



IBM and Linux: Community Innovation for your Business

Hyper PAV & Large Volume Support for Linux on System z

Susanne Wintenberger (swinten@de.ibm.com)
IBM Lab Boeblingen, Germany



Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market. Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business (logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area. All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the Performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

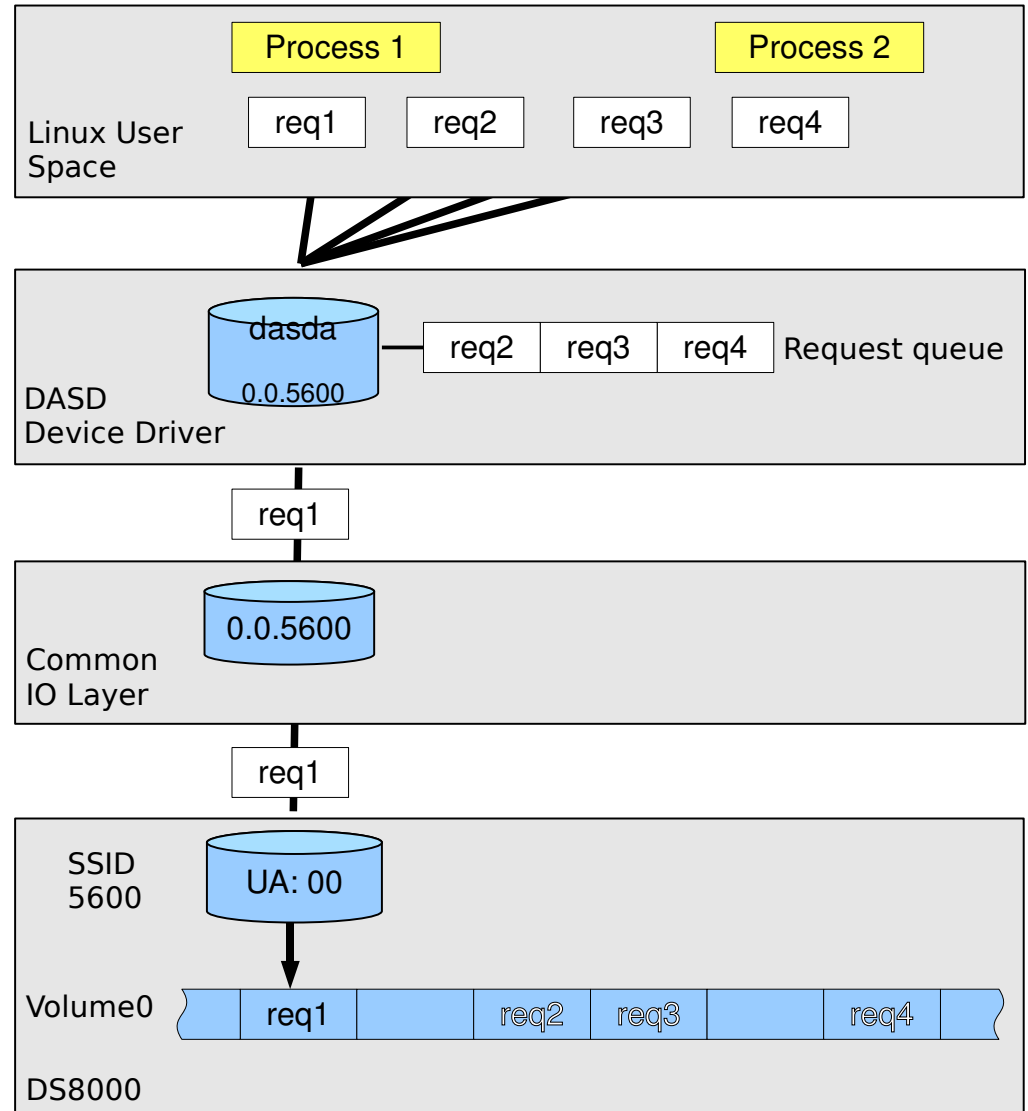
Agenda

- * PAV/ HyperPAV Support
 - Motivation for PAV
 - PAV and Hyper PAV concepts
 - Device mapper based implementation
 - Configuring PAV volumes with multipath tools
 - DASD device driver based implementation
 - Hyper PAV
 - Tool support
 - Hints and Tips
- * Large Volume Support
 - Large Volumes
 - Compatibility

Problem Statement:
Linux on System z customers need to
address more data with good performance
and high availability!

Normal DASD I/O – Why do we need PAV?

- * One subchannel can execute one I/O request at a time.
- * Programs running in parallel often access independent areas of one volume.
- * Storage server could address independent areas in parallel (cache, striping, etc. on storage server).
- * Sending requests in parallel would improve performance.
- * Note: The SSID identifies the logical control unit (LCU) on the storage server.

