Software Engineering and Applications 软件工程与应用, 2016, 5(1), 10-21
Published Online February 2016 in Hans. http://www.hanspub.org/journal/sea
http://dx.doi.org/10.12677/sea.2016.51002

Hans汉斯

A Survey of Middleware Transparency and Observability

Yunfei Yin[1,2], Yanhua Peng[1]

[1]CISDI R & D Co., Ltd., CISDI Group Co., Ltd., Chongqing
[2]College of Computer Science, Chongqing University, Chongqing
Email: yinyunfei@cqu.edu.cn

Received: Jan. 27th, 2016; accepted: Feb. 14th, 2016; published: Feb. 17th, 2016

Abstract

The arrival of the era of large-scale distributed computing has brought new opportunities and challenges to enterprises and society. In order to improve the efficiency, we often use middleware technology. Therefore, the transparency and observability of middleware is the premise of middleware selection. In this paper, we have a summary of the research on the concept, features, research status and technical connotation of the transparency and observability of middleware, and our goal is to help middleware enthusiasts to understand the research key points and research methods of middleware transparency and observability, thus providing a certain reference for the same line who develops or uses the middleware.

Keywords

Middleware, Transparency, Observability

中间件透明性和可观察性综述

尹云飞[1,2]，彭燕华[1]

[1]重庆赛迪冶炼装备系统集成工程技术研究中心有限公司，重庆
[2]重庆大学计算机学院，重庆
Email: yinyunfei@cqu.edu.cn

收稿日期：2016年1月27日；录用日期：2016年2月14日；发布日期：2016年2月17日

文章引用: 尹云飞, 彭燕华. 中间件透明性和可观察性综述[J]. 软件工程与应用, 2016, 5(1): 10-21.
http://dx.doi.org/10.12677/sea.2016.51002

摘　要

大规模分布式计算时代的到来，给企业和社会带来了新的机会和挑战。为了提高效率，我们往往借助于中间件技术，因此中间件的透明性和可观察性是选用中间件的前提。在本文中，我们对中间件的透明性和可观察性概念、特点、研究现状、技术内涵进行了综述性研究，旨在帮助中间件爱好者了解中间件透明性和可观察性研究的要点及研究方法，由此对从事中间件平台开发和使用的同行提供一定的参考。

关键词

中间件，透明性，可观察性

1. 引言

1.1. 概述

透明性是指软件开发平台和软件运行平台对于应用系统而言提供跨平台、跨语言和网络通信的服务，应用系统的跨平台、跨语言和网络通信功能均由软件开发平台和软件运行平台来提供。中间件平台是一种具有透明性的软件开发平台和软件运行平台，它通过 ipc 实现了网络层协议、通过 ipctransport 实现了传输层协议、通过 sock 实现了远程过程之间的连接、通过 support 实现了操作系统底层资源的管理和分配、通过 ocl 完成平台类的特殊封装[1]。

对于透明性，美国 ANSA 将它分为八类：1) 访问透明性(Access Transparency)，2) 位置透明性(Location Transparency)，3) 并发透明性(Concurrency Transparency)，4) 故障透明性(Failure transparency)，5) 迁移透明性(Migration Transparency)，6) 性能透明性(Performance Transparency)，7) 扩充透明性(Scaling Transparency)，8) 复制透明性(Replication Transparency) [2]。

可观察性是指中间件平台和平台上的应用系统是可以被观察的，其中前者是后者的基础后者是前者的外在表现。通过观察平台我们能够了解消息传输的内部机制；通过观察应用系统我们能够了解消息传输的外部表现[3] [4]。

可观察性分为两个层次，一是系统可观察性另一个是应用可观察性，其中前者是从系统层面观察系统对象、消息以及状态的形成和传输机制，后者是从应用本身观察对象、消息的发送、传输、接收和显示。

系统可观察性通过绘出系统的类图、顺序图等可视化手段来展示；而应用可观察性则通过打印消息的发送者、接收者、发送时间、内容等手段来展示[5]。

本文所做的工作是研究中间件的透明性和可观察性，对涉及到的透明性和可观察性概念、特点、研究现状、技术内涵进行了研究，旨在帮助中间件研究者了解中间件透明性和可观察性的要点及研究方法，并提供一定参考。

1.2. 国内外研究现状

对于中间件的透明性和可观察性研究，国内的研究一般是围绕某一具体的中间件平台应用系统而展开的[6]，因此我们将中间件平台应用系统的透明性和可观察性研究分为 PCDP 平台上的透明性和可观察性研究、ACE 平台上的透明性和可观察性研究、TAO 平台上的透明性和可观察性研究、DCOM 平台上的透明性和可观察性研究、ICE 平台上的透明性和可观察性研究。

实现中间件透明性和可观察性的方法有平台消息存储、消息调用次数实时统计、吞吐量实时统计、性能分析、历史消息分析等方法。

1) PCDP 平台上的透明性和可观察性研究

PCDP 平台中冶赛迪集团公司研制的面向冶金领域过程控制的一款中间件平台，它集软件开发平台和软件运行平台于一体。PCDP 平台规模为 C/C++ 50 余万行、开源代码 30 万行、Windows shell 脚本 10 万行，并且具有以下功能：a) 复用组件：集成过程控制应用的通用处理模块，Socket 通讯、OPC 通讯、日志管理、进程管理等；b) 领域应用：L2 后台控制和模型应用，仅关心业务知识，编程技术复用平台技术；c) 编译部署：集成自动编译、链接、部署和维护功能，减少开发和维护的工作量，统一的开发和运行环境及目录；d) 代码生成：快速生成 C/C++代码，提高开发速度，生成类代码、数据库访问代码、组件框架代码；e) 支撑类库：封装专业通用和业务通用的类库，如字符串类、通讯套接字类、线程类等等。

