

Roland Krystian Alberciak  
CS 4440 Project Proposal

**Project Title:**

A scalable platform for content monitoring, processing and P2P 'push' distribution of licensed content: Case Study with Microsoft PowerPoint and Adobe PDF files

Roland Krystian Alberciak  
{krystian}@gatech.edu

**Project Summary:**

We wish to develop a system that acts as a virtual secretary: taking care of webpage content monitoring, processing and delivery for users who wish to be disconnected from the need to keep up with visiting webpages to acquire content published to the web which they may/will need to have in the next 24 hours or 20 minutes.

**Motivation:**

This project has been motivated to address a particular need: When academic professionals like professors and speakers post lecture notes content in the form of PDFs and Powerpoint files for peers to access, we observe that this content is posted at a variety of times. Sometimes content is posted well in advance so peers can comfortably download content, print and attend a talk with it.

However, we note some content is published to the internet in a time window that does not comfortably permit doing downloading and printing it. Some lecture note content is posted from two hours to immediately before class, however we would like students who wish to have such content for class to acquire such content and have it ready before class.

We also would like to remove students from the process of downloading and manually printing said content. We want to create a streamlined system which captures such files, processes them for printing with particular printing preferences, and delivers them to students so that they can obtain said content ahead of class for use in class.

**Objectives:**

[1] Produce an architecture and system which records content changes and distributes changes to users.

[2] Reduce the dependency for students to check course websites before class in order to print powerpoint slides [reducing 5 minutes every day \* ~ 70 school days = 350 minutes, or 6 hours of saved time in a semester. **This is a very very conservative estimate and actual time saved by our system may be 2 or 3 times this amount.** ]

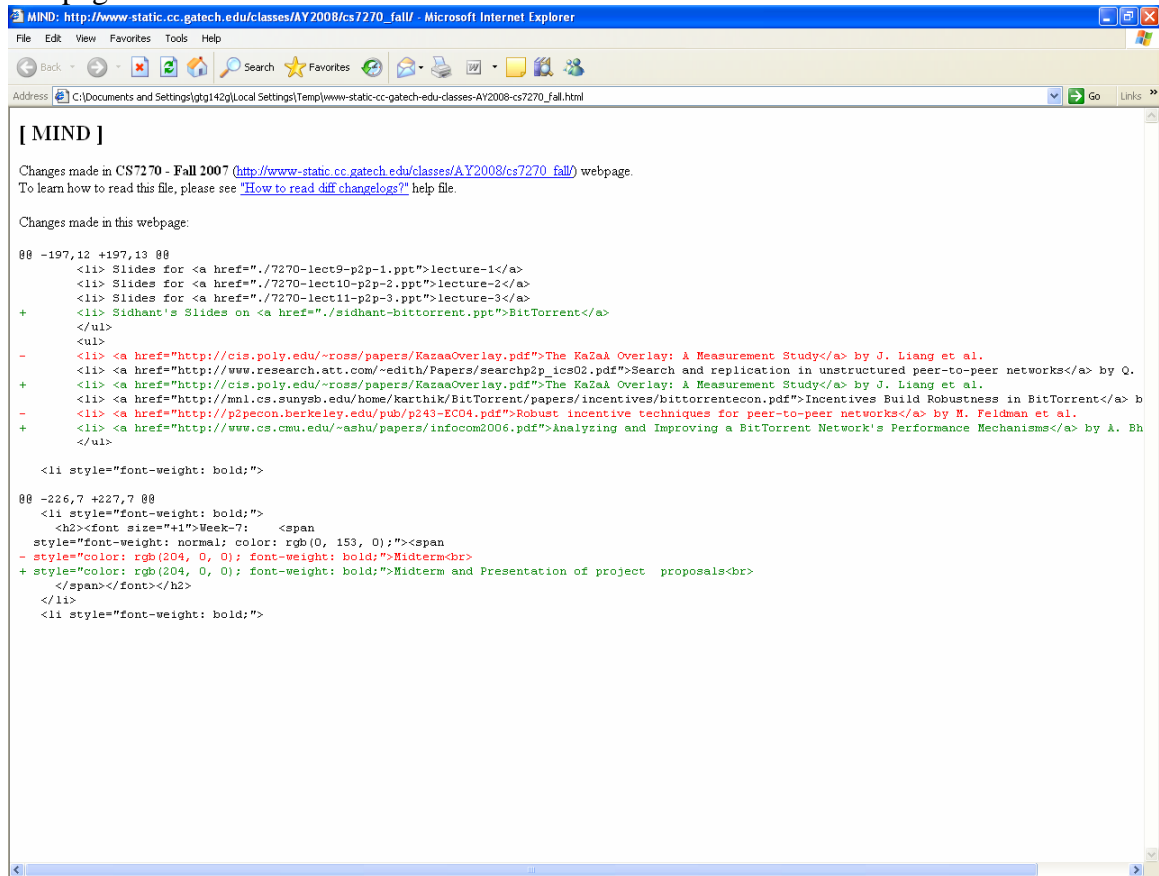
[3] Provide location aware printing distribution of files

