Výpočty matematického modelovania lietadla
pomocou MPI v počítačoch leteckého trenažéra

Peter Kvasnica, Tomáš Páleník

Článok sa venuje metóde matematického modelovania výpočtu informačného systému trenažéra. Na jeho realizovanie využíva poznatky teórie riadenia decentralizovaných systémov tvorených počítačmi. Takto navrhnuté modelovanie sa realizuje niekoľkými počítačmi, ktoré vytvárajú distribuovaný počítačový systém leteckého trenažéra. Uvedené počítanie sa realizuje pomocou nástroja (systému) MPI, ktorý výrazne ovplyvňuje časovú náročnosť modelovania trenažérového systému. Uvedená závislosť je definovaná tvarom rovníc a architektúrou použitého počítačového systému. V článku sa poukazuje aj na riešenie matematických modelov rôznymi nástrojmi. Metóda analýzy simulácie a analytického prístupu implementácie paralelného výpočtu sa vykonáva jednoprocesorovou architektúrou. Paralelné modelovanie na počítačoch reálneho leteckého trenažéra dokumentuje časovú úsporu a požadovanú presnosť. Tento poznatok dovoľuje v jednom časovom okamihu počítať matematické modely pomocou MPI na dekomponovanom výpočtovom systéme.

Úvod

Na počítači možno riešiť niektoré úlohy z kinematiky, ako aj z dynamiky pohybu objektov trenažérov. Pri úlohách z kinematiky hmotného bodu ide najmä o zložené pohyby. Podstata riešenia úloh tohto typu je v tom, že sa na počítači modelujú rovnice jednotlivých zložiek pohybu (spočítaním vzniká výsledný pohyb). Fyzikálny problém grafickej konštrukcie trajektórie zloženého pohybu sa nahradí matematickým zápisom, a to zisťovaním priebehu matematickej funkcie. Metódami súčasnej numerickej matematiky a počítačovej grafiky je to riešiteľná úloha.

V teórii analýzy dynamiky systémov sú informácie o správaní systému centralizované. Návrh a tvorba informačných systémov s distribuovanými bázami údajov sú dôležitými predpokladmi pre decentralizované riadiace systémy [6]. Modelovanie takýchto vlastností trenažéra pomocou matematických modelov je jednou z najdôležitejších činností pri jeho výstavbe.

Vďaka matematickému modelovaniu a počítačovej simulácii sa pomocou počítača úspešne riešia aj niektoré úlohy z dynamiky hmotného bodu (naše riešenie analýzy dynamiky pohybu lietadla možno redukovať na spomenutý problém).

1. Matematické modely lietadla v trenažéri

Matematický model vyššie spomenutých úloh sa opisuje sústavou obyčajných diferenciálnych rovníc. V súčasnosti možno na výpočet použiť dva rozdielne prístupy. Prvý spočíva v analytickom riešení matematického modelu (sústavy diferenciálnych rovníc) metódami matematickej analýzy. V druhom prípade realizuje riešenie matematického modelu výpočtová technika [8].

Podľa súčasných matematických poznatkov možno analyticky riešiť iba malú časť diferenciálnych rovníc. Aplikuje sa predovšetkým metóda neurčitých koeficientov, metóda postupných aproximácií, ako aj metóda Taylorovho rozvoja, ktorá sa dá súčasne použiť aj pri numerickom riešení diferenciálnych rovníc.

Zhrňme si podstatu algoritmu na numerické riešenie diferenciálnych rovníc n-tého. Počiatočnú úlohu pre diferenciálnu rovnicu n-tého rádu možno definovať takto:

Nech je daná diferenciálna rovnica n-tého rádu v tvare:

$$y^{(n)} = f\left(x, y, y', \ldots, y^{(n-1)}\right) \tag{1}$$

Úlohou jej riešenia je nájsť funkciu:

$$y = y(x) \tag{2}$$

vyhovujúcu Cauchyho počiatočným podmienkam:

$$\begin{aligned} &y(x_0) = y_0 \\ &y'(x_0) = y' \\ &\ldots \\ &y^{(n-1)}(x_0) = y_0^{(n-1)} \end{aligned} \tag{3}$$

kde

$$\begin{aligned} &y_0 \\ &y'_0 \\ &\ldots \\ &y_0^{(n-1)}(x_0) = y_0^{(n-1)} \end{aligned} \tag{4}$$

sú zadané konštanty.

