

引用格式:姚广政,隋爱娜,韩鹏斌,刘思源.面向无人飞行器集群协同的分布式存储技术研究[J].中国传媒大学学报(自然科学版),2024,31(02):71-78.

文章编号:1673-4793(2024)02-0071-08

# 面向无人飞行器集群协同的分布式存储技术研究

姚广政,隋爱娜\*,韩鹏斌,刘思源

(中国传媒大学计算机与网络空间安全学院,北京100024)

**摘要:**近年来,随着无人飞行器集群在各个领域的广泛应用,集群上所搭载的算法也越来越复杂,这对数据存储系统提出了更高的要求。现有研究没有考虑无人飞行器集群运行环境的特殊性,无法适应战场上通信间歇弱连接和节点易战损的环境。因此本文基于分布式存储技术设计了一套面向无人飞行器集群协同的分布式存储方案,通过动态自适应存储策略选择存储节点,提升了存储速度和资源的利用效率;同时,本文设计了基于RS码的数据备份与恢复方案,采用客户端与存储节点的交互模式降低了数据备份对于通信资源的占用,并且能够在多节点同时战损时恢复损失数据,提升了数据存储系统的稳定性和可靠性。

**关键词:**无人飞行器集群;分布式存储;数据备份

中图分类号:TP311 文献标识码:A

## Research on distributed storage technology for collaborative unmanned aerial vehicle swarms

YAO Guangzheng, SUI Aina\*, HAN Pengbin, LIU Siyuan

(School of Computer and Cyber Sciences, Communication University of China, Beijing 100024, China)

**Abstract:** In recent years, with the wide application of UVA (Unmanned Aerial Vehicle) cluster in various fields, the algorithms installed on the cluster are becoming more and more complex, which puts forward higher requirements for data storage system. However, the particularity of the operating environment of UAV swarm is not considered in the existing research, which can not adapt to the environment of intermittent weak connection and easy damage of nodes in the battlefield. Therefore, in this paper, a storage scheme based on distributed technology for UAV swarm coordination was designed. Storage nodes were selected through dynamic adaptive storage strategies, which improved the storage speed and resource utilization efficiency of the storage system. Moreover, to reduce the occupation of communication resources by data backup, a data backup and recovery scheme based on RS (Reed-Solomon) code was designed, which realized the recovery of lost data when multiple nodes were damaged at the same time through the interaction between the client and storage nodes, improving the stability and reliability of the data storage system.

**Keywords:** UAV swarm; distributed storage; data backup

基金项目:全国重点实验室基金项目(WDZC2020601A10)

作者简介(\*为通讯作者):姚广政(1999-),男,硕士研究生,主要从事网络与系统安全研究。Email:1013049235@qq.com;隋爱娜(1972-),女,博士,教授,主要从事信息安全与区块链研究。Email:aina@cuc.edu.cn;韩鹏斌(1995-),男,博士研究生,主要从事可搜索加密、属性基加密与区块链研究。Email:cybercommunication@163.com;刘思源(2001-),女,本科生,主要从事网络与系统安全研究。Email:liusiyuan2323@163.com

## 1 引言

随着无人飞行器集群技术的不断进步,其在快速反应和执行精确任务方面的潜力已成为军事策略的重要组成部分<sup>[1]</sup>。无人飞行器集群在战术级别上的运用,特别是在高压且反应时间有限的战场环境中,对实时信息处理和数据流管理提出了前所未有的挑战。在这样的应用场景下,无人飞行器集群信息系统的计算和存储能力,以及网络通信的可靠性,对任务的成功至关重要。

尽管无人飞行器技术日益成熟,但个体平台的计算资源限制和受限的存储容量仍是明显的短板。在紧凑的作战进程中,要求无人飞行器能够快速处理临时储存大量数据,这种需求强调了传统单机无人飞行器系统的性能此时尤为关键。

