Medicon Engineering Themes
Volume 2 Issue 2 February 2022
Research Article

Estimation of Level of Algorithms Parallelization and Efficiency of Using Different Error-Correction Codes in Cloud Realizations of Parallel Computing Algorithms

Igor Melnyk[1]* and Andriy Luntovskyy[2]

[1]Igor Sikorsky KPI, Electronic Faculty, Department of Electronic Devices and Systems, Kyiv, Ukraine

[2]BA Dresden University of Cooperative Education, Computer Science Department, Dresden, Germany

***Corresponding Author:** Igor Melnyk, Igor Sikorsky KPI, Electronic Faculty, Department of Electronic Devices and Systems, Kyiv, Ukraine.

Received: January 22, 2022; **Published:** January 31, 2022

Abstract

The basic matrix algorithms for statement of parallel task in the cloud computing systems are considered in this paper. This algorithm is generally based on using the approach of arithmetic-logic relations and forming of recurrent matrixes. Possibility of using different error-coding methods for sending the request on parallelization task and receiving the answer is also considered. Comparative analyze of corrective ability of RS and convolutional codes is provided. As a result of provided investigations the relation for estimation the level of task parallelization is proposed. Choosing of convolutional codes for transmitting the data for parallelization the calculation task in cloud systems using of convolutional codes is recommended.

Keywords: Parallel Computing; Cloud Computing; Arithmetic-Logic Relation; Error-Correction; RS-Coding; Convolutional Coding

Introduction

Using of parallel computing in cloud realization is very important and significant problem in engineering and scientific researches now a day. Really, parallel computing's are widely used for choosing the best engineering solution in the numerical simulation by solving the complex equation systems [1]. Such approach is very effective in the different advanced branches of industry, like instrument-making, electronics, computer science, medicine and pharmaceuticals, as well as advanced technologies in metallurgy. The problem is that theoretical basis of parallel calculations and of forming parallelization task isn't developed enough, therefore forming of novel approaches in this aspect is necessary. The problem of choosing the best error-coding method for transmitting application data in cloud network is also very important and it need to find the best solution. Therefore, the subject of this paper is considering the novel approach for estimation the level of task parallelization, as well as choosing the best method of error-coding between RS and convolutional codes.

The Statement of Considered Problem

The standard approaches for estimation the level of parallelization of computer tasks is based on well-known Amdahl's law (1967) and Karp-Flatt metric (1990) [1, 2]:

$$T = 1; \qquad A_n = \frac{1}{(1-p)+\frac{p}{n}} \leq \frac{1}{1-p}; \qquad A_{\max} = \frac{1}{1-p};$$

$$A_{nk} = \frac{1}{(1-p)+\frac{p}{n}+kn}, k \to 0{,}1; p = \frac{\frac{1}{A_n}-\frac{1}{n}}{1-\frac{1}{n}}. \qquad (1)$$

where A_n or A_{nk} – speedup factor, or acceleration on *n* Central Processor Units (CPU) thread regarding to single one, A_{max} – maximal value of speedup v, *k* – negative influence of communication channels between CPUs.

The overview of the appropriate speedup models was systematically given in [1, 2, 13]. Among them, the following models are usually applied: Grosch's Law, Amdahl's Law, Barsis-Gustafson Law, Karp-Flatt Law and Sun-Ni Law. In some cases, Karp-Flatt Law (1990) can be used for estimation of parallelization grade.

Relations (1) are generally simple, but they aren't taking into account the complicity of parallelized task and the number of flows, which can be parallelized. Therefore, in the manual book [3] the alternative approach was proposed, based on using of arithmetic-logic relations and forming of recurrent matrixes. Basic principles of this approach, as well as corresponded formulas, are presented in this paper in the Section IV.

Another significant problem of providing parallel calculations in cloud network is choosing the best method of error-coding with taking into account the transferring of information by the wire and wireless channels [4]. Most of well-known methods of error-coding are considered in the monographs and manual books [5-10]. In the Section V the comparative analyze of RS-codes and convolutional codes is provided and it is proved, that for transferring data in nosed communication channels applying of convolutional codes is preferable.

One of the possible features is also Real-Time Capability, which can be estimated via the following expression:

$$T_r \geq (T_{tr} + T_{ex} + \Delta) \cdot (1+a) \qquad (2)$$

Where T_r – the given limit value for the reaction time of the computing system, T_{tr} – the necessary time for data transfer, T_{ex} – time of execution the formulated task, Δ– summarized process delay, *a* – average failure probability. By a repeated failure the given limit can even grow by the polynomial function ($1+a+a^2$).

