Информация о нагрузке и ресурсах

- Статистические параметры
- Форматы вывода статистики
- Статистика транскодирования потоков
- Интегральные показатели нагрузки транскодирования
- Информация о CDN
- Синхронизация в опубликованных потоках
- Метрики опубликованных потоков
- Информация о выравнивающих буферах
- Статистика, собираемая при помощи внешнего скрипта
    - Ограничения
- Получение информации об ошибках Java VM
- Статистика сборщика мусора (garbage collector)
    - Выборка статистики сборщика мусора при использовании ZGC
- Статистика записи видео
- Статистика записи в логи
- Статистика входящего потока
- Статистика TCP каналов
- Статистика jitter буфера входящих медиа потоков
- Статистика H264 депакетайзера
- Статистика разбора аудио пакетов

Информация о нагрузке и ресурсах WCS-сервера доступна на порту 8081 по URL

```
http://test.flashphoner.com:8081/?action=stat
```

где test.flashphoner.com - это адрес WCS-сервера.

Она включает данные о публикуемых и воспроизводимых потоках, SIP-звонках и характеристиках сервера.

Статистические параметры

connections_stats	Соединения с WCS-сервером
connections	Общее количество соединений
connections_rtmfp	Количество RTMFP-соединений (Flash)
connections_websocket	Количество WebSocket-соединений
port_stats	**Порты, слушаемые WCS-сервером**
ports_media_free	Количество свободных медиапортов
ports_media_busy	Количество занятых медиапортов
ports_media_quarantine	Количество медиапортов в карантине
ports_wcs_agents_free	Количество свободных портов, используемых в CDN
ports_wcs_agents_busy	Количество занятых портов, используемых в CDN
ports_wcs_agents_quarantine	Количество портов, используемых в CDN, находящихся в карантине
streams_stats	**Сессии аудио и видеопотоков на WCS-сервере**
streams_rtsp_in	Количество активных RTSP-сессий, раздающих трафик
streams_rtsp_out	Количество активных RTSP-сессий, забирающих трафик
streams_websocket_out	Количество активных потоков, воспроизводимых через WebSocket
streams_rtmp_out	Количество активных RTMP-потоков
streams_webrtc_in	Количество активных потоков, публикуемых через WebRTC
streams_webrtc_out	Количество активных потоков, воспроизводимых через WebRTC
streams_rtmfp_in	Количество активных потоков, публикуемых через RTMFP
streams_rtmfp_out	Количество активных потоков, воспроизводимых через RTMFP
streams_viewers	Количество активных клиентов, воспроизводящих опубликованные потоки

streams_synchronization	Синхронизация в опубликованных потоках
native_resources	**Информация о потребляемых ресурсах**
native_resources	Информация о кодировании/декодировании медиапотока
core_stats	**Параметры системы (полученные с использованием методов Java) и параметры WCS-сервера**
core_threads	Количество активных потоков server-процесса WCS
core_java_threads	Текущее количество живых потоков
core_java_threads_blocked	Текущее количество заблокированных потоков
core_java_committedMemory	Количество гарантированно доступной виртуальной памяти, в байтах
core_java_freePhysicalMemorySize	Количество свободной физической памяти, в байтах
core_java_arch	Архитектура операционной системы
core_java_availableProcessors	Количество процессоров доступных виртуальной машине Java
core_java_freeSwapSpaceSize	Количество свободного swap-пространства, в байтах
core_java_maxFileDescriptorCount	Максимальное количество файл-дескрипторов
core_java_open_file_descriptors	Количество файл-дескрипторов, открытых в виртуальной машине Java
core_java_cpu_usage	"Недавнее использование процессора" для процесса виртуальной машины Java
core_java_totalPhysicalMemorySize	Количество физической памяти, в байтах
core_java_totalSwapSpaceSize	Количество swap-пространства, в байтах
core_java_uptime	Сколько времени прошло от запуска виртуальной машины Java, в миллисекундах
core_heap_memory_used	Использование памяти в куче
core_rss_memory	Использование памяти
core_open_file_descriptors	Количество открытых файл-дескрипторов
core_cpu_usage	Процент времени, затраченного на работу процессов
core_gc	Информация о "сборке мусора"
core_gc_manager	Информация о "сборке мусора" в административном модуле
core_heap_memory_used	Используемая память в куче
core_java_version	Версия Java
core_java_nio_memory_buffer_count	Количество используемых TCP NIO буферов
core_java_nio_memory_used	Объем памяти, используемой под TCP NIO буферы
core_java_nio_memory_capacity	Объем памяти, выделенной под TCP NIO буферы
call_stats	**SIP-звонки на WCS-сервере**
sip_calls	Количество SIP-звонков
sip_calls_established	Количество активных SIP-звонков
sip_calls_in	Количество входящих SIP-звонков
sip_calls_out	Количество исходящих SIP-звонков
sip_calls_per_second (cps)	Количество SIP-звонков в секунду
sip_stats	**SIP-клиенты**
sip_registered	Количество клиентов в состоянии REGISTERED
recording_stats	**Запись аудио и видеофайлов**
recording_sessions	Количество активных сессий записи
recording_threads	Количество процессорных потоков для записи

