Parallel Processing Problem and Solution - A Case Study on MATLAB Parallel Computing Toolbox Batch Processing

Nazurah Kamil [#1], Mohamed Faidz Mohamed Said [#2]

[#] *Faculty of Computer & Mathematical Sciences, Universiti Teknologi MARA*
70300 Seremban, Negeri Sembilan, MALAYSIA

[1] nazurahkamil95@gmail.com

[2] faidzms@ieee.org

***Abstract*—MATLAB or matrix laboratory is a multi-paradigm mathematical calculation and fourth-creation programming language. Parallel Computing Toolbox (PCT) solve computationally and data-intensive problems utilising multicore processors, GPUs, and computer clusters. Batch processing machines manage several jobs synchronously as a batch. The toolbox provides twelve workers, MATLAB computational engines, to execute applications locally on a multicore desktop. The same application on a computer cluster or a grid computing service can be run without changing the code. In the parallel-batching scheduling issue, some jobs can be processed as a batch at the same time on a machine at one time. When a batch machine completes the facilities of a batch and there is at least one product waiting in the queue, a real-time control decision is made to choose whether to start a job with a limited batch or wait until future job arrivals arise. This paper provides reviews of the case study on MATLAB Parallel Computing Toolbox batch processing.**

Keywords: **batch processing, job arrivals, MATLAB**

I. INTRODUCTION

Parallel computing speeds up the computational process and has a larger memory pool. Next, parallel computing mixes task-parallel and serial code in the same function and uses code analyzer to help in converting existing for-loop into parfor-loop.

Moreover, it runs loops on a pool of MATLAB resources. The toolbox gives twelve workers, namely MATLAB computational engines, to execute applications locally on a multicore desktop [1]. Parallel computing can also be upgraded from interactive to scheduling and for prototyping and quick access to MATLAB workers. However, scheduling offloads work to other MATLAB workers which are local or in a cluster, provides access to all the more computing property for enhanced execution. It will authorize local MATLAB session. Furthermore, by submitting parallel MATLAB jobs from a user's desktop, it will restore the outcomes when the job is completed. In summary, it uses batch for off-loading work. Running applications interactively are appropriate when execution time is moderately short. At the point when the applications need to keep running for quite a while, the toolbox will be used to keep the running as batch jobs. MATLAB can free its session for different activities, while large MATLAB and Simulink applications are executed. On the other time, MATLAB session can be shut down to retrieve results later. Next, matlabpool is used in the option to off-load and run in parallel. Finally, in order to retrieve worker's workspace, the load is used.

A. Batch processing

Batch processing is the execution of an arrangement of employments in a program on a PC without a manual intercession, namely non-intelligent. The preparing mode, empowers it to execute a progression of projects each on one set or cluster of sources of info, as opposed to single information. The parallel applications will run interactively or in batch. These jobs can be formed into batches and the number of jobs in a batch is limited by the capacity of the processing machines to accommodate the jobs [1]. So, more jobs can be done with the batch processing process [2]. Moreover, the measure of a batch is constrained by the quantity of job accessible and preparing time of a batch is autonomous of the size of the batch [2]. A batch processing technique is needed to be implemented if repetition is used in series of object [3].

II. CHARACTERISTIC OF BATCH PROCESSING

Firstly, batch processing machines are high in both costs and setup costs, so they are exceptionally restricted in the amount [4]. Secondly, the handling time in batch processing is much longer than different sorts of operation. Finally, batch processing operation from time to time occurs at the end of the processing route hence it affects the delivery dates. Moreover, batch processing can be utilized to process self-assertive inquiries over various arrangements of information. It normally processes results that are derived from every one of the information it includes, and empower deep analysis of huge data information. Amazon EMR, Spark, Hadoop are examples of platforms that bolster batch [5]. Spark is an in-memory batch engine that executes streaming jobs as the arrangement of mini-batches [6]. Moreover, batch processing can reduce boredom or human error because it saves a lot of time.

