

产品开发背景

LogCollector是基于应用日志到Mqtt服务器的一套ETL工具和服务组件。目前常用的ETL工具Flume也可以完成日志的采集、传输、转换和存储，但是Flume工具仅能应用到通信质量无障的局域网环境，在公网环境下可能因网络不稳定等因素导致连接远端服务的发送器组件失败，而此时收集器组件可能并不知情，数据仍然会继续传送到通道组件，这容易导致通道组件内存泄露从而引发OOM错误；另一方面由于通道错误导致实时收集的数据发送失败，收集器也没有记录实时检查点，这意味着发送失败的数据将面临丢失。发生所有这些问题的根源在于公网传递数据的不稳定性所致，因此Flume是一款仅能适用于云域内网的ETL工具。在这种问题背景的需求驱动下产生了LogCollector这款产品，LogCollector完全按照产品级标准使用JAVA语言进行开发，安装时无需再安装外置JDK支持，解压开箱即用。

产品功能特性

LogCollector是一款基于应用方日志到Mqtt服务器的通用ETL传输工具，同时适用于云域内网数据传送和跨云数据传送；同时支持Windows和Linux双系统平台；同时支持实时传送、离线传送和断点续传；同时支持工具化、服务化、扩展化、集成化；收集器可一键安装部署，自动识别平台和系统环境并完成相应配置，无需任何附加操作，解压开箱即用。

产品安装部署

1. 下载LogCollector-1.0

wget <https://github.com/lixiang2114/Software/raw/main/LogCollector.zip>

2. 安装LogCollector-1.0

unzip LogCollector.zip -d /software/LogCollector

备注：

本套产品支持在Windows系统上安装部署和使用，在Windows上的安装和Linux上的安装类同，都是解压即用模式，如：解压到D盘根目录可以执行：unzip LogCollector.zip -d D:/software/LogCollector

产品配置说明

1. 配置应用上下文

```
vi /software/LogCollector/conf/context.properties
context.realTime=true
context.initOnStart=true
emqx.persistenceType=org.eclipse.paho.client.mqttv3.persist.MemoryPersistence
emqx.hostList=192.168.162.127:1883
emqx.filterName=defaultLogFilter
emqx.protocolType=tcp
emqx.batchSize=100
```

参数说明:

参数名称	参数类型	默认值	是否可选	参数说明
context.realTime	boolean	true	可选	是否实时传送模式启动服务
context.initOnStart	boolean	true	可选	是否在启动服务时初始化配置参数
emqx.persistenceType	MqttClientPersistence	MemoryPersistence	可选	MQTT消息持久化类型
emqx.hostList	string	192.168.162.127:1883	可选	MQTT服务器连接地址表,多个地址之间使用英文逗号","分隔
emqx.filterName	string	filter	可选	过滤器配置文件名(不含后缀扩展名)
emqx.protocolType	string	tcp	可选	连接MQTT服务器使用的传输层协议,目前仅支持TCP协议
emqx.batchSize	integer	100	可选	推送数据到MQTT服务器的批处理尺寸

2. 配置日志上下文

```

vi /software/LogCollector/conf/logger.properties
logger.appLogFile=g:/cloudlog/app/my2.log

logger.transferSaveFile=g:/cloudlog/flume/mylogger.0
logger.transferSaveMaxSize=2GB

logger.loggerFile=g:/cloudlog/flume/mylogger.0
logger.lineNumber=0
logger.byteNumber=0

logger.manualLoggerFile=g:/cloudlog/test/mylogger.0
logger.manualLineNumber=0
logger.manualByteNumber=0

```

参数说明：

参数名称	参数类型	默认值	是否可选	参数说明
logger.appLogFile	string	无	必选	需要收集的应用方实时日志文件绝对路径,该日志文件可能由应用方的log4j组件产生
logger.transferSaveFile	string	无	必选	实时转存日志文件绝对路径,该日志文件由转存进程维护,配置文件名必须以数字后缀扩展名0结尾,并由转存进程自动按数字序列依次递增切换
logger.transferSaveMaxSize	string	2G	可选	实时转存日志文件最大尺寸,超过这个尺寸,转存进程将按数字递增序列创建下一个新的转存文件,参数值的单位有:B、KB、MB、GB、TB、EB等
logger.loggerFile	string	无	必选	实时发送器缓冲日志文件绝对路径,该参数值必须与转存日志文件路径完全相同,该文件由发送器进程维护以实现断点续传,后续版本迭代会持续优化
logger.lineNumber	integer	0	可选	实时发送器缓冲日志文件行号检查点
logger.byteNumber	integer	0	可选	实时发送器缓冲日志文件字节检查点
logger.manualLoggerFile	string	无	必选	离线发送器日志文件绝对路径
logger.manualLineNumber	integer	0	可选	离线发送器缓冲日志文件行号检查点
logger.manualByteNumber	integer	0	可选	离线发送器缓冲日志文件字节检查点

3. 配置过滤器上下文

```
vi /software/LogCollector/filter/conf/defaultLogFilter.properties
type=com.github.lixiang2114.etllog.filter.DefaultLogFilterImpl
jwtSecret=bG14awFuZW==
tokenFrom=password
userName=admin
password=public
topic=Topic_Test
tokenExpire=-1
```

