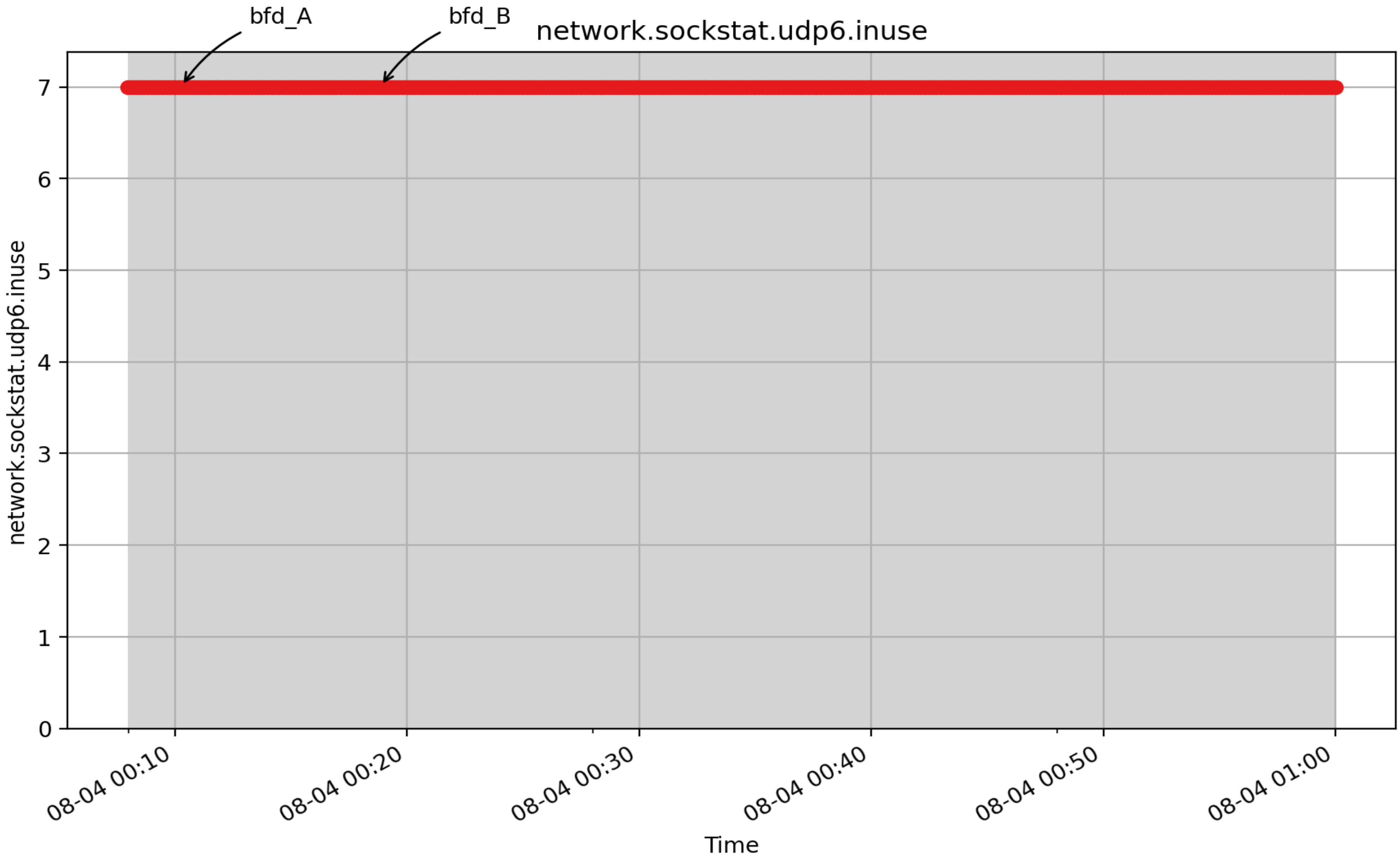
network.sockstat.total: total number of sockets used by the system. (count - 32)



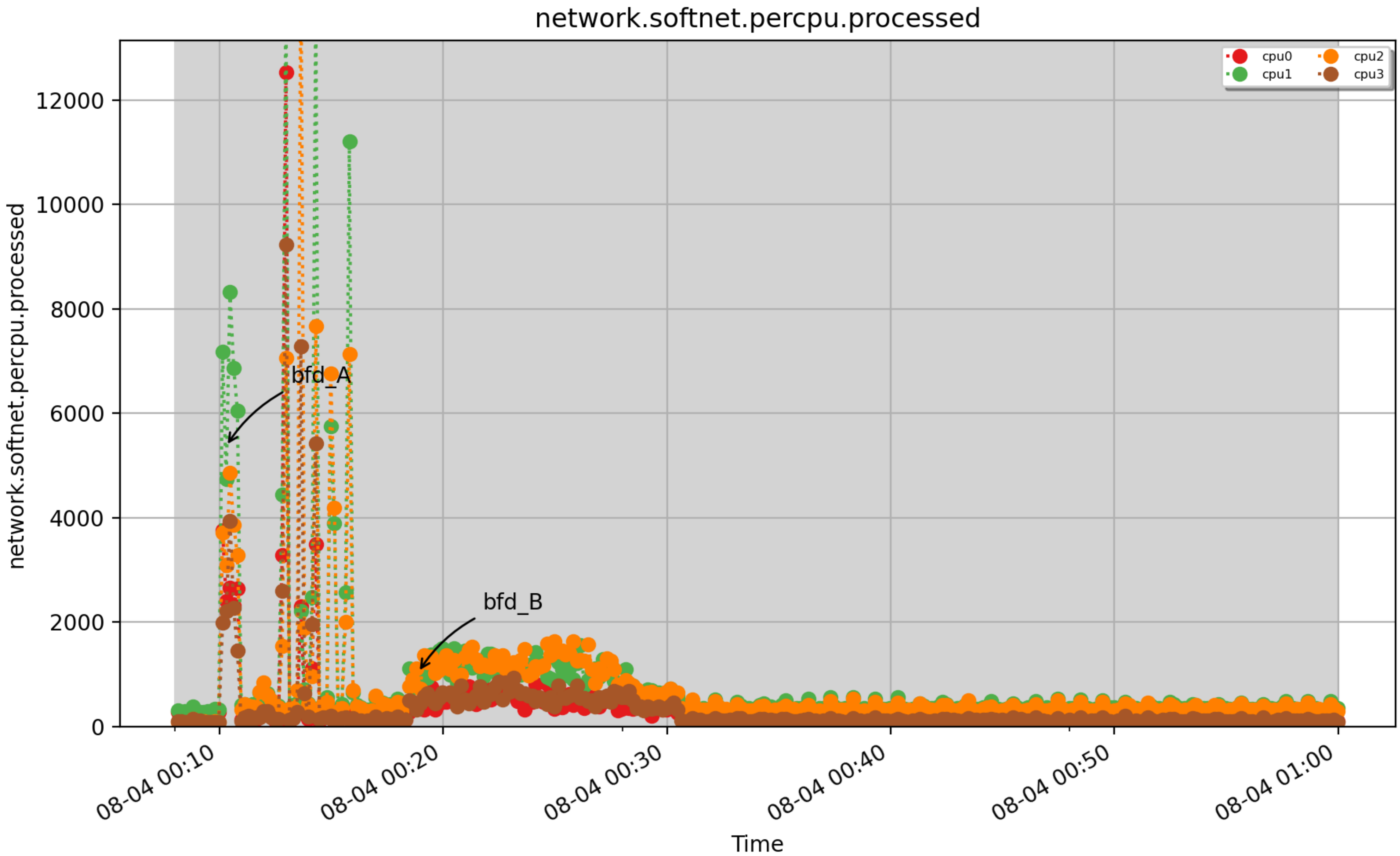
network.sockstat.udp.inuse: instantaneous number of udp sockets currently in use (count - 32)



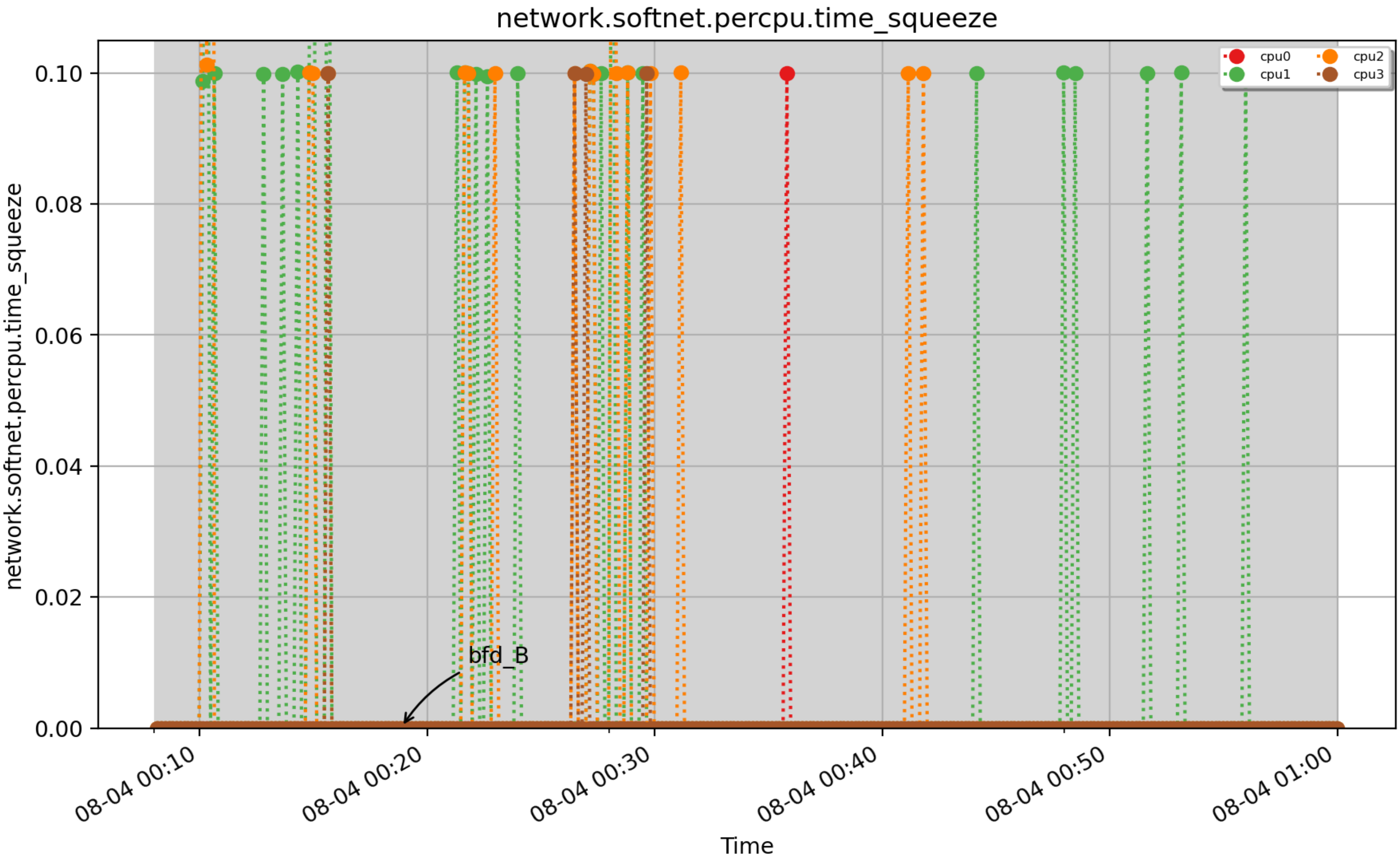
network.sockstat.udp.mem: instantaneous number of used memory for udp (count - 32)



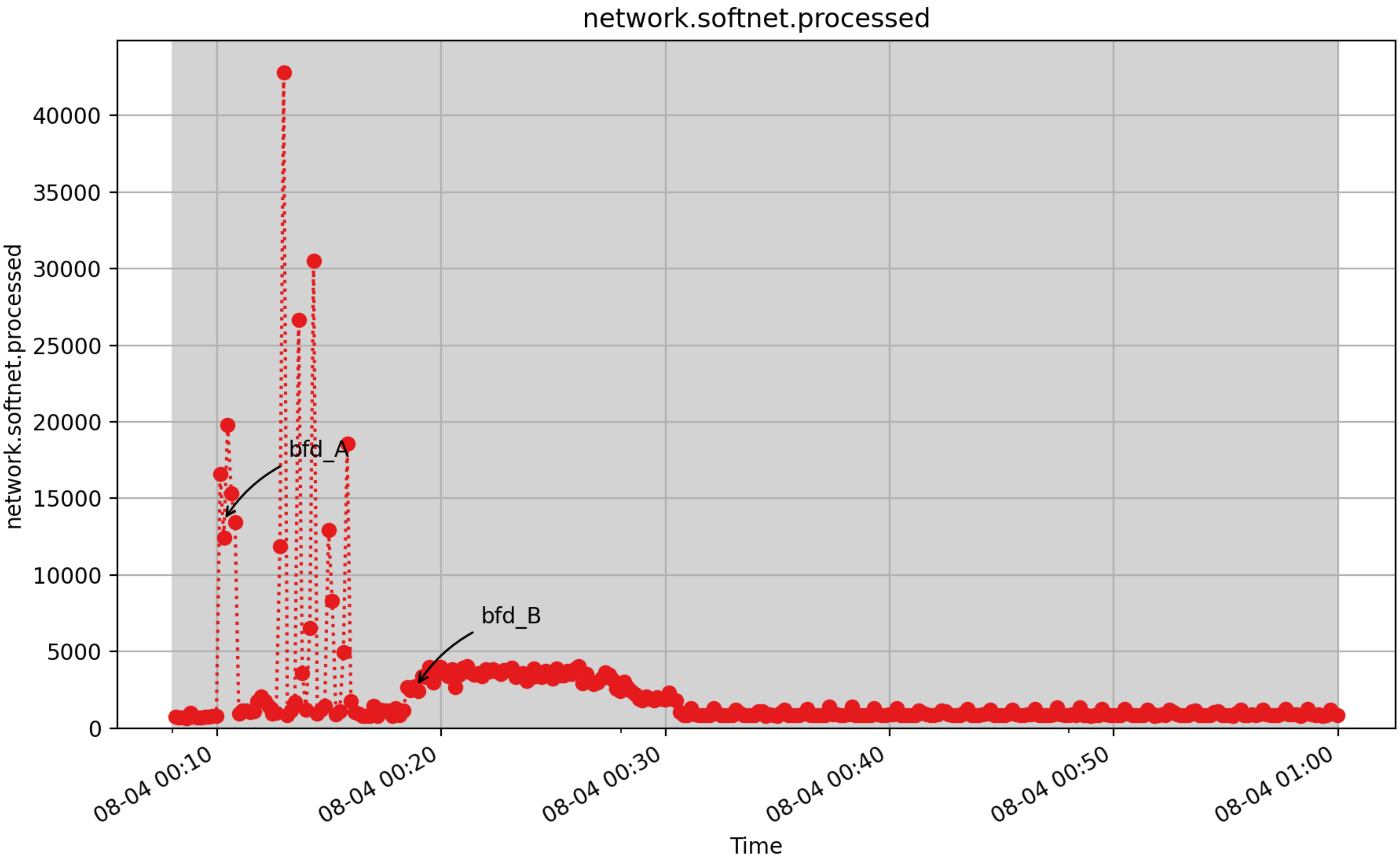
network.sockstat.udp6.inuse: instantaneous number of udp6 sockets currently in use (count - 32)



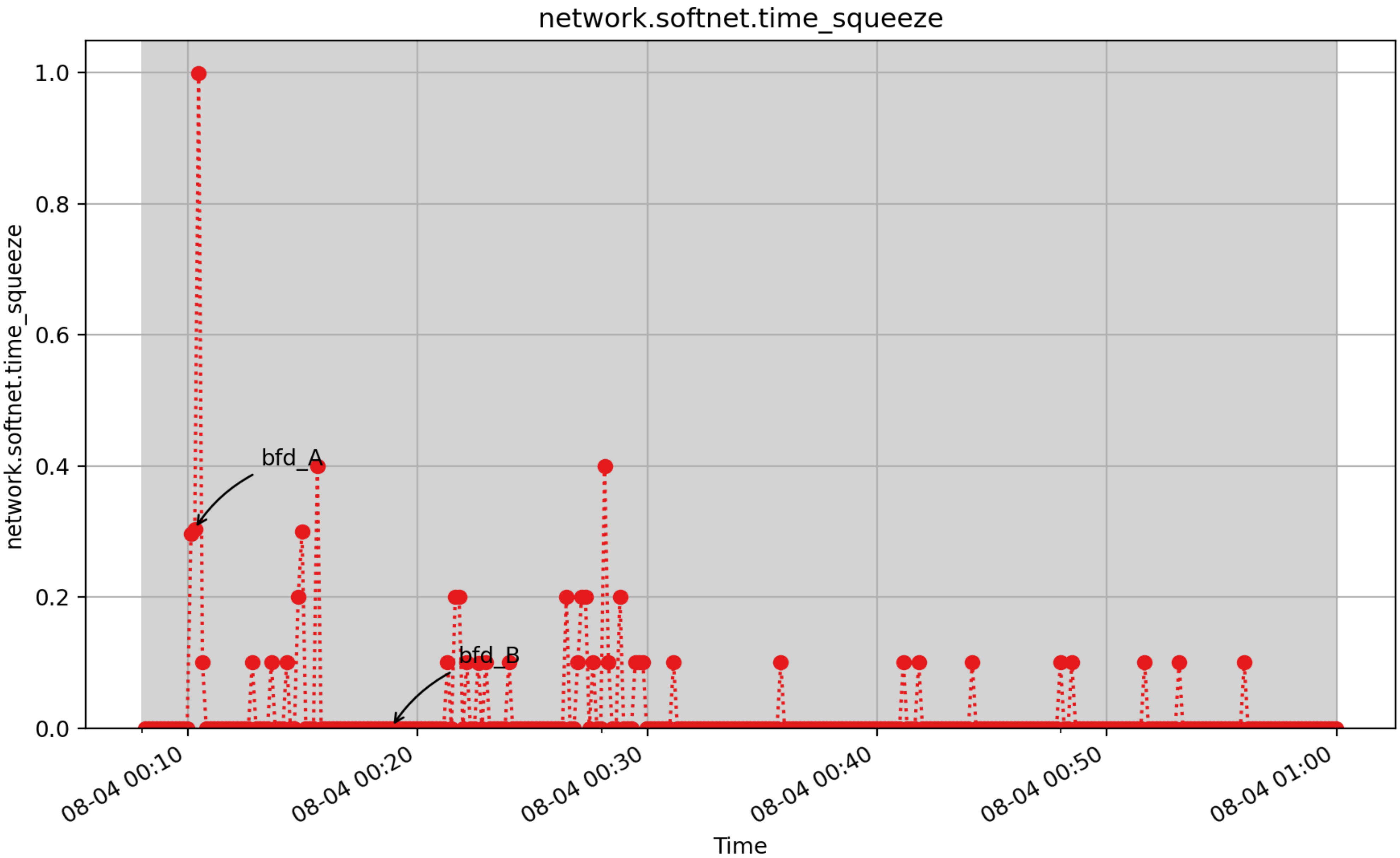
network.softnet.percpu.processed: number of packets (not including netpoll) received by the interrupt handler (count - U64) - *rate converted*



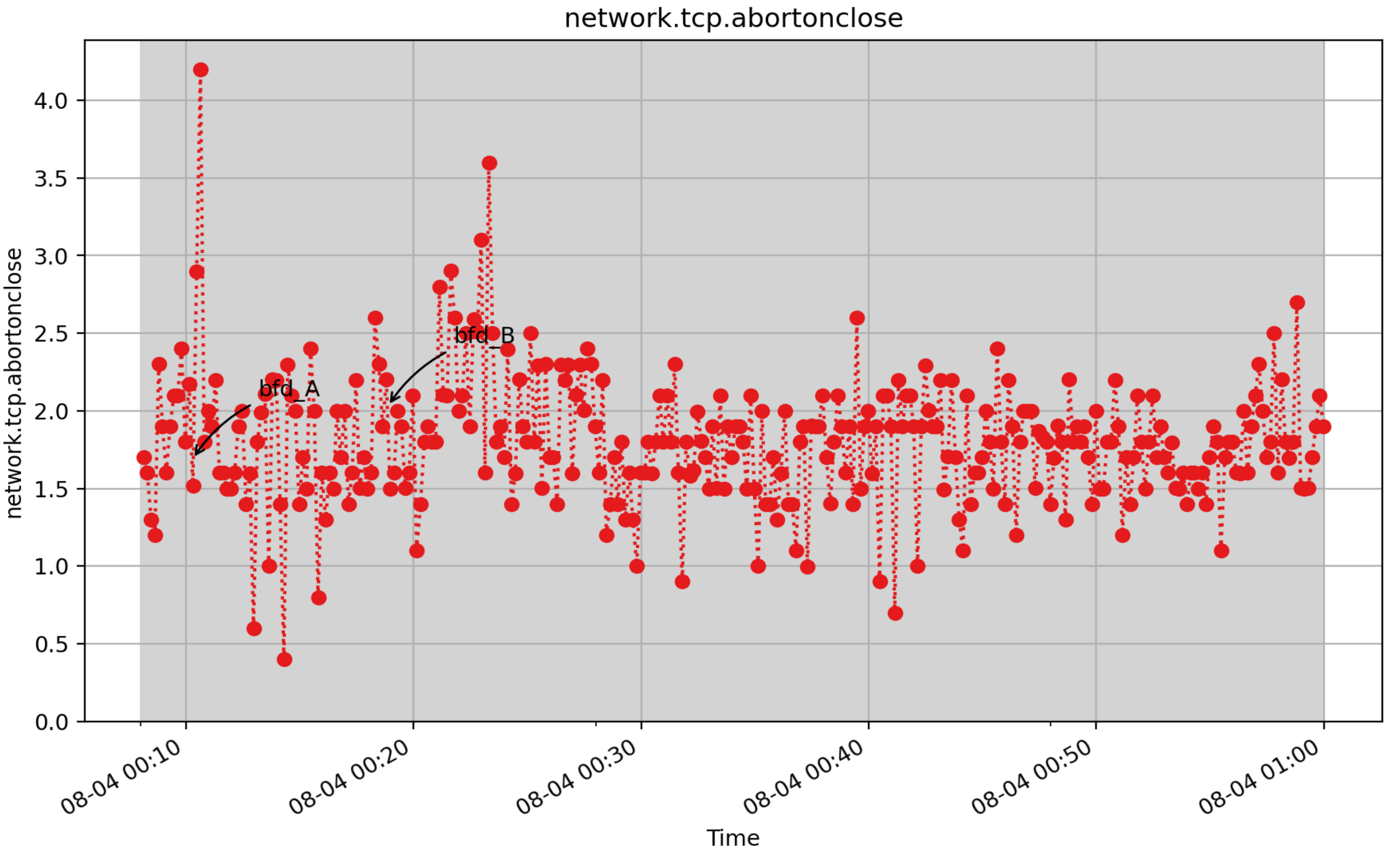
network.softnet.percpu.time_squeeze: number of times ksoftirq ran out of netdev_budget or time slice with work remaining (count - U64) - *rate converted*



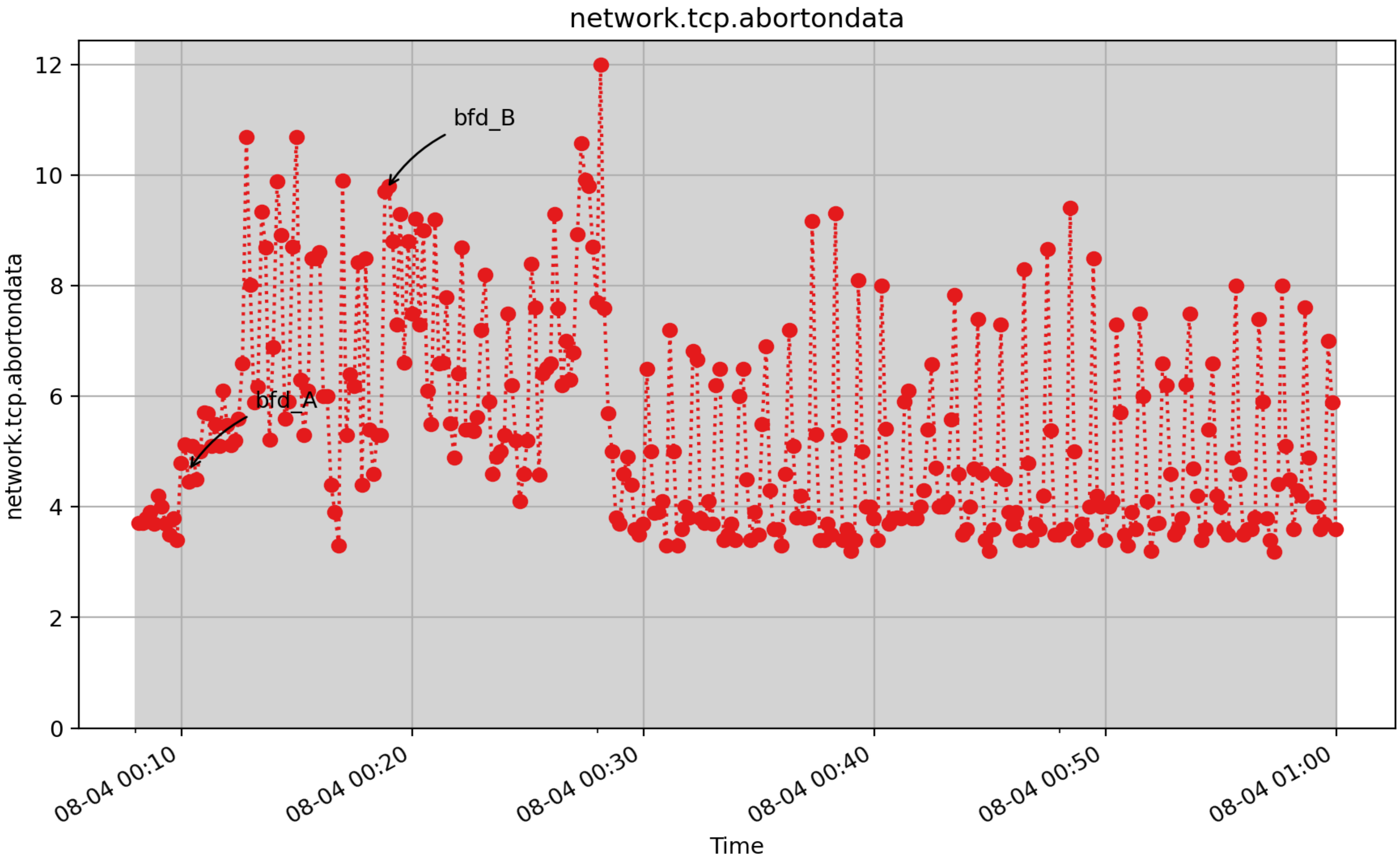
network.softnet.processed: number of packets (not including netpoll) received by the interrupt handler (count - U64) - *rate converted*



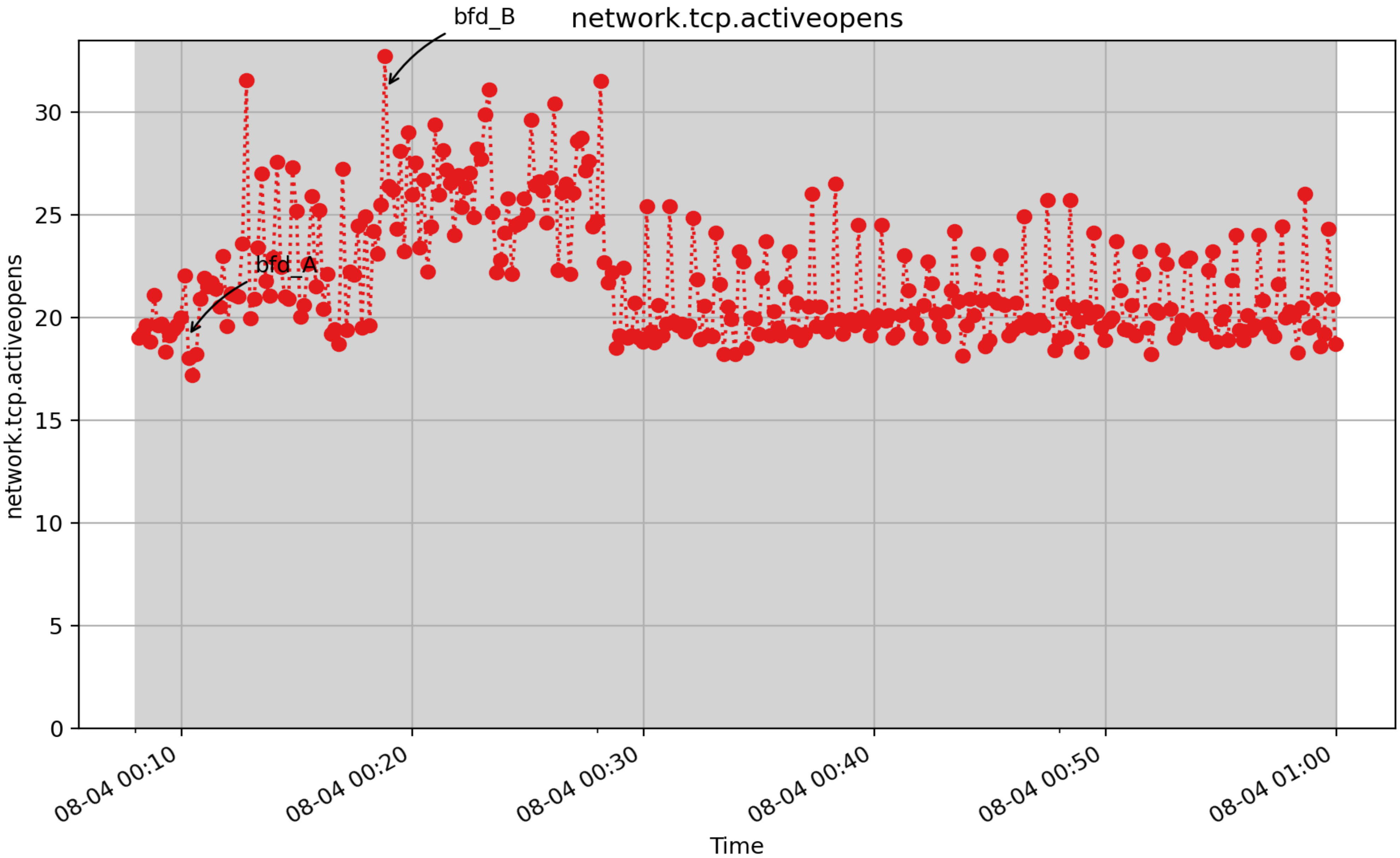
network.softnet.time_squeeze: number of times ksoftirq ran out of netdev_budget or time slice with work remaining (count - U64) - *rate converted*



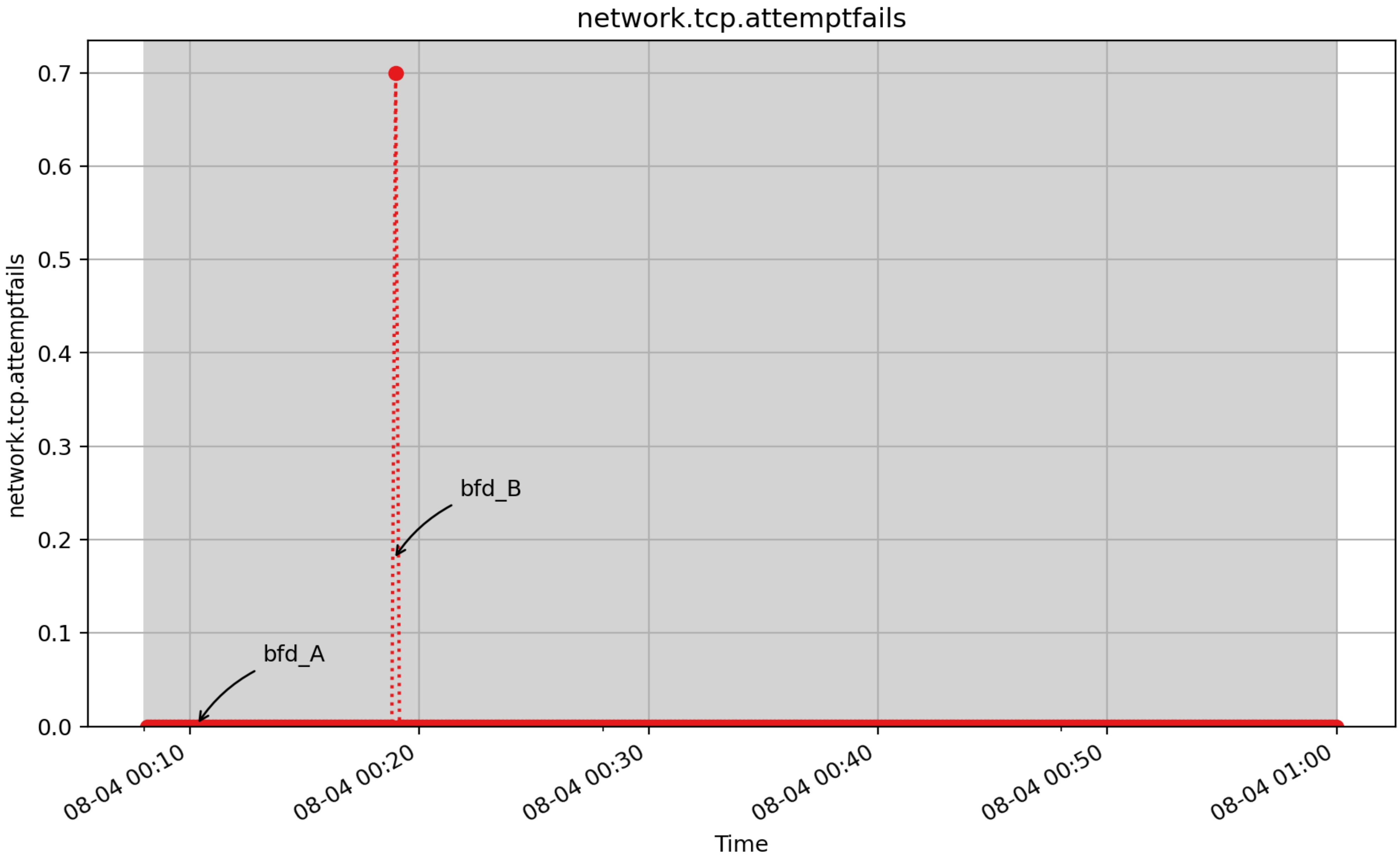
network.tcp.abortonclose: Number of connections reset due to early user close (count - U64) - *rate converted*



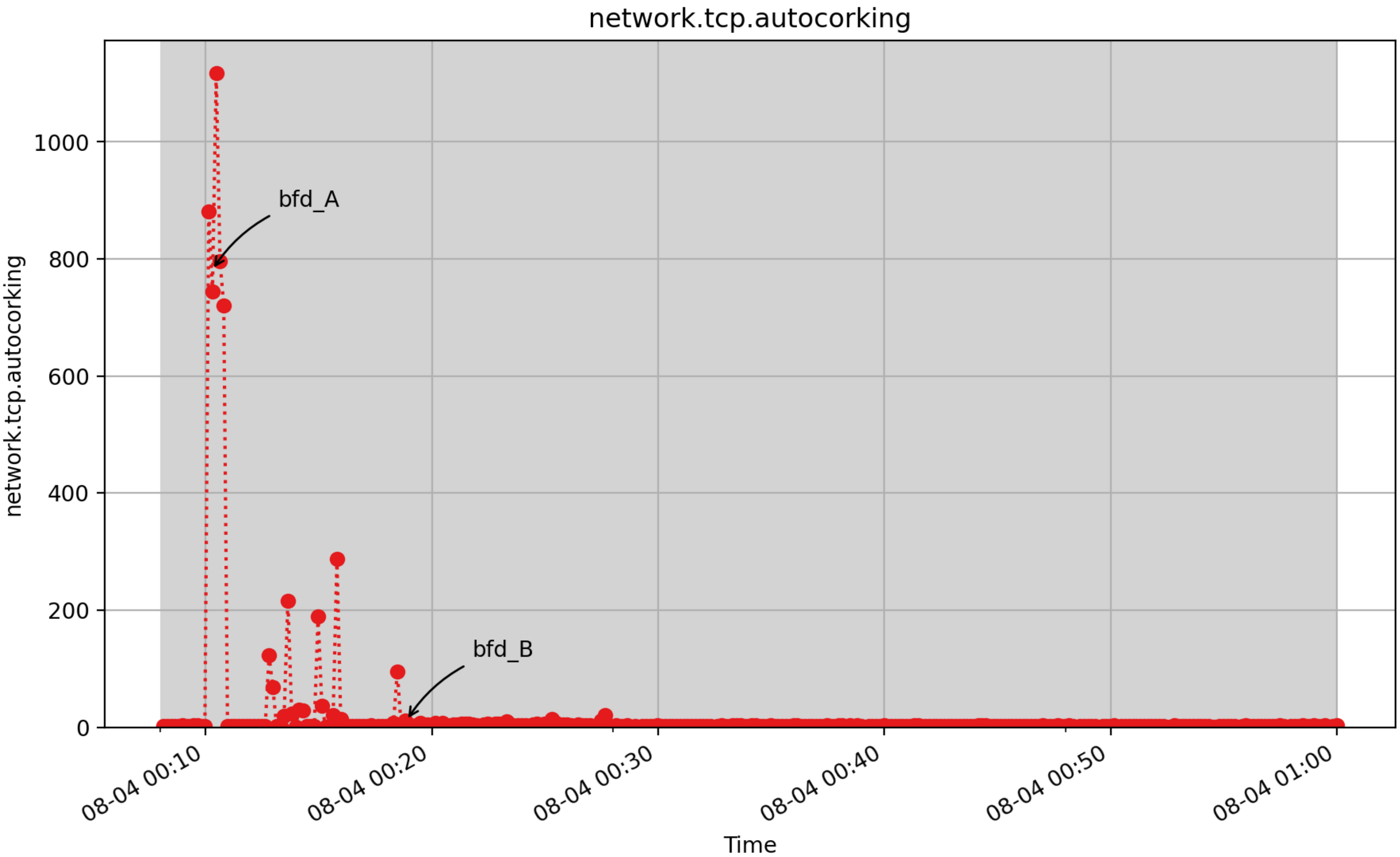
network.tcp.abortondata: Number of connections reset due to unexpected data (count - U64) - *rate converted*



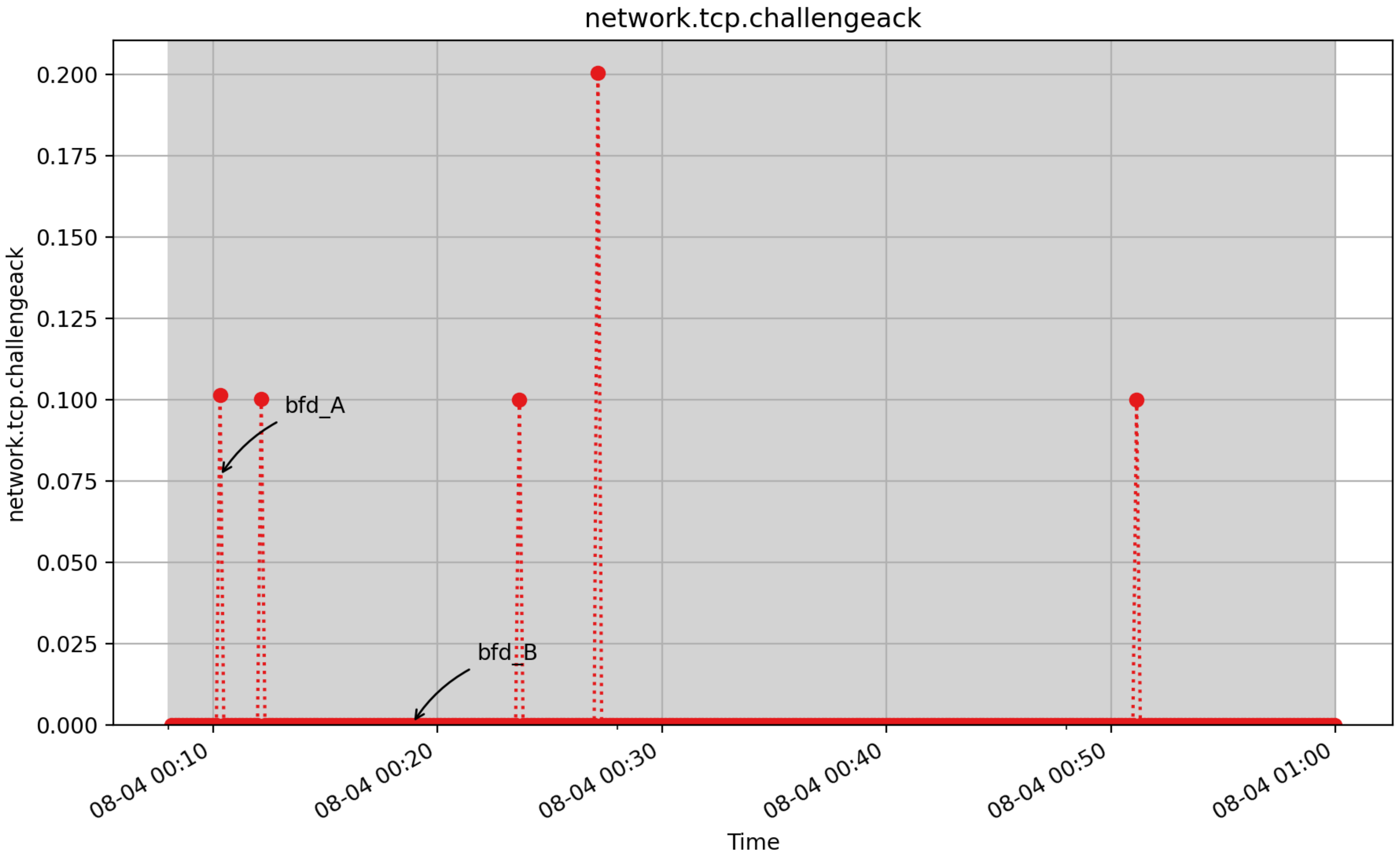
network.tcp.activeopens: count of tcp activeopens (count - U64) - *rate converted*



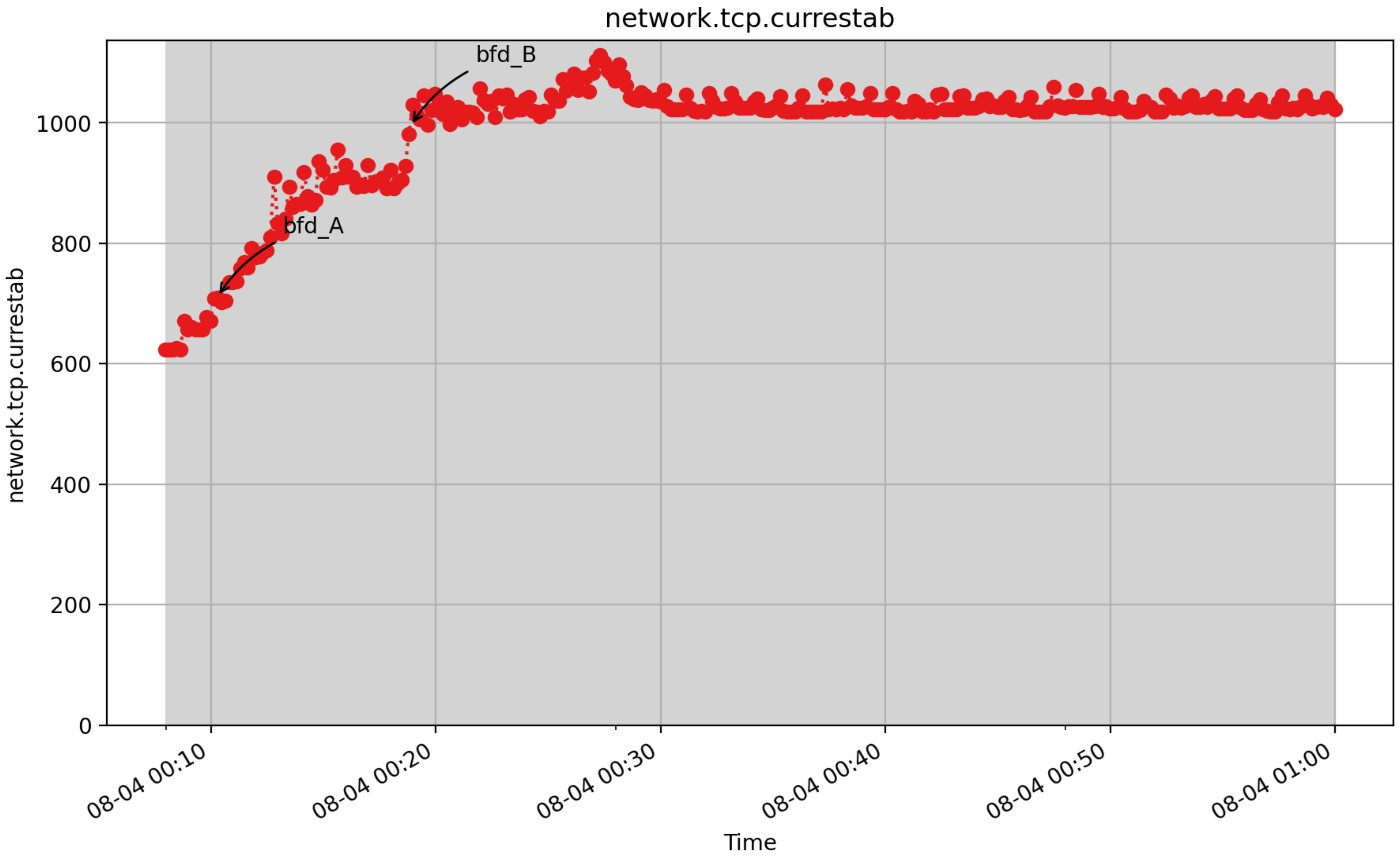
network.tcp.attemptfails: count of tcp attemptfails (count - U64) - *rate converted*



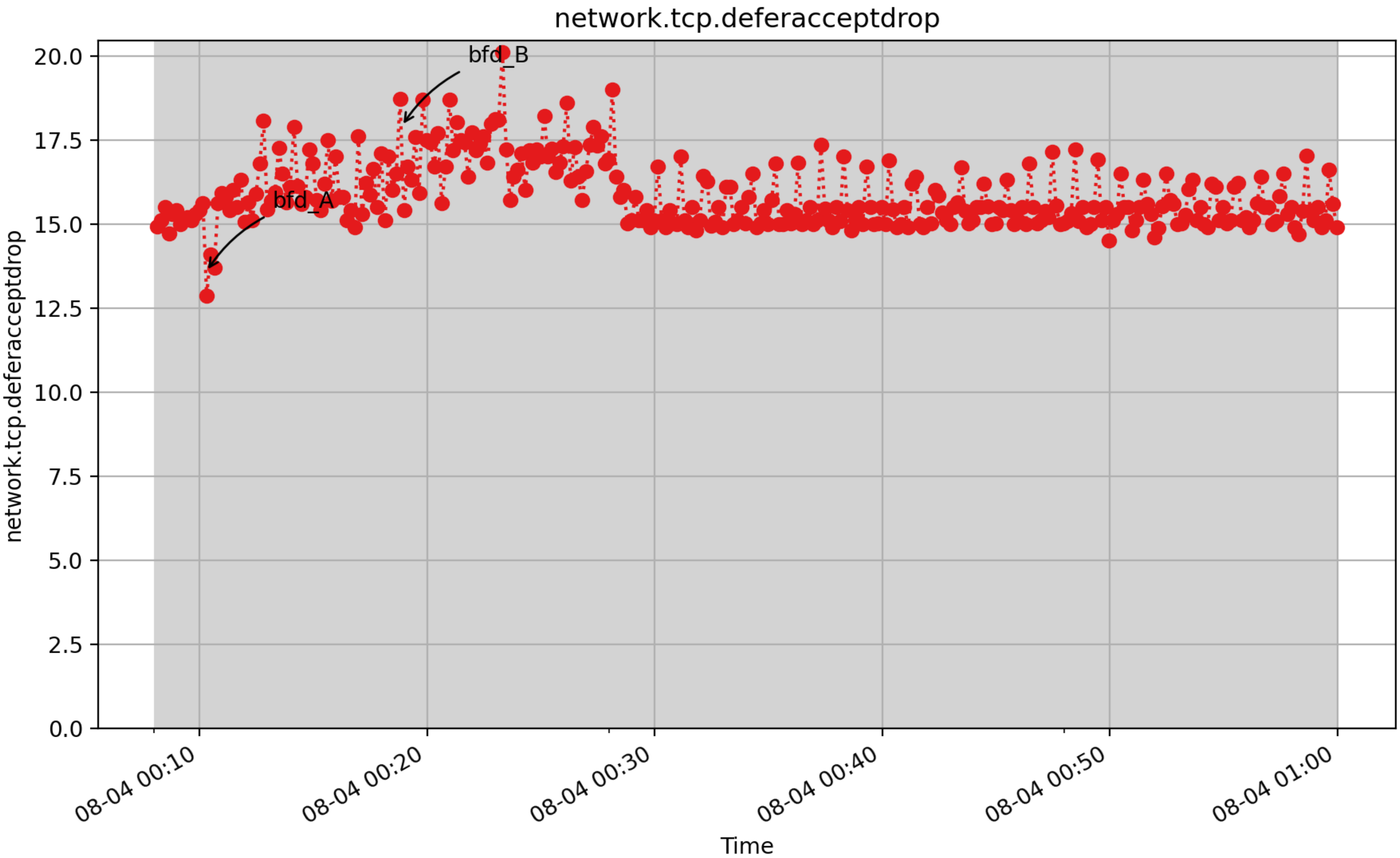
network.tcp.autocorking: Number of times stack detected skb was underused and its flush was deferred (count - U64) - *rate converted*



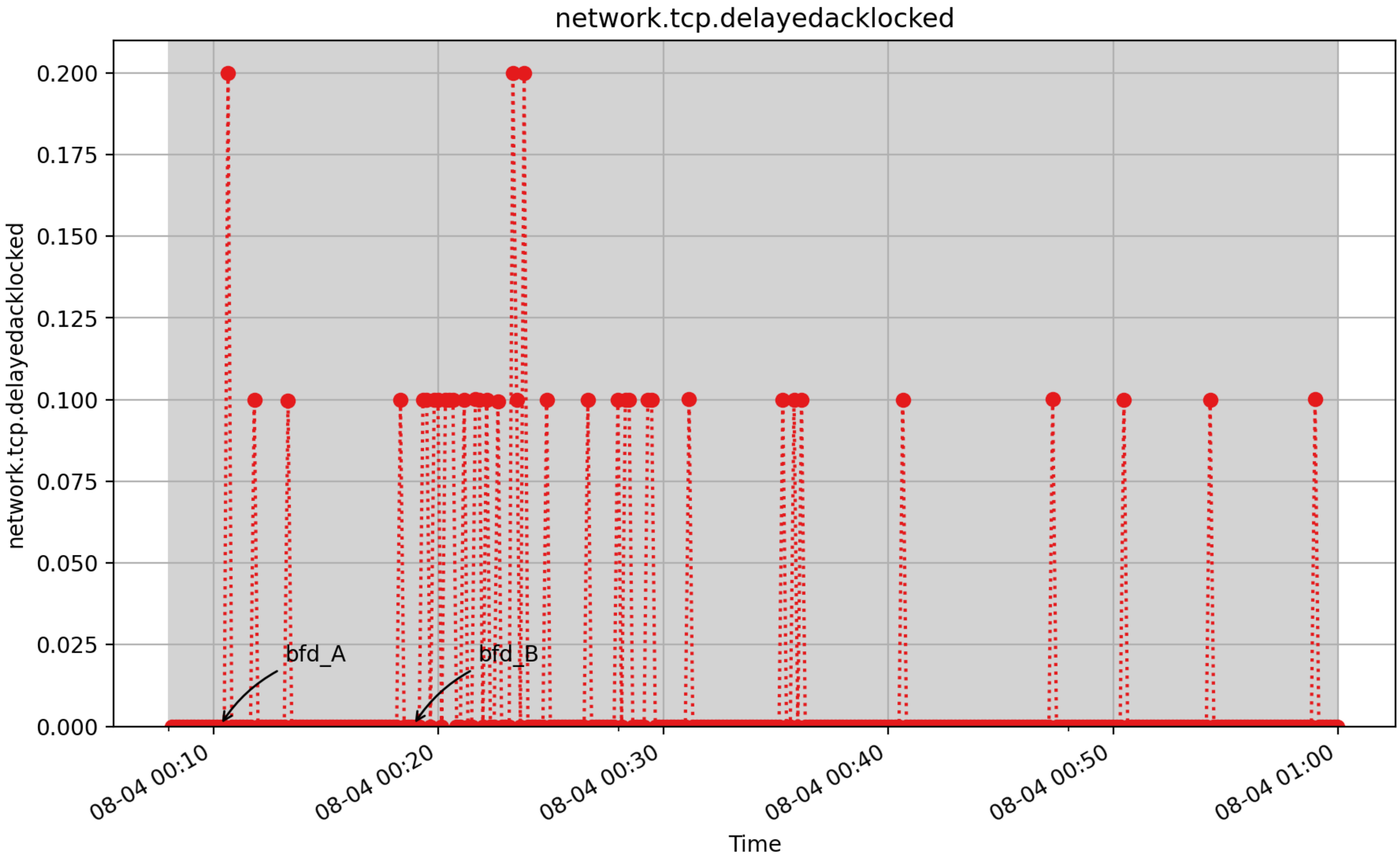
network.tcp.challengeack: Number of challenge ACKs sent (RFC 5961 3.2) (count - U64) - *rate converted*



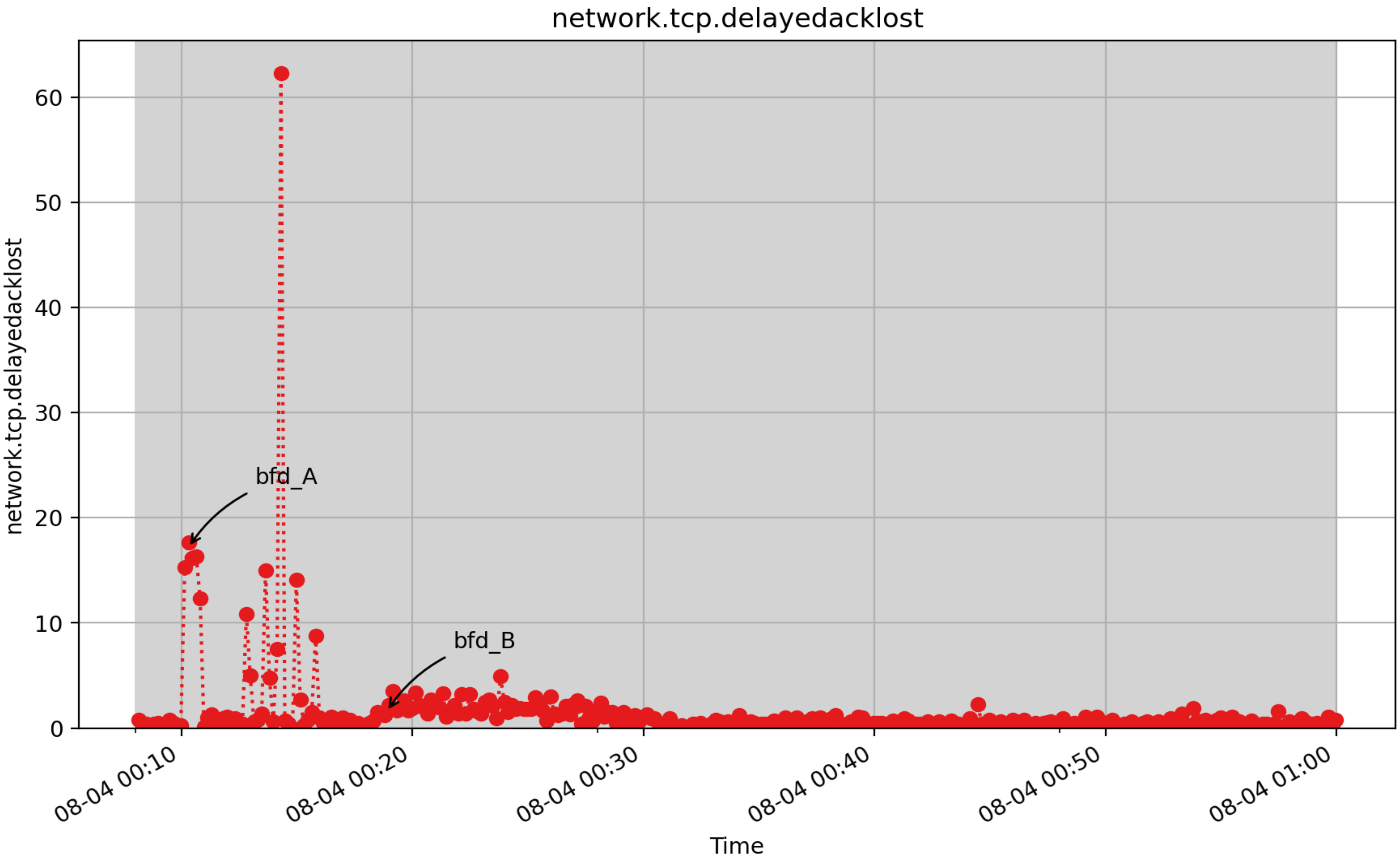
network.tcp.currestab: current established tcp connections (- U64)



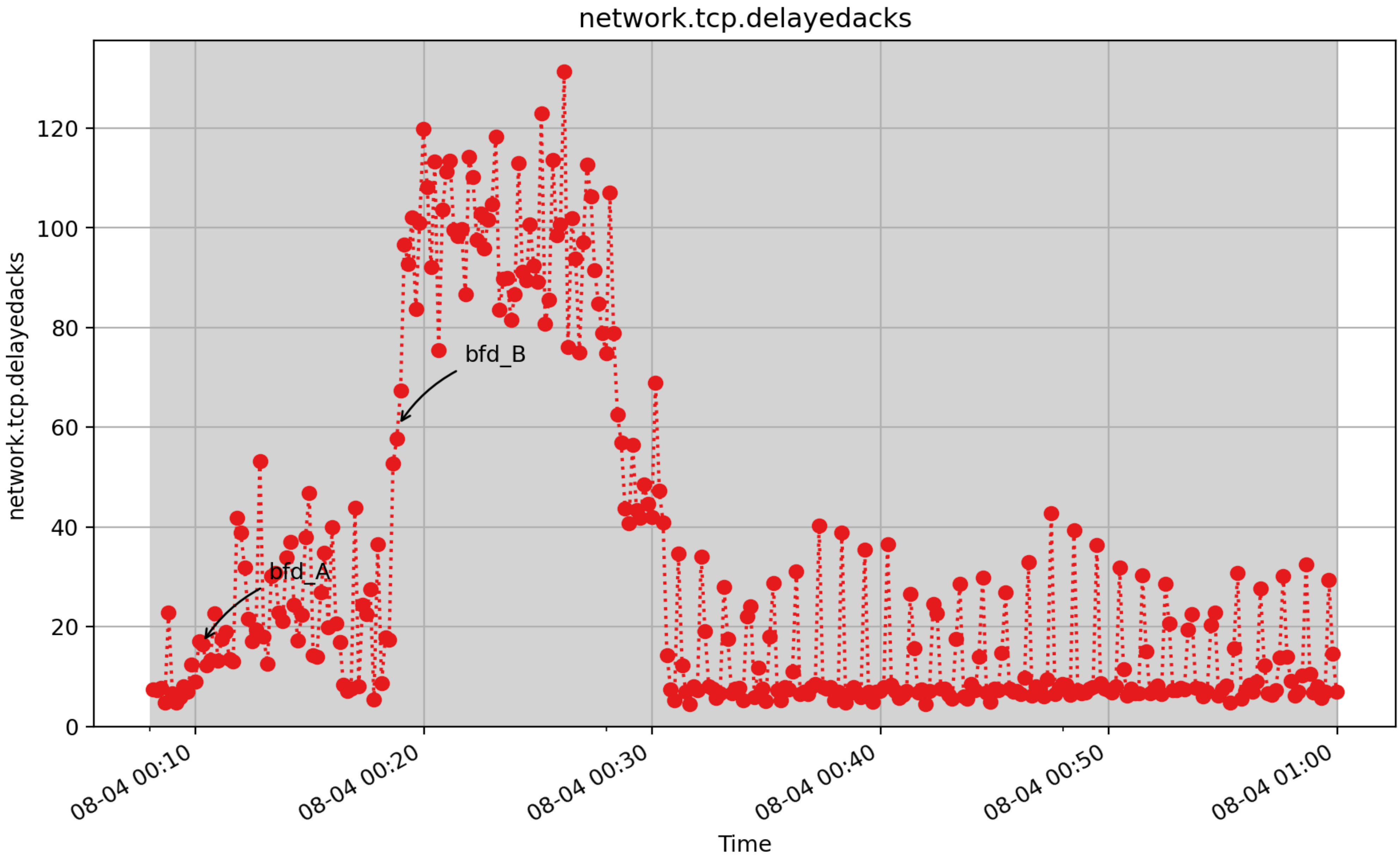
network.tcp.deferacceptdrop: Due to SYNACK retrans count lower than defer_accept value (count - U64) - *rate converted*



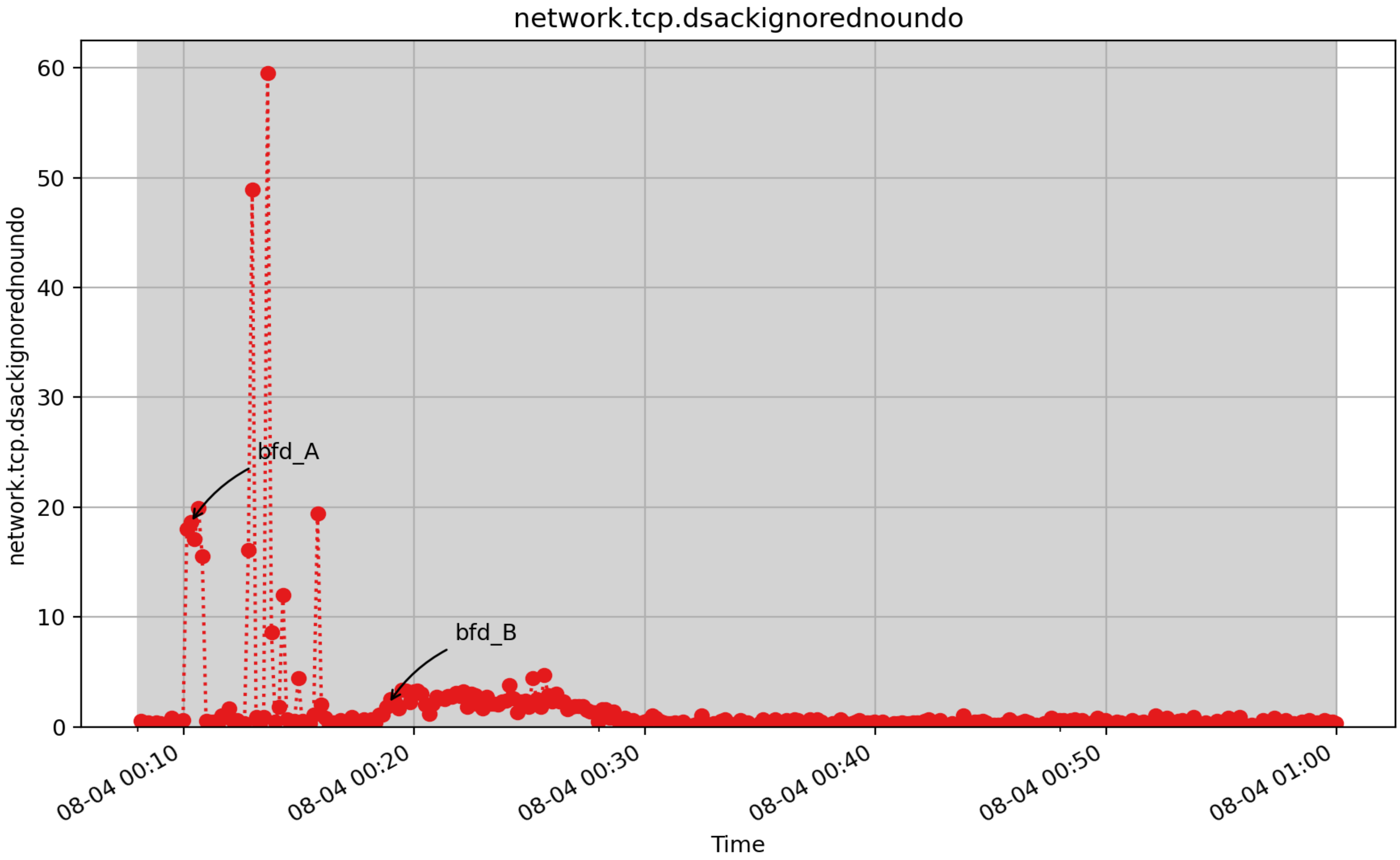
network.tcp.delayedacklocked: Number of delayed acks further delayed because of locked socket (count - U64) - *rate converted*



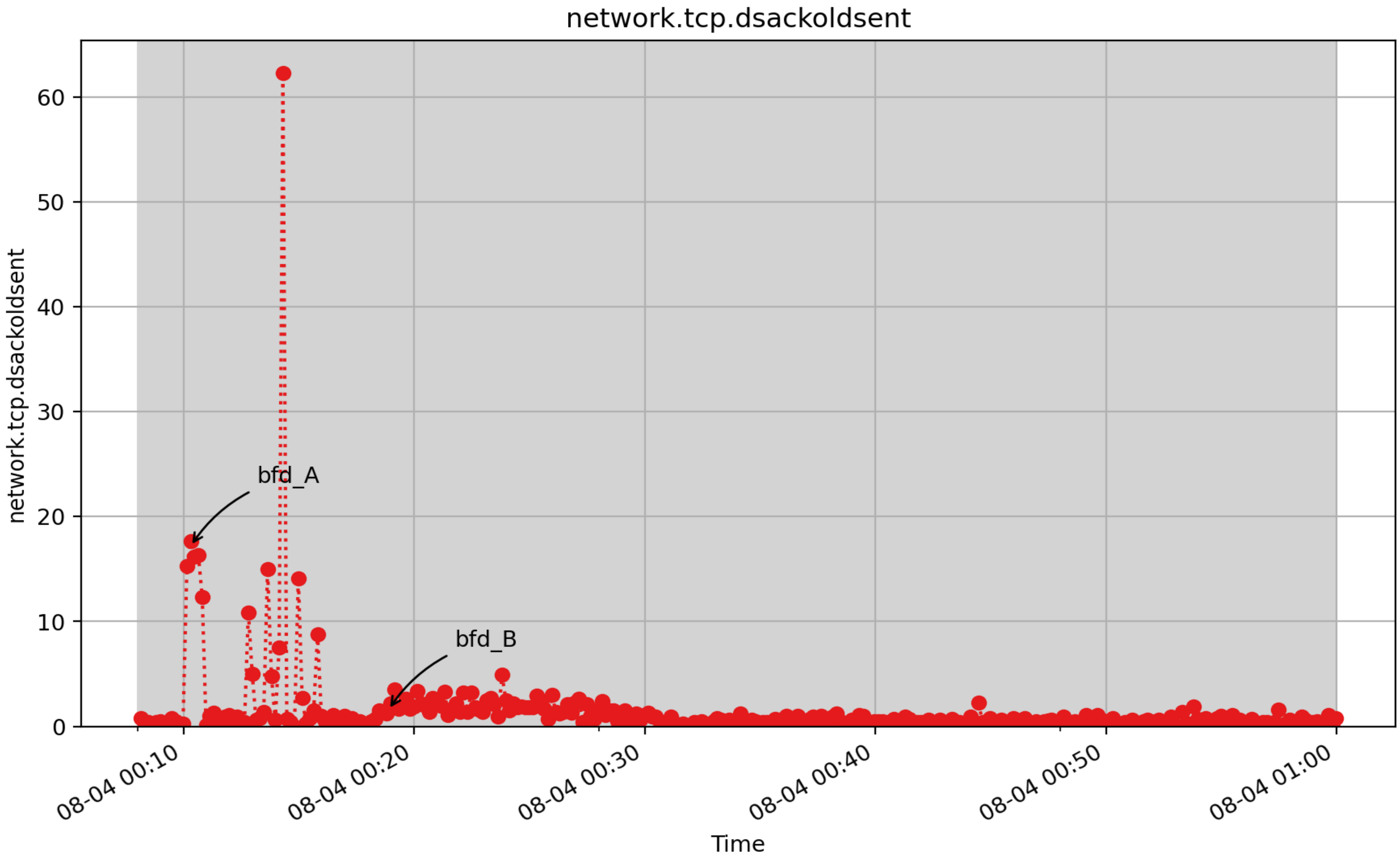
network.tcp.delayedacklost: Number of times quick ack mode was activated times (count - U64) - *rate converted*



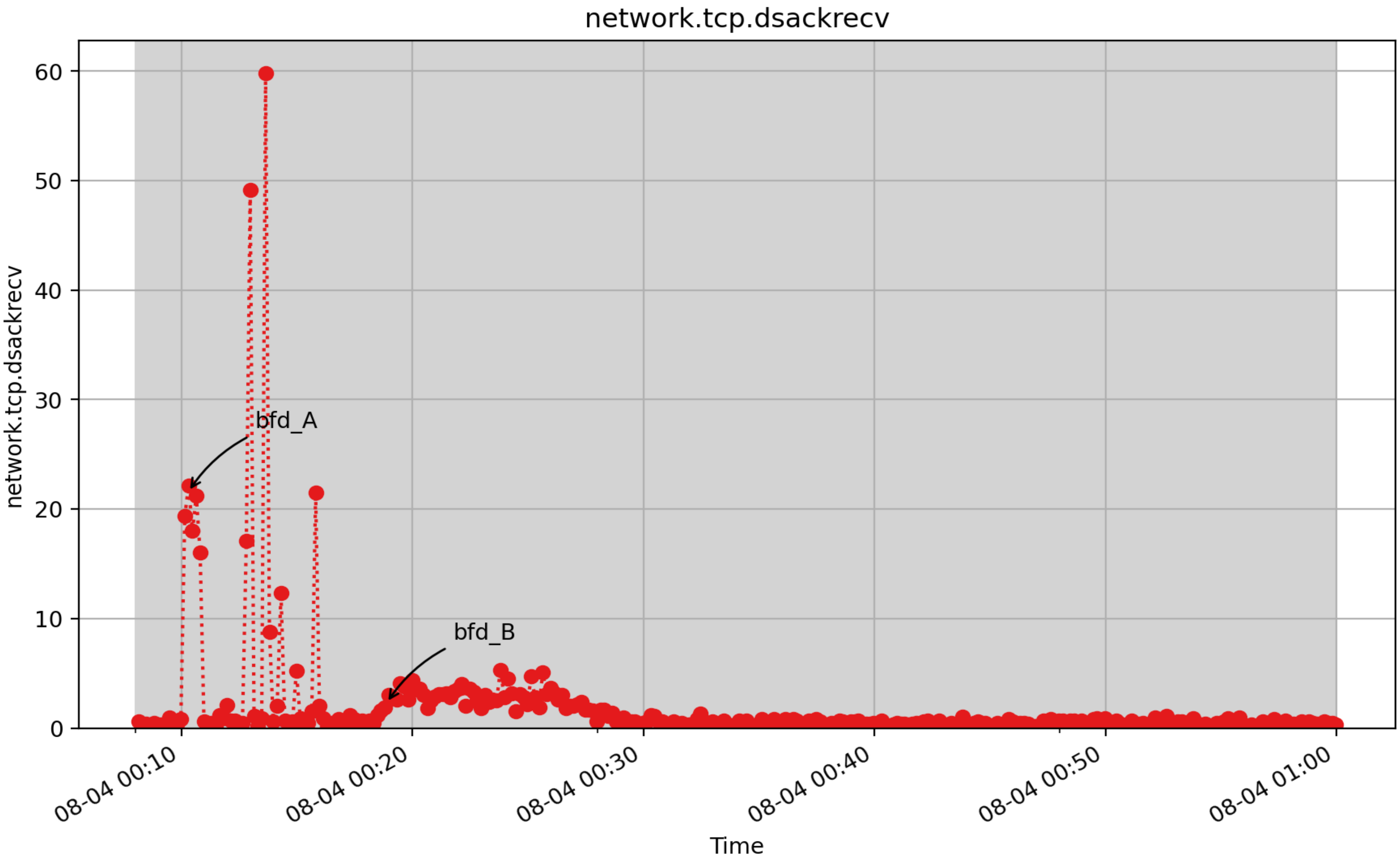
network.tcp.delayedacks: Number of delayed acks sent (count - U64) - *rate converted*



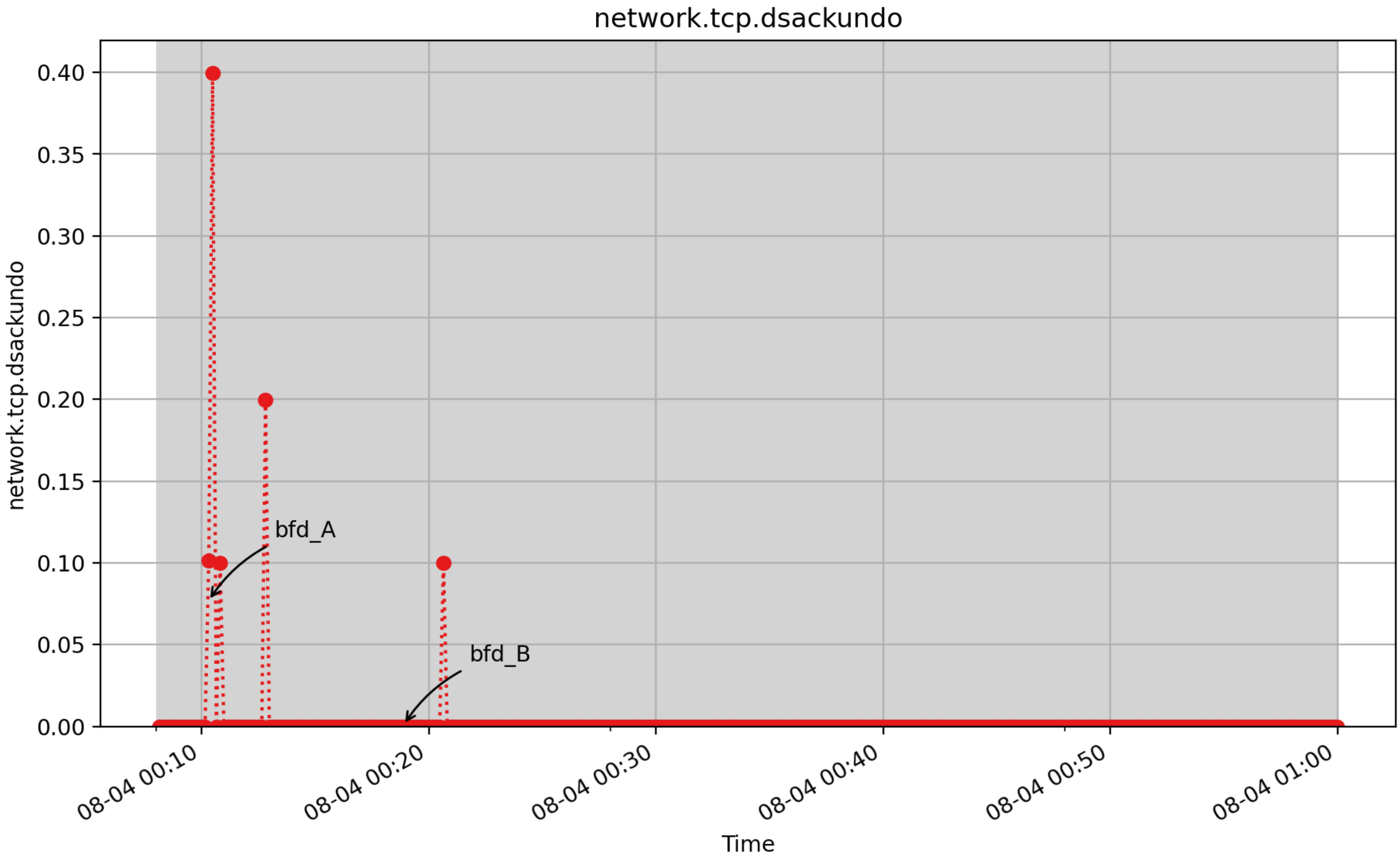
network.tcp.dsackignorednoundo: Number of ignored duplicate SACKs with undo_marker not set (count - U64) - *rate converted*



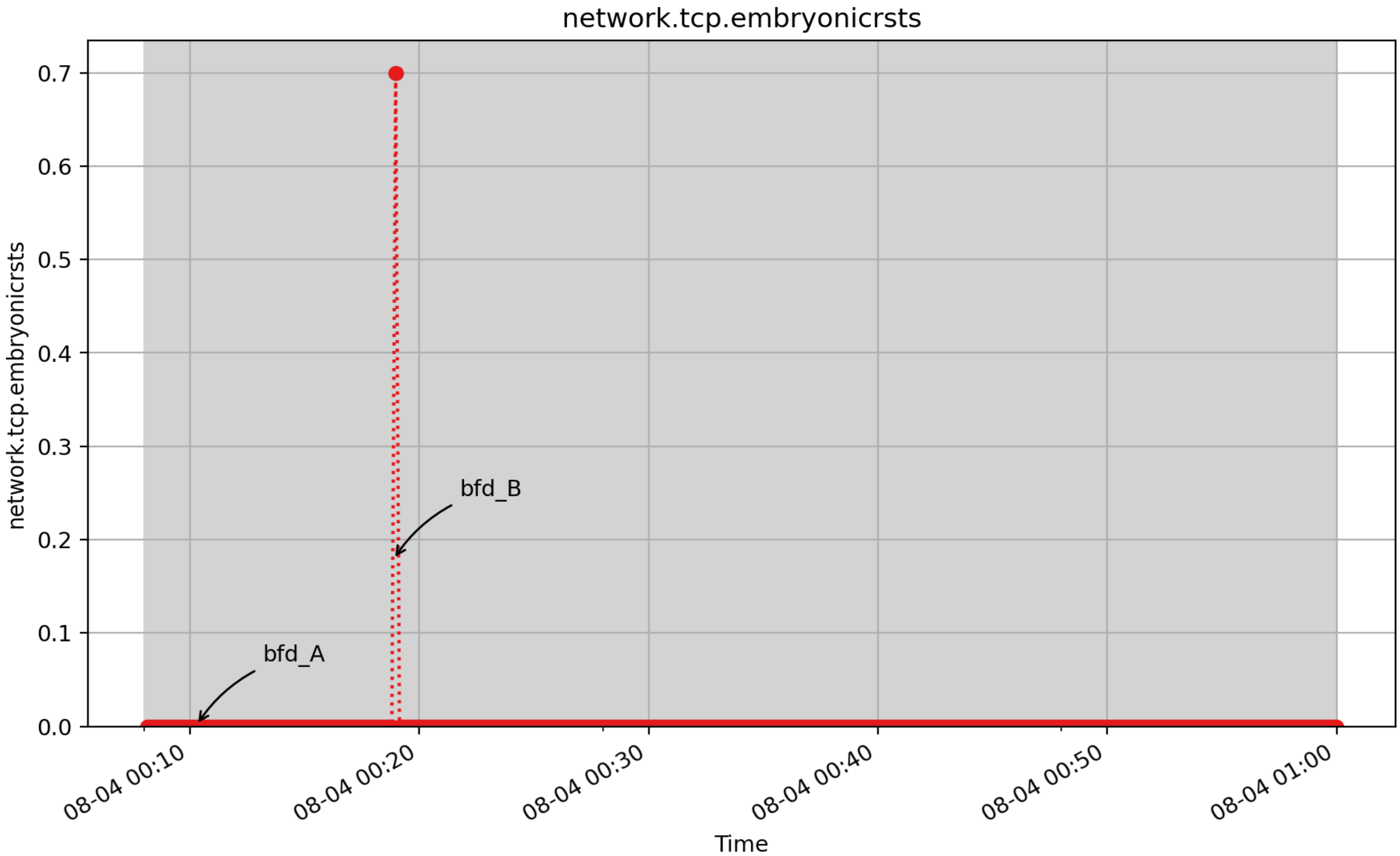
network.tcp.dsackoldsent: Number of DSACKs sent for old packets (count - U64) - *rate converted*



