ISSN 2047-3338

Trends and Prospectives of the Dynamic Resource Management Using Adaptive Techniques in Distributed System

V. S. Tondre[1], Dr. V. M. Thakare[2], Dr. S. S. Sherekar[3] and Dr. R. V. Dharaskar[4]
[1]B.B.Sc.C. Amravati, (M.S.), India
[2,3]S.G.B. Amravati University, Amravati, (M.S.), India
[4]M.P.G.I., Nanded, (M.S.), India
[1]varshatondre@rediffmail.com, [2]vilthakare@yahoo.co.in, [3]ss_sherekar@rediffmail.com, [4]rvdharaskar@rediffmail.com

***Abstract*– Dynamic resource management has been a hot spot in recent years. As the applications like distributed real-time and embedded system can be benefited from dynamic management of computing resources and networking resources by optimizing, reconfiguring system resources and distributing critical applications over real time communication network at runtime in response to the changing situations. These types of applications typically consist of a set of tasks executing on different hosts, exchanging messages over high speed network. In distributed environment, management of resources both computing and networking, resource allocation and resource utilization are the most crucial problem. In this paper, an extensive review has been made on these architecture, methods and techniques for the verity of applications that exhibit adaptive and dynamic approach. All the techniques are classified and analyzed in detail.**

***Index Terms*– Dynamic Resource Management, Resource Allocation, Resource Utilization and QoS Application Resource Management**

1. INTRODUCTION

IN recent years, the requirement of the high-end computational capabilities and resources are increasing. Applications like distributed real time systems and embedded systems, communication networks, e-sciences, e-governance are becoming gradually more common and important as the underlying technology for distributed computing and networking systems continuing to grow-up.

The distributed system provides a large-scale resource sharing such as personal computers, cluster, memory, CPU, databases and online instruments. These resource sharing may be a cross domain, dynamic and heterogeneous. Therefore, to utilize these hybrid heterogeneous computational resources efficiently, the most vital problem is job scheduling and resource management in distributed system.

In distributed run time system, Dynamic resource management (DRM) plays an important role. The DRM's main goal is to adopt system resource allocation to dynamically change conditions. For example, changing priorities of the jobs and resources failures, to maximize the system performance etc. Simultaneously, it manages the Quality of Service (QoS), which is concerned among the jobs that maintain system operations even with partial system failures. Another approach is the adaptive software and resource system QoS service optimization.

To manage resources and allocate resources efficiently at run time, several researchers are pay attention to give an adaptive or dynamic solution.

In this paper, the focus is on the study of the run time process management of application on adaptive Dynamic resource management.

II. CLASSIFICATIONS OF METHODS

In this paper several methods for DRM using adaptive techniques are studied. According to that all methods are categorized into three different classes as shown in Table I. It is classified under three categories functional, technical and architectural.

III. RUN TIME PROCESS MANAGEMENT

In this paper, several methods and techniques are studied for run time process management of different applications for adaptive dynamic resource management. In section 3 various methods for adaptive dynamic resource management are discussed, section 4 presents the analysis of the methods, and in section 5 conclusion and future scope are discussed.

A. Hierarchical Control System for Dynamic Resource Management

Usually, distributed real time systems are designed with static resource management for specific missions. As these allocation strategies are not capable to adopt changes in system goals, resources and environment. Therefore, DRE systems may fail to meet QoS, when condition change.

In [1], present a hierarchical control system for the dynamic resource management system. In this approach, the utility function is defined at each and every level of resource hierarchy. As shown in the figure of system-mission-string hierarchy, the high-level DRE system is controlled by system controllers. At the middle level, missions are controlled by mission controller. At the lowest level, strings are controlled by string controller. All the controllers in the hierarchy communicate with their parent and/or child. The control layers communicate with each other in bottom up approach [1].

In the string utility, a drop is occurred due to the resource failure or contention. When the resource fails, string controller expected to receive notification from the

TABLE I: CLASSIFICATIONS OF METHODS

Functional	Technology	Architecture
Hierarchical control system	An adaptive Connection admission control (CAC)	Resource allocation among high level autonomous mission controller
MMC function	An agent based DRM model	DRM Framework
	DRCM system	Hybrid Adaptive Resource management Middleware (HyRAM)
An adaptive Connection Admission Control (CAC)	Adaptive admission control algorithm and Hose model	Multi Layered Resource Management
	Two layered controller	Adaptive Resource Management middleware architecture
		Reflective resource management framework
		System sensitive partitioning and load balancing framework
		Hierarchical Distributed Resource Management Architecture (HiDRA)
		Adaptive admission control algorithm and Hose model
		PAMS Management Service

Resource Status Service (RSS). If the RSS is not received, it considers the drop is due to the resource contention. These resources failures, which result in drops in a strings utility, are handled by the mission controller. The mission controller again reallocates the resources system-wide.

B. High-level DRM for Distributed Real-time Embedded Systems

As in the previous presented method in [1], mission controllers do not have the sufficient resource rights to redeploy applications in the system from the failed resources to operational resources. Due to the computational complexity issues, it is infeasible to find a resource allocation which will achieve the optimality in the reasonable amount of time for a real world system.