参数说明:

参数名称	参数类型	默认值	是否可选	参数说明
type	string	DefaultLogFilterImpl	可选	过滤器实现类全名
jwtSecret	string	无	必选	用于加密Token的秘钥,该秘钥必须在MQTT服务端和客户端统一
tokenFrom	string	password	可选	在访问MQTT服务时携带Token验证值的字段名
userName	string	admin	可选	若使用Token验证(默认使用)则该参数验证被弱化
passWord	string	public	可选	若使用Token验证(默认使用)则该参数值为Token值
topic	string	无	必选	收集器连接MQTT服务器的主题名称
tokenExpire	integer	-1	可选	连接MQTT服务器的Token过期时间,默认值为-1表示永不过期

4. 配置服务器上下文

LogCollector产品使用SpringBoot框架构建和开发,其运维侧服务配置与常规SpringBoot工程相同。我们可以打开LogCollector产品安装目录下的conf目录,并找到application.yml配置文件。将其打开并修改里面的参数,即可完成运维管理侧的服务配置。在该配置文件中,我们可以修改服务启动端口,服务器日志配置(后续会陆续优化)等参数,通常没有特别的理由,我们无需修改这些服务器参数,目前该配置文件的配置如下:

```
cat /software/LogCollector/conf/application.yml

server.port: 8088

spring:
  profiles.active: test
  application.name: LogCollector

logging.maxFileSize: 100MB
logging.basePkg: com.github.lixiang2114.etllog
```

产品使用说明

1. 在线实时传送日志

- 配置应用上下文

```
vi /software/LogCollector/conf/context.properties
context.realTime=true
context.initOnStart=true
emqx.persistenceType=org.eclipse.paho.client.mqttv3.persist.MemoryPersistence
emqx.hostList=192.168.162.127:1883
emqx.filterName=defaultLogFilter
emqx.protocolType=tcp
```

备注:

注意上述参数都提供了默认值，hostList参数如果不配置则默认为"127.0.0.1:1883"，即连接本地的MQTT服务，filterName参数如果不配置则默认为filter

- 配置日志上下文

```
vi /software/LogCollector/conf/logger.properties
logger.appLogFile=g:/cloudlog/app/my2.log

logger.transferSaveFile=g:/cloudlog/flume/mylogger.0
logger.transferSaveMaxSize=2GB

logger.loggerFile=g:/cloudlog/flume/mylogger.0
logger.lineNumber=0
logger.byteNumber=0
```

备注:

appLogFile参数值必须指向应用方SLF4J或Log4J日志输出文件绝对路径，否则无法采集应用方在线实时日志，其次，transferSaveFile参数值必须与loggerFile参数值保持相同，同时日志文件绝对路径后的后缀扩展名必须以数字0结尾

- 配置过滤器上下文

备注:

- 启动日志收集器服务

[illegible]

备注:

Windows系统下可以直接双击对应目录下的startup.cmd文件即可启动日志收集器服务，日志收集器服务一旦启动之后，就自动开始收集本地应用端日志并将其推送到指定的MQTT服务器了，可以通过日志收集器运维管理侧的接口来控制ETL流程、转存流程、Token调度流程等的启停，甚至可以在运行时动态变更配置收集器各项参数等

- 停止日志收集器服务

Windows端可以直接在收集器本地按下Ctrl+C平滑终止服务，Linux端可以直接执行pkill java命令来平滑终止服务，最后，不论是Windows端还是Linux端都可以直接发送以下命令来平滑终止收集器服务进程：

```
curl -ik -X GET http://127.0.0.1:8088/admin/shutdown
$#8203:
```

2. 离线批量传送日志

- 配置应用上下文

```
vi /software/LogCollector/conf/context.properties
context.realTime=false
context.initOnStart=true
emqx.persistenceType=org.eclipse.paho.client.mqttv3.persist.MemoryPersistence
emqx.hostList=192.168.162.127:1883
emqx.filterName=defaultLogFilter
emqx.protocolType=tcp
```

备注:

相对于在线实时传送而言，只需要将realTime参数改成false即可，其它参数保持不变

- 配置日志上下文

```
vi /software/LogCollector/conf/logger.properties
logger.manualLoggerFile=g:/cloudlog/test/mylogger.0
logger.manualLineNumber=0
logger.manualByteNumber=0
```

备注:

相对于在线实时传送而言，离线传送日志的配置很简单，只需要指定传送的日志文件和检查点即可

- 配置过滤器上下文

```
vi /software/LogCollector/filter/conf/defaultLogFilter.properties
type=com.github.lixiang2114.etllog.filter.DefaultLogFilterImpl
jwtSecret=bG14aWFuZW==
tokenFrom=password
userName=admin
password=public
topic=Topic_Test
tokenExpire=-1
```

备注:

相对于在线实时传送而言，过滤器的配置没有任何变化，因为这些参数实际上是MQTT的连接参数，不论是实时传送还是离线传送，这些参数都是需要的

- 启动日志收集器服务
bash /software/LogCollector/bin/startup.sh