Na numerické riešenie diferenciálnej rovnice vyššieho rádu využívame metódy používané pri riešení sústavy diferenciálnych rovníc 1. rádu, a preto zadanú diferenciálnu rovnicu n-tého rádu vhodnými substitúciami prevádzame na sústavu diferenciálnych rovníc 1. rádu:

$$\begin{aligned} &y_1 = y' \\ &y_2 = y'' = y' \\ &\ldots \\ &y_{n-1} = y^{(n-1)} = y'_{(n-2)} \end{aligned} \tag{5}$$

Uvedené substitúcie vlastne tvoria sústavu diferenciálnych rovníc. Sústavu doplníme o zadanú diferenciálnu rovnicu s dosadenými premennými $y_1, \ldots, y_{n-1}$ a takto novokreovanú sústavu upravíme na normovaný tvar:

$$y' = y_1$$
$$y'_1 = y_2$$
$$\ldots$$
$$y'_{n-2} = y_{n-1}$$
$$y'_{n-1} = f(x, y, y_1, \ldots, y_{n-1}) \qquad (6)$$

Na riešenie sústavy (6) možno použiť známe numerické metódy riešenia diferenciálnych rovníc 1. rádu.

Na analýzu správania systémov opísaných lineárnymi diferenciálnymi rovnicami sa v praxi často využíva inžinierska metóda Laplaceovej transformácie. Výsledky riešenia sa zisťujú inverznou Laplaceovou transformáciou v časovej oblasti. Laplaceova transformácia je prevod funkcie z časovej oblasti, ktorá spĺňa požadované vlastnosti, do komplexnej roviny určenej Levinštejnovým vzťahom [7]:

$$F(p) = \int_0^\infty f(t) e^{-pt} dt = L\{f(t)\} = Lf \qquad (7)$$

kde $p \in C$, $p = x + iy$ sa nazýva Laplaceova transformácia. Funkcia F sa nazýva obraz funkcie v Laplaceovej transformácii a funkcia f premennej $t \in (0, +\infty)$ sa nazýva vzor Laplaceovho obrazu [8].

Inverzná Laplaceova transformácia sa vypočíta podľa:

$$L_{-1}\{F(p)\} = L_{-1}F(p) = f(t) \qquad (8)$$

2. Riešenie matematického modelu trenažéra

Matematický model vyššie spomenutých úloh sa opisuje sústavou obyčajných diferenciálnych rovníc. V súčasnosti možno na výpočet použiť dva rozdielne prístupy. Prvý spočíva v analytickom riešení matematického modelu (sústavy diferenciálnych rovníc) metódami matematickej analýzy. V druhom prípade realizuje riešenie matematického modelu výpočtová technika [8].

Modelovanie a simulácia prakticky priamo závisia od dostupnosti vhodného simulačného nástroja spolu s výkonným technickým vybavením počítača. Vybrať vhodný simulačný nástroj na analýzu údajov, ktoré sú dostupné, podporované viacerými operačnými systémami a vývojom, je ťažké. Dôraz sa kladie na filozofiu otvorených zdrojových kódov (open source) alebo voľne šíriteľného kódu (free software) pre požadované úlohy. Úlohu možno riešiť v programovom (softvérovom) prostredí Matlab, Scilab, Mathematica a pod.

Matlab je výkonný, komplexný, jednoducho použiteľný na analýzy, vizualizácie, modelovanie a vývoj úplných aplikačných modelov trenažérov. Matlab má vlastný jednoduchý programovací jazyk, čo umožňuje k už zabudovaným funkciám pridávať podľa potreby vlastné [9].

