

Obsah

1 Úvod do problematiky	3
1.1 Úvod	3
1.1.1 Základní definice	3
1.1.2 Definice	4
1.2 Motivační příklad	4
1.2.1 Triviální algoritmus	5
1.2.2 Strassenův algoritmus	5
1.2.3 Coppersmith-Winogradův algoritmus	5
1.2.4 Srovnání algoritmů	6
1.3 Fourier-Motzkinova eliminace	6
1.3.1 Algoritmus FME	8
1.4 Krátký úvod do hlavních součástí architektury počítačů	13
1.4.1 CPU	13
1.4.2 Hlavní paměť	15
1.4.3 Virtuální paměť	15
1.4.4 Skrytá paměť (<i>cache</i>)	16
1.4.5 Sekundární paměť a sběrnice	17
1.5 Matematický koprocesor x87	17
1.5.1 Důvod vzniku x87	17
1.5.2 Vnitřní architektura x87	17
1.5.3 Reprezentace čísel v x87	18
1.5.4 Časování FPU instrukcí	18
1.5.5 Instrukce x87 (příklady)	19
1.5.6 Nevýhody výpočtů na FPU	19
1.6 Vektorizace výpočtů na procesorech x86	20
1.6.1 Technologie MMX	20
1.6.2 Technologie 3DNow!	22
1.6.3 Technologie SSE	23
1.6.4 Instrukční sada SSE3 a další	27
1.6.5 AVX	28
1.6.6 Ukázky použití vektorové sady SSE	28
1.6.7 Zjištění procesorem podporovaných vektorových sad	30
1.6.8 Zhodnocení přínosu vektorových sad	32
1.7 Postup optimalizace	33
1.7.1 Pokračování motivačního příkladu	33
1.7.2 Postup při optimalizaci	35

2	Transformace zdrojových kódů	37
2.1	Datové závislosti	37
2.1.1	Legalita transformace cyklů	44
2.2	Modely chování skryté (cache) paměti	45
2.2.1	Paměťový subsystém x86	45
2.2.2	Typy výpadků ve skryté paměti	47
2.2.3	Vztah úrovní skrytých pamětí	47
2.2.4	Modely pro chování skryté paměti	48
2.2.5	Model využívající přístupový interval	49
2.2.6	Zobecněný model využívající přístupový interval (GRD) I	51
2.2.7	Pravděpodobnostní model	52
2.3	Optimalizační techniky	53
2.3.1	Obecné optimalizace	54
2.3.2	Optimalizace zaměřené na cykly	55
3	Paralelní zpracování	59
3.1	Paralelní systémy a mechanismy	59
3.1.1	Flynnova taxonomie paralelních architektur	60
3.1.2	Metody programování paralelních systémů	62
3.1.3	Vykonávání paralelních úloh	62
3.1.4	Hodnocení kvality paralelních programů	65
3.1.5	Vícevláknové programování	66
3.2	OpenMP	67
3.2.1	Základy OpenMP	67
3.2.2	Paralelizace cyklů	69
3.2.3	Další rysy OpenMP	75
3.2.4	Základní OpenMP operace	77
3.2.5	Operace flush	77
3.2.6	Operace s OpenMP zámky	78
3.2.7	Kritické sekce	79
3.2.8	Funkční paralelismus v OpenMP	79
3.2.9	Efektivita OpenMP kódů	84
3.2.10	Proměnné prostředí	86
4	Případové studie optimalizace různých kódů a praktické rady	89
4.1	Případové studie optimalizace algoritmů	89
4.1.1	Výpočet histogramu	89
4.1.2	Výpočet diferenciálního operátoru	90
4.1.3	Gaussova eliminace	91
4.1.4	Násobení řídké matice vektorem	92
4.2	Praktické rady	96
4.2.1	Použití vektorizace v C/C++	96
4.2.2	Vložená část v JSA	96
4.2.3	Použití MMX, SSE, AVX intrinsic funkcí	97
4.2.4	Automatická podpora vektorizace	97

A	Nastavení kompilátoru GCC	99
A.1	Automatická vektorizace v GCC	99
A.2	Třífázová optimalizace	99
A.3	Generování kódu	99
A.3.1	Použití FPU	99
A.3.2	Nastavení cílové architektury	100
A.3.3	Předání parametrů	101

1.1	Tabulka adresy řádek nastavy libeem eliminace pro zkrácení případ	11
1.2	Význam jednotliček 32bit registrů eax a edx po zavření instrukce opud a nastavení režimu 32bit	12
1.3	Význam jednotliček 32bit registrů eax a edx po zavření instrukce opud a nastavení režimu 32bit. Jedná se o rozdílné typy pro AMD procesory.	12
2.1	Počet řádků, které odpovídá výpis k bloku v dané úrovni slyše použití v procesoru Intel	19