Rich Internet Application (RIAs) Survey

MR.TEERANITI KHAMCHUM

MR. RUJ URAPHAMANAN

MR.SOMLERK LEELANIPHON

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE (INTERNATIONAL PROGRAMME)

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2009

Special Project Title Rich internet application (RIAs) Survey

Name	Mr. Teeraniti Khamchum	ID: 49050385
	Mr. Ruj Uraphananan	ID: 49050402
	Mr. Somlerk Leelaniphon	ID: 49050405

Degree	Bachelor of Science
Major Program	Computer Science (International Programme)
Academic Year	2009
Special Project Advisor	Assistant Professor Dr. Sarun Intakosum

Abstract

This survey will help users or developers to choose what RIAs tool between JavaFX, Adobe Flex, and Microsoft Silverlight are the best suit for their project or requirement. By comparing the calculation power and CPU process through a RIAs application benchmark called "Bubblemark test". The test will be done in different environment such as different internet browser and computer system. The result will be a reference for comparing the performance and use for further research in RIAs tool.

Keywords : Rich internet application tools, Benchmark, Internet application

Acknowledgement

The authors would like to sincerely express their highly gratefulness to the advisor Asst.Prof.Dr. Sarun Intakosum for his invaluable guidance, useful advices, kind and constructive criticism, consistent inspiration and encouragement throughout this research.

They greatly appreciate all teachers who have invaluable knowledge while studying in the Faculty of Science, King Mongkut's Institute of Technology Ladkrabang.

Special thanks go to all students in International Program at Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, who always provide the encouragement and collaboration along this study.

Finally, they would like to express deep thanks to their parents and numerous friends for their encouragement and love.

Mr.Teeraniti Khamchum

Mr.Ruj Uraphananan

Mr.Somlerk Leelaniphon

Table of Contents

Page

Table of Contents (Cont.)

page

Table of Contents (Cont.)

page

Table of Contents (Cont.)

page

List of Figures

Figure **Page**

List of Figures (Cont.)

Figure **Page**

List of Figures (Cont.)

Figure **Page**

Chapter 1

Introduction

1.1 Rationale

Rich internet application (shortly call as RIA) is an Internet application which has abilities of Desktop application that work on Client-side on user computer but stored data on Application server. Normally RIA will run on website that doesn't require anymore software installation or work on user computer as a Sandbox in the reason of security. Today there are many internet application which can developed an RIA such as AJAX, ActiveX, JavaFX, Adobe FLEX, Adobe AIR, Microsoft Silverlight and many other. Every language has their own advantages and features, and almost all of them are open for developers to use and create their own application. Each of them has different good and bad features. The problem is which language are suit for the developer or user application requirement.

Among the mention language there are 3 main that have been widely accepted in internet application developers, which is JavaFX from Sun Microsystems, Microsoft Silverlight from Microsoft Corporation and Adobe FLEX form Adobe. All of them have their strong point and different way of use. So in this project we will survey, compare, and analyze the strong point and the weak point of these tools to help the internet application developers consider which one is the best suit for them.

1.2 Objective

For study the performance of Rich Internet Application, giving a review of rich internet application programming language and also, find out which RIA tools are the best in different environment such as Internet browser or computer system. We will compare the performance in calculation and CPU resource management in the RIAs we choose. We hope this survey will help developers to decide which tools are the best suits for them.

1.3 Scope of study

This survey will test the performance in RIAs tools by using the RIAs benchmark tool called "Bubblemark" which can provide a result of CPU process and calculation power in each tool. We test the RIAs tool with 5 different internet browsers which is Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari and Opera. The test will be run on 3 types of computer systems that have different performance.

1.4 Research Methodology

1. Study Adobe FLEX, Microsoft Silverlight and JavaFX tools.
2. Test Bubblemark applications in different environments such as different internet browser or different computer systems.
3. Compare result from each RIAs tool benchmark result.
4. Create a data table from testing result and develop a graph result.
5. Summarize result.

1.5 Benefit

This survey is made to answers the question about what is the best RIAs tool in terms of calculation performance and CPU resource management. Users of this survey such as RIAs developers or normal users who interest in RIAs tool will get the idea how to choose the best RIAs tool which suits their requirement.

Chapter 2

Related Literature

2.1 Rich Internet Applications (RIAs)

Rich Internet Applications (RIAs) are the next generation of Internet applications, mixing between the feature of the web application and the feature of the desktop application to give features, performance and flexibility, but are delivered over the web.(figure 2.1) These applications combine the feature of both application – the power and rich content of the desktop with the connectivity of the internet.

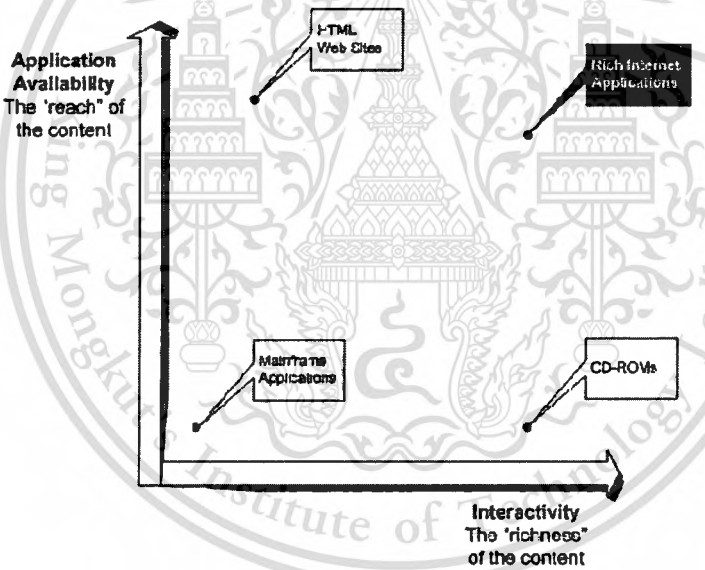


Figure 2.1 : General concept of RIAs "Reach and Richness", Andrea Simmons. "RIA101 : A Primer for Marketing Agencies and Multimedia Developer"

RIAs offer the best mix of "reach" in application availability and "richness" in their content. Since they are web-based and cross platform, RIAs can run on any system or platform. Their efficiencies in processing at both the client-level and server-level allow for the delivery of richer and more interactive content.

With an HTML website, figure 2.2 show when a user fills in data, changes options, or checks boxes and hits, the page must be submitted to the server for data validation and then the screen is reloaded with the new data incorporated.

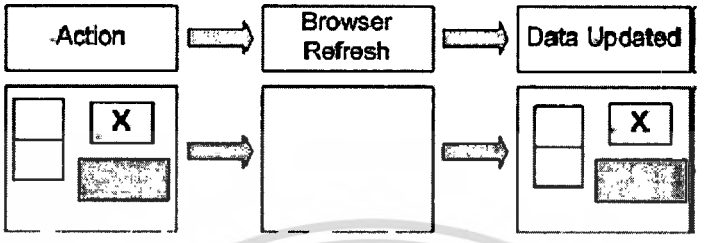


Figure 2.2: Basic idea of HTML structure

With RIAs, figure 2.3 show the data is partially processed by the client, so a full page refresh and return trip to the server are not necessary with every action performed. This means that when users click boxes, fill in forms and modify preferences, the results can be displayed on screen immediately without the data being reloaded.

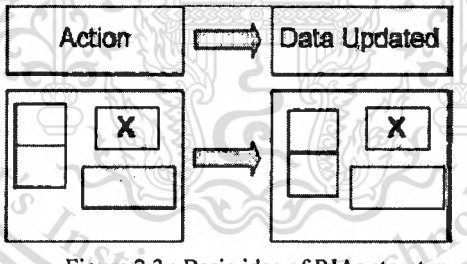


Figure 2.3 : Basic idea of RIAs structure

This responsiveness and smooth interaction creates new possibilities in information flow for the user and new methods of information display for your clients. Web-enabled applications no longer have to present information in a rigid page-by-page format. The user experience is no longer limited by a client's web browser as the shared interaction between the server and the client delivers vast improvements in usability and productivity over today's web sites.

2.2 Microsoft Silverlight

Microsoft Silverlight is a new cross-browser, cross-platform implementation of the .NET Framework for building and delivering the next generation of media experiences and Rich Interactive Applications(RIA) for the web. It is combination of different technologies into a single development platform that allows you to select tools and the programming language you want to use. Now they have 3 versions and feature of that version.

2.2.1 Silverlight 1.0

It delivers high performance multimedia and animation capabilities that can blend seamlessly with HTML. It's capable of playing a variety of audio and video file formats, such as MP3, WMA, and WMV. It handles streaming quite well, so media can start playing immediately without having to wait for the entire file to download. (see example of Silverlight 1.0 on figure 2.4)

2.2.2 Main features of Silverlight 1.0 :

- Built-in codec support for playing VC-1 and WMV video, and MP3 and WMA audio within a browser.
- Silverlight supports the ability to progressively download and play media content from any web-server.
- Silverlight also optionally supports built-in media streaming.
- Silverlight enables you to create rich UI and animations, and blend vector graphics with HTML to create compelling content experiences.
- Silverlight makes it easy to build rich video player interactive experiences.

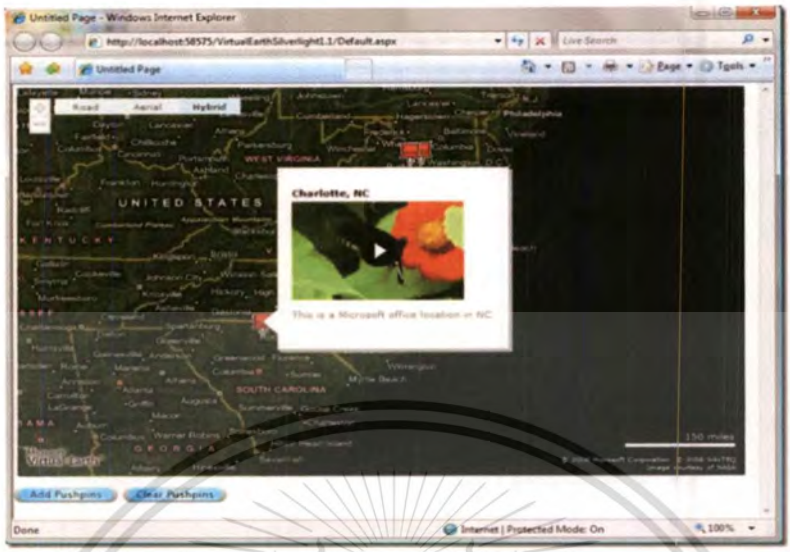


Figure 2.4 : A Silverlight 1.0. HTML controls overlaid on top of Silverlight content

2.2.3 Silverlight 2

the built-in cross platform subset of the .NET Framework. While you can still mix in as much (or as little) JavaScript as you'd like, you'll now have the option of running your own fully compiled .NET code within IE, Firefox, or Safari. Those developers who hate JavaScript should be thrilled to know they can now write their client side code in C#, VB.NET, or any other .NET language. You'll now be able to replace slow performing JavaScript code with fully compiled .NET code that could easily execute hundreds of times faster. (see example of Silverlight 2.0 on figure 2.5 and Silverlight 2 Architecture on figure 2.6)

2.2.4 Main features of Silverlight 2.0 :

- A built-in CLR engine that delivers a super high performance execution environment for the browser. Silverlight uses the same core CLR engine that we ship with the full .NET Framework.
- Silverlight includes a rich framework library of built-in classes that you can use to develop browser-based applications.
- Silverlight includes support for a WPF UI programming model.
- Silverlight provides a managed HTML DOM API that enables you to program the HTML of a browser using any .NET language.
- Silverlight doesn't require ASP.NET to be used on the backend web-server (meaning you could use Silverlight with PHP on Linux if you wanted to).
- Silverlight 2 includes Deep Zoom, with smooth transitions, using the mouse wheel. The images can scale from 2 or 3 megapixels in resolution into the gigapixel range, but the user need not wait for it to be downloaded entirely; rather, Silverlight downloads only the parts in view, optimized for the zoom level being viewed.(see example on figure 2.7)
- Silverlight 2 also allows limited filesystem access to Silverlight applications. It can use the operating system's native file dialog box to browse to any file (to which the user has access).

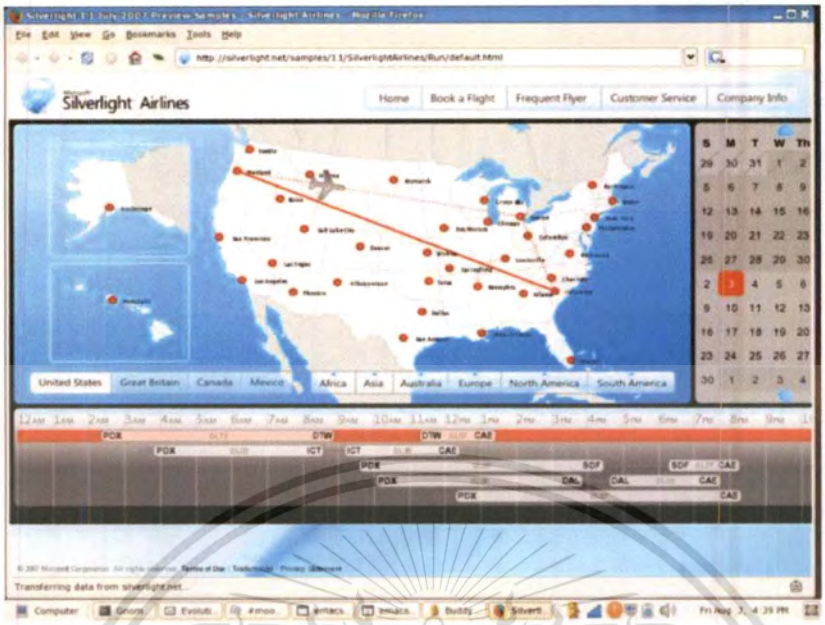


Figure 2.5 : A Silverlight 2 support for .NET languages and development tools.

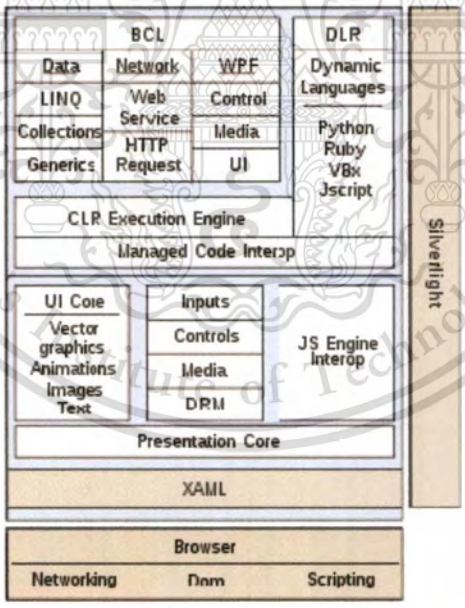


Figure 2.6 : Silverlight 2 Architecture.

Figure 2.7 : A Silverlight 2 application using Deep Zoom

2.2.5 Silverlight 3

Fully supported by Visual Studio and Expression Blend, highlights of new features and functionality of Silverlight 3 include: major media enhancements, out of browser support allowing Web applications to work on the desktop (see example on figure 2.8, figure 2.9); significant graphics improvements including 3D graphics support, GPU acceleration and H.264 video support; and many features to improve RIA development productivity. Also, in order to fully integrate all the .NET developer tools, Visual Studio 2010 will support a fully editable and interactive designer for Silverlight.

2.2.6 Main features of Silverlight 3.0 :

- Highest Quality Video Experience : prepare to see **2.2 Microsoft Silverlight**
- Microsoft Silverlight is a new cross-browser, cross-platform implementation of the .NET Framework for building and delivering the next generation of media experiences and Rich Interactive Applications(RIA) for the web. It is combination of different technologies into a single development platform that allows you to select tools and the programming language you want to use. Now they have 3 versions and feature of that version.

