Performance Computation Of Object Oriented Programs Through UML

Vipin Saxena[1] *Deepa Raj*[2]

Abstract

Nowadays, different kinds of processors are appearing in the business market, therefore it is necessary to judge the performance of the processors over the different kinds of programming languages. In this context, the present paper deals with the evaluation of the performance of Pentium IV architecture which is widely accepted in the global business market & performance is computed for object oriented programming languages. The main objective of this paper is to select the best object oriented programming language for long computations available in the software coding. A well known popular approach of modeling i.e. Unified Modeling Language (UML) is used for the design of Pentium IV processor architecture. UML Class, Sequence and Activity diagrams are designed and comparison is represented among the different object Oriented programming languages namely C#, Java and Visual C++.

Keywords: Pentium IV, UML, C#, VC++, Java, Performance Evaluation

1. Related Work

UML [1] is a powerful modeling language used to represent the research problems visually. A lot of literature is available on modeling problems by the use of UML, but limited research papers are reported in literature on applications of UML in the field of computer architecture problems. By the use of UML, software and hardware architecture problems can be easily solved and performance can be judged after modeling of that problem. Real time system based on UML is described by Selic and Rumbaugh [2]. The first represented of UML in the field of telecommunication sector is described by Holz [3]. Drozdowski [4] explained a technique to find out the execution time for distributed application. In [5], tools and techniques for performance measurement of large distributed multi agent system are explained. Architecture of Pentium IV is reported by Alenn Hinton [6]. The computer architecture models which can be used for the further research work are available in [7]. The UML application is also done on web based application. One of the important papers on this is [8]. UML based Vehicle control system is also reported in the literature by Walther *et al.* [9]. OMG is an important active group for inventing the different versions of the UML. The research papers on these are [10, 11] in which group describes the UML diagram based on XML Meta data specification. Performance modeling and prediction tools for parallel and distributed programs are described by Planna *et al.* [12, 13] and these papers also describe customizing the UML for modeling performance oriented applications. Recently Saxena *et al.* [14, 15] proposed the UML model with performance evaluation for the multiplex system for the process which are executing in the distributed environment and UML model with performance evaluation of the Instruction Pipeline model, respectively.

2. Background

Let us first explain the process which may be the group or block of instructions of program, macro, sub programs and subroutines. For defining the process, there is a need

[1]Reader & Head, Dept. of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow.

[2]Lecturer, Dept. of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow.

of the processing element. The processing element is defined as a stereotype and generally used to handle the concurrent process executing in the parallel and distributed environments. The famous approach to handle the concurrent processes is Torus Topology [7]. The following figure 1 shows the definition of processing unit. The Class Diagram of process is represented in figure 2.

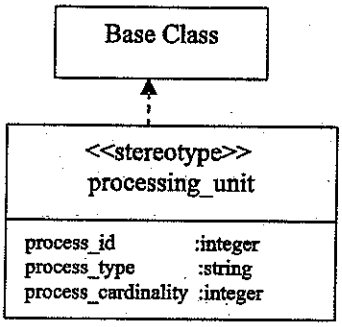

Figure 1 : Representation of Processing Unit

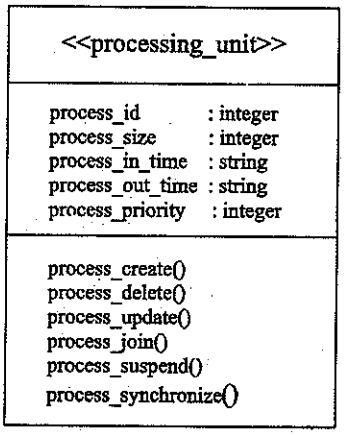

Figure 2 : UML Class Definition of Process

The instance of the process is defined by the use of object xyz which is shown in figure 3.