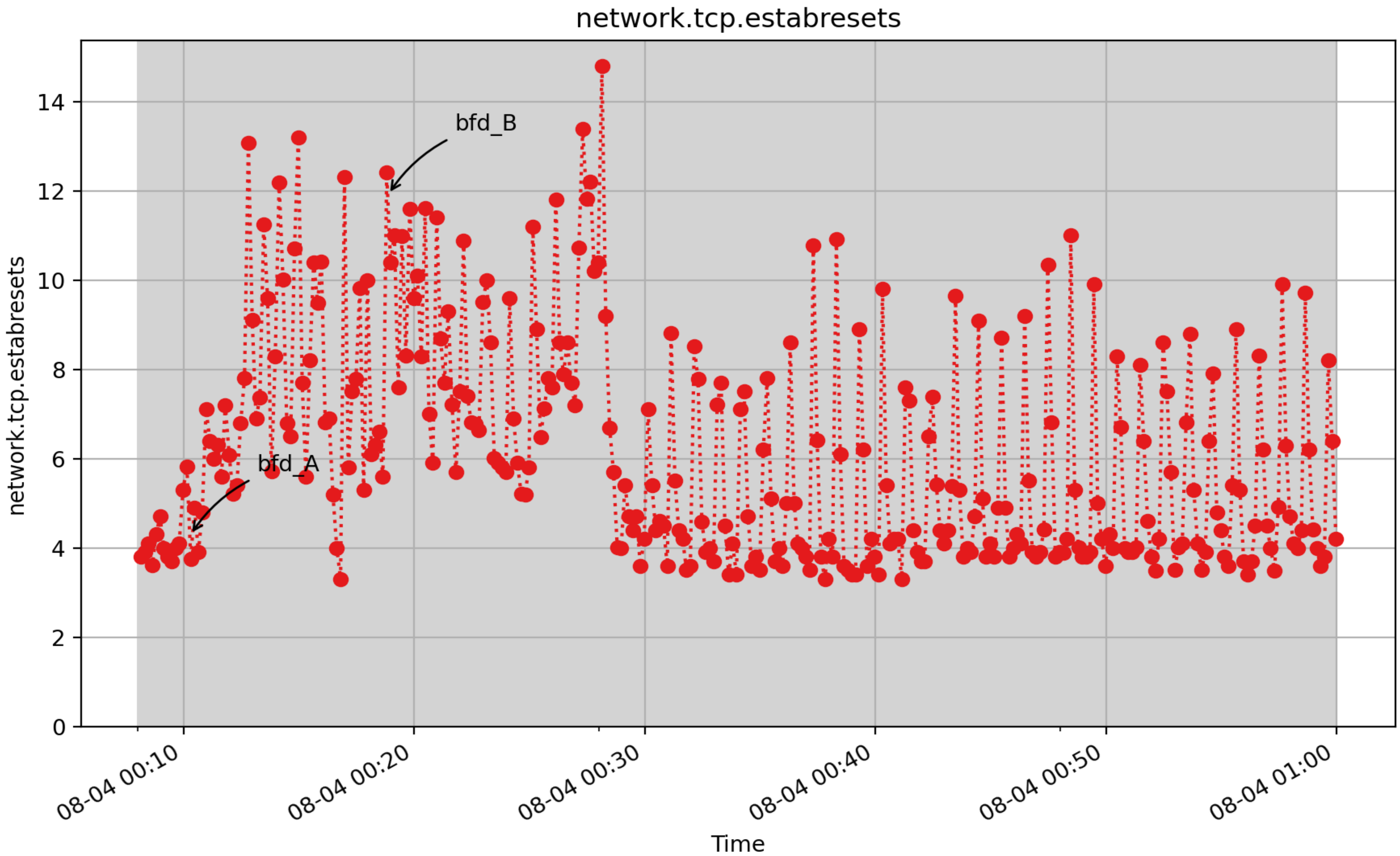
network.tcp.dsackrecv: Number of DSACKs received (count - U64) - *rate converted*



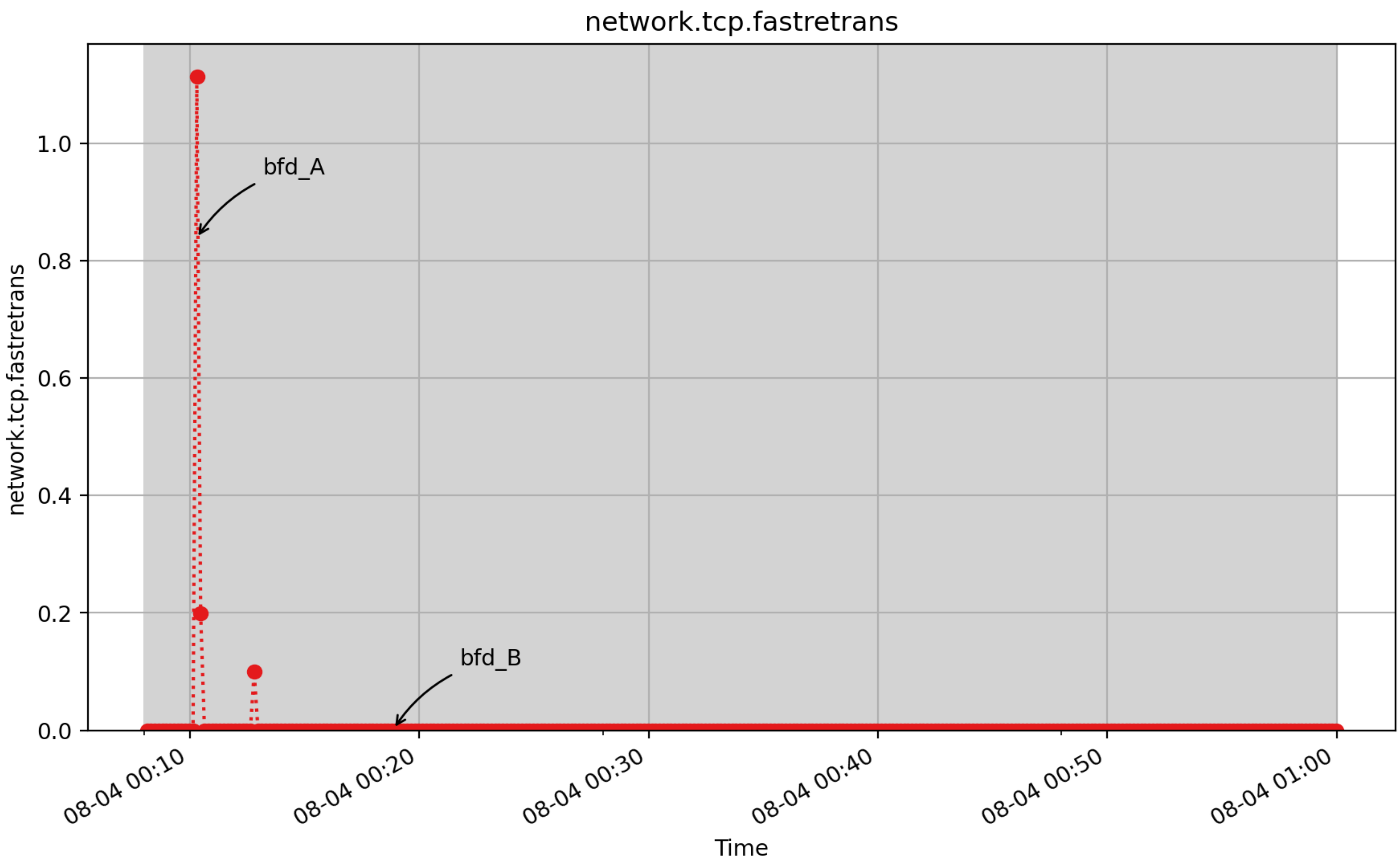
network.tcp.dsackundo: Number of congestion windows recovered without slow start using DSACK (count - U64) - *rate converted*



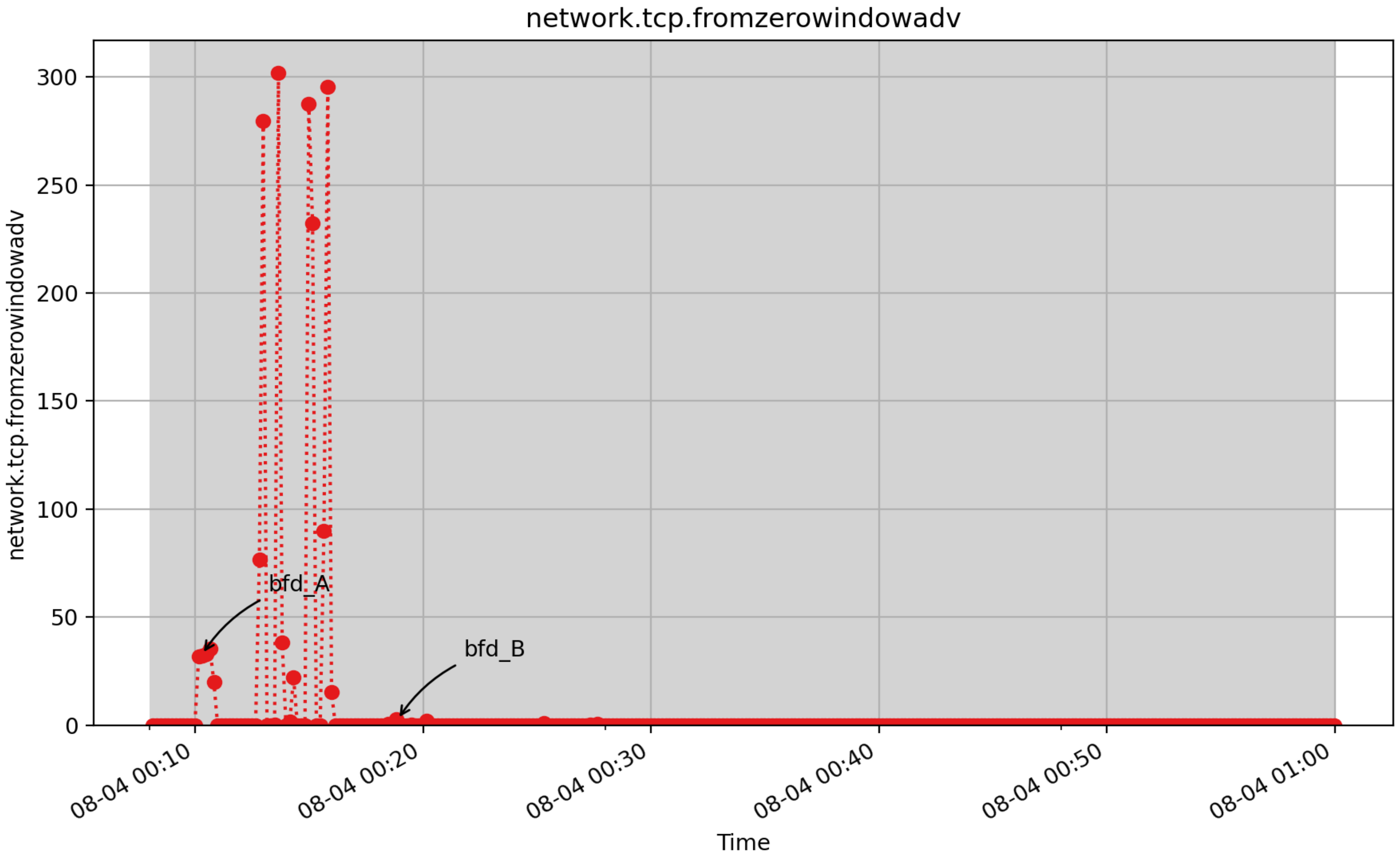
network.tcp.embryonicrsts: Number of resets received for embryonic SYN_RECV sockets (count - U64) - *rate converted*



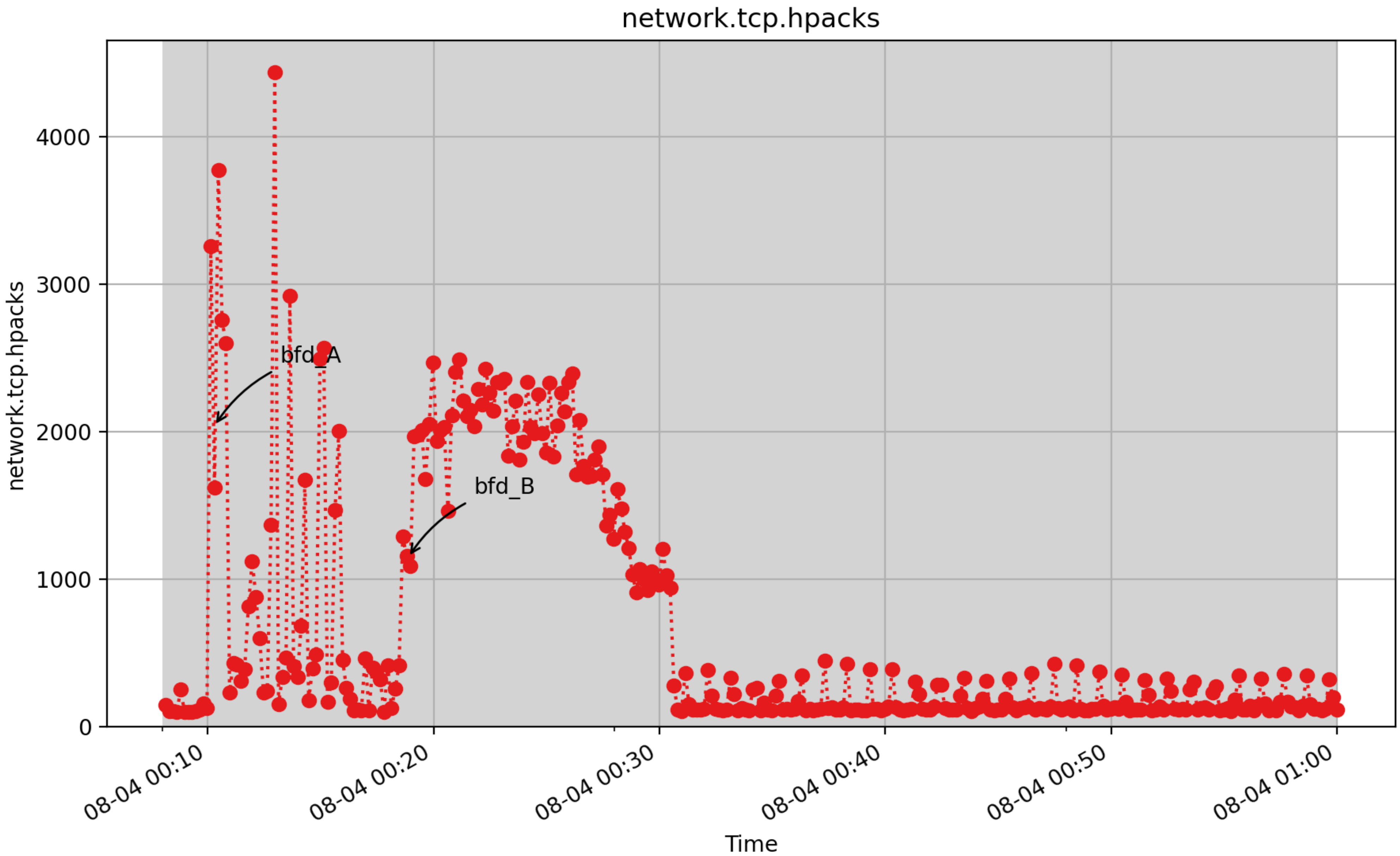
network.tcp.estabresets: count of tcp estabresets (count - U64) - *rate converted*



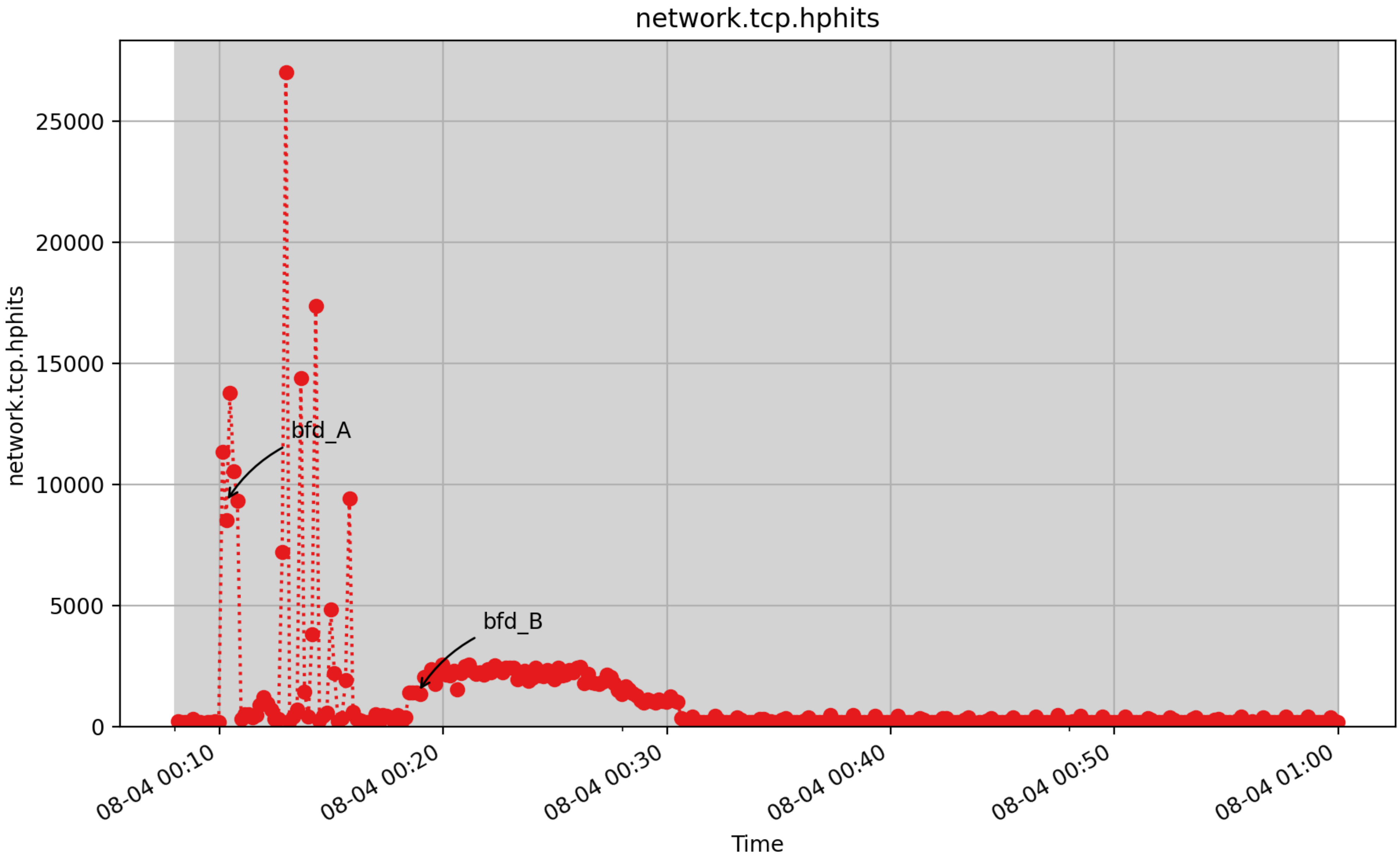
network.tcp.fastretrans: Number of fast retransmits (count - U64) - *rate converted*



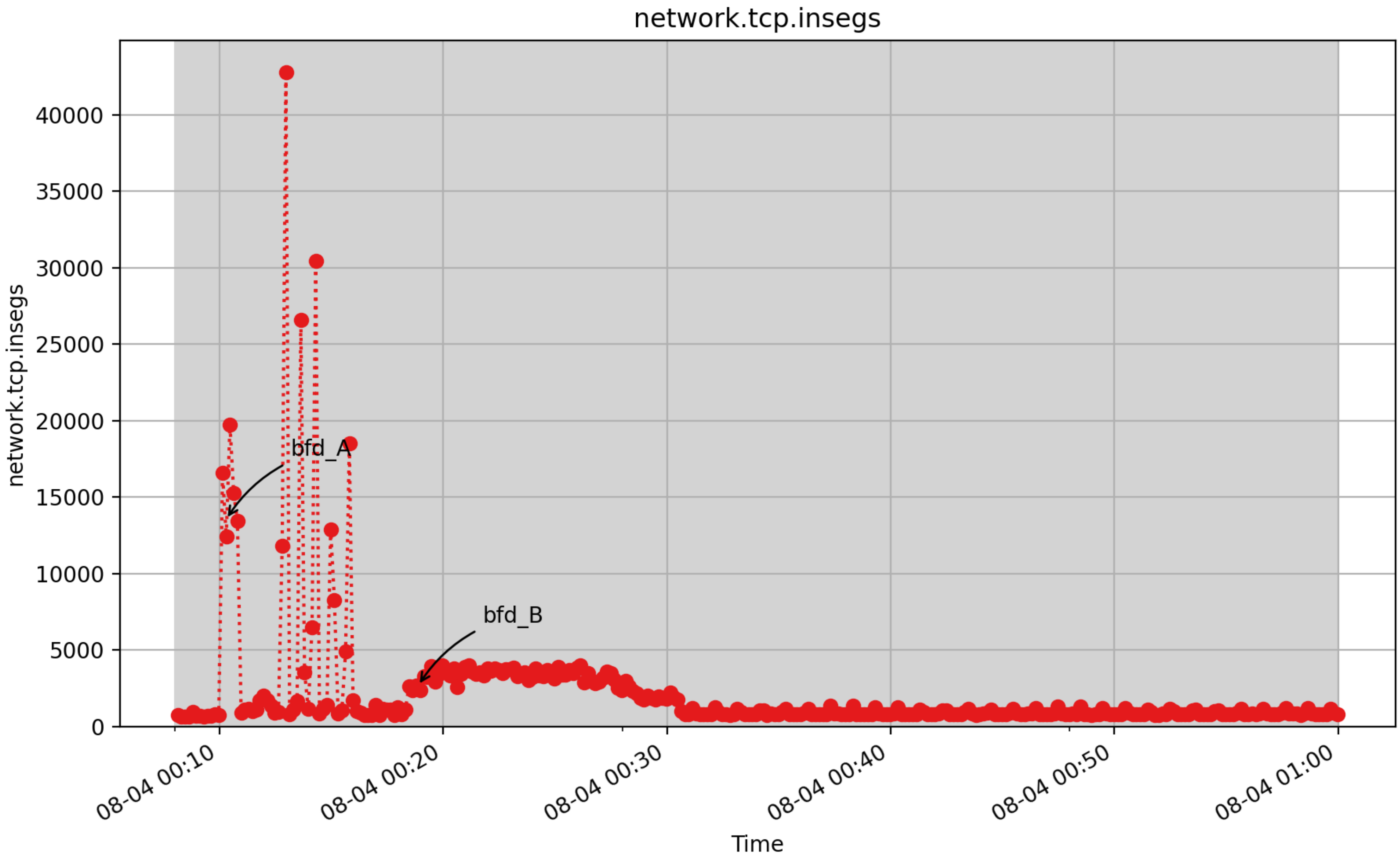
network.tcp.fromzerowindowadv: Number of times window went from zero to non-zero (count - U64) - *rate converted*



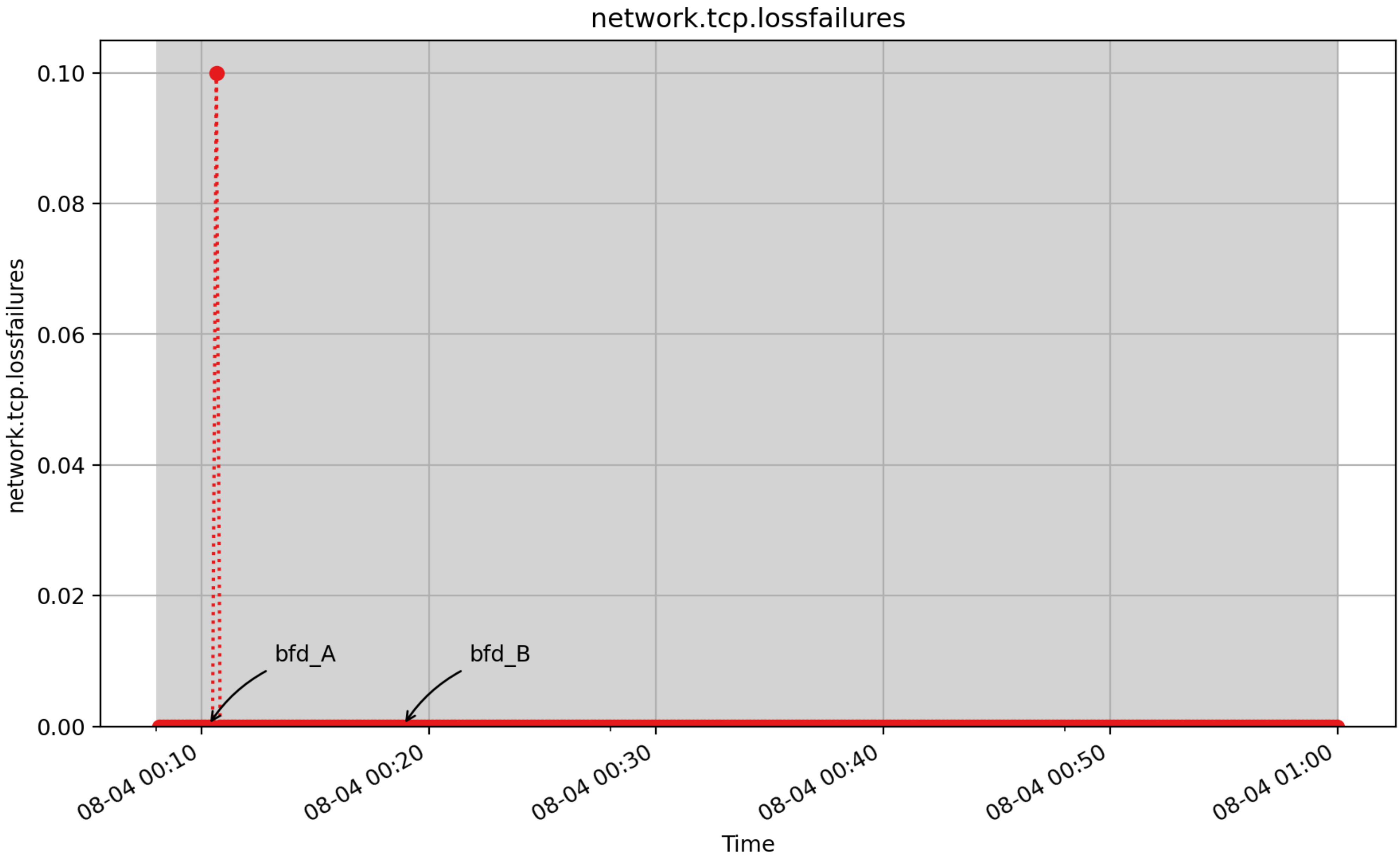
network.tcp.hpacks: Number of predicted acknowledgments (count - U64) - *rate converted*



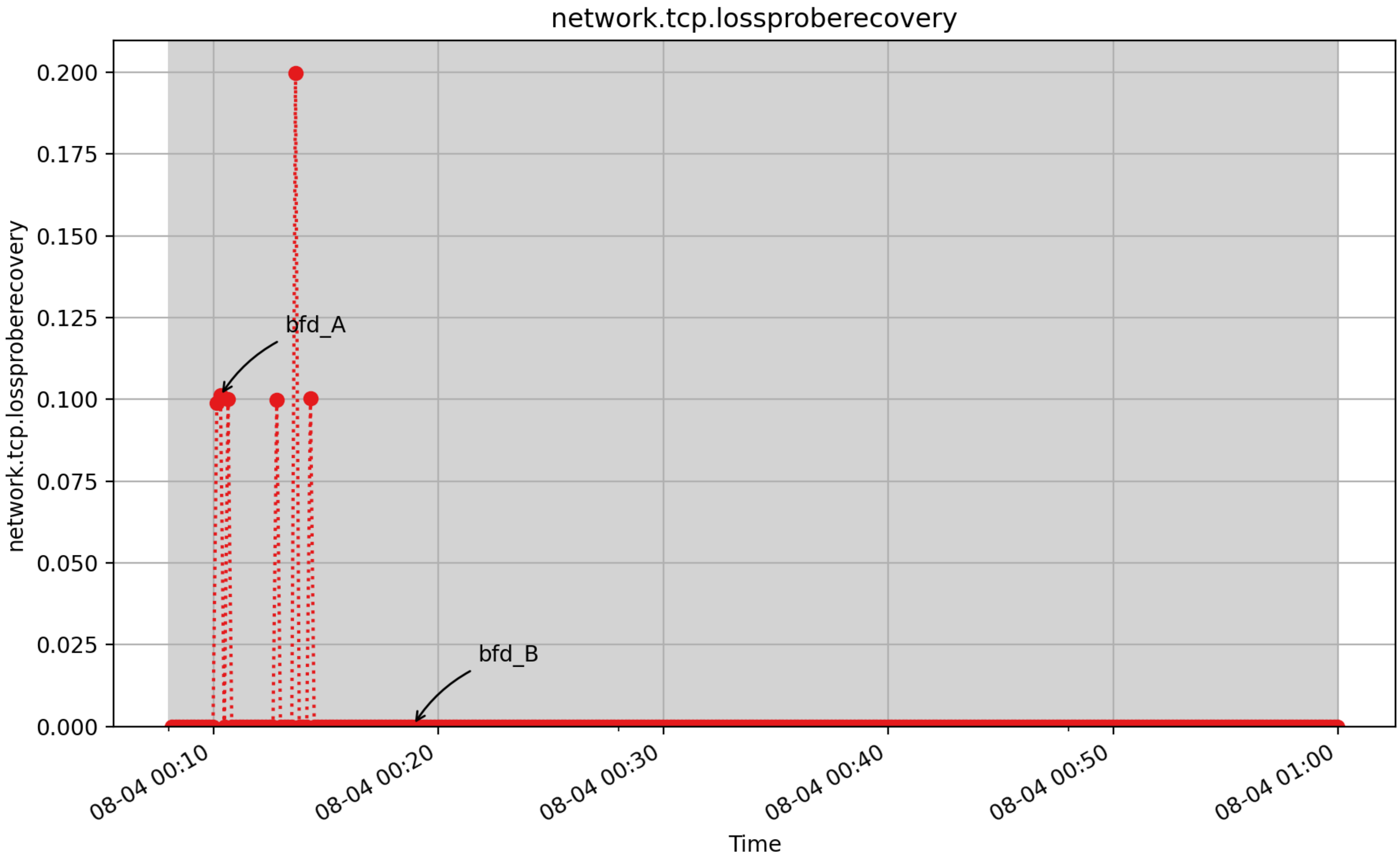
network.tcp.hphits: Number of packet headers predicted (count - U64) - *rate converted*



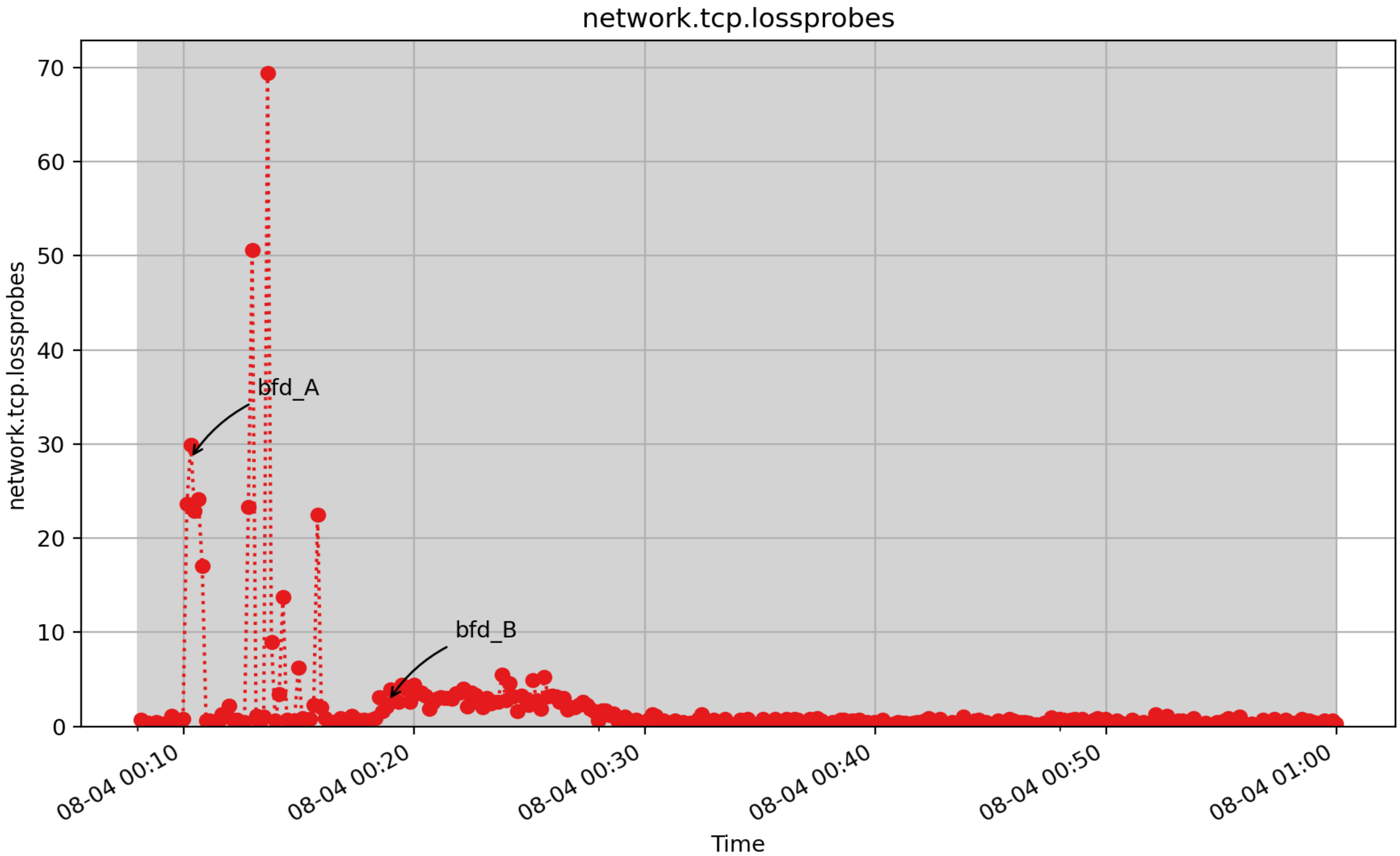
network.tcp.insegs: count of tcp segments received (count - U64) - *rate converted*



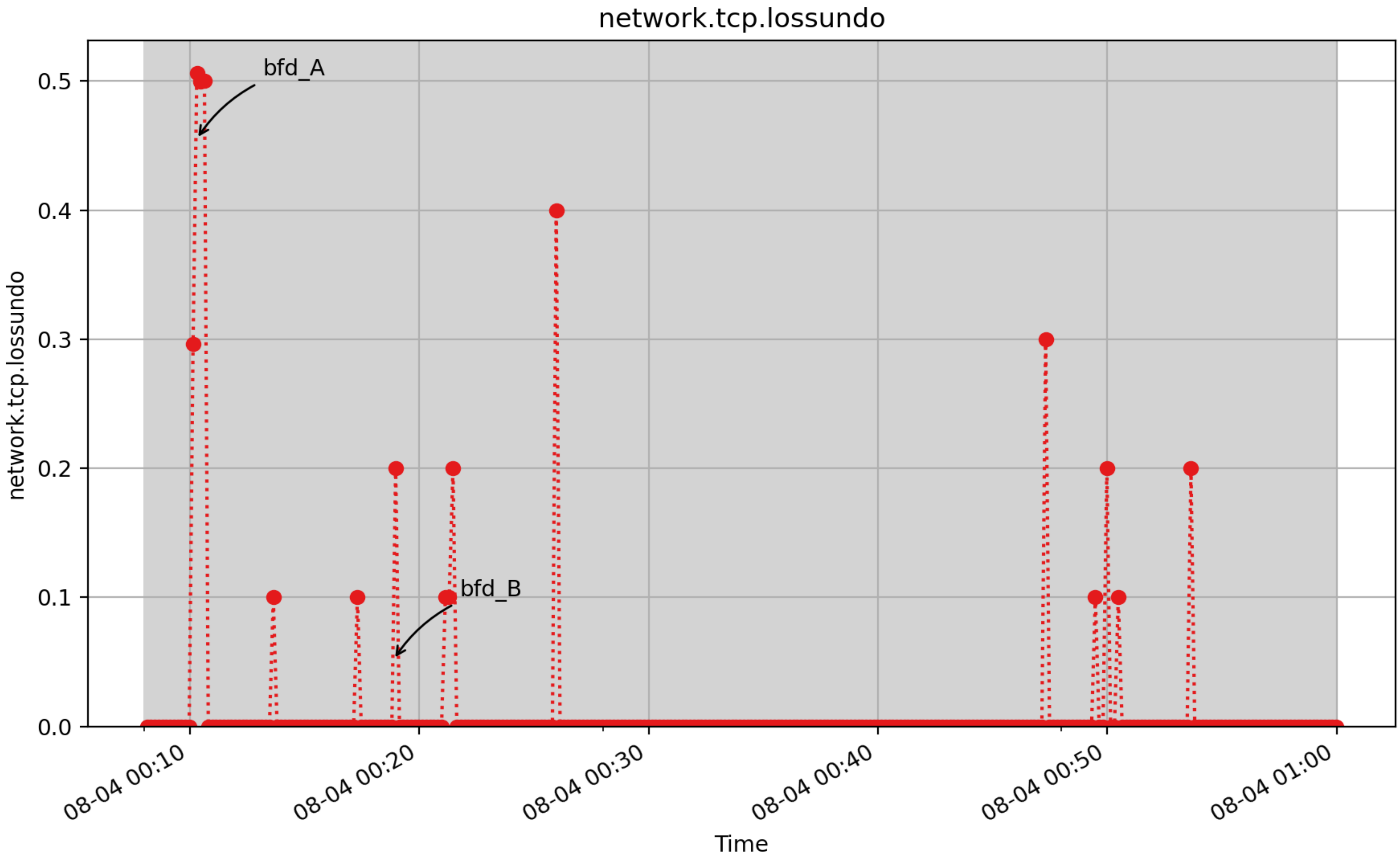
network.tcp.lossfailures: Number of timeouts in loss state (count - U64) - *rate converted*



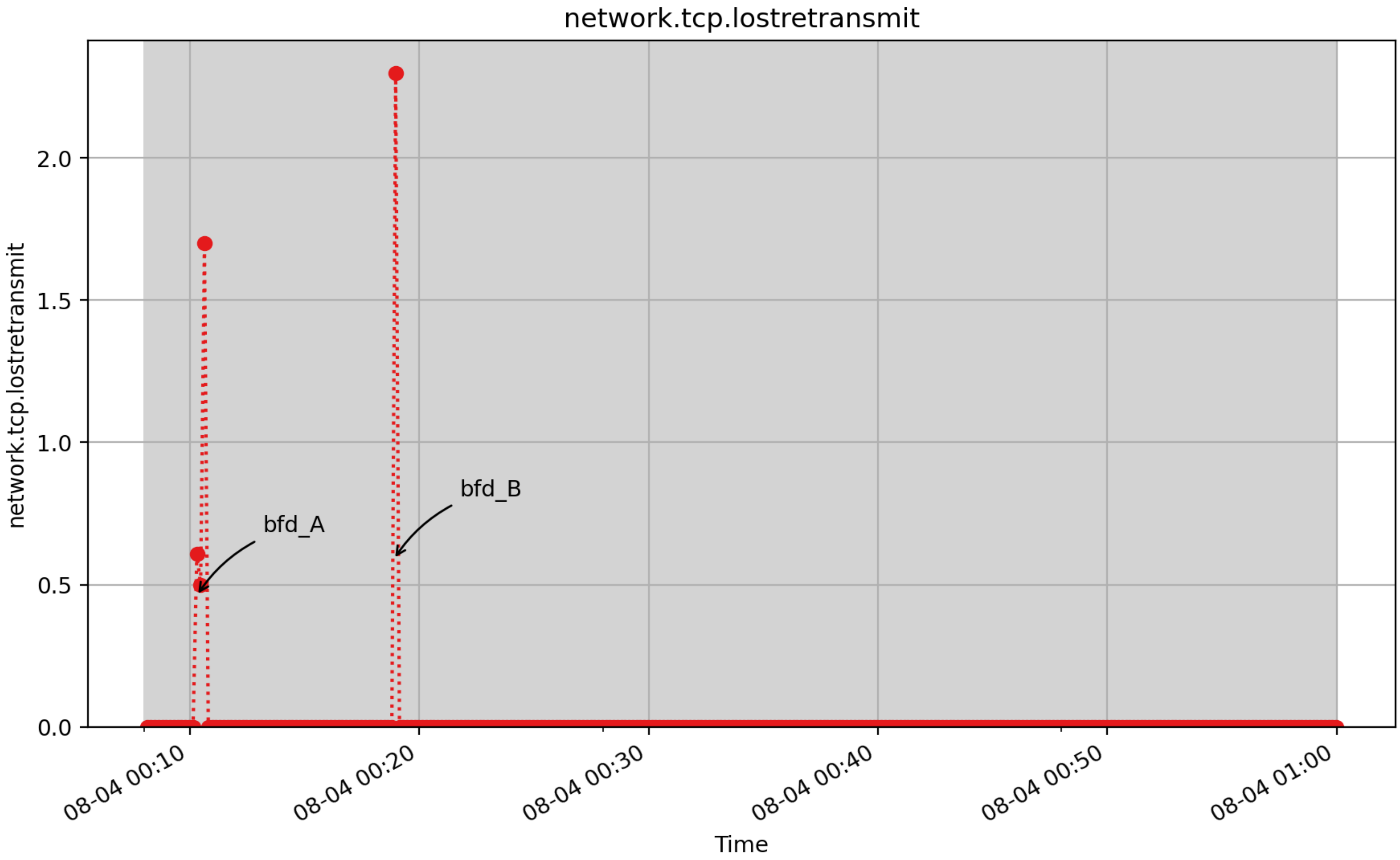
network.tcp.lossproberecovery: Number of TCP loss probe recoveries (count - U64) - *rate converted*



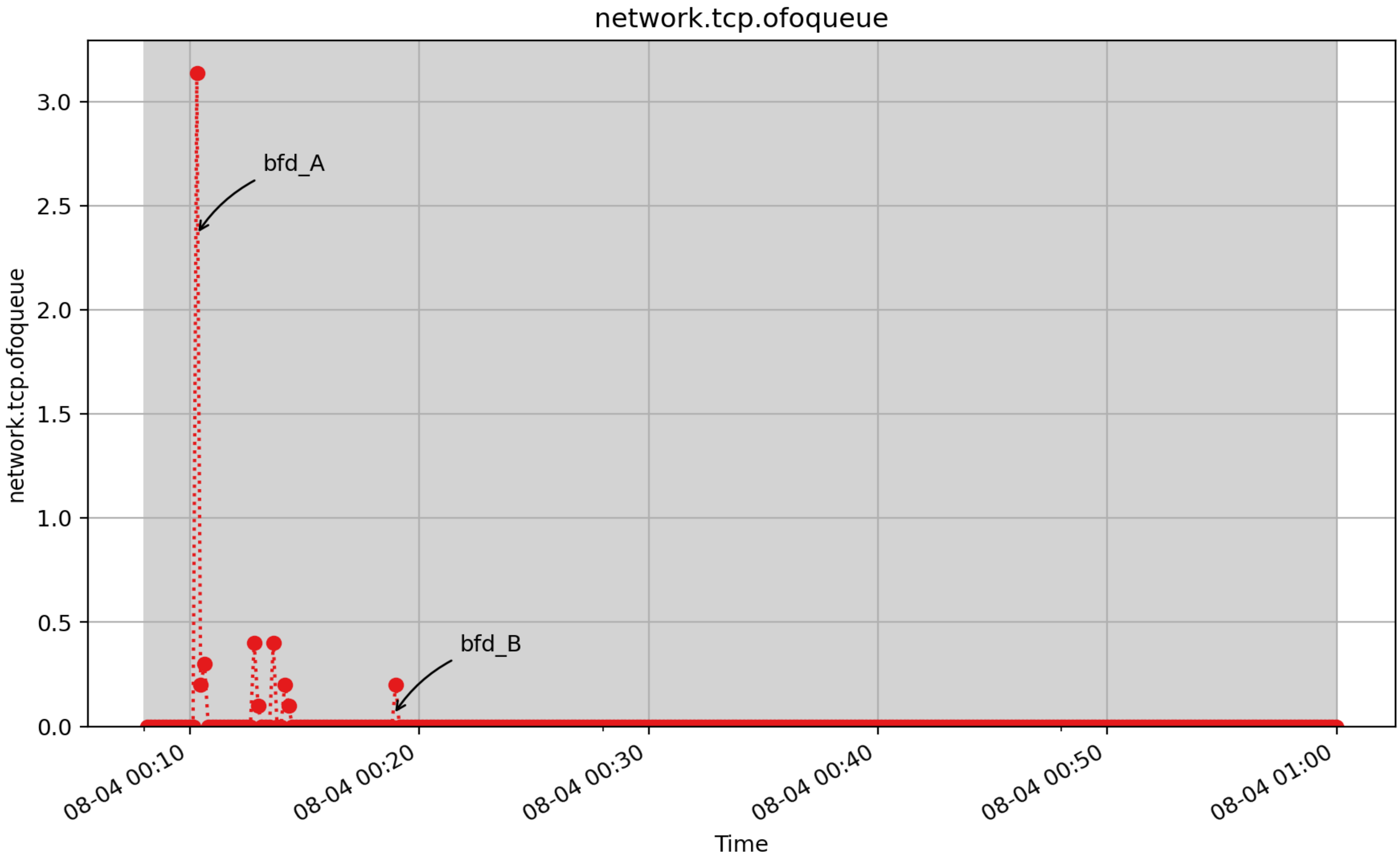
network.tcp.lossprobes: Number of sent TCP loss probes (count - U64) - *rate converted*



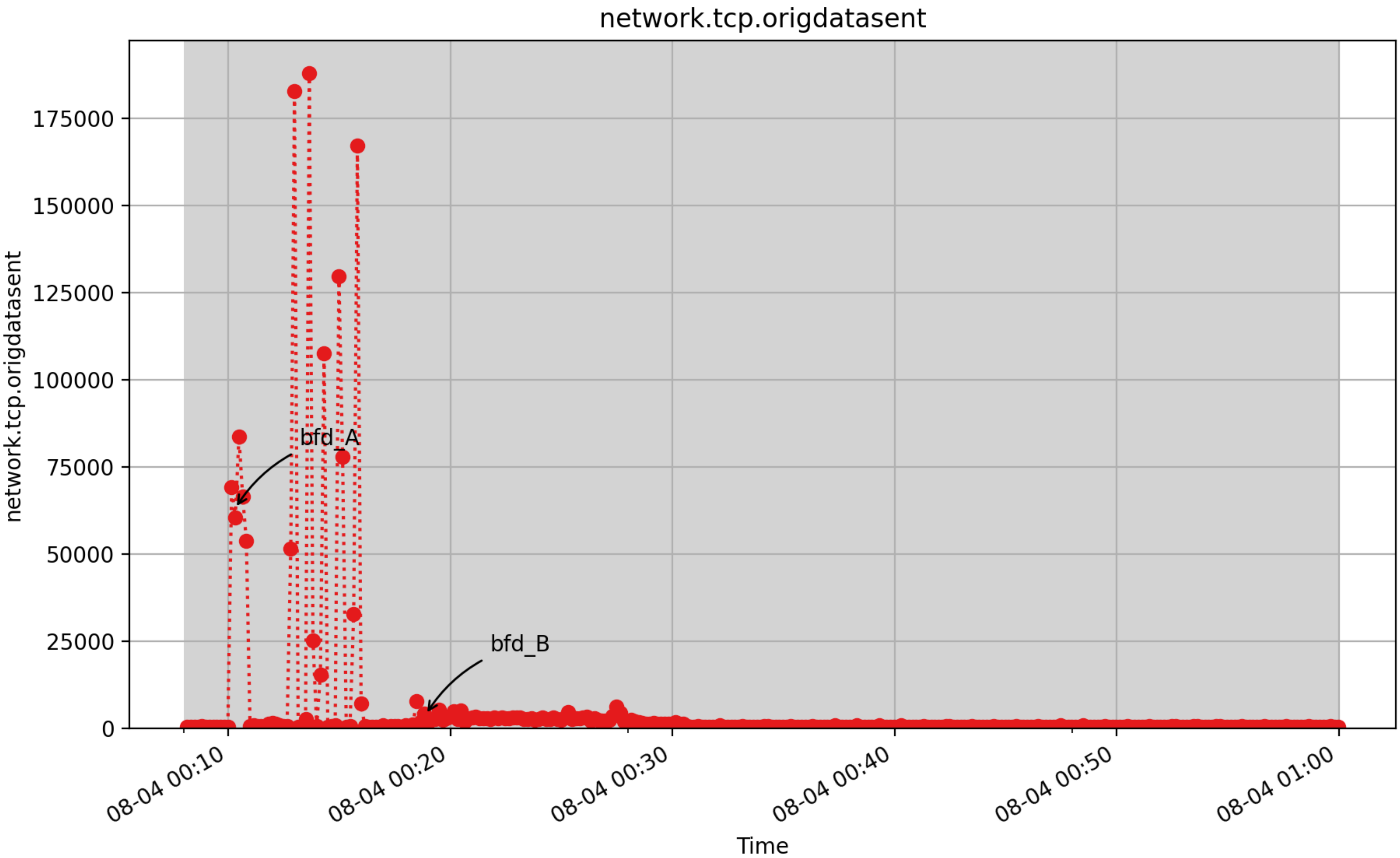
network.tcp.lossundo: Number of congestion windows recovered without slow start after partial ack (count - U64) - *rate converted*



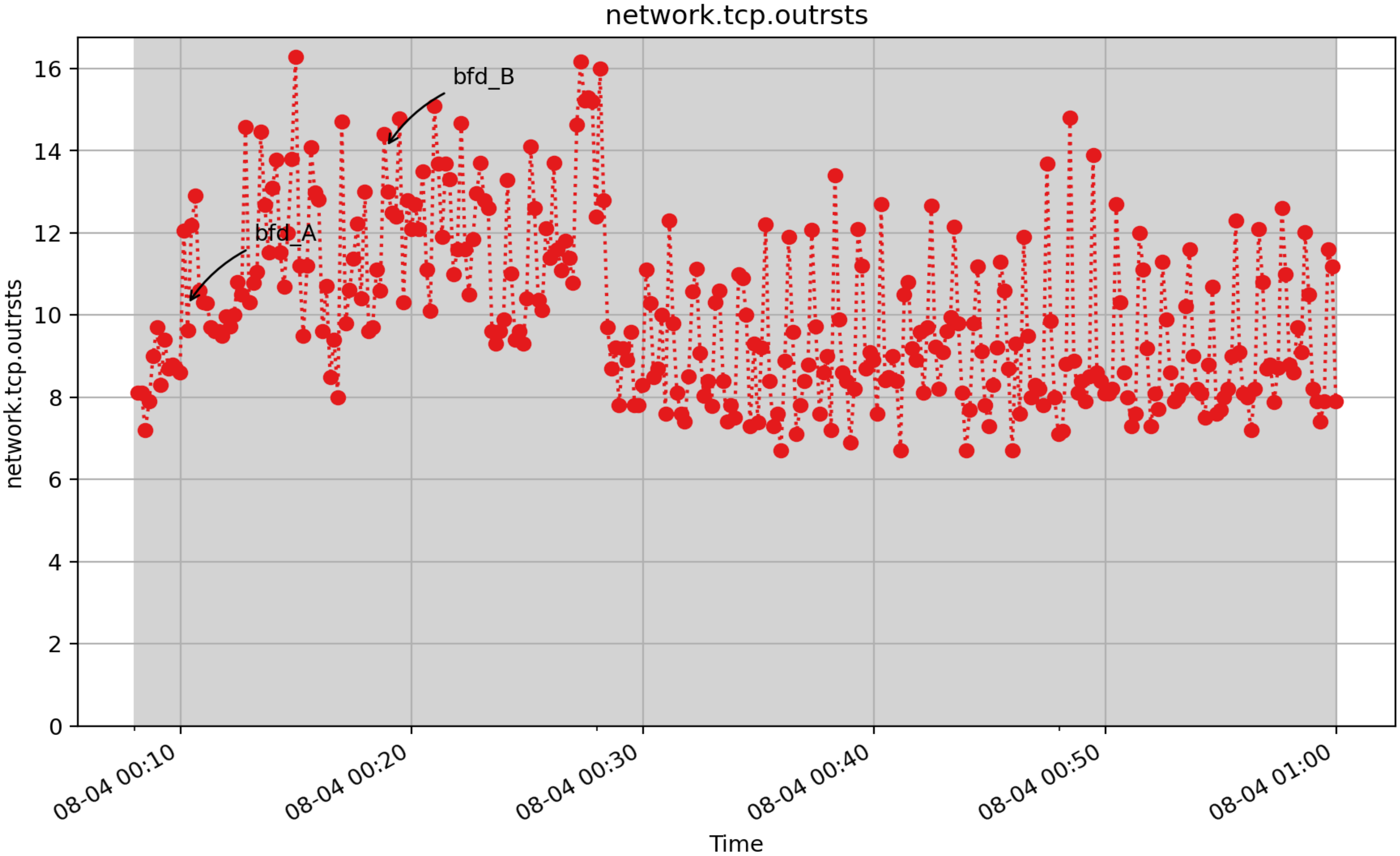
network.tcp.lostretransmit: Number of retransmits lost (count - U64) - *rate converted*



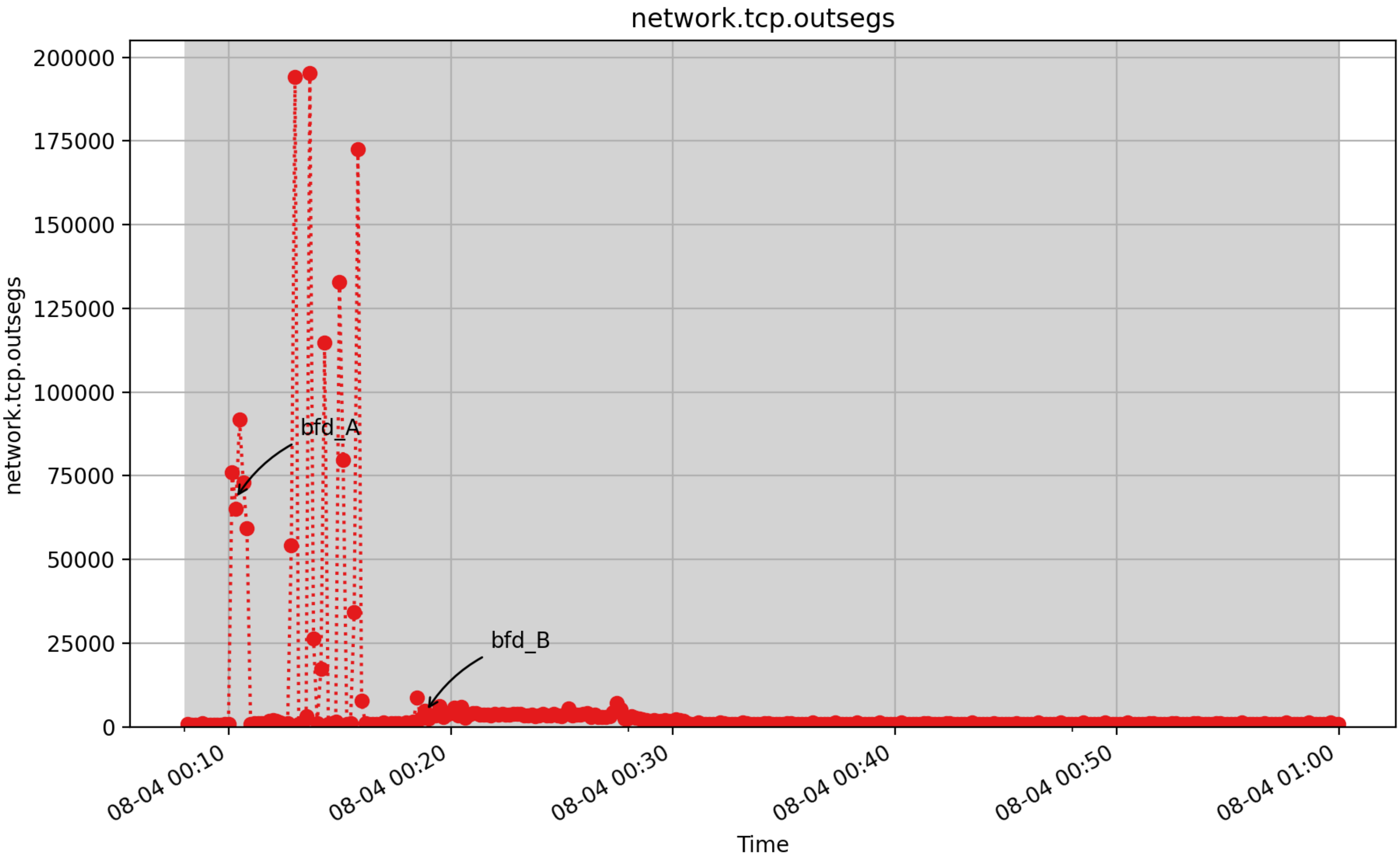
network.tcp.ofoqueue: Number of packets queued in OFO queue (count - U64) - *rate converted*



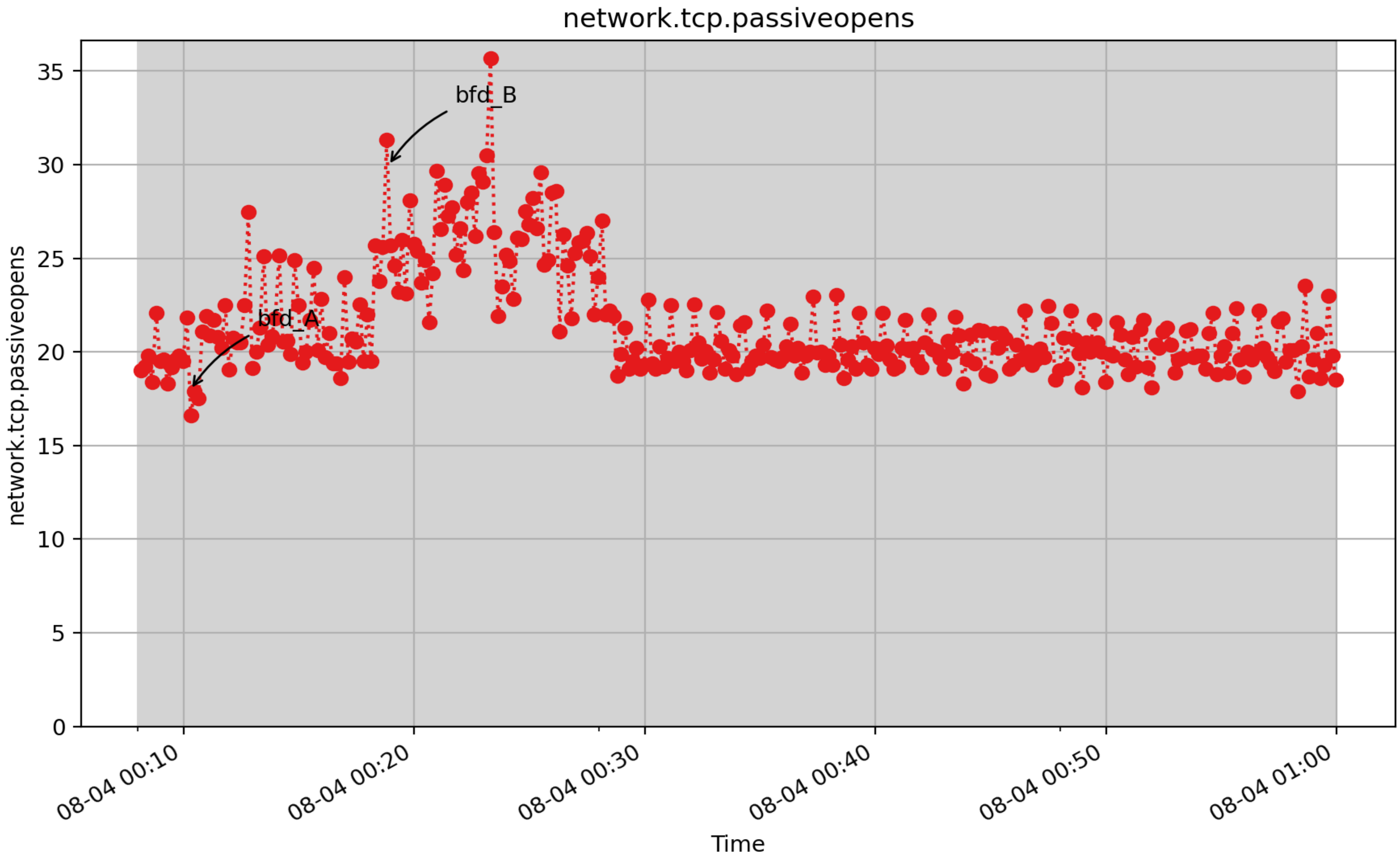
network.tcp.origdatasent: Excluding retransmission but including data-in-SYN). This counter is different from TcpOutSegs because TcpOutSegs also tracks pure ACKs. TCPOrigDataSent is more useful to track the TCP retransmission rate. (count - U64) - *rate converted*



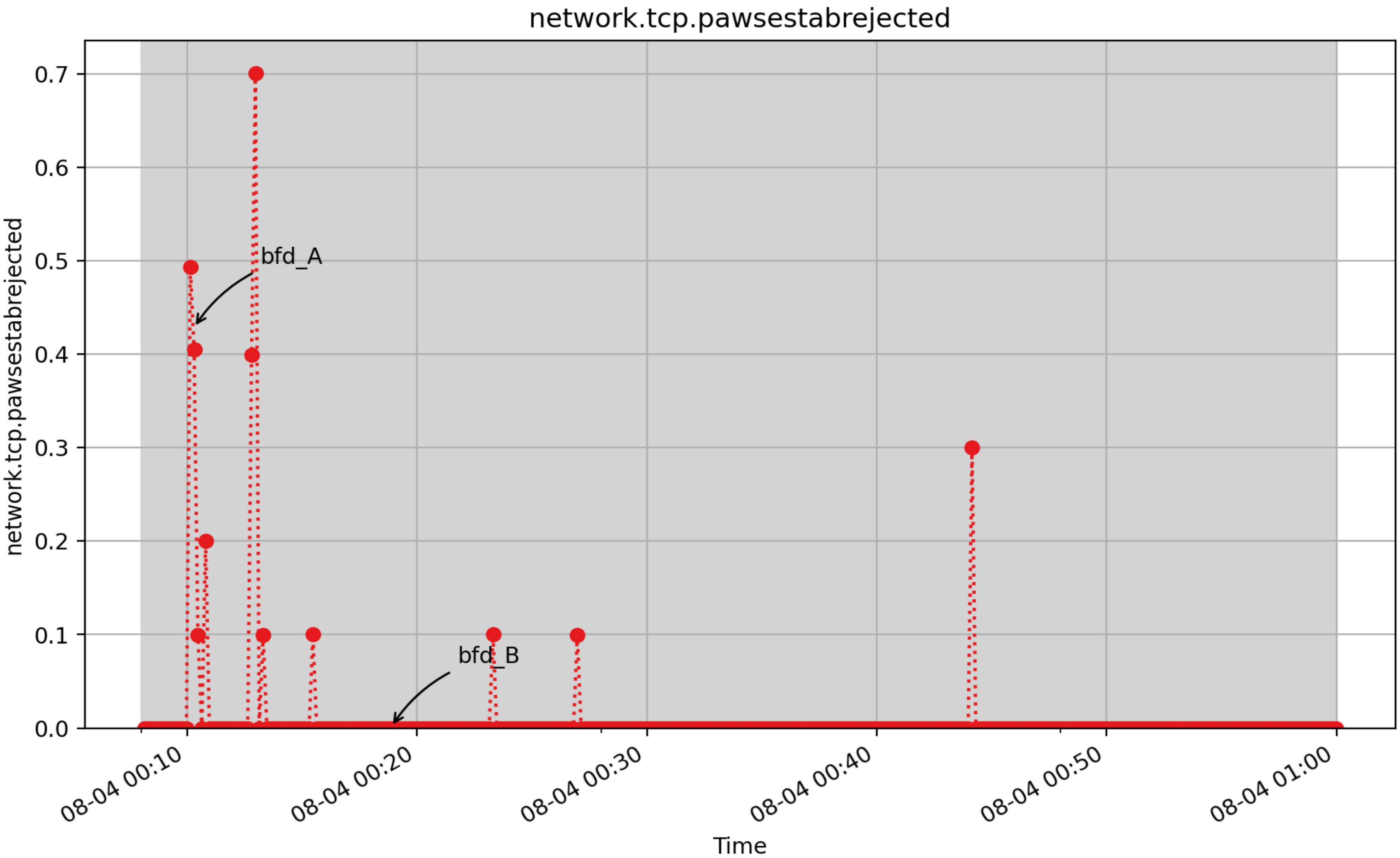
network.tcp.outrsts: count of tcp segments sent with RST flag (count - U64) - *rate converted*



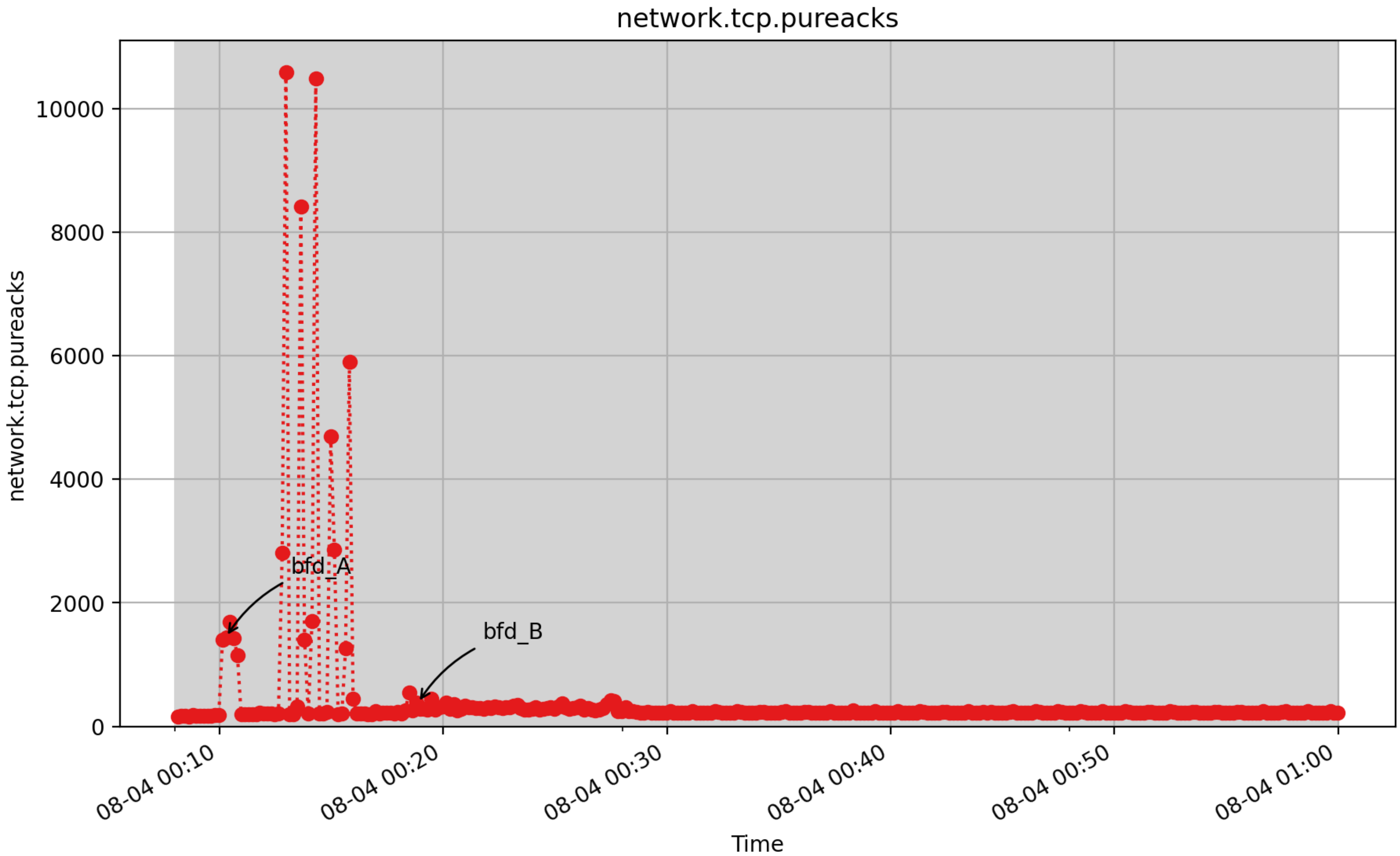
network.tcp.outsegs: count of tcp segments sent (count - U64) - *rate converted*



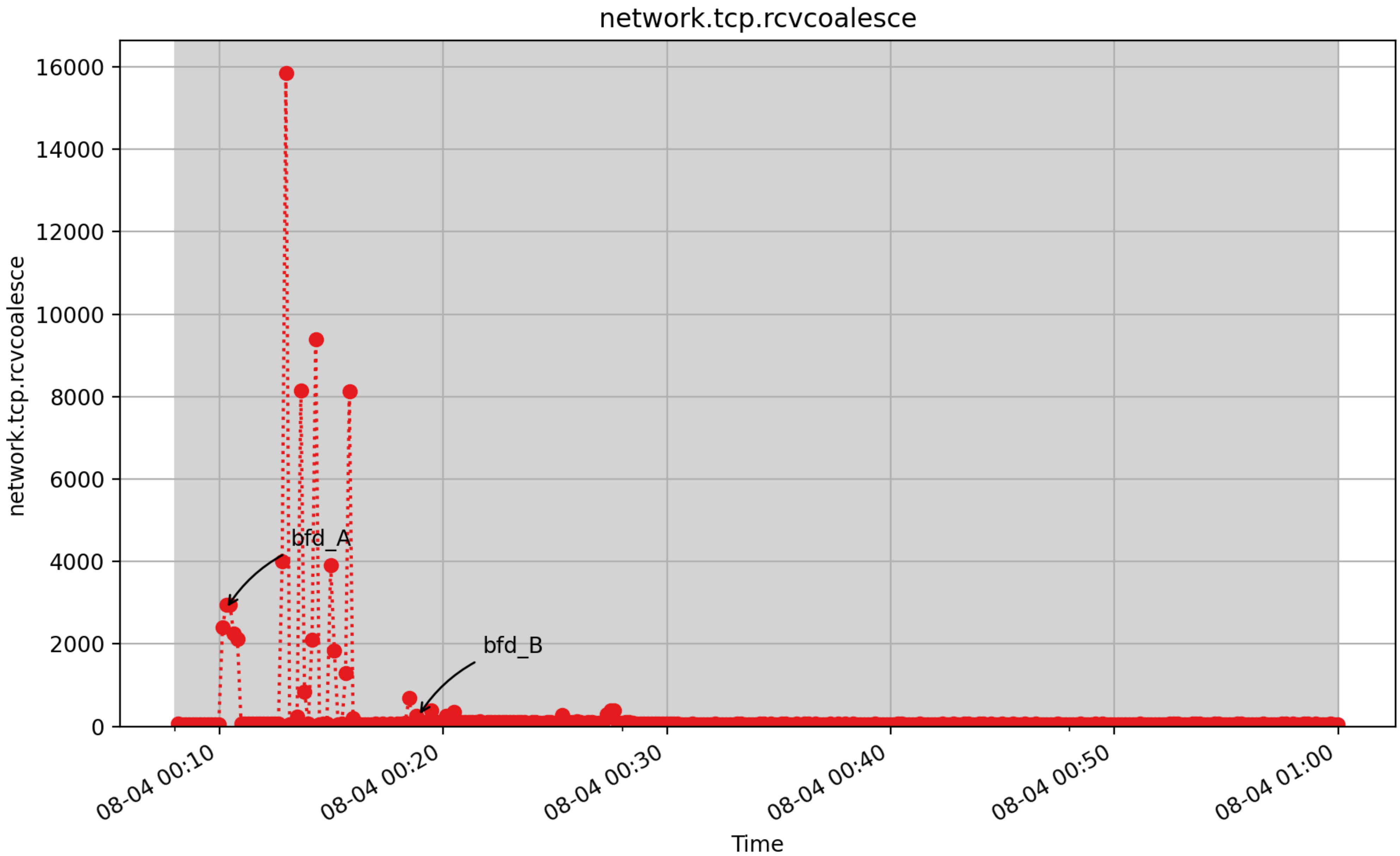
network.tcp.passiveopens: count of tcp passiveopens (count - U64) - *rate converted*



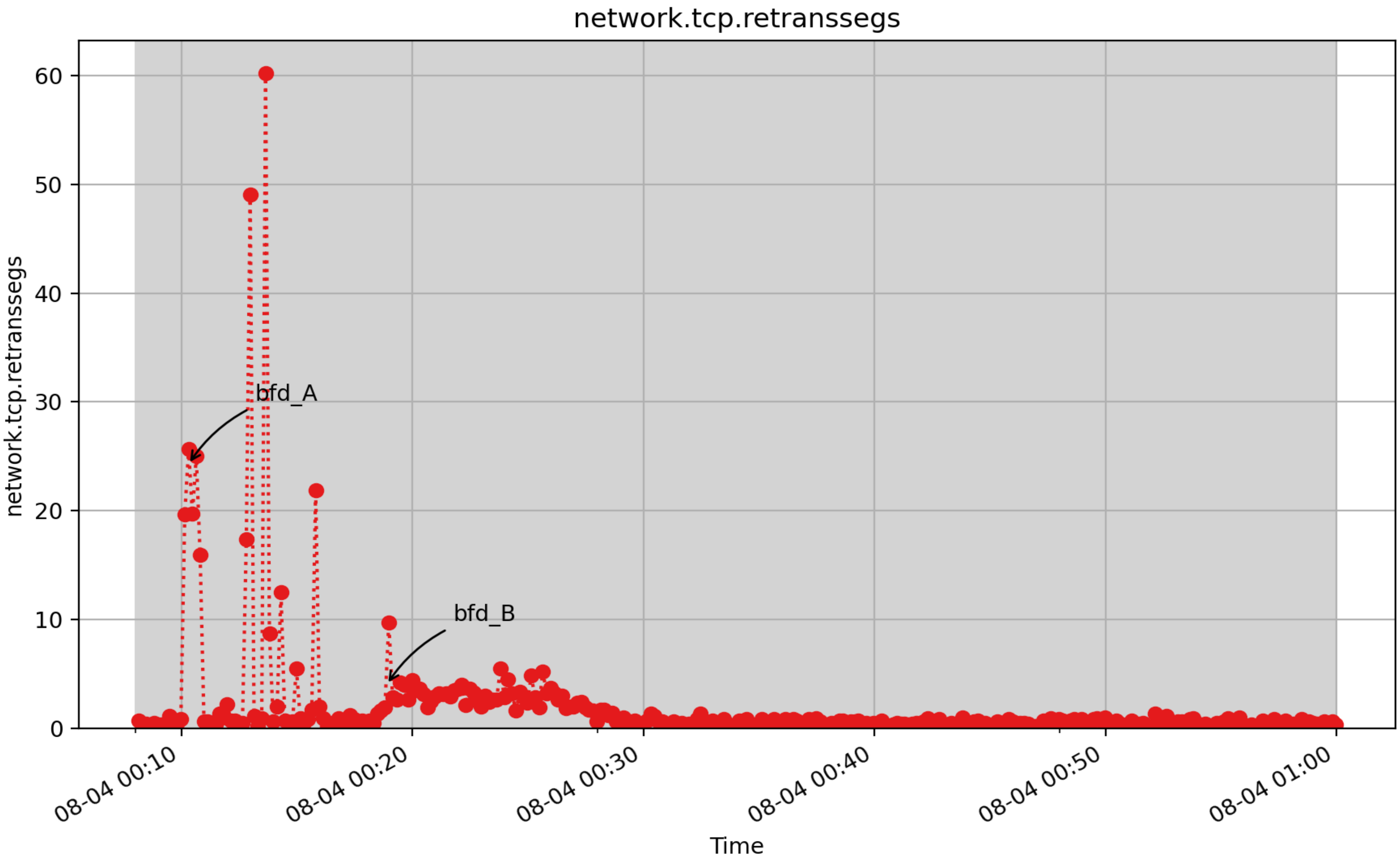
network.tcp.pawsestabrejected: Number of packets rejects in established connections because of timestamp (count - U64) - *rate converted*



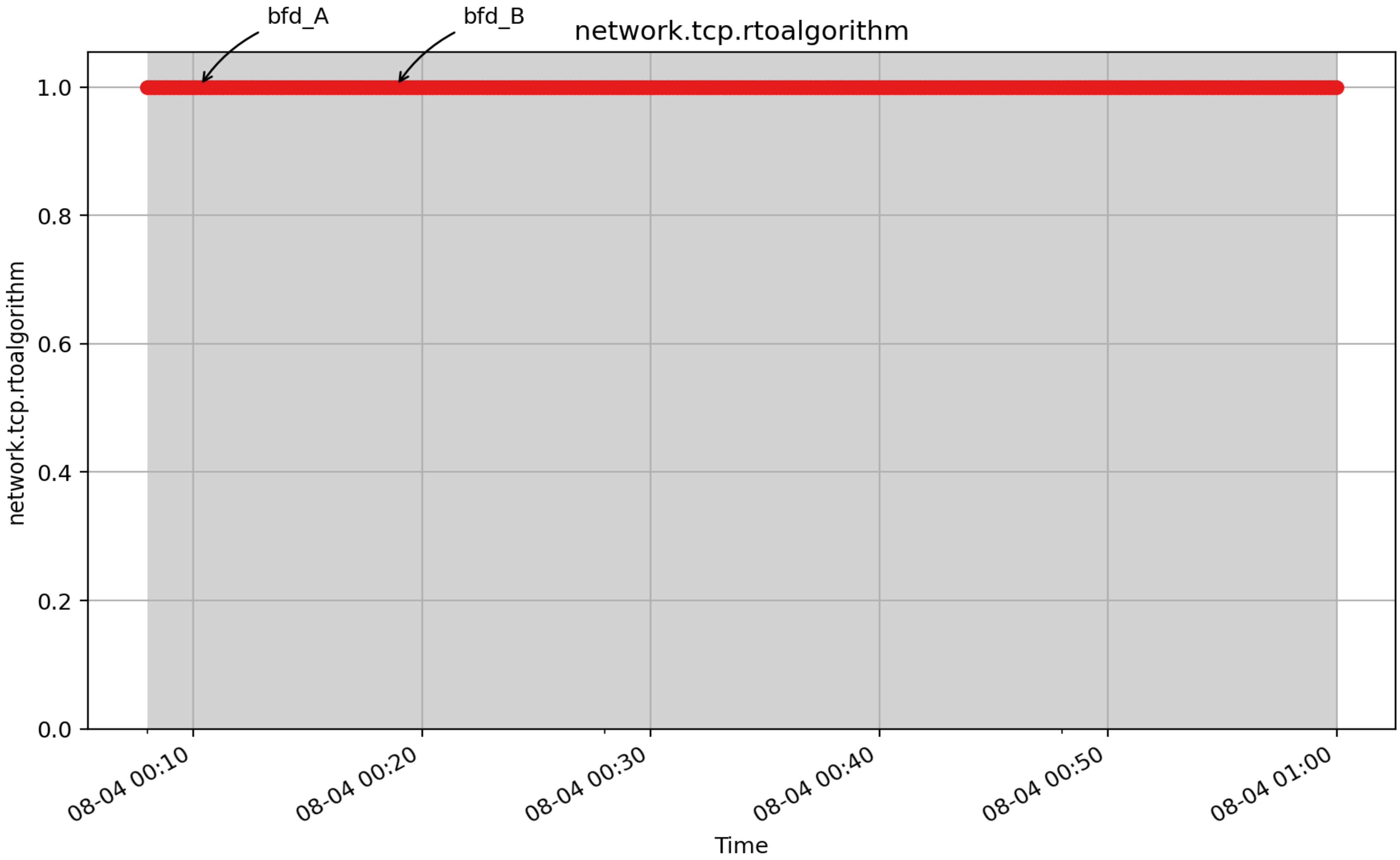
network.tcp.pureacks: Number of acknowledgments not containing data payload received (count - U64) - *rate converted*



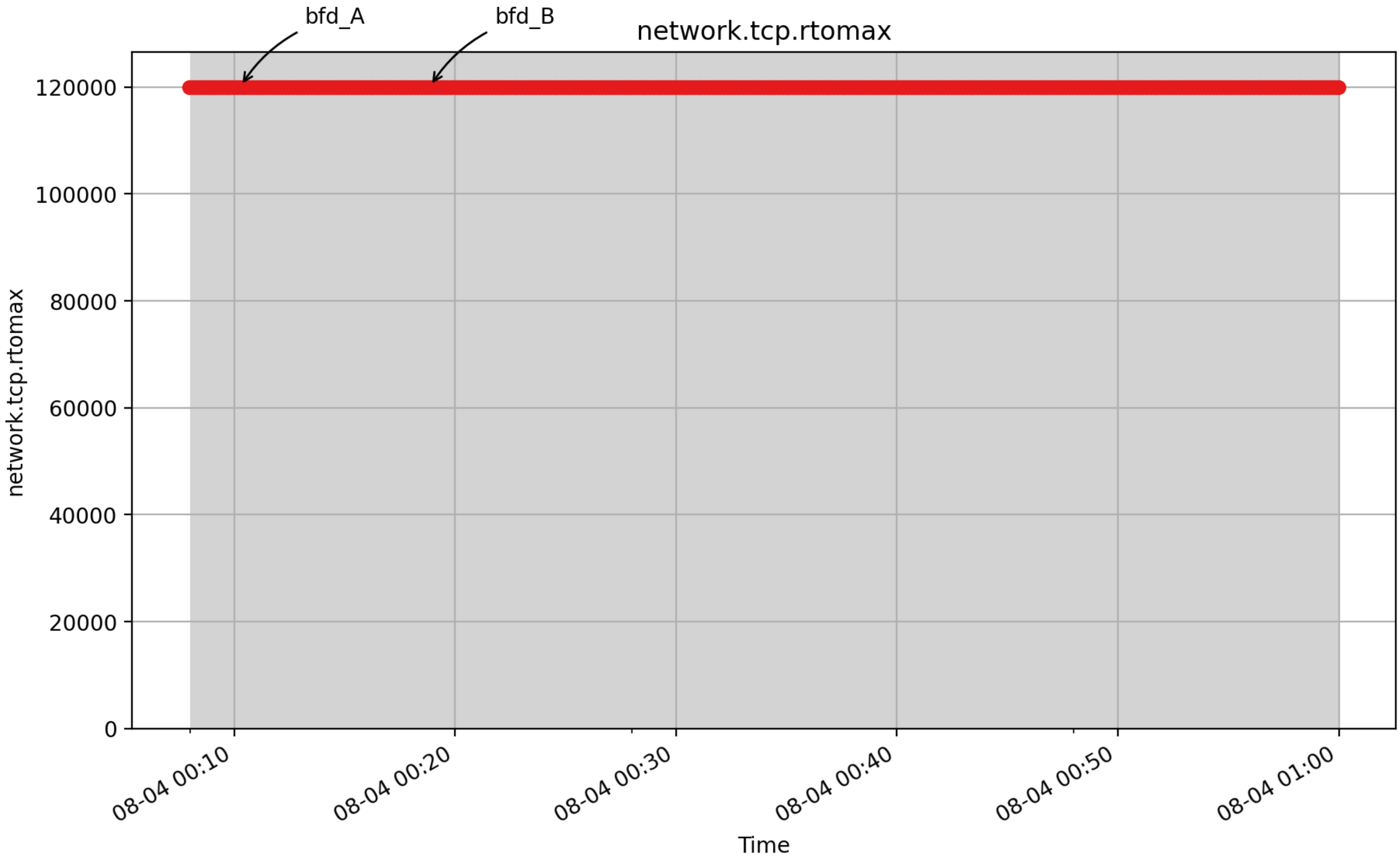
network.tcp.rcvcoalesce: Number of times tried to coalesce the receive queue (count - U64) - *rate converted*