## Methodology:

We build on existing platforms in order to create our streamlining process.

[1] Web content monitoring will be either handled by the WebCQ platform developed at Georgia Tech, or the “MIND: Intelligent Webpage Monitor” project developed and posted on sourceforge.net.

The MIND content monitoring system provides a summary of what has changed in the webpage via diff:



```
[ MIND ]
Changes made in CS7270 - Fall 2007 (http://www-static.cc.gatech.edu/classes/AY2008/cs7270\_fall/) webpage.
To learn how to read this file, please see "How to read diff changelogs?" help file.

Changes made in this webpage:
00 -197,12 +197,13 00
<li> Slides for <a href="/7270-lect9-p2p-1.ppt">lecture-1</a>
<li> Slides for <a href="/7270-lect10-p2p-2.ppt">lecture-2</a>
<li> Slides for <a href="/7270-lect11-p2p-3.ppt">lecture-3</a>
+ <li> Sidhant's Slides on <a href="/sidhant-bittorrent.ppt">BitTorrent</a>
</ul>
<ul>
- <li> <a href="http://cis.poly.edu/~ross/papers/KazaaOverlay.pdf">The Kazaa Overlay: A Measurement Study</a> by J. Liang et al.
<li> <a href="http://www.research.att.com/~edith/Papers/searchp2p_ics02.pdf">Search and replication in unstructured peer-to-peer networks</a> by Q.
+ <li> <a href="http://cis.poly.edu/~ross/papers/KazaaOverlay.pdf">The Kazaa Overlay: A Measurement Study</a> by J. Liang et al.
<li> <a href="http://mml.cs.sunysb.edu/home/karthik/BitTorrent/papers/incentives/bittorrentecon.pdf">Incentives Build Robustness in BitTorrent</a> b
- <li> <a href="http://p2pcon.berkeley.edu/pub/p243-EC04.pdf">Robust incentive techniques for peer-to-peer networks</a> by M. Feldman et al.
+ <li> <a href="http://www.cs.cmu.edu/~ashu/papers/infocom2006.pdf">Analyzing and Improving a BitTorrent Network's Performance Mechanisms</a> by A. Bh
</ul>

<li style="font-weight: bold;">
00 -226,7 +227,7 00
<li style="font-weight: bold;">
<h2><font size="+1">Week-7: <span
style="font-weight: normal; color: rgb(0, 153, 0);"><span
- style="color: rgb(204, 0, 0); font-weight: bold;">Midterm<br>
+ style="color: rgb(204, 0, 0); font-weight: bold;">Midterm and Presentation of project proposals<br>
</span></font></h2>
</li>
<li style="font-weight: bold;">
```

[2] The web content summary will then be processed to interpret changes:

-- What new URI's have been added

-- What URI's have been deleted

Therefore a list of URI's will be generated which need action upon them.

[3] The new URI's [which are links to powerpoint, pdf files] will each then be placed into a new monitoring system.

The content monitoring system will already be populated with rules which require it to monitor previously identified binary files [pdf, ppt]. In addition to these files which are already being monitored, new binary files will be added too.

The MIND monitoring system is optimized to do a diff of html/php web browser accessible content, however polling and comparing binary file types like powerpoint and pdf by diff is rather inefficient and processor intensive.

We seek to do monitoring of these content types by recognizing that content has changed through using several test criteria (size, date, or md5sum).

