A real-time parallel processing system, PIM, and its application to a large-scale railroad operation simulator

Seiichi Yaskawa   Shigeru Yamamoto
Yaskawa Electric Corporation, Japan

Abstract This paper presents how our new computer architecture and software simulates what-if cases realistically, as they might occur in real-world operations, to give proof to the plans, developments, designs and schedules of the Shinkansen (Bullet Train) high speed railway system. A parallel programming and execution system that runs on personal computers came to the market just a few years ago. Railroad system simulators were soon found to be a natural application. With conventional programming schemes and computers it takes as much time and cost to build a simulator as is required to build a real control system. Paracell, the programming language, and the PIM, the parallel execution machine, made it possible to build such a simulator at less than one third of the cost originally estimated. The Tokaido and Sanyo Shinkansen Lines between Tokyo and Hakata are 1,100 km long, are the busiest among all the Shinkansen lines. The lines carry 300,000 passengers everyday on average, with a peak capacity of 1 million passengers per day. These numbers are achieved using approximately 1,000 trains traveling at a speed of 270 km/h. The challenge is to increase the train frequency, and limit operation error to within 15 to 30 seconds, while maintaining their legendary safety record--no passenger accidents in Shinkansen's thirty year history. Central Japan Railway Company (CJR) was faced with the task of planning the next generation of their COMTRAC (COMputer aided TRAffic Control) system, the Shinkansen control and monitoring system. They were wondering to what extent they could improve their control system. They were also wondering how they would test the new system and train the dispatchers. We then introduced the new agent-based and complexity-based parallel computer and software concept. CJR quickly recognized that their staff, without programming experience, could build information systems on their own with our software tool. They can test any what-if scenarios on do-it-yourself systems and evolve the systems by themselves.

1. INTRODUCTION

The COMTRAC (COMputer aided TRAffic Control) system, the Shinkansen (Bullet Train) control and monitoring system, was first developed and put into operation in March 1972. This system has undergone several upgrades over the years. The COMTRAC system for the Tokaido and Sanyo Shinkansen Lines between Tokyo and Hakata is now in the process of the next upgrade. This upgrade is being undertaken in order to increase transportation capacity, improve passenger service and reduce time for recovery from disruptions. The sole control center is in Tokyo, and the second and backup control center is now being built in Osaka. Thus, if the Tokyo control center were struck by disaster, the whole train operation would not stop.

The upgrade of the Tokyo COMTRAC system and the addition of the Osaka COMTRAC system require verifying new control logic and testing new computer systems. As there are more than 130 trains running at a time between Tokyo and Hakata, it is impossible to perform the verification and test on the real system. As a result, it becomes obvious that a simulator is required.

The more realistically and precisely you try to simulate the real world, the more enormous the software becomes. The larger the software gets, the longer the execution time gets, and it becomes very difficult for the system to run real-time and be a simulator. This problem, combined with economic considerations, is why only limited number of simulators have been in practical use even though it is well known that building a good system requires a simulator.

Train operation simulators have been achieving the goals of real-time performance and low cost by compromising their simulation details. For example, simulated trains run between stations at a predetermined average constant speed. This is, however, far too limited a scenario when trying to simulate realistic situations in real-time with enough precision, particularly when trains flock as a result of an unforeseen disruption. Since the planned upgrade of the COMTRAC system for the Tokaido and Sanyo Shinkansen Lines is a major one, it was anticipated that any conventional simulator was not good enough for system verification and test.

We, working jointly with Central Japan Railway Company, developed a CTC (Centralized Traffic Control) simulator for detailed operations of a large number of trains. As the platform for the CTC simulator, we chose the PIM (Parallel Inference Machine) real-time parallel processing system with the following strong points:

(1) Fixed response regardless of system scale or load (true real-time execution)
(2) Easy application programming, modification, and enhancement
(3) Incremental and allocated system development from small to large scale
(4) A Programming language to easily describe numerous and various real-world parallel phenomenon
(5) On-line change and addition of program code and

displays without negative impact
(6) Extensive tools to test and debug code

The CTC simulator is the main part of the PRC (Programmed Route Control system) simulator. The PRC simulator was developed to verify new train tracking and control logic to be used in the PRC system; the core of the COMTRAC system. Based on the successful results on the PRC simulator, a larger-scale simulator was planned to be an on-line subsystem of the next generation COMTRAC system. This system is now in test. This paper presents the CTC simulator as developed and implemented on the PIM. This CTC simulator is also the base for the next generation COMTRAC simulator subsystem.

