

Taming the Skyrocketing Printing Costs in Campus Computing Labs With Open Source Technology

Greg Pepper

University of Wisconsin-Whitewater
800 W. Main St
Whitewater, WI 53190
262.472.1014

pepperg@uww.edu

Lorna Wong

University of Wisconsin-Whitewater
800 W. Main St.
Whitewater, WI 53190
262.472.5017

wongl@uww.edu

ABSTRACT

Printing expenses in college computing labs have been escalating along with the use of technology in instruction. Commercially available print management software requires a huge initial financial investment with uncertain cost recovery. The outcome usually involves shifting most of the print management software expense to students. Find out how the University of Wisconsin-Whitewater controls their printing costs using open source technology without adding a financial burden to students.

Categories and Subject Descriptors

H.4.0 [Information System Applications]: General

General Terms

Management, Measurement, Design, Economics

Keywords

Print management software, open source, computer labs, Novell, web application, Apache Foundation, Jakarta

1. PROBLEM

Skyrocketing printing costs in college computer labs has become one of the top issues Lab Managers have had to deal with in the last few years. Use of technology in instruction including email attachments, library electronic reserve and course management systems have helped instructors communicate better with students, as well as make their materials more easily accessible. At the same time, these electronic formats of course material also directly translate to increases in printing from anywhere. Traditionally, faculty were allocated a copy budget from their department to photocopy handouts for class. Now all of these materials are put online to make them more accessible to the students. In extreme cases, classroom textbooks were replaced entirely with online articles and essays. On a traditional undergraduate resident campus, the computer labs cater to most of the printing needs of students. While the printing cost gets transferred to these labs, the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGUCCS'04, October 10–13, 2004, Baltimore, Maryland, USA.

Copyright 2004 ACM 1-58113-869-5/04/0010...\$5.00.

departmental copy budget does not. This causes a large increase in printing expenses for lab managers without any extra financial backing. Many universities have resorted to transferring the costs to the students by charging for printing. In most cases, not only are the students paying for the cost of the toner, paper and printer maintenance but they are also paying for the very system that is charging them for printing. While this is a viable solution for some universities, it does not work for others.

1.1 UW-Whitewater Dilemma

At the University of Wisconsin-Whitewater, students already pay a technology fee as part of their tuition. A fixed allocation from this fee is budgeted to support the printing costs in the General Access Labs. However, such allocation does not keep pace with the rapidly rising printing costs. Additional charges to students will not be a popular solution. To even consider this option, we would also have to purchase print management software which involves a very expensive up-front investment with a yearly maintenance cost. How can we control printing and gather more data about printing patterns without committing to expensive print management software?

University of Wisconsin-Whitewater serves 9,700 undergraduate students in the General Access Labs and departmental computer labs. The two General Access Labs are run by Instructional Technology Services which is a division of the central IT department. Other computer labs have distributed ownership among various departments while Instructional Technology Services provides technical support for these labs. The budgets for printing in these labs come from the department's technology budget. These labs do not receive any funding from the student's technology fee for printing purposes. This decentralized ownership of labs meant no matter what we did to control printing all lab owners would have to be involved and come to a consensus on a solution. Thus, for any printing containment solution to work, we need the buy-in from all campus computer labs. Without a consistent system across campus, students will migrate to the labs that are the least restrictive to them. This will cause labs that are not participating in the system to take on more printing. While it is ultimately their decision, we wanted a system that contains printing without passing the cost on to someone else.

1.2 Collecting Data to Help Solve the Problem

We have been keeping track of the printing activities in our 350-stations General Access Computing Labs for the past 8 years. The alarming increase in page counts and printing costs urges us to actively look for a solution to help control, if not reduce, the printing costs. We decided that whatever decision we make, it will

have to be a solution that is feasible across all campus labs. We evaluated a number of different commercial products and found that all the products that met our needs were very expensive solutions and that we do not have enough data to justify making such an investment. We realized we need to analyze the printing activities in our labs, such as:

- Who is doing all the printing?
- Is the average number of pages per student evenly distributed?
- What are they printing?
- Is the printing for educational purposes?
- Do they need everything they are printing?

The answers to these questions would help us take the next step should we decide to invest in print management software. We would then be able to make decisions on how much to charge per page and how many free pages students should receive courtesy of their technology fee. These numbers would help determine exactly what the return on investment would be on commercial print management software.

The catch was we need print management software to obtain these numbers. Some institutions will implement the software for a year but not charge their students so that they can develop policies based on the numbers the print management software generates. During this trial year, many universities have seen a decrease of at least 10 percent just by installing the software. If our departmental lab owners decided that 10 percent was enough of a saving, and they didn't want us to charge the students, then we would have a very low return on investment. Installing this software and maintaining it would take a significant amount of time for our staff, so if the return on investment was too low we would just end up costing the University money in the long run. We decided the most economical solution was to deploy our staff expertise to extract such information using our own homegrown software. While this approach might take more staff hours to develop the software, install, and maintain it, we still found this approach to be more economical than the initial cost of most commercial print management software packages.

