Implementation of an Intelligent Urban Traffic Management System Based on a City Grid Infrastructure*

FEILONG TANG, MINYI GUO, MINGLU LI AND CHO-LI WANG+
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai 200240, P. R. China
+Department of Computer Science
The University of Hong Kong
Hong Kong 999077, P.R. China

Grid computing is becoming the preferred basis for large-scale distributed computing and Internet applications. ShanghaiGrid launched in Shanghai, China is a grid infrastructure that aggregates several heterogeneous supercomputers, data centers and other applications scattered in different organizations in Shanghai. One of the major focuses in the ShanghaiGrid project is research and development of an intelligent urban traffic management system based on the ShanghaiGrid system software. The system software provides a platform in a service-oriented way for resource encapsulation and management, service scheduling and accounting, data aggregation and adaptive transmission, as well as service composition and reliability support. This paper presents the main components of the system software, and investigates key techniques for the implementation of ShanghaiGrid and the intelligent urban traffic management system, concentrating on grid transaction and workflow management. Finally, we propose a new wireless sensor network model and routing protocol characteristic to mobile vehicles for dynamical traffic data collection.

Keywords: grid computing, intelligent traffic management, transaction processing, workflow management, wireless sensor network

1. INTRODUCTION

Grid computing is a natural evolution of distributed computing and Internet applications for large-scale science and engineering problems, aiming at effective resource sharing and task collaboration in distributed and self-governing environments [1]. Both academia and computer industry regard the development of grids as another chance to improve the current paradigm of Internet computing. As an internationalized city, Shanghai presents emergent needs for an information infrastructure to enable sharing of heterogeneous resources to improve government efficiency and quick response to emergent events. Due to the 'information island' problem, current resources of computing, storage and data cannot be directly shared because of different hardware and software systems, different databases and different security models. In response to these challenges, Shanghai municipality launched the ShanghaiGrid project in 2003. It is one of five prin-

Received January 11, 2007; revised July 18, 2007; accepted October 18, 2007.
Communicated by Tei-Wei Kuo.
* This work was supported by the National High Technology Research and Development Program (863 Program) of China (Grant Nos. 2006AA01Z172 and 2006AA01Z199), the National Natural Science Foundation of China (NSFC, Grant Nos. 60773089, 60533040, and 60725208), Natural Science Foundation of Shanghai Municipality of China (Grant No. 05ZR14081) and Shanghai Pujiang Program (Grant No. 07pj14049).

cipal grid projects in China [4]. From the perspective of hardware infrastructure, ShanghaiGrid aggregates various distributed and heterogeneous resources, including computers, networks, storage devices and so on. From the perspective of software infrastructure, one of the research focuses is to develop the ShanghaiGrid system software (SHGSS) and then implement an intelligent urban traffic management system.

SHGSS is a layered toolkit that comprises the low-level access systems for encapsulation and management of physical resources, the grid middleware and the top application support layer. It is a joint effort of hundreds of researchers and developers from more than 6 organizations. Currently, SHGSS can support resource encapsulation and management, service scheduling and accounting, data aggregation and adaptive transmission as well as service composition and transaction management. Intelligent traffic management is a main application of ShanghaiGrid to enhance the efficiency of Shanghai traffic systems and make people travel in Shanghai more easily.

The rest of this paper is organized as follows. In the next Section, we introduce the ShanghaiGrid infrastructure. Section 3 investigates key techniques of ShanghaiGrid system software, especially on grid transaction and grid workflow management. In section 4, we present the intelligent urban traffic management system. Section 5 proposes a new solution to traffic data collection by using wireless sensor networks equipped on mobile vehicles. Finally, we draw our conclusions in section 6.

2. SHANGHAIGRID INFRASTRUCTURE

2.1 ShanghaiGrid Hardware Infrastructure

ShanghaiGrid interconnected supercomputers, data storages, devices (*e.g.*, sensors and traffic surveillants) and other resources, with four major computational aggregations and networks in Shanghai, *i.e.*, CHINANET (public Internet backbone built by China Telecom), SHERNET (Shanghai Education and Research Network), STNC (Shanghai Science and Technology Network Communication), and campus networks [4]. As Fig. 1 illustrates, the ShanghaiGrid hardware infrastructure comprises more than five intra-grids, connected to different backbone networks at a 100Mbps network bandwidth, where at the top-left corner is the logo of ShanghaiGrid. We briefly introduce these intra-grids as follows.

Shanghai Supercomputing Center (*SSC*) *intra-grid* mainly includes a SW-I supercomputer with 384 CPUs, 384Gflops peak computing capability and 1280GB disk capability. *Shanghai Jiao Tong University* (*SJTU*) *intra-grid* consists of a IBM P690 with 32 CPUs, 166Gflops peak computing capability, 64GB memory and 4000GB disk capability, and a IBM e-Server Cluster 1350 with 18 CPUs, 72Gflops peak computing capability, 14GB memory and 1200GB disk capability. *Shanghai University* (SHU) *intra-grid* is based on one Ziqiang2000 supercomputer with 215CPUs, 450Gflops peak computing capability, 26GB memory and 2000GB disk capability. *Tongji University* (*TJU*) *intra-grid* mainly includes a Dawn 3000 supercomputer with 106Gflops and 3040GB disk capability. Finally, *Shanghai Urban Traffic Information Center* (*SUTIC*) *intra-grid* consists of 20 servers in the Shanghai Traffic Network that is managed by the SUTIC. The center is a government bureau responsible for managing data of most Shanghai traffic systems.

