

离线语音识别系统 3.1 部署手册

2020 年 3 月

改版记录

版本	发布日期	描述	作者	备注
1.0.0	2017-09-08	编写初稿	吴愁、肖素杰	
1.0.1	2017-09-12	删除冗余内容	吴愁	
1.2	2017-09-22	删除数据库部分、统一功能模块一致性	文俊、吴愁	
1.3	2017-12-22	系统部署方案	张朋	
1.4	2018-09-29	添加情绪检测描述	胡俊锋	
1.5	2019-01-23	添加部署可配置说明	胡俊锋	
1.6	2020-03-24	添加系统架构说明	王超锋	
1.7	2020-06-04	增加 mysql 安装说明	牛彭彭	
1.8	2021-01-08	增加识别模块授权申请说明	王超锋	

目录

1 离线转写系统简介.....	4
1.1 系统功能.....	4
1.2 架构说明.....	4
1.3 服务器软硬件要求.....	6
1.4 系统安装包目录结构.....	7
2 JAVA 能力接口、REDIS 服务配置文件介绍.....	12
2.1 JDBC 配置：	12
2.2 日志配置.....	13
2.3 redis 配置.....	13
3 离线系统配置文件介绍.....	14
3.1 目录结构说明.....	14
3.2 安装软件工具.....	15
3.3 各服务端和总控端配置文件.....	16
4 系统安装与配置.....	17
4.1 分布式部署说明.....	17
4.2 系统安装.....	17
4.3 系统配置.....	21
5 系统运行、关闭与迁移.....	23
5.1 启动系统.....	23
5.2 关闭系统.....	23
5.3 服务器迁移.....	23

6 判断部署是否正确.....	23
6.1.1 检查 TOMCAT 服务是否正常.....	23
6.2 检查系统是否正常启动.....	24
6.3 检查系统是否运行正常.....	26
6.4 查看 log 信息.....	26

1 离线转写系统简介

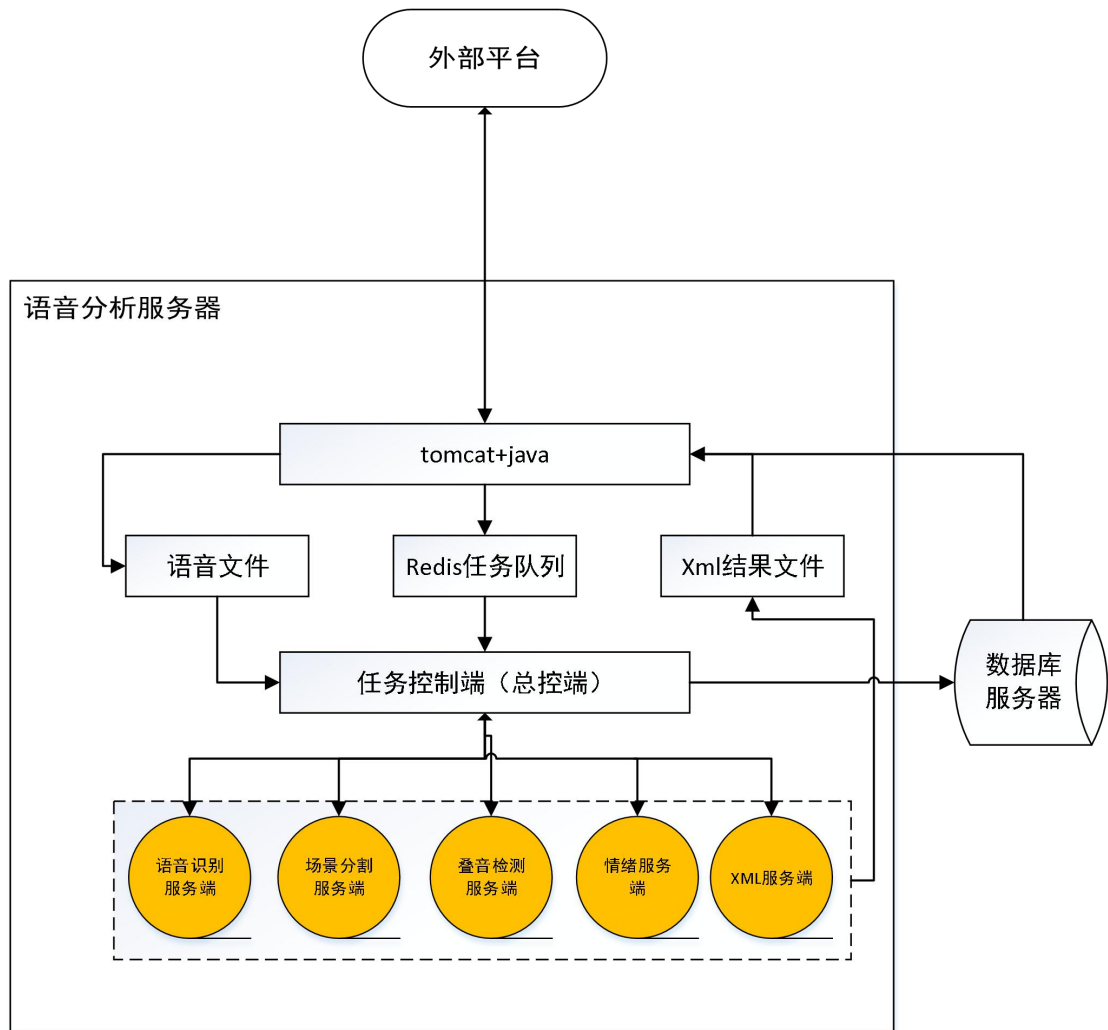
1.1 系统功能

本系统对语音进行全文转写、场景分割、叠音检测、情绪检测、静音检测、XML 服务等。系统主要针对客服语音进行分析处理，得到的一系列的分析结果可供客户进行数据分析，挖掘和质检之类的后续工作。

