ON THE FLY PORN VIDEO BLOCKING USING DISTRIBUTED MULTI-GPU AND DATA MINING APPROACH

Urvesh Devani, Valmik B. Nikam and B.B. Meshram

Department of Computer Engineering and Information Technology
Veermata Jijabai Technological Institute, Mumbai, India

ABSTRACT

Preventing users from accessing adult videos and at the same time allowing them to access good educational videos and other materials through campus wide network is a big challenge for organizations. Major existing web filtering systems are textual content or link analysis based. As a result, potential users cannot access qualitative and informative video content which is available online. Adult content detection in video based on motion features or skin detection requires significant computing power and time. Judgment to identify pornography videos is taken based on processing of every chunk from video, consisting specific number of frames, sequentially one after another. This solution is not feasible in real time when user has started watching the video and decision about blocking needs to be taken within few seconds.

In this paper, we propose a model where user is allowed to start watching any video; at the backend porn detection process using extracted video and image features shall run on distributed nodes with multiple GPUs (Graphics Processing Units). The video is processed on parallel and distributed platform in shortest time and decision about filtering the video is taken in real time. Track record of blocked content and websites is cached, too. For every new video downloads, cache is verified to prevent repetitive content analysis. On the fly blocking is feasible due to latest GPU architecture, CUDA (Compute Unified Device Architecture) and CUDA aware MPI (Message Passing Interface). It is possible to achieve coarse grained as well as fine grained parallelism. Video Chunks are processed parallel on distributed nodes. Porn detection algorithm on frames of chunks of videos can also achieve parallelism using GPUs on single node. It ultimately results into blocking porn video on the fly and allowing educational and informative videos.

KEYWORDS

Porn Video Filtering, Multi GPU, CUDA, MPI, Content Based Analysis, Image Processing, Data Mining.

1. INTRODUCTION

In most of the organizations, precautions for the on campus internet access are necessary, specific filters prevent users from accessing any porn content through campus network. On the other side, traditional filters or firewall do block necessary educational material, i.e. informative videos and online courses, due to high false positive ratio. Unfortunately, users are deprived from what they are supposed to refer. This happens due to filtering techniques applied. Textual content based or link based analysis many times lead to mis blocking. Content based analysis is the need of an hour. But when it comes to the video that user is playing, practically it is difficult to take decision about video content in real time.

The focus of our work is to do adult content detection and prepare a model for implementing that on distributed multi GPU System. Looking into content based analysis of video, it is necessary to get features and statistics about various individual frames. It involves bulk image processing.

DOI:10.5121/ijdps.2014.5401

There are plenty of ways for an adult image classification [1][2][3][4][5]. Due to similar kind of problems and large data, that we want to address in real time events, we select this approach of using distributed multi GPUs for our identified problem of on the fly porn video blocking, utilizing more benefits from latest available GPU architecture.

The paper is organized as follow; Section-2 includes Literature Survey. Section-3 explains our proposed System. Section-4 has performance analysis for our solution and section-5 involves conclusions.

2. LITERATURE SURVEY

Shirali – Shahreza, S. and Mousavi, M.E. (2008) suggested non parametric data mining approach using Bayesian classifier which predicts the pixel being a skin pixel using specific training dataset. This method has high true positive and low false positive detection ratio than the other parametric counterparts [1]. Another way is to detect a close up face with the traditional porn detection using support vector machine and skin colour model together, and by that way reducing cases of large face portion being detected as porn [2] Transformation of RGB colour model to other models like YCbCr or HSV followed by image segmentation through skin detection is also a solution[3].

Apart from statistical model, we can consider video time continuous proposed by L. Yin et al.[4]; an important parameter which contributes to the correct results by considering preceding and forwarding N frames as well. Gaussian windows are used to reduce the overall misclassification of the adult images. Adult video detection can be more accurate in dynamic states [4]. A. Ulges et al. (2012) proposed multi-modal approach for the image classification using visual words, Motion histograms and other features. They presented detailed study on large dataset of 500 hours of video [5].

