コマンドの相違点

Revision 2.6

更新履歴

Rev.	発行日	更新内容
1.3	2018/02/28	the first version
1.4	2018/05/07	- sar コマンドと sadf コマンドが “LINUX RESTART” ではなく “VEOS RESTART” を表示するように変更 - ipcs, ipcrm コマンドの説明を追加 - vmstat, sar コマンドの ‘blocked process’ 値の説明を追加 - “ve_sysstat” サービスが対応する VEOS に連動して再起動する説明を追加
1.5	2018/06/20	- psacct-ve が VE ノード単位で制御を行うことについての説明を追加
1.6	2019/02/08	この版は VEOS v2.0.1 以降に対応します - 表紙の書式を変更
1.7	2019/04/15	この版は VEOS v2.1 以降に対応します - 複数のコマンドにおいて、デフォルトに VE ノード 0 を使用するように変更 - taskset、prlimit、time、 strace コマンドにおいて VE バイナリのみを実行可能とするように変更 - プロセスアカウンティングファイルの PPID に関して、dump-acct コマンドの説明を更新 - 実行時間の値に関して dump-acct、sa コマンドの説明を更新
1.8	2019/07	この版は VEOS v2.1.3 以降に対応します - strace コマンドでトレース中のプロセスが execve() を実行した時の動作についての相違点を追加 - プロセスアカウンティングファイルの PPID に関して、dump-acct コマンドの説明を削除 - 更新履歴の書式を変更
1.9	2020/05	この版は VEOS v2.5 以降に対応します - strace コマンドの RHEL8.1 では対応していないオプションについての相違点を追加 - VE sysstat サービスに関して、RHEL8.1 との相違点を追加
2.0	2020/07	この版は VEOS v2.6.2 以降に対応します - 制御端末(tty)の値が null の場合の’dump-acct’コマンドと’lastcomm’コマンドの用例を追加 - ’dump-acct’コマンドと’lastcomm’コマンドの追加のベクトル情報について説明を追加 - 異なるファイルフォーマット(バージョン 3/バージョン 14)のレコードを持つアカウンティングファイルを読むために利用する’convert-acct’ツールの詳細を’dump-acct’, ’lastcomm’,’sa’コマンドに追加

2.1	2020/08	3.Process accounting について の誤りを訂正
2.2	2020/09	この版は VEOS v2.7 以降に対応します - ‘ps’と‘top’, ‘w’, ‘pidstat’コマンドが異常終了したときの相違点を追加 - ‘dump-acct’と’lastcomm’コマンドが表示する‘NUMA’フィールドの説明を更新
2.3	2020/10	この版は VEOS v2.7.2 以降に対応します - RSS に関する’ps’の相違点についての説明を追加
2.4	2021/9	この版は VEOS v2.9.1 以降に対応します - ‘strace’ コマンドについて、トレーサ(strace コマンド)が SIGKILL で終了されたときのトレーシーの挙動の相違点を追加 - ‘uname’ と ‘arch’ コマンドがオンラインである最初のノードに関する情報を示すことについて修正
2.5	2023/1	この版は VEOS v2.14.1 以降に対応します - ‘free’, ‘iostat’, ‘ipcrm’, ‘ipcs’, ‘lastcomm’, ‘lscpu’, ‘lslocks’, ‘mpstat’, ‘nproc’, ‘pidstat’, ‘pmap’, ‘prtstat’, ‘ps’, ‘sadf’, ‘sar’, ‘uptime’, ‘vmstat’, ‘w’ の各コマンドについて、NQSV ジョブスケジューラに確保された VE ノードに対する動作の相違点、並びに ‘FORCE_VE_NODE_NUMBER’ 環境変数が指定された場合の相違点を追加
2.6	2023/3	この版は VEOS v3.0.2 以降に対応します - 第 3 節のプロセスアカウンティング情報の VE の特定のフィールドを更新 - 第 3 節の ‘convert-acct’ ツールの説明を更新 - VE コマンド名にプレフィックス ‘ve-’ を追加し、コマンド名を更新 - ve-sa1, ve-sa2 と ve-sadc コマンドのパスを ‘/opt/nec/ve/lib64/sa’ から ‘/opt/nec/ve/veos/lib64/sa’ にパスを更新 - ve-libtool の相違点を削除

1. 導入

このドキュメントでは、VE のために移植されたコマンドと現状の VH コマンドとの相違点をすべてリストアップすることを目的としています。

2. コマンドの相違点一覧

以下に、VE のために移植されたコマンドと現状の VH コマンドとの相違点を示します:

パッケージ名	コマンド名	相違点	理由
coreutils-arch-ve	ve-uname	VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、オンライン VE ノードのうち最初のノードの情報を表示します。	VE アーキテクチャには複数のノードがあります。
coreutils-arch-ve	ve-arch	VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、オンライン VE ノードのうち最初の情報を表示します。	VE アーキテクチャには複数のノードがあります。
coreutils-ve	ve-nproc	VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。	VE アーキテクチャには複数のノードがあります。

		VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。	
time-ve	ve-time	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: 指定されたノードでプログラムを実行します。 - VE_NODE_NUMBER が設定されていない場合: オンライン VE ノード上で指定されたプログラムを実行します。 2. VE の ve-time コマンドは VE バイナリのみ実行できるようなっており、VH バイナリは直接このコマンドから起動することはできません。 3. 以下の値は VE では適用できないため値は 0 となります: - プロセスが、カーネルモードで使用した CPU-秒数の合計値 - プロセス実行時に生じたメジャーページフォルトの数 - マイナーページフォルトの数 - プロセスがメインメモリからスワップアウトした回数	1. VE アーキテクチャには複数のノードがあります。 2. VE 用コマンドを使って VH プロセスを実行することはできません。 3. VE アーキテクチャは、指定されたフィールドをサポートしていません。
sysstat-ve	ve-pidstat	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。	1. VE アーキテクチャには複数のノードがあります。

		- VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、interval が指定された場合： デフォルト VE ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから 'interval' が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2.コマンド “/opt/nec/ve/bin/ve-pidstat <interval>” は、指定されたインターバルの時間内に必要な情報を取得できる場合、インターバルの値を考慮します。それ以外の場合、インターバルの値は無視されます。 3.コマンド"/opt/nec/ve/bin/ve-pidstat <interval>"は、プロセスがすべての VE コアで実行されている際、	2. 移植されたコマンドは、IPC 経由で VEOS から情報を取得します。x86_64 と比べ、これには時間がかかります。 3. VE の場合、”user”の値は、タイマー間隔毎にアップデートされます（そのデフォルト値は 100msec です）。

		“%usr”フィールドに 100%を超える値、あるいは 100%未満の値を表示することがあります。	場合によってシナリオは、コマンドが VEOS に最新のユーザ時間を引用するよう要求する際、あるいは要求する場所に到達することがありますが、リターンされるユーザタイムは最後のスケジューラタイマ満了時に更新された値である可能性もあり、その逆の場合もあります。したがって、VEOS から取得された値は、コマンドに数パーセントの違いを引き起こす可能性があります。
		4. コマンド“/opt/nec/ve/bin/ve-pidstat –p SELF”は統計を表示しません。	4.SELF キーワードは、統計が "pidstat"プロセス自体について報告されることを表しています。また ve-pidstat は VH プロセスであり、VE プロセスではないということを示しています。
		5.以下の値はVEでは適用できないため、値は0となります。: - %system :システムレベル、カーネルで実行中にタスクによって使用された CPU の割合 - %guest : 仮想マシン（仮想プロセッサを実行）でタスクが費やした CPU の割合 - minflt/s :タスクが 1 秒間に起こしたマイナーフォルトの数 - minflt-nr :タスクとそのすべての子により引き起こされ、インターバルの間に収	5. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		集されたマイナーフォルト。 - majflt/s :タスクが 1 秒間に起こしたメジャーフォルトの数 - majflt-nr : タスクとそのすべての子により引き起こされ、インターバルの間に収集されたメジャーフォルト - system-ms :システムレベル（カーネル）で実行中にタスクとそのすべての子が費やした合計ミリ秒数 - guest-ms : 仮想マシン（仮想プロセッサを実行）で実行中にタスクとそのすべての子が費やした合計ミリ秒数 - StkRef:スタックとして使用されるキロバイト単位のメモリ。タスクによって参照されます。 6. VE プロセスが終了すると“/opt/nec/ve/bin/ve-pidstat”コマンドが異常終了することがあります。	6. VEOS においてプロセスのチェックに成功したが、統計情報を取得しようとしたときにプロセスが終了していた場合、コマンドがエラーとなって異常終了することがあります。
sysstat-ve	ve-mpstat	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、interval が指定された場合：デフォルト VE	1. VE アーキテクチャには複数のノードがあります。

		ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから 'interval' または ’count’ が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。	
		2. "/ opt / nec / ve / bin / ve-mpstat -I" が、 "VE に割り込みは適用されません"というエラーメッセージを表示します。	2. VE に割り込みはありません。
		3. "/ opt / nec / ve / bin / ve-mpstat -A"は統計の中断を実行しない CPU の情報のみ表示します。	3. VE に割り込みはありません。
		4.コマンド"/opt/nec/ve/bin/ve-mpstat <interval>" は、プロセスがすべての VE コアで実行されている際に、” “%usr”フィールドに 100％を超える値、あるいは 100%未満の値を表示することがあります。	4. VE の場合、”user”の値は、タイマー間隔毎にアップデートされます（そのデフォルト値は 100msec です）。場合によってシナリオは、コマンドが VEOS に最新のユーザ時間を引用するよう要求する際、あるいは要求する場所に到達することがありますが、リターンされるユーザタイムは最後のスケ

			ジューラタイマ満了時に更新された値である可能性もあり、その逆の場合もあります。したがって、VEOS から取得された値は、コマンドに数パーセントの違いを引き起こす可能性があります。
		5. コマンド “/opt/nec/ve/bin/ve-mpstat -P {cpu [, ...] | ON | ALL}” は、シングル VE ノード（指定されたノード、もしくは最初のオンラインノードのいずれか）の情報を表示します。	5. コマンドの設計により、CPU の情報は 1 つのノードに対してのみ取り出すことができます。異なる VE ノードは異なる数の CPU を有することができます。
		6.以下の値はVEでは適用できないため値は0となります。: - %nice : nice 値優先で、ユーザーレベルで実行している際の CPU 使用率 - %sys : システムレベル、カーネルで実行中に、タスクにより使用された CPU の割合 - %iowait : システムが未処理のディスク I / O の要求を処理している際に、1 つまたは複数の CPU がアイドル状態だった時間の割合 - %steal : 物理 CPU からのリソースに対して、（仮想化された）CPU により費やされた時間の割合 - %irq : 割込み処理に、単数または複数の CPU が費やした時間の割合 - %soft : softirqs に、単数または複数の CPU が費やした時間の割合	6. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		- %guest : 仮想マシン（仮想プロセッサを実行）でタスクが費やした CPU の割合 - %gnice : niced guest を実行するために、単数または複数の CPU が費やした時間の割合	
sysstat-ve	ve-iostat	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、interval が指定された場合：デフォルト VE ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから 'interval' が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。	1. VE アーキテクチャには複数のノードがあります。

		2. コマンド"/opt/nec/ve/bin/ve-iostat can は、プロセスがすべての VE コアで実行されている際に、” %usr”フィールドに 100%を超える値、あるいは 100%未満の値を表示することがあります。 3.以下の値はVEでは適用できないため、値は0となります。: - %nice : nice 値優先で、ユーザーレベルで実行しているときの CPU 使用率 - %sys : システムレベル、カーネルで実行中に、タスクによって使用された CPU の割合 - %iowait : システムが未処理のディスク I / O の要求を処理している際に、単数または複数の CPU がアイドル状態だった時間の割合 - %steal : 物理 CPU からのリソースに対し、（仮想化された）CPU により費やされた時間の割合	2. VE の場合、”user”の値は、timer interval 毎にアップデートされます（そのデフォルト値は 100msec です）。場合によってシナリオは、コマンドが VEOS に最新のユーザ時間を引用するよう要求する際、あるいは要求する場所に到達することがありますが、リターンされるユーザタイムは最後のスケジューラタイマ満了時に更新された値である可能性もあり、その逆の場合もあります。したがって、VEOS から取得された値は、コマンドに数パーセントの違いを引き起こす可能性があります。 3. VE アーキテクチャは、指定されたフィールドをサポートしていません。
sysstat-ve	ve-sar	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、	1. VE アーキテクチャには複数のノードがあります。

		指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、interval が指定された場合：デフォルト VE ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから 'interval' が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。	
		2. コマンド “/opt/nec/ve/bin/ve-sar –d” は、 “ブロックデバイスデータは VE に適用されません”というエラーメッセージを表示します。	2. VE のデバイスデータはありません。
		3. “/opt/nec/ve/bin/ve-sar –n” は、 "ネットワーク統計は VE に適用されません"というエラーメッセージを表示します。	3. VE のネットワーク統計はありません。
		4. “/opt/nec/ve/bin/ve-sar –I” は、"割り込みは VE には適用されません	4. VE に割り込みはありません。

		"というエラーメッセージを表示します。 5. “/opt/nec/ve/bin/ve-sar -A” は、ネットワークの統計情報、統計の中断、デバイスブロックの表示は行いません。 6. コマンド"/opt/nec/ve/bin/ve-sar <interval>" は、プロセスがすべての VE コアで実行されている際に、“%usr”フィールドに 100%を超える値、あるいは 100%未満の値を表示することがあります。 7. ファイル名が指定されていない場合、VE 固有の‘ve-sar’コマンドは、標準システムアクティビティの日別データファイル “/var/opt/nec/ve/log/sa/sa<dd>_<node_number>"（dd パラメーターは現在の日付を表す）を使用します。 8. VE “ve-sar” コマンドは VEOS 再起動時に、”LINUX RESTART” ではなく、”VEOS RESTART” を表示します。 9. “ve-sar –q” の I/O 完了待ちのためブロック状態にあるプロセスの数は VE では未使用です。	5. VE のネットワーク、割り込み、およびブロックデバイスの統計情報はありません。 6. VE の場合、”user”の値は、timer interval 毎にアップデートされます（そのデフォルト値は 100msec です）。場合によってシナリオは、コマンドが VEOS に最新のユーザ時間を引用するよう要求する際、あるいは要求する場所に到達することがありますが、リターンされるユーザタイムは最後のスケジューラタイマ満了時に更新された値である可能性もあり、その逆の場合もあります。したがって、VEOS から取得された値は、コマンドに数パーセントの違いを引き起こす可能性があります。 7. VE アーキテクチャには複数のノードがあります。それぞれのノードごとに、別々のシステムアクティビティの日次データファイルを整備しています。 8. ve_sysstat サービスは VEOS に連動して再起動するため “VEOS RESTART”を表示します。 9. VE アーキテクチャは I/O I/O 完了待ちのためブロック

		10.以下の値はVEでは適用できないため値は0となります。: - %nice : nice 値優先で、ユーザーレベルで実行しているときの CPU 使用率 - %system & %sys : システムレベル、カーネルで実行中にタスクによって使用された CPU の割合 - %iowait : システムが未処理のディスク I / O の要求を処理している際に、単数または複数の CPU がアイドル状態だった時間の割合 - %steal : 物理 CPU からのリソースに対し、（仮想化された）CPU により費やされた時間の割合) - %irq : 割込み処理に、単数または複数の CPU が費やした時間の割合 - %soft : softirqs に、単数または複数の CPU が費やした時間の割合 - %guest : 仮想マシン（仮想プロセッサを実行）でタスクが費やした CPU の割合 - %gnice : niced guest を実行するために、単数または複数の CPU が費やした時間の割合 - pswpin/s : システムが 1 秒あたりに取り入れたスワップページの合計数 - pswpout/s : システムが 1 秒あたりに取り出したスワップページの合計数 - fault/s : システムが 1 秒あたりに引き起こしたページフォルト（メジャーフォルトとマイナーフォルトを合わせた）数	状態にあるプロセスの数を管理していません。 10. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		- majflt/s : 1 秒間にシステムが引き起こしたメジャーフォルトの数 - pgfree/s : 1 秒あたりシステムによってフリーリストに置かれたページ数 - pgscank/s : kswapd デーモンによってスキャンされた 1 秒あたりのページ数 - pgscand/s : 1 秒間に直接スキャンされたページ数 - pgsteal/s : メモリ要求を満たすために、システムがキャッシュ（ページキャッシュとスワップキャッシュ）から再要求した 1 秒間にあたりのページ数 - %vmeff : pgsteal / pgscan として計算 - Kbhugfree : まだ割り当てられていないキロバイト単位の巨大・ページメモリの量。 - %hugused : 割り当てられた巨大・ページメモリの割合 - bufpg/s : システムがバッファーとして使用する 1 秒あたりの追加メモリページ数 - campg/s : システムが 1 秒あたりにキャッシュする追加のメモリページの数 - Kbbuffers : カーネルがバッファーとして使用するキロバイト単位のメモリ - Kbcached :データをキャッシュするためにカーネルによって使用されるキロバイト単位のメモリ - Kbcommit: 現在のワークロードに必要なキロバイト単位のメモリ - %commit : メモリの全体値(RAM+swap)に対する現在の	

		ワークロードに必要なメモリの割合 - Kbactive : キロバイト単位のアクティブメモリ - Kbinact : キロバイト単位の非アクティブメモリ - Kbdirty : ディスクへの書き戻しで待機するキロバイト単位のメモリ - Kbswpfree : 空きスワップ領域の量（キロバイト） - kbswpused : スワップ領域の使用量（キロバイト） - %swpused : 使用されたスワップ領域の割合 - Kbswpcad : キャッシュされたスワップメモリの量（キロバイト） - %swpcad : 使用されているスワップ領域の量に対するキャッシュされたスワップメモリの割合 - Dentunusd : ディレクトリキャッシュ内の未使用キャッシュエントリの数	
sysstat-ve	ve-sadc	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を収集します。 - VE_NODE_NUMBER が設定されず、interval に 1 が指定された場合: すべてのオンライン VE ノードの情報を収集します。	1. VE アーキテクチャには複数のノードがあります。 - コマンド "ve-sadc" は、"sa1"により内部で呼び出されます。このコマンドは、cron コマンドによって自動的に起動され、システムアクティビティの日次データを 1 秒間隔で収集するように設計されています。 したがって、sa1 は sadc を interval “1”で呼び出

