ISSN 2348 – 9928
IJAICT Volume 3, Issue 11, March 2017 Doi:01.0401/ijaict.2017.03.12 Published on 05 (04) 2017

AN EFFICIENT FAULT TOLERANT CLUSTERING FOR SCIENTIFIC WORKFLOW

Mr. A. Bharanidharan **Ms. RAJ.Jahashri** **Mr. K.Srinivasan** **Mr. Tarun V Radhakrishnani**

Department of Computer Science and Engineering,
Sri Ramakrishna Engineering College,
Coimbatore, Tamilnadu, India

***Abstract** - An effective method for the reduction of execution overhead and for improving the computational granularity of scientific workflow tasks that are executing on distributed resources is Task clustering. A job is composed of many tasks and may have a higher risk of suffering from failures than in executing a single task job. In this paper, we direct a hypothetical investigation of the effect of transient failures on the runtime execution of logical work process executions .This system proposes a maximum likelihood estimation-based parameter algorithm which is used for a general task failure modeling framework to model the workflow performance. In this paper, the system proposed here is Dynamic Balanced clustering method which combines the methods of vertical clustering, horizontal clustering and dynamic clustering to reduce the execution overhead for the scientific workflow task execution*.

***Keywords** - Scientific Workflows, Task Clustering, Fault Tolerance, Parameter Estimation.*

I. INTRODUCTION

Scientific workflow processes can be made out of some fine computational granularity assignments, where the task runtime might be shorter than the framework overhead—the time of time amid which random work other than the clients calculation is performed. Assigning the grouping techniques blend a few short assignments into a solitary occupation with the end goal that the employment runtime is expanded and the general framework overhead is diminished. Task grouping is the most widely recognized system used to address execution overheads and increment the computational granularity of work process undertakings executed on disseminated assets be that as it may, existing bunching methodologies disregard or think little of the effect of disappointments on the framework, regardless of their noteworthy impact on large scale disseminated frameworks for example, Grids and Clouds . In this work, we concentrate especially on transient disappointments since they are anticipated that would be more predominant than lasting disappointments. A grouped occupation comprises of different undertakings. On the off chance that an undertaking inside a grouped occupation comes up short (i.e., is ended by unforeseen occasions amid its calculation), the occupation is set apart as bombed, even despite the fact that errands inside a similar employment have effectively finished their execution.

A few strategies have been created to adapt to the negative effect of occupation disappointments on the execution of scientific work processes. The most widel7y recognized method is to retry the failed work. Be that as it may, retrying a clustered job can be costly since finished undertakings inside the occupation more often than not should be recomputed, in this way asset cycles are squandered. Also, there is no assurance that recomputed undertakings will succeed. As an option, jobs can be replicated to keep away from the failures of a particular to a work node. Be that as it may, work replication may likewise squander assets, specifically for long-running occupations. To lessen asset squander, work executions can be occasionally check pointed to restrict the measure of retried work. Be that as it may, the overhead of performing check pointing can constrain its advantages [1][2].

II. RELATED WORK

Failure examination and demonstrating of computer frameworks have been broadly concentrated in the course of recent decades. These reviews incorporate, for example, the grouping of normal framework disappointment qualities and disseminations, underlying driver examination of failure , experimental and factual investigation of system framework mistakes and disappointments , and the advancement and investigation of procedures to anticipate and relieve benefit disappointments . In logical work process administration frameworks (WMS), blame resistance issues have likewise been tended to. For example, the Pegasus WMS has consolidated an undertaking level checking framework, which retries an occupation if an assignment disappointment is recognized. Provenance information is likewise followed and used to dissect the reason for failure . A review of blame discovery, avoidance, and recuperation strategies in current

ISSN 2348 – 9928
IJAICT Volume 3, Issue 11, March 2017 Doi:01.0401/ijaict.2017.03.12 Published on 05 (04) 2017

Matrix WMS is accessible in. The review gives an arrangement of recuperation methods, for example, assignment replication, check pointing, resubmission, also, movement. In this work, we join some of these procedures with undertaking grouping strategies to enhance the execution and unwavering quality of fine-grained assignments [3]. To the be best of our insight, none of the current WMS have given such components.

