An Analysis of Node Asymmetries on seaborg.nersc.gov

David Skinner and Nicholas Cardo
NERSC HPCF

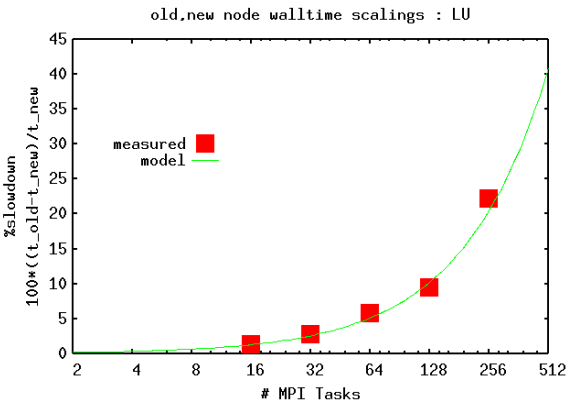
Abstract: A short description of work completed at NERSC over the past 6 months to identify and remedy asymmetries in the in the batch compute resources provided by NERSC's IBM SP seaborg.nersc.gov.

Background: NERSC's IBM SP consists of 375 Mhz NighthawkII power 3 nodes. The number of nodes has grown as the system increased as part of the NERSC3E procurement. During the initial rollout of the new NERSC3E nodes they were separated into a queue used for testing and an early user program. During this acceptance period there were reports of the new nodes performing slightly better than the old hardware. All reports involved parallel MPI codes. No serial performance differences were detected. During its integration the system was in a state of flux. At times the two sets of nodes showed very similar parallel performance. After the new nodes were integrated with the old running comparisons of parallel jobs between old and new nodes became extremely difficult. This has to do with how IBM's LoadLeveler scheduling system deals with requiring a certain set of nodes for job assignment.

Old/new node performance differences were tracked through IBM support, development, and product engineering groups to identify possible causes of the observed differences. The cause of the problem was quite puzzling. Periodic observations and testing provided inconsistent results and at times no asymmetry could be measured. The decision to set aside a group of old and new nodes for in depth testing on Oct 1, 2003 led to conclusively identifying and addressing the performance differences.

Problem Identification:

Jobs run slower, in proportion to their concurrency, on old nodes. The degree of the difference depends on the concurrency and the amount of synchronization in the MPI calls used in the code. A test case employed in the resolution of this issue is the NAS parallel benchmark LU because it was turned out to be a fast, reliable probe that coincided with the performance difference of full scale applications.



Since serial codes show no measurable difference the parts of the parallel codes that involve synchronization are implicated. Interruptions at the OS level or at the switch adapter level can have a minimal impact on serial processes, but compound when many concurrent processes are interrupted. If a linear

model of frequent short interruptions on each node is extrapolated, the old nodes have half the performance of the new nodes (for LU decomposition) at a concurrency of 1250 tasks! Everything observed from the testing shows that synchronization of parallel jobs is being impacted by delays proportional to concurrency. At this point it was not known if these interruptions were from hardware or software.

Pursuing the cause of this issue was done through three paths:

- **Application testing:**

By profiling the runs on separate collections of old and new nodes – including segmented switch subtrees - one can determine which sections of the code account for the overall timing differences. This was done by wrapping all MPI calls with timers. Knowing the time spent in each MPI routine shows that the variation was related to a small number of MPI function, e.g., for the NPB LU code nearly all of the asymmetry in wall clock time is incurred while in the MPI routine MPI_Wait. Since the MPI software on all nodes has been shown to be the same, the MPI_Wait differences must be the result of something outside of the scope of the application and/or MPI libraries. This testing only indicated that MPI_Wait takes longer to synchronize codes on old nodes than it does on new nodes.

- **Hardware testing:**

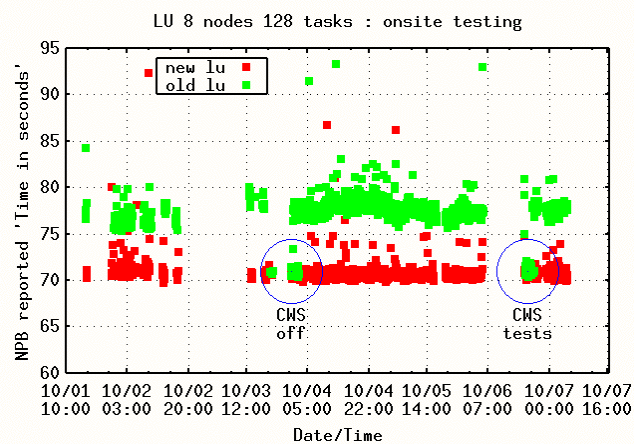
IBM pursued the issue from the point of view of hardware differences. A plan was developed and carried out to swap hardware components between old and new nodes in order to identify hardware that might be responsible for the observed asymmetry. IBM sent hardware engineers out to complete the hardware testing. NERSC staff helped in the running and evaluation of the NPB LU benchmark. CPU's, memory books, system planars, and switch adapters were all swapped between the two sets of nodes without observing a change in the asymmetric timings.

- **OS testing:**

NERSC systems staff double checked that the OS images used to install the batch nodes were uniform. Using identical images for OS install is part of the standard methods for system administration of seaborg. Checksums of system libraries were compared. No asymmetries were found.

