RSS feed aggregator

Ajinkya Kale and Raghuveer Sagar

Introduction:

Today’s internet has various websites that provide news, sports articles, blog entries and tech updates. From the perspective of the user it becomes tedious to keep track of all the websites. Even if that can be done (like bookmarks) but the user needs to keep updating the websites to check for new stories or entries. Hence RSS which stands for Rich Site Summary, was developed So RSS turns this around and pushes the ‘responsibility’ away from the user’s requires websites to keep an updated ‘feed’ stream in a predefined structure. The user can use software’s called ‘RSS Aggregators’ that do the job of getting the ‘feed data’ and presenting it to the user. So user does not need to check the websites by himself.RSS Aggregators are capable of getting feeds from any number of websites and user can access it at his leisure.

There are a lot of aggregators available on web for different platforms. In this paper we describe an aggregator written using bash ,awk and Perl. The scripts read the feeds and generate an html file that refreshes every 5 mins to display any updated content. We begin by describing the format of the feeds, and explain how we extract content from feed files and format it for the final display. The paper also explains about the algorithms used to process the data with the help of few code snippets.

Problem/Background:

The internet is growing enormously and so does the number of websites.RSS is the one such application which retrieves the latest arrived content from the website that user has ‘subscribed’ to. It then displays it to user in a readable form. There are two parts in the process of RSS technology.

1. The content provider needs to update the feeds periodically, as soon as he has content to share.
2. Client needs aggregator which gathers all feeds without user’s intervention and then displays it to user when he needs it.

First we will explain the first part. Any websites that wish to support RSS for the benefit of the users have to maintain an XML file called ‘feed’. Normally websites maintain many feeds, for different types of content. For example, a news website like http://www.reuters.com has different feeds for Arts,Sports,Business,Lifestyle etc.The XMLs follow the format or structure is defined by the RSS version it is supporting.

Below we give an example format of RSS XML version 2.0

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
 <title>RIT Computer Science</title>
 <link>http://www.cs.rit.edu</link>
 <description>RIT Computer Science Dept News</description>
 <item>
          <title>Graduate Scholarships</title>
          <link>http://www.cs.rit.edu/scholarships</link>
          <description>RIT Computer Science Dept announces Graduate Scholarships</description>
 </item>
 <item>
          <title>Finals Dates</title>
          <link>http://www.cs.rit.edu/finals</link>
          <description>RIT Computer Science Announces Finals Dates</description>
 </item>

<item>
          <title>Enrolment Begins for Spring</title>
          <link>http://www.cs.rit.edu/enrol</link>
          <description>SIS is open for course enrolment from today</description>
 </item>

