

# VMware Workstation and GSX Performance and Configuration

Tony Cannon  
Staff Engineer

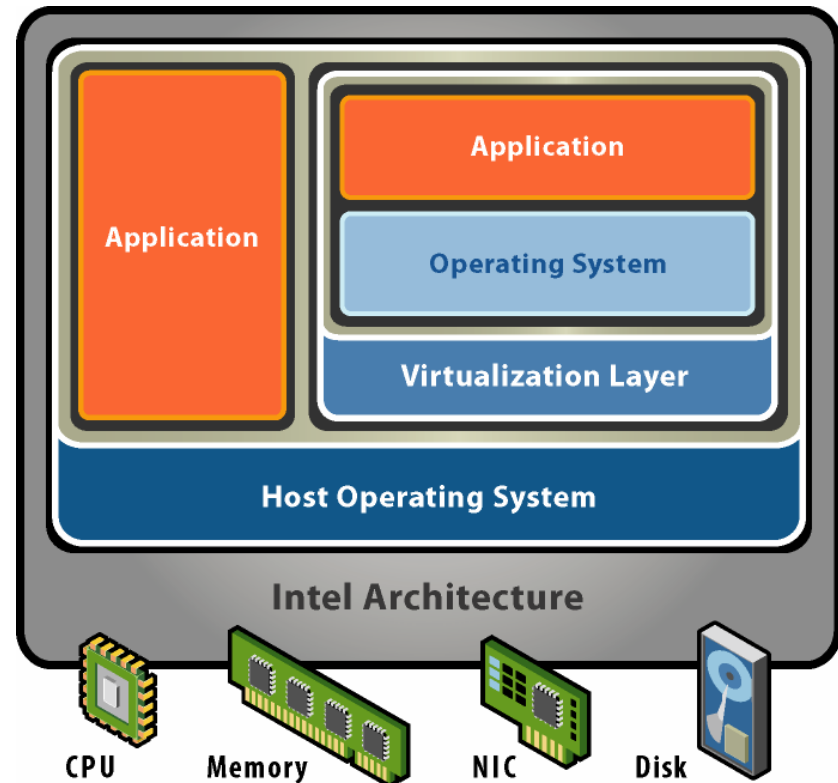


# Agenda

- **Threading Model**
- **Tour Inside Virtual Devices**
- **Questions**

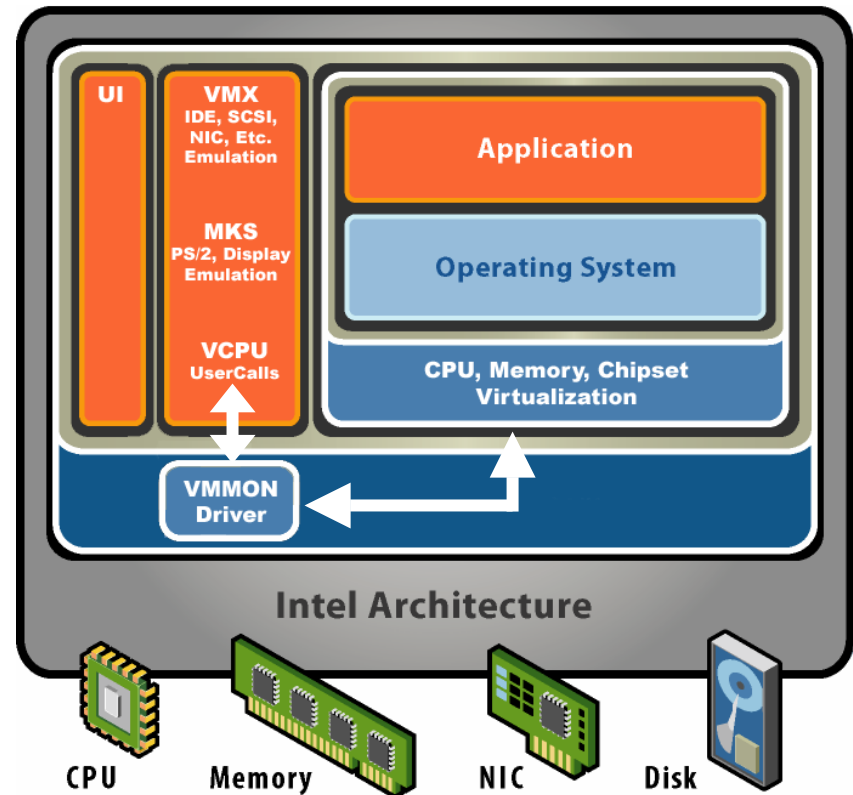
# Hosted Architecture Overview

- **Virtual Machine as an application.**
- **Resource management up to host.**
- **Used for GSX Server and Workstation products.**



# Hosted Architecture Details

- One process for each VM
- Three main threads per VM process.
  - VCPU
  - VMX
  - MKS
- Zero or more UI processes.



# Basic Performance Guide

- **Run on an SMP box**
- **Influence host via VMX process priority.**
  
- **Install the VMware Tools!**

# Inside the Virtual Network

- **Emulates AMD PC/Net 32 NIC**
- **Host drivers interpose on network stack:**
  - Windows, NDIS protocol drivers
  - Linux, put host NIC in promiscuous mode.
- **New in Workstation 5.0!**
  - VMXNet NIC
  - Or in 4.5.2: ethernet0.virtualDev = “vmxnet”

# Inside the Virtual Disk

- **Three types of virtual disks**
  - Raw Disks
  - Plain Disks
  - Sparse Disks
- **You're probably using Sparse Disks.**

