

# An Open Source Print Management System that Works

Louis W. Canaiy

Engineering Network Services, Colorado State University  
0914 Campus Delivery  
Fort Collins, Colorado 80523-0914  
1-970-491-2465

louiec@engr.colostate.edu

Mark Ritschard

Engineering Network Services, Colorado State University  
0914 Campus Delivery  
Fort Collins, Colorado 80523-0914  
1-970-491-7954

ritschrd@engr.colostate.edu

## ABSTRACT

At Colorado State University, the engineering computing labs were running up huge printing costs due to excessive printing. We looked into several commercial print quota solutions that were quite pricey and limited in capability. We wanted something cheap (who doesn't?) and available across several computing platforms. The open source option we settled on consists of a print daemon (LPRng), Samba, a custom accounting script written in Perl, MySQL, Sendmail, and the print filter IFHP. In addition to our command-line administration utilities, we've developed a web-based administration system that uses Apache and PHP as a scripting language. The web-based system greatly simplifies our administration needs. Our system currently runs on the Linux platform, but could conceivably run on any Unix or Unix-like flavor. Our conglomeration of code bases satisfies all our requirements, and we have realized a large decrease in costs, as well as an associated increase in intelligent printing practices on the part of users. In this paper, we will outline the components of the PQuota system, how they interact, and their role in fulfilling our objectives of cost effectiveness, cross platform availability and ease of use for both administrators and users as well as low user impact.

## Categories and Subject Descriptors

K.6.4 [Management of Computing and Information Systems]: System Management – Centralization/Decentralization, Management Audit, Quality assurance

## General Terms

Performance, Design, Economics, Languages

## Keywords

Print Management Systems, Printing Quotas, Open Source Code, Accounting, Cost Reduction, SQL, Perl

## 1. INTRODUCTION

In the fall of 2001, we at Engineering Network Services (ENS) at Colorado State University decided to undertake a print accounting project. This was precipitated by a growing financial concern due to the large amounts of resources being put into supplying consumables for the student labs. ENS was being placed in a situation where a great deal of money was used to pay for paper,

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.

toner, and maintenance where the same money could be used to provide new and requested services to students and faculty. We looked into several commercial solutions for our print accounting system, but found them to be lacking functionality, and high in price. Many commercial solutions were tailored to a windows environment, and were difficult to use in the mixed platform environment that is the engineering network. ENS actively supports several flavors of Unix, Linux, Mac OS, and several MS Windows variants. We wanted a print accounting system that would support all platforms supported on our network. To achieve this, ENS decided to create our own print quota system.

## 2. PQUOTA

PQuota is currently in version 2. The original PQuota was borrowed heavily from work done by Shaun Rowland at Ohio State University. A few of his concepts live on in the current PQuota. PQuota 2 is based very loosely on the accounting.pl script written by Patrick Powell and distributed with the LPRng source distribution. Very little of Mr. Powell's code base exists in PQuota 2, but again, several of his concepts live on in the second generation system. The growth from PQuota version 1 to version 2 was due in large part to collaboration between Louis Canaiy in the college of engineering, and Larry Karbowski in the college of natural sciences. Larry laid the essential groundwork to move PQuota to an SQL based system, and has also been instrumental in testing PQuota in a Windows-only environment.

The initial rollout of PQuota was fairly uneventful. After thorough testing in the ENS office and in several test environments, PQuota was deemed lab-ready. We expected a loud uproar from students as their printing practices were curbed, but we witnessed very few complaints. The system is mostly transparent to the end user, with the exception that the user receives the status of their quota at the end of each print job. The groups with the biggest number of complaints were faculty and staff. They were used to using student-funded equipment to print, but the new accounting system disallowed this.

After rolling out PQuota, ENS immediately saw a 50% decrease in the amount of consumables we were consuming. The savings have been excellent. We have found most of these savings have been a result of student awareness, and not price-gouging on our part. The quota given to a student every year is quite adequate, and what a student doesn't use is rolled over into the next year. This allows students to accumulate quota during the progress of their schooling. When a student reaches their final semester (and a great number of papers are typically due), they have an adequate print quota to use. Students can also purchase additional quota at a very low fee, if needed.

