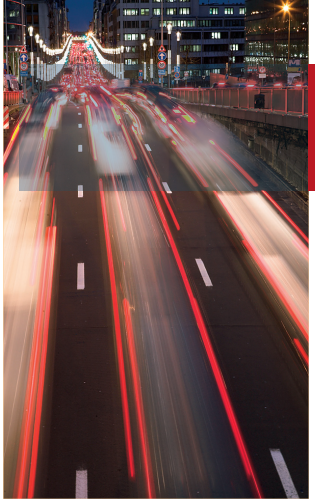

Editor: **Lefei Li,** Tsinghua University, lilefei@tsinghua.edu.cn

Artificial Societies and GPU-Based Cloud Computing for Intelligent Transportation Management

Kai Wang, *National University of Defense Technology and Chinese Academy of Sciences*
Zhen Shen, *Chinese Academy of Sciences*

Many complex systems such as transportation, stock, and economic systems involve human social behaviors. It is challenging to establish accurate mathematical models for such systems, and experiments on them are generally costly or even impossible.[1–7] These issues make it difficult to analyze, control, and manage complex systems.

To tackle these problems, Fei-Yue Wang proposed the ACP approach, which consists of three steps:

1. modeling and representation using artificial societies,
2. analysis and evaluation by computational experiments, and
3. control and management through parallel execution of real and artificial systems.

This approach has shown its advantages for transportation system control and management.[1]

The concept of the artificial societies was proposed based on the theory of artificial life and the society simulation in the 1990s.[5,6] Agent technologies become an effective tool of artificial societies[1,7] because they can take into consideration all the necessary elements of society and give detailed descriptions of complex system behaviors.

For the C part of the ACP approach, researchers usually need to evaluate many configurations and scenarios, and it is necessary to calculate the system's evolution for every configuration and scenario, which makes the computing burden very heavy. One method to address this difficulty is to use cloud computing to trade computation for "intelligence." The clouds can provide the powerful computing and storage capabilities necessary in a distributed and flexible way.[8] The decision support can be provided as services so end users can benefit from conveniences, while the details of computing are hidden in the clouds. Because similar types of agents share similar behaviors, a parallel mechanism, such as cloud computing's virtual clusters, can be used.

Moreover, computing hardware is going through a revolution with the development of graphics processing units (GPUs). A single GPU can run many threads together and is suitable for parallel computing. To successfully solve the control and management problems of complex systems, researchers have employed iterative algorithms such as genetic algorithms (GAs) and simulated annealing (SA). The development of the CPU made such iterative algorithms popular, and we believe that the GPU will enable iterative, parallel algorithms to prevail. In fact, GAs, SA, and some other iterative algorithms have already been modified to be GPU-compatible. Because it is now possible to simulate agent behaviors and run optimization algorithms in parallel, it is promising to combine GPUs into the compute clouds to provide services for the control and management of the complex systems. With the GPU-based cloud computing as the headquarters, commanding the real world can become faster and easier.

1541-1672/11/$26.00 © 2011 IEEE
Published by the IEEE Computer Society

In this article, we focus on the C part of the ACP approach. We explain the advantages of cloud computing and GPUs and present the architectures of GPU-based cloud computing for transportation systems.

Cloud Computing

Cloud computing is derived from the concepts of grid computing, distributed computing, and parallel computing. It uses a computer network to provide computing resources such as data or software as a service to users who pay for it on demand. Thus, it breaks free of the computing power and storage space limitations of the traditional local computing model.[8] Users' computers, mobile phones, and other personal devices might only contain an operation system and an Internet explorer, and they do not need to know where the data is stored or who provide the software. Users submit their computing tasks that cannot be accomplished by their local devices to the clouds. The clouds provide computing services and return the computing results to them.

Cloud computing benefits from several key characteristics:

- *Scalability*. In the cloud computing framework, computing resources can be increased or decreased in response to the users' different application loads.
- *Reliability*. The data is stored and the applications are running on the servers in the clouds. Users do not have to worry about lost or corrupt data.
- *Agility*. The clouds can distribute computing resources according to the users' needs or preferences to provide flexible management.
- *Utility*. Users do not have to buy expensive computing devices. They only need to pay for the computing services provided by the clouds.

Cloud computing provides a platform for computational experiments with abundant computing and storage resources. The system can be considered as a whole and the control and management decisions are sent as services to agents.

What Is a GPU?

