Beamr Video User Manual

Technology Overview

Beamr Video is a patent-pending perceptual video optimizer, which reduces the bitrate of video streams without affecting their perceptual quality. Beamr Video uses standard H.264 video encoders, resulting in files that are fully compatible with any media player or consumer device that supports the H.264 video format.

Beamr Video is capable of reducing the bitrate of H.264 video streams by up to 75% (4X), while preserving their perceptual visual quality. The Beamr Video algorithm imitates the perceptual qualities of the human visual system, ensuring that the video stream is compressed to the maximum extent possible by removing redundancies, without creating any visual artifacts in the process. This enables fully automatic, maximal compression of video streams with no human intervention required.

Supported Formats

Input Formats

- Containers: MP4, MOV, H264, 264, AVC, BSF
- Video Codecs: H.264
- Audio Codecs: Any (pass-through)

Output Formats

- Containers: MP4, MOV, H264
- Video Codecs: H.264
- Audio Codecs: Any (pass-through)

Note that audio tracks and any other non-video tracks in the input file are copied to the output file. Also note that the following content types are not supported in the current version of Beamr Video:

- Files with multiple video tracks
- Interlaced video content
- MPEG-DASH or fragmented mp4 files
- MP4 files with multiple mdat atoms

System Requirements

Beamr Video is supported on the following 64-bit operating systems:

- Ubuntu 12.04 (or higher) and compatible
- RHEL 6.x (or higher) and compatible

A minimum of 3 GB of RAM per instance of Beamr Video is recommended for processing full HD streams. Note that Beamr Video launches multiple instances in parallel to improve performance. The number of instances depends on the number of licenses, number of cores and amount of memory on your server. For details, see the "Multi-core Processing" section below.

Installation Instructions

Installation Script

The simplest way to install Beamr Video is through the provided installation script.

To install Beamr Video on RHEL-compatible systems, use the following procedure:

1. Download the installation script (e.g. beamrvideo_installer_2.3.0.0_rpm.sh) to your server.
2. Add execution permissions to the installation script:

   ```
   chmod a+x beamrvideo_installer_2.3.0.0_rpm.sh
   ```

3. Run the script with root permissions:

   ```
   sudo ./beamrvideo_installer_2.3.0.0_rpm.sh
   ```

To install Beamr Video on Ubuntu-compatible systems, use the following procedure:

1. Download the installation script (e.g. beamrvideo_installer_2.3.0.0_deb.sh) to your server.
2. Add execution permissions to the installation script:

   ```
   chmod a+x beamrvideo_installer_2.3.0.0_deb.sh
   ```

3. Run the script with root permissions:

   ```
   sudo ./beamrvideo_installer_2.3.0.0_deb.sh
   ```

Manual Installation

If you have any issues with the installation script, you can also install the Beamr Video packages manually.

Beamr Video includes 5 installation packages:

1. The Intel IPP redistributable libraries, installed to `/opt/intel_redist`.
2. A pre-compiled package which includes Python-2.7 and virtualenv (for systems without Python 2.7)
3. The Beamr Video software, which includes the Beamr Video optimizer, the H.264/AVC reference encoder, scripts, documentation, and a sample video file. Beamr Video is installed to `/opt/beamrvideo`. A beamrvideo soft-link is created in `/usr/bin`.
4. The Beamr Video Multi Core Add-on, which includes components that enable dividing an input file into multiple segments, and processing them in parallel using multiple instances of Beamr Video running on multiple cores.
5. The Beamr Video Dashboard Add-on, which provides a web-based interface for monitoring and controlling Beamr Video.

The binaries are provided in .deb or .rpm packages. The installation requires root permissions, and some packages require Internet connectivity since they download additional components during installation.

Use the following installation procedure for RPM packages:

1. Install the Intel redistributable libraries:

   ```
   sudo rpm -i beamrvideo-intel-redist-2013.sp1.2.144-1.x86_64.rpm
   ```

2. If your system does not provide Python2.7, install the Beamr Video Python package:

   ```
   sudo rpm –i beamrvideo_python-2.7.5-1.x86_64.rpm
   ```

3. Install the Beamr Video package

   ```
   sudo rpm –i beamrvideo-2.3.0.0.x86_64.rpm
   ```

4. Coinfigure the Beamr Video license using your license-id:

   ```
   bv_activation -a activate --license-id your-license-id --instances
   xxx
   ```

   For details on the Beamr Video licensing mechanism, see the "Licensing" section

below.

