

中图分类号: TP29

论文编号: 10006SY2041109

北京航空航天大学
硕士学位论文

数字孪生平台研发及其
数据和知识建模方法研究

作者姓名 蒋亮亮

学科专业 电子信息与科学技术

指导教师 胡远奇

培养学院 集成电路科学与工程学院

A Dissertation Submitted for the Degree of Master

Candidate: veypi jiang

Supervisor: yuanqi hu

School of Integrated Circuit Science and Engineering

Beihang University, Beijing, China

中图分类号：TP29

论文编号：10006SY2041109

硕 士 学 位 论 文

数字孪生平台研发及其 数据和知识建模方法研究

作者姓名 蒋亮亮

申请学位级别 硕士

指导教师姓名 胡远奇

职 称

学科专业 电子信息与科学技术

研究方向 数字孪生

学习时间自 2020 年 9 月 1 日

起至

2023 年 1 月 1 日止

论文提交日期 (年) 年 (月) 月 (日) 日

论文答辩日期 (年) 年 (月) 月 (日) 日

学位授予单位 北京航空航天大学

学位授予日期 (年) 年 (月) 月 (日) 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写过的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____

日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____

日期： 年 月 日

指导教师签名：_____

日期： 年 月 日

摘 要

数字孪生技术作为一种将物理对象与其虚拟副本相结合的方法，已在智能制造、智慧城市、物联网（IoT）、大数据和人工智能（AI）等领域取得显著成果。本文旨在深入探讨数字孪生技术在实际应用中所面临的挑战及其解决方法。首先，本文回顾了数字孪生的基本概念和发展历程，重点关注了其在仿真制造行业的应用，并分析了在产品全生命周期中的关键作用。接着，本文详细讨论了数字孪生技术目前所遇到的困境，如数据质量与精度、可视化与交互性提升、模型与知识分析存储、数学分析与仿真技术、模型快速构建技术、多领域应用实践以及实时性应用等方面的挑战。

为解决这些挑战，本研究从三个方面提出了高效性能、易用性、以及快速构建场景的通用数字孪生平台方案，并尝试进行实际应用。首先，在数据质量和精度方面，本文探讨了一种自主分布式实时消息通信网络，以确保在硬件网络内满足消息实时性要求，并在整个系统内保证消息的发送、分发、存储和响应效率。其次，在模型与知识分析存储方面，本研究提出了一套基于图谱或语义构建逻辑模型的系统，该系统为用户提供方便的场景内各节点交互响应逻辑编辑以及虚拟场景内仿真分析逻辑。通过参数化设计模型及模型间交互逻辑，并设计合适的模型存储结构，大大简化了模型与知识在各端的存储和传输。最后，在实时性应用方面，本研究基于前两个技术，探索了系统全生命周期应用方案技术，从历史数据分析、实时场景内数据监控与逻辑响应，到未来部分数据推演仿真，实现单个孪生场景的历史、现在与未来数据统一整合，大幅拓展了数字孪生技术的应用范围。

尽管如此，在实际应用场景中，我们仍需面临诸如海量信息的建模与处理、模型的结构化和专有化等复杂现实问题。在单一场景中，可能需要构建数千个数字孪生实体并处理每秒数十万级别的数据。如何利用现有知识快速推演模型，以及如何关联、分析和响应海量异构数据，依然是在探索研究前述两个技术方向时需要努力解决的现实难题。

总之，数字孪生技术在多领域实践中的应用和挑战仍有待深入研究。本文通过回顾数字孪生技术的发展历程、分析其在仿真制造行业的应用和挑战，以及探讨针对这些挑战的解决方案，旨在为数字孪生技术的进一步研究和应用提供借鉴。随着数字孪生技术的不断发展，有望在智能制造、智慧城市等领域实现更高效、可持续和人性化的解决方案。

关键词：数字孪生, 分布式消息网络, 知识模型, 数据推演

Abstract

Key words: digital twins

目 录

第一章 绪论	1
1.1 数字孪生概述	1
1.2 研究背景及现状	2
1.3 研究意义	4
1.3.1 数据困境	5
1.3.2 模型困境	6
1.3.3 实践困境	6
1.4 研究目标	7
1.5 研究方案	7
第二章 OneDT 主系统设计	10
2.1 主系统设计原则	10
2.2 整体结构设计	11
2.3 功能设计	12
2.4 数据库设计	13
2.4.1 设计要点	13
2.4.2 时序数据库设计	14
2.5 文件组织结构	15
2.6 3d 渲染引擎	16
2.6.1 渲染技术基本流程	16
2.6.2 存储格式	17
2.6.3 渲染技术优化	19
第三章 分布式消息网络设计	22
3.1 概述	22
3.2 通讯机制	22
3.3 通讯协议	24
3.3.1 消息帧	24
3.3.2 消息类型	25
3.3.3 握手协议	25
3.4 路由算法	26
3.5 分布式选举协议	27

3.6 硬实时通信	27
3.7 终端通信	29
3.8 自组网测试	30
3.9 性能分析	30
第四章 时序模型	32
4.1 状态模型	32
4.2 事件模型	33
4.3 时序模型	33
4.3.1 统计学预测	34
4.3.2 循环神经网络预测	35
第五章 知识模型设计	36
5.1 基本模型	36
5.2 程序接入机制	38
5.3 代码解析机制	38
5.4 公式解析算法	38
5.5 代码规则案例-碰撞检测	39
5.5.1 点与 AABB	39
5.5.2 AABB 与 AABB	40
5.6 公式规则案例-刚体运动力学模型	40
5.6.1 空间位置	41
5.6.2 空间姿态	42
第六章 应用案例	44
6.1 自组织网络-树莓派集群	44
6.2 时序模型-应力应变分析	44
6.2.1 有限元仿真	44
6.2.2 时序模型预测	45
6.2.3 融合模型	46
结论	48
参考文献	49

插图目录

图 1	f22 试飞与数字模型 [1]	3
图 2	产品/生产/设备三维数字孪生模型 [2]	3
图 3	模型层次图	8
图 4	文章结构图	9
图 5	系统结构图	10
图 6	平台主体结构	12
图 7	自定义 Mesh 存储结构	18
图 8	图像数据	19
图 9	直线网格	19
图 10	结构化网格	20
图 11	非结构化点	20
图 12	多边形数据	20
图 13	非结构化网格	20
图 14	非结构化网格优化	20
图 15	数据对象与算法过滤器生成可视化管道 Pipeline	21
图 16	分部性能要求	23
图 17	通讯网络	23
图 18	节点握手协议	26
图 19	选举协议	27
图 20	测试设备图	29
图 21	消息编码速率分布	30
图 22	消息编码速率拟合曲线	30

图 23 消息分发速率压测结果 31

图 24 RNN 精简结构图 35

图 25 知识模型分类 36

图 26 知识模型上传过程 37

图 27 复合材料 FEM 与时序分析融合 44

图 28 某焊接设备半年采集数据 45

图 29 光学应变采集设备 46

图 30 320 点位应变采集图 46

图 31 复合板材焊接过程的孪生模型 47

表格目录

表 1	通讯帧	24
表 2	消息编码	31
表 3	节点状态转移表	32
表 4	复杂系统状态转移表	34
表 5	时序分析性能	46

主要符号对照表

HPC	高性能计算 (High Performance Computing)
cluster	集群
SMP	对称多处理
MBD	基于模型的设计
API	应用程序编程接口
Digital Twin	数字孪生
entity	实体对象
physical entity	物理对象
virtual entity	虚拟对象
physical domain(space)	物理域 (物理空间)
L	角动量
I	惯性张量
τ	力矩
ω	角速度

第一章 绪论

1.1 数字孪生概述

数字孪生技术作为一种将现实世界与虚拟世界紧密结合的方法，近年来在工业、智能制造、医疗、城市规划等领域取得了显著的进展，三维数字化定义的产品模式已逐渐成熟，并多次证实其效益。

数字孪生技术是一种将物理对象、流程或服务数字化建模的方法，通过利用这些模型来推测其行为和性能。这一全面的数字化方法能够实现从产品生命周期的开始到结束的全方位描述与分析，实现物理世界的可视化、可交互和可预测虚拟化。在数字孪生中，物理系统被视为网络，由一系列包含物理机制、虚拟特征和人际交互关系的特征点组成的对象集合。

数字孪生技术不仅能够映射物理世界中的各种物体，如发动机、电场、建筑物和城市等。此外，数字孪生还可用于映射流程，并通过收集和分析数据来预测它们的行为和性能，利用现实世界数据创建可以预测产品性能或流程行为的虚拟模型。

通过整合物联网、人工智能和软件分析等技术，数字孪生技术可提高效率和精度，在现代工程中推动创新和性能提升。随着机器学习和大数据等领域的发展，数字孪生技术的应用范围将逐渐扩大，为人们提供更全面的数字化解决方案。

数字孪生技术通过数据馈送映射物理实体，并对许多工业领域产生了颠覆性影响。德国信息技术和新媒体协会预测，在制造业市场中，数字孪生的价值巨大，到 2025 年将超过 780 亿欧元。Gartner 已经连续两年（2016、2017）将数字孪生列为十大战略技术发展趋势之一。2017 年 11 月，世界最大的武器制造商洛克希德·马丁公司将数字孪生列为未来国防和航空航天工业的六大顶级技术。

数字孪生技术的发展和應用为制造业带来了巨大的潜力，推动了产品设计、制造和维护的效率和质量的提升。随着技术的不断发展和创新，数字孪生技术将在制造业市场中发挥越来越重要的作用，引领未来制造业的发展方向。

然而在实际应用场景中，数字孪生面临着诸多复杂的现实挑战，如海量信息的建模与处理、模型的结构化与专有化等问题。在单一场景中，可能需要构建数千个孪生体并处理每秒数十万级别的数据。因此，本篇文章主要关注以下几个研究内容：

- 1、 根据已有知识构建模型：研究如何利用已有知识来构建数字孪生模型，包括物理原理、经验数据和专家知识等多方面的信息。
- 2、 描述模型特征和状态：探讨如何准确描述数字孪生模型的特征和状态，包括物理

属性、虚拟特征以及与人交互关系等。

- 3、 关联、分析、响应大量数据：研究如何有效地关联、分析和响应海量数据，以实现数字孪生技术在实际应用场景中的高效运行。这可能包括数据预处理、特征提取、模型优化和实时响应等方面的技术研究。

针对这些挑战，未来研究可以结合现有的技术进展，如机器学习、大数据处理和高性能计算等，以期解决数字孪生技术在实际应用中所面临的问题，进一步提升其在各领域的实际应用效果。

1.2 研究背景及现状

数字孪生 (Digital Twin) 的概念最早出现在 2003 年，由 Grieves 教授在密歇根大学的产品生命周期管理课程中提出。2010 年，“Digital Twin”一词在 NASA 的技术报告中被正式提出，并被定义为“集成了多物理量、多尺度、多概率的系统或飞行器仿真过程”。2011 年起，美国国防部将数字孪生的概念引入了航天器的健康维护等问题解决中。

NASA 将数字孪生定义为：以飞机或系统为导向，充分利用最佳物理模型、传感器和运行历史数据，集成多学科和多尺度概率仿真过程，映射对应物理飞机的状态。

美国空军对数字孪生的定义是基于 NASA 将“面向飞机或系统”扩展到“面向完成建模的系统”和“映射”扩展到“映射和预测”，并给出了机身图形的孪生内涵示例（包括几何模型、材料性能、飞行试验数据、制造/检验/维修数据、气动模型、有限元模型、损伤演化模型）。

全球知名的 IT 研究与顾问咨询公司 Gartner 在连续两年中将数字孪生列为年度十大战略科技发展趋势之一。世界最大的武器制造商洛克希德·马丁公司在 2017 年 11 月将数字孪生列为未来国防和航天工业六大顶尖技术之首。同时，2017 年 12 月 8 日，中国科协智能制造学术联合体在世界智能制造大会上将数字孪生列为世界智能制造十大科技进展之一。

知名公司如达索、西门子等已初步将数字孪生技术应用于产品设计、生产和维护过程。2018 年，ANSYS 发布了 ANSYS 19.1 软件，推出了首款针对数字孪生体的产品软件包——ANSYS Twin Builder，进一步推动了数字孪生与仿真技术的融合应用。2019 年，加拿大国家研究委员会（NRC）与 AFRL 和澳大利亚国防科学技术小组（DST）一起，审查和评估了 ADT 框架对加拿大皇家空军（RCAF）机队的潜在适用性，利用 ADT 框架对 CF-188 的内侧前缘襟翼进行了试验演示。此外，在航空航天领域，传统的验证-管理-维护理念已经无法满足下一代飞行器的需求，发展同步演化仿真技术成为新的技术趋势，NASA 和美国空军研究实验室等也将该技术应用于具体型号研发，并荣获 2016

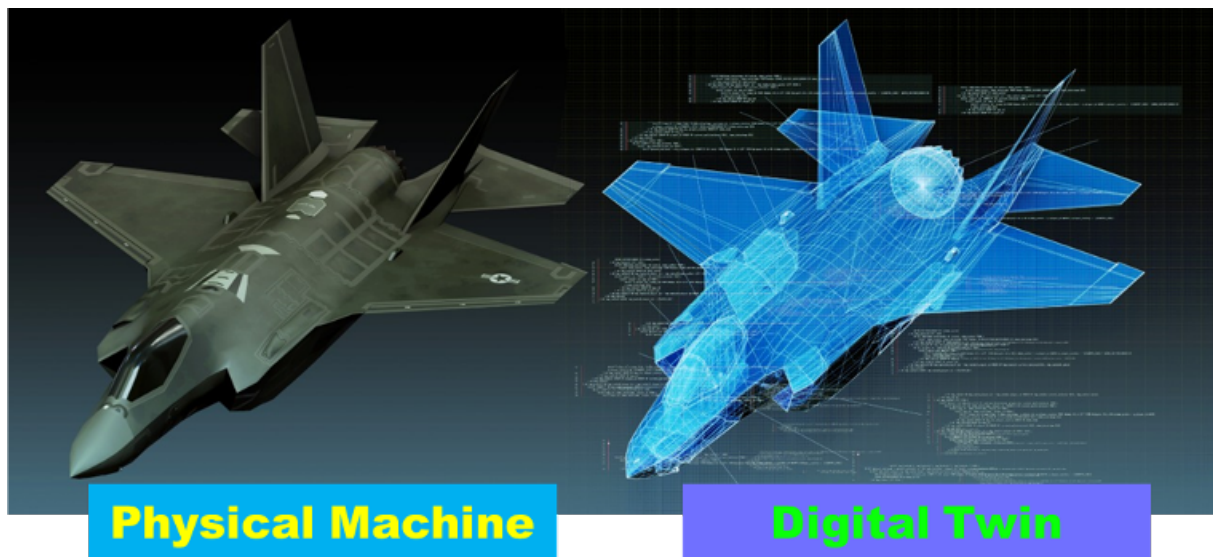


图1 f22 试飞与数字模型 [1]



图2 产品/生产/设备三维数字孪生模型 [2]

年美国国防制造技术奖。

庄存波等人对数字孪生的定义是产品数字孪生，是指产品物理实体在信息空间中的工作状态和工作进度的全要素重构和数字映射。它是一种集成的多物理、多尺度、超逼真的动态概率仿真模型，可用于仿真、诊断、预测和控制真实环境下产品物理实体的实现过程 [3]。

