

# HarmonyOS 分布式流转的应用开发研究

龙军,赵冬冬,茅维

(海口经济学院,海南 海口 571127)

**摘要:**随着 HarmonyOS4.0 的发布,鸿蒙多设备协同应用的需求越来越多。HarmonyOS 通过分布式体验架构实现服务流转,文章在研究分布式流转流程的基础上,分析跨端迁移和多端协同两种模式的应用场景、流程和接口函数,最后完成了2种模式应用模型的初步建立、测试和运行。

**关键词:**HarmonyOS;鸿蒙;分布式流转;跨端迁移;多端协同

**中图分类号:**TP311.1 **文献标识码:**A

**文章编号:**1009-3044(2023)35-0050-03

开放科学(资源服务)标识码(OSID):



华为 1+8+N 战略是其全场景智慧化战略在产品层面的体现,通过相互感知将设备进行能力整合成一个超级终端,鸿蒙借助微内核、分布式等技术实现设备全场景化,为用户带来一致、高效的体验<sup>[1]</sup>。2023 年 8 月 HarmonyOS4.0 正式发布,突显元服务、分布式万物互联、大智慧、安全、个性等特点。

目前鸿蒙生态设备达 7 亿台,随着全场景多设备生活方式的延伸,深度融入 AI 技术,车机系统、多模态交互等领域再次突破,提供了更加高效的交互体验。

## 1 鸿蒙多设备协同应用的需求

鸿蒙设备有多种类型和芯片,连接方式有星闪、Wi-Fi、蓝牙,交互方式有触屏、键盘、语音、按钮、手势,屏幕大小、形状、分辨率多样。不同设备有各自应用场景,如手机功能全、交互好,但屏幕小;智慧屏屏幕大、音视频输入输出能力强,但移动性、交互性差;手表有便携、生理数据检测、安全认证等能力,但交互差、屏幕小。

随着越来越多设备的智能化,在实际应用中会遇到要将在 PC 或平板上进行的应用改到手机上继续操作、将手机上的文档发送到智慧屏上操作、在平板上看视频同时在手机上进行交互、同时用手机和智慧屏操作文档、将手机作为手柄与智慧屏配合玩游戏等场景。如何实现多设备间的通信、消息收发、错误处理和性能优化,保证跨设备应用有正确、流畅的体验,解决多设备形态和能力差异,实现多设备并发、协作、互补,这些问题都要研究解决。

## 2 HarmonyOS 分布式体验架构及流转流程

鸿蒙设备通过分布式操作系统实现取长补短和共同协助,服务流转打破了设备界限,通过多设备联动,使元服务可分可合、可流转。

## 2.1 HarmonyOS 分布式体验架构

HarmonyOS 将分布式体验称为分布式流转,有跨端迁移和多端协同两种类型。跨端迁移是用户在设备 A 上发起操作,切换到设备 B 时能适时继续当前操作,A 端应用退出。多端协同是多个设备相互协同,作为一个整体为用户提供比单设备更高效和沉浸的体验<sup>[2]</sup>。



图1 HarmonyOS 分布式体验架构

## 2.2 HarmonyOS 分布式流转流程

已授权设备在使用元服务时,若系统检测到有更优的可流转设备,可通过自动出现推荐气泡完成流转,也可在元服务中内嵌规范的流转图标,通过流转面板手动选择合适的设备进行流转。

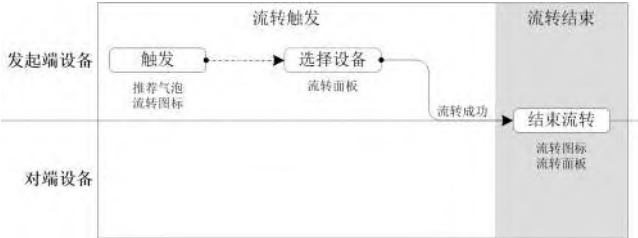


图2 已授权设备间流转流程

未授权设备通过流转图标调出系统的流转面板,选择要流转的未授权设备,此时对端设备显示告知被流转服务信息的卡片,同意后,扫描该设备上显示的多功能码完成设备认证和服务流转。

