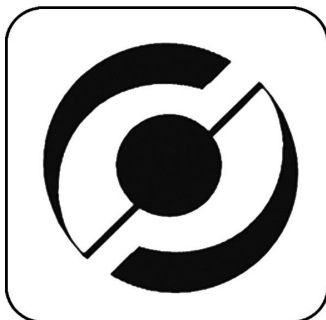


به نام خدا



Multipath in PVM

محدوده:

در این مستند نحوه راه‌اندازی *Multipath* در *PVM* مورد بررسی قرار می‌گیرد.

تاریخچه:

ردیف	نویسنده	تاریخ	شماره ویرایش	توضیحات
۱	تحقیق و توسعه فنی و مهندسی	۹۱/۰۸/۲۳	۰/۱/۰	
۲	تحقیق و توسعه فنی و مهندسی	۹۱/۱۲/۱۳	۱/۰/۰	

فهرست مندرجات

۳	- چکیده
۳	- کلید واژه ها
۳	۱- <i>Multipath</i> چیست؟
۳	۲- نصب <i>Multipath</i> در <i>PVM 6</i>
۴	۲.۱- یک نکته در راهاندازی اولیه
۵	۳- دیوایسهای <i>multipath</i>
۵	۴- اجزای <i>dm-multipath</i>
۶	۵- دستور <i>multipath</i> و انجام تغییرات در حین کار بدون <i>restart</i>
۶	۶- نکات
۷	۷- فایل تنظیمات <i>etc/multipath.conf/</i>
۷	۷.۱- <i>Blacklist</i> و <i>blacklist_exceptions</i>
۱۵	۸- منابع

چکیده

در این مستند نحوه راه اندازی *multipath* در *PVM* مورد بررسی قرار میگیرد.

کلید واژه ها

Multipath, pvm

۱ *Multipath* چیست؟

افزونگی در مسیرهای اتصال داده سرورهای به *SAN Storage* و تشخیصی این افزونگی و مدیریت *fail* شدن مسیرها به نام *multipath* شناخته میشود. انواع روشها برای این موضوع وجود دارد همچون، مسیرهای مختلف به یک کنترلر، مسیرهای مختلف به دو کنترلر.

۲ نصب *Multipath* در *PVM 6*

۱. نصب *rpm*ها

```
rpm -i device-mapper-multipath-0.4.9-41.el6.x86_64.rpm device-mapper-multipath-libs-0.4.9-41.el6.x86_64.rpm kpartx-0.4.9-41.el6.x86_64.rpm
```

۲. ساخت فایل کانفیگ *multipath* (مسیر فایل: */etc/multipath.conf*)

```
rm /etc/multipath.conf # جهت توضیحات قسمت ۲/۱ را مشاهده کنید  
mpathconf --enable --user_friendly_names n --with_module y
```

Example: multipath.conf == یک نمونه ساده که به صورت پیش فرض جوابگو است

```
defaults {  
    user_friendly_names no  
}  
  
blacklist {  
  
}
```

۳. تنظیم شروع خودکار سرویس *multipathd* و شروع آن

```
chkconfig multipathd on  
service multipathd start
```

۴. اطمینان از وجود کار کردن درست *multipath*

```
multipath -l
```

این دستور توپولوژی وضعیت *multipath* را نشان خواهد داد.

۲.۱ یک نکته در راه اندازی اولیه

در راه اندازی هایی که انجام شد مشخص شد که *anaconda* در ابتدای راه افتادن سیستم یک فایل کانفیگ میسازد که بر اساس این فایل کانفیگ دستور *multipath -l* هیچگونه خروبی ندارد.
در نتیجه توصیه بند ۲ از بخش ۲ توصیه شده است که ابتدا فایل */etc/multipath.conf* حذف گردد.

۳ دیوایسهای *multipath*

دیوایسها با *WWID* یا *World Wide Identifier* شناخته میشوند که در تمام کلاستر این عدد برای هر دیوایس یکتا میباشد.

