• Have considered 4 types of parallel architectures
  • shared memory; message passing; dataflow; data parallel
  • they all started as very different machines

• Small scale machines: accepted model is cache-coherent shared-memory

• Large scale machines: significant convergence is happening
  • Distributed physical memory

• Scalable networks
  • HW vs SW combining for hot spots

• Single name space is important
  • single address space vs. single object space
• Locality is key to high performance
  • first reduce latency; then try and tolerate
  • many proposals: coherent caches; relaxed consistency models; prefetching; multiple context PEs; COMA
  • what to do in HW vs. what to do in SW
  • what combination of techniques make most sense (redundancy)

• Powerful processing nodes
  • all say should be powerful; how powerful is enough not resolved

• Synchronization
  • many proposals: F/E bits; fetch-&-op; comp-&-swap; active msges; ...
  • not clear what is needed in HW and what can be done in SW

=> At the broadest level, the big issues are:
  • what do the applications need
  • fundamental physics driving hardware design
  • what can be done in SW and what in HW