INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 6)
Available online at: www.ijariit.com

Survey on parallel data cloning and parallel programming

Abhay Nigam
abhay.nigam2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

Kartik Bansal
kartik.bansal2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

Devdutta Basu
devdutta.basu2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

Pranay Kavishwar
pranay.kavishwar2017@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

P. Sanjeevi
sanjeevi.p@vitbhopal.ac.in
VIT Bhopal University, Bhopal, Madhya Pradesh

ABSTRACT

This paper review about parallel data processing and parallel programming a parallel processing is now become extremely famous for improving the speed to process something in the paper we evaluated.1. Different features of parallel computing.2. Parallel programming models.3. And structured parallel programming. We will be also discussing the machine learning and the importance area where parallel computing can be useful in enhancing the capabilities of what machine learning can do and the generation models and method of parallel programming. We will be also simulating the advantage of parallel computing by comparing it with existing methods and how Hadoop can help for cluster handling.

Keywords— *Parallel computing, sfCluster, Data cloning, parallel programming*

1. INTRODUCTION

To investigate the parallel computer Architecture of both hardware and software aspects initiated in the IBM Research Division, in cooperation with the Ultra computer project of NYU. This paper discusses the machine designing in more detail and ideas that presents the performance analysis techniques and results that lead to the Parallel computer architecture design [1].

1.1 Parallelism

Parallelism is to diminish the turnaround time however even increase the CPU time because of overhead and increase the required memory because of copied information and instructions. Composing and error removal a parallel program is significantly more. Distinctive architectures, diverse programming models are appropriate for various applications thus the characteristics of the applications should make the decision for the determination of parallel architecture and furthermore the parallelization of the applications. A stream is only a succession of things (instruction or information) SISD: A kind of PC architecture in which there is a single instruction cycle and operands are gotten in serial design into a single processing unit before execution. SIMD: A kind of multiprocessor architecture in which there is a one command cycle, however, numerous arrangements of operands might be gotten to different preparing units and might be working upon all the while inside a one instruction cycle. MISD: is a kind of parallel figuring architecture where numerous practical units perform diverse activities on the same information. MIMD: A kind of multiprocessor design in which a few command cycles might be dynamic at any given time, each freely getting command and operands into numerous preparing units and working on them in a simultaneous manner [2].

1.2 Distinctive features of parallel computing

In parallel handling, the assignment is separated among different Processing Elements (PEs) that execute the employment in parallel. It is certainly accepted here that the errand is agreeable to parallel handling and correspondence system is input with the goal that PEs may chip away at the subtasks of the primary undertaking but then total the primary assignment as though the procedure is done on solitary virtual consecutive 3 registering machines. Correspondence worldview shows up at a cross street at this point. Correspondence issues in the prior consecutive PC were connected to interconnection of CPU components, for instance in the plan of CDC 7600 it was identified with empowering the CPU for skimming point division in a solitary cycle or in modem times making a super pipeline structure of Pentium for completing a single skimming point activity per cycle in a pipeline; while for parallel preparing correspondence it ordinarily alludes to correspondence among PEs on a naturally visible or net or at coarse-grained level. Execution of shared memory design in equipment is exceptionally requesting. In fact, it is as troublesome as making a kick the

bucket for a CPU chip. Moreover, as adaptability issues for such engineering have remained unanswered and if the example of improvement is any sign, it is settled in the negative. The present study alludes to the non-shared memory sort of correspondence issue [3].

1.3 Maps on parallel computers

The essential thought of parallel figuring is to disperse the computational outstanding task at hand for a solitary errand over an expansive number of processors. On a basic level, a parallel PC can possibly convey figuring control identical to the aggregate figuring intensity of the processors from which it is developed; a 100-processor machine can possibly convey 100 times the figuring intensity of a solitary processor. Practically, in any case, the execution that can be accomplished is in every case less proficient than this perfect. A consummately productive execution with N processors would give a factor N accelerate in calculation time; the proportion of the real speedup (1 to the perfect speedup N can fill in as a proportion of the effectiveness f of a parallel execution. (1 f= - N (1) For a given calculation, one of the variables that most impacts the general execution is how the calculation is mapped onto the accessible processors. The effectiveness of a specific mapping can be dissected as far as two essential factors: stack equalization and correspondence overhead. Load-balance in cloud [22-31] is a proportion of how consistently the computational remaining burden is conveyed among the accessible processors. Correspondence overhead, then again, is identified with the expense in a time of conveying data between processors [4].

