RESEARCH ARTICLE OPEN ACCESS

Evaluating and Improving the Performance of Cryptosystem with Low Latency Framework on Multi-Core System

Kunal Bhangale, Chaitanya Deshmukh, Shubham Jadhav, Tejas Kshirsagar
Department Of Computer Engineering, DYPIEMR, Akurdi.

Abstract:

The technique of using workload dependent dynamic power management (i.e. variable power and speed of processor cores according to the current workload) to improve system performance and to reduce energy consumption is investigated. Typically, the power supply and the core speed are increased when there are more tasks in a server, such that tasks can be processed faster and the average task response time is reduced. On the other hand, the power supply and the core speed are decreased when there are less tasks in a server, such that energy consumption can be reduced without significant performance degradation. Several speed schemes are implemented and it is demonstrated that for the same average power consumption, it is possible to design a multicore server processor with workload dependent dynamic power management, such that its average task response time is shorter than a multi-core server processor of constant speed. Information security is a challenging issue in today's technological world. There is a demand for a stronger encryption which is very hard to crack. Earlier many researchers have propsed various encryption algorithms such as AES, DES, Triple DES, RSA, Blowfish etc. Some of them are most popular in achieving data security at a great extent like AES and Blowfish. But, as security level is increased, the time and complexity of algorithm is also increased. This is the major cause of decreasing the speed and efficiency of the encryption system. In this paper we have implemented a new encryption algorithm "Byte – Rotation Encryption Algorithm (BREA)" with "Parallel Encryption Model" which enhances the security as well as speed of the encryption scheme

Keywords **— Computer System Organization, Computer Communication Networks, Distributed System, Distributed applications, Distributed databases, Network operating systems, Distributed file systems**

I. INTRODUCTION

For proposing an online resource management frame-work that maximizes profit ratio while minimizing energy expenses by exploiting the distributed task elasticity and price heterogeneity. This is done by reducing the duration during which servers need to be left ON and maximizing the monetary revenues when the charging cost for some of the inelastic tasks depends on how fast these tasks complete, while meeting all resource requirements. The power supply and the core speed are increased when there are more tasks in server, such that tasks can be processed faster and the average task response time is reduced. It is possible to design a multicore server processor with workload dependent dynamic power management, such that its average task response time is shorter than a multicore server processor of constant speed (i.e., without workload dependent dynamic power management). Byte Rotational Algorithm(BRA) provides more security and takes smallest amount of time for transfer file . This algorithm can apply on different types of files like text, image, audio, video files.

Comparative studies conducted using UCI repository data traces show the effectiveness of our implemented framework in terms of improving resource utilization, reducing energy expenses, and increasing profits ratio by calculating memory and bandwidth with increasing speed. The process elasticity is exploited on heterogeneous environment in a distributed system. Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible. Various approaches are considered like serial, parallel and hybrid approaches. Accordingly, profit ratio is calculated. Process mining is taken as a task to calculate the profit ratio. Resources that are considered are CPU, Bandwidth, Time and Temperature and Memory. Tools used to calculate the profit ratio of CPU, Bandwidth, Time and Temperature and Memory is CPU-Z and HW-Monitor. The tasks involved are independent on each other. Profit ratio is calculated of factors i.se CPU, Bandwidth, Memory, Time and Temperature of systems with different processors.

II. RELATED WORK

Author V. Maitri, Dattatray S. Waghole, Vivek S. Deshpande implemented BRA Algorithm in network security.In this work it is seen that BRA algorithm is better algorithm over the traditional algorithms.In this work they had proved that Bra algorithm is better than AES algorithm by giving examples of textfiles,image files ,audio files etc.

Author BingChun Chang implements algorithm

For the Time Improvement for Two Thresholds Two Divisors Algorithm.In this, techniques for time improvement of TTTD algorithm are given.

Author Mehar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, Ammar Rayes implements algorithm for Exploiting profit ratio by process elasticity on heterogeneous environment.

Author M. NoroozOliaee, B. Hamdaoui, M. Guizani implements algorithm for Online multiresource scheduling for minimum task completion time in cloud servers.