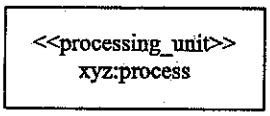

Figure 3 : Instance of Class

The set of the instances of the class process is modeled by the use of multiple objects which is shown below:

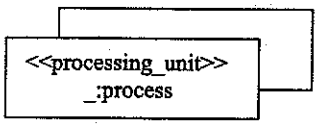

Figure 4 : Multiple Instances of Object

In _:process shows the name of multiple objects. By the use of above definition of process, in this paper, blocks of instructions are considered as a process and three types of object oriented language namely VC++ C# and JAVA are selected for judging the performance of these object oriented software programs for the blocks of instructions on the Pentium IV processor architecture, which is widely accepted processor architecture by software Industries.

C# is an Object Oriented programming language developed by Microsoft as part of .NET initiative. It is applicable for writing application in hosted and embedded system. VC++ is also an integrated development environment developed by Microsoft. It uses Microsoft Foundation Class Library (MFC) standard, which wrap the Windows API in C++ classes and provides a default application framework. On the other hand, Java is an object oriented language and many of the software designs are coded with the help of the Java language. In the present paper the performance of these three programming languages is observed on the Pentium IV architecture system by proposing a model through the UML. The main aim of this paper is to select the best object oriented programming language for writing the software codes for long computations purpose which also saves the execution time. The complete UML Diagram is designed for execution of instructions of a program. UML class diagram, UML sequence diagram and UML activity diagram are also given in the paper.

3. UML Modeling of Pentium IV

Let us consider the Pentium IV processor architecture as shown in figure 5 which is easily available in the market and widely used by the software Industries. In this architecture, the following are the four major sections:

a) In Order Front End

It is a part of machine that fetches the instruction that is to be executed next in the program and prepare them to be used later in the machine pipeline.

b) Execution Engine

In this part instructions are prepared for execution. It has several buffers which are used for smooth and reorder the flow of instruction and optimize performance as they go down to pipeline.

c) Integer and Floating Point Execution Unit

In this part instructions are actually executed. It has register file which stores the integer and floating point data operand value, and L1 data cache stores most load and store operation.

d) Memory Subsystem

It includes L2 Cache and system Bus, L2 Cache stores both Instruction and data that cannot fit in the execution trace cache and L1 data Cache. Trace Cache is primary or L1 Instruction cache of Pentium IV processor it delivers μops to the Out of Order Execution Logic. Most instructions in a program are fetched and executed from the trace cache. When there is trace cache miss then the net burst Micro architecture fetch and decode instruction from level 2 (L2) Cache. The executed and trace cache takes the already decoded μops. When the complex instruction encountered the trace cache jump into the microcode ROM which then issues the μops needed to complete the operation. After completing the work Front end of machine resumes fetching μops from the trace cache. Then instruction executes in double clocked ALU and produces required result.