The issue of quota vs. dollar amount was addressed by simply calling print quota units “credits”. A credit is equal to printing on the cheapest page size on our cheapest printer in the labs. This happens to be a single duplex 8.5x11 page on an HP LaserJet 8150. The dollar amount for a print quota credit works out to be about one eighth of a cent. When students come in to purchase quota we sell it in ten dollar increments, which works out to roughly 890 credits per 10 dollars.

With the introduction of PQuota 2 and its MySQL back end, students are able to view their print history and quota amount through a link from the ENS home page. Students find this most helpful. The SQL nature of PQuota 2's functionality lends itself nicely to generating reports using a variety of tools. The remainder of this paper will focus on PQuota 2. It is turning out to be a very easy system to tailor to satisfy very specific needs.

### 3. PQUOTA COMPONENTS

PQuota is made up of several open source and free components. Their combined interaction creates a very stable, user friendly, and easily administrable system. Currently, ENS has PQuota running on a mid-grade x86 box running Redhat 8.0. On the box, we have installed: Perl 5.8.0, Samba 2.2.7, LPRng 3.8.9, IFHP 3.5.17, MySQL 3.23, the Perl modules required to interface with MySQL, and Sendmail 8.12.8. For our web-based administrative interface, and for user interfaces, we have installed Apache 2.0.4 and PHP 4.2.2.

#### 3.1 Redhat Linux

Redhat Linux was chosen because ENS already supports Redhat in our labs, and we have a great deal of experience with Redhat's distribution. Linux was an obvious choice because it's free and stable. Linux offers an excellent platform from which to host PQuota and have it be available to a wide array of print clients.

#### 3.2 Perl

The brains of PQuota's accounting have always been Perl. Perl exceeds greatly at document processing and tying systems together, so it was an obvious choice for PQuota usage. Also, having a scripting language as the brains of our operation makes on-the-fly changes and testing very easily accomplished. Perl's great extensibility has also been a great asset.

#### 3.3 LPRng & IFHP

LPRng is the brains of the printing subsystem. It makes itself available to most any flavor of Unix “out of the box” (Mac OS X included). LPRng is an enhanced, extended, and portable implementation of Berkeley's lpr print spooler functionality. LPRng handles all document handling, and gets documents to their respective printers. IFHP is a print filter. It handles formatting of raw print jobs, and document translation when a printer won't support a document as submitted. (A Postscript to PCL translation is common.) From a user's standpoint, the most visible function of IFHP is to put a “{username} is printing” message on the lab printers' display. Our lab users seem to find this most helpful.

#### 3.4 Samba

Samba is the piece that allows our Windows clients to talk to the PQuota server. It takes print jobs from the Windows clients and hands them off to LPRng. Samba also acts as a messaging medium, as our system will display print status to users post-

printing. Samba also allows us to notify users of zero or negative quota when the user has exhausted their quota.

#### 3.5 MySQL

MySQL is the memory and data retrieval chunk of PQuota. Putting PQuota 2 on an SQL base has allowed us incredible database querying abilities above and beyond what we had in PQuota 1. MySQL has also allowed us to produce some fancy reports for users and has facilitated our web-based administrative tools.

#### 3.6 Sendmail

Sendmail is not completely necessary for PQuota to function correctly, but it can be used to mail users their print quota status instead of relying on Samba's messaging capability. Sendmail and Samba messaging can be used together though. They both perform the same function of keeping users aware of their quota status.

#### 3.7 Apache & PHP

Apache and PHP serve as our “easy administration” platform. While these two pieces of software are not required, they do make many things much easier. Without them, PQuota tends to just run in its own space without much exposure to the outside world or its users.

### 4. PQUOTA COMPONENT INTERACTION

Figure 1 may be helpful in visualizing the interaction between various PQuota components in conjunction with the following description.