2.2.1 Silverlight 1.0

This tool delivers high performance multimedia and animation capabilities that can blend seamlessly with HTML. It's capable of playing a variety of audio and video file formats, such as MP3, WMA, and WMV. It handles streaming quite well, so media can start playing immediately without having to wait for the entire file to download. (see example of Silverlight 1.0 on figure 2.4)

2.2.2 Main features of Silverlight 1.0 :

- Built-in codec support for playing VC-1 and WMV video, and MP3 and WMA audio within a browser.
- Silverlight supports the ability to progressively download and play media content from any web-server.
- Silverlight also optionally supports built-in media streaming.
- Silverlight enables you to create rich UI and animations, and blend vector graphics with HTML.

- Silverlight makes it easy to build rich video player.
- No support for the .Net framework, C# or Visual Basic.

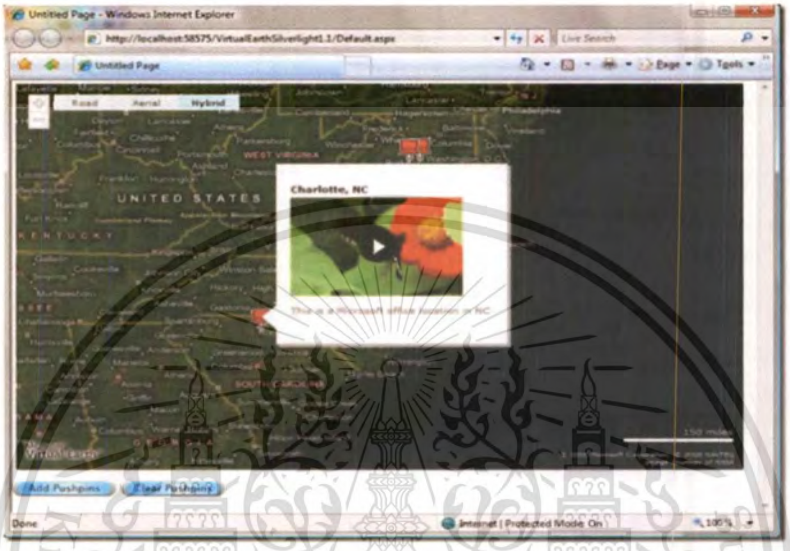


Figure 2.4 : A Silverlight 1.0. HTML controls overlaid on top of Silverlight content

2.2.3 Silverlight 2

The built-in cross platform subset of the .NET Framework. While you can still mix in as much (or as little) JavaScript as you'd like, you'll now have the option of running your own fully compiled .NET code within IE, Firefox, or Safari. Those developers who hate JavaScript should be thrilled to know they can now write their client side code in C#, VB.NET, or any other .NET language. (see example of Silverlight 2.0 on figure 2.5 and Silverlight 2 Architecture on figure 2.6)

2.2.4 Main features of Silverlight 2.0 :

- built-in CLR engine that delivers a super high performance execution environment for the browser. Silverlight uses the same core CLR engine that we ship with the full .NET Framework.
- Silverlight includes a rich framework library of built-in classes that you can use to develop browser-based applications.
- Silverlight 2 includes Deep Zoom, with smooth transitions, using the mouse wheel. The images can scale from 2 or 3 megapixels in resolution into the gigapixel range. (see example on figure 2.7)
- Silverlight 2 also allows limited file system access to Silverlight applications. It can use the operating system's native file dialog box to browse to any file (to which the user has access).
- Provides many new controls that can be used to capture user input and display collections of items
- Ability to write C# or VB.NET code that runs in the Internet Explorer, Firefox, and Safari browsers

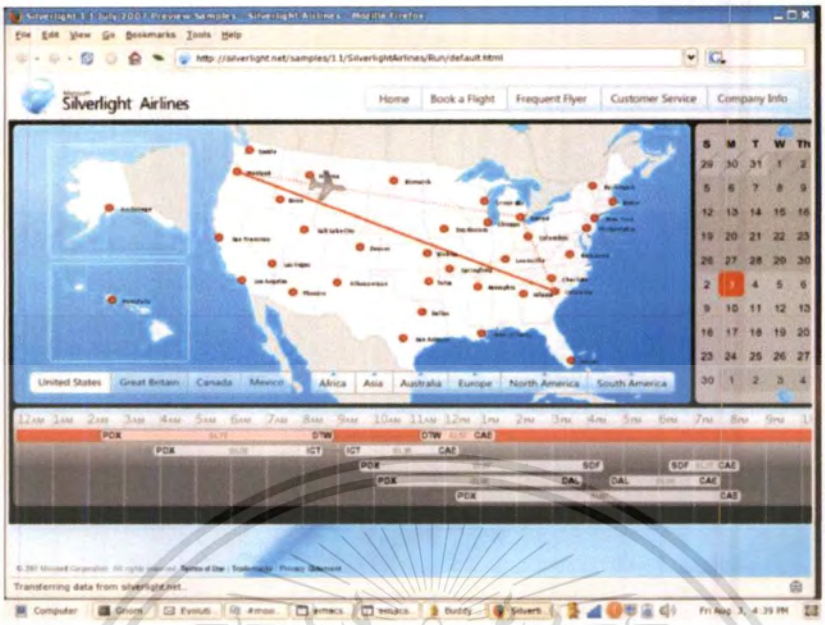


Figure 2.5 : A Silverlight 2 support for .NET languages and development tools.

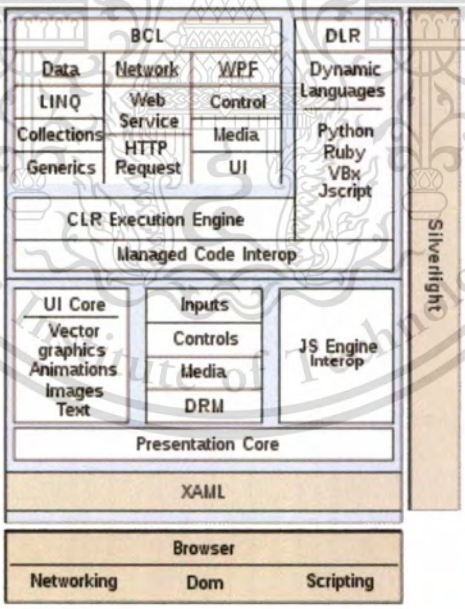


Figure 2.6 : Silverlight 2 Architecture.

Figure 2.7 : A Silverlight 2 application using Deep Zoom

2.2.5 Silverlight 3

Fully supported by Visual Studio and Expression Blend, highlights of new features and functionality of Silverlight 3 include: major media enhancements, out of browser support allowing Web applications to work on the desktop (see example on figure 2.8, figure 2.9); significant graphics improvements including 3D graphics support, GPU acceleration and H.264 video support; and many features to improve RIA development productivity. Also, in order to fully integrate all the .NET developer tools, Visual Studio 2010 will support a fully editable and interactive designer for Silverlight.

2.2.6 Main features of Silverlight 3.0 :

- Highest Quality Video Experience : all embedded in highly graphical websites. The same research and technology that was used for VC-1, the codec that powers BluRay and HD DVD
- Cross-Platform, Cross-Browser : Finally build web applications that work on any browser, and on any operating system.
- Developers familiar with Visual Studio, Microsoft.net will be able to develop amazing Silverlight applications very quickly, and they will work on Windows.
- Support for Ruby, Python, and EcmaScript.
- Cross-Platform, Cross-Browser Remote Debugging : You can set breakpoints, step into/over code, have immediate windows.
- Offers copy protection.

Figure 2.8 : Silverlight3 running in a web browser as well as installed locally

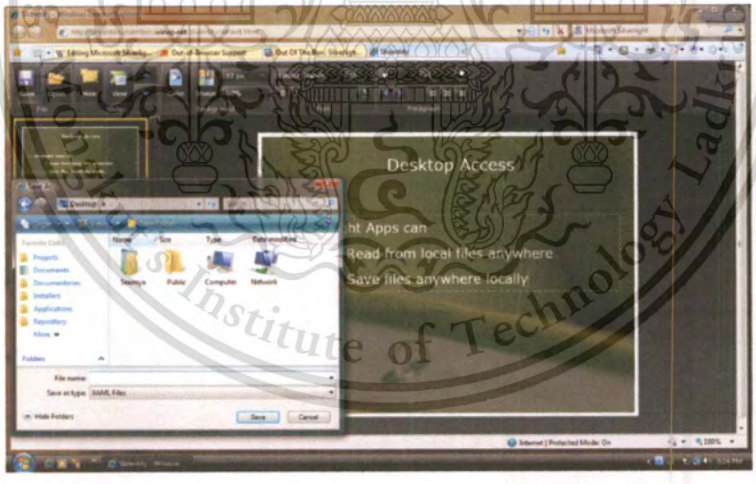


Figure 2.9 : A Silverlight 3 application demonstrating local file access.

2.2.7 Operating systems and web browsers (Silverlight)

The Following table (figure 2.14) present an availability and compatibility matrix of **Silverlight versions** for various operating systems and Web browsers.

OS/Browser	IE 6	IE7/IE8	Firefox	Safari	Opera	Chrome
Window 7	N/A	1.0,2.0,3.0	1.0,2.0,3.0	1.0,2.0,via NPAPI	Unofficially	2.0,3.0
Window Vista	N/A	1.0,2.0,3.0	1.0,2.0,3.0	1.0,2.0,via NPAPI	Unofficially	2.0,3.0
Window XP	1.0,2.0,3.0	1.0,2.0,3.0	1.0,2.0,3.0	1.0,2.0,via NPAPI	Unofficially	2.0,3.0

Figure 2.10 : Table of availability and compatibility of Silverlight

*Netscape Plug-in Application Programming Interface (NPAPI) is a cross-platform plug-in architecture used by many web browsers.

2.3 Adobe Flex

Adobe Flex is a software development kit released by Adobe Systems for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform.(see figure 2.11) Flex applications can be written using Adobe Flex Builder or by using the freely available Flex compiler from Adobe. (see example of Adobe Flex on figure 2.12)

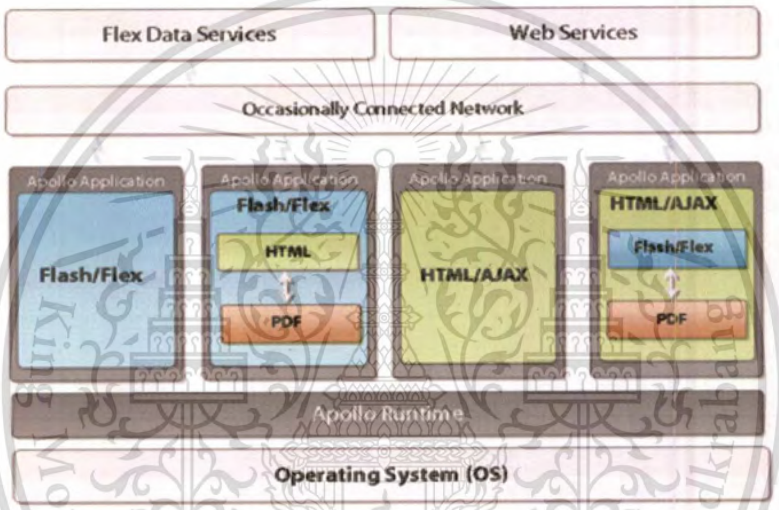


Figure 2.11 : Basic idea of Adobe Flex

2.3.1 Main features of the Adobe Flex:

- client runtime with advanced vector graphics with applications while performing at desktop application speeds.
- client runtime enables rich Internet applications (RIAs) to run on the desktop.
- Flex Builder development environment to debugging, and testing of web and desktop RIAs.

Figure 2.12 : Example of Flex application

2.3.2 Operating systems and web browsers (Adobe Flex)

The Following table(figure 2.15) present an availability and compatibility matrix of **Adobe Flex** for various operating systems and Web browsers.

OS/Browser	IE 6	IE7/IE8	Firefox	Safari	Opera	Chrome
Window 7	N/A	supported	supported	Supported	Unofficially	supported
Window Vista	N/A	supported	supported	Supported	Unofficially	supported
Window XP	supported	supported	supported	Supported	Unofficially	supported

Figure 2.13 : Table of availability and compatibility of Adobe Flex

2.4 JavaFX

JavaFX is a software platform for creating and delivering rich Internet applications that can run across a wide variety of connected devices and rich client platform for building cross-device applications and content. Designed to enable easy creation and deployment of rich internet applications (RIAs) with immersive media and content, the JavaFX platform ensures that RIAs look and behave consistently across diverse form factors and devices. Now they have 2 versions is JavaFX 1.1 and JavaFX 1.2(see example of JavaFX on figure 2.14)

2.4.1 Main features of the JavaFX platform include:

- Build engaging visual experiences across desktop, browser and mobile with a unified development and deployment model.
- you can drag-to-Install applications from your browser directly onto your desktop.
- created to have the feature of integrated development with other Java application components which run on both client and server platforms.
- designer can create a common look across all of your applications and designed to look at home across OS and on mobile devices.

Figure 2.14 : Example of JavaFX application

2.4.2 Operating systems and web browsers (JavaFx)

The Following table(figure 2.16) present an availability and compatibility matrix of **JavaFX** for various operating systems and Web browsers.

OS/Browser	IE 6	IE7/IE8	Firefox	Safari	Opera	Chrome
Window 7	IE6 minimum	supported	FireFox 3.0 minimum	supported	supported	supported
Window Vista	IE6 minimum	supported	FireFox 3.0 minimum	supported	supported	supported
Window XP	IE6 minimum	supported	FireFox 3.0 minimum	supported	supported	supported

Figure 2.15 : Table of availability and compatibility of JavaFX

JavaFX 1.2 SDK Requirements The JavaFX 1.2 SDK Includes the JavaFX compiler and runtime tools, graphics, media, web services, UI controls and rich text libraries to create RIAs for the desktop, browser and mobile platforms. Ensure that you meet the following requirements prior to installing the JavaFX 1.2 SDK on your system.

2.5 Benchmark

explanatory context

In the higher education context a benchmark is usually either (1) a level of performance, resources, or outcome against which an institution or group might be compared, or (2) the specification or codification of comparable processes.

analytical review

benchmark is: A standardized method for collecting and reporting critical operational data in a way that enables relevant comparisons among the performances of different organizations or programmes, usually with a view to establishing good practice, diagnosing problems in performance, and identifying areas of strength. Benchmarking gives the organization (or the programme) the external references and the best practices on which to base its evaluation and to design its working processes.

2.5.1 The 10-step benchmarking process

The 10-step process for conducting a benchmarking investigation consists of the following five essential phases (see Figure 2.5.1).

Phase 1: Planning

- Decide what to benchmark. All functions have a product or output. These are priority candidates to benchmark for opportunities to improve performance.
- Plan the investigation, and conduct it. Collect data sources. A wide array of sources exists, and a good starting point is a business library. An electronic search of recently published information on an area of interest can be requested. Begin collecting. Observe best practices.

Phase 2: Analysis

- Project the future performance levels. Comparing the performance levels provides an objective basis on which to act and helps to determine how to achieve a performance edge.

Phase 3: Integration

- Redefine goals and incorporate them into the planning process.
- Communicate benchmarking findings and gain acceptance from upper management.
- Revise performance goals.

Phase 4: Action

- Best practices are implemented and periodically recalibrated as needed.
- Develop and implement action plans.
- Monitor progress.
- Recalibrate the benchmarks.

