

# 基于超高速线程并行处理技术的民机 试飞数据预处理系统设计

王加熙, 刘涛, 冯灿, 高祥

(中国商飞民用飞机试飞中心 测试工程部, 上海 200231)

**摘要:** 试飞数据作为民机试飞过程中最重要的产物, 是表明符合性、验证设计合理性的依据; 传统模式下, 需要在航后卸载机载记录器中的原始数据, 通过 4~5 小时的预处理时间才能完成试飞数据的工程量转换, 耗时耗力, 严重影响试飞效率; 立足于实际试飞场景, 提出了在机载环境下实时进行试飞数据预处理的理念, 采用超高速线程并行处理技术, 集成了 IoTDB 时序数据库, 设计研制了机载数据在线完全预处理系统; 经实验测试, 系统实现了国产某大型客机 200 Mbps 带宽下近 80 000 个试飞数据的实时接收、预处理和数据库存储, 将以往的预处理周期压缩至 20 分钟, 支持了高强度、高密度试飞要求。

**关键词:** 试飞数据; 预处理; 实时; 超高速线程; 并行处理; IoTDB

## Design of Civil Aircraft Test Flight Data Pre-processing System Based on Ultra-High-Speed Thread Parallel Processing Technology

WANG Jiayi, LIU Tao, FENG Can, GAO Xiang

(Flight Test Center of COMAC, Shanghai 200231, China)

**Abstract:** As the most important product in civil aircraft flight test process, flight test data is the basis for showing compliance and verifying the rationality of design. In traditional modes, the original data in the on-board recorder need be unloaded after flight test, and the engineering conversion of flight test data is completed through 4~5 hours of preprocessing, which is time-consuming and labor-intensive, seriously affecting the efficiency of flight test. Based on actual flight test scenario, a real-time flight test data pre-processing concept in the airborne environment is proposed, an ultra-high-speed thread parallel processing technology is used to integrate the Internet of Thing database (IoTDB), and develop an online complete preprocessing system for airborne data. After testing, this system realizes the real-time reception, preprocessing and database storage of nearly 80 000 flight test data of a large domestic passenger aircraft with a bandwidth of 200 Mbps, compresses the previous preprocessing cycle to 20 minutes, and supports the requirements of high-intensity and high-density test flights.

**Keywords:** flight test data; preprocessing; real time; ultra-high-speed threads; parallel processing; IoTDB

## 0 引言

试飞是在真实的环境下对飞机的性能和功能进行测试的过程, 是民机投放市场前的必经之路<sup>[1]</sup>。用于试飞的飞机会加装各类传感器和测试设备, 对那些表征飞机关键系统状态的参数进行采集、处理和记录, 这些参数就是试飞数据<sup>[2]</sup>。试飞数据是民机试飞中最重要的产物, 其应用贯穿了整个试飞过程, 是用于判断试飞架次有效性、验证飞机设计合理性以及向局方表明符合性的重要依据。试飞数据经机载采集器采集编码后进行传输, 并利用记录设备对所有数据进行实时记录, 这类数据被称作原始数据。原始数据必须使用总线的 ICD (interface control documentation) 和传感器的校线进行解码后才能使用, 这个解码过程被称为工程量转换 (即将二进制编码的原始数据转换为物理

量), 也叫试飞数据预处理<sup>[3]</sup>。试飞数据预处理是进行数据分析、判断飞机状态、决定飞机是否再次飞行的基础, 对飞机飞行安全至关重要<sup>[4-5]</sup>。国内的主流试飞机构一般在飞行试验结束后对机载记录的原始数据进行卸载, 并通过相关软件实现过程量转换, 最后输出数据文件 (为 txt 或者 csv 的文本文件)<sup>[6]</sup>。随着先进航电总线在民机上的大量使用以及网络采集系统的普及, 试飞数据的数量越来越多、体量也越发庞大, 例如空客 380 的试飞参数数量就达到了十几万个, 而国产某大型客机的试飞数据也突破了 80 000 个, 单个架次的原始数据超过了 100 GB, 同时试飞数据的类型也更为复杂, 包括了总线、各类传感器以及音视频这些结构化、非结构化数据, 参数采样率也从 1 Hz 跨度到了 16 kHz, 除此之外, 为了保证试飞安全、提高试飞效率, 人们对试飞数据的时效性要求也越来越高<sup>[7-11]</sup>。传统的预处理

收稿日期: 2023-01-31; 修回日期: 2023-03-30。

作者简介: 王加熙 (1985-), 男, 江苏盐城人, 硕士, 高级工程师, 主要从事传感测量, 网络化试飞数据采集方向的研究。

引用格式: 王加熙, 刘涛, 冯灿, 等. 基于超高速线程并行处理技术的民机试飞数据预处理系统设计[J]. 计算机测量与控制, 2023, 31(7): 234-244.

理技术往往很难满足以上需求, 存在以下不足:

1) 试飞数据预处理周期过长。一般情况下, 试飞数据的预处理过程往往需要 4~5 个小时甚至更长的时间, 为试飞数据的分析和试验点有效性的快速判断带来了障碍, 不利于一天多架次的试飞;

2) 试飞数据的监控方式不灵活。试飞数据的监控一般采用被动接收的方式, 无法根据实际的试飞状态灵活变化;

3) 试飞数据的分析效率较为低下。试飞数据仍以文本文件方式输出, 不利于数据的检索、查询和关联分析, 这严重制约了试飞数据的分析效率。

国内外的试飞机构在试飞数据的预处理处理方面也在不断的探索和研究, 例如中国飞行试验研究院利用 Vx-Works 嵌入式系统实现了 110 Mbps 带宽下网络数据的接收和处理, 但处理能力仅能达到 8 000 个参数<sup>[12-14]</sup>, 空客所研制的预处理设备仍然需要在航后进行二次解析。

因此, 为了缩短试飞周期, 提高试飞效率, 降低试飞运营成本, 需要探索一种新的试飞数据预处理技术, 实现缩短试飞数据预处理周期、改变监控模式、提高分析效率的目的。

### 1 系统结构及原理

本文提出了在实时状态下进行试飞数据预处理的原理, 研制了一套试飞数据机载在线完全预处理系统, 功能是在机载环境下, 实现试飞数据的实时接收、工程量转换和数据库存储, 达到飞机落地即输出数据文件的目的。

系统包括机载数据处理服务器和实时预处理软件两大部分, 其中在线完全预处理软件包括配置信息生成软件、数据实时在线处理软件、监控调试软件、地面数据卸载软

件、机载高频监控软件等五大子软件。

主要业务流程如图 1 所示, 分为试飞任务配置、试飞任务运行和试飞任务结束以及数据卸载交付等四大环节:

1) 试飞任务配置。测试工程师根据实际试飞任务的具体信息, 在地面环境下使用配置信息生成软件以机载配置文件、总线 ICD 文件、参数组导出文件、监控转发文件为输入, 按照既定格式生成试验信息文件, 并导入机载数据处理服务器完成系统配置。其中, 机载配置文件为 XML (extensible markup language) 格式, 描述了机载测试系统的相关信息, 包括系统架构和采集器、交换机等设备信息; 总线 ICD 文件主要记录了 ARINC664 (aeronautical radio Inc.)、ARINC429 以及 IMB (inter model bus) 等机载总线的信息, 如参数的取位、类型和计算方法等, 是总线参数解析的关键; 参数组导出文件规定了每个数据输出文件所包含的参数名称, 可根据实际试飞任务需求动态更新; 监控转发文件说明了试飞数据的转发频率和转发顺序。

2) 试飞任务运行。在试飞任务开展过程中, 数据实时在线处理软件首先加载试验信息文件, 完成软件初始化, 并启动软件内部各模块, 按照设置自动开始进行数据在线处理任务; 监控调试软件对整个系统的软硬件状态进行监控、统计和显示; 数据实时在线处理软件以机载测试系统原始数据流为输入, 完成试飞数据的工程量转换、计算分析、分发和参数组导出以及 IoTDB 的时序数据库存储; 机载高频监视软件以实时数据驱动方式刷新监控画面完成振动、电源等高频数据的实时监控。