All of the models discussed above may lead to poor performance when analyzing multiple frames in the video. Many data mining approaches including those mentioned above are parallel friendly. Real time analysis of videos using one of these approaches is not possible without using multiple GPUs and parallel system. We concentrate on using Bayesian classifier based skin detection. Bayesian approach is quite popular in data mining from whether prediction to image processing areas [16]. There are different approaches available for parallelizing various data mining techniques for applications [18].When using this data mining approach, it is required to have already calculated adult and non adult histograms from large training set of images. Complete process involves calculating these histograms, counting conditional probability, and threshold value using Bayesian equations [4]. Prior Probabilities are not required for decision taking [7]. 3D histogram of 32 bin size performs better than the 256 bin size [8].

The skin detection will run in parallel on our proposed system. Researchers have made sample image data set of 80000 images [1] which can be used instead of famous COMPAQ database for training Bayesian model. Processing any large set of images require significant computing time and power. Going with multi model approach, like using combination of 4 most widely used features [5], will require large data processing. For a Video, it won't be feasible in real time cases when we want to detect porn content on the fly before user reaches to the content.

The only option is to go with high performance system where result can be decided within seconds. Available functionalities of CUDA make it the best choice for GPU in this type of system. It will require load distribution among GPUs. This task can efficiently be performed using any CUDA Aware MPI. The Open MPI project supports GPUs by version 1.7. Due to Unified Virtual Addressing (UVA) feature in CUDA, the host memory and the memory of all

GPUs in a system (a single node) are combined into one large (virtual) address space. NVIDIA GPUDirect technologies provide high-bandwidth, low-latency communications within NVIDIA GPUs on intra nodes and inter nodes [6].

Latest CUDA GPUs support execution of thousands of threads parallel (Tesla K20 GPU). Each CUDA block contains several numbers of threads. Maximum number of threads any CUDA kernel can contain is 1024 for latest GPU devices. Figure 1 [15] demonstrates the simple CUDA Model .These blocks can also be launched in parallel up to the run time limit defined by GPU specification of SMs. Due to shared memory limitations of 48 KB on CUDA, it won't be possible to store 3D histograms in shared memory which is accessible to individual block. Global memory is the only choice. Because we are counting occurrences of colour, if we have multiple threads and they try to access the same location on memory, there might be memory conflicts and making single histogram in global memory common for all images in single GPU will lead to more misaligned access [14]. Now, having multiple GPUs on single node and those types of multiple nodes connected via network makes the complete system high performance enabled.

Many authors have explored the way of generating Multi node or distributed environment along with multiple GPUs and have used it for various purposes [9][10][11][12]. Yi, Zhiwei and Huawei (2013) successfully implemented Distributed Multi-Node GPU Accelerated Parallel Rendering Scheme for Visualization Cluster Environment [10]. An approach for system scalability for video on demand [17] can be utilized in complete system implementation.

GPUs have been used effectively at large scale in Distributed computing environment [11]. Wenger, Ament, Guthe and Lorenz (2012) experienced huge performance gain by exploiting combined graphics memory in distributed cluster [12]. Wasif and Narayanan (2011) achieved 2 times speedup while using multi GPUs for K-means than on single GPU[13].

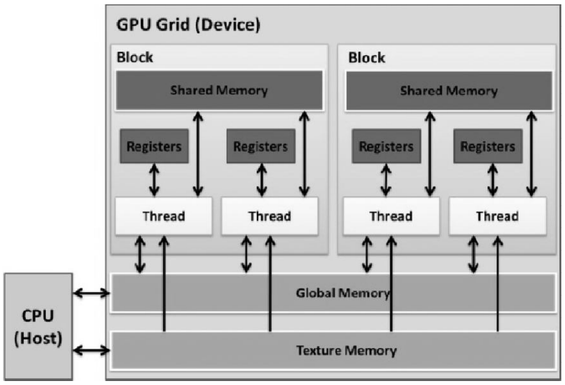


Figure 1.CUDA Memory Model

3. PROPOSED SYSTEM

As shown in Figure 2, overall system works according to the flow given below.

1. Large training data set of images is fed to the distributed system and necessary model is generated.
2. When new video comes , first it is verified with the previously analyzed videos
3. Video is fed in terms of chunks of image frames to multi GPUs of multiple nodes by master node
4. Multiple GPUs process the images and gives decision to master node.

