

Perché se ho 4 GiB di RAM ne vedo di meno?

di Matteo Italia (matteo@mitalia.net)

Abstract

In questo articolo mi propongo di spiegare perché utilizzando un sistema operativo a 32 bit su una macchina x86 con 4 o più GiB di RAM ne vengono rilevati di meno (solitamente un po' più di 3, ma in alcuni casi anche meno).

L'idea di scrivere questo articolo nasce dal fatto che c'è molta confusione sull'argomento, complice anche il fatto che non pochi fattori entrano in gioco nella questione. Molto spesso sui forum a domande su questo problema vengono date risposte errate, incomplete o fuorvianti (io stesso prima di informarmi accuratamente e scrivere questo articolo ho fornito risposte inesatte a proposito).

Nell'articolo fornisco subito la soluzione per chi non avesse la voglia o le competenze necessarie per leggere le motivazioni del problema (paragrafo [La soluzione](#)), per poi proseguire con una trattazione esaustiva sulla causa del problema, sul funzionamento dei metodi per aggirarlo e su alcuni miti che circolano a riguardo (paragrafo [La causa del problema e il funzionamento del PAE](#)).

La trattazione si riferisce, come già detto, a CPU x86 (IA32) e x86-64 (AMD64); per quanto riguarda il sistema operativo faccio riferimento soprattutto a Windows 2000 e successivi e di tanto in tanto a Linux (kernel 2.6).

La soluzione

Se la macchina è dotata di processore a 64 bit (AMD64) **la scelta migliore è installare un sistema operativo a 64 bit**, in grado di supportare tutta la memoria installata senza bisogno di trucchi strani; in alternativa è necessario attivare il PAE, come indicato di seguito.

Sotto Windows

È importante notare che il supporto di Windows per il PAE è disponibile su Windows 2000 Advanced Server e Datacenter Server e su tutte le versioni di Windows successive, ma consente di supportare tutti e i 4 GiB di RAM **solo su Windows Server 2003** nelle versioni **superiori alla standard** e su **Windows Server 2008 Enterprise e Datacenter edition** (le motivazioni di questo sono affrontate in seguito, nella sezione [Le limitazioni di Windows](#)).

Per attivare il PAE su sistemi pre-Windows Vista bisogna aprire il file boot.ini (situato solitamente nella cartella radice del drive in cui è installato il sistema operativo) e aggiungere in coda all'opzione relativa al sistema operativo su cui intendiamo agire l'opzione /PAE. Ad esempio se il file parte così:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003 per Small
Business Server" /noexecute=optout /fastdetect
```

andrà modificato in questa maniera:

Matteo Italia - Perché se ho 4 GiB di RAM ne vedo di meno?

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003 per Small
Business Server" /noexecute=optout /fastdetect /pae
```

Importante: prima di effettuare modifiche al file boot.ini effettuatene una copia di backup, in modo da poterla ripristinare da un CD Live di Linux, da BartPE o dalla modalità provvisoria di Windows qualora qualcosa andasse storto.

Da Windows Vista in poi il file boot.ini non viene più utilizzato; per attivare PAE bisogna impartire da un prompt con privilegi elevati il comando

```
BCDEDIT /set ForceEnable PAE
```

Da Windows XP SP2 in poi PAE è attivato di default sui processori che offrono la funzionalità dell'NX bit per supportare DEP (*Data Execution Prevention*, una funzionalità di sicurezza in grado di bloccare numerosi exploit) ([MSDN3]); in ogni caso, come già detto, su Windows XP e Vista **l'abilitazione di PAE non consentirà comunque la risoluzione del problema dei 4 GiB** (si veda a questo proposito in seguito il paragrafo [Le limitazioni di Windows](#)).

Su Windows Vista dal Service Pack 1 la quantità di memoria RAM segnalata nella maggior parte delle finestre è quella corretta, ma di fatto la quantità utilizzata dal sistema è inferiore, come specificato in [MSKB1]; il problema insomma persiste, ma viene mascherato.

