



Red Hat Enterprise Linux

7

Guide de vidage de mémoire sur incident noyau

Configuration et Analyse du vidage de mémoire sur incident noyau

Mark Flitter

Jaromír Hradílek

Petr Bokoč

Red Hat Enterprise Linux 7 Guide de vidage de mémoire sur incident noyau

Configuration et Analyse du vidage de mémoire sur incident noyau

Mark Flitter
Red Hat Customer Content Services
mflitter@redhat.com

Jaromír Hradílek
Red Hat Customer Content Services

Petr Bokoč
Red Hat Customer Content Services

Notice légale

Copyright © 2016 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

Le Guide de vidage de mémoire suite à un incident noyau documente comment configurer, tester, et utiliser le service de collecte d'informations suite à un incident noyau kdump dans Red Hat Enterprise Linux 7, et fournit un aperçu succinct sur la façon d'analyser les résultats du vidage de mémoire par l'utilitaire de débogage crash. Il s'adresse aux administrateurs de systèmes possédant des connaissances de base du système Red Hat Enterprise Linux.

Table des matières

Chapitre 1. Introduction à kdump	2
1.1. kdump et kexec	2
1.2. Besoins en mémoire	2
Chapitre 2. Installer et configurer kdump	3
2.1. Installer kdump	3
2.2. Configuration de kdump à partir de la ligne de commande	4
2.3. Configurer kdump par l'interface graphique	8
2.4. Tester la configuration kdump	14
2.5. Ressources supplémentaires	14
Chapitre 3. Analyse d'un vidage de mémoire	16
3.1. Installer l'utilitaire crash	16
3.2. Exécuter l'utilitaire crash	16
3.3. Afficher la mémoire tampon des messages	17
3.4. Afficher un backtrace	18
3.5. Explication du processus	19
3.6. Afficher des informations de mémoire virtuelle	19
3.7. Afficher les fichiers ouverts	20
3.8. Sortir de l'utilitaire	20
Annexe A. Foire Aux Questions	21
Annexe B. Cibles et configurations kdump prises en charge	22
B.1. Besoins en mémoire pour kdump	22
B.2. Limite minimum pour la réservation de mémoire automatique	22
B.3. Cibles kdump prises en charge	23
B.4. Niveaux de filtrage kdump pris en charge	23
B.5. Actions par défaut prises en charge	24
B.6. Estimation de la taille de Kdump	24
Annexe C. Historique des versions	26

Chapitre 1. Introduction à kdump

1.1. kdump et kexec

Kdump est un mécanisme de vidage de mémoire après incident noyau qui vous permet de sauvegarder le contenu de la mémoire système pour une analyse ultérieure. Repose sur **kexec**, qui peut être utilisé pour amorcer un noyau Linux à partir d'un contexte appartenant à un autre noyau, sans passer par le BIOS, et en préservant le contenu de la mémoire du premier noyau, qui serait normalement perdue.

Dans le cas d'un crash du système, kdump utilise kexec pour démarrer dans un deuxième noyau (un *capture kernel*). Ce deuxième noyau réside dans une partie réservée de la mémoire système qui est inaccessible au premier noyau. Le deuxième noyau capture ensuite le contenu de la mémoire du noyau accidenté (un *crash dump*) et l'enregistre.

1.2. Besoins en mémoire

Pour que kdump puisse accéder à un vidage de mémoire suite à incident noyau, et le sauvegarder pour l'analyser ensuite, une partie de la mémoire doit être préservée à cet effet. Si elle est réservée, cette partie de la mémoire système ne sera pas disponible au noyau principal.

Les exigences de mémoire varient en fonction de certains paramètres système. L'un des principaux facteurs est l'architecture matérielle du système. Pour connaître le nom exact de l'architecture de la machine (tel que **x86_64**) et l'imprimer dans la sortie standard, saisir la commande suivante à l'invite de commandes shell :

```
uname -m
```

Un autre facteur qui influe sur la quantité de mémoire à réserver est la quantité totale de mémoire système installée. Par exemple, sur l'architecture x86_64, la quantité de mémoire réservée sera de 160 Mo + 2 bits pour chaque 4 Ko de RAM. Sur un système avec 1 TB de la mémoire physique totale installée, cela signifie 224 Mo (160 Mo + 64 Mo). Pour obtenir une liste complète des besoins en mémoire de kdump en fonction de l'architecture du système et de la quantité de mémoire physique, consultez [Section B.1, « Besoins en mémoire pour kdump »](#).

Sur de nombreux systèmes, kdump peut estimer la quantité de mémoire nécessaire et la réserver automatiquement. Ce comportement est activé par défaut, mais ne fonctionne que sur les systèmes qui ont plus d'une certaine quantité de mémoire totale disponible, ce qui varie en fonction de l'architecture du système. Voir [Section B.2, « Limite minimum pour la réservation de mémoire automatique »](#) pour une liste des exigences minimales attenante à la réservation de mémoire automatique basée sur l'architecture du système.

Si le système a moins que la quantité minimale de mémoire requise pour que l'allocation automatique fonctionne ou si votre cas d'utilisation nécessite une valeur différente, vous pouvez configurer manuellement la quantité de mémoire réservée. Pour plus d'informations sur la façon de le faire sur la ligne de commande, voir [Section 2.2.1, « Configuration l'utilisation de la mémoire »](#). Pour plus d'informations sur la configuration de la quantité de mémoire réservée dans l'interface utilisateur graphique, consultez [Section 2.3.1, « Configuration l'utilisation de la mémoire »](#).



Important

Il est fortement recommandé de tester la configuration après avoir configuré le service kdump, même lorsque vous utilisez la réservation de mémoire automatique. Pour obtenir des instructions sur la façon de tester votre configuration, consultez [Section 2.4, « Tester la configuration kdump »](#).

Chapitre 2. Installer et configurer kdump

2.1. Installer kdump

Dans bien des cas, le service **kdump** est installé et activé par défaut sur les nouvelles Red Hat Enterprise Linux 7. Le programme d'installation **Anaconda** fournit un écran de configuration kdump pendant l'installation interactive par l'interface graphique ou texte. L'écran du programme d'installation s'intitule **Kdump** et on y accède à partir de l'écran **Sommaire d'installation**, qui permet une configuration limitée uniquement - vous ne pouvez sélectionner que si kdump sera activé ou non et la quantité de mémoire réservée. Vous trouverez des renseignements sur les besoins en mémoire pour kdump dans [Section B.1, « Besoins en mémoire pour kdump »](#). L'écran de configuration de kdump du programme d'installation est documenté dans le [Red Hat Enterprise Linux 7 Guide d'installation](#).



Note

Dans les anciennes versions de Red Hat Enterprise Linux, la configuration kdump se trouvait dans l'utilitaire **Firstboot** qui était exécuté automatiquement *une fois que* l'utilisation était terminée et que le système était démarré à nouveau pour la première fois. À partir de Red Hat Enterprise Linux 7.1, la configuration kdump est dans le programme d'installation.

Certaines options d'installation, comme les installations kickstart, n'installent, ni ne rendent actif kdump par défaut. Si tel est le cas pour votre système, et que vous souhaitez installer kdump en plus, exécutez la commande suivante en tant qu'utilisateur **root** à l'invite du shell :

```
# yum install kexec-tools
```

Cela va installer kdump et tous les autres paquets nécessaires, en supposant que votre système ait un abonnement actif ou un référentiel personnalisé contenant le package *kexec-tools* pour l'architecture de votre système.



Note

Si vous ne savez pas si kdump est installé ou non sur votre système, vous pouvez exécuter la commande **rpm** pour vérifier :

```
$ rpm -q kexec-tools
```

De plus, il existe un outil de configuration graphique, non installé par défaut, si vous utilisez la commande décrite ci-dessus. Pour installer cet utilitaire, décrit dans [Section 2.3, « Configurer kdump par l'interface graphique »](#), exécutez la commande suivante en tant qu'utilisateur **root** :

```
# yum install system-config-kdump
```

Pour plus d'informations sur la façon d'installer les nouveaux paquets dans Red Hat Enterprise Linux 7 par le gestionnaire de paquets **Yum**, voir le guide [Administrateur de systèmes Red Hat Enterprise Linux 7](#).



Important

Une certaine limitation connue dans le pilote de **Intel IOMMU** empêche le service **kdump** de récupérer l'image de vidage de mémoire. Pour pouvoir utiliser **kdump** dans des architectures Intel de manière fiable, désactiver le support IOMMU.