1.2 架构说明

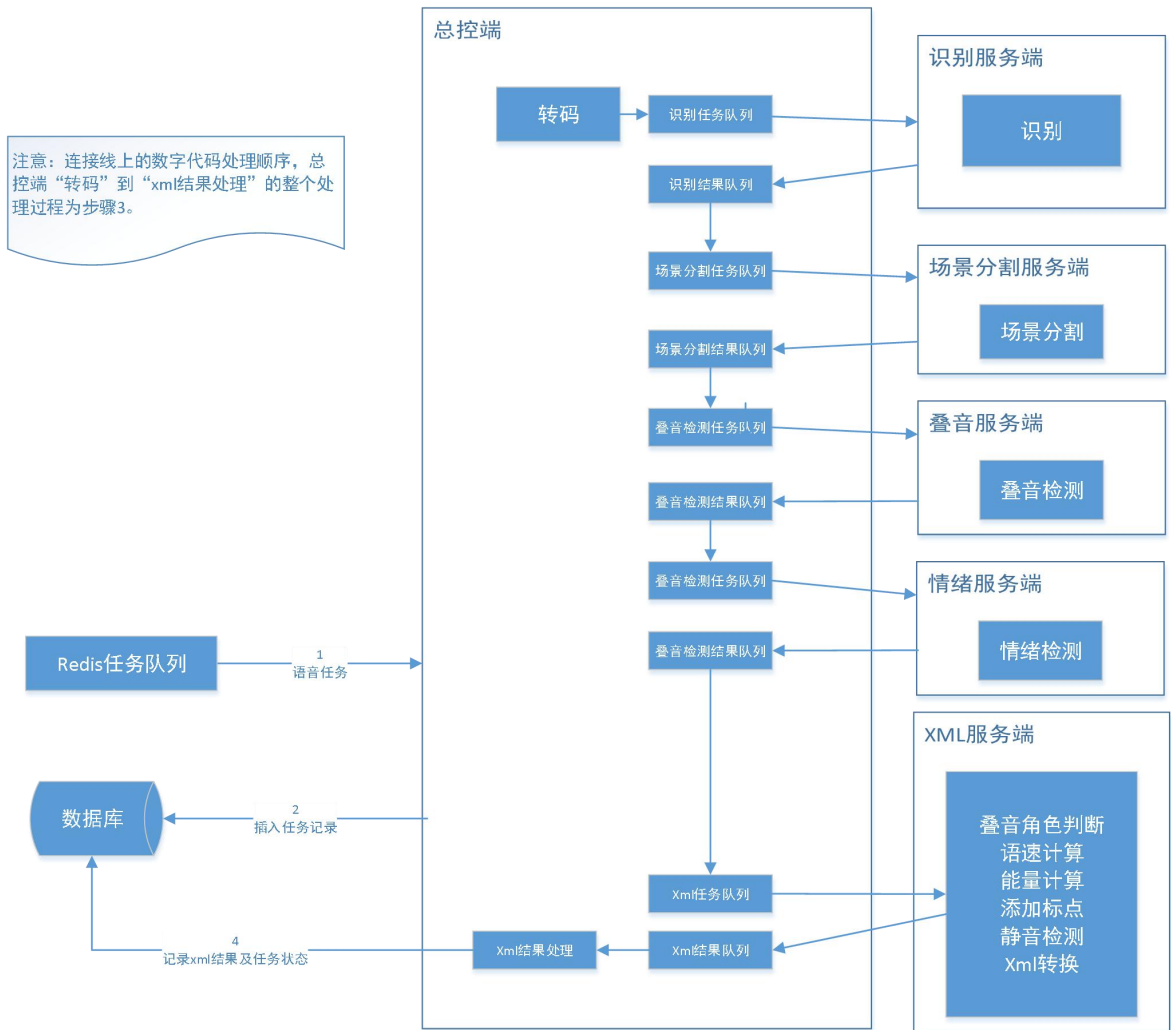
1.2.1 整体架构

离线分析系统整体架构图如下所示：



语音分析服务器上通过 tomcat+java 模块来接收请求，落地语音文件，同时将请求任务推送至 redis 任务队列；同时，总控端从 redis 任务队列中获取任务，并控制任务按“语音识别服务端→场景分割服务端→叠音检测服务端→情绪检测服务端→XML 服务端”的顺序进行处理，将处理结果记录到mysql 数据库中；而后 tomcat+java 模块通过访问数据库来获取识别结果，如果识别成功读取对应 xml 文件内容，并将识别结果返回。