Graphics device companies developed GPUs as a specialized circuit to accelerate the process of building computer graphics on computer monitor screens. GPUs are widely used in personal computers, mobile phones, computer workstations, and embedded systems. They have distinguished parallel computing ability due to their highly parallel structure with many cores working together to process large volumes of computer graphics pixels. Because of this, researchers began to use GPUs for scientific computing. However, they had to map their applications into problems that draw graphs and program with graph programming languages such as OpenGL and Cg.

Cloud computing provides a platform for computational experiments with abundant computing and storage resources.

Nvidia realized the potential to use GPUs for general-purpose computing and developed the General-Purpose GPU (GPGPU) and Compute Unified Device Architecture (CUDA). With CUDA, researchers can develop GPU parallel applications easily with high-level languages such as C, C++, and Fortran.

A GPU is the core of the computer's display adapter and is controlled by the CPU. There are many cores working in parallel, and they are called streaming processors (SPs). Several SPs constitute a streaming multiprocessor (SM). Each SM has its own shared memory, and all the SMs in a GPU share its global, constant, and texture memories. A typical program written in C using a GPU consists of CPU and GPU codes. The CPU codes control the process of the whole program, and the GPU codes do the parallel computing work. A function that executes on the GPU is typically called a kernel. When a kernel is launched, the threads on a GPU organized by two levels are activated. The higher level is called the *grid*, and the lower is called the *block*. One grid can consist of at most $65{,}535 \times 65{,}535$ blocks, and each block can consist of at most 512 threads. The grid is assigned to the GPU with blocks assigned to the SMs and the threads assigned to the SPs.

Figure 1 shows the basic principles of GPU parallel computing. With the SMs working together, the GPU generates excellent performance in processing parallel tasks. The GPU has been applied widely in fields such as computational structural mechanics, bioinformatics, computational finance, imaging, and computer vision.

GPUs in Cloud Computing

Currently, most cloud providers take computer server clusters as the core devices. Nevertheless, for parallel tasks with heavy computational burdens, GPUs usually can accelerate

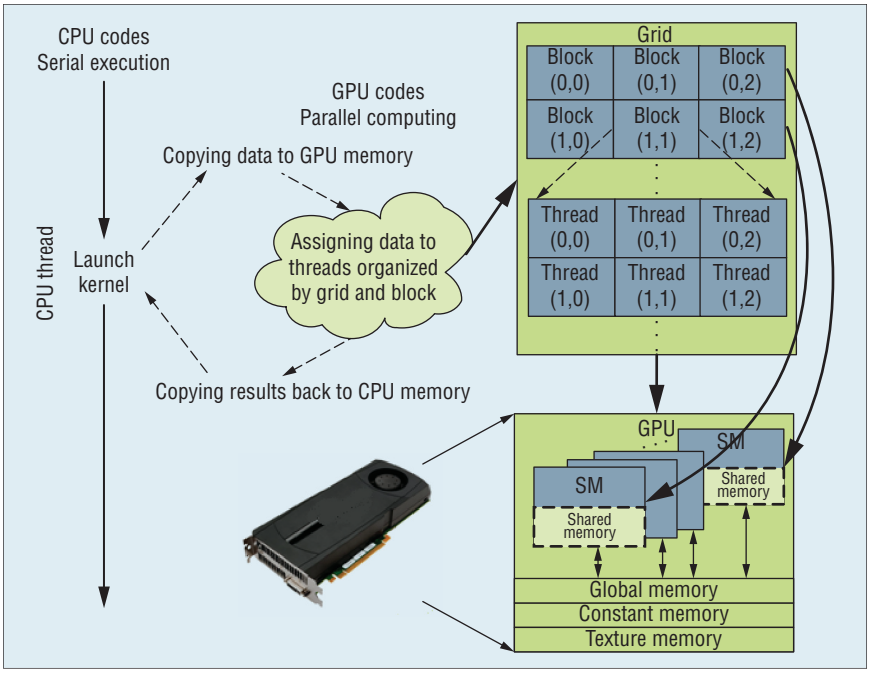

Figure 1. Graphics processing unit (GPU) parallel computing. With the streaming multiprocessors (SMs) working together, the GPU generates excellent performance in processing parallel tasks.

