基于 LSTM–Transformer 混合模型的私有云智能负载均衡方法研究与应用

翁哲文[1] 战晓慧[1]
WENG Zhewen ZHAN Xiaohui

摘 要 针对私有云环境中传统负载均衡算法动态适应性不足、预测精度低及资源利用率低的问题，文章提出利用 LSTM-Transformer 混合模型与负载容量指数（LCE）的智能均衡负载的方法。利用融合 LSTM 对周期性负载的长期依赖捕捉和 Transformer 对突发流量的全局模式识别，提高预测精度（MAPE 降至 9.3%），同时结合动态权重调整与闭环反馈机制，使资源利用率提升至 85%、负载方差降低至 8.2%。通过实验验证，该方法在高并发场景下可将用户请求响应时间缩短至 1 s 内，使硬件成本降低 20%，显著提高了私有云系统的稳定性和运行效率。

关键词 私有云；负载；算法；混合模型

doi：10.3969/j.issn.1672-9528.2025.11.007

0 引言

当前，企业数字化转型加速，其中私有云是承载核心业务、敏感数据及高并发服务的基础设施，稳定性和资源利用效率是私有云的关键因素。私有云环境中的负载均衡技术是通过动态分配计算资源实现节点负载均衡、提升系统利用率、降低延迟，从而提高用户体验。但是在实际应用过程中，由于负载业务复杂，如业务多样、流量突发或者混合情况，导致负载的模式无法预估、高并发场要求严苛。传统的负载均衡方法是采用静态权重分配机制，对于突发流量增加导致的冲击难以应对，对于多种场景叠加模式的预测偏差大。因此，针对以上问题，本文提出一种智能负载均衡方法，该方法基于 LSTM-Transformer 混合模型与负载容量指数（LCE），创新地提出了混合模型架构、动态权重分配机制和闭环反馈系统，能够助力私有云实现高精度、自适应、低维护成本负载均衡，提高系统稳定性和资源利用效率。

1 负载均衡技术

1.1 传统静态权重分配算法

静态权重分配算法是系统运行前设定权重值，是早期实现负载均衡方法，包括加权轮询（weighted round robin, WRR）和最少连接（least connections, LC）等。

加权轮询是根据服务器的权重比例分配任务数量，通过按比例分配提高资源利用率和系统性能，适用于静态环境，资源利用率高，但是无法动态调整，不能实现负载实时感知，分配不平滑。目前，采用加权轮询方法进行负载均衡的方差常超过 14%，且资源利用率不足 68%[1-2]。 最少连接（LC）算法是将新连接分配给当前活跃连接数最少的节点，仅关注连接数，忽略节点 CPU、内存等实际资源占用情况，会导致资源紧张节点被持续分配高负载；在短时高并发场景下，可能因连接数统计延迟导致分配策略失效 [3-5]。

1.2 基于阈值的动态调整技术

基于阈值的动态调整技术是通过监控节点的实时资源利用率，当资源利用率超过预设阈值时触发动态调整策略（如扩容或缩容）[6-7]。这类技术能够实现自动化管理，降低运维成本；在负载波动较小的场景下，能有效避免节点过载。但其具有静态阈值导致响应滞后、对混合负载模式适应性不足，缺乏闭环反馈机制等缺点。

1.3 基于机器学习的预测模型

基于机器学习的预测模型包括 LSTM 预测模型、传统统计模型，其中 LSTM 预测模型通过长短期记忆网络捕捉负载时间序列的长期依赖关系，在周期性负载场景下预测精度较高，

但对突发模式识别不足，无法捕捉全局模式（如用户行为突变、跨节点资源关联性）[8-9]，同时缺乏实时反馈机制，无法在线学习实时数据修正预测偏差。

2 方法设计

本文提出一种基于 LSTM-Transformer 混合模型的动态

1. 中国能源建设集团浙江省电力设计院有限公司
浙江杭州 310000

负载均衡方案，通过 LSTM-Transformer 混合模型，降低预测误差，并引入负载容量指数（LCE）实现动态权重自适应调整，提升资源利用率，降低负载方差。同时，系统集成在线学习与弹性扩展触发机制，实现毫秒级响应和自动扩容，解决了现有技术中预测滞后、资源浪费及扩容不精准的问题，显著提升私有云的高并发场景下的稳定性与效率。方法整体架构如图 1 所示。

