Multi-hardware Real-time Simulator Resource Optimization Configuration Method

Wenbin Mu[1,2], Hao Wu[1,2]

[1] School of Automation and Information Engineering, Sichuan University of Science and Engineering, Yibin 644000, China

[2] Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Yibin 644000, China

Abstract

This article uses the subway traction power supply model as a simulation model to study the efficient and accurate real-time simulation of power systems with limited computing resources and real-time constraints. It focuses on the resource optimization configuration method for multi-hardware real-time simulators. Firstly, the article distinguishes the tasks suitable for different hardware based on their different computing characteristics, the generality of CPU, and the inherent high parallelism and stronger computing capabilities of FPGA. Then, it adopts a physics-decoupled method to perform fine-grained partitioning of the resource occupation rate required for processing tasks on each hardware, so that each module can operate with the minimum resource occupation rate on the hardware, maximizing hardware resource savings and computational speedup.The multi-level parallel real-time simulation architecture is formed through the coordination allocation of the coarse-grained and fine-grained partitioned model on the hardware, from hardware level, module level, to element level.

Keywords

Real-time Simulator; Field Programmable Gate Array; Resource Optimization Allocation.

1. Introduction

Electromagnetic transient real-time simulation accurately models complex power systems by detailed modeling and strictly ensures the real-time solution of each simulation time step, with the time step solution time strictly less than the physical time corresponding to the simulation step length [1]. Electromagnetic transient real-time simulation uses detailed mathematical models and performs three-phase instantaneous value simulations with microsecond-level simulation steps to accurately simulate the dynamic and static characteristics of complex power systems [2, 3]. More importantly, transient real-time simulation has the ability for hardware-in-the-loop (HIL) simulation, which can conduct online testing of actual physical devices [3]. However, the simulation scale in complex power systems is very large, with a large number of high-frequency power electronic devices in the model. Moreover, to obtain the accurate dynamic characteristics of high-frequency power electronic devices, it requires microsecond-level or even smaller simulation steps, posing multiple challenges such as computational capability, computational speed, and simulation accuracy for real-time simulation of complex power systems [4].

2. Multiprocessor Real-time Simulator Features

(1)Improve simulation calculation speed

Currently, commercial real-time simulators such as RTDS, HYPERSIM, and Speedgoat have been widely used in power energy and other fields [5, 6]. These commercial real-time simulators all have powerful underlying hardware, but most of them are serial devices. To further improve simulation speed, Speedgoat has introduced field programmable gate arrays (FPGA) with parallel characteristics as auxiliary hardware to share the calculation tasks of power electronic devices [7]. The CPU and FPGA assign tasks simultaneously, and the CPU in this simulator has four cores. Compared with traditional single-core processor structures, multi-core processors can provide stronger processing capabilities and higher performance. Multi-core processing has several advantages: parallel execution of high-performance tasks; multi-device integration, reducing device-to-device connections; reducing heat load, with lower power consumption per core, providing higher performance per watt; smaller device size, high integration, and higher computing capabilities per unit volume. Compared with serial devices, FPGA has a highly parallel hardware structure, distributed memory units, and pipelined architecture. These advantages make FPGA gradually become the main hardware for real-time simulation calculations in power systems [8, 9]. In Speedgoat, both the CPU and FPGA have parallel processing capabilities, and the two underlying hardware can not only assign tasks in parallel but also divide their tasks into smaller tasks for parallel processing. This feature can greatly improve the speed and capability of real-time simulation calculations.

(2)Low communication delay

In the Speedgoat real-time system, the FPGA board and the CPU are connected through a high-speed PCIe bus. There are two forms of interaction between the electrical system and the control system: voltage and current, and the control quantity and switch signal of the controlled source. These are PCIe registers and DMA (Direct Memory Access: direct memory access). PCIe registers are simple to use, but when performing large-scale high-speed data exchange, it introduces significant read and write latency to the CPU, and the CPU will stay busy for a long time, which is not conducive to the real-time operation of other tasks and may cause CPU overrun errors [10]. DMA is slightly more complex to use, but the data transmission relies on the DMA controller, which can directly transfer data from the FPGA to system memory, allowing the CPU to perform other tasks during this time. This article selects DMA-based data interaction to greatly reduce the CPU's occupation time, ensuring better real-time performance[11].

