

ES/1 NEO

CSシリーズ

CS-KVM

使用者の手引き

第5版 2022年2月

©著作権所有者 株式会社 アイ・アイ・エム 2022年

© COPYRIGHT IIM CORPORATION, 2022

**ALL RIGHT RESERVED. NO PART OF THIS PUBLICATION MAY
REPRODUCED OR TRANSMITTED IN ANY FORM BY ANY MEANS,
ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY RECORDING,
OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM WITHOUT
PERMISSION IN WRITING FROM THE PUBLISHER.**

“RESTRICTED MATERIAL OF IIM “LICENSED MATERIALS – PROPERTY OF IIM

目次

第 1 章	CS-KVM とは	1
第 2 章	CS-KVM の構成とデータの流れ	1
第 3 章	ckvm	2
3.1.	ckvm 機能概要	2
3.2.	動作環境	2
3.2.1.	対象 KVM ホスト	2
3.2.2.	KVM への接続方法	2
第 4 章	データ収集	3
4.1.	CS-KVM で解析対象とするデータについて	3
4.2.	データ収集設定	4
4.2.1.	事前準備	4
4.2.2.	導入	6
4.2.3.	エージェントのスケジュール登録	6
4.2.4.	アンインストール	6
4.3.	統計情報ファイルの出力	7
第 5 章	iim configuration assistant	8
5.1.	転送と変換設定	8
5.1.1.	起動方法	8
5.1.2.	転送と変換設定の追加	8
5.1.3.	その他の設定	12
5.1.4.	転送と変換の実行	13

