IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Effective Strategy Identification for Parallel Job Execution Job Partitioning, Requirement Gathering and Allocation Strategies

Jasleen Kaur[1*], Anil kumar[2]

Department of Computer Science and Engineering, Guru Nanak Dev University, Amritsar, Punjab, India

**Corresponding Author: jasleenahuja8@gmail.com, Mobile.: +91 7986608730*

Available online at: www.ijcseonline.org

Accepted: 18/Jan/2019, Published: 31/Jan/2019

***Abstract*-**Parallel and distributed computing becomes critical in the heavy workload environment. In such situations, job partitioning becomes need of the hour. Smaller junks known as task has limited complexity and hence overall execution speed increased considerably as these allotted to the processors. In case of parallel computing, there exist several distinct tasks that may belong to single or multiple jobs having resource requirements. Assigning resources to tasks need strategies to reduce execution time and prevent starvation. This literature put a light on strategies used to allocate resources optimally to tasks meant to execute on distributed environment. Highlights of distinct literature presented through parameters in the form of comparative table so that useful feature can be extracted for future enhancements.

***Keyword**:* Parallel and distributed computing, execution speed, starvation

I. INTRODUCTION

Traditionally, computer software is written to execute instructions in serial manner. This means as current instruction finished execution only then next instruction comes in processing. This causes slow processing and may not be suitable for heavy and time critical applications. This leads to formation of parallel computing. In parallel computing, multiple processing elements concurrently used to execute tasks. This form of computing divides the entire jobs into tasks and then allocation phases begins. Allocation depends upon the resource requirement possessed by tasks. Resource requirement from jobs could lead to degradation in performance since overhead of computation leads to increase in execution time of jobs. Overall process of job execution partitioned into phases. These phases includes a) Job partitioning b) requirement gathering c) compatibility identification d) allocation.

I. Job Portioning

Multiprocessor systems plays critical role in the execution of jobs parallel. These systems consist of cores associated with each processor. Each core contains resources that could be allotted to jobs. It becomes crucial to test resource requirement of jobs with the resources possessed by cores. In order to establish compatibility of jobs with resources posses by cores, partitioning of jobs becomes critical. To partition the jobs, threshold value that could be average of resources possessed by cores could be used. Let 'n' be the threshold value corresponding to resources of jobs, 'm' indicates current job then total tasks corresponding to current job is given as

$$Total_{Tasks} = \frac{m}{n}$$

Equation 1: Task partitioning equation

These tasks then moved to the next phase to gather requirements.

1.1 Requirement gathering

Requirement gathering is the state of the art problem and must be rectified. To tackle the issue of requirement gathering, pattern recognition mechanism can be used. Requirement gathering using pattern recognition can categorise jobs into two categories.

- CPU bound
- Input output bound

CPU bound processes have demanding configuration and requires more processor and less input and output. Vector processing appears under this category. I/o bound processes on the other hand needs more input and output rather than CPU. To enhance CPU performance, process having mixture of both CPU and I/o must be selected for execution at first place.

Gathered requirements from processes must be compared against the resources possessed by cores known as compatibility identification.

1.1.1 Compatibility identification

Compatibility identification involves checking the resource availability with the cores and resource requirement of the tasks. Let 'p' indicates requirements of the task and 'q'

indicates the resources possessed by cores then following inequalities hold

$$Compatibility = \begin{cases} 1 \; if q \geq p \\ 0 \; if \; q < p \end{cases}$$

Equation 2: Compatibility identification equation
Compatibility if contains '0' then resources cannot be allotted and next core will be examined for resource allocation.

1.1.2 Resource allocation
Once compatible core identified, resource allocation phase begin. Core assigned to task could not be prompted. In other words cores remain assigned to the tasks until burst time of task expires. Result in terms of execution time could be presented.

Rest of the paper is organised as under: section 2 gives the comprehensive evaluation of existing literature along with comparative analysis, section 3 gives our contribution, section 4 gives conclusion and future scope, section 5 gives the references from which existing literature is drawn.

