



Denne guide er oprindeligt udgivet på Eksperten.dk

CPU teknologi for programmører

Denne artikel forklarer nogle af de begreber som bruges omkring moderne CPU'er.

Beskrivelsen er teknisk og forudsætter en indsigt i software og OS som normalt kun fåes gennem low level C og assembler programmering.

Skrevet den **31. Jul 2009** af **arne_v** i kategorien **Hardware / PC** | ★★★★★

Historie:

V1.0 - 31/07/2009 - original

Indledning

Der er allerede skrevet meget om forskellige CPU'er og deres teknologi.

Men det mest af det som er skrevet henvender sig til ikke programmører. Hvilket begrænser muligheden for at forklare tingene.

32 versus 64 bit

Myte: en 64 bit computer opererer på 64 bit integers mens en 32 bit computer opererer på 32 bit integers.

En 32 bit computer kan sagtens operere på 64 bit integers. Og de gør det også i praksis. De fleste C compilere på 32 bit x86 computere understøtter 64 bit integers (typisk kendt som long long eller `_int64` i C).

At en computer er 64 bit betyder at adresser (assembler) eller pointerne (C) er 64 bit.

Myte: 64 bit computere er dobbelt så hurtige som 32 bit computere.

En 64 bit computer kan adressere mere memory, men det giver ingen hastigheds forskel (vi ser bort fra tilfælde hvor programmer skal bruge mere end 4 GB memory for at undgå disk IO).

Adresser/pointerne i et 64 bit program fylder mere, men det gør jo snarere programmet en anelse langsommere fordi der kan være færre adresser/pointerne i cache.

Hvis skiftet fra 32 til 64 bit også introducerer 64 bit integer registre, så vil 64 bit integer operationer normalt blive markant hurtigere, men der er ikke ret mange programmer som bruger 64 bit integers i stor stil.

Imidlertid giver skift fra x86/IA-32 til x86-64/x64/AMD64/EM64T faktisk en vis hastigheds forbedring. Men det har intet med 64 bit egenskaberne at gøre. Men i.s.f. at skifte fra 8 styk 32 bit registre (EAX, EBX, ECX, EDX, ESP, EBP, ESI, EDI) til 8 styk 64 bit registre, så har man lavet 16 styk 64 bit registre (R0-R15 hvor R0-R7 er mappet til de gamle 8). Det kan compilerne udnytte til at generere hurtigere kode.

Endvidere vil floating point operationer også blive hurtigere fordi man skifter fra 8 styk 80 bit stack baserede registre (gamle x87 arkitektur) til 16 styk 128 bit registre (XMM som oprindeligt blev introduceret med SSE).

Note: Itanium/IA-64 har 128 styk 64 bit integer registre og 128 styk 82 bit floating point registre.

virtual address space versus physical memory

Myte: man kan maksimalt have 4 GB RAM i en 32 bit computer.

Alle nyere mainstream styresystemer bruger virtuelt memory.

Virtuelt memory = den abstrakte memory som programmer tilgår
Virtuelt adresse rum = den mængde virtuelt memory som et program kan tilgå
Fysisk memory = RAM klodser
Fysisk adresse rum = den mængde fysisk memory som et system kan tilgå

Indholdet i en page i virtuel memory kan ligge i:

- fysisk memory
 - page/swap filer for modified data som der ikke er plads til i fysisk memory
 - EXE/DLL filer for readonly data som der ikke er plads til i fysisk memory (data genindlæses når der er behov for det)
- etc.

En 32 bit computer har et virtuelt adresse rum på 4 GB d.v.s. at et program ser 4 GB virtuelt memory.

Hvis vi tager et meget udbredt styresystem som 32 bit Windows (XP/2003/Vista/2008), så kan de 4 GB deles op i 2 GB process address space og 2 GB system address space. Hver process har sit eget process space mens system space er shared mellem alle processer. D.v.s. at hvis der kører N processer på maskinen så er der potentiale for brug af $N*2+2$ GB RAM.

Note: Den kendte /3GB switch til 32 bit Windows ændrer fordelingen mellem process og system space fra 2-2 til 3-1.

x86/IA-32 understøtter 32 bit fysiske adresser uden PAE og 36 bit fysiske adresser med PAE. Det betyder at der understøttes op til 4 GB RAM uden PAE og 64 GB RAM med PAE.

Idag er PAE næsten altid aktiveret, da brug af no execute bitten (som forhindrer eksekvering af kode i data område, hvilket beskytter mod mange buffer overrun exploits) kræver at PAE aktiveres.

