

Централизованный сбор данных о работе серверов в БД ClickHouse

- [Описание](#)
- [Архитектура](#)
 - [Описание таблиц данных](#)
 - [Данные соединений \(таблица ConnectionEvent\)](#)
 - [Типы событий \(таблица ConnectionEventTypes\)](#)
 - [Словарь типов событий \(таблица DictionaryConnectionEvents\)](#)
 - [Данные потоков \(таблица StreamEvent\)](#)
 - [Типы событий \(таблица StreamEventTypes\)](#)
 - [Словарь типов событий \(таблица DictionaryStreamEvents\)](#)
 - [Данные CDN \(таблица CDNEvent\)](#)
 - [Типы событий \(таблица CDNEventTypes\)](#)
 - [Словарь типов событий \(таблица DictionaryCDNEvents\)](#)
 - [Настройка](#)
 - [Установка и настройка ClickHouse](#)
 - [Требования к серверу](#)
 - [Установка ClickHouse на примере CentOS 7](#)
 - [Настройка ClickHouse](#)
 - [Настройка WCS](#)
 - [Остановка сбора данных без перезапуска WCS](#)
 - [Изменение адреса сервера ClickHouse без перезапуска WCS](#)
 - [Выборки информации из БД](#)

Описание

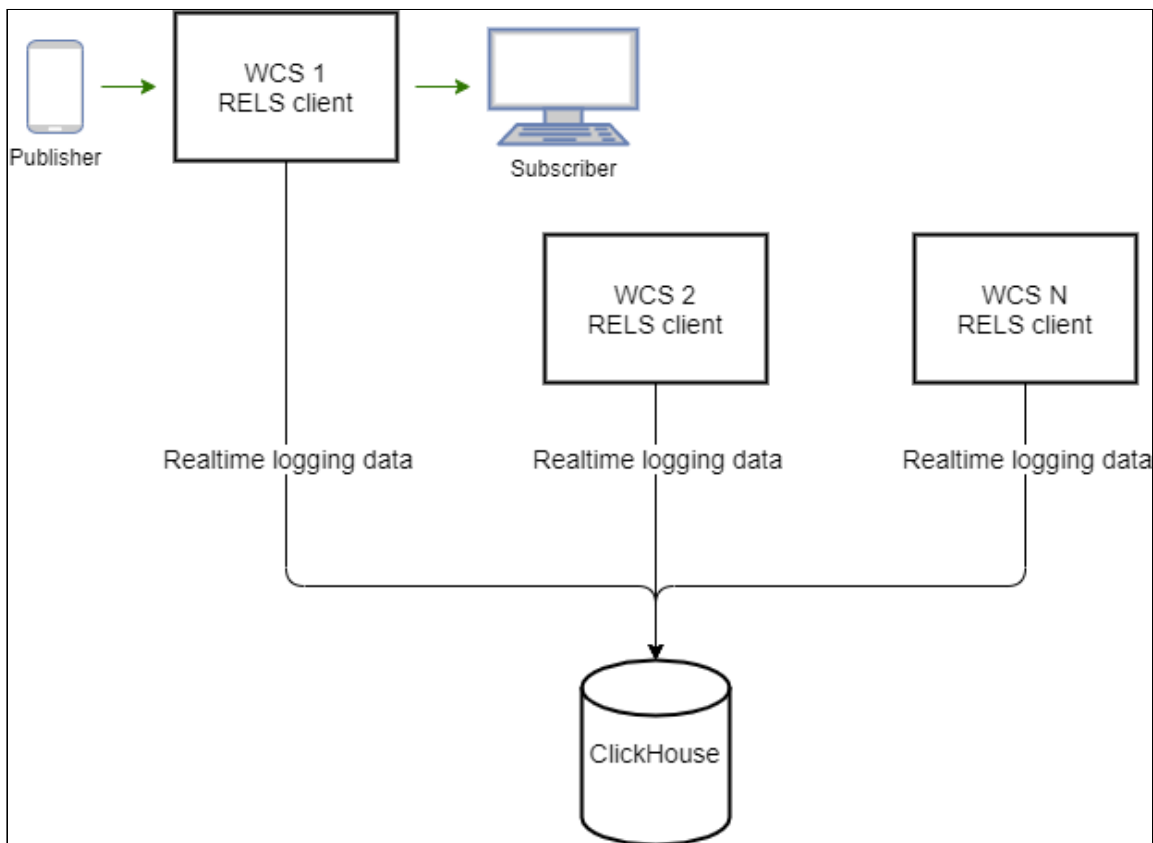
При управлении большим количеством WCS серверов, для отладки возможных проблем с вещанием потоков, возникает необходимость централизованного сбора данных о потоках, клиентских соединениях и событиях CDN. Фактически, необходимо в одной точке собрать информацию, которая пишется в логи каждого сервера. При

этом само по себе логирование в промышленной эксплуатации сведено к минимуму, чтобы не давать нагрузку на дисковую подсистему сервера.

