



# International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: [www.ijarcsse.com](http://www.ijarcsse.com)

## Midori-A Cloud Based Operating System

Varsha Shrivastava\*

Computer Science & Engineering  
India

**Abstract**— Cloud computing is the use of computing resources that are delivered as a service over internet. Cloud computing allows consumers and business to use application without installation and access their personal files at any computer without internet access. Microsoft Corporation is planning to introduce a new cloud based operating system named Midori. This operating system would be highly distributed, concurrent and virtualized. Midori's services such as storage would be provided across a "trusted distributed environment". Midori will store data on a central Microsoft Server and remove the storage service from the single PC. Since Midori will be designed with an asynchronous-only architecture and it will always assume that the user is online. The business world will be impacted by Midori because it will be more virtually-oriented than any other OS we have ever seen.

**Keywords**— Concurrency; Midori; M#; Singularity; Virtualization; Windows

### I. INTRODUCTION

Window is the name that has ruled the whole computer world since its launch by Microsoft Corporation in 20 November 1985. It is a trademark for Microsoft Corporation. With many advanced versions. Of Windows available today such as Windows XP, Windows Vista, Windows 7, Windows 8 it is the most used operating system in the world. But now the here is time to experience some another technology of operating system[1]. Microsoft is working on a new generation of operating system called Cloud-Based Operating System and rumours are there that Midori will be their first such operating system, which will replace Windows fully from computer map. The reason behind Midori is to develop a lightweight portable OS which can be matched easily to lots of various applications. This new OS will be based on the concept of virtualization and singularity.

### II. MIDORI OPERATING SYSTEM

- *Definition*

Midori is the code name for a managed code operating system being developed by Microsoft with joint effort of Microsoft Research. It is a singularly, concurrency control and virtualization operating system. It is fully dependable operating system in which the kernel, device drivers, and applications are all written in managed code [2]. The code name Midori was first discovered through the PowerPoint presentation CHESS: A systematic testing tool for concurrent software [3].

- *Features*

- A. *Virtualization*

Virtualization is a technology that abstract away the detail of physical hardware and provide virtualized resources for high level applications. It provides transparent migration of applications, server consolidation, enhanced system security and online operating system maintenance.

- B. *Singularity*

Singularity was an experimental operating system built by Microsoft Research between 2003 and 2010. It creates "Software-isolated processes" to reduce the dependencies between individual application, and between the application and the operating system itself. This concept enhances security by using micro kernel based architecture [4]. The component of Midori OS execute in the same address space, which contains "software-isolated processes" (SIPs). Each SIP has its own data and code layout, and is independent from other SIPs. These SIPs behave like normal processes, but avoid the cost of context-switches.

- C. *Concurrency*

Concurrency is the interleaving of processes in time to give the appearance of simultaneous execution. With help of Concurrency a person can do multiple tasks. Concurrency allows multiple applications to share resources in such a way that distributed applications appear to run at a same time.

### III. MIDORI DESIGN METHODOLOGY

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

Midori OS design is based on some modules like:-

- A. Microkernel Architecture

- B. Software Isolated Processes(SIP)
- C. Contract Based Channels
- D. Metadata Infrastructure

A. *Microkernel Architecture*

Kernel is the most important part of an operating system. It acts as an interface between user and computer hardware. This is only part of an OS that can direct interaction with the computer hardware. It connects the application software to the hardware of a computer. Midori uses Microkernel architecture which is the near-minimum amount of software that can provide the mechanisms needed to implement an operating system. These mechanisms include low-level address space management, thread management, and inter-process communication (I.P.C). As an operating system design approach, microkernel permit typical operating system services, such as device drivers, protocol stacks, file systems and user interface code, to run in user space. If the hardware provides multiple rings or CPU modes, the microkernel is the only software executing at the most privileged level (generally referred to as supervisor or kernel mode).

TABLE I Monolithic vs. Micro kernel

Factors	Monolithic Kernel	Micro Kernel
Size	Huge	Small
IPC	Signal/Sockets	Message Queues
Security	System-wide halt	Local process hale
Correctness	Hard to ensure	Easier to ensure
I/O Communication	Fully integration	Message-per-IRQ

B. *Software Isolated Processes*

This methodology is to reduce the dependencies between individual application, and the application and the OS itself. It enhance security and provide backward compatibility, thus it can execute codes in case of any device drivers' failure.