Therefore, to coordinate the resource allocation among independent high level goals in a distributed real-time embedded system. The authors in [2] presented heuristic distributed methods that find resource allocations with as high utility.

To implement the DRM, author design the Multi Mission Coordinator (MMC) as shown in Fig.1. It is a design for single system wide, which manage gross level resource allocation among the multiple missions. MMC operations are performed on system events such as partial system failure or changes in mission priorities. In this approach, mission controller sends mission tables to the MMC; contain complete information about the resource requirements, according to the levels of utility that could be provided by the mission [3], [4]. The choice of the quantization levels in the tables are tradeoff between the accuracy of information about resources requested by the missions to provide various levels of quality of service and the communication overhead associated with transmitting this information .

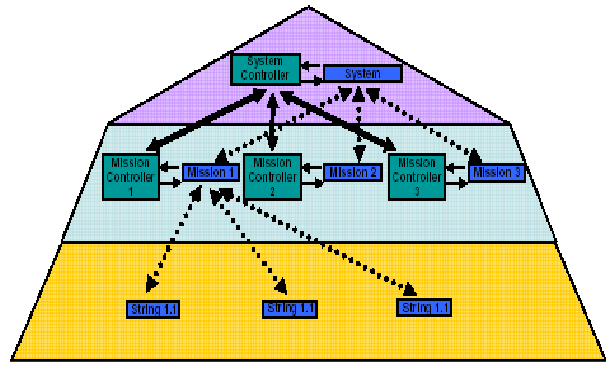

Fig. 1. Hierarchy of DRE System

C. Novel DRM and Job Scheduling in Grid Computing

In the grid computing environment resource management and job scheduling is the most essential problem. Authors in [5] tried to solve this problem by applying the novel agent based grid resource management model. In this model, the grid resource management and job scheduling both concepts are considered as combination integrity.

An agent-based DRM model is defined by using the two layered Heap Sort Tree (HSP) based on the Heap Sort algorithm as shown in Fig. 2. In this model, two kinds of agents are deployed with every node of the grid computing system. The first is Autonomy Representation Agent (ARA) is used to represent the complete autonomy in the top layer of the model and also used to represent the very node selected out from the autonomy, whose current available computational ability is the largest among all the nodes of the autonomy. The second agent is the NSMA (Node State Monitoring Agent) agent which is used to obtain real time available computational ability of various nodes in the grid system. In this way one can assign the new job to the node.

D. Adaptive control of Virtualized Resources

The adaptive resource control system is developed in [6] to dynamically control the resource allocation to individual components of complex and multi-tier enterprise applications in a shared host environment. The main goal of this system is to achieve application level QoS with high resource utilization in virtual data centers where multi-tier applications share a common pool of server resources. For this development the classical control theory is used as a basis for modeling and designing feedback resource driven system. The controller is used to adaptively adjust to varying workloads. The suggested two layered control architecture is used to describe the dependencies and

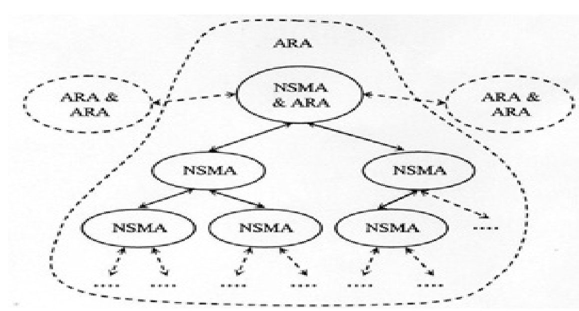

Fig. 2. Sub-HST: Construction of ARA (Autonomy Represented by Agent) (NSMA: Node State Monitoring Agent)

interactions among the multiple tiers in application stack when making resource allocation decision. The utilization controller is used to control the resource allocation for a single application tire and arbiter controller, control the resource allocation across multiple application and multiple application stacks sharing the same infrastructure utilization.

E. Adaptive Connection Admission Control for Mission Critical Real-time Communication Network

To achieve the efficient and effective connection admission control for mission critical application distributed over real time communication networks in [7] an adaptive approach is implemented. In real time communication network, usually a set of tasks are submitted for execution on different hosts and exchange the message over the high speed network.

In the connection oriented communication, the main part is connection establishment before transferring actual data. Connection management is network function, which actually set up, maintains and tears down the connections. The most critical part of the connection management is Connection Admission Control (CAC). In [7], an adaptive strategy is applied on CAC. As shown in Fig.3 the adaptive connection management is divided into two major threads: one for admission control and other for connection termination. In Connection Admission Request (CAR), the connection request specifies an adaptation policy through the QoS shrinkage, which is used when the guarantee test fails.

The adaptation mechanism plays i.e. QoS shrinkage determine the level to which the QoS of a selected set of connection need to be shrunk to successfully admit the new connection at QoS, then resource allocation follows. It free up the enough resources when connection is rejected to admit new connections. Again the connection is admitted, the adaptive CAC allocates resources for the new connection.

A Connection Termination Request (CTR) has the parameter Expansion Directive Sequence and the adoption policy of the connection. EDS determines the level to which the QoS of a selected set of connections can be increased using the resources released at connection termination. When the valid CTR appear, the adaptive connection management module releases all resources that were used at the time of connection. The required resources are then re-allocated to support the increased QoS.

