World Applied Sciences Journal 31 (10): 1791-1803, 2014
ISSN 1818-4952
© IDOSI Publications, 2014
DOI: 10.5829/idosi.wasj.2014.31.10.558

Selection of Programming Languages for Developing Distributed Systems

[1]Laila Elgamel and [2]Mohamed Sarrab

[1]Software Technology Research Laboratory, De Montfort University, LE1 9BH, Leicester, UK
[2]Communication and Information Research Center,
Sultan Qaboos University, Al-Khodh Muscat 123 Sultanate of Oman

Abstract: Computer is a powerful machine for assisting human beings in different applications. Programming languages are the tool that makes these computers usable. Every programming language has its own weaknesses and strengths whether old or modern, high or low level. The choice of programming language to be used in building distributed systems depends on what sort of program and kind of computer the program is to run on. The paper presents how the best programming language is selected for distributed systems via the following criteria for language selection (Scalability, Concurrency, Reliability, Security,Performance, Fle xibility, Portability, High Integrity and Ease of Use). The paper concentrates on comparing Java and C++ languages in respect to programming distributed systems and also presents the main criteria for choosing the proper languages for the right system.

Key words: Programming language · Java · C++ · Distributed systems

INTRODUCTION

The motivation of computers is to perform quick processing and computation of complex data. Centralized system is model of single computer that has one or more CPUs to process different requests. However, because of the reliability, cost and the wide separate nature of the system parts the centralized model is no longer adequate. To address the centralized system model issuesanother model called distributed systems has been proposed. The idea of distributed systems is that many computers communicate and interact with each other through a common network. The heterogeneous nature, independence and distribution of these computers underlie the needs of software for this distributed system to provide a support of the systemcommon view [1].

At the time of first computers development, programming referred to working with zeros and ones. High level programming languages were not provided and developed overnight. A series of efforts lead to different types of advanced programming languages those are used today. In the 1940s a system called Plankalkul was developed for programming using symbols. Alan Turing was one of the inventers of programing in machine language [2].In the 1950s FORTRAN (Formula Translator) was the most popular and used programming language; Which developed by IBM.Then, in 1958, a language named ALGOL (Algorithm Language) was provided to compete with FORTRAN language. There were different releases of ALGOL but, it was not an acceptable as FORTRAN. In 1960 COBOL (Common Business Oriented Language) was designed to work as large scale programing language in business and government. Which is still in use on a large number of computers today. The BASIC (Beginners All-PurposeSymbolic Instruction Code) was first used in 1964. In 1965, a combination of FORTRAN, ALGOL 60 and COBOL called PL/1 Language was developed for the use of business and scientific purposes.

The earlier concepts and contexts of object oriented programming has been introduced by Simula language which has an impact on the languages to come, such as C++. In 1960s, an important context of programming structure that increase readability and reduce errors called Pascal language has been provided. In 1983, another large and complex language is introduced by U.S. Department of Defense. Smalltalk is a language that combined object oriented and graphical together. [2].

Corresponding Author: Laila Elgamel, Software Technology Research Laboratory, De Montfort University, LE1 9BH, Leicester, UK. Tel: +447533968027.

Distributed programing languages can be classified by a simple scheme. Firstly, they can be classified between logically distributed and logically non-distributed languages. In logically distributed languages, parallel computations such as processes are communicated through messages sending between each other. The computations address spaces do not overlap, therefore the whole program address space program is distributed. However, in a logically non-distributed language, the address space is logically between the parallel units that communicate using the data that stored in the shared address space. There is further partition into other classes of the languages in the two categories based on their mechanism of communication. Firstly, the category that includes rendezvous, atomic transactions,synchronous/asynchronous message passing, multiple communication primitives and objects. Secondly, the category which distinguishes between implicit communication via function results that are used in parallel functional languages, distributed data structures and shared logical variables (parallel logic languages) [3, 4].The paper is based on argumentative and philosophical approach. There is no formal research methods used in this type of research; however, personal knowledge and literature review is presented.

Criteria for Distributed Programming Language Selection: The language selection becomes very an important decision at some stage of the software development project. The selection of the languages in a software development project is usually for the wrong reasons such as: fashion, fear of change, prejudice, commercial pressures, inertia, fanaticism and conformism. Professional software engineers must take their language selection decisions depending on relevant economic and technical criteria as the follows.

- Scale, does the language maintain and support the large scale programs development?
- Modularity, does language supports programs decomposition into suitable units?
- Reusability, does the language supports effective program units' reuse? Ifso, the development of software project can be accelerated through reusingprogram units those are tried and tested [10].
- Portability, does the language support portable code writing? Or does the code port from one platform to a dissimilar platform without majorchange? [10, 11].
- Level does the language support and encourages programmers to think in thehigh level abstractions oriented to the application? Or does the languageforces programmers to think in low level details all the time such as pointersand bits? [10].
- Reliability is the language designed such that the errors of programmingare able to be detected and eliminated as fast as possible? The errors thatare detected during the checks at compile time will guaranteed to be absentin the program running. The errors those are detected during the checks atruntime will be guaranteed to cause no harm other than throwing an exception [10, 11].
- Efficiency, someaspects of object-oriented programming entail runtime overheads, such asclass tags and dynamic dispatch. Runtime checks and garbagecollection are costly and slowing the program down at unpredictable times. In case of criticalparts of the program must be highly efficient, does the language allow them to be tuned by using low level coding, or by calls to procedures written inlowerlevel language? [10].
- Readability, language enforces cryptic syntax, short identifiers, default declarations and absence of type information makes it difficult to write readablecode [12].
- Data modeling does the language provide types and associated operations,that are suitable for representing entities in the relevant application area?does it allow programmers to definenew types and operations that accurately model entities in the applicationarea? [10]
- Process modeling does the language provides control structures that aresuitable for modeling the behavior of entities in the relevant application[10].
- Writability on the other hand primarily focuses on consistency and simplification of statements [12].
- Ease of use is a more general criterion. This can be subjective in naturedepending on factors such as user experience, familiarization on the language,or even its direct impact to solve the problem. This criterion is usually associated with the user interface and how easy it is to use the language with the development environment [12].
- Performance this criteria is considered with the speed of program executionand speed of compiler execution.
- Extendibility relates to the possibility of developing the language and itsimplementation, existence of function libraries, class libraries,... etc [13].