2. TRAIN OPERATION SIMULATOR REQUIREMENTS

The railroad system, including the Shinkansen bullet train system, is one of the most important means of transportation in Japan. The increase of carrying capacity and improvement of passenger service are always in demand. Some of the different techniques to fulfill this demand, just to name a few, include train frequency increases, headway reduction, reduced travel time, railroad line extension, establishing new stations, increasing passenger comfort, and quick recovery from disruption. To realize those requirements and maintain safe and smooth train operation, the infrastructure from train scheduling and rolling stock scheduling, to crew allocation and electricity is effected in order to improve and reinforce the overall operations.

Train operation systems are open to the outside world, and as a result, are different from closed automation systems on factory floors. The environments of passengers, weather, electricity, etc. are always changing, and the plans and their execution must be flexible. The optimization problems for such a railroad system with so many variables cannot be solved analytically. Experiment, analysis and verification on an actual system are mandatory. In reality it is almost impossible to use an actual system in operation for experiment. Building an experimental system equivalent to the actual system is not economical either.

Various train operation simulators have been developed. The precision and scale of these simulations are in conflict with real-time and cost effectiveness. Most of the simulators in the past attempted to meet the required real-time performance and cost by limiting the scale and precision to a specific purpose. So-called general purpose simulators with precision and scale equivalent to actual systems have been proposed, but have shown limitation in the interpretation and scope of real-time while remaining at a high cost.

The TTS (Train Traffic Simulator) of the current COMTRAC system uses fixed travel times between stations instead of continuously calculating train speed, due to the limited processing power and memory capacity of the current computer. This is a discrete simulation that describes train movement as discrete events of arrival at and departure from stations. This type of simulation does not take actual speed changes caused by actual acceleration and deceleration, schedule change, speed limit generated from the ATC (Automatic Train Control) system, etc, into consideration. The simulated train speed would be too different from the actual one, especially when trains run in a bunch due to disruption.

To realize simulation that continuously calculates train speeds and has enough accuracy to test and verify a train operation monitoring and control system, all cases including disruptions, various operation facility conditions and train run conditions must be taken into account. The operation facility conditions include CTC (Centralized Train Control) devices, interlocking devices, and ATC (Automatic Train Control) devices. The train run conditions include train performance, ATC signals, and track conditions. The track conditions include gradients, curves, and tunnels. This type of continuous simulation with many trains is difficult to run real-time due to lack of computing power, particularly as the number of trains increases.

At peak hours there may be 130 or more trains on the tracks of the Tokaido and Sanyo Shinkansen Lines between Tokyo and Hakata. The number is expected to increase, and a new method that can simulate the operation with many trains at real-time and high precision has been desired.

3. OUTLINE OF THE SHINKANSEN OPERATION SIMULATOR

This chapter outlines the CTC simulator, which can simulate in detail the whole operation of the Shinkansen bullet trains between Tokyo and Hakata, even during times of disruptions. The CTC simulator is a major portion of the PRC simulator.

The system requirements for the CTC simulator are as follows:

(1) Simulate closely the operation facility conditions.
(2) Simulate closely the train runs according to the schedules, taking into account the train run conditions.
(3) Create or reproduce easily any unusual situation caused by an operation facility problem or a train trouble.
(4) Simulate closely the train runs during times of disruption.
(5) Perform the above simulations in real-time with the actual number of trains.

The PRC simulator includes the CTC simulator, which has the following additional requirement:

(6) Verify and evaluate the new train tracking, route control, and the schedule disruption monitoring function as the next generation PRC system logic.

The PRC simulator consists of the following functions, or subsystems:

(1) CTC simulator to generate CTC information, running virtual trains based on the operation facility conditions, train run conditions and schedules along the tracks between Tokyo and Hakata.

(2) Train tracking and route control function to track the train locations based on the CTC information, and output the track switching commands at appropriate times.

(3) Schedule monitoring function to monitor the train operations, estimate the arrival times at the next stations, and propose feasible schedule changes for recovery from disruptions.

Figure 1 shows the hardware configuration and figure 2 shows the software configuration.

