

**ŽILINSKÁ UNIVERZITA V ŽILINE**

---

**AUTOREFERÁT  
DIZERTAČNEJ PRÁCE**

---

**Žilina, apríl 2013**

**Ing. Filip Janovič**



**Žilinská univerzita v Žiline**  
**Fakulta riadenia a informatiky**

**Filip Janovič, Ing.**

Autoreferát dizertačnej práce

**Modelovanie a optimalizácia výkonnosti paralelných algoritmov**

na získanie akademického titulu „**philosophiae doctor**“ (v skratke **PhD.**)  
v študijnom programe doktorandského štúdia  
**Aplikovaná informatika**

v študijnom odbore:

**9.2.9 aplikovaná informatika**

Žilina, apríl 2013

**Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Katedre technickej kybernetiky, Fakulte riadenia a informatiky Žilinskej univerzity v Žiline**

**Predkladateľ:** Ing. Filip Janovič  
Žilinská univerzita v Žiline  
Fakulta riadenia a informatiky  
Katedra technickej kybernetiky

**Školiteľ:** Prof. Ing. Ivan Hanuliak, CSc.  
Žilinská univerzita v Žiline  
Fakulta riadenia a informatiky  
Katedra technickej kybernetiky

**Oponenti:** Prof. Ing. Imrich Rukovanský, CSc.  
Vysoká škola logistiky, o.p.s.  
Katedra logistiky a technických disciplín

Prof. Ing. Petr Dostál, CSc.  
Vysoké učení technické Brno  
Ústav informatiky

Doc. Ing. Ladislav Schwartz, PhD.  
Žilinská Univerzita  
Katedra telekomunikácií

**Autoreferát bol rozoslaný dňa:** .....

Obhajoba dizertačnej práce sa koná dňa ..... o ..... h. pred komisiou pre obhajobu dizertačnej práce schválenu odborovou komisiou v študijnom odbore **9.2.9 aplikovaná informatika, v študijnom programe aplikovaná informatika**, vymenovanou dekanom Fakulty riadenia a informatiky Žilinskej univerzity v Žiline dňa .....

**prof. Ing. Martin Klimo, PhD.**

predseda odborevej komisie  
študijného programu **aplikovaná informatika**  
v študijnom odbore 9.2.9 aplikovaná informatika\*  
Fakulta riadenia a informatiky  
Žilinská univerzita  
Univerzitná 8215/1  
010 26 Žilina

## Obsah

Úvod.....	3
Ciele dizertačnej práce .....	5
1. Paralelné počítače.....	6
2. Komunikačné siete paralelných počítačov .....	6
3. Analýza paralelných algoritmov.....	6
4. Aplikačné vývojové prostredia.....	7
5. Paralelné výpočtové modely.....	8
6. Metódy hodnotenia výkonnosti .....	8
7. Kritéria výkonnosti.....	8
8. Modelovanie výkonnosti .....	8
9. Modelovanie zložitosti algoritmov .....	9
10. Spoločné vlastnosti MPA .....	10
11. Komplexné modelovanie MPA .....	10
12. Modelovanie výkonnosti dynamických MPA .....	13
13. Overenia a merania výkonnosti MPA.....	14
14. Merania a overenia technických parametrov PP.....	17
15. Prínosy.....	22
Záver a perspektívy .....	23
Literatúra .....	25
Publikačná činnosť autora .....	29

## Úvod

Rast zložitých úloh teórie a praxe vyžaduje výkonnejšie výpočtové prostriedky a efektívne algoritmy. Typickými príkladmi zložitých úloh sú úlohy aplikovanej informatiky (umelá inteligencia, zložité kombinatorické problémy, NP ťažké úlohy, robotiky, predpovedanie počasia, úlohy modernej fyziky častíc a pod. Nastúpený dominantný trend k paralelným princípom je aktuálne podporovaný výrobnými technológiami typu SMP (Symmetrical multiprocessors) implementáciou viacerých jadier (Multicore) resp. procesorov (Multiprocessors) na základnej doske (Motherboard) ako i i podpornými vývojovými prostrediami (OpenMP, MPI, Java). Rovnako implementácie paralelných technológií podporuje rastúca priepustnosť vysokorýchlostných komunikačných sietí.

Prvý úspešný krok pre aplikovanie paralelných princípov vykonala v roku 1960 firma IBM, ktorá pre ďalšie zvýšenie výkonnosti svojho veľkého počítača typu Mainframe aplikovala prepojenie skupiny dvoch resp. viacerých uvedených veľkých počítačov, čím poskytla ekonomicky efektívnejšiu zostavu viacerých počítačov. Tým položila základy skupinovej spolupráce počítačov pre výkonnejšie riešenia úloh (Cluster computing). Inú úspešnú formu zrýchlenia spracovania dát predstavoval mohutný (masívny) modulárny paralelný systém ILLIAC IV z roku 1969. Ďalším úspešným krokom v aplikáciách paralelných princípov bol superpočítač Cray-1 z roku 1976, ktorý ako prvý prepracoval sekvenčný spôsob spracovania inštrukcií paralelnými aplikáciami prúdového (Pipelining) a vektorového spracovania toku inštrukcií (Vector processing) s cieľom zvýšenia výkonnosti. Použitá kombinácia rôznych paralelných princípov položila základy rozsiahleho (masívneho) paralelizmu pre výkonnejšie spracovanie dát. Po intenzívnom nástupe paralelných princípov v 80 - tých rokoch sa zdalo, že masívne paralelné počítače (počet procesorov  $p > 100$ ) sa budú dať efektívne realizovať s minimom prídavného výskumu a úsilia. Tento predpoklad sa ale nepotvrдил.

Napriek tomu nadobúdané skúsenosti s implementáciami rôznych paralelných princípov, rôznych komunikačných sietí, vznik a intenzívne rozširovanie počítačových sietí, a to predovšetkým na univerzálnych osobných počítačov po roku 1980 vyvolali zdanlivý návrat k paralelnému využívaniu prepojených počítačov na výkonnejšie riešenie úlohy (Cluster computing). Tento prístup sa postupne stal pri neustále rastúcej výkonnosti osobných počítačov dominujúcim trendom, ktorý sa nevyhnutne premietol i do výkonného paralelného spracovania

HPC (High performance computing). Tento trend bol podporovaný sieťovými prepojeniami stále výkonnejších osobných počítačov typu pracovné stanice (Workstation) pod označením sieť pracovných staníc typu NOW (Network of workstation). Uvedené trendy poskytovali značné výhody včítane vytvorenia spoločnej technickej platformy pre limitované finančné prostriedky užívateľov. Rozhodujúcim faktorom pre oblasť HPC bola dostatočná výkonnosť NOW pre široký počet aplikačných úloh a používaných pracovných zaťažení na základe jednoduchej modulárnosti NOW. Sprievodným faktorom NOW a ich väčších integrovaných systémov (sieť sietí NOW) bola vysoká modulárnosť použitím rovnakých procesorov na základnej doske pracovnej stanice pod označením SMP (Symmetrical multiprocessor). Rozširovanie a využívanie paralelných počítačov (SMP, NOW) po roku 1985 vyvolala súčinnosť nasledovných trendov

- dostupnosť výkonných procesorov (mikroprocesory)
- vysoko výkonné pracovné stanice (single, SMP)
- vysoko rýchlostné prepojovacie siete (komunikačná sieť)
- štandardizované vývojové prostredia pre vývoj paralelnej aplikácie.

V teoretickej oblasti aplikovanie teórie zložitosti na sekvenčné algoritmy uspokojuvito prispieva k riešeniu problematiky zložitosti a výkonnosti. Na tieto dosiahnuté výsledky nadväzuje teória paralelných algoritmov analýzou algoritmov, ktoré sú vhodné pre efektívne riešenie dominujúcimi paralelnými počítačmi. Analýza zložitosti (výkonnosti) paralelného systému v porovnaní s optimalizáciou doby vykonania pri sekvenčných algoritmoch je podstatne zložitejší proces, a to z dôvodov väčšieho počtu vstupných a vnútorných premenných. Pri analýze je potrebné zohľadňovať ďalšie nové faktory, ktoré ovplyvňujú výkonnosť paralelných algoritmov, a to najmä

- veľkosť úlohy (vstupné dáta, počet premenných)
- počet výpočtových uzlov
- prenosové limity komunikačných sietí (rýchlosť, kapacita)
- dostupná veľkosť pamäti
- špecifické parametre danej úlohy
- iné.

Veľký význam v teórii paralelných algoritmov majú kritéria hodnotenia výkonnosti paralelných počítačov a paralelných algoritmov, ktoré umožňujú ich analýzu pre reálne existujúce paralelné počítače a asymptotické predpovedanie správania sa paralelných algoritmov pre teoretické paralelné počítače. Predkladaná dizertačná práca preto vychádza z analýz vývoja paralelných systémov a paralelných algoritmov vo svete a u nás. Súčasťou týchto analýz sú publikované rozvojové koncepcie aktívnych akademických a priemyselných inštitúcií, doma i v zahraničí. Z vykonaných analýz vyplýva, že vedecký výskum v oblasti predkladanej témy práce je vo svete veľmi aktívny čo presvedčivo dokazujú publikované výsledky na medzinárodných vedeckých konferenciách a vo vedeckých časopisoch akademických a vývojových laboratórií.

Na základe vykonaných analýz dizertačná práca smeruje do oblastí komplexného modelovania podstatných oneskorení paralelných algoritmov. Koncentruje sa preto predovšetkým na zložitosť (výkonnosť) základných komunikačných mechanizmov paralelných vývojových prostredí (OpenMP, MPI) pre dominantné paralelné počítače (SMP, NOW, Grid). Výkonnosť týchto komunikačných mechanizmov a analýza ich chovania prispieva k stanoveniu a predikovaní limitujúcich faktorov výkonnosti paralelných algoritmov. Predložená práca v zmysle zadania je smerovaná do oblasti analytického modelovania a optimalizácie výkonnosti aplikačných maticových paralelných algoritmov (MPA) pre dominantné paralelné počítače. Nevyhnutnou súčasťou modelovania výkonnosti MPA je modelovanie a optimalizácia výkonnosti medziprocesorovej komunikácie paralelných procesov, ktoré sa v podstatnej miere podieľajú na sprievodnej réžii aplikovaných maticových úloh. Minimalizácia komunikačnej réžie je preto predpokladom dosiahnutia vysokej výkonnosti maticových paralelných algoritmov. Cieľom je preto prispieť k spresneniu metodiky vývoja efektívnych paralelných algoritmov v oblasti maticových paralelných algoritmov a ich aplikácií, predikcii ich výkonnosti

a minimalizáciu zložitosti (výkonnosti) uvedených paralelných algoritmov (spoločná pamäť, distribuovaná pamäť), a to najmä pre dominantné paralelné počítače typu SMP, NOW a Grid .

## **Ciele dizertačnej práce**

Ciele dizertačnej práce je možné zhrnúť do nasledovných bodov

### **Analýza oneskorení**

- **paralelných počítačov**
  - so spoločnou pamäťou
  - distribuovaná pamäť
    - ◆ NOW
    - ◆ sieť sietí NOW (Grid)
- **paralelných vývojových prostredí**
  - ◆ so spoločnou pamäťou – OpenMP
  - ◆ distribuovaná pamäť – PVM, MPI, Java

### **Teoretická časť práce**

- **Matematická formulácia modelovania zložitosti PA**
  - výpočtová
  - režijná
    - ◆ medziprocesorová komunikácia
    - ◆ paralelizácia úlohy
    - ◆ synchronizácia
  - kritéria
    - ◆ zrýchlenie
    - ◆ efektívnosť
    - ◆ izoeфекtívnosť
- **Asymptotická analýza maticových PA**
  - dekompozičné modely
  - optimalizácia oneskorení (efektívne PA)
  - predikcie zložitosti (výkonnosti)

### **Experimentálna časť práce**

- **Overenia výsledkov**
  - priame merania maticových PA
    - ◆ SMP procesy (spoločná pamäť)
    - ◆ NOW a Grid (distribuovaná pamäť)
- kolektívne komunikačné mechanizmy
  - gather, allgather
  - scatter
  - reduce
- hodnotenie nameraných výsledkov

**Prínosy**

**Závery**

**Literatúra**

# I. Analýza vývoja a stavu vo svete

## 1. Paralelné počítače

Z pohľadu vývoja architektúr klasifikujeme paralelné počítače nasledovne

- ❖ klasické paralelné počítače
  - superpočítače – vo svete
  - masívne paralelné počítače (MPP) – vo svete
    - spoločná pamäť (OpenMP)
    - distribuovaná pamäť (PVM, MPI, Java)
- ❖ skupiny spolupracujúcich počítačov (clusters)
  - NOW (Network of workstations) včítane SMP (Symmetrical multiprocessing)
  - cluster, PC farm
- ❖ Gridy
  - vo svete – integrované využívanie paralelných počítačov
  - na Slovensku – siete NOW, SMP.