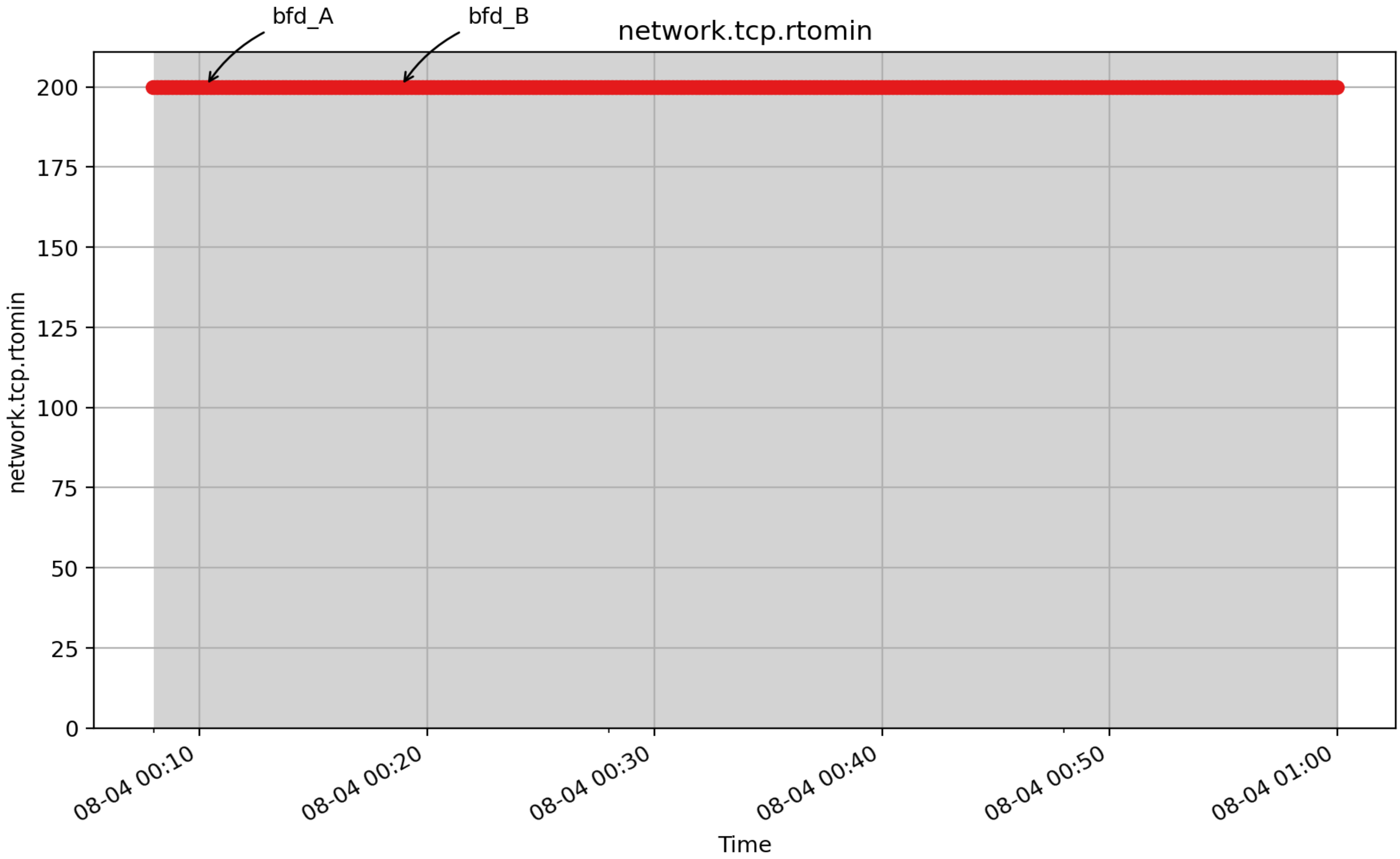
network.tcp.retranssegs: count of tcp segments retransmitted (count - U64) - *rate converted*



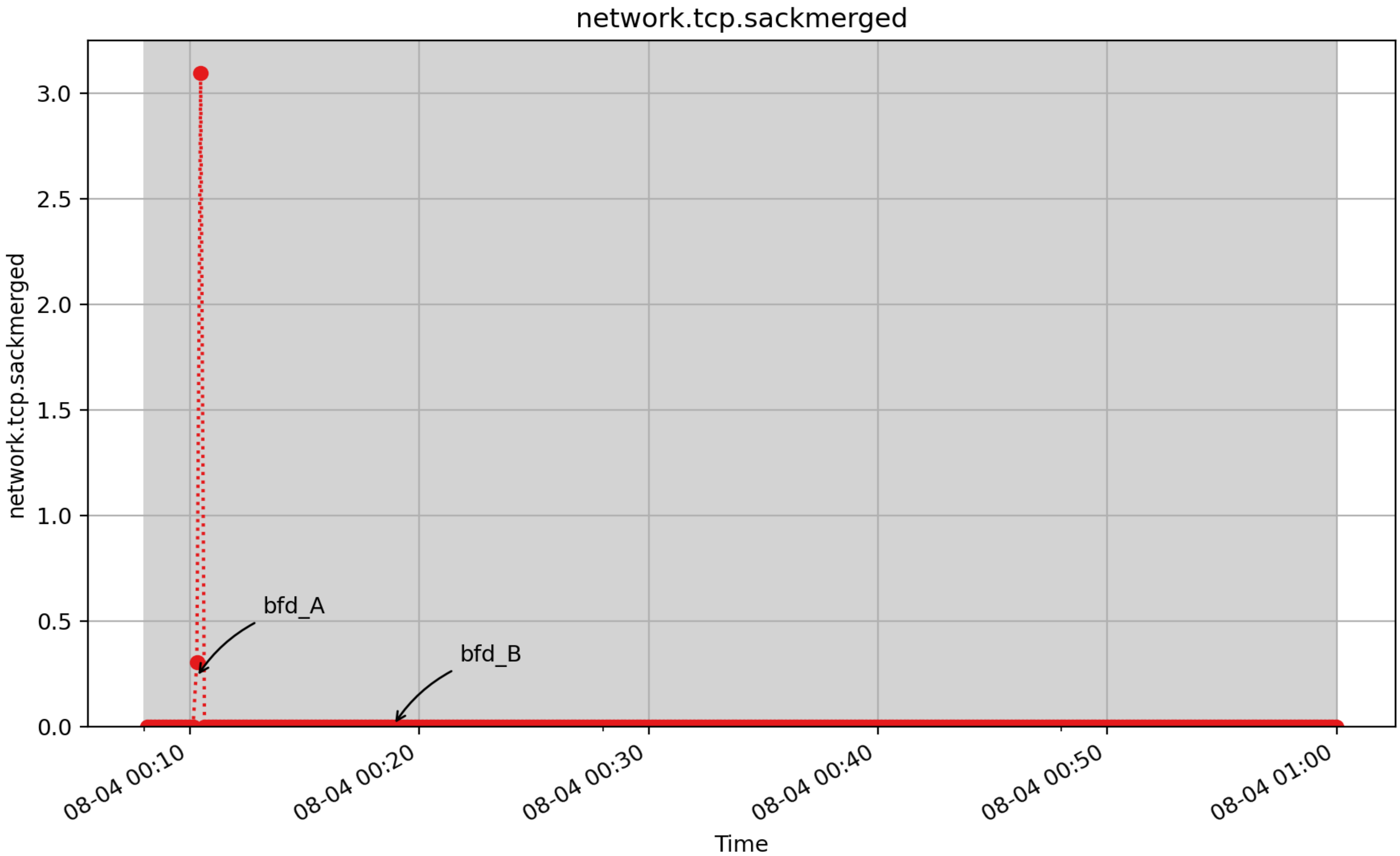
network.tcp.rtoalgorithm: the retransmission timeout algorithm in use (- U64)



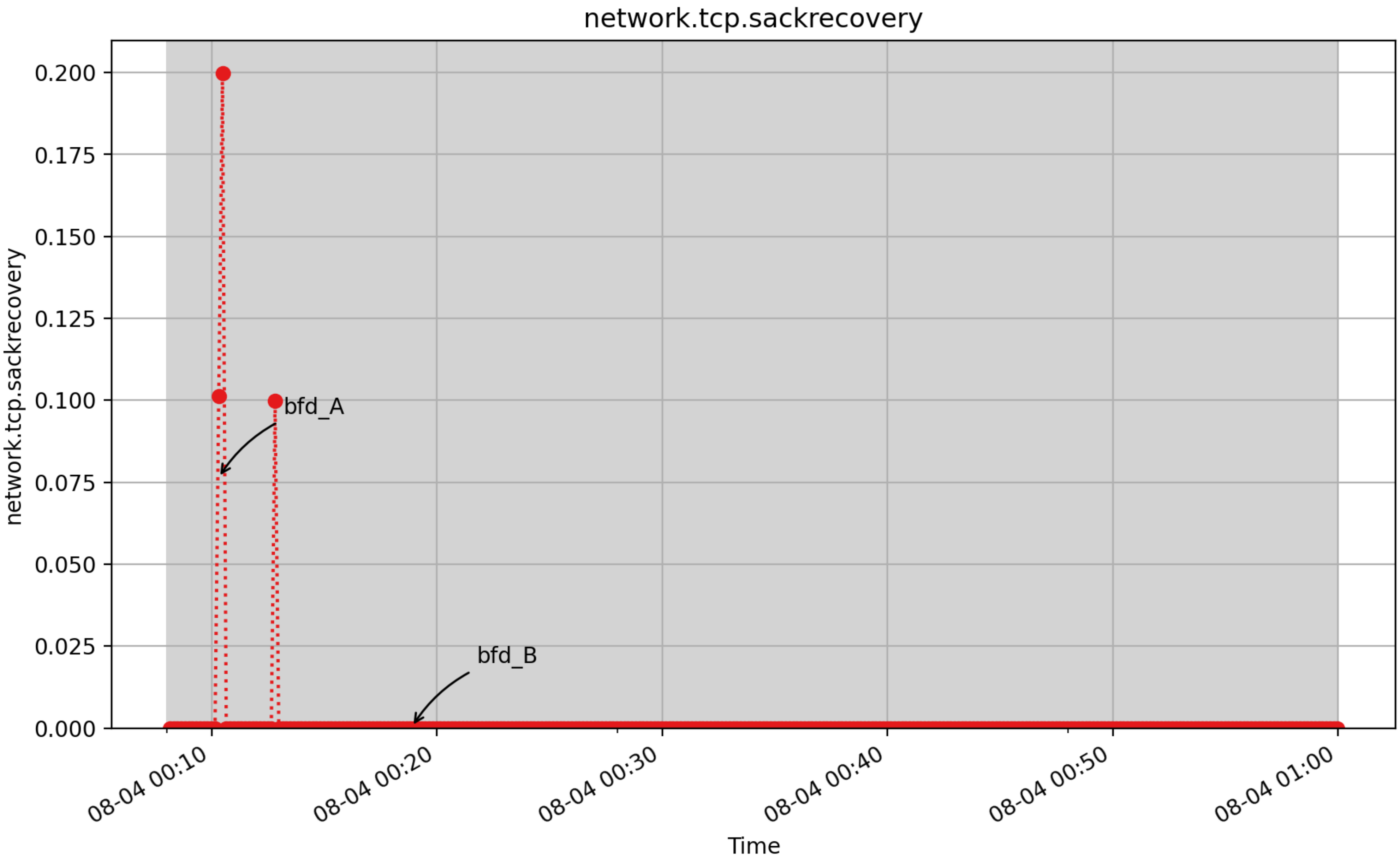
network.tcp.rtomax: maximum retransmission timeout (millisec - U64)



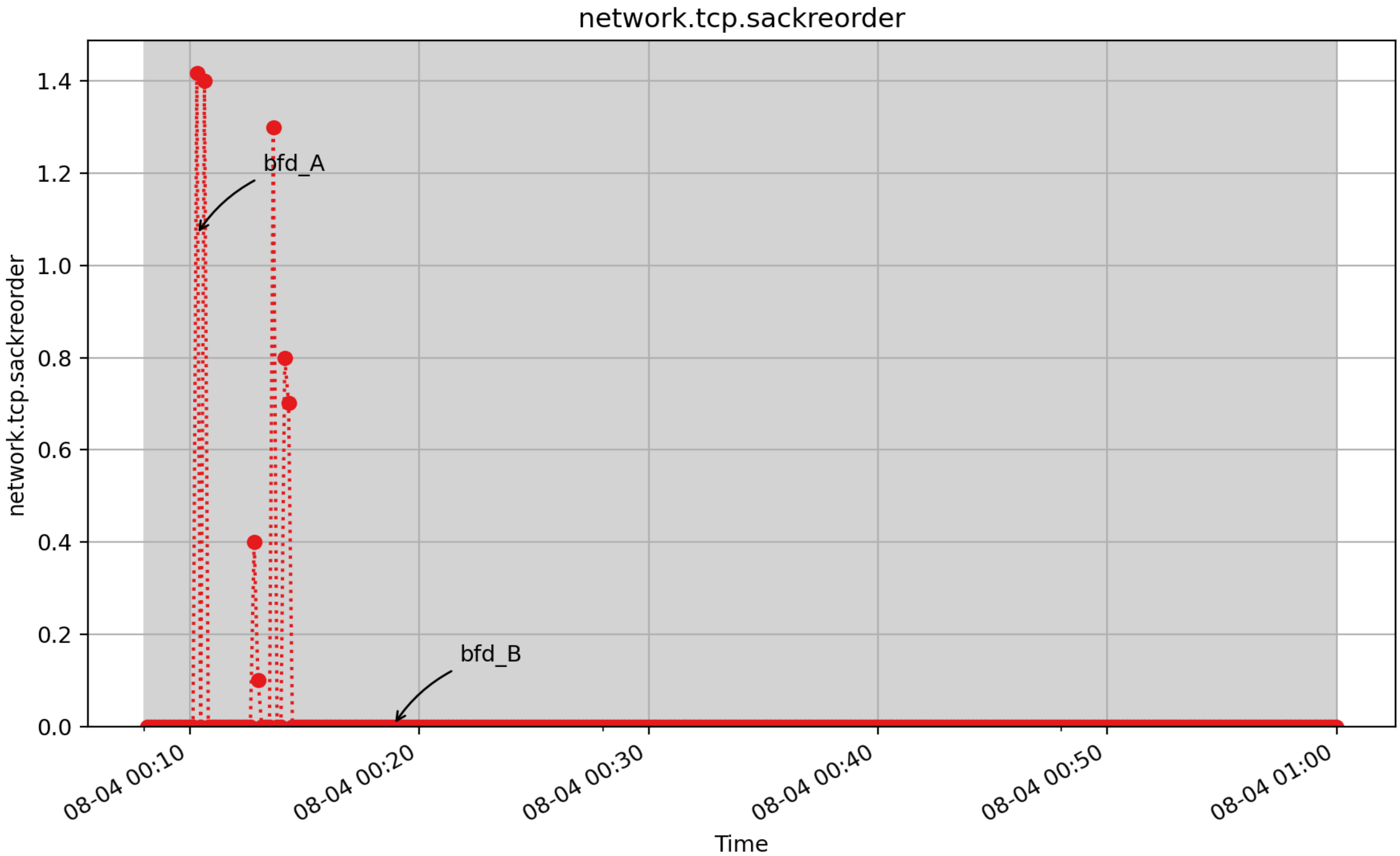
network.tcp.rtomin: minimum retransmission timeout (millisec - U64)



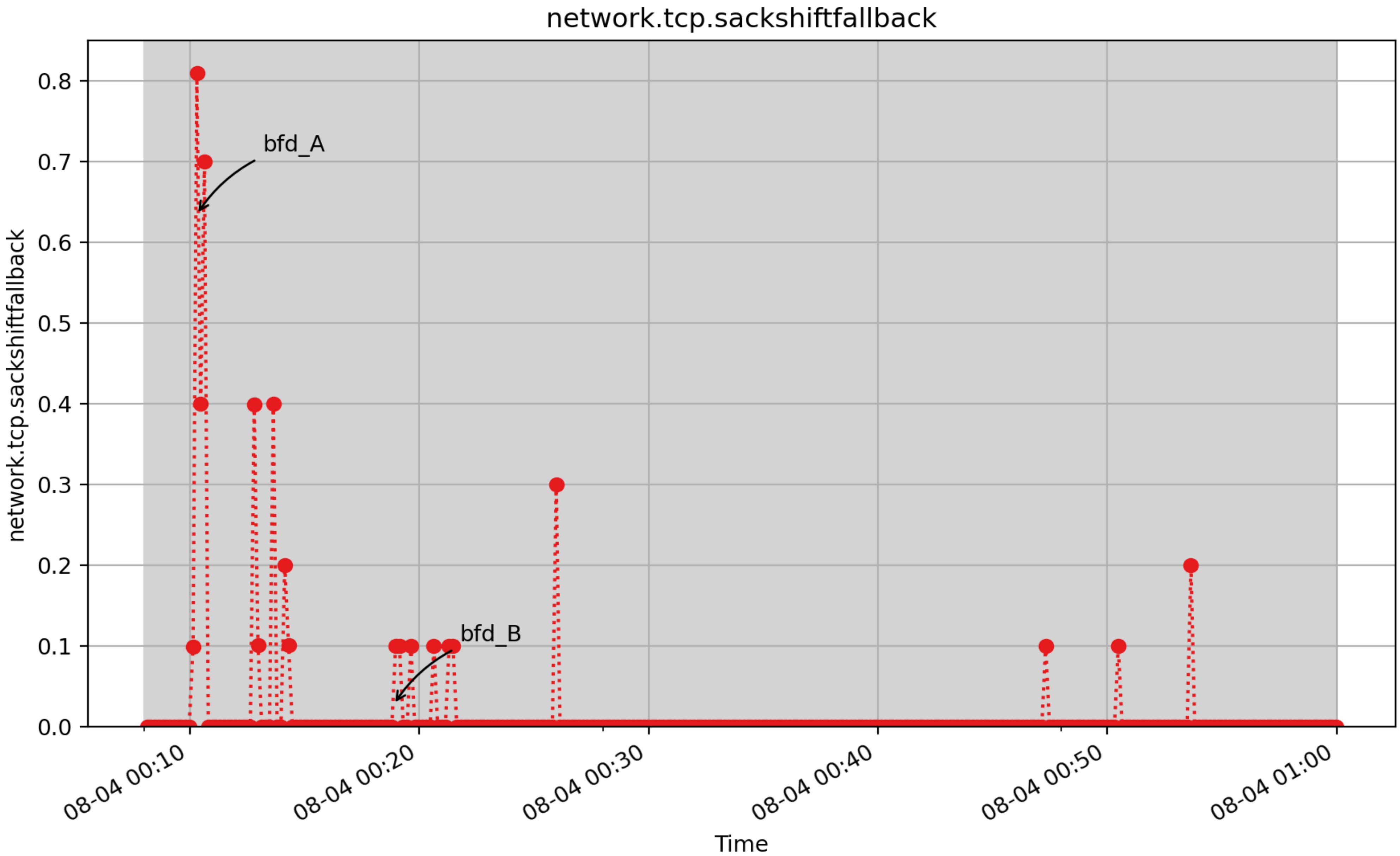
network.tcp.sackmerged: Number of SACKs merged (count - U64) - *rate converted*



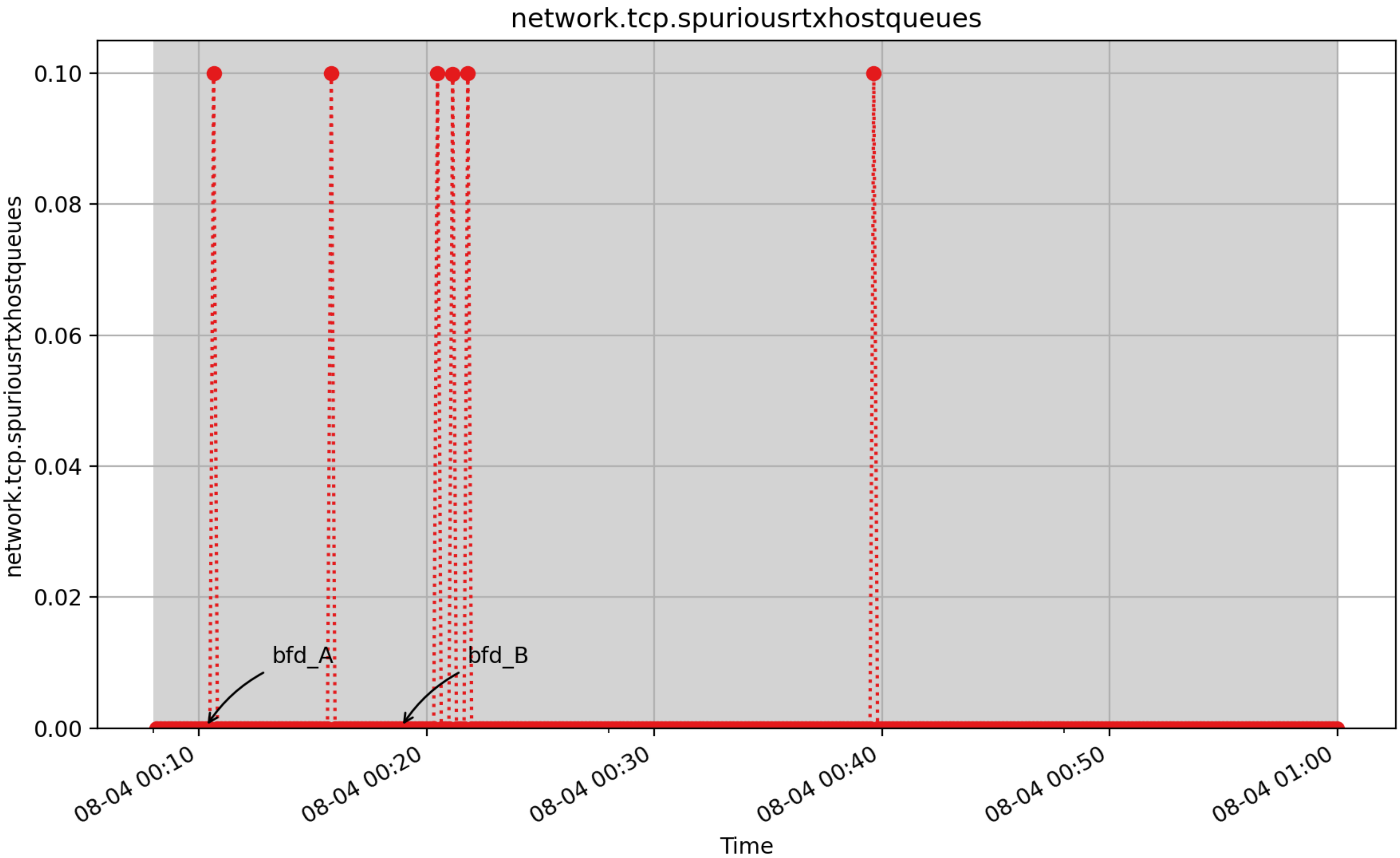
network.tcp.sackrecovery: Number of times recovered from packet loss by selective acknowledgements (count - U64) - *rate converted*



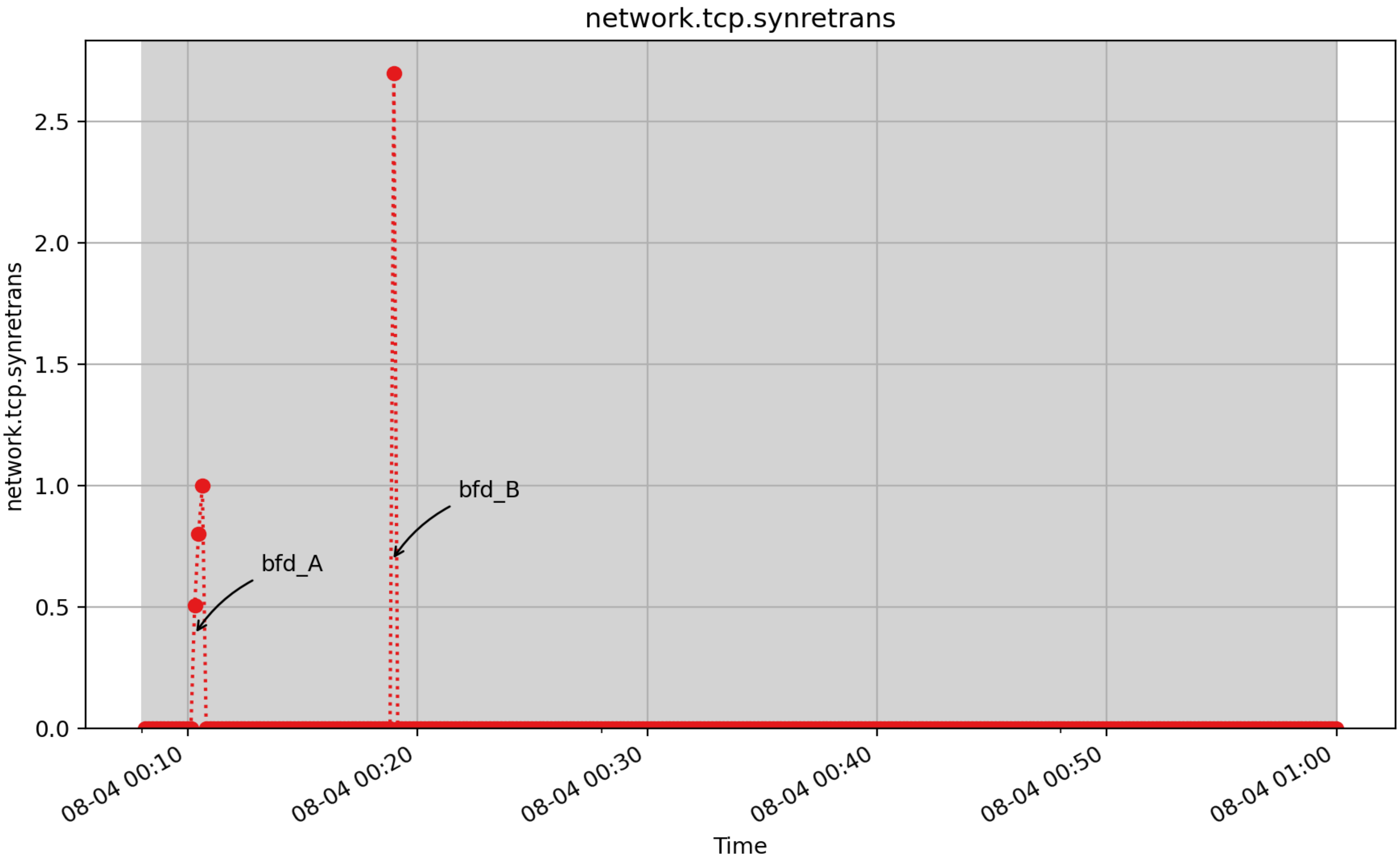
network.tcp.sackreorder: Number of times detected reordering using SACK (count - U64) - *rate converted*



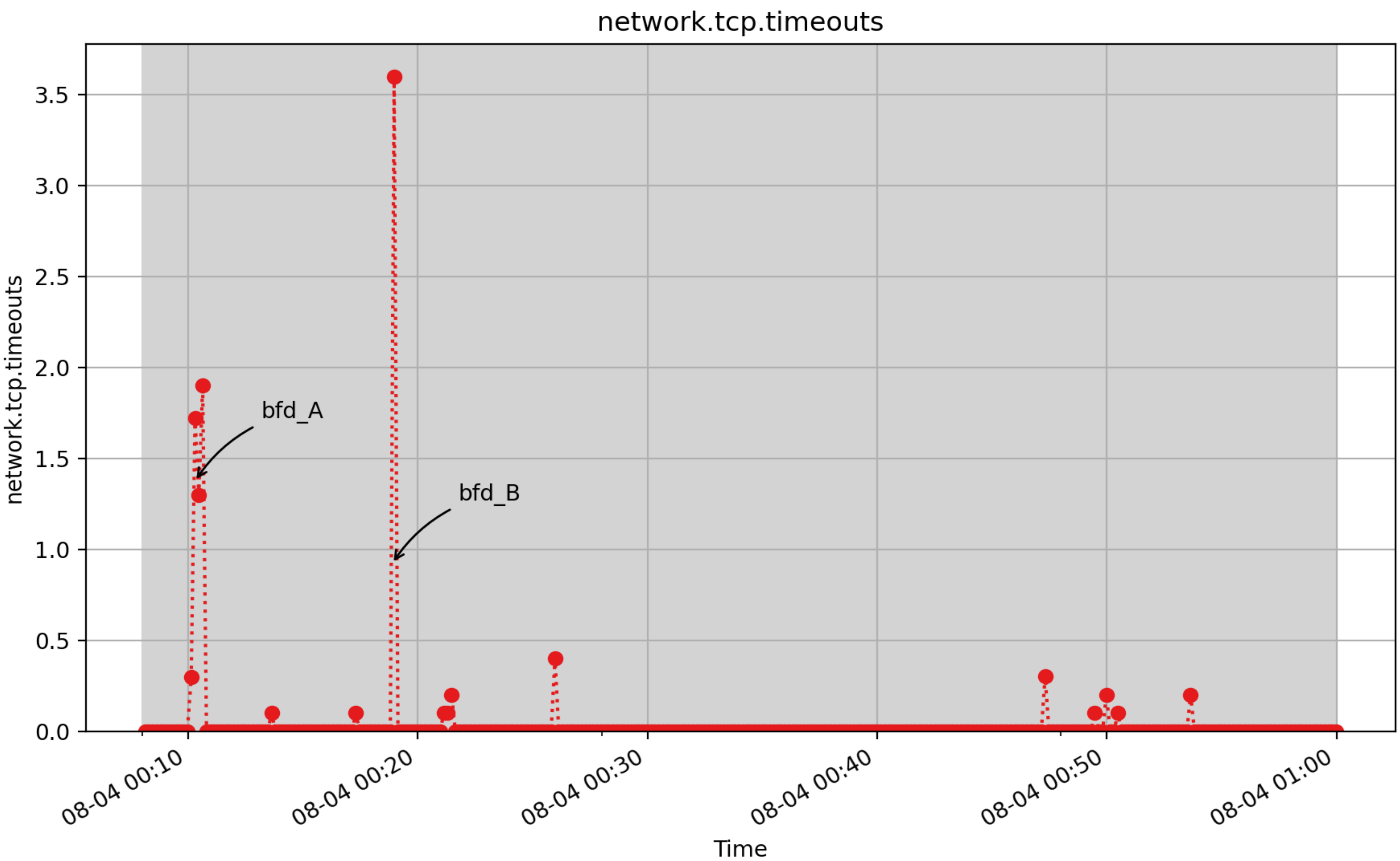
network.tcp.sackshiftfallback: Number of SACKs fallbacks (count - U64) - *rate converted*



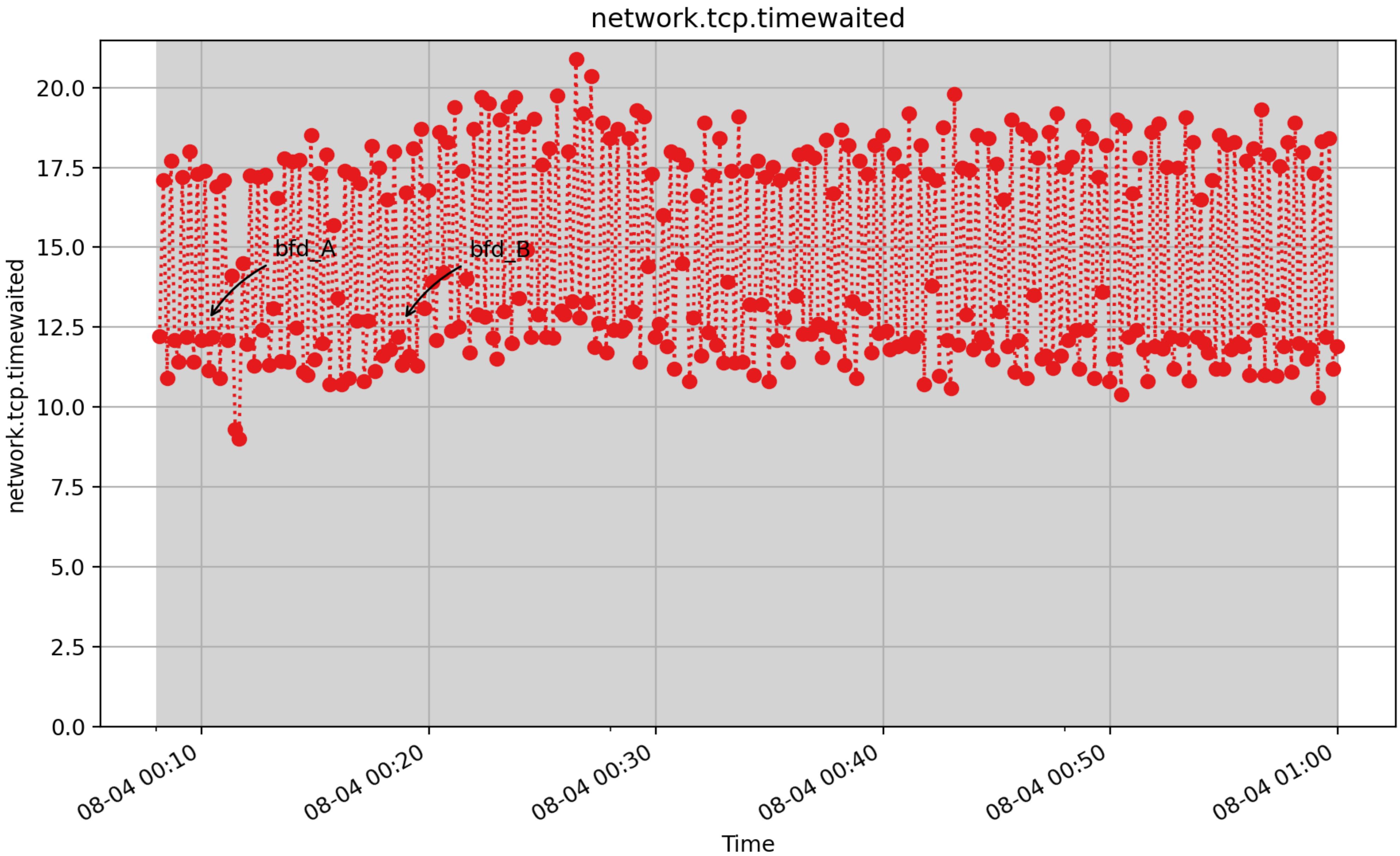
network.tcp.spuriousrtxhostqueues: Number of times that the fast clone is not yet freed in tcp_transmit_skb() (count - U64) - *rate converted*



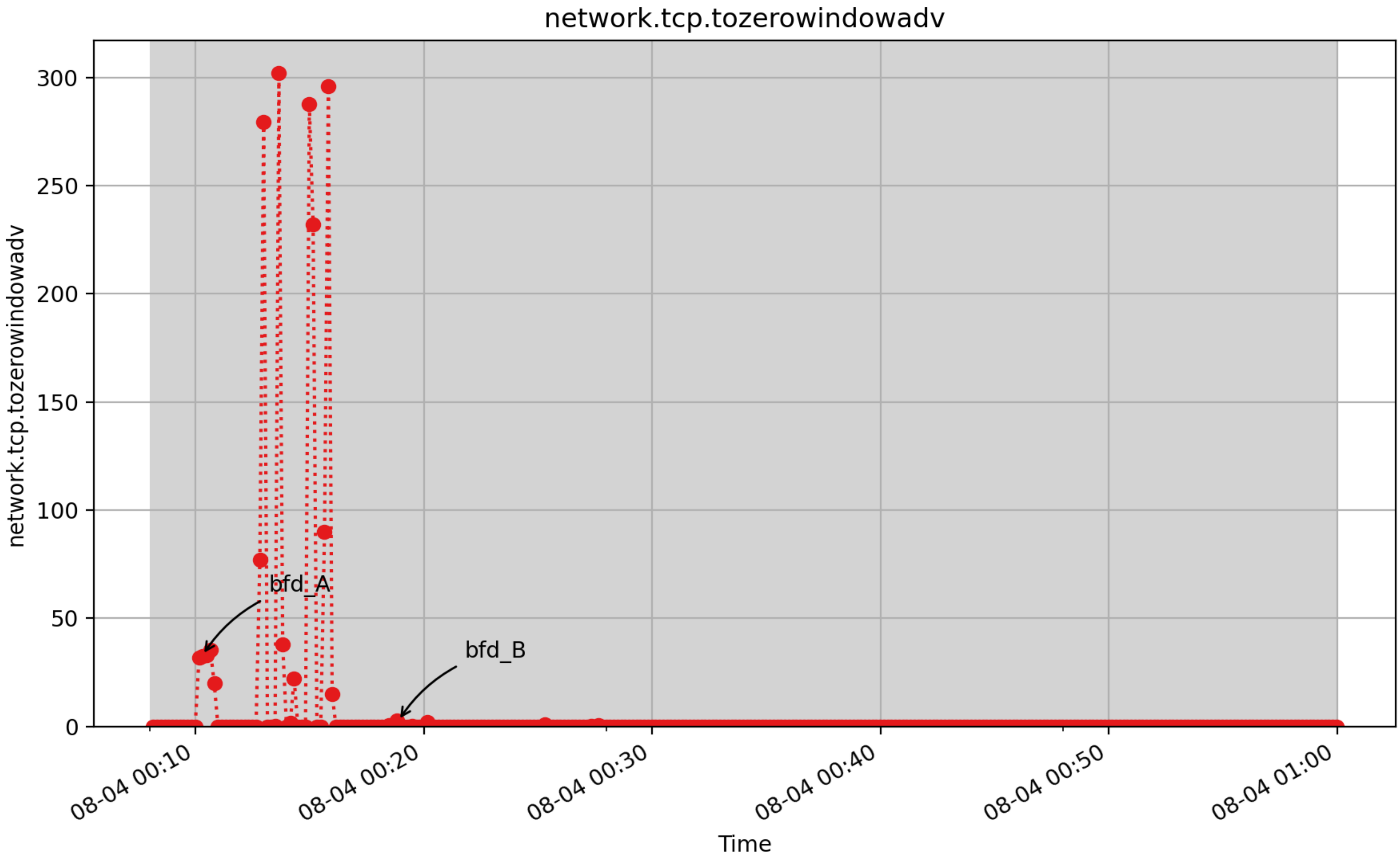
network.tcp.synretrans: Number of SYN-SYN/ACK retransmits to break down retransmissions in SYN, fast/timeout retransmits. (count - U64) - *rate converted*



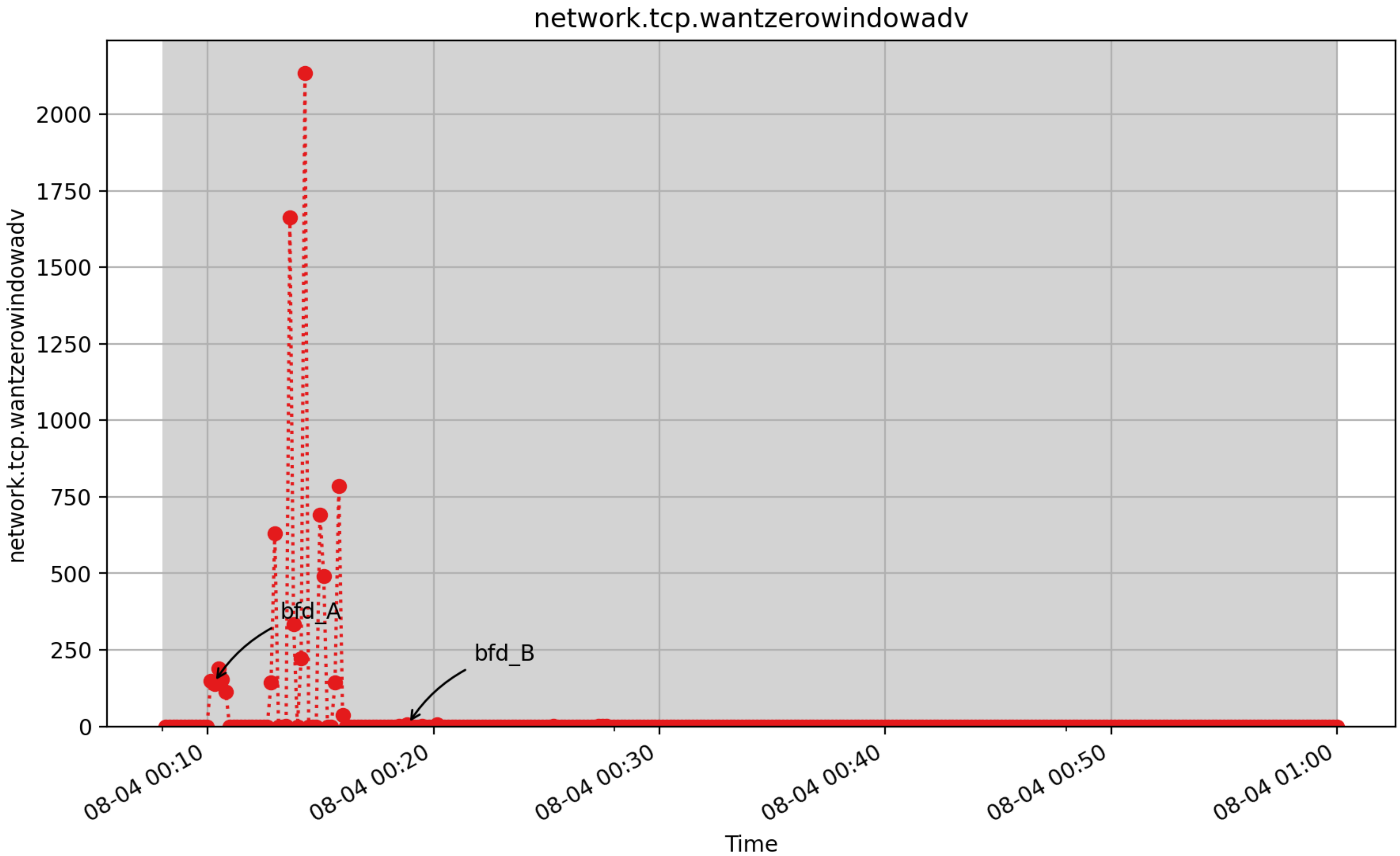
network.tcp.timeouts: Number of other TCP timeouts (count - U64) - *rate converted*



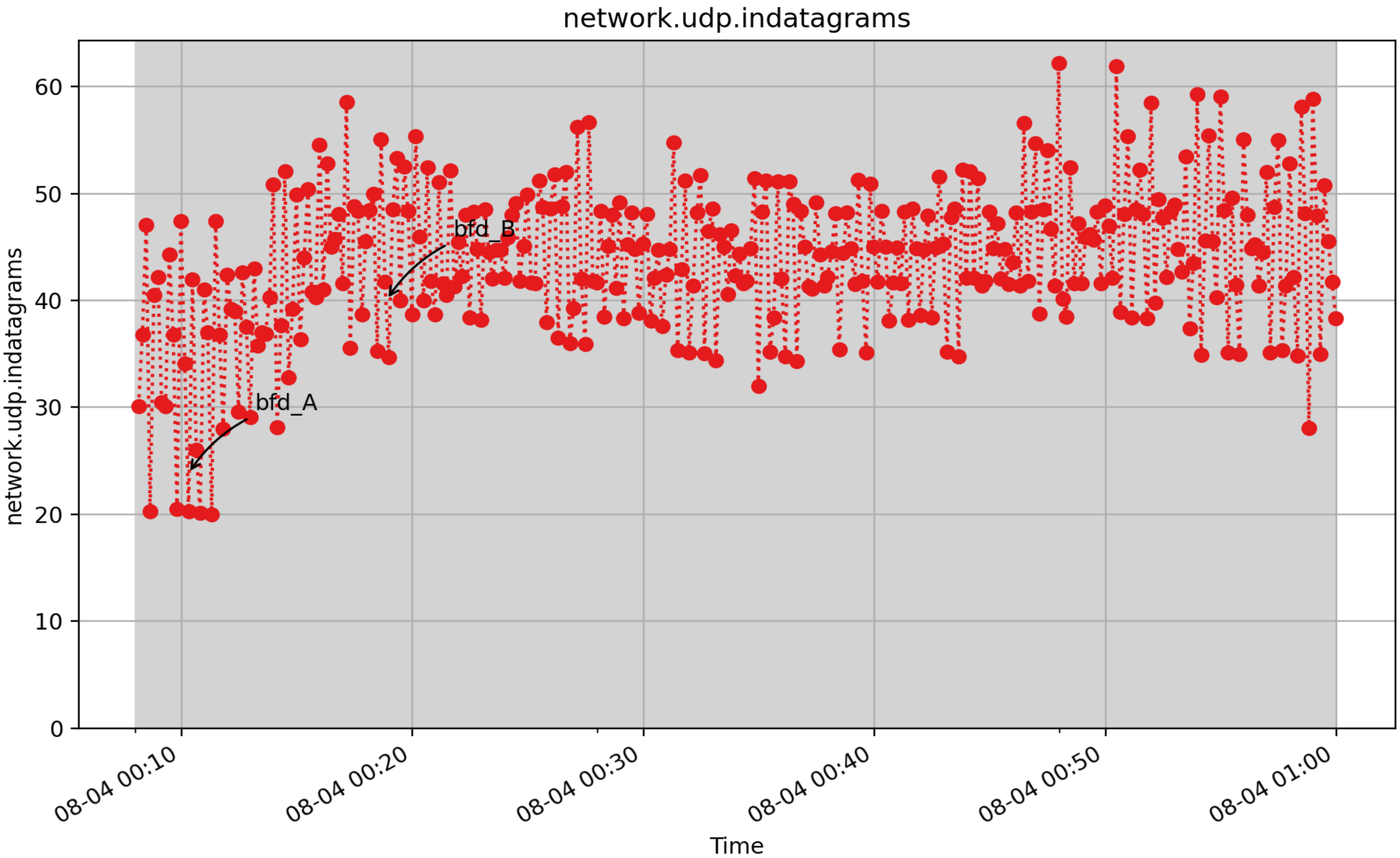
network.tcp.timewaited: Number of TCP sockets finished time wait in fast timer (count - U64) - *rate converted*



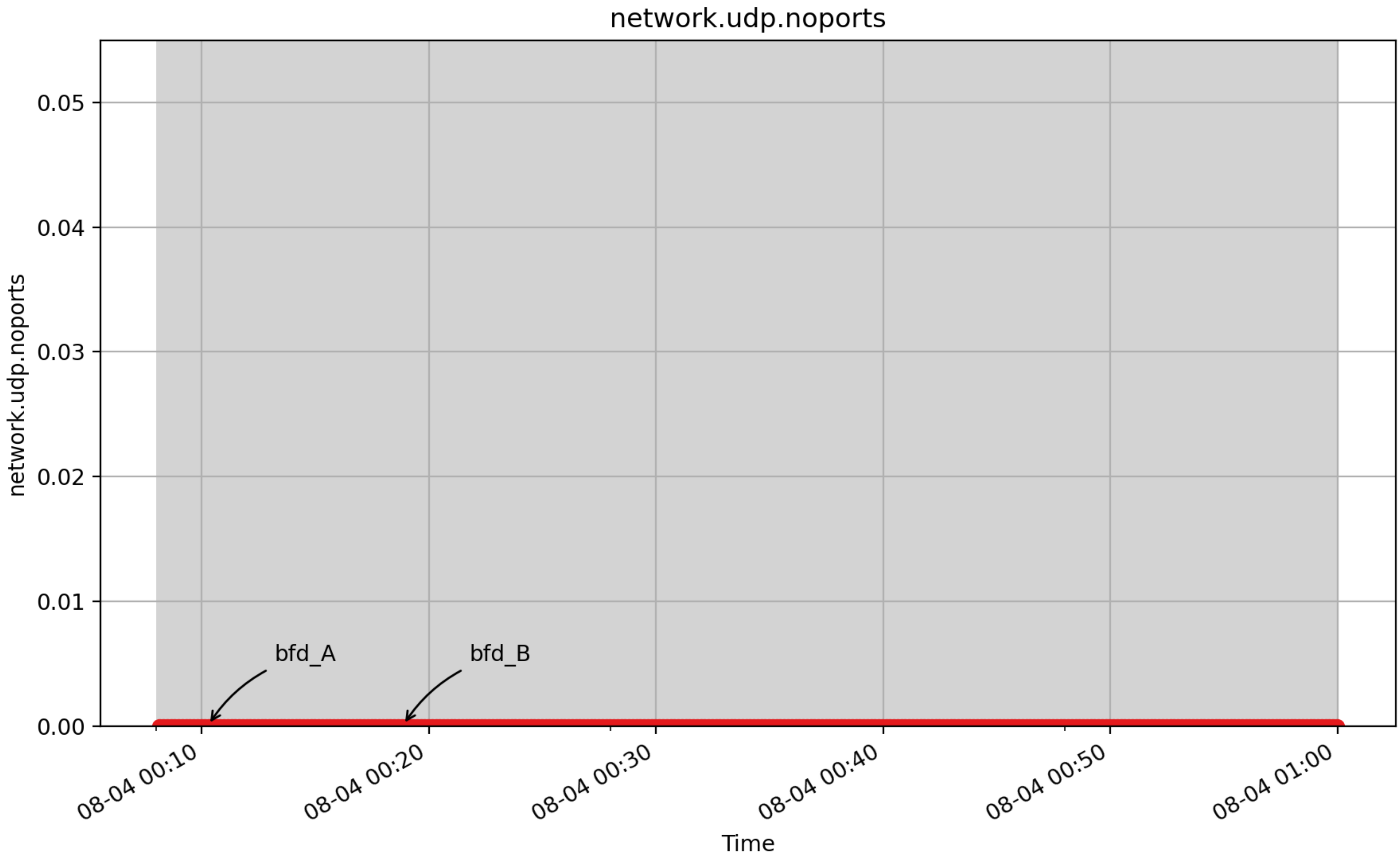
network.tcp.tozerowindowadv: Number of times window went from non-zero to zero (count - U64) - *rate converted*



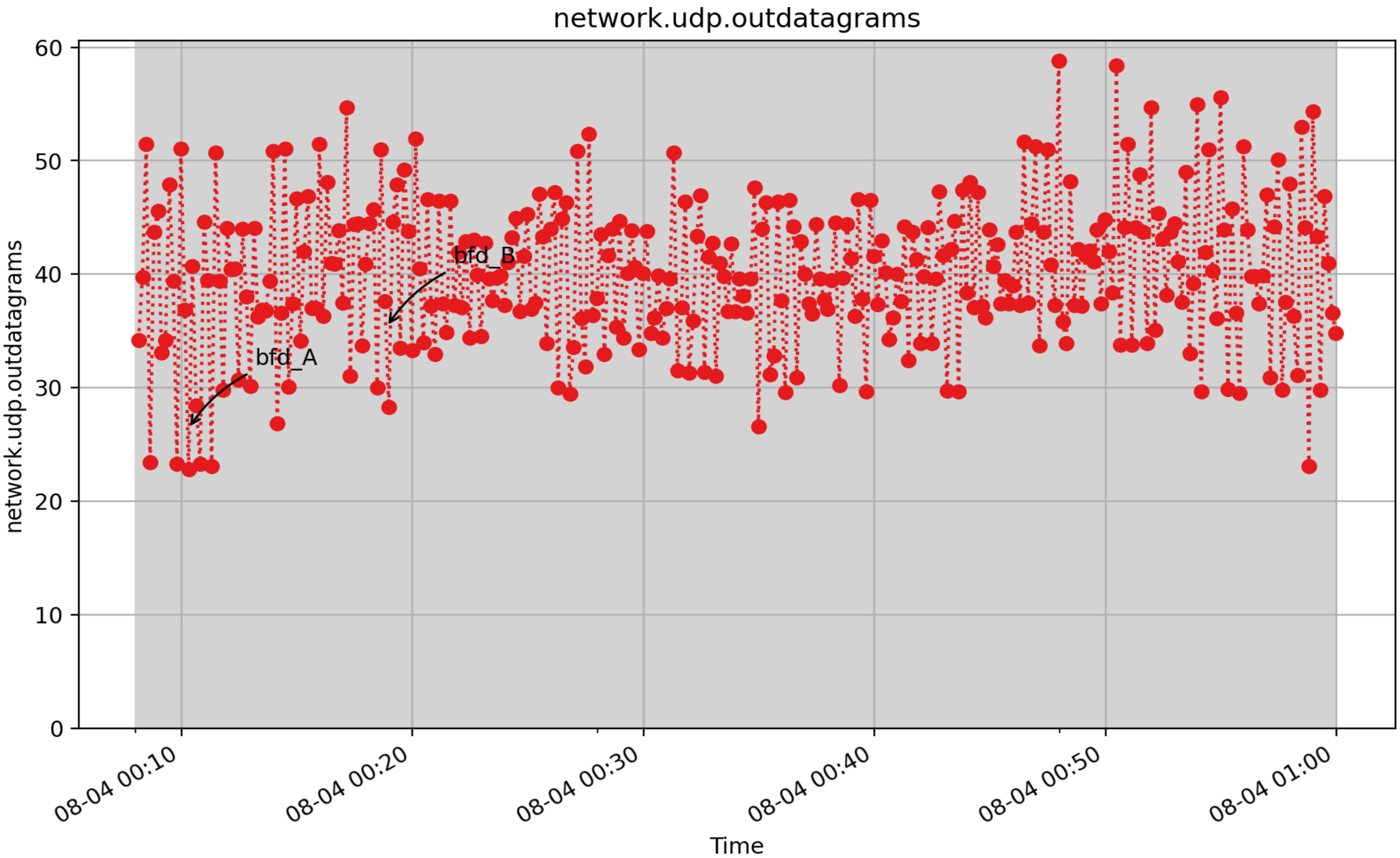
network.tcp.wantzerowindowadv: Number of times zero window announced (count - U64) - *rate converted*



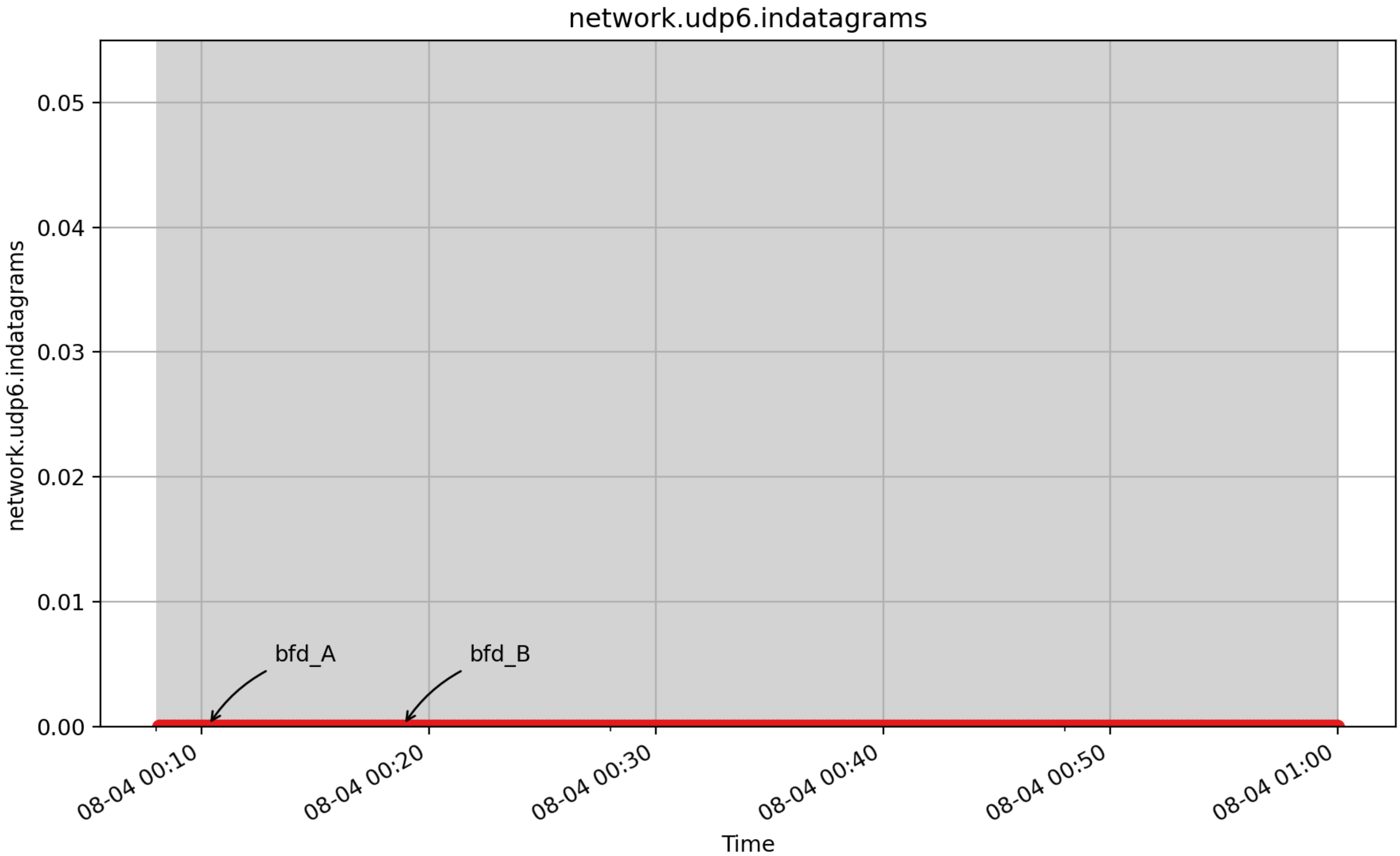
network.udp.indatagrams: count of udp indatagrams (count - U64) - *rate converted*



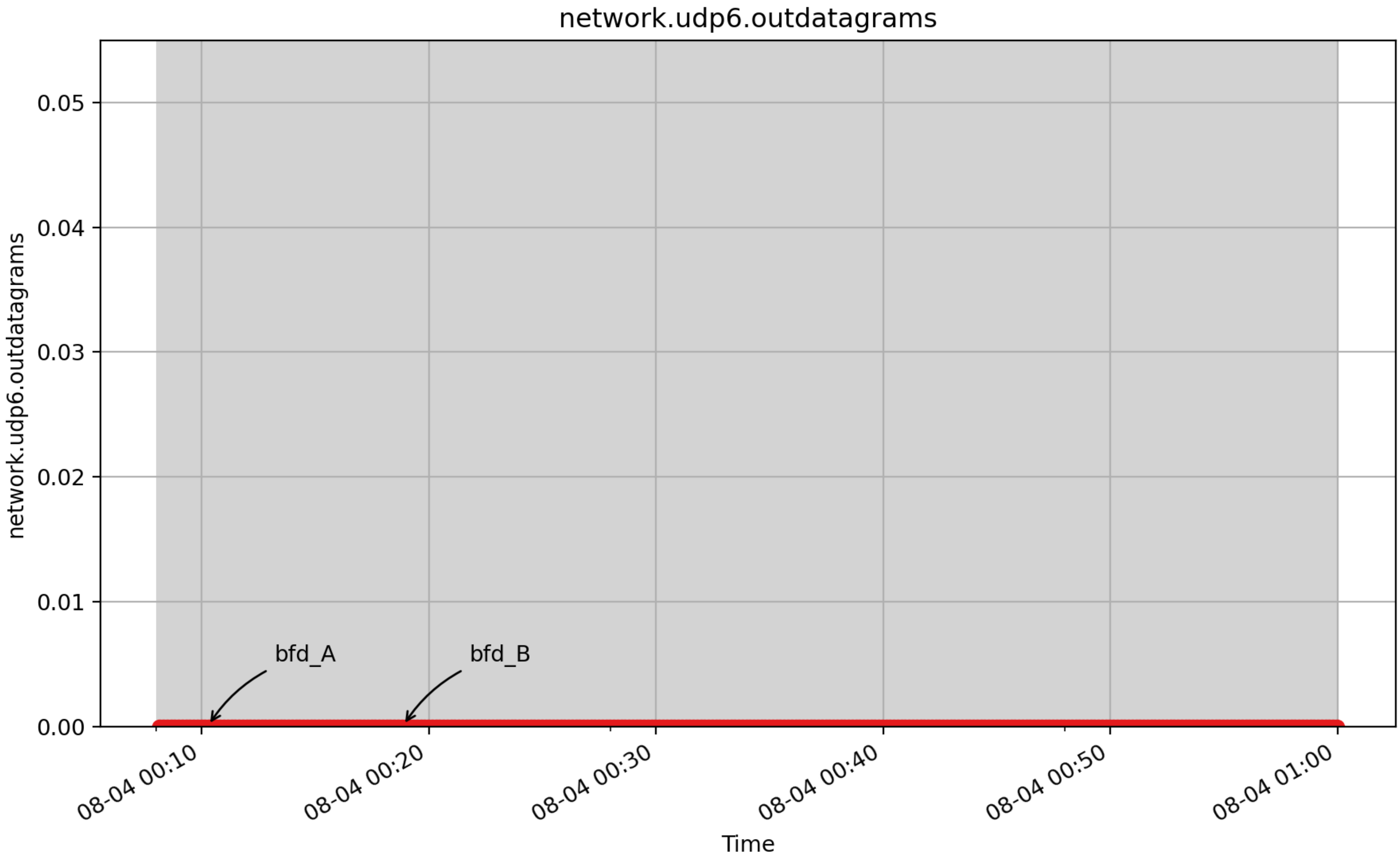
network.udp.noports: count of udp noports (count - U64) - *rate converted*



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

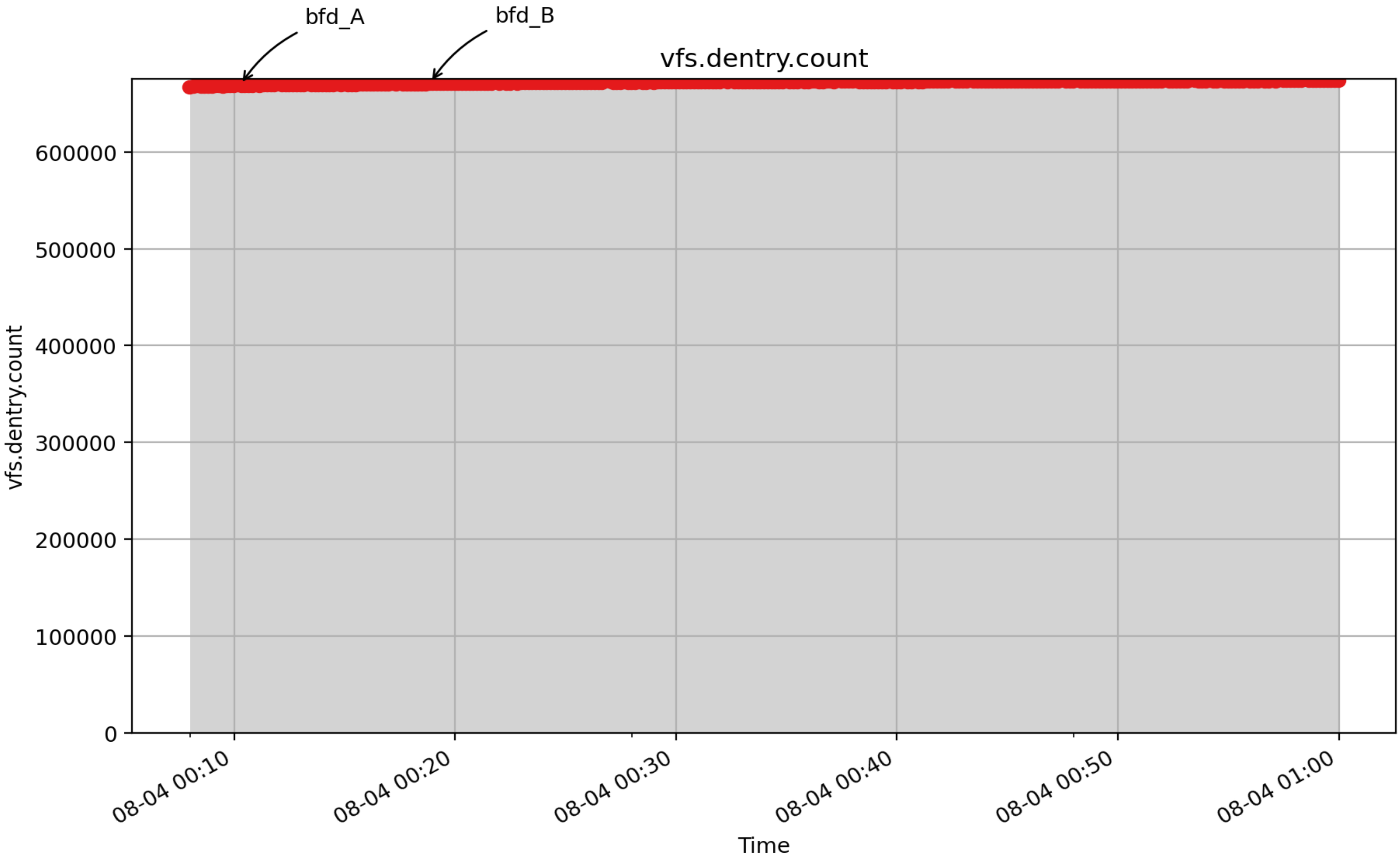


network.udp6.indatagrams: count of udp6 indatagrams (count - U64) - *rate converted*

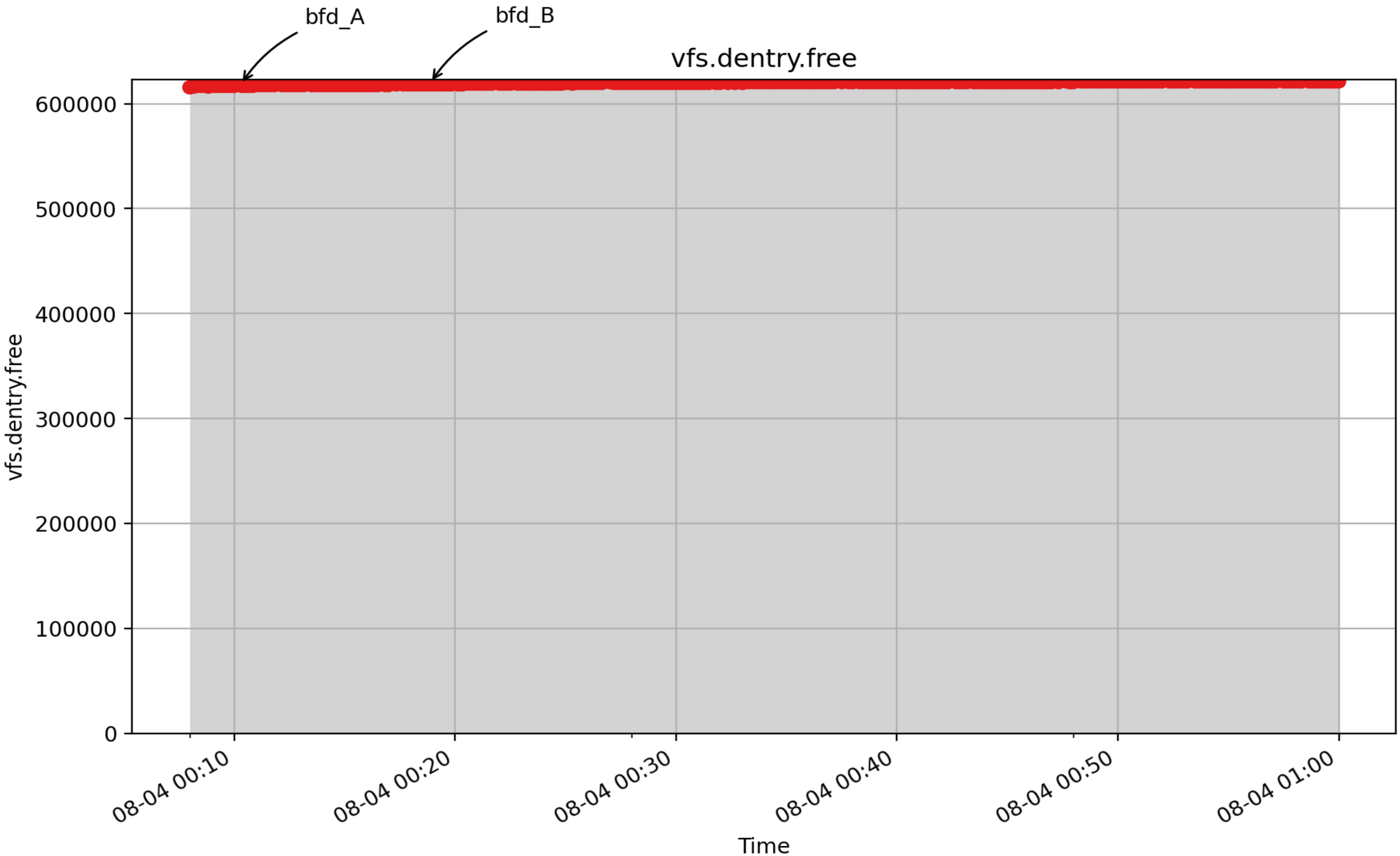


network.udp6.outdatagrams: count of udp6 outdatagrams (count - U64) - *rate converted*

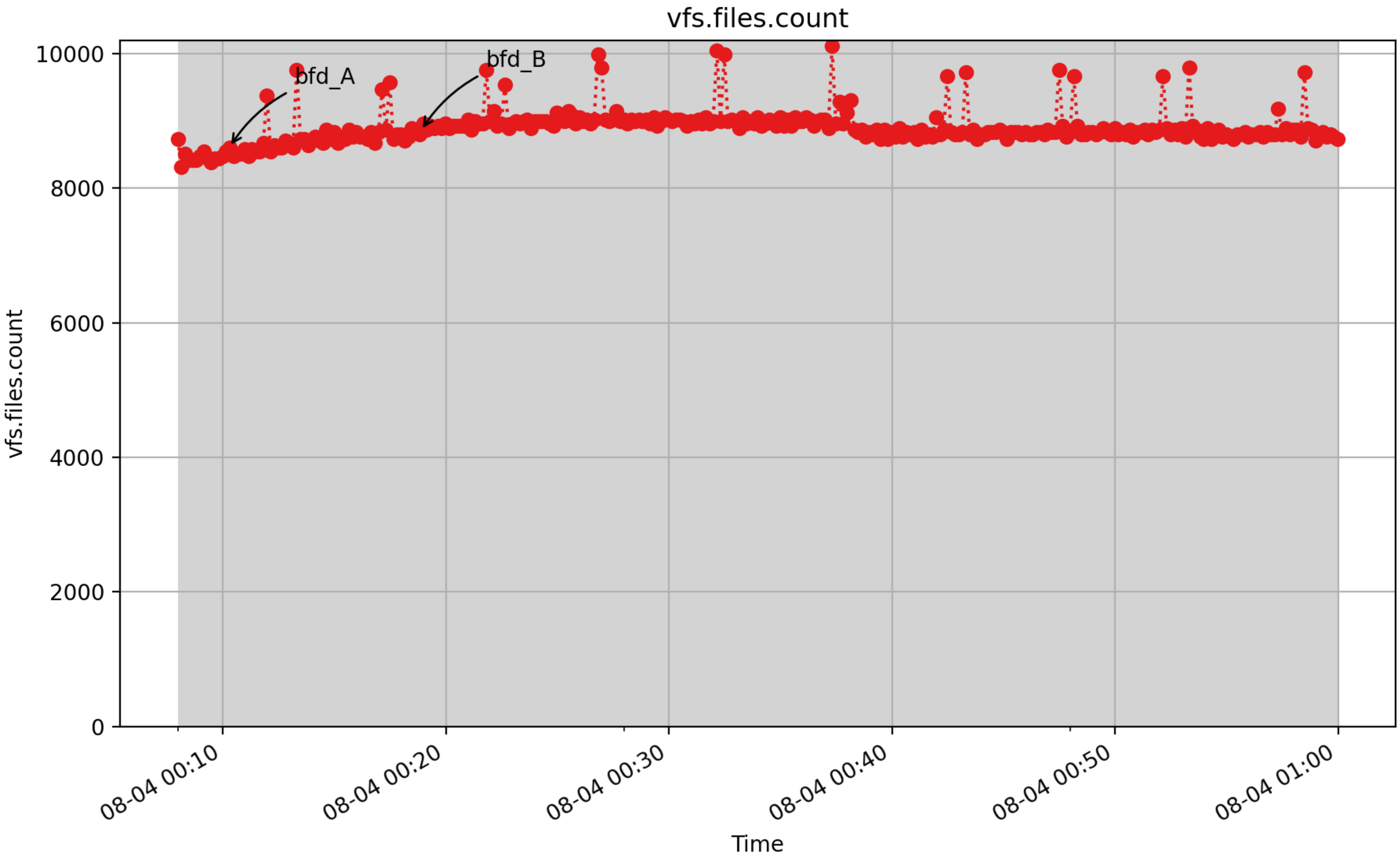
Vfs



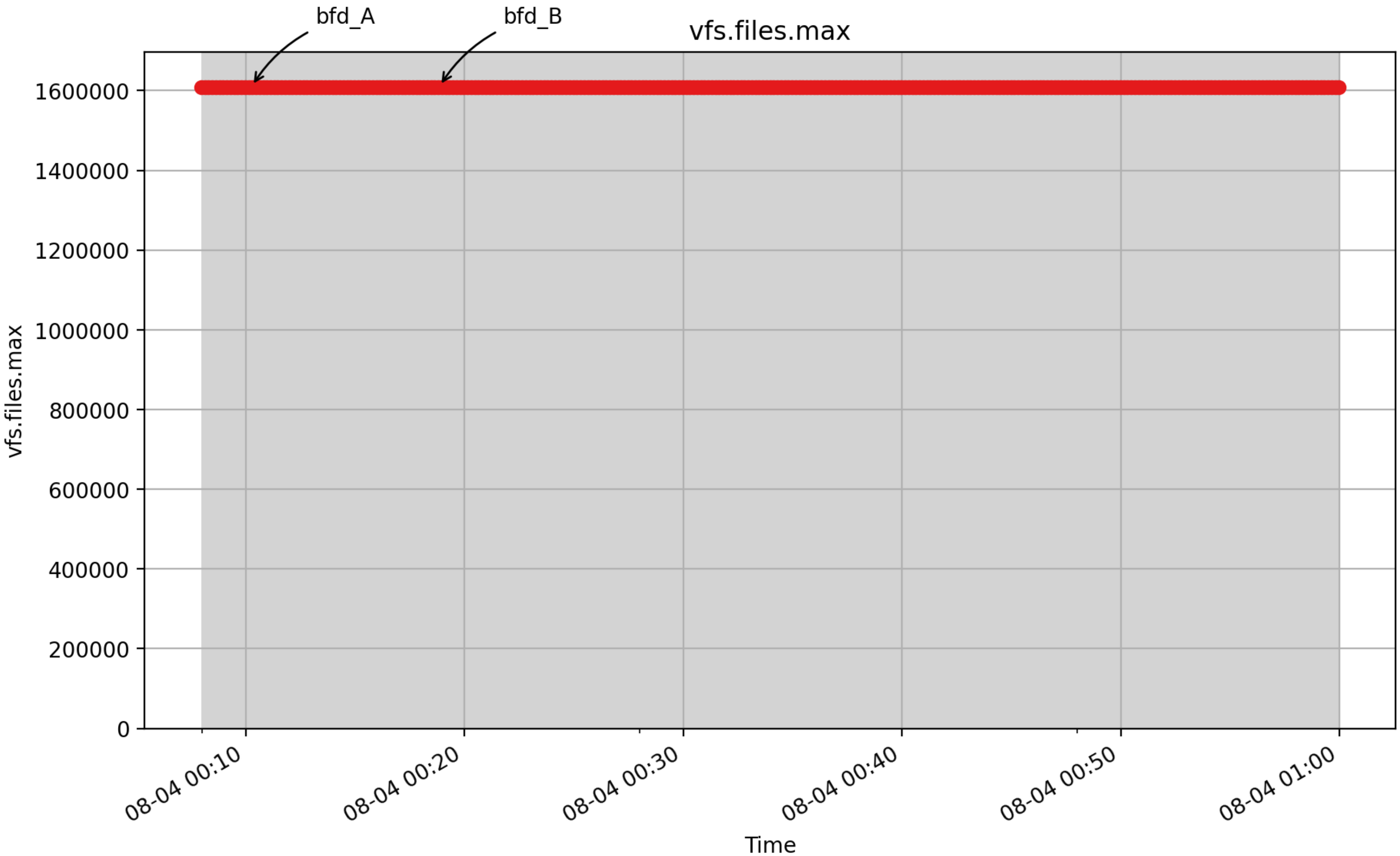
vfs.dentry.count: number of in-use directory entry structures (- 32)



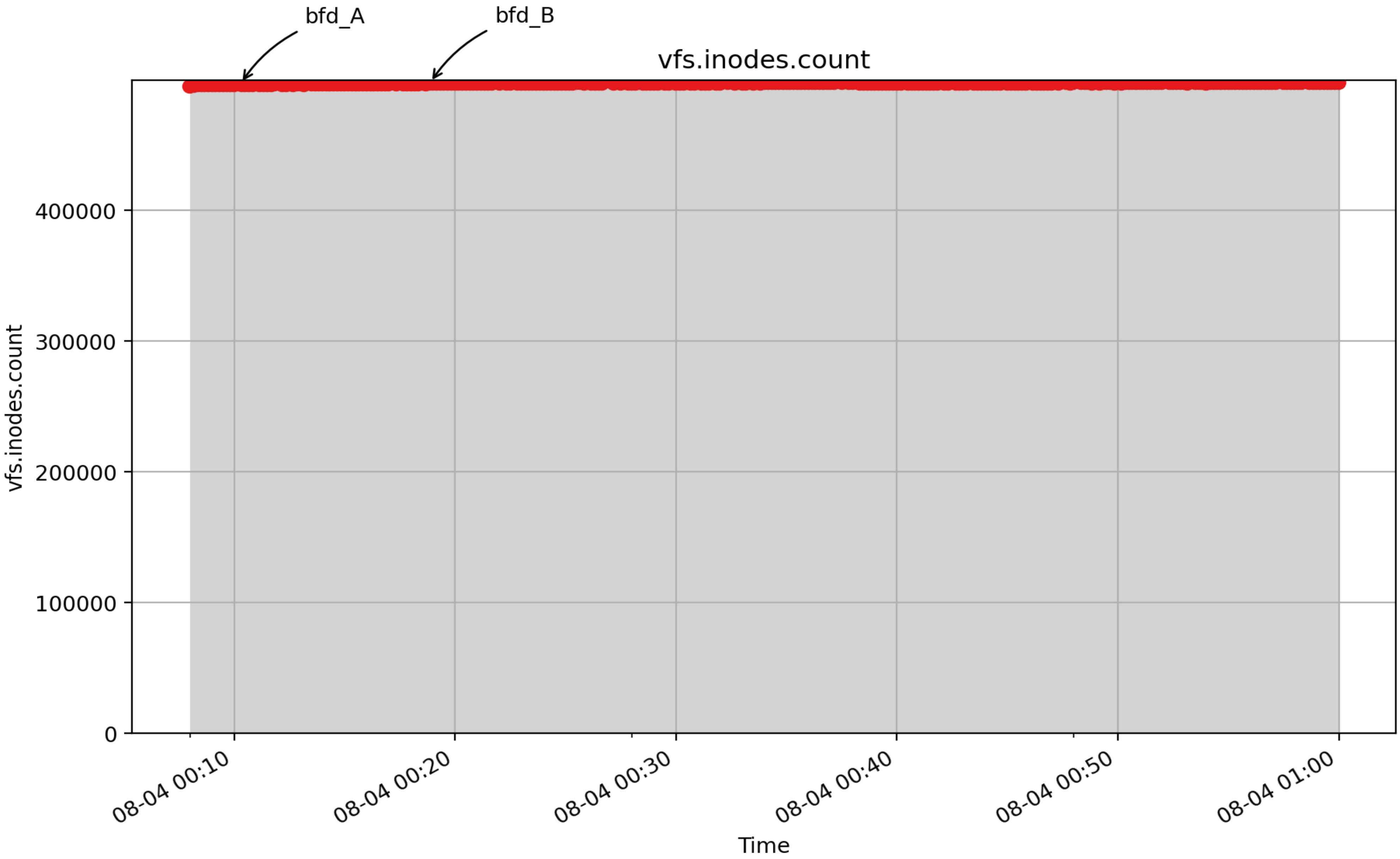
vfs.dentry.free: number of available directory entry structures (- 32)