直接关闭服务或通过流转面板结束流转任务。

## 3 HarmonyOS 的跨端迁移模式

在用户使用设备的过程中,如发生从室内到户外

收稿日期:2023-09-12

基金项目:海口经济学院校级科研项目:基于鸿蒙的分布式应用开发研究(编号:HJKY(ZD)22-03)

作者简介:龙军(1971—),男,海口经济学院副教授,研究方向为计算机应用。

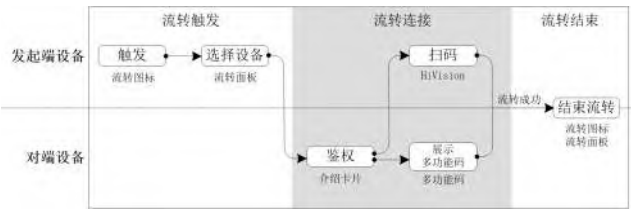


图 3 未授权设备间流转流程

或周围有更合适设备等使用情境变化时,之前使用的设备可能已不适合继续当前的任务,此时,用户可以选择新的设备来继续当前的任务<sup>[3]</sup>,即跨端迁移。迁移后设备要同步原设备任务的状态,且独立运行。

系统根据用户习惯、场景和可用设备,主动推荐最适合设备,用户通过推荐气泡选择完成迁移。应用也可用流转图标作为手动迁移入口,流转面板提供全量可用设备列表,手动选择设备进行迁移。

4 HarmonyOS 的多端协同模式

不同设备的显示、摄像、音频输入输出、交互和传感器等能力存在差异,通过多端协同,根据应用场景需要,将多个设备的优劣势进行匹配,可设计出优秀的多端协同体验<sup>[4]</sup>。

显示协同是应用将界面用多个设备操作同时使用,如把某设备的一个或多个界面的内容分拆到多设备同时显示,或把一个界面中显示和功能操作分拆到多设备操作。

当设备摄像头的拍摄质量、位置、角度不适合当前摄像场景时,通过摄像协同用另一设备摄像头替代或组合使用。

音频输入协同用于间距较远或吵闹的场景,用另一设备的音频输入能力作为补充或话筒来使用。音频输出协同是把某设备上的音频分拆到其他设备或多个设备同时播放同一或不同音频。

交互协同是多个设备共同完成某项应用,如用一设备输入或交互能力帮助另一设备进行操作,以提升效率。

运动、健康类应用常用传感器协同获得如手表等有生理数据检测能力设备的生理或运动数据。

实际设计中,可针对应用所需要的能力进行多设备协同,并将这些能力组合协同,提供更符合场景的整体协同体验。

5 HarmonyOS 跨端迁移的实现

5.1 跨端迁移流程

应用从设备 A 跨端迁移到设备 B 时,设备 A 的应用向流转任务管理服务注册一个流转回调的流转准备。可由系统感知到可用设备 B,主动提供设备信息,用户确认后回调通知应用开始系统推荐流转。或通过流转图标,系统被动为用户提供可交互设备信息,确认选择设备 B 后回调通知应用开始用户手动流转。

