分布式系统

需求分析与概要设计

编制： 王 科 威

审核：

版本： V1.0.0

2020年4月15日

目录

[修改历史 I](#_Toc37939083)

[一 引言 1](#_Toc37939084)

[1.1 编写目的 1](#_Toc37939085)

[1.2 背景 1](#_Toc37939086)

[1.3 定义 1](#_Toc37939087)

[1.4 参考资料 1](#_Toc37939088)

[二 基本约定 1](#_Toc37939089)

[2.1 程序名称 1](#_Toc37939090)

[2.2 目录配置 1](#_Toc37939091)

[三 系统整体设计 1](#_Toc37939092)

[3.1 系统结构图 1](#_Toc37939093)

[3.2 系统结构概述 2](#_Toc37939094)

[3.3 系统功能概述 3](#_Toc37939095)

[3.4 系统运行环境 4](#_Toc37939096)

[四 产品授权 4](#_Toc37939097)

[五 附表 4](#_Toc37939098)

# 修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **说明** | **作者** | **审核** | **日期** |
| V1.0.0.0 | 初稿 | 王科威 |  | 2020-04-15 |

# 引言

## 编写目的

## 背景

## 定义

## 参考资料

# 基本约定

## 程序名称

## 端口号

### 应用组件类（61\*\*\*）

### 流媒体服务类（62\*\*\*）

## 命令行参数

1. 本地服务端监听端口：listen和-l;
2. 上游服务端IP地址：upstream和-u;
3. 上游服务端监听端口：port和-p。

## 目录配置

# 系统整体设计

## 系统结构图



## 系统结构概述

根据终端使用的实际情况，当前系统采用星型拓扑结构完成船舶服务端到陆地服务端间的数据交互，该结构的网络属于集中控制型网络，整个网络由陆地服务端执行集中式通行控制管理，各船舶服务端之间相互不能进行数据交互。

星型拓扑存在如下优点：

* 控制简单。任何一站点只和中央[节点](https://baike.baidu.com/item/%E8%8A%82%E7%82%B9)相连接，因而[介质访问控制方 法](https://baike.baidu.com/item/%E4%BB%8B%E8%B4%A8%E8%AE%BF%E9%97%AE%E6%8E%A7%E5%88%B6%E6%96%B9%E6%B3%95)简单，致使访问协议也十分简单。易于网络监控和管理。
* 故障诊断和隔离容易。中央节点对连接线路可以逐一隔离进行故障检测和定位，单个连接点的故障只影响一个设备，不会影响全网。
* 方便服务。中央节点可以方便地对各个站点提供服务和网络重新配置。

星型拓扑虽然存在很多优点，但也存在以下缺点：

* 中央节点负担重，形成“瓶颈”，一旦发生故障，则全网受影响。
* 各站点的[分布处理](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%A4%84%E7%90%86)能力较低。

在当前系统结构下，所有的船舶服务端各自独立，互不干扰。各个船舶服务端各自负责管理旗下的视频监控设备，各自处理视频监控设备的视频流数据，并对其进行分析和结果输出。所有的输出结果最终汇集到陆地服务端，以提供终端用户进行查询。

## 系统部署图



## 系统部署图概述

### 陆地端

#### Docker私有仓库

Docker私有仓库是生产环境下镜像文件的管理工具，由仓库管理员统一管理，所有通过测试的组件程序通过打包后由管理员上传至仓库。

#### 陆地端后台服务组件

陆地端后台服务组件独立运行于Docker容器中，监听TCP通信的61101端口号与船舶端后台服务组件进行信令交互，监听TCP通信的61102端口号与客户端应用程序进行信令交互。

#### 数据库组件

### 船舶端

#### Watchtower组件

自动从Docker私有仓库更新所有组件的镜像以及容器运行异常的自动重启。

#### 消息分发组件

消息分发组件独立运行于Docker容器中，监听TCP通信的61001端口号与船舶端后台服务组件、AI算法服务组件、报警数据推送组件、海康设备接入组件和大华设备接入组件服务进行信令交互。

#### 船舶端后台服务组件

船舶端后台服务组件独立运行于Docker容器中，主动连接TCP通信的61001端口号与消息分发组件进行信令交互，主动连接TCP通信的61101端口号与陆地端后台服务组件进行信令交互，监听TCP通信的61002端口号与客户端应用程序进行信令交互。

#### AI算法服务组件

AI算法服务组件独立运行于Docker容器中，主动连接TCP通信的61001端口号与消息分发组件进行信令交互，监听TCP通信的62001端口号与客户端应用程序进行信令交互。