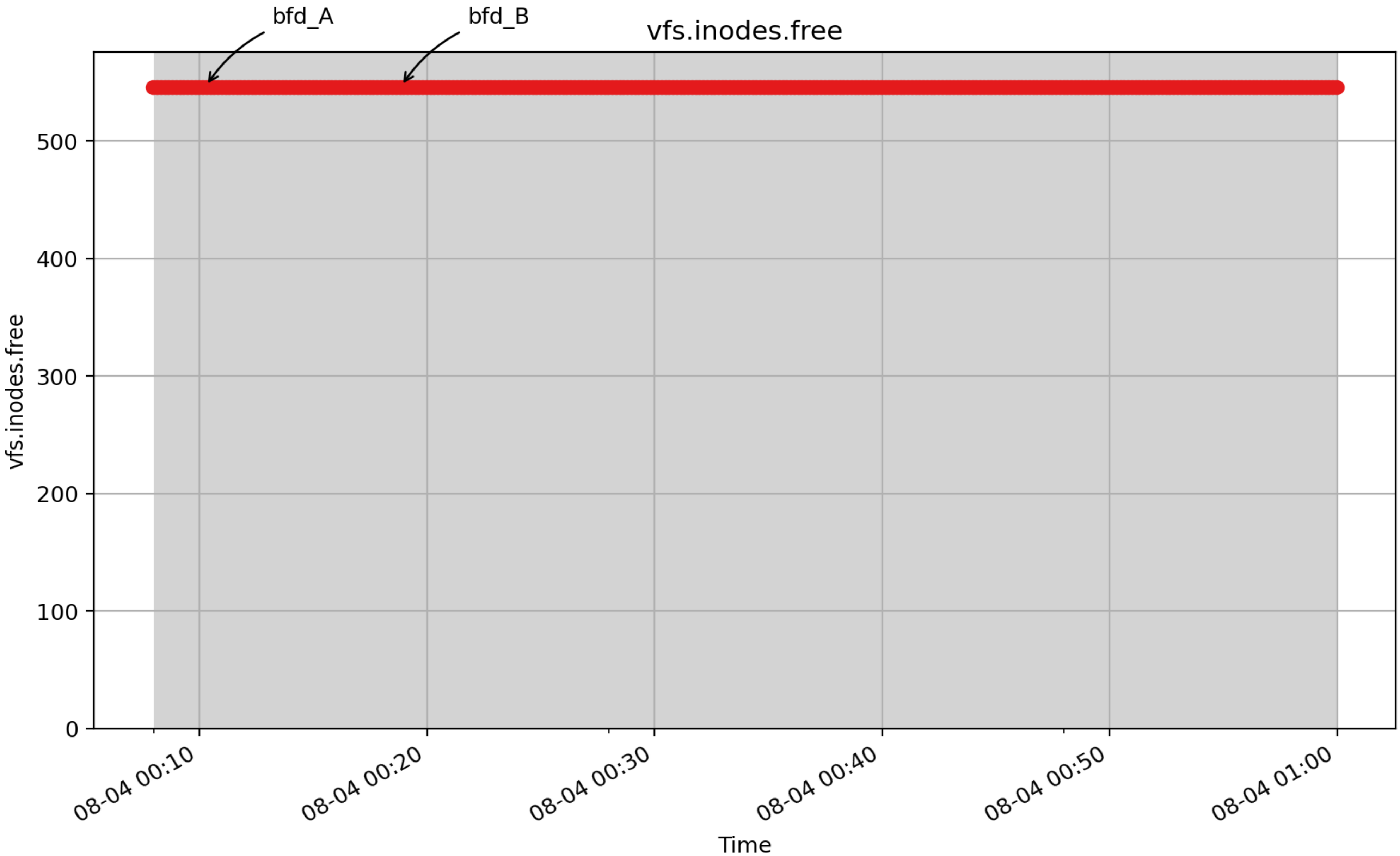
vfs.files.count: number of in-use file structures (- 32)



vfs.files.max: hard maximum on number of file structures (- 32)

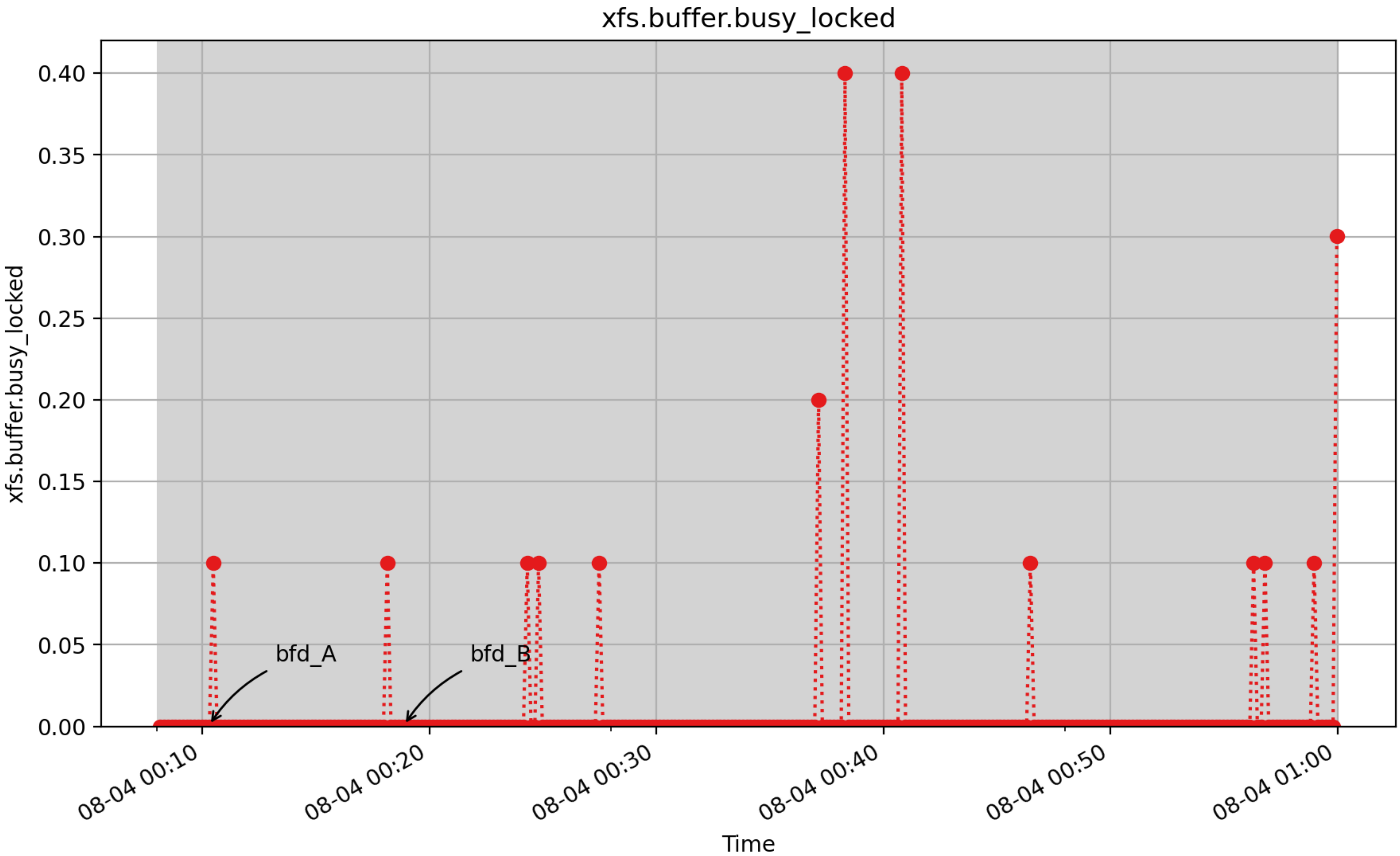


vfs.inodes.count: number of in-use inode structures (- 32)

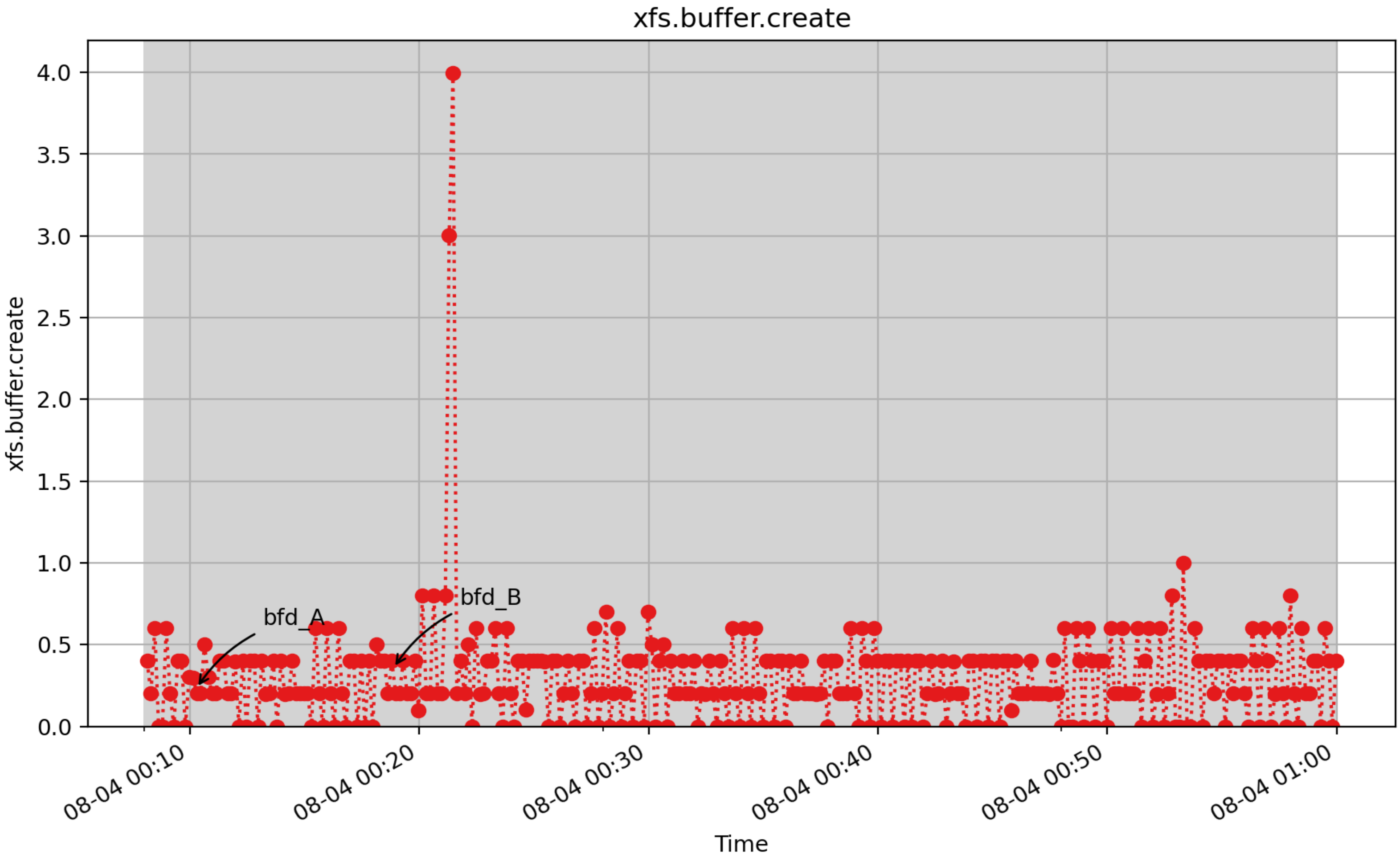


vfs.inodes.free: number of available inode structures (- 32)

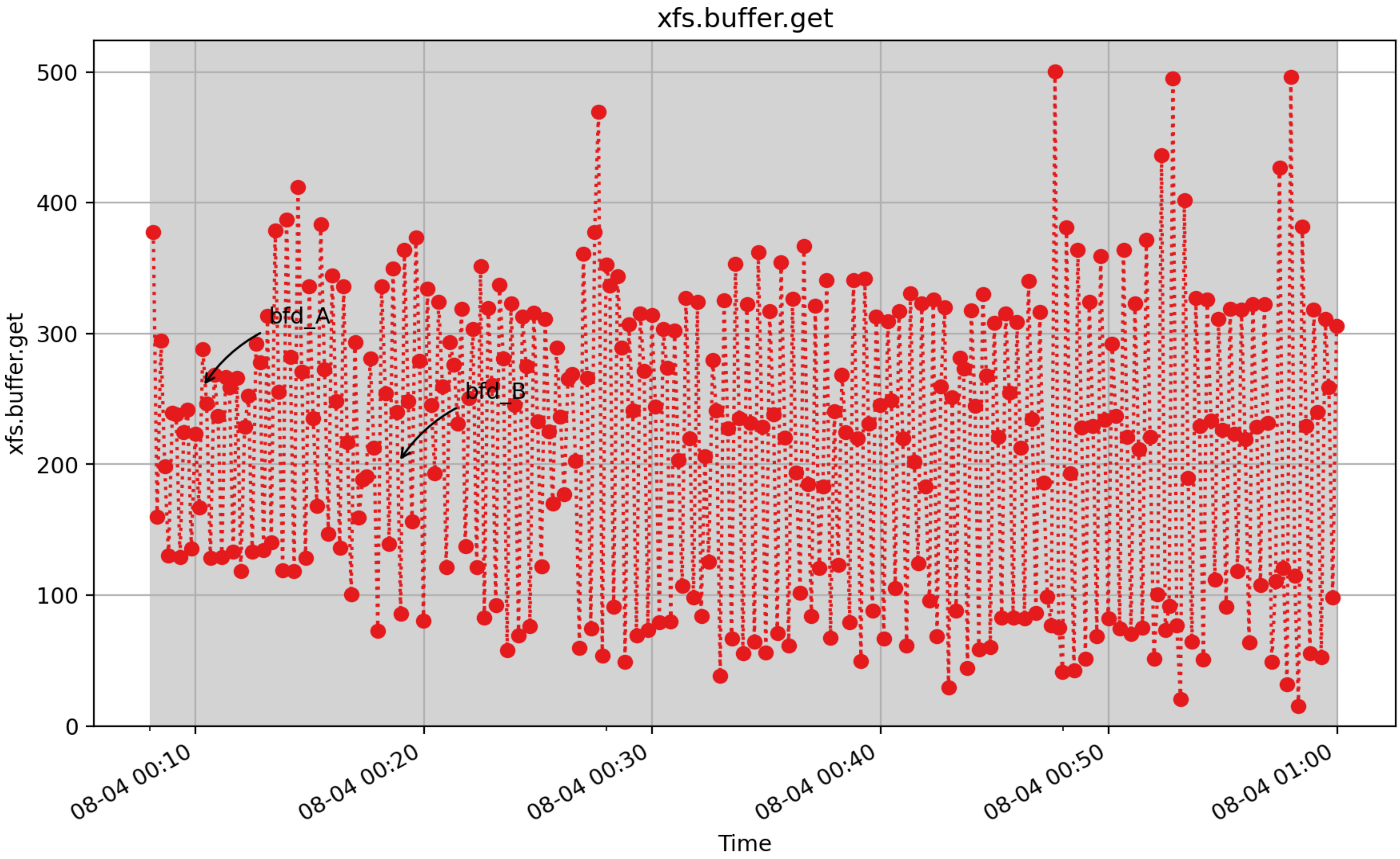
Xfs



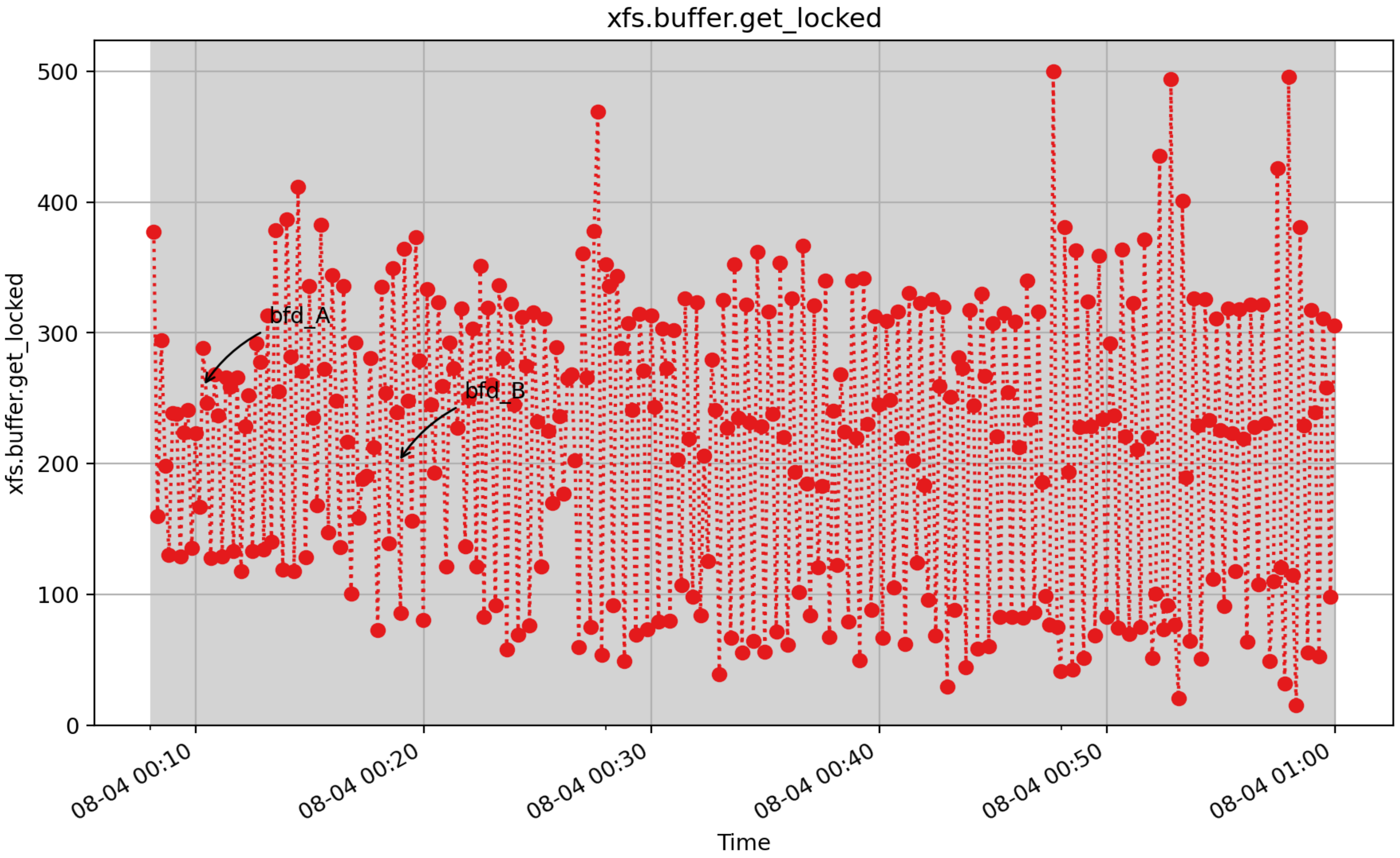
xfs.buffer.busy_locked: number of non-blocking requests for a locked buffer which failed (count - U32) - *rate converted*



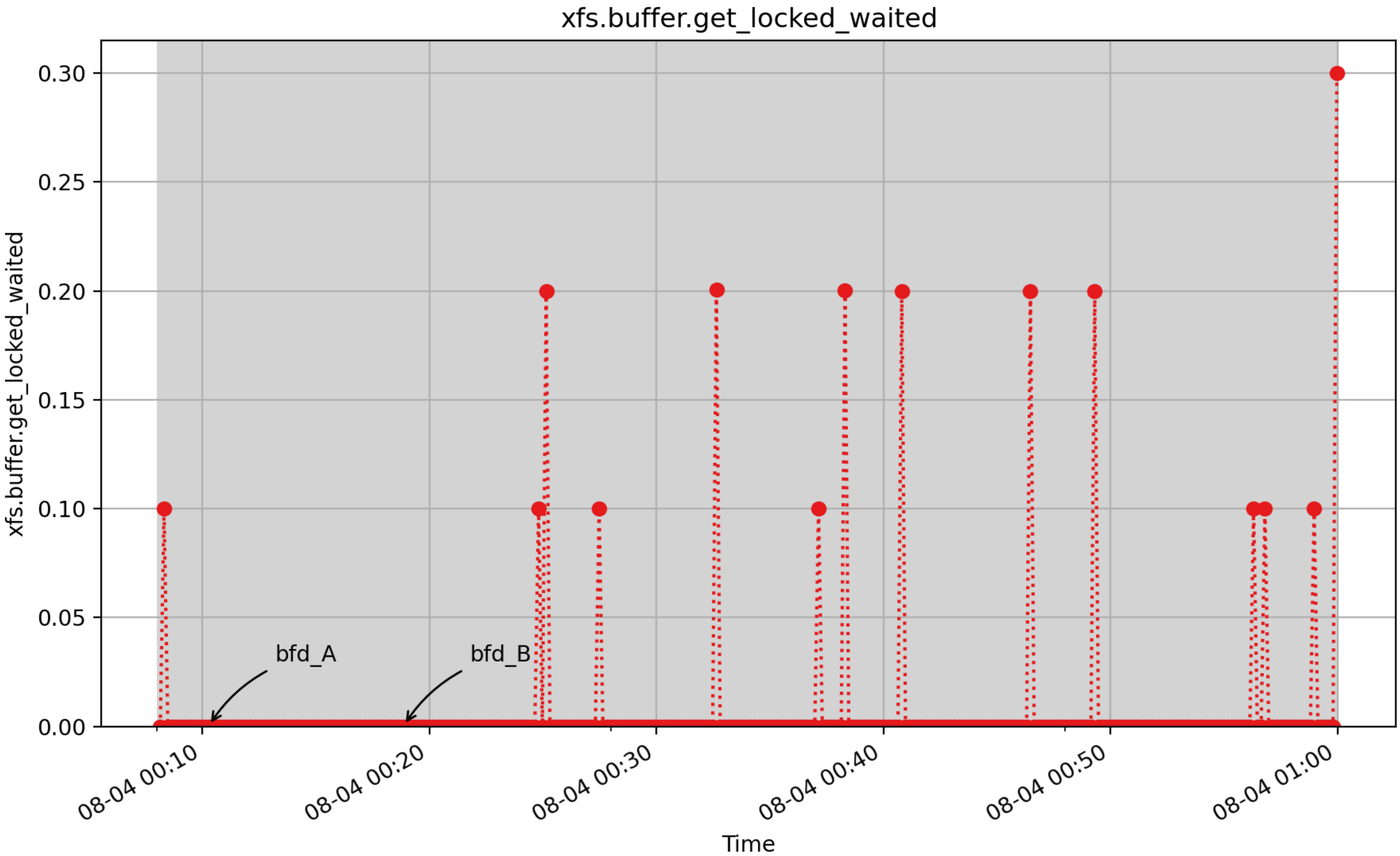
xfs.buffer.create: number of buffers created (count - U32) - *rate converted*



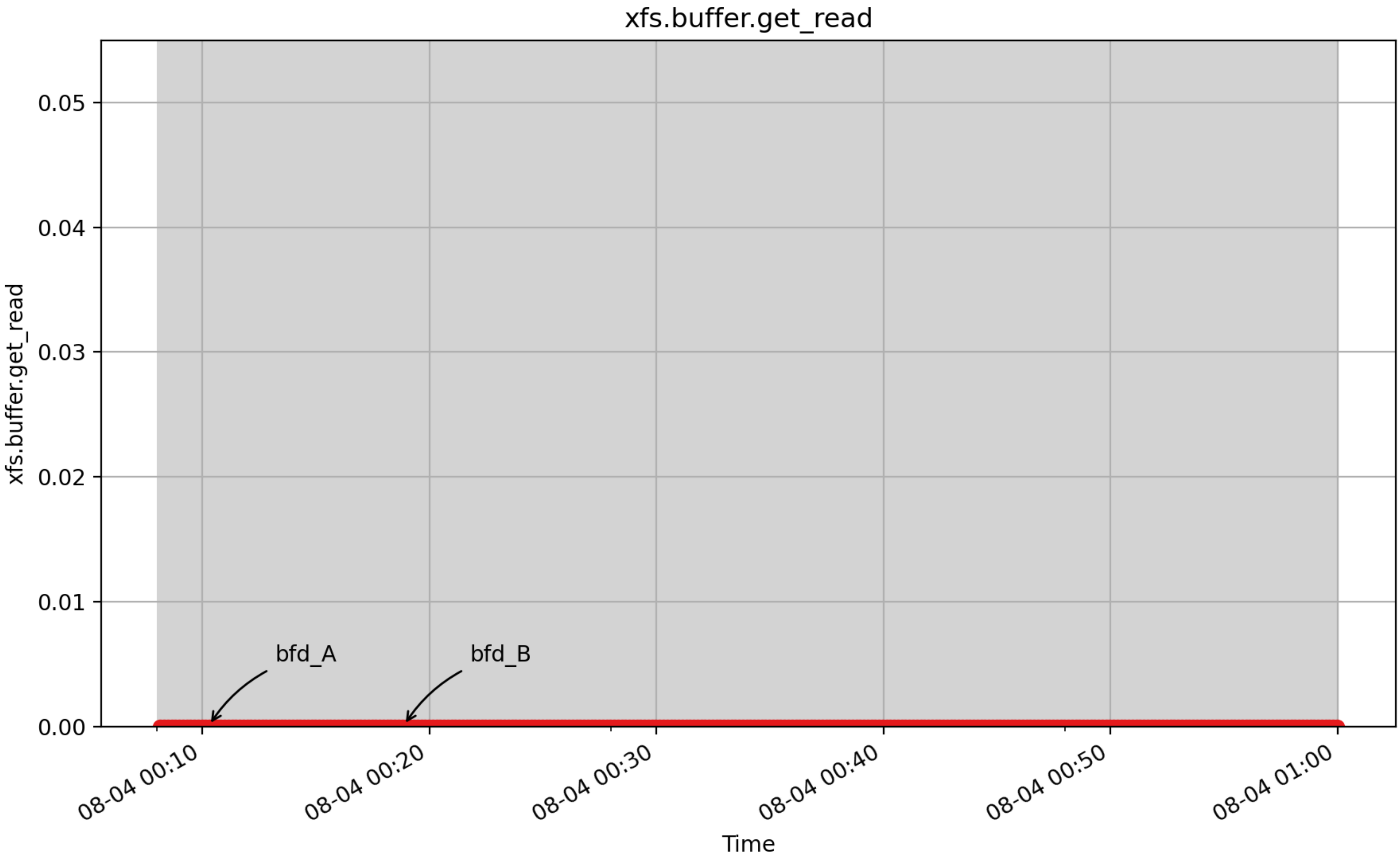
xfs.buffer.get: number of request buffer calls (count - U32) - *rate converted*



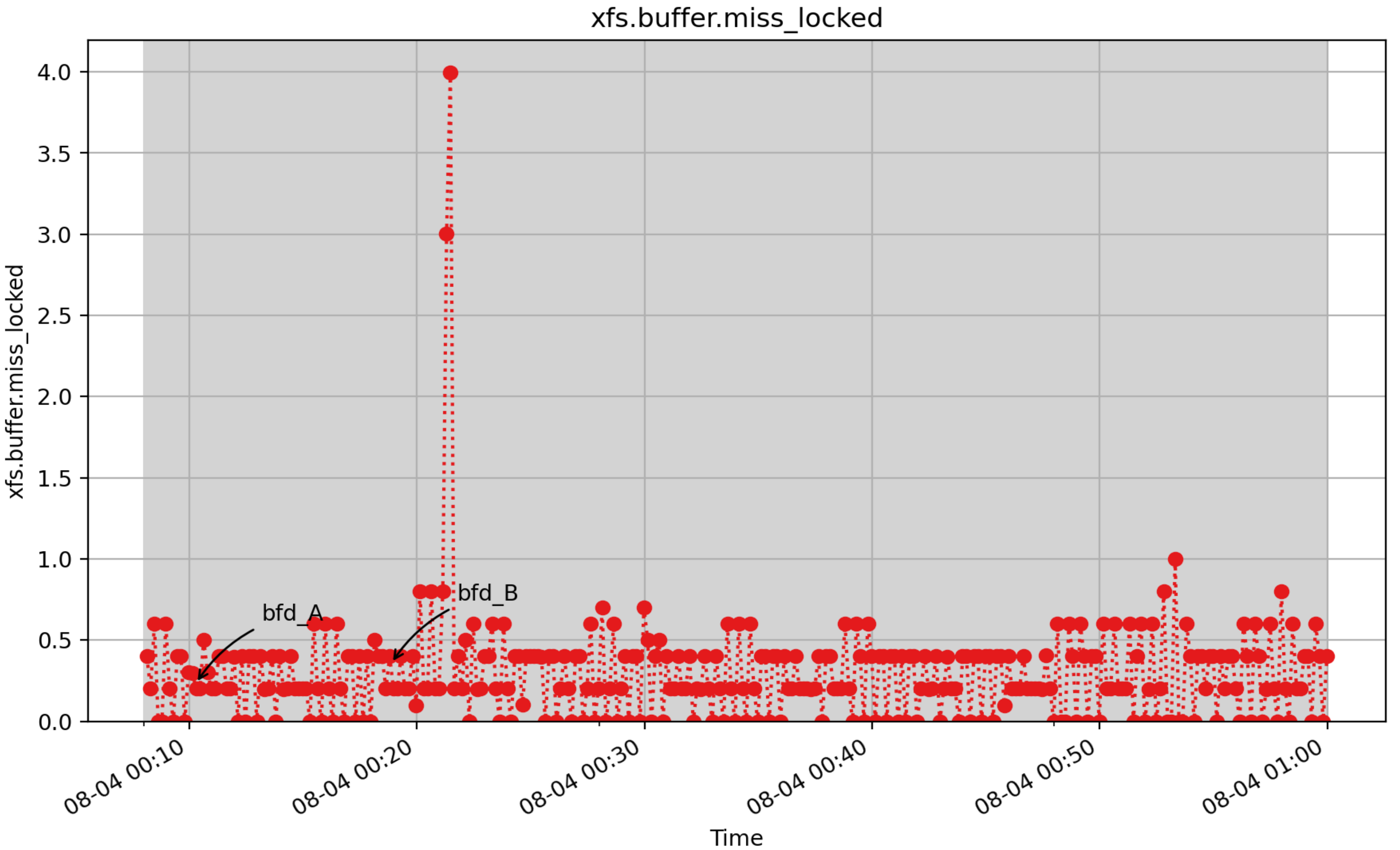
xfs.buffer.get_locked: number of requests for a locked buffer which succeeded (count - U32) - *rate converted*



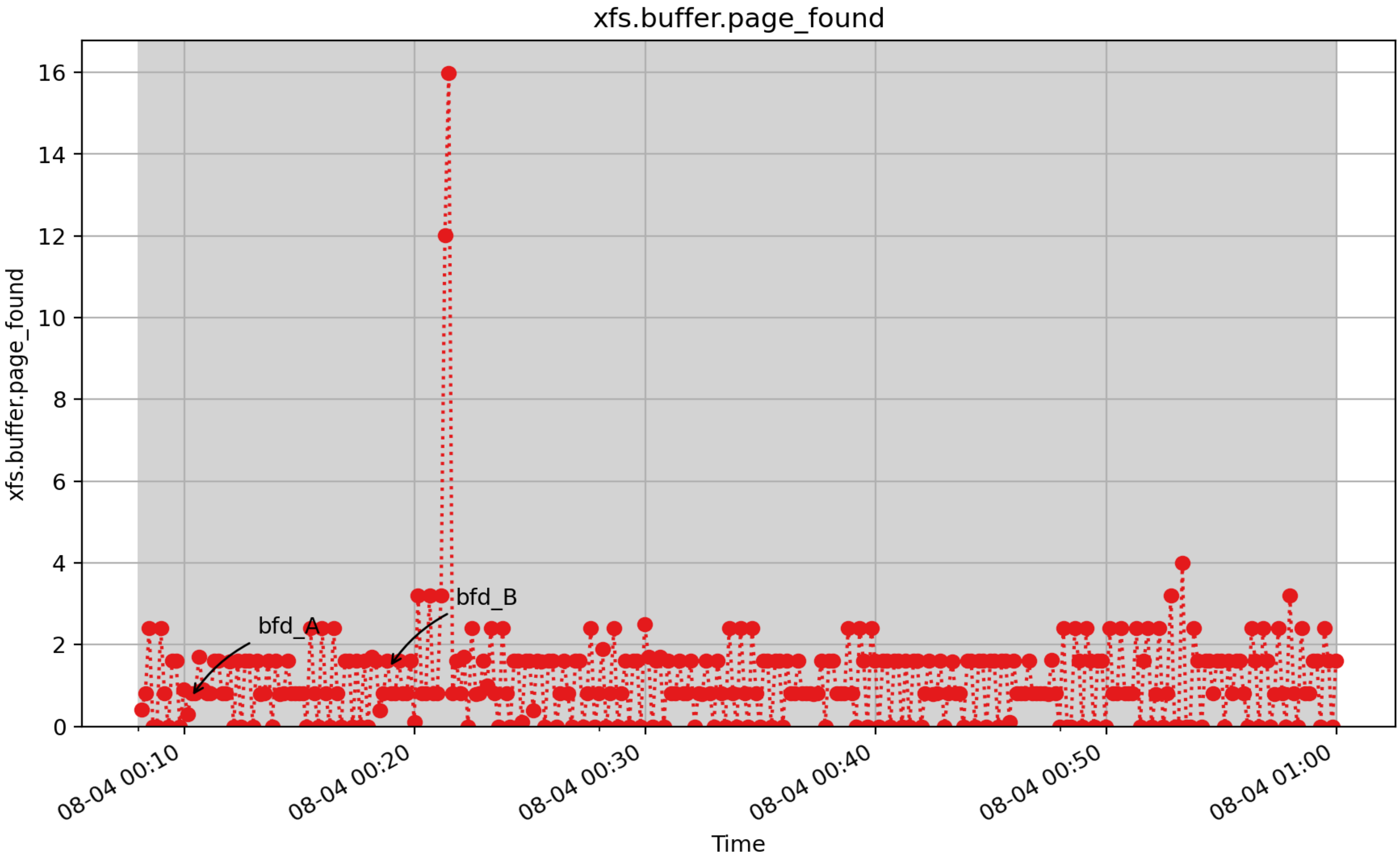
xfs.buffer.get_locked_waited: number of requests for a locked buffer which waited (count - U32) - *rate converted*



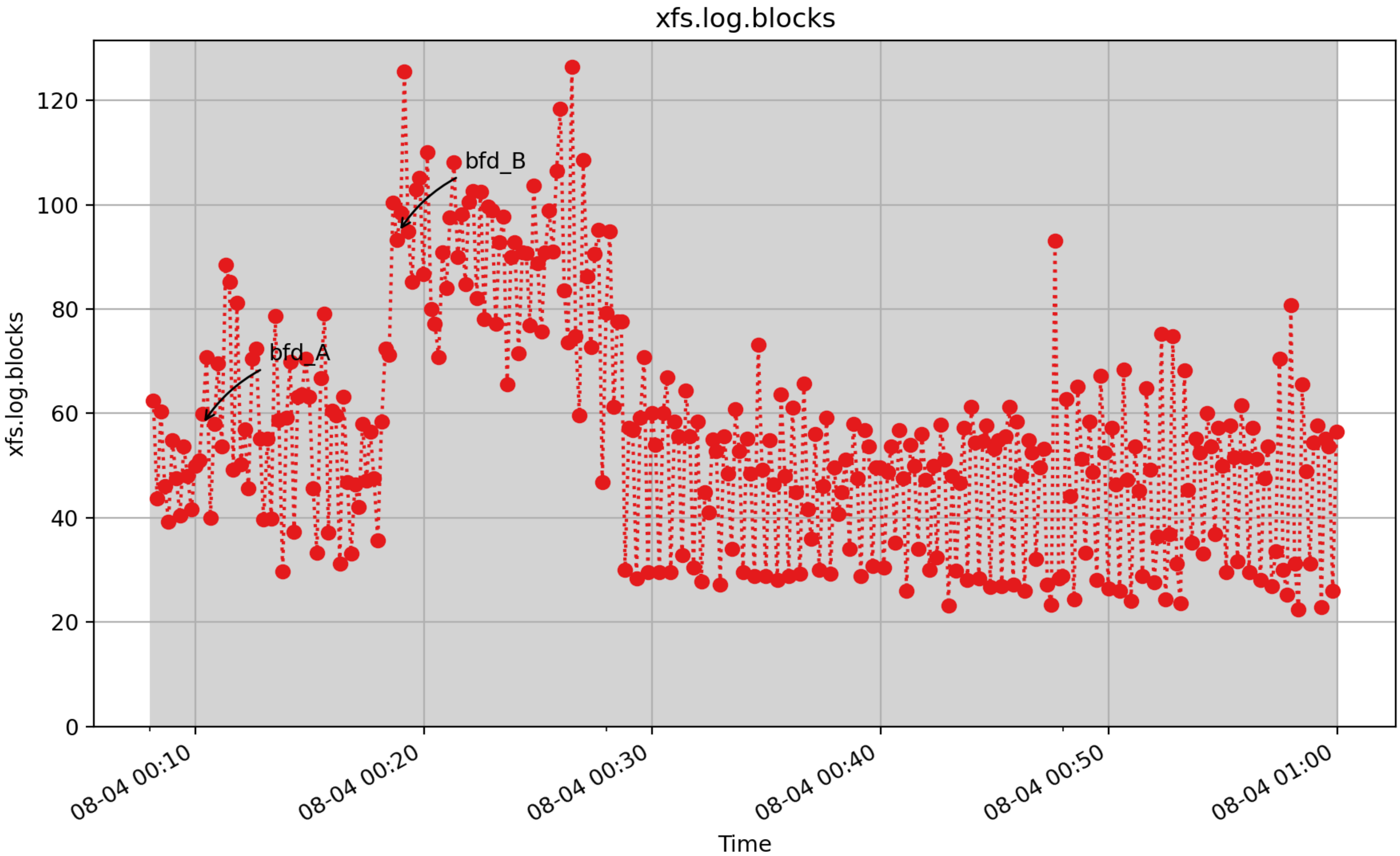
xfs.buffer.get_read: number of buffer get calls requiring immediate device reads (count - U32) - *rate converted*



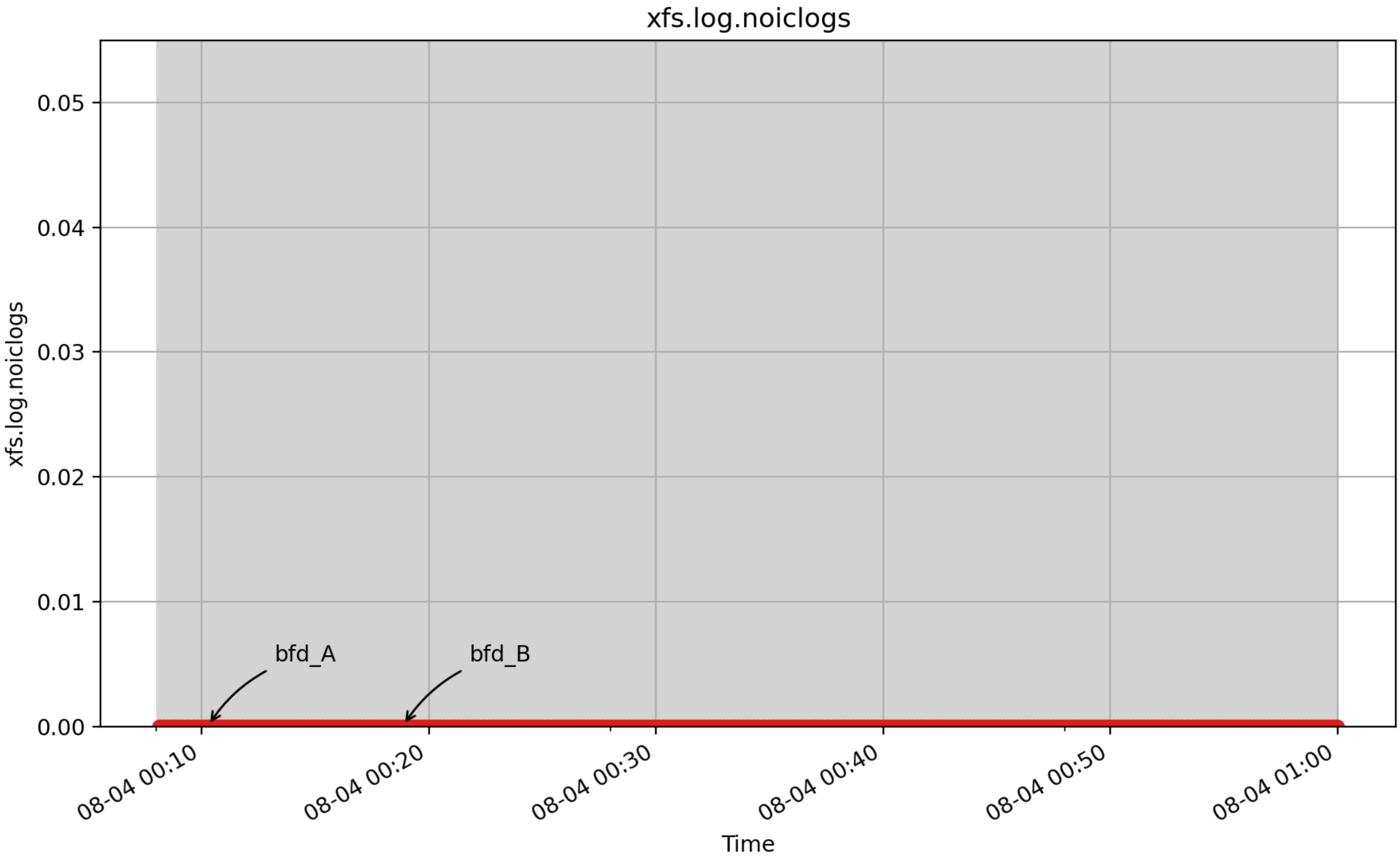
xfs.buffer.miss_locked: number of requests for a locked buffer which failed due to no buffer (count - U32) - *rate converted*



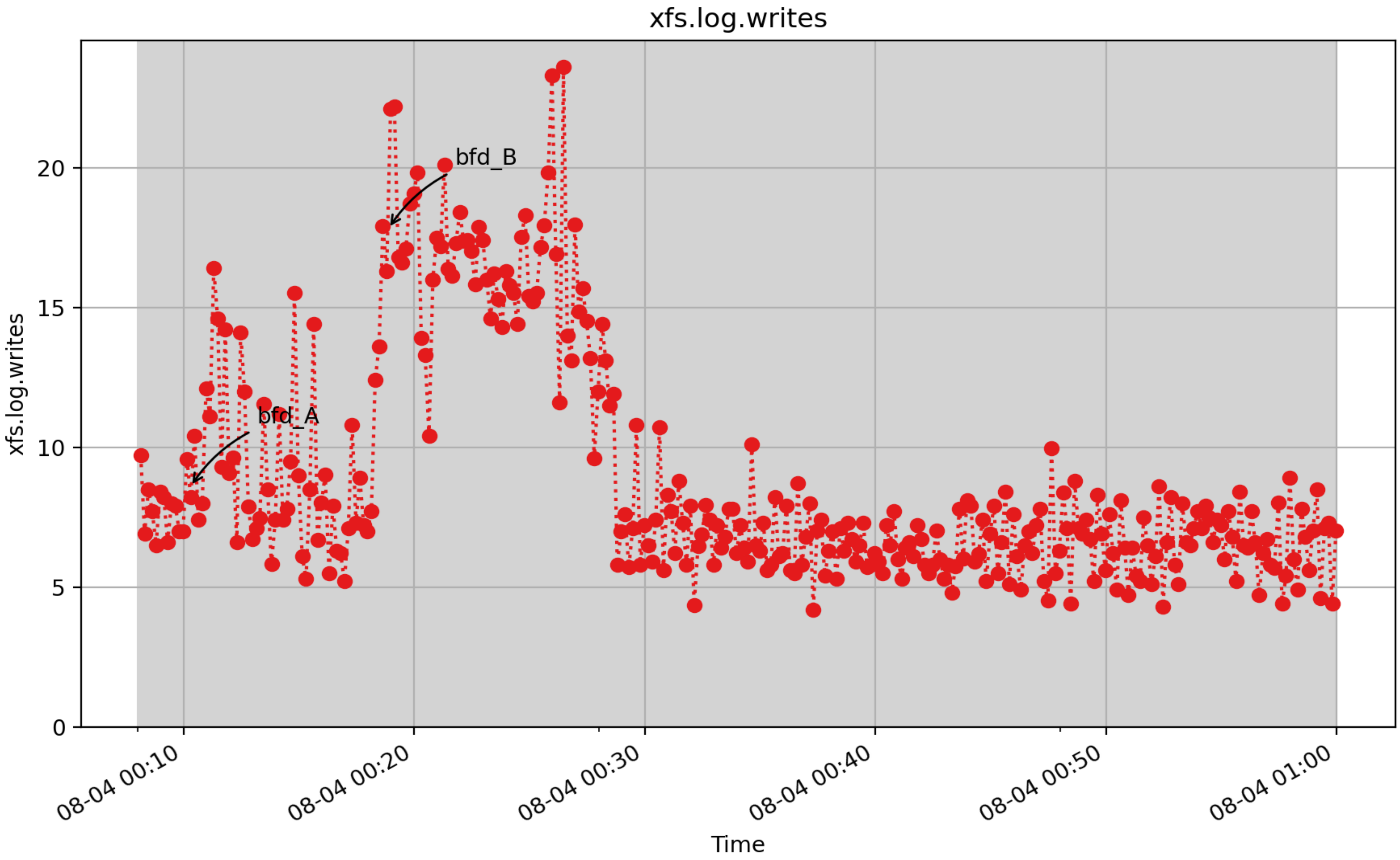
xfs.buffer.page_found: number of hits in the page cache when looking for a page (count - U32) - *rate converted*



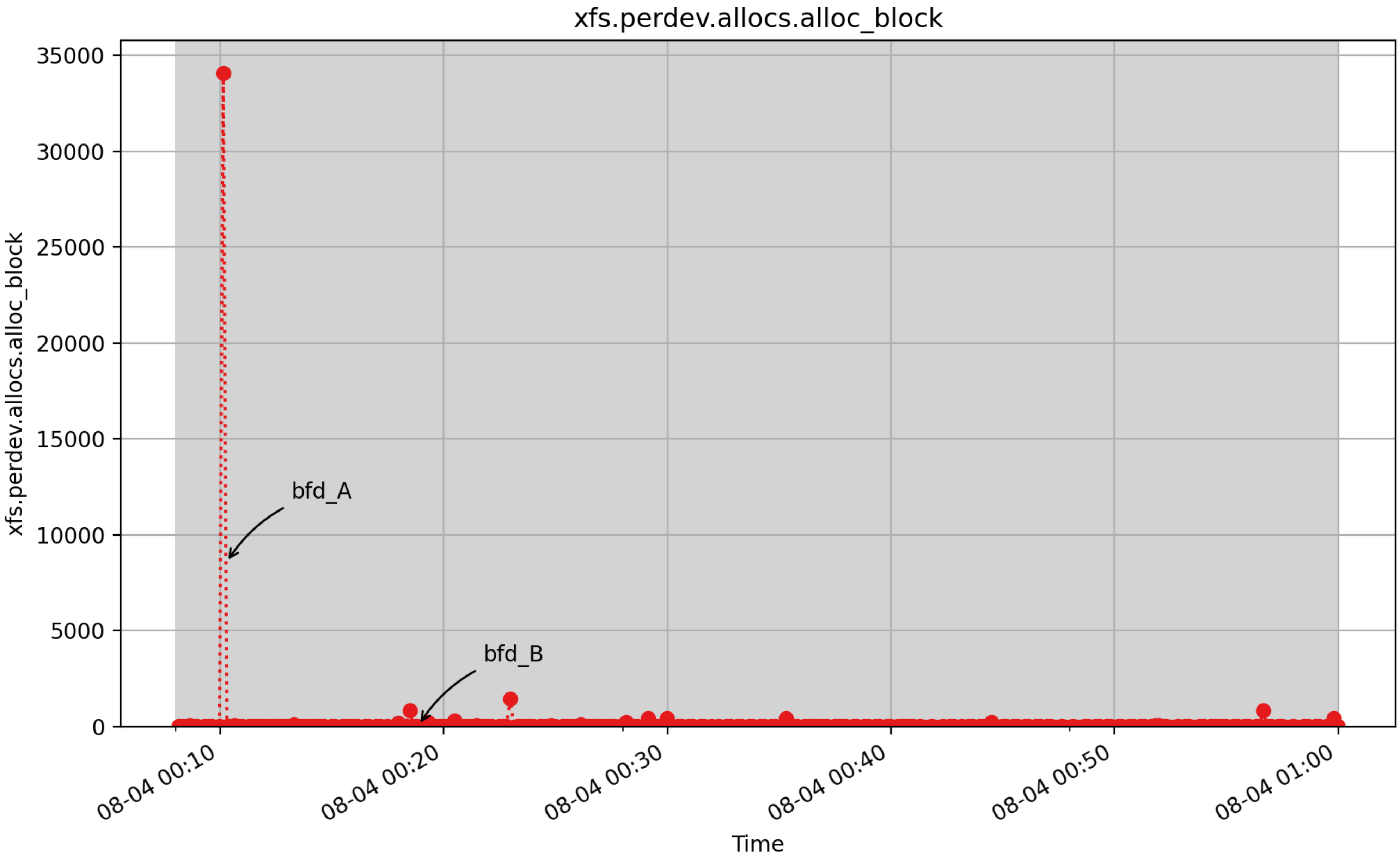
xfs.log.blocks: This variable counts the number of Kbytes of information being written to the physical log partitions of XFS filesystems. Log data traffic is proportional to the level of meta-data updating. The rate with which log data gets written depends on the size of internal log buffers and disk write speed. Therefore, filesystems with very high meta-data updating may need to stripe the log partition or put the log partition on a separate drive. (Kbyte - U32) - *rate converted*



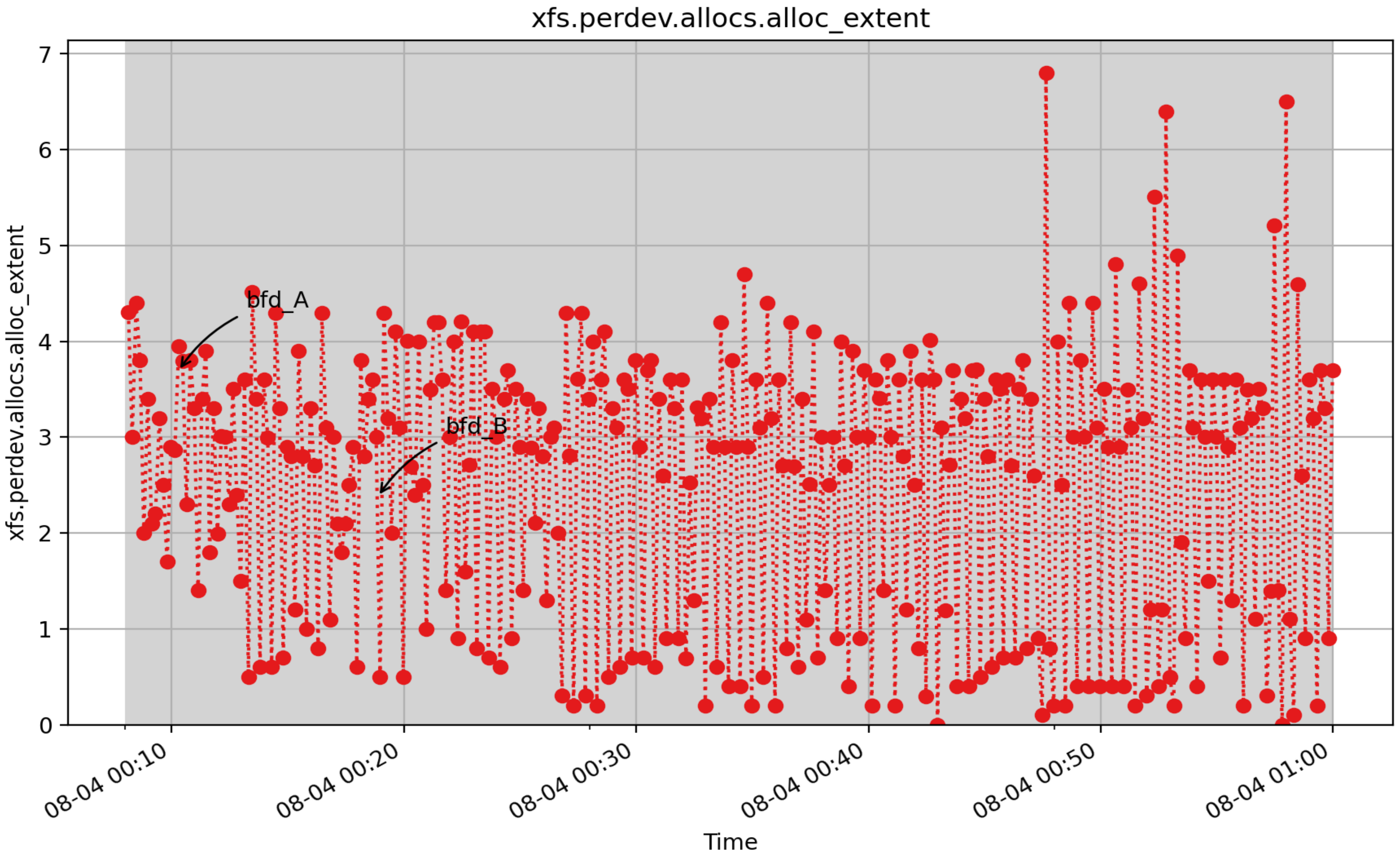
xfs.log.noiclogs: This variable keeps track of times when a logged transaction can not get any log buffer space. When this occurs, all of the internal log buffers are busy flushing their data to the physical on-disk log. (count - U32) - *rate converted*



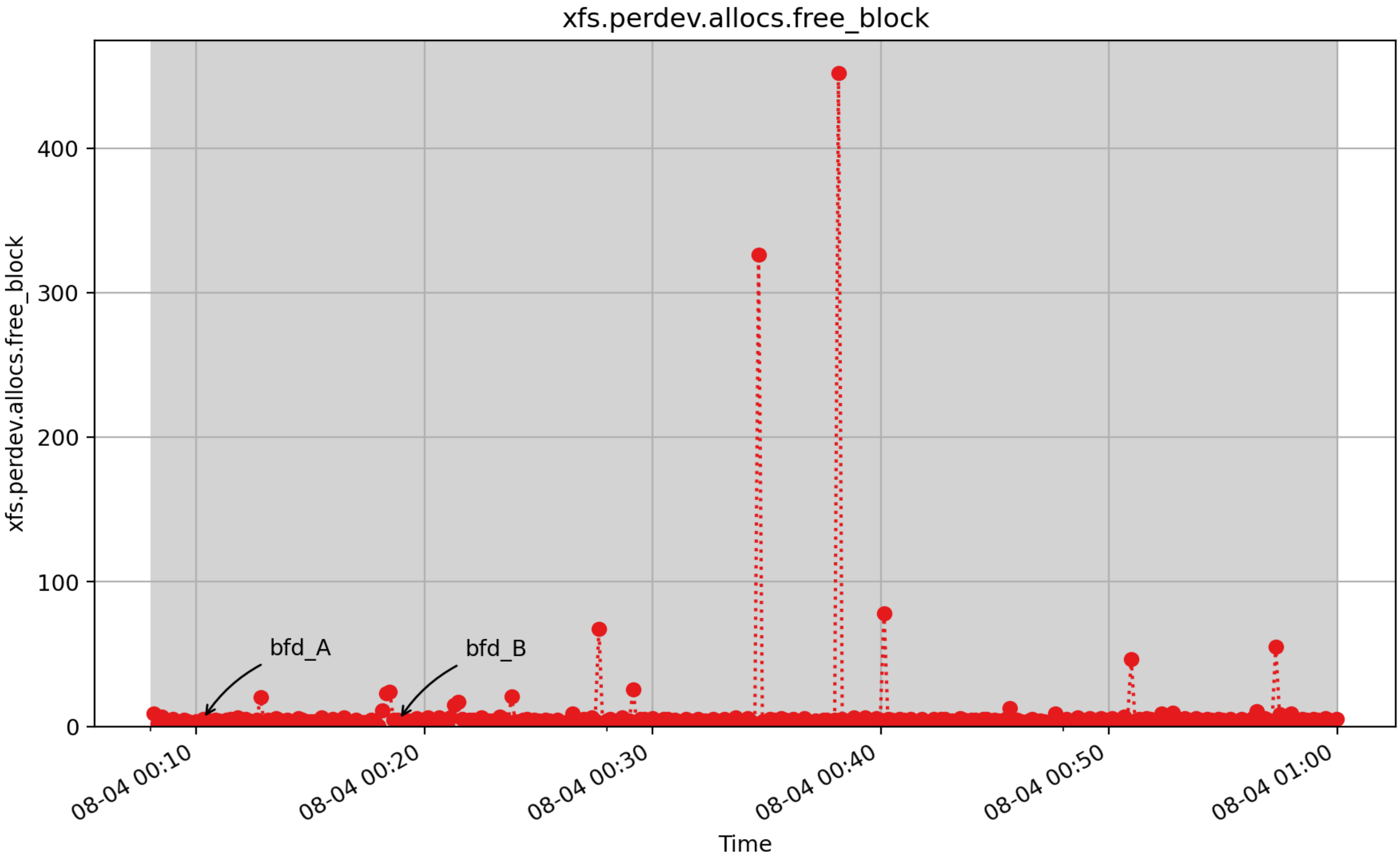
xfs.log.writes: This variable counts the number of log buffer writes going to the physical log partitions of XFS filesystems. Log data traffic is proportional to the level of meta-data updating. Log buffer writes get generated when they fill up or external syncs occur. (count - U32) - *rate converted*



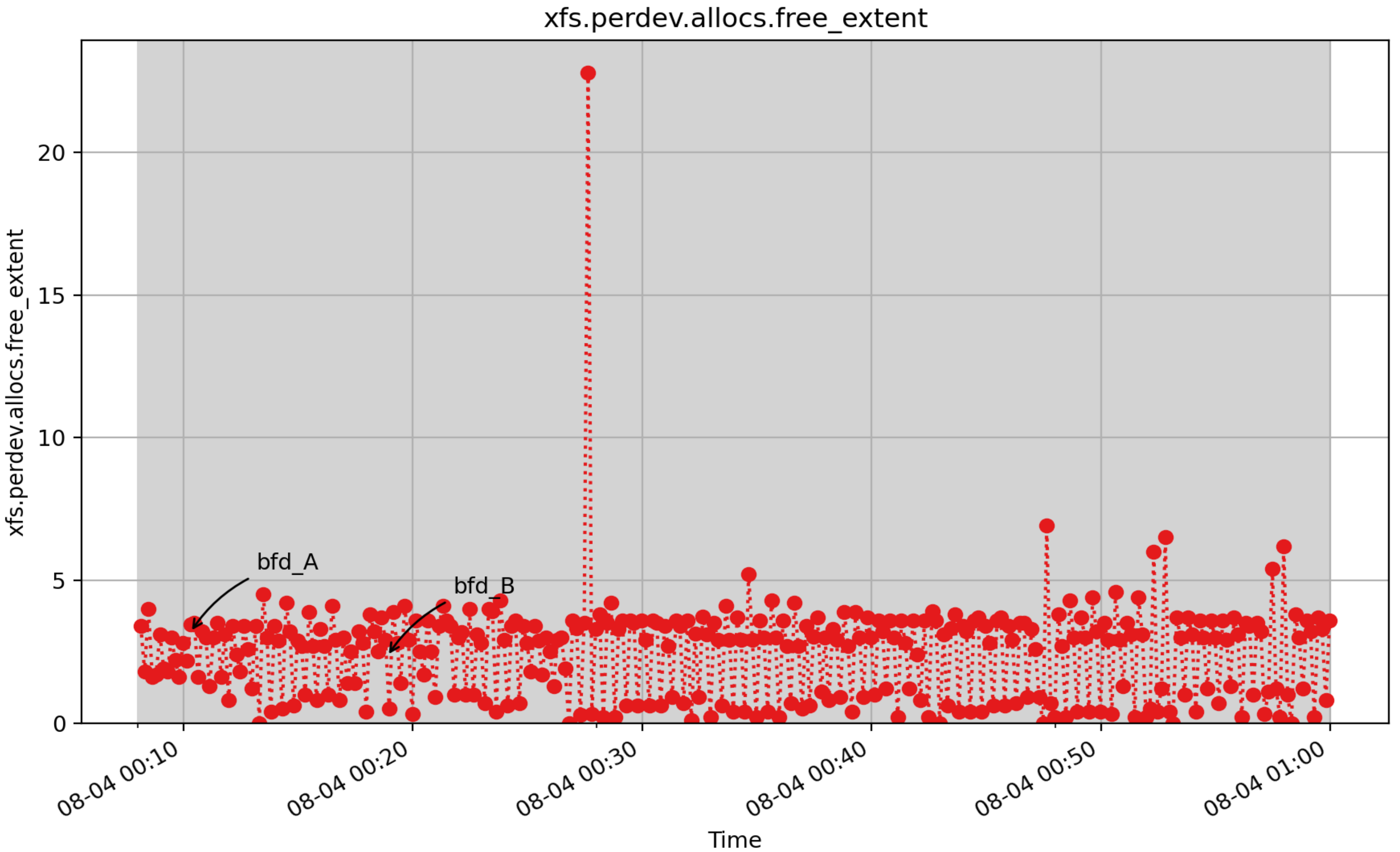
xfs.perdev.allocs.alloc_block: Number of file system blocks allocated over XFS filesystems (count - U32) - *rate converted*



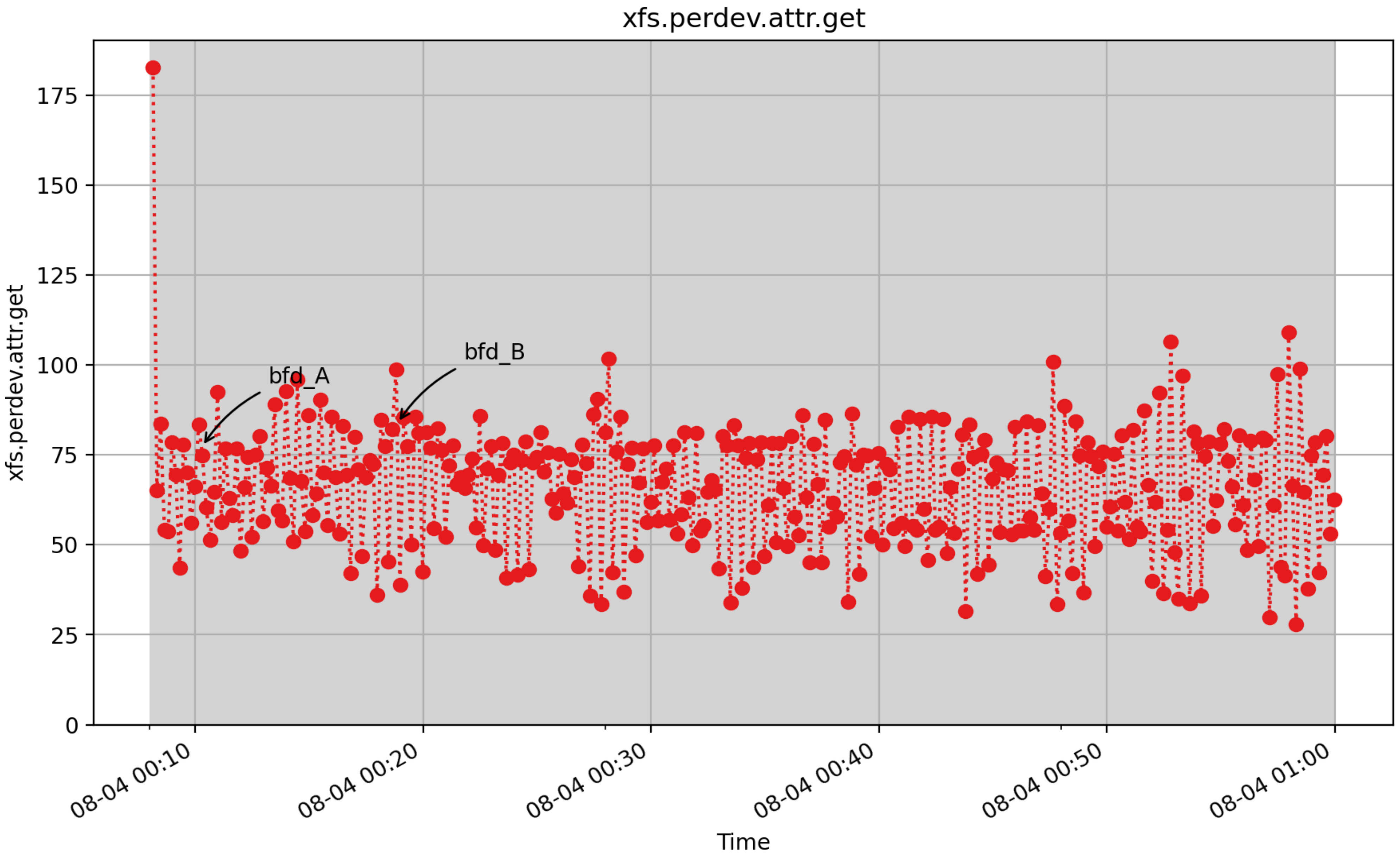
xfs.perdev.allocs.alloc_extent: Number of file system extents allocated over XFS filesystems (count - U32) - *rate converted*



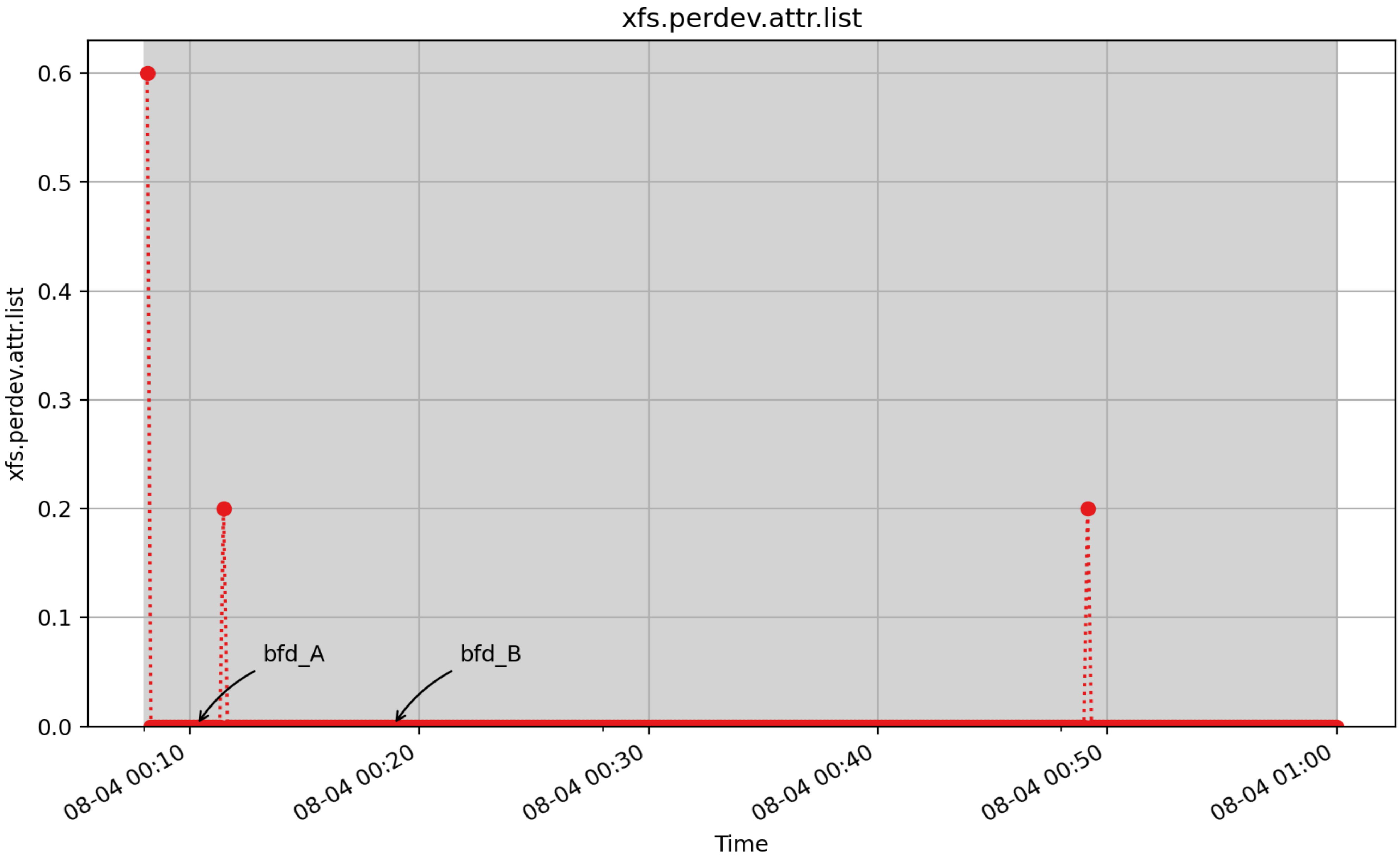
xfs.perdev.allocs.free_block: Number of file system blocks freed over XFS filesystems (count - U32) - *rate converted*



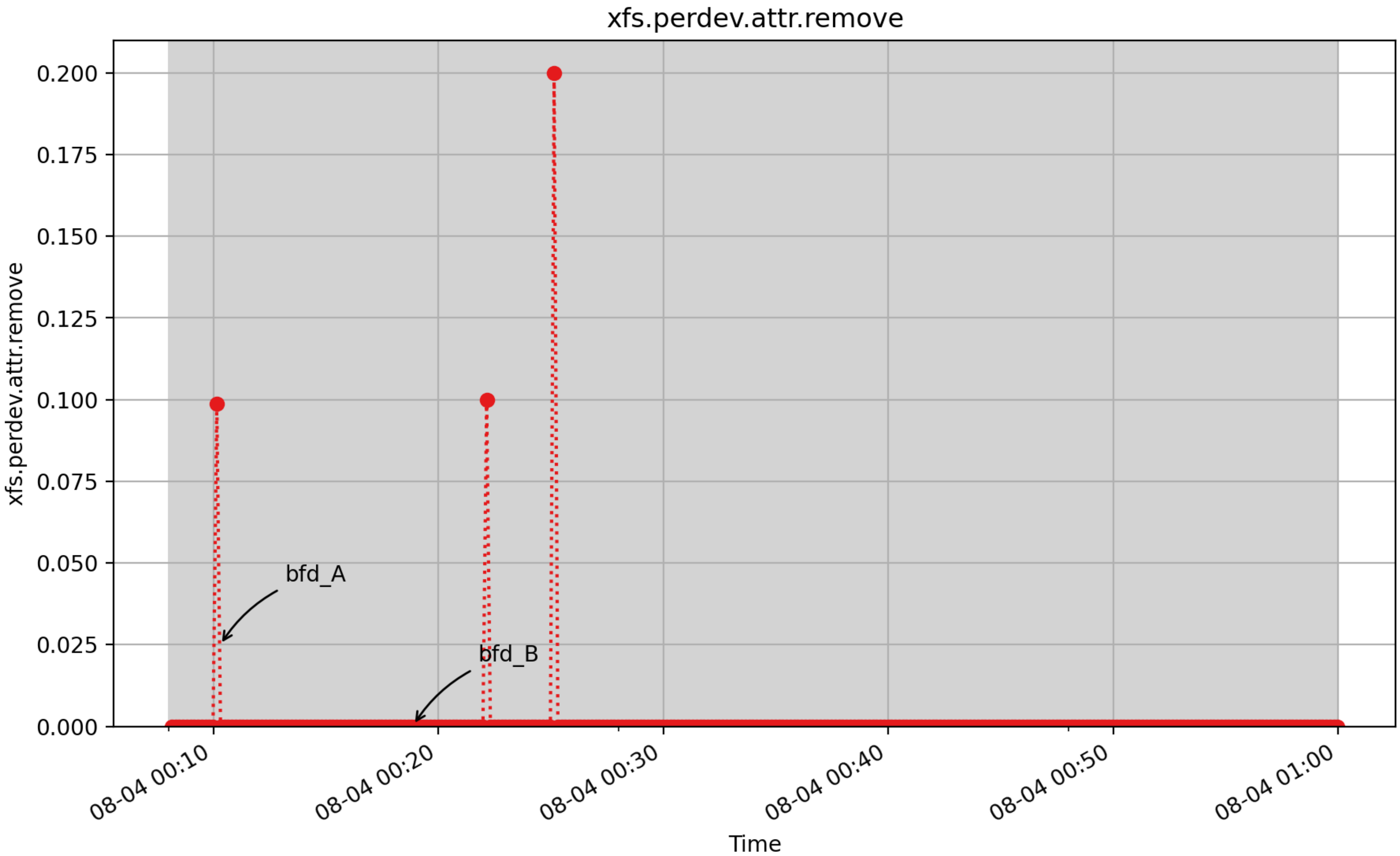
xfs.perdev.allocs.free_extent: Number of file system extents freed over XFS filesystems (count - U32) - *rate converted*



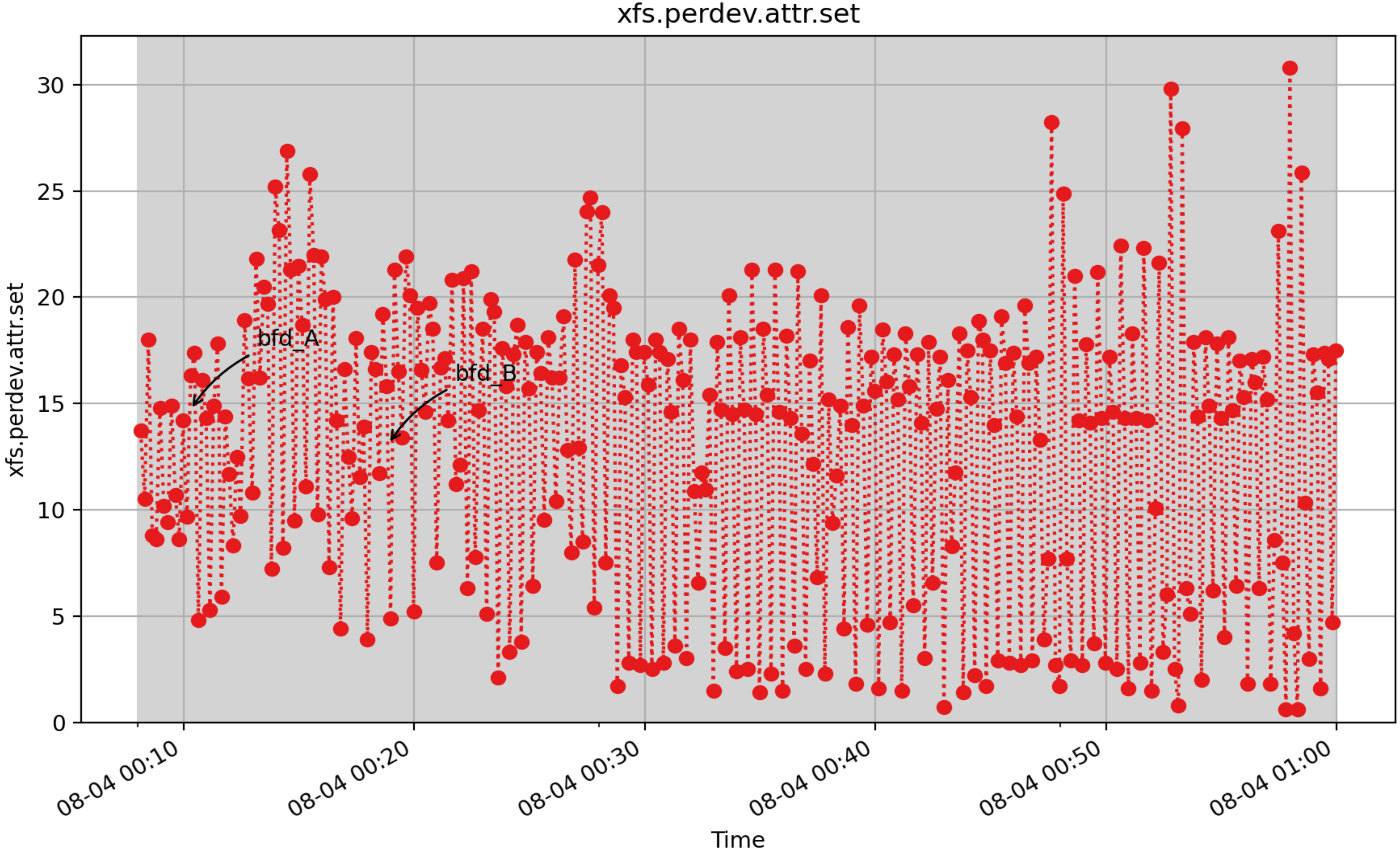
xfs.perdev.attr.get: The number of "get" operations performed on extended file attributes within XFS filesystems. The "get" operation retrieves the value of an extended attribute. (count - U32) - *rate converted*



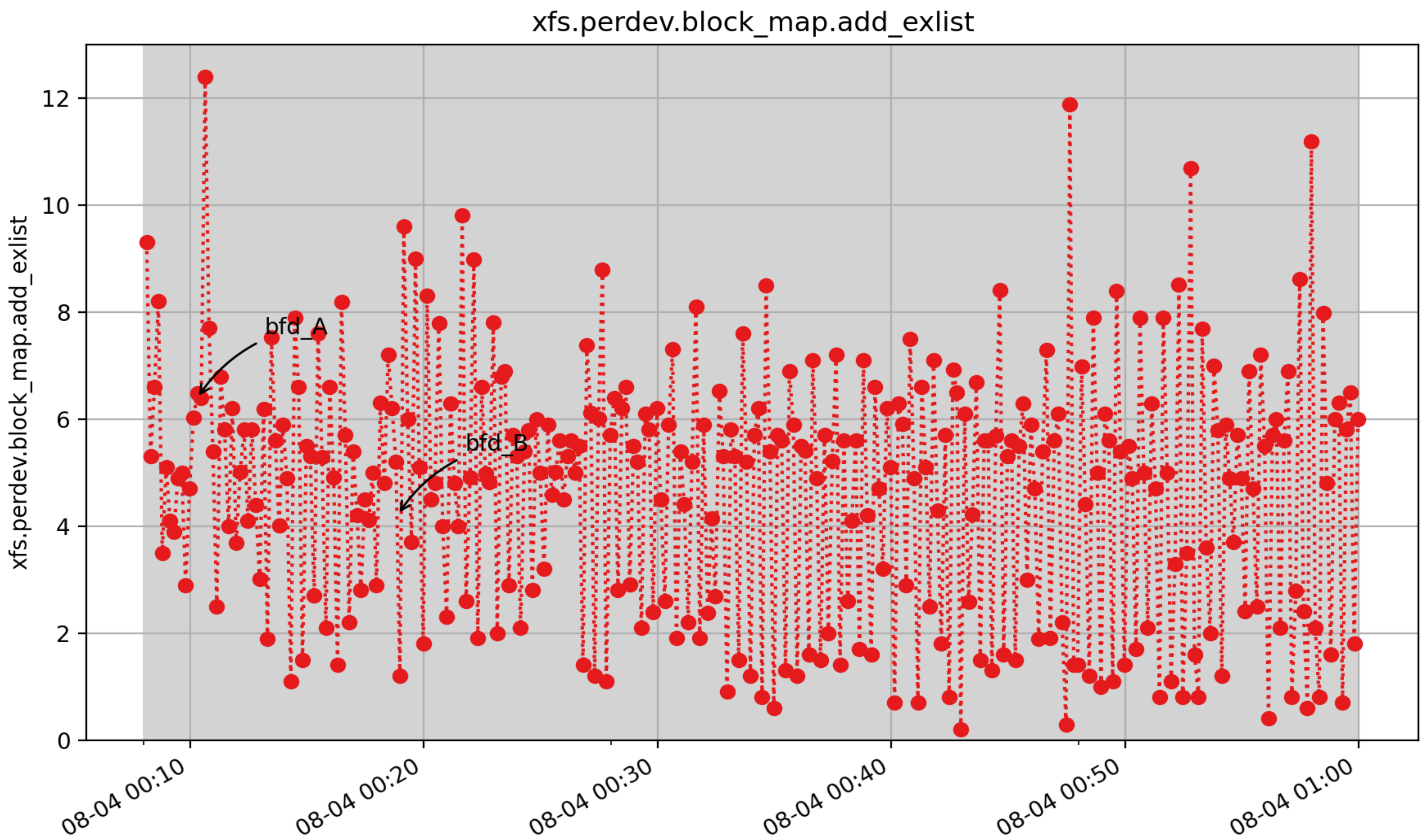
xfs.perdev.attr.list: The number of "list" operations performed on extended file attributes within XFS filesystems. The "list" operation retrieves the set of extended attributes associated with a file. (count - U32) - *rate converted*



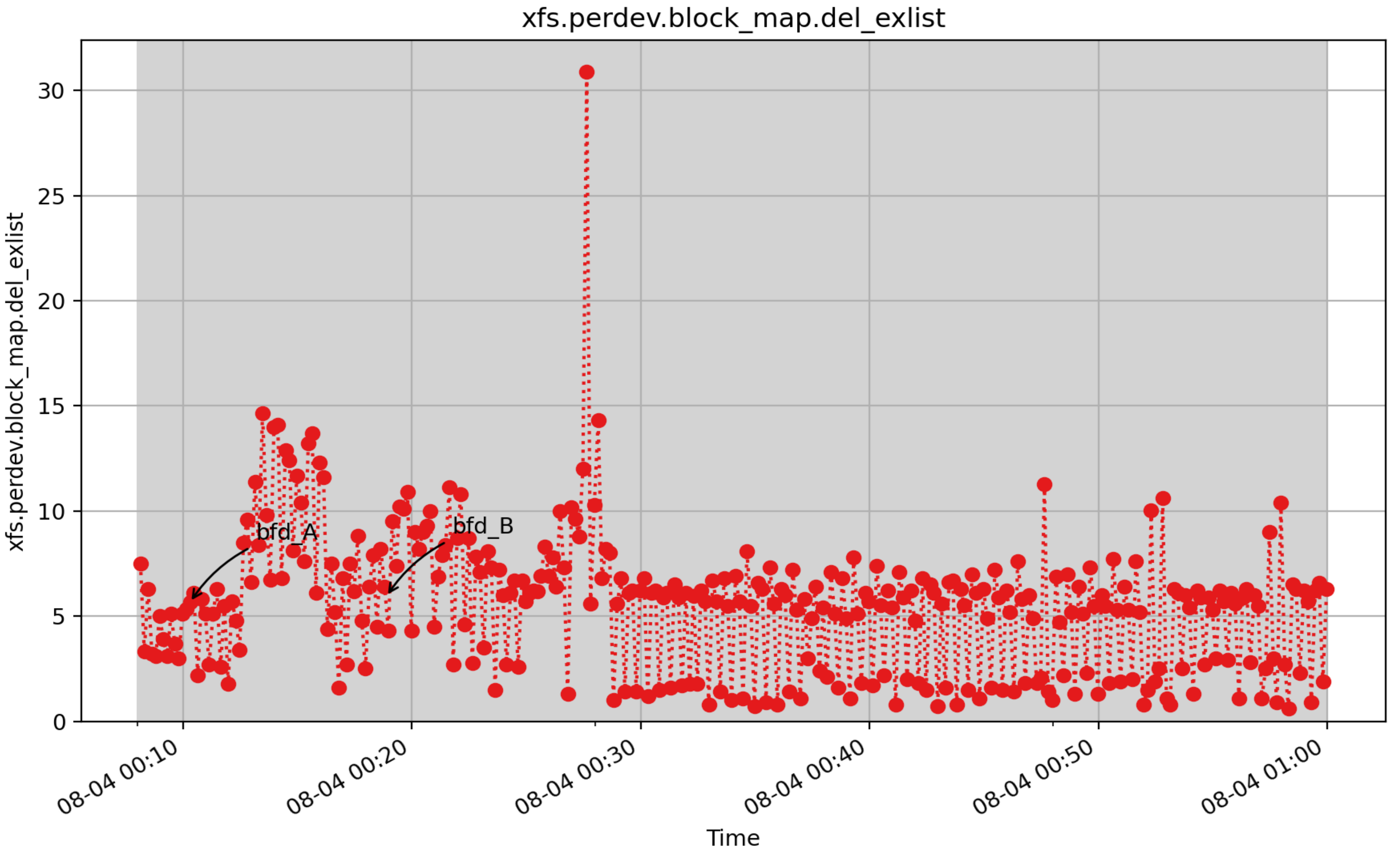
xfs.perdev.attr.remove: The number of "remove" operations performed on extended file attributes within XFS filesystems. The "remove" operation deletes an extended attribute. (count - U32) - *rate converted*