Analysis Of Existing Job Execution Techniques In Parallel Computing
This section presents techniques used to obtain tasks from jobs depending upon resource possessed by cores. In addition techniques of requirement gathering, compatibility identification and resources allocation discussed in brief with comparative table.

II. Task partitioning approaches

Tasks partitioning can be accomplished either by the use of static or dynamic model. The static approach gives better result in case of parallel environment where each job is approximately of same size or scale in terms of resource requirement.

Static Partitioning proposed by [1]uses graph based partitioning in which mapping function maps the tasks to processing element. Task partitioning strictly based on threshold values corresponding resource availability. This approach uses fixed partitioning rules for every job causing execution speed of jobs to enhance that matches the desired qualification. Problem of under execution of jobs leads to problems and requires dynamic job partitioning mechanism. [2] proposed dynamic partitioning mechanism in which proportionate allocation mechanism discussed through real time tasks using multi core processors. Global scheduling becomes critical and heavily used under this situation. Least resource wastage appears in this case. The problem with this approach however in the requirement gathering that could be hectic in nature.

Discussed techniques in literature are highlighted as under

Table 1: Comparison of partitioning strategies

Technique Used	Merit	Demerit
Static partitioning	• Execution speed for deterministic jobs is high. • Jobs with equal resource requirements could be efficiently served through this approach	• Tasks with less requirements waste resources • Tasks with more resource requirements could be starved
Dynamic Partitioning	• Resource requirements efficiently matches with the resource availability • Starvation problem common with static partitioning is resolved using dynamic partitioning.	• Resource utilization and complexity increases subsequently. • Overall cost of execution due to extra methodology employed increases and may not be suitable for less quantity of jobs.

Requirement gathering
This is one of the most critical and difficult phase corresponds to task successful allocation. Requirement gathering process includes machine learning and pattern mining approaches.

Alzahrani, (2016) proposed data mining method for anomaly prediction [3] for this purpose sequential data mining is used in order to accomplish this data pre-processing mechanism is applied. After applying pre-processing mechanism the attributes will be analyzed this will be done using passes on medical data. The first pass determines whether support for each disease is present or not at the end of this phase the frequent disease within the database will be identified, a counter will be maintained to count the occurrence of each disease within the dataset. Next phase determines the second sequence of diseases present within the dataset. The overall process yield the diseases which can cause the occurrence of other diseases. The disease resulting in another disease is termed as candidate generation. And for declaring that it is generated from the previous level Pruning is used.

Alamanda et.al. (2017), proposed sequence pattern mining in order to detect the time duration used for promotion [4] the sequence or pattern is checked from within the database. The

weight of each sequence in each database is achieved from the interval of the successive element in the sequence and the mining is performed on the basis of weight considering time interval. Time interval based pattern is used in this case. In pre-processing missing values are not considered.

Ahmed (2017), proposed an application that utilizes the data mining technique to predict the heart disease[5]. Also it guides the object to take caution at early stage. But is completely dependent upon object input and does not considered predefined dataset values. It also not utilizes the missing value that are essential to predict diseases.

Abbasghorbani et. al. (2015), conducted analysis of various pattern mining techniques are done and also the features of all the algorithms. It introduced various minimizing support counting which is used for minimizing search space[6]. We have generated small search space which will include earlier candidate sequence pruning then database is analyzed and compression technique is used to analyze.

Béchet et al. (2012), proposed paper presents the sequential pattern mining to discover the rare disease within human body where experiments are conducted using data mining tool WEKKA[7]. This show betterment in percentage for classification accuracy.

Chen et al. (2017), used a pattern growth method to analyze the medical database to specify the combination of chronic disease[8].It introduce prefix span algorithm that identify all possible patterns in the images but it constrained only specific disease and can further improved for efficient search, it shows the results in terms of HTN and DP diseases.

