Difference Points for commands

Version 1.5

Revision History

Rev.	Date	Prepared by	Updates / Remarks
1.3	28-Feb-2018	NEC	the first version
1.4	24-May-2018	NEC	Changelog: - 'sar and sadf' commands will show VEOS RESTART instead of LINUX RESTART log message. - Added difference point for newly ported "ipcs & ipcrm" commands. - "ve_sysstat" service will restart only for the node who's VEOS is restarted. - Added 'blocked processes' related difference point in 'vmstat' and 'sar' command. -
1.5	20-June-2018	NEC	Changelog: - Updated difference point for psacct-ve service to handle it node wise.

1. Introduction

This document aims at listing down all the differences between ported commands for VE and as-is VH commands.

2. List of difference points in commands

Following are the difference points between ported commands for VE and as-is VH commands:

Package Name	Command Name	Difference Point	Reason
coreutils-arch-ve	uname	In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	There are multiple nodes in VE architecture.
coreutils-arch-ve	arch	In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	There are multiple nodes in VE architecture.
coreutils-ve	nproc	In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	There are multiple nodes in VE architecture.
time-ve	time	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Runs the specified program on given node.	1. There are multiple nodes in VE architecture.

		- VE_NODE_NUMBER is not set: Runs the specified program on any online VE node. 2. The following values will not be applicable for VE, so the values will be zero: - Total number of CPU-seconds that the process spent in kernel mode - Number of major page faults that occurred while the process was running - Number of minor page faults - Number of times the process was swapped out of main memory	2. VE architecture do not support the given fields.
sysstat-ve	pidstat	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval: Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count. 2. The command "/opt/nec/ve/bin/pidstat <interval>" will consider interval value if it is able to fetch the required information in given interval time period. Else it will ignore the interval value. 3. The command "/opt/nec/ve/bin/pidstat <interval>" can sometimes display more than or less than 100% value in “%usr” field	1. There are multiple nodes in VE architecture. 2. Ported command retrieves the information from VEOS via IPC, which takes more time as compared to x86_64 3. In case of VE, the value of “user” will be updated as per the timer interval only (default value is 100 milli-seconds).

		while processes running on all the VE cores	Sometimes a scenario can arrive where, when the command request VEOS to fetch the latest user time but the user time returned can be the value updated on last scheduler timer expiry and vice-versa. Hence, the values retrieved from VEOS can cause some percentage difference for the command.
		4. The command "/opt/nec/ve/bin/pidstat –p SELF" will not display statistics.	4. The SELF keyword indicates that statistics are to be reported for the "pidstat" process itself and pidstat is a VH process not VE process.
		5. The following values will not be applicable for VE, so the values will be zero: - %system : Percentage of CPU used by the task while executing at the system level, kernel - %guest : Percentage of CPU spent by the task in virtual machine (running a virtual processor). - minflt/s : Number of minor faults the task has made per second - minflt-nr : Minor faults made by the task and all its children, and collected during the interval of time. - majflt/s : Number of major faults the task has made per second - majflt-nr : Major faults made by the task and all its children, and collected during the interval of time. - system-ms : Total number of milliseconds spent by the task and all its children while executing at the system level (kernel)	5. VE architecture do not support the given fields.

		- guest-ms : Total number of milliseconds spent by the task and all its children in virtual machine (running a virtual processor) - StkRef: Memory in kilobytes used as stack, referenced by the task.	
sysstat-ve	mpstat	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval: Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count.. 2. "/opt/nec/ve/bin/mpstat -I" will show error message "Interrupts are not applicable for VE". 3. "/opt/nec/ve/bin/mpstat -A" will display only CPUs statistics not interrupts statistics. 4. The command "/opt/nec/ve/bin/mpstat <interval>" can sometimes display more than or less than 100% value in "%usr" field while processes running on all the VE cores.	1. There are multiple nodes in VE architecture. 2. There are no interrupts on VE. 3. There are no interrupts on VE. 4. In case of VE, the value of "user" will be updated as per the timer interval only (default value is 100 milli-seconds). Sometimes a scenario can arrive where, when the command request VEOS to fetch the latest user time but the user time returned can be the value updated on last scheduler timer expiry and vice-versa. Hence, the values

			retrieved from VEOS can cause some percentage difference for the command.
		5. The command "/opt/nec/ve/bin/mpstat -P {cpu [,...] | ON | ALL }" will display information of single VE node (either for the given node or first online node).	5. As per the design of this command, CPU information can be retrieved only for one node and different VE nodes can have different number of CPUs.
		6. The following values will not be applicable for VE, so the values will be zero: - %nice : Percentage of CPU utilization while executing at the user level with nice priority - %sys : Percentage of CPU used by the task while executing at system level, kernel - %iowait : Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request - %steal : Percentage of time spend by a CPU (which is virtualized), for resources from the physical CPU - %irq : Percentage of time spent by the CPU or CPUs to service interrupt - %soft : Percentage of time spent by the CPU or CPUs to service softirqs - %guest : Percentage of CPU spent by the task in virtual machine (running a virtual processor - %gnice : Percentage of time spent by the CPU or CPUs to run a niced guest.	6. VE architecture do not support the given fields.
sysstat-ve	iostat	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the	1. There are multiple nodes in VE architecture.

