

# What Is a Realistic Vision of the Future?

- The future is clearly multi-core
  - But what architecture, and how heterogeneous
- Near term commodity microprocessors: "SMPs on a chip"
  - With power limiting clock speeds to ~today's
  - And memory subsystem design focused on bandwidth
- So latency limited problems seeing no relief
- Fundamental underlying problem: thread state is *TOO BIG!!!*  
*And Growing*
  - Forces threads to be "persistent" & thus resource to be "named" & "managed"
  - It's TOO HARD to try to optimize BOTH data and execution site placement
- Languages like UPC begin to id thread state & their "locality"
  - But again threads are "persistent" and "named by" local memory
- Sequoia's "Gather-compute-Scatter" interesting
- HPCS languages beginning to break persistence
  - Express "short term" anonymous asynchronous threads "near" some "place"
  - Which today are implemented as "virtual threads" placed on work queue
- My takeaway: need architectures and languages that truly support and target large numbers of efficient, light weight, non-persistent, locality-aware threads