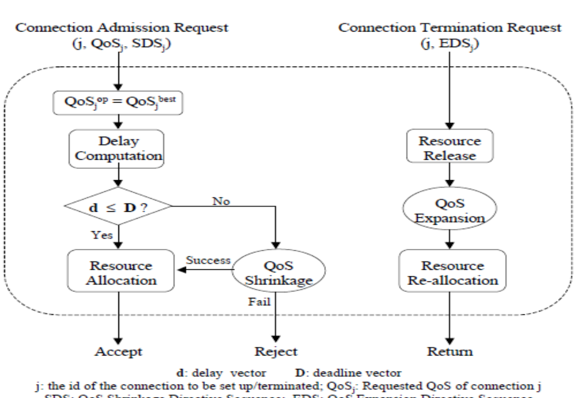

Fig. 3. Adaptive connection management

F. Adaptive Resource Management for Distributed Real-time Embedded Systems

A challenging problem faced by researchers and developers of distributed real-time and embedded (DRE) systems is devising and implementing effective adaptive resource management strategies that can meet end-to-end quality of service (QoS) requirements in varying operational contexts. In [8] presents three contributions to research on adaptive resource management for DRE systems. First, describe the structure and functionality of the Hybrid Adaptive Resource-management Middleware (HyARM), which provides advanced distributed resource management based on hybrid control theoretic techniques to monitor system utilization and adapt to fluctuations in workload. Second, presents an analytical model of HyARM that formalizes the control theoretic behavior of HyARM and conveys the relationship between the system resource utilization and application QoS. Third, highlight the adaptive behavior of HyARM via experiments on a DRE multimedia system that distributes video in real-time.

HyARM employs hybrid control techniques to provide the adaptive middleware capabilities, that are the key to providing the dynamic resource management capabilities for open DRE systems. They employed HyARM to a representative DRE multimedia system that is implemented using Real-time CORBA and the CORBA A/V Streaming Service.

G. System Sensitive Run-time Management

In [9] presents the design and evaluation of a system sensitive partitioning and load-balancing framework for distributed adaptive grid hierarchies that underlie parallel adaptive mesh-refinement (AMR) techniques for the solution of partial-differential equations. The framework uses system capabilities and current system state to select and tune appropriate distribution parameters (e.g., partitioning granularity, load per processor).

A presented framework uses the current system state of the computing nodes to distribute an application among the nodes. The framework monitors the availability of resources at the computing nodes, calculates the relative capacities of these nodes, and then assigns work in proportion to their capacities. This scheme reduces the total execution time of the application and the load imbalance as compared to a scheme that does not take the relative capacities of the computing nodes into account.

H. PAMS Adaptive Management Services

Management of large-scale parallel and distributed applications is an extremely complex task due to factors such as centralized management architectures, lack of coordination and compatibility among heterogeneous network management systems, and dynamic characteristics of networks and application bandwidth requirements. In [10] present an approach to implement a Proactive Application Management System (PAMS). PAMS architecture consists of two main modules: Application Centric Management (ACM) and Management Computing System (MCS). The ACM module provides the application developers with all the tools required to specify the appropriate management schemes to manage any quality of service requirement or application attribute/functionality (e.g., performance fault, security, etc.). The MCS provides the core management services to enable the efficient proactive management of a wide range of network applications. The services offered by the MCS are implemented using mobile agents. Furthermore, each MCS service can be implemented using several techniques that can be selected dynamically by invoking the corresponding mobile agent template for the service implementation. For example, the overhead incurred in the application fault management to tolerate one task failure, two task failures, and three task failures in a medium to large size application is less than 0.02%.

PAMS implementation is based on using mobile agents that can be programmed to maintain the quality of service requirements of distributed applications. They have evaluated three adaptive techniques to manage the performance and fault tolerance of distributed applications. The first approach is based on using active redundancy to improve performance and tolerate faults. The second approach is based on passive redundancy in which a set of machines is designated as backup machines to be used to replace any of the machines assigned to the application tasks in order to improve performance or to tolerate software/hardware failures. The third approach does not introduce redundancy in the system and it requires task migration to another machine in order to improve performance or to tolerate software / hardware failures.

I. Adaptive Resource Management for Dynamic Distributed Real-time Application

The dynamic distributed real-time applications run on clusters with varying execution time, so re-allocation of resources is critical to meet the applications' deadline. In [11] present two adaptive resource management techniques for dynamic real-time applications by employing the prediction of responses of real-time tasks that operate in time sharing environment and run-time analysis of scheduling policies.

In [11], uses prediction of response time for resource reallocation is accomplished by historical profiling of applications resource usage to estimate resource requirements on the target machine and a probabilistic approach is applied for calculating the queuing delay that a process will experience on distributed hosts.

J. A Multilayer Resource Management for DRM in Enterprise DRE Systems

In [12] presented a standard based Multi-Layered Resource Management (MLRM) architecture that provides DRM capabilities to enterprise DRE system. In this architecture, it defined different three layers of abstraction that are DRM service layer, resource pool layer, physical resources layer.

The main goal of the DRM service layer is to maximize the mission coverage and reliability. The key components of this layer: infrastructure allocator and operational string manager. Infrastructure allocator decides the resource pool(s) in which a package's operational string(s) are arranged and operational string manager coordinates the proper arrangement of the operational string.