		information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval: Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count. 2. The command "/opt/nec/ve/bin/iostat can sometimes display more than or less than 100% value in “%user” field while processes running on all the VE cores 3. The following values will not be applicable for VE, so the values will be zero: - %nice : Percentage of CPU utilization while executing at the user level with nice priority - %sys : Percentage of CPU used by the task while executing at system level, kernel - %iowait : Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request - %steal : Percentage of time spend by a CPU (which is	2. In case of VE, the value of “user” will be updated as per the timer interval only (default value is 100 milli-seconds). Sometimes a scenario can arrive where, when the command request VEOS to fetch the latest user time but the user time returned can be the value updated on last scheduler timer expiry and vice-versa. Hence, the values retrieved from VEOS can cause some percentage difference for the command. 3. VE architecture do not support the given fields.

		virtualized), for resources from the physical CPU	
sysstat-ve	sar	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval: Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count...	1. There are multiple nodes in VE architecture.
		2. Command “/opt/nec/ve/bin/sar –d” shows error message “Block devices data is not applicable for VE”	2. There is no device data for VE.
		3. “/opt/nec/ve/bin/sar –n” shows error message “Network statistics is not applicable for VE”.	3. There are no network statistics for VE.
		4. “/opt/nec/ve/bin/sar –I” shows error message “Interrupts are not applicable for VE”	4. There are no interrupts on VE.
		5. “/opt/nec/ve/bin/sar -A” will not display network statistics, interrupts statistics and block devices.	5. There are no network, interrupts and block device statistics for VE.
		6. The command "/opt/nec/ve/bin/sar <interval>" can sometimes display more than or less than 100% value in “%user” field while processes running on all the VE cores.	6. In case of VE, the value of “user” will be updated as per the timer interval only (default value is 100 milli-seconds). Sometimes a scenario can arrive where, when the command request VEOS to fetch the latest user time but the user time returned can be the value updated on last

			scheduler timer expiry and vice-versa. Hence, the values retrieved from VEOS can cause some percentage difference for the command.
		7. When filename is not given, VE specific 'sar' command uses the standard system activity daily data file, "/var/opt/nec/ve/log/sa/sa<dd>_<node_number>", where the dd parameter indicates the current day.	7. There are multiple nodes in VE architecture. So, we maintained separate system activity daily data files for each node.
		8. VE "sar" command will display "VEOS RESTART" instead of "LINUX RESTART" at restart of VEOS.	8. "ve_sysstat" service is restarted at VEOS restart. So, VEOS RESTART message looks more appropriate.
		9. Count of 'blocked processes for i/o' ("blocked" field) in "sar -q" command is unused for VE.	9. VE architecture do not maintain i/o specific blocked processes.
		10. The following values will not be applicable for VE, so the values will be zero: - %nice : Percentage of CPU utilization while executing at the user level with nice priority - %system & %sys : Percentage of CPU used by the task while executing at system level, kernel - %iowait : Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request - %steal : Percentage of time spend by a CPU (which is virtualized), for resources from the physical CPU) - %irq : Percentage of time spent by the CPU or CPUs to service interrupts - %soft : Percentage of time spent by the CPU or CPUs to service softirqs	10. VE architecture do not support the given fields.

		- %guest : Percentage of CPU spent by the task in virtual machine (running a virtual processor) - %gnice : Percentage of time spent by the CPU or CPUs to run a niced guest - pswpin/s : Total number of swap pages the system brought in per second - pswpout/s : Total number of swap pages the system brought out per second - fault/s : Number of page faults (major + minor) made by the system per second - majflt/s : Number of major faults the system has made per second - pgfree/s : Number of pages placed on the free list by the system per second - pgscank/s : Number of pages scanned by the kswapd daemon per second - pgscand/s : Number of pages scanned directly per second - pgsteal/s : Number of pages the system has reclaimed from cache (pagecache and swapcache) per second to satisfy its memory demands - %vmeff : Calculated as pgsteal / pgscan - Kbhugfree : Amount of hugepages memory in kilobytes that is not yet allocated. - %hugused : Percentage of total hugepages memory that has been allocated - bufpg/s : Number of additional memory pages used as buffers by the system per second - campg/s : Number of additional memory pages	

