DOI：10.13245/j.hust.261222

基于纠删码动态资源感知与修复树并行调度的故障恢复方法

陈　李[1,2]　程建华[1]　郭家乐[2]
(1．哈尔滨工程大学智能科学与工程学院，黑龙江 哈尔滨 150001；
2．中国电子科技集团第五十八所，江苏 南京 214062)

摘要　针对纠删码分布式存储故障恢复中节点资源动态变化导致恢复节点选择不准确、异构链路造成传输路径代价与转发负载不均衡，以及节点独占约束限制多跳任务并行执行等问题，提出一种基于动态资源感知与修复树并行调度的故障恢复方法．该方法通过建立六维节点状态演化模型，采用层次分析法与熵权组合赋权，并结合逼近理想解排序法，构建动态资源感知节点选择(DRRNS)方法，实现恢复节点的动态评估与选择；其次，综合节点评分、链路质量、累计传输代价和转发负载构建时间感知修复树，并采用关键路径优先调度估计理论并行工期，提高节点选择的动态适应性和多跳修复任务的并行效率．仿真实验结果表明：在RS(9，6)，RS(12，8)和RS(15，10)三种配置下，与4种端到端组合基线相比，所提方法的平均理论并行工期降低28.06%～60.01%，充分验证了动态节点选择、修复树构建与并行调度的联合优化效果．

关键词　分布式存储；纠删码；故障恢复；恢复节点选择；修复树；并行调度
中图分类号　TN95　　**文献标志码**　A　　**文章编号**　1671-4512(XXXX)XX-0001-09

Fault recovery method based on dynamic resource awareness and repair-tree parallel scheduling for erasure-coded distributed storage

CHEN Li[1,2]　*CHENG Jianhua*[1]　*GUO Jiale*[2]
(1．School of Intelligent Science and Engineering，Harbin Engineering University，Harbin 150001，China；
2．China Electronics Technology Group Corporation No. 58，Nanjing 214062，China)

Abstract　To address inaccurate recovery-node selection caused by dynamically varying node resources，path-cost and forwarding-load imbalance over heterogeneous links，and limited parallel execution of multi-hop tasks under node-exclusive constraints in erasure-coded distributed storage recovery，a failure recovery method based on dynamic resource awareness and parallel repair-tree scheduling is proposed．A six-dimensional node-state evolution model is established，and subjective and objective weights are determined by combining the analytic hierarchy process with the entropy-weight method．The Technique for order preference by similarity to ideal solution is then employed to develop a dynamic resource-aware recovery node selection (DRRNS) method，enabling the dynamic evaluation and selection of recovery nodes．A time-aware repair tree is constructed by considering node scores，link quality，cumulative transmission costs，and forwarding loads．Critical-path-priority scheduling is adopted to estimate the theoretical parallel makespan，thereby improving the adaptability of recovery-node selection to resource variations and the parallel efficiency of multi-hop recovery tasks．Simulation results under RS(9，6)，RS(12，8)，and RS(15，10) configurations show that the proposed method reduces the average theoretical parallel makespan by 28.06%－60.01% compared with four end-to-end combined baselines，demonstrating the effectiveness of jointly optimizing dynamic node selection，repair-tree construction，and parallel scheduling．

收稿日期　2026-08-17．
作者简介　陈　李(1985—)，男，博士研究生，E-mail：chenl@cksic.com．
基金项目　国家自然科学基金资助项目(62073093)．

Key words　distributed storage；erasure codes；failure recovery；recovery node selection；repair tree；parallel scheduling

随着电子对抗侦察、雷达监测、靶场测试等任务的数据采集规模不断扩大，存储系统需持续接收并长期保存海量原始数据[1-2]．多副本机制虽能提高数据可靠性，但将占用大量存储空间，难以满足大容量存储的冗余开销与故障容错需求[3]．纠删码通过生成数据块和校验块，在部分存储节点或数据块失效后，仍能够利用剩余编码块重建丢失数据，以较低冗余获得了较强的数据保护能力．

然而，当存储节点发生故障时，编码块读取、节点传输与解码计算等将会同时占用链路、计算和磁盘I/O资源[4]．随着存储规模和待修复数据量增加，节点负载与链路差异极易造成局部堵塞，导致延长恢复时间并增加数据丢失的风险[5]．为此，相关研究从编码结构、源节点选择和资源协同等角度开展优化．Chen等依据设备异构性、机架层次与数据访问热度，联合调整编码配置与恢复策略[6]；SDNC-Repair感知节点间可用带宽，并联合执行数据源选择、流调度和协作聚合[7]．Paint利用可编程交换机并行聚合修复数据流[8]．上述方法虽从不同层面改善了故障恢复性能，但在恢复期间候选节点的状态持续变化，仅依赖静态快照、单项指标或预设布局难以及时反映节点性能，仍可能使候选节点成为新的资源瓶颈．

除恢复源外，数据传输拓扑与任务执行调度也会影响数据恢复进程．其中，树形再生[9]利用有向拓扑完成数据汇聚；RAN调度全部可用节点，并借助可编程设备实现恢复数据的聚合传输[10]；针对宽条带部署，Liu等通过拓扑局部性和修复贡献度改进数据布局[11]；其他研究则从条带一致性校验、多层次局部校验和双条带分组修复等方面完善纠删码恢复机制[12-13]．现有研究多将节点选择、拓扑构建与任务调度分阶段处理，在父节点决策中较少兼顾节点能力、链路质量、至根节点的累计传输时间和动态转发负载，因此可能产生树深失衡、热点转发及关键路径阻塞[14-15]．

