文章编号:1671-4598(2021)05-0209-06 DOI:10.16526/j.cnki.11—4762/tp.2021.05.042 中图分类号:TP316 文献标识码:A

基于分布式仿真系统的实时架构设计

黄学进，余 婷
（中国航发控制系统研究所，江苏 无锡 214063）

摘要：在大型控制系统全生命周期的研发、测试与验证过程中需要应用到分布式仿真平台，通过加载不同阶段不同格式的模型以满足开展联合仿真的试验需求；文章对分布式仿真系统架构进行深入研究，创造性的设计了基于RTX的实时仿真技术和基于反射内存的实时通讯技术的总体架构，设计了实时模型的加载方法，并针对实时通讯性能开展优化设计；文章基于分布式仿真系统的实时架构搭建了一套完整的仿真平台；通过测试节点间时间，1 kB数据单向延迟160 μs以内，满足控制系统实时性要求；本架构在控制系统的全数字、HIL、半实物测试仿真中进行了多轮应用，满足不同控制产品的测试仿真需求。

关键词：分布式仿真；实时系统；RTX；模型加载；实时通讯；反射内存

Architecture Design of Distributed Real—Time Simulation System

Huang Xuejin，Yu Ting
（AECC AERO Engine Control SystemInstitute，Wuxi 214063）

Abstract：Through the whole life cycle of large—scale control system，a distributed simulation platform is necessary for test and validation. Loading different models in different stages and formats to meet the needs of joint simulation test. This paper focus on the architecture of distributed simulation system，and creatively designs anarchitecture based onRTX and reflective memory . The loading method of real—time model is designed，the real—time communication performance is optimized. This paper builds a complete simulation platform based on the real—time architecture of distributed simulation system. By testing the transfer delay between two nodes，the one—way delay of 1 kB data is less than 160 μs，which meets the real—time requirement of the control system. This architecture has been applied in MIL，HIL and half—physical—hardware—in—the—loop test simulation of the control system to meet the test simulation requirements of different control products.

Keywords：DistributedSimulation；Real—Time System；RTX；Model Loading；Real Time Communication；Reflection Memory

0 引言

复杂的大型系统如航空航天飞行器、舰船、车辆等都是由机械、电子、液压、流体、软件、控制等不同专业系统组成的综合体。如何更好、更快的开展大系统的正向设计工作，仿真技术已经成为复杂产品设计必不可少的重要手段[1]。

复杂的大型系统已经不再是一个单位、一个专业可以独立完成的。面对不同领域的设计模型，如何有效集成这些不同专业、不同格式、不同阶段的模型，从而开展系统级的功能、性能、接口等验证工作，并促进不同系统之间的协调是各大型系统设计验证过程中面临的重要问题，通过模型的联合仿真在设计前期就可开展相关的测试验证工作，促进产品快速迭代以便加速其成熟。

目前，国内外在飞机、飞行器等大型系统研发过程中广泛采用先进的分布式飞行仿真技术[2]。本文针对控制系统研发过程中的测试仿真需求，构建了一套通用的分布式仿真平台，支持不同模型的加载、支持模型的实时仿真、支持模型间的实时通讯且可灵活扩展仿真节点。

1 分布式仿真架构及原理

分布式仿真系统首先需要为多个仿真模型提供通讯技术，用以实现各个模型之间的实时数据交互。为了支持各模型便捷的接入仿真网络以及在仿真过程中不断的升级与变更，分布式仿真系统的拓扑结构要简单易维护，仿真模型的节点数量、节点间的通讯协议均可灵活配置。基本的网络构型见图1。

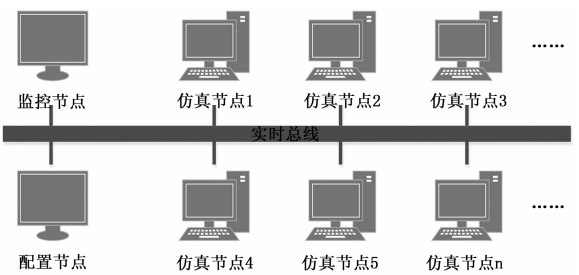

图1 分布式仿真系统基本网络构型

分布式仿真中除了基础的实时数据通讯外，仿真节点的实时仿真能力是另一个关键因素。

针对航空航天等安全关键的领域，实时性至关重要[3]。数控系统对控制时间有着严格的要求，在进行仿真平台构建时，仿真设备必须要按照控制系统规定控制周期完成规定的仿真任务，以保证仿真时序与控制时序的同步性和一

收稿日期:2021-03-23； 修回日期:2021-04-12。

作者简介:黄学进(1984-)，男，湖北咸宁人，硕士，高工，主要从事航空发动机测控仿真方向的研究。

引用格式:黄学进，余 婷.基于分布式仿真系统的实时架构设计[J].计算机测量与控制，2021，29(5)：209-214，219.