Table 1. It shows characteristics of batch processing [1]

Characteristics	Batch Processing
Data scope	Queries or processing over all or most of the data in the dataset.
Data size	Large batches of data.
Performance	Latencies in minutes to hours.
Analyses	Complex analytics.

B. How it works

Batch processing will process numerous of jobs at the same time. There are many assumptions made on how batch processing work [7], which are

- All jobs in a batch will begin and end preparation in the meantime
- All jobs are free at time 0
- The machines do not fail
- Once a machine starts handling a batch of jobs, it is difficult to add new jobs to the batch and existing jobs cannot be erased from the batch
- While a machine finishes processing a batch of jobs, the following batch can be directly loaded with no setup time or delay due to operator unavailability

III. LIFE CYCLE OF A JOB

A job undergoes through a number of stages to be created and functional [1]. There are six stages as shown in the following:

1. Pending: A job is made on scheduler with *createJob* work in customer session of Parallel Computing Toolbox software. The principal job in pending state which a user needs to characterize the job by adding the tasks to it.
2. Queued: While the function of the job is submitted, scheduler puts the job on the line and the job is in queue state. The scheduler executes the job in queues in the succession in which they are presented, all jobs moving up the lines as the jobs before they are done.
3. Running: While the jobs achieve the highest point of the line, the scheduler gives the job's task to worker sessions for assessment. The job's state is currently running. The scheduler starts to execute the following job if a large number of workers are accessible than are required for a job's task.
4. Finished: The job is moved to the completed state when the majority of a job's task has been accessed. The outcomes are getting from all the tasks in the job by utilizing function *fetchOutputs*.
5. Failed: A job may fail if the scheduler confronts an error when attempting to execute its orders or get necessary files when utilizing an outsider scheduler.
6. Deleted: The state job is erased when a job's information is deleted from its information location. This state is accessible as long as the job object stays in the client.

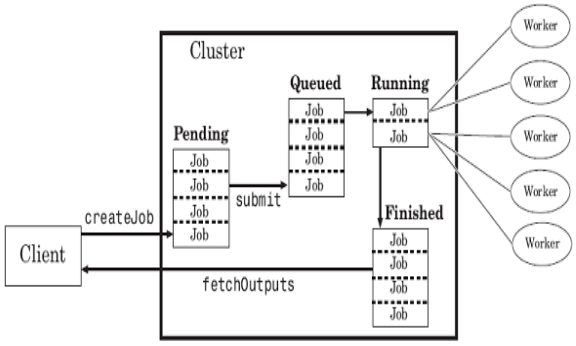

Fig. 1. It displays life cycle of a job [1]

C. Application Using Batch Processing

Batch processing is always related to the application in real life. Batch processing machines (BPMs) are generally utilized as a part of industrial systems which it is processing different jobs in a batch [8].

IV. BATCH PROCESSING TIME

A deviation of the flow shop sequencing problem is proposed, where jobs are formed into batches and each batch is then processed as a unit through the flow shop [9]. This issue, generally, is a blend of the bin-packing problem, where jobs are batched within a given limit and the flow-shop issue where the processing sequence of the batches is resolved. However, the issue of inquiry heuristics is comprehended by inferring a lower bound.

Next, a review of research which is real-time control strategies of batch processing machines in semiconductor manufacturing systems was put forwarded [10]. The problem of the lower bound is appropriate to a partial solution of the problem - some jobs have been produced in batches but other jobs are yet to be batched. In general, the batch processing takes longer time compared to the discrete processing time [10]. The real-time control strategies of BPMs is categorized into two policies which are threshold policy and look-ahead policy, as indicated by the usage of knowledge on future arrivals of products. The threshold policy is applied when there is no information accessible while look-ahead policies utilize information on the coming future system status [10].