Konfiguráciu Matlab-u môžeme hocikedy rozšíriť pridaním rozširujúcich modulov alebo vlastných programov. Simulink je programový balík na modelovanie, simuláciu a analýzu dynamických systémov. Tento grafický a simulačný nástroj umožňuje jednoducho vytvárať viackrát použiteľné modely lineárnych a nelineárnych systémov, modelovaných v spojitom alebo diskrétnom čase, alebo dokonca v oboch súčasne.

Scilab je navrhnutý na aplikácie riadenia systémov a signálových procesov. Je vytvorený z troch samostatných častí: interpret príkazov, knižnice funkcií a knižnice rutín v jazykoch C a Fortran. Tieto rutiny sú nezávislé od prostredia a všetky sú prístupné cez knižnicu Netlib.

Dôležitou vlastnosťou syntaxe Scilab-u je spracovanie matíc a súvisiacich operácii. Scilab manipuluje so zoznamami a typmi zoznamov ako s prirodzenou symbolickou reprezentáciou prekladaných matematických objektov [10]. Aj tieto vlastnosti predurčujú Scilab na efektívne overovanie návrhu, simuláciu a opis matematických modelov leteckých trenažérov.

Mathematica je programový (softvérový) systém na technické výpočty, používaný vo vede, technológii, vývoji a obchode. Tento nástroj sa prvýkrát objavil v r. 1988 a odvtedy sa počet jeho používateľov zväčšuje vďaka jeho dostupnosti v širokom rozsahu počítačových platforiem, od PC s Windows a pracovných staníc až k UNIX-ovým pracovným staniciam [11].

Mathematica je počítačový program a jazyk určený pre matematické a iné aplikácie. Je použiteľný v numerických a symbolických výpočtoch, vizualizačnom systéme funkcií a údajov, ktoré obsluhuje jednotne. Predstavuje programovací jazyk vysokej úrovne, modelovanie a analýzu údajov, systém na reprezentáciu znalostí vo vede a technike a pod.

3. Paralelné matematické modelovanie

Sekvenčný spôsob behu programu matematického modelu sa vyznačuje výpočtom rovníc v jednotnom čase počítača, na ktorom sa realizuje výpočet. Operácie programového kódu sa vykonávajú sekvenčne v danom poradí. Nevýhoda tejto metódy je v obmedzení výkonu procesora realizujúceho výpočet. Riešením uvedeného obmedzenia je paralelné vykonávanie programového kódu matematických modelov, pri ktorom je potom dôležitá synchronizácia a jednotný čas modelovania. Možným spôsobom je počítanie týchto modelov v klastri. Pre samotné paralelné matematické modelovanie si uvedieme jeho základné rysy počítania.

Klaster je najvyššia organizačná jednotka vysokovýkonného počítania na tejto platforme. Klaster pozostáva z týchto elementov [2]:

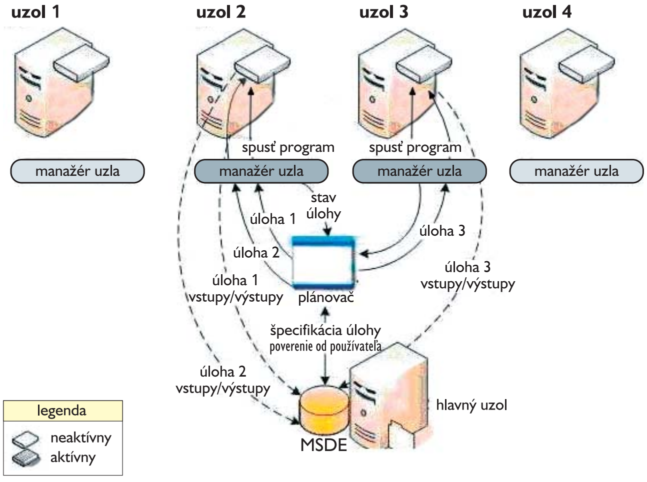
1. uzol (Node) – samostatný počítačový uzol s jedným alebo viacerými procesormi,
2. rad (Queue) – organizačná jednotka, ktorá poskytuje spracovanie a obsluhu úloh,
3. úloha (Job) – množina úloh inicializovaných používateľom.