Для сбора данных в реальном времени в больших объемах хорошо подходят базы данных временных рядов. На основе одной из таких БД с открытым исходным кодом [ClickHouse](#), в сборке [5.2.774](#) добавлена система удаленного сбора логов RELS (Remote Event Logging System).

Архитектура

Каждый WCS сервер отправляет данные в ClickHouse независимо, используя JDBC-драйвер и HTTP-соединение. Для оптимизации работы с БД ClickHouse, данные буферизуются и отправляются пачками по времени



Описание таблиц данных

Данные собираются в таблицы ClickHouse, перечисленные ниже. При этом, в таблицу записывается числовой идентификатор события. Для того, чтобы отобразить события в выборках в удобном для чтения виде, каждой таблице сопоставлен словарь текстовых строк, описывающих события.

Данные соединений (таблица ConnectionEvent)

Поле	Тип	Описание
timestamp	UInt64	Метка времени
ip	IPv4	Адрес сервера
sessionId	String	Идентификатор сессии
eventType	UInt64	Идентификатор типа события
eventPayload	String	Содержимое события

Типы событий (таблица ConnectionEventTypes)

Поле	Тип	Описание
id	UInt32	Идентификатор типа события
type	String	Описание типа события

Словарь типов событий (таблица DictionaryConnectionEvents)

Поле	Тип	Описание
id	UInt64	Идентификатор типа события
type	String	Описание типа события

Данные потоков (таблица StreamEvent)

Поле	Тип	Описание
timestamp	UInt64	Метка времени
ip	IPv4	Адрес сервера
sessionId	String	Идентификатор сессии
mediaSessionId	String	Идентификатор медиасессий
streamName	String	Имя потока
eventType	UInt64	Идентификатор типа события

Поле	Тип	Описание
eventPayload	String	Содержимое события

Типы событий (таблица StreamEventTypes)

Поле	Тип	Описание
id	UInt32	Идентификатор типа события
type	String	Описание типа события

Словарь типов событий (таблица DictionaryStreamEvents)

Поле	Тип	Описание
id	UInt64	Идентификатор типа события
type	String	Описание типа события

Данные CDN (таблица CDNEvent)

Поле	Тип	Описание
timestamp	UInt64	Метка времени
ip	IPv4	Адрес сервера
nodeId	String	Идентификатор узла (с трока IP адреса)
eventType	UInt64	Идентификатор типа события
eventPayload	String	Содержимое события

Типы событий (таблица CDNEventTypes)

Поле	Тип	Описание
id	UInt32	Идентификатор типа события
type	String	Описание типа события

Словарь типов событий (таблица DictionaryCDNEvents)

Поле	Тип	Описание
id	UInt64	Идентификатор типа события
type	String	Описание типа события

Настройка

Установка и настройка ClickHouse

Требования к серверу

- CPU не менее 4 физических ядер, частотой не менее 3 ГГц, например Intel(R) Xeon(R) CPU E3-1246 v3 @ 3.50GHz
- RAM не менее 32 Гб
- HDD не менее 2 Тб

Установка ClickHouse на примере CentOS 7

1. Создайте файл репозитория `altinity_clickhouse.repo` в каталоге `/etc/yum.repos.d`

```
sudo cat <<EOF > /etc/yum.repos.d/altinity_clickhouse.repo
[altinity_clickhouse]
name=altinity_clickhouse
baseurl=https://packagecloud.io/altinity/clickhouse/el/7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/altinity/clickhouse/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300

[altinity_clickhouse-source]
name=altinity_clickhouse-source
baseurl=https://packagecloud.io/altinity/clickhouse/el/7/SRPMS
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/altinity/clickhouse/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
EOF
```

2. Подключите репозиторий

```
sudo yum -q makecache -y --enablerepo='altinity_clickhouse'
```

3. Установите ClickHouse

```
sudo yum install -y clickhouse-server clickhouse-client
```