TABLE II Multiple Address Space vs. Single Address Space

Multiple address space (Current Systems)	Single Address Space
Required context switch for IPC	Efficient IPC through Exchange Heap
Processor state must be saved and restored	Loses memory hardware based
TLB must be flushed	Protection mechanisms

TABLE III Inexpensive in terms of CPU cycles

Systems	API Call	Thread Yield	Message Ping/Pong	Create Process
Midori	91	346	803	352,873
Free BSD	878	911	13,304	1,032,254
Linux	437	906	5,797	719,447
Windows	627	753	6,344	5,375,735

C. *Contract Based Channels*

Contract Based Channels provide the way of communication between Software Isolated processes. These are the bi-directional channels and provide reliability. In this, each point has its own queue and belongs to exactly one thread at a time

D. *Metadata Infrastructure*

This is used to describe program's resources, capabilities and responsibilities.

IV. COMPUTER LANGUAGE USED IN DEVELOPMENT OF MIDORI

M# (M sharp) is a language to develop Midori Operating system. It is an extension of Microsoft's C# language. This new language "grew out of Sing#, the system language of Microsoft Research's Singularity OS. M# language acts as a Code generator and translates entities and page definitions to ASP.NET and C# code which in turn form the User interface and Business logic layer of the application.

*Design Goal- According to its official website [7] these are the motivations behind creating M#:*

- A. Reducing time of development
- B. High quality (beautiful, documented) code
- C. Reduced human error (hence the code generator nature of it)

## V. WHAT'S NEXT FOR MIDORI?

While Midori was in incubation, the Microsoft Research team working on the "Drawbridge" library OS managed to support Midori as a host implementation, alongside a number of other Microsoft operating-system platforms. (A library OS is a form of virtualization that seeks to replace the need for a virtual machine to run software across disparate platforms.)

## VI. MIDORI AS A WEB BROWSER

Midori also act as a web browser in any operating system, but with fact is that Midori OS is separate from Midori web browser. As a web browser, its aim is to be fast and lightweight. It uses the web kit rendering engine. Midori has 14 extensions that are installed by default that can be activated through the side panel [5].

*Features of Midori Web browser-*

- A. Midori handles all the latest web technologies like HTML5 and CSS3.
- B. It supports all facilities like Listen to music on Radio, play a game of Angry Birds.
- C. The lightweight web kit rendering engine lends it the same supremacy in speed that has spurred the growth of Google Chrome.
- D. Midori's source code is freely available for anyone to download and modify, leading to active development and a diverse feature set.

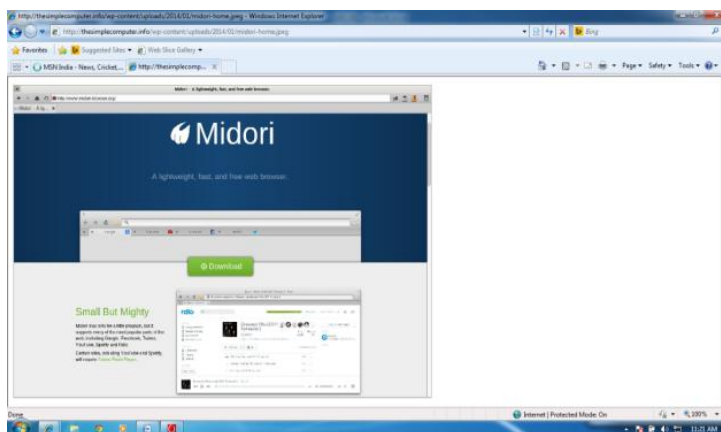


Fig. 1 Freely Available Midori Web Browser

## VII. CONCLUSION

Midori is the next generation, non-windows operating system by Microsoft Corporation. It will be having cloud computing where application components exist in data centers. It will contain two kernel layers. In future, Midori will be used over the web, where applications reside over the centralized server [6]. Midori will focus on a rapidly growing field of cloud computing or the movement to shift hardware and software, particularly storage, out of home PCs and into computing clusters. Midori will be built upon an asynchronous-only architecture that is built for task concurrency and parallel use of local and distributed resources. This will help it manage various hardware and software resources over the net. It will also feature a distributed component-based and data-driven application model, and dynamic management of power and other resources.

## REFERENCES

1. Mary-Jo Foley(2008-06-30). "Goodbye, XP. Hello, Midori". Retrieved 2008-07-22. [1]
2. Marius Oiaga (2008-06-30). "Life After Windows - Microsoft Midori Operating System". Retrieved 2008-07-22[2]
3. Madanlal Musuvathi; Shaz Qadeer; Thomas Ball (November 2007). "CHESS: A systematic testing tool for concurrent software". Retrieved 2008-07-22.[3]
4. "Singularity RDK".Retrieved 2012-01-22.[4]
5. David Worthington Microsoft maps out migration from Windows SD Times: Software development new July 31, 2008.[5]
6. Elizabeth Montabano (2008-07-29). "Microsoft prepares for end of Windows with Midori" Retrieved 2009-11-28.[6]
7. "Why M#?". Retrieved 18 May 2013.[7]