1.2.2 整体数据流图



1.3 服务器软硬件要求

1.3.1 操作系统

操作系统：linux centos 7.2

操作系统可通过“cat /etc/redhat-release”进行查看；

1.3.2 硬件环境

建议配置：内存 64G 及以上、cpu E5 系列支持 sse4 指令集，磁盘大小为 1T。

cpu 需支持 sse4 指令集，sse4 指令集可通过“cat /proc/cpuinfo”进行查看，看是否包含在 flags 中。

1.3.3 软件环境

1) 数据库服务器：mysql (5.7.17 及以上版本)

2) 识别服务器：

jdk1.8.0_171

apache-tomcat-9.0.35

gcc 4.8.5

g++ 4.8.5

redis 3.2.0

gcc 和 g++可分别通过 “gcc --version” 和 “g++ --version” 进行查看；

1.4 系统安装包目录结构

离线语音识别转写系统安装包为 Thinkit_Offline_SystemE.tar.gz 内含识别服务器软件和运行环境、第三方工具等。可以将此安装包整个拷贝到目标服务器上解压安装，或者先解压，之后不同服务器拷贝对应子目录进行安装即可。

离线语音识别转写系统目录结构和重要文件说明如下：

1.4.1 总目录结构

离线语音识别转写系统的系统名称为 “Thinkit_Offline_SystemE”，其包含的目录结构如下：

-- “System_Offline_Server”	识别引擎运行环境（识别服务器安装）
-- “database”	数据库建表 sql 存放目录（数据库若已创建可以忽略）
-- “db_user_pwd”	MySQL 数据库的信息
-- “SystemStartup.sh”	系统统一启动脚本
-- “SystemShutdown.sh”	系统统一关闭脚本
-- “bash_global”	系统配置信息
-- “TaskRecovery”	任务恢复
-- “readme.txt”	目录结构说明

1.4.2 子目录结构

1) System_Offline_Server 目录结构

(1) 整体结构：

-- “tools”	软件工具安装包
-- “xml_server”	xml 服务端环境

-- “TaskRecovery”	任务恢复
-- “DetectOverlap”	叠音检测引擎环境
-- “ClusterSceneServer”	场景分割引擎环境
-- “TBNR_release_time”	全文转写引擎环境
-- “DetectEmotion”	情绪检测引擎环境
-- “ItnRecognize”	转阿拉伯引擎环境
-- “PuncRecognize”	加标点引擎环境
-- “GenderRecognize”	性别引擎环境
-- “Offline_System_Client”	总控端环境
-- “startup.sh”	启动守护进程和各识别引擎、总控端进程的

脚本

-- “shutdown.sh”	关闭守护进程和各识别引擎、总控端进程的
------------------	---------------------

脚本

-- “RunAndMonitor.sh”	各识别引擎、总控端的守护进程脚本，可配
-----------------------	---------------------

置各引擎模块功能的开关、及开启线程数

(2) 子目录结构

Tools:

-- “ffmpeg”	ffmpeg_4.3 安装包目录
-- “redis”	redis-3.2.0 安装包目录
-- “sox ”	sox_14.4.2 安装包目录
-- “java”	jdk_1.8.0_171 安装包目录
-- “tomcat”	tomcat_9.0.35 安装包目录
-- “protobuf”	protobuf_2.5.0 安装包目录
-- “Nginx”	Nginx 安装包目录
-- “FileSync”	FileSync 安装包目录
-- “Encrypt”	Encrypt 安装包目录
-- “SDK”	测试工具目录
-- “AutomaticInstall”	软件工具自动安装脚本存放目录
-- “ffmpeg_install.sh”	ffmpeg_4.3 自动安装脚本

-- “sox_install.sh”	sox_14.4.2 自动安装脚本
-- “java_install.sh”	jdk_1.8.0_171 自动安装脚本
-- “tomcat_install.sh”	tomcat_9.0.35 自动安装脚本
-- “redis_install.sh”	redis-3.2.0 自动安装脚本
-- “protobuf_install.sh”	protobuf_2.5.0 自动安装脚本
-- “encrypt_install.sh”	encrypt 自动安装脚本
-- “filessync_install.sh”	filessync 自动安装脚本
-- “run.sh”	自动安装 ffmpeg、sox、jdk、tomcat、redis、protobuf、encrypt、filessync 的脚本

xml_server:

-- “configure_xml.cfg”	xml 结果文件中相关属性配置
-- “XMLServer”	xml 服务端主程序
-- “punctuation_rule.cfg”	标点判断规则
-- “lib”	
-- “libprotobuf.so”	protobuf 动态库
-- “run.sh”	xml 服务端运行脚本

ClusterSceneServer:

-- “Cluster_Scene_server_test”	场景分割主程序
-- “configure.cfg”	聚类配置文件
-- “Cluster”	聚类 so 存放目录
-- “KWSAPI_LIBLINUX”	依赖库
-- “lib”	protobuf 动态库存放路径
-- “license.dat”	授权文件
-- “logt.txt”	场景分割日志文件
-- “run.sh”	场景分割模块运行脚本

Offline_System_Client:

-- "lib" 依赖的mysql、protobuf、redis、thrift 动态库存放目

| -- "log" | 日志文件存放目录(重点看日志 ASR.log.wf) |

| -- “cfg.ini” | 配置文件(数据库、日志、语言编码格式、开启引擎功能) |

-- “run. sh”	总控端运行脚本
--------------	---------

|-- “configure_code.cfg” 配置文件(处理的语言有多种编码格式时配置)

```
-- "custom_client_multiserver" 总控端程序
```

```
-- “client_run.sh” 启动文件同步服务客户端
```

```
-- "ctrlTask.sh"  执行文件同步服务脚本
```

```
|-- “FileSync_Client”      文件同步服务客户端
```

— “FileSync Server” 文件同步服务服务端

```
-- "server run.sh" 启动文件同步服务服务端
```

TBNR release time

-- “bin”	运行环境
----------	------

| -- “KWSAPI ENV” | 依赖库及声学模型目录

|-- “KWSAPI LIBLINUX” 依赖库

| -- “parse” | 声学模型存放目录

```
-- "final.arpa" 声学模型先验概率文件 #请替换为实际模型
```

```
-- “final.bin” 声学模型文件 #请替换为实际模型
```

| -- “phonemap.bin” | 音素映射文件 |

-- “model” 语言模型、字典、配置文件存放目录

| -- “net” | 语言模型目录

```
-- “final.Tbfsm” 语言模型 #请替换为实际模型
```

| -- “scripts” | 配置文件目录

```
-- "ElvaData"      识别过程中语音特征存放目录
-- "VAD"           VAD 模型存放路径
-- "VAD_CaiLing_5L_128_11frames_AddNoise_4_0.008.bin"
-- "VAD_AutoRegressiveWindow1_smallest_FarDis_7.bin"
-- "hmm"           HTK 声学模型及因素集存放目录
-- "hmmDef"        HTK 声学模型
-- "hmmDef.bin"    HTK 声学模型二进制文件
-- "mono.list"     音素集
```

DetectOverlap:

```
-- "frontend.log"      日志文件
-- "frontend_ErrorMsg.txt" 日志文件（记录出错信息）
-- "in.cfg"            配置文件
-- "run.sh"            叠音检测运行脚本
-- "lib"               叠音检测主程序及依赖库存放路径
-- "license.dat"       叠音授权文件
-- "source"            叠音检测模型存放路径
```

DetectEmotion:

```
-- "frontend_ErrorMsg.txt" 日志文件
-- "frontend.log"          日志文件（记录出错信息）
-- "in.cfg"                配置文件
-- "lib"                   情绪检测主程序及依赖库存放路径
-- "license.dat"           情绪授权文件
-- "run.sh"                情绪检测运行脚本
-- "source"                情绪检测模型存放路径
```

ItnRecognize:

```
-- "itn_recognize_server" 转阿拉伯主程序
-- "conf"                  文法文件
-- "lib"                   动态库
```

|-- “run. sh” 阿拉伯模块运行脚本

PuncRecognize:

|-- “punc_recognize_server” 加标点主程序
 |-- “conf” 配置文件
 |-- “lib” 动态库
 |-- “run. sh” 标点模块运行脚本
 |-- “model” 模型
 |-- “license. dat” 授权文件

GenderRecognize_SVM:

|-- “gender_recognize_server” 性别主程序
 |-- “GIDSysdir” 配置文件
 |-- “lib” 动态库
 |-- “run. sh” 性别模块运行脚本
 |-- “license. dat” 授权文件

2) data 目录结构

|-- “task_info. sql” 建 task_info 表的脚本

2 Java 能力接口、Redis 服务配置文件介绍

Java 应用使用 tomcat 作为 web 容器进行部署，ability.war 包放入./tomcat/webapp 目录下，然后重启 tomcat 应用服务即可。

备注：System_Offline_Server/tools/tomcat/apache-tomcat-9.0.35.tar.gz 已经包含 Java 应用，不需要手动安装；

2.1 JDBC 配置:

配置文件:

~/local/tomcat/apache-tomcat-9.0.35/webapps/ability/WEB-INF/classes/jdbc.properties

修改内容:

driver=com.mysql.jdbc.Driver

```
url=jdbc:mysql://127.0.0.1:3306/asr?&characterEncoding=UTF8
username=root
jdbcType=false
password=123456
passkey=mysql_2018
```

2.2 日志配置

配置文件:

```
~/local/tomcat/apache-tomcat-9.0.35/webapps/ability/WEB-INF/classes/log4j.properties
```

修改内容:

```
log4j.appender.MyRfile1.File=/home/platform2/abilitylog/jsonrpc.log
```

2.3 redis 配置

配置文件:

```
~/local/tomcat/apache-tomcat-9.0.35/webapps/ability/WEB-INF/classes/redis.properties
```

修改内容:

```
redis.queue=voicetask,voicetask1
redisCluster=127.0.0.1:7000,127.0.0.1:7001,127.0.0.1:7002,127.0.0.1:7003,127.0.0.1:7004,127.0.0.1:7005
redisSingle=127.0.0.1:30101
connectionTimeout=2000
maxAttempts=6
soTimeout=2000
initialCapacity=1000
maximumSize=10000
isPassword=true
isCluster=true
redisType=false
```

```
redisPassword=foobared
redisPassKeys=mysql_2018
audio.ServerIP = 127.0.0.1
grsSleepTime=1000
voiceTempPath = /home/wangchaofeng/tempvoice/
shellPath=/home/wangchaofeng/local/FileSync/FileSync/
getCommand=./FileSync.sh GET
sendCommand=./FileSync.sh SEND
stt.originalFilePath=/home/wangchaofeng/tempvoice_Xml/
callbackQueue=callbackList_127.0.0.1
thread.result.corePoolSize=4
thread.result.maxPoolSize=5
thread.result.queueCapacity=100
taskSleepTime=1000
authenticationSwitch=false
```