Úloha reprezentuje vykonateľný program na danom počítačovom (výpočtovom) uzle. Úlohou môže byť sekvenčný program (jeden procesor) alebo paralelný program Message Passing Interface (MPI) s viacerými paralelnými procesmi.

3.1 Metóda paralelného vykonávania

Užívatelia môžu použiť rôzne programovacie jazyky, ktoré vhodne spĺňajú ich požiadavky na tvorbu a zadávanie paralelných výpočtov. Tieto prednosti sú v tom, že spokojne využívajú platformovo nezávislé a spoľahlivé vykonávanie [3].

Úlohy spracovávajú v sériovom alebo paralelnom móde. V sériovom móde, úlohy bežia sekvenčne na dostupných zdrojoch v uzloch (nodes). Obr. 1 ilustruje ako úloha 1 je nastavená na vykonanie na procesor na druhom uzle (node), následne úloha 2 je nastavená na tento procesor, úloha 3 sa presúva na procesor tretieho uzla (node), atď.. Na takejto architektúre prebiehajú matematické modely trenažéra v reálnom čase. Paralelné výpočty sú typicky vykonávané programom Message Passing Interface (MPI) (nazývaným MPI) pomocou volania úlohy mpiexec na počítačových uzloch (nodes) [4].



Obr.1 Paralelné úlohy bežiace na klastrovej platforme

3.2 Matematické modely

Používatelia môžu použiť rôzne programovacie jazyky, ktoré vhodne spĺňajú ich požiadavky na tvorbu a zadávanie paralelných výpočtov. Tieto prednosti sú v tom, že pokojne využívajú platformovo nezávislé a spoľahlivé vykonávanie [3]. Matematické modely lietadla v trenažéri [1], [5] sú v našom prípade upravené na želaný tvar. Pre prírastok rýchlosti platí rovnica (9), definujúca príspevok zo zmeny dodávky paliva (10), resp. príspevok zo zmeny polohy výškového kormidla (11):

$$\Delta V(s) = -W_V^{\delta_T}(s)\Delta\delta_T(s) - W_V^{\delta_B}(s)\Delta\delta_B(s) \tag{9}$$

Pre dodávku paliva:

$$W_V^{\delta_T}(s) = 5\frac{s^3 + 1{,}12s^2 + 62{,}782s + 25{,}32}{s^4 + 1{,}1338s^3 + 62{,}7975s^2 + 28{,}6585s + 4{,}09291}\Delta\delta_T \tag{10}$$

Pre výškové kormidlo:

$$W_V^{\delta_B}(s) =$$
$$= \frac{-0{,}11\cdot(9{,}81s + 620{,}973) - 0{,}42\cdot(-9{,}81s - 10{,}0062)}{s^4 + 1{,}1338s^3 + 62{,}7975s^2 + 28{,}6585s + 4{,}09291}\Delta\delta_B(s) \tag{11}$$

Pre prírastok uhla nábehu platí rovnica (12), definujúca príspevok zo zmeny dodávky paliva (13), resp. príspevok zo zmeny polohy výškového kormidla (14):

$$\Delta\alpha(s) = -W_\alpha^{\delta_T}(s)\Delta\delta_T(s) - W_\alpha^{\delta_B}(s)\Delta\delta_B(s) \tag{12}$$

Pre dodávku paliva:

$$W_\alpha^{\delta_T}(s) =$$
$$= 5\frac{0{,}002s^2 - 0{,}2581s - 0{,}1}{s^4 + 1{,}1338s^3 + 62{,}7975s^2 + 28{,}6585s + 4{,}09291}\Delta\delta_T(s) \tag{13}$$

Pre výškové kormidlo:

$$W_\alpha^{\delta_B}(p) = \frac{-0{,}11\cdot(-s^3 + 0{,}8862s^2 + 0{,}012422s - 2{,}4525}{s^4 + 1{,}1338s^3 + 62{,}7975s^2 + 28{,}6585s + 4{,}09291}$$
$$\frac{-0{,}42\cdot(-s^2 - 0{,}4138s - 0{,}02514)}{s^4 + 1{,}1338s^3 + 62{,}7975s^2 + 28{,}6585s + 4{,}09291}\Delta\delta_B(s) \tag{14}$$

V rovnici (10), resp. (13) predstavuje zmena dodávky paliva jednotkový skok, v Laplaceovej transformácii $\Delta\delta_T(s) = 1/s$. V rovnici (11), resp. (14) predstavuje zmena polohy výškového kormidla jednotkový skok, v Laplaceovej transformácii $\Delta\delta_B(s) = 1/s$.