</channel>
</rss>
```

The XML does not contain all the tags. There are also additional tags like <pubdate>,<guid>.
Below we briefly explain the purpose of each tag.
<channel> ,contains the name of the distributing channel. In the above example it is RIT Computer Science.
<link>, the url for the website.
<description>, a brief explanation on the channel’s content.
<item>, this tag contains the content. There is always at least one item block. This tag has other tags for specifically identifying the content data.
<title>, the title for the article or news story or blog entry.
<link>,the url for where the full story is published.
<description> , a summary on the story or a description ,

Nowadays most news,sports,tech websites support websites. One specific example will be later explained, but below we provide few urls where we can find feed XMLs on web.

feed1:http://www.htmlbasictutor.ca/feed.xml

feed2:http://www.espncricinfo.com/rss/content/story/feeds/0.rss

feed3:http://rss.cnn.com/rss/cnn_tech.rss

feed4:http://rss.sciam.com/ScientificAmerican-Global?format=xml

Now we shall describe a little bit about the aggregator. It is also known as ‘RSS feed aggregator’ or ‘feed reader’ or ‘RSS reader’. The primary purpose of aggregator is to aggregate all the feeds the user has added and display in a viewable format. Whenever a user wishes to follow a particular feed, he adds it to the aggregator. This process is called ‘subscribing to a feed’. Of course today aggregators are available in many formats and on many platforms. Web based aggregators are popular. Browsers like chrome and Mozilla provide plugins and extensions which act as aggregators. Since this process involves a lot of text processing and formatting, we decided to implement the same with Perl and awk.

Procedure-

To implement this project, a suite of scripts are written in bash, Perl and awk. The task was distributed into two modules,

1.**XML parser** contains awk script parseFeed.awk
2.**HTML generator** contains perl scripts html_writer.pl,tags_generator.pl,helpers.pl.

The aggregation process is conducted by a bash script, rss_aggregator.sh.
Another important file is ‘feeds’.The user ‘subscribes’ to a feed by adding that feed url to this file.Once the feeds file modified the user can go ahead and run the aggregator or schedule it to run using Cron every 15 mins.

XML parser

This module downloads the feeds XMLs from the urls in the feeds file.The files are downloaded using *wget* command.XML parser works on that XML file to generate a intermediary ‘Parsed’ file.The parser then parsers this XML ,and extracts the information from the tags and creates a ‘Parsed’ file which will serve as data for HTML generator.

Following is snippet of the feed XML of a website espncricinfo.com

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0"
xmlns:atom="http://www.w3.org/2005/Atom"  xmlns:media="http://search.yahoo.com/mrss/">
<channel>
<title>Cricket news from ESPN Cricinfo.com</title>
<ttl>2</ttl>
<link>http://www.espncricinfo.com</link>
<description>Visit Cricinfo.com for up-to-the-minute cricket news, breaking cricket news, live
cricket commentary, ball-by-ball commentary, cricket video, cricket audio and
features.</description>
<copyright>(c)Cricinfo</copyright>
<language>en-gb</language>
<image>
<title>Cricket news from ESPN Cricinfo.com</title>
<url>http://i.imgci.com/espncricinfo/espnci.png</url>
<link>http://www.espncricinfo.com/</link>
</image>
<item>
```

```
<title>Two days until Clarke fitness decision</title>
<description>Michael Clarke has until Wednesday to resume running after his hamstring injury
or be ruled out of contention for the first Test against India</description>

<link>http://www.espncricinfo.com/australia-v-india-2014-
15/content/story/802727.html?CMP=OTC-RSS</link>

<guid>http://www.espncricinfo.com/australia-v-india-2014-
15/content/story/802727.html</guid>
<pubDate>Mon, 24 Nov 2014 00:46:12 GMT</pubDate>
</item>
<item>
<title>De Villiers blames over-thinking as plans go awry</title>
<description>Given the intricacy of South Africa's planning for next year's World Cup, it is
difficult to believe it has all come down to this: a 4-1 defeat to the opposition they believed they
needed to beat to consider themselves ready</description>

<link>http://www.espncricinfo.com/australia-v-south-africa-2014-
15/content/story/802667.html?CMP=OTC-RSS</link>

<guid>http://www.espncricinfo.com/australia-v-south-africa-2014-
15/content/story/802667.html</guid>
<pubDate>Mon, 24 Nov 2014 02:38:51 GMT</pubDate>
```

XML acts on the above XML producing a ‘Parsed’ file.The format of that file is as below

parseFeed.awk

The awk script receives the XML file as an argument from the bash file, rss aggregator.sh. the script parses the xml file based on the tags described above. It uses regular expression for the extraction of the text data and stores it into a temporary file. The file extracts only following tags which are in the range of the <item> to <\/item> that are:
<title>, <link>, <guid> and <pubdate>

