

# Software pro vícejádrové CPU

CPU s více výpočetními jádry jsou na vzestupu, dosud však málokterý program dokáže využít jejich skutečnou výkonnost. Kdy se více jader uplatní? *Thomas Littschwager, autor@chip.cz*

■ Zní to velice pěkně: „Čtyřnásobný výpočetní výkon za pouhých 6500 Kč“, jak jej nabízí nejlevnější momentálně dostupný čtyřjádrový procesor Core 2 Quad Q6600 od Intelu. Teoreticky to souhlasí: pokud operační systém a software optimálně spolupracují, tato CPU hravě porazí i ty nejdražší dvoujádrové procesory za 23 000 Kč.

Vzdor lákavé ceně však mnozí zákazníci dosud váhají. Stále se v nich totiž zatvrzele drží povědomí, že se čtyřmi jádry se dnes ještě nedá nic dělat. A tento předpoklad není úplně falešný: z více výpočetních jader dokáže skutečně profitovat jen málo současných programů. Existují dokonce nástroje, které v prostředí s více jádry běží ještě pomaleji než na klasickém jednojádrovém PC! Jinými slovy: za zvláště nepříznivých okolností může náročnost správy přídatných jader převýšit jejich přínos.

Ale proč je vlastně tak obtížné vytěžit z nových procesorů jejich plný výkonnostní potenciál? Vždyť nový hardwarový design ze strany výrobců čipů vůbec není tím hlavním problémem. Téměř ze všech programů by dnes vývojáři softwaru dokázali dostat podstatně větší výkon – kdyby už předem počítali s nasazením vícejádrových CPU. Na dalších řádcích se dočtete, v čem spočívají hlavní potíže vývoje odpovídajících programů; navíc jsme také změřili, jak velký nárůst rychlosti může přinést software

optimalizovaný až pro osm procesorových jader.

## Multithreading: Náročné umění dělby práce

Aby program měl vůbec šanci běžet na více jádrech, musí být rozdělen na jednotlivá výpočetní „vlákna“ – v angličtině zvaná „threads“. V programu používajícím tento tzv. „multithreading“ pak výpočetně

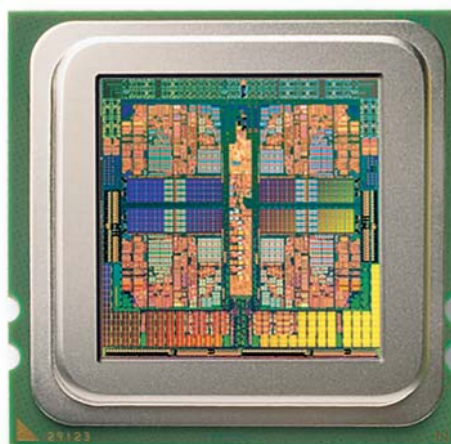
intenzivní část může během volných časů CPU běžet na pozadí; mezitím například malé programové vlákno průběžně aktualizuje grafické ovládací prostředí. To samo o sobě však stále ještě zrychlení téměř nepřináší, neboť disponibilní doba CPU byla sice ideálně rozdělena mezi jednotlivé operace – ty se však stále provádějí jedna po druhé.

Již procesory Pentium 4 pracovaly díky „hyperthreadingu“ se dvěma logickými jádry. Ta mohou v tomtéž čase provádět dvě rozdílné výpočetní operace. Jenomže s tím také začaly problémy: běželi-li současně dva procesy, které mají pracovat s tímtež systémovým prostředkem, například paměťovou buňkou, může se samozřejmě stát, že jedno z vláken je s výpočtem hotovo rychleji než druhé. A to přesto, že třeba ještě není dopočítán důležitý výsledek druhého vlákna, přičemž to už nemá přístup k danému prostředku. Tento stav je znám také pod názvem „race-condition“ (viz rámeček na 33).

Existuje i další problém: thready se mohou navzájem zablokovat, jestliže současně chtějí použít nějaký prostředek a zároveň blokují jiný. Jednotlivá vlákna si tak vzájemně stojí v cestě – mluvíme také o „smrtném objetí“ („deadlock“, viz 33). Zatímco však při hyperthreadingu u dvoujádrových procesorů byla zmíněná problematika omezena na dvě vlákna, se stoupajícím počtem jader se značně komplikuje – a softwaroví vývojáři musí úplně změnit způsob myšlení.