		- VE_NODE_NUMBER が設定されておらず、interval に 2 以上が指定された場合：デフォルト VE ノード 0 に関する情報が出力されます。 2. "/opt/nec/ve/veos/lib64/sa/ve-sadc –S" INT、DISK、SNMP、IPV6、XDISK オプションをサポートしていません。 次のエラーメッセージが表示されます: a) “/opt/nec/ve/veos/lib64/sa/ve-sadc –S INT”: 割り込みは VE には適用されません。 b) “/opt/nec/ve/veos/lib64/sa/ve-sadc –S DISK”: ブロックデバイスのデータは VE には適用されません。 c) “/opt/nec/ve/veos/lib64/sa/ve-sadc –S SNMP”: SNMP 統計は VE には適用されません。 d) “/opt/nec/ve/veos/lib64/sa/ve-sadc –S IPV6”: IPV6 統計は VE には適用されません。 e) “/opt/nec/ve/veos/lib64/sa/ve-sadc –S XDISK”: パーティションとディスクの統計情報は VE には適用されません。 3. "/ opt / nec / ve / veos / lib64 / sa / ve-sadc"コマンドは、パス "/ var / opt / nec / ve / log / sa"にある "sa <dd> <node_number>"ファイルの情報を収集します。outfile（情報を収集するファイル）が " - "に設定されている場合、ve-sadc は標準のシス	し、すべてのオンラインノードのシステムアクティビティデータを収集する必要があります。 2. VE では、パワーマネジメント特有のデータのみ収集可能であるため、「ve-sadc -S」オプションは、POWER、ALL および XALL オプションのみサポートしています。他のオプションに対しては、エラーメッセージを表示します。 3. VE アーキテクチャには複数のノードがあります。したがって、システム活動データを収集するためには VE ノードに従ってファイルを作成する必要があります。

		テムアクティビティの日次データファイルを使用します。これは VE では"/ var / opt / nec / ve / log / sa / sa <dd> _ <node_number>"となります。しかし標準（x86_64） 'sadc'コマンドは 、"/ var / log / sa"というパスの "sa <dd>"ファイルのシステムアクティビティの情報を収集します。（dd パラメーターは現在の日付を表します。）	
sysstat-ve	ve-sadf	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、interval が指定された場合：デフォルト VE ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから 'interval' が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合	1. VE アーキテクチャには複数のノードがあります。

		の動作について" を参照してください。 2. “/opt/nec/ve/bin/ve-sadf” can は、プロセスがすべてのVEコアで実行されている際に、” “%usr”フィールドに100%を超える値、あるいは100%未満の値を表示することがあります。 3. VE 固有の "/opt/nec/ve/bin/ve-sadf"コマンドは、ファイル "/var/opt/nec/ve/log/sa/sa<dd>_<node_number>"を使用してシステム動作データを表示します。outfile（データを抽出して標準出力に書き込むファイル）が省略されている場合、上記と同様のファイルが使用されます。 しかし標準（x86_64） 'sadf'コマンドは、ファイル "/var/log/sa/sa<dd>"を使用します。（dd パラメーターは現在の日付を表します）。 4. VE “ve-sadf” コマンドは VEOS 再起動時に、”LINUX RESTART” ではなく、”VEOS RESTART” を表示します。	2. ve-sadf コマンドは、ve-sadc によって収集されたデータを読み取ります。VE の場合、”user”の値は、タイマー間隔毎にアップデートされます（そのデフォルト値は 100msec です）。場合によってシナリオは、コマンドが VEOS に最新のユーザ時間を引用するよう要求する際、あるいは要求する場所に到達することがありますが、リターンされるユーザタイムは最後のスケジューラタイマ満了時に更新された値である可能性もあり、その逆の場合もあります。したがって、VEOS から取得された値は、コマンドに数パーセントの違いを引き起こす可能性があります。 3. VE アーキテクチャには複数のノードがあります。各ノードごとに、別々のシステムアクティビティの日次データファイルを整備しています。 4. ve_sysstat サービスは VEOS に連動して再起動するため “VEOS RESTART”を表示します。

sysstat-ve	ve-sa1	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは対応するノードの情報を収集します。 - VE_NODE_NUMBER が設定されず、interval に 1 が指定された場合: コマンドはオンラインである全ての VE ノードの情報を収集します。 - VE_NODE_NUMBER が設定されておらず、interval に 2 以上が指定された場合: デフォルト VE ノード 0 に関する情報が出力されます。 2. cron コマンドで自動的に起動される VE 固有の "/opt/nec/ve/veos/lib64/sa/ve-sa1"コマンドは、すべてのオンライン VE ノードの「/var/opt/nec/ve/log/sa/sa<dd>_<node_number>」ファイル内のシステムアクティビティの日次データを収集します。 しかし標準（x86_64） 'sa1'コマンドは、ファイル "/var/log/sa/sa<dd>"内のシステムア	1. VE アーキテクチャには複数のノードがあります。 - コマンド "ve-sa1" は、"ve-sadc"を内部で呼び出し、cron コマンドによって自動的に起動され、システムアクティビティの日次データを 1 秒間隔で収集するように設計されています。 したがって、ve-sa1 は ve-sadc を interval “1” で呼び出し、すべてのオンラインノードのシステムアクティビティデータを収集する必要があります。 2. VE アーキテクチャには複数のノードがあります。 各ノードごとに、別々のシステムアクティビティの日次データファイルを整備しています。

		クティビティの情報を収集します。（dd パラメーターは現在の日付を表します）。	
sysstat-ve	ve-sa2	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは対応するノードの日次レポートを作成します。 - VE_NODE_NUMBER が設定されていない場合: コマンドはオンラインであるすべての VE ノードについて日次レポートを作成します。 2. cron コマンドで自動的に起動される VE 固有の "/ opt / nec / ve / veos / lib64 / sa / ve-sa2"コマンドは、ファイル内のシステムアクティビティの日次データを収集します しかし、標準（x86_64） 'sa2'コマンドは、ファイル "/ var / log / sa sar <dd>"内のシステムアクティビティ情報を収集します（dd パラメーターは現在の日付を示します）。	1. VE アーキテクチャには複数のノードがあります。 2. VE アーキテクチャには複数のノードがあります。各ノードごとに、別々のシステムアクティビティの日次データファイルを整備しています。
sysstat-ve	sysstat services	1. 移植元の sysstat パッケージとは以下の点が異なります: - VE のシステムアクティビティ情報を収集するための sysstat サービスは、"/usr/lib/systemd/system/ve_systat@"で定義されています。標準の sysstat パッケージは、この情報を "/usr/lib/systemd/system/systat"で定義しています。したがって、VE 特有のサービスは、コマンド "systemctl start ve_sysstat@N.service" (N はノード番号) によって開始します。	1. VE アーキテクチャには複数のノードがあります。 VE 固有のサービスは、指定した VE ノード、またはすべての VE ノードのシステム動作情報を収集します。したがって、個別の VE 固有のサービスと設定ファイルを維持する必要があります。

		- 個別または全ての VEOS が再起動した時はいつも、そのノードに対応する ve_sysstat@N サービスも再起動します。 - VE の cron コマンドによって自動的にシステムアクティビティ情報を収集する sysstat サービスは、"/etc/cron.d/ve_sysstat"で定義されています。標準の sysstat パッケージは、この情報を "/etc/cron.d/sysstat" で定義しています。 - 複数のマクロの定義を含む設定ファイルは "/etc/sysconfig/ve_sysstat"で定義されます。標準の sysstat パッケージは、この情報を "/etc/sysconfig/sysstat"で定義しています。 2. RHEL8 環境において、パッケージのインストールの際に VE 用の sysstat パッケージは ve_systat サービスを有効(enable)にし、開始します。	2. RHEL8 において、VH 用の sysstat サービスは sysstat パッケージのインストールの際に有効(enable)にはなりますが開始はされません。ユーザはシステムの統計情報の収集を必要とする際にそれを開始する必要があります。
util-linux-ve	ve-taskset	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: 指定されたプロセスを実行するか、指定されたノード上の指定された PID を検索します。	1. VE アーキテクチャには複数のノードがあります。

		- VE_NODE_NUMBER が設定されていない場合: VE ノード 0 上で指定されたプログラムを実行するか、またはすべてのオンラインノード上の指定 PID を検索します。 2. VE の ve-taskset コマンドは VE バイナリのみ実行できるようなっており、VH バイナリは直接このコマンドから起動することはできません。	2. VE 用コマンドを使って VH プロセスを実行することはできません。
util-linux-ve	ve-lscpu	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. コマンド "/opt/nec/ve/bin/ve-lscpu -c -e"は"VE にオフラインの CPU が存在しません"というエラーメッセージを表示します。 3. コマンド ”/opt/nec/ve/bin/ve-lscpu”と-s または--sysroot オプションを指定すると、エラーメッセージ "-s または--sysroot オプションは	1. VE アーキテクチャには複数のノードがあります。 2. VE の場合、CPU をオフラインにすることはできません。 3. VE の場合、ユーザはディレクトリを指定し、CPU データを集めることはできません。