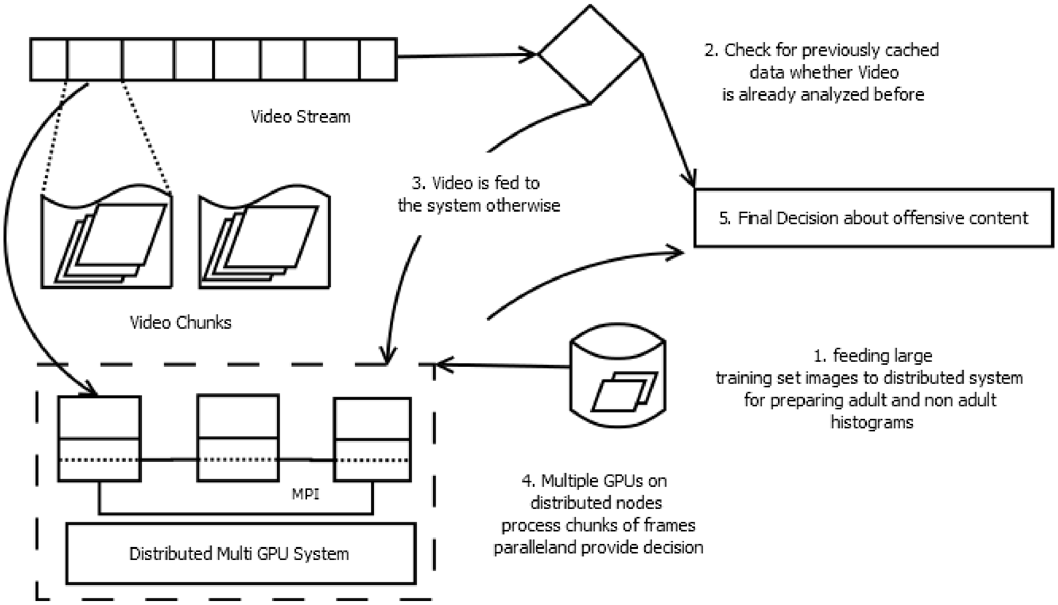

2. Check for previously cached
data whether Video
is already analyzed before
Video Stream
3. Video is fed to
the system otherwise
5. Final Decision about offensive content
Video Chunks
1. feeding large
training set images to distributed system
for preparing adult and non adult
histograms
MPI
4. Multiple GPUs on
distributed nodes
process chunks of frames
paralleland provide decision
Distributed Multi GPU System

Figure 2. Overall System Architecture

Step-I: **Training:** It is required to train a specific dataset with bulk skin and non skin images for creating skin and non- skin histograms. Each of the 3 dimensions should be divided into 32 bins. As mentioned earlier 32 bin size is better option than 256. After training, histogram counts should be normalized creating discrete probability distribution as given in below equation.

$$P_{adult}(c) = \frac{adult[c]}{norm} \qquad (1)$$

Normalization value here is the total count contained in the adult or non adult histogram and *adult[c]* is the bin count for colour c. same equation is used to determine non adult histograms. This training phase is run on multi GPUs of distributed system. Each GPU is provided with specific set of images. Images on single GPU are divided among blocks.

Our solution is to implement 3 kernels for training on CUDA which work in following way.

hist kernel: every single image is processed by single block. Each block consisting about thousand threads will update and maintain individual 3D histogram for image in global memory itself. Image matrix is accessed from global memory because same pixel needs to be processed not more than once. So, copying part of the matrix to shared memory and then processing won't increase CGMA (Compute to Global Memory Access). Apart from that, 3D histogram in which threads of single block are working can be partially loaded in shared memory but it won't contribute to the performance and might add overhead because image might have pixels in different range.

- Each thread reads the multiple pixels according to image size and updates corresponding counter value in its own histogram using atomic operations.
- It is made sure, using *__syncthreads()*, that each thread has updated value in histogram array.

After execution of *hist* kernel we have Histograms of all the images assigned to GPU in global memory of GPU.

reduce sum kernel: This kernel is executed multiple times. Each block works on single element of resultant histogram. Total sum for single element is calculated by threads of block in coalesced manner.