2. CLUSTER HANDLING WITH HADOOP

The main design we portray goes for enhancing handling time for a substantial arrangement of records. It keeps up an essential setup where finish archives are gone through the whole pipeline applying person NLP modules in a settled request. We initially portray Hadoop and the Cascading approach we utilize. This is pursued by the primary consequences of our most recent handling. This pipeline performs NLP examinations on English. Comparative pipelines for Dutch, Spanish and Italian have been created in News Reader. We utilize the English pipeline in our tests since it is the most complex of the four. Errand talk about further advancements [5].

2.1 Foundation of Computer-Assisted Problem-Solving Environment (PSE)

IN SCIENTIFIC COMPUTING PSE is characterized as pursues: "A framework that gives all the computational offices fundamental to explain a target class of issues. It employments the dialect of the target class also, clients require not have specific information about the fundamental equipment or on the other hand programming". In registering sciences, we require PC control, brilliant calculations, what's more, programming control in arranges to understand logical what's more, building issues driving to logical disclosures what's more, improvement of inventive new items. The PC control, what's more, the processing calculations have been created phenomenally, what's more, have given colossal commitments to sciences, building what's more, creations. On the other hand, the programming control has not been created, all things considered, looked at with the PC control, what's more, the calculation control. Memory-administration for information parallel applications in implanted frameworks the productive acknowledgment of uses with multi-centre or many-centre processors in an installed framework is an awesome test. With application-particular models, it is conceivable to spare vitality, decrease idleness or increment throughput as indicated by the acknowledged tasks, in Parallel Embedded Computing Architectures differentiation to the use of standard CPUs. Other than the improvement of the processor engineering, additionally the mix of the centres in the inserted condition assumes a vital job. This implies, the quantity of connected cores1 and their coupling to recollections or transport frameworks must be picked painstakingly, to keep away from bottlenecks in the preparing chain [6].

3. MACHINE LEARNING

A standout amongst the most encouraging regions for the eventual fate of figuring is the utilization of measurable machine figuring out how to bode well from the immense measures of information now accessible because of quicker PCs, bigger plates, and the utilization of the Internet to interface them all together. Michael Jordan and Dan Klein, specialists in machine learning, discovered two smaller people that ought to be added to help machine learning: Dynamic writing computer programs is an algorithmic method that figures arrangements by explaining less complex covering subproblems. It is especially material for improvement issues where the ideal outcome for an issue is developed from the ideal outcome for the subproblems. Backtrack and Branch-and-Bound: These include illuminating different inquiry also, worldwide advancement issues for immovably expansive spaces. Some verifiable strategy is required keeping in mind the end goal to preclude areas of the inquiry space that contain no fascinating arrangements. Branch and bound calculations work by the gap and overcome guideline: the pursuit space is subdivided into littler subregions ("spreading"), and limits are found on every one of the arrangements contained in each subregion under thought [7].

3.1 The Numerical Aerodynamic Simulation

(NAS) The program, which is based at NASA Ames Research Centre, is a huge scale exertion to propel the condition of computational streamlined features. The NAS association points, to give the Nation's aviation inquire about and advancement network continuously 2000 a high-performance, operational registering framework able to do recreating a whole aviation vehicle framework inside a registering time of one to a few hours; (NAS Systems). The fruitful arrangement of this; grand challenge; issue will require the advancement of PC frameworks that can play out the required complex logical calculations at a managed rate almost 1,000 times more prominent than current age supercomputers can accomplish. The engineering of PC frameworks ready to accomplish this level of execution will probably be not at all like the common memory multiprocessing supercomputers of today. While no accord, however, exists on what the outline will be, it is likely that the framework will comprise of no less than 1,000 processors figuring in parallel [8].

