REVERBERATIONS OVERVIEW

A RESTful service that lives in the cloud that applications can hook via API. Code would then be written into Swift to hook the Reverberation API and do the visualizations on the server-side.

TERMINOLOGY

Stimuli – The original content (e.g. blog post, article, tweet)
Echo – Anything referencing stimuli (e.g. retweet, trackbacks, links-in, blogs referencing original content)
Influence - The number of items referencing a particular item of content
Reverberation - The relationship between *stimuli* and all the *echoes* it causes

OVERVIEW

The Reverberations platform should offer users a method for tracking how popular or influential content is across the web. Reverberations is being developed to do six things:

1. If the content is a Tweet, find and store all re-tweets as well as the timestamp of each re-tweet, the user name of the re-tweeter are, and if other re-tweets originate from a re-tweeter or from the original stimuli. (using the Twitter API)
2. If the item is a blog or news article, calculate the number of trackbacks, comments, links-in, and new blog posts referencing original content.
3. Calculate how popular content is across social media (perhaps? using the PostRank API)
4. Calculate the frequency of re-tweets and mentions, the more frequent those mentions, the 'hotter' that content is.
5. Offer a horizontally (left to right) chronological view of content ordered by timestamp.
6. Calculate an 'Influence' score for stimuli based on all of the above.

Other Requirements:

1. Must be 'persistent'. The scores should change dynamically as new echoes occur.
2. Most of the logic should be unique. Only rely on PostRank API for tertiary information (Delicious, Reddit, Digg, etc.)

ORIGINAL THOUGHTS BEHIND THE IDEA

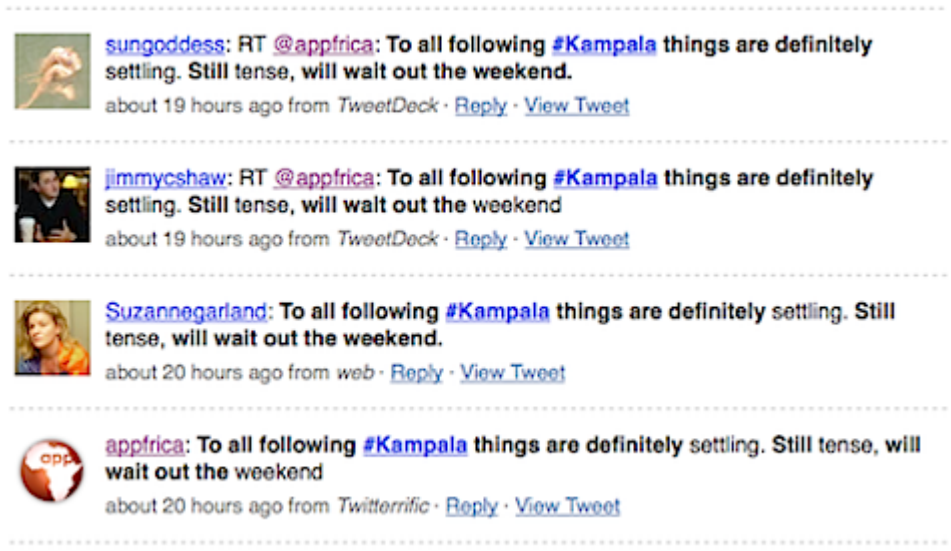
Excerpted from: http://appfrica.net/blog/2009/09/13/asynchronous-info-disjointed-data-and-crisis-reporting/

Another contextual clue that's missing from distributed micro-messaging tools like Twitter is that of time. I suppose it's not missing, it just often gets over-looked.

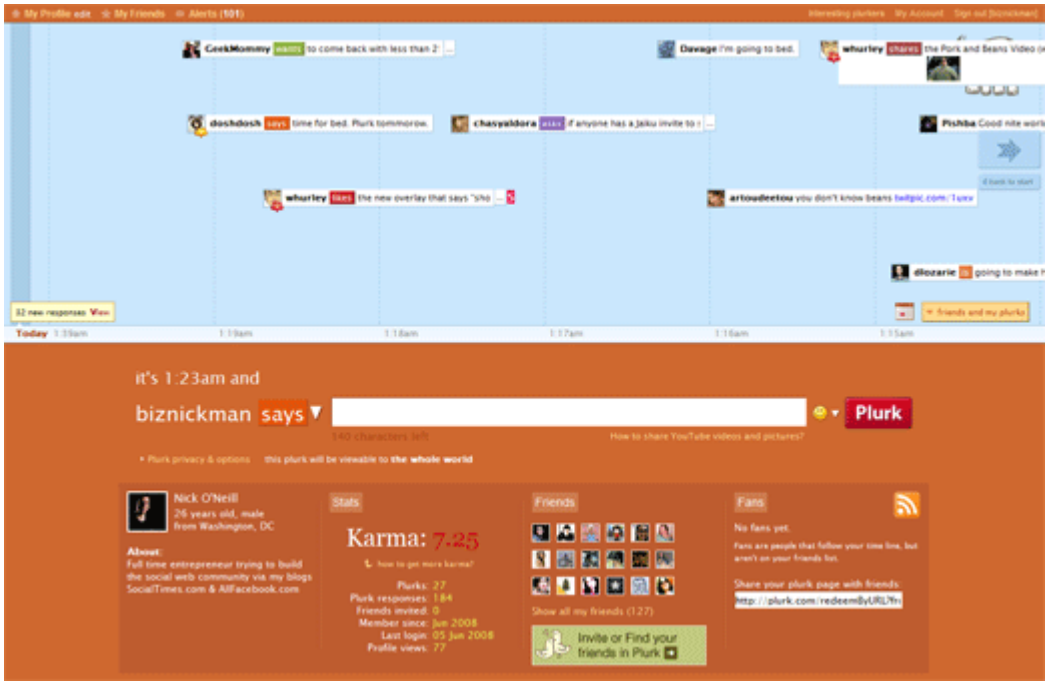
On day three of the Kampala riots we saw evidence of this. People on the ground wrote about the situation as it unfolded through out the day until things began to finally settle down. Then 8 hours later we began to see people in the US re-tweeting things that happened hours ago (in the morning for us) as 'news'. To them it was indeed news as they were just waking up to it. This in turn created confusion for us back here. ("Have the riots started again or is that just an old re-tweet?"). This is not a problem of user behavior, rather it's a problem of the UI itself. Retweeting messages from friends and followers is something users of Twitter have always done. However, what's missing is some way