32 bit Windows i desktop udgaver (XP/Vista) understøtter dog kun de 4 GB RAM selv med PAE aktiveret. Det er en restriktion lagt ind i styre systemet. Den

officielle forklaring er at ikke alle 3. parts drivere kan finde ud af 36 bit fysiske adresser.

32 bit Windows i server udgaver (2003/2008) og 32 bit Linux etc. understøtter mere end 4 GB RAM.

Note: de mere end 4 GB RAM kan så udnyttes enten via flere processer eller via brug af en windowing teknik i en enkelt process (Windows har et API kaldet AWE til formålet).

x86-64/x64/AMD64/EM64T har ikke de restriktioner. Nuværende processorer bruger 48 bit virtuelle adresser og 40 bit fysiske adresser og understøtter dermed 256 TB virtuel memory og 1 TB fysisk memory. Fremtidige procesorer kan bruge 64 bit virtuelle adresser og 52 bit fysiske adresser og kan dermed understøtte 16 EB virtuel memory og 4 PB fysisk memory.

Note: Itanium/IA-64 bruger 64 bit virtuelle adresser og 50 bit fysiske adresser og understøtter dermed 16 EB virtual memory og 1 PB fysisk memory.

Den forsvundne RAM

Myte: Windows XP (32 bit) understøtter kun 3.2 GB RAM (eller 2.8 eller 3.5 eller noget andet).

Som beskrevet ovenfor understøtter Windows XP 32 bit 4 GB fysisk adresse rum. Og det betyder at der understøttes 4 GB RAM.

Men der er en lille finesse.

IO devices som tilgås direkte via memory adresser bruger nemlig også fysiske adresser.

PCI of PCIe (PCI Express) allokerer fysiske memory adresser i klumper af 256 MB.

D.v.s. at de fysiske memory adresser der er til rådighed for RAM er 4 GB minus et multipla af 256 MB som bruges af PCI og PCIe (og yderligere minus nogle forskellige småting i KB klassen).

MP versus SMP versus ASMP

Myte: en computer med flere CPU hedder en SMP computer.

En MP (Multi Processor) computer har flere CPU'er.

Et SMP (Symmetric Multi Processor) styresystem gør at alle CPU'er er lige og kan udføre alle operationer.

Et ASMP (Assymmetric Multi Processor) styresystem gør at alle CPU'er ikke er lige. Typisk er der en master CPU og et antal slave CPU'er, hvor kun master CPU kan lave IO etc. mens slave CPU'erne kun kan lave beregninger.

Det er åbenlyst at ASMP systemer skalerer betydeligt dårligere end SMP systemer til generelt brug. Og ASMP blev da også opgivet for mere end 15 år siden til generelle systemer.

Windows, Linux og diverse Unix er alle SMP styresystemer.

ASMP er stadig relevant for mere specialiserede formål.

socket versus core versus thread

I gamle dage var det nemt at angive antal CPU'er. Et system havde 1/2/4/8/16//32/whatever CPU'er.

Idag er begrebet 1 CPU blevet lidt diffust.

Og man bør derfor tale om CPU sockets, CPU cores og CPU threads.

CPU sockets er antallet af fysiske CPU'er som bogstaveligt er sat ned i en socket.

CPU cores er antallet af fysiske enheder som kan lave beregninger. Idag er det helt almindeligt med 2-6 cores per socket. Og man forventer at antallet vil stige yderligere i de kommende år.

CPU threads er:

- antallet af logiske enheder set af styresystemet som kan lave beregninger
- antallet af states (register sets)

Ofte er der kun 1 thread per core, men Intel har x86 og x86-64 processorer med 2 threads per core (kaldet hyperthreading) og IBM Power og SUN SPARC processorer fåes også med flere threads per core.

Selvom der kun er en tråd per core som faktisk eksekverer, så er der alligevel store fordele ved SMT (Simultaneous MultiThreading) som flere threads per core hedder.

De to store fordele er:

- * eksekverings enheden kan arbejde på andre tråde mens en tråd venter på at hente data fra memory
- * man undgår en del context swiches hvor aktiv thread skal udskiftes

Intel hævdede at den oprindelige Hyperthreading i Pentium 4 kunne give op til 30% mere througput for en multithreaded app. Og det er korrekt - det er muligt at se op til 30% forbedring. Men det er selvfølgelig langt fra altid at man ser så store forbedringer - gennemsnittet er langt lavere - og der er sågar set tilfælde på forværret performance (ekstreme tilfælde hvor de to tråde ødelægger hinandens cache).

Kommentar af Etnogeek d. 20. Dec 2010 | 1

Rigtig nice artikel. Keep at it