

# 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015)



Contribution ID: 3

Type: **oral presentation**

## The Effect of NUMA Tunings on CPU Performance

*Tuesday, April 14, 2015 6:15 PM (15 minutes)*

Non-uniform memory access (NUMA) is a memory architecture for symmetric multiprocessing (SMP) systems where each processor is directly connected to separate memory. Indirect access to other CPU's (remote) RAM is still possible, but such requests are slower as they must also pass through that memory's controlling CPU. In concert with a NUMA-aware operating system, the NUMA hardware architecture can help eliminate the memory performance reductions generally seen in SMP systems when multiple processors simultaneously attempt to access memory.

The x86 CPU architecture has supported NUMA for a number of years. Modern operating systems such as Linux support NUMA-aware scheduling, where the OS attempts to schedule a process to the CPU directly attached to the majority of its RAM. In Linux, it is possible to further manually tune the NUMA subsystem using the "numactl" utility. With the release of Red Hat Enterprise Linux (RHEL) 6.3, the "numad" daemon became available in this distribution. This daemon monitors a system's NUMA topology and utilization, and automatically makes adjustments to optimize locality.

As the number of cores in x86 servers continues to grow, efficient NUMA mappings of processes to CPUs/memory will become increasingly important. This presentation gives a brief overview of NUMA, and discusses the effects of manual tunings and numad on the performance of the HEPSPC06 benchmark.

**Primary author:** HOLLOWELL, Christopher (Brookhaven National Laboratory)

**Co-authors:** Mr ZAYTSEV, Alexandr (Brookhaven National Laboratory (US)); CARAMARCU, Costin (Brookhaven National Laboratory (US)); Dr WONG, Tony (Brookhaven National Laboratory); STRECKER-KELLOGG, William (Brookhaven National Lab)

**Presenter:** HOLLOWELL, Christopher (Brookhaven National Laboratory)

**Session Classification:** Track 8 Session

**Track Classification:** Track8: Performance increase and optimization exploiting hardware features