



# **Streaming Media Capacity Planning Guide**

# Table of Contents

I. Introduction .....	<b>3</b>
Usage Patterns .....	3
Video-On-Demand (VOD) .....	3
Live events .....	4
Live stream recording .....	5
Typical institution/enterprise experience .....	5
II. Monitoring .....	<b>6</b>
Wowza Streaming Engine .....	6
Monitoring with Wowza Streaming Engine Manager .....	6
Other Monitoring Tools.....	7
Windows Task Manager .....	7
Windows Resource Monitor .....	7
Standalone monitoring tools .....	7
III. Load Testing.....	<b>7</b>
The Wowza Load Test Tool .....	8
Testing Live Stream Recording Performance.....	9
Configure live stream playlist.....	10
Configure stream targets.....	10
IV. Scaling Capacity.....	<b>12</b>
Infrastructure Adjustments.....	12
Wowza performance tuning.....	12
Single purpose servers .....	12
Network capacity .....	12
Disk technology .....	12
Architectural Extensions .....	13
Scaling and load balancing.....	13
Appendix A – Video On Demand Typical Usage.....	<b>15</b>
Appendix B – Sample Load Test Tool XML file.....	<b>18</b>
Appendix C – Sample “Live” Stream Playlist SMIL file.....	<b>20</b>

# I. Introduction

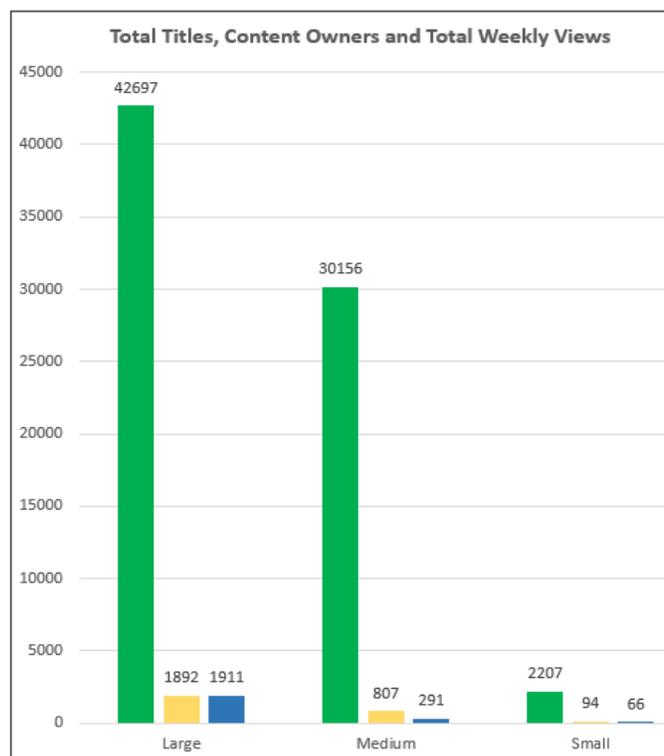
Media streaming is a powerful tool for delivering media to Internet connected viewers, and most of today’s progressive organizations are leveraging this capability for training, instruction, entertainment, event sharing, and so forth. Customers that make serious use of streaming should be certain they have the streaming server/network capacity and architecture to satisfy anticipated viewer demand for streaming content. This document introduces the concepts, tools and mechanisms that can be used to ensure streaming success across three streaming applications in educational and enterprise settings, where video use is focused primarily on internal audiences:

1. **Video-on-demand (VOD) streaming** - This is the most commonly used streaming mode where an enterprise or institution has an archive of videos in storage that they share with viewers via on-demand streams.
2. **Live event streaming** - Many Ensemble Video customers live stream lectures, product announcements, sporting events, public meetings, graduation ceremonies, class meetings and other similar events where there is real value to an audience for live viewing access to the proceedings.
3. **Live stream recording** - Ensemble Video stream recording automatically records live streams, so that the resulting video recording can be immediately available to users for on-demand access at the conclusion of the live stream. Ensemble Studio uses this with Matrox Monarch streaming appliances for automated room capture.

## Usage Patterns

### Video-On-Demand (VOD)

For most of our customers VOD media is internally focused “long tail” content that is viewed very modestly. We examined VOD usage for a typical week for three Ensemble Video implementations: one Large multi-tenant implementation, one Medium sized enterprise implementation, and one very Small implementation. The number of titles and content owners varied significantly across these three Ensemble Video implementations as you can see from “Total Titles, Content Owners and Total Weekly Views”.



For the three implementations, VOD usage was also consistently modest. The average number of video views for the items viewed for that week was consistently below a dozen views per item on average. With VOD streaming the content is available 24x7, so demand is evened out, there are few spikes in concurrent viewing, especially as the number of overall titles increases. For more details see Appendix A.

<b>Large</b>	Concurrent viewers never exceeded 100
<b>Medium</b>	Concurrent viewers never exceeded 30
<b>Small</b>	Concurrent viewers never exceeded 30

### Live events

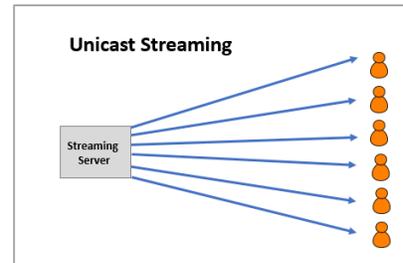
For many live events, customers actively drive traffic to the live stream and generate a clear spike in use. Even so, many events have only dozens of viewers, and high-end on these sorts of live events (e.g., university graduation) often peak out at a few hundred viewers (many more may view a recording of the live stream that is made available on-demand after the event, decreasing the need for real-time viewing). But some events, such as an important company-wide CEO address, can spike to thousands of concurrent viewers. It is critical to monitor usage and anticipate where certain events can cause unusual spikes in viewing traffic -- and have a plan for scaling when events are likely to attract broad interest.

The integrated Wowza Server Engine uses what is called unicast streaming. This mechanism sends a separate stream to each viewer from the server, and each user can start and stop the stream from whatever device they are on (desktop, laptop, tablet or smartphone).

The utilization in terms of network bandwidth can be determined by multiplying the number of viewers and the encoding bit rate of the video stream.

