Int. J. Advanced Networking and Applications
Volume: 01, Issue: 02, Pages: 125-130 (2009)

Impact of Dual Core on Object Oriented Programming Languages through UML

Dr. Vipin Saxena
Associate Professor, Department of Computer Science
Babasaseb Bhimrao Ambedkar University (Central University)
Lucknow (U.P.), 226025, INDIA
Email–vsax1@rediffmail.com
Deepa Raj
Assistant Professor, Department of Computer Science
Babasaseb Bhimrao Ambedkar University (Central University)
Lucknow (U.P.), 226025, INDIA
Email- deepa_raj200@yahoo.co.in

---------- **ABSTRACT**----------

Nowadays, different kinds of processors are appearing in the computer market, therefore it is necessary to observe the performance of these processors at the early stage of computation of object oriented programs. In this context, the present paper deals with the evaluation of the performance of Dual Core and Core 2 Dual processors architecture for the Object Oriented Programming languages. The main objective of this work is to propose the best object oriented programming language for the software development on these said processors architecture. A well known modeling language i.e. Unified Modeling Language (UML) is used to design a performance oriented model, consisting of UML class, UML sequence and UML activity diagrams. Experimental study is performed by taking the two most popular object oriented languages namely C++ and JAVA. Comparative study is depicted with the help of tables and graphs.

Keywords: UML, Activity diagram, C++, Dual core , Core 2 dua

Paper Submitted: 23 June 2009 Accepted: 11 Oct 2009

---

1. Related Work

In the today's scenario, various researchers are developing the models by the use of popular and powerful object oriented modeling language i.e. UML [5]. A lot of literature is available on modeling problems by the use of UML, but limits research papers are available 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 represented pictorially and performance can be judged after modeling of the problems. Complex research problem can be easily solved by designing the UML models and performance of these models can be judged by message passing technique among the objects. Real time system through UML is described by Selic and Rumbaugh [2]. The first represented of UML in the field of telecommunication sector is described by Holz [7]. This is one of the important papers to select the best programming language for distributed computing system. Drozdowski [9] 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. A modeling for architecture of Pentium IV is reported by Alenn Hinton [11]. The computer architecture models which can be easily used for the further research work are available in [6]. Web based application can also be judged through UML and it is reported in [1]. UML based Vehicle control system is also reported in the literature by Walther et al. [12]. OMG is an important active group for inventing the different versions of the UML. The research papers on these are [3,4] in which group describes the different types of UML diagrams based on XML Meta data specification. Performance modeling and prediction tools for parallel and distributed programs are described by Planna et al. [8, 13] and these papers also describe customizing the UML for modeling performance oriented applications. Recently Saxena et al. [14] proposed the UML model with performance evaluation for the multiplex system for the processes which are executing in the distributed computing environment. UML Model of Instruction Pipeline is also explained by Saxena & Raj [15] in which they observed the performance of object oriented instructions executing in distributed computing environment.

Java is an object oriented platform independent language and many of the software designs are coded with the help of the Java programming language and on the other hand C++ is also an object oriented language generally used for software applications as well as for system programming. In the present paper, the performance of these two

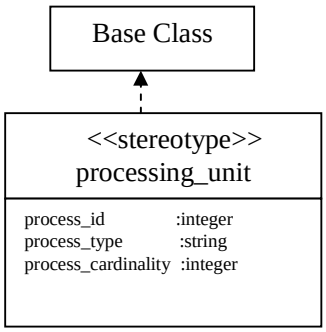
programming languages is observed on the Dual Core and Core 2 Dual processor system by proposing a model through UML. The main aim of this paper is to select the best Object Oriented Programming language for writing the software codes for long computations purpose which also saves the execution time. The complete UML Diagram is designed for execution of instructions of a program for dual core processor. UML class diagram, UML sequence diagram and UML activity diagram are also given in the paper and comparison is shown through tables and graphs.

2. Background

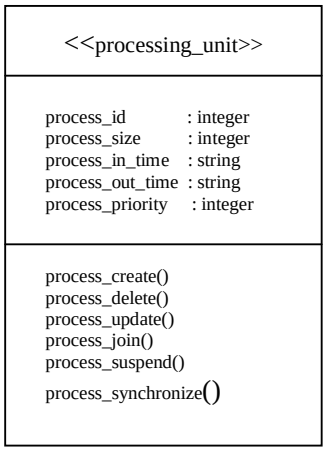
2.1 UML Process Definition

Let us first define 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 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. The following Fig 1(a) shows the UML definition of processing unit. The UML Class Diagram of Process is completely defined in Fig 1(b).

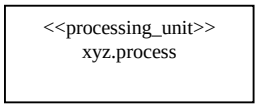
The instance of the process is defined by the use of object .xyz which is shown in Fig 1(c). The set of the instances of the class process is modeled by the use of multiple objects which is also shown below in Fig 1(d).