Se, dopo aver attivato PAE, il sistema operativo non si avvia, è sufficiente avviare Windows in modalità provvisoria, annullare le modifiche fatte e riavviare. Per le versioni di Windows anteriori a Vista basta ripristinare boot.ini allo stato precedente, mentre nel caso di Windows Vista e successivi è necessario eseguire da un prompt con privilegi elevati il comando

```
BCDEDIT /set ForceDisable PAE
```

Sotto Linux

Linux include il supporto per PAE dalla versione 2.6 del kernel; molte distribuzioni recenti forniscono di default il kernel con il supporto per PAE attivato. Se così non fosse è necessario recuperare un kernel con il supporto per il PAE incluso, operazione che varia a seconda della distribuzione; come linea guida si tenga conto che spesso i kernel indicati per i server sono dotati di PAE abilitato (come nel caso di Ubuntu, a proposito del quale si veda [GEEKIT1]).

In ogni caso ribadisco che, se il proprio processore è a 64 bit, il passaggio alla versione a 64 bit della propria distribuzione Linux è caldamente consigliato, ormai praticamente tutti i pacchetti a 32 bit sono stati portati a 64 bit (anche se ogni tanto capita di scontrarsi con qualche incompatibilità), e buona parte del software non “impacchettato” è compilabile senza problemi a 64 bit.

La causa del problema e il funzionamento del PAE

Non è strettamente necessario leggere questa sezione, ma chi si interessa al funzionamento interno del PC e ai motivi di quanto esposto sopra la può trovare interessante.

Nota sulle unità di misura: B sta per byte (ossia 8 bit nell'architettura x86), per i multipli utilizzo i prefissi per i multipli binari (KiB, MiB, GiB, eccetera) come specificato dallo standard IEC 60027-2 2ª edizione (per una spiegazione si veda [IEC1]).

32 bit non bastano

I processori x86 a 32 bit indirizzano la memoria con, appunto, puntatori a 32 bit. Questo significa che la massima quantità di memoria indirizzabile è 2^{32} B = 4294967296 B = 4 GiB. Apparentemente quindi 4 GiB di RAM dovrebbero essere visibili; perché allora spariscono?

Il punto è che lo spazio d'indirizzi fisico del processore (quello a cui fa riferimento in ultima istanza quando deve accedere a della memoria) non comprende solo la RAM, ma riserva anche un certo spazio per diverse periferiche che richiedono di essere mappate in memoria, ad esempio la scheda video, le periferiche PCI, il BIOS, eccetera. Di conseguenza lo spazio di indirizzi disponibile per la RAM è di 4 GiB a cui va sottratta la parte impiegata per le periferiche, che hanno la precedenza sulla RAM.

In ultima analisi, quindi, la RAM sparita non viene vista perché lo spazio di indirizzi a 32 bit, in parte occupato da indirizzi riferiti a periferiche, non è sufficiente ad indirizzarla tutta.

PAE

Una soluzione a questo problema è stata approntata da Intel sin dai tempi del Pentium Pro (1995) ([WIKI1]) con l'introduzione del PAE (Physical Address Extension). Abilitando la modalità PAE il processore diventa in grado di indirizzare lo spazio di indirizzi fisico utilizzando 36 bit invece dei normali 32 (anche se così facendo introduce una lieve penalizzazione delle prestazioni); questo consente di gestire uno spazio di indirizzi fisico di 2^{36} B = 68719476736 B = 64 GB.

I sistemi che supportano il PAE a livello hardware organizzano lo spazio d'indirizzi fisico in maniera tale da funzionare senza problemi anche senza che il sistema operativo supporti il PAE, piazzando nei primi 4 GiB di spazio d'indirizzi gli indirizzi dedicati alle periferiche e tutta la RAM che riescono a farci stare, e sistemando la RAM che avanza oltre il limite dei 4 GiB, disponibile per i sistemi operativi in grado di fare uso del PAE. Uno schema (ripreso da [ONT2]) può far capire meglio la questione:

GiB	Indirizzo	Uso degli indirizzi	
64,0	0x1000000000	Non impiegato	Spazio di indirizzi visibile con indirizzamento a 36 bit
5,5-63,5	0x1600000000 - 0xFD00000000	Non impiegato	
5,0	0x1400000000	Non impiegato	
		Non impiegato	
4,0	0x1000000000	RAM	Spazio di indirizzi visibile con indirizzamento a 32 bit
		Periferiche (BIOS, scheda video, bus PCI, ...)	
3,0	0x0C00000000	RAM	
		RAM	
2,0	0x0800000000	RAM	
1,0	0x0400000000	RAM	
0,0	0x0000000000	RAM	