recording_thread_min_writers	Минимальное количество записей на один процессорный поток
recording_thread_max_writers	Максимальное количество записей на один процессорный поток
recording_thread_average_writers	Среднее количество записей на один процессорный поток
recording_writers_list	Список активных записей
recording_writers_with_max_queue	Список записей с максимальным размером очереди данных
recording_writers_with_min_queue	Список записей с минимальным размером очереди данных
recording_min_writers_queue	Минимальный размер очереди данных на запись
recording_average_writers_queue	Средний размер очереди данных на запись
recording_max_writers_queue	Максимальный размер очереди данных на запись
system_stats	**Параметры системы**
system_java_cpu_usage	"Недавнее использование процессора" для всей системы (получено с использованием методов Java)
system_java_load_average	Средняя нагрузка системы за последнюю минуту (получено с использованием методов Java)
system_uptime	Сколько времени прошло от запуска Linux-сервера
system_memory_total	Доступная к использованию RAM-память, в килобайтах
system_memory_free	Количество физической RAM-памяти, не используемой системой, в килобайтах
system_cpu_usage	Процент времени, затраченного на работу процессов ядра
network_stats	**Статистика сетевого трафика**
global_bandwidth_in	Пропускная способность канала для входящих соединений
global_bandwidth_out	Пропускная способность канала для исходящих соединений
version_stats	**Информация о версии WCS**
wcs_version	Текущая версия WCS-сервера
wcs_client_version	Текущая версия Web SDK
gc_stats	**Информация о последнем цикле сборки мусора**
gc_last_pause_ms	Пауза в работе JVM, в миллисекундах
gc_last_duration_ms	Общая длительность цикла сборки мусора, в миллисекундах
gc_last_heap_before_mb	Объем кучи в мегабайтах, до сборки мусора
gc_last_heap_after_mb	Объем кучи в мегабайтах, после сборки мусора
errors_info	**Информация об ошибках (собирается на основании перехваченных исключений Java)**
java.io.IOException	Количество ошибок ввода-вывода
java.lang. ArrayIndexOutOfBoundsException	Количество выходов за пределы массива
java.lang.IllegalArgumentException	Количество ошибок в аргументах функций
com.flashphoner.server.license. activation.A.C	Количество ошибок при активации лицензии
java.lang.NullPointerException	Количество попыток перехода по нулевому указателю
java.lang.OutOfMemoryError	Количество ошибок выделения памяти (требуется перезапуск сервера)
degraded_streams_stats	**Информация о деградации потоков**
degraded_streams	Количество деградировавших потоков
degraded_streams_percent	Процент деградации от общего числа потоков на сервере
transcoding_stats	**Информация о транскодировании потоков**
transcoding_video_decoding_resolutions	Список декодируемых разрешений в виде "разрешение/количество потоков"

transcoding_video_decoding_average_time	Среднее время декодирования по разрешениям в виде"разрешение/время в мс"
transcoding_video_decoding_max_time	Максимальное время декодирования по разрешениям в виде"разрешение/время в мс"
transcoding_video_decoding_average_queue_size	Средний размер очереди декодирования по разрешениям в виде"разрешение/размер очереди"
transcoding_video_decoding_max_queue_size	Максимальный размер очереди декодирования по разрешениям в виде"разрешение/размер очереди"
transcoding_video_encoding_resolutions	Список кодируемых разрешений в виде "разрешение/количество потоков"
transcoding_video_encoding_average_time	Среднее время кодирования по разрешениям в виде"разрешение/время в мс"
transcoding_video_encoding_max_time	Максимальное время кодирования по разрешениям в виде"разрешение/время в мс"
transcoding_video_encoding_average_queue_size	Средний размер очереди кодирования по разрешениям в виде"разрешение/размер очереди"
transcoding_video_encoding_max_queue_size	Максимальный размер очереди кодирования по разрешениям в виде"разрешение/размер очереди"
transcoding_video_decoding_load	Интегральный показатель нагрузки декодирования
transcoding_video_encoding_load	Интегральный показатель нагрузки кодирования
buffer_output_stats	**Информация о выравнивающем буфере (только в JSON формате)**
cdn_stats	**Информация о CDN**
cdn_version	Версия CDN, поддерживаемая данным сервером
cdn_role	Роль сервера в CDN
cdn_group	Группа, в которую входит сервер
log_stats	**Статистика записи в лог**
log_msg_per_sec	Количество сообщений, записываемых в логи, за секунду
log_mbit_per_sec	Объем записи в логи в мегабитах в секунду
tcp_channel_stats	**Статистика TCP каналов**
channels_not_writable	Количество событий, при которых не удалось отправить данные по TCP
jitter_buffer_stats	**Статистика jitter буфера для входящих медиа потоков**
decodable_drops_old	Количество сброшенных H264 фреймов, собранных из пакетов трафика, на поток
incomplete_drops_old	Количество сброшенных H264 фреймов, не полностью собранных из пакетов трафика, на поток
decodable_drops_reset	Количество сброшенных H264 фреймов до новой точки декодирования, собранных из пакетов трафика, на поток
incomplete_drops_reset	Количество сброшенных H264 фреймов до новой точки декодирования, не полностью собранных из пакетов трафика, на поток
decodable_drops_pli	Количество сбросов всех H264 фреймов, собранных из пакетов трафика, при приходе ключевого фрейма, на поток
incomplete_drops_pli	Количество сбросов всех H264 фреймов, не полностью собранных из пакетов трафика, при приходе ключевого фрейма, на поток
data_packets_with_empty_payload	Количество пакетов с пустым содержимым, высылаются браузером для оценки канала публикации при включенном TWCC
h264_depacketizer_stats	**Статистика H264 депакетайзера**
missed_h264_units	Количество потерянных H264 элементов, на поток
rtp_audio_player_stats	**Статистика разбора аудио пакетов**
dropped_audio_data_packets	Количество аудио пакетов, отброшенных на этапе передачи в движок сервера
custom_stats	**Статистика, собираемая при помощи внешнего скрипта**

