**语音算法集成研发项目**

**详细设计说明书**

2021年09月

**文件使用说明**

|  |  |
| --- | --- |
| 时间及频率 | 本文件适用于“设计开发”期进行详细设计时使用，是阶段交付物之一。 |
| 人员及流程 | 1. 本文件由供应商相关设计人员负责编写； 2. 编写完成后，由建设单位项目经理提交PMO方案管控经理评审。 |
| 备注 |  |

**版本控制信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称 |  | | |
| 文档编号 |  | | |
| 版本 | 变更概要 | 修编时间 | 状态 |
| 0.1 | 初稿 |  | 完成 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | | | |
| 角色 | 人员 | | |
| 编写 |  | | |
| 批准 |  | | |

目 录

[1. 引言 5](#_Toc16970)

[1.1. 编写目的 5](#_Toc8036)

[1.2. 背景 5](#_Toc20291)

[1.3. 定义 5](#_Toc3990)

[1.4. 参考资料 5](#_Toc11900)

[2. 系统的结构 5](#_Toc795)

[3. 能力接口模块设计说明 7](#_Toc15993)

[3.1. 模块描述 7](#_Toc14620)

[3.2. 功能 7](#_Toc31146)

[3.3. 性能 8](#_Toc17038)

[3.4. 输入、输出项 8](#_Toc13418)

[3.4.1. 会话开始输入、输出数据 8](#_Toc29869)

[3.4.2. 会话交互输入、输出数据 8](#_Toc16398)

[3.4.3. 获取会话结果输入、输出数据 9](#_Toc4192)

[3.4.4. 会话结束输入、输出参数 9](#_Toc17212)

[3.5. 流程逻辑 9](#_Toc19877)

[3.6. 接口 10](#_Toc23072)

[3.6.1. 能力接口会话开始输出、输入参数 10](#_Toc13177)

[3.6.2. 会话交互输出、输入数据 11](#_Toc25924)

[3.6.3. 获取会话结果输出、输入数据 11](#_Toc10194)

[3.6.4. 会话结束输出、输入参数 12](#_Toc25610)

[3.7. 存储分配 13](#_Toc16555)

[3.8. 注释设计 13](#_Toc24606)

[3.9. 限制条件 13](#_Toc26163)

[3.10. 测试计划 13](#_Toc22876)

[3.10.1. 测试人员和设备 13](#_Toc6037)

[3.10.2. 进度安排 13](#_Toc24060)

[3.10.3. 测试用例 14](#_Toc28415)

[3.11. 尚未解决的问题 15](#_Toc6401)

[4. 识别模块设计说明 15](#_Toc9928)

[4.1. 模块描述 15](#_Toc12606)

[4.2. 功能 16](#_Toc12521)

[4.2.1. 功能设计 16](#_Toc25970)

[4.3. 性能 17](#_Toc15145)

[4.4. 设计方法（算法） 17](#_Toc30275)

[4.5. 流程逻辑 18](#_Toc3491)

[4.5.1. 主要函数功能介绍 19](#_Toc13609)

[4.6. 接口 20](#_Toc24215)

[4.6.1. 会话开始输入、输出数据 20](#_Toc1919)

[4.6.2. 会话交互输入、输出数据 22](#_Toc19873)

[4.6.3. 获取会话结果输入、输出数据 24](#_Toc26457)

[4.6.4. 会话结束输入、输出参数 25](#_Toc2509)

[4.7. 存储分配 26](#_Toc30670)

[4.8. 注释设计 26](#_Toc287)

[4.9. 限制条件 27](#_Toc12508)

[4.10. 测试计划 27](#_Toc11657)

[4.10.1. 测试人员和设备 27](#_Toc1800)

[4.10.2. 进度安排 27](#_Toc29285)

[4.10.3. 测试用例 28](#_Toc21868)

[4.11. 尚未解决的问题 30](#_Toc22254)

