Association for Information Systems
AIS Electronic Library (AISeL)

CONF-IRM 2009 Proceedings

International Conference on Information Resources Management (CONF-IRM)

5-2009

Inter Process Communication and Prioritization to Enable Desktop Advertisement Mechanism

Shubham Gupta
Jaypee Institute Of Information Technology University, shubham.gupta@techquark.com

Varun Mittal
Jaypee Institute Of Information Technology University, varunmittal@varunmittal.net

Follow this and additional works at: http://aisel.aisnet.org/confirm2009

40. INTER PROCESS COMMUNICATION AND PRIORITIZATION TO ENABLE DESKTOP ADVERTISEMENT MECHANISM

Shubham Gupta
Jaypee Institute Of Information Technology University, India
shubham.gupta@techquark.com

Varun Mittal
Jaypee Institute Of Information Technology University, India
varunmittal@varunmittal.net

Abstract

This research paper tries to bring in a new concept of desktop advertising mechanism by synchronization it with the running processes and the data on users' side. The proposed approach shall be based on inter process communication interaction, scheduling, prioritization, desktop crawling and system calls. The running process status and data will be fetched by the proposed process, which will then seek relevant information with the remote ad server and display the advertisements fetched based on keywords on user side.

1. Introduction

There are various advertisement mechanisms available in the market and only few products deal with desktop advertisement mechanism. Here, in the proposed mechanism, we deal with optimization and improvement of desktop advertisement mechanism by using the concepts of inter-process communication, prioritization, desktop crawling and scheduling. This way the relevance of advertisements being displayed on client sides will be highly relevant, as such advertisements will be determined by the running processes, history and data on hard drive at user end. This way the user will only see advertisements in the areas of their interest or which they work upon.

2. Need for Mechanism

The absence of any desktop advertisement mechanism is one of the main reasons which will deliver the early mover advantage to it.

The online advertisement market is highly competitive and comprises several well entrenched players. The proposed mechanism will open new vistas for advertisers. It will forge a revenue stream for normal users in addition to the site and blog owners.

This solution will benefit advertisers and customers, as in advertisers will be able to deliver information about their products to users who are interested in their domain and users will get relevant information and will also be paid for browsing advertisements.

This solution examines various policies to determine which properties of a inter process communication and prioritization are the most significant determinants of performance and how to improve the relevance of advertisements being displayed without degrading performance of system and applications. Moreover, the privacy policies are also discussed to suit user's requirements.

3. Solution Approach

Our desktop advertisement mechanism uses client-side application that is installed on the user's computer via an agreement by advertisement companies with many advertisers and website owners. Such desktop software product works with:

1. Processes running in background and foreground and data inside them synchronization it with advertisements.

2. The browser history of surfed site and downloads will also give keywords to display advertisements. The data collected by the software will be of anonymous type and user can even turn off this option, if he / she have privacy issues.

3. Desktop crawler will be an integral part of the software which would index the hard drive and get advertisements related to the content and data found. The user can customize search areas and folders to be indexed. This option can also be turned off and limited to certain search filters to deal with privacy issues. Moreover, indexing of data will take place during idle time of system to reduce system load.

This mechanism will use the concept of prioritization by checking the bandwidth of the network (using default 20% reserved bandwidth for the OS) and also load on system and hence set priority of the program and advertisements being displayed as per requirement. In case system is on high load, it will keep displaying static advertisements. This solution will also display advertisements generating high revenue first and also related to the keywords.

The whole process of sending data to server, advertising matching done there and response back from server is transparent to the end user and such a desktop advertising software can be bundled with other software products such as third party freeware applications, games or screensavers.

We can use third party API's like Google Adsense API's to get advertisements based on keywords and display the fetched advertisements on end user's PC or even start our own advertisement server and get advertisers for the same.

This Desktop Advertising Software / Engine can be developed for business of Online Advertising, Ad Serving, Internet Marketing etc and also helping end users to earn. This is first of its kind concept where in we display relevant advertisements on desktop on users by Inter process communication and prioritization and desktop crawling.

