

# Distributed Cluster Processing to Evaluate Interlaced Run-Length Compression Schemes

Ankit Arora  
M.Tech (IT)

Guru Nanak Dev University Asr.  
Asst. Prof. at LLRIET, Moga

Sachin Bagga  
M.Tech (CSE)

Thapar University Patiala  
Asst. Prof. at LLRIET, Moga

Rajbir Singh Cheema  
M.Tech (CSE)

Guru Nanak Dev Eng. College Ldh.  
Associate Prof. at LLRIET, Moga

## ABSTRACT

Parallel computation, a greater advancement in computational hardware as well as new achievement in current scientific computing such as image processing involves huge exhaustive computation and data processing leading towards parallel architectures. Parallel hardware organization basically a suitable interconnection among computational hardware, where current trends now involves clustered organization of distributed hardware to achieve parallel effects. Cluster environment consisting multi-computer network nodes provides flexible architecture towards high complex data parallelism as well as control parallelism operations. Further detail consists interlaced graphics mechanism with run-length encoding to achieve high compression benefits. Run-length compression speedup benefits have already described in the research IJCA-2011 *cluster based performance evaluation of run-length image compression*, which is now updated to cover interlaced lossy compression schemes. In general interlacing provides a lossy compression formulation but acceptable in real-life scenarios. Finally, the interlaced methodology and cluster based analysis results will be discussed.

## General Terms

Massive Parallelism, Multi-Computer Cluster, Interlaced Compression, Client Server TCP/IP Sockets.

## Keywords

Parallelism, Distributed Clustering, Multi-Computers, Run-length Image Compression, Interlacing, twips.

## 1. INTRODUCTION

Massive parallel processing typically suited to high scientific computations generally not well responded by the multiprocessor environments having some limited no. of processor cores, where each core behaves transparently under the control of operating system, without any interference from the programmer side. Other advantage of massive parallel system is that these systems provides not only processor redundancy but also the resource duplicity such as each individual machine has its own processor, memory interface having both primary as well as secondary memory units controlled by its own operating system. Data parallel operations covers workload partitioning and distribution over logically

programmed cluster nodes where the control parallel operations distributes parallel multiple control threads over cluster nodes, each of these control threads performs different task of execution. Although the combination of control as well as data parallel operations can be achieved to obtain multi-programmed multiple data model. Clusters can be further organized/interconnected on the basis of their speed and computational programmability model assigned, in other words the computational structure for which the machine is designed according to that the parallel tasks are assigned i.e. the scheduling over interconnected clusters. Cluster interconnection Scheduling categorized as CSS (cluster specific scheduling) and ISS (interconnection specific scheduling), where interconnection scheduling (external to the clusters) specifies how one server node assigned/shares its workload to other server node and cluster specific scheduling (internal to the cluster) specifies how one server node distributes its workload to its associated connected clients. In addition to cluster interconnection, the workload characterization is another important aspect via scheduling parallel jobs. High computational intensive workload may be distributed to faster processor cluster [5]. Other related parallel aspects, the jobs may be moldable to adapt available parallel architectures of any kind regardless of one specific hardware paradigm [3]. Earlier research carried out covers matrix multiplication over parallel cluster hardware, Multiprocessor Scheduling simulations via Space sharing policies, clustered approach to run-length image compression or many more related work with fractal image theory.

## 2. LITERATURE REVIEW

Previous Literatures reviews around parallel execution stipulate simulation behind space sharing policies environments published in research *simulated performance analysis of multiprocessor dynamic space sharing policies (IJCSNS-2009)*. This Simulation environment covers space-sharing policies, their classifications and scheduling via poison distribution is performed, space sharing structure experiment where multiple processors are assigned to current active job. Other research towards parallel clustering involves large matrix multiplication analysis published in research *cluster based parallel computing framework for evaluating parallel applications (IJCTE-2010)*. Many other research covering cluster-based operations involves

pipelined based *parallel implementation of dijkstra algorithm (FSU.CS research data base)*. Image compression over the clustered architecture gives a new dimension to scientific computing published as *cluster based performance evaluation of run-length image compression (IJCA-2011)* [1], where the images is partitioned among cluster nodes and each of the intended cluster node performs run-length compression over a partitioned image chunks. Other Literature around parallel image compression consists *parallel implementation of fractal image compression in web service environment (IEEE-2011)* [2], wavelets based parallel image compression and analysis (WASET-2005) [4]. The idea behind this research is similar to these previous literatures but follows interlacing with run-length encoding scheme, describes new updated version of earlier research implemented run-length encoding (IJCA-2011) over parallel cluster using divide and conquer paradigm. The previous research is now updated to adapt lossy-based interlaced mechanisms to achieve more compression benefits for high resolution (Twips Unit) image. The image used for compression is same as used in earlier research published. In general the Interlaced run length-encoding scheme is a lossy compression technique providing image lose which is acceptable up to some extents.