为了解决上述问题,研究人员对无人飞行器集群信息系统的架构展开了研究。Mahmoud等<sup>[2]</sup>提出了一种融合无人机和云计算的平台,通过资源导向架构实现无人机资源服务的统一接口,简化了多无人机协作应用的开发和任务分配。Pinto等人<sup>[3]</sup>开发了一种结合雾计算和云计算的架构,优化了无人飞行器集群的数据处理,降低延迟并提升通信效率。臧义华等人<sup>[4]</sup>设计了一种轻量化的区块链架构,通过主链和任务链的双结构信息系统解决无人机集群的数据共享和存储问题,降低了共识算法的资源消耗。闫辉等人<sup>[5]</sup>提出了一种云端融合的无人机集群任务调度架构,通过分层混合通信模式和资源优化调度方法,增强了无人平台间的互操作性和集群的通信稳定性。甘明明<sup>[6]</sup>研究了无人机集群数据资源共享及同步更新算法,通过构建共享资源池和实施时间衰减的同步更新策略,有效管理集群数据资源并保障其时效性。

现有无人飞行器集群信息服务架构的优势和不足如下:

(1)这些架构充分考虑了无人飞行器单节点能力不足的问题,大多数架构的设计模式是基于多节点协同或分布式,利用多个节点构成一个强大的信息服务系统,充分利用多个节点的计算和存储能力完成信息处理与服务。

(2)这些研究中很少有涉及数据的备份。即使在基于区块链的方案中,区块链上数据具有备份属性,未上链数据分片存在网盘也有一定的数据冗余,但仍然达不到战场环境中持续的、多节点的战损条件下对于数据备份的需求。

(3)现有架构未考虑节点间可能出现的间歇性弱连接问题,这在实际应用中可能导致信息服务的不稳定甚至数据丢失。

本文针对以上问题提出一种面向无人飞行器集群协同的分布式存储系统。该系统采用分布式架构,利用无人飞行器集群内部的存储协同能力建立数据冗余,以此来提升集群的存储性能。为了应对间歇弱连接的通信挑战,本文设计一种动态自适应的节点选择策略,根据节点间的通信状态,动态地选取数据存储节点,提高数据存储效率。为了应对节点战损问题,本文提出基于RS(Reed-Solomon)码的数据备份与恢复算法,支持在多节点同时损失时的数据恢复,并且可以对节点整体的数据进行恢复或指定数据恢复。

## 2 分布式存储技术

### 2.1 分布式存储架构

分布式存储系统是一种由多台计算机节点组成的系统,它们共同合作以存储和管理数据。相较于传统的集中式存储系统,分布式存储系统具有更高的可扩展性、可靠性和容错性<sup>[7]</sup>。在分布式存储系统中,数据通常会被分割成小块,并在多个存储节点之间进行分布存储。为了保证数据的冗余备份,分布式存储系统通常会在多个节点之间复制数据,以提高系统的容错性,即使某些节点发生故障,系统仍然能够继续提供服务。分布式存储系统需要确保数据在不同节点之间的一致性<sup>[8]</sup>。为了实现这一点,系统通常会采用一致性协议或者分布式事务来保证数据的正确性和一致性。

分布式存储架构可以分为有中心的分布式存储和无中心的分布式存储两大类模型。

有中心的分布式存储模型通常依赖于一个或多个中心节点来协调数据的存储、访问和管理<sup>[9]</sup>。这种模型数据管理简单,容易实现数据的一致性和完整性控制,容错和恢复机制通常集中设计,但中心节点可能成为性能瓶颈或发生单点故障。

无中心的分布式存储模型没有中心控制节点,通过网络中的节点平等参与数据存储、管理和分发。这种模型高度可扩展,无单点故障,能够有效地分散网络流量和存储压力<sup>[10]</sup>。但实现数据一致性和完整性控制更为复杂,可能需要复杂的算法来管理网络动态性和节点变化。

不同的系统架构具有不同的适用场景。在战场环境中,无人飞行器节点面临着战损的威胁,这就要

求数据存储系统具备高度的鲁棒性和灵活性。采用无中心的分布式存储架构,各节点均等参与工作,从而可以避免中心节点因战损导致的单点故障问题,显著增强了系统的稳定性和可靠性。

