Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.623 – 627

RESEARCH ARTICLE

A Novel Resource Distributed Discovery and Management in Grid Computing

M. Brinda Kumar[1], Dr. K.P. Kaliyamurthie[2]
[1]Department of Information Technology, Bharath University, India
[2]Department of Information Technology, Bharath University, India

Abstract— In grid-computing environment the computer resources are shared under the grid nodes. Resource discovery is an important process for finding suitable nodes that satisfy application requirements in grid environment. In most of the existing resource discovery mechanisms rely mainly on recent observed resource capacities of individual nodes to make their deployment decision based on current status of the nodes have severe limitations to achieve scalability because of the presence of internodes dynamism in addition to the internodes heterogeneity. Individual nodes have widely varying resource capabilities due to varying loads, network connectivity, churn, or user behavior. Besides internodes heterogeneity, many of these systems also show a high degree of internodes dynamism, so that selecting nodes based only on their recently observed resource capacities can lead to poor deployment decisions resulting in application failures or migration overheads.

Key Terms: - Grid computing; Resource discovery

I. INTRODUCTION

Grid computing is a term referring to the combination of computer resources from multiple administrative domains to reach common goal. Hardware and software resources are shared within the grid nodes. It provide mechanisms for sharing and accessing large and heterogeneous collections of remote resources such as computers, online instruments, storage space, data, and applications. Resources are identified based on a set of desired attributes. Resource attributes have various degrees of dynamism, from mostly static attributes, like operating system version, to highly dynamic ones, like network bandwidth or CPU load.

Resource management systems use a system model to describe resources and a centralized scheduler to control their allocation. It does not adapt well to grid systems, to support high throughput computing. Obstacles include heterogeneity of resources, which make uniform allocation algorithms difficult to formulate, and distributed ownership, leading to widely varying allocation policies. By these problems, it developed and implemented the classified advertisement matchmaking framework, and general approach to resource management in grid environment with decentralized ownership of resources.

Efficient resource discovery based on dynamic attributes such as CPU utilization and available bandwidth is a crucial problem in the deployment of computing grids. Existing solutions are either centralized or unable to answer advanced resource queries (e.g., range queries) efficiently. Aggregation particularly hierarchical aggregation is a common technique employed in large distributed systems for the scalable dissemination of information. Aggregation essentially compresses the amount of transmitted data in the system while preserving the overall information content. In the context of resource discovery, this would correspond to a suitable "compression" of the node resource usage patterns to achieve a desirable trade-off between the quality of resource discovery and the overhead of network data transmission in the system.

The goal is to achieve the same quality of resource discovery as a global resource discovery system with full historical node-behavioral knowledge, but to significantly compress the amount of necessary node-behavioral representation data in the system in order to achieve scalability. Such an aggregation of node resource usage distributions for a group of nodes can be used to represent: An accurate approximation of any individual node's resource usage. This is important for the discovery of desirable nodes based on a resource requirement. The collective resource usage behavior of the group of nodes.

This can provide information about load patterns or resource usage behavior for a set of related nodes (e.g., those in the same geographical area or running the same application).The overall capacity of the group. This can be used to track the overall resource usage, e.g., for audit or accounting purposes.

To account for the heterogeneity of nodes, it defines the notion of a resource bundle: an aggregation of a group of nodes with similar resource capacity distributions. By combining only similar nodes together, such an aggregation process will preserve the individual node distributions more accurately. Fig. 3 shows a high-level view of the notion of resource bundles and how they might be constructed. First, a group of nodes are bundled based on the similarity of their resource capacity histograms. Second, each bundle produces a representative distribution that can be used to characterize the whole bundle. The question is how can we identify such groups of similar nodes to construct a resource bundle, and how can we compute a representative to accurately represent the nodes in the bundle? Resource bundles and how they might be constructed.