2.2. Configuration de kdump à partir de la ligne de commande

2.2.1. Configuration l'utilisation de la mémoire

La mémoire réservée au noyau **kdump** est toujours réservée au démarrage du système, ce qui signifie que la quantité de mémoire est spécifiée dans la configuration du chargeur de démarrage du système. Cette section explique comment modifier la quantité de mémoire réservée sur les systèmes AMD64 et Intel 64 et les serveurs IBM Power Systems en utilisant le chargeur de démarrage **grub2**, et sur IBM System z en utilisant **ZIPL**.

Procédure 2.1. Modifier les options de mémoire dans GRUB2

1. Ouvrez le fichier de configuration **/etc/default/grub** en tant qu'utilisateur **root** à l'aide d'un éditeur de texte brut tel que **vim** ou **Gedit**.
2. Dans ce fichier, trouvez la ligne commençant par **GRUB_CMDLINE_LINUX**. Elle ressemblera à ceci :

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto
rd.lvm.lv=rhel/root rhgb quiet"
```

Notez l'option **crashkernel=** surlignée, qui correspond à la mémoire réservée.

3. Modifiez la valeur de l'option **crashkernel=** pour qu'elle corresponde au montant de mémoire que vous souhaitez réserver. Ainsi, pour réserver 128 Mo de mémoire, utilisez la commande suivante :

```
crashkernel=128M
```



Note

Il y a plusieurs façons de configurer la mémoire réservée - par exemple, vous pouvez définir une valeur de décalage ou plusieurs montants de mémoires multiples basés sur le montant de mémoire RAM disponible dans le système au démarrage. C'est décrit plus loin dans cette section.

Puis, enregistrez le fichier et sortez de l'éditeur de texte.

4. Enfin, générer la configuration **GRUB2** en utilisant le fichier **default** édité. Si votre système utilise un firmware BIOS, exécutez la commande suivante :

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Sur un système équipé d'un firmware UEFI, exécutez la commande suivante à la place :


```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

Une fois que vous aurez terminé la procédure ci-dessus, le chargeur de démarrage sera configuré à nouveau, et le montant de mémoire que vous avez spécifié dans le fichier de configuration sera réservé dès le second démarrage.

Procédure 2.2. Modifier les options de mémoire dans zipl

1. Ouvrez le fichier de configuration `/etc/zipl.conf` en tant qu'utilisateur **root** à l'aide d'un éditeur de texte brut tel que **vim** ou **Gedit**.
2. Dans ce fichier, cherchez la section **parameters=**, et modifiez le paramètre **crashkernel=** (ou ajoutez-le s'il n'est pas présent). Ainsi, pour réserver 128 Mo de mémoire, utilisez la commande suivante :

```
crashkernel=128M
```



Note

Il y a plusieurs façons de configurer la mémoire réservée - par exemple, vous pouvez définir une valeur de décalage ou plusieurs montants de mémoires multiples basés sur le montant de mémoire RAM disponible dans le système au démarrage. C'est décrit plus loin dans cette section.

Puis, enregistrez le fichier et sortez de l'éditeur de texte.

3. Finalement, régénérez la configuration de **zipl** :

```
# zipl
```



Note

Exécuter la commande **zipl** uniquement, sans option supplémentaire, utilisera les valeurs par défaut. Consulter la page man de **zipl(8)** pour obtenir des informations sur les options disponibles.

Une fois que vous aurez terminé la procédure ci-dessus, le chargeur de démarrage sera configuré à nouveau, et le montant de mémoire que vous avez spécifié dans le fichier de configuration sera réservé dès le second démarrage.

L'option **crashkernel=** peut être définie de plusieurs manières. La valeur **auto** active la configuration automatique de mémoire réservée, basée sur le montant total de mémoire présente sur le système, en suivant les lignes directrices décrites dans [Section B.1, « Besoins en mémoire pour kdump »](#). Remplacer la valeur d' **auto** par un montant de mémoire spécifique pour changer ce comportement. Par exemple, pour réserver 128 Mo de mémoire, exécutez :

```
crashkernel=128M
```

Vous pouvez définir un montant de mémoire réservée comme étant variable, selon le montant de mémoire installé dans la mémoire. La syntaxe pour la réservation de mémoire est la suivante **crashkernel=<range1>:<size1>, <range2>:<size2>**. Exemple :

```
crashkernel=512M-2G:64M, 2G - :128M
```

L'exemple ci-dessus va réserver 64 Mo de mémoire si le montant total de mémoire système est de 512 Mo ou plus, et correspond à moins de 2 Go. Si le montant total de mémoire correspond à plus de 2 Go, 128 Mo de kdump seront réservés à la place.

Sur certains systèmes, il est sans doute nécessaire de réserver de la mémoire avec une valeur de décalage déterminée. Si la valeur de décalage est définie, la mémoire réservée débutera ainsi. Pour décaler la valeur de mémoire réservée, utiliser la syntaxe suivante :

```
crashkernel=128M@16M
```

L'exemple ci-dessus indique que kdump réservera 128 Mo de mémoire à partir de 16 Mo (adresse physique 01000000). Si le paramètre de décalage (offset) est défini sur 0 ou omis entièrement, kdump décalera automatiquement la mémoire réservée. Cette syntaxe peut également être utilisée lors de la définition d'une réservation de mémoire variable comme décrit ci-dessus; dans ce cas, la valeur de décalage est toujours spécifiée en dernier (par exemple, **crashkernel=512M-2G:64M, 2G - :128M@16M**).

2.2.2. Configuration du type de kdump

Quand un incident noyau est capturé, la mémoire incident peut être stockée dans un fichier qui se trouve dans le système de fichiers local, écrit directement sur un périphérique, ou envoyé via un réseau utilisant le protocole **NFS** (Network File System) ou **SSH**. Pour l'instant, on ne peut définir qu'une option, et l'option par défaut est de stocker le fichier **vmcore** dans le répertoire **/var/crash/** du système de fichiers local. Pour modifier ceci, en tant qu'utilisateur **root**, ouvrir le fichier de configuration **/etc/kdump.conf** dans l'éditeur, et éditer les options décrites ci-dessous.

Pour changer le répertoire local dans lequel le vidage de mémoire doit être sauvegardé, supprimer le signe (« # ») qui se trouve devant **#path /var/crash** et le remplacer par la valeur du chemin d'accès du répertoire souhaité.

```
path /usr/local/cores
```



Important

Dans Red Hat Enterprise Linux 7, le répertoire défini comme la cible kdump à l'aide de **path** doit exister lorsque le service **kdump** systemd est démarré - sinon le service échouera. Ce comportement est différent des versions antérieures de Red Hat Enterprise Linux, où le répertoire était créé automatiquement s'il n'existait pas lors du démarrage du service.

Optionnellement, si vous souhaitez écrire le fichier dans une autre partition, suivre la même procédure avec une des lignes commençant par **#ext4**. Vous pourrez alors utiliser un nom de périphérique **#ext4 /dev/vg/lv_kdump** line), un nom de système de fichier (**#ext4 LABEL=/boot** line) ou un UUID (la ligne **#ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937**). Changer le type de système de fichier, ainsi que le nom du périphérique, le libellé ou l'UUID aux valeurs désirées. Par exemple :

```
ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937
```

**Important**

Nous vous conseillons d'utiliser un **LABEL=** ou un **UUID=** pour spécifier les périphériques de stockage. Les noms de disques comme **/dev/sda3** ne sont pas forcément consistents à travers les démarrages. Voir le [Red Hat Enterprise Linux 7 Guide d'administration de stockage](#) pour plus d'informations sur le nommage des disques persistents.

**Important**

Lors de vidage de mémoire DASD sur s390x, il faut que les périphériques de stockage soient correctement spécifiés dans **/etc/dasd.conf** avant de procéder.

Pour écrire le vidage de mémoire directement sur un périphérique, supprimer le signe (« # ») qui se trouve devant **#raw /dev/vg/lv_kdump** et le remplacer par le nom du périphérique souhaité. Exemple :

```
raw /dev/sdb1
```

Pour stocker le vidage de mémoire dans une machine éloignée via le protocole **NFS**, supprimer le signe de hachage (« # ») qui se trouve devant **#nfs my.server.com:/export/tmp**, et remplacez sa valeur par un nom d'hôte et un chemin d'accès valides. Exemple :

```
nfs penguin.example.com:/export/cores
```

Pour stocker le vidage de mémoire dans une machine éloignée via le protocole **SSH**, supprimer le signe de hachage (« # ») qui se trouve devant **#ssh user@my.server.com**, et remplacez sa valeur par un nom d'hôte et un nom d'utilisateur valides. Pour inclure votre clé SSH dans la configuration également, supprimer le signe de hachage (« # ») qui se trouve au début de la ligne suivante **#sshkey /root/.ssh/kdump_id_rsa** et changer la valeur de l'emplacement d'une clé valide sur le serveur sur lequel vous essayez de vider la mémoire. Exemple :

```
ssh john@penguin.example.com
sshkey /root/.ssh/mykey
```

Pour plus d'informations sur la façon de configurer un serveur SSH et installer une authentification basée-clé, consulter le [Red Hat Enterprise Linux 7 Guide d'administrateur de systèmes](#).

