A Distributed Resource Broker for Spatial Middleware Using Adaptive Space-Filling Curve

Anand Padmanabhan, Shaowen Wang
Cyberinfrastructure and Geospatial Information Laboratory
Department of Geography
National Center for Supercomputing Application
University of Illinois at Urbana Champaign
Email: apadmana@illinois.edu, shaowen@illinois.edu

ABSTRACT

Spatial middleware serves as a glue for high-performance and distributed GIS services to harness the computational capabilities of cyberinfrastructure. This paper focuses on the development of an important component of spatial middleware – a distributed resource broker that matches computation tasks of GIS and spatial analysis to appropriate cyberinfrastructure resources to solve computationally intensive GIS and spatial analysis problems. This distributed resource broker is built on computational intensity estimations and a self-organized grouping (SOG) framework. Specifically, we use computational intensity information to enable cyberinfrastructure resource brokering for spatial middleware by exploiting spatial characteristics; and adapt the SOG framework to enhance resource brokering performance through the use of a space filling curve. A new overlay network is designed to inherit the good performance and distributed self-organizing nature of SOG while the use of computational intensity information enhances computational performance of resource brokering for GIS and spatial analysis applications.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software - *Distributed systems*

General Terms

Algorithms, Design

Keywords

Resource brokering, Self-organization, Cyberinfrastructure, Geographic information systems, Spatial middleware

1. INTRODUCTION

Although cyberinfrastructure provides massive computational power to enable high-performance and distributed GIS (HPDGIS) capabilities, it remains to be complex and evolving in the foreseeable future. Spatial middleware manages the complexity of cyberinfrastructure while exploiting spatial characteristics of data and operations to achieve desirable capabilities and performance of HPDGIS. One challenge for developing HPDGIS based on cyberinfrastructure is centered on resource brokering, i.e. for matching computational tasks of GIS and spatial analysis to appropriate cyberinfrastructure resources for desirable GIS performance and efficient use of cyberinfrastructure.

The purpose of this research is to address this challenge by developing a key element of spatial middleware, a distributed resource broker [1], designed to allocate and manage cyberinfrastructure resources for HPDGIS. The distributed resource broker is built on a self-organized grouping (SOG) framework for grid resource discovery [2; 3] and uses a computational intensity representation [4; 5] to capture computational resource requirements of HPDGIS applications.

It is recognized that computational resources needed by HPDGIS are partly determined by spatial characteristics of data and associated analytical operations [5]. For example, the inverse distance weighted (IDW) spatial interpolator [6] requires more computing cost for clustered datasets versus uniformly distributed datasets given a same problem size. Therefore, spatial characteristics need to be systematically captured in computational intensity estimations for specifying computing, data, communication, and input-output (I/O) requirements. These requirements need to be expressed as part of queries for the resource broker to locate suitable cyberinfrastructure resources for HPDGIS.

In addition to effectively capturing spatial characteristics through computational intensity evaluation, the SOG framework is enhanced to assure consistent performance under dynamic resource needs of HPDGIS, resource availability of cyberinfrastructure, and query load handled by spatial middleware. Specifically, the SOG framework is susceptible to hotspots (i.e. overloaded resources), under heavy management and query load situations. The enhancement described in this paper is designed to mitigate this hotspot problem, and is achieved by the use the Hilbert space-filling curve (SFC) [7] to dynamically adjust the grouping strategy of the SOG framework.

The remainder of the paper is organized as follows. Section 2 describes the architecture of the distributed resource broker and illustrates how it is used for a particular HPDGIS environment. Next, we highlight the challenges associated with the use of the existing SOG framework for distributed resource brokering, and present our enhancing strategy to address these challenges. Finally, we will conclude by summarizing the findings of this research and providing pointers to future work.

HPDGIS'10, November 2, 2010, San Jose, CA, USA.