4. Запустите ClickHouse

```
systemctl start clickhouse-server
```

Настройка ClickHouse

1. Для того, чтобы прослушивать входящие запросы на всех интерфейсах сервера, раскомментируйте строку в файле /etc/clickhouse-server/config.xml

```
<listen_host>:::</listen_host>
```

2. Для того, чтобы создать пользователя, укажите для пользователя default в файле /etc/clickhouse-server/users.xml параметр

```
<access_management>1</access_management>
```

3. Перезапустите ClickHouse

```
systemctl restart clickhouse-server
```

4. Создайте базу данных wcs и таблицы в ней

```
cat wcs_clickhouse.sql | clickhouse-client -mn
```

wcs_clickhouse.sql Expand source

```
CREATE DATABASE IF NOT EXISTS wcs;  
  
DROP TABLE IF EXISTS wcs.StreamEvent;  
  
DROP TABLE IF EXISTS wcs.ConnectionEvent;  
  
DROP TABLE IF EXISTS wcs.CDNEvent;  
  
DROP TABLE IF EXISTS wcs.StreamEventTypes;  
  
DROP TABLE IF EXISTS wcs.ConnectionEventTypes;  
  
DROP TABLE IF EXISTS wcs.CDNEventTypes;  
  
DROP DICTIONARY IF EXISTS wcs.DictionaryStreamEvents;
```

```

DROP DICTIONARY IF EXISTS wcs.DictionaryConnectionEvents;

DROP DICTIONARY IF EXISTS wcs.DictionaryCDNEvents;

CREATE TABLE wcs.ConnectionEventTypes
(
    `id` UInt32,
    `type` String
)
ENGINE = MergeTree()
ORDER BY id
SETTINGS index_granularity = 8192;

INSERT INTO wcs.ConnectionEventTypes VALUES (0, 'CONNECTED'), (1,
'DISCONNECTED');

CREATE TABLE wcs.StreamEventTypes
(
    `id` UInt32,
    `type` String
)
ENGINE = MergeTree()
ORDER BY id
SETTINGS index_granularity = 8192;

INSERT INTO wcs.StreamEventTypes VALUES (0, 'CREATED'),
(1, 'LOCAL_SDP_CREATED'), (2, 'REMOTE_SDP_RECEIVED'), (3, 'ICE_STARTED'),
(4, 'ICE_COMPLETE'), (5, 'DTLS_STARTED'), (6, 'DTLS_COMPLETE'), (7, 'INITIALIZED'),
(8, 'DISPOSING'), (9, 'DISPOSED'), (10, 'AUDIO_RECEIVED'), (11, 'VIDEO_RECEIVED'),
(12, 'VIDEO_KFRAME_RECEIVED'), (13, 'AUDIO_RTCP_RECEIVED'),
(14, 'VIDEO_RTCP_RECEIVED'), (15, 'RESOLUTION_RECEIVED'),
(16, 'VIDEO_ENCODER_CREATED'), (17, 'AUDIO_ENCODER_CREATED'),
(18, 'VIDEO_ENCODER_DISPOSED'), (19, 'AUDIO_ENCODER_DISPOSED'),
(20, 'TERMINATED'), (21, 'AUDIO_SENT'), (22, 'VIDEO_SENT'),
(23, 'VIDEO_JITTER_BUFFER_STALL'), (24, 'SENT_PLI'), (25, 'RECEIVED_PLI'),
(26, 'SYNC_BUFFER_FULL'), (27, 'SYNC_FORCE_FAILED'), (28, 'SYNC_SHIFT'),
(29, 'SYNC_DEVIATION'), (30, 'VIDEO_STATS'), (31, 'RECORD');

CREATE TABLE wcs.CDNEventTypes
(
    `id` UInt32,
    `type` String
)
ENGINE = MergeTree()
ORDER BY id
SETTINGS index_granularity = 8192;

INSERT INTO wcs.CDNEventTypes VALUES (0, 'STATE'), (1, 'CDN_STATE'), (2,
'VERSION'), (3, 'ACL_REFRESH'), (4, 'ACL_UPDATE');

CREATE DICTIONARY wcs.DictionaryStreamEvents (
    `id` UInt16,
    `type` String DEFAULT ''
)
PRIMARY KEY id
SOURCE(CLICKHOUSE(
    host 'localhost'
    port 9000