- Each thread of same block will copy related element, for which resultant value needs to be computed, from every histogram to shared memory.
- Now, each block has all the values that need to be added up for final value. Thus, Sum is done where threads of same block calculates intermediate sum and goes further in same manner till only 1 value is present.

When number of images is more, *reduce sum* kernel needs to be executed multiple times.
When Each GPU has its own histogram, all the histogram arrays are combined on single GPU using MPI and *reduce sum* kernel is executed again.

normalization kernel: Each block of threads works on TILE of specific size of the resultant histogram. TILE from the histogram is copied to shared memory. Threads divide assigned element by *norm* value. Data is copied back to the global memory when all threads have done the job.

Step-II, III: When user starts to play any video, Backend side detection is triggered. Master node first checks previously stored data of analyzed videos whether video content is already analyzed or not using attribute information and metadata. If it is the case then decision is taken and event takes place accordingly.

The master node takes care about distributing load. There is no need of specific scheduling for simple case when sufficient number of GPUs is available as once the load is distributed; all slave nodes work on assigned data without any communication with master except when porn content is detected.

Step-IV: Major task at this stage is to download as much content of the video as possible at very high bandwidth at back end side and feed it to the master node. Master node can break video into image sequences. For breaking the video into image sequences, libraries like OpenCV (Open Source Computing Vision) is used effectively at master node which has GPU enabled module for video readers. Master node then can distribute images to every node's host system as shown in figure 3. Host at node will launch appropriate kernels on GPU device. Data can be transferred from GPU memory to another GPU memory directly due to GPUDirect technology in latest devices. After that, within few seconds, all GPUs analyze allocated frames in parallel and provide judgement.
For classifying individual image, probability of any pixel being adult can be calculated by using Bayes Theorem.

$$P(adult/c) = \frac{(P(adult) * P(c\ /adult))}{P(c)} \qquad (2)$$

Here final consideration of adult content can be done by following equation [7].

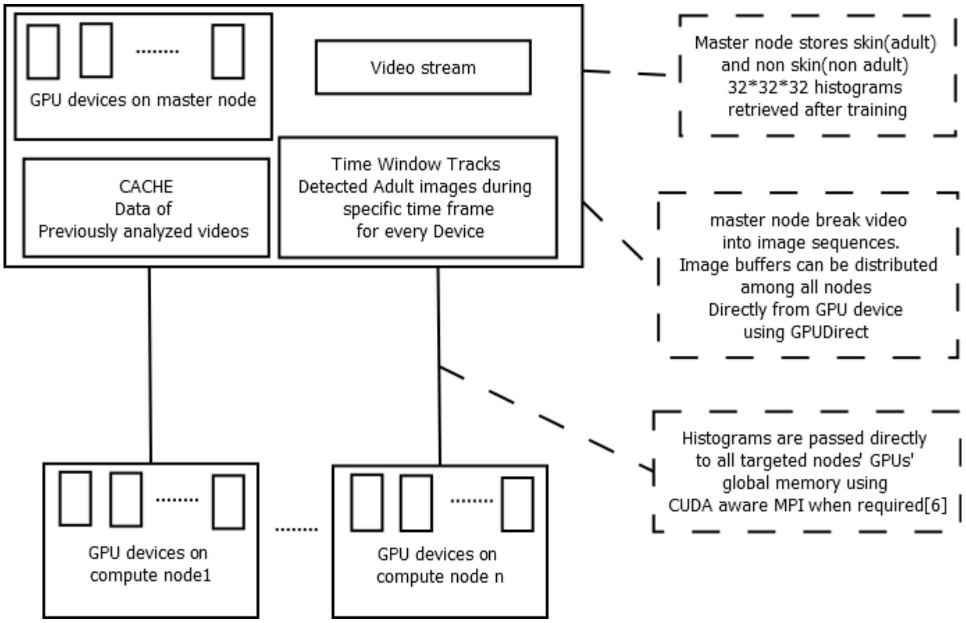
$$\frac{P(c/adult)}{P(c/nonadult)} > \theta \qquad (3)$$

Where, $\theta = K * \frac{1-P(adult)}{P(adult)}$

Now, considering this scenario in multiple GPUs, the execution environment will be different as all threads just need to read form previously stored histograms. **Step IV** is executed in following way.