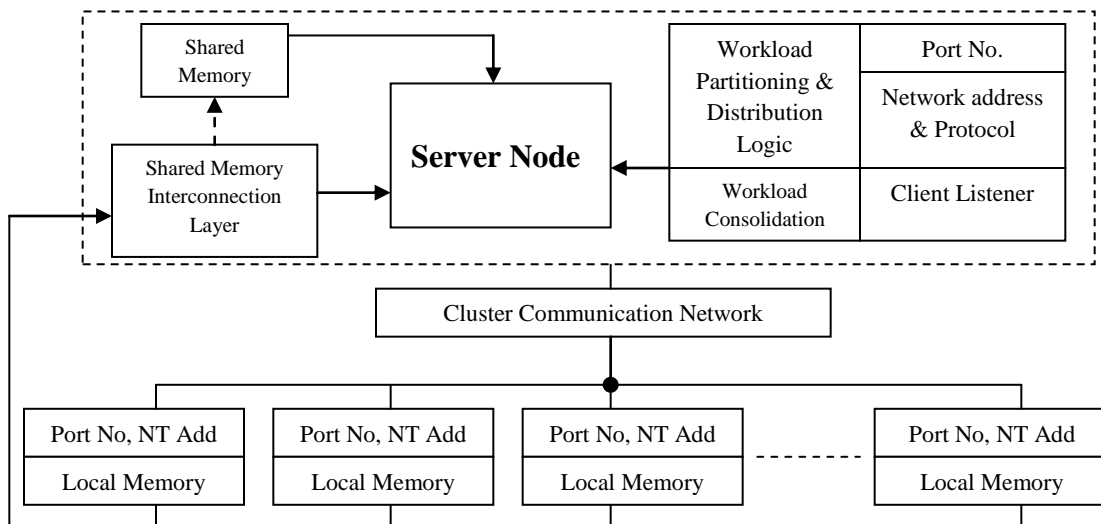
## 2. INTERCONNECTION ANATOMY

Clustered Interconnection composed of client-server model of computation where one machine acts as a server performing job partitioning and final consolidation of individual outcomes, other machines acts as a clients communicated via TCP/IP sockets performs their intended work assigned by the server. Each machine behaves independently of others or having

autonomous structure providing flexibility to encourage parallel theory and applications as described in the figure-1. The experiment covers nine cluster nodes (Pentium4 3.4 GHZ processor with 1GB of RAM and WinXP SP2 OS) organized on the basis of SIMD based computational model for data parallel operations with the underlying idea of workload partitioning and distribution via shared memory, this will implements the asymmetric tightly coupled distributed system [6]. Each cluster node picks its intended sub task from the shared memory (server side) whenever the control message instructing initiation of execution of sub task is received from the server. Control message is sent by server to ensure the completion of workload partitioning and for ready status of subtasks. Finally the cluster node computes their individual outcomes and sent the results back to server’s shared memory via shared memory interface.

## 3. LOGICAL PROGRAM STRUCTURE

Logical programming structure consisting client-server distributed software implemented through VB.6.0 TCP/IP socket programming using Mswinsock.ocx. The control provides a listener interface configured via unique port no. and network address associated with cluster node [9]. Each cluster client sends a connection establishment request to server via unique port no., rest of the network communication is then performed via this connection. Image workload is retrieved and then computes interlaced run-length compression scheme, finally, the results sent back to server’s shared memory, where the final consolidation of individual cluster results will be performed.