5. Install the Beamr Video Multi Core Add-on package

```
sudo rpm –i beamrvideo_ms_addon-2.3.0.0-1.x86_64.rpm
```

6. Install the Beamr Video Dashboard Add-on package

```
sudo rpm –i beamrvideo_dashboard_addon-2.3.0.0-1.x86_64.rpm
```

Use the following installation procedure for DEB packages:

1. Install the Intel redistributable libraries:

```
sudo dpkg -i beamrvideo-intel-redist_2013.sp1.2.144-
ubuntu0.1_amd64.deb
```

2. Install the Beamr Video package

```
sudo dpkg -i beamrvideo_2.3.0.0-ubuntu0.1_amd64.deb
```

3. Coinfigure the Beamr Video license using your license-id:

```
bv_activation -a activate --license-id your-license-id --instances
xxx
```

   For details on the Beamr Video licensing mechanism, see the "Licensing" section below.

4. Install the Beamr Video Multi Core Add-on package

```
sudo dpkg -i beamrvideo-ms-addon_2.3.0.0-ubuntu0.1_amd64.deb
```

5. Install the Beamr Video Dashboardi Core Add-on package

```
sudo dpkg -i beamrvideo-dashboard-addon_2.3.0.0-ubuntu0.1_amd64.deb
```

Uninstalling Packages

The simplest way to uninstall Beamr Video is using the --uninstall parameter with the

provided installation script.

For RPM Packages: sudo ./beamrvideo_installer_2.3.0.0_rpm.sh --uninstall

For DEB Packages: sudo ./beamrvideo_installer_2.3.0.0_deb.sh --uninstall

If you have any issues with uninstalling using the installation scripts, you can also uninstall the Beamr Video packages manually. Note that packages should be uninstalled in reverse order to the order in which they were installed. If you used an installation script to install the packages, refer to the section "Manual Installation" to view the installation order, and un-install in reverse order.

Use the following procedure to uninstall RPM packages:

1. Get the list of installed packages:

```
sudo rpm -qa | grep beamrvideo
```

2. Perform the following command for each installed package:

```
sudo rpm -e packagename
```

Use the following procedure to uninstall DEB packages:

1. Get the list of installed packages:

```
sudo dpkg -l | grep beamr
```

2. Perform the following command for each installed package:

```
sudo dpkg --purge package-name
```

Plug-in Installation

Note that other video encoders can be plugged into the Beamr Video environment instead of the reference encoder, and should be installed separately. For details, please contact support@beamrvideo.com.

Operating Instructions

Beamr Video is activated using the command `beamrvideo` which accepts a video file as input, and outputs an optimized video file with the postfix "_mini". When processing is finished, `beamrvideo` reports the sizes of the input and output files, and the reduction factor achieved. The syntax of the `beamrvideo` command is as follows:

```
beamrvideo [options] --input input-file
```

Note that Beamr Video launches multiple instances in parallel to improve processing performance. For details, see the section on "Multi-core Processing" below. To run only a single instance of the Beamr Video optimizer, use the command `beamrvideo_si` instead of `beamrvideo`, which accepts the same parameters.

Typical Usage

Typical usage includes specifying the input file name and output folder. You can try the following command on the file sample.mov provided in `/opt/beamrvideo/samples`:

```
beamrvideo --input /opt/beamrvideo/samples/sample.mov --output /tmp
```

The output in this case will be the file `sample_mini.mov` located in `/tmp`.

Note that `beamrvideo` runs as service under the *beamrvideo* user. The input files must have read permission for the user *beamrvideo*, and the output folder must have write permission for the user *beamrvideo*.

Command Line Options

Beamr Video supports the following command line options. Note that the support for some command line options depends on the plug-in encoders that have been installed.

`-h`, `--help`: Displays usage instructions. Any other flags used in combination with `-h` or `--help` will be ignored.

`-v`, `--version`: Displays the software version and build number. Any other flags used in combination with `-v` or `--version` will be ignored.

`-i`, `--input`: Specifies the location of the input file to process

`-o`, `--output`: Specifies the folder for the output file and log files [By default, the folder of the input file is used].

`--overwrite`: Overwrites the output file and log files (if they exist).

`--optimize bitrate`: Optimizes the processing to generate a video stream with the lowest possible bitrate, at the expense of high CPU consumption.

`--optimize balanced`: Optimizes the processing for a balance of low bitrate and high performance. Performance will be better than when using the `--optimize bitrate` setting, and bitrate will be lower than when using the `--optimize speed` setting [default].