## 2. Komunikačné siete paralelných počítačov

Prehľad rôznych realizovaných typov komunikačných sietí je nasledovný

- prepínacie siete (Closove a pod.)
- prstence (kruh)
- dvojrozmerné polia (mriežky), anuloidy (torus)
- booleovské n-rozmerné kocky (hyperkocky)
- premiešania (logaritmické siete)
- štruktúra motýľ
- omega
- dokonalé premiešania
- premiešania s výmenou
- k-cestné premiešania
- spätné výmeny
- De Bruijnové siete
- Banyan siete
- Batcherové siete
- Benešove siete
- stromy (X-stromy, H-stromy, tlsté stromy a pod.)
- hyperstromy
- náhodné riadenie komunikácie (hešované a pod.)

### 2.1. Komunikačné technológie prenosu dát

Analýzu komunikačných technológií možno rozdeliť do dvoch základných skupín

- prepojovanie kanálov (Switched technology)
- technológia postupného nespojitého prenosu dát od jedného prenosového uzla k nasledovnému prenosovému uzlu s dočasným uchovávaním prenášaných dát počas prenosu metódou „zapamätaj a pošli ďalej“ S&F (Store – and - forward).

## 3. Analýza paralelných algoritmov

### 3.1. Vývoj paralelného algoritmu

Vo všeobecnosti proces tvorby paralelných algoritmov zahŕňa nasledovné činnosti



- rozdelenie úlohy na nezávislé procesy (Decomposition)
- priradenie jednotlivých procesov uzlom počítačovej siete na výpočet (Mapping)
- komunikácia medzi paralelnými procesmi IPC (Interprocess communication)
- optimalizácia výkonnosti (Tunning)

### 3.2. Dekompozičné modely

Návrh paralelného algoritmu by mal už v počiatočnej etape návrhu obsahovať predpokladanú stratégiu rozkladu danej aplikácie na jej jednotlivé časti (paralelné procesy) a spôsob rozdelenia potrebných dát pre vykonanie paralelných procesov. Výber dekompozičnej metódy má zásadný vplyv na ďalší postup vývoja paralelného algoritmu. Základné dekompozičné metódy sú nasledovné

- prirodzená paralelná dekompozícia
- doménová dekompozícia
- riadená dekompozícia
- objektovo - orientovaná dekompozícia
- dekompozícia rozsiahlych aplikácií.

## 4. Aplikačné vývojové prostredia

Viacprocesorový systém so spoločnou pamäťou (SMP) typicky používa pre komunikáciu spoločné premenné (OpenMP, Java), kým paralelný program pre distribuované paralelné systémy (NOW, Grid), používa komunikáciu správ (PVM, MPI, Java).

### 4.1. Vlastnosti štandardu Open MP

Základné charakteristiky štandardu OpenMP sú nasledovné

- prenosnosť (portability)
- škalovalnosť (scalability)
- efektívnosť (efficiency)
- vysokoúrovňový (high level)
- podpora paralelizmu dát
- jednoduchosť používania
- funkčná mohutnosť

### 4.2 MPI

MPI (Message passing interface) je štandardom pre vývoj aplikačných programov s komunikáciou správ, t. j. pre paralelné počítače s distribuovanou pamäťou (asynchrónne paralelné počítače) typu NOW a Grid. Podstatou MPI je poskytnúť knižnicu štandardných komunikačných programov pre tvorbu prenosných zdrojových kódov paralelných programov medzi rôznymi asynchrónnymi paralelnými počítačmi (transparentnosť) a efektívnych paralelných algoritmov s komunikáciou správ. MPI preto nie je typický programovací jazyk ale je to špecifikácia knižnice programov, ktoré mohli byť volané v programoch programovacích jazykov FORTRAN (vo svete) a C/C++. MPI poskytuje rozsiahly súbor komunikačných služieb pre dvojbodové prenosy a kolektívnych komunikčných operácií, globálne paralelné výpočty a synchronizáciu paralelných procesov.

#### 4.2.1. Java

Je profilovaný ako objektovo-orientovaný jazyk s podporou viacvetvovým procesom (multithreading). Táto podpora zahrňuje paralelné vetvy pri používaní spoločných premenných (spoločná pamäť). Rovnako obsahuje mechanizmy pre vývoj distribuovaných paralelných algoritmov prostredníctvom zabudovaného modulu java.net.

## 5. Paralelné výpočtové modely

Paralelný výpočtový model je abstraktným modelom spôsobu paralelného výpočtu. Model charakterizujú schopnosti a možnosti paralelného počítača, ktoré sú pre paralelný spôsob výpočtu určujúce. Z pohľadu vývojára – programátora vzhľadom výpočtové modely sú

- SPMD (rovnaký proces, paralelné dáta) s nasledovnými vlastnosťami
  - spoločná pamäť
  - komunikácia cez spoločnú pamäť
  - vývojové prostredia OpenMP, OpenMP Threads
- MPMD (paralelné procesy, paralelné dáta) s nasledovnými vlastnosťami
  - žiadna spoločná pamäť (distribuovaná pamäť)
  - riadenie a spolupráca procesov iba cez komunikáciu správami
  - vývojové prostredia PVM, MPI, Java.

## 6. Metódy hodnotenia výkonnosti

Metódy hodnotenia výkonnosti paralelných počítačov a algoritmov sú nasledovné

- analytické metódy
- simulačné
- asymptotická analýza
- zákony, teóremy (Amdahl, Gustafson a pod.).
- priame merania
  - experimentálne
  - modelovacie nástroje
  - výkonnostné testy (benchmarks).

## 7. Kritéria výkonnosti

Kritéria výkonnosti vychádzajú z rozšírených klasických kritérií pre sekvenčné počítače a algoritmy (MIPS, Mflops, Drystone, Whestone, Khornestone, Specs, TPS a pod.) a špecializovaných pre teóriu a prax paralelných počítačov a algoritmov. Medzi základné rozšírené špecializované kritéria patria

- špecializované
  - paralelné zrýchlenie (speed up)
  - efektívnosť (efficiency)
  - izoeftívnosť (isoefficiency).

## 8. Modelovanie výkonnosti

Modelovanie výkonnosti paralelných počítačov je nástrojom pre ich efektívne aplikačné použitie vrátane predpovedanie (predikcia) správania teoretických paralelných počítačov. Pre hodnotenie výkonnosti môžeme použiť nasledovné metódy

- analytické metódy
  - teória hromadnej obsluhy (queueing theory)
  - Petriho siete (Petri nets)
  - asymptotická analýza (asymptotic analysis)
- simulácia
  - diskretná
  - statická

- experimentálne merania
  - nástroje modelovania
  - merania výkonnosti algoritmov
  - merania technických parametrov
  - špecializované testy
  - SPEC testy.

## 9. Modelovanie zložitosti algoritmov

### 9.1. Aplikácia teórie zložitosti

Jedným z najdôležitejších teoretických aspektov zložitosti (výkonnosti) výpočtov je dostupnosť potrebných zdrojov pre riešenie úlohy. Najdôležitejšie parametre pri výpočtových algoritmoch sú doba výpočtu a kapacita pamäti (priestor). Pre konkrétny problém je potrebné odvodiť minimálne ohraničenia (horné, dolné) požadovaných zdrojov.

### 9.2. Komplexná analýza paralelných algoritmov

Pri analýze algoritmov, je často ťažké, alebo dokonca nemožné, odvodiť presné výrazy pre parametre typu výpočtový čas, zrýchlenie, efektívnosť a izoeftívnosť včítane režijných funkcií. Vo väčšine prípadov sa preto usilujeme minimálne o aproximáciu presných výrazov.

### 9.3. Asymptotická analýza algoritmov

Pre asymptotické hodnotenie zložitosti (výkonnosti) sa zaviedli označenia pre určenie rádu ich ohraničení, a to horného, dolného alebo presného. Nech  $f, g$  sú funkcie, ktoré sú definované na množine prirodzených čísel ( $N$ ). Nech  $N$  zodpovedá vstupnej veľkosti problému. Potom

- $f(N) = O(g(N))$ , ak existujú konštanty  $c$  a  $N_0$  také, že  $f(N) \leq c \cdot g(N)$  pre všetky  $N \geq N_0$ . Tým  $O$  udáva hornú hranicu
- $f(N) = \Omega(g(N))$ , ak existujú konštanty  $c$  a  $N_0$  také, že  $f(N) \geq c \cdot g(N)$  pre všetky  $N \geq N_0$ . Tým  $\Omega$  udáva dolnú hranicu
- $f(N) = \Theta(g(N))$ , ak existujú konštanty  $c_1, c_2$  a  $N_0$  také, že  $c_1 \cdot g(N) \leq f(N) \leq c_2 \cdot g(N)$  pre všetky  $N \geq N_0$ . Tým  $\Theta$  udáva presné ohraničenie.

Uvedené označenia  $O, \Omega, \Theta$  predstavujú triedy funkcií podobnej zložitosti pri ignorovaní konštánt. Aplikujeme ich pre hodnotenie zložitosti oneskorenia paralelných algoritmov (paralelný výpočet, režijné oneskorenia, asymptotická analýza).

## II. Teoretická časť – vybrané state

### 10. Spoločné vlastnosti MPA

Pre maticové paralelné algoritmy budeme uvažovať nasledovný obecný tvar štvorcovej matice  $A = n \times n$  (obr. 1.)

$$A = \begin{pmatrix} a_{11}, & a_{12}, & \dots & , & a_{1n} \\ a_{21}, & a_{22}, & \dots & , & a_{2n} \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ a_{n1}, & a_{n2}, & \dots & , & a_{nn} \end{pmatrix}$$

Obr. 1. Štvorcová matica  $n \times n$ .

Dôvodom uvažovania štvorcovej matice je redukcia počtu premenných v procese odvodzovania analytických vzťahov pre výkonnosť a funkciu izoeftívnosti ( $n=m$ ). Tento prehľadnejší postup podporujú i nasledovné ďalšie dôvody

- maticu typu  $n \times m$  môžeme transformovať na štvorcovú maticu  $n \times n$  rozšírením buď počtu riadkov (ak  $m < n$ ) alebo stĺpcov (ak  $m > n$ )
- v odvodených vzťahoch pre uvažovanú výpočtovú zložitosť matice ako  $n^2$  (štvorcová matica) v prípade obdĺžnikovej matice použijeme jej zložitosť ako súčin  $n \times m$ .

Spoločné vlastnosti maticových algoritmov (MPA) sú nasledovné

- používajú doménové dekompozičné modely, v ktorých ako doména prevažne vystupuje pevná dátová štruktúra (dátové prvky matice)
- matica ako doména je dobre paralelizovateľná teoreticky až na úroveň jedného dátového prvku matice. Aplikovanie takéhoto stupňa rozkladu matice by nebolo efektívne.

### 11. Komplexné modelovanie MPA

Pre paralelné počítače typu NOW a Grid modelovanie ich výkonnosti nevyhnutne vyžaduje komplexnú analýzu rozsahu vplyvov všetkých režijných oneskorení, a to konkrétne

- vlastný výpočet ( $T_{comp}$ )
- architektúra paralelného systému ( $T_{arch}$ )
- paralelizácia algoritmu ( $T_{par}$ )
- medziprocessorová komunikácia IPC ( $T_{comm}$ )
  - parameter inicializácie komunikácie  $t_s$  (Start up time)
  - parameter zvolenej dátovej jednotky v prenášaných slovách  $t_w$  (Words) transmission)
  - smerovanie (Routing)
- synchronizácia ( $T_{syn}$ ).

