MPI Collectives

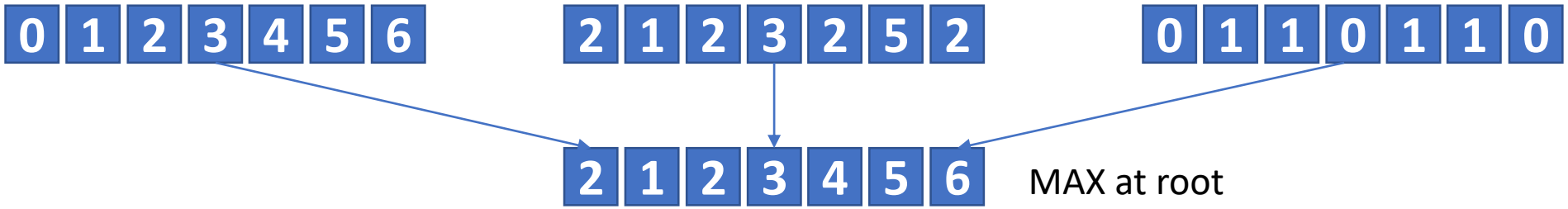
Jan 18, 2019

Previous Class

- MPI_Barrier
- MPI_Bcast
- MPI_Gather
- MPI_Allgather
- MPI_Scatter
- MPI_Alltoall
- Vector variants

Reduce

- MPI_Reduce (inbuf, outbuf, count, datatype, op, root, comm)
- Combines element in inbuf of each process
- Combined value in outbuf of root
- op: MIN, MAX, SUM, PROD, …



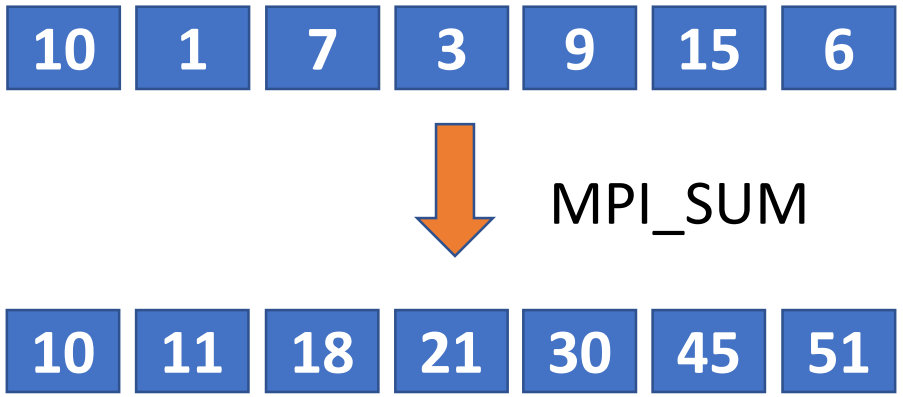
Allreduce

- MPI_Allreduce (inbuf, outbuf, count, datatype, op, comm)
- op: MIN, MAX, SUM, PROD, …
- Combines element in inbuf of each process
- Combined value in outbuf of each process
- inbuf may be MPI_IN_PLACE

Equivalent collective?

