RESEARCH ARTICLE OPEN ACCESS

Dynamic Resource Recovery in Grid

Malika.K.S[1], MRS.V.Elavarasi (Asst Professor) [2]
SRM University, Ramapuram
Chennai.

-------------------------------------*************************-----------------------------------

Abstract:

Resource discovery is a fundamental problem ingrid systems. A P2P network is a distributed system with the attributes of dynamicity and scalability. P2P systems have the same goal as grid systems: to share and exchange various resources. With their development, P2P systems and grid systems can be combined into a new system which has their attributes. In this paper, a novel hierarchical P2P based grid model based on the existing grid resource discovery models is discussed.

A way to provide Grid scalability is to adopt Peer-to-Peer (P2P) models to implement nonhierarchical decentralized Grid services and systems. A core Grid functionality that can be effectively redesigned using the P2P approach is *resource discovery*. This paper proposes a P2P resource discovery architecture aiming to manage various Grid resources and complex queries. Its goal is two-fold: to address discovery of multiple resources, and to support discovery of dynamic resources and queries in Grids.

Keywords: - **Resource discovery, peer to peer, Grid.**

-------------------------------------*************************-----------------------------------

1. INTRODUCTION

Grid and P2P are the two important resource sharing systems at present. Grid systems are applied in some large scientific computing centers in order to make full use of the shared resources, such as computer CPUs, storage, data, software and instruments. Grid systems are usually organized centrally or hierarchically in order to administrate the participating resources effectively.

Although, P2P and grid have many common characteristics such as dynamic behavior and heterogeneity of the involved resources, P2P and grid systems exhibit essential differences. Grids employ relative concentration and distributed resource administrative strategy in order to allow access to all of the resources shared on the network. Grids are trusted in the heterogeneous environment, as a special resource sharing model, users have high security, confidentiality and reliability, but do not solve the issue that nodes randomly join and live. While P2P has a flexible topology, nodes have high degree of autonomy, which leads to frequently joining and leaving a P2P network and shows a high degree of dynamicity, but there is no assurance of QoS and security between the nodes.

Query resolution in these systems can be abstracted as the problem of allowing one peer to obtain a local view of global information defined on all peers of a P2P unstructured network. In particular, we assume that each peer holds a piece of information (the aggregate resource statuses of all nodes in its administrative domain) and that any peer requires to access the values of the data of all other peers periodically at rate _ queries=sec. The goals to be achieved are threefold: first, one wants to guarantee that every node is likely to collect the complete global information in a timely fashion. Moreover, the communication over-head must be kept as limited as possible to avoid congesting the network. Finally, the processing power of each node must be used parsimoniously.

2. RELATED WORK

The problem of data gathering in distributed systems has been faced with many different tools and approaches.

First classes of techniques are those based on probabilistic gossiping. Probabilistic gossiping has been used both to compute a function of the global information, e.g., aver-ages, and to actually spread local information across a net-work as in our settings although such techniques rely on a set of assumptions that are difficult to guarantee in practice. Notable attempts to overcome some of these limitations in the area of epidemic dissemination are that result in close to optimum latency-bandwidth trade-of. In particular, uses flow control on the maximum rate at which a participant can submit updates without creating a backlog and devises content reconciliation mechanisms to reduce message redundancy. In exploitation/ enforcement of topological properties of the network are proposed to improve the performance of the data dissemination process.

Gnutella-based dynamic query strategy is used to reduce the number of messages generated by flooding. Instead of all directions, this strategy forwards the query only to a selected peer. If a response is not returned from a direction,another round of search is initiated in the next direction, after an estimated time. For relatively popular contents this strategy significantly reduces the number of messages without increasing the response time. Broadcast in DHT-based P2P networks adds broadcast service to a class of DHT systems that have logarithmic performance bounds. In a network of N nodes, the node that starts the broadcast reaches all other nodes with exactly $N - 1$ messages (i. e., no redundant messages are generated). The approach proposed for dynamic resource discovery in this paper is inspired by both the dynamic query strategy and the broadcast approach mentioned above. It uses a DHT for broadcasting queries to all nodes without redundant messages, and adopts a similar “incremental” approach of dynamic query. This approach reduces the number of exchanged messages and response time, which ensures scalability in large-scale Grids.

