**ABSTRACT**

With the explosive growth in data volume, the I/O bottleneck has become an increasingly daunting challenge for big data analytics in the Cloud. Recent studies have shown that moderate to high data redundancy clearly exists in primary storage systems in the Cloud. Our experimental studies reveal that data redundancy exhibits a much higher level of intensity on the I/O path than that on disks due to relatively high temporal access locality associated with small I/O requests to redundant data. Moreover, directly applying data deduplication to primary storage systems in the Cloud will likely cause space contention in memory and data fragmentation on disks. Based on these observations, we propose a performance-oriented I/O deduplication, called POD, rather than a capacity-oriented I/O deduplication, exemplified by iDedup, to improve the I/O performance of primary storage systems in the Cloud without sacrificing capacity savings of the latter. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic. We have implemented a prototype of POD as a module in the Linux operating system. The experiments conducted on our lightweight prototype implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure by up to 87.9 percent with an average of 58.8 percent. Moreover, our evaluation results also show that POD achieves comparable or better capacity savings than iDedup.

**CHAPTER-1**

**INTRODUCTION**

Data deduplication has been demonstrated to be an effective technique in Cloud backup and archiving applications to reduce the backup window, improve the storage-space efficiency and network bandwidth utilization. Recent studies reveal that moderate to high data redundancy clearly exists in VM (Virtual Machine) [6], [17], enterprise [20], [29], [37], [8] and High-Performance Computing (HPC) [19], [27] storage systems. These studies have shown that by applying the data deduplication technology to large-scale data sets, an average space saving of 30%, with up to 90% in VM and 70% in HPC storage systems, can be achieved [6], [37], [27]. For example, the time for the live VM migration in the Cloud can be significantly reduced by adopting the data deduplication technology [46]. The existing data deduplication schemes for primary storage,n such as iDedup [37] and Offline-Dedupe [8], are capacityoriented in that they focus on storage capacity savings and only select the large requests to deduplicate and bypass all the *\_ B. Mao and S. Wu are with Xiamen University, Xiamen, Fujian, China.* *E-mail: fmaobo,* [*suzheng@xmu.edu.cn*](mailto:suzheng@xmu.edu.cn) *\_ H. Jiang and L. Tian is with the Department of Computer Science and* *Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA.* *E-mail:* [*jiang@cse.unl.edu*](mailto:jiang@cse.unl.edu) *\_ This is an extended version of our manuscript published in the Proceedings* *of the 28th IEEE International Parallel & Distributed Processing Symposium* *(IPDPS’14), Pheonix, AZ, USA, May 2014.* small requests (*e.g.*, 4KB, 8KB or less). The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved. However, previous workload studies have revealed that small files dominate in primary storage systems (more than 50%) and are at the root of the system performance bottleneck [29], [23], [4], [40], [27]. Furthermore, due to the buffer effect, primary storage workloads exhibit obvious I/O burstiness [23], [4]. From a performance perspective, the existing data deduplication schemes fail to consider these workload characteristics in primary storage systems, missing the opportunity to address one of the most important issues in primary storage, that of performance. With the explosive growth in data volume, the I/O bottleneck has become an increasingly daunting challenge for big data analytics [39] in terms of both performance and capacity. Recent International Data Corporation (IDC) studies indicate that in past five years the volume of data has increased by almost 9 times to 7ZB per year and a more than 44-fold growth to 35ZB is expected in the next ten years [41]. *Managing the data deluge on storage to support (near) real-time data analytics becomes an increasingly critical challenge for Big Data* *analytics in the Cloud, especially for VM platforms where the* *sheer number and dominance of small files overwhelm the I/O* *data path in the Cloud [40], [10]*. Moreover, our workload analysis, detailed in Section 2.1, IEEE Transactions on Computers,Volume:65,Issue:6,Issue Date :June.1.2016 2 shows that data redundancy on the critical I/O path is much more pronounced than on the storage devices, largely due to the high temporal locality of small I/O requests. This suggests that, if such redundant I/Os can be removed from the critical I/O path, the performance bottleneck of a primary storage system may be significantly alleviated, if not removed. Thus, we argue that, for primary storage systems in the Cloud, it may be at least as important, if not more so, to deduplicate the redundant I/Os on the critical I/O path for the sake of performance as to deduplicating redundant data on storage devices for the sake of capacity savings. On the other hand, our experimental studies suggest that directly applying data deduplication to primary storage systems will likely cause space contention in the main memory and data fragmentation on disks. This is in part because data deduplication introduces significant index-memory overhead to the existing system and in part because a file or block is split into multiple small data chunks that are often located

