软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.006986]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

移动分布式存储系统中自适应数据布局策略*

伍代涛[1], 谭玉娟[1,3], 刘 铎[1,2], 魏鑫蕾[1], 吴 宇[1], 陈咸彰[1], 乔 磊[4]

[1](重庆大学 计算机学院, 重庆 400044)

[2](重庆大学 大数据与软件学院, 重庆 400044)

[3](武汉光电国家研究中心, 湖北 武汉 430074)

[4](北京控制工程研究所, 北京 100190)

通信作者: 刘铎, E-mail:liuduo@cqu.edu.cn

摘 要: 分布式存储系统在移动网络场景中正受到越来越多的关注, 作为其关键技术, 数据布局对于提高数据分布式存储的成功率至关重要. 然而, 移动环境下无线信号不稳定, 网络带宽波动大, 传统的数据布局策略, 如随机策略和存储容量感知策略, 在数据布局时并未考虑节点的网络带宽, 导致数据传输成功率低. 面向高动态移动网络环境, 针对移动分布式存储系统面临的数据布局问题, 提出一种带宽感知的自适应数据布局策略. 其基本思想是将网络带宽和节点上的其他信息结合, 从而选择性能良好的节点, 实现自适应数据布局, 提高数据传输成功率. 所提策略包含 3 个设计要点: (1) 采用群组移动模型感知节点的网络带宽; (2) 分组管理节点信息, 减少通信开销, 并利用小根堆的特性构建节点选择树; (3) 自适应数据布局根据节点可用性动态选择性能良好的节点, 提高数据传输成功率. 实验结果表明: 当网络动态变化时, 所提策略的数据传输成功率相较于随机策略和存储容量感知策略分别提升 30.6%, 34.6%, 并始终将通信开销维持在较低的水平.

关键词: 分布式存储; 数据布局; 带宽感知; 移动网络; 群组移动模型

中图法分类号: TP302

中文引用格式: 伍代涛, 谭玉娟, 刘铎, 魏鑫蕾, 吴宇, 陈咸彰, 乔磊. 移动分布式存储系统中自适应数据布局策略. 软件学报. http://www.jos.org.cn/1000-9825/6986.htm

英文引用格式: Wu DT, Tan YJ, Liu D, Wei XL, Wu Y, Chen XZ, Qiao L. Adaptive Data Placement Strategy in Mobile Distributed Storage System. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6986.htm

Adaptive Data Placement Strategy in Mobile Distributed Storage System

WU Dai-Tao[1], TAN Yu-Juan[1,3], LIU Duo[1,2], WEI Xin-Lei[1], WU Yu[1], CHEN Xian-Zhang[1], QIAO Lei[4]

[1](College of Computer Science, Chongqing University, Chongqing 400044, China)

[2](School of Big Data & Software Engineering, Chongqing University, Chongqing 400044, China)

[3](Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China)

[4](Beijing Institute of Control Engineering, Beijing 100190, China)

Abstract: Distributed storage system is receiving more and more attention in mobile network scenarios. Data placement, a key technology of distributed storage, is crucial to improve the success rate of distributed data storage. However, due to unstable wireless signals and fluctuating network bandwidth in mobile environments, the traditional data placement strategies, such as random placement strategy and storage-aware placement strategy, have low success rates of data transmission because both of them do not take network bandwidth into account during data placement. To solve the problem faced by mobile distributed storage systems, this study proposes a bandwidth-aware adaptive data placement strategy (BADP). The main breakthrough is that BADP adopts the group mobility model to sense the network

* 基金项目: 国家自然科学基金 (62072059); 武汉光电国家研究中心开放课题 (2019WNLOKF009); 重庆市自然科学基金 (cstc2020jcyj-msxmX0897); 中央高校基本科研基金 (2020CDJLHZZ-050); 重庆市杰出青年科学基金 (cstc2020jcyj-jqX0012); 重庆市技术创新与应用发展重点项目 (cstc2019jscx-mbdxX0022)

收稿时间: 2022-10-12; 修改时间: 2023-04-03; 采用时间: 2023-06-09; jos 在线出版时间: 2023-11-01

bandwidth of nodes and takes the network bandwidth as an important factor for data placement, thus selecting nodes with good performance to achieve adaptive data placement and improve the success of data transmission. BADP consists of three design features: (1) adopting the group mobility model to sense the network bandwidth of nodes; (2) managing node information in groups to reduce communication overhead, and taking advantage of the heap to build a node selection tree; (3) selecting nodes with good performance using adaptive data placement to improve the success rate of data transmission. Experiments show that when the network changes dynamically, BADP gains at least 30.6% and 34.6% improvements in the success rate of data transmission compared with random placement strategy and storage-aware placement strategy. At the same time, it consistently keeps communication overhead low.

Key words: distributed storage; data placement; bandwidth aware; mobile network; group mobility model

近年来, 随着通信技术的迅速发展和物联网时代的来临, 分布式存储系统在移动场景中受到越来越多的关注. 传统的分布式存储系统 (如 GFS[1], Dynamo[2]等) 通常部署在数据中心用于存储海量的数据. 而移动分布式存储系统[3−5]被广泛部署到各种具有存储和计算能力的移动终端节点上, 用以代替人类执行高风险的信息收集、传输和处理等任务, 例如在地震区域进行灾难救援任务或偏远地区的情报收集作业[6,7]等. 如图 1 所示, 无人机、无人车、手机等多个移动终端节点通过无线网络构成移动分布式存储系统协同工作, 并采用点对点的通信技术进行数据共享和存储, 从而避免因单点故障导致数据不可用的情况.

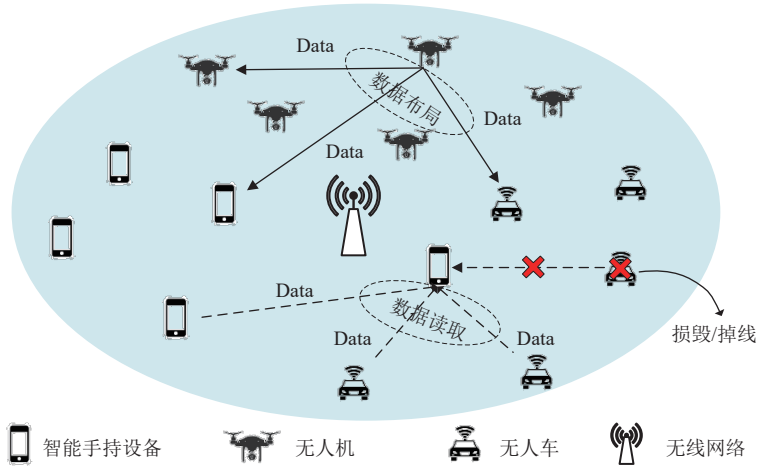


图 1　移动分布式存储系统的应用场景

然而在移动环境下, 由于廉价节点自身的硬件故障、外部环境的影响和移动网络不稳定等因素, 节点的故障率高[8], 数据容易丢失, 数据的可用性低. 为了保证数据的高可靠性, 移动分布式存储系统通常会使用多副本、纠删码等数据容错机制将单节点上的副本块或编码块实时地布局到多个节点上存储, 从而提高集群的容错能力. 但是移动环境中网络波动幅度大, 节点间网络带宽具有差异化, 节点的故障率高, 给数据布局带来了新的挑战. 针对这些问题, 本文对移动分布式存储系统中数据布局展开了研究.

现有的数据布局策略常应用于数据中心的分布式存储系统中, 主要分为两种: 随机数据布局策略 (Random) 和基于存储容量的数据布局策略 (Storage-aware). Random 策略多考虑请求的响应速度, 尽可能迅速地定位到存储节点, 例如 Google 文件系统 (GFS)[1]的数据布局策略, 从可用节点中随机选择候选节点存储数据块, 快速响应读写请求; Storage-aware 策略则将节点存储容量作为布局决策的关键因素, 如 Ceph 分布式文件系统[9]的 CRUSH 算法[10]通过每个存储设备的存储权重来决定数据对象的分布, 使存储负载更加均衡. 然而上述两种数据布局策略主要应用在高性能数据中心, 网络环境相对稳定且带宽很高. 但在移动环境下, 网络状态不稳定, 网络带宽相对较低, 不合理地直接应用这两种数据布局策略会严重增加数据传输所需要的时间. 比如, 在布局时错误地选择一个容量很大但网络带宽差的节点, 很容易出现传输中断, 而在传输中断后又需要重新选择其他节点进行数据布局, 严重增加网络负担, 影响数据传输效率. 并且, 随着节点的移动, 节点间的网络带宽动态变化, 节点故障不断出现, 数据可靠性

和传输成功率都将进一步降低. 因此, 在移动环境下进行高效的数据布局面临着巨大挑战.

(1) 数据布局时需要感知节点间的可用带宽, 以提升数据传输成功率. 但是, 在移动网络环境下, 由于网络资源有限, 采用传统方式主动探测节点间的网络带宽[11]将浪费宝贵的网络资源, 且用于探测的无线信号的发射和接收也将消耗节点所储备的有限电量. 因此, 如何在移动环境下感知节点间的网络带宽是一大难题.

(2) 移动节点的状态信息难以管理和维护. 在数据布局时, 每个节点需要知道其他节点的状态信息, 如网络带宽, 存储容量等, 若每个节点都维护一份其他节点的状态信息, 整个系统逻辑上相当于一个完全图. 并且, 随着节点个数的增加, 节点间的通信开销指数级增长, 如何减少维护节点状态信息所需的通信开销是一大难题.

(3) 移动环境下节点的故障率高, 数据可靠性遭遇挑战. 在分布式存储系统中, 为了保证数据的可靠性和可用性, 通常采用副本和纠删码来增加数据的冗余度[12]. 然而, 在移动网络环境下, 网络资源极其受限, 在多个节点之间采用副本或纠删码增加数据冗余度需要大量的网络通信资源, 如何合理应用容错策略进一步提高数据冗余度是移动分布式存储系统面临的一大难题.