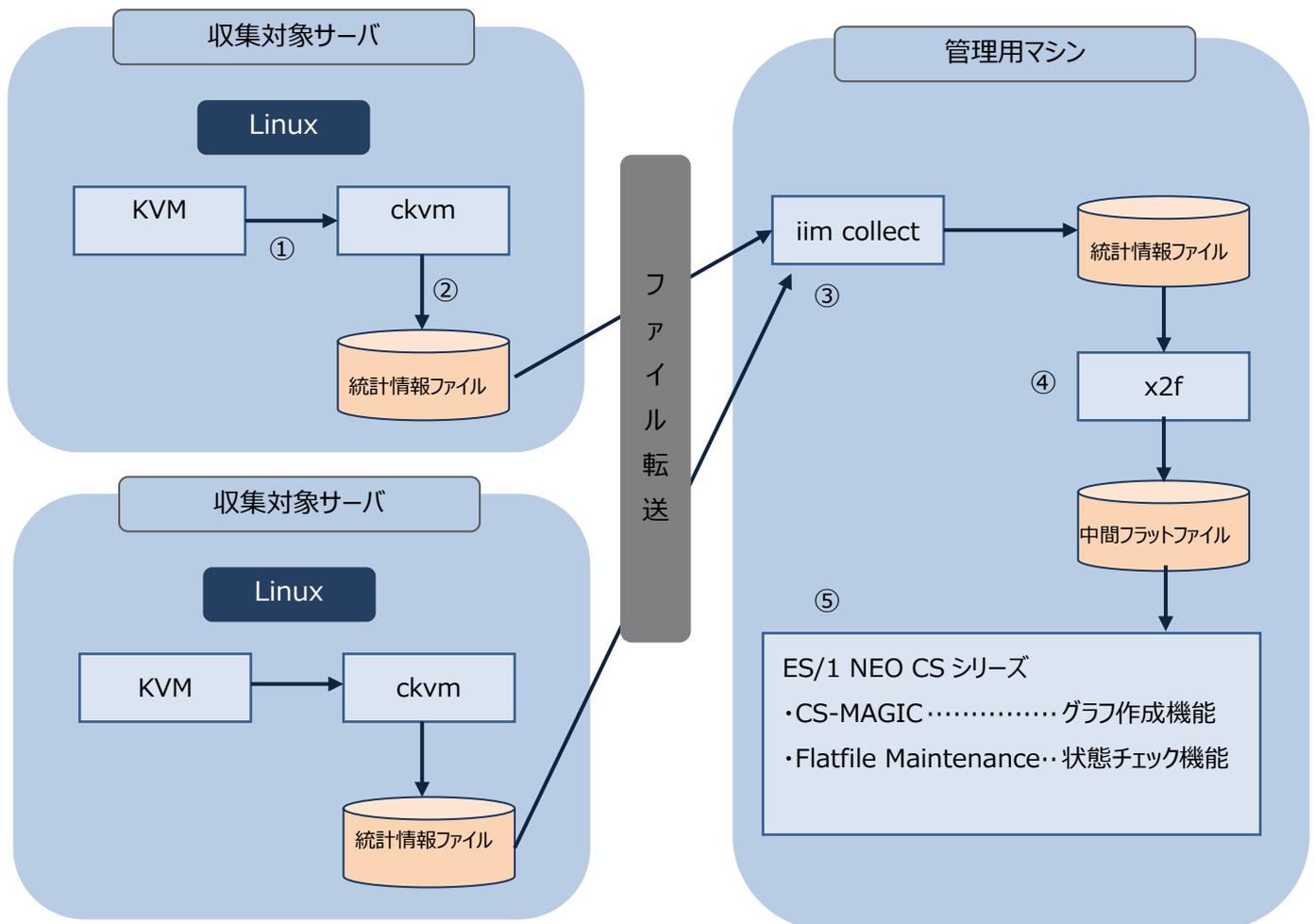
第1章 CS-KVM とは

CS-KVM とは、KVM のパフォーマンス・データを取得し、ES/1 NEO CS シリーズの共通形式データ(フラットファイル)への変換・蓄積を行うプログラムです。

CS-KVM で取り扱うパフォーマンス・データはデータ収集エージェント(ckvm)により収集され、中間ファイルとして保存されます。中間ファイルはデータ変換プログラム(x2f)により、CS-MAGIC 等で取り扱えるフラットファイルに変換されます。フラットファイルは、CS-MAGIC でグラフ作成に利用することができます。

第2章 CS-KVM の構成とデータの流れ

KVM サーバの稼働情報を収集し、グラフファイルや CSV 形式ファイルへ出力するまでの流れと、各コンポーネントの動作を説明します。



- ①データ収集エージェントにより統計情報を収集
virsh コマンドを発行し KVM の統計情報を収集します。
- ②ckvm により統計情報ファイルを生成
gzip 圧縮された統計情報ファイルが生成されます。
- ③ファイル転送プログラム(iim collect)により管理用マシンに統計情報ファイルを転送
別紙マニュアル「CS-Utility 使用者の手引き 2. iim collect」を参照
- ④データ変換プログラム(x2f)により統計情報ファイルを中間フラットファイルに変換
テキスト形式の中間フラットファイルが作成されます。
- ⑤中間フラットファイルを ES/1 NEO CS シリーズにインポートし、グラフ生成、各種チェックを実行
CS-MAGIC で作成されたグラフは Performance Web Service を利用し、Web 閲覧することができます。

第3章 ckvm

3.1. ckvm 機能概要

ckvm は、KVM のデータ収集および中間ファイルを作成するデータ収集エージェントプログラムです。ckvm は内部で virsh コマンドを発行し、KVM サーバの性能情報を取得します。

3.2. 動作環境

3.2.1. 対象 KVM ホスト

ckvm の動作環境については、「サポート環境」の「KVM」をご参照ください。

3.2.2. KVM への接続方法

ckvm は KVM との接続に libvirtd が用意する Unix ドメインソケットを使用します。これには rw(読み書き)ソケットと ro(読み取り)ソケットがありますが、ckvm は virsh コマンドを読み取り専用で実行するため ro ソケットで接続できます。