		VE ではサポートされていません"が表示されます。	
util-linux-ve	ve-prlimit	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: 指定されたプロセスを実行するか、指定されたノード上の指定された PID を検索します。 - VE_NODE_NUMBER が設定されていない場合: VE ノード 0 上で指定されたプログラムを実行するか、すべてのオンラインノード上の指定された PID を検索します。 2. VE の場合、PID は "/opt/nec/ve/bin/ve-prlimit"と "/opt/nec/ve/bin/ve-prlimit [RESOURCE OPTIONS]"を実行しリソース制限を表示するために必須であり、それ以外の場合にはエラーとなります。しかし x86_64 の場合は、現在のプロセスのリソース制限を表示します。 3. VE の場合、コマンド "/opt/nec/ve/bin/ve-prlimit> -p <pid>"は NICE と RTPRIO のリソース制限をブランク（-）として示しています。 4. "nice"と "rtprio"の制限の取得/設定は、VE ではサポートされていません。したがって、次のコマンドは PID の有無に関係なくサポートされていないため、「リソースがサポートされていません」というエラーが表示されます。:	1. VE アーキテクチャには複数のノードがあります。 2. Linux の場合、現在のプロセスにおけるリソースの制限を表示します。すなわち、VE プロセスではなく VH プロセスである 've-prlimit'コマンドの実行インスタンスを表示します。そのため、VE の場合 PID を指定せずに ve-prlimit を実行することはできません。 3. 優先度管理スケジュールは VE ではサポートされていません。したがって getpriority()/setpriority()システムコールはサポートされていません。 4. 優先度管理スケジュールは VE ではサポートされていません。したがって getpriority()/setpriority()システムコールはサポートされていません。

		a) /opt/nec/ve/bin/ve-prlimit -e=<limits> b) /opt/nec/ve/bin/ve-prlimit --nice=<limits> -p <pid> c) /opt/nec/ve/bin/ve-prlimit -r=<limits> d) /opt/nec/ve/bin/ve-prlimit --rtprio=<limits> -p <pid> 5. "memlock"の制限を取得/設定すると、指定された制限は正常に設定されますが、VE プロセスのメモリには影響しません。 6. 環境変数 VE_STACK_LIMIT によって指定されたスタック制限と共に“/opt/nec/ve/bin/ve-prlimit <VE_process>”コマンドを使用し VE プロセスを実行することはできません。 7. VE の ve-prlimit コマンドは VE バイナリのみ実行できるようなっており、VH バイナリは直接このコマンドから起動することはできません。	5. VE ではスワップメモリが無いため、全体のメモリは固定されます。 6.この場合、VEOS は ve-prlimit コマンドで指定されたリソース制限を取得し、コマンドはバイナリを解析できず、スタック制限を計算するためのスタック情報を持ちえません。 7. VE 用コマンドを使って VH プロセスを実行することはできません。
util-linux-ve	ve-lslocks	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。	1. VE アーキテクチャには複数のノードがあります。

		詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。	
util-linux-ve	ve-ipcs	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードについて情報を表示します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. VE 用 “ve-ipcs” と “ve-ipcs -a/--all” は、共有メモリの情報のみ表示します。 3. 以下のオプションはサポートしていません。 • -q/--queue • -s/--semaphores 4. “/opt/nec/ve/bin/ve-ipcs –l” が表示する共有メモリセグメント数の最大値(4096 とします)は、VH と、全ての VE ノード上共有メモリを含みます。VE_NODE_NUMBER の指定は共有メモリの制限値の表示においては無視されます。	1. VE アーキテクチャには複数のノードがあります。 2. VE 固有のメッセージキューとセマフォはありません。 3. VE 固有のメッセージキューとセマフォはありません。 4. VH と VE は同一の共有メモリ制限値を共有します。

		5. VH の共有メモリ制限が枯渇している場合、VE 用 “ve-ipcs” コマンドはエラーメッセージ “No space left on device” を表示します。 6. ユーザが作成可能な VE プロセスの共有メモリセグメントの数は “max number of segments" よりも少なくなります。 7. 以下の値は VE では適用できないため値は 0 となります。: pages swapped: スワップアウトされた共有メモリの数	5. VH の”ipcs”コマンドは /proc から情報を取得しますが、VEOS は VE 用”ve-ipcs”が VE の共有メモリの情報を取得するために、VH 上に共有メモリを作成します。そのため、共有メモリが枯渇していると、VEOS が共有メモリを作成できず、コマンドがエラーを返します。このエラーを回避するには、ユーザは VH の”ipcs” コマンドを使って、共有メモリをいくつか削除しておく必要があります。 6. “ve_exec” 自信が共有メモリを作成し、終了するまで使用します。そのため、VE プロセスが 4095 個の共有メモリを作成した場合、VH では 4096 個の共有メモリが作成されます。このうちの 1 つが “ve_exec” によってつくられた領域です。 7. VE アーキテクチャはこの値をサポートしていません。
util-linux-ve	ve-ipcrm	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードについて情報を表示します。	1. VE アーキテクチャには複数のノードがあります。

		VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. 以下のオプションはサポートしていません。 • -Q/--queue-key <msgkey> • -q/--queue-id <msgid> • -S/--semaphore-key <semkey> • -s/--semaphore-id <semid> • --all=[msg sem] 3. VE 用 “ve-ipcrm” と “ve-ipcrm -a" は、共有メモリのみ削除します。 4. VH の共有メモリが枯渇している場合、VE 用 “ve-ipcrm -a” と “ve-ipcrm --all=shm” は エラーメッセージ “No space left on device” を表示します。	2. VE 固有のメッセージキューとセマフォはありません。 3. VE 固有のメッセージキューとセマフォはありません。 4. VEOS は ”ve-ipcrm -a” または”ve-ipcrm --all=shm” が実行されたとき、削除対象の共有メモリの情報を、VH 上に共有メモリを作成してそこに書き込みます。そのため、共有メモリが枯渇していると、VEOS が共有メモリを作成できず、コマンドがエラーを返します。このエラーを回避するために、ユーザは以下を実行してください。 i. VH の “ipcs” で全ての共有メモリを表示する ii. VH の “ipcrm” コマンドでいくつかの共有メモリを削除する。 iii. “/opt/nec/ve/bin/ve-ipcrm -a“ を実行する

psacct-ve	ve-sa	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 2. 以下のコマンドを実行する時、VE_NODE_NUMBER が指定されていない場合はデフォルト VE ノード 0 が使用されます: a) /opt/nec/ve/sbin/ve-sa <filename> b) /opt/nec/ve/sbin/ve-sa --other-acct-file <filename> 3. コマンドを実行するには、VE_NODE_NUMBER を指定する必要があります。: a) /opt/nec/ve/sbin/ve-sa --other-usracct-file <filename> -s b) /opt/nec/ve/sbin/ve-sa --other-savacct-file <filename> -s 4. --ahz オプションを指定した "/opt/nec/ve/sbin/ve-sa"コマン	1. VE アーキテクチャには複数のノードがあります。 2.これらのコマンドは、'usracct と savacct'ファイル（これらのファイルが存在する場合）を使用します。正確な 'usracct と savacct'ファイルを複数の VE ノード環境で選ぶには、ノード番号を指定する必要があります。 上記以外の場合、任意の VE 固有のファイルを使用し、誤った情報が表示されます。 3.これらのコマンドは、pacct ファイルを使用して usracct / savacct ファイルを作成します。 複数のノード環境で正確な 'pacct'ファイルを選択するには、ノード番号を指定する必要があります。上記以外の場合は、任意の VE 固有の 'pacct'ファイルを使用し、誤った情報が表示されます。 4. 時間に関連したオプションの計算には AHZ 値が使用されます。VE の場合、時間は VEOS から秒またはマイク

		ドは、STDOUT に出力された値には影響しません。	ロ秒単位で受信されるため、この値使用されません。
		5. VE 固有の ported 've-sa'コマンドは、パス "/var/opt/nec/ve/account"におけるファイル usracct_ <node_number>および savacct_ <node_number>内の情報を収集します。 しかし、x86_64 の場合、 'sa'コマンドは、パス "/var/log/sa"における "usracct と savacct"ファイル内のプロセスアカウンティング情報を収集します。	5. VE アーキテクチャには複数のノードがあります。したがって、VE ノードに従ってアカウンティングファイルを作成する必要があります。
		6. ‘ve_exec’ から起動された VE プロセスが VH プロセスを起動した場合、VE アカウンティングファイルは VE プロセスのみの経過時間を記録します。VH プロセスの実行時間は、経過時間に含まれません。	6. VEOS の仕様により、VE プロセスは execve() システムコールで VH プロセスが実行されるとすぐに終了します。
		7. 以下の値はVEでは適用できないため値は0となります。: - min & min/c : マイナーページフォルトの数 - maj & maj/c : メジャーページフォルトの数 - swp & swp/c: スワップページの数 - プロセスのシステム時間 - ディスク I/O 操作(io)	7. VE アーキテクチャは、指定されたフィールドをサポートしていません。
		8. ’/opt/nec/ve/sbin/ve-sa’コマンドが‘ac_version’が 3 のレコードを検出した場合、コマンドはエラーメッセージを出力して終了します。	8. アカウンティングファイルはユーザが veos を v2.5 以前から v2.6 以降にアップデートした場合にバージョン 3 のレコードを含むことがあります。

