Memory Hole in Large Memory X86 Based Systems

By XES Product Development Team

http://www.sun.com/desktop/products

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON- INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Executive Summary

Recent increases in memory density and number of motherboard DIMM sockets have brought large memory configurations within reach of everyone. These advances expose an existing problem with PC architecture. That problem is backward 32-bit compatibility.

You may have noticed by now that an x86 based system with 2G or more of memory, the BIOS and O/S may report less memory in the machine than is actually installed. This problem occurs for 2 reasons, which are ***legacy 32-bit compatibility and*** the ***resource conflict*** caused by maintaining legacy 32-bit compatibility.

Since the system BIOS does not know what operating system will be booted it must setup the system in 32-bit mode. This means that memory and all device address spaces must be mapped below 4GB. When 4GB of memory is installed in the machine there is a resource conflict between the physical memory and the device address space. The way most BIOS resolve this problem is by carving a hole in memory at the top of the 4GB range. This hole is commonly referred to as the “PCI Hole”.

The PCI hole exists below 4GB to ensure that all 32-bit software can reach those addresses. The physical memory hidden by this hole is not usable by the operating system software and therefore is “missing” from the system. The size of the PCI hole is the total amount of PCI/AGP address space consumed for all devices as configured by the BIOS.

A configuration with 4GB of memory, an FX3000 (256M video memory) and an AGP aperture size of 256M results in about 3.0 GB of usable system memory. Reducing the AGP aperture size in BIOS setup to 64M increases the usable memory to about 3.3 GB. The following table shows the impact of the PCI hole based on video card and aperture size.

AGP Card	32MB Aperture	64MB Aperture	128MB Aperture	256MB Aperture
NVS280	3.8 GB	3.8 GB	3.6 GB	3.3 GB
FX500	3.6 GB	3.6 GB	3.6 GB	3.3 GB
FX1100	3.3 GB	3.3 GB	3.3 GB	3.0 GB
FX3000	3.3 GB	3.3 GB	3.3 GB	3.0 GB

It is important to note that the PCI hole is not an upper limit. An 8 GB memory configuration would add 4GB to each figure in the above table (e.g. 3.8 would be 7.8, etc)

Introduction

This rest of paper further explores the memory hole and other related issues. It also explores the reasons why a 64-bit processor and a 64-bit operating system won't simply fix this problem.

Before we explore the details for the apparent missing memory, let's take a look at the problems associated with breaking the 4 GB limit. First, let's do a little review of the basics.

Memory, Devices and processor address space

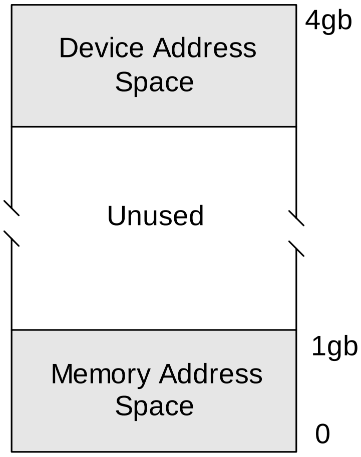
In the X86 architecture, there are 2 types of devices that are mapped into processor address space. The first type is physical memory. There is a 1:1 correlation between memory size and processor address space. A system with 512MB of physical memory will require 512MB of processor address space for it to be accessed.

The second, and perhaps less obvious, type is ***device memory*** or ***device address space***. This gets complex because there are many different types of devices. For simplicity, and purpose of this paper, we will focus on common devices such as AGP and PCI cards. Take for example a graphics card with 128MB of video memory. This is dedicated device memory that is used to store video data displayed on the screen. Video memory must be accessible by the processor and therefore mapped into processor address space. This means that the graphics card will require 128MB of processor address space.

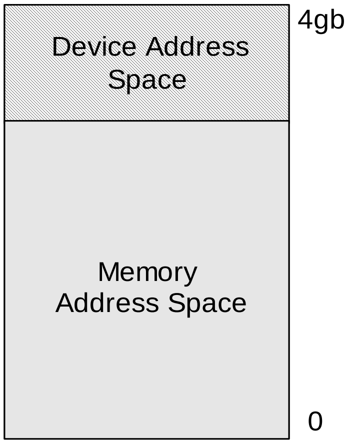
It is important to understand that both physical memory and device memory require processor address space. I used the graphics card example above but remember that there are devices that do not have memory, which require address space as well.

The device address space hole (aka the PCI hole)

The problem arises when the amount of physical memory plus the amount of device memory exceeds the processors address space. Software running on 32-bit processors is constrained to 32-bits of address space. This means that there is a **general limit** of 4 GB (2^32) for memory and device address space. Let's take a look at a typical 32-bit address map in a machine with 1GB of memory.



