HTMT PROGRAM EXECUTION MODEL

PERCOLATION

HTMT-CLASS ARCHITECTURE

DATA VORTEX NETWORK

PERCOLATION MODEL

DATA VORTEX NETWORK

DRAM-PM

MAIN FEATURES OF PERCOLATION:
- Multitiered latency management
- Data aggregation in DPIM
- Threads / context pools
- No "hidden" memory coherence management

PERFORMANCE EVALUATION THROUGH EMULATION / SIMULATION OF IMPORTANT BENCHMARK APPLICATIONS

- Definition of THREADED-C 1.0 (CAPSL-TM19 3/1998) and THREADED-C 2.0 (CAPSL-TM39, 6/2000)
- The RTS will work with architecture and OS layers to provide the PXM interface
- A Draft Definition of HTMT THREADED-C 1.0 (3/2000)
- Design and Implementation of the HTMT Emulation Testbed
- Implementation and Performance Evaluation of Selected Benchmark Programs

INITIAL ROADMAP FOR A HPC SYSTEM SOFTWARE ARCHITECTURE

Applications

High-level language compiler

Intermediate Language Compiler and Tool Set

Runtime System (RTS)

Hardware Architectures

Nano-OS

Program Execution Model (PXM) Interface

Threaded-C Compiler - RTS interface

RTS-OS Interface

RTS-hardware architecture interface

Note:
- The threaded-C compiler has part of its functions embedded in RTS
- The RTS will work with architecture and OS layers to provide the PXM interface
- The performance models are defined across all layers

MULTI-THREADED EXECUTION MODEL

C1 (Asynchronous): a large pool of threads
C2 (Balanced): efficient Load Balancing
C3 (Cheap): fast spawning and termination

Thread within a frame
Parallel function invocation
A sync operation
Invoke a threaded func