

# Web Operating System impediments

Rajesh Sachdeva<sup>1</sup>, Preeti Sharma<sup>2</sup>, Neha Kataria<sup>3</sup>

AP, Deptt of Computer Science, Dev Samaj College For Women, Ferozepur City, India<sup>1</sup>

AP, Deptt of Computer Science, Dev Samaj College For Women, Ferozepur City, India<sup>2</sup>

Research Scholar, Ferozepur, India<sup>3</sup>

**Abstract:** WebOS is a newly emerging Operating System, accessible via a browser, with multiple integrated built-in applications that allow the user to easily manage and organize her/his data from any location. You can use your desktop as a virtual desktop on the web. In the wide area, application developers are forced to build abstractions themselves. This ad-hoc approach wastes programmer effort and system resources. To address these problems, WebOS provides basic operating systems services needed to build applications that are geographically distributed, highly available, incrementally scalable, and dynamically reconfiguring. This paper starts with an introduction of Webs and its benefits, but it also have some flaws related to security.

We have reviewed some most interesting WebOS available nowadays and tried to provide a detailed description of their features. Experience with a number of applications developed under WebOS indicates that it simplifies system development and improves resource utilization.

**Keyword:** Abstraction, Ajax, iCloud, Phishing.

## I. INTRODUCTION

Development of a new single operating system enabling global computing is a burning topic now a days. Such an operating system is called the Web Operating System, or WOS. Web Operating System is an internet service through which a user can access his computer data remotely anywhere on any computer and in any part of earth where internet is available. It has been called as Web Operating System because they are present on the web and not on the computer of the user, all the data is being stored on the servers of the Web OS provider. It was an introduction of one more operating system in the category of web operating systems. Web OS has capacity of storing large amount of user data more than 50 GB It provides a very simplified and user friendly environment. User can share files, listen music, read news and can do many other stuff depending upon the diversity and richness of the web OS used by the user. So, web OS is computer on the internet. Today, although the World Wide Web has made geographically distributed read-only data easy to use, geographically distributed computing resources are not. We communicate with the computer through the user interface which is right now before your eyes. If we want to share any data, internet is engaged. Sharing can be done through many websites available on the web. So this is how a local computer with a normal user works. Now consider yourself as a computer user right now working at office. Suddenly your boss calls you to show some random file to him. But you forgot to bring it. The file is in the hard disk of your home PC and it could not be teleported from there to your office computer in air. This is the situation when most brain thinks if there could be any method by which they could access their local content anywhere. This problem was resolved by the programmers by introducing the concept of Web OS. WebOS are the dynamic computers. With the increasing use of high-speed Internet technologies during the past few years, the concept of cloud computing has become more popular. In

cloud computing, users work with Web-based, rather than local, storage and software. These applications are accessible via a browser and look and act like desktop programs. With this approach, users can work with their applications from multiple computers. In addition, organizations can more easily control corporate data and reduce malware infections. Also, cloud computing makes collaboration easier and can reduce platform-incompatibility problems. Now, a growing number of organizations are adding to the cloud concept by releasing commercial and open source Web-based operating systems. While the idea isn't new, the proliferation of users and applications distributed over the Web, including those at scattered corporate sites, has made it more interesting, relevant, and, vendors hope, commercially viable. It also includes many of a traditional OS's capabilities, including a file system, file management, and productivity and communications applications. As is the case with Web-based applications, the Web OS functions across platforms from any device with Internet access. The applications, hard disk, operating systems are all present at the servers from where they are operated. The web OS service provider has different spaces for application access and database. The user is provided with a graphical user interface which feels like the one at your PC. This operating system consists of application section like calendar, clock, calculator, document editors etc. then there is a section for data storage where user can store data, and there are many other sections depending upon the web OS. Whatever content user wants to store is stored at the hard disk at servers. As the terminology itself says, the web OS make use of the web to connect and upload files to the client server.

The result is that wide area applications that require access to remote CPU cycles, memory, or disk must be programmed in an ad hoc and application-specific.

## II. TECHNOLOGY USED

The WEB OS is fully written in Javascript, exclusively using DHTML techniques to run in any web browser without any additional software installation needed. The system implements a modern kind of web application model, excessively using Ajax for communicating between client components and the Java server backend in an exclusively asynchronous manner. Aim is to offer users the unique interaction behavior following the desktop metaphor, which is the main idea of any web desktop. Also typical for this sort of web application is the broadly use of Javascript-on-demand techniques, cutting the complete project source into pieces and loading them instantly when needed.