- Continuity of the manufacturer, language continuity, implementation continuity, existence of international standards for defining the language, conformity of implementation by following standards, existence of other manufacturers for that language.

Java as a Distributed Programming Language: In 1990s Sun Microsystems provided language referred as Oak for the market of embedded systems. The aim of this language is to be used in the development of software that would support electronics products of consumer e.g. handheld PDA units. This language was just modified a C++ compiler. This language is originally Jana language that has carried a number of the C++ benefits and some object oriented languages. The change to online services caused by the Oak group in Sun Microsystems when they incorporated the provided programming language into a Web browser leads to the language name to be changed and later after, the first Web browser started to be capable to run Java software was produced. The browser was released in 1995 and called HotJava. Nowadays Web has an active document which is executed Java applets. Both Microsoft and Netscape are licensed this technology to be used in their Web browsers and the language became more and more runaway success [8].

Properties of the Java Language: Many of the properties of the Java language are present in other object orientedlanguages, the Java rapid adoption and sheer popularity by programmers demonstrate that Sun Microsystems has the right mix of sophistication and functionality.Many C++ features e.g. the dramatic increase in the complexity of the software andmultiple inheritance. Since that Java has many of its origins in C++, so a numberof the original C and C++ language hang-ups are removed, such as direct memoryaccess, pointers and multiple inheritance. Moreover, Java was originally designedfrom the beginning for supporting the Internet and the World Wide Web to makemore an attractive for network programming [8]. The followingare some of the most important properties of Java:

- Object Orientation,The programmers of object oriented languages deal with objects, classes and methods. The programmers have found that writing an object oriented language code is more productive.
- SimplicityJava is a very simpler object oriented language to learn.Java has no pointers but it has object references that used to access another object. Multiple inheritances are not used by Java; while classes can be inherited byanother class, but they cannot be inherited from a second. This, helps to keep the coding simpler.
- Automatic Garbage CollectionJava has introduced a new concept. When declaring a newobject instance, whereas the Java Virtual Machine (JVM) automatically allocatesthe needed memory.In the case that the object is no longer required, the object reference can beassigned a null value and automatic thread of the garbage collection will reclaim thememory silently for later use, without the known of the programmer for when andhow this occurs.
- PortabilityJava is a portable due to that it is hardware neutral Java software and operatingsystem is capable of running on different platforms with no need of any sourcecode modification or software recompilation.
- Multi-threaded Programming, the main feature of multi-threaded language is the concurrent processing, but using shared memory for thedata and application code.This enables threads to interact with each other and conserve memory to providea collaborative work if required.
- Security, the security model of Java language helps it to become a network developers attractive choice. In fact, Java code might be implicitly trusted to execute with no causing of security breach or damage.
- Internet Awareness, Java offers a rich and fullyfeatured networking API which provides a Java developer with a consistent interfacenevertheless in which platform are running on. The combination that providedby Java language of input/output streams and network classes help Java to become efficient to program and easy to use [8].
- Distributed, Java uses a classes collection in the networked applications to facilitates the distributed applications building. In Java the use of Uniform Resource Locator (URL)class, an application easily allowed to access a remote server. and the classes arealso offered for building socket level connections.
- Interpreted, Java is interpreted language, therefore if a machine has a ported a Java interpreter then themachine can run the Java applications growing body.
- RobustJava does not allow the direct pointer arithmetic or manipulation that makes a Java program impossible corrupt data or overwrite memory. Also, instead of explicit memory freeing, Java uses a garbage collection [14].

- Architecture-Neutral,Java compilercompiles the Java source code into Java bytecode instructions then these sequenceof instructions are interpreted by the Java interpreter, whereas the architecture neutrality for each new architecture is achieved in the Java interpreter implementation.
- High-Performance, achieving high performance is always considered by Java developers as one of thevery important initial design goals. In the application of Java the performance is notachieved for a fully compiled language similar to C and C++.
- Dynamic, Java is interpreted language, thus Java language is an extremely dynamic language.Java environment is able to extend itself at runtime using classes linkingwhich might be located on network remote servers such as the Internet [8, 14].

The Role of Java in Distributed Object Computing: Simplicity, reliability and architecture neutrality are the original design motivations of Java language. The possibility of Java as a language of Internet programming has been realized and also Java supports multithreaded operations, networking and security. Those features are supportive for the development environment of powerful distributed application. The following are some Java features those are distributed applications interest.