## 2.2 数据备份策略

数据备份策略是为了在数据丢失或节点损坏时能够迅速恢复原始数据而制定的一系列预防措施和方法。它包括数据的复制、存储和恢复计划,旨在确保数据的完整性、可用性和一致性。有效的数据备份策略通常包括以下几个关键方面:

(1)备份频率和时间点。根据数据的重要性和更改频率决定备份的频率和具体时间。

(2)备份类型。备份类型包括全量备份、增量备份和差异备份。全量备份是指复制全部数据,算法最简单,但需要的存储空间最多,耗时最长。增量备份是仅备份自上次备份以来发生变化的数据,减少了备份所需的时间和存储空间,但恢复数据时需要更多的时间。差异备份是备份自上次全备份以来所有变化的数据,恢复速度比增量备份快,但所需存储空间随时间增长而增加<sup>[11]</sup>。

(3)备份存储介质和位置。数据备份可以存储在本地磁盘、外部存储设备或云存储服务器上。为了提高数据安全,应将备份数据存储于物理位置分离的地方<sup>[12]</sup>。

在设计无人飞行器集群存储数据的备份策略时,关键是识别数据的访问频率和重要性,以此来决定最合适的备份方法。即时备份策略能够确保几乎零时间点的丢失,适合于战场环境中数据变化频繁且对数据完整性要求极高的应用场景。

## 2.3 RS纠删码

RS纠删码是一种强大的纠错码技术,广泛应用于数据传输和存储系统中,以增强数据的可靠性和容错能力<sup>[13]</sup>。RS纠删码的核心优势在于其灵活性和强大的纠错能力,特别是它支持任意数量的原始数据块 $k$ 和任意数量的备份数据块 $m$ 。这意味着在构建存储或通信系统时,可以根据需要调整 $k$ 和 $m$ 的值,以达到所需的容错水平。

RS纠删码根据 $k$ 个原始数据块进行数学变换生成 $m$ 个额外的冗余数据块。这些冗余块与原始数据块一起存储或传输。如果在传输或存储过程中发生数据丢失或损坏,只要系统中仍然存在至少 $k$ 个任意的数据块,就能够完全重构出所有原始数据<sup>[14]</sup>。在战

场环境的无人飞行器集群中,确保数据能被可靠地还原和高效地存储是至关重要的。RS纠删码可以稳定地恢复数据,同时不占用大量额外存储空间,是无人飞行器集群数据备份算法的理想选择。

## 3 分布式存储系统关键技术与设计

### 3.1 分布式存储模型

本文提出了一种面向无人飞行器集群协同的分布式数据存储模型,旨在应对无人飞行器在执行复杂任务时遇到的数据存储和管理挑战。模型的结构如图1所示。

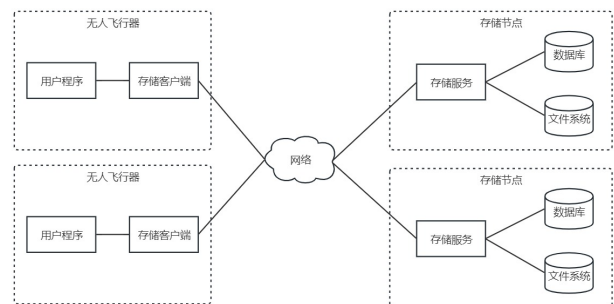


图1 分布式数据存储模型

本模型的设计考虑到无人飞行器集群在执行任务时,对数据存储的高度分布性、实时性和可靠性的需求。模型中,无人飞行器不仅作为数据的生成者,也充当数据的消费者,而存储节点则负责数据的存储和管理。存储客户端代理作为无人飞行器与存储节点之间的桥梁,负责处理数据读写请求,并通过网络与存储服务进行通信。

存储节点是本模型的关键组成部分,负责数据的持久化存储和管理。每个存储节点上运行着存储服务,使得存储节点具有数据处理的能力,并且管理着节点上的数据库和文件系统。这些数据库和文件系统用于存储来自无人飞行器的文件数据和关系数据。存储客户端代理是无人飞行器与存储节点之间的中介,它封装了数据读写的复杂性,提供了一个简化的接口供无人飞行器使用。客户端代理负责构建数据请求,通过网络与存储节点上的存储服务通信,并处理数据传输的所有细节。