2. DISTRIBUTED RESOURCE BROKER ARCHITECTURE

Most resource brokering solutions adopt either a centralized (e.g. Condor matchmaker [8]) or fully decentralized (e.g. request forwarding schemes [9]) lookup strategy. Centralized approaches tend to provide good lookup performance, but they scale poorly in size, while decentralized strategies tend to scale up well, but pay for it with a performance penalty. Our distributed resource broker employs a hybrid approach adopted by SOG that exploits the advantages of both centralized and decentralized solutions while mitigating their shortcomings. Figure 1 illustrates the architecture of distributed resource broker that is composed of three tiers: a) computational intensity evaluation; b) query generation; and c) the SOG framework.

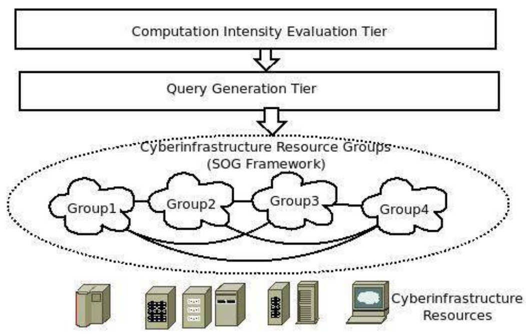


Figure 1: Architecture of distributed resource brokering.

The computation intensity evaluation tier assesses computational requirements of HPDGIS by capturing the spatial characteristics of data and associated analytical operations. This tier is composed of a number of modules, each of which transforms spatial data and associated analysis into computational requirement specifications. A specification represents one or more of the following four types of computational intensities: compute, memory, network, and input and output.

The query generation tier uses computational intensity information and translates it into resource request queries. These queries specify the characteristics of cyberinfrastructure resources needed to meet the computational requirements of HPDGIS applications. This tier serves as a layer between spatially-explicit computational intensity representations and generic cyberinfrastructure resource requirements understood by the lower tier. Request queries are dispatched as resource brokering requests to the SOG framework.

The SOG framework matches resource requests to cyberinfrastructure resources to enable efficient resource brokering. The basic idea guiding the design of the SOG framework is to exploit the inherent similarity among cyberinfrastructure resources, which is achieved by assembling resources into groups based on resource characteristics and the attributes used for brokering [2]. In our architecture, the aforementioned computational intensity information forms the resource characteristics used for resource grouping in the SOG framework. Specifically, for each cyberinfrastructure resource, SOG captures the following characteristics as resource attributes: 1) compute (e.g. compute load, CPU speed, number of cores); 2) memory (e.g. total, available memory per node/core); 3) disk (e.g. disk access speed, latency, availability of fast solid state devices); and 4) network (e.g. bandwidth, latency). These four categories of resource characteristics will form the resource space used for the calculation of the Hilbert SFC that in turn determines resource similarity and facilitates resource brokering [3].

2.1 Case Study

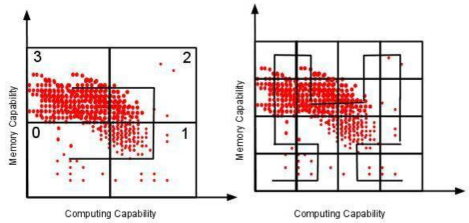
In order to illustrate how this 3-tier architecture works, we consider a HPDGIS use-case scenario - GISolve [10] based on spatial middleware. GISolve integrates high-performance, distributed, and collaborative computational approaches to computationally intensive GIS and spatial analysis problems. Hundreds of registered users can simultaneously perform GIS and spatial analysis functions, including, for example, surface interpolation [6] and local clustering detection [11]. In order to support simultaneous use of HPDGIS by a large number of users and provide results in a timely fashion, GISolve assembles resources from multiple cyberinfrastructure environments including the National Science Foundation TeraGrid [12], and the Open Science Grid [13]. Therefore, GISolve provides a perfect avenue for studying the applicability of the distributed resource broker.

