HyperPAV and Large Volume Support for Linux on System z

Dr. Holger Smolinski
IBM Germany Research & Development GmbH

2010-08-05
9223

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.

*, 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®

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

The following are trademarks or registered trademarks of other companies.

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.

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

Notes:
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.

Agenda

- PAV/ HyperPAV Support
  - Motivation: Overcoming the architectural limit
  - Alternative Approach: LVM Volume Striping
  - PAV and Hyper PAV concepts
  - Old: Device mapper based implementation
  - New: DASD device driver based implementation
  - Hyper PAV
  - Tool support
  - Hints and Tips
- Large Volume Support
  - Large Volumes
  - Compatibility

Overcoming the architectural Limit

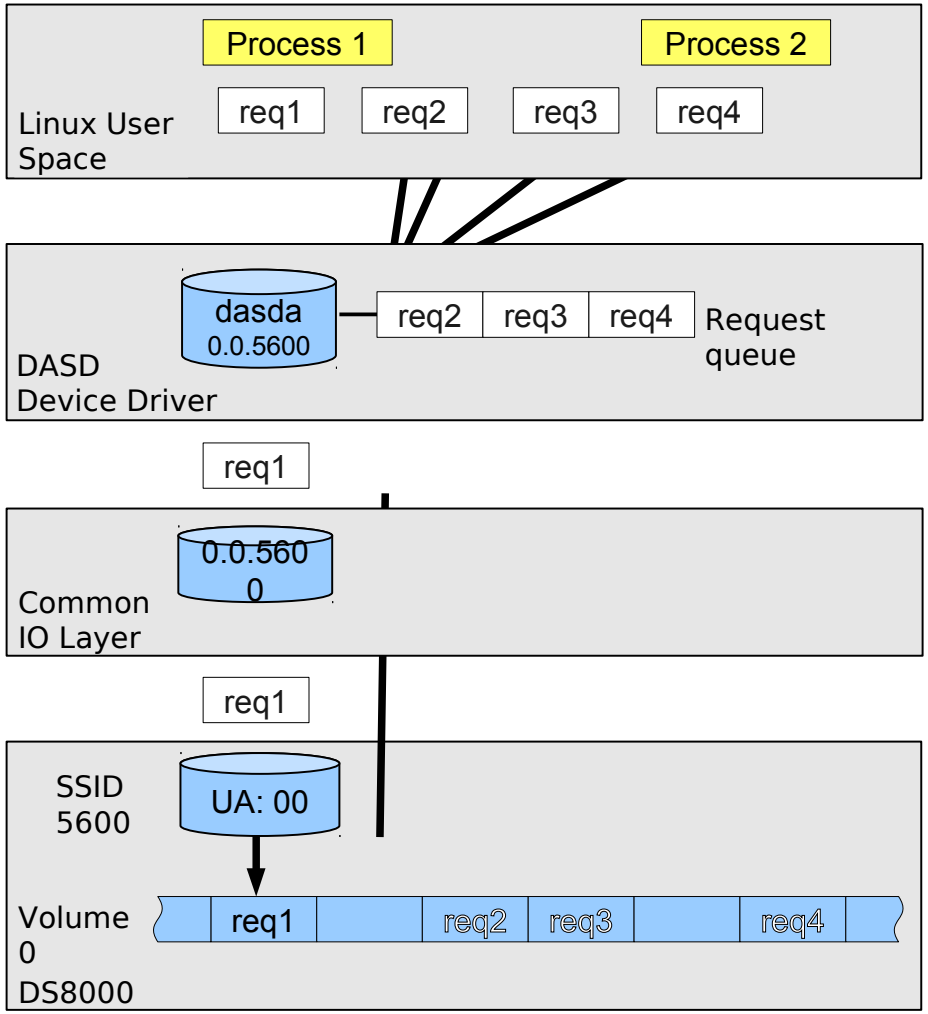
- Problems as seen by the user:
  - More and more data needs to be processed at high speed.
  - Access pattern has limited predictability.
  - Concurrent access required
- System z Architecture has built-in limits
  - Any single subchannel can only process a single IO at a time.
  - Per subchannel: Data rate = request size * io rate

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.
- Sending requests in parallel would improve performance. → see SCSI Tagged Command queueing

- Note: The SSID identifies the logical control unit (LCU) on the storage server.