Pour obtenir une liste des cibles actuellement prises en charge ou non, sur la base de leur type, consulter [Tableau B.3, « Cibles kdump prises en charge »](#).

2.2.3. Configuration d'un client DHCP

Pour réduire la taille du fichier de mémoire vidage **vmcore**, **kdump** vous permet de spécifier une application externe (un *core collector*) pour compresser les données, et si on le souhaite, éviter toutes les informations non pertinentes. Actuellement, le seul core collector qui soit pris en charge totalement est **makedumpfile**.

Pour activer le core collector, en tant qu'utilisateur **root**, ouvrir le fichier de configuration **/etc/kdump.conf** dans un éditeur, et supprimer le signe de hachage (« # ») du début de la ligne **#core_collector makedumpfile -l --message-level 1 -d 31**, et éditer les options de ligne de commande comme décrit ci-dessous.

Pour activer la compression de fichier, ajouter le paramètre **-c**. Exemple :

```
core_collector makedumpfile -c
```

Pour supprimer certaines pages de la mémoire de vidage, ajouter le paramètre **-d value**, avec *value* représentant une somme de valeurs de pages que vous souhaitez omettre comme décrit dans [Tableau B.4, « Niveaux de filtrage pris en charge »](#). Ainsi, pour supprimer les pages zéro ou libres, exécutez la commande suivante :

```
core_collector makedumpfile -d 17 -c
```

Voir la page man **makedumpfile(8)** pour obtenir une liste complète des options disponibles.

2.2.4. Configurer une action par défaut

Par défaut, quand **kdump** ne parvient pas à créer de mémoire de vidage dans l'emplacement cible spécifié dans [Section 2.2.2, « Configuration du type de kdump »](#), le système de fichiers root est monté et **kdump** essaie de sauvegarder le cœur localement. Afin de modifier ce comportement, en tant qu'utilisateur **root**, ouvrir le fichier de configuration **/etc/kdump.conf** dans un éditeur de texte, supprimer le signe de hachage (« # ») en début de ligne **#default shell**, et remplacer sa valeur par celle d'une action souhaitée comme indiqué dans [Tableau B.5, « Actions par défaut prises en charge »](#).

Exemple :

```
default reboot
```

2.2.5. Activer le service

Pour démarrer le démon **kdump** au démarrage, saisir ce qui suit à l'invite du shell, en tant qu'utilisateur **root**:

```
systemctl enable kdump.service
```

Cela aura pour effet d'activer le service pour **multi-user.target**. De même, si vous saisissez **systemctl stop kdump**, vous le désactiverez. Pour démarrer le service dans une session en cours, exécuter la commande suivante en tant qu'utilisateur **root** :

```
systemctl start kdump.service
```



Important

Dans Red Hat Enterprise Linux 7, le répertoire défini en tant que cible kdump doit exister quand le service systemd **kdump** démarre - sinon le service échouera. Ce comportement est différent par rapport aux autres versions de Red Hat Enterprise Linux, quand le répertoire était créé automatiquement s'il n'existait pas déjà quand on démarrait le service.

Pour obtenir plus d'informations sur systemd et configurer les services en général, consulter le [Red Hat Enterprise Linux 7 Guide d'administrateur de systèmes](#).

2.3. Configurer kdump par l'interface graphique

Pour démarrer l'utilitaire **Kernel Dump Configuration**, sélectionner **Activities** → **Other** → **Kernel crash dumps** dans le panneau de bord de **system-config-kdump** à l'invite du shell. Vous aurez accès à une fenêtre, comme expliqué dans [Figure 2.1, « Paramètres de base »](#).

L'utilitaire vous permet de configurer **kdump** ainsi que d'activer ou désactiver le démarrage du service à l'amorçage (boot). Une fois que c'est fait, cliquer sur **Apply** pour enregistrer les changements. À moins d'être déjà authentifié, on vous demandera de saisir le mot de passe du superutilisateur. L'utilisateur vous rappellera également que vous devez redémarrer le système pour pouvoir appliquer les changements que vous avez fait à la configuration.



Important

Sur les systèmes IBM System z ou PowerPC ayant **SELinux** en mode Enforcing, le booléen `kdumpgui_run_bootloader` Boolean doit être activé avant de lancer l'utilitaire Kernel Dump Configuration. Ce booléen permet à `system-config-kdump` d'exécuter le chargeur de démarrage dans le domaine `bootloader_t` SELinux. Pour activer le booléen de manière permanente, exécuter la commande suivante en tant qu'utilisateur root;

```
# setsebool -P kdumpgui_run_bootloader 1
```



Important

Lors de vidage de mémoire DASD sur s390x, il faut que les périphériques de stockage soient correctement spécifiés dans `/etc/dasd.conf` avant de procéder.

2.3.1. Configuration l'utilisation de la mémoire

L'onglet **Basic Settings** vous permet de configurer le montant de mémoire réservée au noyau **kdump**. Pour cela, sélectionner le bouton radio **Manual settings** et cliquer les flèches vers le haut ou vers le bas, à côté du champ **New kdump Memory** pour augmenter ou diminuer le montant de mémoire à réserver. Notez que le champ **Usable Memory** change proportionnellement, montrant le montant de mémoire restante dans le système. Consulter [Section 1.2, « Besoins en mémoire »](#) pour obtenir plus d'informations sur les besoins en mémoire de kdump.