为了解决上述数据布局面临的问题, 提高移动网络环境下分布式存储系统数据布局的性能, 本文提出基于带宽感知的自适应数据布局策略 (bandwidth-aware adaptive data placement, BADP), 与传统的布局策略相比, BADP 策略具有如下 3 个方面的特点和优势: 1) 将群组移动模型应用到数据布局中, 感知节点间的网络带宽, 以避免主动获取网络带宽所需要的网络通信开销. 2) 分组管理数据布局时所需要的节点状态信息, 缩小节点间的通信次数, 以减少通信开销, 并使用小根堆数据结构根据节点综合性能构建节点选择树. 3) 自适应数据布局策略根据节点可用性, 动态选择目的节点, 从而应用不同的冗余容错机制提高数据的可靠性. 本文的主要贡献总结如下.

(1) 深入分析了移动分布式存储系统中网络带宽是影响数据布局的关键因素, 现有的随机策略和存储容量感知策略都无法直接应用于网络动态变化的移动网络环境中.

(2) 提出了适用于移动分布式存储的 BADP 数据布局策略, 将网络带宽作为影响数据布局的重要因素, 并采用群组移动模型, 节点信息分组管理, 自适应数据布局策略提高数据传输成功率.

(3) 经原型验证和仿真实验, 与传统的随机布局策略和存储容量感知的布局策略相比, BADP 策略最大能将数据传输成功率提高 42.1%.

本文第 1 节数据布局相关工作. 第 2 节介绍研究动机. 第 3 节给出数据布局问题的形式化描述与移动模型的介绍. 第 4 节提出本文策略的总体设计. 第 5 节通过原型验证和仿真实验验证策略的有效性. 第 6 节总结本文的工作, 同时展望未来的研究方向.

1 相关工作

数据布局是分布式存储系统的一个重要研究方向, 其能够根据系统的需要进行数据的合理分配, 从而提高数据的可靠性和可用性, 是一个建立数据与存储节点之间映射关系的过程. 目前关于数据布局的研究主要集中在数据中心的分布式存储系统, 而对于移动分布式存储系统中的数据布局研究较少.

高性能数据中心的数据布局通常采用随机布局策略放置数据块, 如 GFS[1]和 Cassandra[13], 或基于设备存储容量权重的布局策略, 如 Ceph[9], 以实现不同节点和不同机架之间的负载均衡. 同时也有考虑其他因素的布局策略, 如: Agarwal 等人[14]考虑数据相关性, 提出一种基于地理位置的自动数据布局策略, 尽可能减少数据调度带来的网络开销和请求响应时间. Yuan 等人[15]提出 K 均值聚类数据布局策略, 该策略根据数据相关度将数据集动态部署至合适的节点. 郑湃等人[16] 提出一种三阶段数据布局策略, 分别针对跨数据中心数据传输、数据依赖关系和全局负载均衡 3 个目标对数据布局方案进行求解和优化. 上述这些解决方案与本文所提出策略的主要区别在于, 它们关注的是高性能数据中心的数据布局, 而本文强调的是移动集群中的数据布局, 移动集群中网络带宽是影响数据布局性能的关键因素.

目前, 针对移动分布式存储系统中数据布局的问题, Huchton 等人[3]设计了一种 K-resilient 移动分布式文件系统 (MDFS), 采用随机策略将使用纠删码编码后的数据块随机传输到多个节点. Chen 等人[17]通过估计节点的故障

概率并监控网络拓扑的重大变化来考虑目标节点的选取, 从而减少系统的能耗. Hong 等人[18]分析了分布式存储系统在移动环境高信噪比条件下的故障恢复概率, 从而根据故障恢复概率, 找出分布式节点间存储最优的分配策略. 而本文主要关注的是移动集群中网络动态变化导致数据传输成功率低的问题.

在数据布局中, 为了保证数据的可靠性, 副本和纠删码是两种常用的容错机制. 副本机制通过为每个数据块存储多个副本来提供最简单的冗余形式. 例如 GFS[1]和 Hadoop 分布式文件系统 (HDFS)[19]默认存储 3 个副本, 可容忍任意两个节点故障. 作为一种替代方案, 纠删码能以更低的存储开销实现与副本相同的容错能力, 现已广泛部署于大规模的分布式存储系统中[20−22]. 例如, Facebook 的 HDFS-RAID[23]部署了 Reed-Solomon (*RS*) 纠删码[24], 其由两个可配置的参数 k 和 m 构造而成, 用 $RS(k,m)$ 表示, *RS* 将原始数据切分为 k 个大小相等的数据块, 并通过线性组合编码生成 m 个校验块, 这 $k+m$ 个编码块被称为一个条带, 能够容忍任意 m 个节点故障.

与现有的研究相比, 本文提出的基于带宽感知的自适应数据布局策略更适应高动态的移动网络环境. 其通过感知节点间的带宽, 选择目的节点, 在群组间使用传输数据量少的纠删码机制, 群组内在纠删码的基础上应用副本机制, 动态增加数据的冗余, 在保证数据传输成功率的同时, 提升数据的可靠性.

2 研究动机

移动分布式存储系统主要应用于移动边缘缓存网络[25]和移动 Ad Hoc 网络[26], 可由不同性能的移动设备组成, 比如手机、车辆, 以及无人机等, 在事故灾害现场勘测、战场情报收集等场景中有着广泛的应用. 但是在移动环境中, 网络状态不稳定, 数据难以在多个节点之间分散存储, 导致数据的可靠性低. 因此, 移动分布式存储要求在数据布局阶段尽可能将所有数据块成功存储到目的节点, 以保证数据的高可靠性.

2.1 传统数据布局策略的问题

传统的数据布局策略主要包括随机策略和存储容量感知策略, 这两种应用于数据中心的策略均未考虑到移动环境下节点网络带宽动态变化的情况, 导致数据传输成功率低. 作为衡量移动分布式存储系统中数据布局性能的重要指标, Wu 等人[27]将数据传输成功率 R_s 定义为:

$$R_s = \frac{N_{\text{suc}}}{N_{\text{total}}} \tag{1}$$

数据传输成功率表示一次数据布局中, 数据传输成功次数 N_{suc} 与数据传输总次数 N_{total} 的比值. 不合理的数据布局策略由于错误地选择目标节点, 导致数据传输中断, 会严重降低数据传输成功率, 增加数据传输的网络流量.

图 2 是在 $RS(3,2)$ 纠删码的配置下, 采用传统布局策略在移动分布式存储系统中存储块大小为 1 MB 的数据传输成功率. 实验结果表明, 随着节点移动速度的增加, 节点间通信质量急剧恶化, 采用这两种布局策略的数据传输成功率大幅下降, 平均下降约 50%. 同时, 移动环境下随机策略比基于存储容量感知策略的数据传输成功率稍高. 主要因为存储容量感知策略倾向于选择存储空间大的节点, 而这些节点很可能是由于带宽不好才导致所存储的数据少, 剩余可用空间大, 但此时选择这些节点作为目标节点, 数据传输很可能不成功. 因此, 使用存储容量感知的数据布局策略, 其整体的数据传输成功率要比随机策略差. 所以, 在数据布局时, 应综合考虑移动环境下影响数据布局的多种因素, 如网络带宽, 可用存储容量和剩余电量等因素. 否则, 不但无益于提升布局性能, 反而会造成更坏的结果.

2.2 带宽感知数据布局传统数据布局策略的问题

将网络带宽作为数据布局时的重要影响因素, 理论上可以极大地提高数据传输成功率. 但在移动集群中利用带宽信息做布局决策有两个难点: 一方面, 感知节点间的带宽需要网络通信开销. 以往的研究中获取带宽信息, 一般采用主动测量法, 通过向网络广播探测包, 并根据探测包所携带的信息来计算网络带宽. 但这样会引入额外的探测流量, 造成更多网络资源的浪费. 另一方面, 管理节点的状态信息也将消耗通信带宽, 因为每个节点在布局时需要知道其余节点的信息, 如果每个节点都维护一份其他节点的信息, 整个系统的拓扑连接相当于一个 $N\times N$ 的完

全图, 将耗费大量通信带宽.

图 3 所示的是移动分布式存储系统的节点数量 10–50 个, 两种传统数据布局策略用于管理节点状态信息所需的通信开销. 由于随机数据布局不需要获取节点的状态信息, 只需要定时的心跳检测信息 (约 17 B) 确保节点是否存活, 而基于存储容量的数据布局策略除了心跳检测信息以外还需要收集节点的存储空间信息 (约 50 B), 所以后者的通信开销比前者的多出两倍左右, 但是总的来说, 两者的通信开销都随着节点数量的增加呈指数级增长.

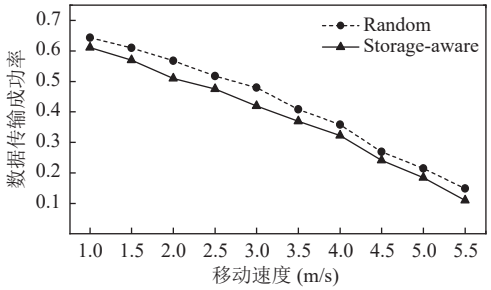


图 2 数据传输成功率随不同移动速度的变化趋势

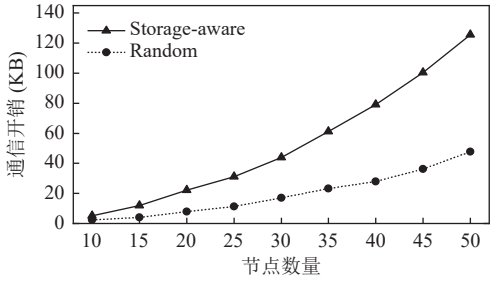


图 3 两种传统数据布局策略在不同节点数量下的通信开销

为了解决上述两个难点, 本文引入群组移动模型对无人集群的运动状态进行建模. 在实际应用中, 无人集群的移动轨迹不是毫无规则的, 而是根据任务场景的不同采用特定的移动模型. 文献 [28] 提出了关于群组移动模型的多种应用场景, 比如灾难区域信息分区收集, 可应用参考点群组移动 (RPGM) 模型. 移动模型影响着网络带宽的变化, 一个很朴素的想法是两节点之间距离越近, 无线信号越好, 传输速度越快. 通过节点当前的运动状态, 并结合节点间历史的带宽信息能够预测在接下来的一段时间内带宽的变化. 并且每个节点通过硬件设备能够获取到自己的运动状态, 不需要消耗网络资源. 同时, 按照移动模型分组管理节点的状态信息能够减少节点之间的通信开销[29].