Grievess 等人对数字孪生的定义是一组虚拟信息结构，对复杂产品从微观原子层次到宏观几何层次全面描述潜在生产或实际制造产品。构建一个数字孪生的最好结果是，任何可以通过检查实际制造的东西，产品获得的信息也可以从它的数字孪生获得。在研究基于数字孪生的复杂产品/复杂产品/系统行为分类系统问题预测时，明确指出构建数字孪生的目的是基于仿真预测，最大程度地消除复杂产品/系统”不可预测和不受欢迎的

行为”，以避免不可知的灾难性问题。以火箭发射为例，在虚拟空间中，模拟真实火箭的数字孪生，即使失败了，也可以用新的虚拟空间快速替换，继续启动 [4]。

数字孪生是以数字化方式创建物理实体的虚拟实体，借助历史数据、实时数据以及算法模型等，模拟、验证、预测、控制物理实体全生命周期过程的技术手段，建立在对一系列多维度的大规模的、实时的真实世界的的数据测量，并辅以一定的物理和行为建模进行状态复现和预测的概率过程。

现阶段，除了航空航天领域，数字孪生领域还应用于电力、矿产采集、城市管理、工业制造等，特别是在智能制造领域，数字孪生被认为是一种实现物理世界、信息世界及人类意识世界交互的重要手段。

如上所述，数字孪生技术的研究主要呈现于以下四点：

- 1、 多学科交叉：数字孪生研究涉及计算机科学、工程、制造、物流等多个领域，需要多学科知识的共同支撑。
- 2、 技术创新：数字孪生的核心技术包括数据采集、模型构建、仿真分析等，学术界正致力于研究更高效、精确的技术方法。
- 3、 应用广泛：数字孪生技术已应用于工业制造、建筑、交通、能源等众多行业，学术界正关注其应用效果，以便为实践提供理论支持。
- 4、 标准与规范：学术界正努力制定数字孪生的相关标准和规范，以便于技术在不同领域的应用和推广。

从数字孪生技术的众多描述中，可以发现其部分特性与仿真技术相似。仿真技术采用包含确定性规律和完整机理的数字模型来模拟现实世界的物理现象，其目的在于依赖精确的模型以及完整的信息和环境数据来分析物理世界的特性和参数，数字孪生技术在仿真的基础上增加了实时性和交互性，使得信息模型和物理模型能够相互关联和影响。

1.3 研究意义

数字孪生技术作为近年来在工业领域及其他行业迅速发展的创新技术，将物理实体、流程或服务的数字表征与现实世界紧密相连。该技术提供了一种在虚拟环境中构建、测试、优化和管理物理系统的新方法，对众多行业的发展具有重大意义，并被不约而同的视为未来发展的关键技术之一，这主要归功于在产品的设计、生产、运维、市场四大生命周期中，数字孪生技术都起到了重要推动作用。

首先，数字孪生技术能够提升产品设计和生产过程的效率。数字孪生技术可以帮助工程师在虚拟环境中对产品进行设计、仿真和优化。这种方式可以显著减少实际试验次数，降低试验成本，提高设计效率和产品质量。

此外，数字孪生技术还可优化生产流程，数字孪生技术可以实现生产过程的实时监控和分析，通过模型预测和调整生产参数，提高生产效率、降低生产成本。此外，数字孪生还可以协助企业制定更为合理的生产计划，提高资源利用率。

在运维阶段，数字孪生技术可以实时监测设备的运行状态，预测可能发生的故障，从而实现智能维护和及时排除故障。此外，数字孪生还可以进行远程诊断和维修，减少现场维护的成本和时间。

其次，数字孪生技术有助于企业更好地了解产品和服务在市场的表现。通过将现实世界数据与虚拟模型相结合，企业可模拟客户体验和产品性能，进而更好地了解客户需求和市场趋势，从而优化产品和服务，提高客户满意度。

同时在数字孪生技术研究过程中，涉及多个学科领域，如物联网、大数据、人工智能、工程学等。这种交叉性为不同学科间的研究合作提供了契机，有助于推动多领域知识的融合与创新。

最后数字孪生技术在落地应用时，涉及工业制造、建筑业、能源、交通、农业等多个领域，有助于推动相关产业的升级与创新。通过数字孪生技术，企业可以实现数字化、智能化生产，提高竞争力。

数字孪生技术在提升效率、改进产品设计和生产过程、提高客户满意度、降低成本和提高生产效率、产品运营和维护等方面具有显著的价值。随着技术的不断发展，数字孪生技术有望成为各行各业推动数字化转型和实现可持续发展的重要工具。

然而，在实际生产和应用过程中，数字孪生技术仍面临诸多困境。

1.3.1 数据困境

数据采集尺度或计量单位的一致性问题的，其中涉及物理数据、几何数据和时间数据等方面。例如，构建实物三维模型时，若坐标与计量单位不一致，可能导致不同模型之间无法融合，从而需增加数据接口并编写数据翻译器。此外，工厂内生产计划数据采集过程中，不同时间单位的数据可能导致数字孪生模型出现数据读取错误。

关于数据采集参数和格式的一致性问题，针对同一对象，多维虚拟模型采集的数据格式若不一致，或参数类型和数量不对等，可能导致不同模型在数据融合时出现问题，从而无法进行交互。

数据采集周期亦是一个关键问题。例如，生产设备数据产生通常以毫秒为单位，采集的数据驱动数字孪生模型时，往往需要扩大时间尺度，否则模型仿真运行压力过大可能导致崩溃。

除此之外，数据采集的稳定性和准确度、海量数据的存储和处理能力、通信接口协

议不一致、多源异构数据整合和数据歧义等问题均对数据采集和处理过程提出了挑战。为实现数字孪生技术的广泛应用，需要解决这些问题，确保数据的准确性、一致性和可靠性。

1.3.2 模型困境

在当前的数字孪生系统中，模型类别相对单一，专有化程度较高，且模型结构相对简单。虚拟世界的基础知识依赖较为有限，当系统从分析预测阶段向自主控制的智能化分析发展时，对现有基础模型知识库提出了大量需求。然而，由于知识库的数据结构和模型缺乏统一标准，多模型互操作困难，数据语义和语法不一致，因此标准化的知识图谱体系仍有待探讨。

本文将模型按层次划分为几何模型、物理模型和规则模型。几何模型和物理模型适用于虚拟世界，其主要作用在于帮助人们更直观地观察系统变化。规则模型涉及实体、虚体、系统和人之间的数据流通及逻辑判断。当前，规则模型的建立主要通过特定场景下的硬编码实现，这导致模型无法通用，且更新和建立模型过程相对复杂。为提高数字孪生技术的应用范围和效果，我们需努力解决这些问题，推动模型类别多样化和标准化。

1.3.3 实践困境

数字孪生技术是一种综合性技术，它涉及数据采集处理、数字孪生与数字模型、数字孪生与产品全生命周期管理（PLM）、数字孪生与大数据分析、数字孪生与信息物理系统（CPS）以及数字孪生与工业互联网等多个技术领域。为了推动数字孪生技术的发展，当前亟需在基础理论和相关技术融合方面取得突破。这包括开展设备泛在接入、工业通信协议适配、异构系统集成和虚实融合等核心关键构件的研发，以及突破多协议数据转换、海量异构数据汇聚、感知数据驱动和数字孪生精准映射等关键技术研究，从而促进数字孪生的应用。目前数字孪生应用主要集中在数字楼宇、数字城市等对真实世界的的数据复刻，如何做到数字世界对真实生产活动的正反馈在当前多技术多领域融合背景下仍是一个挑战。

同时，在多技术融合的背景下，数字孪生的商业模式也难以得到充分保障，实践中的重要性仍需关键突破和应用。从单一环境仿真到大世界综合场景应用，仍需要底层技术发展和相关数据标准统一。数字孪生的多系统特性既体现在物理空间，也体现在虚拟空间，在数据、模型和交互各环节都有所表现。数字孪生实质上是物理世界与数字世界的多维系统融合。首先，我们面临着物理世界中的多系统挑战。据不完全统计，制造业当前的设备数字化率约为 47%，局域联网率仅为 40%，可接入公网的比例约为 20%。因

此，底层运营技术（OT）与信息技术（IT）的融合仍然是一个极为关键的基础性问题。

针对以上三个问题，本文在结合实践形式和数字孪生技术特点，分析了相关技术理论，分别提出了相应的改进方案，探索数字孪生技术在全产品生命周期的应用前景和能力。

1.4 研究目标

数字孪生作为一种实现物理实体向信息空间数字化模型映射的关键技术，通过充分利用部署在物理系统各部分的传感器，对物理实体进行数据分析和建模，形成跨学科、多物理量、多时间尺度和多概率的仿真过程，从而反映物理实体在不同真实场景中的全生命周期过程。然而，在实际应用中，数字孪生面临着一些复杂的现实问题。目前较为成功的案例通常局限于特定场景和特定数据，尚未出现成熟的通用技术方案。针对前文总结的三个问题，本文分别提出相应的解决方案，以应对或部分解决实际应用中所面临的挑战。

- 1、数据困境：编写设计一套自组织分布式消息网络，部分节点实现硬实时通信，整体实现所有节点自组织形成分布式通信网络。
- 2、模型困境：针对规则模型编写设计一套通用机制，物理节点、虚拟节点、服务节点可分别使用和设计不同规则，用户生成规则可以选择内置逻辑、函数、真值表、代码等实现。
- 3、实践困境：以分布式消息网络为基础，编写设计模型通用特征，在系统所有层次都有公开设计的通信接口，任何节点可以选择使用非内置节点，使用第三方或者自行编写的逻辑，提供现有主流的 PLC 通信协议，并提供将任何协议转内置通信协议的方案或物理节点。以上述方案为基础，构建数据推演模型，整合历史、现在、未来数据，扩充数字孪生平台应用范围、降低应用难度。

1.5 研究方案

如图3所示，为改造的网络模型层次图，5/6/7 层模型分别为应对数据困境、实践困境、模型困境的解决方案。其中，5 层模型为分布式消息网络，6 层模型为节点状态与事件机制，7 层模型为知识模型，每一层次的实现皆依赖于上层次的功能和性能实现，针对这三层内容，本文以如图4结构编写该文章。

如图4所示为本文结构图，为解决数字孪生所面临的三个困境，根据现实需要研究设计相应的方案，并在最后给住性能测试和案例应用证明其可行性和实用性。在进行上述研究同时，还开发了 OneDT 数字孪生系统，主要涉及五大主体内容。这些内容为：主

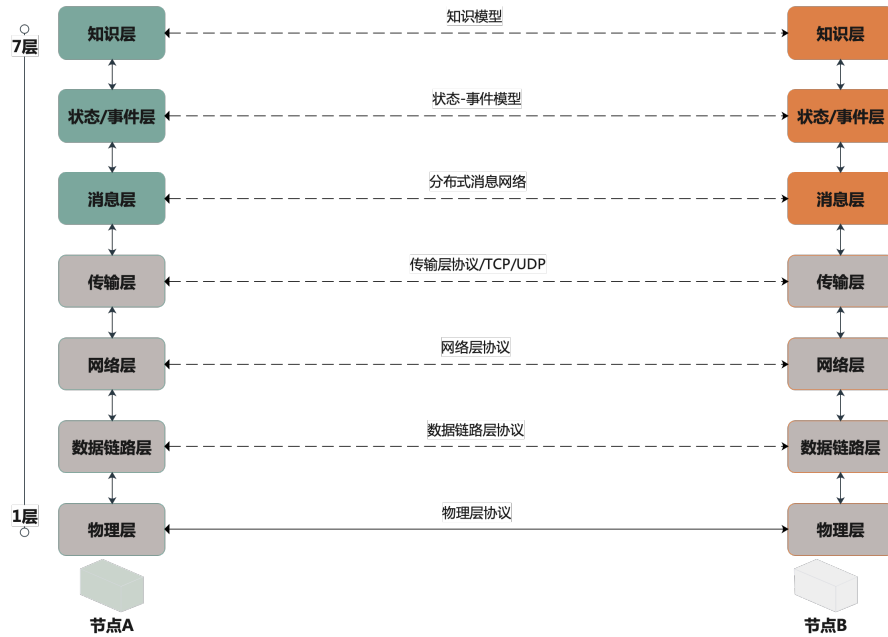


图3 模型层次图

体平台、渲染引擎、分布式消息网络、节点状态与事件机制，以及知识模型，该系统是开展上述研究的基础和载体。

1、 主体平台

主体平台是数字孪生系统的核心组成部分，负责整合各个子系统，以便在一个统一的环境中进行交互。它为用户提供了一个直观的界面，便于管理和监控数字孪生的实体。此外，主体平台还支持多种数据处理和分析功能，以帮助用户实现更高效的运营和决策。

2、 渲染引擎

渲染引擎是数字孪生系统的可视化组件，负责将数字孪生的实体模型转换为图像，以使用户更直观地观察和理解数字孪生的实体。不单纯同于游戏和三维设计软件的渲染引擎，因为模型数据中多了仿真数据和物理数据，同时系统有着实时孪生的需求，因此需要根据数字孪生需要对现有的渲染技术进行优化。

3、 分布式消息网络

分布式消息网络是数字孪生平台的通信基础设施，负责在各个节点之间传输数据和消息。它采用了先进的分布式技术，确保了平台在面对大规模数据和高并发请求时依然能够保持稳定的性能。分布式消息网络还具有很高的可扩展性和容错能力，可以轻松适应不断变化的业务需求。

4、 节点状态与事件机制

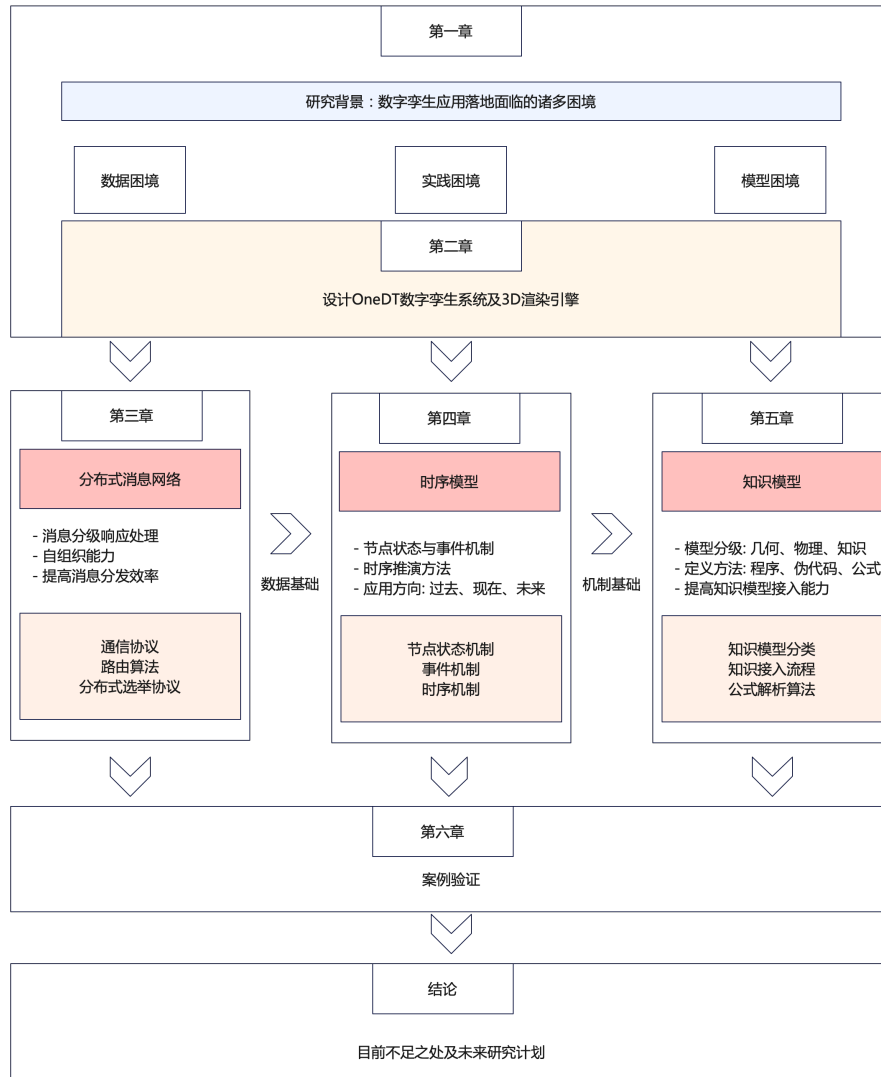


图4 文章结构图

节点状态与事件机制是数字孪生平台中的重要组成部分，负责实时监控和管理系统内的各个节点。它可以捕捉节点之间的状态变化，以及由此产生的各种事件，从而实现对整个系统的动态调整和优化。此外，节点状态与事件机制还为用户提供了丰富的实时数据和预警信息，有助于更加精确地掌握系统的运行状况。

