Today

Linux on IBM Power: Best Practices in Virtualized Environments

Starting at 10:00 am UK time by Dr. Michael Perzl

Smart Meeting → Put questions into the Chat box
or AT&T Toll Free phone for better audio

- 0800-368-0638 = UK Toll Free
- 0203-059-6451 = UK but you pay for the call
- Then 6403785# Participant Code
- Other countries see chat box for the website
- Please Mute with *6

Previous Sessions:
Linux for AIX/IBM i guys
PowerKVM Deep Dive
More Tricks Power Masters
Power8 from hands-on
Power up your Linux
PowerVC
PowerVP
SSP4
Best Practices
Tricks of Power Masters
IBMi and External Storage
Monitoring with ITM
And more…..

Future Sessions →

- To be planned (HMC enhancements, IBM I Licensing, etc)
- Suggestions Welcome

Twitter:
Gareth Coates @power_gaz Nigel Griffiths @mr_nmon
Jyoti Dodhia @JyotiDodhia Mandie Quartly @mandieq
Website: http://tinyurl.com/PowerSystemsTechnicalWebinars
Youtube Channel: http://tinyurl.com/IBMPowerVUGYoutubeChannel

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.

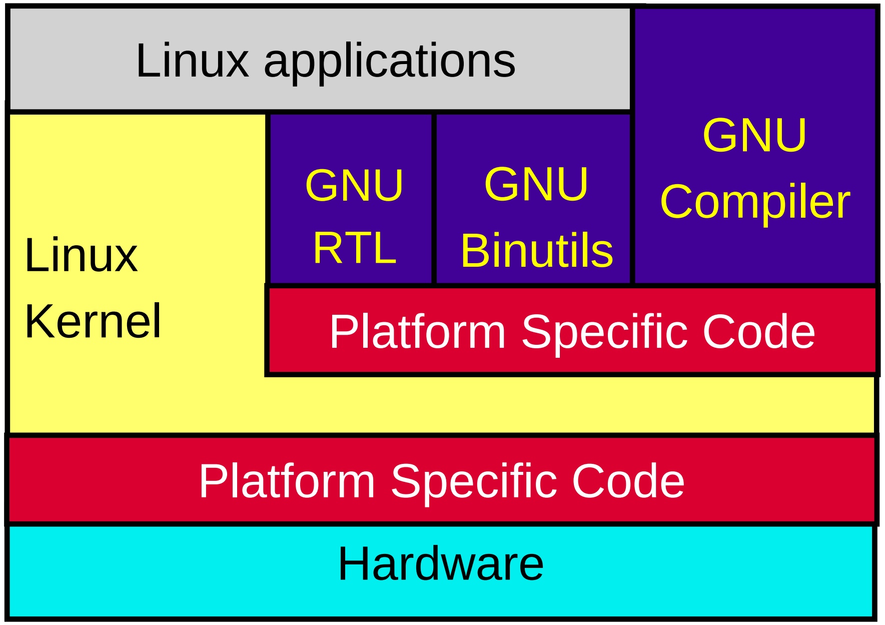
Agenda

- Linux on Power vs. Linux on x86
- Netbooting PowerLinux
- Virtual network options for Linux clients
  - Bonding setup
  - Shared Ethernet Adapter (SEA) failover (and load sharing)
- Virtual storage options for Linux clients
  - Multipath I/O with Linux on Power
  - Software RAID with Linux
  - Device discovery with Linux on Power
  - Adding/removing disks, rescan devices
  - Resize of Linux on Power file systems
  - High availability setup with VSCSI
- Setting up a RHEL repository for YUM
- How to obtain information from within a Linux LPAR
- Change number of SMT threads
- Optimizing for Linux on Power
- Resources, Links

Linux on Power vs. Linux on x86

Linux Kernel Design

- Linux has a hardware-independent design
- Monolithic kernel with dynamically loadable extensions (same as AIX)
- APIs and look & feel consistent across all platforms
- Applications and knowledge can easily be transferred from one platform to another



Linux kernel statistics

- Source code lines counted with “SLOCCount“ by David A. Wheeler

	Linux kernel versions											
Subdirectory	3.12.1		3.10.20		3.4.70		3.2.52		2.6.34.14		2.6.32.61	
arch	2018467	17,13%	2020376	17,95%	1926105	18,85%	1877804	18,86%	1646194	19,69%	1570168	19,82%
block	19163	0,16%	18465	0,16%	18279	0,18%	14433	0,14%	11235	0,13%	11065	0,14%
crypto	61726	0,52%	61380	0,55%	47464	0,46%	42645	0,43%	38331	0,46%	38017	0,48%
Documentation	9162	0,08%	9928	0,09%	9665	0,09%	11580	0,12%	9943	0,12%	9272	0,12%
drivers	6904791	58,58%	6455093	57,34%	5768985	56,47%	5622686	56,46%	4591328	54,91%	4323922	54,59%
firmware	1876	0,02%	1876	0,02%	1876	0,02%	1876	0,02%	1865	0,02%	1865	0,02%
fs	782449	6,64%	770525	6,85%	713068	6,98%	700390	7,03%	670538	8,02%	638657	8,06%
include	365800	3,10%	354560	3,15%	312358	3,06%	302256	3,04%	255012	3,05%	244563	3,09%
init	2551	0,02%	2509	0,02%	2446	0,02%	2394	0,02%	2273	0,03%	2261	0,03%
ipc	6385	0,05%	6365	0,06%	5755	0,06%	5710	0,06%	5478	0,07%	5397	0,07%
kernel	136768	1,16%	135956	1,21%	123350	1,21%	121044	1,22%	102924	1,23%	98061	1,24%
lib	65393	0,55%	35770	0,32%	33705	0,33%	28499	0,29%	22426	0,27%	20965	0,26%
mm	62564	0,53%	60457	0,54%	55366	0,54%	54902	0,55%	46865	0,56%	44553	0,56%
net	564614	4,79%	556628	4,94%	509430	4,99%	494471	4,97%	423296	5,06%	412216	5,20%
samples	2050	0,02%	1991	0,02%	1294	0,01%	1232	0,01%	614	0,01%	565	0,01%
scripts	43467	0,37%	42712	0,38%	37699	0,37%	37333	0,37%	31454	0,38%	31141	0,39%
security	47522	0,40%	46418	0,41%	44847	0,44%	44186	0,44%	30965	0,37%	32059	0,40%
sound	589127	5,00%	580237	5,15%	536311	5,25%	534051	5,36%	438668	5,25%	413207	5,22%
tools	95192	0,81%	89474	0,79%	63180	0,62%	56199	0,56%	28110	0,34%	18966	0,24%
usr	567	0,00%	567	0,01%	567	0,01%	567	0,01%	524	0,01%	524	0,01%
virt	6698	0,06%	5407	0,05%	4819	0,05%	4697	0,05%	4025	0,05%	3689	0,05%
Total	**11786332**		**11256694**		**10216569**		**9958955**		**8362068**		**7921133**	

	Linux kernel versions											
Subdirectory	**3.12.1**		**3.10.20**		**3.4.70**		**3.2.52**		**2.6.34.14**		**2.6.32.61**	
alpha	41188	2,04%	41034	2,03%	41758	2,17%	41977	2,24%	41693	2,53%	41782	2,66%
arc	10934	0,54%	11188	0,55%								
arm	419447	20,78%	435873	21,57%	486569	25,26%	473688	25,23%	358833	21,80%	302182	19,25%
arm64	16232	0,80%	13098	0,65%								
avr32	16419	0,81%	16483	0,82%	16806	0,87%	16932	0,90%	17087	1,04%	17031	1,08%
blackfin	91479	4,53%	91123	4,51%	79678	4,14%	79202	4,22%	78116	4,75%	79957	5,09%
c6x	6668	0,33%	6681	0,33%	6782	0,35%						
cris	69785	3,46%	70005	3,46%	71702	3,72%	71748	3,82%	72218	4,39%	72416	4,61%
frv	18723	0,93%	18743	0,93%	19075	0,99%	19183	1,02%	19338	1,17%	19532	1,24%
h8300	7277	0,36%	7301	0,36%	7571	0,39%	7620	0,41%	7749	0,47%	7921	0,50%
hexagon	6841	0,34%	6842	0,34%	6696	0,35%	6697	0,36%				
ia64	78724	3,90%	78745	3,90%	79521	4,13%	79891	4,25%	80253	4,88%	85138	5,42%
m32r	16315	0,81%	16355	0,81%	16587	0,86%	16694	0,89%	17151	1,04%	17270	1,10%
m68k	114690	5,68%	114736	5,68%	117237	6,09%	117914	6,28%	106055	6,44%	106380	6,78%
m68knommu									13377	0,81%	13446	0,86%
metag	15542	0,77%	15504	0,77%								
microblaze	13401	0,66%	13570	0,67%	13722	0,71%	13476	0,72%	13788	0,84%	11434	0,73%
mips	249489	12,36%	247584	12,25%	189337	9,83%	164230	8,75%	147493	8,96%	139812	8,90%
mn10300	23202	1,15%	23216	1,15%	23450	1,22%	23644	1,26%	18009	1,09%	18114	1,15%
openrisc	5346	0,26%	5399	0,27%	5709	0,30%	5730	0,31%				
parisc	40067	1,99%	40101	1,98%	40053	2,08%	40183	2,14%	40058	2,43%	40113	2,55%
powerpc	**236687**	**11,73%**	**232959**	**11,53%**	**220711**	**11,46%**	**221213**	**11,78%**	**196054**	**11,91%**	**188949**	**12,03%**
s390	54918	2,72%	54409	2,69%	48435	2,51%	47516	2,53%	43173	2,62%	42955	2,74%
score	5086	0,25%	5101	0,25%	5291	0,27%	5284	0,28%	5324	0,32%	5310	0,34%
sh	73588	3,65%	73692	3,65%	83527	4,34%	84448	4,50%	81345	4,94%	77166	4,91%
sparc	89418	4,43%	89447	4,43%	89146	4,63%	89579	4,77%	89492	5,44%	88569	5,64%
tile	39087	1,94%	35514	1,76%	30214	1,57%	30180	1,61%				
um	19374	0,96%	19203	0,95%	19446	1,01%	19621	1,04%	24468	1,49%	24614	1,57%
unicore32	9224	0,46%	9264	0,46%	9717	0,50%	9710	0,52%				
x86	212153	10,51%	210170	10,40%	182514	9,48%	176493	9,40%	159999	9,72%	154914	9,87%
xtensa	17163	0,85%	17036	0,84%	14851	0,77%	14951	0,80%	15121	0,92%	15163	0,97%
Total	**2018467**		**2020376**		**1926105**		**1877804**		**1646194**		**1570168**	

Linux kernel statistics, hardware dependent code

Power Linux kernel = Total LoC of Linux kernel
– “arch subdirectory”
+ “powerpc” subdirectory

	Linux kernel versions					
	3.12.1	3.10.20	3.4.70	3.2.52	2.6.34.14	2.6.32.61
“arch” subdirectory	2018467	2020376	1926105	1877804	1646194	1570168
“powerpc” subdirectory	236687	232959	220711	221213	196054	188949
Total LoC of Linux kernel	11786332	11256694	10216569	9958955	8362068	7921133
Total LoC for Power Linux Kernel	10004552	9469277	8511175	8302364	6911928	6539914
Percentage hardware dependent code	**2,37%**	**2,46%**	**2,59%**	**2,66%**	**2,84%**	**2,89%**

Only about 2-3% of hardware dependent code for Power Linux kernel!

Linux on Power Device and Virtualization Support

- Virtual device support implemented with Linux kernel modules
  - ibmveth - virtual ethernet device driver
  - ibmvscsic - virtual SCSI client device driver
  - ibmvfc - virtual Fibre Channel client device driver
  - ibmvstgt - virtual SCSI target device driver

```
# find /lib/modules -name "ibmv*ko" -print
/lib/modules/x.x.xx/kernel/drivers/net/ibmveth.ko
/lib/modules/x.x.xx/kernel/drivers/scsi/ibmvscsi/ibmvfc.ko
/lib/modules/x.x.xx/kernel/drivers/scsi/ibmvscsi/ibmvscsic.ko
/lib/modules/x.x.xx/kernel/drivers/scsi/ibmvscsi/ibmvstgt.ko
```

- No closed source device drivers for Linux on Power, all Linux on Power device drivers for all virtual and physical devices are open source.
- All contained in the standard “vanilla“ Linux kernel (from http://kernel.org) for a long time!

Linux on Power vs. Linux on x86 commonalities

- Network configuration
  - Linux Bonding
- Storage configuration
  - Multipath I/O
  - Software RAID (MD devices and LVM2)
  - Adding/removing disks, rescan devices etc.
  - Resize of file systems (increase and shrink in size)
- Device discovery of Linux on Power
  - in order of defined devices (PowerVM and PowerKVM)
- Same management tools across platforms
  - Management tools vary depending on Linux distribution

100% compatible

→ **ANY** documentation can be directly applied to Linux on Power too!

Linux on Power vs. Linux on x86 differences

- Disk partitioning
  - Additional PReP partition holding the bootloader (= /dev/hd5 (AIX boot logical volume))
  - MBR is present, but only holds partition table
- System firmware
  - System Management Services (SMS)
  - Config boot order, BOOTP network boot, ...
- Bootloader
  - Yaboot (newer Linux distros now switch to GRUB2)
  - No PXE boot, network boot uses BOOTP protocol
- Full support of Power platform features
  - RAS capabilities
  - DLPAR capabilities
  - SMT settings
- Package names
  - ppc / ppc64 / ppc64le instead of x86_64
- Additional value-add packages
  - iprutils, powerpc-utils, ppc64-utils, servicelog: to better exploit POWER features
  - IBM Software Development Kit: for optimizing source code on POWER
  - IBM Advance Toolchain: optimized GCC and collection of optimized libraries

Netbooting PowerLinux

Linux on Power installation methods

Installation via

- Physical DVD device
- Virtual optical device (virtual DVD/CD-ROM)
- Network based → Netbooting PowerLinux

Netbooting Linux on Power

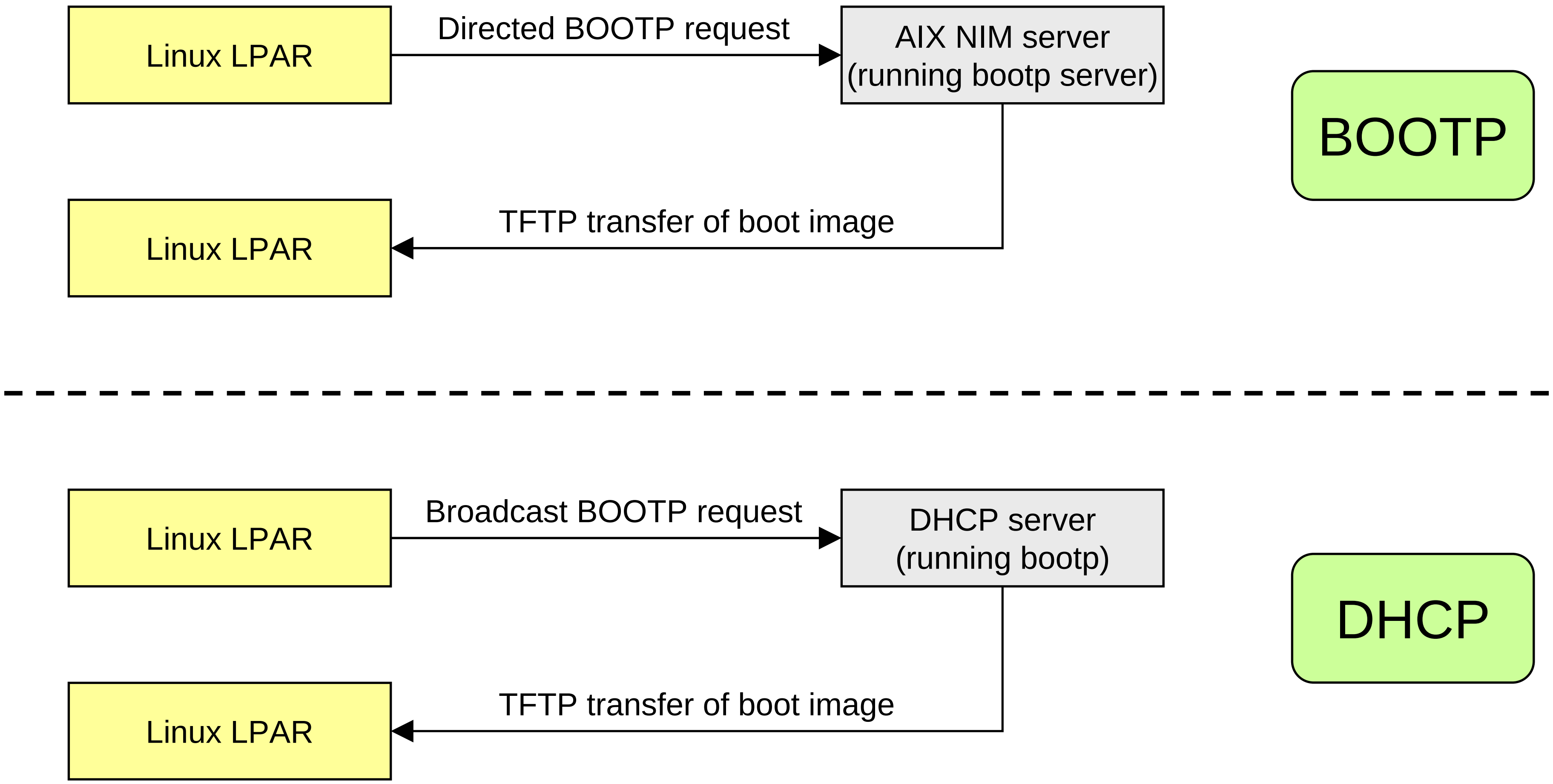
Netbooting Linux on Power uses – same as AIX – the following protocols:

- BOOTP is an IP protocol that informs a computer of its IP address and where on the network to obtain a boot image.
- The TFTP (Trivial File Transfer Protocol) is used to serve the boot image to the client.

Two basic approaches:

1) AIX NIM server
   - Use a directed bootp request
   - Does not require you to know the MAC address of the network boot adapter
2) (Linux) DHCP server
   - Use a broadcast bootp request
     - DHCP server must support BOOTP protocol
   - Requires you to know the MAC address of the network boot adapter

Directed BOOTP vs. broadcast BOOTP request



Linux LPAR

Directed BOOTP request

AIX NIM server (running bootp server)

BOOTP

Linux LPAR

TFTP transfer of boot image

Linux LPAR

Broadcast BOOTP request

DHCP server (running bootp)

DHCP

Linux LPAR

TFTP transfer of boot image

Netbooting PowerLinux – TFTP resources
Red Hat Linux boot images

- Setup of TFTP resources in /tftpboot
- Same for AIX NIM server and Linux DHCP server (for installation via a Linux network install image)

```
root@nim:/tftpboot> ls -l rhel*-netboot.img
-rw-r--r--    1 root     root       6220088 Mar 14  2013 rhel4u3-netboot.img
-rw-r--r--    1 root     root       6355144 Mar 14  2013 rhel4u4-netboot.img
-rw-r--r--    1 root     root       6531312 Mar 14  2013 rhel4u5-netboot.img
-rw-r--r--    1 root     root       6932160 Mar 14  2013 rhel4u6-netboot.img
-rw-r--r--    1 root     root       7218492 Mar 14  2013 rhel4u7-netboot.img
-rw-r--r--    1 root     root       7614464 Mar 14  2013 rhel4u8-netboot.img
-rw-r--r--    1 root     root       8022120 Mar 14  2013 rhel5-netboot.img
-rw-r--r--    1 root     root       8546364 Mar 14  2013 rhel5u1-netboot.img
-rw-r--r--    1 root     root       9139304 Mar 14  2013 rhel5u2-netboot.img
-rw-r--r--    1 root     root       9900620 Mar 14  2013 rhel5u3-netboot.img
-rw-r--r--    1 root     root      10537016 Mar 14  2013 rhel5u4-netboot.img
-rw-r--r--    1 root     root      11333732 Mar 14  2013 rhel5u5-netboot.img
-rw-r--r--    1 root     root      11612524 Mar 14  2013 rhel5u6-netboot.img
-rw-r--r--    1 root     root      15006880 Mar 14  2013 rhel5u7-netboot.img
-rw-r--r--    1 root     root      16072024 Mar 14  2013 rhel5u8-netboot.img
-rw-r--r--    1 root     root      16289260 Mar 14  2013 rhel5u9-netboot.img
-rw-r--r--    1 root     root      36020177 Mar 14  2013 rhel6-netboot.img
-rw-r--r--    1 root     root      38134533 Mar 14  2013 rhel6u1-netboot.img
-rw-r--r--    1 root     root      31051305 Mar 14  2013 rhel6u2-netboot.img
-rw-r--r--    1 root     root      32181745 Mar 14  2013 rhel6u3-netboot.img
-rw-r--r--    1 root     root      32791785 Mar 14  2013 rhel6u4-netboot.img
-rw-r--r--    1 root     root      33237149 Dec  3  2013 rhel6u5-netboot.img
-rw-r--r--    1 root     root      33257321 Nov 18 18:37 rhel6u6-netboot.img
-rw-r--r--    1 root     root      39647370 Nov 18 18:38 rhel7-netboot.img
```

Netbooting PowerLinux – TFTP resources
SUSE Linux boot images

- Setup of TFTP resources in `/tftpboot`
- Same for AIX NIM server and Linux DHCP server (for installation via a Linux network install image)

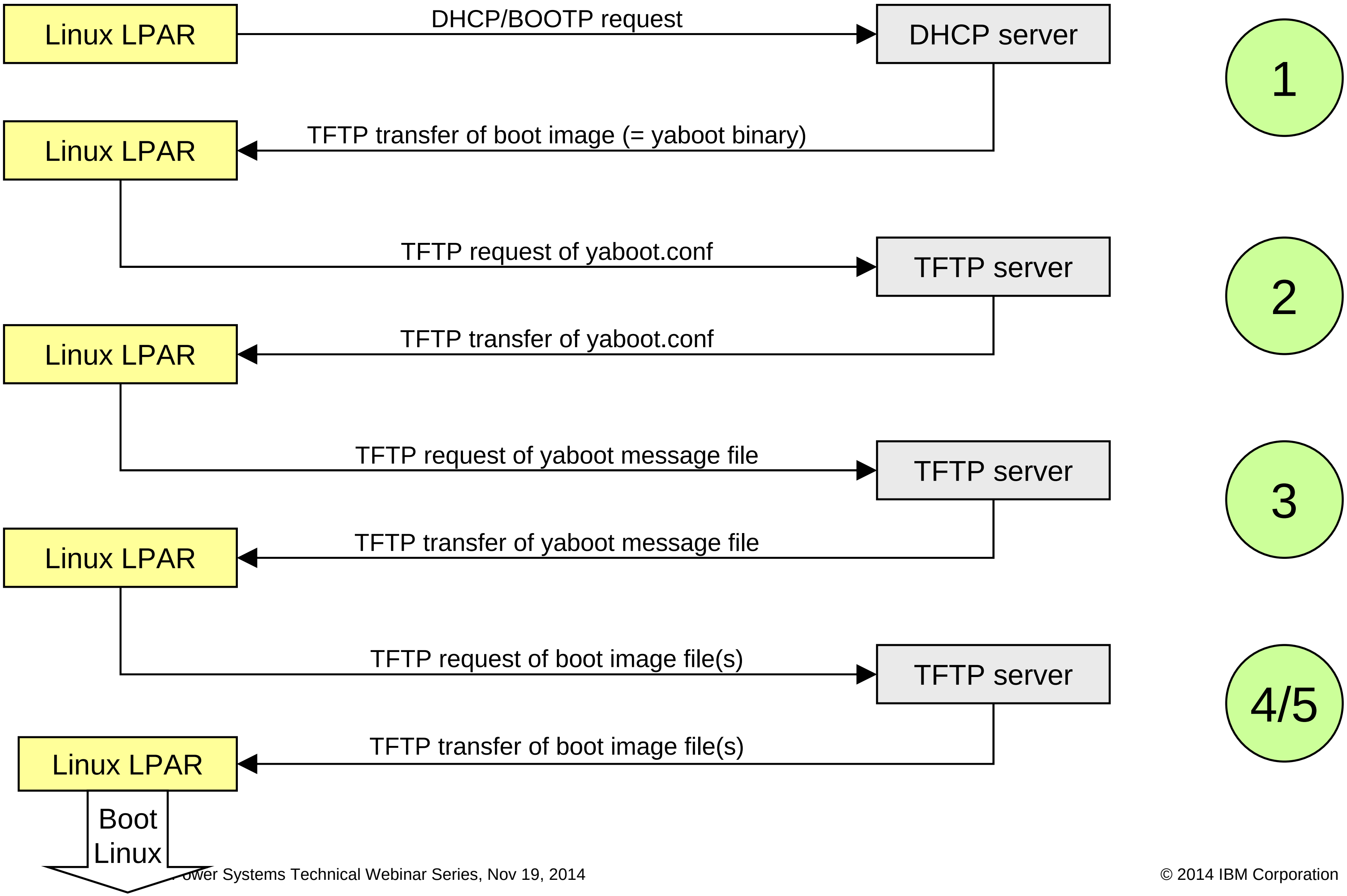
```
root@nim:/tftpboot> ls -l sles*
-rw-r--r--    1 root     root      5714106 Mar 13 20:23 sles9-sp3-install
-rw-r--r--    1 root     root      8087775 Mar 13 20:23 sles10-inst32
-rw-r--r--    1 root     root      8718463 Mar 13 20:23 sles10-inst64
-rw-r--r--    1 root     root      8382855 Mar 13 20:23 sles10-sp1-inst32
-rw-r--r--    1 root     root      9222971 Mar 13 20:23 sles10-sp1-inst64
-rw-r--r--    1 root     root      8717759 Mar 13 20:23 sles10-sp2-inst32
-rw-r--r--    1 root     root      9790331 Mar 13 20:23 sles10-sp2-inst64
-rw-r--r--    1 root     root      9512595 Mar 13 20:23 sles10-sp3-inst32
-rw-r--r--    1 root     root     10682935 Mar 13 20:23 sles10-sp3-inst64
-rw-r--r--    1 root     root     10082439 Mar 13 20:23 sles10-sp4-inst32
-rw-r--r--    1 root     root     11307427 Mar 13 20:23 sles10-sp4-inst64
-rw-r--r--    1 root     root     19809567 Mar 13 20:23 sles11-inst64
-rw-r--r--    1 root     root     23733099 Mar 13 20:23 sles11-sp1-inst64
-rw-r--r--    1 root     root     30475691 Mar 13 20:23 sles11-sp2-inst64
-rw-r--r--    1 root     root     32244849 Mar 13 20:23 sles11-sp3-inst64
```

Netbooting Linux on Power (2/2)

Two images can be used for netbooting:

- Linux network install image
  - Combined image of Linux kernel and initial ramdisk (containing installer)
  - Works up to a size of ~12 MB of the Linux network install image (limit of the Power Open Firmware TFTP buffer size)
- The standard Linux boot loader `yaboot`
  - For Linux network install images larger than 12 MB
  - Two-step boot process:
    1. Boot the yaboot boot loader (size less than 400 kB)
    2. Yaboot then boots the Linux kernel and initial ramdisk
→ thus circumventing the Open Firmware TFTP buffer size limitation

Netbooting yaboot (1/2)


Linux LPAR
DHCP/BOOTP request
DHCP server
1
Linux LPAR
TFTP transfer of boot image (= yaboot binary)
TFTP request of yaboot.conf
TFTP server
2
Linux LPAR
TFTP transfer of yaboot.conf
TFTP request of yaboot message file
TFTP server
3
Linux LPAR
TFTP transfer of yaboot message file
TFTP request of boot image file(s)
TFTP server
4/5
Linux LPAR
TFTP transfer of boot image file(s)
Boot
Linux

Netbooting yaboot (2/2)

- Linux LPAR sends out directed BOOTP or broadcast DHCP request.
- DHCP server answers BOOTP/DHCP request and transfers boot image back to Linux LPAR via TFTP.
- Linux LPAR executes transferred boot image (= yaboot binary).
- Yaboot now requests via TFTP its config file `yaboot.conf`:
  - Depending on Linux distribution client-specific `yaboot.conf` can be located in different locations (i.e., directories).
- TFTP server transfers `yaboot.conf` back to Linux LPAR.
- If “`yaboot.conf`“ contains a statement “`message=yaboot.txt`“ then the yaboot message file is requested via TFTP and transferred back from the TFTP server.
- Linux LPAR now requests the boot image file(s).
- TFTP server transfers boot image files(s) back to Linux LPAR.