		cached by the system per second - Kbbuffers : Memory used as buffers by the kernel in kilobytes - Kbcached : Memory used to cache data by the kernel in kilobytes - Kbcommit: Memory in kilobytes needed for current workload - %commit : Percentage of memory needed for current workload in relation to the total amount of memory (RAM+swap) - Kbactive : Active memory in kilobytes - Kbinact : Inactive memory in kilobytes - Kbdirty : Memory in kilobytes waiting to get written back to the disk - Kbswpfree : Amount of free swap space in kilobytes. - kbswpused : Amount of used swap space in kilobytes - %swpused : Percentage of used swap space. - Kbswpcad : Amount of cached swap memory in kilobytes - %swpcad : Percentage of cached swap memory in relation to the amount of used swap space - Dentunusd : Number of unused cache entries in the directory cache.	
sysstat-ve	sadc	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command collects the information corresponding to given node. - VE_NODE_NUMBER is not set and given interval is “1”: Command collects the	1. There are multiple nodes in VE architecture. - The command “sadc” is internally called by “sa1” which is designed

		information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval (greater than “1”): Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count. 2. "/opt/nec/ve/lib64/sa/sadc –S" does not support options INT, DISK, SNMP, IPV6, XDISK. It will show the following error messages: a) “/opt/nec/ve/lib64/sa/sadc –S INT”: Interrupts are not applicable for VE b) “/opt/nec/ve/lib64/sa/sadc –S DISK”: Block devices data is not applicable for VE c) “/opt/nec/ve/lib64/sa/sadc –S SNMP”: SNMP statistics are not applicable for VE d) “/opt/nec/ve/lib64/sa/sadc –S IPV6”: IPV6 statistics are not applicable for VE e) “/opt/nec/ve/lib64/sa/sadc –S XDISK”: Partition and disk statistics is not applicable for VE 3. "/opt/nec/ve/lib64/sa/sadc” command collects the information in file "sa<dd>_<node_number>" at path "/var/opt/nec/ve/log/sa". If outfile (a file to collect information) is set to “-” then sadc uses the standard system activity daily data file which is	to be started automatically by the cron command and collect system activity daily data at 1 sec of interval. So, it calls sadc with interval “1” and we need to collect system activity data for all online nodes. 2. Only power management specific data can be collected in VE, so, "sadc -S" option supports only POWER, ALL and XALL options. The other options will display error message. 3. There are multiple nodes in VE architecture. Hence we need to create files according to VE nodes to collect system activity data.

		“var/opt/nec/ve/log/sa/ sa<dd>_<node_number>” for VE. But the standard (x86_64) ‘sadc’ command collects system activity information in file "sa<dd>" at path "/var/log/sa", (dd parameter indicates the current day).	
sysstat-ve	sadf	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval: Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count.. 2. “/opt/nec/ve/bin/sadf” can sometimes display more than or less than 100% value in “%user” field while processes running on all the VE cores. 3. VE specific “/opt/nec/ve/bin/sadf” command uses file “/var/opt/nec/ve/log/sa/sa<dd>_<nod	1. There are multiple nodes in VE architecture. 2. sadf command reads the data collected by sadc. In case of VE, the value of “user” will be updated as per the timer interval only (default value is 100 milli-seconds). Sometimes a scenario can arrive where, when the command request VEOS to fetch the latest user time but the user time returned can be the value updated on last scheduler timer expiry and vice-versa. Hence, the values retrieved from VEOS can cause some percentage difference for the command. 3. There are multiple nodes in VE architecture. So, we maintained separate system

		e_number>" to display system activity data. The same file will be used, if outfile (file to extract data and write to standard output) is omitted. But the standard (x86_64) 'sadf' command uses file "var/log/sa/sa<dd>", (dd parameter indicates the current day). 4. VE "sadf" command will display "VEOS-RESTART" instead of "LINUX-RESTART" at restart of VEOS.	activity daily data files for each node. 4. "ve_sysstat" service is restarted at VEOS restart. So, VEOS-RESTART message looks more appropriate.
sysstat-ve	sa1	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command collects the information corresponding to given node. - VE_NODE_NUMBER is not set and given interval is "1": Command collects the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with interval (greater than "1"): Command shows error message because VE_NODE_NUMBER is mandatory to execute command with interval/count. 2. VE specific "/opt/nec/ve/lib64/sa/sa1" command which is started automatically by the cron command will collect the system activity daily data in file	1. There are multiple nodes in VE architecture. - The command "sa1", which internally calls "sadc", is designed to be started automatically by the cron job and collect system activity daily data at 1 sec of interval. So, it calls sadc with interval "1" and we need to collect system activity data for all online nodes. 2. There are multiple nodes in VE architecture. So, we maintained separate system activity daily data files for each node.