# **引言**

## **编写目的**

本文档主要用于为实现系统功能而进行的系统详细设计说明，具体描述了系统包含的的软件模块的实现流程、功能、接口、数据结构等内容，供项目组开发人员和软件维护人员阅读。

## **背景**

* 系统名称：在线识别系统
* 项目提出者：南方电网数字研究院
* 开发者：北京中科信利技术有限公司
* 用户：南方电网智能语音导航

## **定义**

语音识别（Automatic Speech Recognition），简称（ASR）。

## **参考资料**

《国家标准软件开发文档规范》。

# **系统的结构**

在线识别系统软件架构图：



在线识别系统物理架构图：



# **能力接口模块设计说明**

## **模块描述**

能力接口模块用来接收负载均衡服务器转发的请求，并将请求数据进行封装转发。

## **功能**

能力接口负责接收nginx转发过来的请求数据，判断数据的有效性及转换数据格式为识别服务需要的格式。

首先，能力接口需要判断请求是否是新会话的开始，新会话的开始，能力接口将会话信息存放在redis数据库中，等待识别服务获取会话信息。

然后，能力接口从redis数据库中获取是哪台识别服务来处理后续的识别请求，能力接口将接收到的语音数据发送到指定的识别服务器。

## **性能**

软件环境要求：

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 版本 |
| 操作系统 | Centos | 7.0以上 |
| Java运行环境 | Jdk | 1.8 |
| 能力接口 | Tomcat | 9.0.35 |

硬件环境要求：

|  |  |
| --- | --- |
| 分类 | 配置 |
| Cpu | 4 |
| 内存 | 10G |
| 磁盘 | 20G |
| 带宽 | 20M |

## **输入、输出项**

能力接口模块包含四个交互流程，会话开始（ssb）、发送语音（auw）、获取识别结果（grs）和会话结束（sse）。

### 会话开始输入、输出数据

详见接口文档《北京中科信利-在线语音识别接口开发文档-v1.15.pdf》

### 会话交互输入、输出数据

详见接口文档《北京中科信利-在线语音识别接口开发文档-v1.15.pdf》

### 获取会话结果输入、输出数据

详见接口文档《北京中科信利-在线语音识别接口开发文档-v1.15.pdf》

### 会话结束输入、输出参数

详见接口文档《北京中科信利-在线语音识别接口开发文档-v1.15.pdf》

## **流程逻辑**

能力接口数据逻辑流程图如下所示：



## **接口**

说明：说明本模块与其它相关模块间的逻辑连接方式，说明涉及到的参数传递方式。

能力接口和识别引擎交互方式为socket通信，数据格式为json字符串。

逻辑图如下：



具体交互数据如下。

### 能力接口会话开始输出、输入参数

输出参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| aue | 字符串 | 非编码原始语音 |
| auf | 字符串 | 语音采样率 |
| cmd | 字符串 | 会话状态 |
| sid | 字符串 | 会话Id |

输入参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| ret | 整型 | 响应状态 |
| sid | 字符串 | 会话Id |
| addr | 字符串 | 识别服务地址信息 |

### 会话交互输出、输入数据

输出参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| data | 字符串 | Base64加密的语音数据 |
| cmd | 字符串 | 会话状态 |
| syncid | 字符串 | 语音包序号 |
| sid | 字符串 | 会话Id |

输入参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| sid | 字符串 | 会话Id |
| recStatus | 字符串 | 识别状态 |
| engine\_name | 字符串 | 引擎地址 |
| result | json字符串 | 识别结果 |
| bg | 字符串 | 检测到有效语音 |
| ed | 字符串 | 检测到有效语音尾点 |

### 获取会话结果输出、输入数据

输出参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| cmd | 字符串 | 会话状态 |
| syncid | 字符串 | 序号 |
| sid | 字符串 | 会话Id |

输入参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| sid | 字符串 | 会话Id |
| recStatus | 字符串 | 识别状态 |
| engine\_name | 字符串 | 引擎地址 |
| result | json字符串 | 识别结果 |
| bg | 字符串 | 检测到有效语音 |
| ed | 字符串 | 检测到有效语音尾点 |

### 会话结束输出、输入参数

输出参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| cmd | 字符串 | 会话状态 |
| sid | 字符串 | 会话Id |

输入参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| ret | 整型 | 会话结束状态 |
| sid | 字符串 | 会话Id |

