

Linux for zSeries and S/390



How to Improve Performance with PAV January 30, 2004

Linux Kernel 2.4 (June 2003 stream)

Linux for zSeries and S/390



How to Improve Performance with PAV January 30, 2004

Linux Kernel 2.4 (June 2003 stream)

Note

Before using this information and the product it supports, read the information in "Notices" on page 7.

First Edition (January 2004)

This edition applies to Linux kernel 2.4 (June 2003 stream) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

How to improve performance with PAV	1	Notices	7
Enabling volumes for PAV	2	Trademarks	8
Configuring base and alias PAV with LVM	2		

How to improve performance with PAV

The concurrent operations capabilities of the IBM® TotalStorage® Enterprise Storage Server® (ESS) support concurrent data transfer operations to or from the same volume from the same system or system image. A volume that can be accessed in this way is called a Parallel Access Volume (PAV).

The operating system does not attempt to start more than one I/O operation at a time to a device, but today's storage subsystem design, with large caches and RAID 5 arrays, makes it possible for the storage control unit to do I/Os in parallel. When software is using PAV, it can issue multiple channel programs to a volume, allowing simultaneous access to the logical volume by multiple users or processes. Reads can be satisfied simultaneously, as well as writes to different domains. The domain of an I/O consists of the specified extents to which the I/O operation applies. Writes to the same domain still have to be serialized to maintain data integrity.

Prerequisites: Linux on a zSeries® or S/390® mainframe can use PAV if all of the following apply:

- Linux runs as a z/VM® guest.
- The volume resides on an IBM TotalStorage Enterprise Storage Server (ESS).
- Linux uses the logical volume manager (LVM). If you want to set up volumes for PAV, you should be familiar with LVM.

To make use of PAV you need a patched version of LVM1 (or an equivalent multipathing solution). The LVM1 patch can be found on developerWorks®:

http://www10.software.ibm.com/developerworks/opensource/linux390/useful_add-ons_lvm.shtml

Restrictions:

- You cannot make PAV enabled volumes accessible to more than one Linux instance at a time.
- To ensure data integrity Linux must use LVM. By default, Linux interprets each path as leading to a separate volume. LVM allows Linux to recognize where multiple paths lead to the same volume.

Setting up an ESS disk volume for PAV includes these tasks:

1. Configuring the volume on the ESS. The volume must be configured as a base device with at least one alias device. There is no separate real disk space associated with alias devices.

ESS configuration is beyond the scope of this document. Refer to *IBM TotalStorage Enterprise Storage Server Web Interface User's Guide, SC26-7448* for details.

2. Defining the volume to the zSeries or S/390 hardware. See "Enabling volumes for PAV" on page 2.
3. Configuring paths to the volume on Linux. See "Configuring base and alias PAV with LVM" on page 2.

Enabling volumes for PAV

This section describes how you must define ESS volumes to your hardware so that Linux can use them for PAV.

Prerequisites:

- You need to know the device numbers of the base devices and their aliases as defined on the ESS.
- You need privilege class B authorization on z/VM.

Perform the following steps to define the base devices and their aliases to the hardware. In the examples, we assume that device X'5680' is a base device and X'56BF' an alias device for the same physical disk space on the ESS.

1. Define the base devices to the hardware. In an IOCDs IODEVICE statement, use UNIT=3390B.

Example: The following statement defines device number X'5680' as a base device.

```
IODEVICE ADDRESS=(5680),UNITADD=00,CUNUMBR=(5680),          *
          STADET=Y,UNIT=3390B
```

2. Define the alias devices to the hardware. In an IOCDs IODEVICE statement, use UNIT=3390A.

Example: The following statement defines device X'56BF' as an alias device. The mapping to the associated base device X'5680' is in the ESS configuration.

```
IODEVICE ADDRESS=(56BF),UNITADD=18,CUNUMBR=(5680),          *
          STADET=Y,UNIT=3390A
```

3. After the hardware configuration with the base and alias device statements has become active, use z/VM to check the mapping of base and alias devices. Issue CP QUERY PAV.

Example: The output contains lines like this:

```
00: Device 5680 is a base Parallel Access Volume with the following aliases: 56BF
00: Device 56BF is an alias Parallel Access Volume device whose base device is 5680
```

4. From z/VM, use CP ATTACH commands to make base devices and their aliases accessible to the Linux guest.

Example: To make a base device X'5680' and its alias X'56BF' available to a guest with ID "LNX1" issue:

```
ATTACH 5680 LNX1
ATTACH 56BF LNX1
```

You can now configure the devices in Linux.

Configuring base and alias PAV with LVM

This section describes how to define a PAV base device and its aliases as a single logical volume.

Prerequisites:

- You must know the device numbers of the PAV base device and its aliases.
- You need root authorization on the Linux system

From the IPLed Linux guest, perform the following steps:

1. Ensure that the devices are ready for use.

- a. Issue `cat /proc/dasd/devices` to ensure that device nodes exist for your disk devices.

Nodes are created automatically if you have started Linux with the “`dasd=`” kernel parameter. If there are no device nodes, create them dynamically. For information on how to create device nodes see the DASD chapter of *Linux for zSeries and S/390 Device Drivers and Installation Commands*. You can find the latest version at:

http://www10.software.ibm.com/developerworks/opensource/linux390/june2003_documentation.shtml

Example:

```
# cat /proc/dasd/devices
5680(ECKD) at ( 94: 8) is dasdc      : active at blocksize: 4096, 1803060 blocks, 7043 MB
56bf(ECKD) at ( 94: 12) is dasdd    : active at blocksize: 4096, 1803060 blocks, 7043 MB
```

- b. Ensure that the device is formatted. If it is not already formatted, use `dasdfmt` to format it. Because a base device and its aliases all correspond to the same physical disk space, formatting either the base device or one of its aliases formats the base device and all alias devices.