该分布式数据存储模型为无人飞行器集群提供了一种高效、可靠的数据管理方案。模型的设计充分考虑了无人飞行器的动态性和分布式环境的特点,确保了数据存储的弹性和扩展性。

### 3.2 动态自适应存储策略

本文设计了一套在存储数据时动态自适应选择



存储节点的策略。该策略的实现方式如图2所示。

动态自适应的存储策略要求存储客户端维护一个当前所有存储节点的列表,列表中存储节点的节点号和通信状态,并不断更新客户端与这些节点的通信状态。初始状态下使用当前指针指向该列表的第一个元素,当有数据块需要进行存储时,检查指针指向的节点的通信状态。若不能正常通信,则指针后移,跳过该存储节点;若能正常通信,则数据块发送给列表中指针指向的存储节点,并指针后移。当指针达到列表尾部时再次执行指针后移操作,则指针指向列表的第一个元素。

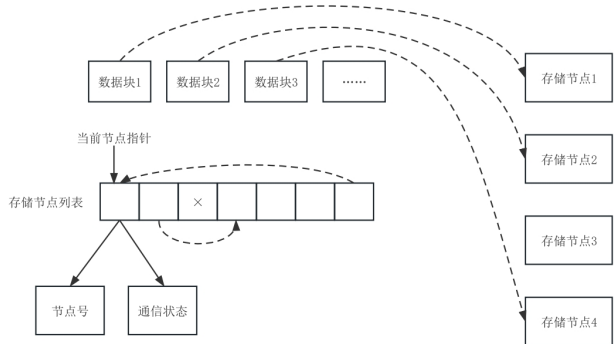


图2 动态自适应存储策略

通过使用动态自适应策略,客户端可以在通信状态不良的条件下,将不同数据块尽可能分散到不同的存储节点上,保证集群资源的高效利用。

### 3.3 数据存储与查询算法

在无人飞行器集群协同执行任务的过程中,数据交互是实现高效协同工作的关键。这些数据不仅涉及飞行器之间的实时数据交换,还可能包括与地面控制站的数据通信。无人飞行器集群上需要存储的数据主要包括飞行状态数据、传感器数据、任务指令数据以及图像数据等。根据对这些数据的分析,本文将无人飞行器集群上需要存储的数据类型分成两类,分别是文件数据和关系数据。文件数据包括传感器数据、视频图像等。这类数据的标志性特征包括较大的单个文件尺寸、需求高吞吐量的数据写入以及顺序访问模式,非常适合以文件的形式进行存储,通常直接存储于操作系统的文件系统上。关系数据如飞行日志、飞行状态信息等。这些数据通常有固定的结构,存储后需要有支持复杂查询操作的能力,通常以数据库形式进行存储。

为了存储这两类数据,本文设计了一套基于分布

式存储和动态自适应策略的数据存储模型,该模型支持对文件数据和关系数据进行高效存储和查询。模型主要有两个重要组成部分,分别是存储客户端和存储节点。客户端负责对用户提供数据存储和查询接口,并合理利用多个存储节点完成数据的存储和查询。存储节点负责存储系统运行所需的各类信息。

文件存储算法的具体流程如图3所示,详细说明如下:

- (1)客户端收到文件存储的请求时,会按照分块大小将待存储文件分块;
- (2)向存储节点发出文件存储请求,请求内容包括文件名称和文件分块数和其他文件元数据;
- (3)存储节点收到请求后,为该文件生成一个唯一标识,并将标识和其他信息一同记录到该节点的数据库中,为客户端返回唯一标识;
- (4)客户端收到文件唯一标识后,将文件唯一标识、分块号以及数据块加入待备份数据队列,并按照动态自适应存储策略将这些数据发送到相应的存储节点上;
- (5)存储节点收到请求后将文件片段存储文件系统,并生成文件路径,将文件标识、分块号和文件路径信息存入数据库。