**Transcoding** -Live stream transcoding, which is used to enable Adaptive Bit Rate (ABR) streaming, can also create capacity challenges. Transcoding is very CPU intensive, and only a modest number of live streams can be transcoded on a single server at any given time. Wowza has done extensive benchmark testing, and provides results here:

<https://www.wowza.com/forums/content.php?332-Wowza-Transcoder-performance-benchmark>

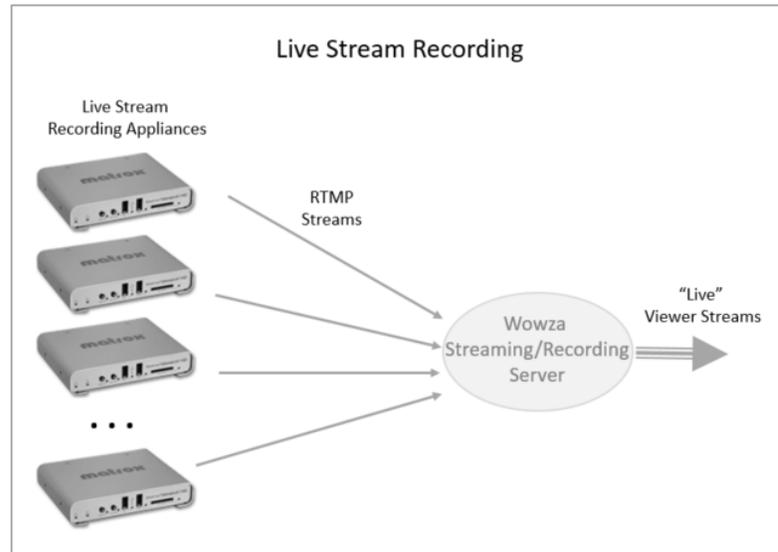


<b>Sporting Event</b>
Stream Bit Rate = 1 Mbps
Number of viewers = 200
Bandwidth required = 200 Mbps

**NOTE: Ensemble Video versions 4.4+ support live stream ABR streaming/transcoding.**

## Live stream recording

With Ensemble Studio live stream recording, there can be many incoming streams being recorded on a single streaming server at any given time. For example there may be recording devices in many classrooms that are all set to record on a fixed class schedule. Typically these streams are not being viewed by many concurrent viewers. The focus is on recording incoming streams for later VOD viewing.



## Typical institution/enterprise experience

For customers using Ensemble Video and Wowza Media Server for internally focused video such as instruction, training, recorded meetings, product announcements and so forth, typical usage will be modest. A single server (e.g., 12 GB RAM, 8 CPUs, and high speed disk storage) is typically adequate to handle the load, which for many institutions and enterprises never exceeds more than a 100 concurrent VOD streams, only a few hundred live streams, and a couple dozen concurrent recordings. Wowza performed tests on Microsoft Azure virtual machine (VM) instances to create a set of deployment examples, here is a summary:

Azure instance type	Virtual Machine CPU/RAM/Disk	Streaming inputs and reach
Azure A1	1 core 1.75 GB memory 70 GB Disk	10 inputs 120 viewers
Azure D2	2 cores 7 GB RAM 100 GB Disk	30 inputs 300 viewers
Azure D3	4 cores 14 GB memory 200 GB Disk	50 inputs 550 viewers

<https://www.wowza.com/uploads/wowza4/wyqgwm/Wowza-Streaming-Engine-Deployment-Examples-MS-Azure-2.pdf>

For Ensemble Video/Wowza Server Engine implementations that will see greater usage, it is crucial to understand how to monitor, test and scale to provide the appropriate server and network infrastructure.

## II. Monitoring

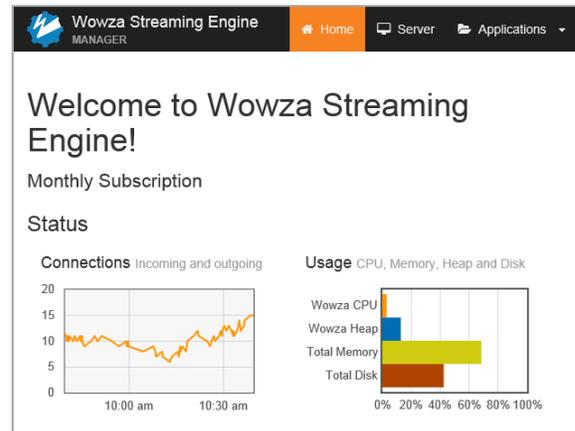
Tools that you can use to monitor and test include tools that come with the streaming media server and operating system, as well as standalone monitoring tools.

### Wowza Streaming Engine

Ensemble Video uses Wowza Server Engine for media streaming, and it offers a variety of useful tools and capabilities for testing, monitoring and scaling streaming capacity.

### Monitoring with Wowza Streaming Engine Manager

The Wowza Streaming Engine Manager is installed when you install Wowza Server Engine, and on Windows servers you can easily start it through the Wowza program group. The Engine Manager is a Web-based application that launches in a browser, and at startup it gives you a snapshot of system usage. You can also easily navigate to server configuration or server monitoring, or to the details of specific applications (e.g., live streaming, live capture, or VOD streaming).



The Server monitoring capability includes interactive graphs for connections, network, memory, disk and CPU usage. This is a great tool for exploring concurrent users and network utilization of your streaming server over whatever time period you want. For example, exploring VOD usage patterns over the past six months, or focusing in on a single live event. In the screen shot at right you can see Maximum concurrent users and Maximum network utilization for an hour long live stream event.

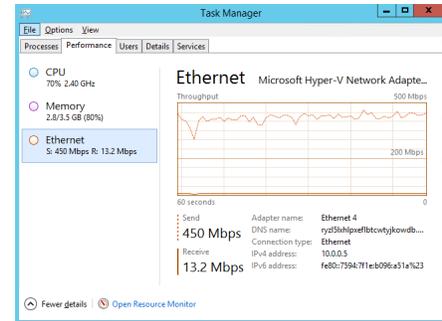


## Other Monitoring Tools

There are several useful tools for monitoring computing, storage and network resources in real time. Many of these tools can also provide historical reports.

### Windows Task Manager

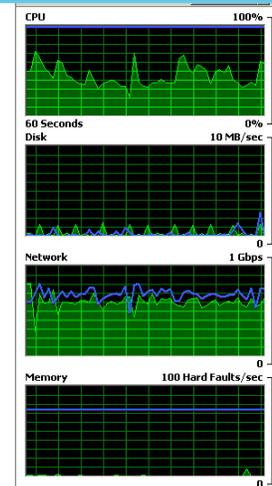
The most commonly used tool is Task Manager, which can show you all running processes, and has a tab for Performance monitoring that shows CPU, Memory and Network utilization.