Каждый параметр может быть запрошен отдельно, например

```
http://test.flashphoner.com:8081/?action=stat&params=native_resources
```

Статистику используемых ресурсов и информацию о нагрузке WCS-сервера,объединенную по группам, можно запросить по имени группы (доступны connections_stats, streams_stats, port_stats, call_stats, degraded_streams_stats, system_stats, core_stats)

```
http://test.flashphoner.com:8081/?action=stat&groups=call_stats
```

В запрос можно включить несколько наименований групп

```
http://test.flashphoner.com:8081/?action=stat&groups=call_stats,connections_stats
```

Параметры системы могут быть исключены из полной выборки при помощи настройки в файле flashphoner.properties

```
parse_system_stats=false
```

В этом случае параметры системы предоставляются только по запросу.

Форматы вывода статистики

По умолчанию, статистика выводится в plain text

Plain text statistics example

```
-----Connection Stats-----
connections=1
connections_rtmfp=0
connections_websocket=1
-----Port Stats-----
ports_media_free=495
ports_media_busy=4
ports_media_quarantine=0
-----Stream Stats-----
streams_webrtc_in=1
streams_webrtc_out=1
streams_websocket_out=0
streams_rtmfp_in=0
streams_rtmfp_out=0
streams_rtmp_in=0
streams_rtmp_out=0
streams_viewers=test/1
streams_rtsp_in=0
streams_rtsp_out=0
streams_rtmp_client_out=0
streams_play_rate=0
streams_stop_rate=0
-----Native Resources-----
native_resources=140537376620832,NENC:H264/FFMPEG,1859;140537236444800,FFDecoderNative:H264/FFMPEG,39404417
native_resources.audio_codecs=0
native_resources.audio_resamplers=0
native_resources.video_transcoders=0
native_resources.video_decoders=1
native_resources.video_encoders=1
native_resources.writers=0
-----Core Stats-----
core_java_threads=71
core_java_committedMemory=3127017472
core_java_freePhysicalMemorySize=69820416
core_java_arch=amd64
core_java_availableProcessors=2
core_java_freeSwapSpaceSize=1044369408
core_java_maxFileDescriptorCount=20000
core_java_open_file_descriptors=206
core_java_cpu_usage=39.88
core_java_totalPhysicalMemorySize=1927303168
core_java_totalSwapSpaceSize=1073737728
core_java_uptime=102191
-----Call Stats-----
sip_processed_calls=0
sip_calls_state=established/0;trying/0;ringing/0;ring/0;ring_media/0;hold/0;busy/0;finish/0;session_progress/0;
pending/0;failed/0
sip_calls=0
sip_calls_established=0
sip_calls_in=0
sip_calls_out=0
sip_calls_per_second=0.00
-----Sip Stats-----
sip_registered=0
-----Recording Stats-----
recording_sessions=0
-----System Stats-----
system_java_cpu_usage=75.00
system_java_load_average=0.87
-----Network Stats (Mbit/s)-----
global_bandwidth_in=0.000
global_bandwidth_out=0.000
-----Version info-----
wcs_version=5.2.416-32aab7dc90527bfe2ffb4711090fe75c6613a2bb
wcs_client_version=0.5.28.2753-9ba78705727d3ee9d75c1c10f488d21cd00a601c
-----Errors info-----
-----Degraded streams-----
degraded_streams=
degraded_streams_percent=0
```

Начиная со сборки 5.2.416, поддерживаются также форматы JSON и Prometheus. Для вывода статистики в формате JSON необходимо указать формат в URI запроса

```
curl -s 'http://localhost:8081/?action=stat&format=json'
```