We also must be intelligent about the content we monitor. We notice that content that is 'very old' may not have to be checked as frequently or can stop being monitored once it's deadline threshold passes. Also if we try to access content and receive a 404 missing webpage error or content unavailable error, we need to retry accessing the content.

[4] If content has changed, distribute content to processing system.

In this section we detail a novel approach to processing content which requires licensing to view. Adobe PDF and Microsoft Office PowerPoint files require a licensed copy of the software in order to view and process.

We want to develop a system that prints such content natively in Windows. We would somehow like to use PowerPoint to print ppt files instead of opting for open source tools, simply because there is more compatibility assured when viewing the content in its native program. The same may be true for PDF files as they can be processed and viewed in windows.

We note that to print these file types requires us to render them. There are currently no approaches or techniques which permit printing such files through command line scripting. Instead, we must view and render the files, and then print them.

Furthermore, users may wish for us to do some inference on these files. If the file is a powerpoint, we should print the file as a handout with 3 or 4 slides a page. If the file is a PDF we will need to determine if it is a previous powerpoint slideshow saved as PDF or if it is a regular PDF document, and to print it accordingly.

In all cases, we will be soft 'printing' the file, rendering the file from a proprietary file format to postscript, which can be read by any printer.

Therefore, content which needs to be printed will be distributed to computers with windows who agree to cooperate with our research study. We wish to avoid having these computers operate as webservers, and instead we will upload content to these computers via key exchange enabled scp.

We will employ the sponly platform on cygwin to have cooperating computers wait for files, receive files in 'drop file' directories, process content, and distribute it to the next step.

[5] Once content is printed to postscript, push to printer.

This is where another novelty of our approach happens.

We want to send files to printers that are near the user who has requested us to print these files, or, if unable to locate user, print to a printing service [oit print service for GaTech students].

We can either do:

1. Location detection to infer where the user is and print content near them and send them a text message once it arrives in the printer near them.
2. Location inference: infer where a user may be located given their particular schedule, and print to printers near that location.
3. Hybrid approach: Attempt to communicate to user on cell phone and ask them where they want it printed. If they are planning on being at the scheduled location, we can print there. If they are not, we can print to near where they are now. If there is no response from the user, we can either print to central printing service as a fall back method.

If the user who requested us to monitor the webpage and print the content has their cell phone on them, we can try to query their cell phone and get the following information:

1. If phone is GPS enabled, what are the current coordinates of the user?
2. If phone is WIFI enabled, what are some WAPs the user's phone can identify?

We then try to distribute our file to these printers.

There have been some security concerns over 'print sharing', but we note we can either communicate with printers that have 'print sharing' enabled, or we can communicate to users who have access to printers through their desktop and have agreed to participate in our service.

This could be done by uploading a file or pointing to a file we upload for their PC's to capture by sending instructions to them over ssh or by having their computers act as web servers and 'requesting a webpage' with information about content. For example:  
<http://your-participating-pc-ip/file-to-print-uploaded-location.ps.html>

Print requests would be routed through their computers, which will act on behalf of the original user to print content to the printers they are connected to. They could then send a reply to let our service know the printing was successful or not [via checking the printing] or that the printer reported no errors in processing our directive.

## References:

1. There are some commercial services which offer file printing to travelers at 'participating printers'. If a traveler is a hotel, he uploads his files to a central website and lists which hotel he is at, which then routes the files to that hotels printer. <http://www.printeron.net/solutions/services/portals.html>

We attempt to do something a bit more interesting, by providing content monitoring processing and distribution. We want this to be a peer2peer service, so having 'timeshares' where users can publish how much they pay for printing a each page [could be real money or some 'credits' system], and having many peers who offer to print a file 'auction' how much they charge, and have the print request routed to nearest reasonable cost printer.

That way if Alice wants to print files for her class, she can pick up printouts from Bob's printer. Bob could then later call upon this printing 'favor' and print to sally's computer next time he needs a printout sent to that location.

## Timeline:

October 15: Have basic service operational [monitor webpage, process uri's, process pdf or ppt, and print to oit's printer]

November 15: Implementing peer2peer print sharing status check

December 5: final turning and project presentation