

基于发布订阅的实时交互平台 NetDDS 的设计与实现^①

郑鹏怡^{②*} 陈进朝^{**}

(* 陕西国际商贸学院信息工程学院 西安 712046)

(** 西北工业大学计算机学院 西安 710129)

摘 要 针对分布式系统应用规模大、范围广、运行环境复杂的特征,分析分布式交互平台在功能和性能上的需求,设计并实现了一个基于发布/订阅机制的实时交互平台 Net-DDS。采取分层设计的思想,依次从分发服务层、QoS 保障层和数据传输层 3 个方面完成实时交互平台的构建,提供基于发布/订阅机制的数据同步功能、灵活可配置的服务质量保障机制以及基于以太网的多协议数据传输机制,确保了数据传输的实时性、正确性与可靠性。实验结果表明,该交互平台具有较低的消息更新延迟,能满足分布式应用中数据分发对实时性的要求。

关键词 分布式系统;实时交互平台;发布订阅;实时;数据同步

0 引 言

随着计算科学的发展,分布式系统逐渐成为计算技术与应用的主流。实时交互平台是构建分布式系统应用的关键技术之一,为分布式应用提供数据通信桥梁,在满足数据交互及时性和正确性的同时,为应用开发提供统一的接口^[1],屏蔽系统底层操作接口、通信机制的差异性,实现不同功能模块在分布式系统中的即插即用^[2],使系统开发人员不再关注具体的通信细节,缩短系统的研发时间。

分布式实时系统具有应用范围广、节点规模大、运行环境复杂等特征,对实时交互平台的实时性、动态性、可配置性等提出了很高的要求^[3]。(1) 实时性。分布式系统应用的很多处理必须在一定的时限内完成,对实时性有很高的需求。尽量缩短数据处理和传输的时延,是实时交互平台一个关键的需求^[4]。(2) 动态性。分布式系统通常运行在一个分布异构的环境中,系统的资源配置和应用间的交

互行为是动态的、变化的,可根据具体要求实时加入和退出,因此实时交互平台必须提供动态性的支持。(3) 可配置性。分布式系统应用需要大量的网络通信,如何在保障数据正确传输的同时,提供丰富的、可配置的服务质量策略^[5],满足分布式应用对资源和性能的各种需求,是实时交互平台需要考虑的一个重要因素。

以往的交互平台过多关注于解决复杂系统中的交互问题,对具体的实现细节进行了封装,是一种典型的黑箱设计方式^[6]。它们为分布式系统提供灵活性强、可重用性好等优势^[7],然而在实时性、可靠性和动态性等方面却面临着新的挑战^[8]。本文针对分布式实时系统特点,分析实时交互平台的功能需求,设计并实现了基于发布订阅机制的实时交互平台 NetDDS,提供了高实时、松耦合、可扩展的分布式数据分发服务,确保了数据传输的实时性、正确性与可靠性。所构建的 NetDDS 平台,具有良好的通用性与可扩展性,可应用于动态性强、分布性高、通信量大的军用、民用数字与半实物仿真试验、测试与

^① 国防重点实验室基金(61420020306)资助项目。

^② 女,1981 年生,硕士,讲师;研究方向:分布式系统;联系人, E-mail: zheng0174@sina.com
(收稿日期:2020-03-20)

评估等实时系统中,不仅能通过解耦促进分布式应用的快速设计与开发,还能利用实时数据分发服务实现分布式应用的高效运行与灵活扩展。

通过仿真实验环境,与美军试验与训练使能体系结构 TENA^[9]、分布式发布订阅消息系统 Kafka^[10]进行对比,验证所提出的实时交互平台在分布式实时系统中的可用性与高效性。

