International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Resource Management in Cluster Based Network Servers

[1]V. Hema, [2]Dr. K. Kungumaraj
[1]Research Scholar, Mother Teresa Women's University, Kodaikanal, Tamil Nadu, India
[2]Assistant Professor, Department of Computer Science, Mother Teresa Women's University, Kodaikanal, Tamil Nadu, India

ABSTRACT

A cluster consists of a set of loosely or tightly connected computers that work together. Clusters have each node set to perform the same task, controlled and scheduled by software. . Clustering is used for parallel processing , load balancing and fault tolerance. A server cluster is a collection of servers, called nodes that communicate with each other to make a set of services highly available to clients. Each node consists of several resources. Our objective is to deliver better services to high priority request classes without over-sacrificing low priority classes. Thus effective utilization of server resources is necessary. It is often desirable to isolate the performance of different classes of requests. Resource management methods are necessary for ensuring performance isolation while enabling high utilization of the server resources.

Keywords : Clustering, Parallel Processing, Load balancing

I. INTRODUCTION

The number of people using the internet has been growing at a very fast rate. Servers based on clusters of workstations or PCs are the most popular hardware platform used to meet the growth of traffic demands in World Wide Web. A cluster based network server consists of a front-end responsible for request distribution and a number of back end nodes responsible for request processing. A server cluster is a group of independent servers running and working together as a single system to provide high availability of services for clients. Server clusters are designed for applications that have long-running in-memory state or frequently updated data. When a failure occurs on one computer in a cluster, resources are redirected and the workload is redistributed to another computer in the cluster. This paper presents the design and implementation of an architecture for resource management in a server cluster.

II. SERVER CLUSTER ARCHITECTURE

A **Server Cluster** is a collection of **servers**, called nodes that communicate with each other to make a set of services highly available to clients. A network node is an active electronic device that is attached to a network, and is capable of creating, receiving, or transmitting information over a communications channel. It is a connection point that can receive, create, store or send data along distributed network routes.

The Cluster service runs on each node of a server cluster and controls all aspects of server cluster operation. The cluster service manages resources and coordinates with other instances of the Cluster Service in the Cluster. It is made up of a number of dependent interconnected components called managers. . The Cluster service includes multiple software components that work together. These components perform monitoring, maintain consistency, and smoothly transfer resources from one node to another.

The following Figure 1 shows the basic cluster components, 1. Database Manager 2. Check point Manager 3. Communication Manager 4. Node Manager 3. Failover Manager 4. Global update Manager 5. Resource Manager 6. Event log Manager 7. Failover Manager 8. Membership manager.

The Database Manager preserves information about all items in a cluster. These items include cluster node membership, resource groups, resource types, and descriptions of specific resources, such as disks and IP addresses. Database Manager also provides an interface through which other Cluster service components, such as Failover Manager and Node Manager, can store changes in the cluster configuration database. The primary purpose for exposing Database Manager functions is to allow custom resource DLLs to save private properties to the cluster database when this is useful for a particular clustered application.

The Checkpoint Manager handles application-specific configuration data that is stored in the registry on the local server. Checkpoint Manager monitors the data while the application is online. When changes occur, Checkpoint Manager updates the quorum resource with the current configuration data. It also supports resources that have application-specific registry trees (not just individual keys) that exist on the cluster node where the resource comes online. When a cluster resource fails and cannot be restarted automatically, the entire cluster group is taken offline and failed over to another available cluster node.

The Membership manager handles the task of the node joining or forming a cluster. It maintains cluster membership by keeping track of active cluster nodes and nodes that are configured to be the part of the cluster. During a node failure and regroup event, Membership Manager and Node Manager work together to ensure that all functioning nodes agree on which nodes are functioning and which are not.

The cluster network driver is responsible for providing cluster communication, monitoring the status of all network paths between nodes, Routing messages over the optimal path, detecting the failure of nodes. It is a part of the communication path between cluster nodes and helps to maintain cluster membership.

Cluster administrator manages cluster server. It also supports standard cluster objects. A cluster resource is a network application or service defined and managed by the cluster application. Cluster resources are hold together in a cluster within a cluster resource group, or a cluster group. Cluster groups are the units of failover within the cluster. When a cluster resource fails and cannot be restarted automatically, the entire cluster group is taken offline and failed over to another available cluster node.

Resource monitors act as an interface between the cluster service and cluster resources. Each resource is handled by a resource DLL that is loaded into the resource monitor process. Resource DLLs are used to manage cluster resources of a particular type. Each resource DLL is written to manage one or more resource types.