Netbooting PowerLinux – AIX NIM bootp setup

```
/etc/bootptab:
# Legend:
#   first field -- hostname (may be full domain name and probably should be)
#   hd -- home directory
#   bf -- bootfile
#   sa -- server IP address to tftp bootfile from
#   gw -- gateways
#   ha -- hardware address
#   ht -- hardware type
#   ip -- host IP address
#   sm -- subnet mask
#   tc -- template host (points to similar host entry)
#   hn -- name switch
#   bs -- boot image size
#   dt -- old style boot switch
js21-5-rhel5:bf=/tftpboot/js21-5-rhel5:ip=10.0.21.52:ht=ethernet:sa=10.0.0.8:sm=255.255.0.0:
js21-6-sles10:bf=/tftpboot/js21-6-sles10:ip=10.0.21.62:ht=ethernet:sa=10.0.0.8:sm=255.255.0.0:
```

```
root@nim:/tftpboot> ls -la js*
lrwxrwxrwx    1 root     system          19 Mar 03 2013  js21-5-rhel5 -> rhel5u8-netboot.img
lrwxrwxrwx    1 root     system          14 Mar 02 2013  js21-6-sles11 -> sles11-sp2-inst64
```

Don’t forget afterwards to activate the changes:

- `refresh –s inetd`

Netbooting PowerLinux – Linux DHCP server

```
[root@ppclinux:/tftpboot]# ls -al
drwxr-xr-x  14 root root      640 Apr 19 00:29 .
drwxr-xr-x  26 root root      888 Apr 16 21:32 ..
lrwxrwxrwx   1 root root       10 Apr  7 13:32 LPAR00001 -> yaboot.RHEL65
drwxr-xr-x   2 root root      104 Apr  7 14:09 RHEL6U5
-rw-r--r--   1 root root     2916 Apr  7 14:40 yaboot.conf
lrwxrwxrwx   1 root root       11 Apr  7 13:51 yaboot.conf-36-9a-c0-00-40-0b -> yaboot.conf
-rw-r--r--   1 root root   263760 Apr  7 13:27 yaboot.RHEL65
-rw-r--r--   1 root root      213 Apr  7 13:58 yaboot.txt

[root@ppclinux:/tftpboot]# ls –al RHEL6U5
./RHEL6-U5:
-rw-r--r--   1 root root 28205108 Apr  7 14:09 initrd.img
-rwxr-xr-x   1 root root 17020184 Apr  7 14:09 vmlinuz
```

Netbooting PowerLinux – Linux DHCP server

/etc/dhcpd.conf:

```
option domain-name "munich.de.ibm.com";
option domain-name-servers 10.0.0.8;
option routers 10.0.0.1;
option ntp-servers 10.0.0.8;
ddns-update-style none;
ignore unknown-clients;
allow bootp;
subnet 10.0.0.0 netmask 255.255.0.0 {
  range 10.0.56.230 10.0.56.250;
  default-lease-time 86400;
  max-lease-time 604800;
}
host LPAR00001 {
  hardware ethernet 36:9A:C0:00:40:0B;
  filename "LPAR00001";
  fixed-address 10.0.56.4;
  next-server 10.0.0.51;
}
```

Netbooting PowerLinux – Linux DHCP server

```
[root@dhcp-server:/tftpboot]# cat yaboot.conf
## This yaboot.conf is for netbooting different Linux distributions

message=yaboot.txt

timeout=100

default=sles11-sp3

image[64bit]=sles11-sp3-inst64
    label=sles11-sp3
    append="quiet sysrq=1 insmod=sym53c8xx insmod=ipr install=nfs://10.0.0.51/export/PPC/SLES11-SP3"

image[64bit]=sles11-sp2-inst64
    label=sles11
    append="quiet sysrq=1 insmod=sym53c8xx insmod=ipr install=nfs://10.0.0.51/export/PPC/SLES11-SP2"

image=/RHEL6-U5/vmlinuz
    label=rhel6-u5
    initrd=/RHEL6-U5/initrd.img
    append="ks=nfs:10.0.0.51:/export/RHEL_Kickstart/kickstart.cfg ksdevice=eth0"
    read-only

. . . .
```

Netbooting PowerLinux – Linux DHCP server

To successfully netboot RHEL 5/6/7

- You need to use the yaboot that comes with RHEL 5/6/7

To successfully netboot SLES 11

- You need to use the yaboot that comes with SLES 11

Virtual network options for Linux clients

Complex network setup example

- **Setup of four different networks**
  - VLAN 110: Management network
  - VLAN 112: Install network
  - VLAN 750: SAP/DB production network
  - VLAN 751: SAP/DB development network

- **Each LPAR has connections to the following VLANs:**
  - VLAN 110
  - VLAN 112
  - VLAN 750 or VLAN 751

Next: Comparison of Linux bonding vs. SEA failover setup

Bonding setup

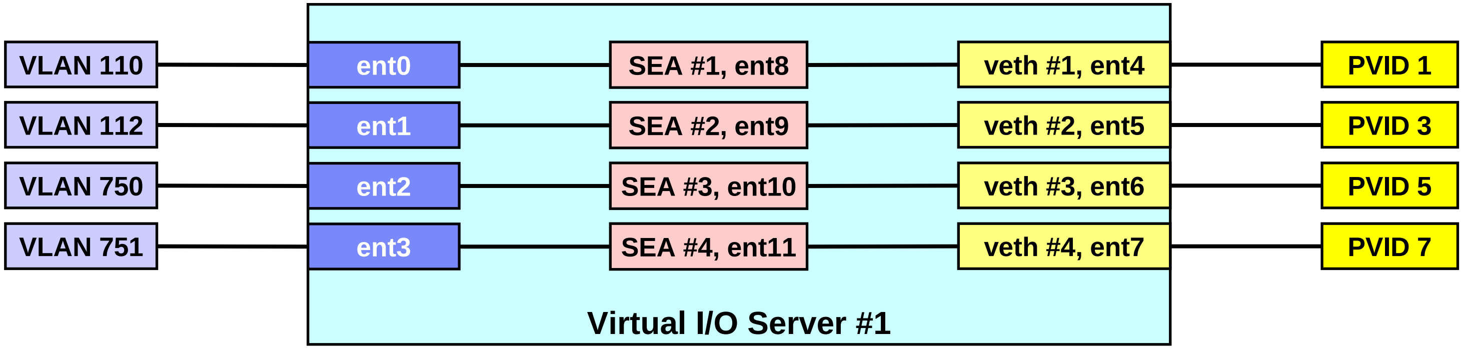
How does bonding work ?

- Ethernet bonding on the client works like Etherchannel in NIB (**N**etwork **I**nterface **B**ackup) mode with AIX.

- **Difference:**
  - Linux bonding uses ARP ping instead of ICMP ping to check for network path availability.

- A ping target outside of the machine is required, typically the gateway for the specific network is used.

Virtual I/O Server setup for bonding



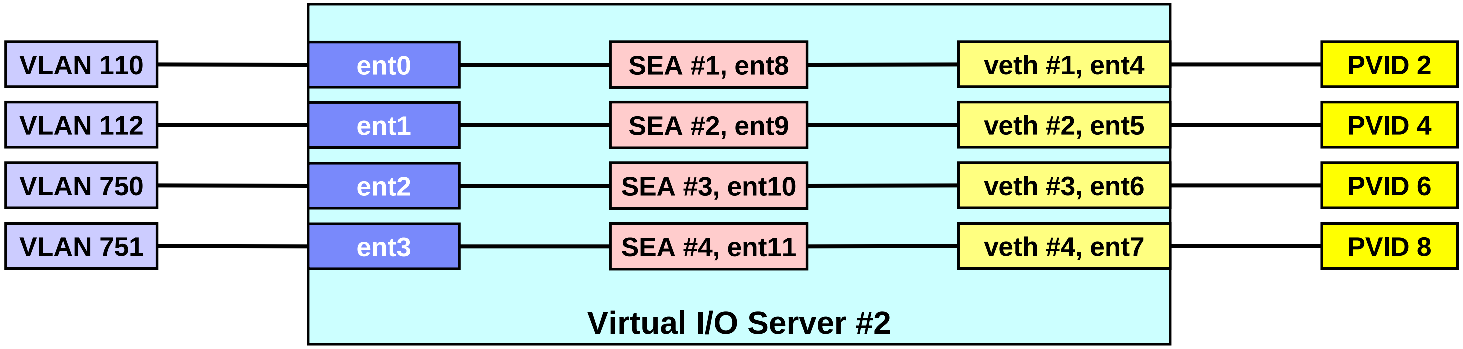
VLAN 110 — ent0 — SEA #1, ent8 — veth #1, ent4 — PVID 1

VLAN 112 — ent1 — SEA #2, ent9 — veth #2, ent5 — PVID 3

VLAN 750 — ent2 — SEA #3, ent10 — veth #3, ent6 — PVID 5

VLAN 751 — ent3 — SEA #4, ent11 — veth #4, ent7 — PVID 7

Virtual I/O Server #1



VLAN 110 — ent0 — SEA #1, ent8 — veth #1, ent4 — PVID 2

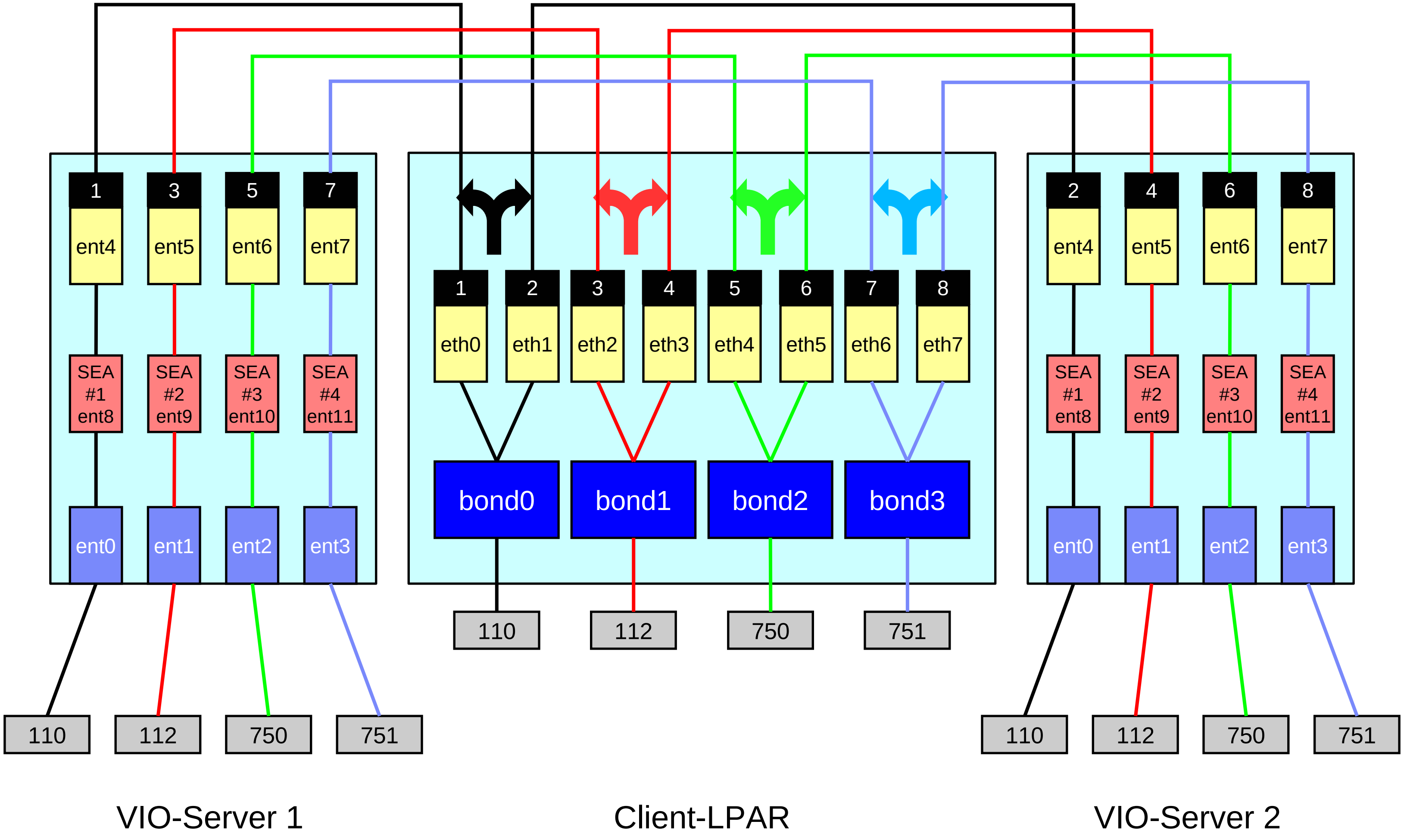
VLAN 112 — ent1 — SEA #2, ent9 — veth #2, ent5 — PVID 4

VLAN 750 — ent2 — SEA #3, ent10 — veth #3, ent6 — PVID 6

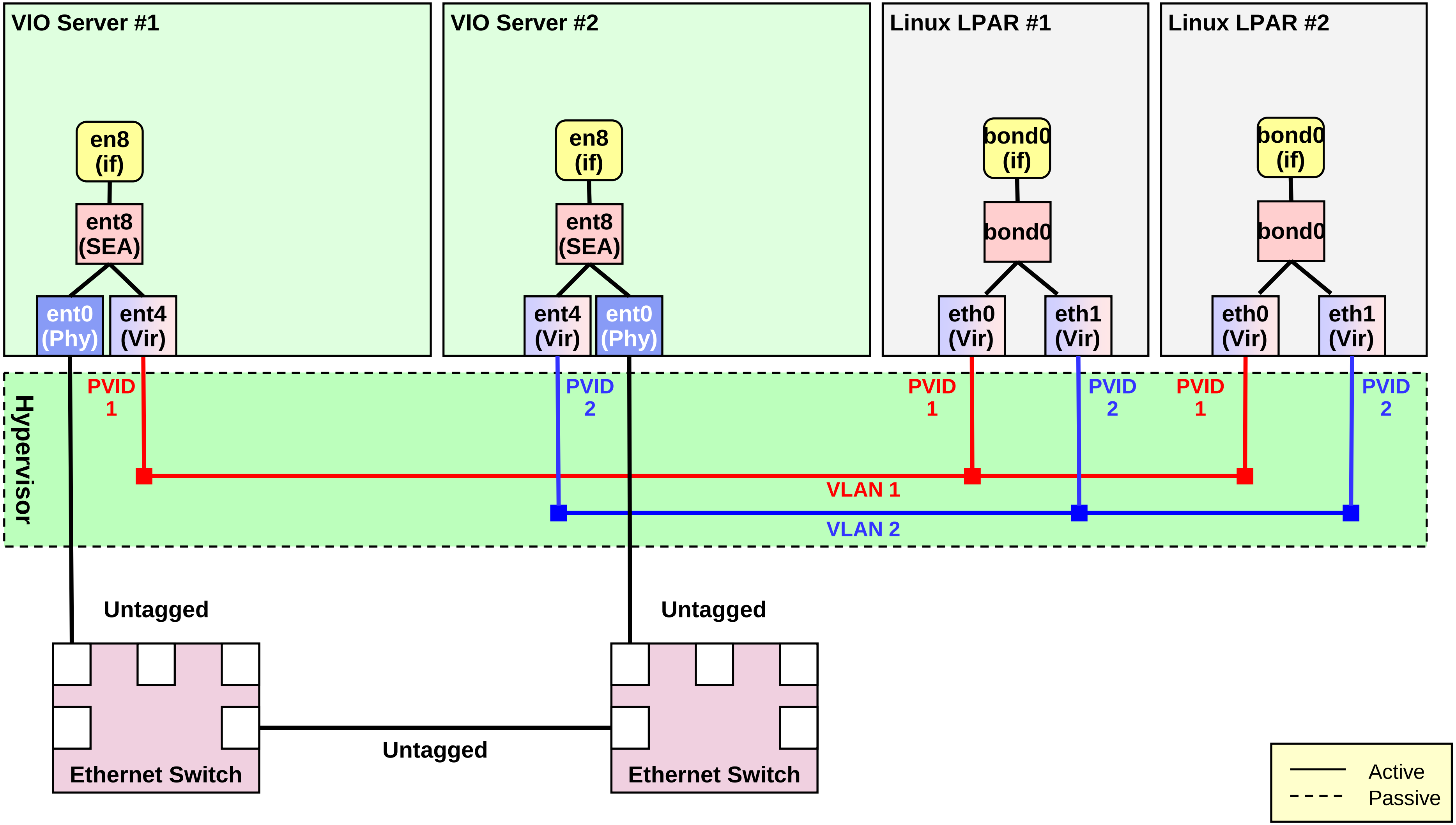
VLAN 751 — ent3 — SEA #4, ent11 — veth #4, ent7 — PVID 8

Virtual I/O Server #2

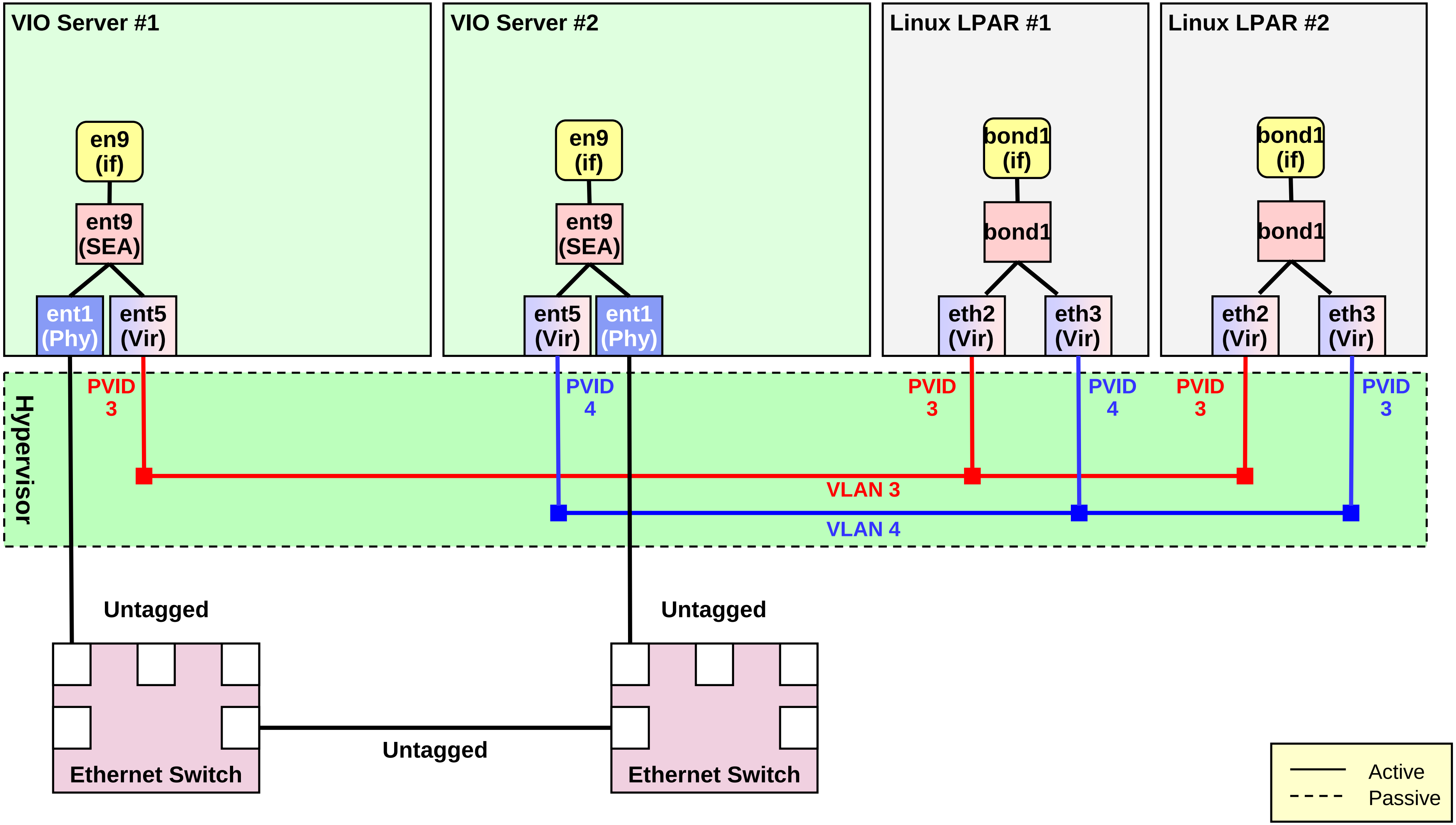
Linux client setup with bonding



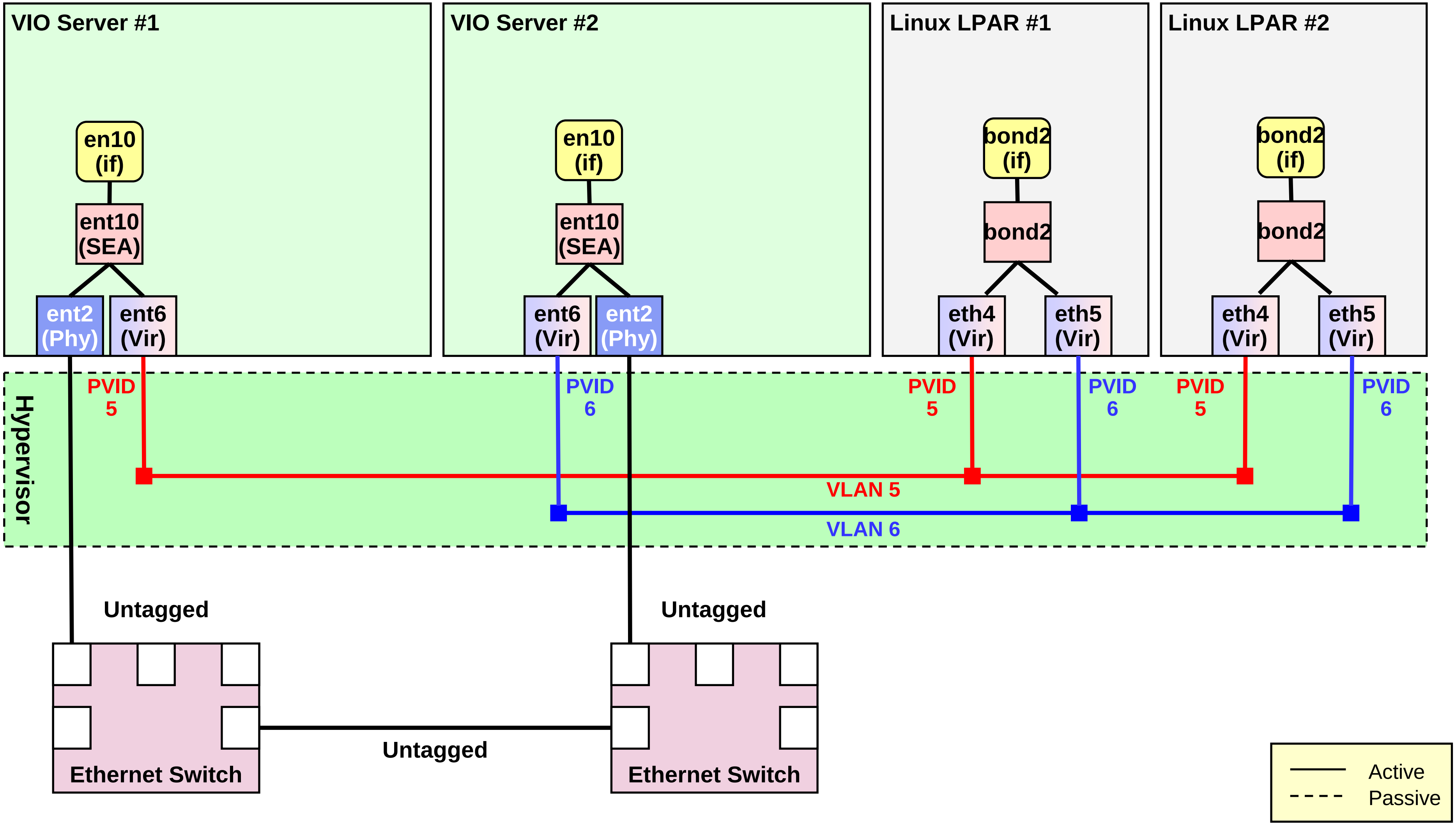
Virtual ethernet – VLAN 110 – bonding setup



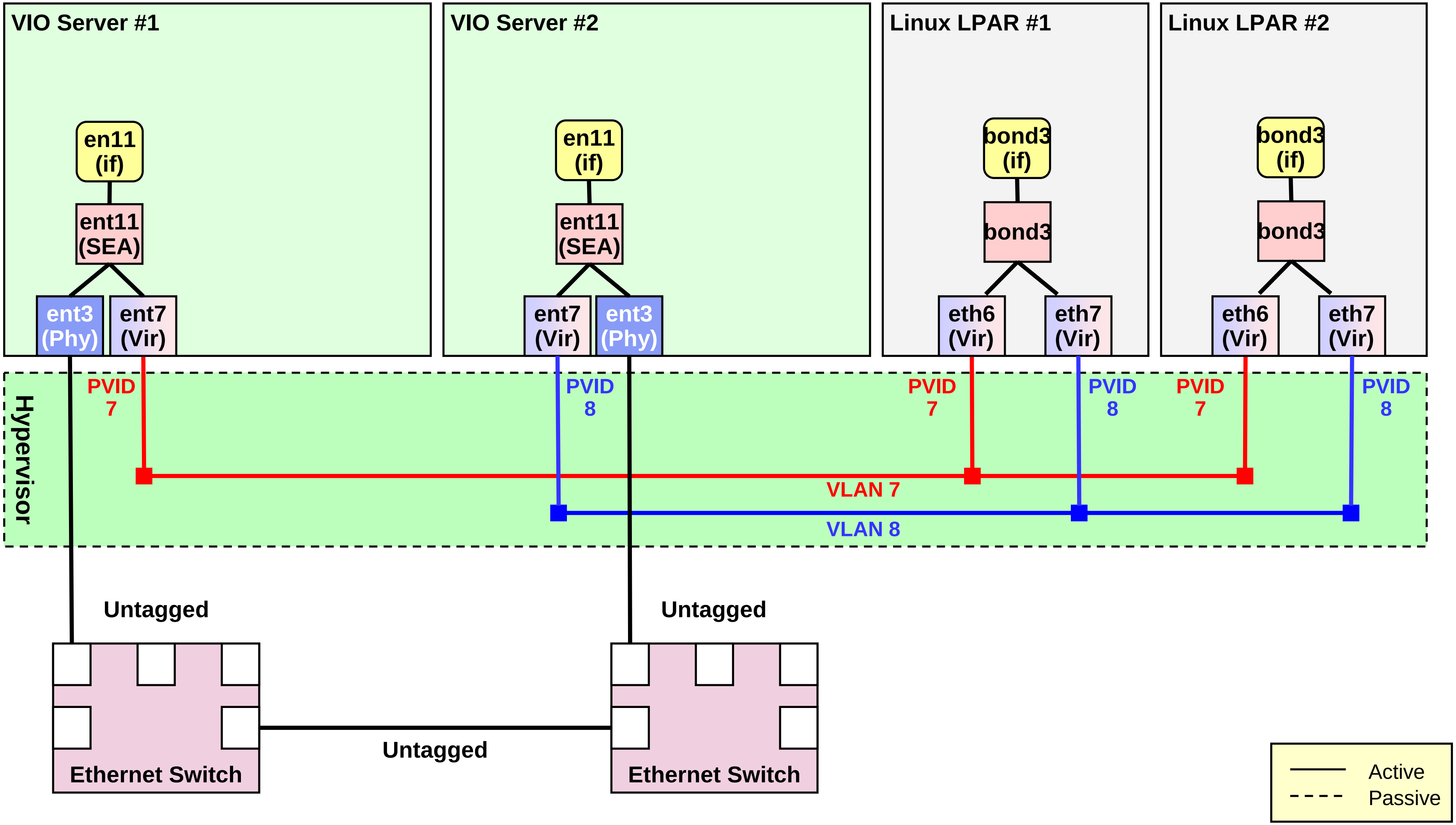
Virtual ethernet – VLAN 112 – bonding setup



Virtual ethernet – VLAN 750 – bonding setup



Virtual ethernet – VLAN 751 – bonding setup



Bonding configuration details

- /etc/modprobe.conf:

```
alias bond0 bonding
options bond0 mode=active_backup arp_interval=2000 arp_ip_target=X.X.X.X \
        arp_validate=all
....
```