As you can see, there is plenty space within the processors 32-bit address range to accommodate both physical memory and device address spaces. Note that this is a simplistic representation of a memory map. Kernels can, and do, map memory in different places. As a result, a system with 1GB of physical memory will not "loose" any memory because of the required device address space. When you add 4GB of memory in a 32-bit system, you get this:



In this configuration, the address space for physical memory and device memory overlap causing a conflict. This will not work because 2 devices can't occupy the same address space. The only way to resolve this conflict (and retain compatibility) is to allow the device address space to be mapped over the memory causing a "hole" in this address range. The memory for this space can't be used and can't be remapped and will not be reported as available memory. This is why you see less available memory.

The size of the hole and devices which create it

The size of the hole will depend on several factors. The three largest factors are:

1. The amount of video memory (most have between 64Mb and 256Mb)
2. The size of the AGP Aperture (Metro default is 256MB)
3. The number of PCI devices, which have large MMIO spaces.

This means the size of the hole is dynamic, not static, as you've already seen. Different video cards will change the size of the hole. The size of the AGP aperture is variable and can be changed in the BIOS. Reducing the size of the aperture will result in an increase in available memory as shown in the table in the executive summary.

Installing PCI expansion cards may also increase the size of the hole. The impact will depend on the amount of Memory Mapped I/O (MMIO) space the PCI card(s) require. MMIO is memory space, which needs to be mapped into CPU address space. An example of this is the video memory on an AGP or PCI video card. Since this is not a published value for PCI cards, and all cards can have different MMIO requirements, it is difficult to generalize this value.

Physical Address Extensions (PAE)

I used the term "general limit" before because many 32-bit x86 processors do have the ability to address more than 4 GB. The Intel Pentium Pro™ added a feature called Physical Address Extension. PAE enables the processor to increase the number of bits used to address physical memory from 32 to 36 allowing the operating system to address up to 64 GB of memory. The AMD Opteron™ increases PAE to 40 bits. This ***workaround*** allowed 32-bit processors to access more than 4 GB of memory and has become the de facto legacy standard since.

PAE availability doesn't eliminate 4 GB limitation. On the contrary, it raises other 32-bit problems. The largest is that all software must explicitly program for and use the PAE extensions. This will affect applications, operating systems, hardware device drivers and hardware itself. Software is still running in 32-bit mode, not 36/40 bit mode and registers and native data types are still 32 bits.

It is important to note that just because an OS supports PAE does not mean the memory hole will disappear. The OS would have to remap all device address space above 4GB for that to happen. The following table defines which 32-bit Operating Systems support PAE and how to enable PAE if necessary.

OS	PAE	Notes
WinXP 32-bit	Yes	Add /PAE to boot.ini to enable PAE
Solaris X86 32-bit	Yes	Enabled by default
RHEL3-WS-32	Yes	Kernels SMP=16GB, Hugemem=64GB

Note: The standard UP kernels must be rebuilt in order to support PAE. The only standard Linux kernels which support PAE are kernel.smp and kernel.hugemem.

Doesn’t 64-bit fix this problem?

So we know now that this hole is really just a product of 32-bit address space limitations and software compatibility.

The next thought might be: Doesn’t a 64-bit CPU with 64-bits of address space solve this? Well, unfortunately, the answer is no. Despite that Opteron™ can address up to 1TB of memory and can run in 64-bit mode, it doesn’t solve the problem since the BIOS does not know which Operating System the machine will boot and must map all devices below 4GB for compatibility reasons.

As long as the BIOS must support 32-bit Operating Systems, the missing memory will still be an issue.

Hardware implications of exceeding 4 GB

Even though this is mainly a software compatibility problem, there are 32-bit related hardware problems as well.

PCI devices have registers in their configuration space called Base Address Registers (BAR for short). The BAR registers can be either 32 or 64 bit depending on the device. 32-bit registers must be mapped into address ranges under 4 GB unless the system chipset provides for an additional level of address translation.

Bus mastering PCI cards can also be impacted by the 4 GB barrier. In order for a bus mastering card to address memory above the 4 GB limit, it must support the dual address cycle or DAC for short. If a PCI device does not support DAC, its bus mastering address range may be limited to memory under 4 GB. Device hardware limitations like this will exist in both 32 and 64-bit operating systems and must be accounted for in the device drivers.

Conclusion

This paper outlines some of the problems associated with the 4 GB limit. It is for reasons like these that the PC must maintain 32-bit compatibility. The Opteron™ processor runs in both 32-bit and 64-bit modes but the BIOS must boot the system and map all devices into 32-bit address space in preparation for installing or booting 32-bit legacy software. As software continues to evolve towards 64-bit, there will be changes in this behavior. Some BIOS already have a setup option, which will allow it to map device space above 4 GB. This will require that the OS be PAE enabled or pure 64-bit.

PAE is a feature which can help during the transition to 64-bit but will be eventually become legacy itself when the transition to 64-bit is complete. The Opteron™ processor can address up to 1 TB of physical memory and 256 TB of virtual address space. This will enable Opteron based systems to efficiently use memory well beyond the current 4GB or 64GB legacy limits. However, in order for that to happen, all Operating Systems running on 64 bit PC processors must become pure, native 64-bit.

Work is being done by the BIOS and/or chip manufacturers that will either remap physical memory or move device address space in order to eliminate the hole. This memory hole may be a thing of the past soon.