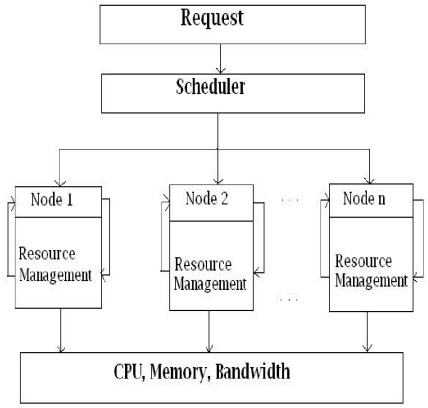


Figure 1: Task Scheduler Modular Approach

III. ARCHITECTURAL MODEL

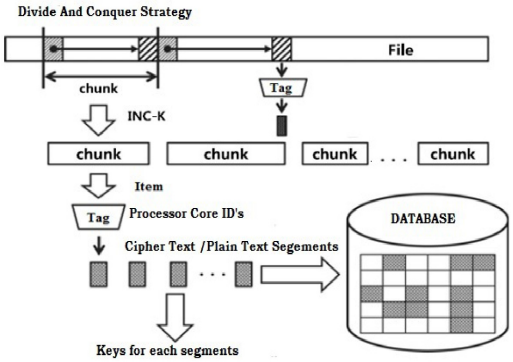


Figure 2: Architecture diagram

LIMITATIONS OF PROJECT

- Multi-core System Required For Execution.
- Achieve Cryptosystem Profit Gains Only.
- Internal Based Parallelism Execution Only.

IV. BYTE ROTATION ALGORITH

1. **Features :** Any key size.
2. Support all file type

Steps for encryption :

1. Design matrix such that column is equal to key length and row is calacuted by using below formaula

   if (len(file) % len(key) == 0) then

   row = len(file) / len(key)

   else

   row = (len(file) / len(key)) + 1

   where,

   len(file) = length of file

   len(key) = length of key

2. Character of file are inserted in rows of matrix from left to right.

3. Encrypt by rearranging column of matrix according to sorted order of key characters.

4. Characters from matrix in column wise from top to bottom are written in file.

Steps for decryption :

1. Design Matrix such that row is equal to key length and column is calacuted by using below formaula

   if (len(file) % len(key) == 0) then

   column = len(file) / len(key)

   else

   column = (len(file) / len(key)) + 1

   where,

   len(file) = length of file

   len(key) = length of key

2. Character of file are inserted in column of matrix from top to bottom.
3. Assign index to key characters in increasing order of ascii value.
4. Decrypt by rearranging row of matrix according to index assign to key characters.
5. Characters from matrix in column wise from top to bottom are written in file.

V. IMPLEMENTATION

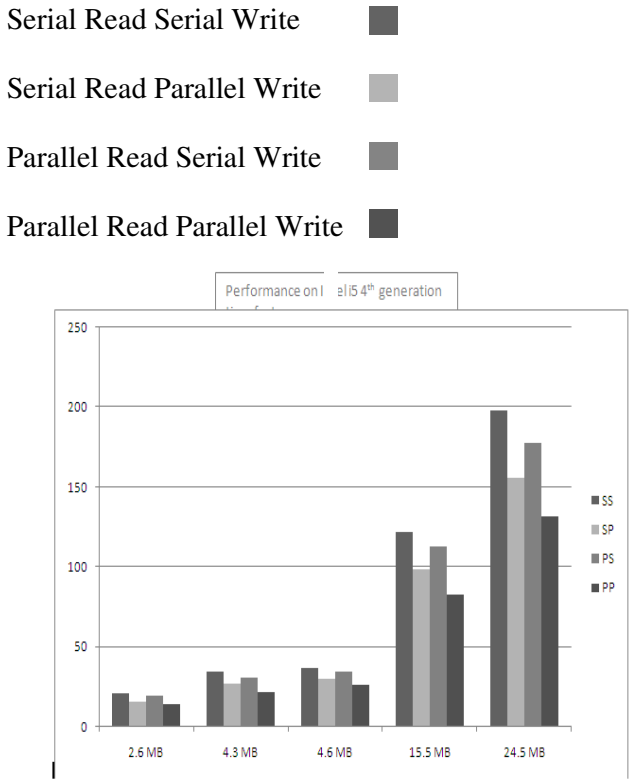
Technology Used :
Implementation is done using java programming language but can be done in any of programming language. Editor tools like eclipse is used for development.

DataSet :
As this algorithm deal at ASCII level there is no restriction on file type. This ascii value are stored in two dimensional array of byte data type to perform operations on it.