4. PARALLEL PROGRAMMING GENERATION

Parallel computation is widely employed in scientific researches, engineering activities, and item development. Parallel program keeping in touch with itself is not always a basic assignment depending on problems comprehended. Substantial scale scientific

computing, huge data analyses, and precise visualizations, for example, would require parallel calculations, and the parallel computing needs the parallelization techniques. In this, a parallel program generation support is discussed, and a PC assisted parallel program age system P-NCAS is introduced. Computer-assisted problem-solving is one of the key methods to promote innovations in science and engineering and contributes to enriching our society and our life toward a programming-free environment in computing science. Problem-solving environments (PSE) research activities had started to enhance the programming power in the 1970's. The P-NCAS is one of the PSEs. The PSE concept provides an incorporated human-accommodating computational software and hardware framework to solve a target class of problems. In the PSE concept, human concentrates on target problems, and a part of problem-solving, which can be solved mechanically, is performed by computers or machines or software. The concept of the PC assisted program generation has been opening the new style of computer programming to reduce the programming hard task. Huge computer software may include errors and bugs. The errors or malfunction of the software infrastructure may induce uncertainty and in like manner serious accidents in our society. The programming process tends to include mechanical parts, which means mechanically programmable parts. When the application area of the software is limited to a reasonable size, a part of the software would be mechanically generated. For example, scientific inquiry about oriented programs would have a similar program structure depending on the numerical scheme. So, the present PSE systems including P-NCAS provide a new powerful tool for the programming help toward a programming-free environment [9].

5. PARALLEL PC ASSOCIATION

The host processor is the fundamental controller of the entire framework. The stream memory controller deals with the information exchange between the stream memory and the memory on the serial processor. The memory administration's capacity is to improve memory framework's information exchange speed to coordinate the coprocessor's processing prerequisite the host processor, the stream memory controller, the stream memory and the SIMD coprocessor are dependable on an equivalent chip and dependably the stream memory is built by enrolling records [10].

5.1 Parallel MCMC chains

Markov Chain Monte Carlo (MCMC) techniques speak to an "embarrassingly parallel issue" and endeavors are being made to empower powerful parallel registering strategies for Bayesian insights, yet prominent programming for fitting graphical models utilizing some tongue of the BUGS dialect (e.g. Win BUGS, Open BUGS, JAGS) don't, however, accompany an implicit office for empowering parallel calculations .Win BUGS has been censured that the independence of parallel chains can't be ensured by means of setting irregular number generator (RNG) seed. Therefore, parallelizing approaches based on Win BUGS (e.g. Grid BUGS) or bugs parallel Metrum Institute (2010)) are not most appropriate for a wide range of parallel Bayesian calculations (e.g. those are most appropriate for clump preparing, however not for producing genuinely autonomous chains). One may contend that the return times of RNG calculations are sufficiently substantial to be sheltered to utilize an alternate seeds approach, however, this methodology may be hazardous for long chains as parallel streams may have some cover, and parts of the successions probably won't be free). Additionally, utilizing legitimate parallel RNG of the rlecuyer bundle utilized in the R session to guarantee the autonomy of beginning qualities or arbitrary seeds won't ensure chain freedom outside of R (e.g. this is the situation in the bugs parallel venture as indicated by its source code). Arbitrary numbers for introduction in JAGS are characterized by the sort and seed of the RNG, and currently, there are four distinctive sorts of RNGs accessible.

