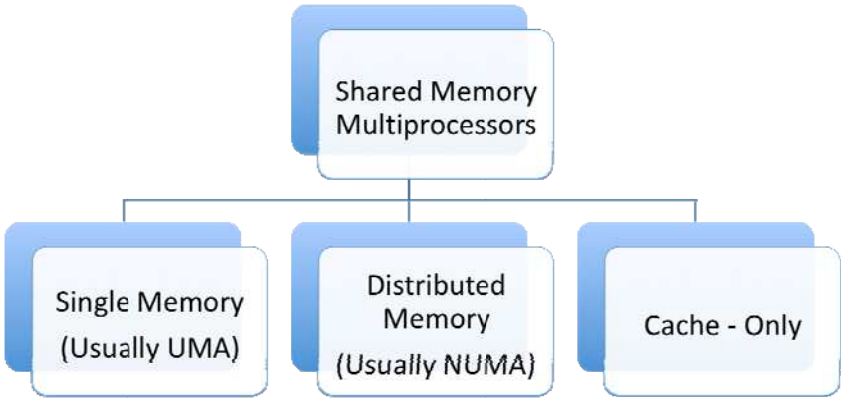
DDM - A Cache-Only Memory Architecture

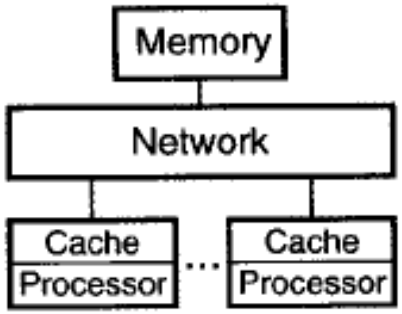
Erik Hagersten, Anders Landlin and Seif Haridi

Presented by
Narayanan Sundaram
03/31/2008

Shared Memory MP - Taxonomy

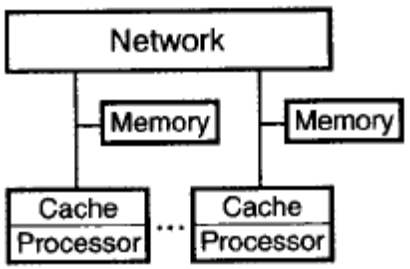


Unified Memory Architecture (UMA)



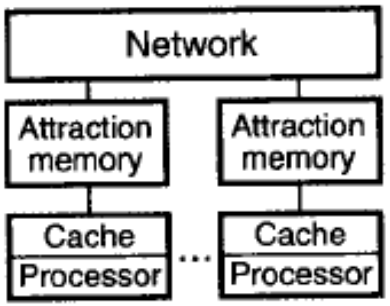
- All processors take the same time to reach the memory
- The network could be a bus or fat tree etc
- There could be one or more memory units
- Cache coherence is usually through snoopy protocols for bus-based architectures

Non-Uniform Memory Architecture (NUMA)



- The network can be anything Eg. Butterfly, Mesh, Torus etc
- Scales well – upto 1000's of processors
- Cache coherence usually maintained through directory based protocols
- Partitioning of data is static and explicit

Cache-Only Memory Architecture (COMA)



- Data partitioning is dynamic and implicit
- Attraction memory acts as a large cache for the processor
- Attraction memory can hold data that the processor will never access !! (Think of a distributed file system)
- USP: Can give UMA-like performance on NUMA architectures

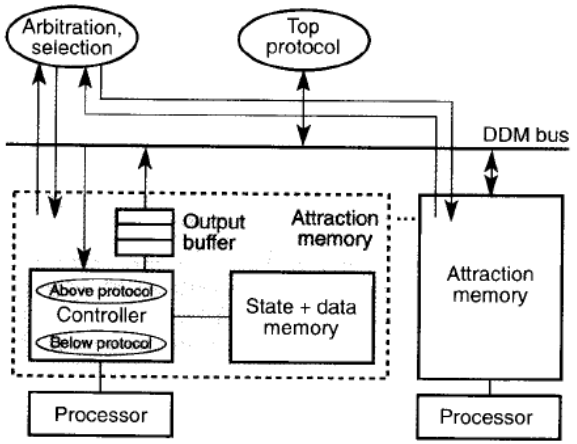
COMA Addressing Issues

- Item
  - Similar to cache line, item is the coherence unit moved around
- Memory references
  - Virtual address -> item identifier
  - Item identifier space is logically the same as physical address space, but there is no permanent mapping
- Item migration improves efficiency
  - Programmer only has to make sure locality holds, data partitioning can be dynamic

Data Diffusion Machine (DDM)

- DDM is a hierarchical structure implementing COMA
- Uses DDM bus
- Attraction memory communicates with
  - processor using below protocol
  - DDM bus using above protocol (snoopy)
- At the topmost level, node uses Top protocol