1 实时交互平台功能需求

分布式实时交互是分布式实时应用的开发、运行平台,它不仅仅为分布式应用提供端到端的信息同步功能,还能保证分布式节点间数据通信的正确性与有效性^[11]。基于发布订阅机制的实时交互平台,提供了表达时间约束的手段,在屏蔽通信节点位置的同时,为应用任务提供时间的高度可预测性^[12]。要想完成以上功能,实时交互平台需要提供以下 3 方面的功能。

(1) 支持基于发布订阅机制的数据同步功能

支持基于发布订阅的数据同步是实时交互平台的一个核心功能与关键要求^[13]。实时交互平台采用基于主题的匹配方式,实现了发布端和订阅端的匿名通信,并在匹配完成后,数据流从发布端直接发布到订阅端,实现了低负载、高效、实时的数据分发。

(2) 提供灵活可配置的服务质量保障机制

灵活可配置的服务质量保障机制是实时交互平台快速部署和高效运行的一个核心要求。服务质量保障机制应提供数据传输质量的描述手段,支持多种粒度、可配置实时管控机制,有效调节交互平台的发送、接收与处理环节,提升系统任务的时间可预测性,有效地管理网络带宽、内存空间等资源的使用,控制数据的生存时间、数据可靠性与实时性。

(3) 支持基于以太网的多协议数据传输机制

基于以太网的多协议数据传输机制提供了基于 TCP 和 UDP 协议的数据通信功能,满足了分布式实时应用在不同应用场景下的通信需求。基于以太网的多协议数据传输机制提高了实时交互平台的适应性和可用性,提升了分布式实时应用研发的效率和性能。

实时交互平台属于基于发布/订阅模型的消息中间件范畴,其核心是建立具有匹配机制的数据传输方式,为数据的发布端与订阅端公告信息并据此建立映射关系提供支持,并为分布式通信节点提供数据发送和接收的功能。

2 实时交互平台的设计与实现

依据实时交互平台在数据同步、服务质量保障、数据传输机制等方面的需求,本文利用基于主题通道的数据分发技术、基于多转发队列的 QoS 保障技术、基于传输组件的以太网通信技术来设计实现实时交互平台。

整个平台采用分层的设计思想,旨在增加代码的可重用性,使平台的实现模块化,功能模块之间达到低耦合高聚合。平台架构如图 1 所示,可以分为分发服务层、QoS 保障层和数据传输层等 3 个层次。

分发服务层向分布式应用程序提供能够支持匿名通信的数据交互功能,并利用发布/订阅机制接口,构建发布端与订阅端的数据通道,完成数据传递的位置无关性;QoS 保障层利用 QoS 策略列表,实现不同服务质量下的数据传递策略,为交互平台提供灵活可配置的服务质量保障机制,提升交互平台的灵活性与适应性;数据传输层采用 Winsock 库提供的 TCP 与 UDP 协议,提供网络节点连接建立及数据传输的接口,实现分布式节点间的数据发送与接收,支持 QoS 保障层与分发服务层功能的实现。

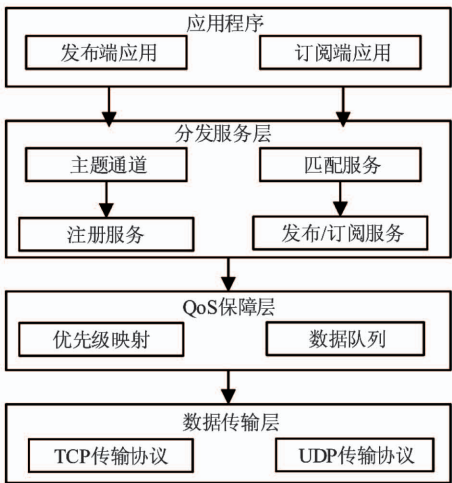


图 1 实时交互平台体系结构

2.1 分发服务层

分发服务层利用主题通道实现了交互平台的基于发布订阅机制的数据同步功能。实时数据分发服务平台具有一个主题通道服务器,发布者和订阅者分别向主题通道服务器公告发布与订阅的主题信息,主题通道服务器通过桥接机制将接收到的主题信息进行分发,发布者和订阅者通过主题匹配建立映射关系。映射关系建立后,发布者和订阅者就可以直接进行通信,不再需要主题通道服务器的干预。这不仅保障了发布端与接收端在物理位置上的无关性,还支持了通信关系与数据传输的匿名性。

