基于云原生的人工智能训练业务监控系统设计

孙辽东[1] 王 超[1] 陈 培[1] 王德奎[1] 李世刚[1] 张书博[1] 荆荣讯[1] 王文潇[1]
SUN Liaodong WANG Chao CHEN Pei WANG Dekui LI Shigang ZHANG Shubo JING Rongxun WANG Wenxiao

摘 要 近年来，人工智能技术不断发展，已经在各个领域得到了广泛的应用和推广，成为推动社会进步和经济发展的重要引擎。但如何有效地对训练过程进行全程监控，保持系统稳定可靠，提高开发效率和效果，是人工智能技术开发应用过程中的关键问题。通过构建一种针对云原生人工智能业务的监控报警系统，实现全流程的监控，解决当前人工智能开发过程中由于硬件故障、网络通信故障、软件故障导致训练中断的问题，提高系统的灵活性、可靠性和效率。从云原生人工智能业务监控系统的现状和问题出发，提出了包括监控 / 报警管理、数据采集、数据存储、数据分析、报警引擎等关键环节在内的基于人工智能业务负载的全流程监控技术实现方案。实验结果表明，所提出的基于全流程监控的云原生人工智能业务监控系统方案具有较高的实用性和可行性，可以有效地提高算法人员模型训练效率、提升系统可靠性和稳定性。所提出的研究成果为人工智能开发领域的监控问题提供了一种解决方案，具有一定的借鉴意义和推广价值。

关键词 人工智能；训练全流程监控；监控模块；报警模块；云原生

doi：10.3969/j.issn.1672-9528.2024.01.041

0 引言

近年来，随着人工智能技术的不断发展，越来越多的企业和组织开始使用基于云原生技术的人工智能开发平台进行模型的开发。其中，模型训练是人工智能应用开发的重要环节，对训练过程的稳定性和持续性要求非常高。在实际训练过程中会经常由于多种原因导致训练中断，如硬件故障、网络通信故障等。以上这些意外情况如有及时的监控系统支撑，提供有效资源信息给决策系统，则能快速处理异常情况，帮助深度学习任务及时停止和恢复，尤其巨量参数大模型训练场景下，保障任务的健壮性和稳定性是降低训练成本的有效手段。

1 研究背景

当前，国内外已有许多人工智能框架，如 TensorFlow、PyTorch 等。然而大多数人工智能框架的监控功能相对较为简单，难以满足全流程监控的需求。

首先，人工智能模型训练过程是一个复杂的过程，并且训练过程通常非常耗时且具有计算资源密集型特点。如果仅仅监控物理资源的使用情况是不能保障业务的健壮性和稳定性。诸如硬件故障、节点之间的连通性、不同计算节点的计算能力、软件资源可用性等问题，都有可能会导致任务中断和资源浪费。在大规模集群场景下尤为突出。因此，全流程监控的云原生人工智能业务监控系统是十分必要的。

其次，目前绝大多数的人工智能框架和基于云原生的人工智能平台自动化程度还不够高，当出现故障导致训练中断时，用户需要花费大量的时间排查故障原因，然后重新分配资源来重新训练，这些都能导致浪费计算资源和时间并且难以追溯训练过程。如果能够提供自动化识别故障的能力，根据故障类型重新调度分配资源与训练过程快速恢复，将会大大降低训练消费的成本。

2 云原生人工智能训练业务监控系统

本文提出一种云原生人工智能训练全流程监控系统，解决当前人工智能开发训练中存在的缺陷和问题，包括监控指标不足、故障识别不足、故障自动容错能力欠缺等问题，提出全流程监控的技术实现，并已在云原生人工智能开发平台中应用。全流程监控可以实时追踪和记录物理资源的使用情况，包括 CPU、GPU[1]、内存、磁盘、以太网络、InfiniBand 网卡、RoCE 网卡、MIG 实例、任务内存等，为资源使用者提供资源使用情况的查看，便于调整算法参数，提高模型性能和效率。实时追踪和记录硬件资源故障，包括 GPU、InfiniBand、RoCE 等设备的意外失联。实时追踪和记录软件