# Anatomy of a Sparse Disk

- **Virtual disk sectors stored in grains**
  - 128 sectors per grain.
- **Grains added on demand to end of file.**
- **Grain Table maps sectors to grains**



Sparse Virtual Disk File Format

# Sparse Disks Disk Performance

- **Writes to unallocated grains may be slow.**
- **Accessing data spanning grain boundaries may be slow due to fragmentation.**

# Impact of Fragmentation

- **Numbers in diagram correlate data location across layers.**
- **Demonstrate different fragmentation scenarios.**
- **Simplify: Assume sectors everywhere.**



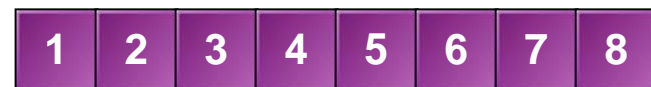
File in Guest FileSystem



Sparse Disk File Layout



Virtual Disk File on Host



Host Physical Disk

# Impact of Fragmentation

- Application tries to read entire file.

## Read File



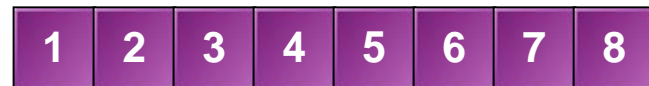
File in Guest FileSystem



Sparse Disk File Layout



Virtual Disk File on Host



Host Physical Disk

# Impact of Fragmentation

- Application tries to read entire file.
- File is fragmented into 2 pieces.



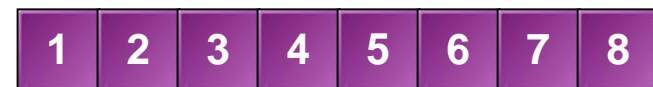
Fragmented!



