Centralized stream statistics and CDN events collection to MySQL DB

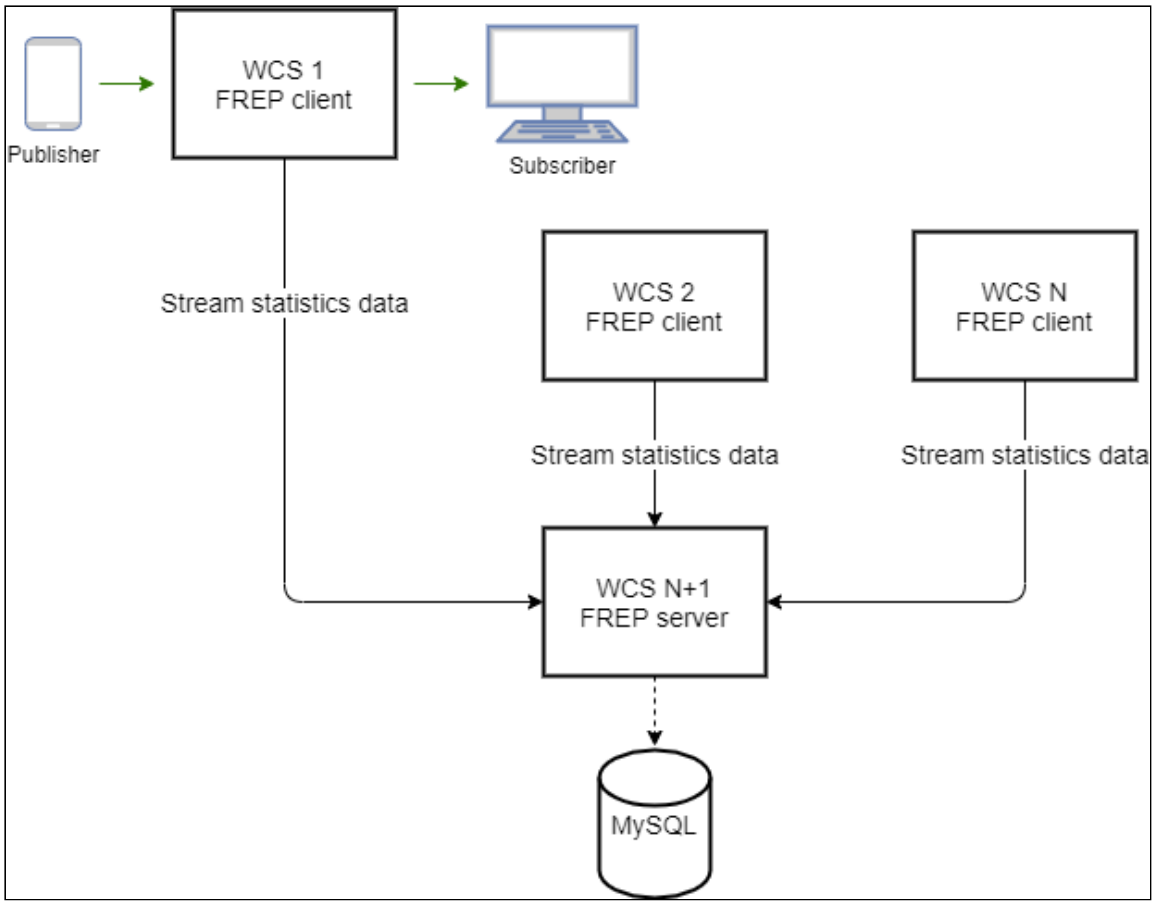
Overview

It may be necessary to collect stream statistics while managing a big number of WCS servers, to check broadcasting quality of service. If servers are in CDN, signaling events collection may also be useful to debug a network issues.

Since build 5.2.749 it is possible to collect stream statistics and CDN events centrally.

Architecture

Statistics collector server role can be assigned to one of WCS servers, in this case it also performs a usual media server functions (streams publishing and playback). Another servers send statistics data to the collector using a special TCP-based FREP protocol. Collector server writes received data to MySQL DB.



Staistics collector server role can be assigned to any WCS server. In theory, all the WCS servers can be collectors and send data directly to common MySQL server, but this can dramatically increase network load. In CDN, it is recommended to assign collector role to Controller node.

Data tables description

All the data are written to the following tables

Client sessions statistics

Field	Type	Description
serverIp	varchar(15)	WCS IP address
start	datetime	Connection establishing t ime
sessionId	varchar(300)	Session Id
status	varchar(30)	Session status

Field	Type	Description
info	varchar(300)	Reason phrase for FAILED status
duration	int(11)	Session duration

Client sessions statistics data are sent at session start and session end.