- Interfaces, the context of the interface is to describe the queries, messages and operations that an object class is able of servicing, with no information is provided on the implementation abilities.
- Platform Independent, The Java code can be compiled by Java compilers into platform independent byte codes. The compiled and produced sequence of Java byte codes can be executed on Java Virtual Machine (JVM) and any platform with JVM.
- Network Support and Security, the Java programming language (API) has multilevel support for the communications of the network. Also the sockets of low level can be established and built between agents and the protocols of data communication are able to be layered ontop of the socket connection.
- Runtime Environment and Remote Transactions, Java facilitates the system elements distribution through the network, it helps inmaking it simply and easy for the system elements receiver to verify that it is notpossible to compromise the local environment security. The main point that Java makes it simple and easyfor the network communication sockets to be created, extended and manipulated.This environment capability makes it easy to add data encryption and user authentication to establish and build secure network links.
- Multithreading Support, the concept of multithreading is associated with the operating systems but Javahas brought this innovative concept into the programming language level. Therefore,the multithreaded agents generating ability has become a Java fundamental feature [15].

Applications of Java Network Programming: In some cases it is impractical to run and execute systems those are large and complex on a single machine. There are many reasons for this such as a required taskis very complex which requires a number of CPUs working concurrently together in order to achieve the required result in a reasonable amount of time [8].

Design, there are three distinct components of the system design as follows:

- Client,the main objective of the client software (processing node) is to process and request data units continually from the server which is one of the scheduling nodes.
- Remote Interface, IT is a standalone application that interacts and communicates via TCP/IP and enables the administrator to completely control allsystem scheduling nodes and problems. The Java RMI communications technology is used for passing messages among the server and remote interface
- Server,the clients are processing nodes and the server is scheduling node which is the entire distributed system engine, that controls subordinate servers. A server is allowed to be added to the distributed system during the systemrunning and also is allowed to be removed from the system with no effect onthe system stability or causing any type of running problems [16, 17].
- Communication, the system communication is based on the combination of ordinary Java sockets and Java RMI. RMI is a type of facility that is built-in in Java which enables theinteraction with objects those are actually running on remote hosts in JVMs ona network. Which allows to avoid the designer needs to worry about the issue ofthe low level communication. Data files, those might be large, are transmitted viaordinary sockets, where it is more efficient than RMI [17].

Advantages of Java Distributed System Usingrmi: The following are the advantages of Java distributed system using RMI:

- Object Oriented, RMI is not only for predefined data types but it can also pass arguments of fullobjects and return values. Which means that the complex types can be passed aswell e.g. a hash table object in Java can be passed as a single argument. The clientdecompose has to be in existing RPC systems such an object into primitive datatypes, ship those data types and the recreate a hash table on the server. RMIallows the objects shipping directly over the wire without any extra client code.
- Mobile Behavior, RMI has the ability to move behaviors or a class implementations from server toclient and client to server.
- Design Patterns, The objects passing technique in distributed computing is the object oriented technology full power e.g. two and three tier systems. If the behavior can be passedthen the object oriented design patterns can be used in the solutions. The differenttypes of the object oriented design patterns based on the behaviors for the power;without complete objects passing.
- Safe and Secure, A built-in Java security technique is used by RMI which support the safety ofparticular system in the implementations downloading. The security manager is usedby RMI which is defined to protect network from any possible hostile downloadedcode and systems from ant hostile applets. The severe may refuse downloading anyimplementations at all.
- Easy to Write and Easy to Use, RMI makes the creation and write of a remote Java servers very simple and Java clients as well are simple which access those servers. In fact the remote interface isa Java interface. The Java simplicity supports thatservers are written very easy for distributed object systems those are full-scale andto quickly growing up software prototypes and early software versions for evaluationand testing. Because of the RMI programs are written easily, they are also maintained easily.
- Connects to Existing Legacy Systems, Java's native interface (JNI) is used by the RMI to interact with existing systems. Using JNI and RMI client can be written in Java and existing server implementation can be used. Using RMI/JNI connecting with existing serversallows any server parts rewrite and get needed Java benefits in new code.
- Write Once, Run Anywhere, the Java code is written once run anywhere and the RMI is part of Java. All RMIbased system is 100% portable which can run in any JVM, as is RMI/JDBC system. If RMI/JNI is used for the interaction with an existing system,the written code using Java Native Interface (JNI) will be compiled and run in anyJVM.
- Distributed Garbage Collection, the feature of distributed garbage collection is used by the RMI to collect remoteserver objects which are no longer needed and referenced by any network clients.The distributed garbage collection supports defining the required number ofserver objects due to the fact that they can be removed or dropped when they areno longer required to be accessible by the network clients.
- Parallel Computing, RMI is multi-threaded which allows the servers to exploit the threats of the Javalanguage for client requests better concurrent processing [18].

C++ as a Distributed Programming Language: In 1970s, C language was augmented for the creation of a new programming language by the team of BjarneStroustrup at Bell Laboratories in (AT&T, USA). The name of C language was changed to C++ after its internal use in 1983 at AT&T. The same team of BjarneStroustrup created C++ programming language tosupport the simulation projects implementation in efficient way and anobject oriented. The earliest releases of the language were called C with classes and later C++ derived. The C++ is the increment operator in C. In 1989 the committee of American NationalStandards Institute (ANSI) was founded to standardize the C++ language. Foravoiding the confusion caused by the dialects variety, the main aim was to have asmany software developers and compiler vendors as can agree on the language unifieddescription. In 1998 the C++ language was approved by the international organization for standardization (ISO) (ISO/IEC 14882) [19].

Characteristics and Advantages of C++: C++ programming language has two main characteristics:

- C++ provides enhanced version of C and an easier to use version.
- C++ includes classes to facilitates an object oriented programming language.

