



DYNAMIC PAGE RETIREMENT

vR331 | April 2014



TABLE OF CONTENTS

- Chapter 1. Overview..... 1
- Chapter 2. Implementation..... 2
- Chapter 3. Availability..... 3
- Chapter 4. Visibility.....4
 - 4.1. XIDs.....4
 - 4.2. NVML..... 4
 - 4.3. Nvidia-smi..... 5
- Chapter 5. Caveats.....7
- Chapter 6. FAQ..... 8

Chapter 1.

OVERVIEW

The NVIDIA® driver supports "retiring" of bad framebuffer memory cells, by retiring the page the cell belongs to. This is called "dynamic page retirement" and is done automatically for cells that are degrading in quality. This feature can improve the longevity of an otherwise good board and is thus an important resiliency feature on supported products, especially in HPC and enterprise environments.

Chapter 2. IMPLEMENTATION

The marking of a page for exclusion is called "blacklisting", while the actual act of excluding that page from subsequent memory allocations is called "retirement". The NVIDIA[®] driver will blacklist a page once it has experienced a single Double Bit ECC Error (DBE) or 2 Single Bit ECC Errors (SBE). These addresses are stored in the InfoROM. When the driver loads it will retrieve these addresses from the InfoROM, then have the framebuffer manager set these pages aside, such that they cannot be used by the driver or user applications.

Ideally, the NVIDIA[®] driver will catch weakening cells at the 2 SBE point and retire the page, before the cell degrades to the point of a DBE and disrupts an application.

A retired page will be stored in the InfoROM for persistence for the life of the board. However, the driver will need to be reloaded after a page has been blacklisted for the retirement to take effect. See [Caveats](#) for more information.

Chapter 3.

AVAILABILITY

Dynamic page retirement is supported on the following products and environments:

- ▶ Drivers: R319 and higher
- ▶ OSes: All standard driver-supported Linux and Windows TCC platforms with ECC enabled
- ▶ GPUs:
 - ▶ Tesla K20 and higher products
 - ▶ No Quadro or Geforce products are currently supported

Chapter 4.

VISIBILITY

Three main mechanisms provide visibility into page retirement: XID errors in system logs, the NVML API and the `nvidia-smi` command line tool.

4.1. XIDs

XID errors are driver errors that are logged to the system error log. Please see the XID Whitepaper for general info on XIDs.

There are three main XIDs related to dynamic page retirement:

- ▶ XID 48: A DBE has occurred.
- ▶ XID 63: A page has successfully been retired.
- ▶ XID 64: A page has failed retirement due to an error.

In the system log these XIDs show up in the following forms:

- ▶ XID 48: "XID 48 An uncorrectable double bit error (DBE) has been detected on GPU (<id>)"
- ▶ XID 63: "XID 63 Dynamic Page Retirement: New retired page, reload the driver to activate. (<address>)"
- ▶ XID 64: "XID 64 Dynamic Page Retirement: Fatal error, unable to retire page (<address>)"

4.2. NVML

The NVIDIA[®] Management Library (NVML) is a public C-based library for GPU monitoring and management. It includes APIs that report the status and count of retired pages. Please see the NVML API docs for general info on the library.

The set of currently retired pages, and their addresses, can be retrieved using:

```
nvmlReturn_t nvmlDeviceGetRetiredPages (nvmlDevice_t device,  
nvmlPageRetirementCause_t cause, unsigned int* pageCount, unsigned long long*  
addresses)
```

Where cause is one of
 NVML_PAGE_RETIREMENT_CAUSE_MULTIPLE_SINGLE_BIT_ECC_ERRORS or
 NVML_PAGE_RETIREMENT_CAUSE_DOUBLE_BIT_ECC_ERROR

The current state of the driver (whether any pages are pending retirement) can be retrieved using:

```
nvmlReturn_t nvmlDeviceGetRetiredPagesPendingStatus (nvmlDevice_t device,
  nvmlEnableState_t* isPending)
```

4.3. Nvidia-smi

Nvidia-smi is a public command line interface for GPU monitoring and management. It implements most of the NVML APIs and supports reporting the status and count of retired pages. Please see the Nvidia-smi man page for general info on the tool.