4. MPI modelovanie v jednoprocesorových systémoch trenažéra

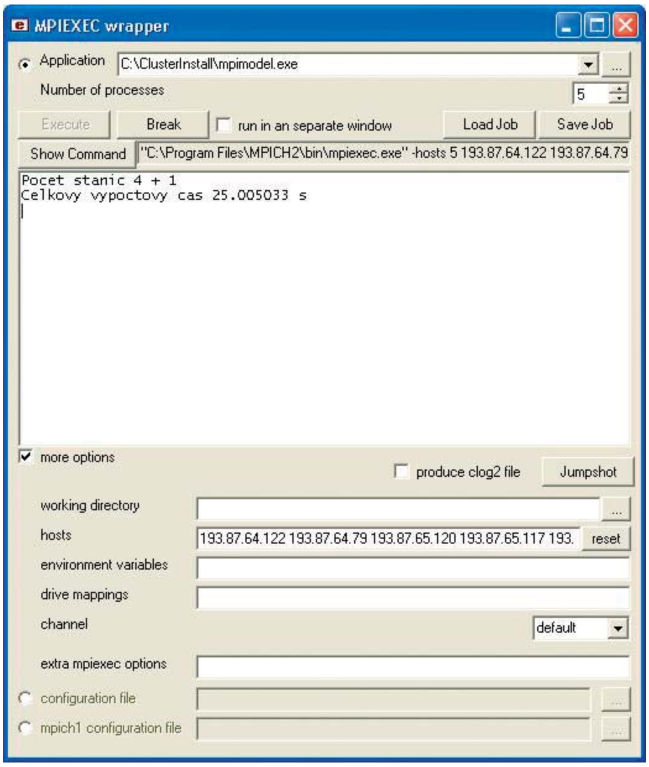
Pre procesy matematického modelovania využijeme systém MPI, ktorý v krátkosti opíšeme. Tento systém pozostáva z viacerých uzlov, zložených z procesorov a pamätí spojených prepájacou sieťou. Pre absenciu globálnej pamäte treba údaje medzi jednotlivými uzlami presúvať práve pomocou systému message passing, ktorý je založený hlavne na dvojici príkazov Pošli/Prijmi správu [12]. Tie sú vhodným spôsobom implementované v zdrojovom kóde aplikácie určenej na výpočet na jednotlivých uzloch. Autonómne uzly sú schopné tieto údaje v podobe správ ukladať do zásobníka. Zásobník (buffer) tu predstavuje vyhradené pamäťové miesto na ukladanie správ, ktoré čakajú na prijatie alebo zaslanie.

```
MPI_Init()
...
MPI_Barrier()
if (processor == 0){        //centrálny uzol
  // zhromazdovanie udajov
} else {
  switch (processor)        //výpočtové uzly
  {
          case 1: ModelovanieRovnice1(); break;
          case 2: ModelovanieRovnice2(); break;
          case 3: ModelovanieRovnice3(); break;
          case 4: ModelovanieRovnice4(); break;
  }
}
MPI_Finalize()
```

Obr.2 Pseudokód pre použitú aplikáciu využívajúcu MPI

Systém MPI obsahuje aj základné prostriedky na koordináciu paralelne bežiacich procesov, ako je bariérová synchronizácia, precedenčná synchronizácia a pod. [13]. V prípade nášho modelovania leteckého trenažéra bola použitá bariérová komunikácia, ktorá má tú vlastnosť, že tesne po inicializácii MPI zabezpečí vzájomné časové zladenie výpočtu všetkých uzlov. Volanie funkcie MPI_Barrier() v programe spôsobí zablokovanie volajúceho uzla až dovtedy, pokiaľ k tomuto volaniu nedospejú aj ostatné programy v uzle. Situáciu približuje pseudokód na obr. 2.

