DSM Research

- DASH Project, Stanford Uni, USA.
- MINT Project Rochester Uni, USA.
- DDM Swedish Institute of Computer Sci.
- Odin Project La Trobe Uni, Australia.

Lecture 10

DSM, classifying computer architectures using the Flynn Taxonomy

DSM Objectives

- Provide access to a shared memory programming environment.
- Reduce data access delays.
- Reduce system bottlenecks.

- Let us look in detail at one such system.

DSM

- Distributed Shared Memory
  - Provide a shared memory programming system.
  - Use distributed hardware.
  - Incorporate hardware caching and other efficiency features.
- How can this be achieved?

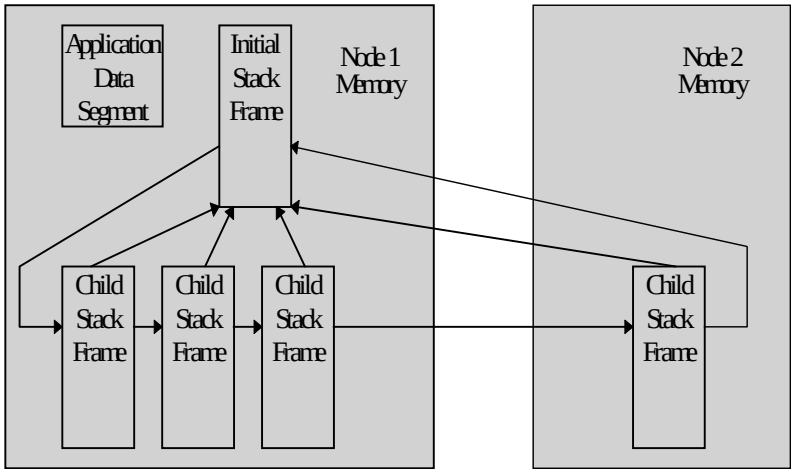
Kernel

- Copy of distributed Kernel on each node
- User level scheduling library (TIMS)
- Segmented stack (Cactus stack) permits rapid low cost thread migration.

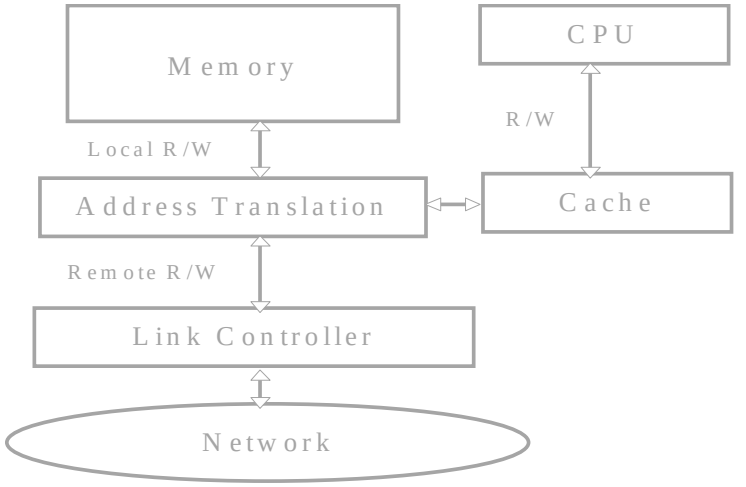
Odin Design Overview

- Interconnection topology
  - mesh
- Latency
  - segmented stack
  - thread migration
  - selective memory
  - forgetful cache consistency
- False sharing
  - cache line based consistency

Cactus Stack



Node Architecture



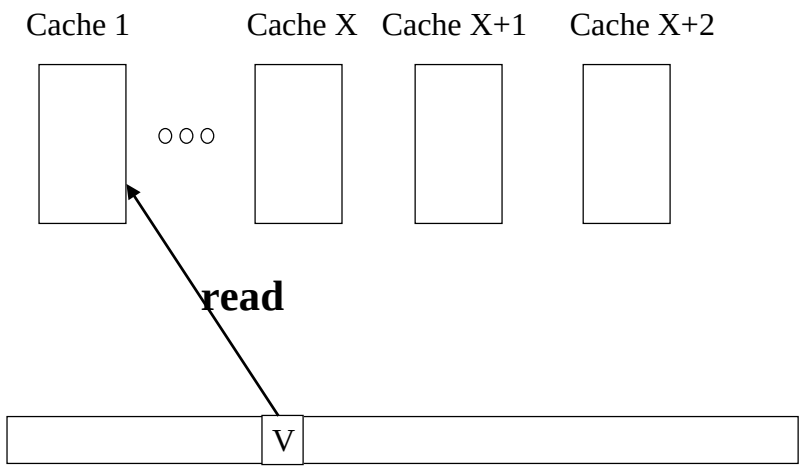
Cache Protocol

- Duplicate read copies permitted
- Consistency operations are intiated by synchronisation operations.
- Before synchronisation can complete all shared data must be flushed from the cache.
- Subsequent data fetches will retrieve consistent values.