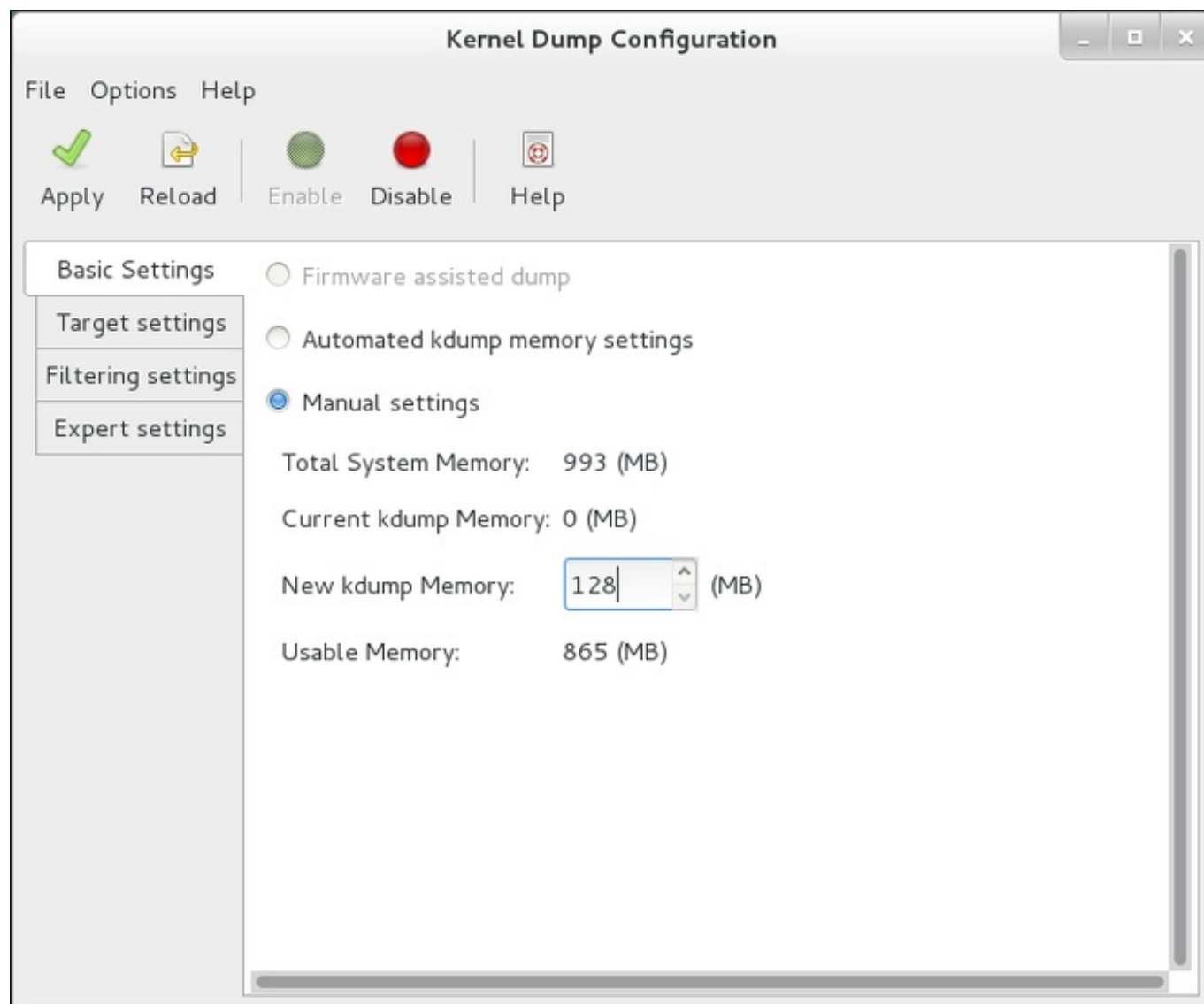


Figure 2.1. Paramètres de base

2.3.2. Configuration du type de kdump

L'onglet **Target Settings** vous permet de spécifier l'emplacement cible pour la mémoire de vidage **vmcore**. La mémoire de vidage peut être stockée en tant que fichier dans un système de fichiers local, ou bien figurer directement sur un périphérique, ou encore, elle peut être envoyée via réseau par l'intermédiaire des protocoles **NFS** (Network File System) ou **SSH** (Secure Shell).

Figure 2.2. Configurations de la cible

Pour sauvegarder la mémoire de vidage dans le système de fichiers local, sélectionner le bouton radio **Local filesystem**. Si vous le souhaitez, vous pouvez personnaliser les configurations en sélectionnant une partition différente **Partition** dans la liste du menu déroulant et un répertoire cible par le champ **Path** (chemin).



Important

Dans Red Hat Enterprise Linux 7, le répertoire défini en tant que cible kdump doit exister quand le service systemd **kdump** démarre - sinon le service échouera. Ce comportement est différent par rapport aux autres versions de Red Hat Enterprise Linux, quand le répertoire était créé automatiquement s'il n'existait pas déjà quand on démarrait le service.

Pour écrire dans la mémoire de vidage directement, sélectionner le bouton radio **Raw device**, et sélectionner le périphérique cible désiré dans le menu déroulant qui se trouve à proximité.

Pour envoyer la mémoire de vidage dans une machine à distance ou via connexion réseau, sélectionner le bouton radio **Network**. Pour utiliser le protocole **NFS**, sélectionner le bouton radio **NFS**, et remplir les champs **Server name** et **Path to directory**. Pour utiliser le protocole **SSH**, sélectionner le bouton radio **SSH**, et indiquer dans les champs **Server name**, **Path to directory**, et **User name** l'adresse du

serveur, le répertoire cible et un nom d'utilisateur valides respectivement.

Pour obtenir des informations sur la façon de configurer un serveur SSH et installer une authentification basée clé, consulter le guide [Red Hat Enterprise Linux 7 System Administrator](#). Pour obtenir une liste complète des cibles actuellement prises en charge, consulter [Tableau B.3, « Cibles kdump prises en charge »](#).

2.3.3. Configuration d'un client DHCP

L'onglet **Filtering Settings** vous permet de sélectionner le niveau de filtrage de la mémoire de vidage **vmcore**.

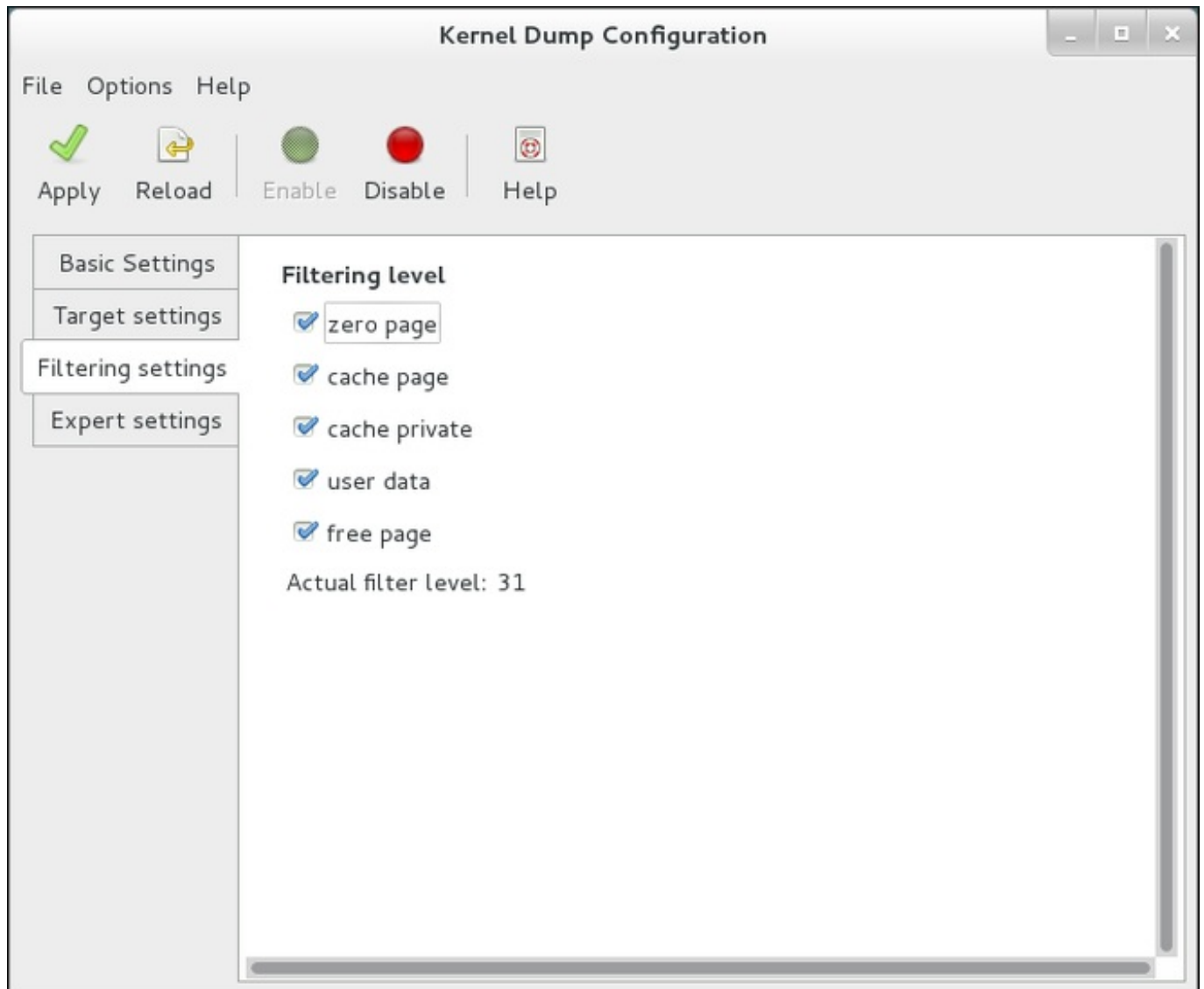


Figure 2.3. Configuration du filtrage

Pour exclure **zero page**, **cache page**, **cache private**, **user data**, ou **free page** de la mémoire de vidage, sélectionner la case à côté de l'étiquette qui convient.

2.3.4. Configurer une action par défaut

Pour sélectionner l'action appropriée quand **kdump** ne parvient pas à créer de mémoire de vidage, sélectionner une option à partir du menu déroulant **Default action**. Les options possibles sont **dump to rootfs and reboot** (l'action par défaut qui tente de sauvegarder le coeur localement et reboot le système),

reboot (reboot système), **shell** (invite shell interactive), **halt** (pour arrêter le système), et **poweroff** (pour éteindre le système).