3. SYSTEM DESCRIPTION

In this paper we model the interface peers of a Grid system and the connections among them as a graph GðV; EÞ, where V and E are the set of interface peers and edges connecting them, respectively. Each node of the network is uniquely identified by an identifier ID. The ID can be assigned by a fixed rendezvous node, e.g., a tracker, or can be represented by the IP, port address of the node. Each node v_j 2 V owns an m-bits information $x^{t_j}_{v_j}$, where t_j is a time-stamp or an integer that is incremented each time the information in v_j changes. To simplify the notation in the rest of the paper we assume that v_j coincides with the ID of node; t_j is usually referred to as the generation number. In our settings a node can update its information asynchronously with respect to the rest of the network, increasing the generation number associated with the information.

The goal of nodes is to communicate with one another the respective information, so as to realize a concurrent broadcasting of all the information collected by all the nodes in the network. This must be done indefinitely often at an arbitrary rate _ by each node. This observation rules out any centralized solution where all nodes report to a com- mon monitoring node that in turn must propagate the collected data to all the participants. This approach is clearly infeasible because it imposes a huge amount of traffic to and from the monitoring node, not to mention the issues related to the election and vulnerability of a centralized sink.

Therefore, in this paper we propose a fully distributed solution based on random walks. Each node is allowed to start a limited number w of packets that are the random walkers propagating the information in the network. The parameter w clearly allows one to control the amount of traffic injected in the network. On every reception by a node, the packet is forwarded to a random neighbor thus realizing a simple form of probabilistic gossiping. It is well known that network coding solutions, e.g., carrying linear combinations of the collected information, increases the performance in terms of throughput, robustness and persistence. On the other hand, coding approaches exhibit two main shortcomings. The first and most studied issue is represented by the added computational complexity. A possible solution that has already been proposed in the literature is to simplify the original random network coding approach that requires one to combine the data blocks in high order Galois Field, with systems based on simple binary combinations, e.g., XOR. Our work copes with the complexity issue using a simple class of rateless codes, known as Luby Transform codes. The second most relevant shortcoming of NC is represented the impossibility for a node to update asynchronously the informationit combines without catastrophically impacting on the decoding capability of all the other nodes.Indeed, the nodes keep collecting linear combinations of a set of unknowns until they successfully invert the corresponding system of equations.

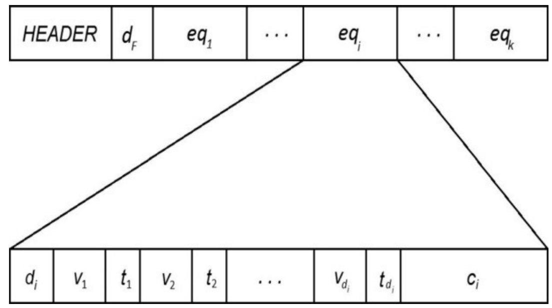

Fig. 1.Packet structure.

Clearly, the system of linear equations is meaningful if one keeps combining the same information. On the contrary, in this paper we propose a novel decoding approach for LT codes that is resilient to asynchronous changes of the information. In conclusion, we let each node propagate a fixed number of packets carrying coded information of the nodes that the packets have hit performing a random walk along GðV ; EÞ. All the nodes use the received packets to solve a system of linear equations allowing them to retrieve the data associated with all the information collected by the network in a timely, complete and robust way.

4. RESOURCES& QUERY TYPES

In Grids, resources belong to different resource classes. A resource class is a “model” for representing resources of the same type. Each resource class is de- fined by a set of attributes which specify its characteristics. A resource is an “instance” of a resource class. Each resource has a specific value for each attribute defined by the corresponding resource class. Resources are univocally identified by URLs. An example of resource class is “computing resource” that defines the common characteristics of computing resources. These characteristics are described by attributes such as “OS name”, “CPU speed”, and "Free memory". An instance of the "computing resource" class has a specific value for each attribute, for example, "OS name = Linux", "CPU speed = 1000MHz", and "Free memory = 1024MB". Table 1 lists some examples of Grid resources classes. A more complete list of resource classes can be found.

