ANNALS of Faculty Engineering Hunedoara – International Journal of Engineering
ISSN: 1584 - 2673

EDUCATIONAL SOFTWARE FOR ANALYSIS OF PARALLEL ALGORITHMS USING PRAM MODEL

Alexandru-Ştefan OPREAN, Tatiana-Elena DUȚĂ, Manuela PĂNOIU

Department of Electrical Engineering and Industrial Informatics,
"Polytechnic" University of Timisoara, ROMANIA

Abstract
This paper presents visual interactive software which shows through simulation the parallel access memory for PRAM model. The software was implemented in Java. It was also performed a comparative study between a classic sequential algorithm and a parallel algorithm in terms of execution times.
Keywords:
Algorithm visualization, parallel access memory, Java, PRAM model

1. INTRODUCTION

The purpose of the theoretical models for parallel computation is to give frameworks by which we can describe and analyze algorithms. These ideal models are used to obtain performance bounds and complexity estimates. One of the models that have been used extensively is the parallel random access machine (PRAM) model [1]. A PRAM consists of a control unit, a global memory shared by p processors, each of which has a unique index as follows: P1 , P2 , . . . , Pp . In addition to the global memory via which the processors can communicate, each processor has its own private memory.
Based on the different modes for read and write operations, the PRAM can be further divided into the following subclasses:

- EREW PRAM: Access to any memory cell is exclusive. This is the most restrictive PRAM model.
- ERCW PRAM: This allows concurrent writes to the same memory location by multiple processors, but read accesses remain exclusive.
- CREW PRAM: Concurrent read accesses are allowed, but write accesses are exclusive.
- CRCW PRAM: Both concurrent read and write accesses are allowed.

2. DESCRIPTION OF THE SOFTWARE

The target of this application is to help students in understanding the parallel and sequential algorithms. Another goal for this project is to show a comparative study between a classic sequential algorithm and a parallel algorithm in terms of execution times.

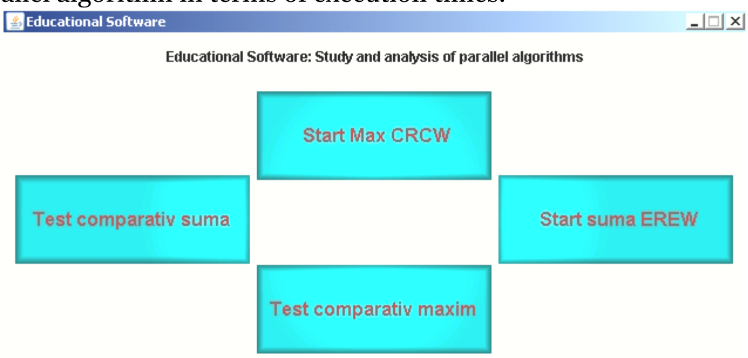

Figure 1. Main Menu

The application was implemented in Java as independent application. The application can easily convert in a Java applet. For simulation the It is made so that it possible for students and beginners to use the application. From the main menu that you can see in Figure 1 you can choose one of four options of the software:

- Description of the PRAM CRCW maximum search algorithm
- Description of the PRAM EREW sum algorithm
- A comparative study between the PRAM CRCW maximum search algorithm and the classic sequential.
- A comparative study between the PRAM EREW sum algorithm and the classic sequential variant.

By selecting any of these options from the application interface a new window will open which will contain each part of the application.

Buttons are simulations realized with the help of images created with the 3D Button program.

2.1. Maximum search CRCW Simulation

This is the main frame for the Maximum Search algorithm simulation. This presents the steps taken by the Maximum Search CRCW algorithm.

