Dynamic Dataflow Resilience in Heterogeneous Infrastructure

Dr. Priya Nair
Department of Electronics and Communication Engineering, National Institute of Technology Calicut, Kozhikode, Kerala, India.

ABSTRACT
The important and key concept of the clouds is sharing their resources on many different nodes that are making beneficial for any application. Among these according to the user requirements scheduling the resource is very challenging in cloud environment indeed for the low latency over high velocity data streams. As the resources vary in their performance while allocating to the streaming application on infrastructure , which leads to the distraction of QoS of the application meanwhile the resource cost also should be concise to balance the deployment cost. To formalize the optimization with respect to balance the application cost (domain value) QoS, and resource cost by representing the deployment and runtime (dynamic) resource allocation .So we are proposing a new concept alternate path for the dataflow which dynamic in nature to the infrastructure. We propose a Ranking with deadline based algorithm that will implement in the alternative path to provide the end users with more sophisticated control and resource mapping heuristics for communal of dataflow to give near optimal solution. The effect of both variable dynamic data rate and reactive resource performance should be maintained to balance the throughput constraint and application value.

Keywords:-*Cloud, resource allocating, scheduling, dataflow, application, runtime adaptation*

INTRODUCTION

The emerging of new applications in expansion of ubiquitous virtual and physical sensors such as real time high frequency data trading is pushing the limits of processing of traditional data system and that leads to Internet of things which stimulate [1], the quantity of data which is being continuously generating with respect to different rate of its generation.so this needs the extreme scalable stream computing solution to process massive amounts of data for high data rates. So as a result the past decade has been the mirror of the many import applications need to process data that arriving in real time for distributed stream processing that is expensive treat streaming as a series of short batch jobs to down the latency. Parallel recovery technique truly cost-efficient recovery. But there will be class of streaming applications which are growing continuously in diverse domains like demand-response in smart power grids [2],and mission- critical [4], use-cases such as smart traffic signaling[3], trend analysis and social network modeling for online advertising[5,6].

So these kind of applications has to characterize by variety of input data streams with respect to variable data rates. Meanwhile the data which arrives at high velocity should guarantee the low latency in the presence of fluctuating data rate. A distributed stream processing engine inspired by the Map Reduce model. Where the engine is to solve real-world problems in the context of search applications that use data mining and machine learning algorithms. Current commercial search engines, such as Google, Bing, and Yahoo!, typically provide organic web results in response to user queries and then supplement with textual advertisements that provide revenue based on a “cost-perclick” billing model [7]. A major search engine may process thousands of queries per second, which may include several ads per page. An architecture that could be suitable for both research and production environments. The main requirement for research is to have a high degree of flexibility to deploy algorithms to the field very quickly for stream processing in large clusters. Many import applications need to process data arriving in real time for distributed stream processing is expensive that treats streaming as a series of short batch jobs to down the latency. To handle run time scalability and

full fault tolerance with respect to high velocity fluctuating data rate. In this paper we are concentrating and proposing Notion of "dynamic dataflow", realistic execution procedure on Cloud Infrastructure.

Exploit cloud elasticity and the flexibility offered by dynamic dataflow, based on Alternative paths in dynamic dataflow Deadline constraints to PE We propose a greedy allocation procedure to optimize the deadlines constraints, resource throughput and cost of execution. The proposed system will site at Cloud Scheduler and take care of allocating resource to the applications based on greedy allocation.

Recent studies and pre work of existing continuous data flow such as Esc [8] and stream cloud [9], have brought under control the cloud's scalability, selfmanageability, which will dynamically acquire and release the resources based on application load shown in Figure 1**.** But the several works [10-13] have shown that private and public clouds performance will vary for the different resources which will impact latency streaming applications across the data center .This leads to fail the resource allocation over static input data rate as well. So along with this the payper-use cost model for commercial of clouds needs efficient accurate resource management to satisfy the streaming application [14] QoS with respect to varied infrastructure.

Data fluctuation rate and their velocity of the existing stream processing system that consume and process continuous data scalability of the static resource with respect to fluctuating high or low leads of over or under allocation of resources which leads to resource wastage[5,7-9]. Self-manageability, scalability and elasticity with respect fluctuating input data rate to the dynamic infrastructure.