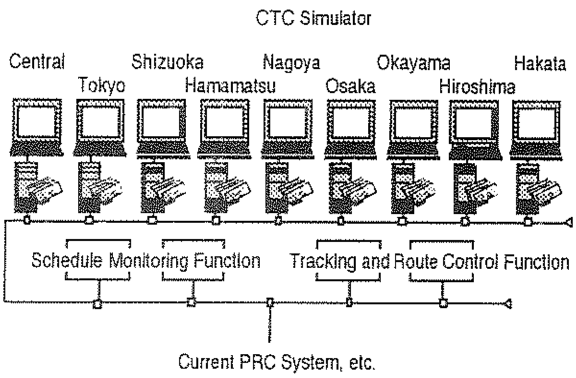

Figure 1: Hardware Configuration

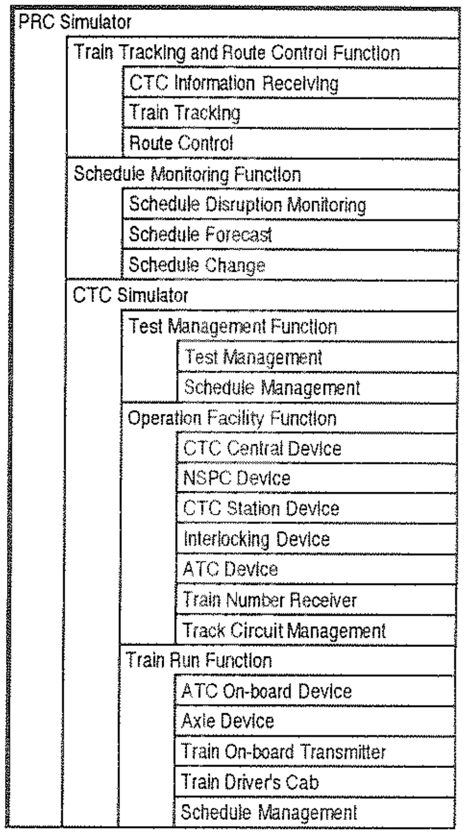

Figure 2: Software Configuration

4. REALIZATION OF THE CTC SIMULATOR ON THE PIM

The simulator for the Shinkansen train operation between Tokyo and Hakata is to simulate closely the interactions between the many parallel events as shown below, in real-time:

(1) Simulate the operation facility conditions, including elements such as over 2000 of track circuits (train detector), interlocking devices, and ATC signals. The basic elements are on and off, and the simulation is discrete.

(2) Calculate continuously the location of more than 130 trains on the tracks at a time, and simulate the train runs based on the schedules and train run conditions including variable elements such as gradients. This means performing more than 130 continuous real-time simulations with simultaneous interactions. Even a state-of-the-art engineering workstation could run no more than ten such operations in real-time continuous simulations.

(3) Create or easily reproduce any operation facility trouble or train trouble that would cause schedule disruptions. The scale of operation facility is proportional to more than 2000 track circuits, and there are more than 130 trains on the tracks. Each element can be handled independently as an object, and it is easy and simple to set any realistic trouble.

(4) Simulate closely the train runs during disruptions as interactions among all the above elements.

To realize the above, we chose a real-time parallel processing system, the PIM, with the following features.

Figure 3 shows the theory of operation of the PIM.

(1) Fine grain multiple virtual processors by time division (Cell):

System elements such as interlocking devices, ATC devices and train run function are assigned to each cell. Each cell executes the logic of its system element in parallel. System elements can be directly assigned to each cell as an object.

(2) Constant cyclic synchronized operation of all cells:

Each cell of the PIM repeatedly executes its code in synchronization with a fixed cycle (33ms in this

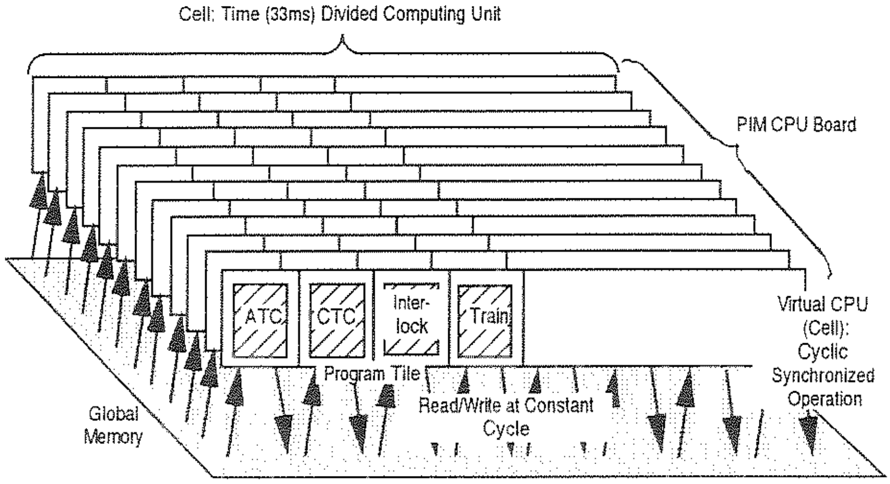

Figure 3: The PIM Theory of Operation

development). Utilizing this basic PIM behavior, the location of all trains can be updated simultaneously at every cycle. In a cycle of 33ms a train at 270 km/h travels about 2.5 m, and the speed of a train accelerating at 1 km/h/s changes 0.033 km/h. This illustrates that the cycle time of 33ms is a fast enough sampling period for train run simulation, and an extended (full day) simulation is possible with minimal simulation errors.

