A SCALABLE SEARCH ENGINE AGGREGATOR

Advisor: Dr. Chris Pollett
Committee Members : Dr. Sami Khuri and Dr. Robert Chun

By
Pooja Mishra
Agenda

• Yioop
• About Project
• Preliminary Work Summary
• News Updater
• Web Interface for Manage Machines
• Video Updater
• Mail Distribution
• Experiments
• Conclusion
• Demo
Yioop

• Yioop, an open source PHP search engine based on GPLv3 license, is designed and developed by Dr. Chris Pollett

• It allows user to index a website or a collection of websites

• It is designed to work on PC, smartphone and tablet
Yioop MVC

Yioop is a web application designed using its own Model View Controller framework
About Project

Goal of the project is to primarily enhance some of the media updater features and enhance some of them. It includes –

• News Aggregation
  • Web Interface to Include Media Updater
• Video Updater Feature
• Mail Distribution
# Comparison of News Feed Feature

<table>
<thead>
<tr>
<th></th>
<th>Based on Categories</th>
<th>Customized for user</th>
<th>Basis for generating customized news feed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Google news</strong></td>
<td>YES</td>
<td>YES</td>
<td>Google search queries and articles visited</td>
</tr>
<tr>
<td><strong>Yahoo news</strong></td>
<td>YES</td>
<td>NO</td>
<td>Trending stories</td>
</tr>
<tr>
<td><strong>Facebook news</strong></td>
<td>YES</td>
<td>YES</td>
<td>Connections and likes, post views etc.</td>
</tr>
</tbody>
</table>

Facebook – influenced by connections  
Google/Yahoo – Follow their own algorithms
News Updater Scalability

- Many web sites allow users to create a personalized feed
- Associate feeds with consumers and event streams with producers
- Selectively materializing each consumer’s feed: events
- Minimize global cost by making local decisions
- Hybrid strategy results in the lowest system load (and hence improves scalability) under a variety of workloads
News Updater in Yioop

News updater process that can be used to automatically update news feeds from various news sources.

News feeds can either be RSS feeds, or can be scraped from an HTML page using XPath queries and re-indexed.

Improve the quality of the search results.
Existing News Updater Feature

The single machine/name server periodically (once an hour as per current settings in Yioop) fetches the news feeds from the news sources added into the database, then will update the database and rebuild the index shard.
Initial Experimentation Results

Configuration used for testing –
Intel core i5 with CPU @1.60 GHz , 6GB RAM and windows platform.
Proposed Distributed News Updater Feature

Distribution of news sources - Hashing mechanism
News Updater in Distributed Setup

Now, based on the type of Yioop instance a request is made to name server to get news sources and news feeds from respective sources are fetched.
Manage Machines Activity in Yioop

Initially name server was by default running only on name server.
Manage Machines Activity Now..

- Admin [Manage Machines]

Add Machine:
- Machine Name:
- Machine URL:
- Is Mirror:
- Has Queue Server:
- Number of Fetchers: [0] Submit

Machine Information:
- Media Updater Mode: [Name Server] [Distributed]
  - Name Server
    - Media Updater [Log] [On] [Off]
    - Row 0 to 1 of 1 Show [50] [Delete]

media_slave
http://localhost/yloop_media slave/

Machine has no queue server

Status of Media Updater only on Name Server is visible

Start
Log in to Yloop
Click on manage machines under system settings
Distributed
Yes
Status of Media Updater on client machines and NS is visible
No
Status of Media Updater only on Name Server is visible
Stop
Background Work for Video Updater

- Purpose
- Video Formats Supported by Popular Browsers

<table>
<thead>
<tr>
<th>Browsers</th>
<th>webm</th>
<th>Mp4</th>
<th>mov</th>
<th>avi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Safari</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Firefox</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Internet explorer</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Some of the very popular browsers all support both webm and mp4 formats.
Basics of Audio/Video

• Huge amounts of storage space required
• Traditional, lossless compression algorithms
• Need of elaboration of new algorithms with lossy compression
• Lossy compression
  • Compression that is far more efficient but with a trade-off in that the picture and sound quality
• Codecs - Algorithms that allow us to encode the data in order to transport it and to decode the data the other end
Container

• Encapsulated encoded audio and video files into a single file, packaged, transported, and presented. (AVI, WAV files)
• Informs the media player about the audio and video codecs used

• Playback of a Multimedia File
What is FFmpeg?