این دیوایسها در مسیر */dev/mapper/* قرار دارند.

دستور *l-multipath* به شما نشان خواهد داد که هر دیوایس شامل چه دیسکهایی است.

چنانچه *user_friendly_names* فعال باشد، دیوایسهایی با نام *mpath** ایجاد میشوند که به جای *WWID*

میتوانند مورد استفاده قرار گیرند و در مسیر */dev/mapper/* قرار دارند.

لیست *WWID*ها همچنین در فایل */etc/multipath/wwids/* وجود دارد که توسط *multipathd* و *multipath*

ایجاد و مدیریت میشود.

۴ اجزای *dm-multipath*

Component	Description
<i>dm_multipath</i> kernel module	Reroutes I/O and supports failover for paths and path groups.
<i>mpathconf</i> utility	<p>Configures and enables device mapper multipathing.</p> <p>این دستور فایل کانفیگ را چنانچه وجود دارد تغییر میدهد و اگر نیست میسازد پارامترهای این دستور:</p> <p><i>mpathconf -h</i></p> <p>چنانچه این دستور به صورت خالی در خط فرمان وارد شود وضعیت کانفیگهای را نشان میدهد:</p> <p><i>mpathconf</i></p> <p>طبق گفته مستند منبع، تنظیمات پیش فرض به صورت کامپایل شده در <i>multipath</i> وجود دارد و چنانچه نیاز به انجام تغییرات در مورد تنظیمات پیش فرض بود باید فایل <i>multipath.conf</i> را تغییر داد.</p> <p>ساده ترین محتوای فایل <i>multipath.conf</i> به این شرح است:</p> <pre>defaults { user_friendly_names no }</pre>

Component	Description
	<code>blacklist { }</code>
<code>multipath command</code>	<i>Lists and configures multipath devices. Normally started up with <code>/etc/rc.sysinit</code>, it can also be started up by a <code>udev</code> program whenever a block device is added or it can be run by the <code>initramfs</code> file system.</i>
<code>multipathd daemon</code>	<i>Monitors paths; as paths fail and come back, it may initiate path group switches. Provides for interactive changes to multipath devices. This must be restarted for any changes to the <code>/etc/multipath.conf</code> file.</i>
<code>kpartx command</code>	<i>Creates device mapper devices for the partitions on a device It is necessary to use this command for DOS-based partitions with DM-MP. The <code>kpartx</code> is provided in its own package, but the <code>device-mapper-multipath</code> package depends on it.</i>

۵ دستور `multipath` و انجام تغییرات در حین کار بدون `restart`

تجربه کار نشان داد که در صورت `user_friendly_names` ابتدا `yes` باشد و بعد `no` شود و `multipathd` هم `reload` شود نامها `user_friendly` حذف نمی شوند.

همچنین `stop` کردن `multipathd` موجود حذف شد دیوایسهای که ساخته شده است نمی شود.

برای حذف دیوایسها و ساخت مجدد آن از رویش زیر استفاده کنید:

```
service multipathd stop
```

تمام دیوایسها را حذف میکند `# multipath -F`

```
service multipathd start
```

از این شیوه برای تغییراتی که بعد از انجام در فایل کانفیگ و `reload` اعمال نمیشوند استفاده کنید.

دقت داشته باشید که در هنگام استفاده از دستور `multipath -F` سیستم در وضعیت عملیاتی نباشد.

۶ نکات مهم

۱. هرگونه تغییر در فایل کانفیگ `/etc/multipath.conf/` میبایست سرویس `multipathd` راه‌اندازی مجدد یا `reload` گردد.
۲. دستور `multipathd -k` کنسول تعاملی با سرویس `multipathd` را در اختیار قرار میدهد.