## **存储分配**

本模块对存储分配要求不高，仅存储识别的会话信息。

## **注释设计**

主要函数功能说明，输入、输出参数解释说明。

## **限制条件**

服务运行环境：限于centos操作系统；

技术要求：java

调用请求：模块服务的调用必须按照接口文档进行参数传递。

## **测试计划**

### 测试人员和设备

A．人员：

       测试人员：

       编程人员：

B．设备：安装有centos7操作系统的服务器2台

C. 测试语音：8k16bit和16k16bit各一条

### 进度安排

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试阶段 | 测试任务 | 工作量估计 | 人员分配 | 起止时间 |
| 第一阶段功能测试 | 1.服务启动验证  2.接受数据验证  3.8k16bit pcm语音验证  4.16k16bit pcm语音验证  5.单句识别验证  6.实时识别验证  7.日志级别验证  是否能正确实现其功能，是否有操作错误 | 1日 | 测试组人员 |  |
| 第二阶段异常测试 | 1.少传或多传参数  2.错传参数 | 1日 | 测试组人员 |  |

### 测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试功能 | 期望结果 | 测试结果 |
| 1 | 服务启动验证 | 正确配置服务参数，启动服务，CPU、内存满足情况下，服务能够正常启动。 |  |
| 2 | 接受数据验证 | 请求正确调用情况下，能力接口能够收到请求，并且Redis、Mysql连接正常。 |  |
| 3 | 8k16bit pcm语音验证 | 参数传递正确情况下，输出识别结果。 |  |
| 4 | 16k16bit pcm语音验证 | 参数传递正确情况下，输出识别结果。 |  |
| 5 | 单句识别验证 | 参数传递正确情况下，输出识别结果。 |  |
| 6 | 实时识别验证 | 参数传递正确情况下，输出识别结果。 |  |
| 7 | 日志级别验证 | 调整不同的日志级别，日志输出内容不同 |  |
| 8. | 减少传递参数 | 识别不正常 |  |
| 9 | 增加传递参数 | 识别不正常 |  |

## **尚未解决的问题**

无

# **识别模块设计说明**

## **模块描述**

识别模块接收语音流，识别语音，返回识别文本。

## **功能**

识别服务负责接收能力结果发送过来的语音，返回识别出的文字结果；首先，识别服务从redis服务中获取会话信息，成功后将本服务器信息写入redis数据库中，然后，能力接口从redis数据库中获取服务器信息，将数据发送到识别服务。

### 功能设计

识别模块包含连接redis、识别、更新引擎并发信息、设置引擎线数和识别结果写入redis功能；

连接redis：首先，识别服务需要重redis中读取会话信息，写入自己的ip和端口信息，然后才能接收到语音数据流，连接redis是识别模块的重要功能，识别模块中的多数功能都和redis有直接的关系。

连接redis在识别模块中的作用如下图所示：



识别：识别服务接收到语音流后，将语音数据通过调用引擎进行识别，获取识别结果。

更新引擎并发信息：识别服务会定时的将本服务的并发信息更新到redis中，便于后期通过redis查看服务的并发量，进行扩容或减少并发设置。

识别结果写入redis：该功能主要是防止比较长的识别结果第三方调用端无法完全接收时，可以通过读取redis的形式获取识别结果。

## **性能**

软件环境要求：

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 版本 |
| 操作系统 | Centos | 7.0以上 |
| 识别引擎 | Decoder | 3.10.9、3.9.3 |

硬件环境要求：

|  |  |
| --- | --- |
| 分类 | 配置 |
| Cpu | 10 |
| 内存 | 20G |
| 磁盘 | 100G |
| 带宽 | 20M |

## **设计方法（算法）**

**并发设计**：软件设计使用线程池的方式，通过读取配置文件线程数量一次启动所有需要的线程，线程数配置参考机器配置，线数=cpu核数\*2；

**网络设计**：程序通过单独的线程池接收、发送数据，接收到的数据放到数据队列中，数据处理线程从队列中读取数据进行识别处理，然后将识别出的结果放入到对应的结果队列中，网络线程读取结果队列发送数据。

**任务线程池**、**网络线程池**、**任务队列**和**结果队列**之间的交互关系如下图所示：



## **流程逻辑**

识别模块数据逻辑流程图如下所示：



### 主要函数功能介绍

#### 连接redis函数

void redis\_connect(acl::redis\_client\_cluster &client\_cluster)