Arithmetic-Logic Relations and Recurrent Matrixes

Arithmetic-Logic Relations

The basic definition of arithmetic-logic relations was given in the manual book [3]. There was pointed out, that arithmetic-logic relation is simply defined as a sum of production corresponded algebraic and logic functions and written as follows [3]:

$$\mathrm{AL}(x) = F_1(x)\cdot L_1(x)+F_2(x)\cdot L_2(x)+F_3(x)\cdot L_3(x)+F_n(x)\cdot L_n(x), \qquad (3)$$

Where $F_1(x)$ … $F_n(x)$ – arithmetic functions, $L_1(x)$ … $L_n(x)$ – logical functions, AL(x) – corresponded arithmetic logic relation, which created on the base of these arithmetic and logic functions.

Since the set of logical functions $L_1(x)$ … $L_n(x)$ must meet to the requirements of isolation and consistency, main properties of logic equations $L_1(x)$ … $L_n(x)$ have to be followed [3].

1. All numerical intervals, defined by the logical relations $L_1(x)$ … $L_n(x)$, have to bead join each other and not intersect.
2. The range of values of the variable x on the numerical axis should be fully and consistently described by the choosing set of logical functions$L_1(x)$, …, $L_n(x)$.

The typical location of numerical intervals, described by logic relations $L_1(x)$, …, $L_n(x)$ on the number axis, is presented on Fig. 1.

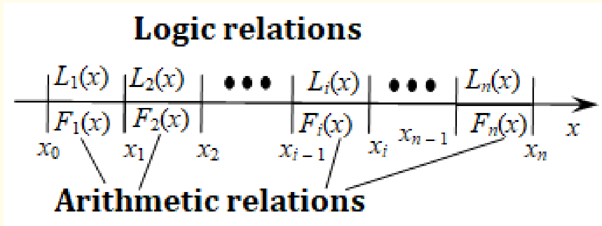

Figure 1: Illustration of the basic principle of forming arithmetic-logic equation.

On the base of arithmetic-logic relation definition and formula (3) formed the new definition of recurrent arithmetic-logic relation, where the based logic relations formed as a discrete function of natural numbers *n*. In this case relation (4) is rewritten as follows [3]:

$$\mathrm{AL}(n) = \left(\sum_{i=1}^{n_r} \mathrm{AL}(i)\right) \cdot (i \le n_r) +$$

$$+ \left(\textstyle\sum_{i=n_r+1}^{n} \mathrm{F_j}\left(\mathrm{AL(j)}\right)\Big|_{\mathrm{j}=i-n_r}^{i}\right) \cdot (i > n_r), \quad (4)$$

If the arithmetic-logic functions AL(*i*) and AL(*j*) in equation (4) are the vectors of the same length, final created complex structure AL(*n*) is usually considered as recurrent matrix [3].

Recurrent Matrixes

In the simple form the recurrent matrix is defined as the set of corresponded equations for defining the elements of matrix rows [3]:

$$\mathrm{M}_{<1>}=v_1,\ \mathrm{M}_{<2>}=v_2,\ \ldots,\ \mathrm{M}_{<n>}=v_n;\ \mathrm{M}_{<i>}=F(i,\ \mathrm{M}_{<i-1>},\mathrm{M}_{<i-2>},\ldots,\mathrm{M}_{<i-n>}), \qquad (5)$$

Where v – vectors, $\mathrm{M}_{<i>}$ – row of matrix with number *i*, F– vector-function, which defined the function for calculation the matrix elements [3]. For the particular tasks F is defined as multidimension arithmetic-logic function with the following components F_1, F_2, ..., F_n. In the form of mathematic relations vector F is written as follows [3]:

$$\mathrm{F} = \{F_1, F_2, \ldots, F_n\};\ F_1 = \mathrm{AL}(1),\ F_2 = \mathrm{AL}(2),\ F_n = \mathrm{AL}(n). \qquad (6)$$

Generally, the main conception of forming the recurrent matrix is using for all rows of matrix the same set of basic functions F_1, F_2, ..., F_n, defined by equations (6).

Described basic principle of forming the recurrent matrixes is illustrated at the Fig. 2 [1, 3].