3. Hardware Resource Optimization Configuration Method

(1)Coarse-to-fine segmentation model

Coarse-grained partitioning is the process of dividing tasks according to the different computing characteristics of different hardware.In the case of FPGA and CPU, the entire simulation system is divided into two parts: the electrical system and the control system. Taking the subway traction power supply system as an example, the electrical system mainly includes the main substation, the inverter main circuit, the rectifier circuit, the traction network, and the load, while the control system includes the inverter controller and the load controller.If the time required for the hardware to complete one cycle of operation after coarse-grained partitioning is still greater than the set step size, and cannot meet the hardware operation conditions, fine-grained partitioning of the model is necessary.

For ultra-large-scale models with thousands of nodes, it is necessary to decompose them into various computing cores for parallel computing. The decoupling problem is complex and inevitable. For longer lines, the use of distributed parameter models can achieve natural decoupling through transmission delay[12]. However, relying solely on natural decoupling of long lines cannot meet the simulation requirements. The decoupling contradictions caused by complex interwoven network topologies and the situation where a single task is too heavy to cause timeout require human intervention. Using decoupling elements to artificially split the network structure into reasonable task quantities is necessary[13].

(2)Multilevel Parallel Architecture

The parallelism of the resource optimization configuration architecture of a hardware-oriented real-time simulator directly determines the size and speed of the simulation. Based on the coarse-grained and fine-grained parallel computing core, the partitioned model is mapped to the corresponding CPU and FPGA, focusing on the correspondence between the partitioned mathematical model and hardware resources. The hardware-level - module-level - element-level multi-level parallel simulation architecture is proposed based on this.

Hardware-level parallelism refers to the allocation of suitable calculation tasks to CPU FPGA and so on hardware according to their own characteristics, and the entire task is divided into 2 or more parts according to different calculation characteristics and allocated to the corresponding hardware for simultaneous execution.

Module-level parallelism refers to the fact that there are often multiple cores in a hardware, and tasks are assigned to multiple cores for parallel processing. Since the cores are parallel, the longest time for a core to process a task is also the highest time required for the model to calculate. After decoupling the module, it is necessary to allocate resources reasonably to each core. Each module is allocated to each core based on the calculation amount to achieve a basically consistent processing time for four cores, and the optimal module-level parallel computing speed.

Element-level parallelism mainly refers to the parallel calculation of electrical elements and control structures. Independent resources are allocated for calculation for RLC, power electronics, circuit breakers and other elements, and corresponding modules are designed for each element. The corresponding electrical elements are processed in parallel by each module in the pipeline, thus achieving parallel processing of various element modules. In the common dq dual-loop control structure of power electronics controllers, the calculation process of the two channels of the d-axis and q-axis is completely independent, with a good parallel structure, and independent calculation resources can be allocated in FPGA to achieve high parallelization.

4. Application of Resource Optimization for Multi-Hardware Real-Time Simulator

(1) System-level hardware/software co-design

System-level hardware/software co-design requires considering the mutual influence between hardware and software. Optimizing hardware resources at the system level can bring the following advantages:

Improved system performance: By optimizing hardware resources, the potential of hardware can be fully utilized, resulting in the maximum enhancement of system performance.

Reduced development costs: By optimizing resources, the performance of hardware/software can be fully utilized, completely solving the problems between hardware and software, and avoiding repeated modifications due to insufficient computing resources later on, thus reducing development costs.

Shortened development cycle: Hardware/software co-design can accelerate the design process, allowing products to be launched faster into the market.

Enhanced system reliability: By optimizing hardware resources, the coherence between hardware/software can be improved, avoiding compatibility issues between hardware and software, thus improving system reliability.

In summary, optimizing hardware resources is an important method in system design, and it is beneficial for improving system performance, reducing development costs, shortening development cycles, and enhancing system reliability.

(2) Hardware acceleration

The parallelism of the resource optimization configuration architecture for multi-hardware real-time simulators directly determines the simulation scale and speed. Based on coarse-grained and fine-grained parallel computing centered on mathematical model segmentation, the segmented models are mapped to corresponding CPUs and FPGAs, focusing on the correspondence between the segmented mathematical models and hardware resources. The multi-level parallel simulation architecture is proposed, including hardware-module-element multi-level parallelism.

The less time a segmented module takes to complete one cycle, the fewer computing resources it requires. The segmented modules are processed in parallel on hardware because it is parallel computing, so the maximum time for hardware to complete one cycle is the time it takes for the module with the maximum required computing resources to complete one cycle in the parallel module. By processing the segmented modules in parallel, the method of hardware real-time simulator resource optimization not only accelerates the speed of hardware simulation but also greatly saves on hardware computing resources compared to the original whole model.

In summary, simulating hardware to optimize hardware resources, evaluating hardware acceleration performance and power consumption, and determining the optimal hardware acceleration scheme can bring higher performance and faster calculation speed, providing significant advantages for calculate intensive tasks.