函数功能：该函数主要实现连接redis，作为通用的调用方法，供其他功能调用。

#### 获取会话函数

**int processSsb（**asr\_var\_t &threadVar, asr\_var\_temp\_t &threadVarTemp,

**void** \*parameter, redis\_param &rparam, std::string &bitRate**）**

函数功能：通过调用改函数，识别服务的空闲线程获取新的会话信息，然后接收语音数据进行识别。

#### 网络交互函数

**void** **CallBackFunc\_Server**(std::string &out\_action, std::string &out\_msg, **const** std::string &\_in\_action, **const** std::string &\_in\_msg)

函数功能：该函数主要接收能力接口端发送的请求数据，并将识别后的数据返回给能力接口。

#### 逻辑控制函数

**static** **void** \***DecodeProcThread**(**void** \*parameter)

函数功能：该函数主要实现数据处理工程的逻辑控制功能；

#### 数据队列读取函数

**int** **getRequest**(redis\_param &rparam,**int** num)

函数功能：该函数的功能为从当前会话的数据队列中读取语音数据，进行后续的处理。

#### 并发线数设置函数

**void** **UpdateOnlineInfo**()

函数功能：该函数主要功能是从reids中读取对当前识别服务设置的线程数，调整当前服务的最大任务处理线数。

#### 线程信息通知函数

**void** **sendOnlineInfo**()

函数功能：该函数功能为定时的将服务的线数信息写入reids中。

## **接口**

识别引擎和能力接口交互方式为socket通信，数据格式为json字符串。

逻辑图如下：



### 会话开始输入、输出数据

会话开始输入数据：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| aue | 字符串 | 非编码原始语音 |
| auf | 字符串 | 语音采样率 |
| cmd | 字符串 | 会话状态 |
| sid | 字符串 | 会话Id |

如下实例：

{  
    **"request"**:{  
        **"aue"**:**"raw"**,  
        **"auf"**:**"audio/L16;rate=8000"**,  
        **"cmd"**:**"ssb"**  
    },  
    **"sid"**:**"16e374f04d8348e487dab492552537a8"**  
}

会话开始输出数据：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| ret | 整型 | 响应状态 |
| sid | 字符串 | 会话Id |
| addr | 字符串 | 识别服务地址信息 |

**实例如下：**

{  
    **"response"**:{  
        **"ret"**:**0**  
    },  
    **"sid"**:**"16e374f04d8348e487dab492552537a8"**,  
    **"addr"**:**"192.168.0.52:10200"**  
}

### 会话交互输入、输出数据

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| data | 字符串 | Base64加密的语音数据 |
| cmd | 字符串 | 会话状态 |
| syncid | 字符串 | 语音包序号 |
| sid | 字符串 | 会话Id |

实例如下：

{  
    **"request"**:{  
        **"data"**:**""**,  
        **"cmd"**:**"auw"**,  
        **"syncid"**:**"1"**  
    },  
    **"sid"**:**"16e374f04d8348e487dab492552537a8"**  
}

输出数据：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| sid | 字符串 | 会话Id |
| recStatus | 字符串 | 识别状态 |
| engine\_name | 字符串 | 引擎地址 |
| result | json字符串 | 识别结果 |
| bg | 字符串 | 检测到有效语音 |
| ed | 字符串 | 检测到有效语音尾点 |

**实例如下：**

<?xml version="1.0" encoding="utf-8"?>

<results>

<sid>**16e374f04d8348e487dab492552537a8**</sid>

<recStatus>5</recStatus>

<engine\_name>192.168.0.52</engine\_name>

<result>[{ "text": "你好。"]</result>

<bg>1</bg>

<ed>0</ed>

</results>

### 获取会话结果输入、输出数据

输入参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| cmd | 字符串 | 会话状态 |
| syncid | 字符串 | 序号 |
| sid | 字符串 | 会话Id |

**实例如下：**

{  
    **"request"**:{  
        **"cmd"**:**"grs"**,  
        **"syncid"**:**"1306"**  
    },  
    **"sid"**:**"16e374f04d8348e487dab492552537a8"**  
}