Resource classes can be broadly classified into intra-node and inter-node resources. "Computing resource" is an example of intra-node resource class. Anexample of inter-node resource class is "network connection", which defines network resource characteristics. Fig. 1 shows a simple Grid including four nodes and three resource classes. As examples of inter-node resources, NodeA includes two instances of resource class *a*and one instance of resource class b. The figure also shows two inter-node resources: one between NodeA and NodeD, and the other between NodeB and NodeD.

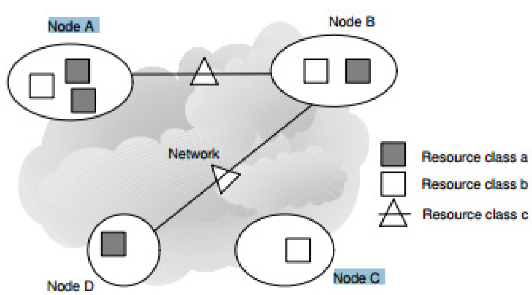

Fig. 2.Inter-node and intra-node resources.

The attributes of each resource class are either static or dynamic. Static attributes refer to resource characteristics that do not change frequently, such as "OS name" and "CPU speed" of a computing resource. Dynamic attributes are associated to fast changing characteristics, such as "CPU load" and "Free memory". The goal of resource discovery in Grids is to locate resources that satisfy a given set of requirements on their attribute values. Three types of queries apply to each attribute involved in resource discovery: – Exact match query, where attribute values of numeric, Boolean, or string types are searched. – Range query, where a range of numeric or string values is searched. – Arbitrary query, where for instance partial phrase match or semantic search is carried out. A multi-attribute query is composed of a set of sub-queries on single attributes. Each sub-query fits in one of the three types as listed above, and the involved attributes are either static or dynamic. Complex Grid applications involve multiple resources. Thus, multi-resource queries are often needed. For instance, one can be interested in discovering two computing resources and one storage resource; these resources may not be geographically close to each other. A multi-resource query, in fact, involves a set of sub-queries on individual resources, where each sub-query can be a multiattributequery.Taking into consideration both characteristics and query requirements of Grid resources, the appropriate
P2P discovery techniques are listed in Table 2.

Table 2. Query techniques used for different types of resources and queries.

	Static Grid resources	Dynamic Grid resources
Exact query	DHT	Flooding
Range query	DHT	Flooding
Arbitrary query	Flooding	Flooding

As shown in the table, DHTs are used only for exact and range queries on static Grid resources. For dynamic resources and arbitrary queries, classical DHT-based approaches are not suitable. These approaches were not originally designed for resource discovery queries of arbitrary expression forms. Moreover, fast-changing resources, such as CPU load, require frequent updates on DHTs, and thus cause prohibitive maintenance costs. Flooding approaches are used for both dynamic Grid resources and arbitrary queries on static resources. This is because flooding does not require table updates and maintenance. Nevertheless, the massive amount of messages they generate do not scale with network sizes.

4. P2P –BASED GRID RESOURCE DISCOVERY:

In a grid system, an agent of the grid can be employed as a peer to form a P2P system. When a user launches a grid resource query message via the grid agent, it will be transmitted in the P2P network. As the resource providers update their resource states, resource update messages will also be spread in the P2P network.

For the inter-grid resource discovery problem, we can set one or more servers for each grid system, to provide storage and provision of information on available resources. These servers can also be employed as the network nodes to form a P2P virtual layer. When a client user initiates a resource query request, the request will firstly be carried out to find the corresponding resource in the local grid. If the resource can be found, the result will be returned directly to the user. When the local delay checking is timed out, the server will launch inquiries to other grids, namely using the P2P virtual layer between the grids for resource queries. When a node receives a query request from other nodes, firstly it checks whether there is matching resources in its internal grid resources. If the corresponding resources can be found, the query result will be sent back to the initiator directly. Otherwise, it forwards the query request to other grids.

In general, as discussed above, resource discovery may happen among communities or inside a community. One community can be regarded as one VO, and such VOs are independent to each other. With the development of the global grids, the number of the communities will increase, and the scalability of such system comprised by communities is important. Obviously, P2P network is the prime choice to organize those communities.