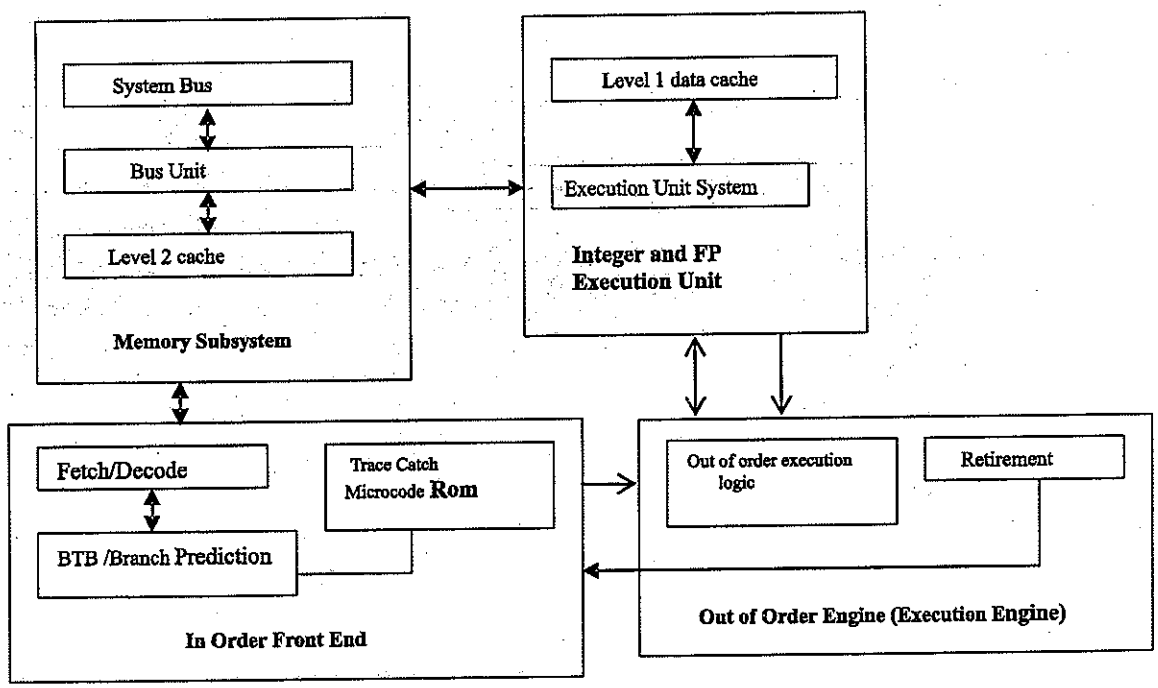

Figure 5 : Representation of Pentium IV Architecture

A. UML Class Model

For the above Pentium IV architecture, UML class model is designed and shown in figure 6 which has following major classes- namely Process, Cache, L2_cache, L1_cache, Tc_fetch, Trace_cache, Queue, Memory_Queue, Float_Queue, ALU,Scheduler, Resource_alloc, Microcode_ROM, Int_Register_file, Float_Register_file, Check_branch, Instruction_Decoder, Int_scedular, Float_schedular. In this class diagram initially process is cached by the cache class which is inherited into the L1_Cache and L2_cache. Instruction_Decoder is used to decode the instructions

one by one and Trace_catch is used to load into Trace_catch. Tc_Fetch is used for fetching instruction from Trace_cache and also check the branching in the set of instructions, if instruction is complex Microcode_ ROM is used to execute the instruction then load into Queue after allocating the resources, if instruction is simple then it goes into the Queue according to their type after allocating resources. Scheduler is used to schedule the instruction according to their type and as per availability of processes Arithmetic Logic Unit (ALU) class is designed to execute the instructions, then result is store in L1_cache. Same procedure will be repeated for the next set of instructions.