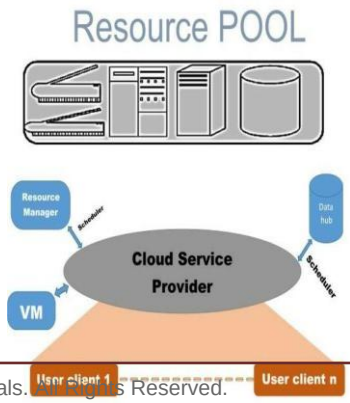

Fig.1:-*Basic resource allocation in cloud Architecture*

REACTIVE ADAPTATION APPROACH

We introduced the notion of "dynamic dataflow" a more realistic execution procedure on Cloud Infrastructure to exploit cloud elasticity and the flexibility offered by dynamic dataflow, to exploit cloud elasticity and the flexibility offered by dynamic dataflow with variable infrastructure such as:

- Deadline Constraint
- Alternate path
- Runtime scalability
- Handle variable data streams
- Need to minimize the real cost to satisfy the QoS of the application

Deadline Constraint

The existing system which is continuous in nature to be considered and being provided for the resource allocation will not be meeting the deadline constraint as the real time applications need to be deadline constraint. Our approach in this paper will provide a deadline constraint to the streaming application. Real time applications is our major concern to provide resource allocation As the existing continuous data flow system have only concentrated on the static allocation proactively and will not be as good as to provide with given deadline.

Alternate Path

The existing system which is continuous in nature to be considered and being provided for the resource allocation will be less manageable when any of the processing element which has been in the network and failed to work that leads to less selfmanageable and effect the entire process output. So to not to harm and effect the entire process output will provide an alternate [19], approach that is if the selected path fails then will go for an alternate resource selection for the allocation.

Runtime Scalability

Data fluctuation rate and their velocity of the of the existing stream processing system that consume and process continuous data scalability of the static resource with respect to fluctuating high or low leads of over or under allocation of resources which leads to resource wastage. So scalability [16] with respect to fluctuating input data rate to the dynamic infrastructure will not create an imbalance of storm.

Handle Variable Data Streams

There are class of streaming applications which grows continuously in diverse domains like demand-response in smart power grids, and mission-critical use-cases such as smart traffic signaling: trend analysis and social network modeling for online advertising.

So these kind of applications has to characterize by variety of input data streams with respect to variable data rates. Meanwhile the data which arrives at high velocity should guarantee the low latency in the presence of fluctuating data rate. Our approach will manage the variable data streams.

Need to Minimize the Real Cost to Satisfy the QoS of the Application

Alternate path selection lead to manage the QoS and as well our application model cost with respect variable infrastructure. With static streaming application when the input data rate changes then the dynamic resource allocation proactively will be monitoring and to concise the application cost and will maintain the QoS well. The existing systems fail to maintain the QoS.

Based on the dynamic dataflow and cloud infrastructure models, we propose a deployment and autonomic runtime adaptation approach that attempts to balance simplicity, realistic cloud characteristics elasticity scalability [15,17] as shown in Figure 2**.** To allocate the resources in runtime by checking adaptively at that time for the client input application.

OPTIMIZATION PROBLEM

We build an architecture to design and develop an application that dynamically allocates best resource available at current time based on client's job requirement. Developed system should be scalable and should provide a better QoS. over varied input data rate streaming application model for the dynamic dataflow to represent the IaaS cloud infrastructure, then to formalize the optimization problem with respect to balance the application cost(domain value),QoS, and resource cost by representing the deployment and variable (runtime) resource allocation.

Algorithms: We present a Promthee Multi criteria decision analysis Algorithm and heuristic for deployment and runtime adaptation of continuous dataflow to solve the optimization problem, that provide the optimal solution for the low latency over high velocity data.

Runtime Application Model

There will be a model which allow us to compose loosely coupled application from individual task that has data dependency because of continues data flow between them. The streaming application such as downloading a video, audio and any file. A continuous dataflow is a group of heuristic values such as bandwidth, availability, CPU, which will be the multicriteria for the application.