针对上述问题，本文提出一种基于动态资源感知节点选择(Dynamic Resource-Aware Recovery Node Selection，DRRNS)与修复树并行调度的故障恢复方法．该方法将恢复节点筛选视为动态多属性决策问题，通过构建剩余带宽、CPU、内存、磁盘I/O、历史稳定性和历史贡献度的六维节点状态演化模型，并采用层次分析法(Analytic Hierarchy Process, AHP)、熵权法与逼近理想解排序法(Technique for Order Preference by Similarity to Ideal Solution，TOPSIS)，形成DRRNS方法为修复树构建恢复节点集合及统一评分．其次，在父节点选择中联合引入节点评分、链路质量、累计传输时间和转发负载，构建满足树深与扇出约束的时间感知修复树．最后，依据树上“子节点→父节点”的依赖关系执行关键路径优先并行调度．通过仿真和数据一致性校验验证所提方法有效性．

1　基于纠删码动态资源感知的故障恢复方法

本文以系统型Reed－Solomon（RS）码为基础进行数据编码，但RS解码只规定由足够数量的可用编码块恢复数据，并不区分编码块所在节点的实时状态，也不决定恢复数据的传输组织方式．因此，本文设计由DRRNS恢复节点选择、时间感知修复树构建和关键路径优先并行调度组成的一体化方法进行故障恢复．各阶段依次传递节点集合、评分与拓扑信息，使资源评价、路径构建和任务调度在恢复流程中有序衔接．

1.1　方法总体框架

本文考虑分布式存储系统中的单节点故障场景．设系统节点集合为V，故障节点记为v_{f}，持有可用编码块且满足恢复前置条件的节点构成候选集合S．从S中选择d个节点参与恢复，恢复节点集合记为R．为满足RS解码条件，通常令$d=k$．

编码块是否可用决定数据能否恢复，但不能反映节点在恢复期间所承担额外任务的能力．为刻画候选节点之间的资源状态差异，将候选节点v_i在时刻t的运行状态统一表示为六维向量

$$S_i(t)=\left[b_i(t),u_i(t),M_i(t),L_i(t),a_i(t),r_i(t)\right]^{\mathrm{T}},\quad(1)$$

式中指标依次表示剩余带宽、CPU利用率、内存利用率、磁盘I/O利用率、历史稳定性和历史贡献度．其中剩余带宽、历史稳定性和历史贡献度为效益型指标，CPU、内存和磁盘I/O利用率为成本型指标．

故障恢复流程如图1所示．节点选择阶段以候选节点集合与状态矩阵为输入，通过综合评价获得恢复源节点集合及相应评分；修复树构建阶段进一步结合节点链路质量、累计传输代价与转发负载，确定各节点间的父子连接关系；任务调度阶段则依

