Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 3439-3445

Research Article

Heuristics based Dynamic Multi Task scheduling framework using Differential Equation for distributed computing

Dr.M.Rajarajeswari

Abstract

Task scheduling on the heterogeneous distributed computing architectures have driven a new initiative on heuristics based paradigm to handle dynamic distributed workloads. Due to the expanding volume and the variety of Multi tasking work flows, task scheduling is often processed on scheduler. However existing scheduler has been modelled using map reduce architecture to task execution with map task and reduce task components. Despite of many advantageous, those model results in performance degradation such as make span and tardiness. In order to eliminate those issues, a new Heuristics based Dynamic Multi Task scheduling framework using Differential Equation for distributed computing has been proposed in this paper. The optimal scheduling solution has been generated using heuristics through differential equation that can quickly identify the efficient resource for execution. Heuristics of task scheduling include the ordering of system resources and predicting the worst case behaviour of the system on the map task and reduce task is based on the adaptive schemes. Further heuristics computes the priority of the task assigned to the resources on the task locality allocation strategies to particular resources. Finally trajectory of the task has been computed on aspect of best configuration on different communication modes towards its task propagation and efficient execution resource. Simulation results of the proposed architecture has been proven to be highly scalable on the proposed distributed computing architecture against traditional state of art approaches on the Google workloads and improves the make span and reduces the tardiness on the task execution.

Keywords: Task Scheduling, Heuristics, Differential Equation, Distributed Computing, Trajectory strategies

1. Introduction

Tasks in terms of workloads are evolving and increasing in volume has gained considerable attention in recent year on heterogeneous resource execution environment. Especially task has been represented in form of raw data, researches on pre-processing of the raw data, such as size changing, format conversion, and noise handling on execution system has been modelled to alleviate this problem[1]. Pre-processed data contains one or more jobs and each job consists of multiple stages. There are two types of stages from high level namely map

[1]Associate Professor, Department of Mathematics Hindustan College of Arts and Science

stage and reduce stage, which in turn are carried out by several map tasks and reduce tasks respectively[2].

Task is usually loaded into distributed file systems[3][4] and organized as data blocks. In the map stage, the map task (mapper) reads the data block to process and outputs intermediate data. In the reduce stage, the reduce task (reducer) fetches its own share of intermediate data from all the map tasks in the shuffle process[5]. This is an all-to-all communication. The resulting network traffic in the map stage and the reduce stage has become the significant performance bottleneck, which can saturate network bandwidth and prolong task execution time.

The configurationally heterogeneity makes it non-trivial to minimize the total tardiness in the presence of the processing constraint[6]. In such heterogeneous systems[7], the processing time and routing delay of a flow depends on the routing strategies. In this work, a Heuristics based Dynamic Multi Task scheduling framework using Differential Equation for distributed computing has been proposed in this paper. The optimal scheduling solution has been generated using heuristics through differential equation that can quickly identify the efficient resource for execution. Heuristics of task scheduling include the ordering of system resources and predicting the worst case behaviour of the system on the map task and reduce task is based on the adaptive schemes.

Further heuristics computes the priority of the task assigned to the resources on the task locality allocation strategies to particular resources. Finally trajectory of the task has been computed on aspect of best configuration on different communication modes towards its task propagation and efficient execution resource. The rest of paper is organized as follows. Section 2 provides the related works on workload scheduling on cloud computing paradigm. Section 3 defines and designs the proposed heuristics based dynamic multitask scheduling using different equation in Big Data cloud on evolving data volumes. In Section 4, experimental results and performance of the proposed model has been detailed against state of art approaches on large volume of data. Finally Section 5 concludes the paper

2. Related work

In this section, various existing model on multitask scheduling of the raw data for execution on evolving volume and dynamic update of the data has analysed on basis of scheduling solution and adaptive mechanism to generate the optimal solution has been detailed as follows

2.1. Locality-Aware Task Scheduling Algorithm

In this mechanism, a unified graph model for the map task scheduling and the reduce task scheduling has been employed to generate the optimal scheduling solution[8]. A locality-aware executor allocation strategy has been constructed with the data locality-awareness and reduces the distance of data transmission when the reduce tasks. In a computing system where resources release quickly, the delay scheduling can achieve high localization rate for tasks. Further an optimal reducer placement to minimize data transfer in Map Reduce-style processing.