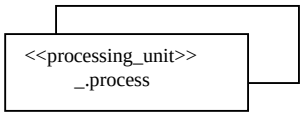
(a) Definition of Processing Unit



(b) Class Definition of Process



(c) Instance of Class



(d) Multiple Instances of Object

Figure 1. UML Class Representation of Process and Processing Unit

In the above figure process shows the name of multiple objects.

2.2 Dual Core Processors

In the today's scenario a different kinds of processors are appearing in the computer market designed by the various hardware companies. To increase the computing speed in respect of single core processors, dual core processors are designed. The dual core is an architecture that refers to a central processing unit (CPU) with two complete execution cores in a single processor. These two cores, theirs caches and cache controller all are built together on a single integrated circuit (IC). Performance of dual core processor is not double as compared to a single core processor but it is significantly better than the single core processor. Since there are two pipelines, two instructions can be executed simultaneously and two processor caches allow more data on processing unit for quick access. The dual core block diagram is represented in Figure 2 which is currently used by most of software developers due to its scalability and time consuming. On the basis of this diagram, the comparative specifications of dual core and core 2 duo processors architecture are recorded in Table 1.

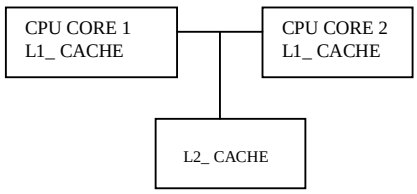


Figure 2. Block Diagram of Dual Core Processor

In a dual core and core 2 duo, there are two CPU cores, one L1_CACHE in each core one L2_CACHE is shared by both the cores mounted on a single chip.

Table 1. Specifications of Dual Core and Core 2 Duo Processor Architecture

Specification Item	Dual Core	Core 2 Dua
Technology	45nm	45nm
Cache	2MB L2	6MB L2
Clock speed	2.70 GHz	3.33GHz.
Processor No.	E5400	E8600
Front side Bus	800 MHz	1333 MHz

3. UML Modeling of Dual Core Processor Architecture

3.1 UML Class Model

Now let us design UML Class model and shown in Figure 3 which consists of thirteen major classes namely process, cache, L1_cache, L2_cache, I_cache, D_cache, ALU, Fetch, I_queue, Decode, Execute, Branch. In this class diagram initially process loads the instruction from memory to both the L1_cache and L2_ cache. Fetch class fetches the instruction from both L1_cache to Pipeline. If there is L1_cache miss then fetch takes place from L2_cache. Decoder is used to decode the instructions one by one and then execution takes place by Execute class. ALU, Branch is a component of Execute class used to execute the instructions, Branch used to check the branch in the instructions. Same procedure repeats for the next set of instructions.

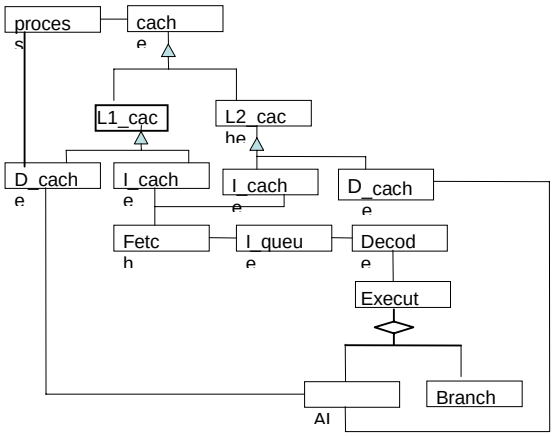


Figure 3. UML Class Diagram for Process Execution for Dual Core Processor

3.2 UML Sequence Diagram

For Dual core and Core 2 Duo processors architecture, execution sequences of instructions are arranged by means of a UML sequence diagram as given in Figure 4. In this diagram one can see that how message passing takes place among the different objects. This figure also shows that how long computation time represented through object life line is required for execution of instructions and arranged for process.

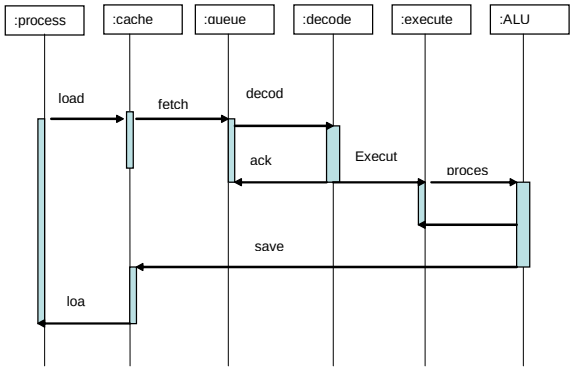


Figure 4. UML Sequence Diagram for Process Execution for Dual Core Processor

3.3 UML Activity Diagram

For the execution of processes on Dual Core and Core 2 Duo processor architecture, UML activity diagram for process execution is also designed and given below in Fig 5. This diagram shows the steps involved in executing a process under both the processors.