1. 浪潮电子信息产业股份有限公司 山东济南 250010

资源故障，包括 NFS 客户端、docker 进程、存储客户端、kubelet 进程、数据库等。故障产生时上报业务层完成任务容错，保持任务继续运行。全流程监控支持按照监控指标定义报警策略以及报警产生时的处理策略，比如 GPU 掉卡后触发平台容错功能，实现任务重新调度和继续运行。

本研究提出的全流程监控技术系统，可以有效地提高人工智能模型的训练稳定性、训练效率、系统可靠性等。同时，该系统具有很强的可扩展性和适应性，可以应用于不同领域的人工智能模型训练。

2.1 整体架构设计

云原生人工智能开发训练全流程监控系统实现了全局的监控管理，监控内容主要包括：开发环境资源使用情况、训练任务进度、集群服务器 CPU/ 内存 / 磁盘 / 网络使用情况、平台关键组件状态、GPU 性能情况以及 GPU 错误率。

云原生人工智能开发平台训练全流程监控整体架构上分为监控 / 报警管理、数据采集、数据存储和报警引擎四个模块，如图 1 所示。

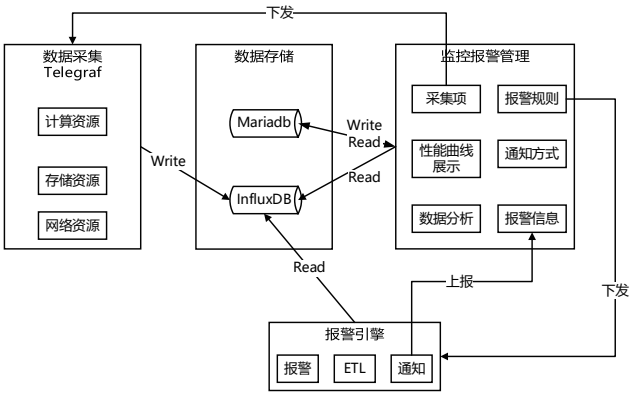

图 1 云原生人工智能开发平台训练全流程监控架构

监控 / 报警管理：基于 SpringBoot[2] + MyBatis + SpringCloud 技术栈实现，主要用于采集项配置、报警规则配置、资源性能数据图形展现和报警信息通知设置等功能，方便用户更加直观地查看资源性能曲线并且可以灵活控制报警策略。

数据采集：采用开源组件 telegraf[3] 完成物理资源和逻辑资源性能数据的采集，并且可以支持通过自定义脚本的方式对采集范围进行动态扩展，比如容器的性能数据、CPU 温度等。

数据存储：由于监控采集到的数据都是时序型数据，所以采用开源组件 InfluxDB[4] 作为监控数据存储介质，InfluxDB 具有处理高负载读写、定期清理垃圾数据的功能，并且 InfluxDB 支持多样的数据分析聚合函数，可用于监控性能数据的快速分析。

数据分析：以集群维度、节点维度、任务维度、租户维度、机时维度分析集群资源的使用情况。

报警引擎：基于业务模块的报警规则以及采集到的性能数据完成报警信息的生成和通知（邮件、站内信、内部业务通知），可以快速准确地完成报警信息的上报。

2.2 架构技术要点

2.2.1 数据采集

在人工智能平台中，数据采集的技术主要包括采集项定义、采集项自动下发、采集状态监测等方面。

首先，需要明确需要采集的对象以及数据来源，并且在监控管理模块完成采集项的配置。采集项信息包括采集项名称、采集频率、采集项描述、采集范围、采集可执行脚本、是否启用等关键属性。采集管理模块通过脚本解析可以支持自定义采集项的添加和配置，来完善数据采集的动态扩展。