Formatted JSON statistics example


```
{
  "connections_stats": {
    "connections": "1",
    "connections_rtmfp": "0",
    "connections_websocket": "1"
  },
  "port_stats": {
    "ports_media_free": "495",
    "ports_media_busy": "4",
    "ports_media_quarantine": "0"
  },
  "streams_stats": {
    "streams_webrtc_in": "1",
    "streams_webrtc_out": "1",
    "streams_websocket_out": "0",
    "streams_rtmfp_in": "0",
    "streams_rtmfp_out": "0",
    "streams_rtmp_in": "0",
    "streams_rtmp_out": "0",
    "streams_viewers": [
      "test/1"
    ],
    "streams_rtsp_in": "0",
    "streams_rtsp_out": "0",
    "streams_rtmp_client_out": "0",
    "streams_play_rate": "0",
    "streams_stop_rate": "0"
  },
  "native_resources": {
    "native_resources": [
      "140537376620832,NENC:H264/FFMPEG,9819",
      "140537236444800,FFDecoderNative:H264/FFMPEG,209561611"
    ],
    "native_resources.audio_codecs": "0",
    "native_resources.audio_resamplers": "0",
    "native_resources.video_transcoders": "0",
    "native_resources.video_decoders": "1",
    "native_resources.video_encoders": "1",
    "native_resources.writers": "0"
  },
  "core_stats": {
    "core_java_threads": "67",
    "core_java_committedMemory": "3127017472",
    "core_java_freePhysicalMemorySize": "73224192",
    "core_java_arch": "amd64",
    "core_java_availableProcessors": "2",
    "core_java_freeSwapSpaceSize": "1044107264",
    "core_java_maxFileDescriptorCount": "20000",
    "core_java_open_file_descriptors": "188",
    "core_java_cpu_usage": "37.19",
    "core_java_totalPhysicalMemorySize": "1927303168",
    "core_java_totalSwapSpaceSize": "1073737728",
    "core_java_uptime": "358833"
  },
  "call_stats": {
    "sip_processed_calls": "0",
    "sip_calls_state": [
      "established/0",
      "trying/0",
      "ringing/0",
      "ring/0",
```

```
      "ring_media/0",
      "hold/0",
      "busy/0",
      "finish/0",
      "session_progress/0",
      "pending/0",
      "failed/0"
    ],
    "sip_calls": "0",
    "sip_calls_established": "0",
    "sip_calls_in": "0",
    "sip_calls_out": "0",
    "sip_calls_per_second": "0.00"
  },
  "sip_stats": {
    "sip_registered": "0"
  },
  "recording_stats": {
    "recording_sessions": "0"
  },
  "system_stats": {
    "system_java_cpu_usage": "50.00",
    "system_java_load_average": "0.73"
  },
  "network_stats": {
    "global_bandwidth_in": "0.000",
    "global_bandwidth_out": "0.000"
  },
  "version_stats": {
    "wcs_version": "5.2.416-32aab7dc90527bfe2ffb4711090fe75c6613a2bb",
    "wcs_client_version": "0.5.28.2753-9ba78705727d3ee9d75c1c10f488d21cd00a601c"
  },
  "errors_info": {},
  "degraded_streams_stats": {
    "degraded_streams": [],
    "degraded_streams_percent": "0"
  }
}
```


Для вывода статистики в формате Prometheus необходимо указать формат в URI запроса

```
curl -s 'http://localhost:8081/?action=stat&format=prometheus'
```

Prometheus statistics example

```
connections_stats{param="connections"} 1
connections_stats{param="connections_rtmfp"} 0
connections_stats{param="connections_websocket"} 1
port_stats{param="ports_media_free"} 495
port_stats{param="ports_media_busy"} 4
port_stats{param="ports_media_quarantine"} 0
streams_stats{param="streams_webrtc_in"} 1
streams_stats{param="streams_webrtc_out"} 1
streams_stats{param="streams_websocket_out"} 0
streams_stats{param="streams_rtmfp_in"} 0
streams_stats{param="streams_rtmfp_out"} 0
streams_stats{param="streams_rtmp_in"} 0
streams_stats{param="streams_rtmp_out"} 0
streams_stats{param="streams_viewers",name="test"} 1
streams_stats{param="streams_rtsp_in"} 0
streams_stats{param="streams_rtsp_out"} 0
streams_stats{param="streams_rtmp_client_out"} 0
streams_stats{param="streams_play_rate"} 0
streams_stats{param="streams_stop_rate"} 0
native_resources{param="native_resources",id="140537376620832",name="NENC:H264/FFMPEG"} 11129
native_resources{param="native_resources",id="140537236444800",name="FFDecoderNative:H264/FFMPEG"} 239113192
native_resources{param="native_resources.audio_codecs"} 0
native_resources{param="native_resources.audio_resamplers"} 0
native_resources{param="native_resources.video_transcoders"} 0
native_resources{param="native_resources.video_decoders"} 1
native_resources{param="native_resources.video_encoders"} 1
native_resources{param="native_resources.writers"} 0
core_stats{param="core_java_threads"} 63
core_stats{param="core_java_committedMemory"} 3127017472
core_stats{param="core_java_freePhysicalMemorySize"} 67538944
core_stats{core_java_arch="amd64"} 1
core_stats{param="core_java_availableProcessors"} 2
core_stats{param="core_java_freeSwapSpaceSize"} 1043853312
core_stats{param="core_java_maxFileDescriptorCount"} 20000
core_stats{param="core_java_open_file_descriptors"} 188
core_stats{param="core_java_cpu_usage"} 37.02
core_stats{param="core_java_totalPhysicalMemorySize"} 1927303168
core_stats{param="core_java_totalSwapSpaceSize"} 1073737728
core_stats{param="core_java_uptime"} 401113
call_stats{param="sip_processed_calls"} 0
call_stats{param="sip_calls_state",name="established"} 0
call_stats{param="sip_calls_state",name="trying"} 0
call_stats{param="sip_calls_state",name="ringing"} 0
call_stats{param="sip_calls_state",name="ring"} 0
call_stats{param="sip_calls_state",name="ring_media"} 0
call_stats{param="sip_calls_state",name="hold"} 0
call_stats{param="sip_calls_state",name="busy"} 0
call_stats{param="sip_calls_state",name="finish"} 0
call_stats{param="sip_calls_state",name="session_progress"} 0
call_stats{param="sip_calls_state",name="pending"} 0
call_stats{param="sip_calls_state",name="failed"} 0
call_stats{param="sip_calls"} 0
call_stats{param="sip_calls_established"} 0
call_stats{param="sip_calls_in"} 0
call_stats{param="sip_calls_out"} 0
call_stats{param="sip_calls_per_second"} 0.00
sip_stats{param="sip_registered"} 0
recording_stats{param="recording_sessions"} 0
system_stats{param="system_java_cpu_usage"} 66.67
system_stats{param="system_java_load_average"} 0.84
network_stats{param="global_bandwidth_in"} 0.000
network_stats{param="global_bandwidth_out"} 0.000
version_stats{wcs_version="5.2.416-32aab7dc90527bfe2ffb4711090fe75c6613a2bb"} 1
version_stats{wcs_client_version="0.5.28.2753-9ba78705727d3ee9d75c1c10f488d21cd00a601c"} 1
degraded_streams_stats{param="degraded_streams"} 0
degraded_streams_stats{param="degraded_streams_percent"} 0
```

