Posted to the Gilder forum - December 5, 2000

Challenge to GG's Caching Vision

posted to: Telecosm Lounge
poster: Rich Fagan
date: 12/5/00 3:09:39 AM

The following is written by Frank Levinson, Chief Technology Officer and one of the founders of Finisar. Levinson writes monthly essays on various technology topics and publishes them on Finisar's public website. Good stuff.

This essay, written in June, addresses GG's views on caching (and the XLA/Mirror Image solution) and references text from the February 2000 GTR. Finisar, by the way, is one of the companies GG included in his "Missed Noggin Nine" post in October.

Link to essay: http://www.finisar.com/franks_news_4.html

Disclosure: Long FNSR and GG

Regards,

Rich

Frank Levinson proposes for his model to work:

>>>a much simpler solution is for HTTP and its cousins to migrate towards a single transmission request for all information and for the sender to gather together the information and send it just one time as a single group or stream.<<<

Let's have a look at a typical Yahoo! page stock quote page. The page itself is on the Yahoo! server. In addition there are ads that are served from an ad server, there are charts served from a chart server. So far there are three servers required to build the page. This is the transaction:

1. The user requests a page from Yahoo
2. Yahoo send the page to the user including the URLs of the chart and the ads that are supposed to show up on the page
3. The user gets the page and displays it
4. The user requests a chart from the chart server
5. The user requests the ads from the add server
6. The chart server sends the chart to the user
7. The ad server sends the ads to the user
8. The user gets the chart and the ads and displays them

Levinson's transaction would be as follows:

1. The user requests a page from Yahoo
2. The yahoo requests a chart from the chart server
3. The yahoo requests the ads from the add server
4. The chart server sends the chart to the yahoo
5. The ad server sends the ads to the yahoo
6. Yahoo send the page to the user
7. The user gets the page and displays it

Levinson's method is slower because it sends more data and sends it later than with the current method. (1) before Yahoo can send the page to the user it has to wait for the chart and the ads to arrive to form the page. If they don't show up, they have to be requested again and if they still don't show up,then the page has to be sent without whatever is missing but by then a whole lot of time has been wasted. With the current method, the user is the one that figures out what is missing and what to do about it and the Yahoo server can start sending its stuff immediately on receiving the request. And (2) there is more data flowing with Levinson's: The charts and the ads have to go from their respective servers to Yahoo and from Yahoo to the user. With the current method,Yahoo sends the URLs to the user, the user sends the URLs to the servers and the servers return the requested objects. With Levinson's method, the objects make two trips each and the URLs one. With the current method, the objects make one trip and the URLs two. The URLs being smaller than the objects, time and data are less with the current method.

In effect, Levinson's method requires more intelligence in the Yahoo server than the current method. On several occasions GG has talked about where the intelligence should be,at the edges and not at the center and this is a good example of it. there are additional problems but they are not worth mentioning just now.

>>>This worst case figure is really exaggerated for the point of his article … a more typical intercontinental distance is likely to be 12000 km and then the time drops to 80 ms. And for normal intra-continental distances of 3000 km this is then 17 ms. Actually most traffic will be of this latter category since each citizen/user will most of the time be fetching information localized for their country.<<<

I ran a test with Route Tracing to 13 sites in the US (I presume). They all went for Caracas to Atlanta and from there to the destination

	Total	DSL	Atlanta	US
Avg	294	90	63	141
Max	498	90	63	345
Min	179	90	62	27
Mean	213	90	63	60

I have very little access to local web sites. Most of my traffic is with Yahoo, the GTF, my web site in Seattle, DogPile, Hoover, Barnes& Noble and news sites, none of them local. I don't think I'm that different from other international users. Maybe the Japanese access Japan a lot and maybe the Europeans access Europe a lot. In any case, to the times he gives, you have to add the delays at the routers. In my case, the DSL consumes a lot of the time, 30% on the average!

>>>Let's use Gilder's assumption of 25 objects per web page for our starting place. We shall assume the cache resides at an effective distance of 250 km for a round trip time of 1.7 ms and therefore 25 objects will take 43 ms. This is substantially slower than the direct sending of the page over typical intra-continental distances of 2500 km but sent as a single object (17 ms).<<<

Assuming that Atlanta is 2000 miles (3000 KM) from Caracas, my test gave me 135 ms per page from a cache 250 KM away and 54 ms from a distant server. In either case, the relation between the cache served data and the distant server served data is 2.5 to 1. But Frank Levinson totally ignores three things:

- Error correction, and
- Page makeup
- Nodes on the network

Should a single long message not arrive, the whole message needs to be resent. If the message is split into 25 pieces, then only some smaller part of the message needs to be resent. I don't know if this is typical, but about 10% of the test packets in the Trace Route did not get through and would have been resent in real life. That would increase the time from the distant server to at least double, 108 ms and the time from a cache would increase to around 149 ms. Certain kinds of transmissions cannot take such long delays, for example VoIP. The 108 ms is continuos and not acceptable while the longer 149 ms is divided into 25 parts and is easier to handle.

The page makeup I have already discussed at the beginning.

Last but most important, the nodes of the network. If we are to go to a switchless lambda network, the number of nodes is very important. With no caches, every server has to be able to connect to every user. In a totally switch free network that requires a connection from each server to each user., But if 30 to 40 caches can service the world as GG claims, then the number of connections drops radically because you only need to connect each server to each cache (max 40) instead of to millions of users.

Denny
"Demand creates queues. Supply gets rid of them."