C++ is a hybrid language and is not a purely an object oriented language. because it has the C programming language functionality whereas, it has all the available features in C language. Therefore, the characteristics of C++ language are:

- Portable programs for many platforms.
- Universally usable modular programs.
- Efficient that is closed to the machine programming.
- The most of the existing C source code can be used in C++ programs. C++ supports object oriented programming concepts e.g. (Data abstraction, Data encapsulation, Inheritance, Polymorphism).

Many language features were added to C++ e.g. exception handling, templates and references. Even though these different features of the language are important for program efficiency and are not strictly object oriented programming features [20].C++ language has some advantages over other programminglanguages. The following are the most remarkable once:

- Object oriented programming, enabling programmers to design applications by orientate programming to objectsas a communication between objects instead of structured sequence of code.
- Portability, where the C++ code can be compiled in any type of computer and operatingsystem with no need of any changes.
- Brevity, Comparing to other languages the C++ written code is very short, while the usedspecial characters is preferred to key words, saving some effort to the programmer.
- Modular programming, In C++ programming language the body of the application are allowed to madeup of several files of source code which are compiled separately and then can be linked together.
- Compatibility, C++ is backwards compatible with the C language. WhereasCcode can be included in a C++ programs easily with no need of any changes.
- Speed, the C++ compilation resulted code is very efficient, because of its indeed to dualityas low level and high level language and to the language reduced size [21].

Making C++ a Distributed Programming Language: To support the development of multimedia applications using the multi service storage architecture (MSSA) in a real time context and distributed, a PC++ has been developed as persistent programming language where it is as extended C++ with persistent classes. The PC++ persistent class is similar to C++ classes because it has all the features of class with providing additional features those are very important for multimedia applications. PC++ supports data persistency by using the MSSA for persistent storage using the MSSA PC++ supports data persistency and the data structures stored may manipulated by programmers across many special purpose in a distributed environment servers. Using a remote object invocation (ROI) mechanism PC++ offers distribution transparency. A remote object invocation (ROI) mechanism is supported to provide users the illusion that programs are still executing in a centralized environment. The programmer does not need to know where the actual objects are located in the distributed system,what needed is a means by which invocations operation can be sent to particularobjects wherever they reside. The main aim of PC++ is tailored particularly tomultimedia applications [22].

- The Object Model, the client/server model is supported by PC++. This model is very useful in thedistributed systems design and analysis. The main aim of this programming styleis to support distributed services collection those are available on workstations orcomputers networks. Every abstraction or resource is represented by an object. Aset of objects are used by the server to provide service. In this kind of model, many worker or server processes are assigned to or created for each object to handle its requests invocation. Every process is restricted and bounded to the specific object. The operation invocation should be allowed by the programmer on a remote object in similar way such as on a local object in order to offer distribution transparency.Thus, for an object that is a located on the server side, named the serverobject, a corresponding object that is created on the client side, referred as the clientobject. A client object has similar interface as the corresponding server object. In PC++ whenever a client object operation is invoked, the mechanism of ROIwill pass this invocation to its corresponding server object. As has been mentionedthat the server object process accepts the request and execute the operations on theclients behalf. The other alternative technique is mapping the object locally to theclient and execute the operation there [22].

- The Remote Object Invocation mechanism, the ROI mechanism helps in providing distribution transparency. Using the underlying remote procedure call (RPC) mechanism helps to make a client objectinvocation become its corresponding server object invocation. On the client side,the mechanism of ROI includes the client objects definitions. Despite, that theclient object has similar interface as the corresponding server object, it has different implementation. The implementation of the client object is in such a waythat every operation on the server side is actually an RPC to a procedure. On theserver side, the mechanism of ROI involves group of procedures every procedureis corresponded to an operation [22].

Building a Distributed Systems Using Mace: Mace is a source to source compiler and is C++ language extension for creatingdistributed systems in C++. Mace easily combines events, aspects and objects tosimultaneously address and control the issues of analysis, failures, concurrency andlayering. While these are the main ideas of all programming language, Mace hastwo main advances.

- The unified in one development environment for the diverse elements those arerequired to build a high performance and robust distributed systems.
- The definition of a language extension to write distributed systems allow therestriction of the waysthose systems can be built [23].

Concrete Benefits of Mace, the Following Are Some Concrete Benefits of Mace:

- Mace enables programmer to concentrate on the description of every distributed system layer as a reactive state transition system.
- The embedded semantic information in the system specification is used byMace for automatic generation of the needed code of failure detection.
- Mace also supports automatic profiling of individual causal paths the sequenceof communication and computation between a distributed system nodes thoseare corresponding to higher level operations.
- Maces state transition model allows distributed systems practical checkingof implementations to find both aliveness bugs and safety.

Mace has been used and deployed by many researchers at handful, MSR-Asia, HPLabs and UCSD in different universities worldwide [24].

Building a Distributed Systems: Mace is an event driven, Mace is a programming language that is reactive for building and creating distributed systems in C++. The compiler of Mace uses a hybrid implementation specification Mace/C++ and translates its source into the implementation of C++. Then it can be linked with the Mace runtime and the applicationof the users is able to run on the internet. The familiar C++ environment supportsdevelopers for incorporate into the existing work environment andreuse existing applications and libraries with Mace.

MaceMChelps the unmodified Mace implementations of the simulated environmentto be automatically testing. The simulated environment enables MaceMC to expose implementations to many other scenarios, for simplifying the bugs finding task.MaceMC is used in the Mace implementations to find numerous bugs, which isa part of the development cycle for the developers of Mace. Performance issuesand problems are very difficult for automatically detection, due to the absent of blackand white boundaries.

The Challenges for Building Correct High Performance Distributed Systems, distributed systems must be able to deal with the widely varying latencies, nodereboots, disconnections, network errors and throughputs. The following are the challenges for building andcreating correct high performance distributed systems [25].