Result Analysis:

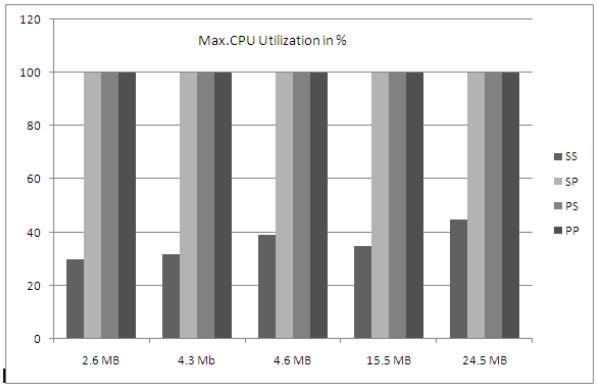
Result of Implementation is calculated on System with Intel i5 4th generation (a quad core CPU with 3.4 GHz frequency) and 8 Gigabytes of Main Memory. Eclipse tools is used on Window 10 Operating System for implementation.

Combinations of parallel and serial mode of operation on read and write operation are used and represented in graph bellow are mode of operations used



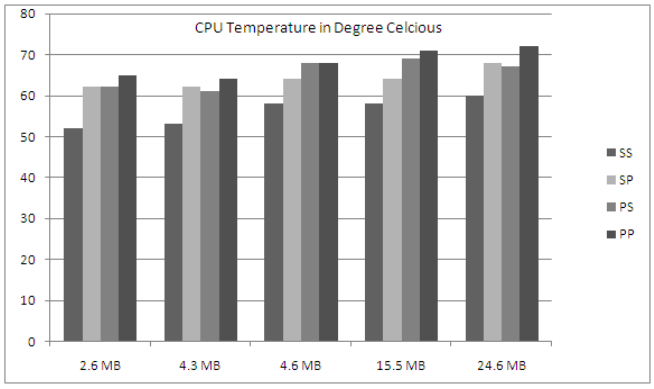
Graph 1: Time Factor in Milliseconds

Above graph shows time factor for all four modules. By observing graph we can see that for any size of file serial read serial write module takes maximum time for completion of process.and parallel read parallel write module takes minimum time.



Graph 2: CPU Utilization

In above graph we can observe that serial read serial write module utilizes only max 30-35% of cpu.And other three modules utilizes max.100% of cpu during execution.



Graph 3:Cpu Temp.

In above graph,temperature factor on various modules is given.there is a slight change in cpu temperature as we put extra load on cpu by doing tasks in parallel.

VI. CONCLUSION

In this work,four different modules are generated and tested on various processors.

By observing the results,we found that some modules are best on dual core,some are better on i3 and some on i5.processing power of dual core cpu's is less than that of Quad core processors.so first module i.e. serial read serial write module works better on dual core cpu's.

I5 4th generation and 6th generation having good computational power than that of previous ones. So second,third and 4th module works better on these processors.we observe that serial read parallel write module takes less time for both encryption and decryption than that of parallel read serial write module.parallel algorithms put large load on centrino processors so we should not use modules other than serial read serial write on

these processors.if we try to use such modules on centrino,cpu temperature changes to suddenly high range and it can have effects on cpu performance.

Parallel read parallel write is the best module which gives best results on i3,i5,i7 processors.so instead of using first three modules,fourth module is used for the execution of encryption and decryption for the best results.At last we conclude that for the slow processors serial read serial write approach is to be used and for high speed processors parallel real parallel write approach is best which gives best results compared with existing systems.

VII. REFERNCES

[1] Mehar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, Ammar Rayes, Exploiting profit ratio by process elasticity on heterogeneous environment & quot; VOL. 27, NO.6, JUNE 2015

[2] V. Maitri, Dattatray S. Waghole, Vivek S. Deshpande, IEEE Senior Member, Low latency for encryption and decryptionusing BRA algorithm in network security & quot;, 2015 International Conference on Pervasive computing.

[3] BingChun Chang, Running Time Improvement for Two Thresholds Two Divisors Algorith & quot;, December 2009

[4] M. NoroozOliaee, B. Hamdaoui, M. Guizani, Online multiresource scheduling for minimum task completion time in cloud servers & quot; in Computer Communications Workshops, 2014 IEEE Conference