- /etc/sysconfig/network-scripts/ifcfg-bond0
- /etc/sysconfig/network-scripts/ifcfg-eth0
- /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE=bond0
BOOTPROTO=static
IPADDR=10.186.38.163
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
TYPE=Ethernet
```

```
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
TYPE=Ethernet
```

Bonding summary (1/2)

- Ethernet bonding on the client works like Etherchannel in NIB (network interface backup) mode with AIX
- Linux bonding uses ARP ping instead of ICMP ping to check for network path availability
- A ping target outside of the machine is required, typically the gateway for the specific network is used
- Requires a feature (`arp_validate`) only present since Linux kernel v2.6.19
  - Being backported to Linux distributions
- Setup works for
  - Red Hat Enterprise Linux 4 Update 6 and higher
  - Red Hat Enterprise Linux 5 Update 1 and higher
  - Red Hat Enterprise Linux 6 and higher
  - SUSE Linux Enterprise Server 9 SP 4
  - SUSE Linux Enterprise Server 10 SP 2 and higher
  - SUSE Linux Enterprise Server 11 and higher

Bonding summary (2/2)

- For IVE (EHEA) adapter bonding with link detection can work with new option:
  - `prop_carrier_state` (propagates carrier state of physical port to stack)

- VLAN tagging would be possible but more tedious to set up properly!

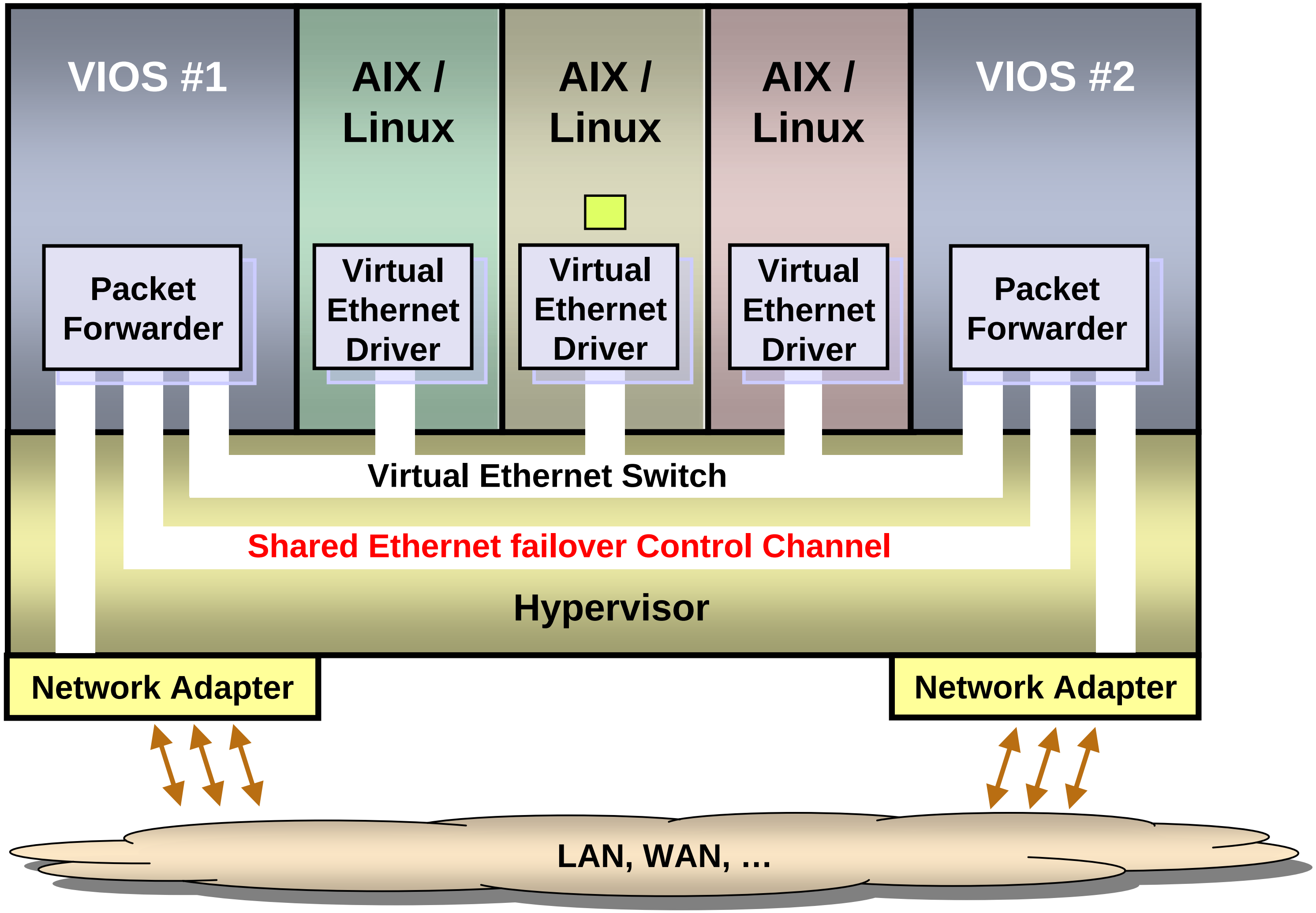
- Bonding works fine with physical network adapters

  ➔ this is what you use for instance for PowerKVM hosts to increase availability and throughput

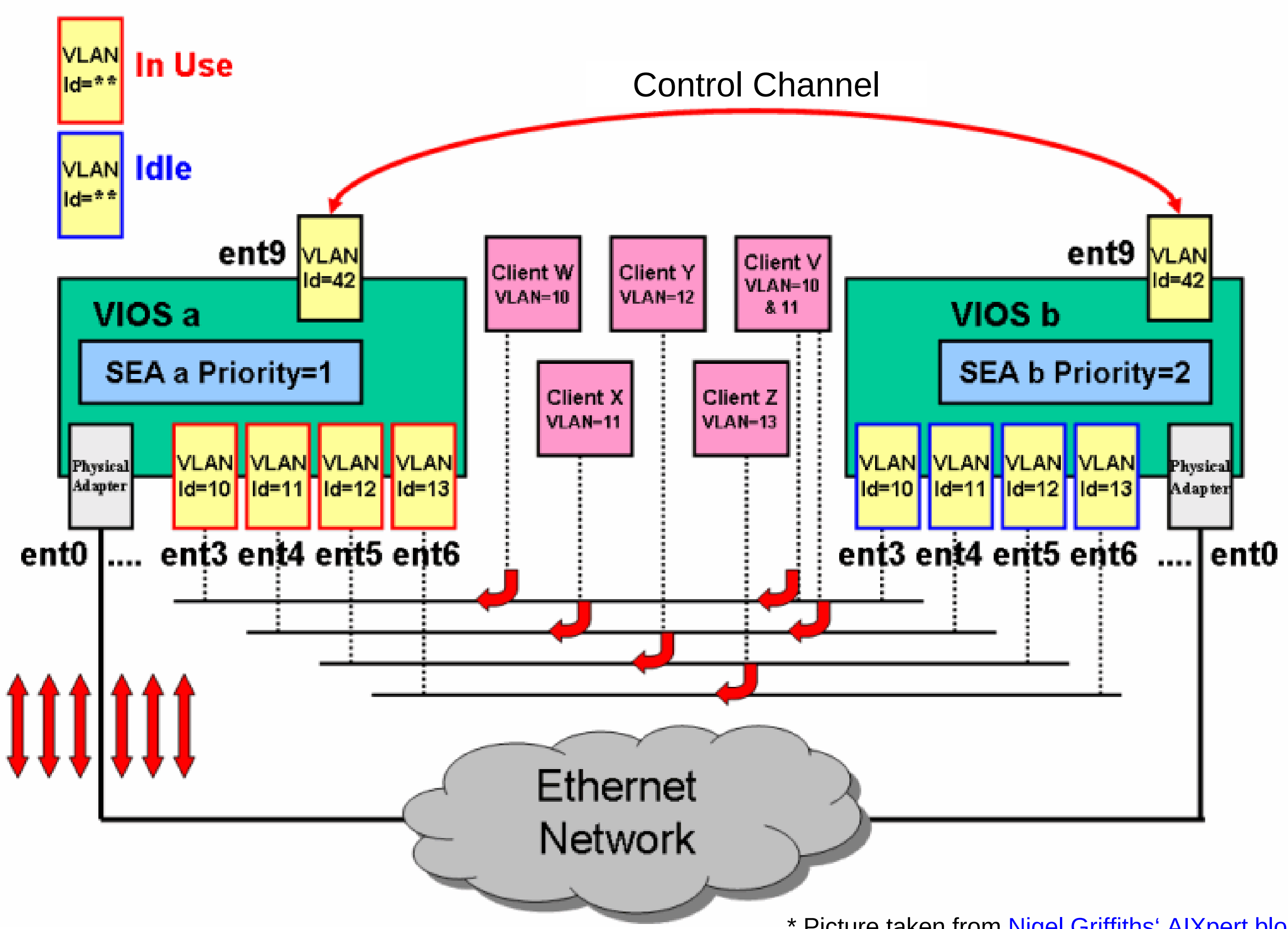
Recommendation: For VIOS setups use SEA failover (with load sharing) !

Shared Ethernet Adapter (SEA) failover (and load sharing) setup

How does SEA failover work ?

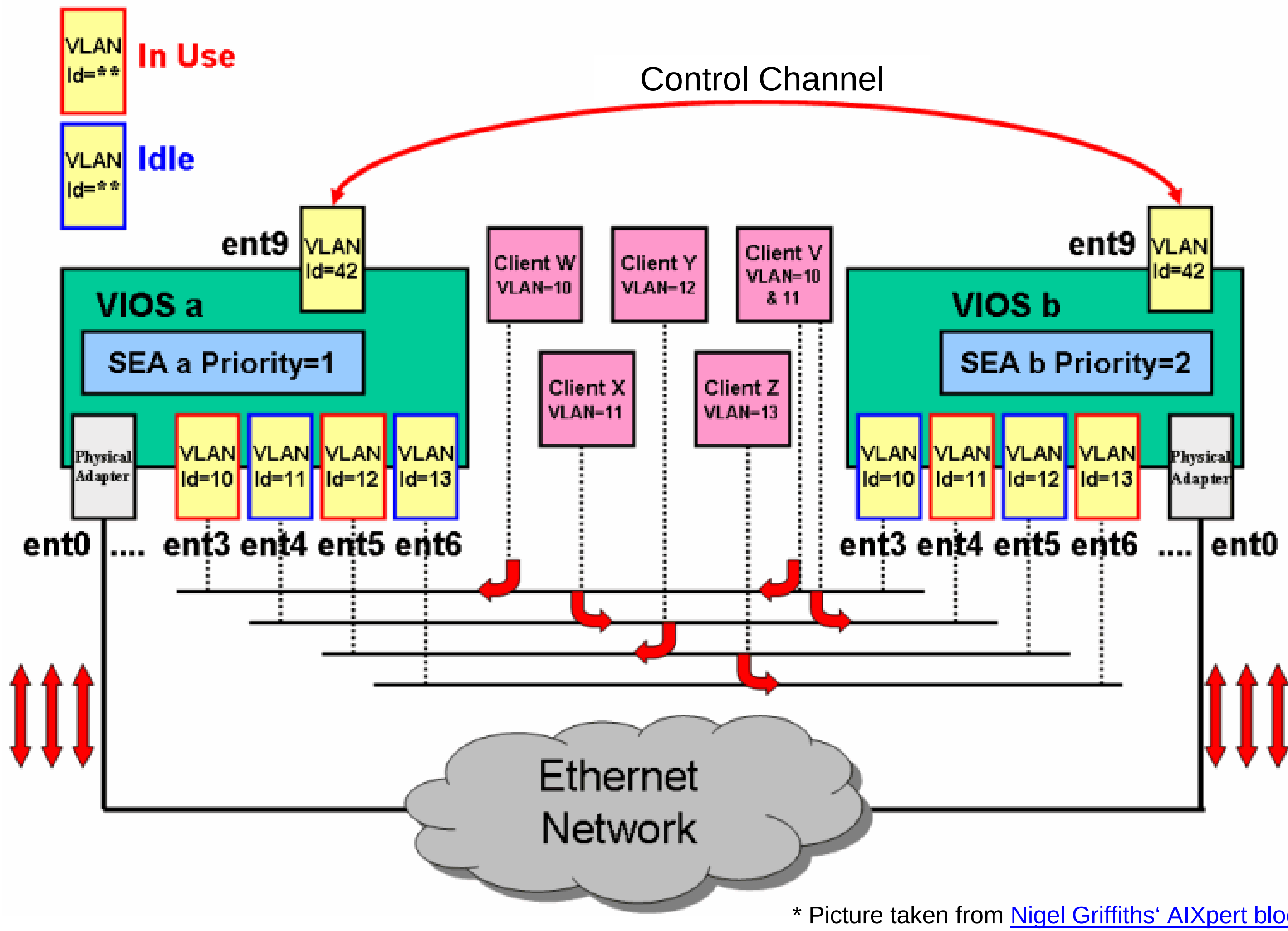


Shared Ethernet Adapter failover



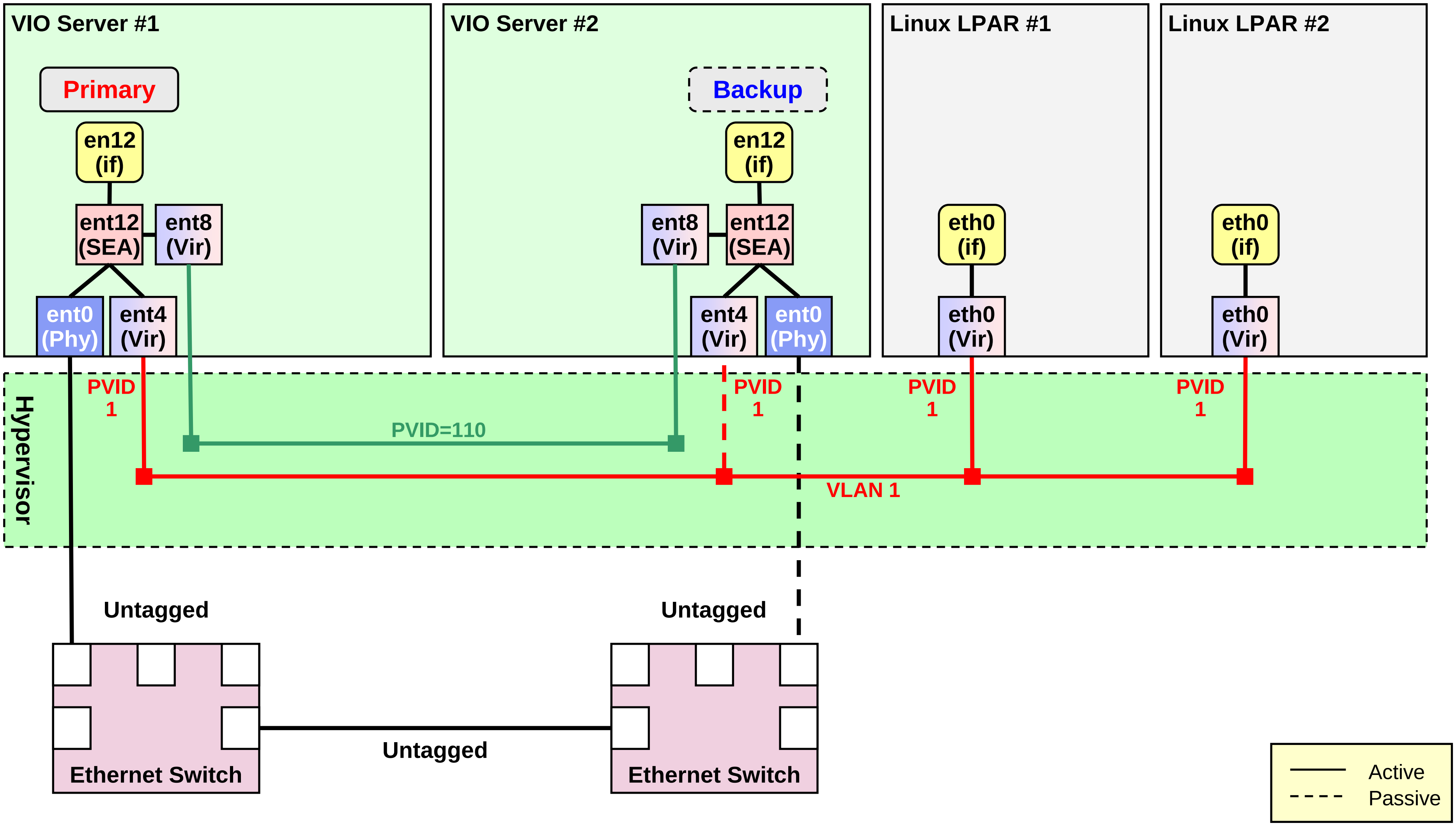
* Picture taken from Nigel Griffiths‘ AIXpert blog on this topic.

Shared Ethernet Adapter failover with load sharing

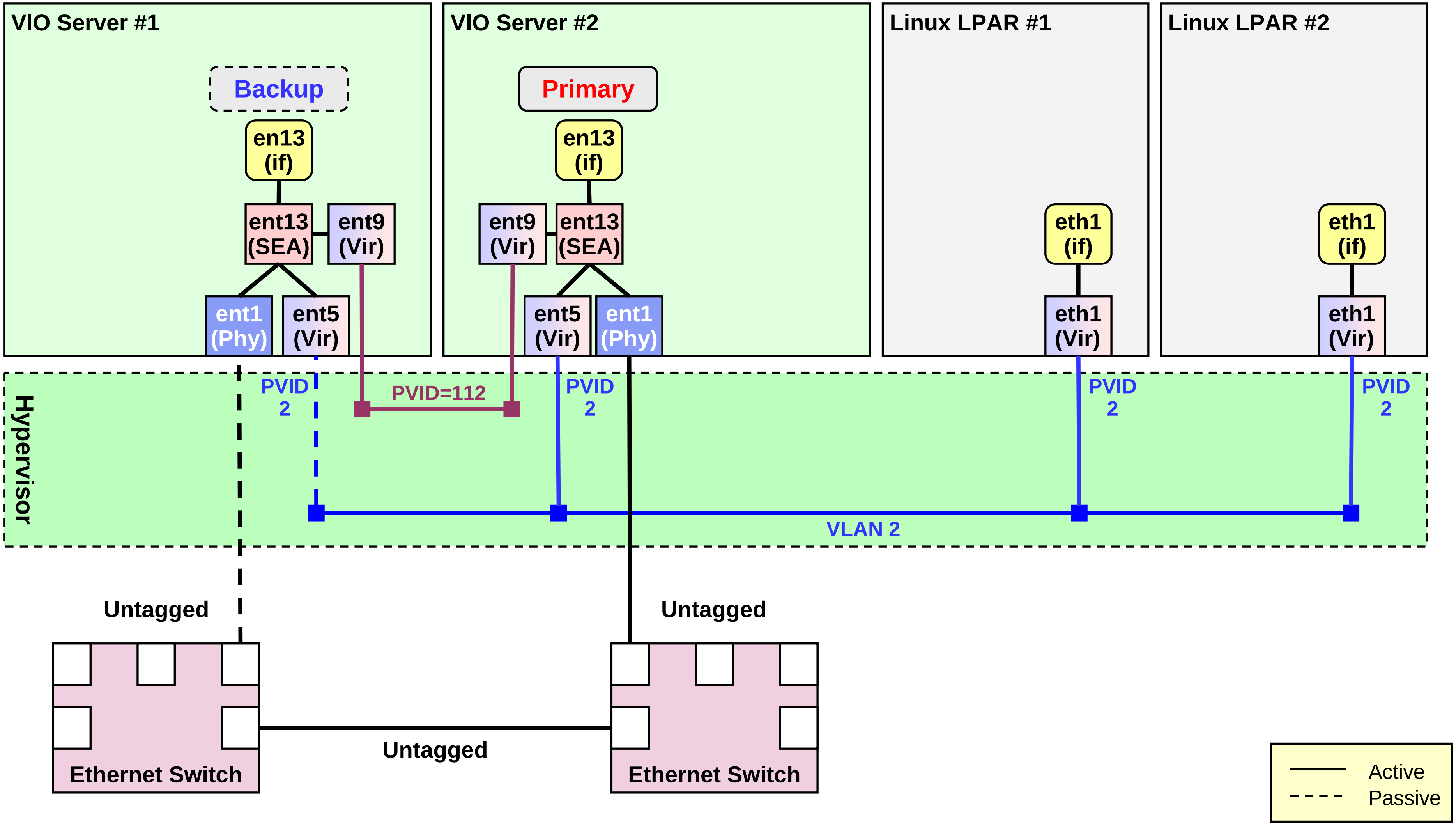


* Picture taken from Nigel Griffiths' AIXpert blog on this topic.

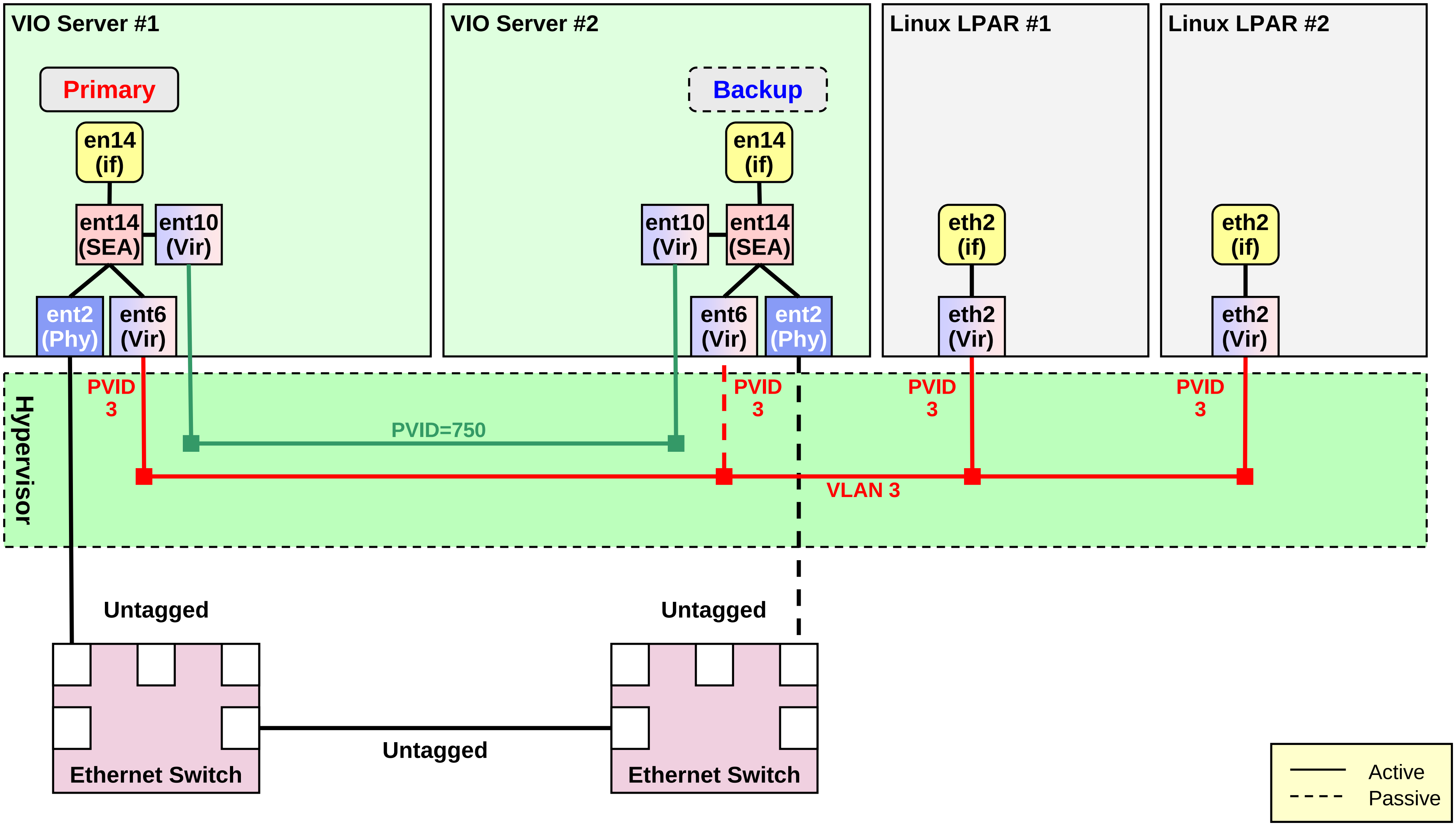
Virtual ethernet – VLAN 110 – SEA failover



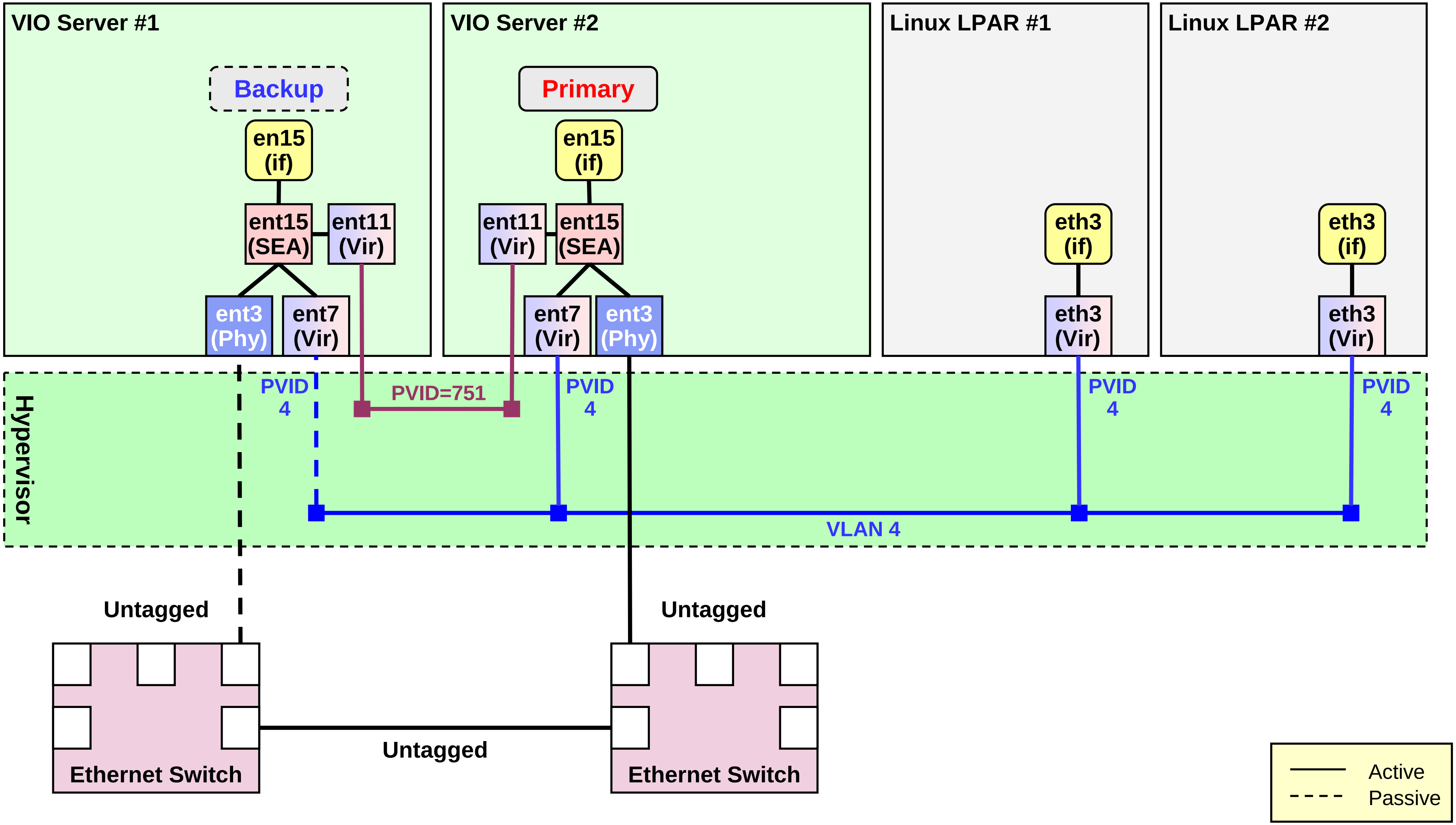
Virtual ethernet – VLAN 112 – SEA failover



Virtual ethernet – VLAN 750 – SEA failover



Virtual ethernet – VLAN 751 – SEA failover



SEA failover (and load sharing) remarks

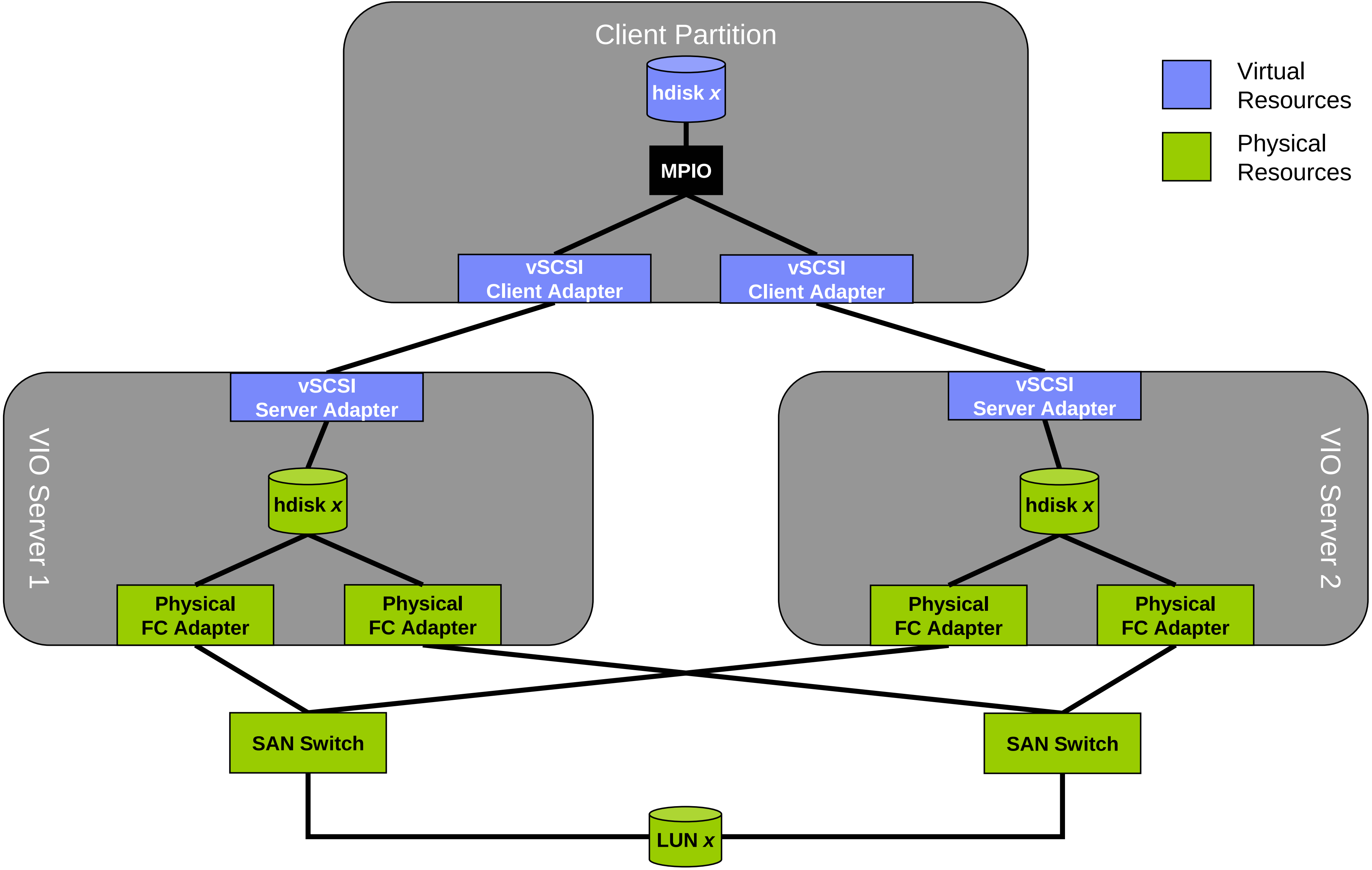
- VLAN tagging can easily be implemented
- For VLAN tagging the SEA control channels for the examples would have a different PVID as the VLAN ID would then be assigned to the virtual adapter PVID.
- Works with **any** Linux distribution and level!
- SEA failover setup with load sharing requires at least VIOS version 2.2.1.0.
- More details here:
  - Nigel Griffiths‘ AIXpert blog on this topic.

Network summary

- Network configuration with SEA failover/load sharing is very simple for the client and more complex on the VIOS side and works reliable.
- Network configuration using bonding on the client is more complex and simple on the VIOS side, however, it requires a recent Linux kernel feature.
- Static load balancing can be provided by both while SEA with load sharing does this automatically.
- Both variants do not require user intervention after single VIOS failure or reboot.
- Recommended setup variant however is SEA failover with load sharing.

Virtual storage options for Linux clients

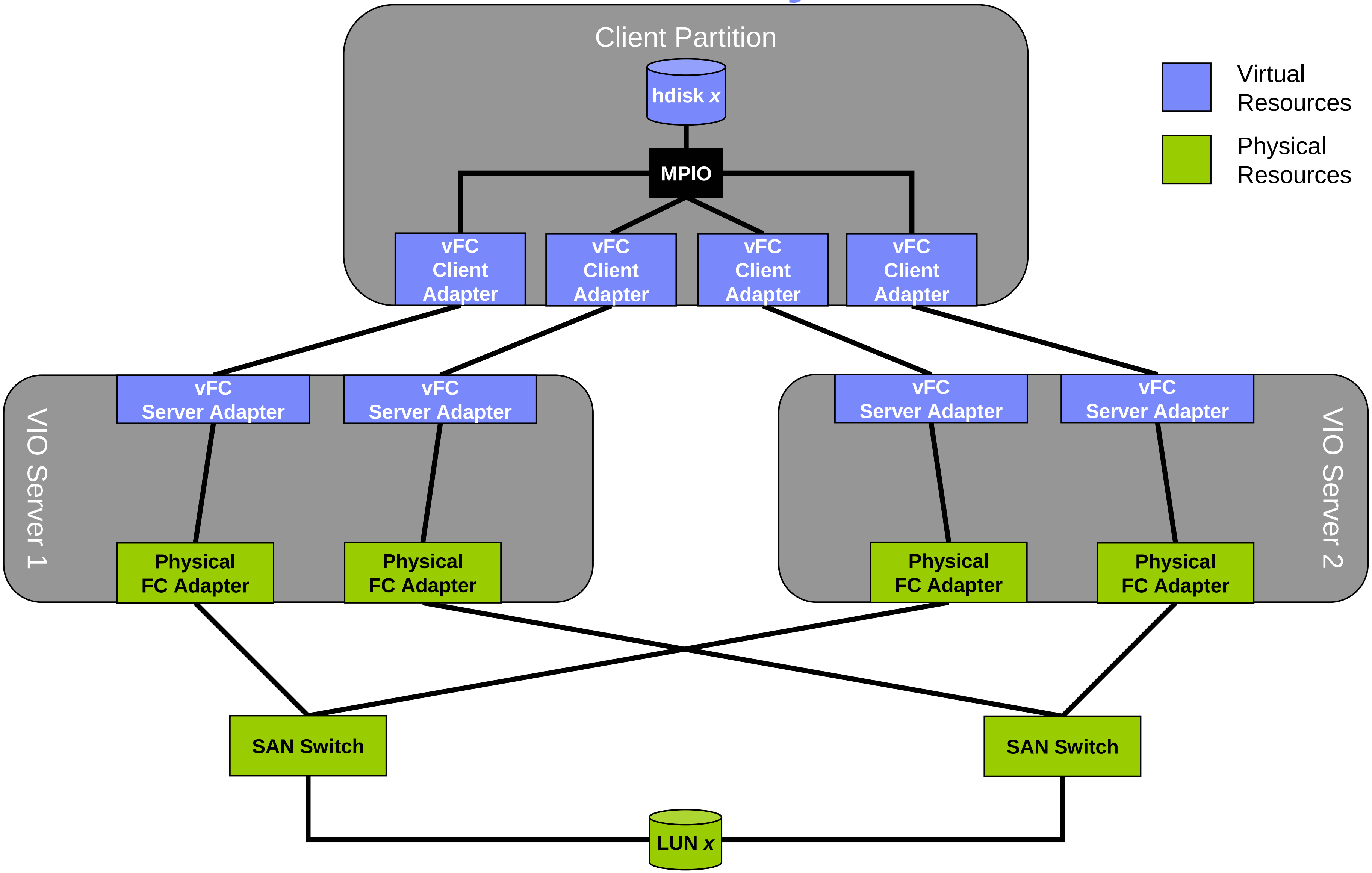
Dual VIOS Server Redundancy: VSCSI



🟦 Virtual Resources

🟩 Physical Resources

Dual VIOS Server Redundancy: NPIV/VFC



🟦 Virtual Resources

🟩 Physical Resources

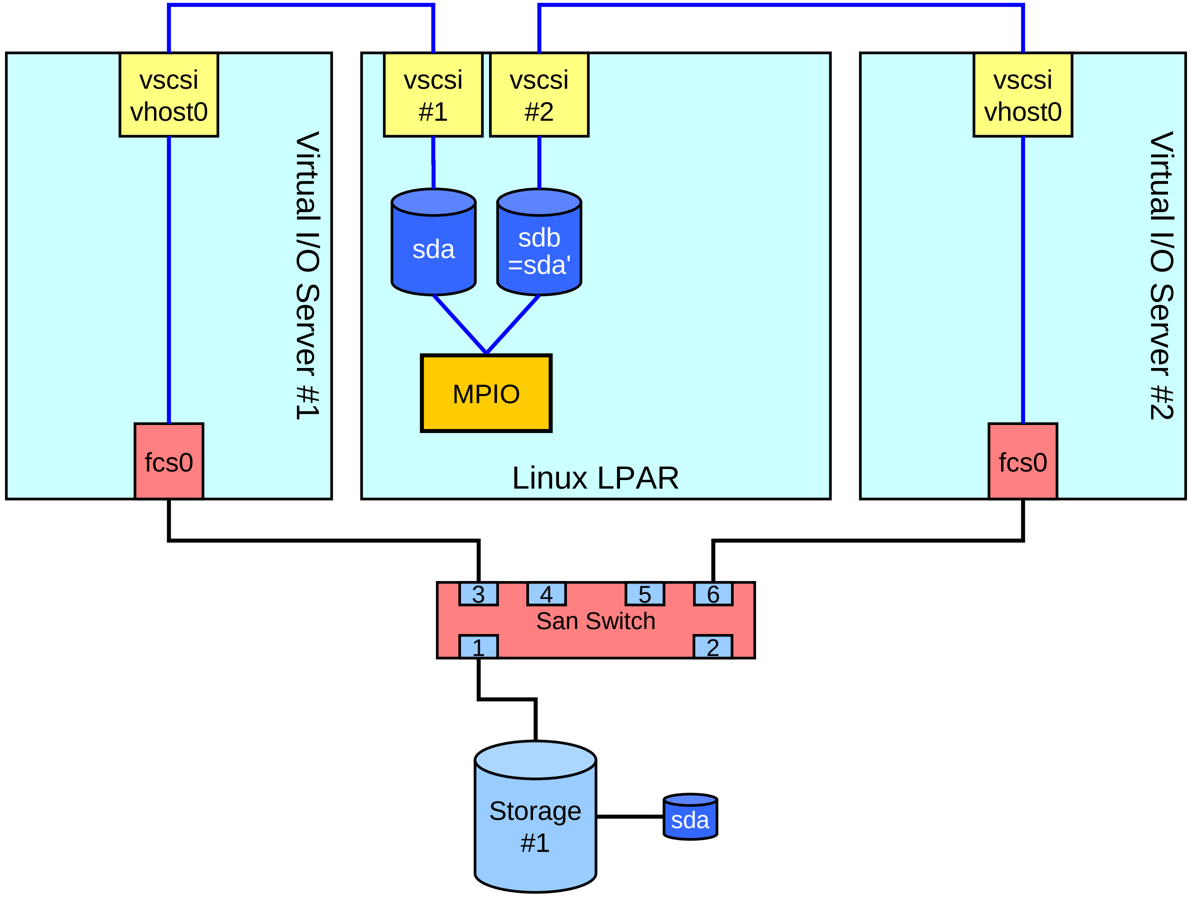
Power Systems

IBM

Multipath I/O with Linux on Power

Simple Multipath I/O example with Linux on Power

Single LUN for OS/data provided by each VIOS from same storage subsystem



Multipath I/O (MPIO) on Linux on Power

- The module `dm_multipath` must be loaded to detect multipath devices in the system.
- The round-robin algorithm is used for load balancing, making the I/O operations to be split among the paths.
- Multipath devices are named as `mpath0`, `mpath1`, and so on.
- Although there are no special considerations to implement a multipath solution on Linux on Power, certain requirements must be observed:
  - The **`ibmvscsic`** driver must be loaded when using virtual SCSI.
  - The **`ibmvfc`** driver must be loaded when using virtual Fibre Channel.
  - The **`dm_multipath`** module must be loaded.
  - The **`/etc/multipath.conf`** must be edited accordingly.
  - The **`multipathd`** daemon must be started.
  - The **`multipath`** command must be used for tracing.

Configuring MPIO on Linux on Power is identical to Linux on x86 !

Example for /etc/multipath.conf file