Sparse Disk File Layout



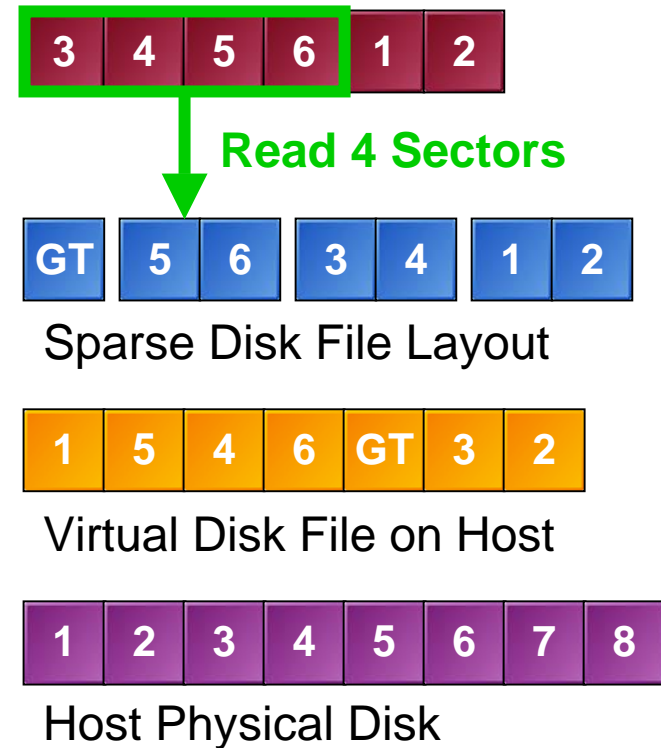
Virtual Disk File on Host



Host Physical Disk

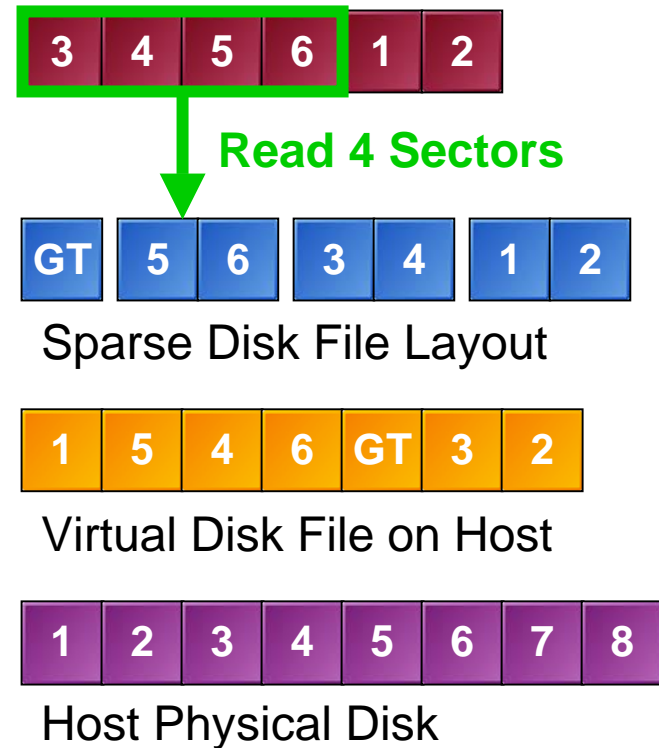
# Impact of Fragmentation

- Application tries to read entire file.
- File is fragmented into 2 pieces.
- Guest file system reads first 4 sectors.