- Programming Languages and Abstractions, the force of theprogrammers to choose between and control over performance and simple expression.
- Runtime Variations, Race conditions are known as programmers bane. Mace uses of atomic events abstraction to help in avoiding race conditions problems.[23, 25].

Selected Comparison Criteria: The study has concluded that the distributed systems basic challenges and problems are Scalability, Flexibility, Performance, Dependability and Transparency. These, also seen as a distributed system desired properties. Therefore the chosen criteria (properties) are:

Scalability is one of the important distributed system goals and should consider at least following three dimensions.

- Scalability, respect to size, where, resources and users are able to be added to system.
- Scalability with aspect to administration, whereas the proper and easy system management is allowed even if the system is split widely between different organizations.
- Scalability with respect to geographic area, whereas the resources and the users might be lied at different locations (geographically).

The maximum weighting of the scalability is 9 because it is the first distributed systems challenges and the most important distributed systems characteristic to increase the number of users and resources [11].

Concurrency is very important for system efficiency and behavior. Concurrency is one of the main sources of complicity in system analysis, design andverification. The maximum weighting for the concurrency property is 8 due to that it is very important in improving overall performance of the distributed system. This is because in the absent of concurrency, the performance will suffer dueto the fact that every request will be processed sequentially as a list [11, 26].

Dependabilityisone of the distributed operating systems fundamental requirement and thatincludes security, consistency and fault tolerance. Thus, the following criteriaare needed to build dependable distributed systems:

- Reliability, a reliable distributed systemmust be designed to be as fault tolerant as possible. Fault tolerance is themeaning of making and supporting the system function in presenceof faults [27]. The maximum reliability weighting is 7 because it is one of the important criteria in building a dependable distributed system.
- Security, the main distributed systems issues is application security. Hence thesystem must be secure, to ensure that only the required users are allowedto perform a particular operations [28]. The maximum security weighting is 7 due to the fact thatsecurity is already one of major issues and problems that may have animportant effect on the building of dependable distributed system.
- Performance is the most important features, where every system must aimand strive for maximum performance. Note there isa direct conflict with other desirable distributed system properties such asscalability, dependability, security and transparency and can easily effect thesystem [29]. The maximum performance weightingis 6 due to the fact that the speed of the program execution is very importantfeature to perform a distributed system.
- Flexibility, aflexible distributed system is a configurable distributed system that canexactly provide the services those programmer and users' needs [26]. Themaximum flexibility weighting is 5 due to it is important in the possibilityfor adding a new system to the overall system.
- High integritysystem plays major roles in power management, defense systems, transportation and communications. The highintegrity offers manage in design, raise flexibility and improve functionality [11]. The maximum integrity weighting is 5 due to it is important in increasing flexibility.
- Portabilityis the ability of an application transformsfrom one hardware or software platform to another. The platform enablesa language to run on different operating systems platforms [11]. The maximum portability weighting is 6 due to it is effecting on distributedsystem openness.
- Ease of use, distributed systems tend to be complex. The tools involved in building thesesystems usually have a steep learning curve. The language should help the users learn how to use the it. The language should be easy to teach, learn and use [11].The maximum ease of useweighting is 3 due to that the language might be complex and does not haveany effect on the distributed system.

Comparison of Java and C++ Languages Against Criteria: The rates those will be given to each language criteria is out of 10.

Scalability:

- The scalability is an important factor in the distributed systems construction for multithreading in server systems. Java and its JVM have become a scalable using the J2EE patterns and many other interesting technologies platform. Rating:8

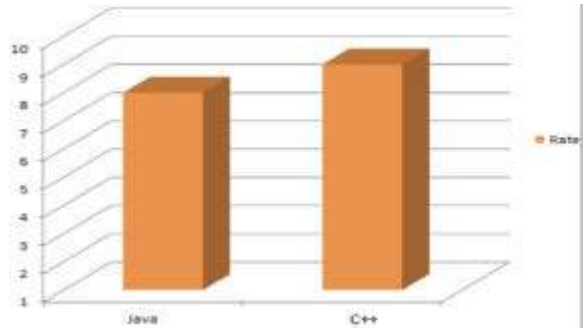

Fig. 1: Comparison of languages against scalability

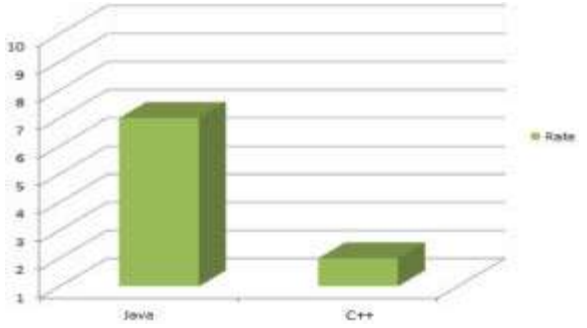

Fig. 2: Comparison of languages against concurrency

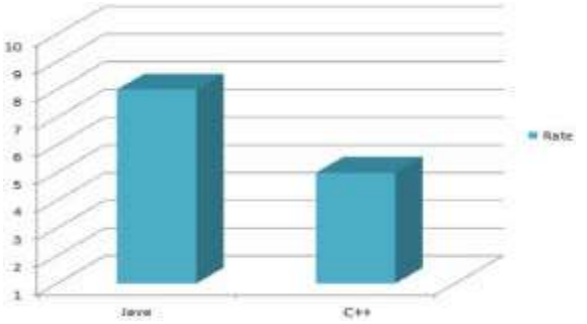

Fig. 3: Comparison of languages against reliability

- C++ adds generic programming and object oriented features to the C language. This can significantly help in the improvement of C++ programs abstraction level relative to C programs, which helps to make the language more scalable. Rating:9

In my opinion, the scalability of C++ programming language is better than Java as shown in Figure 1, due to the fact that C++ supports arbitrary classes multiple inheritance. However, in Java a class can implement multiple interfaces but it can be derived from only one class, thus, it can be said that it supports multiple types inheritance, but only the implementation single inheritance [30].