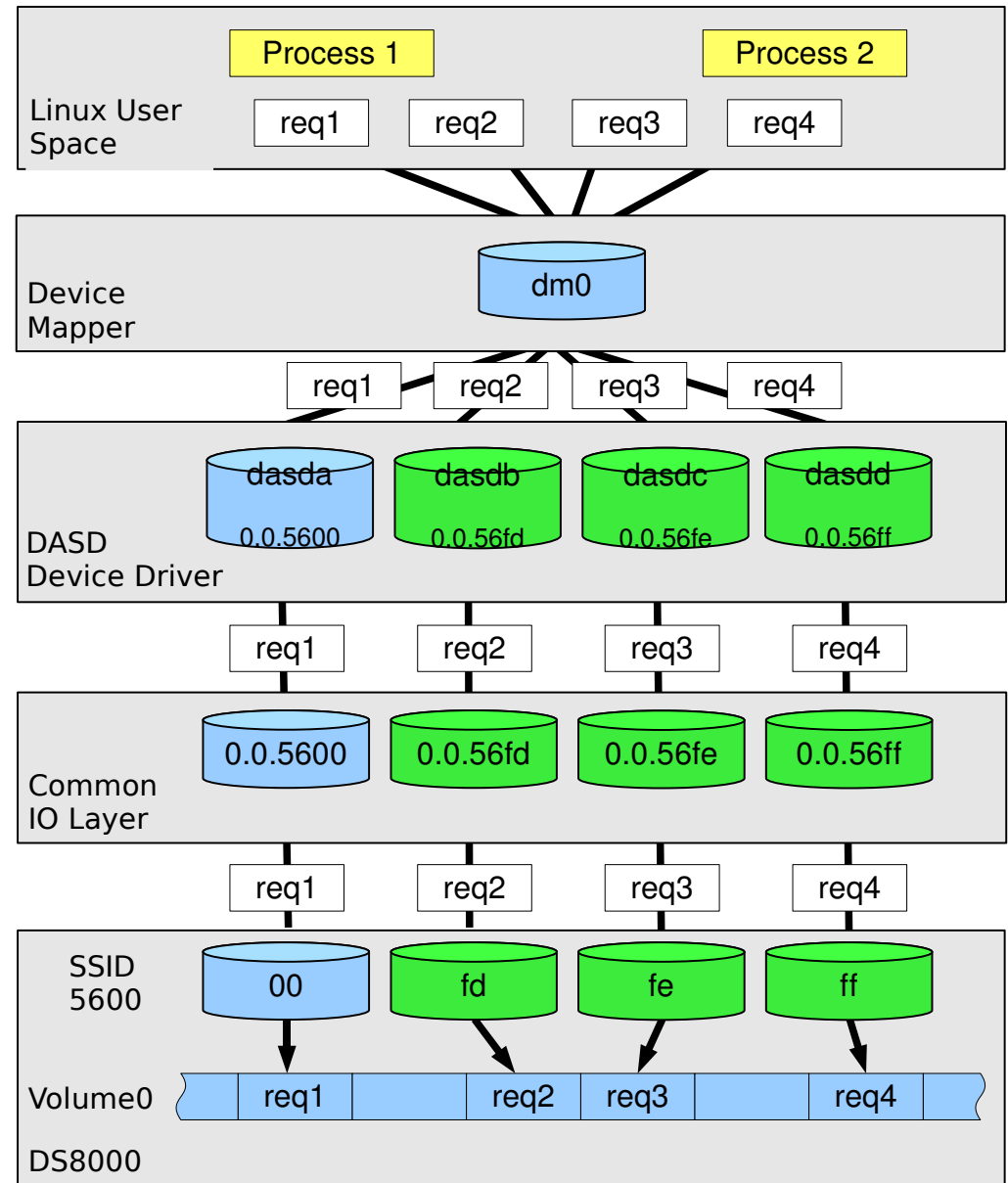


Parallel Access Volumes - Concepts

- * Still one I/O per subchannel, but
- * Several subchannels per volume.
- * One PAV base device per volume.
- * Several additional PAV alias devices.
- * Base PAV
 - I/O on an alias will be directed to fixed base volume.
- * Hyper PAV
 - I/O on alias may be directed to any base volume in the same logical control unit (LCU)
 - Backward compatible: Will work like Base PAV if the Linux version does not support Hyper PAV

Original device mapper based implementation

- * Used on SLES10 and RHEL4 & 5
- * Through the PAV feature, storage systems can present the same physical disk space as a **base device** and one or more **alias devices**.
- * The DASD device driver initially senses the base device
- * Each DASD base device and each alias device has a separate device node assigned
- * Example: If device 5600 is a base device and devices 56fd, 56fe and 56ff are alias devices, all device are available at the common IO layer and device nodes dasda to dasdd will be created
- * This design relies on the fixed base/alias association.
- * Finally device-mapper is required to combine the device dasda to dasdd to a single multipath device dm0
- * It does not support Hyper PAV!