### Windows Resource Monitor

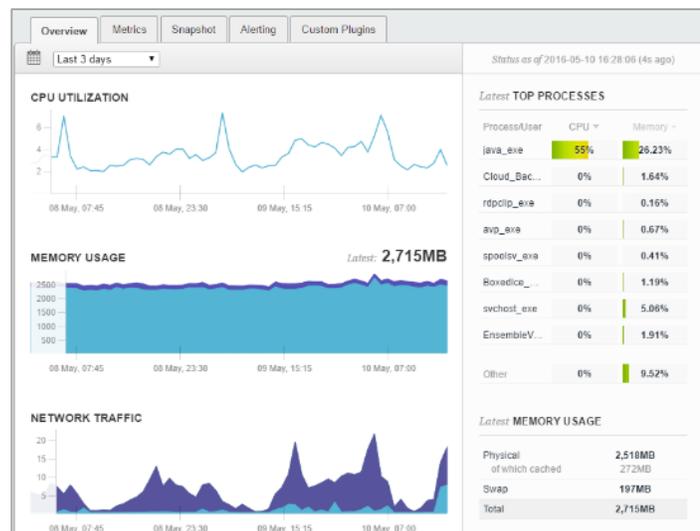
You can launch Resource Monitor by clicking on the Resource Monitor... button on the Task Manager Performance tab. Resource monitor has interactive, real time graphs that show greater detail and can provide more information in one screen.

The screen shot at right shows statistics in real time when testing 700 concurrent live viewers who are all viewing a single source stream.



### Standalone monitoring tools

There are a variety of standalone monitoring tools that are very helpful in monitoring streaming server utilization. For example, the screen shot at right shows a tool called Server Density. It is relatively inexpensive, Web-based monitoring tool that provides real-time info and historical usage reporting. It also enables you to set Alerts on network bandwidth utilization and other key server statistics. This enables you to configure your streaming infrastructure and then configure alerts that will let you know when there are spikes you'll need to investigate and possibly address.



## III. Load Testing



testing VOD streaming. For each you can specify the number of stream to test, and how many unique streams to test:

- **workerCount** tag specifies the number of RTMP viewers to simulate.
- **fileCount** tag specified how many unique streams to view.

The fileCount tag pulls unique streams based on files that use the same filename except for an numerical increment at the end, for example *video1.mp4*, *video2.mp4*, *video3.mp4*, etc. for VOD content, or *event1*, *event2*, *event3*, etc. for live streams. Specifying fileCount of 0 pulls a single stream (e.g., *video.mp4* for VOD or *event* for live stream)

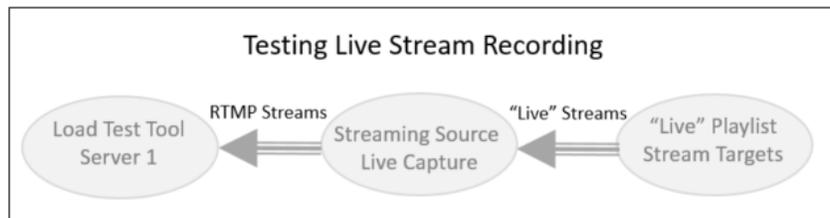
Here are a few sample scenarios:

<b>Live Event</b> Single live stream event All viewers tuned in	<b>Video on Demand</b> Many unique video views Few viewers per video	<b>Internal IPTV streaming</b> Limited live stream channels Many viewers per channel
<pre> &lt;Property&gt;   &lt;Name&gt;workerCount&lt;/Name&gt;   &lt;Value&gt;400&lt;/Value&gt;   &lt;Type&gt;Integer&lt;/Type&gt; &lt;/Property&gt; &lt;Property&gt;   &lt;Name&gt;fileCount&lt;/Name&gt;   &lt;Value&gt;0&lt;/Value&gt;   &lt;Type&gt;Integer&lt;/Type&gt; &lt;/Property&gt;                     </pre>	<pre> &lt;Property&gt;   &lt;Name&gt;workerCount&lt;/Name&gt;   &lt;Value&gt;400&lt;/Value&gt;   &lt;Type&gt;Integer&lt;/Type&gt; &lt;/Property&gt; &lt;Property&gt;   &lt;Name&gt;fileCount&lt;/Name&gt;   &lt;Value&gt;300&lt;/Value&gt;   &lt;Type&gt;Integer&lt;/Type&gt; &lt;/Property&gt;                     </pre>	<pre> &lt;Property&gt;   &lt;Name&gt;workerCount&lt;/Name&gt;   &lt;Value&gt;400&lt;/Value&gt;   &lt;Type&gt;Integer&lt;/Type&gt; &lt;/Property&gt; &lt;Property&gt;   &lt;Name&gt;fileCount&lt;/Name&gt;   &lt;Value&gt;20&lt;/Value&gt;   &lt;Type&gt;Integer&lt;/Type&gt; &lt;/Property&gt;                     </pre>

## Testing Live Stream Recording Performance

One way to test Live Stream Recording performance is to generate a set of live streams that are all streamed to a Live Capture application on target Wowza Streaming Server. To do this you need to first generate the live streams. One way to do this is by setting up a series of encoders, but this can be tedious and time consuming, and require considerable resource. Another option is to create a Playlist of on-demand videos that start up as live streams using the Wowza Stream Publisher capability.

You can set up the playlist on a Wowza server and then send the live streams to your test Streaming Server using the Targets features within the Wowza



Engine Manager Interface. If you send the live streams to a Live Capture application, then ALL streams will be recorded as they arrive at the server, so you then simulate concurrent recording of many live streams. You can also use the Wowza Load Test Tool to simulate a live stream or VOD viewing load to test streaming on a server that is doing significant stream recording.

## Configure live stream playlist

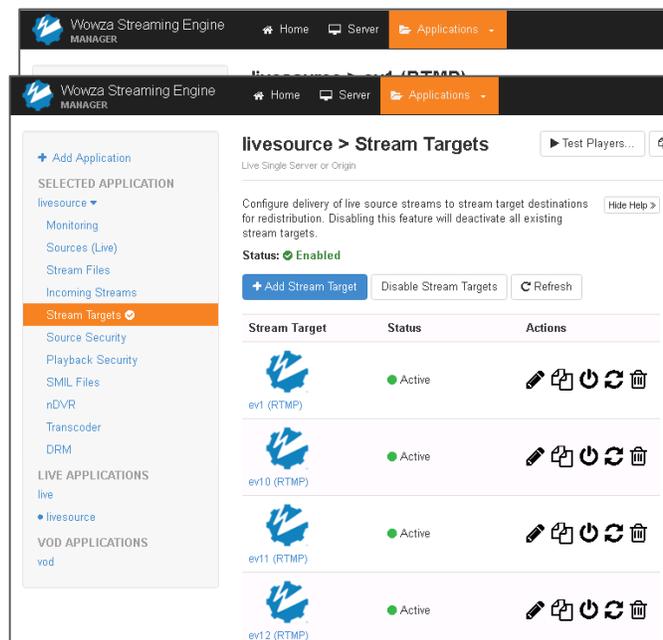
[https://www.wowza.com/forums/content.php?145-How-to-schedule-streaming-with-Wowza-Streaming-Engine-\(StreamPublisher\)](https://www.wowza.com/forums/content.php?145-How-to-schedule-streaming-with-Wowza-Streaming-Engine-(StreamPublisher))

