Chapter 6
Conclusion & Future Scope

6.1 Conclusion

1. Dynamic memory allocator has been proposed for symmetric multiprocessing system which provides consistent and optimum execution time, less memory fragmentation as well as satisfies maximum number of memory request as compared to other existing dynamic memory allocators.
2. As per the need of high-performance computing, a dynamic memory allocator for NUMA architecture based real-time operating system has been proposed which also provides consistent and optimum execution time, less memory fragmentation as well as satisfies maximum number of memory request.
3. MemSimRT has been designed to simulate various memory allocators for both SMP as well as NUMA architecture based RTOS.

6.2 Future Work

The analysis presented here points to several areas for future work.

1. The implementation presented here is a user-level memory allocation allocator; so one can reform the kernel of Linux to minimize the distance when remote access is required.
2. Synchronization for sharing the resource is a significant thought in the proposal of a memory allocation algorithm which supports concurrency and scalability. But, for real-time systems, it is essential to consider schedulability with the synchronization. So one can design a scheduler which can integrate with DmRT.
3. The garbage collector is still a challenge in the real-time system because of its random delays. Still, there are real-time garbage collectors available which provide satisfactory performance on soft real-time systems. The incorporation of these garbage collector with DmRT can be another area of future work.