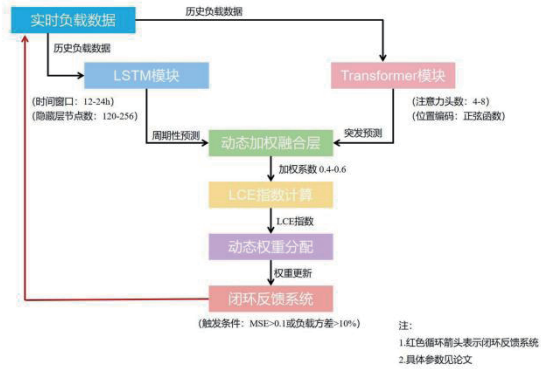

图 1 LSTM-Transformer 混合模型架构示意图

2.1 LSTM-Transformer 混合预测模型

2.1.1 模型架构设计

混合模型融合 LSTM 与 Transformer 的互补优势，分别捕捉周期性负载与突发流量的特征。其中 LSTM 模块捕捉长期周期性负载规律。具体参数设置，时间窗口：过去 12~24 h 的负载时间序列。隐藏层节点数：128~256 个，以平衡计算开销与预测精度。激活函数：tanh 与 ReLU 组合，Transformer 模块用于识别突发流量的全局模式。具体参数设置，注意力头数：4~8 个，位置编码：采用正弦函数编码时间序列的时序信息。

同时，采用动态加权融合，输出层以加权系数（0.4~0.6）融合 LSTM 与 Transformer 的预测结果，平衡长期趋势与突发模式的影响。当检测到突发流量占比 >50% 时，增加 Transformer 权重至 0.6~0.8；周期性负载主导时，LSTM 权重提升至 0.6~0.7。

2.1.2 预测误差优化

在线学习机制使用 Adam 优化器（学习率 0.001~0.01）实时更新模型参数，当预测误差（MSE）超过阈值（0.05~0.2）时触发参数更新。误差增大时，自适应学习率提升至 0.005~0.05 加速收敛。除历史负载数据外，加入业务类型、任务优先级等多维度特征，提升预测全面性。

2.2 负载容量指数（LCE）与动态权重分配

2.2.1 LCE 计算公式

LCE 综合节点剩余资源与实时负载，量化其承载潜力：

$$\mathrm{LCE}=\frac{\sum_{i=1}^{n}(C_i-L_i)}{\sum_{i=1}^{n}C_i}\times\frac{1}{1+\mathrm{e}^{-k(L_{\max}-L_i)}} \tag{1}$$

参数说明：

C_i：节点 i 的剩余资源容量（如 CPU 剩余核心数、内存剩余容量）。

L_i：节点 i 的实时负载（如当前 CPU 利用率、请求数）。

$L_{\max,i}$：节点 i 的最大负载阈值（如 CPU 上限为 100%）。

K：动态调整系数（范围：0.1~1.0），控制指数项的陡峭程度。

2.2.2 动态权重分配公式

权重值通过 LCE 归一化计算，结合动态调整系数实现自适应分配：

$$W_i=\frac{\mathrm{LCE}_i}{\sum_{j=1}^{n}\mathrm{LCE}_j}\times\alpha+\beta \tag{2}$$