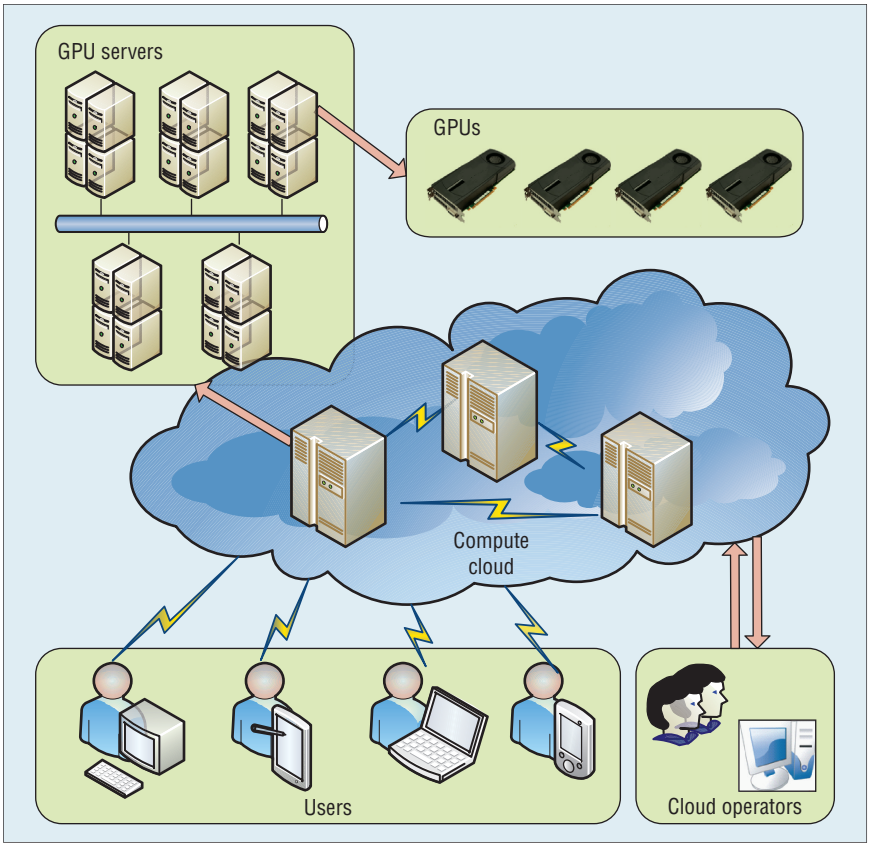

Figure 2. GPUs in cloud computing. To deliver the same performance using only CPUs would require much more resources.

computing effectively. One example is the Tianhe-1A, currently the second-fastest (taken over by the K computer in Jun. 2011) supercomputer in the world. It uses 7,168 Nvidia Tesla M2050 GPUs and 14,336 Intel Xeon CPUs. It would require more than 50,000 CPUs to deliver the same computing performance, and the power consumption would increase from 4.04 megawatts to more than 12 megawatts (see www.nvidia.com/object/tesla_computing_solutions.html).

The GPU has already been used in cloud computing (see Figure 2). Nvidia released the cloud computing platform RealityServer based on its Tesla GPU servers. The platform can provide 3D graphics rendering Web services to product designers, architects, and consumers around the world (see www.nvidia.com/object/realityserver_webapps.html). Amazon provides GPU resources for general-purpose computing in its Elastic Compute Cloud (Amazon EC2) (see http://aws.amazon.com/ec2).

From Traffic Simulations to Artificial Transportation Systems

Limited by the computing power of early computers, initial traffic flow simulation systems tended to be macroscopic or mesoscopic based on hydromechanics or statistical physics. The Lighthill Whitham Richards (LWR) method and the Lattice Boltzmann Method (LBM)[7] are two typical methods. They are good at describing the overall properties of a traffic system and require little computational resources and computer storage space. However, they are not flexible enough to describe the microscopic traffic behaviors, such as passing vehicles or bicycles, lane changing, and various individual pedestrian behaviors.

With the development of computer simulation technologies in the 1960s, traffic simulations gradually became an important tool in the area of transportation engineering to help design, plan, and control traffic systems. The typical methods were the car-following model, cellular automata (CA), and the Social Force (SF) model.[7] However, vehicles in these models share the same rules for updating their states (position, speed and acceleration), as do bicycles and pedestrians.

In the last three decades, with the development and deployment of intelligent transportation systems (ITS)[1] and the wide application of multicore CPUs and computer clusters, multiagent system (MAS) modeling has become increasingly important in traffic simulations. Agents in an MAS are autonomous, collaborative, and reactive, as they can communicate, compete, and cooperate with each other. Vehicles, bicycles, and pedestrians are modeled as different kinds of agents, so critical macroscopic traffic phenomena can be observed and analyzed as they interact with each other.