You must configure the module per the instructions via the above link. Once configured, a *schedule.smil* file in the **/content** folder in the Wowza installation determines the live streams that be generated from the playlist, using MP4 content that is also in the **/content** folder. The live application that will provide access to the streams defaults to the standard **/live** application. This is configurable in the Server module properties. The live streams from the playlist will fire as soon as you start the application if you specify a date/time from the past. A sample SMIL file appears in Appendix C. In this example we are using the same *test.mp4* file to generate twenty separate live streams, they will be recorded on the target Wowza server and be available to viewers as:

```
rtmp://<server>/live/ev1
rtmp://<server>/live/ev2
rtmp://<server>/live/ev3
...
rtmp://<server>/live/ev20
```

## Configure stream targets

Once you have your live stream Playlist set up and working, you can use the Stream Targets capability in Wowza Server Engine Admin to send those streams to Live Capture application on your test streaming server. Just navigate to the live application you want to send your live streams from, and go to the Stream Targets menu. There you can add and edit as many stream targets as you want. In this case the *livesource* application is where the Playlist live streams are accessible, and in the Stream Targets setup we targeted those streams to another server where there is a *liverecord* application that has been configured.



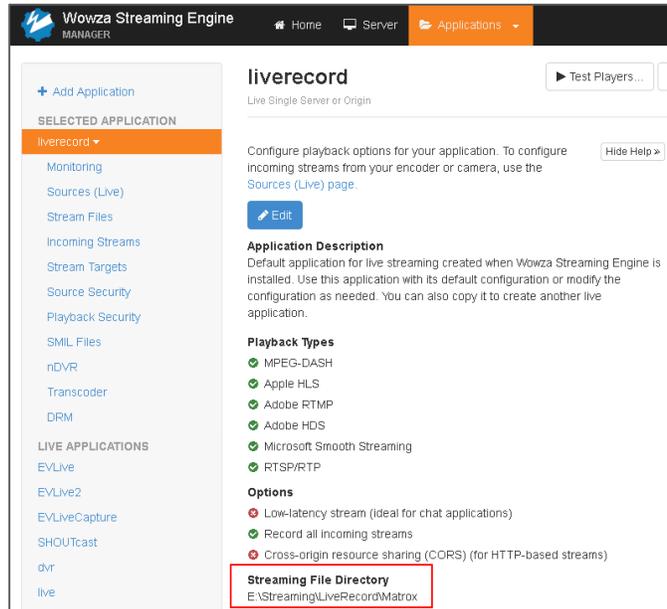
The screenshot shows the Wowza Streaming Engine Manager interface. The left sidebar lists various application settings, with 'Stream Targets' selected under the 'livesource' application. The main panel displays the 'Stream Targets' configuration for 'livesource > Stream Targets'. The status is 'Enabled'. Below this, there is a table of stream targets:

Stream Target	Status	Actions
ev1 (RTMP)	Active	[Edit] [Copy] [Refresh] [Delete]
ev10 (RTMP)	Active	[Edit] [Copy] [Refresh] [Delete]
ev11 (RTMP)	Active	[Edit] [Copy] [Refresh] [Delete]
ev12 (RTMP)	Active	[Edit] [Copy] [Refresh] [Delete]

Once you define your Stream Targets, you just Enable Stream targets and restart the application and the streams defined in the **schedulestream.smil** Playlist will fire up and stream to the destination server's *liverecord* streaming application.

Once you've got this configured and running, you will see all of your source live streams as incoming streams on the target server (*liverecord* application in our example) and you will also see the recording in real time in the recording directory specified on the target Wowza streaming server.

You can test performance further by simulating streaming load with Wowza Load Test Tool so you can gauge how your system performs while recording simultaneous streams.



## IV. Scaling Capacity

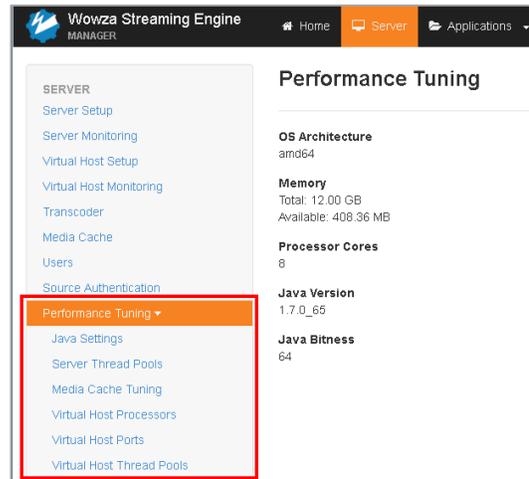
### Infrastructure Adjustments

As you monitor server resources you may find situations where CPU, memory or disk activity becomes a bottleneck. A few steps can be taken to address or prevent this from happening.

### Wowza performance tuning

Make sure you are using production level Java settings and follow the performance tuning instructions from Wowza in this article:

<https://www.wowza.com/forums/content.php?46-How-to-do-performance-tuning>



### Single purpose servers

In some streaming deployments, Wowza is placed on a server that has other software on it that consumes valuable resources. This is often the case at customer implementations where there are very low expectations for streaming traffic, and Ensemble Video software, IIS, and MS-SQL Express are installed on a single VM together with Wowza Streaming Engine. If this is the case and you have any performance concerns, an obvious first step is to break the software and services onto separate servers or virtual machines.

### Network capacity

A common bottleneck is network capacity, especially for live stream events where many concurrent viewers are invited to simultaneously connect to view a stream. If you can allocate more network capacity, that can prevent bottlenecks that could otherwise occur. It is very often the case that a server with even modest hardware can deliver a lot of content concurrently, but limited network connectivity (e.g., institutional/enterprise WAN, Internet connectivity, or LAN routers, switches) causes a bottleneck.