CHENG et al. (2017), proposed a sequential mining approach for early assessment of chronic disease[9]. The clinical database is considered .A dataset of objects derived from Taiwan, it derives richest of risk patterns. Data pre-processing as performed to rectify the problem if found but missing values are not considered .sequential pattern mining is used to observe the risk pattern and generate the result. The problem with this approach is that no precautions have been suggested. The classification accuracy is 80% further improvement in classification is needed. The chronic disease is analyzed in this paper built in over the existing problem.

Eenan (2009), proposed a non-homogeneous mark over model[10] which is used to identify the chronic disease in the objects. The algorithm uses global optimization that efficiently identifies the number of frequent pathway required to analyses the object. The result shows that the proposed methodology probability is better than existing ones but this approach can be extended using admission scheduling policy.

The comparative analysis of techniques give in table 2 used for requirement gathering suggest appropriate solution to be used in future for result improvement in terms of execution time.

Table 2: Requirement gathering mechanisms

Technique	Merit	Demerit
Sequential pattern mining	• Large item set can easily be categorised • Easy to implement • Concurrency can easily be accommodated within this algorithm	• Memory resident approach causes high space utilization. • It is iterative approach and can cause execution time to increase beyond specified deadline
Pattern Growth	• Feedback system is employed where generated output becomes input for the next phase • Suitable for small to intermediate item-set	• Not suitable for large item-set • Execution time and space requirements varies as size of item-set increases.

II.1 Compatibility identification mechanisms

[11] proposed a multi model approach that is used to identifying parameter for scheduling. It also describes non linear process that identifies the ARX and also dynamic local parameters. The results indicate effective identification for optimal scheduling. It introduces a scheduling variable named Markov jump process that describes different mode to set random process. It identifies the scheduling parameter effectively but does not give efficient processing of resources.

[12] describes a bottleneck identification based algorithm that identify the sequence of operations in a machine that executes different tasks. It presented average flow time of all operations and mutation process is also constructed. This mutation process gives the priority to each operation and

avoids the premature operations. The results show validity and efficiency of the algorithm. It reduces the flow time but burden of computing resources have been enhanced. The search efficiency is high.

Table 3: Comparison of compatibility establishing algorithm

Technique	Merit	Demerit
LPV approach	• Identify parameter for scheduling • Used for estimating dynamic local parameters • Identification process is efficient	• Resources cannot be handled effectively • Sequence can be generated randomly so starvation must be there
Differential based algorithm	• Utilizes the genetic approach to identify sequences • Avoids premature operations	• Burden of computing resources have been enhanced • Space requirements are more

II.1.2 Resource allocation

[13] proposes cooperation based strategy for resource allocation that allocates resources with cooperation of the nodes in parallel system. It first describes significance of resource allocation in parallel system and then approximate optimal solution for NP hard problems. The results indicate that resources are allocated fast and efficiently. But as the nodes increases the performance degrades.

[14] proposes and approach named FRAME that efficiently manages multiple parallel jobs and allocate resources to them efficiently. It enhances the overall utility and fairness of processors that handles the resources for jobs in parallel system. It is used for spatial allocation of resources in system. The time for allocation of resources is more so it enhances the waiting time.

Table 4: Allocation algorithm comparison

Technique	Merit	Demerit
Cooperation based algorithm	• Approximate the optimal solution then allocate the resources • Allocation of resources is fast and efficient	• Performance degrades as the nodes increases • Flow time is more
Fair resource allocation algorithm	• Utilizes the linear programming based on resource allocation. • Processor utilization is more	• Waiting time is more • Space requirements are more