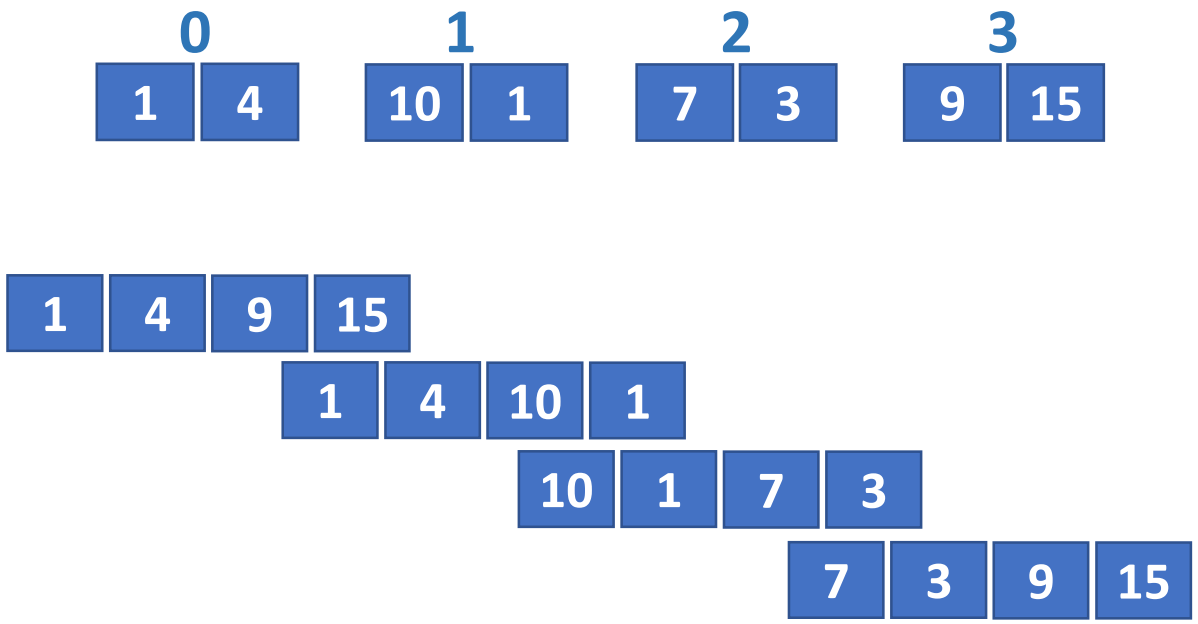
Scan

- MPI_Scan (inbuf, outbuf, count, datatype, op, comm)
- op: MIN, MAX, SUM, PROD, …
- Perform a prefix reduction on distributed data
- Reduction of values in the send buffers of processes with ranks 0:i-1 is returned in receive buffer of rank i