Po synchronizácii všetkých procesov dochádza v kóde aplikácie k rozdeľovaniu úloh pre jednotlivé uzly. Jeden z týchto uzlov predstavuje centrálny uzol, určený na zhromažďovanie a ukladanie počítaných výsledkov. Na samotné počítanie sú pridelené zvyšné uzly a ich úlohou je modelovanie jednotlivých rovníc matematického modelu trenažéra (10), (11), (13) a (14) v reálnom čase.

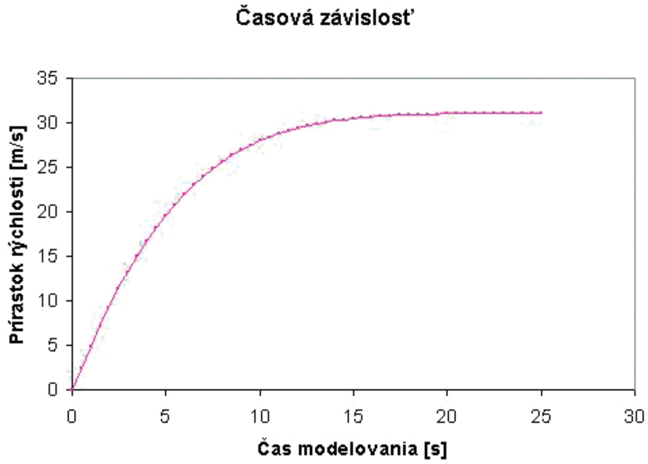


Obr.3 Dialógové okno MPIEXEC MPI

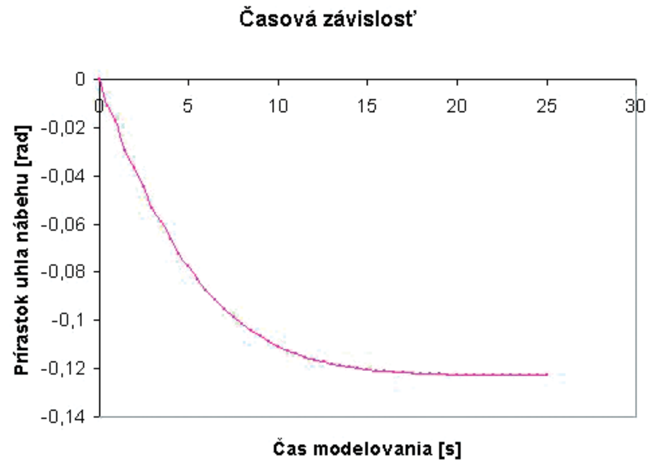
Na obr. 3 je dialógové okno zobrazujúce stav spúšťania aplikácie mpiexec matematického modelu trenažéra. Aplikácia mpimodel.exe sa inicializuje z podadresára, ktorý je určený položkou „Application“. Počet procesov aplikácie matematického modelu v zostave klastra je 5 („Number of processes“). Aktivácia výpočtu sa vykonáva z riadku príkazom C:\ProgramFiles\ MPICH\bin\mpiexec.exe -hosts… („Show command“). Počítače v zostave klastra (hosts) majú tieto IP adresy: 193.87.64.122, 193.87.64.79, 193.87.65.120, 193.87.65.117, 193.87.65.101. Vo výstupnom grafickom okne je zobrazený počet procesov a celkový čas modelovania.