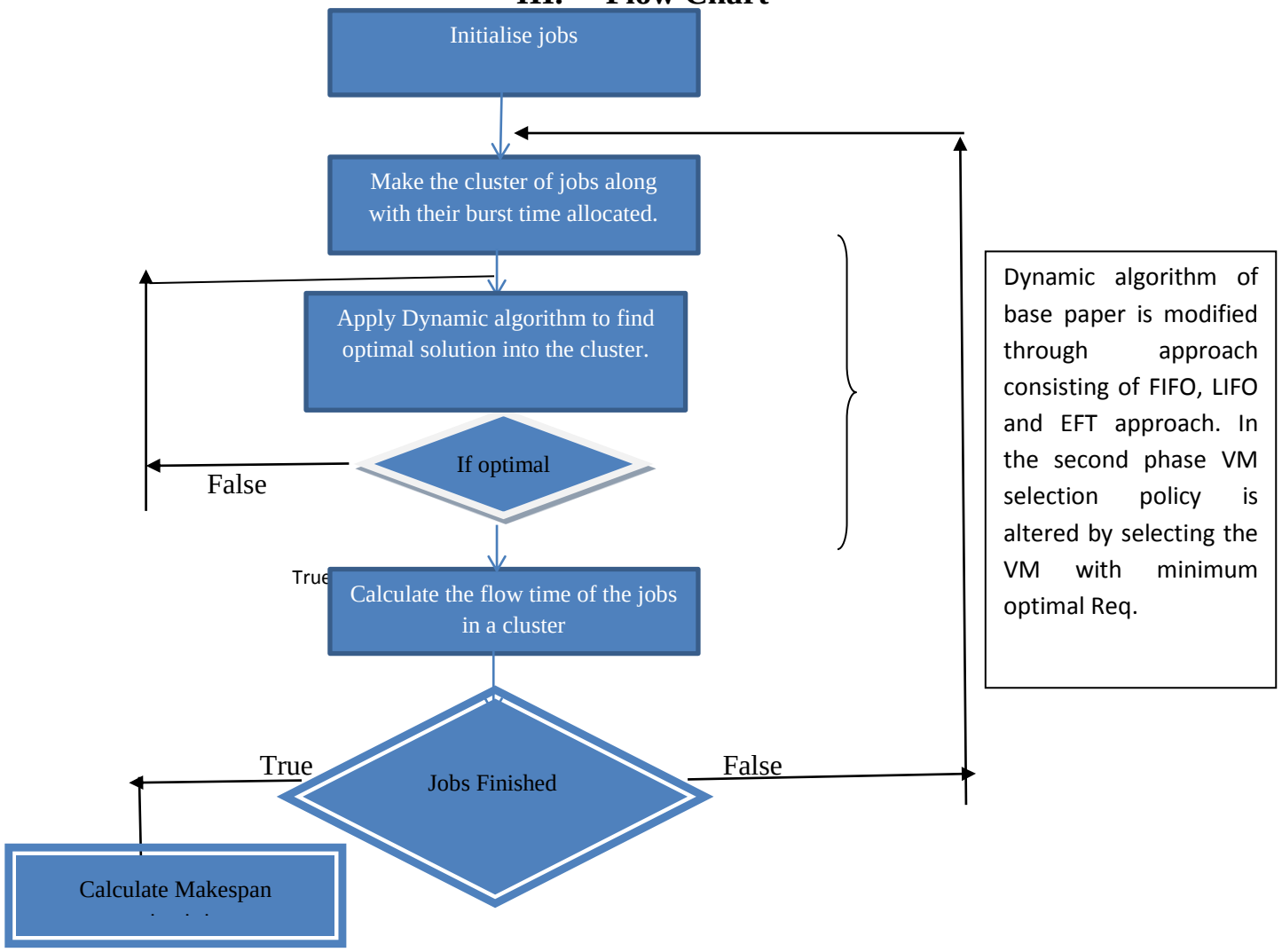
From the literature, some positive and some negative characteristics are extracted. Modification to existing literature in terms of cluster formation and requirement identification can be made. The requirement extraction process can enhance the performance of the system in terms of execution time. in addition allocation process can be collaborated with the proportionate algorithm.