The responsibility of the layer is to allocate the computing system and network resources to tracks items of interest and to plan necessary actions. It also coordinates and monitors the execution of the operational strings that are sequence of components which capture the partial order and workflow of a set of executing software capabilities.

The second layer of the architecture is the Resource Pool Layer. It is the abstraction for a set of nodes. These nodes are managed by the Pool manager through the interaction with Resource allocator to run algorithm that arrange application components to various nodes within the resource pool.

The third layer is the Physical Resource Layer. It deals with only specific cases of resources in the system. To support the QoS need of a mission, it configures the physical resources according to the dynamically generated allocations.

K. DRM for Adaptive Distributed Information Fusion in Large Volume Surveillance

In [13] presented Dynamic Resource Configuration Model (DRCM) for the surveillance of the distributed information fusion in large volume. As the surveillance of the large volume of data demands the robust and scalable network architecture that works in an adverse and uncertain environment, the ability to flexibly adopt to dynamic changes in available mobile resources.

In order to make the system highly adaptive and network enabled, the numbers of issues are considered in network architecture. The two main aspects of the architecture are resource configuration management and task management.

Here, discuss the task management of the DRCM. The DRCM assign high level abstract tasks to high level nodes in command hierarchy. If there is a single child node or a combination of children capable of performing the task, then the abstract tasks bubble down the tree. If there is no abstract class then the task is decomposed into subtasks. This process provides the set of subtasks that include the process of orchestration of the sub tasks. Some tasks that are unordered in a sequence are also included in the sequence so that the sequence is performed and sort these tasks with respect to the availability of resources, time constraints, work balance and etc. It creates the partially ordered sequence in that some of the elements left ordered so that there is no priority between those elements. This type of decomposition is representation as control state ASM diagram. It is treated as subtask.

DRCMA is a distributed process that carried out by individual nodes of the network. Each and every node goes through a cycle of three activities: observation, communication, process. Once a task in injected into the network, it goes through the lifecycle. It means from assigning a node to being completed or rejected. It is the integrated approach for dynamically manage the resource configuration and task execution. The author used Core ASM as platform for validation.

L. A Resource Management for Adaptive Middleware

In arrived new technologies such as Internet, mobile computing, etc, required a new middleware platform. A middleware is used to solve the problem of heterogeneity in distributed environment. But the mobile computing may require a more hostile environment. Therefore, it requires including adaptation in middleware so that it is able to deal with the new demanding issues.

In the process of adaptation, resource management plays an important role with resource awareness and dynamic reallocation of resources. It must allow a system to be aware of the availability of computational resources. The management policies are used to allocate the resources among the activities that system performs. In [14], the concept of reflection is used to achieve the openness and flexibility into middleware platform. In [14], introduces architecture for reflective middleware with introspection and adaptation capabilities.

In the system architecture, one of the important parts is Meta-space, which contains the four Meta-models. Out of that, Resource Meta-model is responsible for resource awareness and resource management of objects. In this model, it maintains the hierarchy of resource abstraction and a factory of hierarchy. Here, a manager allows its users to share the resources it manages by providing an abstract view. And factories are in charge of creating new instances of abstract resources. Virtual Task Machine (VTM) is uses for the top level of abstraction. It includes an abstraction of the resources allocated to a particular task in the system and information about the management policies of the resources.

A new task oriented approach is implemented into the resource model. In this approach, a system is divided into tasks according to the type of activities it performs. Each and every task is represented into the resource model. Therefore, a VTM represents the abstract resources that task uses for execution. It also achieves both fine and coarse-grained reconfiguration of resources and a task oriented approach is that the changes in the underlying resources of a system are localized.

M. Heuristic Algorithms for Adaptive Resource Management for Periodic Tasks

In [15] presents adaptive resource management middleware architecture for real time computer systems. This system performs distributed mission management. It presents a heuristic approach for adaptive resource management of periodic tasks in dynamic real time systems with the objective of minimizing missed deadline ratios.

In the middleware architecture, two core components are included in the architecture discussed in [15] that is resource manager and a system daemon. The resource manager performs several tasks that are responsible for collecting and maintaining all application information such as dynamically measured performance metrics of applications and hardware resources like CPU and memory utilization of hosts. This information is reported to the resource manager periodically through the system daemon that is set one per host machine [16], [17]. It also calculates the CPU and memory utilization by parsing the data and same calls are used to evaluate the host metrics on another operating system. The resource manager continuously watches the degrading timeliness state and receives time-stamped event tags from the application programs, transfer them in task latencies. It compares the latencies against task deadlines for detecting degrading timeliness state. If the task shows low timeliness then task manager detects the state and determine the components of the task that shows a low timeliness. After that it performs allocation analysis to identify the possible resource allocation action for the task which will improve the task timeliness. Once the action for the task, the resource manager selects the resources for the actions, the resource manager notifies the system daemon for enacting the actions. The system daemons are also responsible for notifying the resource manager when failures of application programs occur. The system daemon on the host of the failed program immediately alerts the resource manager.

