INTERNET BOOK PUBLISHING -- A CASE STUDY

John C. Nash
Faculty of Administration
University of Ottawa
136 Jean-Jacques Lussier Private
Ottawa, Ontario, K1N 6N5
fax: (613) 564 6518
email: jcnash@aix1.uottawa.ca

Mary M. Nash
Nash Information Services Inc.
1975 Bel-Air Drive
Ottawa, Ontario, K2C 0X1
fax: (613) 225 6553
email: mnash@nis.synapse.net

Abstract *This paper presents the history and analysis of the publishing of the book "Scientific Computing with PCs" via the Internet. We detail some of the obstacles to such endeavours and to the analysis of log files of user actions. Recommendations and opinions are offered to those considering Internet publishing ventures.*

Introduction

Since Gutenberg developed printing with moveable type, mass marketing of books to the public has been possible. Publishers have had to devise ways to finance and market material: subscriptions in the 18th century; Dickens' serialization in newspapers; paperbacks; joint paper / microfiche editions in the 1970s; paper / disk in the 1980s. The present article considers electronic-only publishing via the Internet.

Our efforts are motivated by costs. Conventional paper publishing imposes inventory costs associated with storage, handling and remaindering. There can be a waste of resources in the form of unsold materials but also an opportunity cost in that out-of-print works generate no revenue.

In 1975 one of us (MN) prepared a Master's thesis on the feasibility of publishing materials using microforms. This study was published as a microbook: a small paperbound book, the size of a microfiche, with important chapters in paper and the remainder on microfiche (Nash, M. M. 1976).

FTP and other Internet Publishing

FTP (File Transfer Protocol) is a popular mechanism for moving computer files between computers via the Internet. File collections may be made available to restricted users who must supply a username and password, or else can be open to anyone who gives "anonymous" or "ftp" as a username and their electronic mail handle as a password. The user, once connected, is permitted to access and download files in a restricted set of directories.

The FTP mechanism is now common for transferring binary program files for specific machine types, or text files for any platform. The binary mechanism is used for most word processing or desktop publishing files, usually on a single hardware/software platform. To overcome the machine dependencies, many workers have chosen to provide a PostScript form of their work. PostScript (TM Adobe Systems, Incorporated) is a plain text programming language that instructs a printer how to "draw" on the page. Few humans actually program in PostScript, but many computer programs output material in this form, and many printers can deal with it. The Free Software Foundation has versions of a program called GhostScript for many platforms that allow PostScript files to be viewed or to be printed on most popular printers, even very inexpensive dot-matrix types. A disadvantage of PostScript is that printing speeds, especially for grey-scale graphics such as digitized photographs, can be glacially slow. There are some other choices for machine independent output files such as Adobe Acrobat or TEX DVI files, but at present PostScript is more common.

PostScript files are unfortunately large. One can compress the files, sometimes into a single archive, using such tools as COMPRESS, STUFFIT, LHA, ARC or PKZIP. The resulting files are then binary and may be platform dependent.

Since PostScript is text, we could consider sending it by electronic mail. Unfortunately, operational restrictions usually require that we limit email files to about 50000 bytes and have line lengths less than 80 characters. PostScript files will thus have to be broken into pieces and line lengths checked. Bitmaps of digitized images often have long lines. We have had trouble printing such documents that were sent by automatic email file servers.

Our book -- history

"Scientific Computing with PCs" was originally commissioned and published by Reston Publishing Company in Reston, Virginia under the title "Effective Scientific Problem Solving with Small Computers". It was released in August 1984, recalled to replace some pages we reported missing, and published late in 1984, a day or two before the collapse of Reston. Prentice-Hall took over Reston, but failed to list this book!

Despite this marketing failure, we noticed the book on shelves in Australia, Belgium and the UK. It strains credulity that the author would randomly encounter three copies when the publisher claimed only 38 were sold world-wide. In 1985 we got the rights back together with 200 free copies of the book. In 1991, CRC Press of Boca Raton, Florida contracted with us for revised version of the book, though we had indicated a preference for a project on risk management. A camera-ready manuscript was delivered to the publisher in November of 1992. Apart from an acknowledgement of receipt, we were unable to raise any response by letter, telephone or fax until, under threat of legal action, we obtained a release in June of 1993 with CRC's decision not to publish. (Fortune Magazine reported exceptionally heavy losses by the Times-Mirror, the parent company of CRC.)

We undertook a search for a new publisher, and had several requests for the manuscript during 1993 and 1994. The general economic climate, however, led to the closure of the division of an international publisher that was considering it. Several other publishers indicated that the generality of the topic made it difficult to market, a point with which we agree.