The low execution of fine-grained undertakings is a typical issue in generally appropriated stages where the planning overhead and lining times at assets are high. A few papers have tended to the control of undertaking granularity of inexactly coupled assignments. For example, Muthuvelu et al. proposed a bunching calculation that gatherings sack of undertakings based on the runtime, and later in view of assignment record measure, CPU time, furthermore, asset requirements. As of late, they proposed an web based planning calculation that consolidations undertakings in view of asset arrange use, client's financial plan, and application due date. Likewise, Ng et al. nd Ang et al. Moreover considered system transfer speed to enhance the execution of the assignment planning calculation. Longer task are allotted to assets with better system data transfer capacity. Liu and Liao proposed a versatile planning calculation to amass fine-grained assignments as per the handling limit and the system data transfer capacity of the as of now accessible assets [4].

A few papers have tended to the work process mapping issue by utilizing coordinated non-cyclic chart (DAG) planning heuristics. Specifically, HTCondor utilizes matchmaking to abstain from planning assignments to register hubs without adequate assets (CPU control, and so on). Beforehand, we received a comparable way to deal with abstain from scheduling work process task to figure nodes with high failure rates. In this work, we concentrate on the execution pick up of task clustering, specifically on the best way to conform the clustering size to adjust the cost of task retry and of the scheduling overheads. Machine learning techniques have been used to predict execution time and system overheads, and to develop probability distributions for transient failure characteristics. Duane et.al. [5][6] used Bayesian network to model and predict workflow task runtimes. The important attributes (e.g. external load, arguments, etc.) are dynamically selected by the Bayesian network and fed into a radial basis function neural network to perform further predictions. Ferreira da Silva et al. used regression trees to dynamically estimate task needs including process I/O, runtime, memory peak, and disk usage. In this work, we use the knowledge obtained in prior works on failure[7][8], overhead, and task runtime analyses as the foundations to build the prior knowledge based on the maximum likelihood estimation that integrates both the knowledge and runtime feedbacks to adjust the parameter estimation accordingly.

III. PROPOSED SYSTEM

Main aim of this proposed system is to reduce the execution overhead and to improve the computational granularity of scientific work flow task that are executing on distributed resources.

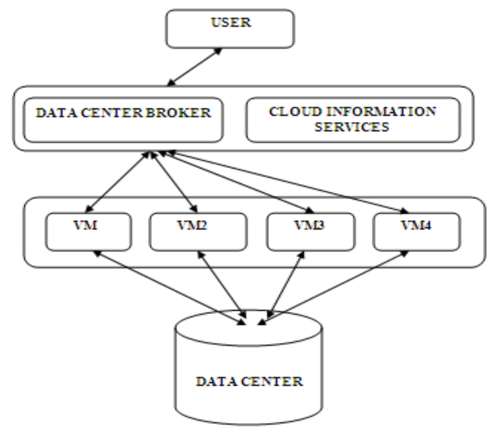

Fig.1. System Overview Diagram

Fig.1 Shows the System Overview Diagram that describes how the user sends a job to the data center broker and how it is getting partitioned into cluster of jobs that are executed in different vm and sent to the data center.

2.1 Workflow System Model

A work process is displayed as a coordinated non-cyclic diagram (DAG), where every hub in the DAG regularly speaks to a work process errand, and the edges speak to conditions between the undertakings that compel the request in which assignments are executed. Conditions regularly speak to information stream conditions in the application, where the yield documents created by one errand are utilized as contributions of another assignment. Each errand is a computational program and an arrangement of parameters that should be executed.

A work process is demonstrated as a Directed Acyclic Graph (DAG). Every hub in the DAG regularly speaks to a work process undertaking (t), and the edges speak to conditions between the errands that compel the request in which the assignments are executed. Conditions ordinarily speak to information stream conditions in the application, where the yield documents created by one errand are utilized as contributions of another assignment. Each undertaking is a program and an arrangement of parameters that should be executed. The submit have readies a work process for execution (bunching, mapping, and so forth.), and laborer hubs, at an execution site, execute employments separately.