第4章 データ収集

4.1. CS-KVM で解析対象とするデータについて

ckvm は、以下のコマンドを発行して性能情報を取得します。

```
uname -s -r
virsh --readonly --connect=qemu:///system --quiet version
virsh --readonly --connect=qemu:///system --quiet uri
virsh --readonly --connect=qemu:///system --quiet sysinfo
virsh --readonly --connect=qemu:///system --quiet freecell
virsh --readonly --connect=qemu:///system --quiet hostname
virsh --readonly --connect=qemu:///system --quiet node-memory-tune
virsh --readonly --connect=qemu:///system --quiet nodeinfo
virsh --readonly --connect=qemu:///system --quiet nodecpustats
virsh --readonly --connect=qemu:///system --quiet nodememstats
virsh --readonly --connect=qemu:///system --quiet list
virsh --readonly --connect=qemu:///system --quiet iface-list --all
virsh --readonly --connect=qemu:///system --quiet net-list --all
virsh --readonly --connect=qemu:///system --quiet pool-list --all
cat /proc/meminfo
virsh --readonly --connect=qemu:///system --quiet list --all
virsh --readonly --connect=qemu:///system --quiet dommemstat --domain={domain}
virsh --readonly --connect=qemu:///system --quiet dominfo --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domiflist --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domblklist --details --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domblkerror --domain={domain}
virsh --readonly --connect=qemu:///system --quiet cpu-stats --domain={domain}
virsh --readonly --connect=qemu:///system --quiet vcpuinfo --domain={domain}
virsh --readonly --connect=qemu:///system --quiet vcpucount --domain={domain}
virsh --readonly --connect=qemu:///system --quiet schedinfo --domain={domain}
virsh --readonly --connect=qemu:///system --quiet memtune --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domifstat --interface={interface} --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domiftune --interface={interface} --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domblkinfo --device={device} --domain={domain}
virsh --readonly --connect=qemu:///system --quiet domblkstat --device={device} --domain={domain}
virsh --readonly --connect=qemu:///system --quiet iface-list
virsh --readonly --connect=qemu:///system --quiet iface-dumpxml --interface={interface}
virsh --readonly --connect=qemu:///system --quiet net-list
virsh --readonly --connect=qemu:///system --quiet net-info --network={network}
virsh --readonly --connect=qemu:///system --quiet net-dumpxml --network={network}
virsh --readonly --connect=qemu:///system --quiet pool-list
virsh --readonly --connect=qemu:///system --quiet pool-info --pool={pool}
virsh --readonly --connect=qemu:///system --quiet pool-dumpxml --pool={pool}
virsh --readonly --connect=qemu:///system --quiet vol-list --details --pool={pool}
```

4.2. データ収集設定

4.2.1. 事前準備

①エージェント用の OS ユーザの作成（収集対象サーバの OS ユーザ）

任意のユーザ名で ckvm 用の OS ユーザを作成します。このユーザで virsh コマンドを実行できる必要があります。

```
[root@host]# useradd -d /home/iim -m iim
```

②エージェント用の OS ユーザで virsh コマンドの実行確認

virsh コマンドを実行できることを確認するため、①で作成した OS ユーザで次のコマンドを実行してください。

```
[iim@host]> virsh --readonly --connect=qemu:///system --quiet version
```

以下のような結果が出力される場合、正常に virsh コマンドが発行されています。4.2.2.導入に進んでください。

(結果例)

```
コンパイル時に使用したライブラリ: libvirt 0.10.2  
使用中のライブラリ: libvirt 0.10.2  
使用中の API: QEMU 0.10.2  
実行中のハイパーバイザー: QEMU 0.12.1
```

以下のようなパーミッションエラーが出力される場合、virsh コマンドの実行に必要な権限が不足しています。③に進んでください。

```
error: Failed to connect socket to  
'/var/run/libvirt/libvirt-sock-ro': Permission deniederror: failed to connect to the hypervisor
```

③libvirtd.conf の設定値確認

①で作成したユーザで virsh コマンドを実行できるようにするため、root ユーザで次のコマンドを実行し libvirtd 設定ファイルの 2 つのキーの値を確認してください。値に応じて続く(1)~(5)の操作を一部または全部実施します。

```
[root@host]# grep -v "^#" /etc/libvirt/libvirtd.conf | grep -v "^$"
```

