Annales UMCS Informatica AI 5 (2006) 15-22

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

A parallel implementation in the investigations of the controllability of an infinite dimensional system in the three dimensional rectangular prism

Jerzy Stefan Respondek*

Department of Theory of Information, Institute of Computer Science, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

Abstract

The article is devoted to the parallel implementation of an algorithm for finding triples generating equal sums of the squares. Based on a known single thread algorithm, the parallel algorithm is built. The obtained improvement of the time performance is verified on a multi-core server.

1. Introduction

Controllability is one of the most often investigated notions in the control theory, apart from stability and observability. It was defined in 1960 in the paper by Kalman [1]. After its defining, a fast progress of this notion could be observed. In literature there exist the following:

- criteria of the controllability of continuous-time and discrete-time linear finite dimensional dynamical systems,
- conditions of the controllability for time-varying and time-invariant systems,
- theorems for the verification of the controllability of dynamical systems in the Jordan Canonical form,
- conditions for the controllability of composite systems,
- the controllability with cone-type constrained controls and compact controls,
- the controllability of infinite dimensional systems.

In the article we extended the results from paper [2]. Article [2] presents a numerical algorithm for finding the triples, generating equal sums in the linear

* *E-mail address*: Jerzy.Respondek@polsl.pl

combination of the sums of the integer squares and its application for the controllability problems. The sum has the form (1):

$$S = \frac{i_1^2}{a^2} + \frac{j_1^2}{b^2} + \frac{k_1^2}{c^2}. \tag{1}$$

In this article we presented a parallel version of the algorithm from paper [2].

Below, we showed, as an example, 3 triples (i_1,i_2,i_3) generating an equal sum of the squares with parameters $m_1 = 1$, $m_2 = 3$, $m_3 = 5$:

$$\frac{143}{14} = \frac{2^2}{1} + \frac{4^2}{3} + \frac{1^2}{5} = \frac{1^2}{1} + \frac{4^2}{3} + \frac{4^2}{5} = \frac{3^2}{1} + \frac{1^2}{3} + \frac{1^2}{5}. \tag{2}$$

Such a linear combination of the integer squares exists in the eigenvalues of the Laplace second order differential operator defined in the 3-dimensional rectangular prism. The knowledge of the multiplicity of these eigenvalues is important in different dynamical properties of dynamical systems with the Laplace differential operator in its mathematical model, including controllability.

2. Problem formulation

Let us have a look at the following equation:

$$\frac{i_1^2}{a^2} + \frac{j_1^2}{b^2} + \frac{k_1^2}{c^2} = \frac{i_2^2}{a^2} + \frac{j_2^2}{b^2} + \frac{k_2^2}{c^2}, \tag{3}$$

where:

$a, b, c \in R_+$ are the given real numbers

$$i_1, j_1, k_1, i_2, j_2, k_2 \in Z_+, \tag{4}$$

where Z_+ is a set of positive integer numbers. Additionally, in the further investigations of the controllability, the following constraint is necessary:

$$i_1, j_1, k_1, i_2, j_2, k_2 \le N, \tag{5}$$

where N is a given positive integer number.

3. Single thread solution

The single thread solutions of the considered problem is presented in paper [2]. All positive, non-trivial and non-symmetrical integer solutions of equation (3) can be expressed by the following two series [2] pp.144-145:

$$\begin{cases} \left(i_1, j_1, k_1, i_2, j_2, k_2\right)^{(1)} = \left(i_1^-, j_1^+, k_1^{+-}, i_2^+, j_2^-, k_2^{+-}\right) \\ \left(i_1, j_1, k_1, i_2, j_2, k_2\right)^{(2)} = \left(i_1^+, j_1^+, k_1^{--}, i_2^-, j_2^-, k_2^{--}\right) \end{cases} \tag{6}$$

where:

$$\begin{cases} i_1^{\mp} = \frac{1}{2}\left(p_i \mp \frac{r}{p_i} \right) \\ i_2^{\pm} = \frac{1}{2}\left(p_i \pm \frac{r}{p_i} \right) \end{cases}, \qquad \begin{cases} j_1^{\pm} = \frac{1}{2}\left(p_j \pm \frac{m_4}{p_j} \right) \\ j_2^{\mp} = \frac{1}{2}\left(p_j \mp \frac{m_4}{p_j} \right) \end{cases} \tag{7}$$

$$\begin{cases} k_1^{\pm\mp} = \frac{1}{2}\left(p_k + \frac{\pm n_1 n_2 r \mp m_1 n_2 m_4}{n_1 m_2 p_k} \right) \\ k_2^{\pm\mp} = \frac{1}{2}\left(p_k - \frac{\pm n_1 n_2 r \mp m_1 n_2 m_4}{n_1 m_2 p_k} \right) \end{cases} \tag{8}$$

and appearing in the above equation (5), both the series are taken if and only if the inequality $m_4 \cdot r \neq 0$ holds true.