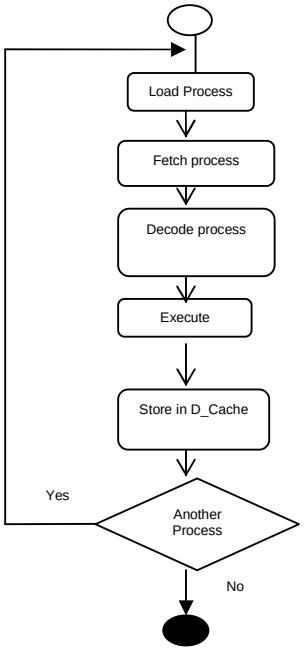


Figure 5. UML Activity Diagram for Process Execution for Dual Core Processor

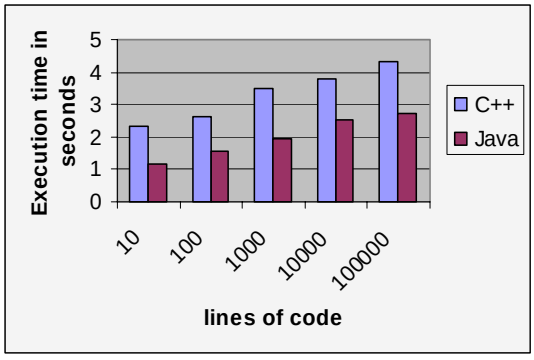
4. Experimental Study and Results

To check the performance of the above UML designed model, a case study is considered by taking the sequence of instructions to be executed in the five run by taking average technique. Sample programs are designed in C++ and JAVA by increasing the sequence of instructions up to 10^5 lines of code. These two object

oriented software languages for executing a program of different sizes under Dual Core and Core 2 Duo processor are specially considered since most of the software companies are developing the applications by writing software codes in these object oriented languages. Therefore, we compare these two object oriented languages on recently used processor. This architecture system is well accepted by most of the software companies and most of the applications are done on this architecture.

A comparative study of execution of instructions for these two object-oriented software languages is observed. The execution time is increasing for these two programming languages as the sets of instructions are increasing. The computed execution timing in seconds for these languages is given in Tables 2-3. Table 2 and 3 are prepared for Dual Core and Core 2 Duo processor architecture, respectively. From the tables, It is observed that JAVA is more suitable object oriented language in comparison of C++ as a number of instruction are increasing, the computation time is much smaller in comparison of C++. It is happening for both the processors architecture. Therefore, JAVA programming language is recommended for long computations.

In the Table 3, average execution time is also recorded for C++ and JAVA on both the Processor's architecture as shown in Table 3 and it is observed that Core 2 Duo processor architecture is better processor architecture in comparison of Dual Core processor architecture since for both the programming languages, the computation time is lower. For getting the quick interpretation, the above tabulated data is also compiled in the form of graph and represented below in Figures 6 & 7 for Dual Core and Core 2 Duo Processors architecture, respectively.