Phase 5: Maturity

- Determine when a leadership position is attained. Maturity is achieved when best practices are incorporated in all business processes; when benchmarking becomes a standard part of guiding work; and when performance levels are continually improving toward a leadership position. Assess benchmarking as an ongoing process.

formal 10-step benchmarking process

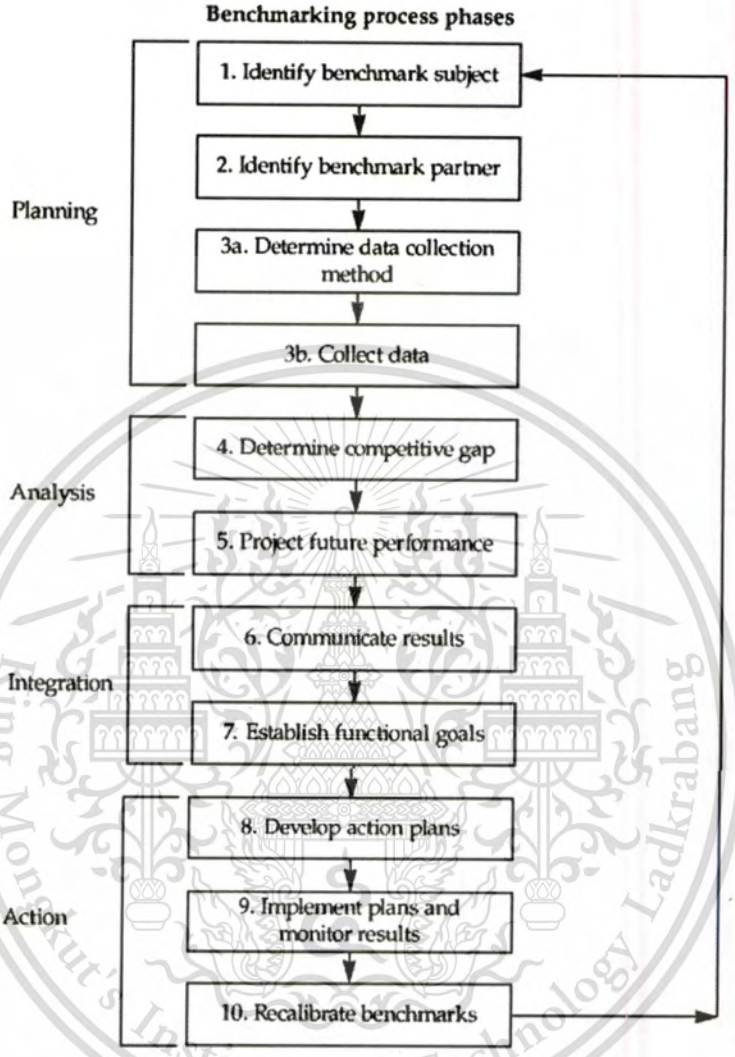


Figure 2.16 : The formal 10-step benchmarking process. (Quality Resources, a division of The Kraus Organization Limited, White Plains, NY, through ASQC Quarterly Press.)

2.5.2 Bubblemark test

Bubblemark is one of benchmark tools, mainly use for measure the performance of each RIAs tools by calculate the direction of the bubbles when it collide, the more bubble in the application, the more calculation process for application to do. The goal of this test is to directly compare performance of different RIA frameworks approaches in different browsers. The test result will give out as frame rates (FPS) which show how much frame it gets from the bubble collision calculation. The high frame rates in this test is good, its means that this RIAs tool run very fast and give a great quality in running application which mainly focus at calculation process. This survey also includes the CPU usage data of each RIAs tool when run the Bubblemark test. The more CPU process given from test the more work for computer system to do which makes the computer system run slowly and give the poor quality in application presentation.

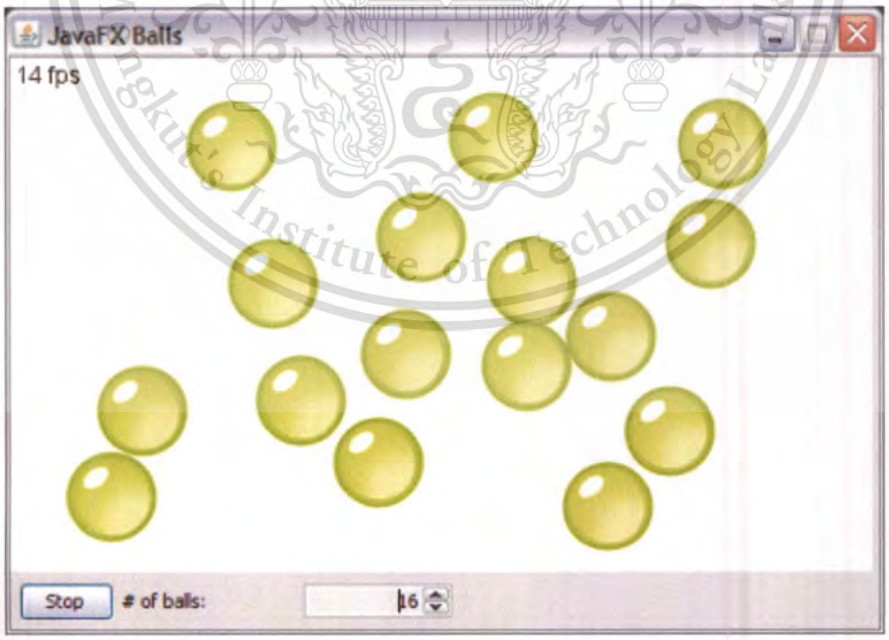


Figure 2.17 : Bubblemark test in JavaFX platform

2.5.3 Task Manager

"Task manager", a default program included in all Microsoft Windows operating system which allow user to view the data such as currently running processes, network status, etc.

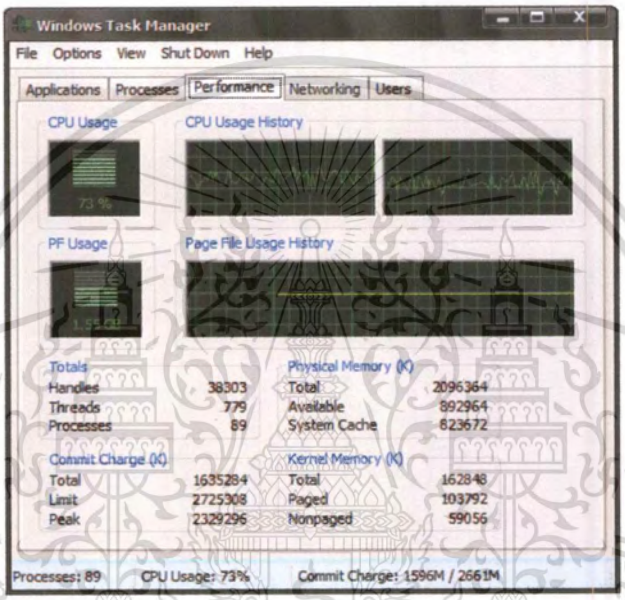


Figure 2.18 : Task Manager Interface

Figure 2.19 : CPU Process viewer

Chapter 3

Experiment Analysis and Design

This chapter will cover designs we have for this project. These designs are system analysis and experiment design. System analysis part will explain the idea behind this project, problem analysis and what result will be gave when this experiment finish. While experiment design explains how we put the tools to test and how we collect results from it. The testing result and conclusion will be shown in Chapter 4 and 5 respectively.

3.1 Experiment Analysis

3.1.1 Problem

In RIA development process, RIAs application developer would prefer to see what is the best tools that fulfill customer requirement and uses resource as low as possible to save time and raise the profit of the organization.

3.1.2 Result Giving

The result will give out in CPU usage (as percent) and Framerates (as frame per second). Framerates result will show the calculation performance and CPU usage result will show the performance of CPU resource management in each RIAs tool.

3.2 Experiment Design

We will compare the calculation performance and CPU resource management of each RIA tools by running the application called Bubblemark in different computer set and different internet browsers to find out which RIA tools gives us the best performance in both of calculation power and CPU resource management. Also find out what internet browser gives the best performance in running RIAs application.

The computer set we use are,

- Intel Core T2500@ 2.00GHz, 1.49 GB of Ram, Run on Windows XP
- Intel Core Duo T2250@ 1.8GHz, 2GB of Ram, Run on Windows XP
- Intel Core 2 Duo P8600 @ 2.40GHz, 4 GB of Ram, Run on Window Vista

And these are Internet browser which we use to test in this survey,

- Internet Explorer 8
- Mozilla Firefox 3.6
- Google Chrome 4.1
- Opera 10.10
- Apple Safari 4.04

We separate the test of each RIAs tools into 5 parts, separate by each internet browsers. And also separate the each RIAs tool test again into 6 parts, separate by the amount of balls in the test which is 1 ball, 8 balls, 16 balls, 32 balls, 64 balls, and 128 balls. Each of the tests will tell the performance in RIAs in both of calculation performance and CPU resource management in different situations (Ex. 1 ball test will represent the lowest computer resource consumption test while 128 balls will represent the maximum computer resource consumption test)

3.3 Data Collection and Visualization

We collected the Frame rates and CPU process in every 5 second until 1 minute every single test. Frame rates can be viewed from the top left of Bubblemark application while CPU usage can be viewed from Task manager. We average the result in each test then put all result onto a table (Figure 3.1), and after all we'll summarize it in a graph form (Figure 3.2). All the result will show at Chapter 4.

	Chrome											
	[illegible]		[illegible]		[illegible]		[illegible]		[illegible]		[illegible]	
	CPU	FPS	CPU	FPS	CPU	FPS	CPU	FPS	CPU	FPS	CPU	FPS
Silverlight(Java Script)	[illegible]											
Silverlight 3.0	12.83	203.08	11.5	197	13.08	193	15.66	186	22.08	175	33.66	157.66
Silverlight 2.0	[illegible]											
Flex	15.33	164.25	28.25	110	37.58	81.08	44.41	51	48.08	29	51.58	14.16
Flex with cacheAsBitmap	[illegible]											
Java FX 2.2	5.08	958.66	24.25	693.41	24.58	660.83	32.83	526.5			45.75	245

Figure 3.1 : Example of Bubblemark test result table from Google Chrome

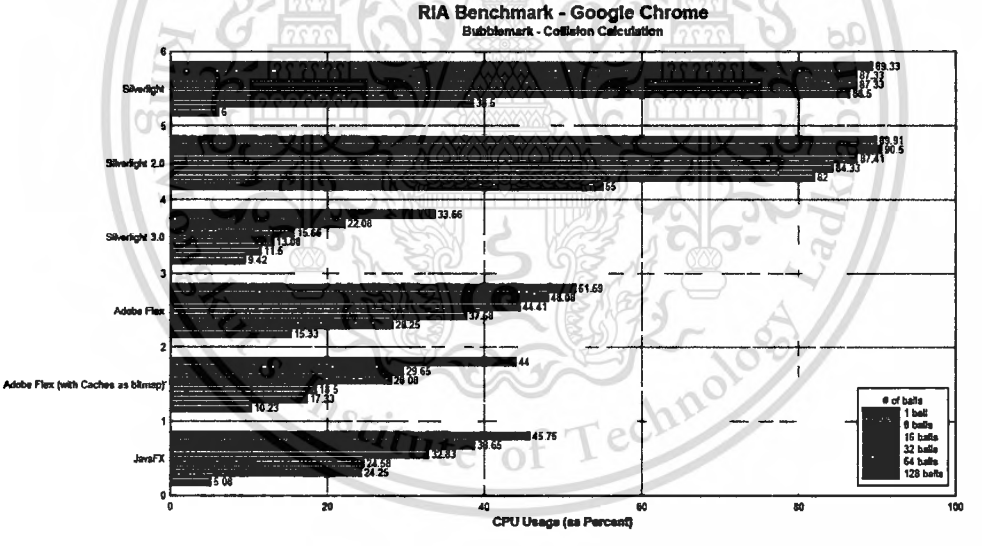


Figure 3.2 : Example of graph result from Google Chrome (CPU Usage)

Chapter 4

Result and Discussion

4.1 Experiment purpose

Experiment result will show the difference of RIAs performance between each RIAs tool when runs the Bubblemark test in different Internet browser. The connection between the performances is a factor that judges the performance of RIAs tool in calculation performance and CPU resource management. Good RIAs tool will provide a good frame rates when running benchmark program and using CPU resource as low as possible but give a good result in frame rates. Also, get the highest score while running in the most of Internet browsers which available in the market today

4.1.1 Introduction to 1st experiment

This is the first experiment of this survey, and will set to be a main reference to compare with later experiments. This experiment will use this computer set to test,

Intel Core T2500@ 1.80GHz, 2 GB of Ram, Run on Windows XP

Experiment result will show the difference of RIAs performance between each RIAs tool when runs the Bubblemark test in different Internet browser. The connection between the performances is a factor that judges the performance of RIAs tool in calculation performance and CPU resource management. Good RIAs tool will provide a good frame rates when running while use CPU resource as low as possible and give a good result or support the most Internet browsers in the market.

4.1.2 Experiment 1 graph result: Intel Core T2500@ 1.80GHz, 2 GB of Ram, Run on Windows XP

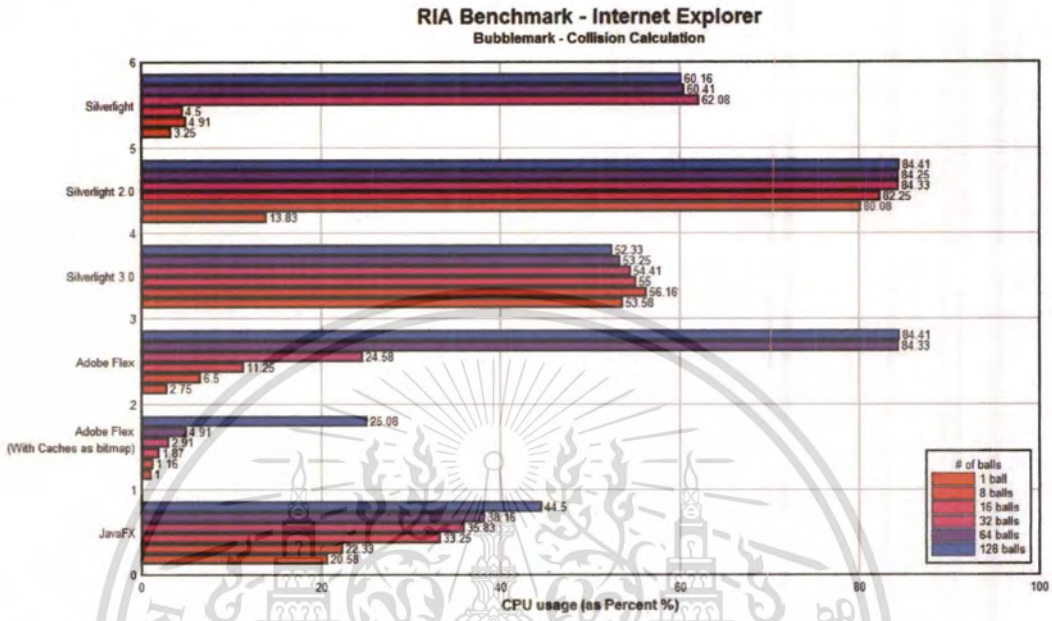


Figure 4.1 : is a CPU usage graph of running the benchmark application on Internet Explorer 8. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Adobe Flex (with caches as bitmap) gets the lowest value. It's mean that Adobe Flex (with caches as bitmap) use CPU resource less than other tools in this experiment.

