

Projekt „Slovenská infraštruktúra pre vysokovýkonné počítanie“ a paralelné výpočty

Milan Šujanský

Technická univerzita v Košiciach
Letná 9

Košice, Slovenská republika
e-mail: Milan.Sujansky@tuke.sk

Abstrakt

Cieľom príspevku je podať informáciu o produktoch projektu "Slovenská infraštruktúra pre vysokovýkonné počítanie" (SIVVP) s ukončením v 2015 roku. Projekt je riešený v rámci štrukturálnych fondov Európskej únie. V druhej časti sú uvedené niektoré aplikácie softvéru MATLAB s využitím paralelných výpočtov. Záver príspevku popisuje postupy ďalšieho využívania produktov projektu SIVVP

KLúčové slová: Projekt SIVVP, superpočítače, softvér MATLAB, paralelné výpočty.

JEL klasifikácia: C63

1. Projekt „Slovenská infraštruktúra pre vysokovýkonné počítanie“ (SIVVP)

1.1 Technické údaje o projekte SIVVP

Projekt SIVVP je hradený zo štrukturálnych fondov s dobou riešenia od 1. 1. 2010 do 31. 12. 2015. Pozostáva z dvoch častí:

Cieľ: Konvergencia

Kód výzvy: OPVaV/K/RKZ/NP/2009-1

Číslo zmluvy: OPVaV/NP/1/2010

ITMS kód projektu: 26210120002

Celkové oprávnené výdavky: 13 280 750,00 EUR

Partneri projektu sú: Ústav experimentálnej fyziky SAV v Košiciach, Technická univerzita v Košiciach (TUKE), Žilinská univerzita v Žiline, Univerzita Mateja Bela v Banskej Bystrici.

Cieľ: Regionálna konkurencieschopnosť a zamestnanosť

Kód výzvy: OPVaV/K/RKZ/NP/2009-1

Číslo zmluvy: OPVaV/NP/2/2010

ITMS kód projektu: 26230120002

Celkové oprávnené výdavky: 12 684 250,00 EUR

Zodpovedné riešiteľské pracovisko: Výpočtové stredisko SAV v Bratislave. Partneri projektu sú: Ústav informatiky SAV v Bratislave, Slovenská technická univerzita v Bratislave.

Spolu celkové oprávnené výdavky: 25 965 000,00 EUR

Rozpočet pre TUKE: 1 371 780,00 EUR, z toho rozpočet na softvér: 540 000,00 EUR. Z uvedeného vyplýva, že projekt pokrýva investičné výdavky s hlavným zameraním na hardvér a softvér.

S projektom súvisia tiež aktivity Centra vedeckotechnických informácií v Bratislave, ktoré zabezpečuje štatistické výpočty a zo systému Matlab výpočty pre biológiu.

Podmienky vyplývajúce z opisu projektu sú nasledovné: Hardvér a softvér bude použitý na nekomerčné použitie s možnosťou publikovania výsledkov výpočtov vo vedeckých publikáciách. Zakúpené softvérové licencie budú trvalého charakteru a budú zahŕňať ročnú

bezplatnú podporu a v rámci tejto podpory bude možnosť aktualizácie (upgrade) nových verzií programov. Zakúpený hardvér a softvér bude k dispozícii všetkým organizáciám výskumu a vývoja ako aj vysokým školám na celom Slovensku (SAV, VŠ, univerzity).

O riešení projektu SIVVP boli pravidelne podávané informácie na konferenciách UNINFOS, SIT 2014, naposledy [1].

Zakúpený hardvér v rámci projektu možno rozdeliť na dve skupiny: superpočítače – VS SAV a klastre – ostatní partneri. Podrobnejšie údaje sa nachádzajú na www.sivvp.sk. V ďalšom bude venovaná pozornosť výsledkom riešenia projektu SIVVP u partnera TUKE.