۷ فایل تنظیمات `/etc/multipath.conf/`

این فایل از ۵ قسمت اصل تشکیل میشود:

blacklist

Listing of specific devices that will not be considered for multipath.

blacklist_exceptions

Listing of multipath candidates that would otherwise be blacklisted according to the parameters of the blacklist section.

defaults

General default settings for DM-Multipath.

multipaths

Settings for the characteristics of individual multipath devices. These values overwrite what is specified in the defaults and devices sections of the configuration file.

devices

Settings for the individual storage controllers. These values overwrite what is specified in the defaults section of the configuration file. If you are using a storage array that is not supported by default, you may need to create a devices subsection for your array.

۷

۷.۱ *Blacklist* و *blacklist_exceptions*

نمونه‌هایی از نحوه بلاک کردن به شرح زیر آمده است:

```
blacklist {
    wwid 26353900f02796769
    devnode "^sd[a-z]"
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*" devnode "^hd[a-z]"
    device {
        vendor "IBM"
        product "3S42"          #DS4200 Product 10
    }
    device {
        vendor "HP"
        product "*"
    }
}
```

قابل ذکر است که *Exepstion* هم به همین طریق قابل انجام است.

قسمت *defaults*

این قسمت مقادیر پیش فرض را برای تمامی *multipath*ها در خود دارد. نمونه‌ای از مقادیر که میتوانند ذکر

شوند و توضیح مقدار آنها به شرح زیر است:

```
#defaults {
#    udev_dir                /dev
#    polling_interval        5
#    path_selector            "round-robin 0"
#    path_grouping_policy    failover
#    getuid_callout           "/lib/dev/scsi_id --whitelisted --device=/dev/%n"
#    prio                     const
#    path_checker             directio
#    rr_min_io                1000
#    rr_weight                uniform
#    failback                 manual
#    no_path_retry            fail
#    user_friendly_names      no
#}
```


Description	Attribute
<p>Specifies the interval between two path checks in seconds. For properly functioning paths, the interval between checks will gradually increase to (4 * <code>polling_interval</code>). The default value is 5</p>	<p><code>polling_interval</code></p>
<p>The directory where <code>udev</code> device nodes are created. The default value is <code>/dev</code></p>	<p><code>udev_dir</code></p>
<p>The directory where the dynamic shared objects are stored. The default value is system dependent, commonly <code>/lib/multipath</code></p>	<p><code>multipath_dir</code></p>
<p>Defines the mode for setting up multipath devices. If this parameter is set to yes, then multipath will not try to create a device for every non-blacklisted path. Instead multipath will create a device only if one of three conditions are met</p> <p>There are at least two non-blacklisted paths with the same WWID</p> <p>The user manually forces the creation of the device by specifying a device with the <code>multipath</code> command</p> <p>A path has the same WWID as a multipath device that was previously created. Whenever a multipath device is created with <code>find_multipaths</code> set, multipath remembers the WWID of the device so that it will automatically create the device again as soon as it sees a path with that WWID. This allows you to have multipath automatically choose the correct paths to make into multipath devices, without having to edit the multipath blacklist. For instructions on the procedure to follow if you have previously created multipath devices when the <code>find_multipaths</code> parameter was not set, see Section 4.2. ”Configuration File Blacklist”</p> <p>The default value is no</p>	<p><code>find_multipaths</code></p>
<p>The default verbosity. Higher values increase the verbosity level. Valid levels are between 0 and 6. The</p>	<p><code>verbosity</code></p>