V zmysle zavedeného označenia základných parametrov  $s$  (zložitosť úlohy) a  $p$  (počet výpočtových uzlov) pre celkové oneskorenie paralelného algoritmu  $T(s, p)$  potom platí

$$T(s, p) = T_{comp}(s, p) + T_{arch}(s, p) + T_{par}(s, p) + T_{comm}(s, p) + T_{syn}(s, p)$$

Sumu označených oneskorení označíme ako funkciu režijných strát  $h(s, p)$  nasledovne

$$h(s, p) = T_{arch}(s, p) + T_{par}(s, p) + T_{comm}(s, p) + T_{syn}(s, p)$$

Potom pre celkové oneskorenie paralelného algoritmu  $T(s, p)$  platí

$$T(s, p) = T_{comp}(s, p) + h(s, p)$$

Najzávažnejším oneskorením pre súčasné dominantné paralelné počítače (NOW, Grid) je práve komunikačné oneskorenie vo forme medziprocessorovej komunikácie IPX (Interprocess communication). Pre odvodenie špecializovaných výkonnostných kritérií je potrebné odvodenie analytických výrazov včítane vplyvov podstatných režijných oneskorení. Rovnako pre analýzu izoefektívnosti, ako kritéria rozsahu škálovateľnosti, je potrebné uvažovať nielen vlastný paralelný výpočet ale i vplyv režijnej funkcie  $h(s, p)$  ako

$$w(s) = \max[\text{výpočet}, h(s, p)]$$

### 11.1. Dekompozičné modely matice

Maticové paralelné algoritmy (MPA) pre paralelné počítače typu NOW a Grid zodpovedajú výpočtovému modelu typu MPMD, pričom ako domény môžeme uvažovať

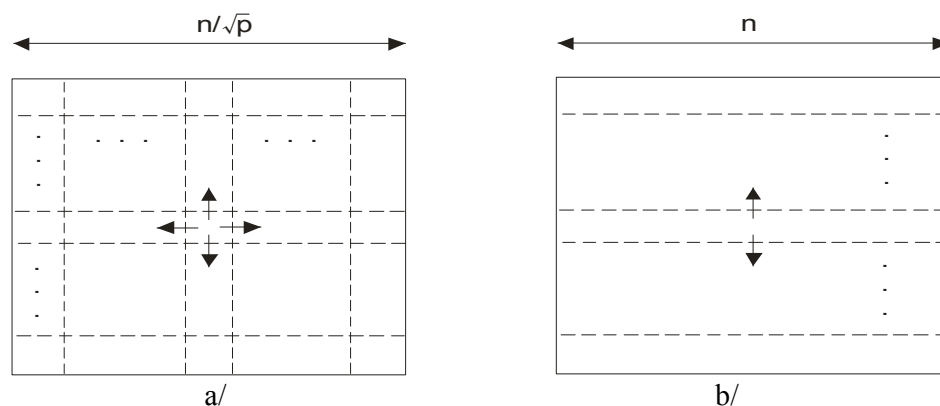
- vykonávané činnosti nad prvkami matice (výpočtový doménový paralelizmus)
- prvky matice (doménový paralelizmus dát).

Uvažovaná štvorcová matice obsahuje  $n^2$  prvkov, ktoré sa rovnakým počtom rozdelia medzi  $p$  procesorov daného paralelného počítača, t. j. každý výpočtový uzol dostane na výpočet dostane skupinu  $n^2 / p$  prvkov matice ( $p$  paralelných procesov). Pridelovanie skupín prvkov matice umožňuje dva základné dekompozičné modely, a to

- rozklad matice na bloky (Blocks)
- dekompozičný model (rozklad) matice na pásy (Strips), a to nasledovne
  - na riadky (Rows)
  - na stĺpce (Collums).

Pre medziprocessorovú komunikáciu budeme pre každú dekompozičnú metódu predpokladať maximálnu medziprocessorová komunikácia v ktorej nová vypočítaná hodnota každého  $ij$  - prvku matice sa odosiela všetkým štyrom susedným prvkom ( $i+1, i-1, j+1, j-1$ ) matice a rovnako každý  $ij$  prvok prijíma nové vypočítané hodnoty od všetkých štyroch susedných prvkov matice. Vo všeobecnosti rozkladom na bloky dostaneme  $B_p$  – štvorcových blokov. Štvorcové bloky  $B_p$  pridelieme po jednom každému výpočtovému uzlu daného paralelného počítača (paralelný proces). Komunikačné dôsledky rozkladu matice na bloky (komunikačný model) ilustruje obr. 2. a/. Každý paralelný proces preto vysiela štyri správy a rovnako prijíma štyri správy na konci každého výpočtového kroku. V dôsledku toho oneskorenie medziprocessorovou komunikáciou pre každý výpočtový krok  $T_{commb}$  je daný ako Táto rovnica je platná pre počet paralelných procesov  $p \geq 9$ , pretože iba vtedy je možné vytvoriť minimálne jeden štvorcový blok so štyrmi komunikačnými hranami. Pre komunikačné oneskorenia pri aplikovaní dekompozície matice na bloky (režijná funkcia) platí

$$T(s, p)_{commb} = T_{commb} = h(s, p) = 8 \left( t_s + \frac{n}{\sqrt{p}} t_w \right)$$



Obr.2. Komunikačné dôsledky dekompozičných modelov a/ na bloky b/ na pásy.

Podobne premietnutie komunikačných dôsledkov pre dekompozičný model rozkladu švorcovej matice na pásy (riadky alebo stĺpce) ilustruje obr. 2./b. Pre každý vytvorený paralelný proces (riadky, stĺpce) medziprocesorová komunikácia znamená odoslanie dvoch správ susedným procesorom a rovnako prijatie dvoch správ od susedných procesorov. Predpokladom je uloženie prvkov pásu matice (riadok alebo stĺpec) do jednej správy. Doba oneskorenia medziprocesorovou komunikáciou pre jeden výpočtový krok  $T_{comms}$  je potom určená ako

$$T_{comms} = 4 (t_s + n t_w)$$

, kde  $t_s$  a  $t_w$  je sú zavedené komunikačné parametre medziprocesorovej komunikácie. Pre konkrétnu architektúru paralelného počítača a konkrétne architektúru komunikačnej siete NOW veličiny  $t_s$  a  $t_w$  sú konštanty (súčasť technických parametrov).

### 11.1.1. Odvođenje vzťahy pre výkonnostné kritéria

Pre analýzu výkonnosti MPA pri použití modelu dekompozície matice na bloky sumárne uvedieme odvodené výsledné vzťahy pre zavedené kritéria výkonnosti, a to nasledovne

- Oneskorenie sériovým výpočtom  $T(s, 1)$

$$T(s, 1) = n^2 t_c$$

- Oneskorenie vlastným paralelným výpočtom  $T(s, p)_{comp}$

$$T(s, p)_{comp} = \frac{n^2 \cdot t_c}{p}$$

- Zavedená režijná funkcia  $h(s, p)$

$$T(s, p)_{comm} = T_{commb} = h(s, p) = 8 \left( t_s + \frac{n}{\sqrt{p}} t_w \right)$$

- Komplexné oneskorenie výpočtom paralelného algoritmu  $T(s, p)$

$$T(s, p) = T(s, p)_{comp} + h(s, p) = \frac{n^2 \cdot t_c}{p} + 8 \left( t_s + \frac{n}{\sqrt{p}} t_w \right)$$

- Komplexné paralelné zrýchlenie  $S(s, p)$

$$S(s, p) = \frac{T(s, 1)}{T(s, p)} = \frac{n^2 p t_c}{n^2 t_c + 8 (p t_s + \sqrt{p} n t_w)}$$

- Komplexná efektívnosť  $E(s, p)$

$$E(s, p) = \frac{S(s, p)}{p} = \frac{n^2 t_c}{n^2 t_c + 8 (p t_s + \sqrt{p} n t_w)}$$

a pre dekompozičný model rozkladu matice na pásy (stĺpce, riadky) pre komplexné hodnotenie výkonnosti MPA platia výsledné vzťahy

- Oneskorenie sériovým výpočtom  $T(s, 1)$

$$T(s, 1) = n^2 t_c$$

- Oneskorenie vlastným paralelným výpočtom  $T(s, p)_{comp}$

$$T(s, p)_{comp} = \frac{n^2 t_c}{p}$$

- Zavedená režijná funkcia  $h(s, p)$

$$T(s, p)_{comm} = T_{coms} = h(s, p) = 4 (t_s + n t_w)$$

- Komplexné oneskorenie paralelného výpočtu  $T(s, p)$

$$T(s, p) = T(s, p)_{comp} + h(s, p) = \frac{n^2 t_c}{p} + 4(t_s + n t_w)$$

- Komplexné paralelné zrýchlenie  $S(s, p)$

$$S(s, p) = \frac{T(s, 1)}{T(s, p)} = \frac{n^2 p t_c}{n^2 t_c + 4 p (t_s + n t_w)}$$

- Komplexná efektívnosť  $E(s, p)$

$$E(s, p) = \frac{S(s, p)}{p} = \frac{n^2 t_c}{n^2 t_c + 4 p (t_s + n t_w)}$$

## 11.2. Asymptotická analýza škálovateľnosti MPA

Pre asymptotickú analýzu škálovateľnosti MPA použijeme definované kritérium izoeftívnosti  $w(s)$ . Pre asymptotickú veľkosť  $w(s)$  všeobecne platí

$$w(s) = \max [T(s, p)_{comp}, h(s, p)]$$

Pre asymptotickú analýzu oneskorenia vlastným výpočtom  $T(s, p)_{comp}$  vzhľadom na rastúci počet procesorov  $p$  platí nasledovná limita

$$\lim_{p \rightarrow \infty} T_{comp}(s, p) = \lim_{p \rightarrow \infty} \frac{T(s, 1)}{p} = \lim_{p \rightarrow \infty} \frac{n^2 \cdot t_{comp}}{p} = 0$$

pri predpoklade dostatočnej paralelizácie úlohy. Tento predpoklad je pre každý maticový algoritmus splnený.

Celkové asymptotické funkcie izoeftívnosti  $w(s)$  pre dekompozíciu na pásy a bloky sú dané ako

$$w(s)_{bloky} = \max \left[ 8 K p \frac{t_s}{t_c}, 8 K n \sqrt{p} \frac{t_w}{t_c} \right] \quad w(s)_{pásy} = \max \left[ 4 K p \frac{t_s}{t_c}, 4 K n p \frac{t_w}{t_c} \right]$$

Spoločná (optimalizovaná) asymptotická funkcia izoeftívnosti  $w(s)_{opt}$  je daná ako

$$w(s)_{opt} = \max \left[ 8 K p \frac{t_s}{t_c}, 4 K n p \frac{t_w}{t_c} \right]$$

Optimalizovanú funkcia izoeftívnosti určuje pri splnení podmienky  $n \geq 2 t_s / t_w$  jediný nasledovný výraz

$$w(s)_{opt} = 4 K n p \frac{t_w}{t_c}.$$

## 12. Modelovanie výkonnosti dynamických MPA

Paralelné maticové algoritmy sú v rade aplikácií spojené s dynamicky premennou veľkosťou paralelných procesov pri opakovaných činnostiach s dekomponovanými časťami prvkov matice. V takom prípade počet vykonávaných činností v jednotlivých dekomponovaných častiach matice sa mení a analytické odvodenie celkového počtu komunikačných operácií v dekomponovaných paralelných procesoch je veľmi obtiažne. Jedná sa preto o maticové paralelné algoritmy, v ktorých postupná zmena veľkosti počtu prvkov matice (doména) vyvoláva zmeny výpočtovej zložitosti. V dôsledku toho dynamický charakter domény prvkov matice v priebehu vykonávania MPA sa premieta do dynamických zmien veľkostí dekomponovaných paralelných procesov. Maticové paralelné algoritmy s týmito vlastnosťami označíme ako dynamické. Pre funkciu režijných strát dynamických MPA použijeme dva nasledovné postupy

- rozloženie dynamického charakteru vykonávania MPA na postupnosť po sebe nasledujúcich krokov so statickým charakterom domén počas jedného kroku
- modelovanie výkonnosti kolektívnych príkazov MPI prostredia, ktoré sú spojené s oneskoreniami príkazov pre synchronizáciu ( $T_{syn}(s, p)$ ) a medziprocesorovú komunikáciu  $T_{comm}(s, p)$ . Pre konkrétny aplikačný maticový algoritmus režijné straty dostaneme ako sumu násobkov oneskorení použitých synchronizačných a komunikačných príkazov.  
Pre analýzu oneskorení sa jedná o nasledovné kolektívne komunikačné MPI príkazy
- zber dát typu Gather
- zber dát typu Allgather
- zber a sumarizácie dát typu Reduce
- rozdelenia dát (rozptyl) typu Scatter
- rozdelenia dát (rozptyl) typu Broadcast

### III. Experimentálna časť – vybrané výsledky

#### 13. Overenia a merania výkonnosti MPA

Priame merania výkonnosti (MPA) so spoločnou pamäťou vyžadujú navrh metodiky meraní pre jeden (sekvenčný), viaceré procesy so spoločnou pamäťou paralelných počítačov typu SMP a viaceré procesy s distribuovanou pamäťou.