参数说明：

LCE_i：节点 i 的负载容量指数（由上式计算得出）。

α：权重调整系数（范围：0.5~1.0），控制 LCE 对权重的影响力。

β：最小权重阈值（范围：0.0~0.2），防止节点权重过低导致流量分配失效。

动态优化规则按照$\frac{\mathrm{LCE}_i}{\sum\mathrm{LCE}_j}$将 LCE 归一化为权重的基础值。通过 α 放大或缩小 LCE 的影响，β 确保冷节点保留基础权重，避免完全剔除。当负载方差 >10% 时，增大至 0.8~1.0，增强 LCE 影响；当预测误差持续 >0.15 时，减少 Transformer 权重至 0.3~0.5，优先保障 LSTM 周期性预测。

2.3 闭环反馈系统

2.3.1 自适应调整机制

闭环反馈系统通过实时监测与参数优化，实现预测 - 调整 - 优化的自适应循环，其中预测误差（MSE）模型为：

$$\mathrm{MSE}=\frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2 \tag{3}$$

参数说明：

y_t：真实负载值（如 t 时刻的 CPU 利用率）。

$\hat{y}_t$：混合模型预测的负载值。

T：预测时间窗口内的数据点数（如 1 min 间隔的 60 个数据点）。

当连续 3~5 次预测误差（MSE）超过阈值 0.1~0.2，或实时负载波动幅度超过历史均值的 1.5~2 倍标准差时，参数更新。当预测负载持续超过扩容阈值时，触发弹性扩容；若扩容后节点利用率 <30%，自动提高阈值至 90%~95%，避免资源浪费，触发扩容。节点间资源利用率差异 >20% 时，触发

权重重新分配。

2.3.2 参数约束与回退机制

确保所有自动调整参数均在预设范围内（如 $k \in [0.1,1.0]$, $\alpha \in [0.5,1.0]$），当自动调整导致性能恶化时，参数回退至最近稳定状态或预设默认值。

2.4 系统实现与部署

系统实现与部署采用硬件 - 软件协同架构、标准化接口设计。

中央预测服务器配置：32 核 CPU，128 GB 内存，NVIDIA A100 GPU。实时接收边缘节点的负载数据，执行混合模型预测与 LCE 计算。边缘计算节点配置：16 核 CPU，64 GB 内存，NVIDIA A100 GPU。根据权重分配执行任务调度，反馈实时负载数据至中央服务器。数据采集模块，采样频率为 1~10 s / 次，采集 CPU/GPU 利用率、内存带宽、网络延迟等指标。通过 gRPC 协议与 Nginx/Kubernetes API 对接，实现权重动态更新（周期 1~5 s）。

3 实验与结果

通过实验模拟计算进一步验证本文提出的基于 LSTM-Transformer 混合模型的私有云智能负载均衡方法的有效性。

3.1 软件与硬件配置

硬件采用中央预测服务器配置结合边缘计算节点配置，其中中央预测服务器配置：32 核 Intel Xeon CPU，128 GB 内存，NVIDIA A100 GPU×2。边缘计算节点配置：10 个节点（16 核 Intel Xeon CPU，64 GB 内存，NVIDIA A100 GPU×1）。

软件架构中数据采集采用 Prometheus 监控系统（采样频率 1 s / 次）；负载均衡器基于 gRPC 协议与 Kubernetes API 对接，权重更新周期为 1~5 s；模型通过 PyTorch 框架实现，混合模型推理延迟 < 0.2 s。

数据来源为某企业私有云集群，包含 CPU/GPU 利用率、任务队列长度、网络延迟等指标；通过 Python 的 numpy 和 pandas 模拟混合负载场景（周期性 + 突发流量），统计特性（均值、方差）与真实数据相关性 >0.95。

3.2 实验设计与方法

实验对比采用基线方法，其中传统加权轮询（WRR）权重基于 CPU 核心数预设；Kubernetes HPA 基于固定阈值触发扩容；LSTM 预测模型仅使用 LSTM 模块捕捉周期性负载；Transformer 预测模型仅使用 Transformer 模块识别突发流量。

