

## 基于 Linux 实时操作系统的配电网保护控制终端平台设计与应用

古展基<sup>1,2</sup>, 谷武健<sup>1</sup>, 蔡泽祥<sup>3</sup>, 苏忠阳<sup>1</sup>, 黄勇<sup>1</sup>, 余传坤<sup>1</sup>

(1. 广州穗华能源科技有限公司, 广东 广州 510530; 2. 深圳市国电科技通信有限公司, 广东 深圳 518109;  
3. 华南理工大学电力学院, 广东 广州 510640)

**摘要:** 当前配电网保护终端平台仍采用封闭架构, 高昂的平台操作系统授权价格以及封闭的开发环境, 抑制了配电网保护控制业务海量的需求。针对开发者代码隔离诉求和网络保护性能约束, 提出了一种基于 Linux 实时操作系统的配电网智能保护控制终端新平台。新平台拥有 Linux 开源、代码隔离、业务实时性支撑等特点。首先基于普通 Linux 操作系统嵌入实时补丁、设计实时进程调度和应用内存共享机制, 以构造新平台基础。然后, 设计了数据实时处理模块和保护业务服务模块, 以服务配电网保护控制应用。最后, 通过搭建软件测试环境, 验证了新平台进程实时处理能力、重采样抖动控制能力。通过搭建 RTDS 实时数字仿真系统, 验证了新平台下的网络保护应用性能。测试均取得了预期效果, 验证了新平台设计的工程适应性。

**关键词:** 配电网; Linux; 保护控制; 操作系统; 实时

### Design and application of distribution network protection and control terminal platform based on the Linux real-time operating system

GU Zhanji<sup>1,2</sup>, GU Wujian<sup>1</sup>, CAI Zexiang<sup>3</sup>, SU Zhongyang<sup>1</sup>, HUANG Yong<sup>1</sup>, YU Chuankun<sup>1</sup>

(1. Guangzhou Smart Energy Technology Co., Ltd., Guangzhou 510530, China; 2. China Gridcom Co., Ltd., Shenzhen 518109, China; 3. School of Electric Power Engineering, South China University of Technology, Guangzhou 510640, China)

**Abstract:** The current distribution network protection terminal platform still adopts a closed architecture, and incurs a high licensing price of the platform operating system as well as a closed development environment. This all inhibits the massive demand for distribution network protection and control services. Considering developers' code isolation requirements and network protection performance constraints, a new platform for distribution network intelligent protection and control terminal based on the Linux real-time operating system is proposed. The new platform has the characteristics of Linux open source, macro kernel, real-time business guarantee and so on. First, this paper embeds real-time patches, designs real-time process scheduling and applies a memory sharing mechanism to a common Linux operating system to construct a new platform foundation. Then, the sampling data real-time processing module and the protection business service module are designed to serve the distribution network protection control application. Finally, by building a software test environment, the real-time processing capability and resampling jitter control capability of the new platform process are verified; by building an RTDS real-time digital simulation system, the network protection application performance under the new platform is verified. The tests all achieve the expected results, which verifies the engineering adaptability of the new platform design.

This work is supported by the National Natural Science Foundation of China (No. 51577073).

**Key words:** distribution network; Linux; protection and control; operating system; real-time

## 0 引言

在电力系统中, 短路故障元件受动稳定和热稳

定单一或双重的约束, 亟需继电保护装置在毫秒级别的时间内快速识别故障和出口控制信号<sup>[1-5]</sup>。在继电保护装置中, 继电保护各种复杂功能都是由相应的应用软件(保护逻辑)来实现的, 继电保护应用程序对实时性要求很高, 各种应用软件都需运行在实时性极高的平台中, 在平台程序设计过程中, 任务

**基金项目:** 国家自然科学基金项目资助(51577073); 广东省重点领域研发计划项目资助(2019B111109002)

需要及时响应, 进程调度得当, 避免多个进程争夺资源进入到死锁状态<sup>[6-10]</sup>, 平台需保障继电保护应用的正常运行。在电力、航天、军工等实时性要求较高的领域, 常用的是 vxWorks 操作系统, vxWorks 需要高昂的授权使用费用和封闭的开发环境, 并且非自主可控。

配电网保护控制新平台需要的是一套高实时、稳定可靠、易开发、易裁剪、多任务且自主可控的操作系统。而 Linux 操作系统为国产化平台提供了一个非常好的选择: (1) 支持广泛的芯片硬件, 包括 X86、ARM、Intel 等现有的大部分芯片<sup>[11-14]</sup>; (2) 源代码全部公开, 任何厂商都可以移植定制并在通用公共许可证(GNU general public license, GNU GPL)下发布, 且相互兼容, 使应用跨平台部署成为可能; (3) 开发生态极其活跃, 互联网社区使得问题解决便利化; (4) 完善的开发工具, 如编译器、调试器、解析工具和接口文档; (5) 强大的网络功能, 支持 OSI 七层模型, 使得实现 IEC61850 通信体系成为可能。

本文在分析当前配电网平台技术现状的基础上, 针对开发者诉求和网络保护性能, 主要介绍一种基于 Linux 实时操作系统的配电网保护控制终端新平台。

### 1 配电网保护控制终端平台综述

#### 1.1 配电网保护控制终端平台描述

配电网保护控制终端平台定位于: (1) 配电网保护控制终端平台介于硬件层和应用层之间; (2) 平台向上为保护应用提供数据服务, 向下通过内核管理硬件, 保护应用只需要关心业务逻辑本身。

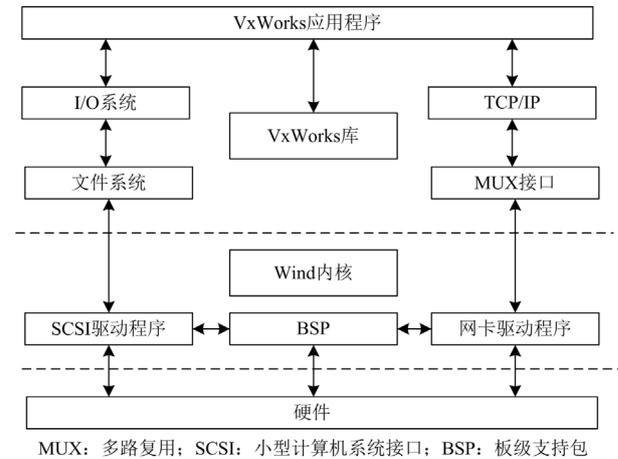
配电网保护控制终端平台的应用场景包括:

(1) 平台负责数据采集、处理及存储, 为应用提供数据服务, 解耦应用与硬件的代码交叉; (2) 平台支持 IEC61850 通信协议, 可有效解决配电网各终端之间的通信, 为保护提供信息共享业务; (3) 平台具有统一的数据模型, 支持 ICD/SSD/SCD 等模型, 通用的模型为信息共享提供解析基础; (4) 平台实时完成采样数据到重采样数据的计算过程, 为配电网保护提供差动等所需的采样值服务, 同时也能为其他保护提供计算值, 统一数据处理实现了业务与数据的代码隔离。

#### 1.2 当前终端平台技术现状

如图1所示, 目前保护应用大多是基于 vxWorks 操作系统开发的。vxWorks 是基于微内核, 即内核只提供了基本的服务, 如任务管理、内存管理、中断处理等。与 Linux 分开内核空间和用户空间不同, vxWorks 没有用户空间和内核空间之分。设备驱动

和应用都运行于同一代码空间, 相互之间的内存都可以访问<sup>[15]</sup>。



MUX: 多路复用; SCSI: 小型计算机系统接口; BSP: 板级支持包

图1 vxWorks 体系保护平台

Fig. 1 vxWorks system protection platform

传统的保护应用与 vxWorks 捆绑甚至与硬件驱动捆绑<sup>[16-17]</sup>, 形成了终端以业务纵向划分的体系, 即业务需求的数据获取方式都采用以纵向新增终端装置, 或者采用独立网络获取现存终端设备数据。不同业务之间的终端交互较少, 进行应用扩展时, 由于不同终端接口标准不一, 往往采用“打补丁”的方式, 造成了资源浪费, 进一步加深了业务割裂程度。配电网存在保护、测控、电压控制、无功控制、微网控制等需求, 传统的终端平台模式已无法适应。

#### 1.3 vxWorks 和 Linux 实时操作系统的差异分析

分别从开发成本、实时性等方面对 vxWorks 和 Linux 实时操作系统进行定性分析, 如表 1 所示。

表1 基于 vxWorks 和 Linux 实时操作系统的差异

Table 1 Difference between vxWorks and Linux real-time operating system

平台名称	vxWorks	实时 Linux
开发成本	大量付费, 小部分开源	全开源, 全免费
实时性	强实时性	强实时性 (须打实时补丁)
设备驱动的支持	部分	绝大部分
稳定性	非常稳定	稳定
扩展性	一般	非常好
内核结构	微内核	宏内核
运行模式	实模式	保护模式: 区分用户 地址空间和内核 地址空间

如表 1 所示, 从实时性和稳定性两个维度看, vxWorks 和实时 Linux 操作系统都能满足电力系统

保护应用要求。然而, 从我国实情、开发成本、维护成本和可扩展性来讲, 实时 Linux 应是未来终端平台操作系统的应用趋势之一。

## 2 配电网保护控制应用对新平台的约束

### 2.1 配电网保护控制应用对新平台性能的约束

网络保护技术(含主从式和对等式通信的网络保护)是对终端平台数据处理性能要求最高的典型应用之一, 要求能对网络拓扑任意多端口网络进行差动计算。因此需要新平台提供多端口的采样值(来自于多个终端)<sup>[18-22]</sup>。

此外, 电力系统的故障多为突发性故障, 系统内周期性数据流和突发性数据流会挤占平台的网络端口和缓存<sup>[23]</sup>。

因此, 新平台应具备数据大流量接收能力、数据解析加工服务能力以及数据实时处理能力。

### 2.2 开发者对新平台代码隔离的约束

#### 1) 平台与继电保护应用的隔离

由于继电保护应用的特殊性和需求多样性, 继电保护应用随地区或场景的变化会出现多个不同的版本, 即平台需要支撑多种业务的运行。因此, 从开发者的角度来讲, 平台与应用的分离能够隔离开发的风险, 有利于应用的快速迭代, 有利于地区特殊业务的定制。

从运维者的角度来讲, 平台与应用的分离能够快速定位软件漏洞, 快速解决运维问题。

#### 2) 平台与硬件的隔离

脱离目标硬件, 基于新平台开发调试软件, 是提升开发效率的重要措施, 也是硬件变化后, 软件便捷移植的要求。因此, 需要对平台和硬件(包括硬件设备和硬件服务)进行代码隔离。

## 3 新平台架构

约束下的新平台整体架构和上下层关系如图 2 所示。具体如下。

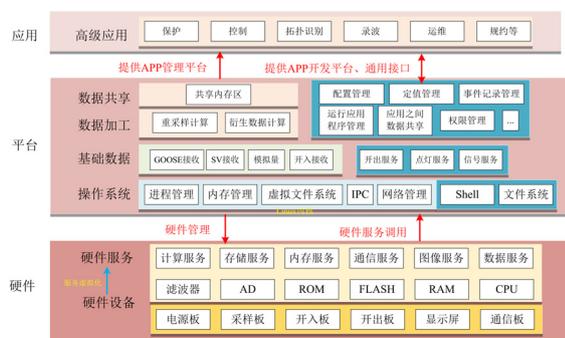


图 2 软硬件整体架构

Fig. 2 Software and hardware overall architecture

- 1) 上层: 主要为继电保护等配电网专业应用。
- 2) 下层: 硬件服务虚拟化, 即对硬件资源软件定义, 将终端的滤波器、AD、ROM、Flash、RAM、CPU 等硬件资源按照一定规则定义为计算服务、存储服务、内存服务、通信服务、图像服务、数据服务。

3) 平台分为操作系统层、基础数据层、数据加工层、数据共享层。

平台具体定义如下。

- 1) 平台下层操作系统裁剪移植普通 Linux 内核、Linux Shell、Linux 文件系统, 其中 Linux 内核嵌入 RT-preempt Patch 实时补丁。
- 2) 基础数据包括 SV、GOOSE 接收、模拟量接收、开入开出、点灯服务以及信号服务。
- 3) 数据加工: 重采样计算、衍生数据计算服务。
- 4) 数据共享: 包括数据加工后的数据共享, 还包括配置管理、定值管理、事件记录管理以及其他服务组件。