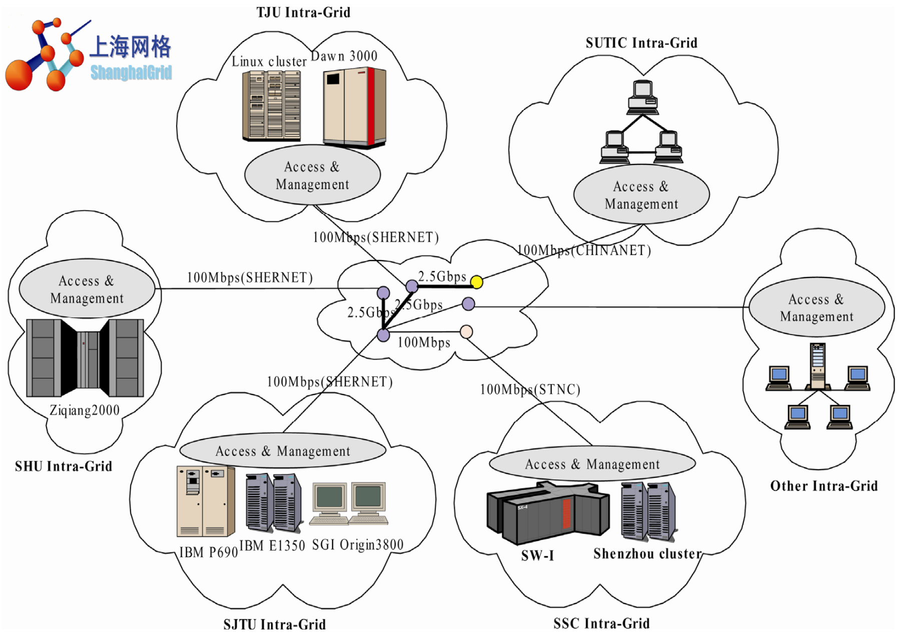


Fig. 1. ShanghaiGrid hardware infrastructure.

2.2 ShanghaiGrid System Software

ShanghaiGrid system software SHGSS was designed with three levels: the low-level access and management software, the grid middleware and the top application support layer to enable various applications to access ShanghaiGrid effectively and conveniently, as shown in Fig. 2.

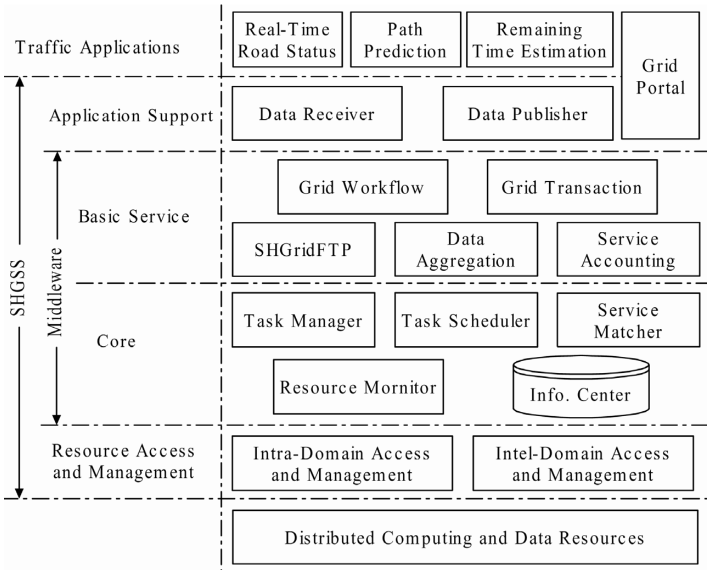


Fig. 2. ShanghaiGrid system software (SHGSS).

2.2.1 Resource access and management

Resource access and management is used to encapsulate heterogeneous resources as grid services, register them in the information center as well as provide users with a set of resource description, service storage and query. It comprises two modules: intra-domain management and intel-domain management. Resources in ShanghaiGrid infrastructure are organized into different domains based on their locations to improve access efficiency. Intra-domain module deployed in a leaf domain manages computing nodes, devices and data registered in the domain, using the interfaces: RS_Temp_regis-ter(), RS_register(), RS_Temp_list(), RS_query() and RS_update(). Intel-domain management module coordinates different intra-domain management modules.

2.2.2 Grid middleware core

To improve the adaptability and flexibility, ShanghaiGrid middle is divided into two sublayers: *Core* and *Basic Service*. The former is a kernel that all applications have to use while the latter is optional.

The *Core* provides top-level applications with transparent access to physical grid services, no matter where grid services are located [5-7]. It comprises Task Manager, Task Scheduler, Service Matcher, Information Center and Resource Monitor. The interaction relationship among the components will be illustrated in section 4 (see Fig. 6). Main functions of these components are described as follows.

Task Manager responds to the task requests from top-level applications, queues the tasks in a task pool, and in turn dispatches queuing task requests to the Task Scheduler. It queues the incoming tasks in the First In First Out fashion in default. Moreover, it also supports the priority policy, *i.e.*, a task with the highest priority is served first.

Information Center maintains the up-to-date information of grid resources and services. It provides the interface RegisterInfo() for a service provider to register grid resources (*e.g.*, one IBM P690 supercomputer) and services (*e.g.*, a computation service on P690). It also matches qualified services and returns the service information for a service query request. Whenever a service container creates or destroys a grid service instance, it notifies the Information Center of this action.