**Fig 1: Cluster Interconnection Autonomy**

## 4. INTERLACING

Interlacing is generally a technique used by raster scan video controller in computer graphics to avoid flicking or to provide

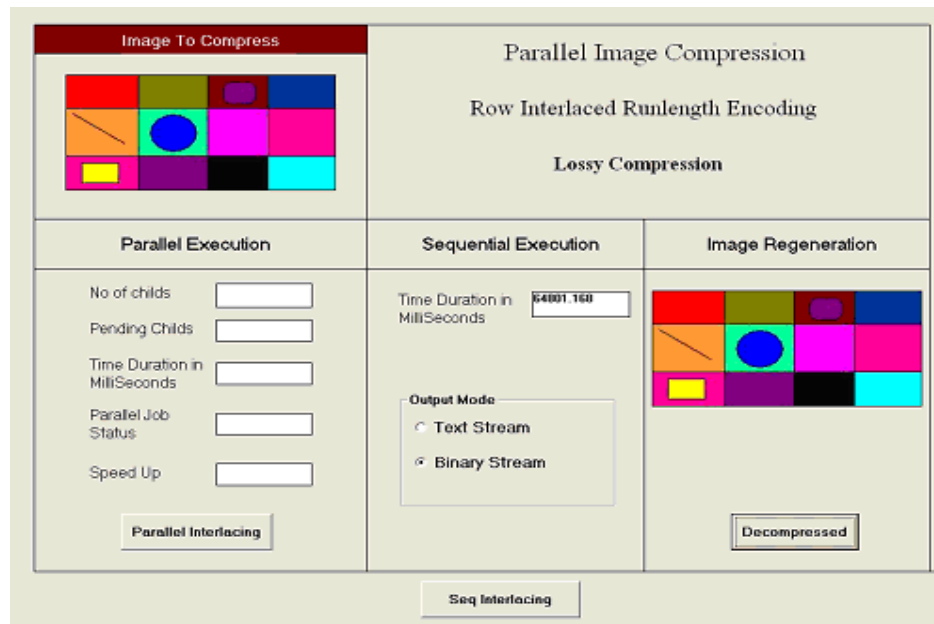
user a view that entire image is displayed in one go, the controller firstly display all of the odd image scan lines and then all of the even image scan lines, also the refresh rate is of two

level process firstly for odd lines and then for even lines, half time faster refresh rate than non-interlaced system without flicking [8]. This user view of seeing entire picture in one go can be incorporated in compression schemes. As the distance between image scan lines are very small, so eliminating one adjacent scan line will not be noticeable or in other words this type of fidelity is almost ignored by human eye or visually imperceptible. This technique can be further utilized as lossy based compressions, although some of the picture information will be lost but insignificant. In further research, the idea comprised with run-length encoding scheme over parallel cluster will be discussed. The analysis results covered row based interlacing where one row has been eliminated from each pair of image scan lines. Other version contains both row as well as column based lossy compression where one row as well as one column is eliminated from each pair of adjacent horizontal and vertical scan lines. The technique can be utilized for medical images extracted from nuclear scanners or tomography systems and as well as for animations, where a frame emerged over the display for small extent of time. Quality degradation cannot be perceived over high-resolution systems