2.3 现有容错技术存在的问题带宽感知数据布局

分布式存储系统中现有的容错技术主要有多副本和纠删码技术. 在移动环境下, 采用多副本布局时会在短时间内传输大量数据, 给系统造成巨大的网络压力. 相较而言, 采用纠删码技术在提供相同容错能力下能极大地降低网络开销, 但其在数据恢复时需要从多个节点获取 k 个数据块或校验块进行解码, 比副本需要更多的网络流量. 在高动态的移动网络环境中, 网络资源受限, 节点故障率高, 数据布局时需要以尽可能少的网络资源将数据更快地存储到目标节点, 但同时又需要快速恢复丢失数据以保证数据的可靠性. 因此, 本文采用自适应数据布局策略以提高数据可靠性, 在移动集群划分为多个群组后, 由于跨群组的可用带宽仅为内部带宽的 1/2–1/5[30,31], 群组间带宽通常被视为稀缺资源, 所以群组间使用传输数据量少的纠删码机制. 在此基础上, 组内根据节点状态信息自适应地选择多个性能良好的节点存储副本. 从而在保证数据传输成功率的同时, 提高数据可靠性.

综上所述, 为了更进一步提升移动分布式本文以移动模型为基础, 将网络带宽作为影响数据布局的重要因素, 在数据存储时尽可能选择网络带宽高的节点实现数据在群组间和群组内的自适应动态布局.

3 问题形式化描述与移动模型

本节首先对移动场景下数据布局问题进行形式化的描述, 然后介绍了本文使用的群组移动模型.

3.1 数据布局问题形式化

数据布局策略要解决的问题是在采用了数据容错机制的移动分布式存储系统中, 如何最优地将数据分配给不同的节点, 从而保证数据传输成功率、通信开销、可靠性等指标满足系统需求. 为了更清晰地理解数据布局问题, 本节将对其进行形式化描述.

假设由 N 个节点组成的分布式存储集群部署在边长为 $L\times L$, 高度相同的矩形区域中, 并执行长时间的数据收

集任务. 任务过程中, 集群的网络拓扑从逻辑上可以视为一个无向连通图$G=(V,E)$, 其中$V=(v_1,v_2,v_3,\ldots,v_N)$是节点集合, v_i表示每个节点的唯一标识. $E=\left\{E_{ij}|v_i,v_j\in V\right\}$是节点能直接通信的有限边集, 例如$E_{ij}$表示节点$v_i$和$v_j$之间的直接通信边. 设节点间的最大通信距离为$R$, 网络带宽的变化范围为$B_{\min}$–$B_{\max}$, 即在任意两个节点之间的实际距离$D\leqslant R$, 则两个节点可以直接通信. 基于上述假设, 我们定义以下术语确定数据传输成功率.

定义 1. 数据传输期望时间$T_{\exp}$.

在数据布局过程中, 如果数据能在期望的时间内传输完成, 那么称这个时间为数据传输期望时间$T_{\exp}$. 由于节点在数据传输过程中不断移动, 两节点之间的实际距离D大于最大通信距离R, 导致数据传输中断. 所以理想情况下, 数据期望在通信断开之间完成传输, 即数据传输期望时间还可以表示为从数据传输开始到节点间通信断开的时间.

定义 2. 数据传输时间T_{suc}.

数据实际完成点对点传输的时间称为数据传输时间, 用T_{suc}表示. T_{suc}与数据传输量F, 以及节点间的网络带宽B_{ij}有关, 如公式 (2) 所示.

$$T_{\text{suc}}=\frac{F}{B_{ij}} \tag{2}$$

某个时刻, 节点v_i需要将收集到的数据进行分布式存储, 首先将原始数据切分为k个数据块, 每块大小为F, 然后通过$RS(k+m,k)$纠删码编码得到包含n个编码块的条带$S=\{d_1,d_2,\ldots,d_k,p_1,\ldots,p_m\}$. 随后数据布局策略选择$n$个目标节点进行数据传输, 如果数据传输未在期望时间$T_{\exp}$内完成, 即$T_{\exp}<T_{\text{suc}}$, 那么系统会重新选择节点进行传输, 总的数据传输次数将增加. 由于数据传输成功率R_s为数据传输成功次数N_{suc}与数据传输总次数N_{total}的比值, 并且最终会有n个节点存储数据, 因此随着总的传输次数增加, 数据传输成功率将降低. 进一步地表示数据传输成功率R_s为:

$$R_s=\frac{N_{\text{suc}}}{N_{\text{suc}}+\sum_1^n p(T_{\text{suc}})}\quad \text{其中, } p(T_{\text{suc}})=\begin{cases}1, F/B_{ij}>T_{\exp}\\ 0, F/B_{ij}\leqslant T_{\exp}\end{cases} \tag{3}$$

由公式 (3) 可知, 在一次数据布局中, 数据传输成功率主要与数据传输成功次数N_{suc}(即n)、数据量大小F, 节点间网络带宽B_{ij}和数据传输期望时间$T_{\exp}$有关. 由于N_{suc}、F是系统存储机制决定的, 可以自由配置, 所以影响数据传输成功率的因素主要有B_{ij}、$T_{\exp}$. 由此可见, 如何挑选出网络带宽高, 数据传输期望时间大的节点对提高数据布局性能至关重要.

另一方面, 在由N个节点组成的集群中, 每个节点在数据布局时都需要知道全局节点的状态信息, 逻辑上是一个全连接图, 假设状态信息大小为K, 则在数据布局过程中会产生通信开销C为:

$$C=N\times N\times K \tag{4}$$

如果N很大, 则通信开销占据大量的网络带宽, 导致数据传输的可用带宽受到严重挤占, 数据传输成功率可预见地降低. 所以如何管理节点的状态信息, 减少通信开销也是数据布局要解决的问题.

3.2 参考点群组移动模型

无人集群在执行任务时往往不是随机移动的, 而是根据任务的需要在不同区域分组成群地移动. 因此, 本文采用 RPGM 参考点群组移动模型对集群的运动状态进行建模. 在 RPGM 模型下, 集群分为多个群组, 每个群组都有一个逻辑中心节点 leader 和多个 follower 节点. leader 的运动引导了群组的运动行为, 包括位置、速度、方向等. RPGM 模型通过为每个 leader 提供运动路径来引导群组的运动, 在初始化配置时, 沿着给定的时间间隔t定义一系列参考点, 给出群组将遵循的路径. 随着时间的推移, 一个组不断地从一个参考点移动到下一个参考点.

每当节点到达新的参考点时, 它根据当前和下一个参考点位置以及时间间隔t计算新的运动矢量V_T. 因此, 每个节点能够感知到自己接下来的运动状态, 然后将位置和速度信息通过心跳检测上报给 leader, 用于感知节点间的带宽变化. 通过正确选择参考点, 可以很容易地模拟许多现实情况. 比如整个灾难区域被划分为几个相邻区域, 每个区域有一个群组, 不同的群组在各自区域收集灾难现场的信息, 如图 4 所示.

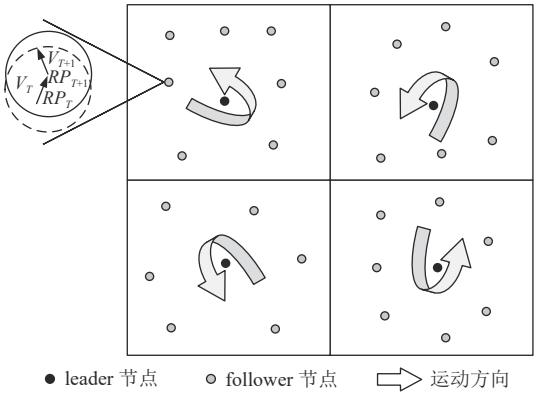


图 4 参考点群组移动模型示意图

图 4 还说明了一个节点从时间周期 T 到 T+1 是如何移动的. 首先, 节点以速度 V_T 从参考点 $RP(T)$ 移动到 $RP(T+1)$, 然后在新参考点$RP(T+1)$的基础上以一个随机的速度 V_{T+1} 移动到新的位置. 其移动轨迹表示为:

$$RP(T+1)=RP(T)+V_T\times t \tag{5}$$

该移动轨迹决定群组节点的动态变化, 实现群组在局部时间 t 内的移动. V_T 的大小在群组初始化时给定, 其方向在 0–2π 范围内等概率取值, 速度越大, 群组平均驻留时间越短, 则网络越不稳定, 反之则趋向静态网络. 第 4 节将介绍如何利用移动模型中节点的运动状态信息感知网络带宽.

4 带宽感知自适应数据布局策略

带宽感知自适应数据布局策略 (BADP) 的主要思想是将网络带宽信息和节点上的其他信息相结合, 选择网络带宽、存储容量以及剩余电量相对较高的节点作为目的节点, 实现数据自适应布局, 提升数据传输成功率. 图 5 给出了 BADP 的工作流程, 首先源节点向管理节点请求数据布局方案, 然后向 (k+m) 个群组分发数据, 最后由群组根据网络状况自适应增加副本数量. 其在总体设计上主要分为 3 个模块: 首先利用带宽感知模块预测目的节点的带宽信息, 然后将带宽信息和节点上的其他信息交给信息管理模块处理, 动态调整节点的选择顺序, 最后通过自适应布局模块选择群组或节点.

