Universität Augsburg University

Analysing Real-Time Behaviour of Collective Communication Patterns in MPI

Alexander Stegmeier, Martin Frieb,
Jörg Mische, Theo Ungerer

University of Augsburg, Germany

26th International Conference on
Real-Time Networks and Systems

11 October 2018

Motivation

- increase in performance needs for real-time applications
- multicore analysis with shared memory difficult

- apply manycores with
  - Network-on-Chip (NoC)
  - local memory per node
  - explicit message passing
- message passing interface (MPI)
  - standarad programming model

- special focus on collective communication
  - programming similar to Bulk Synchronous Parallel

Outline

Universität Augsburg University

Motivation

Basic Knowledge

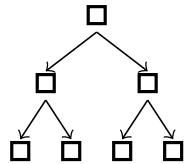
Analysis

Evaluation

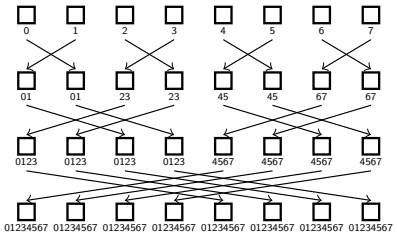
Conclusion

Communication Structure

- based on a central node (MPI_Bcast, MPI_Gather, ...)
  - communication along tree structures
  - investigated structures:
    - pipeline, chains, binary tree, binomial tree



- uniform data exchange (MPI_Allgather, MPI_Barrier, ...)
  - based on point-to-point communication
  - investigated structures:
    - ring, recursive doubling, neighbour exchange, bruck



Time-Division Multiplexing

- time-division multiplexing (TDM) for message scheduling
  - fixed time slots for sending
  - prevents conflicts between delivered flits
  - enables upper bounds for releasing and transporting flits

- WCTT for TDM:

$$WCTT = t_a + t_t$$

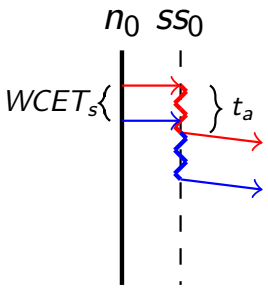
t_a: admission time
t_t: transportation time

Analysis flow

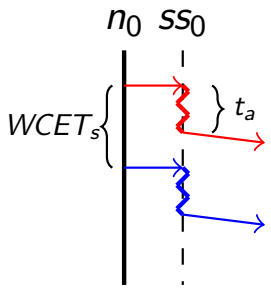
1. investigation of internal structure
   - separation of code execution and data transfer
   - send/receive operations as boundaries
2. analysis of components (WCET, WCTT)
3. combination regarding to communication pattern

Analysis issues

Boundary between WCET and WCTT



(a) send driven by t_a



(b) send driven by $WCET_s$

$$(f-1) \cdot max(WCET_s, t_a) + WCET_s + t_a$$

- similar for receive

Analysis issues

Dispatch along multiple nodes

- multiple options to accumulate times
  - identify longest path in terms of time

Analysis issues

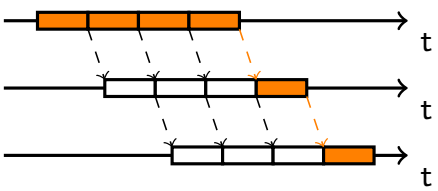
Dispatch along multiple nodes

- multiple options to accumulate times
  - identify longest path in terms of time
- three candidates for longest path

Analysis issues

Dispatch along multiple nodes

- multiple options to accumulate times
  - identify longest path in terms of time
- three candidates for longest path

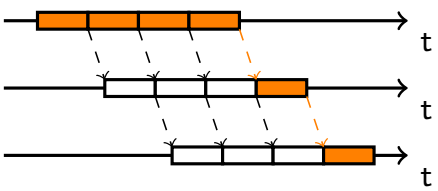


(a) send operation takes longest time

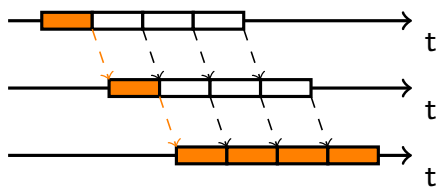
Analysis issues

Dispatch along multiple nodes

- multiple options to accumulate times
  - identify longest path in terms of time
- three candidates for longest path



