

**Microsoft**



Ján Hanák

# **Praktické paralelné programovanie v jazykoch C# 4.0 a C++**



Ján Hanák

**Praktické paralelné  
programovanie  
v jazykoch C# 4.0 a C++**

Artax  
2009

Autor: Ing. Ján Hanák, MVP

## **Praktické paralelné programovanie v jazykoch C# 4.0 a C++**

Recenzenti:	doc. RNDr. Jozef Fecenko, CSc. Ing. Magdaléna Cárachová, PhD.
Vydanie:	prvé
Rok prvého vydania:	2009
Náklad:	500 ks
Jazyková korektúra:	Ing. Peter Kubica
Vydal:	Artax a.s., Žabovřeská 16, 616 00 Brno pre Microsoft s.r.o., Vyskočilova 1461/2a, 140 00 Praha 4
Tlač:	Artax a.s., Žabovřeská 16, 616 00 Brno
ISBN:	978-80-87017-06-7



## Obsah

Úvod .....	3
Pod'akovanie .....	7
Venovanie.....	7
1 Charakteristika praktických ukážok paralelného programovania .....	8
1.1 Konfigurácie počítačových systémov určených na empirické testovanie sekvenčných a paralelných programov .....	12
2 Praktická ukážka č. 1: Matematické operácie s 3D vektormi .....	13
2.1 Empirické testovanie sekvenčného a paralelného programu a kvantifikácia nárastu výkonnosti.....	21
3 Praktická ukážka č. 2: Riešenie masívnych súprav sústav 3 lineárnych rovníc s 3 neznámymi .....	25
3.1 Empirické testovanie sekvenčného a paralelného programu a kvantifikácia nárastu výkonnosti.....	32
4 Praktická ukážka č. 3: Lineárna algebra – Násobenie matíc typu 300x300.....	36
4.1 Empirické testovanie sekvenčného a paralelného programu a kvantifikácia nárastu výkonnosti.....	44
5 Praktická ukážka č. 4: Paralelné grafické transformácie bitových máp (paralelizmus s vysokou úrovňou abstrakcie).....	48
5.1 Empirické testovanie sekvenčného a paralelného programu a kvantifikácia nárastu výkonnosti.....	54
6 Praktická ukážka č. 5: Paralelné grafické transformácie bitových máp (paralelizmus s nízkou úrovňou abstrakcie) .....	58
6.1 Empirické testovanie paralelného programu a kvantifikácia nárastu výkonnosti .....	63

6.2 Diagnostika a monitorovanie výkonu paralelného programu profilovacím programom integrovaným v prostredí produktu Visual Studio 2010 .....	68
7 Praktická ukážka č. 6: Vyhľadávanie prvočísel (riadený a natívny paralelizmus)..	75
7.1 Empirické testovanie riadeného sekvenčného programu, natívneho sekvenčného programu, riadeného paralelného programu a natívneho paralelného programu s kvantifikáciou nárastu výkonnosti .....	82
8 Praktická ukážka č. 7: Numerická integrácia - natívny paralelizmus pomocou rozhrania OpenMP .....	91
8.1 Empirické testovanie sekvenčného a paralelného natívneho programu a kvantifikácia nárastu výkonnosti.....	94
8.2 Diagnostika a monitorovanie výkonu paralelného natívneho programu profilovacím programom Intel Parallel Amplifier integrovaným v prostredí produktu Intel Parallel Studio .....	98
9 Praktická ukážka č. 8: Implicitný natívny paralelizmus - automatická paralelizácia algoritmu numerickej integrácie pomocou prekladača Intel C++ Compiler 11.1 .....	102
10 Finálny sumár kvantitatívnych charakteristík praktických ukážok .....	109
11 Metodika riešiaci vývoj paralelných programov optimalizovaných pre beh na počítačových systémoch s viacjadrovými procesormi .....	111
11.1 Metodika riešiaci paralelizáciu pôvodne sekvenčného programu .....	112
11.2 Metodika riešiaci vývoj úplne nového paralelného programu .....	124
Záver .....	128
O autorovi.....	129

