CSCI 6360: Parallel Computing Lecture Summary - 9

Anirban Das (dasa2@rpi.edu)

February 13, 2018

Summary on the papers: (1)*The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q* by Petrini et.al. and (2) *Benchmarking the effects of operating system interference on extreme-scale parallel machines* by Beckman et.al.

In the first paper, the authors detailed how they were able to achieve maximum performance out of then second fastest supercomputer of the world, ASIC-Q. They used SAGE, a sophisticated hydrodynamic simulation code, on ASIC-Q, to test if it is performing sub-optimally and subsequently find the reason and implement a solution. Initially they found that the performance of SAGE on ASIC-Q is significantly worse than the predicted performance. It was found that fewer processes per node boosted system performance at higher number of nodes. The significant variability (noise) in performance, acc. to the authors originated from sources like kernel activities, RMS daemons etc. while collective operations were being performed. The system noise that was generated due to the presence of a straggler node in a collective operation on large number of processors was the main problem according to the micro benchmark they performed. A single slow process on a barrier synchronization application was able to delay the whole application by even 100 times! After some optimization for e.g. removing unnecessary daemons etc. they were able to push SAGE's performance by 2x. One other paradigm the paper suggests is that of harmonic noise , i.e. when some sources of noise tend to resonate or match with application level noise and therefore have more impact. The authors found that high frequency fine grained noise tends to affect only high frequency application, and the low frequency affects coarse grained applications. The authors concluded that their paradigm helps in profiling and optimizing the system artifacts instead of the application itself and thereby gains on the traditional methodologies.

In the second paper, authors Beckman et.al. also focus on operating system noise and how to reduce its negative influence by synchronising. Noise is caused when the progress of the application is interrupted by background activities. They reiterate the fact that the performance/ over all speed in a complex all to all communication is determined by the slowest performing node. They performed several benchmarks to detect detours such as delay in measurement of the CPU time accurately by applications, then detecting the inherent noise. They also performed a noise injection method on applications needing synchronous behavious such as barrier, all to all, allreduce, etc. collective operations. It turns out that the synchronous noise injections had minimal effects but unsynchronous or random noise created more delays in execution time i.e. coarse grained noise effects system's performance more than fine grained (synchronous) noise. The authors concludes that the detours need to be significantly large in order to be detrimental to the performance of large scale architecture machines, and OS noise as long as it is synchronised and fine grained does not cause major impact.