		“/var/opt/nec/ve/log/sa/sa<dd>_<node_number>” for all online VE nodes. But the standard (x86_64) ‘sa1’ command collects system activity information in file “/var/log/sa sa<dd>”, (dd parameter indicates the current day).	
sysstat-ve	sa2	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command writes a daily report corresponding to given node. - VE_NODE_NUMBER is not set: Command writes a daily report corresponding to all online VE nodes. 2. VE specific “/opt/nec/ve/lib64/sa/sa2” command which get started automatically by the cron command will collect the system activity daily data in file “/var/opt/nec/ve/log/sa/sar<dd>_<node_number>" for all online VE nodes. But the standard (x86_64) ‘sa2’ command collects system activity information in file "/var/log/sa sar<dd>", (dd parameter indicates the current day).	1. There are multiple nodes in VE architecture. 2. There are multiple nodes in VE architecture. So, we maintained separate system activity daily data files for each node.
sysstat-ve	sysstat services	The following services related files will be different for ported sysstat package: - Sysstat service for collecting system activity information of VEs are defined in “/usr/lib/systemd/system/ve_sysstat@.service”. The standard sysstat package defines this information in “/usr/lib/systemd/system/sysstat.service”. Hence VE specific service will be started by command “systemctl start ve_sysstat@N.service”. (Where N indicates VE node number)	There are multiple nodes in VE architecture. VE specific service will collect the system activity information for specified/all VE nodes. Hence, we need to maintain separate VE specific services and configuration files.

		- In case of VE, whenever any/all VEOS is restarted then ve_sysstat@ service of corresponding node/s will be restarted. - Sysstat service for collecting system activity information automatically by cron command of VEs are defined in "/etc/cron.d/ve_sysstat". The standard sysstat package defines this information in "/etc/cron.d/sysstat". - The configuration file which includes definition of multiple macros will be defined in "/etc/sysconfig/ve_sysstat". The standard sysstat package defines this information in "/etc/sysconfig/sysstat".	
util-linux-ve	taskset	In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Runs the specified process or search the given PID on given node. - VE_NODE_NUMBER is not set: Runs the specified program on any online VE node or search the given PID on all online nodes.	There are multiple nodes in VE architecture.
util-linux-ve	lscpu	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	1. There are multiple nodes in VE architecture.

		2. Command "/opt/nec/ve/bin/lscpu -c -e" will show error message “No offline CPU exists for VE”. 3. Command "/opt/nec/ve/bin/lscpu” with ‘-s or –sysroot’ option will show error message “-s or --sysroot option is not supported for VE”.	2. In case of VE, we can't make a CPU offline. 3. In case of VE, user can not specify the directory to gather CPU data.
util-linux-ve	prlimit	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Runs the specified process or search the given PID on given node. - VE_NODE_NUMBER is not set: Runs the specified program on any online VE node or search the given PID on all online nodes. 2. In case of VE, PID is mandatory to run “/opt/nec/ve/bin/prlimit” and “/opt/nec/ve/bin/prlimit [RESOURCE OPTIONS] to show the resource limit otherwise it will give error. But in case of x86_64, it shows the resource limits of the current process 3. In case of VE, command “/opt/nec/ve/bin/prlimit> -p <pid>” shows the resource limit for NICE and RTPRIO as blank (-). 4. Get/set the limits for “nice” and “rtprio” are not supported for VE. So the following commands are not supported either with or without PID and shows the error “Resource not supported”: a) /opt/nec/ve/bin/prlimit -e=<limits> b) /opt/nec/ve/bin/prlimit --nice=<limits> -p <pid> c) /opt/nec/ve/bin/prlimit -r=<limits>	1. There are multiple nodes in VE architecture. 2. In case of Linux, it shows the resource limits of the current process, i.e. running instance of ‘prlimit’ command which is a VH process and not a VE process”. So in case of VE, prlimit cannot be run without specifying PID 3. Priority scheduling is not supported in VE, hence, getpriority()/setpriority() system calls are not supported. 4. Priority scheduling is not supported in VE, hence, getpriority()/setpriority() system calls are not supported.

		d) /opt/nec/ve/bin/prlimit --rtprio=<limits> -p <pid> 5. Get/set the limits for “memlock” will successfully set the given limit but will not affect the VE process memory. 6. We cannot run a VE process using command “/opt/nec/ve/bin/prlimit <VE_process>” with given stack limit specified by the environment variable VE_STACK_LIMIT.	5. There is no swapped memory in VE, so the complete memory in locked. 6. In this case, VEOS will get the resource limits given by prlimit command and the command cannot parse the binary and do not have stack information to calculate stack limit.
util-linux-ve	lslocks	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	1. There are multiple nodes in VE architecture.
util-linux-ve	ipcs	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. 2. VE command “ipcs” and “ipcs -a/--all” option only displays shared memory information. 3. The following options are not supported for VE: • -q/--queue • -s/--semaphores 4. The shared memory limit (say 4096) displayed using command “/opt/nec/ve/bin/ipcs -l” includes the	1. There are multiple nodes in VE architecture. 2. There are no VE specific message queues and semaphores. 3. There are no VE specific message queues and semaphores. 4. VH and VE both shares the same shared memory limit.