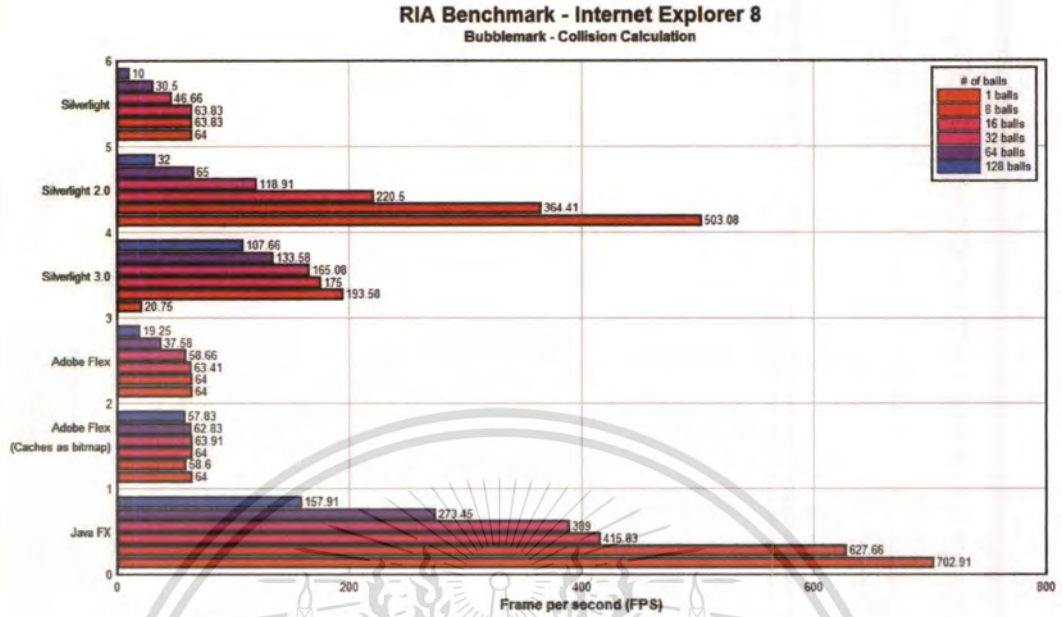


Figure 4.2 : is a graph result of frame rate while running a bubblemark test on Internet Explorer 8. Which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

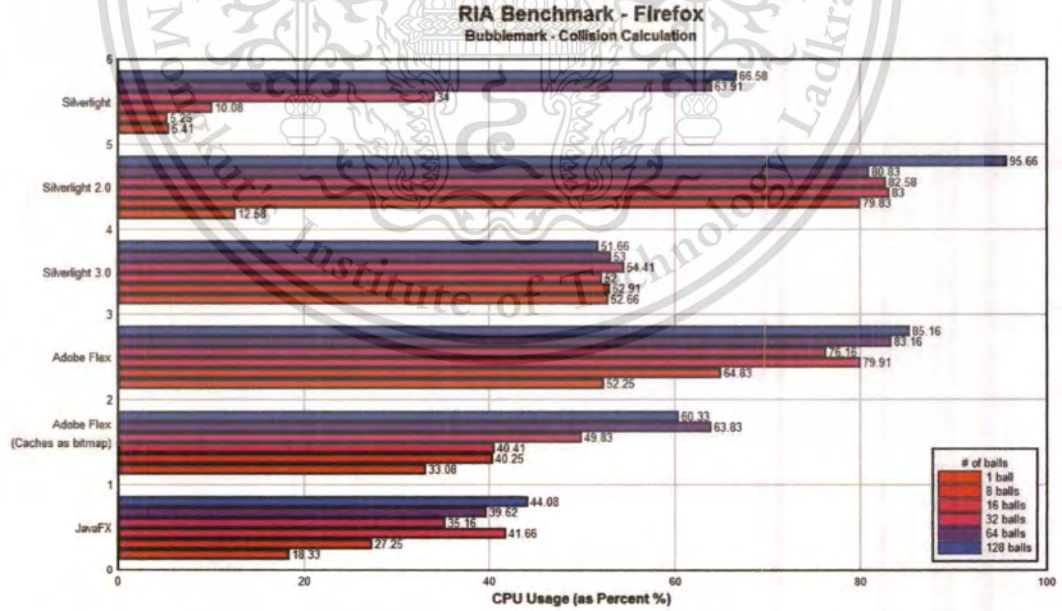


Figure 4.3 : is a CPU usage graph of running the benchmark application on Mozilla Firefox 3.6. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight gets the lowest value. It's mean that Silverlight use CPU resource less than other tools in this experiment.

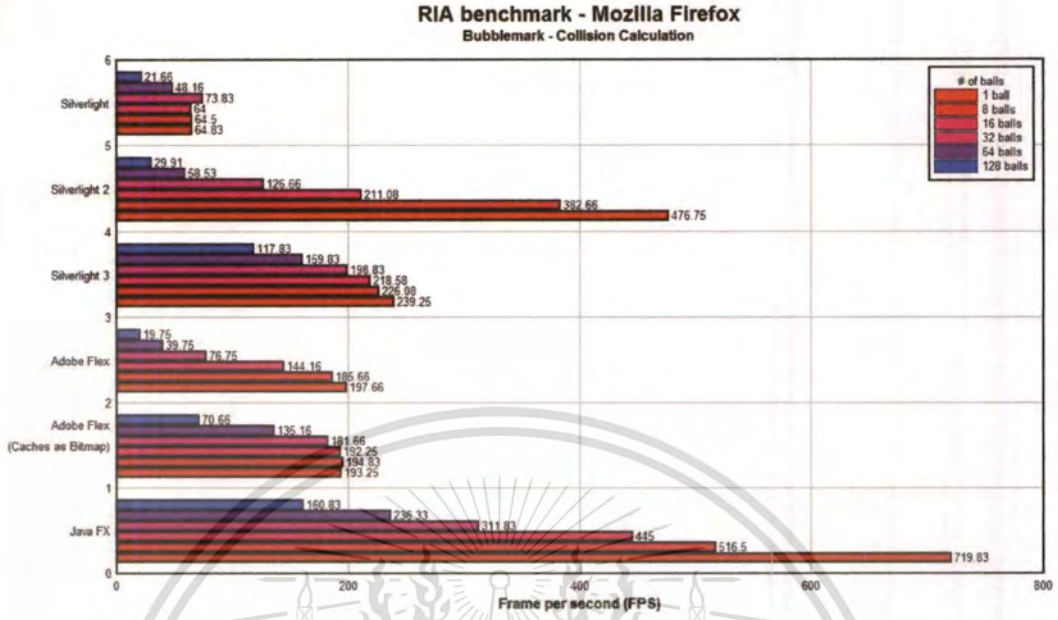


Figure 4.4 : is a graph result of frame rate while running a bubblemark test on Mozilla Firefox 3.6. Which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

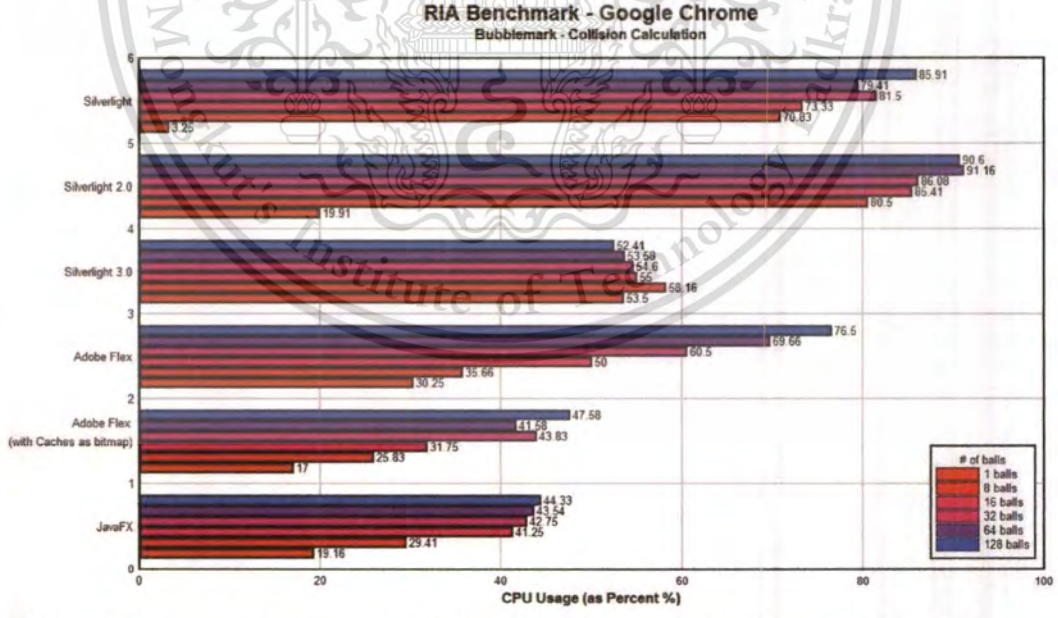


Figure 4.5 : is a CPU usage graph of running the benchmark application on Google Chrome 4.1. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Adobe Flex (with caches as bitmap) gets the lowest value. It's mean that Adobe Flex (with caches as bitmap) use CPU resource less than other tools in this experiment.

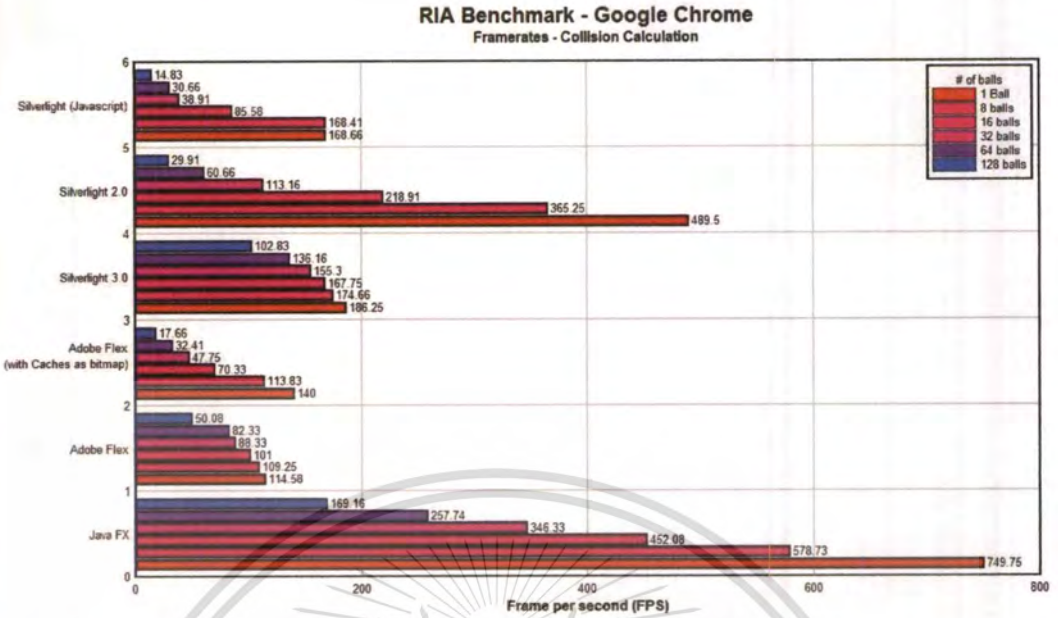


Figure 4.6 : is a graph result of frame rate while running a bubblemark test on Google Chrome 4.1. Which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

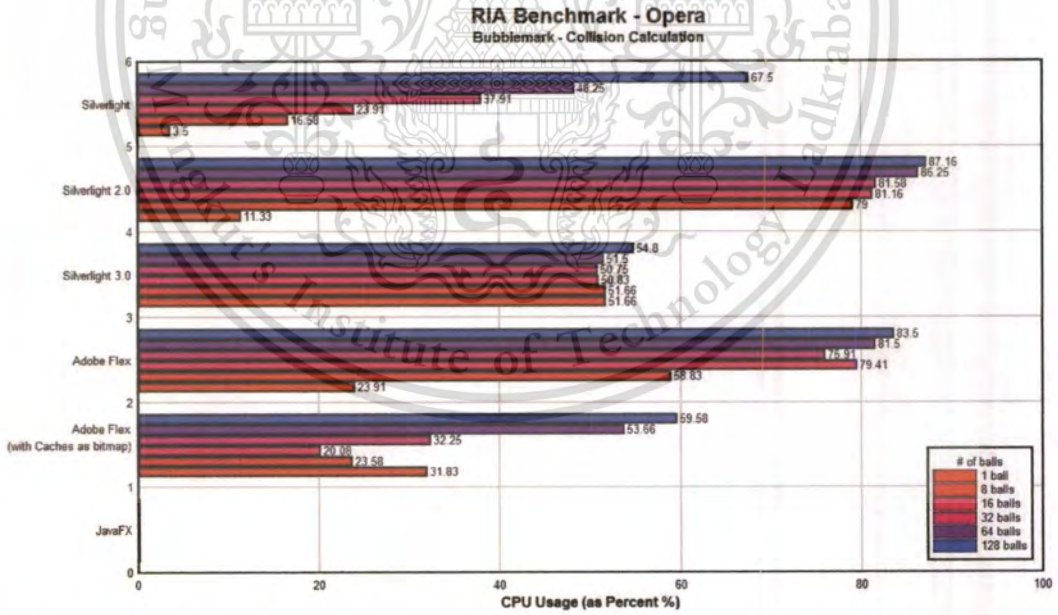


Figure 4.7 : is a CPU usage graph of running the benchmark application on Opera 10.10. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight gets the lowest value. It's mean that Silverlight use CPU resource less than other tools in this experiment.

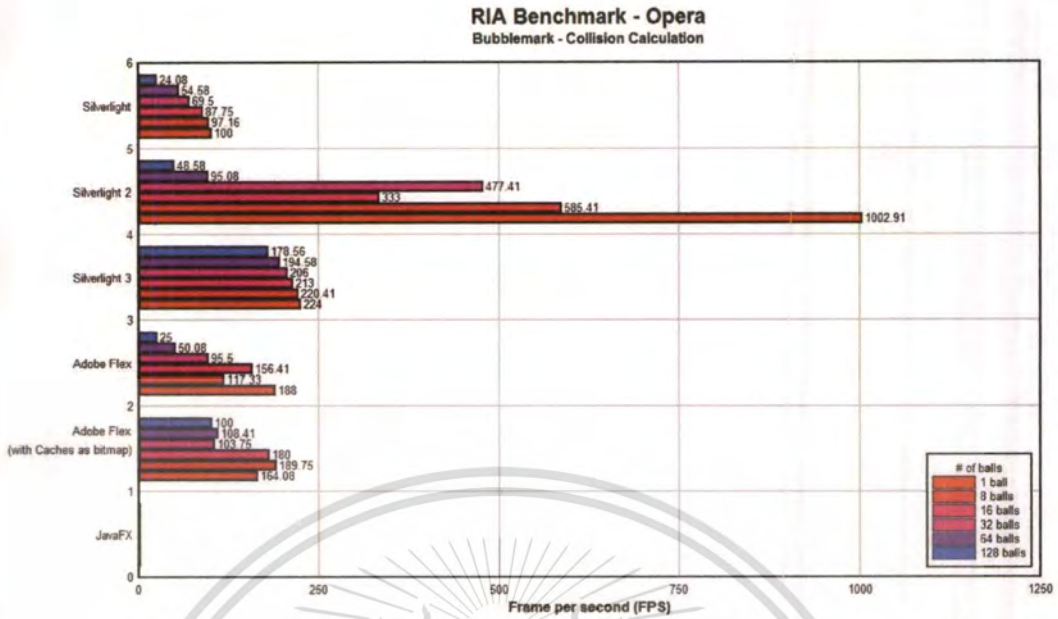


Figure 4.8 : is a graph result of frame rate while running a bubblemark test on Opera 10.10. Which show how much frame rate had been showed while doing an experiment. Graph result show that Silverlight2 got the highest scores in this test while comparing with other tools in this test .