Description	Attribute
<p><i>default value is 2</i></p> <p><i>Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include</i></p> <p><i>round-robin 0: Loop through every path in the path group, sending the same amount of I/O to each</i></p> <p><i>queue-length 0: Send the next bunch of I/O down the path with the least number of outstanding I/O requests</i></p> <p><i>service-time 0: Send the next bunch of I/O down the path with the shortest estimated service time, which is determined by dividing the total size of the outstanding I/O to each path by its relative throughput</i></p> <p><i>The default value is round-robin 0</i></p>	<p><i>path_selector</i></p>
<p><i>Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include</i></p> <p><i>.failover: 1 path per priority group</i></p> <p><i>multibus: all valid paths in 1 priority group</i></p> <p><i>group_by_serial: 1 priority group per detected serial number</i></p> <p><i>group_by_prio: 1 priority group per path priority value. Priorities are determined by callout programs specified as global, per-controller, or per-multipath options</i></p> <p><i>group_by_node_name: 1 priority group per target node name. Target node names are fetched in /sys/class/fc_transport/target*/node_name</i></p> <p><i>The default value is failover</i></p>	<p><i>path_grouping_policy</i></p>
<p><i>Specifies the default program and arguments to call out</i></p>	<p><i>getuid_callout</i></p>

Description	Attribute
<p>to obtain a unique path identifier. An absolute path is required</p> <p>The default value is <code>/lib/udev/scsi_id</code> <code>--whitelisted --device=/dev/%n</code></p>	
<p>Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable <code>prio</code> value. Possible values include</p> <p><code>.const</code>: Set a priority of 1 to all paths</p> <p><code>.emc</code>: Generate the path priority for EMC arrays</p> <p><code>alua</code>: Generate the path priority based on the SCSI-3 ALUA settings</p> <p><code>tpg_pref</code>: Generate the path priority based on the SCSI-3 ALUA settings, using the preferred port bit</p> <p><code>.ontap</code>: Generate the path priority for NetApp arrays</p> <p><code>rdac</code>: Generate the path priority for LSI/Engenio RDAC controller</p> <p><code>hp_sw</code>: Generate the path priority for Compaq/HP controller in active/standby mode.</p> <p><code>hds</code>: Generate the path priority for Hitachi HDS Modular storage arrays. The default value is <code>const</code>.</p>	<p><code>prio</code></p>
<p>The default extra features of multipath devices. The only existing feature is <code>queue_if_no_path</code>, which is the same as setting <code>no_path_retry</code> to <code>queue</code>. For information on issues that may arise when using this feature, see Section 5.6, "Issues with <code>queue_if_no_path</code> feature</p>	<p><code>features</code></p>
<p>Specifies the default method used to determine the state of the paths. Possible values include:</p> <p><code>readsector0</code>: Read the first sector of the device.</p>	<p><code>path_checker</code></p>

Description	Attribute
<p><i>tur</i>: Issue a TEST UNIT READY to the device.</p> <p><i>emc_clariion</i>: Query the EMC Clariion specific EVPD page 0xC0 to determine the path.</p> <p><i>hp_sw</i>: Check the path state for HP storage arrays with Active/Standby firmware.</p> <p><i>rdac</i>: Check the path stat for LSI/Engenio RDAC storage controller.</p> <p><i>directio</i>: Read the first sector with direct I/O. The default value is directio.</p>	
<p><i>Manages path group failback</i></p> <p>A value of immediate specifies immediate failback to the highest priority path group that contains active paths</p> <p>A value of manual specifies that there should not be immediate failback but that failback can happen only with operator intervention</p> <p>A value of followover specifies that automatic failback should be performed when the first path of a path group becomes active. This keeps a node from automatically failing back when another node requested the failover</p> <p>A numeric value greater than zero specifies deferred failback, expressed in seconds</p> <p>The default value is manual</p>	<p>failback</p>
<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. This setting is only for systems running kernels older than 2.6.31. Newer systems should use <i>rr_min_io_rq</i>. The default value is 1000</p>	<p>rr_min_io</p>
<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath. This setting should be used on systems running current kernels. On systems running kernels older than 2.6.31, use <i>rr_min_io</i>. The default value is 1</p>	<p>rr_min_io_rq</p>

