Dynamic Task Scheduling of Parallel Processing for Massive FCD

Feng Chen, Gang Xiong

Abstract— **Floating Car Data (FCD) technique is new way for real-time traffic flow collection of large-scale urban network. The dynamic guidance and control of urban traffic need to process massive FCD quickly. In this work, the characteristics of FCD are experimentally investigated, and a dynamic task scheduling algorithm is first proposed for FCD parallel computation. In allusion to the uncertainty and dynamics of FCD, a dynamic task partition method is introduced to divide FCD packages properly. The load balance among computing nodes can be achieved using the dynamic task allocation strategy. Our algorithm is developed on high performance parallel computer KD-50 platform. The field FCD of Beijing is used to test this proposed algorithm. The experimental results illustrate our algorithm has high parallel performances for massive FCD processing.**

I. Introduction

Large-scale traffic flow collection is the requisite of urban traffic guidance and active control. Traditional traffic flow collection methods such as indicator loop, microwave, radar, and video are of fixed location, which can merely offer section data of road segments. Floating car data (FCD) is an emerging technique for large-scale traffic flow collection. In FCD method, a number of vehicles equipped with global positioning system (GPS) are utilized to collect traffic flow data of road network. In nature, these vehicles constitute a mobile sensor network. Each floating car transmits its longitude, latitude, time, instantaneous velocity, and heading information to remote monitoring center through wireless communication network at a certain time interval. In remote monitoring center, the FCD packages are processed to acquire real-time traffic conditions of urban road network. Therefore, the underlying data can be provided for urban traffic guidance, control, and management.

FCD has received considerable attention in recent years. In application aspect, some traffic dynamic guidance systems including ALI-SCOUNT (Germany), MAYDAY (USA), and P-DRGS (Japan) have been developed successfully [1-4]. The theoretical research of FCD mainly concentrates on selection and optimization of floating car numbers, OD estimation, and traffic incidence detection, etc [5-10].

Manuscript received August 15, 2011. This work was supported financially by the Anhui key Scientific and Technological project of "12th-five-year plans" under Grant 11010402081.

Feng Chen is with the Department of Automation, University of Science and Technology of China, Hefei, 230027 China (corresponding author: +86-551-3600917; fax: +86-551-3603244; e-mail: chenfeng@ustc.edu.cn).

Gang Xiong is with the Institute of Automation, Chinese Academy of Sciences, Beijing, 100190 China (e-mail: gang.xiong@ia.ac.cn).

In order to accurately model traffic flow of urban road network, a certain amount of folating cars (about 5% of total numbers of motor vehicle on the road) are needed for data collection. In view of the communication costs, the sample frequency of floating car is usually 1/15 or 1/30 Hz, while the time interval which floating car uploading data to remote monitoring center is set to 3 or 5 minutes. Thus, there are massive FCD packages needed to be processed. For instance, there are about 25,000 taxies served as floating cars in Beijing, which generate more than 2.5M byte FCD packages every 5 minutes.

The massive FCD must be handled and analyzed rapidly to achieve urban traffic guidance and control dynamically. The existing serial processing modes such as MAYDAY and P-DRGS paradigms can not meet high real-time requirements. Parallel processing of FCD is a inevitable choice. But up to now, research on parallel processing of FCD is still at an early stage, and little work has been carried out from a literature survey.

To effectively keep the processing delay within an acceptable level, Lu et al. developed a structure of distributed FCD processing system based on multiple servers [11]. Grouped FCD workload is assigned based on volume of each FCD group and computing capability of each server. However, the differences among calculation costs of FCD records are not taken into account. In addition, this method groups FCD based on its location so as to decrease the time of loading/unloading area based maps to/from memory. It leads inevitably to the increase of communication costs.

In literature [12], attention is focused on modeling the road situation with FCD based on principle curves. A parallel algorithm is introduced to build the models in a heterogeneous computing platform. The task scheduling policy of the algorithm assigns tasks of different sizes to processors with different speed such that all subtasks complete at the same time. But it is based upon the premises that the calculation costs of different FCD records are identical and the performances of servers are known. The network communication delays are neglected as well.

In this work, the characteristics of FCD are experimentally investigated, and a dynamic task scheduling algorithm is first proposed for parallel computation of FCD. In allusion to the uncertainty and dynamics of FCD package processing, FCD packages are partitioned dynamically. The load balance among computing nodes can be achieved using the dynamic task allocation strategy. Our algorithm is developed on high performance parallel computer KD-50 platform. The field FCD of Beijing is used to test this proposed algorithm. The experimental results illustrate our algorithm has high parallel processing performances.

This paper is organized as follows. In section Ⅱ, the characteristics of FCD are investigated and discussed, and then a parallel processing architecture is presented for massive FCD processing. In next section, a dynamic task partition method is introduced to address FCD package in view of the FCD characteristics. Dynamic task scheduling algorithm is proposed and described. Section Ⅳ presents the experimental results. This paper is concluded in final section.