## 4 新平台基础和关键软件模块设计

### 4.1 新平台中操作系统与普通 Linux 操作系统的共有部分

新平台基于 Linux 操作系统, 裁剪移植普通开源 Linux 内核、Linux Shell、Linux 文件系统套件<sup>[24]</sup>。普通 Linux 架构如图 3 所示。

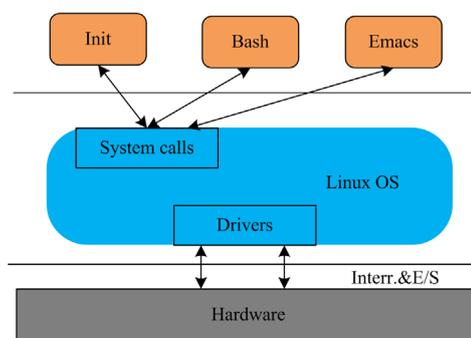


图 3 普通 Linux 操作系统架构

Fig. 3 Ordinary Linux operating system architecture

其中新平台 Linux 内核机制与普通 Linux 操作系统一致。对下, 统一调度系统的硬件设备; 对上, 通过系统调用, 向 Library Routine(例如 C 库)或者其他应用程序提供接口<sup>[25]</sup>。

为实现新平台工程应用, 需要在进程实时性、进程调度、内存共享机制方面构建新平台基础模块, 同时在保护业务方面进行关键模块设计。

### 4.2 新平台基础模块一: 进程实时性设计

所谓实时任务执行, 即要求平台能及时响应外

部事件的请求，在规定的时间内完成对该事件的处理。实时又分软实时和硬实时。软实时，在某些情况下不完成任务，也不会对系统本身产生致命影响。硬实时，要求在任何情况下都须完成任务，否则后果不可控。

一次元件故障时，智能终端的 SV 和 GOOSE 报文从接收、计算分析、出口实时性要求极高，要求新平台能够接收并以足够快的速度予以处理，其处理的结果又能在规定的时间内通过控制开出节点继电器快速响应，要求拥有控制所有实时任务协调一致运行的操作系统。所以，新平台应是硬实时平台。

影响配电网保护控制终端新平台实时性的主要为抢占延时，大多数实时系统都是处理一些周期性或非周期性的重复过程，事件产生的周期确定了任务的执行时限，因为每次事件发生时，相应的处理任务必须及时响应处理，否则将无法满足时间要求<sup>[26]</sup>。

普通 Linux 的实时性问题在于：(1) 内核不可抢占；(2) 中断关闭；(3) 自旋锁；(4) 中断为最高优先级；(5) 调度机制和调度不当等造成的进程延迟响应。

为解决普通 Linux 的实时性问题，全球最大的 Linux 社区 Linux Kernel Organization，提供了一种 RT-preempt Patch 实时补丁(互联网地址：<http://cdn.kernel.org/>)，即实时抢占补丁。Linux Kernel Organization 是加利福尼亚州的一家公益组织，旨在免费向公众分发 Linux 内核和其他开源软件。其中补丁由 Ingo Molnar 和 Thomas Gleixner 维护，OSADL 实验室对其进行稳定性测试<sup>[27]</sup>。

RT-preempt Patch 实时抢占补丁采用的是整体式内核实时改造方法：(1) 中断线程化设计，使硬实时任务可以有比中断线程更高的优先级，保证进程抢占时仍有实时性；(2) 用互斥锁(Mutex)取代自旋锁(Spinlock)，使 Spinlock 可抢占；(3) 优先级继承和死锁检测，采用优先级继承协议(priority inheritance protocol)，即 spinlock 的保持者将继承高优先级竞争者的优先级，从而能先于中间优先级进程运行；(4) 等待队列优先级化。这些技术可以令 Linux 系统达到硬实时的要求<sup>[28]</sup>。目前，补丁的内容也逐渐并入到新版本的主线内核中。

因此新平台操作系统在普通 Linux 的基础上打入 RT-preempt Patch(实时补丁)，如图 4 所示，实时补丁嵌入到内核层，保障实时任务(RT-Task)优先执行。

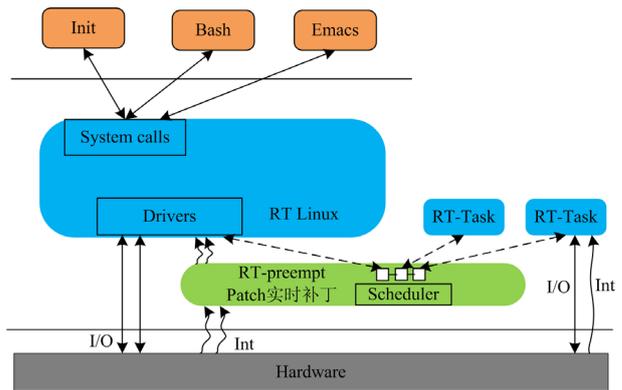


图 4 实时操作系统

Fig. 4 Real-time operating system

### 4.3 新平台基础模块二：进程调度设计

在终端硬件中，CPU 资源是有限的，而多个进程都需要占用 CPU 资源时，为了防止进程之间无序竞争，引入了进程调度规则，即 CPU 的访问控制管理，采用进程优先级策略和抢占式控制策略。

进程优先级如图 5 所示。

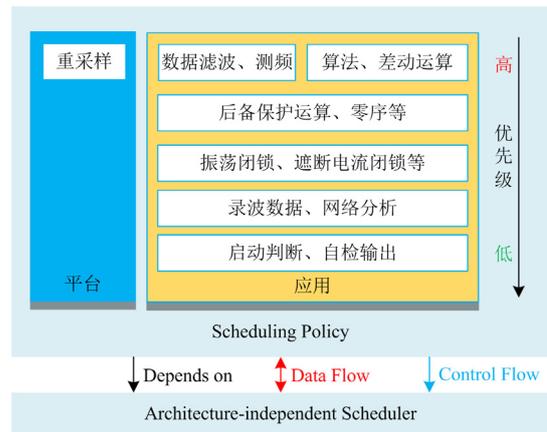


图 5 进程优先级

Fig. 5 Process priority

在新平台的实时操作系统内核中，重写 Scheduling Policy、Architecture-independent Scheduler、Architecture-specific Schedulers 模块。Scheduling Policy 是实现进程调度的策略，决定着是否分配进程 CPU 资源；Architecture-independent Scheduler 是与体系结构无关的部分，它会向“Scheduling Policy”模块请求并确定接下来要执行哪个进程，最后通过“Architecture-specific Schedulers 模块” resume((中断后)继续)运行指定的进程。