2.2. Iterative-Improvement-Based Heuristics for Adaptive Scheduling of Tasks

In this mechanism, scheduling heuristics employed based on a common greedy criterion which depends on the momentary completion time values of the tasks. This greedy decision criterion exploiting the file-sharing interaction among tasks since completion time values are inadequate to extract the global view of this interaction. It is a three-phase scheduling approach which involves initial task assignment, refinement, and execution ordering phases. For the refinement phase, hypergraph model using elegant hypergraph-partitioning on iterative-improvement-based heuristics for refining the task assignments according to two novel objective functions has been analysed[9].

3. Proposed model

In this section, a heuristics based dynamic multitask scheduling architecture using differential equation for distributed computing has been proposed with design steps on terms of make span and tardiness of the system

3.1. Map- Reduce Architecture

In distributed computing environments, dynamic input task from various sources is usually loaded into a file or files in a distributed file system of the resources where each file is partitioned into many data blocks[10]. A data block may have multiple replicas among the nodes of cluster. The job is divided into stages and it will be assigned to different resources for it execution in order. In order to accomplish the task execution in the resources, the map stage (the initial stage), the map tasks read data blocks as input splits, and then process and output intermediate results on local nodes for task propagation[11].

The intermediate results produced in map task will collected in the reduce task towards task execution. Reduce task process according the scheduling model. In this work, differential equation has been employed to carry out the identification of the optimal resource for task execution has been discussed in the below sections.

3.2.Optimal Scheduling Solution using differential Equation

Differential Equation[12] has been employed for its efficient search capability in predicting the resource for task execution on modelling various heuristics. Heuristics generated using differential equation includes four steps which is Initialisation, Mutation, Recombination and Selection. In first stage of heuristics determination, population vector will be formed through collection of the data from the reduce task. Collected population vector will be applied to mutation to interchange the sequence of the task execution in random way. Muted vector will undergo recombination stage to generate the trial vector on the value of cover over rate, resultant vector containing the task will be produced on basis of the configurations.

Heuristic objective Function

Heuristics composed of different resource cluster will be generated for task execution on the resultant recombination trial vector using assumption of intra vector variation. The objective function for intra vector variation is as follows

$$U \equiv \sum_{i=1}^{N} \frac{C_i}{T_i} \leq N(2^{1/N} - 1)$$

Where N denotes iteration number for the generation of cluster C

The heuristic generation is dynamic and requires a more complex run-time system which will have higher overhead due to change of volume and velocity of the task. Overhead of the task execution will be eliminated on modelling adaptive scheme through determination of worst case behaviour.

3.3.Estimating Worst Case Behaviour on adaptive schemes

Worst case behaviour of task on the resource towards execution is computed using adaptive schemes due to variety of sizes and volume changes in task. Adaptive scheme[13] computes feasible schedule to dynamic changes of the map task composed of evolving task from distributed systems. Adaptive scheme generate adaptive cluster containing task for different heuristics generated using following equation

$$w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil C_j$$

The set of values of the w_i^{n+1} will be provide adaptive solutions for the particular heuristics. The computed heuristics from adaptive schemes provides best configuration to execute the workload variation on various range and determines the worst configuration on run time on basis of latency and minimal resource capabilities.

3.4.Computing Task Trajectory

In the cluster network, the communication delay depends not only on the size of data transfer, but also on the transmission distance: the larger the amount of data transfer and the farther the transmission distance, the greater the communication delay[14]. In order to reduce those complexities, task trajectory is computed on basis of communication nodes. Trajectory of the task depends completely on communication cost. Partition skew has been used to determine the trajectory of the task on reconfigured vector containing task. Partition skew determines the execution efficiency of task on following equation

$$P_i = \max_{k=1}^{k} usage(k,i)C(k)$$

C(k) is cluster size and usage(k,i) determines the execution speed of the resource. The partition skew computation provides better performance on the collocated task executions.

