**采集代理服务详细设计**

**（KingIOServer代理）**

**北京恒泰博远科技有限公司**

**2018-4-23**

# 

**更新记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **描述** | **作者** |
| 2018年04月23日 | 1.0 | 初始化 | 刘逢缘 |
| 2018年05月22日 | 1.1 | 增加缓存文件格式 | 刘洪文 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[1. 概述 4](#_Toc514766053)

[1.1. 关于本文档 4](#_Toc514766054)

[1.2. 系统简介 4](#_Toc514766055)

[1.3. 系统架构 4](#_Toc514766056)

[2. 系统设计 4](#_Toc514766057)

[2.1. 服务架构 4](#_Toc514766058)

[2.2. 实现流程 5](#_Toc514766059)

[2.3. 发送接收队列结构 9](#_Toc514766060)

[2.4. 缓存文件格式 9](#_Toc514766061)

[2.5. 主要功能模块 9](#_Toc514766062)

# 概述

## 关于本文档

本文档主要遵循《01\_基于分散设备的数据中心概要设计》文档，对3.2章节采集代理服务进行详细设计。

## 系统简介

KingIOServer采集代理服务：通过KingIOServer API接口与KingIOServer通讯，获取KingIOServer设备信息、KingIOServer定义变量信息，并把与KingIOServer通讯的结果转化为系统通用消息格式，按照系统通用消息格式（IOTP协议）与物联网接口服务器通讯。

采集代理服务采集数据可单独工作，采集策略远程配置本地保存，采集数据在无法传输到物联网接口服务器时需要本地缓存，缓存大小可配置。本地缓存策略需支持两种：1、发送失败缓存；2、数据包队列超过制定数量启动文件缓存。消息发送策略需支持：优先发送当前数据包队列数据，当队列空，从存储文件中获取1个待发送数据包。

## 系统架构

KingIOServer采集代理服务遵循《01\_基于分散设备的数据中心概要设计》基本框架，通过多线程任务队列来实现各种功能。

# 系统设计

## 服务架构

服务采用Windows服务程序架构，服务主线程控制服务程序，功能程序由设计文档给出类图，并参照类图开发，开发语言：C#。

采集代理服务类图：



图 1代理服务类图

类图说明：

代理服务功能主要通过四个线程实现，其中windows服务主线程不在四个功能线程之列。

* Dispatch线程：调度线程，启动并加载配置文件，并启动其它线程，当有控制命令和配置命令时，通知相应线程执行具体操作。
* Collect线程：采集线程，执行与KingIOServer的连接和数据通讯，把数据更新到设备和变量实例中。
* PackMsg线程：负责消息打包，消息队列维护，文件缓存维护。
* MsgComm线程：负责与物联网接口服务器采集服务和控制服务通讯 。
* 日志线程：图略，采用公司现有日志线程模块（要求异步写文件）。

