
PRESENTED BY RED HAT

# LEARN. NETWORK.
# EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

# GETTING BEYOND 2 TERABYTES:

## USING GPT WITH STORAGE DEVICES

Steven Bonneville
Manager, Linux Curriculum Team
Red Hat
May 4, 2011



# A Quick Note on SI Prefixes

- kB = $10^3$ bytes
- MB = $10^6$ bytes
- GB = $10^9$ bytes
- TB = $10^{12}$ bytes

- kiB = $2^{10}$ bytes
- MiB = $2^{20}$ bytes
- GiB = $2^{30}$ bytes
- TiB = $2^{40}$ bytes

- In this talk, we'll be consistent in distinguishing between decimal and binary prefixes



# Disk Partitions

- Used for years to break large disks into smaller pieces
- Standard disklabel format in the x86/x86-64 world is used by most operating systems (“MBR” or “MSDOS”)
- 32-bit format designed many years ago

**Disks have grown too large to use this format**



# Why do we need partitions anymore, anyway?

- After all, we can make the whole disk device an LVM physical volume, and carve it up into as many logical volumes as we want:
  - **pvcreate /dev/sdb**

- “I’m not planning to use this disk with non-Linux operating systems anyway!”
  - ...we begin to get a hint about why here...

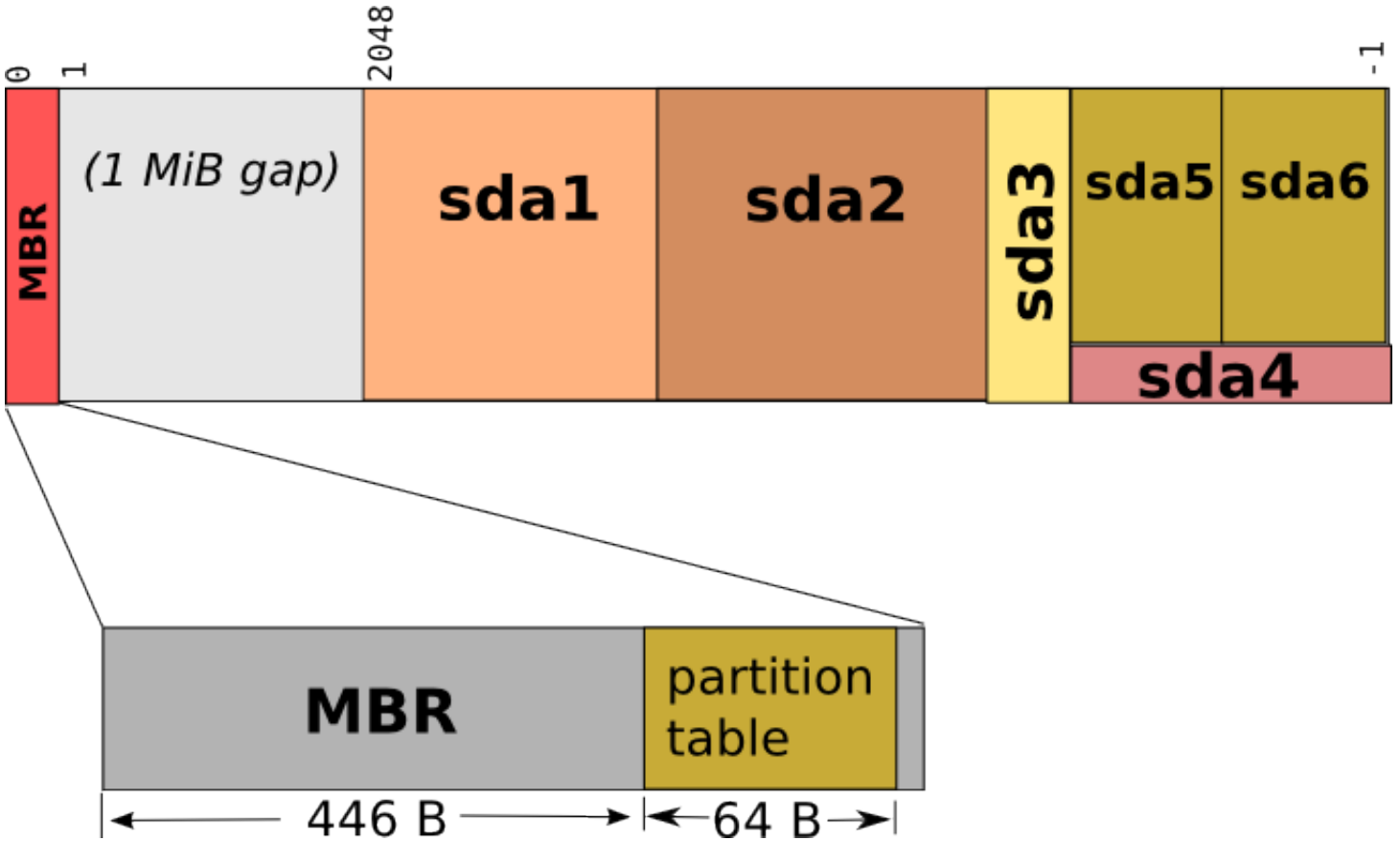


# Why do we need partitions anymore, anyway?

- Even if we aren't using the disk with other operating systems, they may still see it
  - Do you have a SAN?  Configuration mistakes happen
- Without a standard disk label it is
  - Easy to think a disk is empty when it is not
  - Harder to determine what a disk contains or is for



# “Standard” MBR / MSDOS Disklabel




(not to scale!)

# Limitations of MBR / MSDOS Disklabels

- $2^{32}$ sector limit = 2,147,483,648 kiB (2 TiB / ~2.2 TB)
- The MBR is a single point of failure
- More than four partitions is a fragile kludge
  - Primary/Extended/Logical?  Really?
- Legacy cruft that's no longer relevant (C/H/S etc.)