5.2 Parallel Data Cloning

Information cloning utilizes MCMC strategies to compute the back conveyance of the model parameters contingent on the information. If the quantity of clones (k) is sufficiently huge, the mean of the back appropriation is the most extreme probability gauge and k times the back fluctuation is the relating asymptotic change of the most extreme probability gauge. In any case, there is no all-inclusive number of clones that is expected to achieve assembly, so the number of clones must be resolved observationally by utilizing e.g. the datable and dcdiag capacities given by the clone bundle. For this, we need to fit a similar model with the distinctive number of clones of the information and check if the changes of the parameters will zero. Fitting the model with a number of clones, say (1, 2, 5, 10, 25), implies that we need to fit the demonstration five times. Without parallelization, we can utilize the ensuing calls of the jags. Fit work with diverse informational collections relating to a various number of clones, or utilize the dc.fit wrapper capacity of the clone R bundle. Parallelization can occur in two different ways: (1) we parallelize the calculation of the MCMC chains (usually we utilize more than one bind to check legitimate blending conduct of the chains) [11].

5.3 Structured parallel programming

The PPE SkIE, originating from our work on parallel programming models and dialects, is in view of the idea of parallel coordination dialect. Coordination in SkIE pursues the parallel skeleton programming model. The worldwide structure of the program is communicated by the builds of the dialect, giving an abnormal state portrayal that is machine free. Skeleton-based models have numerous intense highlights, like compositionality, execution models and semantic-safeguarding changes permitting to characterize improvement procedures. The organized way to deal with coordination blends these points of interest without hardly lifting a finger of programming reuse. Since the compositional idea of skeletons, parallel modules can settle inside one another to create intricate, parallel structures from the basic, essential ones. The cooperation through unmistakably characterized interfaces makes autonomous the usage of various parallel and consecutive modules. The idea of module interface additionally facilitates the communication among various consecutive host dialects (like C/Cpp, Fortran, Java) and nature. The properties of the fundamental skeleton show can be misused for worldwide enhancements while holding the current successive instruments and advancements for the successive modules. All the low-level subtle elements of correspondence and parallelism administration are left to the dialect bolster. The SkIE-CL coordination dialect furnishes the client with a subset of the parallel skeletons considered in the writing. The general semantics of the skeletons is information stream like, with bundles of information we call assignments spilling between the interfaces of connected program modules. The least complex skeleton, the seq, is an intention to exemplify successive code from the different host dialects into a measured structure with very much characterized interfaces. Pipeline arrangement of various phases of a capacity is acknowledged by the pipe skeleton. The free utilitarian assessment over undertakings of a stream, in a heap adjusting design, is

communicated through the homestead skeleton. The specialist module contained in the homestead is consistently repeated, each duplicate working on a subset of the information stream. The circle skeleton is utilized to characterize cyclic, potentially interleaved, information stream calculations, where undertakings must be over and over figured until the point when they fulfill the condition assessed by a successive test module [12].

5.4 Parallel Scheme and its execution in TOUGH2

TOUGH2 (Pruess, 1991) is a broadly useful numerical recreation program for demonstrating multidimensional liquid and warmth streams of multiphase, multicomponent in permeable and cracked media. The parallel adaptation TOUGH2 code protects every one of the capacities and highlights of its unique one-processor form. It settles a similar arrangement of conditions for its scientific model. The numerical plan of the TOUGH2 code depends on the necessarily limited contrast (IFD) strategy (Narasimhan and Witherspoon, 1976). Protection conditions including the mass of air, water, and compound segments and warm vitality are discretized in space utilizing the IFD strategy. Time is discretized completely verifiably, utilizing a first-arrange in reverse limited distinction conspire. The subsequent discretized limited contrast conditions for mass and vitality adjusts are nonlinear and are tackled utilizing the Newton/Raphson conspire. In this paper, in any case, just the module for settling Richards' condition of the parallel code is connected to the Yucca Mountain field scale issue Fork or join design • A primary UE forks off some number of different UEs that at that point proceed in parallel to achieve a few parts of the general work. • Parent assignments make new errand (fork) at that point holds up until the point when all they finish (join) before proceeding on with the calculation Patterns for parallel programming Timothy et al. Para Prof Hyesoon Kim [13].