5、知识模型

知识模型是数字孪生平台的智能核心，负责对各种数据和各个节点进行建模和分析，包含涉及节点所遵从的逻辑规则，是一种复杂系统交互规则，其实现时为用户提供更有价值的洞察和建议。采用了先进的仿真技术和机器学习技术，可以实现对复杂数据的处理和复杂场景的建模分析。

第二章 OneDT 主系统设计

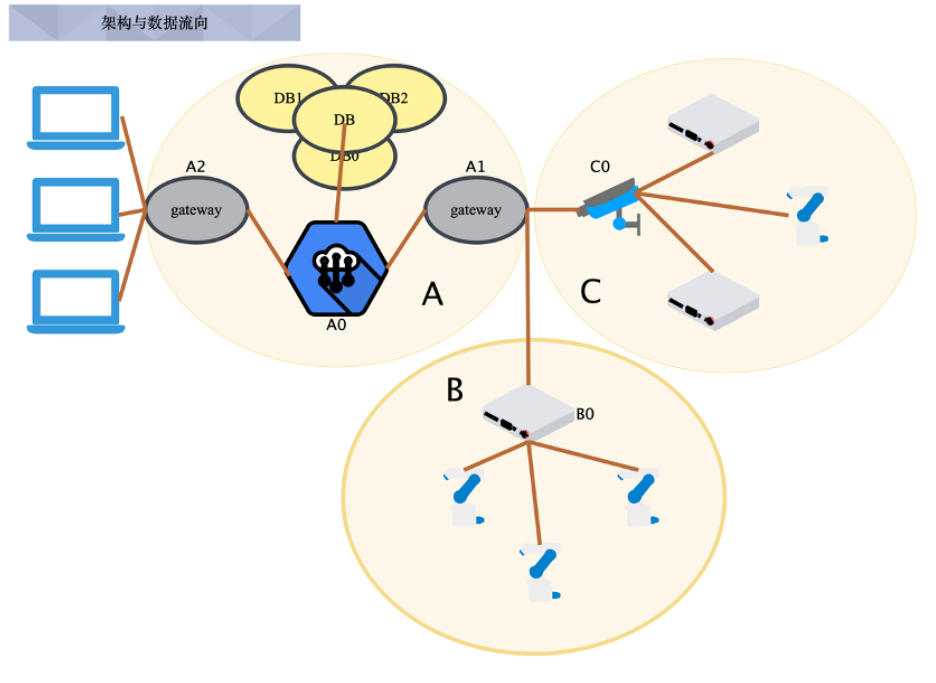


图 5 系统结构图

图5所示，A 代表数字孪生主体系统，它是数字孪生平台的核心组成部分，包括基本的数据存储、数据分析、核心消息节点、用户交互、模型展示和知识存储等功能。所有其他功能都依赖于主体平台进行显示和控制，是实现系统所有功能的基础和载体。BC 分别代表不同的孪生场景，每个场景皆有一个核心节点与 A 集群进行信息交互。

2.1 主系统设计原则

在设计主体平台 A 时，为了保障整体系统的可靠性和功能性，在设计每一个功能模块和子系统时需要遵循以下原则：

- 1、 高可用：高可用性是系统设计中的重要目标之一，旨在确保系统在面对故障、错误或意外情况时能够保持正常运行。在系统设计中要注意负载均衡、自动化任务设计、冗余备份、快速恢复、无状态设计等等问题。
- 2、 高性能：在本平台设计中，核心性能指标有消息网络的分发能力，硬实时消息的响应速度，主体平台对多种规则模型的处理速度，海量消息的存储能力，用户访问时模型的操作性能、仿真性能等。
- 3、 可扩展性：可扩展性是本平台设计中核心设计的一点，不单单指服务的可扩展性，

比如服务节点可以水平扩增等，还指集群网络的开放程度，能对接现有消息协议，开放消息网络对接和规则模型设计。

- 4、安全性：在保持消息网络的开发性上去保证整体平台、节点的安全性，在实践中，消息网络如何在开放协议的基础上保证通信安全。
- 5、单一职责原则：任何模块保持职责和功能的最小化，专一化，模块之间不含功能交集，这有助于确保系统的模块化和可维护性，并减少模块之间的依赖性和耦合度

2.2 整体结构设计

在设计数字孪生整体结构时，需要综合考虑用户使用、硬件通信、云服务特性，以及平台的通用性、可扩展性、容灾恢复和安全性等多方面因素。

整体上，采用微服务架构，将系统根据业务功能模块拆分成若干个独立的微服务，每个微服务都能独立部署和扩展。微服务架构有助于提高系统的可伸缩性和可用性，同时降低系统的维护成本。

在前端，采用前后端分离的架构，使用现代化的前端框架如 Vue，通过 API 连接到后端服务，实现前后端分离，以提高系统的可维护性和可伸缩性。

后端则采用分布式架构，将不同的业务模块拆分成独立的微服务，通过 API 或消息队列进行通信，以提高系统的可伸缩性和可用性。同时，利用缓存技术（例如 Redis）来提高系统的性能。

在数据库方面，采用分布式数据库架构，将不同的数据拆分到不同的数据库实例中，并通过数据分片技术实现分布式存储，从而提高系统的可伸缩性和性能。此外，利用读写分离技术将读操作和写操作分别路由到不同的数据库实例，以提高系统性能。针对不同类型的数据，采用相应的存储方式，例如将主体结构数据存入 MySQL，时序数据存入 Elasticsearch，非参数化模型存入文本或对象存储服务（OSS）中。

在安全设计方面，采用统一的非对称密钥认证机制，对用户访问和节点通信进行私钥本地解密远程密文认证，例如身份验证、访问控制、节点接入和数据加密等，以确保系统安全性。

如图6所示，云平台主体结构中，通用数字孪生平台的架构设计需充分考虑多方面因素。通过采用现代化架构技术和安全机制，可实现高性能、可伸缩、可用和安全的系统。

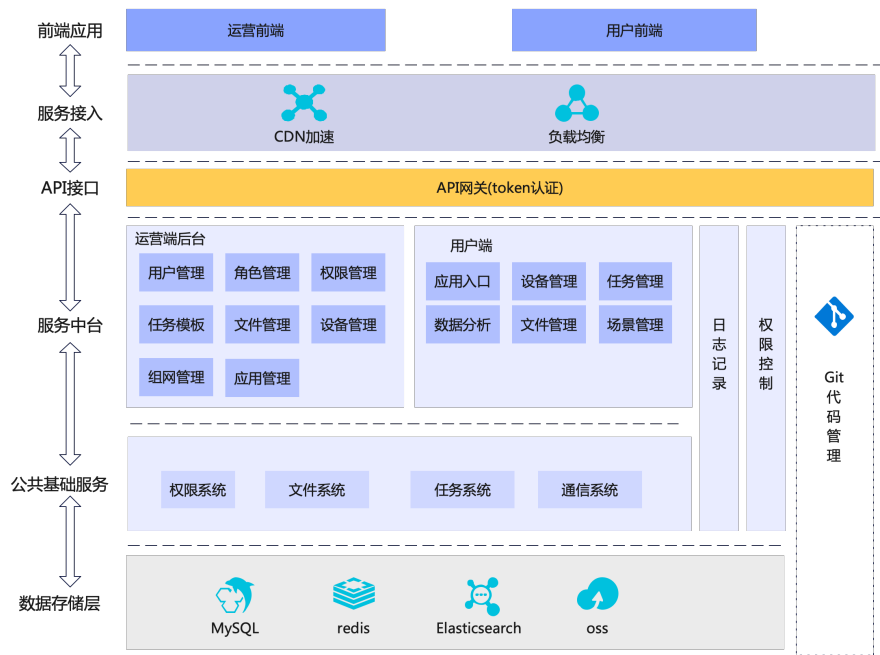


图 6 平台主体结构

2.3 功能设计

在设计该平台功能时，重点考虑其通用性和扩展性后，将平台功能点分为基础功能和拓展功能，拓展功能以本文提供的数据通信方案和规则模型方案为基础构建，由基础功能和额外编写的逻辑规则拓展而成，用户也可以此设计拓展功能。

- 1、数字孪生建模功能：提供用户友好的建模工具，支持各种模型类型和复杂性。用户可以通过可视化界面创建、编辑和删除模型。同时，平台应该提供模型验证、合法性检查、版本管理等功能。
- 2、规则编写：提供多种方式的规则输入体系，使得用户方便的去构建场景和节点交互逻辑和内置规则。
- 3、数据采集和集成功能：数字孪生平台需要支持各种数据源的集成，包括传感器、设备、网络、云服务等等，提供丰富的 API 和数据连接功能，以便数据源的快速接入和管理。
- 4、数据分析和可视化功能：数字孪生平台需要提供数据分析和可视化工具，以便用户对数字孪生系统中的数据进行分析 and 探索。平台提供可视化工具和分析功能，支持数据探索、数据可视化和可视化交互等等。
- 5、仿真和预测功能：数字孪生平台需要支持仿真和预测功能，以便用户可以基于数字孪生系统中的数据进行模拟和预测分析。平台应该提供各种仿真和预测工具，包

括数据建模、算法开发和模型训练等等。

- 6、安全性和可扩展性：数字孪生平台需要具备高可靠性和可扩展性。平台应该提供安全保障机制，包括数据安全、身份认证和访问控制等等。同时，平台应该具备可扩展性，以使用户可以轻松地扩展数字孪生系统的功能和规模。
- 7、API 和开放平台：数字孪生平台需要提供 API 和开放平台，以使用户可以轻松地开发自己的应用程序和服务。平台应该提供完整的 API 文档和示例代码，同时支持第三方开发者的集成和扩展。

2.4 数据库设计

在设计该系统数据库时，根据数字孪生场景需要和数据采集的一些特点，数字孪生平台的数据库设计需要考虑多个方面，包括数据库的范围和目的、结构和模式、安全性和可靠性、性能和扩展性以及数据采集和清洗等。在设计过程中需要综合考虑各个方面的需求，灵活应用不同的技术和方法，确保数字孪生平台的高效、安全、可靠运行。

2.4.1 设计要点

- 1、数据库的范围和目的：数字孪生平台通常需要处理大量的数据，包括实时数据、历史数据、静态数据等。因此，需要确定数据库的范围和目的，包括存储数据的类型、数据的来源和流向、数据的格式等。
- 2、数据库的结构和模式：为了支持数字孪生平台的多样化需求，数据库需要具有灵活的结构和模式。可以采用关系型数据库或非关系型数据库，或者二者结合使用。同时，需要确定数据库的表结构、字段类型、主键、外键等。
- 3、数据库的安全性和可靠性：数字孪生平台通常涉及到重要的实时数据和设备信息，因此需要确保数据库的安全性和可靠性。可以采用多重备份、加密、权限控制等方式来保证数据的完整性和保密性。
- 4、数据库的性能和扩展性：数字孪生平台需要支持高效的数据处理和分析，因此需要考虑数据库的性能和扩展性。采用分布式数据库、内存数据库等技术来提高性能，并采用分区、分片等方式来实现扩展性。
- 5、数据库的数据采集和清洗：数字孪生平台需要从多个来源获取数据，并进行数据清洗和处理。因此需要设计适合的数据采集和清洗模块，确保数据的准确性和一致性。

2.4.2 时序数据库设计

数字孪生平台除了大量结构化的数据需要使用关系型数据库存储外，还需要存储大量的时序数据，例如 200 个采集点位每天就可以产生 4GB 数据，因此需要设计一个专门用于存储时序数据的数据库用以存储和查询。时序数据有着非常强的特殊性，根据其特点，时序数据库着重优化设计以下几个方面：

1、 数据模型

确定时序数据结构。时序数据可以表示为一个包含时间戳、度量值和标签（例如设备 ID、位置等）的数据点。数据模型需要能够支持对这些数据点的高效插入、查询和删除操作。

2、 数据压缩

时序数据库需要处理和存储大量的数据。为了节省存储空间和提高性能，采用 Gorilla 压缩算法对数据进行编码。

3、 数据分片和分布式存储

根据时序数据弱关系性的特点，非常适合进行数据分片和分布式存储。数据分片是将数据按照一定的规则分成多个子集，每个子集存储在不同的数据库节点上。这样可以将查询和写入操作的负载分散到多个节点，提高性能。

4、 索引和查询优化

时序数据索引较为固定，主要查询是根据时间戳和标签进行查询，为了提高数据查询效率，时序数据库需要对这些数据添加索引，以便快速定位数据。同时，时序数据库还需要对查询进行优化，以便提高查询性能。

5、 数据保留策略

由于时序数据具有时效性，可能需要根据数据的重要性和时效性来制定数据保留策略。数据保留策略可以按照时间或数据量来设置，例如保留最近一个月的数据或保留最近 1000 万条数据等。

6、 数据降采样和聚合

为了提高查询性能和降低存储需求，时序数据库可以支持数据降采样（downsampling）和聚合操作。数据降采样是将高频率的数据点合并为低频率的数据点，例如将每秒的数据点合并为每分钟的数据点。数据聚合是对一组数据点进行统计分析，例如计算平均值、最大值和最小值等。通过降采样和聚合，可以极大降低存储需求和提高查询性能。

2.5 文件组织结构

该平台在设计之初就考虑到涉及用户端、后端、数据库、硬件端，编写语言涉及 c, c++, python, go, rust, js, matlab 等，项目管理和组织也较为复杂，故根据项目各部分特点设计了如下的代码 2 级目录结构。