The PIM has an architecture suitable for multi-processor configuration (250 cells/processor, 250 x N cells/system), and still keeps the same constant cyclic operation (33ms). This simulator is provided with N = 36 processors, i.e. 9000 cells, and can perform the real-time precision simulation of more than 130 train operations. As all the cells run in synchronization with a constant cycle, it is possible to store in total the current status of a simulation, and resume the simulation from the stored status. These are called cross section storage and recovery.

(3) Object oriented development environment (Navigator) and programming language (Paracell):

The Paracell language has the basic structure of IF...THEN...ELSE... and is a declarative language. Operation facilities and train functions can be directly described in the unit of each function, i.e., object.

Since element or function of operation facilities and trains is treated as an object, it is easy and simple to create or reproduce any realistic trouble. Various set functions of the Navigator development environment are available to force various disruption states without the requirement of developing additional code to do so. Since the simulator will generate the evolution after a problem, i.e., disruptions, it is not necessary for the user to input the disruption data as with conventional simulators

(4) Cell independency based on the global memory scheme:

Each cell, synchronized in a constant cycle, repeatedly imports data from the global memory and exports the processed data back to the global memory. As cells do not communicate directly with each other, they can work independently from each other. With this feature, any single addition or change of operation facilities, trains, etc. can be done for each cell. It is easy to start development when a portion of the specification is fixed, continue the development incrementally, and expand the system in the future.

5. FUNCTIONS OF THE CTC SIMULATOR

The CTC simulator has three major functions: Test Management, Operation Facilities, and Train Run.

(1) Test and Schedule Management Function

To set up the simulation environment and receive and edit schedule information. The simulation setup items include test date, section, starting and ending time, timer base set and change, cross section storage and recovery, and simulator configuration.
To receive schedule information and schedule change information from the PRC system, and extract and edit necessary information.

(2) Operation Facility Function

This function consists of the following:

- CTC device: to generate CTC information and simulate the CTC scan.
- NSPC device (train number processing device): to track and manage train numbers.
- Interlocking device: to generate interlocking logic.
- ATC device: to generate ATC signals.
- Train number receiver: to manage the N-point

information (received train number).

- Track circuit: to exchange data between operation facility function and train run function.

(3) Train Run Function

The train run function consists of the following elements:

- ATC on-board device: to receive ATC signals of track circuit.
- Axle device: to update train-on-track information such as create trains, update train locations, and shuttle trains.
- Driver's cab (Driver): to operate the train such as generate the N-point information, check with the schedule and accelerate the train.
- Simulation schedule management: to receive schedule information and manage it.

Figure 4 shows the block diagram of data flow in the CTC simulator.

(4) Train Location Update Formula

The CTC simulator simulates train runs by repeatedly updating the train speed and location using the PIM cycle (dt:33ms) as the sampling period.

(a) Update average run speed

$$\text{Average Speed } Vave = \frac{Pp - Pt}{Tp - Tt} \quad (1)$$

where Pp: Next station position, Pt: Train position, Tp: Next station arrival time, Tt: Present time.

(b) Update present target speed

Choose the feasible speed, i.e., the lowest speed among the maximum performance speed, the ATC signal speed limit, and the average run speed.