The above method is implemented using certain algorithms that algorithm produce lower missed deadline ratios. Therefore, profile computation and regression equations are effective. The limitation of the method is the application model used here is assumed to be adaptable by replicating some of the bottleneck subtasks. In general, it may not be with true with all dynamic real time applications. The application subtasks must be programmed so that it interfaces with the resource management middleware and stay on its communication semantics.

N. Adaptive Resource Management for DRE System

In [18], present a Hierarchical Distributed Resource management Architecture (HiDRA) for the increasing demand for the adaptive capabilities in distributed real time and embedded system. It execute, where operational conditions, input workload, resource availability is previously not known is called open environment.

In HiDRA, a control theoretic approach is used to concurrently manage processor and network bandwidth. As shown in the Fig.4 control frame work consists of three entities: monitors, controllers and effectors. A monitor is periodically updates the controller with the current resource utilization. It is associated with a specific system resource. The controller employs an algorithm and computes the adaptation decisions for each application. It achieves the desired system resource utilization.

Each effector is associated with an application. It achieves the controller recommended application adaptation by modifying application parameters.

O. Adaptive Admission Control for Bandwidth Brokers

The Diffserv architecture with individual flows at the edges of a domain is offer the better scalability. So, it allows the core elements of the network to only handle class of service. In this framework, a Bandwidth Broker (BB) is an object which is responsible for providing QoS within a

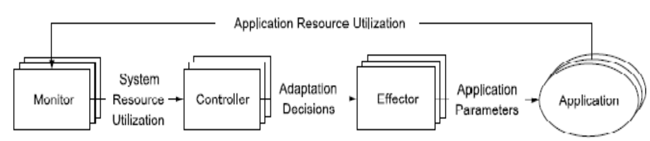

Fig. 4. Control framework for HiDRA

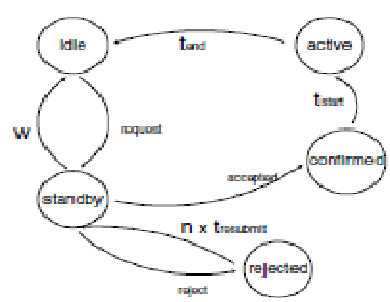

Fig. 5. Request states

network domain. The BB manages the resources within the specific domain. It is done by controlling the network load and by accepting and rejecting the bandwidth request. The above Fig. 5 shows the different states of the request. The standby requests are the requests that have not up till now received an answer. Confirmed is a book-ahead request that has received an affirmative answer but waits to be activated. The hose model is used which is also used in VPN provisioning [19].

In DiffServ domain an aggregated approach is used at core routers. An adaptive admission control algorithm [19], the bandwidth management is simplified by assigning a limit at the bandwidth that each edge router is allowed to accept in the domain. The edge device sends the traffic to a set of endpoints without having the detailed traffic matrix. Due to aggregation of flows between endpoints, the edge router reduces the size of access links through multiplexing gains. The unifying parameters are used for the admission control and dimensioning of network resources, which can be computed based on traffic parameters such as mean rate or maximum burst size.

Bandwidth broker's admission control module keeps a list of unwanted requests, which it call waiting queue [20]. It is sorted according to their waiting time. As soon as the first item is about to expire the admission control module calculate the answers that it will provide to this and a number of other requests. Basically, it is done by offline scheduling problem.

IV. ANALYSIS

The Table II shows the analysis of the methods. All the methods are belongs to the adaptive dynamic resource management.

The Table II shows that in all methods the parameters like QoS, resource utilization, resource management, resource allocations, and task management are considered for adaptation.

In the Table III, we analyzed the methods in detail.

TABLE II: ANALYSIS OF METHODS

System	Parameter	Author	Year	Adaptive Technique / Method	Advantage	Limitation
1. Hierarchical control system for DRM	Resource allocation, QoS	K. Rohloff and et al.	2006	Utility function is defined at each and every level.	1. Capable of managing multiple resources and QoS. 2. It reflects the local performance with respect to the QoS requirement and overall system performance.	1. Mission controller do not have sufficient rights to redeploy applications in the system
2. Hierarchical control system for high-level DRM	Resource allocation, QoS	K. Rohloff and et al.	2006	Implement the MMC	1. It is scaleable. 2. It achieves a good level of utility. 3. it gives a better performance under the condition of high resource deficiency.	1. The design of MMC is centralized.
3. Novel DRM and job scheduling in grid computing	Resource allocation, Computational ability & load balance	F. Li & et. al.	2006	An agent-based DRM model with two layered Heap Sort Tree based on Heap Sort Algorithm	1. The node has the largest computational power due to the execution of a set of the little benchmark. 2. It improves the system load ability and performance with load balance. 3. Algorithm and model are feasible and rational	1. It is not worked in real grid environment.
4. Control of Virtualized Resources	Resource utilization & QoS	P. Padala & et. al.	2007	Classical control theory is used & two layered control architecture is designed.	1. It is able to maintain high resource utilization of data centers. 2. It provides a specified level of QoS differentiation between applications under overload conditions. 3. Adaptive integral controller with a self tunning gain makes a better tradeoff between stability and efficiency.	1. Only static models are used. 2. It is not share the other resources.
5. Connection admission control for mission critical real time connection network	Resource allocation, QoS	B. Devalla & et. al.	1998	An adaptive strategy is applied on Connection Admission Control (CAC)	1. Flexible QoS Model. 2. The resources are dynamically re-allocated in order to meet the needs of incoming connection. 3. It improves the connection level flexibility. 4. Practical and compatible technology.	1. It is not supported to delay and fault tolerant techniques.
6. Resource management for distributed real time embedded systems	Resource utilization , QoS , jitter, latency, resolution, frame rate	Nishanth Shankaran & et.al.	2005	Hybrid Adaptive Resource Management (HyARM)	1. It provides effective resource management to DRE systems. 2. It provides the resource capabilities like resource monitoring and application adaptation. 3. Improve application QoS and increase the system resource utilization. 4. Provide the better predictability to QoS enabled applications.	1. Constant QoS attacks degrade the performance of the system. 2. Due to the attacks it losses its system resources.
7. System Sensitive Run Time Management.	Node capability, Load per processor,	Shweta Sinha and Manish Parashar	2001	System sensitive partitioning and load balancing framework	1. It maximizes the overall performance of the application. 2. It reduces the total execution time of the application load imbalance.	…..