II. PARALLEL PROCESSING OF FCD

A. FCD Characteristic Analysis

Each floating car percepts its surrounding traffic conditions with a certain sample frequency, and then generates FCD records. A FCD record mainly includes vehicle ID, coordinates, instantaneous velocity, heading and time fields. A floating car transmits its FCD package to remote monitoring center at a certain time interval. In remote monitoring center, FCD map-matching is implemented and vehicle trajectory is found by path-searching process. Further, dynamic traffic conditions of road network are obtained by the estimation of traffic flow parameters such as average velocity and density of road network.

The processing procedure of one FCD record includes map-matching, vehicle trajectory estimation, and calculating average velocity of road segments. Obviously, there are large differences of processing time among FCD records. We experimentally investigated FCD characteristics with Loongson 2F CPU platform. The instruction system of Loongson 2F CPU is analyzed. The processing time of different FCD records are obtained according to the worse-case execution time (WCET) principle. The maximal difference among FCD records is about 5 seconds. FCD characteristics are found and summarized in the following:

1) Dynamic and low sample frequency

It may lead to FCD record missing due to noise disturbance, so the sample interval between the consecutive FCD records is uncertain and dynamic.

2) Uncertainty of processing time for FCD record

Because of velocity differences among various floating cars and the dynamics of FCD sample intervals, it causes different path-searching range for calculating average velocities of road segments. There thus are quite large differences of the processing time for different FCD records.

B. Parallel Processing Architecture for FCD

In order to meet the real-time requirements of urban traffic dynamic guidance and control, we develop a parallel processing architecture for FCD (shown as Fig.1).

In this system, FCD collection server conducts data storage and management. Protocol conversion module transforms FCD record into XML document, and then transfers it to the master control server. The status of computing nodes is monitored by server running status monitoring module. In master control server, the XML document is parsed. FCD packages are regrouped by sorting vehicle ID. By considering the volume of FCD packages and current status of various computing nodes, FCD is partitioned into multiple smaller packages which are assigned to idle computing nodes dynamically.

Every computing node preprocesses data for filtering noisy data contained in FCD packages. It matches GPS point with the corresponding location in road segment. On this basis, the running trajectory of floating car is inferred. The average velocities of road segments on floating car trajectory are calculated and then returned back to master control server.

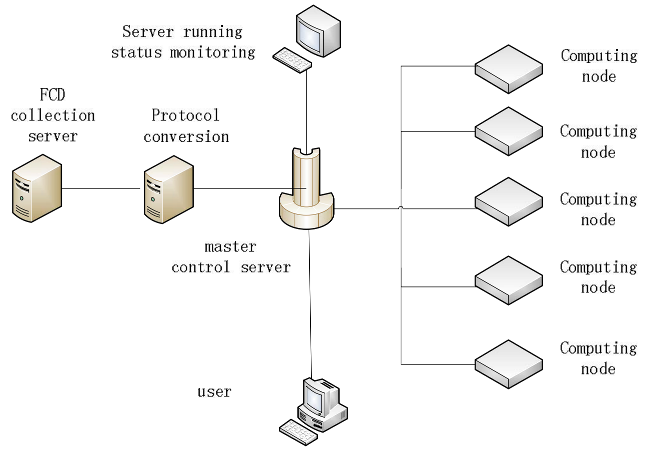

Figure 1. parallel processing architecture of FCD

After all the computing nodes finish the assignment tasks and return the results, master control server calculates the average velocities of various road segments in road network. It then estimates real-time density and volume of traffic flow according to flow-speed and speed-density models. Therefore, the necessary data can be provided for the dynamic guidance and control of urban traffic.

III. DYNAMIC TASK SCHEDULING FOR FCD PARALLEL PROCESSING

Because loosely coupled relations exist among FCD records, parallel computation method is suitable for FCD processing. Dynamic task scheduling is the core technique of FCD parallel processing. In this parallel processing system, master control server sorts all FCD records according to floating car ID. It conducts task partition and assignment dynamically based on idle computing nodes. The task assignment aims to achieve load balance among computing nodes. In order to avoid communication frequency among computing nodes, all data of a floating car is only assigned to a computing node.

Owning to environmental noise disturbances, raw FCD packages contain a certain amount of noise data. This thus may cause large difference of processing time between FCD packages with the same size. The synchronization of computing nodes can not be achieved through dividing whole FCD packages equally only relying on the number of idle computing nodes. It may lead to large time delay of the parallel computation system.

In view of FCD characteristics, we propose a dynamic task scheduling algorithm. In order to improve the performances of FCD parallel computation, our task partitioning method is described as follows.