4.1 带宽感知

已有文献对分布式存储系统中带宽的预测方法表明未来带宽大小的变化与历史带宽大小是相关联的[32,33], 在短时间内带宽变化较小. 因此, 带宽感知模块主要通过节点当前的运动状态结合节点间历史的带宽信息预测在接下来的一段时间内带宽的变化. 图 6 给出了两个群组中节点的移动示例. 设在 T 时刻, 处于同一维度的两节点 N_i, N_j, 距离为 dis, 最大通信距离为 max_com_dis. N_i 的速度矢量为 V_i, 角度为 θ_i, N_j 的速度矢量为 V_j, 角度为 θ_j, 为了便于理解, 假设两节点速度大小相同, 仅方向不同. 则两速度矢量的夹角余弦$\cos\theta$公式为:

$$\cos\langle V_i,V_j\rangle=\frac{V_i\cdot V_j}{|V_i|\cdot|V_j|}=\frac{V_{i\_x}\cdot V_{j\_x}+V_{i\_y}\cdot V_{j\_y}}{\sqrt{V_{i\_x}^2+V_{i\_y}^2}\cdot\sqrt{V_{j\_x}^2+V_{j\_y}^2}} \tag{6}$$

设历史传输带宽 b_{ij}, 则在接下来的局部时间 t 内, 参考无线信号的衰减公式[34], 两节点间的网络带宽随着运动状态有如下的变化趋势:

(1) 当 $0\leqslant\theta_i<90°$, $90°\leqslant\theta_j<180°$时, 或者 $270°\leqslant\theta_i<360°$, $180°\leqslant\theta_j<270°$时, 两节点有相互靠近的趋势, 在接下来的局部时间内, 网络状况将变好, 此时的预测带宽如公式 (7) 所示, 在历史传输带宽 b_{ij} 的基础上, 有增大的趋势.

$$b_{ij}^T=b_{ij}+b_{ij}\times\left(1-\frac{\lg(dis-|\cos\theta|\times(|V_i|+|V_j|)\times t)}{max\_com\_dis}\right),\ -1\leqslant\cos\theta\leqslant 0 \tag{7}$$

(2) 其余情况, 两节点有相互远离的趋势, 在接下来的局部时间内, 网络状况将变差, 此时的预测带宽如公式 (8)

所示, 在历史传输带宽 b_{ij} 的基础上, 有减小的趋势.

$$b_{ij}^{T}=b_{ij}-b_{ij}\times\left(1-\frac{\lg(dis-|\cos\theta|\times(|V_i|+|V_j|)\times t)}{max\_com\_dis}\right),\ 0\leqslant\cos\theta\leqslant 1 \tag{8}$$

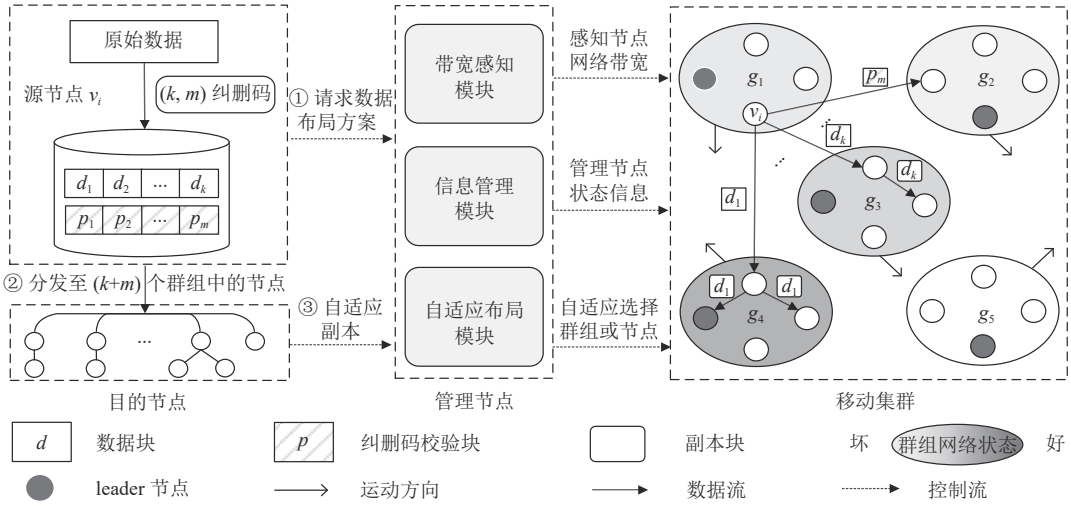


图 5 BADP 架构图

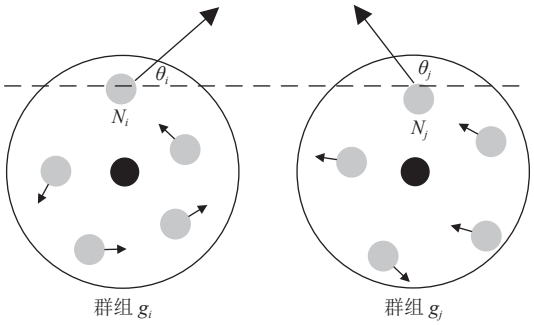


图 6 移动集群中两节点的运动状态

该策略只需要每个节点将由硬件获取到的速度、方向和位置信息通过心跳检测传输给管理节点 leader, 然后 leader 结合源节点与目的节点间的历史带宽信息, 从而预测出在接下来的时间 t 内节点间网络带宽的变化. 由于心跳检测是维持分布式存储系统不可缺少的操作, 因此不会造成额外带宽的消耗, 相比于主动测量获取节点间带宽的方法能够节约宝贵的通信开销.

4.2 信息管理

带宽感知策略通过节点的运动状态预测出节点间的网络带宽, 节约了主动探测消耗的通信开销. 然而, 不同于数据中心的分布式存储系统具有专门用于数据布局的服务器, 移动分布式存储系统中每个节点是对等的, 都可以自主的进行数据布局. 因此每个节点为了更新节点间的带宽信息矩阵, 所需的通信开销规模达到 $N\times N$ 级别.

针对上述问题, 节点信息分组管理根据群组移动模型对节点进行逻辑分组 (物理位置位于同一片区域). 如图 7(a) 所示, 一个拥有 6 个节点的移动分布式存储集群, 两两节点之间通过心跳检测传递运动状态信息 *Inf*, 将消耗大量的通信资源. 为了降低维护节点状态信息所需的通信开销, 图 7(b) 将集群分组, 每个群组根据节点电量选举出电量最充足的节点作为管理节点 leader, 其余节点将自己的运动状态信息通过心跳检测发送给 leader. 然后各 leader 之间同步群组中所有节点的信息. 最终, 各 leader 根据带宽感知策略计算出节点间的带宽信息矩阵 $\left[B_{ij}\right]_{N\times N}$.

当节点进行数据布局时, 通过询问所属群组的 leader 得到目标节点集.

为了分析该策略的性能, 假设系统中共有 N 个节点, 通过群组移动模型分组后有 g 个群组, 其中 $N \gg g$, 每次传输的信息大小为 K. 由于未分组前节点之间两两通信, 则通信开销 C_1 为:

$$C_1 = N \times N \times K \tag{9}$$

分组后, 节点只需和所属群组的 leader 进行单向通信, 而所有群组的 leader 之间两两通信, 通信量为群组中所有节点信息之和 $(N-g)\times K$, 则通信开销 C_2 为:

$$C_2 = (N-g)\times K + g \times g \times (N-g)\times K = (g \times g + 1)\times (N-g)\times K \tag{10}$$

由于群组个数远远小于节点个数, 所以 $C_2 \cong N \times K$, 因此该策略将通信规模从原本的 $N \times N$ 缩小到 N, 极大地降低了维护节点间带宽信息产生的通信开销.

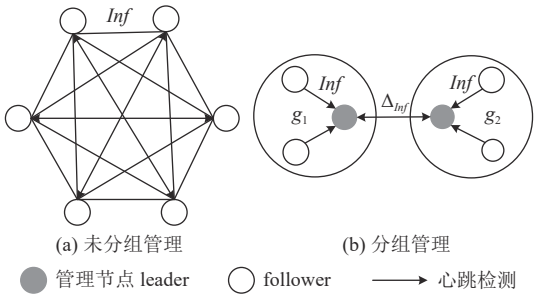


图 7 节点信息分组与未分组管理

4.3 自适应布局

为了充分利用副本和纠删码两种容错机制各自的优势, 弥补其在移动分布式存储系统中的不足之处. 自适应布局策略在移动集群划分为多个群组后, 由于跨群组的可用带宽通常被视为稀缺资源, 所以在群组间使用传输数据量少的纠删码机制. 在此基础上, 由于组内可用带宽相对充足, 该策略根据节点可用性自适应地选择多个性能良好的节点存储编码块的副本, 从而在保证较低存储开销的同时, 能够快速利用副本修复损毁数据, 提高数据可靠性.

由于群组间和群组内的自适应布局原理相同, 此处以群组内的布局过程作为详细介绍的例子. 定义目的群组中的节点属性集合如下:

$$DN = \bigcup_{i=1,2,\ldots,l} \{nd_i, na_i, ne_i, nb_i\} \tag{11}$$

其中, l 表示群组内的节点数量, nd_i 表示第 i 个有效节点, na_i 表示第 i 个节点的剩余可用存储容量, ne_i 表示第 i 个节点的剩余电量, nb_i 表示第 i 个节点与发送源节点之间的网络带宽.