Статистика транскодирования потоков

В сборке5.2.443добавлена возможность вывода краткой и подробной статистики транскодирования потоков. Краткая статистика группируется по разрешениям и доступна во всех форматах, например:

```
-----Transcoding info-----
transcoding_video_decoding_resolutions=640x360/1
transcoding_video_decoding_average_time=640x360/2.0
transcoding_video_decoding_max_time=640x360/2
transcoding_video_decoding_average_queue_size=640x360/0.0
transcoding_video_decoding_max_queue_size=640x360/0
transcoding_video_encoding_resolutions=426x240/1;640x360/1;852x480/1
transcoding_video_encoding_average_time=426x240/2.0;640x360/2.0;852x480/6.0
transcoding_video_encoding_max_time=426x240/2;640x360/2;852x480/6
transcoding_video_encoding_average_queue_size=426x240/0.0;640x360/0.0;852x480/0.0
transcoding_video_encoding_max_queue_size=426x240/0;640x360/0;852x480/0
```

Здесь отображаются:

- разрешение и количество декодируемых и кодируемых потоков
- среднее и максимальное время декодирования и кодирования по разрешениям
- средний и максимальный размер очередей декодирования и кодирования

Подробная статистика выводится только в формате JSON, например

```
curl -s 'http://localhost:8081/?action=stat&format=json&groups=transcoding_stats' | jq
```

и группируется по опубликованным потокам

```
    "transcoding_video_full_info": {
      "test1": {
        "codec": "H264",
        "queueSize": 0,
        "distributors": [
          {
            "codec": "H264",
            "queueSize": 0,
            "resolution": "426x240",
            "averageProcessingTime": 3
          },
          {
            "codec": "H264",
            "queueSize": 0,
            "resolution": "640x360",
            "averageProcessingTime": 5
          },
          {
            "codec": "H264",
            "queueSize": 0,
            "resolution": "852x480",
            "averageProcessingTime": 10
          }
        ],
        "resolution": "640x360",
        "averageProcessingTime": 3
      }
    }
```

Здесь:

- codec - кодек потока
- queueSize - размер очереди
- resolution - разрешение потока
- averageProcessingTime - среднее время кодирования или декодирования
- distributors - параметры кодируемых (выходных) потоков

Интегральные показатели нагрузки транскодирования

В сборке5.2.570в статистику транскодирования добавлены интегральные показатели нагрузки декодирования и кодирования

```
-----Transcoding info-----
...
transcoding_video_decoding_load=22118400
...
transcoding_video_encoding_load=7983360
```

Нагрузка декодирования рассчитывается следующим образом

```
transcoding_video_decoding_load =  width * height * fps
```

Здесь

- width - ширина картинки декодируемого потока
- height - высотакартинки декодируемого потока
- fps - FPS декодируемого потока

Нагрузка кодирования рассчитывается следующим образом

```
transcoding_video_encoding_load =  width * height * fps
```

Здесь

- width - ширина картинки кодируемого потока согласно профилю
- height - высотакартинкикодируемого потока согласно профилю
- fps - FPS заданный профилем кодирования, либо FPS транскодируемого (исходного) потока

Информация о CDN

В сборке5.2.471добавлен вывод информации о CDN, в которой участвует данный сервер

```
curl -s 'http://localhost:8081/?action=stat&groups=cdn_stats'
```

```
-----CDN info-----
cdn_version=2.3
cdn_role=ORIGIN
cdn_group=
```

Здесь:

- cdn_version - версия CDN, поддерживаемая данным сервером
- cdn_role - роль сервера в CDN
- cdn_group - группа, к которой принадлежит сервер, либо пустая строка, если сервер не введен ни в одну группу

Синхронизация в опубликованных потоках

В сборке5.2.508в статистику потоков добавлен вывод информации о синхронизации аудио и видео в опубликованных потоках:

```
-----Stream Stats-----
streams_synchronization=stream1/90,stream2/-11
```

Значение метрик вычисляется как разность между текущими значениями аудио и видео синхронизации

```
var metric = lastAudioSync - lastVideoSync;
```

Таким образом, положительное значение метрики показывает, что аудио в данный момент обгоняет видео, а отрицательное - что аудио отстает от видео.