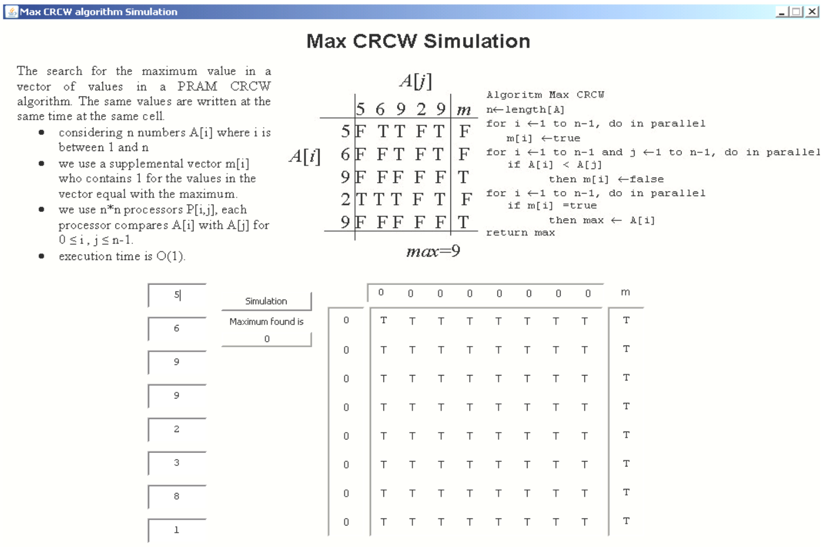

Figure 2. Maximum Search Simulation

The algorithm which searches for the maximum in a vector of values with the aid of the Maximum search CRCW (Concurrent Read Concurrent Write) algorithm is presented and simulated in our application.

The theory behind the algorithm is presented on the upper part. The simulation of the algorithm is made so that the user can input some 8 values for the vector. By pressing the Simulate button the process begins. First the top and left vectors are initialized with the values introduced by the user. Then the middle matrix is initialized with the start (F=False) value. The m vector is initialized also.

After the first phase the simulation of the algorithm is started. The middle matrix is obtained by the tests between A[i] and A[j]. Meanwhile the m vector is obtained. After the m vector is found, the maximum value of the vector is found: where m[i] is true, v[i] is the maximum.

2.2. Summing algorithm EREW Simulation

This is the frame which presents the simulation for the summing EREW algorithm.

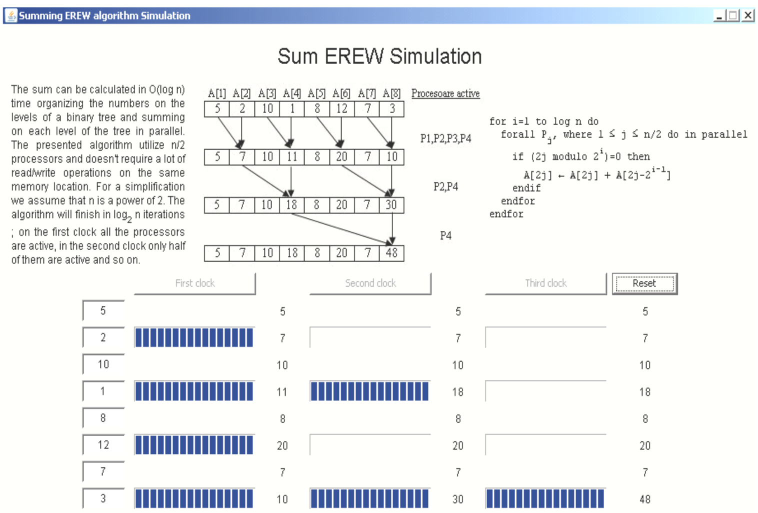

Figure 3. Summing algorithm Simulation

To simulate the summing algorithm in parallel the user must input some start values for the vector to be summed. To start the algorithm we first click the First Clock button. After the first clock of the simulation (the simulation is done by filling 4 progress bars showing that 4 processors are active) we can pass to the second clock of the simulation. In this clock, only half of the processors active in the previous clock are active (showed by only two progress bars). At the last clock only one processor is active. After this clock, the sum of all 8 values is in the last field on the down right part of the frame.