Task Scheduler queries the Information Center for a task scheduling request based on the task requirements. By analyzing and matching the information of grid resources and service instances that run on the resources, the task scheduler can recognize the current load of the grid resources, and allocate the task to the resources with light load.

Service Matcher actually dispatches the task to the matched physical resources, then collects and returns execution results to Task Manager.

Resource Monitor monitors the registered grid services and resources, including the status of service instances (Pending, Active, Suspended, Destroyed, *etc*.), the utilization rate of CPU, memory and disk, *etc.*

2.2.3 Grid middleware basic service

ShanghaiGrid middleware's basic service provides components needed often, which can be extended in "Plug and Play" way. Currently, we implemented Data Transmission (called SHGridFTP), Data Aggregation, Service Accounting, Grid Transaction and Grid Workflow as basic services.

SHGridFTP is a grid based large-scale adaptive data transmission component that supports FTP (File Transfer Protocol), GridFTP [9], BBFTP (an open-source software for massive data transmission across a wide area network), SFTP (Secure FTP) and HTTP (HyperText Transfer Protocol) data transmission protocols. SHGridFTP consists of a PAFTP (protocol-adaptive FTP) for data transmission and a user-friendly GUI for users to manage data sites, data transmission and logs.

Data aggregation seamlessly integrates heterogeneous and geographically distributed data to virtualize multiple data sources, based on the Open Grid Services Architecture-Data Access and Integration (OGSA-DAI) [11]. By using the data aggregation service, top-level applications can transparently access traffic data collected and managed autonomously in different government bureaus.

Service accounting evaluates the service cost and then transfers it to the account of the service provider, with five modules: Collector, Charger, Biller, Account Manager and Database Accessor. Collector queries prices of services involved in a task, then transfers the price to Charger. Charger actually computes the service cost. Biller generates and stores bills according to the accounting results, and handles the bill query from users and system administrators. Account Manager maintains users' accounts, and supports balance updating and query. Database Accessor actually accesses to databases for other modules.

Grid Transaction and *Grid Workflow* services will be discussed in details in the next section.

2.2.4 Application support

Application Support is an application-specific layer, and can be extended to enable SHGSS to support more kinds of applications. Current implementation includes Grid Portal, Data Receiver and Data Publisher, characteristic to traffic applications. Data Receiver gathers traffic data from different data sources (*e.g.*, GPS data centers) and Data Publisher displays traffic information on appropriate media (*e.g.*, e-displays, PDAs and mobile phones).

ShanghaiGrid Portal was designed for convenient access to Shanghai traffic information. Currently, it provides user with management portlet, actor management portlet, actor assignment portlet, log portlet, certificate portlet and chat portlet.

3. KEY TECHNOLOGIES IN SHGSS

We have implemented the ShanghaiGrid system software SHGSS version 1.0 and established an SHGSS-based ShanghaiGrid testbed in 2004. SHGSS 1.0 is able to register and discover grid services, monitor grid services and resources, and submit tasks to grid services. In 2006, we upgraded the system software to SHGSS 2.0, which was extended to support grid transaction and grid workflow management, which will be presented in this section in details.

3.1 Grid Transaction Management

Grid transaction provides reliable execution environments for advanced grid applications [12, 13]. In transactional grid applications, if some sub-tasks of a transaction cannot complete, other committed sub-tasks have to been undone with compensating transactions [14, 15]. The ShanghaiGrid transaction service consists of the following modules (see Fig. 3): *Coordinator* and *Participant* control the execution of a transaction through interacting coordination messages, *Scheduler* for creation and promulgation of a transaction context, *Compensating Transaction Generator* (*CTG*) for automatic generation of compensating transactions [10], and *Log Service* for maintaining the coordination process and state information.

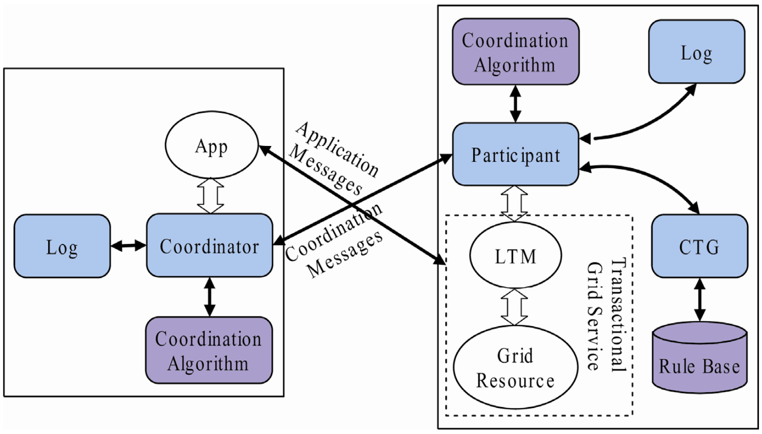


Fig. 3. ShanghaiGrid transaction service.