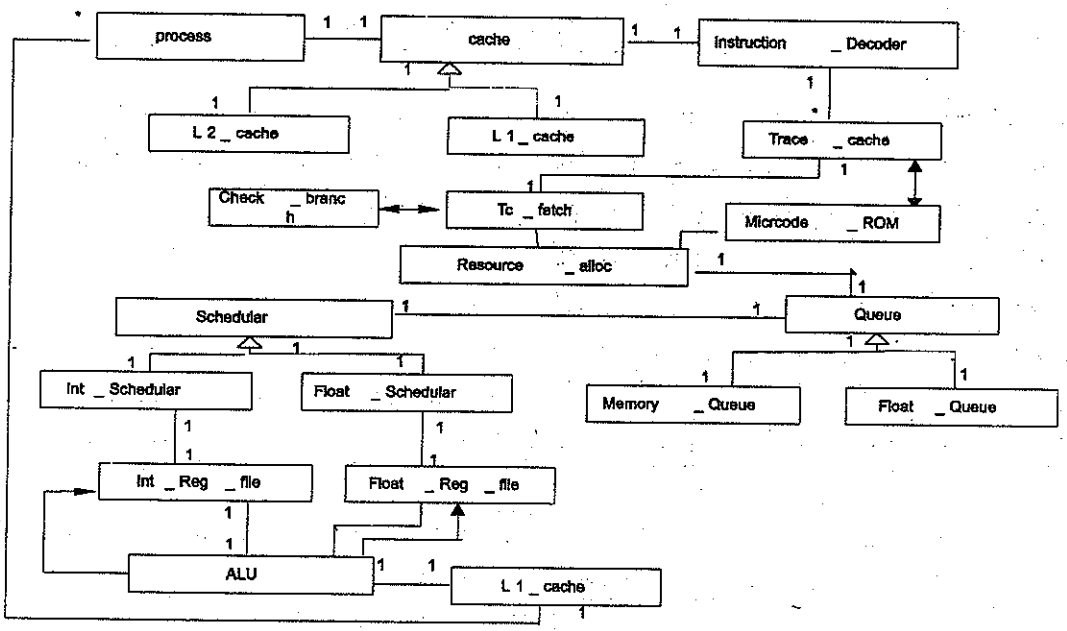

Figure 6 : UML Class Diagram for Execution of Process

B. UML Sequence Diagram

For the Pentium IV architecture, sequences of instructions to be executed are arranged by means of a sequence diagram which shows the dynamic behavior of the modeled Pentium IV architecture and shown below in figure 7. In this diagram one can see that how message passing takes place among the different objects like Process, Cache, Instruction_ Decoder, Tc_fetch, Resource_Alloc, Queue, Schedular, Register_file, ALU. From the diagram one can compute the time that is used for execution of block of instructions.

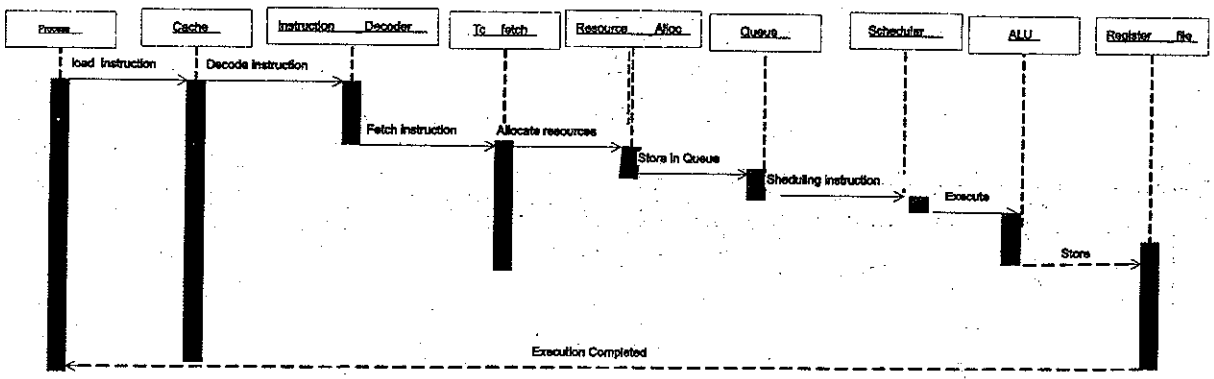

Figure 7 : UML Sequence Diagram for Execution of Process

C. UML Activity Diagram

For the execution of processes through Pentium IV architecture, an activity diagram for process execution is designed and shown below in figure 7. This diagram shows the steps involved in executing a program under Pentium IV.