LVM striping was good for a long time

- Idea:
  - Distribute I/Os among many devices
  - Concurrent IO to N devices possible, if
    - Data needed concurrently is not located on the same device
  - The more stripes, the more performance
- Caveats:
  - More stripes require more volumes
    - Large databases require hundreds of volumes
    - almost unmanageable
  - Sequential access benefits less than real random access
  - Some access patterns do not benefit at all

(Hyper) Parallel Access Volumes - Concepts

- Still one I/O per subchannel,
  - but: multiple subchannels per device.
- Storage server could address independent areas in parallel
- One PAV base + multiple alias devices per volume
- 'Base' PAV
  - Alias address assigned to base address
  - Assignment can be changed 'dynamic PAV'
    - Currently unsupported in Linux - DO NOT USE!
- Hyper PAV
  - I/O on alias is be directed to base volume per request
  - Limited to base addresses in same logical control unit
  - Fallback to Base PAV, if Hyper PAV is not supported

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)
  - Vendor supplied Storage Serves (e.g. EMC, HDS)
- The HyperPAV feature is available, for example, for IBM System Storage DS8000 series systems.

- PAV base and alias volumes require special configuration in the IOCDS
- 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
```

z/VM PAV setup

- After hardware configuration is active:
  enter the z/VM QUERY PAV command

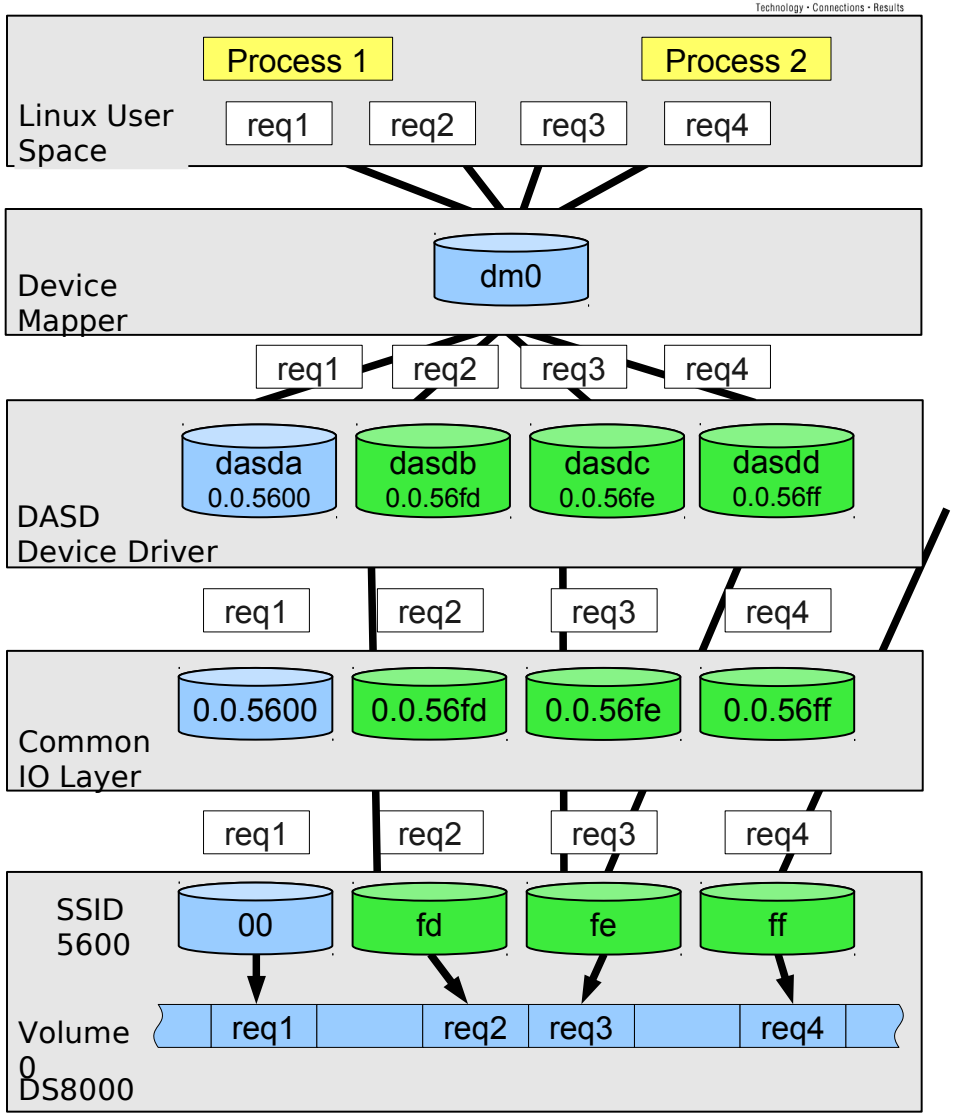
```
#CP QUERY PAV
00: Device  5600 is a base Parallel Access Volume with
  the following aliases: 56FF
00: Device  56FF is an alias Parallel Access Volume
  device whose base device is 5600
CP Q PAV  4DE1
00: Device  4DE1 is not a Parallel Access Volume
```

- Attach Base AND Alias devices to guest

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

Old: device mapper based implementation

- Used on SLES10 and RHEL4 & 5
- With PAV feature, storage systems 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 and alias device have a separate device node assigned
- Example: If device 5600 is a base device and devices 56fd, 56fe and 56ff are alias devices, all devics 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 at least the base device is set online.
- Perform preparation steps on the base device:
  - Use `chccwdev` to set the base device online, if needed.
  - Ensure that the base device is formatted. If it is not already formatted, use `dasdfmt` to format it.
    - 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.
    - Example: `fdasd -a /dev/dasdc`
  - Although the physical device is prepared now, Linux is not aware of the changes made. Hence an additional step is required after any change of the formatting, partitioning or the volume label.
- 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)

- Remove blacklist entries for DASD devices from the multipath.conf configuration file

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

- There is also a built-in configuration – DASD device may be blacklisted: add an exception

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

Configuring PAV volumes with multipath tools (cont'd)

- If it is not already loaded, load the dm_multipath module: (e.g: modprobe dm_multipath)
- Run multipath command to detect and coalesce multiple paths to the same devices
- Enter multipath -ll to display the resulting multipath configuration.

