**IBC 2013 EBU demo**

Version 04 July 2013

**Title**: EBU Media Cloud Orchestrating

**Subtitle**: Open Source Cloud Infrastructure for Encoding and Distribution (OSCIED)

**Github**: From the opening of the IBC the OSCIED software is available on the EBU GitHub repository.

**Description**: The demo will show elastic scaling of virtualised encoding and distribution services in a hybrid cloud. A broadcaster is envisaged that needs to efficient allocate recourses to 3 main content flows related to online delivery:

1. Encoding of a constants content flow of live and on-demand files
2. A variable amount concurrent users during the day
3. Transcoding of an archive library for on-demand services

The first operation requires a constant capacity, the second typically peaks in the evening hour while the third one is not a daily routine but more an onetime effort with a separate allocated budget. Even though one can efficiently allocate recourses to a constant process, this is not the case when one has to permanently install servers for a one time action or to handle a short traffic peak.

Virtualisation of services using a hybrid cloud setup will optimise recourses and minimise operational costs by elastically changing the amount of encoding or distribution nodes in a private and public cloud. The system is hybrid as there is a payoff between temporally rented and structural allocated capacity. The flexibility of the cloud approach also allows to adapt sudden growth of popularity of a service as one can temporally upscale the capacity.

*Illustration 1: While the capacity of the local cloud can be accessed constantly the public cloud can be used to offload the peaks.*

**Example:**

Let's say a broadcaster operates a private cloud of 10 machines supporting 160 encoding or distribution nodes as visualised in the illustrations. This rack should be located close to the master control room in order to reduce latency and costs for transporting a constant feed the high quality source material (live and on-demand) over the local IP-network\*. In this example 40 nodes are needed as minimum during broadcast hours to encode the constant flow of linear programmed content. During the day the distribution nodes are up-scaled conform the activity patterns of the audience. Normally concurrent use grows slowly during the day while peaking around 8 pm with also a small temporal increase at 8 am. As is it is a shared environment rest capacity can be used all times to transcode the archive.

When content is extra popular, this can be the peak time of the day or a program that goes viral, the public cloud is addressed to temporally step in. For example peak offload in between 17:00 and 01:00 hours. Scaling distribution nodes in the cloud can also be helpful if it improves the peering with certain ISPs. While competing cloud provider have different price levels and some of them use different pricings for time slots during the day one could also use the cloud to budget efficient speed up transcoding of an archive. In the illustration add 200 nodes between 1 and 7 AM, none between 7 AM and 7 pm and 100 after that hour and twelve o'clock in the night.

*Illustration 2: The total constant capacity of the private cloud can be used by different virtualised services during the day. The public cloud is only used for offloading distribution peaks or cost efficient upscaling of transcoding jobs.*

\*The private cloud can also be located in gateways close to important POPs and connected via a fiber-ring. This scenario is not reflected in the demo but would only set some parameters mainly related to the bitrate of the contribution files.

2 virtualised services in OSCIED:

1. Encoding: Archive encoding where you have a large media set that needs to be transcoded to different output file formats. Capacity needs to be balanced between the time it takes to transcode the files and the related cost.
2. Distribution: Live encoding is a constant dataflow optimised for low latency and an audience viewing at different times during the day with different devices. Therefore encoding capacity is almost constant and done in the private cloud while distribution is variable and automatically scaled up in the cloud.

**Questions and answers**

An encoding node is a virtualised transoding server, in OSCIED based on ffmpeg/DashCast. The output of one node depends on the allocated CPU (perhaps also GPU support?), RAM and IO-speed (data speed from storage to and from CPU). Q: How does this work in a cloud?

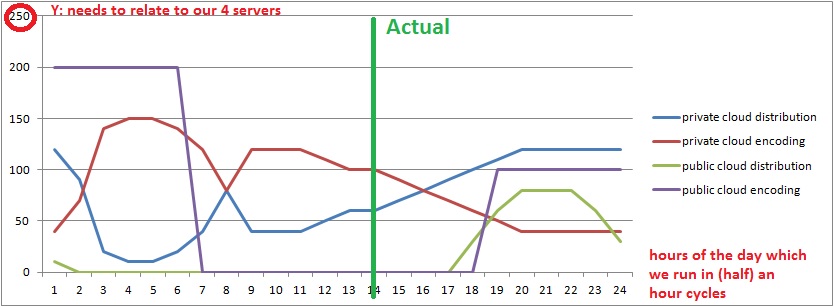
A distribution node is in OSCIED a virtualised Apache webserver using Unified Streaming MOD264 module. The output is less intensive for CPU, RAM and IO-speed but needs to output over IP the concurrent streams to the end user. Normally a server with a dual Gigabit card can serve 2500 concurrent streams of 800kbs. Q: How does this work in a cloud?

Questions

1. What it the optimal output of a node?
2. How do we calculate the cost of a node in private and public cloud?
3. How to identify what the ideal state is between private and public cloud?
4. How to calculate the data costs from the private to the public cloud?

**Physical setup and logistics:**

* One big screen showing the private and public cloud being used at one moment. We simulate an amount of nodes during the day but run this cycle every (half) hour according.



* 4 servers for demo (can we see them through a window?) connected with a GB wireless router.
* An internet connection with maximum upload and download capacity (budget dependant).
* 1 laptop that is used to demonstrate the user interface and IO. This laptop will also be used to show the software layers of this project.
* 1 iPad and 1 Android machine for playout of files on distribution nodes of the private and public cloud