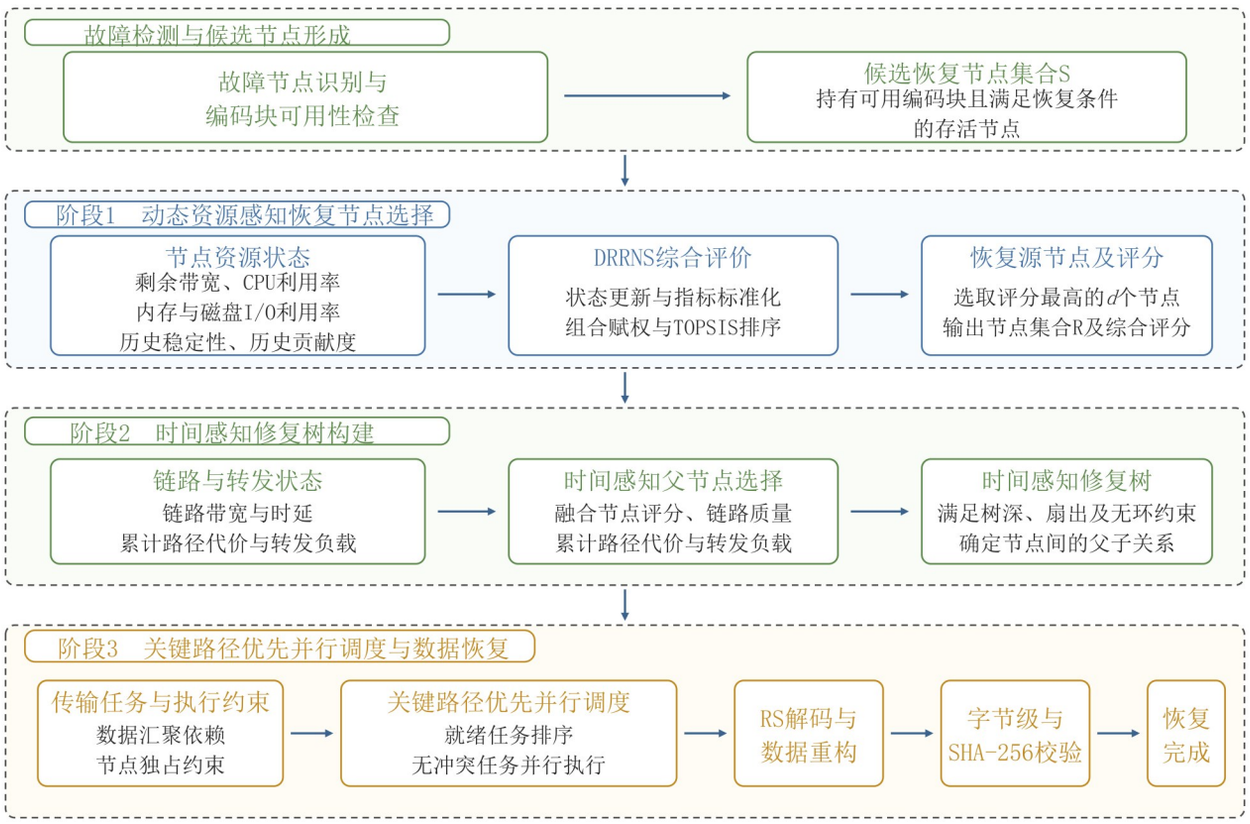


图1 故障恢复流程

据修复树上的数据依赖与节点独占约束，确定各项传输任务的执行次序与并行关系．上述三个阶段依次衔接，共同构成完整的故障恢复流程．

1.2 DRRNS恢复节点选择方法

为了适应候选节点资源状态的动态变化，降低恢复节点选择开销并避免资源瓶颈，本文提出动态资源感知恢复节点选择方法DRRNS，通过描述候选节点的状态变化，并结合AHP—熵权组合赋权与TOPSIS评价完成节点排序，为后续修复树构建提供恢复源节点集合及统一评分．

设候选节点v_i在时刻t承担的恢复数据量为$D_i(t)$，将恢复任务引起的资源消耗和任务完成后的资源释放分别表示为$\Delta S_i^{c}(t)$和$\Delta S_i^{r}(t)$，则节点状态更新过程为

$$\begin{aligned}\Delta S_i^{c}(t) &= D_i(t)\,[-\alpha_1,\alpha_2,\alpha_3,\alpha_4,0,0]^{\mathrm{T}}\\ \Delta S_i^{r}(t) &= -\mathrm{diag}(\beta_1,\beta_2,\beta_3,\beta_4,0,0)\,\Delta S_i^{c}(t)\\ S_i(t+1) &= \Pi_{\Omega}[\,S_i(t)+\Delta S_i^{c}(t)+\Delta S_i^{r}(t)\,]\end{aligned}\tag{2}$$

其中，α表示单位恢复数据量引起的资源消耗；β表示相应资源在一个决策周期内的释放比例；Π_{Ω}表示可行域投影．投影后令剩余带宽不小于0，其余状态指标限制在[0,1]内．历史稳定性和历史贡献度在单个决策周期内保持不变．

状态更新采用线性近似描述恢复任务对节点资源的影响，适用于资源未严重饱和的场景．按照式(1)的指标顺序，将m个候选节点的状态向量组成动态评价矩阵，并对效益型与成本型指标分别进行极差标准化

$$\begin{aligned}X(t) &= \left[x_{ij}(t)\right]_{m\times q},\ q=6\\ X(t) &= \left[S_1(t),S_2(t),\cdots,S_m(t)\right]^{\mathrm{T}}\end{aligned}\tag{3}$$

$$z_{ij}(t)=\begin{cases}\dfrac{x_{ij}(t)-\min\limits_i x_{ij}(t)}{\max\limits_i x_{ij}(t)-\min\limits_i x_{ij}(t)+\varepsilon}, & j\in J_{\mathrm{b}}\\[2ex] \dfrac{\max\limits_i x_{ij}(t)-x_{ij}(t)}{\max\limits_i x_{ij}(t)-\min\limits_i x_{ij}(t)+\varepsilon}, & j\in J_{\mathrm{c}}\end{cases}\tag{4}$$

式中，效益型指标$_{\mathrm{b}}$={1，5，6}，成本型指标$_{\mathrm{b}}$={2，3，4}，编号与式（1）中六项指标的排列顺序一致；ε为防止分母为零的极小正数．标准化后，各指标均为数值越大节点状态越优．

为了兼顾恢复任务的资源优先关系与候选节点间的状态差异，本文采用AHP确定主观权重，并利用熵权法计算客观权重，形成动态组合权重．根据式(1)的指标顺序构造AHP判断矩阵，取矩阵最大特征值对应的归一化特征向量作为主观权重w^{AHP}，并进行一致性检验．由标准化矩阵计算各指标的信息熵及客观权重

$$p_{ij}(t)=z_{ij}(t)\Big/\sum_{r=1}^{m}z_{rj}(t);\tag{5}$$

$$e_j(t)=-\frac{1}{\ln m}\sum_{i=1}^{m}p_{ij}(t)\ln\left(p_{ij}(t)+\varepsilon\right);\tag{6}$$

$$w_j^{\mathrm{E}}(t)=\frac{1-e_j(t)}{\sum_{l=1}^{q}\left(1-e_l(t)\right)}.\tag{7}$$

当某项指标标准化后全为0时，令该指标对应的列p_{ij}=1/m；若全部指标的差异系数均为0,则将客观权重设为等权来保持计算稳定．进一步引入组合系数θ，获得动态组合权重

$$w_j(t)=\theta w_j^{\mathrm{AHP}}+(1-\theta)w_j^{\mathrm{E}}(t),0\leqslant\theta\leqslant1\tag{8}$$

其中，θ取值由参数敏感性实验确定．

在组合权重基础上构造加权标准化矩阵，并确定各指标的正、负理想状态

$$\begin{aligned}&A_j^{+}(t)=\max_i y_{ij}(t),A_j^{-}(t)=\min_i y_{ij}(t)\\&y_{ij}(t)=w_j(t)z_{ij}(t)\end{aligned}\tag{9}$$

候选节点v_i到正、负理想状态的欧氏距离及其综合评分定义为

$$\begin{aligned}D_i^{+}(t)&=\sqrt{\sum_{j=1}^{q}\left(y_{ij}(t)-A_j^{+}(t)\right)^2}\\D_i^{-}(t)&=\sqrt{\sum_{j=1}^{q}\left(y_{ij}(t)-A_j^{-}(t)\right)^2}\end{aligned}\tag{10}$$

$$\sigma_i(t)=\frac{D_i^{-}(t)}{D_i^{+}(t)+D_i^{-}(t)+\varepsilon}\tag{11}$$

式中，$\sigma_i(t)$越大，表示节点越接近正理想状态，其综合资源条件越适合参与恢复．

DRRNS依据式(2)更新候选节点状态，经指标标准化、组合赋权和TOPSIS评价后，按照综合评分由高到低选取前d个节点构成恢复源节点集合．链路质量仅用于后续修复树的父节点选择，不参与本阶段的节点评价．