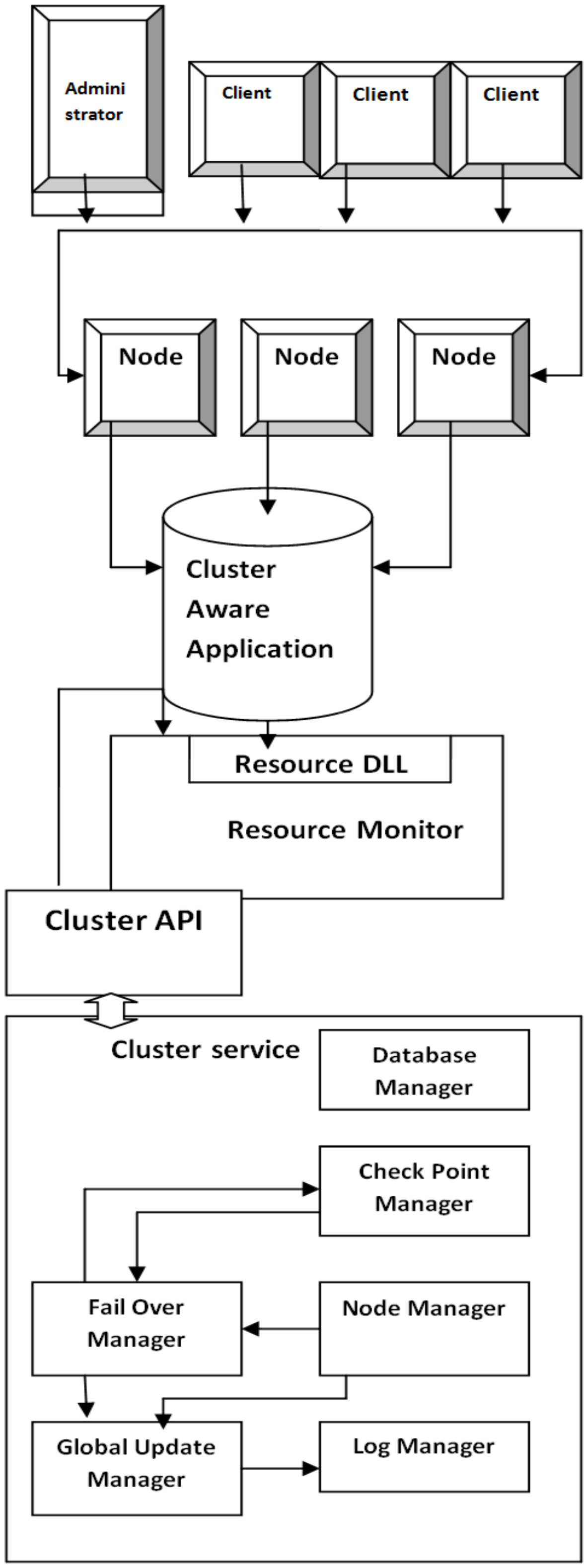

Figure 1 : Server Cluster Architecture

III. BACKGROUND TECHNIQUES

This section presents some technical background on resource management in individual server nodes.

3.1 Cluster Quality Metrics

There are a variety of different metrics that attempt to evaluate the quality of a clustering by capturing the notion of intra-cluster density and inter-cluster sparsity. Letting $G = (V, E)$ be an undirected graph with adjacency matrix A, we use three of the standard cluster quality metrics in our study: modularity, conductance, and coverage. All three are normalized such that scores range from 0 to 1, and 1 is the optimal score.

3.1.1. Modularity

The modularity of a graph compares the presence of each intra-cluster edge of the graph with the probability that that edge would exist in a random graph . Although modularity has been shown to have a resolution limit , some of the most popular clustering algorithms use it as an objective function . Modularity is given by Eq (1),

$$\sum k(ekk-a2k) \quad (1)$$

where e_{kk}, the probability of intra-cluster edges in cluster S_k, and a_k, the probability of either an intra-cluster edge in cluster S_k or of an inter-cluster edge incident on cluster S_k, are

$e_{kk} = |\{(i, j): i \in S_k, j \in S_k, (i, j) \in E\}|/|E|$,

$a_k = |\{(i, j): i \in S_k, (i, j) \in E\}|/|E|$

and where $S_k \subseteq V$.

3.1.2.Conductance

We define the conductance of a cluster by the number of inter-cluster edges for the cluster divided by either the number edges with an endpoint in the cluster or the number of edges that do not have an endpoint in the cluster, whichever is smaller. The conductance for a cluster is given by Eq (2),

$$Sk=\sum i\in Sk,j\notin SkAijmin\{A(Sk),A(Sk)$$

where $S_k \subset V$ and $A(S_k) = \sum_{i\in Sk} \sum_{j\in V} A_{ij} - \sum_{i\in Sk} \sum_{j\in Sk} A_{ij}$, the number of edges with an endpoint in S_k.

We define the conductance of a graph G to be the average of the conductance for each cluster in the graph, subtracted from 1. The conductance for a graph falls in the range 0 to 1, and the subtraction makes 1 the optimal score. The conductance for a graph is given by Eq (3),

$$(G)=1-1k\sum k(Sk) \quad (3)$$

3.1.3.Coverage

Coverage compares the fraction of intra-cluster edges in the graph to the total number of edges in the graph. Coverage is given by Eq (4),

$$\sum i,jAij\delta(Si,Sj)\sum i,jAij \quad (4)$$

where S_i is the cluster to which node i is assigned and $\delta(a, b)$ is 1 if $a = b$ and 0 otherwise. Coverage falls in the range 0 to 1, and 1 is the optimal score.