分发服务层通过由主题来标志发布端和订阅端的数据属性,并进行匹配,将主题信息一致的双方建立数据连接关系。从图 2 中可以看出,在本文所采用的发布订阅机制中,主题是发布端和订阅端能否建立连接、进行数据传输的决定因素,发布端和订阅端首次运行时,利用以太网向主题通道服务器传递所必需的主题信息。主题服务器对所有主题信息进行分类与匹配,并将此信息转发到系统内的各个注册节点。注册节点接收到主题信息后,选择与自己信息匹配的节点建立连接,完成数据的推送与接收。

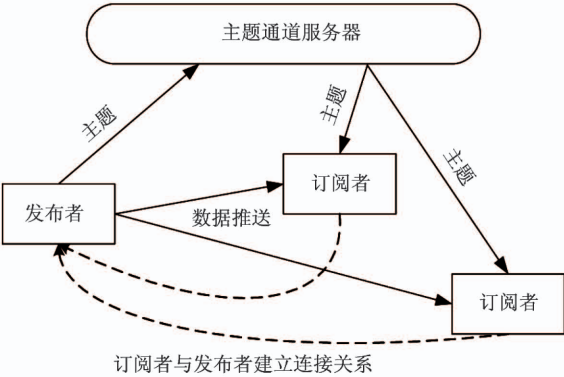


图 2 主题通道工作流程图

基于发布/订阅的数据分发服务为分布式结点之间的通信提供了统一的底层支撑,其技术原理如图 3 所示。发布/订阅机制通过主题通道实现了数据发布端和数据订阅端的解耦,满足了分布式系统中各功能与实体间通信的独立性。该通信具备异步性与实时性,发布端与订阅端在通过主题通道建立连接关系后,一旦发布端有新的数据产生,将直接推送给订阅端。因此,对于具有实时性、异步性、异构

性的分布式实时系统,基于发布/订阅通信模型的实时交互平台可以实现系统节点间的实时通信。

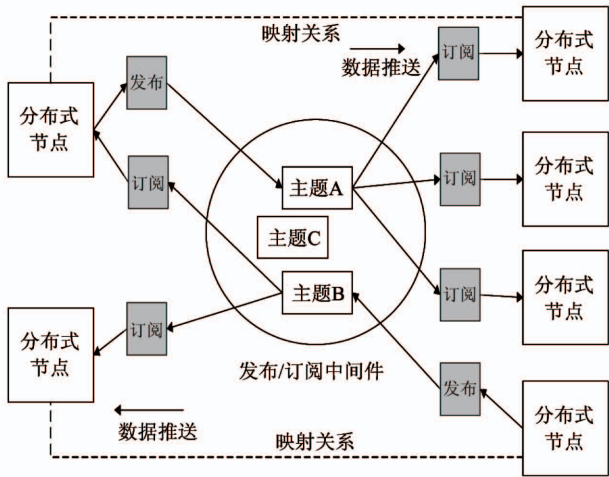


图 3 发布/订阅中间件模型

2.2 QoS 保障层

QoS 保障层保证了实时交互平台服务行为的可预见性。在实时交互平台的实现过程中,发布者和订阅者可以通过提供的 QoS 接口,完成中间件的设置,确定数据的行为和特征,实现数据的实时处理。QoS 保障层降低了交互平台的消息更新延迟,是满足分布式应用中数据分发对实时性要求的关键技术手段。