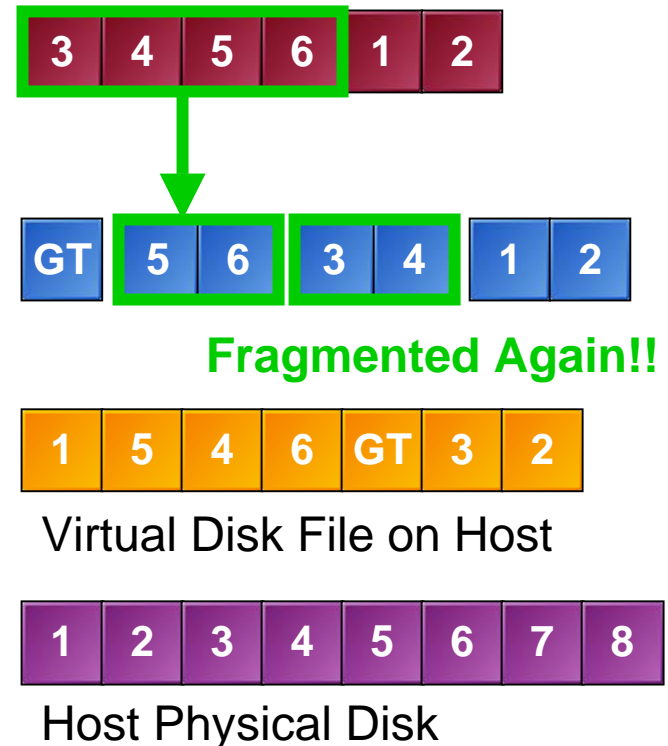
# Impact of Fragmentation

- **VMware virtual disk backend receives read.**



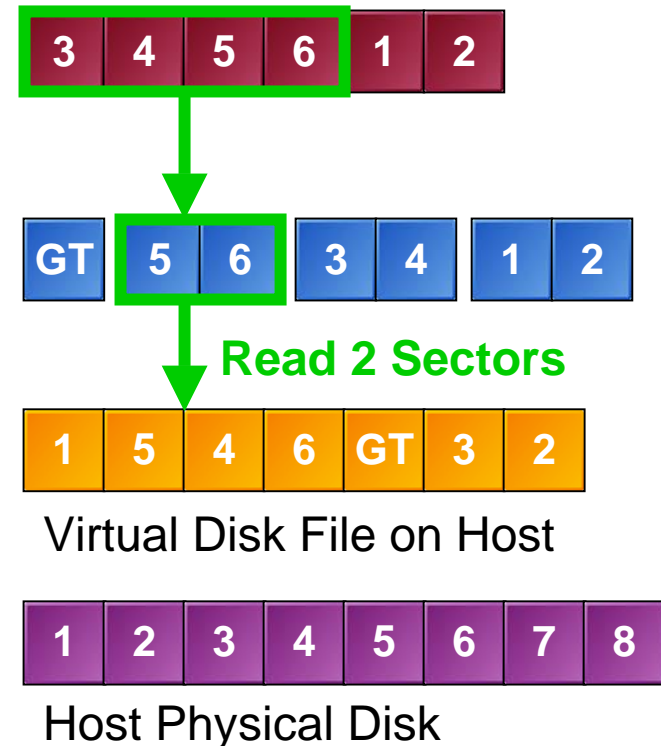
# Impact of Fragmentation

- VMware virtual disk backend receives read.
- Maps sectors to non-contiguous grains.



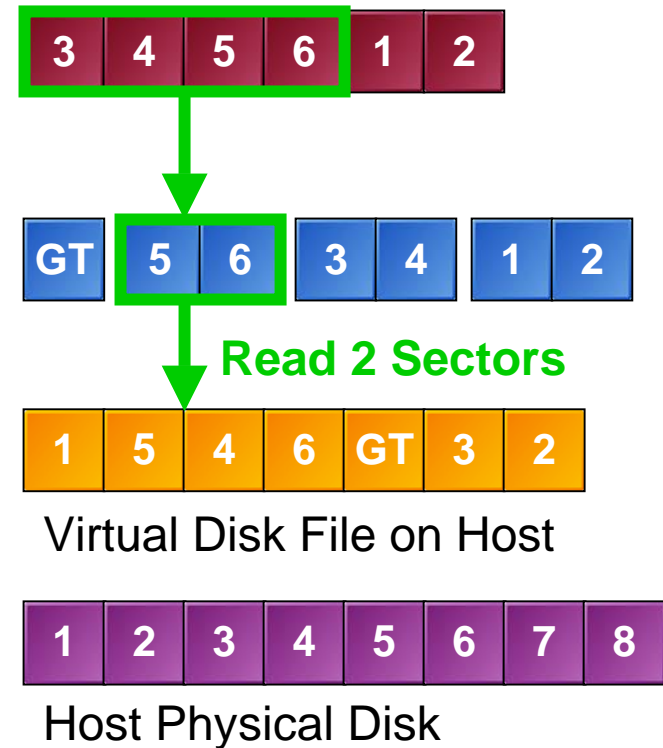
# Impact of Fragmentation

- VMware virtual disk backend receives read.
- Maps sectors to non-contiguous grains.
- Asks host to read 2 sectors from grain 0.



