A Congestion Avoidance Technique Using Packet Pacing toward Exascale Interconnect

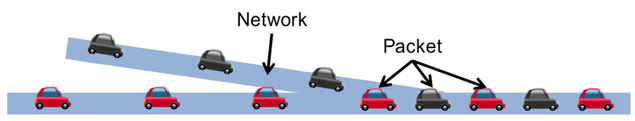
"Packet Pacing" makes Your Communication Fast !

Introduction

- **Reducing communication latency is still important in HPC**
  - **Some collective comm. may not be able to use in the near future**
    - Time-consuming comm. is caused by *network congestion* in heavy traffic.
  - **Packet pacing avoids the critical network congestion**

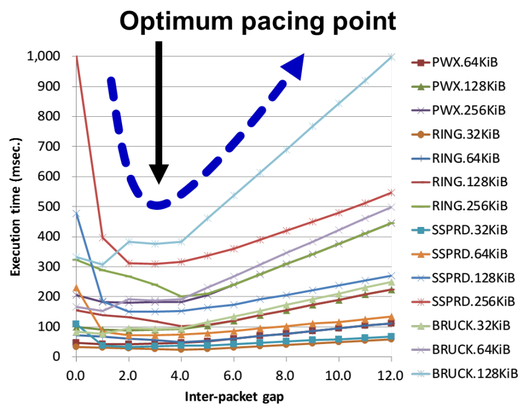
What is Packet Pacing?

- **Insertion of non-sending period (inter-packet gap) between sending packets**
  - **Controls packet injection rate by interleaving packets**
    - Gives an optimum inter-packet gap for each message.
  - **Reduces network congestion**
    - Decreases stop & go latency.
  - **Maximizes throughput & minimizes network latency**
    - Applications with heavy traffic comm. get faster.



Keep space for smooth traffic

  - **Inter-packet gap**
    - An interval between sending packets.
    - Gap = 0: No packet pacing.
      Gap = *N*: Transfer time for *N* packets.



Execution times for each inter-packet gap
(3D Torus 512nodes)

☑ Each comm. has an optimum pacing point

NSIM: Our Interconnect Simulator

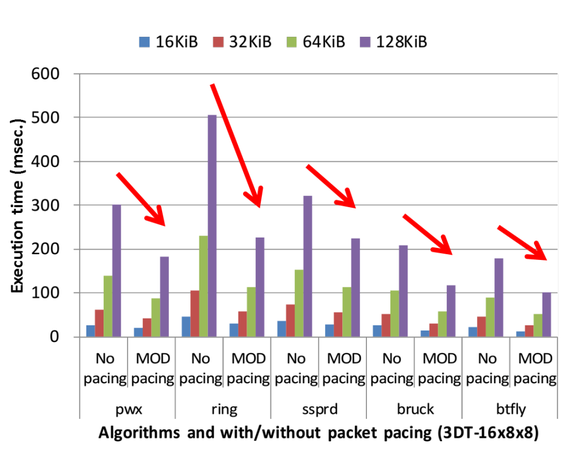
- **Support rapid performance evaluation and communication analysis toward exascale interconnect**
  - **Feature**
    - *On-demand comm. pattern (message) generation* using MPI-like C program (MGEN program).
    - *Accurate simulation* with very fine network specifications and various MPI overheads (Interconnect Configuration File).
    - Wide variety of simulation outputs.
    - Simulation with behavior of OS jitter/load imbalance.
  - **Supported topologies**
    - Tofu (K computer, FX10), ~6D Mesh/Torus, FatTree.
  - **Very fast simulation**
    - *18-min.* for a 1MiB random ring comm. (HPCC) on *128K-node 3D-Torus* by using 128-core real system.
  - **Design and Implementation**
    - As an MPI program based on PDES(Parallel Discrete Event Simulation).
    - By Kyushu Univ., Fujitsu, and ISIT since 2009.