1. Both adult and non adult dataset histograms of 32*32*32 are stored in the global memory of all the GPUs so every thread in every block on a GPU can access it.
2. Each block on single GPU works on single image frame. Threads of same block count *P(adult / c)* for every pixel in parallel and stores the total count in shared memory of block. This requires retrieving *P(c/adult)* and *(P(c/nonadult)* value from training data we calculated in **Step-I** and comparing their ratio withθ. Due to limited number of threads per block it may take few more cycles to cover the whole image i.e. each thread
3. Here, equation (3) shows that Prior Probability does not matter while deciding threshold because we can set K according as P (adult) changes as discussed in literature survey. Scanned Pixel is an adult pixel if the ratio is aboveθ.
4. Count of skin or adult pixels per image frame, stored in shared memory, is checked. When enough large number of skin pixels is found for any image in any shared memory, notification is sent to master node. Master node keeps a track of detected adult images for particular time window for every GPU device.
   If number of images from same device exceeds decided threshold number during specific time interval, download is stopped instantly. This technique helps reducing false positives for porn detection.
5. Process 2 is executed by every block on GPU in parallel. Number of blocks running parallel at the same time may be limited by particular GPU in case of resource scarcity.
6. Process 2 and 3 both run in parallel on multiple GPUs on multiple nodes.

Figure 3 demonstrates the scenario of proposed system for porn video detection where master node distributes the load among multi nodes and then gets updated from any node which detects porn image. At very moment, master node fires query to stop the download and updates cache of already analyzed video content.



Master node stores skin(adult) and non skin(non adult) 32*32*32 histograms retrieved after training

master node break video into image sequences. Image buffers can be distributed among all nodes Directly from GPU device using GPUDirect

Histograms are passed directly to all targeted nodes' GPUs' global memory using CUDA aware MPI when required[6]

Figure 3.Detailing of events

5. PERFORMANCE ANALYSIS

Massively parallel processing on distributed nodes results in very high performance gain. For training phase, mathematically, it is not possible to present exact computing overhead due to non coalesced memory access and shared memory bank conflicts even after optimized kernel.

However, effect of misaligned access will be less in GPUs with higher compute capabilities. Total number of threads running in parallel in a GPU is also limited by the number of Streaming Multi Processors (SMs) on GPU. Atomic Time to count number of pixels in each image plus compute overheadT_{sw}, plus T_{ld} – an initial load distribution time contributes to overall training time. As stated above, T_{sw} depends on the GPU architecture and strategy to use memory. Total training time on GPU is

$$T_{tr} = O(1) + T_{sw} + \mathrm{T}_{\mathrm{ld}} \qquad (4)$$

Training process for each colour of both adult and non-adult histograms,T_{tr} should take way lesser time than sequential approach.

As we are storing 32*32*32 histogram, memory on GPU may limit the number of images that can be processed on single GPU. If number of images handled on single device for training purpose at the same time is n then

$$n < \left(\frac{Total\ memory(GPU)}{memory\ for\ histogram * memory\ for\ image\ matrices}\right) \qquad (5)$$

When the number of GPUs is very limited and larger number of images (greater than n) is required to be processed then images can be queued for access on the host. Turn by turn, each chunk of images is transferred to GPU. Talking about main real time detection process, good amount of time is taken in transferring image buffers from master node to the other nodes' GPUs but comparatively less than the regular transfer due to GPUDirect. T1 is the time of overall serial approach of detecting porn video. T2 is overall time for proposed approach.