##### 13.1. Meranie oneskorení

Pre experimentálne merania oneskorení môžeme použiť dostupné služby daného paralelného vývojového prostredia (služby MPI, Win API 32 resp 64 a pod.) Napr. na merania doby vykonávania jednotlivých častí paralelných algoritmov použijeme nasledovné funkcie Win API 32 resp. 64

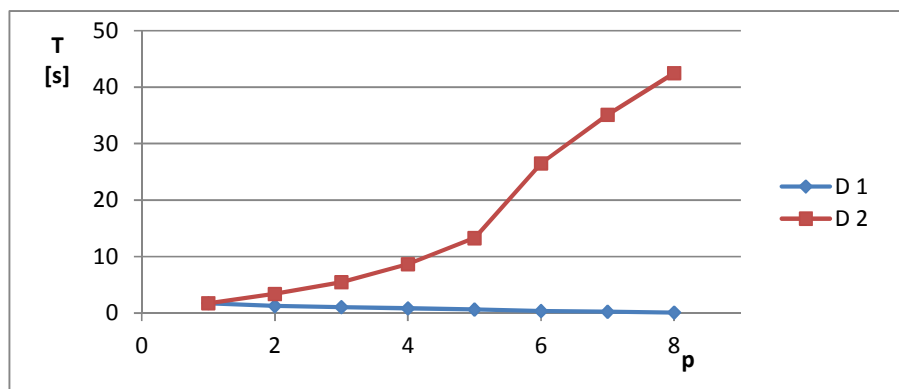
- QueryPerformanceCounter (vracia aktuálnu hodnotu stavu čítača)
- QueryPerformanceFrequency (frekvencia počítania za sekundu).

Uvedený postup je pre merania sledovaných oneskorení univerzálny, t.j. možno ho použiť pre merania výpočtovej časti paralelného algoritmu rovnako ako pre merania režijných oneskorení paralelných procesov alebo typických technických parametrov paralelných počítačov.

##### 13.2. Merania na paralelných počítačoch typu SMP

Pre ilustráciu vplyvu dekompozičného modelu na výkonnosť maticového paralelného algoritmu (MPA) sme vykonali merania dvoch variant paralelného násobenia, ktoré sme analyzovali v teoretickej časti práce. Merania sme vykonali na 8 jadrovom paralelnom počítači typu Intel Xeon (2x Intel Xeon 5335 quad core) na pracovisku školiteľa. Prvý dekompozičný model (D1) zodpovedá klasickému paralelnému násobeniu. Druhý alternatívny dekompozičný model (D2) zodpovedá alternatívne paralelnému násobeniu, v ktorom sa pre získanie výsledného prvku matice C popri násobení príslušných prvkov matice A a B vyžaduje zlučovanie vytváraných priebežných medzivýsledkov. Prídavná výpočtová náročnosť je úmerná veľkostiam vstupných matic A i B ako to vyplýva z porovnania časovej náročnosti uvedených dekompozičných modelov podľa obr. 3.





Obr. 3. Porovnanie časovej náročnosti paralelného násobenia matíc.

### 13.3. Merania na paralelných počítačoch vo svete

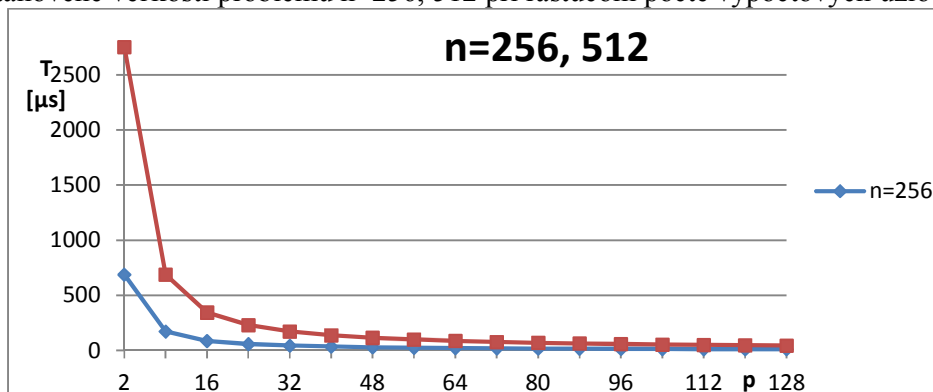
Po overení dosiahnutých teoretických výsledkov na dostupných paralelných počítačoch predpokladáme porovnanie s podobnými architektúrami masívnych paralelných počítačov vo svete pre vysoko výkonné počítanie HPC. Jedná sa o strediská paralelných počítačov Európskej únie, a to HPC strediská EU (EPCC Edinburgh, CESCA Barcelona, CINECA Bologna, IDRIS Paris, SARA Amsterdam, HLRS Stuttgart). Predpokladané merania môžeme vykonať buď v rámci umožnenia diaľkového prístupu pre vhodný paralelný počítač vo svete alebo priamo pobytom v niektorom z uvedených HPC stredísk EU.

### 13.4. Merania pre paralelné počítače typu NOW

Druhy vykonávaných meraní MPA pre paralelné počítače typu NOW rozdelíme nasledovne

- testovanie výkonnosti (kalibrácia) pracovných staníc NOW pre rôznu vstupnú záťaž
- merania oneskorení vlastným paralelným výpočtom  $T(s, p)_{\text{comp}}$ , komunikačných oneskorení  $T(s, p)_{\text{comm}}$ , resp iných podstatných režijných oneskorení nasledovne
  - meraní celkového oneskorenia paralelného vykonávania algoritmu  $T(s, p)_{\text{complex}}$  od vstupnej veľkosti problému  $s$ , počtu výpočtových uzlov  $p$  a pod.
  - meraní oneskorení častí celkovej doby paralelnej činnosti algoritmu  $T(s, p)_{\text{complex}}$ 
    - ◆ výpočtová časť  $T(s, p)_{\text{comp}}$
    - ◆ komunikačné zaťaženie  $T(s, p)_{\text{comm}}$
    - ◆ iné podstatné režijné oneskorenia.
- merania technických parametrov paralelných počítačov
  - oneskorenie priemerného počtu vykonávaných operácií  $t_c$
  - komunikačné technické parametre  $t_s$  a  $t_w$ .

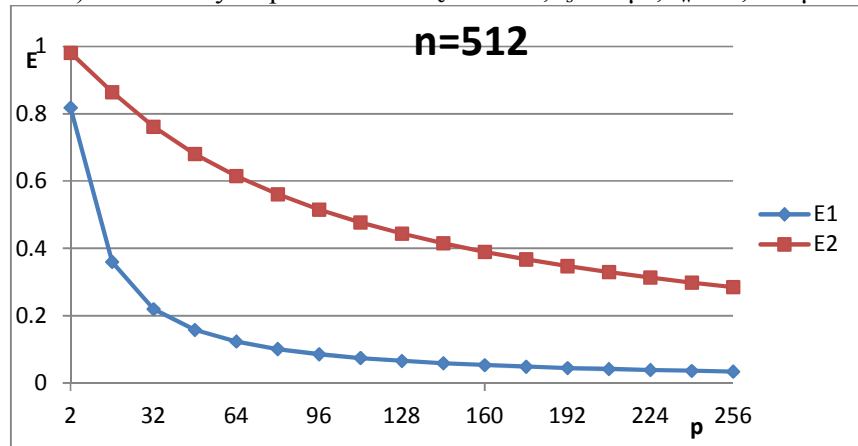
Obr.4 ilustruje overenie limitného charakteru klesania oneskorenia výpočtom  $T(s, p)_{\text{comp}}$  pre dve stanovené veľkosti problému  $n=256, 512$  pri rastúcom počte výpočtových uzlov  $p$ .



#### Obr. 4. Ilustrácia limitného chovania $T(s,p)_{comp}$ .

Obr.5. ilustruje vplyv architektúry paralelného počítača na efektívnosť  $E(s,p)$  vykonávaného paralelného algoritmu s rovnakým dekompozičným modelom. Použité paralelné počítače majú nasledovné technické parametre

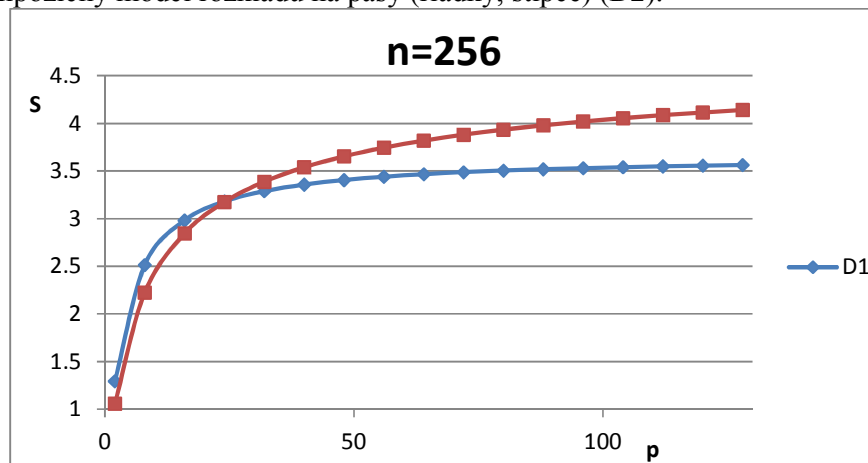
- efektívnosť  $E1 = E1(s,p)$  zodpovedá paralelnému počítaču typu NOW, a to SP2 s technickými parametrami  $t_c = 4,2 \text{ ns}$ ,  $t_s = 35 \text{ } \mu\text{s}$ ,  $t_w = 0,23 \text{ } \mu\text{s}$
- efektívnosť  $E2 = E2(s,p)$  zodpoveda paralelnému počítaču Cray T3DE (klasický superpočítač) s technickými parametrami  $t_c = 11 \text{ ns}$ ,  $t_s = 3 \text{ } \mu\text{s}$ ,  $t_w = 0,063 \text{ } \mu\text{s}$ .



Obr.5. Vplyv architektúry PP na efektívnosti  $E(s,p)$ .

Obr.6 ilustruje závislosť rastu paralelného zrýchlenia  $S(s,p)$  pri zvyšovaní sa veľkosti paralelného systému n (počet procesorov) pre veľkosť problému n = 256 (rozmer matice) pre dekompozičné modely

- dekompozičný model rozkladu na bloky (D1)
- dekompozičný model rozkladu na pásy (riadky, stĺpce) (D2).



Obr. 6. Vplyv dekompozičných modelov na zrýchlenie  $S(s,p)$ .