Изменения данной метрики в небольших пределах являются нормой. Если метрика для потока остается высоким и постоянным по знаку, это может означать проблемы с данным потоком.

Метрики опубликованных потоков

Начиная со сборки 5.2.518, поддерживается вывод метрик публикуемых потоков в формате Prometehus по запросу

```
curl -s 'http://localhost:8081/?action=stat&format=prometheus&groups=publish_streams'
```

Запрос возвращает набор метрик для каждого потока, опубликованного в данный момент на сервере, например

```
publish_streams{param="AUDIO_SYNC",name="test"} 3834464913756
publish_streams{param="AUDIO_CODEC",name="test"} 111
publish_streams{param="AUDIO_RATE",name="test"} 19192
publish_streams{param="AUDIO_LOST",name="test"} 0
publish_streams{param="VIDEO_SYNC",name="test"} 3834464913764
publish_streams{param="VIDEO_K_FRAMES",name="test"} 6
publish_streams{param="VIDEO_NACK",name="test"} 0
publish_streams{param="VIDEO_LOST",name="test"} 0
publish_streams{param="VIDEO_CODEC",name="test"} 119
publish_streams{param="VIDEO_B_FRAMES",name="test"} 0
publish_streams{param="VIDEO_PLI",name="test"} 0
publish_streams{param="VIDEO_RATE",name="test"} 377952
publish_streams{param="VIDEO_WIDTH",name="test"} 640
publish_streams{param="VIDEO_GOP_SIZE",name="test"} 60
publish_streams{param="VIDEO_HEIGHT",name="test"} 360
publish_streams{param="VIDEO_FPS",name="test"} 27
publish_streams{param="VIDEO_P_FRAMES",name="test"} 342
```

Если на сервере нет публикаций, то запрос вернет пустую страницу

Информация о выравнивающих буферах

В сборке5.2.543добавлена возможность вывода информации об использовании выравнивающих буферов на выходе транскодеров. Информация доступна только в формате JSON

```
curl -s 'http://localhost:8081/?action=stat&format=json&groups=buffer_stats'
```

и группируется по транскодируемым потока и подписчикам на эти потоки

```
{
  "buffer_stats": {
    "buffer_output_info": {
      "test": {
        "buffer_output_video_average": "0.0",
        "subscribers": {
          "25b94cd0-5eaf-11ea-a9b7-abda8d208547": {
            "overflows": 0,
            "buffer_output_audio": 4,
            "buffer_output_video": 0
          },
          ...
        },
        "buffer_output_audio_average": "4.0"
      }
    }
  }
}
```

Здесь

- buffer_output_audio_average - среднее количество аудио пакетов в буфере по всем подписчикам
- buffer_output_video_average - среднее количество видео пакетов в буфере по всем подписчикам

- buffer_output_audio - количество аудио пакетов в буфере для подписчика
- buffer_output_video -количество видео пакетов в буфере для подписчика
- overflows - количество переполнений буфера, при которых буфер очищается

Статистика, собираемая при помощи внешнего скрипта

В сборке5.2.579добавлена возможность сбора статистики, недоступной из JVM, при помощи вызова внешнего скрипта.

Имя скрипта указывается настройкой

```
custom_stats_script=/path/to/custom_stats.sh
```

По умолчанию, если указано только имя скрипта без пути, он должен располагаться в каталоге /usr/local/FlashphonerWebCallServer/bin

Скрипт должен возвращать параметры в виде

```
key1=value1
key2=value2
...
```

Пример скрипта, который возвращает версию Java и последнюю строку из лога сборщика мусора

```
#!/bin/bash

ver=$(java -version 2>&1 | grep "version" | cut -d" " -f 3 | sed 's/\"//g')
echo "java_ver=$ver"
gc_log=$( ls -t /usr/local/FlashphonerWebCallServer/logs/gc-core* | head -1)
echo "gc=$(tail -n1 $gc_log)"

exit 0
```

Этот скрипт добавит на страницу статистики

```
-----Custom info-----
java_ver=1.8.0_222
gc=2020-04-23T15:20:56.138+0700: 1546.835: [GC (Allocation Failure) 2020-04-23T15:20:56.138+0700: 1546.835:
[ParNew: 8978K->325K(9216K), 0.0103299 secs] 26379K->18056K(36172K), 0.0104582 secs] [Times: user
```

Параметры, возвращаемые скриптом, доступны во всех форматах статистики. В формате Prometheus ключ и значение преобразуются в метку:

```
custom_stats{java_ver="1.8.0_222"} 1
custom_stats{gc="2020-04-23T15:11:11.235+0700: 961.933: [GC (Allocation Failure) 2020-04-23T15:11:11.235+0700:
961.933: [ParNew: 9216K->793K(9216K), 0.0042971 secs] 26617K->18195K(36172K), 0.0044029 secs] [Times: user"} 1
```

Ограничения

1. Не допускается использование длительных операций и задержек в этом скрипте, поскольку это вызовет задержку выдачи страницы статистики.