```
# multipath -ll
IBM.75000000092461.2a00.1a dm-0 IBM,S/390 DASD ECKD
[size=2.3G][features=0][hwhandler=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]
```

- Make sure multipathd is started (e.g: /etc/init.d/multipathd start)
  - This is required for recovery of transient failures

Configuring PAV volumes with multipath tools (cont'd)

- 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
```

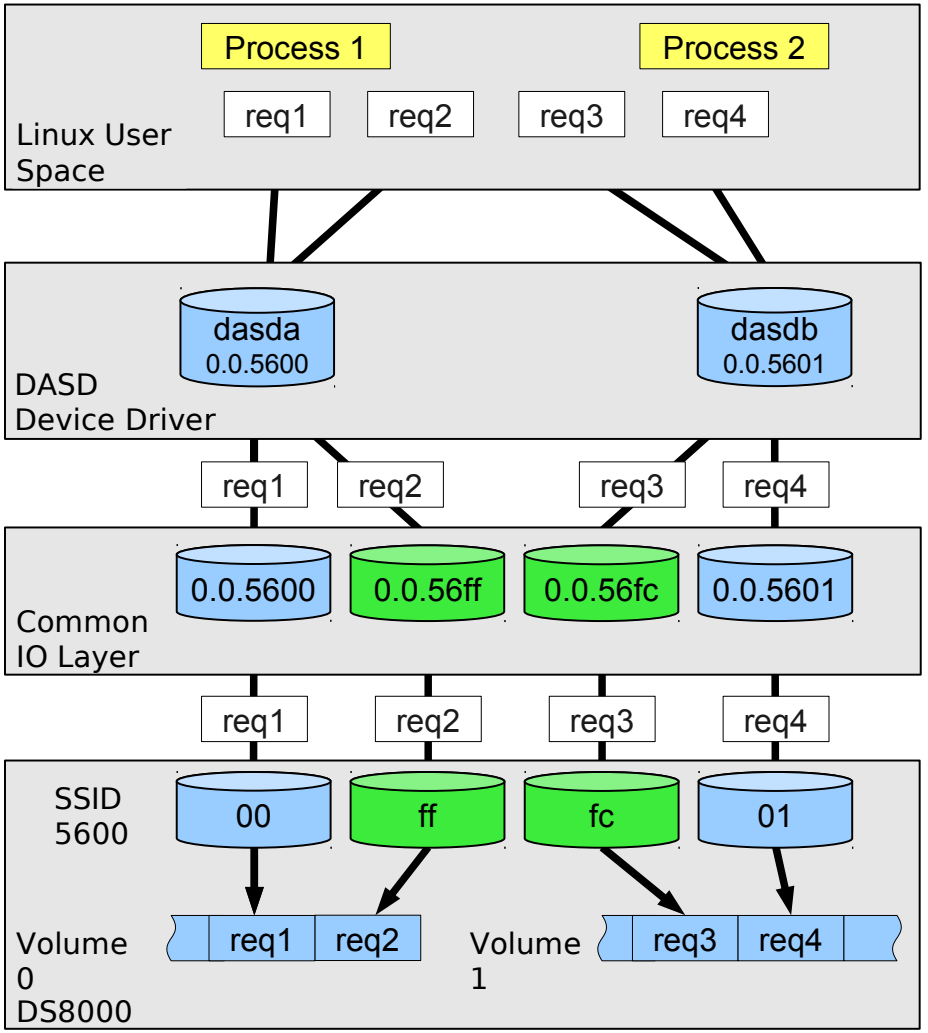
- There is a device node for each multipath device and for each partition on these multipath devices.
- Use these device mapper device nodes for ANY access to the device from now on (mkfs, mount, etc.)
  - Although updates to a DASD device will update the pyhsical device, Linux will not reflect them for other aliases or the device mapper device and aliases.