1.3　时间感知修复树构建方法

DRRNS确定恢复节点后，还需进一步组织节点间的数据传输关系．若所有恢复节点均直接向目标节点发送数据，容易造成根节点链路拥塞；若仅依据节点评分或单条链路质量构建多跳拓扑，又难以反映至根节点的累计传输时间和中间节点转发负载．因此，本文提出时间感知修复树的构建方法，其结构示意图如图2所示．时间感知修复树综合考虑了候选父节点的资源评分、链路质量、累计路径传输代价和当前转发负载，在树深、子节点数及无环约束下确定父子关系，为后续并行调度提供满足数据汇聚关系的传输拓扑．

首先，对所有非自环链路的带宽与时延进行全局极差归一化．节点v_i与v_j之间的链路质量定义为

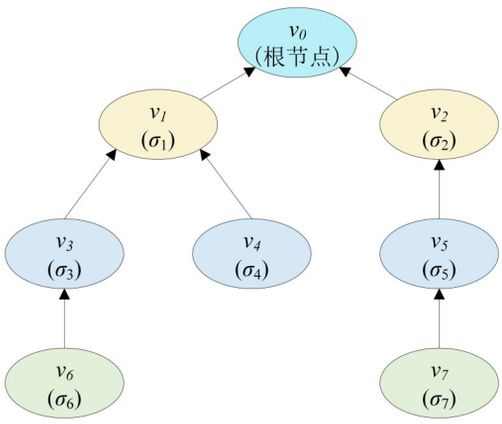


图2　修复树结构示意图

$$Q_{ij}=\eta\overline{B_{ij}}+(1-\eta)\left(1-\overline{\delta_{ij}}\right),0\leqslant\eta\leqslant1\tag{12}$$

式中，$\overline{B_{ij}}$和$\overline{\delta_{ij}}$分别表示归一化后的链路带宽和时延，η为二者的权衡系数．Q_{ij}越大，表示该链路越适合承担恢复数据传输，且链路质量仅用于修复构建，不参与DRRNS阶段的节点资源评价．

设节点v_i需要传输的恢复数据量为D_i，单位为MB；链路带宽B_{ij}单位为Mbps，链路时延δ_{ij}单位为s．若候选节点v_j已在修复树中，则v_i经v_j到达根节点的估计路径代价为

$$\begin{aligned}T_{ij}^{\mathrm{est}}&=\frac{D_i\times8}{B_{ij}}+\delta_{ij}\\K_{ij}&=T_{ij}^{\mathrm{est}}+K_j\end{aligned}\tag{13}$$

式中，T_{ij}^{est}表示恢复数据的理论传输时间；K_j表示节点v_j到根节点v_0的累计路径代价．

设V_T为当前已加入修复树的节点集合．对尚未加入节点v_i的候选父节点集合定义为

$$P_i=\left\{v_j\in V_T\middle|h_j<D_{\max},c_j<C_{\max},v_j\notin\mathcal{D}_i\right\},\tag{14}$$

式中，h_j和c_j分别表示候选父节点的深度和当前直接子节点数，$\mathcal{D}_i$表示节点v_i的后代节点集合；$D_{\max}$和$C_{\max}$分别表示允许的最大树深和单节点最大子节点数．对于尚未加入修复树的节点v_i，其候选父节点只能从当前已入树节点中选取．新增边由v_i指向候选父节点，因此不会在当前修复树中形成环路．

为消除累计路径代价和子节点数量之间的量纲差异，分别在候选父节点集合P_i内进行极差归一化

$$\begin{aligned}\overline{T_{ij}}&=\frac{K_{ij}-K_i^{\min}}{K_i^{\max}-K_i^{\min}+\varepsilon}\\\ell_{ij}&=\frac{c_j-c_i^{\min}}{c_i^{\max}-c_i^{\min}+\varepsilon}\end{aligned}\tag{15}$$

式中，$\overline{T_{ij}}$为归一化后的累计路径传输代价；ℓ_{ij}为节点v_i选择候选父节点v_j时的归一化转发负载；$K_i^{\min}$和$K_i^{\max}$分别为候选父节点累计路径代价的最小值与最大值；$c_i^{\min}$和$c_i^{\max}$分别为候选父节点子节点数的

最小值与最大值．

综上述因素，定义节点v_i选择候选父节点v_j的适配度为

$$\begin{aligned} F_{ij} &= w_1\sigma_j + w_2 Q_{ij} - w_3\overline{T_{ij}} - w_4\ell_{ij} \\ \pi(v_i) &= \operatorname*{argmax}_{v_j \in P_i} F_{ij} \end{aligned} \tag{16}$$

式中，$\pi(v_i)$表示节点v_i的父节点；σ_j表示候选父节点的综合资源能力；w_1、w_2、w_3与w_4为调节参数，总和为1．根节点v_0不属于DRRNS候选节点，其评分项置为0，而非令F_{i0}=Q_{i0}．因此根节点的选择值为

$$F_{i0} = w_2 Q_{i0} - w_3\overline{T_{i0}} - w_4\ell_{i0} \tag{17}$$

上述处理适用于根节点接收与解码能力不构成主要瓶颈的场景．由于根节点是固定恢复目标而非候选数据提供者，其资源状态不参与DRRNS评分，但仍通过链路质量、累计路径传输代价和当前接收负载参与父节点评价．若根节点处理能力明显受限，则应将其资源状态纳入评价，或增加接收带宽和并发处理能力约束．