致性[4]。否则仿真时序验证不到位，与真实时序不一致则可能在后续试验及使用过程中造成重大的隐患。

实时性并不是指传输速度快，它包含了一种确定性[5]。实时性是指在制定的时间内完成制定工作的能力。

在实时仿真系统中，需要与外界设备进行交互，整个响应流程包含外部事件数据的采集功能、仿真模型的运行功能、与其它系统的数据交互功能等。在安全关键控制系统的运行过程中，如何在限定的时间内完成对外部的响应至关重要，直接影响整个大系统的运行安全。针对实时系统异常情况产生的后果等级不同，将实时系统标准分为了两类，硬实时系统和软实时系统。硬实时系统的时限是刚性不可改变的任何一次偶然的超时都会带来灾难性的影响。软实时系统的时限相对灵活，偶然的超时不会造成不可挽回的后果[6]。

针对分布式仿真系统架构的需求，本文主要开展实时仿真、实时通讯技术研究。为仿真系统提供硬实时的仿真环境，加载用户的自定义模型开展实时仿真。利用实时通讯总线实现仿真节点的数据交互，为分布式通用仿真平台的研发建立实时技术基础。

2 仿真架构设计

对于仿真系统来说，实时性的一个关键点在于计算机操作系统的实时性。一般来说，操作系统负责管理计算机的硬件资源和运行在上面的应用程序。实时操作系统用于运行非常精确定时和高可靠性的应用。这些应用在仿真控制中尤为重要，不确定的时间延长对仿真性能及控制效果都会产生影响。

对于实时系统来说，任何操作都会有一个已知的最大操作时间。实时操作系统通过为程序员提供灵活的控制任务优先级，而且通常还允许检查任务优先级，以确保重要任务的定时要求。

与实时操作系统相比，应用最广泛的个人电脑操作系统（如 Windows）被称为通用操作系统。通用操作系统（如 Windows）是为了响应用户的各种操作和相关程序（确保“公平”），内核任务的最小时间片为 10 ms[7]。实时操作系统为关键应用程序提供可靠、精确的定时（关注程序的优先级）。

实时操作系统具有以下特点：

1）确定性。实时系统中运行的应用程序或者程序的部分代码，能够保证在一定的时间误差范围内，称其为确定性；

2）抖动小。程序开始后，首次循环相对于后续循环在时间上的差异，称之为抖动。实时操作系统可确保在编程合理的情况下，抖动为最小水平；首次执行与后续循环执行所用的时间非常接近。

根据有关统计，目前存在的实时操作系统有上百种，应用场景各不相同，实时性能好且可应用于大型的实时应用开发系统包含 QNX、VxWork、RTX 等[8]，针对这些常见的实时系统及非实时的 Windows 系统性能进行比较，其基本性能见表 1。

表 1 操作系统实时性指标 （μs）

Cycle Times	QNX	Vxworks	RTX	Windows XP
Required	200	200	200	1 000
Average	197	199	200	～980
Jitter	－3	－1	0	～－20
Minimum	183	185	197	～400
Maximum	209	214	203	～10 000

QNX 和 Vxworks 是当前应用较广的实时操作系统，在国内外广泛应用于航空航天等实时性要求严苛的领域，操作系统实时性好，但整体应用复杂，代码移植难度较大，且费用高。

RTX 是基于 Windows 操作系统的实时解决方案，是目前 Windwos 平台的唯一纯软件的硬实时扩展子系统[9]。RTX 不对 Windows 系统进行任何封装或修改，其通过在 HAL 层增加实时 HAL 扩展来实现基于优先级的抢占式的实时任务的管理和调度，详细架构见图 2。