Considerations when using LVM

- Optionally use LVM2 to configure the multipath device(!) into a volume group
- You must make sure, that LVM uses the multipath devices only, since there also exist device nodes for base and alias devices.
  - set a filter in /etc/lvm.conf:
    `devices { filter=[ "r|dasd|", “a|mapper|” ] }`
- Remember: NEVER use a dasd device for other purposes than dasdfmt or fdasd – keep in mind to set base and all alias device offline and back online after changes to the formatting, the partitioning or the volume label.

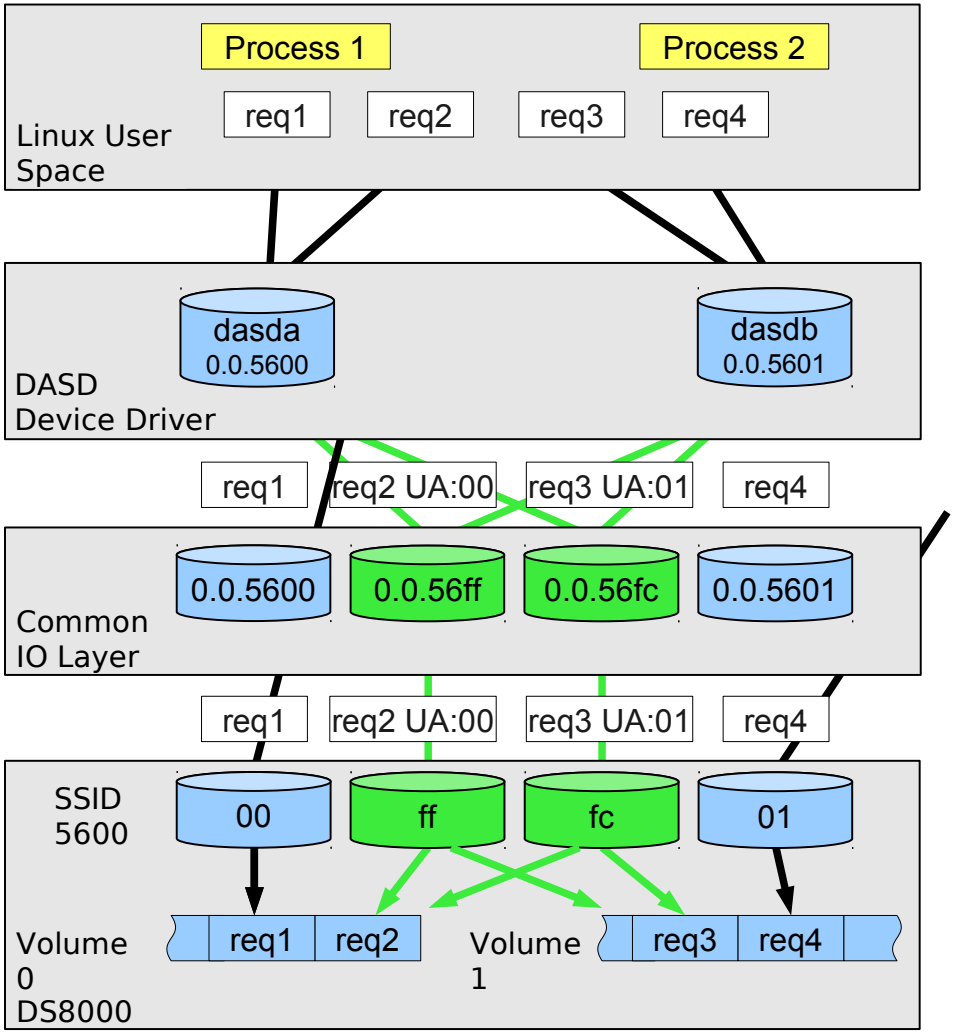
New DASD device driver implementation Base PAV example