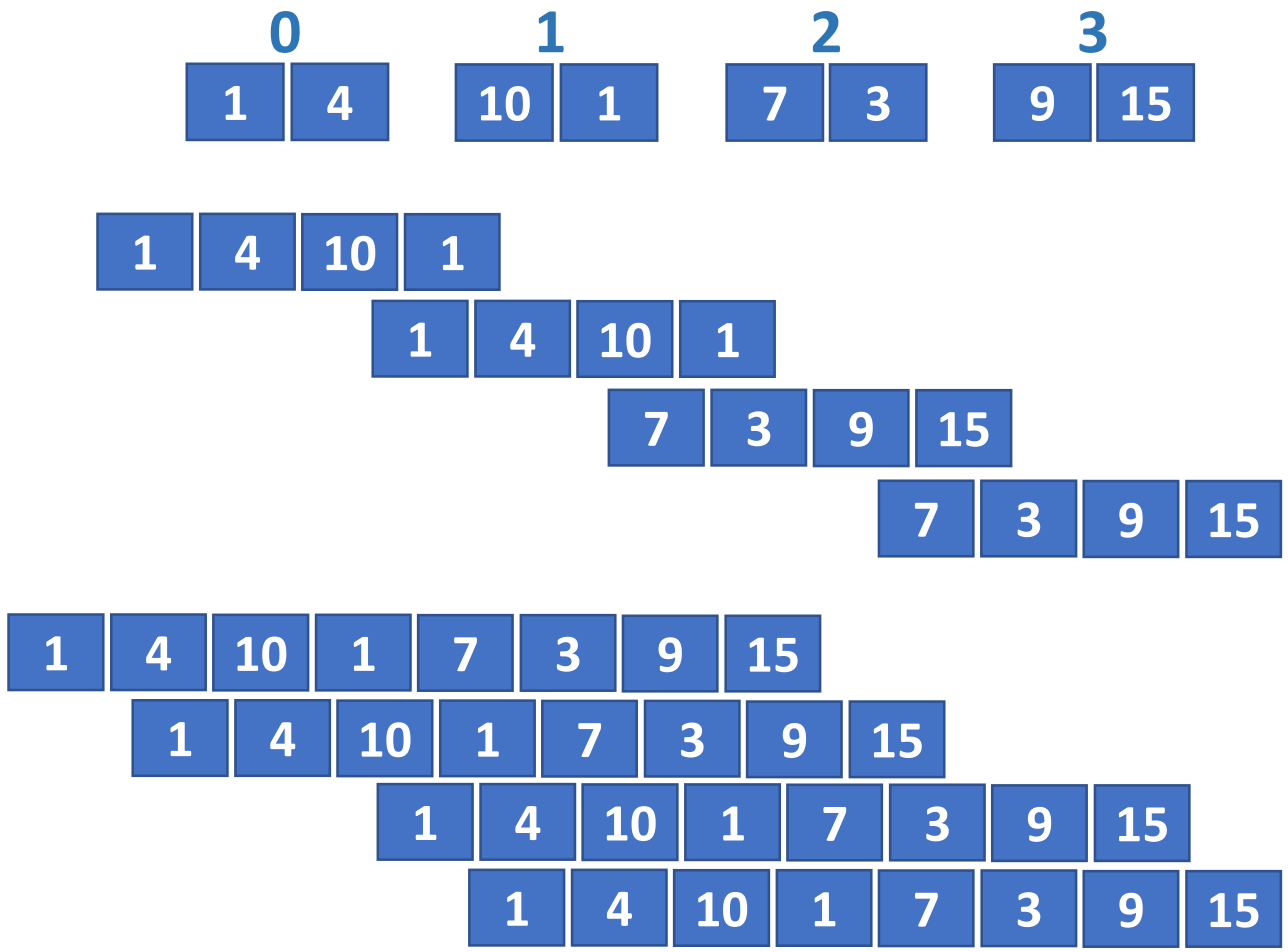
Allgather – Naïve Algorithm

- Every process sends to and receives from everyone else
- Assume p processes and total n bytes
- Every process sends and receives n/p bytes
- Ring algorithm
- Time ?
  - (p – 1) * (l + n/p*(1/b))
- How can we improve?



Allgather – Recursive Doubling

- Every process sends and receives $(2^{k-1})^*$ n/p bytes in step k
- Time ?
  - (log p) * l + (p-1)*n/p*(1/b)

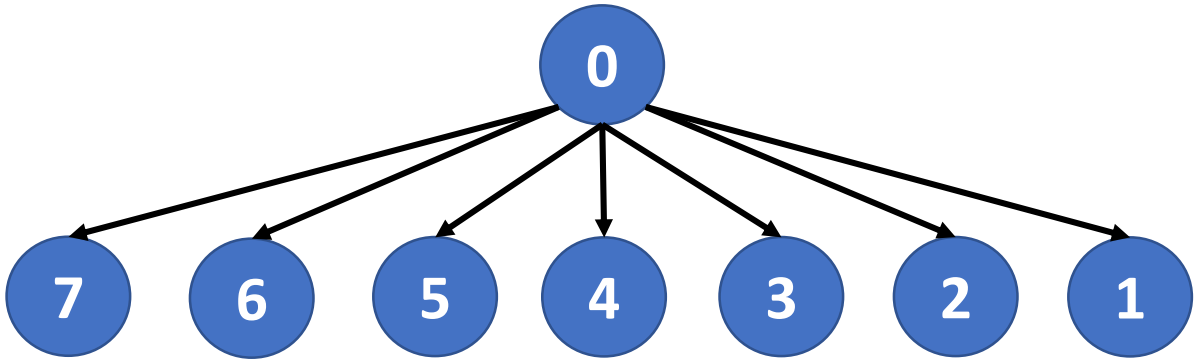


Broadcast – Naïve Algorithm

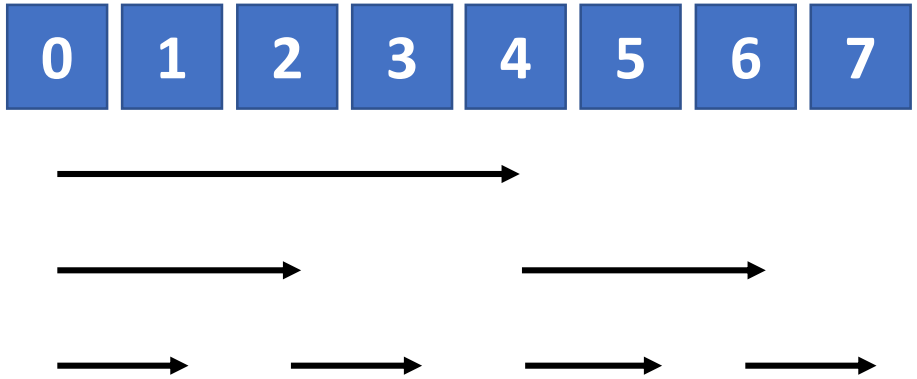
- Root process sends to every other process