实时交互平台通过提供发布端 QoS 设置和订阅端 QoS 设置,实现了基于优先级的数据转发队列,并将按照数据的优先级对其进行排队。

实时交互平台中每一个需要传递的数据都有一个优先级。该优先级由初始优先级与浮动优先级组成。数据的初始优先级由应用程序中的订阅端通过配置接口指定,浮动优先级则是根据通信截止时间与当前系统时间的差值计算产生,并根据数据到达接收方的截止时间不同而动态提升变化。浮动优先级的大小跟截止时间成反比。传递数据的最终优先级是初始优先级与浮动优先级按照一定的规则进行加权计算获得。图 4 所示为基于数据队列的 QoS 保障机制的工作流程。

转发节点每隔一定的时间对转发队列刷新一次,根据截止时间计算出每个数据的浮动优先级,并与初始优先级相加得到每个数据的最终优先级。根

据最终优先级大小修改数据在转发数据队列中转发顺序。

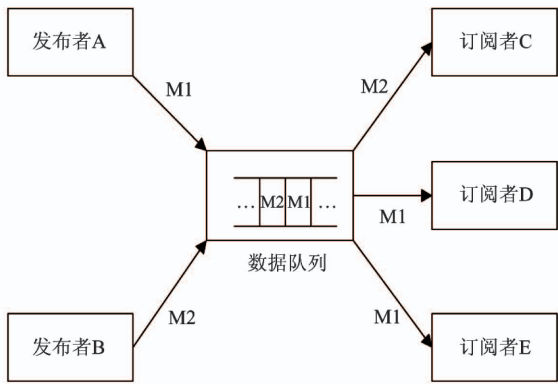


图 4 基于数据队列的 QoS 保障

在转发节点的排队策略中,订阅者为订阅的每个数据指定一个订阅优先级,由所有订阅该数据的订阅优先级之和确定数据的初始优先级,这样能够快速满足更多的订阅者的需求,提高了中间件的吞吐量。由数据的截止时间来动态提升数据的优先级,使得对时间要求紧迫的数据的优先级能够尽快提高,尽早发送给订阅者,满足实时性的要求。

2.3 数据传输层

实时交互平台的数据传输层对 Socket 通信进行了封装,完成了对 TCP 和 UDP 通信能力的支持,并提供了网络节点连接建立及数据传输的接口,使用户不必再关心 Socket 建立、连接、参数配置、监听等底层通信细节。

数据传输层由 TCPComponent、TCPPublisherComponent、UDPComponent、UDPPublisherComponent 等 4 个组件组成,依次实现 TCP 协议和 UDP 协议下的数据接收和发送功能。

TCPComponent 是完成订阅端 TCP 套接口的封装,负责 TCP 连接之间的通信。在实例化过程中会向发布端发起连接请求,并在成员变量中保存建立连接的本地套接口以及对端 IP 和端口。

TCPPublisherComponent 是 TCP 发布端组件,该组件对监听套接口封装,需要提供通用 TCP 发布端提供的功能,包括监听客户请求、维护并管理与多个客户的 TCP 连接、定义发布端向多个订阅端广播发送数据的接口、定义发布端从多个订阅端接收数据的接口、定义发布端向指定客户发送数据的接口、定

义发布端从指定接口接收数据的接口。TCPPublisherComponent 监听到某一客户请求后会在本地创建 TCPComponent 与订阅端建立连接,两者建立的 TCP 传输关系如图 5 所示。

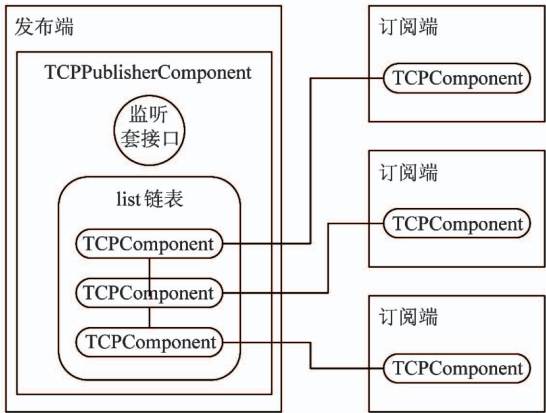


图 5 TCP 传输关系图

UDPPublisherComponent 是 UDP 发布端组件,该组件用于创建套接口,建立组播组,并提供发布端发送数据的接口。UDPComponent 是 UDP 订阅端组件,该组件用于创建套接口,加入组播组,并提供订阅端接收数据的接口。这两个组件建立的 UDP 传输关系如图 6 所示。