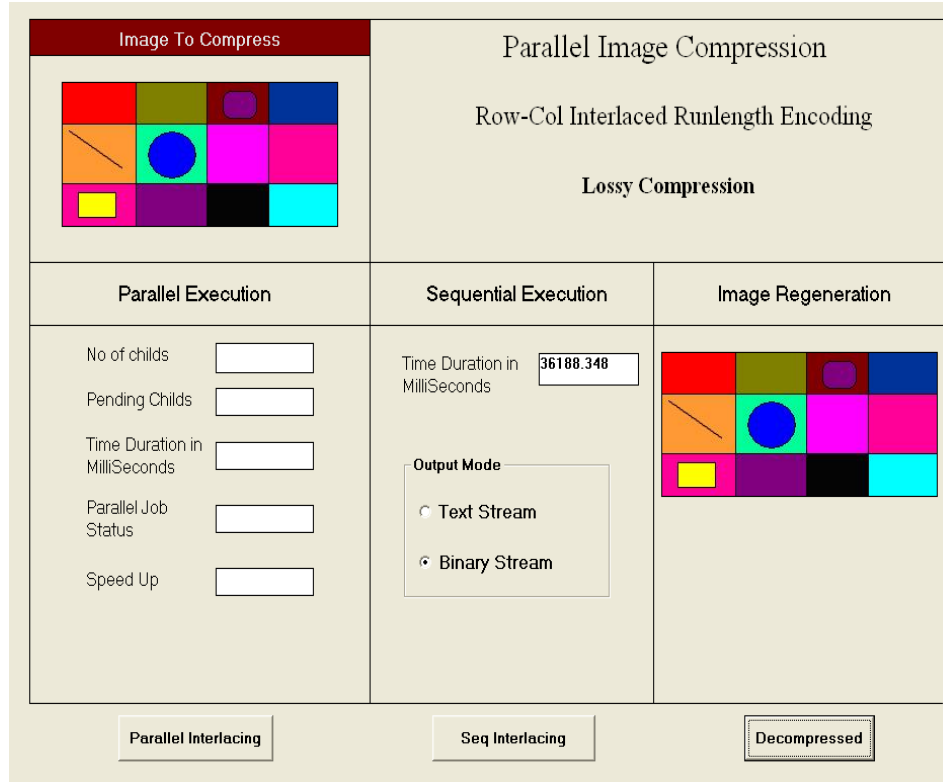
## 5. PERFORMANCE ANALYSIS

Interface below consisting row as well as column interlaces mechanisms, the compression results stored either by means of text or binary mode. As below row as well as row-column interlacing provides lossy compression, which is visually imperceptible and not noticeable over a high-resolution system. Pixel based operations can also be performed rather than twips based units, later the image by applying interlaced run length

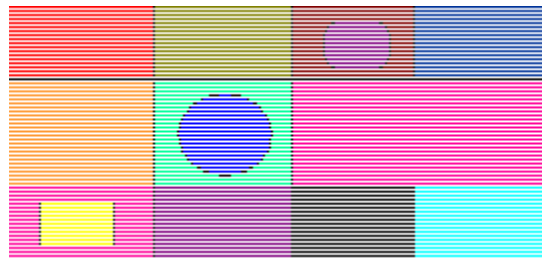
over pixel based image will also be produced. This will not provide any usual benefits during display, although the size of the file is reduced up to very large extent but the quality loss some times not acceptable. In this cluster operation the results computed by taking twips based image as a basic source because 1 pixel is equivalent to 15 twips so quality loss is acceptable and imperceptible up to very large extent. Despite of this, file size for both twips based row interlace and row-column interlace is same, because when the run-length encoding is performed with row-column interlace, even the columns are eliminated, once the memory is allocated to one twip then how many no. of twip of same color will be stored with in the that memory is vary. Consider a 4 byte memory for storing 32 bit true color code and a 2 byte memory for storage of no. of twips of same color value. Now suppose there are 1200 twips of color red in one scan line if using row interlace so 2 byte memory is sufficient for this, but again if column interlaced is also embedded along with it then same memory will be used for storing this time only 600 twips. So memory capacity is same, only the underlying value will be changed (no. of twips). So this provides the benefits only when the picture is displayed, the speed of row-column interlace will be faster during display as compare to row interlace. Although, file size for pixel based interlacing is vary because eliminating one column pixel means 15 twips elimination at once. So pixel identity is completely lost but in twips unit format nearly half of the twips under one pixel are eliminated as in even/odd fashion (interlacing). So pixel identity is still available partially, that's why the memory is still required for that pixel in twips format during row-col interlacing.



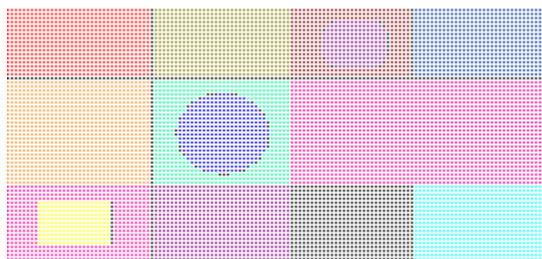
**Fig 2: Row Interlaced Run-length Compression over twips based image**



**Fig 3: Row-Col Interlaced Run-length Compression over twips based image**



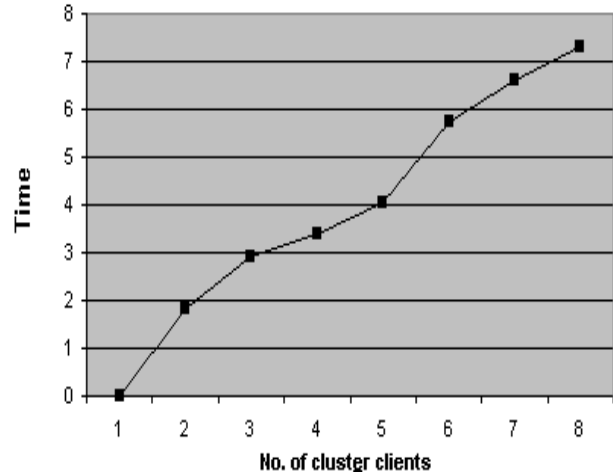
**Fig 4: Row Interlaced Run-length Compression over pixel based image**