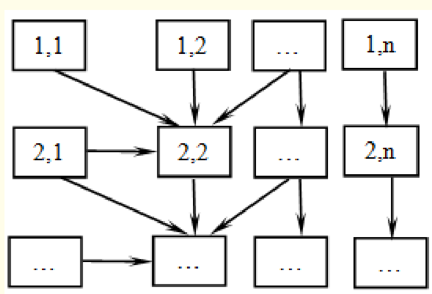

Figure 2: Illustration of the basic principle of forming the recurrent matrixes.

In the manual book [3] was pointed out and proved, that the recurrent matrixes approach is the is a real alternative to structured programming techniques and that the basic algorithm of discrete and calculation mathematic can be realized in arithmetic-logic relations and recurrent matrixes approach.

Considering here the possibilities of using this approach for estimation the level of algorithm parallelization.

Using of Recurrent Matrix Approach for Estimation the Level of Algorithm Parallelization

Basic approach for analyzing the speedup factor of considered algorithm with using the matrix approach is dividing the connections between matrix elements into separate independent flows [1]. There was pointed out, that with including the sequential connections between the elements of matrix in the row the parallelization of such algorithm is impossible. Corresponded structure of matrix is presented in Fig. 3. At this figure the hierarchical and sequential connections between the matrix elements are marked separately.

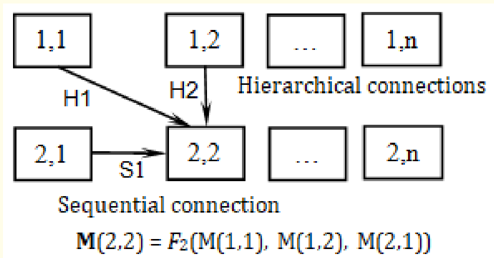

Figure 3: Illustration of the hierarchical and sequential connections at the recurrent matrix.

If the structure of recurrent matrix, corresponded to considered algorithm, included only hierarchical connections, data threads T_1, T_2 ..., T_n for all matrix elements can be considered separately. In this case with assumption, that all connections in threads have the similar complicity for calculation, speedup factor p of considered algorithm is estimated by the following relation [1]:

$$p_1 = 1 - \frac{\max_{i=1...n} N(T_i)}{\sum_{i=1}^{n} N(T_i)}, \qquad (7)$$

Where N – number of connections for considered matrix element.

Equation (7) can be simply modified for the threads with different complicity of calculations, and, corresponding, different time of treatment by CPU. In such conditions the complicity of elementary connections between elements F_j have to be analyzed and the factor of complicity of each elementary operation F_j is defined relatively to the simplest operation by factor α_j. For such condition equation (7) rewritten as:

$$p_2 = 1 - \frac{\max_{i=1...n} \sum_{j=1}^{k_i} \alpha_j F_j}{\sum_{i=1}^{n} \sum_{j=1}^{k_i} \alpha_j F_j}. \qquad (8)$$

Usually using of relation (8) given the more realistic value of speedup factor p, than using of equation (7). In any case, equations (7) and (8) allows estimate the level of parallelization the algorithm in the cloud network calculations.

Estimation the Efficiency of Using Error-Coding Methods for Cloud Network Calculations

Estimation of Probability of Error in the RS Codes

The RS-codes are multiposition codes, which structure is based on the Galois Field theory [5-10]. The probability of error in RS-codes P_e, corresponding to the basic principles of combinatorial analysis and probability theory, is defined as follows [11, 12]:

$$P_e = P(\tau > \theta) = \sum_{\tau=t+1}^{n} C_n^{\theta} (1 - (1 - p_b)^m)^{\theta} ((1 - p_b)^m)^{n-\theta}, (9)$$

$$t < n.$$

Where t – number of detected errors, n – number of symbols in code combination, θ – number of distortion of bits in code, m – order of Galois Field, p_b – the probability of distortion 1 bit in symbol, C_n^{θ} – the number of combinations from n elements by θ.

The maximal number of corrected symbols in RS-codes t_{max} is defined by the following equation [6, 7]:

$$t_{\max} = \begin{cases} 2^{m-1} - \frac{k}{2} - 1, & \text{when the value } k \text{ iseven;} \\ 2^{m-1} - \frac{k+1}{2}, & \text{when the value } k \text{ is odd.} \end{cases} \quad (10)$$

Where k– number of bits in coded word.