PCDP 平台上的透明性和可观察性研究起始于 2013 年 8 月，已具备的功能包括平台消息存储功能、消息调用次数实时统计功能、吞吐量实时统计功能、性能分析功能、历史消息分析功能、阻塞检测与报警功能等。

2) ACE 平台上的透明性和可观察性研究

ACE 是一个通信软件的开发工具包，是一个 CORBA 协议的具体实现。ACE 综合了许多主流的软件设计模式和组件技术，通过 ACE 我们可以解决在软件开发和维护过程中的一些繁琐的、易错的、不可移植的问题。ACE 提供了强大而高效的进程通信、同步互斥机制、共享内存、客户服务配置等功能。

国内对于 ACE 平台上的透明性和可观察性研究始于 2003 年马维达在程序员杂志上发表《ACE 与 GoF 设计模式——Adapter 模式在 ACE 内存管理类中的应用》一文，介绍了开发高性能网络化应用与下一代中间件的面向对象框架，这实际上拉开了研究 ACE 平台上透明性和可观察性的序幕[7]。随后，北京交通大学信息科学研究所也对 ACE 实现通信软件的设计机制和优越性进行了研究[8]。另外，高强文也介绍了可靠消息传送的一般实现方法——消息队列，并使用 ACE 的主动对象、连接器、接受器等技术实现消息队列及其接口[9]；中国科学院研究生院的李明介绍了基于 ACE 在实现某监控软件中的设计机制，这中间离不开对透明性和可观察性的研究[10]。2013 年，东北大学轧制技术及连轧自动化国家重点实验室结合国内中厚板生产的自动化控制需求，开发出基于 ACE 中间件的多进程轧机二级控制系统，该系统包含多个进程，进程之间相对独立，系统具有更好的稳定性和可伸缩性，使用自适配通信环境 ACE 实现进程及线程间的通信大大降低了系统开发难度，缩短了开发周期[11]。

3) TAO 平台上的透明性和可观察性研究

TAO 是 The ACE ORB 的缩写，是一个基于 CORBA 标准和 RT-CORBA 标准的中间件平台，是 CORBA 协议和 RT-CORBA 协议的一个具体实现。TAO 使用 ACE 框架内的组件和模式 ，是一个高性能、实时的(QoS)的分布式应用平台。TAO 可以实现远程对象调用(而不用关心如何去进行对象定位)、跨平台应用、跨编程语言应用、跨硬件平台和通讯协议应用等。

TAO 可以与其它公司的 ORB 很好的互操作，比如 Orbix、JacORB、ORB Express、VisiBroker 等。

国内对于 TAO 平台上的透明性和可观察性研究始于 2004 年国防科技大学 IKE2 项目[12]，该项目基于 TAO 平台开发应用级的可互操作的实时分布式应用，对应用系统的开发需要对 TAO 平台应用系统的透明性和可观察性有深入的了解。随后，南京航空航天大学直升机旋翼动力学国家重点实验室为了解决飞行控制系统在不同计算机、不同操作系统、应用不同语言编写的组件之间的通信也研究了 TAO 及其事件服务机制[13]，这种研究主要定位在消息的可观察性上。武汉数字工程研究所探讨了 TAO 的架构完成了基于 TAO 中间件舰载作战指挥系统的构件化设计与实现[14]。2007 年，山东大学控制科学与控制工程学院以 ACE/TAO 作为开发平台，构建了一个基于 CORBA 技术的异构机器人互操作系统，实现了分布

式环境下具有不同硬件结构、操作系统、通信协议和编程语言的机器人之间的相互通信[15]，为多异构机器人的协作提供了最佳的技术路线。2013 年，电子科技大学研究了 ACE/TAO 使用的 IDL 编译器 TAO_idl 编译出来的桩和框架的透明性，提出了用 C 语言设计实现的 IDL 编译器三模块设计模式的思想[16]。

4) DCOM 平台上的透明性和可观察性研究

DCOM 平台是一系列微软的概念和程序接口，利用这个接口，客户端程序对象能够请求来自网络中另一台计算机上的服务器程序对象，因此 DCOM 一种面向对象的中间件平台。DCOM 基于组件对象模型(COM)，COM 提供了一套允许同一台计算机上的客户端和服务器之间进行通信的接口。DCOM 的升级版本是 COM+，而 COM+综合了 COM、DCOM 和 MTS 技术要素，把 COM 组件提升到应用层而不再是底层的软件结构，它通过操作系统的各种支持，使组件对象模型建立在应用层上，把所有组件的底层细节留给操作系统，因此，COM+与操作系统的结合更加紧密。

由于DCOM只能运行在Microsoft的操作系统上，这很大程度上限制了DCOM/COM+的应用和研究。

国内对于 DCOM 平台上的透明性和可观察性研究始于 1997 年每周电脑报发表的一篇文章《“对象”之战升温——JavaSoft 要在 JavaBeans 中建立一种新型的对象模型 Microsoft 则计划做一个 Java 版本的 DCOM》，自此吸引了一些对象中间件的研究者对 DCOM 的原理和机制(包括透明性和可观察性)进行研究。经过长达近 20 余年的研究与实践，广大对象中间件的研究者也逐渐对 DCOM/COM+的透明性和可观察性有了一定的研究，例如，空军工程大学针对目前串口通信组件设计中的异步调用与并发运行的问题以及功能上局限在串口管理或简单数据收发的缺陷，采用 DCOM 提出了一种基于异步缓冲区模板类和线程管理器的设计方案[17]。