in non-sequential locations on disks after deduplication. This fragmentation of data can cause a subsequent read request to invoke many, often random, disk I/O operations, leading to performance degradation. Our preliminary evaluations on the VM disk images reveal that the restore times with deduplication are much higher than those without deduplication, by an average of 2.9*\_* and up to 4.2*\_* [24]. These two problems will be particularly acute with the deployment of the data deduplication technology into the primary storage systems for big data analytics in the Cloud. To address the important performance issue of primary storage in the Cloud, and the above deduplication-induced problems, we propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one (*e.g.*, iDedup), to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select- Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic. More specifically, Select-Dedupe takes the workload characteristics of small-I/O-request domination into the design considerations. It deduplicates all the write requests if their write data is already stored sequentially on disks, including the small write requests that would otherwise be bypassed from by the capacity-oriented deduplication schemes. For other write requests, Select-Dedupe does not deduplicate their redundant write data to maintain the performance of the subsequent read requests to these data. iCache dynamically adjusts the cache space partition between the index cache and the read cache according to the workload characteristics, and swaps these data between memory and back-end storage devices accordingly. During the write-intensive bursty periods, iCache enlarges the index cache size and shrinks the read cache size to detect much more redundant write requests, thus improving the write performance. During the read-intensive bursty periods, on the other hand, the read cache size is enlarged to cache more hot read data to improve the read performance. Thus, the memory efficiency is maximized. The prototype of the POD scheme is implemented as an embedded module at the block-device level and a subfile deduplication approach is used. To examine the net effect of the POD scheme, in our trace-driven evaluation we use the blocklevel traces that were collected beneath the memory buffer cache so that the caching/buffering effect of the storage stack is already fully captured by the traces. In other words, all the small I/O requests in our evaluation are issued from the buffer cache to the block devices after the former has processed the filesystem-issued requests. The extensive trace-driven experiments conducted on our lightweight prototype implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure of primary storage systems without sacrificing the space savings of the latter. Moreover, as an application of the POD technology to a background I/O task in primary cloud storage, it is shown to significantly improve the online RAID reconstruction performance by reducing the user I/O intensity during recovery.

**CHAPTER. 2**

**SYSTEM STUDY**

**2.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMIC FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

**CHAPTER .5**

**SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE**

1. View data owners and authorize.
2. View End users and authorize.
3. View all data owner file details
4. Add Memory and cost details and view the same.
5. View all file and memory transaction of each and every data owner.
6. View download or search req and give permission
7. View all owner and user transactions(Upload and download)
8. View file attackers
9. Find File rank results in chart

**Data Owner**

**1. Register and Login Data owners   
2.Purchase cloud by paying cost and view purchased details  
2.Browse file, Split into 4 blocks and enc each block and Upload.  
3. View all Uploaded Files with digital sign  
4.View Deduplication file from data deduplication and confirm your deletion or share with other data owner  
5. View your files and Update Block contents  
6. View your files and Delete**

**End User**

**Login**

**Data Deduplicator**

Register and Login

1. **View all Data owner files with shared or deduplication verification details**
2. **View users dedulication tried and uploaded details**
3. **View saved memory of the Data Owner after sharing file from other user.**
4. **Find CPU Speed while storing file**
5. **Req Search or download permission from cloud**
6. **View response for download or search**
7. **Search file and Download the file**