© copyright FACULTY of ENGINEERING - HUNEDOARA, ROMANIA

2.3. Comparative Study for the maximum search algorithms

In this frame we want to show the execution times for the classic sequential algorithm and the parallel CRCW algorithm. For this study we have made it possible for us to choose the length of the vector for the search and the maximum value of each value of the vector.

To run the comparative test of the two algorithms the user must insert the length of the vector and the generation interval for the values (this generation interval is found between 0 and 2 at the power of the value entered). In case the introduced data is wrong the application will show an error message. If we will not introduce a number in one of the input box, the application will show the next error message.

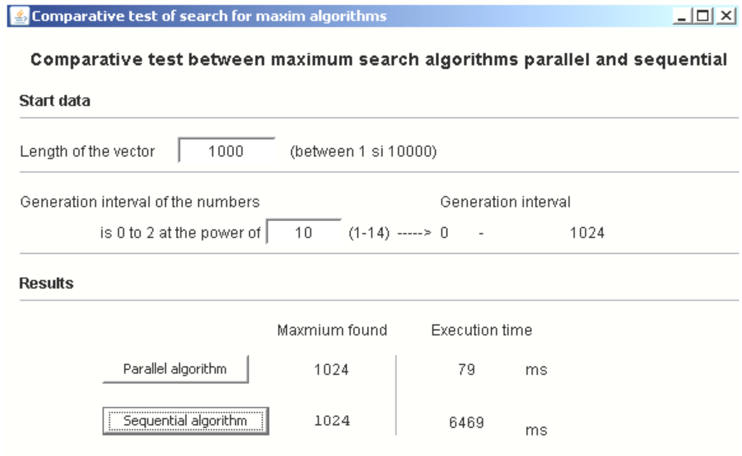

Figure 3. Comparative Study for the maximum search algorithms

Figure 4. Error message

In case of a value lower of larger than the provided interval the application will show a attention message. In case the user didn't fill the start values and he whishes to execute the parallel or sequential algorithm the application will show another attention message.

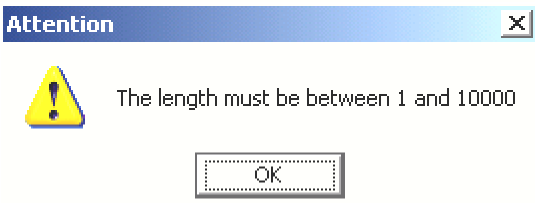

Figure 5. Attention message

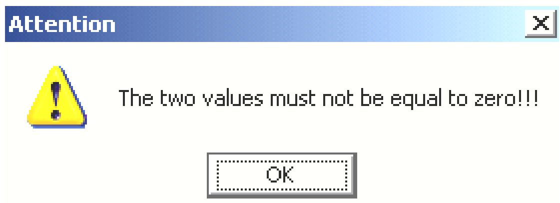

Figure 6. Attention message

2.4. Comparative study for the summing algorithms

This frame presents a comparative study for the execution times of the summing algorithms, the sequential and the parallel one.

To start the comparative test between the two algorithms the user must fill in the initial data. If the values are not filled in, the application sends the user error or warning messages.

After the input parameters are filled in the two algorithms are executed and the application calculates and shows the execution times for the two algorithms. To verify if the two algorithms are working on the same set of numbers we can compare the two sums displayed. The results of some comparative tests are show in Table 1 for finding maxim