2013 年，和利时集团提出 DCOM 等组件在工业控制信息传输中的安全性问题，引起了社会较大的关注。

5) ICE 平台上的透明性和可观察性研究

ICE 平台是由 ZeroC 公司的分布式系统开发专家实现的一种高性能、面向对象的中间件平台。它号称标准统一，开源，跨平台，跨语言，分布式，安全，服务透明，负载均衡，面向对象，性能优越，防火墙穿透，通讯屏蔽。ICE 提供了完善的分布式系统解决方案，适合所有的异构网络环境、提供了客户端和服务器端的完全分离、ICE 采用软总线的机制、ICE 面向对象将所有应用看作是对象及相关操作的集合，构建在 ICE 之上的分布式系统的对象的获取只取决于网络的通畅性和获取服务对象特征的准确程度，而与对象的位置以及对象所处的设备环境无关。

ICE 提供了简单的对象模型和类型系统，精简而强大的运行时 API，简单的语言映射，紧凑高效并可扩展的协议，丰富的客户端调用和服务器端分派方式，完善的安全解决方案，大量高效而实用的服务和工具。基于这些，ICE 特别适合对技术和性能要求都很高的分布式系统开发。现在 ICE 已经被很多大公司采用，作为安全、伸缩性强的底层通信平台。

国内对于 ICE 平台上的透明性和可观察性研究始于 2003 年马维达在程序员杂志上发表的一篇文章《叛之冰：Internet Communications Engine》[18]，这引起了广大对象中间件研究者对 ICE 平台上的透明性和可观察性研究的兴趣。2004 年西南科技大学的刘裕和吴坚进一步对 ICE 进行了深入的剖析，从中间件的技术角度详细比较和分析了 ICE 与现今作为工业标准的中间件技术 COBRA 之间的异同点，阐述 ICE 作为新型中间件的优点和优势[19]。中南大学无机非金属材料研究所研究了基于 ICE 中间件开发多智能体系统的方法[20]，实现了多智能体控制系统与神经网络、模糊控制和专家系统技术的融合。中国海洋大学信息科学与工程学院应用了 ICE 中间件通信技术在异构三维仿真系统环境中实现高效的数据传输[21]，在功能上利用仿真数据为驱动实现多种空间任务的实时仿真。

针对配网自动化主站系统具有分布式结构、应用服务多、系统复杂等特点，2014 年，山东科汇电力

自动化有限公司的赵义奎等人提出了一套基于 ICE 中间件的配网自动化主站分布式应用管理框架[22]，描述了框架的组成结构及各个部分的实现原理。

国外对于中间件平台应用系统的透明性和可观察性研究起步较早，基础较坚实。为了便于比较，我们也将国外关于中间件平台应用系统透明性和可观察性的研究分为 ACE 平台上的透明性和可观察性研究、TAO 平台上的透明性和可观察性研究、DCOM 平台上的透明性和可观察性研究、ICE 平台上的透明性和可观察性研究，其中 PCDP 平台版权归中冶赛迪集团公司所有，目前没有对外开源，因此在国际上没有公开研究。

1) 国外 ACE 平台上的透明性和可观察性研究

在国外，ACE 早在 1999 年就被应用于移动通信中，这项研究是基于对 ACE 平台上消息的透明性而进行的。Johansen 等人在 1999 年第 19 届 IEEE 分布式计算大会上正式提出基于 ACE 的移动通信思想[23]，在当时引起了不小的反响。随后，Kutlusan 等人撰文介绍了 ACE 的事件和命名服务[24]，由于 ACE 的事件和命名服务卓越的性能引起了对象中间件研究者们研究 ACE 透明性和可观察性的兴趣。

2010 年，Chen 等人在研究 ACE 消息通信机制的基础上，设计了一个基于 ACE 的移动 IPC 平台[25]，与当时非 ACE 的 IPC 平台相比，这种平台在可移植性、兼容性方面具有较大的优势。Henfri 等人研究 ACE 的事件调度、事件过滤、数据传输等特点，并将 ACE 的事件通道作为可重用组件开发了无人机的分布式控制系统[26]。

2) 国外 TAO 平台上的透明性和可观察性研究

TAO 是 ACE 在实时领域中的版本，大多用于嵌入式领域。在国外，对于 TAO 平台上的透明性和可观察性研究起源于美国国防部先进计划预研局(DARPA)和美国空军研究实验室(AFRL)于 1998 年首次提出的 OCP 项目[27]；在对中间件平台上应用系统的透明性和可观察性深入研究的基础上提出了一种分层次的体系结构，其中任务规划和情景感知在最高层次，飞行控制(包括增稳控制)在最低层次，中间层次的控制器用于匹配模式之间的转换和提供容错重构控制，利用层次结构控制器有助于时间基准不同的控制组件集成，有助于低层连续动力学组件与高层离散事件组件之间同步协调作用。Pruett 等人在实施 OCP 项目时利用了 TAO 中间件平台[28]，并且提出了新的改进框架，这种框架对应用系统而言更加清晰更加透明。Schrage 等人研究了 TAO 事件通道机制、跨平台机制、C++封装包、IDL 接口语言[29]，开发出基于 TAO 的分布式环境下的 Yamaha R50/RMAX 直升机自动控制模型。Wills 等人研究了 TAO 在 Linux 操作系统上的消息传输(可观察性)问题[30]。