The transaction service orchestrates the grid services involved in an application and maintains system consistency from failures. The flow of transaction processing is illustrated in Fig. 4, where t is the system time, and T_{valid} the valid time before which a Coordinator must send a Confirm or Cancel message and the Participants must report their status. Otherwise, if the Coordinator does not confirm or cancel a sub-transaction before T_{valid}, the corresponding Participant will automatically undo the committed sub-transaction by the compensating transaction. On the other hand, a Coordinator presumes that a participant has failed if the Participant does not return the commit result before T_{valid}. When a global transaction successfully finishes, Coordinator confirms all committed subtransactions. Once one subtask cannot be performed, Coordinator notifies Participants to compensate subtransactions committed previously.

3.2 Grid Workflow Management

Grid workflow management is responsible for automatic composition of multiple services involved in an application into a virtual service [2]. With the following modules (see Fig. 5), it supports process definition, workflow enactment, workflow process administration and monitoring. The *Workflow Design* provides users with a graphical design

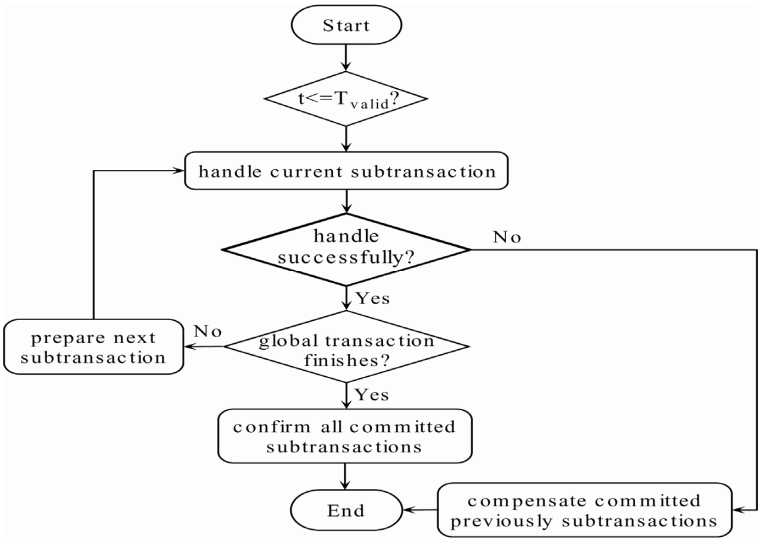


Fig. 4. The flow of grid transaction processing.

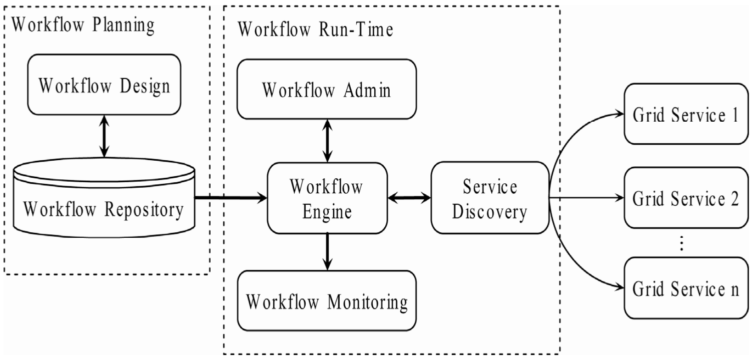


Fig. 5. ShanghaiGrid workflow service.

environment to model the processes as workflows, which are stored in the *Workflow Repository*. The *Workflow Engine* is the core of the workflow service. When a workflow is instantiated, the engine creates the process data, binds and invokes the grid services, and executes the workflow. The *Workflow Admin* and the *Workflow Monitoring* are used to manage the workflow service and monitor the real-time information of workflows, such as process numbers, user numbers, *etc.* ShanghaiGrid workflow service has the following features.

(1) Event-condition-action (ECA) based workflow triggering mechanism.

The ShanghaiGrid workflow service employs an event (atomic and composite) triggering mechanism based on the ECA [3, 8]. We define five types of atomic events: the time (absolute and relative) event, the method call event, the state transition event, the transaction processing event (*e.g.*, Begin and Confirm) and the user-defined event. The composite events are composed of the atomic events or lower-level composite events, using the operators with specified priorities listed in Table 1, where E_i ($1 \le i \le n$) indicates an atomic event or a composite event. The highest priority is 1.

Table 1. Workflow composite operators.

Operator	Priority	Expression	Description
ANY	1	ANY(m, E_1, E_2, …, E_n)	m out of n events E_i occur(s)
EXC	1	EXC(E_1, E_2, …, E_n)	Only l out n event E_i occurs
NOT	2	NOT E_1	E_1 does not occur
AND	3	E_1 AND E_2	Both E_1 and E_2 occur
ALL	3	ALL(E_1, E_2, …, E_n)	E_1 AND E_2 AND…AND E_n
OR	4	E_1 OR E_2	At least one of E_1 and E_2 occurs(s)
REP	5	REP E_1 n	E_1 occurs n times
PRE	6	E_1 PRE E_2	E_1 occurs before E_2
SEQ	6	SEQ(E_1, E_2, …, E_n)	E_1 PRE E_2 PRE…PRE E_n

(2) Automatic workflow execution.

The Workflow Engine controls automatic execution of grid workflows through the ECA rule parser, the workflow instance manager, the event manager, the variable manager and the state manager. When a workflow request comes, the workflow engine obtains a copy of the workflow model from the workflow repository and then automatically handles the specific invocation and routing based on the defined ECA rules. Workflow run-time environment provides a container for workflow applications, and manages workflow lifecycle such as deployment, undeployment, initiation, startup and stop.

(3) User-friendly workflow design.