Il sistema in questione è equipaggiato con 4 GiB di RAM ed impiega 0,5 GiB di spazio di indirizzi per le periferiche; questi 0,5 GiB sono collocati subito sotto al limite di 4 GiB, e il resto della zona sotto i 4 GiB viene utilizzato per la RAM. Subito sopra dei 4 GiB la zona 0x1000000000 - 0x1200000000 viene impiegata per i 512 MiB di RAM rimanenti. Un sistema operativo senza supporto per il PAE vedrà le periferiche e i primi 3,5 GiB di RAM, mentre un sistema operativo con supporto per il PAE sarà in grado di vedere l'intero spazio di indirizzi a 36 bit (0x0000000000 - 0xFFFFFFFF), ottenendo così anche la RAM collocata oltre il limite dei 32 bit.

Per informazioni dettagliate sul funzionamento del PAE si veda [INTEL1].

Il ruolo della scheda madre

È possibile che, per quanto il processore supporti il PAE, la scheda madre su cui è montato non lo supporti: diverse schede madri di qualche anno fa montano chipset che non supportano il PAE, o addirittura non impiegano i piedini del processore utilizzati per i 4 bit aggiuntivi necessari per il PAE ([IG1], [ONT2]). In questo caso non c'è nulla da fare, tranne che cambiare PC. Per ulteriori informazioni su incompatibilità PAE-hardware si veda [WHDC2].

Le limitazioni di Windows

Le diverse versioni di Windows che supportano il PAE impongono comunque dei limiti allo spazio d'indirizzi fisico o alla RAM utilizzabile, come riportato nella seguente tabella (da [MSDN2], [WHDC1], [WHDC2]):

Versione	Limite di RAM (R) o di spazio d'indirizzi (I)
Windows 2000 Advanced Server	8 GiB (R)
Windows 2000 Datacenter Server	32 GiB (R)
Windows XP	4 GiB (I)
Windows Server 2003 Enterprise Edition	32 GiB (R)
Windows Server 2003 R2 (o SP1) Enterprise Edition	64 GiB (R)
Windows Server 2003 Datacenter Edition	64 GiB (R)
Windows Server 2003 Standard Edition	4 GiB (I)
Windows Vista	4 GiB (I)
Windows Server 2008 Enterprise or Datacenter Edition	64 GiB (R)
Windows Server 2008 (altre edizioni)	4 GiB (I)

Perché queste limitazioni? Le ragioni sono sia commerciali che tecniche; nel caso dei sistemi operativi server ovviamente Microsoft vuole incentivare il passaggio alle versioni più “prestigiose” (e costose) sulle macchine dotate di più RAM. Tuttavia in tutti i sistemi in cui la limitazione è ai 4 GiB di spazio d'indirizzi fisico (e non di RAM) questo limite ha anche un motivo tecnico precauzionale: molti driver di schede PCI attualmente in circolazione sono scritti senza tenere in considerazione la possibilità di avere a che fare con uno spazio d'indirizzi fisico di più di 32 bit, per cui, trovandosi di fronte ad indirizzi a 36 bit, è facile che diano problemi ([WHDC1], [WHDC2]).

Di conseguenza in questi casi il PAE, anche se supportato, risulta di fatto inutile per accedere alla RAM aggiuntiva; l'unica soluzione quindi è passare ad un sistema operativo a 64 bit, a meno che non si sia interessati a fornire la RAM oltre i 4 GiB solo ad applicazioni che utilizzano AWE (un sistema di API che consente alle applicazioni di gestirsi autonomamente la memoria fisica, si veda a questo proposito [MSDN4]).

Il PAE aumenta solo lo spazio di indirizzi fisico

Dopo aver abilitato il PAE e aver fatto riconoscere la memoria aggiuntiva al sistema operativo bisogna fare una considerazione aggiuntiva: molte applicazioni non pensate appositamente per sfruttare grandi quantità di memoria non ricaveranno alcun vantaggio in termini di quantità di dati apribili.