`--optimize speed`: Optimizes the processing for highest performance (lowest processing time).

Note: The `--optimize` flag does not affect the quality of the output video. It only controls

the tradeoff between CPU consumption and output bitrate.

`--quality best`: Provides the best output video quality possible. In this mode, the output video stream will be perceptually identical to the input video stream, even when viewed by an expert viewer.

`--quality high`: Produces an output video stream with a lower bitrate than `--quality best`, which is perceptually identical to the input video stream when viewed by an average user [default].

`--preserve-bluray`: Preserves the Blu-ray compatibility of the input video stream. If the input video stream is compatible with the Blu-ray format, this flag ensures that the output video stream will also be Blu-ray compatible. Note that Blu-ray compatibility is preserved only for input files that are in elementary stream formats (H264, 264, AVC, BSF).

Printouts

During its execution, Beamr Video prints out the following information:

- Application version
- Processing progress
- Name and size of input video file
- Name and size of output video file
- File recompression ratio: The ratio between the input and output size of the whole file (including audio), and the difference between them in percent.
- Video stream recompression ratio: The ratio between the input and output size of the video stream only, and the difference between them in percent.
- Total encoding CPU time: The total time spent by all CPUs on the server in processing the file
- Total wall time: The total duration of the file processing

Below is an example of a typical printout for a file processed by Beamr Video.

```
Beamr Video version 2.3.0.0 Copyright (C) 2010-2015 ICVT Ltd.
==========================================
Starting Beamr Video
==========================================
Analyzing input file /home/beamr/videos/Hanna.mp4
Starting to optimize video
Progress: 100.00%
Analyzing output
Input file: /home/beamr/videos/Hanna.mp4  size: 156.26 MB
Output file: /home/beamr/videos/Hanna_mini.mp4  size: 31.77 MB
File recompression ratio: 4.91X (79.67% savings)
Video stream recompression ratio: 5.26X (80.99% savings)
Total encoding CPU time: 3m 10s
Total wall time: 3m 4s
==========================================
Finishing Beamr Video 2.3.0.0
```

Dashboard

The Beamr Video Dashboard provides a web-based interface for creating and monitoring optimization jobs. To access the Dashboard, point your web browser to http://your-server-ip-address:8888/

Note that files optimized using the `beamrvideo_si` command do not appear in the Dashboard interface.

Usage Notes

1. The optimization ratios achieved depend on the video resolution, quality and content, and therefore are different for each file. However, the perceptual quality of the resulting video file will always be the same as the original.
2. Beamr Video generates the highest optimization ratios for high-quality video files, such as those captured by video cameras or originating from BluRay discs. Using Beamr Video on files that have been edited or compressed may cause lower optimization ratios.
3. If Beamr Video detects that the compression level of the input video stream is very high, and it cannot be further recompressed without degrading visual quality, the input file will be copied to the output.

Multi-core Processing

Overview

To shorten the processing time, the input file is processed using multiple instances of Beamr Video that run on multiple cores in parallel. When you execute the regular `beamrvideo` command, the input file is divided into multiple segments, which are processed in parallel on

different cores, and finally "stitched" to create the output file.

Note that if you run `beamrvideo`, and then try to run another instance of `beamrvideo` from a different console, you might see the message `"Pending…".` This message means that another file is already being processed on all available (and licensed) cores by `beamrvideo`, so the current file will remain pending until the processing of the previous file is finished.

If you try to stop the execution of `beamrvideo` using `^C` for example, only the progress monitoring printouts will stop, but Beamr Video processing will continue in the background.

Components of Multi-core Processing

Below is a description of the major components used in the Beamr Video multi-core processing.

Server: The server is a daemon which manages the processing workflow. It manages the processing of multiple jobs, and dispatches tasks to workers.

Job: A job is the processing of a single input file.

Segment: When processing a job, the input file is split into segments, and each segment is processed by one of the available workers. The number of segments is set according to the file size.

Worker: A worker is a process that performs Beamr Video processing. In general it is a single instance of Beamr Video. The number of workers is set by default according to the number of cores and the available memory, but can also be set manually using the `beamrvideo_mgr workers` command.

Task: A task is processing that is performed on a single worker. A task can be dividing an input file into segments, processing a single segment, or stitcing the segments into the output file.

beamrvideo_mgr: Basic Usage

The `beamrvideo_mgr` command enables you to control the multi-core processing of Beamr Video on a deeper level than `beamrvideo`. This command enables you to directly control the server, workers, jobs and other multi-core processing components.