An operating system (OS) is a special kind of program that organizes and controls computer hardware and software. Operating systems interact directly with computer hardware and serve as a platform for other applications. Whether it's Windows, Linux, UNIX or Mac OS X, your computer depends on its OS to function.

That's why some people object to the term Web OS. A Web OS is a user interface (UI) that allows people to access applications stored completely or in part on the Web. It might mimic the user interface of traditional computer operating systems like Windows, but it doesn't interact directly with the computer's hardware. The user must still have a traditional OS on his or her computer.

With so many different Web operating systems either currently available or in development, it should come as no surprise that programmers use different approaches to achieve the same effect. While the goal of a Web OS is to provide an experience similar to using a desktop OS, there are no hard and fast rules for how to make that happen. The two most popular approaches rely on Flash technologies or Asynchronous JavaScript and XML (AJAX) technologies.

**FLASH TECHNOLOGY:** Flash is a set of technologies that enable programmers to create interactive Web pages. It's a technology that uses vector graphics. Vector graphics record image data as a collection of shapes and lines rather than individual pixels, which allows computers to load Flash images and animation faster than pixel-based graphics.

Flash files stream over the Internet, which means the end user accessing the file doesn't have to wait for the entire file to download to his or her computer before accessing parts of it. With Flash-based programs like YouTube's video player, this means you can start watching a film clip without having to download it first.

More than 98 percent of all computers connected to the Internet have a Flash player installed that makes flash an attractive approach for many programmers. They can create a Web OS knowing that the vast majority of computer users will be able to access it without having to download additional software.

**ASYNCHOROUS JAVASCRIPT and XML TECHNOLOGY:** AJAX technologies rely on hypertext markup language, the JavaScript programming language,

Cascading Style Sheets and extensible Markup Language. It's a browser technology. The HTML language is a collection of markup tags programmers use on text files that tell Web browsers how to display the text file as a Web page. CSS is a tool that gives programmers more options when modifying a Web site's appearance. Programmers can create a style sheet with certain attributes such as font style and color, and then apply those styles across several Web pages at once. JavaScript is a programming language that allows applications to send information back and forth between servers and browsers. XML is a markup language, which means programmers use it to describe the structure of information within a file and how it relates to other information.

The asynchronous aspect of AJAX means that AJAX applications transfer data between servers and browsers in small bits of information as needed. The alternative is to send an entire Web page to the browser every time something changes, which would significantly slow down the user's experience. With sufficient skill and knowledge, a programmer can create an AJAX application with the same functions as a desktop application.

Like Flash, most computers can run AJAX applications. That's because AJAX isn't a new programming language but rather a way to use established Web standards to create new applications. As long as an application programmer includes the right information in an application's code, it should run fine on any major Web browser. Some well known Web applications based on AJAX include Google Calendar and Gmail.

## III. WORKING OF WEB OS

As the Web evolves, people invent new words to describe its features and applications. Sometimes, a term gains widespread acceptance even if some people believe it's misleading or inaccurate. Such is the case with Web operating systems.

An operating system OS is a special kind of program that organizes and controls computer hardware and software. Operating systems interact directly with computer hardware and serve as a platform for other applications. Whether it's Windows, Linux, UNIX or Mac OS X, your computer depends on its OS to function.

A Web OS is a user interface that allows people to access applications stored completely or in part on the Web. It might mimic the user interface of traditional computer operating systems like Windows, but it doesn't interact directly with the computer's hardware. The user must still have a traditional OS on his or her computer.

While there aren't many computer operating systems to choose from, the same can't be said of Web operating systems. There are dozens of Web operating systems available. Every WOS user should be able to share his or her local resources with other users. In addition, users should be able to combine and use different resources for interactive problem solving.

A lot of communication is needed to find these resources. Thus strategies for communication and searching are necessary. Two strategies are available to use:

### The broadcast strategy

The requesting machine submits the request to each machine in the list. Each of these machines then sends messages back to the requesting machine. Since these machines can almost work in parallel the answer will be quickly available on the requesting machine. If the list contains  $n$  machines,  $2n$  messages will be generated.  $n$  messages from requesting machine and  $n$  answers both positive and negative. Thus the network load is high. Furthermore, the broadcast implementation must be realized, hence delaying data transmission.

### The serial request strategy

In this case the requesting machine sends one message containing the list of the remaining machines to one of the machines from the list. If the service is available on this machine, a positive answer is directly sent back to the requesting machine. So the generated network load is much less than in the first case. On the other hand, the respond time is much higher than in the first case and any communication problems or long transfer times directly influence the respond time.

## IV. APPLICATION OF WEB OS

This section provides various applications built using the WEB OS. The application demonstrate that WEB OS enable and simplify their implementation.