2.2 Task Failure Model

The strategies utilizing an task transient disappointment demonstrate in light of a parameter learning process that gauges the circulation of the assignment runtimes, the framework overheads, and the between entry time of disappointments. The procedure utilizes the Maximum Likelihood Estimation (MLE) in view of earlier and back learning to construct the appraisals. The earlier information about the parameters is displayed as an appropriation with known parameters. The back information about the undertaking execution is additionally demonstrated as an appropriation with a known shape parameter and an obscure scale parameter. The shape parameter influences the state of dispersion, while the scale parameter influences the extending or contracting of a conveyance. Likelihood conveyances, for example, Weibull and Gamma are generally portrayed with two parameters: the shape parameter (f), and the scale parameter (q). The shape parameter influences the state of a circulation, for instance, regardless of whether it is symmetrical or not. The scale parameter influences the extending or contracting of a circulation, for instance, regardless of whether it is roughly uniform or it has a pinnacle. Both parameters control the attributes of a circulation.

Let a, b be the parameters of the earlier information, D the watched dataset, and q the parameter we expect to assess. In Bayesian likelihood hypothesis, if the back circulation p(qjD;a;b) is in an indistinguishable family from the earlier conveyance p(qja;b), the earlier and the back appropriations are then called conjugate dispersions, and the earlier is known as a conjugate earlier for the probability work. For example, the Inverse-Gamma family is conjugate to itself (or self-conjugate) concerning a Weibull probability work: if the probability capacity is Weibull, picking an Inverse-Gamma earlier over the mean will guarantee that the back dissemination is additionally Inverse-Gamma. In view of this definition, the parameters estimation of our errand disappointment show has its establishments on earlier deals with failure and execution investigates Therefore, once watched information D, the posterior distribution is

$$\mathbf{P(\theta|D,a,b)} = \frac{\mathbf{P(\theta|a,b) \times P(P|\theta)}}{\mathbf{P(D|a,b)}} \propto \mathbf{P(\theta|a,b)} \ \times \mathbf{P(P|\theta)}$$

where D is the observed inter-arrival time of failures X, the observed task runtime RT, or the observed system overheads S; p(qjD;a;b) is the posterior we aim to compute; p(qja;b) is the prior, which we have already known from previous works; and p(Djq) is the likelihood.

The runtime of a employment may be An arbitrary variable shown by d. An grouped particular occupation succeeds just if at from claiming its assignments succeed. The employment runtime will be the entirety of the combined errand runtime about k assignments and the framework overhead. We expect that those undertaking runtime for each errand will be autonomous from claiming every other, subsequently those combined assignment runtime from claiming k assignments is Additionally a gamma appropriation since the aggregate from claiming gamma circulations with the same scale parameter will be at present a gamma circulation. We likewise expect the framework overhead may be free of every last one of undertaking runtimes.

2.3 Fault-Tolerant Clustering

A o-DAG model, the system could unequivocally express those transform for assignment grouping. In this work, we address undertaking grouping horizontally further more vertically. Horizontal grouping (HC) merges various errands inside the same level level of the workflow the level level of a errand may be characterized as those longest separation from those DAG's passage undertaking will this errand. Vertical grouping (VC) merges assignments inside a pipeline of the workflow. Task in the same pipeline impart An single-parent-single-child relationship, i.e a task tb a unique parent ta, which has a unique child tb.

On circumstances the place the planning Also queue overheads are important, the utilization of assignment clustering systems could essentially move forward those workflow execution. For a Perfect scenario, the place disappointments are absent, those number about assignments done An grouped vocation (clustering size, k) might make characterized Likewise the amount about every one errands in the queue partitioned Toward those amount of accessible assets. Such a credulous setting assures that those number of occupations is equivalent to those number of assets and the workflow might fully use those assets. However, On An broken earth the clustering size ought to be characterized as stated by the disappointment rates, over particular, those undertaking disappointment rate. Intuitively, though those assignment failure rate will be high, those bunched employments might need will a chance to be re-executed additional regularly contrasted of the the event without grouping. Such execution corruption will neutralize those profits of lessening planning overheads. We will indicate how will alter k dependent upon those evaluated parameters of the task runtime t, the system overhead s, and the inter arrival time of task failures.