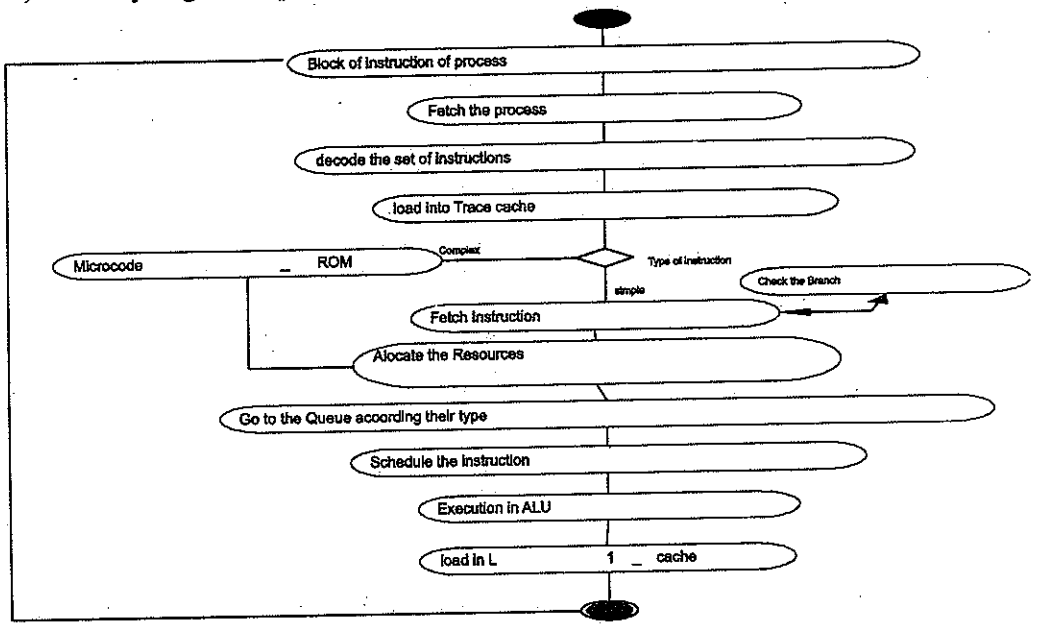

Figure 8 : UML Activity Diagram Process Execution for Pentium IV

4. Numerical Experiments

To judge the performance of object oriented programming languages like C#, VC++ and Java on the Pentium IV processor architecture, it is necessary to check the performance of the designed UML model for Pentium IV processor architecture. Let us consider a sequence of instructions varying from 10^1, 10^2, 10^3, 10^4 & 10^5 lines of codes are to be executed in the five run and results are reported by taking the average technique. The following table 1 gives the total execution time of three object oriented programming languages and time is recorded in milliseconds.C#, VC++ and Java oriented software languages for executing a program of different sizes under Pentium IV processor are specially considered since the most of the software companies are developing the applications by writing software codes in these languages. From the table 1 it is found that for the long computations on Pentium IV, Visual C++ execution time is lesser in comparison of the other two object oriented programming languages namely C# and Java. Therefore, Visual C++ is recommended on Pentium IV for long computations. These results which are recorded in the table 1 are also depicted graphically and shown in the figure 9 and 10 for 10, 10^2, 10^3 and 10^4, 10^5, respectively.

Table 1 : Recorded Execution Time for C#, VC++ and Java

Lines of Code	Time in Milliseconds		
	C#	VC++	JAVA
10	15	9	02
10^2	31	13	15
10^3	203	155	157
10^4	1178	790	1093
10^5	11875	10937	11762

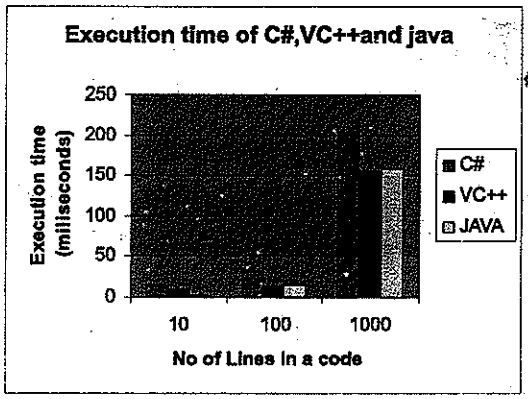

Figure 9 : Execution Time for 10-10^2 and 10^3 LOC

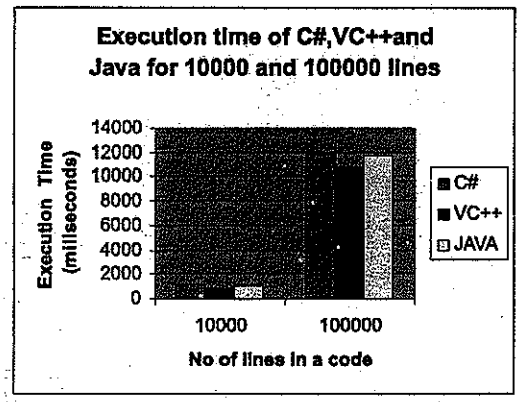

Figure 10 : Execution Time for 10^4 & 10^5 LOC.