Figure– 6 Execution Time of C++ and Java on Dual Core Processor

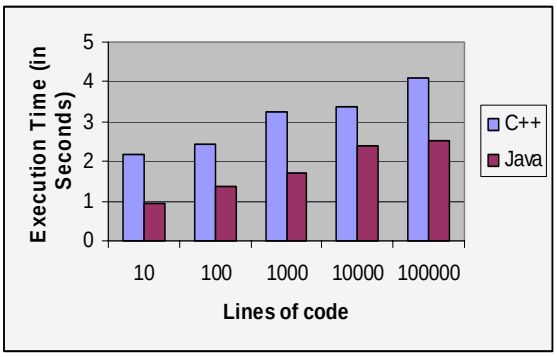


Figure- 7 Execution Time of C++ and JAVA on Core 2 Dua Processor

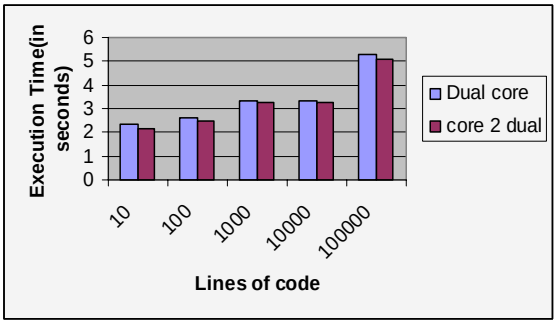


Figure- 8 Execution time of C++ on Dual Core and Core 2 Dua Processor

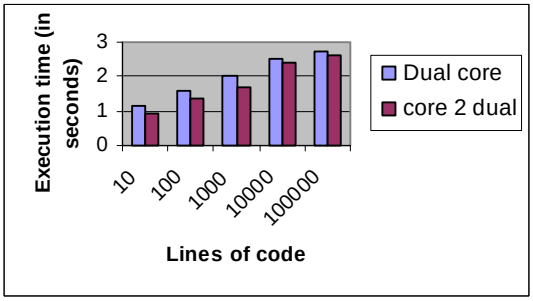


Figure- 9 Execution time of JAVA on Dual core and Core 2 Dua Processor

Table 2: Execution Time of C++ and Java on Dual Core Processor Architecture

Lines of Code	**10**		**10^2**		**10^3**		**10^4**		**10^5**	
P/L	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**
Execution Time in Seconds	2.34	1.15	2.64	1.56	3.49	1.93	3.79	2.51	5.31	2.74
	2.36	1.16	2.64	1.56	3.48	1.95	3.70	2.52	5.31	2.73
	2.37	1.18	2.65	1.54	3.44	1.97	3.72	2.51	5.32	2.73
	2.33	1.14	2.66	1.58	3.43	1.90	3.77	2.52	5.32	2.74
	2.32	1.13	2.63	1.52	3.42	1.91	3.80	2.51	5.33	2.74

Table 3. Execution Time of C++ and Java on Core 2 Duo Processor Architecture

Lines of Code	**10**		**10^2**		**10^3**		**10^4**		**10^5**	
P/L	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**
Execution Time in Seconds	2.14	.926	2.47	1.36	3.29	1.703	3.39	2.375	4.10	2.54
	2.34	.926	2.57	1.37	3.09	1.702	3.49	2.375	4.15	2.54
	2.04	.926	2.40	1.37	3.39	1.700	3.33	2.375	4.14	2.54
	2.22	.925	2.40	1.36	3.29	1.702	3.42	2.375	4.13	2.53
	2.11	.927	2.41	1.36	3.17	1.704	3.36	2.376	4.11	2.55

Table 4: Comparison of Execution Time of C++ & Java on Dual Core and Core 2 Duo Processors

Lines of Code	**10**		**10^2**		**10^3**		**10^4**		**10^5**	
P/L	C++	Java	C++	Java	C++	Java	C++	Java	C++	Java
Dual Core Processor	2.34	1.15	2.64	1.56	3.49	1.93	3.79	2.51	5.31	2.74
Core 2 Duo Processor	2.17	.926	2.45	1.36	3.24	1.703	3.39	2.375	4.12	2.54

5. Concluding Remarks

From the above work it is concluded that UML modeling is a powerful modeling language to represent software architecture and research work visually. Performance of Dual Core and Core 2 Duo processors architecture for two most popular object-oriented software languages by taking the variation in lines of code, is observed. It is concluded that Java programming language takes less execution time as compared to C++ for both the processors. It is also found that Core 2 Duo Processor is much faster than the Dual Core processor architecture. Therefore Core 2 Duo processor is very powerful and recommended for long computations related to the Java object-oriented Programming language.

6. Acknowledgements

Authors are very thankful to Prof. B. Hanumaiah, Vice Chancellor, Babasaheb Bhimrao Ambedkar University (A Central University), Vidya Vihar, Rai Bareli Road, Lucknow, India, for providing the excellent computation facilities in the University Campus. Thanks also due to the University Grant Commission, India for providing financial assistance to the University for this Research Work.

References

[1]. J. Conallen ,"Modeling Web Application Architectures with UML", Communication of the ACM 42(10), 63-70, (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]. OMG, "Unified Modeling Language Specification", Available Online Via www.omg.org , (2001).

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

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

[6]. K. Hwang. "Advanced Computer Architecture", McGraw–Hill Series in Computer Engineering, Inc Publishing, (1993).

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

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

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

[10]. A. Helsinger, R. Lazarus, W. Wright & J. Zinnky, "Tools and Techniques for erformance Measurement of Large Distributed Multi Agent System", Proceedings of AAMAS 03 Conference, Australia. p. 843-850 (2003).

[11]. H. Allen, "Microarchitecture of Pentium IV processor", Desktop Platform Group, Intel Corporation

[12]. 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 SAI 2001 World Congress Detroit, Michigan, USA, (2001).

[13]. S. Pllana, and T. Fahringer, "UML based Modeling of Performance Oriented Applications", Winter Simulation 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 Modeling for Instruction Pipeline Design", International Conference on Software Engineering 2008,organized by **www.waset.org** from Aug. 29-31, 1 Sept., pp. 293-296, SINGAPORE (2008).