A Case for Virtual Memory Mapped Network Interface

Kai Li
Princeton University

Future Nodes Will Be Commodity PCs

"Commodity Chip" MPPs won't make it

MPPs	Processor	Year	WS	Price/node	Future
T3D	150 Mhz Alpha	93/94	92/93 (/w L2)	70-100k	Recycle
Paragon	50 Mhz i860	92/93	91	30k	Recycle
CM5	32 Mhz SS-2	91/92	89/90	30k	Recycle

Expensive and 1-2 year lag in performance

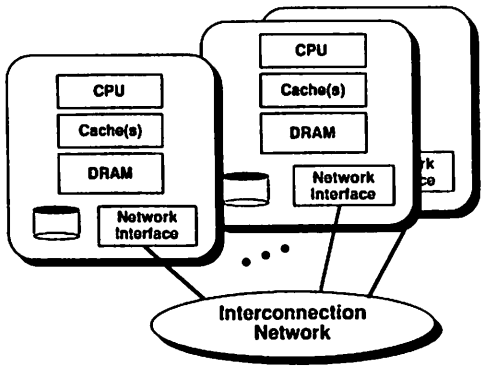
More commodity components help, but not quite

SP2	R6000	93/94	same	WS+10k	Recycle

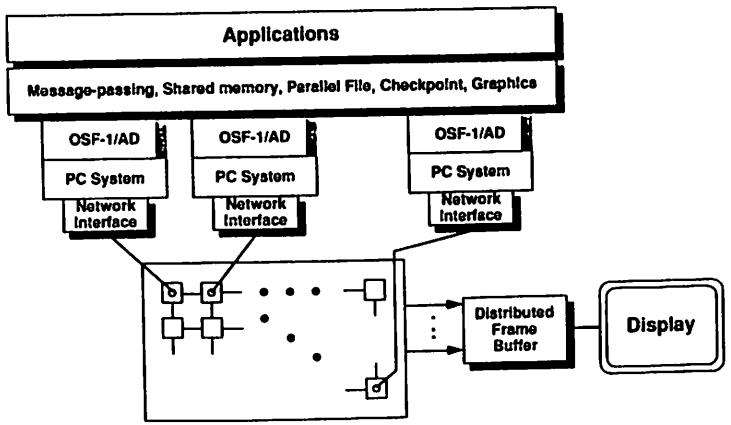
SHRIMP or NOW

Shrimp	Pentium/Alpha /PowerPC			PC+2k?	Grad office

Converged Architectures for Parallel Computing?



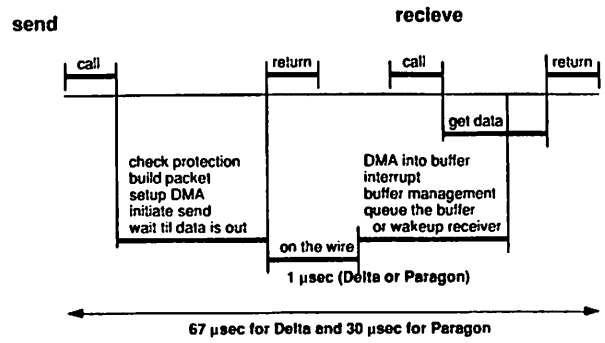
SHRIMP



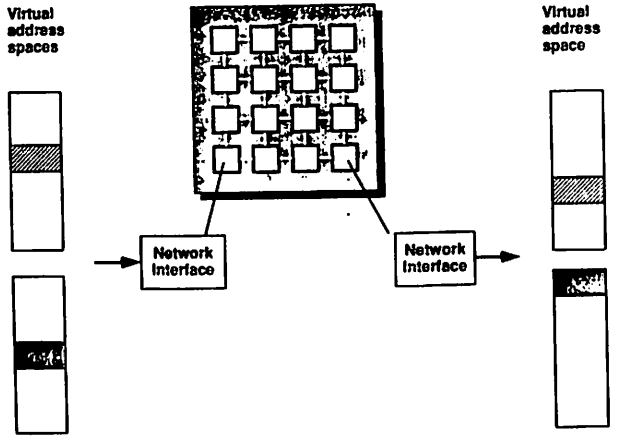
What is the "Low Level" Programming Models

- Two decisions for shared memory
  - Implements shared memory coherence in hardware
  - Support software shared memory
- Three things for message passing
  - Data transfer
  - Control transfer
  - Protection
- SHRIMP approach
  - Support software shared virtual memory
  - Virtual memory mapped communication for message passing

Message Passing Overhead



Virtual Memory Mapped Communication



I/O Bandwidth > 2 * bcopy Bandwidth ?