### Internet Chat

Internet chat allows for individuals to enter and leave chat rooms to converse with others co-located in the same logical room. In our implementation, chat rooms are modeled as WebFS files accessed by Smart Clients. The file system interface is well-matched to chat semantics in a number of ways:

- A simple file append abstracts the required network communication necessary to send messages.
- The chat file provides a persistent log of chat activity.
- Access control lists allow for private and secure (through WebFs encryption) chat rooms. For scalability, we allow multiple WebFS servers to handle client requests for a single file (room). Each WebFS server accumulates updates, and periodically propagates the updates to other servers in the WebFS group, who in turn transmit the updates to local clients. Smart Clients choose the least loaded WebFS server for load balancing and connect to alternative servers on host failure or network partition for fault Transparency.

To quantify the benefits available from the WebOS framework, we implemented two versions of chat with identical semantics, both with and without WebOS. The main reason for this savings in complexity was the replacement of separate code for managing communication and persistent storage of chat room contents with a single file. As an added benefit, the WebFS interface becomes available for similarly structured distributed applications. For example, we are currently implementing a shared distributed whiteboard application using this interface to both validate our design

and to provide an immediate benefit to the Internet by doing more intelligent caching of Web content.

### Internet Weather

A number of sites are currently attempting to provide regular updates of congestion, latency, and partitions in the Internet. Such information is invaluable for services making placement and load balancing decisions. However, all current efforts take network measurements from a centralized site, making it difficult to measure network characteristics between two arbitrary sites. We are addressing this limitation by using the WebOS framework to generate more comprehensive snapshots of Internet conditions. In our implementation, a centralized server provides Smart Client applets for those wishing to view the current Internet weather. In exchange for the weather report, the user implicitly agrees to allow the applet to execute trace route to a subset of server-determined sites and to transmit the result back to the server. Using these results from multiple sites, the service is able to construct fairly comprehensive snapshots of Internet weather.

## V. SECURITY CHALLENGES TO WEB OS

Many things working against the security of a Web system are unavoidable facts of the Internet. Therefore, a few fundamental principles must be understood to successfully approach the task of securing the Web system.

**The Internet and the machines that connect to it are inherently insecure.** The Internet was not built on a secure foundation. For much of the Internet's history, data travelled between hosts in an unencrypted, plaintext form. This data could readily be intercepted and read by a third party, potentially compromising passwords or important content. For electronic commerce, transmitting information in plaintext format was clearly not acceptable, given the transmission of private, financial, and other sensitive data. With the advent of security protocols, such as the Secure Sockets Layer (SSL), this problem of insecure data transmission has been largely remedied since data can now travel over the Internet in an encrypted form. Nonetheless, other Internet security concerns remain.

The modern Internet was built primarily on the use of servers running the UNIX operating system. Although some security controls were necessary to enable the Internet's use as a multiuser system, it was not long before holes were discovered that allowed unscrupulous users to penetrate system security. Although the security of UNIX systems has been vastly improved over the years, and other secure operating systems, such as Windows NT/2000, have been developed, the host security problem still exists. In fact, as more users and computers continue to be connected to the Internet and as additional vulnerabilities are discovered and exploited, the host security problem may never be completely resolved.

**Web sites are continuously being scanned by attackers probing for potential entry points.** On the Internet, no Web systems are exempt from examination by would-be intruders. One must assume that no host is safe and that all security mechanisms must be in place and active at all times to prevent intrusion. Any door left open has a high degree of probability of being exploited by an attacker. Even small security holes can lead to an attacker ultimately gaining administrative access to the system.

**All server software and operating systems are inherently insecure.** It has been said that the only secure computer is one that is physically inaccessible and not connected to any network. New security flaws are discovered nearly every day, even within the latest, most secured operating systems and software. Therefore, it must be assumed that a Web system's services and software are not without security flaws and are at risk of being compromised at any time.

**Even the most thorough security preparations may miss a security flaw.** Although this sounds quite disheartening, the basic principle that applies to Web system deployment is that the system cannot rely on any single security mechanism. For example, simply having a firewall to block unwanted traffic is not enough to ensure the security of a Web system and its data. It must be assumed that a knowledgeable attacker will find a way to bypass the firewall and to gain access to a computer residing on an internal network. In view of such grim odds, it is clear that each aspect of the system must be considered from a security point of view and that any applicable, and practical, security measures must be applied. The odds of an attacker's defeating one mechanism are high; the odds of an attacker's defeating multiple different mechanisms, however, are much lower.

