Critique #2: Stephen Orchowski
Paper: “Facilitating Efficient Synchronization of Asymmetric Threads on Hyper-Threaded Processors”
Presenter: James Coleman, Wednesday, 11/12/2008

Summary of Paper:

This paper presents a software implementation for taking advantage of two specialized processor instructions, Monitor and Mwait, to enhance the performance of synchronization constructs between multiple threads. The authors of the study conclude that their implementation brings about significant performance improvements by both reducing the response time of synchronization constructs and the processor utilization for such a construct. Thus, the scheme is able to obtain the speedups without the overhead normally attributable to responsive synchronization mechanisms.

Summary of Strengths:

1. The scheme seems to get the best of both worlds. The use of the processor instructions allows the scheme to be both responsive and less processor intensive (i.e. with spin locks).
2. There are obvious speedup enhancements for improving the thread synchronization performance and throughput. This is going to be much more important especially in multi-core chip processor architectures.

Summary of Weaknesses:

1. The implementation of the performance enhancement relies on two special instructions on an Intel processor. This is potential drawback to the design since these instructions might not be available on other processors. Thus the effectiveness of this mechanism might restrict the variety and number of applications/OSes that can take advantage of it.
2. The implementation of the scheme requires modifications to the Linux Kernel and the design and implementation of various kernel drivers. This isn’t a problem per se. But it does reduce the attractiveness of actually implementing this type of enhancement. Such an implementation means that an OS or application is more tightly coupled to the underlying processor architecture.
3. The experiment seemed to focus on a single benchmark case of a powerful busy thread coinciding execution with a more idle thread. The study could have done more to benchmark various other thread cases and different combinations of thread characteristics and numbers.
4. The study does not offer suggestions on how such an implementation might factor in the design or use of virtualized systems. The instructions require special considerations in the OS running on the architecture and this might cause issues or problems with other guest OS instances running on top of this scheme.