其次，采集插件需要运行在每个节点，以完成监控数据的采集和上报，开源 telegraf 初始消耗 1Core 的 CPU 资源和 200 MB 系统内存资源，并且随着采集项增多和采集插件的长时间运行则会消耗更多的物理资源，从而会影响节点中训练任务稳定的执行。通过优化 telegraf 数据采集的线程分配策略，可以支持线程数的自定义控制、支持在系统繁忙时 telegraf 自动挂起并释放资源。telegraf 采集数据写入策略，在系统繁忙时由直接写入 InfluxDB 修改为先写入本地磁盘，然后在系统空闲时再回写 InfluxDB，做到了快速释放内存占用。经过反复测试，60 个采集项在 30 天不间断运行情况下，telegraf 每个节点消耗资源为 0.1Core CPU 和 30 MB。

采集项自动下发是指首先在采集项的基础上，自动把需要采集的数据下发到每个节点上；然后完成采集组件 telegraf 的重启和加载；最后，采集组件重启完成之后采集数据会按照采集频率写入 InfluxDB 数据库，内置的采集指标。整个下发过程参考图 2。

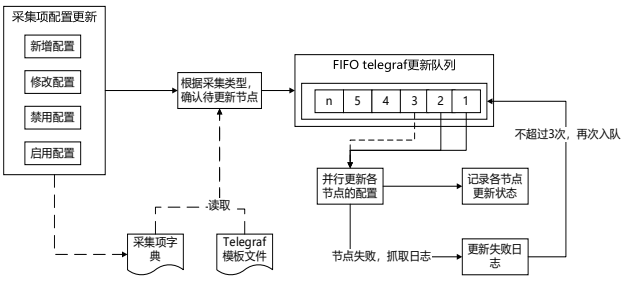

图 2 采集项自动下发流程图

2.2.2 海量数据存储

人工智能平台中在面向大规模计算场景时需要支持大规模节点和大量计算卡的同时纳管。监控模块数据通常全部存储在 InfluxDB 中，由于 InfluxDB 开源方案在规划和性能上无法完全满足场景，虽然企业版本可满足但价格比较昂贵。为了更好地解决该问题，通过分库和数据并发查询的方式，本系统可承担 400 以上节点和 3200 计算卡规模的数据秒级存

储和查询，具体优化策略如下。

集群部署：针对大规模人工智能计算集群中的计算资源扩缩容场景，为防止扩缩容导致节点位置变化，选择使用一致性 hash 算法 [5] 完成集群节点的分组划分，并且使用缓存动态维护分组节点与 InfluxDB 的关系。同时，为了加快管理节点之间缓存数据共享，选择使用 RDMA[6] 技术。

优化数据采集：在 telegraf 中增加 output 解析组件，在数据写入之前首先通过查询“节点与 InfluxDB 的关系”快速获取要写入的 InfluxDB 地址，然后调用 InfluxDB 写入接口完成数据写入。

数据读取：精确读取，根据节点可以定位到具体的 InfluxDB，直接使用 InfluxDB 本身的数据分析函数。全局读取，分库查询并且完成数据的汇总，包括使用并发函数完成分库查询，使用分析函数完成数据聚合；分析函数，使用 java 封装均值函数、峰值函数、方差函数；并发函数，使用代理模式封装并发执行工具类，快速得到基础数据用于后续业务处理。通过以上技术点彻底完成业务逻辑与 InfluxDB 原生接口的解耦。

通过以上优化方案可以实现 400 Byte 数据的秒级存储和查询。

2.2.3 报警引擎

在人工智能平台中，报警引擎是一种实时报警和预警机制，可以在监控数据出现异常时及时发出警报，提示开发人员或运营人员采取相应措施。整个处理过程参考图 3。报警引擎是云原生人工智能开发平台训练全流程监控重要的一环，可以通过实时监控、动态设置报警条件、多种报警等方式，确保云原生人工智能开发平台稳定性和正常运行。为了实现这一目标，采用多层次保护、数据加密等多种设计方案，以确保警报系统的可靠性和安全性。