Before you process a file with `beamrvideo_mgr`, verify that the server is running using the following command:

```
beamrvideo_mgr server status
```

To process a file, use the `beamrvideo_mgr jobs create` command. Below is an example of using `beamrvideo_mgr jobs create` to process the sample.mov file:

```
beamrvideo_mgr jobs create --monitor –output /tmp --input
/opt/beamrvideo/samples/sample.mov
```

Note that the `beamrvideo` command is actually an alias for `beamrvideo_mgr jobs create --monitor`.

Below is a description of the command line options for the `beamrvideo_mgr` command.

`-h`, `--help`: Displays usage instructions. Any other flags used in combination with `--help` will be ignored.

`server`: Commands for controlling the server.

`server start`: Start the server. Note that multi-core processing of Beamr Video can only happen when the server is running.

`server stop`: Stop the server.

`server restart`: Stop and re-start the server.

`server reset`: Stop the server, remove all entries in the server database, and re-start the server.

`server status`: Query the status of the server.

`jobs`: Commands for controlling jobs.

`jobs create`: Create a new job for processing an input file. The input file and all other `beamrvideo` command line options are specified in the same way as for the regular `beamrvideo` command (see the "Command Line Options"section above). In addition, the `beamrvideo_mgr jobs create` command supports the following command line options:

`--monitor`: Display the progress of the created job on the console.

`--cleanup`: Determines which temporary files to keep or remove after processing. This command accepts the following arguments:

`keep-all`: All temporary files are kept.

`keep-logs`: Only log files are kept, and other temporary files are deleted [default].

`remove-all`: All temporary files are deleted, including logs.

`jobs status`: Query the status of a certain job. The following command line options are supported:

`--job-id JOB_ID`: Specify the id of the job for which the status is queried. Note that the job id is displayed on the console following a `beamrvideo_mgr jobs create` command.

`count`: Displays the number of pending segments for the specified job-id.

`jobs cancel`: Cancel a certain job or all jobs. The following command line options are supported:

`--job-id JOB_ID`: Specify the id of the job to cancel. Note that the job id is displayed on the console following a `beamrvideo_mgr jobs create` command.

`--all`: Cancel all jobs.

beamrvideo_mgr: Advanced Usage

This section describes the more advanced command line options of the `beamrvideo_mgr` command.

`workers`: Commands for controlling workers.

`workers status`: Query the status of all active workers.

`workers add`: Add one worker.

`workers remove`: Remove one worker..

`license`: Command line options for controlling the Beamr Video license.

`license status`: Query the status of Beamr Video licenses.

`license activate`: Activate licenses on the current node. The following command line options are supported:

`--instances INSTANCE_COUNT`: The number of instances to activate on the current node.

Note that more advanced licensing functionality is available via the use of the `bv_activation` tool. For details on Beamr Video licensing and the usage of the `bv_activation` tool, see the "Licensing"section below.

Licensing

Beamr Video uses a cloud-based licensing system which controls the number of instances that can run simultaneously on a single server (node) or on several nodes. When you run Beamr Video for the first time, you will be asked to enter the License ID you received from Beamr, and the number of licenses you would like to activate on the current node. The number of licenses determines the maximum number of concurrent instances of Beamr Video that you will be able to run on the current node. Note that in order to process an input file on multiple cores in parallel using the `beamrvideo` or `beamrvideo_mgr` commands you will need a license for each concurrent processing core.

You can use the `bv_activation` licensing tool to view and change the allocation of licenses between your nodes. Using this tool, you can activate and deactivate licenses on the current node, and on other nodes on which you have installed Beamr Video.

Licensing Tool Usage Examples

To view the allocation of licenses between nodes for your License ID, run `bv_actvation` without any parameters:

```
bv_activation
```

The licensing tool will display your current License ID and Node ID, and a structure in json format that has a list of nodes, and the number of active licenses on each node.

To activate 3 licenses on the current node, run `bv_activation` with the following parameters:

```
bv_activation --action activate --instances 3
```

To remove all the licenses for the current node, run `bv_activation` with the following parameters:

```
bv_activation --action deactivate
```

If you encounter any issues with Beamr Video licensing, you can delete the license file located at `/opt/beamrvideo/system/run/beamrvideo.lic`. After you delete this file, you will be asked to enter your License ID again the next time you run Beamr Video, and select the number of licenses you would like to activate on the current node.

Licensing Tool Options

The licensing tool supports the following command line options.

`--help` or `-h`: Displays usage instructions. Any other flags used in combination with this flag will be ignored.