Configuring PAV volumes with multipath tools

- * Issue `lsdasd` to ensure that device nodes exist for the PAV base volume and its aliases and that the devices are online.
- * Use `chccwdev` to set the DASD online, if needed.
- * 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`
- * Ensure that the device is partitioned. If it is not already partitioned, use `fdasd` to create one or more partitions. The following command creates both a partition `/dev/dasdc1` for the base device and also a partition `/dev/dasdd1` for the alias.
 - Example: `fdasd -a /dev/dasdc`
- * Set the base device and all its aliases offline and back online to assure that the device driver detects the partitions for each device name.
 - Example: `chccwdev -d 0.0.5600,0.0.56ff && chccwdev -e 0.0.5600,0.0.56ff`

Configuring PAV volumes with multipath tools (cont'd)

- * If it is not already loaded, load the `dm_multipath` module: (e.g: `modprobe dm_multipath`)
- * Make sure that your `multipath.conf` configuration file does **not** contain blacklist-entries for dasd devices

```
# cat /etc/multipath.conf
...
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z][[0-9]*]"
    devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
    devnode "^dasd[a-z]+[0-9]*"
}
...
```

- * Use the `multipath` command to detect multiple paths to devices for failover or performance reasons and coalesce them
- * Make sure `multipathd` is started (e.g: `/etc/init.d/multipathd start`)
- * Enter `multipath -ll` to display the resulting multipath configuration.

Configuring PAV volumes with multipath tools (cont'd)

```
# multipath -ll
IBM.75000000092461.2a00.1b dm-1 IBM,S/390 DASD ECKD
[size=2.3G][features=0][hwandler=0]
\_ round-robin 0 [prio=4][enabled]
  \_ 0:0:10779:0 dasde 94:16 [active][ready]
  \_ 0:0:10924:0 dasdf 94:20 [active][ready]
  \_ 0:0:10925:0 dasdg 94:24 [active][ready]
  \_ 0:0:10926:0 dasdh 94:28 [active][ready]
IBM.75000000092461.2a00.1a dm-0 IBM,S/390 DASD ECKD
[size=2.3G][features=0][hwandler=0]
\_ round-robin 0 [prio=4][enabled]
  \_ 0:0:10778:0 dasdc 94:12 [active][ready]
  \_ 0:0:10927:0 dasdd 94:32 [active][ready]
```

- * The DASDs can now be accessed as multipath devices
IBM.75000000092461.2a00.1a and IBM.75000000092461.2a00.1b.
- * You can find the corresponding device nodes in /dev/mapper.
 /dev/mapper/IBM.75000000092461.2a00.1a
 /dev/mapper/IBM.75000000092461.2a00.1ap1
 /dev/mapper/IBM.75000000092461.2a00.1b
 /dev/mapper/IBM.75000000092461.2a00.1bp1
 /dev/mapper/control
- * There is a device node for each multipath device and for each partition on these multipath devices.

You can now use LVM2 or an equivalent logical volume manager to configure the multipath device into a volume group, for example, for striping. If you use LVM2 to work with multipath devices, set a filter to ensure only the multipath devices are used and not the underlying base and aliases.

z/VM PAV setup

- * If your Linux system runs as a z/VM guest operating system, you can confirm the mapping of base and alias devices.
- * After the hardware configuration with the base and alias device statements has become active, enter the z/VM QUERY PAV command

```
#CP QUERY PAV
00: Device 5600 is a base Parallel Access Volume with the
    following aliases: 56FF
00: Device 56FF is an alias Parallel Access Volume device whose
    base device is 5600
CP Q PAV 4DE1
00: Device 4DE1 is not a Parallel Access Volume
```

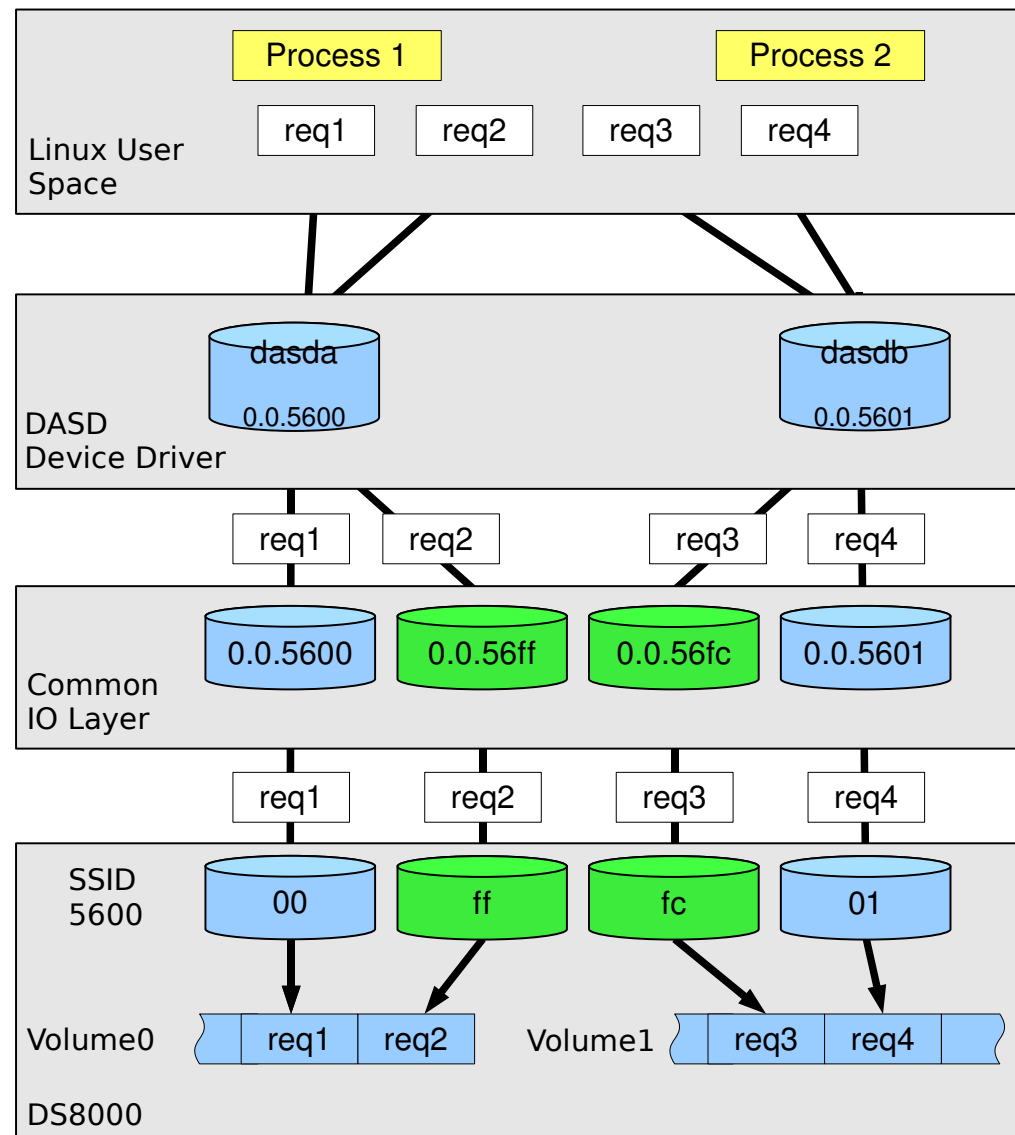
- * To make a base device 0x5600 and its alias 0x56FF available to the current z/VM guest, enter the following commands:

```
#CP ATTACH 5600 *
#CP ATTACH 56FF *
```