First, a group of nodes are bundled based on the similarity of their resource capacity histograms. Second, each bundle produces a representative distribution that can be used to characterize the whole bundle. The question is how we can identify such groups of similar nodes to construct a resource bundle, and how can us compute a representative to accurately represent the nodes in the bundle. To identify nodes with similar distributions, we propose the use of clustering algorithms that have traditionally been used in data mining applications to group together statistically similar data elements. However, a clustering algorithm must meet several requirements in this context:

1. The data to be clustered (i.e., the resource histograms) are not single-point, but multi element (consisting of multiple histogram bins). The clustering algorithm must be able to handle such multi element data.
2. The node resource usage histograms could represent arbitrary distributions, and cannot be assumed to conform to standard distributions (e.g., Gaussian, uniform, etc.). The clustering algorithm must be distribution-free, i.e., it must not assume the existence of a standard distribution or certain parameters.
3. The clustering algorithm must not only identify the closely related set of nodes, but it is desirable if it can also produce a compact representation of the collective resource usage of these nodes. Such a representation can be used to easily characterize the nodes in a bundle (e.g., high capacity/low capacity, etc).

II. SYSTEM ANALYSIS

2.1 Statistical Resource Requirements:

During the resource discovery process, applications typically seek nodes meeting certain resource requirements, e.g., minimum CPU spare capacity of 1 GHz, memory capacity of 512 MB, etc. Such resource requirements can be expressed as a tuple {R,c}, where R is a resource type, and c is the desired minimum capacity of the resource.

However, the resource capacities of nodes in loosely coupled distributed systems often vary a lot over time, which could result in application performance degradation, failures, or need for frequent migrations resulting in large overhead. It would be desirable to provide statistical resource guarantees so that applications can be deployed on nodes that are likely to satisfy the minimum desired requirement for a certain period of time

2.2 Resource Usage Representation:

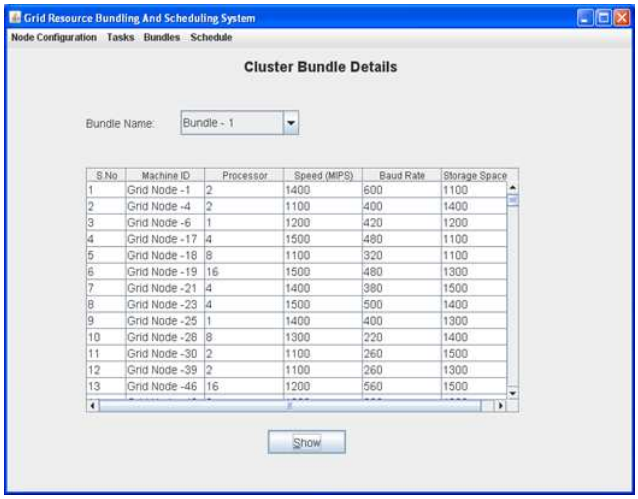
Since different applications can specify different values of c, p, and t, maintaining only a few values such as the capacity corresponding to a fixed percentile (e.g., 95th percentile) of resource usage on each node, or the percentile corresponding to a fixed capacity level (e.g., 1 GHz) may not provide enough information to satisfy different resource discovery queries. Similarly, it may need to capture the resource usage behavior over different time durations (such as an hour, day, week, etc.) to incorporate requirements over different time granularities and capture long-term versus short-term trends.