Without loss of generality, we can consider two commonly used spatial analysis functions: IDW spatial interpolator [6], and $G_i^*(d)$ statistic [14], as examples of HPDGIS applications that the resource broker will serve. The first step of resource brokering involves evaluating the computation intensity. For example, computing requirements are specified based on equation 1 by the computational intensity evaluation module, which calculates the compute intensity associated with cell (i, j) for the IDW, denoted as $\mathrm{CI\_IDW}_{(i,j)}$ [5].

$$CI\_IDW_{(i,j)} = timeUnit \times \frac{ne_{(i,j)}}{ns_{(i,j)} \times \sqrt{dp_{(i,j)} + densityThreshold}} \quad (1)$$

where ne is the number of grid cells invoking k-nearest neighbor search; ns the number of sampling points; dp denotes the local sampling point density; $timeUnit$ is a conversion factor; and $densityThreshold$ is a constant that prevent zeroing of the denominator. Since computing time is the only significant factor affecting the performance of spatial interpolator [4], this would be the only criteria considered for resource brokering. In contrast, the calculation of the $G_i^*(d)$ statistic is both compute and memory intensive [5]. Using the computational intensity equations for $G_i^*(d)$ analysis from [5], the computational intensity evaluation tier, identifies computing and memory requirements.

The query generation tier will take the computational requirements identified by upper tier and generates the request for cyberinfrastructure resources needed to meet the computational challenges associated with this analysis. For the spatial interpolator, the computational intensity generated using equation 1 indicates that the clustered dataset tend to require a lot more computational resources that uniformly distributed datasets. So based on the spatial characteristics of dataset, a medium or high-end computing resource is requested. On the other hand since the $G_i^*(d)$ calculations are both compute and memory intensive, resource requests generated by this tier will request high-end compute resource with a large amount of memory. GISolve supports a number of spatial analysis application, and for every such application, we define a module to assess the computational intensity which is translated to resource requirements by query generation tier. So a large variety of resource requests can be expected. The queries are then dispatched to the SOG framework for looking up resources.



(A) **(B)**

Figure 2: Hilbert SFC applied to resources in a *2*-dimensional resource space: (A) order 1, (B) order 2.

SOG matches the requests with appropriate cyberinfrastructure resources. To better understand how SOG is able to match a diverse set of queries from HPDGIS, let us consider a straightforward example concerning two primary attributes: computing, and memory capabilities. These attributes form the *2*-dimensional resource space, on which every resource is located. For example, every point in figure 2 represents a resource and its location in the resource space is determined by the attribute values (e.g. resource with high computing and memory capabilities will be in group 2 in figure 2-A, while resources with low computing and memory capabilities will be in group 0). SOG exploits the inherent clustering of cyberinfrastructure resources, and the locality preserving nature of Hilbert SFC [7] mapping to organize similar resources into groups. Figure 2 illustrates how resources are grouped together in SOG and how the granularity of grouping can be controlled with the order of Hilbert SFC. By grouping similar resources together, the process of matching query to a resource is simplified into a 2-stage operation: 1) route the query to appropriate resource group; and 2) find a suitable resource within the group. The locality preserving Hilbert SFC mapping facilitates the former, while a lookup at the group leader enables the later. If, for example, a requests is made for a high end compute resource with large memory, the query will get routed to group 2 in figure 2-A. Once the queries are directed to appropriate groups, they are matched to available resources within the group.

3. SOG FRAMEWORK ENHANCED BASED ON HILBERT SFC

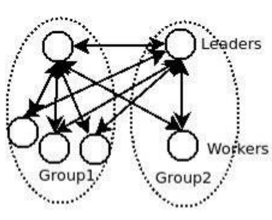


Figure 3: Upper level graph of SOG framework (after [2]).