(a) send operation takes longest time

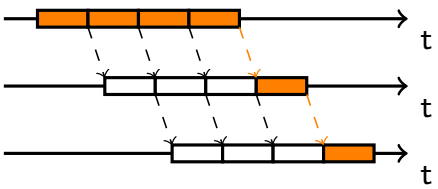


(b) receive operation takes longest time

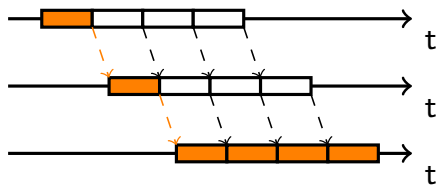
Analysis issues

Dispatch along multiple nodes

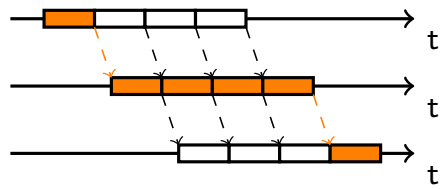
- multiple options to accumulate times
  - identify longest path in terms of time
- three candidates for longest path



(a) send operation takes longest time



(b) receive operation takes longest time

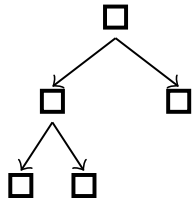


(c) receive and forward takes longest time

Analysis issues

Consideration of communication pattern

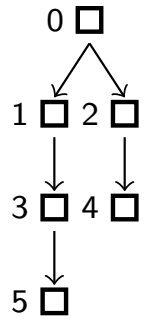
- treatment of tree structures
  - occurance of leaf at different tree levels
- sending procedure for nodes with multiple children
  - deepest sub tree first
- options for longest path regarding time
  - early forwarding + delivery along long sub tree
  - late forwarding + delivery along short sub tree



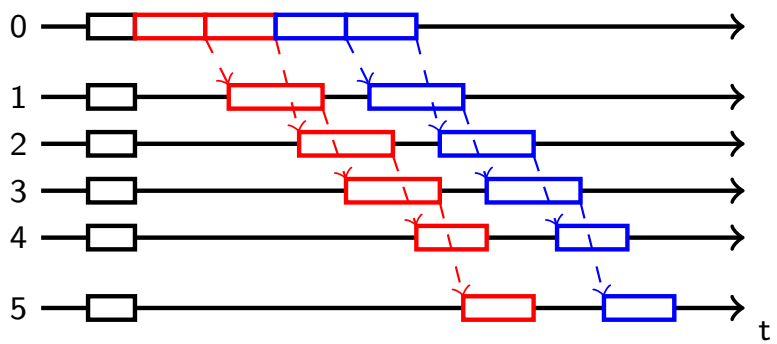
Applying the procedure

Illustration with example

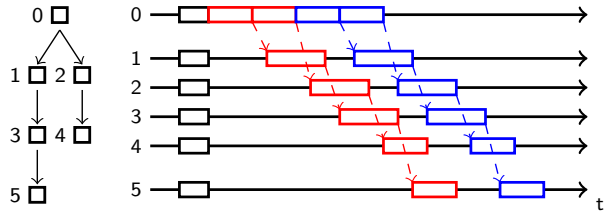
- broadcast to 5 nodes
  - message contains f flits
  - chain pattern with 2 chains



Communication details

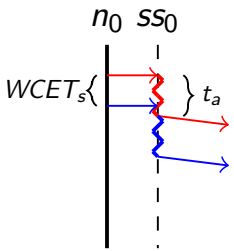


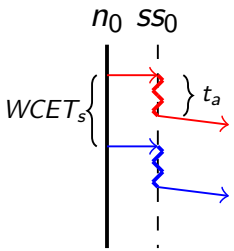
Applying the procedure



Boundaries between WCET/WCTT

issue:





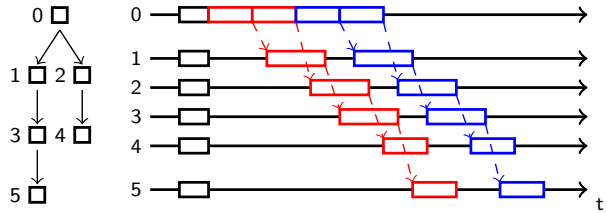
resulting timing:

$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \qquad (1)$$

$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \qquad (2)$$

Applying the procedure

Delivery along multiple nodes



$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

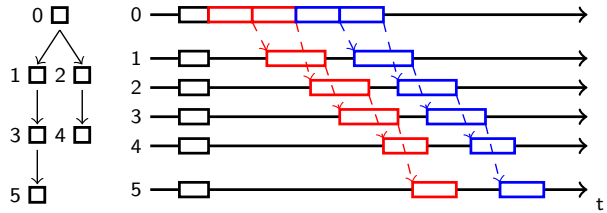
$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

- consideration of 1 flit

$$W_f = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (3)$$

Applying the procedure

Delivery along multiple nodes



$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

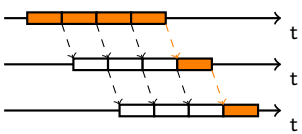
$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

- consideration of 1 flit

$$W_f = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (3)$$

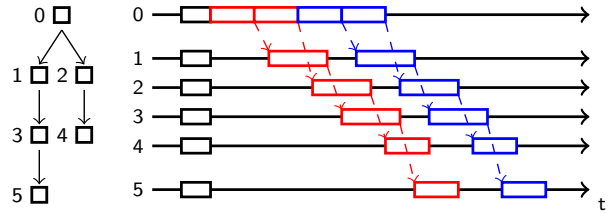
- consideration of f flits

$$W_a = \mathbf{f} \cdot W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (4)$$



Applying the procedure

Delivery along multiple nodes



$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

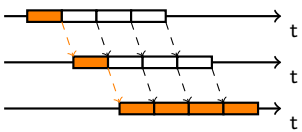
$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

- consideration of 1 flit

$$W_f = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (3)$$

- consideration of f flits

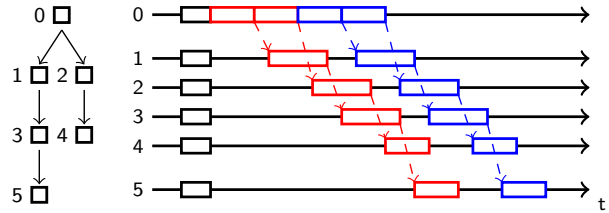
$$W_a = \mathbf{f} \cdot W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (4)$$



$$W_b = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + \mathbf{f} \cdot WCET_r \quad (5)$$

Applying the procedure

Delivery along multiple nodes



$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

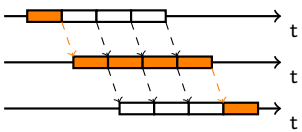
$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

- consideration of 1 flit

$$W_f = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (3)$$

- consideration of f flits

$$W_a = \mathbf{f} \cdot W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (4)$$

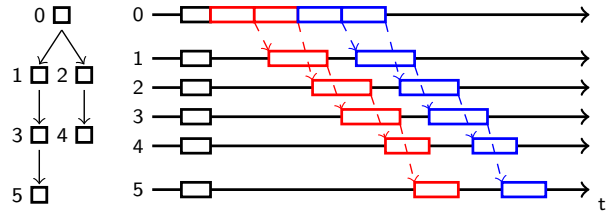


$$W_b = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + \mathbf{f} \cdot WCET_r \quad (5)$$

$$W_c = W_s + \mathbf{f} \cdot W_{sr} + (I-1) \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (6)$$

