Gnip.com Intro

(*and* a SMART@znmeb update at no extra optional additional charge)

M. Edward (Ed) Borasky

Portland Data Plumbers

June 23, 2009

Where Is It?

Gnip Home Page: `http://www.gnip.com/`

What Does It Do?

Collects activity data from publishers and provides an API for accessing the data

How Does It Work?

- An *actor* performs an *activity* on one of the publisher sites.
- Gnip collects these activities and re-publishes them in a standardized XML format.
- Users then access the activites via the Gnip API.

What Publishers Are There?

clipmarks, dailymotion, delicious, deviantart, digg, diigo, flickr, flixster, fotolog, friendfeed-search, gamespot, hulu, hulu-search, identica, ilike, intensedebate, multiply, photobucket, photobucket-search, plurk, reddit, seesmic, slideshare, smugmug, smugmug-search, stumbleupon, tumblr, twitter, twitter-search, vimeo, webshots, xanga, youtube, youtube-search

Twitter API Change!

As of last week, the "Twitter" publisher for free Gnip accounts only has access to the Twitter "spritzer" streaming feed. This is a step down (way down!) from their previous Twitter publisher. It's not clear to me what value Gnip adds in this case. Stay tuned. :-(

How Do You Use It?

- Get an account: both free and paid available
- Create filters
- Either read the data from the filters or have the filters push data to your collector

What's In A Filter?

- Optional POST URL if you want the filter to push data to your collector
- Choice of full data or just notification
- Rules to specify the activites to filter in
  - List of (from) *actors* to be tracked
  - List of publisher-specific *regarding* tokens
  - List of publisher-specific *source* tokens
  - List of *tags* to be tracked
  - List of *to* (actors) to be tracked
  - List of *keywords* to be tracked
- Not all publishers use all rule types
- A filter must specify at least one rule

Example: Delicious Link Updates

- Actors: znmeb
- Regarding: null
- Source: not applicable to Delicious
- Tag: ruby
- To: not applicable to Delicious
- Keywords: not applicable to Delicious

What Does That Give Me?

- Activity *buckets*
- Each bucket is an XML document covering all activities within one minute
- Buckets remain on line for an hour
- There is a REST API to retrieve the data from the bucket
- As noted above, one can also have Gnip POST the activities to a subscriber

A Sample Result

- I created a filter: Delicious.com, actor=znmeb, tag=ruby
- I posted a link about the Ruby Benchmark Suite with tags “ruby”, “benchmark” and “benchmarking”
- And the XML is ...

XML Result

<activities publisher="delicious">

<activity>

<at>2009-06-22T22:34:16.000Z</at>

<action>bookmark</action>

<activityID> http://delicious.com/url/989dc13ff450623702c9c61ae04848d8#znmeb </activityID>

<actor metaURL="http://delicious.com/znmeb/">znmeb</actor>

<destinationURL metaURL="http://delicious.com/url/989dc13ff450623702c9c61ae04848d8">

http://github.com/acangiano/ruby-benchmark-suite/tree/master </destinationURL>

<tag metaURL="http://delicious.com/ruby">ruby</tag>

<tag metaURL="http://delicious.com/benchmark">benchmark</tag>

<tag metaURL="http://delicious.com/benchmarking">benchmarking</tag>
<regardingURL> http://github.com/acangiano/ruby-benchmark-suite/tree/master </regardingURL>

More XML

<payload>

<title> acangiano's ruby-benchmark-suite at master - GitHub </title>

<raw>

H4sIAOwGQEoC/63TTU+DMBgH8LufouLBg25lDCeQjmRxicZoYoy3ZYdSHlgzaJe+uMxPb8cGusVE

s9gLBPr8/jxtIdxAnZ6hZhDDTQUpZVSUnAp5qZGy2aaXgWCLmqplT1s3H1GDaqoNKNRD99w82Izg

XWkL-

rWw2pQbSZymuURCgRytQ4Puxu0+GYTIYoSvfDYLbiW1haXmOuH4BVdMnLpZjr6CVBi9dGLNK

MM6h4oxLq/tM1tiqCsdRnLPBsCjCG38UDG/9gMVsNKDgh1EY5dHFh6jBfeBW7mIqR7dkyc3CZo3X

dY5/6hsbBYB3nRPcEK2Xs4QpoEaqlJzP7qaTt8msyZ3PU4K/vW0LXFoNwujT+yK4M1p0XayT/cNX

3dEFQK77hwHvAVZa/zHniG3TtLSKAXLG2Ps9qlkNL20u7lxlUi63K6sJ3jlfK+OOQynVBuWyplx0

9qHaONhLt/vkVmJfcyLS7fP/SVyUxxjBzb/2CVUli49yAwAA </raw>
</payload> </activity> </activities>

So Now You're Ready For The Tutorials

- Main API Document:
  `http://docs.google.com/Doc?id=dpw6zj9_0fdcnttgd`
- Convenience Libraries: `http://github.com/gnip`
- Windows Tutorial:
  `http://www.gnip.com/docs/tutorial_win.pdf`
- MacOS X / Linux Tutorial:
  `http://www.gnip.com/docs/tutorial_mac.pdf`

Oh, By The Way ...

- It's pronounced "guh-nip"
- It's "ping" spelled backwards

What Is SMART@znmeb?

- SMART@znmeb is a *S*ocial *M*edia *A*nalytics *R*esearch *T*oolkit
- An *appliance*, available in five formats:
  - VirtualBox Virtual Machine Image
  - VMware Virtual Machine Image
  - Disk Image / Bootable USB Drive
  - Xen Virtual Machine Image
  - Bootable LiveDVD with installer
- Open Source Project on Github: `http://github.com/znmeb/twitter-appliance/tree/master`

What's In The Machine?