	Time, partitioning					
8. Management of large scale parallel/distributed application	QoS, Fault, Performance, security	Yoonhee Kim & et. al.	2000	PAMS Management Service	1. It dynamically manages the performance and fault of the application. 2. The agent based approach lead to significant gains in the performance and low overhead fault management.	1. It is centralized management.
9. Resource Management for dynamic distributed real-time application	Response time, queuing delay for Resource allocation	Huh & et al.	2006	Adaptive resource management technique	1. Response of real-time task operates in time sharing environment.	……
10 Multi-layered Resource Management for DRM in DRE system	Resource allocation, QoS	Patrick Lardier & et. al.	2006	Multi-layered Resource Management architecture	1. It provides the DRM capabilities. 2. MLRM is feasible as well as it can handle dynamic resource allocation in wide array of configuration. 3. It also provides continuous availability even in the presence of node and pool failure. 4. It provides effective software infrastructure for DRM.	----
11.DRM for distributed information fusion in large volume surveillance	Resource configuration & task management.	R. Farahbod & et. al.	2009	Dynamic Resource Configuration Model (DRCM)	1. It is dynamically manage the resource configuration and task execution. 2. Ability to flexibly adapt to dynamic changes in available mobile resources.	1. The concurrent and reactive behavior of the underlying algorithm and protocols does not predict the resulting properties accurately.
12. Resource Management for middleware	Resource awareness	Hector A. Duran & et. al.	2000	Architecture for reflective middleware with introspection and adaptation capabilities and task oriented approach.	1. It provide a various levels of resource abstraction as well as hierarchies of both resource managers & resource factories	QoS is not considered in resource model
13. Resource Management for periodic tasks	Resource utilization, task management.	Ravi K. Devarasetty	2001	Adaptive Resource Management middleware architecture	1. It is capable of automatically profile the application tasks to determine regression equations. 2. It minimizes the aggregate missed deadline ratios when there are multiple application tasks.	1. The application subtasks must be interfaces with the resource management middleware. 2. The design of the application subtasks behavior must be amenable to regression analysis.
14. Resource Management for DRE system	QoS, Resource Utilization	Nishanth Shankaran & et. al.	2008	Hierarchical Distributed Resource Management Architecture (HiDRA)	1. It provides efficient resource utilization by maintaining system resource utilization within specified bounds. 2. It gives the performance even under the fluctuating workloads. 3. It ensure system stability and delivering effective QoS.	1. QoS attacks are not considered in HiDRA. 2. Resource Contention is not handled in HiDRA.
15. Admission control for Bandwidth Broker	QoS, Network traffic	Ch. Bouras and K. Stamos	2007	Adaptive admission control algorithm and Hose model	1. It offers the better scalability. 2. BB manages the resources within the specific domain.	

TABLE III: CRITICAL ANALYSIS OF TECHNIQUES

Method	Technique	Input Parameters	Output Parameters	Constraints	Result
1. Hierarchical control system for DRM	Utility function is defined at each and every level.	1. Computation resources 2. Communication resources	1. Quality 2. Throughput	Timeliness	Quality and throughput are continually decrement until a local maximum of the measured string utility is found.
2. Hierarchical control system for high-level DRM	Implement the MMC	1. Accuracy of Information about resources required. 2. optimality 3. Resource available.	1. QoS, 2. Communication overhead.	Reasonable amount of time.	The ratio of the performance for the dynamic and static MMCs varies with resource deficiency. As post-failure resource deficiency increases the MMC performance increases better with respect to baseline