- SLES11 and upcoming distributions only!
- With the PAV feature, storage systems 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 must be set online first, followed by the aliases
- The aliases can lead to gaps in the naming scheme for device nodes if using the `dasd =` … kernel parameter
- *Apart from assuring that the corresponding aliases for a base device are online, user space processes need no special handling for accessing a PAV.*



New DASD device driver implementation Hyper PAV example

- SLES11 and upcoming distributions only!
- With the HyperPAV feature there is no 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)
- If the prerequisites for HyperPAV are not met, 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 base 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 (Hyper) PAV configuration on Storage Server

(2)Define System z storage configuration (IOCDS)

(3) Basic DASD configuration

(4) That's it – nothing else to do. *No multipath configuration needed. No more pitfalls related to formatting / partitioning / LVM*

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: a long uid for VM :

```
uid = IBM.75000000010671.5600.00.0000000000037400000000000000000
```

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.0000001500000032000000000000000000
ID_SERIAL=0X0202
```

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

```
$ /sbin/dasdinfo -x -b dasda
75000000092461.4a00.df.0000001500000032000000000000000000
```

- 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 for (Hyper) PAV

- It is possible to define alias devices in the second subchannel set.
  - Example:
    - Base device: 0.0.5600
    - Alias device: 0.1.5600
- Backward compatible: Hyper PAV will work like Base PAV if the Linux version or Storage Controller do not support Hyper PAV