## 实现流程

流程图和时序图

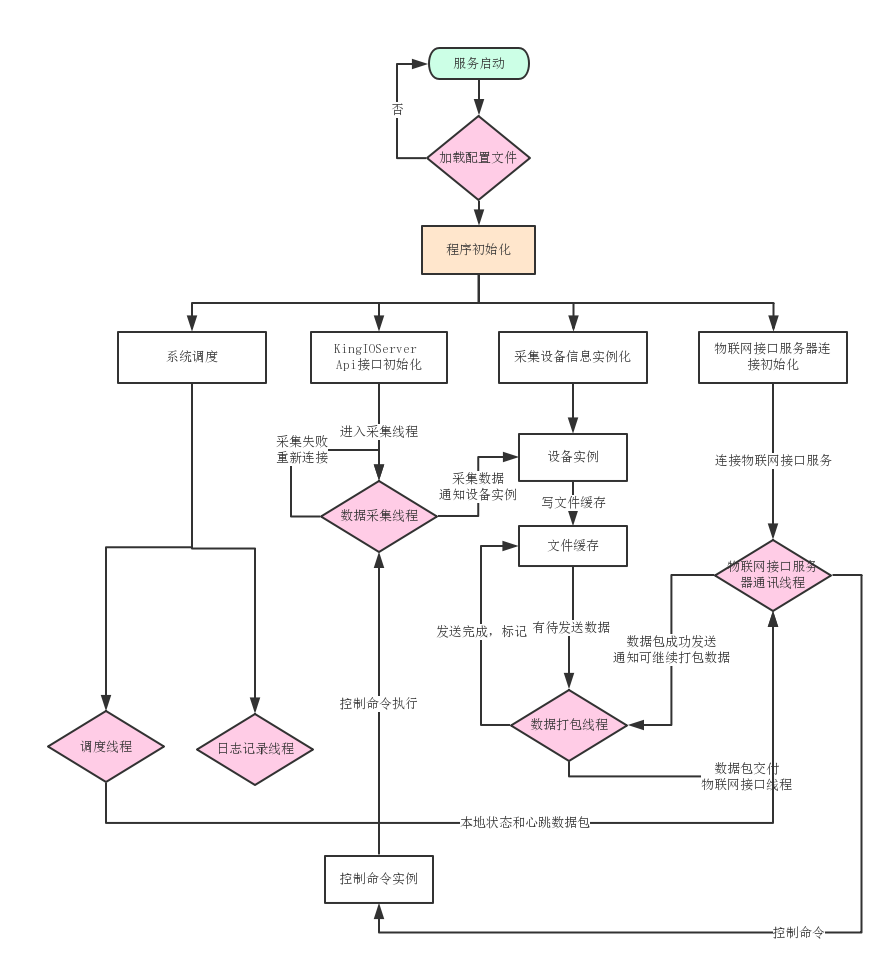


图 2服务主流程图

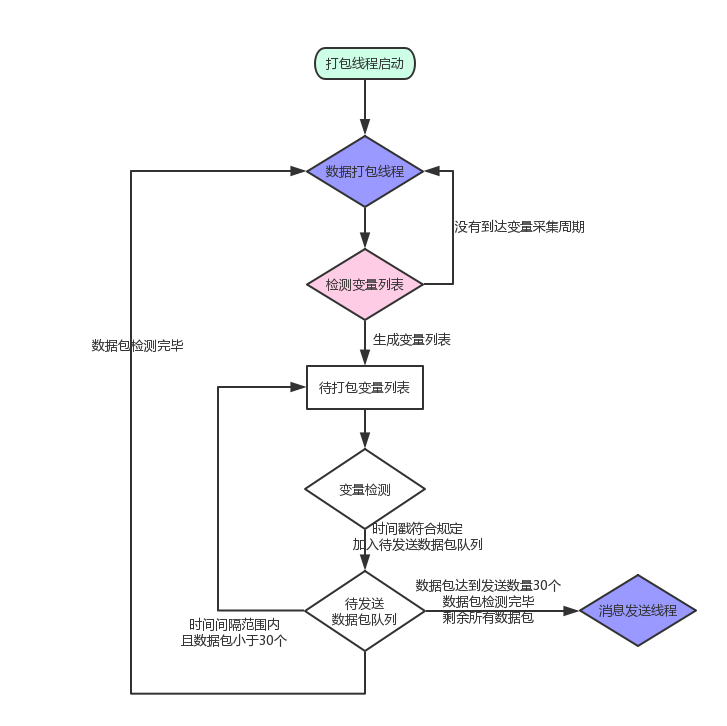


图 3打包过程流程图



图 4采集线程时序图

采集线程工作，循环执行7—12。



图 5打包线程工作时序图

打包线程执行2—6循环。



图 6通讯线程（与物联网接口服务器）

## 发送接收队列结构

IOTP协议体结构（参见《02\_IOTP协议描述.doc》2.协议描述）

协议根据节点授权码可以进行加密或者压缩等操作

## 缓存文件格式

缓存文件分为数据缓存文件和索引缓存文件，索引缓存文件定义数据缓存文件的收发状态和文件位置信息。

* 文件名称：

YYYYMMDD.ind、YYYYMMDD\_X.dat

* 索引缓存文件格式：

X(4字节)pos(4字节)length(4字节)status（4字节）lastsend\_time （8字节），createtime（8字节）。其中X为文件序号，pos为文件偏移位置，status为发送状态（0，发送成功；1未发送），lastsend\_time为最后一次发送时间（0未发送过，其它为最后一次发送时间），createtime创建时间。

* 数据缓存文件格式：

IOTP消息格式，包含消息头和消息体。每个数据缓存文件最多保存大小与保存文件个数可设定，设定值见配置文件定义，超过限定文件会按照时间次序进行删除，删除策略为按天删除，删除策略执行主体为打包线程写操作时检查并触发。缓存文件发送时读取的待发送数据需要更新以下几项数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sequence\_id | 协议序列号 | Integer | 4 | 源端标识唯一ID，循环使用 |
| sequence\_sub\_id | 协议子序列号 | Integer | 4 | 默认0，当有多包连续数据时，从1开始增加，-1表示最后一个数据包 |
| date\_time | 协议包生成时间 | Time | 8 | 时间戳 |

## 主要功能模块

类说明

* Dispatch：调度类，基于线程，加载配置文件，线程启动后启动其它线程，线程退出时，通知其他线程正常退出。当有控制命令和配置命令时，通知相应线程执行具体操作。
* Collect：KingIOServer的连接和数据通讯类，基于线程。线程启动时，根据Dispatch传递的配置创建设备和变量实例；工作过程中，把数据更新到设备和变量实例（访问过程加锁）中。
* PackMsg：消息打包线程类，基于线程。根据Dispatch传递的配置策略加载文件缓存，创建消息队列；工作过程中，维护消息队列（访问过程加锁）和文件缓存。
* MsgComm：物联网接口通讯类，基于线程。根据Dispatch传递的配置策略侦听网络，当有连接时，校验连接属性并与之通讯，传递采集的数据消息和状态消息到物联网采集服务，接收控制消息，传递给Dispatch线程，执行结果传递到物联网控制服务，与物联网采集服务保持心跳（定时发送服务状态）。
* device\_list：设备对象实例类，通过方法访问数据，访问方法中加锁，读取对象需要创建副本传递。
* msgQ：缓存消息队列类。用来缓存消息包，保存消息存储和读取策略，执行消息存储和读取。打包线程打包时，调用put方法，当需要写缓存时写入缓存文件；通讯线程读取后，当队列数量少于一定数量触发打包线程读取操作，从缓存文件读取消息，放入消息队列等待通讯线程获取，文件缓存只由打包线程操作。消息包发送成功，通讯线程调用update\_file\_state()函数更新消息包状态，缓存文件中更新状态，保障数据不被重复发送。