2. SOLUTION

2.1 A Home Grown Novell Solution

In the spring of 2003, we took the first step in developing our own print management software. Our campus file and print services are provided using the network operating system Novell Netware, which is a platform that packages and supports open source applications with their operating system. The Novell Distributed Printing Service (NDPS) module provides print job auditing service. This kind of functionality allows us to develop our application without having to worry about the technical details of how to interpret PCL and Postscript printer code. Novell also provides a Java API that allows developers to access the print engines data and control its actions. The Java API is written at a very high level and makes it very easy to do low level tasks with the print engine of the operating system. The auditing and Java API were exactly what we needed to develop this software quickly without expending too many staff hours. All we had to do was develop an interface for users to release their print jobs and then store all the data associated with those print jobs in a data base.

How the users would use this application and what kind of machine it needs to run on became our main focus.

2.2. What is Under the Hood?

To implement print release software a number of machines are typically dedicated to be print release stations. When a user wants to print, they walk up to one of these machines, login, find their job, and select it to be printed (See Figure 1). Some commercial solutions required these release stations to be touch screen kiosks which are very costly. We wanted our application to be as cost effective as possible. We decided that our application would use a thin client web architecture so we could run this software on our older lab workstations that were going to surplus. Thus, we developed our application using JSP, Java's server side web application language.

Printer Queue		Printer	Job Limit	Page Limit	User
RED_MGgalabs.uwv		RED_MGgalabs.uwv	3	150	pepperg
					Logout

Print Jobs	Document Name	Computer Name	Time Submitted	Page Count
<input type="checkbox"/>	http://www.kidneyyup.org/organ_donationFAQ.htm	MG170P	03:20PM Apr 19 2004	4
<input type="checkbox"/>	http://www.kidneyyup.org/organ_donationFAQ.htm	MG170P	03:20PM Apr 19 2004	4
<input type="checkbox"/>	http://www.kidneyyup.org/organ_donationFAQ.htm	MG170P	03:21PM Apr 19 2004	4
<input type="checkbox"/>	http://www.kidneyyup.org/organ_donationFAQ.htm	MG170P	03:21PM Apr 19 2004	4
<input type="checkbox"/>	Microsoft Word - Document in Microsoft Internet Explorer	MG184P	03:39PM Apr 19 2004	10
<input type="checkbox"/>	http://cccw.ac.uk/schoolofdesign/MA_COURSE/LMaze01.htm	MG162P	03:39PM Apr 19 2004	3
<input type="checkbox"/>	http://hojobs.yahoo.com/jobseeker/jobsearch/job_detail_print.h	MG190P	03:40PM Apr 19 2004	2
<input type="checkbox"/>	http://facstaff.wvu.edu/benigec/imagesearch/images/autunportall	MG161P	03:44PM Apr 19 2004	1
<input type="checkbox"/>	http://pro.corbis.com/popup/Enlarge.asp?mediaids=1667a69a	MG162P	03:49PM Apr 19 2004	2
<input type="checkbox"/>	Microsoft Word - Document in Microsoft Internet Explorer	MG184P	03:49PM Apr 19 2004	10
<input type="checkbox"/>	http://pro.corbis.com/popup/Enlarge.asp?mediaids=ffe0171	MG162P	03:50PM Apr 19 2004	2

Figure 1. User interface that students use to release their print jobs.

Our JSP web application was built using an open source Java framework called Jakarta Struts. Jakarta Struts implements a Java-based Model-View-Controller model which does not just apply to Java or web applications. Applications written in SmallTalk during the 1970's and early 1980's used this kind of model to help keep the View or presentation layer separate from the Controller or business logic. Using a structured framework like this allowed us to develop cleaner and more efficient code in a shorter amount of time. We also leverage other Jakarta projects such as Log4J to record system errors in different formats and DBCP to allow database connection pooling. All of these open source products gave us much needed functionality without having to design it all ourselves.

This web application runs on a Novell Netware sever using Apache Tomcat as the Java application server and Apache HTTP Server as the web server. The database server is also an open source product called MySQL. All of these open source applications are packaged and supported by Novell as part of their operating system. They are all very robust and stable since they have been in development for many years. Novell has also taken these open source applications and tested them for compatibility and performance optimization with their operating system. The support by Novell on these open source products lessens the risk of using open source application in an enterprise environment.

2.3 How Our Print Release System Works

After six months of part-time development by one developer, we produced the print management software that we are currently using. Our software allows us to collect the print data we were

looking for, as well as control printing with a conditional release of print jobs. Students print from their desktop the same way as they would anywhere else. The job is then held in a print queue on the server until the student is ready to release the job. The student then walks up to a print release workstation next to the printer and logs in with their campus e-mail username and password. Once they have authenticated, they are shown a list of their print jobs waiting to print. They then select the job they would like printed and it is released. The job is then printed. The data associated with that job is stored in our database. If the print jobs are not released after a certain amount of time, usually an hour, they are automatically deleted from the server's print queue. The time that a job is held in the print queue is configurable on a per printer basis. The lab workstations require no client software and the print release stations require a base install of Windows with any modern web browser.