6. sfCLUSTER

sfCluster is a Unix command line apparatus which is worked to deal with group setup, checking and shutdown naturally and in this manner, conceals these assignments from the client. This is done as securely as conceivable empowering group registering notwithstanding for unpractised clients. Utilizing snowfall as the R frontend, clients can change asset settings without changing their R program. sfCluster is composed in Perl, utilizing just Open Source instruments. On the backend, sfCluster is as of now based upon MPI, utilizing the LAM usage (Burns et al., 1994), which is accessible on most basic Unix disseminations. Essentially, a bunch is characterized by two assets: CPU and memory. The observing of memory utilization is exceptionally vital, as a machine is for all intents and purposes unusable for superior purposes on the off chance that it is coming up short on physical memory and begins to swap memory on the hard circle. sfCluster can test memory utilization of a program naturally (by running it in the consecutive mode for a specific time) or set the upper bound to a client given esteem. Message Passing Interface • MPI is a message-passing library interface determination • MPI isn't a programming dialect • MPI isn't a usage • MPI characterizes just the interface for the correspondence systems • MPI utilization is more dynamic and more confuse than OpenMP • MPI-1 of every 1994, MPI-2 of every 1997, MPI-3 of every 2012 and MPI-3.1 out of 2015 • MPI-3.1 backings Fortran 2008 dialect ties and non-blocking aggregate correspondences. [14].

7. DISAPPOINTMENT DETECTION AND RECUPERATION

Disappointment recognition utilizes the RPVM work. PVM. pstats, this profits procedure status and enables us to effortlessly distinguish fizzled hubs. If a hub comes up short, the recuperation method replaces it with another made hub. At that point, any calls and statements made on the hub before the calculation must be made. Going back to the initial model, these futures assessing the library articulation, calling myinitfun what's more, introducing the arbitrary number generator (RNG). For the use in clusterApplyFT, the client characterizes a capacity containing all calls that are made preceding the calculation and passes it to clusterApplyFT as contention. Since disappointment identification should be possible quicker than message exchange, results may touch base after the producing computational hub has been recognized as fizzled and the suitable recuperation has been made. Subsequently, with a specific end goal to distinguish the arrived result, it is helpful to keep a rundown of the fizzled hubs. An elective method for executing disappointment location is utilizing the work. PVM.notify. For this situation, the ace hub is automatically [15].

7.1 Compiler and Runtime System

The assemblage of an application show into code that can keep running on different GPUs dispersed over a system is done progressively. At the point when a Cog application is begun, the client's model of communicating dynamic fields is parsed, powerfully meant GPU code, streamlined, parcelled and put onto accessible GPU assets, and downloaded into the GPUs for execution. The Cog runtime framework arranges the GPUs, trading dynamic field information between GPUs on various system hubs, and arranges the calculation [16].

8. INFORMATION PROPERTIES

At every accumulation point we gather the number of columns furthermore, the normal line length. is measurement will illuminate regardless of whether information develops or recoils and by how much as it owes through the question tree. Creating these statistics is easy– for instance, the aggregate number of columns are a select task is just the aggregate of the number of columns seen by the authorities that watch the yield of that select over every one of the undertakings that contain that select [17].

8.1 Parallelisation Strategy of Varsha Model

Huge application codes created till late 70's or mid 80's proliferate; these codes are created around consecutive PC having von Neumann's design. Expansion of these codes both regarding extension and proficiency is a whiz necessity which can just happen through the main conceivable course, i.e. parallel courses. Such codes are generally called heritage code. Code VARSHA utilized for climate expectations, for the most part, comprises of numerical calculation of conditions in the unearthly space and correspondence of information at each time advance of figure that requests higher data transmission [18].

8.2 MPI (Message Passing Interface)

Message Passing Library intended for parallel programming through circulated shared memory. Basic correspondence is taken care of through send() and get() crude for passing a message. It gives standard message passing activity with synchronous and nonconcurrent variations. It gives convenient proficient and blames tolerant condition for parallel programming on shared memory. It gives an intense and general method for communicating parallelism [19].