Ad esempio, attualmente iniziano a diffondersi adesso immagini di grandi dimensioni (oltre i 2 GB in memoria, per dare un'idea immagini più grandi di 26755*26755*24bpp), che molti visualizzatori di immagini si rifiutano di aprire per carenza di memoria. Qual è il problema in questo caso e perché il PAE non lo può risolvere? Il problema non è tanto la carenza di memoria fisica (a cui i moderni sistemi operativi possono sopperire tramite la memoria virtuale), ma piuttosto le dimensioni insufficienti dello spazio di indirizzi privato del processo. Molti visualizzatori di immagini per caricare un'immagine richiedono al sistema un blocco di memoria contiguo per ospitare in memoria tutta l'immagine non compressa per poi lavorarci agevolmente. Ora, lo spazio di indirizzi virtuale di ciascun processo, che sia attivo il PAE o meno, resta sempre di 2^{32} B = 4 GiB, ma di fatto nella maggior parte dei sistemi operativi lo spazio disponibile per l'applicazione si riduce a 2-3 GiB, poiché 1-2 GiB dello spazio di indirizzi di ogni processo sono riservati al sistema operativo e sono condivisi con tutti i processi. Di conseguenza il sistema operativo non può in ogni caso allocare per l'applicazione un blocco di memoria contiguo più largo di 2 o 3 GiB (a seconda delle dimensioni dello spazio dedicato al sistema operativo), e in generale non può allocare più di 2-

3 GiB di memoria mappata direttamente nello spazio di indirizzi privato del processo, per cui questo genere di errori non vengono risolti dal PAE.

In pratica, quindi, il PAE, consentendo l'accesso a più memoria fisica, *consente al sistema operativo e alle applicazioni appositamente pensate per lavorare con grandi quantità di memoria* di sfruttarla meglio, di ricorrere in maniera minore allo swapping e consente di allocare una quantità di memoria complessivamente (ossia, contando tutti i processi) maggiore.

Il mito dello switch /3GB di Windows

Si legge spesso in giro che lo switch /3GB dovrebbe avere qualche effetto sulla questione della RAM “sparita”, ma si tratta di sciocchezze. Lo switch /3GB serve a modificare la divisione applicazione-sistema dello spazio di indirizzi di ciascun processo, a cui ho accennato nel paragrafo precedente; senza lo switch /3GB la divisione è equa, 2 GiB per l'applicazione e 2 GiB per il sistema, mentre con lo switch in questione lo spazio disponibile per l'applicazione aumenta a 3 GiB ai danni dello spazio destinato al sistema, che si riduce ad 1 GiB.

Questa opzione risulta vantaggiosa esclusivamente per pochi applicativi, come alcuni DBMS, che sono in grado di gestire la memoria in maniera avanzata, e presenta numerosi svantaggi e limitazioni, come ad esempio:

- “compressione” dei dati del sistema operativo in uno spazio esiguo (1 GiB invece di 2 GiB), per cui minore cache di file e di rete e pool paged e nonpaged di dimensioni inferiori, il che in determinate condizioni può determinare un sostanziale peggioramento delle prestazioni;
- solo le applicazioni linkate con un particolare flag possono usufruire di tutti e 3 i GiB di spazio d'indirizzi virtuale, le altre sono comunque limitate ai primi 2 GiB per questioni di compatibilità (gli indirizzi fino ai 2 GiB hanno il bit più significativo impostato a 0, mentre quelli oltre ce l'hanno impostato a 1; diverse applicazioni hanno sfruttato il fatto che il sistema ha sempre restituito indirizzi sotto i 2 GiB per finalità varie, e l'uso da parte loro di memoria allocata oltre i 3 GiB può causare loro seri problemi, si veda a questo proposito in particolare [ONT1]);
- nonostante questo flag non è possibile allocare più memoria contigua rispetto a prima, visto che la zona intorno al limite dei 2 GiB è occupata da alcune dll di sistema che non possono essere rilocate, per cui lo spazio di indirizzi risulta comunque spezzato.

Tutto questo fermo restando che applicazioni scritte per gestire grandi quantità di dati possono comunque allocare più di 4 GiB di memoria tramite l'uso dei memory mapped file, con cui possono allocare tutta la memoria necessaria e mapparne in memoria di volta in volta solo la porzione su cui devono lavorare.

In conclusione, lo switch /3GB non ha nulla a che vedere con il problema della “memoria sparita”, è uno strumento da usare solo in casi estremamente rari e che in situazioni normali ha il solo effetto di penalizzare le performance del sistema.