# Úvod

Vážené čitateľky, vážení čitatelia,

po mimoriadne pozitívnej akceptácii vysokoškolskej učebnice „Základy paralelného programovania v jazyku C# 3.0“<sup>1</sup> sme usúdili, že tematika paralelného a paralelného objektovo orientovaného programovania (POOP) je poslucháčmi informatických vysokých škôl univerzitného typu, akademickými pracovníkmi, komerčnými vývojármi a fanúšikmi programovania ponímaná ako veľmi atraktívna a žiadaná. Keďže chceme v osvete paradigmy POOP prostredníctvom najmodernejších softvérových technológií pokračovať i naďalej, prichádzame s novou knihou, ktorá sa sústreďuje na praktické paralelné programovanie. Cieľom predkladanej publikácie je predstaviť základy POOP na praktických ukázkach, ktoré boli vyvinuté v programovacích jazykoch C# 4.0 a C++ (podľa štandardu C++0x), s využitím vývojových prostredí Microsoft Visual Studio 2008, Microsoft Visual Studio 2010 a Intel Parallel Studio.

Vzhľadom na to, že kniha „Praktické paralelné programovanie v jazykoch C# 4.0 a C++“ priamo nadväzuje na dielo „Základy paralelného programovania v jazyku C# 3.0“, predpokladáme, že čitatelia ovládajú elementárne princípy, postuláty a techniky paralelného programovania v rozsahu, v akom sme ich vysvetlili v predchádzajúcej vysokoškolskej učebnici. K ďalším prerekvizitám, ktoré u čitateľov očakávame, patrí solídne zvládnutie programovacích jazykov C# a C++ na pokročilej úrovni.

Na najvyššej úrovni abstrakcie môžeme všetky praktické ukážky tejto knihy rozdeliť do 2 samostatných skupín:

- 1. Praktické ukážky demonštrujúce implementáciu explicitného riadeného paralelizmu.** Tieto praktické ukážky sme vyvinuli

---

<sup>1</sup> Hanák, J.: Základy paralelného programovania v jazyku C# 3.0. Brno: Artax, 2009. Vysokoškolská učebnica je v elektronickej podobe zdarma k dispozícii na nasledujúcej adrese: <http://msdn.microsoft.com/cs-cz/dd727769.aspx>.

v programovacom jazyku C# 4.0 s podporou knižnice Task Parallel Library (TPL), ktorá je súčasťou bázeovej knižnice tried (BCL) vývojovo-exekučnej platformy Microsoft .NET Framework 4.0. Ako hlavné vývojové prostredie sme použili Microsoft Visual Studio 2010.

2. **Praktické ukážky demonštrujúce implementáciu explicitného natívneho paralelizmu.** Tieto praktické ukážky sme vytvorili v programovacom jazyku C++ s podporou nových syntakticko-sémantických inovácií, ktoré zavádza pripravovaný ISO štandard so zatiaľ pracovným označením C++0x. Okrem najaktuálnejšej verzie jazyka C++ s výhodou uplatňujeme programové konštrukcie a entity z knižníc Microsoft Parallel Patterns Library (PPL) a OpenMP. Citeľný nárast pracovnej produktivity pri tvorbe natívnych paralelných programov sme zaznamenali pri použití kombinácie produktov Microsoft Visual Studio 2008 a Intel Parallel Studio. Produkt Intel Parallel Studio je balíkom nástrojov, ktoré umožňujú efektívne absolvovať všetky náročné štádiá návrhu, tvorby a ladenia paralelných natívnych aplikácií napísaných v jazykoch C a C++.



---

**Poznámka:** Keďže v čase tvorby tohto diela sa produkt Intel Parallel Studio integroval do prostredia produktu Microsoft Visual Studio 2008, použili sme v tých častiach knihy, ktoré sa produktu Intel Parallel Studio venujú, spomínané staršie vývojové prostredie spoločnosti Microsoft. (Ako však býva u spoločnosti Intel dobrým zvykom, verzia nástroja Parallel Studio pre Visual Studio 2010 sa objaví zakrátko po uvedení finálnej verzie vývojového prostredia od firmy Microsoft.)