Description	Attribute
<p>If set to priorities, then instead of sending rr_min_io requests to a path before calling path_selector to choose the next path, the number of requests to send is determined by rr_min_io times the path's priority, as determined by the prio function. If set to uniform, all path weights are equal. The default value is uniform</p>	<p>rr_weight</p>
<p>A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queueing</p> <p>A value of fail indicates immediate failure, without queueing</p> <p>A value of queue indicates that queueing should not stop until the path is fixed</p> <p>The default value is 0</p>	<p>no_path_retry</p>
<p>If set to yes, specifies that the system should use the /etc/multipath/bindings file to assign a persistent and unique alias to the multipath, in the form of mpathn. If set to no, specifies that the system should use the WWID as the alias for the multipath. In either case, what is specified here will be overridden by any device-specific aliases you specify in the multipaths section of the configuration file. The default value is no</p>	<p>user_friendly_names</p>
<p>If set to no, the multipathd daemon will disable queueing for all devices when it is shut down. The default value is no</p>	<p>queue_without_daemon</p>
<p>If set to yes, the multipathd daemon will disable queueing when the last path to a device has been deleted. The default value is no</p>	<p>flush_on_last_del</p>
<p>Sets the maximum number of open file descriptors that can be opened by multipath and the multipathd daemon. This is equivalent to the ulimit -n command. As of the Red Hat Enterprise Linux 6.3 release, the default value is max, which sets this to the system limit from /proc/sys/fs/nr_open. For earlier releases, if this is not set the maximum number</p>	<p>max_fds</p>

Description	Attribute
<p>of open file descriptors is taken from the calling process; it is usually 1024. To be safe, this should be set to the maximum number of paths plus 32, if that number is greater than 1024</p>	
<p>The timeout to use for path checkers that issue SCSI commands with an explicit timeout, in seconds. The default value is taken from <code>sys/block/sdx/device/timeout</code></p>	<p><code>checker_timeout</code></p>
<p>The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before failing I/O to devices on that remote port. This value should be smaller than the value of <code>dev_loss_tmo</code>. Setting this to <code>off</code> will disable the timeout. The default value is determined by the OS</p>	<p><code>fast_io_fail_tmo</code></p>
<p>The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before removing it from the system. Setting this to infinity will set this to 2147483647 seconds, or 68 years. The default value is determined by the OS</p>	<p><code>dev_loss_tmo</code></p>
<p>Red Hat Enterprise Linux Release 6.3 and later)) Controls how multipath integrates the device configurations from the <code>devices</code> section of the configuration file with the built-in device configurations</p> <p>Each device configuration in the <code>devices</code> section of the <code>multipath.conf</code> file will either create its own device configuration or it will modify one of the built-in device configurations. Prior to Red Hat Enterprise Linux 6.3, if the vendor, product, and revision strings in a user's device configuration exactly matched those strings in a built-in device configuration, the built-in configuration was modified by the options in the user's configuration. Otherwise, the user's device configuration was treated as a new configuration</p> <p>If <code>hwtable_regex_match</code> is set to <code>yes</code>, a regular expression match is used instead. The vendor, product, and revision strings are all regular expressions. The user device configuration values for these options are matched against the built-in device configuration values. This match works the same way</p>	<p><code>hwtable_regex_match</code></p>

Description	Attribute
<p>that an actual device's vendor product and revision strings are matched against a device configuration's strings to see which configuration should be used for the device. If the user's device configuration matches, then the built-in configuration is modified by the options in the user's configuration. Otherwise the user's device configuration is treated as a new configuration</p>	
<p>Red Hat Enterprise Linux Release 6.4 and later) If this parameter is set to yes and the scsi layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the mutipath.conf file. If the scsi layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual. The default value is no</p>	<p>retain_attached_hw_handler</p>
<p>Red Hat Enterprise Linux Release 6.4 and later) If this is set to yes, multipath will first check if the device supports ALUA, and if so it will automatically assign the device the alua prioritizer. If the device doesn't support ALUA, it will determine the prioritizer as it always does. The default value is no</p>	<p>detect_prio</p>