```
    defaults {
             find_multipaths         yes
             user_friendly_names     yes
    }

    blacklist {
    }

    multipaths {
             multipath {
                         wwid "3600a0b80002644360000bee053b57897"
                         alias lun_rhel65
             }
             multipath {
                         wwid "3600a0b80002644040000a9f952b1a3c3"
                         alias lun_rhel70
             }
             multipath {
                         wwid "3600a0b80002644360000bef553bcf790"
                         alias lun_fedora20
             }
             multipath {
                         wwid "3600a0b80002644040000aaac53b56d22"
                         alias lun_sles11sp3
             }
             multipath {
                         wwid "3600a0b80002644360000c0575421a23a"
                         alias lun_ubuntu1404
             }
    }
```

VIOS VSCSI settings for MPIO for AIX and Linux clients

AIX only

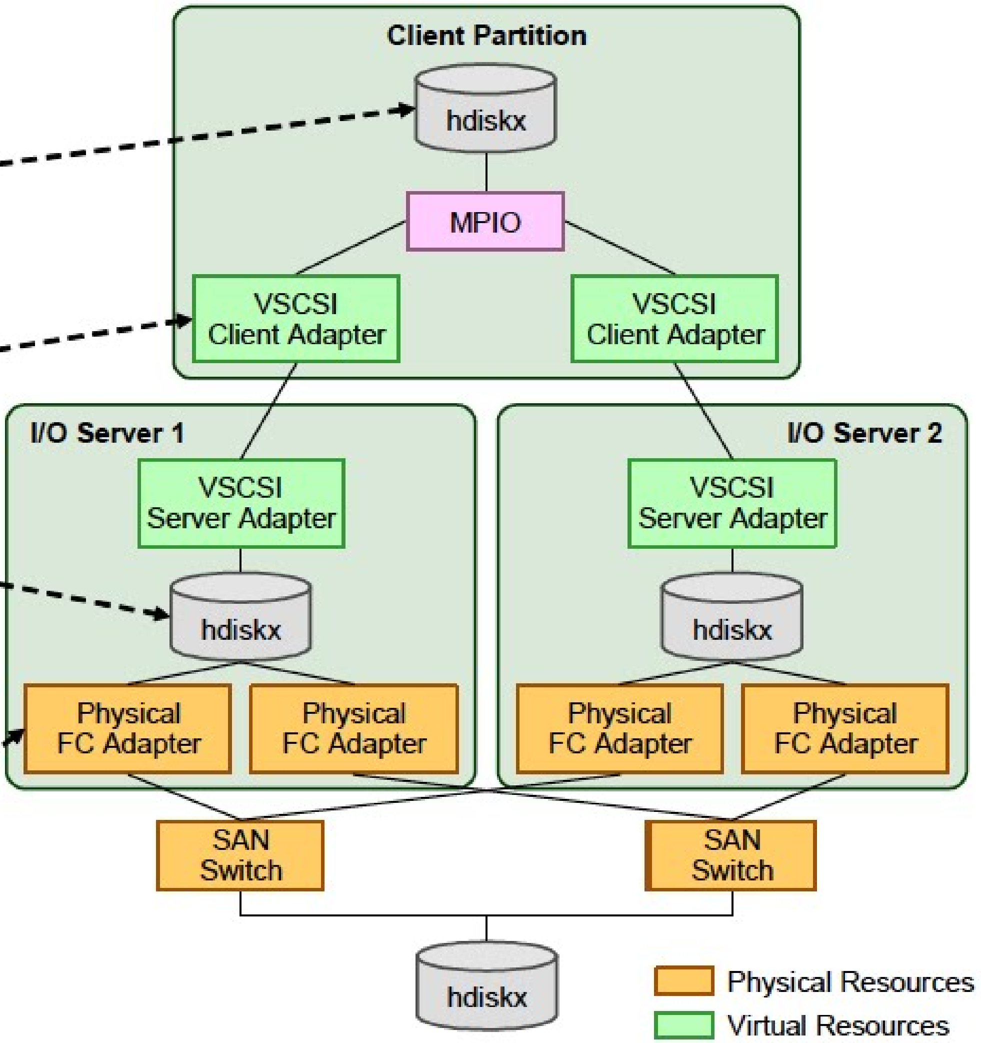
hdisk devices in client
- algorithm=failover
- reserve_policy=no_reserve
- hcheck_mode=nonactive
- hcheck_interval=60
- queue_depth=xxx

vscsi devices in client
- vscsi_path_to=30
- vscsi_err_recov=fast_fail

AIX & Linux

hdisk devices on VIOS
- algorithm=load_balance
- reserve_policy=no_reserve
- hcheck_mode=nonactive
- hcheck_interval=60

fscsi devices on VIOS
- dyntrk=yes
- fc_err_recov=fast_fail



VIOS VSCSI settings for MPIO for Linux clients

Equivalent to AIX parameter "vscsi_path_to":

- Linux will time every command that gets sent to a VSCSI disk and enter error recovery if a command times out, first attempting to issue aborts and ultimately breaking the CRQ as a last resort.
- You can tune the read/write timeout per disk via sysfs.
- The following will set the read write timeout to 30 seconds.

```
echo 30 > /sys/block/sda/device/timeout
```

Linux equivalent to AIX parameter "vscsi_err_recov":

- The recommendation is to enable fast_fail.
- It is enabled by default in the Linux VSCSI client driver.
- You can force it off via the `fast_fail` module parameter if you want it disabled for some reason.

Lab recommendation:

- We have already tuned the default settings for VSCSI, so we wouldn't expect customers to typically need to do anything here.

Virtual adapter/device settings for Linux clients

- You can find all settable parameters for virtual SCSI and virtual Fibre Channel adapters with these commands:
  - Virtual SCSI: `modinfo ibmvscsic`
  - Virtual Fibre Channel: `modinfo ibmvfc`

- Each block device has its individual “`queue_depth`” and “`nr_requests`” parameter:

```
# find /sys -name nr_requests | grep sd
/sys/devices/vio/30000002/host0/target0:0:1/0:0:1:0/block/sda/queue/nr_requests

# find /sys -name queue_depth
/sys/devices/vio/30000002/host0/target0:0:1/0:0:1:0/queue_depth
```

- The default value on all Linux versions (SLES 11, RHEL 6, Debian, etc.) is:

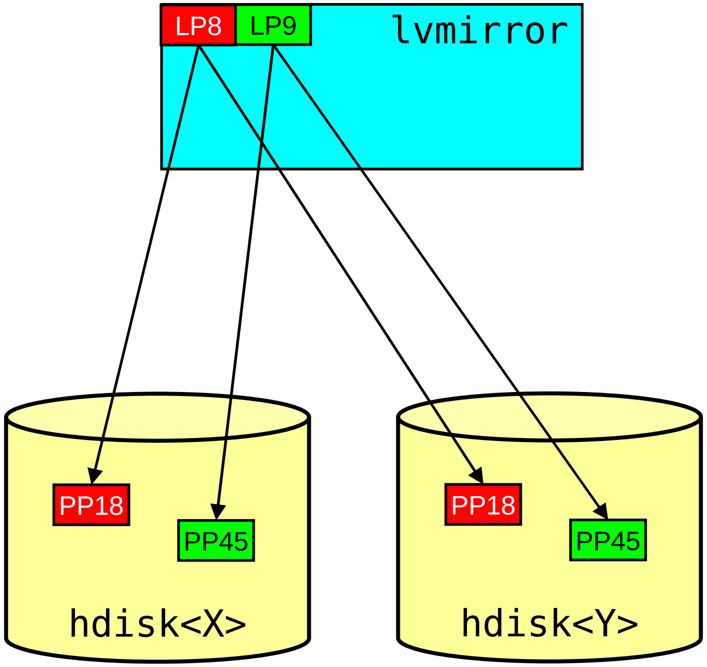
  `nr_requests = 128`

  `queue_depth = 16`

Software RAID with Linux

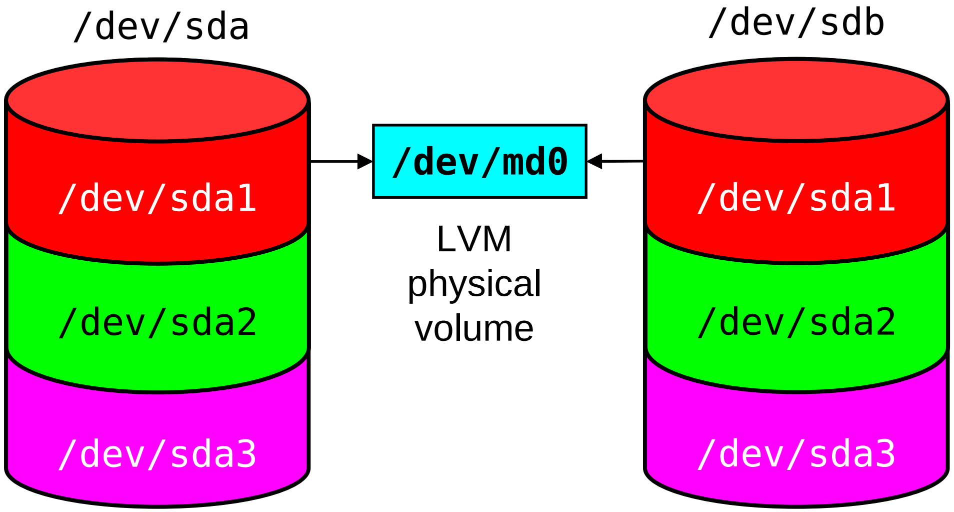
RAID1 comparison between AIX and Linux

AIX Logical Volume Manager

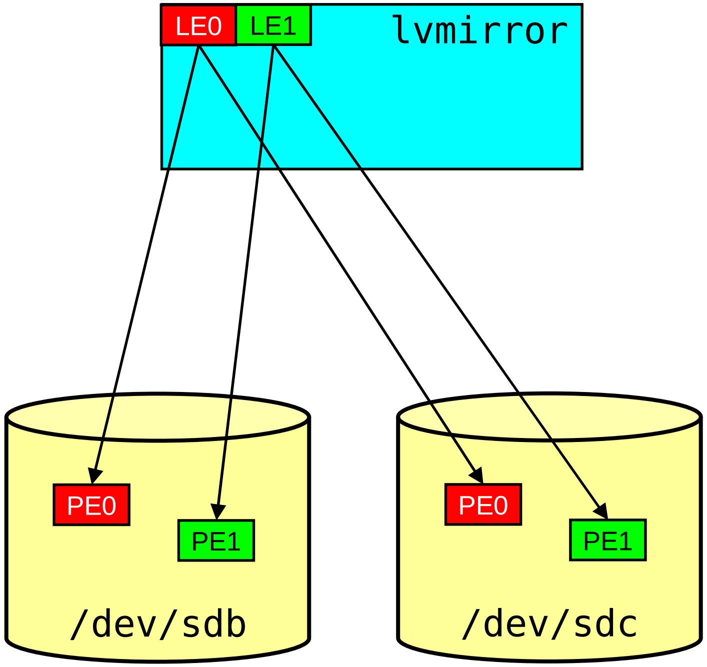


LP = Logical Partition
PP = Physical Partition

Linux MD Devices + LVM1/2



Linux LVM2 (recent versions of RHEL and SLES)
→ not recommended for /boot file system (RHEL only)

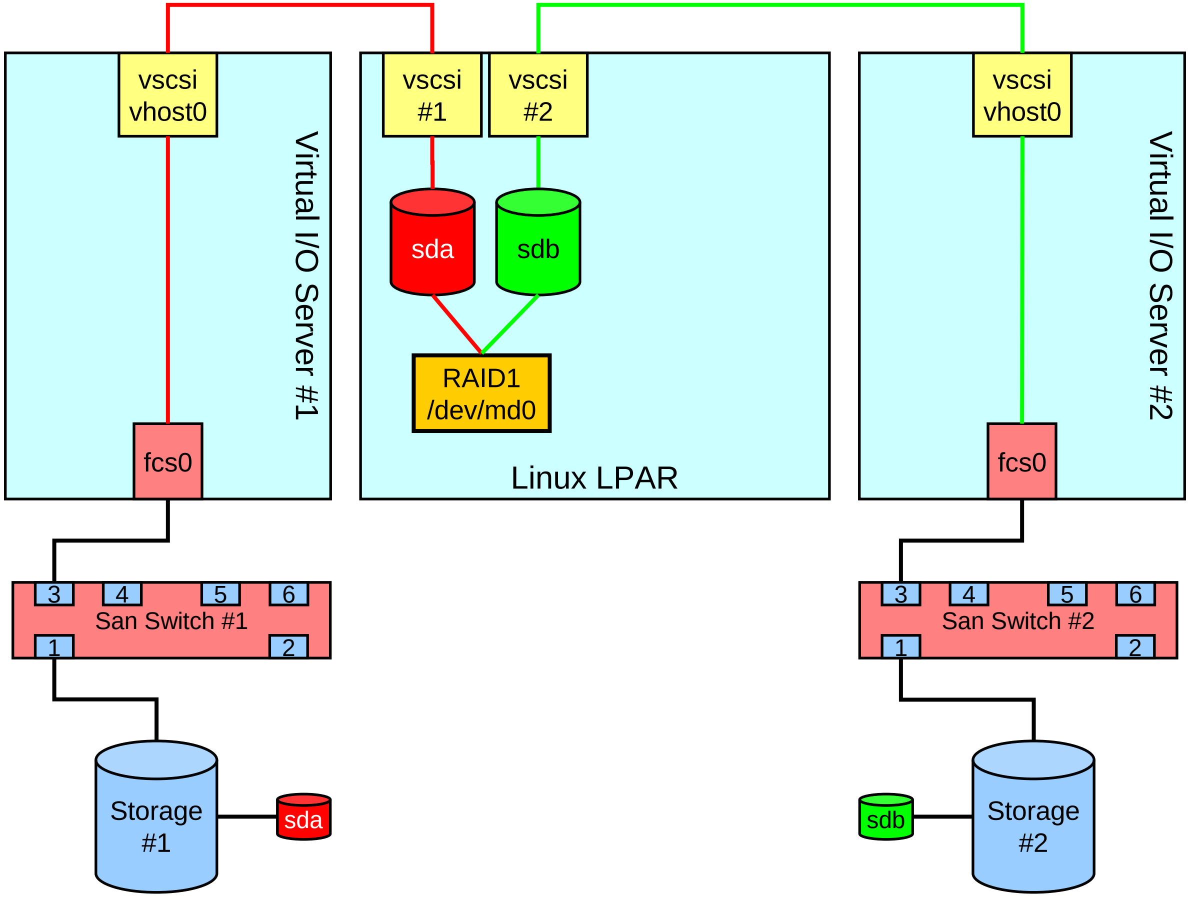