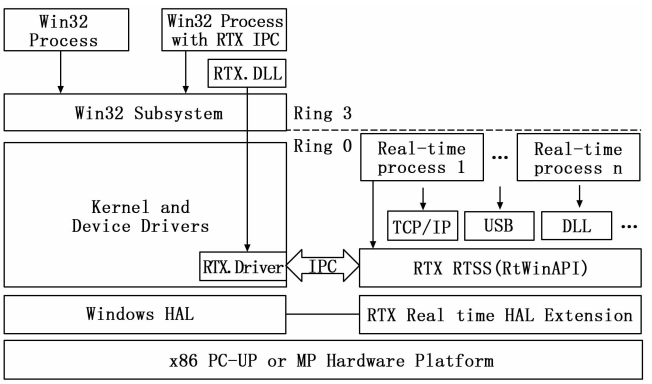

图 2 RTX 架构图

RTX 实时子系统 RTSS 的线程优先于所有 Windows 线程，提供了对 IRQ、I/O、内存的精确直接控制，以确保实时任务的 100％可靠性。通过高速的 IPC 通讯和同步机制，RTX 方便地实现与 Windows 之间的数据交换。RTX 的定时器时钟分辨率为 100 纳秒，最低定时器周期为 100 微秒[10]。

在 RTX 的定时器回调函数中，对 I/O 板卡进行操作，输出 10 kHz（0.1 ms）高低电压方波，通过逻辑分析仪抓取定时器的时间间隔。RTX 最小定时精度 100 us 的实测均值为 99.76 μs，抖动范围在 99 μs 至 100.4 μs 之间。详细数据见图 3。

RTX 作为 Windows 的嵌入式实时子系统，可通过共享内存与 Windows 进行及时的数据交互[11]。仿真平台中可利用 Windows 提供友好的人机交互界面，利用 RTX 提供实时仿真功能。基于 RTX 的应用开发难度相对较小，性能指标与 QNX 等相当，可兼顾实时性和友好性[12]。因此，本文选用 RTX 作为分布式仿真系统中各仿真节点中的实时操作系统。

仿真节点作为通用模块，需要提供通用的数据交互功能、通用的数据处理功能、能够灵活的加载用户自定义的不同模型功能，详细架构见图 4。

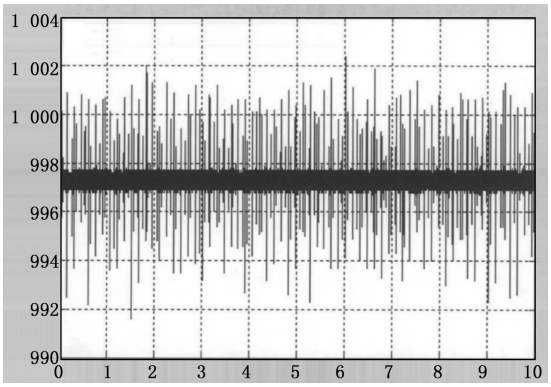

图 3　RTX100 μs 定时精度测试图（100 ns）

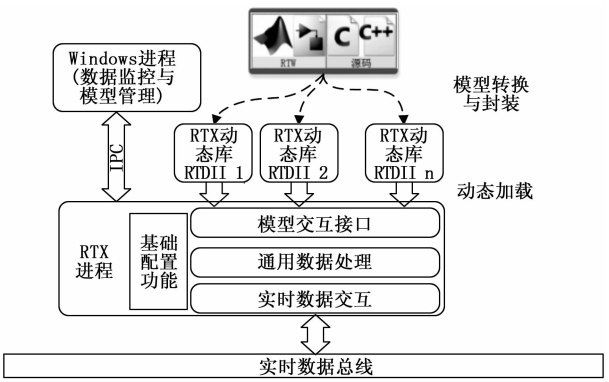

图 4　仿真节点详细架构图

仿真平台要实现通用性要求，核心在于仿真节点通用。仿真节点的通用需要设计以下几个方面来实现，首先是要构建通用的数据接口，可通过实时总线与其它节点实现灵活的数据交互，其次需要支持模型加载功能，实现不同模型的仿真与数据交互。

通用数据接口可在实时通讯基础上，定义通用的数据协议，利用配置的方式来实现不同节点的灵活交互。模型加载方面通过动态调用 Rtdll 的方式来实现[13]。仿真平台中设计了基于 Rtdll 的通用接口，首先设计规范的输入、输出格式，其次在模板 Rtdll 中规范初始化、运行、关闭 3 个通用接口。各模型根据实际情况，可自定义通用模型接口中的逻辑，或者嵌套调用已有的 Rtdll 的模式，从而实现自定义逻辑的封装。详细模型接口见图 5。