在新平台中，采用“SCHED\_FIFO”抢占式实时调度策略(按照优先级占用 CPU，一旦占用则一直运行，直到比其更高优先级任务到达或自动释

放)。SCHED\_FIFO 与优先级策略组成整个进程调度机制。

#### 4.4 新平台基础模块三：共享内存设计

采用基于共享内存的进程间通信方式，如图 6 所示，为减少进程间读写的环节，即进程地址空间直接连接到物理内存中，直接访问共享内存中的地址。

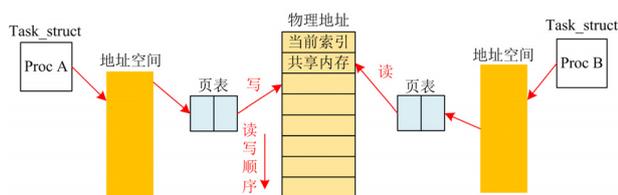


图 6 共享内存的数据写入读取过程

Fig. 6 Shared memory data writing and reading process

数据计算写入进程向共享内存写入数据时，可能会影响到同时访问同一段共享内存的其他进程。为了保证数据读取在数据写入之后完成，需要协调共享内存的访问。共享内存缓冲策略如图 7 所示。其中： $T_c$  为重采样计算周期； $T_j$  为应用计算周期。

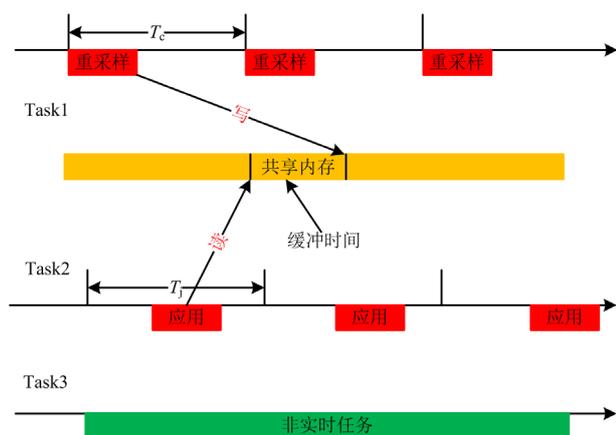


图 7 共享内存缓冲时序图

Fig. 7 Shared cache buffer timing diagram

为保证读取序列号滞后于写入序列号，依靠索引号  $i_{dx}$  缓冲，策略如下。

每次写入：重采样值和索引号加 1；

每次读取：读取索引号和索引号减 1 的重采样值。

#### 4.5 新平台关键模块一：保护控制业务架构设计

基于保护控制应用和硬件之间的数据分析，新平台保护业务模块如图 8 所示。

##### 1) 配置管理模块

配置解析为单独的程序，支持高级语言开发，能够快速开发和迭代。配置隔离设计可以为“源端维护”

等功能预留接口空间。配置接口封装为库，提供给应用进程调用。

##### 2) 数据处理模块

对 GOOSE、SV、模拟量、常规开入的读取、解析、计算和数据共享。

##### 3) 信号、开出、点灯模块

对信号、出口、灯进行控制。

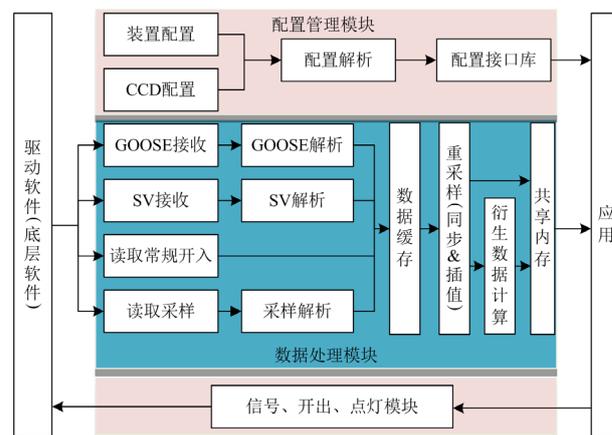


图 8 新平台保护业务模块架构设计

Fig. 8 Protection module design of the new platform

#### 4.6 新平台关键模块二：重采样设计

##### 4.6.1 电气量重采样模块设计

对于连接多侧的联络电房，智能终端需要接入多侧的 SV 和 GOOSE 报文，由于每个终端的原始模拟采样时刻不同，采集电路的耗时也会有差异。智能终端需要对侧报文进行采样、对齐、同步和等间隔重新采样，即为重采样过程。采用线性插值算法，产生的误差对保护应用的影响可以忽略<sup>[29-30]</sup>。

新平台的关键服务是数据的高效处理，主要对数据缓存中的数据，包括 SV、GOOSE、模拟量、常规开入进行重采样，将重采样数据输送到共享内存中，供应用读取。数据的关键处理流程如图 9 所示。

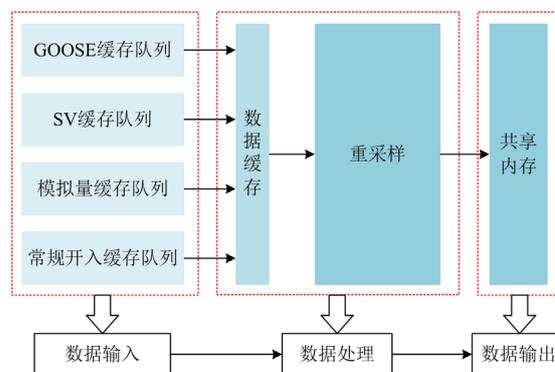


图 9 数据处理流程

Fig. 9 Data processing flow

重采样过程是对前一采样周期的数据缓存进行处理,即回退到前一时间周期内计算缓存中的采样数据(如 10 ms)。

重采样周期可为 250/500/1000 μs,由应用调用(保护应用一般为 500/1000 μs,录波要求一般为 250 μs)。如果采用向 AD 模拟量对齐的方式,此时相当于抽点重采样,会存在两个时钟(重采样平台软件时钟、AD 硬件时钟)。正常运行时这两个时钟会有一个固定的误差,然而运行一段时间,两个时钟误差会加剧。因此,每间隔一段时间需要修正重采样平台的软件时钟。