3.2 Resource Containers

A resource container is an abstract operating system entity that logically contains all the system resources being used by an application to achieve a particular independent activity. With resource containers, the binding between a thread and a resource principal is dynamic, and under the explicit control of the application; we call this the thread's *resource binding*. Resource containers allow accurate accounting and scheduling of resources consumed on behalf of a single client requests and enable performance isolation and differentiated quality of service when combined with an appropriate resource scheduler.

3.3 Cluster Reserves

Cluster reserves are cluster-wide resource principals obtained by logically combining the resource principals on individual cluster nodes. Resources can then be allocated to this cluster-wide resource principal and get translated into allocations for the resource principals on the individual cluster nodes. Performance isolation on individual cluster nodes is extended into performance isolation for the cluster.

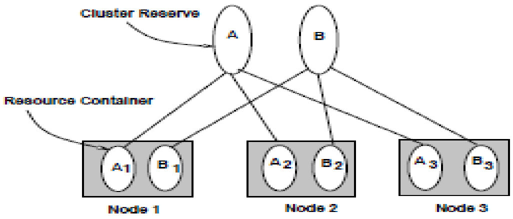

Figure 2 : Cluster Reserves

Figure 2 shows the hierarchical relationship between cluster reserves and resource containers. It has three nodes and two cluster reserves A and B. The portioning of resources is determined by a cluster resource manager. To compute the partitioning, the cluster resource manager collects resource usage statistics from the cluster nodes and maps the allocation problem to an equivalent constrained optimization problem.

The following steps show the mapping of cluster resource management problem to a constrained optimization problem.
Step 1 : Choose the matrix R such that the cluster-wide allocation deviates the least from its desired allocation.
Step 2 : A desirable solution matrix R is such that it deviates minimally from the reported resource usage values.
Step 3 : Some service classes do not add up to their desired cluster-wide allocation. The unassigned resources are allocated to such service classes to bring up their aggregate cluster-wide allocation to desired allocation level. Any unused resources not used are proportioned amongst other service classes dynamically by the resource container mechanism.

3.4 Centralized Clustering Algorithm

In network clustering, the literature defines "similarity" based on topology. Clustering algorithms seek to capture the instinctive notion that nodes should be connected to many nodes in the same community, but connected to few nodes in other communities.

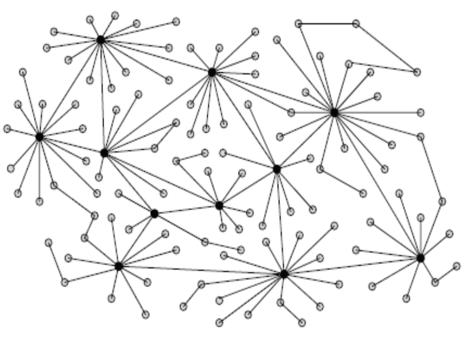

Figure 3. Scale free network

The majority of the nodes are directly connected with the nodes of higher degree. The core nodes presume management functions and the other nodes presume the task of computational and storage servers. Most of the server nodes are at distance one, two, or three from a core node which could gather more accurate state information from these nodes and with minimal overhead.

The goal of the algorithm is to identify: (a) The service nodes connected to a core node; (b) The network connecting the core nodes. The algorithm uses two types of messages: (1) *Type 1 - cluster initiation*, messages sent by a core node to all its neighbors; *Type 2 - request to join a cluster*, message sent by a service node to the core node at the shortest distance.

IV. CONCLUSION AND FUTURE WORK

Client request rates for Internet services tend to be burst and thus it is important to maintain efficient resource utilization under a wide range of load conditions. A variety of mechanisms for resource management policies facilitate the coordination of multiple autonomic controllers, support the applications of control theoretical principles for resource management , support real-time applications and support the strategy based on random walks for selecting nodes with desirable properties. Cluster reserves afford performance isolation among multiple service classes, which can be defined based on the requested content, the client who issues the request or both. It can be extended to provide performance isolation for operating system resources like memory , secondary storage like hard disks, network throughput, battery power and external devices.

V. REFERENCES

[1]. Cluster reserves: a mechanism for resource management in cluster-based network servers - Mohit Aron Peter Druschel Department of Computer Science Rice University - 2000
[2]. Clustering Algorithms for Scale-free Networks and Applications to Cloud Resource Management -Ashkan Paya and Dan C. Marinescu Computer Science Division Department of Electrical Engineering and Computer Science University of Central Florida, Orlando, FL 32816, USA - May 2013

[3]. How a Cluster Server works - https://technet.microsoft.com/en-in/library.
[4]. C. Yang and M. Luo, "Realizing Fault Resilience in Web-Server Clusters", Supercomputing 2000: the 13th ACM/IEEE Conference on High Performance Networking and Computing (SC2000), Dallas, TX, 2000.
[5]. Casalicchio and Colajanni, "A Client-Aware Dispatching Algorithm for Web Clusters Providing Multiple Services", Proceedings of WWW10, May 2001, Hong Kong.
[6]. T. Schroeder, S. Goddard, and Bno Ramamurthy. Scalable web server clustering technologies. IEEE Network, 2000