# DEVS형식론을 이용한 다중프로세서 운영체제의 모델링 및 성능평가

홍 준성, 김 탁곤, 박 규호

한국과학기술원 전기및 전자공학과

305-701 대전직할시 유성구 구성동 373-1 한국과학기술원 전기및 전자공학과

042-869-5425

In this example, a message passing based multicomputer system with general interconnection network is considered. After multicomputer systems are developed with worm-hole routing network, topologies of interconnection network are not major consideration for process management and resource sharing. There is an independent operating system kernel on each node. It communicates with other kernels using message passing mechanism. Based on this architecture, the problem is how much does performance degradation will occur in the case of processor sharing on multicomputer systems.

Processor sharing between application programs is very important decision on system performance. In almost cases, application programs running on massively parallel computer systems are not so much user-interactive. Thus, the main peformance index is system throughput. Each application program has various communication patterns, and the sharing of processors causes serious performance degradation in the worst case such that one processor is shared by two processes and another processes are waiting the messages from those processes. As a result, considering this problem is important since it gives the reason whether the system allows processor sharing or not.

Input data has many parameters in this simulation. It contains the number of threads per task, communication patterns between threads, data generation interval, mapping threads to processors, and so forth. There are some benefits and also defects in random input data. Many parallel application programs has its specific communication patterns, and there are computation and communication phases. Therefore, this phase information cannot be obtained from random input data. If we get trace data from some real applications, we can simulate the problem more realistic. On the other hand, simulation results will be wasteful unless sufficient trace data with various communication patterns is gathered. In this project, random input data are used for simulation. Only controllable data are the number of threads of each task and mapping strategy. First, each task runs independently. After that, each task shares one and more processors with other tasks. As more processors are shared, there will be performance degradation. From this degradation rate, we can know the overhead of processor sharing. Process scheduling policy can affects the results of simulation. For process scheduling, priority queue and FIFO queue are implemented to support round-robin scheduling and priority scheduling.