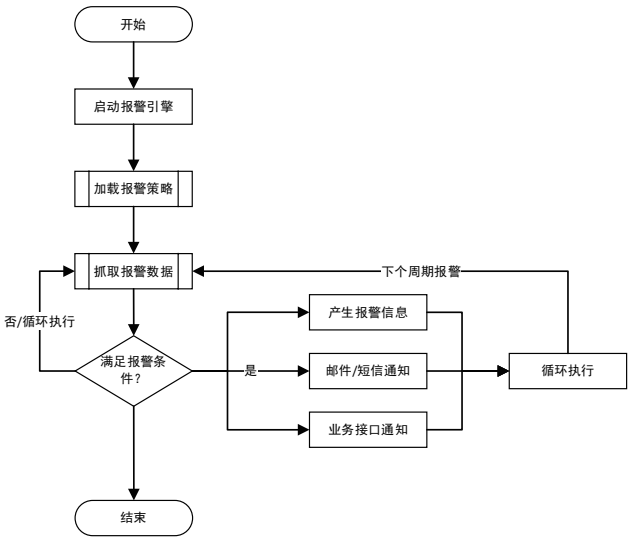

图 3 报警引擎流程图

3 系统性能验证

为验证在本研究中提出监控系统在实际场景下的性能，以下设计并详细阐述了实验设计和结果分析。

3.1 实验环境

表 1 为实验环境信息。

表 1 实验环境信息

服务器	总计 400 台服务器，服务器配置：CPU×2/ 内存 16 GB×8/ GPU V100-32 GB×4/ 磁盘 SSD_1T×2/100 G IB×4
数据集	CIFAR-10
训练框架	Tensorflow
训练环境	Docker+ Kubernetes

3.2 实验设计

首先，通过云原生人工智能开发平台训练过程常见故障进行调查和分析，发现在分布式场景下经常会由于网络通信故障或 GPU 掉卡导致任务失败或者卡住，而且 GPU 运行温度过高也会经常导致 GPU 掉卡，以及网线松动 / 交换机故障导致网络通信故障等。基于此选取“GPU 温度过高（>=80℃）掉卡故障”“InfiniBand 网卡故障”指标来验证云原生人工智能开发平台的稳定性和可靠性。基于选定的故障分别设计故障数据采集脚本，搜集故障产生时平台训练任务的状态。设计场景如表 2 所示。

表 2 实验设计场景

分布式任务名称	GPU 温度过高（≥80℃）掉卡故障	InfiniBand 网卡故障
任务 1	可识别	可识别
任务 2	不可识别	不可识别

其次，把设计的故障脚本通过自动下发组件下发到每个节点中，并且完成 telegraf 的重启和初始化。数据采集模块开始正常运行，并且把采集的数据写入到 InfluxDB 中。数据采集持续 30 min。在这一步，针对每种不同的可能情况进行了多次实验，以获取尽可能准确的结果。

然后，根据故障采集项设计报警策略，并且基于 tensorflow 训练框架的 checkpoint 机制和 Kubernetes 的 pod 的状态机制进行整合。通过 pod 的状态反映训练任务的状态，并对任务的状态进行实时的监控，对于出现异常的任务，通过 checkpoint 进行容错处理。平台的容错机制可以有效保证任务的持续训练，减少因物理设备意外情况造成的时间浪费。

最后，通过 GPU 压力测试工具触发 GPU 高温温度报警、InfiniBand 网卡插拔触发 InfiniBand 网卡故障。在故障模拟过程中实时记录 GPU 的温度、InfiniBand 读写流量、GPU 利用率等数据。

综上所述，在实验设计部分，充分利用具备全流程监控系统的云原生人工智能开发平台，对不同故障采集项进行了细致的实验设计，同时还进行了全流程的监控与收集。

3.3 实验结果分析

在本研究中，使用本文所述的具备全流程监控系统的云原生人工智能开发平台，针对全流程监控的训练进行了一系列实验。本节将对这些实验的结果进行详细分析，其中表 3 展示了优化前后响应时间对比效果，表 4 为优化前后可靠性对比。