Resolution:

A critical insight occurred when the control workstation (CWS) happened to be inoperable during a period when old/new node testing was being done. While the CWS was off-line on Oct 4 the timings of LU NPB on old nodes improved to the better timings consistently given by the new nodes. It was this unscheduled observation,

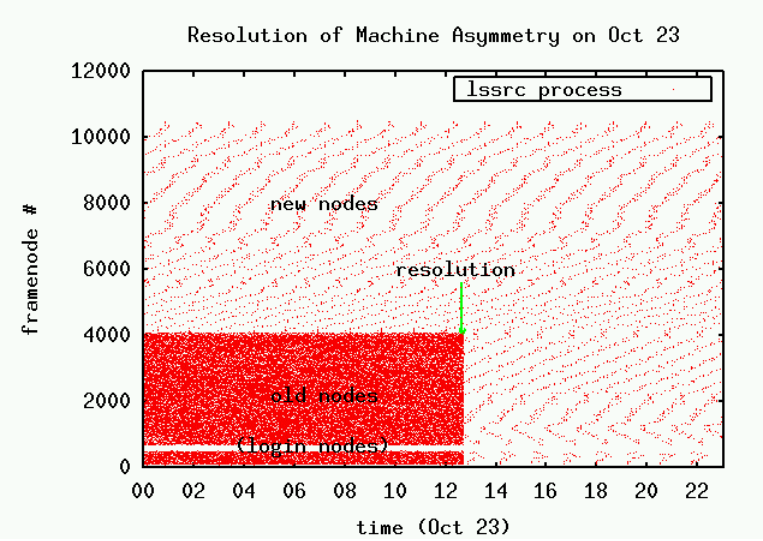


which pointed to the CWS as a source for the asymmetry that lead to the resolution of the problem.

A series of further CWS related tests done by IBM onsite staff and NERSC staff showed that two specific subsystems run from the CWS, HATS and HAGS, led to the discrepancy in the timings. With these services turned off, old nodes performed as well as new nodes. IBM took this information back to their support staff and developers who began to look at how HATS/.HAGS design and implementation might impact the performance of parallel applications.

NERSC staff examined the problem from a different more bottom up perspective. By looking at UNIX process accounting logs it was possible determine how the CWS impacts processes run on compute nodes. This showed that with the CWS on certain nodes ran lssrc, spget, odmget, and other system administrative commands up to 27 times more often than other nodes.

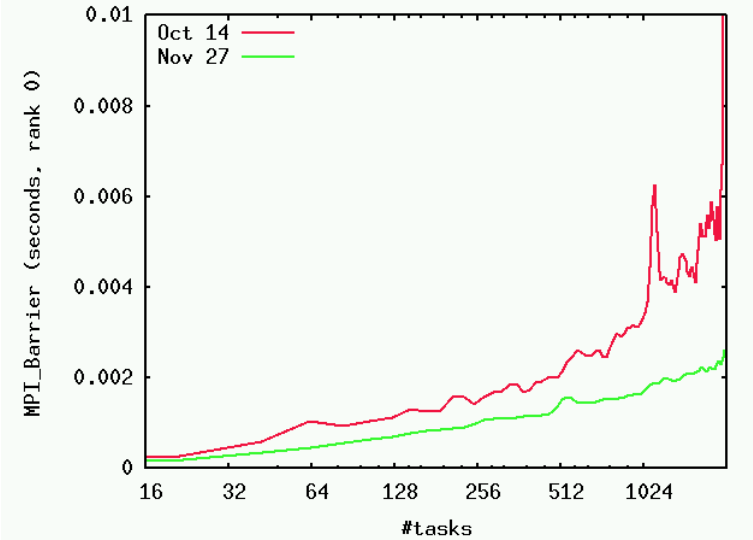
Process Name	#calls old	#calls new	Asymmetry ratio
#spget_sy	191817	6864	27.945367
#fcistm	192121	7188	26.728019
#lssrc	194608	7780	25.013882
#basename	385701	15550	24.803923
#odmget	193918	8481	22.864992
#ksh	390625	20129	19.406081
#rm	197514	12449	15.865853
#sed	397999	29482	13.499729
ksh	206206	23159	8.903925



Knowing the specific processes and their frequencies provides a fingerprint of the subsystem causing the interruptions. Using grep to find these command names occurring in the system script /usr/lpp/csd/bin/ha.vsd which is invoked as part of the First Failure Data Collection subsystem. This led IBM down the path of investigating the problem management subsystem (pman). It was found that the pman commands indicated that 4 definitions were deactivated in the system management GUI, however, lssrc showed them still running. The definitions required explicit deletion from the SDR to remove them, rather than deactivation as is documented. PMR #38446 has been opened to correct this defect in the problem management subsystem, while the workaround is to actually delete the files.

Thanks to everyone who helped in the resolution of this issue.

Some before and after MPI testing:



The above graph is smoothed to make the trend clearer. The lower one is the raw timings. Resolving this issue lead to a definite improvement for synchronizing MPI codes at high concurrency. In normal operation, jobs use a combination of old and new nodes. Thus, the end result is that all codes see a benefit of faster and more consistent run times, particularly those codes at higher concurrency.