3.5.Locality Allocation Strategies for Task Prioritization

On priority-based scheduling, a high-priority process may be released during the execution of a lower priority one. In these locality allocation strategies has been set to the efficient resource executor on basis of locality based strategies. In a preemptive scheme, there will be an immediate switch to the higher-priority process[15]. With non-preemption, the lower-priority process will be allowed to complete before the other executes. In addition, dynamic priority that is the maximum of its own static priority and any it inherits due to it blocking higher-priority processes for locality task initial as priority.

$$w_i^{n+1}(q) = B_i + (q+1)C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n(q) + J_j}{T_j} \right\rceil C_j$$

Preemptive schemes enable higher-priority processes to be more reactive, and hence they are preferred. Alternative strategies allow a lower priority process to continue to execute for a bounded time. These schemes are known as deferred preemption or cooperative dispatching.

Algorithm 1: Heuristics based Resource Scheduling

Input: Task T1, T2, T3...Tn

Output: Resource R1, R2, R3..Rn

Process

Map Task ()

Reduce Task ()

Heuristics Computation ()

Population Vector V = [V_1[x],V_2[x],V_3[x]]

Mutation M = Flip instance of V

Reconfiguration M

Task Priority based on locality

$$P_i > P_j \wedge D_i > D_j$$

Where D_i is the distance between task from Resource

Task Trajectory()

$$\text{Task Weight (p,c)} = \sum_{k=0}^{n} \binom{n}{k} D^k P^{n-k}$$

$$\text{Communication Cost (p,c)} = \sum_{k=0}^{n} \binom{n}{k} D^k P^{n-k} B_i$$

The scheduler allocates sufficient resources to task so that its perceived performance is satisfactory on basis of the locality and trajectory. Measuring schedule behaviour to task with worst-case figures may lead to very low processor utilizations being observed in the actual running system.

4. Experimental Results

In this section, we evaluate our proposed algorithm and strategies of heuristics based scheduler on the adaptive scheme using differential equation has been formulated and configured as simulation environment in map reduce architecture. Hence, the task scheduler towards execution using map and reduce function has been carried out initially. Map reduce

function provide initial task placement to random way. Further differential equation has been employed to provide mixed workloads using four stage processes.

Differential equation generates the optimal task to resources. It waits a certain time before launching non-local tasks, thereby achieving better locality level from other nodes using intra variation of the task. Figure 1 represents the performance of the proposed model against state of art approach.

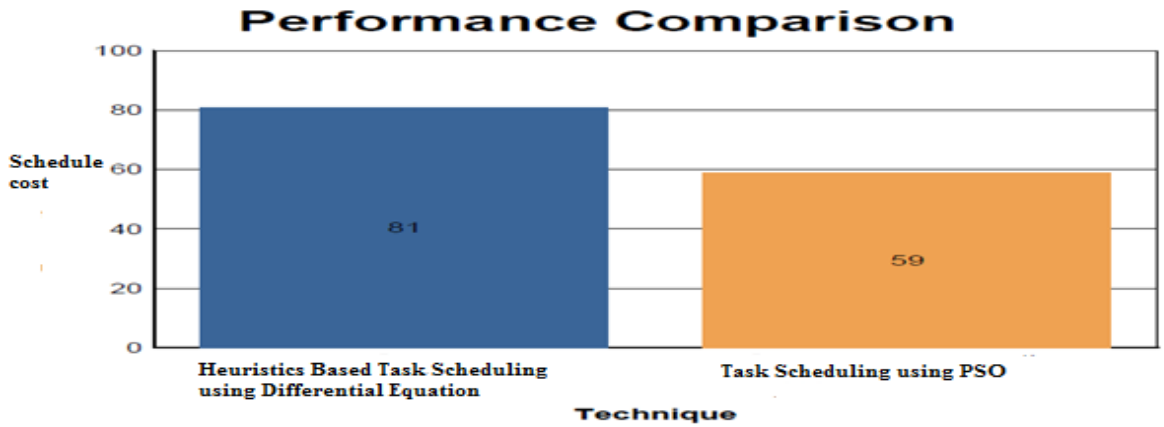

Figure 1: Performance evaluation of the task scheduling techniques

Dynamic task with complex run time system produces better results basis of heuristics against change of volume and velocity. Table 1 provides the performance comparison of the task scheduling techniques

Technique	**Scheduling Cost in percentage**	**Overhead in ms**
Heuristics based task scheduling using Differential Equation – Proposed	81	4
Task Scheduling using Particle Swarm optimization -Existing	50	33.23

Adaptive scheme based on heuristics computes feasible schedule to dynamic changes of the map task for the better schedule cost on evolving task in the distributed systems. Proposed technique is superior to other algorithms in terms of job execution time in each test for the various type of job task to execution.