		shared memories in VH and all VE nodes and there are no impact of VE_NODE_NUMBER to display shared memory limit. 5. When VH shared memory limit is exhausted and VE “ipcs” command is executed then it will return error “No space left on device”. 6. The number of shared memory segment which a user can create through VE process will be less than "max number of segments". 7. The following value will not be applicable for VE, so the values will be zero: pages swapped: Total number of swapped shared memory pages	5. VH command reads the information from “/proc” file system but VEOS creates a shared memory on VH to write all VE specific shared memory information and the command “ipcs” reads this information to display the output. So when shared memory limit is exhausted and VEOS failed to create a shared memory then command will return error. To overcome from this error, user need to remove some shared memories using VH ‘ipcrm’ command. 6. “ve_exec” itself consumes a shared memory and uses the memory until its termination. So let’s say, a VE process create 4095 shared memory than it will totally create 4096 segments on VH. One of them will be created by “ve_exec”. 7. VE architecture do not support the given fields.
util-linux-ve	ipcrm	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	1. There are multiple nodes in VE architecture.

		2. The following options are not supported for VE: • -Q/--queue-key <msgkey> • -q/--queue-id <msgid> • -S/--semaphore-key <semkey> • -s/--semaphore-id <semid> • --all=[msg sem] 3. VE command “ipcrm” and “ipcrm –a” will remove only shared memory. 4. When VH shared memory limit is exhausted and VE “ipcrm -a” or “ipcrm –all=shm” command is executed then it will return error “No space left on device”.	2. There are no VE specific message queues and semaphores. 3. There are no VE specific message queues and semaphores. 4. VEOS creates a shared memory on VH to write information about removed shared memories when command is executed with ‘–a’ or ‘–all=shm’ option. So when shared memory limit is exhausted and VEOS failed to create a shared memory then command will return error. You can perform following steps to overcome from this error: i. Display all shared memory by VH’s ‘ipcs’ command ii. Delete some shared memories by VH's ‘ipcrm’ command. Now, execute /opt/nec/ve/bin/ipcrm –a
psacct-ve	sa	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	1. There are multiple nodes in VE architecture.

		2. VE_NODE_NUMBER should be specified to execute commands: a) /opt/nec/ve/sbin/sa <filename> b) /opt/nec/ve/sbin/sa --other-acct-file <filename>	2. These commands will use 'usracct and savacct' files (if these files exists). To pick the correct 'usracct and savacct' file in multiple VE node environment, we need to specify node number. Otherwise it will use any random VE specific file. Hence, will display wrong information.
		3. VE_NODE_NUMBER should be specified to execute commands: a) /opt/nec/ve/sbin/sa --other-usracct-file <filename> -s b) /opt/nec/ve/sbin/sa --other-savacct-file <filename> -s	3. These commands will use pacct file to generate usracct/savacct file. To pick the correct 'pacct' file in multiple node environment, we need to specify node number. Otherwise it will use any random VE specific 'pacct' file. Hence, will display wrong information.
		4. "/opt/nec/ve/sbin/sa" command with - -ahz option will not have any effect on the values printed in STDOUT.	4. AHZ value is used in calculation of time related options. In case of VE, this value is not used, as the time is received in seconds/ microseconds from VEOS.
		5. VE specific ported 'sa' command collects the information in file usracct_<node_number> and savacct_<node_number>" at path "/var/opt/nec/ve/account". But in case of x86_64, 'sa' command collects the process accounting information in file "usracct and savacct" at path "/var/log/sa".	5. There are multiple nodes in VE architecture. Hence we need to create accounting files according to VE nodes.
		6. The following values will not be applicable for VE, so the values will be zero: - min & min/c : Number of minor page faults - maj & maj/c : Number of major page faults - swp & swp/c: Number of swap pages	6. VE architecture do not support the given fields.