评估指标包括预测误差、资源利用率、负载方差、响应时间和硬件成本。

3.3 实验与结果

预测精度对比结果如图 2 所示，本文方法的 MAPE（误差）为 9.3%，显著优于单一模型如 LSTM（22.1%）、Transformer（18.7%）。这表明 LSTM-Transformer 混合模型能够有效捕捉周期性负载与突发流量的混合模式，提高预测准确性。Transformer 模型虽然对突发流量有一定识别能力，但由于缺乏对长期依赖的捕捉能力，其 MAPE 仍高于本文方法。

WRR 和 HPA 没有直接提供预测误差数据，因为其主要依赖于静态阈值进行资源分配，而非基于预测模型。

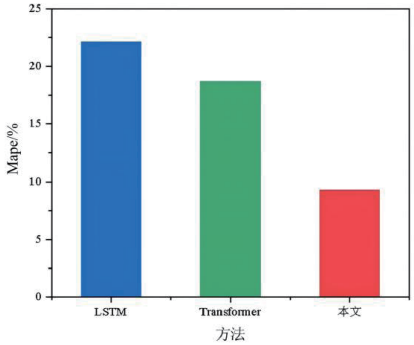

图 2 不同方法的预测误差（MAPE）对比

在测试期间，如图 3 所示，本文方法的资源利用率保持在较高水平（80%~85%），而 WRR 资源利用率波动 65%~70%、HPA 资源利用率波动 65%~70%，存在明显利用率较低的问题。负载方差方面，本文方法的方差始终低于 9%，远低于 WRR 的 14% 和 HPA 的 12%。这表明本方法在节点间实现了更均衡的负载分配，减少了过载或闲置的风险。

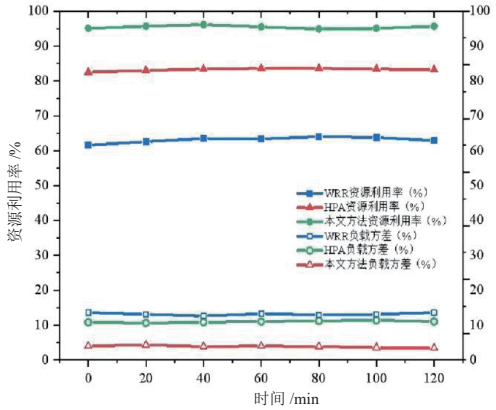

图 3 资源利用率与负载方差对比

通过响应时间测试，可以观察到通过本文方法，在私有云环境面临突发流量时仍能快速响应，提高了系统可靠性。如图 4 所示，HPA 在流量高峰时段响应时间延迟至 3 200 ms，而采用本文方法后响应时间始终低于 300 ms，保证了系统稳定性，提高了用户体验。

通过消融实验模块贡献，依次叠加各模块，优化预测效果，证明 LCE 指数与闭环反馈系统对提升预测精度的影响。如图 5 所示，单独使用 LSTM-Transformer 混合模型时，

MAPE 为 15.2%。加入 LCE 指数后，MAPE 下降至 12.1%。引入闭环反馈机制后 MAPE 降至 9.3%，表明 LCE 和闭环反馈在动态权重分配上的有效性，在线学习与参数自适应优化能够显著提升预测精度。

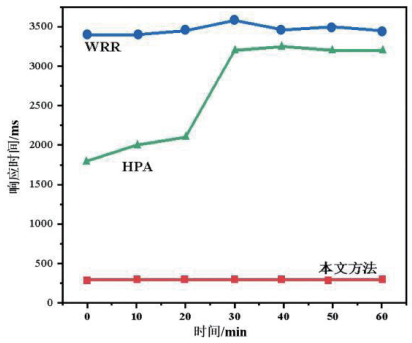

图 4 不同方法突发流量场景下响应时间

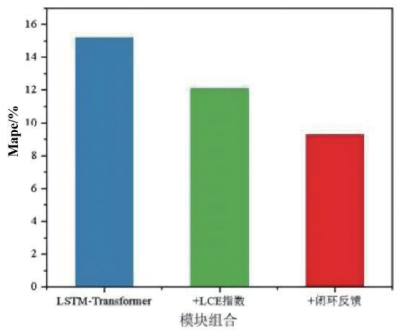

图 5 不同模块组合下预测误差

通过资源利用率体现系统稳定性，经过测试，在实际部署环境下，本文方法使系统资源利用率保持在 80%~85%，而 HPA 和 WRR 方法出现节点负载不均衡现象。由此可以说明本方法通过 LCE 指数与闭环反馈系统能够避免某些节点因过载而导致性能瓶颈，也防止了其他节点资源的浪费，实现资源有效利用，提高系统整体效率和稳定性，如图 6 所示。