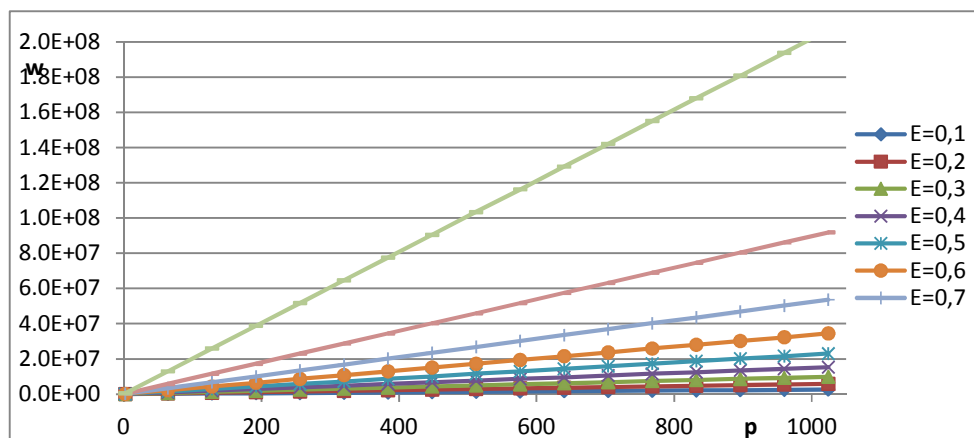
### 13.5. Funkcie izoeftívnosti MPA

Optimalizovaná funkcia izoeftívnosti predstavuje spoločnú funkciu pre oba analyzované dekompozičné modely (bloky, pásy) a je nasledovná

$$w(s)_{opt} = \max \left[ 8 K p \frac{t_s}{t_c}, 4 K n p \frac{t_w}{t_c} \right]$$

V teoretickej časti sme ilustratívne použili technické parametre zástupcu paralelných počítačov typu NOW, a to SP 2. a odvodili hraničné podmienky pre  $n$ .

Obr. 7 ilustruje priebeh funkcií izoeфекtívnosti pre jednotlivé konštantné hodnoty efektívnosti  $E(s,p)$  ( $E = 0,1$  až  $0,9$ ) pre  $n = 1024$  (dominantný druhý výraz  $w(s)_{opt}$ ) a pre technické parametre  $t_s = 35 \mu s$ ,  $t_w = 0,23 \mu s$ ,  $t_c = 0,021 \mu s$ ).



Obr. 7. Funkcie izoeфекtívnosti pre  $n \geq 305$  ( $n=512$ ).

## 14. Merania a overenia technických parametrov PP

### 14. 1. Špecifikácia meraní

Merania predpokladajú experimentálne zistenia resp. upresnenia požadovaných technických parametrov, a to

- priemerná doba výpočtovej operácie (pre daný paralelný počítač konštanta  $t_c$ )
- prenosové parametre (pre daný paralelný počítač konštanty  $t_s$ , a  $t_w$ ).

### 14.2. Priemerná doba výpočtovej časti matíc

Pre určenie technického parametra  $t_c$  (priemerná doba výpočtovej operácie) sme stanovili nasledovné kritéria výberu vhodného paralelného algoritmu

- proporcionálne zastúpenie všetkých aritmetických operácií – sčítanie, odčítanie, násobenie, delenie s presnými hodnotami pre ich počty
- dostatočný reprezentatívny počet vykonávaných aritmetických operácií pre štatistické vyhodnotenie
- jednoduché aplikačné použitie pre iné paralelné počítače
- eliminácia vplyvu medziprocessorovej komunikácie IPC.

Na základe uvedených kritérií sme vybrali výpočtovú časť algoritmu (sekvenčný, paralelný) pre Gaussovú eliminačnú metódu (GEM) rozkladom na dolnú trojuholníkovú metódu. Výpočtová zložitosť GEM algoritmu je daná ako  $n^3/3$  operácií. Pri aplikovaní zavedeného označenia asymptotická zložitosť je daná ako  $O(n^3)$  operácií. Výpočtová zložitosť sa uvádza v analyzovaných bezrozmerných výpočtových krokoch (v uvažovanom prípade počet inštrukcií). Skutočný celkový počet vykonávaných aritmetických operácií IPC (Instruction per computation) dostaneme ako sumu jednotlivých počtov vykonávaných operácií delenia  $IPC_{div}$ , násobenia  $IPC_{mul}$  a operácií sčítania  $IPC_{add}$  nasledovne

$$IPC_{real} = (IPC_{div} + IPC_{mul} + IPC_{add}) = \left( \frac{n(4n^2 + 9n - 7)}{6} \right)_{div+mul+add}$$

Pri zanedbaní prvého výrazu pre  $IPC_{div}$  zlúčením druhého a tretieho výrazu dostaneme jednoduchší aproximačný vzťah počtu vykonávaných operácií. Pre  $IPC_{aprox}$  potom platí

$$IPC_{aprox} \cong (IPP_{mul} + IPP_{add}) \cong \left( \frac{(2n^3 + 3n^2 - 5n)}{3} \right) \cong \left( \frac{n(2n^2 + 3n - 5)}{3} \right)_{mul+add}$$

Pre prípad znalosti hodnoty parametra  $t_{add}$  (rozšírený dostupný technický parameter) pre presné určenie  $T(s, p)_{comp}$  prepočítame počty všetkých ostatných aritmetických inštrukcií iba na inštrukcie sčítania s oneskorením  $t_{add}$ . Pri prepočte uvažujeme nasledovné relácie oneskorení pre aritmetické inštrukcie násobenia (Multiply) a delenia (Divide)

- $t_{mul}$  – oneskorenie inštrukcie násobenia, pričom  $t_{mul} = 3 t_{add}$
- $t_{div}$  – oneskorenie inštrukcie delenia, pričom  $t_{div} = 3 t_{add}$ .

Po naznačenom dosadení uvedených relácií a úpravach dostaneme výsledný výraz pre celkový skutočný počet inštrukcií sčítania  $(IPC_{real})_{add}$  a pre  $(IPC_{aprox})_{add}$  ako

$$(IPC_{real})_{add} = \left( \frac{8n^3 + 21n^2 - 11n}{6} \right)_{add} \quad (IPC_{aprox})_{add} \cong \left( \frac{4n^3 + 6n^2 - 10n}{3} \right)_{add}$$

Pre aplikačné použitie ukážeme postup pre získanie hodnôt technických parametrov  $t_{ci}$  pre testované paralelné počítače. Následne ukážeme aplikačné použitie získaného technického parametra  $t_c$ , a to pre

- overenie presnosti aproximačného vzťahu  $IPC_{aprox}$ .
- simulované výpočty oneskorenia vlastným výpočtom  $T(s, p)_{comp}$
- paralelný počítač typu NOW paralelný počítač typu superpočítač.

### 14.3. Postup získania technického parametra $t_c$

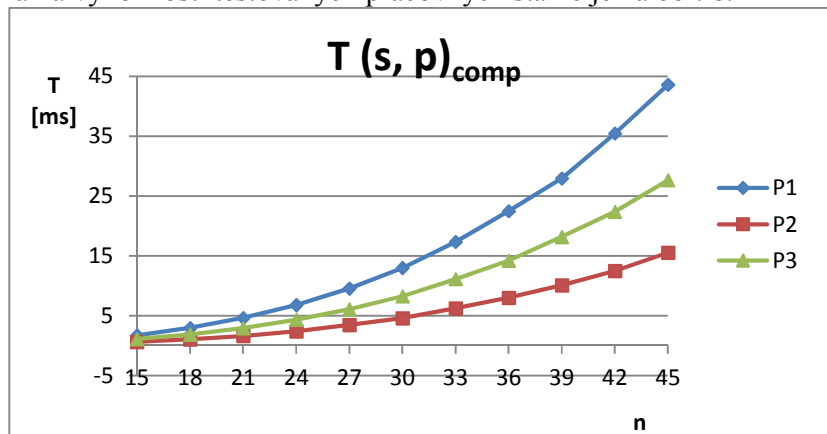
Pri uvážení priemernej doby pre vykonanie inštrukcie  $t_c$  (technický parameter paralelného počítača) dostaneme hodnotu výpočtového oneskorenia  $T(s, p)_{comp}$  nasledovne

$$T(s, p)_{comp} = IPC_{real} t_c = \left( \frac{n(4n^2 + 9n - 7)}{6} \right) t_c$$

Z tohto výrazu pri danom  $T(s, p)_{comp}$  vieme priemernú hodnotu oneskorenia inštrukcie  $t_c$  vypočítať. Pre tento účel sme sme na rôznych pracovných staniciach typu SMP merali hodnotu vykonávania  $T(s, p)_{comp}$  pre rozličné hodnoty veľkosti problému (parameter  $n$ ) na nasledovných paralelných počítačoch typu SMP

- P1 - Intel Core 2 Duo T 7400 (dvojjadrový,  $f=2,16$  GHz)
- P2 - Intel Core 2 Quad (štvorjadrový, 2,5 GHz)
- P3 - Intel SandyBridge i5 2500S (štvorjadrový,  $f=2,7$  GHz).

Porovnanie výkonnosti testovaných pracovných staníc je na obr. 8.



**Obr. 8. Porovnanie výkonnosti pracovných staníc.**

Z nameraných hodnôt najskôr získame pre každú nameranú hodnotu  $T(s, p)_{comp}$  hodnotu hľadaného parametra  $t_{cj}$  ( $j=1, 2, \dots, 11$ ) ako

$$t_{cj} = \frac{T(s,p)_{comp}}{IPC_{real}} \text{ pre } j=15, 18, \dots, 45.$$

Z takto získaných hodnôt parametrov  $t_{cj}$  pre každú nameranú hodnotu  $T(s,p)_{comp}$  vypočítame hľadaný technický parameter  $t_{ci}$  ( $i=1,2,3$ ) pre testované paralelné počítače  $P_i$  ako aritmetický priemer podľa vzťahu

$$t_{ci} = \frac{\sum_{j=1}^{11} t_{cji}}{11}$$

Pre testované počítače získané hodnoty technických parametrov  $t_{ci}$  sú nasledovné

- pre P1  $t_{c1} = 0.678 \mu s$ ,  $t_{s\check{c}}=0.219 \mu s$
- pre P2  $t_{c2} = 0.241 \mu s$ ,  $t_{s\check{c}}=0.078 \mu s$
- pre P3  $t_{c3} = 0.433 \mu s$ ,  $t_{s\check{c}}=0.141 \mu s$ .

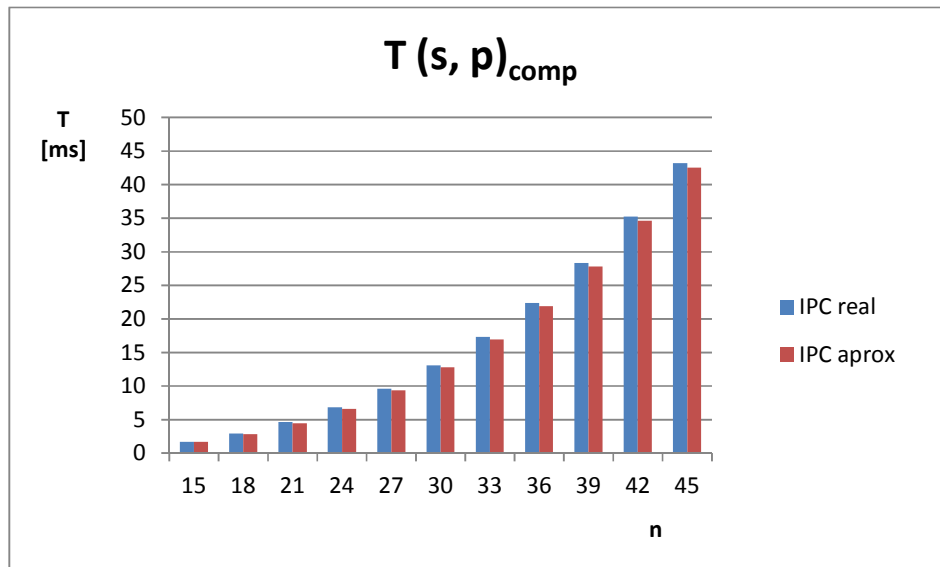
#### 14.4. Aplikačné použitia technického parametra $t_c$

##### 14.4.1. Overenie presnosti aproximačného vzťahu

Pre overenie presnosti odvodeného aproximačného vzťahu pre  $IPC_{aprox.}$  ako

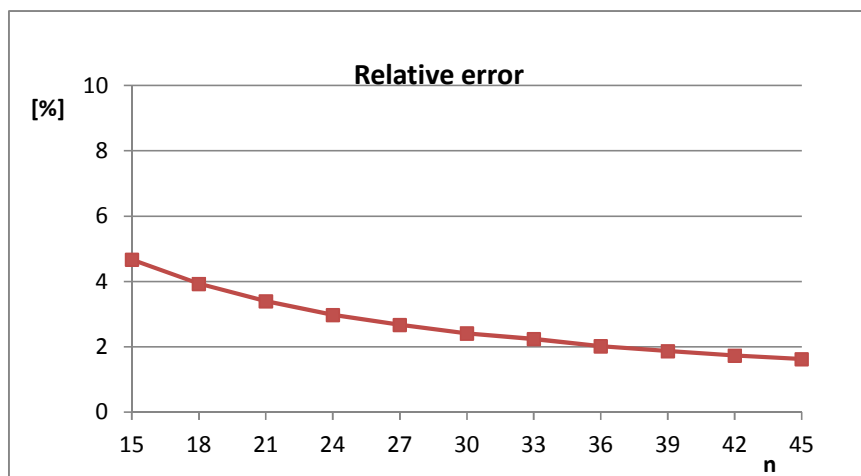
$$IPC_{aprox.} \cong (IPP_{mul} + IPP_{add}) \cong \left( \frac{(2n^3 + 3n^2 - 5n)}{3} \right)_{mul+add}$$

Porovnanie oneskorení výpočtom  $T(s,p)_{comp}$  pre  $IPC_{real}$  a  $IPC_{aprox}$  paralelného počítača Core 2 Duo – P1 ilustruje obr. 9.