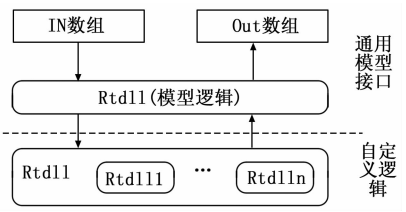

图 5　模型模板架构

仿真系统中采用 Rtdll 的格式实现模型的加载，但在实际仿真过程中，来自不同单位的不同模型的格式也不尽相同。分布式仿真系统可以加载哪些类型的模型、如何加载不同类型的模型？本文设计了模型加载技术路径可以支持源码、Simulink、Rtdll 三种格式模型的封装路径，详见图 6。Rtdll 类型的模型通过模型模板规范接口，以及再次封装实现模型加载。C/C++源码类型的模型可直接参照模型模板生成目标模型 Rtdll。Simulink 类型的模型可通过 Matlab 的 RTW 环境自动生成 C/C++代码，再通过模型模板封装成目标模型 Rtdll[14]。

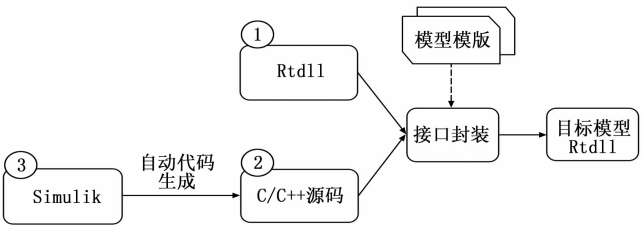

图 6　模型加载技术路径

仿真节点基于 RTX 硬实时操作系统开展设计，利用 Rtdll 的技术路线对用户的模型进行加载与实时运行，同时为后续实时通讯提供了基础的实时环境。

3　实时通讯技术设计

由于传统的以太网不能满足实时性能的苛刻要求，于是出现了反射内存网。反射内存是一种通过局域网在互连计算机之间提供高效的数据传输的技术。反射内存网硬件架构简单[15]，直接在计算机、工控机上安装专用的反射内存卡，并用光纤网在收发之间互联即可。整个网络最多可支持 256 个通讯节点。

反射内存卡是一组双口内存板。当数据存储到反射内存卡时，板上的高速逻辑会自动将此数据连同地址送到网络的其它反射内存卡板上，数据传递完全由硬件驱动，不需要 CPU 干预。反射内存网一节点写入数据时，其它节点在 400 纳秒内就会在相同地址处有相同数据。反射内存卡可以看成是由网上所有节点共享的单元内存卡，不存在访问限制与仲裁，每个节点就像访问本地内存一样，实现了节点间的数据共享。

反射内存卡支持多种总线方式，包含 PCI、PXI/PXIe 等。反射内存网支持两种拓扑结构，一是节省成本的环形拓扑结构，二是高可靠的星形拓扑结构[16]。

基于反射内存的通讯具有以下特点：高速度和高性能、使用方便、独立于操作系统和处理机、确定的数据传输时间、经济、高效等。

在分布式仿真实验中，常规的以太网通讯网络的通讯延迟与系统的运行状态、网络通讯负载相关，具有不可预测性，整体抖动大[17]。为了提高系统的置信度，需要引入先进的实时通讯技术，确保不同节点之间的通讯任务在指定的时间内完成。反射内存通讯技术是一种高速、实时通讯技术，已被广泛应用于实时仿真、实时控制等多种领域，提升了实时系统的数据传输实时性以及传输速率[18]。

针对分布式仿真平台的需求，为了支持模型、数采、监控等节点间的实时通讯，需要研发 RTX 实时系统下的反

射内存实时驱动[19]。在驱动设计过程中，采用不同的实现方式进行设计时，实际应用过程中用于通讯的耗时差异较大，可达到从 μs 级至 ms 级的差异，为了满足控制系统仿真的实时通讯要求，需要开展实时通讯性能的优化设计。对反射内存的通讯链路进行分析见图 7。

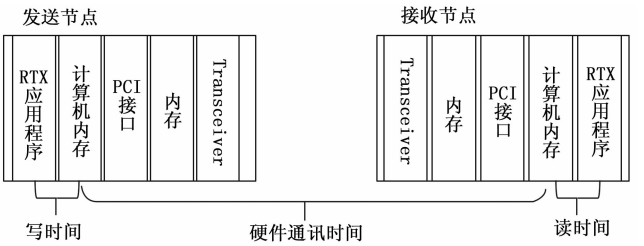

图 7 反射内存链路分析

从反射内存链路上分析，识别出反射内存通讯的延迟时间影响因子包含板卡间的通讯延迟和节点基本读写时间。通讯延迟是指板卡间的光纤通讯延迟及板卡内存到计算机内存之间的映射延时，该延时为纯硬件时间，对整体影响较小[20]。写时间和读时间是软件和计算机内存之间的交互时间，与驱动的设计方式密切相关，本次重点针对该延迟进行优化。

