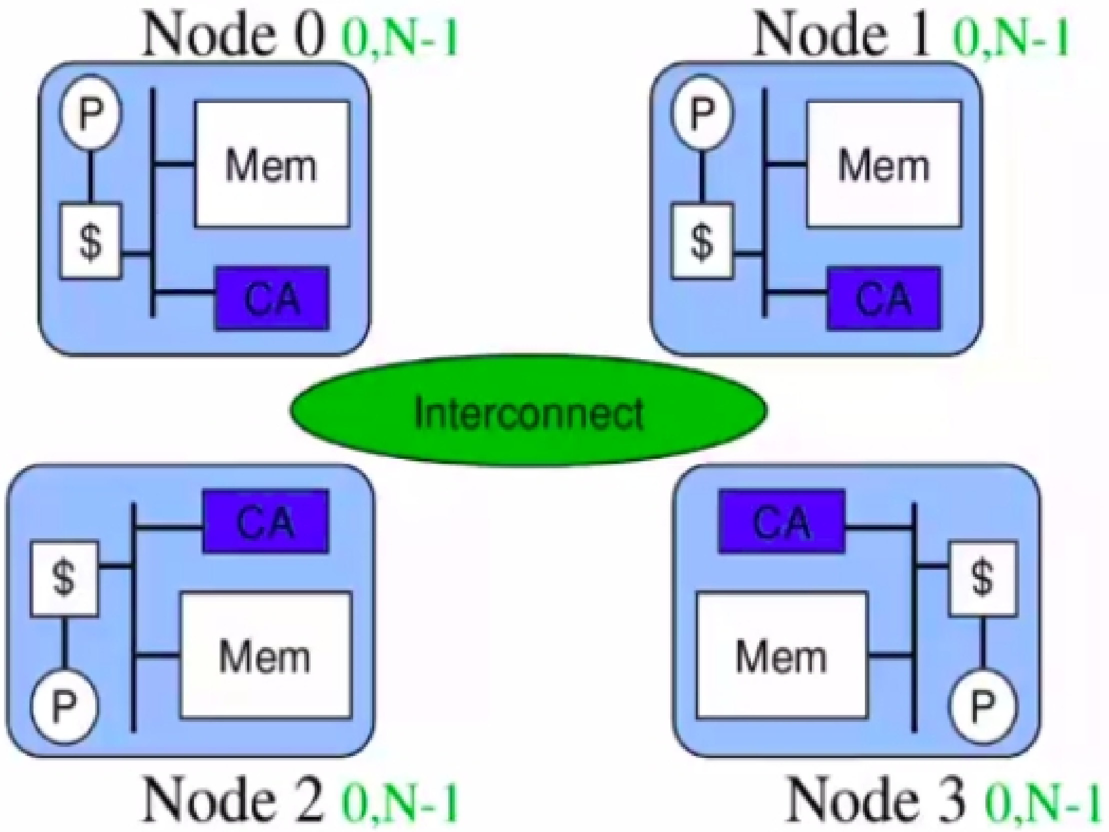
Message Passing

Message Passing

- Message passing architectures have computers as building blocks (processor + I/O system) + provides communication between processors as explicit I/O operations
- Takes help of OS and library calls for actual implementation
- Unlike local area networks, this has more tight integration
- This is processor to processor communication
- Common user-level operations are variants of SEND/RECEIVE
- SEND: specifies local buffer that has to be transmitted + to whom (receiving process)
- RECEIVE: specifies sending process + local buffer into which the received data has to be kept