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| sid | 字符串 | 会话Id |
| recStatus | 字符串 | 识别状态 |
| engine\_name | 字符串 | 引擎地址 |
| result | json字符串 | 识别结果 |
| bg | 字符串 | 检测到有效语音 |
| ed | 字符串 | 检测到有效语音尾点 |

输出参数：

**实例如下：**

<?xml version="1.0" encoding="utf-8"?>

<results>

<sid>**16e374f04d8348e487dab492552537a8**</sid>

<recStatus>5</recStatus>

<engine\_name>192.168.0.52</engine\_name>

<result>[{ "text": "你好。", "phoneme": "你好。", "segtime": "0.31 0.86", "score": 1.0 }]</result>

<bg>1</bg>

<ed>1</ed>

</results>

### 会话结束输入、输出参数

输入参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| cmd | 字符串 | 会话状态 |
| sid | 字符串 | 会话Id |

实例如下：

{  
    **"request"**:{  
        **"cmd"**:**"sse"**  
    },  
    **"sid"**:**"16e374f04d8348e487dab492552537a8"**  
}

输出参数：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| ret | 整型 | 会话结束状态 |
| sid | 字符串 | 会话Id |

**实例如下：**

{  
    **"ret"**:0,  
    **"sid"**:**"16e374f04d8348e487dab492552537a8"**  
}

## **存储分配**

本模块对存储分配要求不高，仅存储识别的会话信息及识别结果信息。

## **注释设计**

主要函数功能说明，输入、输出参数解释说明。

## **限制条件**

服务运行环境：限于centos操作系统；

技术要求：c/c++

## **测试计划**

### 测试人员和设备

A．人员：

       测试人员：

       编程人员：

B．设备：安装有centos7操作系统的服务器2台

C. 测试语音：8k16bit和16k16bit各一条

### 进度安排

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试阶段 | 测试任务 | 工作量估计 | 人员分配 | 起止时间 |
| 第一阶段功能测试 | 1.服务启动验证  2. 8k16bit pcm语音验证  3. 16k16bit pcm语音验证  4. 保存语音  5. 关闭保存语音  6. 保存识别结果  7. 关闭保存识别结果  8. 检查数据包序号  9. 不检查数据包序号  10. 单句识别  11.实时识别  12. 实时识别，配置不同返回结果形式  13. 日志级别  14. 日志文件大小  是否能正确实现其功能，是否有操作错误 | 1日 | 测试组人员 |  |
| 第二阶段异常测试 | 1.少传或多传参数  2. 错传参数 | 1日 | 测试组人员 |  |

### 测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试功能 | 期望结果 | 测试结果 |
| 1 | 服务启动验证 | 正确配置服务参数，启动服务，CPU、内存满足情况下，服务能够正常启动。 |  |
| 2 | 8k16bit pcm语音 | 参数传递正确情况下，输出识别结果。 |  |
| 3 | 16k16bit pcm语音 | 参数传递正确情况下，输出识别结果。 |  |
| 4 | 保存语音 | 配置保存语音参数，保存语音正常。 |  |
| 5 | 关闭保存语音 | 配置不保存语音参数，不保存语音。 |  |
| 6 | 保存识别结果 | 配置保存识别结果参数，保存识别结果正常。 |  |
| 7 | 关闭保存识别结果 | 配置不保存识别结果参数，不保存识别结果 |  |
| 8 | 检查数据包序号 | 配置检查数据包序号，数据包乱序返回错误信息 |  |
| 9 | 不检查数据包序号 | 配置不检查数据包序号，数据包乱序，不返回错误信息 |  |
| 10 | 单句识别 | 参数传递正确情况下，输出识别结果。 |  |
| 11 | 实时识别 | 参数传递正确情况下，输出识别结果。 |  |
| 12 | 实时识别，配置不同返回结果 | 返回不同的结果形式 |  |
| 13 | 日志级别 | 调整不同的日志级别，日志输出内容不同 |  |
| 14 | 日志文件大小 | 调整日志文件大小，保存日志文件大小不同 |  |

## **尚未解决的问题**

无