		- system time of process - Disk I/O operations (io)	
psacct-ve	accton	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command enables/disable accounting corresponding to given node. - VE_NODE_NUMBER is not set: Command enables/disable accounting corresponding to all online VE nodes. 2. "/opt/nec/ve/sbin/accton on" will not show any error if "/var/opt/nec/ve/account/pacct_<N>" file does not exist. But in case of x86_64, if "/var/account/pacct" file is not present, <accton on> will show error. 3. VE specific ported accton command collects the information in file "pacct_<node_number> at path "/var/opt/nec/ve/account". But in case of x86_64, accton command collects the process accounting information in file "pacct" at path "/var/log/sa". 4. In case of VE, if the required file doesn't exist while running command "/opt/nec/ve/sbin/accton <filename/on>" then it will display error "No such file and directory". But in case of x86_64, command displays the error "permission denied".	1. There are multiple nodes in VE architecture. 2. In case of VH, the acct file is created at the time of package installation, but in case of VE the file is created when 'accton on' command is executed, because the acct files are created per online node and at the time of installation of package, the number of nodes which are online are not known 3. There are multiple nodes in VE architecture. Hence we need to create files according to VE nodes to enable accounting. 4. The design of VE and VH command is different. The VE specific command will check the file existence before checking permission and VH command will check for permission before checking the file existence.
psacct-ve	dump-acct	N/A	N/A
psacct-ve	Psacct Services	The following points related to services will be different for ported psacct-ve package: - Psacct service for collecting process accounting	There are multiple nodes in VE architecture. So, separate services are required to handle process accounting for specified/all VE nodes.

		information for VE is defined in "/usr/lib/systemd/system/psacct-ve@.service". The standard psacct package defines this information in "/usr/lib/systemd/system/psacct.service". - VE 'psacct-ve' service gets started by command "systemctl start psacct-ve@$N.service" and stopped by command "systemctl stop psacct-ve@$N.service" (Where $N specifies VE node number). - VE psacct-ve service enable/disable accounting for all VE nodes. It is not possible to enable/disable psacct-ve service for specified VE node. So, if psacct-ve service is enabled by command "systemctl enable psacct-ve@$N.service" then it will enable psacct service for all VE nodes (Where $N specifies VE node number). - To disable psacct-ve service, we need to disable all services which was enabled earlier. Let's say, psacct-ve service gets enabled by command: "systemctl enable psacct-ve@$N.service" To disable this service, we need to use following command: "systemctl disable psacct-ve@$N.service" (Where $N specifies VE node number).	- There are multiple nodes in VE architecture. So, VE psacct-ve@$N.service used to start and stop VE process's accounting for specified/all VE nodes. - When service gets enabled then it creates link with psacct-ve@.service at path "/etc/systemd/system/multi-user.target.wants/" which will enable accounting for all VE nodes. - When VE psacct-ve service needs to be disabled then all the links exist at path "/etc/systemd/system/multi-user.target.wants/" with "psacct-ve@.service" should be removed.

		- The logrotate file which is used by psacct-ve package will be defined in "/etc/logrotate.d/psacct-ve". The standard psacct package defines this information in "/etc/logrotate.d/psacct".	
strace-ve	strace	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Runs the specified process or search the given PID on given node. - VE_NODE_NUMBER is not set: Runs the specified program on any online VE node or search the given PID on all online nodes. 2. System calls which are executed at the time of program loading, are not captured in ported "strace" command. 3. '-D' option is not supported.	1. There are multiple nodes in VE architecture. 2. VE strace command cannot attach a process which is not running on VE. So, first we have to "execv" to run VE program with --traceme flag and then will continue to trace system call. In this case system call executed at loading time are missed. 3. With -D option, tracer process runs as a detached grandchild, not as parent of the tracee. In x86_64, the process is first attached and then loaded with execve (In parent). But In VE, the process is loaded using execve (in parent) with --traceme flag instead to attaching it. VE Ptrace gets its ppid and considers it as its tracer. But with -D option, tracee process's tracer is its detached grandchild not its parent process and VE ptrace will get its parent as 0. So, ported

			strace command cannot trace VE process using its detached grandchild.
		4. To trace multiple VE PIDs, all the given PIDs must be running on same VE node, i.e in command "/opt/nec/ve/bin/strace -p pid1,pid2" should be running on same node	4. Strace internally uses 'ptrace' system call for tracing, and in case of VE, this a limitation of ptrace() system call.
		5. Command "/opt/nec/ve/bin/strace -S" will show stime as '0' for all the sytem calls.	5. There is no system time in case of VE.
		6. /opt/nec/ve/bin/strace -p pid: System call tracing, which was executed right before the command "/opt/nec/ve/bin/strace -p pid" fired, will get skipped.	6. When tracing is enabled on any running process, the system call running at that time get interrupted and then restarted (move some instruction back) using PTRACE_SYSCALL and PRACE_CONT calls of ptrace. This is handled by kernel. So, it can trace that system call. But in case of VE, ptrace with PTRACE_SYSCALL is handled by libveptrace instead of kernel. So, we cannot handle such scenario and system call tracing is skipped.
		7. The command "strace" cannot trace VE specific "ve_grow" system call.	7. "ve_grow" system calls will not write its arguments on registers. So, 'strace' command cannot read its arguments from registers to show tracing.
procps-ng-ve	pmap	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command will search the given PID on given node. - VE_NODE_NUMBER is not set: Command will search the given PID on all online nodes.	1. There are multiple nodes in VE architecture.