**5.2 DATA FLOW DIAGRAM**

**Level -0**

CLOUD SERVER

DATA OWNER

**Data Deduplicator**

End user

**CHAPTER.6**

**SYSTEM ANALYSIS**

**6.1 EXISTING SYSTEM**

The existing data deduplication schemes for primary storage, such as iDedup and Offline-Dedupe, are capacity oriented in that they focus on storage capacity savings and only select the large requests to deduplicate and bypass all the small requests (e.g., 4 KB, 8 KB or less). The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved. However, previous workload studies have revealed that small files dominate in primary storage systems (more than 50 percent) and are at the root of the system performance bottleneck. Furthermore, due to the buffer effect, primary storage workloads exhibit obvious I/O burstiness.

**6.2 Disadvantages of Existing System:**

1. From a performance perspective, the existing data deduplication schemes fail to consider these workload characteristics in primary storage systems, missing the opportunity to address one of the most important issues in primary storage, that of performance.
2. Our experimental studies suggest that directly applying data deduplication to primary storage systems will likely cause space contention in the main memory and data fragmentation on disks. This is in part because data deduplication introduces significant index-memory overhead to the existing system and in part because a file or block is split into multiple small data chunks that are often located in non-sequential locations on disks after deduplication. This fragmentation of data can cause a subsequent read request to invoke many, often random, disk I/O operations, leading to performance degradation.

**6.3 PROPOSED SYSTEM**

To address the important performance issue of primary storage in the Cloud, and the above deduplication-induced problems, we propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one (e.g., iDedup), to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic.

**6.4 Advantages of Proposed System:**

1. POD significantly improves the performance and saves capacity of primary storage systems in the Cloud

**CHAPTER . 7**

**SYSTEM REQUIREMENTS**

# Hardware Configuration

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

# Software Configuration

* Operating System : Windows XP
* Programming Language : JAVA/J2EE

cc

**CHAPTER . 8**

**UML DIAGRAMS**

**8.1 USE CASE DIAGRAMS**

View all Data owner files with shared or deduplication verification details

**Data Deduplicator**

Register and Login Data owners, Purchase cloud by paying cost and view purchased details,Browse file, Split into 4 blocks and enc each block and Upload

View users dedulication tried and uploaded details

View saved memory of the Data Owner after sharing file from other user

View all Uploaded Files with digital sign

Find CPU Speed while storing file.

**Data Owner**

View data owners and authorize,View End users and authorize,View all data owner file details

View Deduplication file from data deduplication

**Cloud Server**

Add Memory and cost details and view the same

View download or search req and give permission

**End user**

View all owner and user transactions,find File rank results in chart.

Req Search or download permission from cloud

Search file and Download the file.

View response for download or search,

**8.2 CLASS DIAGRAMS**

**METHODS**: View data owners and authorize,View End users and authorize,View all data owner file details,Add Memory and cost details and view the same,View all file and memory transaction of each and every data owner,View download or search req and give permission,View all owner and user transactions,find File rank results in chart.

**MEMBERS:**

Cloud server name,password.

**METHODS:** Register and Login Data owners, Purchase cloud by paying cost and view purchased details,Browse file, Split into 4 blocks and enc each block and Upload,View all Uploaded Files with digital sign,View Deduplication file from data deduplication,View your files and Update Block contents.

**MEMBERS:**Owner name,password.

Cloud Server

Data Owner

server

**End User**

**METHODS:** Req Search or download permission from cloud,View response for download or search,Search file and Download the file.

**MEMBERS:**

End user name,password,file name ,secret key, mobile number,email address,pin code,date of birth,gender,select image.

**Data Deduplicator**

**Methods** : View all Data owner files with shared or deduplication verification details ,View users dedulication tried and uploaded details, View saved memory of the Data Owner after sharing file from other user,Find CPU Speed while storing file.