II. PROPOSED WORK

The main objective of this literature is to identify the problems that could appear as jobs begin to execute on processing element within distributed and parallel system. Most of the existing system do not consider pattern recognition algorithm for requirement analysis. This mechanism can effectively gather requirements from the jobs and then allocation strategies that could be static or dynamic in nature can be applied. This could allow better identification of resource requirements and allocation that leads to removal of starvation problem and reduced Makespan and Flowtime. Metric consider for the same includes

- Makespan
- Flowtime
- Waiting time
- Overall execution time

III. Flow Chart



IV. CONCLUSION AND FUTURE SCOPE

The proposed work analyse the mechanism used for job partitioning, requirement gathering, compatibility identification and then allocation. The job partitioning mechanism divides the jobs into tasks depending upon the resources possessed by cores. Requirement analysis gathers the requirement from the tasks for effective allocation. In most of the existing algorithms for parallel job execution does not focus on requirement gathering before allocation. The pattern recognition mechanism can be used for requirement gathering as discussed in literature section 2.1. In addition FP growth algorithm provide state of the art solution to the correct identification of requirements for that starvation problem can be completely removed from real time task allocation.

REFERENCES

[1] A. Vasudevan, "Static Task Partitioning Techniques for Parallel Applications on Heterogeneous Processors," *Trinity Coll. dublin*, no. December, 2015.

[2] N. Saranya and R. C. Hansdah, "Dynamic partitioning based scheduling of real-time tasks in multicore processors," *Proc. - 2015 IEEE 18th Int. Symp. Real-Time Distrib. Comput. ISORC 2015*, pp. 190–197, 2015.

[3] M. Y. Alzahrani, "Discovering Sequential Patterns from Medical Datasets," 2016.

[4] S. Alamanda, S. Pabboju, and N. Gugulothu, "An Approach to Mine Time Interval Based Weighted Sequential Patterns in Sequence Databases," *2017 13th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, pp. 29–34, 2017.

[5] F. Ahmed, "A Simple Acute Myocardial Infarction (Heart Attack) Prediction System Using Clinical Data and Data Mining Techniques," pp. 22–24, 2017.

[6] S. Abbasghorbani and R. Tavoli, "Survey on Sequential Pattern Mining Algorithms," *2015 2nd Int. Conf. Knowledge-Based Eng. Innov.*, pp. 1153–1164, 2015.

[7] N. Béchet, P. Cellier, T. Charnois, B. Cremilleux, and M. C. Jaulent, "Sequential pattern mining to discover relations between genes and rare diseases," *Proc. - IEEE Symp. Comput. Med. Syst.*, 2012.

[8] C. J. Chen, T. W. Pai, S. S. Lin, C. C. Yeh, M. H. Liu, and C. H. Wang, "Application of PrefixSpan Algorithms for Disease Pattern Analysis," *Proc. - 2016 Int. Comput. Symp. ICS 2016*, pp. 274–278, 2017.

[9] Y. CHENG, Y.-F. Lin, K.-H. Chiang, and V. Tseng, "Mining Sequential Risk Patterns from Large-Scale Clinical Databases for Early Assessment of Chronic Diseases: A Case Study on Chronic Obstructive Pulmonary Disease," *IEEE J. Biomed. Heal. Informatics*, pp. 1–1, 2017.

[10] B. R. M. Eenan, "Non-homogeneous Markov models for sequential pattern mining of healthcare data," *IEEE*, pp. 327–344, 2009.

[11] J. Wei, Y. Yin, and F. Liu, "Multi-model LPV approach to CSTR system identification with stochastic scheduling variable," *Proc. - 2015 Chinese Autom. Congr. CAC 2015*, pp. 303–307, 2016.

[12] Y. Sun and P. Jiang, "A Novel Bottleneck Identification Based Differential Evolution Algorithm for Scheduling Complex Manufacturing Lines," *Proc. - 2016 3rd Int. Conf. Inf. Sci. Control Eng. ICISCE 2016*, pp. 774–778, 2016.

[13] J. Dai, B. Hu, L. Zhu, H. Han, and J. Liu, "Research on dynamic resource allocation with cooperation strategy in cloud computing," *2012 3rd Int. Conf. Syst. Sci. Eng. Des. Manuf. Informatiz. ICSEM 2012*, vol. 1, pp. 193–196, 2012.

[14] A. Mohtasham, R. Filipe, and J. Barreto, "FRAME: Fair resource allocation in multi-process environments," *Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS*, vol. 2016-January, pp. 601–608, 2016.