Stream statistics

Field	Type	Description
serverIp	varchar(15)	WCS IP address
start	datetime	Mediasession start time
sessionId	varchar(300)	Client session Id
mediaProvider	varchar(30)	Mediaprovider
name	varchar(300)	Stream name
mediaSessionId	varchar(300)	Mediasession Id
duration	int(32)	Mediasession duration
status	varchar(30)	Mediasession status
info	varchar(300)	Reason phrase for FAILED status
type	varchar(100)	Mediasession type (PUBLISH or SUBSCRIBE)
subscribers	int(16)	Subscribers count
audioLost	int(2)	Lost audio packets count
audioRate	int(4)	Audio bitrate
audioCodec	varchar(13)	Audio codec
videoWidth	int(2)	Picture width
videoHeight	int(2)	Picture height
videoPLI	int(2)	PLI requests count
videoNack	int(2)	NACK count
videoRate	int(4)	Video bitrate

Field	Type	Description
videoFPS	int(2)	Video FPS
videoBframes	int(2)	B-frames count
videoCodec	varchar(4)	Video codec
localSdp	varbinary(1500)	Server side SDP
remoteSdp	varbinary(1500)	Browser SDP

Stream statistics data are sent at stream publishing ir playback start and at stream end.

CDN events statistics

Field	Type	Description
serverIp	varchar(15)	WCS IP address
date	datetime	Event date and time
ts	varchar(30)	CDN signaling timestamp
srcNode	varchar(30)	CDN signaling source
dstNode	varchar(30)	CDN signaling destinatio n
event	varchar(60)	CDN event type
payload	varbinary(1500)	CDN signaling payload

CDN event statistics data are sent directly when event occurs. Therefore, CDN statisctics collection is disabled by default, and it is not recommended to collect the CDN statistics in production, this is for debug purposes only.

Configuration

MySQL server setup

Installation

Below, MySQL installation to Centos 7 example described:

1. Download the necessary repository

```
wget https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
```

2. Install the repository

```
sudo rpm -ivh mysql57-community-release-el7-9.noarch.rpm
```

3. Install MySQL

```
sudo yum install mysql-server
```

4. Launch MySQL

```
sudo systemctl start mysqld
```

Configuration

1. Create database and user for statistics collection

```
CREATE DATABASE wcs;
CREATE USER 'wcs'@'localhost' IDENTIFIED BY 'wcs';
GRANT ALL PRIVILEGES ON wcs.* TO 'wcs'@'localhost';
FLUSH PRIVILEGES;
```

2. Import database schema

```
mysql -u wcs -p wcs < wcs.sql
```

Database schema file wcs.sql Expand source

```
-- MySQL dump 10.13  Distrib 8.0.20, for Linux (x86_64)
--
-- Host: 172.17.0.2    Database: wcs
-- ------------------------------------------------------
-- Server version   5.7.31

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `cdn`
--
```

```sql
DROP TABLE IF EXISTS `cdn`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `cdn` (
  `serverIp` varchar(15) NOT NULL,
  `date` datetime NOT NULL,
  `ts` varchar(30) NOT NULL,
  `srcNode` varchar(30) NOT NULL,
  `dstNode` varchar(30) NOT NULL,
  `event` varchar(60) NOT NULL,
  `payload` varbinary(1500) NOT NULL,
  PRIMARY KEY (`ts`,`srcNode`,`event`,`dstNode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `conndr`
--

DROP TABLE IF EXISTS `conndr`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `conndr` (
  `serverIp` varchar(15) NOT NULL,
  `start` datetime NOT NULL,
  `sessionId` varchar(300) NOT NULL,
  `status` varchar(30) NOT NULL,
  `info` varchar(300) DEFAULT NULL,
  `duration` int(11) DEFAULT NULL,
  PRIMARY KEY (`sessionId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `sdr`
--

DROP TABLE IF EXISTS `sdr`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `sdr` (
  `serverIp` varchar(15) NOT NULL,
  `start` datetime NOT NULL,
  `sessionId` varchar(300) NOT NULL,
  `mediaProvider` varchar(30) NOT NULL,
  `name` varchar(300) NOT NULL,
  `mediaSessionId` varchar(300) NOT NULL,
  `duration` int(32) DEFAULT NULL,
  `status` varchar(30) NOT NULL,
  `info` varchar(300) DEFAULT NULL,
  `type` varchar(100) DEFAULT NULL,
  `subscribers` int(16) DEFAULT NULL,
  `audioLost` int(2) DEFAULT NULL,
  `audioRate` int(4) DEFAULT NULL,
  `audioCodec` varchar(13) DEFAULT NULL,
  `videoWidth` int(2) DEFAULT NULL,
  `videoHeight` int(2) DEFAULT NULL,
  `videoPLI` int(2) DEFAULT NULL,
```

```
  `videoNack` int(2) DEFAULT NULL,
  `videoRate` int(4) DEFAULT NULL,
  `videoFPS` int(2) DEFAULT NULL,
  `videoBframes` int(2) DEFAULT NULL,
  `videoCodec` varchar(4) DEFAULT NULL,
  `localSdp` varbinary(1500) DEFAULT NULL,
  `remoteSdp` varbinary(1500) DEFAULT NULL,
  PRIMARY KEY (`sessionId`,`mediaSessionId`,`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2020-08-26 11:30:24
```