•unix_sock_ro_perms

読み取りの virsh コマンドで libvirtd との通信に使用する Unix ドメインソケットのパーミッションを指定します。
0770 のようにグループに読み取り権限が与えられている事を確認してください。

•unix_sock_group

virsh コマンドで libvirtd との通信に使用する Unix ドメインソケットのオーナーグループを指定します。
設定されている場合、次の(2)と(5)のみ実施してください。
設定されていない場合、次の(1)~(5)をすべて実施してください。

(1)virsh コマンドを実行できる専用の OS グループ(以降 libvirt と表記)を作成します。

```
[root@host]# groupadd libvirt
```

(2)(1)で作成した、または `unix_sock_group` で指定されているグループに①で作成した OS ユーザを属させます。

```
[root@host]# usermod -g libvirt iim
```

(3) `virsh` コマンドで `libvirtd` との通信に使用する Unix ドメインソケットのオーナーとして(1)で作成したグループを設定します。

```
[root@host]# vi /etc/libvirt/libvirtd.conf

...略...

#####
#
# UNIX socket access controls
#
# Set the UNIX domain socket group ownership. This can be used
# allow a 'trusted' set of users access to management capabilities
# without becoming root.
#
# This is restricted to 'root' by default.
unix_sock_group = "libvirt"    <===== libvirt のグループを指定

...略...
```

(4) `libvirtd` を再起動して(3)の設定を反映します。この操作は QEMU-KVM の動作に影響しないので、動作中の VM およびゲスト OS に影響を与えません。

```
(Red Hat Enterprise Linux 6 までの場合)
[root@kvmhost]# /etc/init.d/libvirtd restart
(Red Hat Enterprise Linux 7/8 の場合)
[root@kvmhost]# systemctl restart libvirtd
```

(5)再び②を実施し、①で作成した OS ユーザで `virsh` コマンドを実行できることを確認してください。

4.2.2. 導入

- ①ES/1 NEO CS シリーズのインストールメディアにおける、以下のディレクトリ内の「ckvm-bin-5.21.2.tar.gz」ファイルを、「4.2.1. 事前準備」で用意した専用ユーザのホームディレクトリに、FTP のバイナリーモードを使用してファイル転送してください。

```
インストールメディア - /ckvm/Linux
```

- ②サーバに転送した「ckvm-bin-5.21.2.tar.gz」ファイルを解凍します。

```
$ gzip -d ckvm-bin-5.21.2.tar.gz
$ tar xvf ckvm-bin-5.21.2.tar
$ ls -l
drwxr-xr-x. 4 iim libvirt 4096 5月 31 2021 ckvm
```

解凍後に作成される「ckvm ディレクトリ」を、これ以降「ckvm インストールディレクトリ」と表記します。

ckvm インストールディレクトリ/bin ディレクトリ以下の「ckvm」が実行可能になっているかを確認してください。実行可能になっていない場合は実行可能に変更してください。

Red Hat Enterprise Linux 8 の場合は、「ckvm インストールディレクトリ/binpy3」と読み替えてください。

```
$ cd ckvm/bin
$ ls -l
-r-x-----. 1 iim libvirt 202 5月 31 2021 ckvm
-r-x-----. 1 iim libvirt 12869 5月 31 2021 ckvm.py
-r-x-----. 1 iim libvirt 3815 5月 31 2021 magiclog.py
-r-x-----. 1 iim libvirt 5175 5月 31 2021 magiclog.pyc
```

4.2.3. エージェントのスケジュール登録

ckvm 行をスケジューリングしてデータを取得します。ckvm インストールディレクトリ/bin 下の「ckvm」を、「4.2.1. 事前準備」で用意した専用ユーザの cron に登録してください。