		2. The following values will not be applicable for VE, so the values will be zero: - Shared_Clean: Shared Pages not modified since they were mapped - Shared_Dirty: Shared Pages modified since they were mapped - Private_Clean: Private Pages not modified since they were mapped - Private_Dirty: Private Pages modified since they were mapped - Referenced: Amount of memory currently marked as referenced or accessed - Swap: Swap memory - Locked: Locked Pages which cannot be swapped out	2. VE architecture do not support the given fields.
procps-ng-ve	w	- In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes.	There are multiple nodes in VE architecture.
procps-ng-ve	tload	- In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the error "Specify the Node to get tload".	There are multiple nodes in VE architecture. 'tload' command runs continuously without exiting. So the command cannot display information for all nodes.
procps-ng-ve	vmstat	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the	1. There are multiple nodes in VE architecture.

		information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. - VE_NODE_NUMBER is not set and command executed with some delay: Command shows error message because VE_NODE_NUMBER is mandatory to execute command with some delay/count. 2. "/opt/nec/ve/bin/vmstat -m" shows error message "slabinfo not supported in this configuration". 3. Count of 'blocked processes for i/o' ("b" field) is unused in vmstat command. 4. The following values will not be applicable for VE, so the values will be zero: - swpd : Swap memory used - buff: Memory used as buffers - cache : Memory used as cache - si : Memory swapped in from disk - so : Memory swapped to disk - bi : Blocks received from a block device - bo : Blocks sent to a block device - in : Number of interrupts per second - active memory : Memory that has been used recently - inactive memory : Memory which has been less recently used - swap cache : In-memory cache for files read from the disk - total swap : Total swap space size	2. There is no slabinfo for VE. 3. VE architecture do not maintain i/o specific blocked processes. 4. VE architecture do not support the given fields.

		- used swap : Total used swap memory - free swap : Available swap memory size - sy : Time spent running kernel code. (system time) - st : Time stolen from a virtual machine. - ni (nice user cpu ticks) : Time spent by all CPU's to execute niced processes in user mode - wa (IO-wait cpu ticks): Time spent by all CPU's waiting for I/O to complete - IRQ cpu ticks : Time spent by all CPU's in servicing interrupts - softirq cpu ticks : Time spent by all CPU's in servicing softirqs - stolen cpu ticks : Time spent by all CPU's during involuntary wait - pages paged in - pages paged out - pages swapped in - pages swapped out - interrupts : counts of interrupts serviced since boot time, for each of the possible system interrupts	
procps-ng-ve	free	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. 2. VE_NODE_NUMBER is mandatory to execute “/opt/nec/ve/bin/free” command with –c or –s option.	1. There are multiple nodes in VE architecture. 2. There are multiple nodes in VE architecture.

		3. The following values will not be applicable for VE, so the values will be zero: - Mem: (buffers) :: Memory used by buffers - Mem: (cache) :: In-memory cache for files read from the disk - Low: (total) :: Total low memory - Low: (used) :: Used low memory - Low: (free) :: Free low memory - High: (total) :: Total high memory - High: (used) :: Used high memory - High: (free) :: Free high memory - '-/+ buffers/cache (total) :: Total memory for buffer and cache - '-/+ buffers/cache (used) :: Total memory used for buffer and cache - Swap: (total) :: Total swap space size - Swap: (used) :: Used swap space size - Swap: (free) :: Memory which has been evicted from RAM, and is temporarily on the disk	3. VE architecture do not support the given fields.
procps-ng-ve	uptime	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. 2. In case of VE, "/opt/nec/ve/bin/uptime –p"	1. There are multiple nodes in VE architecture. 2. Whenever VE "uptime -p" command is executed just after VEOS started then the

		command output sometimes display “up”	command output displays “up” without showing any minutes because VEOS is started just 0 minutes before. However, In case of X86_64, when system is restarted and reaches at terminal to execute the same command then it consists of some value in minutes. So ‘uptime -p’ command displays “up <value> minutes”.
procps-ng-ve	ps	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command shows the information corresponding to all online VE nodes. 2. In case of VE, “/opt/nec/ve/bin/ps” command displays blank (-) for ‘priority’ and ‘nice’ value with all the applicable options. 3. “/opt/nec/ve/bin/ps” command displays blank (-) for all the namespaces (IPC, MNT, NET, PID, USER, UTS) values. 4. “/opt/nec/ve/bin/ps” command will not display the current instruction pointer (EIP) and stack pointer (ESP) values of VE process. 5. “/opt/nec/ve/bin/ps s” command will not display PENDING signal for VE	1. There are multiple nodes in VE architecture. 2. Priority scheduling is not supported in VE, hence, getpriority()/setpriority() system calls are not supported. 3. Namespaces are not supported for VE. 4. VEOS doesn’t fetch instruction and stack pointer values from the running VE core at the time of command request for it. But VEOS provides the last updated values of it. 5. VEOS cannot distinguish between shared pending signals and signals pending for a particular TID.