In recent years, researchers have realized that it is not enough to focus on the traffic systems alone to solve traffic problems.[1,3] Some other metropolitan systems, such as the logistic, infrastructure, legal and regulatory, weather, and environmental subsystems should also be taken into consideration. Thus, the Artificial Transportation System (ATS)[1] concept was proposed based on the idea of artificial societies and agent technologies. Figure 3 shows the framework of the ATS.

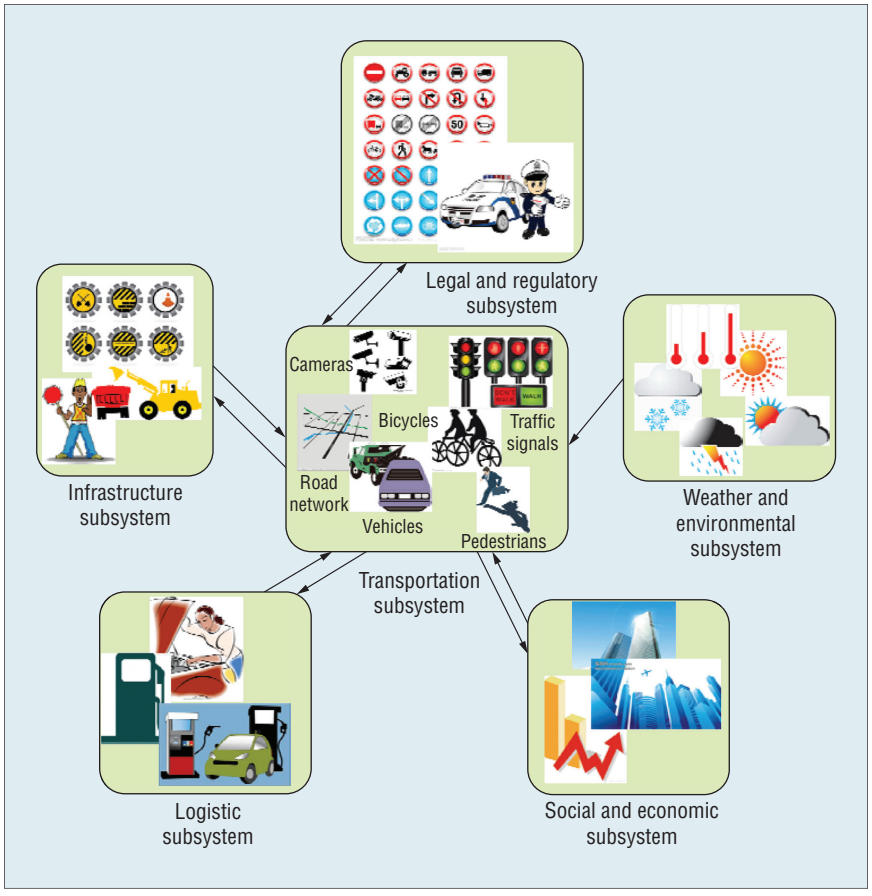

Figure 3. Artificial transportation system (ATS) framework. In addition to vehicles, bicycles, and pedestrians, an ATS incorporates weather, legal, and other social environments.

ATS and Challenges in Computing

An ATS must deal with a range of information and activities, such as weather, legal, and other social environments, besides transportation. In Beijing for example, the number of vehicles reached up to 4.89 million by April 2011 and the resident population reached approximately 20 million by the end of 2010.[1] Correspondingly, Beijing's ATS must deal with millions of agents in the transportation system to calculate the state evolution of each agent. Moreover, the optimization of the transportation system with the ATS often involves algorithms that need to evaluate the system many times, such as GAs, which are reported to work well for traffic signal timing and demand optimization problems.[9] With the burdens from computing agent behaviors and the algorithms' evaluating process, the computing task becomes a bottleneck that prevents more practical applications of the ATS.

Architectures of GPU-Based Cloud Computing for ATS

We believe that ATS can benefit from the architectures of GPU-based cloud computing. Here we give the architectures for the transportation systems first and then show how to consider all the subsystems in the ATS as a whole.

