High Performance Computing in Undergraduate Operating Systems

David R. Monismith Jr.
Northwest Missouri State University
800 University Dr.
2210 Colden Hall
(660) 562-1802, +1
monismi@nwmissouri.edu

ABSTRACT
This poster provides an experience report on an undergraduate Operating Systems (OS) course emphasizing High Performance Computing (HPC).

Categories and Subject Descriptors
K.3.2 [**Computer and Information Science Education**]: Computer Science Education and Curriculum.

General Terms
Algorithms, Measurement, Performance, Design, Experimentation, Languages, Theory.

Keywords
High Performance Computing, Operating Systems, Education.

1. INTRODUCTION
Fall 2012, the author taught undergraduate Operating Systems (OS). High Performance Computing (HPC) was included as a major course focus. Use of HPC was inspired by novel developments in OS courses at other universities including pedagogical OS's, an Android OS course, and a graduate level OS course that used the Message Passing Interface (MPI) [1-2]. Students were introduced to UNIX/Linux system commands, processes, threads, scheduling, memory management, file systems, synchronization, and deadlock via HPC resources [3].

2. INSTRUCTION AND ASSIGNMENTS
Adding HPC to an OS course provides exciting opportunities to both educator and student. HPC resources provide for students to learn about UNIX/Linux, batch scripts, batch operating systems, C, and MPI. Teaching these tools gives students an opportunity to learn multithreaded and multiprocess programming. MPI initialization and finalization provides a starting point for discussion of multiprogramming. Since rank and size functions provide both process count and identifiers, these reinforce understanding of the process control block (PCB) and process table and contrast with UNIX/Linux commands. Send and receive functions allow for discussion of interprocess communication, process states, and allow for introduction of waiting and deadlock. Coverage of deadlock, file systems, I/O, and memory management is straightforward. MPI file I/O tools allow for introduction to networked file systems. Ordered sending and receiving allows for discussion of deadlock [2]. Additionally, assignments can be tuned for OS concepts. Students completed a matrix multiplication project with in Java with threads and C with MPI, including both scaling and timing. They also completed a three-tiered producer consumer assignment for sorting, which used a top-level producer process that created bins of random integers for multiple consumer processes. Those same consumers sorted the bins and provided them to a bottom-level consumer that combined the bins into an ordered list. Other interesting assignments included multithreaded Round Robin and First-Come First Served schedulers and a performance comparison of Lustre and NTFS. MPI allow students to learn OS fundamentals and functions contrast with OS tools [2-5]. When combined with HPC resources, students gain scaling and performance experience.

XSEDE'13, July 22–25, 2013, San Diego, CA, USA.

3. CONCLUSION
All students involved were able to access XSEDE resources and learn about a language and toolset outside their comfort zone as all had Java experience but little C or command line experience. Retrospectively, students could have spent much more time using HPC resources. In future OS courses, the author plans to include additional tutorial labs and assignments with a higher burn rate. Additionally, surveys will be conducted for further research.

4. ACKNOWLEDGMENTS
This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation (NSF) grant number OCI-1053575.

5. REFERENCES
[1] Andrus, J. and Nieh, J. 2012, Teaching Operating Systems Using Android. *Proceedings of the SIGCSE'12 Conference* (Raleigh, NC, Feb. 29 – Mar. 3, 2012).

[2] Scherger, M. 2011. A Course in Advanced Operating Systems Using MPI for Projects. *Journal of Computing Sciences in Colleges*, 26, 4 (2011), 57-63.

[3] Garrido, J. and Schlesinger, R. 2008. *Principles of Modern Operating Systems*. Jones and Bartlett Publishers Inc., Sudbury, MA.

[4] Gropp, W., Lusk, E., Ross, R., and Thakur, R. 2012. Advanced MPI Tutorial. *Proceedings of the International Conference on High Performance Computing* (Salt Lake City, UT, Nov. 10 – 16, 2012).

[5] Grama, A., Gupta, A., Karypis, G., and Kumar, V. 2003. *Introduction to Parallel Computing*, 2nd Ed. Pearson Education Ltd., Harlow, Essex, England.