3) 国外 DCOM 平台上的透明性和可观察性研究

由于 DCOM 只能运行在 Microsoft 的操作系统上且 DCOM 源代码不开放，因此国外对于 DCOM 平台上透明性和可观察性研究也相对较少。

Choi 和 Wu 早在 1998 年就对 DCOM 和 CORBA 进行了对比研究，提出了负载均衡、扩大共发度、精简通信模型等问题。Saleh 和 Justo 在 2000 年研究了 DCOM 的服务质量问题，提出了不少质疑。Ntawanga 等人从用户体验的角度研究了基于 DCOM 的电子商务网站开发的可行性。

4) 国外 ICE 平台上的透明性和可观察性研究

国外对于 ICE 平台上透明性和可观察性的研究相对较多，下面列举几个有代表性系统和研究。

2004 年，Henning 在国际期刊 IEEE Internet Computing 发表论文《A new approach to object-oriented middleware》用了大量篇幅介绍 ICE 中间件平台[31]，此文后来被翻译成多国语言，引起了对象中间件研究者们对 ICE 透明性和可观察性的关注。随后不久，HP、Naval Undersea Warfare Center、Solution Space (新加坡)、Seagha Analytical Engineering, Inc.、Skype、Baosteel (宝钢)、Lockheed Martin、Baosight (宝信软件)等公司纷纷宣布其分布式通信产品均使用 ICE 平台来搭建。

2007 年，Reggiani 等人研究了 ICE 平台在生产制造领域中的应用[32]，论述了 ICE 能够满足生产制造领域中过程控制的性能和功能需求，是一个可以信赖的中间件平台。

Villanueva 等人认为基于 ICE 中间件平台开发通信产品是一个低成本的方案[33]。人们认识到 ICE 不仅能支持 PC 终端的开发还能支持微电子、嵌入式 JAVA 虚拟机甚至低端的 FPGA 产品的开发。

2011 年，Chavas 等人在研究了 ICE 平台的特性的基础上，将 ICE 中间件平台用于核物理实验中[34]。

综上所述，国外在中间件的透明性和可观察性研究方面起步早、基础牢，并且取得了许多工程实际应用。

PCDP、ACE/TAO、DCOM、ICE 对透明性和可观察性提供了不同程度的支持，其中 PCDP 支持得最好功能也最强大并已具备基于 B-S 架构的透明性和可观察性功能，ACE/ICE 支持得较好但是十分复杂不便于使用；DCOM 在迁移透明性的支持上最差，它仅限于在 Windows 平台上的迁移，对于性能透明性的支持也不如其他好，但是 DCOM 使用最简单这是其他不可比拟的；ICE 简化了 ACE/TAO 的架构，支持了八种透明性但是功能上不如后者强大，是一种轻量级的 ACE/TAO。具体分析见表 1。

2. 中间件的透明性

对于中间件的透明性，我们分别从访问透明性、位置透明性、并发透明性、故障透明性、迁移透明性、性能透明性、扩充透明性、复制透明性八个方面进行综述。

1) 访问透明性

访问透明性是指用户的应用系统对远程服务的访问与对本地服务的访问所采取的操作是一致的。

分布式平台应用系统在物理上包括本地服务和远程服务，但是对应用系统来讲就好像只有一个本地服务。如何对本地服务和远程服务采用相同的访问机制，这便是访问透明性所要解决的问题。

我们通过一个简单的示例来说明中间件平台的访问透明性。

通过中间件平台 PCDP，我们开发出两个应用系统 Svrtst_c_d 和 TestSvrtst_c_d，其中 Svrtst_c_d 用于提供加法服务(Add)、TestSvrtst_c_d 作为客户程序来使用加法服务。

如图 1 所示，在本地计算机(localhost)上 TestSvrtst_c_d 成功地使用了 Svrtst_c_d 提供的加法服务。

当我们把 Svrtst_c_d 移植到另一台计算机上，其 IP 为 10.65.55.49。按照相同操作方法，我们得到了相同的结果。

通过以上示例，我们不难发现 CISDI 中间件平台提供了访问透明性的功能。

作为一种指标，要求中间件平台提供的访问透明性不仅支持局域网、城域网和广域网，甚至还支持 3G/4G 移动网络。

2) 位置透明性

位置透明性是指应用系统在使用远程服务时无需了解远程服务的位置。不管应用系统位于本地还是远程，它访问平台的功能以及与平台上另一个应用系统进行通信都是畅通无阻的，也即是说，应用系统不需要知道自己所处的是什么平台、是本地的还是远程的并且通信的连接过程和数据的转发过程完全由开发平台和运行平台来完成。位置透明性分为物理位置透明性与网络位置透明性。

物理位置透明性只需要知道服务程序的 IP 地址和端口就可以使用该服务；网络位置透明性不需要知道彼此的 IP 地址和端口、不需要知道彼此间怎样建立通信、不需要同时处于运行状态就可以通信(这些工作由智能代理来处理)，是一种平台应用系统松耦合的连接方法。

例如，对于同一个服务 Svrtst_c_d，应用系统 TestSvrtst_c_d 不需要知道服务所处的平台是 Windows 还是 Linux、也不需要知道服务是在远程还是在本地，只需要知道服务程序的 IP 地址和端口就可以使用该服务了。

Table 1. Comparison of transparency and observability in different middleware platform
表 1. 不同中间件平台透明性和可观察性的比较