- /odtc: 主要为可视化前端代码
- /odtc/Common: 核心类
- /odtc/Filtering: pipeline 中数据过滤器
- /odtc/GenericFiltering: 连接到外部仿真器的包装器
- /odtc/GeoVis: 处理地形可视化的类
- /odtc/Graphics: 处理 3D 数据的类
- /odtc/GUISupport: ui 界面依赖库
- /odtc/Hybrid: 依赖于多个其他目录中的复杂类
- /odtc/Image: 图像处理过滤器
- /odtc/IO: 用于读写不同类型文件的类
- /odtc/Parallel: 并行算法处理，用于支持 MPI 计算
- /odtc/Rendering: 渲染类
- /odtc/Utilities: 工具类
- /odtc/VolumeRendering: 用于不同类型的 mesh 数据绘制
- /odtc/Wrapping: python 接口支持
- /webds: 为分布式消息网络的代码
- /webds/cfg: 配置库
- /webds/client: 不同语言的客户端，如 go/js/python/rust/c
- /webds/cluster: 分布式协议及集群节点管理
- /webds/command: 命令行支持，如启动集群，启动节点，扫描集群，扫描消息等
- /webds/conn: 节点链接管理类
- /webds/core: 核心类
- /webds/device: 设备依赖库
- /webds/doc: 文档
- /webds/errors: 错误处理
- /webds/libs: 通用依赖库
- /webds/message: 消息协议
- /webds/trie: 消息路由算法

- /odtlib: OneDT 系统节点的依赖库
- /odtlib/atom: 节点通信依赖
- /odtlib/iopack: 消息包依赖
- /odtlib/command: 命令行支持
- /odtlib/topic: 消息主题管理
- /odtlib/status: 节点状态管理
- /odtb: OneDT 系统后端
- /odtb/api: 后端接口代码
- /odtb/auth: 权限系统及安全协议
- /odtb/cfg: 配置库
- /odtb/doc: 文档库
- /odtb/lib: 通用依赖库
- /odtb/media: 媒体库
- /odtb/report: 自动化报告生成
- /odtb/models: 数据库模型
- /odtb/static: 静态文件
- /odtb/stage: 场景及设备管理
- /odtb/utlis: 工具类
- /odtb/ws: websocket 支持

2.6 3d 渲染引擎

3D 渲染技术是计算机图形学的核心技术，也是数字孪生技术用户交互技术中最为重要的一环，将现实世界的设备、系统或过程与虚拟模型相结合，并通过三维图像的形式呈现出来，方便使用者更直观地了解和探索模型的结构、状态和运行情况。现代 3d 渲染技术已经是一项非常成熟的技术，考虑到数字孪生技术不单纯是对现实世界的视觉数据复刻，在研究数值仿真计算的相关原理后对现有的渲染技术进行优化。

2.6.1 渲染技术基本流程

渲染技术最基本流程包括几何表示、坐标变换、光栅化、着色和后处理。

1、几何表示

渲染引擎首先需要对三维场景中的物体进行几何表示。这通常涉及将物体表面划分为多边形网格（如三角形），并将顶点、法线、纹理坐标等属性数据存储在顶点

缓冲区中。为了构建完整的三维场景，还需要添加光源、相机等元素。

2、坐标变换

坐标变换是将物体从一个坐标系映射到另一个坐标系的过程。这包括：模型变换：将物体从模型坐标系（Model Space）变换到世界坐标系（World Space）。观察变换：将物体从世界坐标系变换到视点坐标系（View Space，也称摄像机坐标系）。投影变换：将物体从视点坐标系变换到投影坐标系（Projection Space），并执行透视除法，将坐标规范化为归一化设备坐标（Normalized Device Coordinates, NDC）。

3、光栅化

光栅化阶段负责将变换后的三维几何数据转换为屏幕上的像素。这包括确定哪些像素位于三角形内部，并计算这些像素的深度值。为了消除隐藏面，渲染引擎通常使用深度缓冲区进行深度测试。

4、着色

在着色阶段，渲染引擎根据物体的表面属性、光源、相机位置等因素计算像素的颜色。光照模型（如 Phong 模型或 Blinn-Phong 模型）用于模拟环境光、漫反射光、镜面反射光等效果。现代渲染引擎通常使用可编程着色器（如顶点着色器、片段着色器等）来实现更复杂的着色效果。

5、后处理

在渲染流程结束后，可以对生成的二维图像进行后处理。这包括添加阴影、环境光遮蔽、光泽反射、景深、运动模糊等效果，以增强图像的真实感和视觉质量。

2.6.2 存储格式

三维模型格式是用于存储和表示三维模型数据的文件类型。这些格式包含几何形状、纹理、材质、动画以及其他与 3D 模型相关的信息。在计算机图形学、游戏开发、电影制作和其他领域，有多种不同的三维模型格式，它们各自具有独特的特点和用途，常见的模型格式有 obj、stl、ply、3mf、dae、fbx、glTF 等。

但在数值仿真领域，通用的存储格式为 vti、vtu 等，除了包含基本的三维模型，这些格式还用于存储数值仿真计算结果，如应力场、应变场、温度场等，且不包含纹理、材质、动画等信息，因此不适用于数字孪生技术的应用场景。

数字孪生时代的存储模型不单要包含以上内容，还需包含采集到的真实的物理信息，如多个点的温度、压力、流量等，因此需要自定义存储格式，以便于存储读取和分

析，如图7所示，定义了包含更多数据源的存储格式。

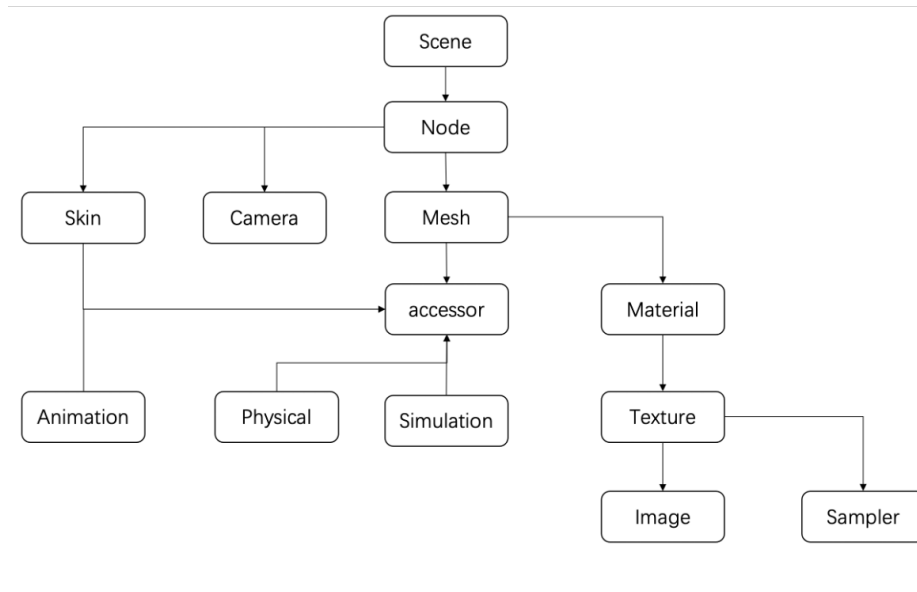


图7 自定义 Mesh 存储结构

- scene: 场景结构描述条目。它通过引用 node 来定义场景图。
- node: 场景图中的一个结点。它可以包含一个变换 (比如旋转或平移)，引用更多的子结点。它可以引用网格和相机，以及描述网格变换的蒙皮。
- camera: 定义了用于渲染场景的视锥体配置。
- mesh: 描述了场景中出现的 3D 对象的网格数据。它引用的 accessor 对象可以用来访问真实的几何数据。它引用的 material 对象定义了 3D 对象的外观。
- skin: 定义了用于蒙皮的参数，参数的值通过一个 accessor 对象获得。
- animation: 描述了一些结点如何随时间进行变换 (比如旋转或平移)。
- accessor: 一个访问任意数据的抽象数据源。被 mesh、skin 和 animation 元素使用来提供几何数据，蒙皮参数和基于时间的动画值。它通过引用一个 bufferView 对象，来引用实际的二进制数据。
- material: 包含了定义 3D 对象外观的参数。它通常引用了用于 3D 对象渲染的 texture 对象。
- texture: 定义了一个 sampler 对象和一个 image 对象。sampler 对象定义了 image 对象在 3D 对象上的张贴方式。
- image: 定义了一个 2D 图像。它引用了一个 bufferView 对象，来引用实际的二进制数据。
- sampler: 定义了一个 texture 对象的采样器。它定义了纹理的过滤器和张贴方式。
- simulation: 定义了仿真数据，包括仿真数据的类型、仿真数据的维度、仿真数据的

大小、仿真数据的范围、仿真数据的分辨率、仿真数据的数据类型、仿真数据的数据等。

- **physical**: 定义了物理数据, 包括物理数据的类型、物理数据的维度、物理数据的大小、物理数据的范围、物理数据的分辨率、物理数据的数据类型、物理数据的数据等。

因为上述格式根据 gltf 格式进行扩展, 因此可以使用 gltf 格式的解析器进行解析, 同时兼容现在主流的三维模型格式, 如 obj、stl、ply、3mf、dae、fbx、gltf 等, 但是对于仿真领域的存储格式需要单独开发解析, 如 vti/vtk/vtr/vtmb/vtmg/bthd 等 18 种格式, 该领域格式较为多且比较复杂, 因此目前仅适配对于最通用的 vti/vtu 格式, 后续会逐渐增加对于并行仿真计算格式的支持。

2.6.3 渲染技术优化

在现有游戏及相关 3d 渲染引擎中, 由于对于显示细节的追求, 对于非规格化平面, 经常会采取三角面进行细分, 因为三角形可以以任意精度表达任意平面, 在计算样条插值、各种复杂的数学变换、等值面等计算代价最小, 最简单。

但是在数字孪生系统中, 并不单纯的追求对于现实世界的视觉复刻, 而是需要对于现实世界的数值计算进行可视化, 因此在渲染技术上, 需要对于渲染技术进行优化, 以减少渲染时间, 提高渲染效率, 同时与数值仿真计算中单元相对应, 实现真实物理采集点-仿真计算单元-渲染单元三者统一。

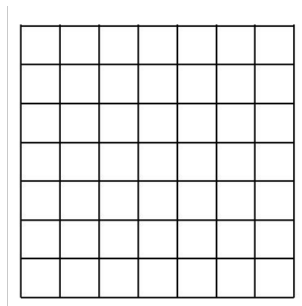


图 8 图像数据

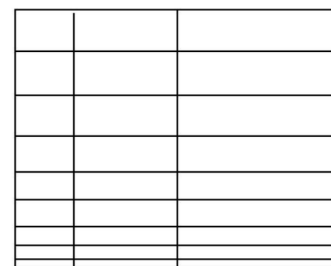


图 9 直线网格

如图8到图13所示, 图像数据, 直线网格, 结构化网格, 非结构化点, 多边形数据, 非结构化网格, 这些是在数值仿真计算中真实遇到的网格计算单元, 这些数据都可以通过不同的方式进行渲染, 在仿真计算中代表一个数据单元, 在传感器网络中可能代表一个有效采集点。在进行同步仿真时, 通常需要对模型进行降阶处理, 以满足实时仿真需要, 如图14所示, 从左到右为网格细化过程, 这个过程通常对应的是计算精度增加, 计算时间极大增加, 选择一些合适的模型降阶算法, 结合实际采集到的数据信息, 可以在

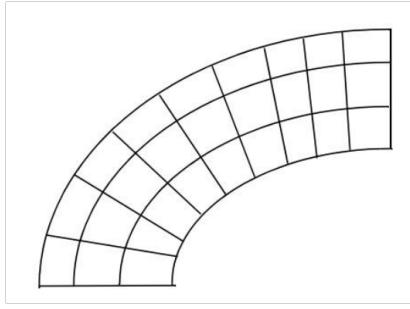


图 10 结构化网格

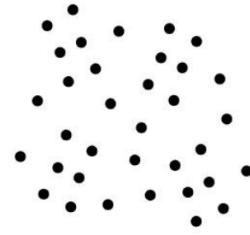


图 11 非结构化点

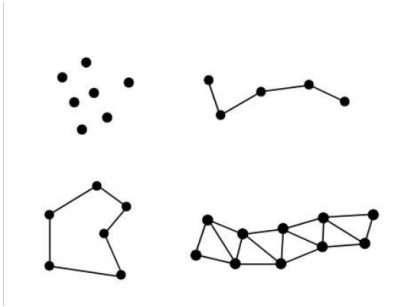


图 12 多边形数据

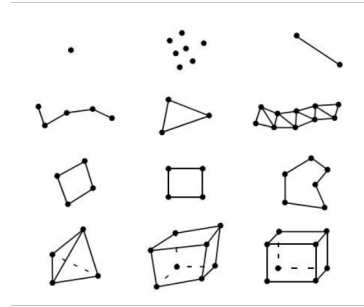


图 13 非结构化网格

保证 99 精度的情况下有效的降低模型单元数，极大提高计算效率，实现从右向左模型降阶。

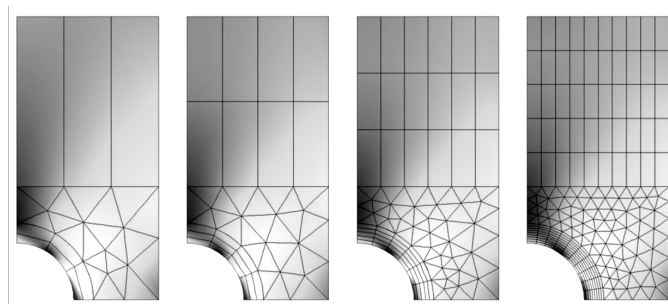


图 14 非结构化网格优化

如图15对于孪生数据的最终显示，除了简单的三维结构和贴图外，对于采集到的信息和仿真得到的信息以及根据历史数据预测的信息等多个数据源，选择合适的过滤算法，去显示最终的场变量信息，实现更有价值信息的实时显示，

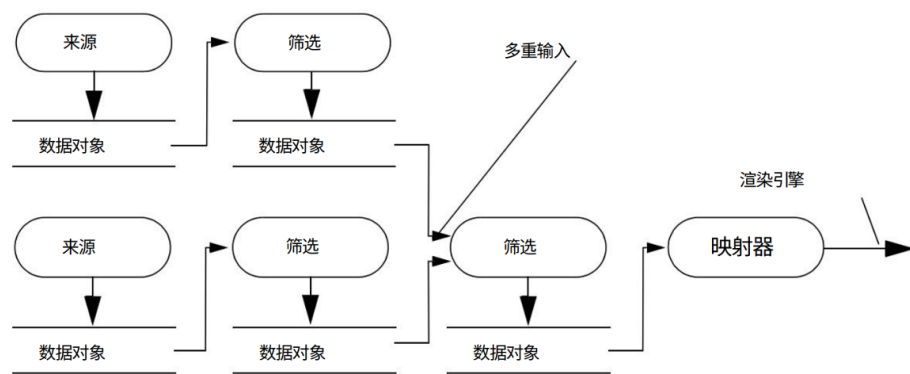


图 15 数据对象与算法过滤器生成可视化管道 Pipeline

第三章 分布式消息网络设计

3.1 概述

现代软件应用程序通常不是孤立存在的，而是依赖于服务或远程实体提供的信息。在这种分布式架构中，集成至关重要。近年来，消息传递已成为解决分布式系统挑战的主流方案，例如网络不可靠性、生产者与消费者之间的紧密耦合以及应用的异质性。得益于强大的社区支持以及对标准和集成的共同努力，消息代理现已成为许多项目和服务的传输层构建模块，近年来涌现出许多消息服务，如 MQTT、RabbitMQ、RocketMQ 和 Kafka 等。

现代分布式消息网络是一种基于分布式系统和消息传递模式的网络架构，支持高可用性、可扩展性和可靠性的通信。这种网络通常由多个节点组成，这些节点可以是物理机器、虚拟机或容器。