2. Не допускается наличие кавычек в именах и значениям параметров, для совместимости с Prometheus. Например, значения

```
java_ver=openjdk version "1.8.0_222"
```

использовать нельзя, а

```
java_ver=1.8.0_222
```

можно

Получение информации об ошибках Java VM

В сборке5.2.769добавлена возможность получения информации о количестве определенных ошибок (исключений) Java VMпри помощи запроса

```
http://localhost:8081/?action=stat&params=wcs_errors,<exception class name>
```

Например, для того, чтобы своевременно контролировать нехватку физической памяти, необходимо периодически запрашивать

```
http://localhost:8081/?action=stat&params=wcs_errors,java.lang.OutOfMemoryError
```

Если полученный ответ содержит указанное исключение, и количество ошибок больше либо равно единицы

```
-----Errors info-----
java.lang.OutOfMemoryError=4
```

необходимо выводить клиентов и перезапускать сервер.

Статистика сборщика мусора (garbage collector)

В сборке5.2.897добавлена возможность получения статистики сборщика мусора (garbage collector)

```
http://localhost:8081/?action=stat&groups=gc_stats
```

В статистику входят следующие параметры последнего цикла сборки мусора на момент запроса:

```
-----Gc info-----
gc_last_pause_ms=62
gc_last_duration_ms=62
gc_last_heap_before_mb=315
gc_last_heap_after_mb=78
```

Здесь:

- gc_last_pause_ms - пауза в работе JVM, в миллисекундах
- gc_last_duration_ms - общая длительность сборки мусора, в миллисекундах
- gc_last_heap_before_mb - объем кучи в мегабайтах, до сборки мусора
- gc_last_heap_after_mb - объем кучи в мегабайтах, после сборки мусора

Информация выбирается из соответствующих параметров Java MX Beans.

Выборка статистики сборщика мусора при использовании ZGC

Для ZGC статистика выбирается из лога сборщика мусора, поскольку информация в Java MX Beans может возвращаться некорректно.
Поэтому при использовании ZGC в настройках необходимо указать

```
zgc_log_parser_enable=true
```

Также необходимо задать формат меток времени в логах.По умолчанию, используется полный формат с датой

```
zgc_log_time_format=yyyy-MM-dd'T'HH:mm:ss.SSSZ
```

Если в лог выводятся только секунды от начала работы JVM, формат задается как

```
zgc_log_time_format=ss.SSS
```

При необходимости, можно задать шаблон для поиска файлов лога.По умолчанию, предполагается, что имя файла содержит дату и время

```
zgc_log_parser_path=logs/gc-core-[0-9]{4}-[0-9]{2}-[0-9]{2}_[0-9]{2}-[0-9]{2}.log
```

Статистика записи видео

В сборке5.2.992добавлена возможность вывода статистики записи видео на диск. В статистику входят следующие параметры

```
-----Recording Stats-----
recording_sessions=10
recording_threads=8
recording_thread_min_writers=1
recording_thread_max_writers=2
recording_thread_average_writers=1
recording_writers_list=95c8f5d3/1;3881dab6/1;545e59b8/1;54e7a01c/1;06de077d/1;b71fa871/1;193aa3bb/1;91219b12/1;
c314201c/1;6f5241be/1
recording_writers_with_max_queue=95c8f5d3/3;545e59b8/3;06de077d/4;b71fa871/5;193aa3bb/5;91219b12/1;c314201c/7;
6f5241be/3
recording_writers_with_min_queue=95c8f5d3/3;545e59b8/3;06de077d/4;b71fa871/5;193aa3bb/5;91219b12/1;c314201c/7;
6f5241be/3
recording_min_writers_queue=1
recording_average_writers_queue=4
recording_max_writers_queue=7
```

Здесь отображаются:

- количество процессорных потоков, используемых для записи на диск
- количество активных записей
- количество записей на один процессорный поток
- размеры очередей данных на запись

Например, если размеры очередей начинают устойчиво расти, и при этом загрузка процессора мала (до 25%), это может говорить о том, что данные не успевают сброситься на диск. Если же загрузка процессора при росте очередей велика, это означает. что не хватает мощности процессора для транскодинга звука или видео при записи.

Подробная статистика записи выводится в формате JSON

```
http://localhost:8081/?action=stat&format=json&groups=recording_stats
```

```
{
  "recording_stats": {
    "recording_sessions": "10",
    "recording_threads": "8",
    "recording_thread_min_writers": "1",
    "recording_thread_max_writers": "2",
    "recording_thread_average_writers": "1",
    "recording_writers_list": [
      "95c8f5d3",
      "3881dab6",
      "545e59b8",
      "54e7a01c",
      "06de077d",
      "b71fa871",
      "193aa3bb",
      "91219b12",
      "c314201c",
      "6f5241be"
    ],
    ...
    "recording_full_info": [
      {
        "threadId": 109,
        "writersCount": 2,
        "usageCounter": 2,
        "writersInfo": [
          {
            "queueSize": 0,
            "streams": [
              "95c8f5d3"
            ]
          },
          {
            "queueSize": 0,
            "streams": [
              "3881dab6"
            ]
          }
        ]
      },
      ...
    ]
  }
}
```

Здесь:

- queueSize - текущий размер очереди данных на запись
- streams - список записываемых потоков
- threadId - идентификатор процессорного потока, который занимается записью
- writersCount - количество записей на один процессорный поток
- usageCount - количество использований процессорного потока для записи

Статистика записи в логи

В сборке 5.2.1210 добавлена возможность получения статистики записи в логи, что может быть полезно на сервере под нагрузкой

```
curl -s 'http://localhost:8081/?action=stat&groups=log_stats'
```