设备 A 的应用通过调用分布式任务调度能力,向设备 B 的应用发起跨端迁移。流转中将流转状态上报到流转任务管理服务。

跨端迁移后,设备 A 的应用自行退出。

5.2 接口说明

流转任务管理服务提供注册、解注册、显示设备列表、上报业务状态等接口,通过跨端迁移能力,可实现如文档跨设备编辑、视频跨设备续播等场景<sup>[5]</sup>。

表 1 跨端迁移接口功能介绍

接口名	功能描述
register(String bundleName, ExtraParams params, IContinuationDeviceCallback deviceCallback, RequestCallback requestCallback)	注册并连接到流转任务管理服务,并获取对应的注册 token。参数 bundleName 为本应用包名;params 为系统推荐流转时所需的过滤条件;deviceCallback 设备列表中选择后的回调,返回选择设备 ID;requestCallback 注册请求回调,返回应用注册后的 token
unregister(int token, RequestCallback requestCallback)	用 token 从流转任务管理服务解注册,RequestCallback 的 onResult 回调返回操作是否成功
updateConnectStatus(int token, String deviceId, int status, RequestCallback requestCallback)	通知流转任务管理服务更新当前用户程序的连接状态。token、deviceId 为注册流转任务管理服务的回调;status 为状态;RequestCallback 的 onResult 回调返回操作是否成功
showDeviceList(int token, ExtraParams params, RequestCallback requestCallback)	显示组网内可选择设备列表信息, params 指定设备过滤的条件,用于手动多端协同;token 为注册流转任务管理服务的回调,RequestCallback 的 onResult 回调返回操作是否成功
disconnect()	应用退出时主动调用断开和流转任务管理服务的连接
onConnected(ContinuationDeviceInfo deviceInfo)	从设备列表选择设备时调用,返回设备 ID、设备类型和设备名称
onDisconnected(String deviceId)	当流转任务管理服务断开连接设备时调用
onResult(int result)	与流转任务管理服务交互成功时调用,返回对应的 token
getContinuationRegistrarManager()	获取流转任务管理服务注册服务管理类,包括注册/解注册、更新设备连接状态、显示可选择设备列表等
continueAbility(String deviceId)	把当前 FA 单向流转到同一分布式网络中另一设备
onStartContinuation()	FA 请求迁移后系统回调此方法
onSaveData(Intent Params saveData)	onStartContinuation() 返回 true 时回调,保存必须传递给对端设备用于恢复 FA 状态的数据
onRestoreData(Intent Params restoreData)	发起端设备 FA 保存完数据后,系统再对端设备回调此方法,接收用于恢复 FA 状态的数据
onCompleteContinuation(int result)	对端设备恢复数据完成后,系统再发起端设备回调 FA 的方法,通知应用迁移流程已结束

5.3 模型实现

1) 在 MainAbility 中实现 IAbilityContinuation 接口,并重写方法。

2) 1 在 config.json 中声明跨端迁移访问权限,在 MainAbility 的 onStart() 中申请权限。

```
requestPermissionsFromUser(new String[] {"ohos.permission.DISTRIBUTED_DATASYNC"}, 0);
```



3) 在卡片的流转任务管理服务回调函数的 on-Connected()中获得设备 ID,更新选择设备后的流转状态。

```
DevId = deviceInfo.getDeviceId();
cRMan. updateConnectStatus(aToken, DevId , DeviceConnect-
State.CONNECTED.getState(), null);
```

在 onStart()注册流转任务管理服务。

```
cRMan= getContinuationRegisterManager();
```

在 onStop() 中解注册流转任务管理服务并断开连接。

```
cRMan.unregister(aToken, null);
cRMan.disconnect();
```

#### 4) 功能实现

用参数设置过滤条件，用 `register(BUNDLE_NAME,params,callback,requestCallback)`注册流转任务管理服务。

用参数设置过滤设备类型,用 showDeviceList(aToken,params,null)显示设备列表,选择后在回调中获得设备ID。

用 `continueAbility(DevId)` 将运行的 FA 不可回迁地迁移到目标设备, 实现业务在设备间的无缝迁移。

用 `continueAbilityReversibly(DevId)` 迁移可回迁 FA, 迁移后用 `reverseContinueAbility()` 将已迁移 FA 拉回到本端。

5) 通过 `IAbilityContinuation` 接口实现迁移过程中状态数据的传递和管理,发起端完成迁移后在 `onCompleteContinuation()`回调中用 `terminateAbility()`关闭本端 FA,接收到远端迁移传递数据后,在 `onSaveData()`回调中用 `setParam()`保存状态数据。接收到远端数据时在 `onRestoreData()`回调中用 `getParam()`捕获。