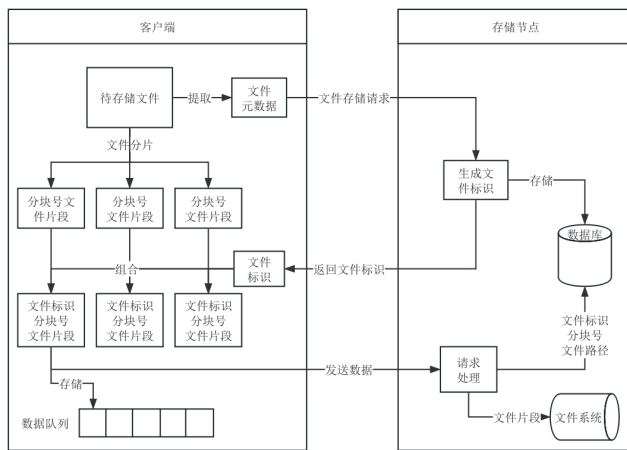


图3 文件存储算法流程图

客户端收到了文件数据的读取请求时,客户端通过用户指定的文件唯一标识,向所有存储节点发出请求,各存储节点返回该文件的各个分块号和数据块,客户端收到文件片段后对这些数据块进行拼接得到原始文件。若某些节点无法通信导致数据块缺失,可以通过数据恢复程序对缺失数据块进行恢复。

客户端收到关系数据存储的请求时,可直接将关系数据按照动态自适应存储策略发送到指定的存储

节点,由存储节点返回该数据的唯一标识,并将该标识和存储的关系数据一同保存在待备份数据队列。

在关系数据查询的过程中,由于涉及分布式数据库事务的复杂性,数据查询设计必须考虑事务的具体细节,因此本章不讨论关系数据的查询流程。

### 3.4 数据备份算法

无人飞行器集群所需存储的数据主要有关系数据和文件数据。数据备份算法为了高效处理和存储,通常要求数据以多块等长的二进制数据块形式存在。因此,数据库和文件系统中的数据需要经过特定的转换过程,才能满足备份算法的需求。

数据预处理的主要步骤如下:

(1)数据提取。数据提取是指将不同类型的数据从数据源中提取出来,重点是保留数据的完整性和准确性,确保所有必要信息都被成功提取。

(2)数据序列化。将提取出的关系数据序列化成为一种中间格式,以统一不同数据源中表示。对于文件数据,虽然文件内容已经以二进制形式存在,但文件还包含一些元数据,如文件的修改时间、照片的EXFI信息等,需要对其进行拼接,需要此步骤主要确保数据格式的一致性和后续处理的兼容性。

(3)数据分块。将序列化的关系数据和文件数据分割成多个等长的数据块。这一步骤中,关键是选择合适的数据块大小,并对不满足分块大小的数据进行适当的填充,以确保每个数据块的长度一致。

(4)二进制编码。将所有数据块转换为二进制格式。这一转换过程不仅优化了数据的存储和传输效率,也是执行数据备份算法前的必要步骤。

(5)存储。将二进制编码后的数据块存放在数据队列中。

经过数据预处理步骤后,文件数据和飞行数据都被分割成若干个长度相等的二进制块,再利用备份数据生成器即可生成数据备份。

数据备份器由两部分组成:编码算法和编码参数。通过调整编码参数,针对不同的输入块可以得到不同的备份数据。备份数据的生成和存储过程如图4所示,详细说明如下:

(1)备份程序通过预定义或动态变更的方式确定了当前编码的原始块数量 $k$ 和备份数据块数量 $m$ ,并通过不同的编码参数构造出 $m$ 个不同的数据备份器;