					system.
3. Novel DRM and job scheduling in grid computing	An agent-based DRM model with two layered Heap Sort Tree based on Heap Sort Algorithm	1. real time information of node 2. CPU, memory, network bandwidth,	1. fault tolerant ability 2. Intelligent ability to adapt to diverse circumstances. 3. load balance	Job submitted to top –HST at current time to be fulfilled.	After submitting the sample jobs to the system it shows the design is feasible and rational and it works well. It is robust, scalable, and efficient.
4. Control of Virtualized Resources	Classical control theory is used & two layered control architecture is designed.	1. CPU utilization.	1. application level QoS 2. Maximum Throughput 3. Low response time.	Response time and Loss ratio of the Arbiter controller	In spite of changes in resource demands throughout the run, to achieve the maximum throughput and low response time, except during the transient period.
5. Connection admission control for mission critical real time connection network	An adaptive strategy is applied on Connection Admission Control (CAC)	1. QoS as fixed values. 2. Calculated delay	1. Best possible QoS on available resources. 2. Performance guarantees on end-to-end message transfer delay. 3. Efficient adaptive CAC.	Dalay calculated quickly means in respective deadline.	1. Increase the utilization of system resources and improve QoS.
6. Resource management for distributed real time embedded systems	Hybrid Adaptive Resource Management (HyARM)	1. Utilization of resources (low/high). 2. application workload	1. The QoS of received data such as jitter and average latency is affected. 2. Qos may degrade	Period of time.	The network utilization in HyRAM was as high as 0.9 during increase in work load condition. It is greater then the utilization set point of 0.7 by 0.2.
7. System Sensitive Run Time Management.	System sensitive partitioning and load balancing framework	1. work load 2. resource monitoring	1. decrease the memory and CPU availability of the processor 2. CPU time available to processor, TCP network latency, bandwidth, free memory, disk space.	Time Interval	1. It decreases the processor capacity to do any additional work. 2. reduces the total execution time of application and load imbalance as compare to other scheme.
8. Management of large scale parallel/distributed application	PAMS Management Service	1. Work load 2. resource monitoring.	1.Provide Application requirement in terms of performance, fault, security, appropriate management scheme	Time	The task performance may degrade due to bursty traffic conditions or due to s/w or h/w failures.
9. Resource Management for dynamic distributed real-time application	Adaptive resource management technique	Resource usage to estimate resource requirements on the target machine.	Queuing delay	varying execution time	Compare to other technique this method is used system resource more efficiently.
10 Multi-layered Resource Management for DRM in DRE system	Multi-layered Resource Management architecture	Varying conditions Computing resources, Network resources, Various QoS mechanism	QoS and operational requirement.	Time	1. Recovery from the data center failure is achieved in the half time compare to manual recovery. 2. Multiple dynamic allocations were achieved. 3. Resource allocation adjustment to permit steady state increase in resource availability.
11.DRM for distributed information fusion in large volume surveillance	Dynamic Resource Configuration Model (DRCM)	Higher order node, task	Task decomposition and output worked as input to the next task sequence.	Time , Resource failure	It is more beneficial to run an executable model to measure the result with respect to certain criteria such as time limitation, resource failure.
12. Resource Management for middleware	Architecture for reflective middleware with introspection and adaptation capabilities and task oriented approach.	1. abstract resources 2. scheduler 3. VTM 4 thread 5 buffer memory	1. Hierarchy of resource abstractions. 2. unit of resource management 3. Reconfiguration of resources.	------	The task oriented resource makes ease for customization and the dynamic reconfiguration of the resource.
13. Resource Management for periodic tasks	Adaptive Resource Management middleware architecture	1. Concurrency can used. 2. Task deadlines	1. Latency of tasks reduced. 2. Degrade timeliness situation.	Time slack	Highest number of replicas offers the highest steady state. Algorithm employing the regression equation computed using the automatic profiling method for prediction of number of replicas of tasks perform better than employing equations derived analytically by hand.
14. Resource Management for DRE	Hierarchical Distributed Resource	1.Bandwidth availability,	1. Bandwidth utilization.	Time	HiDRA respond to: 1. Resource utilization

system	Management Architecture (HiDRA)	2. workload	2. Processor utilization. 3. application QoS target-tracking precision and average end-to-end delay		increases above the desired set point. 2. System resources are not over-utilized 3. Enough resources are available for important applications.
15. Admission control for Bandwidth Broker	Adaptive admission control algorithm and Hose model	1. Request for resource, 2. Resubmission of request. 3. adaptation parameter and threshold	1. Performance of the computation time threshold. 2. effect of adaptation parameter on algorithm 3. total available bandwidth, duration of each simulation, minimum and maximum reservation request. 4. acceptance rate, network utilization, delay, average size	Duration of simulation 50 time slots.	After comparing the two algorithms, SAC algorithm slightly achieves the highest acceptance rate. AAC and AACR required more efficiency in the utilization of the network resources

In this Table III, it is seen that in each and every method, the constraint time was considered. By employing the adaptive method the resource utilization and network utilization was increased compare to simple other methods in the different applications. All the adaptive methods were efficient, easy, scalable and robust.

V. CONCLUSION AND FUTURE SCOPE

In this paper, DRM with adaptive techniques for real time and embedded systems are essential. In this paper various DRM and adaptive methods are discussed based on the presently existing literature. However, after explicitly reviewed all the methods it is conclude that making a system dynamic and adaptive is a challenging task. Moreover, it also requires other issues need to be taken in to consideration. Such as in [1], [2] the system's mission controller does not have the sufficient rights and the design is not decentralized. Only static models are used in [5]. The system in [6] is not sharing the other resources and so on. These architectures have a lack of standardization. Otherwise, whatever the techniques are studied, offered an efficient and effective scalable and robust performance of the resource management in the distributed system.

Several extensions in the adaptive DRM applications are possible. In many DRM systems fault, delay, execution time is not considered. It is the need of the real time and critical applications. There is scope to work on decentralized systems and security.