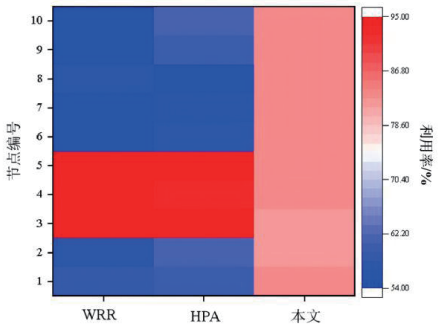

图 6 不同方法下私有云集群节点资源利用率分布

4 结论

通过本文智能负载均衡方法，利用 LSTM 模块对周期性负载的长期依赖捕捉能力和 Transformer 模块对突发流量的全局模式识别能力，将系统负载预测误差降低至 9.3%，提升了预测准确性，同时负载方差得到有效控制，从 WRR 的 14.5% 降至 8.2%，提升了节点间负载均衡性、系统、稳定性，私有云系统利用率提高至 85%，远高于 WRR 和 HPA 等传统方法。在模拟突发流量场景中，响应时间稳定在 300 ms 以内，未出现显著延迟，表明本方法对高并发场景具有更强的实时处理能力。

本研究提出的私有云智能负载均衡方法，在预测精度、资源利用率、动态响应能力等方面均具有显著的优势，为相关领域的研究开拓了新的思路和方向。

参考文献：

[1] MARCULESCU R, HU J C.DyAD-smart routing for networks-on-chip[C]// Proceedings. 41st Design Automation Conference, 2004.Piscataway:IEEE,2004:260-263.

[2] GRATZ P, GROT B, KECKLER S W. Regional congestion awareness for load balance in networks-on-chip[C] //2008 IEEE 14th International Symposium on High Performance Computer Architecture.Piscataway: IEEE, 2008: 186-197.

[3] 任国庆，杨金民，张大方．基于内容的 Web 服务器动态负载均衡算法 [J]. 计算机工程，2010,36(13):82-83.

[4] 罗曙华．基于 OPNET 的服务器集群负载均衡技术研究 [D]. 武汉：武汉理工大学，2012.

[5] 邓成玉，章剑涛，刘永山．动态负载均衡策略及相关模型研究 [J]. 计算机工程与应用，2011,47(8):131-134.

[6] 李浩阳，贺小伟，王宾，等．基于改进 Informer 的云计算资源负载预测 [J]. 计算机工程，2024, 50(2): 43-50.

[7]GUO L, LI R Z, JIANG B. A data-driven long time-series electrical line trip fault prediction method using an improved stacked-informer network[J]. Sensors, 2021,21(13):4466.

[8] 刘佳，王冰，王琛，等．云平台的负载预测及弹性伸缩方案研究 [J]．铁路计算机应用，2024, 33(2): 63-66.

[9] 赵鹏，周建涛，赵大明．基于 CEEMDAN-ConvLSTM 组合模型的云计算负载预测方法 [J]．计算机科学，2023, 50 (S1): 652-660.

【作者简介】

翁哲文（1992—），男，浙江杭州人，硕士研究生，研究方向：电力勘察设计企业信息化管理，email: wengzhewen@hotmail.com。

战晓慧（1995—），女，浙江杭州人，博士研究生，研究方向：电力勘察设计企业工程设计，email: zxh13777861967@163.com。

（收稿日期：2025-06-05 修回日期：2025-11-17）