`--action info`: Displays information about active Beamr Video licenses.

`--action activate`: Activates Beamr Video licenses.

`--action deactivate`: Deactivates Beamr Video licenses.

`--node-id`: Perform the action specified by the `--action` command for a specific node-id. If the node-id parameter is not specified, the node-id of the current node (the server on which you are running `bv_activation`) will be used.

`--license-id`: Perform the action specified by the `--action` command for a specific license-id. If the license-id parameter is not specified, the license-id found in the Beamr Video license file located at `/opt/beamrvideo/system/run/beamrvideo.lic` will be used.

--`instances`: Specifies the number of instances to activate. This parameter is used only with the `--action activate` command.

Optimizing Streaming Content

Many providers encode each content item into a set of bitrates and resolutions, and then use Adaptive Bitrate (ABR) streaming techniques to select the most suitable version of the stream for each client based on available bandwidth and client capabilities. Beamr Video can accept this set of streams and optimize each one to the lowest bitrate possible, providing significant improvement in the overall streaming experience.

Beamr Video places IDR frames in the output video stream in the same locations that have IDR frames in the input video stream. This feature ensures that multi-bitrate ABR source streams that have synchronized IDRs to enable stream switching will remain synchronized after Beamr Video processing. Beamr Video also preserves the H.264 profile and level of the input stream, and the maximum number of consecutive B-frames. This ensures that playback devices compatible with the original stream will also be able to play back the Beamr Video optimized stream.

Troubleshooting

Listed below are some issues that may occur when using Beamr Video, and suggestions for solving these issues.

Issue: ERROR: Cannot create xxx_mini.mp4

Cause: `beamrvideo` does not have write permissions to the output folder

Solution: Provide write permissions using the following command, or specify a different output folder.

```
sudo chmod a+w output-folder-path
```

Issue: ERROR: Cannot open input file xxx.mp4

Cause: `beamrvideo` does not have read permissions to the input file or folder

Solution: Provide permissions using the following command

```
sudo chmod a+r full-path-to-input-file
```

Issue: ERROR: One of the Celery services is not running.

Cause: One or more of the `beamrvideo` background services is not running. To verify this is the cause, run the following command

```
beamrvideo_mgr server status
```

Verify that all the components of the Beamr Video server (celery, redis, dashboard, etc.) have the status "RUNNING".

Solution: If one or more of them have a different status, restart the server by executing the

following command:

```
beamrvideo_mgr server restart
```

After restarting the server, check the status again:

```
beamrvideo_mgr server status
```

Issue: ERROR: supervisor daemon not running.

Cause: The `beamrvideo` supervisor watchdog is not running. To verify this is the cause, run the following command:

```
 ps -ef | grep supervisor | grep beamrvideo
```

if no running processes are shown, this is the cause

Solution: Restart the `beamrvideo` supervisor daemon with the following command:

```
sudo service bv_supervisord start
```

Issue: bv_supervisord: unrecognized service

Cause: The installation processes failed to register the supervisor service on the system.

Solution: Manually start the supervisor process with the following command:

```
/opt/beamrvideo/bin/supervisord -c /opt/beamrvideo/etc/supervisord.conf
```

Issue: After restarting `beamrvideo`, the `celery-optimize` component fails to start. and remains in FATAL or ERROR state.

Cause: The license could be misconfigured. To verify use the following command:

```
bv_activation
```

Verify the tool output includes the license-id, node-id, and shows how many instances are activated on your node (server).

Solution: Activate the node using the following command:

```
bv_activation --action activate --license-id your-license-id --instances
your-instance-count
```

See the “Licensing” section for more details.

Issue: After restarting `beamrvideo`, the `redis` component fails to start. and remains in FATAL or ERROR state

Cause 1: A second process of `redis` is already running. Verify this is the case with the following command:

```
ps -ef | grep redis-server | grep beamrvideo
```

if no output is shown, this is not the cause

Solution: Manually kill the running `redis` process, and then restart all `beamrvideo` services:

```
sudo killall redis-server
beamrvideo_mgr server restart
```

Cause 2: `redis` has encountered an error with the persistence logs.

Solution: Use the `redis-check-aof` utility to fix the logs:

```
sudo apt-get install redis-tools
cd /opt/beamrvideo
sudo redis-check-aof --fix /opt/beamrvideo/var/run/appendonly.aof
beamrvideo_mgr server restart
```

Support

If you have any questions about Beamr Video technology or the Beamr Video optimizer, please contact support@beamr.com.