۷.۲ قسمت **multipath**

مشخصات ویژه هر مسیر **multipath** را در اینجا ذکر میشود همچون **alias** یک نمونه از این مورد به شرح زیر

میباشد.

```

multipaths {
    multipath {
        wwid                3600508b4000156d70001200000b0000
        alias                yellow
        path_grouping_policy multibus
        path_selector        "round-robin 0"
        failback              manual
        rr_weight             priorities
        no_path_retry        5
    }
}

```

```

}
multipath {
    wwid                1DEC_____321816758474
    alias               red
    rr_weight           priorities
}
}

```

Description	Attribute
Specifies the WWID of the multipath device to which the multipath attributes apply. This parameter is mandatory for this section of the multipath.conf file	wwid
Specifies the symbolic name for the multipath device to which the multipath attributes apply. If you are using user_friendly_names, do not set this value to mpathn; this may conflict with an automatically assigned user friendly name and give you incorrect device node names	alias
Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include failover = 1 path per priority group multibus = all valid paths in 1 priority group group_by_serial = 1 priority group per detected serial number group_by_prio = 1 priority group per path priority value group_by_node_name = 1 priority group per target node name	path_grouping_policy
Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include round-robin 0: Loop through every path in the path group, sending the same amount of I/O to each queue-length 0: Send the next bunch of I/O down the path with the least number of outstanding I/O requests service-time 0: Send the next bunch of I/O down the path with the shortest estimated service time, which is	path_selector

Description	Attribute
<p>determined by dividing the total size of the outstanding I/O to each path by its relative throughput</p>	
<p><i>Manages path group failback</i></p> <p>A value of immediate specifies immediate failback to the highest priority path group that contains active paths</p> <p>A value of manual specifies that there should not be immediate failback but that failback can happen only with operator intervention</p> <p>A value of followover specifies that automatic failback should be performed when the first path of a path group becomes active. This keeps a node from automatically failing back when another node requested the failover</p> <p>A numeric value greater than zero specifies deferred failback, expressed in seconds</p>	<p>failback</p>
<p>Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable prio value. Possible values include</p> <p>.const: Set a priority of 1 to all paths</p> <p>.emc: Generate the path priority for EMC arrays</p> <p>alua: Generate the path priority based on the SCSI-3 ALUA settings</p> <p>tpg_pref: Generate the path priority based on the SCSI-3 ALUA settings, using the preferred port bit</p> <p>.ontap: Generate the path priority for NetApp arrays</p> <p>rdac: Generate the path priority for LSI/Engenio RDAC controller</p> <p>hp_sw: Generate the path priority for Compaq/HP controller in active/standby mode</p> <p>hds: Generate the path priority for Hitachi HDS Modular storage arrays</p>	<p>prio</p>

Description	Attribute
<p>A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling <code>queueing</code></p> <p>A value of <code>fail</code> indicates immediate failure, without <code>queueing</code></p> <p>A value of <code>queue</code> indicates that <code>queueing</code> should not stop until the path is fixed</p>	<p><code>no_path_retry</code></p>
<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. This setting is only for systems running kernels older than 2.6.31. Newer systems should use <code>rr_min_io_rq</code>. The default value is 1000</p>	<p><code>rr_min_io</code></p>
<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath. This setting should be used on systems running current kernels. On systems running kernels older than 2.6.31, use <code>rr_min_io</code>. The default value is 1</p>	<p><code>rr_min_io_rq</code></p>
<p>If set to <code>priorities</code>, then instead of sending <code>rr_min_io</code> requests to a path before calling <code>path_selector</code> to choose the next path, the number of requests to send is determined by <code>rr_min_io</code> times the path's priority, as determined by the <code>prio</code> function. If set to <code>uniform</code>, all path weights are equal</p>	<p><code>rr_weight</code></p>
<p>If set to <code>yes</code>, then multipath will disable <code>queueing</code> when the last path to a device has been deleted</p>	<p><code>flush_on_last_del</code></p>
<p>If set to <code>yes</code>, specifies that the system should use the <code>/etc/multipath/bindings</code> file to assign a persistent and unique alias to the multipath, in the form of <code>mpathn</code>. If set to <code>no</code>, specifies that the system should use the WWID as the alias for the multipath. In either case, what is specified here will be overridden by any device-specific aliases you specify in the <code>multipaths</code> section of the configuration file</p>	<p><code>user_friendly_names</code></p>