Then, ideally

$$T2 \cong \frac{T1}{\lambda * \eta} \qquad (6)$$

Where λ = Number of GPUs and η =Average number of threads working in parallel per GPU

Actually there is communication overhead and waiting time in specific cases as discussed earlier. So considering $\lambda * \eta$ as single computing unit

$$\sigma = \lambda * \eta$$

And overhead as T_{O} , equation (6) can be re written as

$$T2 \cong \frac{T1}{\sigma} + \mathrm{T}_{\mathrm{O}} \qquad (7)$$

In a case of multiple GPUs each having thousands of threads, T_{O} can be neglected.
So in spite of the communication, load distribution and waiting overhead,

$$T2 \lll T1 \qquad (8)$$

5. CONCLUSIONS

In this paper, we address recent and necessary problems of organizations about blocking access to porn video content from campus wide network without affecting access to educational videos. We have proposed a feasible solution of using nonparametric data mining approach for adult content detection in video on distributed multiple GPU system. This approach can achieve high parallelism due to image processing and data mining on the proposed system. Parallel time window handling for every device at master node helps in reducing overall false positives. Theoretical calculation shows that our model is expected to perform effectively fast and serve our purpose.

Given rich training dataset and proper infrastructure, proposed system can be implemented for the real time usage. We look forward to establish complete working system and thus address the challenges to be faced during implementation on distributed multi GPU environment.

REFERENCES

[1] Shirali- Shahreza, S. and Mousavi, M.E. 2008. A new Bayesian Classifier for Skin Detection. In 3rd international conference on innovative computing information and Control, (Dalian, Liaoning, June 18-20, 2008), IEEE, 172. DOI= http://dx.doi.org/10.1109/ICICIC.2008.54

[2] Choi, B., Kim, J., and Ryou, J. 2009. Adult image detection with close-up face classification.In international conference on consumer electronics, (Las Vegas, NV, January 10-14, 2009),IEEE,1,DOI=http://dx.doi.org/10.1109/ICCE.2009.5012282

[3] Jorge A. Marcial-Basilio, Gualberto Aguilar-Torres Gabriel Sánchez-Pérez and L. Karina Toscano.2011. Detection of Pornographic Digital Images. International journal of computers. 5(2). 298-305.

[4] Yin, L.; Dong, M, Deng, W., Guo, J., Zhang, B. and Tavel, P. 2012. Statistical Color Model Based Adult Video Filter IEEE International Conference on Multimedia and Expo Workshops, (Melbourne, VIC, July 9-13 2012), IEEE, 349-353, DOI= http://dx.doi.org/10.1109/ICMEW.2012.66

[5] Ulges, A., Schulze, C., Borth, D., and Stahl, A. 2012. Pornography detection in video benefits (a lot) from a multi-modal approach. Proceedings of the 2012 ACM international workshop on Audio and multimedia methods for large-scale video analysis,(Nara , Japan , November 2 2012). ACM, New York, NY, 21-26. DOI= http://dx.doi.org/10.1145/2390214.2390222

[6] Potluri,S., Hamidouche, K., Venkatesh,A., Bureddy,D. and Panda, D.K.. 2013. Efficient Inter-node MPI Communication using GPUDirect RDMA for InfiniBand Clusters with NVIDIA GPUs. 2013 42nd International Conference on Parallel Processing. (Lyon, October -4, 2013). IEEE, 80-89. DOI= http://dx.doi.org/10.1109/ICPP.2013.17

[7] Vezhnevets, V., Sazonov, V. and Andreeva, A. 2003. A survey on Pixel-Based Skin Color Detection Techniques. International Conference Graphicon. (Moscow, Russia).

[8] Jones, M.J. and Rehg, J.M. 2002. Statistical color models with application to skin detection, International Journal of Computer Vision, 46(1), Kluwer Academic Publishers,81–96. DOI= http://dx.doi.org/10.1023/A:1013200319198

[9] Song, F. and Dongarra, J. 2012. Ascalable Framework for Heterogeneous GPU-Based Clusters. Proceedings of the 24th ACM symposium on Parallelism in algorithms and architectures. (Pittsburgh, Pennsylvania, USA, June 25-27). SPAA'12, ACM, New York, NY, 91-100. DOI= http://dx.doi.org/10.1145/2312005.2312025

[10] Cao, Y., Ai, z. and Wang, H. 2013. A Distributed Multi-Node GPU Accelerated Parallel Rendering Scheme for Visualization Cluster Environment. 2013 International Conference on Virtual Reality and Visualization (Xi'an, September 14-15, 2013), IEEE, 153-160. DOI= http://dx.doi.org/10.1109/ICVRV.2013.32

[11] Fogal, T. Childs, H., Shankar, S., Kruger, J., R., Bergeron, D. and Hatcher, P. 2010. Large data visualization on distributed memory multi-GPU cluster.Proceeding of the Conference on High Performance Graphics. Eurographics Association, Aire-la-Ville, Switzerland, 57-66

[12] Wenger, S., Ament, M., Guthe, S. and Lorenz, D. 2012. Visualization of Astronomical Nebulae via Distributed Multi-GPU Compressed Sensing Tomography. IEEE Transactions on Visualization and Computer Graphics. 18(12).2188-2197. DOI=http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.281.