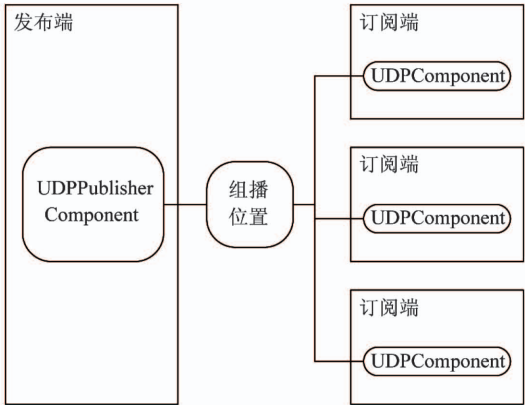


图 6 UDP 传输关系图

3 实时交互平台的测试

本节构建实验系统,在不同 CPU 负载和网络负载下,对本文设计与实现的实时交互平台 (NetDDS) 的性能进行对比与分析。实验系统的网络拓扑结构如图 7 所示。实验系统有 A、B、C 3 个节点,其中 C 运行主题通道,A、B 同时运行发布端和订阅端。测

试过程中,节点 A、B 同时订阅对方发送的主题兴趣。

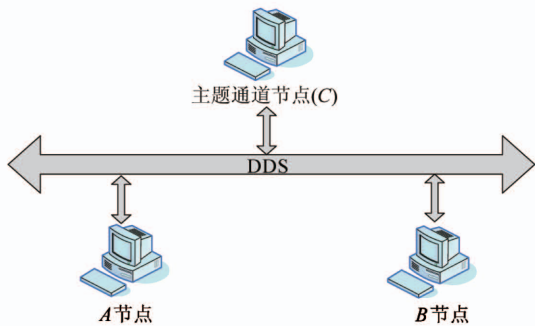


图 7 延迟测试系统拓扑结构

实验过程中,对比的主要指标为交互平台的消息更新延迟。消息更新延迟是交互平台发布端调用 DataWriter() 接口之前的时间戳与订阅端调用 DataReader() 接口获取消息之后的时间戳之间的差值,是影响分布式实时通信平台实时性能最关键的因素,决定发布者能否在合适的时间将正确的数据交付给订阅者。

实验中采用乒乓法获取往返时延的方式测试消息更新延迟。A 发送消息前记录时戳 t_1 并发送消息,发送消息完成后,转变成订阅者,等待消息返回,收到返回的消息后,记录时戳 t_2 ;B 接收到消息后,转变为发布者,将消息返回,消息更新延迟的计算公式为 $T = (t_2 - t_1)/2$,往返测试 10 000 次,采用 Windows 高精度计时器对时间延迟进行计时。

基于以上方法,测试 TENA 平台、Kafka 平台和本文设计实现的 NetDDS 平台在不同处理器负载和不同网络负载下的消息更新延迟。TENA 平台是美国国防部设计与研发的试验与训练领域的公共体系结构,具备强大的互操作、可组合与可重用特性,被广泛应用于大规模分布式靶场应用中。Kafka 是由 Apache 软件基金会开发的一个开源流处理平台,是一种高吞吐量的分布式发布订阅消息系统。实验结果如图 8 和图 9 所示。