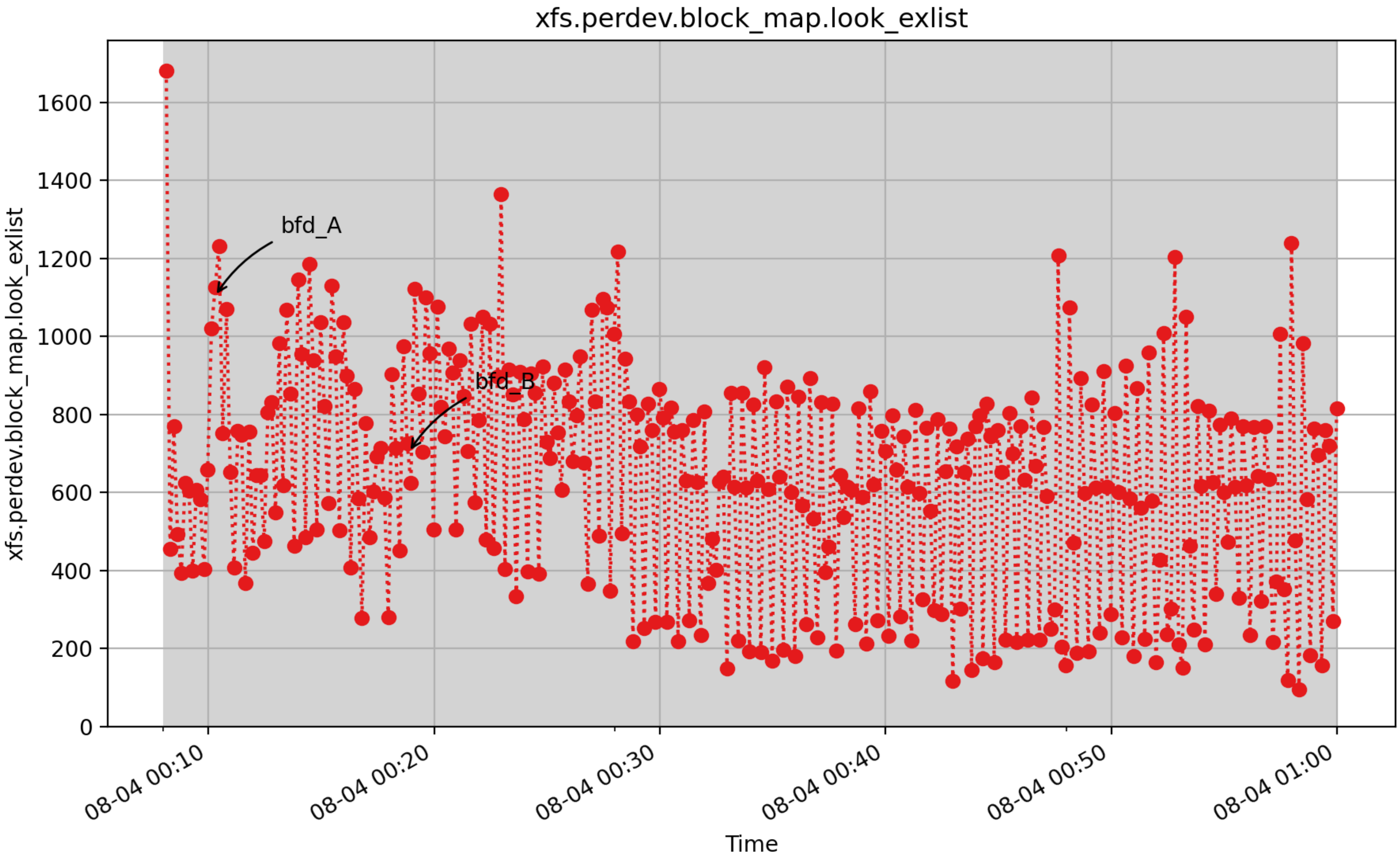
xfs.perdev.attr.set: The number of "set" operations performed on extended file attributes within XFS filesystems. The "set" operation creates and sets the value of an extended attribute. (count - U32) - *rate converted*



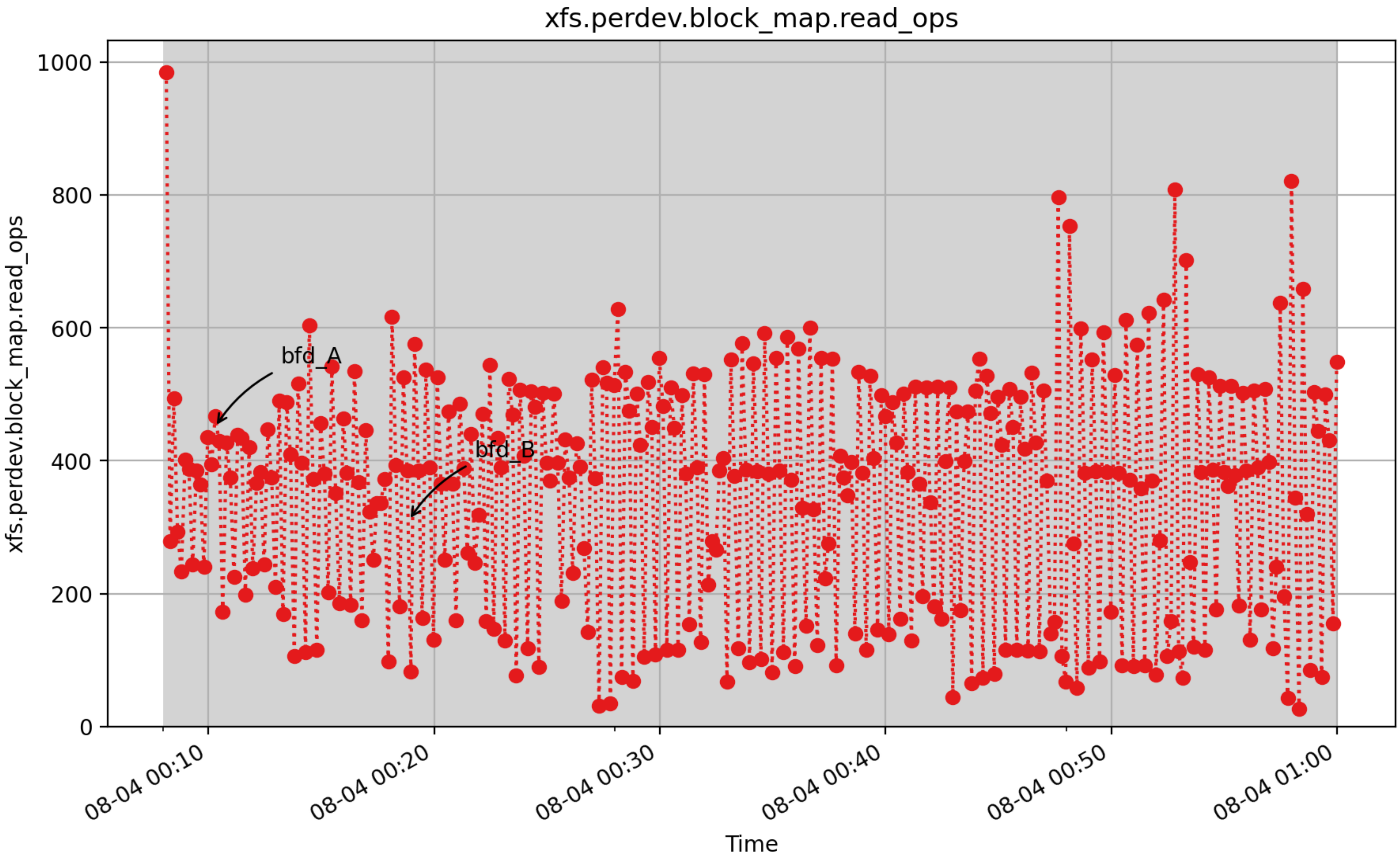
xfs.perdev.block_map.add_exlist: Number of extent list insertion operations for XFS files (count - U32) - *rate converted*



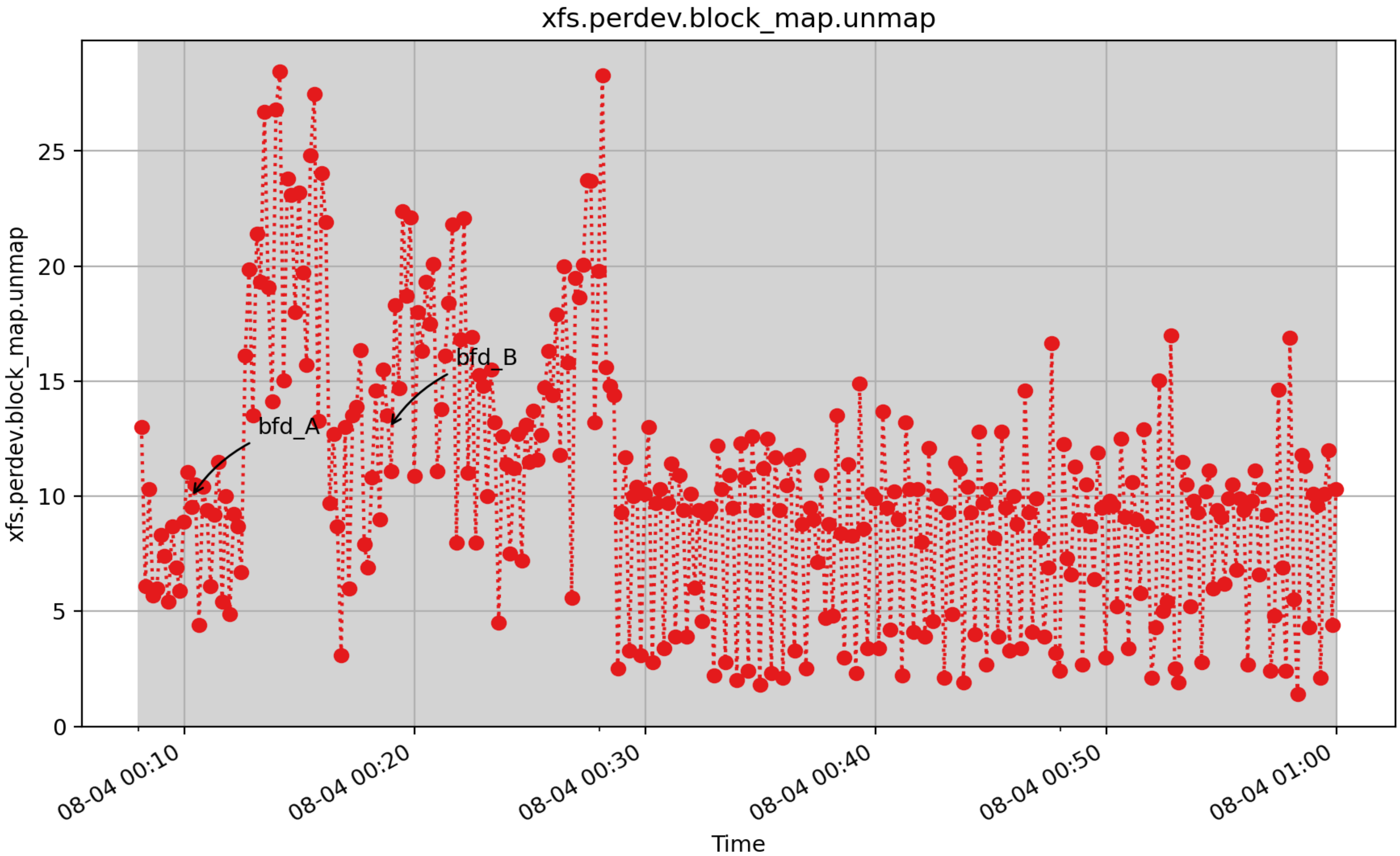
xfs.perdev.block_map.del_exlist: Number of extent list deletion operations for XFS files (count - U32) - *rate converted*



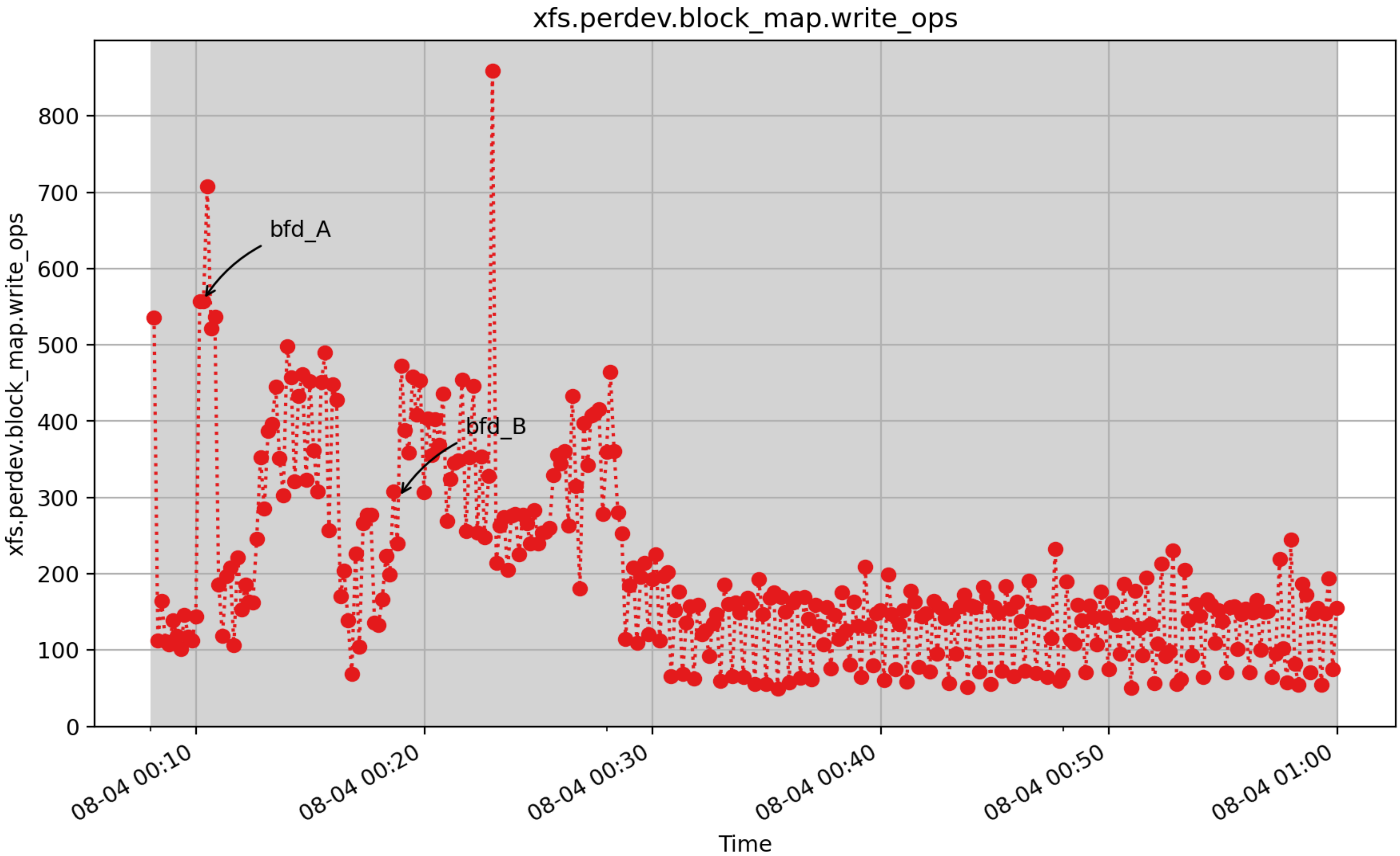
xfs.perdev.block_map.look_exlist: Number of extent list lookup operations for XFS files (count - U32) - *rate converted*



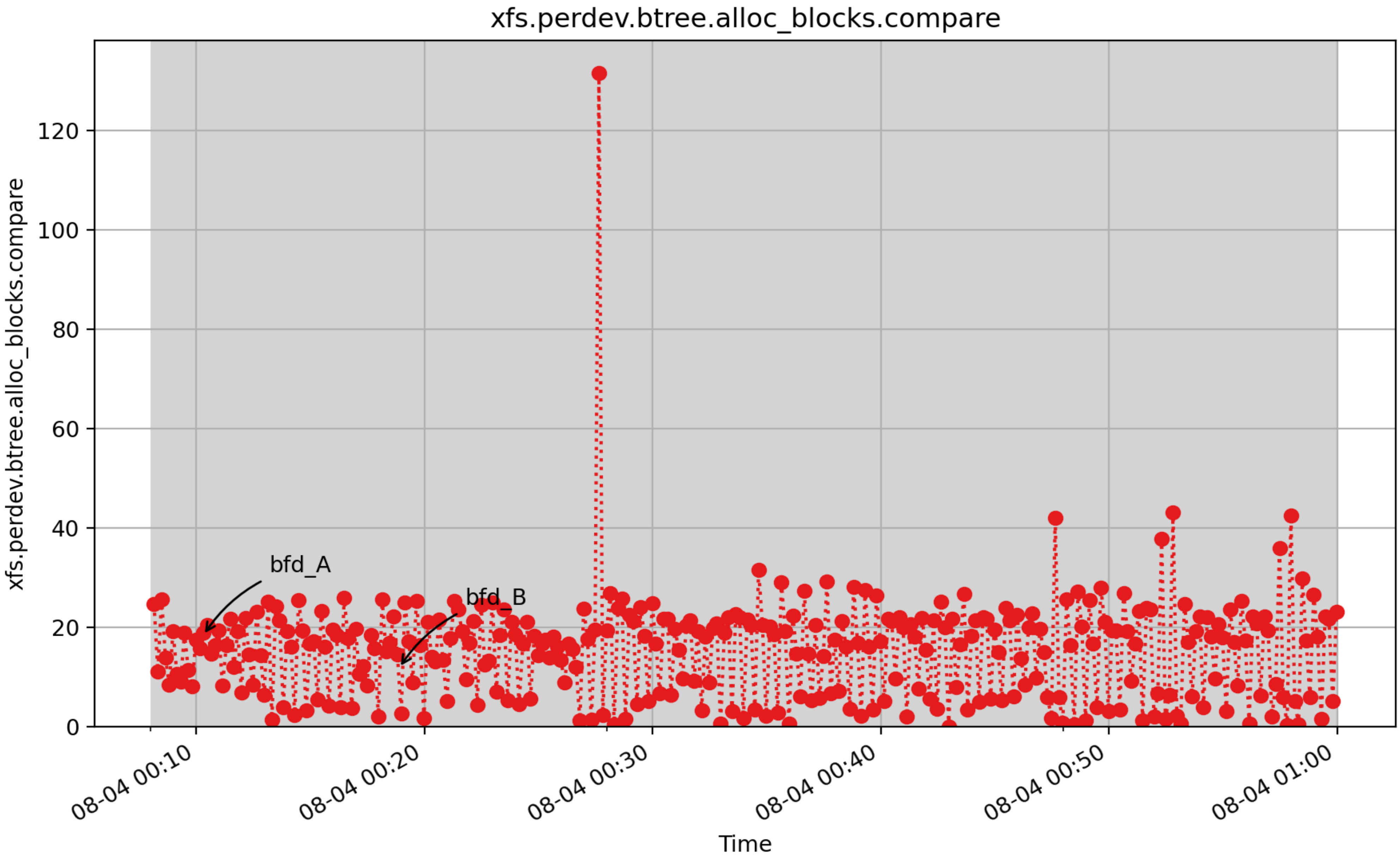
xfs.perdev.block_map.read_ops: Number of block map for read operations performed on XFS files (count - U32) - *rate converted*



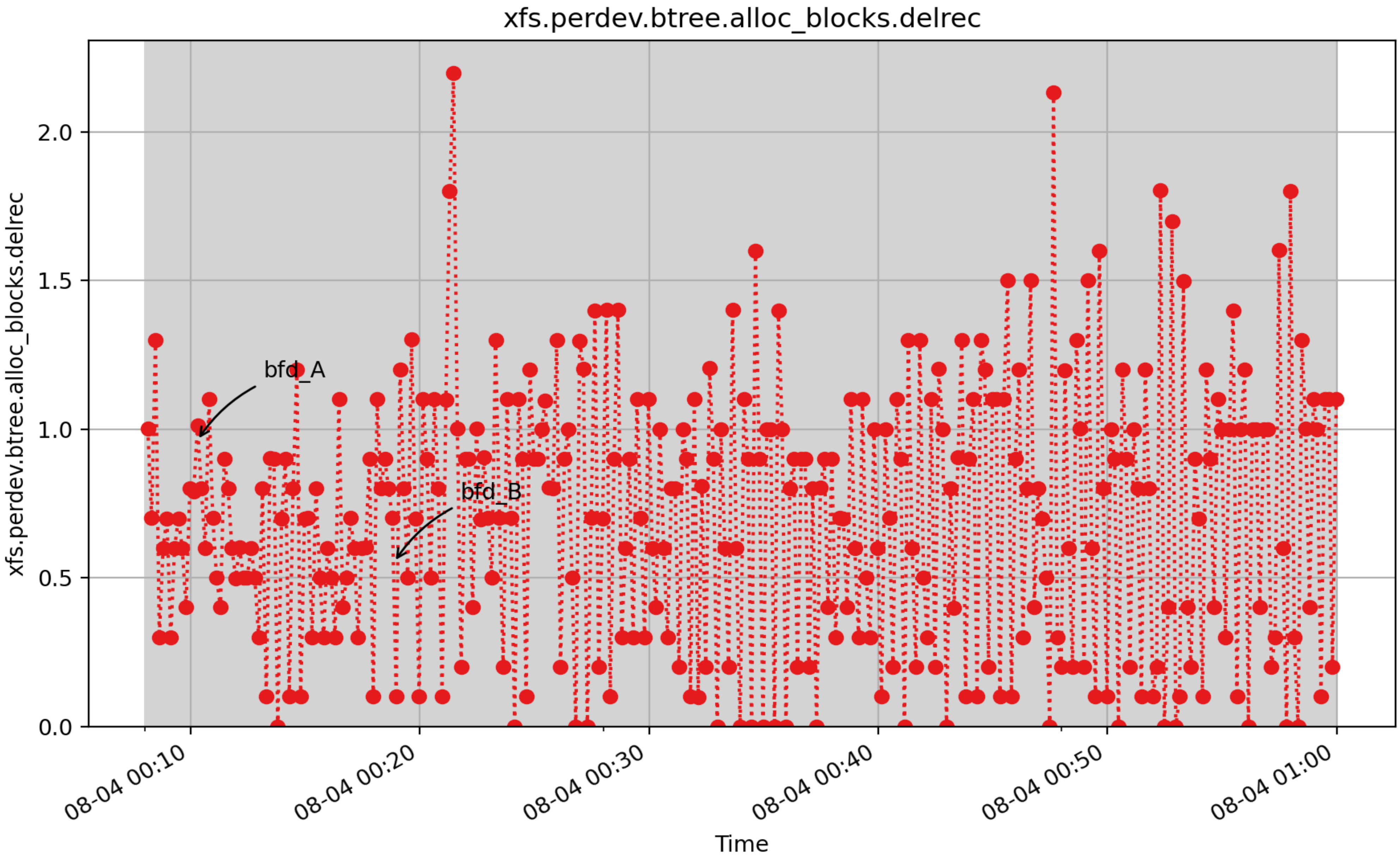
xfs.perdev.block_map.unmap: Number of block unmap (delete) operations performed on XFS files (count - U32) - *rate converted*



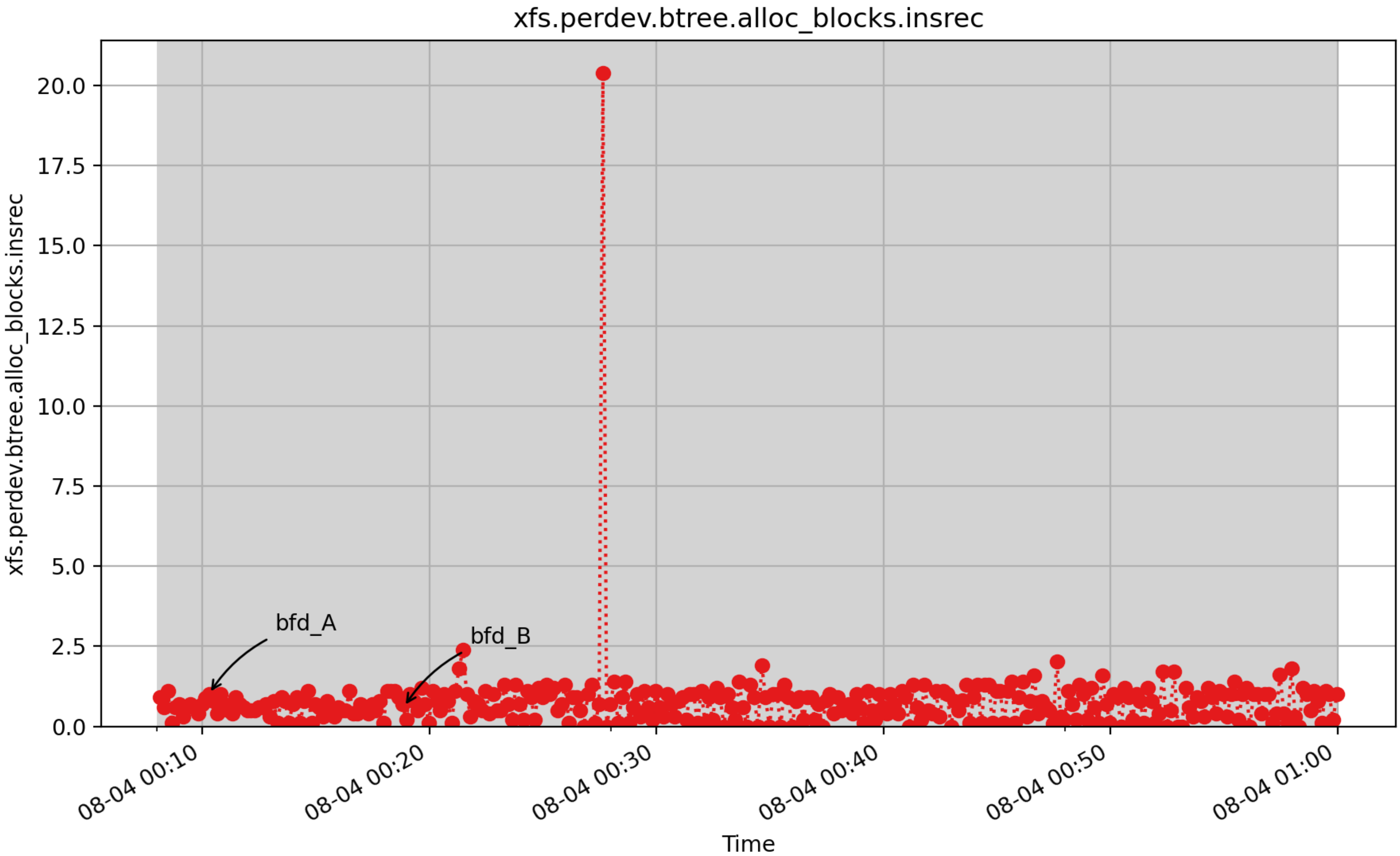
xfs.perdev.block_map.write_ops: Number of block map for write operations performed on XFS files (count - U32) - *rate converted*



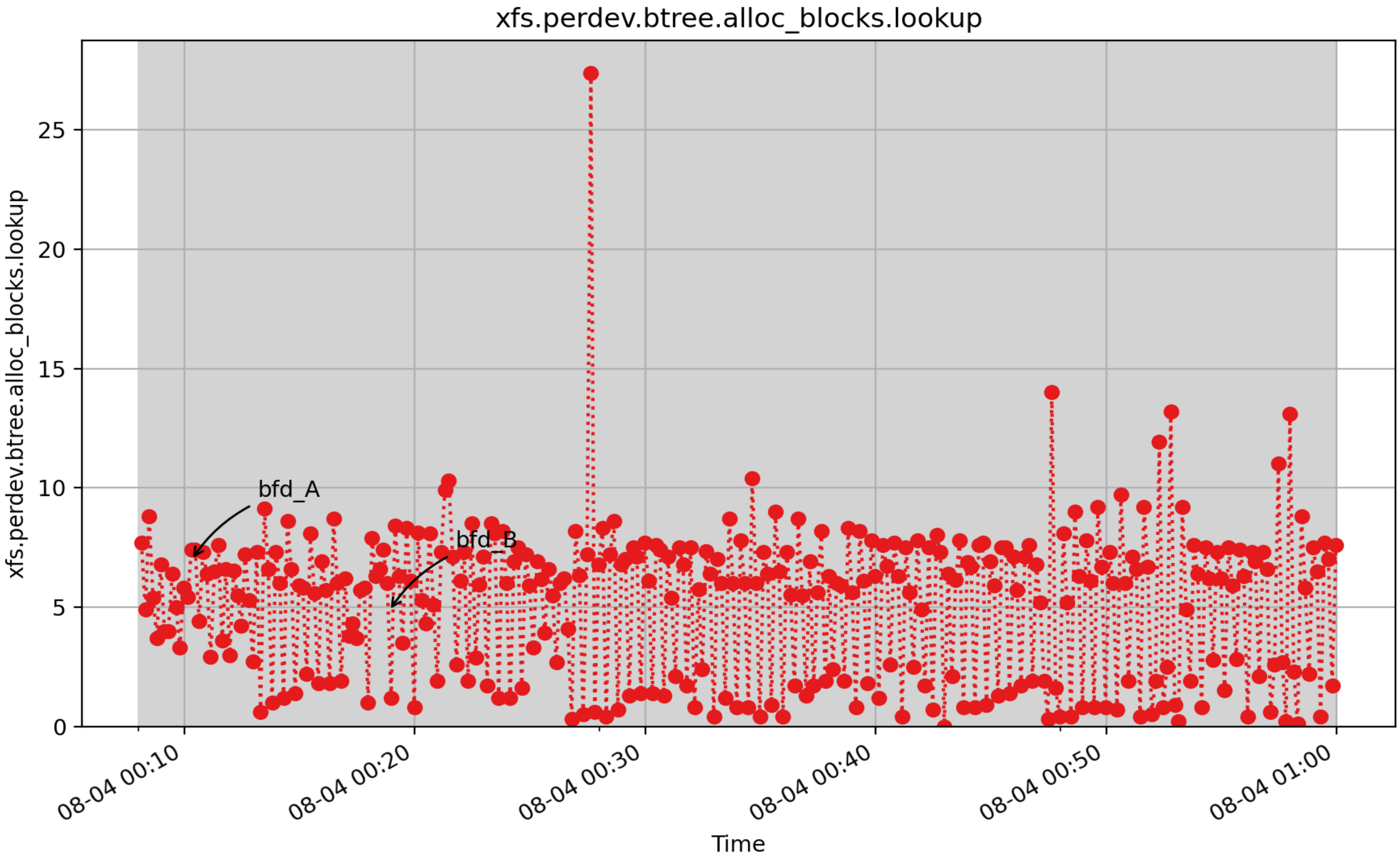
xfs.perdev.btree.alloc_blocks.compare: Number of free-space-by-block-number btree record compares (count - U32) - *rate converted*



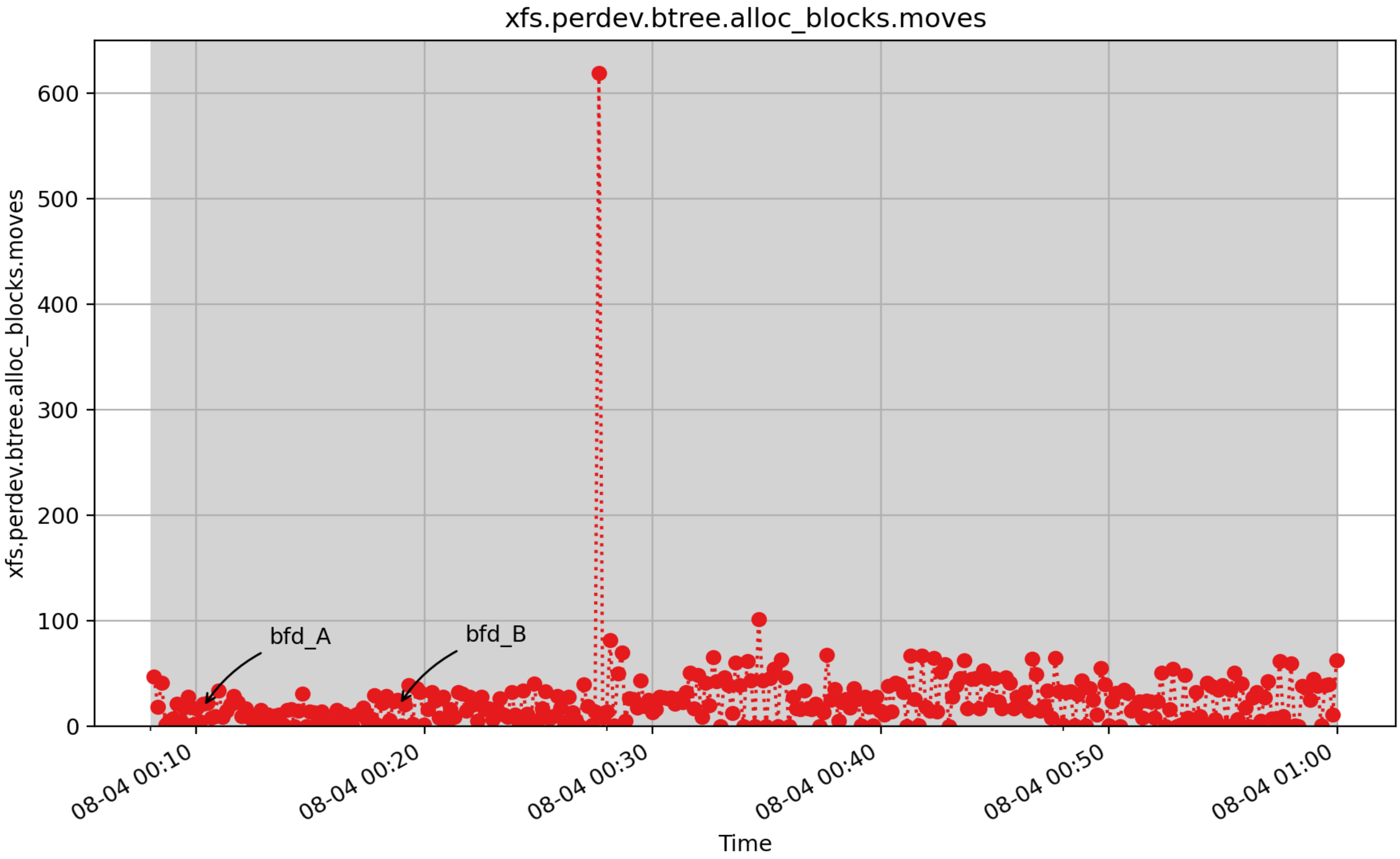
xfs.perdev.btree.alloc_blocks.delrec: Number of free-space-by-block-number btree delete record operations executed (count - U32) - *rate converted*



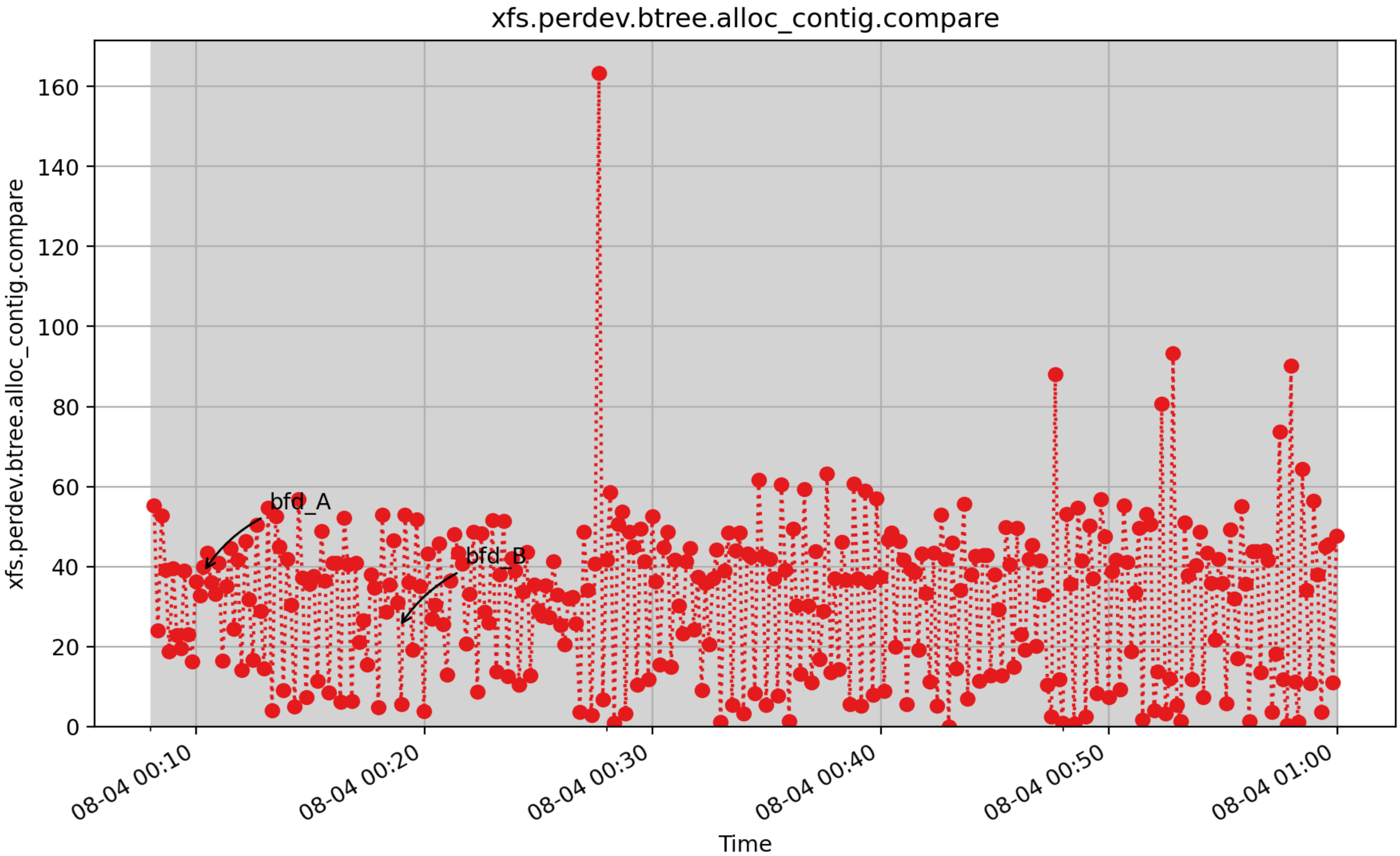
xfs.perdev.btree.alloc_blocks.insrec: Number of free-space-by-block-number btree insert record operations executed (count - U32) - *rate converted*



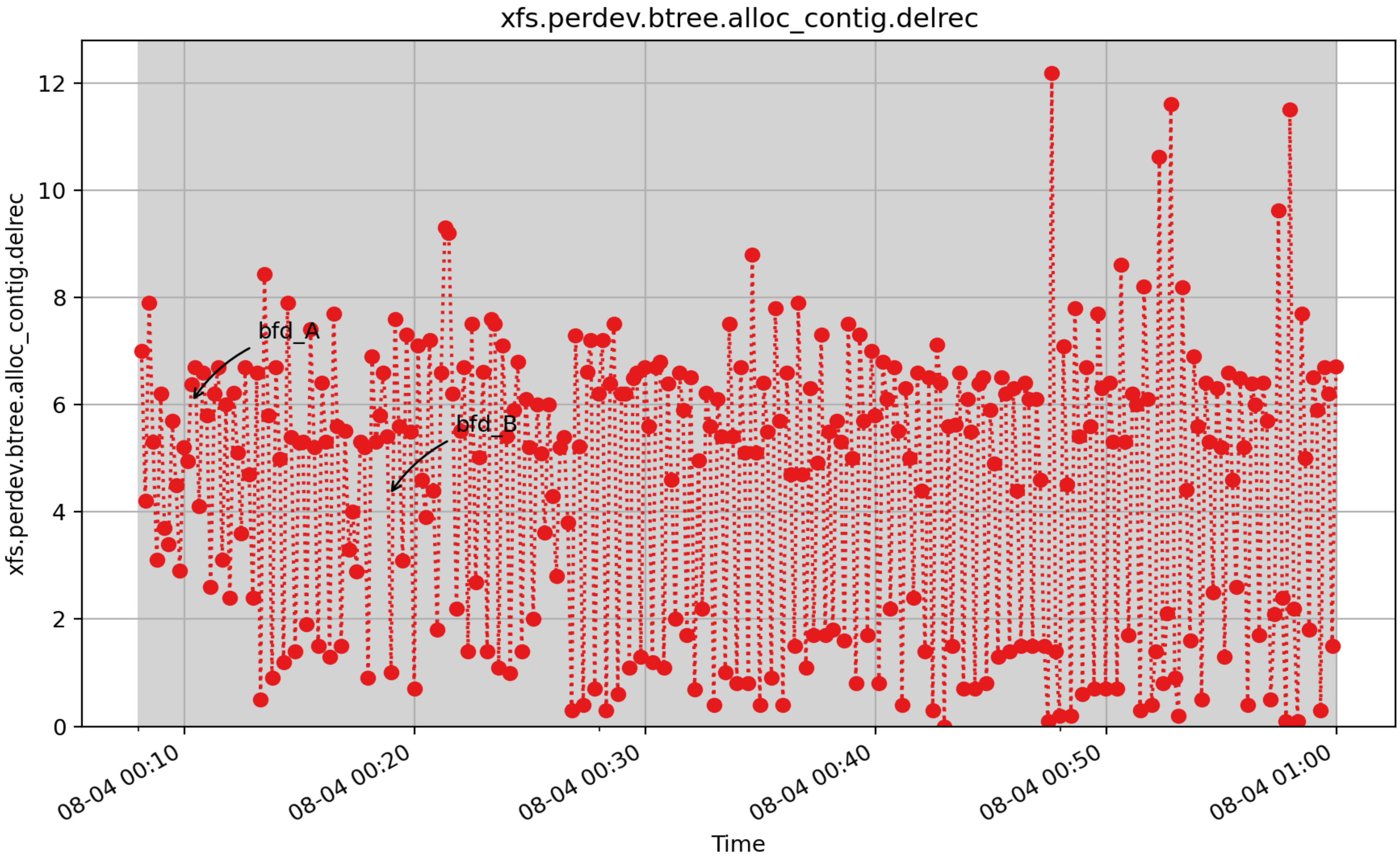
xfs.perdev.btree.alloc_blocks.lookup: Number of free-space-by-block-number btree record lookups (count - U32) - *rate converted*



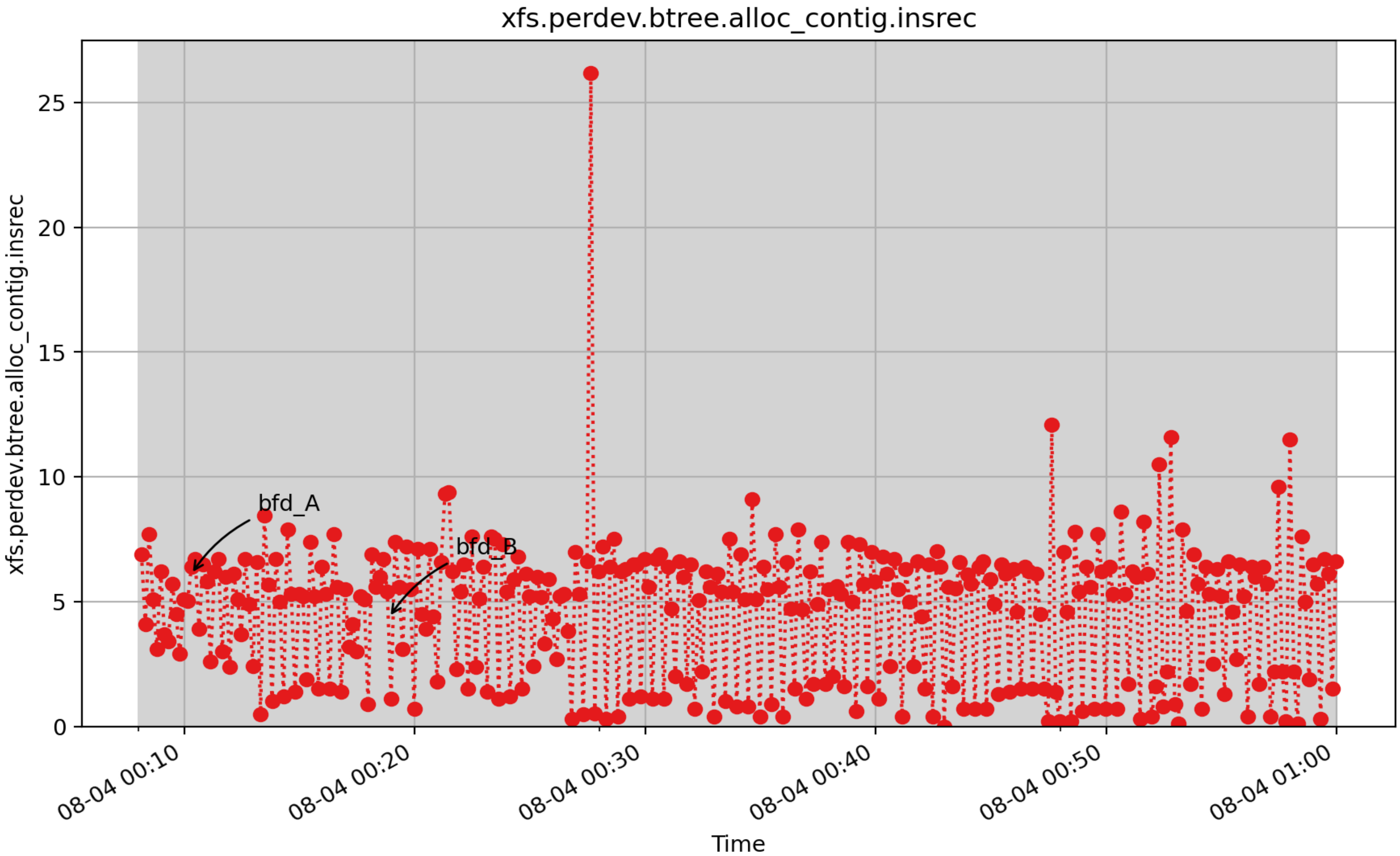
xfs.perdev.btree.alloc_blocks.moves: Records moved inside blocks during free-space-by-block-number btree operations (count - U32) - *rate converted*



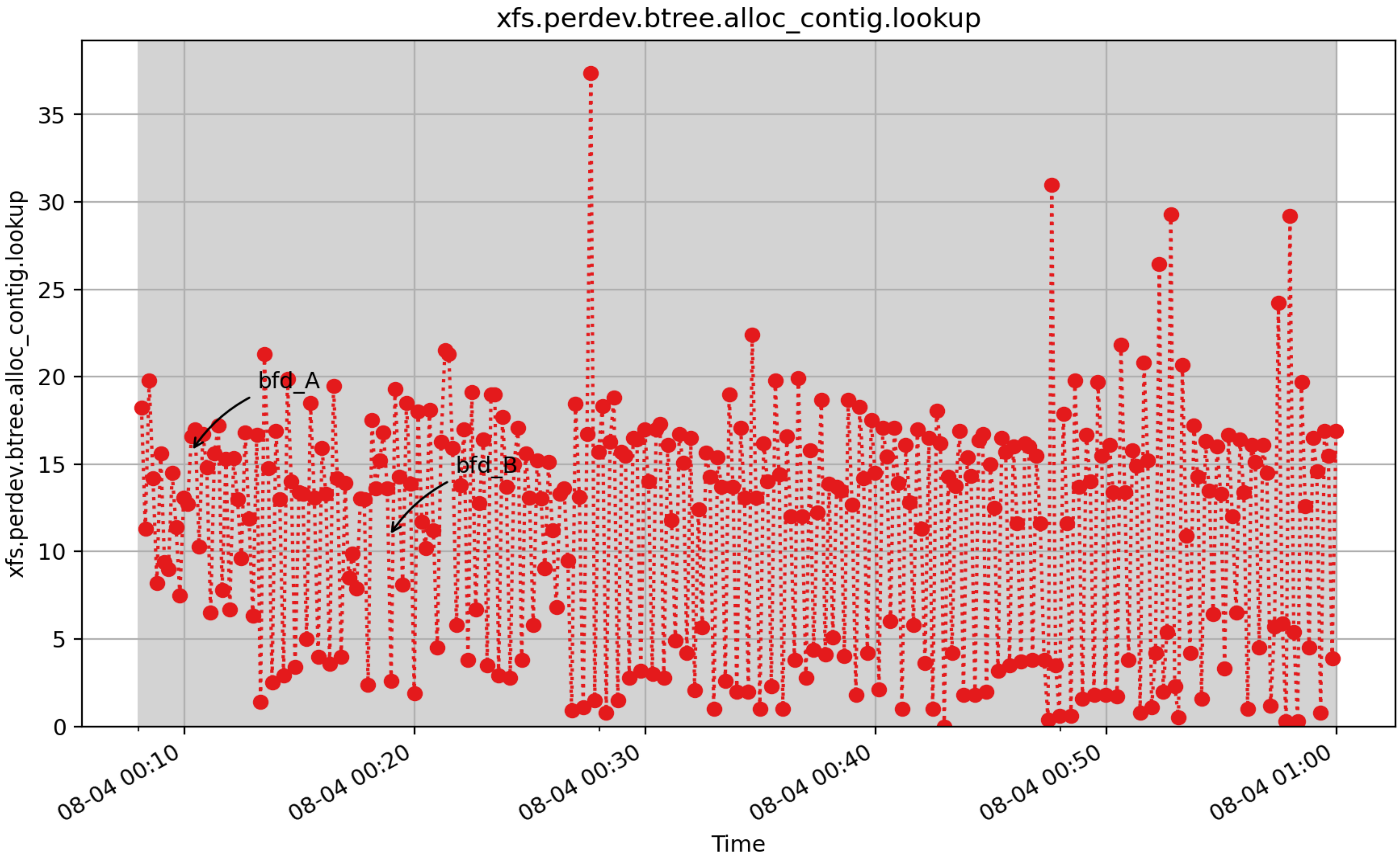
xfs.perdev.btree.alloc_contig.compare: Number of free-space-by-size btree btree record compares (count - U32) - *rate converted*



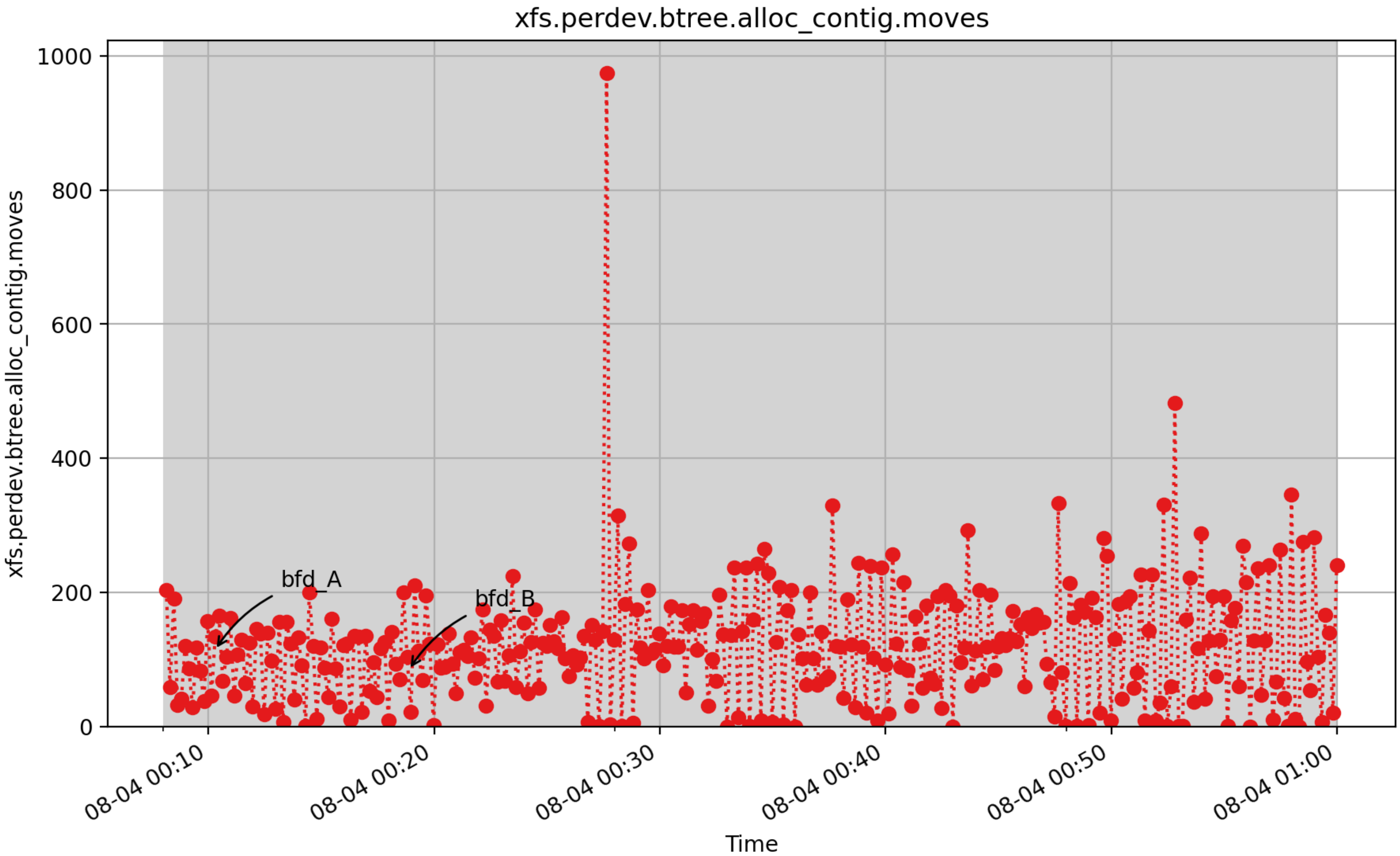
xfs.perdev.btree.alloc_contig.delrec: Number of free-space-by-size btree delete record operations executed (count - U32) - *rate converted*



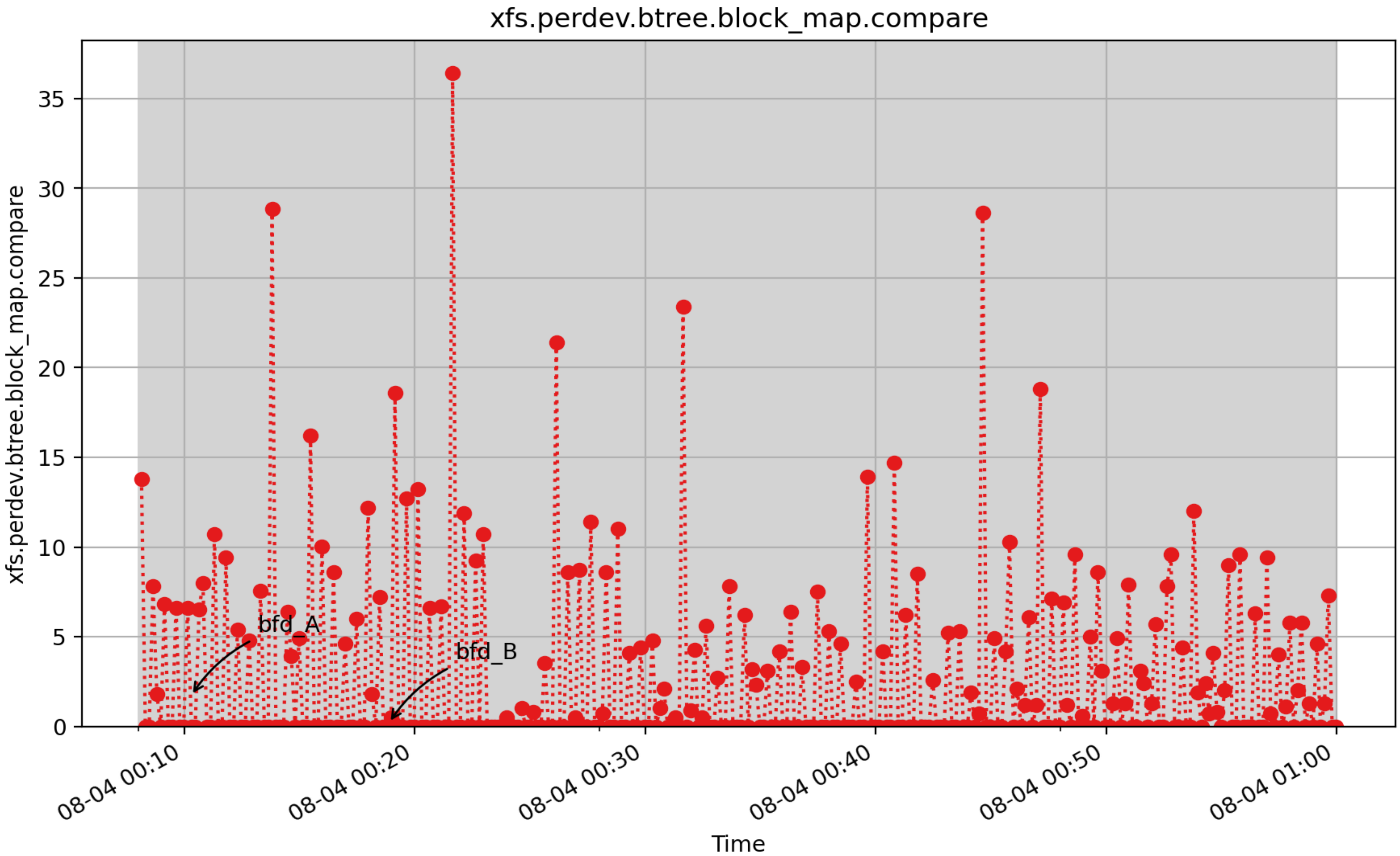
xfs.perdev.btree.alloc_contig.insrec: Number of free-space-by-size btree insert record operations executed (count - U32) - *rate converted*



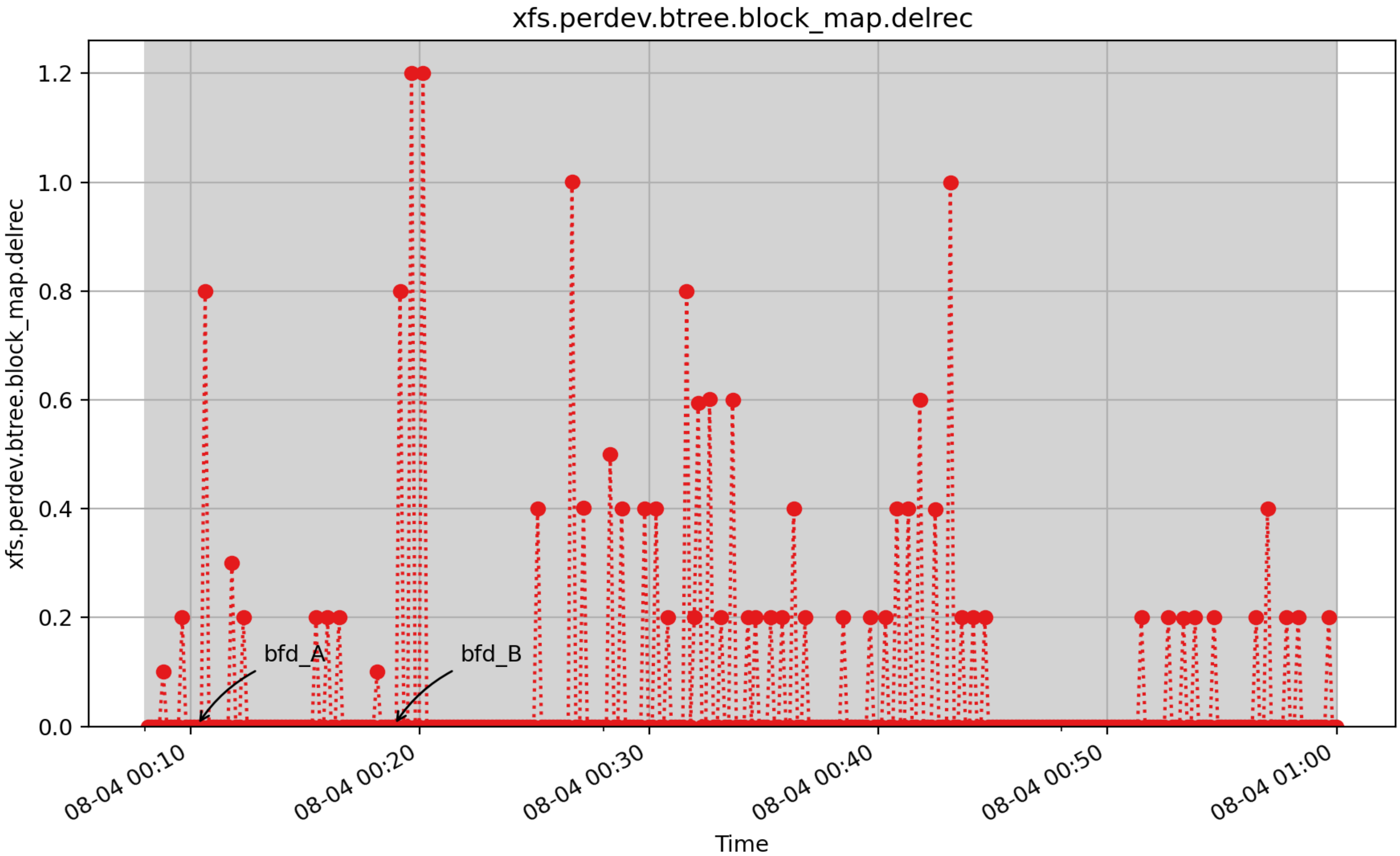
xfs.perdev.btree.alloc_contig.lookup: Number of free-space-by-size btree record lookups (count - U32) - *rate converted*



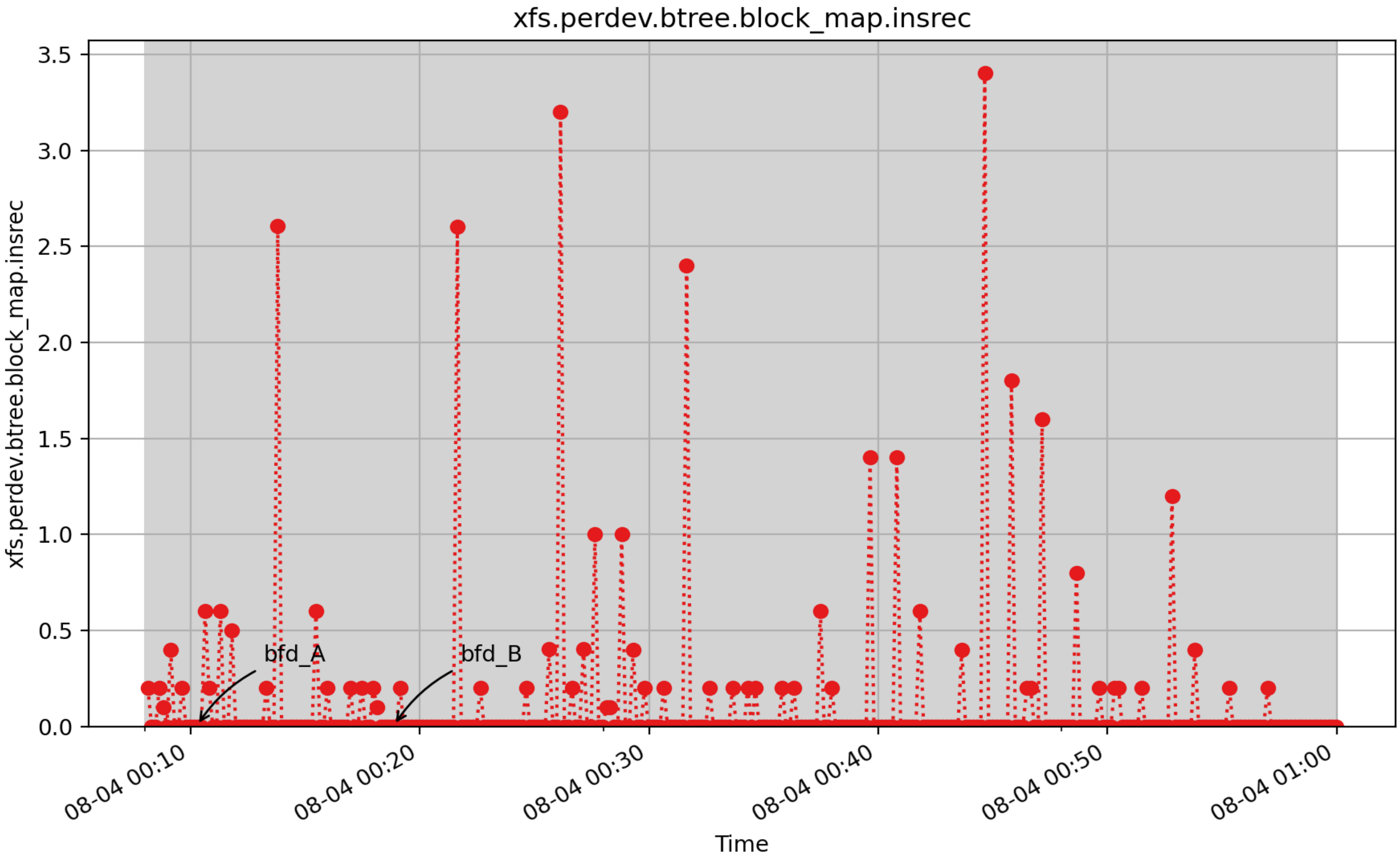
xfs.perdev.btree.alloc_contig.moves: Records moved inside blocks during free-space-by-size btree operations (count - U32) - *rate converted*



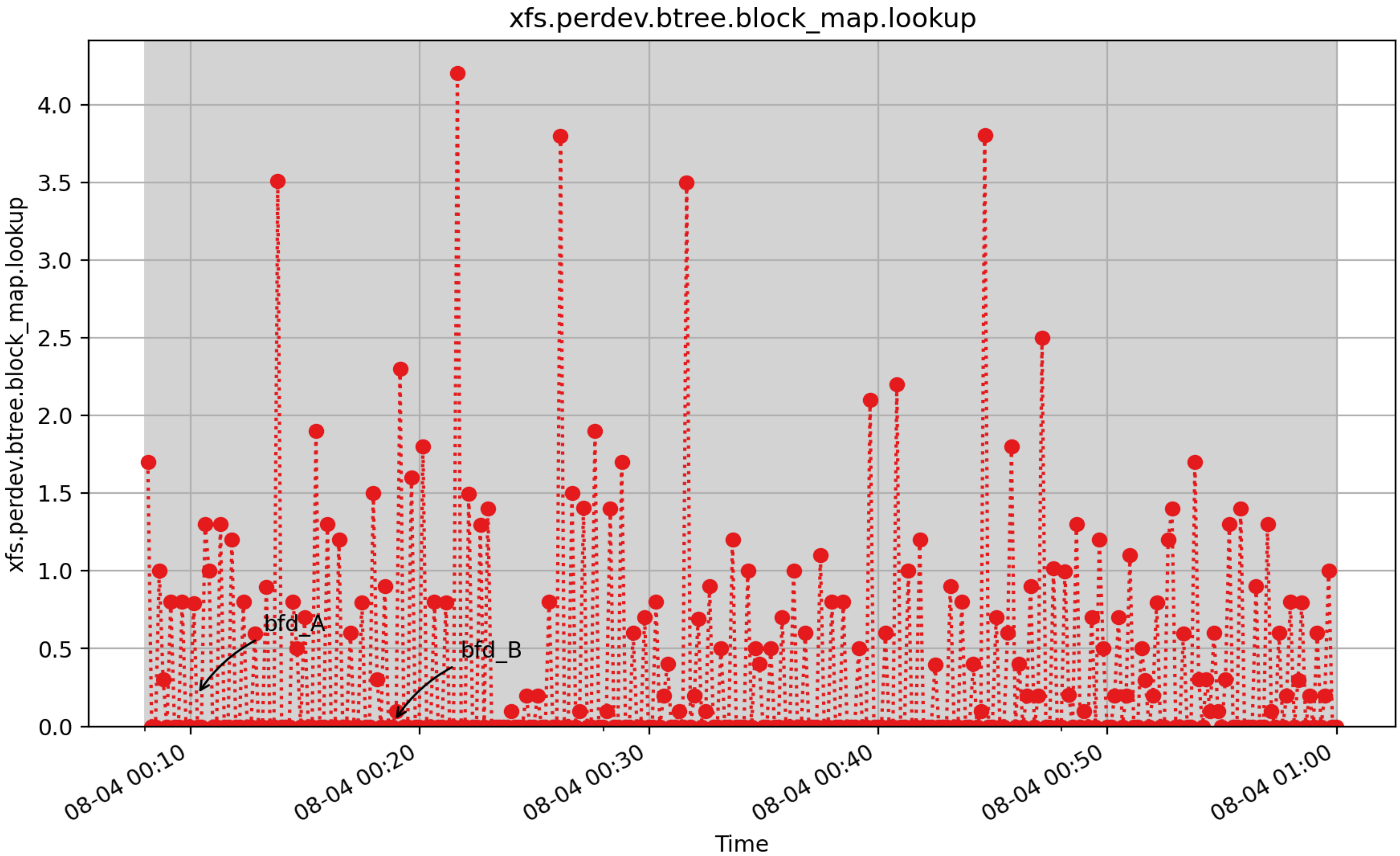
xfs.perdev.btree.block_map.compare: Number of inode-block-map/extent btree record compares (count - U32) - *rate converted*



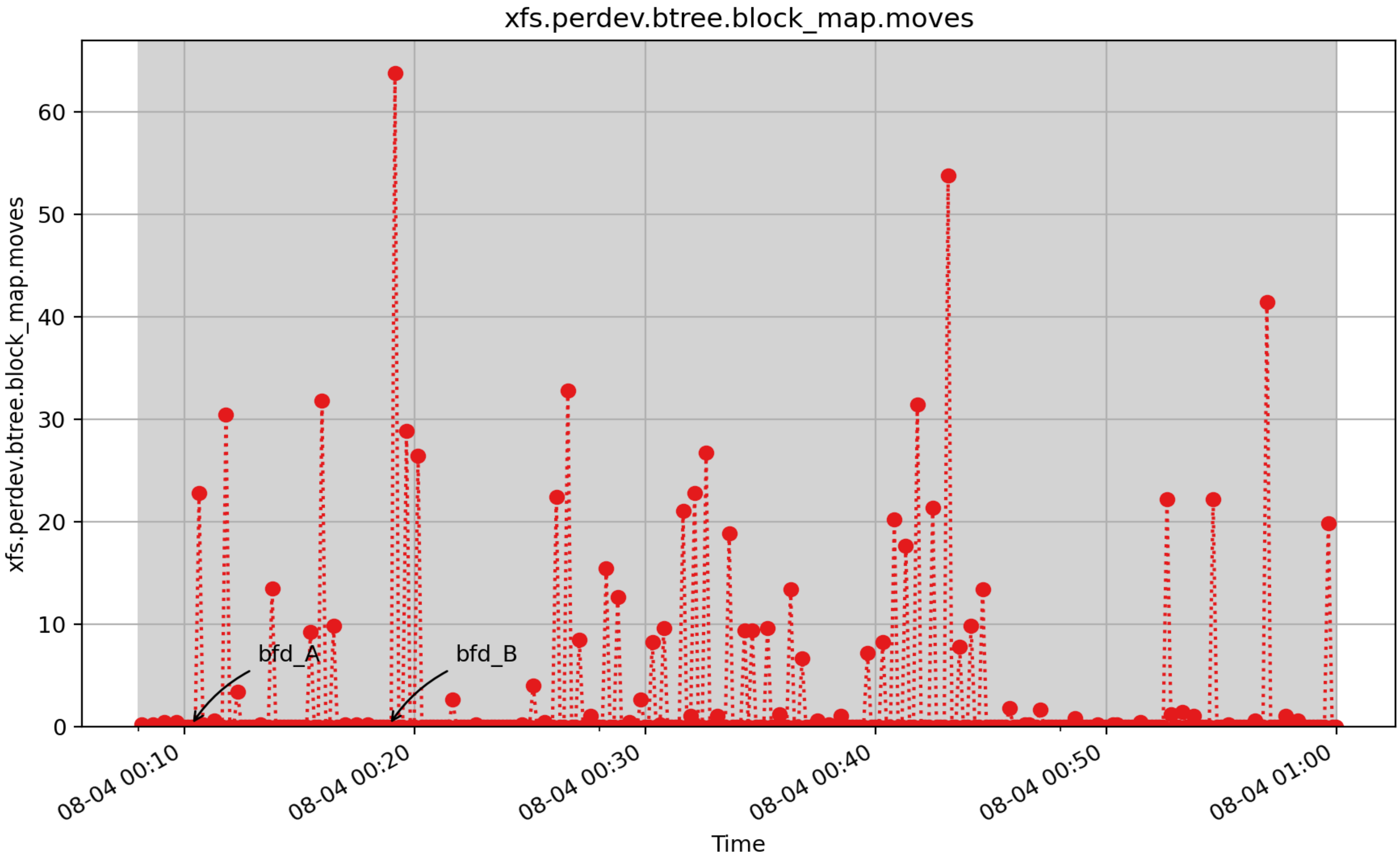
xfs.perdev.btree.block_map.delrec: Number of inode-block-map/extent btree delete record operations executed (count - U32) - *rate converted*



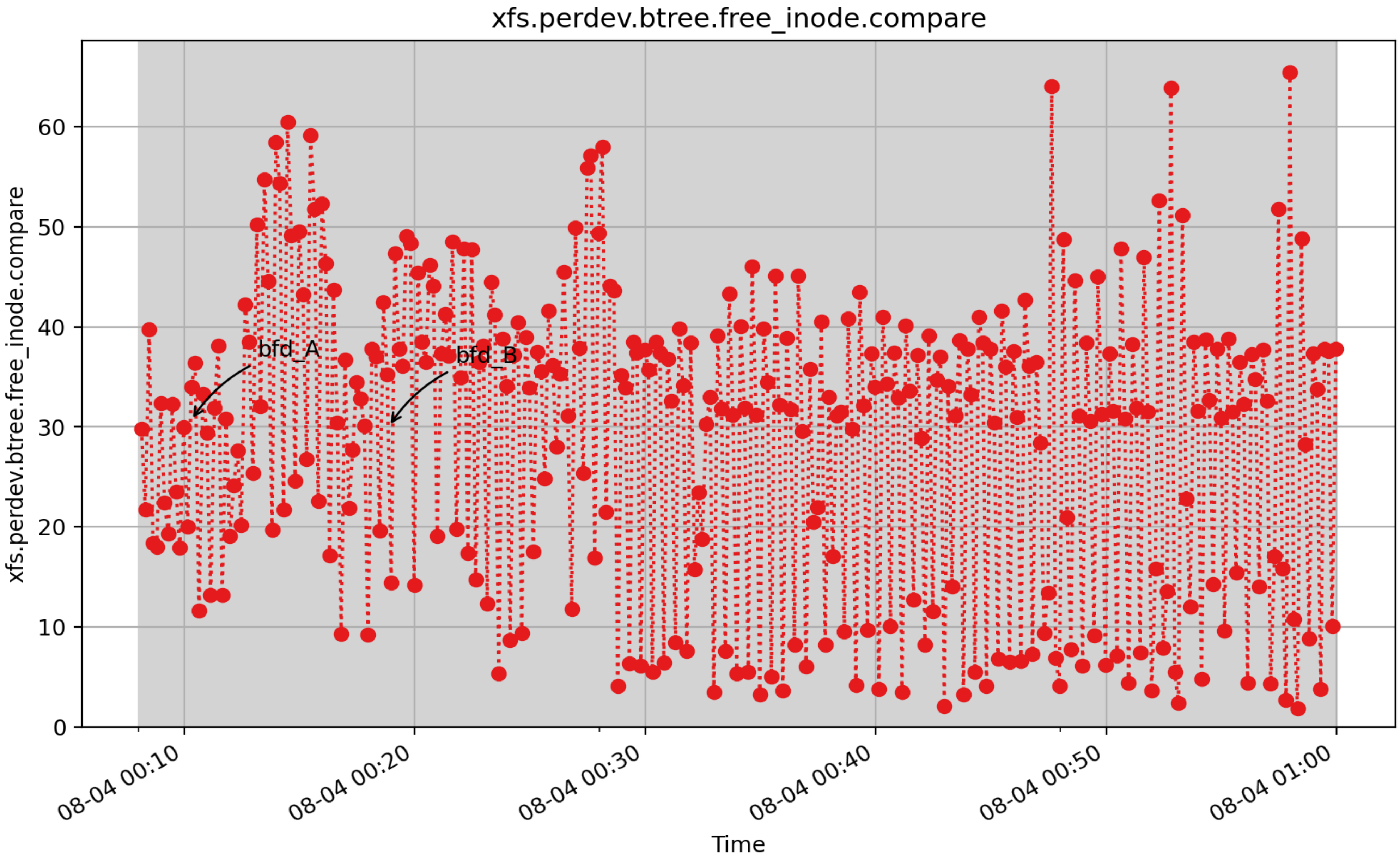
xfs.perdev.btree.block_map.insrec: Number of inode-block-map/extent btree insert record operations executed (count - U32) - *rate converted*



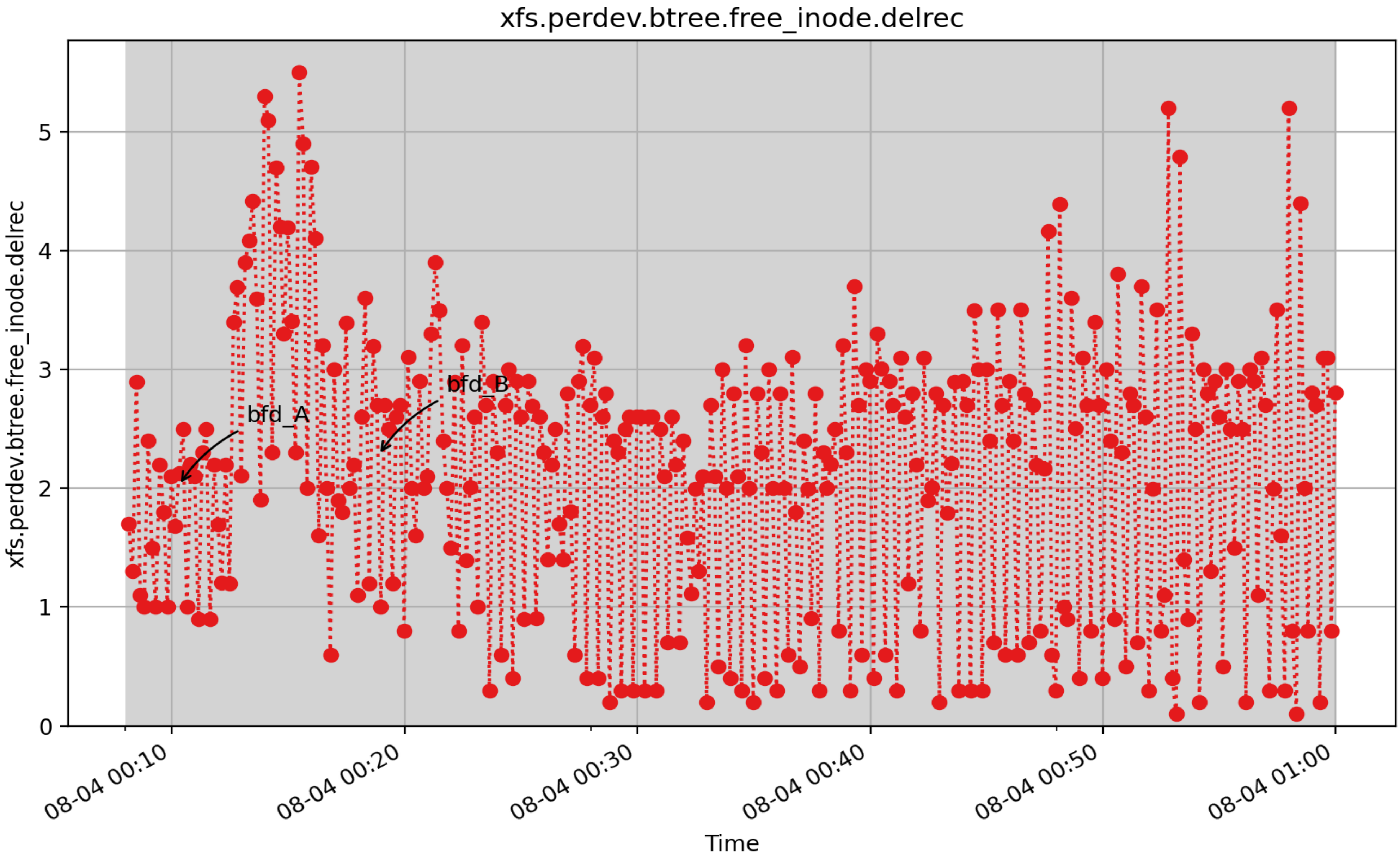
xfs.perdev.btree.block_map.lookup: Number of inode-block-map/extent btree record lookups (count - U32) - *rate converted*