(2)数据预处理程序不断地向数据队列中写入待备份的数据块。当数据队列中的数据块数量达到 $k$

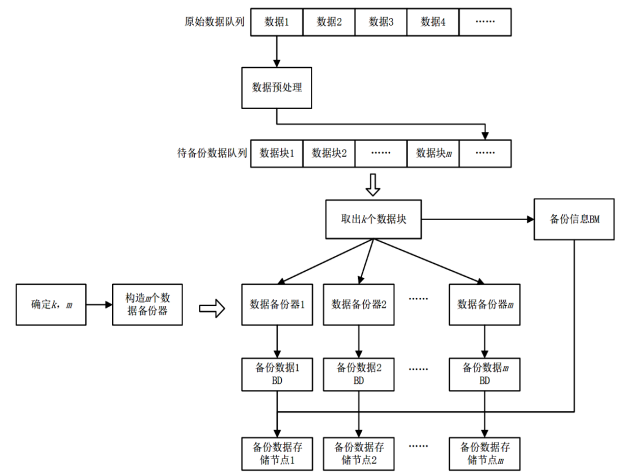


图4 数据备份生成过程

时,备份程序从原始数据队列中读取 $k$ 个块,分别输入到不同的数据备份器中,得到 $m$ 个不同的数据备份;

(3)根据取出的 $k$ 个数据块生成备份信息,备份信息包括这 $k$ 个数据块的唯一文件标识、分块号和存储位置;

(4)将 $m$ 个备份数据和备份信息分别发送到不同的存储节点进行存储。

通过上述方法,可以将原始数据队列中的数据生成备份数据和备份信息存储到数据备份节点的数据库和文件系统中。在此算法中,备份数据在客户端进行生成,网络中传输的仅有 $m$ 份备份数据,相比于在备份数据存储节点上生成数据备份的方案,可以极大程度上降低集群网络的通信压力。

### 3.5 数据恢复算法

在分布式存储系统中,节点的战损可能导致重要数据的丢失。为确保系统的鲁棒性和数据的完整性,必须迅速恢复受损节点上的数据。本文设计的分布式存储系统中有两种情况进行数据恢复,分别是存储节点战损和存储节点通信异常。

节点战损时,节点数据完全损失,需要对战损节点所存储的数据进行完整的恢复,存储在新节点中,这个节点将承担数据恢复后的数据服务任务,取代战损节点作为新的存储节点。数据恢复的详细过程如下:

(1)选择存储节点。在系统中选择一个可靠性较强的健康节点作为恢复数据的目标节点。

(2)初始化数据恢复。根据先前设定的数据备份器的参数初始化数据恢复器。数据恢复器将使用与数据备份相同的编码算法,但在恢复模式下运行。

(3)数据收集。从系统中的其他健康节点收集恢

复受损节点所需的数据。这包括原始数据块和校验块。根据RS纠删码的特性,只需任意 $k$ 个块就足以恢复所有原始数据。

(4)数据恢复。将收集到的数据块输入到数据恢复器中,结合收集到的数据块和校验信息,计算出丢失的原始数据块,再将原始数据块拼接、解码生成原始数据,并写入数据库和文件系统。

(5)系统状态更新。在数据恢复完成后,更新系统的元数据,标记恢复操作的完成,以及新的数据存储位置。

如果节点没有战损,但节点由于网络通信问题导致节点暂时无法访问,在这种情况下只需要临时恢复出用户当前所需要的数据。文件块的数据恢复过程与节点战损的数据恢复过程相似,虽然不需要选择存储节点和系统状态更新的步骤,但文件块恢复时所需的数据收集过程较为复杂,需要通过数据备份节点上的信息来定位恢复文件块所需的数据。本文设计了文件块数据收集协议来进行数据收集的过程,协议的具体流程如图5所示,具体说明如下:

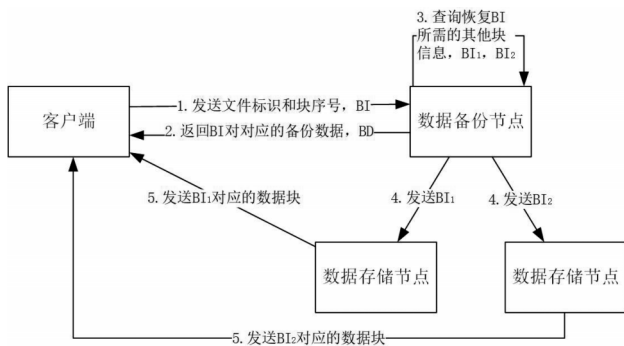


图5 文件块数据收集协议

(1)客户端向存储备份数据的节点发送需要恢复的文件块的文件唯一标识和块序号(BI);