---

V knihe sa nachádzajú nasledujúce praktické ukážky:

1. **Matematické operácie s 3D vektormi.**  
Technológie: C# 4.0 a TPL.
2. **Riešenie masívnych súprav sústav 3 lineárnych rovníc s 3 neznámymi.**  
Technológie: C# 4.0 a TPL.

3. **Lineárna algebra – Násobenie matíc typu 300x300.**  
Technológie: C# 4.0 a TPL.
4. **Paralelné grafické transformácie bitových máp (paralelizmus s vysokou úrovňou abstrakcie).**  
Technológie: C# 4.0 a TPL.
5. **Paralelné grafické transformácie bitových máp (paralelizmus s nízkou úrovňou abstrakcie).**  
Technológie: C# 4.0 a BCL.
6. **Vyhľadávanie prvočísel (riadený a natívny paralelizmus).**  
Technológie: C# 4.0 a TPL, C++0x a PPL.
7. **Numerická integrácia – explicitný natívny paralelizmus.**  
Technológie: C++0x a OpenMP.
8. **Numerická integrácia – implicitný natívny paralelizmus.**  
Technológie: C++0x, automatický paralelizér prekladača Intel C++ 11.1.

Každá praktická ukážka je vybavená informačným panelom, ktorý podáva nasledujúce informácie:

- Cieľ praktickej ukážky.
- Vedomostná náročnosť praktickej ukážky.<sup>2</sup>
- Časová náročnosť praktickej ukážky<sup>3</sup>.
- Softvérové technológie, ktoré boli použité pri vytváraní praktickej ukážky.
- Druh paralelizmu, ktorý praktická ukážka implementuje.

---

<sup>2</sup> Úroveň vedomostnej náročnosti je determinovaná relatívne s ohľadom na cieľové publikum knihy.

<sup>3</sup> Úroveň časovej náročnosti je determinovaná relatívne s ohľadom na cieľové publikum knihy.

Pri tvorbe praktických ukážok sme uplatnili metodiku, ktorú detailne predstavujeme v kapitole *1 Charakteristika praktických ukážok paralelného programovania*. Sekvenčné a paralelné programy, ktoré v jednotlivých praktických ukážkach konštruujeme, sme podrobili výkonnostným testom na variabilne dimenzovaných počítačových systémoch (ich bližší opis uvádzame v kapitole *1.1 Konfigurácie počítačových systémov určených na empirické testovanie sekvenčných a paralelných programov*). Našou intenciou bolo vždy kvantifikovať nárast výkonnosti paralelných programov, ktoré vznikli paralelizáciou pôvodne sekvenčných programov.

Najväčšou konkurenčnou výhodou tejto knihy je nová metodika, ktorá rieši vývoj paralelných programov optimalizovaných pre beh na počítačových systémoch s viacjadrovými procesormi. S využitím tejto metodiky môžu softvéroví vývojári maximalizovať svoju produktivitu pri paralelizácii existujúcich sekvenčných programov, či pri vývoji úplne nových paralelných programov.

Ján Hanák

Bratislava, november 2009

## Pod'akovanie

Ako autor tejto knihy by som chcel vyjadriť svoje pod'akovanie recenzentom, doc. RNDr. Jozefovi Fecenkovi, CSc., a Ing. Magdaléne Cárachovej, PhD., za dôkladné posúdenie tohto diela a hodnotné námety na jeho ďalšie skvalitnenie.

Veľká vďaka patrí rovnako Mgr. Miroslavovi Kubovčíkovi zo spoločnosti Microsoft Slovakia za výbornú podporu, ústretový prístup a skvelú niekoľkoročnú spoluprácu, výsledkom ktorej je aj táto kniha (a ako obaja veríme, aj mnoho ďalších diel, ktoré pre vývojárov pripravíme v nasledujúcich rokoch).