**Fig 5: Row-Col Interlaced Run-length Compression over pixel based image**

### 5. PERFORMANCE MEASUREMENTS

The experiment implemented via visual basic 6.0 language tool with image scan lines as the basic parameter for distribution. The total numbers of scan lines are then divided among available or designated cluster size (no. of client machines) for execution. Each client then performs its intended interlaced mechanism and finally send result back to the server’s shared memory. Metrics used for performance measurements are speedup, efficiency as well as parallel overhead [1]. Following are the computed results and timing variation (Sec.) graphs-



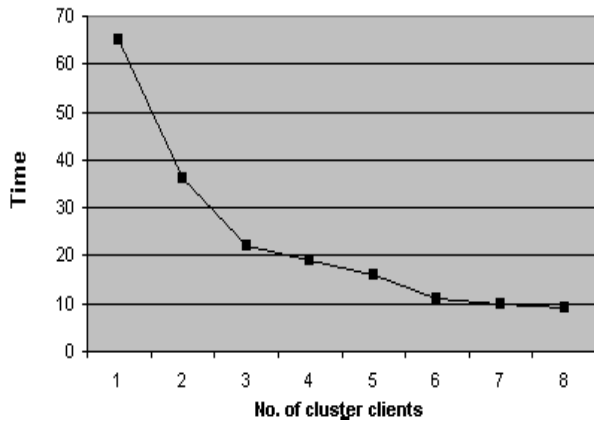
**Fig 7: Row-Interlace Speedup Variations**

**Table 1: Row-Interlace Timing Variations**

Cluster Clients	Time (Ms)	Time (Sec)
1	64801.168	65
2	35619.922	36
3	22161.497	22
4	19157.003	19
5	16017.916	16
6	11314.188	11
7	9816.213	10
8	8875.153	9

**Table 2: Row-Interlace Speedup Variations**

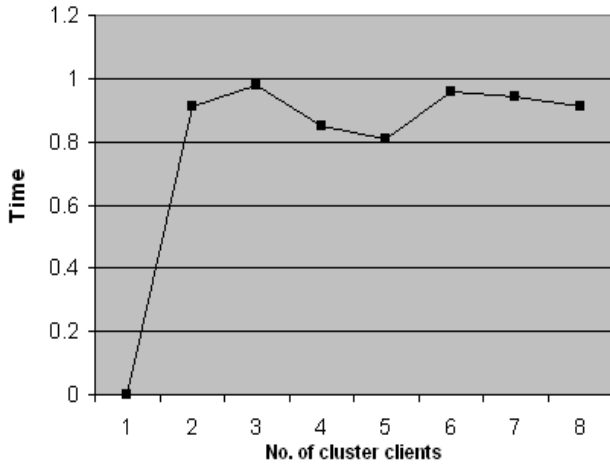
No. of Clients	Speed Up
1	0
2	1.82
3	2.92
4	3.38
5	4.05
6	5.73
7	6.60
8	7.30



**Fig 6: Row-Interlace Timing Variations**

**Table 3 Row Interlace Efficiency Per Cluster Machine**

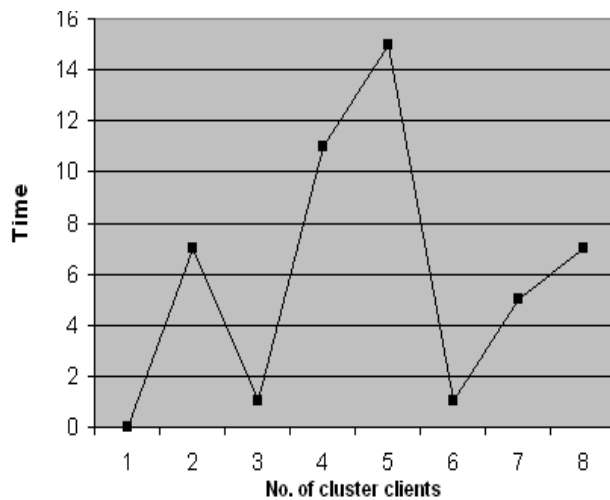
No. of Cluster Clients	Time (Sec)
1	0
2	0.91
3	0.98
4	0.85
5	0.81
6	0.96
7	0.94
8	0.91