4. Using IPC with Advertisements

IPC or inter process communication [1] is a method by which processes can interact or communicate with each other. In our domain we would be using it to communicate with the running process and obtain keywords from them. For example if a word document is open containing information about technical gadgets, system will try to communicate with the word process and fetch information from it to get acquire keywords and fetch advertisements based on that keywords. Hence, advertisements will be based on technical gadgets and will attract users' attention.

Most of the operating systems come with mechanisms for facilitating communications and data sharing between applications and processes. Collectively, the activities enabled by these mechanisms are called *inter-process communications* (IPC).

Typically, applications can use IPC categorized as clients or servers.

- A *client* is an application or a process that requests a service from some other application or process.

- A *server* [2] is an application or a process that responds to a client request.

In this case, our advertisement software will serve as client trying to fetch information from different processes and the running processes will serve as server giving out the same. This way the client i.e. advertisement software will get to know data residing and being used in those processes and will pass it to advertisement server and fetch relevant advertisements to display. This way the user will see advertisements relevant to him/her. For example, our software might act as a client in requesting a summary table of manufacturing costs from a spreadsheet application or a document from a word file or even the currently played movie in media player acting as a server.

After deciding that our application would benefit from IPC, we decided which of the available IPC methods to use. It is likely that our application might use several IPC mechanisms. The IPC mechanism which will prove best for our solution can be decided by answering the questions and factor below:

- The user of the application may either choose the other applications with which the application should communicate, or let the application implicitly find its cooperating partners.

- Is performance a critical aspect of the application? All IPC mechanisms include some amount of overhead.

- Should the application be a GUI application or a console application? Our IPC mechanisms require a GUI application.

- After taking various factors and conditions into consideration, we decide that our solution will be using two general approaches for operating system for handling the inter process communication task of the application. They can be termed as:

- Shared memory
- Pipes

4.1. Shared Memory

Shared memory, as the name defines, is the sharing of a memory portion by two or more programs running at the same time, so that both the processes can access the shared memory like regular working area. It is local to each machine and can only be used for communication between processes on the same machine.

The amount of shared memory used is limited by the amount of virtual memory on the machine, thus the shared memory being virtual is implemented by shared page table. The pages are referenced by page table entries in each of the sharing processes' page tables. It does not have to be at the same address in all of the processes' virtual memory.

As with all System V IPC objects, access to shared memory areas is identified by an integer key so if two processes call with the same key, they will get the same piece of memory. Thus, the access to shared memory areas is controlled via keys and access rights checking. Once the memory is being shared, there are no checks on how the processes are using it. They must rely on other mechanisms, for example System V semaphores, to synchronize access to the memory.
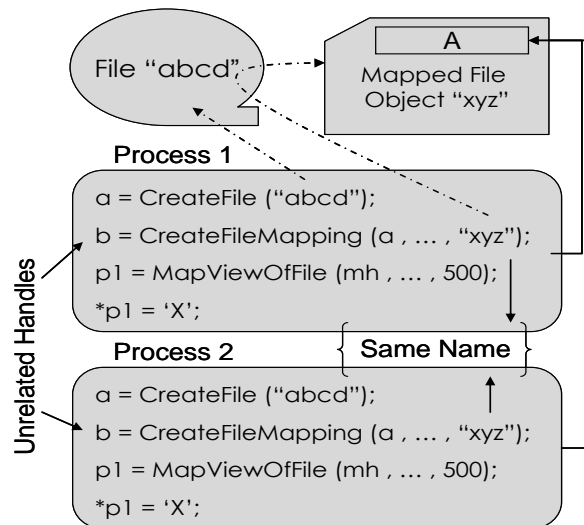

Figure 1. Using shared memory for IPC

The ability of many processes to access shared memory simultaneously is both an advantage and a potential problem.

When a process is part way through modifying shared memory, another process can read the shared memory and find that its contents are inconsistent, leading to intermittent errors. This can be avoided by using semaphores.

Another way to share memory in Win95/NT/2000 is through global variables of dynamic link libraries. This is a less efficient way, because the access to shared memory is achieved through function calls.