#### 报警数据推送组件

报警数据推送组件独立运行于Docker容器中，主动连接TCP通信的61001端口号与消息分发组件进行信令交互。

#### 海康设备接入组件

海康设备接入组件独立运行于Docker容器中，主动连接TCP通信的61001端口号与消息分发组件进行信令交互，监听TCP通信的61003端口号与AI算法服务组件和客户端应用程序进行流媒体数据发送。

#### 大华设备接入组件

大华设备接入组件独立运行于Docker容器中，主动连接TCP通信的61001端口号与消息分发组件进行信令交互，监听TCP通信的61004端口号与AI算法服务组件和客户端应用程序进行流媒体数据发送。

#### 数据库组件

## 网络通信结构图

### 

### 通信结构说明

## 系统功能概述

### 组件管理

#### 注册/注销

船舶端消息分发组件按照启动项参数的配置主动向陆地端消息分发组件发送注册请求，陆地端消息分发组件生成UUID作为应答返回。船舶端消息分发组件首次注册成功后必须把UUID保存下来，以便重启后能使用正确的UUID进行注册。

船舶端组件程序按照启动项的配置主动向船舶端消息分发组件发送注册请求，消息分发组件生成UUID作为应答返回，在程序退出时主动向消息分发组件发送注销请求。组件程序首次注册成功后必须把UUID保存下来，以便重启后能使用正确的UUID进行注册。

各类组件在注册成功后启动心跳，失败则间隔30秒重新注册。所有重复注册成功的事件都视为心跳。

#### 心跳检测

心跳检测功能只在消息分发组件上实现。在组件注册成功后连续3次（90秒）检测到无时间更新则视为组件离线。如果消息分发组件在判断组件离线后再次收到组件的心跳请求则视为组件重新上线。

#### 时序图



### 设备分配

作为视频源输入端设备，系统支持包括海康DVR/NVR、海康IPC、大华DVR/NVR和大华IPC等安防视频监控设备。设备所对应的摄像机通道（包括模拟和数字通道）由系统从设备中获取。设备分配包括增、删和改操作。已增加的设备在系统启动时将自动分配到海康设备接入组件或大华设备接入组件。时序图如下：



### 实时视频流分配

实时视频流的分配包括打开和关闭操作。打开操作由用户通过客户端启用摄像机通道至少一种AI算法触发；关闭操作由用户通过客户端停用摄像机通道所有AI算法触发。时序图如下：



### 视频解码

为能向人工智能算法提供可用的视频帧数据，系统在接收到视频监控采集设备发送的视频流数据后，使用开源解码库将其解码。由于不同的采集设备厂商SDK提供的解码数据格式有所不同，在解码后需要将帧数据统一转换为一种格式；除此之外，视频解码方式也有所不同。

当前系统结构设计中包含的视频数据解码分为以下2种：

* CPU解码
* GPU解码

其中，CPU解码时首选，选择它的原因是避免和人工智能算法对GPU的使用产生冲突。但最终选择依然是GPU解码。

### 图片编码

在当前系统中需要将解码后的视频帧数据进行重新编码，并将编码数据用于网络传输，所以当前系统使用开源解码库对其进行编码。当前系统结构下采用的是CPU编码。但最终依然选择GPU编码。

### AI算法运算

AI算法从功能上分为以下几种：

* 安全帽检测
* 打手机检测
* 疑似睡觉检测
* 打架行为检测
* 人脸识别检测

在当前系统结构下，人工智能算法采用动态库的形式交由上层应用进行调用，每一个算法对应一路视频流数据。在计算机硬件条件支持的情况下，人工智能算法是可以同时运行多个不同的实例的，各个实例也相互没有依赖关系。

### 报警推送

每一帧视频数据被送入人工智能算法进行运算后，都会产生一个运算结果。在满足一定阈值或条件时，运算结果将被上层应用判定为一个报警。与此同时，上层应用将对此报警的所有数据进行数据封装，再将其进行网络传输。最终由用户终端的上层应用对数据进行解析和展示。时序图如下：



### 自动更新

在当前系统结构下的每一个船舶服务端应用程序都可以通过陆地服务端进行远程自动更新。当更新包发布到陆地服务端后，船舶服务端都将自动对其进行下载和更新。

### 运行监视

为保证各个应用组件的有效运行，每个船舶服务端上都有一个监视组件，该组件负责对其所在服务端上所有的上层应用进行监视。当上层应用出现异常时，将自动重启上层应用，以保证系统的正常工作。

## 系统运行环境

系统运行环境分为硬件和软件环境，概述如下：

* 硬件环境。计算机服务器包含16核心，48线程的CPU，64G内存和一张以上16G的NVIDIA显卡。
* 软件环境。所有组件或模块都运行在64位Windows操作系统下。

# 产品授权

# 附表