## VI. IMPEDIMENTS TO WEB OS

### Cloud storage:

While a browser-based OS offers plenty of benefits, it's also hampered by severe limitations. Most notably: everything is stored in the cloud. If you're working from the office or your home, that's generally not a concern. However if you travel, accessing a reliable and fast broadband connection can be tricky. Many areas have dead zones, limited coverage and inconsistent throughput rates.

### Constant Online Access:

Complicating matters further, many wireless ISPs impose a data cap on their mobile broadband service. A computer that requires constant online access to transmit data or stream music and video could hit those caps very quickly. It wouldn't be as troubling if you could work offline, but the majority of apps currently available for Chrome OS won't work without a broadband connection. This makes working while traveling difficult or, in some cases, impossible.

### Limited File Management

Other issues include the lack of proper VPN support, limited file management and some weird browser

compatibility issues that prevent some websites from loading or functioning correctly. And, while the Chrome App Store offers a wide variety of apps, it's still rather limited. As a result, finding what you need can sometimes prove difficult.

### Security

Security is another major computer concern, especially if you are storing some sensitive data. You are usually required to purchase some antivirus software and do regular tests of your drives to make sure there are no infected files. Google OS handles all the security for you so there is no need to waste time on that. The jury is still out on whether this is a pro or a con, though – if all your sensitive data is online and you play no part in keeping it safe.

### Hackers

When you have a web-based initiative, you also have to be on the lookout for hackers. Hackers could potentially hack into your servers and change your website. By doing this, the hackers can change the links on your website so that they are paid when sales are generated. They could also sabotage your site to negatively influence customer's buying decisions. Because of these threats, you will need to constantly monitor your website or web interface to ensure that hackers are not involved.

### Phishing Attacks

In some cases, phishing attacks can negatively impact your web-based initiatives. If customers need to login to your platform for some reason, it is possible for scam artists to use phishing as a way to get into their accounts. This type of attack involves a scam artist sending an email posing to be your business, in an attempt to get the login information. Once this happens, the scam artists have full access to your customer's accounts.

## VII. CONCLUSION

In this paper, we have demonstrated the collaboration available from exporting traditional operating system functionality to wide area applications. Our model implementation, WebOS, describes one possible organization of these system services. In this framework, we make the following contributions. First, we show that extending server functionality onto client machines allows for more flexible implementation of name resolution, load balancing, and fault tolerance. Second, by providing a file system abstraction combining communication and persistence, we simplify the implementation of a number of wide area applications. In Future you will work with many WebOS, We should think for the future design and risks involve to develop a next generation operating system. However, we wonder if the lack of offline capability might hinder its adoption since most of the time PC users take their machine overseas where Internet access may be spotty or expensive. Still, there is about many years to go to take full advantages of Web based operating systems. The conclusion is that a web-based operating system will imminent in Web 2.0's age, and we

should design new WebOS based on reviewed services that will give users the power of computing on the Web. No more hard-drive backups required – just turn on your browser and get going with these Web OS services after some years. Next, we present a methodology for coherently caching program results through the file system, speeding the performance of applications which must repeatedly execute programs with common inputs. Finally, we demonstrate how Rent-AServer, an application developed in our framework, both improves system performance and more efficiently utilizes system resources for Web server access.

### REFERENCES

- [1] “Transparent Result Caching,” Amin Vahdat and Thomas Anderson. June 1998. Proceedings of the 1998 USENIX Technical Conference.
- [2] Amazon Inc. Amazon Web Services EC2 site. <http://aws.amazon.com/ec2>, 2008.
- [3] “WebOS: Operating System Services For Wide Area Applications,” Amin Vahdat, Thomas Anderson, Michael Dahlin, David Culler, Eshwar Belani, Paul Eastham, and Chad Yoshikawa. July 1998. The Seventh IEEE Symposium on High Performance Distributed Computing
- [4] <https://dennosecqtinstien.wordpress.com/2012/04/10/a-detailed-discussion-over-pros-and-cons-of-web-based-operating-systems/>
- [5] <http://www.computerweekly.com/news/2240089111/Top-five-cloud-computing-security-issues>
- [6] <http://www.informit.com/articles/article.aspx?p=24480> Marshall, Gary (1 March 2013).
- [7] R.S. Cox, J.G. Hansen, S.D. Gribble, and H.M. Levy. A Safety-Oriented Platform for Web Applications. In Proc. IEEE Symposium on Security and Privacy, 2006.
- [8] C. Reis, B. Bershad, S. Gribble, and H. Levy. Using processes to improve the reliability of browserbased applications. In Under submission./.
- [9] Big WebOS roundup - 10 online operating systems reviewed. <http://franticindustries.com/blog/2006/12/21/big-webos-roundup-10-online-operating-systems-reviewed/>.