**libfwapp设计文档**

1. **描述**

本模块用于实现事件数据的转发，并针对转发失败的数据进行本地存储及后续转发处理。

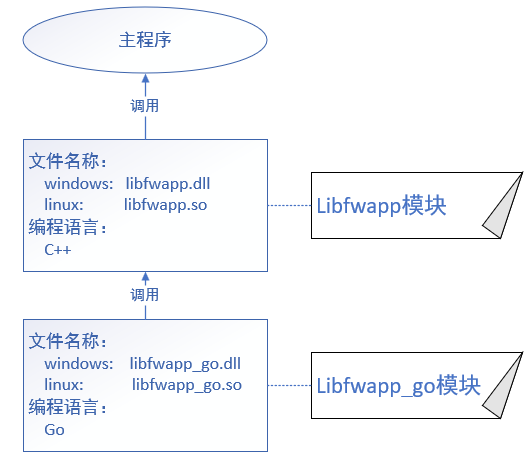
支持配置配置文件配置数据参数，包括日志管理配置、图片存储管理、并发协程数量控制。

1. **总体设计**

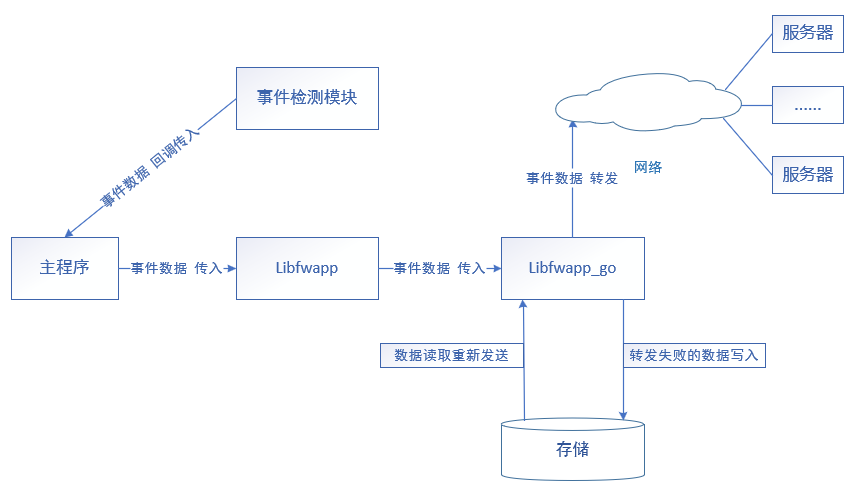
模块包含两部分内容：

**libfwapp模块**用于作为业务模块libfwapp\_go的桥接，被主程序调用，采用C++语言编写。

**libfwapp\_go模块**实现了数据转发、数据管理业务，采用GO语言编写。



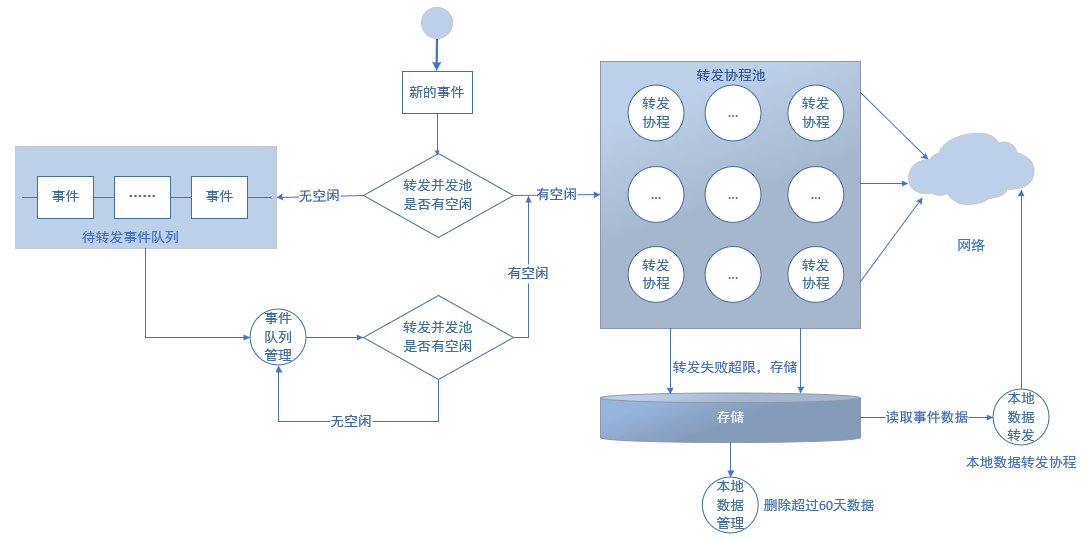
1. **业务说明**



1. 接收外部事件数据，包括事件描述信息、图片数据、转发服务器URL(s)
2. 事件数据转发至目标服务器，目前支持http服务器，可扩展。
3. 事件数据转发失败的数据写入本次磁盘，释放并发资源，用于处理后续事件数据转发。
4. 定时读取本地磁盘存储的事件数据用于转发，转发成功的从本地删除，转发失败等待后续转发。
5. 业务可配置：
6. 日志配置，包括终端回显开关（默认开），日志文件开关（默认关）、日志文件存储路径配置、日志文件保留天数（默认15天）。
7. 数据管理配置：包括转发失败数据的存储路径、最大存储空间数（默认4096MB），内存并发量上限（默认800）
8. **事件接收**