消息网络的核心是消息队列，它采用一种异步通信模式，在多个节点之间传递消息。消息队列通常包括生产者、消费者和代理（broker），其中生产者将消息发送到队列中，消费者从队列中接收消息进行处理，代理作为消息队列的中心节点，负责维护队列、路由消息和确保消息的可靠传递。

现代分布式消息网络通常采用发布/订阅模式或点对点模式进行消息传递。在发布/订阅模式中，消息生产者将消息发布到主题（topic），消费者订阅感兴趣的主题并接收相应的消息。在点对点模式中，消息生产者将消息发送到队列，只有一个消费者能够接收并处理该消息。

尽管这些消息框架已经非常成熟，但它们并不适用于本文的使用场景。这些框架通常用于云服务之间的消息通信，而单纯的工控 PLC 通信又显得过于僵硬，不适合构建信息物理系统（CPS）。目前尚无适合涉及大量设备、服务、人和虚拟实体四方消息互动的通信方案。如图 16 所示，这四方之间的消息传递需求和性能要求各不相同，因此尝试设计一套新的消息网络以解决数字孪生下的数据困境是非常有必要的。

3.2 通讯机制

图 17 为具体某一集群内的消息节点图，在同一子网下的 client 节点上线时会进行广播，发现内网节点后进行选举，产生一个子网内核心节点，其余节点连接该节点，核心节点在向上连接，最终链接到核心服务器集群，服务器集群也会选举产生一个核心节点，这个节点为整个网络的核心节点，用来维护节点状态并向下同步状态，整体网络中任意

节点下线，皆会触发重新选举或状态更新，核心节点因故障下线时，会导致当前层次子网重新选举，中断服务 1s，但是不影响其余层次网络集群功能，最终上线后会同步节点状态，实现整体网络的最终一致性。

对于任意节点来说，其上级节点其实与下级节点一致，在功能逻辑上存在上下级，但是在消息分发上不存在上级，连接的所有节点都是下级或者说同级，会维护一套消息表，记录消息标记和需要发往的目标节点。如节点 A 刚开始上线时该表为空，当 A 需要订阅消息时，就广播给周边的节点；B 收到记录下消息类型和目标 A 以及路径长度 1，在广播给除了 A 以外的临近节点，同时路径长度加 1；每个节点收到同步消息时根据表内数据，没有则添加，有则根据路径长度比较，短则替换目标，长则抛弃。这样，任意节点订阅消息时将会在整个消息网络广播其订阅记录，并在每个节点记录的是最短传播路径。当节点收到消息时，根据消息表内记录，存在标记则转发，不存在则抛弃。

3.3 通讯协议

3.3.1 消息帧

如表 1 所示，消息通讯协议采用了类似于以太网的帧结构，消息帧由前缀、长度、类型、标记、源节点、目标节点、数据组成，其中前缀用于判断消息帧的开始，长度用于判断消息帧的结束，类型用于区分消息类型，标记用于区分消息，源节点用于标识消息来源，目标节点用于标识消息目标，数据用于携带消息内容。

表 1 通讯帧

prefix		count
typ	tag	source
target		data
data		
....		
data		

- prefix(8bit): 帧前缀，用于判断消息帧的开始
- count(8bit): 消息长度
- typ(4bit): 消息类型，数字、字符串、二进制、文件地址、json 等
- tag(12bit): 消息标记
- source(8bit): 消息来源节点
- target(8bit): 消息目标节点

- payload: 搭载的数据

3.3.2 消息类型

在该分布式消息网络中, 采取的是发布订阅模式, 消息的类型分为两种, 一种是广播消息, 一种是订阅消息, 广播消息是指所有节点都会收到的消息, 订阅消息是指只有订阅了该消息的节点才会收到的消息。由于系统内部需要, 对部分消息频道进行了保留, 设计了不同功能目的的消息类型体系。

- /: 公众频道
- /inner: 用户保留频道, 不会进行广播
- /sys: 系统频道, 用于系统内部通讯, 不会触发任何用户定义的回调函数和事件
- /sys/base/log: 系统日志频道, 用于系统内部日志记录
- /sys/base/heartbeat: 系统心跳频道, 用于系统内部心跳检测
- /sys/base/auth: 系统认证频道, 用于系统内部认证
- /sys/topic/subscribe: 系统订阅频道, 用于消息订阅
- /sys/topic/cancel: 系统取消订阅频道, 用于取消消息订阅
- /sys/node/admin: 系统节点管理频道, 用于节点管理
- /sys/cluster: 系统集群频道, 用于集群管理
- /sys/cluster/id: 用于系统集群内部 id 交换
- /sys/cluster/level: 用于系统集群内部层级交换
- /sys/cluster/info: 用于系统集群内部邻居节点信息交换
- /sys/cluster/redirect: 用于通知子节点重定向到新的父节点
- /sys/cluster/heartbeat: 系统集群心跳频道, 用于集群内部心跳检测

3.3.3 握手协议

如图18所示, 节点握手协议采用了类似于以太网的握手协议, 同时加入了权限的认证机制。

- 主动节点握手请求: 主动节点向节点发送握手请求, 并声明自己的 id
- 被动节点握手响应: 被动节点向主动节点声明自身 id
- 被动节点握手认证: 被动节点根据对方 id 和公钥进行加密, 然后发送给主动节点
- 主动节点握手确认: 主动节点成功使用公钥和自身 id 解密, 确认对方身份, 并根据被动节点 id 和公钥进行加密, 发送给被动节点
- 被动节点握手确认: 被动节点成功使用公钥和自身 id 解密, 确认对方身份, 握手

成功，发送心跳包

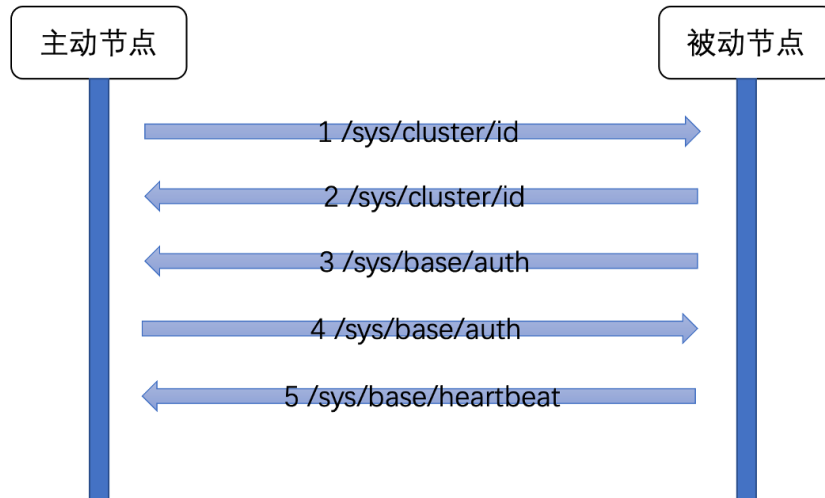


图 18 节点握手协议

3.4 路由算法

在订阅消息时存在父子订阅需求，对于很多节点来说可能需要订阅的是某个具体的消息，对于某些规则类节点或者虚拟节点来说，订阅的是某个类别的消息，所以设计树状订阅机制，其订阅子树仅会收到子树消息，订阅父节点则会收到下属所有子树消息，这样在进行消息分发时不能简单的通过判断标记想到与否转发，所以采用了 trie 树算法加速消息分发。

trie 树又被称为前缀树、字典树是一种用于快速检索的多叉树结构。字典树把字符串看成字符序列，根据字符串中字符序列的先后顺序构造从上到下的树结构，树结构中的每一条边都对应着一个字符。字典树上存储的字符串被视为从根节点到某个节点之间的一条路径，并在终点节点上做个标记“该节点对应词语的结尾”，正因为有终点节点的存在，字典树不仅可以实现简单的存储字符串，还可以实现字符串的映射，只需要将相对应的值悬挂在终点节点上即可。

Trie 的核心思想是空间换时间，有如下基本性质：

- - 根节点不包含字符，除根节点外每一个节点都只包含一个字符
- - 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串
- - 每个节点的所有子节点包含的字符都不相同

字典树能够利用字符串中的公共前缀，这样可能会节省内存，利用字符串的公共前缀可以减少查询字符串的时间，能够最大限度的减少无谓的字符串比较，同时在查询的过程中不需要预知待查询字符串的长度，沿着字典树的边进行匹配，查询效率比较高。

3.5 分布式选举协议

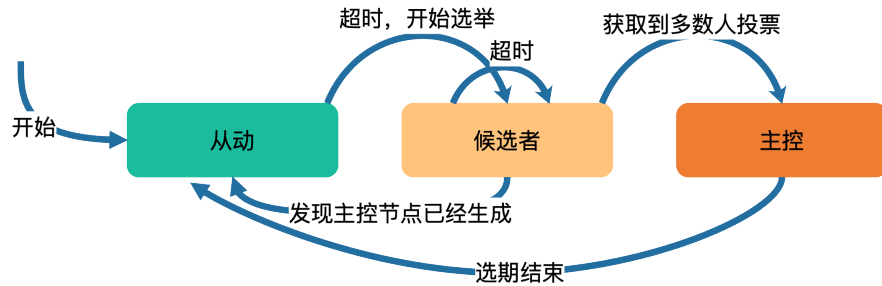


图 19 选举协议

每个节点上线时如果有现有的消息网络会直接连入现有的消息网络，当大量节点同时上线或者子网间关键节点下线时会触发节点选举，选举产生一位核心节点代表他们和其他集群进行消息通信，并在这些核心节点中再次选举产生一个核心节点，这样所有的节点会自动的根据网络环境分层出消息层，并在消息传递之间保持最小路径在分布式选举协议中，一个节点任一时刻处于以下三个状态之一：

- 从动
- 候选者
- 主控

如上图所示，可以看出所有节点启动时都是从动状态；在一段时间内如果没有收到来自主控的心跳，从从动切换到候选者，发起选举；如果收到大多数的同意票（含自己的一票）则切换到主控状态；如果发现其他节点比自己更新，则主动切换到从动。

总之，系统中最多只有一个主控节点，如果在一段时间里发现没有主控，则大家通过选举-投票选出主控。主控会不停的给从动发心跳消息，表明自己的存活状态。如果主控节点故障，那么从动节点会转换成候选状态，重新选出主控。

3.6 硬实时通信

在工业控制领域，实时性（Real-Time）是一个关键性的需求。实时系统是一类计算机系统，其准确性取决于计算逻辑的正确性以及产生结果的时机。这意味着实时系统需要在预定的时间限制内完成特定任务，否则可能导致系统失效。实时性在工业控制领域的重要性表现在以下几个方面：

- 1、 生产过程监控：实时系统能够实时监测生产过程中的各种关键参数（例如温度、压力和流量），以确保生产过程的连续性和稳定性。如果实时系统无法在预定的时间

限制内对异常情况做出响应，可能会导致生产流程中断，甚至发生安全事故。

- 2、设备运行与维护：实时性对于设备的运行和维护具有关键意义。实时系统可以收集和分析设备状态数据，及时发现设备故障，并为操作人员提供故障诊断和处理建议。如果实时系统无法在预定的时间限制内完成这些任务，设备可能会因故障而停机，从而影响生产进度。
- 3、工业自动化与优化：实时性在工业自动化和优化过程中起着至关重要的作用。实时系统可以在短时间内完成大量复杂数学计算任务，为生产过程提供智能决策支持。这有助于提高生产效率、降低成本，并确保产品质量。
- 4、安全与事故防范：在工业领域，实时系统在安全和事故防范方面发挥着重要作用。实时监控和报警功能可以帮助工作人员及时发现潜在的安全隐患，并采取必要的措施防范事故。若实时性要求得不到满足，可能会导致灾难性的后果，如设备损坏、生产中断，甚至人员伤亡。

因此，为满足设备间实时性消息和硬中断响应的需求，需要根据硬件单独编写相应的通信库去支持该性能要求。在 CPU 资源调度时，OS 主要提供一个多任务 (multitasking) 的运行环境，以方便应用的开发。在开发某个应用时首先把工作拆解成多个任务 (Task/Thread)，每个任务都可以简化成一个简单的无限循环：

```
1 void MyTask (void)
2 {
3     while (1) {
4         Wait for an event to occur;
5         Perform task operation;
6     }
7 }
```

代码 3.1 基本调度过程

如上面代码所示，任务 (Task) 都是等待 event，然后处理事务。任何一个任务得以运行，都是因为它收到了一个 Event，这个 Event 可能是一个中断、也可能是超时到期、还有可能是其他任务发出的 IPC 信号，继续追查发出 IPC 信号的任务最后的源头 Event 肯定是一个外部设备硬件中断或者是内部的 Timer 中断。中断引起了 Event 传递，形成了逐个运行多个任务的链条 (Chain)。一个系统内部会存在很多条这种链条。

对实时系统来说，不仅仅要求 OS 能提供多任务环境，更要求任务能在极短的时间之内响应外部的中断事件。

对于终端节点来说硬实时通讯较好实现，利用时钟定时中断即可强制切换到高级别

消息发送，但是终端节点一般作为信息发送和执行节点，不具备逻辑处理功能，在整体通讯延迟受最慢一级也就是逻辑处理层一般也是主机节点影响最大，为处理此问题，对消息进行分级，高优先级消息会先触发中断响应，并切换到执行状态, 且不可被抢占。

3.7 终端通信

在上述协议下可以实现在场景内部的毫秒级硬中断响应通信需要，对于已有的不同终端，根据通信协议和线材不同可连入已有的物理节点进行消息转发，同时目前已开发了 c/rust 通信依赖库可供开发使用，直接在设备内嵌入通信逻辑。用户可以选择是外接设备还是内嵌代码接入到整个数字孪生平台，该物联终端接口可同样应用于自定义仿真算法和逻辑算法节点通信接入。

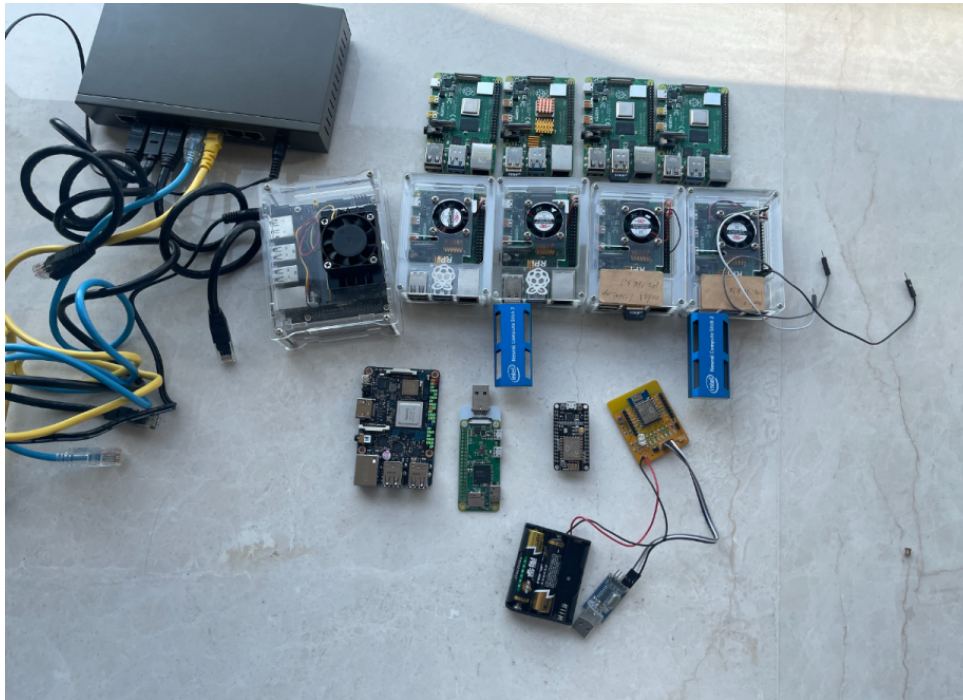


图 20 测试设备图

如图20所示，在 raspberry4/esp32/esp8266/intel jetson nano/thinkerboardS/STM32 等不同芯片平台和不同实时非实时系统上通过了通信库的基础通信功能测试,在 esp32/raspberry/stm32 上通过了硬中断通信功能测试。