		9. ‘/opt/nec/ve/sbin/ve-convert-acct’ツールはレコードを’ac_version 14’から’ac_version 15’または’ac_version 16’へ変更するために用意されています。“/opt/nec/ve/sbin/ve-convert-acct -t <version> <filename_to_convert_accounting>”のように、ユーザは変更したいファイルを引数として指定します。	9. 同一のアカウンティングファイルはバージョン 3、バージョン 14、バージョン 15、バージョン 16 のフォーマットの異なるタイプのレコードを持つことがあります。そして、この’convert-acct’ツールはレコードをバージョン 14 のフォーマットに変更するために利用できます。
psacct-ve	ve-accton	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合:指定されたノードに対応するアカウンティングを有効/無効にするコマンド - VE_NODE_NUMBER が設定されていない場合:すべてのオンライン VE ノードに対応するアカウンティングを有効/無効にするコマンド 2. "/var/opt/nec/ve/account/pacct_ <N>"ファイルが存在しない場合、"/opt/nec/ve/sbin/ve-accton on"はエラーを表示しません。x86_64 では、 "/var/account/pacct"ファイルが存在しない場合、<accton on>はエラーを表示します。 3. VE 用に移植された ve-accton コマンドは、パス "/var/opt/nec/ve/account"のファイル "pacct_ <node_number>"の情報を収集します。しかし、x86_64 の場合、accton コマンドは、パス "/var/log/sa"におけるファイル	1. VE アーキテクチャには複数のノードがあります。 2. VH の場合は、パッケージのインストール時に acct ファイルが作成され、VE の場合は、 've-accton on'コマンドの実行時にファイルが作成されます。acct ファイルはオンラインノードごとに作成されるため、パッケージのインストール時には、オンラインになっているノードの数は不明となります。 3. VE アーキテクチャには複数のノードがあります。したがって、VE ノードに従ってファイルを作成し、アカウンティングを有効にする必要があります。

		"pacct"のプロセスアカウンティング情報を収集します。 4. VE の場合、コマンド "/opt/nec/ve/sbin/ve-accton <filename/on>"の実行中に必要なファイルが存在しない場合、 "そのようなファイルとディレクトリは存在しません"というエラーが表示されます。"x86_64 の場合、コマンドは "アクセス拒否"というエラーを表示します。 5. VE_NODE_NUMBER が設定されていない場合に、 ”/opt/nec/ve/sbin/ve-accton <filename>” コマンドが実行されると、デフォルトでは VE ノード 0 でのみアカウンティングが有効になります。	4. VE と VH コマンドのデザインは異なります。VE 特有のコマンドは、許可の確認前にファイルの存在を確かめます。VH コマンドは、ファイルの存在の確認前に許可の確認を行います。
psacct-ve	ve-dump-acct	1. ‘ve_exec’ から起動された VE プロセスが VH プロセスを起動した場合、VE アカウンティングファイルは VE プロセスのみの経過時間を記録します。VH プロセスの実行時間は、経過時間に含まれません。 2. RHEL8 環境において、’ve-dump-acct’コマンドは以下の場合でフィールド’tty’に NULL を表示します: - VE プロセスが”nohup”コマンドによって実行されている。 - VE プロセスが system()システムコールによってバックグランド実行されている。 - VE プロセスが何らかの bash スクリプトを経由してバックグラウンドで実行されている。	1. VEOS の仕様により、VE プロセスは execve() システムコールで VH プロセスが実行されるとすぐに終了します。 2. VE プロセスに対応する端末がない場合、'tty' は NULL として表示されます。

		- VE プロセス('conftest'バイナリ)が'./configure'コマンドによって実行されている。 3. 'ac_version'が 15 または 16 の時、プロセスアカウント情報のレコードはベクトル情報を含んでいます。"/opt/nec/ve/sbin/ve-dump-acct --ve-info <filename_to_read_accounting>" のように、'--ve-info'オプション付きの've-dump-acct'コマンドがベクトル情報を表示します。 4. '/opt/nec/ve/sbin/ve-dump-acct' コマンドが'ac_version'が 3 または 14 のレコードを検出した場合、コマンドはエラーメッセージを出力して終了します。 5. '/opt/nec/ve/sbin/ve-convert-acct'ツールはレコードを'ac_version 3'と'ac_version 14'から'ac_version 15'と'ac_version 16'へ変更するために用意されています。"/opt/nec/ve/sbin/ve-convert-acct -t <version> <filename_to_convert_accounting>"のように、ユーザは変更したいファイルを引数として指定します。	3. 追加されたベクトル情報は、バージョン 14 のアカウンティングファイルで導入されます。しかし v3.0 以降のアカウンティングバージョンはバージョン 15(VE10 または VE20)とバージョン 16(VE30)になります。 4. アカウンティングファイルはユーザが veos を v2.5 以前から v2.6 以降にアップデートした場合にバージョン 3 のレコードを含むことがあります。 5. 同一のアカウンティングファイルはバージョン 3、バージョン 14、バージョン 15、バージョン 16 のフォーマットの異なるタイプのレコードを持つことがあります。そして、この've-convert-acct'ツールはレコードを同一のフォーマットに変更するために利用できます。
psacct-ve	ve-lastcomm	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは指定されたノードにのみ対応するアカウンティング情報を表示します - VE_NODE_NUMBER が設定されていない場合: コマンドはすべてのオンライン VE ノードに対応するアカ	1. VE アーキテクチャには複数のノードがあります。

		ウンティング情報を表示します VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. ’ve-lastcomm’コマンドは以下の場合でフィールド’tty’に NULL を表示します: - VE プロセスが”nohup”コマンドによって実行されている。 - VE プロセスが system()システムコールによってバックグランド実行されている。 - VE プロセスが何らかの bash スクリプトを経由してバックグラウンドで実行されている。 - VE プロセス('conftest’バイナリ)が’./configure’コマンドによって実行されている。 3. ’ac_version’が 15 または 16 の時、プロセスアカウント情報のレコードはベクトル情報を含んでいます。“/opt/nec/ve/bin/ve-lastcomm --ve-info –f <filename_to_read_accounting>”のように、’--ve-info’オプション付きの’ve-lastcomm’コマンドがベクトル情報を表示します。 4. ’/opt/nec/ve/bin/ve-lastcomm’コマンドが‘ac_version’が 3 または 14 のレコードを検出した場	2. VE プロセスに対応する端末がない場合、'tty' は NULL として表示されます。 3. 追加されたベクトル情報は、バージョン 14 のアカウンティングファイルで導入されます。 しかし v3.0 以降、アカウンティングバージョンは 15(VE10 と VE20)と 16(VE30)になります。 4. アカウンティングファイルはユーザが veos を v2.5 以前から v2.6 以降に

		合、コマンドはエラーメッセージを出力して終了します。 5. ‘/opt/nec/ve/sbin/ve-convert-acct’ツールはレコードを’ac_version 3’と’ac_version 14’から’ac_version 15’と’ac_version 16’へ変更するために用意されています。“/opt/nec/ve/sbin/ve-convert-acct -t <version> <filename_to_convert_accounting>”のように、ユーザは変更したいファイルを引数として指定します。	アップデートした場合にバージョン 3 のレコードを含むことがあります。 5. 同一のアカウンティングファイルはバージョン 3、バージョン 14、バージョン 15、バージョン 16 のフォーマットの異なるタイプのレコードを持つことがあります。レコードを同一のバージョンのフォーマットに変更するために利用します。
psacct-ve	Psacct Services	サービスに関連した以下の事項は、移植された psacct-ve パッケージとは異なります: VE のプロセスアカウンティング情報を収集するための Psacct サービスは、"/usr/lib/systemd/system/psacct-ve@.service"で定義されています。標準の psacct パッケージは、この情報を"/usr/lib/systemd/system/psacct.service"で定義しています。 - psacct-ve サービスは “systemctl start psacct-ve@$N.service” コマンドで開始します。また、“systemctl stop psacct-ve@$N.service”コマンドで停止します。($N は VE ノード番号を指定します) - psacct-ve サービスの有効/無効(enable/disable)は全ての VE ノードに対し設定されます。特定の VE ノードについて enable/disable を設定することはできません。“systemctl enable psacct-	VE アーキテクチャには複数のノードがあるため、個別または全ての VE ノードのプロセスアカウンティングを有効にするには、独立したサービスが必要です。 - VE アーキテクチャには複数のノードがあるため、特定または全ての VE ノードのプロセスアカウンティングを開始/停止するには psacct-ve@$N.service を使用します。 - サービスを有効にすると、”/etc/systemd/system/multi-user.target.wants/” に psacct-ve@.service へのリンクが作成されます。このファイルは

		ve@$N.service” ($N は VE ノード番号)でサービスを有効にした場合、 psacct-ve サービスは全ての VE ノードで有効になります。 - psacct-ve サービスを無効にする場合、事前に有効にした全てのサービスを無効にする必要があります。 例えば、psacct-ve サービスを次のコマンドで有効にしたとします。 “systemctl enable psacct-ve@$N.service” サービスを無効にするには、次のコマンドを実行します。 “systemctl disable psacct-ve@$N.service” ($N は VE ノード番号) - psacct-ve で使用される logrotate ファイルは "/etc/logrotate.d/psacct-ve" で定義されます。 標準の psacct パッケージは、この情報を "/etc/logrotate.d/psacct"で定義しています。	全ての VE ノードのアカウンティングを有効にします。 - psacct -ve サービスを無効にする場合、“etc/systemd/system/multi-user.target.wants/” にある全ての psacct-ve@.service へのリンクを削除しなければなりません。
strace-ve	ve-strace	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: 指定されたプロセスを実行するか、指定されたノード上の指定された PID を検索します。 - VE_NODE_NUMBER が設定されていない場合: VE ノード 0 上で指定されたプログラムを実行するか、すべてのオンラインノード上の指定された PID を検索します。	1. VE アーキテクチャには複数のノードがあります。