LPRng is the hub about which everything else revolves. It provides the means of actually getting documents from users to the printers. The flow of a document through the print system is thus:

- A job is submitted  
A job can be submitted in one of two ways -- via lpr (or similar method), or Samba. Since LPRng complies with Berkeley's lpr print spooler, most any Unix flavor can submit jobs in this fashion. We have Samba configured to distribute print drivers and participate in a Windows 2000 domain, so Windows users see the PQuota server as just another Windows server.
- LPRng activates our Perl accounting script with a start flag  
The accounting script 'accounting.pl' queries the PQuota database to see if the user has sufficient quota. The script then parses the job for any postscript or PCL information that will be useful in charging the user's quota. Once parsing is done, the gleaned information is saved in an accounting file (each printer on the PQuota system gets its own accounting file), and the job is allowed to continue. If the submitter does not have sufficient quota, a message is sent to the submitter via Samba and/or Sendmail to notify the user of their inadequate quota, and the job is killed.

(continued)

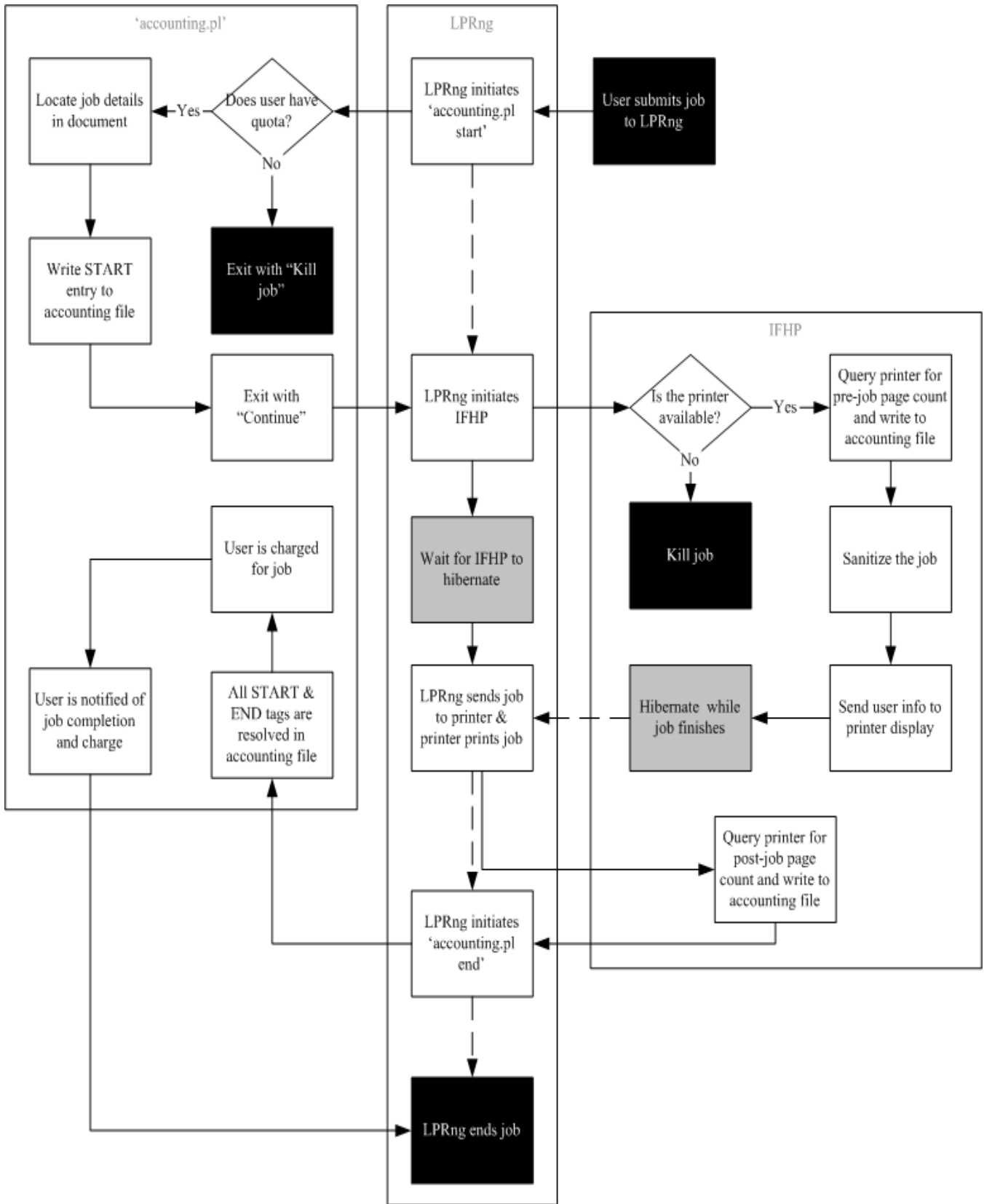


Figure 1. PQuota Component Interaction

- LPRng activates IFHP  
IFHP's first task is sanitizing the submitted job. This is done as per the printers' definitions contained in an 'ifhp.conf' file. We at ENS had nothing to do with these magical definitions and the excellent job they do. We have the developers of the IFHP project to thank. Sanitizing is crucial as the process removes any odd or malicious postscript commands, and makes the job fit its destined printer. Thereby, IFHP's sanitizing makes the job sane. IFHP then queries the printer to find its status, and if the printer is unavailable, IFHP instructs LPRng to re-try the job submission three times after waiting a predefined interval. If after trying three times, the printer is unavailable, IFHP tells LPRng to kill the job. If the printer is available, IFHP asks it for a starting page count, and writes that information to the accounting file. IFHP's next task is purely cosmetic. It sets a "{username} is printing" dialog on the front display of the printer. Once the above is completed, IFHP hibernates until the printer notifies the PQuota server that it's done printing.
- LPRng sends the job to the printer  
The printer prints the job, and upon completion, notifies IFHP and LPRng that it's done.
- IFHP comes out of hibernation  
IFHP queries the printer for a post-printing page count and records this number to the accounting file.
- LPRng initiates the 'accounting.pl' script with an end flag  
