

Handling live content worldwide distribution through cascaded splitting

adolfo.rosasgomez@telefonica.com

December 2012

Introduction:

In this short paper I want to analyze the impact of different architectures in worldwide distribution of live streams over internet: cascaded splitting vs. non-cascaded multi region local splitting.

In a worldwide CDN overlaid on top of Internet we will need to pay attention to regional areas (REG) equipped each one with a single Entry-point that is a First Level Splitter (FLS) that gathers many input channels from customers. We have in the same REG many endpoints/streamers or Second Level Splitters (SLS) that split each input channel into many copies on demand of consumers. The SLS are classical splitters, classical caches in the border of any current CDN, in brief: Endpoints. They talk to consumers directly. Each endpoint must obtain its live input from the FLS output. FLS + endpoints act as a hierarchy of splitters inside the REG.

We have two very different scenarios:

-Non cascading entry-points: every endpoint in a REG can connect to local channels (local FLS) and can connect to far (foreign) channels (foreign FLSs). The FLS in the REG receives as input ONLY local channels and gives as output all endpoint connections: local and foreign.

-Cascading entry-points: endpoints in a REG connect ONLY to the FLS in that REG. The FLS in a REG receives as input local channels and also foreign channels from other REGs. The FLS gives as output ONLY local connections to local endpoints.

We want to compare these two scenarios in terms of:

-Quality of connections: the shortest the connection the less latency, the less losses and the highest usable throughput. We want to keep connections local (both ends in the same REG) as much as possible.

-Capacity of worldwide CDN: We do not want to open more connections than strictly needed.

So we need to count connections in both scenarios and we need to evaluate the average length of connections in both scenarios.

There are also other properties of connections that matter in our study:

-input connections cost MORE to a splitter than output connections. This applies both to FLS and endpoint.

(Note: Intuitively every input connection is a 'write' and it is different from any other write. It requires a separate write buffer and separate quota of the transmission capacity. Output connections on the contrary are 'reads'. Two reads about the same input have separate quota of transmission capacity but they share the same read buffer. This buffer can even be exactly the same write buffer of the input. So creating an input connection is more expensive than creating an output connection. It consumes much more memory. Both input and the output from that input consume exactly the same transmission capacity quota.)

Statements and nomenclature:

O = Number of REGs

S_i = Number of source channels in REG 'i'

E_i = Number of endpoints in REG 'i'

p_{ihk} = Probability that channel 'k' ($1 \leq k \leq S_h$) originated in REG 'h' is requested by ANY consumer in REG 'i'. (Note that it is possible that $i=h$, and it is also possible that $i \neq h$.)

p_{eijhk} = Probability that channel 'k' ($1 \leq k \leq S_h$) originated in REG 'h' is requested by ANY consumer in Endpoint 'j' ($1 \leq j \leq E_i$) in REG 'i'. (Note that it is possible that $i=h$, and it is also possible that $i \neq h$.)

d_{ik} = distance REG 'i' to REG 'k'. (Note that, as usual, $d_{ii}=0$, $d_{jk}=d_{ki}$.)

Analysis

We try to compare two architectural options that present a problem in worldwide distribution of channels. This problem will not be present if we just had local channels that do not need to be distributed far apart of production site. We will not have this problem also if there was a single source and a worldwide distribution tree for that source, in that case we will build a static tree for that source that could be optimal. What I'm dealing with here is the choices that we have in case there are many live sources in many regions and there is cross-consumption of live distribution from region to region. Questions are:

-which is the best splitting architecture in terms of quality and CDN effort?

-which are the alternatives?