We make several practical assumptions on the continuous dataflow processing framework:

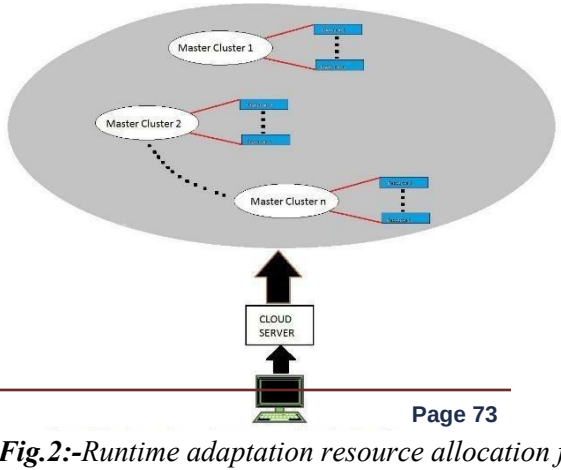

Fig.2:-*Runtime adaptation resource allocation for*

The dataflow application is deployed on distributed machines, with a set of instances of each physical or virtual machine ('(t) at time t) running in parallel across them. Incoming messages are actively load-balanced across the different PE instances based on the processing power of the CPU cores they run on and the length of the pending input queue. This allows us to easily scale the application by increasing the number of PE instances if the incoming load increases.

Within a single multi-core machine, multiple instances of different PEs run in parallel, isolated on separate cores. A core is exclusively allocated to one PE instance. We assume that there is minimal interference from system processes.

Running n data-parallel instances of a PE on n CPU cores, each with processing power $\pi = 1$, is equivalent to running 1 instance of the PE on 1 CPU core with $\pi = n$. The active alternate for a PE is not dependent on the active alternate of any other PE, since each alternate for a given PE follows the same input/output format. This allows us to independently switch the active alternate for different PEs during runtime.

The framework can spin up or shut down cloud VMs on-demand (with an associated startup/shutdown latency) and can periodically monitor the VM characteristics, such as CPU performance and network bandwidth.

RELATED WORK

The Runtime adaptation as shown in the above model where the streaming
application with varied input data rate [79, 19] such as downloading and uploading audio and video file. The request will come from client to the cloud server.

Then Cloud server will start up and waits for the client and as well for the resource manager in the master cluster. The client which will give different data rate streaming application for downloading a video, if the client is not valid then it has to again need to deploy in the cloud server for the proper deployment to allocate resource. When deployed properly in cloud architecture it will maintain log for the client. The connected client will then request for the resource through resource manager. If resource manager is not connected to client then it will need to make a connection to client .After establishing a connection to request for the resource allocation if resource manager connects to resources. If not then resource manager need to again reestablish a connection between them. At this time when server forward the request to resource manager it will reactively Monitor the best resources with respect changing structure of resources and will allocate with high bandwidth resource to the requested client streaming application to complete the job. When there are multiresources connected to master cluster resource who will take care of allocating the best resource by reactively adapting the varying structure.

As Server start-up and opens connection. Then Connection of client and resource manager to server on specific port numbers. After start-up master cluster and provide its parameters such as (bandwidth, CPU, RAM, storage) then connect to server. Cloud server maintains logs of connected master clusters which build-up resources by providing parameters (bandwidth, CPU, RAM, storage) then connect to master cluster. Here Master cluster maintains logs of connected resources and Client start-up at specific port number and provides its credentials to connect. Authentication of client takes place at server side.

If credentials entered by client are valid then connection is established. Otherwise, he is considered as BOT attack and connection is denied. Once connection is established the client can request server for resource to perform its job. Server forwards the resource to master cluster.
Master cluster run Promethee Algorithm[18] and finds out best resource available at current time. Once resource is found for that particular job then the resource is allocated to client. Client runs its job on allocated resource.