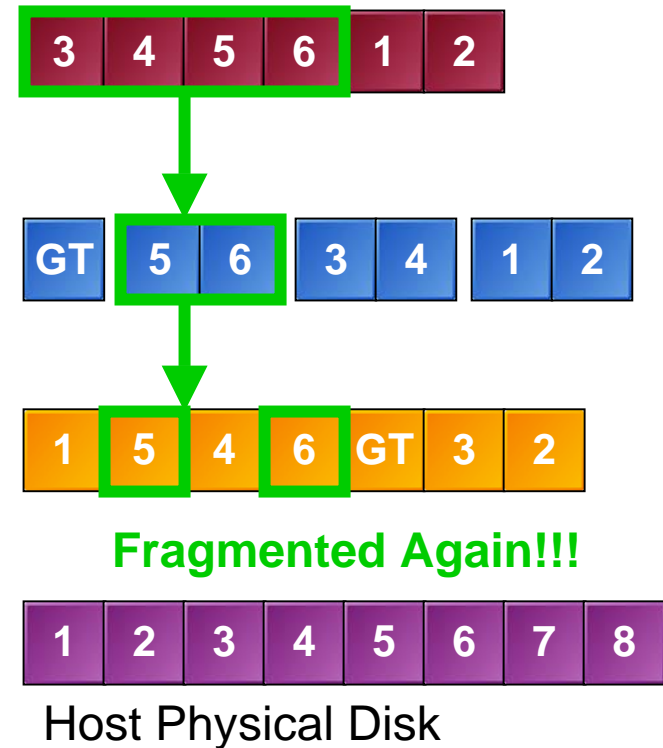
# Impact of Fragmentation

- Host filesystem receives read request.



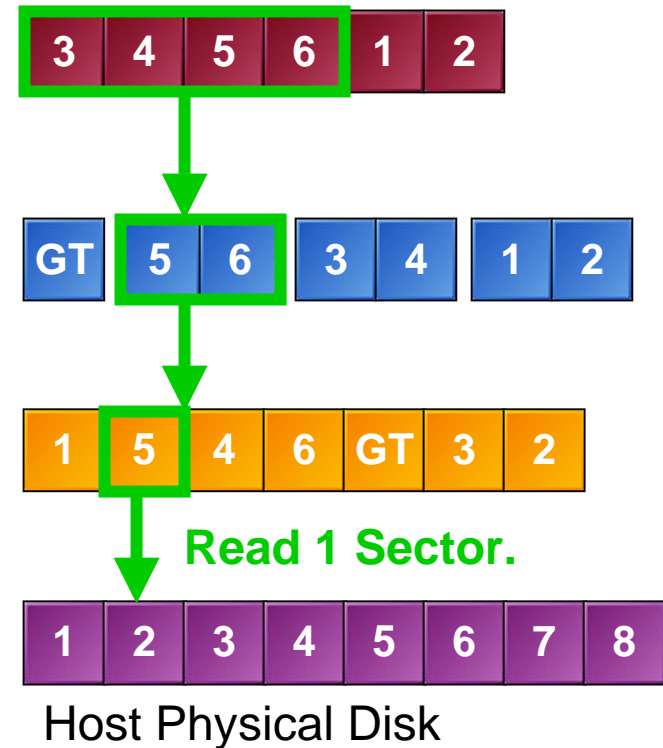
# Impact of Fragmentation

- Host filesystem receives read request.
- Maps offset/len to non-contiguous blocks.



# Impact of Fragmentation

- Host filesystem receives read request.
- Maps offset/len to non-contiguous blocks.
- Issues 1 sector read to physical device.