1.2 Softvérové produkty zakúpené na TUKE

Systémový softvér a vývojové nástroje: MOAB, VMware vCloud Standard Suite, Intel Parallel Studio XE Composer Edition 2015, PGI Accelerator Fortran/C/C++ s podporou pre použitie GPU akcelerátorov, OpenMPI - A High Performance Message Passing Library, PETSc - Portable, Extensible Toolkit for Scientific Computation a ďalšie nástroje dostupné v rámci operačného systému Scientific Linux 6.

Aplikačný softvér: (Welding solution (SYSWELD, zakúpila STU) - zvaranie, tep. spracovanie, PAM-STAMP - tvárnenie, ProCAST - zlievanie (zakúpilo VS SAV), ABAQUS, Matlab, ANSYS, EPLAN, (Mentor Graphics mimo projektu TUKE).

Okrem uvedených softvérov je možnosť použitia softvérových prostriedkov vlastných, komerčných a voľných.

Pre pracovníkov TUKE v zmysle opisu projektu SIVVP sú k dispozícii aj produkty zakúpené inými partnermi projektu SIVVP.

1.3 Spôsoby realizácie výpočtov na TUKE

Hardvérové prostriedky tvoria dve vetvy: HPC klaster a CLOUD.

Vetva HPC klaster

Vetva HPC klaster je predovšetkým určená na spracovanie výpočtovo náročných vedeckotechnických výpočtov hlavne simulačného charakteru v dávkovom režime, čo znamená, že jednotlivé výpočtové úlohy sa radia do frontu, z ktorého sú postupne radené do výpočtu, ktorý prebieha samostatne bez interaktívneho zásahu používateľa. Príprava výpočtu a prípadné spracovanie výsledkov sa realizuje na klientskej stanici prípadne využitím možnosti vetvy CLOUD.

Vhodné použitie vetvy HPC je v nasledovných prípadoch:

- výpočet je značne časovo náročný,
- výpočet sa dá rozdeliť na viac samostatných úloh,
- výpočet potrebuje viac operačnej pamäte ako je k dispozícii na pracovnej stanici,
- výpočet sa dá zrýchliť použitím viacerých výpočtových uzlov prípadne použitím akcelerátorov typu GPU.

Výpočet musí byť prenositeľný do prostredia operačného systému Scientific Linux v.6. Pre spustenie úlohy je potrebné pripraviť vstupné súbory, vykonateľný súbor a skript na zavedenie úlohy do dávkového systému.

Vetva CLOUD

Vetva CLOUD je predovšetkým určená na interaktívnu prácu, pri ktorej sa využíva vzdialený prístup z klientskej pracovnej stanice k výpočtovým prostriedkom sprístupneným prostredníctvom CLOUDu. Z uvedeného vyplýva vhodnosť pre časovo menej náročné výpočty. Tiež voľba prostredia daného operačného systému je voliteľná používateľom.

U oboch vetiev je možnosť využitia paralelných výpočtov na viacerých jadrách procesora alebo na viacerých procesoroch.

Porovnanie výpočtov vo vetvách Cloud a HPC je v nasledovnej tabuľke:

	Cloud	HPC
primárne určenie	poskytovanie privátnych virtuálnych serverov	spracovanie paralelných, výpočtovo náročných výpočtov
režim spracovania	Interaktívny	Dávkový
operačný systém	MS Windows, Linux	Scientific Linux 6
správa systému	používateľ	správca HPC klastra
vnútorné prepojenie	10 Gbit Ethernet	Infiniband
Prístup	RDP, VNC, SSH	Secure Shell, VNC