当多个候选父节点的适配度相同时，优先选择深度较小的节点；若深度仍相同，则选择局部编号较小的节点，保证算法输出的确定性．修复树满足

$$h(T) \leqslant D_{\max}, c_j \leqslant C_{\max}, d \leqslant \sum_{h=1}^{D_{\max}} {C_{\max}}^h \tag{18}$$

式中，$h(T)$表示修复树深度，最后一项为修复树的容量可行性条件．当恢复源节点数超过给定树深与扇出约束所能容纳的节点数量，或某个待加入节点不存在可行父节点时，当前参数组合将被判定为不可行，算法终止并要求重新设置约束参数，不通过放宽$D_{\max}$或$C_{\max}$继续构树．因此，算法输出始终满足式(18)．

修复树构建完成后，恢复数据按照“子节点→父节点”的方向汇聚至根节点．设C_i为节点v_i的直接子节点集合，则以v_i为根的子树所承载的数据量及边(v_i,v_j)的评估传输时间分别为

$$D_i^{\text{sub}} = D_i + \sum_{v_c \in C_i} D_c^{\text{sub}} \tag{19}$$

$$\hat{T}_{ij} = \frac{D_i^{\text{sub}} \times 8}{B_{ij}} + \delta_{ij} \tag{20}$$

式(13)用于构树期间的候选父节点决策，仅使用节点本地数据量；式(20)则在树结构确定后使用子树累计数据量，用于评价最终拓扑的传输开销．

1.4 关键路径优先并行调度方法

修复树确定了恢复数据的传输路径，但并未直接给出各链路任务的执行次序．由于子节点数据需逐级汇聚，且树上任务之间存在前驱依赖；同时共享端点的传输任务不能并发执行．顺序调度难以利用链路并行性，按深度同步调度又可能因全局层间屏障产生等待．因此，本文设计关键路径优先并行调度方法，在满足数据依赖和节点独占约束的前提下，优先执行对理论恢复工期影响较大的传输任务，提高树上链路并行利用程度，并为恢复工期评估提供统一依据．

修复树构建完成后，将每条有向边(v_i,v_j)∈E_T视为传输任务τ_{ij}，其方向仍为子节点指向父节点．任务持续时间采用式(20)的子树累计数据传输时间．令任务开始和结束时刻分别为s_{ij}和e_{ij}，则

$$\tau_{ij} = (v_i, v_j), p_{ij} = \widehat{T_{ij}}, e_{ij} = s_{ij} + p_{ij} \tag{21}$$

式中，p_{ij}表示任务的持续时间．

任务之间的前驱关系由修复树的数据汇聚结构确定，仅当节点v_i的全部直接子节点数据均已传输完成后，任务τ_{ij}才可执行．当前驱集合为空或其中任务均已完成时，该任务进入就绪状态．为避免共享端点造成的资源冲突，调度采用节点独占约束，即同一时刻一个节点至多参与一条传输边．为优先释放对总体工期影响较大的传输链路，定义当前任务沿修复树至根节点方向的剩余关键路径长度为

$$\kappa_{ij} = p_{ij} + \sum_{(v_u, v_p) \in P_{j0}} p_{up} \tag{22}$$

式中，$\mathcal{P}_{j0}$表示父节点v_j到根节点v_0的有向路径；p_{up}表示路径中后继传输任务的持续时间．当v_j=v_0时，κ_{ij}=p_{ij}，且κ_{ij}越大表示任务延迟对最终完成时刻的潜在影响越显著．

在每个离散事件时刻，首先选取前驱均已完成的就绪任务，按κ_{ij}降序排列．若优先级相同，则依次按边持续时间降序，子节点和父节点局部编号升序确定顺序．随后从排序后的就绪集合中依次选取节点互不冲突的任务并行启动，将系统时钟推进至最近的任务完成时刻，释放相关节点并更新后继任务状态，直至全部边传输完成．将全部传输任务的最晚结束时刻定义为调度工期T_{sched}

$$T_{\text{sched}} = \max_{(v_i, v_j) \in E_T} e_{ij} \tag{23}$$

设W_v为节点v所关联传输边的总工作量．在前驱约束和节点独占约束下，节点工作量及调度工期下界定义为

$$W_v = \sum_{(v_i, v_j) \in E_T;\, v \in \{v_i, v_j\}} p_{ij} \tag{24}$$

$$T_{\mathrm{LB}}=\max\left\{\max_{(v_i,v_j)\in E_T}\kappa_{ij},\ \max_{v\in V_T}W_v\right\} \tag{25}$$

$$g=\frac{T_{\mathrm{sched}}}{T_{\mathrm{LB}}}-1 \tag{26}$$

式中，T_{LB}的第一项为树依赖关系决定的关键路径下界，第二项为任一节点需要串行处理的关联边工作量下界．相对间隙g越小，表明当前调度结果越接近所构造的理论下界．此外，任务持续时间由链路带宽、时延及子树累计数据量估算，因此T_{sched}属于给定传输模型下的理论调度工期，不等同于实际分布式存储系统中的实测恢复时间．

2　实验参数设置

实验由理论网络仿真和并行调度仿真两部分组成．理论网络仿真用于评价DRRNS节点选择和时间感知修复树构建的拓扑与传输代价；并行调度仿真采用第1.4节的离散事件模型，在前驱关系和节点独占约束下计算恢复任务的理论完成时间；两部分采用相互独立的计时口径，理论网络时间不与单机RS计算时间直接相加．