Assume maximal processing time of a FCD record is $T_{\max}$, the number of idle computing nodes is C_{idle}, and the constraint time of FCD processing is T_{res}. Let the volume of FCD packages be M. FCD packages are partitioned initially into N_0 groups according to Equation 1.

$$N_0 = \mathrm{int}(\frac{M \times T_{\max}}{T_{res}}) + 1 \qquad (1)$$

Where int() is a rounding function. Obviously, the constraint condition that the parallel computation system completes the task of FCD processing within the specified time is $N_0 \le C_{idle}$. Thus, the initial grouping number is $N_0 = C_{idle}$.

There may be large difference of processing time for various FCD records. Master control server can not model traffic conditions of road network until all computing nodes return their results. Therefore, time delay of the parallel processing system increases, and utilization of computing nodes decreases as well. Thereby, it is necessary to partition FCD packages further using Equation 2..

$$N = N_0 \times \Delta \qquad (2)$$

Where Δ is maximal difference of processing time among FCD records, N is the number of FCD groups.

In this work, suppose that user-defined constraint time T_{res} can be satisfied. This proposed dynamic task scheduling algorithm is described as follows.

Step 1. Initializing arguments. Suppose $T_{\max}$ is maximal processing time of a FCD record, C_{idle} is the number of idle computing nodes. $C_{idle}(i)$ denotes the number of idle computing nodes in the ith iteration. T_{res} is the constraint time of FCD processing. The volume of FCD records is M. The number of FCD groups N can be obtained according to Equation 2.

Step 2. Uploading GIS map to C_{idle} computing nodes, C_{idle} FCD groups are assigned to these idle computing nodes. $M \leftarrow M - C_{idle} \times M / N$, $N \leftarrow N - C_{idle}$, iteration number $i = 1$

Step 3. While $(N > 0)$ do

{ Master control server monitors whether the results are returned from computing nodes or not.
If not results return then keep monitoring endif
if $N < C_{idle}(i)$
{ Dividing the residual N groups FCD into $C_{idle}(i)$ equally, $N \leftarrow C_{idle}(i)$ }
Assigning $C_{idle}(i)$ groups FCD to $C_{idle}(i)$ idle computing nodes,
$N \leftarrow N - C_{idle}(i)$; $i = i + 1$ }

Step 4. Master control server integrates the results from computing nodes, and estimates average velocity, density, and flow of road network.

Step 5. Processing next batch of FCD

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Based on this proposed algorithm, we built parallel computation system for massive FCD on KD-50 platform. High performance computer KD-50 is first developed independently and its property right is completely owned by China. It is a parallel processing system which consists of 144 Loongson CPU with 750M frequency. The gigabit switch is used to implement communication among computing nodes. Therefore, KD-50 offers a powerful platform for parallel computation of FCD [13].

We employed the field FCD of Beijing to validate this proposed algorithm. In Beijing, there are about 25,000 taxies which are used as floating cars for collecting real-time traffic information of road network. Each floating car transmits a group of FCD records to remote monitoring center every 5 minutes through GPRS network. The amount of FCD per hour is about 2.5M bytes. In the experiments, we selected the FCD packages which were sampled in 9:00~10:00am, July 6, 2010, to test the proposed algorithm.

Speedup, efficiency and utilization ratio of CPU were chosen as evaluation metrics. The experimental results are shown as Fig.2-3 and Table 1.

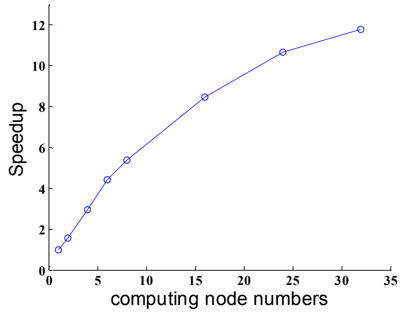

Figure 2. The results of speedup change with the number of computing node numbers

From Fig.2, it is easy to see that there is nearly linear relation between speedup and the number of computing nodes. While the number of computing nodes increases to 24, the parallel speedup has a slight decrease. Figure 3 shows the results of the efficiency varying with computing node numbers. When our proposed algorithm is employed for parallel computing of FCD, higher efficiency can be obtained, which is still more than 53% until the number of computing nodes increases to 24.

For the purpose of comparison, typical dynamic task scheduling method proposed by Fujimoto et al. and this algorithm are respectively adopted to handle 5 min FCD packages (including 37017 FCD records) of Beijing [14]. The experiments were conducted in KD-50 parallel computer platform, and maximum of computing node numbers is set to 12. The node operation situations are listed in Table Ⅰ. Average utilization ratio of Loongson CPU is 89% using RR method, while Average utilization ratio is 95% using our proposed algorithm. The experimental results illustrate our algorithm has better parallel performances.