Example:

```
dasdfmt -f /dev/dasdc
```

- c. Ensure that the device is partitioned. If it is not already partitioned, use `fdasd` to create one or more partitions. Because a base device and its aliases all correspond to the same physical disk space, partitioning either the base device or one of its aliases creates partitions for the base device and all alias devices.

Example: The following command creates both a partition `/dev/dasdc1` for the base device and also a partition `/dev/dasdd1` for the alias.

```
fdasd -a /dev/dasdc
```

You now have PAV enabled devices for which multiple subchannels are configured. You can display the subchannels for a particular PAV enabled device by issuing a command like this:

```
cat /proc/subchannels | egrep "<devno base device>|<devno alias1>|<devno alias2>| ..."
```

Example: For a base device X'5680' and alias X'56BF' the command and its output might look like this:

```
# cat /proc/subchannels | egrep "5680|56BF"
5680 0030 3390/0C 3990/E9 yes FC FC FF C6C7C8CA CBC90000
56BF 0031 3390/0C 3990/E9 yes FC FC FF C6C7C8CA CBC90000
```

In the example:

- The base device X'5680' maps to device node `dasdc` and can be accessed through subchannel X'0030'.
- The alias device X'56BF' maps to device node `dasdd` and can be accessed through subchannel X'0031'.

2. Issue `vgscan` to check for existing volume groups and to create LVM configuration data.

Example: If no volume group has been defined yet, the output might be:

```
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of your volume group
```

3. For each base device, create a new LVM physical volume. Issue a command like this:

```
pvcreeate /dev/<volume name>
```

Example:

```
# pvcreate /dev/dasdc1
pvcreate -- physical volume "/dev/dasdc1" successfully created
```

4. Create a new volume group. Issue a command like this:

```
vgcreate <group name> <physical volume>
```

Example:

```
# vgcreate vg_kb /dev/dasdc1
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "vg_kb"
vgcreate -- volume group "vg_kb" successfully created and activated
```

5. Display details about the new volume group. Issue a command like this:

```
vgdisplay -v <volume group>
```

Example:

```
# vgdisplay -v vg_kb
--- Volume group ---
VG Name                vg_kb
VG Access              read/write
VG Status              available/resizable
VG #                   0
MAX LV                 256
Cur LV                0
Open LV               0
MAX LV Size           255.99 GB
Max PV                256
Cur PV               1
Act PV                1
VG Size               6.87 GB
PE Size               4 MB
Total PE              1759
Alloc PE / Size      0 / 0
Free PE / Size       1759 / 6.87 GB
VG UUID               3nwJYn-SxW1-gKym-OvZs-TYIf-CrHP-in05Yp
```

```
--- No logical volumes defined in "vg_kb" ---
```

6. From the volume group, create a new logical volume. Issue commands like this:

```
lvcreate --name <volume name> --extents <no of extents> <group name>
```

Example: In this example, a logical volume `lv_kb` is created that uses all 1759 available extents of the volume group `vg_kb`.

```
# lvcreate --name lv_kb --extents 1759 vg_kb
lvcreate -- doing automatic backup of "vg_kb"
lvcreate -- logical volume "/dev/vg_kb/lv_kb" successfully created
```

LVM is aware that `dasdd` is an alias for `dasdc`. You can confirm this by issuing `cat /proc/lvm/global:`.

Example:

```
# cat /proc/lvm/global:
LVM module LVM version 1.0.5(mp-v6)(15/07/2002)
```

```
Total: 1 VG 1 PV 1 LV (0 LVs open)
Global: 32300 bytes malloced IOP version: 10 3:18:35 active
```

```
VG: vg_kb [1 PV, 1 LV/0 open] PE Size: 4096 KB
Usage [KB/PE]: 7204864 /1759 total 7204864 /1759 used 0 /0 free
PV: [AA] dasdc1 7204864 /1759 7204864 /1759 0 /0
    +-- dasdd1
LV: [AWDL ] lv_kb 7204864 /1759 close
```

The output shows that the logical volume lv_kb corresponds to both partition dasdc1 and its alias dasdd1.

You can also issue lvscan: to get an overview of your logical volumes.

Example:

```
# lvscan:
lvscan -- ACTIVE          "/dev/vg_kb/lv_kb" [6.87 GB]
lvscan -- 1 logical volumes with 6.87 GB total in 1 volume group
lvscan -- 1 active logical volumes
```

7. Issue # pvpath -qa to display the path information. The command output shows that there is an enabled path that corresponds to the to the base device, and that there are one or more disabled paths that correspond to the aliases.

Example:

```
# pvpath -qa
Physical volume /dev/dasdc1 of vg_kb has 2 paths:
      Device  Weight Failed Pending State
#  0:  94:9      0      0      0 enabled
#  1:  94:13     0      0      0 disabled
```

8. Enable the other paths. Issue commands like this:

```
pvpath -<number of disabled path> -ey <physical device>
```

Example:

```
# pvpath -p1 -ey /dev/dasdc1
vg_kb: setting state of path #1 of PV#1 to enabled
```

9. Issue # pvpath -qa to confirm that all paths are enabled.

Example:

```
# pvpath -qa
Physical volume /dev/dasdc1 of vg_kb has 2 paths:
      Device  Weight Failed Pending State
#  0:  94:9      0      0      0 enabled
#  1:  94:13     0      0      0 enabled
```

Now LVM is ready to use multiple paths to the PAV volumes.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

developerWorks
Enterprise Storage Server
IBM
S/390
TotalStorage
z/VM
zSeries

Other company, product, and service names may be trademarks or service marks of others.



LNUX-HTPA-00