中间件平台	透明性	可观察性
PCDP	通过消息自动存储、消息自动统计、自动检测自动报警等支持透明性	通过 ECharts 组件支持系统可观察性和应用可观察性
ACE/TAO	通过 CORBA 模型支持透明性	不支持系统可观察性
DCOM	通过组件模型支持透明性	不支持系统可观察性
ICE	通过 Publisher/Subscriber 模型支持透明性	不支持系统可观察性

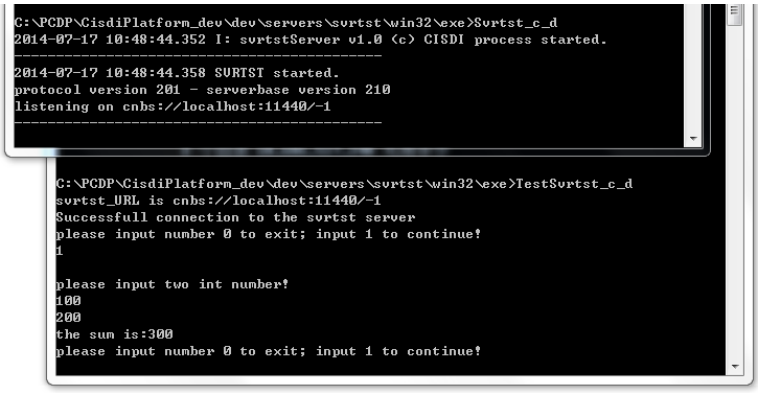

Figure 1. Access transparency of CISDI middleware platform
图 1. CISDI 中间件平台的访问透明性(本地访问)

3) 并发透明性

并发透明性是指多个用户的应用系统可同时访问同一服务而不会引起服务的不一致性。通过平台自身的信号量和通信基础设施来完成应用系统之间数据互斥、共享与传输。

图 2 显示了中间件平台 PCDP 提供的并发性。

在图 2 中，一共有三个应用系统，其中最上面的一个作为服务程序，其 IP 为 localhost，端口为 11440，下面的两个作为客户程序。显然，两个客户程序可以同时地、独立地访问服务程序。

作为中间件平台的并发性指标，要运行于平台之上的多个应用系统可以同时访问同一个服务而不出错误。并发透明性可以通过多线程机制来实现。

4) 故障透明性

故障透明性是指用户在使用服务时不会因为计算机或网络通信发生故障而产生服务的不一致性。也就是说，当客户程序发生故障时，服务器程序应该意识到这种故障并释放掉保存于内存中的连接状态信息和收发数据信息。这种技术被称为“掉队”检测和消除技术。

图 3 显示了 PCDP 中间件平台提供的故障透明性功能。

在图 3 中，上面的进程是服务程序进程，下面的客户程序。当客户程序出现故障时，服务器程序仍然能够向其他客户程序提供服务，并且第一个客户程序在故障解决后又能正确地与服务程序连接。PCDP 中间件平台对于故障的处理是打印出相关的提示信息然后中断执行。

5) 迁移透明性

迁移透明性又称移植透明性，是指服务器提供的远程服务不会因为客户端程序迁移到新的系统上而发生改变。

例如，客户端程序在 Windows 7 上能够正常地使用服务器提供的远程服务，当客户端程序移植到

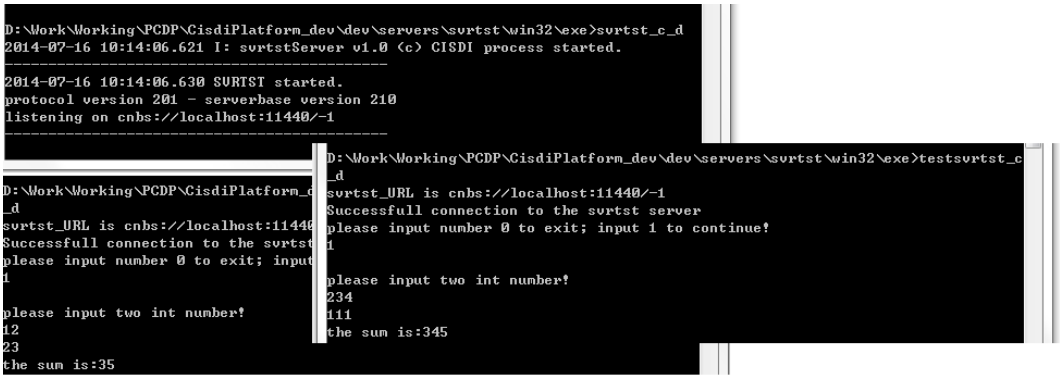

Figure 2. Concurrency transparency of CISDI middleware platform
图 2. PCDP 中间件平台并发性

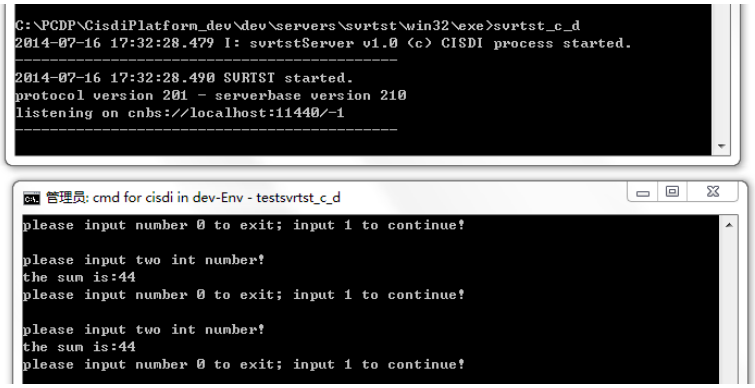

Figure 3. Failure transparency of CISDI middleware platform
图 3. CISDI 中间件平台故障透明性