Conclusion

We design and simulate the heuristics based dynamic task scheduling architecture using differential equation for better communication cost and to reduce the overhead of the task scheduling. In this work, optimal solution to task scheduling has achieved using mutation and reproduction process on DE, obtained solutions has been considered against worst case behaviour of the resource. Finally task priority has been set on the locality of the task on the best configured resource towards execution through effective task trajectory using weight function. A

simulation result demonstrates the proposed model is effective against various state of art approaches on various test scenario of the task size and resource types.

References

[1] J. Chen et al., "A parallel random forest algorithm for big data in a spark cloud computing environment," IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 4, pp. 919–933, Apr. 2017.
[2] Z. Fu and Z. Tang, "Optimizing speculative execution in spark heterogeneous environments," IEEE Trans. Cloud Comput., to be published, doi: 10.1109/TCC.2019.2947674.
[3] R. Xue, S. Gao, L. Ao, and Z. Guan, "BOLAS: Bipartite-graph oriented locality-aware scheduling for MapReduce tasks," in Proc. Int. Symp. Parallel Distrib. Comput., 2015, pp. 37–45.
[4] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in Proc. Eur. Conf. Comput. Syst., 2010, pp. 265–278.
[5] N. S. Naik, A. Negi, T. B. B. R., and R. Anitha, "A data locality based scheduler to enhance MapReduce performance in heterogeneous environments," Future Gener. Comput. Syst., vol. 90, pp. 423–434, 2019.
[6] M. Hammoud and M. F. Sakr, "Locality-aware reduce task scheduling for MapReduce," in Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci., 2012, pp. 570–576.
[7] J. Xie, Y. Shu, X. Ruan, Z. Ding, and Q. Xiao, "Improving MapReduce performance through data placement in heterogeneous hadoop clusters," in Proc. IEEE Int. Symp. Parallel Distrib. Process.,
2010, pp. 1–9.
[8] S. Suresh and N. P. Gopalan, "Delay scheduling based replication scheme for hadoop distributed file system," Int. J. Inform. Technol. Comput. Sci., vol. 7, no. 4, pp. 73–78, 2015.
[9] Z. Tang, X. Zhang, K. Li, and K. Li, "An intermediate data placement algorithm for load balancing in spark computing environment," Future Gener. Comput. Syst., vol. 78, pp. 287–301, 2018.
[10] M. Alicherry and T. V. Lakshman, "Optimizing data access latencies in cloud systems by intelligent virtual machine placement," in Proc. IEEE INFOCOM, 2013, pp. 647–655.
[11] X. Meng and L. Golab, "Optimal reducer placement to minimize data transfer in MapReduce-style processing," in Proc. IEEE Int. Conf. Big Data, 2017, pp. 339–346.
[12] D. Cheng, X. Zhou, P. Lama, J. Wu, and C. Jiang, "Cross-platform resource scheduling for spark and MapReduce on YARN," IEEE Trans. Comput., vol. 66, no. 8, pp. 1341–1353, Aug. 2017.
[13] Z. Tang, W. Lv, K. Li, and K. Li, "An intermediate data partition algorithm for skew mitigation in spark computing environment," IEEE Trans. Cloud Comput., to be published, doi: 10.1109/TCC.2018.2878838.
[14] Z. Ren, X. Xu, W. Jian, W. Shi, and Z. Min, "Workload characterization on a production Hadoop cluster: A case study on taobao," in Proc. IEEE Int. Symp. Workload Characterization, 2013, pp. 3–13.
[15] J. Tan, S. Meng, X. Meng, and L. Zhang, "Improving ReduceTask data locality for sequential MapReduce jobs," in Proc. IEEE INFOCOM, 2013, pp. 1627–1635.