2 Aplikácie softvéru MATLAB s využitím paralelných výpočtov

2.1 Požiadavky na používateľa pri realizácii vysokovýkonného počítania

Veľká väčšina náročných vedeckotechnických výpočtov má charakter simulačných výpočtov. U analytických výpočtov z pôvodného matematického modelu popisujúceho chovanie daného systému na základe matematických postupov dostávame výsledné všeobecné exaktné riešenie. U simulačných výpočtov na číslicovom počítači z pôvodného matematického modelu vytvárame výsledný simulačný model realizovaný na číslicovom počítači, ktorý je funkčný pre konkrétne parametre daného matematického modelu. Dominujúce je tu použitie rôznych numerických postupov, ktoré u dynamických spojitých systémov zároveň predstavujú diskretizáciu riešenia. Výsledkom je jedno partikulárne riešenie (simulačný beh) v diskretnom tvare. Pokiaľ cieľom je nejaký výsledok, napr. optimalizácia parametrov a pod. je potrebné k simulačnému modelu definovať algoritmus simulačného experimentu. Uvedené je možno dokumentovať na nasledovnom jednoduchom príklade riešenom v systéme Matlab.

Úlohou je určiť hodnotu vlastnej frekvencie systému popísaného diferenciálnou rovnicou druhého rádu. Experimentálne určenie bude spočívať v tom, že systém budeme „budiť“ vstupom so sínusovým priebehom s postupne sa meniacou frekvenciou. Keď budiaca frekvencia bude sa rovnať vlastnej frekvencii systému amplitúda kmitania systému bude dosahovať maximum. Rovnica popisujúca systém nech má nasledovný tvar:

$$y'' + 0,1 y' + (0,628)^2 y = 10 \sin(\omega \cdot t)$$

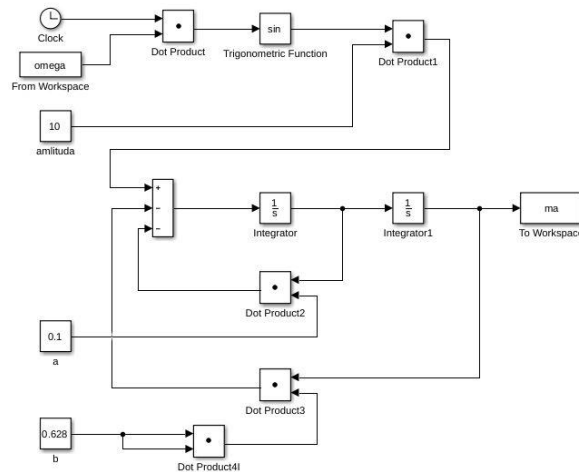
kde t je hodnota simulovaného času a ω je budiaca frekvencia. (Hodnota parametra **0,628** vlastne predstavuje hľadanú vlastnú frekvenciu systému).. Riadiaci algoritmus simulačného pokusu je realizovaný v jazyku MATLAB – obr.1. Tu je definovaná hodnota parametra ω pre riešenie uvedenej rovnice (jeden simulačný beh), ktoré je riešené prostredníctvom modulu

SIMULINK obr. 2. Priebeh výstupnej hodnoty y je prostredníctvom vektora \mathbf{ma} sprístupnený spätne riadiacemu algoritmu pre jej nájdenie maximálnej hodnoty amplitúdy.

```

1 clear;
2 Om = 0.6; deltaOm = 0.001;
3 k = 0;
4 omega = [0,Om];
5 amak = 0; amold = -1;
6
7
8 while amak - amold > 0
9     k = k+1;
10    sim('v1frekv2');
11    pole(k) = max(ma);
12    Om = Om + deltaOm;
13    omega = [0,Om];
14    amold = amak;
15    amak = pole(k);
16    kk(k)=k;
17 end
18
19 display('vlastna frekvencia')
20 Om = Om - deltaOm;
21 disp(Om)
22 display('pocet simulacnych behov')
23 disp(k)
24 plot(kk,pole)
25 legend('x sim.behy y max ampl.','Location','West')
26

```



Obr.1: Riadiaci algoritmus simulačného pokusu

Obr.2: Riešenie rovnice v SIMULIKu

Výsledky riešenia sú uvedené na obr.2 a v grafickej forme na obr.4. Z porovnania vyplýva, že výsledok od presnej hodnoty sa odlišuje o 0,003. Otázkou je, aký postup je potrebné zvoliť, aby sme výsledok dostali s väčšou presnosťou?