To view the number of retired pages and the page retirement state of the driver in human readable form:

```
$ nvidia-smi -i <target gpu> -q -d PAGE_RETIREMENT
...
Retired pages
      Single Bit ECC           : 2
      Double Bit ECC          : 0
      Pending                  : No
...
```

If pages have been retired the affected addresses can be viewed through nvidia-smi's scriptable outputs, either XML:

```
$ nvidia-smi -i <target gpu> -q -x
...
      <retired_pages>
        <multiple_single_bit_retirement>
          <retired_count>2</retired_count>
          <retired_page_addresses>
            <retired_page_address>0xABC123</
retired_page_address>
            <retired_page_address>0xDEF456</
retired_page_address>
          </retired_page_addresses>
        </multiple_single_bit_retirement>
        <double_bit_retirement>
          <retired_count>0</retired_count>
          <retired_page_addresses></
retired_page_addresses>
        </double_bit_retirement>
        <pending_retirement>No</pending_retirement>
      </retired_pages>
...
```

or CSV:

```
$ nvidia-smi -i <target gpu> --query-retired-
pages=gpu_uuid,retired_pages.address,retired_pages.cause --format=csv
```

```
...  
gpu_uuid, retired_pages.address, retired_pages.cause  
GPU-d73c8888-9482-7d65-c95c-4b58c7d9eb4c, 0xABC123, Double Bit ECC  
GPU-d73c8888-9482-7d65-c95c-4b58c7d9eb4c, 0xDEF456, Double Bit ECC  
GPU-d73c8888-9482-7d65-c95c-4b58c7d9eb4c, 0x123ABC, Single Bit ECC  
...
```

Chapter 5.

CAVEATS

There are currently several page retirement behaviors that should be noted:

- ▶ As mentioned above, the driver requires a reload before a page is actively retired. The page will be written to the InfoROM, but cannot be retired until the driver is reloaded. The driver can be reloaded in multiple ways:
 - ▶ reboot the system
 - ▶ temporarily exit persistence mode and all driver applications (X, NVML, etc)
 - ▶ reset the GPU
 - ▶ unload and reload the kernel module
- ▶ There exists a race condition between logging errors to the InfoROM and ending a CUDA™ job while in persistence mode. This race condition is most often hit when shutting down in response to a DBE. The effect of this condition is that a page may fail to retire in certain corner cases.

Exiting persistence mode before rebooting the system will forcibly flush any pending writes to the InfoROM. If XID 48 is seen and XID 63 is not seen, it is recommended to exit persistence mode via the command:

```
% nvidia-smi -i <target GPU> -pm 0
```

At this point, the XID 63 should be seen and the NVML query can be used to verify the page was written to the InfoROM.

Chapter 6.

FAQ

6.1. Is available memory reduced by retired pages?

Each retired page decreases the total memory available to applications. However, the total maximum size of memory retired in this way is only on the order of 256 KiB. This is insignificant relative to other factors, such as natural fluctuations in memory allocated internally by the NVIDIA driver during normal operation.

6.2. Can page retirement be disabled?

No, page retirement is an important reliability feature and cannot be disabled. Any pages already marked as retired will continue to be excluded in all future allocations. Note though that if ECC is disabled no new memory errors will be detected and thus no new pages will be blacklisted for future retirement.

6.3. Is application behavior affected?

No, applications behave the same. Since pages are retired only after the driver has been restarted the act of retiring a page occurs outside the lifetime of any GPU process or application. An application running on a GPU with pages scheduled for retirement (blacklisted) will continue to see those pages in its memory space, though any page retired due to a double bit error (DBE) will necessarily cause an application to terminate. This is true even without page retirement.

6.4. Is application performance affected?

No, application performance is unaffected by either the blacklisting of pages or their subsequent retirement. Blacklisting is the only act taken during application execution,

while the actual retirement event happens only after the application has terminated. As noted in the first FAQ question above, the memory impact of retired pages is also negligible.

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2013-2014 NVIDIA Corporation. All rights reserved.