9. VERSATILE PARALLELISM

Give n a chance to be the aggregate number of things in the database. At that point, there are possibly visit k-item sets that we would need to tally amid cycle k. In any case, in rehearsing the number is normally significantly littler, as is demonstrated by our test results. We found that help tallying overwhelmed the execution time to the tune of around 85% of the aggregate calculation time for the databases we consider in Segment 6. Then again, for emphasis with a substantial number of k item sets there was sufficient work in the hopeful age stage. This recommends a need for some type of dynamic or versatile parallelization considering the number of visit k-item sets. On the off chance that there are not a sufficient number of incessant item sets, at that point, it is better not to parallelize the applicant age [20].

10. CONCLUSION

In this paper, we have described what parallelism in computing is and what are its advantage over existing computing models, about its features, how it works and maps on parallel computing. Also, we have seen how Hadoop can be useful for cluster handling and how machine learning can be enhanced by the use of parallel computing. Parallel computing also requires different programming implementation so we have also covered parallel programming generation and its concepts. Then we discussed parallel data cloning and sfCluster. At last, we talked about the parallelization strategy of Varsha model, MPI, and versatile parallism.

11. REFERENCES

[1] G. F. Plister, W. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfelder, K. P. McAuliffe, E. A. Melton, V. A. Norton and J. Weiss, "The IBM RP3: introduction and architecture", 764-771 1985.

[2] Prabhudev S Irabashetti1, Gawali Anjali B, Betkar Akshay S," Architecture of parallel processing in computer organization ", 12-17, 2002.

[3] Mihalca, Loredana; Miclea, Mircea, "Survey of Parallel Computing Trends in India – her Contributions to Communication Technology" 1-33,2007

[4] Mark E. Nelson James M. Bower, "Computational Efficiency: A Common Organizing Principle for Parallel Computer Maps and Brain Maps?" 60-67. 1990

[5] Mathijs Kattenberg, Zuhaitz Beloki, Aitor Soroa, Xabier Artola, Antske Fokkens, Paul Huygen "Two Architectures for Parallel Processing of Huge Amounts of Text ", 4513-4519.2015

[6] Shigeo Kawata Graduate School of Engineering, Utsunomiya University, "Computer Assisted Parallel Program Generation", Japan, 1-10.2015

[7] Krste Asanovic, Ras Bodik, Bryan Christopher Catanzaro, Joseph James Gebis, Parry Husbands, Kurt Keutzer, David A. Patterson, William Lester Plishker, John Shalf Samuel Webb Williams Electrical Engineering and Computer Sciences, University of California at Berkeley,"The Landscape of Parallel Computing Research:A view from Berkeley" , 1-54, December 18, 2006.

[8] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga "THE NAS PARALLEL BENCHMARKS",1991

[9] Nancy Hitschfeld-Kahler and Luis Mateu, "A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures", Vol. 15, No. 2, pp. 285-329, 2013.

[10] Zongtao Duan, Tao Lei and Haiwei Fan, Information Engineering School, Chang'an University, Xi' an, 710064, China, "Parallel Computing System for Image Intelligent Processing", ISSN 1812-5638, 1-5, 2012.

[11] Peter Solymos (University of Alberta), "Parallel Computing with Bayesian MCMC and Data Cloning in R with the dclone package", 1-12.2011

[12] Massimo Coppola, Marco Vanneschi, Dipartimento di Informatica, Universita di Pisa, Corso Italia 40, 56125 Pisa, Italy, "High-performance data mining with skeleton-based structured parallel programming", 793-813, 2002.

[13] Keni Zhang, Yu-Shu Wu, and G. S. Bodvarsson, Earth Sciences Division, Lawrence Berkeley, National Laboratory Berkeley, "Parallel Computing Simulation of Fluid Flow, in the Unsaturated Zone of Yucca Mountain, Nevada", CA 94720 1-38.2003

[14] Jochen Knaus, Christine Porzelius, Harald Binder and Guido Schwarzer, "Easier Parallel Computing in R with snowfall and sfCluster", ISSN 2073-4859, The R Journal Vol. 1/1, 54 – 59, May 2009.

