Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 937-940
doi:10.4028/www.scientific.net/AEF.6-7.937

A Dynamic Resource Management Scheme on P2P-based Grid Systems

Yunsong Tan[1 2]

[1] Hubei Province Key Laboratory of Intelligent Robot, Wuhan, 430073, China

[2] School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430073

yunsongtan@126.com

Keywords: P2P; Grid; Resource Management ; DHT

Abstract. Efficient multi-dimensional data indexing mechanism is one of the fundamental requirement for Grid computing systems. Peer-to-peer organization of Grid resource discovery approach would have several desirable features, including high scalability, high reliability, self-organization and self-healing. Most of the research in this area only considers static resource.In this paper, we describe a tree vector indexing scheme for Grid systems, in order to support range queries and at the same time preserve the important attributes of grid systems.

Introduction

Distributed systems are becoming more and more popular nowadays, drawing the attention of current research in networking. P2P systems are a subset of distributed systems and they have gained popularity during the last years as they allowed users to share files (primarily music files without paying for them). They are independent of centralized servers (most of them), they offer unique capabilities and much of today's research on P2P systems is about how certain aspects of their functionality can be improved, e.g. data storing, searching, flexibility, etc. Such systems can be used in the implementation of many kinds of applications, like multicast systems, anonymous communications systems, web caches, database management systems and many others systems that can operate in a distributed environment. We can broadly classify existing P2P systems, based on their ability to store and retrieve information, as unstructured, hybrid, or structured.

We may consider a set of resources on a typical Grid system organized as peers on a P2P network. Each resource has some attributes whose values identify its characteristics: CPU speed, free disk space, available memory and so on. Users query the system to locate resources satisfying some criteria, such as:(CPU Speed 1GHz)and (OS Type = "UNIX")and (Free Space 500MB)to denote all computational resources with a CPU speed of at least 1GHz, running the UNIX operating system, and with available disk space in the range [200; 500] MB. In this example, some attribute values (e.g., the amount of free disk space) may change over time.

Traditionally, Distributed Hash Table (DHT) has been efficient for exact queries such as finding all resources that match the given attribute value. Extending DHTs to support multi-dimensional range query, to index all resources whose attribute value overlap a given search space, is a complex problem. Multi-dimensional range query is based on ranges of values for attributes rather than on specific values. Compared to 1-dimensional queries, resolving multi-dimensional range query is far more complicated, as there is no obvious total ordering of the points in the attribute space. Further, the query interval has varying size, aspect ratio and position such as a window query. The main challenges involved in enabling multi-dimensional range query in a DHT network[2] include efficient: (i) data distribution mechanisms; and (ii) data indexing or query routing techniques.

The rest of this paper is organized as follows: in Section 2 we relate our approach with other Grid systems described in the literature. In Section 3 we describe our approach in detail. Finally, conclusions and future work are reported in Section 4.

Related Works

The work described in this paper exploits *Peer-to-Peer* techniques in order to extend the capabilities of the *Grid* architecture. The Peer-to-Peer approach provides a scalable alternative solution to the conventional server-client approach, where the server usually constitutes a vulnerable point of delays and failures.

Depending on their structure, Peer-to-Peer systems can be divided into three categories, namely structured, unstructured and hybrid. In a structured overlay, a DHT is constructed, where (key, value) pairs are stored. A DHT-based system, such as Chord [5], CAN [1] and Pastry [6] guarantees that, if a key exists in the overlay, the lookup algorithm will find it with a lookup cost bounded to the logarithm of the search space, namely the number of nodes participating in the overlay. Unstructured systems, such as Gnutella[3] and Freenet [2], use flooding techniques and support complex queries, including wildcards and ranges. But they do not offer any search guarantees. A query might not find the stored information. Hybrid systems, such as Napster[4], use centralized directories to provide guarantees, which can limit their scalability and flexibility. But the most interesting category, and the one that current research focuses on, is structured data-lookup systems. They build on structured overlays and essentially implement Internet-scale DHTs. Information is located using unique and globally known data identifiers, but a huge disadvantage of these systems is that they don't support complex queries, like range queries.

BATON[6] is a one-dimensional index that uses a distributed search tree where in-level sideway links are employed to reduce the number of accesses to higher levels of the hierarchy and achieve routing efficiency, fault-tolerance, and load balancing. Work in [8] proposed the VBI-tree, a P2P network based on BATON that can adapt many data partitioning schemes.

P-Grid [4] is a one-dimensional distributed index based on a randomized binary prefix tree (trie) which is induced by recursively bisecting the data space. Each peer is associated with one partition of the key space and maintains random connections to other peers such that prefix routing is enabled. A peer stores also some data index replicas belonging to other peers, to guarantee fault-tolerance. However, this way still has high index costs. The work in [6] proposes two algorithms of range query in P-Grid, *i.e.* min-max traversal algorithm (*sequential*) and shower algorithm (*parallel*).

SpatialP2P[8] is a recently proposed structured P2P framework, targeted to spatial data (2 dimensions only). SpatialP2P uses a statically grid-partitioned key space, where nodes handle areas, which are either cells of the grid-partitioned space or sets of cells that do not necessarily form a rectangular region. Although this choice makes the load balancing task more flexible, however the size of metadata needed to manage complex regions may be very large especially for higher dimensions. Similarly to the above methods, SpatialP2P converts the query range into the a set of predefined cells, and route them to the relevant nodes. To reduce the routing overhead, SpatialP2P group non-resolved cells and send them to a neighboring node that is closer to all of them which in its turn will continue the query resolution.

In this paper, We study a tree vector indexing scheme for Grid systems, focusing our attention on two aspects which turn out to be crucial for Grid systems: the tree vector representation and the strategy adopted for indexing Grid resources.