**Fig 8: Row Interlace Efficiency per cluster Machine**

**Table 4: Row Interlace Parallel Overhead (Sec)**

No. of Cluster Clients	$P * \beta$	$P * \beta - \alpha$
1	65	0
2	72	7
3	66	1
4	76	11
5	80	15
6	66	1
7	70	5
8	72	7

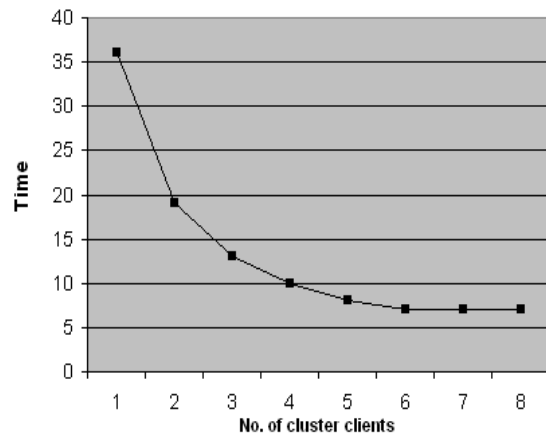


**Fig 9: Row Interlace Parallel Overhead**

Consider other performance measurements generally described as parallel overhead. Parallel overhead is the overhead, which specifies the time spent in parallel computation managing the computation rather than computing results. Here  $\beta$  specifies the time consumed by parallel cluster having  $p$  machines and  $\alpha$  refers to the time consumed by single machine for the same task [1]. The row-interlaced overhead is calculated as described above in the Table-4.

**Table 5: Row-Col. Interlace Timing Variations**

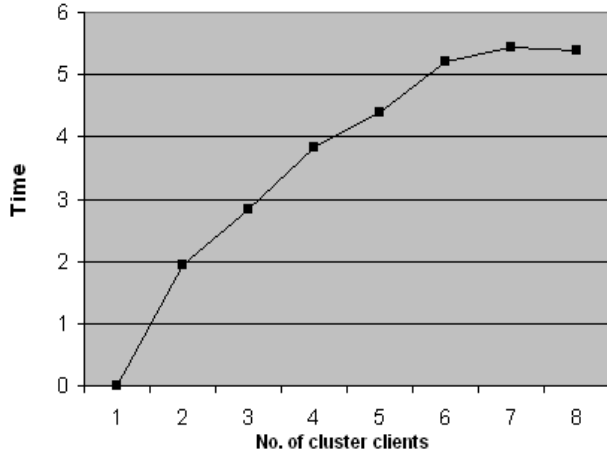
No. of Cluster Clients	Time (Ms)	Time (Sec)
1	36188.348	36
2	18488.367	19
3	12744.959	13
4	9454.881	10
5	8210.707	8
6	6941.466	7
7	6647.326	7
8	6714.897	7



**Fig 10: Row col Interlace Timing Variations**

**Table 6: Row-Col. Interlace Speedup Variations**

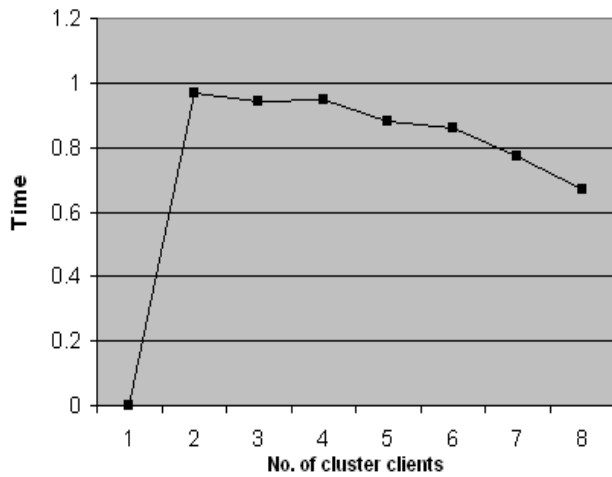
No. of Clients	Speed Up
1	0
2	1.95
3	2.83
4	3.82
5	4.40
6	5.21
7	5.44
8	5.38