The 'accounting.pl' script uses the pre-printing and post-printing information IFHP entered into the accounting file to resolve all unresolved START entries. It does this by creating END entries in the accounting file. As it creates END entries, it also charges users the appropriate quota for the job being resolved, and messages the user. The method of using START and END flags in an accounting file was borrowed from Patrick Powell. Mr. Powell wrote the original accounting.pl script that is distributed along with LPRng. We have reworked his process into something much more efficient, reliable, and user friendly.
- LPRng ends the job  
The print job has been completed. Of course there are a multitude of things that can go awry in this process, but with the help of LPRng and IFHP, our system does an excellent job of getting documents churned out and users charged their fair quota even when problems do arise.

## 5. FULFILLING OUR OBJECTIVES

PQuota has done an excellent job in fulfilling our objectives. All of its components are free for our use and most have their source open for all to see. The biggest cost to us has been that of development.

The open nature of PQuota has also been a huge asset. It has allowed us to share information with other colleges at Colorado State University. It has allowed us to tie into their PQuota system, and allow them into ours. This is useful for students taking classes in another college, and allows the students to use their print quota from their home college in foreign college labs. PQuota's open nature also played into our requirement of cross-platform availability. PQuota was envisioned as a very versatile system, and its implementation has closely followed that vision. It can avail itself of almost any client.

Our last requirements of PQuota were ease of use and low user impact. As far as ease of use goes, PQuota is pretty much "set it up, and it goes." It makes reporting extremely easy (its data storage is SQL after all), users have to simply point and click in the Windows and Mac world, and Unix connectivity is no less difficult than it normally is. Our web-based administration system is an excellent, easy, and malleable way to administer a system. One large benefit that we were hoping for, and have realized on a grand scale, is user awareness.

PQuota was developed as more of an awareness generator than a tool to nickel and dime users to death. We are fair with the amount of quota both given and deducted, and users seem to realize and appreciate this. We have had little to no complaints about PQuota, and have heard much praise.

## 6. ACKNOWLEDGEMENTS

We would like to thank Larry Karbowski for his work and help on PQuota and especially for his early work on integration with MySQL. We would also like to acknowledge the work done by Patrick Powell for inspiring a template for PQuota 2's accounting method, and Shaun Rowland for his inspiration on how a home grown print quota system might be developed. Lastly, we'd like to thank and acknowledge all the large amounts of work many developers have put into all the systems that comprise PQuota.