```

```

user 'default'
password ''
db 'wcs'
table 'StreamEventTypes'
))
LAYOUT(FLAT())
LIFETIME(300);

CREATE DICTIONARY wcs.DictionaryConnectionEvents (
    `id` UInt16,
    `type` String DEFAULT ''
)
PRIMARY KEY id
SOURCE(CLICKHOUSE(
    host 'localhost'
    port 9000
    user 'default'
    password ''
    db 'wcs'
    table 'ConnectionEventTypes'
))
LAYOUT(FLAT())
LIFETIME(300);

CREATE DICTIONARY wcs.DictionaryCDNEvents (
    `id` UInt16,
    `type` String DEFAULT ''
)
PRIMARY KEY id
SOURCE(CLICKHOUSE(
    host 'localhost'
    port 9000
    user 'default'
    password ''
    db 'wcs'
    table 'CDNEventTypes'
))
LAYOUT(FLAT())
LIFETIME(300);

CREATE TABLE wcs.StreamEvent
(
    `timestamp` UInt64,
    `ip` IPv4,
    `sessionId` String,
    `mediaSessionId` String,
    `streamName` String,
    `eventType` UInt64,
    `eventPayload` String
)
ENGINE = MergeTree()
ORDER BY (sessionId, mediaSessionId, streamName)
SETTINGS index_granularity = 8192;

CREATE TABLE wcs.ConnectionEvent
(
    `timestamp` UInt64,
    `ip` IPv4,

```



```

        `sessionId` String,
        `eventType` UInt64,
        `eventPayload` String
    )
ENGINE = MergeTree()
ORDER BY (timestamp, sessionId)
SETTINGS index_granularity = 8192;

CREATE TABLE wcs.CDNEvent
(
    `timestamp` UInt64,
    `ip` IPv4,
    `nodeId` String,
    `eventType` UInt64,
    `eventPayload` String
)
ENGINE = MergeTree()
ORDER BY (nodeId, eventType)
SETTINGS index_granularity = 8192;

```

5. Создайте пользователя wcs и дайте ему права на таблицы в базе данных wcs

```
cat wcs_clickhouse_users.sql | clickhouse-client -mn
```

wcs_clickhouse_users.sql Expand source

```
CREATE USER IF NOT EXISTS wcs IDENTIFIED BY 'wcs';
GRANT ALL ON wcs.* TO wcs WITH GRANT OPTION;
```

6. Отключите управление пользователями для пользователя default, указав в файле /etc/clickhouse-server/users.xml параметр

```
<access_management>0</access_management>
```

7. Перезапустите ClickHouse

```
systemctl restart clickhouse-server
```

Настройка WCS

Сбор данных в БД ClickHouse включается настройкой

```
rels_enabled=true
```

Адрес сервера ClickHouse и базы данных задается настройкой

```
rels_database_address=jdbc:clickhouse://clickhouseserver:8123/wcs?
user=wcs&password=wcs
```

Остановка сбора данных без перезапуска WCS

При необходимости, передача данных с конкретного WCS сервера в ClickHouse может быть остановлена без перезапуска WCS. Для этого:

1. Отключите сбор данных в настройках сервера

```
rels_enabled=false
```

2. Перезагрузите настройки сервера из [интерфейса командной строки](#)

```
reload node-settings
```

Изменение адреса сервера ClickHouse без перезапуска WCS

Адрес сервера ClickHouse может быть изменен без перезапуска WCS. Для этого:

1. Измените адрес в настройках сервера

```
rels_database_address=jdbc:clickhouse://newclickhouseserver:8123/wcs?  
user=wcs&password=wcs
```

2. Отключите сбор данных в настройках сервера

```
rels_enabled=false
```

3. Перезагрузите настройки сервера из [интерфейса командной строки](#)

```
reload node-settings
```

4. Включите сбор данных в настройках сервера

```
rels_enabled=true
```

5. Перезагрузите настройки сервера из [интерфейса командной строки](#)

```
reload node-settings
```

Выборки информации из БД

Выборки информации из БД производятся при помощи SQL запросов в клиенте ClickHouse

```
select
timestamp,ip,sessionId,mediaSessionId,streamName,dictGetString('wcs.DictionarySt
eventType) as eventType from wcs.StreamEvent where streamName = 'test'
```

```
select
timestamp,ip,sessionId,dictGetString('wcs.DictionaryConnectionEvents','type',
eventType) as eventType from wcs.ConnectionEvent
```

```
select timestamp,ip,nodeId,dictGetString('wcs.DictionaryCDNEvents','type',
eventType) as eventType,eventPayload from wcs.CDNEvent
```

Attachments:

- [RELS.png](#) (image/png)