表 3 优化前后响应时间对比

单位：ms

并发数接口	优化前				优化后			
	10	20	50	100	10	20	50	100
加速卡总览	738	1815	3535	5446	108	542	526	1411
加速卡色块	3666	5311	12 396	22 149	106	550	740	1751
曲线图	234	587	1690	3959	127	524	598	1751
1H/12H/24H 百分比	137	300	3502	3366	127	213	607	1759
节点监控列表	2067	3074	5372	10 361	106	545	902	1752
节点 GPU 卡信息	256	855	2886	24 089	107	545	900	1615
IO 监控列表	1952	4133	7577	17 288	110	595	499	1440
系统信息	1255	2848	6335	20 679	128	539	509	1285
任务对应用户信息	128	567	514	793	114	539	449	1189
主机性能曲线	1856	1945	4503	13 423	270	268	498	1202
GPU 性能曲线	1879	2243	4182	10 856	334	257	598	995

表 4 优化前后可靠性对比

对比项	优化前	优化后
telegraf 资源消耗	1Core，200 MB 内存	0.1Core，30 MB 内存
任务运行（GPU 温度过高、Infiniband 网卡故障）	运行失败	运行成功，并且训练结果不低于无故障场景
任务运行（无故障）	运行成功	运行成功，准确率不低于优化前
可扩展性	无	动态添加故障指标
业务全流程监控	只能以物理资源为维度监控任务运行	可以以任务、物理资源、集群分组等多维度监控任务的运行

综上所述，实验结果表明，本文提出云原生人工智能业务监控系统可以有效提高全流程监控的效果，并且可以通过自定义故障采集项来完善平台的故障识别能力，可以有效地提高开发效率、提升系统可靠性、接口响应和稳定性。

3.4 实验结论

在本研究的实验验证中，利用所提出的云原生人工智能业务监控系统对全流程监控进行了训练，并获取了相应的实验结果。通过对实验结果的分析，得出以下几点结论。

首先，云原生人工智能业务监控系统能够有效地进行全流程监控的训练，能够准确地捕捉到故障发生的过程，并及时发出预警信息，有助于提高系统的安全性和稳定性。

其次，整套监控系统具有良好的可扩展性和可定制性。在实验过程中，还可以通过自定义策略添加故障采集项和报警策略，实现了更高精度的数据监控，证明了平台具有很好的可扩展性和可定制性。

最后，结合实际应用情形对所开发的云原生人工智能业务监控系统进行了评价。经过实验验证，该平台具有很高的实用性和可靠性，能够满足不同应用场景的需求，为开发人员提供了有效的工具支持。

4 结论

在研究一种云原生人工智能开发平台训练全流程监控的过程中，本文主要通过分析该平台的开发与应用场景，探讨了一些关键技术在应用时的问题，并给出了一些较为理想的解决方案。

首先，该平台可以有效提高人工智能模型的训练效率和精度，为应用场景带来更好的实际效果。其次，平台的架构设计和技术实现也得到了较好的验证，为今后类似平台的开发提供了参考。而在实际应用过程中，也发现了一些问题，比如 GPU 性能的实时监控、数据聚合与分析等方面还需进一步优化。

基于以上结论，对该云原生人工智能开发平台未来的发展趋势做出以下展望。第一，更加智能的监控与优化方案将会应用到平台的实际开发中，以提高训练效率和工作效果。第二，在平台开发过程中，数据的质量、数量和多样化也将会成为重点，为人工智能模型的训练提供更充分的资源和支持。第三，该平台的可拓展性和适用性还需进一步提升，并加强与其他技术领域的结合，以满足未来更为广泛的需求。

综上所述，该云原生人工智能开发平台训练全流程监控的研究具有重要意义，其技术研究与应用将为人工智能领域的发展做出更大的贡献。

参考文献：

[1] 王召选 . 基于 Prometheus 的 GPU 服务器运维监控系统 [J]. 信息与电脑 , 2021,33(9):131-133.

[2] 张嘉豪 , 赵亮 , 翁铭隆 , 等 . 基于 SSM+SpringBoot 技术实现服务器监控的研究 [J]. 科学技术创新 , 2020(33):101-102.