Hardware Innovations

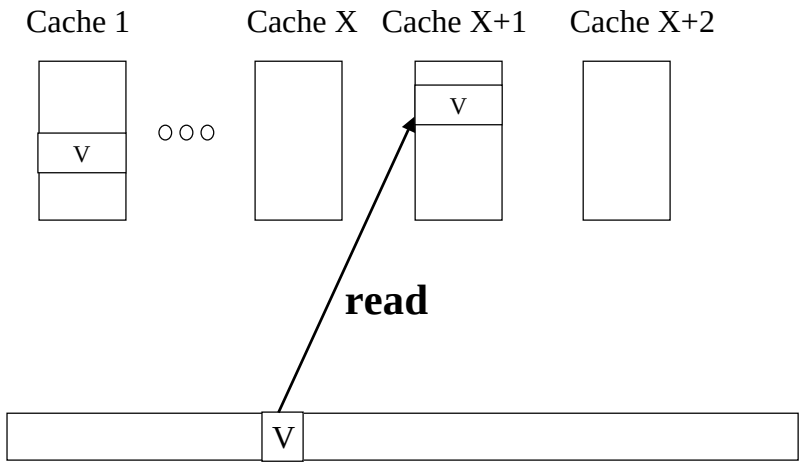
- Memory structure (selective memory)
- Cache coherence (forgtful caching)
- Link controller