In order to provide the key functionality of efficiently searching cyberinfrastructure resources, the SOG framework maintains two-level overlay network. The lower level layer, formed using a lightweight gossip protocol [15], consists of all the cyberinfrastructure resources, and is used to form and maintain the SOG groups by supporting a distributed leader election [2]. The upper level layer, that interconnects leaders and workers, is a graph consisting of all resources and composed of links from every resource to all the leaders, and every leader to its workers (figure 3). A leader is a resource group's representative, while workers are all its members. The upper layer is designed to be used for efficiently routing queries in the distributed environments. Queries are routed in two stages: first to the resource groups and then to the resources. Since group leaders represent resource groups, queries directed to a resource group will be routed to group leaders in the first stage. In the second stage, a leader will conduct a lookup on the resources in its group and route the query to a matching resource. Hence, group leaders are critically important for query processing. Furthermore, leaders also perform the important task of managing SOG groups. It is therefore not hard to see that under heavy management and query load situations, the leaders can become severely overloaded and turn into bottlenecks. We identify a group leader under such a condition as a hotspot.

Because of the dynamic nature of resource environments and queries encountered by SOG in spatial middleware, hotspots can develop dynamically and without notice. An approach to handling hotspots, which is currently employed by SOG framework, is to control granularity of SOG groups, with the help of the order parameter (k) in the Hilbert SFC (figure 2). Though it is possible to mitigate the problem of hotspots by choosing a high-order curve, the selection of an appropriate k value is a challenge. Furthermore, a high order Hilbert curve will partition the entire resource space finely creating a lot of unnecessary groups by splitting resource groups of leaders that are not overloaded. This may negatively impact performance. Additionally, since hotspots can appear dynamically due to query routing load, it is unlikely one would be able to select an optimal value for the order suitable for all of the possible load patterns.

To tackle the aforementioned challenge, we propose a solution based on adaptive Hilbert SFC. Adaptive SFC in this context implies that instead of having a Hilbert SFC of a particular order for the entire resource space, different partitions of the space have different order curve passing through it. To facilitate this we enhance the SOG overlay and provide mechanisms by which once a hotspot is detected the leader that is getting overloaded will split its group. On the other hand if subgroup leaders realize that they are lightly loaded they can merge together into a single group. By the application of this strategy the order of the Hilbert SFC is dynamically adjusted based on the number of resources in a group and the load on each group leader.

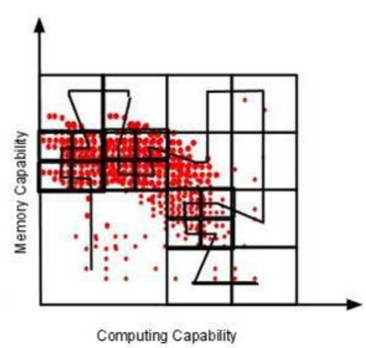


Figure 4: Adaptive Hilbert SFC applied to resources in a *2*-dimensional resource space.

Figure 4, illustrates the result of applying adaptive SFC on a *2*-dimensional resource space. As is apparent from figure 4, orders 1, 2 and 3 of Hilbert SFC is applied on different regions depending on the density of resource and load distribution. By doing so we mitigate the problem of leaders overloading as well as refrain from creating lots of unnecessary groups that may negatively affect performance. The development of such a solution however requires enhancing the overlay network to enable group splits and merges based on adaptive Hilbert SFC, which we outline next.

3.1 Enhancement of SOG Overlay Network

The lower level layer is left unchanged by our enhancing strategy. Figure 5 illustrates the modification we made to the upper-level layer graph. Like the original overlay, this graph contains all the resources, and all resources still have links to level 1 (top level) group leaders, but the links within the groups have been updated. If a group has no subgroups, the leader will have links to all the workers. If however, a group is further divided into subgroups, the group leader will only maintain links with the subgroup leaders. Essentially this graph is established as a tree structure with added direct links to level 1 leaders (not shown in the figure). The number of levels is determined by the order *k* of the Hilbert SFC used at the deepest level. Even with this new overlay the diameter of the graph is kept relatively small (diameter=*k* (order of Hilbert SFC), 2 for original overlay), which means that the query performance of SOG will not be adversely affected.