3) 试飞任务结束时 (飞机落地后至切断整机电源系统前), 数据实时在线处理软件对落盘存储的数据文件进行在

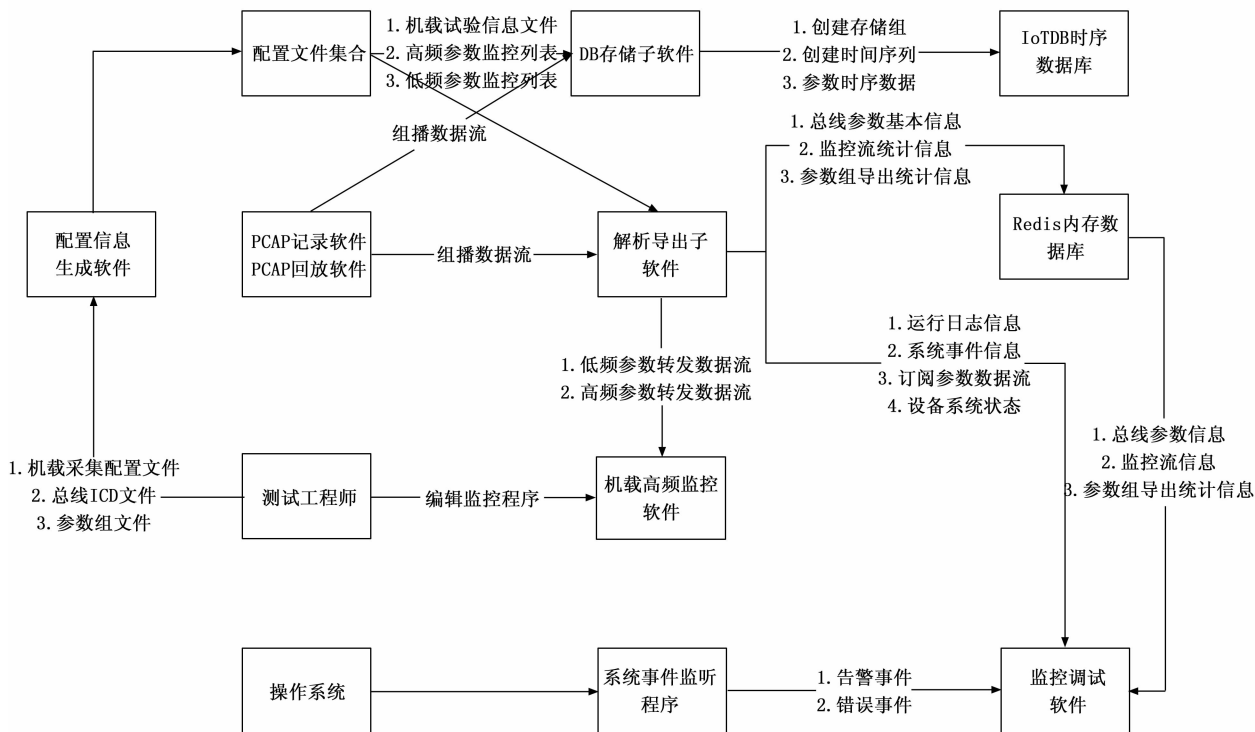


图 1 业务流程示意图

线压缩处理。

4) 数据卸载交付。技术人员从机载数据处理服务器取出数据磁盘,并挂载到数据分发设备(即计算机)后,运行在该设备上的地面数据卸载软件识别磁盘及数据文件,结合交付策略使用 FTP(file transfer protocol)等方式将数据分发至各目标终端。

## 2 机载数据处理服务器

机载实时处理服务器是实时预处理软件的运行载体,采用成熟的组件并通过加固方式实现,结构如图 2 所示。

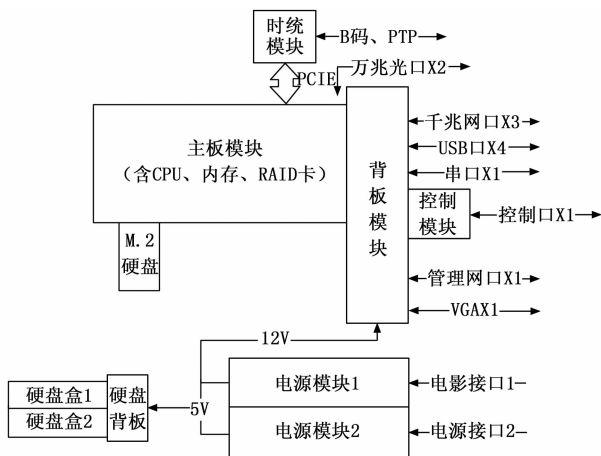


图 2 机载数据处理服务器组成框架

机载数据处理服务器各功能模块如下。

1) 主板模块:含 CPU(central processing unit)、板载内存、板载 RAID(redundant array of independent disks)卡及网卡等组件,其中 CPU 采用 Intel Xeon Gold 6148 处理器,内存为 DDR4,容量为 256 GB,使用板贴芯片替代传统槽加方式,加强主板的抗振动冲击能力;

2) 背板模块:由背板、对外/内连接器组成,主要为整个设备各模块间接口提供互联,包括千兆网口、USB、VGA 等,各模块间接口采用 CPCIE 连接器(如图 3(a)所示);

3) 存储模块:包含硬盘背板及 2 块模块化硬盘,通过楔块结构与存储模块支架紧固,可方便进行插拔(如图 3(b)所示);

4) 电源模块:将直流 28 V 电源转化为主板需要的直流 12 V 电源和硬盘所需的直流 5 V 电源;

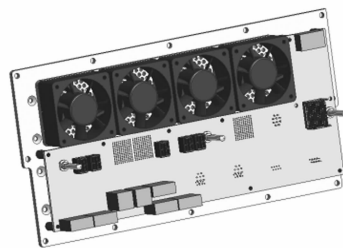
5) 时统模块:提供 PTP(precise time protocol)输入,IRIG-B 码输出功能,用于时间同步;

6) 控制模块:提供一路控制信号用于整个系统存储服务的启动和停止。

机载数据处理服务器采用加固模式,设计尺寸如图 3(c)所示。

## 3 实时预处理软件

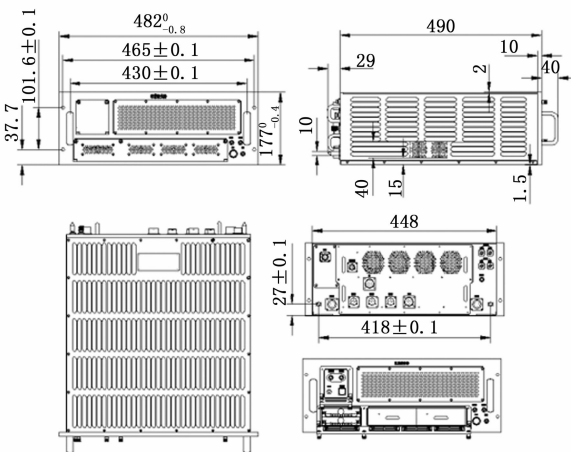
实时预处理软件是整个系统的核心,运行时通过组播方式实时接收机载测试系统的原始数据流,按照总线类型和具体参数配置,进行工程量转换、时间对齐,并将转换



(a) 背板模块



(b) 存储模块



(c) 服务器外形结构

图 3 模块及设计尺寸

后的数据进行两路并行输出,一路存储至 IoTDB 时序数据库,一路以文本文件存储于硬盘中。同时,软件集成了 Redis 内存数据库,主要负责数据的缓存和转发,用来集中管理预处理阶段各类统计信息,驱动监控调试软件完成状态监控。