Visualisation

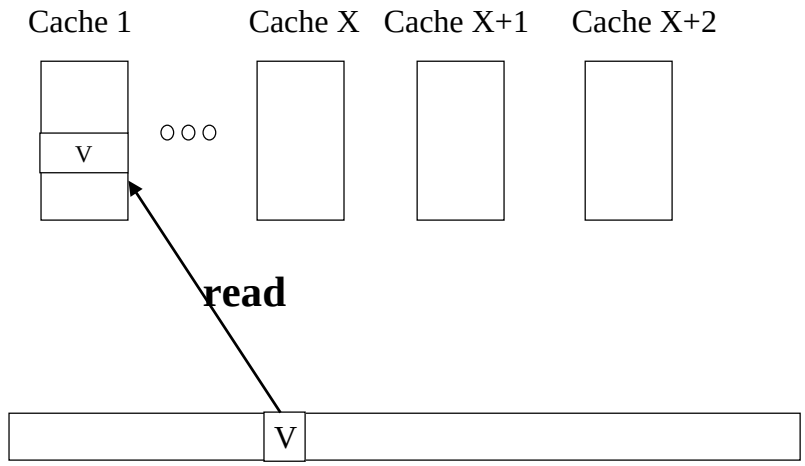


Selective Memory

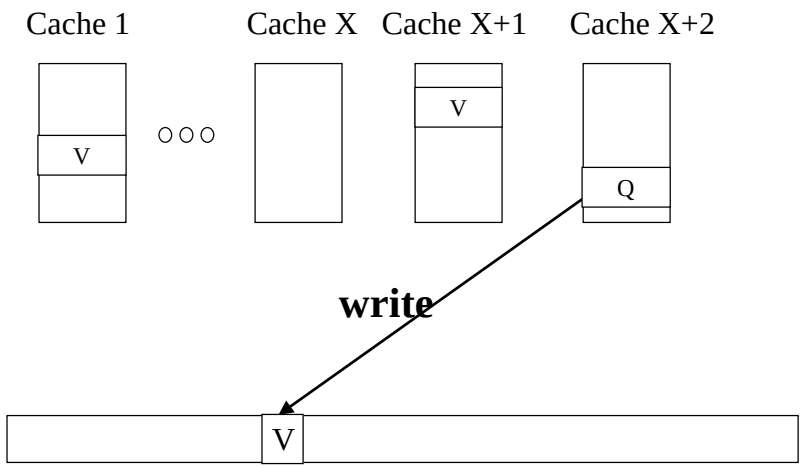
Visualisation



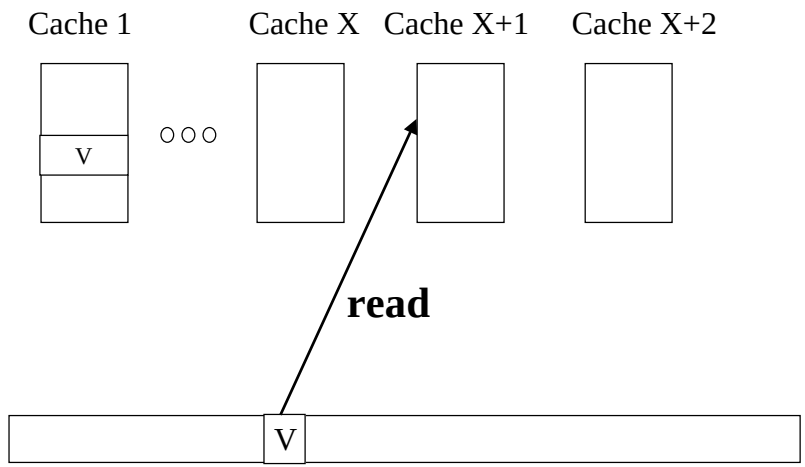
Visualisation



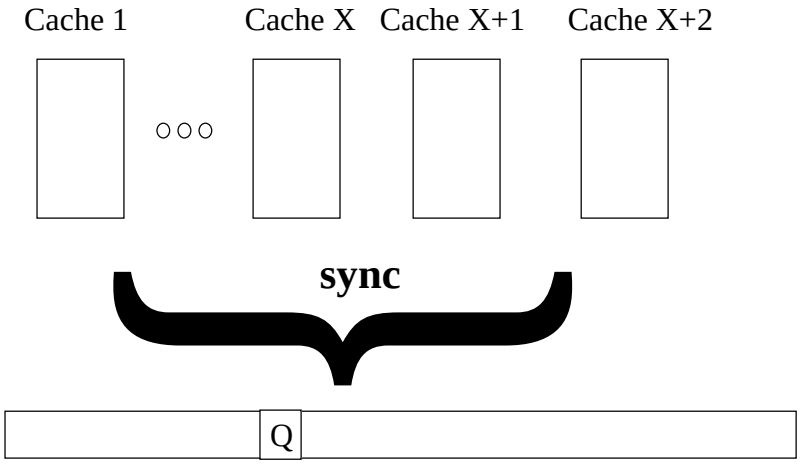
Visualisation



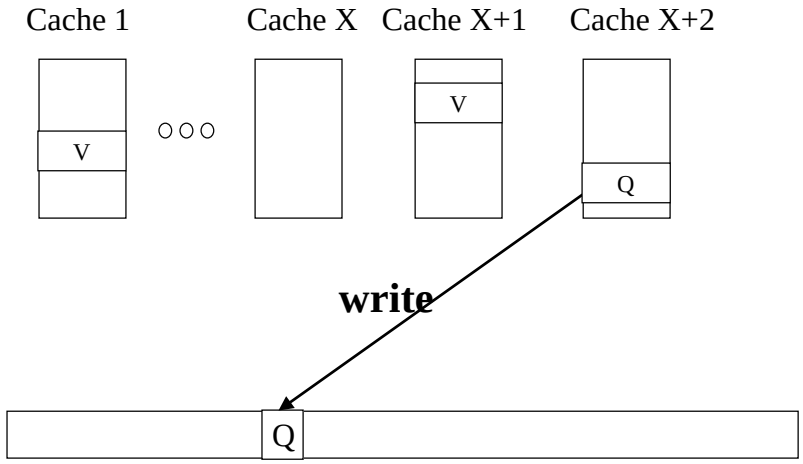
Visualisation



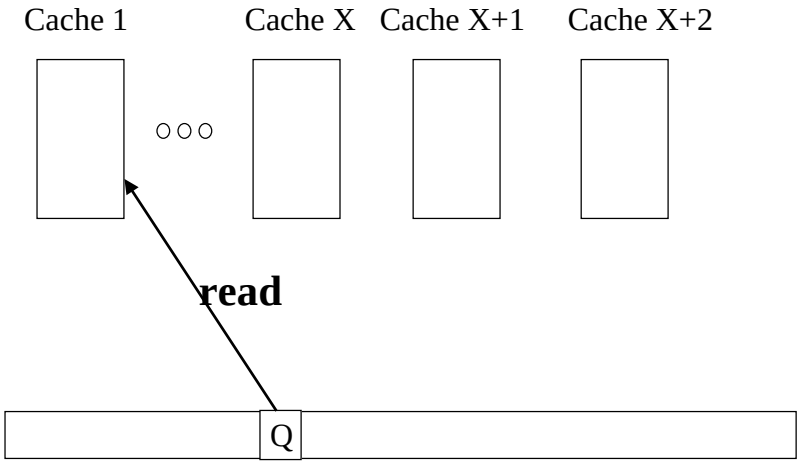
Visualisation



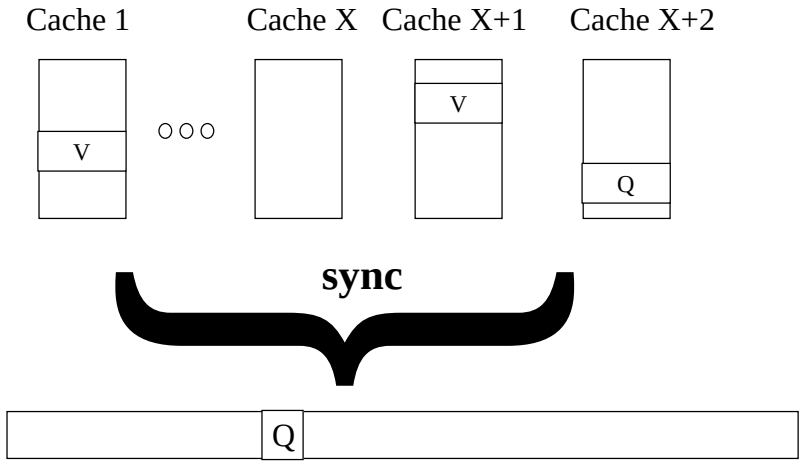
Visualisation



Visualisation



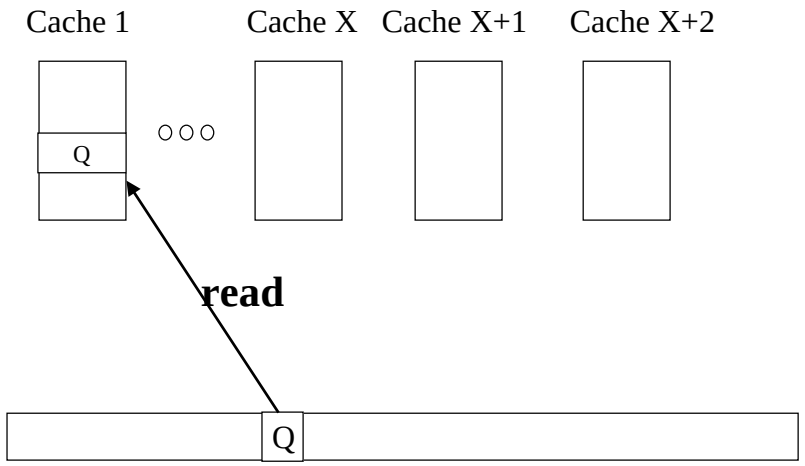
Visualisation



Conclusions

- Odin combines a range of techniques to provide a new approach to fine grain DSM.
- The strength of the design is that it combines hardware level design inovations with new user level thread and stack management methods to achieve locality.
- Obtains comparable performance to bus based machines.
- Promises scalability to at least 128 processors.

Visualisation



Classifying Architectures

- Flynn (1972), proposed four classes of parallel computer.
- The classification scheme was based on the number of concurrent instruction and data streams a machine could manipulate.
  - SISD
  - SIMD
  - MISD
  - MIMD

Advantages

- Simplifies memory design by eliminating invalidation directories
- Reduces overall network traffic for machines of up to 64 nodes
- When combined with write pipelining yields about 30% improvement over a standard weak consistency implementation.

MISD

- Multiple Instruction, Single Data Stream.
  - No such machines exist.

SISD

- Single Instruction, Single Data Stream.
  - standard von Neumann machines
  - single cpu, processing a single stream of data supplied over the bus.
  - Examples include the IBM PC, Many SUN workstations, most home, mini, and mainframe computers in current use.

MIMD

- Multiple Instruction, Multiple Data Stream.
  - Most modern parallel computers.
  - Any machine with powerful processing nodes, either distributed or shared memory.
  - Examples, Sequent Symetry, Newer SUN Sparc Servers, INTEL Paragon, etc....

SIMD

- Single Instruction, Multiple Data Stream.
  - array processing machines.
  - many cpus executing the same instruction, processing a multiplicity of data supplied in the architecture itself.
  - Examples, MasPar, Meiko Computing Surface, Connection Machine.