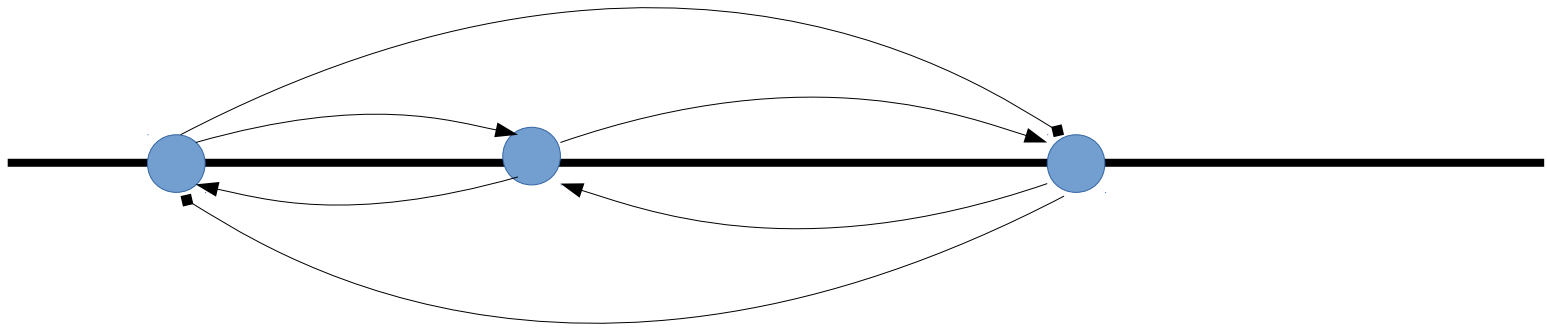# Social Media Physics

Daytime
Weekends
Spread out

Simulation of anti-gravity in a one dimensional ring shaped universe with the size 1.
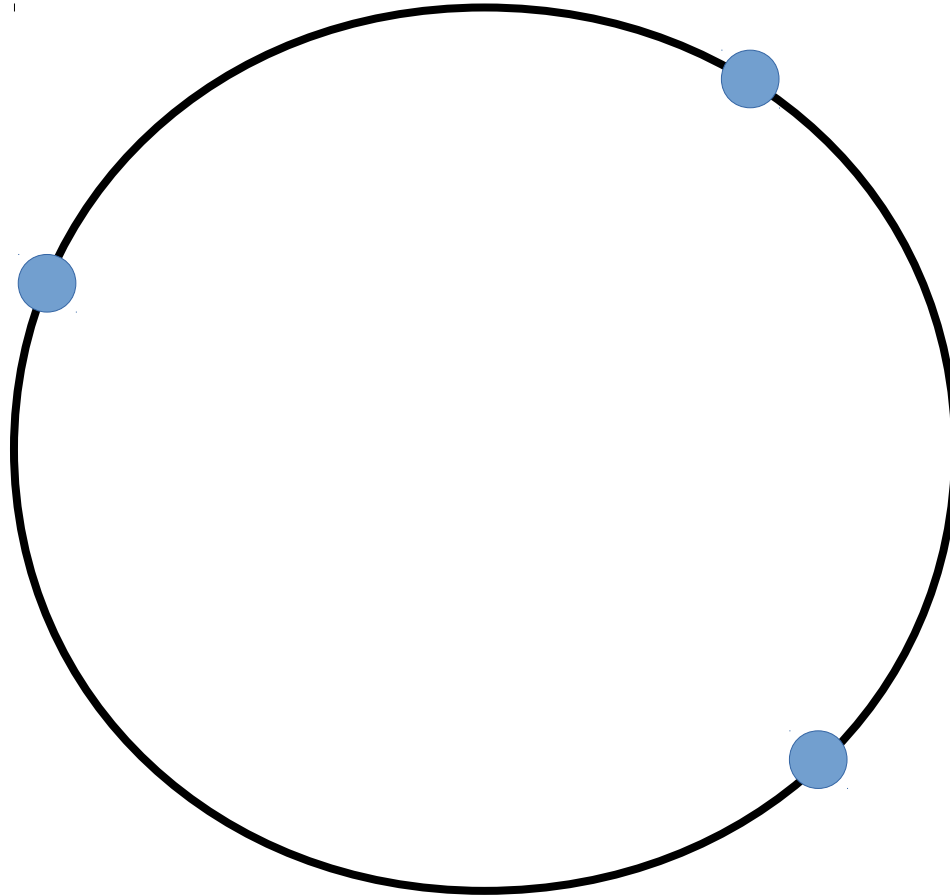
WAT?!

Simulation of anti-gravity in a one dimensional ring shaped universe with the size 1.

anti-gravity in a one dimensional universe

ring shaped universe

Demo time!

Perl 6

atikon

Social Media Physics

At work we are managing Facebook accounts of about 100 customers. We supply content that's posted on their accounts. That is, we post the same news text on 100 facebook accounts and of course we automated the whole process. Now there are a couple of requirements.

Daytime
Weekends
Spread out

Postings should be published during the day so they
do not look like cron jobs.
They should be published mostly on weekends
because there the audience is larger.
And our customers don't really like it when their next
door competitors post the same news item at the
same time.
These are all rather fuzzy requirements, so it's fair to
say, that there is no one correct solution. Some
schedules are better than others.
So when I got these requirements I thought, oh, the
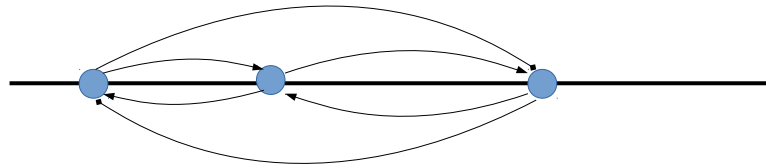best way to find a good schedule is to treat it as a

Simulation of anti-gravity in a one dimensional
ring shaped universe with the size 1.