**MEMBERS :**

Data duplicator name,password

**8.3 SEQUENCE DIAGRAMS**

**Data Deduplicator**

**End user**

**Cloud server**

**Data Owner**

Register and Login

Register and Login

Browse file, Split into 4 blocks and enc each block and Upload,View all Uploaded Files with digital sign

Store all Owner and end user registration details

View all Data owner files with shared or deduplication verification details

Add Memory and cost details and view the same,View all file and memory transaction of each and every data owner

View Deduplication file from data deduplication

View users dedulication tried and uploaded details

View your files and Update Block contents

View saved memory of the Data Owner after sharing file from other user

View download or search req and give permission,View all owner and user transactions

find File rank results in chart.

Find CPU Speed while storing file.

Req Search or download permission from cloud

Search file and Download the file.

View response for download or search

File download response

**CHAPTER . 9**

**SOFTWARE ENVIRONMENT**

### OVERVIEW

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine reveled that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from $200 million in 1992 to more than $1.2 billion in 1996.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

### What is a Client Server

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

### Why Client Server

### Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as “SLAVE-MASTER”.

### Time-sharing changed the picture. Remote terminal could view and even change the central data, subject to access permissions. And, as the central data banks evolved in to sophisticated relational database with non-programmer query languages, online users could formulate adhoc queries and produce local reports with out adding to the MIS applications software backlog. However remote access was through dumb terminals, and the client server remained subordinate to the Slave\Master.

### Front end or User Interface Design

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-Based Architecture for achieving the Distributed Concept.

The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

### Communication or Database Connectivity Tier

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity.

The standards of three-tire architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

### Features of The Language Used

In my project, I have chosen Java language for developing the code.

### About Java

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language.
* Java is cohesive and consistent.
* Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

### Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

### Java can be used to create two types of programs

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java –compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

### Features Of Java

### Security

Every time you that you download a “normal” program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a “firewall” between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

### Portability

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed .As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient.

### The Byte code

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

### Java, Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that’s has been generated by the compiler will not corrupt the machine that it’s loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

Overall Description

# Java Source

## Java byte code

# JavaVM

Java

.Class

Picture showing the development process of JAVA Program

Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a. class file, which contains the byte code. The. Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

### Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

### Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

### Compiling and interpreting Java Source Code

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or Sun SARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets.

**Source**

**Code**

**………..**

**………..**

**………..**

**…………**

# PC Compiler

**Macintosh**

**Compiler**

**SPARC**

###### Compiler

**Java**

**Byte code**

**(Platform**

**indepen**

**dent)**

**Java**

**Interpreter**

**(PC)**

**Java**

**Interpreter**

**(Macintosh)**

**Java**

**Interpreter**

**(Sparc)**

**Simple**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

### Object-Oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

### Robust

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time.

Java virtually eliminates the problems of memory management and deallocation, which is completely automatic. In a well-written Java program, all run time errors can –and should –be managed by your program.

### JAVASCRIPT

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser’s display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

<SCRIPTS>..</SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript :

* Validate the contents of a form and make calculations.
* Add scrolling or changing messages to the Browser’s status line.
* Animate images or rotate images that change when we move the mouse over them.
* Detect the browser in use and display different content for different browsers.
* Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

### JavaScript Vs Java

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

* Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
* While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

**ADVANTAGES**

* JavaScript can be used for Sever-side and Client-side scripting.
* It is more flexible than VBScript.
* JavaScript is the default scripting languages at Client-side since all the browsers supports it.

**Hyper Text Markup Language**

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags :

<!-- --> Specifies comments

<A>……….</A> Creates hypertext links

<B>……….</B> Formats text as bold

<BIG>……….</BIG> Formats text in large font.

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<DD>…</DD> Definition of a term

<DL>...</DL> Creates definition list

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM> Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#> Creates headings of different levels

<HEAD>...</HEAD> Contains tags that specify information about a document

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags

<META>...</META> Provides meta-information about a document

<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table

### ADVANTAGES

* A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
* HTML is platform independent.
* HTML tags are not case-sensitive.