```
pvcreate /dev/sdb
pvcreate /dev/sdc
vgcreate datavg /dev/sdb /dev/sdc
lvcreate --type raid1 -m 1 -L 5G -n \
         lvmirror datavg
```

LE = Logical Extent
PE = Physical Extent

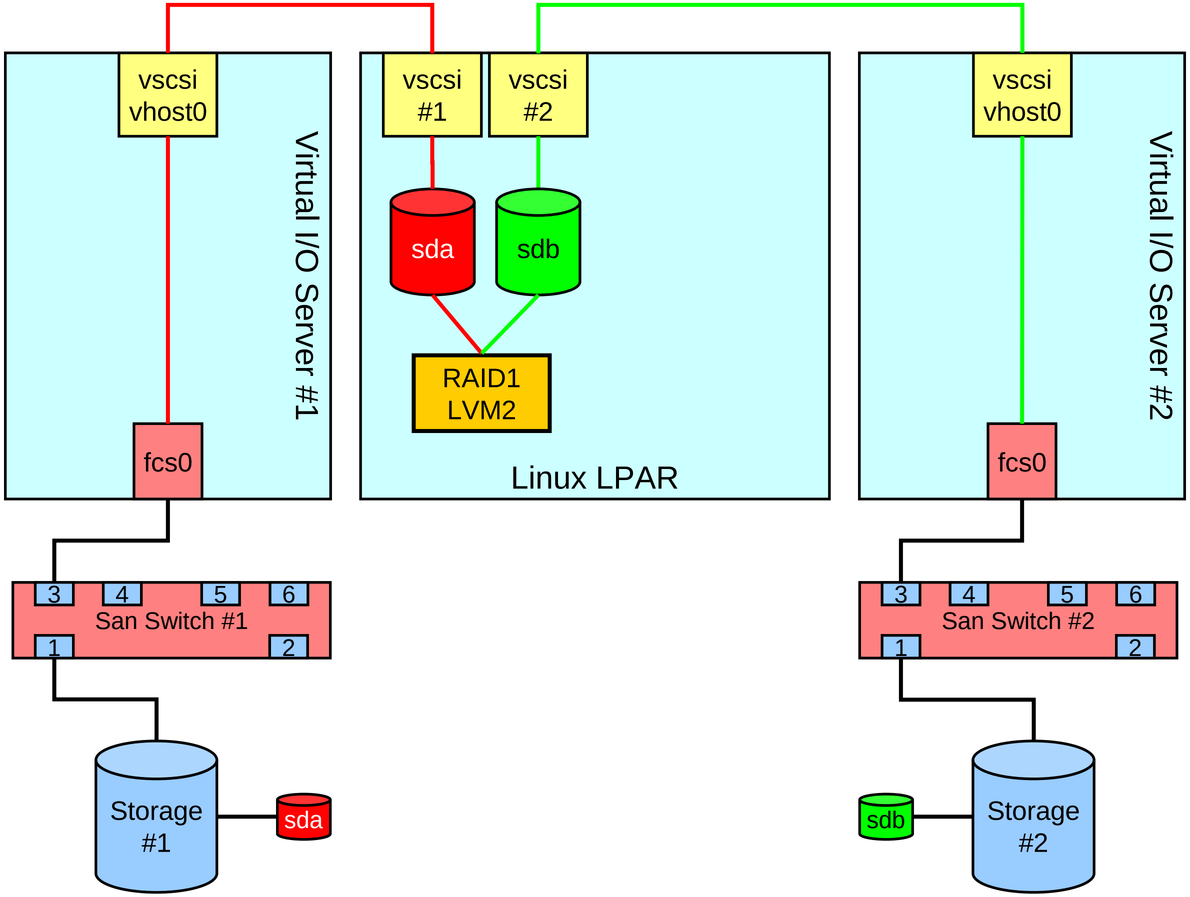
Simplest software RAID1 with Linux md devices

Single LUN for OS/Data provided by each VIOS from separate storage subsystem



Simplest software RAID1 with Linux LVM V2

Single LUN for OS/Data provided by each VIOS from separate storage subsystem



Set the sync speed of Linux MD devices

Change the sync speed of MD devices

```
echo  50000 > /proc/sys/dev/raid/speed_limit_min
echo 200000 > /proc/sys/dev/raid/speed_limit_max
```

or use `sysctl` (edit `/etc/sysctl.conf` for permanent settings)

```
sysctl -w dev.raid.speed_limit_min=50000
sysctl -w dev.raid.speed_limit_max=200000
```

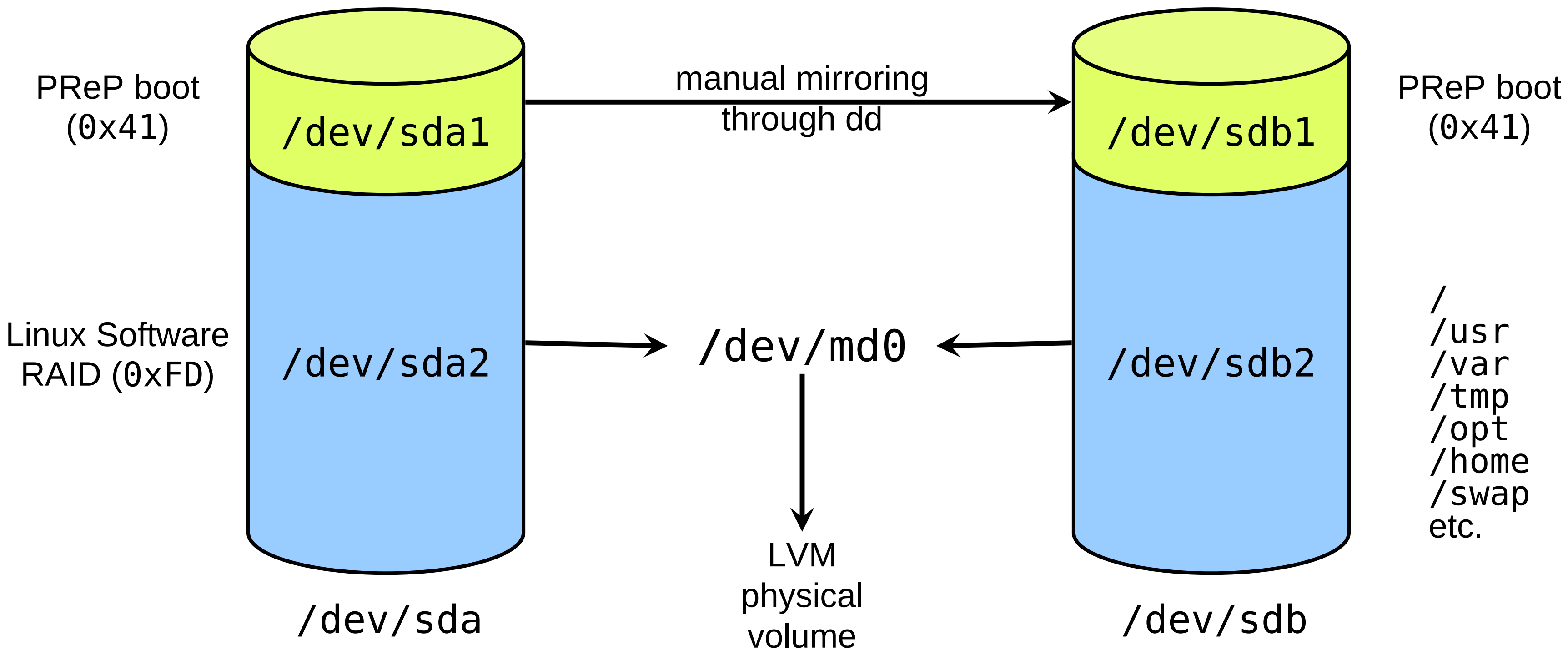
Linux mdadm RAID1 for root

- Software RAID implementation on Power slightly different compared to x86
  - PPC PReP Boot partition (type 0x41) used instead of boot sector
  - PPC PReP Boot partition (type 0x41) = AIX LV “`/dev/hd5`“ = boot logical volume

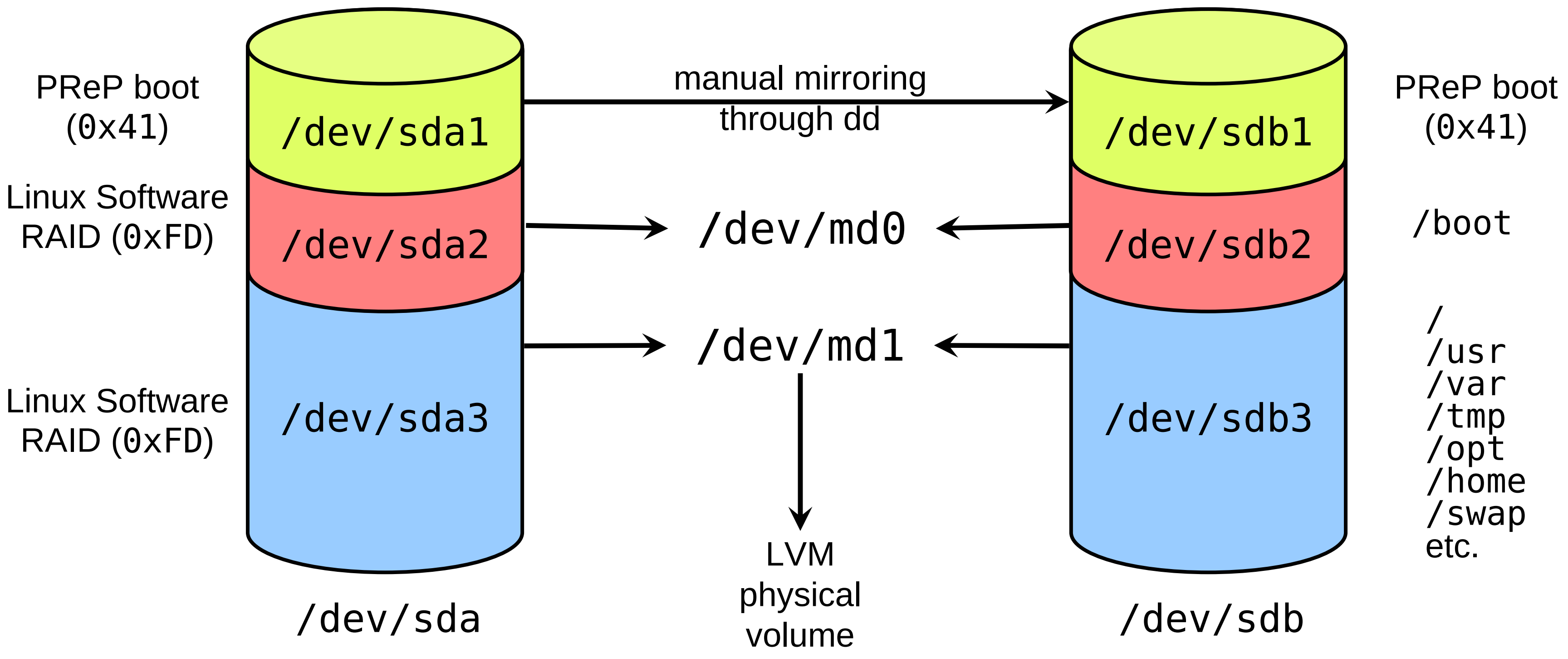
- yaboot (**y**et **a**nother **boot** loader) is used to boot PowerLinux

- Software RAID partitioning and setup different for SUSE and Red Hat
  - SUSE changes PPC PReP Boot partition to FAT and puts `/etc/yaboot.cnf` there
  - Red Hat requires a separate `/boot` partition (can be on mirrored device, e.g., `/dev/md0`)

Disk partitioning for SUSE Linux: mdadm RAID-1 setup



Disk partitioning for Red Hat Linux: mdadm RAID-1 setup



Software RAID1 with mdadm repair action

Example: /dev/md2 with /dev/sda4 and /dev/sdb4

```
# /dev/sda4 (of /dev/md2) is faulty
mdadm --manage --set-faulty /dev/md2 /dev/sda4
mdadm --manage --remove /dev/md2 /dev/sda4

# explanation of the "funny numbers":
# <SCSI-Adapter>:<Channel(bus)>:<Target>:<Lun>
# we rescan the whole "/dev/sda" disk here, not just /dev/sda4

# rescan of SCSI device
echo 1 > /sys/class/scsi_device/0:0:1:0/device/rescan

# hot-add device again
mdadm --manage --add /dev/md2 /dev/sda4
```

Important files for Linux mdadm

- `/proc/mdstat`
  - Shows the status of all currently active MD devices
  - Usage: `"cat /proc/mdstat"`

- `/sbin/mdadm`
  - Manage Linux MD devices

- `/etc/mdadm.conf`
  - Configuration file for mdadm

Device discovery of Linux on Power

Device discovery of Linux on Power (1/3)

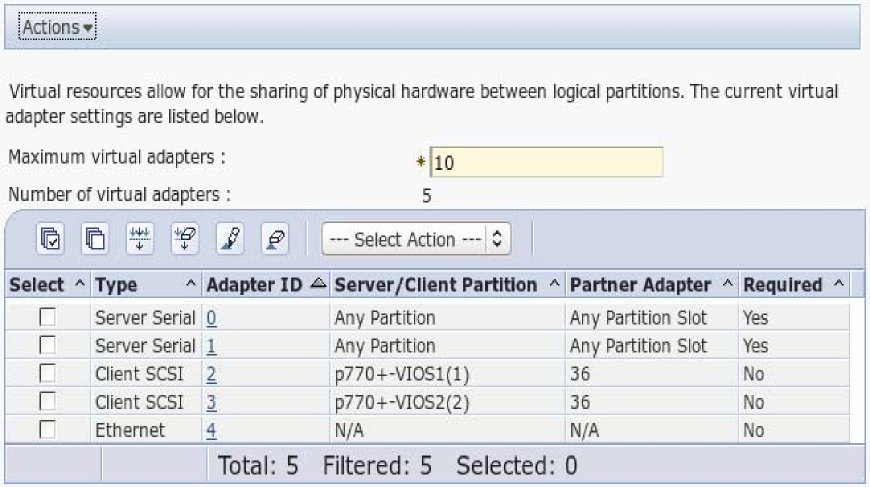
How does Linux discover its devices ?

- It scans all device drivers in the order in which they are defined in the LPAR profile and discovers all devices attached/mapped to the device driver.
- All discovered VIOS SCSIs disk are then numbered consecutively, starting with `/dev/sda`, `/dev/sdb` etc.
- This default disk labeling (`/dev/sd?`) scheme should be avoided and different disk labeling schemes be used instead:
  - `/dev/disk/by-id`
  - `/dev/disk/by-label`
  - `/dev/disk/by-path`
  - `/dev/disk/by-uuid`

Device discovery of Linux on Power (2/3)

Device discovery example with VSCSI devices:

1. All devices attached to VSCSI (slot #2)
2. All devices attached to VSCSI (slot #3)



Actions ▾

Virtual resources allow for the sharing of physical hardware between logical partitions. The current virtual adapter settings are listed below.

Maximum virtual adapters : * 10

Number of virtual adapters : 5

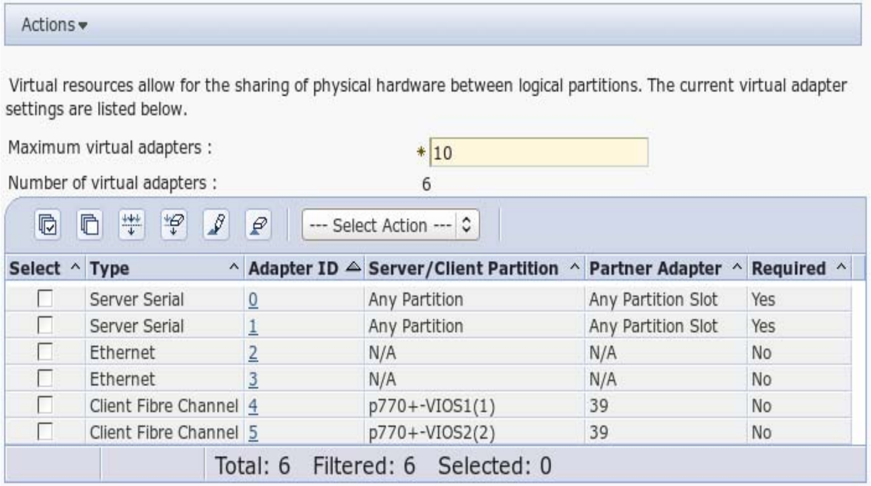
--- Select Action ---

Select	Type	Adapter ID	Server/Client Partition	Partner Adapter	Required
☐	Server Serial	0	Any Partition	Any Partition Slot	Yes
☐	Server Serial	1	Any Partition	Any Partition Slot	Yes
☐	Client SCSI	2	p770+-VIOS1(1)	36	No
☐	Client SCSI	3	p770+-VIOS2(2)	36	No
☐	Ethernet	4	N/A	N/A	No

Total: 5 Filtered: 5 Selected: 0

Device discovery example with VFC devices:

1. All devices attached to VFC (slot #4)
2. All devices attached to VFC (slot #5)



Actions ▾

Virtual resources allow for the sharing of physical hardware between logical partitions. The current virtual adapter settings are listed below.

Maximum virtual adapters : * 10

Number of virtual adapters : 6

--- Select Action ---

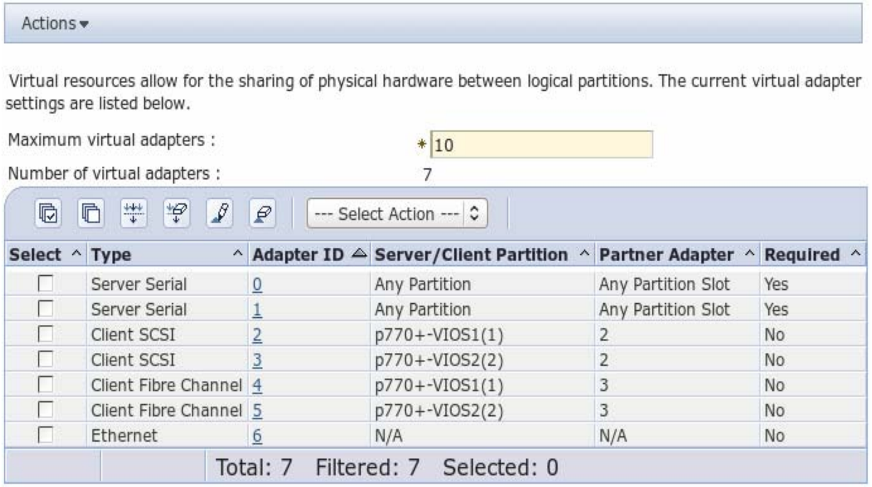
Select	Type	Adapter ID	Server/Client Partition	Partner Adapter	Required
☐	Server Serial	0	Any Partition	Any Partition Slot	Yes
☐	Server Serial	1	Any Partition	Any Partition Slot	Yes
☐	Ethernet	2	N/A	N/A	No
☐	Ethernet	3	N/A	N/A	No
☐	Client Fibre Channel	4	p770+-VIOS1(1)	39	No
☐	Client Fibre Channel	5	p770+-VIOS2(2)	39	No

Total: 6 Filtered: 6 Selected: 0

Device discovery of Linux on Power (3/3)

Device discovery example with mixed (VSCSI + VFC) device types:

1. All devices attached to VSCSI (slot #2)
2. All devices attached to VSCSI (slot #3)
3. All devices attached to VFC (slot #4)
4. All devices attached to VFC (slot #5)



Actions ▾

Virtual resources allow for the sharing of physical hardware between logical partitions. The current virtual adapter settings are listed below.

Maximum virtual adapters : * 10

Number of virtual adapters : 7

--- Select Action ---

Select	Type	Adapter ID	Server/Client Partition	Partner Adapter	Required
□	Server Serial	0	Any Partition	Any Partition Slot	Yes
□	Server Serial	1	Any Partition	Any Partition Slot	Yes
□	Client SCSI	2	p770+-VIOS1(1)	2	No
□	Client SCSI	3	p770+-VIOS2(2)	2	No
□	Client Fibre Channel	4	p770+-VIOS1(1)	3	No
□	Client Fibre Channel	5	p770+-VIOS2(2)	3	No
□	Ethernet	6	N/A	N/A	No

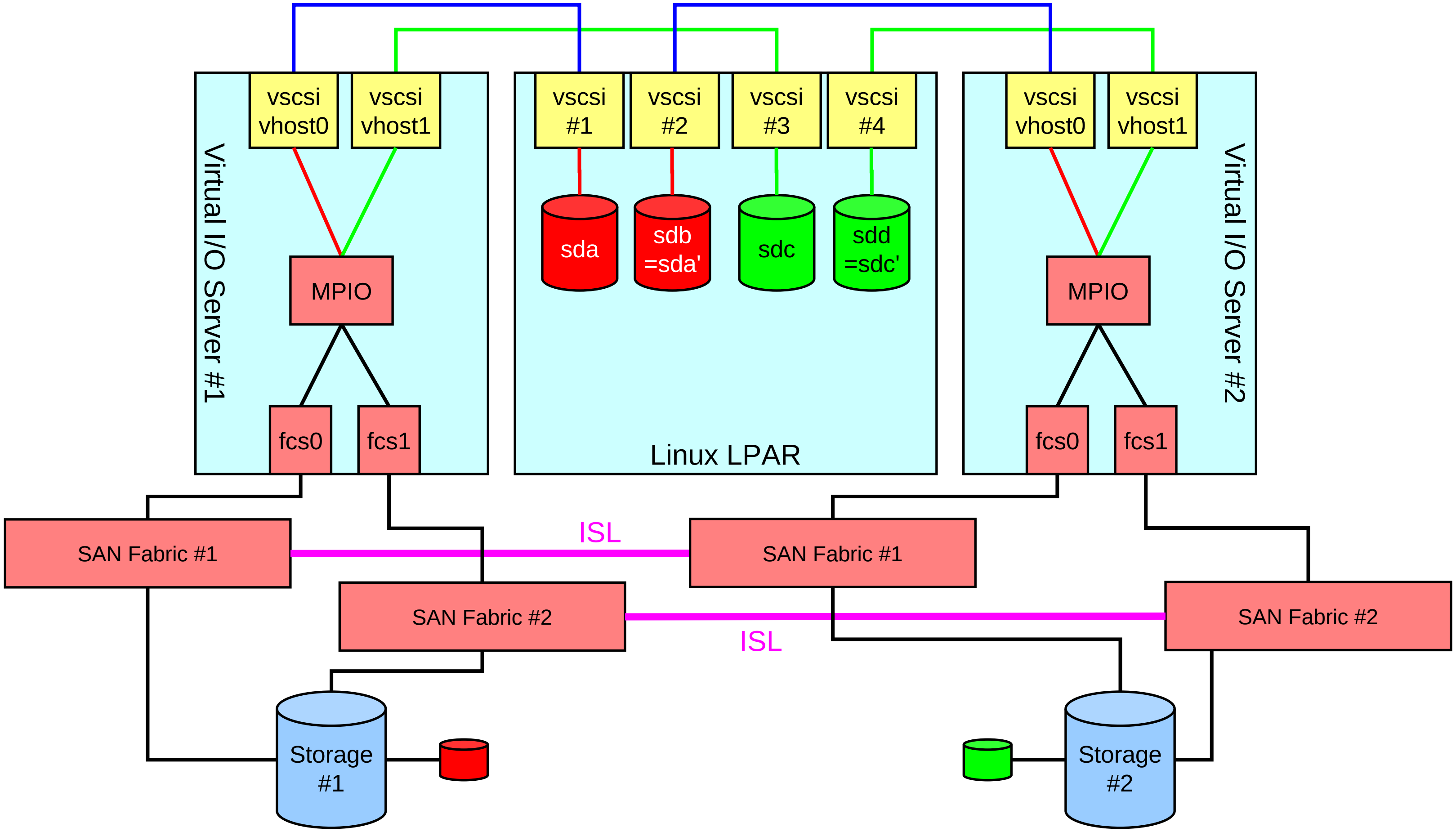
Total: 7 Filtered: 7 Selected: 0

Possible problem:

- After adding some virtual disks to a VIOS adapter and a Linux rescan the device order might/will change.
- This can be avoided by using different disk labeling schemes than `/dev/sd?`, for instance:
  - `/dev/disk/by-id`
  - `/dev/disk/by-label`
  - `/dev/disk/by-path`
  - `/dev/disk/by-uuid`

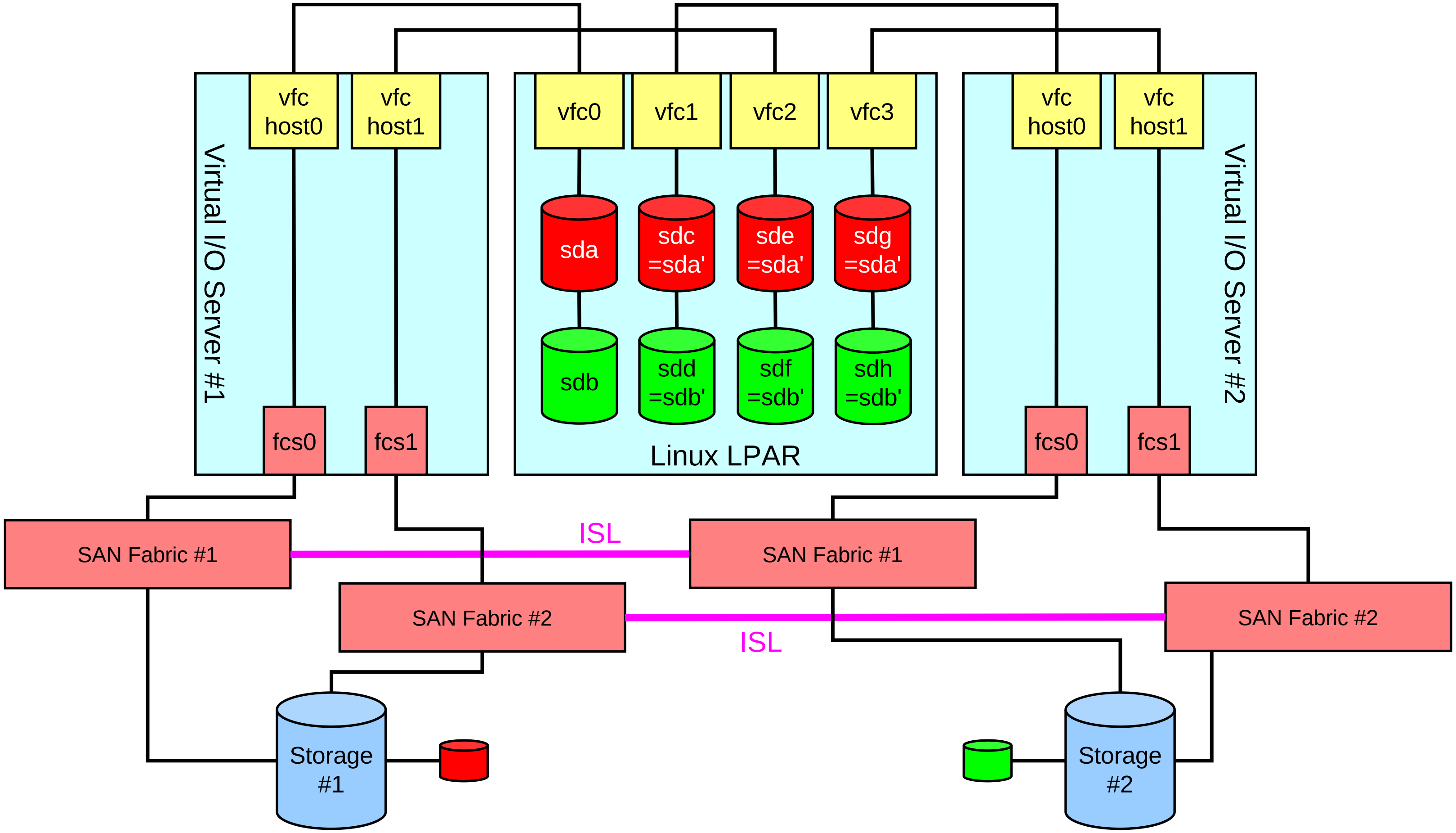
“Real life setup“ VSCSI

Single LUN for OS/data



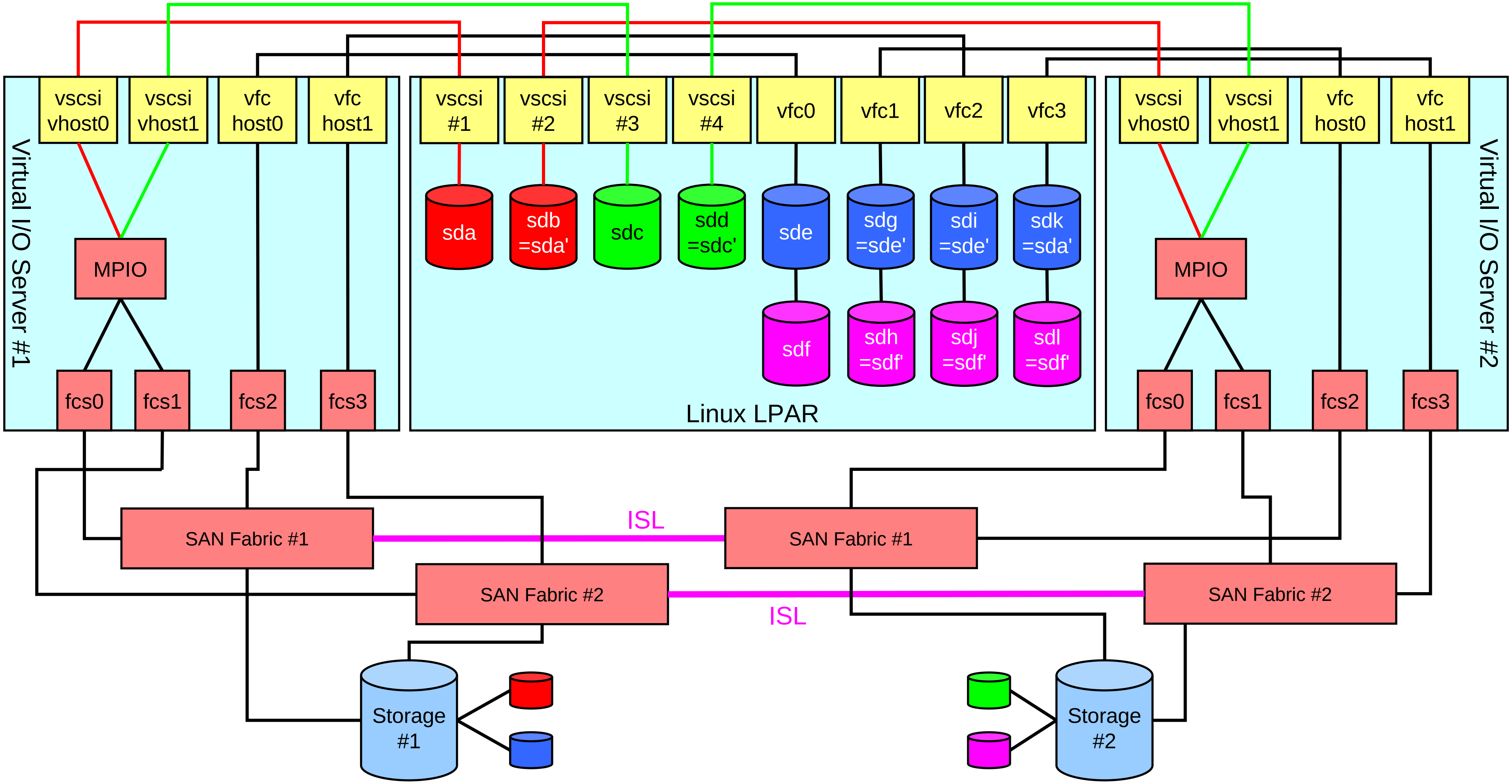
“Real life setup“ NPIV

Single LUN for OS/data

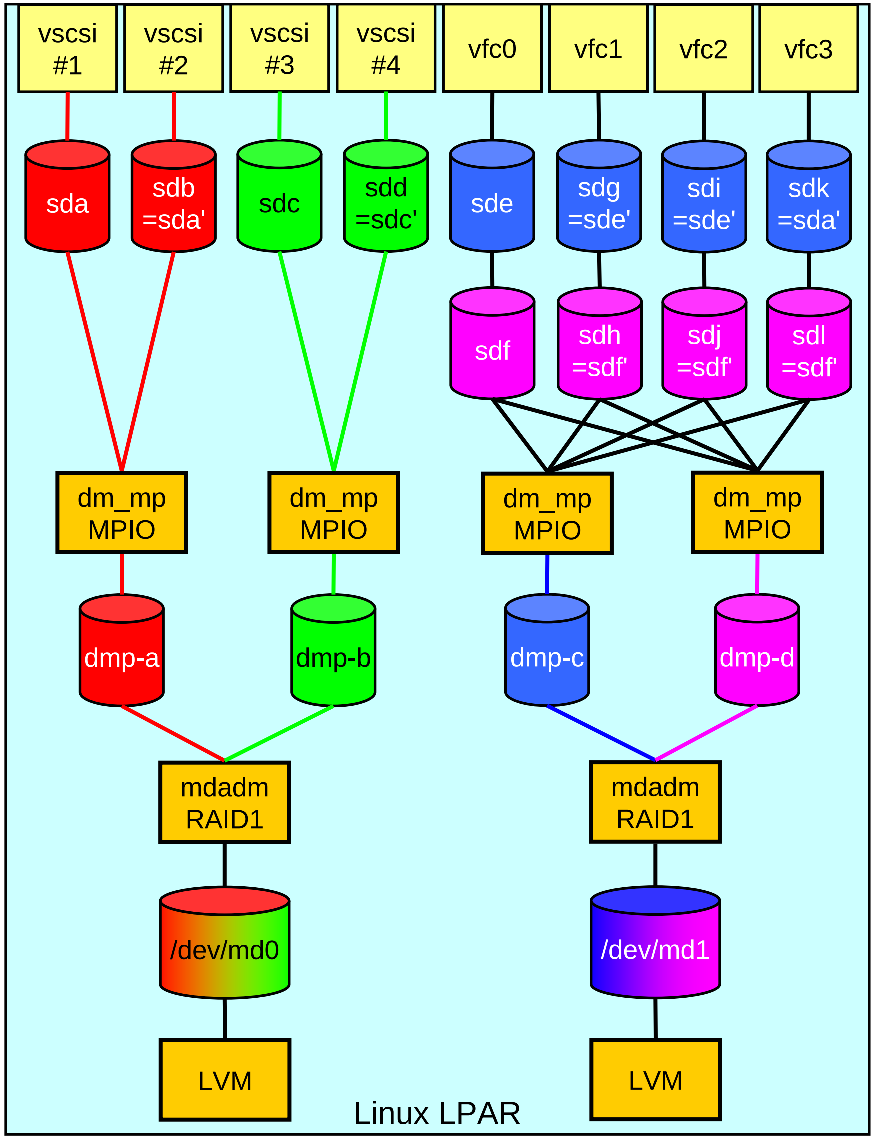


“Real life setup“: VSCSI for rootvg, NPIV for data

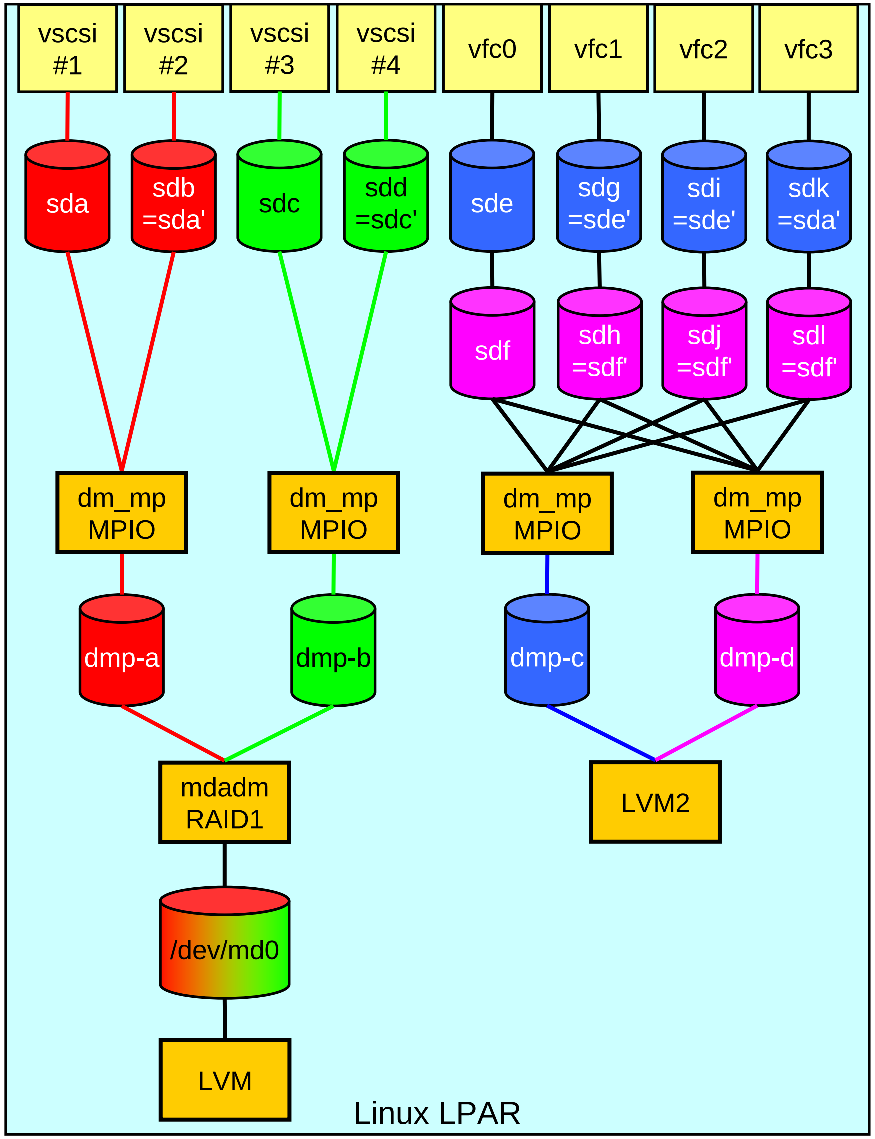
Single LUN for OS, single LUN for data



“Real life setup“: VSCSI for rootvg, NPIV for data



“Real life setup“: VSCSI for rootvg, NPIV for data



Adding/removing disks, rescan devices

Adding/removing disks, rescan devices

Please note:

- **This is generic Linux and not specific to Power! (same on x86 Linux)**

- List of all SCSI host adapters:

```
cd /sys/class/scsi_host
ls -l host*
```

- `host0`, `host1`, etc. then point as a symlink to the real host devices.
  For Virtual I/O Server devices those are located in:

```
/sys/devices/vio
```

- There you find for each virtual device a directory named as follows:

```
300000<xy>
```

where <`xy`> represents the virtual slot number in hexadecimal notation

SCSI Device Addressing

A four-part addressing scheme is used to define the location of SCSI devices:

```
0:0:1:0
```

Host: Instance of host adapter to which device is attached

Bus: SCSI Bus or Channel on the host adapter

Target: SCSI ID assigned to an individual device

LUN: Logical unit number on the device

```
# lsscsi
[0:0:1:0]    disk    AIX     VDASD             0001  /dev/sda
[0:0:2:0]    cd/dvd  AIX     VOPTA                   /dev/sr0
[0:0:3:0]    disk    AIX     VDASD             0001  /dev/sdb
[0:0:4:0]    disk    AIX     VDASD             0001  /dev/sdc
```

sysfs /sys/class/scsi_host/ (2.6 kernel)

- The 2.6 (and higher) kernel provides the `/sys` (sysfs) interface for interacting and managing system devices.
- In the case of SCSI devices, the `/sys/class/scsi_host/` interface can be used to dynamically rescan a host adapter, as well as add or remove specific devices.

- To rescan a host adapter:

```
echo '- - -' > /sys/class/scsi_host/host<X>/scan
```

where `host<X>` refers to the host adapter or the instance of host adapter where multiple (of the same type) exist on the system.

- To rescan an individual device (replace `0:0:4:0` with your parameters):

```
echo "1" > /sys/class/scsi_device/0:0:4:0/device/rescan
```

Virtual devices example

```
# find /sys -name scan
/sys/devices/vio/30000002/host0/scsi_host/host0/scan
/sys/module/scsi_mod/parameters/scan

# find /sys -name rescan
/sys/devices/vio/30000002/host0/target0:0:1/0:0:1:0/rescan
/sys/devices/vio/30000002/host0/target0:0:2/0:0:2:0/rescan
/sys/devices/vio/30000002/host0/target0:0:3/0:0:3:0/rescan
/sys/devices/vio/30000002/host0/target0:0:4/0:0:4:0/rescan
/sys/bus/pci/rescan
```

Removing disk example

```
# lsscsi
  [0:0:1:0]    disk    AIX      VDASD             0001  /dev/sda
  [0:0:2:0]    cd/dvd  AIX      VOPTA                   /dev/sr0
  [0:0:3:0]    disk    AIX      VDASD             0001  /dev/sdb
  [0:0:4:0]    disk    AIX      VDASD             0001  /dev/sdc
```

- To remove disk **/dev/sdc**

```
echo 1 > /sys/block/sdc/device/delete
```

- or to remove a specific device (again **/dev/sdc**):

```
echo 1 > /sys/class/scsi_host/host0/device/target0:0:4/0:0:4:0/delete
```

Adding disks / rescan devices

- **RHEL/Fedora and SLES/openSUSE** (need to install package `sg3_utils`)

```
/usr/bin/rescan-scsi-bus.sh  [-r]
```

- **Ubuntu/Debian** (need to install package `sg3_utils`)

```
/sbin/rescan-scsi-bus.sh  [-r]
```

- **Direct access** – for the “hard core Linux guys”

```
# <Host adapter>:<Bus>:<Target>:<LUN>
echo 1 > /sys/class/scsi_device/0:0:1:0/device/rescan
```

Resize of Linux on Power file systems

Supported file systems for Linux on Power (1/2)

	RHEL 6.5	RHEL 7.0	SLES 11 SP3, SLES 12	Fedora 20	Debian 7, Ubuntu 14.04, Debian 14.10	openSUSE 13.1, openSUSE 13.2
ext2	yes	yes	yes	yes	yes	yes
ext3	yes	yes	yes	yes	yes	yes
ext4	yes	yes	yes	yes	yes	yes
reiserfs	no	no	yes	yes	yes	yes
xfs	partial *	yes	yes	yes	yes	yes
btrfs	not yet	not yet	yes	yes	yes	yes
fat/vfat/msdos	yes	yes	yes	yes	yes	yes
ntfs	yes	yes	yes	yes	yes	yes

partial *: Please check the notes on the next slide.

Supported file systems for Linux on Power (2/2)

SLES 11 SP3

- Btrfs supported since SLES 11 SP2 (not for /boot).
  - Btrfs is supported on top of MD (multiple devices) and DM (device mapper) configurations.
  - Please use the YaST partitioner to achieve a proper setup.
  - Multivolume/RAID with btrfs is not supported yet and will be enabled with a future maintenance update.
- XFS is supported since the release of SLES 8.

SLES 12

- Btrfs is now the default file system (and not XFS).

RHEL 6.5

- ”Btrfs is not a production quality file system at this point.”
  - With Red Hat Enterprise Linux 6 it is at a technology preview stage and as such is only being built for Intel 64 and AMD64.
- XFS is an addon product and not supported as a root file system.
  - http://www.redhat.com/products/enterprise-linux-add-ons/file-systems/

RHEL 7.0

- XFS is the default file system (instead of ext4)
- BTRFS is a Technology Preview in Red Hat Enterprise Linux 7.

File system size change support for Linux on Power

	Utility	Increase Size (Grow)	Decrease Size (Shrink)
ext2	`resize2fs`	Offline only	Offline only
ext3	`resize2fs`	Online or offline	Offline only
ext4	`resize2fs`	Offline only	Offline only
reiserfs	`resize_reiserfs`	Online or offline	Offline only
xfs	`xfs_growfs`	Online or offline	Not possible *
btrfs	`btrfs filesystem resize`	Online or offline	Online or offline

* You can NOT make a XFS partition smaller online.
The only way to shrink is to do a complete dump, mkfs and restore.

And the clear winner (in my opinion) is/will be… btrfs

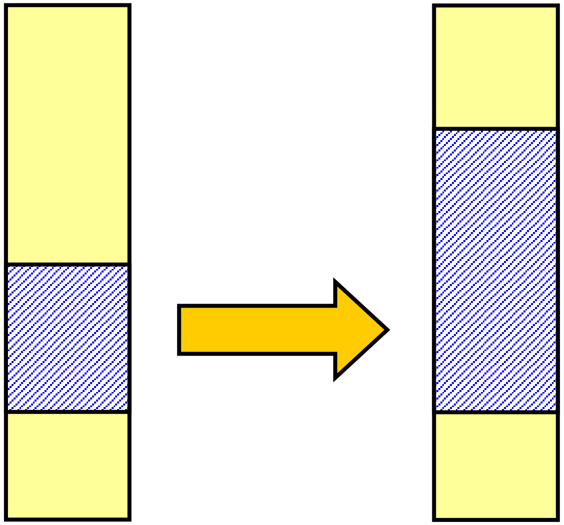
Resize file system considerations (1/2)

Resize file system scenarios

1) Resize a fdisk partition
2) Resize a logical volume

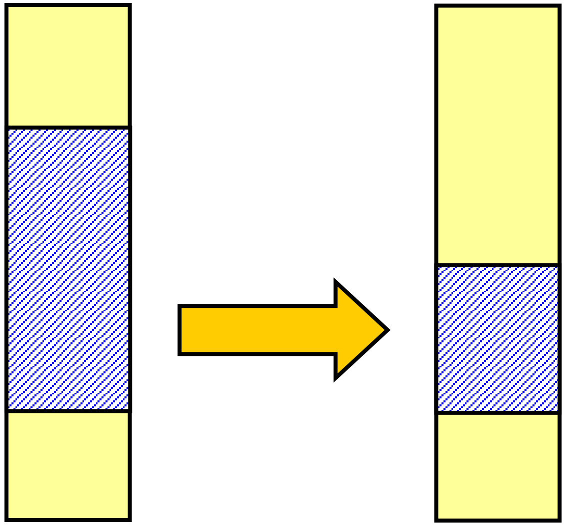
Resize file system operations

a) Increase size (grow)



→ Easier to do

b) Decrease size (shrink)



→ Much harder to do

Resize file system considerations (2/2)

Increase the size of a file system on a fdisk partition:

1) Increase the size of the underlying fdisk partition.
2) Increase the size of the file system.

→ Can be done online if supported by file system.

Shrink the size of a file system on a fdisk partition:

1) Shrink the size of the file system.
2) Shrink the size of the underlying fdisk partition.

→ Can be rather dangerous!

→ Data loss possible!

Increase the size of a file system on a logical volume:

1) Increase the size of the underlying logical volume.
2) Increase the size of the file system.

→ Can be done online if supported by file system.

Shrink the size of a file system on a logical volume:

1) Shrink the size of the file system.
2) Shrink the size of the underlying logical volume.

→ Can be done online if supported by file system.

File system size changes examples (only LVs)

Increase file system examples

- Increase ext3 file system
- Increase ext2/ext4 file system
- Increase reiserfs file system
- Increase xfs file system
- Increase btrfs file system

Shrink file system examples

- Shrink ext3 file system
- Shrink ext2 or ext4 file system
- Shrink reiserfs file system
- Shrink xfs file system
- Shrink btrfs file system

Use always the following settings:

```
# fdisk -l | grep Disk | grep sd | sort
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectors
```

Increase file system size for simple LV

```
# pvcreate /dev/sdb
# vgcreate testvg /dev/sdb
# lvcreate -L 5G -n lvtest testvg

# pvs --segments -o+lv_name,seg_start_pe,segtype
# lvs -a -o +devices,stripes,stripesize,seg_pe_ranges --segments

# mkfs.<FS> -f /dev/testvg/lvtest            # <FS>=ext2,ext3,ext4,
                                                    reiserfs,xfs,btrfs
# mkdir -p /testfs
# mount /dev/testvg/lvtest /testfs
# df -k | grep testfs

# lvextend -L 8G /dev/testvg/lvtest
# lvs -a -o +devices,stripes,stripesize,seg_pe_ranges -segments

# fsadm resize /dev/testvg/lvtest            # ext2,ext3,ext4,xfs
  resize_reiserfs /dev/testvg/lvtest         # reiserfs
  btrfs filesystem resize max /testfs        # btrfs
# df -k | grep testfs
```

Increase file system size for mirrored LV (LVM2)

```
# pvcreate /dev/sdb /dev/sdc
# vgcreate testvg /dev/sdb /dev/sdc
# lvcreate --type raid1 -m 1 -L 5G -n lvtest testvg

# pvs --segments -o+lv_name,seg_start_pe,segtype
# lvs -a -o +devices,stripes,stripesize,seg_pe_ranges --segments

# mkfs.<FS> -f /dev/testvg/lvtest            # <FS>=ext2,ext3,ext4,
                                                    reiserfs,xfs,btrfs
# mkdir -p /testfs
# mount /dev/testvg/lvtest /testfs
# df -k | grep testfs

# lvextend -L 8G /dev/testvg/lvtest
# lvs -a -o +devices,stripes,stripesize,seg_pe_ranges -segments

# fsadm resize /dev/testvg/lvtest            # ext2,ext3,ext4,xfs
  resize_reiserfs /dev/testvg/lvtest         # reiserfs
  btrfs filesystem resize max /testfs        # btrfs
# df -k | grep testfs
```

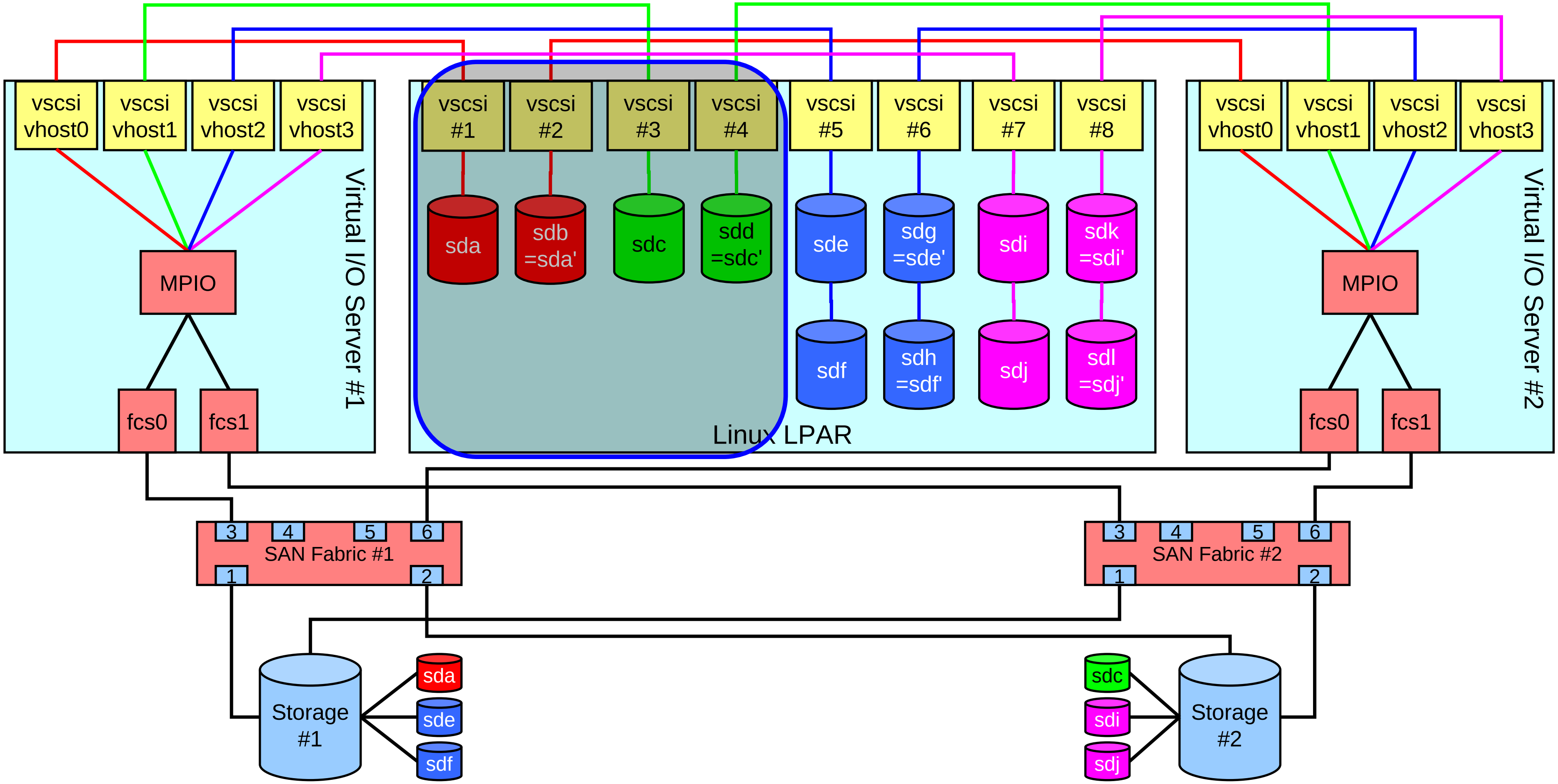
Shrink file system size for (mirrored) LV