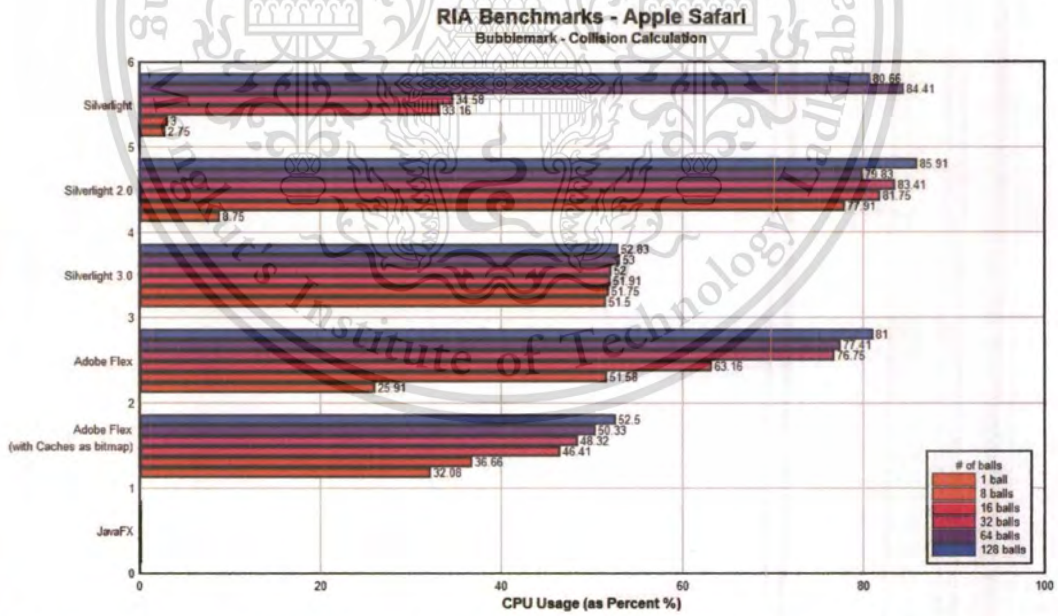


Figure 4.9 : is a CPU usage graph of running the benchmark application on Safari 4.04. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight gets the lowest value. It's mean that Silverlight use CPU resource less than other tools in this experiment.

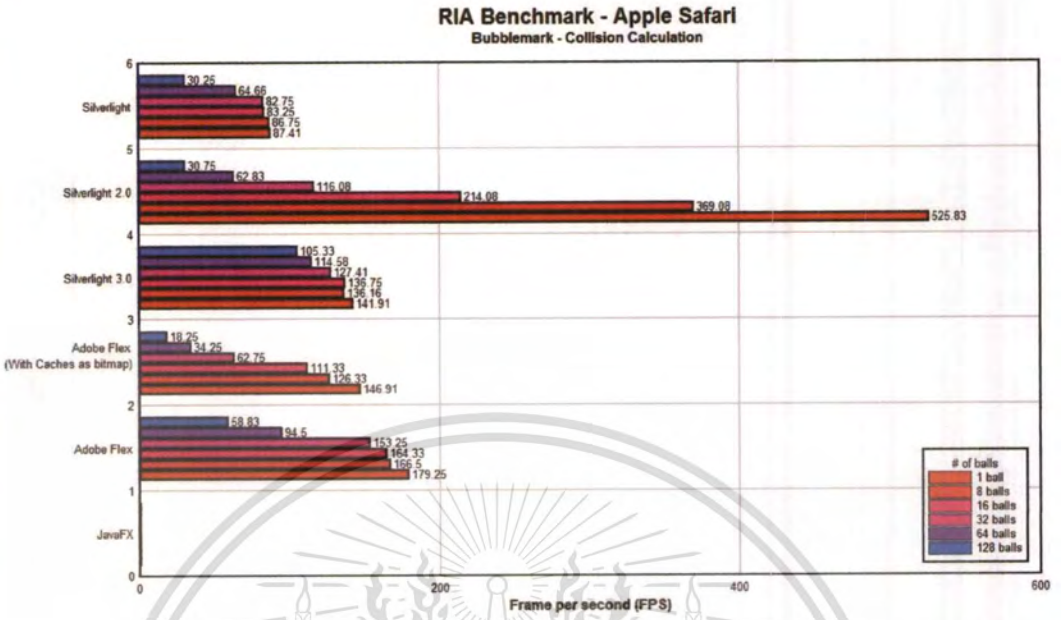


Figure 4.10 : is a graph result of frame rate while running a bubblemark test on Safari 4.04. Which show how much frame rate had been showed while doing an experiment. Graph result show that Silverlight2 got the highest scores in this test while comparing with other tools in this test .

4.1.3 Experiment 1: result conclusion

CPU USAGE (Percent)						
RIAs Tools / Browser	Silverlight	Silverlight2.0	Silcerlight3.0	Adobe Flex	Adobe Flex (Cache as Bitmap)	JavaFX
Internet Explorer 8	62.08	84.33	54.41	24.58	2.91	35.83
Mozilla Firefox 3.6	34	82.58	54.41	76.16	49.83	35.16
Google Chrome 4.1	81.5	86.08	54.6	60.5	43.83	42.75
Opera 10.10	37.91	81.58	50.75	75.91	32.25	Not Support
Apple Safari 4.04	34.58	83.41	52	76.75	48.32	Not Support

Figure 4.11 : Average result in CPU usage (Percent)

Calculation Performance (Frame Per sec)						
RIAs Tools / Browser	Silverlight	Silverlight2.0	Silcerlight3.0	Adobe Flex	Adobe Flex (Cache as Bitmap)	JavaFX
Internet Explorer 8	46.66	118.91	165.08	58.66	63.91	389
Mozilla Firefox 3.6	73.83	126.66	198.83	76.75	181.66	311.83
Google Chrome 4.1	85.58	133.16	155.3	47.75	88.33	346.33
Opera 10.10	39.83	129.16	168.66	77.91	118.5	Not Support
Apple Safari 4.04	82.75	116.08	127.41	62.75	153.25	Not Support

Figure 4.12 : Average result in Calculation performance (FPS)

From all experiment result we had done, we choose the result from 32 balls test in each of Internet browsers and all RIAs tool to compare the performance (as shows in Figure 4.11 and Figure 4.12). We can briefly conclude that JavaFX is done great in both of CPU resource management and calculation performance while Internet Explorer done best in CPU resource management and Mozilla Firefox done best in calculation performance.

4.1.4 Introduction to 2nd experiment

This experiment will proof the topic about "Each RIAs tool needs Graphic card to improve their performance in running any application" and test about the performance of computer system which has GPU and computer system which not has GPU when run RIAs.

This experiment will use this computer set to test,

Intel Core Duo T2250@ 1.8GHz, 2GB of Ram, Run on Windows XP

This experiment focuses at the different between computer that has GPU (Graphic Processing Unit) and computer that not has GPU, which directly relate to Calculation power in RIAs tool. The CPU resource management result is not important to this experiment because the computer system which we use in this experiment is almost the same spec and run in the same environment.

As we predicted before in chapter 3, the different of Calculation performance in this test should be at least about 20%. If the test result shows that the different is not reach at our target, we can claim that the topic about GPU and RIAs is not true at all also can show that normal computer system can run RIAs with no different when compare with computer which has GPU.

4.1.5 Experiment 2 graph result: Intel Core Duo T2250@ 1.8GHz, 2GB of Ram, Run on Windows XP

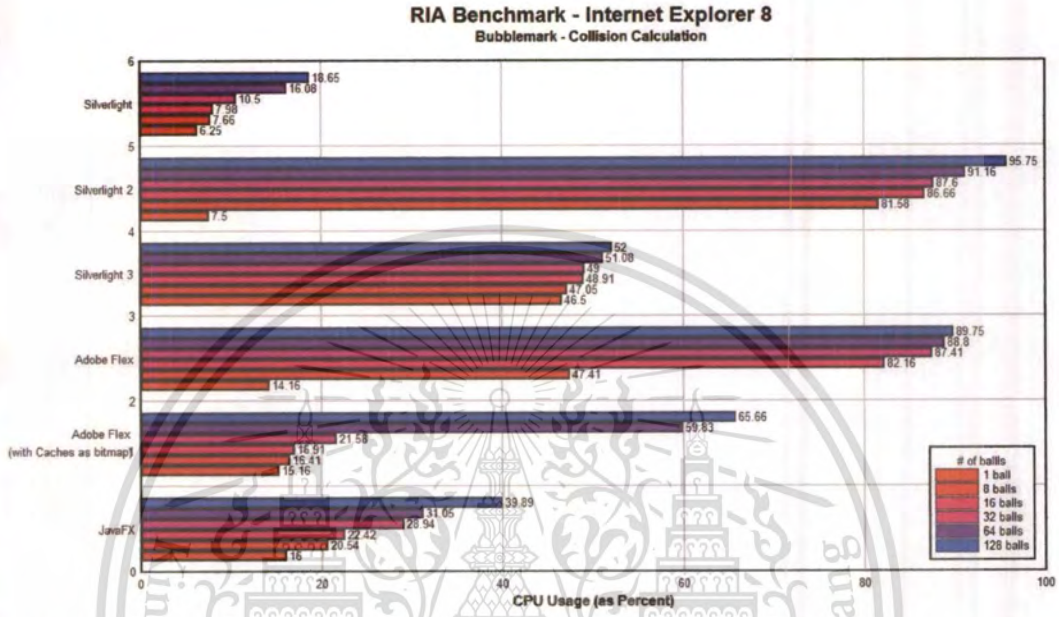


Figure 4.13 : is a CPU usage graph of running the benchmark application on Internet Explorer 8. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight gets the lowest value. It's mean that Silverlight use CPU resource less than other tools in this experiment.

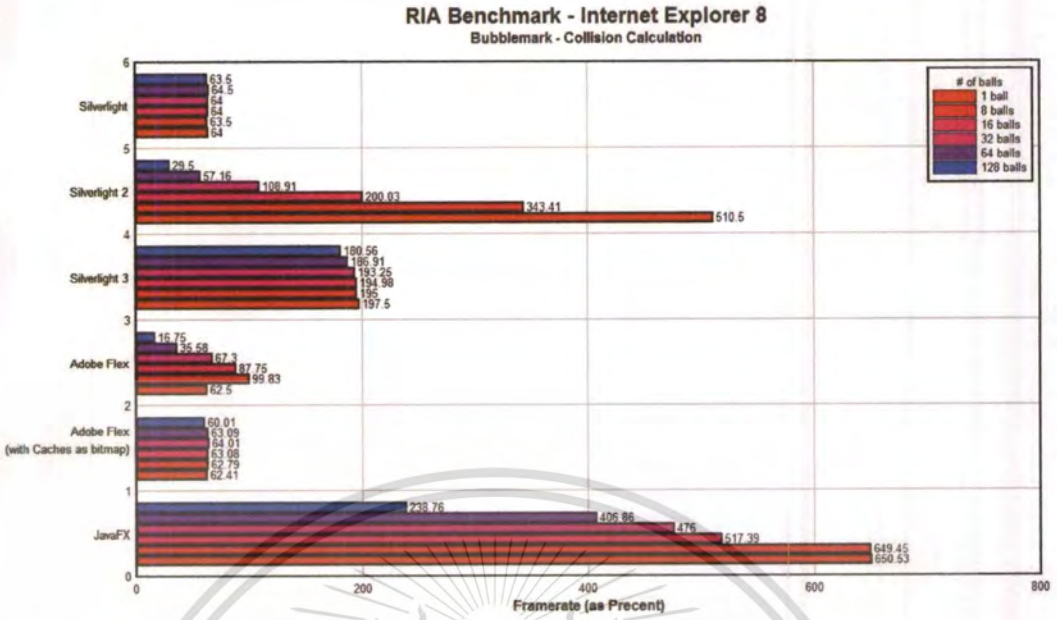


Figure 4.14 : is a graph result of frame rate while running a bubblemark test on Internet Explorer 8. Which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

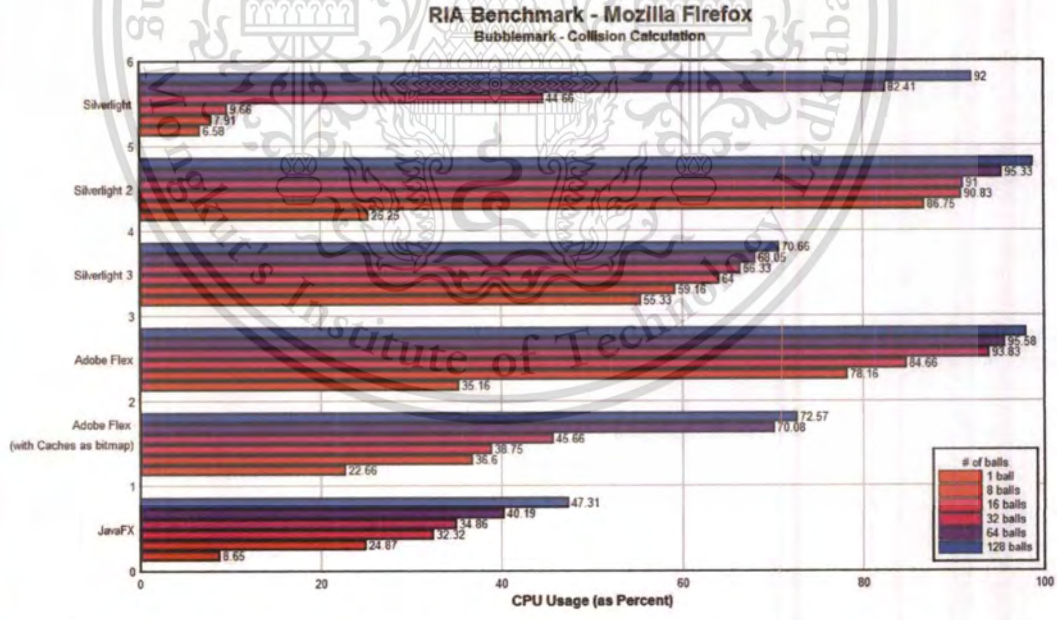


Figure 4.15 : is a CPU usage graph of running the benchmark application on Mozilla Firefox 3.6. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that JavaFX gets the lowest value. It's mean that JavaFX use CPU resource less than other tools in this experiment.