5. Conclusion

From the above work it is concluded that UML modeling which is widely used by the scientists/researchers is excellent modeling language used to represent the scientific problem visually. The produced UML design for the Pentium IV architecture is an efficient and different kinds of object oriented programming languages are tested on this architecture and found that for long computations Visual C++ programming language is recommended. The software industries can design the code on Pentium IV architecture by the use of Visual C++. The present work further can be extended in the direction of judging the performance of other processor architectures; day by day coming into the business market around the globe; and can be tested on the multiple object oriented languages.

References

[1] G. Booch , J. Rumbaugh and I. Jacobson, *"The Unified Modeling Language User Guide"*, Addison Wesley, Reading, MA (1999).

[2] B. Selic and J. Rumbaugh, *"UML for Modeling Complex Real Time Systems"*, Available Online Via www.rational.com/Products/Whitepapers/100230.Jsp.

[3] E. Holz, *"Application of UML within the Scope of New Telecommunication Architectures"*, In GROOM Workshop on UML, Mannheim : Physicaverlag , 1997.

[4] Drozowski. M, *"Estimating Execution Time of Distributed application"*, Parallel Processing and Applied Mathematics: 4th International conference PPAM 2001 , LNCS 2328, Springer – Verlag , PP. 137-142, 2002.

[5] A.Helsinger ,R. Lazarus, W. Wright & J. Zinnky, *"Tools and Techniques for Performance Measurement of Large Distributed Multi Agent System"*, proceedings of AAMAS 03 conference, Australia, PP. 843-850.

[6] H.Allen, *"Micro Architecture of Pentium IV processor"*, Desktop Plateform Group, Intel Corporation.

[7] K. Hwang, *"Advanced Computer Architecture"*, McGraw – Hill Series in Computer Engineering , Inc Publishing-1993.

[8] J.Conallen, *"Web Application Architecture with UML"*, Communication of the ACM 42(10), 63-70, 1999.

[9] M. Walther, J. Schirmer , P.T. Flores, A. Lapp, T. Bertram and J. Peterson, *"Integration of the Ordering Concept for Vehicle Control System CARTRONIC into the Software Development Process using UML Modeling Methods"*, in

SAI2001 World Congress Detroit, Michigan, USA, 2001.

[10] OMG, *"Unified Modeling Language Specification"*, Available Online Via www.omg.org , 2001.

[11] OMG, *"XML Metadata Interchange (XMI) Specification"*, Available Online Via. www.omg.org, 2002.

[12] S. Pllana and T. Fahringer, *"UML based Modeling Performance Oriented Applications in <<UML>>2002"*, Model Engineering Concepts and Tools, Springer-Verlag. 2002, Dreseden, Germany.

[13] S. Pllana and T. Fahringer , *"UML based Modeling of Performance Oriented Applications"*, WinterSimulation Conference 2002.

[14] V. Saxena, D. Arora and S. Ahmad, *"Object Oriented Distributed Architecture System through UML"*, IEEE International Conference proceedings on Advanced in Computer Vision and Information Technology, November 28-30 (Sponsored by IEEE Transactions, U.S.A.), 305-310, ISBN:978-81-89866-74-7, 2007.

[15] V.Saxena , D. Raj , *"UML Model of the Instruction Pipeline"*, International Conference on Software Engineering 08 august 30-01September 2008 in Singapur.

Author's Biography

Dr. Vipin Saxena got his M.Phil. Degree in Computer Application in 1991 & Ph.D. Degree work on Scientific Computing from University of Roorkee (renamed as Indian Institute of Technology, India) in 1997.He has more than 12 years teaching experience and 16 years research experience in the field of Scientific Computing & Software Engineering. Currently he is proposing software designs by the use of Unified Modeling Language for the various research problems related to the Software Domains & Advanced Computer Architecture. He has published more than 60 International and National publications.

Deepa Raj got her M.Sc. Degree in Computer Science from J.K. Institute of Applied Physics & Technology, Allahabad Central University, Allahabad. She has more than eight years teaching experience in field of Computer science. Currently she is solving software designs research problems by the use of Unified Modeling Language (UML). She has several outstanding research papers in this field.