Moje pod'akovanie a úcta patrí aj Ing. Petrovi Ulvrovi zo spoločnosti Intel za výborné softvérové zabezpečenie našej spolupráce.

V neposlednom rade by som sa rád pod'akoval aj Ing. Martinovi Lukáškovi a Ing. Vítovi Obůrkovi zo spoločnosti Artax za veľkolepú a vždy hladkú spoluprácu pri príprave grafických materiálov, zlome a tlači tejto knihy.

## Venovanie

*Túto knihu venujem všetkým nadaným ľuďom,  
ktorí každodenne vynakladajú svoju energiu, úsilie a čas na to,  
aby sa z tohto sveta stalo lepšie miesto pre život.*

## 1 Charakteristika praktických ukážok paralelného programovania

Táto kniha je venovaná praktickým ukážkam paralelného programovania. Predstavené praktické ukážky majú nasledujúce charakteristiky:

1. Riešime parciálne problémy, respektíve inštancie parciálnych problémov.
2. Konštruujeme sekvenčné a paralelné algoritmy, ktoré riešia parciálne problémy. Pritom sa sústreďujeme na analýzu nárastu výkonnosti pôvodne sekvenčných programov po ich paralelizácii.

Celý proces prebieha v týchto krokoch:

- Najskôr zostavíme sekvenčný algoritmus, ktorý rieši parciálny problém.
- Sekvenčný algoritmus implementujeme do sekvenčného programu.
- Empirickými testami zisťujeme výkonnostné metriky sekvenčného programu. Empirické testy sekvenčného programu sú realizované na počítačoch s viacjadrovými procesormi<sup>4</sup>. Testy skúmajú dĺžku exekučného času, ktorý je nutný na spracovanie sekvenčného programu. Každý test sa skladá z piatich relácií, v rámci ktorých zaznamenáme dĺžku exekučných časov spracovania sekvenčného programu. Potom vypočítame priemerný exekučný čas spracovania sekvenčného programu.

---

<sup>4</sup> Presné technické konfigurácie testovacích počítačových systémov sú uvedené v časti 1.1 *Konfigurácie počítačových systémov určených na empirické testovanie sekvenčných a paralelných programov.*

- Vytvorený sekvenčný program podrobíme transformácii, ktorej výsledkom je ekvivalentný paralelný program. Skonstruovaný paralelný program vznikne paralelizáciou pôvodne sekvenčného algoritmu, ktorý implementuje sekvenčný program.
- Empirickými testami skúmame výkonnostné metriky paralelného programu. Empirické testy paralelného programu sú realizované na rovnakých počítačoch, na ktorých boli uskutočnené empirické testy sekvenčného programu. Podobne ako pri testoch sekvenčného programu, tak aj pri testovaní paralelného programu analyzujeme priemerný exekučný čas jeho spracovania (ktorý plynie z 5 relácií).
- Porovnávame výkonnostné charakteristiky sekvenčného a paralelného programu a determinujeme nárast výkonnosti paralelného programu vo vzťahu k sekvenčnému programu. Pritom používame nasledujúci matematický vzťah:

$$N_V = \frac{T_S}{T_P}$$

kde:

- ❖  $N_V$  je nárast výkonnosti paralelného programu voči sekvenčnému programu.
- ❖  $T_S$  je priemerný exekučný čas spracovania sekvenčného programu.
- ❖  $T_P$  je priemerný exekučný čas spracovania paralelného programu.

Po určení exaktnej hodnoty nárastu výkonnosti paralelného programu prijímame záver o trende jeho výkonnosti, ktorý môže byť:

- ❖ Sublineárny, ak  $N_V < n$ .
- ❖ Lineárny, ak  $N_V = n$ .
- ❖ Superlineárny, ak  $N_V > n$ .

kde:

❖  $n$  je počet exekučných jadier viacjadrového procesora.

