Multithreading Architecture Based Restaurant Management System

Mayawati Namdeo Tayade,Prof. P.A.Khune

Dept of Computer Engineering,
PVPIT, Savitribai Phule Pune University, Pune,
Maharashtra, India.

1) INTRODUCTION: Multitasking always been a crucial part of many industries. Specially, in restaurant industries multitasking operations have been performed on a huge scale. When it comes to integrate such multitasking operations with advanced software system it becomes more intricate while handling a wide range of complex processes in just a single program.

Co-ordinate multiple pieces of information from different sources of data is the core capability of multitasking based system. These data sources can be manual or can be automatic responses from electronic systems. So, the actual challenge while working with multithreaded application is dynamic execution of processes without affecting resources of computer and without having a latency issue. Even though these days computer processors are enormous powerful than the previous generations, still there some glitches to reduce the latency factor in process execution. Perhaps, there have been many strategies to slice down latency factor and smooth dynamic execution but it takes a huge effort to accomplish the projected expectations.

With this multithreading application we can resolve this latency factor using delegates.

When multiple thread invokes delegates comes into coordinate these multiple threads and allows these threads to perform the runtime operation smoothly and without latency. This application is very useful in restaurant industry now-a-days.

2) CONCEPTUAL BASIS OF MULTITHREADING ARCHITECTURE: The actual concept multithreading has been used since the evolution of computer operating systems. This is exclusive technique which enables the end user to perform multiple parallel functions in just one process. Multithreading initiates instance of thread in a dynamic memory stack and this stack is flexible enough to be controlled by a thread handler which enables any thread to start, pause, resume or end the process. This complex mechanism allots a single instance to each of the core of processor which shares the process. So, this sharing of instance among the cores multiplies the execution speed of a task.

3) FACTORS WITHIN MULTITHREADING ARCHITECTURE: To review the factors within this interesting architecture of multithreading, one has to go through whole phases of setting up an environment for multithreading application. In the earlier description, restaurant industry has key role managing and fast processing the order tickets received from different personnel on the different levels. However, the adequate conveyance of multiple instructions from one node to the destined node simultaneously is one the most important factor. However, to enable the smooth and accurate communication between multiple node between one process must be accomplished with minimum time. So, the latency does matter whenever a new instance is being created as an order to perform multiple processes with an array of complex information. On the other hand, co-ordination between multiple threads is also important in order to protect the ongoing process from critical failure while performing cross threading operations between multiple threads. Cross thread is a typical complex situation when one instance is trying to access the resources of another ongoing thread which may belong to a separate process or a control. Above all, a greater sense of handling a complex multithreading is when all the primitive hardware resources like memory hops and processor engagement gets perfectly optimized to diminish the possibility of runtime exception.

4) LIMITATIONS: While working with multithreading architecture, even though we have a plethora of options to perform it well, it still has some glitches and catches with makes it bounded in a frame of setbacks. One has to be aware about how hardware works and has to be very careful with the hardware resources while handling such an intricate method. There is some technique available which can synchronize the threads in order keep them in line with other. This synchronization is completely depending on the available on the capability of actual availability of runtime resources like memory and processor usage. In many cases, programmer creates a separate access to cache memory of processor in order expand the quote data caching for faster execution of threads, but these are the most sensitive and reserved part of the processor which needs tremendous practice to handle the its core functional behavior. The most dependent factor of multithreading architecture is hardware specification is determined not by the capacity of memory of processor but by the size of organization where a multithreaded based application is being deployed; because larger the organization gets, more the number of threads it needs and henceforth it needs more powerful hardware resources.

The ultimate challenge is to trim down the memory allocation by maximum amount sharing the variables for common storage. This strategy works when one has to optimize the memory using enormous amount of code with precise execution time. So, this idea of handling the maximum amount of processes along with a complex conditions sounds promising and impactful in every aspect of perfect process management with least latency and high accuracy.

However, as a matter of the fact, coming across a tricky situation where one has to administrate a smooth functioning of prolific part of the restaurant, the developer has to go through a thorough examination of the live scenario instead of relying on the imaginary visualization of performing multitasking operation using advanced software system. There have been many efforts taken in order to enhance the functional parameters of multithreading architectures which can extend the limits of fast execution of parallel threads
in a single pipe line of process.

5) CONCLUSION: This System enable the smooth and accurate communication between multiple node between one process must be accomplished with minimum time. With this multithreading application we can resolve this latency factor using delegates. When multiple thread invokes delegates comes into coordinate these multiple threads and allows these threads to perform the runtime operation smoothly and without latency. This application is very useful in restaurant industry now-a-days.

6) REFERENCES:

[1] Noor Azah Samsudin, Shamsul Kamal Ahmad Khalid, Mohd Fikry Akmal Mohd Kohar, Zulkifli Senin, Mohd Nor Ihkasan; "A Customizable Wireless Food Ordering System With Real-Time Customer Feedback."; 2011 IEEE Symposium on Wireless Technology and Applications (ISWTA), September 25-28, 2011, Langkawi, Malaysia

[2] Sakari Pieskä, Markus Liuska, Juhana Jauhiainen, and Antti Auno Of Centria University Of Applied Sciences Ylivieska; "Intelligent Restaurant System Smart Menu That Digital Technology"; coginfocom 2013 4th IEEE International Conference on Cognitive Infocommunications December 2–5, 2013 , Budapest, Hungary.

[3]Jiawei Han and Micheline Kamber, "Data Mining: Concept And Techniques", University of Illinois.

[4] M. Z. H. Noor, A. A. A. Rahman, M. F. Saaid, M. S. A. M. Ali, M. Zolkapli ―The Development of Self-service Restaurant Ordering System(SROS)‖ 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012)

[5] C. Chen, and Y, Guan, "Experience design of the theme restaurant, make the dining be a memorable experience," in Proceedings of 9th International Conference on Computer-Aided Industrial Design and Conceptual Design, Shanghai, China, pp. 982–985, November 2008.