[13] Wasif, M.K. and Narayanan, P.J. 2011. Scalable clustering using multiple GPUs. 2011 18th International Conference on High Performance Computing (HiPC).(Bangalore, December 18-21, 2011), IEEE, 1-10. DOI= http://dx.doi.org/10.1109/HiPC.2011.6152713

[14] Harris, M. 2013. How to Access Global Memory Efficiently in CUDA C/C++ Kernels. Developer Blog [Online], Available: http://devblogs.nvidia.com/parallelforall/how-access-global-memory-efficiently-cuda-c-kernels/ .

[15] Wang, Y. 2012. A GPU Memory System Comparison for an Elliptic Test Problem. Retrieved January 20, 2004,[Online], Available: http://www.yuwang-cg.com/project1.html

[16] Nikam V.B., Meshram, B.B. 2013. Modeling Rainfall Prediction Using Data Mining Method: A Bayesian Approach. Fifth International Conference on Computational Intelligence. (Seoul, September 24-25, 2013), IEEE, 132-136. DOI= http://dx.doi.org/10.1109/CIMSim.2013.29

[17] V.B. Nikam, Kiran Joshi, B.B. Meshram, “An Approach For System Scalability for Video on Demand”, Interface,2011

[18] Shrikant Gond, Akshay Patil And V. B. Nikam, "A Survey On Parallelization Of Data Mining Techniques" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 3, Issue 4, Jul-Aug 2013, pp. 520-526 http://www.ijera.com/papers/Vol3_issue4/CK34520526.pdf .

Authors

Urvesh P. Devani is pursuing his Master of Technology in Information Technology (with specialization in Software Engineering) from VJTI, Matunga, Mumbai, Maharashtra state. His research interests include Parallel Processing, Video Analytics, Data mining, Big Data, High Performance Computing and Human Computer Interaction. He is an associate member of IETE.

Valmik B Nikam is Bachelor of Engineering (Computer Science and Engineering) from Government College of Engineering Aurangabad, Master of Engineering (Computer Engineering) from VJTI, Matunga, Mumbai, Maharashtra state, and pursuing PhD in Computer Department of VJTI. He was faculty at Dr. Babasaheb Ambedkar Technological University, Lonere. He has 12 years of academic experience and 5 years of administrative experience as a Head of Department. He has one year of industry experience. He has attended many short term training programs and has been invited for expert lectures in the workshops. Presently he is Associate Professor at deparment of Computer Engineering & Information Technology of VJTI, Matunga, Mumbai. His research interests include Scalability of Data Mining Algorithms, Data Warehousing, Big Data, Parallel Computing, GPU Computing, Cloud Computing. He is member of CSI, ACM, IEEE research organizations, and also a life member of ISTE. He has been felicitated with IBM-DRONA award in 2011.

B.B.Meshram is a Professor and Head of Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga, Mumbai. He is Ph.D. in Computer Engineering. He has been in the academics & research since 20 years. His current research includes database technologies, data mining, securities, forensic analysis, video processing, distributed computing. He has authored over 203 research publications, out of which over 38 publications at National, 91 publications at international conferences, and more than 71 in international journals, also he has filed two patents. He has given numerous invited talks at various conferences, workshops, and training programs and also served as chair/co-chair for many conferences/workshops in the area of computer science and engineering. The industry demanded M.Tech program on Network Infrastructure Management System, and the International conference “Interface” are his brain childs to interface the industry, academia & researchers. Beyond the researcher, he also runs the Jeman Educational Society to uplift the needy and deprived students of the society, as a responsibility towards the society and hence the Nation.