**Fig 11: Row col Interlace Speedup**

**Table 7: Row-Col. Interlace Efficiency per cluster machine**

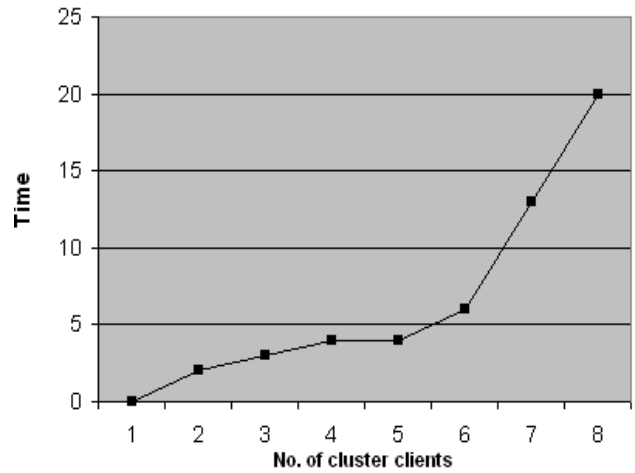
No. of Cluster Clients	Time (Sec)
1	0
2	0.97
3	0.94
4	0.95
5	0.88
6	0.86
7	0.77
8	0.67



**Fig 12: Row col Interlace Efficiency**

**Table 8: Row-Col. Interlace Parallel overhead**

No. of Cluster Clients	$P * \beta$	$P * \beta - \alpha$
1	36	0
2	38	2
3	39	3
4	40	4
5	40	4
6	42	6
7	49	13
8	56	20



**Fig 13: Row col Interlace Overhead**

**Table 9: Compression Results**

Compression Type	Mode	Unit	File Size
JPEG Image	JPG	Pixel	102 KB
Run length	Binary	Twips	96KB
Row Interlace with Run-length	Binary	Twips	48.5KB
Row Col Interlace with Run-length	Binary	Twips	48.5KB
Row Interlace with Run-length	Binary	Pixel	3.08 KB
Row Col Interlace with Run-length	Binary	Pixel	2.61 KB

## 6. CONCLUSION & FUTURE WORK

Experiment estimated using multi-computer cluster with lossy-based compression schemes produce very effective results as described in the Table-9. As described above the compression results are very beneficial for online data transmission over the network, where video conferencing and animations consumes less bandwidth over distant data transmissions, also lossy effects perceptible only over low resolution system as above covered pixel based operations shows quality degradations, whereas twips based image shows high resolution and quality loss is imperceptible. Although pixel based interlaced compression can not be discarded in real-life because after decompression still the image shows their interior effects or their inner components strength and shades. Future versions will cover more improved parallel architectures to enhance the capability of such compression schemes. Because from this research it has been concluded that maximum time will be consumed during large workload transmission from machine to machine. So this can be improved via mesh or multiple interconnection transmission lines, still the results are very efficient.

## 7. REFERENCES

- [1] Ankit Arora, Amit chhabra Nov 2011, Cluster Based Performance evaluation of Run length Image Compression, Vol.33, International Journal of Computer Application, Foundation of Computer Science, New York.
- [2] Yan Fang Oct 2011, parallel implementation of fractal image compression in web service environment (IEEE-2011).
- [3] Gerald Sabin, Matthew Lang 2006, Moldable parallel job scheduling using job efficiency: an iterative approach 12<sup>th</sup> International Conference, Springer Verlag Berlin Heidelberg ISBN: 978-3-540-71034-9.
- [4] M. Kutila, J. Viitanen, Parallel Image Compression and Analysis of Wavelets, Word Academy of Science Engineering and Technology 2005.
- [5] TD Nguyen, 1996 Parallel Application Characterization for Multiprocessor Scheduling, Department of Computer Science and Engineering, Box 352350 University of Washington, Seattle, WA 98195-2350 USA.
- [6] Kai Hwang and Faye A. Briggs, Computer Architecure and parallel processing, Tata McGraw Hill Publishing Ltd. 1985, Computer Science Series, ISBN: 007-066354-8.
- [7] Joseph JaJa, Introduction to Parallel Algorithms, University of Maryland 03/24/1992, ISBN-13: 9780201548563, Addison-Wesley Professional
- [8] John Amanatides, Antialiasing of Interlaced Video Animation 1990, ACM-0-89791-344-2/90/008/0077.
- [9] Carl Franklin, Visual Basic 6.0 Internet Programming 1999, ISBN-10: 0471314986, Wiley Publishing Ltd.