### Java Database Connectivity

### What Is JDBC?

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things:

* Establish a connection with a database
* Send SQL statements
* Process the results.

### JDBC versus ODBC and other APIs

At this point, Microsoft's ODBC (Open Database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms.

So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

1. ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
2. A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void \*". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.
3. ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.
4. A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

### Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access.

In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an Intranet, which, for example, connects employees within a corporation, or it can be the Internet.

**JAVA**

**Application**

### JDBC

### DBMS

**Client machine**

**DBMS-proprietary protocol**

**Database server**

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL

**Java applet or**

**Html browser**

**Application**

**Server (Java)**

**JDBC**

## DBMS

**Client machine (GUI)**

**HTTP, RMI, or CORBA calls**

**Server machine (business Logic)**

**DBMS-proprietary protocol**

**Database server**

statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features. JDBC is important to allow database access from a Java middle tier.

### JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

* JDBC-ODBC bridge plus ODBC driver
* Native-API partly-Java driver
* JDBC-Net pure Java driver
* Native-protocol pure Java driver

### JDBC-ODBC Bridge

If possible, use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, the ODBC driver library, and the database client library).

### What Is the JDBC- ODBC Bridge?

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge implements JDBC for any database for which an ODBC driver is available. The Bridge is implemented as the

sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is a joint development of Intersolv and JavaSoft.

### Java Server Pages (JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other’s work.

### Features of JSP

### Portability:

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

### Components

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

### Processing

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

**Access Models:**

A Java Server Pages file may be accessed in at least two different ways. A client’s request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

**Steps in the execution of a JSP Application:**

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
3. JSP engine is program which can understands the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the result is given back to the JavaWebServer and then it is transferred back to the client.

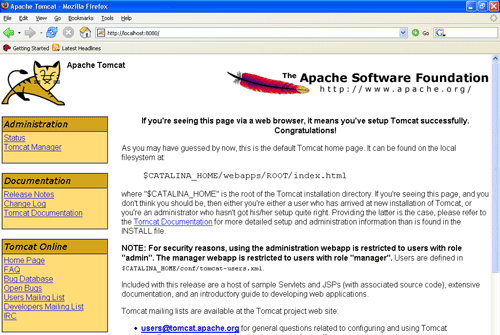
### JDBC connectivity

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

* Perform connection and authentication to a database server
* Manager transactions
* Move SQL statements to a database engine for preprocessing and execution
* Execute stored procedures
* Inspect and modify the results from Select statements.

### Tomcat 6.0 web server

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Weblogic, is one of the popular application server).To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.



**CHAPTER.10**

**IMPLEMENTATION**

**DATA OWNER**

In this module, Data owner has to register to cloud and logs in, the data owner has to purchase the cloud by paying cost to upload the file. While uploading the file the file is divided into 4 blocks and each block is encrypted and uploaded with the Digital Sign. If the uploaded file seems to be duplication the data owner will get the notification from the Deduplication to delete or share the file with the original file present. And will have the authority of updating the blocks.

**CLOUD SERVER**

In this module, the cloud will authorize both the owner and the user. View all the uploaded Files details, has a feature of adding the memory and the cost, will have provide search req from the users. Shows the transactions of uploaded and downloaded.

**END USER**

In this module, the user has to register to cloud and logs in. before the user can search or download the file the user must request for the search and download permission only then user is allowed to search file and download the same .

**DATA DEDUPLICATOR**

In this module, the deduplicator can see the dataowner files and see if it’s duplicate. Send the deduplication log to the corresponding dataowner, and also if the dataowner shares the file deduplicator show how much memory is saved by the dataowner who shared the file. And it shows the speed of the CPU which was during the file storing process.