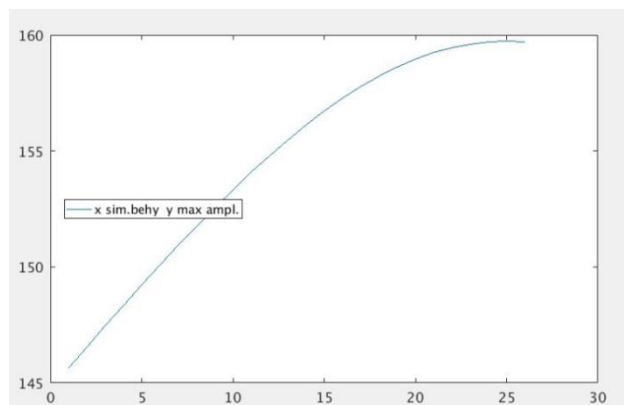
```

vlastna frekvencia
0.6250

pocet simulacnych behov
26

```

Obr.3: Výsledok riešenia

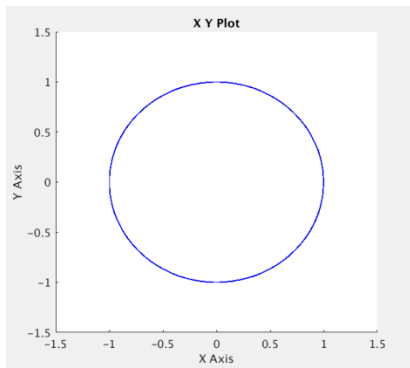


Obr.4: Výsledok riešenia v grafickej forme

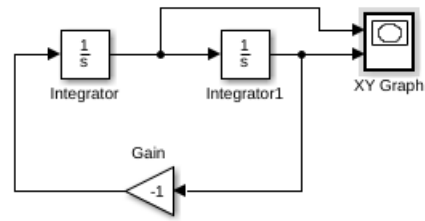
Nutnou podmienkou pre realizáciu náročných vedeckotechnických výpočtov sú základné znalosti z oblasti matematických metód. Uvedené je možné dokumentovať na jednoduchom príklade. Majme jednoduchú diferenciálnu rovnicu:

$$y'' + y = 0 \quad \text{s počiatočnými podmienkami } y(0) = 0 \quad \text{a} \quad y'(0) = 1$$

Jej presné riešenie je $y = \sin(t)$, resp. $y' = \cos(t)$. Ich grafické znázornenie pri prírastku t o hodnotu 0,1 vytvorí kružnicu s polomerom 1, obr.5. Riešenie diferenciálnej rovnice prostredníctvom modulu SIMULINK je znázornené na obr. 6.



Obr.5: Presné riešenie



Obr.6: Riešenie rovnice v SIMULIKu

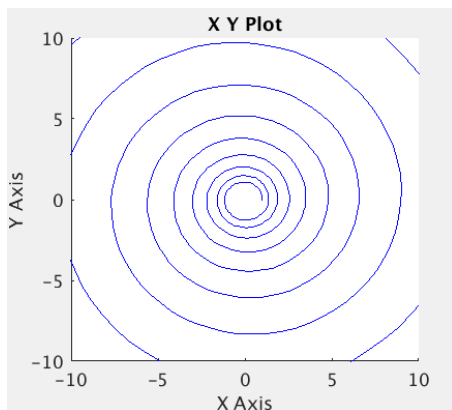
Nasledovné obrázky znázorňujú rôzne priebehy výsledkov podľa navolenej numerickej metódy jej definovaných parametroch.

Obr.7. riešenie Eulerovou metódou pri simulačnom kroku $h = 0,1$.