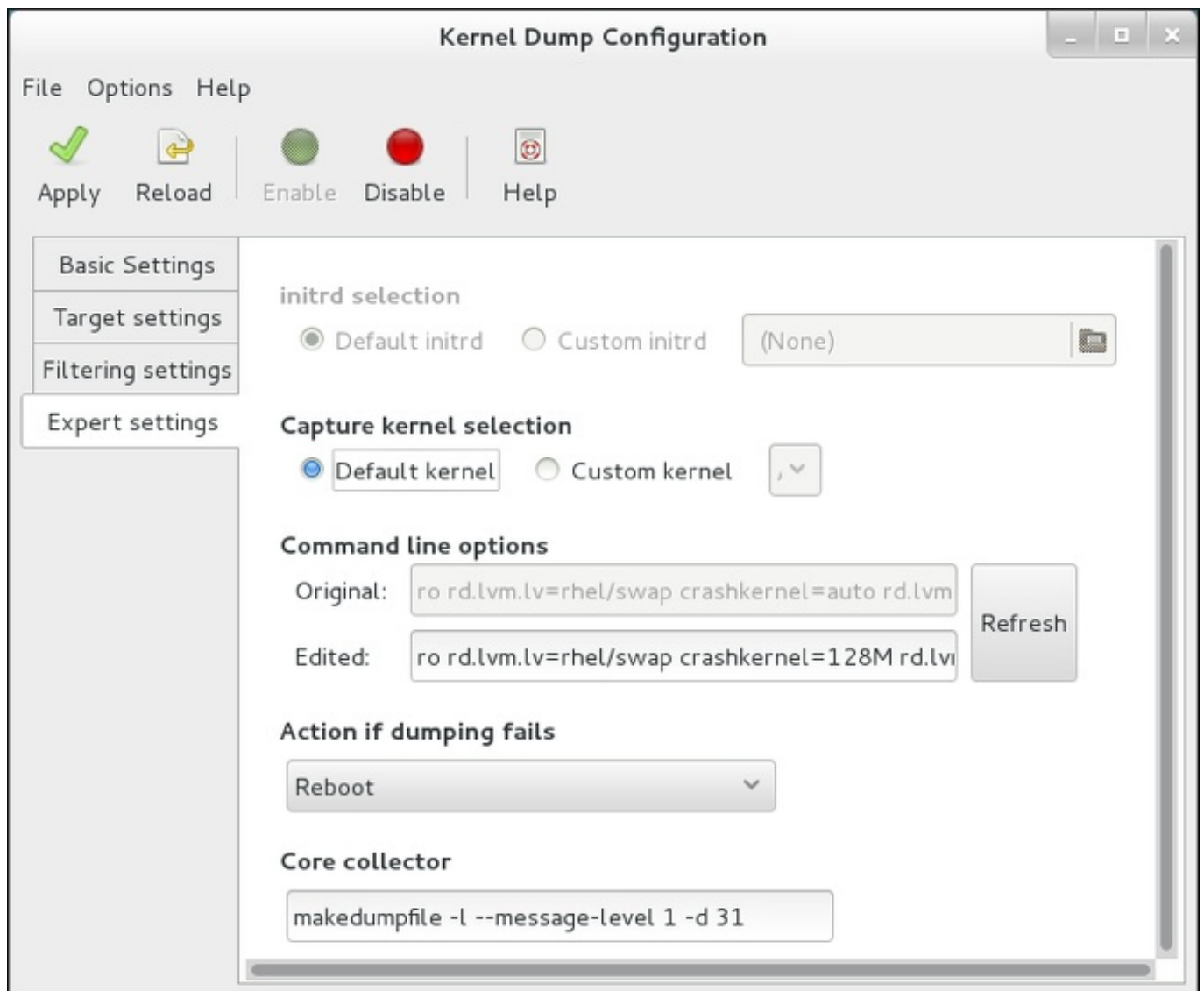


Figure 2.4. Configuration du filtrage

Afin de personnaliser les options qui sont passées au core collector **makedumpfile**, modifier les champs de texte du **Core collector**; consulter [Section 2.2.3, « Configuration d'un client DHCP »](#) pour obtenir plus d'informations.

2.3.5. Activer le service

Pour démarrer le service **kdump** dès l'amorçage, cliquer sur le bouton **Enable** qui se trouve sur la barre d'outils, puis cliquer sur **Apply**. Cela permettra l'activation du service pour **multi-user.target**. Cliquer sur le bouton **Disable** et confirmer en cliquant sur **Apply** qui désactivera le service immédiatement.



Important

Dans Red Hat Enterprise Linux 7, le répertoire défini en tant que cible **kdump** doit exister quand le service **systemd** **kdump** démarre - sinon le service échouera. Ce comportement est différent par rapport aux autres versions de Red Hat Enterprise Linux, quand le répertoire était créé automatiquement s'il n'existait pas déjà quand on démarrait le service.

Pour obtenir plus d'informations sur les cibles **systemd** et configurer les services en général, consulter le [Red Hat Enterprise Linux 7 Guide d'administrateurs de systèmes](#).

2.4. Tester la configuration **kdump**



Avertissement

Les commandes ci-dessous entraîneront un plantage du noyau. Utilisez-les donc avec précaution en suivant ces étapes, et ne surtout pas les utiliser dans un système à dessein de production.

Pour tester la configuration, rebooter le système avec **kdump** activé, et veillez à ce que le système soit en cours d'exécution :

```
~]# systemctl is-active kdump
active
```

La mise à jour d'un paquetage est semblable à son installation. Saisir la commande suivante à l'invite du shell :

```
echo 1 > /proc/sys/kernel/sysrq
echo c > /proc/sysrq-trigger
```

Cela entraînera le plantage du noyau Linux, et le fichier **address-YYYY-MM-DD-HH:MM:SS/vmcore** sera copié à l'emplacement que vous aurez sélectionné dans la configuration (c-a-d dans **/var/crash/** par défaut).



Note

En plus de confirmer la validité de la configuration, cette action peut être utilisée pour enregistrer la durée nécessaire pour qu'un vidage de mémoire puisse être complété s'il est effectué à l'occasion d'une charge de test représentative.

2.5. Ressources supplémentaires

2.5.1. Documentation installée

- ✧ **kdump.conf(5)** — page man pour le fichier de configuration **/etc/kdump.conf** qui contient la documentation complète des options disponibles.
- ✧ **zipl.conf(5)** — page man du fichier de configuration **/etc/zipl.conf**.

- ✱ **zipl(8)** — page man de l'utilitaire du chargeur de démarrage **zipl** des IBM System z.
- ✱ **makedumpfile(8)** — page man du core collector **makedumpfile**.
- ✱ **kexec(8)** — page man de **kexec**.
- ✱ **crash(8)** — page man de l'utilitaire **crash**.
- ✱ **/usr/share/doc/kexec-tools-version/kexec-kdump-howto.txt** — aperçu général de **kdump**, installation et utilisation de **kexec**.

2.5.2. Documentation en ligne

<https://access.redhat.com/site/solutions/6038>

L'article de la base de connaissance de Red Hat sur la configuration **kexec** et **kdump**.

<https://access.redhat.com/site/solutions/223773>

L'article de base de connaissance de Red Hat sur les cibles **kdump** prises en charge.

<http://people.redhat.com/anderson/>

system-config-users

<https://www.gnu.org/software/grub/>

La page d'accueil et la documentation du chargeur de démarrage **GRUB2**.

Chapitre 3. Analyse d'un vidage de mémoire

Pour déterminer la cause du plantage-système, ayez recours à l'utilitaire **crash**, qui fournit une invite similaire à GDB (GNU Debugger). Cet utilitaire vous permettra d'analyser de façon interactive un système Linux, ainsi que le vidage de mémoire créé par **netdump**, **diskdump**, **xendump**, ou **kdump**.

3.1. Installer l'utilitaire crash

Pour installer l'outil d'analyse **crash**, veuillez saisir la commande suivante dans une invite de shell en tant qu'utilisateur **root** :

```
yum install crash
```

En plus de **crash**, il faut également installer le package *kernel-debuginfo* pour le noyau actuel, qui fournit les données utiles à l'analyse du vidage de mémoire. Pour installer *kernel-debuginfo*, nous utilisons la commande **debuginfo-install** en tant qu'utilisateur **root**:

```
debuginfo-install kernel
```

Pour plus d'informations sur la façon d'installer les nouveaux paquets dans Red Hat Enterprise Linux par le gestionnaire de paquets **Yum**, voir le guide [Administrateur de systèmes Red Hat Enterprise Linux 7](#).