通过链路分析，基本读写时间的影响因素包含读写操作、通讯方式选择、DMA 阈值、传输量等，主要影响因素及取值范围见表 2。

表 2 基本读写时间影响因素分析

影响因子	类型	取值范围	
读写方式	离散量	Read	Write
通讯方式	离散量	DMA	PIO
DMA 阈值	连续量	—	
传输量	连续量	—	

分布式仿真平台中各节点数据传输量较小，在几百字节左右，首先针对传输量固定的情况设计试验，对读写方式、通讯方式、DMA 阈值的影响进行分析，详细试验设计见表 3，传输量为 1 000 B。

表 3 影响因素试验设计

	读写方式	通讯方式	DMA 阈值
测试 1	读	PIO	—
测试 2	读	DMA	256
测试 3	读	DMA	4 096
测试 4	读	DMA	10 240
测试 5	写	PIO	—
测试 6	写	DMA	256
测试 7	写	DMA	4 096
测试 8	写	DMA	10 240

通过测试，获得 1 000 B 传输数据下的读耗时试验数据如图 8，不同测试的读耗时详细数据见图 9～12，写耗时对比见图 13，详细写耗时数据见图 14～17。各模式下的平均耗时及抖动数据见表 4。

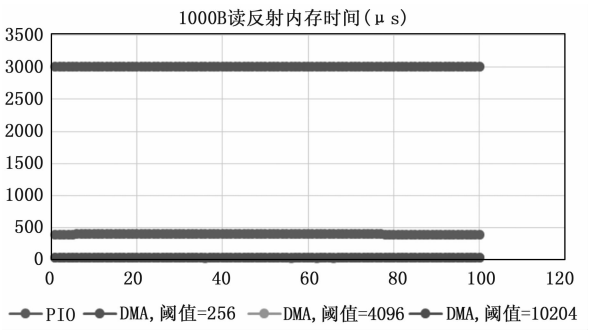

图 8 1 000 B 读数据耗时数据比对

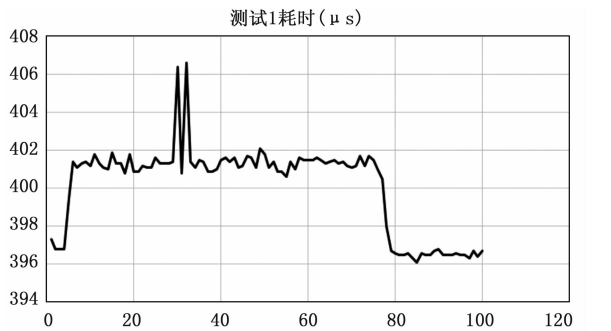

图 9 1 000 B 读数据耗时数据（测试 1）

表 4 读写数据耗时均值及抖动统计 (μs)

	PIO 模式		DMA 阈值＝256		DMA 阈值＝4096		DMA 阈值＝10 240	
	耗时	抖动	耗时	抖动	耗时	抖动	耗时	抖动
读	400.1	10.5	3 007.5	5.3	31.6	3.8	32.1	5.2
写	61.7	1.1	3 007.7	2.2	34.2	7.4	34.7	7

从表 3 中测试数据中可获得通讯方式 PIO 和 DMA 模式会对读写时间产生影响，且 DMA 阈值对性能也会产生影响。DMA 阈值设置合理的情况下，DMA 模式下的延迟性能优于 PIO 模式。

为了验证 DMA 阈值对性能的具体影响，在 DMA 模式下设计试验，对 DMA 阈值、传输量与耗时之间的影响进行分析。

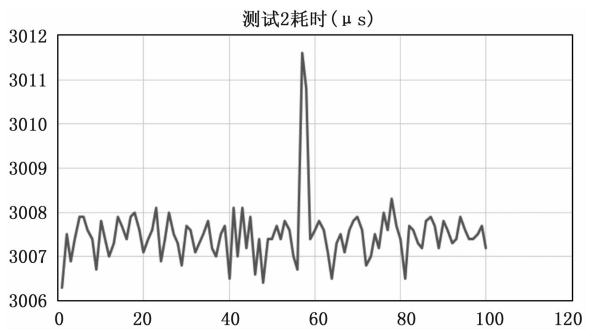

图 10 1 000 B 读数据耗时数据（测试 2）

通过测试，读耗时的测试数据见表 5，针对阈值与传输数据量的大小不同进行分析，读耗时与传输数据量大小的关系见图 18～19。写耗时的测试数据见表 6，针对阈值与传输数据量大小不同，写耗时与传输量之间的关系见图 20～

