Spatio-Temporal Portals for Continuously Changing Network Nodes

Byunggu Yu
National University, USA

Ruben Gamboa
University of Wyoming, USA

INTRODUCTION

OGSA is a service-oriented architecture (SOA); that is, server nodes in the grid advertise the services they offer, client nodes use the grid to find servers that meet their specific requirements, but neither server nodes nor client nodes are closely tied to each other. When a client has a request, the grid infrastructure identifies a set of the servers that can fulfill the request: OGSA grid is based on platform-neutral technologies and, given a request, identifies appropriate servers directly by the interfaces (i.e., services) they offer.

In recent years, many emerging applications, such as mobile network applications and sensor network applications (Gaynor, Moulton, Welsh, LaCombe, Rowan, & Wynne, 2004; Ghanem, Guo, Hassard, Osmond, & Richards, 2004), involve network nodes that can continuously change or move over time (*moving grid nodes,* or simply, *MGNs*). For example, mobile, floating, and airborne sensors/\computers are MGNs. These spatio-temporal nodes are typically connected using wireless technology and customized, energy-preserving protocols with energy drawn from a limited power source such as a battery. Importantly, in these applications, most requests come with the necessary spatio-temporal attributes of the MGNs. For example, the current air pollution level in a certain city can be found using the sensors (MGNs) that are currently in the city.

To make MGNs accessible in the standard grid, one can use intermediate hosts (*MGN portals*) that communicate with a set of MGNs using protocols designed to extend the MGNs' lifespan while exposing the MGNs to the client network using the standard grid (or Internet) protocols (a similar approach can be found in (Gaynor et al., 2004; Ghanem et al., 2004)). It is the MGN portals and not the MGNs themselves that use the standard networking protocols. Each MGN portal may represent a set of MGNs, and applications may interact with the MGNs via the MGN portals using the standard APIs; thus bringing the benefits of a standard programming environment to the developers of various MGN network applications.

This article investigates the relevant issues in designing an MGN portal. The proposed framework's spatio-temporal data models, update models, and query system can significantly improve the performance and scalability of MGN portals.

BACKGROUND

The open grid services architecture (OGSA) (Foster & Kesselman, 1998, 2001) is built on top of standard Web services technology. Web services allow a client running on one computer to access a service function running on a possibly different computer with a different architecture, written in a possibly different language. The grid also borrows another important concept from Web services: discovery. Web services use universal description, discovery, and integration (UDDI) to advertise and find services. OGSA also addresses many issues of common interest to distributed applications, such as security, scheduling, and monitoring. These are important considerations, but this article focuses on the discovery and matching of services and requests in the grid (Foster & Kesselman, 1998, 2001). When a client has a request, the grid infrastructure identifies a set of connected servers that can fulfill the request. OGSA identifies servers directly by the interfaces (i.e., services) they offer. These interfaces are described using WSDL (Web Service Description Language) and registered a priori.

On the other hand, recent sensor network applications require wireless networks that interconnect spatially distributed wireless servers and clients with energy drawn from a limited power source, such as battery. This limited energy requires a parsimonious approach to networking, including minimizing the number of bits used to transmit a message by using customized protocols. This stands in contrast to the grid protocols, which value interoperability over economy. To resolve this mismatch, sensor grid networks (Gaynor et al., 2004; Ghanem et al., 2004) use intermediate hosts (*sensor gates*) that communicate with the sensors using protocols designed to extend the sensors' lifespan while exposing the sensors to the grid using the grid protocols. It is the sensor gates and not the sensors themselves that use the grid protocols. Each sensor gate may represent a set of sensors in the grid. Nevertheless, applications may interact with the sensors via the sensor gates using the standard grid APIs, thus bringing the benefits of a standard programming environment to the developers of sensor network applications.

