RECOVERY SCHEME OF DISTRIBUTED COMPUTING BASED ON IDEAL RING RELATIONSHIP

Riznyk O.Y., Povshuk O.V., Vyushkova K.V

National University "Lviv Polytechnic", 79013, Lviv, vul.Bandery 12, riznyk@meta.ua

Introduction

Distributed computing systems are much more protected from the failure of multiple computers that are being used and have high productivity by sharing the computing load. Ideally, the computing load should be evenly spread among all computers. When one or more computers go down, the load from these computers should be redistributed to other working machines in the group of distributed computing systems. This redistribution is defined by a recovery scheme. The recovery scheme should distribute the computing load equally between working computers as much as possible, even when there are computer failures of the unfavorable combination. There is a need to find optimal recovery schemes for any number of computers in a distributed computing system.

Problem

The goal is to find the optimal recovery schemes that must be quickly calculated for a large number n of computers and work better than any existing schemes that are already known. Let's consider a cluster with n number of identical computers. There is one process on each computer. The work is evenly divided between these n processes. There is a recovery list associated with each process. This list defines where the process should start again if the current computer is turned off. The process moves back as soon as idle computer returns to its working condition.

So in such case there is a need to receive a number, which you can find the longest sequence of positive integers for, so that the sum of the sequence is less than or equal to the sum of numbers, and the sum of all sequences should be unique as well. This problem is equivalent to the problem of defining the Golomb ruler provided that the sum of the sequence is less than or equal to L_n, and is equivalent to the problem of finding an ideal ring bundle (IRB) on condition that the sum of the sequence equals exactly S_n, so that we can use the results of Golomb rulers and ideal ring bundles accordingly. Golomb ruler is the set of positive integers, arranged in ruler sections, so that the distance between any two of them is unique. In other words, when following along the ruler, we cannot find such two numbers with subtraction between them repeated twice [2]. Maximum number of pairs that can be defined from distances n between neighboring ruler's points are determined by the formula:

$$L_n = \binom{2}{n} = \left(\frac{n(n+1)}{2}\right) \tag{1}$$

Solving the problem

Simple ideal ring bundle (IRB) is a sequence $K_n = (k_1, k_2, ..., k_n)$ of numbers on which all possible circular sums are drawn by natural numbers $1, 2, ..., S_n$, where:

$$S_n = n(n-1)+1 \tag{2}$$

The recovery list is obtained by adding the values of sequences, and is a sequence of partial sums. The first part of recovery lists consists of the sum of elements from the numeric ruler-bundle, or IRB, so that the sum of the optimal sequence length is less than $L_n + 1$ for Golomb rulers or $S_n + 1$ for IRB. The remaining part of the recovery list is filled with the rest of the numbers (computers) up to the amount of Golomb rulers L_n or the sum of IRB S_n. The second part of the recovery schemes based on ideal ring bundles is being built as a row of consecutive numbers that do not match the weights of sums of the elements from ideal ring bundles.

For example, IRB (1, 3, 2, 7) with parameters $N = 4$ and $S_N = 13$ the first part of the recovery scheme corresponds to the following sums of IRB items: 0, 0 + 1 = 1, 1 + 3 = 4, 4 + 2 = 7, 7 + 7 = 14, and the second part of the scheme corresponds to the number, which are missing in the first part 2 3, 5, 6, 8, 9, 10, 11, 12, 13.

Let's compare proposed model of recovery scheme with the best existing example based on Golomb rulers (numerical ruler bundles). Let's assume $V_{0,i}$ is a recovery list of Golomb ruler for the zero process in a cluster with j computers. Then it will contain numerical ruler-bundle with the sum of less than or equal j and the rest numbers (computers) up to the $j-1$For example, for the zero process in a cluster of 12 computers: $V_{0,12}(0)=\{1,4,9,11,2,3,5,6,7,8,10\}$. All other recovery lists exit from $V_{0,j}(0)$ by using $V_{i,j}(x)=(V_{0,j}(0)+i)\bmod(j+1)$ for all $i \le j$. Let $V_{0,j}$ is the IRB recovery list for zero process in a cluster with j computers. Then it contains IRB with the sum equal to j and the rest numbers (computers) to the j-1. For example, the zero process in a cluster of 14 computers, we have: $V_{0,14}(0)=\{1,4,6,13,2,3,5,7,8,9,10,11,12\}$. Other recovery lists for the processes x we have from $V_{0,j}(0)$ by using $V_{i,j}(x)=(V_{0,j}(0)+i)\bmod(j+1)$ for all $i \le j$.