- Vypočítavame efektivitu využitia výpočtových zdrojov počítačového systému podľa tohto matematického vzťahu:

$$e = \frac{N_V}{n} \times 100[\%]$$

kde:

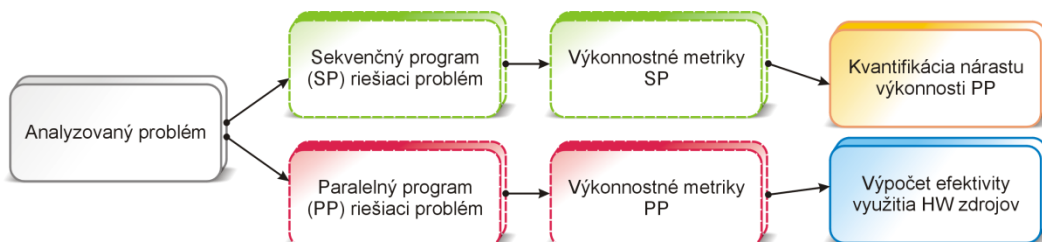
- ❖  $e$  je efektivita využitia výpočtových zdrojov počítačového systému.
- ❖  $N_V$  je nárast výkonnosti paralelného programu voči sekvenčnému programu.
- ❖  $n$  je počet exekučných jadier viacjadrového procesora.

Efektivita je vyjadrená percentuálnou hodnotou, ktorá determinuje priemernú mieru využívania exekučných jadier viacjadrového procesora počas trvania životného cyklu paralelného programu. Ak napríklad zaznamenáme pri testovaní paralelného programu na počítači s 2-jadrovým procesorom nárast výkonnosti 1,8, efektivita využitia výpočtovej kapacity systému bude takáto:

$$e = \frac{1,8}{2} \times 100 = 90 \%$$

Tento výsledok interpretujeme nasledujúcim spôsobom: V priemere využíval paralelný program v čase svojej exekúcie 90 % disponibilných výpočtových zdrojov počítačového systému. Každé exekučné jadro 2-jadrového procesora bolo v priemere 90 % času exekúcie paralelného programu zaneprázdnené vykonávaním strojového kódu tohto programu. Alebo opačne, v priemere bolo každé jadro 2-jadrového procesora 10 % času spracovania paralelného programu nečinné (zostávalo v neaktívnom stave). Prirodzene, optimálnym stavom je 100 % efektivita. Čím bližšie sa k tejto hodnote priblížime, tým je vytvorený paralelný program efektívnejší,

pretože maximalizuje mieru využívania hardvérovej platformy počítačového systému. V tejto súvislosti musíme uviesť, že koeficient efektivity môže nadobúdať aj vyššiu ako 100% hodnotu. S týmto javom sa stretávame pri paralelných programoch, ktoré preukazujú superlineárny nárast výkonnosti.



Obr. 1: Metodika použitá pri realizácii praktických ukážok paralelného programovania

3. Pri stavbe paralelných programov uplatňujeme spravidla koncepciu explicitného paralelizmu s rôznou úrovňou abstrakcie. Okrem jednej praktickej ukážky rozoberáme vždy explicitný paralelizmus s rôznou úrovňou abstrakcie (explicitný paralelizmus s nízkou, strednou a vysokou úrovňou abstrakcie).
4. Pri vytváraní sekvenčných a paralelných programov používame programovacie jazyky C# 4.0 a C++ (podľa štandardu ISO/IEC 14882:2003 a C++0x). V prípade jazyka C#4.0 pracujeme s jeho implementáciou v produkte Visual C# 2010 od spoločnosti Microsoft. C# 4.0 spolupracuje s bázovou knižnicou tried vývojovo-exekučnej platformy Microsoft .NET Framework 4.0. V prípade jazyka C++ používame implementáciu v produkte Visual C++ 2010. Jazyk C++ kooperuje s knižnicou jazyka C++, rovnako ako aj so štandardnou šablónovou knižnicou jazyka C++ (STL).
5. Všetky sekvenčné a paralelné programy sú štandardné konzolové aplikácie. Ich zdrojové kódy sú preložené v ostrých zostavovacích režimoch (Release). Vygenerované priamo spustiteľné súbory sú potom priamo spúšťané