## Rozdíly: Kde může multicore zabodovat – a kde ne

Efektivní paralelizaci výpočtů dosti často znemožní jejich vzájemná závislost. Rozdíly objasní několik jednoduchých příkladů: bez jakéhokoliv zrychlení se



**QUADCORE**  
AMD a Intel (na obrázku Core 2 Quad Q6600) dnes nabízejí CPU se čtyřmi výpočetními jádry - ta však přinášejí zvýšení výkonu jen v případě, že software co nejvíce jader zatěžuje současně.

## Benchmarky: Kde vícejádrové procesory využijí svůj potenciál

### Komprimační program: WinRAR 3.71

1 jádro	100 %
2 jádra	105 %
4 jádra	105 %
8 jader	105 %

**BEZ ÚSPĚCHU:** WinRAR si s více jádry nedokáže nic počít. Komprimace souborů se prakticky nezrychlí.

### 3D hra: World in Conflict

1 jádro	100 %
2 jádra	180 %
4 jádra	230 %
8 jader	230 %

**ZRYCHLENÍ:** Současné 3D hry jako „World in Conflict“ profitují zpravidla „jenom“ ze dvou či čtyř jader CPU.

### Videokodek: MainConcept H.264 Pro

1 jádro	100 %
2 jádra	180 %
4 jádra	280 %
8 jader	440 %

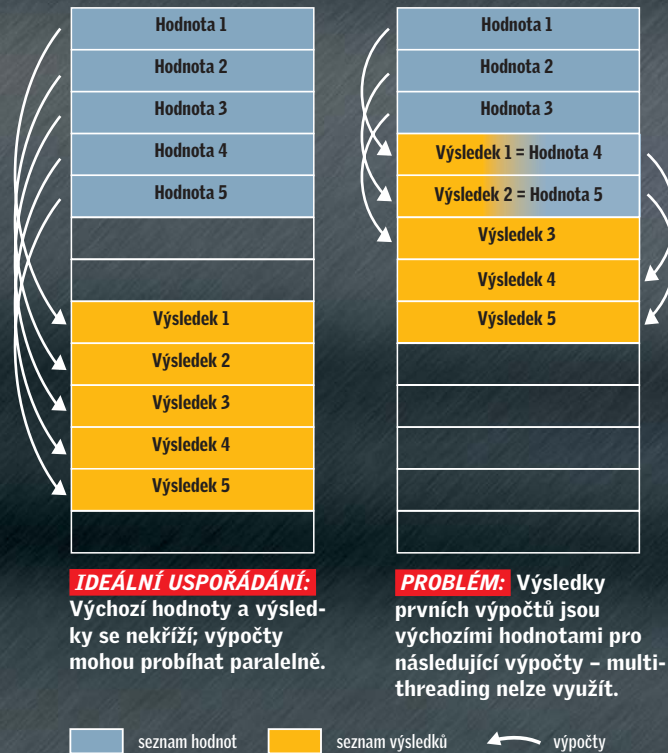
**VÝBORNÝ VÝSLEDEK:** Rozdělení kódování videa na vlákna se u kodeku H.264 Pro od firmy MainConcept projevilo velmi efektivně.

obrázek: „World in Conflict“

# Dilema multithreadingu: Paralelně, nebo ne?

Nástroje jako **Compiler 10** od Intelu, které mají optimalizovat programový kód pro více jader, někdy nerozpoznají úmysly programátora - a vícejádrový efekt je pryč.

Předpokládejme, že jednoduchý program čte jedno číslo po druhém ze seznamu hodnot (šedě), přičítá k němu jedničku a poté součet zapisuje do seznamu výsledků (žlutě). Při optimalizaci ovšem kompilátor neví, zda seznamy hodnot a výsledků jsou v dalším průběhu programu uspořádány pro multithreading vhodně (vlevo), nebo nevhodně (vpravo) - přičemž vlastní kód může vypadat identicky. Aby vyloučil možnost chyby, vyrobí kompilátor pro jistotu jen jednovláknový program - přídavná jádra zůstanou nevyužita.



musí obejít především takový software, u něhož každý jednotlivý výpočet potřebuje výsledky předchozích výpočtů. Klasickým příkladem z informatiky je zde „množení králíků“ neboli výpočet tzv. Fibonacciho posloupnosti, kde jsou pro výpočet dalšího členu zapotřebí hodnoty dvou jeho předchozích členů. Ale i některé běžné aplikace ponechávají další výpočetní jádra bez povšimnutí: například aktuální verze 3.71 komprimačního programu WinRAR prostě využívá jen jedno jádro (viz graf na 32).