重采样按照固定的时间间隔进行,如果没有采样点,或者该时刻采样点不存在,应当将数据的质量置为异常。

采样值共享内存和开入量共享内存,每个时刻的数据都应当是自动对齐的,即,同一个索引号访问到的采样值和开入值须是在同一时刻的。

重采样的算法设计如图 10 所示。

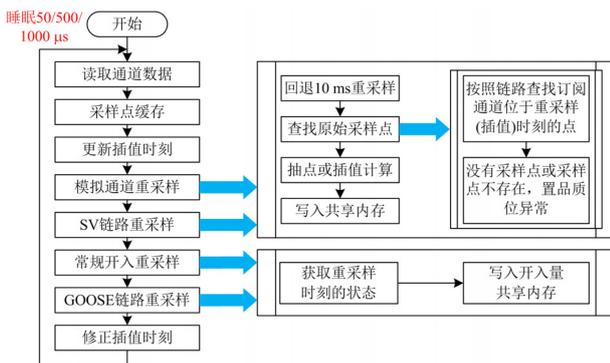


图 10 重采样算法设计

Fig. 10 Resampling algorithm design

### 3) 开入量重采样设计

保护等应用取到重采样数据,在满一周波后,对每个采样点进行计算,然后根据计算值进行逻辑判断,要求每个计算值所处的时刻都要有一个对应的开入状态,开入值采样采用抽点采样模式。

#### 4.6.2 采样间隔均匀度保障设计

为了保证重采样频率一致、均匀且在合理抖动范围内,每完成一个重采样,需要对采样进程进行合理的“睡眠”,以释放 CPU 资源给其他进程使用。因此,引入“nanosleep”函数<sup>[31]</sup>:

```
intnanosleep(conststructtimespec *req,
structtimespec *rem);
structtimespec{
time_ttv_sec; /* seconds */
longtv_nsec; /* nanoseconds */
};
```

函数的功能是暂停某个进程直到规定的时间后恢复,参数 req 就是设置的暂停时间,其中 tv\_sec 是以秒为单位,而 tv\_nsec 以纳秒为单位(10<sup>-9</sup> s)。由于调用 nanosleep 函数是进程进入 TASK\_INTERRUPTIBLE(任务中断),这种状态可能会被相应信号打破而进入 TASK\_RUNNING(任务运行)状态。这就意味着有可能会没有等到规定的时间就因为被其他信号唤醒,此时函数返回-1,且还剩余的时间会被记录在参数 rem 中。

return: 若进程暂停到参数 req 所指定的时间,成功则返回 0,若有信号中断则返回-1,并且将剩余微秒数记录在 rem 中,以备下次中断累计使用,保障后续采样间隔恢复均匀。

## 5 新平台性能仿真测试

### 5.1 新平台实时性仿真测试

Cyclictest 程序用于测试新平台延时的精度<sup>[32]</sup>,线程间隔定时器(interval timer, iTimer)设定定时时间 T<sub>iTimer</sub>,在启动前记录下当前时间 T<sub>1</sub>,定时器到时后记录下当前时间 T<sub>2</sub>,则测出的时延 T 为

$$T = T_2 - (T_1 + T_{iTimer}) \quad (1)$$

测试场景:分别运行 5 个进程,线程运行周期分别是 1000、1500、2000、2500、3000 μs,进程优先级均为 80;硬件方面配置相同,基于 FPGA+ARM 异核硬件架构,分别记录普通 Linux 和实时 Linux 内核运行测试的结果。

测试程序:“#sudo ./cyclictest -p 80 -t5 -n”(p 80:进程优先级为 80;t5:运行 5 个进程;n:使用 clock\_nanosleep;单位:μs)。

测试结果如表 2 和表 3 所示。

由表 2 和表 3 的测试结果可以看出,实时 Linux 的进程调度时间可控制在 90 μs 以内。时延抖动低,进程调度低时延。

### 5.2 新平台保护业务数据处理能力测试

测试场景:运行傅里叶计算进程,2 kHz 采样频率,每次计算 100 个通道,循环 10 000 次。

测试重复 10 次。分别基于实时 Linux 和普通 Linux 平台运行测试结果。

测试程序:“fft\_test”。

测试结果如表 4 所示。

由表 4 的测试结果可知,基于实时 Linux 平台计算抖动很小,抖动约 10 μs,满足保护应用对衍生数据计算抖动的要求。而普通 Linux 平台的计算抖动时间达到了 200 μs。

表 2 普通 Linux 线程延时测试结果

Table 2 Ordinary Linux thread delay test results

线程	优先级	线程间隔/ $\mu\text{s}$	计数器/次	最小延时/ $\mu\text{s}$	最近一次的延时/ $\mu\text{s}$	平均延时/ $\mu\text{s}$	最大延时/ $\mu\text{s}$
T:0	80	1000	733 966	5	10	10	6398
T:1	80	1500	489 312	5	11	10	5444
T:2	80	2000	366 984	5	10	10	5454
T:3	80	2500	293 588	5	10	10	4488
T:4	80	3000	244 656	5	10	11	5490

表 3 实时 Linux 线程延时测试结果

Table 3 Real time Linux thread delay test results

线程	优先级	线程间隔/ $\mu\text{s}$	计数器/次	最小延时/ $\mu\text{s}$	最近一次的延时/ $\mu\text{s}$	平均延时/ $\mu\text{s}$	最大延时/ $\mu\text{s}$
T:0	80	1000	1 131 760	7	12	12	43
T:1	80	1500	754 507	7	15	12	45
T:2	80	2000	565 880	7	12	12	34
T:3	80	2500	452 704	7	12	12	44
T:4	80	3000	377 253	7	12	13	89

表 4 计算时间统计

Table 4 Calculation time statistics

测试次数	实时 Linux 平台/ $\mu\text{s}$	普通 Linux 平台/ $\mu\text{s}$
1	3061	3061
2	3061	3062
3	3060	3262
4	3060	3061
5	3060	3061
6	3079	3217
7	3060	3061
8	3074	3218
9	3079	3129
10	3060	3061