	CPU	Clock (Mhz)	bcopy 4MB (MB/sec)	Peak I/O (MB/sec)	Mem:I/O Ratio
Express PC	i486	50	11.8	33	.36
SUN SPARC-2	SPARC	40	10.3	88	.12
DEC-5000/200	R3000	25	14.0	100	.14
DEC-5000/240	R3000	40	23.6	100	.24
SGI Indigo	R3000	33	18.6	132	.14
HP 730	HP-PA	66	24.7	132	.18
DEC AXP/500	Alpha	150	41.6	100	.41

A Typical Multicomputer Program Case

Original approach

```
LOOP {
    ...
    send( dest, buffer, n )
    ...
    }
```

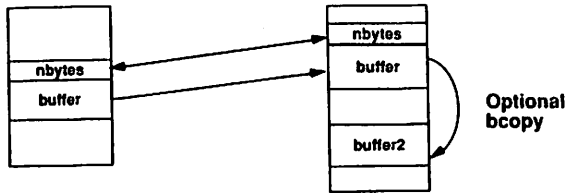
SHRIMP approach

```
map( send_buf, dest, recv_buf )
LOOP {
    ...
    send( send_buf, n )
    ...
    }
```

Data movement in loop is in user-level and has no protection latency

Buffer management has been moved to the user level.

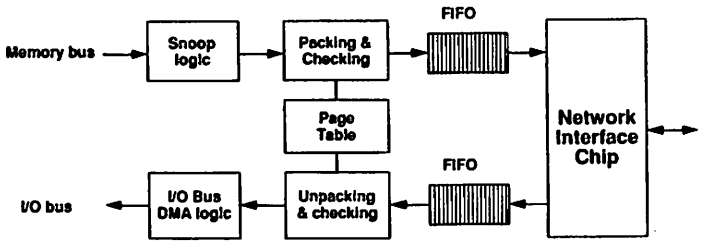
Single-Buffer Message Passing




```
while ( nbytes == 0) ;
compute
put data in buffer
...
nbytes = n;
```

```
while ( nbytes != 0) ;
bcopy( buffer2, buffer, n);
nbytes = 0;
consume
...
```

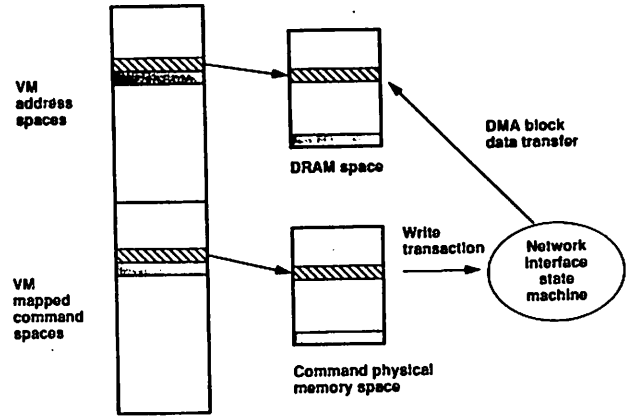

Data Path of VM Mapped Network Interface



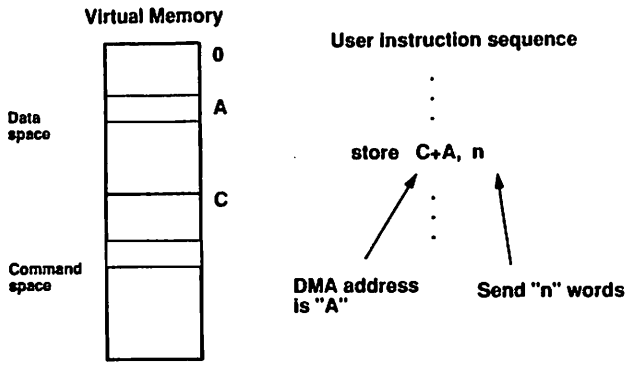
Shrimp Latency

Outgoing latency	200 nsec
Nework latency	400 nsec (16 node backplane)
Incoming latency	1250 nsec
Total	**1.85 µsec**

VM Mapped Command Space



User-Level DMA Block Transfer



Conclusion

Virtual memory mapped network interface
- Supports fine-graine, protected user-level communication
- Supports latency hiding
- Supports shared virtual memory

Our current implementation
- Takes < 10 instructions in message passing
- Total overhead < 2 usecs
- Fast interrupt is adequate for kernel handler < 2 usecs
- A simple design