The application is completely customizable on a per printer basis. Lab managers can control how long they would like to keep jobs in their print queues, the maximum number of pages per job that a user can print in their lab and the number of jobs they can release without having to authenticate again. All of these options give our lab owners better control and allow them to configure the application in a way that best suits their needs and budget.

The amount of data we are gathering is crucial in helping us make more informed decisions. We are collecting important information such as who is printing the job, what they are printing, how many pages and copies the job contains, and the times they submitted and released their print jobs. We also can easily keep track of secondary information such as what workstation they were sitting at and how big the file was that they are printing. We are even tracking data for jobs that aren't being printed.

3. OUTCOME

During the Fall semester, we used our software only in the two General Access Labs. We felt it is best to ease the software into production and wanted to get students accustomed to the process of using a print release station to get their print jobs. We also wanted to work out unforeseen technical issues or user interface problems before distributing it to the rest of campus. What we found was that our interface was a little confusing and it was taking students a long time to find their jobs and release them. As a result, we had our student lab attendants watch the users' habits and we also collected user comments on what they would like to see done to the software to make it easier to use. We then made gradual modifications over the first semester to improve the user interface and make it friendlier to the students.

3.1 Reducing Printing Without Charging

In the first semester alone, we saw a 25% drop in printing in our General Access Labs. Approximately 10% of that was a result of the conditional release of print jobs and the other 15% was due to changes in our business process. In the past, students would print out their jobs and go up to the front counter to pick up their printout. The General Access Labs had three students on duty to pull jobs from the printers and put them on the front counter. Each job had a colored cover sheet on it to identify the job. When we moved to this conditional release system, the cover sheet was no longer needed since the release stations are right next to the printer and once a job is released, it immediately prints.

By the end of the first semester, we started offering this service to other departmental labs across campus. In a short time period of one semester, our departmental labs have already seen a reduction in the amount of printing that goes on in their labs. While the lab managers have a short list of enhancements that they would like to see implemented, they are satisfied with our solution overall.

3.2 Data Reflect Effectiveness of Print Release System

We have been running our homegrown print management software for a full year now. We are successfully collecting data on: how much is being printed, who is printing it, and what they are printing. The controlled release of print jobs allows us to realize cost saving immediately by discarding obsolete and casually sent print jobs. Yet, we did not restrict students' printing activities, nor did we impose a cost on printing.

Table 1. Yearly Printing Trends at UW-Whitewater

School Year	Pages	% Increase
1997-1998	2,539,981	-
1999-2000	2,959,970	14.2%
2001-2002	2,924,584	-1.2%
2003-2004*	2,173,468	-34.6%

* With homegrown print management software

On the whole, we are encouraged with the reduction we saw over the past year. We are now analyzing the data collected and planning on our next step. There are a number of interesting findings: Our statistics show that 20% of the students are doing 80% of the printing. This means that all of our student body is paying for the excessive printing of the minority. We identified students who are printing over three times the average number of pages and sent those students a friendly letter to urge them to be more conservative in their printing. We are tracking the major field of studies of these 'top' printing students and trying to find if there is a trend of heavy printing among majors.

3.3 Looking Into the Future

We have been very pleased with our homegrown software and have plans of enhancement for next year. Among these include the accommodation of on-demand print release in teaching labs and areas where guest printing is needed and authentication may not be practical. While charging students per page is not our top priority goal, and we plan on continuing to the next academic year without charging our students, we will look into adding a billing interface into our software should the need arise. Should we need to charge students, our system will allow us to keep the charge per page low since we do not have the overhead maintenance of a commercial print release system. No matter what our next steps are, they would not have been possible without the valuable statistics generated by our system.

4. CONCLUSION

Sky rocketing printing costs is a major problem on college computing labs nationwide. Many universities are paying a substantial amount of money in order to recover their printing costs. An unofficial estimate of average charge is around 6 cents per page for printing. The actual cost of the paper and toner is at most 2 cents per page (See Table 2). That means 4 cents per page

goes to pay for the print management software and improve the return on investment.

Table 2. UW-Whitewater cost per page

Printing Costs			
Paper Costs	200,000	pages	\$904.00
Cost per Page			\$0.0045
Cartridge Costs	65,000	pages	\$463.45
Cost per Page			\$0.0071
Total Cost per Page			\$0.0117

Our homegrown solution has allowed us to significantly reduce our printing without any significant additional costs to us or our students. We believe this is a notable success story and we are optimistic that with more enhancements on the software, this system will solve our printing dilemma for the foreseeable future.

Thanks to open source technologies, we were able to develop robust and stable software. Without these technologies, we would have to purchase expensive commercial products or else invest

substantial staff time in developing the components ourselves. Either way it would have increased the cost of our project substantially. Open source technologies have become a proven competitor to commercial products. With companies like Novell offering support for these products, it makes successful and inexpensive application development projects possible.

5. ACKNOWLEDGMENTS

We appreciate the trust and support from the University of Wisconsin-Whitewater Management which allows us to explore and gives us the opportunity to develop this print management software in-house. We would also like to thank Novell and all of the open source developers out there because without all your time and effort none of this would have been possible.

6. REFERENCES

<http://facstaff.uww.edu/pepperg/warhawkprinting>

Informational website including documentation, presentations and online demos.