(2)数据备份节点根据BI查询该块的备份数据(BD),数据备份节点为客户端返回BD;

(3)数据备份节点根据BI查询恢复该文件块所需的其他数据块的信息(BM),该信息包含文件唯一标识、块序号和存储位置;

(4)数据备份节点根据BM中的存储位置向指定存储节点发出请求,请求包含BM中的文件唯一标识、块序号;

(5)数据存储节点收到请求后向客户端发送BM中所需的文件数据块。

通过上述流程,可以根据情况对节点战损时丢失的数据或通信异常时缺失的数据进行恢复,保证

间歇弱连接通信和节点战损环境下数据的完整性和可用性。

## 4 实验结果与分析

### 4.1 实验环境与配置

本文所实现的系统测试环境如表1所示,实验将不同节点部署在同一台主机上,通过不同进程监听不同端口来模拟多个无人飞行器节点。

表1 系统软硬件环境

名称	版本型号
操作系统	Windows11专业版21H2
CPU	AMD Ryzen 7 5800H
内存	16GB 3200MHz
Python	3.10.2
Flask	2.2.3
Mysql	5.7.26
Pycharm	2022.2.3(Professional Edition)

### 4.2 数据存储测试与分析

飞行数据的读写功能采用了ApiPost工具进行测试,数据包含14个字段。文件的读写功能采用Python的requests库编程进行测试,测试文件大小为492MB。测试结果如表2和表3所示。

无人飞行器集群每个节点一般每秒钟产生一条或两条飞行数据,本系统可以支持140至280个节点使用,集群中通常仅有特定节点执行协同算法时需要数据进行查询,本系统完全可以满足现有集群飞行数据的读写需求。

对于文件数据读写功能,网络带宽的影响对数据读写的响应速度影响很大。文件在进行读写时由于可以进行文件分块,可同时在多个存储节点进行读写,对于网络带宽的利用率较高,数据的传输速度较快,可以支撑集群的文件读写需求。

### 4.3 数据备份测试与分析

本小节对系统的数据恢复功能进行了测试,测试包含了飞行数据和文件数据的数据恢复时间。

飞行数据的恢复测试,实验在存储节点中分别存入1000、3000、5000和10000条数据进行测试,记录了系统在不同飞行数据量下进行数据恢复所需要的时间以及恢复数据的错误率,具体的实验结果如表4所示。

对于文件数据的恢复测试,实验在存储节点中分别存入100MB、300MB、500MB和1000MB的文件数



据进行测试,记录了系统在不同文件数据量下进行数据恢复所需要的时间以及恢复数据的错误率,具体的实验结果如表5所示。

由测试结果可知,当一个存有1000条飞行数据的节点发生战损时,恢复其数据仅需4.27秒,具有很

快的恢复速度;当数据量增大到10000条数据时,数据恢复时间提升到了76.94秒,仍然具有较快的恢复速度;文件数据的恢复速度在不同数据量下维持一个较为稳定的状态,约为10MB/s,具有较快的恢复速度。

表2 飞行数据读写测试

名称	请求频率	错误率	平均响应时间	最大响应时间	最小响应时间
飞行数据写入	280.4次/秒	0%	23.6ms	52ms	10ms
飞行数据查询	117.5次/秒	0%	83.99ms	150ms	20ms

表3 文件读写测试

名称	平均传输速率	错误率	平均响应时间	最大响应时间	最小响应时间
文件写入	32.43MB/s	0%	15.18s	18.99s	13.57s
文件读取	15.31MB/s	0%	32.15s	37.47s	27.92s

表4 飞行数据恢复时间测试

飞行数据损失量	错误率	总恢复时间	平均单条数据恢复时间
1000条	0%	4.27s	4.27ms
3000条	0%	13.78s	4.59ms
5000条	0%	28.67	5.73ms
10000条	0%	76.94	7.69ms