(3) Reduce waste of hardware computing resources

The model contains a large number of parallel structures, which can be assigned independent computing resources in hardware to achieve high parallelism. In the control subsystem, the distributed power supply and power electronic controller represented by the control loop are solved in the control system, where the common dq dual-loop control structure is taken as an example. The solution processes of the two channels in the d-axis and q-axis are completely independent, having a good parallel structure feature. Assigning these parallel structures independently in hardware can greatly reduce waste of hardware computing resources.

In the electrical subsystem, all elements are divided into basic passive elements (resistors, inductors, capacitors, and series-parallel composite elements composed of them), power electronic elements, transformer elements, line elements, etc., and solved separately, with independent hardware computing resources assigned to each module. Each module processes the corresponding electrical elements in a pipelined manner, thus achieving parallel processing of various element modules.

After the above processing of parallel structures and pipelined handling of corresponding electrical elements, hardware computing resource waste can be reduced, allowing the hardware to have more remaining computing resources to handle more tasks. This lays the foundation for the expandability of the model and reduces hardware computing resource waste, enabling it to execute more massive and complex models.

5. Conclusion

The resource optimization method for multi-hardware real-time simulators aims to fully utilize the different computing capabilities of CPU and FPGA hardware, and parallel execute calculation tasks on different hardware resources after coarsely and finely dividing the mathematical model of the system to be simulated. This method meets the requirements of detailed modeling of metro traction power system transient real-time simulation.A multi-level parallel simulation architecture is designed, including hardware level, module level, and element level. The hardware-level architecture design can achieve complete simulation functions on multiple hardware, making the entire simulator scalable. The simulation scale can be expanded by connecting more hardware, The suitability of CPU, GPU, and FPGA for different calculation tasks can be considered. How to allocate heterogeneous simulation tasks based on research requirements of simulation examples and achieve optimized configuration of heterogeneous simulation platform is also a topic worth further study, laying the foundation for hardware in loop simulation.

Acknowledgments

Sichuan University of Science and Engineering Graduate Student Innovation Fund (Y2022115).

References

[1] M. D.O.F, et al, Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis. IEEE Power and Energy Technology Systems Journal, 2015. 2(2): p. 63-73.

[2] Zhao Jinli, Liu Juntao, Li Peng, et al. GPU parallel algorithm for electromagnetic transient simulation facing exponential integration method [J]. Power System Automation, 2018,42 (06): 113-119.

[3] Zeng Jie, Leng Feng, Chen Xiaoke, et al. High-power digital-analog hybrid real-time simulation of modern power system [J]. Power System Automation, 2017,41 (08): 166-171.

[4] Li Peng, Wang Zhiying, Wang Chengshan, et al. Parallel architecture design of real-time emulator for active distribution network based on multi-FPGA [J]. Power System Automation, 2019,43 (08): 174-182.

[5] X. G, et al, Applications of Real-Time Simulation Technologies in Power and Energy Systems. IEEE Power and Energy Technology Systems Journal, 2015. 2(3): p. 103-115.

[6] W. H, et al, A Co-Simulation Method Based on Coupled Thermoelectric Model for Electrical and Thermal Behavior of the Lithium-ion Battery. IEEE Access, 2019. 7: p. 180727-180737.

[7] J. X, et al, FPGA-Based Sub-Microsecond-Level Real-Time Simulation for Microgrids With a Network-Decoupled Algorithm. IEEE Transactions on Power Delivery, 2020. 35(2): p. 987-998.

[8] Y. C. and D. V, FPGA-Based Real-Time EMTP. IEEE Transactions on Power Delivery, 2009. 24(2): p. 892-902.

[9] Wang Chengshan, Ding Chengdi, Li Peng, et al. The transient real-time simulation of photovoltaic power generation system based on FPGA [J]. Automation of Electric Power System, 2015,39 (12): 13-20.

[10] H. K, M. S. and B. C, High Performance FPGA-Based DMA Interface for PCIe. IEEE Transactions on Nuclear Science, 2014. 61(2): p. 745-749.

[11] W. J.L, et al, PISA-DMA: Processing-in-Memory Instruction Set Architecture Using DMA. IEEE Access, 2023. 11: p. 8622-8632.

[12] Zhang Runze. Research on vehicle network coupling simulation and resonance suppression strategy based on digital real-time simulation system [D]. Beijing Jiaotong University, 2018.

[13] Wei Wei Wang, Yiying Zhu, Chong Liu, et al. The real-time simulation implementation technology of large-scale power grid electromagnetic transient based on HYPERSIM [J]. Power Grid Technology, 2019,43 (04): 1138-1143.