New DASD device driver implementation

Base PAV example

- * Used on SLES11 and upcoming distributions
- * Through the PAV feature, storage systems can present the same physical disk space as a **base device** and one or more **alias devices**.
- * The DASD device driver initially senses the base device.
- * The DASD device driver creates device nodes for the base devices but not for the aliases.
- * The base device is set online.
- * The aliases can lead to gaps in the naming scheme for device nodes.
- * If multiple user space processes concurrently access a base device, the device driver uses the aliases to issue multiple channel programs.
- * *Apart from assuring that the corresponding aliases for a base device are online, user space processes need no special handling for accessing a PAV.*



PAV Prerequisites

- * Before you can use PAV on your Linux instance, the PAV feature must be enabled on your storage system.
- * The PAV feature is available, for example, for the following systems:
 - IBM System Storage DS8000 series systems
 - IBM System Storage DS6000 series systems
 - IBM TotalStorage Enterprise Storage Server (ESS)
- * The HyperPAV feature is available, for example, for IBM System Storage DS8000 series systems.
- * PAV base and alias volumes require special IOCDs specifications
- * You need to know the device numbers of the base devices and their aliases as defined on the storage system.
- * If your Linux system runs as a z/VM guest operating system, you need privilege class B authorization.

IOCDs Configuration

- * Configuring base and alias volumes for PAV or HyperPAV on the storage system is beyond the scope of this presentation.
 - See your storage system documentation for details.
 - For information about IOCDs specifications for multiple subchannel sets see the Input/Output Configuration Program User's Guide for your mainframe system.
- * The IOCDs examples in this presentation apply to mainframe systems with a single subchannel set.
- * Perform the following steps to define the base devices and their aliases to the hardware:
 - Define the base devices to the storage hardware
 - Define the alias devices to the storage hardware

- * Example: The following statement defines device number 0x5600 as a base device.