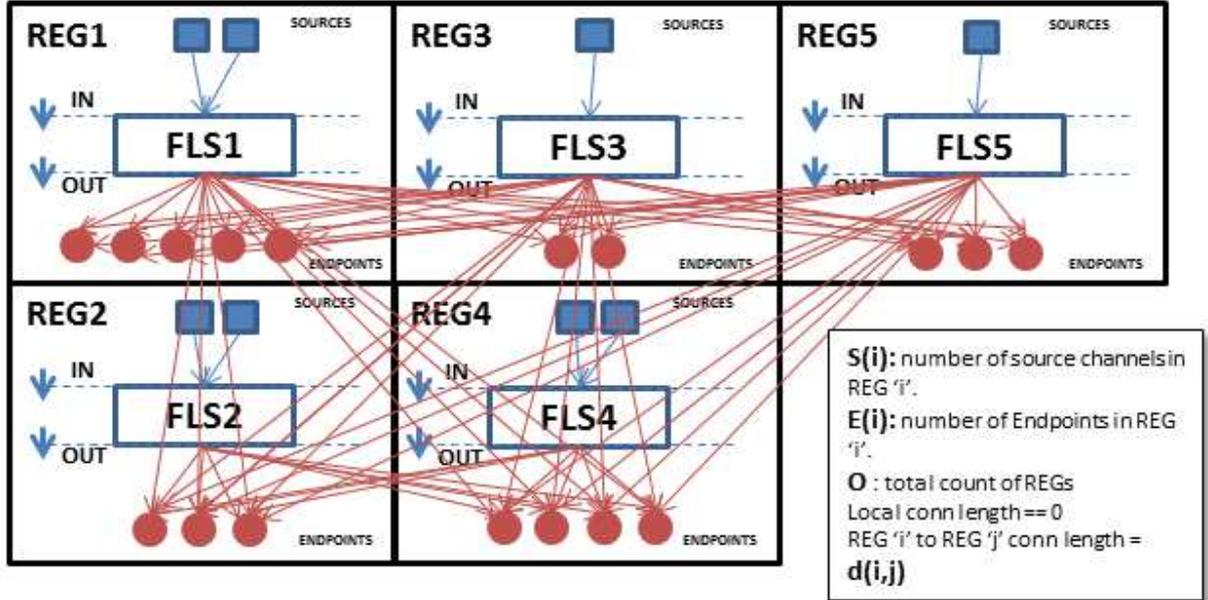
-which is the quantitative difference between them?

I've presented two alternatives that exist in the real world

- 1) Non-cascading distributed splitting: two level splitting: first at entrypoint, second at the edge, in endpoints. When a source is foreign, the local endpoints must establish long reach connections.
- 2) Cascading distributed splitting: two level splitting: first at entrypoint second at the edge AND entrypoint to entrypoint. When a source is foreign the local endpoints are not allowed to connect directly to it. Instead the local entrypoint connects to the foreign source and turns it local making it available to local endpoints as if had been a local channel.

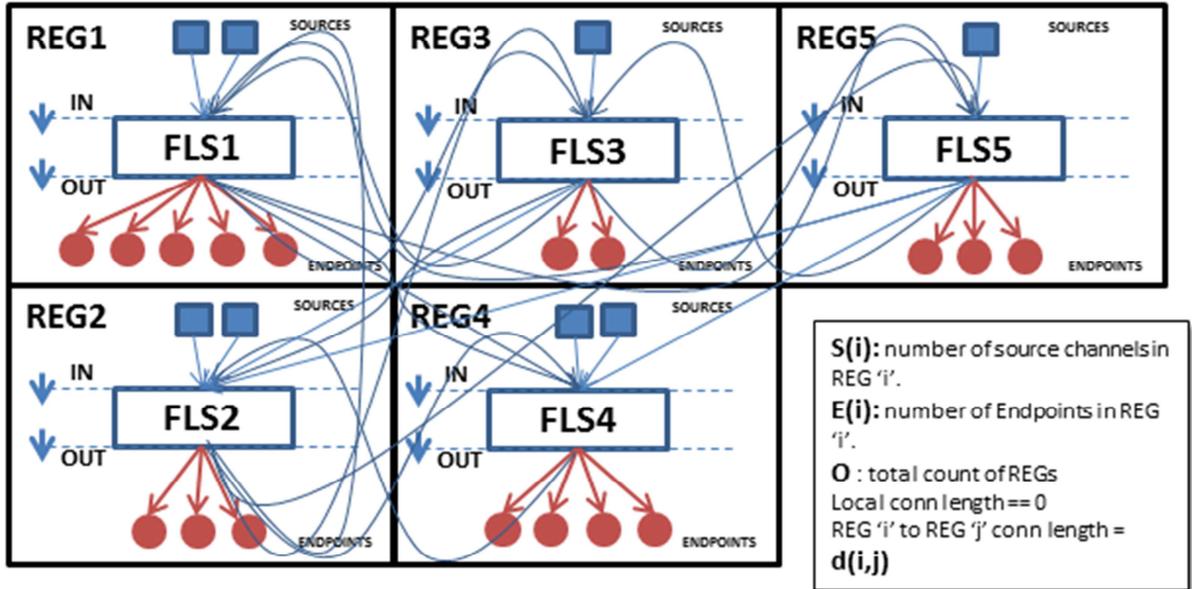
In the following pages you can see a diagram of several (5) REGs with cross traffic and the analysis of CDN effort and resulting quality in terms of the nomenclature introduced in the above paragraph.