Per ulteriori informazioni a proposito si vedano [ONT1] e [MSDN1].

Conclusione: c'è una soluzione migliore

L'ho già detto più volte, lo ripeto un'ultima volta: il PAE più che una vera e propria soluzione di fatto sembra essere un rattoppo. L'unica reale soluzione al problema attualmente risulta essere il passaggio a macchine e sistemi operativi a 64 bit. Con 64 bit è possibile indirizzare teoricamente

fino a 18446744073709551616 B (16777216 TiB = 16 EiB) nello spazio di indirizzi fisico come in quello virtuale, rimuovendo così ogni limite di RAM e di allocazione ragionevolmente raggiungibile in tempi brevi; in realtà attualmente i processori si limitano a gestire spazi di indirizzi di 48 bit ([INTEL1], [WIKI2]), ma il passaggio ai veri 64 bit basta che avvenga a livello di processore, il software applicativo non necessiterà di essere riscritto né ricompilato perché già da ora lavora con puntatori a 64 bit.

Anche le applicazioni a 32 bit possono beneficiare di un sistema operativo a 64 bit: infatti la parte di spazio di indirizzi virtuale di ogni processo dedicata al sistema nei sistemi a 64 bit si trova in prossimità della fine dello spazio di indirizzi a 64 bit, ossia in una parte non visibile al codice a 32 bit, che a questo punto è in grado di utilizzare l'intero spazio di indirizzi virtuale da 4 GiB ([MSDN2]). Sotto Windows tuttavia questo si verifica solo se l'applicazione è linkata con un particolare flag (LARGEADDRESSAWARE) per i motivi descritti in precedenza.

Riferimenti e letture aggiuntive

- GEEKIT1: Andrea Guida, Ubuntu Linux, come utilizzare oltre 3 GB di RAM (32 bit) (<http://www.geekissimo.com/2008/10/04/ubuntu-linux-come-utilizzare-oltre-3-gb-di-ram-32-bit/>)
- IEC1: IEC, Prefixes for binary multiples (http://www.iec.ch/zone/si/si_bytes.htm)
- IG1: Ian Griffiths, 3GB or 4GB - a 'Virtual' Memory Upgrade (<http://www.interact-sw.co.uk/iangblog/2005/08/05/is3gbenough>)
- INTEL1: Intel Corporation, Intel® 64 and IA-32 Architectures Software Developer's Manual, 2008 (<http://www.intel.com/products/processor/manuals/>)
- MSDN1: MSDN, 4-Gigabyte Tuning ([http://msdn.microsoft.com/en-us/library/bb613473\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb613473(VS.85).aspx))
- MSDN2: MSDN, Memory Limits for Windows Releases ([http://msdn.microsoft.com/en-us/library/aa366778\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa366778(VS.85).aspx))
- MSDN3: MSDN, Physical Address Extension (<http://msdn.microsoft.com/en-us/library/aa366796.aspx>)
- MSDN4: MSDN, Address Windowing Extensions ([http://msdn.microsoft.com/en-us/library/aa366527\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa366527(VS.85).aspx))
- MSKB1: Microsoft KB, In Windows Vista SP1 verranno segnalati 4 GB di memoria di sistema (RAM) in sistemi in cui sono installati 4 GB di memoria (<http://support.microsoft.com/kb/946003/>)
- ONT1: Raymond Chen, /3GB Articles Series (<http://blogs.msdn.com/oldnewthing/archive/2004/08/22/218527.aspx>)
- ONT2: Raymond Chen, Why can't I see all the 4 GB of RAM in my machine? (<http://blogs.msdn.com/oldnewthing/archive/2006/08/14/699521.aspx>)
- WHDC1: WHDC, Operating Systems and PAE support (http://www.microsoft.com/whdc/system/platform/server/PAE/pae_os.msp)
- WHDC2: WHDC, Physical Address Extension - PAE Memory and Windows (<http://www.microsoft.com/whdc/system/platform/server/PAE/PAEdrv.msp>)
- WIKI1: Wikipedia EN, Physical Address Extension (http://en.wikipedia.org/wiki/Physical_Address_Extension)
- WIKI2: Wikipedia EN, x86-64 (<http://en.wikipedia.org/wiki/x86-64>)