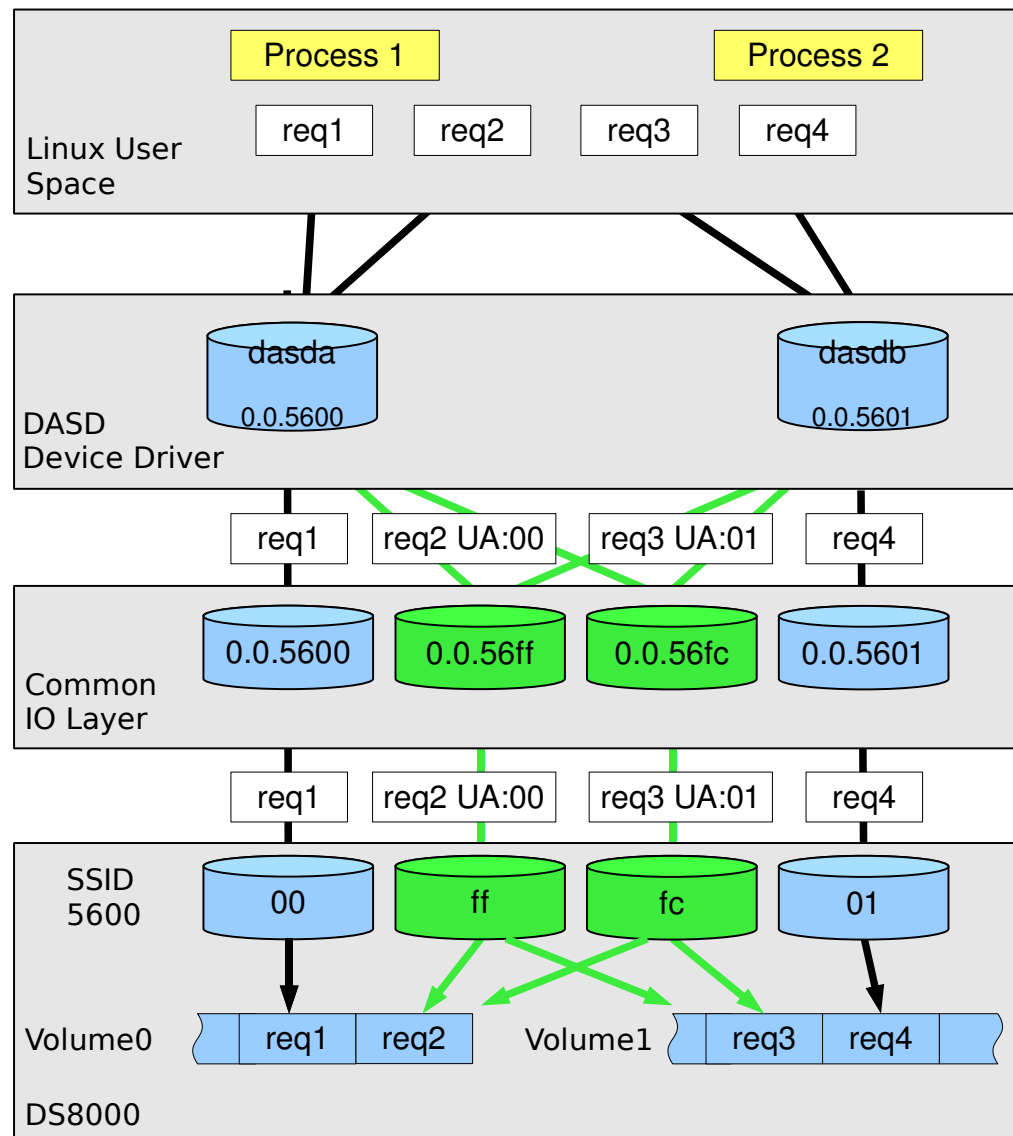
```
IODEVICE ADDRESS=(5600),UNITADD=00,  
CUNUMBR=(5600), *  
STADET=Y,UNIT=3390B
```

- * Example: The following statement defines device 0x56ff as an alias device. The mapping to the associated base device 0x5600 is given by the storage system configuration.

```
IODEVICE  
ADDRESS=(56FF),UNITADD=FF,  
CUNUMBR=(5600), *  
STADET=Y,UNIT=3390A
```

New DASD device driver implementation Hyper PAV example

- * With the old base-PAV support the mapping between base and alias devices was static.
- * Now HyperPAV removes the requirement to dedicate alias devices to specific base devices.
- * Alias devices are used with all base devices of the same LCU
- * Target unit address is encoded into the request itself
- * Same user interface as Base PAV
- * HyperPAV is activated automatically when the necessary prerequisites are there (DS8000 with HyperPAV LIC, z/VM 5.3)
- * SLES11
- * If the prerequisites for HyperPAV are not there, base-PAV is used if the PAV feature is enabled on the storage server. Otherwise the DASD driver works without using PAV.



HyperPAV Setup

HyperPAV simplifies systems management and improves performance using an on demand I/O model

- * HyperPAV Base and Alias subchannels are defined on control unit's Hardware Management Console and in IOCDs no differently than traditional PAVs
 - * HyperPAV hardware, priced feature enables floating Alias function associated with the HyperPAV architecture for each LSS (logical control unit)
 - * Operating system host determines which LCU (logical control unit) is in HyperPAV vs. traditional PAV mode
- (1) Setup PAV configuration on Storage Server
 - (2) zSeries storage configuration (IOCP)
 - (3) Basic DASD configuration
 - (4) That's it – nothing else to do. *No multipath configuration needed. No formatting / partitioning related pitfalls!*

PAV/HyperPAV Toolbox

New sysfs interfaces can allow to identify devices:

```
$ ls -al /sys/bus/ccw/devices/0.0.5600/
total 0
drwxr-xr-x 4 root root    0 Sep  2 16:25 .
drwxr-xr-x 4 root root    0 Sep  2 16:25 ..
-r--r--r-- 1 root root 4096 Sep  2 16:25 alias
-r--r--r-- 1 root root 4096 Sep  2 16:33 availability
drwxr-xr-x 3 root root    0 Sep  2 16:25 block
-rw-r--r-- 1 root root 4096 Sep  2 16:33 cmb_enable
-r--r--r-- 1 root root 4096 Sep  2 16:33 cutype
-r--r--r-- 1 root root 4096 Sep  2 16:33 devtype
-r--r--r-- 1 root root 4096 Sep  2 16:25 discipline
lrwxrwxrwx 1 root root    0 Sep  2 16:25 driver -> ../../../../bus/
ccw/drivers/dasd-eckd
-rw-r--r-- 1 root root 4096 Sep  2 16:33 eer_enabled
-rw-r--r-- 1 root root 4096 Sep  2 16:33 erplog
-rw-r--r-- 1 root root 4096 Sep  2 16:33 failfast
-r--r--r-- 1 root root 4096 Sep  2 16:33 modalias
-rw-r--r-- 1 root root 4096 Sep  1 18:35 online
drwxr-xr-x 2 root root    0 Sep  2 16:25 power
-rw-r--r-- 1 root root 4096 Sep  2 16:25 readonly
-r--r--r-- 1 root root 4096 Sep  2 16:33 status
lrwxrwxrwx 1 root root    0 Sep  2 16:33 subsystem ->
../../../../bus/ccw
-rw-r--r-- 1 root root 4096 Sep  2 16:25 uevent
-r--r--r-- 1 root root 4096 Sep  2 16:25 uid
-rw-r--r-- 1 root root 4096 Sep  2 16:33 use_diag
-r--r--r-- 1 root root 4096 Sep  2 16:33 vendor
```