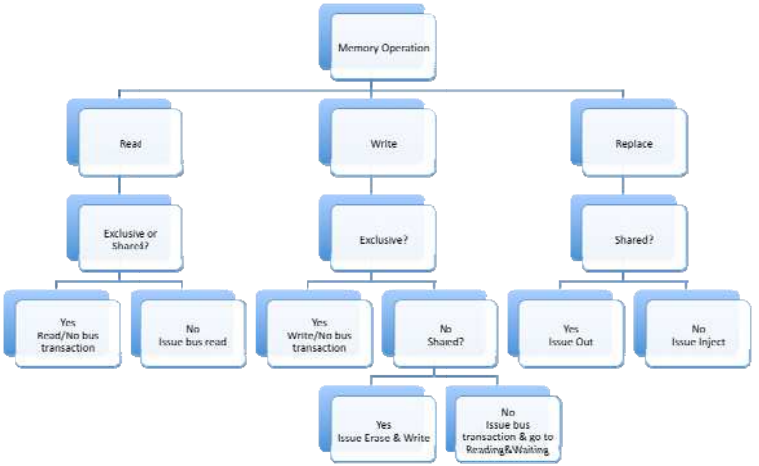
Architecture of single bus DDM



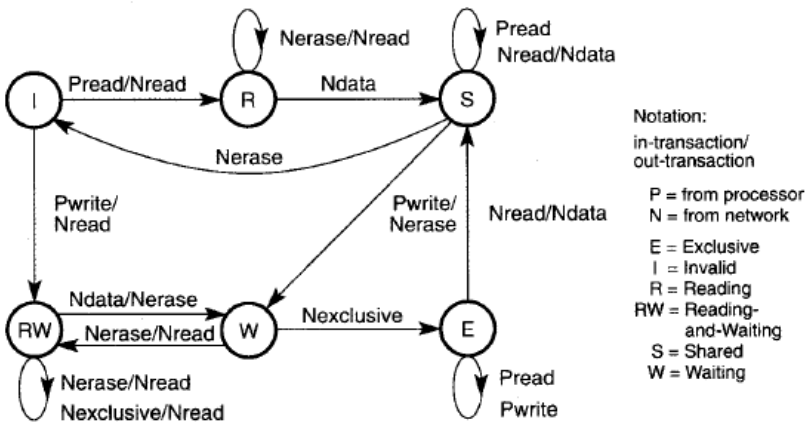
Single-bus DDM protocol

- An item can in one of the seven states
  - Invalid
  - Exclusive
  - Shared
  - Reading
  - Waiting
  - Reading and waiting
  - Answering
- The bus carries the following transactions
  - Erase
  - Exclusive
  - Read
  - Data
  - Inject
  - Out

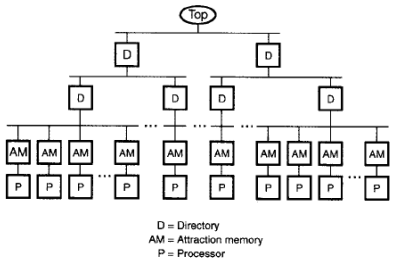
Single bus DDM protocol



Attraction Memory Protocol (without replacement)



Hierarchical DDM protocol

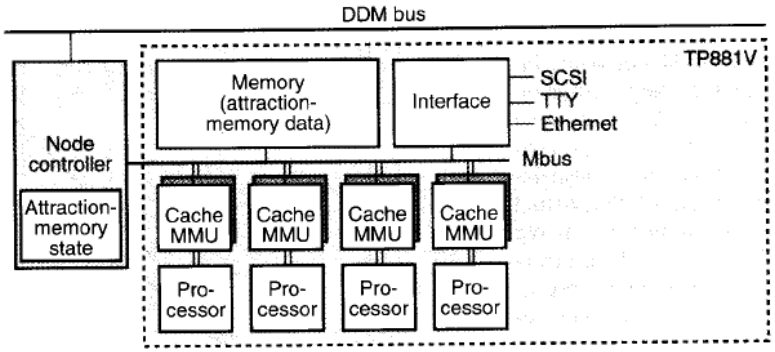


- Directory is similar to Attraction Memory, except that they do not store any data
- For the bus below, it behaves like Top protocol
- For bus above, it behaves like above protocol
- Multilevel read
- Multilevel write
- Multilevel replacement

Multilevel DDM protocol

- Directory requirement
  - Size: $Dir_{i+1} = B_i * Dir_i$
  - Associativity: $Dir_{i+1} = B_i * Dir_i$ where B_i is the branching factor for level I
  - Too much hierarchy will be costly and slow
  - Could use "imperfect directories"
- Protocol is sequentially consistent
- Bandwidth requirements
  - Fat tree network
  - Directory + Bus splitting
  - Heterogeneous networks

COMA Prototype



Prototype description

- For address translation, DDM uses normal virtual to physical address translation mechanism
- For item size = 16 bytes
  - Overhead is 6% for 32-processor system
  - Overhead is 16% for 256-processor system
- For larger item sizes, the overhead is lower, but false sharing may cause problems

Performance

100 ¬

	Water	Water	MP3D	MP3D-Diff	Cholesky	Cholesky	Matrix	Muse
Input data	192 molecules	384 molecules	75,000 particles	75,000 particles	m14 (small)	m15 (large)	500×500	Pundit
Cold start included?	yes	yes	no	no	yes	yes	yes	no
DDM topology	2×8×4	2×8×4	2×8×2	2×8×2	2×8×2	2×8×2	8×4	4×4
Hit rate (data) percent								
D cache	99	99	86	92	96	89	92	98.5
Attraction memory	44	65	40	88	6	74	98	91
Remote access rate	0.6	0.4	8.4	1.0	3.8	2.8	0.16	0.20
Bus utilization percent								
Mbus	31	26	86	54	70	60	55	—
Lower DDM bus	39	30	88	24	80	66	—	—
Top DDM bus	25	20	66	13	70	49	4	—
Speedup per number of processors	52/64	53/64	6/32	19/32	10/32	17/32	29/32	—/16

Conclusion

- COMA is middle ground between UMA and NUMA
- In the prototype, overhead is 16% in access time and 6-16% in memory
- Programmer productivity improved by not worrying about NUMA issues