Lecture 9 Summary

Two papers were included in this lecture. The first one is: The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q by Fabrizio Petrini, Darren J. Kerbyson and Scott Pakin. In this paper, they met a problem that there was a difference between the measured performance in September 21, 2002 and November 25, 2002 and the expected performance from a performance model. To identify the source of the problem, they performed a measurement by varying the number of processors on one node, from 1 processor per node to 4 processors per node. And the results revealed that the problem only occurred on the scenario that a node utilized all processors. They also observed that this performance deficit appears to originate from the collective operations like **allreduce**. To identify factors that affected the performance, they first optimized the **allreduce**. They observed that then using 4 processors, there was a huge increase of latency. By optimization of improving the latency by a factor of 7, 78% performance gain in SAGE. Second, they had an intuition that periodic system activities were interfering with application execution. This could happen when there is no spare processor and a processor is temporarily taken from the application. And they used microbenchmark on all 4096 processors of QB and confirmed that there was a system noise. Although that it is not possible to remove all the noise in a system, they still optimized the system by removing unnecessary daemons and decreasing the frequency of RMS monitoring. And in May 1, 2003 measurement, there was a huge improvement on this system.

The second one is: Benchmarking the effects of operating system interference on extreme-scale parallel machines by Pete Beckman, Kamil Iskra, Kazutomo Yoshii, Susan Coghlan and Aroon Nataraj. They did a microbenchmark on the effects of operating system interface on supercomputers with injection of artificial noise. The conclusion is that the performance is related to the largest interruption to the application, even if the interruption is extremely small. And synchronizing can significantly reduce the negative effect caused by the noise.