'alias':0 for base device,
1 for alias device

'uid': unique-id of the base device
(vendor.serial.SSID.UA)

PAV/HyperPAV Toolbox (cont'd)

- * Base/Hyper PAV base device:

```
uid = IBM.75000000010671.5600.00  alias = 0
```

- * Base PAV alias device:

```
uid = IBM.75000000010671.5600.00  alias = 1
```

- * Hyper PAV alias device:

```
uid = IBM.75000000010671.5600.xx  alias = 1
```

- * On z/VM multiple minidisks can reside on the same device.

An additional qualifier allows to distinguish them.

(needs PTFs for VM APAR VM64273 on z/VM 5.2.0 and higher and SLES 10 SP2 or RHEL 5.3)

Long uid for VM :

```
uid = IBM.75000000010671.5600.00.00000000000037400000000000000000
```

PAV/HyperPAV Toolbox (cont'd)

- * New option `-u` / `--uid` allows to display and sort by uid.
- * Groups of base and alias devices are easy to identify:

```

$ lsdasd -u
Bus-ID      Name      UID
=====
0.0.7500    dasde     IBM.75000000010671.7500.00
0.0.7501    dasdf     IBM.75000000010671.7500.01
0.0.75fb    alias     IBM.75000000010671.7500.xx
0.0.75fc    alias     IBM.75000000010671.7500.xx
0.0.75fe    alias     IBM.75000000010671.7500.xx
0.0.75ff    alias     IBM.75000000010671.7500.xx
0.0.7e9e    dasda     IBM.75000000058251.7e00.9e
0.0.7e9f    dasdb     IBM.75000000058251.7e00.9f
0.0.5600    dasdc     IBM.75000000092461.5600.00
0.0.56fe    alias     IBM.75000000092461.5600.00
0.0.56ff    alias     IBM.75000000092461.5600.00
0.0.5601    dasdd     IBM.75000000092461.5600.01
0.0.56fb    alias     IBM.75000000092461.5600.01
0.0.56fc    alias     IBM.75000000092461.5600.01
    
```

} Hyper PAV group
} Base PAV group
} Base PAV group

- * For a full description of all features see `'man lsdasd'`.

PAV/HyperPAV Toolbox (cont'd)

- * `dasdinfo` provides various device identifiers as used in scripts and configuration files.

- * Example 1: export all information (e.g. for use in `udev`):

```
$ ./dasdinfo -a -e -b dasde
ID_BUS=ccw
ID_TYPE=disk
ID_UID=IBM.75000000092461.4a00.df
ID_XUID=IBM.75000000092461.4a00.df.00000015000000320000000000000000
ID_SERIAL=0X0202
```

- * Example 2: print extended uid (e.g. callout in `multipath.conf`):

```
$ /sbin/dasdinfo -x -b dasda
75000000092461.4a00.df.00000015000000320000000000000000
```

- * Example 3: print uid (e.g. callout in `multipath.conf`):

```
$ /sbin/dasdinfo -u -b dasda
IBM.75000000092461.4a00.df
```

- * For a full description of all features see '`man dasdinfo`'.

General Hints and Tips

- * Base PAV and Hyper PAV are priced features (LIC) of your storage server.
- * No support for Dynamic PAV!
 - If you use a workload manager to manage PAV alias devices with Dynamic PAV, you must exclude Linux devices.
- * On LPAR: Alias devices only become 'visible' after at least one base device of the same LCU (same SSID) has been set online.
 - It may take a few seconds before the alias devices are available.
- * Alias devices 'consume' minor numbers and device names.
 - Example: The third device in your configuration will always get the internal name 'dasdc' assigned, even if it is an alias device and not a block device on it's own.