When such a PC grid joins an existing grid, the *IS* root node involves in the resource discovery in the existing grid. The resource discovery can be described as follows. When a task request arrives, the *IS* root node involves resource discovery with other grid nodes in the same grid, and the resource discovery strategy should be consistent with existing strategies, such as the Globus and MDS (Monitoring and Discovery Service) resource discovery strategies. When they find there are available resources at the *IS* root node, the request will be forwarded to the *IS* root node. The *IS* root node will decompose the query task based on the information provided by the super nodes in the underlying P2P network and then the decomposed operations will be sent to a super node. The super node initiates resource discovery and the task distribution, and the specific executions of the subtasks are completed at the underlying PC resources.

The corresponding pseudocode is as follows.

```
//When an IS root node receives a request from a peer or other IS root node.
if ( local.find() != NULL && TTL <Max_hops)
{
  tast.decompose();
  for (i =0 ;i<Max_supper_nodes;i++)
  {
    request.send();
    ttl.set();
  }
}
else
{
  exit;
}
//When a supper node of the secend layer receives a query task.
```

```
if (local.find() != NULL && TTL <Max_hops)
 {
   search.initiate();
   for (i =0 ;i<Max_PC_Domain;i++)
   {
     jobdistribut
     e();
     ttl_set();
   }
 }
else
{
   sendmessage("Not
   find"); exit;
}

//When a PC of the third layer receives a
subtask. If (local.find() != NULL && TTL
<Max_hops)
{
    search();
}
else
{
   sendmessage("Not
   find"); exit;
}
```

The overlay network and the group do not originally exist. In order to effectively organize the *IS* nodes, we need the support of the overlay network construction method. In the grid environment, resource nodes are dynamic. In order to ensure the network connectivity, packet services, the availability and the correctness of the discovery, we also need the support of the overlay network maintenance methods. The following is to discuss of the organizations of the two self-organized P2P layers in the novel model.

5. CONCLUSIONS & FURTHER WORK

In this paper we have shown that the recent advances in rateless coding and decoding can be profitably exploited to achieve a robust and timely P2P resource location technique in Grid systems. The major novelty of the pro-posed approach lies in the use of network coding principles in a scenario where local data can be updated asynchronously. Moreover, as opposed to some forms of distributed storage proposed in the literature, our proposal realizes a continuous update of the global information across the whole distributed system, while keeping the amount of traffic under control.
From the algorithmic point of view, the major contribution is represented by the design of a novel decoder for rate-less codes that is robust to asynchronous updates of the information. Another interesting result that we achieved is the development of a simple analytical model for the estimation of the time required to spread the information as a function of the network and information sizes, given a constraint on the MTU allowed by the available transmission protocol. Such a model can be exploited for the estimation of the performance and for the selection of some important parameters of the system. The analytical results show that the proposed coded approach reduces the time required to communicate all the information with respect to an analo-gous system without coding. Furthermore we prove that such gain increases with the size of the information to be spread, or analogously when the MTU shall be very limited. Another paramount result is that the encoded system scales better than the uncoded one when the number of nodes in the distributed system increases.
Finally, the designed simulator shows that the system is very efficient in many different scenarios characterized by network dynamics and information updates that cannot be analyzed with the analytical approach.

6. REFERENCES

[1] P. Trunfio, D. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi, “Peer-to-Peer Resource Discovery in Grids: Models and Systems,” Future Generation Computer Systems, vol. 23, no. 7, pp. 864-878, 2007.

[2] A. Kermarrec, L. Massouli_e, and A.

Ganesh, "Probabilistic Reliable Dissemination in Large-Scale Systems," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 3, pp. 248-258, Mar. 2003.

[3] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-Based Aggre-gation in Large Dynamic Networks," ACM Trans. Computer Systems, vol. 23, no. 3, pp. 219-252, Aug. 2005.

[4] M. Ferreira, J. Leitao, and L. Rodrigues, "Thicket: A Protocol for Building and Maintaining Multiple Trees in a P2P Overlay," Proc. 29th IEEE Symp. Reliable Distributed Systems, pp. 293-302, 2010.

[5] R. Gaeta, M. Grangetto, and M. Sereno, "Local Access to Sparse and Large Global Information in P2P Networks: A Case for Compressive Sensing," Proc. IEEE 10th Int'l Conf. Peer-to-Peer Computing (P2P), pp. 1-10, Aug. 2010.