A workflow model consists of activities, events, conditions, logic nodes, a control flow and a data flow. The workflow design tool provides a graphical representation for a user to create workflow models quickly and easily. This GUI based modeling tool supports drag and drop operations to add new components to workflow models, and transforms the graphical model into a set of ECA rules. The user can also use the right-click menu to find the properties of selected components. Moreover, the design tool is able to import existing workflows to facilitate workflow modeling.

4. INTELLIGENT URBAN TRAFFIC MANAGEMENT SYSTEM

ShanghaiGrid aggregates more than 1300Gflops peak computing capability and 4TB storage, providing the full support for the traffic data processing. For collecting traffic data, Shanghai traffic bureau has deployed a large-scale GPS network in the public traffic system. Currently, more than 1000 buses and 5000 taxis have been equipped with GPS terminals, which transmit GPS dada every 18 seconds to ShanghaiGrid through a satellite. Moreover, the *Data Aggregation service* is deployed in ShanghaiGrid to gather other traffic data from multiple distributed data sources such as traffic lights, sensors and video cameras.

We have developed an intelligent urban traffic management system based on the ShanghaiGrid system software. In the first stage, the traffic management system focuses on traffic guidance, *e.g.*, *road status report* (free or jammed), *the best path prediction* (to an specified destination) and *remaining time estimation* (how long the next bus is going to arrive at a stop), which are presented in this section.

4.1 Traffic Management System Based on ShanghaiGrid

Traffic management system comprises a set of applications built on ShanghaiGrid system software SHGSS. We illustrate the traffic management system and its work flow by the road status report (RSR) application, as shown in Fig. 6, where the GPS Data belongs to ShanghaiGrid Infrastructure. RSR initiates a computing process every five seconds in the following steps: (1) RSR initializes a road status computing, requests and gets current traffic data from Data Receiver module; (2) RSR transfers computing task and input parameters to Task Manager; (3) Task Manager requires Task Scheduler to allocate grid resources; (4) Task Scheduler queries Information Center, and allocates resources with light load to the task; (5) Task Scheduler returns assigned results to Task Manager, while Resource Monitor updates the assigned resource information in Information Center; (6) Task Manager transfers scheduling results to Service Matcher module; (7) Service Matcher actually allocates the task to a group of grid services, where road status information is actually computed in parallel; (8) Service Matcher collects computing results and transfers them to Task Manager; (9) Task Manager drives Data Publisher module to publish current road status; and (10) Data Publisher displays road status information on e-displays.

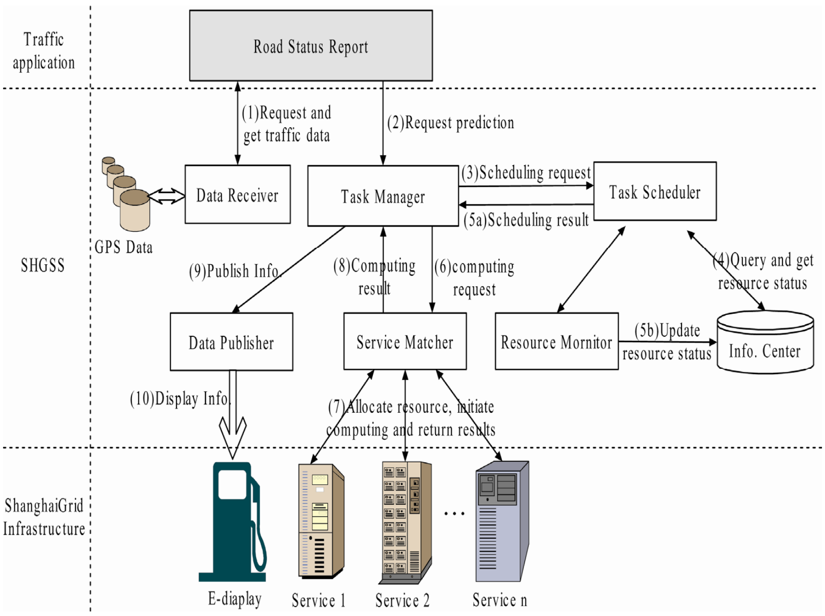


Fig. 6. Traffic management system based on the city grid ShanghaiGrid.

4.2 Real-Time Road Status Report

The road status denotes the average traffic flux in a road. The bigger the traffic flux in a road, the more crowded the road and the slower the traffic moves. The GPS data carries the location and speed information of vehicles, from which we can compute the average speed of vehicles in a road. As an example, Fig. 7 shows the real-time roads' status of Shanghai at a time, where a line represents a road and different colors indicate different traffic statuses. More specifically, roads colored in red, yellow and green are congested, usual and free respectively.