۷.۳ قسمت devices

این قسمت بر اساس `Product ID` و `Vendor ID` تنظیمات مورد نظر را اعمال میکند.

این تنظیمات، قسمت *defaults* را *overwrite* میکند ولی توسط *multipaths* برای مسیری که این دیوایس عضو آن است *overwrite* میشود.

```
#devices {
#   device {
#       vendor          "COMPAQ  "
#       product         "MSA1000  "
#       path_grouping_policy multibus
#       path_checker    tur
#       rr_weight       priorities
#   }
#}
```

Description	Attribute
<i>Specifies the vendor name of the storage device to which the device attributes apply, for example COMPAQ</i>	<i>vendor</i>
<i>Specifies the product name of the storage device to which the device attributes apply, for example HSV110 (C) COMPAQ</i>	<i>product</i>
<i>Specifies the product revision identifier of the storage device</i>	<i>revision</i>
<i>Specifies a regular expression used to blacklist devices by product</i>	<i>product_blacklist</i>
<i>Specifies a module that will be used to perform hardware specific actions when switching path groups or handling I/O errors. Possible values include:</i> <ul style="list-style-type: none"> 1 <i>emc</i>: hardware handler for EMC storage arrays. 1 <i>alua</i>: hardware handler for SCSI-3 ALUA arrays. 1 <i>hp_sw</i>: hardware handler for Compaq/HP controllers. 1 <i>rdac</i>: hardware handler for the LSI/Engenio RDAC controllers. 	<i>hardware_handler</i>
<i>Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include</i> <ul style="list-style-type: none"> <i>failover</i> = 1 path per priority group <i>multibus</i> = all valid paths in 1 priority group <i>group_by_serial</i> = 1 priority group per detected serial 	<i>path_grouping_policy</i>

Description	Attribute
<p><i>number</i></p> <p><i>group_by_prio = 1 priority group per path priority value</i></p> <p><i>group_by_node_name = 1 priority group per target node name</i></p>	
<p><i>Specifies the default program and arguments to call out to obtain a unique path identifier. An absolute path is required</i></p>	<p>getuid_callout</p>
<p><i>Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include</i></p> <p><i>round-robin 0: Loop through every path in the path group, sending the same amount of I/O to each</i></p> <p><i>queue-length 0: Send the next bunch of I/O down the path with the least number of outstanding I/O requests</i></p> <p><i>service-time 0: Send the next bunch of I/O down the path with the shortest estimated service time, which is determined by dividing the total size of the outstanding I/O to each path by its relative throughput</i></p>	<p>path_selector</p>
<p><i>Specifies the default method used to determine the state of the paths. Possible values include</i></p> <p><i>.readsector0: Read the first sector of the device</i></p> <p><i>.tur: Issue a TEST UNIT READY to the device</i></p> <p><i>emc_clariion: Query the EMC Clariion specific EVPD page 0xC0 to determine the path</i></p> <p><i>hp_sw: Check the path state for HP storage arrays with Active/Standby firmware</i></p> <p><i>rdac: Check the path stat for LSI/Engenio RDAC storage controller</i></p> <p><i>directio: Read the first sector with direct I/O</i></p>	<p>path_checker</p>
<p><i>The extra features of multipath devices. The only existing feature is queue_if_no_path, which is the same as setting</i></p>	<p>features</p>