```
# umount /testfs                                  # not for btrfs

# fsadm resize /dev/testvg/lvtest 4G
  resize_reiserfs -s 4G /dev/testvg/lvtest
  btrfs filesystem resize 4G /testfs

# lvreduce -f -L 4G /dev/testvg/lvtest
# lvs -a -o +devices,stripes,stripesize,seg_pe_ranges --segments

# mount /dev/testvg/lvtest /testfs                # not for btrfs
# df -k | grep testfs
```

High availability setup with VSCSI

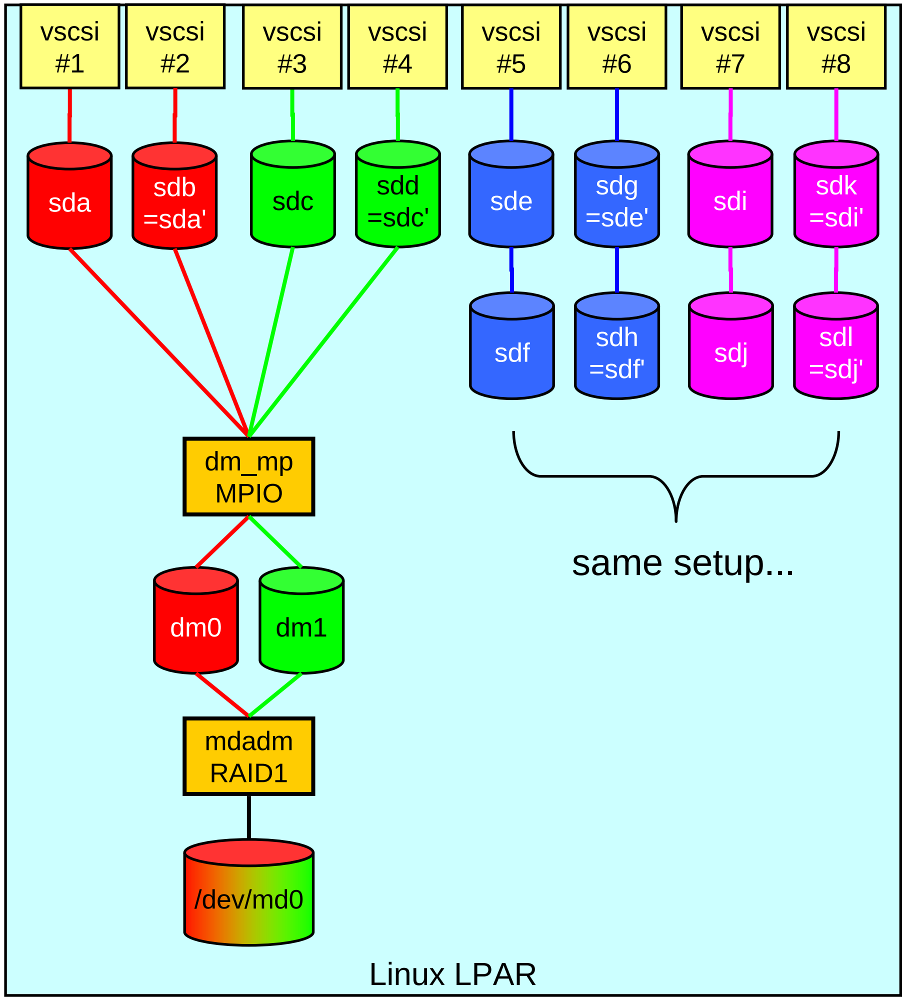
Putting it all together... – high availability setup (1/4)

- Single LUN for OS, multiple LUNs for data provided by each VIOS from separate storage subsystems
- A complete VIOS failure will cause only loss of one path to disks but nothing else !



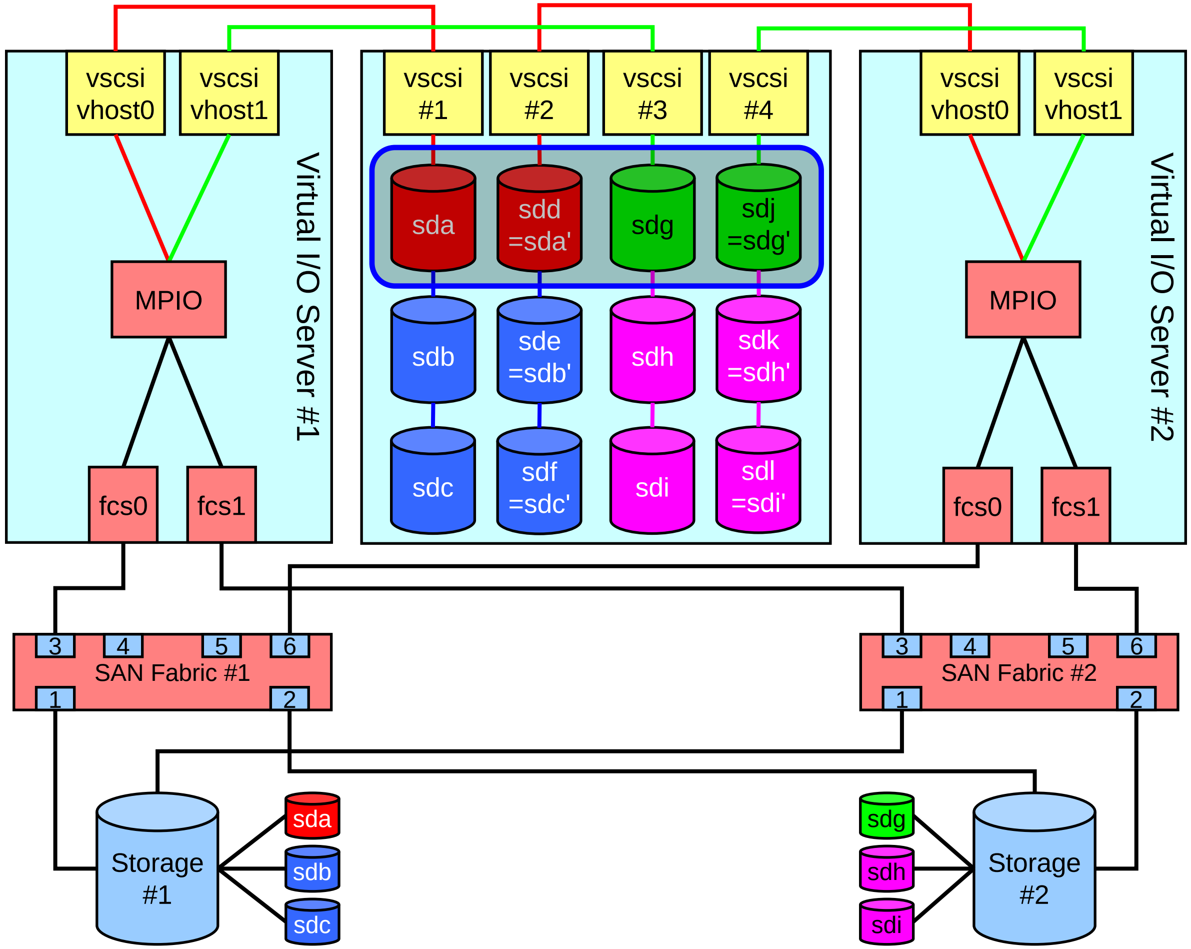
Putting it all together... – high availability setup (2/4)

- Single LUN for OS, multiple LUNs for data provided by each VIOS from separate storage subsystems
- A complete VIOS failure will cause only loss of one path to disks but nothing else !



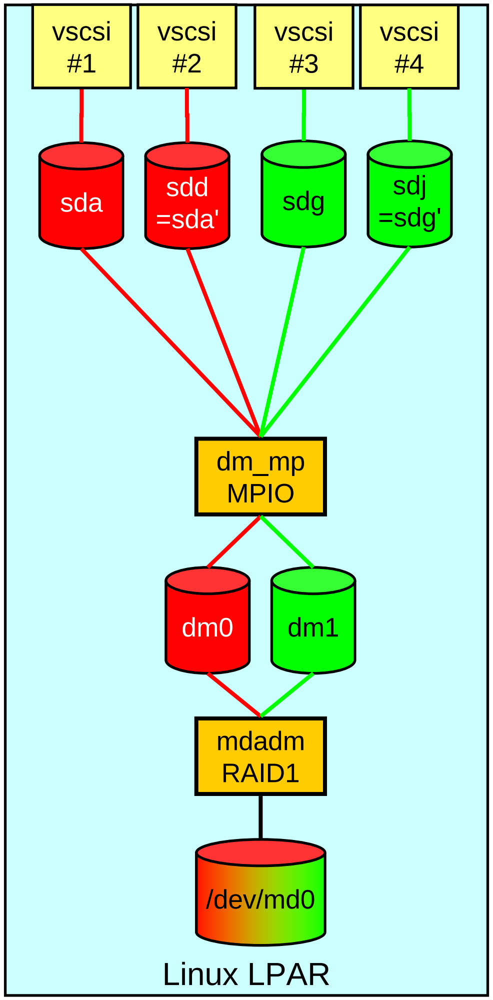
Putting it all together... – high availability setup (3/4)

- Single LUN for OS, multiple LUNs for data provided by each VIOS from separate storage subsystems
- A complete VIOS failure will cause only loss of one path to disks but nothing else !



Putting it all together... – high availability setup (4/4)

- Single LUN for OS, multiple LUNs for data provided by each VIOS from separate storage subsystems
- A complete VIOS failure will cause only loss of one path to disks but nothing else !



Example: Activation and stopping of resources

Activation involves the following steps:

- Assembly of RAID array
- Import and activation of volume group
- Mount of file system

Stopping involves the following steps:

- Unmount of file system
- Stop and export of volume group
- Stop of RAID array

Variable definitions for this example:

```
MD="/dev/md1"
MDDEV1="/dev/mapper/A_DATA01-part1"
MDDEV2="/dev/mapper/B_DATA01-part1"
VG="vgdata"
LV="lvdata"
FS="/data"
MDADM="/sbin/mdadm"
VGIMPORT="/sbin/vgimport"
VGCHANGE="/sbin/vgchange"
LVCHANGE="/sbin/lvchange"
MOUNT="/bin/mount"
UMOUNT="/bin/umount"
VGEXPORT="/sbin/vgexport"
```

Activation of resources:

```
# starting RAID array ${MD}
${MDADM} --assemble ${MD} ${MDDEV1} ${MDDEV2}
# importing volume group ${VG}
${VGIMPORT} ${VG}
# activating volume group ${VG}
${VGCHANGE} -ay ${VG}
${LVCHANGE} --refresh ${VG}/${LV}
# mounting file system ${FS}
${MOUNT} ${FS}
```

Stopping of resources:

```
# unmount file system ${FS}
${UMOUNT} ${FS}
# stop volume group ${VG}
${VGCHANGE} -an ${VG}
${VGEXPORT} ${VG}
# stop RAID array ${MD}
${MDADM} --stop ${MD}
```

Setting up a RHEL repository for YUM

Setting up a RHEL repository for YUM (1/2)

- NFS server provides the contents of the RHEL DVD, for instance RHEL 6.5

```
<nfs-server>:/export/linux/ppc/RHEL6U5
```

- Create a RHEL base software repository for YUM

```
[root@rhel65 /]# cat /etc/yum.repos.d/rhel-base.repo
[rhel-base]
name=Red Hat Enterprise Linux \$releasever - \$basearch
baseurl=file:///misc/yum/Server
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Setting up a RHEL repository for YUM (2/2)

- Enable automounter

```
[root@rhel65 /]# chkconfig --list | grep autofs
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
[root@rhel65 /]# chkconfig autofs on
```

- Add this line at the end of `/etc/auto.misc` (config file of automounter)

```
yum      -ro,soft,intr,vers=3     <nfs-server>:/export/linux/RHEL6U5
```

- Restart the automounter

```
/etc/init.d/autofs restart
```

Now you can use YUM with this new repository…

How to obtain information from within a Linux LPAR

/proc/ppc64/lparcfg (virtual file)

- `/proc/ppc64/lparcfg`
  - Details can be found here
- **Description**
  - The lparcfg file is a virtual file which contains information related to an IBM Power Logical Partition.
  - The AIX command “`lparstat`” is also available for Linux (using `/proc/ppc64/lparcfg`).
- **Options**
  - The fields displayed in the lparcfg file are sorted below according to the version of the lparcfg file. Generally, fields are only added and not removed (unless otherwise noted), so the latest version of lparcfg contains all fields in all previous versions of the file as well.
- **Linux distribution version mapping**

– SLES 9	lparcfg 1.6	
– SLES 10	lparcfg 1.7	
– SLES 11	lparcfg 1.8	← addition of Active Memory Sharing
– SLES 11 SP1	lparcfg 1.9	
– RHEL 4	lparcfg 1.6	
– RHEL 5	lparcfg 1.7	
– RHEL 6	lparcfg 1.9	
– RHEL 7	lparcfg 1.9	

/proc/ppc64/lparcfg 1.6 based values (1/5)

- `serial_number`
  - The serial number of the physical system in which the partition resides.
- `system_type`
  - The machine,type-model of the physical system in which the partition resides.
- `partition_id`
  - The numeric partition ID.
- `R4`
  - The hexadecimal representation of partition_entitled_capacity. This field is deprecated and not displayed on more recent versions of the Linux kernel (lparcfg 1.8 or greater). The definition is only provided for historical purposes.
- `R5`
  - The hexadecimal representation of unallocated_capacity. Not displayed on more recent versions of the Linux kernel. This field is deprecated and not displayed on more recent versions of the Linux kernel (lparcfg 1.8 or greater). The definition is only provided for historical purposes.
- `R6`
  - This is a hexadecimal value representing both the group and pool. This field is deprecated and not displayed on more recent versions of the Linux kernel (lparcfg 1.8 or greater). The definition is only provided for historical purposes.
- `R7`
  - This is a hexadecimal value representing capped, capacity_weight, unallocated_capacity_weight, pool_capacity, and system_active_processors. This field is deprecated and not displayed on more recent versions of the Linux kernel (lparcfg 1.8 or greater). The definition is only provided for historical purposes.
- `BoundThrds`
  - For virtual processor dispatches, if the hypervisor always dispatches a set of virtual threads together on a physical processor, the threads are said to be bound. This allows an operating system to make scheduling decisions based on cache affinity and work load. Set to 1 if threads are bound, 0 otherwise. This value is informational and is not a tunable value.

/proc/ppc64/lparcfg 1.6 based values (2/5)

- `CapInc`
  - This defines the delta by which the entitled capacity of a partition can be incremented or decremented by DLPAR/WLM. The capacity increment is expressed as a percentage of a physical processor. This value is informational and is not a tunable value.
- `DisWheRotPer`
  - The duration of the hypervisor's scheduling window. The time over which the entitled capacity of a virtual processor has to be utilized by the partition. At the start of a dispatch wheel rotation period, each virtual processor is eligible for CPU time corresponding to its entitled capacity. If the entire entitled capacity of a virtual processor is not utilized during a dispatch wheel rotation period, the unused entitled capacity is lost. The dispatch wheel rotation period is expressed as N number of time base ticks. The dispatch wheel duration of a partition with a capacity increment of 100 is 0. This value is informational and is not a tunable value.
- `MinEntCap`
  - The minimum entitled capacity that is needed to boot the partition. The capacity is expressed as a percentage of a physical processor. The minimum entitled capacity is set by the system administrator in the partition definition. DLPAR cannot take the entitled capacity below the minimum entitled capacity. A change in the minimum entitled capacity takes effect on the next reboot of the partition. Linux running in a partition can give up its entitled capacity to be below the minimum entitled capacity, but this is generally not recommended.
- `MinEntCapPerVP`
  - The minimum entitled capacity that the platform requires for a virtual processor of any partition on the platform. The minimum capacity per virtual processor is enforced by the HMC in the partition definition and by the hypervisor. A change in the minimum entitled capacity per virtual processor takes effect on the next reboot of the partition. This is a physical system setting and is not considered a Linux partition tunable.
- `MinMem`
  - The minimum amount of main store that is needed to boot the partition. Minimum memory is expressed in MB of storage. The minimum memory is set by the system administrator in the partition definition. DLPAR cannot take the partition memory below the minimum memory. A change in the minimum memory takes effect on the next reboot of the partition. Linux running in a partition can always give up its memory to go below the minimum memory.
- `MinProcs`
  - The minimum number of virtual processors that are needed to boot the partition. The minimum number of virtual processors is set by the system administrator in the partition definition. DLPAR cannot take the number of virtual processors below the minimum number of processors. A change in the minimum number of processors takes effect on the next reboot of the partition. A partition can always give up its virtual processors to go below the minimum number of processors. The number of virtual processors is a simulated physical core view. Additional logical CPUs are defined in the Linux partition to account for the possible hardware threads.

/proc/ppc64/lparcfg 1.6 based values (3/5)

- `partition_max_entitled_capacity`
  - The maximum entitled capacity currently that can be assigned to the partition through DLPAR/WLM. The capacity is expressed as a percentage of a physical processor. The Maximum entitled capacity is set up by the system administrator in the partition definition. A change in the maximum entitled capacity maximum takes effect on the next reboot of the partition.
- `system_potential_processors`
  - The maximum number of physical processors that can be active on the platform. A change in the maximum platform processors takes effect on the next reboot of the partition.
- `DesEntCap`
  - The desired entitled capacity is the number of processing units, expressed as a percentage of a physical processor, which is desired for a logical partition. The desired entitled capacity is the same as the desired processing units on the HMC. If the system has at least the desired number of processing units available when you activate the partition, then the system commits the desired number of processing units to the logical partition. If the desired number of processing units is not available, but at least the minimum number of processing units is available, then the system activates the logical partition with the processing units it has.
- `DesMem`
  - The desired memory set by the system administrator in the partition definition. The desired memory is expressed in MB of storage. The desired memory can change without a reboot of the partition. The desired memory that the partition is currently using may differ from the desired memory because of WLM actions or because of failed system memory.
- `DesProcs`
  - The desired number of virtual processors set by the system administrator in the partition definition. The desired number of processors can change without a reboot of the partition. The number of processors that the partition is currently using may differ from the desired number of processors because of WLM actions or because of failed system processors.
- `DesVarCapWt`
  - The desired variable capacity weight set by the system administrator in the partition definition. The desired variable capacity weight is a number between 0 and 255. The desired variable capacity weight can change without a reboot of the partition. The variable capacity weight that the partition is currently using may differ from the desired variable capacity because of WLM actions.
- `DedDonMode`
  - For a partition with a capacity increment of 100, the platform uses a dedicated CPU to actualize a virtual processor of the partition. For such a partition, the platform can increase the capacity of the shared processor pool by utilizing the unused processor capacity of the partition. If the platform supports the dedicated donate function, it can be enabled by the system administrator in the partition definition. The value of this characteristic can change without a reboot of the partition. The values for this field are 0 and 1.

/proc/ppc64/lparcfg 1.6 based values (4/5)

- `partition_entitled_capacity`
  - Entitled Processor Capacity Percentage. The percentage of a physical processor that the hypervisor guarantees to be available to the partition's virtual processors (distributed in a uniform manner among the partition's virtual processors -- thus the number of virtual processors affects the time slice size) each dispatch cycle. Capacity ceded or conferred from one partition virtual processor extends the time slices offered to other partition processors. Capacity ceded or conferred after all of the partition's virtual processors have been dispatched is added to the variable capacity kitty. The initial, minimum and maximum constraint values of this parameter are determined by the partition configuration definition. The OS can set this parameter within the constraints imposed by the partition configuration definition minimum and maximums plus constraints imposed by partition aggregation. To change this value, echo the new partition_entitled_capacity into /proc/ppc64/lparcfg like this:
- `group`
  - LPAR group number of the partition
- `system_active_processors`
  - The number of processors active on the underlying physical system.
- `pool`
  - The pool number of the shared processor pool for the partition. This field is not displayed in the case of a dedicated processor partition.
- `pool_capacity`
  - The number of physical processors active in the partition's processor pool. This field is not displayed in the case of a dedicated processor partition. This value is expressed as a percentage so is 100* the number of active physical processors.
- `pool_idle_time`
  - If no virtual processor is ready to run, the pool_idle_count is incremented the total number of idle processor cycles in the physical processor pool. This field contains the total number of idle processor cycles up to the current point in time. If unsupported or if performance information collection is not enabled for the partition on the HMC, this will report 0. This field is not displayed in the case of a dedicated processor partition. pool_num_procs
  - The number of physical processors in the partition's processing pool. This field is not displayed in the case of a dedicated processor partition.
- `unallocated_capacity_weight`
  - Unallocated Variable Processor Capacity Weight. The amount of variable processor capacity weight that is currently available within the constraints of the partition's current environment for allocation to the partition's variable processor capacity weight.

/proc/ppc64/lparcfg 1.6 based values (5/5)

- `capacity_weight`
  - Variable Processor Capacity Weight. The unitless factor that the hypervisor uses to assign processor capacity in addition to the Entitled Processor Capacity Percentage. This factor may take the values 0 to 255. In the case of a dedicated processor partition this value is 0. A virtual processor's time slice may be extended to allow it to use capacity unused by other partitions, or not needed to meet the Entitled Processor Capacity Percentage of the active partitions. A partition is offered a portion of this variable capacity kitty equal to: (Variable Processor Capacity Weight for the partition) / (summation of Variable Processor Capacity Weights for all competing partitions). The initial value of this parameter is determined by the partition configuration definition. The OS can set this parameter within the constraints imposed by the partition configuration definition maximum. Certain partition definitions may not allow any variable processor capacity allocation. To change this value, echo the new capacity_weight into /proc/ppc64/lparcfg like this:
- `capped`
  - The partition's virtual processor(s) are capped at their entitled processor capacity percentage if this is 1. If capped=0, the partition is uncapped, and can use processor capacity from the uncapped pool, if available and according to the weighted values. In the case of dedicated processors this bit is set.
- `unallocated_capacity`
  - Unallocated Processor Capacity Percentage. The amount of processor capacity that is currently available within the constraints of the partition's current environment for allocation to Entitled Processor Capacity Percentage.
- `purr`
  - The Processor Utilization of Resources Register. Summation of the PURR value for all of the partition's virtual processors.
- `partition_active_processors`
  - The total number of virtual processors assigned to the partition. This does not include the potential SMT threads. For dedicated processor partitions, this is the number of physical processors assigned to the partition. Linux will define virtual CPUs for the possible SMT threads across all of the virtual processors defined here.
- `partition_potential_processors`
  - The maximum number of virtual processors that can be assigned to the partition. This does not include SMT threads. For dedicated processor partitions, this is the maximum number of physical processors that can be assigned to the partition.
- `shared_processor_mode`
  - This is set to 1 if the partition is running with shared processors. This is set to 0 for dedicated processor partitions.

/proc/ppc64/lparcfg 1.7 additional values

- `slb_size`
  - The total number of entries in the Segment Lookaside Buffer (SLB). This is an attribute of the underlying processor architecture and is provided for informational purposes. The Linux OS uses this when determining the ability to perform Live Partition Migration with differing processor families.

/proc/ppc64/lparcfg 1.8 additional values (1/2)

- `entitled_memory`
  - The number of bytes of main storage that the partition is entitled to DMA map for virtual I/O devices. In the case of a dedicated memory partition this is the size of the partition's logical address space.
- `mapped_entitled_memory`
  - The number of bytes of main storage that the partition has DMA mapped. In the case of a dedicated memory partition this is not displayed.
- `entitled_memory_group_number`
  - Entitled Memory Group Number.
- `entitled_memory_pool_number`
  - Entitled memory pool number. In the case of a dedicated memory partition, this is 65535.
- `entitled_memory_weight`
  - The partition's shared memory weight. In the case of a dedicated memory partition this is 0.
- `unallocated_entitled_memory_weight`
  - The unallocated shared memory weight for the calling partition's aggregation. In the case of a dedicated memory partition this is 0.
- `unallocated_io_mapping_entitlement`
  - The unallocated I/O mapping entitlement for the calling partition's aggregation divided by 4096. In the case of a dedicated memory partition this is 0.
- `entitled_memory_loan_request`
  - The signed difference between the number of bytes of logical storage that are currently on loan from the calling partition and the partition's overage allotment (a positive number indicates a request to the partition to loan the indicated number of bytes else they will be expropriated as needed). In the case of a dedicated memory partition this is 0. In the case of a shared memory partition, when running the Collaborative Memory Manager (cmm module), this will typically be 0, as the CMM will monitor and fulfill the hypervisor's loan requests.

/proc/ppc64/lparcfg 1.8 additional values (2/2)

- `backing_memory`
  - The number of bytes of main storage that is backing the partition logical address space. In the case of a dedicated memory partition this is the size of the partition's logical address space.
- `cmo_enabled`
  - If Active Memory Sharing is enabled for the partition, this is set to 1. For dedicated memory partitions, this is 0.
- `cmo_faults`
  - Displayed only for shared memory partitions. Indicates the total number of times the partition has accessed a page in memory which was paged out to disk by firmware, requiring it to be paged back in. If the Collaborative Memory Manager is disabled, this value may be large. If it is enabled (default setting for most Linux distributions), this number is typically small. If this value is large and is increasing, it may be an indication that the partition's shared memory pool has too high of an overcommit ratio, in which case you may need to assign additional physical memory to the shared memory pool.
- `cmo_fault_time_usec`
  - Displayed only for shared memory partitions. Indicates the total amount of time in microseconds the partition has had a virtual processor blocked in order for firmware to page in data. Directly related to cmo_faults.
- `cmo_primary_psp`
  - Displayed only for shared memory partitions. Partition ID of the primary paging VIOS.
- `cmo_secondary_psp`
  - Displayed only for shared memory partitions. Partition ID of the secondary paging VIOS. If there is no secondary paging VIOS, this will be set to 65535.
- `cmo_page_size`
  - Displayed only for shared memory partitions. Physical page size in bytes.

/proc/ppc64/lparcfg 1.9 based values

- `physical_procs_allocated_to_virtualization`
  - The number of physical platform processors allocated to processor virtualization. This is a physical system attribute and has no bearing on the Linux partition.
- `max_proc_capacity_available`
  - The maximum processor capacity percentage that is available to the partition's shared processor pool.
- `entitled_proc_capacity_available`
  - The entitled processor capacity percentage available to the partition's pool.
- `dispatches`
  - Virtual Processor Dispatch Counter. Counter that is incremented each time a virtual processor is dispatched/preempted.
- `dispatch_dispersions`
  - Virtual Processor Dispatch Dispersion Accumulator. Incremented on each virtual processor dispatch if the physical processor differs from that of the last dispatch.

Change number of SMT threads

Turn SMT on/off from Linux

- SMT can be disabled at boot time by adding this command to the append string of the kernel boot parameters:

  `smt-enabled=off`

- To determine the current SMT state:

  `ppc64_cpu –smt`

- To set the SMT state dynamically:
  - `ppc64_cpu --smt={on|off}`
  - `ppc64_cpu --smt=<number>`
    - Possible settings for `<number>`
      - Power6: 1, 2
      - Power7: 1, 2, 4
      - Power8: 1, 2, 4, 8

- `ppc64_cpu` is part of the “`powerpc-utils`“ RPM package (installed by default)

Optimizing for Linux on Power

GCC versions for RHEL, SLES and Ubuntu

Facts:

- Linux distributors select a base GCC version as their “default“ version.
- During the lifetime of that Linux enterprise distribution version they typically stick with that specific GCC version.
- Unfortunately, GCC development keeps going at a fast pace.
- Newer GCC technology and features mostly only available with latest version of GCC.
- Selective new GCC features and bug fixes from the current GCC development version are being backported to their base GCC versions.

Consequence:

- Best exploitation of newest Power systems not always guaranteed with older GCC versions!

GCC versions that come with RHEL, SLES and Ubuntu

Red Hat Enterprise Linux

- RHEL 6
  - gcc-4.4.4-13.el6.ppc64.rpm
- RHEL 6 Update 1
  - gcc-4.4.5-6.el6.ppc64.rpm
- RHEL 6 Update 2
  - gcc-4.4.6-3.el6.ppc64.rpm
- RHEL 6 Update 3
  - gcc-4.4.6-4.el6.ppc64.rpm
- RHEL 6 Update 4
  - gcc-4.4.7-3.el6.ppc64.rpm
- RHEL 6 Update 5
  - gcc-4.4.7-4.el6.ppc64.rpm
- RHEL 6 Update 6
  - gcc-4.4.7-11.el6.ppc64.rpm

- RHEL 7
  - gcc-4.8.2-16.el7.ppc64.rpm

SUSE Linux Enterprise Server

- SLES 11
  - gcc-4.3-62.198.ppc64.rpm
- SLES 11 Service Pack 1
  - gcc-4.3-62.198.ppc64.rpm
- SLES 11 Service Pack 2
  - gcc-4.3-62.198.ppc64.rpm
- SLES 11 Service Pack 3
  - gcc-4.3-62.198.ppc64.rpm

- SLES 12
  - gcc-4.8-6.189.ppc64le.rpm

Ubuntu

- 14.04
  - gcc-4.8_4.8.2-19ubuntu1_ppc64el.deb
- 14.10
  - gcc-4.9_4.9.1-16ubuntu6_ppc64el.deb

IBM Software Development Kit for Linux on Power

- The IBM Software Development Kit for Linux on Power (SDK) is a free, Eclipse-based Integrated Development Environment (IDE) and integrates
  - C/C++ source development with the Advance Toolchain
  - Post-Link Optimization
  - classic Linux performance analysis tools, including Oprofile, Perf and Valgrind

The IBM SDK for Linux on Power package includes:

- IBM Advance Toolchain for Linux on Power integration, Versions 7.0-5, 7.1-0, and 8.0-0
- IBM SDK for Linux on Power, Version 1.6.0
- Feedback Directed Program Restructuring (FDPR), Version 5.6.2-6b
- Pthread Monitoring tool for Linux on Power (pthread-mon), Version 0.5.10-1
- IBM SDK Java Technology Edition Version 7.1
- IBM POWER8 Functional Simulator

IBM Advance Toolchain for PowerLinux

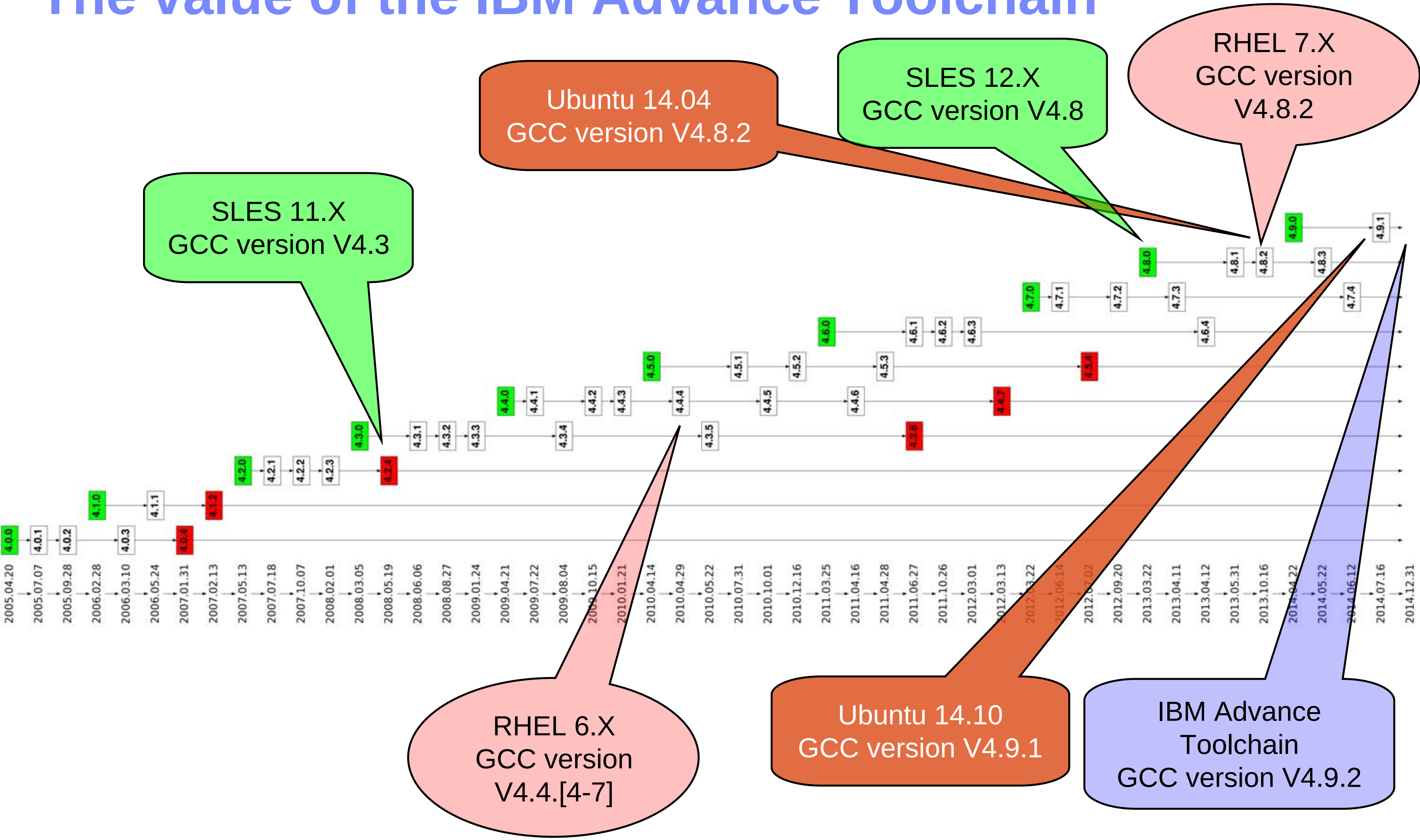
URLs:

- PowerLinux Community wiki
- IBM Advance Toolchain for PowerLinux Documentation
- Improving performance with IBM Advance Toolchain for PowerLinux

Description:

- The IBM Advance Toolchain for PowerLinux™ provides early and easy access to libraries and the latest compiler technologies for Linux distributions.
- Over time, these libraries and latest compiler technologies are integrated into the shipping distributions.
- However, the IBM Advance Toolchain for PowerLinux contains the latest tested and supported GNU Compiler Collection (GCC) compiler versions, tailored for Power systems, and packaged together with an expanding set of processor-tuned libraries, allowing you to take advantage of the latest technology without waiting..