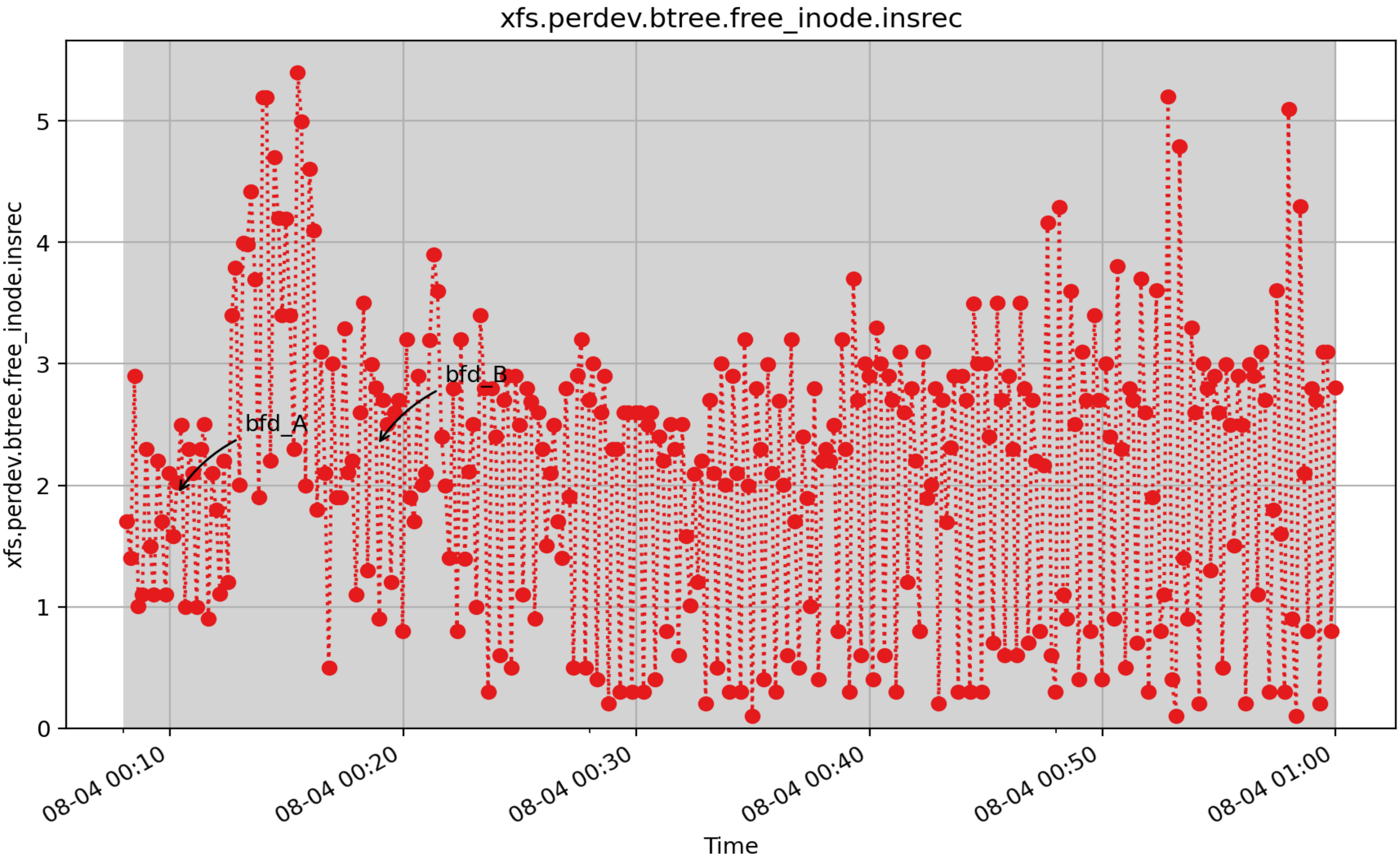
xfs.perdev.btree.block_map.moves: Records moved inside blocks during inode-block-map/extent btree operations (count - U32) - *rate converted*



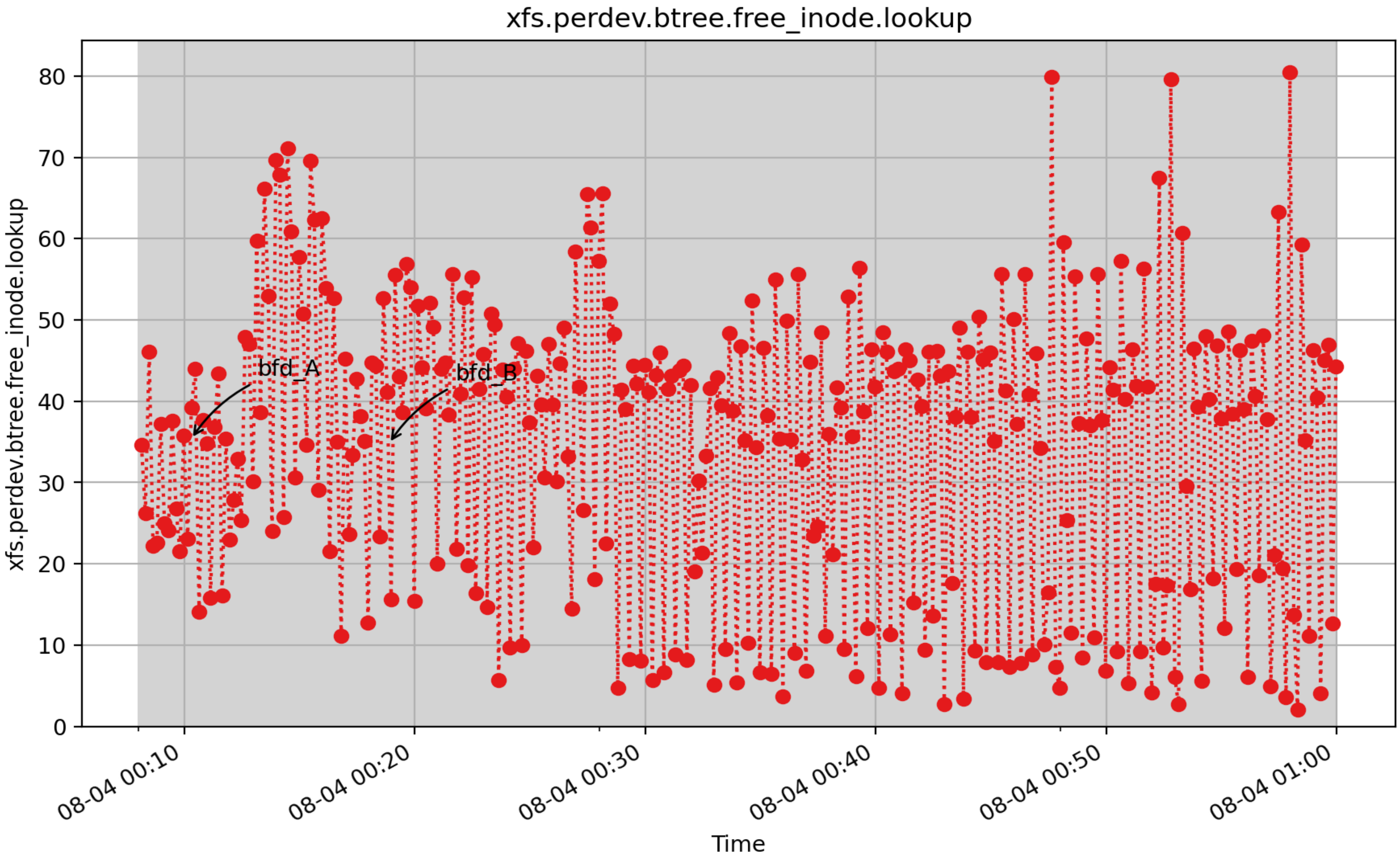
xfs.perdev.btree.free_inode.compare: Number of free-inode btree record compares (count - U32) - *rate converted*



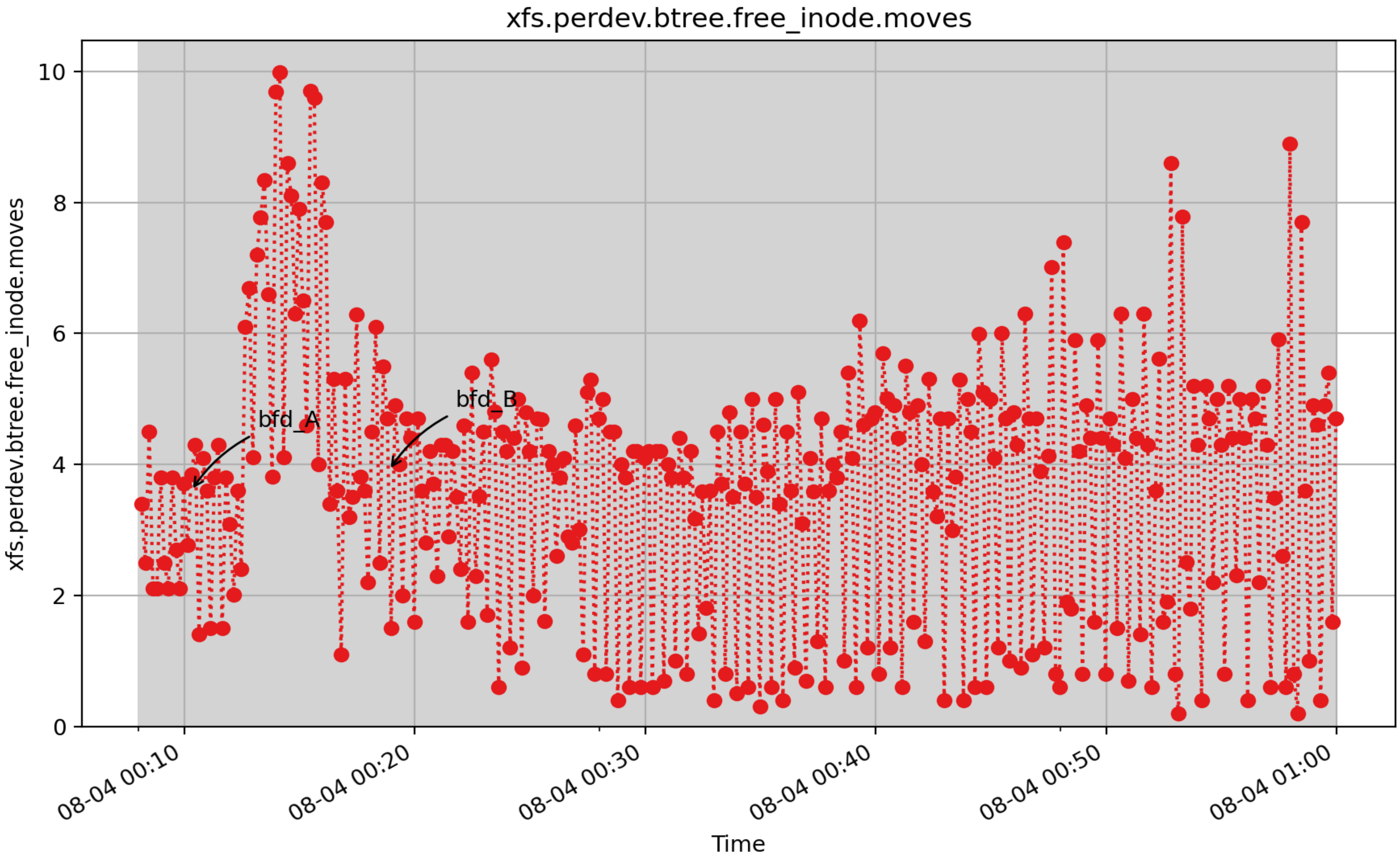
xfs.perdev.btree.free_inode.delrec: Number of free-inode btree delete record operations executed (count - U32) - *rate converted*



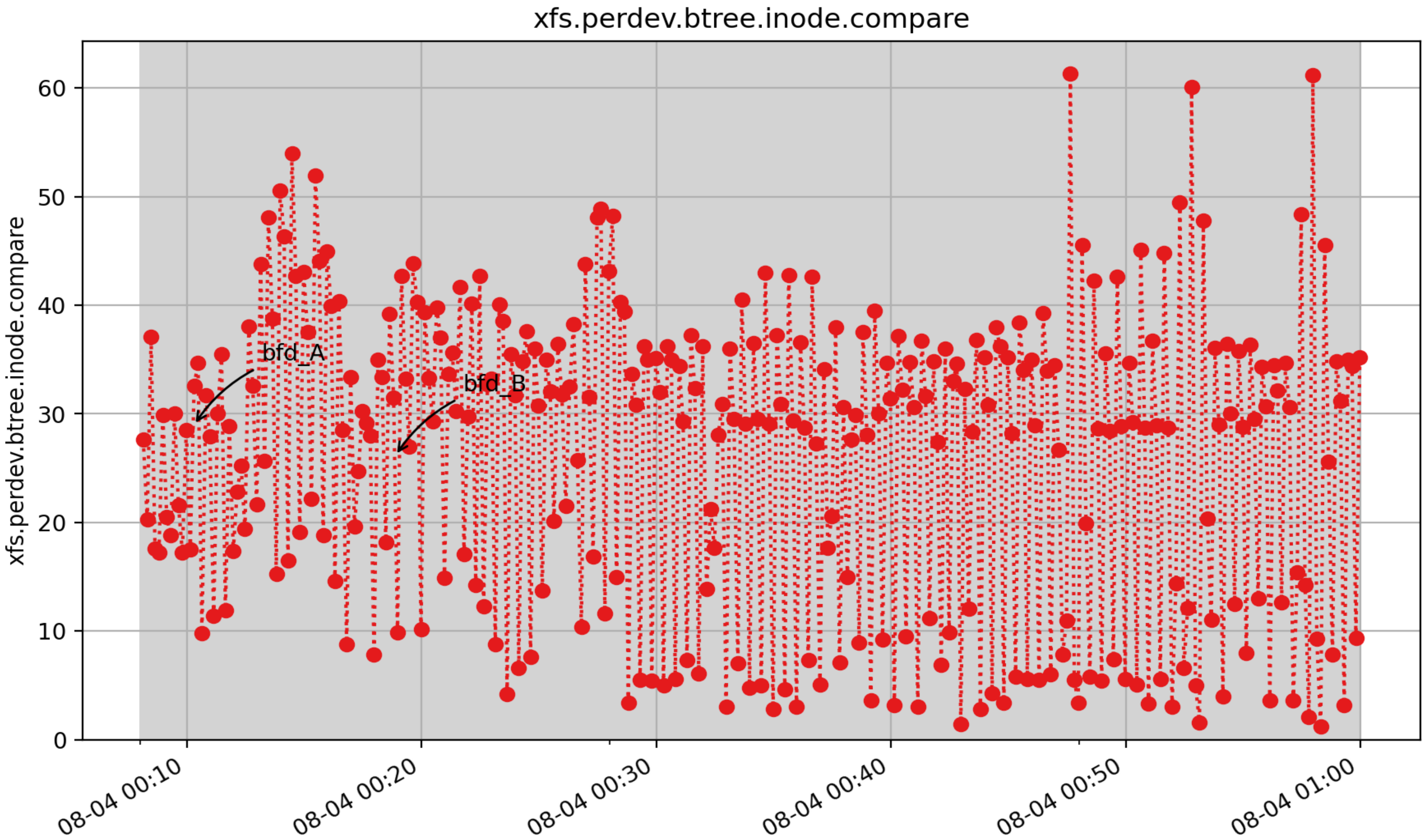
xfs.perdev.btree.free_inode.insrec: Number of free-inode btree insert record operations executed (count - U32) - *rate converted*



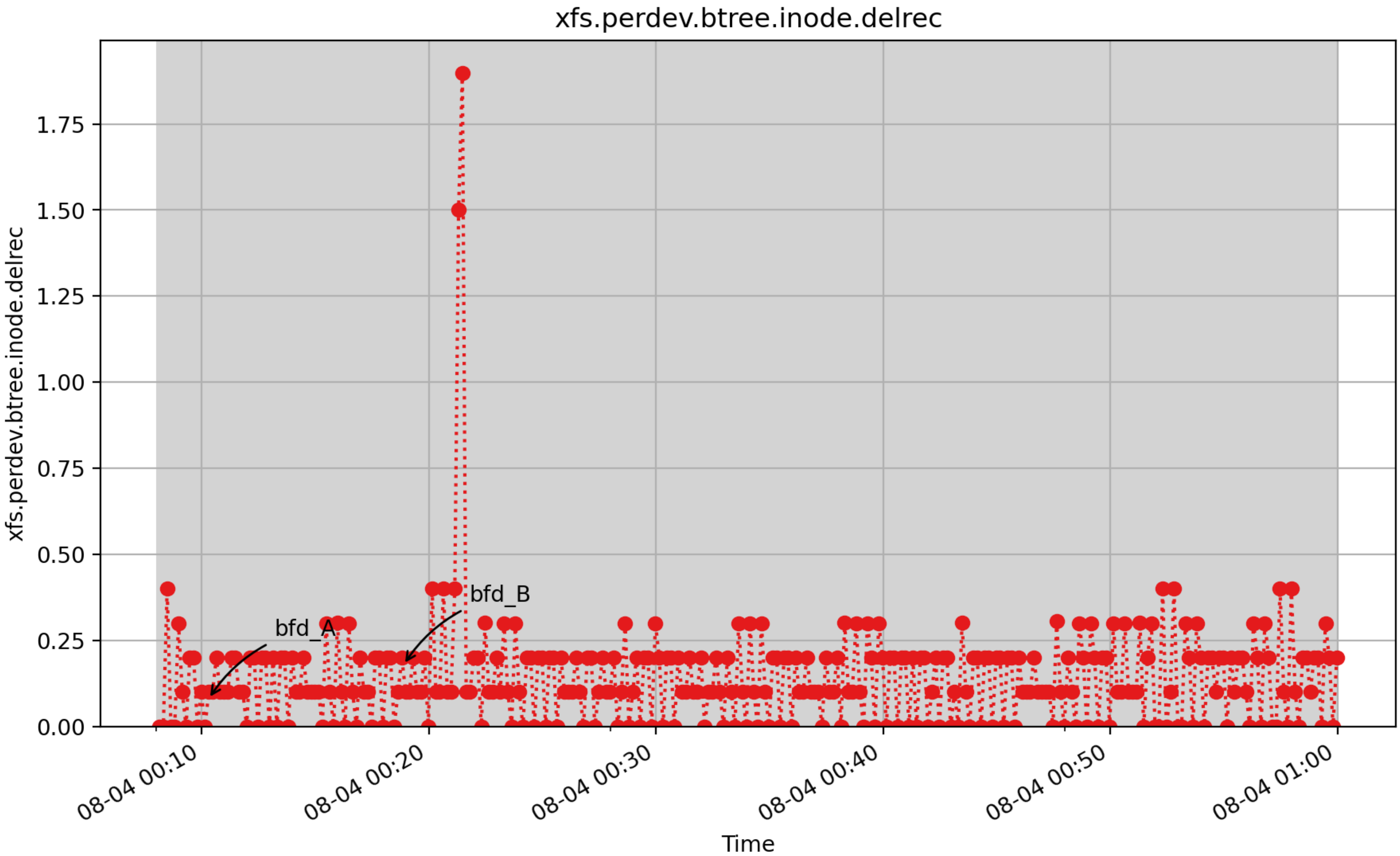
xfs.perdev.btree.free_inode.lookup: Number of free-inode btree record lookups (count - U32) - *rate converted*



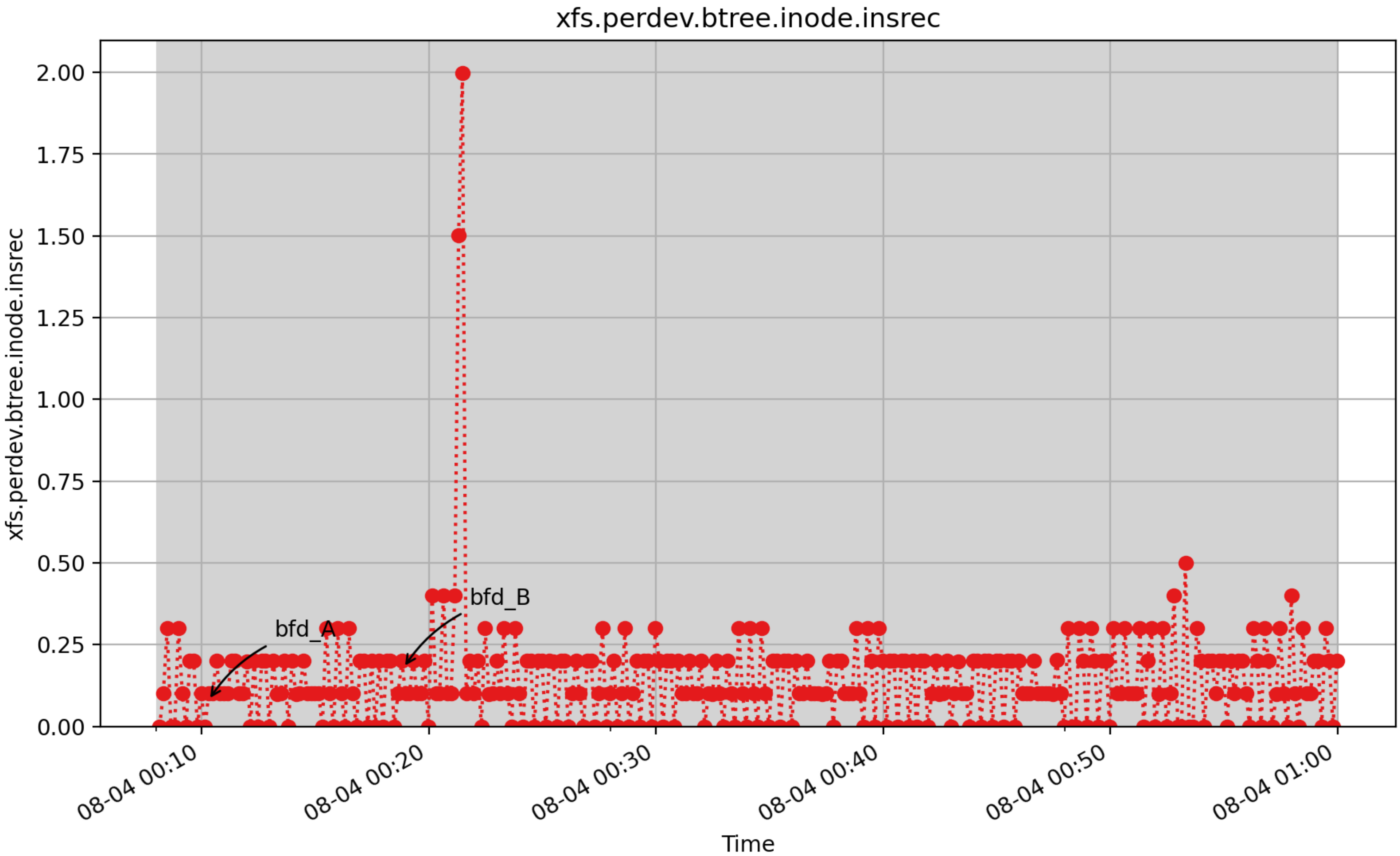
xfs.perdev.btree.free_inode.moves: Records moved inside blocks during free-inode btree operations (count - U32) - *rate converted*



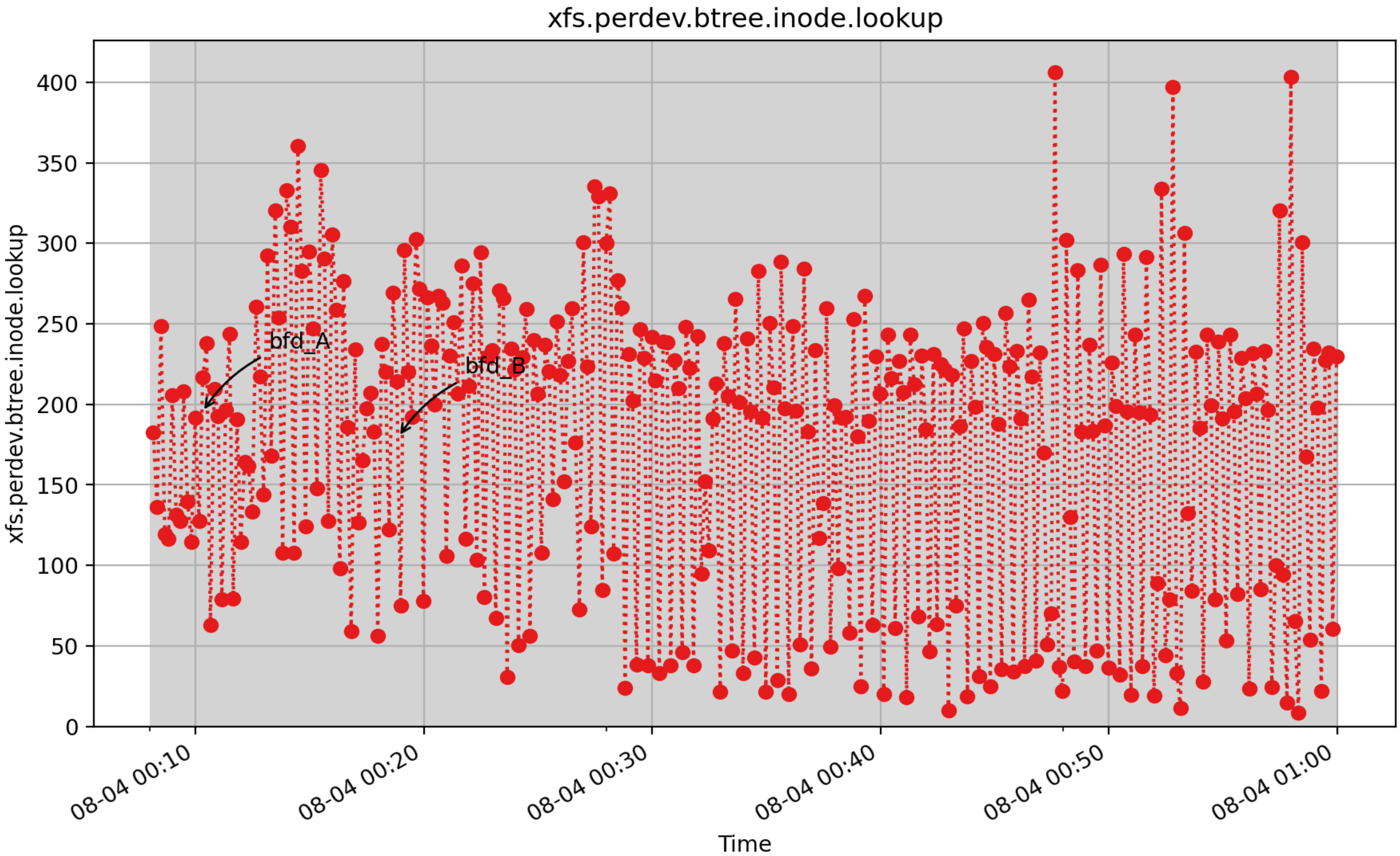
xfs.perdev.btree.inode.compare: Number of inode-allocation btree record compares (count - U32) - *rate converted*



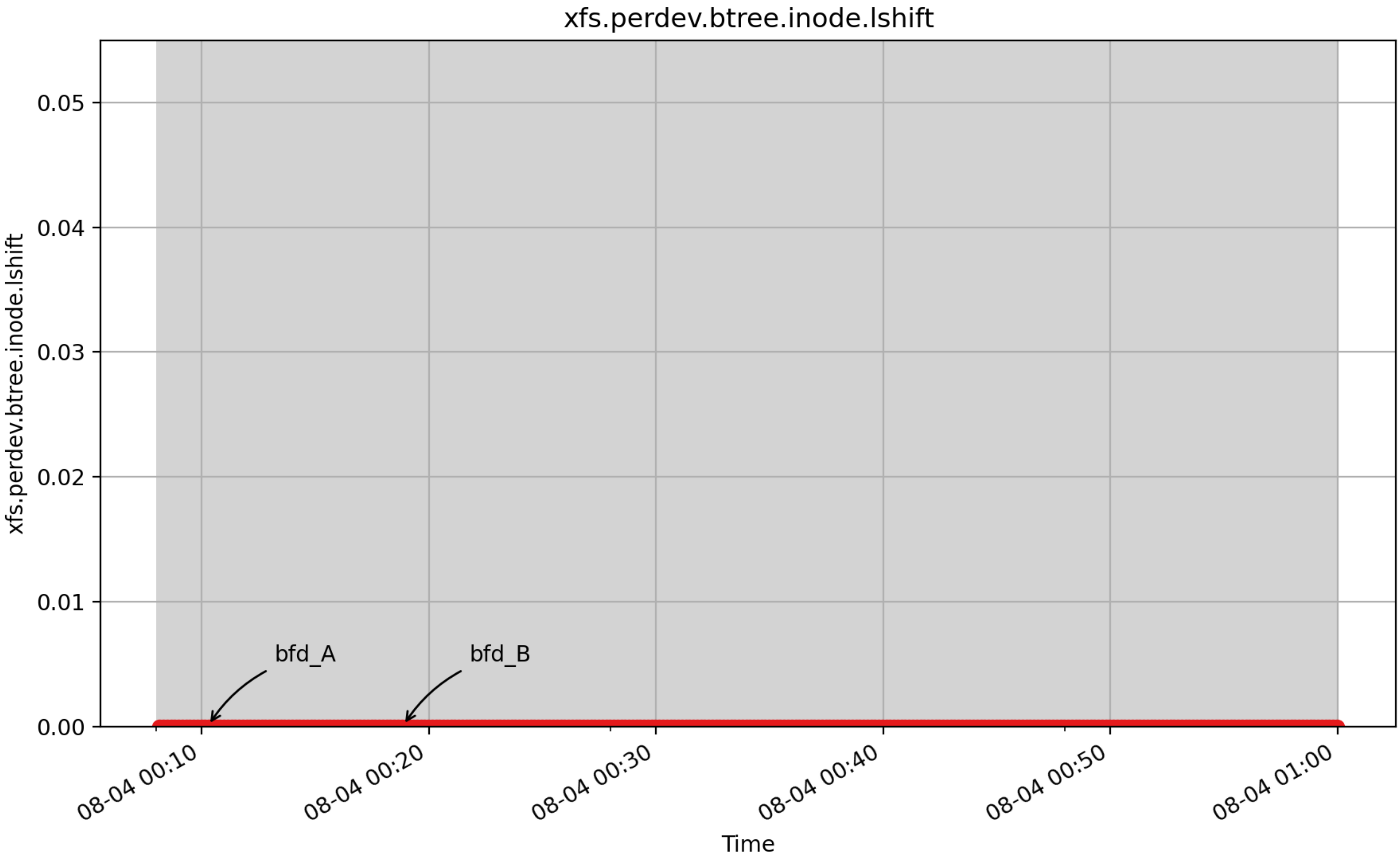
xfs.perdev.btree.inode.delrec: Number of inode-allocation btree delete record operations executed (count - U32) - *rate converted*



xfs.perdev.btree.inode.insrec: Number of inode-allocation btree insert record operations executed (count - U32) - *rate converted*

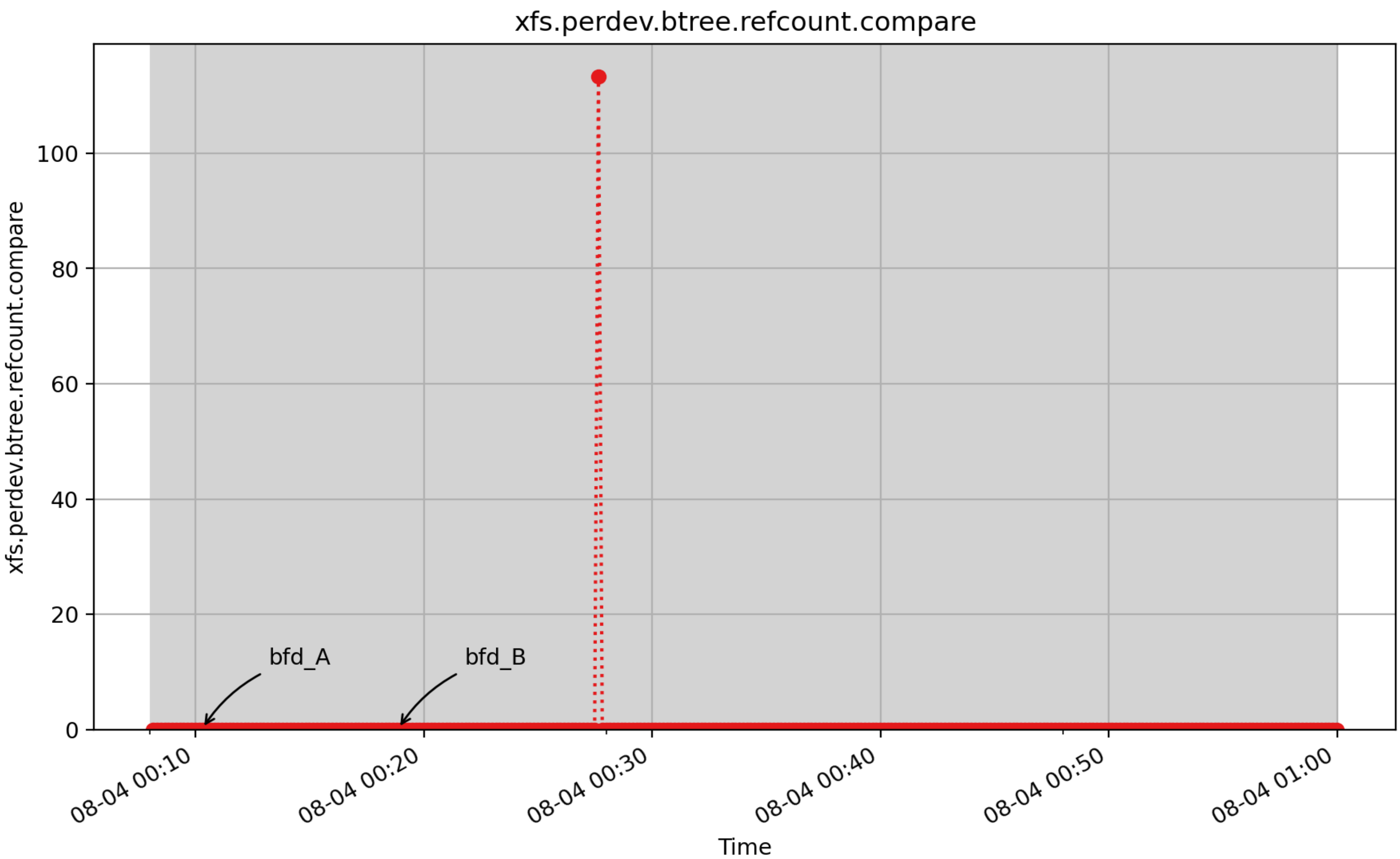


xfs.perdev.btree.inode.lookup: Number of inode-allocation btree record lookups (count - U32) - *rate converted*

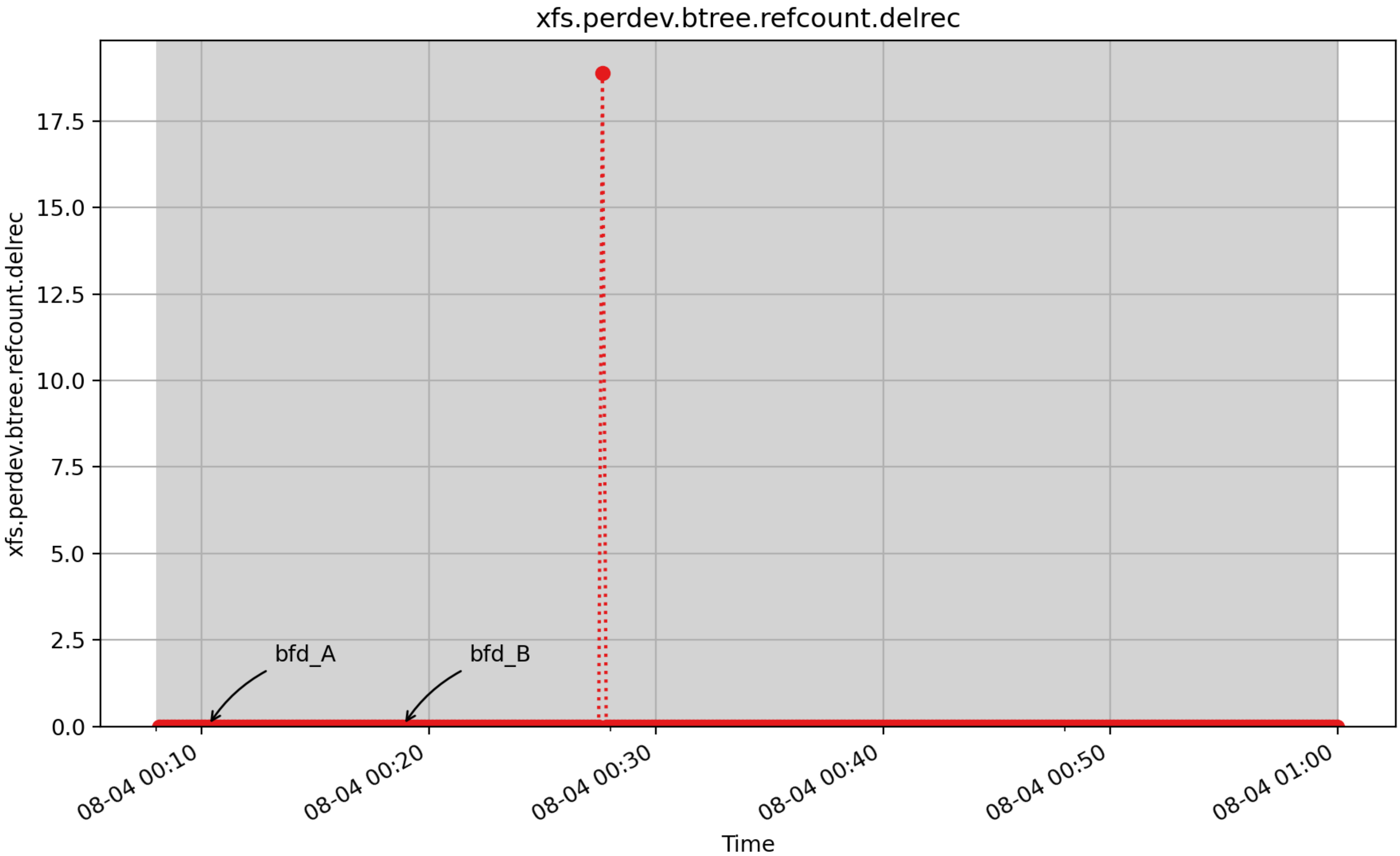


xfs.perdev.btree.inode.lshift: Left shift block operations to make space for a new inode-allocation btree record (count - U32) - *rate converted*

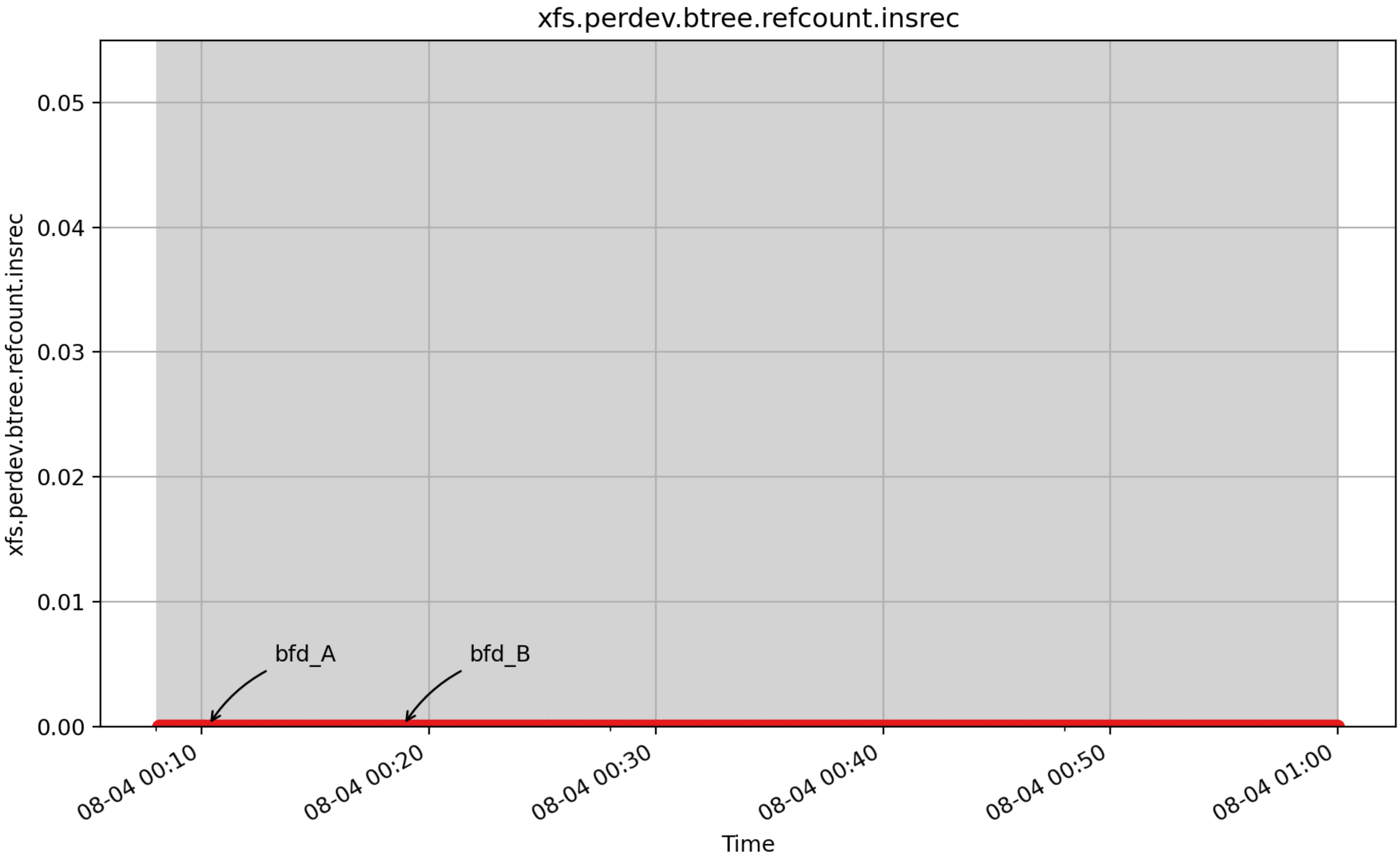
xfs.perdev.btree.inode.moves: Records moved inside blocks during inode-allocation btree operations (count - U32) - *rate converted*



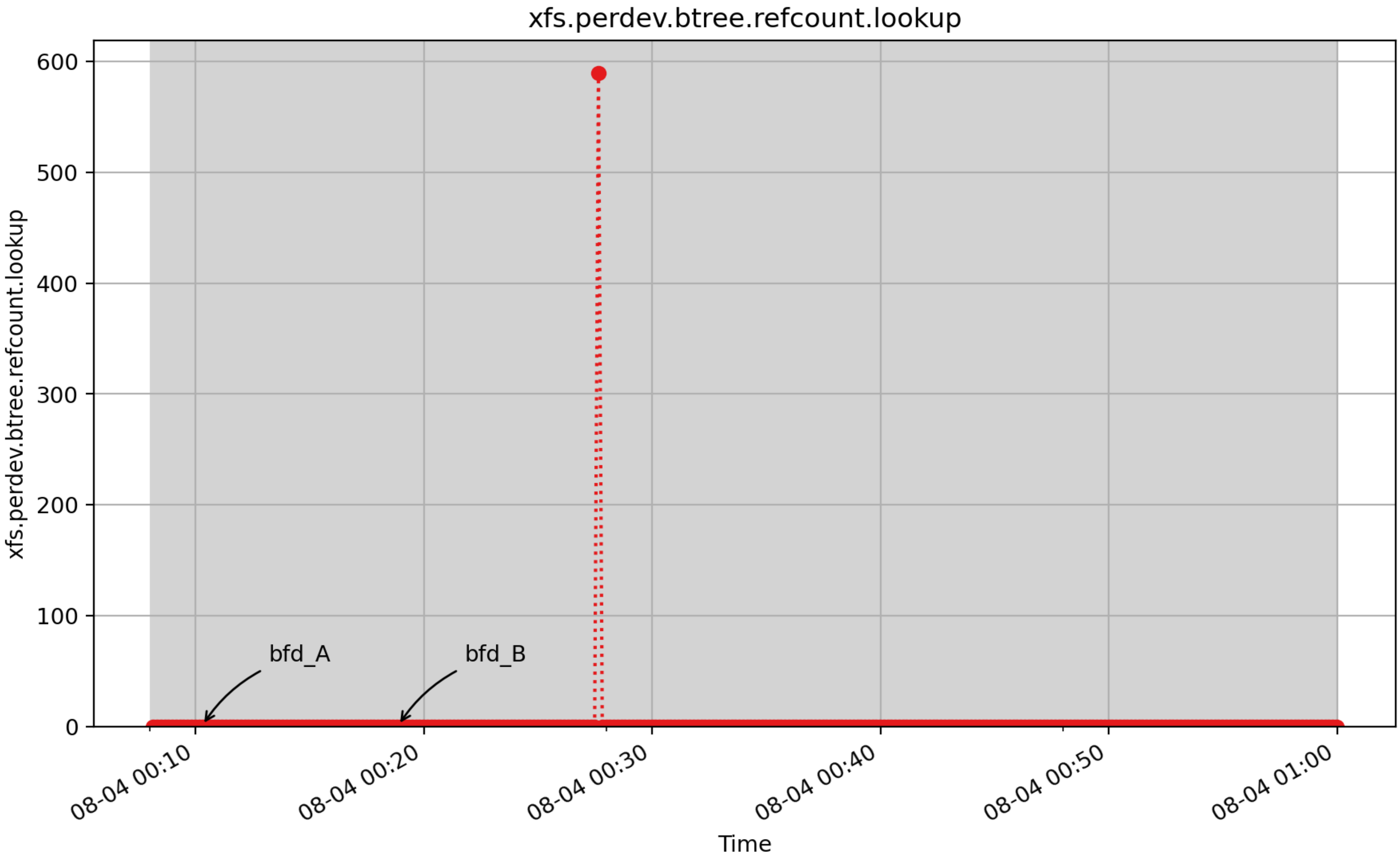
xfs.perdev.btree.refcount.compare: Number of reference-count btree record compares (count - U32) - *rate converted*



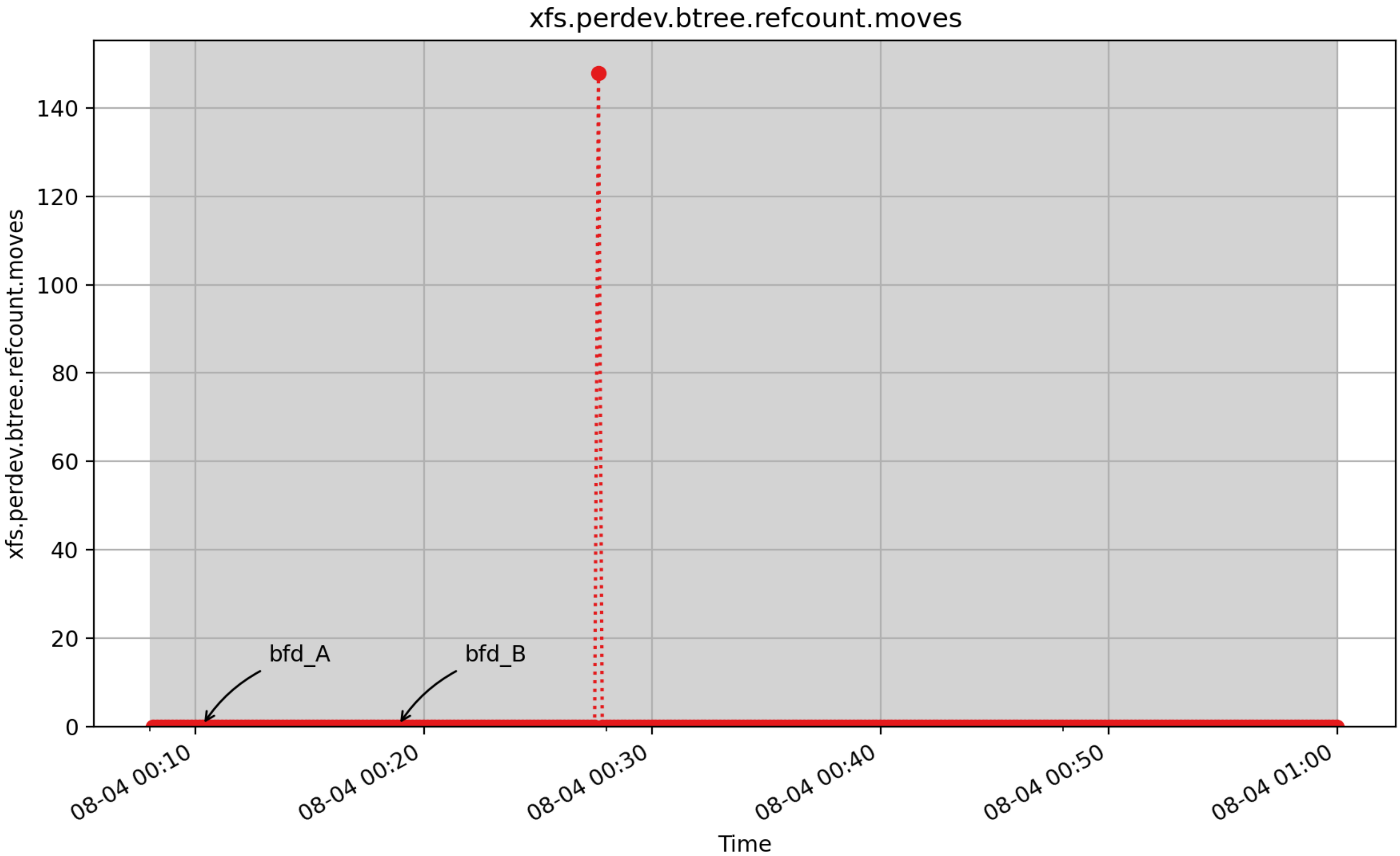
xfs.perdev.btree.refcount.delrec: Number of reference-count btree delete record operations executed (count - U32) - *rate converted*



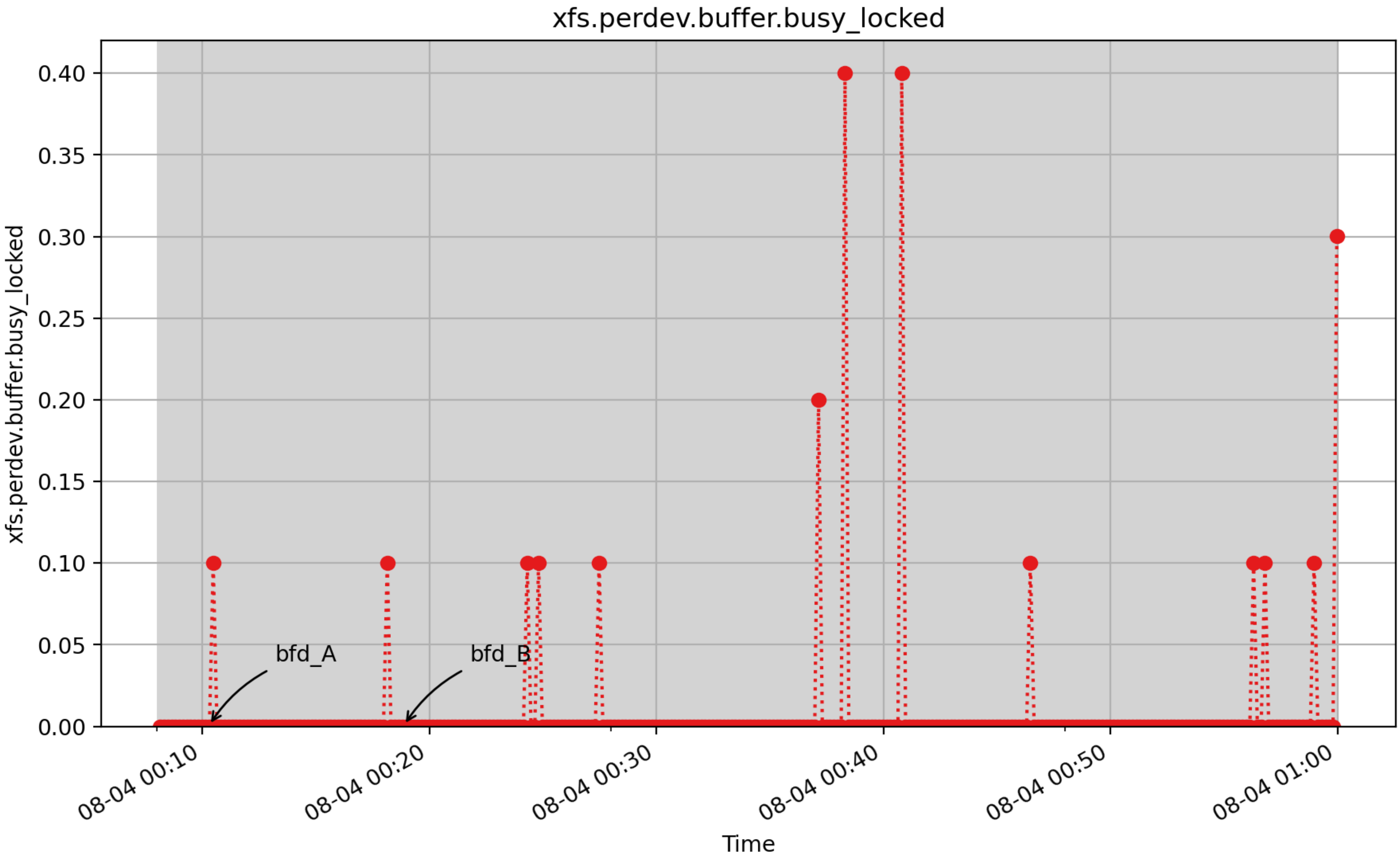
xfs.perdev.btree.refcount.insrec: Number of reference-count btree insert record operations executed (count - U32) - *rate converted*



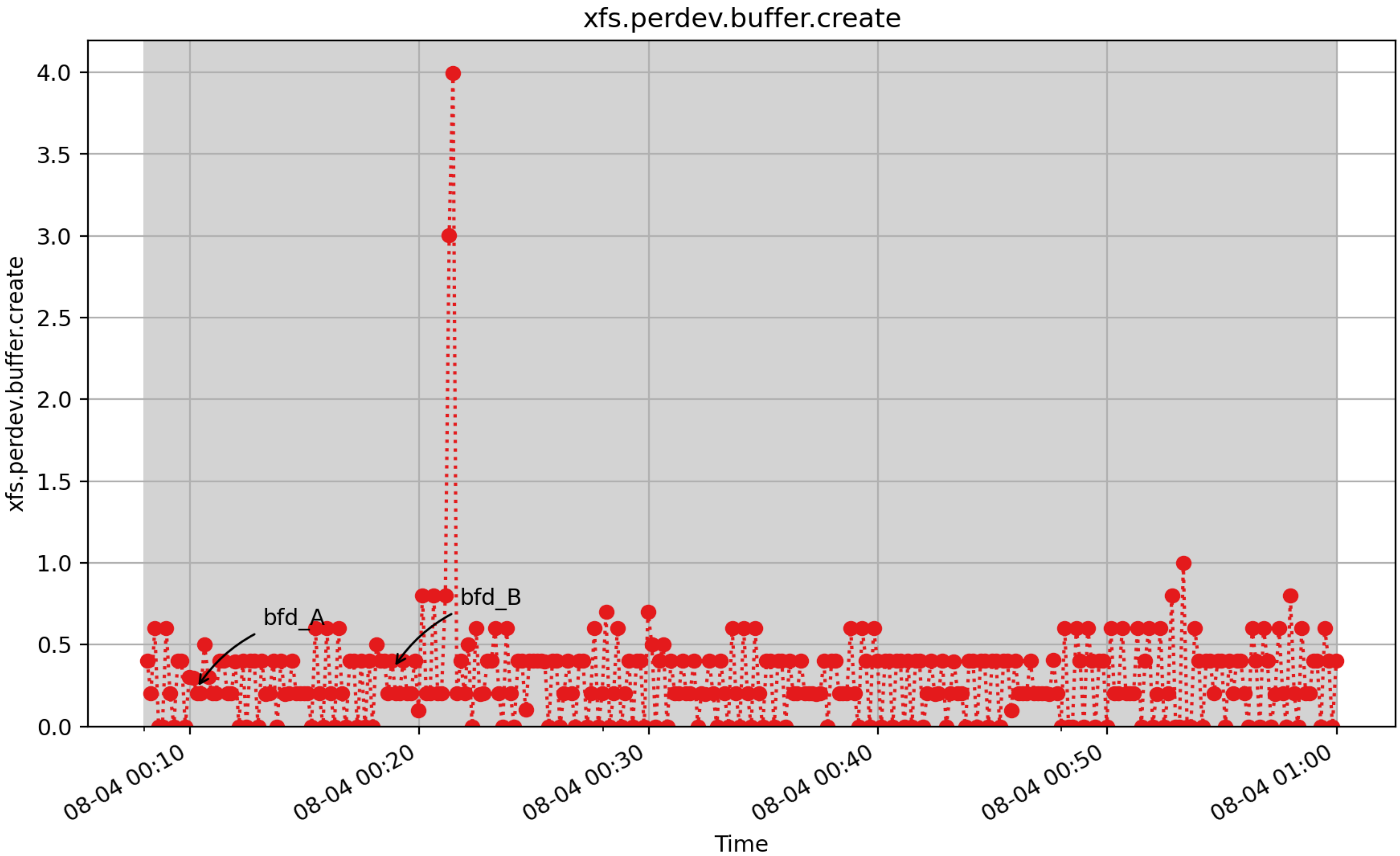
xfs.perdev.btree.refcount.lookup: Number of reference-count btree record lookups (count - U32) - *rate converted*



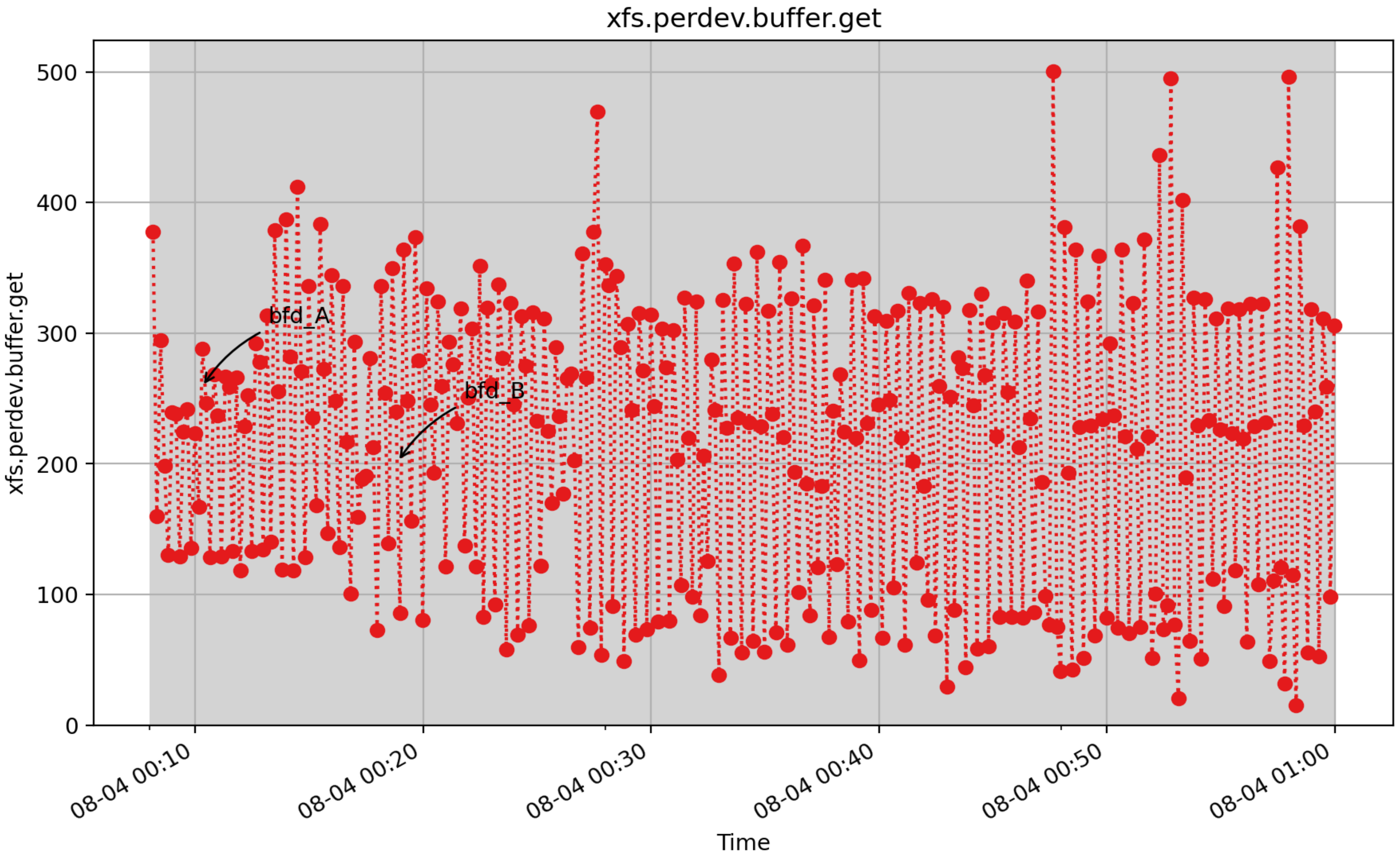
xfs.perdev.btree.refcount.moves: Records moved inside blocks during reference-count btree operations (count - U32) - *rate converted*



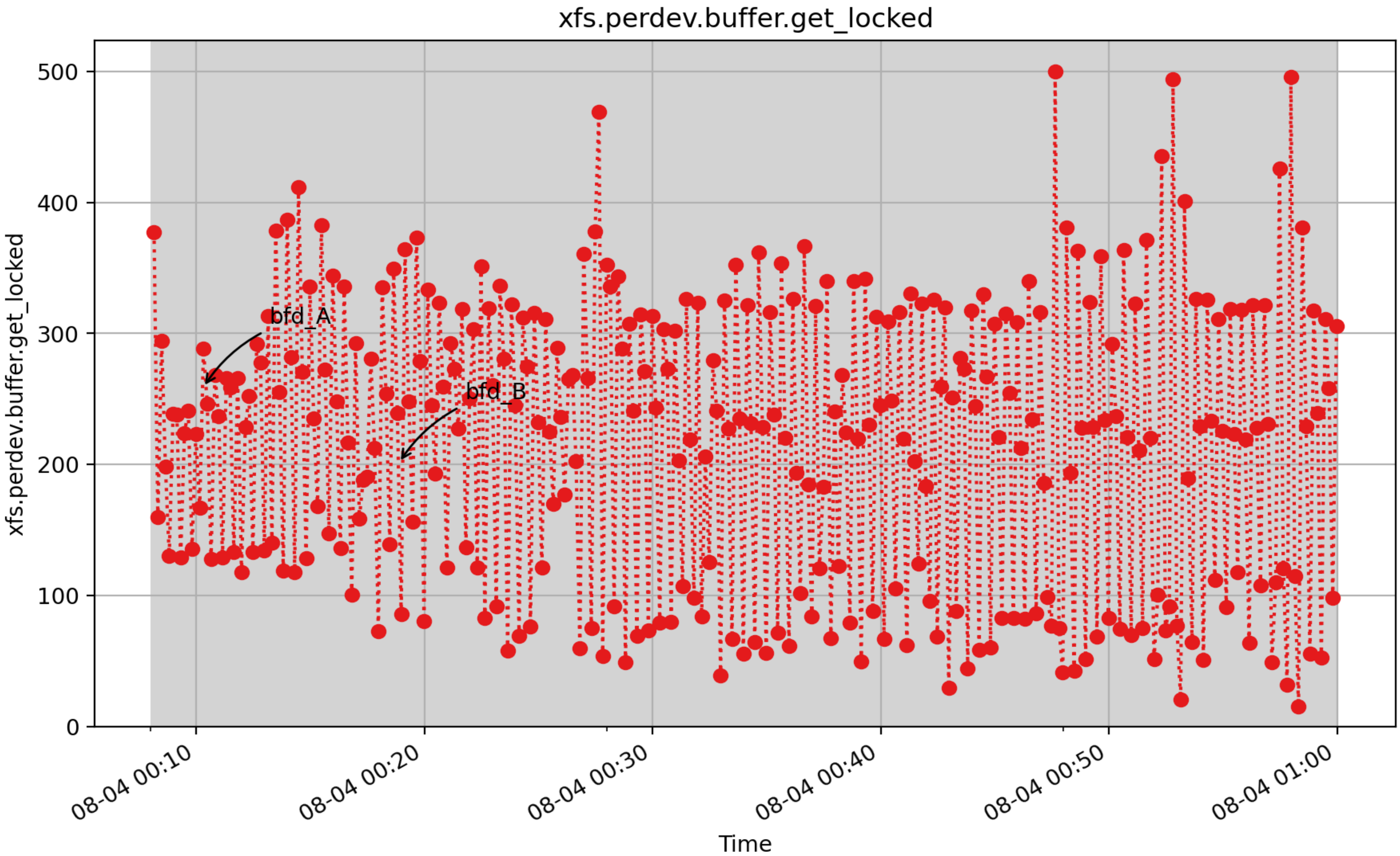
xfs.perdev.buffer.busy_locked: Number of times a new level is added to a free-inode btree (count - U32) - *rate converted*



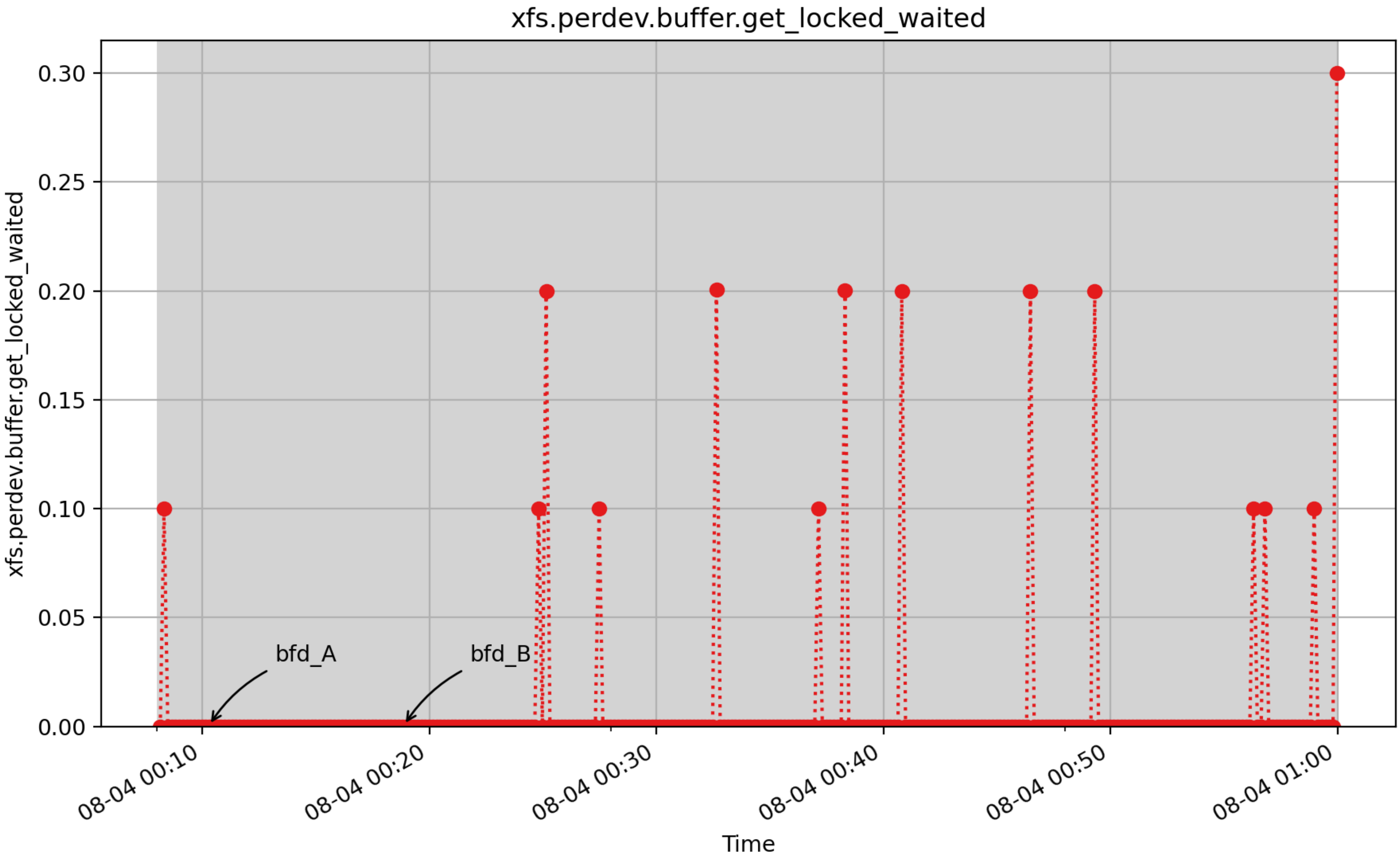
xfs.perdev.buffer.create: Number of free-inode btree record compares (count - U32) - *rate converted*



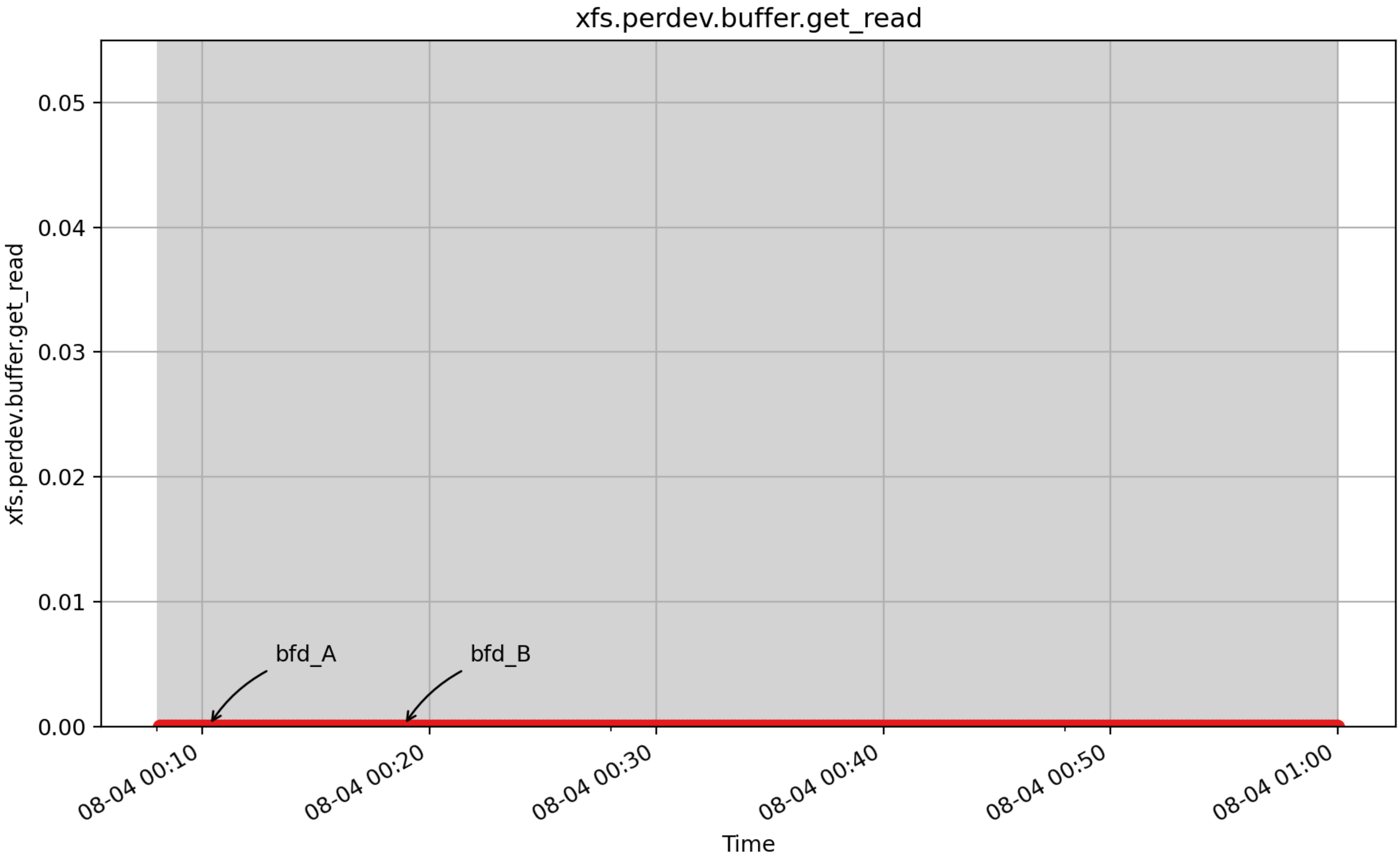
xfs.perdev.buffer.get: Number of free-inode btree record lookups (count - U32) - *rate converted*



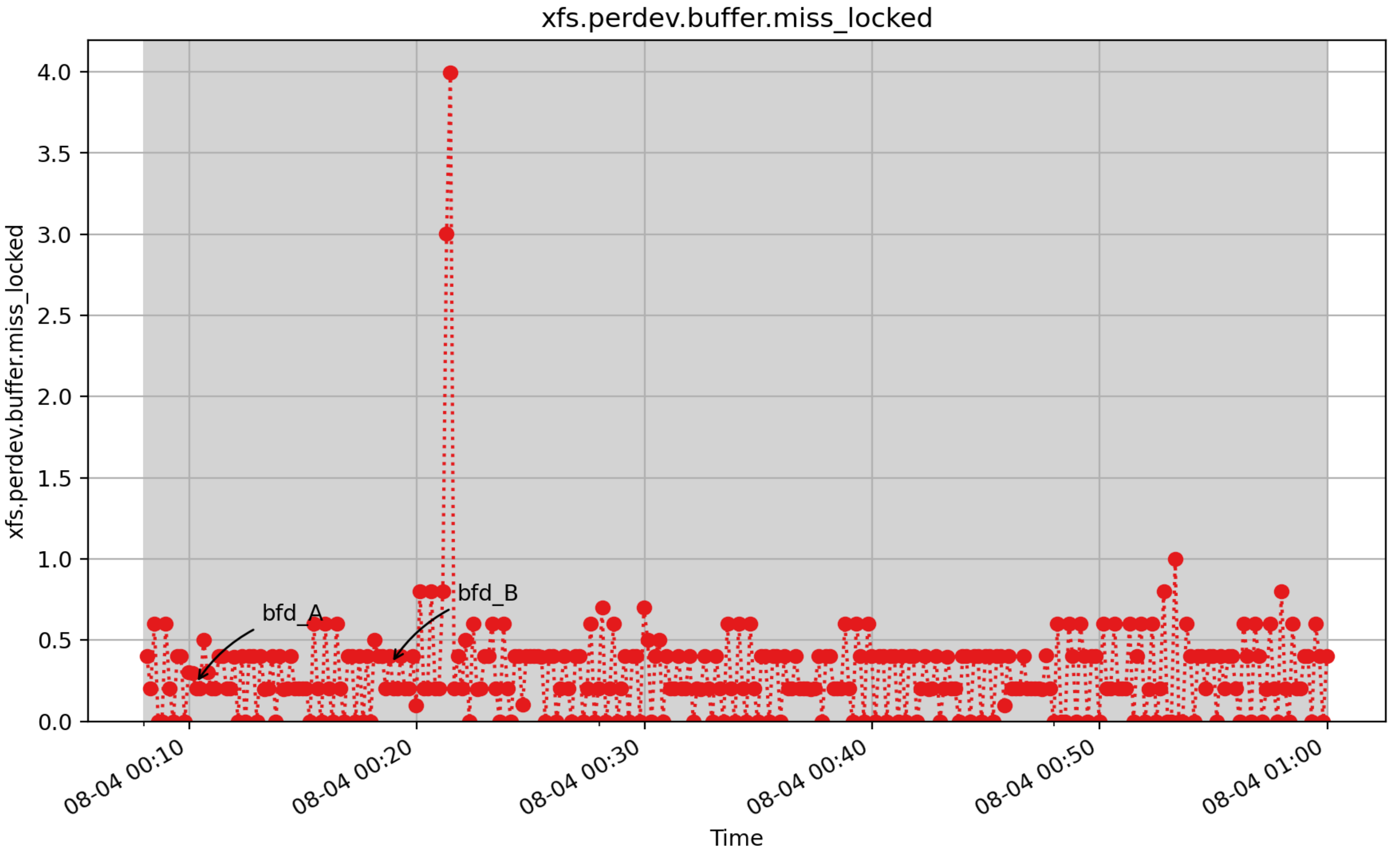
xfs.perdev.buffer.get_locked: Number of free-inode btree insert record operations executed (count - U32) - *rate converted*



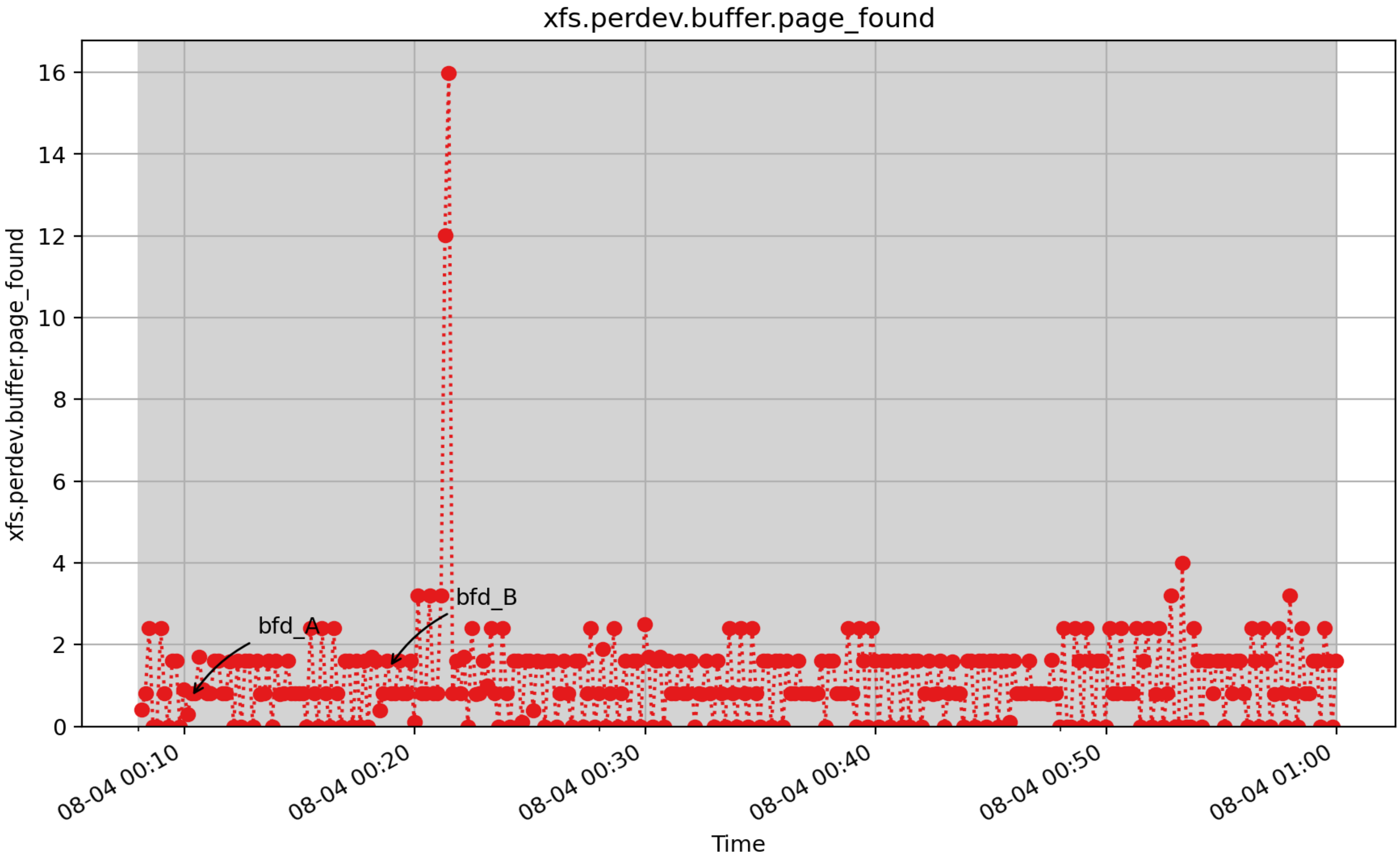
xfs.perdev.buffer.get_locked_waited: Number of free-inode btree delete record operations executed (count - U32) - *rate converted*