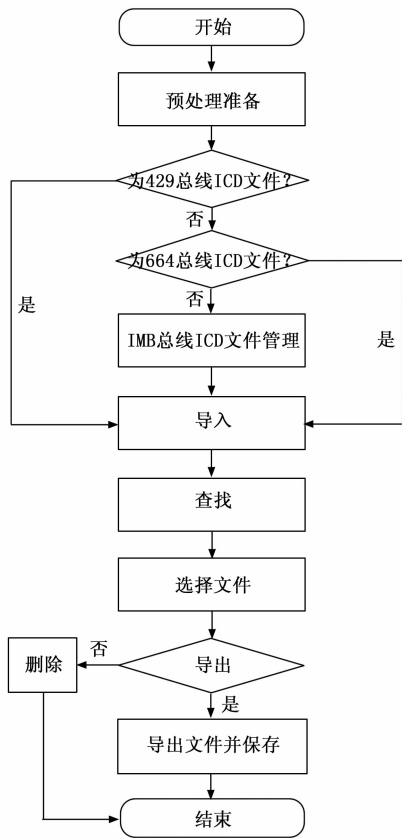
### 3.1 配置信息生成软件

配置信息生成软件工作在试飞准备阶段,主要功能是以机载配置文件、总线 ICD 文件、传感器校线文件、试飞参数分组文件为输入,生成试验信息文件。软件包括机载配置处理、总线 ICD 处理、校线文件处理、试验信息处理、日志管理、显示与查询等功能模块,处理流程和架构如图 4 所示。

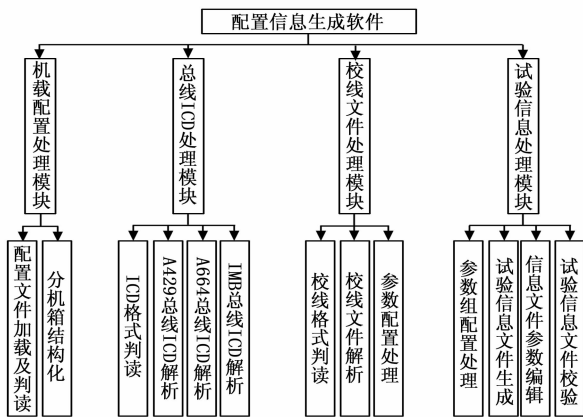
1) 机载配置处理模块主要负责机载配置文件(描述了机载采集系统,如机箱、板卡与参数之间详细结构关系的 XML 文件)的加载、编辑、解析和保存,并以机箱为单位对试飞参数进行结构化定义;

2) 总线 ICD 处理模块主要负责对 ARINC429、ARINC664、IMB 等总线的 ICD 信息进行判读、校验、加载和解析,并对参数的取位、数据类型、系数等基本属性进行编辑,生成参数对象;

3) 校线文件处理模块负责对传感器的校线文件进行加



(a) 配置信息生成流程



(b) 配置信息生成软件功能模块

图 4 处理流程和架构

载和解析, 校准类型主要定义以下四种形式:

(1) Polynomial/Inverse Polynomial 主要代表  $a * x + b$ 、

$$\frac{a}{x} + b \text{ 和 } a_0 + a_1x + a_2x^2;$$

(2) Bit Weighted 基于比特位进行描述, 即比特位与其系数形成映射关系, 当码值中比特为 1 时, 取其对应系数, 最终得到参数工程量数据, 并且按照最低位比特系数保留参数工程量精度;

(3) Lookup Table 基于点对之间插值, 输入多达 32 对输入和输出点, 输入数据点之间的值将被插值。

4) 试验信息处理模块主要完成试验信息文件的生成, 通过上述机载配置模块、ICD 处理模块、校线模块处理完成后得到的参数关联信息, 联合参数组配置文件内容, 生成可用于实时预处理软件可加载的配置文件。

### 3.2 数据实时在线处理软件

#### 3.2.1 总体设计

数据实时在线处理软件包含解析导出和 DB 存储两个子软件, 底层采用了多线程无锁队列基础框架, 集成了 Tinyxml、NanoMsg 消息队列以及 Redis 内存数据库等成熟组件<sup>[15]</sup>, 使用 C++11 标准编程语言实现。

解析导出子软件作为核心预处理解析软件, 包括配置解析、数据接收及处理、数据解析、数据缓存管理、监控转发、参数组导出存储、状态管理、实时数据转发等八个功能模块 (DB 存储子软件与解析导出子软件共用配置解析、数据接收、数据解析模块、数据缓存管理四个模块, 除此之外, 集成了 IoTDB 时序数据库, 用于实现试飞数据的数据库存储)。数据工程量转换、参数组导出、监控转发等核心业务流程如图 5 所示。

首先, 软件在启动时分别读取试验信息文件和监控列表文件, 使用配置解析模块对试验信息文件内各总线机箱及参数进行处理, 在内存中形成“总线-参数”的结构化数据结构, 构建所有参数解析所需属性信息, 并对监控列表中各参数进行位置映射;

其次, 数据接收及预处理模块根据配置对象中关于采集机箱 IP 端口的描述, 创建 UDP (user datagram protocol) 网络连接, 实时接收数据流, 调用包头处理线程按照协议格式判读包头类型及通道信息, 对数据进行分类转发至解析模块 (例如 NPD (network packet data) 包头类型为 0x35A1 代表 ARINC429 总线或模拟量数据, 按照机箱通道和 SFID (sub frame ID) 映射关系转发至子帧处理线程; 若是 0x35A8 则代表 ARINC664 和总线数据, 按 VLID (virtual link ID) 标识循环提取 Message 对应数据转发至总线解析线程; 如果是 0x35B0, 代表 IMB 总线数据, 按照 Label 分类提取 Message 数据并转发)<sup>[16-17]</sup>;

第三, 使用数据解析模块接收经过包头处理、数据区提取后的数据进行工程量转换, 其中, 针对 ARINC429 总线或模拟量数据, 需要进一步进行单帧提取处理, 并按照各总线规则解析, 每个参数工程量转换完成后立即转发;

第四, 使用数据缓存管理模块对接收所有参数工程量, 用链表数据结构方式存储至内存空间中 (每个参数对应一个链表结构), 每个参数对应的链表存储空间则根据导出完成状态自动删除历史工程量数据;

第五, 自动根据事先在试验信息文件中配置好的一个或多个导出参数组设置, 从参数链表中按照规定的导出设定进行数据采样提取, 并按照规定导出文件格式进行存盘, 在导出时, 高低频参数组导出分开处理, 当预处理完成后, 对所有参数组文件进行压缩处理;