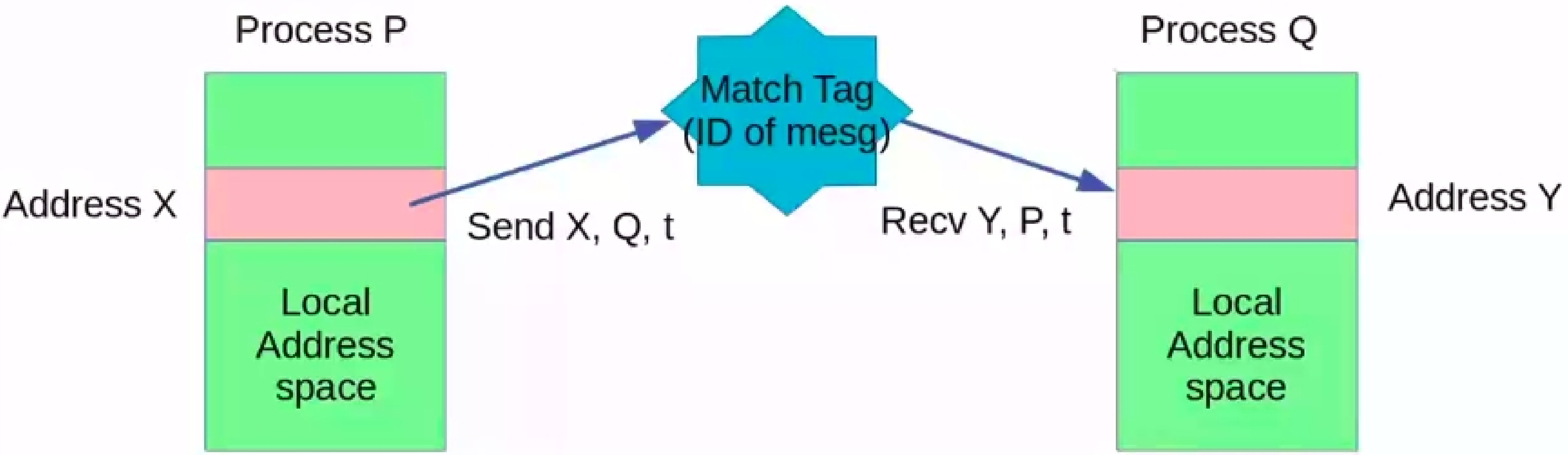
Message Passing Architectures



- Cannot directly access memory on another node

- IBM SP-2, Intel Paragon, clusters of PCs (“beowulf”)

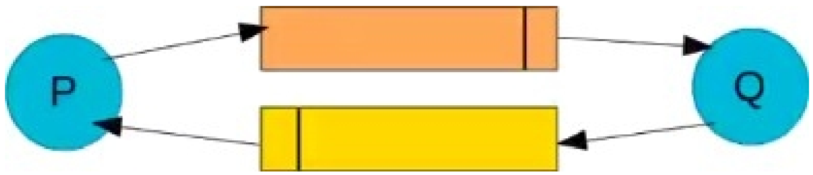
Message Passing

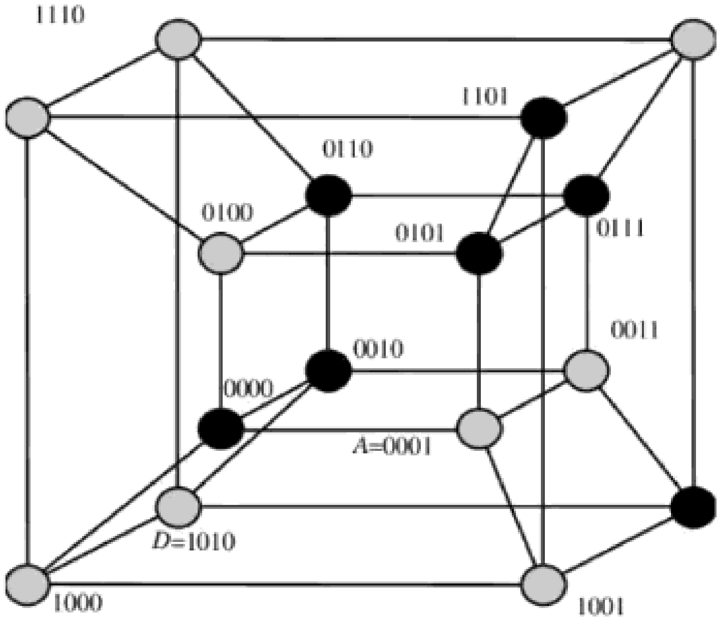


Message Passing

- Send and Receive accomplish a pairwise synchronisation event and perform a memory-to-memory copy, as each process provides its local address
- Possible variants:
  - Whether Send
    - completes when receiver completes
    - When is send buffer available for reuse
    - When new requests can be accepted
  - Whether receiver
    - Waits till matching send occurs Or simply post the request
- Message passing was used in earlier programming langauges like: CSP, Occam and OS functions like sockets