Obr. 9. Porovnanie  $T(s,p)_{comp}$  pre  $IPC_{real}$  a  $IPC_{aprox}$ .

Relatívna chyba aproximačného vzťahu pre vybraný testovaný paralelný počítač Intel Core Duo (paralelný počítač P 1) je na obr. 10.



**Obr. 10. Relatívna chyba aproximačného vzťahu.**

Relatívna chyba je v celom porovnanom rozsahu menšia ako päť percent. Súčasne s rastom  $n$  (počet rovníc) relatívna chyba bude klesať ako to plynie z nasledovného postupu. Percentuálny podiel operácií delenia pri rastúcej hodnote  $n$  klesá, pretože zložitosť chyby je daná nasledovným výrazom

$$Error = IPC_{real} - IPC_{aprox} = \frac{n(n+1)}{2} = \frac{n^2 + n}{2}$$

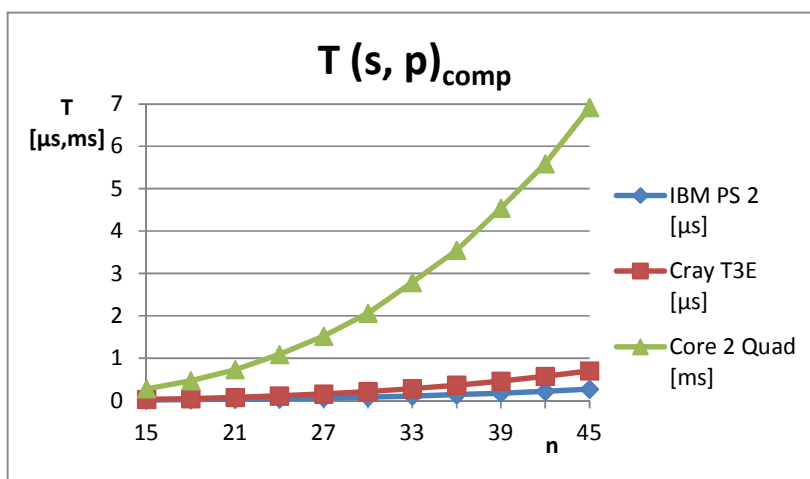
, ktorý má asymptotickú zložitosť  $O(n^2)$  pričom asymptotické zložitosti  $IPC_{real}$  resp.  $IPC_{aprox}$  sú dané ako  $O(n^3)$ , t.j. pre rastúce  $n$  pri  $IPC_{aprox}$  počet zanedbávaných operácií rastie pomalšie ako počet uvažovaných operácií.

#### 14.4.2. Simulované porovnanie výkonnosti paralelných počítačov

Pre porovnanie výkonnosti najrýchlejšieho testovaného paralelného počítača (Intel Core 2 Quad,  $t_c = t_{c2} = 0.241 \mu s$ ) použijeme zástupcov klasických paralelných počítačov vo svete s hodnotami ich technických parametrov  $t_c$ ,  $t_s$ ,  $t_w$ , a to

- paralelný počítač typu NOW IBM SP2 (Scalable POWER parallel System), s technickými parametrami  $t_c = 4,2 \text{ ns}$ ,  $t_s = 35 \mu s$ ,  $t_w = 230 \text{ ns}$ .
- klasický masívny paralelný počítač typu superpočítač Cray T3E s technickými parametrami  $t_c = 11 \text{ ns}$ ,  $t_s = 3 \mu s$ ,  $t_w = 0,063 \mu s$ .

Porovnanie simulovaných oneskorení výpočtom  $T(s,p)_{comp}$  obr. 11.



**Obr. 11. Porovnanie výpočtovej výkonnosti paralelných počítačov.**

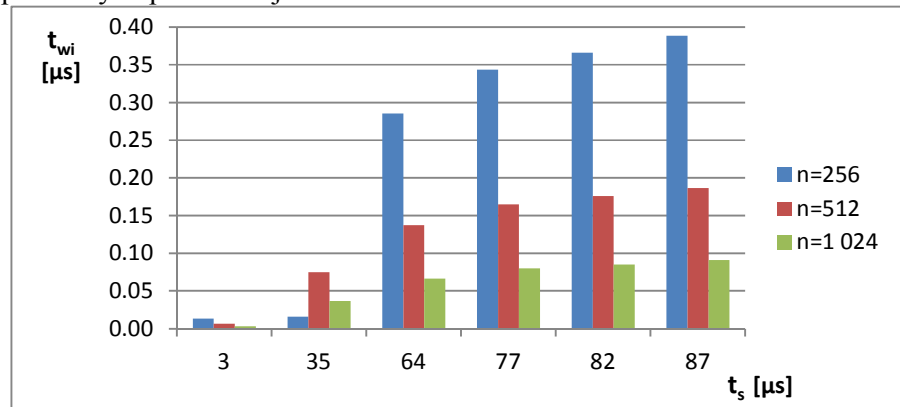
## 14.2.4. Komunikačná zložitosť a komunikačné parametre

### 14.2.4.1. Klasické paralelné počítače

Pre aplikácie MPA je typické priradenie  $n = p$ . V prípade rozsiahlych matic, v ktorých  $n > p$ , výpočtové uzly vykonávajú opakovane činnosti pre zostávajúce  $n > p$  pásy. Ak pre jednoduchosť predkladáme, že hodnota parametra  $n$  je deliteľná parametrom  $p$  bezo zvyšku, výpočtové uzly budú opakovane vykonávať činnosti až do vyčerpania hodnoty podielu  $k=n/p$ . Pre zložitosť (výpočtová, komunikačná) to znamená  $k$  – násobok zložitosti pre  $n = p$ , t. j. základom stanovenia výpočtovej resp. komunikačnej zložitosti je ich stanovenie pre  $n = p$ . Po dosadení tejto rovnosti do vzťahu pre hodnotu parametra  $t_s$  po úpravách dostaneme

$$t_s > t_w (n - 2\sqrt{n})$$

Grafická ilustrácia pre optimalizáciu voľby dekompozičnej metódy celého spektra použitých paralelných počítačov je na obr. 12.



Obr. 12. Voľba dekompozičnej metódy pre menšie hodnoty  $t_s$ .

Hraničné hodnoty voľby optimálnej dekompozičnej stratégie sú pri danom  $n$  pre väčšie hodnoty  $t_w$  väčšie. Tým dekompozícia na pásy je efektívnejšia pre vyššie hodnoty  $t_s$  (NOW, Grid) a dekompozícia na bloky zase pre menšie hodnoty  $t_s$ .

### 14.2.4.2. Paralelné počítače typu NOW

#### 14.3.2.1. Kolektívne komunikačné mechanizmy

Pre typické nasledovné komunikačné mechanizmy MPI na sieti Ethernet platia nasledovné vzťahy

- príkaz rozdelenia dát (rozptyl) typu Broadcast.
- príkaz rozdelenia dát (rozptyl) typu Scatter
- príkaz zberu dát typu Gather
- príkaz zberu dát typu Allgather
- príkaz zberu a sumarizácie dát typu Reduce

#### 14.3.2.1. Kolektívny komunikačný mechanizmus typu Broadcast

Kolektívny komunikačný mechanizmus Broadcast je jediný kolektívny komunikačný mechanizmus, ktorý môže byť efektívny na komunikačnej sieti Ethernet. Komunikačná zložitosť  $T(s,p)_{commbr}$  tohto príkazu je  $O(1)$  a s vyjadrením komunikačných parametrov  $t_s$ ,  $t_w$  ako

$$T(s, p)_{commbr} = t_s + t_w$$

Pre prenos  $m$  rôznych dátových jednotiek vždy iba jednému procesoru výpočtová zložitosť bude daná ako  $O(m)$  a s podporou komunikačných parametrov nasledovným vzťahom

$$T(s, p)_{commbr} = m (t_s + t_w)$$

Pre prenos rôznych  $m$  dátových jednotiek  $p-1$  zostávajúcim procesorom výpočtová zložitosť bude daná ako  $O(p)$  a  $s$  podporou zavedených komunikačných parametrov ako

$$T(s, p)_{commbr} = \sum_{i=1}^{p-1} m(t_s + t_w)$$

Rovnaké komunikačné oneskorenie bude pre všetky ostatné kolektívne komunikačné mechanizmy na sieti Ethernet, a to ako

$$T(s, p)_{commEth} = \sum_{i=1}^{p-1} m(t_s + t_w) = (p-1) m(t_s + t_w)$$

## 15. Prínosy

### 15.1. Teoretické

- rozšírenie teórie zložitosti pre maticové paralelné algoritmy (MPA)
  - oneskorenie výpočtom  $T(s, p)_{comp}$
  - oneskorenie medziprocesorovou komunikáciou  $T(s, p)_{comm}$ 
    - ✓ dekompozičný model rozkladu na bloky
    - ✓ dekompozičný model rozkladu na pásy (riadky, stĺpce)
  - zložitosť zdrojov (výpočtové uzly PP)
- odvodenie vzťahov pre výkonnosť MPA
  - dekompozičný model rozkladu na bloky
  - dekompozičný model rozkladu na pásy (riadky, stĺpce)
- odvodenie funkcií izoeftívnosti pre predikciu zložitosti (výkonnosti) MPA
  - dekompozičný model rozkladu na bloky
  - dekompozičný model rozkladu na pásy (riadky, stĺpce)
  - spoločná funkcia izoeftívnosti  $w(s)_{opt}$
- návrh, vývoj, optimalizácia a implementácie MPA použitím
  - spoločná pamäť (štandard OpenMP – threads)
  - distribuovaná pamäť (štandard MPI)
- optimalizácia MPA
  - výber dekompozičného modelu
  - minimalizácia komunikačných oneskorení  $T(s, p)_{comm}$
  - analýza oneskorení MPA (režijná funkcia)
    - so spoločnou pamäťou
    - distribuovaná pamäť
      - ◆ NOW
      - ◆ sieť sietí NOW (Grid)
- odvodenie analytických výrazov výkonnosti MPA s režijnými funkciami
  - optimalizácia výberu dekompozičného modelu
  - paralelný výpočet, zrýchlenie, efektívnosť, izoeftívnosť
  - predikcia zložitosti (výkonnosti), škálovateľnosť.
- metodika priamych meraní výkonnostných parametrov MPA
  - zrýchlenie
  - efektívnosť
  - izoeftívnosť
- metodika stanovenia technických parametrov PP.