对于每个编码块 b 而言, 都有数据量大小和副本数量两个属性, 分别用 f 和 c 表示. 其副本数量由群组内总网络带宽 B 决定, 在高动态的场景下, 在高动态的移动场景下, 为了尽可能保证数据传输, 副本数量根据群组内的网络带宽 B 自适应调整, 如公式 (12) 所示.

$$c = \max\left(1, \left\lfloor \frac{t \times B}{f} \right\rfloor\right) \text{ 即} c = \begin{cases} 1, & \dfrac{t}{2} \leqslant \dfrac{f}{B} \\ 2, & \dfrac{t}{3} \leqslant \dfrac{f}{B} < \dfrac{t}{2} \\ 3, & \dfrac{t}{3} > \dfrac{f}{B} \end{cases} \tag{12}$$

其中, t 为第 3 节带宽感知的局部时间, 公式 (12) 表示当群组内网络带宽较差, 编码块预计传输时间 f/B 超过 $t/2$ 时, 说明此时群组整体的数据传输性能差不足以传输两个副本, c 配置为单副本, 从而尽量减少注入网络中的数据, 保证数据传输成功率; 当群组内网络带宽较好, 编码块预计传输时间 f/B 小于局部时间 $t/2$ 但超过 $t/3$ 时, c 配置为 2 副本; 当群组内网络带宽良好, 编码块预计传输时间 f/B 不超过 $t/3$ 时, c 配置为 3 副本. 从而使群组中的编码块

副本数量根据网络性能自适应地做出调整. 因此编码块到群组内节点一对多的映射关系可以用决策函数$\delta(b)$表示.

$$\delta(b)=\bigcup_{i=1,2,\ldots,c}\{nd_i\} \tag{13}$$

决策函数δ对每个节点的3个维度属性进行计算, 得到节点可用性, 用公式(14)的Δ表示. 在公式(15)、(16)中, $prop_rc_i$表示第i个节点可用存储容量占群组总可用存储容量的比值; $proc_bc_i$表示第i个节点的传输性能占群组总传输性能的比值. 参数β作为可配置的权重比例, 其取值可以决定存储均衡和传输性能在自适应布局策略中的重要程度, $0<\beta<1$且默认情况下$\beta=0.5$. 节点可用性Δ描述节点对存储编码块副本的优先级, Δ越高则优先级越高.

$$\Delta=\beta\cdot prop\_rc_i+(1-\beta)\cdot prop\_bc_i \tag{14}$$

$$prop\_rc_i=\frac{na_i}{\sum_{i=1,2,\ldots,n}na_i} \tag{15}$$

$$prop\_bc_i=\frac{ne_i\times nb_i}{\sum_{i=1,2,\ldots,n}(ne_i\times nb_i)} \tag{16}$$

$proc_rc_i$越大表明节点可用存储容量越多, 系统写请求发生时节点存储负载较轻. $proc_bc_i$越大表明节点的数据传输性能越高, 电量充足且网络带宽高. 当一个节点可用存储容量超过平均值时, 被选中的概率会提高, 从而增大Δ的值导致存储编码块副本的优先级增加. 当一个节点的网络带宽增大, 使得传输性能超过平均值时, 也会相应增大Δ的值导致优先级增加.

在群组中对每个编码块进行副本布局决策时, 决策函数δ通过每个节点的属性依次计算出节点可用性Δ, 并利用小根堆特性快速地选出组内的前c个节点分别对应存储编码块的c个副本. 群组内的自适应布局算法语言描述如算法1.

算法 1. 群组内自适应布局算法.

输入: 节点列表 *node_list*, 编码块列表 *chunk_list*;
输出: 存储编码块到目标节点的映射 *map*.

```
1) FOR (chunk, c_replicas) IN chunk_list DO
2)     FOR node IN node_list DO
3)         na, ne, nb = getNodeInfo(node) /*获取节点信息*/
4)         sum_na += na
5)         sum_ne_nb += (ne × nb)
6)     END FOR
7)     FOR node IN node_list DO
8)         Δi = (β × na) / sum_na +
               ((1 – β) × (ne × nb) / sum_ne_nb)
9)         appendPairInto(node, Δi, heap) /*添加到小根堆*/
10)    END FOR
11)    LOOP c_replicas TIMES
12)        node = popNodeFrom(heap) /*选出 c 个副本节点*/
13)        appendPairInto(chunk, node, map) /*添加到 map*/
14)    END LOOP
15)    UPDATE node_list
16) END FOR
```

复杂度分析: 在每一轮迭代中, 收集L个节点信息的时间复杂度为O(L), 然后根据L个节点的放置权重Δ利

用小根堆特性选择 c 个副本节点, 其时间复杂度为 O(Llog$c$), 则最终总体的时间复杂度为 O(Llog$c$).

5 实验与分析

在本节中, 首先通过真实的硬件设备构建了原型系统, 验证了 BADP 策略在小范围真实场景下的可行性. 然后为了更全面地评估该策略的性能, 本文实现了一个移动分布式存储模拟器 MDSS-SIM (mobile distributed storage system simulator), 测试移动环境中的不同参数对 BADP 策略性能的影响.

5.1 原型系统验证

本文使用 20 辆无人车搭载嵌入式平台 (树莓派 4b) 构建了一个具有 20 个节点的移动分布式存储系统, 并在该系统上实现了本文所提出的 BADP 策略, 如图 8 所示. 树莓派 4b 的具体配置如表 1 所示.

图 8 原型系统

表 1 树莓派 4b 配置

名称	详细配置
硬件配置	CPU: Broadcom BCM 2711 64位 1.5 GHz 4核 内存: 4 GB DDR4 闪存: 64 GB 无线: 802.11ac协议 (2.4/5 GHz) 电源: 3 A, 5 V, 5 000 mA
软件配置	操作系统: Ubuntu 20.04 LTS 语言环境: C/C++

该原型系统使用 C/C++语言实现, 主要由通信模块、数据布局模块、元数据管理模块以及存储模块组成. 系统中的每个节点都是对等的, 均可以存储和读取数据. 但是每个群组在初始化时会通过一致性协议选举出 leader 节点管理群组各节点的状态信息, 负责为其余节点提供数据布局决策.

实验中, 集群节点通过最大通信距离为 150 m 的无线路由器 (TL-WDR7660 千兆版) 连接在同一个局域网中, 节点的最大移动速度为 3 m/s, 在 100×100 m^2 的空旷区域内按照 RPGM 移动模型分组运动, 最大带宽为 12.5 MB/s, 能够在最大通信距离内直接通信. 集群运行过程中, 实际网络带宽会随着节点的移动而发生改变. 首先对带宽感知策略的准确性进行了测试, 然后为了对比 BADP 在实际场景中的数据传输成功率, 还将随机策略 (Random) 以及存储容量感知策略 (Storage-aware) 部署到原型系统上.

5.1.1 带宽感知策略准确性验证

图 9 展示了移动集群运行过程中, 使用带宽感知策略预测的节点间网络带宽与使用 ping 方法测得的实际网络带宽的对比. 其中节点的最大移动速度为 3 m/s, 最大通信距离为 200 m, 带宽感知策略的局部时间 t 为 3 s. 实验过程中, 每隔 5 s 分别用两种方法对节点间的网络带宽进行采样. 从图中可以看出, 在 100 s 内, 本文所提出的带宽感知策略预测的节点间网络带宽虽然在实际的网络带宽上下浮动, 但是差距很小. 通过计算, 两种方法所得网络带宽的方差为 0.14, 说明带宽感知策略的准确性很高, 与实际测得的网络带宽差异很小. 理论上使用该方法得到的网络带宽进行数据布局能够极大地提高数据传输成功率.

5.1.2 块大小对数据传输成功率的影响

实验采用 RS(3, 2) 纠删码, 随机指定节点分布式存储块大小为 1 MB、4 MB、16 MB 的文件数据, 重复实验 20 次, 在相同条件下, 测得 3 种不同策略的数据传输成功率如图 10 所示.

相较于两种传统的策略, BADP 在不同的数据量大小下, 其数据传输成功率始终保持 30% 以上的优势, 这是由于该策略在选择节点存储数据时, 充分考虑了节点间网络带宽对数据布局的影响, 尽可能选择带宽良好的节点

作为存储数据的目的节点, 从而显著提高了数据布局的性能. 从图中还可以看出, 3 种策略的数据传输成功率都随着数据块的增大而减小. 因为当传输数据时, 发送者和接收者可能在移动, 无线连接变得不稳定. 由于数据量大, 需要的传输时间过长, 在此期间两节点连接中断的概率比起数据量小的时候更大, 导致数据传输成功率随着数据量增大而减小. 然而, 在传输数据量增大的过程中, BADP 策略的数据传输成功率只下降了 10%, 两种传统策略却下降了 20% 以上. 由此可见, 该策略在数据量较大时比传统策略更具有优势. 总体而言, BADP 策略在实际的移动环境中极大地提高了移动分布式存储系统的数据传输成功率, 从而保证了数据的可靠性.

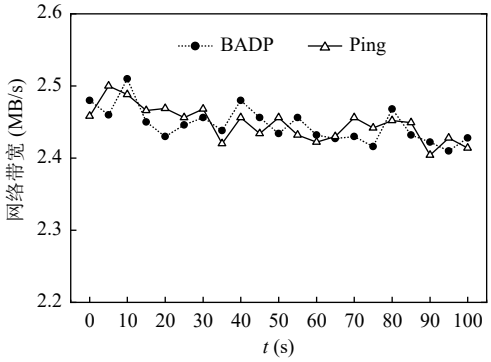


图 9 带宽感知策略与 ping 方法得到的节点间网络带宽对比

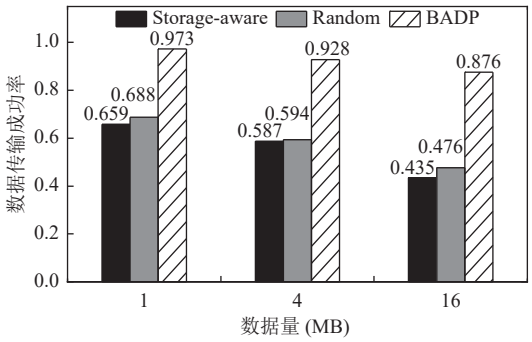


图 10 3 种策略存储不同数据量大小的数据传输成功率

5.1.3 不同容错策略的性能和开销

实验采用 *RS*(3, 2)、3 副本 (replica-3) 以及 *RS*(3, 2) 和 3 副本的混合容错方案 (hybrid), 随机指定节点分布式存储块大小为 1 MB 的文件数据, 测得在不同数据布局下的容错性能和网络开销, 其中容错性能为理论容错个数与数据传输成功率的乘积, 表示通过数据布局后数据实际的容错性能, 网络开销为传输数据所需的网络流量. 实验结果如图 11 所示, 从图 11(a) 可以看出混合容错策略的容错性能在不同的数据布局下相比于其他两种容错策略平均提升了 51.3%, 原因在于混合容错策略在纠删码的基础上动态地增加了 1–3 个冗余副本, 总体提升了容错性能. 但是从图 11(b) 可以看出, 其网络开销相比于纠删码平均增加了 90% 以上, 总体上接近于副本策略的网络开销. 同时, 图 11 还表明 BADP 策略相比于其他两种传统的数据布局策略, 无论是在容错性能还是网络开销方面都是最优的.