3.2. Exécuter l'utilitaire crash

Pour démarrer l'utilitaire, saisir la commande sous la forme suivante à l'invite shell :

```
crash /var/crash/<timestamp>/vmcore  
/usr/lib/debug/lib/modules/<kernel>/vmlinux
```

Notez que la version *<kernel>* doit être la même que celle qui a été capturée par **kdump**. Pour savoir quel noyau vous exécutez actuellement, saisir la commande **uname -r**.

Exemple 3.1. Exécuter l'utilitaire crash

```
~]# crash /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux \  
/var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore  
  
crash 5.0.0-23.el6  
Copyright (C) 2002-2010 Red Hat, Inc.  
Copyright (C) 2004, 2005, 2006 IBM Corporation  
Copyright (C) 1999-2006 Hewlett-Packard Co  
Copyright (C) 2005, 2006 Fujitsu Limited  
Copyright (C) 2006, 2007 VA Linux Systems Japan K.K.  
Copyright (C) 2005 NEC Corporation  
Copyright (C) 1999, 2002, 2007 Silicon Graphics, Inc.  
Copyright (C) 1999, 2000, 2001, 2002 Mission Critical Linux, Inc.  
This program is free software, covered by the GNU General Public License,  
and you are welcome to change it and/or distribute copies of it under  
certain conditions. Enter "help copying" to see the conditions.  
This program has absolutely no warranty. Enter "help warranty" for  
details.
```

```

GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-pc-linux-gnu"...

        KERNEL: /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux
        DUMPFILE: /var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore [PARTIAL
DUMP]
        CPUS: 4
        DATE: Wed Aug 25 08:44:47 2010
        UPTIME: 00:09:02
LOAD AVERAGE: 0.00, 0.01, 0.00
        TASKS: 140
        NODENAME: hp-dl320g5-02.lab.bos.redhat.com
        RELEASE: 2.6.32-69.el6.i686
        VERSION: #1 SMP Tue Aug 24 10:31:45 EDT 2010
        MACHINE: i686 (2394 Mhz)
        MEMORY: 8 GB
        PANIC: "Oops: 0002 [#1] SMP " (check log for details)
        PID: 5591
        COMMAND: "bash"
        TASK: f196d560 [THREAD_INFO: ef4da000]
        CPU: 2
        STATE: TASK_RUNNING (PANIC)

crash>

```

3.3. Afficher la mémoire tampon des messages

Pour afficher la mémoire tampon des messages, saisir la commande **log** lorsque vous y serez invités.

Exemple 3.2. Afficher la mémoire tampon des messages du noyau

```

crash> log
... several lines omitted ...
EIP: 0060:[<c068124f>] EFLAGS: 00010096 CPU: 2
EIP is at sysrq_handle_crash+0xf/0x20
EAX: 00000063 EBX: 00000063 ECX: c09e1c8c EDX: 00000000
ESI: c0a09ca0 EDI: 00000286 EBP: 00000000 ESP: ef4dbf24
DS: 007b ES: 007b FS: 00d8 GS: 00e0 SS: 0068
Process bash (pid: 5591, ti=ef4da000 task=f196d560 task.ti=ef4da000)
Stack:
c068146b c0960891 c0968653 00000003 00000000 00000002 efade5c0 c06814d0
<0> ffffffff c068150f b7776000 f2600c40 c0569ec4 ef4dbf9c 00000002 b7776000
<0> efade5c0 00000002 b7776000 c0569e60 c051de50 ef4dbf9c f196d560 ef4dbfb4
Call Trace:
[<c068146b>] ? __handle_sysrq+0xfb/0x160
[<c06814d0>] ? write_sysrq_trigger+0x0/0x50

```

```

[<c068150f>] ? write_sysrq_trigger+0x3f/0x50
[<c0569ec4>] ? proc_reg_write+0x64/0xa0
[<c0569e60>] ? proc_reg_write+0x0/0xa0
[<c051de50>] ? vfs_write+0xa0/0x190
[<c051e8d1>] ? sys_write+0x41/0x70
[<c0409adc>] ? syscall_call+0x7/0xb
Code: a0 c0 01 0f b6 41 03 19 d2 f7 d2 83 e2 03 83 e0 cf c1 e2 04 09 d0 88
41 03 f3 c3 90 c7 05 c8 1b 9e c0 01 00 00 00 0f ae f8 89 f6 <c6> 05 00 00
00 00 01 c3 89 f6 8d bc 27 00 00 00 00 8d 50 d0 83
EIP: [<c068124f>] sysrq_handle_crash+0xf/0x20 SS:ESP 0068:ef4dbf24
CR2: 0000000000000000

```

Saisir **help log** pour plus d'informations sur l'utilisation de la commande.



Note

La mémoire tampon des messages du noyau incluent les informations principales de plantage-système, et, de ce fait, elles sont toujours déversées dans le fichier **vmcore-dmesg.txt**. Cela est bien utile quand on n'a pas pu obtenir le fichier **vmcore** complet, par exemple, en cas de manque d'espace dans l'emplacement cible. Le fichier par défaut utilisé est **vmcore-dmesg.txt** qui se trouve dans le répertoire **/var/crash/**.

3.4. Afficher un backtrace

Pour afficher un historique de pile d'appels du noyau, saisir la commande **bt** dans l'invite shell. Vous pouvez saisir **bt <pid>** pour afficher l'historique d'appels d'un seul processus.

Exemple 3.3. Ne pas afficher l'interface utilisateur

```

crash> bt
PID: 5591   TASK: f196d560   CPU: 2   COMMAND: "bash"
#0 [ef4dbdcc] crash_kexec at c0494922
#1 [ef4dbe20] oops_end at c080e402
#2 [ef4dbe34] no_context at c043089d
#3 [ef4dbe58] bad_area at c0430b26
#4 [ef4dbe6c] do_page_fault at c080fb9b
#5 [ef4dbee4] error_code (via page_fault) at c080d809
   EAX: 00000063   EBX: 00000063   ECX: c09e1c8c   EDX: 00000000   EBP:
00000000
   DS: 007b      ESI: c0a09ca0   ES: 007b      EDI: 00000286   GS: 00e0
   CS: 0060      EIP: c068124f   ERR: ffffffff   EFLAGS: 00010096
#6 [ef4dbf18] sysrq_handle_crash at c068124f
#7 [ef4dbf24] __handle_sysrq at c0681469
#8 [ef4dbf48] write_sysrq_trigger at c068150a
#9 [ef4dbf54] proc_reg_write at c0569ec2
#10 [ef4dbf74] vfs_write at c051de4e
#11 [ef4dbf94] sys_write at c051e8cc
#12 [ef4dbfb0] system_call at c0409ad5

```

```

EAX: ffffffffda  EBX: 00000001  ECX: b7776000  EDX: 00000002
DS:  007b        ESI: 00000002  ES:  007b      EDI: b7776000
SS:  007b        ESP: bfc2088  EBP: bfc20b4   GS:  0033
CS:  0073        EIP: 00edc416  ERR: 00000004  EFLAGS: 00000246

```

Saisir **help bt** pour plus d'informations sur l'utilisation de la commande.

3.5. Explication du processus

Pour afficher le statut des processus du système, saisir la commande **ps** dans l'invite shell. Vous pouvez saisir **ps<pid>** pour afficher le statut d'un seul processus.

Exemple 3.4. Afficher le statut des processus dans le système

```

crash> ps
  PID   PPID  CPU  TASK      ST  %MEM   VSZ   RSS  COMM
>    0      0   0  c09dc560  RU   0.0     0     0  [swapper]
>    0      0   1  f7072030  RU   0.0     0     0  [swapper]
    0      0   2  f70a3a90  RU   0.0     0     0  [swapper]
>    0      0   3  f70ac560  RU   0.0     0     0  [swapper]
    1      0   1  f705ba90  IN   0.0  2828  1424  init
... several lines omitted ...
 5566     1   1  f2592560  IN   0.0  12876   784  auditd
 5567     1   2  ef427560  IN   0.0  12876   784  auditd
 5587   5132   0  f196d030  IN   0.0  11064  3184  sshd
>  5591   5587   2  f196d560  RU   0.0   5084  1648  bash

```

Saisir **help ps** pour plus d'informations sur l'utilisation de la commande.

3.6. Afficher des informations de mémoire virtuelle

Pour afficher des informations de mémoire virtuelle de base, exécuter la commande **vm** à l'invite de commande. Vous pouvez utiliser **vm <pid>** pour afficher des informations sur un processus unique.