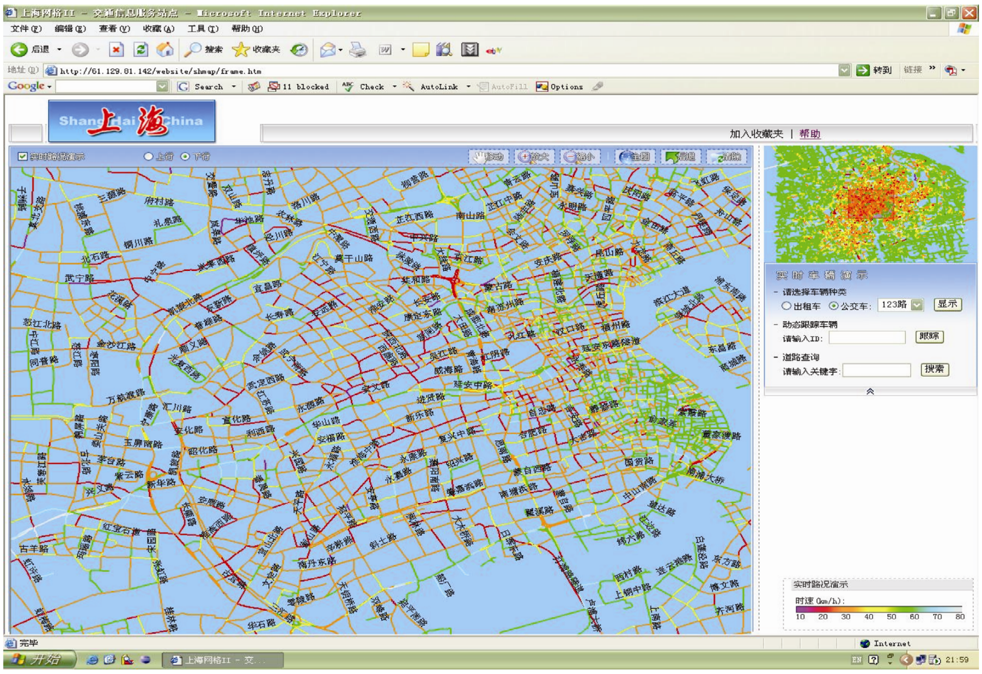


Fig. 7. Real-time traffic roads' status in Shanghai.

Through this application, users can quickly know general traffic status. If the user clicks on one road, the map also shows the detailed traffic information. When this application is deployed, the traffic information will be shown in public e-displays and users can also access it by browsers and PDAs.

4.3 The Best Path Prediction

The best path refers to a sequence of roads from one place to a specified destination, with a minimal cost dependent on different prediction policies. The best path between two specified places may change from time to time. For example, the best path based on the least time policy is different at different time due to changing road status.

The Best Path Prediction service finds the best path to enable people travel in Shanghai more conveniently. Currently, this service supports the least time policy and the least fee policy. Fig. 8 illustrates a recommended route that needs the least time to travel from Huaihai Road to Liyang Road, with a sequence of road names at the top left corner. Users can access this service using Web browsers or mobile phones. For a mobile phone user, the service returns a Short Message Service (SMS) showing a sequence of road names.

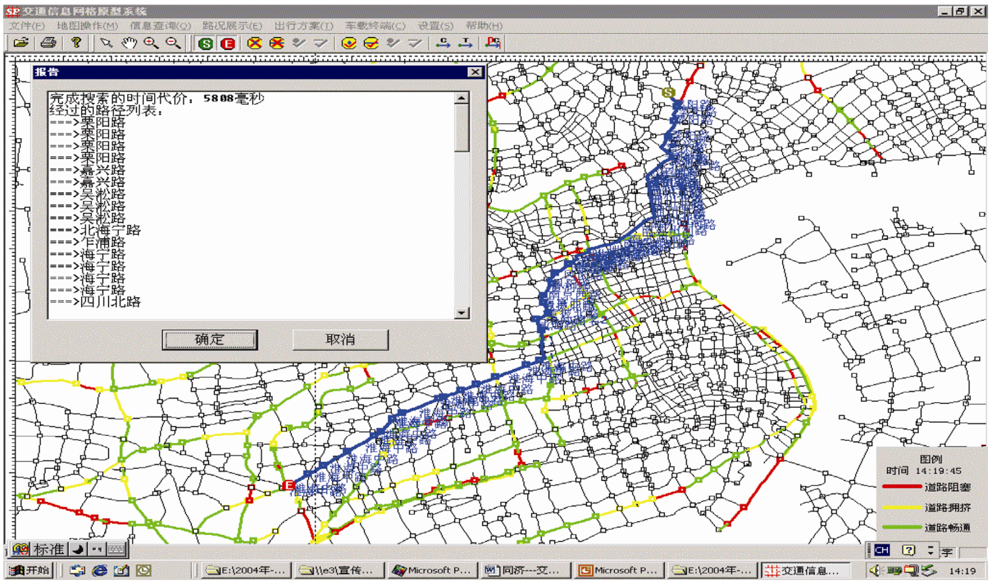


Fig. 8. A recommended best route colored in blue.

4.4 Remaining Time Estimation

This service computes the estimated remaining time, *i.e.*, how long the next bus will arrive at a stop, which synthetically considers real-time roads' status, intersections, direction changes of vehicles and so on. It will be available in bus lines 71 and 72 soon, where all the buses are equipped with GPS terminals.

5. TRAFFIC DATA COLLECTION USING WIRELESS SENSOR NETWORKS

Our intelligent traffic management system mainly involves two issues: real-time traffic data collection and real-time data processing presented above. Currently, data collection in ShanghaiGrid mainly relies on GPS systems. However, we argue that the wireless sensor network is a better substitute for sensing traffic data from various vehicles because: (1) GPS data is often lost owing to high building, bad weather and other disadvantageous factors, and (2) many drivers especially taxi drivers dislike using GPS devices owing to the high cost and inconvenience of GPS terminals. This Section will present a self-organized sensor network model used to collect real-time traffic data from buses and a protocol for routing traffic data to ShanghaiGrid system.