流程：

1. 主程序调用libfwapp.dll/.so库接口，传入事件相关数据
2. libfwapp.dll/.so库调用libfwapp\_go.dll/.so库接口，传递事件数据。libfwapp\_go内部会在内存中复制传入的数据，因此数据传入接口返回后，外部需要实时释放内存空间。
3. libfwapp\_go收到事件数据后，解析url列表，重新封装成本地格式数据后抛给数据管理模块处理，并返回。
4. 事件接收结束。
5. **事件转发**



**5.1 实时事件转发**

libfwapp\_go在初始化后会开启一定数量的并发协程池用于并发处理后续的事件数据，并发数量由配置文件配置，默认800。并发协程池未满时，新收到的数据扔进协程池并发处理。并发协程池满时，数据进入待转发队列。

协程池中的事件转发说明：依次发送至一个或多个目标服务器，发送成功的服务器从本事件的服务器列表中清除。如果全部发送成功，释放本事件数据内存，本协程状态变为空闲等待接收新的数据。如果有发送失败的服务器，则默认进行10次（每次等待5秒总时长）尝试发送。在尝试发送过程中，如果并发池满则此事件停止转发并将数据写入磁盘，释放并发资源。

待发送队列中的时间数据的转发采用后到先发的顺序，确保最新数据的实时转发。

**5.2 历史事件转发**

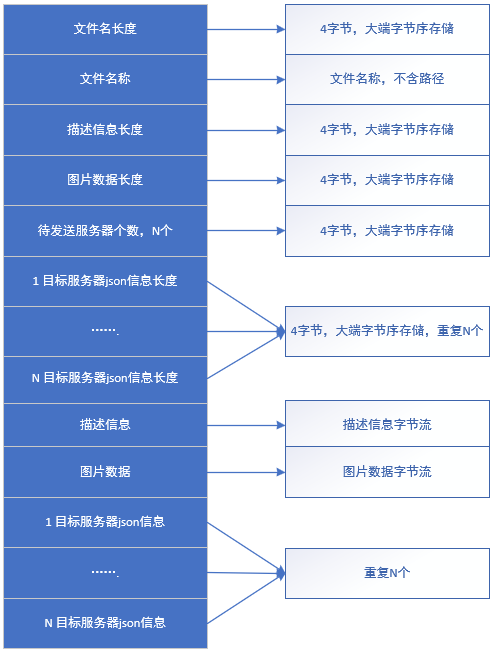
定时从本地存储中读取事件数据，并以此发送至目标服务器。发送成功后删除数据，否则继续发送。

发送顺序采用时间逆序发送，最近的事件数据优先转发。

1. **事件存储**

针对发送失败的数据进行本地保存，保存空间额度由配置文件配置，默认4096MB，超过60天的数据不再转发并进行删除（即使没有超过存储空间额度）。

事件数据存储路径由配置文件配置，单次事件存成一个文件，文件名称采用接收到事件时的UNIX纳秒数作为文件名称，文件结构说明：



说明：

1. 为了便于文件在不同系统下转移，文件结构中的4字节整型数据采用网络模式的大端字节序存储，取出时需要根据本机大小端类型做转换。
2. 存储在文件中的文件名称在取出后，用于和外部的文件名称做对比，用于判断本文件是否为事件数据文件
3. 事件待转发的服务器在存入时会被序列化为json信息，取出后再进行反序列化。
4. **其他**

**7.1 日志管理**

日志管理支持终端回显开关设置，文件写入开关，日志文件存储保留天数配置，存储路径配置。

日志终端显示默认开启，支持彩色回显。

日志文件管理：

1、日志文件每天0点0分0秒自动产生一个新的日志文件。

2、针对一天前的日志文件进行压缩处理，windows采用zip压缩，linux采用tar.gz压缩，压缩后会自动更改压缩文件的修改时间和原文件相同（便于有效地判断是否超过最大保留天数），并删除原文件。

3、超过最大保留天数的日志文件将会被清除。

**7.2 编译**

1、libfwapp\_go模块：

windows: build.bat [debug]

linux：make [DEBUG=1]

2、libfwapp模块：

windows：vs2010

linux: make