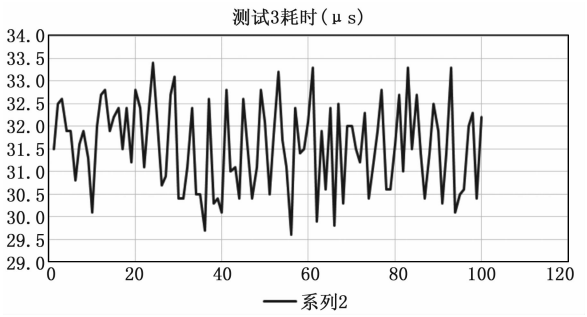

图 11　1 000 B 读数据耗时数据（测试 3）

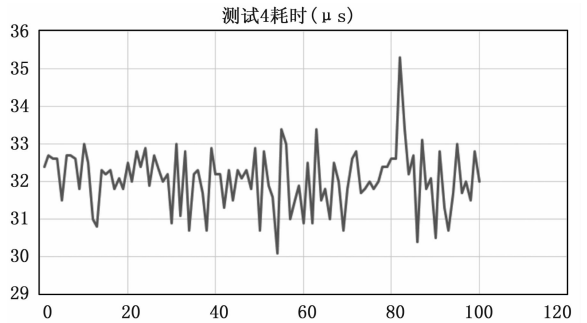

图 12　1 000 B 读数据耗时数据（测试 4）

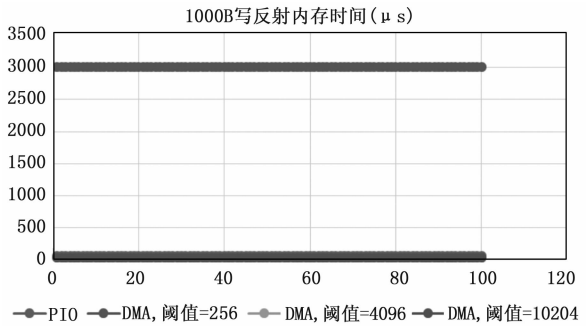

图 13　1 000 B 写数据耗时比对图

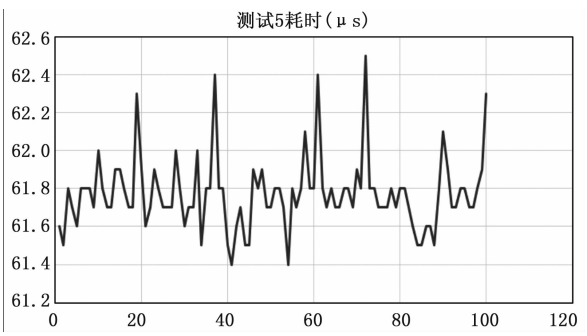

图 14　1 000 B 读数据耗时数据（测试 5）

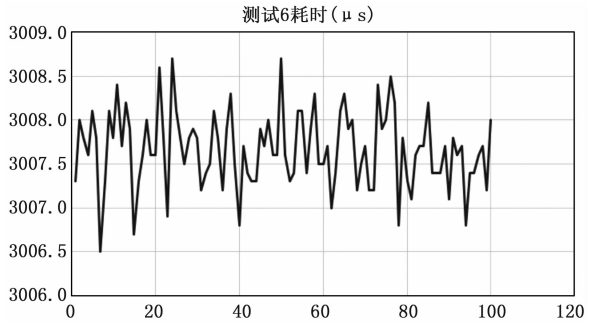

图 15　1 000 B 读数据耗时数据（测试 6）

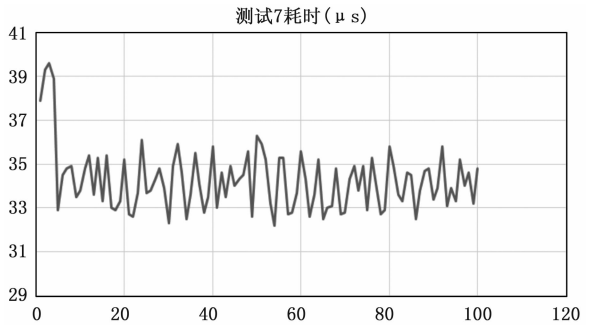

图 16　1 000 B 读数据耗时数据（测试 7）

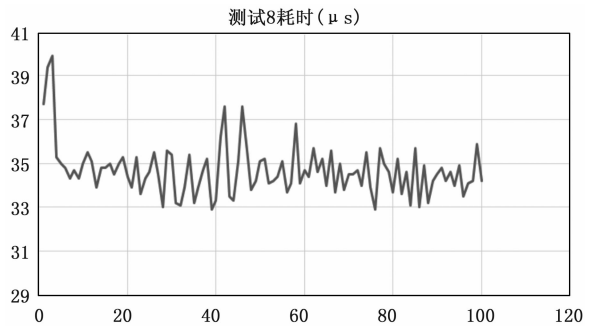

图 17　1 000 B 读数据耗时数据（测试 8）

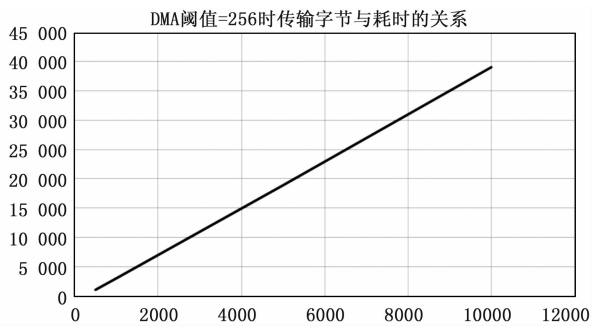

图 18　传输量与耗时关系（阈值＜传输量）

21 所示。

表 5　不同数据量和 DMA 阈值下读耗时统计

读耗时(us)	传输数据(B)			
DMA 阈值	500	1 000	5 000	10 000