3.8 自组网测试

3.9 性能分析

以下性能测试皆使用 2.4 GHz 八核 Intel Core i9 作为测试芯片，并进行相关内核性能调优后得到的测试结果。

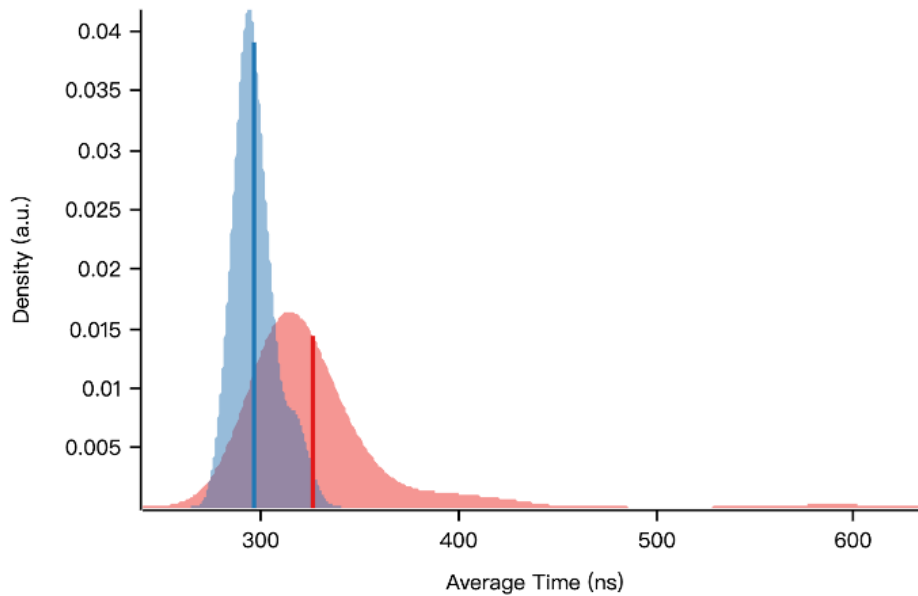


图 21 消息编码速率分布

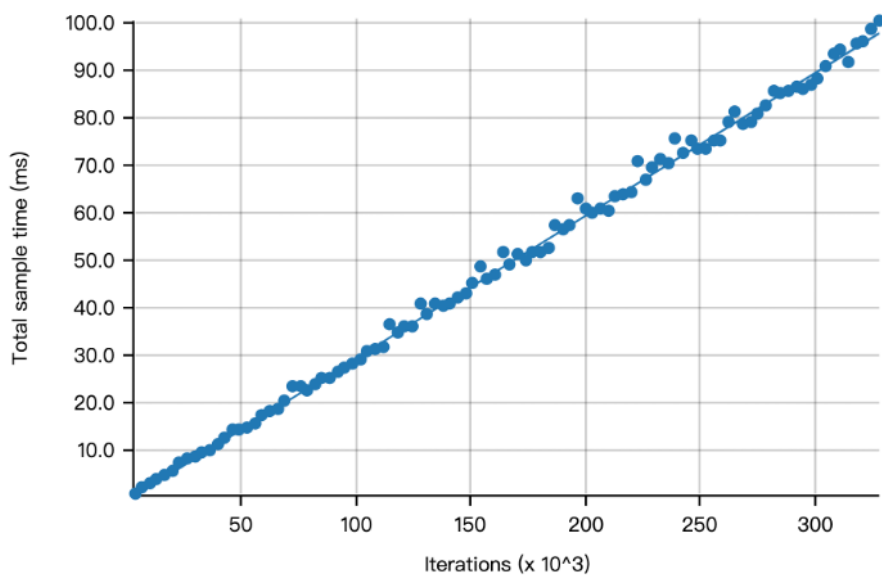


图 22 消息编码速率拟合曲线

由图21和表2可知，消息分发网络中消息编码时间在 300ns 左右，设备负载对消息

表 2 消息编码

编码时间	下限	估值	上限
回归斜率	296.52 ns	298.26 ns	300.04 ns
拟合度	0.9228055	0.9263008	0.9225949
均值	295.37 ns	297.35 ns	299.41ns
样本标准偏差	8.6420 ns	10.325 ns	11.813 ns
中值	293.45 ns	295.80 ns	297.40 ns
绝对中位差	6.1489 ns	8.4741 ns	11.089 ns

编码速率分布有较大影响. 针对消息分发速率, 使用 wrk 工具对系统进行压力测试, 得到图23

```

→ ~ wrk -t8 -c1000 -d15s http://127.0.0.1:4000/api/topic/derive
Running 15s test @ http://127.0.0.1:4000/api/topic/derive
8 threads and 1000 connections
Thread Stats   Avg      Stdev     Max    +/-  Stdev
Latency       6.33ms    1.83ms   59.51ms  98.16%
Req/Sec      19.82k    2.76k   22.24k   97.25%
2366694 requests in 15.01s, 291.16MB read
Socket errors: connect 0, read 3291, write 0, timeout 0
Requests/sec: 157651.06
Transfer/sec:   19.39MB

```

图 23 消息分发速率压测结果

由图23可以看出, 消息分发在未绑定任何响应函数的情况下单机可以达到每秒 157651 条, 在未读取数据库情况下消息分发延迟 6.33ms, 以上数据目前满足数字孪生平台要求。

第四章 时序模型

在数字孪生实践困境中涉及到了诸多现实难题，如多系统融合、数字化率低、应用场景单一等等。本文在考虑现实应用场景需要后，根据另外两项研究基础，建立虚拟世界的时序模型，扩充了数字孪生系统的应用范围和场景，降低系统使用难度和场景构建难度。

在设计时序模型前，还需要设计清楚多复杂系统的状态模型和事件模型，时序模型建立在该两项技术基础上。

4.1 状态模型

在所有的节点集群中，将节点分为三大类：

服务节点：主要为系统服务节点，运行平台核心功能或一部分核心规则

物理节点：一个接入到消息网络的真实物理设备，其响应会在物理世界生效

虚拟节点：主要为规则节点或者虚体逻辑节点，代表一个虚拟物品或者一段规则。

每个节点在接入消息网络时会自动同步输入输出接口，为方便节点运算管理和展示，对节点库内置有限状态机逻辑，可在算法内直接调用和修改。除了在节点算法内自行配置状态机外，还可以在用户端对节点状态转移表进行直接控制，如表3所示根据不同的输入编辑状态转换逻辑

表3 节点状态转移表

	状态 A	状态 B	状态 C
事件 A		...	...
事件 B		...	...
事件 C		状态 C	...
事件 D		...	...

在一个复杂系统中，可能涉及成千上万的异构节点，节点在各个时间点的状态的合集组成了该复杂系统的状态，该状态可以有一个 n 维的向量描述， n 为节点数量，元素为每个节点的状态，复杂系统每秒的状态变换可以描述为一个向量在一组事件下转变为另一个 n 维向量。

4.2 事件模型

在本平台中，一条或多条包含特定逻辑的消息被归纳为事件。事件是整个平台流转的最小价值信息单元，也是消息网络订阅主题的发布内容。每个主题仅发布相同类型的事件，而事件具有特定的数据结构。需要注意的是，事件的数据结构可能与节点的输入输出数据结构不一致，因此在事件到达和离开节点时，需要进行一次结构转换。通过这种方式，特定主体的事件可以在整个消息网络和节点网络中传播，并且实时生效和发挥作用。

事件还作为数据存储的单位，每个事件都会作为一条消息存储在数据库中。后续基于数据的分析实际上是在这些事件中挖掘潜在规律。

其数据格式设计为:

```
1 {  
2   "distinct_id": "2b0a6f51a3cd6775",  
3   "time": 1434556935000,  
4   "type": "track",  
5   "anonymous_id": "2b0a6f51a3cd6775",  
6   "event": "",  
7   "properties": {  
8     "ip" : "180.79.35.65",  
9     "data": {  
10      "demo1": test_data,  
11      "demo2": .....,  
12      .....,  
13    }  
14  }  
15 }
```

4.3 时序模型

时序模型是本平台最核心的高级功能和分析手段之一。在运行中的集群网络中，会收集所有历史事件，这些历史数据为所有数据分析提供基础。整个集群包含三个状态：过去、现在和未来。现在状态是物理集群目前的状态，按照预定的规则和节点逻辑运行。过去和未来状态均为模拟场景，发生在用户端。

用户可在客户端自由调整时间进度条。向前拉动时，虚拟场景会自动重新模拟当时

的事件数据并按照当时的逻辑规则运行。向后拉动时，系统会根据历史事件数据进行拟合，预测未来的集群状态和数据。这相当于在虚拟世界中可以自由调整时间线，平台会根据时间戳自动重播历史事件消息或预测未来数据。

时序模型允许在 OneDT 系统中随时在虚拟世界中将复杂系统恢复到过去、现在以及推演的未来某一刻。推演数据来源于过去的状态数据和事件数据，推演方法包括统计学方法和循环神经网络预测。

4.3.1 统计学预测

表 4 复杂系统状态转移表

	过去 A	过去 B	...	现在	...	未来 A	未来 B
节点 a	a1	a2	...	an	...	an+1	an+2
节点 b	b1	b2	...	bn	...	an+1	an+2
....			...		...	an+1	an+2
节点 n	n1	n2	...	nn	...	an+1	an+2

本文验证时序模型功能时采用的时序分解法去预测未来数据，将一个时间信号视为由四部分趋势影响叠加合成：

长期趋势 (Secular trend,T) 指在较长时期内持续发展变化的一种趋向或状态。

季节变动 (Seasonal Variation,S) 由于季节的变化引起的规则变动

循环波动 (Cyclical Variation,C) 指以若干年为期限，不具严格规则的周期性连续变动

不规则波动 (Irregular Variation,I) 指由于众多偶然因素对时间序列造成的影响

对于不同的信号，其特征不同，预测算法参数也应该有偏重，具体的预测算法参数和模型可以由用户配置，也可使用默认配置。

预测模型可采用：

加法模型 $Y_t = T_t + S_t + C_t + I_t$

乘法模型 $Y_t = T_t * S_t * C_t * I_t$

加乘模型 $Y_t = T_t * S_t * C_t + I_t$

对于四种因素预测，可采用最典型的移动平均法，也可以如第五章所示自行提交预测函数进行预测。针对不同特点的数据源应该采用不同的时序预测方法，其机制已经在本系统内实现，具体的预测方法可以根据不同节点选择或者填写不同的预测方法或函数。

4.3.2 循环神经网络预测

循环神经网络（Recurrent Neural Network, RNN）是一种适用于处理序列数据的神经网络结构。相较于传统的前馈神经网络（Feedforward Neural Network），RNN 可以处理不定长的序列数据，因此在时序预测、自然语言处理、语音识别等领域中具有广泛应用。

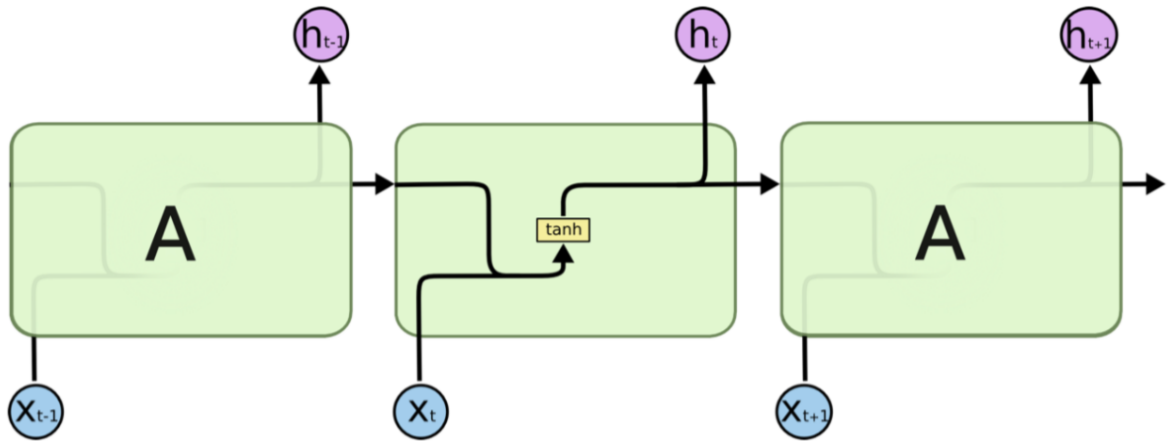


图 24 RNN 精简结构图

在参数预测中，考虑到数据的时间序列特性，采用循环神经网络（RNN）来预测未来的数据。RNN 的结构如图24所示，为使用了单个全连接层的单向 RNN 网络。采用 RNN 神经网络预测的好处是可以对整个系统未来状态进行预测，在使用传统方法进行预测时，只能对单个参数进行预测，而 RNN 可以以某刻所有节点状态作为输入，如表4所示，过去所有节点的状态变换皆可一次性作为 rnn 网络训练数据输入，一次输入即可得到整个系统的状态预测。

第五章 知识模型设计

5.1 基本模型

在建立数字孪生虚拟世界过程中，会有大大小小的规则、算法、逻辑等知识纳入其中，任何规则算法归纳来讲都是描述任意虚拟或实体对象相关作用关系，这些作用关系对于该系统内部世界来说就是一个个具体的知识。本文归纳了这些知识的共同特征，建立了一个一般性的知识模型，该模型是一个动态模型，其规则求解过程可根据特点纳入平台内运算或使用独立进程进行运算求解。如图25所示，基本模型需要先建立三个外部特征，触发域、输入域、输出域。触发域主要用于设定触发求解过程的条件。输入域是指定该模型需要哪些虚拟/实体对象的什么参数。输出域是指定该模型输入影响范围和参数。对于不同的知识或者说规则，根据其作用域和功能特点，将其分为现实类知识、虚拟类知识、仿真类知识