「ckvm」には引数でインターバル長を指定してください。

「crontab.sample」にサンプルの crontab があります。

Red Hat Enterprise Linux 8 用のサンプルは「crontabpy3.sample」になります。

(例)

```
0,15,30,45****/home/iim/ckvm/bin/ckvm 900
```

4.2.4. アンインストール

(1)4.2.3.で登録したスケジュールを削除します。

(2)4.2.1.①で与えた、エージェント専用ユーザに対する権限を削除します。

(3)4.2.1.①で作成した OS 側のエージェント専用ユーザーを削除します。

(4)4.2.1.①で作成した OS 側のエージェント専用ユーザーのホームディレクトリを削除します。

4.3. 統計情報ファイルの出力

スケジュール登録後に実行され、取得されたKVMの統計情報ファイルは、ckvm インストールディレクトリ以下の「out」ディレクトリに作成されます。

(例)

```
ckvm_ yyyymmddhhmmss _< ホスト名 >.xml.gz
```


サイト名、システム名については下記の注意を参照してください。

注意！

サイト／システム名は全角 31 文字以内、半角 63 文字以内で指定してください。また、下記の文字は使用できません。

- ・半角片仮名
- ・¥ / : , ; * ? " < > | .
- ・#
- ・機種依存文字 (①②③..., I II III..., (株)ドル登...等)
- ・JIS X 0201、JIS X 0208 (Shift_JIS、CP932、Windows-31J) に含まれない文字、および、外字

また、Windows のファイル名、ディレクトリ名として使用できない予約名についてもサイト／システム名として使用できません。

- ・CON、PRN、AUX、CLOCK\$, NUL、COM0～COM9、LPT0～LPT9

サイト／システム名は製品間の内部キーやデータの保存フォルダ名等に使用します。
容易に変更できませんので、将来的に変更する可能性が発生する名前は避けてください。

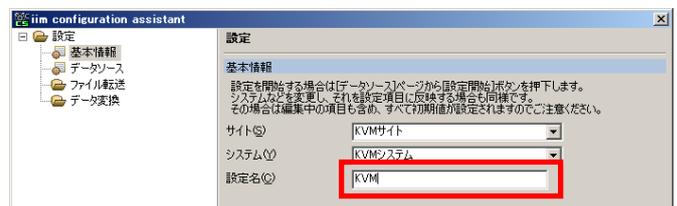
サイト／システム名として、推奨できない例

- ・次期システム
- ・本番システム
- ・テスト期間中システム

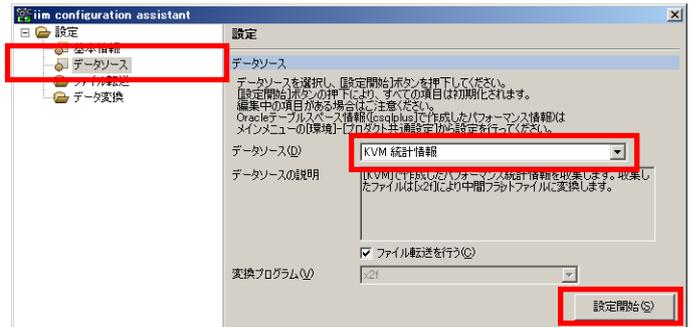
サイト／システム名が反映される箇所

- ・CS シリーズの入力データファイルを格納するフォルダ名
- ・CS シリーズの出力結果ファイル名の一部
- ・CS シリーズの出力結果ファイルを格納するフォルダ名
- ・CS シリーズの出力結果を Web コンテンツ化して Web ブラウザで閲覧する際のパス名
- ・CS シリーズの出力結果を Web コンテンツ化して専用データベースに登録する際の識別名