256	1 007.654	3 007.507	19 007.72	39 007.87
4 096	27.279	31.568	1 007.765	2 007.642
10 240	27.266	32.069	66.006	108.562
20 480	27.201	31.778	65.979	108.968

从上述数据可以看出：

当 DMA 阈值大于传输的数据量时，读、写耗时明显减小，且读写耗时与传输数据量成正比；

在 DMA 模式下，读写耗时基本相当。

通过优化，控制系统仿真平台中常用的 1 000 B 的实时数据通讯性能明显比最初的 PIO 模式通讯性能更好，读写耗时从 400 多 μs 降到了 30 多 μs，极大的提升了分布式仿真中的通讯的性能。

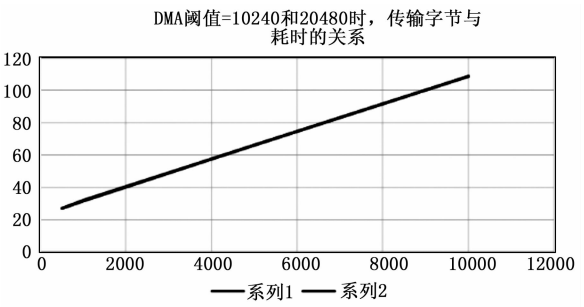

图 19 传输量与耗时关系（阈值＞传输量）

表 6 不同数据量和 DMA 阈值下写耗时统计 （μs）

写耗时	传输数据(B)			
DMA 阈值	500	1 000	5 000	10 000
256	1 007.828	3 007.668	19 008.061	39 008.022
4 096	30.094	34.267	1 007.922	2 007.672
10 240	29.898	34.69	79.741	131.908
20 480	30.371	34.589	79.932	132.321