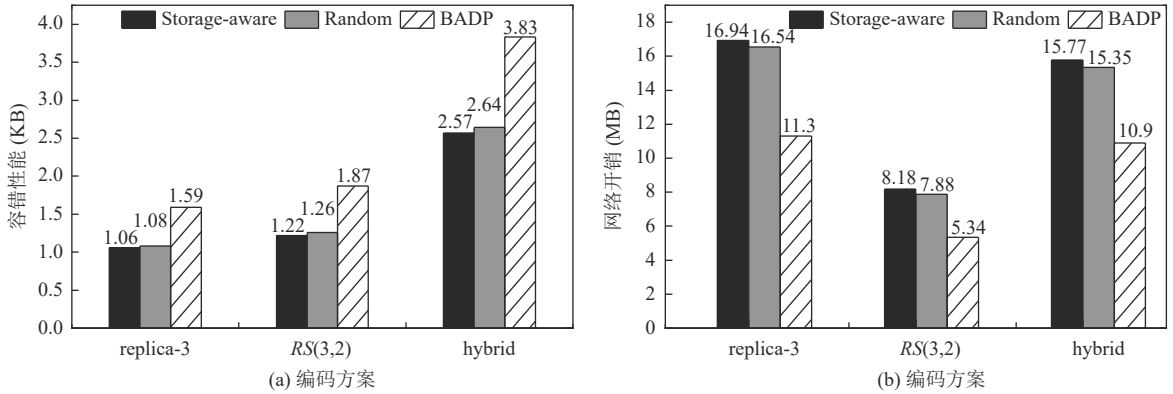


图 11 不同容错方案的性能和开销对比

5.2 大规模仿真

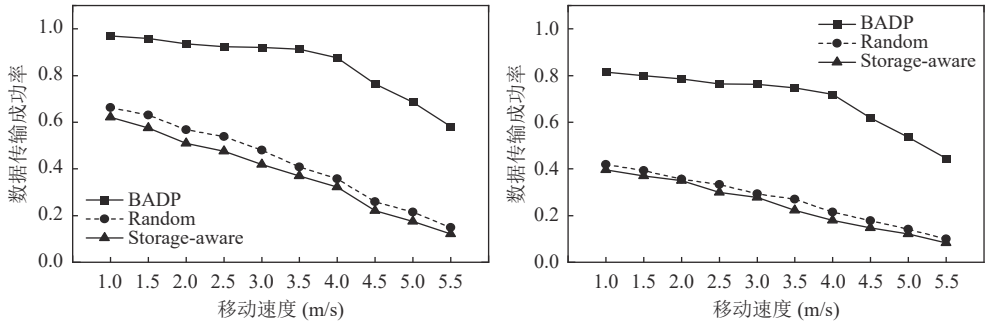
由于原型系统只能在小范围内改变集群在移动环境下的动态参数, 比如: 移动速度、通信距离、节点数量等,

因此, 本文实现了一个移动分布式存储模拟器 MDSS-SIM 来全面评估 BADP 策略. 该模拟器实现了分布式存储系统在移动环境下数据分发的仿真, 通过设置集群的移动模型、节点个数和移动区域的大小, 以及节点的最大通信距离、移动速度和状态信息, 从而模拟实际环境中移动节点存储数据的过程. 仿真器采用了 C++语言进行编写, 源码大概为 800 行. 在实验中, 集群的总移动区域设定为 500×500 m^2, 采用 RPGM 移动模型划分群组, 节点在该区域内均匀分布并遵循 RPGM 模型自由移动, 最大带宽设置为 12.5 MB/s. 在其上分别部署了 3 种数据布局策略, 随机选择节点存储 100 次数据, 每次块大小为 1 MB 的文件数据, 并采用 *RS*(3, 2) 纠删码编码原始数据, 重复实验 20 次, 计算平均的数据传输成功率.

实验将两种传统的数据布局策略与 BADP 策略进行比较. 为了对比不同策略在移动环境下的性能, 首先测试了 3 种参数对数据传输成功率的影响: 1) 节点移动速度; 2) 最大通信距离; 3) 节点数量. 这 3 种参数反映了集群的动态特性, 影响着节点网络带宽的变化, 从而影响数据传输成功率. 然后测试了节点数量对 3 种策略通信开销的影响. 最后为了评价带宽感知的有效性, 测试了不同移动速度下带宽感知时局部时间 t 和不同网络情况下自适应布局参数 β 对数据传输成功率的影响.

5.2.1 节点移动速度

图 12 展示了节点最大移动速度对数据传输成功率的影响, 其中节点数为 50 个, 最大通信距离为 100 m 和 300 m. 当最大移动速度从 1 m/s 变化到 5.5 m/s 时, 3 种策略的数据传输成功率都在逐渐下降, 尤其在移动速度超过 4 m/s 后, 下降得更明显. 这主要是因为更快的移动速度导致集群的网络波动更剧烈, 通信质量更加恶化, 节点更容易断开连接, 导致更多的数据块传输中断.



(a) 最大通信距离为 300 m (b) 最大通信距离为 100 m

图 12 节点移动速度对数据传输成功率的影响

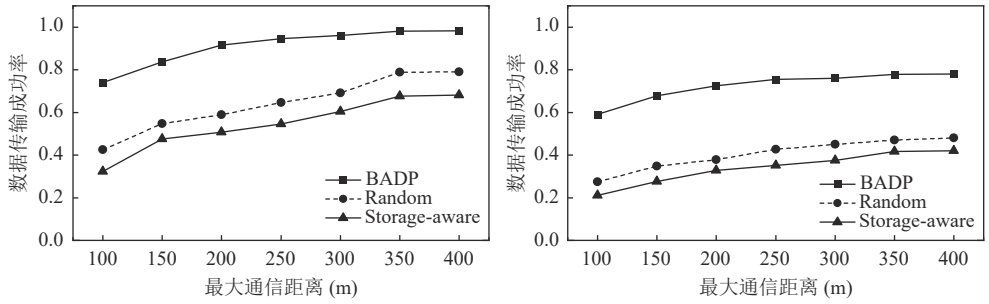
但是, 无论是图 12(a) 中最大通信距离为 300 m, 集群网络状况良好时, 还是图 12(b) 中最大通信距离为 100 m, 集群网络状况较差时, BADP 的数据传输成功率始终比两种传统策略高 35% 以上. 并且在 1–4 m/s 内, BADP 策略的下降幅度比其他两种更平缓, 这意味着在一定速度范围内 BADP 策略的性能更稳定. 因为该策略在选择目的节点时, 充分考虑了节点的带宽信息, 预测在接下来的局部时间内节点带宽的变化, 从而选择综合性能更好的节点存储数据. 并在组内自适应完成数据的动态备份, 由于群组内通信相对稳定, 数据传输成功率得以提升. 而其他两种策略没有将节点的网络状况纳入考虑范围, 显然在这种网络动态变化的移动环境下, 数据传输更容易中断, 导致数据传输成功率低.

5.2.2 最大通信距离

图 13 展示了集群最大通信距离对数据传输成功率的影响, 其中节点数分别设为 15 个和 50 个, 节点最大移动速度设为 1.5 m/s, 最大通信距离从 100 m 变化到 400 m.

在图 13(a) 节点密度大的情况下, 随着最大通信距离的增加, 集群的总体通信带宽越来越好, 通信在更长时间内是可靠的. 数据传输在移动中出现中断的情况减少, 数据传输成功率逐渐上升. 可以看到在最大通信距离从 100 m 增加到 200 m 时, 3 种策略的数据传输成功率都有很大提升的幅度, 提升约 20%. 200 m 以后, BADP 策略提升放

缓, 并且数据传输成功率已经达到 90% 以上. 而其他两种策略则继续大幅提升, 这说明 BADP 策略的适应性更强, 能更好更快地适应网络带宽低的移动网络环境. 而在图 13(b) 节点密度小的情况下, 由于可选节点数量太少, 3 种策略随着通信质量的改善, 提升的性能都很缓慢. 但总体上 BADP 策略的数据传输成功率比两种传统策略提高了 30% 左右, 尤其在图 13(a) 中最大通信距离增加到 200 m 以上后, 数据传输成功率已经达到 91.52%, 即传输 10 个数据块少于 1 个块失败, 这对于能容忍任意 2 个块失效的 *RS*(3, 2) 纠删码而言, 能够保证数据的可靠性. 而随机策略和存储容量感知策略的数据传输成功率最好时分别为 79.01% 和 68.9%, 在后续读取数据时无法重构出原始数据, 降低了数据的可靠性.

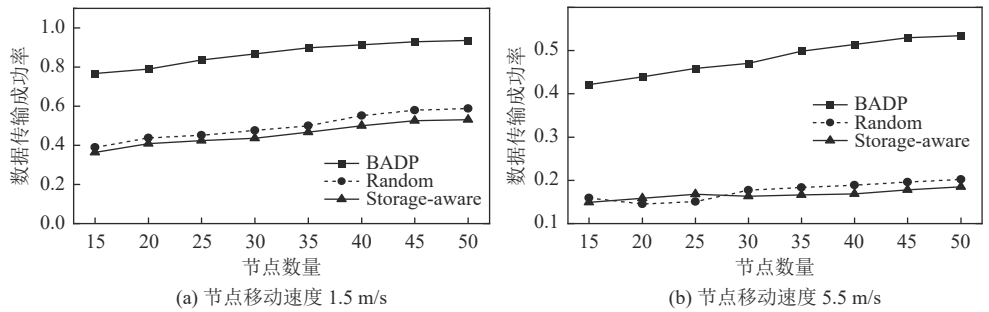


(a) 节点数量为 50　　(b) 节点数量为 15

图 13　最大通信距离对数据传输成功率的影响

5.2.3　节点数量

图 14 展示了集群中节点的数量对数据传输成功率的影响, 其中节点最大移动速度分别为 1.5 m/s 和 5.5 m/s, 最大通信距离为 300 m, 节点数量从 15 个增加到 50 个, 纠删码编码方案为 *RS*(3, 2). 在图 14(a) 最大移动速度为 1.5 m/s 时, 当节点数量从 15 个增加到 50 个时, 网络的连通性不断增强, 节点之间的无线信号变好, 所以 3 种策略的数据传输成功率逐渐提高, 平均都提升了 30% 左右. 最后随着节点均匀分布到整个移动平面后, 3 种策略提升的性能逐渐不明显. 原因是受最大通信距离的限制, 集群中节点带宽好坏的比例趋于固定. 但是在图 14(b) 最大移动速度为 5.5 m/s 时, 节点迅速移动, 随着节点数量的上升, 两种传统策略的数据传输成功率始终保持在极低的水平上, 没有明显提升. 而 BADP 策略在这种极端的情况下, 性能依然能够得到提升, 大约提升了 11.6%.