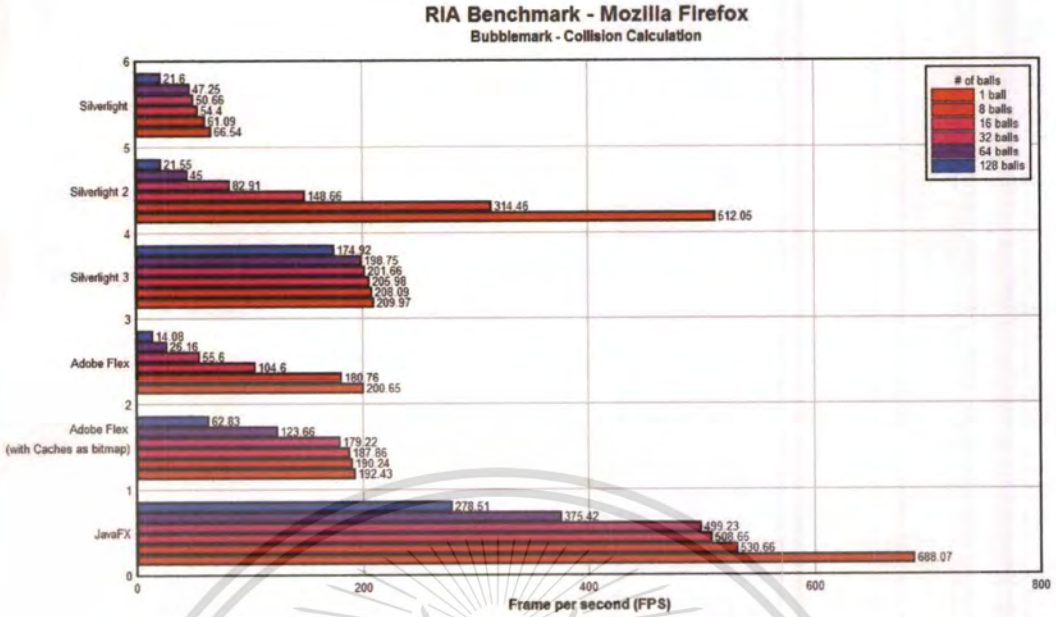


Figure 4.16 : is a graph result of frame rate while running a bubblemark test on Mozilla Firefox 3.6. Which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

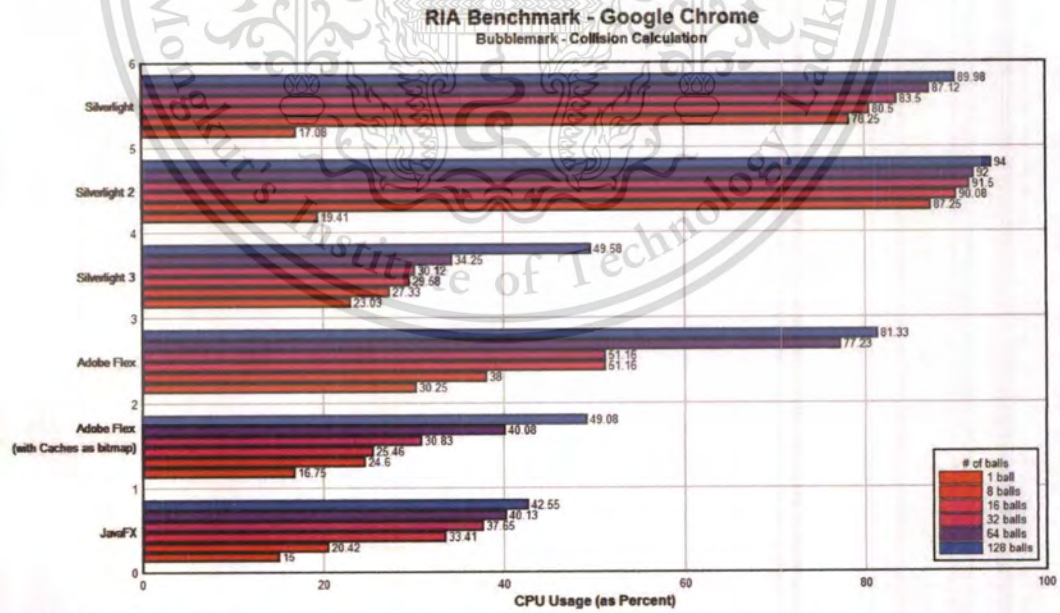


Figure 4.17 : is a CPU usage graph of running the benchmark application on Google Chrome 4.1. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight 3 gets the lowest value. It's mean that Silverlight 3 use CPU resource less than other tools in this experiment.

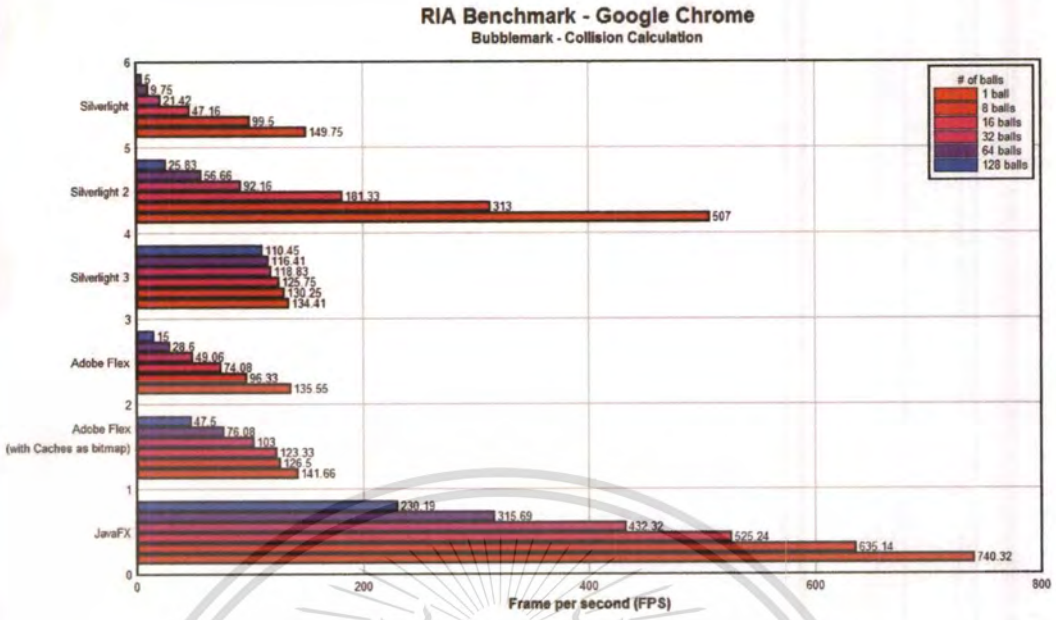


Figure 4.18 : is a graph result of frame rate while running a bubblemark test on Google Chrome 4.1. Which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

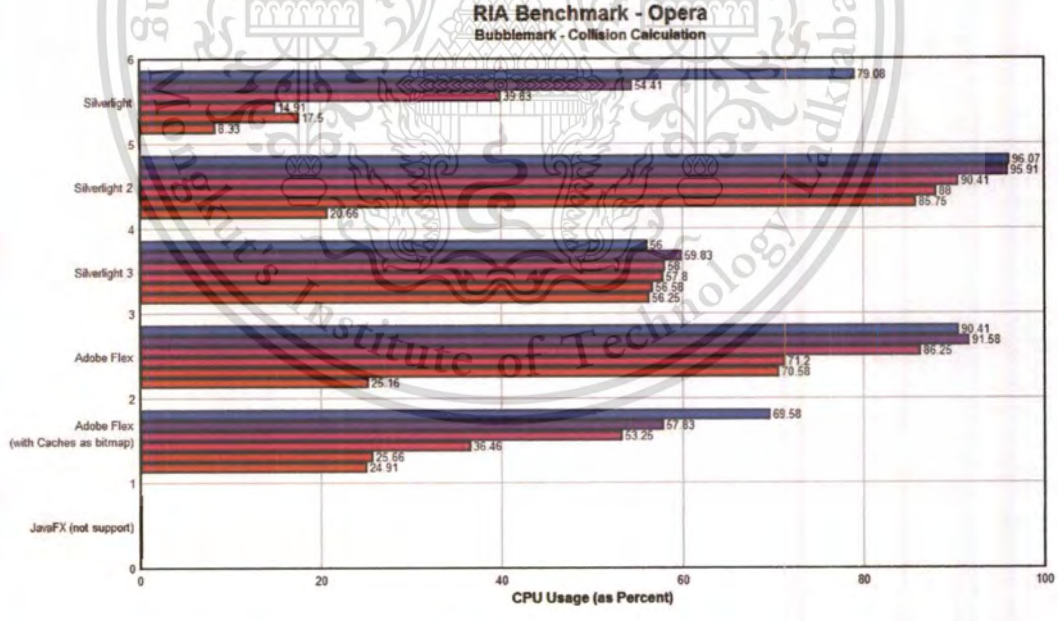


Figure 4.19 : is a CPU usage graph of running the benchmark application on Opera 10.10. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight gets the lowest value. It's mean that Silverlight use CPU resource less than other tools in this experiment.

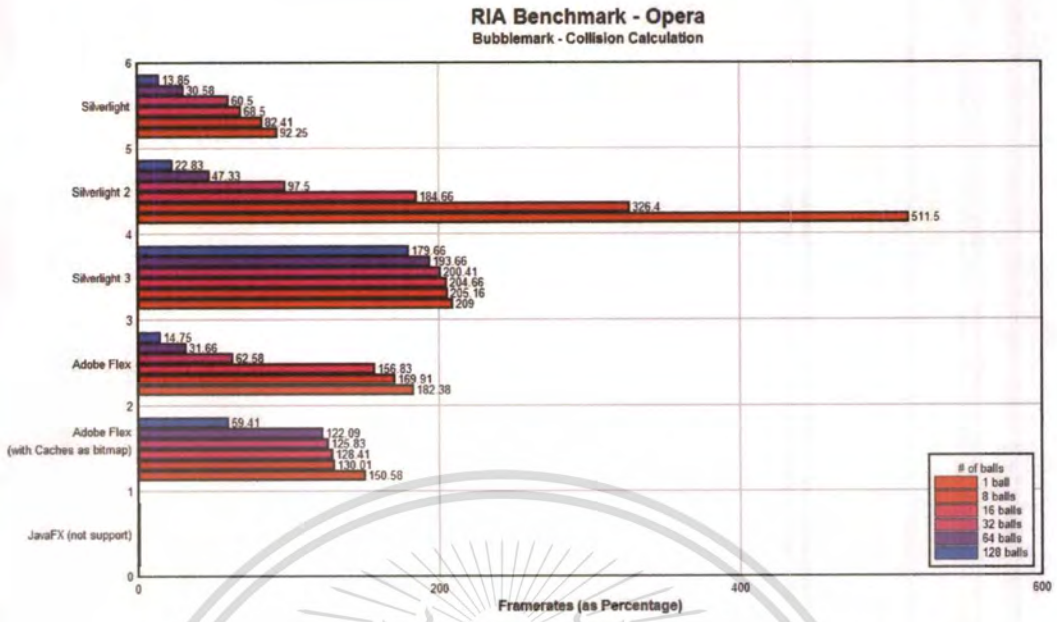


Figure 4.20 : is a graph result of frame rate while running a bubblemark test on Opera 10.10. Which show how much frame rate had been showed while doing an experiment. Graph result show that Silverlight3 got the highest scores in this test while comparing with other tools in this test .

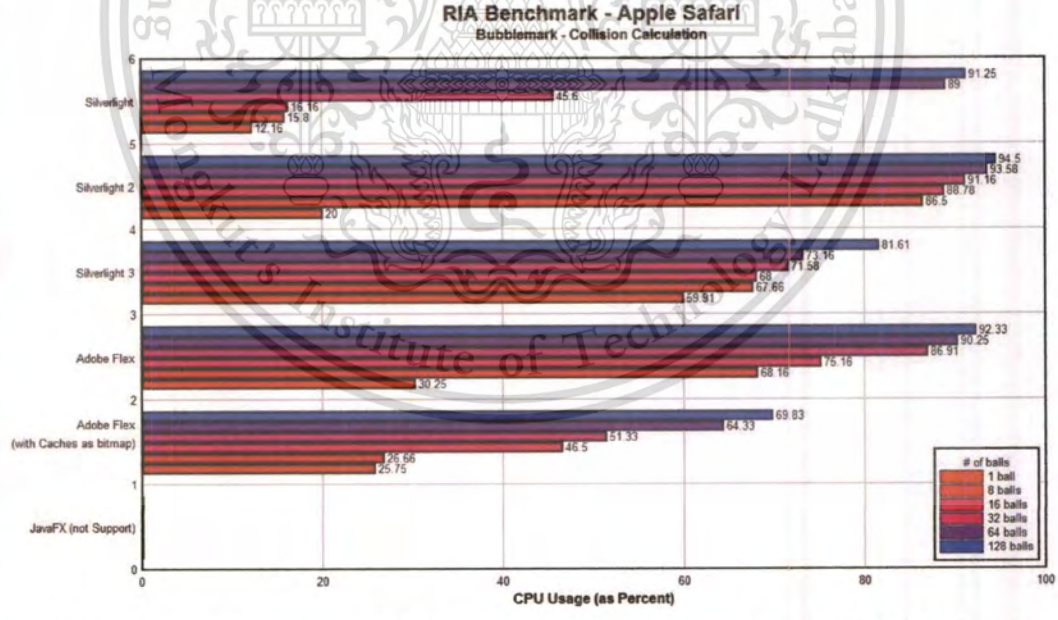


Figure 4.21 : is a CPU usage graph of running the benchmark application on Apple Safari 4.04. Which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows Silverlight gets the lowest value. It's mean that Silverlight use CPU resource less than other tools in this experiment.

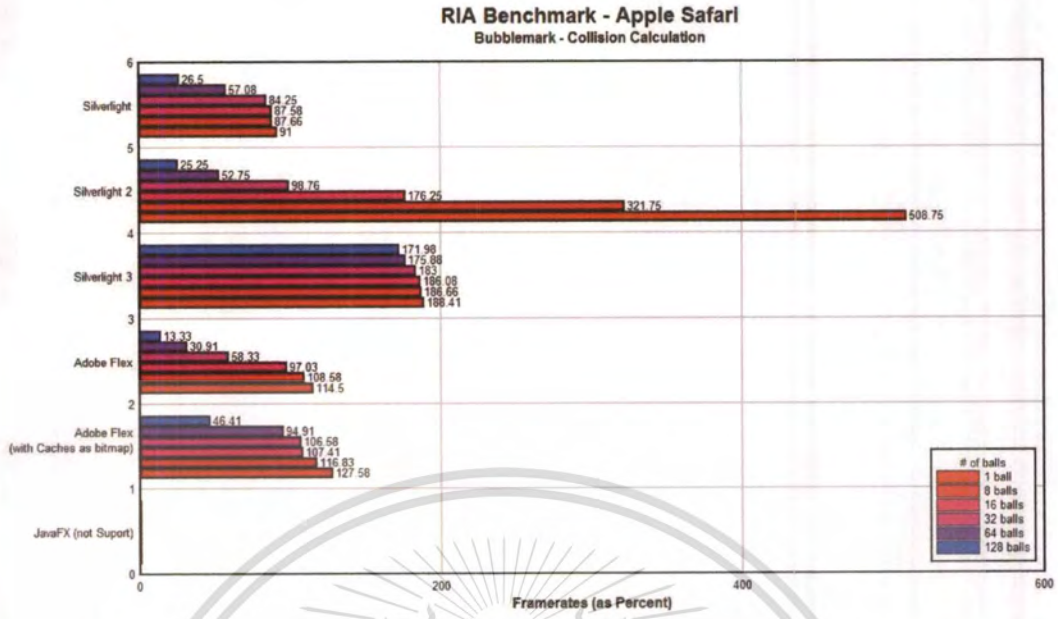


Figure 4.22 : is a graph result of frame rate while running a bubblemark test on Apple Safari 4.04. Which show how much frame rate had been showed while doing an experiment. Graph result show that Silverlight3 got the highest scores in this test while comparing with other tools in this test .

4.1.6 Experiment 2: graph conclusion

CPU USAGE (Percent)						
RIAs Tools / Browser	Silverlight	Silverlight2.0	Silcerlight3.0	Adobe Flex	Adobe Flex (Cache as Bitmap)	JavaFX
Internet Explorer 8	10.5	87.6	49	87.41	21.58	28.94
Mozilla Firefox 3.6	44.66	91	66.33	93.83	45.66	34.86
Google Chrome 4.1	83.5	91.5	30.12	51.16	30.83	37.65
Opera 10.10	39.83	90.41	68	86.25	53.25	Not Support
Apple Safari 4.04	45.6	91.16	71.58	86.91	51.33	Not Support

Figure 4.23 : Average result in CPU usage (Percent)

Calculation Performance (Frame Per sec)						
RIAs Tools / Browser	Silverlight	Silverlight2.0	Silcerlight3.0	Adobe Flex	Adobe Flex (Cache as Bitmap)	JavaFX
Internet Explorer 8	64	108.91	193.25	67.3	64.01	476
Mozilla Firefox 3.6	50.66	82.91	201.66	55.6	179.22	499.23
Google Chrome 4.1	21.42	92.16	118.83	49.06	103	432.32
Opera 10.10	60.5	97.5	200.41	62.58	125.83	Not Support
Apple Safari 4.04	84.25	98.76	183	58.33	106.58	Not Support

Figure 4.24 : Average result in Calculation Performance (FPS)

From all experiment result we had done, we choose the result from 32 balls test in each of Internet browsers and all RIAs tool to compare the performance (as shows in Figure 4.23 and Figure 4.24). We can briefly conclude that JavaFX is done great in both of CPU resource management and calculation performance while Internet Explorer done best in CPU resource management and Mozilla Firefox done best in calculation performance.