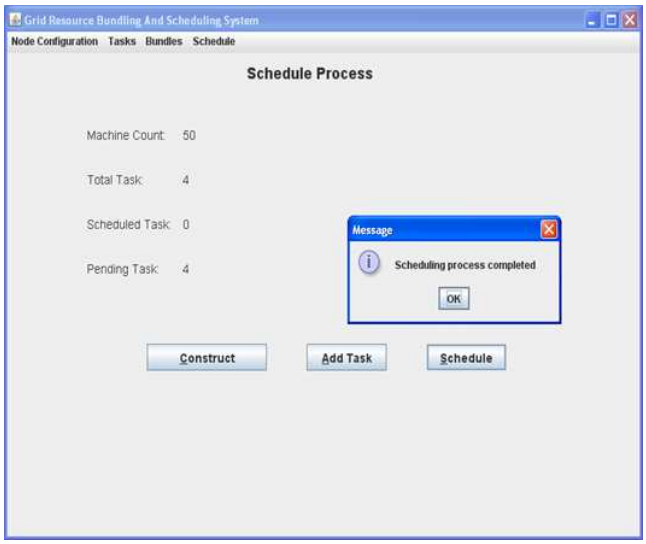
III. PROPOSED SYSTEM

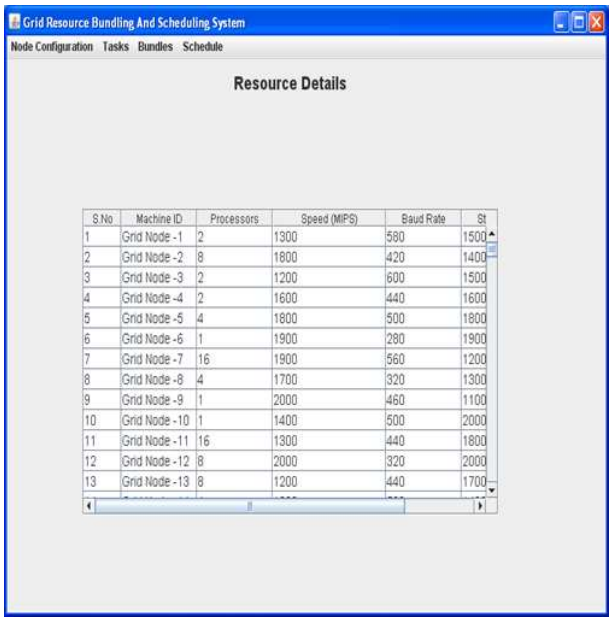
The grid resource discovery and management scheme is designed with the resource bundling mechanism. The resource information are collected from the grid nodes and maintained under the scheduler. The resource information is frequently updated under the resource management process. The resource update overhead is high is large-scale grid environment. The resource bundling scheme is used to group up the resource information. The resources details are collected and relevant resource information are grouped as resource bundles. The resource bundle information is used to schedule resources for the tasks.

The proposed system is designed to allocate resources using the resource bundle information. The resource bundles are prepared in two ways. They are statistical resource bundling model and cluster based resource bundling model. The statistical resource bundling model uses the aggregation function for the bundling process.

The cluster based resource bundling scheme is used to group the grid resources with respect to the similarity values.







S.No	Machine ID	Processors	Speed (MIPS)	Baud Rate	St
1	Grid Node -1	2	1300	580	1500
2	Grid Node -2	8	1800	420	1400
3	Grid Node -3	2	1200	600	1500
4	Grid Node -4	2	1600	440	1600
5	Grid Node -5	4	1800	500	1800
6	Grid Node -6	1	1900	280	1900
7	Grid Node -7	16	1900	560	1200
8	Grid Node -8	4	1700	320	1300
9	Grid Node -9	1	2000	460	1100
10	Grid Node -10	1	1400	500	2000
11	Grid Node -11	16	1300	440	1800
12	Grid Node -12	8	2000	320	2000
13	Grid Node -13	8	1200	440	1700

IV. CONCLUSION AND FUTURE ENHANCEMENT

The presence of node-level dynamism means that selecting nodes based only on recently observed capacities can lead to poor deployments resulting in application failures or migrations. However, existing resource discovery techniques rely only on recent observations to achieve scalability.

It proposes a resource bundle which used as a representative for resource usage distribution in a group of nodes by using similar resource usage patterns. Two efficient techniques used in a Resource Bundles to capture the long-term resource usage behavior of a set of nodes: Resource usage histograms and clustering-based resource aggregation to achieve compact representation of a set of similarly behavior and scalability.