		2. プログラムのロード時に実行されるシステムコールは、移植された "ve-strace"コマンドではキャプチャされません。	2. VE ve-strace コマンドは、VE 上で実行されていないプロセスを添付することはできません。したがって、まず VE プログラムを--traceme フラグと実行するために "execv"を実行してから、システムコールのトレースを続けます。この場合、読み込み時に実行されたシステムコールは失われます。
		3. '-D'オプションはサポートされていません。	3. -D オプションでは、トレーサプロセスは、トレーシーの親としてではなく、独立した孫として実行されます。 x86_64 では、プロセスは最初に添付され、execve（親プロセス）と共にロードされます。 VE では、プロセスが添付される代わりに-、-traceme フラグが添付され execve（親）を使ってロードされます。VE Ptrace はその ppid を取得し、トレーサとみなします。しかし、-D オプションを指定すると、トレースプロセスのトレーサは親プロセスではなく、分離された孫であるため、VE ptrace はその親を 0 にします。したがって、ported strace コマンドは、分離された孫を使用して VE プロセスをトレースすることはできません。
		4. 複数の VE PID をトレースするには、指定されたすべての PID が同じ VE ノード上で実行されている必要があります。つまり、コマンド "/opt/nec/ve/bin/ve-strace -p	4. strace はトレースのために内部的に'ptrace'システムコールを使用し、VE の場合には、これが ptrace() システムコールの制限となります。

		pid1,pid2"が同じノード上で実行されていなければなりません。	
		5. コマンド "/opt/nec/ve/bin/ve-strace -S"はすべてのシステムコールに対して stime を '0'として表示します。	5. VE の場合、システム時間はありません。
		6. /opt/nec/ve/bin/ve-strace -p pid：コマンド"/opt/nec/ve/bin/ve-strace -p pid"が実行される直前に実行されたシステムコールトレースはスキップされます。	6.実行中のプロセスでトレースが有効になると、その時点で実行中のシステムコールは中断され、ptrace の PTRACE_SYSCALL および PRACE_CONT コールを使用して再開されます（いくつかの命令を戻します）。これはカーネルによって処理され、システムコールをトレースすることができます。しかし、VE の場合、PTRACE_SYSCALL 付きの ptrace はカーネルの代わりに libveptrace によって処理されます。したがって、このようなシナリオは処理できず、システムコールのトレースはスキップされます。
		7. コマンド“ve-strace” は、VE 特有の “ve_grow” システムコールをトレースすることはできません。	7. "ve_grow"システムコールはその引数をレジスタに書き込みません。 したがって、 've-strace'コマンドは、レジスタから引数を読み取ってトレースを表示することはできません。
		8. VE の ve-strace コマンドは VE バイナリのみ実行できるようなっており、VH バイナリは直接このコマンドから起動することはできません。	8. VE 用コマンドを使って VH プロセスを実行することはできません。

		9. VE 用 ve-strace コマンドが トレースしている VE プロセスが execve() システムコールまたは exec 系のライブラリ関数を実行した場合、VE 用 strace コマンドはトレース中のプロセスをデタッチします。 Linux の strace コマンドの場合、トレースされているプロセスはデタッチされず、strace コマンドはプロセスをトレースし続けます。 10. VE の ve-strace コマンドは”-e inject”, “-e fault” そして”-e kvm” オプションをサポートしていません。 11. VE の ve-strace が SIGKILL で終了されるとトレーシーは直ちに終了します。 x86 の場合、strace コマンドが SIGKILL で終了された場合、トレーシーは通常通り実行したあと終了します。	9. VE 用 strace は PTRACE_O_TRACEEXEC フラグをサポートしていません。 10. これらのオプションは RHEL8 用の strace パッケージに新しく追加されました。RHEL7 はこれらのオプションを持っていません。 11. トレーサプロセスの終了に続いてトレーシープロセスが終了するように、VE の ‘ve-strace’ コマンドの“PTRACE_O_EXITKILL” フラグが提供されました。
procps-ng-ve	ve-pmap	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノード上で指定された PID を検索します。 - VE_NODE_NUMBER が設定されていない場合:コマンドは、すべてのオンラインノード上で指定された PID を検索します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。	1. VE アーキテクチャには複数のノードがあります。

		詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. 以下の値は VE では適用できないため値は 0 となります。: - Shared_Clean: マップされてから変更されていない共有ページ - Shared_Dirty: マップされてから変更された共有ページ - Private_Clean: マップされてから変更されていないプライベートページ - Private_Dirty: マップされてから変更されたプライベートページ - Referenced: 参照済または接続済として現在マークされているメモリの量 - Swap: スワップメモリ - Locked: スワップアウトできないロックされたページ	2. VE アーキテクチャは、指定されたフィールドをサポートしていません。
procps-ng-ve	ve-w	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合	1. VE アーキテクチャには複数のノードがあります。

		の動作について" を参照してください。 2. VE プロセスが終了すると”/opt/nec/ve/bin/ve-w”コマンドが異常終了することがあります。	2. VEOS においてプロセスのチェックに成功したが、統計情報を取得しようとしたときにプロセスが終了していた場合、コマンドがエラーとなって異常終了することがあります。
procps-ng-ve	ve-tload	VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: デフォルト VE ノード 0 に関する情報が出力されます。	VE アーキテクチャには複数のノードがあります。 - ‘ve-tload'コマンドは終了せずに連続して実行されます。したがって、コマンドはすべてのノードの情報を表示することはできません。
procps-ng-ve	ve-vmstat	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、delay が指定された場合：デフォルト VE ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケ	1. VE アーキテクチャには複数のノードがあります。

		ジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから 'delay' 又は 'count' が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. "/opt/nec/ve/bin/ve-vmstat -m" は、" slabinfo not supported in this configuration" (この構成では slabinfo はサポートされていません) というエラーメッセージを表示します。 3. vmstat は、I/O 完了待ちのためブロック状態にあるプロセスの数 ("b" フィールド) は VE では未使用です。 4. 以下の値は VE では適用できないため値は 0 となります。: - swpd : 使用されたスワップメモリ - buff: バッファとして使用されるメモリ - cache : キャッシュとして使用されるメモリ - si : ディスクからスワップインされたメモリ - so : ディスクにスワップされたメモリ - bi : ブロックデバイスから受信したブロック	2. VE には slabinfo はありません。 3. VE アーキテクチャは I/O I/O 完了待ちのためブロック状態にあるプロセスの数を管理していません。 4. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		- bo : ブロックデバイスに送信されたブロック - in : 1 秒あたりの割り込み数 - active memory : 最近使用されたメモリ - inactive memory : あまり使用されていないメモリ - swap cache : ディスクから読み込まれたファイルのインメモリキャッシュ - total swap : スワップ領域の合計サイズ - used swap : 使用されたスワップメモリの合計 - free swap : 使用可能なスワップメモリサイズ - sy : カーネルコードの実行に費やされた時間。（システム時刻） - st : 仮想マシンからスチールされた時間。 - ni (nice user cpu ticks) : すべての CPU が、ユーザーモードで niced プロセスを実行するために費やす時間 - wa (IO-wait cpu ticks): すべての CPU が I / O の完了のために待機する時間 - IRQ cpu ticks : 割込みの処理に全 CPU が費やす時間 - softirq cpu ticks : すべての CPU が softirq を処理するのに費やす時間 - stolen cpu ticks : 非自発的な待機中にすべての CPU が消費した時間 - ページインしたページ - ページアウトしたページ - スワップインしたページ - スワップアウトしたページ - interrupts : ブート以降に処理された割り込みの数とそれぞれの可能なシステム割り込み	

procps-ng-ve	ve-free	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 - VE_NODE_NUMBER が設定されておらず、-c または -s オプションが指定された場合: デフォルト VE ノード 0 に関する情報が出力されます。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 VE の場合、NQSV ジョブスケジューラから '-c' 又は '-s' が指定されて実行されるとデフォルトでは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. 以下の値は VE では適用できないため値は 0 となります。: - Mem: (buffers) :: バッファによって使用されるメモリ - Mem: (cache) :: ディスクから読み込まれたファイルのインメモリキャッシュ	1. VE アーキテクチャには複数のノードがあります。 2. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		- Low: (total) :: 合計ローメモリ - Low: (used) :: 使用されたローメモリ - Low: (free) :: 空きのローメモリ - High: (total) :: 合計ハイメモリ - High: (used) :: 使用されたハイメモリ - High: (free) :: 空きのローメモリ - '-/+ buffers/cache (total) :: バッファとキャッシュの合計メモリ - '-/+ buffers/cache (used) :: バッファとキャッシュに使用された合計メモリ - Swap: (total) :: スワップ領域の合計サイズ - Swap: (used) :: 使用されたスワップ領域のサイズ - Swap: (free) :: RAM から evict され、一時的にディスク上にあるメモリ	
procps-ng-ve	ve-uptime	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合	1. VE アーキテクチャには複数のノードがあります。