表5 文件数据恢复时间测试

文件数据损失量	错误率	总恢复时间	平均数据恢复速度
100MB	0%	10.41s	9.61MB/s
300MB	0%	29.07s	10.32MB/s
500MB	0%	47.49s	10.54MB/s
1000MB	0%	92.76s	10.78MB/s

## 5 结论

本文针对无人飞行器集群在战场环境中面临的数据丢失和数据访问难题,提出了一套创新的协同分布式存储系统架构。该架构融合了客户端代理和动态自适应存储策略,旨在优化战场环境中的数据存储方式。通过深入分析当前信息服务技术的发展现状并结合分布式存储系统的特性,本文为复杂多变的战场环境提供了有效的数据存储解决方案。针对战场环境特有的通信间歇弱连接和节点易损问题,本文还研究并提出了一套数据备份与恢复方案。该方案通过有效的客户端与存储节点间交互模式,显著降低了数据备份过程中对通信资源的需求。利用RS纠删码技术,该方案不仅减少了节点损失下的数据丢失风险,而且支持对特定文件片段的数据恢复,增强了系统在间歇性弱连接状态下的通信稳定性。实验结果表明,本文提出的分布式存储系统架构和数据备份与恢复方案,不仅满足无人飞行器集群在战场环境中的数据存储需求,还具

有良好的数据恢复能力和数据恢复速度。

综上所述,本文为战场环境下的无人飞行器集群提供了一种高效、稳定的数据存储和保障方案,为未来相关领域的研究和应用奠定了基础。

## 参考文献(References):

- [1] Gargalakos M. The role of unmanned aerial vehicles in military communications: application scenarios, current trends, and beyond[J]. The Journal of Defense Modeling and Simulation, 2021: 15485129211031668.
- [2] Mahmoud S, Mohamed N. Collaborative uavs UAVs cloud[C]// Proceeding of 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014: 365-373.
- [3] Pinto M F, Marcato A L M, Melo A G, et al. A framework for analyzing fog-cloud computing cooperation applied to information processing of UAVs[J]. Wireless Communications and Mobile Computing, 2019, 2019: 7497924.
- [4] 臧义华, 李小娟. 基于区块链的无人集群作战信息共享架构[J]. 指挥控制与仿真, 2020, 42(4): 19-22.

- [5] 闫辉. 面向云端融合的无人集群信息处理任务调度研究[D]. 长沙: 中国人民解放军国防科技大学, 2022.
- [6] 甘明明. 无人机集群控制系统及数据资源共享技术研究[D]. 广州: 华南理工大学, 2019.
- [7] Saadoon M, Hamid S H A, Sofian H, et al. Fault tolerance in big data storage and processing systems: a review on challenges and solutions[J]. *Ain Shams Engineering Journal*, 2022, 13(2): 101538.
- [8] Liang W, Fan Y, Li K C, et al. Secure data storage and recovery in industrial blockchain network environments[J]. *IEEE Transactions on Industrial Informatics*, 2020, 16(10): 6543-6552.
- [9] Zeebaree S R M, Shukur H M, Haji L M, et al. Characteristics and analysis of Hadoop distributed systems[J]. *Technology Reports of Kansai University*, 2020, 62(4): 1555-1564.
- [10] 张晓丽, 杨家海, 孙晓晴, 等. 分布式云的研究进展综述[J]. *软件学报*, 2018, 29(7): 2116-2132.
- [11] 丁丽媛. 基于数据生命周期的金融数据安全风险管理研究[J]. *信息安全研究*, 2018, 4(6): 548.
- [12] 李丹, 叶廷东. “异地多活”分布式存储系统设计和实现[J]. *计算机测量与控制*, 2020, 28(04): 211-216.
- [13] Lu M, Xiong C, Fan X. Erasure-code algorithm for distributed stream processing[C]// *Proceeding of 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, 2020: 1-5.
- [14] 罗继尧. 信息系统数据可用性恢复方法研究[J]. *计算机与数字工程*, 2018, 46(6): 1187-1192.

编辑: 赵志军