Tera MTA

- Shared memory machine, throughput-oriented
  - flat, UMA access to shared memory, no caches
- Support large numbers of threads, at user level
  - cheap to spawn thread, and thread execution interleaved on a processor
  - streams are lightweight threads – instruction sequences started and stopped eventually, and used everywhere in the software stack design
- Memory has full/empty bits to make synchronization cheap and easy
- Somewhat simple compiler target
  - no need for “heroic” compiler technology – still need to find parallelism, either automatically or from compiler
  - not even branch prediction, or pipeline considerations, which were common even when the MTA was built

Cray X1

- Distributed memory machine, with hardware and software support for a form of DSM - each node is a multi-pipe vector machine
  - Cray T90 vector machine
  - Cray T3E a distributed memory machine with some DSM support, no vector units
- Node called an MSP
  - 4 SSPs per node each with a scalar and two vector units
  - 2MB Ecache per node, to improve memory bandwidth
  - max memory bandwidth only half of what the 4 SSPs can consume, so Ecache can sometimes make up the difference from reuse (even for vectors)
- Programming model is both parallelized across nodes, and vectorize within an MSP (across the SSPs too)

Notes

- Need at least 1 more volunteer for Thursday papers
  - and more days posted by tomorrow, so sign up
- Sample topics for group project posted later today, and proposal deadline set
- Talk to me about GRA positions

Sample topics for group project posted later today, and proposal deadline set

Talk to me about GRA positions
Cray X1 (cont.)

- Fast interconnect between nodes
  - for up to 128 nodes, a 4D hypercube
  - for more nodes (up to 1024), "enhanced" 3D torus

- Synchronization via atomic memory operations
  - for locks, barriers, etc.

- One MSP acts like an SMP, but each processor can also directly address memory on another MSP
  - remote accesses go over the network, and bypass local cache
  - use VM address translation to turn 64-bit virtual address into physical address with node number and local 36-bit physical address
  - remote access not good for single read/write (high latency, no caching), but good for block put/get operations (one-sided)

- Benchmarks show good performance from high bandwidth and relatively low latency to local and remote memory, fast interconnect, compared to other contemporary DSM machines
  - on micro-benchmarks and 2 real applications