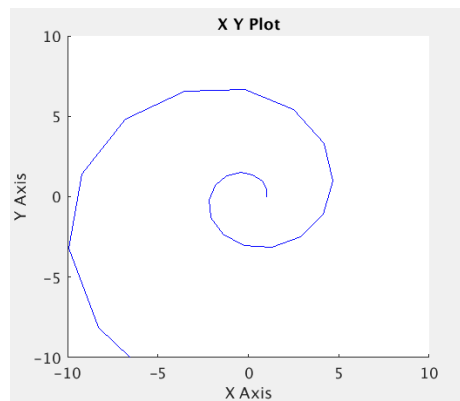
Obr.8 riešenie Eulerovou metódou pri simulačnom kroku $h = 0,5$.

Obr.9. riešenie metódou ode45 s testom presnosti, implicitnými parametrami a $t_{max} 10000$.

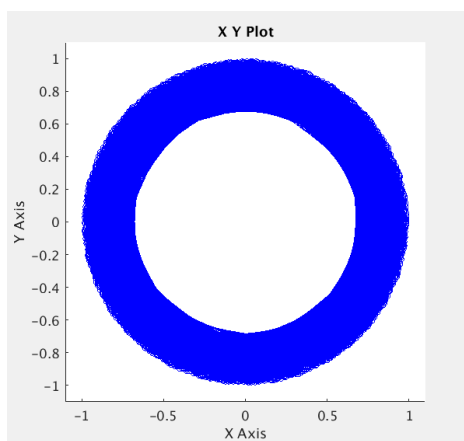
Obr.10 riešenie metódou ode45 s testom presnosti, max. integ. krok = 0,1 a $t_{max} 10000$.



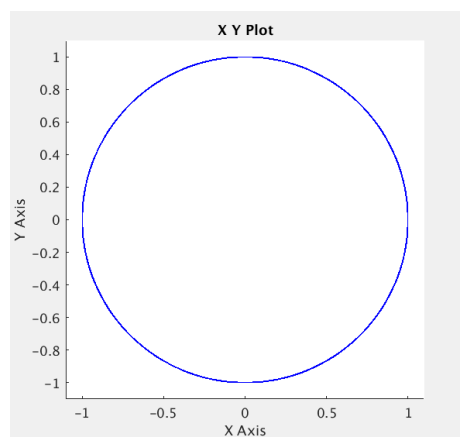
Obr.7: Riešenie Eulerovou metódou $h = 0,1$



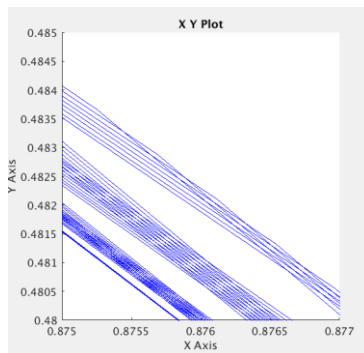
Obr.8: Riešenie Eulerovou metódou $h = 0,5$



Obr.9: Riešenie metódou ode45



Obr.10: Riešenie metódou ode45 , max. integ. krok = 0,1



Obr.11: Fragment z obr. 10

Obr.10 a zväčšený fragment z tohto obrázku na obr.11 predstavuje už riešenie s numerickou stabilitou a zodpovedajúcemu výsledku presného riešenia z obr.5.

Z porovnania jednotlivých riešení vyplýva nutnosť znalostí z numerického riešenia diferenciálnych rovníc, nakoľko je na používateľovi, aké numerické metódy s príslušnými parametrami si zvolí, čím je aj daný správny, resp. nesprávny výsledok riešenia.

2.2 Paralelné výpočty v prostredí softvérového balíka Matlab

Technické vybavenie klastra na TUKE umožňuje paralelné výpočty na viacjadrových procesoroch s využitím vzájomne prepojených viacerých procesorov a s využitím akceleratorov GPU. Zakúpené aplikačné programové vybavenie v rámci projektu SIVVP, orientované na využitie metódy konečných prvkov, uvedené možnosti paralelných výpočtov už plne využíva. Taktiež hodne rozšírený širokospektrálny softvér Matlab uvedené technické možnosti dokáže využiť. Na toto využitie bolo vykonaných niekoľko prác, ktoré zastrešuje diplomová práca [2], z ktorej niektoré výsledky budú prezentované v ďalšom.