5.1 Sensor Network Model

There are hundreds of bus lines in Shanghai. Each bus line has hundreds of buses, associated with a scheduling office. Each bus stays at the scheduling office for a few minutes before the next round, during which it is the bus closest to the scheduling office.

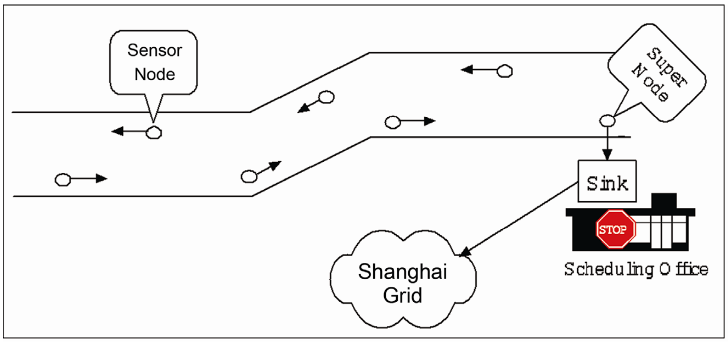


Fig. 9. Sensor networks for collection of bus information.

We assume to deploy sensors in each bus. A bus line forms a sensor network with a number of ordinary sensor nodes, a super sensor node and a sink, as shown in Fig. 9. All sensor nodes have the capabilities to observe, temporarily store and route data back to the *sink* by a multihop routing mechanism. Moreover, a super node acts as a gateway of a sensor network. The sink may communicate with the ShanghaiGrid system via Internet or satellite. Sensor nodes do not forward data from other sensor networks through checking their bus line numbers. Moreover, sensor nodes in the same network only route data from nodes further from the super node.

Energy consumption of a sensor node significantly increases with transmission distance. In our target environment, the topology of a sensor network for sensing traffic data changes very frequently because sensors keep continuously mobile. In order to maximize lifetime of a network, we assume: (1) in a sensor network, only one special node (called *super node*) closest to the *sink* can directly transmit data to the *sink* in a hop; (2) sensor can directly communicate only with neighbor nodes to reduce its energy consumption; (3) each sensor node only relays data from nodes farther from the *super node* in distance; and (4) all nodes work as the *super node* in turn by elections. After an election, the sensor network keep a stable topology for a period of time. During this period, the *super node* forwards data for all nodes to the *sink*.

Sensors deployed in the same bus line self-organize into a network by an election. A sensor network in a bus line L can be modelled as SN(L) = {Sink, Super, S, D}, where the Sink is a communication node deployed in the scheduling office, and connected to ShanghaiGrid system, Super is the super node, $S = \{S_1, S_2, \ldots, S_n\}$ is a sensor set and $D = \{D_1, D_2, \ldots, D_n\}$ is a sensing data set. Sensor S_i ($i = 1, 2, \ldots, n$) is deployed on the bus B_i. It collects data D_i all the time and transmits D_i to the sink node every 18 seconds. Note that the value of D_i changes continuously.

5.2 Election of a Super Node

Election of a super node occurs periodically. Whenever a sensor node detects that a signal from the sink exceeds a threshold, it initiates an election to run for a super node. We call such a sensor node as a candidate super node. In fact, the super node from a successful election is that the closest to the sink. Election includes the following two phase.

Initiation of an election. A candidate super node sends an ELECTION message to its two neighbors. The message at least contains a node name N_0, a bus line number and an election time.

Promulgation of the ELECTION message. After a sensor node receives an ELECTION message from the node N_i with the same direction, it changes the node name into N_{i+1}, then relays the message to further nodes. This process continues until the end of this bus line.

An election forms a hierarchical graph with two nodes in every level except the super node N_0, as shown in Fig. 10, where a solid line between two sensor nodes denotes a hop transmission. The two nodes with the same name N_i move towards reverse directions. Note that a node with the name N_i only routes an ELECTION message from nodes with the name N_{i-1}.

5.3 Routing Traffic Data

Routing traffic data is based on the topology generated from an election, where each sensor node associates with a logical name N_i. Between two elections, data transmission can happen many times.

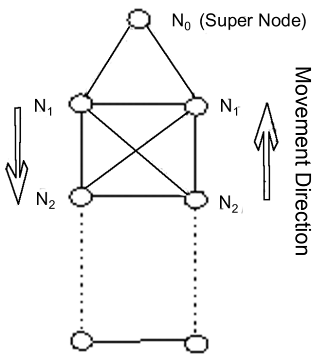


Fig. 10. The network topology after an election.

Input: traffic data D_j
Output: transmission of the D_j
N_i receives traffic data D_j from N_j
If N_j in S // transmit data from its own network
  If $j > i$ // transmit data for farther nodes
    If N_j moves to the same direction
      broadcast the data D_j
    EndIf
  EndIf
EndIf

Fig. 11. Routing algorithm for the node N_i.

Sensor nodes route traffic data using the directed routing algorithm (see Fig. 11), where S is a set of nodes in a sensor network, N_i represents a sensor node with a logical name N_i generated from the last election. In our routing algorithm, a sensor node only relays traffic data from its own sensor network. It can be realized by checking the number of bus line (LineID) in the data D_j because data sensed from different bus lines carries different LineIDs. As a result, each sensor network only reports the traffic data from its own bus line. In addition, a sensor only routes the data from the node that moves to the same direction and is farther from the super node than itself, which reduces redundant communication.