4.1.7 Introduction to 3rd experiment

This experiment purpose is to see the different in result when compare the high performance computer system and normal computer system for claim the topics such as "The better CPU performance, the better result RIAs will provide".

This experiment will use this computer set to test,

Intel Core 2 Duo P8600 @ 2.40GHz, 4 GB of Ram, Run on Window Vista

As we predicted the result of this experiment in chapter 3, the result in this experiment should be better than normal computer system (Experiment 1) about 50%. The question is why we have to test in the computer system which we know its better and know that the result will be different? The answer for this question is some organization or work office want to know that they have to upgrade their computer system or not to develop or run RIAs. If the result is not satisfactory so there is no need to update the machine.

4.1.8 Experiment 3 graph result: Intel Core 2 Duo P8600 @ 2.40GHz, 4 GB of Ram, Run on Window Vista

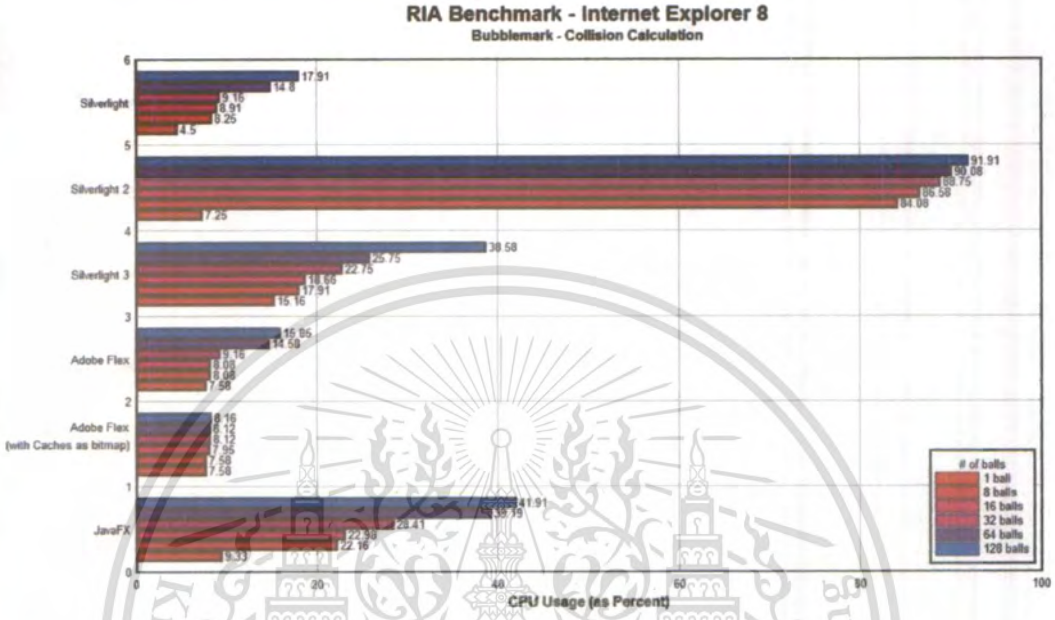


Figure 4.25 : is a CPU usage graph of running the benchmark application on Internet Explorer 8 which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Adobe Flex (with caches as bitmap) gets the lowest value. It's mean that Adobe Flex (with caches as bitmap) use CPU resource less than other tools in this experiment.

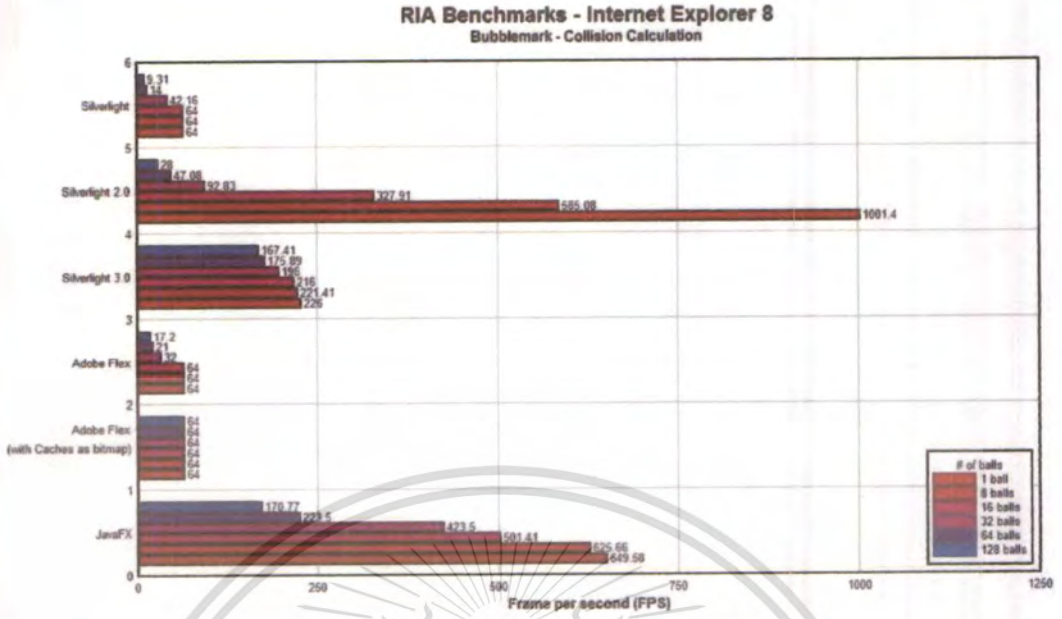


Figure 4.26 : is a graph result of frame rate while running a bubblemark test on Internet Explorer 8. which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

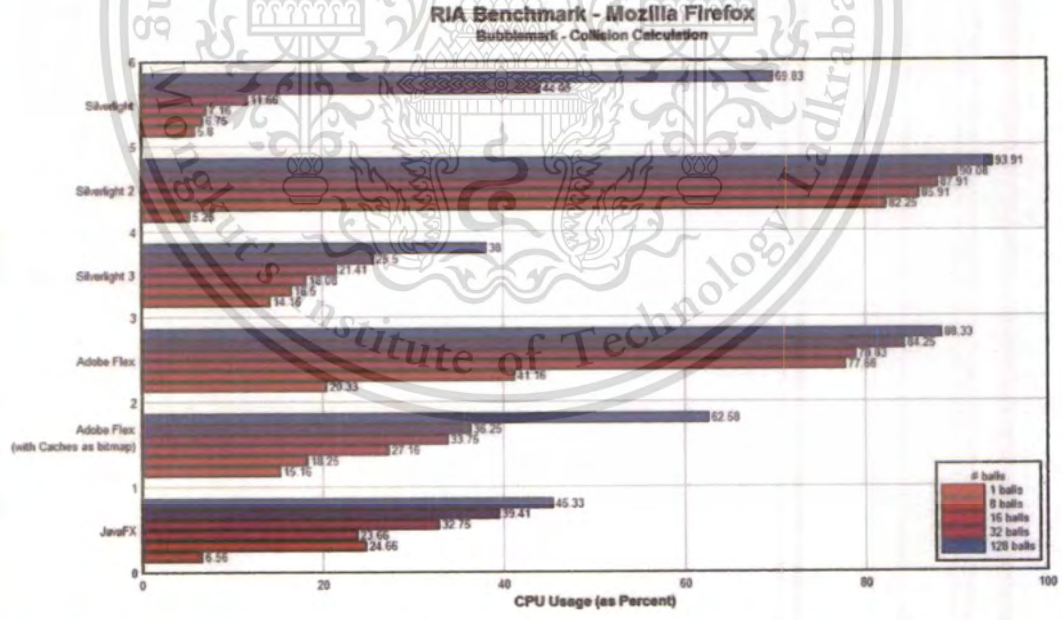


Figure 4.27 : is a CPU usage graph of running the benchmark application on Mozilla Firefox 3.6. which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight 3 gets the lowest value. It's mean that Silverlight 3 use CPU resource less than other tools in this experiment.

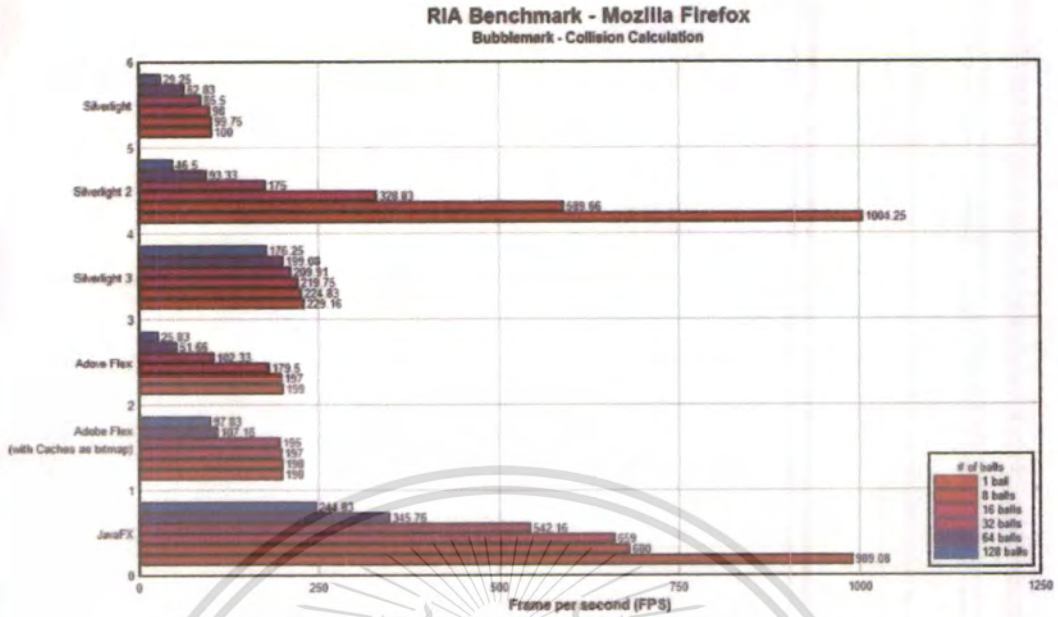


Figure 4.28 : is a graph result of frame rate while running a bubblemark test on Mozilla Firefox 3.6. which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

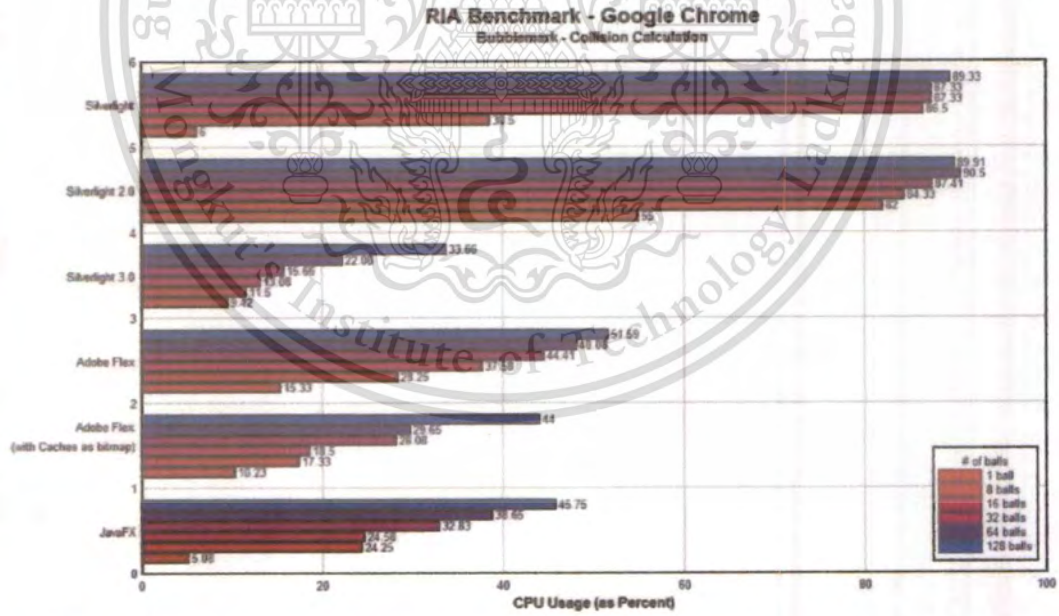


Figure 4.29 : is a CPU usage graph of running the benchmark application on Google Chrome 4.1. which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight 3 gets the lowest value. It's mean that Silverlight 3 use CPU resource less than other tools in this experiment.

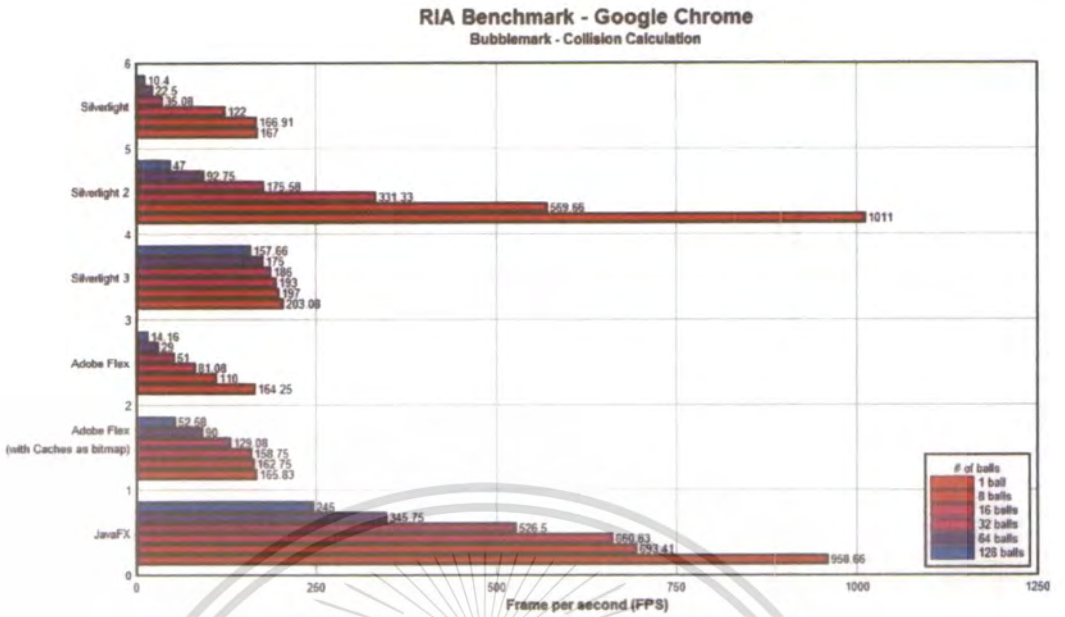


Figure 4.30 : is a graph result of frame rate while running a bubblemark test on Google Chrome 4.1. which show how much frame rate had been showed while doing an experiment. Graph result show that JavaFX got the highest scores in this test while comparing with other tools in this test .