Multi Criteria Decision Analysis Scheduling Algorithm

The Multicriteria decision analysis used for scheduling algorithm is used to allocate to reduce as much as possible the number of heuristics or parameter among one should be the best one to consider to allocate. It is not only

depends on basic data included but also on the decision – maker itself[18]. Individual preferences of each decision makers and belongs to outranking methods. Objectives must be classified and alternatives must be evaluated against all the criteria.
Let us assume the following multicriteria problem where there is a finite set of possible alternatives A and set of evaluation criteria. Along with maximized and minimized criteria to the optimization.

Usually it is a mathematical problem as there is no alternative optimizing all the criteria at the same time as shown below equation.

$$\max\{g1(a), g2(a), \dots, gj(a), \dots, gk(a) | a \in A\}$$

- $\begin{cases} \forall j : gj(a) \geq gj(b) \\ \exists k : gk(a) > gk(b) \end{cases} \Leftrightarrow$ aPb
- $\forall j : gj(a) = gj(b) \Leftrightarrow$ aIb
- $\begin{cases} \exists s : gs(a) > gs(b) \\ \exists r : gr(a) < gr(b) \end{cases} \Leftrightarrow$ aRb

The above equation describes how an alternative *s* is better on the criteria *r* even though if the better one on the criteria to decide is impossible with the additional information, with the abbreviations of P, I, and R as preference, indifference and incomparability respectively. Where the evaluation table can be given in the following to describe the selection among multi criteria.

a	$g_1(\cdot)$	$g_2(\cdot)$	$\cdots$	$g_j(\cdot)$	$\cdots$	$g_k(\cdot)$
a_1	$g_1(a_1)$	$g_2(a_1)$	$\cdots$	$g_j(a_1)$	$\cdots$	$g_k(a_1)$
a_2	$g_1(a_2)$	$g_2(a_2)$	$\cdots$	$g_j(a_2)$	$\cdots$	$g_k(a_2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_i	$g_1(a_i)$	$g_2(a_i)$	$\cdots$	$g_j(a_i)$	$\cdots$	$g_k(a_i)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_n	$g_1(a_n)$	$g_2(a_n)$	$\cdots$	$g_j(a_n)$	$\cdots$	$g_k(a_n)$

In the case of pairwise comparisons, an appropriate multicriteria method should provide the following information: a is preferred to b. a and b are indifferent. a and b are incomparable.

For each criteria *g* preference function is *p* over weights *w*, So to reduce possible number of incomparaibilities when it is not realistic to the problem. Then it is considered as fair procedure. Then generates a rank for each criteria When for a particular procedure, all the incomparabilities are systematically withdrawn the provided information can be more disputable the solution of a multicriteria problem depends not only on the basic data included in the evaluation table but also on the decision-maker himself.

It depends on the individual preferences of each decision-maker, preference, indifference and incomparability a certain number of iterations or the convergence of a fitness value, is met. As the algorithm explains how the streaming application could get an efficient resource on variable infrastructure. When a client request for efficient resource by sending through cloud server on cloud server on line 3. Client application if valid with varying input rate then will go through master cluster resource manager [20, 21] which is connected to many resources. These resources are having many heuristics such as availability bandwidth and CPU usage .By checking ranking criteria among the resources resource manager will schedule the efficient, on line 16. Then over all will interpret infrastructure architecture.

Algorithm: Runtime Adaptation Allocation
1: Procedure RUNTIME ADAPATAION ALLOCATION (Dataflow, CPU, bandwidth, availability, QoS)
2: while start up server
3: Server waits for client request and waits for resource manager
4: if client is valid
5: if no it has to maintain log to connect 6: if yes client will connect and maintain and request for resource
7: end if
8: if master cluster resource manager connects client
9: if no server makes again request and connects
10: end if
11: if yes checks for resource manager
12: end if
13: If resource manager connects to dynamic resources
14: if yes checks for the multicriteria evaluation

15: By evaluation value an efficient resource will get allocate at that time by the resource manager 16: Allocates resource to user
17: perform the job
18: end if
19: end while
20: stop

RESULTS

Evolution of the heuristic parameters through private cloud .The real –world characteristics could be emulated by private cloud architecture [20] such as IaaS. This can be incorporated execution of dynamic dataflow that is varied input data rate on variable platform to meet up Scalable and Elasticity that allocates best resource available with respect to dynamic dataflow without wastage of resource.