## 6 HarmonyOS 多端协同的实现

## 6.1 多端协同流程

设备 A 的应用和设备 B 的应用进行多端协同业务流程的第一步流转准备向跨端迁移。

设备A的应用通过分布式任务调度的能力,向设备B的应用发起多端协同,流转过程中将流转状态上报到流转任务管理服务。

用户通过设备 A 的流转任务管理界面结束流转, 流转任务管理服务回调通知应用取消流转, 终止和设备 B 的多端协同。流转结束后将流转状态上报到流转任务管理服务, 注销流转回调。

## 6.2 接口说明

多端协同的接口多数同跨端迁移,表2列出部分不同接口。

## 6.3 模型实现

1) ~ 3) 同跨端迁移。

#### 4) 功能实现。

注册流转任务管理服务同上。

在设备列表中选择设备同上。

将目标设备ID作为Intent参数,用startAbility(intent)

表2 多端协同部分接口功能介绍

接口名	功能描述
startAbility(Intent intent)	启动 Intent 指定设备上 FA 和 PA, Intent 包括指定待启动设备的 deviceId、bundleName 和 abilityName
connectAbility(Intent intent, IAbilityConnection conn)	连接 Intent 指定设备上 PA, 连接成功后, 通过在 conn 定义的 onAbilityConnectDone 回调中获取对端 PA 服务代理的 IAbilityConnection 对象
stopAbility(Intent intent)	关闭 Intent 指定的远程 PA
disconnectAbility (IAbilityConnection conn)	断开 IAbilityConnection 对象指定的远程 PA 连接, conn 需要指定为 connectAbility 连接指定设备上 PA 时, 传入的 IAbilityConnection 对象
onAbilityConnectDone (ElementName element, IRemoteObject remote, int resultCode)	当连接 Service 服务成功时调用, 可使用返回的 IRemoteObject 对象与连接的 Service 服务通信

跨设备启动FA/PA,用stopAbility(intent)关闭设备PA。

用 `IAbilityConnection` 创建管理连接关系对象 `mConn`, 跨设备 PA 连接完成后, 返回序列化 `IRemoteObject` 对象 `mProxy`。在本地发起连接侧和对端被连接侧分别实现代理以提供跨设备连接能力。用 `RemoteObject` 类和 `IRemoteBroker` 接口创建对应客户端业务类。创建服务, 实例化客户端, 返回给连接方可调用的代理, PA 收到连接请求时, 将客户端转为代理返给发起侧。将指定目标设备 ID 作为 `Intent` 参数, 用 `connectAbility(intent, mConn)` 实现跨设备连接 PA, 执行对象 `mProxy` 的业务逻辑, 用 `disconnectAbility(mConn)` 断开与远程 PA 连接。

## 7 结束语

跨端迁移实现应用和设备间流转,多端协同实现多设备共同完成应用。通过这两种分布式流转应用模型的建立与运行,实现了相应的流转功能。

随着 HarmonyOS 4.0 的发布,开发者达到 220 万,鸿蒙产品陆续推出,生态更加完善,全场景业务的分布式应用不断拓宽。在华为持续深入研发和开发者的努力下,会出现更多创造性的产品和应用,用科技推动国产产业持续地发展。

### 参考文献:

- [1] 潘炳征. 基于鸿蒙的分布式创作播报应用研究[J]. 价值工程, 2022,41(23):166-168.
- [2] 华为. 分布式体验架构-通用设计基础-指南-设计-HarmonyOS Developer[EB/OL]. [2022-10-22]. <https://developer.harmonyos.com/cn/docs/design/des-guides/architecture-0000001054605984>.
- [3] 刘小芬. 鸿蒙系统架构及应用程序开发研究[J]. 电脑编程技巧与维护, 2021(12):3-5,12.
- [4] 华为. 流转-开发-指南-文档-HarmonyOS Developer[EB/OL]. [2023-09-02]. <https://developer.harmonyos.com/cn/docs/documentation/doc-guides/hop-overview-0000001092995092>.
- [5] 陈美汝. 鸿蒙操作系统应用开发实践[M]. 北京:清华大学出版社, 2021.

【通联编辑：梁书】