Exemple 3.5. Afficher des informations de mémoire virtuelle sur le contexte en cours

```

crash> vm
PID: 5591  TASK: f196d560  CPU: 2  COMMAND: "bash"
  MM      PGD      RSS  TOTAL_VM
f19b5900  ef9c6000  1648k  5084k
  VMA      START      END  FLAGS  FILE
f1bb0310  242000  260000  8000875  /lib/ld-2.12.so
f26af0b8  260000  261000  8100871  /lib/ld-2.12.so
efbc275c  261000  262000  8100873  /lib/ld-2.12.so
efbc2a18  268000  3ed000  8000075  /lib/libc-2.12.so
efbc23d8  3ed000  3ee000  8000070  /lib/libc-2.12.so
efbc2888  3ee000  3f0000  8100071  /lib/libc-2.12.so
efbc2cd4  3f0000  3f1000  8100073  /lib/libc-2.12.so
efbc243c  3f1000  3f4000  100073

```

```
efbc28ec    3f6000    3f9000 8000075  /lib/libdl-2.12.so
efbc2568    3f9000    3fa000 8100071  /lib/libdl-2.12.so
efbc2f2c    3fa000    3fb000 8100073  /lib/libdl-2.12.so
f26af888    7e6000    7fc000 8000075  /lib/libtinfo.so.5.7
f26aff2c    7fc000    7ff000 8100073  /lib/libtinfo.so.5.7
efbc211c    d83000    d8f000 8000075  /lib/libnss_files-2.12.so
efbc2504    d8f000    d90000 8100071  /lib/libnss_files-2.12.so
efbc2950    d90000    d91000 8100073  /lib/libnss_files-2.12.so
f26afe00    edc000    edd000 4040075
f1bb0a18    8047000   8118000 8001875  /bin/bash
f1bb01e4    8118000   811d000 8101873  /bin/bash
f1bb0c70    811d000   8122000 100073
f26afae0    9fd9000   9ffa000 100073
... several lines omitted ...
```

Saisir **help vm** pour plus d'informations sur l'utilisation de la commande.

3.7. Afficher les fichiers ouverts

Pour afficher des informations sur les fichiers ouverts, saisir la commande **files** dans l'invite shell. Vous pouvez exécuter la commande **files <pid>** pour afficher les fichiers ouverts par un processus unique sélectionné.

Exemple 3.6. Afficher des informations sur les fichiers ouverts dans le contexte actuel

```
crash> files
PID: 5591    TASK: f196d560  CPU: 2    COMMAND: "bash"
ROOT: /      CWD: /root
  FD      FILE      DENTRY      INODE      TYPE      PATH
   0      f734f640   eedc2c6c    eecd6048   CHR       /pts/0
   1      efade5c0   eee14090    f00431d4   REG       /proc/sysrq-trigger
   2      f734f640   eedc2c6c    eecd6048   CHR       /pts/0
  10      f734f640   eedc2c6c    eecd6048   CHR       /pts/0
 255      f734f640   eedc2c6c    eecd6048   CHR       /pts/0
```

Saisir **help files** pour plus d'informations sur l'utilisation de la commande.

3.8. Sortir de l'utilitaire

Pour sortir de l'invite shell et en finir avec l'utilitaire **crash**, saisir **exit** ou **q**.

Exemple 3.7. Sortir de l'utilitaire crash

```
crash> exit
~]#
```


Annexe A. Foire Aux Questions

Q : Quelles méthodes de vidage sont-t-elles disponibles pour les machines virtuelles ?

R : Dans la plupart des cas, la commande **kdump** est suffisante pour obtenir un vidage de mémoire d'une machine après un crash ou en cas de panique. Cela peut être mis en place de la même manière que les installations en bare metal.

Cependant, il faut sans doute travailler directement avec l'hyperviseur pour avoir un vidage de mémoire après incident. Deux mécanismes sont en place avec **libvirt** pour cela : **pvpanic** et **virsh dump**. Ces deux méthodes sont décrites dans le guide [Virtualization Deployment and Administration Guide](#).

Le mécanisme **pvpanic** est décrit dans le guide [Virtualization Deployment and Administration Guide - Setting a Panic Device](#).

La commande **virsh dump** est expliquée dans le guide [Virtualization Deployment and Administration Guide - Creating a Dump File of a Domain's Core](#).

Q : Comment télécharger un fichier de vidage volumineux dans Red Hat Support Services ?

R : Dans certains cas, il pourrait être utile d'envoyer un fichier de vidage de mémoire après incident noyau à Red Hat Global Support Services pour analyse. Toutefois, le fichier de vidage peut être très volumineux, même après avoir été filtré. Étant donné que les fichiers de plus de 250 Mo ne peuvent pas être téléchargés directement via le portail client Red Hat lors de l'ouverture d'un nouveau cas de support, un serveur FTP est fourni par Red Hat pour le téléchargement de fichiers volumineux.

L'adresse du serveur FTP est **dropbox.redhat.com** et les fichiers doivent être téléchargés dans le répertoire **/incoming/**. Votre client FTP doit être défini en mode passif ; si votre parefeu n'autorise pas ce mode, vous pourrez utiliser le serveur **origin-dropbox.redhat.com** en mode actif.

Assurez-vous que les fichiers téléchargés sont compressés à l'aide d'un programme tel que **gzip** et qu'ils sont bien nommés correctement et descriptivement. L'utilisation de votre numéro de dossier de support dans le nom de fichier est recommandé. Après avoir téléchargé avec tous les fichiers utiles, donnez à l'ingénieur responsable de votre cas de support le nom exact du fichier et son checksum SHA1 ou MD5.

Pour obtenir des informations plus précises ou supplémentaires, voir <https://access.redhat.com/site/solutions/2112>.

Q : Combien de temps faut-il pour compléter un vidage de mémoire suite à un incident ?

R : Il est souvent nécessaire, dans un but planification de recouvrement après incident, de savoir combien de temps il faudra pour compléter un vidage de mémoire. Cependant, la durée nécessaire pour compléter un vidage de mémoire suite à incident dépend fortement de la quantité de mémoire copiée sur le disque et de la vitesse des interfaces entre la mémoire RAM et le stockage.

Pour tout test de minutage, le système doit fonctionner sous une charge représentative, sinon les choix d'exclusion de page peuvent présenter un faux aperçu du comportement **kdump** avec un système de production chargé au maximum. Cette différence sera observée plus particulièrement lorsqu'on travaille avec de très grandes quantités de RAM.

Les interfaces de stockage doivent également être considérées dans votre planification lors de l'évaluation du temps de vidage. En raison des contraintes de réseau, un dumping de connexion via **ssh**, par exemple, peut prendre plus de temps qu'un disque SATA connecté localement.

Annexe B. Cibles et configurations kdump prises en charge

B.1. Besoins en mémoire pour kdump

Pour que kdump puisse accéder à un vidage de mémoire suite à incident noyau, et le sauvegarder pour l'analyser ensuite, une partie de la mémoire doit être préservée à cet effet. Le tableau ci-dessous contient une liste des besoins en mémoire minimum pour kdump basés sur l'architecture du système et le total de mémoire physique disponible.

Pour obtenir des informations sur la façon de modifier les paramètres de configuration de la mémoire, voir [Section 2.2.1, « Configuration l'utilisation de la mémoire »](#). Pour obtenir des informations sur la façon de fixer le montant de mémoire réservée dans l'interface utilisateur graphique, consulter [Section 2.3.1, « Configuration l'utilisation de la mémoire »](#).

Tableau B.1. Montant minimum de besoin en mémoire réservée pour kdump

Architecture	Mémoire disponible	Mémoire réservée minimum
AMD64 et Intel 64 (x86_64)	2 Go et plus	160 Mo + 2 bits pour chaque 4 KB de RAM. Pour un système de 1 To de mémoire, 224 Mo est le minimum (160 + 64 Mo).
IBM POWER (ppc64)	2 Go à 4 Go	256 Mo de RAM.
	4 Go à 32 Go	512 Mo de RAM.
	32 Go à 64 Go	1 Go de RAM.
	64 Go à 128 Go	2 Go ou RAM.
	128 Go et plus	4 Go de RAM.
IBM System z (s390x)	2 Go et plus	160 Mo + 2 bits pour chaque 4 KB de RAM. Pour un système de 1 To de mémoire, 224 Mo est le minimum (160 + 64 Mo).

B.2. Limite minimum pour la réservation de mémoire automatique

Sur certains systèmes, il est possible d'allouer la mémoire pour kdump automatiquement, soit en utilisant le paramètre **crashkernel=auto** dans le fichier de configuration du bootloader, soit en activant cette option dans l'utilitaire de configuration graphique. Pour que cette réservation automatique fonctionne, cependant, une certaine quantité de mémoire totale doit être disponible dans le système. Ce montant diffère selon l'architecture du système.