In the transportation subsystem, the agents' data can be organized from top to bottom in five levels: road network, intersections, roads, lanes (bicycle paths and sidewalks), and agents. The agents in a lane are organized into a queue, and the lanes going toward an intersection are organized as "belonging to" the intersection.

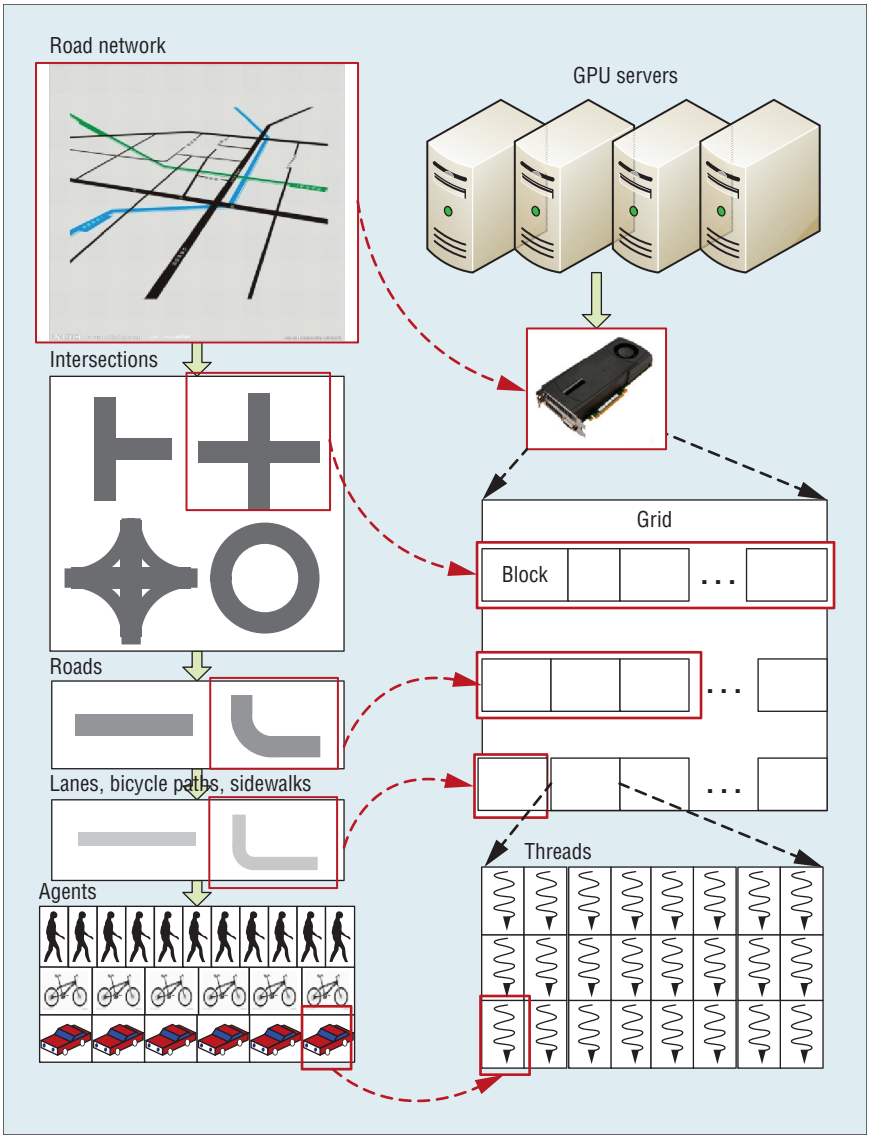

Figure 4. Hierarchical structures and the computing resources allocation. Each row of blocks in the grid handles the computing task for one intersection in the road network and each thread handles a single agent on the lane.

Figure 4 shows how each row of blocks in the grid handles the computing task for one intersection in the road network and how each thread handles a single agent on the lane.

Currently, one Nvidia GPU server can have several Tesla GPUs in it and can provide more than 100 Gbytes of system memory. The GPU server, N4282GPU, is equipped with 8 Nvidia Tesla M2050 GPUs along with two Intel Xeon 5600 processors. Its system memory is 144 Gbytes. Moreover, with the latest version of CUDA (CUDA 4.0), programmers can use all the GPUs in the system concurrently from a single CPU thread and transfer data between the GPUs in a peer-to-peer mode without the help of the CPU as a transfer station. In this situation, a GPU server is enough for the traffic simulation of a large area or even a city. Computing clouds with many GPU servers can be powerful enough to control and manage a large ATS.