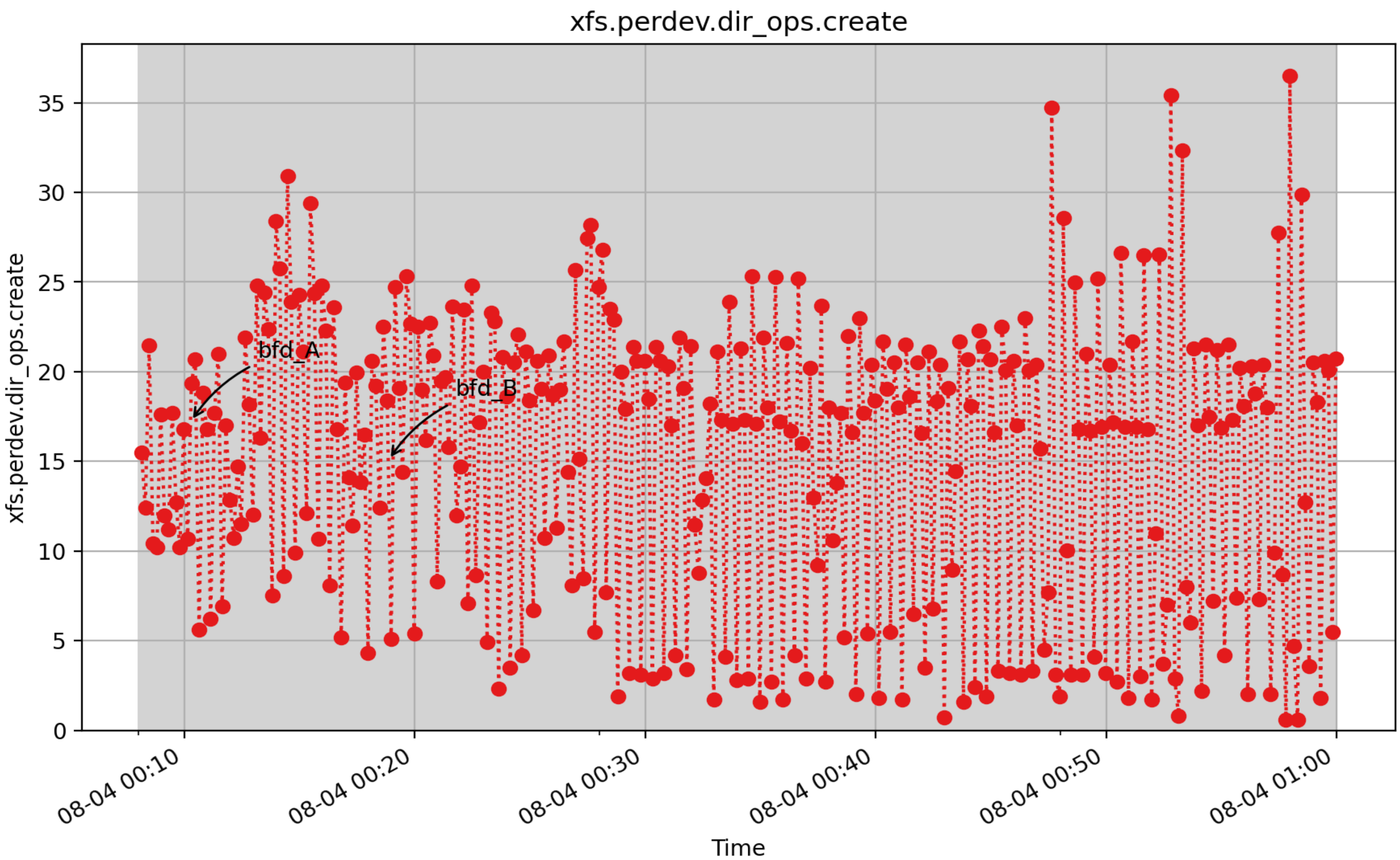
xfs.perdev.buffer.get_read: Left shift block operations to make space for a new free-inode btree record (count - U32) - *rate converted*



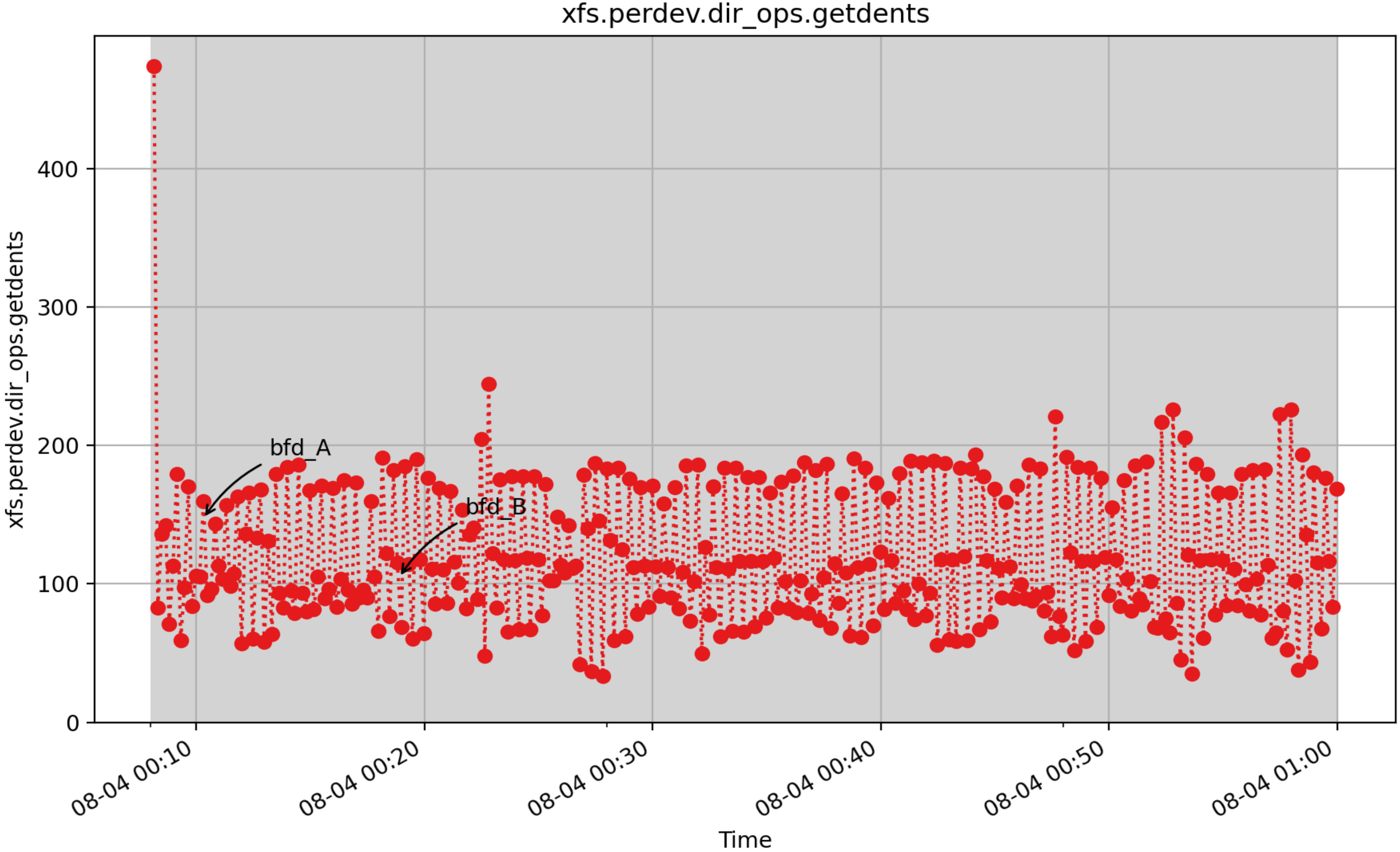
xfs.perdev.buffer.miss_locked: Number of times a level is removed from a free-inode btree (count - U32) - *rate converted*



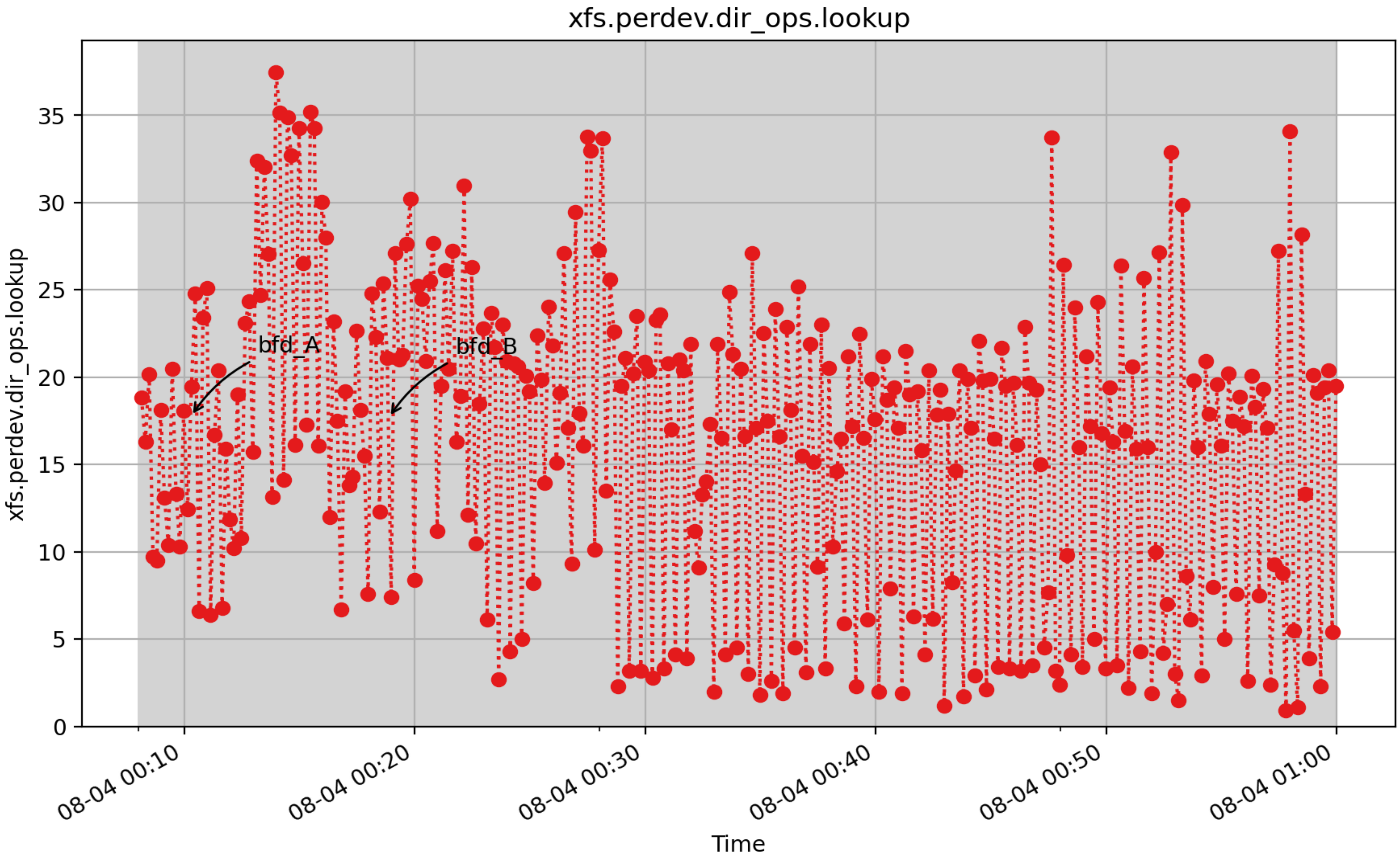
xfs.perdev.buffer.page_found: Number of times a cursor has been moved backward one free-inode btree record (count - U32) - *rate converted*



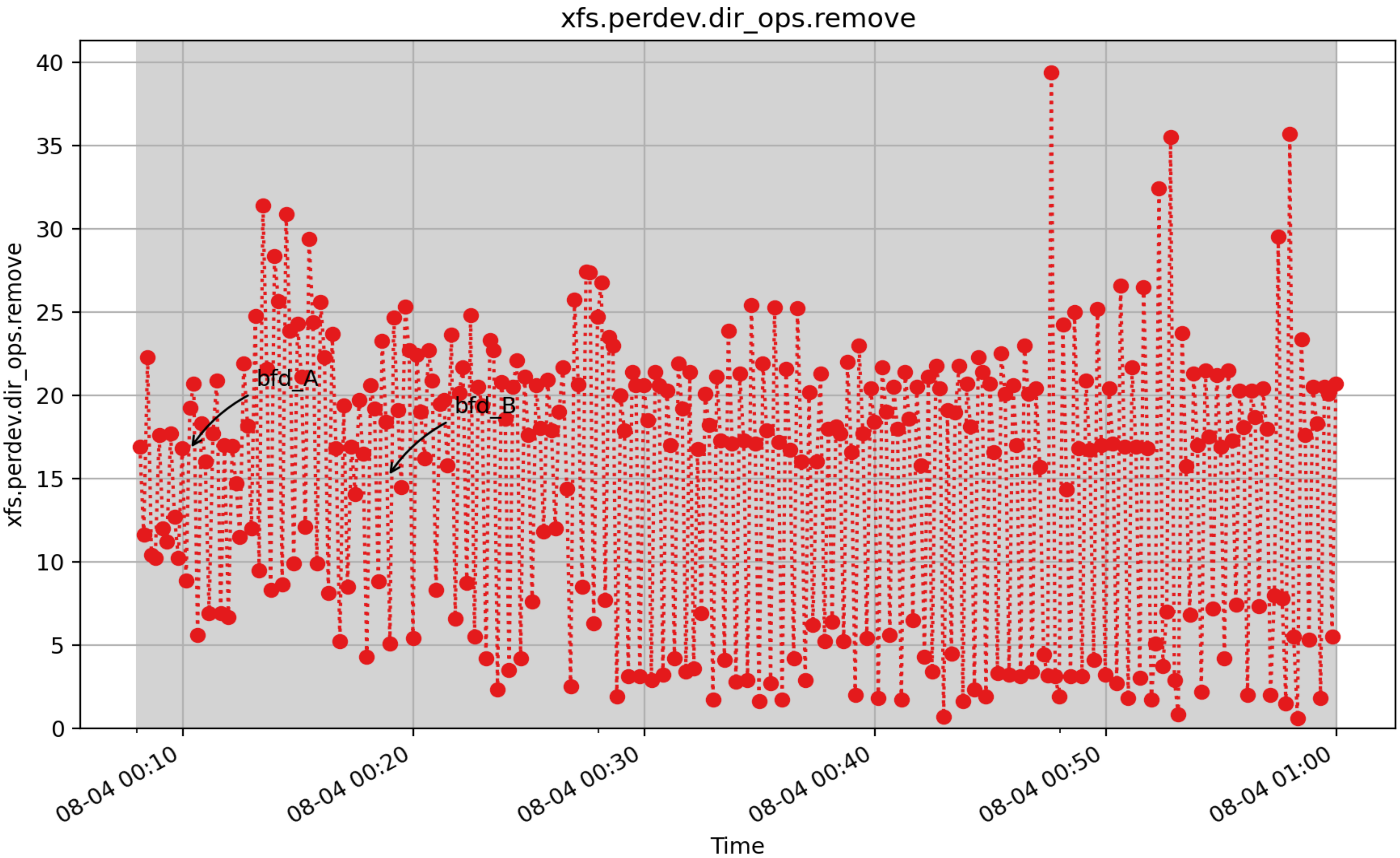
xfs.perdev.dir_ops.create: This is the number of times a new directory entry was created in XFS filesystems. Each time that a new file, directory, link, symbolic link, or special file is created in the directory hierarchy the count is incremented. (count - U32) - *rate converted*



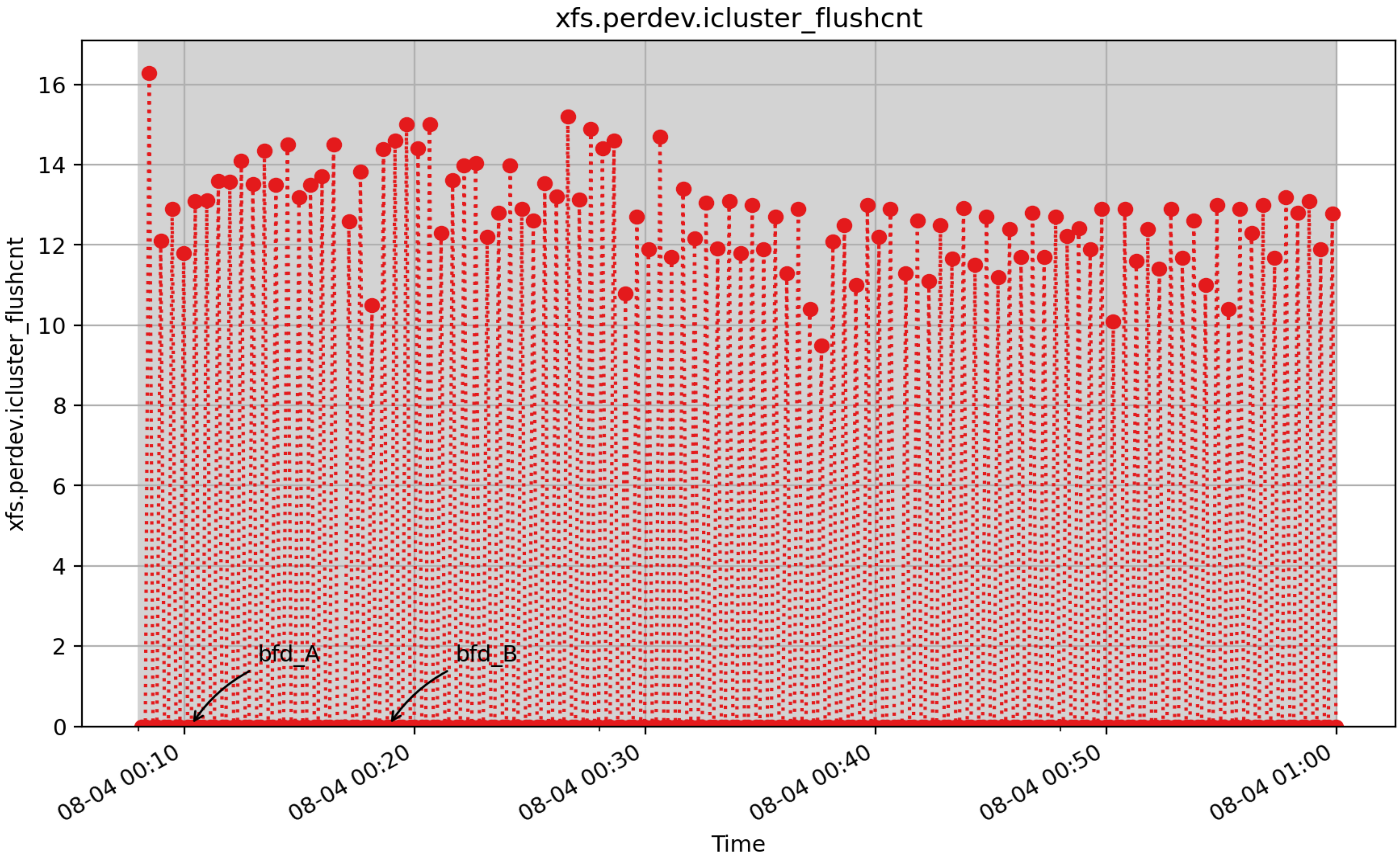
xfs.perdev.dir_ops.getdents: This is the number of times the XFS directory getdents operation was performed. The getdents operation is used by programs to read the contents of directories in a file system independent fashion. This count corresponds exactly to the number of times the getdents(2) system call was successfully used on an XFS directory. (count - U32) - *rate converted*



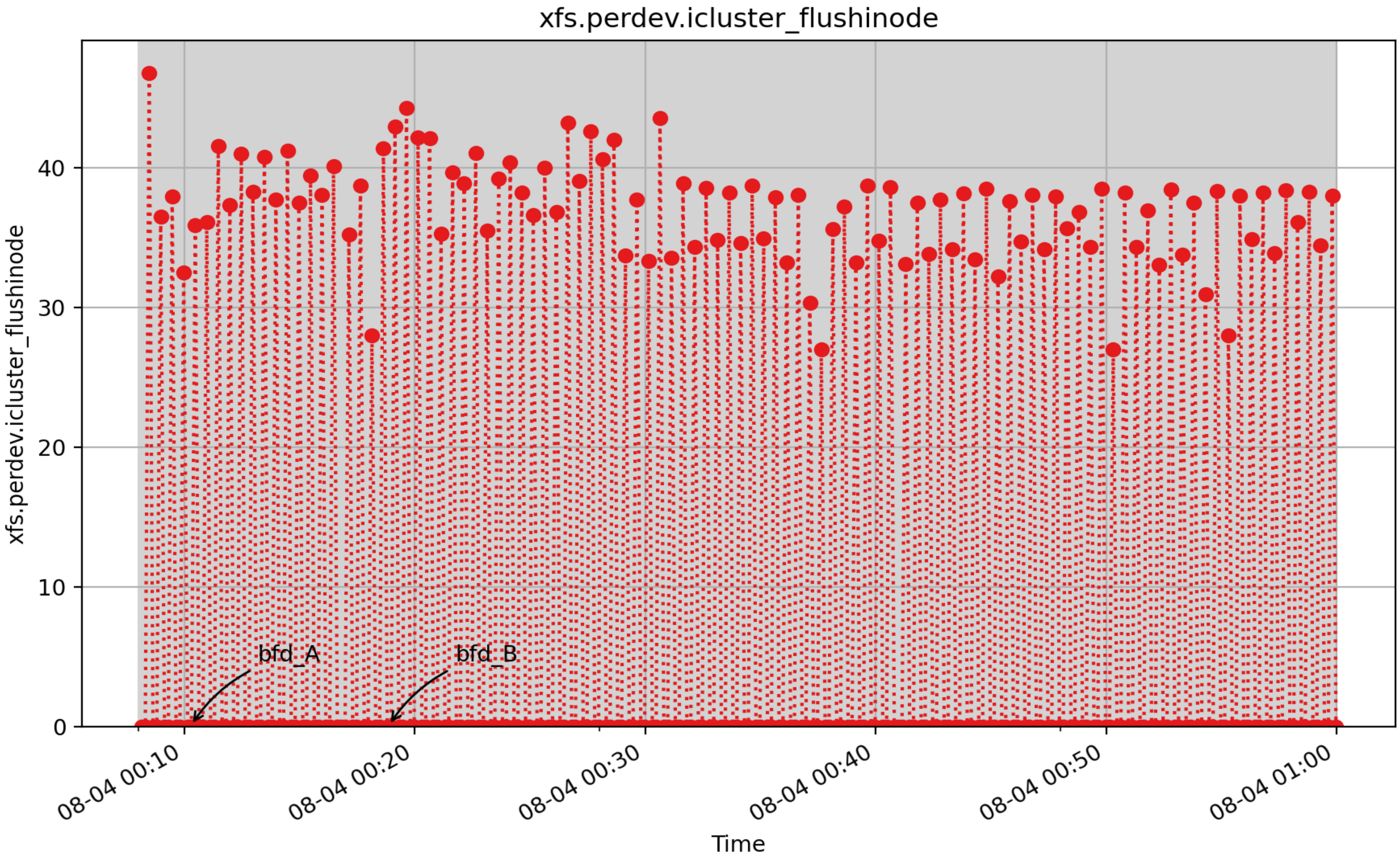
xfs.perdev.dir_ops.lookup: This is a count of the number of file name directory lookups in XFS filesystems. It counts only those lookups which miss in the operating system's directory name lookup cache and must search the real directory structure for the name in question. The count is incremented once for each level of a pathname search that results in a directory lookup. (count - U32) - *rate converted*



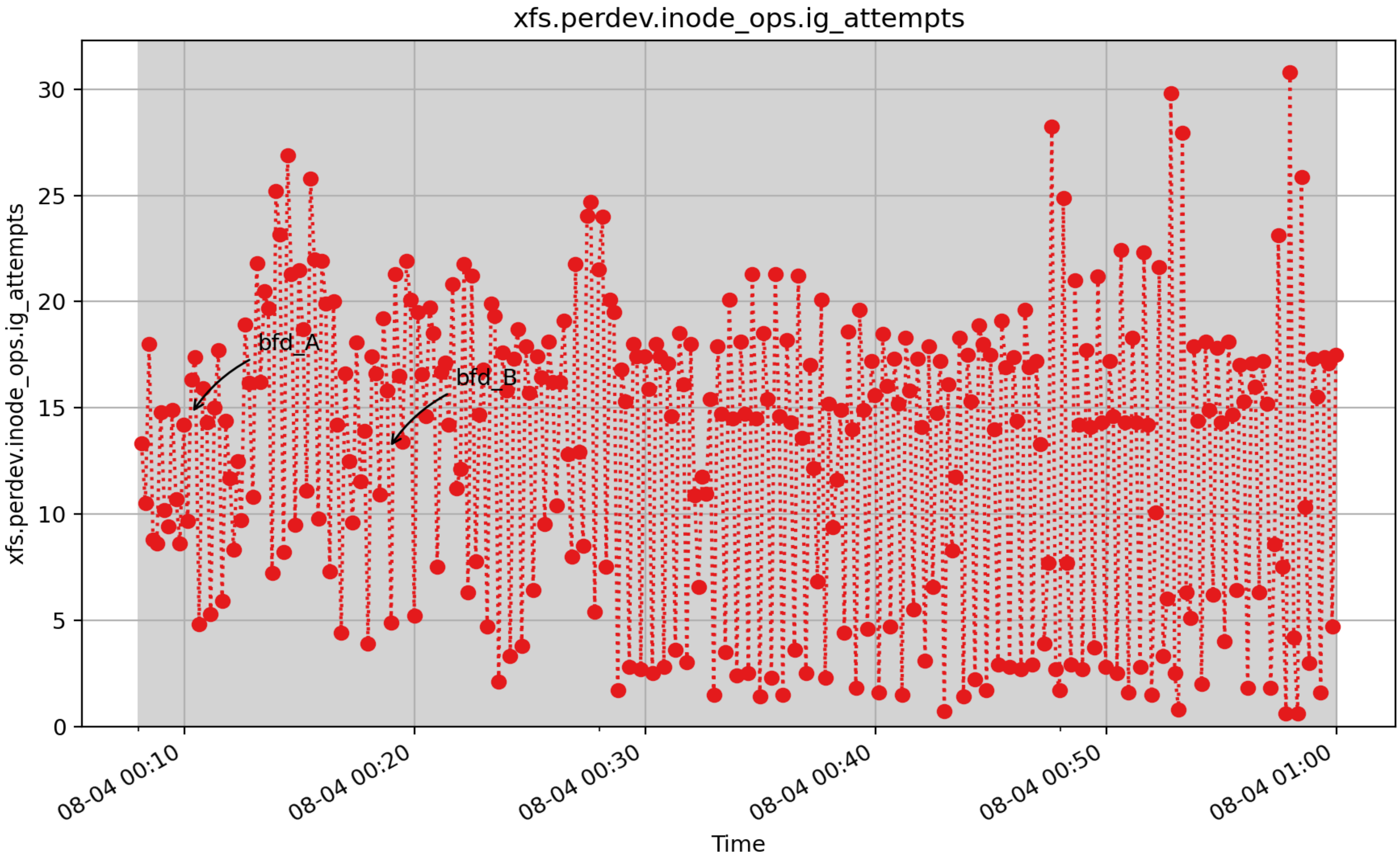
xfs.perdev.dir_ops.remove: This is the number of times an existing directory entry was removed in XFS filesystems. Each time that a file, directory, link, symbolic link, or special file is removed from the directory hierarchy the count is incremented. (count - U32) - *rate converted*



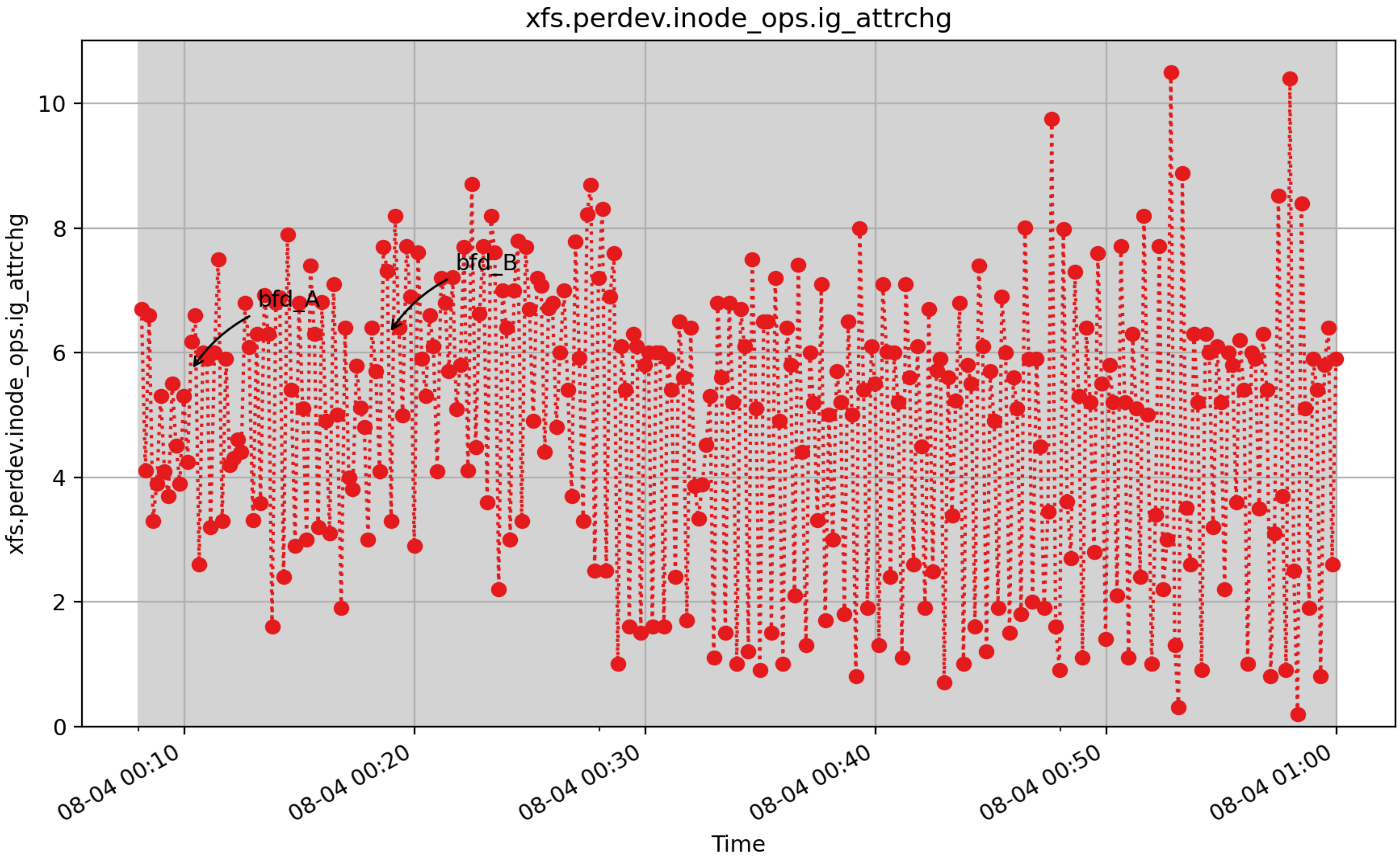
xfs.perdev.icluster_flushcnt: value from xs_icluster_flushcnt field of struct xfsstats (count - U32) - *rate converted*



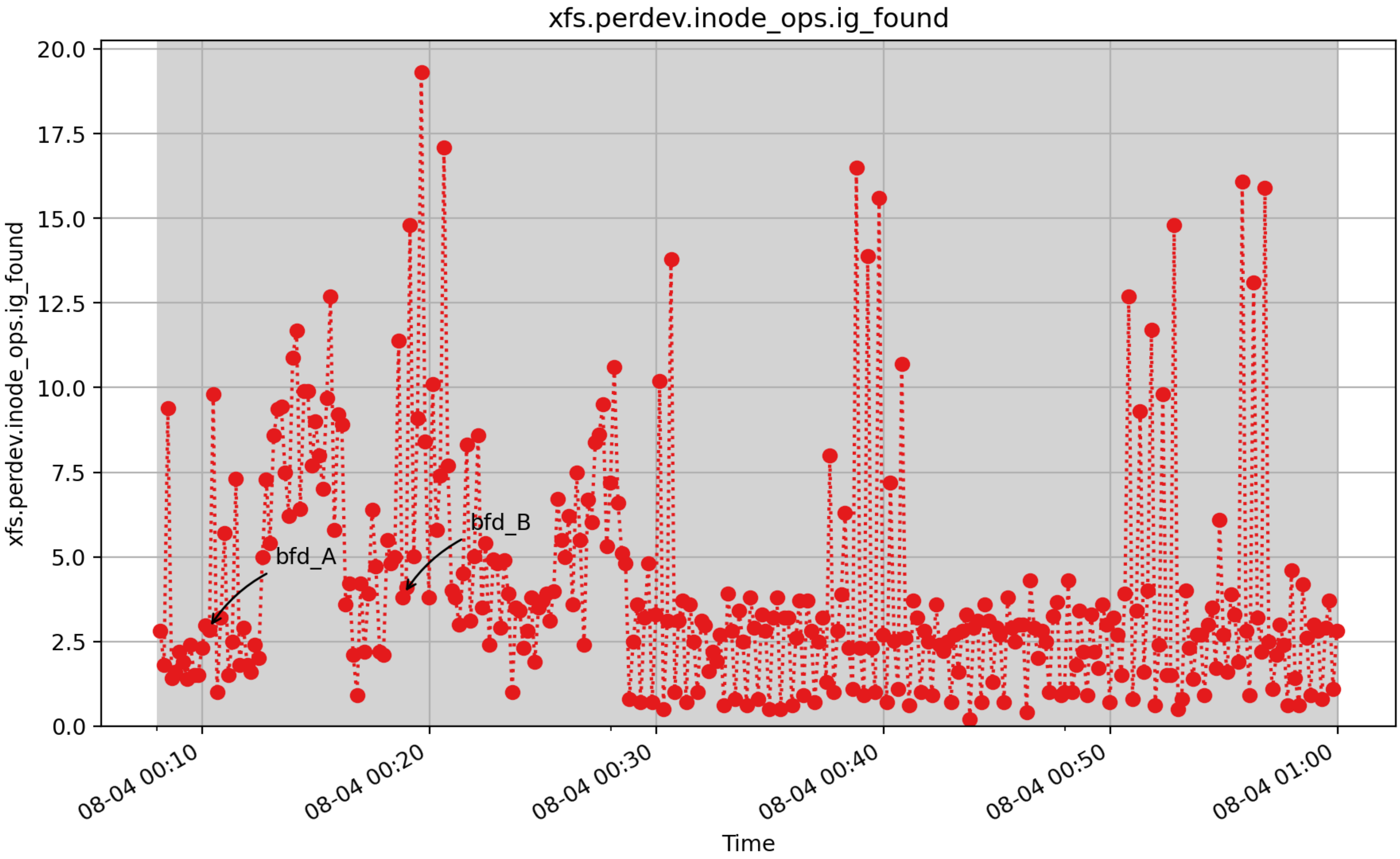
xfs.perdev.icluster_flushinode: This is the number of times that the inode clustering was not able to flush anything but the one inode it was called with. (count - U32) - *rate converted*



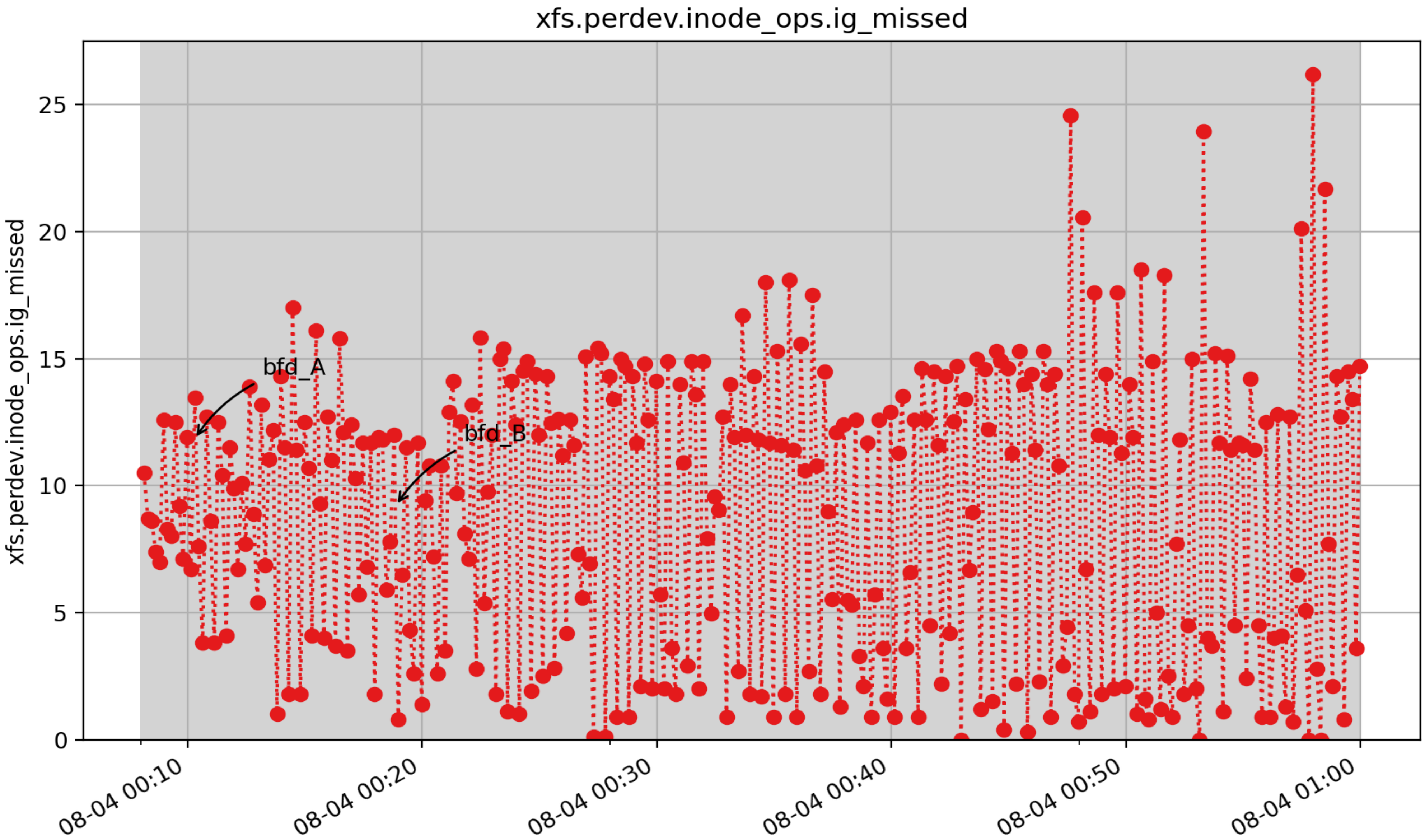
xfs.perdev.inode_ops.ig_attempts: This is the number of times the operating system looked for an XFS inode in the inode cache. Whether the inode was found in the cache or needed to be read in from the disk is not indicated here, but this can be computed from the ig_found and ig_missed counts. (count - U32) - *rate converted*



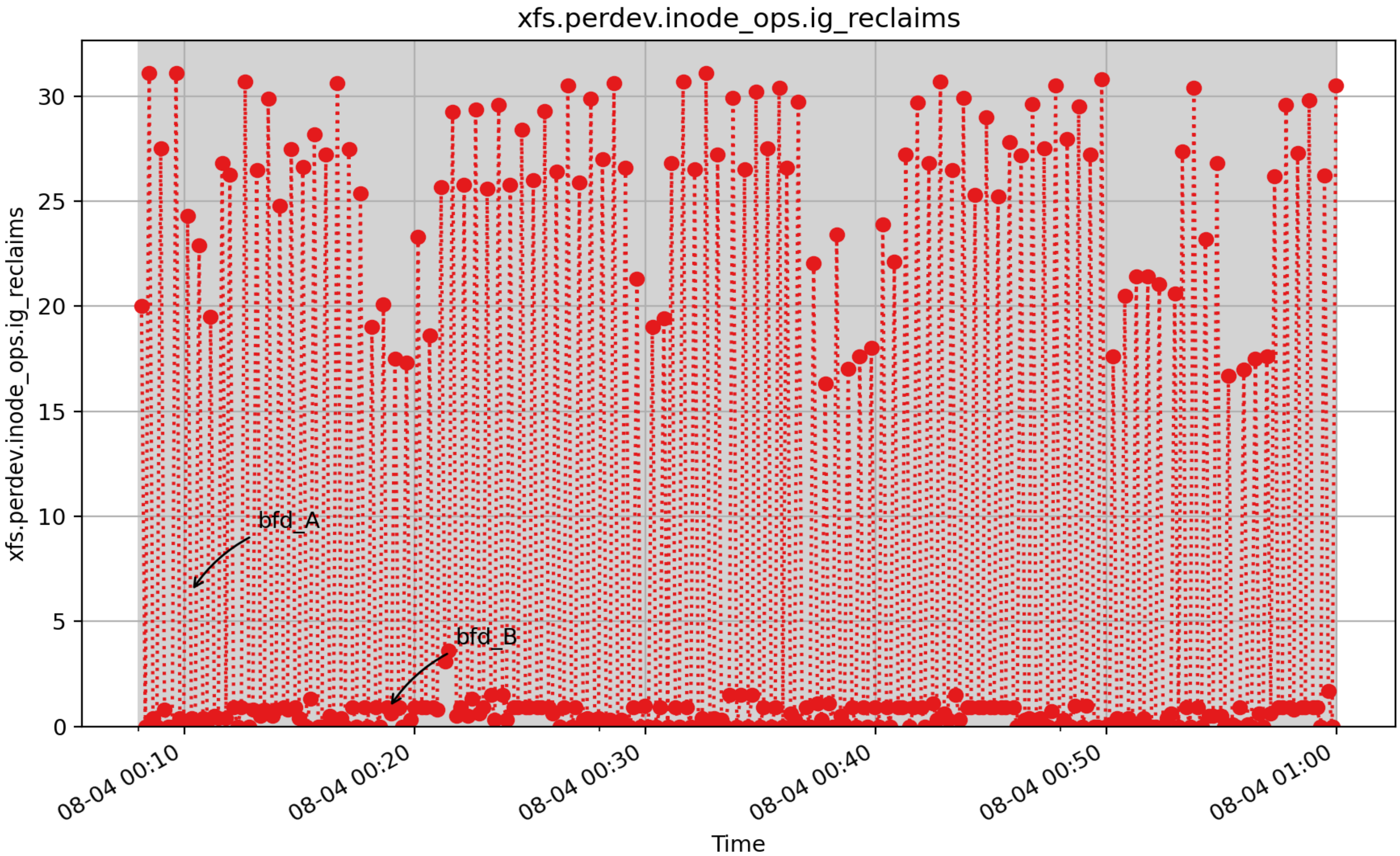
xfs.perdev.inode_ops.ig_attrchg: This is the number of times the operating system explicitly changed the attributes of an XFS inode. For example, this could be to change the inode's owner, the inode's size, or the inode's timestamps. (count - U32) - *rate converted*



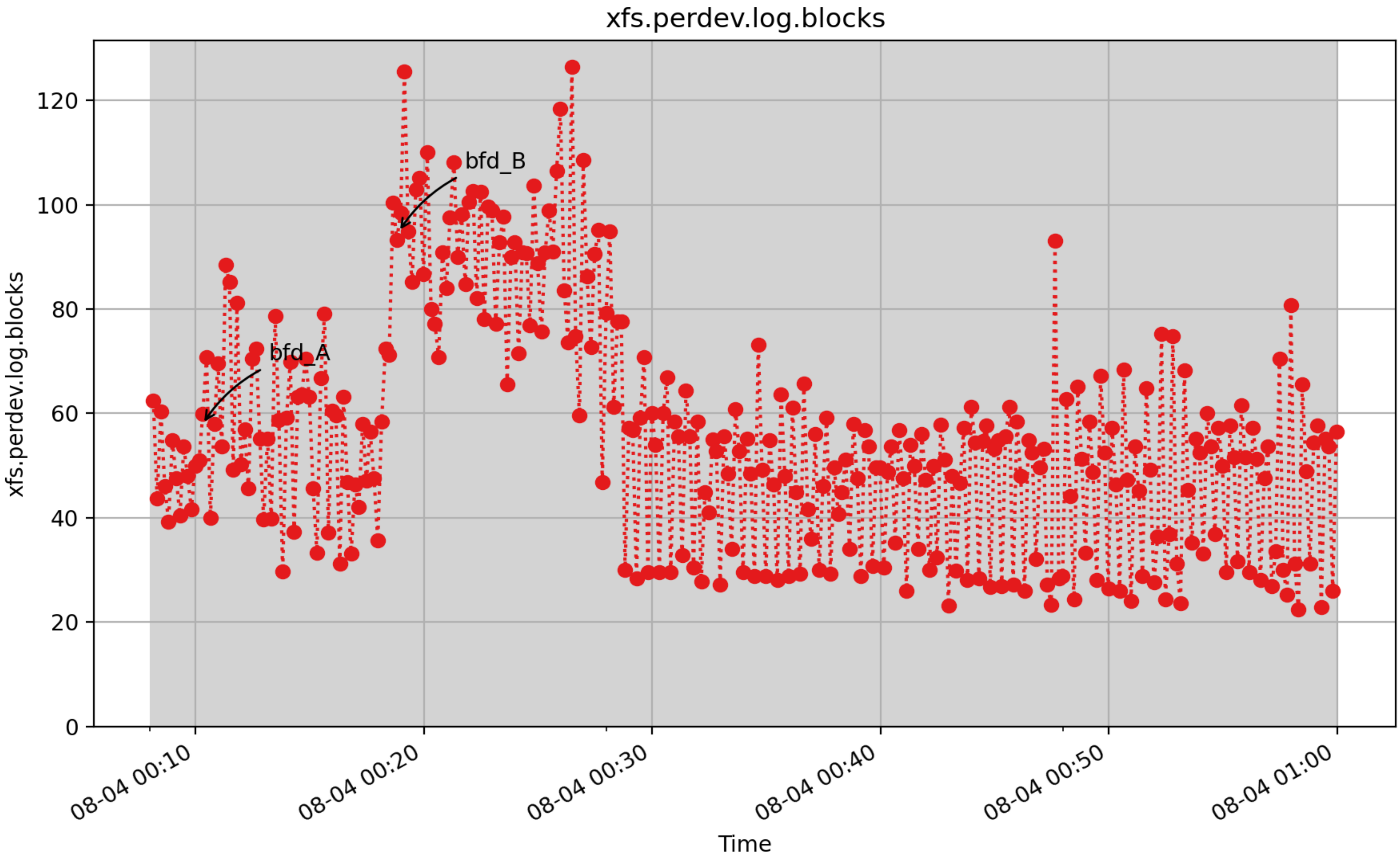
xfs.perdev.inode_ops.ig_found: This is the number of times the operating system looked for an XFS inode in the inode cache and found it. The closer this count is to the ig_attempts count the better the inode cache is performing. (count - U32) - *rate converted*



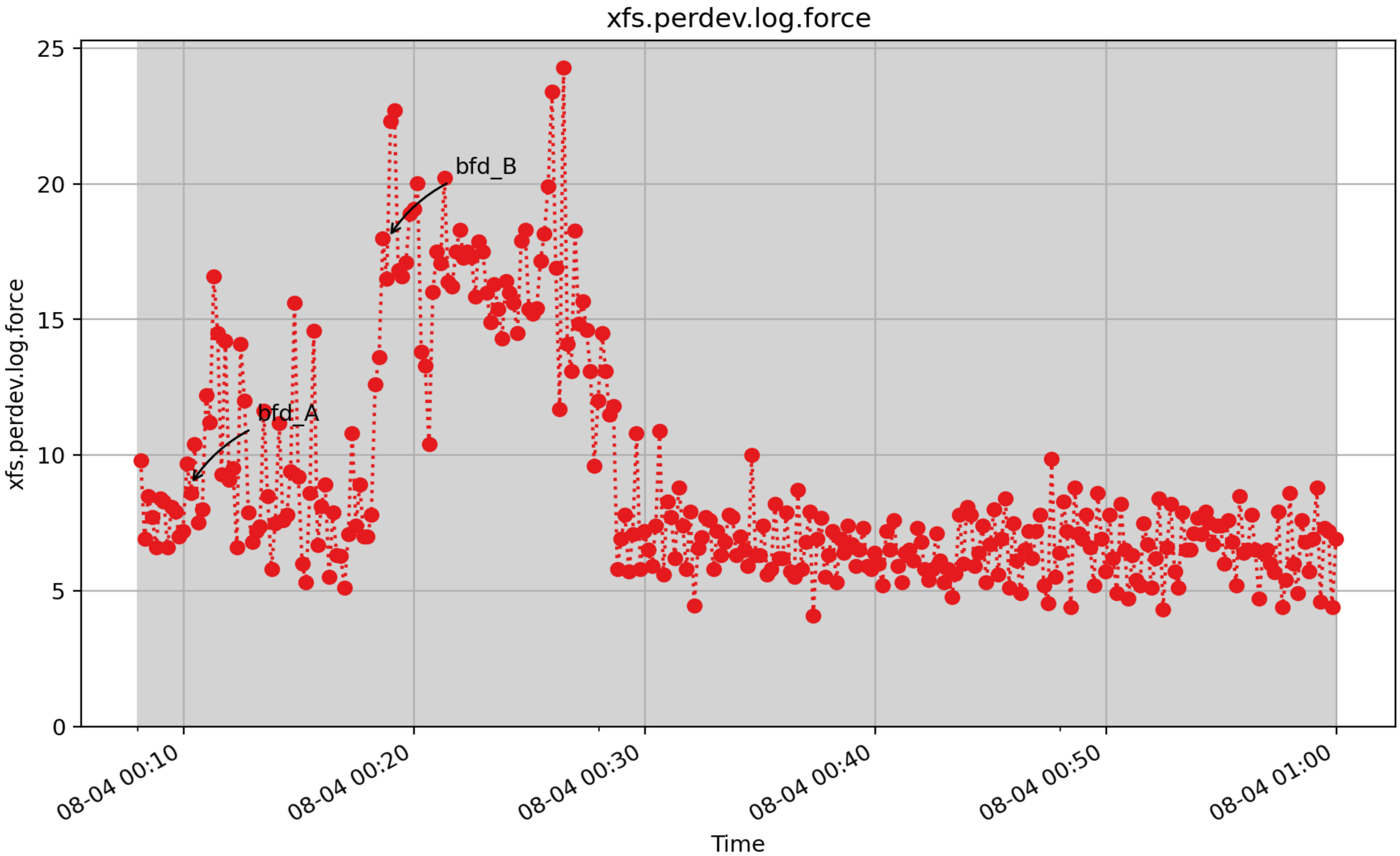
xfs.perdev.inode_ops.ig_missed: This is the number of times the operating system looked for an XFS inode in the inode cache and the inode was not there. The further this count is from the ig_attempts count the better. (count - U32) - *rate converted*



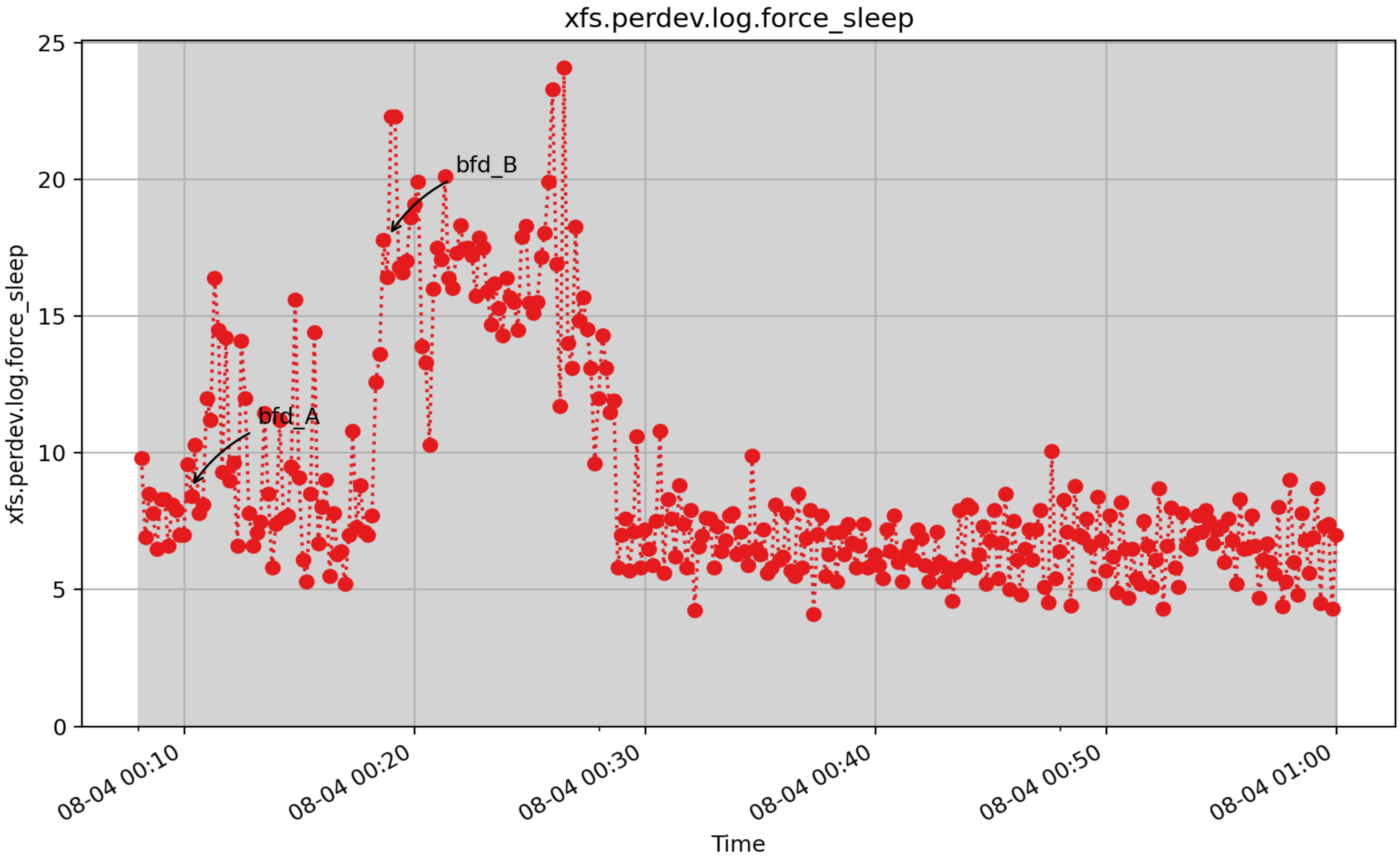
xfs.perdev.inode_ops.ig_reclaims: This is the number of times the operating system recycled an XFS inode from the inode cache in order to use the memory for that inode for another purpose. Inodes are recycled in order to keep the inode cache from growing without bound. If the reclaim rate is high it may be beneficial to raise the vnode_free_ratio kernel tunable variable to increase the size of inode cache. (count - U32) - *rate* *converted*



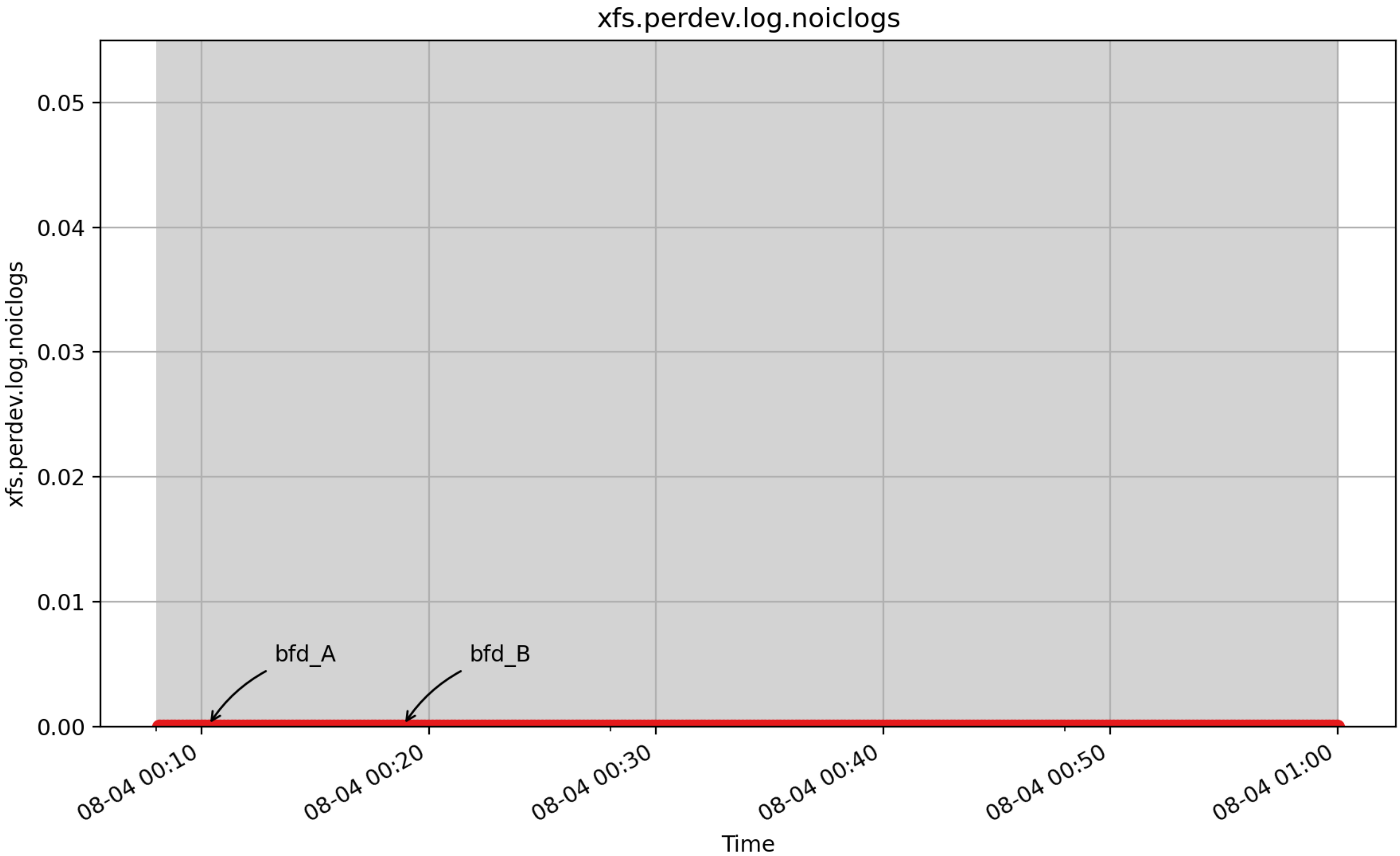
xfs.perdev.log.blocks: This variable counts the number of Kbytes of information being written to the physical log partitions of XFS filesystems. Log data traffic is proportional to the level of meta-data updating. The rate with which log data gets written depends on the size of internal log buffers and disk write speed. Therefore, filesystems with very high meta-data updating may need to stripe the log partition or put the log partition on a separate drive. (Kbyte - U32) - *rate converted*



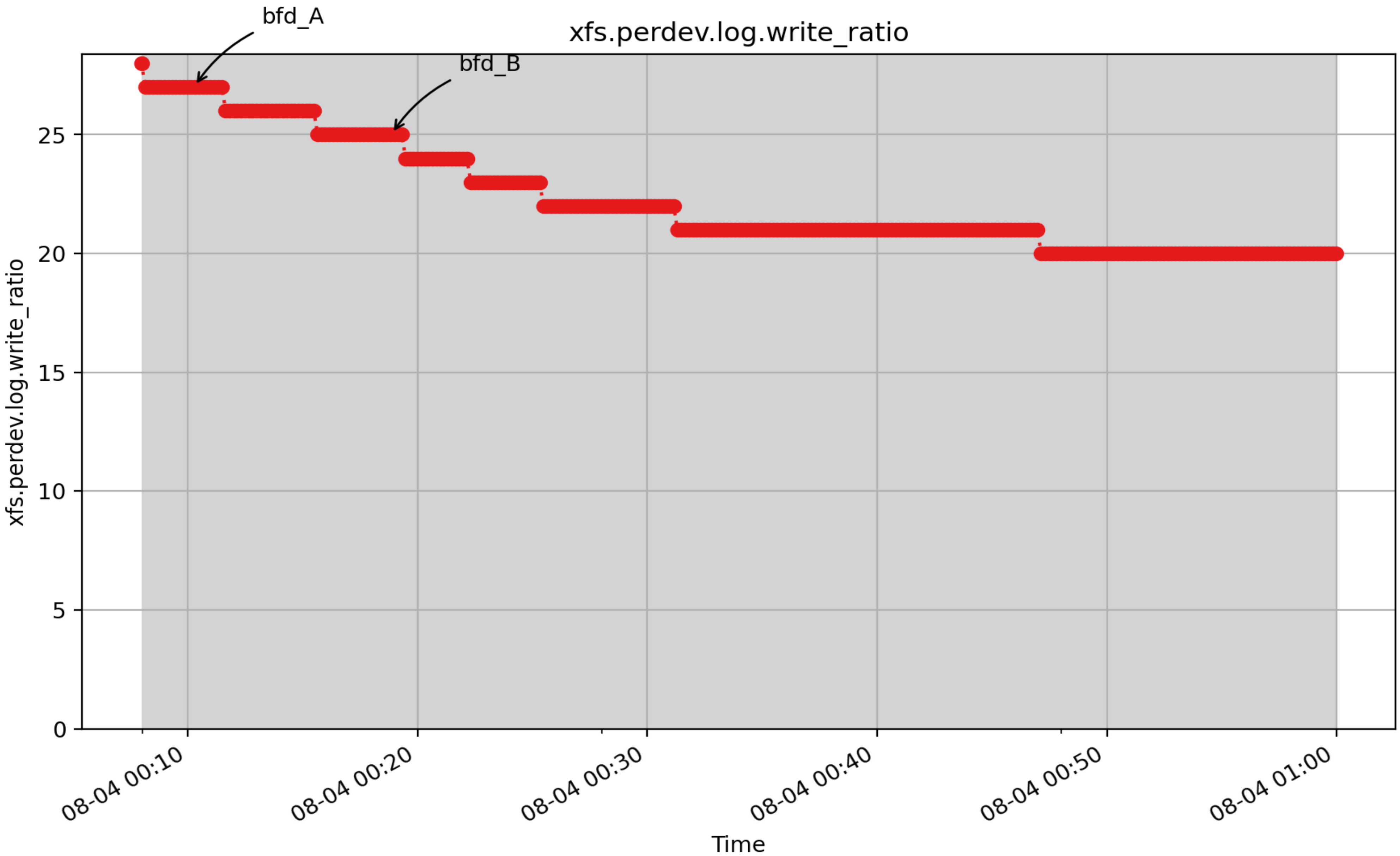
xfs.perdev.log.force: The number of times the in-core log is forced to disk. It is equivalent to the number of successful calls to the function xfs_log_force(). (count - U32) - *rate converted*



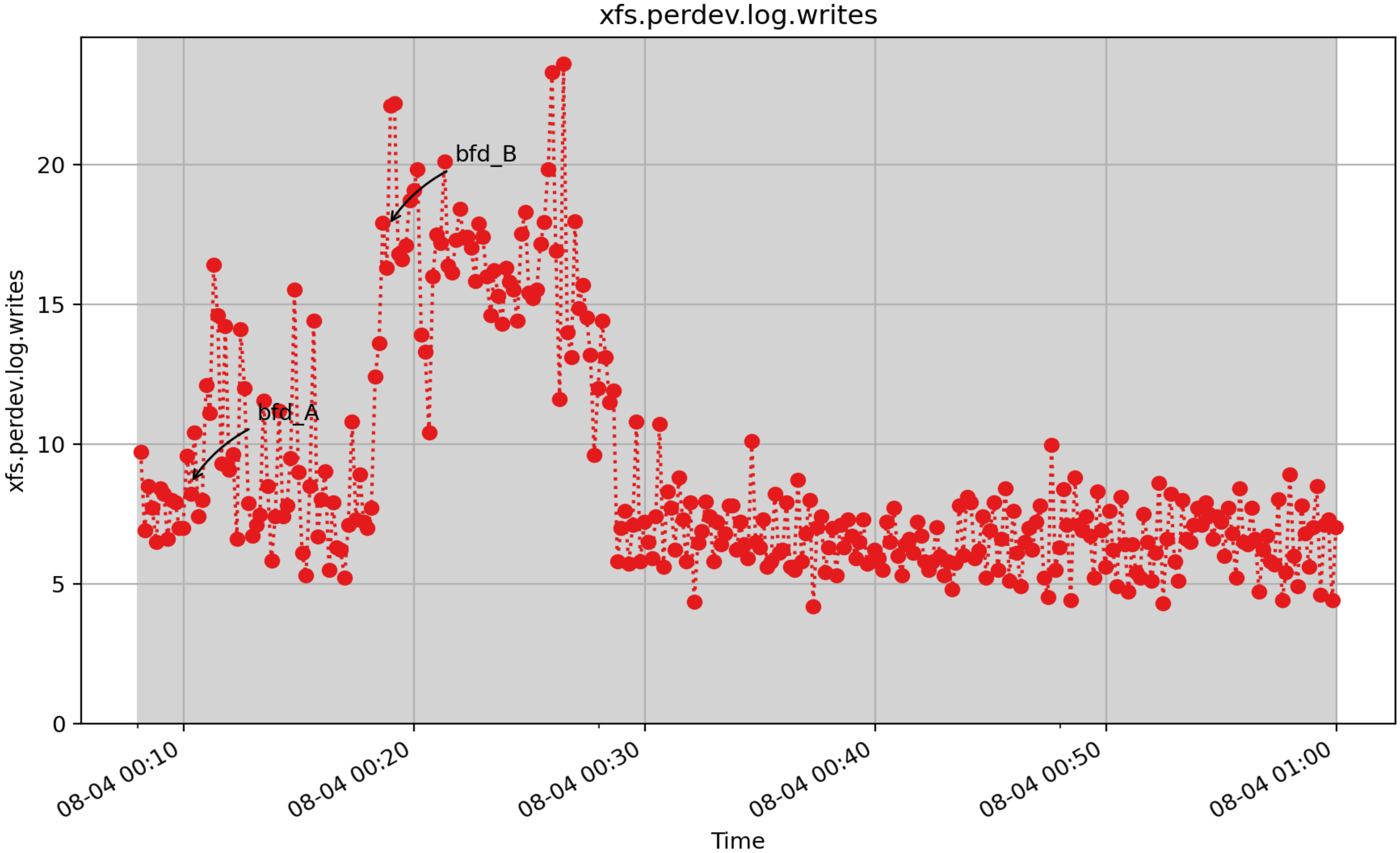
xfs.perdev.log.force_sleep: This metric is exported from the xs_log_force_sleep field of struct xfsstats (count - U32) - *rate converted*



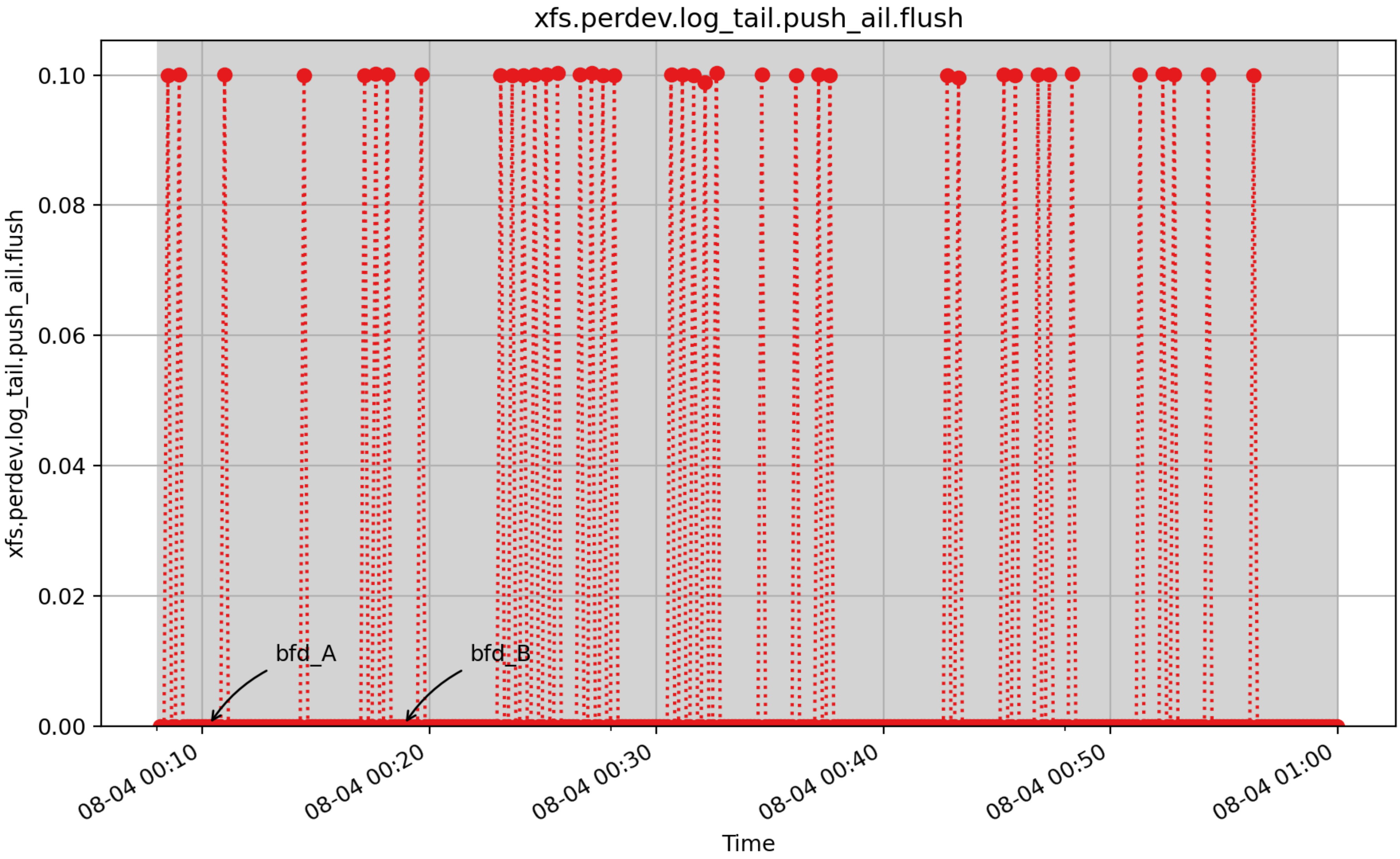
xfs.perdev.log.noiclogs: This variable keeps track of times when a logged transaction can not get any log buffer space. When this occurs, all of the internal log buffers are busy flushing their data to the physical on-disk log. (count - U32) - *rate converted*



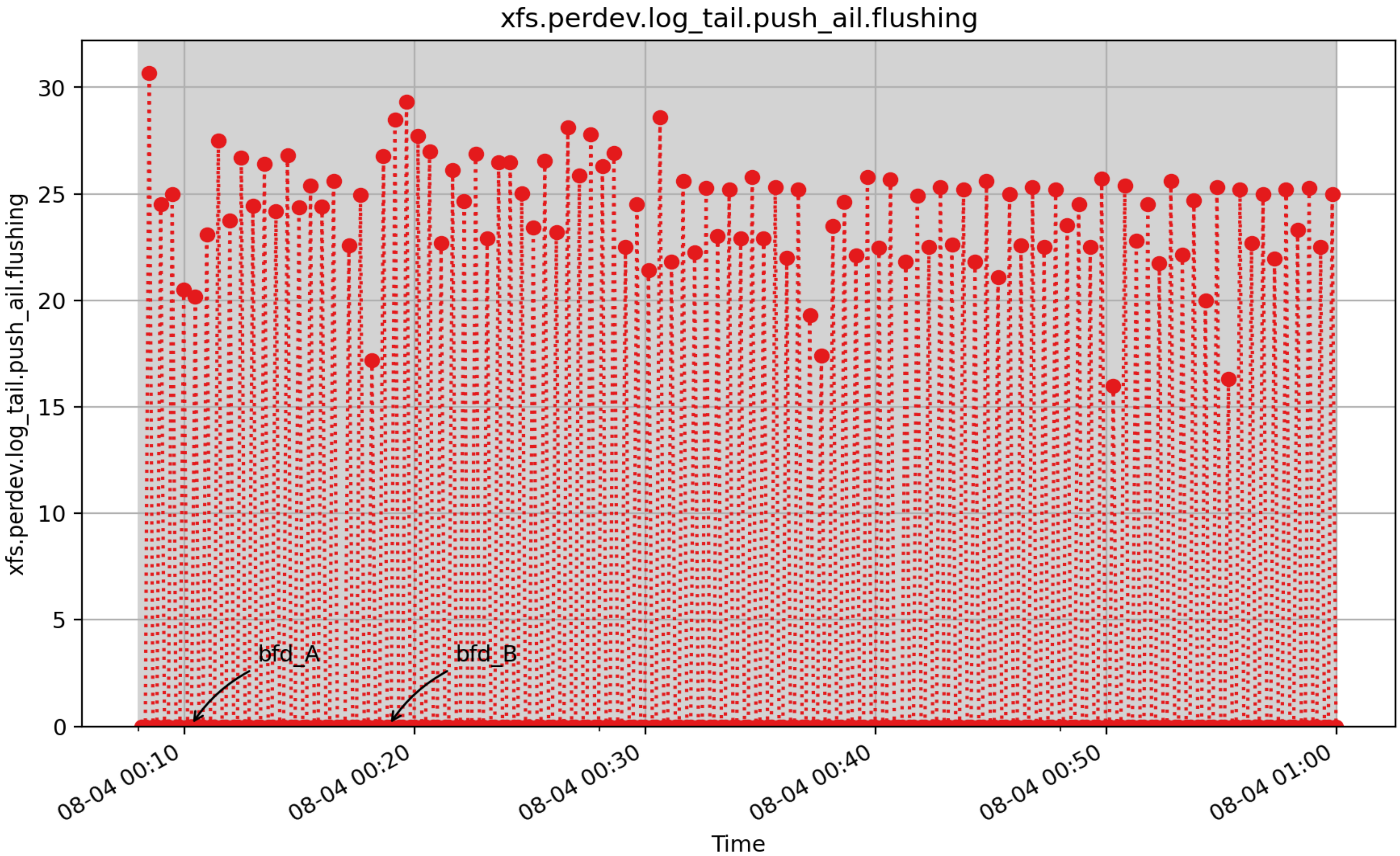
xfs.perdev.log.write_ratio: The ratio of log blocks written to log writes. If block count isn't a "reasonable" multiple of writes, then many small log writes are being performed - this is suboptimal. Perfection is 64. Fine-grain control can be obtained when this metric is used in conjuntion with pmstore(1) and the xfs.control.reset metric. (- FLOAT)



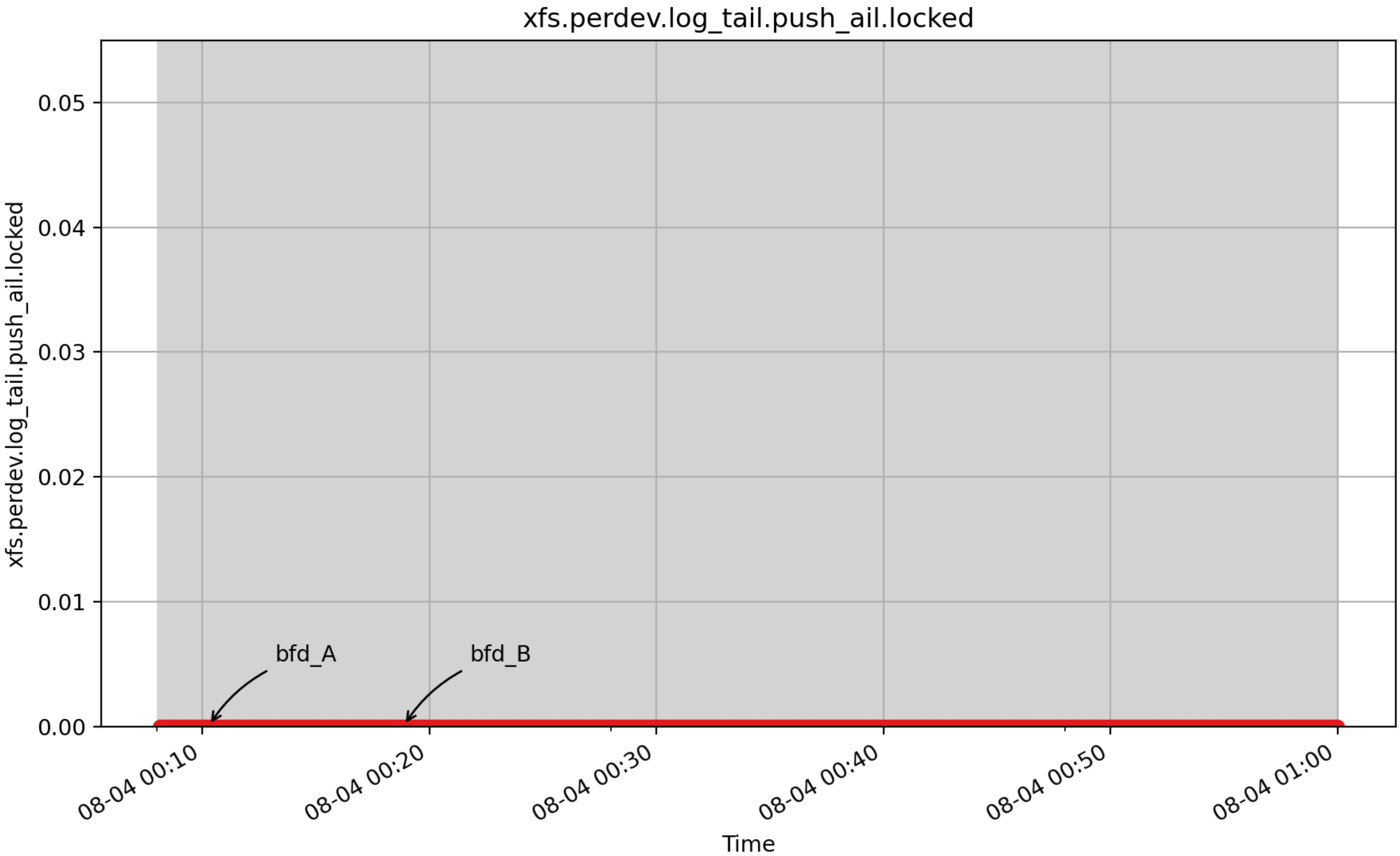
xfs.perdev.log.writes: This variable counts the number of log buffer writes going to the physical log partitions of XFS filesystems. Log data traffic is proportional to the level of meta-data updating. Log buffer writes get generated when they fill up or external syncs occur. (count - U32) - *rate converted*



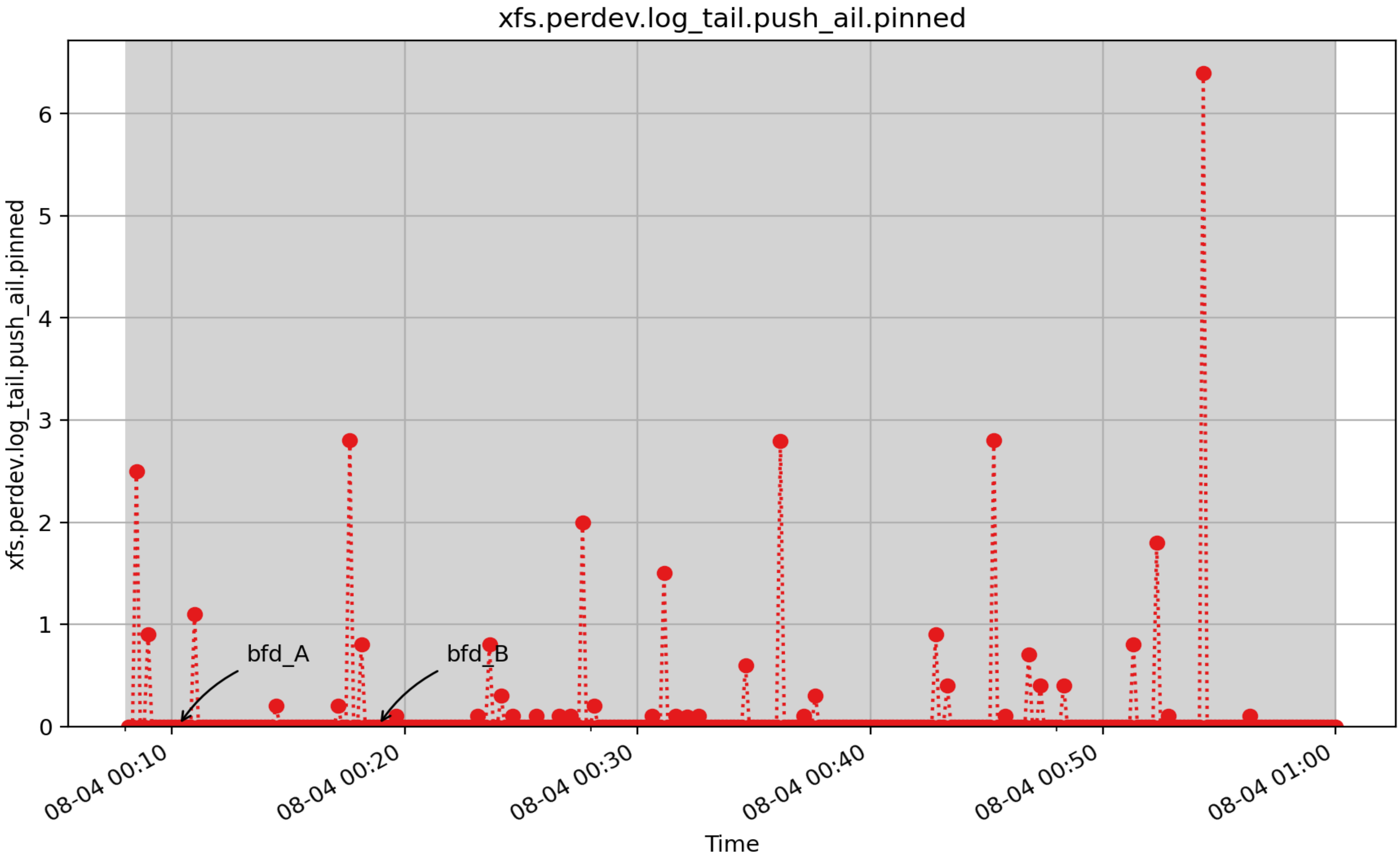
xfs.perdev.log_tail.push_ail.flush: value from xs_push_ail_flush field of struct xfsstats (count - U32) - *rate converted*