(a) 节点移动速度 1.5 m/s　　(b) 节点移动速度 5.5 m/s

图 14　节点数量对数据传输成功率的影响

从图 14 中可以看出, BADP 策略的数据传输成功率始终是最高的, 比随机策略平均提高了 37.7%, 比存储容量感知策略平均提高了 42.1%. 因为在数据布局时, 有更多的节点可供选择, 可以通过带宽感知策略选取性能更好的节点.

图 15 展示了在大规模移动分布式存储系统中, 节点数量的取值对通信开销 *C* 的影响, 其中节点最大速度为 5 m/s, 最大通信距离为 100 m. BADP 策略按 50 个节点一组分组管理节点信息, 而两种传统策略未分组. Random 策略每次传输的数据量大约为 17 B, Storage-aware 策略每次传输的数据量大约为 50 B, BADP 策略包含了心跳检

测和运动状态信息, 每次传输的数据量大约为 60 B. 为了测试 BADP 策略大规模集群中的性能, 实验过程中节点数量从 100 个增加到 1 000 个.

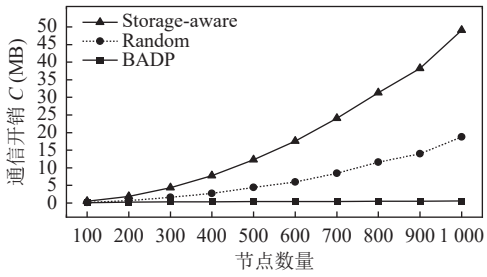
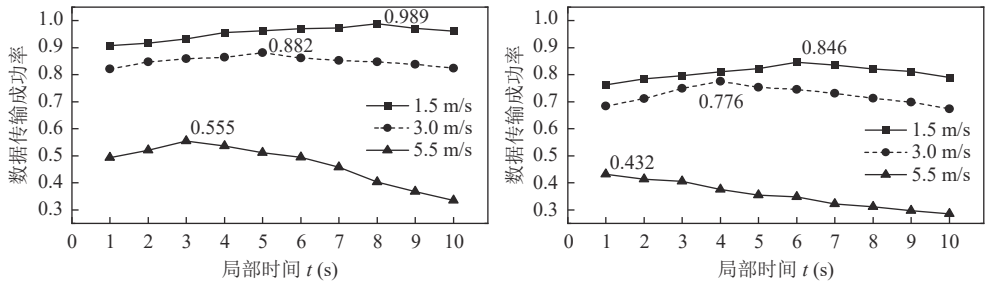


图 15　节点数量对通信开销的影响

从图 15 中可以看出, BADP 策略的通信开销一直处于很低的水平, 始终没有超过 1 MB, 而其余两种策略的通信开销在最初节点较少时处于较低的水平, 但随着节点数量不断增长, 两种传统策略的通信开销呈指数级增长, Storage-aware 策略增长了 96.4 倍, Random 策略增长了 89.1 倍. 在大规模移动集群中, 两种传统的数据布局策略浪费了太多宝贵的带宽资源, 挤占了属于数据传输的可用带宽, 进一步导致数据传输成功率的降低. 而本文提出的 BADP 策略的通信开销呈线性增长, 最终只增长了 7.8 倍. 由此可见, 随着节点数量大规模地增长, BADP 策略在移动环境中能以较低的通信开销为系统提供数据布局服务, 并且极大地提高系统的效率.

5.2.4　局部时间 t

图 16 显示了带宽感知时局部时间 t 的取值对数据传输成功率的影响, 其中节点数为 50 个, 最大通信距离为 300 m 和 100 m. 由于 t 的取值和速度大小有关, 因此该实验测试了节点最大移动速度为 1.5 m/s, 3.0 m/s, 5.5 m/s 时, t 取值从 1 s 到 10 s 的数据传输成功率.



(a) 最大通信距离为 300 m　　(b) 最大通信距离为 100 m

图 16　局部时间 t 对数据传输成功率的影响

由图 16 可知在不同的最大速度下, t 值设置得过大或过小都会导致数据传输成功率降低, 也就意味着通过带宽感知策略获取的带宽信息与实际带宽变化有差距. 原因是局部时间 t 是用于预测两节点间在接下来的时间 t 的网络带宽情况, 如果 t 取值太小, 预测的网络带宽表示两节点一瞬间短暂的带宽变化. 虽然在那一刻节点间的网络带宽是准确的, 但是数据传输是一个长期的过程, 在整个数据传输的过程中, 如果 t 取值太小, 得到的网络带宽不足以反映真实的情况, 可能在传输的过程中依然会出现传输中断的情况, 导致数据传输成功率下降, t 取值过大同理. 而且, 速度越大, 通信质量越差, 数据传输成功率的峰值就会越小并且越提前到来. 在最大通信距离为 300 m 时, 节点移动速度 1.5 m/s, 3 m/s, 5.5 m/s 出现最好情况时的 t 取值分别为 8 s, 5 s 和 3 s; 在最大通信距离为 100 m 时, 3 种移动速度出现峰值的 t 都有所提前, 分别为 6 s, 4 s, 和≤1 s. 这表明移动速度越大, 通信质量越差, 集群变动得越频繁, 保持相对稳定状态的时间越短, 导致能够预测的局部时间也越小, 数据能够稳定传输的时间就越短, 在还没有成功传输到目的节点时就出现了中断. 所以对于不同的移动速度, 合适的参数 t 对优化 BADP 的性能能够产生积极的作用.

5.2.5 参数 β

图 17 展示了自适应布局参数 β 的取值对数据传输成功率的影响, 其中节点数为 50 个, 最大通信距离为 300 m. 由于参数 β 的取值能够调整网络带宽在数据布局决策中的权重, 因此该实验测试了节点最大网络带宽为 1 MB/s、3 MB/s 和 5 MB/s 时, 参数 β 取值从 0.1 到 1.0 的数据传输成功率.

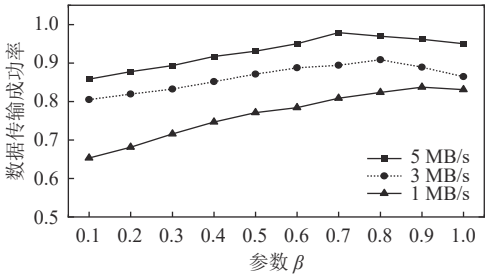


图 17 参数 β 对数据传输成功率的影响

从图 17 中可以看出, 在不同的网络情况下, 随着参数 β 的增大, 即网络带宽在数据布局决策中所占权重增加, 数据传输成功率总体呈上升趋势, 并且节点的最大网络带宽越小 (1 MB/s), 提升的比例越大. 当节点的最大网络带宽为 3 MB/s 和 5 MB/s, β 取值超过 0.8 和 0.7 时, 数据传输成功率出现下降趋势. 原因是当集群整体的网络情况良好时, 参数 β 取值过高使得存储空间的权重变小, 掩盖了存储空间在数据布局的作用. 但是在战场环境的极端网络条件中, 网络通信一般在 50 KB/s–1 MB/s 范围内, 此时存储空间对数据布局决策的影响较小, β 取值可自适应网络环境调整为 0.7–0.9, 从而得到最优的数据传输成功率. 因此对于分布式集群不同的网络情况, 合适的参数 β 对优化 BADP 的性能也能够产生积极的作用.

5.2.6 小 结

通过上述实验可以得出, 本文所提出的 BADP 策略, 其数据传输成功率在各种不同参数的影响下都是最优的, 比随机策略和存储容量感知策略的数据传输成功率分别提高 30.6%、34.6%, 在某些情况下能达到 42.1%; 并且随着集群规模的增长, BADP 策略能够维持较低的通信开销. 此外, 在实验中发现, 最大通信距离和移动速度对数据布局的数据传输成功率有着显著影响. 这两个参数直接影响通信质量, 导致节点之间带宽下降, 数据传输中断增加, 数据传输成功率下降. 而节点数量对数据传输成功率的影响较小, 但其对通信开销影响较大. 另外, 带宽感知预测时的局部时间 t 在不同移动速度和不同通信质量的情况下存在不同的最优值, 过大或过小都将影响带宽预测的准确度. 而自适应布局的参数 β 在不同的网络情况下存在不同的最优取值, 合适的参数 β 能够进一步提高 BADP 的数据传输成功率.

6 总结与展望

本文研究了移动分布式存储系统面临的数据布局问题. 传统的数据布局策略没有考虑节点在移动环境下网络带宽的差异, 数据传输成功率低. 本文提出了适用于移动分布式存储系统的 BADP 数据布局策略, 该策略将群组移动模型应用到数据布局中, 通过群组移动模型感知节点间的带宽信息, 并进一步对所有节点进行分组管理, 自适应选择节点采用不同的容错策略进行数据布局. 系统原型验证和大规模仿真实验证明: 使用 BADP 策略的数据传输成功率比使用随机策略和存储容量感知策略的数据传输成功率分别提升 30.6% 和 34.6%, 最好情况下能提升 42.1%. 并且随着集群规模的增长, BADP 策略仍然能够维持较低的通信开销. 下一步的工作是在 BADP 策略的基础之上, 研究如何利用副本和纠删码的混合容错机制对丢失的数据进行快速修复, 进一步提高移动分布式存储系统的可靠性.

