Parallel Simulation of High-speed Trains Using Ray-based Cloud Computing

Hongyi Xue[a], Yonghua Zhou[a], and Yiduo Mei[b]

Beijing Jiaotong University[a] and BioTech Incorporated[b]

Abstract

Train operation simulation of high-speed railways is an important part to ensure safety and efficiency. With the development of high-speed railways, there exist a lot of train operation data to be processed, and the calculation speed requirements for simulation prediction are increasing. To meet the requirements of big data processing and real-time simulation of high-speed railways, the idea of parallel computing is introduced into train operation simulation and cloud computing is employed to improve the speed of train operation simulation. The parallel simulation algorithm of high-speed trains is established based on the moving-block system through the Ray distributed computing framework. The calculation time of parallel simulation is significantly less than that of serial simulation. It can reduce the train operation simulation and prediction time, facilitate timely adjustment of train operation plans, and ensure safe train operation.

Introduction

With the continuous development of rail transit industry, the demand for real-time and accurate train simulation is gradually increasing. The moving block system (MBS) is a core component of next-generation high-speed railway control systems. To meet the requirements of high operation efficiency of high-speed trains, the MBS shortens train running interval distances. Therefore, it requires the high real-time performance of train simulation.

There are some studies on train operation simulation, but most of them are based on serial simulation. However, there exists scarce literature on train operation parallel simulation, which is a challenging topic.

Ray is a high-performance distributed framework with better computing performance than MapReduce, Hadoop, and other frameworks. Therefore, we introduced distributed computing into big data processing in high-speed railways and designed a parallel simulation algorithm for train operation simulation based on the Ray distributed computing framework. We verified the correctness and rapidity of the parallel simulation algorithm, which can meet the requirements of real-time simulation in high-speed railways. The calculation time of parallel simulation is reduced by more than 31% compared with that of single node serial simulation. In addition, we compared train operation states under different station conditions, which is convenient for the study of actual train operation in high-speed railways.

Methods

The idea of parallel train operation simulation is based on the Ray distributed computing framework. The Ray distributed computing framework has the advantages of high granularity, good computing performance, and strong scalability.

The Ray is a new high-performance distributed framework launched by UC Berkeley. It uses a different architecture and an abstract way of distributed computing from the traditional distributed computing system. The Ray distributed computing framework has the advantages of high granularity, good computing performance, and strong scalability. It has been applied to the actual computing business by Alibaba Ant group. Comparing Ray's design idea from the perspective of the core framework, we can see that the most prominent advantage is the ability to dynamically build the task topology logic diagram. Therefore, it is more suitable for complex task processes, and apt to be adjusted according to requirements.

The distributed computing process based on Ray is as follows:

- A task object is created, and the task object will be sent to the local scheduler.
- The local scheduler determines whether the task object is scheduled locally or sent to the global scheduler. The global scheduler decides to send the task to a local scheduler in the cluster according to the current task status information.
- After the local scheduler received the task object, it puts it into a task queue and assigns it to the Workerprocess for execution after the computing resources and local dependencies are met.
- After receiving the task object, the worker executes the task, stores the function return value in the Objectstore, and updates the object table information of the master.

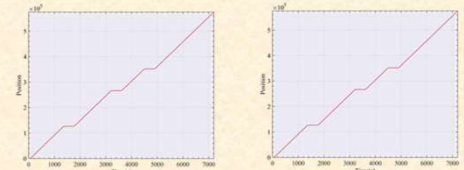

Fig.1. Position update.

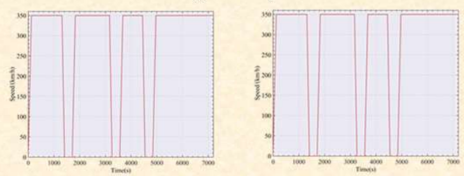

Fig.2. Speed update.

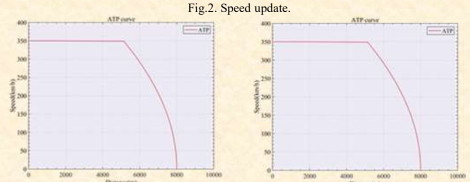

Fig.3. Simulated ATP curve.

Results

Serial Simulation and Parallel Simulation Comparison:

Figs. 1, 2 and 3 show the dynamic changes of positions and speeds with time, and simulated ATP curves of the first train, respectively. Comparing the results between serial simulation and parallel simulation, it can be found that both output results are the same.

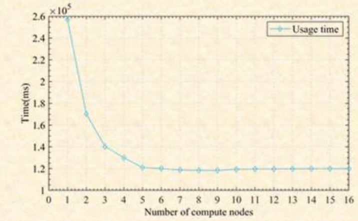

Fig. 7. Comparison of runtime using different number of computing nodes.

Results

While, the serial simulation time is 277021s, and the best parallel simulation time is 99367s, which implies that the computing time of parallel simulation is 64% shorter than that of serial simulation. Fig. 7 shows the runtimes of the parallel algorithm for calculating 140 trains using different number of computing nodes within a short prediction horizon of 180 seconds. The results show that when the cluster has less than 9 computing nodes, the parallelization effect of the algorithm is obvious. When the number of computing nodes exceeds 9, the program runtime tends to be stable.

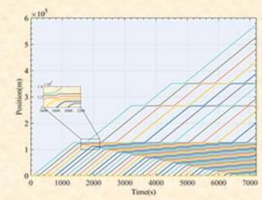

Fig.4. Simulation results when the station has a track.

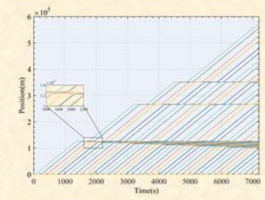

Fig.5. Simulation results when the station has 2 tracks.

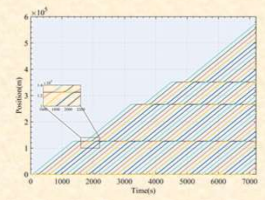

Fig.6. Simulation results when the station has 3 tracks.

Results

The parallel simulation results under different station track conditions are shown in Figs. 4, 5 and 6. They respectively show the position changes with time under the station conditions with single track, double tracks, and three tracks. The scheduling principle of first come first service (FCFS) is applied at stations. We can also see that except for at the stations, there are no overlapping points along railway tracks, which indicates that the trains have no operation conflict, and the expected rational results have been achieved.

Conclusion

Train operation parallel simulation based on the Ray distributed computing has been accomplished, which can effectively improve the calculation speed of train operation simulation. This meets the demand for real-time simulation prediction of train operation and provides a support for the future development of high-speed railway safety justification.