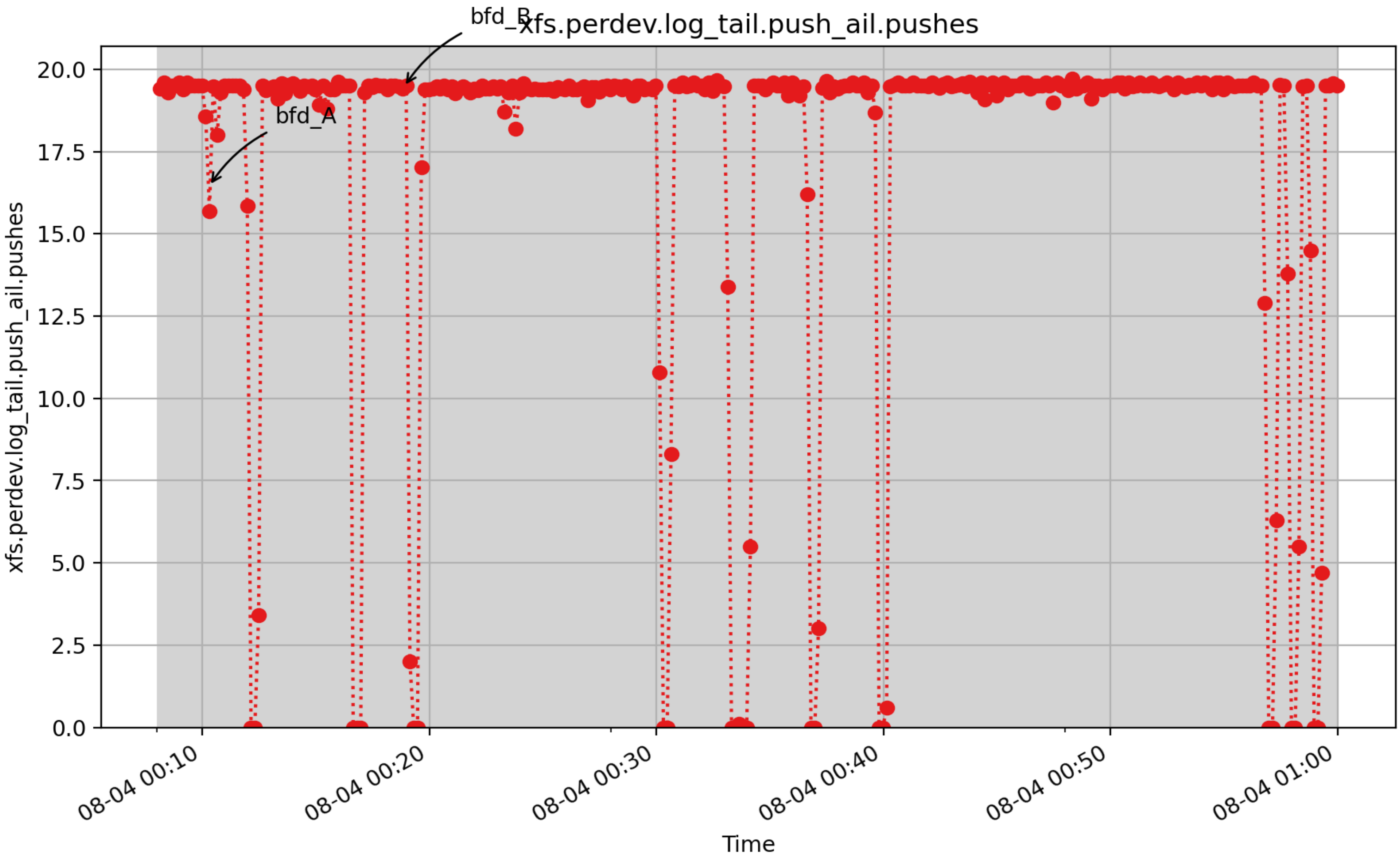
xfs.perdev.log_tail.push_ail.flushing: value from xs_push_ail_flushing field of struct xfsstats (count - U32) - *rate converted*



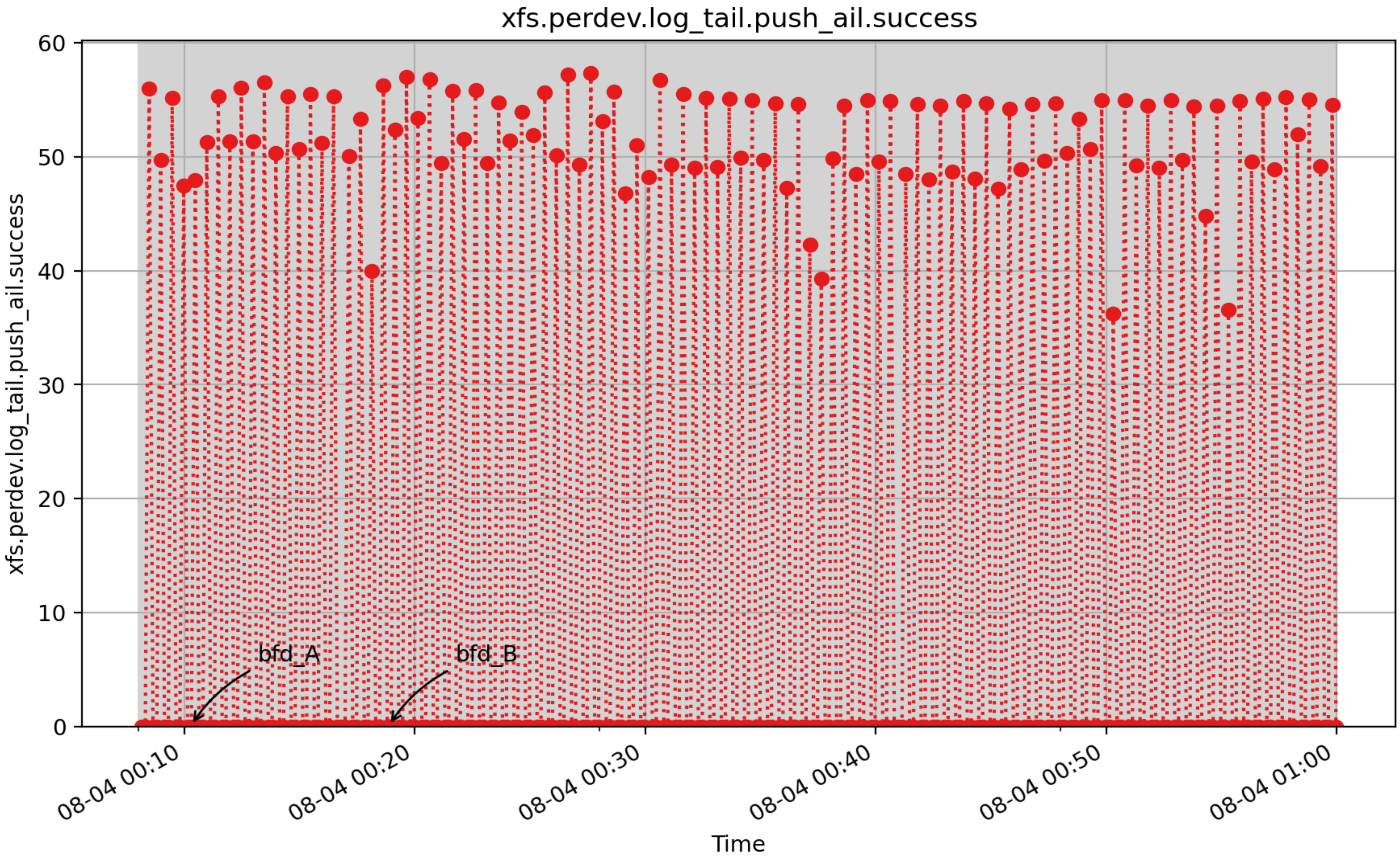
xfs.perdev.log_tail.push_ail.locked: value from xs_push_ail_locked field of struct xfsstats (count - U32) - *rate converted*



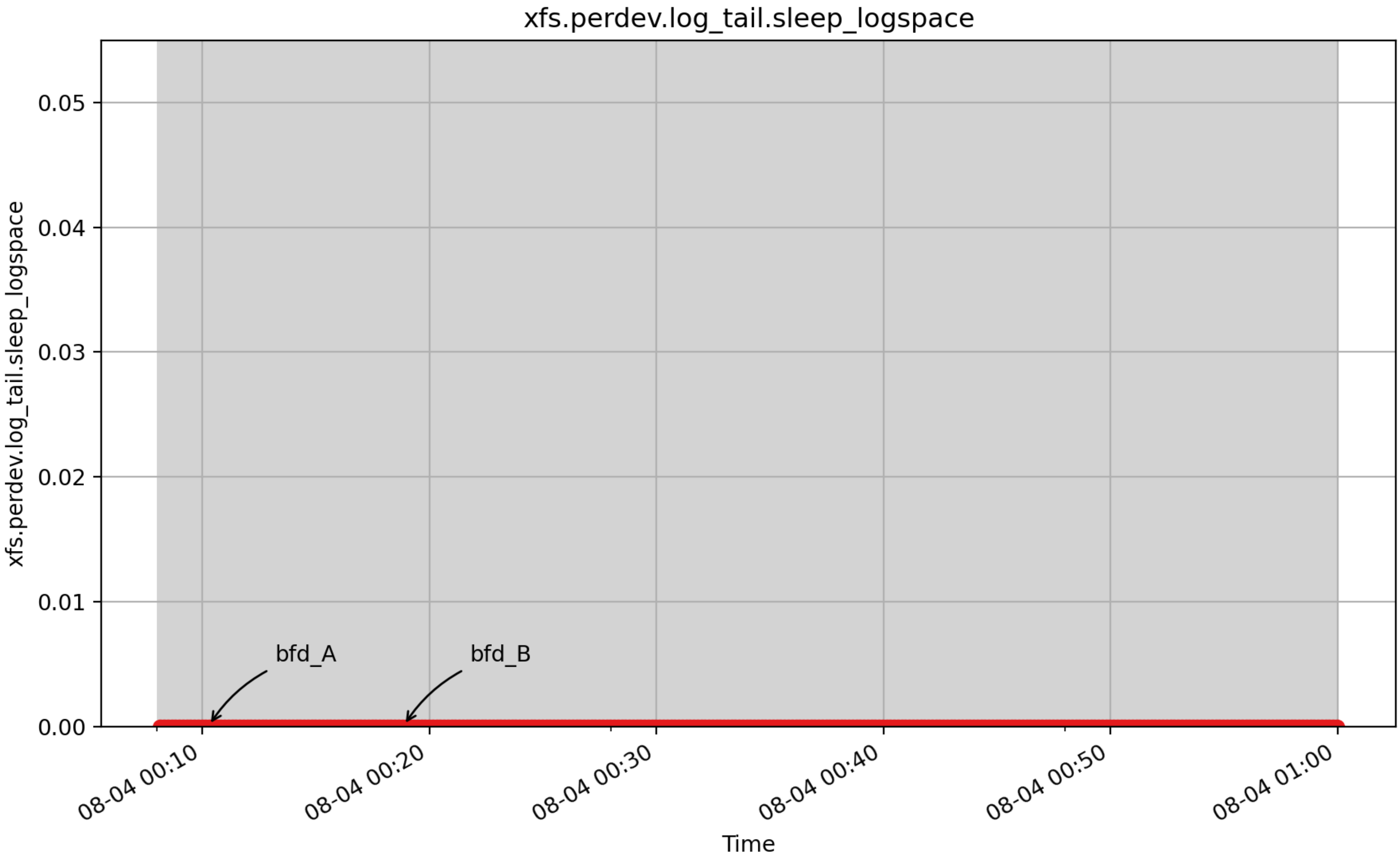
xfs.perdev.log_tail.push_ail.pinned: value from xs_push_ail_pinned field of struct xfsstats (count - U32) - *rate converted*



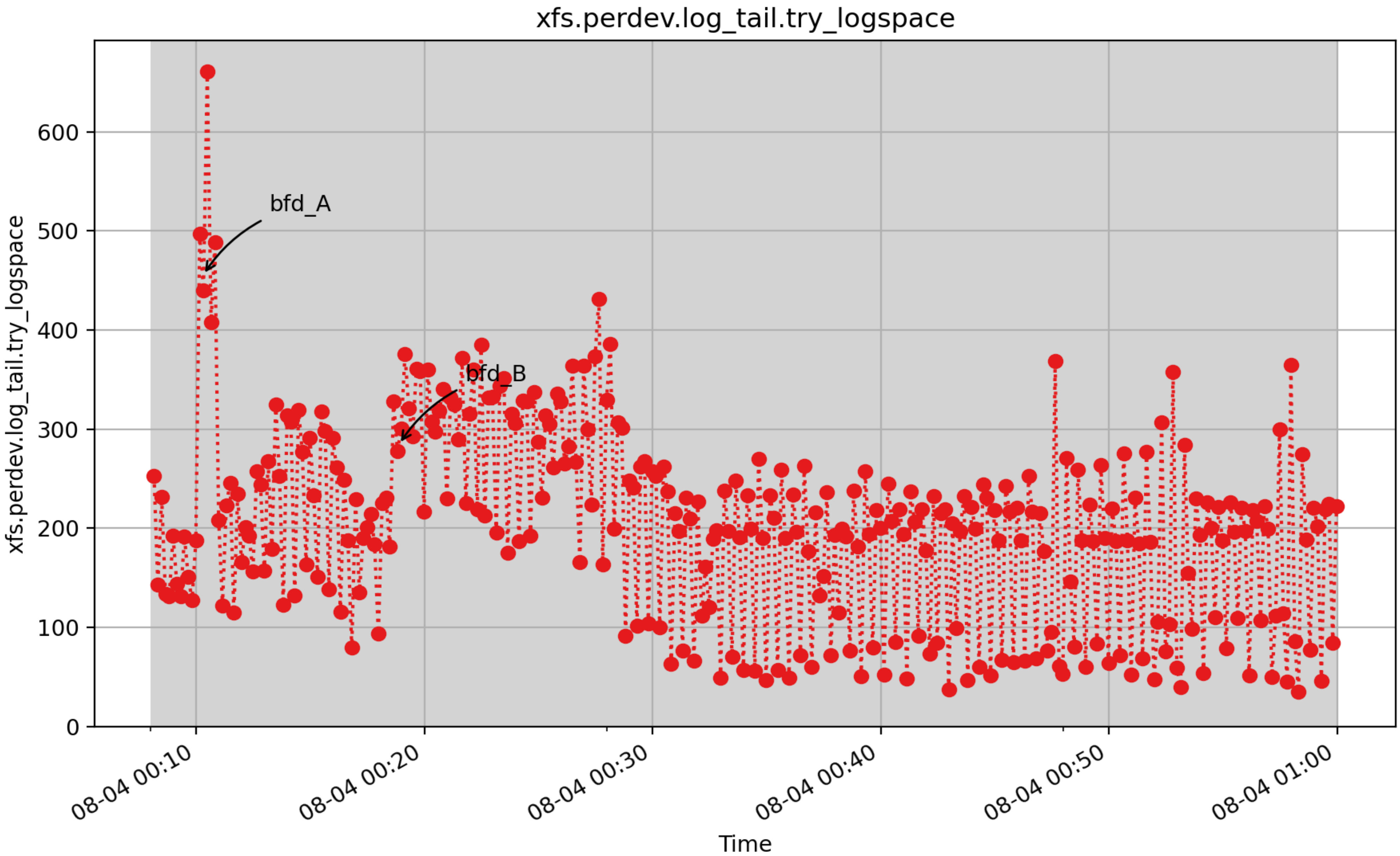
xfs.perdev.log_tail.push_ail.pushes: The number of times the tail of the AIL is moved forward. It is equivalent to the number of successful calls to the function xfs_trans_push_ail(). (count - U32) - *rate converted*



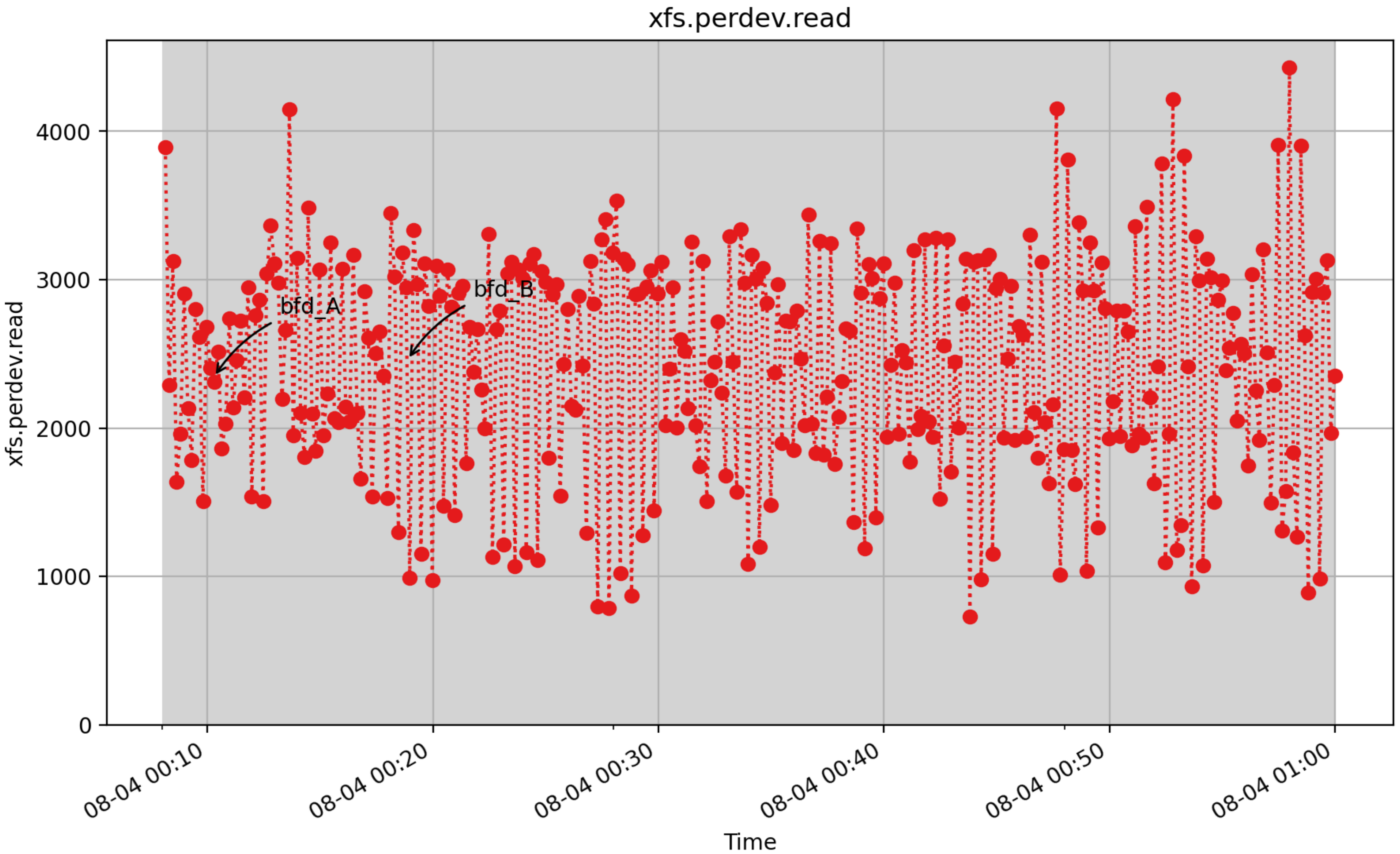
xfs.perdev.log_tail.push_ail.success: value from xs_push_ail_success field of struct xfsstats (count - U32) - *rate converted*



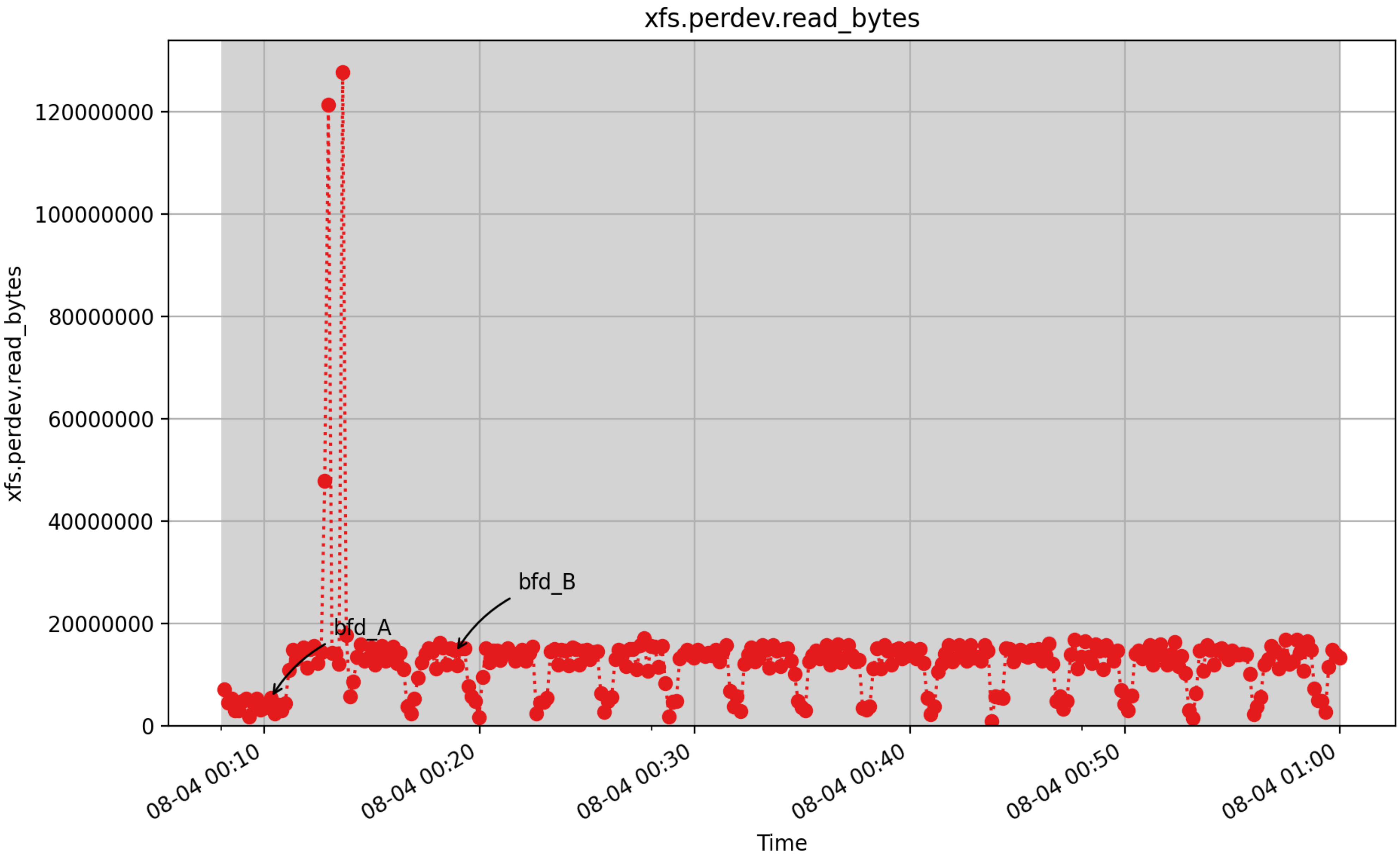
xfs.perdev.log_tail.sleep_logspace: This metric is exported from the xs_sleep_logspace field of struct xfsstats (count - U32) - *rate converted*



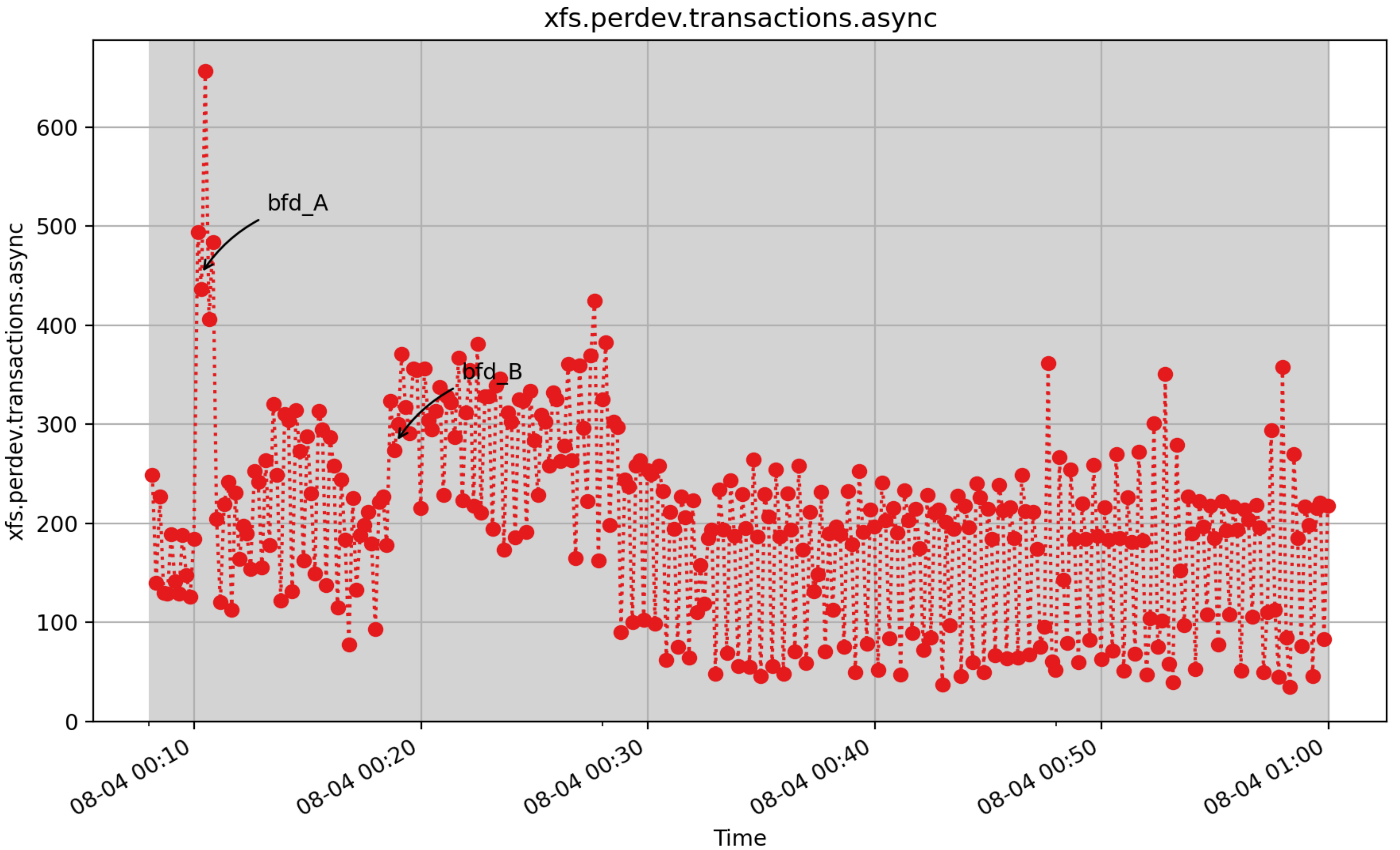
xfs.perdev.log_tail.try_logspace: This metric is exported from the xs_try_logspace field of struct xfsstats (count - U32) - *rate converted*



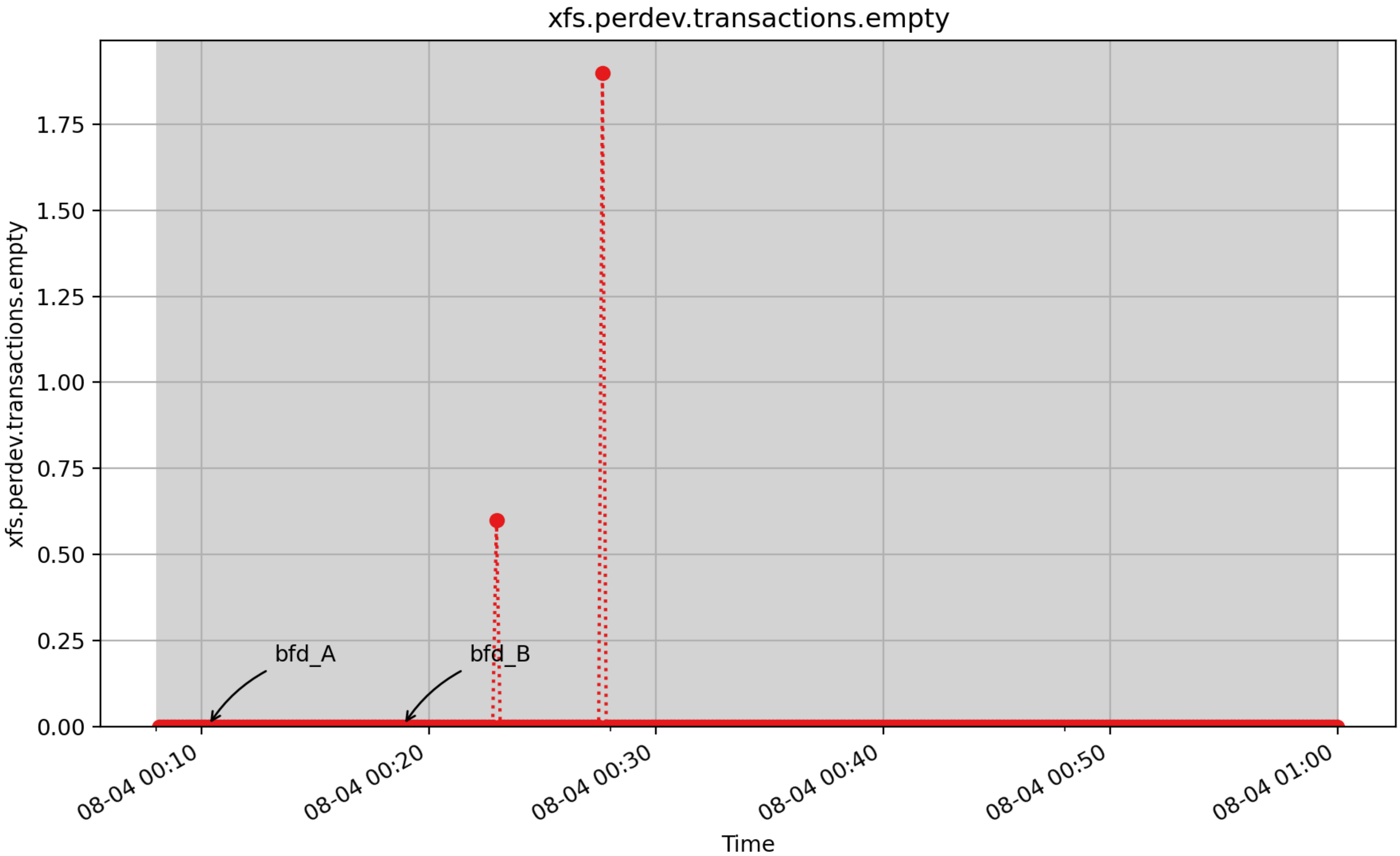
xfs.perdev.read: This is the number of read(2) system calls made to files in XFS file systems. (count - U32) - *rate converted*



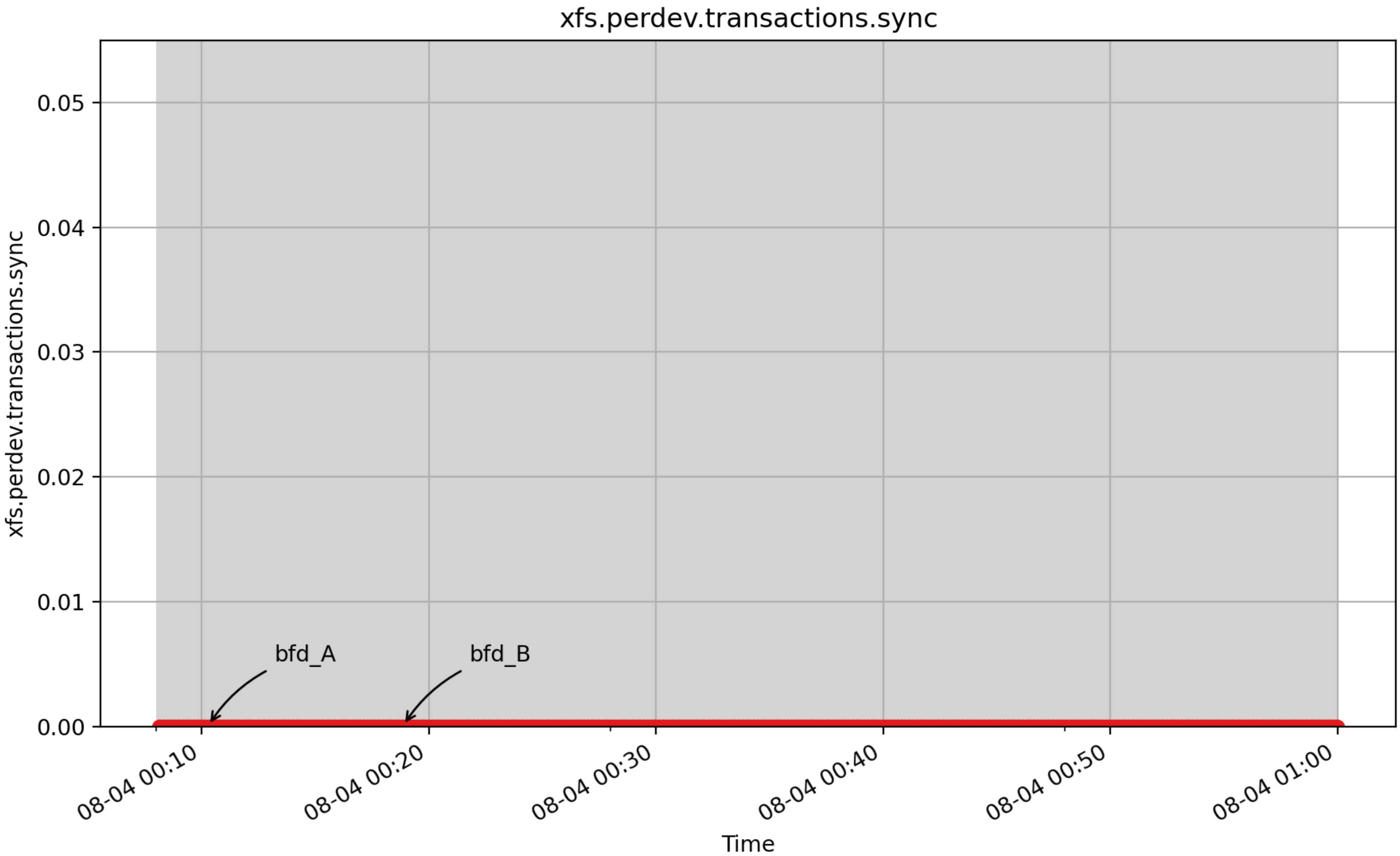
xfs.perdev.read_bytes: This is the number of bytes read via read(2) system calls to files in XFS file systems. It can be used in conjunction with the read_calls count to calculate the average size of the read operations to files in XFS file systems. (byte - U64) - *rate converted*



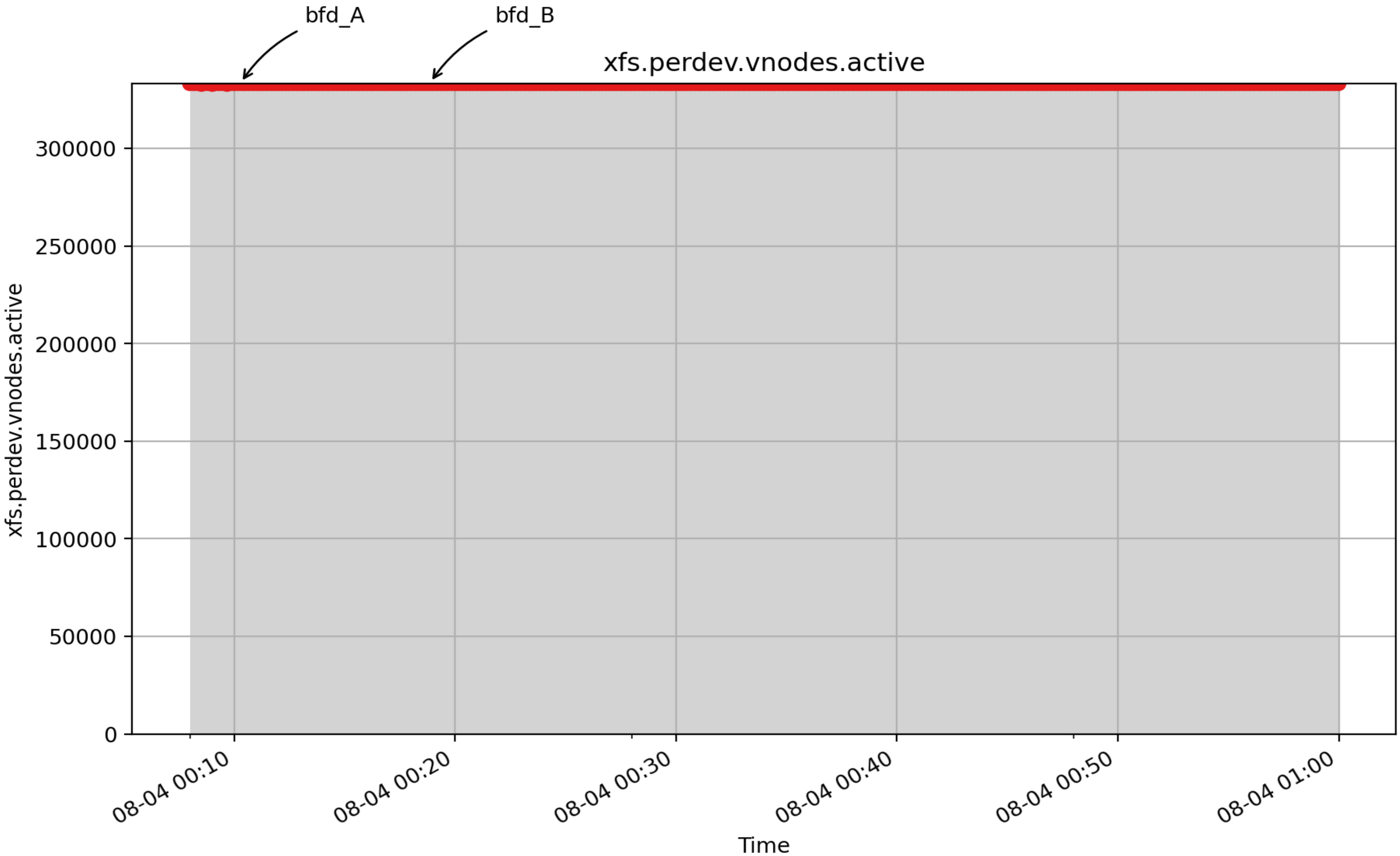
xfs.perdev.transactions.async: This is the number of meta-data transactions which did not wait to be committed to the on-disk log before allowing the process performing the transaction to continue. These transactions are faster and more efficient than synchronous transactions, because they commit their data to the in memory log buffers without forcing those buffers to be written to disk. This allows multiple asynchronous transactions to be committed to disk in a single log buffer write. Most transactions used in XFS file systems are asynchronous. (count - U32) - *rate converted*



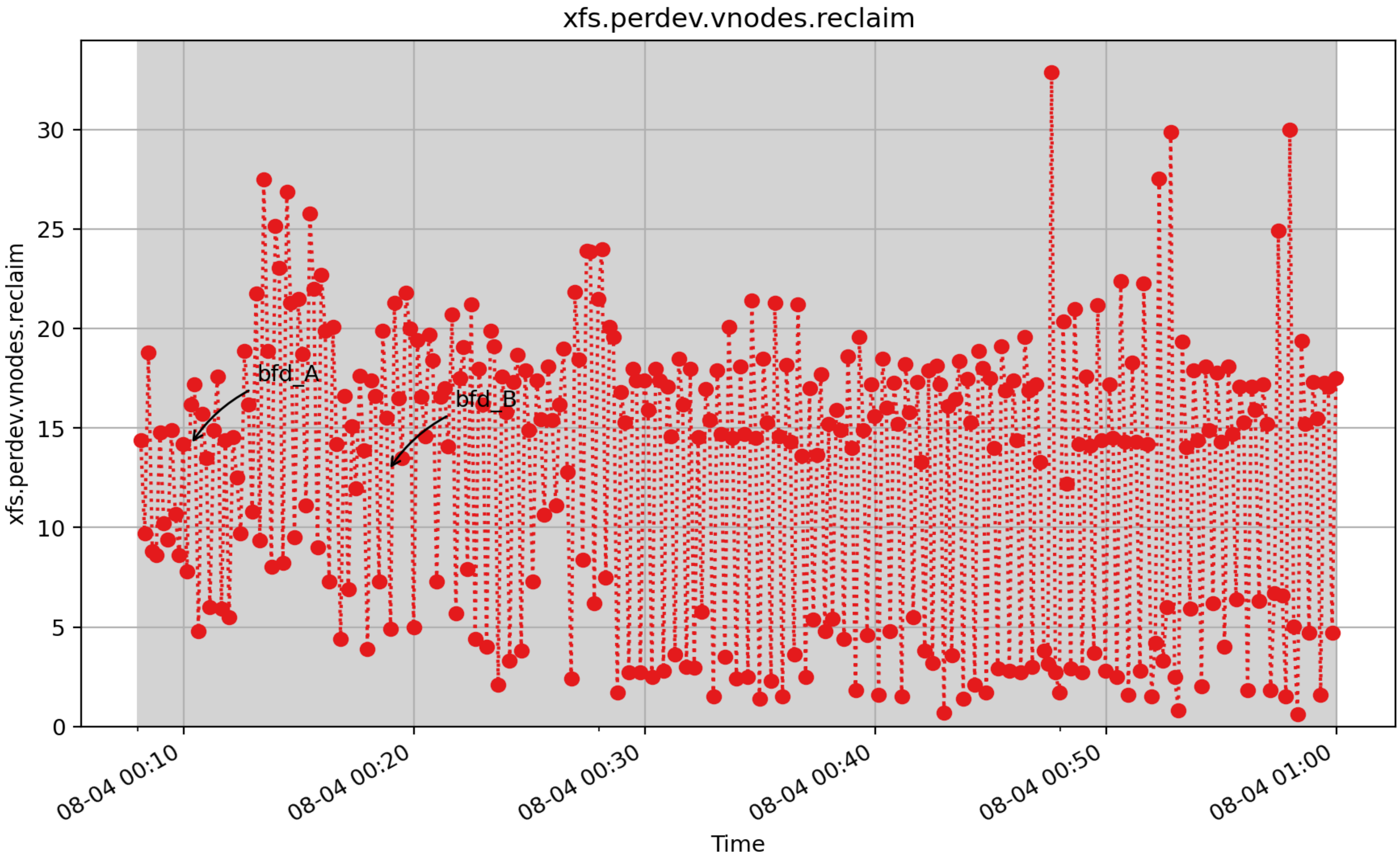
xfs.perdev.transactions.empty: This is the number of meta-data transactions which did not actually change anything. These are transactions which were started for some purpose, but in the end it turned out that no change was necessary. (count - U32) - *rate converted*



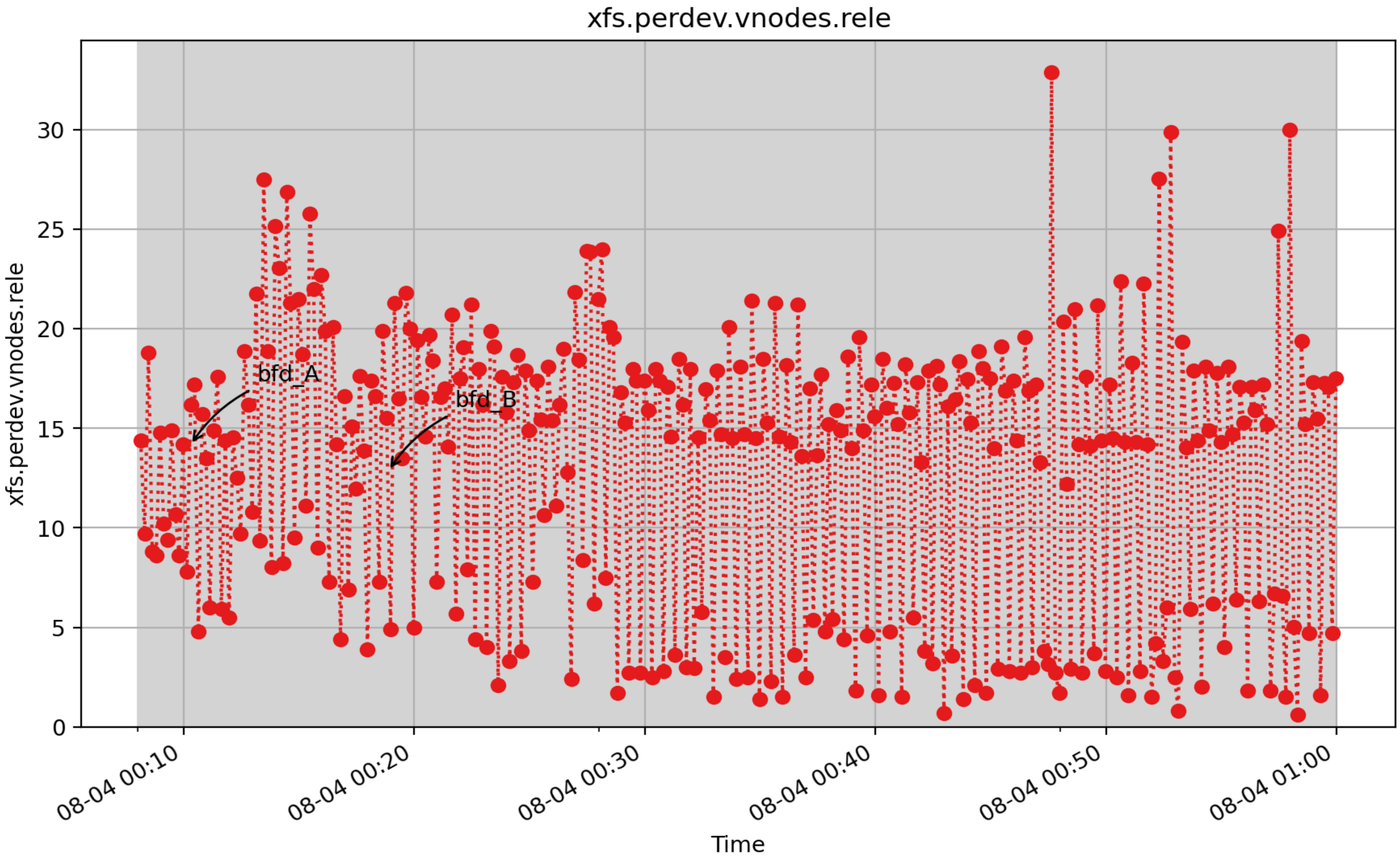
xfs.perdev.transactions.sync: This is the number of meta-data transactions which waited to be committed to the on-disk log before allowing the process performing the transaction to continue. These transactions are slower and more expensive than asynchronous transactions, because they force the in memory log buffers to be forced to disk more often and they wait for the completion of the log buffer writes. (count - U32) - *rate* *converted*



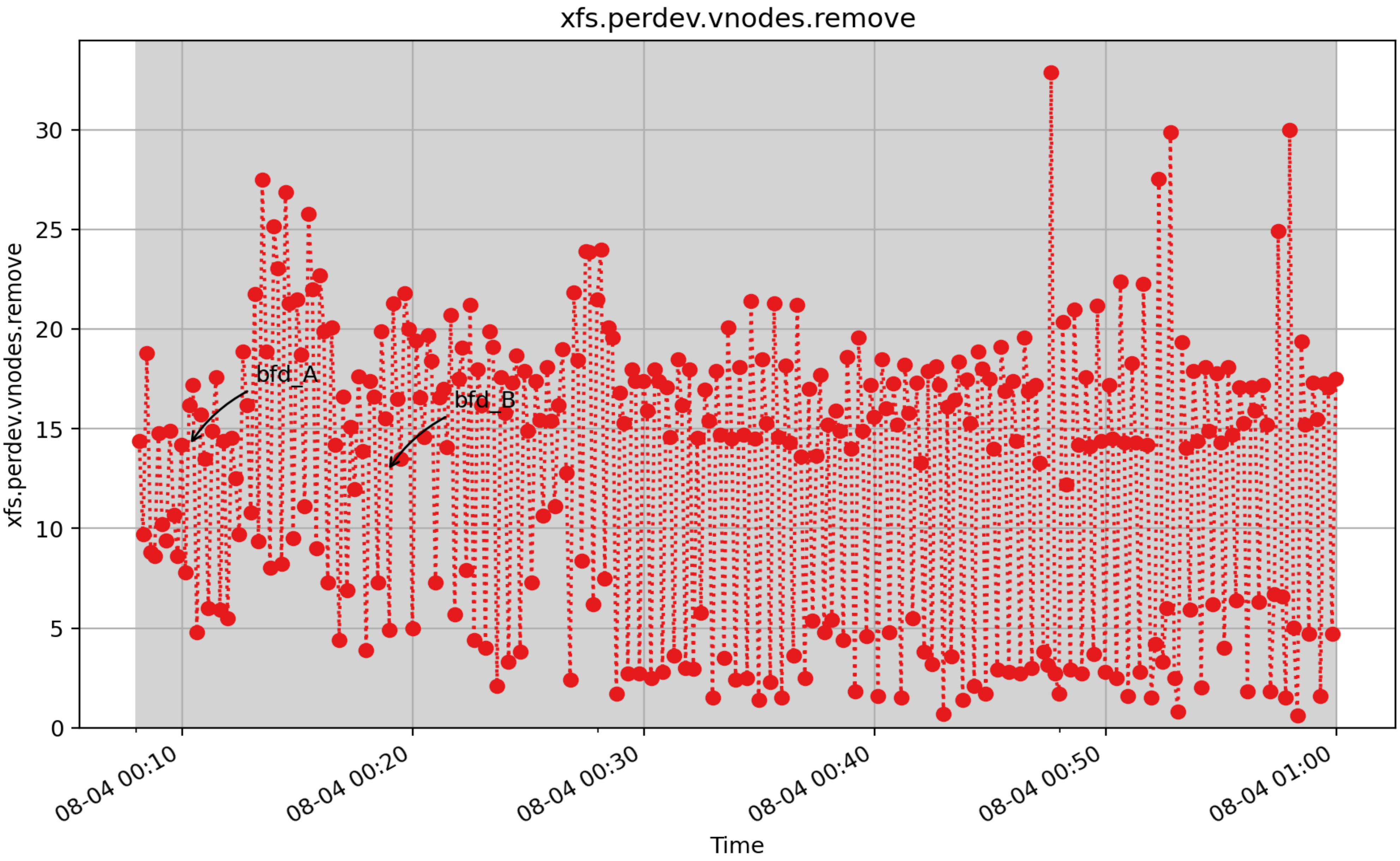
xfs.perdev.vnodes.active: number of vnodes not on free lists (- U32)



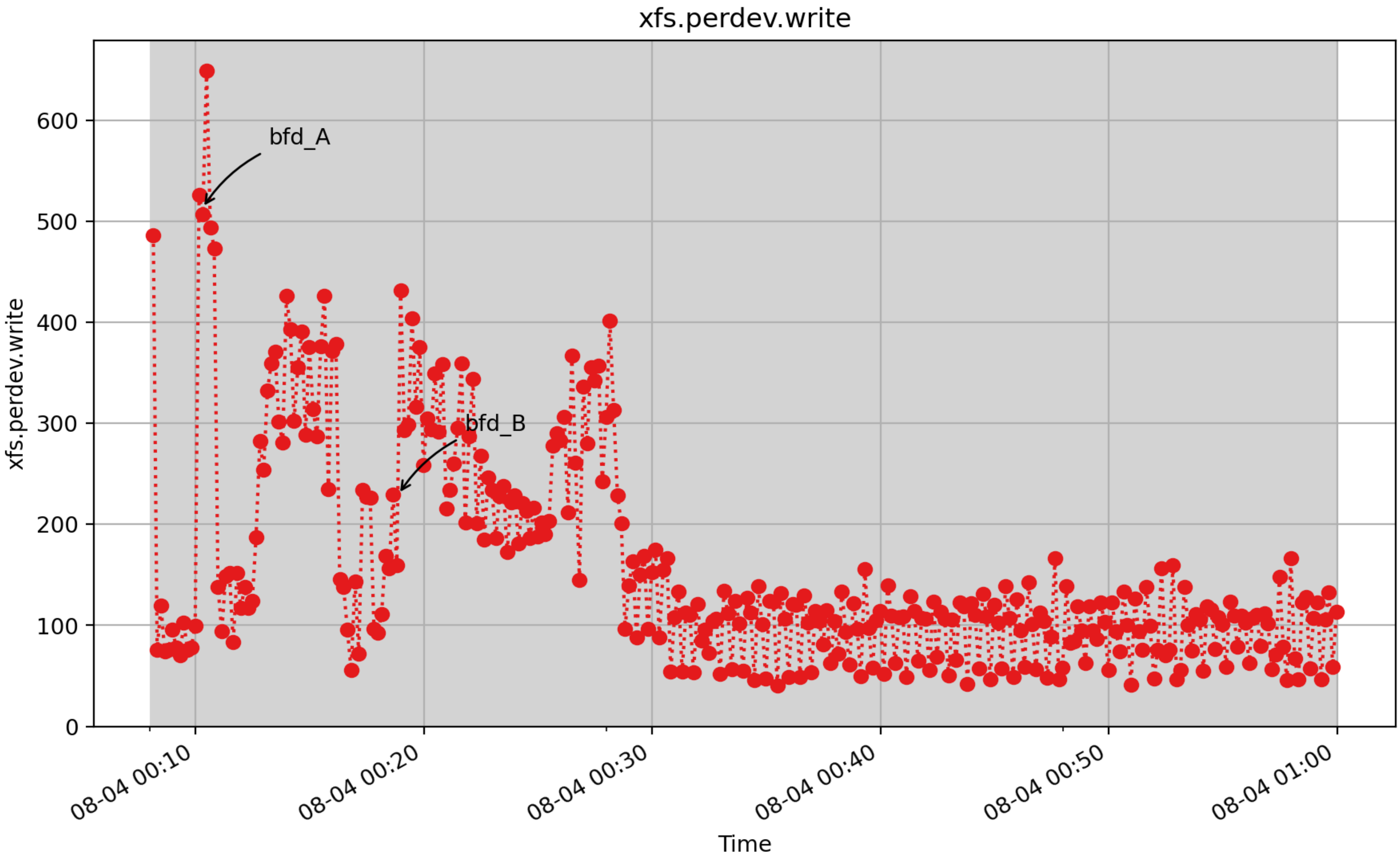
xfs.perdev.vnodes.reclaim: number of times vn_reclaim called (count - U32) - *rate converted*



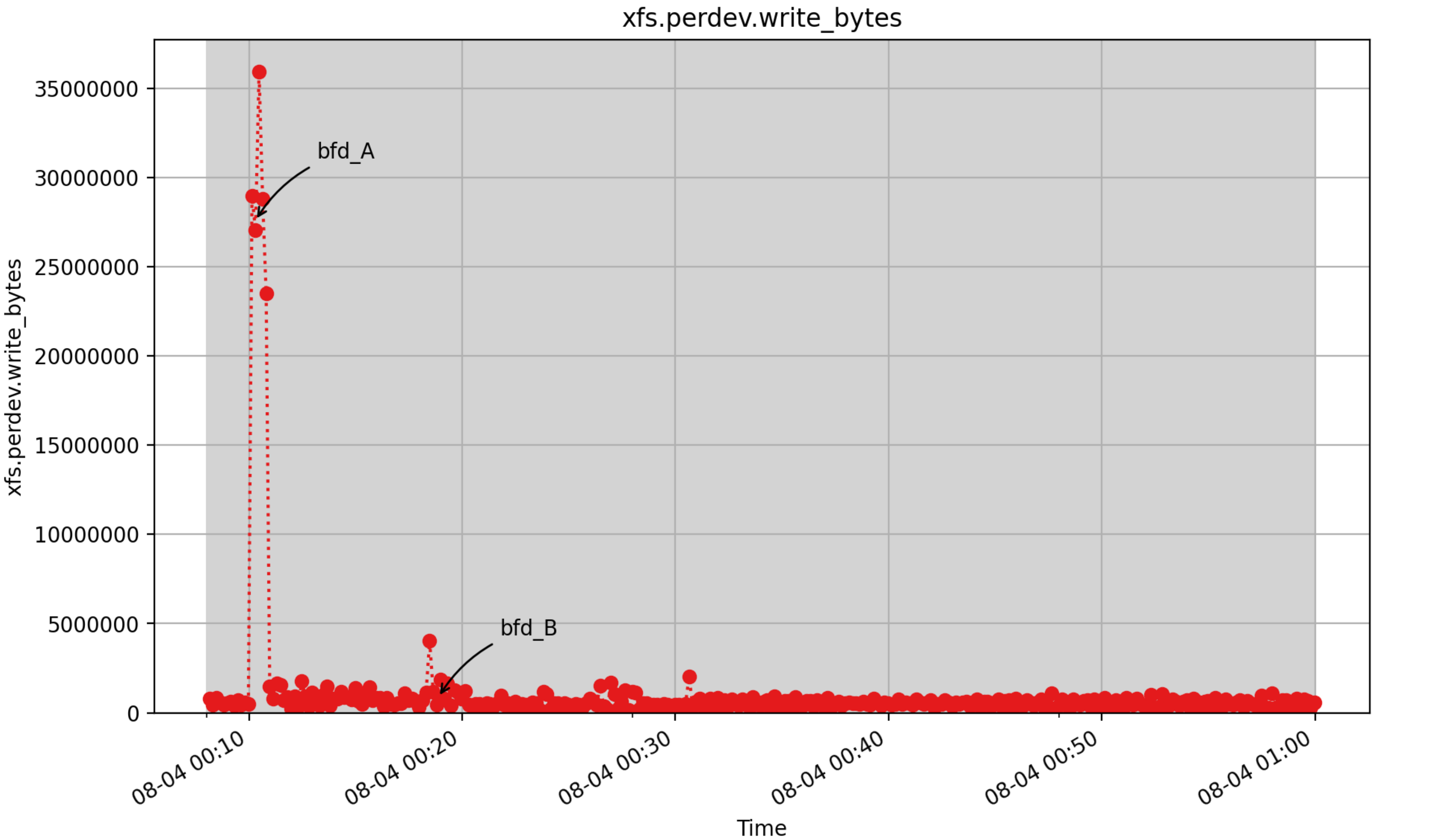
xfs.perdev.vnodes.rele: number of times vn_rele called (count - U32) - *rate converted*



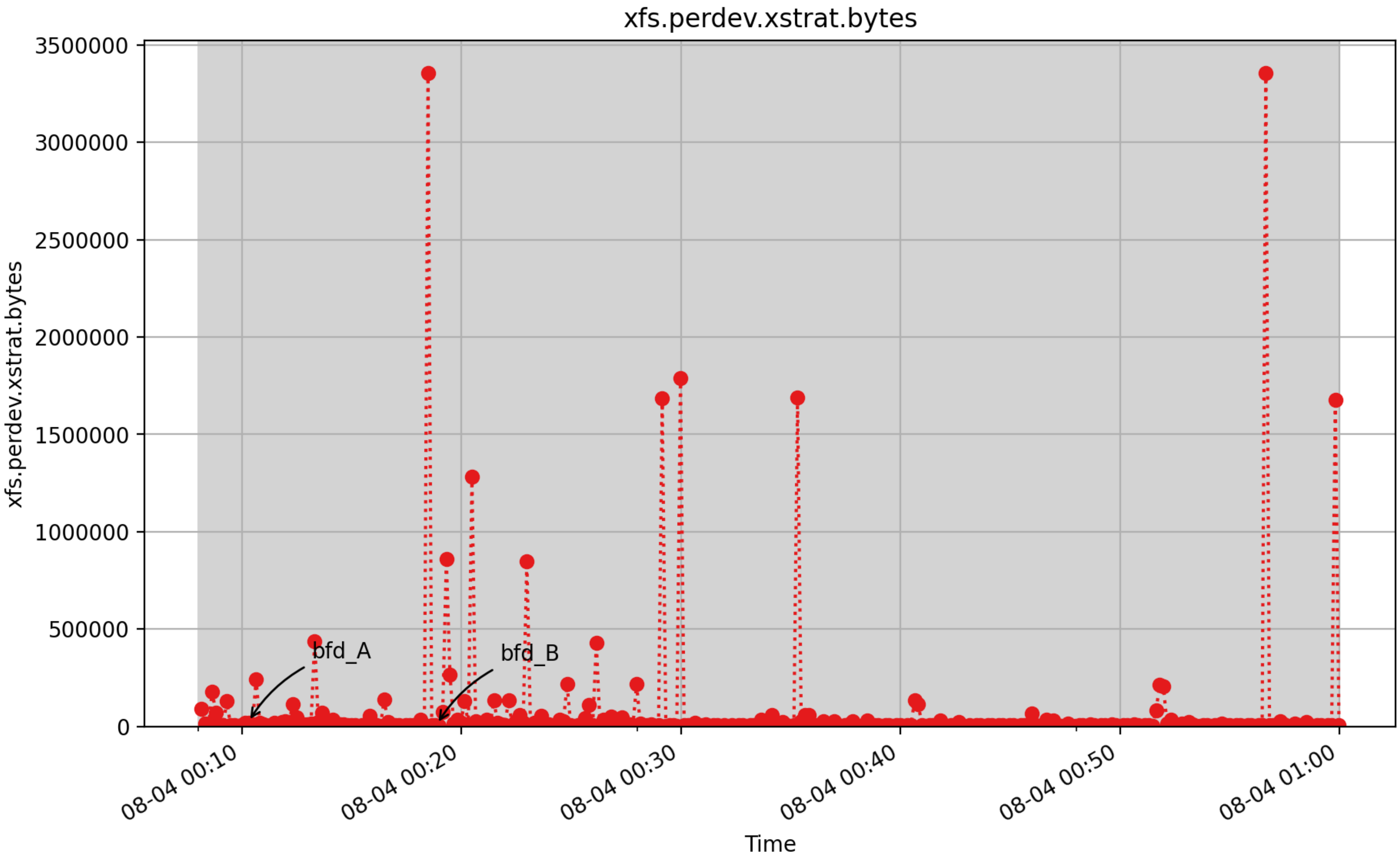
xfs.perdev.vnodes.remove: number of times vn_remove called (count - U32) - *rate converted*



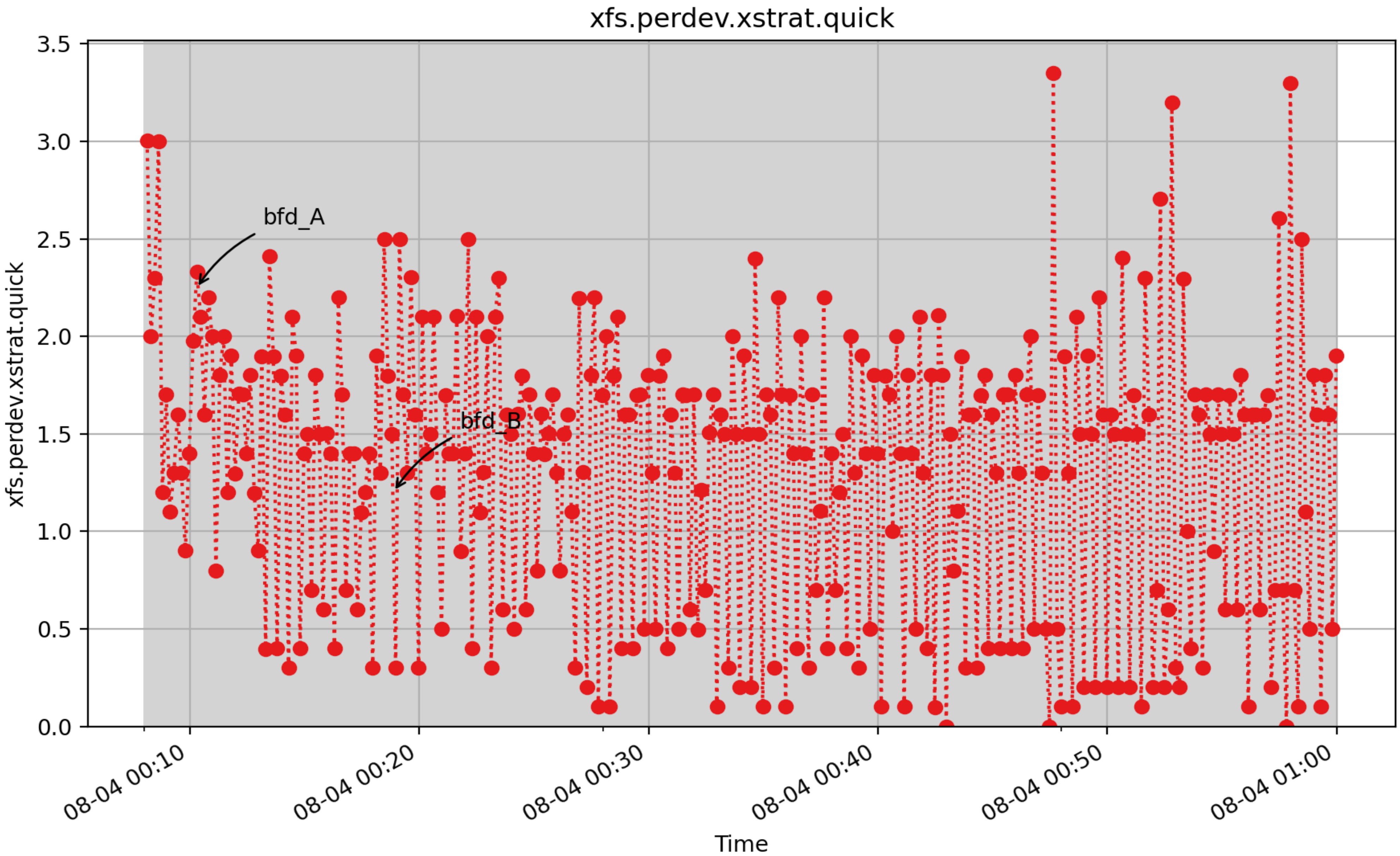
xfs.perdev.write: This is the number of write(2) system calls made to files in XFS file systems. (count - U32) - *rate converted*



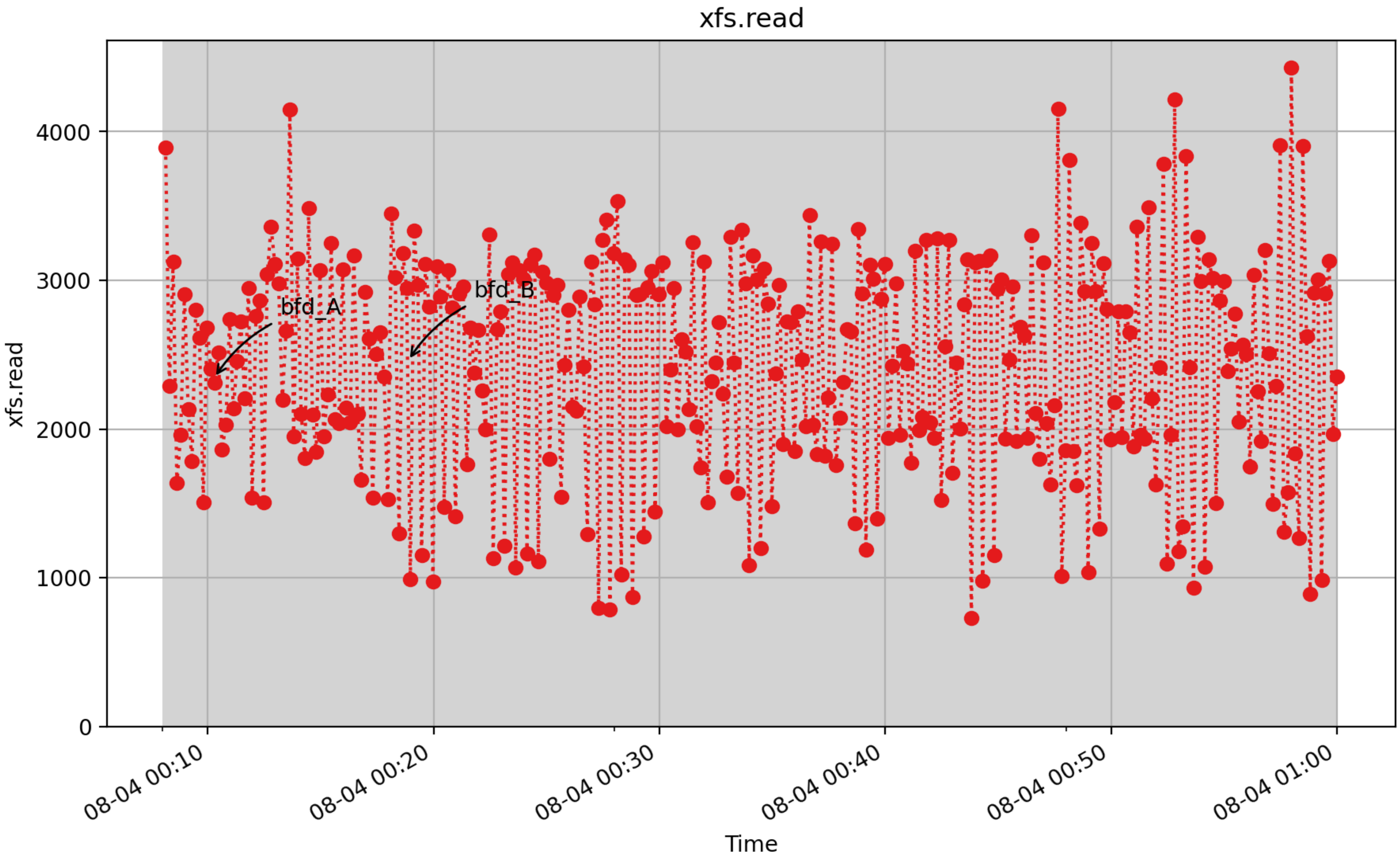
xfs.perdev.write_bytes: This is the number of bytes written via write(2) system calls to files in XFS file systems. It can be used in conjunction with the write_calls count to calculate the average size of the write operations to files in XFS file systems. (byte - U64) - *rate converted*



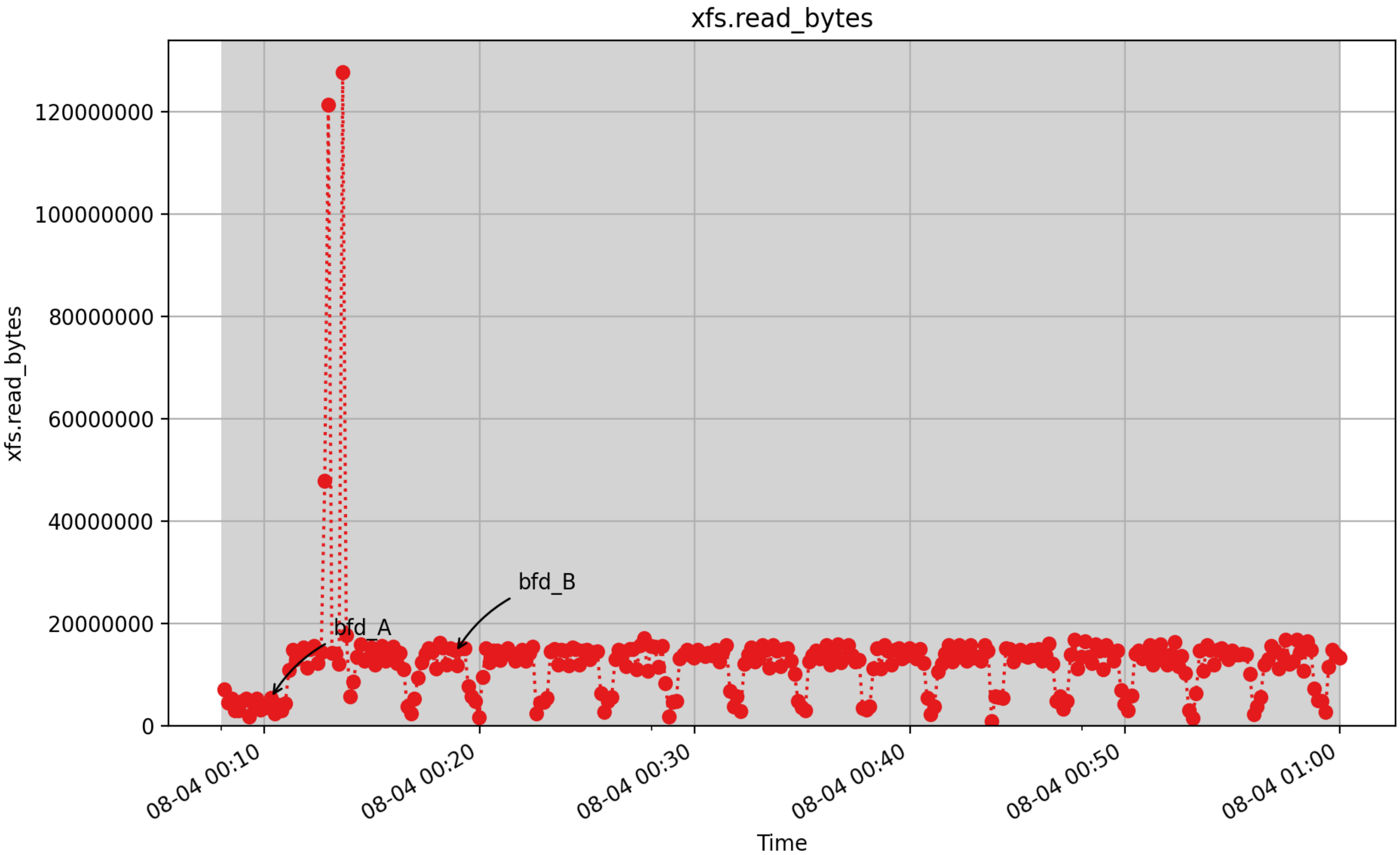
xfs.perdev.xstrat.bytes: This is the number of bytes of file data flushed out by the XFS flushing daemons. (byte - U64) - *rate converted*



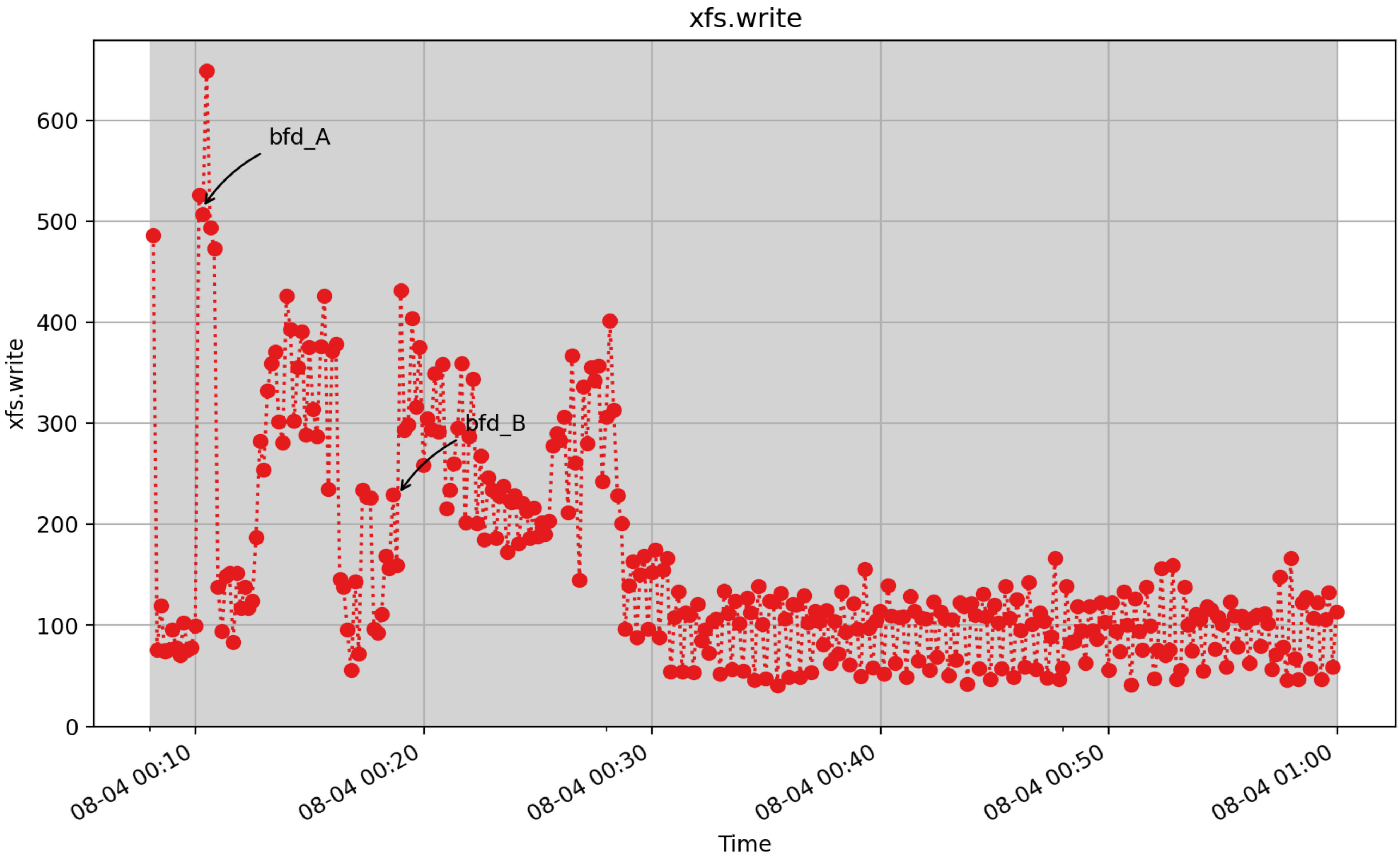
xfs.perdev.xstrat.quick: This is the number of buffers flushed out by the XFS flushing daemons which are written to contiguous space on disk. The buffers handled by the XFS daemons are delayed allocation buffers, so this count gives an indication of the success of the XFS daemons in allocating contiguous disk space for the data being flushed to disk. (count - U32) - *rate converted*



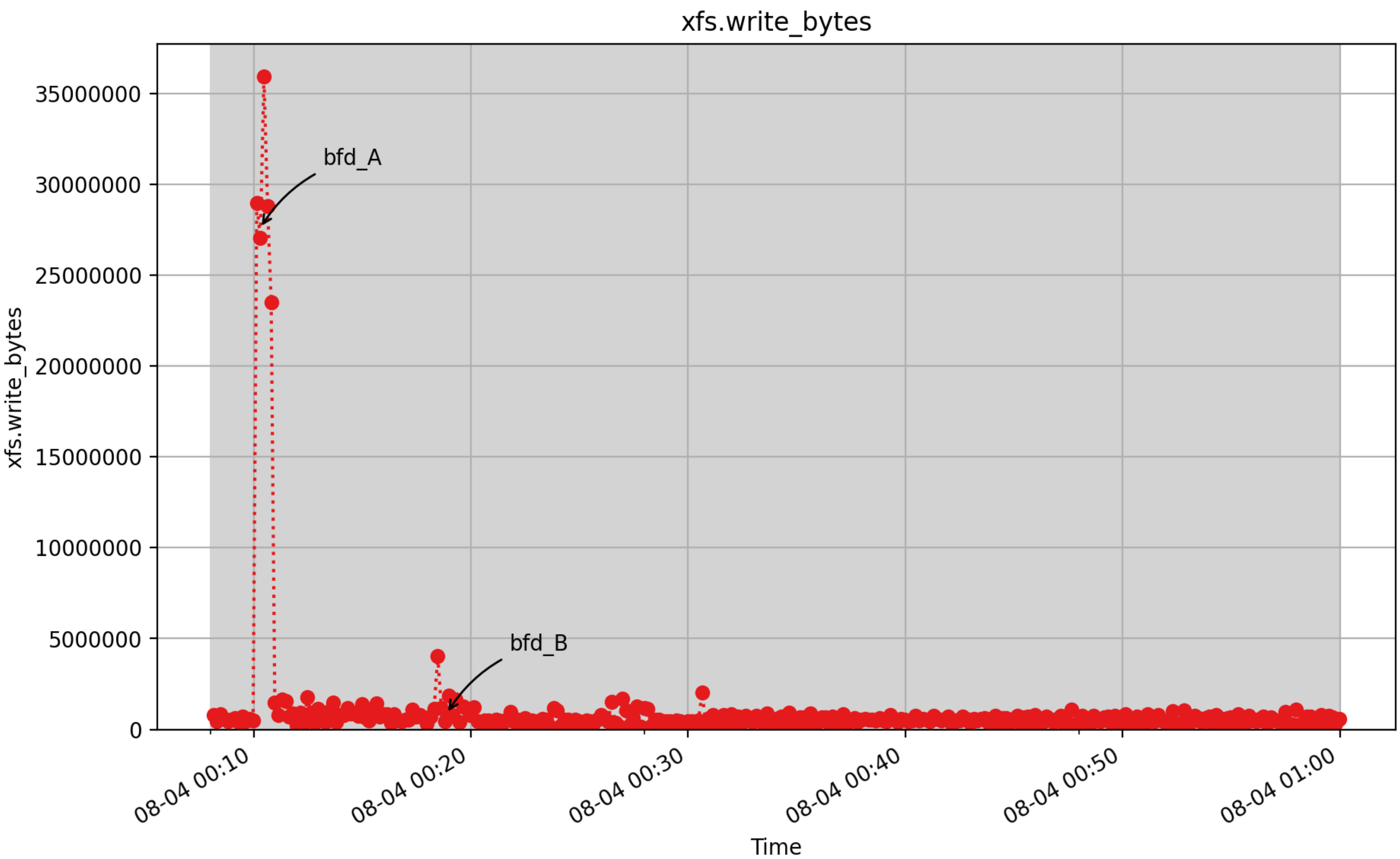
xfs.read: This is the number of read(2) system calls made to files in XFS file systems. (count - U32) - *rate converted*



xfs.read_bytes: This is the number of bytes read via read(2) system calls to files in XFS file systems. It can be used in conjunction with the read_calls count to calculate the average size of the read operations to files in XFS file systems. (byte - U64) - *rate converted*



xfs.write: This is the number of write(2) system calls made to files in XFS file systems. (count - U32) - *rate converted*



xfs.write_bytes: This is the number of bytes written via write(2) system calls to files in XFS file systems. It can be used in conjunction with the write_calls count to calculate the average size of the write operations to files in XFS file systems. (byte - U64) - *rate converted*