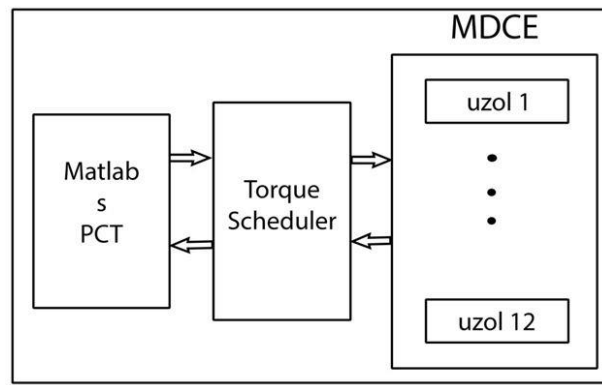
Pre podporu paralelných výpočtov Matlab obsahuje modul Modul Parallel Computing Toolbox (PCT) a Matlab Distributed Computing Server (MDCS) 0 0 0 0.

Modul Parallel Computing Toolbox (PCT) umožňuje riešiť výpočtovo a dátovo náročné problémy s využitím viacjadrových procesorov, grafických akceleratorov a výpočtových klastrov. Bez väčšieho zásahu do kódu umožňuje aby na jednom počítači bežalo niekoľko paralelných procesov. Modul umožňuje vykonať paralelný výpočet na maximálne dvanástich paralelných procesoroch.

Modul PCT v spojení s „Matlab Distributed Computing Server“ (MDCS) poskytuje spúšťanie úloh nielen na lokálnom stroji ale aj v sieti počítačov a na klastri. Výhodou je, že vytvorená úloha v prostredí Matlab pomocou PCT je ľahko prenositeľná na MDCS. Paralelné výpočty sa môžu spúšťať lokálne na viacprocesorovom počítači, ladiť, testovať a vyvíjať dané úlohy. Na spúšťanie úloh v klastri je potreba mať na ňom nainštalovaný MDCS, čo umožní využiť väčší výkon a paralelizmus.

Matlab Distributed Computing Server v spolupráci s Parallel Computing Toolbox umožňuje riešiť výpočtovo veľmi náročné úlohy v prostredí Matlab na viacjadrových a viacprocesorových počítačových systémoch ako sú napr. tiež klastre. Obsahuje podporu viacerých plánovačov, ako napr. interný plánovač MathWorks Job Manager, Microsoft Compute Cluster Server a plánovače iných výrobcov ako napríklad PBS Pro a TORQUE (nainštalovaný na systéme TUKE). MDCS umožňuje aj centrálnu správu licencií, t.j. výpočtové uzly nepotrebujú vlastnú licenciu, ale tá je spravovaná licenčným manažérom. Taktiež je tu podpora dynamických licencií závislých produktov, to znamená, že všetky produkty dostupné na klientskej stanici sú dostupné aj na výpočtových uzloch.

Paralelné spracovanie umožňuje spracovanie paralelných slučiek, bloky kódu, distribuovaných polí, paralelných numerických výpočtov.



Obr.12: Schéma spolupráce Matlabu s PCT a MDCE

Využitie GPU predstavuje možnosť vykonať výpočet pomocou vysokovýkonných grafických akceleratorov s veľkým počtom jadier. Podmienkou je prítomnosť grafického akceleračného procesora NVIDIA podporujúceho architektúru CUDA a s výpočtovou schopnosťou úrovne 1.3 a vyššie.

Parallel Computing Toolbox poskytuje špeciálne dátové pole, **gpuArray**, ktoré spolu s niekoľkými asociovanými funkciami umožňujú vykonať výpočty priamo na grafickom akceleračnom procesore priamo z Matlabu. Funkcie, ktoré sú podporované sú napríklad **fft**, **chol**, **lu**, **mldivide** známe ako operátor delenia zľava (backslash operátor).