Appearing in the above formulas (6), (7) the variables m_1,n_1,m_2,n_2, are relatively prime numbers fulfilling the following identities:

$$\frac{m_1}{n_1} = \left(\frac{a}{b} \right)^2, \quad \frac{m_2}{n_2} = \left(\frac{a}{c} \right)^2.$$

Furthermore, the parameters appearing in formulas (6),(7) fulfill the following inequalities:

$$2 \leq p_i, p_k, p_j \leq 2N, \quad 1 \leq i_1, j_1, k_1, i_2, j_2, k_2 \leq N \tag{9}$$

$$0 \leq m_4, r \leq N^2 - 1 \tag{10}$$

$$m_4^2 + r^2 \neq 0 \tag{11}$$

Finally, the following divisibilities have to hold true:

$$p_i \mid r, \; 2 \left| \left(p_i + \frac{r}{p_i} \right), \right. \tag{12}$$

$$p_j \mid m_4, \; 2 \left| \left(p_j + \frac{m_4}{p_j} \right), \right. \tag{13}$$

$$\left(n_1 m_2 p_k \right) \left| \left(n_1 n_2 r \pm m_1 m_4 n_2 \right), \; 2 \right| \left(p_k + \frac{n_1 n_2 r \pm m_1 m_4 n_2}{n_1 m_2 p_k} \right). \tag{14}$$

3.1. Single-Thread algorithm 1

Based on the equalities (6)-(14) one can propose the following algorithm for finding the triples (3) with assumptions (4)-(5) [2] pp.152 (the *mod* operator denotes the rest from the integer dividing)

0) Input the values: m_1, n_1, m_2, n_2, N
1) Substitute:
   *m4_max := N*N-1*
   *r_max := N*N-1*
2) For each integer *m4* from the range *0..m4_max* repeat the steps 3-16
3) For each integer r from the range *0..r_max* repeat the steps 4-16
4) If (r<>0) Or (*m4*<>0) then repeat the steps 5-16
5) For each integer pi from the range 2..2**N* repeat the steps 6-16
6) If (r mod *pi* = 0) And ((*pi+q/pi*) mod 2)=0 then repeat the steps 7-16
7) For each integer *pj* from the range 2..2**N* repeat the steps 8-16
8) If (*m4* mod *pj* = 0) And ((*pj+m4/pj*) mod 2)=0 then repeat the steps 9-16
9) For each integer *pk* from the range 2..2*N repeat the steps 10-16
10) If ((*n1*n2*r-m1*m4*n2*) mod (*m2*n1*pk*)=0) And ((((*n1*n2*r-m1*m4*n2*) / (*m2*n1*pk)+pk*) mod 2)=0) then repeat the step 11
11) Perform the following calculations and substitutions:
   i1 := (*pi-r/pi*)/2
   i2:= (*pi+r/pi*)/2
   j1 := (*pj+m4/pj*)/2
   j2 : =(*pj-m4/pj*)/2
   k1 := (*pk+(n1*n2*r-m1*m4*n2)/(m2*n1*pk*))/2
   k2 := (*pk-(n1*n2*r-m1*m4*n2)/(m2*n1*pk*))/2
   S := *i1*^2 / *a*^2 + *j1*^2 / *b*^2 + *k1*^2 / *c*^2
12) If (i1,i2,j1,j2,k1,k2 >=1) And (i1,i2,j1,j2,k1,k2 <=N) then add the 7-th *(i1, j1, k1, i2, j2, k2, S*) to the set of the solutions.
13) If ((*n1*n2*r+m1*m4*n2*) mod (*m2*n1*pk*)=0) And *((((n1*n2*r+m1*m4*n2)/(m2*n1*pk)+pk*) mod 2)=0) And (r**m4*=0) then repeat the step14
14) Perform the following calculations and substitutions:
   i1 := (*pi+r/pi*)/2
   i2:= (*pi-r/pi*)/2
   j1 := (*pj+m4/pj*)/2
   j2 : =(*pj-m4/pj*)/2
   k1 := (*pk+(-n1*n2*r-m1*m4*n2*)/(*m2*n1*pk*))/2
   k2 := (*pk-(-n1*n2*r-m1*m4*n2*)/(*m2*n1*pk*))/2
   S := *i1*^2 / *a*^2 + *j1*^2 / *b*^2 + *k1*^2 / *c*^2
15) If (i1,i2,j1,j2,k1,k2 >=1) And (i1,i2,j1,j2,k1,k2 <=N) then add the 7-th *(i1, j1, k1, i2, j2, k2, S*) to the set of the solutions.
16) End of the algorithm 1

4. Parallel algorithm

The sole of the article is the adaptation of the algorithm 1 to the execution in the multiple-processor computer system. Finally, the presented algorithm is verified on a particular multiprocessor system.