6. CONCLUSIONS AND FUTURE WORK

ShanghaiGrid is an excellent research and development platform for grid applications. Based on this infrastructure, we have set up an intelligent urban traffic management system, providing the traffic guidance services for people. Although we currently focus on traffic applications, we can extend the ShanghaiGrid system software to support other grid applications, leaving its overall structure and existing components unchanged. This would basically require enriching application-specific components in the application support layer.

In our future work, we will further enhance the traffic management system for traffic-decision support and traffic-congestion control for Shanghai government. Moreover, we plan to extend the ShanghaiGrid system software to support other significant grid applications, focusing on real-time computing, multiagent coordination, as well as security and other issues. As a final result along this direction, we could get a general grid system software that can support more applications.

ACKNOWLEDGMENT

Feilong Tang would like to thank The University of Aizu (UoA), Japan for the excellent research environment during his visiting research in UoA. Thanks are also given to Professor Wanlei Zhou in Deakin University, Australia, Professor Zixue Cheng and Dr. Song Guo in UoA, Japan for their precious helps.

REFERENCES

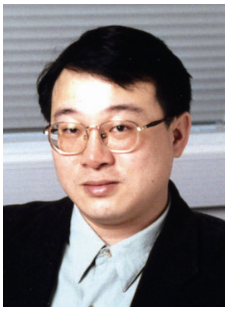
1. I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "Grid services for distributed system integration," *IEEE Computer*, Vol. 35, 2002, pp. 37-46.
2. J. Nichols, H. Demirkan, *et al.*, "Autonomic workflow execution in the grid," *IEEE Transactions on Systems*, *Man and Cybernetics*, Vol. 36, 2006, pp. 353-364.
3. J. Bailey, A. Poulovassilis, and P. T. Wood, "Analysis and optimisation of event-condition-action rules on XML," *Computer Networks*, Vol. 39, 2002, pp. 239-259.
4. Y. G. Wang, H. Jin, M. L. Li, *et al.*, "Grid computing in China," *Journal of Grid Computing*, Vol. 2, 2004, pp. 193-206.
5. Y. W. Wu, J. Liu, G. Chen, *et al.*, "Adapting to application workflow in processing data integration queries," in *Proceedings of the 6th International Conference on Grid and Cooperative Computing*, 2007, pp. 745-748.

6. F. Fox and D. Gannon, "Computational grids," *Computing in Science and Engineering*, Vol. 3, 2001, pp. 74-77.
7. Y. W. Wu, S. Wu, H. S. Yu, and C. M. Hu, "CGSP: an extensible and reconfigurable grid framework," in *Proceedings of the 6th International Workshop on Advanced Parallel Processing Technologies*, 2005, pp. 292-300.
8. J. Bae, H. Bae, *et al*., "Automatic control of workflow processes using ECA rules," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, 2004, pp. 1010-1023.
9. J. Feng, L. L. Cui, G. Wasson, and M. Humphrey, "Toward seamless grid data access: design and implementation of GridFTP on .NET," in *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, 2005, pp. 164-171.
10. F. L. Tang, M. L. Li, and J. Huang, "Automatic transaction compensation for reliable grid applications," *Journal of Computer Science and Technology*, Vol. 21, 2006, pp. 529-536.
11. S. Y. Crompton, B. M. Matthews, W. A. Gray, *et al*., "OGSA-DAI and bioinformatics grids: challenges, experience and strategies," in *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid*, 2006, pp. 193-200.
12. F. L. Tang, M. L. Li, and J. Z. Huang, "Real-time transaction processing for autonomic grid applications," *Engineering Applications of Artificial Intelligence*, Vol. 17, 2004, pp. 799-807.
13. S. Dalal, S. Temel, and M. Little, "Coordinating business transactions on the Web," *IEEE Internet Computing*, Vol. 7, 2003, pp. 30-39.
14. H. Garcia-Molina and K. Salem, "SAGAS," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, Vol. 16, 1987, pp. 249-259.
15. J. Gray, "The transaction concept: virtues and limitations," in *Proceedings of the 7th International Conference on VLDB*, 1981, pp. 144-154.

Feilong Tang (唐飞龙) graduated with a Ph.D degree in the Department of Computer Science and Engineering of Shanghai Jiao Tong University (SJTU), China. Now, he works with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests focus on grid computing, Web services, wireless sensor network, and distributed computing.

Minyi Guo (过敏意) received his Ph.D. degree in Computer Science from University of Tsukuba, Japan. He is now a full professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include parallel and distributed processing, parallelizing compilers, pervasive computing and software engineering. He is a member of the ACM, IEEE, IEEE Computer Society, and IEICE.

Minglu Li (李明禄) is a full professor and a deputy director in Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He also is a director of Grid computing center of Shanghai Jiao Tong University, and expert-in-chief of ShanghaiGrid project. His research interests mainly include grid computing, Web services and multimedia computing.



Cho-Li Wang (王卓立) received his Ph.D. degree in Computer Engineering from University of Southern California in 1995. He is currently an associate professor with the Department of Computer Science at The University of Hong Kong. Wang's research mainly focuses on the system software for Cluster/Grid Computing. Dr. Wang serves as an editorial board member of IEEE Transactions on Computers, International Journal of Pervasive Computing and Communications, and Multiagent and Grid Systems.