WCS setup

Statistics collection can be enabled using the following parameter

```
frep_enabled=true
```

Role (client or server) can be set using the following parameter (CLIENT by default)

```
frep_role=CLIENT
```

Statistics collector server address can be set using the following parameter

```
frep_controller_ip=192.168.1.64
```

This parameter should be set on statistics client.

Database URI can be set using the following parameter

```
frep_database_address=jdbc:mysql://localhost/wcs?user=wcs&password=wcs
```

This parameter should be set on statistics server.

The port for statistics data transmission can be set using the following parameter (8085 by default)

```
frep_port=8085
```

Statistics authetication key can be set using the following parameter

```
frep_secret_key=dsjfoiewqhriywqtrfewfiuewqiufh
```

This parameter must be the same on statistics server and client.

Settings applying

All the settings can be applyed without server restart using CLI command

```
reload node-settings
```

In this case, it is recommended to change the settings by the following way:

1. Disable statistics collection on the WCS server

```
frep_enabled=false
```

2. Reload settings

```
reload node-settings
```

3. Set the necessary parameters and enable statistics collection

```
frep_enabled=true
```

4. Reload settings

```
reload node-settings
```

Data collection filter

Statistics data can be filtered before sending by the following parameter

```
frep_filter_events=CONNECT,CONNECTION_STATUS_EVENT,STREAM,STREAM_STATUS_EVENT
```

Possible filter values

Filter	Description
CONNECT	Connection establishing data
CONNECTION_STATUS_EVENT	Session status data
STREAM	Stream publishing/playing start data

Filter	Description
STREAM_STATUS_EVENT	Stream status data

For debug purposes, all the CDN events collection can be enabled

```
frep_filter_events=CONNECT,CONNECTION_STATUS_EVENT,STREAM,STREAM_STATUS_EVENT,CD
```

or CDN events can be filtered

Filter	Description
`ROLE_REFRESH`	CDN node role
NODE_REFRESH	CDN node information on entering to C DN
ROUTE_REFRESH	Stream playback route information
ROUTE_UPDATE	Stream playback route update informati on
EXTENDED_ROUTE_UPDATE	Extended stream playback route update information
EXTENDED_ROUTE_REFRESH	Extended stream playback route inform ation
VERSION_REFRESH	CDN version information
STATE_REFRESH	CDN node state information
GROUP_REFRESH	CDN group information
PROFILE_UPDATE	Transcoding profile update information
STREAM_ACCESS_LIST_REFRESH	Stream access keys information
STREAM_ACCESS_LIST_NODE_ID_UPD ATE	Node Id update for stream access keys information

Data retrieving from DB

Statistics data can be retrieved from MySQL DB using SQL queries

```
SELECT serverIP, start, name, status, audioCodec, videoCodec FROM wcs.sdr
WHERE name = 'test';
+---------------+--------------------+-------------+------------+-----------
```

```
-+------------+
| serverIP      | start               | name        | status     | audioCodec
| videoCodec |
+---------------+---------------------+-------------+------------+-----------
-+------------+
| 95.191.131.64 | 2020-08-10 13:44:00 | test        | PUBLISHING | opus
| H264       |
| 95.191.131.64 | 2020-08-10 13:44:10 | test        | PLAYING    | opus
| H264       |
+---------------+---------------------+-------------+------------+-----------
-+------------+
2 rows in set (0.00 sec)
```

SDP data retrieving

SDP data are compressed due to big size. To decompress data while retrieving, use xxd and zlib-flate utilities

```
mysql -u wcs -p -B --disable-column-names -e 'select hex(localSdp) from
wcs.sdr where name = "test"' | xxd -r -p | zlib-flate -uncompress
```

Attachments:

- FREP-arch.png (image/png)