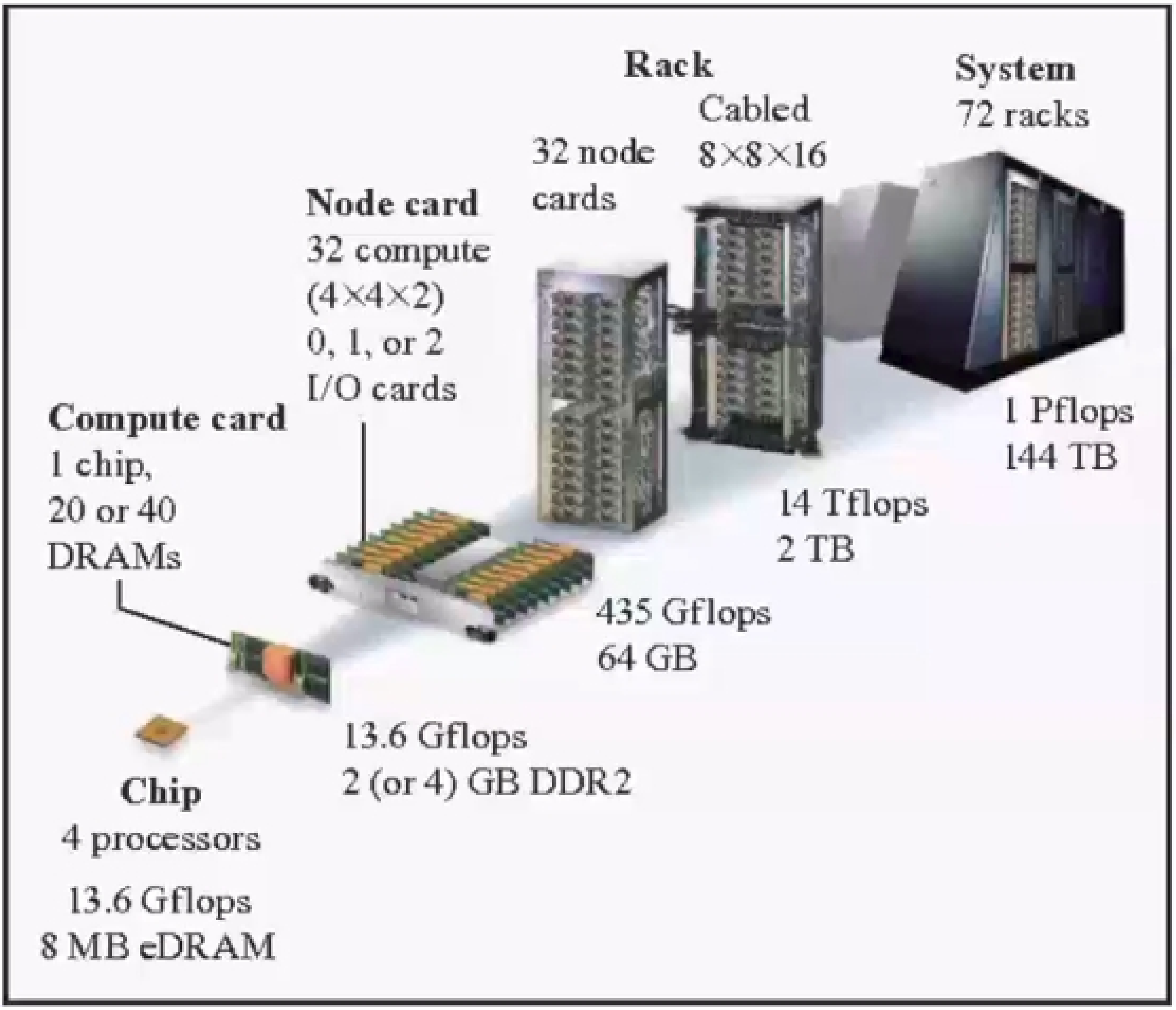
Message passing



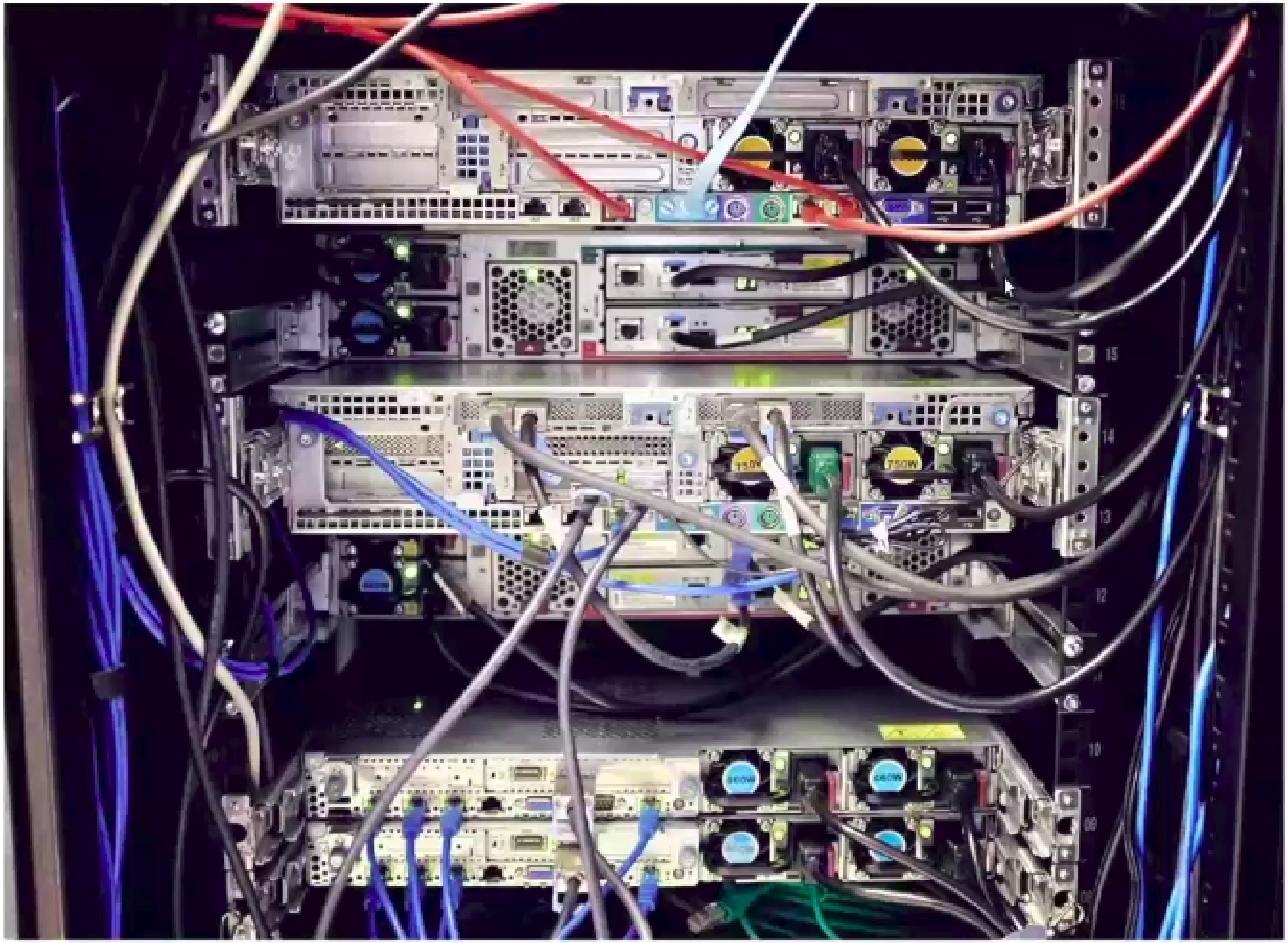


- Earlier message passing machines used point-to-point network
- Used FIFO to send data. Network topology was important
  - Sender wrote to the link and Receiver read from the link
  - Sender will block until receiver reads the value = called synchronous message passing
- Some used DMA + dedicated processor for send/receive
  - Allowed non-blocking send
- Later developments allowed message to go to any destination via other nodes. Network level routing protocols
  - Store-and-forward networks
  - Latency depended on number of hops traversed