### 5.3 新平台的保护应用性能测试

此次测试以对等式网络保护应用为测试业务, 在保护配置上, 终端之间交换对侧信息, 实现区内区外故障的快速识别、定位以及故障就地切除。

为了验证开发的新平台在网络保护中的性能, 在 RTDS 上搭建 10 kV 配电网仿真模型。

实验过程如下。

#### 1) 实验原型

在 RTDS 中搭建测试电“3 供 1 备”小电阻(10  $\Omega$ ) 接地电缆电网模型, 如图 11 所示。具体结构参数如下。

(1) 每个配电房编号统一: 中间开关接负载, 两边开关分别接其上下级的电房。

(2) 电源: 共 4 处接入 10 kV 供电电源 S1—S4。

(3) 开关: 在电源 S1—S4 接入点处安装断路器, 配电房 N9 负荷出线装设断路器, 其他未说明的均为负荷开关。

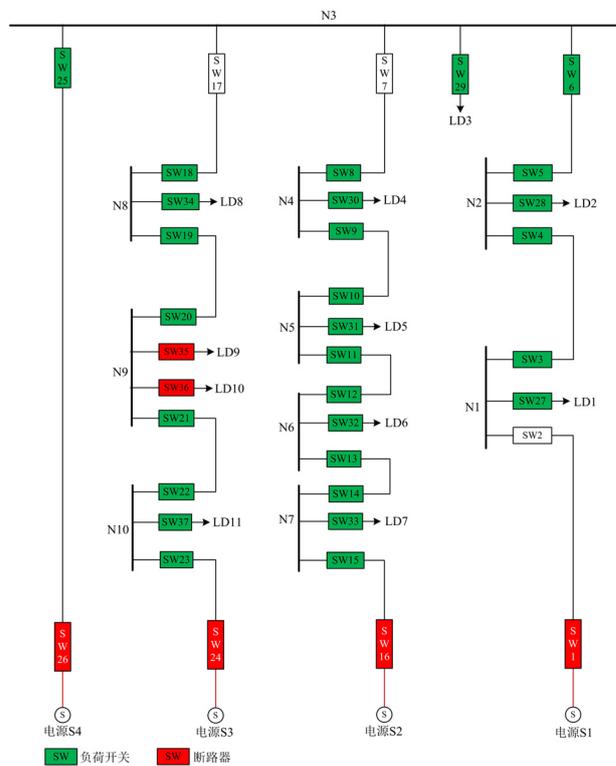


图 11 测试拓扑原型

Fig. 11 Test topology prototype

#### 2) 实验场景设置

实验的节点元件如图 12 所示, 典型实验如图 13、图 14 所示。

#### 3) 保护配置实例

为减小电压跌落对敏感负荷的影响, 配电网故障最好能在 150 ms 内切除。本次实验的保护应用实例配置对时延敏感性较高的差动保护, 如表 5 所示。

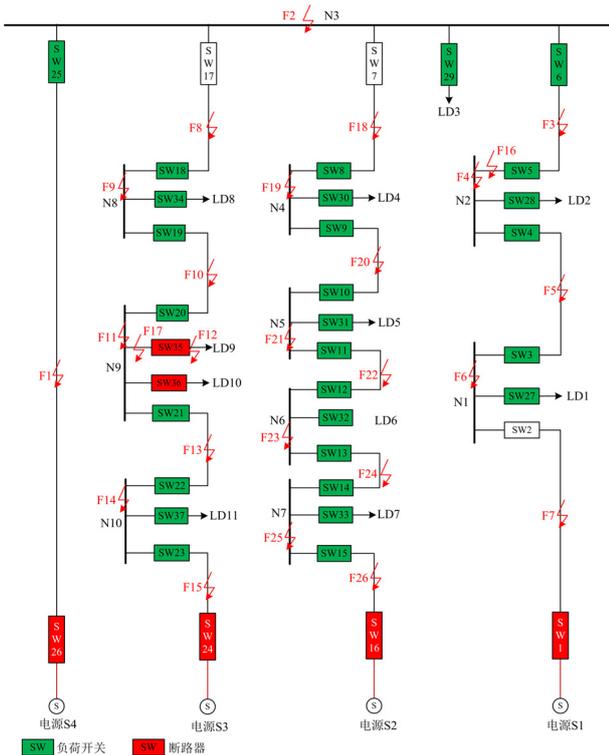


图 12 F1—F26 故障示意图  
Fig. 12 F1—F26 fault diagram

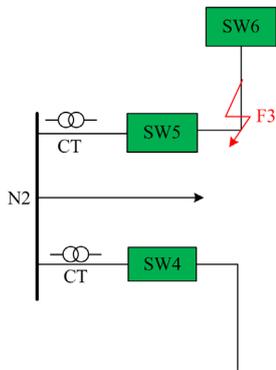


图 13 F3(线路)故障示意图  
Fig. 13 F3 line fault diagram

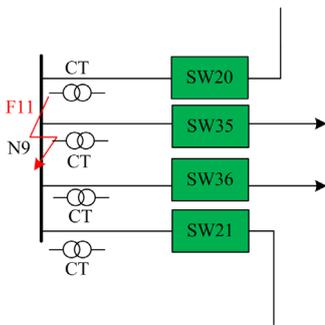


图 14 F11(母排)故障示意图  
Fig. 14 F11 bus fault diagram

表 5 实验终端保护配置实例

Table 5 Experimental terminal protection configuration

配置名称	配置描述
保护	母线差动保护、线路差动保护、零序差动保护 APP
隔离	拓扑搜索隔离故障 APP
自愈	拓扑搜索网络备自投 APP

4) 实验结果

挑选了部分典型的数据记录，如表 6 所示。

表 6 基于实时平台的保护应用动作时间统计

Table 6 Operating time statistics of protection applications based on real-time platform