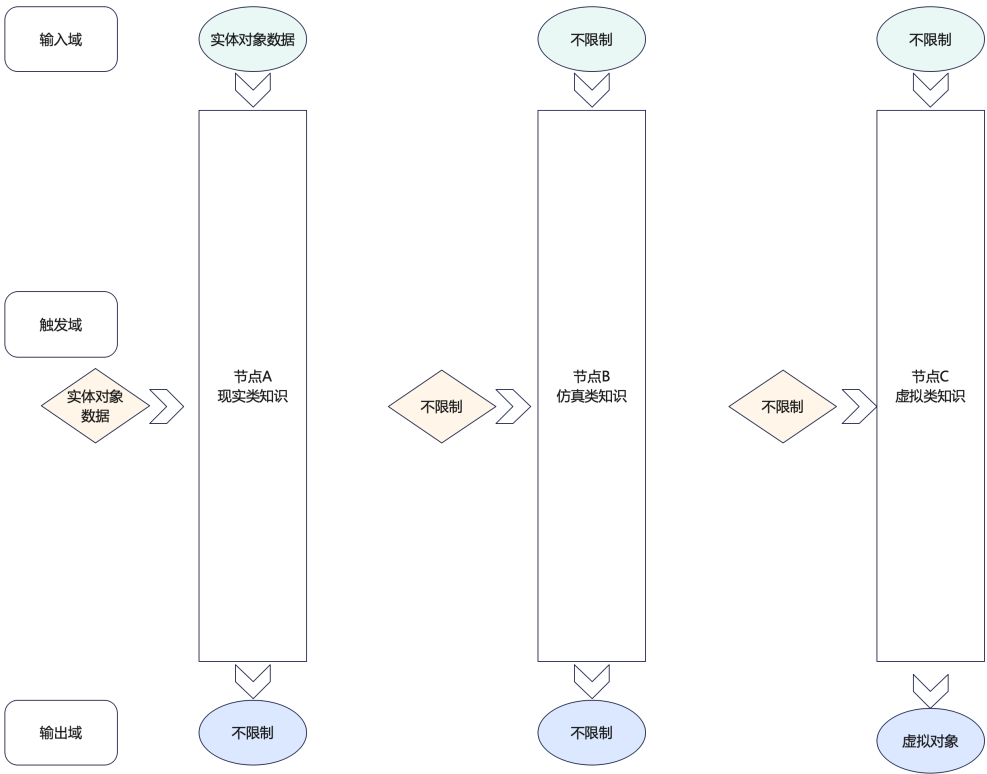


图 25 知识模型分类

现实类知识 该类知识是具体场景下最底层知识，其触发域和输入域仅能为实体对象数据，虚拟场景运行与否不影响该类知识触发和生效作用。

仿真类知识 该类知识为虚拟环境中与实体对象对应的虚拟对象的数据的关联规则，其输出域可以设定于虚拟世界，也可作用于现实世界。

虚拟类知识 该类知识其输出域仅能作用于虚拟世界。

如图26所示，基本模型内部求解过程可以根据类型和功能特点选择提交公式或算法代码，或直接运行独立的求解程序。本章后面以构建碰撞检测和刚体运动力学规则为例讲解代码和公式确定知识模型。

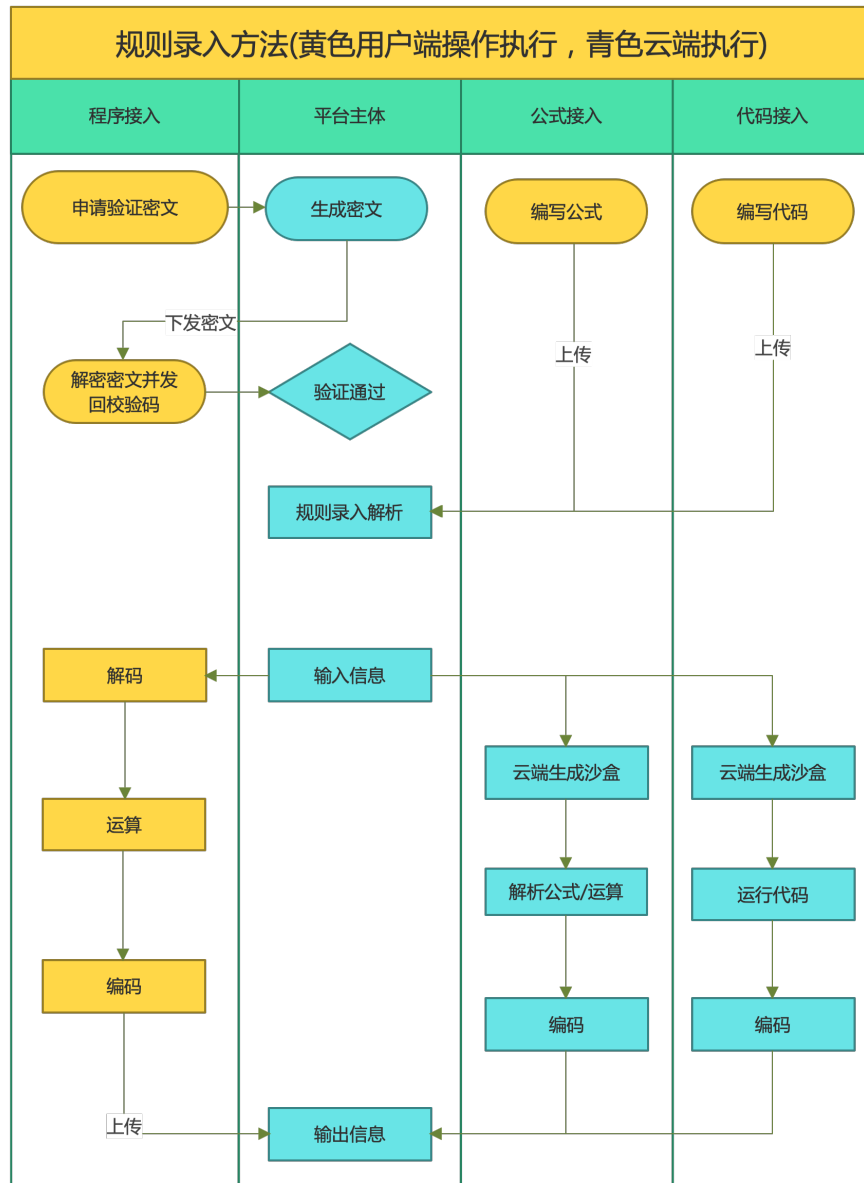


图 26 知识模型上传过程

5.2 程序接入机制

如图26所示，用户仅需要申请访问密钥，调用本平台开发的依赖库即刻纳入到通信网络中作为一个计算节点执行计算任务，与内部的计算节点无任何差别，唯一要考虑的是跨地区网络后消息实时性无法满足要求，不适合去做实时性分析人物。目前支持的依赖库 python/go/c/rust。

5.3 代码解析机制

如图 8 流程所示，代码在用户上传完毕后会根据需要在平台上生成运行环境，并根据需要提供一定的运算资源和数据背景，提供更高阶的数据查询和接入方法。目前代码解析部分采取现成的语法，根据语言特性和场景需要，目前仅支持 js 和 python，尽管采用了一定的沙盒机制去运行，但是仍然面临很大的安全问题，所以现在的代码机制仅供内部使用，后续根据需要开放出自定义代码或者代码检查。

5.4 公式解析算法

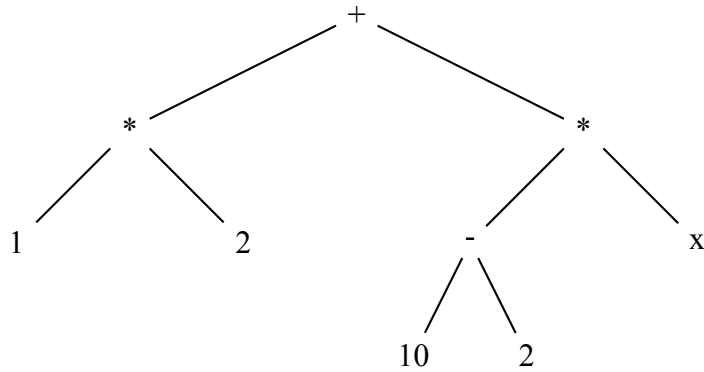
在构建规则模型其中很重要的一步是支持公式输入，对于一条规则，我们可以提供节点数据并转换成合适的格式，比如对于某一理想形状体，我给予 100N 的力 10s，如何通过规则模型解析公式获得该物体 10s 内的运动状态，首先我们需要去解析公式，确定输入变量数量和格式，在确定输出变量数量和格式，根据公式以一定的频率去输出结果。在公式撰写规则上直接采用先用的 latex 语法去语义化编写公式，采用 AST 算法去解析公式结构。以该公式为例：

$$y = 1 * 2 + (10 - 2) * x$$

将右侧表达式转换为 RPN 表达式

$$1\ 2\ *\ 10\ 2\ -\ x\ *\ +$$

根据 RPN 表达式构建出的 AST 语法树为



以 AST 树构建公式算法，输入目前支持浮点数及矩阵格式，支持常用函数表达，整体具体公式需要去做输入输出数据格式转换。

5.5 代码规则案例-碰撞检测

碰撞检测目前定义为虚拟类规则，暂时其输出域数据无法作用于现实世界，主要用于虚拟世界用户交互检测。为了简化物体之间的碰撞检测运算，通常会对物体创建一个规则的几何外形将其包围。在本系统中，碰撞检测中将物体分为三种检测模型，点、AABB、球体。其中，AABB（axis-aligned bounding box）包围盒被称为轴对齐包围盒。

轴对齐包围盒是判断两个物体是否重叠的最快算法，物体被包裹在一个非旋转的（因此轴对齐的）盒中，并检查这些盒在三维坐标空间中的位置，以确定它们是否重叠。

由于性能原因，轴对齐是有一些约束的。两个非旋转的盒子之间是否重叠可以通过逻辑比较进行检查，而旋转的盒子则需要三角运算，这会导致性能下降。如果你有旋转的物体，可以通过修改边框的尺寸，这样盒子仍可以包裹物体，或者选择使用另一种边界几何类型，比如球体（球体旋转，形状不会变）。

在该例中，根据不同需要适合使用代码片段去定义该规则。

5.5.1 点与 AABB

如果检测到一个点是否在 AABB 内部就非常简单了——我们只需要检查这个点的坐标是否在 AABB 内；分别考虑到每种坐标轴。如果假设 P_x , P_y 和 P_z 是点的坐标, $B_{minX} - B_{maxX}$, $B_{minY} - B_{maxY}$, 和 $B_{minZ} B_{maxZ}$ 是 AABB 的每一个坐标轴的范围，我们可以使用以下代码计算两者之间的碰撞是否发生：

```

1 function isPointInsideAABB(point, box) {
2   return (point.x >= box.minX && point.x <= box.maxX) &&
3     (point.y >= box.minY && point.y <= box.maxY) &&
4     (point.z >= box.minZ && point.z <= box.maxZ);

```

```
5 | }
```

5.5.2 AABB 与 AABB

检查一个 AABB 是否和另一个 AABB 相交类似于检测两个点一样。我们只需要基于每一条坐标轴并利用盒子的边缘去检测。

```
1 function intersect (a, b) {
2   return (a.minX <= b.maxX && a.maxX >= b.minX) &&
3         (a.minY <= b.maxY && a.maxY >= b.minY) &&
4         (a.minZ <= b.maxZ && a.maxZ >= b.minZ);
5 }
```

5.6 公式规则案例-刚体运动力学模型

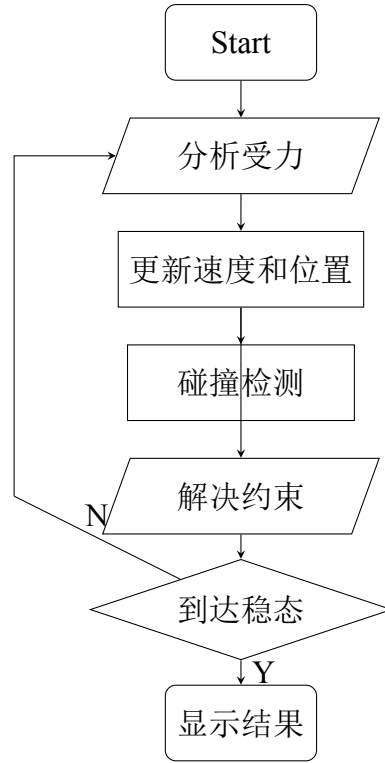
在虚拟世界中为关联现实设备运动状态，需要根据已有的传感器数据，如加速度、里程信息等估计实体对象运动姿态，或者由虚拟对象指导影响实体对象运动，两者相互作用皆需要实现基本的刚体运动。刚体运动的典型计算特征可用于验证公式规则模型，其评估效率和验证性能直观有效。

刚体的运动主要基于牛顿三大定律来模拟：

- 1. 惯性物体在不受力时，总是保持速度不变.
- 2. 力，质量，加速度力在物体上产生加速度，满足 $F = ma$.
- 3. 力的作用是相互的

基于牛顿三大定律，在计算机中来模拟刚体的运动流程如此：

对于每个物体，使用循环的方式来模拟：



最后将所有运动分解为粒子运动和旋转运动，每秒以一定频率分别计算其空间位置和空间姿态，最终合成为完整运动分析过程。

5.6.1 空间位置

先不考虑物体的形状和旋转，把物体当成理想粒子来对待，根据牛顿定律来循环计算物体的速度和位置：

$$dt = t_{i+1} - t_i$$

$$v(t_{i+1}) = v(t_i) + \left(\frac{f(t_i)}{m}\right)dt$$

$$p(t_{i+1}) = p(t_i) + v(t_{i+1})dt$$

将其泰勒展开

$$p(t_{i+1}) = p(t_i) + p'(t_i)dt + p''(t_i)\frac{dt^2}{2!} + p'''(t_i)\frac{dt^3}{3!} + \dots$$

对于浏览器按像素点实时位置渲染精度而言，3 阶泰勒展开已符合精度要求。

因此对于理想点的运动规则模拟可以提交公式

$$p(t_{i+1}) = p(t_i) + p'(t_i)dt + p''(t_i)\frac{dt^2}{2!} + p'''(t_i)\frac{dt^3}{3!}$$

5.6.2 空间姿态

要计算物体的空间姿态，即物体绕某一定点在三自由度上的旋转量，可以用四元数 q 来表示刚体在极短时间内的旋转量，

惯性张量

在三维空间中任取一点 Q 及一个直角坐标系 Q_{xyz} ，可以得到物体的惯性张量：

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yz} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (5.1)$$

对角元素 I_{xx}, I_{yy}, I_{zz} 是物体分别相对于 x, y, z 轴的转动惯量。

$$I_{xx} = \int (r_y^2 + r_z^2) dm$$

$$I_{yy} = \int (r_x^2 + r_z^2) dm$$

$$I_{zz} = \int (r_x^2 + r_y^2) dm$$

计算惯量积

$$I_{xy} = I_{yx} = - \int (r_x r_y) dm$$

$$I_{xz} = I_{zx} = - \int (r_x r_z) dm$$

$$I_{yz} = I_{zy} = - \int (r_z r_y) dm$$

根据惯性张量计算力矩 τ ，角将速度 $d\omega$ ，角速度 ω 。

$$\tau = \frac{dL}{dt} = I \frac{d\omega}{dt}$$

$$d\omega = I^{-1} \tau dt$$

$$\omega(t_{i+1}) = \omega(t_i) + (I^{-1} \tau(t_i)) dt$$

用四元数 q 来表示刚体的旋转状态，刚体在 dt 时间内沿着 ω 旋转了 $|\omega dt|$ 的角度，得到

$$q(t_{i+1}) = q(t_i) * [\cos \frac{|\omega dt|}{2}, \sin \frac{|\omega dt|}{2} \frac{\omega}{|\omega|}] \quad (5.2)$$

对于获取刚体旋转状态的规则特征可以提交为

$$\begin{array}{l} 1 \\ 2 \\ 3 \end{array} \left\{ \begin{array}{l} q(t_{i+1}) = q(t_i) * [\cos\{\frac{|\omega dt|}{2}\}, \sin\{\frac{|\omega dt|}{2}\} \frac{\omega}{|\omega|}] \end{array} \right.$$