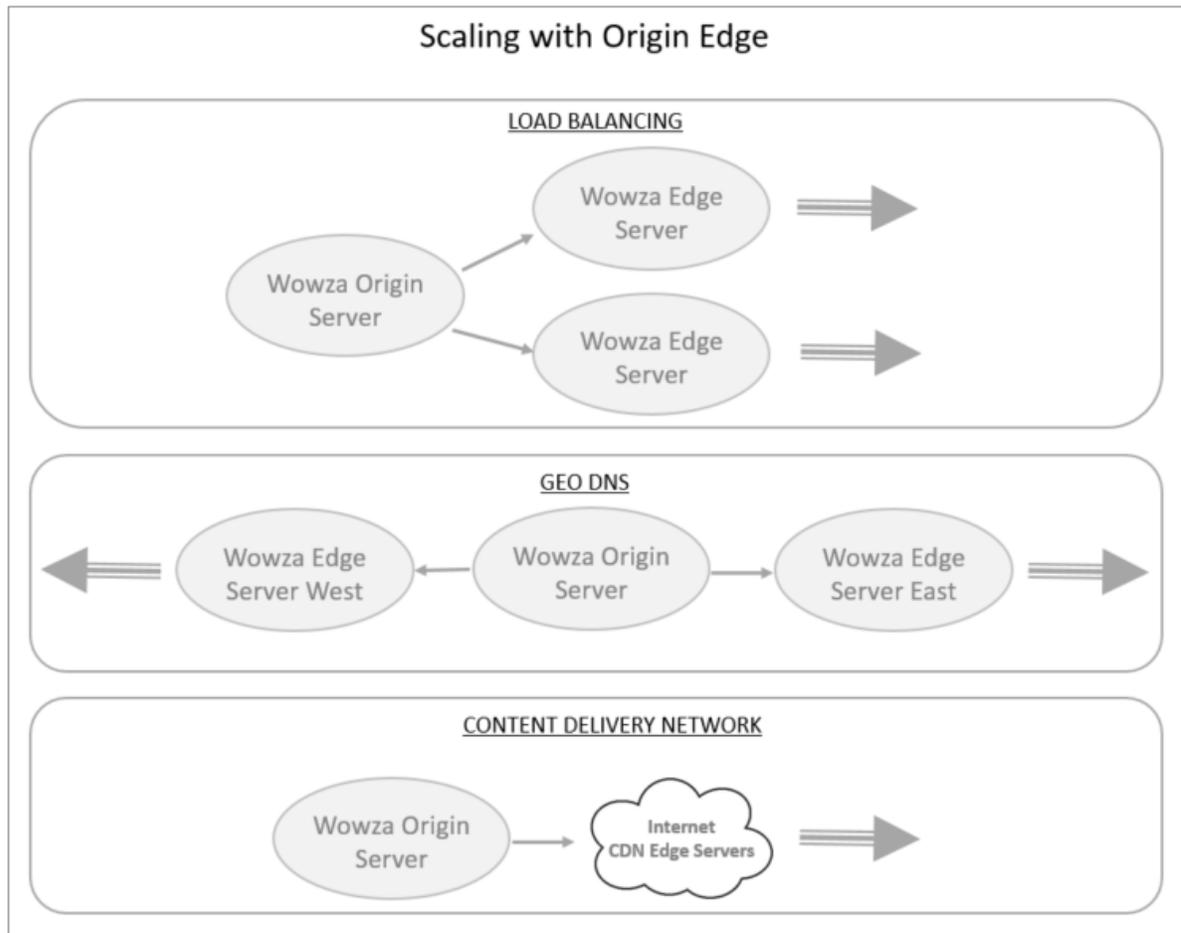
### Disk technology

If your implementation does a lot of live stream recording, you may get a performance boost by using properly configured RAID disk (RAID 5 or better) or Solid State Disks (SSD) storage, rather than standalone hard disks.

## Architectural Extensions

### Scaling and load balancing

If live stream viewing and recording is expected to tax a single streaming server to the point that it will not keep up with demand, you will need to implement some form of streaming server scaling/load balancing.



Wowza Streaming Engine supports Origin/Edge streaming architectures for both video-on-demand and live streaming. You can also use Origin/Edge with a Content Delivery Network.

Scaling can be done with a variety of Wowza specific tools, or with third party technologies. Here is a quick summary of available options, be sure to check out these useful documents from Wowza, starting with this Scaling and Load Balancing overview:

<https://www.wowza.com/forums/content.php?666-scaling-and-load-balancing>

**Wowza Live Stream Repeater** - This is an origin/edge configuration where a single Wowza server can ingest live stream and then multiple Wowza edge servers can stream the content to end users, thereby splitting up the load. Edge servers can be configured in different geographical locations to provide efficient regional or remote office streaming, or multiple edge servers can be configured in a single data center to spread the load out to avoid server bottlenecks.

<https://www.wowza.com/forums/content.php?668-Scaling-with-Wowza-live-stream-repeater>

**Wowza MediaCache** – this is also an origin/edge configuration that works well for video-on-demand content. MediaCache servers can be configured in different geographic regions to provide efficient and reliable regional or remote office streaming, or multiple MediaCache servers can be configured in a single data center to spread out the load and avoid server bottlenecks.

<https://www.wowza.com/forums/content.php?669-Scaling-with-Wowza-Media-Cache>

**Load balancing** - You can use the Wowza-Dynamic Load Balancing AddOn, you can load balance with round-robin DNS or geo-DNS tools, or you can use third party load balancers such as an F5 Big-IP. Load-balancing a Wowza streaming server takes a degree of care and attention to detail to get it right and avoid possible streaming issues. See details in the Wowza load balancing documentation.

<https://www.wowza.com/forums/content.php?671-Load-balancing-overview>

**Content Delivery Network** – a CDN can distribute your VOD or live streams to a global audience. This is a fee-based service where Ensemble Video will provide a CDN streaming URL that you can add to any of your Ensemble Video live streaming workflows (Live Stream, Live Capture or Ensemble Studio).

Ensemble Video currently offers this CDN option for live streaming using the Mirror Image CDN. We are actively exploring some new alternatives for live and on-demand CDN content distribution, so watch our Newsletter for news about new CDN options.