节点选择、修复树及端到端比较在独立测试种子5000～5099上分别重复100轮；参数确定使用种子3000～3029，每组参数重复30轮．并行调度实验对每种RS配置、修复树方法和调度策略执行100轮．实验采用RS(9,6)、RS(12,8)、RS(15,10)三种编码配置，均考虑单节点故障，恢复源节点数取d=k. 初始剩余带宽b为400～1000Mbps，CPU、内存和磁盘I/O利用率分别为0.15～0.75、0.20～0.70和0.10～0.60，历史稳定性a和历史贡献度r分别为0.85～1.00和0～1.00. 单节点恢复数据量为50～200MB，节点间链路带宽为100～1000Mbps，链路时延为0.001～0.050s. 按照剩余带宽、CPU、内存和磁盘I/O的顺序，资源消耗系数α设为(0.5, 0.0005, 0.0004, 0.0006)，资源恢复比例β设为(0.3, 0.2,0.2,0.15). 对同一种子生成的节点和链路状态进行配对比较．均值差的95%置信区间由2000次配对Bootstrap重采样获得；显著性检验采用双侧精确符号检验，并对全部比较的p值实施全局Holm校正，显著性水平取0.05．

3　实验结果与分析

3.1　参数敏感性与参数确定

参数搜索采用两轮坐标细化，每轮依次搜索主客观权重系数θ、链路质量系数η、父节点选择权重(w_1、w_2、w_3与w_4)以及深度和扇出约束($D_{\max}$，$C_{\max}$)．θ和η均在[0, 1]内以0.1为步长取值，共形成121组组合；四项父节点权重采用步长为0.1且各分量均为正数的单纯形网格，共84组；$D_{\max}$和$C_{\max}$在1～10内搜索，并剔除容量不足的组合后保留82组．两轮共评价574组候选参数，每组在三种RS配置和30个验证种子上计算．

验证目标综合考虑并行工期、带宽开销、负载离散度、树深、平均链路质量和恢复节点综合评分，其权重依次为0.40、0.15、0.15、0.10、0.10和0.10．为避免待选θ同时参与节点质量的自评价，参数搜索阶段的节点质量统一按参考值θ_{ref}=0.5重新计算．图3为第二轮θ-η网格的验证目标，其中矩形框对应当轮最小目标区域．

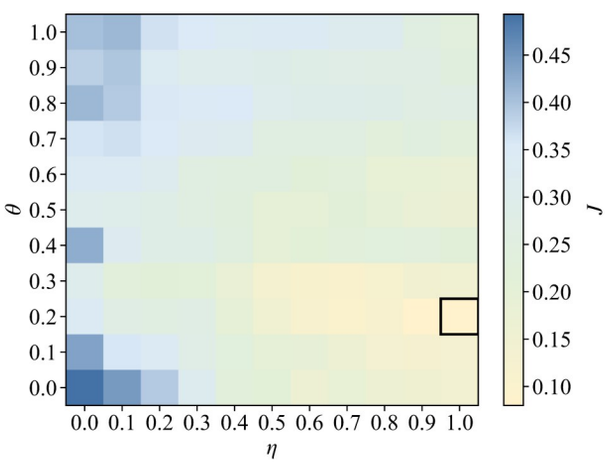


图3　主客观权重与链路质量系数敏感性

经上述搜索，主客观权重组合系数θ取0.2，链路质量系数η取1.0；父节点选择中节点评分、链路质量、累计路径传输代价和转发负载的权重分别为0.1、0.1、0.7和0.1；最大树深$D_{\max}$和单节点最大子节点数$C_{\max}$均取4. 上述参数为当前搜索范围和验证集上的选定结果，不作全局最优性解释．

η=1.0表明当前验证集上的链路质量项Q_{ij}仅保留归一化带宽分量，但链路时延并未从父节点选择中删除．路径传输代价$\overline{T_{ij}}$仍同时包含传输时延与传播时延，且对应权重w_3=0.7，因此时间感知约束仍是最终模型的主要组成部分．

3.2　恢复方法对比与并行调度结果

节点选择实验固定后续修复树为时间感知修复树，仅替换候选节点选择方法，并使用相同的DRRNS综合评分口径评价各方法选出的节点集合．在100轮测试后的平均综合评分如表1所示．DRRNS在三种RS配置下的评分均最高：相对Ran-

dom、BW-based、AHP-TOPSIS和EW-TOPSIS，平均评分分别提高9.76%～10.53%、4.79%～5.01%、2.11%～2.51%和0.09%～0.17%．所有评分比较配对Bootstrap置信区间均大于0，且校正后$p\leqslant 0.0035$．

表1　不同节点选择方法的平均综合评分

RS配置	Random	BW-based	AHP-TOPSIS	EW-TOPSIS	DRRNS
RS(9,6)	0.4793	0.5010	0.5145	0.5252	0.5261
RS(12,8)	0.4843	0.5104	0.5239	0.5345	0.5350
RS(15,10)	0.4881	0.5149	0.5263	0.5387	0.5395

节点评分优势不等同于链路时间优势．仅改变节点选择方法时，DRRNS相对各基线的并行工期在-2.12%～7.04%范围内变化，全部配对比较经全局Holm校正后均不显著($p_{\mathrm{Holm}}=1$)．这是因为1.2节的节点评分不包含Q_{ij}和路径时间，节点资源状态与随机生成的链路状态也未设相关性．该结果说明节点选择阶段主要保证恢复节点的综合资源质量，链路时间优化需要由1.3节的修复树阶段继续完成．