第六章 应用案例

6.1 自组织网络-树莓派集群

6.2 时序模型-应力应变分析

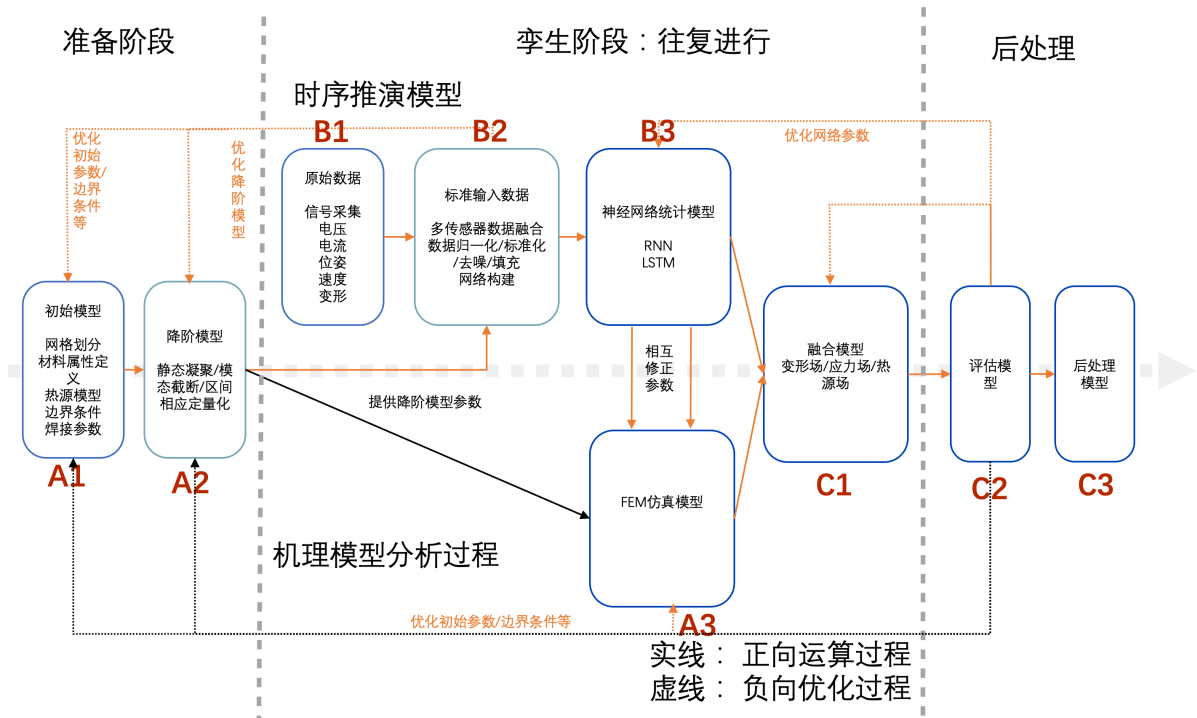


图 27 复合材料 FEM 与时序分析融合

如图27所示，在做某复合板材梁结构焊接工艺的孪生过程时，设计了如图所示的孪生与传统有限元仿真融合分析模型。

6.2.1 有限元仿真

有限元分析（FEA）是一种广泛应用于工程领域的计算机仿真技术，通过将连续的物理系统划分为离散的有限元，以便在较低的计算复杂度下解决复杂问题。在焊接仿真中，有限元分析用于预测和优化焊接过程中的热传递、应力、变形等现象。如图27A1/A2/A3为焊接工艺的有限元仿真过程。

首先，需要使用 CAD 软件创建焊接结构的三维几何模型。这通常包括焊件、焊缝、夹具等部件。

其次，将几何模型划分为较小的有限元，以便在每个元素上进行数值计算。对于焊接仿真，通常需要在热影响区和焊缝附近使用较细的网格，以提高模拟精度。网格类型

然后界定材料属性和边界条件，为模型指定材料属性，如热传导系数、比热容、熔化温度等。此外，还需定义边界条件，如初始温度、对流边界条件等。对于焊接过程，还需要考虑焊接电源的输入参数，如电流、电压和焊接速度等。

在计算完每个单元的应力、变形、温度等信息后，可以将结果可视化，以便更好地理解焊接过程。此外，还可以将计算结果与实验数据进行比较，以评估模拟的准确性。

6.2.2 时序模型预测

B1、B2、B3 为时序推演模型，根据过往采集的 320 个点位形变信息及材料相关信息，进行时序分析。

[illegible]

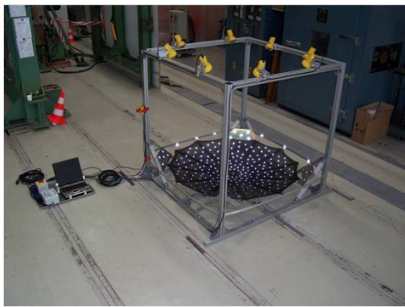


图 29 光学应变采集设备

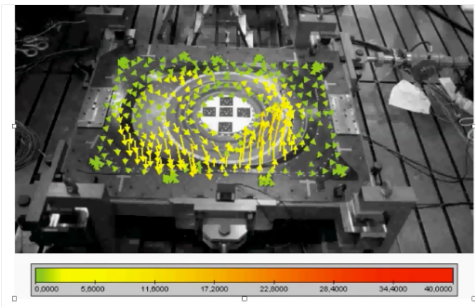


图 30 320 点位应变采集图

时间上进行推演预测，输出全局应变场、应力场等。

6.2.3 融合模型

C1 为融合模型，结合 FEM 模型和时序模型信息，相互校准后进行融合统一输出。如图31所示，在该案例中，时序模型不仅在时间中进行推演，而且在结合一定机理信息后在空间范围内推演，由初始 320 个点位应变信息，生成全局应力场、应变场，并与 A3 的仿真数据进行融合分析，达到机理分析与统计分析融合的效果，利用机理分析的主成作用和 RNN 神经网络对于非线性关系的拟合效果，同时基于平台完成该项分析的实时性计算与展示，有效凸显数字孪生技术在设计制造领域作用。

如表5所示，在未经过融合的情况下，时序模型的精度较低，但是在融合后，时序模型的精度提高了，同时也减少了 FEM 模型的计算量，提高了计算效率。原有网络模型输出频率为 100hz，在加入融合计算后在保证精度 95 基础上输出频率保证 30hz。

表 5 时序分析性能

	A1 初始模型 FEM	A2 降阶模型 FEM	时序分析融合模型
时间	1000s+	100s+	30hz
精度	100	99	95

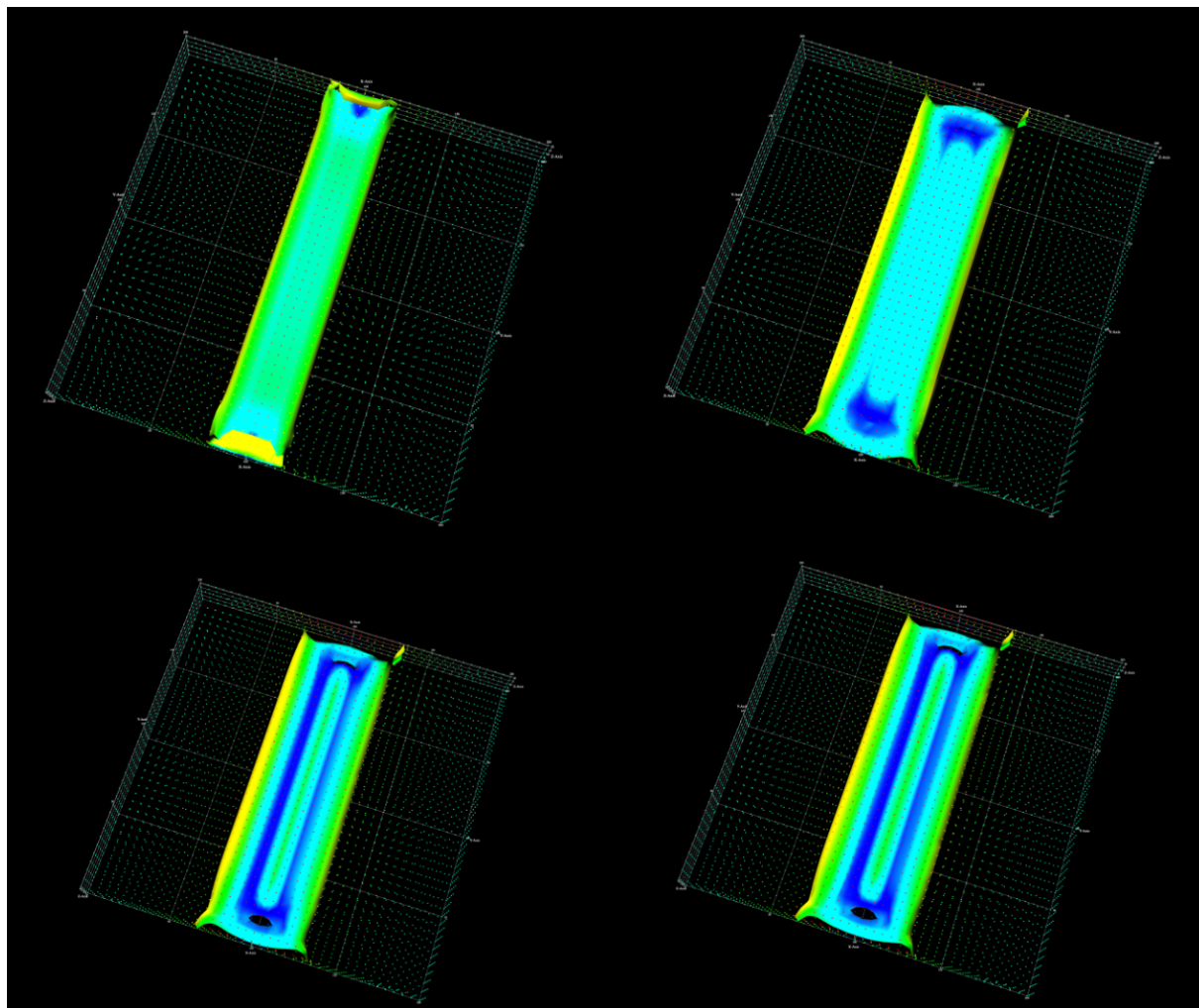


图 31 复合板材焊接过程的孪生模型

结论

参考文献

- [1] The lockheed martin aeronautics win -integration and digital twins the secret to siemens' a&d success[EB/OL]. <http://plm-erpnews.se/aktuellt-pa-engineering-com-the-lockheed-martin-win-integration-is-the-secret-to-siemens-success-in-aerospace-defense-meet-vp-dale-tutt/>.
- [2] aircraft-performance-engineering[EB/OL]. <https://www.plm.automation.siemens.com/global/fr/industries/aerospace-defense/aircrafts-airframes/aircraft-performance-engineering.html>.
- [3] 庄存波, 刘检华, 熊辉, 等. 产品数字孪生体的内涵, 体系结构及其发展趋势[J]. 计算机集成制造系统, 2017, 23(4):16.
- [4] GRIEVES M, VICKERS J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems[M/OL]. Cham: Springer International Publishing, 2017: 85-113. https://doi.org/10.1007/978-3-319-38756-7_4.
- [5] GLAESSGEN E, STARGEL D. The digital twin paradigm for future nasa and u.s. air force vehicles[C]//Aiaa/asme/asce/ahs/asc Structures, Structural Dynamics ; Materials Conference Aiaa/asme/ahs Adaptive Structures Conference Aiaa. [S.l.: s.n.], 2012.
- [6] TUEGEL E J, INGRAFFEA A R, EASON T G, et al. Reengineering aircraft structural life prediction using a digital twin[J]. International Journal of Aerospace Engineering, 2011, 2011(1687-5966).
- [7] GRIEVES M, VICKERS J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems[J]. Springer International Publishing, 2017.
- [8] LEE J. Integration of digital twin and deep learning in cyber-physical systems: Towards smart manufacturing[J]. 2020, 38(8):901-910.
- [9] UHLEMANN H J, LEHMANN C, STEINHILPER R. The digital twin: Realizing the cyber-physical production system for industry 4.0[J]. Procedia CIRP, 2017, 61:335-340.
- [10] ALAM K M, SADDIK A E. C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems[J]. IEEE Access, 2017, 5:2050-2062.
- [11] RAMTILAK A, JOSEPH A, SIVAKUMAR G, et al. Digital twin spark ignition for improved fuel economy and emissions on four stroke engines[C]//SIAT 2005. [S.l.: s.n.], 2005.
- [12] CERRONE A, HOCHHALTER J, HEBER G, et al. On the effects of modeling as-

- manufactured geometry: Toward digital twin[J]. International Journal of Aerospace Engineering, 2014, 2014.
- [13] QI Q, TAO F. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison[J]. IEEE Access, 2018:3585-3593.
- [14] ANWER, NABIL, SCHLEICH, et al. Shaping the digital twin for design and production engineering[J]. CIRP Annals, 2017.
- [15] 陶飞, 张萌, 程江峰, 等. Digital twin workshop:a new paradigm for future workshop[J]. Computer Integrated Manufacturing Systems, 2017, 23(第 1 期).
- [16] BANERJEE A, DALAL R, MITTAL S, et al. Generating digital twin models using knowledge graphs for industrial production lines[C]//9th ACM Web Science Conference, Industrial Knowledge Graphs. [S.l.: s.n.], 2017.
- [17] WAGNER, C, GROTHOFF, et al. The role of the industry 4.0 asset administration shell and the digital twin during the life cycle of a plant[J]. IEEE Int C Emerg, 2017.
- [18] 韩辉. Digital Twin Service towards Smart Manufacturing[J]. 疾病监测, 2019.
- [19] FISHER C R, NYGREN K E, BEAUDOIN A J. Validation of materials-informed digital twin: Mapping residual strains in hsla steel weldment using high energy x-rays[J]. Journal of manufacturing processes, 2022(74-Feb.).
- [20] PAU F, JOAN G, V 柳 CtorGIC, 等. SARS-CoV-2 Spread Forecast Dynamic Model Validation Thorough Digital Twin Approach, Catalonia Case Study[J]. 2021.
- [21] WANG Y C, ZHANG N, LI H, et al. Research on digital twin framework of military large-scale uav based on cloud computing[J]. Journal of Physics: Conference Series, 2021, 1738 (1):012052 (6pp).
- [22] ZHOU H, YANG C, SUN Y. Intelligent ironmaking optimization service on a cloud computing platform by digital twin - sciencedirect[J]. 工程 (英文), 2021.
- [23] HADLEY F, DUNLAP T A, POON T. Precision treatment with machine learning and digital twin technology for optimal metabolic outcomes[Z]. [S.l.: s.n.], 2021.
- [24] SPRUNT B, SHA L, LEHOCZKY J. Scheduling sporadic and aperiodic events in a hard real-time system[J]. 1989.
- [25] STANKOVIC J A. Decentralized decision-making for task reallocation in a hard real-time system[J]. IEEE Computer Society, 1989.
- [26] 娄佩丽, 杜玲. 一种基于 P2PMMOG 的分布式消息分发算法[J]. 沈阳航空工业学院学报, 2007, 24(005):43-45.

- [27] 卢本捷. 分布式消息队列的理论、实现与应用[D]. [出版地不详]: 华中科技大学, 2003.