在图 8 中,当 CPU 负载为 0、传输数据大小为 2048 Bytes 时,本文提出的 NetDDS 的延迟时间为 300 μs ,而此时 TENA 的延迟时间为 1405 μs ,是 NetDDS 平台的 5.1 倍。同时,Kafka 的延迟时间为 655 μs ,是 NetDDS 平台的 2.2 倍。当 CPU 负载变

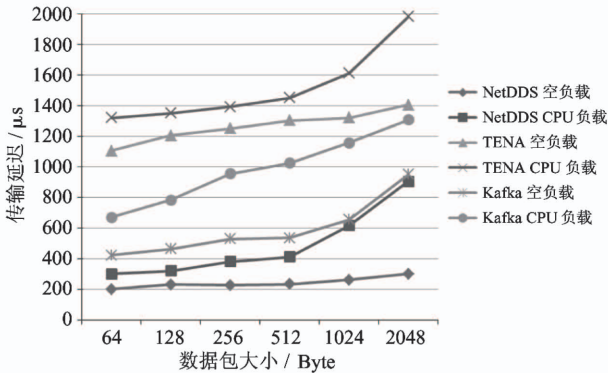


图 8 不同 CPU 负载下的测试结果

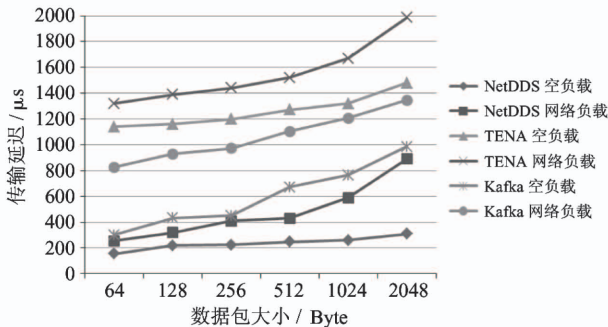


图 9 空负载和网络负载测试结果

为 75% 时,NetDDS、TENA、Kafka 的延迟时间都有了明显的增加,分别为 905 μs 、1980 μs 、1156 μs ,NetDDS 的性能依然优于 TENA 和 Kafka。同样,图 9 中,在相同数据包大小和相同网络负载情况下,NetDDS 平台的延迟时间分别小于 TENA 平台、Kafka 平台。

测试结果表明,在相同情况下,NetDDS 平台比 TENA、Kafka 平台的更新延迟小,而且数据包大小在 2048 Bytes 以内,不论当前系统内的两种负载情况如何,平台的更新延迟都在 1 ms 以下,达到了预期的效果。这是因为 NetDDS 利用 QoS 保障层实现了不同服务质量下的数据传递策略,为分布式实时应用提供了灵活可配置的服务质量保障机制,实现了数据的高效处理与实时传递。

针对 NetDDS 平台下传输一个 2 MB 大小的数据块,其吞吐量数据如表 1 所示。测试结果表明,在该组实验数据中,实际值与理论值比较吻合,误差最大值为 4.87%,表明 NetDDS 能够在合理范围达到较好的运行效果。

表 1 吞吐量实验数据对比表

数据 (Bytes)	均时 /ms	理论值 /kpbs	实际值 /kpbs	误差 /%
512	62	265.37	257.23	3.07
512	24	664.58	646.25	2.76
1024	14	1330.28	1275.68	4.10
1024	7	2667.34	2557.72	4.11
2048	4	3905.67	3715.55	4.87

4 结 论

针对分布式实时系统特点,分析大规模分布式应用对实时性和资源可控性的需求,设计并实现了基于发布订阅机制的实时交互平台,向应用层提供了基于主题的发布/订阅模型,应用发布端和订阅端以数据为中心直接进行数据交互,满足分布式应用对实时性的需求。