General Hints and Tips (cont'd)

- * Examples in this presentation use a very simple naming scheme:
 - Device number = SSID + Unit Address
- * This is just one possible definition.
- * Actual device numbers will depend on your I/O configuration (HCD / IOCP).
- * It is possible to define alias devices in the second subchannel set.
- * Example:
 - Base device: 0.0.5600
 - Alias device: 0.1.5600
- * Resources on the web:
http://www.ibm.com/developerworks/linux/linux390/development_documentation.html
- * PDF for download: 'How to Improve Performance with PAV'

General Hints and Tips (cont'd)

- * Basic instructions on using dm multipath:
 - SLES10: Storage administration Guide, Chapter: Managing Multipath I/O for Devices
 - RHEL5: DM Multipath, DM Multipath Configuration and Administration
- * When working directly with a base device (e.g. dasdfmt, fdasd) make sure to set all alias devices offline.
- * DASDs are blacklisted per default.
Add a blacklist exception to your multipath.conf:

```
blacklist_exceptions {  
    device {  
        vendor IBM  
        product S/390.*  
    }  
}
```

PAV vs. HyperPAV

* PAV

- Well defined mapping between base and alias devices.
- This results in a predictable resource utilization

* HyperPAV

- Dynamic allocation and utilization of resources via pooled alias devices
- Advantage: In average this leads to a higher utilization, resulting in I/O transfer rates

If you run current Linux on System z distributions, prefer HyperPAV over PAV

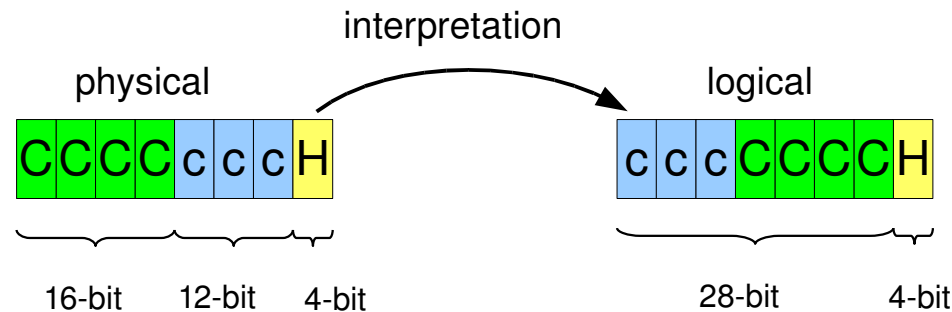
Problem Statement: Legacy ECKD size limitation

Large Volume Support

- * Largest model 9 ECKD DASD:
 - 65520 cylinders (0xFFF0), 15 heads
 - 54GB raw capacity, or
 - 45GB when formatted with 4096 byte blocks
- * Size is restricted by the legacy ECKD interface
 - 16-bit cylinder address
 - 16-bit head address, but more than 15 heads would break legacy code.
- * Legacy ECKD interface is implemented in many software systems (z/OS, z/VM, z/VSE).
- * Extension must be compatible to legacy software.

Large Volume Support (cont'd)

- * Number of heads per cylinder cannot be extended without breaking legacy software.
- * Idea: Use 'free' head bits to extend the cylinder address:



- * 3390 Model A
- * Other name: Extended Address Volume (EAV)
- * Currently up to 262668 cylinders
(approximately 223.2 GB raw, 180GB formatted)

Linux support

- * Upstream Kernel 2.6.30, s390-tools 1.8.2
- * Working with distribution partners on integration into current distributions.
- * Large Volumes can be used like any other volume
 - Usable as boot / IPL device
 - Usable as dump device
- * Can be combined with PAV / Hyper PAV

Compatibility

- * Kernel and tools without Large Volume support will recognize such a volume with 65535 (0xFFFFE) cylinders.
- * When formatted without Large Volume support, part of the DASD will stay unformatted and unusable.
 - When moved to a system with large volume support, the formatted part will stay usable.
- * **Caution:**
When partitions were created on a system with Large Volume support, these partitions are not recognized correctly on a system without support.
 - CDL partitions are not recognized at all
 - LDL partition will be recognized with a wrong size
- * **Recommendation:**
Use Large Volumes only with kernel and tools that support them.

More information

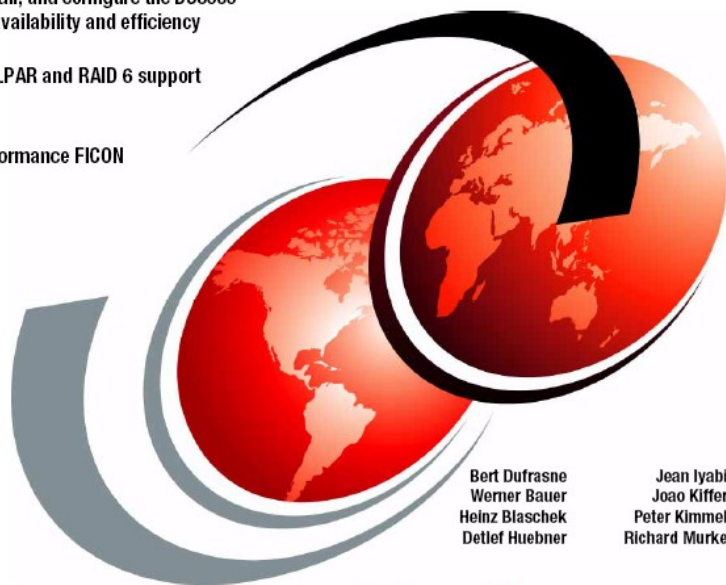


IBM System Storage DS8000 Architecture and Implementation

Plan, install, and configure the DS8000 for high availability and efficiency

Variable LPAR and RAID 6 support

High Performance FICON



Bert Dufrasne
Werner Bauer
Heinz Blaschek
Detlef Huebner

Jean Iyabi
Joao Kiffer
Peter Kimmel
Richard Murke

Redbooks

ibm.com/redbooks

<http://www.vm.ibm.com/storman/pav/pav3.html>

The screenshot shows the IBM website interface for the article 'HyperPAV on z/VM'. The page includes a navigation menu, a search bar, and a sidebar with related links. The main content area contains the title, a paragraph of text, and a list of links.