- When working directly with a base device (e.g. dasdfmt, fdasd) make sure to set all alias devices offline.
- PDF for download: 'How to Improve Performance with PAV'

General Hints and Tips for (Hyper) PAV

- Base PAV and Hyper PAV are priced features (LIC) of your storage server.
- No Linux support for Dynamic PAV (will come later)!
  - 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.

HyperPAV: LPAR and z/VM

- Alias devices can not be detected until at least one base device is set online
- z/VM V5.3 and above support the Hyper Parallel Access Volume (HyperPAV) function optionally provided by the IBM System Storage DS8000™ disk storage systems without additional PTFs.
- z/VM provides support of HyperPAV volumes as linkable minidisks.
- This support is also designed to transparently provide the potential benefits of HyperPAV volumes for minidisks owned or shared by guests that do not specifically exploit HyperPAV volumes, such as Linux and CMS.
- With Linux distributions, which do not support BasePAV or HyperPAV, z/VM exploits the feature only, if multiple minidisks or guests are using the same device

PAV vs. HyperPAV

- PAV
  - Well defined mapping between base and alias devices.
  - This results in a predictable resource utilization
- Dynamic PAV
  - + Reconfigurable mapping of base and alias devices
- 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 (currently SLES11) prefer HyperPAV over PAV

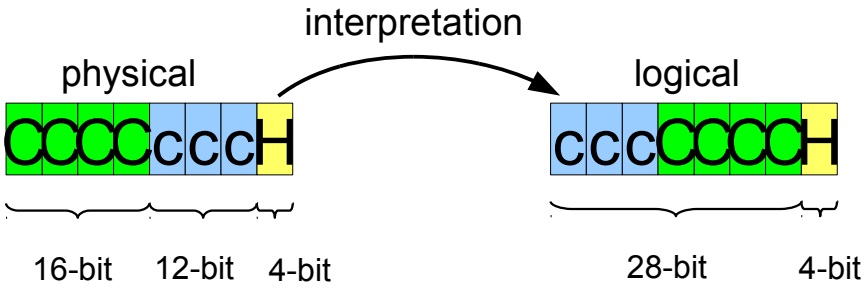
Problem Statement:
Legacy ECKD size limitation

Large Volume Support

- Largest model 54 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
  - CCHHR address scheme
  - 16-bit cylinder address
  - 16-bit head address, but
    more then 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)

