The Stanford Dash Multiprocessor

Presented by Meng Zhang

Spring 2010 ECE 259

Motivation

Investigate highly parallel architecture

- Scalability

  scale up performance while maintaining cost-performance ratio
- Programmability

  provide a simple, flexible and general programming model

Dash Architecture

A distributed shared-memory system composed of clusters

- Within cluster
  - 4 processors
  - shared-memory UMA with a snooping bus
- Across clusters
  - a pair of wormhole routed meshes
  - one for requests, the other for replies

Cache Coherence Protocol

- Hierarchical memory system, 4 levels
  - Processor
  - Local cluster
  - Home cluster
  - Remote cluster

  } often the same

- Cache block in three states: M, S, I
  - Inside cluster, snooping protocol
  - Inter clusters, directory protocol

Memory consistency and optimization

- Relaxed consistency model
  - √ reasonable programming model
  - √ higher performance
- Access optimization
  - prefetch operations
  - update and deliver operations
- Support for synchronization
  - queue based locks
  - fetch-and-increment operations

Dash implementation

Each cluster is a 4 processors 4D/340 system

- Augmented with two boards
  - directory controller
  - network interface

- Modification
  - a bus retry signal
  - masking capability of the bus arbiter

Evaluation

- Simulation
- Actual hardware

Results

- Remote miss takes 3.5 times of the local miss
- Scale well for Mincut, Water, but not for MP3D (require frequent interprocessor communication)
- Limited to 64 processors (simulation) and 16 processors (actual hardware)

Thoughts and questions

- Novel architecture at that time
- First implementation of CC-NUMA

But…

- Insufficient evaluation (local miss, remote miss, different interconnect, comparison)
- Scale to thousands of cores?
- Complexity of implementing controllers?
- How difficult to rewrite applications?
- Power? Storage overhead?