Active Testing in UPC

- **Active Testing**: Leverage program analysis to make testing quickly find real concurrency bugs
  - Phase 1: Use imprecise static or dynamic program analysis to find abstract states where a potential concurrency bug can happen (Race Detector)
  - Phase 2: Directed testing based on the abstract states obtained from phase 1 (Race Tester)
- **THRILLE** – THRead Interposition Library and Lightweight Extensions: Active testing framework for UPC

Hybrid Shared/Distributed Memory

- **BUPC** allows programs to use arbitrary combinations of Pthreads and Processes with shared memory
- **Mixing Pthreads and Processes** is required for:
  - Interoperability with MPI and non thread-safe libraries
  - Hybrid/hierarchical parallelism (for best performance)
- **New PSHM layer** – Process Shared Memory
  - Shared memory comms through POSIX, SYSV or mmap()
  - Shared memory “network” for Active Messages support
  - Hybrid processes/pthreads execution

Collective Communication

- **UPC** collectives implemented by GASNet collectives
- **GASNet collective features**:
  - **Auto-tuning**
    - Support online and offline search
    - A repository of popular algorithms with tunable parameters
    - Portable performance without user intervention
  - **Non-blocking multi-threaded collectives**
    - Overlap communication with computation/communication
    - Improve load balance with threaded communication
  - **Teams**
    - Enable efficient sub-group collective communication
    - Multi-threaded team collectives (research prototype)

For more information – [http://upc.lbl.gov/task.shtml](http://upc.lbl.gov/task.shtml)