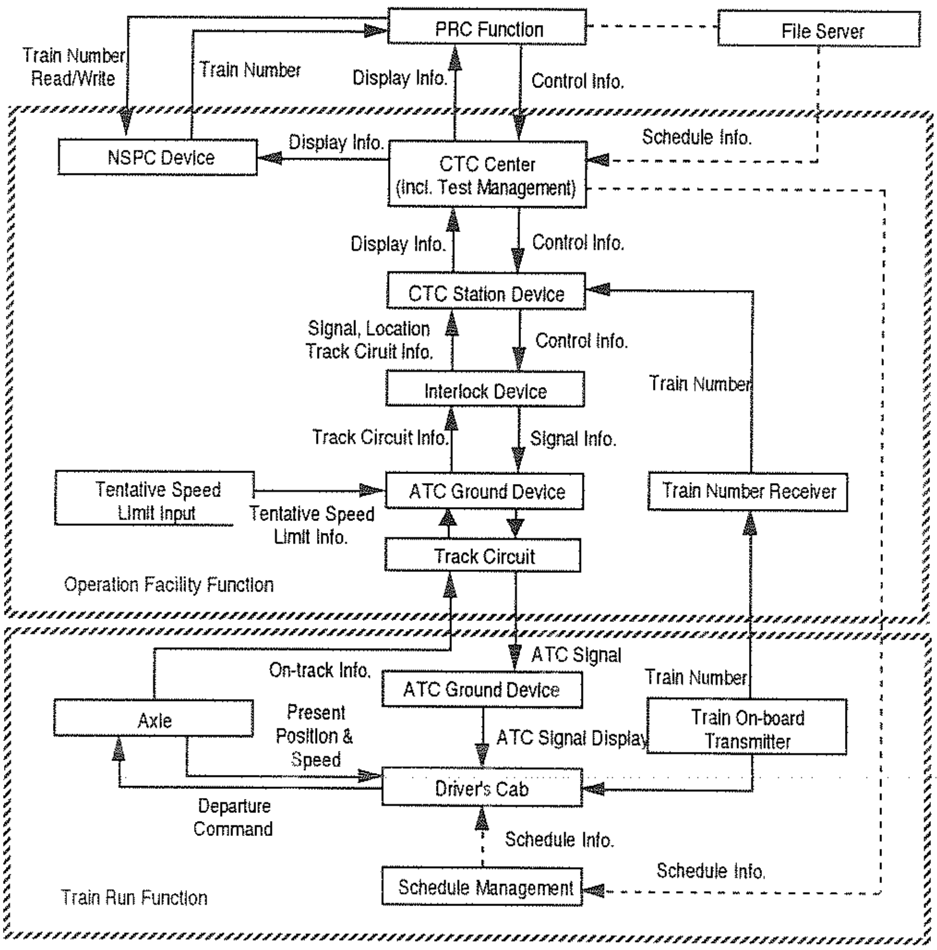

Figure 4: CTC Simulator Block Diagram

Present target speed
$Vp = \min(Vmax, Vatc, Vave \times UP)$ (2)

where $Vmax$: the maximum performance speed,
$Vatc$: ATC signal speed limit, $Vave$: average run speed,
UP: operation parameter (1 for passing, 2-3 for stopping)

(c) Update the train speed for the next cycle

Compare the present target speed Vp and the actual train speed Vt, calculate the necessary acceleration At if $Vp > Vt$ or deceleration At if $Vp < Vt$, and reflect them to the next cycle train speed.

$$Vt = Vt + At \times dt \quad (3)$$

The acceleration/deceleration At should be within the conditions of the specific train performance (Traction force Fa, Braking force Fb) determined by each run speed, and the resistance (Linear resistance Rm, Curve resistance Rs, Aerodynamic resistance Ra).

$$At = \frac{Fa\ (-Fb \text{ for deceleration}) - Rm - Rs - Ra}{W\ (= \text{train weight})} \quad (4)$$

(d) Update present position

Update the train position by adding the travel distance (= actual train speed × cycle period) for one cycle to the present train position for the last cycle.

$$Pt = Pt + Vt \times dt \quad (5)$$

6. PERSPECTIVE OF THE CTC SIMULATOR

The accuracy of the CTC simulator has been verified to be accurate enough to reproduce disruption situations by comparing with the actual data. The simulator has already been applied to verification of the new logic of the next generation PRC system, the improvement of a new train control method, and the new schedule to include high performance trains.

The next generation COMTRAC simulator system, which is an extended version of this CTC simulator, will first be used for test of the next generation PRC system and the systems for the second control center. It is also planned to be used for the following applications.

(1) Training of train dispatchers
(2) Verification of new facility plan such as a new station
(3) Verification of new schedule to include high performance trains
(4) Verification of new control logic such as new ATC

With the electric power functions added, this simulator would be further extended to test energy saving control methods, test disruption recovery plans with power substation capacity in consideration, and estimate the capacity of power substations. To forecast the schedule disruption transmission effects, this simulator can run at an accelerated mode, i.e., faster-than-real-time mode.

7. CONCLUSION

We have described the development of the Shinkansen bullet train operation simulator (CTC simulator) on the real-time parallel processing system PIM. With this CTC simulator development, the PIM has been proven to fit the train operation simulator applications. We will further apply this technology to other transportation systems.