Concurrency: Concurrency supports, code construction with parallel processing or multiple threads:

- Java offers a built in features for handling threads. Rating:7
- C++ does not support direct concurrency. This is usually supported by operating system library. Rating:2

The concurrency of Java is better than C++ as depicted Figure 2, because Java supports multiple threads of control. e.g. Unix OS uses the concept of processes heavily and a process is able to split into different parts, these parts are executed concurrently using the fork() command. Java network programming and Distributed computing [8].

Reliability:

- Java language requires the information specification such as type specifications, the omission of which can make a program unreliable. Rating: 8
- C++ language improves the characteristics of C language for supporting system reliability with other features such as encapsulation and improved expression (Reinhardt 2004). Rating:5

The reliability of java is better than C++ as described in Figure 3, due to the fact that Java automatically produces specifications instead of using redundant specifications. Java supports runtime check [11].

Security: Java is considered to be secure compared to other languages for the following reasons:

- The compiler of Java language catches more errors at compile time.
- Java does not use direct memory pointers thus it is impossible to have accident memory references that belongs to other the kernel or programs.
- Finally, Java programs run in Java Runtime Environment. Rating:6

The design of C++ language did not consider enough security support for online programming (website), Rating: 3 [31]

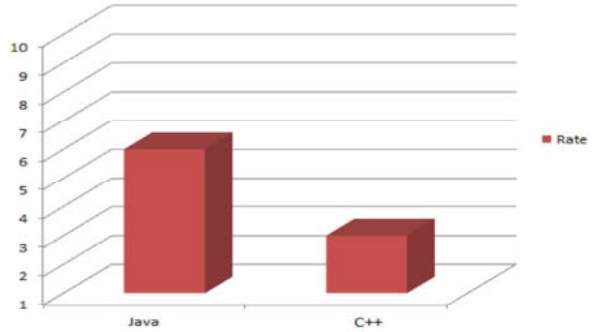

Fig. 4: Comparison of languages against security

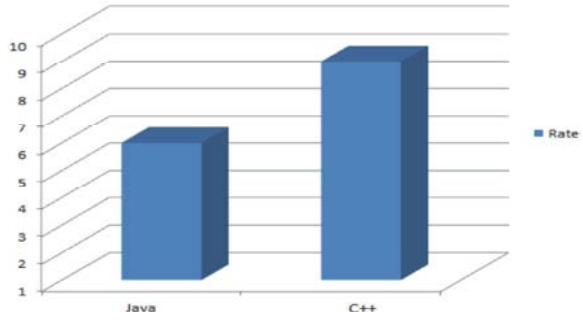

Fig. 5: Comparison of languages against performance

Security is one of the important criteria for selecting programming language for constructing distributed systems. As can be seen in Figure 4, that Java has the rate 6 and C++ has rate 3. It can be concluded that, the security in Java is better than C++, because Java uses the sandbox approach, which means that untrusted code that may include classes downloaded over a network within a Web browser, are inserted in the sandbox and needed to meet certain expectations [8].

Performance:

- Java applications do not achieve the fully compiled language performance compared with C++ applications. However, for many applications, including graphics intensive applications, the Java performance is more than adequate. Rating:6.
- The design of C++ was strongly concentrates on executable performance. The compilers of C++ offer different levels for optimizing executable code. The C++ performance together with templates and classesconstitutes a powerful tool for applications, including numerical computing [32]. Rating:9.

As illustrated in Figure 5, that Java has the rate 6 and C++ has rate 9. The performance of C++ is better than Java, due to the fact that running a compiled Java program extremely requires the running of the Java Virtual Machine JVM. However, the running of a compiled C++ program does not require any external applications.

Flexibility:

- Java language is flexible enough for supporting many different development and deployment environment.

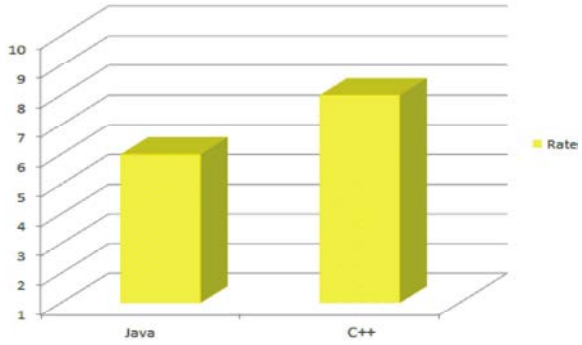

Fig. 6: Comparison of languages against flexibility

Extensions are easily made and distributed because the language is built upon the concept of an object oriented foundation. Rating:6

- The C++ is a flexible programming language which fully supports the template programming, OOP and finally traditional procedural programming. The classes in C++ are more flexible than the classes in Java e.g. C++ has class overload operators [32]. Rating: 8.

As illustrated in Figure 6 thatflexibility of Java and C++ has the rate 6 and 8respectively. In my opinion, C++ is more flexible than Java because C++ isa template programming supportive and it was designed to have the object orientedlanguages flexibility [32].