Applying the procedure

Delivery along multiple nodes



$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

- consideration of 1 flit

$$W_f = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (3)$$

- consideration of f flits

$$W_a = \mathbf{f} \cdot W_s + I \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (4)$$

$$W_b = W_s + I \cdot W_{sr} + (I+1) \cdot t_t + \mathbf{f} \cdot WCET_r \quad (5)$$

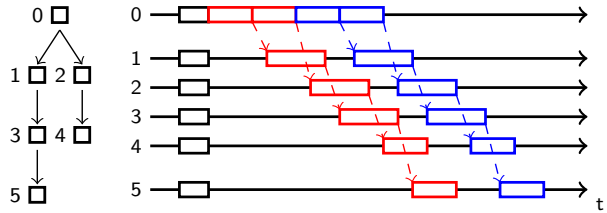
$$W_c = W_s + \mathbf{f} \cdot W_{sr} + (I-1) \cdot W_{sr} + (I+1) \cdot t_t + WCET_r \quad (6)$$

$$W_{chain} = max(W_a, W_b, W_c) \quad (7)$$

Applying the procedure

Consideration of communication pattern

- respect subtrees of different lengths
  1. long chain but early flit supply
  2. short chain but late flit supply



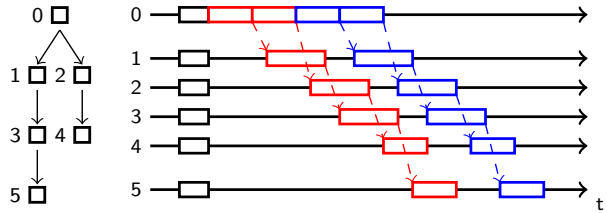
$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

Applying the procedure

Consideration of communication pattern

- respect subtrees of different lengths
  1. long chain but early flit supply
  2. short chain but late flit supply