4.2. Pipes

Pipes provide a simple one way byte stream to connect two (usually related) processes. Use of pipes is made better by the fact that is the processes don't have to share the same language.
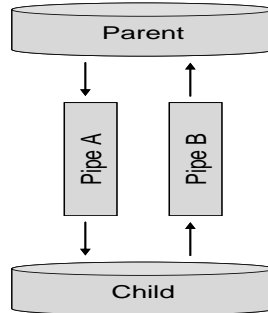

Figure 2. Using pipes by parent and child communication

There are two types of pipes:

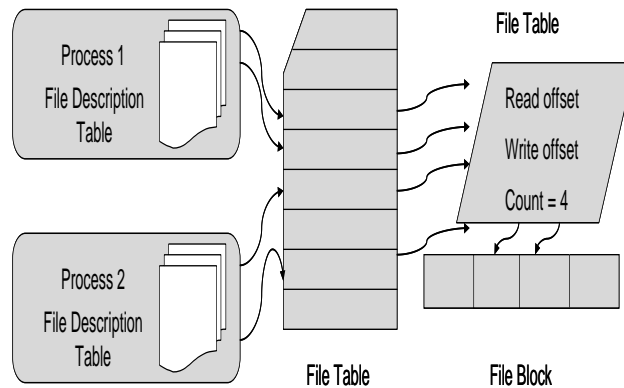1.   Anonymous pipes
2.   Named pipes

Figure 3. Using pipes for IPC

Anonymous pipes enable only the related processes and not the unrelated processes to transfer information to each other by redirecting the standard input or output of a child process so that the data can be exchanged with its parent process. Anonymous pipes cannot be used over a network.

While in contrast to anonymous pipes named pipes are used to transfer data between processes that are not related processes and between processes on different computers. Typically, a named-

pipe server process creates a named pipe with a well-known name or a name that is to be communicated to its clients, so that they can exchange data by performing read and write operations on the pipe.

In order to communicate two processes (P1 and P2) through anonymous pipes by redirecting the standard I/O, the processes don't have to be aware of the existence of pipes, i.e. their sources and images don't have to be modified.

Anonymous pipes provide an efficient way to redirect standard input or output to child processes on the same computer. Named pipes provide a simple programming interface for transferring data between two processes, whether they reside on the same computer or over a network.

In order to communicate two processes (P1 and P2) through anonymous pipes by redirecting the standard I/O, the processes don't have to be aware of the existence of pipes, i.e. their sources and images don't have to be modified.

5. Desktop Crawling for Keywords

Our solution will crawl i.e. search and browse through the files and data on users' computer to fetch keywords based on the collected data. As a desktop crawler our program will have two goals: to find the desired information, and to find it quickly. Our search programs will accomplish its speed by indexing the data it searches through. Instead of poring through every file as if it was being searched for the first time, the program will prepare an index of the essential information about each file. If you have 100GB of videos sitting around, our desktop search program will ignore the data, culling out titles, authors, and other metadata. Even if a search algorithm operates by brute force, the smaller data set of an index is faster to get through.

The index database will be optimized and compressed to reduce its size and improve performance. The user will also have option of encrypting the database for privacy issues. Since encrypting the index database reduces the performance of the solution, hence user if needed can encrypt the index database to satisfy his privacy and security issues on the cost of performance.

This way indexing of the data will help generate keywords which in turn will help retrieving relevant advertisements from the advertisement server.

6. Prioritization and scheduling of tasks

Using IPC to get information and data continuously from other processes might degrade the performance of the system and the applications. Hence there is need of prioritization and scheduling of the mechanism in order to reduce load on the system.

The prioritization and scheduling [3] can be done by many processes but the main aim will be to reduce the load on applications and system. This will be done by scheduling our solution to send request for IPC at varying time intervals. The time the system is busy or the CPU load is high, our program will suspend the IPC operation and will keep displaying advertisements which were

fetched before i.e. the program will keep displaying static advertisements till the CPU load is high or system is busy in some high load work. Moreover, if an application is busy with some operations it will skip IPC with that process and move on to next process.