5. Výsledné modely systému trenažéra

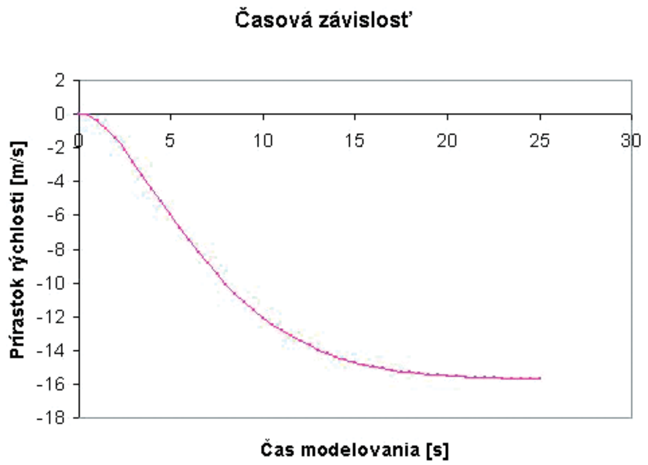
V našom prípade sme realizovali prepojenie 5 uzlov Celeron 2,4 GHz/ 512 MB RAM pomocou 100 Mb/s siete typu ethernet. Na prepojenie sme použili implementáciu MPI s názvom MPICH2 a v komunikácii zvolili blokujúce operácie [13].



a) výsledok simulácie rovnice (10)



c) výsledok simulácie rovnice (13)



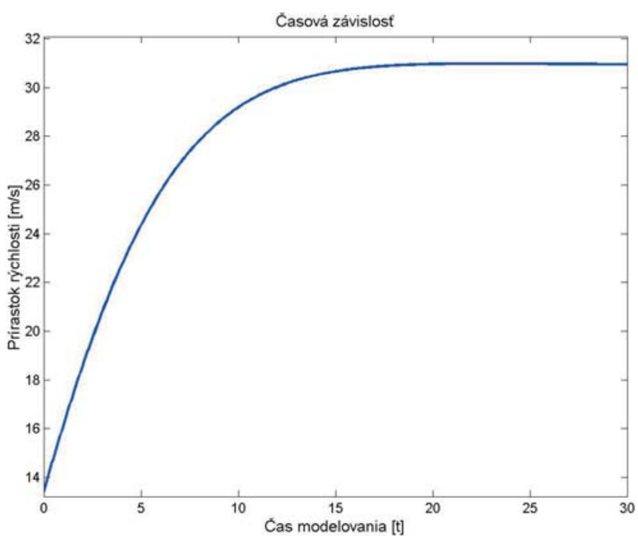
b) výsledok simulácie rovnice (11)

d) výsledok simulácie rovnice (14)

Obr.4

Uzol s adresou 193.87.64.122 predstavuje hlavný riadiaci uzol a zvyšné uzly sú určené na modelovanie rovníc matematického modelu. Po spustení aplikácie cez grafické rozhranie sa štyrom výpočtovým uzlom pridelí modelovanie jednotlivých rovníc v reálnom čase. Procesy bežia na uzloch 193.87.64.79, 193.87.65.120, 193.87.65.117, 193.87.65.101.

Celá simulácia trvá 25 s a medzivýsledky rovníc sa v pravidelných intervaloch zasielajú na hlavný centrálny uzol. Ten výsledky ukladá na vyhradené pamäťové miesto, ktoré po skončení simulácie uloží do súboru vo formáte spracovateľnom v programe Microsoft Excel. Po vykreslení grafov z vypočítaných údajov, založených na systéme MPI, získame obr. 4 a 5.



Obr.5 Výsledok simulácie rovnice (10) v Matlabe

Z grafických závislostí na obr. 4a) až 4d) vyplýva, že časová závislosť modelovania je zhodná vo všetkých prípadoch s modelovaním v nástroji Matlab.

Z modelovania výpočtu závislosti jednoprocesorového systému uzlov klastra vyplýva, že:

1. integráciu veličín na dekomponovaných podsystémoch treba upravovať podľa jednotného času komplexu trenažéra (bariérová komunikácia),
2. efekt dekompozície výpočtového systému sa synchronizáciou po sieti výrazne ovplyvňuje, v zložitejších prípadoch je veľmi ťažký,
3. z absolútneho porovnania modelovaných veličín uvedenej metódy výpočtov vyplýva, že závislosť výpočtov na klastrovom systéme je rovnako presná ako pri iných nástrojoch a je rýchlejšia ako na jednoprocesorovom.