References:

[1] Ghemawat S, Gobioff H, Leung ST. The Google file system. In: Proc. of the 19th ACM Symp. on Operating Systems Principles. Bolton

Landing: ACM, 2003. 29–43. [doi: 10.1145/945445.945450]

[2] DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshall P, Vogels W. Dynamo: Amazon's highly available key-value store. In: Proc. of the 21st ACM Symp. on Operating Systems Principles. Stevenson: ACM, 2007. 205–220. [doi: 10.1145/1294261.1294281]

[3] Huchton S, Xie G, Beverly R. Building and evaluating a K-resilient mobile distributed file system resistant to device compromise. In: Proc. the 2011 Military Communications Conf. (2011-MILCOM). Baltimore: IEEE, 2011. 1315–1320. [doi: 10.1109/MILCOM.2011.6127484]

[4] García-Valls M, Domínguez-Poblete J, Touahria IE, Lu CY. Integration of data distribution service and distributed partitioned systems. Journal of Systems Architecture, 2018, 83: 23–31. [doi: 10.1016/j.sysarc.2017.11.001]

[5] Calis G, Shivaramaiah S, Koyluoglu OO, Lazos L. Repair strategies for mobile storage systems. IEEE Trans. on Cloud Computing, 2021, 9(4): 1575–1591. [doi: 10.1109/TCC.2019.2914436]

[6] Razaque A, Elleithy KM. Low duty cycle, energy-efficient and mobility-based boarder node—MAC hybrid protocol for wireless sensor networks. Journal of Signal Processing Systems, 2015, 81(2): 265–284. [doi: 10.1007/s11265-014-0947-3]

[7] Yue YG, He P. A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions. Information Fusion, 2018, 44: 188–204. [doi: 10.1016/j.inffus.2018.03.005]

[8] Morreale P, Goncalves A, Silva C. Mobile ad hoc network communication for disaster recovery. Int'l Journal of Space-based and Situated Computing, 2015, 5(3): 178–186. [doi: 10.1504/IJSSC.2015.070949]

[9] Weil SA, Brandt SA, Miller EL, Long DDE, Maltzahn C. Ceph: A scalable, high-performance distributed file system. In: Proc. of the 7th Symp. on Operating Systems Design and Implementation. Seattle: USENIX Association, 2006. 307–320.

[10] Weil SA, Brandt SA, Miller EL, Maltzahn C. CRUSH: Controlled, scalable, decentralized placement of replicated data. In: Proc. of the 2006 ACM/IEEE Conf. on Supercomputing. Tampa: IEEE, 2006. 31. [doi: 10.1109/SC.2006.19]

[11] Robinson YH, Balaji S, Julie EG. PSOBLAP: Particle swarm optimization-based bandwidth and link availability prediction algorithm for multipath routing in mobile ad hoc networks. Wireless Personal Communications, 2019, 106(4): 2261–2289. [doi: 10.1007/s11277-018-5941-9]

[12] Luo XH, Shu JW. Summary of research for erasure code in storage system. Journal of Computer Research and Development, 2012, 49(1): 1–11 (in Chinese with English abstract).

[13] Lakshman A, Malik P. Cassandra: A decentralized structured storage system. ACM SIGOPS Operating Systems Review, 2010, 44(2): 35–40. [doi: 10.1145/1773912.1773922]

[14] Agarwal S, Dunagan J, Jain N, Saroiu S, Wolman A, Bhogan H. Volley: Automated data placement for geo-distributed cloud services. In: Proc. of the 7th USENIX Conf. on Networked Systems Design and Implementation. San Jose: USENIX Association, 2010.

[15] Yuan D, Yang Y, Liu X, Chen JJ. A data placement strategy in scientific cloud workflows. Future Generation Computer Systems, 2010, 26(8): 1200–1214. [doi: 10.1016/j.future.2010.02.004]

[16] Zheng P, Cui LZ, Wang HY, Xu M. A data placement strategy for data-intensive applications in cloud. Chinese Journal of Computers, 2010, 33(8): 1472–1480 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2010.01472]

[17] Chen CA, Won M, Stoleru R, Xie GG. Resource allocation for energy efficient *k*-out-of-*n* system in mobile ad hoc networks. In: Proc. of the 22nd Int'l Conf. on Computer Communication and Networks. Nassau: IEEE, 2013. 1–9. [doi: 10.1109/ICCCN.2013.6614183]

[18] Hong B, Choi W. Optimal storage allocation for wireless cloud caching systems with a limited sum storage capacity. IEEE Trans. on Wireless Communications, 2016, 15(9): 6010–6021. [doi: 10.1109/TWC.2016.2577025]

[19] Shvachko K, Kuang HR, Radia S, Chansler R. The hadoop distributed file system. In: Proc. of the 26th IEEE Symp. on Mass Storage Systems and Technologies. Incline Village: IEEE, 2010. 1–10. [doi: 10.1109/MSST.2010.5496972]

[20] Rashmi KV, Shah NB, Gu DK, Kuang HR, Borthakur D, Ramchandran K. A solution to the network challenges of data recovery in erasure-coded distributed storage systems: A study on the Facebook warehouse cluster. In: Proc. of the 5th USENIX Conf. on Hot Topics in Storage and File Systems. San Jose: USENIX Association, 2013.

[21] Huang C, Simitci H, Xu YK, Ogus A, Calder B, Gopalan P, Li J, Yekhanin S. Erasure coding in Windows Azure storage. In: Proc. of the 2012 USENIX Conf. on Annual Technical Conf. Boston: USENIX Association, 2012. 15–26.

[22] Muralidhar S, Lloyd W, Roy S, Hill C, Lin E, Liu WW, Pan S, Shankar S, Sivakumar V, Tang LP, Kumar S. f4: Facebook's warm BLOB storage system. In: Proc. of the 11th USENIX Symp. on Operating Systems Design and Implementation (OSDI 14). Broomfield: USENIX Association, 2014. 383–398.

[23] Xia MY, Saxena M, Blaum M, Pease DA. A tale of two erasure codes in HDFS. In: Proc. of the 13th USENIX Conf. on File and Storage Technologies. Santa Clara: USENIX Association, 2015. 213–226.

[24] Reed IS, Solomon G. Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics, 1960, 8(2): 300–304. [doi: 10.1137/0108018]

[25] Zhang S, He P, Suto K, Yang P, Zhao L, Shen XM. Cooperative edge caching in user-centric clustered mobile networks. IEEE Trans. on Mobile Computing, 2018, 17(8): 1791–1805. [doi: 10.1109/TMC.2017.2780834]

[26] Khan A, Aftab F, Zhang ZS. UAPM: An urgency-aware packet management for disaster management using flying ad-hoc networks. China Communications, 2019, 16(11): 167–182. [doi: 10.23919/JCC.2019.11.014]

[27] Wu Y, Liu D, Chen XZ, Ren JT, Liu RP, Tan YJ, Zhang ZL. MobileRE: A replicas prioritized hybrid fault tolerance strategy for mobile distributed system. Journal of Systems Architecture, 2021, 118: 102217. [doi: 10.1016/J.SYSARC.2021.102217]

[28] Hong XY, Gerla M, Pei GY, Chiang CC. A group mobility model for ad hoc wireless networks. In: Proc. of the 2nd ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems. Seattle: ACM, 1999. 53–60. [doi: 10.1145/313237.313248]

[29] Hao S, Zhang HY, Song MK. An adaptive clustering strategy for MANET based on learning automata theory and stability control. Chinese Journal of Computers, 2018, 41(9): 2089–2105 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.02089]

[30] Benson T, Akella A, Maltz DA. Network traffic characteristics of data centers in the wild. In: Proc. of the 10th ACM SIGCOMM Conf. on Internet Measurement. Melbourne: ACM, 2010. 267–280. [doi: 10.1145/1879141.1879175]

[31] Greenberg A, Hamilton JR, Jain N, Kandula S, Kim C, Lahiri P, Maltz DA, Patel P, Sengupta S. VL2: A scalable and flexible data center network. In: Proc. of the 2009 ACM SIGCOMM Conf. on Data Communication. Barcelona: ACM, 2009. 51–62. [doi: 10.1145/1592568.1592576]

[32] Eswaradass A, Sun XH, Wu M. A neural network based predictive mechanism for available bandwidth. In: Proc. of 19th IEEE Int'l Parallel and Distributed Processing Symp. Denver: IEEE, 2005. [doi: 10.1109/IPDPS.2005.51]

[33] Cong X, Shuang K, Su S, Yang FC, Zi LL. SBDP: Bandwidth prediction mechanism towards server demands in P2P-VoD system. Peer-to-peer Networking and Applications, 2015, 8(3): 501–511. [doi: 10.1007/s12083-014-0273-3]

[34] Oyeleke OD, Thomas S, Idowu-Bismark O, Nzerem P, Muhammad I. Absorption, diffraction and free space path losses modeling for the terahertz band. Int'l Journal of Engineering and Manufacturing, 2020, (1): 54–65. [doi: 10.5815/ijem.2020.01.05]

附中文参考文献:

[12] 罗象宏, 舒继武. 存储系统中的纠删码研究综述. 计算机研究与发展, 2012, 49(1): 1–11.

[16] 郑湃, 崔立真, 王海洋, 徐猛. 云计算环境下面向数据密集型应用的数据布局策略与方法. 计算机学报, 2010, 33(8): 1472–1480. [doi: 10.3724/SP.J.1016.2010.01472]

[29] 郝圣, 张沪寅, 宋梦凯. 基于学习自动机理论与稳定性控制的自适应移动无线Ad Hoc网络分簇策略. 计算机学报, 2018, 41(9): 2089–2105. [doi: 10.11897/SP.J.1016.2018.02089]

伍代涛(1997－), 男, 硕士, 主要研究领域为分布式存储系统中的数据布局和容错技术.

吴宇(1995－), 女, 博士, 主要研究领域为纠删码, 分布式存储.

谭玉娟(1983－), 女, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为高性能计算机体系结构和云存储系统.

陈咸彰(1989－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为新型内存系统, 文件系统, 嵌入式系统, 软件和云计算.

刘铎(1980－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为计算机系统结构, 新型存储架构, 嵌入式系统, 软硬件协同优化.

乔磊(1982－), 男, 博士, 研究员, CCF 杰出会员, 主要研究领域为计算机体系结构.

魏鑫蕾(1998－), 女, 硕士, 主要研究领域为分布式存储系统中的容错技术.