		の動作について" を参照してください。 2. VE の場合, “/opt/nec/ve/bin/ve-uptime –p” コマンドの実行結果は、”up”と表示されることがあります。	2. VEOS が起動した直後に VE "ve-uptime -p"コマンドを実行すると、”0 分前”に VEOS が起動されているため、コマンド出力には何も表示されずに "up"と表示されます。 しかし、X86_64 の場合、システムが再起動され、同じコマンドを実行するためにターミナルに到達すると、いくつかの値は分単位で構成されます。 したがって、'uptime -p'コマンドは "up <value> minutes"を表示されます。
procps-ng-ve	ve-ps	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンライン VE ノードに対応した情報を表示します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合の動作について" を参照してください。 2. VE の場合、コマンド"/ opt / nec / ve / bin / ps"は、'priority'と'nice'に対	1. VE アーキテクチャには複数のノードがあります。 2.優先スケジューリングは VE ではサポートされていな

		して該当するすべてのオプションとともに空白（-）を表示します。	いため、getpriority（）/ setpriority（）システムコールはサポートされていません。
		3. "/ opt / nec / ve / bin / ve-ps"コマンドは、すべての名前空間（IPC、MNT、NET、PID、USER、UTS）の値に対して空白（-）を表示します。	3.名前空間は VE ではサポートされていません。
		4. "/ opt / nec / ve / bin / ve-ps"コマンドは、VE プロセスの現在の命令ポインタ（EIP）およびスタックポインタ（ESP）の値を表示しません。	4. VEOS は、コマンド要求時に実行中の VE コアから命令およびスタックポインタの値を取り出しません。 しかし、VEOS は最後に更新された値を提供します。
		5. コマンド“/opt/nec/ve/bin/ve-ps s”は VE では PENDING シグナルを表示しません。	5. VEOS は、共有された保留中の信号と、特定の TID に対して保留されている信号とを区別することはできません。
		6. 以下の値は VE では適用できないため値は 0 となります。: - maj_flt : このプロセスで発生したメジャーページフォルト - min_flt : このプロセスで発生したマイナーページフォルト - nwchan : スリープしているプロセスのカーネル関数のアドレス - wchan : スリープしているプロセスのカーネル関数の名前	6. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		- size : プロセスがすべての書き込み可能なページを dirty にしてスワップアウトする必要がある場合のスワップ領域 7. –L, -H, -m, -f, -T, --sort などのオプションを指定して”/opt/nec/ve/bin/ve-ps”コマンドを実行した場合、VE プロセスが終了したときに、コマンドが異常終了することがあります。 -p オプションで VE PID を指定して、”/opt/nec/ve/bin/ve-ps”コマンドを実行した場合、VE プロセスが終了したときに、コマンドが異常終了することがあります。 8. VE の場合、RSS(Resident set size)は USS(Unique set size)と PSS(Proportional set size)の合計になります。プロセスの USS はプロセスの非共有メモリのサイズの合計になります。プロセスの PSS はプロセスの共有メモリのサイズの合計になります。各 PSS 領域はアタッチしているプロセスの数によって等分されます。	7.VEOS においてプロセスのチェックに成功したが、統計情報を取得しようとしたときにプロセスが終了していた場合、コマンドがエラーとなって異常終了することがあります。
procps-ng-ve	ve-top	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - -VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノードの情報を表示します。. - VE_NODE_NUMBER が設定されていない場合: デフォルト VE ノード 0 に関する情報が出力されます。 2. "/opt/nec/ve/bin/top"コマンドは、 'priority'と 'nice'の値にブランク（-）を表示します。	1. VE アーキテクチャには複数のノードがあります。 - ‘ve-top’コマンドは終了せずに継続して実行されます。 したがってコマンドはすべてのノードの情報を表示することはできません。 2. 優先スケジューリングは VE ではサポートされていないため、getpriority() / setpriority()システムコールはサポートされていません。

		3. "/opt/nec/ve/bin/ve-top"コマンドは、すべてのネームスペース（IPC、MNT、NET、PID、USER、UTS）の値に対して空白（-）を表示します。	3. 名前空間は VE ではサポートされていません。
		4. “1を押すと、"/opt/nec/ve/bin/ve-top"コマンドは%CPU <core_id>フィールドの"us"と "id"に誤ったパーセント値を表示することがあります。	4. VE の場合、”user”の値は、timer interval 毎にアップデートされます（そのデフォルト値は 100msec です）。場合によってシナリオは、コマンドが VEOS に最新のユーザ時間を引用するよう要求する際、あるいは要求する場所に到着することがありますが、リターンされるユーザタイムは最後のスケジューラタイマ満了時に更新された値である可能性もあり、その逆の場合もあります。したがって、VEOS から取得された値は、コマンドに数パーセントの違いを引き起こす可能性があります。
		5. 以下の値は VE では適用できないため値は 0 となります。: - システムプロセスの CPU パーセンテージ - I / O 操作待ちの CPU プロセスの割合 - ハードウェアの割り込みを処理している CPU の割合 - ソフトウェアの割り込みを処理している CPU の割合 - 仮想マシンからスチールされた時間 - バッファによって使用されるメモリ - スワップメモリの合計 - 現在使用中のスワップメモリ	5. VE アーキテクチャは、指定されたフィールドをサポートしていません。

		- 空きスワップメモリ - システムによりキャッシュされたメモリ - wchan : タスクが現時点でスリープしているカーネル関数の名前またはアドレス - nDRT : Dirty ページカウント - nMaj : メジャーページフォルトカウント - nMin : マイナーページフォルトカウント - vMj : メジャーページフォルトカウントデルタ - vMn : マイナーページフォルトカウントデルタ 6. VE プロセスが終了すると”/opt/nec/ve/bin/ve-top”コマンドが異常終了することがあります。 -p オプションで VE PID を指定して、”/opt/nec/ve/bin/ve-top”コマンドを実行した場合、VE プロセスが終了したときに、コマンドが異常終了することがあります。	6. VEOS においてプロセスのチェックに成功したが、統計情報を取得しようとしたときにプロセスが終了していた場合、コマンドがエラーとなって異常終了することがあります。
psmisc-ve	ve-prtstat	1. VE の場合、環境変数 VE_NODE_NUMBER が指定できます: - VE_NODE_NUMBER が設定されている場合: コマンドは、指定されたノード上で指定された PID を検索します。 - VE_NODE_NUMBER が設定されていない場合: コマンドは、すべてのオンラインノード上で指定された PID を検索します。 VE の場合、NQSV ジョブスケジューラからコマンドが実行されるとデフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。 詳細は、"4.NQSV ジョブスケジューラからコマンドが実行された場合	1. VE アーキテクチャには複数のノードがあります。

		の動作について" を参照してください。 2. "/opt/nec/ve/bin/ve-prstat"コマンドは、 'priority'、 'rt priority'、 'nice' の値を空白（ - ）で表示します。 3. "/opt/nec/ve/bin/ve-prstat"コマンドは、VE プロセスの現在の命令ポインタ（EIP）およびスタックポインタ（ESP）の値を表示しません。 4. 以下の値は VE では適用できないため、値は 0 となります。 : - minflt, majflt : このプロセスのマイナー・メジャーフォルト - cminflt, cmajflt : 子プロセスのマイナー・メジャーフォルト - stime : プロセスのシステム時間 - guest_time : プロセスのゲスト時間 - delayaccr_blkio_ticks : プロセスの blkio - cstime : 子プロセスのゲスト時間 - cguest_time : 子プロセスのゲスト時間 - wchan : プロセスがスリープ状態になったアドレス - nswap : プロセスののスワップ領域のサイズ - cnswap : 子プロセスのスワップ領域のサイズ	2.優先スケジューリングは VE ではサポートされていないため、getpriority（）/ setpriority（）システムコールはサポートされていません。 3. VEOS は、コマンド要求時に実行中の VE コアから命令およびスタックポインタの値を取り出しません。 しかし、VEOS は最後に更新された値を提供します。 4. VE アーキテクチャは、指定されたフィールドをサポートしていません。
autmake-ve	ve-automake	N/A	

autoconf-ve	ve-autoconf	N/A	
libtool-ve	ve-libtool	N/A	

3. Process accounting について

Process accounting サービスの起動

Process accounting を利用する場合は、以下のコマンドで psacct-ve サービスを起動してください。

for i in `seq 0 7`; do if [-e /dev/veslot$i]; then systemctl enable psacct-ve@$i; systemctl restart psacct-ve@$i; fi done

psacct-ve サービスが有効であるとき、VE プロセスのプロセスアカウンティング情報はプロセスの終了時に記録されます。VE 番号に関連したプロセスアカウンティングファイルを指定した lastcomm コマンドもしくは dump-acct コマンドによってこの情報を読むことが出来ます。