Linux 系统上时仍然能够使用服务器提供的远程服务。另外迁移透明性也体现在不同操作系统版本之间的一致性，例如当客户端程序从 Windows 7 系统移植到 Windows XP 系统上时，其使用服务器提供的远程服务应该保持不变。

PCDP 中间件平台目前的版本是“中间件平台 for windows 1.0”，它可以实现在不同 Windows 操作系统平台与 Unix 版本之间的迁移。

6) 性能透明性

性能透明性是指用户在使用远程服务时无需了解远程服务的性能，而只需要向远程服务器发起服务请求。

性能透明性使得服务器端性能的改进和客户端的性能独立起来，无论客户端性能如何低下都能享受服务器端提供的高性能服务。

性能透明性从总体上降低了整个分布式网络的硬件投资费用。

PCDP 中间件平台具有性能透明性，使得转炉炼钢过程中的氧枪控制和转炉底吹控制可以分布在不同的设备上，提高执行设备的性能，而远程控制设备的性能保持不变，从整体上提高控制的灵活性降低设备总投资。

7) 扩充透明性

扩充透明性是指远程服务功能的扩充对于客户端用户是透明的，也就是说，客户端调用服务器服务的接口不变而其功能增强了。

在需求文档和开发文档的约束下，提高远程服务的功能可以提升中间件平台的服务水平和质量。利用扩充透明性就可以使客户端的用户无需重新学习新的功能接口而得到更强的功能。

PCDP 中间件平台具有扩充透明性，这体现在该中间平台的许多接口还有进一步改进和功能提升的空间，而这些改进和功能提升均在需求文档和开发文档约束的范围内。

8) 复制透明性

使用资源的多个实例提升可靠性和性能，而用户和应用程序员无须知道副本的相关信息。

在分布式系统中，可以把一个场地的数据复制到其他场地存放，应用程序可以使用复制到本地的数据在本地完成分布式操作，避免通过网络传输数据，提高了系统的运行和查询效率。但是对于复制数据的更新操作，就要涉及到对所有复制数据的更新。

3. 中间件的可观察性

可观察性是指中间件平台和平台上的应用系统是可以被观察的，其中前者是后者的基础后者是前者的外在表现。通过观察平台我们能够了解消息传输的内部机制；通过观察应用系统我们能够了解消息传输的外部表现。

可观察性分为两个层次，一是系统可观察性另一个是应用可观察性，其中前者是从系统层面观察系统对象、消息以及状态的形成和传输机制，后者是从应用本身观察对象、消息的发送、传输、接收和显示。

系统可观察性通过绘出系统的类图、顺序图等可视化手段来展示；而应用可观察性则通过打印消息的发送者、接收者、发送时间、内容等手段来展示。

图 4 是一个 IPC 系统可观察性的示例。IPC (Inter-Process Communication，进程间通信)是一个进程间通信的传输控制协议。

在图 4 中，tcPort 是传输控制的接口，它是 IPC 的服务窗口。tcPort 依赖于 tcServer 和 tcPortH，前者是传输控制协议的实现类，后者是 tcPort 的一个对象句柄。tcPort、tcServer 和 tcPortH 构成了依赖关系。pcPort 与 ostream、istream、ostrstream、istrstream 四个流相关联，通过这四个流实现 IPC 信息的序列化，pcPort 和它们构成了关联关系。