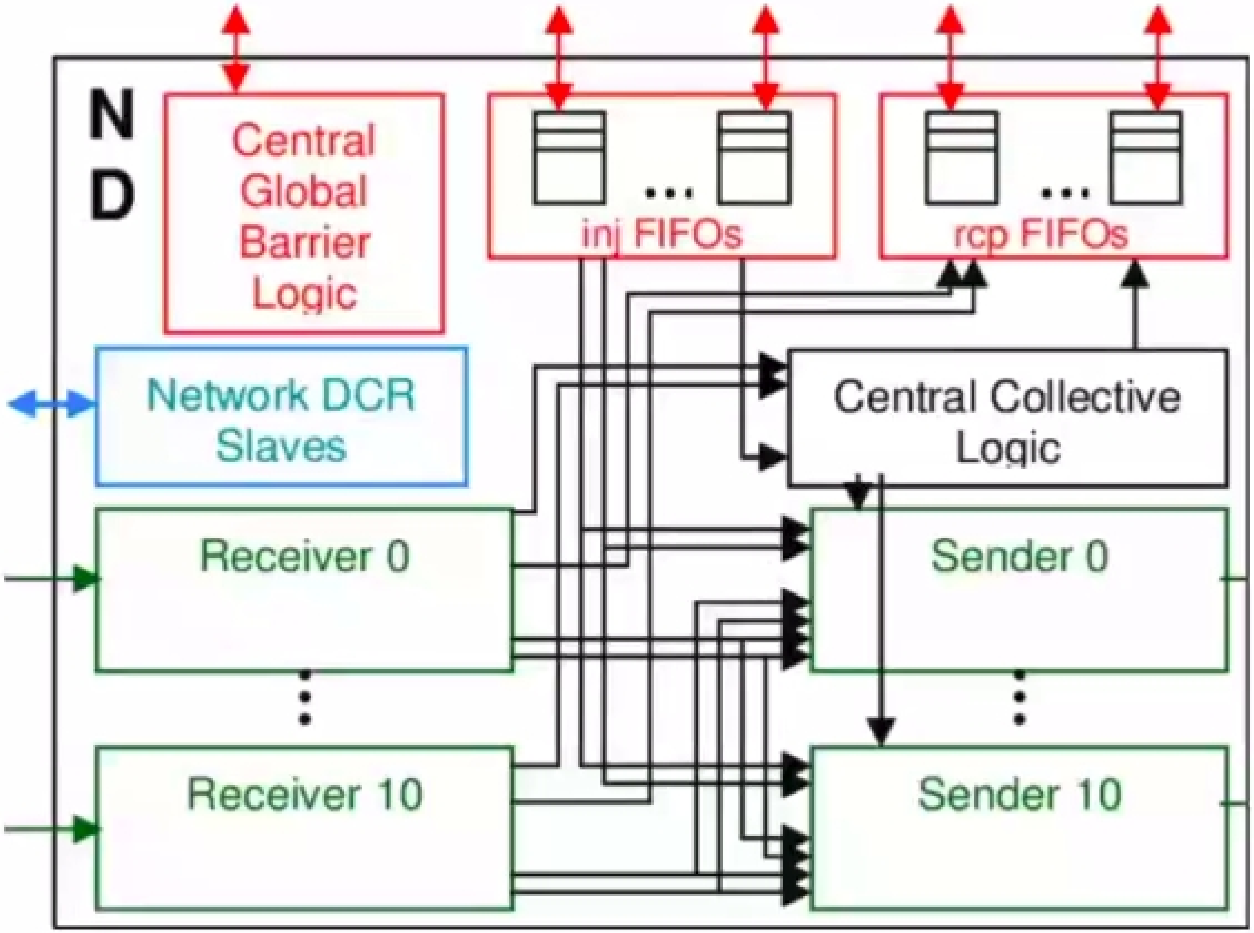
IBM Blue Gene/P supercomputer



Infiband network



Blue Gene Router logic


N
D
Central Global Barrier Logic
inj FIFOs
rcp FIFOs
Network DCR Slaves
Central Collective Logic
Receiver 0
Receiver 10
Sender 0
Sender 10