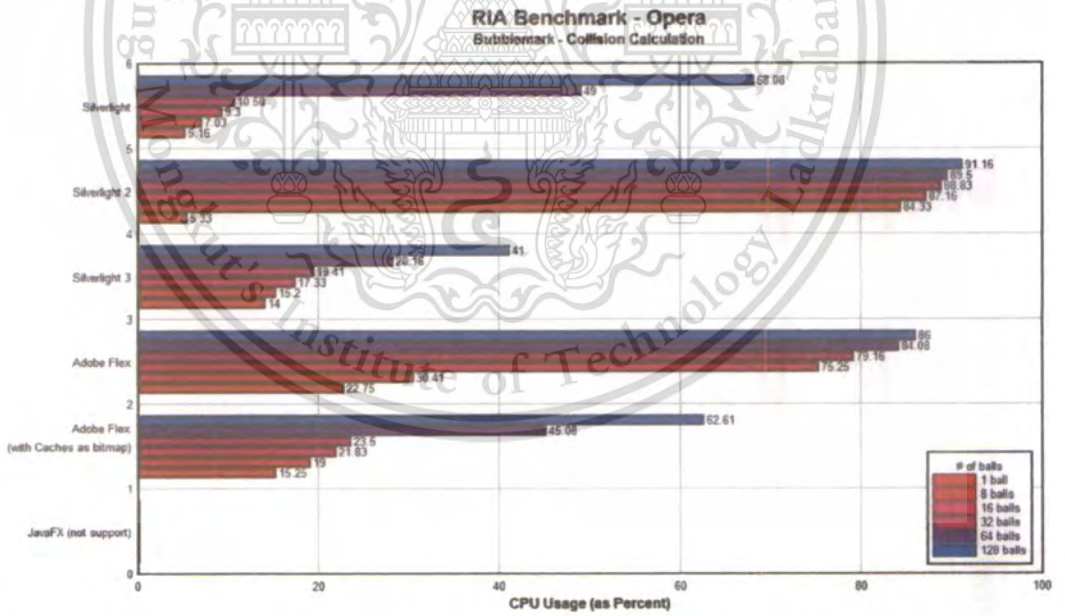


Figure 4.31 : is a CPU usage graph of running the benchmark application on Opera 10.10. which show how much CPU resource had been used while doing an experiment. Result from CPU usage graph shows that Silverlight 3 gets the lowest value. It's mean that Silverlight 3 use CPU resource less than other tools in this experiment.

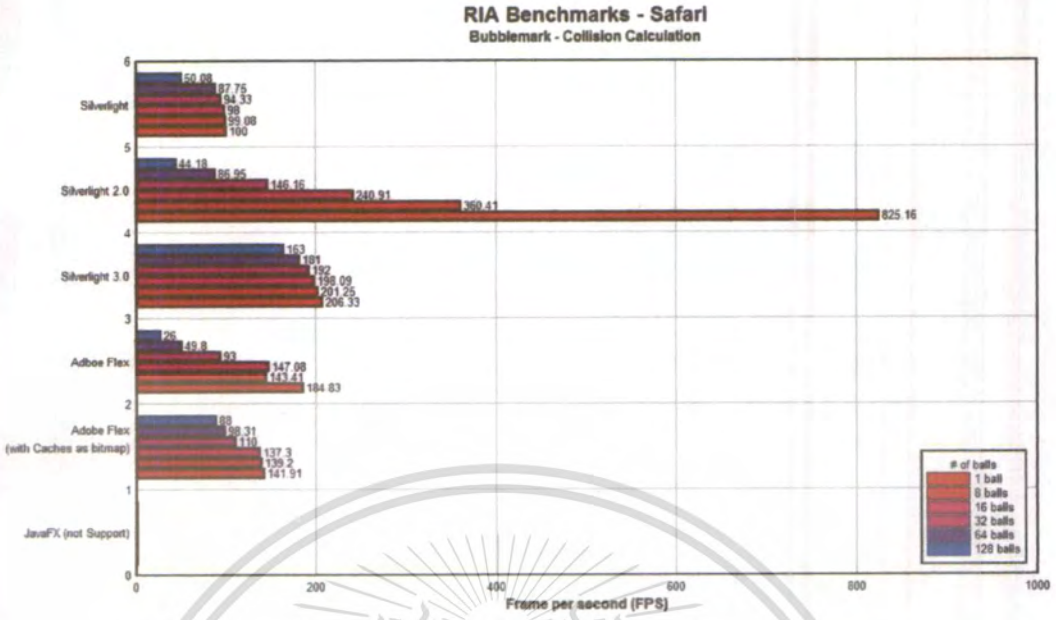


Figure 4.34 : is a graph result of frame rate while running a bubblemark test on Apple Safari 4.04. which show how much frame rate had been showed while doing an experiment. Graph result show that Silverlight 2 got the highest scores in this test while comparing with other tools in this test .

4.1.9 Experiment 3: graph conclusion

CPU USAGE (Percent)						
RIAs Tools / Browser	Silverlight	Silverlight2.0	Silcerlight3.0	Adobe Flex	Adobe Flex (Cache as Bitmap)	JavaFX
Internet Explorer 8	9.16	88.75	33.75	9.16	8.12	28.41
Mozilla Firefox 3.6	11.66	87.91	32.41	78.83	33.75	32.75
Google Chrome 4.1	87.33	87.41	26.66	44.41	28.08	32.83
Opera 10.10	10.58	88.83	30.41	79.16	23.5	Not Support
Apple Safari 4.04	39.83	82.75	38.25	81.5	48	Not Support

Figure 4.35 : Average result in CPU usage (Percent)

Calculation Performance (Frame Per sec)						
RIAs Tools / Browser	Silverlight	Silverlight2.0	Silcerlight3.0	Adobe Flex	Adobe Flex (Cache as Bitmap)	JavaFX
Internet Explorer 8	42.16	92.83	196	32	64	423.5
Mozilla Firefox 3.6	85.3	175	209.91	102.33	195	542.16
Google Chrome 4.1	35.08	175.58	186	51	129.08	526.5
Opera 10.10	69.5	477.41	206	95.5	103.75	Not Support
Apple Safari 4.04	94.33	146.16	192	93	110	Not Support

Figure 4.36 : Average result in Calculation Performance (FPS)

From all experiment result we had done, we choose the result from 32 balls test in each of Internet browsers and all RIAs tool to compare the performance (as shows in Figure 4.35 and Figure 4.36). We can briefly conclude that JavaFX is done great in both of CPU resource management and calculation performance while Internet Explorer done best in CPU resource management and Mozilla Firefox done best in calculation performance.

4.2 Discussion

The test results in CPU usage and Calculation power of all Internet browser and computer system, JavaFX do best in almost every test which the CPU usage result compared to other RIA is quite high in all Browser and computer system but FPS we get is about 40-50% different in almost every test. The reason that the test results come out like this may because of JavaFX is quite new and it likely fixed the problems in using the resources too much from previously released version of JavaFX. JavaFX is quite new so some browsers might not support it, such as Safari and Opera. Developer must focus on this point before development any application with JavaFX because Safari is the default browser of Mac OS, so most people using Mac OS will not be able to use the RIAs on Safari if it's created from JavaFX.

The next RIA tool we want to focus is Adobe Flex CAS (Caches as bitmap). The result is not too high but the stability of result is very great while other RIAs result are reduction when added the balls into the experiment, while FLEX CAS maintain the value of the FPS quite steadily in most of internet browser. The exchange is more CPU usage, but also can be considered good.

Silverlight 2 has given the best result in FPS in processing 1 ball. But start to decline sharply when added the ball and also raise up CPU the CPU usage. So Silverlight 2 might suit with the application which use low calculation power but need a very smooth animation.

Silverlight 3 didn't give us an impressive result of frame rates or CPU usage but the frame rates is very stable and the CPU usage is very low compared to frame rates it produces in each experiment. It might good for application which is not use much of calculation process but need a good presentation to user.

4.2.1 Internet browser performance

Internet Explorer has done great in CPU resource management which we can see from most of the experiment result. No matter we add how much ball into the test, the CPU usage in Intenet Explorer isn't raise up at all. Even it doesn't provide much good result in calculation performance but it still the best choice for internet browser for running RIAs.

For Firefox performance result, Firefox done great in calculation performance but there's a little problem in CPU resource management. Whether we get how much frame rates in Firefox while running, the CPU process is also raise up too.

Google Chrome performance is also tested in this survey. The result is quite near the Firefox performance but CPU usage is the highest among all Internet browser which might not good when running many application in the same time.

4.2.2 Different computer system performance

The different between computer systems is also the topic that we want to focus. In this survey we choose 3 computer systems that have different performance to be a reference and to proof the most discussed topic such as "The better performance, the better result is" and "Each RIAs tool need Graphic card to improve their performance". The result has proof that graphic card isn't need for each RIAs applications. The result of frame rates which is directly related to the graphic rendering has show that the difference is not more than 5% in each RIAs tools and each internet browser. Also the high performance computer has proved that it's better than normal computer system about 30-50% different.

The applicability of results is quite limited of course to the performance of animations. However, this is correlated to many practical patterns of modern RIA applications such as drag-and-drop and layout transitions. The result can be a reference for any user or developer to choose what is the best suit for them to use in their own environment.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

This survey gives user an idea how to choose the RIAs tool which suits their requirement in term of calculation performance and CPU resource management. We designed an experiment to test the performance of RIAs tools by using benchmark application called "Bubblemark". The tests were run in different environments such as internet browsers and computer systems to find out which RIAs tool is the best. All results were put into result table and then we brought the result table to average and make a graph result. Both graph results will show the different between the performances in RIA tool, one will show CPU resource management and another one show calculation performance of each tool.

The result of this survey will give an idea to user about what is the best RIAs tool in terms of calculation performance and CPU resource management. Users of this survey such as RIAs developers or normal users who interest in RIAs tool will get the idea how to choose the best RIAs tool which suits their requirement and use it to make RIAs as good as they want.

5.2 Recommendation

There are many areas where we can improve this survey. For example:

1. More application for compare the RIA tools, which will show us more feature, such as 3D rendering or animation rendering speed.
2. More testing computer system to see different in calculation performance such as computer which is Quad-Core or computer system that has Hyper treading technology.

References

Andrea Simmons. "RIA101 : A Primer for Marketing Agencies and Multimedia Developer"

prof. Joanicjusz Nazarko. "Basic benchmarking concepts and conditions for their introduction in the corporate and public sectors"

David Tondreau, Building "Rich Web Applications With OpenROAD".

Tom Noda ,Shawn Helwig. "Rich Internet Applications Technical Comparison and Case Studies of AJAX, Flash, and Java based RIA"

Microsoft Silverlight

[http://en.wikipedia.org/wiki/Microsoft_Silverlight]

[http://faq.programmerworld.net/programming/silverlight.html]

[http://riathoughts.com/blog/silverlight/what-is-silverlight/]

Adobe Flex

[http://en.wikipedia.org/wiki/Adobe_Flex]

[http://www.avoka.com/ad_flex/flex_home_pradeep.shtml]

[http://www.onflex.org/ted/archive/2008_...hive.php]

[http://www.javabeat.net/tips/138-what-is-adobe-flex.html]

[http://www.adobe.com/flex]

CacheAsBitmap

[http://polygeek.com/171_flex_cacheasbitmap]

Java FX

[http://en.wikipedia.org/wiki/JavaFX]

[http://javafx.com/about/overview/]

[http://www.itwriting.com/blog/1084-suns-javafx-is-launched-another-go-at-applets.html]

[http://www.rialabz.com/hire-javafx-developers-india.html]

Benchmark

[http://en.wikipedia.org/wiki/Benchmark]

[http://www.qualityresearchinternational.com/glossary/benchmarking.htm]

Software and System Requirements for JavaFX Technology

[http://java.sun.com/javafx/1/reference/system-requirements-1-2.html]

Comparison of RIA technologies with standard Internet applications

[http://www.ecreo.eu/en/our_offer/ria_application]

Appendix A

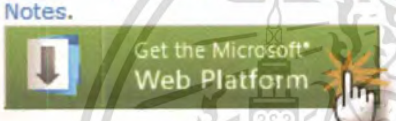
Download Microsoft Silverlight

Go to http://silverlight.net/getstarted to download

1st in the webpage select the Get the Microsoft Web Platform icon to next webpage

Get Started Building Silverlight 3 Applications

1. Install the developer tools for Silverlight. If you don't have Visual Web Developer, download both using Web Platform Installer. This will install the SDK, developer runtimes, and Visual Studio project templates. If you already have Visual Studio, download the tools directly (the Silverlight 3 SDK is also available as a standalone download). For additional information, read the Overview and the Silverlight 3 Release Notes.



2. Install Microsoft Expression Blend 3 + SketchFlow Trial
Expression Blend 3 + Sketchflow allows designers to graphically create UI's for Silverlight 3 applications.

3. Install Deep Zoom Composer
This tool allows you to prepare your images for use with the Deep Zoom feature in Silverlight 3.

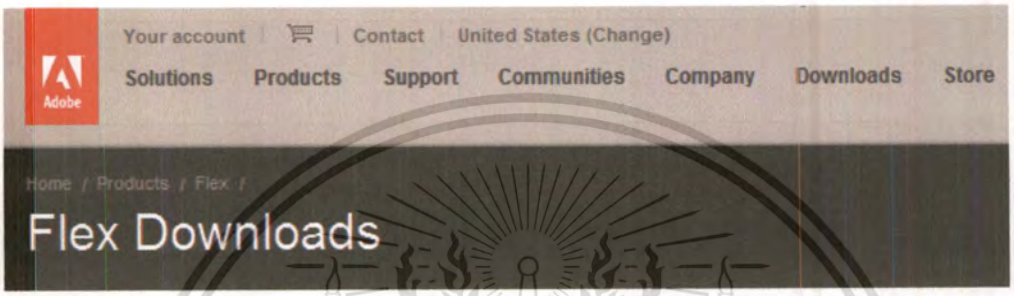
4. Download Silverlight Toolkit
This Toolkit is a Microsoft project containing Silverlight controls, components and utilities that can be downloaded and used in your Silverlight applications. It includes full source code, samples and tests.

5. Download .NET RIA Services
Microsoft .NET RIA Services simplifies the traditional n-tier application pattern by bringing together the ASP.NET and Silverlight platforms. The RIA Services provides a pattern to write application logic that runs on the mid-tier and controls access to data for queries, changes and custom operations.

Appendix B

Download Adobe Flex

Go to http://www.adobe.com/products/flex/flexdownload to download



The following Flex related downloads are available

Flex SDK
Flex Builder 3
Flex Skin Design Extensions

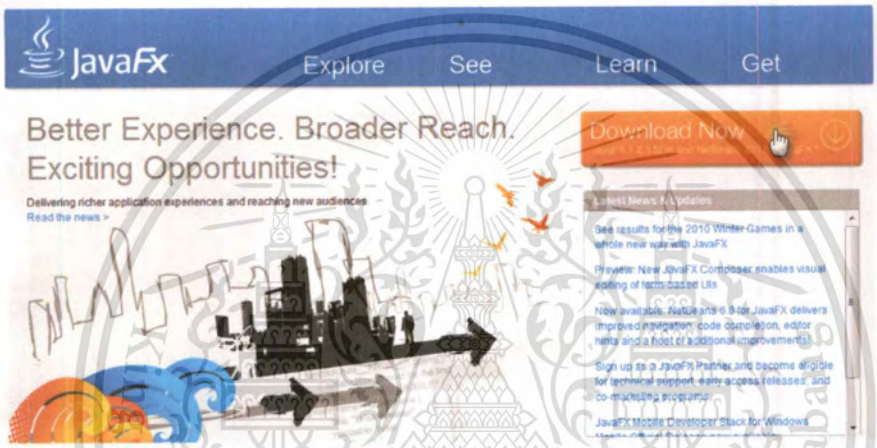
select Flex SDK and select Download now.

Appendix C

Download JavaFx

Go to http://javafx.com/ to download

select Download Now JavaFx 1.2.3 sdk and NetBeans IDE for JavaFx



select list and download