In recent years, we are witnessing even more challenging demands: the spatio-temporal properties of servers and the clients are also required to identify a match in many current and future grid applications, including moving-sensors network applications. Example applications include monitoring patient stats (e.g., pulse, oxygenation) in a hospital setting, optimizing a supply chain using RFID systems, and monitoring air pollution (sensors used to detect SO2 and NO2) using airborne or mobile sensors, to name a few (Gaynor et al., 2004; Ghanem et al., 2004). In these applications, wireless networks interconnect spatially distributed moving grid nodes (MGNs). This leads to the following research challenges: (1) How can the grid scheduler estimate the current, and future positions (and other changing parameters, e.g., CPU and memory loads) of MGNs, such as moving sensors and moving sensor gates?; (2) How can the grid keep a history of MGN positions (and other changing parameters, such as speed, direction, CPU utilizations, and memory utilizations) in order to monitor the behavior of the grid or to support grid services that refer to the past locations of MGNs?

MGN PORTAL

One approach to the main challenges is developing a new grid service layer consisting of one or more instance nodes that can provide the standard grid with spatio-temporal data and requirements of MGNs and their services. Importantly, as in the sensor gate approach, these intermediate nodes (more specifically, the grid resources used by the nodes) represent pure overhead of this approach. Therefore, this new layer must be designed to efficiently scale to large set of MGNs. Because of this reason, we call this layer MGN "portal." To develop an MGN portal in the grid, one can consider the evolution of *metadata and catalog service* (*MCS*) (Deelman et al., 2004) as a related case. Recently proposed grid-based MCS (Deelman et al., 2004) is built on OGSA-DAI (open grid service architecture—data access interface) and MySQL DBMS. The DAI, which was designed to smoothly connect relational database systems to OGSA grids, provides a security infrastructure based on public-key authentication. This is the basis on which the MCS can provide fine-grain (i.e., record-level) access control and organizational security policies. This existing MCS system provides a sound basis for developing an MGN portal.

The MCS is based on well-established relational database technology that can efficiently manage conventional (relational) databases in the grid. To use this system for our own purposes, the following question needs to be answered: *how to keep track of MGN whereabouts and their registered grid services on a relational database system*.

This section provides a basis for designing an MGN portal managing an MGN database and a set of services that access the MGN database to store, update, and retrieve the information of MGN services and whereabouts. An MGN database consists of two connected data sets: one is a set of MGNs; the other one is a table of services provided by the MGNs. The latter set is a conventional data set that can be well managed by a relational database system. However, managing the former set on a relational database system poses a major design challenge. This is due to the fact that MGNs have continuously changing attributes.

For example, how do we support a request asking for the average and spread of NO2 levels of a certain geographic region *R* over the past 24 hours? To find the matching MGNs: (1) generate a spatio-temporal query region *Q* in such a way that the projection of *Q* onto the geographic space is *R* and *Q* extends from the current point C in time to the time point that is 24-hours earlier to *C*; (2) select all MGNs *S* such that the trajectory of *S* intersect *Q*; (3) select all NO2 sensor equipped MGNs from *S*.

Relational MGN Representation with Uncertainty

As explicated in Table 1, in an MGN portal, each MGN's trajectory, which represents the spatio-temporal properties of the MGN, is stored as a sequence of connected segments in space-time, and each segment has two endpoints that are consecutively reported *states* of the MGN. Figure 1 shows a generic MGN dataset schema that can be created on a relational database system, and that can support the ontological concepts explicated in Table 1. Examination of Table 1 and the commensurate ER diagram in Figure 1, one may observe the following: (1) snapshots are not represented in the schema; (2) only a subset of states, called reported states, are included in the schema. These differences exist due to the fact that a database cannot be continuously updated. All in-between states and future states of the MGNs are then interpolated and extrapolated on the fly (Yu, Kim, Bailey, & Gamboa, 2004) only when it is necessary for request-service matching, MGN trajectory data visualization, index maintenance, or data management. Therefore, a mathematical model and computational approach is required to efficiently manage the "in-between" and "future" states' snapshots.

Figure 2(a) shows an example of a trajectory segment connecting two known (reported) states of an MGN. Let M_v be the maximum rate of change (i.e., the norm of the maximum possible velocity) of the MGN, *A* be the reported state (value) of this MGN at t_i, and *B* be the state at time t_j. Then all possible states of the MGN between t_i and t_j are bounded by the lines where $|\cot\theta| = M_v$. The shaded region covers all possible locations (i.e., more generically, states) of the MGN between t_i and t_j. We call this region the "*spatiotemporal uncertainty region*" of the trajectory segment <A B>. The