第六, 针对需要高频转发、分析的参数, 软件在初始

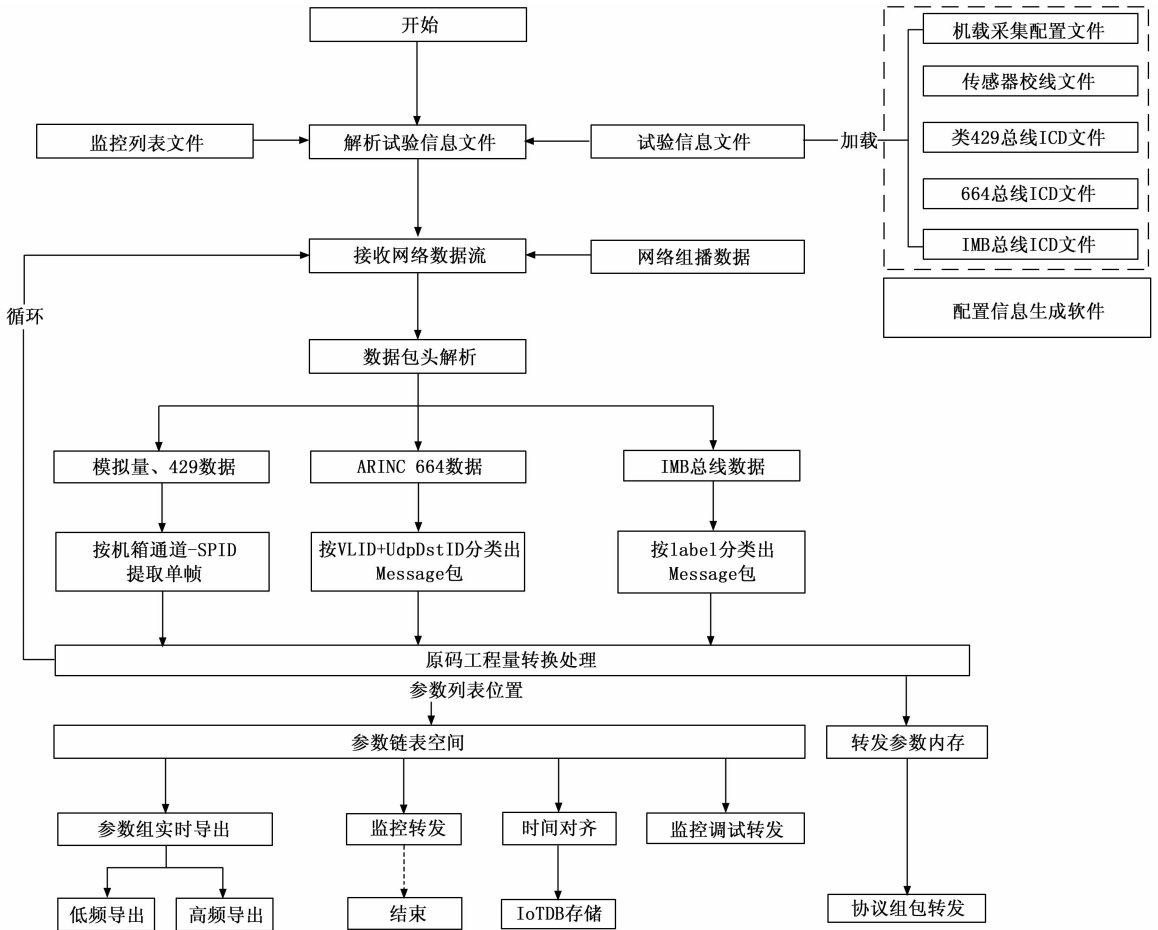


图 5 核心业务流程

化阶段按参数个数和频率申请固定大小内存空间，每次参数工程量解析完成后，改写对应位置数据，满足转发频率的数据长度要求后，按照协议格式进行组包，并使用 TCP/UDP 实时转发。

### 3.2.2 软件架构设计

数据实时在线处理软件采用超高速线程并行处理技术在实时状态下完成 NPD 或者 IENA 等数据流的实时处理。各个线程与功能模块的映射关系如表 1 所示。

表 1 功能模块与线程对应表

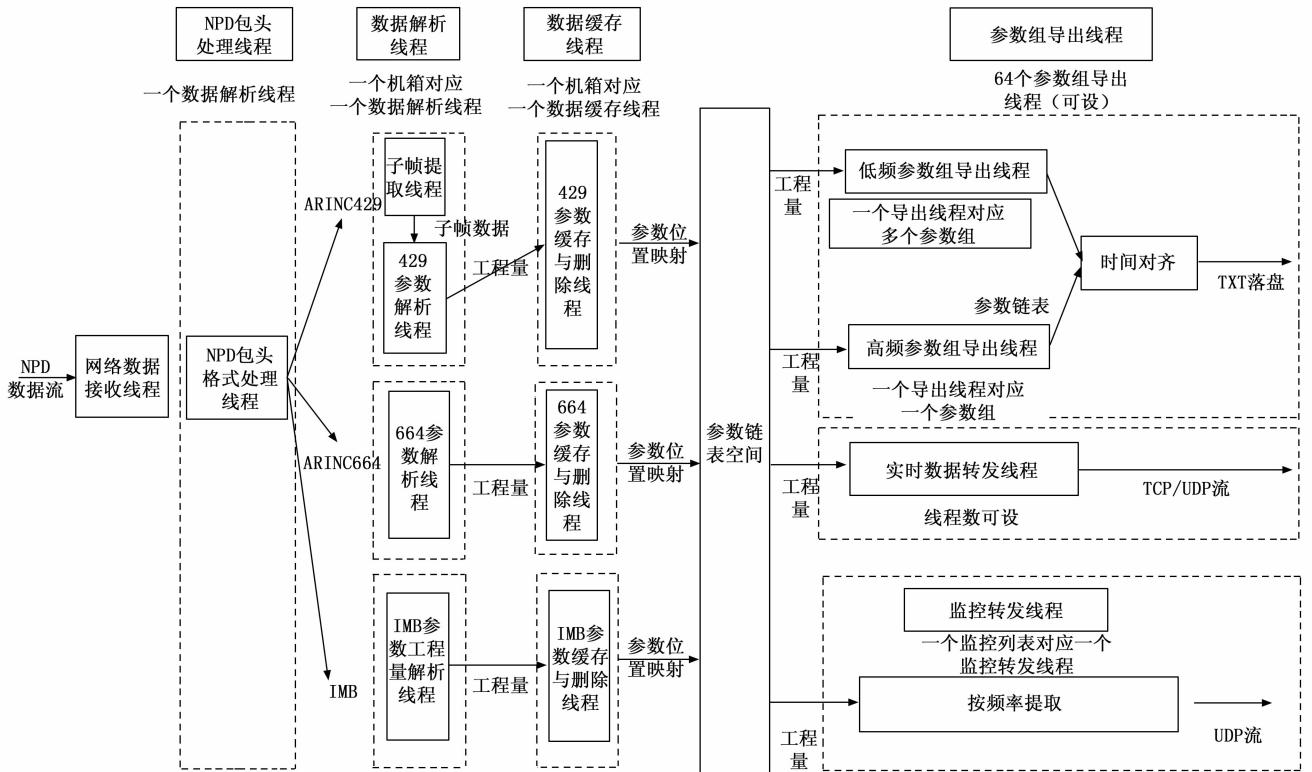
序号	功能模块	线程
1	配置解析模块	主调度线程
2	数据接收及预处理模块	网络接收线程
		NPD 包头处理线程
3	数据解析模块	子帧处理线程
		总线解析线程
4	数据缓存管理模块	数据缓存管理线程
5	数据存储模块	时序数据库存储线程
6	参数组导出模块	参数组并行导出线程
7	监控转发模块	监控转发线程
		实时转发线程
8	监控转发模块	监控参数转发线程
9	状态管理模块	状态管理线程