Description	Attribute
<p><i>no_path_retry</i> to queue. For information on issues that may arise when using this feature, see Section 5.6, “Issues with <i>queue_if_no_path</i> feature”</p>	
<p>Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable <i>prio</i> value. Possible values include</p> <p><i>.const</i>: Set a priority of 1 to all paths</p> <p><i>.emc</i>: Generate the path priority for EMC arrays</p> <p><i>alua</i>: Generate the path priority based on the SCSI-3 ALUA settings</p> <p><i>tpg_pref</i>: Generate the path priority based on the SCSI-3 ALUA settings, using the preferred port bit</p> <p><i>.ontap</i>: Generate the path priority for NetApp arrays</p> <p><i>rdac</i>: Generate the path priority for LSI/Engenio RDAC controller</p> <p><i>hp_sw</i>: Generate the path priority for Compaq/HP controller in active/standby mode</p> <p><i>hds</i>: Generate the path priority for Hitachi HDS Modular storage arrays</p>	<p><i>prio</i></p>
<p><i>Manages path group failback</i></p> <p>A value of <i>immediate</i> specifies immediate failback to the highest priority path group that contains active paths</p> <p>A value of <i>manual</i> specifies that there should not be immediate failback but that failback can happen only with operator intervention</p> <p>A value of <i>followover</i> specifies that automatic failback should be performed when the first path of a path group becomes active. This keeps a node from automatically failing back when another node requested the failover</p> <p>A numeric value greater than zero specifies deferred failback, expressed in seconds</p>	<p><i>failback</i></p>

Description	Attribute
<p>If set to priorities, then instead of sending rr_min_io requests to a path before calling path_selector to choose the next path, the number of requests to send is determined by rr_min_io times the path's priority, as determined by the prio function. If set to uniform, all path weights are equal</p>	<p>rr_weight</p>
<p>A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queueing</p> <p>A value of fail indicates immediate failure, without queueing</p> <p>A value of queue indicates that queueing should not stop until the path is fixed</p>	<p>no_path_retry</p>
<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. This setting is only for systems running kernels older than 2.6.31. Newer systems should use rr_min_io_rq. The default value is 1000</p>	<p>rr_min_io</p>
<p>Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath. This setting should be used on systems running current kernels. On systems running kernels older than 2.6.31, use rr_min_io. The default value is 1</p>	<p>rr_min_io_rq</p>
<p>The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before failing I/O to devices on that remote port. This value should be smaller than the value of dev_loss_tmo. Setting this to off will disable the timeout</p>	<p>fast_io_fail_tmo</p>
<p>The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before removing it from the system. Setting this to infinity will set this to 2147483647 seconds, or .68 years</p>	<p>dev_loss_tmo</p>
<p>If set to yes, the multipathd daemon will disable queueing when the last path to a device has been deleted</p>	<p>flush_on_last_del</p>
<p>If set to yes, specifies that the system should use the /etc/multipath/bindings file to assign a persistent and unique alias to the multipath, in the form of mpathn. If set to no, specifies that the system should use the WWID as the alias for</p>	<p>user_friendly_names</p>

Description	Attribute
<p><i>the multipath. In either case, what is specified here will be overridden by any device-specific aliases you specify in the multipaths section of the configuration file. The default value is no</i></p>	
<p><i>Red Hat Enterprise Linux Release 6.4 and later) If this parameter is set to yes and the scsi layer has already attached a hardware handler to the path device, multipath will not force the device to use the hardware_handler specified by the mutipath.conf file. If the scsi layer has not attached a hardware handler, multipath will continue to use its configured hardware handler as usual</i></p>	<p>retain_attached_hw_handler</p>
<p><i>Red Hat Enterprise Linux Release 6.4 and later) If this is set to yes, multipath will first check if the device supports ALUA, and if so it will automatically assign the device the alua prioritizer. If the device doesn't support ALUA, it will determine the prioritizer as it always does</i></p>	<p>detect_prio</p>

۸ منابع

[1] https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html-single/DM_Multipath/index.html