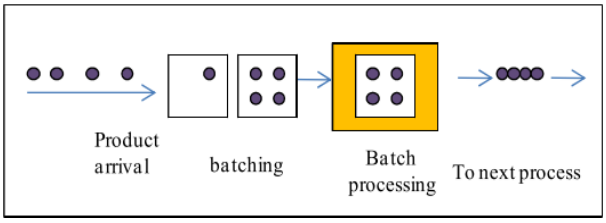

Fig. 2. It shows batch processing time [10]

Some authors have specified lower bounds for the permutation flow-shop problem [9]. The lower bounds are created to identify the release dates or "heads" while delivery

times or "tails" of the batches [9]. Moreover, a formulation of the two-machine job shop scheduling problem is proposed by [11]. It is said that jobs do not visit the same machine. It is hard to characterize the optimum solutions when the Common Due Date (CDD) is restricted. Finally, a DP algorithm is used to find the optimal solutions for the two-machine which is E/T JSSP when the CDD is semi-restricted. Three cases of the due date are considered in order to solve the two-machine E/T JSPP over CDD. The JSSPET algorithm decides the type of CDD, namely restricted, semi-restricted, or restricted, and then applies the appropriate procedure [11].

Algorithm JSSPET

Calculate F1, F2, Δ1, Δ2
Apply SCHED procedure to M1
Apply SCHED procedure to M2
Calculate T1 =Number of tardy jobs in M1^C
Calculate T2 =Number of tardy jobs in M2^C
If CDD ≥ max{F1 +Δ1,F2 +Δ2} and T1 =0 and T2 = 0 then
Optimal schedule is given by SCHED procedure
on both machines
Schedule first operations on machine 1 and
machine 2 by using EDD rule.
Stop.

Else
Apply SCHED to machine where min {F1 +
Δ1,F2 +Δ2} holds
Apply EVS to the other machine
Apply TVS to the other machine
Apply Nosplit to the other machine
Solution for the other machine is
min {EVS, TVS, Nosplit}
Calculate T1 =Number of tardy jobs on M1^C
Calculate T2 =Number of tardy jobs on M2^C
If CDD < max{F1 +Δ1, F2 +Δ2} and CDD ≥
min {F1 +Δ1, F2 +Δ2} and T1 =0 and T2 =0
then
Optimal schedule is given by SCHED and
min {EVS, TVS, Nosplit}
Schedule first operations on machine 1 and
machine 2 by using EDD rule.
Stop.
Else
Apply Restricted Heuristic
End If
End If
End Algorithm

V. BATCH-PROCESSING SERVICES

Dai and Li propose the stabilizing batch-processing networks [12]. A general plan for sending out policy is converted to batch policy.

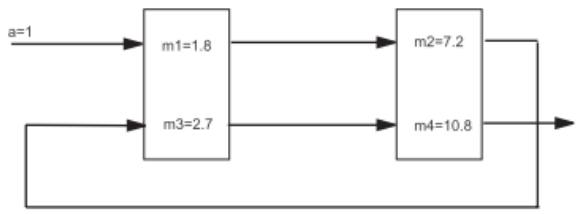

Fig. 3 It demonstrates batch networks [12]

The following shows that throughput relies upon the batch policy employed. The open multiclass batch processing networks, called batch-processing systems is first introduced [12]. The relating standard-processing networks that are undefined to batch processing networks avoid for a job that is processed each one in turn is proposed. Next, the LBFS batch-policy in the batch-processing network is utilized. Under the LBFS approach, every server frequently stacks the most elevated nonempty class to make a batch, despite the fact that the chosen class may have just one work in it. Finally, a general mechanism of building an induced batch policy is shown for the batch network from a transmit approach for the standard system.

Next, a new heat integrated distillation column (HIDiC) is introduced for batch processing [13]. A conventional batch distillation (CBD) columns work with two ordered stages, to be specific start-up and production phase. In this CBD, heat is presented at the most astounding temperature point, that is still pot and the same goes discarded at the least temperature, that is condenser. Another energy productive plan is acquainted with reaching their highest aim which is to enhance the heat reversibility [13].