Z uvedeného vyplýva, že problematika paralelných výpočtov u tohto programového systému je značne rozpracovaná. Pre dokumentovanie je uvedený jednoduchý príklad s využitím príkazu pre paralelné slučky **parfor**. V príklade sa vytvorí náhodná matica s rozmerom 1000x1000 a tá sa následne 256 krát invertuje. Generovanie čísiel zabezpečuje funkcia **rand**, ktorá generuje rovnomerne rozložené čísla v intervale 0 až 1.

```

%nastavenie paralelnej konfiguracie pre MPIEXEC
sched= findResource('scheduler', 'type', 'mpiexec')
set(sched, 'MpiexecFileName', 'mpiexec')
set(sched, 'EnvironmentSetMethod', 'setenv')

matlabpool (sched, 4) %otvorenie matlabpool
matlabpool size

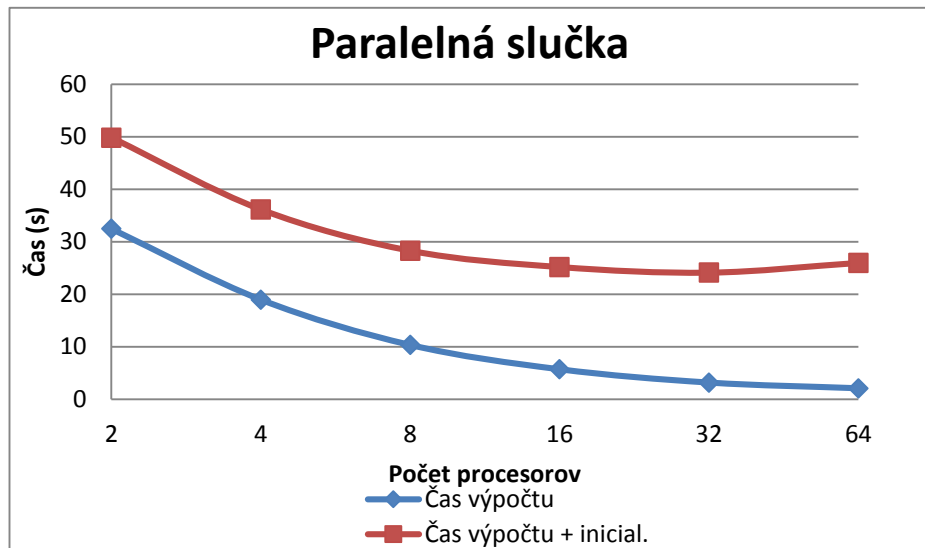
tic; %zaciatok merania casu
parfor i=1:(4*64); %paralelna slucka
    inv(rand(1000)); %inverzia nahodnej matice
end;
toc %koniec merania casu

matlabpool close %zavretie matlabpool
  
```

Obr.13: Príklad paralelnej slučky

Príklad (obr. 13) bol modifikovaný pre počet jadier 2, 4, 8, 16, 32 a 64, tzn. bolo potrebné meniť veľkosť premennej matlabpool v rozmedzí od 2 do 64 (riadok „matlabpool (sched, x)“). Pomocou matlabpool sa otvára resp. zatvára združenie jadier pre paralelné výpočty, v podstate to je definovanie, na koľkých paralelných jadrách bude prebiehať výpočet. Dôležité je, aby počet jadier rezervovaných v skripte pre dávkový systém a počet jadier v matlabpool bol rovnaký.

Úloha bola riešená v dávkovom režime. Na Obr. 14 je graf časov potrebných pre výpočet. Čas výpočtu bol meraný dvojakým spôsobom. Jeden meraný čas reprezentuje čas potrebný na samotný výpočet úlohy v Matlabe. Druhý čas reprezentuje čas potrebný na inicializáciu Matlabu spolu s časom potrebným na samotný výpočet. Na odmeranie druhého času bol využitý linuxový príkaz `time`, zadaný pred každým spustením výpočtu. Môžeme vidieť, že do počtu 16 jadier je potrebný približne rovnaký čas na inicializáciu. Avšak pri zvyšovaní počtu jadier sa inicializačný čas začína zväčšovať z dôvodu potrebnej rezervácie väčšieho počtu jadier pre výpočet.