Le tableau ci-dessous répertorie les limites d'allocation automatique de la mémoire. Si le système a moins de mémoire que spécifié dans le tableau, la mémoire devra être réservée manuellement.

Pour obtenir plus d'informations sur la façon de modifier ces paramètres de configuration, consulter [Section 2.2.1, « Configuration l'utilisation de la mémoire »](#). Pour obtenir des instructions sur la façon de modifier le montant de mémoire réservée dans l'interface utilisateur graphique, consulter [Section 2.3.1, « Configuration l'utilisation de la mémoire »](#).

Tableau B.2. Montant minimum de mémoire requise pour une réservation de mémoire automatique

Architecture	Mémoire requise
AMD64 et Intel 64 (x86_64)	2 Go
IBM POWER (ppc64)	2 Go
IBM System z (s390x)	4 Go

B.3. Cibles kdump prises en charge

Quand on capture un plantage de noyau, la mémoire d'incident noyau peut être écrite directement sur un périphérique, stockée dans un fichier sur un système local, ou envoyée sur le réseau. Le tableau ci-dessous contient une liste complète des mémoires de vidage cibles actuellement prises en charge ou non par kdump.

Pour obtenir plus d'informations sur la façon de configurer la cible en ligne de commande, consulter [Section 2.2.2, « Configuration du type de kdump »](#). Pour obtenir des informations sur la façon de procéder en interface graphique utilisateur, consulter [Section 2.3.2, « Configuration du type de kdump »](#).

Tableau B.3. Cibles kdump prises en charge

Type	Cibles prises en charge	Cibles non prises en charge
Périphérique brut	Toutes les partitions et tous les disques bruts attachés localement.	—
Systèmes de fichiers locaux	Les systèmes de fichiers ext2 , ext3 , ext4 , btrfs et xfs sur disques durs attachés directement, sur lecteurs logiques RAID hardware, périphériques LVM, et disques mdraid .	Tout système de fichiers local non listé explicitement, supporté dans ce tableau, y compris le type auto (détection de système de fichiers automatique)
Répertoire personnel	Répertoires à distance accédés via protocoles NFS ou SSH sur IPv4 .	Répertoires à distance sur le système de fichiers rootfs auxquels on a accédé en utilisant le protocole NFS .
	Répertoires à distance auxquels on a accédé via le protocole iSCSI par l'intermédiaire d'initiateurs logiciels et matériels à la fois.	Répertoires à distance auxquels on a accédé via le protocole iSCSI sur matériel be2iscsi .
	Stockages basés multipath	—
	—	Répertoires à distance auxquels on a accédé via le protocole IPv6 .
	—	Répertoires à distance auxquels on a accédé via protocoles SMB/CIFS .
	—	Répertoires à distance auxquels on a accédé via le protocole FCoE (<i>Fibre Channel over Ethernet</i>).
	—	Répertoires à distance auxquels on a accédé par des interfaces de réseau wireless.

B.4. Niveaux de filtrage kdump pris en charge

Pour réduire la taille de la mémoire de vidage, kdump utilise le core collector **makedumpfile** pour compresser les données et parfois délaisse des informations importantes. Le tableau ci-dessous contient une liste complète des niveaux de filtrage actuellement pris en charge par l'utilitaire **makedumpfile**.

Pour obtenir plus d'informations sur la façon de configurer le core collector en ligne de commande, consulter [Section 2.2.3, « Configuration d'un client DHCP »](#). Pour obtenir des informations sur la façon de procéder en interface graphique utilisateur, consulter [Section 2.3.3, « Configuration d'un client DHCP »](#).

Tableau B.4. Niveaux de filtrage pris en charge

Option	Description
1	Pages zéro
2	Pages de cache
4	Cache privé
8	Pages utilisateur
16	Pages libres



Note

La commande **makedumpfile** supporte la suppression de pages huges et hugetlbfs transparentes dans Red Hat Enterprise Linux 7.3 et versions supérieures. Ces deux types de pages huge sont considérées comme des pages utilisateur et seront supprimées par le niveau **-8**.

B.5. Actions par défaut prises en charge

Par défaut, lorsque kdump ne parvient pas à créer un vidage de base, il monte le système de fichiers racine (root) et tente d'enregistrer le noyau localement. Vous pouvez toutefois configurer kdump pour effectuer une opération différente au cas où il ne parvienne pas à enregistrer le vidage de base vers la cible principale. Le tableau ci-dessous répertorie toutes les actions par défaut actuellement prises en charge par kdump.

Pour obtenir plus d'informations sur la façon de configurer l'action par défaut en ligne de commande, consulter [Section 2.2.4, « Configurer une action par défaut »](#). Pour obtenir des informations sur la façon de procéder en interface graphique utilisateur, consulter [Section 2.3.4, « Configurer une action par défaut »](#).

Tableau B.5. Actions par défaut prises en charge

Option	Description
dump_to_rootfs	Essaie d'enregistrer la mémoire de vidage dans le système de fichiers racine. Cette option est particulièrement utile en combinaison avec une cible réseau : si la cible réseau est inaccessible, cette option configure kdump pour enregistrer le vidage de la mémoire du noyau localement. Le système est redémarré par la suite.
reboot	Redémarre les systèmes, en perdant la mémoire de vidage au cours du processus.
halt	Arrête le système, en perdant la mémoire de vidage au cours du processus.
poweroff	Éteint le système, en perdant la mémoire de vidage au cours du processus.
shell	Exécute une session shell dans initramfs, autorisant l'utilisateur à enregistrer la mémoire de vidage manuellement.

B.6. Estimation de la taille de Kdump

Lors de la planification et de la construction de votre environnement **kdump**, il est nécessaire de savoir combien d'espace est nécessaire pour le fichier de mémoire de vidage avant qu'on l'ait produite. La commande **makedumpfile** peut vous y aider.

L'option **-mem-usage** fournit un rapport utile sur les pages qui peuvent être exclues, et qui peut être utilisé pour déterminer le niveau de mémoire de vidage que vous souhaitez affecter. Cette commande doit être exécutée lorsque le système est sous charge représentative, sinon **makedumpfile** renverra une valeur inférieure à celle qui est attendue dans votre environnement de production.

```
[root@hostname ~]# makedumpfile --mem-usage /proc/kcore
```

TYPE	PAGES	EXCLUDABLE	DESCRIPTION
-----	-----	-----	-----
ZERO zero	501635	yes	Pages filled with
CACHE	51657	yes	Cache pages
CACHE_PRIVATE private	5442	yes	Cache pages +
USER	16301	yes	User process pages
FREE	77738211	yes	Free pages
KERN_DATA data	1333192	no	Dumpable kernel



Important

La commande **makedumpfile** produit un rapport en **pages**. Cela signifie que vous devez calculer la taille de la mémoire utilisée par rapport à la taille de la page du noyau, ce qui, dans Red Hat Enterprise Linux kernel, correspond à 4 kilooctets (4096 octets).

Annexe C. Historique des versions

Version 1.3-2.1	Mon Jul 3 2017	Terry Chuang
Fichiers de traduction synchronisés avec les sources XML 1.3-2		
Version 1.3-2	Fri Nov 4 2016	Mark Flitter
Version pour la distribution GA 7.3.		
Version 1.2-9	Thu 18 Aug 2016	Mark Flitter
Mises à jour 7.3 Beta, notes spécifiques pour Z Series et estimation de la taille des vmcores.		
Version 1.2-0	Fri 06 Mar 2015	Petr Bokoč
Mise à jour pour corriger plusieurs problèmes comme des informations erronées sur la configuration de la mémoire et la mise à jour d'instantanés obsolètes.		
Version 1.1-3	Wed 18 Feb 2015	Petr Bokoč
Version Red Hat Enterprise Linux 7.1 GA du Guide de vidage de la mémoire suite à un incident noyau		
Version 1.1-0	Fri 05 Dec 2014	Petr Bokoč
Version Red Hat Enterprise Linux 7.1 Beta du Guide de vidage de la mémoire suite à un incident noyau		
Version 1.0-0	Mon 02 Jun 2014	Jaromír Hradílek
Version Red Hat Enterprise Linux 7.0 GA du Guide de vidage de la mémoire suite à un incident noyau		
Version 0.0-8	Thu Jan 17 2013	Jaromír Hradílek
Création initiale de l'ouvrage.		