Tree Vector Indexing Scheme

We suppose that each peer in the system holds a (possibly empty) set of data items, also called local repository; each data item is described by a set of attribute-value pairs. For example, in a distributed relational database, a data item would be a database record, and the attribute-value pairs would be the names and corresponding value of the attributes of each table. We suppose that data items are dynamic, in the sense that the value of the attributes may change over time. Users of the P2P system want to locate data items satisfying given search criteria, which are expressed as partial range queries over the set of attributes.

A. Notation and Data Structures

We suppose that each peer in the system holds a set of data items, also called local repository; each data item is described by a set of attribute-value pairs. More specifically, we consider a Grid system where each peer implements the following operations[10]:

insert(D; {A_1 : V_1; : : :A_r; }) Insert a new data item D on the local repository; the data item will be associated with attributes A_1; : : :A_r with values V_1; : : : V_r respectively.

update(D;A : V_{new}) Change the value of attribute A for data item D on the local repository; the new value will be V_{new}.

lookup(Q) Search for any data item matching query Q over the whole Grid system (including the current node).

Additionally, peers may join and leave the system at any time; as usual in Grid systems, we want to rely as few as possible on any centralized information.

In the following we consider a Grid system with a set P = {P_1; P_2; : : : P_N}of N peers. We denote with Data (P_i) the local repository on peer P_i. Each data item is labelled with a set of attribute-value pairs. We suppose that there is a limited number of different attribute names. We denote with {A_1 : T_1; : : :A_M : T_M} the set of all the M possible attribute names with their corresponding types. Data types T_i can be any arbitrary data types, subject to the constraint that there must be a total ordering defined over T_i. Each data item can be labelled with any nonempty subset of attributes of fA_1{A_1; : : :A_M}. For each data item D, we denote with AttList (D) the set of all attribute names defined for D. Moreover, for each attribute A ∈AttList(D), D[A] denotes the value of attribute A for data item D. The following is notation used in this paper.

Data (P_i) :The set of data items present in peer P_i

Nb (P_i): The set of neighbors of node Pi on the overlay network

AttList (D): The set of attributes defined for data item D

D[A]: The value of attribute A for data item D

BitIdx (D[A]) :Bit-vector representation of D[A]

T (P_i→ P_j): The subtree of T = (P;E- {P_i; P_j}) which does not contain P_i

LinkBitIdx (P_i → P_j ;A) :The bit vector index for attribute A associated with the link from P_i to P_j

B. Handling Queries and Updates

Recall from the previous section that node P knows the bit vector LinkBitIdx (P→ P' ;A), for each P'∈ Nb (P), where LinkBitIdx (.) is defined above. Suppose that node P receives query Q := v1≤A ≤ v2 from one of its neighbors P_{in}. The query is propagated along the connection from P to P_{out} ∈ Nb (P) - P_{in} if a match is likely to be present in T (P→P_{out}). A necessary condition for the existence of a match is that the logical "and" between LinkBitIdx (P→P_{out};A) and the bit vector representation of the interval [v_1; v_2], is nonzero.

Upon receiving a query from neighbor P_{in}, the query is forwarded to the remaining neighbors which have a potential match. Results are fanned back to P_{in}, until they eventually reach the originator. Note that this approach only works if the overlay network is guaranteed to be acyclic (i.e., is a tree), as we are assuming. The result of a query is the set of all peers with local data items matching the search criteria.

In order to handling updates, firstly, the new value v_{new} is converted into the corresponding bit vector representation. If BitIdx (v_{new}) is equal to BitIdx (v_{old}), then the update is not propagated to neighbors; if the bit vector representations are different, then the update message is propagated in order to preserve the property. Update messages consists of the name of the attribute whose value is changed, and its up-to-date bit vector representation[10].

Summary

In this paper, In this paper we described a Grid system which supports range queries over dynamic content for Grid systems. Data location is implemented using a distributed data structure based on tree vectors. Routing informations are used to drive queries away from regions of the network where matches cannot be found. The routing informations can be efficiently updated when data is modified.

Acknowledgements

This research is partially supported by the Opening Project of Hubei Key Laboratory of Intelligent Robot (HBIR 201009).

References

[1] S. Ratnasamy et al., “A Scalable Content-Addressable Network”, Proc. ACM SIGCom, ACM Press, 2001, pp. 161-172.

[2] I. Clarke et al., “Freenet: A Distributed Anonymous Information Storage and Retrieval System”, Proc. ICSI Workshop Design Issues in Anonymity and Unobservability, LNCS 2009, Springer-Verlag, 2001, pp. 311-320.

[3] Gnutella. http://www.gnutella.com.

[4] Napster. http://www.napster.com.

[5] I. Stoica et al., “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, Proc. ACM SIGComm, ACM Press, 2001, pp. 149-160.

[6] A. Rowstron and P. Druschel, Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems, Proc. IFIP/ACM Int’l Conf. Distributed Systems Platforms, LNCS 2218, Springer-Verlag, 2001, pp. 329-350.

[7] Chi-Yin Chow and Alvin T. S. Chan. GroCoca:group-based peer-to-peer cooperative caching in mobile environment. In the 33rd International Conference on Parallel Processing (ICPP), Montreal, Quebec, Canada, August 2004.

[8] A. Crespo and H. Garcia-Molina. Routing indices for peerto-peer systems. In Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS’02),Vienna, Austria, July 2002.

[9] Maha Abdallah, Hung Cuong Le. Scalable Range Query Processing for Large-Scale Distributed Database Applications. In Proc. of the International Workshop on the Web and Databases , July 2005.

[10] Moreno Marzolla, Matteo Mordacchini. Tree Vector Indexes: Efficient Range Queries for Dynamic Content on Peer-to-Peer Networks. In 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP'06). 2006.