Také vývojáři počítačových her se zatím ještě drží zpět, ačkoliv se hry pro paralelizaci vlastně velmi dobře hodí - ostatně grafické čipy pracují paralelně „odjakživa“. Přesto dosud jen málo současných her, například „World in Conflict“, rozkládají umělou inteligenci herních figur a fyzikální výpočty na více jader. Zde pak uživatel přinejmenším při použití až čtyř jader už jistý výkonnostní zisk pocítí.

Paralelizace je však velmi zajímavá například pro kódo-

vání videa, neboť zde může podle okolností přinést impozující nárůst rychlosti. Vytváření vláken tady má smysl buď mezi dvěma jako statický obraz uloženými scénami (I-frames), nebo při interpolaci pohybových vektorů v obrazu. Například kodek H.264 Pro firmy MainConcept škáluje od jednoho až do osmi jader velmi dobře - při rychlejším pevném disku by byly dosažitelné i výsledky ještě lepší.

## Pohodlně, nebo perfektně: Intel Compiler 10 proti OpenMP

Jako toho času přední dodavatel vícejádrových procesorů dává Intel programátorům do rukou použitelný prostředek: aktuální Intel Compiler 10 (30denní testovací verze ke stažení na [www.intel.com/software](http://www.intel.com/software)) analyzuje kód programu během kompilačního procesu a - pokud je to možné - automaticky jej rozloží do více vláken. Intel Compiler 10 ovšem není žádný zázračný prostředek: samičinné rozpoznávání funguje

jen tehdy, je-li si kompilátor zcela jist, že nemůže nastat žádný deadlock nebo race-condition. Při tomto rozhodování si však počíná silně konzervativně - a proto často produkuje pouze programy pracující postupně (viz rámeček nahoře).

Skutečnou optimalizaci tedy dokáže zajistit jedině sám pro-

gramátor, který přesně ví, které partie jeho softwaru mohou běžet nezávisle. Od roku 1997 je k dispozici programovací prostředek OpenMP, u něhož může programátor v kódu přímo specifikovat, zda lze daný prvek paralelizovat, nebo ne. Počet vláken přitom může buď definovat sám, nebo jeho volbu přenechat systému. OpenMP spolupracuje s programovacími jazyky C/C++ a Fortran a představuje tak ideální základnu pro programování vícejádrových aplikací.

## Výhled: Vícejádrový software je téma budoucnosti

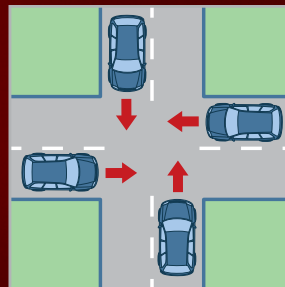
U AMD jsou jednojádrové procesory na vyměření, Intel se již začátkem roku rozhodl žádné další jednojádrové modely nevyrobět - a už bájí o CPU s 80 i více jádry. Bez podpory multicore se tedy v blízké budoucnosti žádný software neobejde - a programátoři udělají jen dobře, když se už nyní soustředí na své „domácí úlohy“ z multithreadingu.

### INFO:

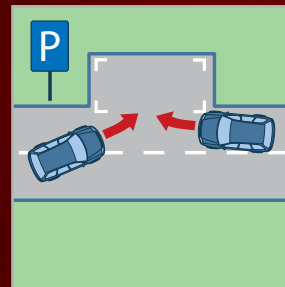
**whatif.intel.com:** Webová stránka pro vývojáře vícejádrového softwaru

**www.openmp.org:** Download a dokumentace OpenMP

## Nebezpečí paralelizace



**DEADLOCKS** Pokud několik vláken čeká na prostředek, který některé jiné vlákno blokuje, všechno se zastaví. Vznikne situace podobná známému problému čtyř vozidel přijíždějících ke křižovatce s předností zprava.



**RACE-CONDITION** Pokud si dvě vlákna nárokují tentýž prostředek, zvítězí to rychlejší - druhé vlákno vyjde naprázdno a musí si hledat jiné „parkoviště“.