# Appendix A – Video On Demand Typical Usage

## Multi-Tenant

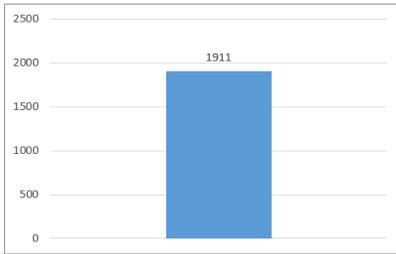
14 Sep 2015 - 18 Sep 2015

Video Titles: 42,697  
Windows 2008 R2 64 bit

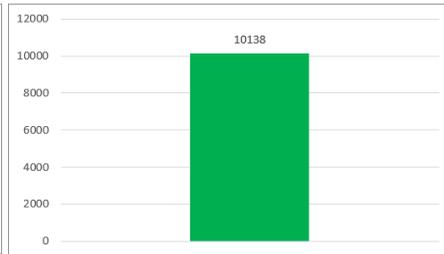
Content Owners: 1,892  
Intel Xeon 2.3 GHz -3 Processors/6 cores

Storage Used: 4.8 TB  
8 GB RAM

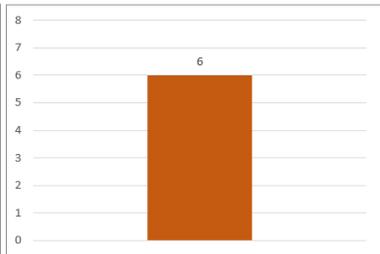
Items viewed



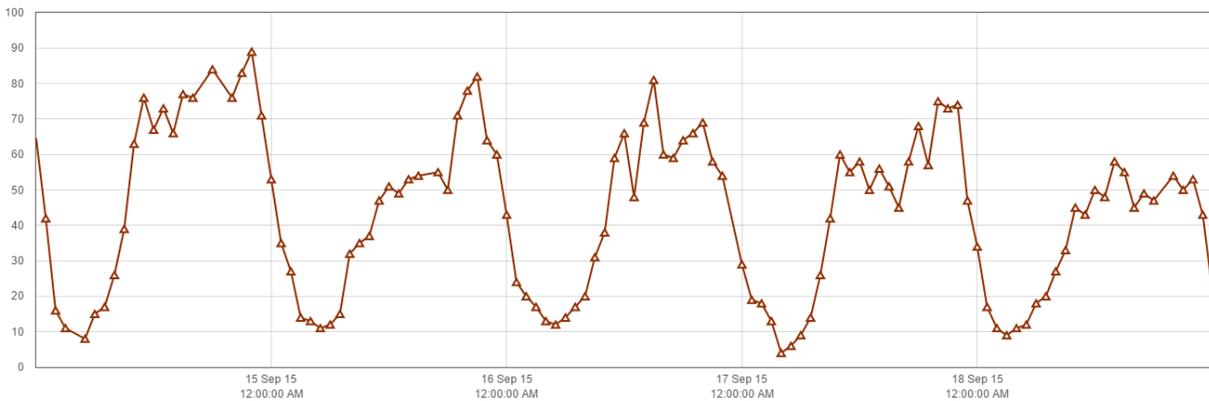
Total views



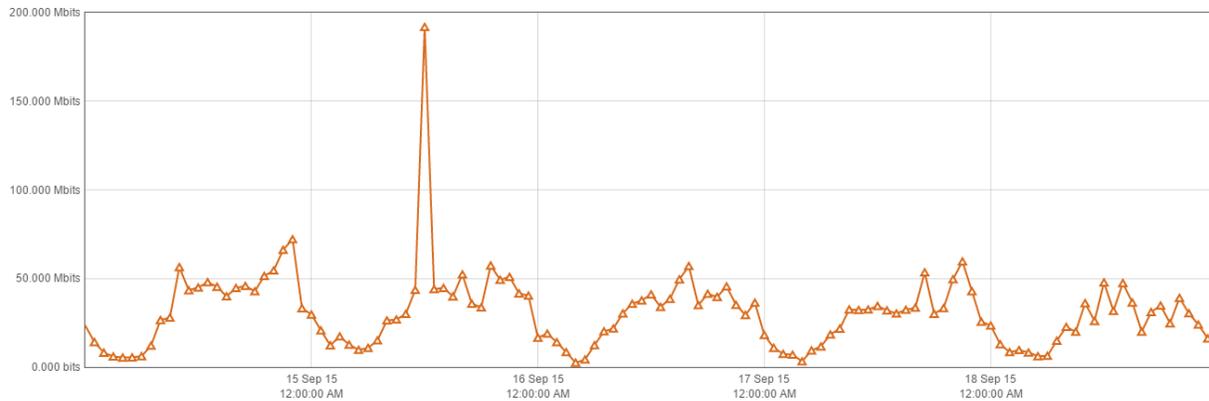
Average views



Max viewer connections



Max bandwidth – outgoing



### Medium Self-Hosted

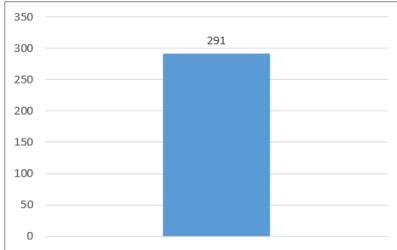
14 Sep 2015 - 18 Sep 2015

Video Titles: 30,156  
Windows 2008 R2 64 bit

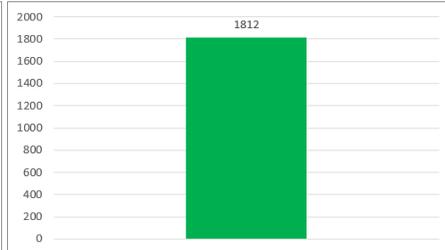
Content Owners: 807  
Intel Xeon 2.7 GHz single Processor/8 cores