### CHAPTER.12

### SYSTEM TESTING

### The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**12.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

### 12.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**12.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER . 13**

**CONCLUSION**

SIn this paper, we propose POD, a performance-oriented deduplication scheme, to improve the performance of primary storage systems in the Cloud by leveraging data deduplication on the I/O path to remove redundant write requests while also saving storage space. It takes a request-based selective deduplication approach (Select-Dedupe) to deduplicating the I/O redundancy on the critical I/O path in such a way that it minimizes the data fragmentation problem. In the meanwhile, an intelligent cache management (iCache) is employed in POD to further improve read performance and increase space saving, by adapting to I/O burstiness. Our extensive tracedriven evaluations show that POD significantly improves the performance and saves capacity of primary storage systems in the Cloud. POD is an ongoing research project and we are currently exploring several directions for the future research. First, we will incorporate iCache into other deduplication schemes, such as iDedup, to investigate how much benefit iCache can bring to saving extra storage capacity and improving read performance. Second, we will build a power measurement module to evaluate the energy efficiency of POD. By reducing write traffic and saving storage space, POD has the potential to save the power that disks consume. We will compare the extra power that CPU consumes for computing fingerprints with the power that the storage saves, thus systematically investigating the energy efficiency of POD.

**CHAPTER . 14**

**REFERENCES**

[1] N. Agrawal, William J. Bolosky, John R. Douceur, and Jacob R. Lorch. A Five-Year Study of File-System metadata. In *FAST’07*, Feb. 2007.

[2] A. Anand, S. Sen, A. Krioukov, F. Popovici, A. Akella, Andrea C. Arpaci-Dusseau, Remzi SH. Arpaci-Dusseau, and S. Banerjee. Avoiding File System Micromanagement with Range Writes. In *OSDI’08*, Dec. 2008.

IEEE Transactions on Computers,Volume:65,Issue:6,Issue Date :June.1.2016 14

[3] A. Batsakis, R. Burns, A. Kanevsky, J. Lentini, and T. Talpey. AWOL: An Adaptive Write Optimizations Layer. In *FAST’08*, Feb. 2008.

[4] P. Carns, K. Harms, W. Allcock, C. Bacon, S. Lang, R. Latham, and R. Ross. Understanding and Improving Computational Science Storage Access through Continuous Characterization. *ACM Transactions on* *Storage*, 7(3):1–26, 2011.

[5] F. Chen, T. Luo, and X. Zhang. CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives. In *FAST’11*, pages 77–90, Feb. 2011.

[6] A. T. Clements, I. Ahmad, M. Vilayannur, and J. Li. Decentralized Deduplication in SAN Cluster File Systems. In *USENIX ATC’09*, Jun. 2009.

[7] L. Costa, S. Al-Kiswany, R. Lopes, and M. Ripeanu. Assessing Data Deduplication trade-offs from an Energy Perspective. In *ERSS’11*, Jul. 2011.

[8] A. El-Shimi, R. Kalach, A. Kumar, A. Oltean, J. Li, and S. Sengupta. Primary Data Deduplication - Large Scale Study and System Design. In *USENIX ATC’12*, Jun. 2012.

[9] FIU traces. <http://iotta.snia.org/traces/390>. [10] D. Frey, A. Kermarrec, and K. Kloudas. Probabilistic Deduplication for Cluster-Based Storage Systems. In *SOCC’12*, Nov. 2012.

[11] M. Fu, D. Feng, Y. Hua, X. He, Z. Chen, W. Xia, F. Huang, and Q. Liu. Accelerating Restore and Garbage Collection in Deduplication-based Backup Systems via Exploiting Historical Information. In *USENIX’14*, Jun. 2014.

[12] Garth Gibson. Storage at Exascale: Some Thoughts from Panasas CTO. Interview. May 2011.

[13] B. S. Gill, M. Ko, B. Debnath, and W. Belluomini. STOW: A Spatially and Temporally Optimized Write Cache Algorithm. In *USENIX ATC’09*, Jun. 2009.