修复树实验固定DRRNS选出的恢复节点，只替换父节点选择和树构建策略．实验结果见表3，时间感知修复树在三种RS配置下均取得最小的关

键路径并行工期，且显著优于Random、Bandwidth、Score及Score+Q四种树基线(12组配对比较经Holm校正后$p\leqslant 5.38\times10^{-5}$)，其平均树深为2.19～2.64，整体低于$D_{\max}=4$．为进一步分析深度与扇出约束的边界达到情况，统计各配置下100轮修复树构建结果，如表2所示．三种配置下，最大子节点数均在多数轮次达到扇出上限；树深达到深度上限的情况仅在RS(12,8)中出现少量轮次．全部300轮结果均未发生约束违规，说明生成的修复树满足给定的拓扑约束．

表2　时间感知修复树的约束边界统计

RS配置	树深达到上限的轮次	最大子节点数达到上限的轮次	约束违规轮次
RS(9,6)	0	57	0
RS(12,8)	5	79	0
RS(15,10)	0	96	0

修复树比较中Bandwidth Tree和Score+Q Tree的平均链路质量高于时间感知修复树，但其工期同时更长；Score Tree的树深固定为2，却具有更高的转发负载离散度．上述现象与父节点选择函数的多目标性质一致，时间感知修复树不追求单一链路质量或最浅拓扑，而是在路径传输时间、节点能力、链路质量和转发负载之间进行权衡．

端到端实验同时替换节点选择和修复树方法，所提方法为“DRRNS+Time-aware Tree”．实验结果见表4，与Random + Random Tree、BW-based + Bandwidth Tree、AHP-TOPSIS + Score Tree和EW-TOPSIS + Score Tree相比，在三种RS配置下，所提方法的平均并行工期分别减少52.52%～60.01%、28.06%～35.30%、31.60%～39.04%和32.93%～38.84%．全部12组均值差的95%配对Bootstrap置信区间均大于0，且校正后$p\leqslant 1.32\times10^{-4}$，充分说明了DRRNS+Time-aware Tree能够降低恢复过程中的节点选择代价和多跳传输开销，并进一步改善传输任务的并行执行效率．

表3　不同修复树方法的平均理论并行工期(s)

修复树方法	RS(9,6)	RS(12,8)	RS(15,10)
Time-aware Tree（本文方法）	10.9229	14.3151	16.1573
Random Tree	22.3942	31.3108	36.6113
Bandwidth Tree	16.1004	20.1870	23.6928
Score Tree	16.3554	23.5839	25.6711
Score + Q Tree	16.2538	20.6217	24.1838

表4　不同端到端恢复方法的平均理论并行工期(s)

端到端恢复方法	RS(9,6)	RS(12,8)	RS(15,10)
Random + Random Tree	23.0076	30.9906	40.4032
BW-based + Bandwidth Tree	15.5772	19.8985	24.9709
AHP-TOPSIS + Score Tree	15.9699	23.4826	25.4785
EW-TOPSIS + Score Tree	16.2868	23.4052	25.7053

在并行调度实验中，进一步比较顺序调度、深度同步调度和关键路径优先调度三种方法．为保持对比策略一致，顺序调度按子节点深度降序逐边执行，同一时刻仅运行一个任务；深度同步调度按树深分层推进，同层无共享父节点的任务可并行执行，下一层需等待当前层全部任务完成．实验结果如表5所示，与顺序调度相比，关键路径优先调度的平均工期降低23.82%～35.28%；与深度同步调度相比，平均工期降低18.25%～24.57%，这表明采用关键路径长度确定任务优先级能够减少非关键任务对传输资源的占用，从而提高修复树中可并行传输任务的利用程度．此外，关键路径优先调度的平均加速比达到1.328、1.471和1.568，平均调度工期与理论下界之间的相对间隙小于2.5%，说明该方法能够在满足数据汇聚依赖和节点独占约束的条件下充分获取修复树中的并行传输机会．随着恢复节点规模扩大，修复树中的可调度任务数量及潜在并行关系增加，但实际加速效果仍受到任务前驱

依赖和节点资源独占约束的共同限制.

表5的结果来自独立开展的100轮调度策略对比实验，其随机测试实例与3.3节消融实验不同.两组实验虽然均采用时间感知修复树和关键路径优先调度，但分别用于评价调度策略和方法组成模块.

表5　不同调度策略的工期及关键路径优先调度性能

RS配置	顺序调度工期/s	深度同步调度工期/s	关键路径优先调度工期/s	平均加速比	调度间隙/%
RS(9,6)	14.0311	13.0753	10.6884	1.328	1.30
RS(12,8)	19.8408	17.7154	13.6815	1.471	1.89
RS(15,10)	27.0992	23.2516	17.5381	1.568	2.41

3.3　消融实验

消融实验与修复树及端到端比较采用相同的100轮随机测试实例，分别去除动态资源演化(w/o-DE)、AHP权重(w/o-AHP)、熵权(w/o-EW)、历史特征(w/o-HF)以及父节点函数中的综合评分项(w/o-Score)、链路质量项(w/o-Q)、累计路径传输代价项(w/o-Time)和转发负载项(w/o-Load).各消融变体的平均理论并行工期如表6所示，完整方法的共同参照值见表3.w/o-Time使工期显著增加(校正后$p=6.63\times10^{-28}$)，其余变体与完整方法的差异经Holm校正后均未达到显著水平，表明显式路径时间是控制关键路径工期的主要因素.