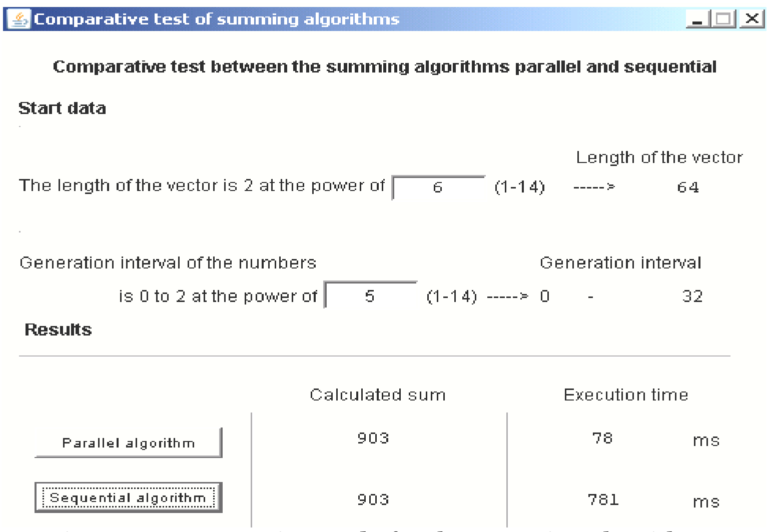

Figure 4. Comparative study for the summing algorithms

© copyright FACULTY of ENGINEERING - HUNEDOARA, ROMANIA

algorithm and in Table 2 for summing algorithm. The computer used for the test was an Intel Pentium Mobile, processor frequency 1,7 GHz, 1024 MB RAM, operating system Microsoft Windows Xp SP2.

Table 1 The run time for CRCW maximum search algorithm

Input data	Nr. crt of test	Parallel algorithm	Sequential algorithm
Vector length: 4096 (2^{12}) Values interval: 0 – 256 ($0 – 2^8$)	1.	1782 ms	44344 ms
	2.	1622 ms	44384 ms
	3.	1682 ms	44664 ms
	4.	1712 ms	47618 ms
	5.	1683 ms	48530 ms
Vector length: 4096 (2^{12}) Values interval: 0 – 256 ($0 – 2^8$)	1.	110 ms	721 ms
	2.	130 ms	771 ms
	3.	150 ms	721 ms
	4.	121 ms	741 ms
	5.	70 ms	731 ms

Table 2 The run time for EREW summing algorithm

Input data	Nr. crt	Parallel algorithm	Sequential algorithm
Vector length: 128 (2^7) Values interval: 0 – 256 ($0 – 2^8$)	1.	10 ms	731 ms
	2.	10 ms	771 ms
	3.	10 ms	751 ms
	4.	15 ms	721 ms
	5.	10 ms	761 ms
Vector length: 1024 (2^{10}) Values interval: 0 – 512 ($0 – 2^9$)	1.	30 ms	6740 ms
	2.	20 ms	6810 ms
	3.	10 ms	6829 ms
	4.	40 ms	6900 ms
	5.	31 ms	6659 ms

3. CONCLUSIONS

Analyzing the tests results it can be observe that form point of view of the execution time, the parallel algorithms are more efficient than the sequential algorithms, but the total cost of the parallel algorithm are higher in terms of processors numbers. Since a good sequential algorithm can sum the list of n elements and also find the maximum of the elements in O(n), these algorithms is not cost optimal.

However, the PRAM model is a very useful model for study the parallel access to the memory, and the present application can do this an interactive manner, so that the students will better understand these concepts.

REFERENCES

[1.] *Hesham El-Rewini, Mostafa Abd-El-Barr, Advanced Computer Architecture and Parallel Processing, Whiley+Interscience, 2005*

[2.] *Herlo D. - Metodologia educațională, Editura Universității „Aurel Vlaicu", Arad, 2000*

[3.] *Ioan Jurca – Programarea orientată pe obiecte. Limbajul Java, Editura de Vest, Timişoara, 2005*

[4.] *Ştefan Tanasă, Cristian Olaru, Ştefan Andrei – Java de la 0 la expert, Editura Polirom, Iaşi, 2003*

[5.] *Motogna, S. – Metode şi tehnici de proiectare a algoritmilor, Editura Universității Babeş-Bolyai, Cluj Napoca, 1999*

© copyright FACULTY of ENGINEERING - HUNEDOARA, ROMANIA