Cons

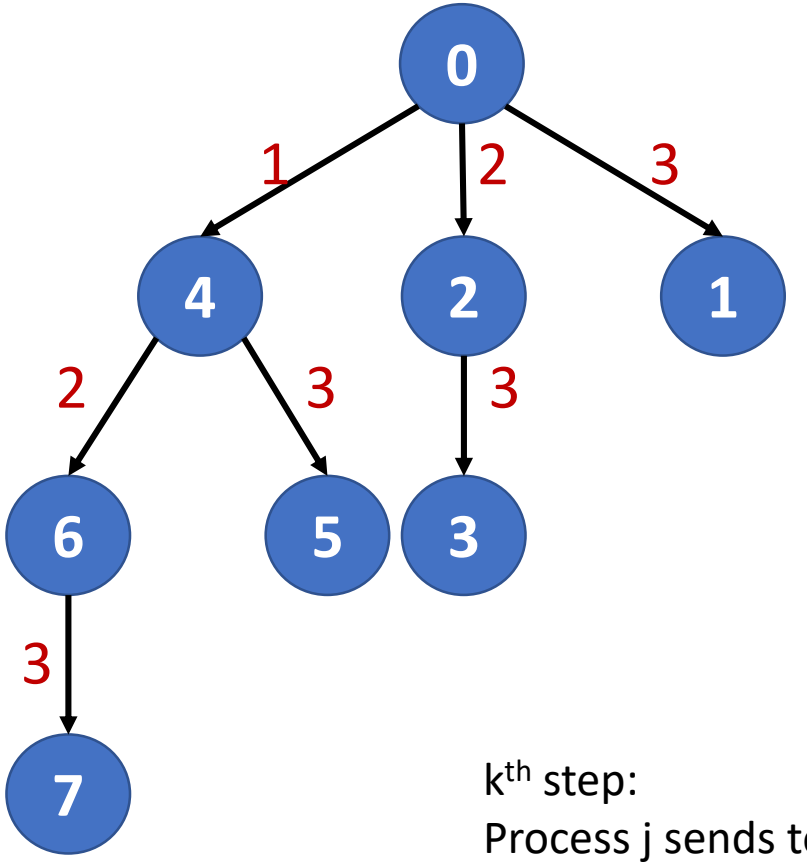
- Root is a bottleneck
- Idling processes
- Communication links are under-utilized



Broadcast – Binomial Tree



- #Steps for p (=2^d) processes?
  - log p
- Transfer time for n bytes
  - T(p) = log p * (l + n/b)
  - $T(p^2)$ = 2 log p * (l + n/b)



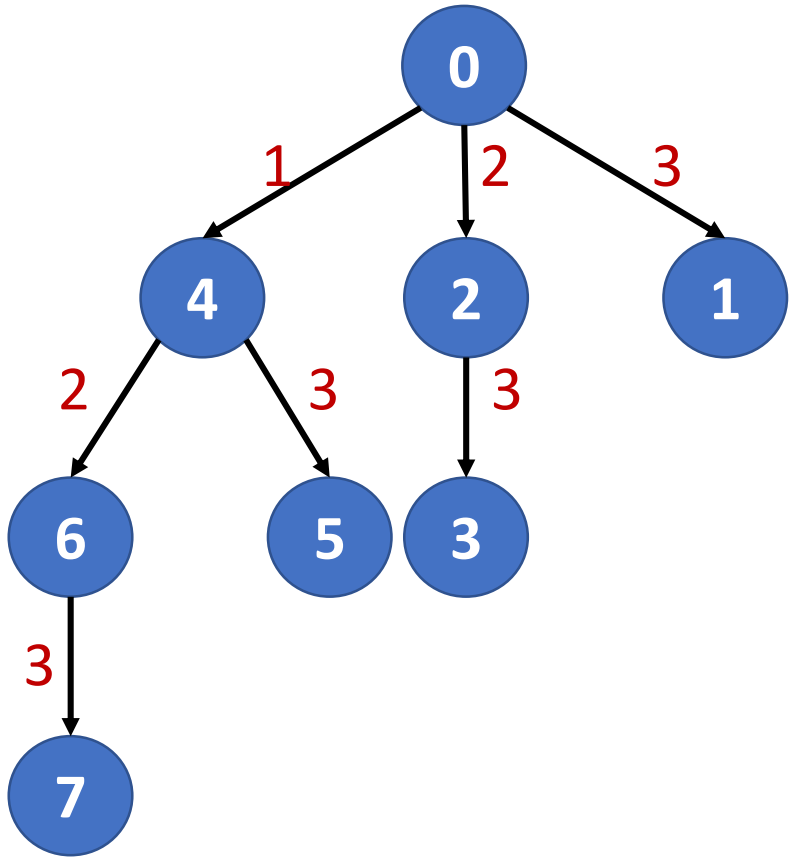
k^{th} step:
Process j sends to $j \oplus 2^{d-k}$

