Server tuning recommendations

Server default settings are mostly universal and need to be tuned to certain client case.

Server load and memory consumption optimization

Garbage collector tuning

Garbage collector (GC) is an important part of Java VM. When GC is running, itdramatically increases the server load and may stop another tasks execution, therefore it is recommended to minimize GC launch with the following settings inwcs-core.propertiesfile

```
#Disable heuristic rules
-XX:+UseCMSInitiatingOccupancyOnly

#Reduce Old Gen threshold
-XX:CMSInitiatingOccupancyFraction=70

# Use System.gc() concurrently in CMS
-XX:+ExplicitGCInvokesConcurrent

# Disable System.gc() for RMI, for 10000 hours
-Dsun.rmi.dgc.client.gcInterval=36000000000
-Dsun.rmi.dgc.server.gcInterval=36000000000
```

Heap memory tuning

Many objects with data are created and destroyed in memory while streaming. Therefore it is recommended to allocate at least 1/2 of server physical memory for Java memory heap. For example, if server RAM is 32 Gb, then it is recommended to allocate 16 Gb with the following settings inwcs-core.propertiesfile

```
-Xmx16g
-Xms16g
```

Besides, if monitoring functions in WCS web interface are not used, the work with memory may be optimized by switching off the interaction between WCS core and WCS manager with the setting inflashphoner.propertiesfile

```
disable_manager_rmi=true
```

REST client tuning

WhenREST hooksare used, on every WCS server action (establishing client connection, publishing and playing a stream, making a SIP call etc) HTTP REST connection to backend server is established.With a large number of simultaneously publishing clients or subscribers, with the default WCS settings it is possible to exhaust the WCS REST client thread pool, that is lead to deadlocks. Then, server stops to publish and play streams.

By default, a maximum number of simultaneous REST connections is set to 200 with the following parameterinflashphoner.propertiesfile

```
rest_max_connections=200
```

To escape thred poolexhausting and deadlocks this value should be reduced, for example

```
rest_max_connections=20
```

If REST hooksare not used, REST client can be disabled with the following parameter

```
disable_rest_requests=true
```

Excessive logging supression

WhenREST hooksare used, REST client operations, EchoApp default backend operations and REST API server operations are written to WCS core logs. That leads to large number of entries in the log file and, therefore, inceases the server load. The excessive logging may be decreased if necessary using the following parameters inlog4j.propertiesfile:

```
log4j.logger.RestClient=WARN
log4j.logger.EchoApp=WARN
log4j.logger.RestApiRouter=WARN
```

UDP tuning

Streaming mediadata are transferred with UDP packets. Those packets can be dropped, for example if server does not have enough time to parse packet queue, that leads to picture quality loss and freezes. To escape this it is necessary to tune UDP sockets buffers with the following settings inflashphoner.propertiesfile

```
rtp_receive_buffer_size=131072
rtp_send_buffer_size =131072
```

and to tune system queues with command

```
ip link set txqueuelen 2000 dev eth0
```

To diagnose UDP problem, it is necessary to track UDP packets dropping with command

```
dropwatch -l kas
>start
```

Channel load optimization

Users' playback picture quality depends on bitrate: the higher the bitrate, the higher the quality. However, the higher the bitrate, the higher data transfer channel load and, ifthe bandwidth between the server and clients is limited,there is a possibility that the channel will be fully loaded. This leads the bitrate droppingand a sharp decline in quality.

In this regard, it is necessary to limit the bitrate to ensuresufficient picture quality with an acceptable channel load.

Publisher bitrate limiting

To reduce the load to the channel from publisher to server, maximum and minimum bitrate values in kbps may be set in publisher script with JavaScript API

```
session.createStream({
    name: streamName,
    display: localVideo,
    constraints: {
        video: {
            minBitrate: 500
            maxBitrate: 1000
        }
    }
    ...
}).publish();
```

Server bitrate limiting

Minimum and maximum bitrate values in bps on server may be set with the following parameters inflashphoner.propertiesfile

```
webrtc_cc_min_bitrate=500000
webrtc_cc_max_bitrate=1000000
```

To exclude fast bitrate rise bu=y browser, the following parameter should be set

```
webrtc_cc2_twcc=false
```

Stream decoding on demand only must be switched on to reduce server load:

```
streaming_video_decoder_fast_start=false
```

Changing dynamic ports range in Linux

Dynamic or ephemeral port is a temporary port that is opened when establishing IP-connection from certain range of TCP/IP stack. Many Linux kernel versions use ports range 32768 — 61000 as dymanic ports. Enter the following command to check what range is used on server

```
sysctl net.ipv4.ip_local_port_range
```

If this range overlaps with WCS standard ports, it should be changed with the following command

```
sysctl -w net.ipv4.ip_local_port_range="59999 63000"
```

Adjusting the maximum number of opened files

In the launch script `webcallserver` that is in subfolder `bin` in WCS home folder, for example

```
/usr/local/FlashphonerWebCallServer/bin/webcallserver
```

in `start()` function the maximum number of opened files is set

```
function start() {
 ...
 echo -n $"$PRODUCT: starting"

 ulimit -n 20000
 if [[ "$1" == "standalone" ]]; then
     ...
 fi
 ...
}
```

By default, this value is set to 20000, but it may be increased if necessary, following the limitations of the operating system used.