WAT?!

Simulation of anti-gravity in a one dimensional
ring shaped universe with the size 1.

Ok, seriously, what the hell am I talking about?

anti-gravity in a one dimensional universe

Well, we start with a time line and call it "one dimensional universe".

On this time line we place our Facebook postings.

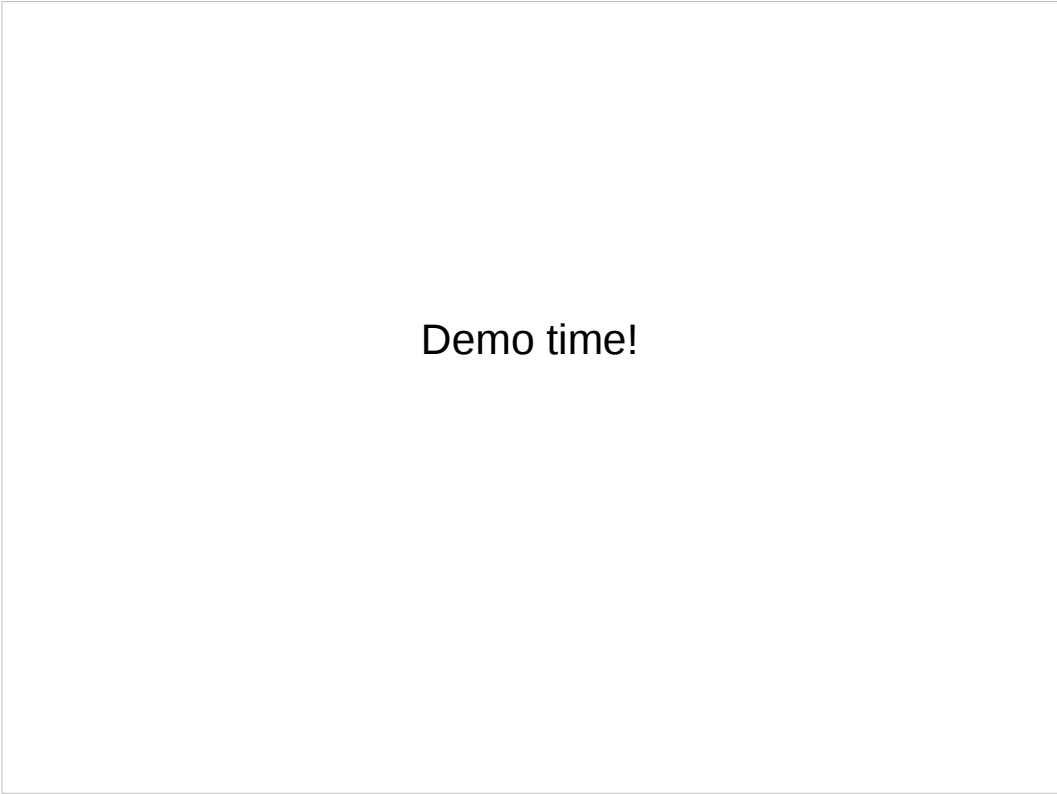Then we define that those postings repel each other. They all push against the other postings. And the closer two postings are, the harder they push, hence anti-gravity.

The simulation part is basically a loop where for each iteration we calculate for each posting the sum of all pushing forces. Then we move the posting accordingly.

ring shaped universe
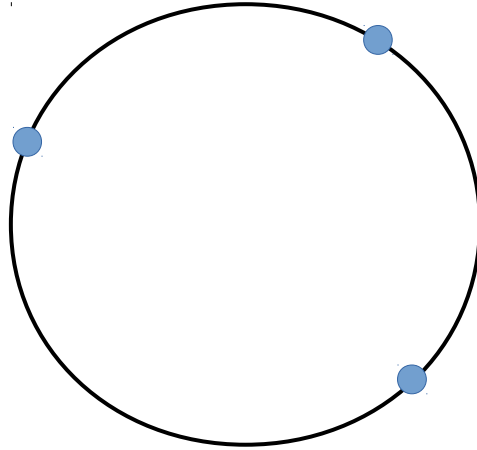
As this would lead to postings just being pushed to the ends of our time line, we bend this line into a ring, so there no longer is a beginning and an end.
And to make calculations easier, we define this ring to have the circumference of exactly 1.

Demo time!

To avoid postings being placed in the middle of the night and during the week, we add some static points for each night and on Wednesdays that just push postings away.
The result looks a bit like this:

Perl 6

Now there are a couple of parameters that need tuning
 for the simulation to give good results.
I didn't want to figure them out manually, so I had the
 computer just try them.
But as you've seen the simulation takes a lot of time,
 so I tried to parallelize it.
The obvious way is to use Perl 6 and Inline::Perl5 and
 run each simulation in its own thread and collect the
 results.
Yes, that's why I have a threading example in my
 Inline::Perl5 talk now :)

The morale of the story: I'm head of IT of a company called Atikon where we actually enjoy pulling these stunts.

And we're hiring.

You'd be working for me, that means, your boss would understand a bit about software engineering and understand that proper solutions take time that's well spent!

So please talk to me, if you're interested.