4.1. Parallel algorithms theoretical basis

In the parallel algorithms analysis some performance factors are defined, giving information about the quality of the parallel algorithm, by comparison with the single-thread one. The basic notion is the so-called acceleration coefficient of a task of a size n after parallelization to the p single-thread processors. Based on the book [3] this coefficient is defined by (15):

$$S(n,p)=\frac{T(n,1)}{T(n,p)}, \tag{15}$$

where:

n - size of the task,

p - number of the single-thread processors,

$T(n,k)$ - time of the execution of an algorithm on p processors with n data size.

The inequality $S(n,p) \le p$ comes directly from equation (15). It is usable to give the relative acceleration, by comparison with the ideal acceleration given in percents. It is given by equality (16), based also on the book [3]:

$$S_{\%}(n,p)=\frac{S(n,p)}{p}\cdot 100\%. \tag{16}$$

Ideally, the relative acceleration coefficient is equal to 100%. All parallel algorithms can be divided into two general groups: synchronous and asynchronous. The algorithm 1 will be parallelized by the synchronous so called *loop spreading* method.

4.2. Problem analysis

It can be noticed that the algorithm 1 consists of 5 nested loops. The values of the calculated triples depend neither on the previous values of the loop parameters nor on the set of solutions that has been found so far. So the job can be divided into any independent sets. Moreover, working threads do not need mutual synchronization. That is an ideal situation for parallel programming giving maximal algorithm performance. The method of parallelization will be shown in the following items.

4.3. Master algorithm

As we have already said, the calculations do not depend on the previous values of the loop parameters so we can divide the job by choosing arbitrarily any of the least nested loop parameters m_4,r. In step 1 of the algorithm 1 we can see that the r parameter varies in the range $[0,N^2 - 1]$. So the master algorithm can have the following form:

- Input number of threads T,
- Divide the range $[0,N^2 - 1]$ into T equal sub-ranges R_i,
- Execute the slave algorithm in T independent threads, transferring to each one of the sub-ranges R_i.

4.4. Slave algorithm

The slave algorithm is similar to the base algorithm 1, but the initial step 0 is necessary and there is a sole modification of step 2:

0. Enter the sub-range $R_i = [r_{imin},r_{imax}]$
   ...
3. For $r \in [r_{imin},r_{imax}]$ repeat the steps 3-16
   ...

The remaining steps 1-16 without 3 are the same as in the algorithm 1.

4.5. Measured time performance

The figure below shows the received time of execution of the presented algorithm for two different values of the constraint: N = 32,28, on the 2xXeon 2.4 GHz PC workstation with hyper-threading[tm].

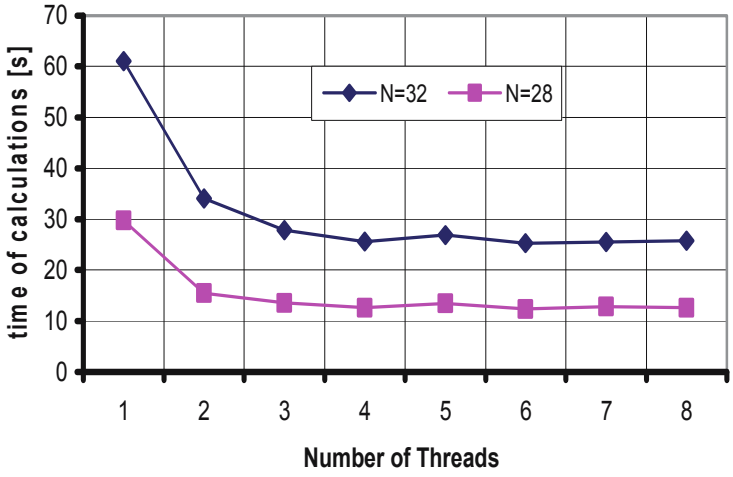

Fig. 1. Execution time

The next graph shows the acceleration coefficient (formula (15)).