/opt/nec/ve/bin/ve-lastcomm -f /var/opt/nec/ve/account/pacct_N (Where $N specifies VE node number)

例えば、以下のコマンドは VE ノード 0 番のプロセスアカウンティングファイルのアカウンティング情報を示します。

/opt/nec/ve/bin/ve-lastcomm -f /var/opt/nec/ve/account/pacct_0

Process accounting サービスの停止

Process accounting の利用を止める場合は、以下のコマンドで psacct-ve サービスを停止してください。

for i in `seq 0 7`; do systemctl stop psacct-ve@$i; systemctl disable psacct-ve@$i $i; done

追加の VE 特有のフィールド

ve-lastcomm コマンドと ve-dump-acct コマンドにおいて、”--ve-info”オプション付きのコマンドは追加の VE 特有のフィールドをプロセスアカウント情報に出力します。

そのフィールドは以下のように定義されます。

ve-lastcomm コマンド

COMMAND	Command name
FLAG	Accounting flags: S -- command executed by the super-user F -- command executed after a fork but without a following exec D -- command terminated with the generation of a core file X -- command was terminated with the signal SIGTERM
OWNER	The name of the user who ran the process
TTY	Terminal on which the process was executed

EXECUTION TIME	Time the process executed
START TIME	Time the process started
SID	Session ID
TIMESLICE	Timeslice [µs]
NTHREADS	Maximum number of threads whose state are "RUNNING" or "WAIT" at the same time
NUMA	NUMA node number
TOTAL MEM	VE's total memory usage in clicks [kb * tick]
MAX MEM	VE's max memory usage [kb]
SYSCALL	The number of system calls
TRANSDATA	Data transfer amount between VE-VH [kb]
COREBITMAP	Core bitmap
EX	Execution count
VX	Vector execution count
FPEC	Floating point data element count
VE	Vector element count
L1IMC	L1 instruction cache miss count
VECC	Vector execution in microseconds
L1MMC	L1 cache miss in microseconds
L2MMC	L2 cache miss in microseconds
VE2	Vector element count 2
VA REC	Vector arithmetic execution in microseconds
L1IMCC	L1 instruction cache miss in microseconds
VLDEC	Vector load execution in microseconds
L1OMCC	L1 operand cache miss in microseconds
PCCC	Port conflict in microseconds
LTRC	Load instruction traffic count
VLPC	Vector load packet count
STRC	Store instruction traffic count
VLEC	Vector load element count
LLVML	LLC vector load cache fill line count
LLVME	LLC vector load cache miss element count
FMAEC	Fused multiply add element count
PTCC	Power throttling in microseconds
TTCC	Thermal throttling in microseconds
L3VSAC	VLD+SLD elements accessing L3 count (Applicable for VE3 only)
L3VSME	L3 VLD+SLD miss-hit element count (Applicable for VE3 only)
L3VSML	L3 VLD+SLD miss-hit cache line count (Applicable for VE3 only)
LLVSME	LLC miss-hit element count (Applicable for VE3 only)
LLVSML	LLC miss-hit cache line count (Applicable for VE3 only)

ve-dump-acct コマンド

COMMAND	Command name
VERSION	Acct version
UTIME	User time
ETIME	Elapsed time in clock ticks [tick]
UID	User ID
GID	Group ID
PID	Process ID

PPID	Parent process ID
FLAG	Accounting flags: S -- command executed by the super-user F -- command executed after a fork but without a following exec D -- command terminated with the generation of a core file X -- command was terminated with the signal SIGTERM This field is only on RHEL8 environment.
EXIT STATUS	Process exit status This field is only on RHEL8 environment.
TTY	Terminal name This field is only on RHEL8 environment.
START TIME	Process creation time
SID	Session ID
TIMESLICE	Timeslice [µs]
NTHREADS	Maximum number of threads whose state are "RUNNING" or "WAIT" at the same time
NUMA	NUMA node number
TOTAL MEM	VE's total memory usage in clicks [kb * tick]
MAX MEM	VE's max memory usage [kb]
SYSCALL	The number of system calls
TRANSDATA	Data transfer amount between VE-VH [kb]
COREBITMAP	Core bitmap
EX	Execution count
VX	Vector execution count
FPEC	Floating point data element count
VE	Vector element count
L1IMC	L1 instruction cache miss count
VECC	Vector execution in microseconds
L1MMC	L1 cache miss in microseconds
L2MMC	L2 cache miss in microseconds
VE2	Vector element count 2
VA REC	Vector arithmetic execution in microseconds
L1IMCC	L1 instruction cache miss in microseconds
VLDEC	Vector load execution in microseconds
L1OMCC	L1 operand cache miss in microseconds
PCCC	Port conflict in microseconds
LTRC	Load instruction traffic count
VLPC	Vector load packet count
STRC	Store instruction traffic count
VLEC	Vector load element count
LLVML	LLC vector load cache fill line count
LLVME	LLC vector load cache miss element count
FMAEC	Fused multiply add element count
PTCC	Power throttling in microseconds
TTCC	Thermal throttling in microseconds

L3VSAC	VLD+SLD elements accessing L3 count (Applicable for VE3 only)
L3VSME	L3 VLD+SLD miss-hit element count (Applicable for VE3 only)
L3VSML	L3 VLD+SLD miss-hit cache line count (Applicable for VE3 only)
LLVSME	LLC miss-hit element count (Applicable for VE3 only)

LLVSML	LLC miss-hit cache line count (Applicable for VE3 only)

‘ve-convert-acct’ ツール

‘convert-acct’ツールはレコードを’ac_version 3’から’ac_version 14’へ変更するために用意されています。VEOS を v2.5 以前から v2.6 以降にアップデートした場合、バージョン 3 のフォーマットとバージョン 14 のフォーマットの 2 つの異なるタイプのレコードが同一のアカウンティングファイルに記録されることがあります。そのため、このツールはレコードをバージョン 14 のフォーマットに変更するために利用します。convert-acct ツールは単独で利用するよりも lastcomm コマンドと合わせて利用するほうが便利です。例: /opt/nec/ve/sbin/ve-convert-acct <file> | /opt/nec/ve/sbin/ve-lastcomm -f –

v3.0.2 **以降、**’ve-convert-acct’ **ツールはレコードを**’ac_version 3’と’ac_version 14’から’ac_version 15’と’ac_version 16’へ変更するために用意されています。現在、新たなオプションの’-t’ オプションが、変更が必要なアカウンティングデータのバージョンを指定するため、このコマンドに追加されました。

例: バージョン 15 のフォーマットに変更するため、次のコマンドを実行します。: /opt/nec/ve/sbin/ve-convert-acct –t 15 <file> | /opt/nec/ve/sbin/ve-lastcomm -f –

4. NQSV ジョブスケジューラからコマンドが実行された場合の動作について

下記のコマンドは、NQSV ジョブスケジューラから実行された場合、デフォルトでは NQSV ジョブスケジューラが確保したノードの情報を表示します。

- ve-free *
- ve-iostat *
- ve-ipcrm
- ve-ipcs
- ve-lastcomm
- ve-lscpu
- ve-lslocks
- ve-mpstat *
- ve-nproc
- ve-pidstat *
- ve-pmap
- ve-prtstat
- ve-ps
- ve-sadf *

- ve-sar *
- ve-uptime
- ve-vmstat *
- ve-w

具体的には、環境変数 VE_NODE_NUMBER と FORCE_VE_NODE_NUMBER に応じて、下記の通り動作します。

- 環境変数 VE_NODE_NUMBER が指定され、NQSV ジョブスケジューラからコマンドが実行された場合：
  - コマンドは NQSV ジョブスケジューラが確保したノードの情報を表示します。VE_NODE_NUMBER は無視されます。
- 環境変数 VE_NODE_NUMBER と FORCE_VE_NODE_NUMBER が指定され、NQSV ジョブスケジューラからコマンドが実行された場合：
  - FORCE_VE_NODE_NUMBER に'YES'が設定されている場合：コマンドは、VE_NODE_NUMBER で指定されたノードの情報を表示します。
  - FORCE_VE_NODE_NUMBER に'YES'以外が設定されている場合：コマンドは NQSV ジョブスケジューラが確保したノードの情報を表示します。

また、* がついているコマンドに’interval’ などのオプションを指定し、定期的に実行する場合、環境変数 VE_NODE_NUMBER に応じて、下記の通り動作します。

- VE_NODE_NUMBER が設定されている場合：コマンドは VE_NODE_NUMBER で指定されたノードの情報を表示します。
- VE_NODE_NUMBER が設定されていない場合：コマンドは NQSV ジョブスケジューラが確保した最初のノードの情報を表示します。