3 离线系统配置文件介绍

3.1 目录结构说明

识别服务器上需要安装的软件工具及程序都位于 System_Offline_Server 目录里，目录结构如下：

```
--TBNR_release_time 识别服务端
--ClusterSceneServer 场景分割服务端
--DetectOverlap 叠音检测服务端
--DetectEmotion 情绪检测服务端
--xml_server XML 服务端
--ItnRecognize 转阿拉伯服务端
--PuncRecognize 标点服务端
--GenderRecognize 性别服务端
--Offline_System_Client 总控端
```

--tools 软件工具安装包

--RunAndMonitor.sh 守护进程脚本

--startup.sh 启动 RunAndMonitor.sh 和各服务端、总控端进程脚本

--shutdown.sh 关闭 RunAndMonitor.sh 和各服务端、总控端进程脚本

注意：务必保持该目录结构不变，否则修改的地方会很多。

介绍一下以上几个脚本：

1) RunAndMonitor.sh（守护进程脚本）：启动和维护各服务端和总控端进程的正常运行。通过监控 RunAndMonitor.sh 的运行状态即可完成对语音识别服务子系统的监控。（默认不删除 TBNR_120.pid 文件，需要删除修改 IsDelPid=true）

2) startup.sh：后台启动运行 RunAndMonitor.sh 的脚本，且避免重复启动 RunAndMonitor.sh。

3) shutdown.sh：关闭 RunAndMonitor.sh 和各服务端、总控端进程的脚本。shutdown.sh 用于关闭系统功能。

3.2 安装软件工具

语音分析服务需用到的软件工具及库文件包括：jdk, tomcat, sox, ffmpeg, libprotobuf.so, libxml2.so。

- jdk, tomcat：语音识别服务、分布式任务接收服务使用。
- sox：总控端对语音进行转码使用。
- ffmpeg：总控端对语音进行转码使用。
- redis：Java 应用于转写服务之间通信的消息队列。
- libprotobuf.so：总控端和各语音识别服务之间通信使用。
- libxml2.so：xml 服务端生成 xml 文件时使用。

其中依赖库对应的版本说明如下：

依赖库名称	版本号（推荐版本）
gcc	4.8.5
g++	4.8
jdk	1.8.0_171
sox	14.4.2

ffmpeg	4.3
tomcat	9.0.35
redis	3.2.0
libprotobuf.so	2.5.0
libxml2.so	—

注意：

(1) 库文件 libprotobuf.so , libxml2.so 已包含在运行环境中，无需再安装。

3.3 各服务端和总控端配置文件

3.3.1 场景分割服务端 tomcat 服务

场景分割服务端 tomcat 服务默认使用 20100 端口，若 tomcat 端口有变动，则需相应修改 System_Offline_Server/RunAndMonitor.sh 中 roleServerPort。

#角色区分服务端口

roleServerPort=20100

3.3.2 总控端配置

1) 配置 System_Offline_Server/Offline_System_Client/cfg.ini。

数据库、redis 队列、日志、开启引擎功能等相关参数的配置，重点关注如下参数的配置，保持默认即可。

#redis 连接数

redisThread=1

DeleteTempResult=true

DeleteOriginalWav=true

logEvents=5

2) 当 cfg.ini 中 wavFormat=0 (多种格式混合) 时，还需配置 configure_code.cfg。configure_code.cfg 中有详细的参数说明。

3.3.3 xml 服务端配置

1) 配置 System_Offline_Server/xml_server/configure_xml.cfg

主要按实际需求配置 xml 结果格式、静音阈值和叠音阈值（单位：毫秒）。

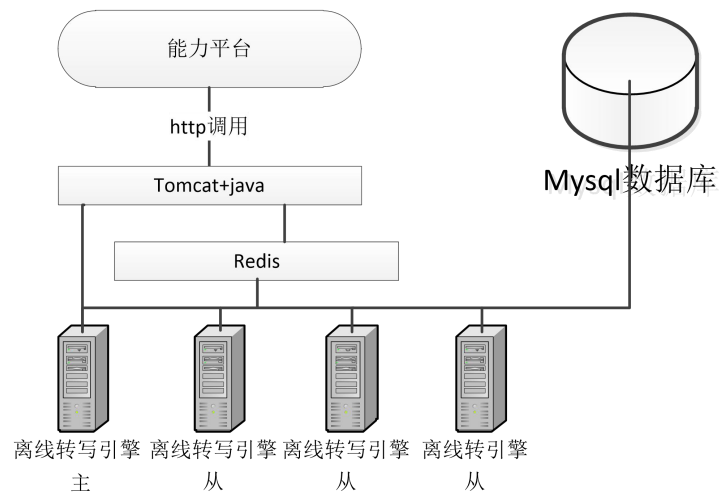
Xml_Format=2

LongSilenceDuration=3000

InterruptedDuration=100

4 系统安装与配置

4.1 分布式部署说明



分布式集群示意图

离线系统部署示意图如上图所示，包括一台主服务器和多台从服务器。tomcat 存储语音文件到主服务器，并添加转写任务到 redis，主服务器和从服务器获取 redis 任务，并根据任务信息获取主服务器的语音文件，进行转写。