试验条件	装置动作情况	动作时间/ms	
A 相接地		①106.25 ②1496	
		③1902 ④2110	
		①100 ②1508	
		③1916 ④2130	
B 相接地		①107.75 ②1507	
		③1917 ④2131	
		①87.25 ②1501	
		③1907 ④2117	
AB 相间短路	① 跳开关 SW26	①87.75 ②1501	
		③1907 ④2121	
		①87 ②1501	
		③1907 ④2121	
BC 相间短路	② 跳开关 SW6、SW5	①87.5 ②1497	
		③1903 ④2111	
		①87 ②1505	
		③1911 ④2121	
AC 相间短路	③ 合开关 SW2	①87 ②1501	
		③1907 ④2121	
		①87.5 ②1497	
		③1903 ④2111	
AB 两相接地		①87 ②1505	
		③1911 ④2121	
		①87 ②1499	
		③1911 ④2125	
BC 两相接地		①87.75 ②1507	
		③1915 ④2125	
		①107.75 ②1501	
		③1907 ④2125	
AC 两相接地		①101.25 ②1503	
		③1909 ④2119	
		①107.75 ②1495	
		③1899 ④2107	
ABC 三相接地		①87.5 ②1497	
		③1901 ④2113	
		①89 ②1501	
		③1907 ④2117	
模拟 F11 故障 (含负荷母线段故障)	① 跳开关 SW24	①87.75 ②1497	
		③1901 ④2115	
		② 跳开关 SW20、SW21	①87.25 ②1503
		③ 合开关 SW24	③1907 ④2115
AB 两相接地	④ 合开关 SW17	①87.25 ②1499	
		③1903 ④2111	
		①87.25 ②1501	
		③1905 ④2113	

注：接地过渡电阻为 30 Ω。

### 5) 实验结论

由表 6 可得, 基于实时平台的网络保护应用接地动作出口时间为 100~120 ms, 短路故障出口动作时间为 70~90 ms, 满足配电网保护出口时间要求。

## 6 结论

随着国产芯片、IT、通信技术的飞速发展以及 IEC61850 愈加广泛的应用, 传统配电网保护控制系统正与互联网深度融合。本文提出的基于 Linux 实时操作系统的新平台有以下优点: 为配电网保护应用运行提供开源的、实时的数据平台; 依托实时 Linux 对硬件资源虚拟化, 对数据、进程和资源进行合理的调度管理; 实现数据快速共享, “应用-平台-硬件”三层结构代码隔离。新平台提供统一的接口, 为应用开发者、运维者提供极大的开发便利。

配电网的特点在于终端分布面广、数量多、厂家繁多、需求丰富、标准难以统一, 因此在配电网中实现终端硬件、数据平台、应用软件的解耦和代码隔离, 对于支撑配电网保护应用的生态建设是一种较好的解决方案。

### 参考文献

- [1] 姚春球. 发电厂电气部分[M]. 北京: 中国电力出版社, 2020.
- [2] 朱声石. 高压电网继电保护原理与技术[M]. 4版. 北京: 中国电力出版社, 2014.
- [3] 徐丙垠. 配电网继电保护与自动化[M]. 北京: 中国电力出版社, 2017.
- [4] 全蕾, 詹红霞, 张勇, 等. 考虑多 DG 接入的配电网自适应电流主保护方案[J]. 智慧电力, 2021, 49(8): 63-69. QUAN Lei, ZHAN Hongxia, ZHANG Yong, et al. Adaptive current main protection scheme of distribution network accessed with multiple distributed generations[J]. Smart Power, 2021, 49(8): 63-69.
- [5] 周斌, 李赫, 郑海涯, 等. 基于零序电压相关系数的配电网单相断线故障保护方法[J]. 供用电, 2023, 40(2): 9-16. ZHOU Bin, LI He, ZHENG Haiya, et al. Single-phase line break fault protection method for distribution network based on zero-sequence voltage correlation coefficient[J]. Distribution & Utilization, 2023, 40(2): 9-16.
- [6] RANNIANZHIXIA.Linux—死锁(产生死锁的原因、条件和避免方法)[EB/OL]. [2017-06-10]. <https://www.pianshen.com/article/73041793712>. RANNIANZHIXIA.Linux—deadlock (causes, conditions and avoidance methods of deadlock)[EB/OL]. [2017-06-10]. <https://www.pianshen.com/article/73041793712>.
- [7] 郭鹏, 王文焕, 詹荣荣, 等. 基于 Petri 网出现序列的继电保护业务模型[J]. 电力科学与技术学报, 2020, 35(4): 169-175. GUO Peng, WANG Wenhuan, ZHAN Rongrong, et al. Research on model of protection relay procedure based on the Petri net occurrence sequence[J]. Journal of Electric Power Science and Technology, 2020, 35(4): 169-175.
- [8] 罗琨, 时永肖, 李正新, 等. 智能变电站继电保护装置寿命模型及其辨识方法[J]. 智慧电力, 2021, 49(1): 96-101. LUO Kun, SHI Yongxiao, LI Zhengxin, et al. A life model and its identification method for relay protection device in smart station[J]. Smart Power, 2021, 49(1): 96-101.
- [9] 杨胡萍, 黄煌, 何志勤, 等. 基于数据拟合的继保设备失效率调整因子模型[J]. 中国电力, 2021, 54(9): 96-101. YANG Huping, HUANG Huang, HE Zhiqin, et al. Adjustment factor model of relay protection equipment failure rate based on data fitting[J]. Electric Power, 2021, 54(9): 96-101.
- [10] 来智浩, 高钰琛, 翟常营, 等. 基于信号路径跟踪算法与并发分层架构的热过载保护装置自动校验方法研究[J]. 高压电器, 2022, 58(11): 121-127. LAI Zhihao, GAO Yuchen, ZHAI Changying, et al. Research on automatic verification method of thermal overload protection device based on signal path tracking algorithm and concurrent hierarchical architecture[J]. High Voltage Apparatus, 2022, 58(11): 121-127.
- [11] 康健. 基于 ARM 的嵌入式继电保护系统设计与实现[D]. 苏州: 苏州大学, 2008. LIAN Jian. Design and realization of embedded relay protection system based on ARM[D]. Suzhou: Suzhou University, 2008.
- [12] 李雪源, 丁玮, 张拓智. 基于 PowerPC 处理器的嵌入式 Linux 系统构建研究[J]. 航空计算技术, 2020, 50(5): 107-110. LI Xueyuan, DING Wei, ZHANG Tuozhi. Linux system construction based on PowerPC processor[J]. Aeronautical Computing Technique, 2020, 50(5): 107-110.
- [13] 张欢庆, 高丽, 宋承祥. 基于 ARM 的嵌入式 Linux 交叉编译环境的研究与实现[J]. 计算机与数字工程, 2012, 40(2): 151-153. ZHANG Huanqing, GAO Li, SONG Chengxiang. Research and implementation of embedded Linux cross-compiling environment based on ARM[J]. Computer and Digital Engineering, 2012, 40(2): 151-153.
- [14] 余文魁, 刘田豹, 李孜, 等. 基于 i.MX6Q 的低压配电台区智能终端关键模块的设计与实现[J]. 电力信息与通信技术, 2022, 20(4): 41-47. SHE Wenkui, LIU Tianbao, LI Zi, et al. Design and realization of key modules of smart terminal in low-

voltage power distribution area based on i.MX6Q[J]. Electric Power Information and Communication Technology, 2022, 20(4): 41-47.