Unseemly task grouping might negatively sway those workflow make span to faulty distributed environments. The recommend three fault-tolerant task grouping methods—Horizontal Clustering (HC), Dynamic Clustering (DC), and Vertical Clustering (VC) that alter the grouping measure (k) of the occupations to decrease the sway about task failure on the workflow execution. These routines are In view of the Horizontal Clustering (HC) technique that need been executed what's more utilized in the Pegasus workflow administration framework (WMS). Horizontal Clustering (HC). Level grouping merges various errands inside the same level level of the workflow. The grouping granularity (number for errands inside An cluster) of a grouped occupation will be controlled by the user, who characterizes whichever the number from claiming assignments for every bunched occupation (clusters. Size), or the number of bunched employments for every level of the workflow (clusters. Num). To simplicity, we set groups. Num to be those same concerning illustration the measure of accessible assets. For bring assessed those runtime execution for separate grouping granularities. Those grouping What's more blend methods would conjured in the beginning undertaking grouping process, same time those Clustering system is conjured At a neglected work will be distinguished towards those observing framework.

IV. CONCLUSION

In this paper, the system proposed is work flow failures of both depended and independent job in a distributed environment and assess their influence on task clustering. A Dynamic Balanced Clustering which combines the features of Vertical Clustering, Horizontal Clustering and Dynamic Clustering is been proposed. Results obtained in the experiment showed that the proposed methods significantly improve the workflows make span when compared to an existing task clustering method used in scientific workflow management systems.

In the future, we plan to combine our work with fault-tolerant scheduling in heterogeneous environments, i.e, avoiding the mapping of clustered jobs to failure prone nodes using a scheduling algorithm. We also plan to consider other factors such as the execution site, which may improve the accuracy of the model. Future work will consider heterogeneous network models to explore their impact on our fault-tolerant clustering techniques.

References

[1] Weiwei Chen, Rafael Ferreira da Silva, Ewa Deelman, Thomas Fahringer “Dynamic and Fault-Tolerant Clustering for Scientific Workflows”, in Ieee Transactions On Cloud Computing, Vol. 4, No. 1, January-March 2016

[2] R. Ferreira da Silva, T. Glatard, and F. Desprez, “Controlling fairness and task granularity in distributed, online, non - clairvoyant workflow executions,” Concurrency Comput., Practice Experience, vol. 26, no. 14, pp. 2347–2366, 2014.

[3] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K.Wenger, “Pegasus, a workflow management system for science automation,” Future Gen. Comput. Syst., pp. 17–35, Doi:10.1016/j.future.2014.10.008, 2014.

[4] R. Ferreira da Silva, W. Chen, G. Juve, K. Vahi, and E. Deelman, “Community resources for enabling and evaluating research on scientific workflows,” in Proc. 10th IEEE Int. Conf. e-Sci., pp. 177–184,2014.

[5] F. Jrad, J. Tao, and A. Streit, “A broker-based framework formulti-cloud workflows,” in Proc. Int. Workshop Multi-Cloud Appl.Federated Clouds, pp. 61–68,2013

[6] K. Maheshwari, A. Espinosa, M. Wilde, Z. Zhang, I. Foster, S. Callaghan, and P. Maechling, “Job and data clustering for aggregateuse of multiple production cyberinfrastructures,” in Proc. 5th Int. Workshop Data-Intensive Distrib. Comput., pp. 3–12,2012.

[7] W. Chen and E. Deelman, “Integration of workflow partitioning and resource provisioning,” in Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. , pp. 764–768, May 2012.

[8] N. Yigitbasi, M. Gallet, D. Kondo, A. Iosup, and D. Epema, “Analysis and modeling of time-correlated failures in large-scale distributed systems,” in Proc. 11th Int. Conf. Grid Comput., pp. 65–72,2010.