Dependences of error probability in RS-code on the probability of distortion one-bit $P_e(p_b)$ for different order of Galois Field m and number of detected errors t are presented at Fig. 4. It is clear from obtained dependences, that even with middle level of bit-error probability p_b> 0.1 the probability of distortion RS-code word is very high, namely, $P_e \approx 1$. Therefore, using of RS-code in noised wire and wireless communication channels of cloud computer systems isn't recommended. Such type of codes is suitable only for stable operated un-noised apparatus, such as controllers of hard discs in PC or optic disk drivers [5].

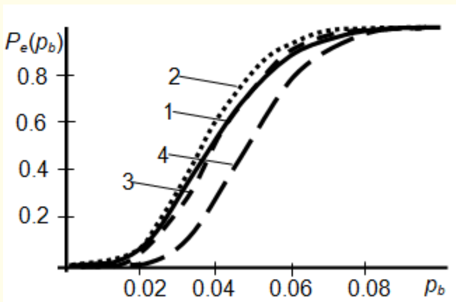

Figure 4: Dependences of probability of error in RS-codes on probability of bit error. 1 – m = 5, t = 5; 2 – m = 6, t = 5; 3 – m = 6, t = 7; 3 – m = 6, t = 10.

Estimation of Probability of Error in the Convolutional Codes

Defining the probability of error in the convolutional codes is more sophisticated problem, than the same task for RS-codes. It is due to the more complex structure of convolutional codes, which correction functions are generally based on the theory of events prediction. The structure of convolutional codes is complexly described by the diagrams of Finite-States-Machine (FSM) and its polynomial transmitting function $T(D, L, N)$, where D – the Hemming distance between zero and waiting codes sequences, L – the counter of transmissions between the start and current FSM state, and N – the mark to transmissions, which corresponded to the input signal 1 [5-10]. Minimal number of unrecognized errors in convolutional cede is always corresponded to the minimal power of variable D at the polynomial presentation of transmitting function. For example, for standard potential coding, like Alternative Mark Inversion (AMI) [4], the relation for defining $P_e(p_b)$ is written as follows [5, 10]:

$$P_e(p_b) \leq \frac{dT(D,N,L)}{dN}\Big|_{N=1,L=1,D=2\sqrt{p_b(1-p_b)}}. \quad (11)$$

For example, solving equation (10) for the convolutional code with basic parameters $\left(5,\frac{1}{3}\right)$, where k = 5 – the number of bits in the shift register and 1/n=1/3 redundancy factor of code, transferring function $T(D, L, N)$ is defined by analyzing the FSM structure as follows [5]:

$$T(D,L,N) = -L^6N^4D^7 - 5L^7N^5D^8 - \\ -25L^8N^6D^9 - 125L^9N^7D^{10} - \cdots. \quad (12)$$

Therefore, the maximal value of unrecognized errors at the code construction with such structure is d_f= 7. Substituting of equation (12) into equation (11) giving following result:

$$\left.\frac{dT(D,N)}{dN}\right|_{N=1,L=1} = D^7 \frac{10D-3}{(5D-1)^2}. \qquad (13)$$

After substituting in relation (13) the value

$$D = 2\sqrt{p_b(1-p_b)} \qquad (14)$$

From basic equation (11), we obtaining the dependence $P_e(p_b)$ in complete form as follows:

$$P_e(p_b) \le \frac{2560\left(p_b(1-p_b)\right)^4 - 384\left(p_b(1-p_b)\right)^3\sqrt{p_b(1-p_b)}}{\left(10\sqrt{p_b(1-p_b)}-1\right)^2}. \qquad (15)$$

The dependence $P_e(p_b)$, obtained with using the set of equations (12 –15), is presented at Fig. 5.

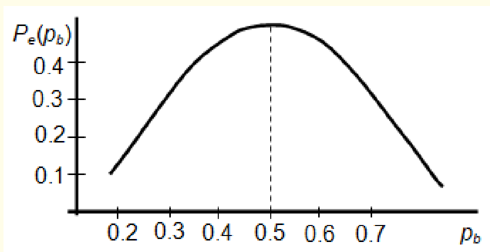

Figure 5: Dependence of the probability of error in convolutional code on probability of bit error.