图 5 是 PCDP 中间件平台的应用可观察性示例。

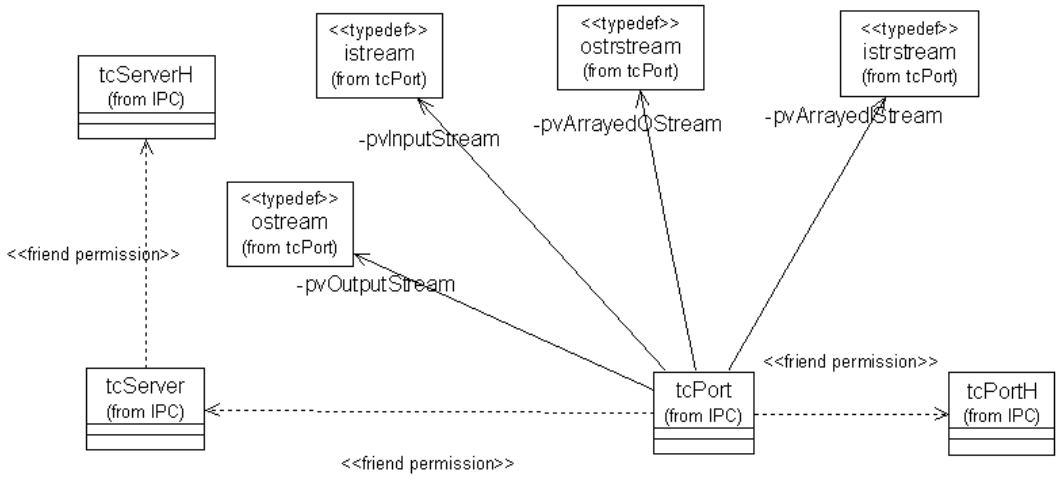

Figure 4. System observability example of IPC
图 4. IPC 系统可观察性示例

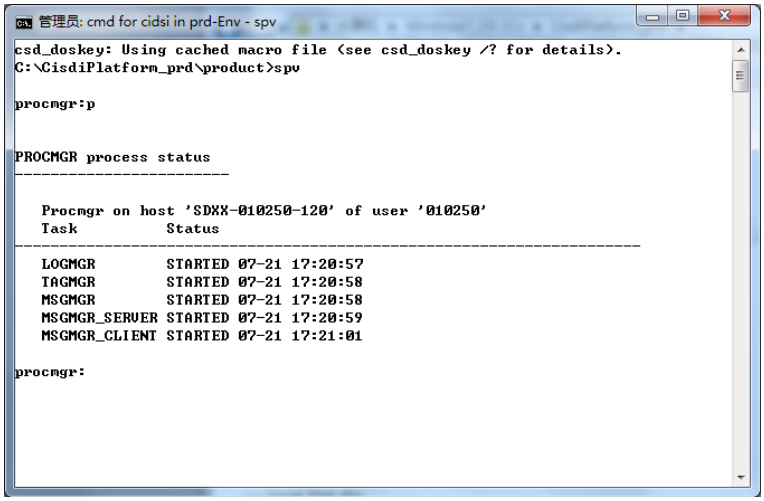

Figure 5. Application observability example
图 5. 应用可观察性示例

在图 5 中，我们可以看到有五个进程，它们的状态均为启动(STAETED)，而这五个进程是建立在平台之上的应用系统，通过一定的手段或工具来查看应用系统的状态是应用可观察性研究范畴之一。

此外，从不同的角度来研究平台和平台上的应用系统的可观察性还可以分为类(对象)可观察性、消息可观察性、关联关系可观察性、依赖关系可观察性等，这些共同组成了平台和平台上应用系统的可观察性。

4. 结论

中间件的透明性和可观察性是中间件的关键技术之一，对于未来的大规模物联网和云计算软件而言，透明性和可观察性尤为重要，因此我们介绍了中间件的透明性和可观察性。中间件的透明性分为访问透明性、位置透明性、并发透明性、故障透明性、迁移透明性、性能透明性、扩充透明性和复制透明性八种；中间件的可观察性分为系统可观察性和应用可观察性。我们对主流的中间件平台 PCDP、ACE、TAO、DCOM 和 ICE 进行了透明性和可观察性研究，并以 PCDP 中间件平台为例详细论述 PCDP 中间件的透明性和可观察性。

中间件透明性和可观察性技术的未来发展方向是面向行业、智能化、基于组件的个性化中间件。

基金项目

重庆市基础与前沿研究计划项目(cstc2015jcyjA40006)；高效能服务器和存储技术国家重点实验室开放基金项目(2014HSSA08)；重庆市沙坪坝区科委项目(KP201502)。

参考文献 (References)

[1] 匡付华, 程朋胜, 胥布工. OPC 及 DasRdb 数据库在城市轨道交通中的应用[J]. 自动化仪表, 2015, 36(6): 36-38.

[2] Coulouris, G., Kindberg, J.D.T., Blair, G. 金蓓弘, 马应龙, 等, 译. 分布式系统: 概念与设计[M]. 北京: 机械工业出版社, 2013.

[3] Yin, Y.F., Wang Y. and Wang Y.X. (2007) Analysis on Open Control Platform. *Proceedings of the* 2007 *IEEE International Conference on Mechatronics and Automation*, Harbin, 5-8 August 2007, 3371-3376. http://dx.doi.org/10.1109/ICMA.2007.4304104

[4] Yin, Y.F., Zeng, Y.F. and Guan, H.C. (2014) A Weighted Dynamic Information Systems Reduction Method. *Intelligent Automation and Soft Computing*, **20**, 101-114. http://dx.doi.org/10.1080/10798587.2013.828907

[5] Yin, Y.F. and Guan, H.C. (2013) Dynamic Software Testing and Evaluation with State Space Method. *Journal of Testing and Evaluation*, **41**, 403-408. http://dx.doi.org/10.1520/JTE20120207

[6] 李剑锋. 物联网中间件在公共领域中的应用[J]. 电脑编程技巧与维护, 2015(12): 34-34, 44.

[7] 马维达. ACE 与 GoF 设计模式——Adapter 模式在 ACE 内存管理类中的应用[J]. 程序员, 2003(10): 92-93.

[8] 王辉, 徐锦法, 高正. 基于事件的无人直升机分布式飞行控制系统实现[J]. 南京航空航天大学学报, 2005, 37(2): 212-216.

[9] 高强文. 用 ACE 实现可靠消息传送[J]. 电脑编程技巧与维护, 2005(8): 16-20.

[10] 李明, 高海军, 吴海涛. 基于 ACE 的监控软件设计[J]. 工业控制计算机, 2005, 18(11): 58-62.

[11] 崔海涛, 王国栋. 基于 ACE 的中厚板轧机二级控制系统改造[J]. 武汉科技大学学报, 2013, 36(1): 17-20.

[12] 冯润明, 王国玉, 黄柯棣. TENA 中间件的设计与实现[J]. 系统仿真学报, 2004, 16(11): 2373-2377.

[13] Yin, Y.F., Wang, X.N. and Guan, H.C. (2014) Online Joint Control Approach to Formation Flying Simulation. *IEEE Aerospace and Electronic Systems Magazine*, **29**, 24-36. http://dx.doi.org/10.1109/MAES.2014.130161

[14] 刘海. 基于 TAO 中间件的舰载作战指挥系统构件化设计与实现[J]. 舰船电子工程, 2006, 26(1): 13-17, 24.