• A free software project that produces libraries and programs for handling multimedia data.

• An application that allows Linux users to convert video files easily

• Implements a decoder and then an encoder enabling the user to convert files from one container/codec combination to another
How conversion takes place?

1. The original container is examined and identified.
2. The encoded data is extracted and fed through the codecs.
3. The newly-decoded data is then fed through the "target" codecs.
4. The new container is created.

Specific examples:
- QuickTime file
- SVQ3 video
- MP3 audio
- H263 video
- AMR wideband audio
- 3GP file
FFmpeg Commands

```bash
ffmpeg [global_options] {[input_file_options] -i input_file} ...
{[output_file_options] output_file} ...
```

- FFmpeg Convert

```bash
ffmpeg -i sample.mov -vcodec h264 -acodec aac -preset veryfast -crf 28 -strict -2 sample.mp4
```

Mainly used for manipulating videos – split, merge etc.
Server converting video as a whole

Pros
• Easy to implement.
• Less bookkeeping
• No loss of data

Cons
• Will take longer in terms of time.
• Name server’s other functions will also be interrupted and delayed during the media conversion.
Distributed setup converting split video as a whole

Pros

• Easy to implement
• Less bookkeeping
• Greater ease of downloading and uploading the video files for conversion.

Cons

• Videos are assigned in the order that they are assigned for conversion.
• Longer video files assigned to a media updater will take longer even if they were assigned first.
Distributed setup converting split video

Pros
- Time reduction for the video conversion.
- Easy to upload and download split video files to and from name server.

Cons
- Difficult to implement
- Lots of bookkeeping on the server side.
Choosing an Approach

We chose the Distributed setup converting split video approach.
### Name Server Media Updater Process

- Handles the requests made by slave Yioop instances
- Performs book-keeping

<table>
<thead>
<tr>
<th>Function task</th>
<th>Look for</th>
<th>Generate</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media split</td>
<td>Split.txt</td>
<td></td>
<td>Split.txt</td>
</tr>
<tr>
<td>Move folders to converted</td>
<td>Count.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate ready to assemble file</td>
<td>Concatenated.txt</td>
<td>Ready to assemble.txt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ready to assemble.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Media merge</td>
<td>Concatenated.txt</td>
<td>Concatenated.txt</td>
<td>All text files</td>
</tr>
</tbody>
</table>
Media Updater Slave Process

- Media updater slave performs only the task of converting videos from one form (mov/avi) to another (mp4).
Mail Distribution

• In Yioop, when a user starts a new thread or comments to an existing thread, several people will be notified.

• Can cause congestion if too many emails are to be sent.

• Currently, emails are sent using two ways –
  • Using PHP mail() function
  • SMTP server configuration

• These emails can be aggregated over a period of time and sent periodically.
Mail Distribution (2)

- Emails to be sent are aggregated in a text file every 5 minutes
- Locking mechanism
- Media updater process looks for mailer text files and sends out emails
- Email task offloaded from mail server (Yioop Webapp) to media updater process
Experiments

• Unit Testing

• Created a cluster on AWS of a few machines.

• The configuration for machines -
  ▪ 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory
  ▪ Platform : Ubuntu

• Set up Yioop on one such instance and then created clones of this instance using the AMI option
News Updater Performance Testing

News Updater (2 instances)

<table>
<thead>
<tr>
<th>News Sources</th>
<th>Time taken (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td>200</td>
<td>85</td>
</tr>
<tr>
<td>350</td>
<td>80</td>
</tr>
<tr>
<td>400</td>
<td>90</td>
</tr>
<tr>
<td>500</td>
<td>95</td>
</tr>
<tr>
<td>680</td>
<td>90</td>
</tr>
<tr>
<td>700</td>
<td>95</td>
</tr>
</tbody>
</table>
News Updater Performance Testing(2)

The graph above shows the time taken in seconds for different news sources. The x-axis represents the number of news sources, while the y-axis shows the time taken in seconds. The graph compares different scenarios:

- **3 instances** (blue line)
- **2 instances** (orange line)
- **single** (gray line)

As the number of news sources increases, the time taken also increases. The graph illustrates that having more instances generally reduces the time taken compared to a single instance.
Video Updater Testing

Upload Progress: 1%

test_videoUpdater [Feed|Wiki]

Start New Thread

Subject
video updater testing

[Thu, 14 May 2015 23:06:42 -0700] Video updates done...
[Thu, 14 May 2015 23:06:42 -0700] Concatenating videos...
[Thu, 14 May 2015 23:06:42 -0700] Inside generateAssembleVideoFile function...

[Thu, 14 May 2015 23:08:42 -0700] Moving video folders from media_convert to converted...
[Thu, 14 May 2015 23:08:42 -0700] Looking for video files to split...
[Thu, 14 May 2015 23:08:42 -0700] Checking for video files in process...
[Thu, 14 May 2015 23:08:42 -0700] Checking for News Updates...
[Thu, 14 May 2015 23:08:42 -0700] Setting media mode to distributed
[Thu, 14 May 2015 23:08:42 -0700] Checking Name Server for Media Updater properties...

Ensure minimum loop time by sleeping...10
Video Updater Testing(2)


[Thu, 14 May 2015 23:47:59 -0700] Checking for video files to process...


[Thu, 14 May 2015 23:08:43 -0700] Inside generateAssembleVideoFile function...

[Thu, 14 May 2015 23:08:43 -0700] Moving video folders from media_convert to converted...
[Thu, 14 May 2015 23:08:43 -0700] Looking for video files to split...
[Thu, 14 May 2015 23:08:43 -0700] Checking for video files to process...
[Thu, 14 May 2015 23:08:43 -0700] Checking for News Updates...

[Thu, 14 May 2015 23:49:07 -0700] Video updates done...
[Thu, 14 May 2015 23:49:07 -0700] Concatenating videos...

[Thu, 14 May 2015 23:49:07 -0700] Inside generateAssembleVideoFile function...
[Thu, 14 May 2015 23:49:07 -0700] Moving video folders from media_convert to converted...
[Thu, 14 May 2015 23:49:07 -0700] Looking for video files to split...
[Thu, 14 May 2015 23:49:07 -0700] Checking for video files to process...

[Thu, 14 May 2015 23:49:07 -0700] Video updates done...
[Thu, 14 May 2015 23:48:57 -0700] Concatenating videos...
[Thu, 14 May 2015 23:48:57 -0700] Inside generateAssembleVideoFile function...
# Video Updater Performance Testing

<table>
<thead>
<tr>
<th>Length of videos</th>
<th>No of machines</th>
<th>Time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 min</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>7-8 mins</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>14-15 mins</td>
<td>1</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>15-20</td>
</tr>
<tr>
<td>24 mins</td>
<td>1</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>50 mins(500 Mb)</td>
<td>1</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>180</td>
</tr>
</tbody>
</table>
## Conclusion

<table>
<thead>
<tr>
<th>Feature /Modification</th>
<th>What we have done</th>
<th>How is it impacting Yioop</th>
</tr>
</thead>
</table>
| **News updater**      | • Design for distributed setup  
                         • Code for distributing the pre-existing news updater feature | • Scaled the news updater feature  
                         • Improved performance to fetch news feeds |
| **Manage machines web interface** | • Mockup for UI to add media updaters to existing web interface  
                                          • Code for adding media updater to Yioop instances | • Switch between name server and distributed setup  
                                           • Control the media updater independently for each machine |
| **Video updater feature** | • Design solution  
                           • Code for incorporating video updater in Yioop | • Convert videos and upload back for viewing purposes.(web-friendly)  
                                           • Distributed setup giving improved performance |
| **Mail Distribution** | • Design for mail aggregation  
                           • Code for mail distribution(first patch) | • Media updater will send out emails  
                                           • Prevention of denial of Service attack |
Future Scope

• More video formats can be added in video updater feature to cover most of the used video formats

• Building a framework for such aggregation jobs

• Crash recovery
Questions ?
Thank You!