Server and resource manager component and initially requests served are 0 and simultaneously it increases the count as per the requests served. Then opens up specified port and waits for clients to provide its credentials and connect. Master cluster starts at specified port in declaration code and connects to server.

Then it will creates a resource by accepting the parameters then connects to master cluster by accepting master clusters Ip address and its port number on specified id address. By taking multi-dimensional array values that are specified in the code. Then computes mark of each resource and generates a rank for each resource in a cluster by taking mark that is generated in above code. So Master cluster finds the best resource available at current time based on the generated mark and rank. If the resource is idle then it is allocated to client otherwise, Resource will not allocate.

CONCLUSION

In this paper we have shown and expressed the need for reactive adaptation monitoring against the continuous static dataflow applications which will not meet the QoS of a streaming application who's input data rate and infrastructure variability

REFERENCES

1. L. Douglas. *3d data management: Controlling data volume, velocity, and variety.* Gartner, Tech. Rep., 2001
2. Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou,and V. Prasanna. *Cloud-based software platform for big data analytics in smart grids.* Computing in Science Engineering, July 2013.15(4):38–47p.
3. A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov,O. Verscheure, H. Koutsopoulos, and C. Moran. *Ibm infosphere streams for scalable, real-time, intelligent transportation service.* International Conference on Management of data. ACM SIGMOD,2010:1093–1104p.
4. M. Gatti, P. Cavalin, S. B. Neto, C. Pinhanez, C. dos Santos, D. Gribel, and A. P. Appel. *Large-scale multi- agent-based modeling and simulation of microblogging-based online social network.* Multi-Agent-Based Simulation XIV. Springer, 2014:17– 33p.
5. Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna. *Cloud-based software platform for big data analytics in smart grids.* Computing in Science Engineering, July 2013.15(4):8–47p.
6. J. Dean and S. Ghemawat. *Mapreduce: simplified data processing on large clusters.* Communications. ACM, 2008.51(1):107–113p.
7. L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. *S4: Distributed stream computing platform.* IEEE International Conference on Data Mining Workshops (ICDMW), 2010.
8. *Storm: Distributed and fault-tolerant realtime computation.* http://storm.incubator.apache.org/, accessed: 2014-06-30.
9. M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica. *Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters.* USENIX conference on Hot Topics in Cloud Ccomputing.2012:10–10p.
10. B. Satzger, W. Hummer, P. Leitner, and S. Dustdar. *Esc: Towards an elastic stream computing platform for the cloud*. IEEE. International Conference on Cloud Computing (CLOUD), July 2011:348–355p.
11. V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, and P. Valduriez.

Streamcloud: An elastic and scalable data streaming system. Transactions on Parallel and Distributed Systems. IEEE, 2012.23(12):2351–2365p.

12. A. Iosup, N. Yigitbasi, and D. Epema. *On the performance variability of production cloud services.* IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2011.
13. A. Li, X. Yang, S. Kandula, and M. Zhang. *Cloudcmp: comparing public cloud providers.* conference on Internet measurement.ACM SIGCOMM, 2010:1–14p.
14. I. Moreno, P. Garraghan, P. Townend, and J. Xu. *Analysis, modeling and simulation of workload patterns in a large-scale utility cloud.* Transactions on Cloud Computing. IEEE, April 2014.2(2):208–221p.
15. A. Kumbhare, Y. Simmhan, and V. K. Prasanna. *Exploiting application dynamism and cloud elasticity for continuous dataflows.* International Conference for High Performance Computing, Networking, Storage and Analysis. ACM.2013:57p.
16. W. M. van Der Aalst, A. H. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. *Workflow patterns.* Distributed and parallel databases. Springer, 2003.14(1):5–51p.
17. Y. Simmhan and A. G. Kumbhare. *Floe: A continuous dataflow framework for dynamic cloud applications.* CoRR, vol. abs/1406.5977, 2014.
18. Service de Mathématiques de la Gestion Université Libre de Bruxelles Boulevard du Triomphe CP 210-01,
19. IEEE *Transactions on Cloud Computing*