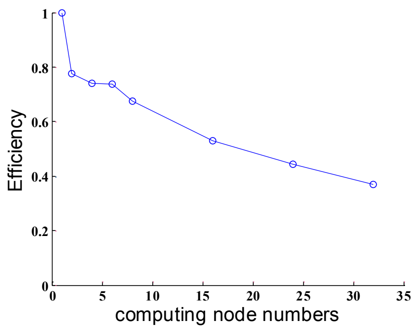

Figure 3. The results of efficiency change with computing node numbers

V. CONCLUSION

The current serial processing modes of FCD can not meet the real-time requirement of urban traffic dynamic guidance and control. In this paper, a parallel processing architecture is developed for massive FCD based on KD-50 platform. The characteristics of FCD are found by experimental investigation. In allusion to uncertainty and dynamics of FCD, we first propose a dynamic task scheduling algorithm to realize task partition and assignment properly for massive FCD. The experimental results illustrate our algorithm has better parallel processing performances. It provides an effective way for urban traffic dynamic guidance and control.

REFERENCES

[1] Turksma, S., "The various uses of floating car data," *Tenth International Conference on Road Transport Information and Control*, April 2000, pp.51-55.

[2] De Fabritiis C., Ragona, R., Valenti, G., "Traffic estimation and prediction based on real time floating car data," *11th International IEEE Conference on Intelligent Transportation Systems,* 2008, pp. 197-203.

[3] Stefano Messelodi, Carla M. Modena, Michele Zanin, et al., "Intelligent extended floating car data collection," *Expert Systems with Applications*, 2009, 36(3), pp. 4213-4227.

[4] Yu Quan, Rong Jian, Liu Xiao-ming, "Floating vehicle data detection system in Beijing," *IEEE on Intelligent Vehicles Symposium*, June, 2006, pp.320-324.

[5] Pfoser, D., Tryfona N., Voisard A., "Dynamic travel time maps- enabling efficient navigation," *18th International Conference on Scientific and Statistical Database Management*, 2006, pp. 369 – 378.

[6] Kerner B.S., Demir C., "Traffic state detection with floating car data in road networks," *IEEE Proceedings on Intelligent Transportation Systems*, Sept. 2005, pp. 44 – 49.

[7] Gurcan Comert, Mecit Cetin, "Queue length estimation from probe vehicle location and the impacts of sample size," *European Journal of Operational Research*, August 2009, 197(1), pp. 196-202.

[8] Cohen S., Bosseboeuf J.L., Schwab N., "Probe vehicle sample sizes for travel time estimation on equipped motorways," *Eleventh International Conference on Road Transport Information and Control*, March 2002, pp. 172 – 176.

[9] Guiyan Jiang, Longhui Gang, Zhili Cai, "Impact of Probe Vehicles Sample Size on Link Travel Time Estimation," *Proceedings of the IEEE ITSC*, Toronto, Canada, September 2006, pp.505-509.

[10] Eisenman S.M., List G.F., "Using probe data to estimate OD matrices," *The 7th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2004, pp.291-296.

[11] Wei Lu, Wengjia Wang, Kimita K., Yokota T., "Decreasing FCD processing delay by deploying distributed processing system," *6th International Conference on ITS Telecommunications*, June 2006, pp.206-209.

[12] Zhaohui Zhang, Changjun Jiang, Yu Fang, "Road situation modeling and parallel algorithm implementation with FCD based on principle curves," *Eighth International Conference on High-Performance Computing in Asia-Pacific Region*, Dec. 2005, pp.6-11.

[13] Feng Chen, Hao Pang, et al., "FCD parallel computing and dynamic guidance for urban traffic on KD-50-I-E platform," *Journal of university of science and technology of china*, 39(5), pp. 558-560. 2009.

[14] Noriyuki Fujimoto, Kenichi Hagihara, "Near-optimal dynamic task scheduling of independent coarse-grained tasks onto a computational grid," *International Conference on Parallel Processing-ICPP*, 2003, pp. 391-398.

TABLE I
THE UTILIZATION RATIO OF COMPUTING NODES USING MAX-MIN METHOD AND OUR ALGORITHM RESPECTIVELY

Computing node NO.	Execution time(sec) (Max-min method)	waiting time(sec) (Max-min method)	Execution time(sec) (our algorithm)	waiting time(sec) (our algorithm)
01	13.83	1.664	14.19	0.824
02	14.27	1.224	13.78	1.236
03	12.62	2.872	14.81	0.204
04	11.38	4.120	14.19	0.824
05	15.29	0.204	14.98	0.0320
06	13.66	1.836	15.01	0.011
07	14.86	0.6319	14.19	0.824
08	14.48	1.020	15.02	0.001
09	14.48	1.020	13.78	1.236
10	14.07	1.428	13.78	1.236
11	13.86	1.632	13.37	1.644
12	13.46	2.040	13.37	1.644