Given the inexorable aging of the material, we decided to carry out a re-editing of the material and to publish it via the Internet. We paid a trusted colleague to review the material and suggest areas for improvement and implemented these suggestions during the summer of 1994. We also reformatted the work from a 9" by 6" size based on CRC's needs, to one that fits well on standard 11" by 8.5" paper. We believe, but have not confirmed directly, that the pages also fit on A4 paper. In September/October 1994 we prepared a thorough index, but keyed this to sections rather than pages in case we change pagination later.

Since the work was prepared in WordPerfect 5.1, PostScript output is easily obtained by selecting a PostScript printer and specifying a filename instead of an output port. A minor delay was occasioned by trying WordPerfect 6.0a for DOS; this proved so frustrating that we now avoid this version. The PostScript output from WordPerfect is not, however, fully satisfactory. Because it is intended for use directly by a printer, it lacks some labels and other useful features that allow programs such as GhostScript to select pages. This would be especially useful when viewing PostScript on the screen, or in re-printing single pages when there has been a paper jam or loss of ink density.

The total PostScript output is very large, so we broke up the book into four sections:

file	# bytes	
SCPC-1.PS	358750	contents, preface, and pages 1-23
SCPC-2.PS	1023058	pages 24-101
SCPC-3.PS	1146671	pages 102-148
SCPC-4.PS	258871	pages 149-199

We also prepared a "teaser" consisting of the Preface, Contents and Chapter 14, called SCPCDOC.PS, as well a test file 00TESTPG.PS. The files were loaded on our server, named macnash.admin.uottawa.ca, a Macintosh Quadra 650. Since the line ending of text files differs between Macintosh, MS-DOS and Unix, we used the MacLink software to load files onto the Macintosh from a floppy disk.

Tests showed that WordPerfect outputs ASCII character 4 at the end of its PostScript. We used an editor to remove this. It is translated in the Macintosh into a character that upsets some PostScript printers and may upset GhostScript.
To shorten FTP transmission times, we wanted to compress the files. As mentioned, compression programs may be platform dependent, but again the Free Software Foundation has introduced GZIP. We found programs for Unix, MS-DOS and Macintosh. However, when we tried to move files between machines we found that the FTPd server software on the Macintosh and the WS_FTP client on our PC would conspire together to translate the binary gzip file to a text form (Binhex), but the name would not be changed. Without extreme care in selecting options, successful transfers were difficult. We finally decided to use PKZIP for compression. There appears to be a Macintosh decompressor for this format, but possibly none for Unix. However, we have left the raw PostScript files on the server. The ZIP files is approximately 750K bytes.
The book was launched on the Internet on Jan. 1, 1995 at 18:00. Its availability was announced through six technical newsgroups dealing with numerical analysis and science. Within minutes of its announcement people were downloading the material. We also announced the material via the NA-NET News, a weekly email digest about numerical analysis and scientific computing. We have since made a follow-up announcement when we loaded the compressed file onto the server ftp.synapse.net. Initially the files were in the info/nashinfo/ directory, but more recently the structure of the server has been revised and files are in the private/n/nis/ directory.
We should note that our announcements have come to the attention of a small publisher who has expressed interest in releasing the book conventionally.

User Reaction

Users have expressed several concerns:

- Rising traffic levels on the Internet and the large file sizes have resulted in time-outs for many users. We have delivered two copies of the files by reverse FTP, that is, where we uploaded them to a user site, one in Germany and one in

Australia. We have also converted the ZIP file to the UUencoded text form as 18 email files. This has been sent to five users.

- Many users think PostScript files are binary, or try to download ZIP files as text, with the predictable failure to receive the material.
- Some users have complained that they cannot print PostScript. We have informed them about GhostScript (available by ftp). However, one needs to be a fairly sophisticated user to be able to profit from this information. We have also been asked to supply the book in text form; since it has quite extensive graphics this is not possible.
- At times our server had more than five people trying to download files and refused connections. We increased the limit but the FTPd server software then ran out of memory. We also noted two crashes that we believe are related to running Gopher service with this software. We removed this service in mid-January and have had no further crashes.
- Less than a dozen users have actually paid the requested $10 licence fee. Some have made excuses about banks, currency regulations or student poverty. We have had requests for free site licences from very large corporations on the grounds that they would edit the work for us. (We suspect our use of Canadian spellings to be behind this. Interestingly the source of this request does not show up in our log files, so must have received the files via an intermediary.)
- The distribution of the work has been truly global. See Figure 4 below.
- Some people will download everything they find. A number of those who have connected have clearly not bothered to read welcome messages and have downloaded only portions of the files. However, it is clear that there is a body of serious users who take the time to find out what is available and who do download the sample information.
- Some users have sent email messages complaining that they want plain text files of the material because they "cannot print PostScript". We have made a conscious

decision NOT to accede to this request: our book has been carefully typeset and laid out and includes graphics that are an important part of the content. Furthermore, we do not wish to tempt others to use our words.