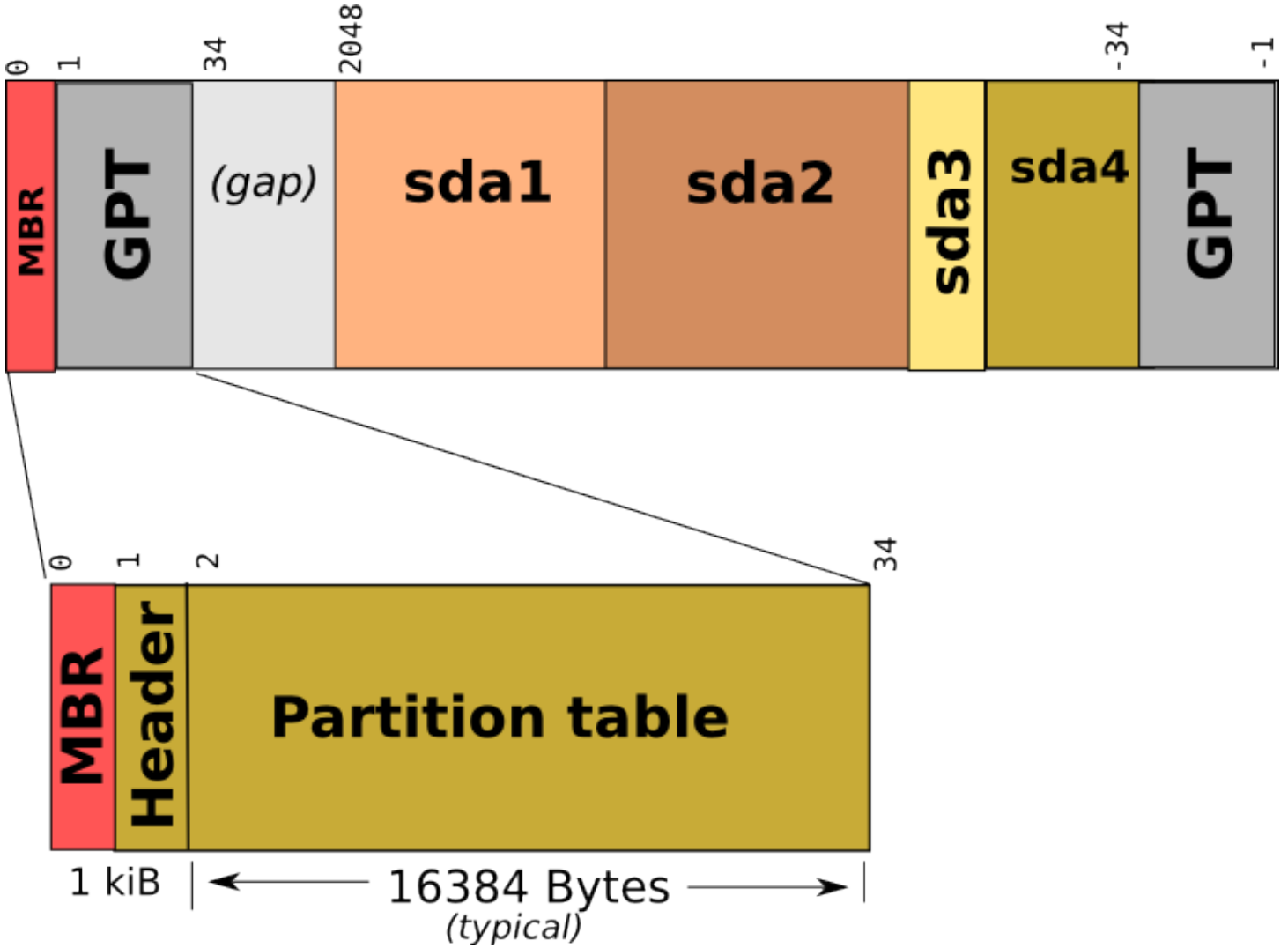


# GUID Partition Table (GPT)

- $2^{64}$ sector limit = 8 ZiB (about 9.2 billion TB)
- Complete partition table at front and back of disk
- Flexible maximum number of partitions (128 by default!)
- Room for more information about a partition's type, and a partition-unique GUID and Unicode label



# GPT Disklabel




(also not to scale!)

# GPT Disklabel

- LBA 0 is still an MBR
  - “Protective MBR”
  - Dummy partition table with one partition of type 0xEE, takes up whole disk or 2 TiB, whichever is less
- LBA 1 is header of GPT partition table
  - Defines maximum number of partitions
  - Defines size of a single partition entry
  - Checksums and disk UUID



# GPT Disklabel

- Default GPT format used normally allows up to
  - 128 partition entries, each 128 bytes in size
- Each partition entry includes the following information
  - Partition’s 64-bit starting LBA and ending LBA
  - Partition type: 128-bit UUID replaces MBR’s 16-bit type
  - Name (up to 36 UTF-16LE “code units”)
  - Some other miscellaneous info



# GUID Partition Types

- Linux/Windows data
  **EBD0A0A2-B9E5-4433-87C0-68B6B72699C7**
- Linux swap
  **0657FD6D-A4AB-43C4-84E5-0933C84B4F4F**
- Linux LVM
  **E6D6D379-F507-44C2-A23C-238F2A3DF928**
- Linux RAID
  **A19D880F-05FC-4D3B-A006-743F0F84911E**

Luckily, we do not need to memorize them!



# Using GPT Disks in Red Hat Enterprise Linux

- Normal **fdisk** is designed for MBR disks only
  - Will warn you if it sees a GPT disk and displays the “protective MBR”
- Use **parted** to work with GPT disklabels
  - Can use interactively
  - Can use from the command line



# Using parted

- Relabel the disk for GPT partitions
  - (parted) **mklabel gpt**
- Listing GPT partitions
  - (parted) **p**
- Exit parted (changes are written to disk as you go)
  - (parted) **q**



# Using parted

- Creating a basic data partition
  - (parted) **mkpart data 1G 2G**
    - Makes a new partition on **/dev/sda** named “**data**”
    - Starts **1 GiB** from the start of the disk