to group these updates together, so as to build a more accurate timeline of activity. On services like Friendfeed when identical messages are posted, the service automatically groups them in the users stream.

At a glance now we can see that we have several messages that are so similar that they are probably identical, or at the very least related. In the case of Twitter, any and all 'retweets' could simply be grouped as they're essentially all reverberations of that one initial event that we can think of as the stimulus. (ex. My twitter handle is @jongos, I send the message "fire!", it becomes the stimulus and thus any message that says "r/t @jongos fire!" or "fire (via @jongos") is just an echo.)

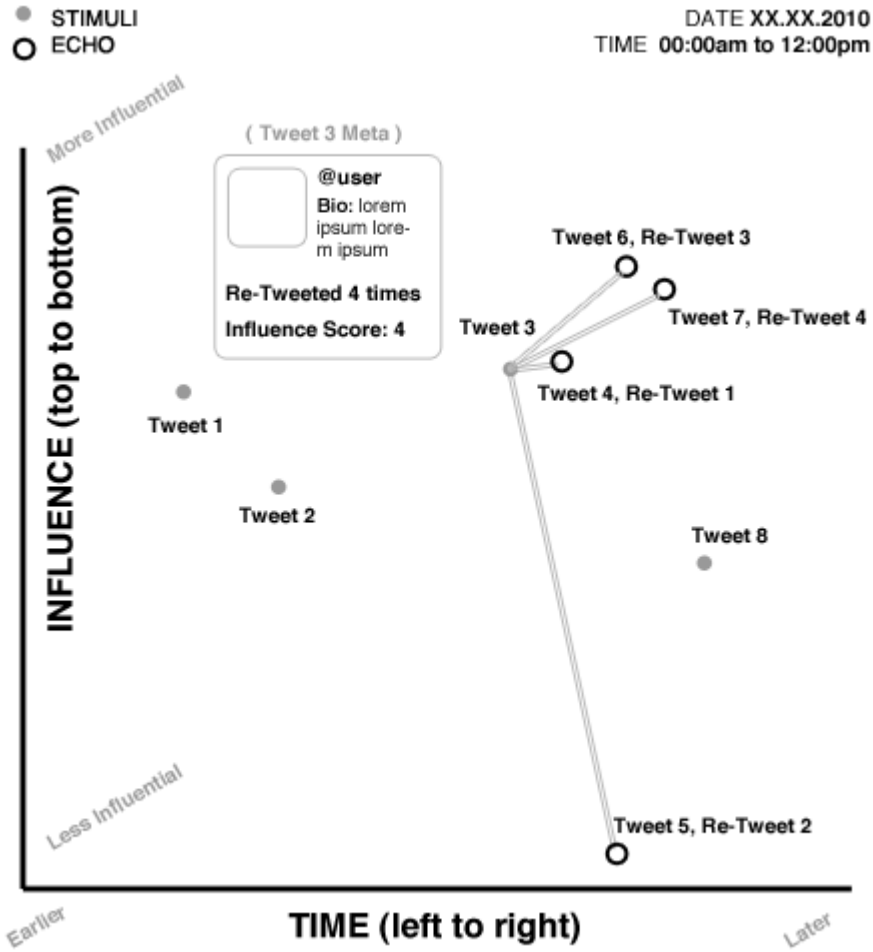


It would be more useful to have all related tweets collapsible. Additionally, it would be useful if there were a way to view these happenings with some sort of chronological barometer. If shifted along a horizontal axis, like a timeline, we'd end up with something that looks a bit like the social media application Plurk.



With Plurk the updates appear along a horizontal grid that has temporal indicators. An update at 1:14am will appear farther left than one at 1:19am. As time progresses the data flows from right to left (so as to be read from left to right).

NEW THINKING AND MOCKUP



In the figure above we are measuring time and influence of Tweets and Re-Tweets to the second order between noon and midnight. Each dot represented is a tweet positioned on a graph where the x-axis measures time and the y-axis measure influence Clicking on **Tweet 3** reveals the **Tweet 3** meta card displayed which offers more insight. We can see that **Tweet 3** has been re-tweeted 4 times. We also get the Twitter user's handle and bio. We can also pull other content in here, including the users RiverID score or other social network profiles.

So why are the re-tweets positioned differently along the y-axis? If we were to click on each user re-tweeting **Tweet 3** we'd see that those re-tweets had also been re-tweeted to a certain degree which would determine their influence. **Tweets 4, 6,** and **7** must have also been re-tweeted at least 4 times to maintain their position (which is the same or higher than **Tweet 3**.)

It's also important to note that all Re-Tweets are Tweets but not all Tweets are Re-Tweets.