4.2 系统安装

离线系统安装不要使用 root 用户，安装包放在 “~/” 目录下直接安装，不要在外层新建文件夹。

4.2.1 数据库配置

1. 安装数据库（如已安装跳过此步骤）
2. 解压 tar 包

```
tar -zxvf mysql-5.7.30-1.el7.x86_64.rpm-bundle.tar
```

3. rpm -Uvh *.rpm --nodeps --force
4. 关闭 mysql 服务

```
service mysqld stop
```

5. 修改配置文件

```
vi /etc/my.cnf
```

添加

```
skip-grant-tables
```

6. 开启 mysql 服务

```
service mysqld start
```

7. 空密码登录

```
mysql -u -p root
```

8. 设置 mysql 密码

```
mysql> set password for root@localhost = password('123456');ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option so it cannot execute this statement

mysql> flush privileges; #更新权限 Query OK, 0 rows affected (0.00 sec)

mysql> set password for root@localhost = password('123456'); Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>flush privileges; #更新权限

mysql>quit; #退出

service mysqld stop # 停止 mysql 服务, 恢复 mysql 配置

vim /etc/my.cnf      #修改配置文件# Disabling symbolic-links is recommended to prevent assorted security risks# skip-grant-tables # 注释掉这句话

symbolic-links=0

service mysqld start # 启动 mysql 服务

mysql -uroot -p # 输入新密码登录
```

9. 设置远程访问权限

A: grant all privileges on *.* to 'root'@'%' identified by '123456' with grant option;

B: flush privileges;

此时密码如果简单, 可能会报错 (解决版本, 修改密码策略或者更新密码强度)

ERROR 1819 (HY000): Your password does not satisfy the current policy requirements

10. 查看 mysql 初始的密码策略，

输入语句 “SHOW VARIABLES LIKE 'validate_password%';” 进行查看，

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password_check_user_name | OFF |
| validate_password_dictionary_file |  |
| validate_password_length | 8 |
| validate_password_mixed_case_count | 1 |
| validate_password_number_count | 1 |
| validate_password_policy | MEDIUM |
| validate_password_special_char_count | 1 |
+-----+-----+
7 rows in set (0.00 sec)
```

mysql 密码策略!!!

首先需要设置密码的验证强度等级，设置 validate_password_policy 的全局参数为 LOW 即可，输入设值语句 “set global validate_password_policy=LOW;” 进行设值，

```
mysql> set global validate_password_policy=LOW;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password_check_user_name | OFF |
| validate_password_dictionary_file |  |
| validate_password_length | 8 |
| validate_password_mixed_case_count | 1 |
| validate_password_number_count | 1 |
| validate_password_policy | LOW |
| validate_password_special_char_count | 1 |
+-----+-----+
7 rows in set (0.00 sec)
```

修改为 LOW 后，就只验证密码的长度了!!!

11. 当前密码长度为 8，如果不介意的话就不用修改了，按照通用的来讲，设置为 6 位的密码，设置 validate_password_length 的全局参数为 6 即可，输入设值语句 “set global validate_password_length=6;” 进行设值

12. 添加数据库信息

cd Thinkit_Offline_SystemE/database

使用 task_info.sql 建表

将数据库信息添加到 db_user_pwd

4.2.2 配置“环境变量”

1. 修改 Thinkit_Offline_SystemE/bash_global

一般只需要修改 THINKIT_GLOBAL_MASTER_IPADDR=(主服务器 IP 地址，引擎集群使用 redis 单机是设置为连接 redis 的 IP 地址)THINKIT_GLOBAL_OWN_IPADDR=(当

前服务 IP 地址) 和 THINKIT_DB_KEY_WORD= (本服务器对应的省份简写)。数据库信息, 省份简写 (或关键词) 在 ./ database/db_user_pwd 文件中查找, 其他配置保持默认即可。