REFERENCES

[1] K. Rohloff, Jianming Ye, Joseph Loyall and Richard Schantz, "A Hierarchical Control System for Dynamic Resource Management", In proceeding of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2006) work in progress Symposium, San Jose, CA, 2006.

[2] K. Rohloff and Y. Gabay, "High-Level Dynamic Resource Management for Distributed, Real-Time Embedded Systems", 2006.

[3] Peng Li, Binoy Ravindran and E. Douglas Jensen, "Adaptive Time-Critical Resource Management Using Time/Utility Functions: Past, Present, and Future", Proceedings of COMPSAC workshop, Vol. 2, pp.12-13, 2004.

[4] F. Drews, L. Welch, D. Juedes, D. Fleeman, A. Bruening, K. Ecker and M. Hoefer, "Utility function Based Allocation for Adaptable Applications in Dynamic, Distributed Real-Time Systems", Parallel Distributed Processing Symposium, pp. 123-133, 2004.

[5] F. Li, D. Qi, L. Zhang, X. Zhang, and Z. Zhang, "A Research On Novel Dynamic Resource Management And Job Scheduling In Grid Computing", IEEE Proceedings Of The First International Multi-Symposiums On Computer And Computational Sciences (IMSCCS'06), 2006.

[6] P. Padala, K.G. Shin, X.Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant and K. Salem, "Adaptive Control of virtualized Resource in Utility Computing Environments", ACM EuroSys'07, Mar 2007.

[7] B. Devalla, A. Sahoo, Y. Guan, C. Li, R. Bettati and W. Zhao, "Adaptive Connection Admission Control for Mission Critical Real-Time Communication Networks", MilCom '98.

[8] Nishanth Shankaran., Xenofon Koutsoukos, Douglas C. Schmidt, and Aniruddha Gokhale, "Evaluating adaptive resource management for distributed real-time embedded systems", ACM International Conference Proceeding Series; Vol. 116, Proceedings of the 4th workshop on Reflective and adaptive middleware systems, 2005.

[9] 9. Shweta Sinha and Manish Parashar, "System Sensitive Runtime Management of Adaptive Applications", Conference: International Parallel and Distributed Processing Symposium/International Parallel Processing Symposium – IPDPS (IPPS), 2001.

[10] Yoonhee Kim, Salim Hariri, and Muhamad Djunaedi, "Evaluation of PAMS' Adaptive Management Services", Conference: Heterogeneous Computing, Workshop - Heterogeneous Computing Workshop, 2000.

[11] Huh, Eui-Nam; Welch, Lonnie, "Adaptive resource management for dynamic distributed real-time applications", (16) Publisher: Springer, The Journal of Supercomputing, Volume 38, Number 2, pp. 127-142, November 2006.

[12] Patrick Lardieri, Jaiganesh Balasubramanian, Douglas C. Schmidt, Gautam Thaker, Aniruddha Gokhale and Thomas Damiano, " A Multi-layered Resource Management Framework for Dynamic Resource Management in Enterprise DRE Systems", J. System Software 80(7): 984-996, 2007.

[13] R. Farahbod, U. Glasser and A. Khalili, "Dynamic Resource Management for Adaptive Distributed Information Fusion in Large Volume Surveillance", 2009

[14] Hector A. Duran and Gordon S Blair, "A Resource Management Framework for Adaptive Middleware", Object Oriented Real-time distributed computing (ISORC-2000), Proceeding Third IEEE International Symposium, pp. 206-209, 2002.

[15] Ravi K. Devarasetty, "Heuristic Algorithms for Adaptive Resource Management of Periodic Tasks in Soft Real-Time Distributed Systems", Thesis Virginia Polytechnic Institute and state University, Blacksburg, VA, Feb. 2001.

[16] B. Ravindran and T. Hegazy, "A predictive algorithm for adaptive resource management of periodic tasks in asynchronous real-time distributed systems", Parallel and Ditributed Processing Symposium, 2002.

[17] B. Ravindran, R. Devarasetty and B. Shirazi, "Adaptive Resource Management Algorithm for Periodic Tasks in Dynamic real-time distributed systems", Parallel and Ditributed Computing, Vol. 62, Issue 10,pp. 1527-1547, 2002.

[18] Nishanth Shankaran, Xenofon D. Koutsoukos, Douglas C. Schmidt, Yuan Xue and Chenyang Lu, "Hierarchical control of multiple resources in distributed real-time and embedded systems", Springer Science + Business Media, LLC 2007, Real-Time Systems, Vol. 39, pp. 237–282, 2008.

[19] Ch. Bouras and K. Stamos, "Performance Analysis of Adaptive Admission Control Algorithms for Bandwidth Brokers", Journal of Network and Systems Management, Vol. 15, No. 2, pp. 191-218, June 2007.

[20] B. Dasarathy, S. Gadgil, R. Vaidynathan, K. Parmeswaran, B.Coan, M. Conarty and V. Bhanot, "Network QoS Assurance in a Multi-Layer Adaptive Resource Management Scheme for Mission-Critical Applications using the CORBA Middleware Framework", Proc. 11th IEEE Real Time and Embedded Technology and Applications Symposium, 2005.