GPUs have been used for traffic simulation, and previous research reported a speedup of 67 compared with highly optimized Java version.[10] We go further to use GPUs to accelerate both the traffic simulation and the optimization. As an example and preliminary work, we use GA to optimize the traffic signal timing of a road network consisting of four intersections to maximize the number of vehicles leaving the road network in a given period. We set the population number in the GA to 500—that is, 500 traffic signal timing configurations. We simulate 3,600 seconds for each generation.

We implemented both a CPU alone (an AMD AthlonTM 64 X2 dual-core processor 4000+) and the CPU plus GPU methods. The results show that a single generation of a GA takes approximately 3,720 seconds on the CPU alone on average. The CPU and GPU working together, however, take only 19 seconds on average, for a speedup of 195. For 1,000 generations, the GA takes 19,044 seconds on the GPU-based method. This would take the CPU alone 43.1 days! The road network within the Second Ring Road in Beijing consists of 119 intersections.[8] The GPU-based clouds can have a much stronger computing power than the clouds with only CPUs and are much more qualified for the traffic optimization of a large road network such as Beijing.

Figure 5 presents an overview of the traffic control and management with the GPU-based cloud computing.

To model the influences of other subsystems on the transportation subsystem in the ATS, we can build data support centers to provide necessary information to the transportation subsystem online or offline. The centers collect the weather, geographic information system (GIS), legal and regulatory, facilities, and

population information. Moreover, we can obtain a large amount of travel demand data through personal diary surveys and analysis of individual transport behaviors based on GPS data. With the road network, facility, and population data provided by the ATS subsystems, we can fuse different kinds of data to generate complete daily travel plans for every agent in the ATS's transportation subsystem.

For demand optimization, users can submit their travel demand to the clouds by devices such as personal computers, mobile phones, and pocket PCs before they travel. The clouds can take the users as special agents and add them to the ATS transportation subsystem to evaluate various travel routes. The clouds can then provide detailed daily travel plans to the users to guide them to their destinations with minimal delays while balancing the traffic load of the whole road network.

For the traffic signal timing optimization, the clouds can adjust the cycle time, offset, and green splits of the traffic lights in the road network based on the traffic condition in the ATS. We can allocate different traffic signal timing configurations to different GPUs in the clouds that evaluate and optimize the configurations using computational experiments.

Adding GPUs to the compute cloud is just like the Chinese idiom "adding wings to a tiger." A much faster system for the intelligent transportation management can be obtained. Guided by the ACP approach, we aim to build a practical GPU-based cloud computing system to provide transportation managers and participants more convenient and faster service.