以 NPD 数据流为例，并行处理的示意图和时序图如图 6 所示。

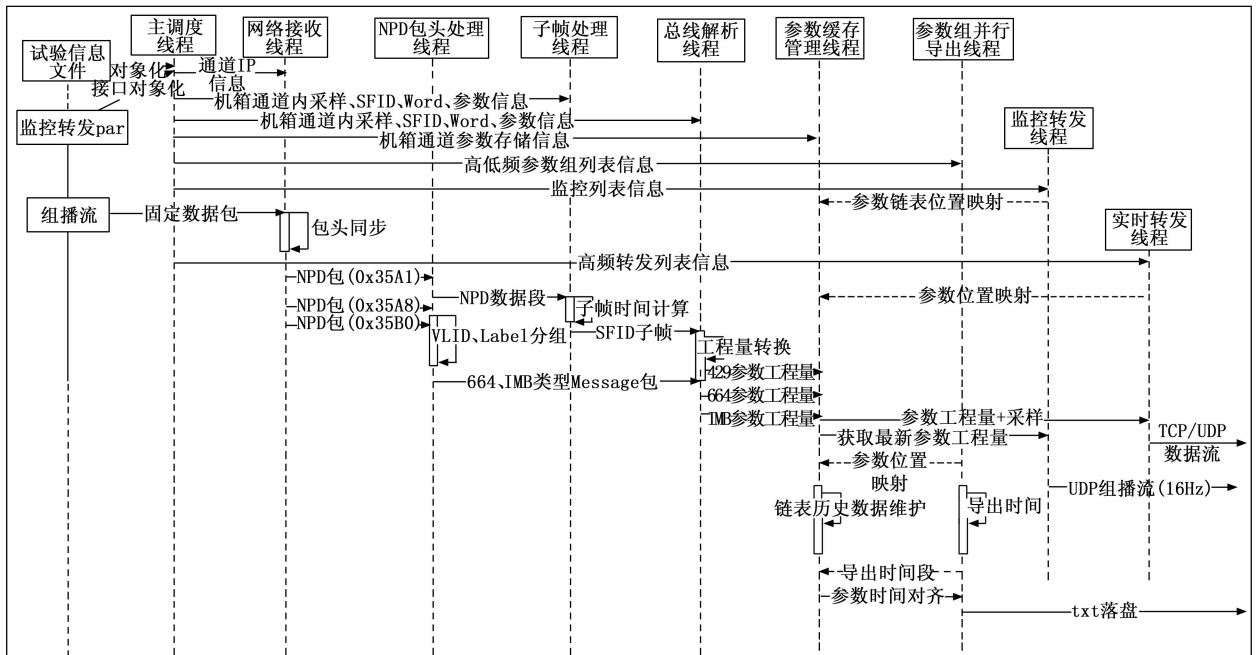
如图 6 所示，针对网络数据流，软件包括包头处理、数据解析、数据缓存以及数据存储等 4 大类线程，所有线程通过线程池统一管理和调度，运行原理如下：

- 1) 当软件启动时，主调度线程将试验信息文件中的信息进行对象化，并传递至各个子线程；
- 2) 网络接收线程根据组播地址和端口实时接收 NPD 数据包；
- 3) 包头处理线程对所有的数据包进行顺序处理，提取有效数据，同时建立数据类型、采集机箱和有效数据之间的映射关系；
- 4) 根据 3) 所创建的映射关系新建解析线程和数据缓存线程，创建原则是每一个机箱对应一个解析和缓存线程，多个机箱数据进行并行处理；
- 5) 解析后的数据被缓存至内存链表，等待订阅；
- 6) 根据试验信息文件中的参数组信息创建参数导出线程，每一个参数组为一个单独线程；
- 7) 参数导出线程从链表中提取对应数据并写入数据文件。

为了保证软件运行的稳定性，在线程以及资源的使用



(a) 数据实时在线处理软件并行处理流程



(b) 线程时序示意图

图 6 数据实时在线处理软件并行处理流程和线程时序示意图

上做以下优化:

首先, 对数据解析、参数导出、时间对齐等高负载线程, 与 CPU 核心数线程进行了绑定, 避免系统自动分配抢占计算资源;

其次, 参数缓存线程按照试验信息文件中的通道数量进行分配, 每个参数缓存线程管理和存储该机箱通道内的所有参数工程量, 各参数存储方式使用链表数据结构, 减少数据拷贝开销;

第三, 针对参数组导出存储线程, 区分高频和低频参数, 在数据解析前, 按照参数组导出配置, 根据高、低频参数数量进行线程分配, 提升参数导出效率。

### 3.2.3 配置解析模块

配置解析模块主要对配置信息生成软件产生输出的 XML 格式文件进行加载, 处理机箱与参数属性关系, 针对不同的总线, 定义不同结构体描述层级关系:

- 1) 针对 ARINC429 总线, 形成机箱通道-Message (SFID) - 参数-EuFunction 的层级结构;
- 2) 针对 ARINC664 总线, 形成机箱通道-Message (VLID) - 参数-数据类型的层级结构;
- 3) 针对 IMB 总线, 形成机箱通道-Label-参数-数据类型的层级结构。

### 3.2.4 数据接收及处理模块

数据接收及处理模块主要接收网络中各采集机箱发来的 NPD 或者 IENA 实时数据流 (每个机箱对应一个实时组播流接收线程), 按照总线类型和机箱信息对数据进行分组和数据包头处理以及子帧提取, 形成总线 Message, 该处理过程属于试飞数据进行工程量转换前的预处理过程, 经预处理后的各个总线 Message 转发至数据解析模块。

### 3.2.5 数据解析模块

数据解析模块是软件的核心模块, 承担各类总线数据的工程量转换。

#### 3.2.5.1 NPD-ARINC429<sup>[18]</sup>

- 1) 首先根据配置信息遍历 Message 下的参数列表, 对属于该消息的所有参数源码进行定位;
- 2) 对定位源码进行移位操作提取参数源码, 并根据参

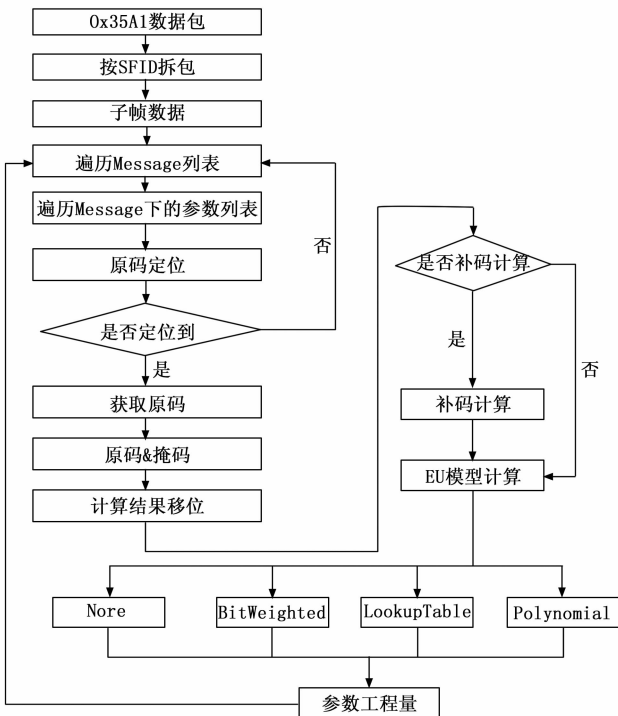


图 7 ARINC429 总线参数解析流程

数类型进行补码运算;

- 3) 根据参数类型调用 EU Function 结构对象完成工程量转换。

表 2 试验信息文件解析映射表

输入	输出类型	模块输出	数据类型
试验信息文件	ARINC 429	Eu Function 结构对象	枚举
		ARINC429 总线类型下 Message 的子参数结构对象_tagSubParam	Vector
		ARINC429 总线类型下 Message 对应的参数结构对象_tagParam_Message	Vector
	ARINC 664	ARINC429 总线类型下机箱属性所有 Message 结构对象_tagBoxMessage ARINC429 总线类型下机箱属性结构对象_tagBoxInfo	Map
		ARINC664 总线类型下的参数属性结构对象_tagAFDX_Msg	Map
		ARINC664 总线类型下的 Message 属性结构对象_tagParam_664 ARINC664 总线类型下机箱对应的属性结构对象_tagAFDX664	Map
	IMB	IMB 总线类型下的机箱属性结构对象_tagImbBox	Map
		IMB 总线类型下的机箱下 label 结构对象_tagImbLable	Vector
		IMB 总线类型下的参数属性结构对象_tagParam_IMB	Map
参数组	参数组	各参数组基本信息及参数列表结构对象化_tagGroupParamInfo	Vector
监控列表文件	参数组	各监控列表属性及参数列表结构对象化_tagMonitorParamInfo	Vector