Efektívnym a účelným sa javí riešiť výpočty matematických modelov v klastrovej platforme. Výsledky treba distribuovať po sieti k ďalším počítačom, kde pracujú doplnkové programové aplikácie. Týmto postupom sa zachová časový zisk oproti nedekomponovanému systému z pôvodného riešenia.

Záver

Simulovanie matematických modelov leteckého trenažéra klastrovou technológiou v súlade matematickým zápisom bolo vykonané podľa rovníc (10) a (11), resp. (13) a (14). Použila sa jednoprocesorová architektúra počítača a knižnice MPI (message passing interface) s podporou programového vybavenia v prostredí operačného systému Windows.

Vstupné parametre pre modelovanie dekomponovaných podsystémov leteckého trenažéra sú stacionárne, nemenné počas výpočtu.

Z výsledkov získaných simulovaním časovej závislosti na klastri vyplýva, že modelovanie podsystémov je náročnejšie, ale rýchlejšie.

Uvedenú metódu možno použiť aj pri iných modelovaných konfiguráciách trenažéra lietadla podľa Christophera [3] a Rolfa [5]. Potom je potrebné vhodným spôsobom spresniť a upraviť opis matematického zápisu na výpočet modelovania časovej závislosti modelovania na uzloch klastra (počítačoch) v leteckom trenažéri.

Literatúra

[1] BLAKELOCK, J. H.: Automatic control of aircraft and Missiles. Second Edition. New York: John Wiley & Sons Inc. 1991. ISBN 0-471-50651-6.

[2] POTA, S., SIPOS, G., JUHASZ, Z., KACSUK, P.: Distributed and Parallel Systems. Parallel Program Execution Support in the JGRID System. Springer Science, New York 2005. ISBN 0-387-23094-7. s. 13 – 19.

[3] FEITELSON, D. G., RUDOLF, L.: Parallel Job Scheduling: Issues and Approaches. Lecture Notes in Computer Science, vol. 949, 1995, s. 1 – xx.

[4] Sun Microsystems, Jini Technology Core Platform Specification, http://www.sun.com/jini/specs, 2005.

[5] ROLFE, J. M., STAPLES, K. J.: Flight Simulation. Cambridge: Cambridge University Press 1986, s. 42. ISBN 0-521-35751-9.

[6] SARNOVSKÝ, J. a kolektív: Riadenie zložitých systémov. Bratislava: Alfa 1992. ISBN 80-05-00945-3.

[7] LEVINŠTEJN, M. L.: Operátorový počet v elektrotechnike. 1977. s. 14 – 15.

[8] BARTSCH, H. J.: Matematické vzorce. Praha: Mladá fronta 1996. s. 720. ISBN: 80-204-0607-7.

[9] The MathWorks Inc. MATLAB, 9/1995, s. 3.

[10] INRIA. Introduction to Scilab. 1998, s. 8.

[11] WOLFRAM, S.: Mathematika A System for Doing Mathematics by Computer. Preface. Addison-Wesley Pub 1991. ISBN: 978-0201515022.

[12] EL-REWINI, H., ABD-EL-BARR, M.: Advanced Computer Architecture and Parallel Processing. New York: John Wiley & Sons, Inc. 2005. ISBN 0-471-46740-5.

[13] MPICH2 dostupné na: <www.mcs.anl.gov/mpi/mpich2>

Ing. Peter Kvasnica, PhD.

47

Trenčianska univerzita A. Dubčeka v Trenčíne
Fakulta mechatroniky
Katedra informatiky
Študentská č. 2, 911 50 Trenčín
Tel.: 032/74 00 104
e-mail: kvasnica@tnuni.sk

Ing. Tomáš Páleník

Trenčianska univerzita A. Dubčeka v Trenčíne
Fakulta mechatroniky
Katedra informatiky
Študentská č. 2, 911 50 Trenčín
Tel.: 032/74 17 583
e-mail: palenik@tnuni.sk