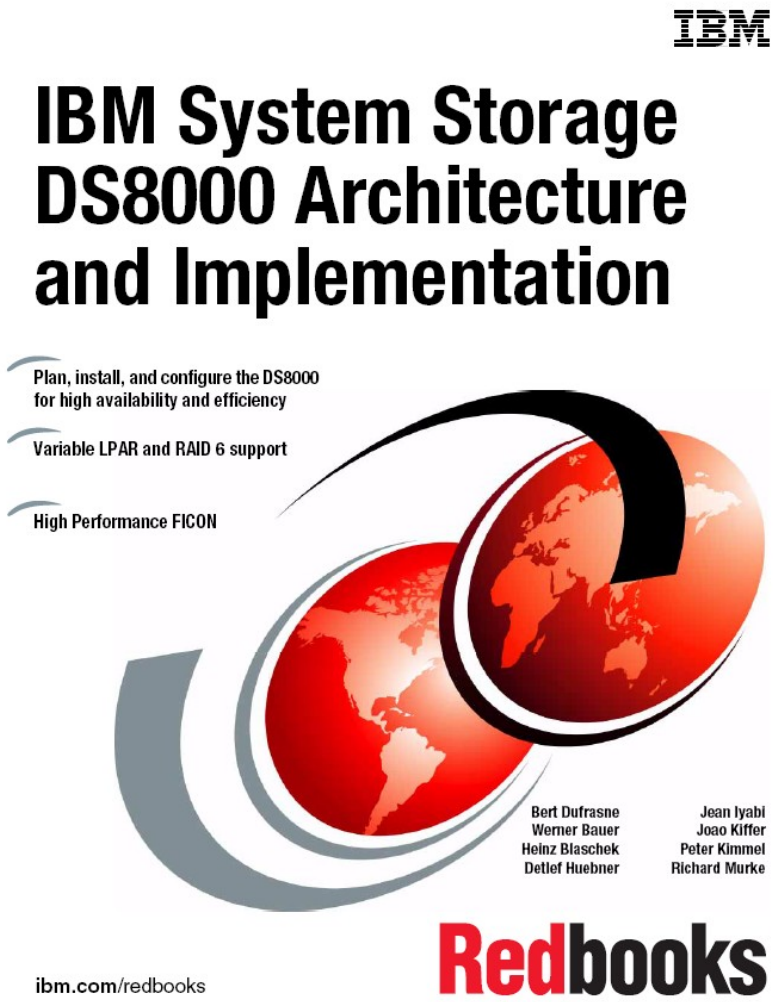
Compatibility

- Kernel and tools without Large Volume support will recognize such a volume with 65535 (0xFFFE) 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.

Linux support

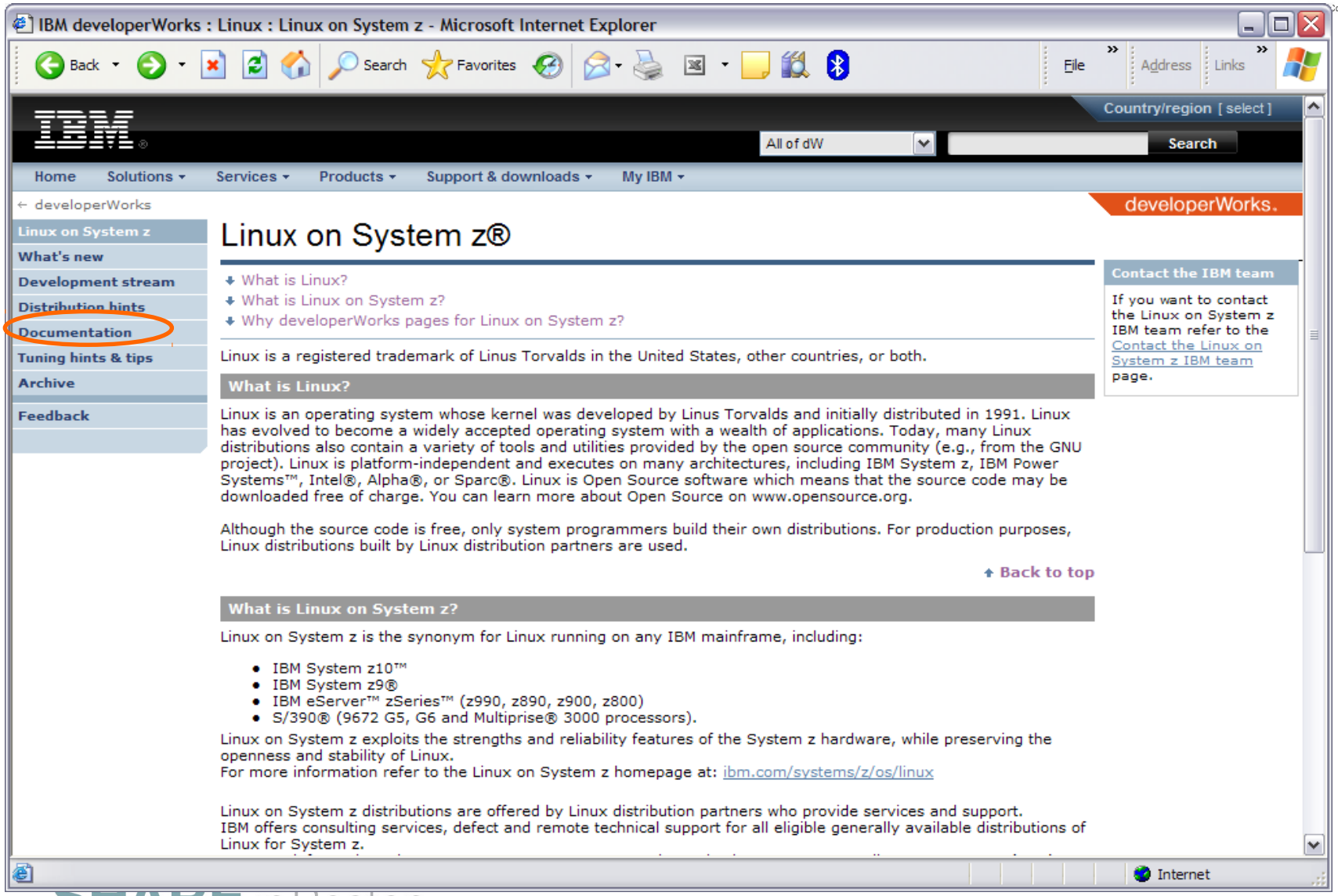
- Upstream Kernel 2.6.30, s390-tools 1.8.2
- Feature included in
  - SLES11 SP1
  - Integration in Red Hat distributions ongoing
- Large Volumes can be used like any other volume
  - Usable as boot / IPL device
  - Usable as dump device
- Can/Should be combined with PAV / Hyper PAV
  - Remember: only a single concurrent IO per subchannel
  - w/o PAV: a single concurrent IO per ~200GB!

More information

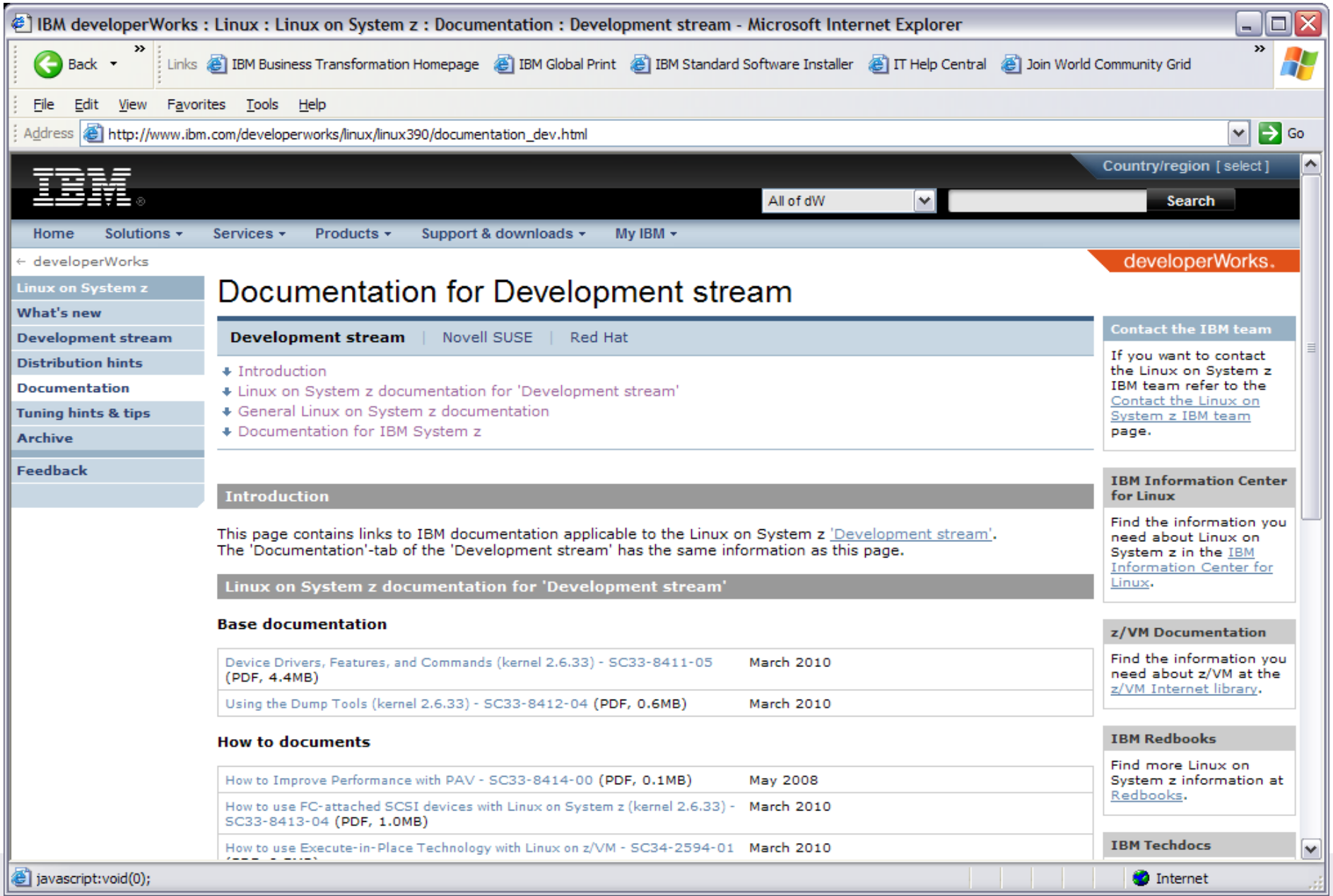


http://www.vm.ibm.com/storman/pav/pav3.html

developerWorks – entry page for documentation



Development stream – Novell SUSE – Red Hat documentation



More information

ibm.com/systems/z/linux

www.vm.ibm.com

Questions?

HyperPAV and Large Volume Support for Linux on System z

Dr. Holger Smolinski
IBM Germany Research & Development GmbH

2010-08-05
9223

Appendix

Where to find information

The Linux on System z documentation can be found at these key locations:

IBM developerWorks	ibm.com/developerworks/linux/linux390/documentation_dev.html ibm.com/developerworks/linux/linux390/perf/index.html
IBM Redbooks	http://www.redbooks.ibm.com
IBM Techdocs	http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs
z/VM Internet Library	http://www.vm.ibm.com/library/
IBM Information Center for Linux	http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp
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

PAV Performance

- Generally speaking, disk performance depends on the time required to do a single I/O operation on a single disk volume in z/VM.
