

guifi.TV - Suport #59

proves de multiples connexions client

09-07-2015 17:43 - Eloi Rebes

Estat:	In Progress	Inici:	02-06-2016
Prioritat:	Normal	Data de venciment:	
Assignat a:	eloi rebes	% realitzat:	70%
Categoria:		Temps previst:	0.00 hora
Versió objectiu:		Temps invertit:	5.00 hores
Descripció			
<p>Plantejar una prova de multiples connexions client per saber el rendiment, estabilitat, i limit d'usuaris simultanis possibles. Usant la plataforma Community-Lab que en participa la UPC¹, creant multiples nodes i cadascun amb unes 5 connexions cap al servidor d'streaming.</p> <p>Comanda per connectar-se com a client, des de terminal, i sense consumir recursos del sistema client:</p> <pre>rtmpdump -r "rtmp://tv.guifi.net/live/guifitv" -a "live/guifitv" -f "LNx 11,2,202,425" -W "http://tv.guifi.net/player.swf" -p "http://tv.guifi.net/" -y "guifitv" -v --quiet > /dev/null</pre> <p>o be:</p> <pre>for i in \$(seq 10); do rtmpdump -r "rtmp://tv.guifi.net/live/guifitv" -a "live/guifitv" -f "LNx 11,2,202,425" -W "http://tv.guifi.net/player.swf" -p "http://tv.guifi.net/" -y "guifitv" -v --quiet > /dev/null & done</pre> <p>o be sense resolucio de nom, nomes per guifi:</p> <pre>for i in \$(seq 10); do rtmpdump -r "rtmp://10.38.141.73/live/guifitv" -a "live/guifitv" -f "LNx 11,2,202,425" -W "http://10.38.141.73/player.swf" -p "http://10.38.141.73/" -y "guifitv" -v --quiet > /dev/null & done</pre> <p>i per parar les 10 simultanees, la ordre següent:</p> <pre>killall rtmpdump</pre> <p>·Cal tenir instal·lat el rtmpdump ·No cal disposar d'entorn de finestres</p> <p>[1]Community-Lab https://wiki.confine-project.eu/_media/wiki:logo.png</p>			

Historial

#1 - 09-07-2015 18:22 - Eloi Rebes

- Descripció updated

#2 - 09-07-2015 18:24 - Eloi Rebes

- Descripció updated

#3 - 26-10-2015 16:34 - ivilata ivilata

Recomane llegir la [Guia de l'administrador d'slices](#) per a aprendre com desplegar experiments o *slices* a Community-Lab.

L'experiment és perfectament possible sota Community-Lab, ja que només s'empraria trànsit a nivell d'aplicació. A més a més, parlem de connexions

d'eixida cap a un servidor que té adreces de Guifi i d'Internet, per tant podem emprar nodes de Community-Lab de tot tipus tant si tenen connexió a Internet com si només tenen connexió a Guifi, i tant si tenen adreces natives de la xarxa on estan com si estan darrere de NAT.

Si el codi de l'experiment s'escriu amb gràcia (p. ex. provant primer l'adreça de Guifi i si no funciona la d'Internet —a la UPC alguns nodes de Guifi tenen accés NAT a Internet) es pot fer independent de la xarxa a què estiga connectat el node. Si és així, totes les *slivers* (és a dir, cada màquina virtual que corre a cada node per a l'experiment) es poden configurar de forma idèntica, simplificant la preparació. Jo els configuraria:

- Una plantilla basada en Debian (de les que ja hi haja a Community-Lab).
- Una interfície de gestió (*management*) per si cal connectar-se a la *sliver* via SSH des de la xarxa de gestió del *testbed* per a gestionar l'experiment. No cal configurar-los una interfície pública ja que no esperem connexions entrants cap a les *slivers*. El node farà NAT de l'adreça privada (interna) de la *sliver* en la connexió cap al servidor, però això no ens hauria de preocupar.
- Reserves de memòria i disc mínimes, ja que només cal executar *rtmpdump* sense desar les dades. Per al consum de memòria es pot fer una execució de prova (probablement amb 512 MiB hi haja més que de sobra), per al disc bàsicament el que calga per a la instal·lació del paquet *rtmpdump* i dependències com a *sliver data*.
- Un mateix *sliver data* bolcat després d'instal·lar el paquet Debian *rtmpdump* (com es descriu sota *Preparing an experiment* a la Guia).

Tot açò es configuraria com a *sliver defaults* comuns a la *slice* de l'experiment.

També caldria pensar com posar en marxa l'experiment una volta estiguen corrent les *slivers* (si cadascuna a la seua bola, si totes a una hora, si amb sincronització automàtica o manual...), i com recollir i mesurar els resultats, però això ja depèn de l'experiment que es vulga fer. I també caldria triar adequadament els nodes a participar en l'experiment.

Des del meu punt de vista el més difícil és crear l'arxiu amb *sliver data* (emprant VCT), però és més feineta que quelcom difícil. Hi puc ajudar sense problemes.

#4 - 02-06-2016 19:47 - Eloi Rebes

- *Descripció updated*

#5 - 02-06-2016 19:49 - Eloi Rebes

- *Descripció updated*

#6 - 02-06-2016 23:44 - Eloi Rebes

- *Descripció updated*

#7 - 09-06-2016 00:13 - Eloi Rebes

- % *realitzat* ha canviat de 10 a 70

#8 - 09-06-2016 00:13 - Eloi Rebes

- *Assignat a* ha canviat de *eloi rebes* a *Eloi Rebes*

#9 - 09-06-2016 00:26 - Eloi Rebes

- *S'ha afegit Fitxer 2016-06-03_provesstressstreaming.pdf*

- *Estat* ha canviat de *New* a *In Progress*

- *Assignat a* ha canviat de *Eloi Rebes* a *eloi rebes*

- *Inici* ha canviat de *02-07-2015* a *02-06-2016*

gracies a 3 guifilabs junt amb Ivan Vilata, ja hem pogut fer les primeres proves!

#10 - 12-06-2016 17:10 - ivilata ivilata

Eloi Rebes va escriure:

| *gracies a 3 guifilabs junt amb Ivan Vilata, ja hem pogut fer les primeres proves!*

Bona feinada Eloi, els resultats són impressionants, el servidor aguanta molt bé. M'alegra que la comunitat faça ús de Community-Lab!

Fitxers

2016-06-03_provesstressstreaming.pdf	1.473 MB	08-06-2016	Eloi Rebes
--------------------------------------	----------	------------	------------