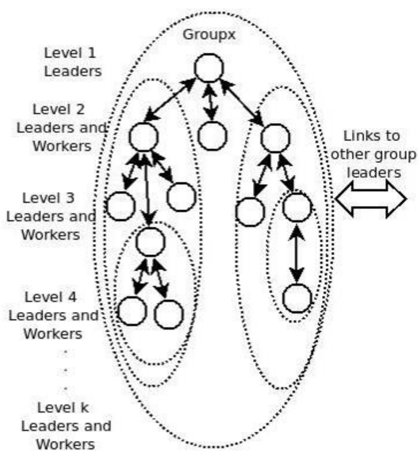


Figure 5: A single group in the modified upper level graph of SOG framework.

With the application of the proposed changes to the SOG framework, the distributed resource broker architecture is capable of addressing the challenges it is expected to encounter when used as a component of spatial middleware.

4. CONCLUSION AND FUTURE WORK

This paper described a distributed resource broker designed for use in spatial middleware based on the SOG framework and leveraging computational intensity information. This was accomplished by developing an enhanced overlay network for SOG framework, and advocating the use of adaptive Hilbert SFC to group resources, to improve the performance of SOG framework in scenarios it is likely encounter when used in spatial middleware. Furthermore, we enabled the utility of computational intensity information to characterize resource requirements. By utilizing the spatial characteristics of analysis and data encapsulated in computational intensity, the distributed resource broker facilitates the matching of HPDGIS with cyberinfrastructure resources, while the use of the enhanced SOG framework will enable the resource broker to inherit SOG's distributed and self-organizing characteristics.

In this article we have elucidated the architecture of the distributed resource broker for spatial middleware. As a next step, we plan to conduct experiments to study the effect of using adaptive Hilbert SFC and overlay modifications on the performance of spatial middleware.

5. REFERENCES

[1] S. Wang, A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis. Annals of the Association of American Geographers (2010).

[2] A. Padmanabhan, S. Wang, S. Ghosh, and R. Briggs, A self-organized grouping (SOG) method for efficient Grid resource discovery, GRID 2005, 2005, pp. 312-317.

[3] A. Padmanabhan, S. Ghosh, and S. Wang, A Self-Organized Grouping (SOG) Framework for Efficient Grid Resource Discovery. Journal of Grid Computing 8 (2010) 365-389.

[4] S. Wang, and M.P. Armstrong, A Theory of the Spatial Computational Domain, Proceedings of GeoComputation 2005 (CDROM), Ann Arbor, MI, USA, 2005.

[5] S. Wang, and M.P. Armstrong, A theoretical approach to the use of cyberinfrastructure in geographical analysis. International Journal of Geographical Information Science 23 (2009) 169 - 193.

[6] S. Wang, and M.P. Armstrong, A quadtree approach to domain decomposition for spatial interpolation in Grid computing environments. Parallel Computing 29 (2003) 1481-1504.

[7] H. Sagan, Space-filling Curves, Springer-Verlag, 1994.

[8] R. Raman, Matchmaking: Distributed Resource Management for High Throughput Computing. in: M. Livny, and M. Solomon, (Eds.), 1998, pp. 140-140.

[9] A. Iamnitchi, and I.T. Foster, On Fully Decentralized Resource Discovery in Grid Environments, Proceedings of the Second International Workshop on Grid Computing, Springer-Verlag, 2001.

[10] GISolve, The GISolve Toolkit 2010.

[11] S. Wang, M.K. Cowles, and Marc P. Armstrong, Grid computing of spatial statistics: using the TeraGrid for Gi*(d) analysis. Concurrency and Computation: Practice and Experience 20 (2008) 1697-1720.

[12] TeraGrid, The NSF TeraGrid Project, 2010.

[13] OSG, The Open Science Grid Project, 2010.

[14] J.K. Ord, and A. Getis, Local spatial autocorrelation statistics - Distributional issues and an application. Geographical Analysis 27 (1995) 286-306.

[15] A.J. Ganesh, A.-M. Kermarrec, and L. Massouli, SCAMP: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication, Proceedings of the Third International COST264 Workshop on Networked Group Communication, Springer-Verlag, 2001.