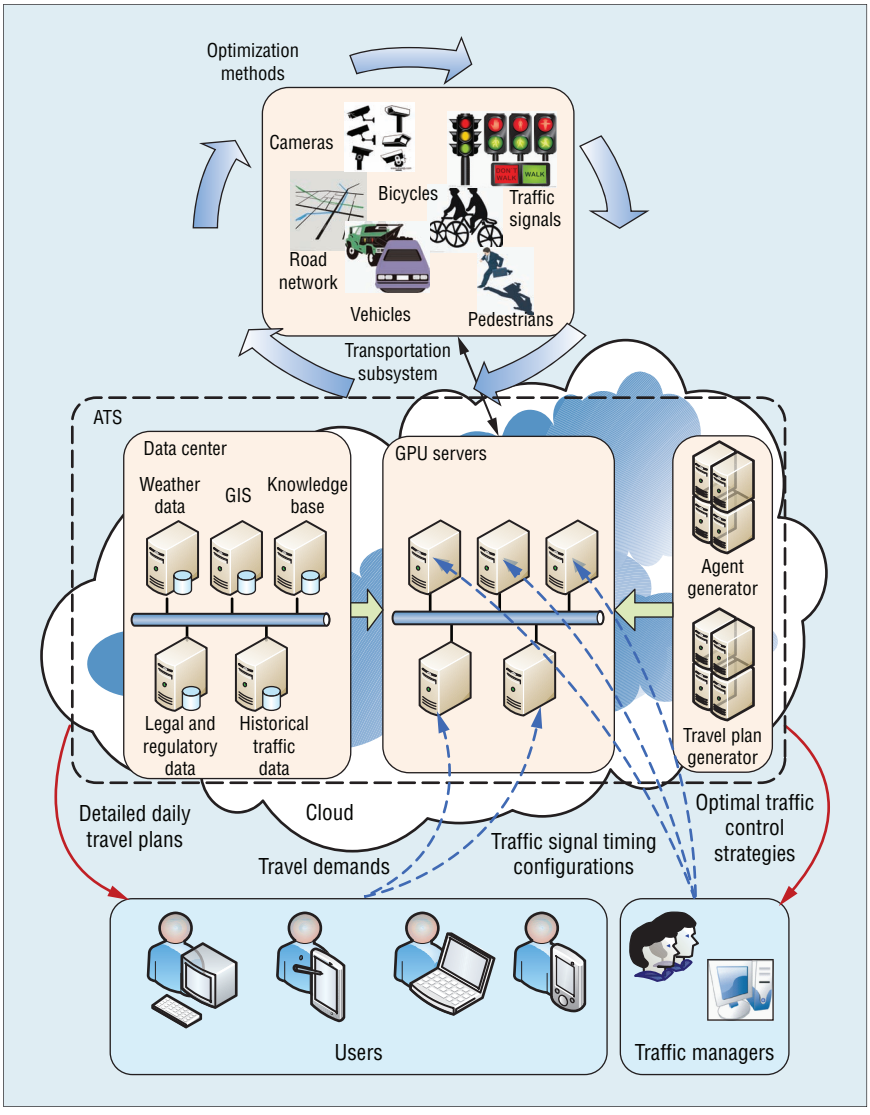

Figure 5. Intelligent traffic management. ATS and GPU-based cloud computing enable great improvement over traditional traffic simulation systems.

Acknowledgment

This work is supported in part by NSFC grant numbers 70890084, 60921061, and 90920305 and CAS grant numbers 2F09N05, 2F09N06, 2F10E08, and 2F10E10. We thank EIC Fei-Yue Wang for helpful and insightful discussions.

References

1. F.-Y. Wang, "Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 3, 2010, pp. 630–638.
2. F.-Y. Wang, "Artificial Societies, Computational Experiments, and Parallel Systems: An Investigation on Computational Theory of Complex Social Economic Systems," *Complex Systems and Complexity Science*, vol. 1, no. 4, 2004, pp. 25–35.
3. F.-Y. Wang, "Parallel System Methods for Management and Control of Complex Systems," *Control and Decision*, vol. 19, no. 5, 2004, pp. 485–489.
4. F.-Y. Wang and J.S. Lansing, "From Artificial Life to Artificial Societies—New Methods for Studies of Complex Social Systems," *Complex Systems and*

Complexity Science, vol. 1, no. 1, 2004, pp. 33–41.
5. N. Gilbert and R. Conte, *Artificial Societies: The Computer Simulation of Social Life*, UCL Press, 1995.
6. M. Sipper, "Studying Artificial Life Using a Simple, General Cellular Model," *Artificial Life*, vol. 2, no. 1, 1994, pp. 1–35.
7. B. Chen and H.H. Chen, "A Review of the Applications of Agent Technologies in Traffic and Transportation Systems," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 2, 2010, pp. 485–497.
8. Z.-J. Li, C. Chen and K. Wang, "Cloud Computing for Agent-Based Urban Transportation Systems," *IEEE Intelligent Systems*, vol. 26, no. 1, 2011, pp. 73–79.
9. J.J. Sánchez-Medina, M.J. Galán-Moreno, and E. Rubio-Royo, "Traffic Signal Optimization in 'La Almozara' District in Saragossa under Congestion Conditions, Using Genetic Algorithms, Traffic Microsimulation, and Cluster Computing," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 1, 2010, pp. 132–141.
10. D. Strippgen and K. Nagel, "Using Common Graphics Hardware for Multi-agent Traffic Simulation with CUDA," *Proc. 2nd Int'l Conf. Simulation Tools and Techniques*, Brussels, 2009, pp. 1–8.

Kai Wang is a PhD candidate at the Center for Military Computational Experiments and Parallel Systems Technology, College of Mechatronics Engineering and Automation, National University of Defense Technology (NUDT), and the State Key Laboratory for Intelligent Control and Management of Complex Systems, Institute of Automation, Chinese Academy of Sciences. Contact him at kai.wang_nudt@hotmail.com.

Zhen Shen is an assistant professor at the State Key Laboratory for Intelligent Control and Management of Complex Systems, Institute of Automation, Chinese Academy of Sciences. Contact him at zhen.shen@ia.ac.cn.