ABSTRACT

The thesis presents novel mechanisms to accomplish reliable and optimized process migration and load balancing in distributed systems. The process migration mechanism is used to gain the benefits, such as – dynamic usage of resources, load-balancing, fault-tolerance, reliability, higher throughput, etc. in the network of workstations. In this thesis, we have presented distinctive algorithms and their implementations to fulfill optimized process migration in order to handle various challenging issues such as – resuming the migrated process with a certain process-id value, checkpointing the memory regions occupied by a process, and resuming the migrated process in certain system call which the process was executing earlier before process migration took place, etc. Moreover, in this thesis we have presented algorithms for implementing proficient load balancing technique. The suggested solutions are aimed at providing optimized process migration and efficient load balancing facilities in a distributed system.

The thesis is organized into seven chapters.

Chapter 1 of the thesis attempts to justify the significance of the process migration and load balancing terminologies. The chapter talks about the principal types of process migration mechanisms including – preemptive and non-preemptive, homogeneous and heterogeneous, user-level and kernel-level, weak and strong process migration, etc. The chapter highlights the preemptive and homogeneous nature of the process migration algorithms proposed in this thesis. Apart from explanation of the research problem, the chapter introduces the ‘OptiMigrator’ – the solution software which has been described in this thesis.
Chapter 2 of the thesis presents a study about the types of process migration techniques – the user-level and the kernel-level process migration. A literature survey about the existing kernel-level techniques, such as Amoeba, V Distributed System, MOSIX, BLCR, Sprite, EPCKPT, CRAK ZAP, AutoPod and CHPOX along with their capabilities and weaknesses is presented in this chapter. The chapter also presents a study about the available user-level process migration techniques, such as Condor, CKPT, Esky,UTOPIA, MPVM, REXEC, rsh, Libckpt, Libtckpt, CoCheck and Xen along with their capabilities and limitations thereafter. In addition, the chapter points out significant remarks on the literature survey by insisting on the necessity of more research contributions especially for (i) checkpointing of the memory regions, (ii) migration of the system calls, and (iii) migration of the process credentials. Finally, the chapter concludes with the discussion on the process migration mechanism.

Chapter 3 of the thesis highlights the prime challenging issues arising in the course of migration of a process that is in the state of execution of some system call. The chapter presents significant guidelines to resume the execution of a system call after process migration on the remote node. Moreover, the chapter highlights the design characteristics of the software solution presented in this thesis such as – transparency, consistency, optimality, flexibility, efficiency and responsiveness.

Chapter 4 of the thesis deals with critical technical concerns experienced during the execution of the mechanism of checkpointing of virtual memory areas which are assigned to a particular process by an operating system. Apart from discussion on the memory descriptor and the virtual memory area descriptor, the chapter presents a solution to checkpoint the process address space. The designs and implementations of various unique algorithms that are applicable to checkpoint the memory regions, such as data segment, stack
segment, heap segment and other segments, have been presented in this chapter at the end.

**Chapter 5** of the thesis discusses the major issues emerging out of the migration of a process which is assigned a certain process-id value such as – after migrating to the destination node, the migrated process does not resume with the same process-id value which it possessed on the source node before migration took place. The chapter presents significant directives in the form of a novel algorithm and its implementation in order to resume the migrated process on the destination node with the original process-id value which it was assigned earlier on the source workstation.

**Chapter 6** of the thesis deals with the issues concerned with the technique of load balancing with the help of process migration mechanisms on distributed systems. Certain issues appearing in the area of load balancing such as – static load balancing and dynamic load balancing have been discussed in this chapter. The chapter suggests the dynamic load balancing solution in the form of both the server-managed and the client-initiated algorithms and presents implementation of these algorithms. The suggested algorithms provide novel mechanisms through which the lightly loaded workstations do not suffer from sudden overburdening caused by the server's consecutive multiple load sharing advises. Some of the important implementation functions also have been explained in this chapter.

**Chapter 7** of the thesis presents the detailed summary of the work presented in the prior chapters of this thesis. Moreover, the chapter highlights the performance results achieved by using the suggested mechanisms in this thesis. The chapter also depicts the behavior of the suggested process migration and load balancing techniques in prior chapters. The chapter concludes by mentioning the scope of further research in the area of process migration and load balancing.