Сбор статистики логирования дает дополнительную нагрузку на CPU при большом объеме медиапотоков на сервере. Поэтому, начиная со сборки 5.2.1252 по умолчанию, сбор статистики логирования по умолчанию отключен, но может быть включен при помощи настройки

```
log_metrics_stats=true
```

В статистику входят следующие параметры

```
-----Logger info-----
log_msg_per_sec=0.30
log_mbit_per_sec=0.00
```

- log_msg_per_sec - количество сообщений, записанных во все логи за секунду
- log_mbit_per_sec - объем данных, записанных во все логи, в мегабитах в секунду

По умолчанию, статистика буферизуется на интервале 10 секунд. Если в лог в течение этого времени не было выведено ни одного сообщения, значения в статистике обнулятся. Интервал буферизации может быть изменен настройкой

```
log_metrics_time_buffer=10000
```

Интервал не может быть задан меньше 1 секунды, в таких случаях будет использовано значение по умолчанию.

Статистика входящего потока

В сборке 5.2.1257 добавлена возможность получения статистики публикации конкретного потока

```
curl -s "http://localhost:8081/?action=stat&format=json&groups=transcoding_stats" | jq '.[].
transcoding_video_full_info'
```

Статистика выводится только в формате JSON и содержит следующие параметры:

```
{
  "test": {
    "codec": "H264",
    "queueSize": 0,
    "distributors": {
      ...
    },
    "minDeltaArrivalTime": 1,
    "maxDeltaArrivalTime": 62,
    ...,
    "streamDelay": 38
  }
}
```

- имя потока
- codec - кодек публикации
- queueSize - размер очереди декодирования
- distributors - информация о подписчиках
- minDeltaArrivalTime - минимальное время между получением двух последовательных пакетов, мс
- maxDeltaArrivalTime - минимальное время между получением двух последовательных пакетов, мс
- streamDelay - задержка потока, мс

Статистика по времени собирается на интервале, который задается настройкой в миллисекундах

```
media_processor_incoming_stat_window=30000
```

Статистика TCP каналов

В сборке 5.2.1883 добавлен вывод статистики TCP канального уровня

```
curl -s "http://localhost:8081/?action=stat&groups=tcp_channel_stats"
```

Статистика содержит следующие параметры

```
-----Tcp channel info-----
channels_not_writable=
```

- channels_not_writable - количество событий, в результате которых не удалось записать данные на отправку в TCP сокет, на медиасессию

Статистика отображается только для текущих публикаций.

Статистика jitter буфера входящих медиа потоков

В сборке 5.2.1883 добавлен вывод статистики jitter буфера входящих медиа потоков

```
curl -s "http://localhost:8081/?action=stat&groups=jitter_buffer_stats"
```

Статистика содержит следующие параметры

```
-----JitterBuffer info-----
decodable_drops_old=
incomplete_drops_old=f49f8cb0-dc52-11ee-81df-51ad589334c0/7;
decodable_drops_reset=
incomplete_drops_reset=
decodable_drops_pli=
incomplete_drops_pli=f49f8cb0-dc52-11ee-81df-51ad589334c0/10;
data_packets_with_empty_payload=
```

- decodable_drops_old - количество сброшенных H264 фреймов, собранных из пакетов трафика, на медиасессию
- incomplete_drops_old - количество сброшенных H264 фреймов, не полностью собранных из пакетов трафика, на медиасессию
- decodable_drops_reset - количество сброшенных H264 фреймов до новой точки декодирования, собранных из пакетов трафика, на медиасессию
- incomplete_drops_reset - количество сброшенных H264 фреймов до новой точки декодирования, не полностью собранных из пакетов трафика, на медиасессию
- decodable_drops_pli - количество сбросов всех H264 фреймов, собранных из пакетов трафика, при приходе ключевого фрейма, на медиасессию
- incomplete_drops_pli - количество сбросов всех H264 фреймов, не полностьюсобранных из пакетов трафика, при приходе ключевого фрейма, на медиасессию
- data_packets_with_empty_payload - количество пакетов с пустым содержимым, высылаются браузером для оценки канала публикации при включенном TWCC, на медиасессию

Статистика отображается только для текущих публикаций

Статистика H264 депакетайзера

В сборке 5.2.1883 добавлен вывод статистики H264 депакетайзера

```
curl -s "http://localhost:8081/?action=stat&groups=h264_depacketizer_stats"
```

Статистика содержит следующие параметры

```
-----H264 RTPDepacketizer info-----
missed_h264_units=f49f8cb0-dc52-11ee-81df-51ad589334c0/443;
```

- missed_h264_units - количество потерянных H264 элементов, на медиасессию

Статистика отображается только для текущих публикаций

Статистика разбора аудио пакетов

В сборке 5.2.1923 добавлен вывод статистики разбора аудио пакетов

```
curl -s "http://localhost:8081/?action=stat&groups=rtp_audio_player_stats"
```

Статистика содержит следующие параметры

```
-----RtpAudioPlayer info-----
dropped_audio_data_packets=
```

- dropped_audio_data_packets - количество аудио пакетов, отброшенных на этапе передачи в движок сервера, на медиасессию

Статистика отображается только для текущих публикаций