#### 3.2.5.2 NPD-ARINC664

ARINC664 总线参数的解析过程如图 8 所示, 当 VLID + UdpDstID 一致时, 遍历参数列表, 用每个参数内的取位信息在 Message 对应的数据区中定位原码, 数据类型与分辨率系数计算后得到最终的工程量。

#### 3.2.5.3 NPD-IMB

IMB 总线参数的解析与 ARINC664 总线解析过程大致相同, 流程如图 9 所示。

#### 3.2.5.4 IENA

IENA 数据包的解析与 NPD 相似, 当从数据流中提取到数据包后, 结合试验信息文件中机箱与参数映射关系, 从参数列表中遍历各参数的具体位置进行原码定位, 并结合参数掩码进行取位, 通过参数校准类型进行工程量转换, 得到最终物理量。

#### 3.2.6 数据缓存管理模块

数据缓存管理模块以二维数组指针形式维护所有参数

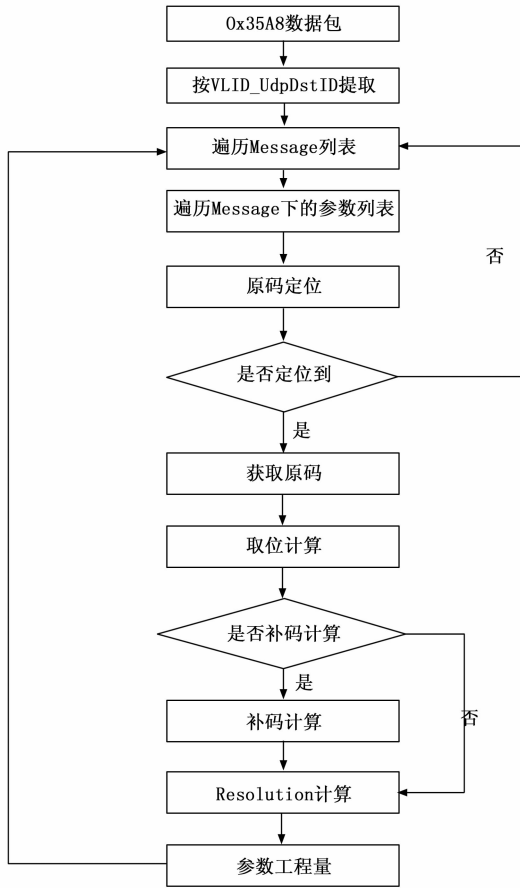


图 8 ARINC664 总此案参数解析流程

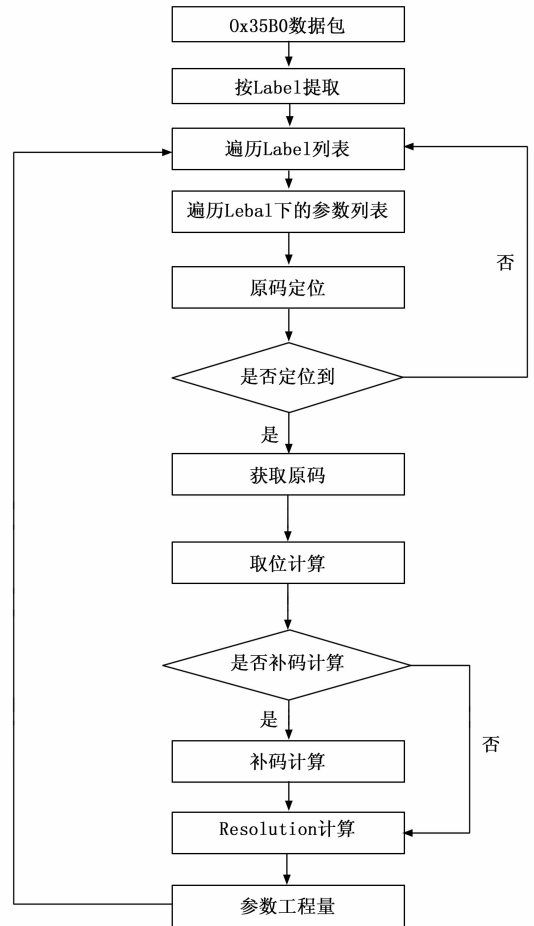


图 9 IMB 总线参数解析流程

工程量在内存中存储, 并按照参数在总参数列表中的索引位置, 进行链表的插入和删除, 且插入和删除过程互斥。

插入操作: 总线解析线程产生的参数工程量发送至数据缓存管理线程, 线程依据参数名对应的位置定位所有参数在链表指针, 进行拼接插入;

删除操作: 当完成某一次时间段内参数组导出后, 数据缓存管理线程依据从主控线程收到的删除时间段消息从链表中删除线程内所有参数的历史工程量数据, 防止历史数据在链表空间累积。

### 3.2.7 监控转发模块

监控转发模块通过加载多个监控转发列表文件, 完成监控参数与总参数列表的位置映射。当数据开始解析后, 按照监控转发列表文件设定的固定频率从各参数的链表中获取最新工程量数据, 以一定协议打包成二进制数据包, 通过 UDP 组播流形式按照预设转发频率周期性转发。

### 3.2.8 参数组导出存储模块

参数导出模块的主要功能是按照预设参数组信息, 对不同采样率的参数进行时间对齐, 完成试飞数据在数据文件中的存储和压缩。导出过程如图 10 所示。

在参数组导出时, 假定该组参数参数采样率包括 16 Hz 和 5 Hz, 导出采样率为 16 Hz。在共同导出时无法得出合理

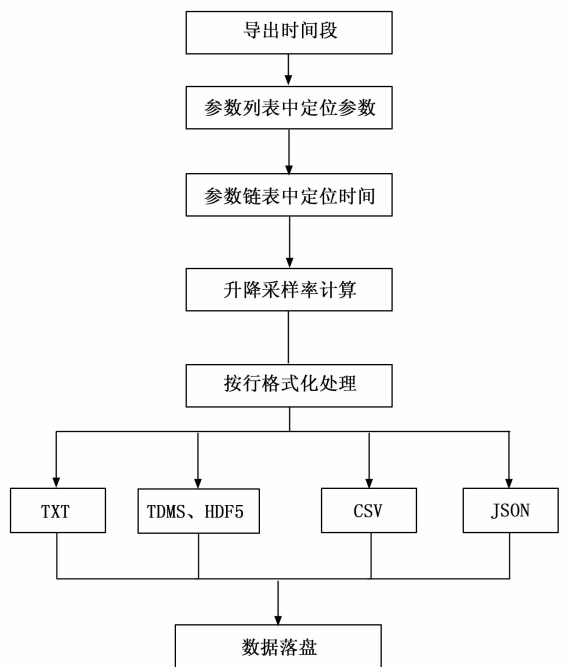


图 10 参数组导出过程



的工程量数据，所以必须将每个参数的采样率进行升序（目标采样率的倍数关系，如 80 Hz），然后按照升序后的采样率（80 Hz）与导出时采样率（16 Hz）倍数关系，提取对应的参数数据。由于网络采集器在实际采样过程中，会出现一定采样间隔偏差，为了弥补这一偏差，需要在导出某一参数时，先按照导出后采样率进行时间计算，再根据所得到的时间在参数内存区域查找最接近该时刻的时间戳。原理如图 11 所示。