[15] Hana Sev˘c´ıkov´a and A.J. Rossini. "Pragmatic parallel computing". 1-8,2004.

[16] Ll.M. Evansa, b, *, J.D. Arregui-Menab, P.M. Mummeryb , R.J. Akersa , E. Surreya , A. Shterenlikhtc, M. Broggid , L. "Use of massively parallel computing to improve modeling accuracy within the nuclear sector" (16) 15957, 2014.

[17] Michael Garland Scott Le Grand John Nickolls NVIDIA Joshua Anderson Iowa State University and Ames Laboratory Jim Hardwick TechniScan Medical Systems Scott Morton Hess Everett Phillips the Yao Zhang University of California, Davis Vasily Volkov University of California, Berkeley "PARALLEL COMPUTING EXPERIENCES WITH CUDA",13-27, 2008.

[18] Mohammed Mahfooz Sheikh*, A M Khan*, "bandwidth Requirements of Large Scale Computing Systems – A Case Study" Volume 79 – No4, 19-23, October 2013.

[19] Helly M. Patel, Krunal Panchal, Prashant Chauhan, M. B. Potdar," Large Scale Image Processing Using Distributed and Parallel Architecture", Vol. 6 (6), ISSN 0975 -9646, 5531-5535, 2015.

[20] Systems S. Parthasarathy1, M. J. Zaki2, M. Ogihara3, Parallel Data Mining for Association Rules on Shared-Memory Systems, 1-29, 2001.

[21] P. Sanjeevi and P. Viswanathan, 'NUTS scheduling approach for cloud data centers to optimize energy consumption', Computing (Springer), Vol. 99, No. 12, pp. 1179-1205, 2017.
[22] P. Sanjeevi and P. Viswanathan, 'Workload Consolidation Techniques to Optimize Energy in Cloud: Review', Int. J. of Internet Protocol Technology, Vol. 10, No. 2, pp. 115–125, 2017.
[23] P. Sanjeevi and P. Viswanathan, 'DTCF: Deadline Task Consolidation First for energy minimization in cloud data centers', International Journal of Networking and Virtual Organizations, Inderscience, Vol. 19, No. 3, pp. 209–233, 2017.
[24] P. Sanjeevi and P. Viswanathan, 'Employing Smart Homes IoT Techniques for Dynamic Provision of Cloud Benefactors', Int. J. of Critical Computer-Based Systems, Inderscience, Vol. 7, No. 3, pp. 209–224, 2017.
[25] P. Sanjeevi and P. Viswanathan, 'A survey on various problems and techniques for optimizing energy efficiency in cloud architecture', Walailak Journal of Science and Technology, Vol. 14, No. 10, 2017
[26] P. Sanjeevi, P. Viswanathan, M. R. Babu, and P. V. Krishna, 'Study and Analysis of Energy Issues in Cloud Computing', International Journal of Applied Engineering Research, Vol. 10, No. 7, pp. 16961-16969, 2015.
[27] P. Sanjeevi, G. Balamurugan, and P. Viswanathan, 'The Improved DROP Security based on Hard AI Problem in Cloud', Int. J. of Internet Protocol Technology, Vol. 9, No. 4, pp. 207-217, 2017.
[28] P. Sanjeevi and P. Viswanathan, 'A green energy optimized scheduling algorithm for cloud data centers', IEEE International Conference on Computing and Network Communications, Trivandrum, pp. 941-945, 2015.
[29] P. Sanjeevi and P. Viswanathan, 'Towards energy-aware job consolidation scheduling in the cloud', International Conference on Inventive Computation Technologies (ICICT 2016), IEEE Xplore, pp. 361- 366, 2016.
[30] G. Kesavan, P. Sanjeevi and P. Viswanathan, 'A 24 hour IoT framework for monitoring and managing home automation', International Conference on Inventive Computation Technologies (ICICT 2016), IEEE Xplore, pp. 367-371, 2016.