It is clear from graphic dependences, presented at Fig 4 and Fig. 5, that for the convolutional code with parameters $\left(5,\frac{1}{3}\right)$ significantly grates error-correction ability, than for the constructions of RS-codes with different parameters *m* and *t*, is provided. The maximum value of error at the convolutional code $\left(5,\frac{1}{3}\right)$ obtained for the value of bit-error $p_b = 0.5$, which is due by the same probability of receiving the correct and wrong signals. If the probability of wrong signal became higher, the convolutional code intelligent system automatically converted the wrong signal to correct one [5-10]. Therefore, the probability of error in the convolutional codes in conditions of high value of bit error is generally smaller. By this reason namely such type of codes is recommended to applying in the communication systems with noised channels [5-10], therefore using these codes in the cloud computing systems also can be considered as the best solution.

Conclusion

In the paper is shown that the approach of recurrent matrixes is very effective to estimation the level of parallelization of computational algorithm in cloud network calculations. This approach based on analyzing the connections between recurrent matrix elements and dividing the hierarchical connections into independent thread with the same or different complicity. After that the level of parallelization is estimated with using relations (7) or (8).

Possibility of using error-correction coding for transferring data in the cloud network communication channels is also analyzed. Alternative consideration of RS-codes and convolutional codes have been provided. Provided analyze shown, that applying of RS-codes is preferable in the un-noised apparatus and it using in noised communication channels usually isn't effective. In the contrary, convolutional codes constructions can be successfully applied in the noised communication channels even with high probability of bit-error. For estimation the probability of unrecognized error at the convolutional codes words the FSM theory and basic equations (11) for FSM transferring function, written at the polynomial form, is used. As an example, the result of such estimations for convolutional code with parameters $\left(5,\frac{1}{3}\right)$ are presented at Fig. 5.

The results of this work are very significant and important for the experts in the branch of creating specific computer software for realizing the distributed parallel calculations in the cloud systems. For estimation the basic level of algorithm parallelization the simple relations (7) or (8) can be used, and for choosing the best convolutional code construction the FSM theory and equation (11) can be successfully applied. For providing polynomial operations with the FSM transferring function $T(D, L, N)$modern CAD systems for mathematic transforming, like MatLab, MAPLE and Mathematica can be effectively used.

References

1. Luntovskyy AO and Melnyk IV. "Simulation of Technological Electron Sources with Use of Parallel Computing Methods". XXXV IEEE International Scientific Conference "Electronic and Nanotechnology (ELNANO)", Conference Proceedings, Kyiv, Ukraine (2015): 454-460.
2. Luntovskyy AO. "Technologii Rozpodilenyh Programnyh Dodatkiv. Monography". Published in Ukrainian Language. Kyiv, State University of Information and Communication Technologies DUIKT (2010): 452.
3. Melnyk IV. Systema Naukovo-Technichnyh Rozrahunkiv MatLab ta ii Vykorystannia dlia Rozviazannia Zadach z Electroniky. Tom 2. Osnovy Programuvannia ta Rozviazuvannia Prykladnyh Zadach. Published in Ukrainian Language. Kyiv, University Ukraina (2009): 327.
4. Irvine J and Harle D. "Data Communication and Networks: An Engineering Approach". John Wiley & Sons, LTD (2001).
5. Sklar B. Digital Communications: Fundamentals and Applications, 2nd Edition. University of California, Los Angeles (2001).
6. Berlekamp ER. Algebraic Coding Theory. McGraw-Hill Book Company. New-York (1968): 478.
7. Lathi BP. "Modern digital and analog communication systems". Holt. Rinehart and Whinston Inc (1989): 720.
8. Birkhoff G. and Bartee TC. Modern Applied Algebra. McGraw-Hill Book Company. New-York (1970): 471.
9. Peterson WW and Weldon EJ. Error-Correcting Codes, second edition. MIT Press (1972): 560.
10. Blahut R. Theory and Practice of Error Control Codes. Addison-Wesley Press (1983).
11. Gubner JA. Probability and random processes for electrical and computer engineers. Cambridge, UK: Cambridge University Press (2006).
12. Anderson JA. Discrete Mathematics with Combinatorics. Second Edition. University of South Carolina-Spartanburg (2004).
13. Luntovskyy A and Spillner J. "Architectural Transformations in Network Services and Distributed Systems". Service Vision. Case Studies, Monography (2017): 344.

Volume 2 Issue 2 February 2022