Portability:

- Java was designed for fully support of portability. The Java compiler produces a byte code that is a platform independent. Then the produced byte code is translated to native machine code at runtime using a given platform Java Virtual Machine [32]. Rating:9

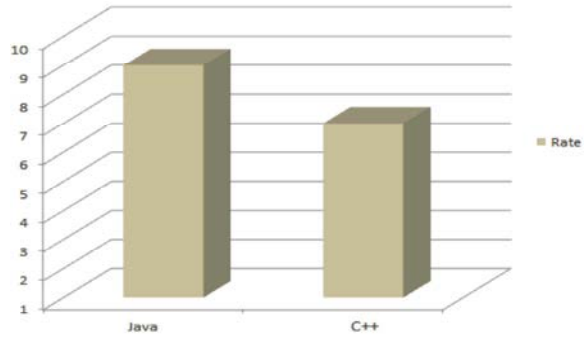

Fig. 7: Comparison of languages against portability

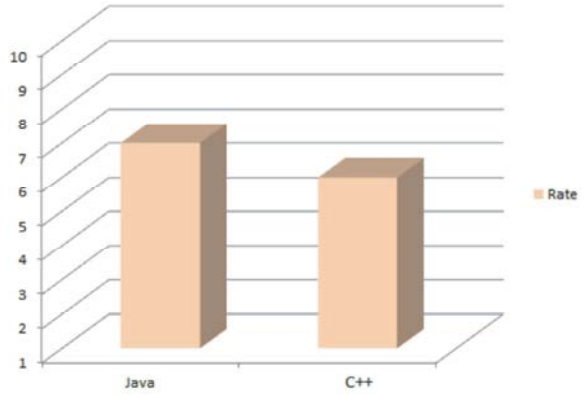

Fig. 8: Comparison of languages against high integrity

- C++ supports the dependencies encapsulation, this feature facilitates portability. C++ tools and tool sets are also widely available on many platforms [31]. Rating:7

As shown in Figure 7 that the portability of Java has the rate 9 and C++ has rate 7. The portability of Java language is better than C++, due to the fact that Java has very important characteristic of being able to run on different platforms e.g. Os2, Linux and MS-Windows. However, C++ is able to execute itself only in the same environment where the program is written [11].

High Integrity:

- Java language supports and encourages high integrity software for ensuringdata integration while sending or receiving data. Rating:7
- C++ supports high integrity software for increasing integration and performance. Rating: 6.

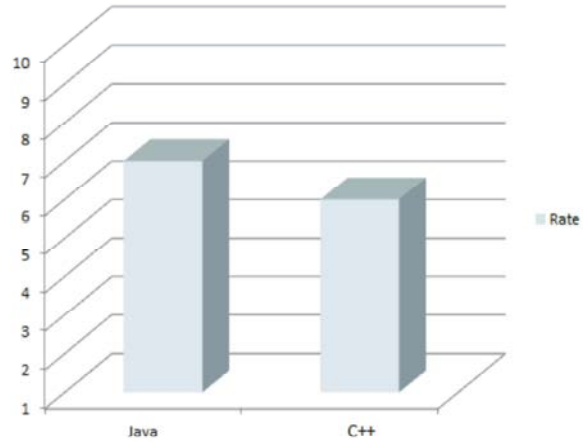

Fig. 9: Comparison of languages against ease of use

As depicted in Figure 8 that high integrity of Java has the rate 9 and C++ has rate 7. The high integrity of Java is better than C++, because Java language supports high integrity software for ensuring data integration while sending or receiving data. Where, Java has many features that support the development of high integrity real time e.g. data accessibility, maintainability and reusability.

Ease of Use:

- Java is an easy OPP language to use and learn; The syntax of Java is very simple. Rating:7.
- C++ is complex object oriented language to use and learn compared to Java; The syntax of C++ is complicated especially for the first using time [11]. Rating:5.

Figure 9 shows that Java has the rate 7 and C++ has rate 5. In my opinion, the Java language is simpler than C++, because Java is a simpler language to learn. Moreover, Java language does not use the concept of pointers in which memory can be accessed. Instead it uses object references to access another object. The coding of Java simpler, which is very important in all types of application specially in networking applications. However, C++ is complex object oriented language in use and learns comparing to Java [8].

Comparison Table: The above criteria rating are summarized in below table along with the associated ratings.

Criteria	Java	C++
Scalability	8	9
Concurrency	7	2
Reliability	8	5
Security	6	3
Performance	6	9
Flexibility	6	8
Portability	9	7
High Integrity	7	6
Ease of Use	7	5
Total	64	54

The above table showed the criteria's and rates of each language (Java and C++).

Conclusion and Future Work: The paper discussed the criteria by which a computer programming language can be selected for developing distributed systems. The essential aspects of each criteria were discussed in detail. Therefore, the main characteristics which affect the building of a good distributed system has been concluded such as: resources sharing, openness, scalability, transparency, fault-tolerant and concurrency.The paper has provided the history, classification and techniques of distributed programming languages. Therefore, many criteria for selecting distributed system has been founded such as, scale, modularity, reusability, portability, performance, level, reliability, efficiency, readability, data modeling, process modeling, writability, ease of use, performance, extendibility, continuity, security, concurrency, integrity and flexibility.After comparing these criteria with the main characteristics considering the basic challenges and problems of distributed systems, the following criteria has been concluded as a key of building successful distributed systems: Scalability, security, flexibility,portability, reliability, high integrity, concurrency, ease of use and performance. The paper has discussed the chosen criteria against Java and C++ languages.This research provided the path to build good distributed systems.Infuture work, this type of research needs to include more object oriented programming language and supports the comparison of the selected language with case studyexamples or practical experiment.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Amelia Platt and Dr. Ali H. Al-Bayatti for their valuable comments, constructive criticism, scientific support and guidance.