### 15.2. Praktické

- metodika priamych meraní výkonnosti MPA a jej overenie
  - jeden proces (sekvenčný, paralelný)



- paralelné procesy pre SMP (spoločná pamäť)
- paralelné procesy pre NOW a Grid (distribučovaná pamäť)
- vývoj MPA a ich alternatív
  - spoločná pamäť
  - distribuovaná pamäť
- aplikačné použitie štandardizovaných paralelných vývojových prostredí
  - OpenMP Threads
  - MPI
- overenia výsledkov teoretickej časti práce
  - vplyv dempozíčného modelu
  - výkonnostné parametre MPA
  - technické parametre PP
- simulované výpočty oneskorenia výpočtom  $T(s,p)_{comp}$  pre PP
  - viacprocesorové a viacjadrové typu SMP
  - siet NOW
  - superpočítač T3DE.
- aplikačné využívanie paralelných počítačov – masívne, SMP, NOW, Grid.
- kolektívne komunikačné mechanizmy v sieti Ethernet
  - broadcast
  - scatter
  - gather, allgather
  - reduce.

## Záver a perspektívy

Rozvoj masívnych paralelných počítačov vo svete v ich hlavnom období rozvoja (do roku 2000) bol vedený snahou o zvýšenie výkonnosti paralelnou organizáciou pri použití v tom čase dostupných procesorov. Tieto masívne paralelné počítače v súčasnosti pre ich odlišenie označujeme ako klasické paralelné počítače. Jedná sa predovšetkým o vo svete rozšírené a dostupné masívne paralelné počítače typu superpočítač, škálovateľné RISC architektúry a pod. I v súčasnosti sú dostupné ich inovované verzie na báze výkonných RISC procesorov ale tieto z dôvodu ich vysokej ceny nepatria medzi dominantné aplikačné trendy.

Dominujúce vývojové trendy vo svete smerujú k nahradzovaniu týchto veľkých počítačov (mainframes, superpočítače) paralelnou organizáciou prepojených pracovných staníc na báze vysokovýkonných štandardne dostupných procesorov resp. počítačov (jednoprocessorové, viacprocesorové, viacjadrové). Dynamický rast i u nás zaznamenávajú paralelné počítače typu SMP (viacprocesorové, viacjadrové), siete osobných počítačov NOW a ich vyššie mohutné integrované formy typu Grid (sieť sietí NOW). Práve tieto v súčasnosti dominujúce architektúry paralelných počítačov typu SMP, NOW, Grid sú vo svojej podstate prepracované asynchrónne paralelné systémy typu SIMD a MIMD. Počet použitých procesorov pre architektúru typu SMP nebol doteraz vysoký a bol limitovaný najmä spôsobom riešenia komunikácie pri používaní spoločných zdrojov (pamäť, V/V zariadenia a pod.). Nové smery v oblasti SMP zahŕňujú masívnejšie viacjadrové procesory (multicore) a perspektívne i mohutné paralelné systémy s podstatne väčším počtom jadier resp. procesorov (masívne SMP). Sprievodné znaky takýchto paralelných počítačov sú nasledovné

- eliminácia rozdielov medzi vysoko výkonnými paralelnými počítačmi typu HPC a distribuovanými paralelnými počítačmi
- dôraz na zvyšovanie výkonnosti použitých komunikačných sietí (zvyšovanie prenosovej rýchlosti a počtu paralelných komunikačných kanálov)
- integrácia použitia rozšírených štandardov (OpenMP, MPI) pre vývoj paralelných algoritmov, ich optimalizáciu a efektívnu implementáciu

- intenzívna paralelná podpora pre mohutnejšie viacprocesorové SMP systémy v rámci jednej pracovnej stanice.

Tieto trendy sa už v súčasnosti premietajú do štruktúry operačných systémov (Unix, Windows Server a pod.). Paralelné počítače typu NOW získavajú vlastnosti univerzálneho paralelného počítača, pretože obsahujú pracovné stanice na báze výkonných procesorov so SIMD inštrukciami a viacprocesorové systémy typu SMP (pracovná stanica typu SMP). Pre aplikačnú oblasť je preto dôležité rozširovanie transparentného a aplikačné používanie spoločného výpočtového modelu typu MPMD (Multiple processes, Multiple data) pre univerzálny virtuálny paralelný počítač. Tým sa otvárajú možnosti aplikačného využívania výkonného nedeterministického počítača pre efektívne riešenie zložitých numerických a zložitých optimalizačných a kombinatorických úloh ako i predpokladaný posun v riešení úloh typu NP (Non polynomial) paralelným prístupom.

Predložená DDP je súčasťou riešeného vedeckého projektu na FRI (školiť Prof. Ing. Ivan Hanuliak, PhD.) pre modelovanie, optimalizáciu a predpovedanie výkonnosti paralelných počítačov (masívne paralelné architektúry vo svete, SMP, NOW, Grid) a paralelných algoritmov (spoločná pamäť, distribuovaná pamäť). Získané teoretické poznatky a praktické skúsenosti by som preto chcel i v budúcnosti aplikovať vo vytváraných virtuálnych laboratóriách vysokovýkonného aplikačného zamerania v svojom pracovnom prostredí, a to v spolupráci s budovaným strediskom HPC ŽU a HPC FRI. Dostupnosť univerzálneho paralelného počítača na báze prepojenia kombinácií SMP, NOW a Grid (HPC ŽU a Grid FRI) vytvára reálne možnosti prispieť vyvinutými modelmi a stratégiami k rozšíreniu metodiky riešenia zložitých aplikačných úloh efektívnymi (optimalizovanými) paralelnými algoritmi i do iných aplikačných oblastí, ktoré vyžadujú efektívne riešenia zložitých aplikačných úloh výkonnými paralelnými algoritmi.

## Abstract

The theses “Performance modelling of parallel algorithms“deals with a problematic of complexity and performance evaluation of matrix parallel algorithms (MPA). The main goals were to perform

- analysis of the development trends in the world, their actual state and perspectives in the areas parallel architectures, parallel algorithms, criteria for performance evaluation and computing parallel models
- mathematically supported formulations of complex performance modelling of MPA including their most important components (computation, communication control and synchronisation)
- theoretical and practical analysis on all used forms of parallel computers (NOW, SMP and Grid) of MPA including
  - decomposition models (blocks, strips)
  - analytical performance relation
    - ◆ computation latency
    - ◆ speed-up
    - ◆ effectivity
  - performance prediction
    - ◆ isoefficiency
    - ◆ optimisation
- methodology of suggestion and its verification to measure MPA of
  - single process
  - parallel proces in a shared memory
  - parallel proces in a distributed memory.

For real communication models we have been demonstrated the role of decomposition model on the practical examples of parallel multiplication and the way of deriving technical

parameters of Gauss elimination method (GEM). For these purposes we have developed following parallel algorithms

- parallel multiplication
  - standard decomposition model
  - alternative decomposition model
- GEM
  - shared memory (OpenMP)
  - distributed memory (MPI).

For these parallel algorithms we have derived and verified analytical equations on various practical simulation experiments. The achieved and verified results could be methodically applied to other matrix parallel algorithms for actually dominant parallel computers based on SMP, NOW and Grid.

**Key-words:** parallel computer, parallel algorithm, performance evaluation methods, decomposition model, communication overheads, performance modelling, speed-up, efficiency, isoefficiency, NOW, Grid, SMP, scalability, shared memory, distributed memory, Gauss elimination method, OpenMP, MPI.

## Literatúra

1. Allen M., Wilkinson B., Parallel Programming – Techniques and Applications Using Parallel Computers, Prentice Hall, 431 pp., 1999
2. Andrews G. R., Foundations of Multithreaded, Parallel, and Distributed Programming, Addison Wesley, 664 pp., 2000
3. Bader D. A., Petascale Computing: Algorithms and applications, CRC Press, 2007
4. Bahi J. H., Contasst-Vivier S., Couturier R., Paralel iterative algorithms: From Segquential to Grid Computing, CRC Press, 2007
5. Bal E. H., Parallel Programmeren, Wrije Universiteit, Amsterdam, 2002, 551 pp.
6. Baugh J. W., Konduri R. K., S., Discrete element modeling on a cluster of workstations, Enginneering with computers, pp. 1-15, 2001
7. Berman F., Geoffrey F., Hey T., Grid Computing: Making the Global Infrastructure a Reality, 1060 pp., 2003, John Wiley&Sons
8. Burns M. V., George A. D., Wallace B. A., Modeling and simulative perf. analysis of SMP and clusters, Simulation, Vol. 74, pp. 74-92, 2000
9. Carpinelli J., Computer Systems Organization and Architecture, Prentice Hall, 584 pp., 2001
10. Casanova H., Legrand A., Robert Y., Paralel Algorithms, CRC Press, 2008
11. Cohen A., Numerical analysis of wawelet methods, JAI Press, 354 pp., 2003
12. Comer D. E., Computer networks and Internet with Internet Applications IPE (3<sup>rd</sup> Edition), 720 pp., Prentice Hall, 2001, United Kingdom
13. Coulouris G., Dollimore J., Kindberg T., Distributed Systems – Concepts and Design (Third Edition), Addison Wesley, 800 pp., 2001, United Kingdom
14. Dally W., Towles B., Principles and Practises of Interconnection Networks, 550 pp., 2004, Morgan Kaufmann publishers
15. Dasgupta S., Papadimitriou Ch.H., Vazirani U., Algorithms, McGraw-Hill, 336 pp., 2006
16. Davis T. A., Direct methods for sparse Linear Systems, 184 pages, 2006, Cambridge University Press, United Kingdom
17. Dennis A., NET Multithreading, Manning Publications Co., 360 pp., 2002
18. Devlin, K., Problémy pro třetí tisíciletí (preklad), 269 strán, Praha 2005
19. Dvořák V., Architektúra a programování paralelních systému, VUTIUM Brno, 165 strán, 2004, Brno, Česká republika
20. Duato J., Yalamanchili S., Lionel N., Interconnection networks, Morgan Kaufmann, 624 pp., 2002

21. Edmonds J., How to think about algorithms, Cambridge University Press, 472 pages, 2010
22. Eldén L., Matrix Methods in Data Mining and Pattern Recognition, 184 pages, 2007, Cambridge University Press, United Kingdom
23. Fortier P., Howard M., Computer Systems Performance Evaluation and Prediction, 544 pp., 2003, Digital Press
24. Foster I., Kesselman C., The Grid 2, - Blueprint for a New Computing Infrastructure (Second Edition), Morgan Kaufmann, 748 pp., 2003, USA
25. Gelenbe E., Analysis and synthesis of computer systems, 324 pages, published April 2010, Imperial College Press
26. Gelenbe E., Computer system performance modeling in perspective, 288 pages, published September 2006, Imperial College Press
27. Goldreich O., Computational complexity, Cambridge University Press, 632 pages, 2010
28. Goodrich M., Tamassia R., Algorithm Design: Analysis, and Internet Examples, 2002, John Wiley&Sons
29. Hager G., Wellein G., Introduction to High Performance Computing for Scientists and Enginners, 356 pages. July 2010, CRC Press
30. Hanuliak I., Parallel architectures - multiprocessors, computer networks (in Slovak), 187 pp., July 1997, Publ.: Book center, Žilina, Slovakia
31. Hanuliak I., Paralelné počítače a algoritmy, 327 pp., Vyd.: ELFA Košice, 1999
32. Hanuliak P., Hanuliak I., Performance evaluation of iterative parallel algorithms, Kybernetes, Volume 39, No.1, 2010, pp. 107- 126, United Kingdom
33. Hanuliak P., Analytical method of performance prediction in paralel algorithms, The Open Cybernetics and Systemics Journal, reviewed (in print)
34. Hanuliak M., Modelovanie a optimalizácia výkonnosti dátových prenosových sietí, 116 strán, Fakulta riadenia a informatiky, Žilinská univerzita, 2008, Žilina
35. Hanuliak J., Modelovanie a predikcia výkonnosti DPA, máj 2006, 109 pp., FRI, Žilina, 2006
36. Hanuliak P., Modelovanie, optimalizácia a predikcia výkonnosti iteračných paralelných algoritmov, April 2007, pp. 112, Žilinská univerzita, Žilina
37. Hanuliak P., Virtual parallel computer, In Proc.: TRANSCOM 2007, 2007, Žilina
38. Hanuliak M., Performance modelling of computer systems, In Proc. ICSC 2013, Kunovice, Czech republic, accepted, in print
39. Hanuliak M., Hanuliak J., Analytical modelling of distributed computer systems, NOW, In Proc.: TRANSCOM 2005, pp. 103-10, 2005, Žilina, Slovak Republic
40. Heath M. T., Scientific Computing, McGraw-Hill Publishers, 576 pp., 2002
41. Heuring Vincent P., Jordan Harry I., Computer System Design and Architecture, New Jersey: Pearson Prentice Hall, 2004
42. Hillston J., A Compositional Approach to Performance Modelling, University of Edinburg, 172 pages, 2005, Cambridge University Press, United Kingdom
43. Hughes C., Hughes T., Parallel and Distributed Programming Using C++, 720 pp., 2004, Addison Wesley
44. Hudik M., Hanuliak P., Analysing Performance of Parallel Algorithms for Linear System Equations, In Proc.: GCCP 2011, pp. 70-77, 2011, UI SAV , Bratislava
45. Hudik M., Hanuliak P., Parallel complexity of linear system equation, In Proc.: TRANSCOM 2011 - section 3, pp. 107-110, 2011, Žilina
46. Chandra R., Menon R., Dagum L., Kohr D., Maydan D., McDonald J., Parallel Programming in OpenMP, 231 pp., 2001, Morgan Kaufmann Publishers
47. Chapra S. C., Canale R., Numerical Methods for Enginners with Software Programming Applications, 960 pp., 2006, McGraw-Hill
48. Janovič F., Hanuliak I., Programovacie techniky, stran 163, EPI Kunovice, 2011
49. Janovič F., Hanuliak P., To prediction of performance in paralel algorithms, AD ALTA, Vol. 2, issue 1, pp. 81-85, 2012, Magnanimitas Assn., The Czech republic