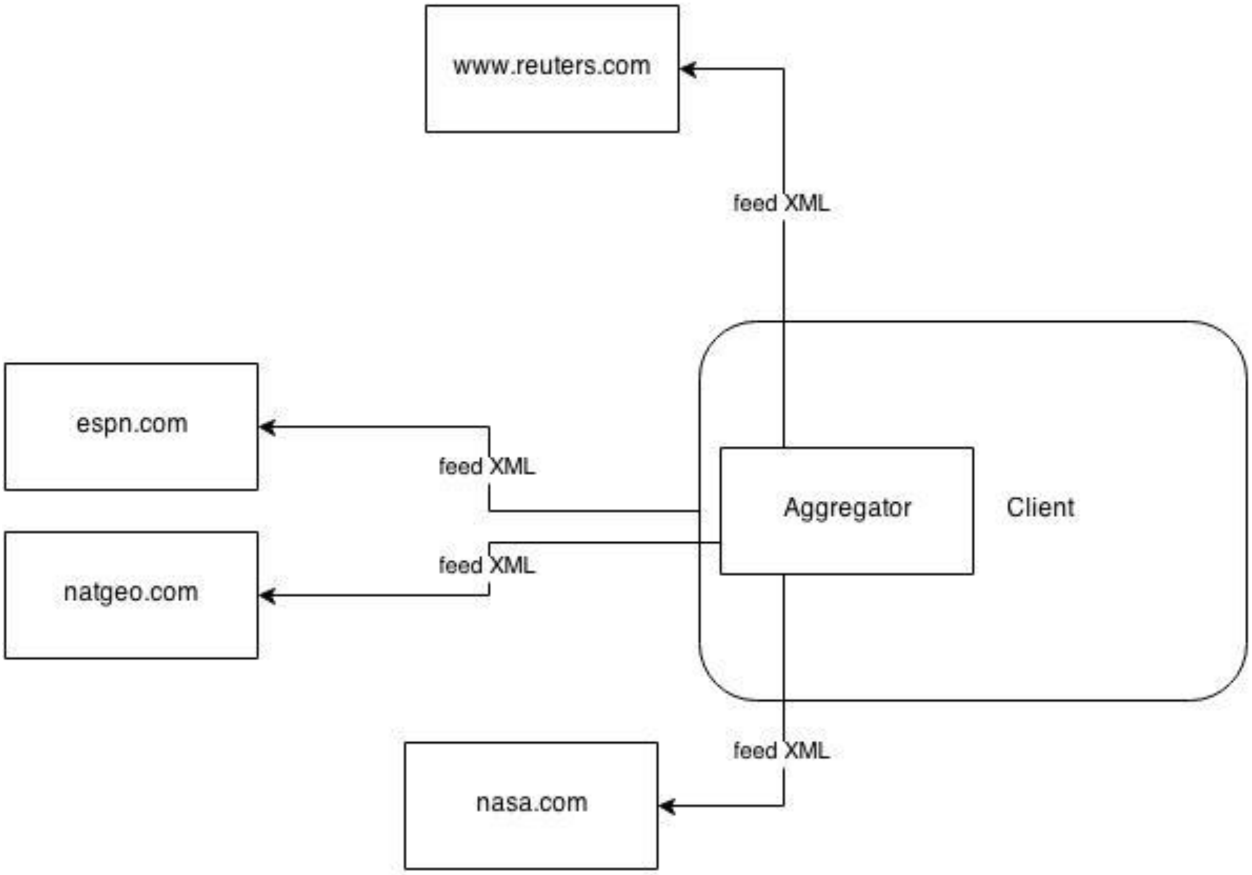
```
channel:Cricket news from ESPN Cricinfo.com
mainlink:http://www.espncricinfo.com
headline:Two days until Clarke fitness decision
link:http://www.espncricinfo.com/australia-v-india-2014-
15/content/story/802727.html?CMP=OTC-RSS
guid:http://www.espncricinfo.com/australia-v-india-2014-15/content/story/802727.html
pubdate:Mon, 24 Nov 2014 00:46:12 GMT
headline:De Villiers blames over-thinking as plans go awry
link:http://www.espncricinfo.com/australia-v-south-africa-2014-
15/content/story/802667.html?CMP=OTC-RSS
guid:http://www.espncricinfo.com/australia-v-south-africa-2014-15/content/story/802667.html
pubdate:Mon, 24 Nov 2014 02:38:51 GMT
```

HTML generator:
This module's primary goal is to display the data in 'Parsed' files and generate an HTML document that can be used to display the formatted output to the user in a browser.
The HTML file will always remain open in the browser and it automatically refreshes every 5 mins.
Each headline is a link which takes the user to the published story in the respective websites.

Conclusion:
This project has been successfully completed and has been tested. To run this application bash scripts needs to executed and the Rss xml link has to be entered in the input file. The output will be displayed on browser webpage. The sample output upon the execution is attached with submitted tar file.

Future works:
This project has its various applications. we can further incorporate this into an browser plugin or a web based application with appropriate User Interface. In future version of the application, the notification feed will be displayed along with the pictures. This application can be extended to various paradigms like poll aggregator, social network aggregator, data aggregator etc.

Following is the diagram of the project.


www.reuters.com
feed XML
espn.com
feed XML
Aggregator
Client
feed XML
natgeo.com
feed XML
nasa.com