辅助指标结果见表7.节点评价侧的消融结果与各模块的作用范围一致.完整方法相较于w/o-EW，

表6　各消融变体的平均理论并行工期(s)

消融变体	RS(9,6)	RS(12,8)	RS(15,10)
Full Method	10.9229	14.3151	16.1573
w/o-AHP	11.0377	14.2979	16.5543
w/o-DE	11.0815	14.2584	16.6781
w/o-EW	11.0724	14.4130	16.5256
w/o-HF	10.7937	14.0828	16.5756
w/o-Load	10.8403	14.1315	16.3399
w/o-Q	10.9852	14.2536	16.1443
w/o-Score	10.8278	14.1592	16.1966
w/o-Time	32.5289	43.7084	46.3672

统一综合评分提高了2.11%～2.51%；相较于w/o-HF提高了1.56%～2.02%，表明组合赋权能够改善多资源指标的综合表征结果.在平均链路质量方面，完整方法在三种RS配置下均高于w/o-Q，其中，RS(12,8)和RS(15,10)的差异达到显著水平，而RS(9,6)经全局Holm校正后差异不显著($p=0.109$).在子节点数标准差方面，完整方法相较于w/o-Load分别降低5.85%、7.19%和9.01%，三种配置下的差异均达到显著水平($p\leqslant2.70\times10^{-7}$).除w/o-Time外，其余单项消融对并行工期的差异经全局Holm校正后均未达到显著水平.这表明Q项和ℓ项主要用于保持链路质量与转发负载约束，不能仅依据单一工期均值判断其必要性.

表7　关键消融指标均值

RS配置	统一综合评分		平均链路质量		子节点数标准差	
	w/o-EW	w/o-HF	完整方法	w/o-Q	完整方法	w/o-Load
RS(9,6)	0.5145	0.5157	0.6828	0.6722	1.2686	1.3475
RS(12,8)	0.5239	0.5244	0.6963	0.6803	1.2946	1.3950
RS(15,10)	0.5263	0.5313	0.7172	0.7020	1.2988	1.4274

4　结语

针对纠删码分布式存储故障恢复中候选节点资源动态变化、异构链路多跳传输代价较高以及树上传输任务并行性不足等问题，本文提出一种基于动态资源感知的故障恢复方法.该方法将恢复过程划分为节点选择、修复树构建和并行调度三个连续阶段.首先采用DRRNS描述候选节点的资源状态演化，通过AHP、熵权组合赋权与TOPSIS评价选择综合资源状态较优的恢复节点；其次综合节点评分、链路质量、累计路径传输代价和转发负载构建时间感知修复树，协调多跳传输中的路径代价与节点负载；最后在满足数据汇聚依赖和节点独占约束的条件下，采用关键路径优先策略调度树上传输任务.实验结果表明，在3种RS配置下，所提方法显著优于其他4种端到端组合，实现了动态节点选择、传输拓扑优化与任务调度的有效衔接，降低了纠删码故障恢复的理论工期.

参　考　文　献

[1] Shen Z R，Cai Y H，Cheng K Y，et al. A survey of the past，present，and future of erasure coding for storage systems[J]. ACM Transactions on Storage，2025，21(1)：1-39.

[2] 陈李，程建华，李明明. 基于TCN-VLSTM网络的全脉冲多雷达信号调制识别方法[J/OL]. 系统工程与电子技术，2026-09-02. https://link.cnki.net/urlid/11.2422.TN.20260902.1610.004.

[3] 邓见光，赵跃龙，袁华强. 云计算环境下的动态数据副本管理策略[J]. 华中科技大学学报(自然科学版)，

2015，43(10)：53-57．

[4] Lucas Filho E R，Efstathiou A，Yang L，et al．DITIS：an end-to-end system-level simulator and optimizer for distributed tiered storage[J]．SN Computer Science，2025，6(6)：746．

[5] Li J，Zhou Z R，Ding J L．Data-loss models for proactive-tolerance Reed-Solomon storage systems[J]．Future Generation Computer Systems，2025，170：107832．

[6] Chen J M，Li Z P，Fang G，et al．A comprehensive repair scheme for distributed storage systems[J]．Computer Networks，2023，235：109954．

[7] Chen N J，Liu W T，Pu W J，et al．SDNC-repair：a cooperative data repair strategy based on erasure code for software-defined storage[J]．Sensors，2023，23(13)：5809．

[8] Xia J X，Luo L L，Sun B W，et al．Parallelized in-network aggregation for failure repair in erasure-coded storage systems[J]．IEEE/ACM Transactions on Networking，2024，32(4)：2888-2903．

[9] Liu L，Wang Y，Liang Y F，et al．In-network aggregation enabled multiple sub-blocks parallel repair in erasure-coded storage system[J]．Computer Networks，2025，270：111523．

[10] Yang C H，Zhong K，Tan Y J，et al．RAN：accelerating data repair with available nodes in erasure-coded storage[C]//Proc of 2025 IEEE International Conference on Cluster Computing．Piscataway，NJ：IEEE，2025：1-12．

[11] Liu X Z，Hu Y C，Feng D，et al．Repair friendly wide-stripe erasure coding for in-memory key-value stores[J]．Journal of Systems Architecture，2025，167：103516．

[12] 许亮业，石连星，单蓉胜．一种纠删码条带化数据的一致性检查方法[J]．上海交通大学学报，2024，58(4)：579-584．

[13] 邓文杰，洪铁原，唐聃，等．一种基于多层次校验的低恢复成本纠删码[J]．计算机应用研究，2024，41(5)：1441-1447．

[14] Zhou Z J，Deng X H，Pei X J，et al．ZJC：constructing fully local repair in erasure codes for distributed cloud storage[J]．Journal of Systems Architecture，2026，174：103735．

[15] Liu X Z，Hu Y C，Wang W C，et al．Optimizing encoding and repair for wide-stripe minimum bandwidth regenerating codes in in-memory key-value stores[J]．Journal of Systems Architecture，2025，161：103369．