Storage Used: 5.9 TB  
12 GB RAM

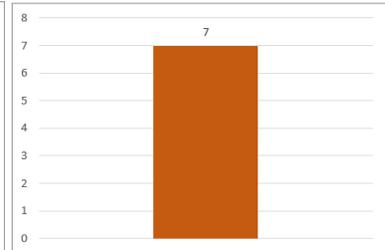
Items viewed



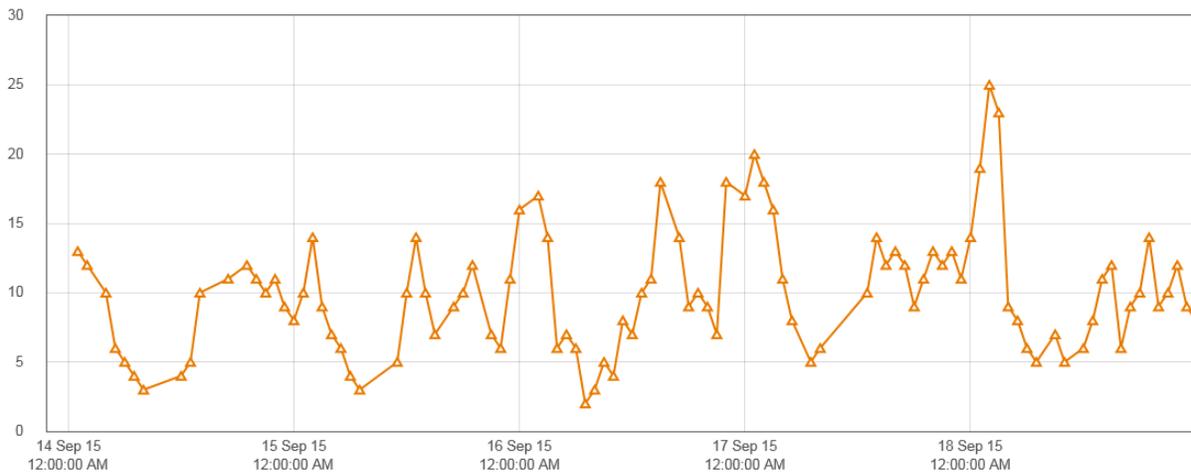
Total views



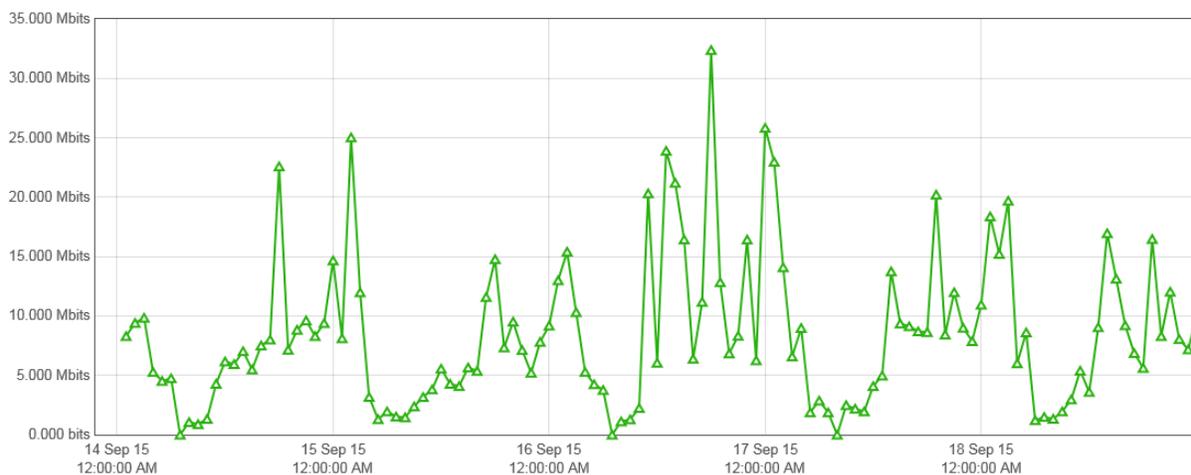
Average views



Max viewer connections



Max bandwidth – outgoing



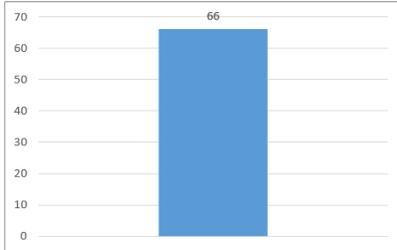
## Small Self-Hosted 14 Sep 2015 - 18 Sep 2015

Video Titles: 2,207  
Windows 2008 R2 64 bit

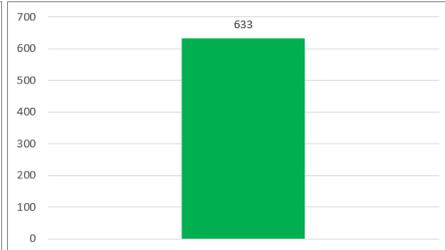
Content Owners: 94  
Intel Xeon 2.3 GHz -2 Processors/4 cores