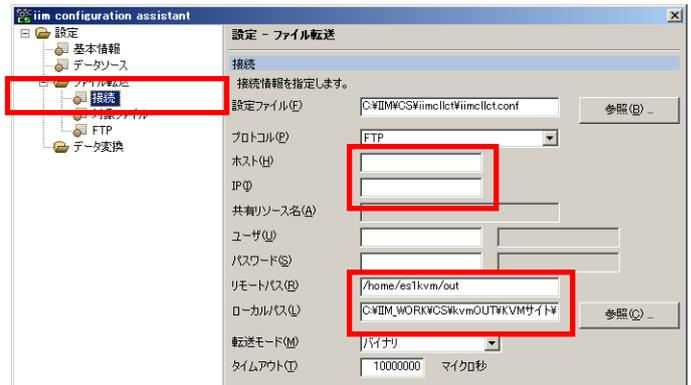
② 設定名に任意の名前を入力します。



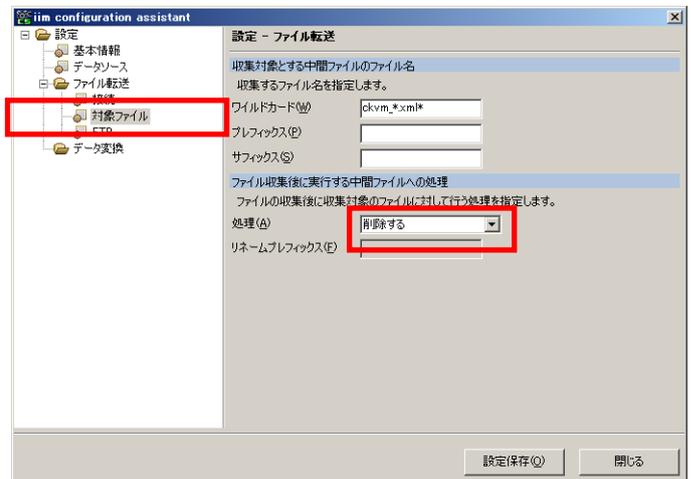
③「データソース」を選択し、「データソース(D)」に「KVM 統計情報」を選択します。
[設定開始(S)]をクリックします。



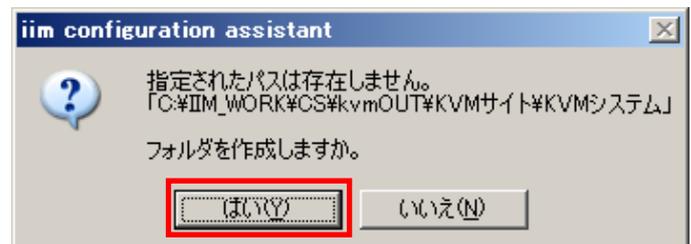
④「接続」を選択し、「ホスト(H)」、「IP(I)」、「リモートパス(R)」(ckvm が、KVM サーバ上に出力するパフォーマンスデータの格納フォルダ)、「ローカルパス(L)」(管理マシン上の格納フォルダ)を設定します。



⑤「対象ファイル」を選択し、「処理(A)」に「削除する」を選択します。
[設定保存(O)]をクリックします。



⑥「はい(Y)」をクリックします。



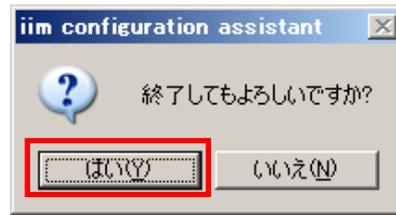
⑦[OK]をクリックします。



⑧[OK]をクリックし、設定完了です。



⑤ [はい(Y)] をクリックします。



⑥ 作成直後のバッチファイルは、コメントアウトされています。
バッチファイルを実行する前に、バッチファイルを編集し、コメントアウトを外してください。

```

::@echo off
~

:: -----
::
:: ファイル転送(iim collect)
::
:: -----

::start /d "C:¥IIM¥CS¥iimcllct" /w iimcllct.exe "C:¥IIM¥CS¥iimcllct¥iimcllct.conf"

:: -----
::
:: データ変換(x2f(i5,mysql,z/VM,KVM))
::
:: -----

::start /d "C:¥IIM¥CS¥x2f" /w x2f.exe "C:¥IIM¥CS¥x2f¥kvmconv"
    
```

5.1.4. 転送と変換の実行

iim configuration assistant で作成したバッチファイル (例 : KVM.bat) をタスクに登録、またはダブルクリックします。
バッチ実行すると、KVM サーバから統計情報ファイルを転送し、フラットファイルへの変換処理が実行されます。