```
export THINKIT_GLOBAL_MASTER_IPADDR=127.0.0.1
```

```
## Owner   IP Address
```

```
export THINKIT_GLOBAL_OWN_IPADDR=127.0.0.1
```

```
## Install Dir
```

```
export THINKIT_HOME_LOCAL_DIR=$HOME/local
```

```
export THINKIT_HOME_VOICE_DIR=$HOME/tempvoice/
```

```
export THINKIT_HOME_LOG_DIR=$HOME/abilitylog
```

```
export THINKIT_SYSTEM_DIR=$PWD
```

```
## DataBase cfg
```

```
export THINKIT_DB_KEY_WORD=TEST_0
```

```
export THINKIT_DB_USER_PWD=$THINKIT_SYSTEM_DIR/database/db_user_pwd
```

```
export THINKIT_DB_EXPIRE=7
```

```
#数据库密码是否加密
```

```
export THINKIT_DB_Encrypt=false
```

```
## Redis cfg
```

```
#redis 是否是集群
```

```
export THINKIT_ISCLUSTER=false
```

```
export THINKIT_REDIS_SERVER=$THINKIT_GLOBAL_OWN_IPADDR
```

```
export THINKIT_REDIS_PORT=30101
```

```
#redis 是否有密码
```

```
export THINKIT_REDIS_ISPASSWD=false
```

```
export
```

```
THINKIT_REDIS_ADD=127.0.0.1:7010,127.0.0.1:7011,127.0.0.1:7012,127.0.0.1:7013,127.0.
```

0.1:7014,127.0.0.1:7015

export THINKIT_REDIS_PASS=EzU6VqD430skfU95T4c1fA==

export THINKIT_REDIS_PASS_KEYS=mysql_2018

#redis 密码是否加密

export THINKIT_REDIS_Encrypt=false

export THINKIT_REDIS_QUEUE_ENGIN=voicetask

export THINKIT_REDIS_QUEUE_TOMCAT=voicetask,voicetask1

export THINKIT_REDIS_TMP_VOICE=\$THINKIT_HOME_VOICE_DIR

export THINKIT_SHELLPATH=\$HOME/local/FilesSync/FilesSync/

export THINKIT_AUTHTOKEN=false

2.执行脚本 source ./bash_global

4.2.3 离线系统安装

1. 进入目录~/Thinkit_Offline_SystemE
2. ./SystemStartup.sh install 安装所有的应用和依赖包
3. 如果 **安装失败**，查找原因，并删除~/local 目录下已安装的文件

4.2.4 Nginx 安装（多 tomcat 需安装 nginx，默认单使用 tomcat，无需安装 nginx）

1. 进入目录~/Thinkit_Offline_SystemE/System_Offline_Server/tools/Nginx
2. tar -xvf nginx_tools.tar.gz 解压 Nginx 安装包
3. ./install.sh 安装 Nginx

4.2.5 配置 Nginx

1. 修改 server_conf_offline.txt

server=127.0.0.1:20100 weight=2

server=127.0.0.1:20100 weight=2
2. 重新加载生效：

./sbin -s reload

4.3 系统配置

4.3.1 秘钥申请

系统的每个服务模块都有权限限制，需要对应的秘钥才能正常使用。

1.需要申请的 license 如下:

Thinkit_Offline_SystemE/System_Offline_Server/ClusterSceneServer/license.dat##产品号 26

Thinkit_Offline_SystemE/System_Offline_Server/DetectOverlap/license.dat##产品号 199

Thinkit_Offline_SystemE/System_Offline_Server/TBNR_release_time/bin/license_120.dat##产品号 120 (需替换 4.3 版本的授权文件)

/Thinkit_Offline_SystemE/System_Offline_Server/PuncRecognize//license.dat##产品号 120

Thinkit_Offline_SystemE/System_Offline_Server/DetectEmotion/license.dat##产品号 196

/Thinkit_Offline_SystemE/System_Offline_Server/GenderRecognize/GenderRecognize_SVM/license.dat###产品号 195

2.申请流程

2.1 执行 Thinkit_Offline_SystemE/System_Offline_Server/tools/getinfo 脚本, 当前目录得到机器信息文件 localhost.localdomain_machine.info。

2.2 执行 Thinkit_Offline_SystemE/System_Offline_Server/TBNR_release_time/bin/getinfo_4.3 脚本, 当前目录得到机器信息文件 localhost.localdomain_machine.info。(识别模块授权使用 4.3 版本, 只需申请 120 号授权)

2.3 发送 localhost.localdomain_machine.info 和 getinfo_4.3 生成的机器信息 (localhost.localdomain_machine.info) 文件给北京相关人员, 标明密钥产品号和服务器线数 (一般为 CPU 核心数的 1/2), 申请密钥 (需申请两个版本的 120 号授权)。

2.4 替换对应的产品密钥。

4.3.2 根据能力对应的省份替换模型

1. 模型路径:

~/Thinkit_Offline_SystemE/System_Offline_Server/TBNR_release_time/KWSAPI_ENV/pars
e/lvcsr_standard/plp_pitch_hlda_fmpe_mpe/nnmodel/final.arpa

~/Thinkit_Offline_SystemE/System_Offline_Server/TBNR_release_time/KWSAPI_ENV/pars
e/lvcsr_standard/plp_pitch_hlda_fmpe_mpe/nnmodel/final.bin

~/Thinkit_Offline_SystemE/System_Offline_Server/TBNR_release_time/model/net/final.Tbfs
m

2. 使用目标省份的模型替换即可。

5 系统运行、关闭与迁移

5.1 启动系统

使用统一启动脚本：

```
./SystemStartup.sh
```

5.1.1 能力服务 http 地址

1. 单机服务由 Tomcat 提供服务，地址：<http://IP:20100/ability/offlineasr>
2. 多 tomcat 由 nginx 提供服务，地址：<http://IP:30100/offlineasr>

5.2 关闭系统

使用统一关闭脚本：