REFERENCES

1. Tari, Z. and O. Bukhres, 2001. "Fundamentals of Distributed Object Systems", The CORBA Perspective, John Wiley. US.
2. Chen, Y., R. Dios, A. Mili, L. Wu and K. Wang, 2005. "An empirical study of programming language trends", Software, IEEE, USA, 22: 72-79.
3. Bal, H.E., 1990. Programming distributed systems, USA, Silicon Press,
4. Bal, H., J. Steiner and A. Tanenbaum, 1989. "Programming languages for distributed computing systems", ACM Computing Surveys, 21: 261-322.
5. AIX, "Aix version 4.3 communications programming concepts", chapter 8 remote procedure call', URL http://www.unet.univie.ac.at/aix/aixprggd/progcomc/toc.htm,2004.
6. Golub, M. and D. Jakobovic, "An overview of distributed programming techniques", Proceedings of the H ypermedia and Grid Systems, MIPRO, pp: 215-220.
7. Ahmed, S., 1998. Corba programming unleashed, USA, SamsIndianapolis, IN.
8 Reilly, D. and M. Reilly, 2002. "Java Network Programming and Distributed Computing", MA: Addison-Wesley Professional.
9. Hoang, B., Dcom: Overviewandarchitecture, http://sern. ucalgary. ca/ Courses/ CPSC/ 547/ F98/Slides/Hoang/DCOM.html,December, 2004.
10. Watt, D.A., 2004. Programming Language Design Concepts,New York, Wiley.
11. Aldrawiesh, K. and A. Al-Ajlan, 2009. "Selecting the best object-oriented programming language for developing Distributed Computing Systems", International Conference on Computer Engineering & Systems, ICCES2009, pp: 440-446.
12. Jison, A., 2011. "What programming language criteria will you choose?", Tuldok System: Technical Blog of a Frustrated Programmer, Word Press.com.
13. Al-Sukairi, A., 2003. "Fundamentals of programming languages", King Fahd University of Petroleum and Minerals. Information and Computer Department.
14. King, K., 1997. "The case for java as a first language", Proceedings of the 35th Annual ACM Southeast Conference, pp: 124-131.
15. Peter, G., 2009. "Distributed object computing by java, Information Technology Portal" www.peterindia.net,

16. Page, A., T. Keane, T. Naughton and M. Ireland, 2004. "Adaptive scheduling across a distributed computation platform", In 3rd International Symposium on Parallel and Distributed Computing, pp: 141-148,USA,IEEE Computer SocietyWashington, DC, 2004.
17. Page, A., T. Keane and T. Naughton, 2005. "Bioinformatics on a heterogeneous java distributed system", In Proceedings of the 19th IEEE/ACM International Parallel and Distributed Processing Symposium, pp. 181-184, Denver, Colorado, USA. IEEE Computer SocietyWashington, DC, 2005.
18. Farley, J., 1998. Java Distributed Computing, O'Reilly Media.
19. Breedlove, T.W. and R.L. Albert, 2008. C++: An Active Learning Approach, Jones and Bartlett Publishers, Burlington, USA, 2008.
20. Prinz, P. and U. Kirch-Prinz, 2001. A Complete Guide to Programming in C++, Jones and Bartlett Publishers, Burlington, USA.
21. Amaya, S., 2001. "C++: Information: A brief description", cplusplus.com,
22. Wu, Z., 1993. "Making C++ a distributed programming language", Proceedings of the Fourth Workshop on Future Trends of Distributed Computing Systems, pp: 228-233, Lisbon, Portugal.
23. Charles, K., 2008. "Systems and language support for building correct, high performance distributed systems", PhD thesis, University of California, San Diego, USA,
24. Killian, C., J. Anderson, R. Jhala and A. Vahdat, "Life, death and the critical transition: Finding liveness bugs in systems code",In Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI), pp. 243-256, ACM, New York, In NSDI, 2007.
25. Killian, C., J. Anderson, R. Braud, R. Jhala and A. Vahdat, 2009. "Building distributed systems using mace",IEEE Ninth International Conference on Peer-to-Peer Computing (IEEE P2P'09 - Sept), pp: 91-92.
26. Deshpande, N. and S. Kamalapur, Distributed Systems, India, Technical Publications Pune, 2009.
27. Redwine, E. and J. Holliday, 1998. "Reliability of distributed systems", http://www. cse.scu. edu/~jholliday/REL-EAR.htm, Santa Clara University.
28. Nadiminti, K., M. Dias and R. Buyya, 20087. "Distributed systems and recent innovations: Challenges and benefits", http://www.cloudbus.org/ raj/papers/InfoNet-Article06.pdf, 2008.
29. Kuz, I., M. Chakravarty and G. Heiser, 2008. "Distributed systems", The University of New South Wales, School of Computer Science & Engineering, COMP9243 Week 1(10s1), http://www.inf.puc-rio.br/~noemi/sd-10/intro-notes.pdf, 2008.
30. Vanier, M., 2011. "Scalable computer programming languages", California Institute of Technology, http://www.cs.caltech.edu/mvanier, 2011.
31. Reinhardt, F., 2004. "Use of the C++ programming language in safety critical systems", Master's thesis, University of York, UK,
32. Einarsson, B., 2005. Accuracy and reliability in scientific computing, USA, Society for Industrial and Applied Mathematics.