50. Janovič F., Holúbek A., Load balance and data decomposition for distributed prime number algorithm, In Proc.: TRANSCOM 2011, pp. 99 - 101, 2011, Žilina
51. Janovič F., Performance modelling of distributed paralel algorithm, ICSC Leden 2012, pp. 57-64, Kunovice s. r. o., Czech republic
52. Janovič F., Slováček D., Programming methods, ICSC 2011, sekcia č. 2, pp 137 - 141, Kunovice, Czech republic
53. Janovič F., Hanuliak P., Optimisation of decomposition strategy in paralel matrix algorithms, In Proc. ICTIC, pp. 19-23 March 2012, Zilina, Slovak republic
54. Janovič F., Bartoněk D., Slováček D., Teoretické základy informatiky, EPI Kunovice 2011, stran 178, ISBN: 978-80-7314-234-6, 13 AH, podiel 4 AH
55. John L. K., Eeckhout L., Performance evaluation and benchmarking, CRC Press, 2005
56. Jordan H., Alaghand G., Fundamentals of parallel processing, 536 pp., 2002, Pearson Education
57. Kenneth H. Rosen et all., Handbook of discrete and combinatorial mathematics, CRC Press, pp. 1232, 2000
58. Kenyon T., High performance data network design, Digital Press, 480 pp., 2002
59. Kenyon T., Data networks – routing, security and performance optimisation, Digital Press, 806 pp., 2002
60. Kirk D. B., Hwu W. W., Programming massively parallel processors, Morgam Kaufmann, 280 pages, 2010
61. Kostin A., Ilushechkina L., Modeling and simulation od distributed systems, 440 pages, Jun 2010, Imperial College Press.
62. Kshemkalyani A. D., Singhal M., Distributed Computing, University of Illinois, 756 pages, 2011, Cambridge University Press, United Kingdom
63. Kumar V., Grama A., Gupta A., Karypis G., Introduction to parallel computing (Second Edition), 636 pp., Addison Wesley, 2003, Netherlands
64. Kurose J. F., Ross K. W., Computer Networking: A Top – Down Approach Featuring the Internet, Addison Wesley, 712 pp., 2001, USA
65. Kumar A., Manjunath D., Kuri J., Communication Networking – An Analytical Approach, 960 pp., 2004, Morgan Kaufmann Publishers
66. Lilja D. J., Measuring Computer Performance, 280 pages, 2005, University of Minnesota, Cambridge University Press, United Kingdom
67. Liu M. L., Distributed Computing: Principles and Applications, 448 pp., 2004, Addison-Wesley Publishers
68. Lekass P., Network Processors, 456 pp., 2003, McGraw-Hill Publishers
69. Magoules F., Pan J., Tan Kiat-An., Kumar A., Introduction to Grid Computing, CRC Press, 2008
70. Magoules F., Nguyen T. M. H., Yu L., Grid Ressource Management: Towards Virtual and services Complaint grid Computing, CRC Press, 336 pp., 2008
71. Matiaško K. a kol., Základy informatiky, 394 pp., 2004, EDIS ŽU, Žilina
72. Mc Cabe J., Network Analysis, Architecture and Design, 450 pp., 2003, Morgan Kaufmann Publishers
73. Mertens S., Schinner A., Cluster Computing, Springer Verlag, 300 pp., 2002
74. Mitscheile-Thiel A., System Engineering with SDL: Developing Performance-Critical Communication Systems, 380 pp., 2001, John Wiley&Sons
75. Nemeth Z., Sunderam V., Characterizing Grids: Attributes, Definitions, and Formalisms, Journal of Grid computing, Number 1, pp. 9 – 23, 2003
76. Pan L. and col., Distributed parallel computing using navigational programming, Int. J. of parallel programming. Vol. 32, No. 1, pp. 1-36, 2004,
77. Pacheco P., An Introduction to parallel computing, Morgan Kaufmann, 2011
78. Paterson D. A., Hennessy J. L., Computer Organisation and Design, 4 – th Edition, 912 pp., Morgan Kaufmann, 2009

79. Peterson L., Davie B., Computer Networks – A System Approach (3rd edition), 813 pp., 2003, Morgan Kaufmann Publishers
80. Pióro M., Routing, Flow, and Capacity Design in Communication and Computer Networks, 400 pp., 2004, Morgan Kaufmann Publishers
81. Plaszczak P., Wellner R., Grid computing, Morgan Kaufmann, 288 pp., 2005
82. Powers D., Boundary value problems and partial differential equations, Elsevier, 520 pp., 2005
83. Rajasekaran S., Reif J., Handbook of Paralel Computing: Models, Algorithms and Applications, 2007
84. Ramaswami, B. J., Sivarajan, K., Sasaki, G., Optical networks (3rd Edition) - A practical perspective, 928 pp., 2010, Morgan Kaufmann, USA
85. Quinn M. J., Parallel Programming in C with MPI and Open MP (First Edition), 544 pp., McGraw-Hill Publishers, 2004,
86. Shan H and col., Message passing and shared address space parallelism on an SMP cluster, Parallel computing, Volume 29, Issue 2, pp. 167-186, 2003
87. Shroff G., Enterprice Cloud Computing, 290 pages, 2010, Cambridge University Press, United Kingdom
88. Schubert G. and. col., Hybrid- paralel sparse matrix-vector multiplication with explicit communication overlap on current multicore-based systems, Parallel Processing Letters, pp. 339-358, Vol. 21, No. 3, 2011
89. Sodan A. C., First G., Application on a multithreaded architecture, Parallel computing, Vol. 28, pp. 3-33, 2002
90. Sokolovski P. J., Grosu D., Performance of the NAS parallel benchmarks on Grid enabled clusters, In Proc. NCA'04), 2004
91. Stallings W., Computer Organisation and Architecture – Designing for Performance (Fifth Edition), Prentice Hall, 815 pp., 2003, UK ŽU - knižnica
92. Stallings W., High-Speed Networks and Internets – Performance and Quality of ervice (2nd Edition), Prentice Hall, 550 pp., 2002
93. Tannenbaum A. S., Distributed Systems – Principles and Paradigms, Prentice Hall, 840 pp., 2002
94. Tannenbaum A. S., Computer Networks (Fourth Edition), Prentice Hall, 848 pp., 2003, The Netherlands
95. Vlassov V., Ayani R., Analytical modeling of multithreaded architectures, Journal of system architecture, Vol. 46, pp. 1205-1230, 2000
96. Vaniček, J., Papík, M., Pergl, R. a Vaniček T., Teoretické základy informatiky, 431 stran, Praha, Kernberg Publishing, 2007
97. Wang L., Jie Wei., Chen J., Grid Computing: Infrastructure, Service, and Application. CRC Press, 2009
98. Williams R., Computer System Architecture – A Networking Approach, 680 pp., 2001

**www stránky**

99. [www.top500.org](http://www.top500.org)
100. [www.spec.org](http://www.spec.org)
101. [www.en.wikipedia.org](http://www.en.wikipedia.org)
102. [www.intel.com](http://www.intel.com)
103. [nic.uniza.sk/hpc](http://nic.uniza.sk/hpc)

## **Publikačná činnosť autora**

### **Vysokoškolské učebnice**

- [1] Janovič F., Bartoněk D., Slováček D., Teoretické základy informatiky, EPI Kunovice 2011, stran 178, ISBN: 978-80-7314-234-6
- [2] Janovič F., Slováček D., Operační systémy, stran 159 EPI Kunovice 2011, 12 AH
- [3] Janovič F., Hanuliak I., Programovacie techniky, stran 163, EPI Kunovice, 2011
- [4] Janovič F., Rukovanský I., Základy podnikových informačních systému a informačních systému státní správy a samosprávy, EPI Kunovice 2011, stran 161, 12 AH

### **V zahraničí**

- [5] Janovič F., Hanuliak P., To prediction of performance in paralel algorithms, AD ALTA, Volume 2, issue 1, pp. 81-85, 2012, Magnanimitas Assn.
- [6] Janovič F., Slováček D., Parallel Computing Networks Managed Linux, Collection of scientific articles, issue 7, pp. 195 – 205, Bucovinian univerzity, 2011, ISSN 2219 - 5378
- [7] Janovič F., Hanuliak P., To prediction of performance in parallel algorithms, In Proc.: Quare 2012, 975-984, Magnanimitas, 2012 Hradec Králové, ISBN 978-80-905243-0-9
- [8] Janovič F., Slováček D., Kodrila P., Calculation program for machine working time, Comp. Sc. in the Age of XXI, Polit. Radom, pp. 111- 122, 2011, ISBN 978-83-7789-006-6
- [9] Janovič F., Performance modelling of distributed parallel algorithm, ICSC Leden 2012, pp. 57-64, Kunovice s. r. o., Czech republic, ISBN 978-80-7314-279-7
- [10] Janovič F., Slováček D., Programming methods, ICSC 2011, sekcia č. 2, pp 137 - 141, Kunovice, Czech republic, ISBN 978-80-7314-221-6
- [11] Janovič F., Slováček D., Efficiency Network Regulated By Linux. In ICSC 2010. pp. 35 – 41, EPI Kunovice s.r.o, 2010, ISBN 978-80-7314-201-8.

### **Doma**

- [12] Janovič F., Hanuliak P., Optimisation of decomposition strategy in parallel matrix algorithms, In Proc. ICTIC, pp. 19-23 March 2012, Zilina, ISBN 978-80-554-0513-1
- [13] Janovič F., Holúbek A., Load balance and data decomposition for distributed prime number algorithm, In Proc.: TRANSCOM 2011, pp. 99 - 101, 2011, Žilina, ISBN 978-80-554- 0372-4.

### **Výskumné projekty (spoluriešiteľ)**

- [1] Národný grantový projekt MŠ SR (Projekt SIVP 2009), Slovenská infraštruktúra pre vysokovýkonné počítanie, spoločný projekt akademických pracovísk a vybraných slovenských univerzít, 2009
- [2] Efektívne paralelné algoritmy zložitých úloh, Projekt VEGA 1/0050/09, 2009-2010
- [3] Modelovanie a optimalizácia výkonnosti paralelných algoritmov, Projekt VEGA 1/0017/11, 2011-2013.