```
./SystemShutdown.sh
```

5.3 服务器迁移

5.3.1 停止服务

```
cd ~/Thinkit_Offline_SystemE/System_Offline_Server/  
./shutdown.sh
```

5.3.2 配置修改

按照 4.2 系统配置：4.1/4.2/4.3 对系统进行重新配置

5.3.3 任务恢复(将未完成的任务恢复到主服务器)

```
cd ~/Thinkit_Offline_SystemE/TaskRecovery  
./run.sh
```

5.3.4 启动服务

```
cd ~/Thinkit_Offline_SystemE/System_Offline_Server/  
./startup.sh
```

6 判断部署是否正确

按 5.1 节启动系统后，可按如下方法判断系统是否部署正确：

6.1.1 检查 Tomcat 服务是否正常

部署好 tomcat 服务后，检查 tomcat 是否可以正常运行：

运行 `apache-tomcat-9.0.35/bin/startup.sh` 来启动 tomcat 服务；

若启动失败或者启动之前，先运行 `apache-tomcat-9.0.35/bin/shutdown.sh` 关闭一下，之后再启动。检查 tomcat 运行正常后，请运行 `apache-tomcat-9.0.35/bin/shutdown.sh` 关闭 tomcat。后续将统一启动 tomcat 服务。

6.2 检查系统是否正常启动

查看识别服务器检查 `System_Offline_Server` 目录下是否有 `RunAndMonitor.log` 文件生成？

如果有 `RunAndMonitor.log` 生成，且 `RunAndMonitor.log` 文件末尾有类似如下内容输出（重点看是否有 `custom_client_multiserver is running`），则表示引擎启动成功。

```
Sat Sep 29 04:00:24 EDT 2018
0 offline_customer_server_test_dnnvad running,will restart soon !
restart all server and client...
FileSync_Server startup OK !
Wait for client initial...
offline_customer_server_test_dnnvad is running
Cluster_Scene_server_test is running
itn_recognize_server is running
punc_recognize_server is running
DetectOverlapPro is running
DetectEmotionPro is running
gender_recognize_server is running
XMLServer is running
Wait for client initial...
custom_client_multiserver is running
Sat Sep 29 04:01:40 EDT 2018
```

注意：执行 `startup.sh` 启动各引擎服务，到引擎初始化完毕整个过程中约耗时 3 分钟。

引擎启动不成功的情况大致有以下几种：

1) 无 RunAndMonitor.log 生成：

可能原因：System_Offline_Server 目录下 startup.sh 中 SystemPath，以及 RunAndMonitor.sh 中 SystemPath 没有配置正确。

解决方法：正确配置之后，重新运行 startup.sh。

2) 部分服务端或总控端启动失败

如果某服务端或者总控端启动失败，则会报错 “Error: restart xxx failed!”。

解决方法：进入 System_Offline_Server 目录下相应引擎子目录查看对应日志文件或者运行引擎目录下的 run.sh，看屏幕上的报错提示。按提示修改错误，之后重新运行 System_Offline_Server 目录下 startup.sh。

以报错 “Error: restart offline_customer_server_test_dnnvad failed!” 为例，说明如何检查错误：

步骤 1：进入 System_Offline_Server/TBNR_release_time/bin 目录，执行：

```
./ run.sh 1
```

步骤 2：当屏幕显示类似如下信息时，表示识别服务端正常（注意：确定引擎正常初始化后请用 ctrl+c 关闭进程）。否则需要根据屏幕打印信息定位出差原因，识别服务端无法启动常见的问题：是否有授权的 license_120.dat 文件、以及开启的线程数是否超过了允许开启的线程数、端口是否已被占用。

```
[root@node192 bin]# ./run.sh 1
[Nov. 18, 2016 18:23:44:245]:engine begin to init.....
configFile is in ../model/scripts/WFSTDecoder-inputMethod_dnn_onlyrec.cfg
*****[LVCSR VERSION : 220160819433]*****
error : No such device
loading acoustic models ....
setup hmm ok!
start load wfst network...
Finish reading      14990766 states
Finish reading      49731299 transitions
finish load wfst network...
silMarker = 11253 spMarker = 11678
syll dict!
finish load word dictionary.....
300 300 300 1024 512 256 1024 1024 512 256 1024 1024 512 256 1024 5000
Total parameters: 33933380
iInputVecSize 60 FrameSize 60
[Nov. 18, 2016 18:23:48:731]:engine init done.....
[InitSocket]:ip is 127.0.0.1 ports is 6608
hostname is node192
```

步骤 3：按照错误提示进行修改后（比如重新申请替换授权文件），需按步骤 2

检查错误是否被修复，务必保证引擎可被正常初始化。

步骤 4：重新运行 System_Offline_ServerE 目录下 startup.sh。

6.3 检查系统是否运行正常

往 redis 任务队列中插入任务，等一会儿，去相应的 xml 结果目录下查看是否有对应的 xml 结果文件生成，检查 xml 结果内容是否完整、正确？检查 task_info 表中各字段信息是否正确。

6.4 查看 log 信息

查看识别服务器总控端日志文件

System_Offline_ServerE/Offline_System_Client/log/ASR.log.wf 中是否有 Warning 或 Error 信息，并据此定位问题原因进行修复。

导致任务失败的原因比较多，如：网络连接问题、文件读写权限问题、语音本身问题（如：文件不存在、文件编码格式错误）等等。