[14] A. Gupta, R. Pisolkar, B. Urgaonkar, and A. Sivasubramaniam. Leveraging Value Locality in Optimizing NAND Flash-based SSDs. In *FAST’11*, Feb. 2011.

[15] Y. Hu and Q. Yang. DCD - Disk Caching Disk: A New Approach for Boosting I/O Performance. In *ISCA’96*, May 1996.

[16] Y. Hua and X. Liu. Scheduling Heterogeneous Flows with Delay-aware Deduplication for Avionics Applications. *IEEE Transactions on Parallel* *and Distributed Systems*, 23(9):1790–1802, 2012.

[17] K. Jinand and E. L. Miller. The Effectiveness of Deduplication on Virtual Machine Disk Images. In *SYSTOR’09*, pages 1–12, May 2009.

[18] Stephanie Jones. Online De-duplication in a Log-Structured File System for Primary Storage. Technical Report UCSC-SSRC-11-03, University of California Santa Cruz. May 2011.

[19] S. Kiswany, M. Ripeanu, S. S. Vazhkudai, and A. Gharaibeh. STDCHK: A Checkpoint Storage System for Desktop Grid Computing. In *ICDCS’08*, Jun. 2008.

[20] R. Koller and R. Rangaswami. I/O Deduplication: Utilizing Content Similarity to Improve I/O Performance. In *FAST’10*, pages 1–14, Feb. 2010.

[21] C. Li, P. Shilane, F. Douglis, H. Shim, S. Smaldone, and G. Wallace. Nitro: A Capacity-Optimized SSD Cache for Primary Storage. In *USENIX’14*, Jun. 2014.

[22] M. Lillibridge, K. Eshghi, and D. Bhagwat. Improving Restore Speed for Backup Systems that Use Inline Chunk-Based Deduplication. In *FAST’13*, Feb. 2013.

[23] J. Lofstead, M. Polte, G. Gibson, S. Klasky, K. Schwan, R. Oldfield, M. Wolf, and Q. Liu. Six Degrees of Scientific Data: Reading Patterns for Extreme Scale Science IO. In *HPDC’11*, Jun. 2011.

[24] B. Mao, H. Jiang, S. Wu, Y. Fu, and L. Tian. SAR: SSD Assisted Restore Optimization for Deduplication-based Storage Systems in the Cloud. In *NAS’12*, Jun. 2012.

[25] B. Mao, H. Jiang, S. Wu, Y. Fu, and L. Tian. Read Performance Optimization for Deduplication-based Storage Systems in the Cloud. *ACM Transactions on Storage*, 10(2):1–22, 2014.

[26] N. Megiddo and D. Modha. Arc: A self-tuning, low overhead replacement cache. In *FAST’03*, Mar. 2003.

[27] D. Meister, J. Kaiser, A. Brinkmann, T. Cortes, M. Kuhn, and J. Kunkel. A Study on Data Deduplication in HPC Storage Systems. In *SC’12*, Nov. 2012.

[28] J. Menon. A Performance Comparison of RAID-5 and Log-Structured Arrays. In *HPDC’95*, pages 167–178, Aug. 1995.

[29] D. T. Meyer and W. J. Bolosky. A Study of Practical Deduplication. In *FAST’11*, Feb. 2011.

[30] Y. Oh, J. Choi, D. Lee, and Sam H. Noh. Caching less for better performance: Balancing cache size and update cost of flash memory cache in hybrid storage systems. In *FAST’12*, Feb. 2012.

**Bibliography:**

References for the Project Development were taken from the following Books and Web Sites.

Oracle

PL/SQL Programming by Scott Urman

SQL complete reference by Livion

JAVA Technologies

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

JAVA2 Networking by Pistoria

JAVA Security by Scotl oaks

Head First EJB Sierra Bates

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

HTML

HTML Black Book by Holzner

JDBC

Java Database Programming with JDBC by Patel moss.

Software Engineering by Roger Pressman