Storage Used: 358 GB  
4 GB RAM

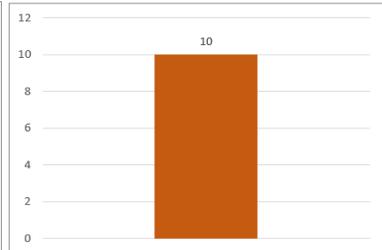
Items viewed



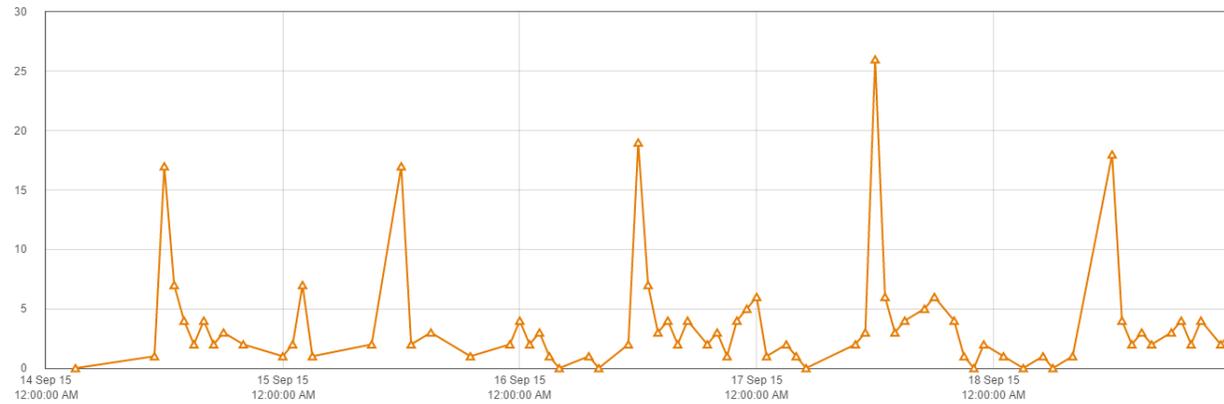
Total views



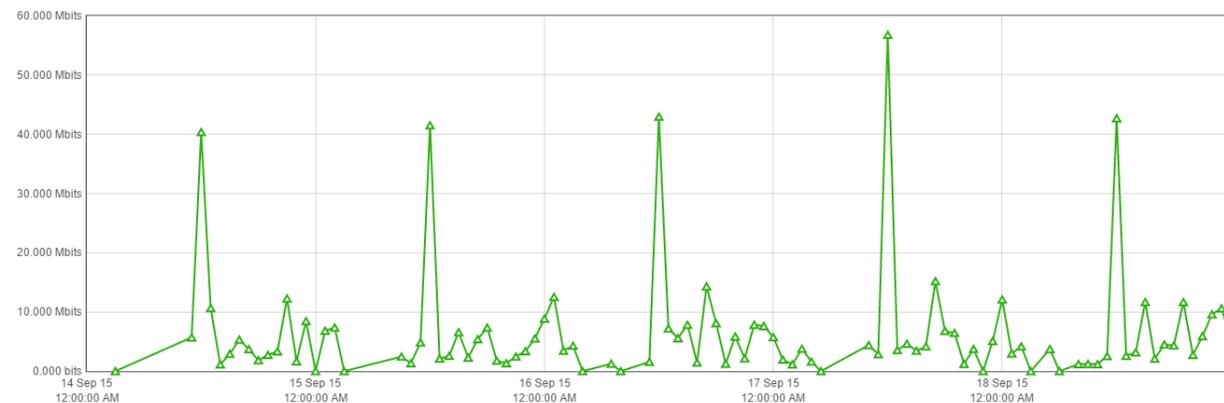
Average views



Max viewer connections



Max bandwidth - outgoing



## Appendix B – Sample Load Test Tool XML file

```
<Root>
  <Tests>
    <Test>
      <Name>live</Name>
      <ClassBase>com.wowza.wms.performance.remote.PerformanceRemoteClientTest</ClassBase>
      <Properties>
        <Property>
          <Name>workerCount</Name>
          <Value>400</Value>
          <Type>Integer</Type>
        </Property>
        <Property>
          <Name>fileCount</Name>
          <Value>0</Value>
          <Type>Integer</Type>
        </Property>
        <Property>
          <Name>streamName</Name>
          <Value>event</Value>
        </Property>
        <Property>
          <Name>streamExt</Name>
          <Value></Value>
        </Property>
        <Property>
          <Name>vhostName</Name>
          <Value>_defaultVHost_</Value>
        </Property>
        <Property>
          <Name>connectionString</Name>
          <Value>live/_definst_</Value>
        </Property>
        <Property>
          <Name>bufferTime</Name>
          <Value>3000</Value>
          <Type>Integer</Type>
        </Property>
        <Property>
          <Name>doRandom</Name>
          <Value>>false</Value>
          <Type>Boolean</Type>
        </Property>
        <Property>
          <Name>doRepeat</Name>
          <Value>>true</Value>
          <Type>Boolean</Type>
        </Property>
        <Property>
          <Name>ipAddress</Name>
          <Value>192.168.1.105</Value>
        </Property>
        <Property>
          <Name>ports</Name>
          <Value>1935</Value>
        </Property>
      </Properties>
    </Test>
    <Test>
      <Name>vod</Name>
      <ClassBase>com.wowza.wms.performance.remote.PerformanceRemoteClientTest</ClassBase>
      <Properties>
        <Property>
          <Name>workerCount</Name>
          <Value>200</Value>
          <Type>Integer</Type>
        </Property>
      </Properties>
    </Test>
  </Tests>
</Root>
```

```

</Property>
<Property>
  <Name>fileCount</Name>
  <Value>200</Value>
  <Type>Integer</Type>
</Property>
<Property>
  <Name>streamName</Name>
  <Value>sample</Value>
</Property>
<Property>
  <Name>streamExt</Name>
  <Value>.mp4</Value>
</Property>
<Property>
  <Name>vhostName</Name>
  <Value>_defaultVHost_</Value>
</Property>
<Property>
  <Name>connectionString</Name>
  <Value>vod/_definst_</Value>
</Property>
<Property>
  <Name>bufferTime</Name>
  <Value>3000</Value>
  <Type>Integer</Type>
</Property>
<Property>
  <Name>doRandom</Name>
  <Value>>false</Value>
  <Type>Boolean</Type>
</Property>
<Property>
  <Name>doRepeat</Name>
  <Value>>true</Value>
  <Type>Boolean</Type>
</Property>
<Property>
  <Name>ipAddress</Name>
  <Value>192.168.1.105</Value>
</Property>
<Property>
  <Name>ports</Name>
  <Value>1935</Value>
</Property>
</Properties>
</Test>
</Tests>
</Root>

```

## Appendix C – Sample “Live” Stream Playlist SMIL file

```
<smil>
<head>
</head>
<body>

  <stream name="ev1"></stream>
  <stream name="ev2"></stream>
  <stream name="ev3"></stream>
  <stream name="ev4"></stream>
  <stream name="ev4"></stream>
  <stream name="ev5"></stream>
  <stream name="ev6"></stream>
  <stream name="ev7"></stream>
  <stream name="ev8"></stream>
  <stream name="ev9"></stream>
  <stream name="ev10"></stream>
  <stream name="ev11"></stream>
  <stream name="ev12"></stream>
  <stream name="ev13"></stream>
  <stream name="ev14"></stream>
  <stream name="ev15"></stream>
  <stream name="ev16"></stream>
  <stream name="ev17"></stream>
  <stream name="ev18"></stream>
  <stream name="ev19"></stream>
  <stream name="ev20"></stream>

  <playlist name="pl1" playOnStream="ev1" repeat="true" scheduled="2013-09-25 16:00:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>

  <playlist name="pl2" playOnStream="ev2" repeat="true" scheduled="2013-09-25 16:30:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>

  <playlist name="pl3" playOnStream="ev3" repeat="true" scheduled="2013-09-25 16:30:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>

  <playlist name="pl4" playOnStream="ev4" repeat="true" scheduled="2013-09-25 16:00:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>

  <playlist name="pl5" playOnStream="ev5" repeat="true" scheduled="2013-09-25 16:30:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>

  <playlist name="pl6" playOnStream="ev6" repeat="true" scheduled="2013-09-25 16:30:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>

  <playlist name="pl7" playOnStream="ev7" repeat="true" scheduled="2013-09-25 16:00:00">
    <video src="mp4:test.mp4" start="0" length="-1"/>
  </playlist>


```

```
<playlist name="pl8" playOnStream="ev8" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl9" playOnStream="ev9" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl10" playOnStream="ev10" repeat="true" scheduled="2013-09-25 16:00:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl11" playOnStream="ev11" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl12" playOnStream="ev12" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl13" playOnStream="ev13" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl14" playOnStream="ev14" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl15" playOnStream="ev15" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl16" playOnStream="ev16" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl17" playOnStream="ev17" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl18" playOnStream="ev18" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl19" playOnStream="ev19" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

<playlist name="pl20" playOnStream="ev20" repeat="true" scheduled="2013-09-25 16:30:00">
  <video src="mp4:test.mp4" start="0" length="-1"/>
</playlist>

</body>
</smil>
```