Broadcast Algorithm

Q: Which interconnect would most likely exhibit minimum link contention for binomial tree broadcast algorithm?

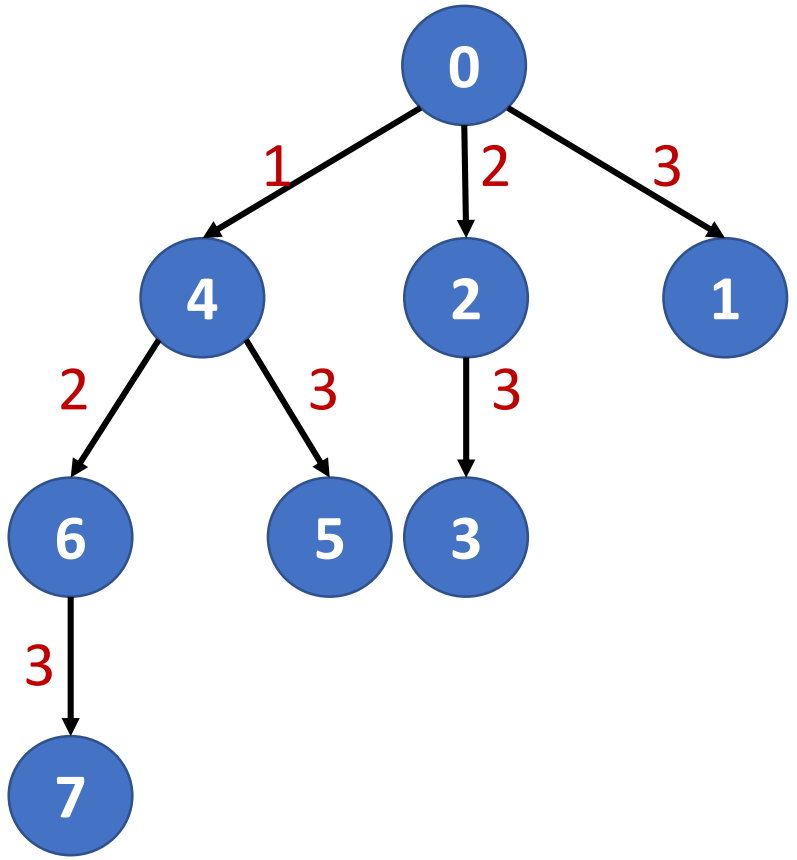
Q: What about non-power of 2 processes?

Q: Equivalent collective?



Time Analysis

- Time for broadcasting n bytes from root
  - log p * (l + n/b)
  - Latency term: log p
  - Bandwidth term: log p
- Time for scatter of n bytes from root
  - log p * l + (p-1)*(n/p)*(1/b)
- Time for allgather (ring) of n/p bytes
  - (p-1) * l + (p-1)*(n/p)*(1/b)
- Time for broadcast of n bytes using scatter and allgather
  - (log p + p-1) * l + 2((p-1)/p)*(n/b)



Broadcast Algorithms in MPICH

- Short messages
  - < MPIR_CVAR_BCAST_SHORT_MSG_SIZE
  - Binomial
- Medium messages
  - Scatter + Allgather (Recursive doubling)
- Large messages
  - > MPIR_CVAR_BCAST_LONG_MSG_SIZE
  - Scatter + Allgather (Ring)