ALL-TO-ALL COMMUNICATION ON MESHES WITH WORMHOLE ROUTING

Rajeev Thakur and Alok Choudhary
Dept. of Electrical and Computer Eng. and Northeast Parallel Architectures Center Syracuse University

April 28, 1994
• All-to-All Communication
• Architectural Model
• Performance Model
• Direct Algorithms
• Indirect Algorithms
• Performance on the Touchstone Delta
• Conclusions
All-to-All Communication

- Each processor needs to communicate with every other processor.
- Also called Complete Exchange.
- Densest communication pattern.
- Can result in a lot of node and link contention.
- Parallel quicksort, 2D FFT, matrix transpose, array redistribution etc.
Previous Work

- Power-of-two number of processors.
- Avoid node and link contention.
- Hence require more than $p$ steps.

Contributions of this Paper

- Communication links on machines like Touchstone Delta and Paragon have excess bandwidth and can tolerate small amount of link contention.
- Hence, allow small amount of link contention to exist.
- Algorithms require less than $p$ steps.
- Algorithms for power-of-two as well as non power-of-two meshes
- Intel Touchstone Delta.
- $16 \times 32$ mesh of Intel i860 microprocessors.
- Uses wormhole routing.
- Hence communication time independent of distance.
- Uses deterministic X–Y routing.
- Bidirectional communication links.
- Links have excess bandwidth.
- Two messages can travel on the same link in the same direction without increasing communication time.
**Performance Model**

- $\alpha$ = message startup time
- $\beta_{ex}$ = transfer time per byte for an exchange with no link conflicts
- $\beta_{sat}$ = transfer time per byte on a saturated link
- $L$ = number of bytes to be exchanged per processor pair
- $f(i)$ = maximum number of messages contending for a saturated link at step $i$
- Time required for exchange step $i$ is
  \[ T = \alpha + L \max (\beta_{ex}, f(i) \beta_{sat}) \]
- We assume that $\beta_{ex} = 2^\gamma \beta_{sat}$ for some integer $\gamma$.
- For the Delta, $\gamma = 1$ (Ref. Barnett et al, IPPS '93)
Direct Algorithms

- Data is sent directly from source processor to destination processor.

Indirect or Store-and-Forward Algorithms

- Data is sent from source processor to destination processor through one or more intermediate processors.
- Reduces the number of steps.
- Reduces the link contention in each step.
- More data is communicated per step.
- Involves the overhead of rearranging and buffering data.
Pairwise Exchange for Power-of-Two Mesh (PEX)

- Direct Algorithm.
- On a hypercube, it guarantees no contention at any step.
- In step $i$, $1 \leq i \leq p - 1$, processor $x$ exchanges data with processor $x \wedge i$, where $\wedge =$ bit-wise exclusive-or.
- Requires $p - 1$ steps.
- The entire communication pattern is decomposed into a sequence of pairwise exchanges.
Pairwise Exchange for General Mesh (PEX-GEN)

- PEX works only for power-of-two mesh.
- PEX-GEN is an extension of PEX for any arbitrary mesh.
- Find the smallest power-of-two (say $q$) greater than the number of processors.
- Use this to schedule $q - 1$ steps of pairwise exchange.
- In each step, every processor checks to see if the calculated destination processor number is less than the actual number of processors.
- If so, it exchanges data with the processor, else it goes ahead to the next step.
- Requires $q - 1$ steps where $q = 2^{\lceil \log p \rceil}$.
**General Algorithm for any Mesh (GEN)**

- PEX-GEN requires $q - 1$ steps where $q = 2^{\lceil \log p \rceil}$.
- The GEN algorithm requires $p - 1$ steps even if $p$ is not a power-of-two.
- In step $i$, a processor $j$ sends data to processor $\text{mod}(j + i, p)$ and receives data from processor $\text{mod}(j - i + p, p)$.
- Processor pairs do not exchange with each other.
- Each processor sends data to some processor and receives data from some other processor.
Recursive Exchange (REX)

- Indirect Algorithm.
- The mesh is first halved in the $x$ direction.
- Each processor sends all data destined for the other half of the mesh to the corresponding processor in the other half.
- The mesh is further halved recursively in the $x$ direction and this process is repeated.
- This takes $\lg c$ steps, where $c$ is the number of columns in the mesh.
- The same process is repeated in the $y$ direction
- This takes $\lg r$ steps, where $r$ is the number of rows in the mesh.
- Total number of steps = $\lg c + \lg r = \lg p$
Indirect Pairwise Exchange (IPEX)

- Aims at reducing link contention in Pairwise Exchange.
- Each processor communicates only with processors in its row and its column.
- The complete exchange along columns is nested within the complete exchange along rows.
- Each simple exchange along rows is followed by a complete exchange along columns.
- Requires $p - 1$ steps.
Performance of Algorithms on a $16 \times 32$ mesh

- IPEX performs the best followed by PEX, GEN and REX
• For small message sizes (< 1Kbytes) and small number of processors (< 64), GEN performs the best.
• For 64 to 256 processors, PEX performs the best.
• For larger meshes, IPEX performs the best.
Observed v/s predicted times for PEX and GEN
Observed v/s predicted times for REX and IPEX

![Graph showing observed vs predicted times for REX and IPEX. The x-axis represents Message Size (bytes) ranging from 1000 to 16000, and the y-axis represents Time (s) ranging from 0 to 0.9. The graph compares IPEX-predicted, IPEX-observed, REX-predicted, and REX-observed lines.](image)
Conclusions

- All-to-all communication can be done in less than $p$ steps by allowing a certain amount of link contention to exist.
- The excess bandwidth on the links allows more than one message to travel on the same link in the same direction without increasing communication time.
- For small number of processors and small message sizes, the direct algorithms perform better than indirect algorithms.
- For large number or processors and large message sizes, IPEX performs the best because it reduces contention.
- As the communication bandwidth increases in future machines, link contention will be less of a problem, so direct algorithms can be expected to perform better than indirect algorithms.