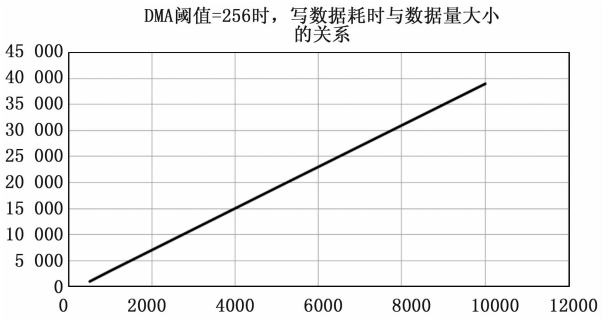

图 20 传输量与写耗时关系（阈值＜传输量）

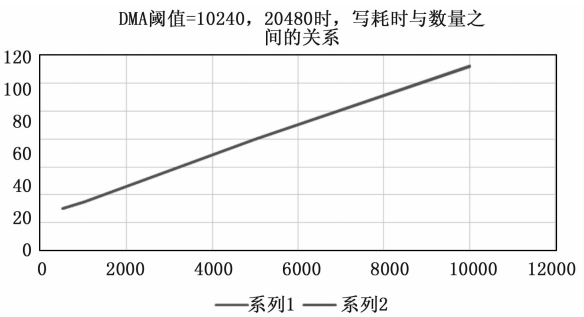

图 21 传输量与写耗时关系（阈值＞传输量）

4 结果及应用

在 RTX 实时操作系统基础上，利用 Rtdll 的模型加载与实时运行技术方案，结合基于反射内存的实时通讯技术，本文完成了仿真节点的设计和分布式通讯网络的设计，形成了一套通用的分布式实时仿真系统，并在某系统联合仿真中得到应用。

针对系统中的实时仿真节点的通讯性能进行测试，测试方案见图 22。

首先，系统中反射内存采用 DMA 模式，阈值设置为 4 kB。其次，由于系统中节点间的传输数据均小于 1 kB，通过自研的配置工具对模型节点的输出数据进行配置，设置为 1 kB，模型 1 的运行周期在 20 ms，模型 2 采用 while 方式运行。随后，在两个节点中加载模型，模型 1 用于记录数据，模型 2 收到数据后直接转发输出。最后，运行模型并记录 t2－t1。通过多轮测试，记录数据。测试结果建图 23。1 kB 数据的传输延迟均值为 125.5 μs，最大延迟不超过 160 μs。

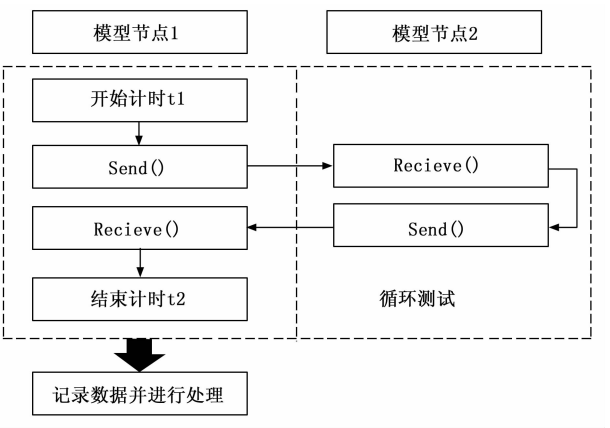

图 22 节点间传输时间测试方案

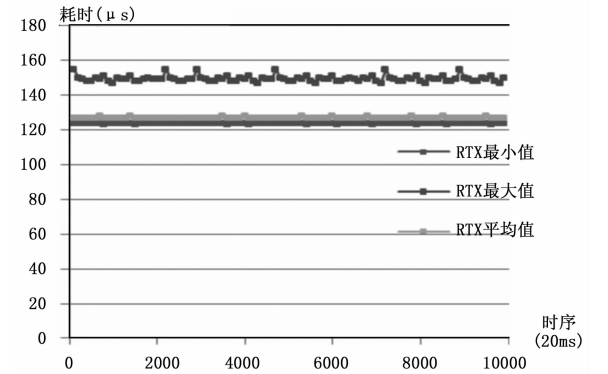

图 23 模型节点间的通讯延迟测

图 22 方案中，传输延迟包含反射内存写耗时、读耗时、光纤传输时间以及其它影响因素造成的延迟。通过表 4～5 中的数据进行线性处理，1 kB 数据读时间为 31.8 μs，写时间为 34.4 μs。光纤传输时间小于 1 μs，只考虑上述 3 种影响因素的理想时间延迟为 67.2 μs，小于测试延迟 125.5 μs，符合理论分析预期。

本分布式实时仿真系统应用于毫秒级的联合仿真环境，最高仿真频率为 1 kHz，节点间的延迟在 125.5 μs，完全满足仿真的实时性需求。

5 结束语

本文针对分布式实时仿真需求，基于 RTX 实时环境对仿真系统、模型加载、实时通讯技术进行研究。设计了基于分布式的仿真系统实时架构，并通过影响因子分析及详细的数据测试，设计出实时反射内存驱动，确保系统节点间的通讯延迟在 200 μs 以内，保证了仿真系统实时通讯性能。基于上述关键技术构建的分布式通用仿真平台已在具体项目上进行了应用，为分布式联合仿真提供了一种有效的解决方案。

（下转第 219 页）