Obr.14: Graf časov pre paralelnú slučku

3 Záver

Pri príprave projektu SIVVP a tiež v jeho počiatku riešenia boli oslovené všetky akademické a vysokoškolské pracoviská s cieľom konkretizácie využitia produktov tohto projektu. Po spracovaní výsledkov sa uskutočnil výber aplikačného softvéru, pričom požiadavky presahovali finančné možnosti projektu približne 1,7 krát.

Z hľadiska udržateľnosti výsledkov projektu významnú položku tvorí jeho ďalšie finančné zabezpečenie, kde ako hlavné položky je možné uviesť:

1. Finančné náklady na inovácie hardvéru a jeho ďalšieho rozvoja.
2. Finančné náklady udržiavacie licenčné poplatky, na inovácie softvéru a nákup ďalších softvérových balíkov.
3. Finančné náklady na prevádzku, kde významnú zložku tvoria náklady na elektrickú energiu.

Okrem uvedeného významnou oblasťou je personálne zabezpečenie, nakoľko tu ide o nárast prác nielen v oblasti systémového zabezpečenia, ale tiež tu vznikajú značné odborné nároky na využívanie daného aplikačného softvéru, nakoľko sa tu ide o rozsiahle softvérové balíky.

Z hľadiska budúcnosti bude najvhodnejšie vytvorenie Národného centra pre SIVVP. Týmto by sa dosiahol hlavný cieľ, aby SIVVP bola poskytovaná ako ucelená služba pre podporu výskumu s patričným finančným zabezpečením. Je potrebné uviesť, že z hľadiska ďalšieho plánovacieho obdobia sa počíta

s ďalšími aktivitami v uvedenej oblasti.

Určitým vzorom pre oblasť vysokých škôl by mohlo byť riešenie v Čechách. V rámci ČESNETu (podobné združenie v ČR ako je u nás SANET) je vytvorené tzv. METACENTRUM, ktoré je budované už dlhé obdobie a jeho základným cieľom je poskytovanie informačných a komunikačných technológií ako služba pre výskum a výučbu, podrobnejšie vid' <https://metavo.metacentrum.cz/>, <https://metavo.metacentrum.cz/>.

Článok je zverejnený s finančnou podporou združenia EUNIS-SK.

Literatúra

- [1] Lacko T., Šujanský M.: Informácia o stave riešenia projektu „Slovenská infraštruktúra pre vysokovýkonné počítanie“ – 5, Zborník z konferencie UNINFOS 2015, SZU v Banskej Bystrici, 24. - 26. 10. 2015, v tlači.
- [2] Durkáč I.: Metodika použitia aplikačných programov v distribuovanom a paralelnom prostredí Diplomová práca, ved. a konz. Šujanský, TU v Košiciach, FEI, 2013.
- [3] HUMUSOFT: Parallel Computing Toolbox and MDCS [online]. HUMUSOFT, 2016 [cit 2016-03-26]. Dostupné na internete: <<http://www.humusoft.com>>.
- [4] MATHWORKS: Parallel Computing Toolbox [online]. Mathworks, 2016 [cit 2016-03-29]. Dostupné na internete: <<http://www.mathworks.com/products/parallel-computing/>>.
- [5] MATHWORKS: Matlab Distributed Computing Server [online]. Mathworks, 2016 [cit 2016-03-29]. Dostupné na internete: <<http://www.mathworks.com/products/distriben/>>.
- [6] MATHWORKS:MATLAB GPU Computing Support for NVIDIA CUDA-Enabled GPUs [online]. Mathworks, 2016 [cit. 2016-03-30]. Dostupné na internete: <<http://www.mathworks.com/discovery/matlab-gpu.htm>>.