It showed that resource bundles are able to provide high accuracy for statistical resource discovery, while achieving high scalability and also showed that resource bundles are ideally suited for identifying group-level characteristics such as finding load hot spots and estimating total group capacity.

The resource bundle based grid resource discovery scheme is designed to fetch resource in a large-scale grid environment. The system manages the scalability factor. Statistical bundling techniques are replaced with cluster-based scheme. Processor, bandwidth and storage space are shared under the grid environment. Resource monitoring process is simplified with the bundle-based technique.

The scheduling scheme distributes the load to all the bundles. The system can be enhanced with the following features.

1. The resource management scheme can be improved with security features.
2. The resource discovery and scheduling scheme can be enhanced to manage resources under cloud computing environment.
3. The resource bundling scheme can be improved to handle cost based resource scheduling process.
4. Failure management and task migration features can be added with the bundle based resource discovery and management scheme.
5. The resource bundling scheme can be adapted to the data grid environment to manage storage data and spaces.
6. The resource bundle scheme can be adapted to other real life problems such as inventory management and supply chain management.

REFERENCES

[1] D. Anderson, "BOINC: A System for Public-Resource Computing and Storage," Proc. IEEE/ACM Int'l Workshop Grid Computing (GRID), (2004).

[2] V. Lo, D. Zappala, D. Zhou, Y. Liu, and S. Zhao, "Cluster Computing on the Fly: P2P Scheduling of Idle Cycles in the Internet," Proc. IEEE Fourth Int'l Conf. Peer-to-Peer Systems, (2004).

[3] Grid2: Blueprint for a New Computing Infrastructure, I. Foster and C. Kesselman, eds. M. Kauffman,

(2004).
[4] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S.Shenker, "Making Gnutella Like P2P Systems Scalable," Proceedings.ACM SIGCOMM, (August 2003).
[5] B. Cohen, "Incentives Build Robustness in Bittorrent," Proc. First Workshop the Economics of P2P Systems, (June 2003).
[6] S. Guha, N. Daswani, and R. Jain, "An Experimental Study of the Skype Peer-to-Peer Voip System," Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS), (2006).
[7] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M.Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," ACM SIGCOMM Computer Comm.Rev., vol. 33, no. 3, pp. 3-12, (July 2003).
[8] A. Iamnitchi and I. Foster, "On Fully Decentralized Resource Discovery in Grid Environments," Proc. IEEE/ACM Int'l Workshop Grid Computing (GRID), (2001).
[9] D. Oppenheimer, J. Albrecht, D. Patterson, and A. Vahdat, "Distributed Resource Discovery on PlanetLab with SWORD,"Proc. Workshop Real, Large Distributed Systems (WORLDS '04), (December 2004).
[10] R. Raman, M. Livny, and M. Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing," Proc. Int'l Symp. High Performance Distributed Computing (HPDC '98), (July 1998).
[11] D. Oppenheimer, B. Chun, D. Patterson, A.C. Snoeren, and A. Vahdat, "Service Placement in a Shared Wide Area Platform," Proc. USENIX Ann. Technical Conf., (June 2006).
[12] P. Yalagandula and M. Dahlin, "A Scalable Distributed Information Management System," Proc. ACM SIGCOMM, (2004).
[13] J.M. Schopf, "A Practical Methodology for Defining Histograms for Predictions and Scheduling," NU technical report, (1999).
[14] B. Urgaonkar, P. Shenoy, and T. Roscoe, "Resource Overbooking and Application Profiling in Shared Hosting Platforms," Proc. Symp. Operating Systems Design and Implementation (OSDI '02), (December 2002).
[15] R.V. Renesse, K.P. Birman, and W.Vogels, "Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining," ACM Trans. Computer Systems, volume 21, no. 2, pp. 164-206, (2003).
[16] J. Mickens and B. Noble, "Exploiting Availability Prediction in Distributed Systems," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '06), (May 2006).