- A complete Linux desktop
  - openSUSE 11.1 operating system
  - Gnome 2.24 desktop
  - Mozilla Firefox 3.0 browser
  - Evolution 2.24.1.1 email / contact / calendar management package
  - OpenOffice.org 3.0 office suite
  - Pidgin 2.5.1 instant messaging / IRC client
  - Games, multimedia, imaging

And?

- PostgreSQL 8.3.7 relational database management system, including PgAdmin3
- Perl DBI package with DBD drivers for PostgreSQL, MySQL, CSV, SQLite, ODBC and XBase
- R 2.9.0 language and environment for statistical and graphical computing
- GGobi 2.1.8 data visualization system, and
- Perl Net::Twitter 2.12 Twitter API module / WWW::Mechanize "web scraper"

And?

- Ruby
- Python
- Tcl/Tk
- But primary implementation language will be Perl (5.10)
  - CPAN is your friend!

What's *Not* In The Machine?

- Gumballs, soda, peanuts, meat by-products,
- Lua, Rails, Django,
- KDE or XFCE desktops, or
- *Software with non-free-as-in-freedom licenses*

Licensing

- SMART@znmeb uses the same license as openSUSE 11.1: `http://zonker.opensuse.org/2008/11/26/opensuse-sports-a-new-license-ding-dong-the-eulas-dead/`
- openSUSE 11.1 License: `http://en.opensuse.org/OpenSUSE_License`
- If you find a package in SMART@znmeb that's not freely distributable, I will pull it from the distribution
- If you need proprietary packages (wireless, multimedia) you can get them

What Can You Do With It?

Anything you can do with openSUSE 11.1 Gnome desktop, and ...

- Interact automagically with the social web
- Manage data
- Analyze data

Interact Automagically With The Social Web

- Collect data
  - Social media or otherwise
- Create Twitter bots
- Build monitors, alerts and dashboards

Manage Data

- Internal
  - Evolution Data Server,
  - OpenOffice Base,
  - PostgreSQL,
  - CSV, and
  - SQLite
- External
  - MySQL,
  - ODBC,
  - PostgreSQL, or
  - XBase

Analyze Data

- Perl libraries are available, but not currently installed
- R & GGobi
  - CRAN library packages and task views are available, but not currently installed
- Create static & animated visualizations of your data, presentation-quality graphics
- Intended analysis realms
  - Natural language processing
  - Machine learning
  - Exploratory data analysis
  - Geospatial analysis
  - Data visualization

Why An Appliance?

- Can be run as a guest inside a Windows, MacOS X or Linux desktop / laptop virtualizer
- Can be run from a bootable USB device, or LiveDVD
- Can be backed up, duplicated, distributed and interchanged *as a whole*, complete with data & documents
- Can be deployed as a server in "the cloud"

Status

- It works as a VMware or VirtualBox VM now
  - Runs with Linux or Windows host, don't know about MacOS X
- Data collection focused on Twitter
  - "spritzer", "track" and "follow" streaming API feeds work now
  - Compressed JSON output
  - Perl filter to convert to CSV
  - Uses PostgreSQL COPY feature for ETL :-)
  - Still need to write Twitter search collector

Some Numbers For Your Am(aze|use)ment

- I started a "spritzer" capture at 2009-06-16 19:01:05 UTC
- It disconnected at 2009-06-21 22:31:00 UTC
- That's about

```
> difftime("2009-06-21 22:31:00",
+ "2009-06-16 19:01:05")
Time difference of 5.145775 days
>
```

- The compressed (bzip2) "spritzer" stream is about 393 megabytes
- Uncompressed, it's about 3.1 gigabytes

How's That Again?

- How many tweets? 2,533,434
- Or about 20,500 tweets per hour!

```
> difftime("2009-06-21 22:31:00", "2009-06-16 19:01:05", units="hours")
Time difference of 123.4986 hours
> 2533434/as.double(difftime("2009-06-21 22:31:00",
+ "2009-06-16 19:01:05", units="hours"))
[1] 20513.87
>
```

- And that's just a *sample*!

Road Map

- Next steps
  - Twitter search collector
  - Determine what needs to be done to use PostgreSQL full-text indexing / search with tweets
  - Preliminary exploration of one week of "spritzer" data
- Xen VM, bootable USB and LiveDVD available on request, but main testing will be with VMware / VirtualBox image

Why Are Twitter Text Analytics Different?

- Multiple human languages
- Links
- Hashtags
- @replies
- "RT"
- Tweets are an emerging / evolving language!
  - Example: people changed location and timezone to Tehran in response to #iranelection
- That's why my focus is on sampling the public timeline and exploratory data analysis
- But ...

I'm Looking For Use Cases And Alpha Testers

- I need some "marketing / CRM" use cases
  - I'm skeptical of many of the claims I see, especially for "sentiment analysis"
  - And I question the "need" for "24x7 brand monitoring"
- I don't know what VMware host solutions there are on MacOS X
- And, of course, there's always the issue of Twitter data delivery reliability
- Side benefit is that I can plot the scalability function for Twitter ;-)

Do The Math

- Handbook of Latent Semantic Analysis `http://bit.ly/1s9h0B`
- Understanding Complex Datasets: Data Mining with Matrix Decompositions `http://bit.ly/8kmCd`
- Twitter API: Up and Running `http://bit.ly/ls2na`