Moreover, the program needs some bandwidth to fetch advertisements from advertisement server; this might also degrade the user online experience. To tackle this problem the program will fetch advertisements from advertisement server when the network connection is idle, till that time it will keep displaying static advertisements pre-fetched from the server. If the operating system is Microsoft Windows then the program can have the 20% reserved bandwidth of windows system for its operations as that reserved bandwidth is hardly used.

The program will also fetch advertisements on the basis of data fetched by desktop crawling. The system might degrade because of such a task, hence prioritization and scheduling of this task also needs to be incorporated in this solution. The system will crawl the hard drive and maintain an index of each files and data inside them and will fetch advertisements on the data fetched. The database of the files and data will be optimized to save disk space and encrypted for privacy.

Hence, the concept of prioritization and scheduling of these tasks will be an important part of the solution to keep up the system performance as a whole and also delivering relevant advertisements at user side.

7. Mechanism used for desktop advertisement

Our desktop advertisement mechanisms have two options to fetch advertisements from the server. The first being using any third party advertisements like Google Adsense API and second starting our own advertisement server and having advertisers get attracted towards it.

7.1. Using third party advertisement server or API:

We can go for third party advertisement server to fetch advertisements which enables us to programmatically generate ad code snippets for insertion into our program. We can also filter out unwanted advertisements and generate detailed reports about click through performance. To access accounts programmatically, we can build web service clients that connect to one or more of the third party web services. We will have user controls for interactively setting the various fields, properties and operations, with immediate visual previews of the advertisements and search boxes we are producing.

7.2. Starting own advertisement server:

Another option can be starting our own advertisement server [4] to get advertisements. The following points needs to be incorporated for the success of the same.

7.2.1. Choice of pricing and allocation scheme:

It has been seen that the allocation and pricing scheme considerably affects the revenue generated by the advertisements. Hence, we will examine basic theoretical and experimental results

comparing it with Google Adsense and Yahoo advertisement mechanisms as well as first and second price auction schemes. This would help us to decide the pricing scheme of our software.

7.2.2. Click fraud:
Most of the advertisement systems uses concept of pay per click i.e. they sell online advertisements on a per-click basis. This invites substantial opportunity for people to do fraud and inflate the advertiser's budget. To handle click frauds system will keeps track of the IP address, MAC address and even cookies of user system to identity any click fraud. If multiple clicks are occurring for same system, the system will ban that user off the system. The user can also try to fool the system by using proxy servers or by other schemes. We will deal with various methods proposed and studied by researchers for dealing with click-fraud.

7.2.3 Budget constraints:
In order to limit risk of advertisers, our advertisement system will permit advertisers to define a budget constraint [5] for them which the system will keep in database and set the frequency and number of times that advertisement is displayed in a day / month to suit advertisers budget needs. This way advertiser can manage his budget needs. Another point here is that different keywords would have different values. High paying keywords would require advertisers to pay more to book a space for that keyword. The keyword values would be decided on various market factors and algorithms.

7.2.4. Online nature of auctions:
The real-time nature of these advertisements poses some challenges [6] for the system as the keywords arrive in an unknown sequence [7], with an unknown number of each. Advertisement space must be property allocated to advertisers with available budget. It should not be such that an advertisement is displayed above the needed limit and some advertisements starve for space. Researchers have addressed these issues in several simplified scenarios. Moreover, reaching good quality of relevance of advertisement is also necessary in advertisement business which would be a challenging task to accomplish.

8. Privacy Issues
Privacy issues may be saddled with following concerns

8.1 Customizing search areas and filters:
A user may want to block certain category of files for example *.avi or *.doc from being monitored by the client. The proposed solution will provide for provision of file type, file location, file name, keyword based filter to ensure user privacy.

8.2 Disable browsing data collection:
The solution will also have option to collect anonymous data collected from browsing history of the user. Keeping in mine the privacy implications, the solution will support feature to disable this feature.