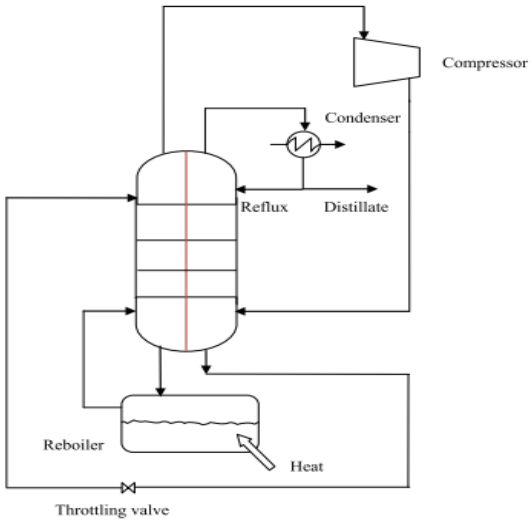

Fig. 4. It shows schematic representation of the proposed divided-wall HIDiC column for batch processing [13]

VI. CONCLUSION

With batch processing, the researcher has shown the properties of batch processing, how the batch processing operating and the application that uses batch processing. Initially, the batch method operates under Parallel Process Toolbox (PCT) in MATLAB which is used for mathematical programming. In the parallel-batching scheduling issue, some jobs can be processed as a batch that can be gathered up to the limit of the machine, at the same time on a machine at one time. The machine cannot be interrupted once the process begins. The scheduler in PCT will assign the job to a worker in batch processing. Moreover, batch processing can be used for process self-assured request over different arrangements of data. It regularly forms output results that are provided from each one of the information it incorporates and enables profound investigation of gigantic information data.

REFERENCES

[1] https://www.mathworks.co.uk/searchresults/?search_submit=matlabcentral&query=matlab+parallel+computation+example&q=matlab+parallel+computation+example&c[]=matlabcentral

[2] Roth, P.J. and J.D. Daniels, *Doing More with Less- Exploring Batch Processing and Outsourcing in Academic.* Management and Administration: p. 361-365.

[3] Li, L., et al., *A MATLAB-based image processing algorithm for analyzing cupping profiles of two-layer laminated wood products.* Measurement, 2014. 53: p. 234-239.

[4] Li, X., et al., *Scheduling unrelated parallel batch processing machines with non-identical job sizes* Computer & Operations Research, 2013. 40: p. 2983-2990.

[5] Bhullar, D. and N. Verma, *Review Paper on Batch Processing and Streaming Processing.* International Journal Of Technology And Computing(IJTC), 2017. 3: p. 44-48.

[6] Bingmann, T., et al., *Thrill : High-Performance Algorithmic Distributed Batch Data Processing with C++*. 2012.

[7] Suhaimi, N., C. Nguyen, and P. Damodaran, *Lagrangian approach to minmize makespan of non-identical parallel batch processing machines.* Computers & Industrial Engineering 2016. 101: p. 295-302.

[8] Zhou, S., et al., *An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes.* Int. J. Production Economics, 2016. 179: p. 1-11.

[9] Basnet, C., *Heuristics for batching and sequencing in batch processing machines.* Croatian Operational Research Review, 2016. 7: p. 291-302.

[10] Koo, P.-H. and D.H. Moon, *A Review On Control Strategies of Batch Processing Machines in Semiconductor Manufacturing.* International Federation of Automatic Control, 2013: p. 1690-1695.

[11] Salem, M.A., L. Bedoya-Valencia, and G. Rabadi, *Heuristic and Exact Algorithms for the Two-Machine Just in Time Job Shop Scheduling Problem.* Mathematical Problems in Engineering, 2016: p. 1-11.

[12] Dai, J.G. and C. Li, *Stabilizing Batch-Processing Networks.* Operations Research 2003. 51: p. 123-136.

[13] Jana, A.K., *A new divided-wall heat integrated distillation column (HIDiC) for batch processing : Feasibility and analysis.* Applied Energy, 2016. 172: p. 199-206.

[14] M, M. and A.I. Sivakumar, *A literature review, classification and simple meta-analysis on schedulling of batch processors in semiconductor.* International Journal Of Advanced Manufacturing Technology, 2006. 29: p. 990-1001.

[15] https://www.youtube.com/watch?v=EgTWBHcxxZs&t=5s