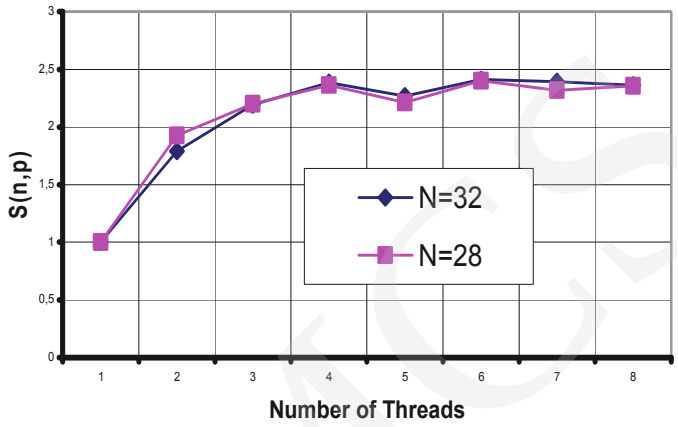

Fig. 2. Acceleration coefficient

And the last graph shows the relative acceleration (formula (16)).

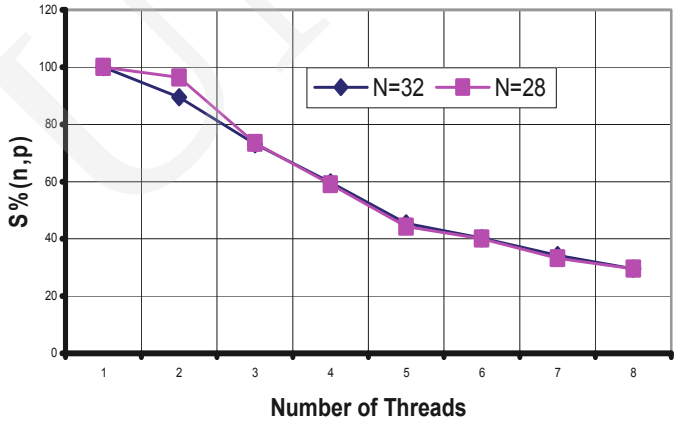

Fig. 3. Relative acceleration

5. Exemplary application of the presented parallel algorithms

The main application of the results from article [2] is the verification of the controllability of the parabolic infinite dimensional dynamical system [2] pp.158:

$$\frac{\partial x(z,t)}{\partial t} = \frac{\partial^2 x(z,t)}{\partial z_1^2} + \frac{\partial^2 x(z,t)}{\partial z_2^2} + \frac{\partial^2 x(z,t)}{\partial z_3^2} + \sum_{i=1}^{p} b_i(z) u_i(t), \qquad (17)$$

where the domain D of system (17) is the 3 dimensional rectangular prism with the sides as follows :

$$D = \left\{ z = \left(z_1, z_2, z_3 \right) \in R^3 : : z_k \in \left[0, \ a_k \right], a_k \in R_+, \, k = 1,2,3 \right\}. \tag{18}$$

This comes from the form of the eigenvalues of the differential operator of the system (17):

$$\lambda_{ijk} = -\pi^2 \left(\frac{i^2}{a^2} + \frac{j^2}{b^2} + \frac{k^2}{c^2} \right) \quad i, j, k = 1,2,3,... \ . \tag{19}$$

It can be seen that they are proportional to the sides of equation (3) considered in this paper. It can be proved that the necessary condition for controllability of system (17) is that the multiplicity of the eigenvalues (19) does not exceed the number of controls p.

6. Final conclusions

Fig. 1 proves that the task to make the algorithm 0 to work in a parallel environment is achieved. Fig. 2 shows that the acceleration coefficient is above two, which is a satisfactory achievement. The fastest slump of time execution can be observed up to 4 threads, further improvement of time is much smaller. This is because the tests were performed on a workstation with 4 logical processor units.

References

[1] Kalman R.E., *On the general theory of control systems*. Proceedings I IFAC Congress, London, (1960) 481.

[2] Respondek J., *The Applications of the Decomposition of the Real Numbers to the Linear Combination of Three Natural Number's Squares to the Investigations of The Controllability of an Infinite Dimensional System in the Three Dimensional Space Domain*. Archiwum Informatyki Teoretycznej i Stosowanej, 15(2) (2003) 143.

[3] Karbowski A., Niewiadomska-Szymkiewicz E., *Obliczenia równoległe i rozproszone*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, (2001), in Polish.