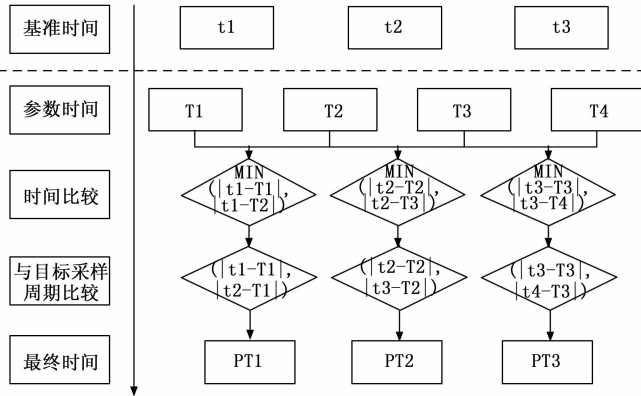


图 11 时间对齐原理

据导出子软件、DB 存储子软件的状态信息（其中 DB 存储子软件只发送软件运行状态和日志信息，原始数据流、参数组、监控转发等信息均由数据导出子软件产生）。

监控调试软件的界面采用 Qt Widgate 框架实现<sup>[19]</sup>，用于人机交互，可展示设备的负载、软件运行状态、数据参数可视化、工作日志以及参数组导出、监控转发等状态信息，软件界面与工作流程如图 13 所示。

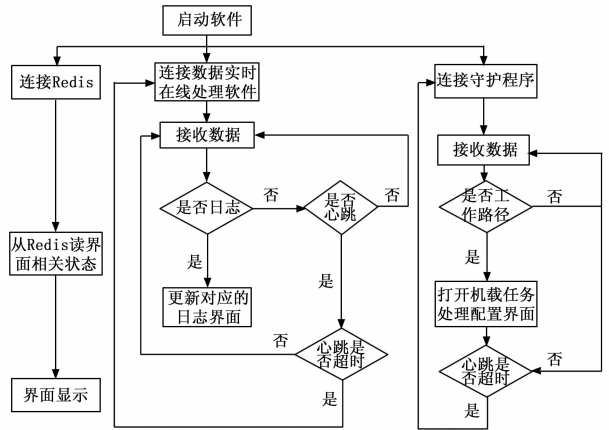


图 13 监控调试软件工作流程

### 3.2.9 基于 IoTDB 的数据存储模块

IoTDB 数据存储模块主要是将解析后的工程量数据实时写入 IoTDB 时序数据库，利用其高压缩比的特性压缩数据体量，方便后续的查询及分析。在 IoTDB 中，每一个试飞参数就是一个时间序列，命名为“root.飞机型号.架机号.试飞任务单号.试飞地点.试飞日期.试验类型.参数名称”，其中“root.飞机型号.架机号”为存储组。

### 3.3 监控调试软件

监控调试软件运行在机载监控终端上，主要对机载处理服务器设备和数据实时处理软件的运行状态以及数据存储情况等监控、统计和可视化显示，架构如图 12 所示。

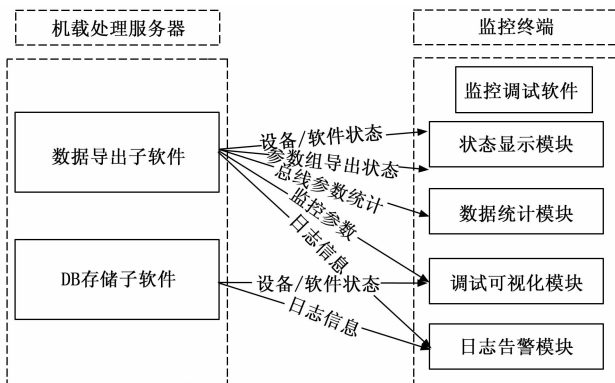


图 12 信息交互示意图

### 3.4 机载高频参数监控软件

机载高频参数监控软件包括监控画面编辑和监控画面显示两个子软件，主要功能是在机载环境下，实现振动等采样频率超过 8 192 Hz 高频参数的实时监控和分析。

两个子软件分别对应这监控画面的设计模式和运行模式。在设计模式下，用户使用监控画面编辑软件，通过新建或者打开已有的 XML 文件，拖拽显示控件进行监控画面的布局设计，在完成基础界面设计后，通过“绑定”的方式为显示控件加载相应的参数和分析算法，并保存为 XML 文件；在运行模式下，由监控画面显示软件加载并解析 XML 文件，根据指定的 IP 地址和端口向 IoTDB 时序数据库完成数据的订阅和实时显示。

### 3.5 地面数据卸载软件

地面数据卸载软件工作于地面环境，主要功能是在飞行试验结束后，从机载数据处理服务器中实现数据输出文件的压缩和卸载，并上传至指定的 FTP 服务器，完成数据的分发和交付，包括数据导出和数据移交两大功能模块。

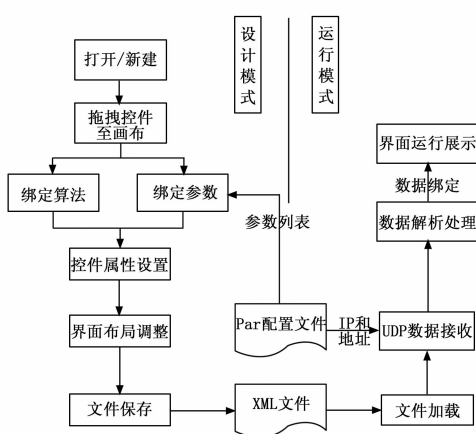
地面数据卸载软件启动后，数据导出模块从数据实时在线处理软件中读取数据导出格式信息（如 txt、csv 等），并按照预设文件名称自动压缩后存储在硬盘中；数据移交模块的功能是将压缩后的数据包自动推送至指定的服务器。

## 4 实验结果与分析

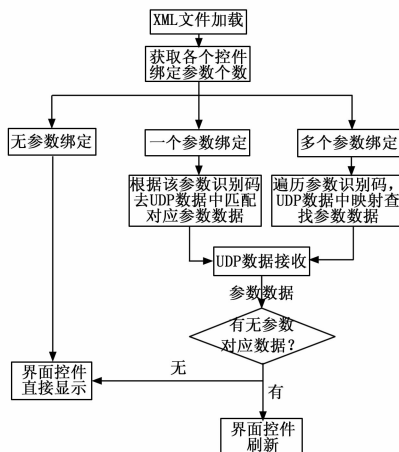
### 4.1 试验环境

试飞数据预处理系统按照图 15 的架构在地面和机载环境进行部署，软件运行环境如表 3 所示。

实际工作时，监控调试软件从 Redis 内存数据库获取数



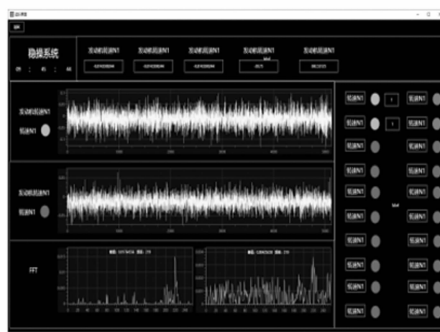
(a) 机载高频软件工作流程



(b) 数据流实时接收过程



(c) 画面编辑软件



(d) 画面显示软件

图 14 软件相关信息

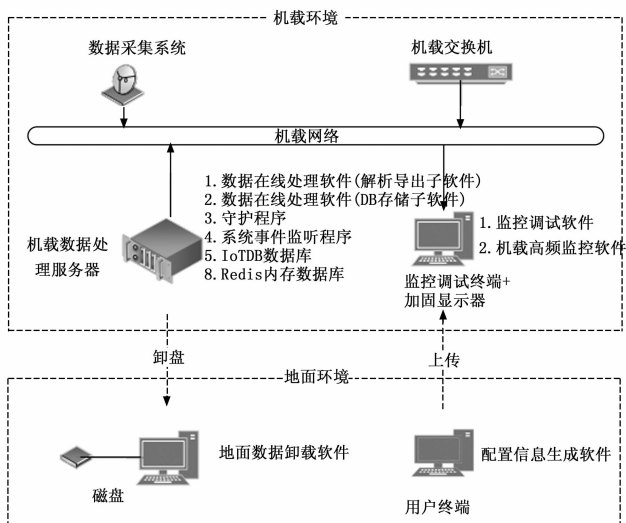


图 15 试飞数据预处理系统运行架构

### 4.2 试验步骤

将机载实时处理服务器安装于国产某大型客机, 按照以下步骤进行试验:

1) 启动机载测试系统的 ARINC429、ARINC664、IMB 等总线以及模拟量采集器;

表 3 测试环境参数

序号	类型	项目	指标
1	机载实时处理服务器	CPU	2 颗 Intel Xeon Gold 6 148 处理器, 2.4 GHz 主频, 20 核心
		内存	128 GB
		操作系统	Windows Server 2012
2	数据源	流量	220 Mbps
		数据流类型	NPD
		参数数量	78 837 个
		高频参数	391 个(采样率为 8 192 Hz)
		参数预设分组	350 个

2) 启动机载实时处理服务器上的数据实时在线处理软件和机载记录器;

3) 启动监控终端上的监控调试软件, 对数据流量、数据实时在线处理软件的运行状态以及数据存储状态进行可视化监控;

4) 启动监控终端上机载高频参数监控软件, 加载监控画面 XML 文件, 观察高频参数的时、频域曲线;

5) NPD 数据包包头中解析出高频参数装包绝对时间  $T_1$ , 记录高频数据经频谱分析后的绝对时间  $T_2$ , 计算延时  $T_2 - T_1$ ;

6) 系统持续运行 5 小时后, 将存储模块拆卸至地面, 使用地面数据卸载软件卸载数据文件;

7) 利用航后预处理系统解析机载记录器中的原始数据, 将输出的数据文件与步骤 (6) 的数据文件进行对比。

试验结果如表 4 所示。

表 4 系统测试结果

序号	项目	机载在线完全预处理系统	航后预处理系统
1	导出参数数量	78 837 个	78 837 个
2	数据包导出数量	350 个	350 个
3	参数值对比	参数值一致, 时间戳信息无偏差	
4	高频参数实时监控	延时约为 200 ms, 振动数据频谱正常	/
5	数据交付时间	20 分钟	4~5 小时

经试验测试, 机载在线完全预处理系统在实时状态下完成了 220 Mbps 流量下 NPD 数据包的解析, 实现了 78 837 个试飞参数 350 个数据文件的实时存储。

### 5 结束语

本文利用超高速线程并行处理技术设计研发了一套试飞数据预处理系统, 在机载环境下实现了某国产大型客机全量试飞数据的实时预处理, 在线完成了 200 Mbps 带宽下近 80 000 个参数的工程量转换、在线存储和实时监控, 将原先 4~5 小时的预处理周期压缩到了 20 分钟, 大大提高了民机试飞效率。

#### 参考文献:

[1] 黄涛, 王伟. 大型民机试飞测试发展与挑战 [J]. 计算机测量与控制, 2016, 24 (9): 5-7.

[2] 刘涛, 冯灿, 李成浩. 民机试飞遥测地面实时数据处理软件设计与研发 [J]. 计算机测量与控制, 2022, 30 (12): 195-202.

[3] 孙健, 霍培锋. 大型飞机试飞测试现状与对策 [J]. 测控技术, 2007, 26 (3): 19-21.

[4] 刘海波, 李国柱, 赵峰. 飞行记录系统发展概况 [J]. 运输机工程, 2002 (124): 19-21.

[5] 张鹏, 张建业, 李学仁. 飞行数据趋势监测在飞机“健康状态”评估中的应用 [J]. 空军工程大学学报 (自然科学版), 2004, 5 (3): 5-8.

[12] PAN X, WU J, LI Z, et al. Self-calibration for linear structured light 3D measurement system based on quantum genetic algorithm and feature matching [J]. Optik-International Journal for Light and Electron Optics, 2021, 225 (1): 165749.

[13] 任同群, 江海川, 张建昆, 等. 微小磁性零件装配设备自动标定及误差补偿 [J]. 光学精密工程, 2023, 31 (2): 214-225.

[14] 程俊森, 吴文荣, 杨毅, 等. 智能微装配技术研究综述 [J]. 现代制造工程, 2022 (6): 142-152.

[15] 王晓东. 精密微小装配中的传感器与测量技术 [J]. 计测技术, 2021, 41 (5): 10-22.

[16] RANA S, JASOLA S, KUMAR R. A review on particle swarm optimization algorithms and their applications to data clustering [J]. Artificial Intelligence Review, 2011, 35 (3):

[6] 李利荣, 孙立伟, 杨浩, 等. 基于 Spark 的飞机试验数据预处理技术研究 [J]. 计算机测量与控制, 2018, 26 (12): 260-264.

[7] 段宝元, 穆永花. 以太网在新支线飞机试飞测试中的应用 [J]. 测控技术, 2011, 30 (4): 91-93.

[8] 郑先成, 张晓斌, 雷涛. 大型飞机综合试验分布式测试系统研究 [J]. 测控技术, 2008, 27 (1): 70-71, 83.

[9] 郑先成, 张晓斌. 基于以太网的大型飞机电气特性测试系统 [J]. 计算机测量与控制, 2008, 16 (6): 749-751.

[10] 宋政斌, 郭晓玲, 于艳. 网络监听技术在飞行试验机载测试中的应用与研究 [J]. 计算机测量与控制, 2009, 17 (10): 1917-1919, 1922.

[11] 李五一, 闫楚良, 赵丽娜. 飞行试验机载网络数据实时处理技术研究 [J]. 计算机测量与控制, 2015, 23 (10): 78.

[12] 吴海东, 聂睿. C919 机载数据实时处理系统设计 [J]. 计算机测量与控制, 2019, 27 (4): 159-162.

[13] 祁春, 聂瑶佳, 周训强. 嵌入式新型机载实时数据处理系统的设计与实现 [J]. 计算机测量与控制 [J]. 2018, 26 (3): 208-211.

[14] 刘语乔, 齐婵颖, 贾雨. 机载网络数据实时处理系统设计 [J]. 计算机测量与控制, 2017, 25 (12): 240-242, 246.

[15] 李彦辰, 艾庆忠, 王少非. 基于 Redis 的分布式搜索引擎研究 [J]. 软件导刊, 2018 (3): 25-26.

[16] 段宝元, 穆永花, 穆永河. 新飞机测试改装网络数据到 PCM 的转换设计 [J]. 计算机测量与控制, 2014, 22 (10): 3356-3358.

[17] 袁炳南, 张建琳. PCM 与网络数据采集系统技术分析 [J]. 测控技术, 2009, 28 (4): 29-31.

[18] 石冯磊, 李国星, 高盼, 等. 消息重复型 ARINC429 总线事件消息包机载实时处理技术研究 [J]. 计算机测量与控制, 2017, 25 (5): 195-198.

[19] 孙好婕, 赵利强, 郑惠泽, 等. 基于 Qt 的集成电路测试软件设计与实现 [J]. 计算机测量与控制 [J]. 2021, 29 (5): 150-153.

[20] 白仲斐, 周星星, 张袁志. 颤振试飞测试系统的设计与实现 [J]. 计算机测量与控制, 2015, 23 (2): 348-350, 354.

[17] 肖晓伟, 肖迪, 林锦国, 等. 多目标优化问题的研究概述 [J]. 计算机应用研究, 2011, 28 (3): 805-808, 827.

[18] 张利彪, 周春光, 马铭, 等. 基于粒子群算法求解多目标优化问题 [J]. 计算机研究与发展, 2004 (7): 1286-1291.

[19] 周涛, 张艳宁, 袁和金, 等. 基于改进粒子群算法的支持向量机 [J]. 计算机工程与应用, 2007, 574 (15): 44-46.

[20] 吴海, 孙永雄, 韩伟, 等. 多核环境下的粒子群算法 [J]. 吉林大学学报 (信息科学版), 2012, 30 (5): 549-554.

[21] GU Q, LIU Y, CHEN L, et al. An improved competitive particle swarm optimization for many-objective optimization problems [J]. Expert Systems with Applications, 2022, 189: 116118.