提出的实时交互平台采取了分层设计的思想,具备有分发服务层、QoS 保障层和数据传输层,提供了基于发布/订阅机制的数据同步功能、灵活可配置的服务质量保障机制以及基于以太网的多协议数据传输机制,实现了高实时、松耦合、可扩展的分布式数据分发服务,确保了数据传输的实时性、正确性与可靠性。实验结果表明,提出的实时交互平台的消息更新延迟都在 1 ms 以下,满足了分布式应用中数据分发对实时性的要求。

然而本文实现的实时交互平台底层采用的是通用以太网协议,而以太网的 TCP/IP 协议并没有专门实时性能保障机制,对强实时分布应用的支持力度不足,因此本文下一步工作的重点是研究将 TCP/

IP 协议进行改造,以增强平台的实时性支撑能力。

参考文献

[1] 郑鹏怡,张振国,袁战军. 基于发布订阅机制的实时中间件的设计与实现[J]. 计算机应用与软件,2018,35(2):44-47,53

[2] 万豪,史浩山,林卓,等. 基于订阅发布机制的中间件系统设计与实现[J]. 计算机测量与控制,2012,20(9):234-237

[3] 石春竹,柴处处,兰培霖,等. 一种支持 ModbusTCP 协议的 OPC UA 服务器中间件[J]. 信息技术与网络安全,2019,38(1):58-61,67

[4] Zhou P, Su Y K, Shen C. Design and implementation of distributed simulation system based on DDS[J]. *Journal of System Simulation*, 2014,26(8):1678-1683

[5] 冯国良. 基于 DDS 的数据分发中间件的设计与实现[D]. 南京:南京航空航天大学计算机科学与技术学院,2012:23-58

[6] 屈志坚,林宏平. 基于发布订阅的分布式监控主动消息交互研究[J]. 计算机测量与控制,2015,23(2):529-532

[7] 张学利,马娜,杨燕,等. 基于消息调度机制的地质服务体系构建及应用实践[J]. 国土资源遥感,2019,31(1):271-276

[8] 吴璨,王小宁,肖海力,等. 高性能计算环境中间件的优化设计与实现[J]. 计算机应用研究,2019,36(1):169-173

[9] 程景平. 基于 TENA 的靶场虚拟试验验证系统中间件研究[D]. 西安:西安电子科技大学空间科学与技术学院,2014:15-42

[10] 孙龙龙. 面向 Kafka 消息中间件的监控管理系统设计与实现[D]. 武汉:华中科技大学软件学院,2019:45-72

[11] 杨慧慧,岳峻,贾世祥,等. 物联网通信中间件的设计与实现[J]. 鲁东大学学报(自然科学版),2019,35(1):10-17

[12] 王小宁,肖海力,曹荣强. 面向高性能计算环境的作业优化调度模型的设计与实现[J]. 计算机工程与科学,2017,39(4):619-626

[13] 冯国良,谷青范. 基于 DDS 的实时中间件的研究与设计[J]. 航空电子技术,2011(3):41-46

Design and implementation of real-time interactive platform named NetDDS based on publish-subscribe

Zheng Pengyi *, Chen Jinchao **

(* College of Information Engineering, Shaanxi Institute of International Trade and Commerce, Xi'an 712046)

(** School of Computer Science, Northwestern Polytechnical University, Xi'an 710129)

Abstract

According to the large scale, wide application and complex running environment of distributed systems, the functionality and performance of distributed interactive platforms are studied, and a real-time interactive platform named NetDDS based on publish/subscribe mechanism is designed and implemented. The hierarchical design idea is adopted to build the middleware from the distribution service, QoS guarantee and data transmission layers, providing the data synchronization function based on the publish/subscribe mechanism, flexible and configurable quality of service guarantee mechanism and multi-protocol data transmission mechanism based on Ethernet. The proposed middleware guarantees the real-time, correctness and reliability of data transmission. Experiment results show that the middleware has a low message update delay and meets the requirements of real-time data distribution in distributed applications.

Key words: distributed system, real-time distributed interactive platform, publish/subscribe, real-time, data synchronization