8.3 Encrypted index database:

Many users might have privacy implications for the index database of files being maintained by the client software. The user can encrypt the index in order to secure his privacy of his / her data.

8.4 Suspending the program operations:

A user may want to shut down the client for a specific duration when he is doing memory intensive work or work which he may not wish to be shared with the client. He will have the feature to stop the client to be restarted later manually or set a timer for it to be started automatically.

9. Misuse:

9.1 Mock Usage:

A person may keep his machine on with random application on. To prevent such usage from being billed there will be periodic checks to check the presence of human on machine through character recognition codes or other methods. This will prevent frauds and ensure value for advertiser.

9.2 Click Frauds:

User might use automated bots and scripts to click on advertisements even when he is not actually seeing them. This might lead to inflation of cost at advertiser's end. Hence, click frauds will be checked and when found the service for that applicant will be terminated or he will be warned about the same.

9.3 Forcefully getting high paying advertisements:

Each keyword would have it's own worth as per how much advertiser is paying for that keyword. A user might have a system with data and processes running related to a high paying keyword only in order to get only high paying advertisements. This problem will be taken care of and checked if the client system is not "Made for earning only".

10. Conclusion:

Online advertising is a booming industry and is developing into a major advertising outlet for millions of firms. Our solution takes the online advertisement to a new level close to user in form of desktop advertisement mechanism. User will be in touch of advertisements (relevant to his/her interest) at his own desktop. Our desktop advertisement solution is new of its kind which involves the concept of inter process communication, prioritization and desktop crawling to give user the most relevant advertisements and that to without lowering the performance.

This solution will help both advertisers and users. The advertisers will be able to market their solution to wide users and most of the users will be interested in the advertisements displayed as they would be fetched based on user's data. Moreover, the user will earn on clicking advertisements and he will be exposed to various products related to him. This way the solution will help the society and will revolutionize the advertisement industry.

We can tie up with third party software vendors (like office suites, media players, web browsers, chat clients) so that our software can be bundled along with their applications. The use of their software shall automatically trigger our application without hindering users' experience.

References

[1]   Craig E. Wills, "Process synchronization and IPC" ACM Computing Surveys (CSUR); Volume 28, Issue 1 (March 1996), Pages: 209 – 211, Year of Publication: 1996, ISSN: 0360-0300

[2] U. Ramachandran, M. Solomon, M. Vernon "Hardware support for interprocess communication" Proceedings of the 14th annual international symposium on Computer architecture. Pittsburgh, Pennsylvania, United States. Pages: 178 - 188. Year of Publication: 1987 ISBN 0-8186-0776-9

[3] Yoav Etsion,Dan Tsafrir,Dror G. Feitelson, "Desktop scheduling: how can we know what the user wants?", Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video , Cork, Ireland, Pages: 110 - 115,Year of Publication: 2004,ISBN:1-58113-801-6

[4] Yung-Ming Li, Jhih-Hua Jhang-Li, Yi-Lin Lee, "Efficiency analysis for display ads and contextual search", Proceedings of the ninth international conference on Electronic commerce, ACM International Conference Proceeding Series; Vol. 258,Pages: 361 - 368,Year of Publication: 2007, ISBN:978-1-59593-700-1

[5] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, Mohammad Mahdian "Dynamics of bid optimization in online advertisement auctions" ; International World Wide Web Conference Proceedings of the 16th international conference on World Wide Web Banff, Alberta, Canada ; SESSION: Advertisements & click estimates ; Year of Publication: 2007 ; ISBN:978-1-59593-654-7

[6] Ben Edelman, Michael Schwarz, "Internet advertising and optimal auction design", Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, Las Vegas, Nevada, USA ,Pages 1-1, Year of Publication: 2008 ISBN: 978-1-60558-193-4

[7] Mohammad Mahdian, Hamid Nazerzadeh, Amin Saberi, "Allocating online advertisement space with unreliable estimates" ; Electronic Commerce Proceedings of the 8th ACM conference on Electronic commerce San Diego, California, USA ; Year of Publication: 2007 ISBN:978-1-59593-653-0