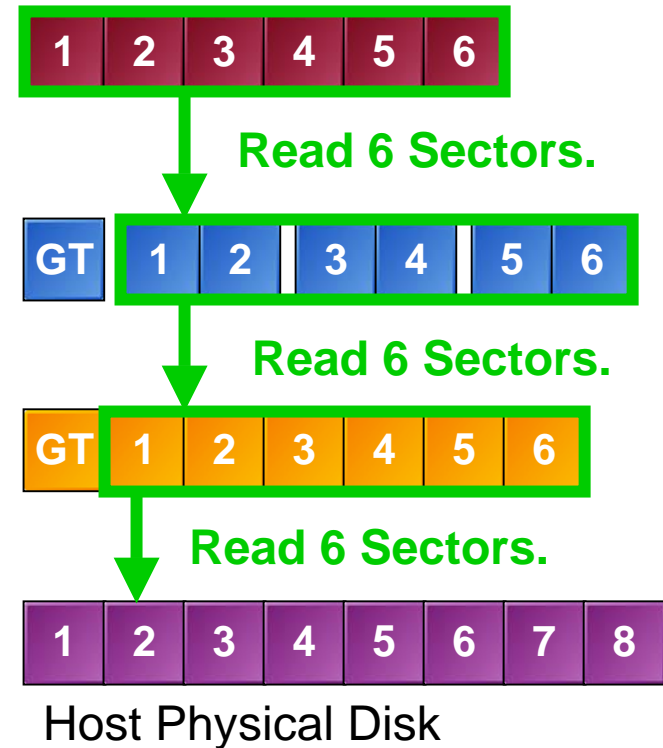


# Fixing Fragmentation

- **Goal: Put everything in order.**
- **Methodology:**
  - Defragment the guest filesystem
  - Put virtual disk grains in order
  - Defragment the host filesystem

# Zero Fragmentation

- All sectors in guest filesystem are in order.
- All grains in virtual disk are in order.
- All sectors of virtual disk on host filesystem are in order.



# More Virtual Disk Advice

- **Consider plain disks if sparse are slow.**
  - Risky: or turn on write caching on host.
- **Keep fragmentation low.**
- **Use SCSI in the guest.**
- **Use vmware-vdiskmanager**

# Inside the Virtual Mouse Device

- **PS/2 Virtual Mouse**
  - Relative packets over virtual PS/2 port.
- **VMware Absolute Pointing Device**
  - Absolute packets over virtual PS/2 port.
  - Installed with VMware Tools.

# Absolute Pointing Device Perks

- **Seamless Host/Guest mouse responsiveness.**
- **Enables Host Cursor.**
- **Better Terminal Services Support.**

# Absolute or Relative?

- **Use Relative if:**
  - Need pure PS/2 emulation
  - Need relative mouse behavior
  
- **Otherwise, use Absolute.**

# Virtual Mouse Configuration

- **Disable host cursor**
  - `svga.noHostCursor = TRUE`
- **Disable the absolute pointing device**
  - `vmmouse.present = FALSE`
- **Disable everything!**
  - `svga.noHWCursor = TRUE`

# Inside the Virtual Display

- **VGA, VESA BIOS, and proprietary SVGA interface.**
- **Why not “standard” SVGA?**
  - No such interface exists.
  - 3<sup>rd</sup> party specifications exist, but:
    - Emulating them is relatively slow.
    - Freezes out new features.

# Virtual Display Internals

- **VGA and VESA BIOS**
  - Guest draws via writes to VRAM
- **SVGA**
  - Guest calls SVGA driver, which forwards to host.
  - Updates are batched at some virtual refresh rate.
- **Host Rendering**
  - Output is displayed via host rendering calls.

# Virtual Display Advice

- **Adjust the virtual refresh rate:**
  - `svga.maxChangeTick = [100 / hz]`
- **Enable traces in SVGA modes:**
  - `svga.forceTraces = TRUE`
- **Consider KVM + Fit Guest to Window instead of fullscreen mode.**

# Questions?

# Why Install VMware Tools?

- **Optimizes guest configuration.**
- **Installs better drivers**
- **Installs virtualization utilities.**
  
- **Not required, but highly recommended.**

**Thank you!**