$$W_s = (ch_i - 1) \cdot max(WCET_s, t_a) + WCET_s + t_a \quad (1)$$

$$W_{sr} = (ch_i - 1) \cdot max(WCET_{sr}, t_a) + WCET_{sr} + t_a \quad (2)$$

calculate overall timing: combine both cases

$$W_{total} = max(W_{chain}, W'_{chain})$$

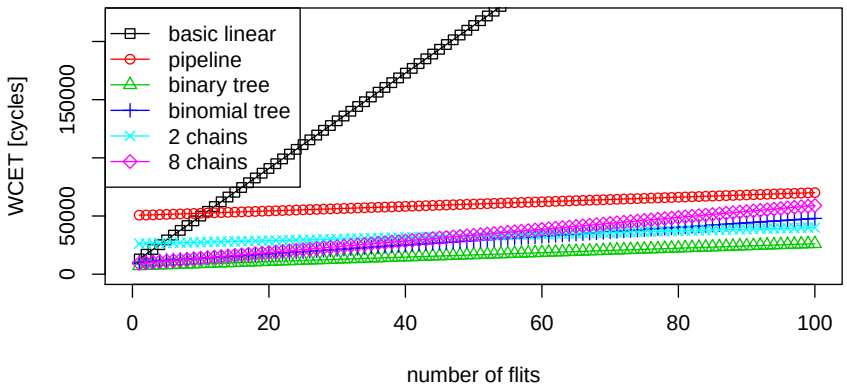
Evaluation

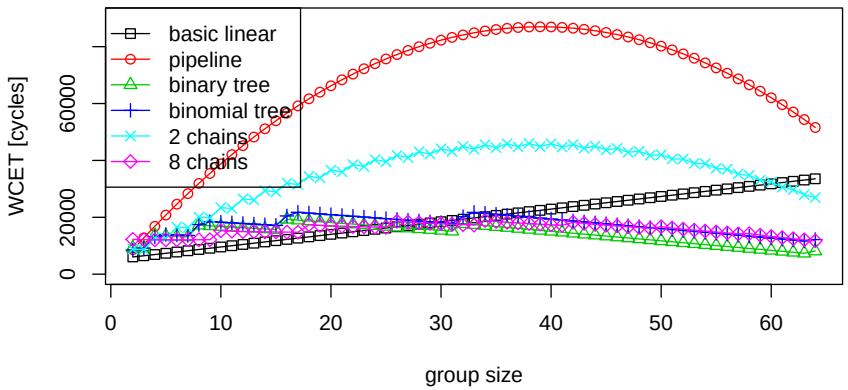
Assumptions

- platform: RC/MC manycore
  - NoC topology: uni-directional quadratic torus
  - 64 ARM-V7 cores
  - local scratchpad memory for each core
- MPI collectives ported from OpenMPI
  - synchronization done in software
- representation for each communication structure
  - MPI_Bcast for tree structures
  - MPI_Allgather for uniform data exchange
- OTAWA for calculation of core local WCET bounds

Communication based on a central node

MPI_Bcast





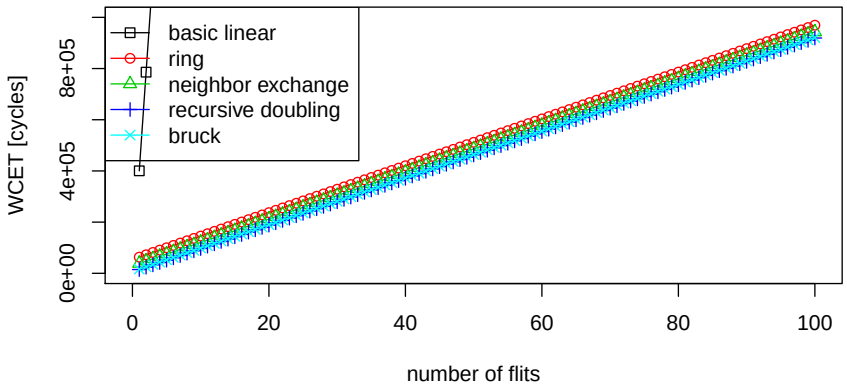
- linear growth with respect to flits
- best performance:
  - binary tree

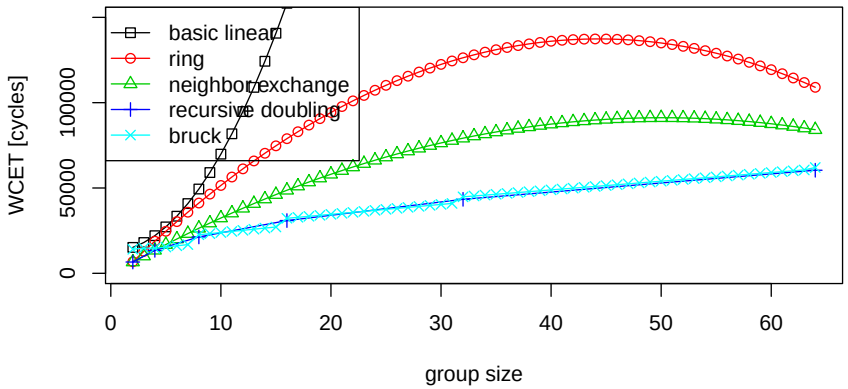
- influenced by software synchronization
- best performance:
  - basic linear (small groups)
  - binary tree (otherwise)

Communication based on uniform data exchange

Universität Augsburg University

MPI_Allgather





- only marginal differences (except basic linear)
- best performance:
  - bruck
  - recursive doubling

- significant differences
- best performance:
  - bruck
  - recursive doubling

Summary and Conclusion

Summary

- described timing analysis of collective communication
- focus on combination of code WCET bounds and WCTT
- evaluation on concrete platform
  - MPI collectives as representatives
  - comparison of communication patterns

Results

- high impact of communication patterns
- recommended communication patterns
  - binary tree
  - bruck

Thank you for your attention.

Data transfer

- based on flits (equally sized atomic data unit)
- send operation: put flit to send buffer
- flits in send buffer:
  - ejected to a appropriate slot in the NoC
- flits at target: store to receive buffer
- receive operation: handle flit from receive buffer