[15] RIVERW. VxWorksBSP 开发人员指南[M]. 王金刚, 等译. 北京: 清华大学出版社, 2003.

[16] 伍小刚, 沈健, 俞波, 等. 一种基于嵌入式系统的线路保护新平台[J]. 电力系统自动化, 2008, 32(24): 65-69. WU Xiaogang, SHEN Jian, YU Bo, et al. A new platform of line protection based on an embedded real-time operating system[J]. Automation of Electric Power Systems, 2008, 32(24): 65-69.

[17] 仲伟, 李亚锋. 面向中低压系统保护的嵌入式软硬件平台设计[J]. 江苏电机工程, 2010, 29(4): 4-7, 11. ZHONG Wei, LI Yafeng. The design of the software and hardware platform for medium-low voltage relay protection[J]. Jiangsu Electrical Engineering, 2010, 29(4): 4-7, 11.

[18] 古展基. 配电网区域保护与自愈控制系统建模及其通信网络性能分析[D]. 广州: 华南理工大学, 2020. GU Zhanji. Modeling of regional protection and self-healing control system for distribution network and performance analysis of its communication network[D]. Guangzhou: South China University of Technology, 2020.

[19] VOLKER L, CHEN Hao, CHEN Dehui, et al. Towards a standardisation for digital inputs and outputs of protection functions in IEC 60255 series[J]. Protection and Control of Modern Power Systems, 2022, 7(2): 382-392.

[20] 林志超. 自适应配电网拓扑与开关配置的智能分布式保护方法[J]. 电力系统保护与控制, 2021, 49(5): 149-156. LIN Zhichao. Intelligent-distributed distribution network protection strategy for self-adaptive topology and different switch configurations[J]. Power System Protection and Control, 2021, 49(5): 149-156.

[21] 娄为, 沈冰, 裘愉涛, 等. 钻石型配电网的保护与自愈策略分析[J]. 电力系统保护与控制, 2022, 50(23): 161-168. LOU Wei, SHEN Bing, QIU Yutao, et al. Protection and self-healing strategy analysis of a diamond type distribution network[J]. Power System Protection and Control, 2022, 50(23): 161-168.

[22] 李文君, 段登伟, 朱雨, 等. 基于 5G 通信模式下的配电网自愈保护应用[J]. 电力系统保护与控制, 2022, 50(24): 152-159. LI Wenjun, DUAN Dengwei, ZHU Yu, et al. Application of distribution network protection based on a 5G end-to-end communication mode[J]. Power System Protection and Control, 2022, 50(24): 152-159.

[23] 张志丹. 电力系统数据流的建模分析与应用研究[D]. 长沙: 湖南大学, 2015. ZHANG Zhidan. Research on modeling of data flows in power systems and applications[D]. Changsha: Hunan University, 2015.

[24] BOVETD P, CESATIM. 深入理解 LINUX 内核[M]. 3 版. 北京: 中国电力出版社, 2007.

[25] LOVER. Linux 内核设计与实现[M]. 陈莉君, 康华, 张波, 译. 北京: 机械工业出版社, 2004.

[26] HART D, STULTZJ, TSO T. Real-time Linux in real time[J]. IBM Systems Journal, 2008, 47(2): 207-220.

[27] WU ZJ, MC GUIRE N. Porting RT-preempt to Loongson 2F[C] // Eleventh Real-Time Linux Workshop, September 28-30, 2009, Dresden, Germany: 169-174.

[28] FAYYAD-KAZAN H, PERNEEL L, TIMMERMAN M. Linux PREEMPT-RT v2. 6.33 versus v3.6.6: better or worse for real-time applications?[J]. ACM Sigbed Review, 2014, 11(1): 26-31.

[29] 丁泉, 李帅. 智能变电站重采样应用研究及其线性插值法误差分析[J]. 电力系统保护与控制, 2015, 43(23): 132-136. DING Quan, LI Shuai. Application study on resampling in smart substation with error analysis of linear interpolation[J]. Power System Protection and Control, 2015, 43(23): 132-136.

[30] 吕航, 陈军, 杨贵, 等. 基于交换机数据传输延时测量的采样同步方案[J]. 电力系统自动化, 2016, 40(9): 124-128. LÜ Hang, CHEN Jun, YANG Gui, et al. Synchronous sampling method based on measurement of switch data transmission delay[J]. Automation of Electric Power Systems, 2016, 40(9): 124-128.

[31] 史蒂文斯, 拉戈. UNIX 环境高级编程[M]. 3 版. 北京: 人民邮电出版社, 2014.

[32] 王朴, 周晴. 基于龙芯 1E 的嵌入式 Linux 实时性的优化与可靠性设计[J]. 微电子学与计算机, 2019, 36(11): 11-15. WANG Pu, ZHOU Qing. Real-time optimization and reliability realization of embedded Linux based on Loongson1E[J]. Microelectronics & Computer, 2019, 36(11): 11-15.

收稿日期: 2023-02-28; 修回日期: 2023-06-12

作者简介:

古展基(1993—), 男, 通信作者, 硕士研究生, 工程师, 研究方向为电力系统自动化; E-mail: guzhangji@qq.com

谷武健(1985—), 男, 本科, 工程师, 研究方向为继电保护平台技术; E-mail: guwuajian@qq.com

蔡泽祥(1960—), 男, 教授, 博士生导师, 研究方向为电力系统保护与控制. E-mail: eplxh@scut.edu.cn

(编辑 周金梅)