- This time is generally understood as response time, which is composed of queue time and service time. Queue time is the time a guest's I/O spends waiting to access the real volume.
- Service time is the time required to finish the real I/O operation.
- PAV can help to increase the access
- *But if a volume is not experiencing queuing, adding aliases does not help to enhance the volume's performance, for the time required to perform an I/O operation is not appreciably changed by the queue time.*
- In some cases, increasing the number of aliases for a volume might result in the increased service time for that volume.
- Most of the time, the decrease in wait time outweighs the increase in service time, so the response time improves.
- The z/VM Performance Toolkit provides a report on the response time and related performance statistics for the real devices. Use the command DEVICE in the performance toolkit to access the report.

Impact of DS8000 Storage Pool Striping

Starting with DS8000 License Machine Code 5.30xx.xx, it is possible to stripe the extents of a DS8000 Logical Volume across multiple RAID arrays.

DS8000 Storage Pool Striping will improve throughput for some workloads.

It is performed on a 1 GB granularity, so it will generally benefit random workloads more than sequential ones.

If you are already using a host-based striping method (for example, LVM Striping or DB2 database container striping), there is no need to use Storage Pool Striping.

However, it is possible. You should then combine the wide stripes on DS8000 with small granularity stripes on the host. The recommended size for these is usually between 8 and 64 MB. If large stripes on both DS8000 and attached host interfere with each other, I/O performance may be affected.