    - Ends **2 GiB** from the start of the disk (is **1 GiB** in size)
- Creating a swap partition
  - (parted) **mkpart swap linux-swap 2G 3G**
    - Makes a new swap partition on /dev/sda named “swap”
    - Will be labeled with the correct GUID partition type



# Using parted

- Creating an LVM physical volume partition
  - Make a normal data partition
  - Mark as LVM: **parted /dev/sda set *2* lvm on**
    - (Marks /dev/sda*2* with “Linux LVM” GUID)
- Creating a Software RAID partition
  - Make a normal data partition
  - Mark as RAID: **parted /dev/sda set *3* raid on**
    - (Marks /dev/sda*3* with “Linux RAID” GUID)



# GPT and Booting

- You can freely use GPT-formatted secondary disks
- To use a GPT disk as your boot drive, you need a system that supports uEFI boot process
  - Intel Sandy Bridge motherboards
  - Technology is still maturing



# GPT and Booting

- MBR no longer contains bootloader
- “Dead space” no longer contains bootloader
- First partition on boot disk is special **EFI System Partition (“ESP”)** formatted as FAT filesystem
  - Linux normally mounts it on **/boot/efi**
- uEFI variables determine how to find the bootloader on the ESP
  - Adjusted by **efibootmgr** utility



# Practice Exercise

- Creating a GPT disklabel and normal GPT disk partitions
- Creating Linux LVM and Software RAID partitions
- Creating large numbers of GPT data partitions



# Thank you for attending!

- Learn more about other storage topics in our course: **Red Hat Enterprise Clustering and Storage Management** (RH436)

- Please complete your survey and log out of your system before you leave

- If you have further questions, please feel free to visit the Red Hat Training booth in the Partner Pavilion to speak with a member of our Curriculum Team



# LIKE US ON FACEBOOK

www.facebook.com/redhatinc

# FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

# TWEET ABOUT IT

#redhat

# READ THE BLOG

summitblog.redhat.com

# GIVE US FEEDBACK

www.redhat.com/summit/survey