Tracking activity

Most FTP server software allows the operator ("sysop") to log activity. We have done this and have attempted to analyze the activity on the SCPC files. There are, however, a number of obstacles to such analysis. In consequence, an accurate picture of the true number of successful downloads may be elusive. First, different server programs provide different log formats, so we must arrange different tools for their analysis. Here we shall only look at the log up to 16:30 on 13 March 1995, on the macnash server, which uses the Mac-FTPd software. A small segment of the raw log is shown in Figure 1. The log entries include server startup and shutdown, gopher operations, ftp login, logout and timeout, and, most importantly, ftp get-file. The client Internet Protocol (IP) number is included, as well as the email handle provided as login password. The latter item is quite often useless, as our software does not check even for minimal conformance to conventions. The IP number should tell us the machine used, but this is NOT always the case:

- Many service providers dynamically assign IP numbers, so that a given user may have a different IP number each time they log in.
- Some IP numbers have no corresponding name in a Domain Name Server (DNS). The PC clone next to "macnash" has IP number 137.122.26.86, but no machine name. Macnash itself is 137.122.26.157.

Figure 1. Sample portion of FTPd log file (8K)

We also cannot easily be sure that a file has been received. If we examine the log files, it is clear that some getfile operations are followed many transactions as much as two weeks later by time-outs or log-outs. With some considerable effort, mainly to take care of largely unimportant details relating to operator shutdown or restart of the service, we wrote a program to reorganize the log file to allow

creation of a database of actions. In each getfile record we included the time until the next operation by the same user in the same session. If the next action was a time-out, we made this time negative and have assumed here that it was unsuccessful. We also assumed transactions taking over 30 minutes were unsuccessful. A "bug" in the program that caused a lot of delay turned out to be due to the American month/day/year date convention in a subprogram we used. In raw terms, the logfile, even though a partial set of transactions relating to the book at hand, had 9026 records. We deleted all that were not FTP getfile operations, leaving 3741. (There were a few attempts to use gopher to get files, but these appear to "crash" the server software and we disabled the gopher feature of FTPd in mid-January.) We then classified the getfile operations as Type 1-4 for files SCPC-1.PS, ..., SCPC-4.PS, Type 5 for SCPCALL.ZIP and Type 6 for SCPCDOC.PS. All other getfile records (1316) were deleted. Then all records where the email source had the string "nash@" in it were dropped, as these are our tests (46 in all). We confirmed this using our IP numbers. We deleted any 1994 entries, representing trial attempts of a few people we informed prior to the official launch. This left 2372 transactions.

Using dBase III+, since we are familiar with its programming, we then output the data in a text form suitable for use in either Minitab or Stata statistical software. This gave the transaction type, the hour, the day number in 1995 and the transaction time.

A graph of number of transactions by day number is given as Figure 2. A similar graph by time of day is given as Figure 3. These show obvious buildup and then drop off of activity with newsgroup appearance of the announcement and a relatively stable connection rate around the clock.

Figure 2. Day of year distribution of Getfile operations. (7K)

We also put out a list of IP numbers in text form, sorted it and stripped out all duplicates. We then edited the command "host " (space significant) in front of each IP number, transferred the file by Kermit to a Unix machine on which one of us has an account, changed the mode of the file to "execute" and "ran" the file

while capturing terminal output so we could find the machine names connecting. Figure 4 shows the wide geographic diversity of the connections, as well as a large number of non-standard sites.

Despite the information we show, the database queries to determine the following interesting information about server/user interactions are not obvious or easy:

- the precise number of times an individual attempts a download, given the IP number / email handle issues above;
- the number of users who download all the files in the set, should they do so in separate sessions;
- the number of users who check the SCPCDOC.PS file then log in later to get the rest of the files.

Figure 3. Time of day distribution of GetFile operations (12K)

Figure 4. Geographic distribution of GetFile connections (8K)

The FTPd software for the Macintosh does not let us see how many users are active. Indeed we know we have shut down the system while users were active.

Our recommendations

The experiences reported here lead us to the following conclusions.

1) FTP publishing on the Internet is useful as a way to make available works to a wide audience at low distribution cost.

2) Until simple and reliable credit-card payment methods are in place, revenues from such publishing will be very low. It is useful, as in this case, when it is more important to have a work published than to be compensated for the effort.

3) Many, and perhaps the majority, of current Internet users do not understand the processes. Furthermore, they will complain, even if their complaints derive from ignorance. Unless they have paid money, we can choose to ignore them. Eventually we may be able to direct their messages to a conventional publisher.

4) It is important to have the files mirrored at secondary sites because of the growing traffic on the Internet and peculiarities of some connections.

5) Analysis of the log files to determine what has really transpired is time consuming. The task is magnified if there are mirror sites as the log file formats may be different.

Reference

Nash, Mary M. 1976. *Books on Demand: a microbook*. Oxford: Oxford Microform Publications.