The value of the IBM Advance Toolchain



Ubuntu 14.04
GCC version V4.8.2

SLES 12.X
GCC version V4.8

RHEL 7.X
GCC version
V4.8.2

SLES 11.X
GCC version V4.3

RHEL 6.X
GCC version
V4.4.[4-7]

Ubuntu 14.10
GCC version V4.9.1

IBM Advance
Toolchain
GCC version V4.9.2

Optimized compile of PostgreSQL V9.2.6

- Customer in Karlsruhe/Germany required optimized PostgreSQL V9.2.6 binary RPMs for production databases for RHEL 6.X.
- http://yum.postgresql.org/ provides Enterprise PostgreSQL binary RPM packages but only for x86 and x86_64 platforms.
- Idea was now to recompile those original source RPM files on RHEL 6.X for Power.

Recompile on PowerLinux:

- Recompile went very smooth and out-of-the-box, no source code changes at all required.
- Also repackaging as RPM files on PowerLinux/RHEL required no changes.
- Compilation of highly optimized binary RPMs with
  - Advance Toolchain with GCC V4.8.2

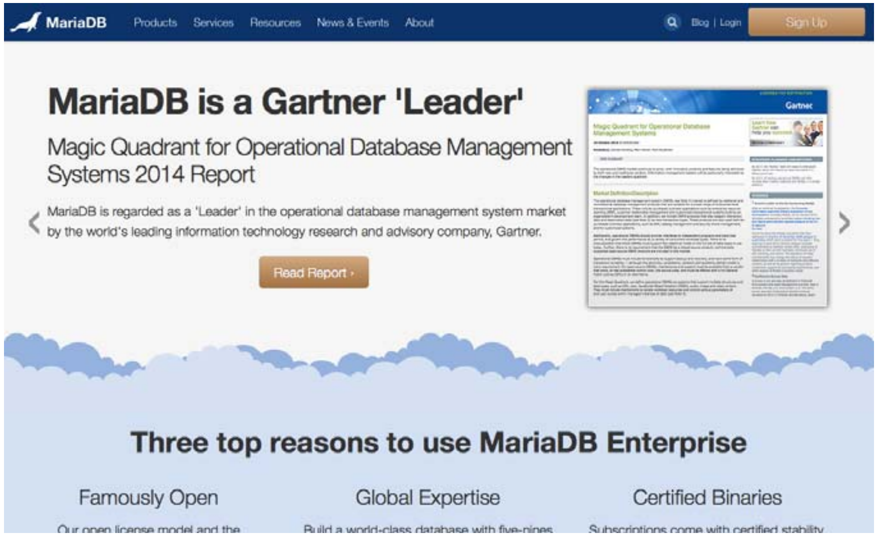
  instead of
  - Base V4.4.7 GCC as part of RHEL 6.5

  improved the run times (measured with pgbench) by 30-35% !!

MariaDB Corporation and MariaDB

- MariaDB Corporation, formerly SkySQL
  - Strong team of MariaDB/MySQL experts
  - Provides support and services for MariaDB, MySQL and derived databases
  - Strong Supporter of MariaDB Foundation

- MariaDB
  - Compatible with MySQL
  - Superior replication and scalability for cloud
  - Addresses Big Data and IoT needs
  - MariaDB is becoming the leading database platform for cloud
  - MariaDB is adopted by key open source communities



MariaDB is a Gartner 'Leader'

Magic Quadrant for Operational Database Management Systems 2014 Report

MariaDB is regarded as a 'Leader' in the operational database management system market by the world's leading information technology research and advisory company, Gartner.

Three top reasons to use MariaDB Enterprise

Famously Open

Our open license model and the

Global Expertise

Build a world-class database with five-nines

Certified Binaries

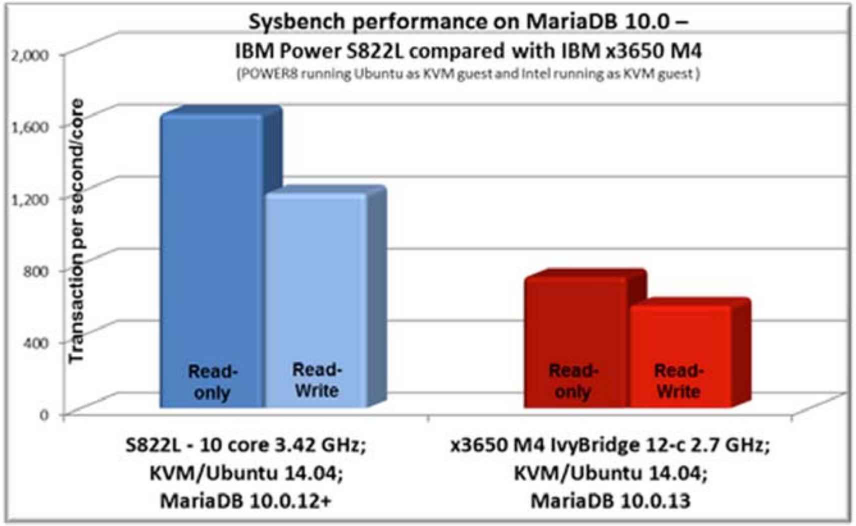
Subscriptions come with certified stability

mariadb.com

Optimizations and economic impact

- MariaDB optimized for System z and POWER8

- Power8 – Big + Little Endian architectures have been made part of the build system of MariaDB



- Preliminary benchmarking results shows that MariaDB performance on POWER8 per core is 2.2× more than on x86!

- MariaDB built with “IBM’s Advance Toolchain for POWER8”
- MariaDB makes full use of the 8 threads per core provided by POWER8:
  - 12 cores @4.x Ghz × 8 threads per core = 96 parallel processing threads
  - Intel – only 2 threads per core

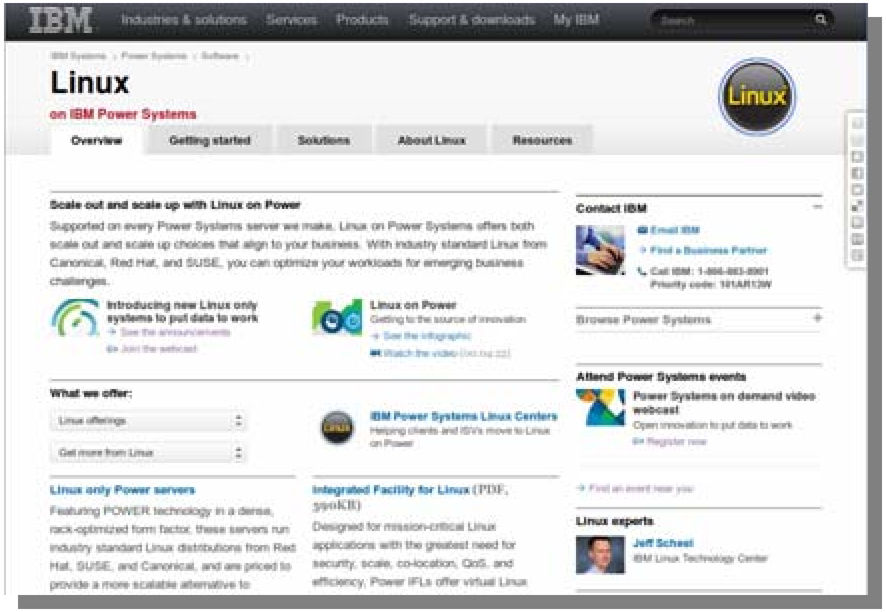
Resources, Links

Where to find more information?

Power Systems Linux Portal

(Product Information)

www.ibm.com/systems/power/software/linux/



IBM Industries & solutions Services Products Support & downloads My IBM

IBM Systems › Power Systems › Software ›

Linux

on IBM Power Systems

Overview | Getting started | Solutions | About Linux | Resources

Scale out and scale up with Linux on Power

Supported on every Power Systems server we make, Linux on Power Systems offers both scale out and scale up choices that align to your business. With industry standard Linux from Canonical, Red Hat, and SUSE, you can optimize your workloads for emerging business challenges.

Introducing new Linux only systems to put data to work
→ See the announcements
Join the webcast

Linux on Power
Getting to the source of innovation
→ See the infographic
Watch the video

What we offer:

Linux offerings

Get more from Linux

IBM Power Systems Linux Centers
Helping clients and ISV's move to Linux on Power

Linux only Power servers
Featuring POWER technology in a dense, rack-optimized form factor, these servers run industry standard Linux distributions from Red Hat, SUSE, and Canonical, and are priced to provide a more scalable alternative to

Integrated Facility for Linux (PDF, 390KB)
Designed for mission-critical Linux applications with the greatest need for security, scale, co-location, QoS, and efficiency. Power IFLs offer virtual Linux

Contact IBM
Email IBM
→ Find a Business Partner
Call IBM: 1-866-883-8901
Priority code: 101AR13W

Browse Power Systems

Attend Power Systems events
Power Systems on demand video webcast
Open innovation to put data to work
Register now

→ Find an event near you

Linux experts
Jeff Scheel
IBM Linux Technology Center

OpenPOWER

Home | About Us | Membership | Technical | News/Events | Get Involved | Member Login

Get Involved Today!

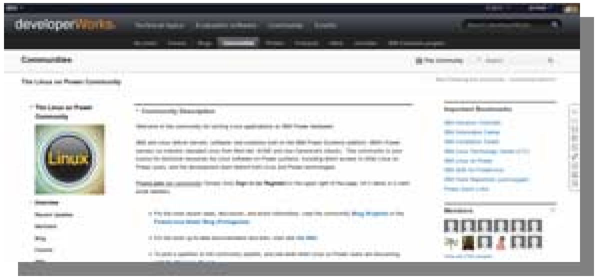
Join OpenPOWER

OpenPOWER Overview

Latest News

The OpenPOWER Foundation

http://openpowerfoundation.org/



developerWorks

Communities

The Linux on Power Community

Community Description

Important Bookmarks

Members

The PowerLinux Community

(developerWorks)

plus.google.com/communities/100156952249293416679

@ibmpowerlinux

Links

- **IBM Wikis:**
  - http://www.ibm.com/developerworks/wikis/
- **All commands for scanning and deleting virtual SCSI devices:**
  - SCSI - Hot add, remove, rescan of SCSI devices
- **RHEL 6:** Adding/Removing a Logical Unit Through rescan-scsi-bus.sh
- **SLES:** HOWTO: Add or Resize a LUN without restarting SLES or OES Linux
- **IBM PowerLinux Wiki**
  - http://www.ibm.com/developerworks/wikis/display/LinuxP/Home
- IBM Redbook SG24-7338:
  Tuning Linux OS on System p – The POWER of Innovation
- IBM Redbook SG24-7286:
  Deploying Mission Critical Applications with Linux on POWER
- IBM Redbook SG24-7499:
  Virtualizing an Infrastructure with System p and Linux

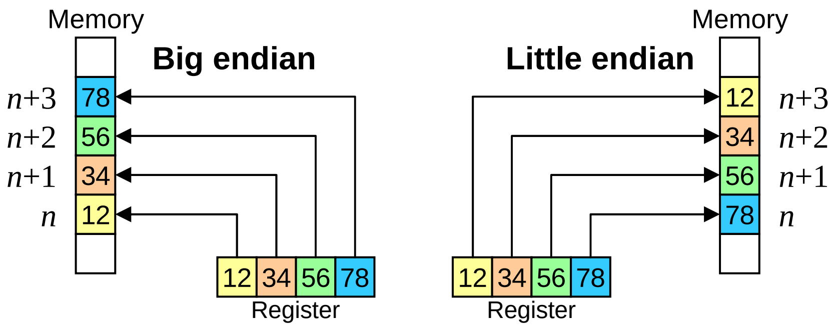
Questions ?

Thank you for your attention !

Backup slides

Endianness

Why do I care about endianness?



- Linux on Power has chosen to exploit little endian (LE) processor mode based on OpenPOWER partner feedback instead of big endian (BE).
  - Eases the migration of applications from Linux on x86.
  - Enables simple data migration from Linux on x86.
  - Simplifies data sharing (interoperability) with Linux on x86.
  - Improves Power I/O offerings with modern I/O adapters and devices, e.g. GPUs.
- Creation of an LE operating system for Linux on Power means creating a whole new software “platform” (**ppc64le**) (in addition to BE **ppc** (32-bit) and BE **ppc64** (64-bit)).
- LE distributions for Linux on Power **does NOT mean** x86 applications magically run: applications must still be compiled for Power.
- Power8 CPU can be either big or little endian
  ➔ mixed endianness (big and little) on same system will be possible.

Linux distributions for Power Systems

Linux support for POWER

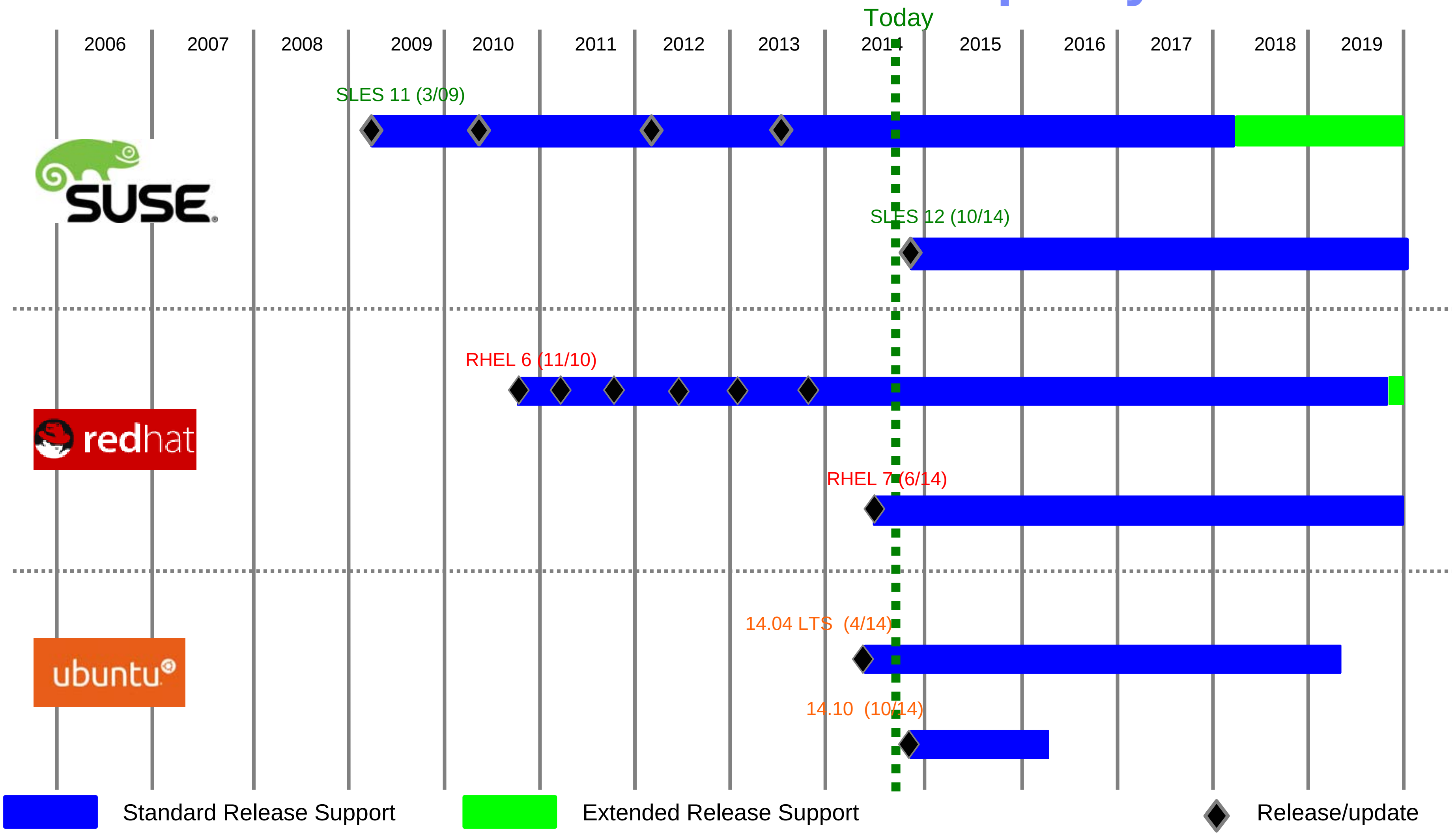
> - ➢ Built from the same source as x86
> - ➢ Delivered on the same schedule as x86
> - ➢ Supported at the same time as x86

- RHEL 7
  - POWER8 (native mode) and POWER 7/7+
- RHEL 6
  - POWER8 supported with U5 (P7-compatibility mode)
  - Full support of POWER6 and POWER7
    (native mode)
- Fedora
  - Fedora 16 was first release to re-launch POWER
  - Fedora 20 has POWER8 support
- Supported add-ons
  - JBoss
  - High Performance Network Add-on

- SLES 12
  - POWER8 (native mode) and POWER 7/7+
- SLES 11
  - POWER8 with SP3 (P7-compatibility mode)
  - POWER7+ encryption, RNG accelerators with SP3
  - Full support of POWER7 (native mode)
- openSUSE
  - openSUSE 12.2 re-launched for IBM POWER
  - openSUSE 13.2 includes POWER8 support
- Supported add-ons
  - SUSE Linux Enterprise High Availability Extension

- Ubuntu 14.10
  - POWER8 enabled (native mode)
- Ubuntu 14.04
  - POWER8 enabled (native mode)
  - No official support for POWER7+ and older systems
  - No support for 32-bit applications. 64-bit only.
  - Supported in KVM only at this time
- Supported add-ons
  - JuJu Charms
  - MaaS (Metal as a Service)
  - Landscape
- Debian
  - Debian community now supports Power as of Sid release

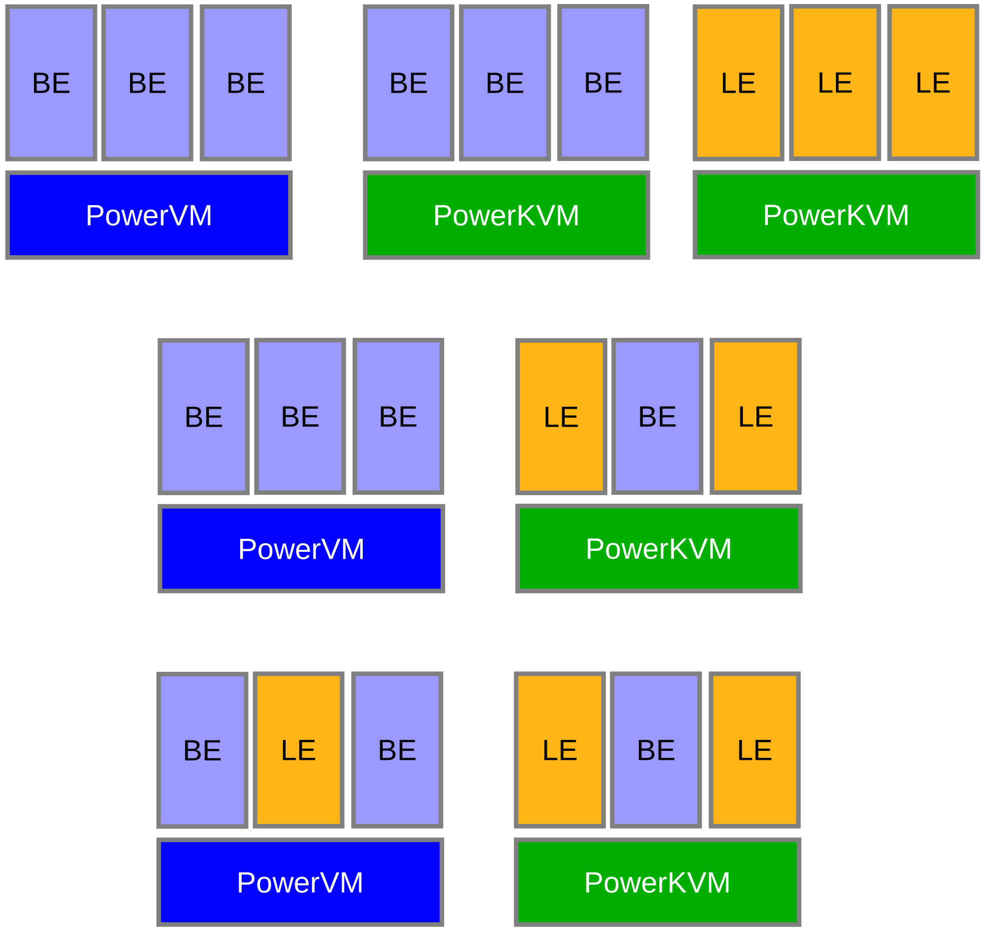
Linux on Power has Linux “release parity” with x86



See for more details:

- Red Hat lifecycle information - https://access.redhat.com/support/policy/updates/errata/
- SUSE lifecycle information – http://support.novell.com/inc/lifecycle/linux.htm/
- Ubuntu lifecycle information - https://wiki.ubuntu.com/Releases

Hypervisor support for endianness



April 2014

BE BE BE
PowerVM

BE BE BE
PowerKVM

LE LE LE
PowerKVM

4Q 2014

BE BE BE
PowerVM

LE BE LE
PowerKVM

Strategic outlook

BE LE BE
PowerVM

LE BE LE
PowerKVM

Transition to strategic state may involve multiple interim steps as product testing completes.

Migrating Linux/x86 applications to Linux on Power

Script languages and interpreted languages

- All script languages and interpreted languages should be platform-independent once they are compiled for the particular platform.
- Also compiled code should be platform-independent (e.g., Perl, Python etc.).
- Examples include:
  - Perl
  - Python
  - PHP
  - Ruby
  - Lua
  - Tcl
  - etc.

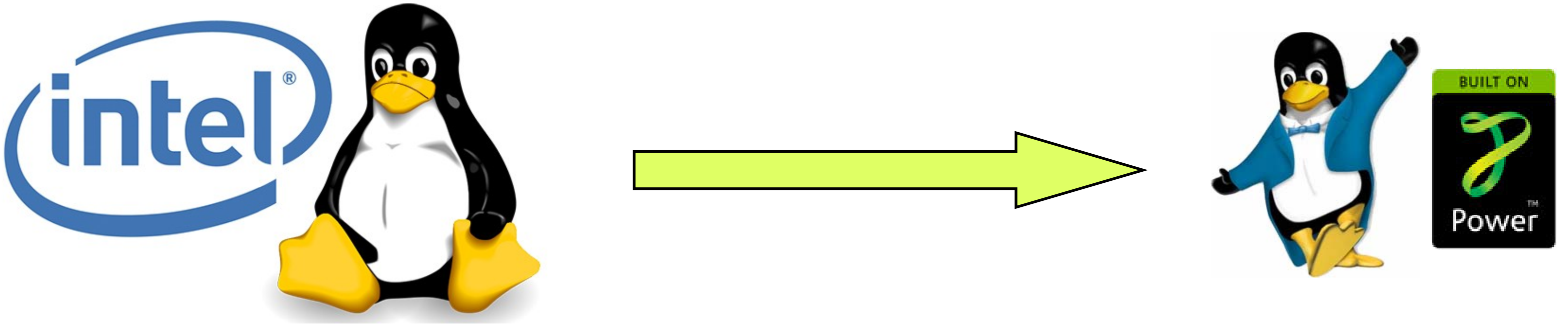
Java migrations

- Java compiled byte-code is platform-independent and thus portable across different platform if the Java specification has been adhered to, i.e., no APIs/syscalls beyond the specification have been used.
- For PowerLinux the Java JVM options are:
  - IBM JVM
  - OpenJDK
- For Linux/x86 multiple different JVMs are available.
- Differences in behavior between the IBM JVM and the Oracle JVM exist.

- Problems in migrating Java code typically arise only if Java extensions were used that are not part of the standard Java specification:
  - For instance, lots of security-relevant Java code differs between JVMs of different vendors.

- For the remainder only C/C++ code now considered!

Considered migration scenarios C/C++ (1/2)



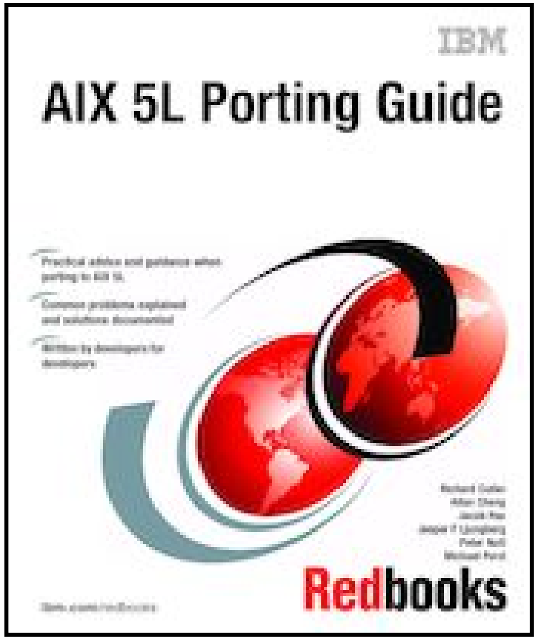
Other Linux
(not RHEL, SLES, Ubuntu) → RHEL, SLES, or Ubuntu

RHEL
SLES
Ubuntu
→
RHEL
SLES
Ubuntu

Considered migration scenarios C/C++ (2/2)

Not considered here:

- Changing the address space for the application, i.e., converting it from a 32-bit to a 64-bit application
  - Might be required if porting 32-bit application to 64-bit only like Ubuntu (ppc64le)
- A 32-bit Linux/Intel application can always be recompiled as a 32-bit Linux/Power application, no need to change anything here!
  - The exception is new ppc64le platform (e.g., Ubuntu)
- Converting a 32-bit application to 64-bit address space can present a huge challenge depending on the code quality!
- Please see the redbook "AIX5L Porting Guide" for details:



IBM

AIX 5L Porting Guide

Practical advice and guidance when porting to AIX 5L

Common problems explained and solutions documented

Written by developers for developers

[illegible]

Redbooks

ibm.com/redbooks

Dealing with endianness

Sources of endianness problems:

- Nonuniform data referencing
  - It is often featured by data type mismatches resulting from either data element casting, use of a union data structure, or the use and manipulation of bit fields.
- Sharing data across platforms
  - For example, a big-endian system retrieves database data stored by a little-endian system.
- Exchanging of data between devices of different endianness and devices on a network
  - For example, AIX on Power systems uses the big-endian model, but the PCI bus uses the little-endian model.
  - TCP/IP protocols requires data to be sent in network byte order, which is the big-endian model.

IBM Software Development Kit for PowerLinux

- The IBM Software Development Kit for PowerLinux includes a Migration Advisor to help in moving Linux applications from x86 systems to Power systems.
- The advisor uses the Eclipse C/C++ Development Tools code analysis tool.
- The code analysis tool locates potential migration problems within a project, such as source code that might produce different results when run on Power systems.
- It contains several checkers that look for code in the project that might produce a different result in Power systems.
- Warnings are displayed showing the kind of problem found.

Linux/x86 to PowerLinux application migration

PowerLinux Migration Advisor checkers:

- x86-specific compiler built-in checker
- x86-specific assembly checker
- Struct with bit fields checker
- Cast with endianness issues checker
- Linux/x86-specific API checker
- Union with endianness issues checker
- Long double usage checker
- Performance degradation checker
- Syscall not available for PowerLinux checker