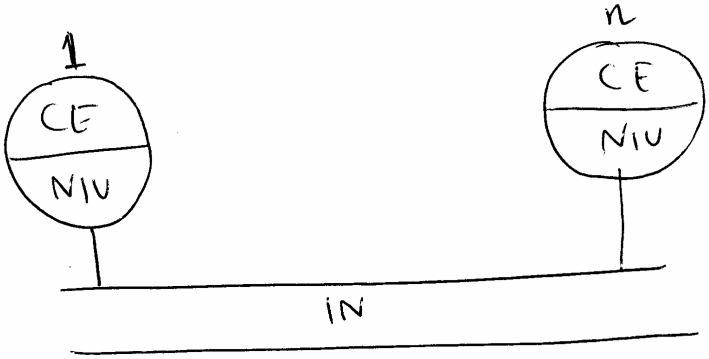
Distributed Shared Memory Model



NIU: Network Int. unit

CE: Computing Element $\{\underbrace{\text{Cache, PE, MM, D}}_{\text{Local components.}}\}$

- All nodes use a single global address space.

- If a processor issues a read request (load data from memory) it will specify the address from where data is to be retrieved.

address — a node #

address — address of location within the node.

Thus, if the node # is that of the processor issuing the read command, the data is retrieved from the memory attached to it. Otherwise, the node # is used to send a message via IN to the appropriate node's memory from where the data is to be retrived & delivered to the requesting node.

We summarize the procedure below:

Procedure A :

Command : Load from specified address into a register

1) Translate address $\{ \text{Addr} \leftarrow \text{CE\#} + \text{memory addr in CE} \}$

2) Is CE local ?

3) **If yes**, load from local CE. **Else**, send request via n/w to remote CE

4) Retrieve data from remote CE's mem.

5) Transport data via n/w to requesting CE

6) Load data in specified reg.

Procedure B:

Command: Store register in address.

(1) Translate address

(2) Is CE local ?

(3) Yes ⇒ done ; Else, send contents of reg via n/w to remote CE.

(4) Store contents of reg in the specified location in the mem of remote CE.

(5) Send message to CE which issued store command that the task is completed.

formats:

1) Load Req:

(Source CE addr, Dest. CE addr, address in mem from where data is to be loaded)

2) Retrieval data (sent over IN to the requesting CE)

(Source CE addr, Dest. CE addr., data)

3) Store instruction (store req.)

(Source CE addr, Dest. CE addr, Addr. in mem where data is to be stored, Data)

(4) Ack packet (back to originator)

(Source CF addr, Dest. CF addr/. , Store successful)

- Error detecting bits may be added to the above packets.

Timing : (for data retrieval from a remote memory)

- fixed time T needed by the system program @ the host node to issue a command over the n/w. This will include time to decode (to which remote node the req. is to be sent) & formatting a packet.

- Time taken by the load req. packet to travel via IN. This depends on bandwidth of n/w & packet size, i.e., $\left(\frac{n}{B}\right)$secs, where n: pkt size in bytes & B is the bandwidth (bytes/sec)

- Time taken to retrieve word from remote memory, q.

- Fixed time T needed by the destination CE system program to access the N/W.

- Time taken by dest. CE to transport the reply packet (which contains the data retrieved) over the N/W, i.e., (m/B) secs where m : size of the packet.

Thus, total time: $2T + q + \frac{(n+m)}{B}$

Note: For store operation time taken is similar, however m & n may be different.

Note: Above model does not claim to be an exact model. Can communication & computation be time-overlapped?

Anothe issue is how frequently service requests to a remote processor are issued by a CE.

A full-fledged Example

System: NUMA parallel computer

# of nodes: 256 CEs

Each CE has 16MB memory

In a set of programs, 10% of instructions are loads and 15% are stores.

memory access time for local load/store : 5 clock cycles

Overhead to initiate transmission of a request to a remote CE : 20 clock cycles

Bandwidth of IN : 100 MB/sec

Assume 32 bit words & a clock cycle time of 5 nsec

Now, if 400,000 instructions are to be executed,

compute:

1. Load/store time if all accesses are to local CEs
2. Repeat (1) if 25% of accesses are to a remote CE.

Solution:

(1) No. of load/store instructions

$= 400,000 \times \frac{1}{4} = 100,000$

Time to execute load/store locally

$= 100,000 \times 5 \times 5 = 2500\ \mu secs$ ... (1)

---

(2) · # of load instructions : 40,000. (10%)

# of local loads : $40,000 \times \frac{3}{4} = 30,000$

$\left.\begin{array}{r}\text{Time taken for local}\\ \text{loads}\end{array}\right\} = 30,000 \times 25$

$= 750\ \mu secs.$

# of remote loads : 10,000

req. packet format : $\left(\begin{array}{l} \text{Src ; 8 bits} \\ \text{dest : 8 bits} \\ \text{addr : 24 bits} \end{array} \right\}$ 40 bits.

∴ request packet length = 5 bytes.

Response packet length = $\underline{\text{6 bytes}}$ $\left\{ \begin{array}{c} \text{32 bits} \\ \text{per word} \end{array} \right\}$

(Src : 8, Dest : 8 ; Data : 32 bits)

Time taken for remote load of one word =

$20 \times 5 \times 10^{-9}$ (overhead, fixed) ←

$+ \frac{5}{100} \times 10^{-6}$ (transmit a req. back)

$+ 5 \times 5 \times 10^{-9}$ (data retrieval)

$+ 20 \times 5 \times 10^{-9}$ (overhead, fixed) ←

$+ \frac{6}{100} \times 10^{-6}$

$= 335$ nsecs.

# of requests to remote CE = 10,000

Total time : 3350 μsecs

Time for local loads = 750 μsecs (see page 7)

Total time taken for loads : 3350 + 750

= 4100 μsecs

# of Store instructions:

$400,000 \times 0.15 = 60,000$

# of local stores: $60,000 \times 0.75$

$= 45,000$

# of remote Stores = 15,000

Time for local stores: $45,000 \times 25 = \underline{1125\ \mu secs}$.

Time taken for $\underline{1}$ remote store:

$20 \times 5 \times 10^{-9}$

$+ \frac{9}{100} \times 10^{-6}$ (remote store packet length: 9 bytes)

$+ 25 \times 10^{-9}$

$+ 20 \times 5 \times 10^{-9}$

$+ \frac{3}{100} \times 10^{-6}$ (ack packet length 3bytes)

$\underline{345\ nsecs}$.

Time to store 15,000 words

$= 345 \times 15,000 = \underline{5175}$ μsecs

Time for local stores = 1125 μsecs (see page 9)

$\Rightarrow$ Total time for stores =

$$\begin{array}{r} 5175\ + \\ 1125 \\ \hline \underline{6300} \end{array}$$ μsecs

Total time for loads + stores :

$$\begin{array}{r} 4100\ + \\ 6300 \\ \hline \underline{10,400} \end{array}$$ μsecs . (2)

$$\frac{\text{Total time for load \& store (local + remote)}}{\text{Total time for load \& store if enitrely local}} = \frac{(1)}{(2)} =$$

$$= \frac{10,400}{2500} = \underline{\underline{4.16}}.$$

<u>Inference</u> : ?