		6. The following values will not be applicable for VE, so the values will be zero: - maj_flt : major page faults that have occurred with this process - min_flt : minor page faults that have occurred with this process - nwchan : Address of the kernel function where the process is sleeping - wchan : Name of the kernel function in which the process is sleeping - size : Swap space that would be required if the process were to dirty all writable pages and then be swapped out.	6. VE architecture do not support the given fields.
procps-ng-ve	top	1. In case of VE, the environment variable VE_NODE_NUMBER can be given: - VE_NODE_NUMBER is set: Command shows the information corresponding to given node. - VE_NODE_NUMBER is not set: Command show the error "Please specify VE Node to display information". 2. "/opt/nec/ve/bin/top" command displays blank (-) for 'priority' and 'nice' values. 3. "/opt/nec/ve/bin/top" command displays blank (-) for all the namespaces (IPC, MNT, NET, PID, USER, UTS) values. 4. "/opt/nec/ve/bin/top" command can sometimes display incorrect percentages values in "us" and "id"	1. There are multiple nodes in VE architecture. - 'top' command runs continuously without exiting. So the command cannot display information for all nodes. 2. Priority scheduling is not supported in VE, hence, getpriority()/setpriority() system calls are not supported. 3. Namespaces are not supported for VE. 4. In case of VE, the value of "user" will be updated as per the timer interval only (default value is 100 milli-seconds).

		of %CPU<core_id> field when pressing 1. 5. The following values will not be applicable for VE, so the values will be zero: - Percentage of the CPU for system processes - Percentage of the CPU processes waiting for I/O operations - Percentage of the CPU serving hardware interrupts - Percentage of the CPU serving software interrupts - Time stolen from a virtual machine. - Memory used by buffers - Total swap memory - Swap memory in use currently - Free swap memory - Cached memory by system - wchan : Name or the address of the kernel function in which the task is currently sleeping. - nDRT : Dirty pages count - nMaj : Major page fault count - nMin : Minor page fault count - vMj : Major page fault count delta - vMn : Minor page fault count delta	Sometimes a scenario can arrive where, when the command request VEOS to fetch the latest user time but the user time returned can be the value updated on last scheduler timer expiry and vice-versa. Hence, the values retrieved from VEOS can cause some percentage difference for the command. 5. VE architecture do not support the given fields.
psmisc-ve	prtstat	1. In case of VE, the environment variable VE_NODE_NUMBER can be given:	1. There are multiple nodes in VE architecture.

		- VE_NODE_NUMBER is set: Command will search the given PID on given node. - VE_NODE_NUMBER is not set: Command will search the given PID on all online nodes. 2. "/opt/nec/ve/bin/prtstat" command displays blank (-) for 'priority', 'rt_priority' and 'nice' values. 3. "/opt/nec/ve/bin/prtstat" command will not display the current instruction pointer (EIP) and stack pointer (ESP) values of VE process. 4. The following values will not be applicable for VE, so the values will be zero : - minflt, majflt : This Process minor & major faults - cminflt, cmajflt : Child processes minor & major faults - stime : Process's system time - guest_time : Process's guest time - delayaccr_blkio_ticks : Process's blkio - cstime : Child processes system time - cguest_time : Child processes guest time - wchan : Address where process went to sleep - nswap : Size of swap space of the process - cnswap : Size of swap space of children of the process	2. Priority scheduling is not supported in VE, hence, getpriority()/setpriority() system calls are not supported. 3. VEOS doesn't fetch instruction and stack pointer values from the running VE core at the time of command request for it. But VEOS provides the last updated values of it. 4. VE architecture do not support the given fields.
autmake-ve	automake	N/A	
autoconf-ve	autoconf	N/A	

libtool-ve	libtool	In case of VE, command “/opt/nec/ve/bin/libtool --mode=finish <libname> <path>” will not complete the installation of libtool libraries.	In case of VE, programs are compiled using musl-libc and it does not provide "ldconfig". So ported libtool command will not be able to create the necessary links and cache for dynamic linker to the generated shared libraries found at specified path.