Non-Cascading entrypoints



Element / IN / OUT	Connection count	Avg. distance
Source 'j' in REG 'i' IN	---	---
Source 'j' in REG 'i' OUT	1	0
FLS 'i' IN local	$INL_i = \sum_{k=1}^{k=S_i} p_{iik}$	0
FLS 'i' IN foreign	$INF_i = 0$	
FLS 'i' OUT local	$OUTL_i = \sum_{j=1}^{j=E_i} \sum_{k=1}^{k=S_i} p_{e_{ijk}}$	0
FLS 'i' OUT foreign	$OUTF_i = \sum_{h=1}^{h=O} \sum_{j=1}^{j=E_h} \sum_{k=1}^{k=S_i} p_{ehjik}$	$ADOUTF_i = \frac{\sum_{h=1}^{h=O} d_{hi} \sum_{j=1}^{j=E_h} \sum_{k=1}^{k=S_i} p_{ehjik}}{OUTF_i}$
EP 'j' in REG 'i' IN	$IN_{ij} = \sum_{h=1}^{h=O} \sum_{k=1}^{k=S_h} p_{e_{ijk}}$	$ADIN_{ij} = \frac{\sum_{h=1}^{h=O} d_{hi} \sum_{k=1}^{k=S_h} p_{e_{ijk}}}{IN_{ij}}$
EP 'j' in REG 'i' OUT	Not important for us.	0

Cascading entrypoints



Element	Connection count	Avg. distance
Source 'j' in REG 'i' IN	---	----
Source 'j' in REG 'i' OUT	1	0
FLS 'i' IN local	$INL_i = \sum_{k=1}^{k=S_i} p_{iik}$	0
FLS 'i' IN foreign	$INF_i = \sum_{h=1}^{h=O} \sum_{k=1}^{k=S_h} p_{ihk}$	$ADINF_i = \frac{\sum_{h=1}^{h=O} d_{hi} \sum_{k=1}^{k=S_h} p_{ihk}}{INF_i}$
FLS 'i' OUT local	$OUTL_i = \sum_{j=1}^{j=E_i} \sum_{k=1}^{k=S_i} p_{e_{ijk}} + \sum_{j=1}^{j=E_i} \sum_{h=1}^{h=O} \sum_{k=1}^{k=S_h} p_{e_{ijhk}}$	0
FLS 'i' OUT foreign	$OUTF_i = \sum_{h=1}^{h=O} \sum_{k=1}^{k=S_h} p_{hik}$	$ADOUTF_i = \frac{\sum_{h=1}^{h=O} d_{hi} \sum_{k=1}^{k=S_h} p_{hik}}{OUTF_i}$
EP 'j' in REG 'i' IN	$IN_{ij} = \sum_{h=1}^{h=O} \sum_{k=1}^{k=S_h} p_{e_{ijhk}}$	0
EP 'j' in REG 'i' OUT	Not important for us.	0

Practical calculations:

It is clear to anyone that the most difficult to obtain piece of information is the list of probabilities p_{ihk} and $p_{e_{ijhk}}$.

No matter the complexity of our real world demand, there are some convenient simplifications that we could do. These simplifications come from our observation of real world behavior:

Simplification 1:

p_{ihk} is always '0' or '1'. A channel is either 'exported for sure' from REG 'h' to REG 'i' or 'completely prohibited'.

(Note: there are so many consumers in REGs that once a channel is announced in a REG it is really easy to find someone interested in it. A single consumer is enough. If one out of 500000 wants the channel in REG 'i' then the channel MUST be exported to REG 'j'. The probability p_{ihk} here models behavior of isolated people and so we observe that once the population is high enough p_{ihk} approximates 1. If the channel 'k' in REG 'h' is geo-locked to its original REG, it cannot be requested by any REG 'i', and then p_{ihk} is 0.

Simplification 2:

p_{ijk} is always '1'. Following the same reasoning that we see above, once a channel is announced in a REG it is always consumed in that REG.

Simplification 3:

$p_{e_{ijhk}}$ has approximately the same value for all 'j' in the REG 'i'. That means that every endpoint 'j' in REG 'i' receives equal load of requests about channel 'k' from REG 'h'. Why is this? This happens Just because endpoints receive requests that are balanced to them by a regional Request Router (usually a supercharged local DNS). The aim of the Req. Router is to distribute requests evenly to streamers working for the REG.

P.S: this behavior depends on implementation of req. Router. A reasonable strategy is based on specializing endpoints in some content (through the use of URL hashes for instance), and thus a given channel will always be routed through the same endpoint. If correctly implemented this would mean that for a given pair (h,k), $p_{e_{ijhk}}$ is '0' for all values of 'j' except one in REG 'i', and for that value of 'j' $p_{e_{ijhk}}$ is '1'.