- **John Kelm**
  - **Title:** Hybrid Coherence for Scalable Heterogeneous Computing
  - **Abstract:** Two broad classes of memory models are available today: models with hardware cache coherence, used in conventional chip multiprocessors, and models that rely upon software to manage coherence, found in compute accelerators. In some systems, both types of models are supported using disjoint address spaces and/or physical memories. In this talk I will present the concept of a hybrid memory model that enables fine-grained temporal reassignment of data between hardware-managed and software-managed coherence domains, allowing a system to support both. I will present our proof-of-principle implementation called Cohesion. I will show how Cohesion can be used to dynamically adapt to the sharing needs of both applications and system software without requiring copy operations or multiple address spaces.

    Cohesion offers the benefits of reduced message traffic and on-die directory overhead when software-managed coherence can be used and the advantages of hardware coherence for cases in which software-managed coherence is impractical. Cohesion is implemented using the Rigel accelerator architecture. Rigel is a hierarchical, cached 1024-core processor with a single address space that supports both software-enforced coherence and directory-based hardware coherence protocols. Relative to an optimistic, hardware-coherent baseline, a realizable Cohesion design achieves competitive performance with a 2x reduction in message traffic, a 2.1x reduction in directory utilization, and greater robustness to on-die directory capacity.
  - **Bio:** John H. Kelm is a PhD student at the University of Illinois at Urbana-Champaign.His thesis is titled, "Hybrid Coherence for Scalable Multicore Architectures". John's PhD work investigates a 1024-core CMP architecture called Rigel and various methods for supporting cache coherence on that system. His research interests include scalable parallel architecture and microarchitecture focusing on memory models and cache management. John received an MS degree in computer engineering from the University of Illinois in 2006 and a BSE in computer science and engineering from the University of Connecticut in 2005. His masters thesis is titled, "Operating System Interfaces for Reconfigurable Accelerators". That work investigates adding support to a commodity operating system to efficiently and transparently support fine-grained FPGA-based accelerators used in heterogeneous CPU/accelerator systems. He has published papers in the areas of photonics, reconfigurable systems, performance analysis, parallel architectures, fault masking, memory model design, and cache coherence.