[15] 周风余, 宋洪军, 刘涛, 冯国瑞, 牟龙芳, 李贻斌. 基于中间件技术的异构机器人系统设计及实现[J]. 山东大学学报(工学版), 2007, 37(3): 41-45, 50.

[16] 李颖, 胡明. 基于 C 语言实现的 IDL 编译器[J]. 计算机技术与发展, 2013, 23(3): 5-9.

[17] 徐嵩, 孙秀霞, 董文瀚, 李湘清, 李大东. 基于 DCOM 的无人机地面站串口通信模块设计[J]. 计算机工程与设计, 2011, 32(9): 3213-3217, 3234.

[18] 马维达. 反叛之冰: Internet Communications Engine[J]. 程序员, 2003, (8): 91-94.

[19] 刘裕, 吴坚. 中间件技术与 ICE [J]. 微机发展, 2004, 14(10): 37-39.

[20] 王海东, 廖小文, 李海亮, 张海. 水泥生产过程分层递阶多智能体控制系统[J]. 中国水泥, 2006, (6): 59-63.

[21] 李泽朋, 马纯永, 陈戈. 空间应用仿真支持平台的研究与实现[J]. 计算机工程与设计, 2013, 34(6): 2109-2113.

[22] 赵义奎, 李惠民, 王乐挺, 徐丙垠. ICE 在配网自动化主站中的应用[J]. 电力系统保护与控制, 2014, 42(1): 102-107.

[23] Johansen, D., Marzullo, K. and Lauvset, K. (1999) An Approach towards an Agent Computing Environment. *Proceedings of the* 19*th IEEE International Conference on Distributed Computing Systems*, Austin, 31 May-4 June 1999, 78-83. http://dx.doi.org/10.1109/ecmdd.1999.776418

[24] Kutlusan, A., Altmidort, N., Oruk, T. and Duman, A. (2000) A Combat Management System Middleware Based on CORBA. *Proceedings of the DOA* '00 *International Symposium on Distributed Objects and Applications*, Antwerp, 21-23 September 2000, 345-354. http://dx.doi.org/10.1109/DOA.2000.874206

[25] Chen, D., Chang, G.R., Li, J.J. and Wang, X.W. (2010) Design and Implementation of a Portable ACE-Based IPC Platform. *Proceedings of the* 2010 *International Conference on Intelligent System Design and Engineering Application* (*ISDEA*), **1**, 502-505. http://dx.doi.org/10.1109/ISDEA.2010.121

[26] Henfri, B.E., Ha, T.K., Seo, Y.B. and Choin, J.W. (2008) Communication Architecture for AUV Test-Bed Using ACE/TAO Real-Time Event Channel. *Proceedings of the* 27*th Chinese Control Conference*, Kunming, 16-18 July 2008, 260-264.

[27] Paunicka, J.L., Mendel, B.R. and Corman, D.E. (2001) The OCP—An Open Middleware Solution for Embedded Systems. *Proceedings of the American Control Conference*, Arlington, 25-27 June 2001, 3445-3450. http://dx.doi.org/10.1109/ACC.2001.946163

[28] Pruett, S.H., Slutz, G.J. and Paunicka, J.L. (2003) Hardware-in-Loop Simulation Using Open Control Platform. *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit*, Austin, 11-14 August 2003, 1-11. http://dx.doi.org/10.2514/6.2003-5759

[29] Schrage, D.P. and Vachtsevanos, G. (1999) Software-Enabled Control for Intelligence UAVs. *Proceedings of the* 1999 *IEEE International Symposium on Computer Aided Control System Design*, Hawai'i, 22-27 August 1999, 528-532. http://dx.doi.org/10.1109/CACSD.1999.808703

[30] Wills, L., Sander, S. and Kannan, S. (2000) An Open Control Platform for Reconfigurable, Distributed, Hierarchical Control Systems. *Proceedings of the* 19*th Digital Avionics System Conference* (*DASC*-2000), Philadelphia, 7-13 October 2000, 1-8. http://dx.doi.org/10.1109/dasc.2000.886955

[31] Henning, M. (2004) A New Approach to Object-Oriented Middleware. *IEEE Internet Computing*, **8**, 66-75. http://dx.doi.org/10.1109/MIC.2004.1260706

[32] Reggiani, M., Zuppini, M. and Fiorini, P. (2007) A Software Framework for Process Control in the Agroindustrial Sector. *Proceedings of the IEEE International Conference on Automation Science and Engineering CASE*, Scottsdale, 22-25 September 2007, 164-169. http://dx.doi.org/10.1109/coase.2007.4341806

[33] Villanueva, F.J., Villa, D., Moya, F., Barba, J., Rincon, F. and Lopez, J.C. (2007) Leightweight Middleware for Seamless HW-SW Interoperability with Application to Wireless Sensor Networks. *Proceedings of the DATE* '07 *Design, Automation & Test in Europe Conference & Exhibition*, Nice, 16-20 April 2007, 1-6.

[34] Chavas, J., Chateau, F., Druillole, F., Pollacco, E., Sizun, P., Usher, N. and Anvar, S. (2011) Mdaq-D3, a C++ Distributed Driver Development Framework Used in a Nuclear Physics Experiment. *Proceedings of the* 2011 *IEEE Nuclear Science Symposium and Medical Imaging Conference* (*NSS*/*MIC*), Valencia, 23-29 October 2011, 179-182. http://dx.doi.org/10.1109/NSSMIC.2011.6154474