基于 ASEC 的多模态虚假新闻检测的研究

甘甜甜[1] 王 亮[1] 黄世奇[1]
GAN Tiantian WANG Liang HUANG Shiqi

摘 要 当前的新闻检测模型尚存在模态特征融合困难、不够准确等问题，针对这些问题融入用户社交特征，设计了一种 ASEC 融合模型。首先采用 ALBERT 和 Swin transformer 模型分别提取文本和图片特征，加入归一化处理的用户模态特征；然后通过 co-attention 注意力机制和 ECANet 注意力机制组成的 ASCE 将这三种模态进行融合，合理分配这三个特征的权重，经过全连接层实现检测；最后在 MediaEval2015 数据集上检测出虚假性的准确率为 93.92%，精确率为 94.07%，比起单一的机制融合多个模态，ASEC 模型的准确率提升了 3.35%。实验结果表明，使用所提出的检测虚假新闻算法模型，检测的准确率、精确度较高，能避免模态信息的损失，也能够更好地识别虚假新闻。

关键词 多模态虚假新闻检测；ALBERT；Swin transformer；ECANet 注意力机制；co-attention 注意力机制

doi：10.3969/j.issn.1672-9528.2024.01.042

0 引言

近年来，信息技术的飞跃式发展使得社交媒体在人们的社会生活中愈发不可或缺。人们可以通过社交媒体更便捷、更高效地获取各类信息，随时随地都能够通晓天下事。然而新闻报道一味地追求吸引人的眼球，虚假新闻利用人工智能、大数据等相关技术鱼目混珠，导致了很多悲剧，例如夸大造谣胡鑫宇失踪事件博大众眼球、造谣北京协和医院 2022 级硕士复试存在“黑幕”、疫情期间散播涉及疫情的虚假信息引起民众的恐慌等。信息的不确定性使得网民们陷入惶恐、迷茫，损害了媒体公信力和社会秩序，所以对虚假新闻的监控和修正尤为重要。因此，综合分析新闻的图片、文字、用户社交等特征是虚假新闻检测的重中之重。

近些年，多模态虚假新闻检测大致有两个方向，一个是基于虚假信息的传播进行检测，另一个是基于信息模态检测。基于传播信息来检测新闻代表的有 DAVOUDI 等人 [1] 通过对传播树和立场进行动态、静态、结构综合研究分析构建特征来检测虚假新闻。Hu 等人 [2] 提出了含有句子、实体和主题的传播图，加入图注意力网络和实体对比网络，更好地捕捉实体之间的关联。SONG 等人 [3] 提出了 TGAT（时序图注意力）模型，把过往的信息形成传播图的记忆向量，来实时学

1. 沈阳化工大学计算机科学与技术学院 辽宁沈阳 110142

[3] RATTANATAMORONG P,BOONPALIT Y,SUWANJINDA S,et al.Overhead study of telegraf as a real-time monitoring agent[C]//International Joint Conference on Computer Science and Software Engineering.Piscataway:IEEE,2020:42-46.
[4] 张世贤，张少春，谢晓东．基于 InfluxDB 的监控设备通用运维管理平台 [J]．计算机系统应用，2021,30(12):123-127.
[5] 李宁．基于一致性 Hash 算法的分布式缓存数据冗余 [J]．软件导刊，2016(1):47-50.
[6] 陈茂棠，郑圣安，游理通，等．一种基于 RDMA 多播机制的分布式持久性内存文件系统 [J]. 计算机研究与发展，2021, 58(2):384-396.

【作者简介】

孙辽东（1988—），男，山东济宁人，学士，中级工程师，研究方向：人工智能平台。

王超（1988—），男，黑龙江齐齐哈尔人，博士，中级工程师，通信作者，研究方向：深度学习算法、人工智能平台。

陈培（1982—），男，河南郑州人，博士，高级工程师，研究方向：深度学习算法、人工智能平台。

王德奎（1985—），男，山东德州人，硕士，中级工程师，研究方向：人工智能平台。

李世刚（1987—），男，山东德州人，学士，中级工程师，研究方向：人工智能平台。

张书博（1996—），男，辽宁本溪人，学士，中级工程师，研究方向：人工智能平台。

王文潇（1991—），男，山东泰安人，硕士，中级工程师，研究方向：人工智能平台。

（收稿日期：2023-10-13）