By using numerical scheme of ruler-bundle optimal behavior of only 27 damaged computers can be guaranteed, where $L_{27}=553$, as for larger n we don't know yet whether the corresponding numeric ruler bundles are optimal or not. By using IRB schemes a close to optimal behavior for any number of damaged computers can be guaranteed, as Singer's IRB exist for n provided that:

$$n = p^{\alpha}+1, \quad (3)$$

where p - prime number, α - an integer.

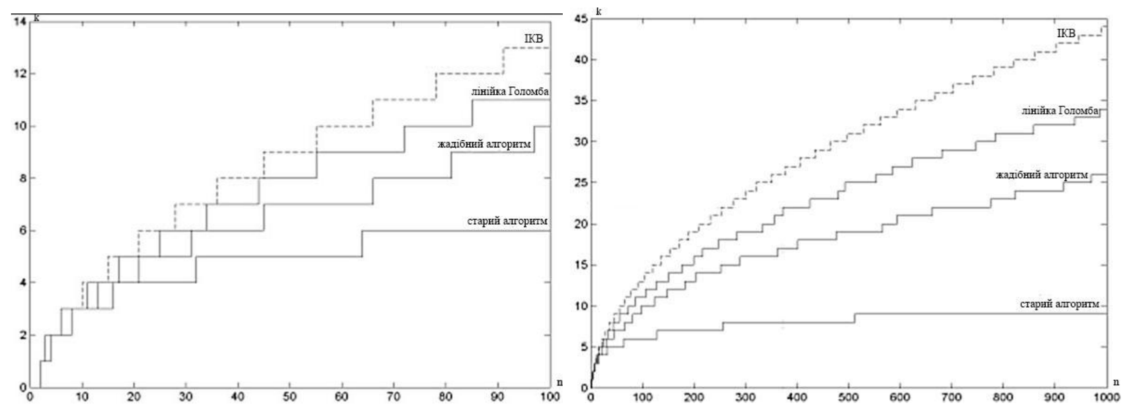

Fig.1. Recovery scheme based on IRB, Golomb rulers, greedy algorithm, the old algorithm

On Fig. 1 recovery schemes that use IRB, Golomb rulers, greedy algorithm (an algorithm that takes the best decision based on available data at the current stage, hoping to eventually get the optimal solution), the old algorithm (an algorithm that is some function of the number of computers) are being compared. Execution is defined as the number of computers that went down, and based on them construction of optimal load distribution can be ensured [3]. Fig. 1a shows that with the increasing number of n computers in a distributed computing system usage of recovery IRB scheme is more appropriate for more computers n, for example, for $n=100$ IRB recovery schemes guarantee an optimal solution if $k=13$ computers went down, usage of Golomb rulers guarantee an optimal solution if $k=11$ computers went down, greedy algorithm ensures optimal behavior if $k=10$ computers were broken and old algorithm ensures optimal behavior only for $k=6$ computers [1. 3]. Fig. 1b shows recovery schemes on IRB for n = 1000, which guarantee an optimal solution if $k=44$ computers went down, schemes on Golomb rulers, which guarantee an optimal solution if $k=34$ computers went down, greedy algorithm ensures optimal behavior if $k=26$ computers were broken, and old algorithm ensures optimal behavior for only $k=9$ computers [1, 3].

Conclusions

n identical computers, which normally perform each process under the same conditions were considered. All processes perform the same amount of work to the nearest unit.

Recovery scheme of distributed system that guarantees optimal load distribution in the worst case, when k computers went down. This is the optimal recovery based on criteria of the even load for values n and k based on IRB.

1. VV Ripper Combinatorial Synthesis of optimal systems. - Lviv, 1989.

2. Ripper OJ, Balych BI Using of numerical ruler-bundle to encode information. Institute Bulletin "Computer Science and Information Technology," 2006. s.62-64.

3. Klonowska, K., Lundberg, L., Lennerstad, H .: Using Golomb Rulers for Optimal Recovery Schemes in Fault Tolerant Distributed Computing, in Proceedings of 17th International Parallel & Distributed Processing Symposium IPDPS 2003, Nice, France, April 2003, pp. 213.