IBM HyperPAV Support on z/VM

On October 31, 2006, IBM announced the plan to offer enhancements for Parallel Access Volumes (PAV) with support for HyperPAV on the IBM System Storage DS8000 series (M/T 2107). The HyperPAV capability was offered on z/OS 1.6 and later releases in November 2006. Announcement letter: [US ENUS106-811](#)

We had been asked about this support for z/VM and we understood your interest in having this support in z/VM. On Feb. 6, 2007, IBM announced that z/VM V5.3 supports the IBM Hyper Parallel Access Volume (HyperPAV) function optionally provided by the IBM System Storage(TM) DS8000 disk storage systems.

For the announcement letter and information about z/VM V5.3, see [z/VM V5.3 resources](#).

Refer to these pages for VM PAV support information,

- [introduction](#) to VM PAV support
- [initial PAV support](#) for z/VM V5.2 systems without the PTF for APAR VM63952 and for earlier z/VM systems
- [PAV minidisk support](#) for z/VM V 5.2 with PTF for APAR VM63952

This page contains

- [IBM HyperParallel Access Volumes \(HyperPAV\) Overview](#)
- [Using IBM HyperPAVs](#)
- [HyperPAV Pools](#)
- [HyperPAV Dedicated DASD](#)
- [Using HyperPAV Minidisks](#)
- [Using HyperPAV Minidisks with Exploiting Operating Systems](#)
- [Using HyperPAV Minidisks with Non-Exploiting Operating Systems](#)
- [z/VM Restrictions on Using HyperPAV](#)
- [Reference documents](#)

IBM HyperPAV Support Overview

HyperPAV support complements the existing basic PAV support in z/VM V5.2, for applicable supporting disk storage systems. The HyperPAV function potentially reduces the number of alias-device addresses needed for parallel I/O operations since HyperPAVs are dynamically bound to a base device for each I/O operation instead of being bound statically like basic PAVs.

More information

ibm.com/systems/z/linux

www.vm.ibm.com

United States [change]

Home Solutions Services Products Support & downloads My IBM

Welcome [IBM Sign in] [Register]

IBM Systems > Mainframe servers > Operating systems >

Linux on IBM System z™

Request a quote

System z10

Featured topics

Linux on System z can help transform your IT infrastructure in dynamic infrastructure

How? Linux on System z can provide an efficient, green and optimized infrastructure.
→ Learn more

Web 2.0 on Linux on System z

The Web 2.0 capabilities of Linux on System z demonstrate the flexibility and openness of the System z environment.
→ Learn more

New IFL-pricing on z10 BC to support the deployment and grow workloads

- Lower priced IFL for the System z10 BC - \$47,500 USD²
- Lower memory prices when coupled with the purchase on an IFL \$2,250 USD / GB
- Hot-pluggable I/O drawers help reduce downtime and increase flexibility.

United States [change]

Home Solutions Services Products Support & downloads My IBM

IBM Systems > System z > z/VM >

z/VM®

the newest VM hypervisor based on 64-bit z/Architecture.

Request a quote

Currently supported releases of z/VM

Available:	z/VM V5.3
Also supported:	z/VM V5.2

The z/VM hypervisor is designed to help clients extend the business value of mainframe technology across the enterprise by integrating applications and data while providing exceptional levels of availability, security, and operational ease. z/VM virtualization technology is designed to allow the capability for clients to run hundreds to thousands of Linux servers on a single mainframe running with other System z operating systems, such as z/OS, or as a large-scale Linux-only enterprise server solution. z/VM V5.3 can also help to improve productivity by hosting non-Linux workloads such as z/OS, z/VSE, and z/TPF.

Summary of News and Updates

View 03 June 2008 updates.
Read the [z/VM and VM Site News and Changes](#) for a summary of VM-related news, announcements, pointers, new classes, and places to hear about z/VM virtualization technology.

Worldwide announcement letters (US letters / product links below)

- May 06, 2008 z10™ EC Internet access and coupling improvements
- Feb. 26, 2008 Announcing System z10™ Enterprise Class Internet delivery for z/VM orders via ShopzSeries
- Jan. 25, 2008 IBM Integrated Removable Media Manager (IRMM)
- Aug. 07, 2007 IBM z/VM V5.3 - Additional enhancements available
- Jun. 12, 2007 z9 EC and z9 BC - delivering greater value for everyone
- Apr. 18, 2007 IBM z/VM V5.3 - Improving scalability, security, and virtualization technology
- Feb. 06, 2007 z/VM V5.2 New Function Added in Support of System z9
- Apr. 27, 2006

Mainframe history

1964 2004
40 years and counting
Explore IBM mainframe innovation

Is your VM current?
Thinking about migration?

Technical Conference
IBM System z Expo
featuring z/OS, z/VM, z/VSE, Linux on System z
October 13-17, 2008
Las Vegas, NV

The future runs on System z... and your future begins today.
→ Learn more

Questions?