MOD Packet Pacing: A Pacing Strategy

- **Message Overlap Degree packet pacing**
  - **Gap is calculated based on the degree of message overlap**
  - **Gap = (Comm. Hop) – 1**
    - Each link is shared by messages of the number of hop count at most in alltoall.

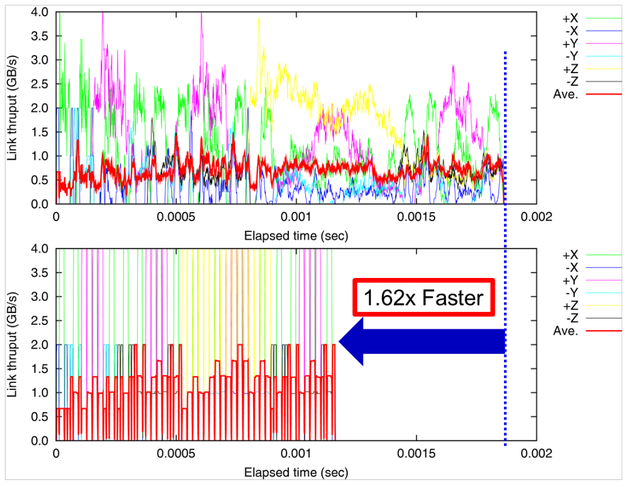
- Alltoall algorithm: PWX(Pairwise exchange), RING, SSPRD(Simple spread), BRUCK, BTFLY(Butterfly), and A2AT(by Ishihata et al. at Tokyo Univ. of Tech.).
- Link bandwidth: 4GB/s. Routing algorithm: DOR(Dimension Ordered Routing).

Experimental Results on Simulation (NSIM)



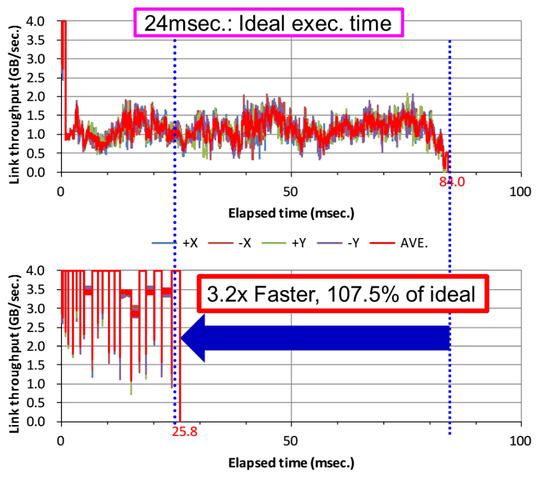
Execution times with/without MOD pacing
(3D Torus 16x8x8nodes)

☑ Faster comm. by MOD pacing



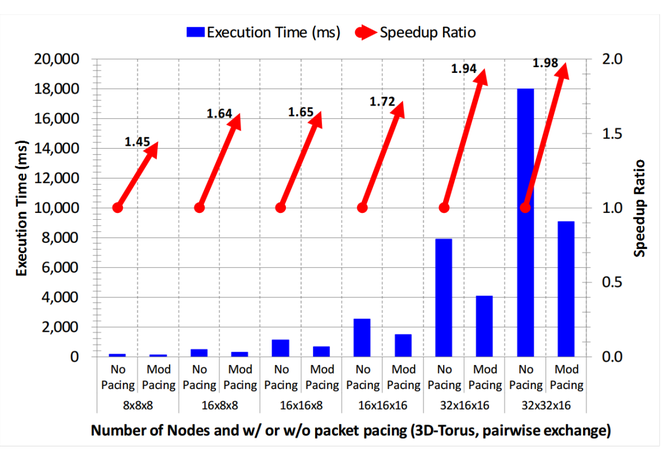
Link throughputs of an alltoall communication
(3D Torus 4x4x4nodes, PWX)

☑ Pacing effect by MOD pacing (1)



Link throughputs of an alltoall communication
(2DT-9x9nodes, A2AT)

☑ Pacing effect by MOD pacing (2)

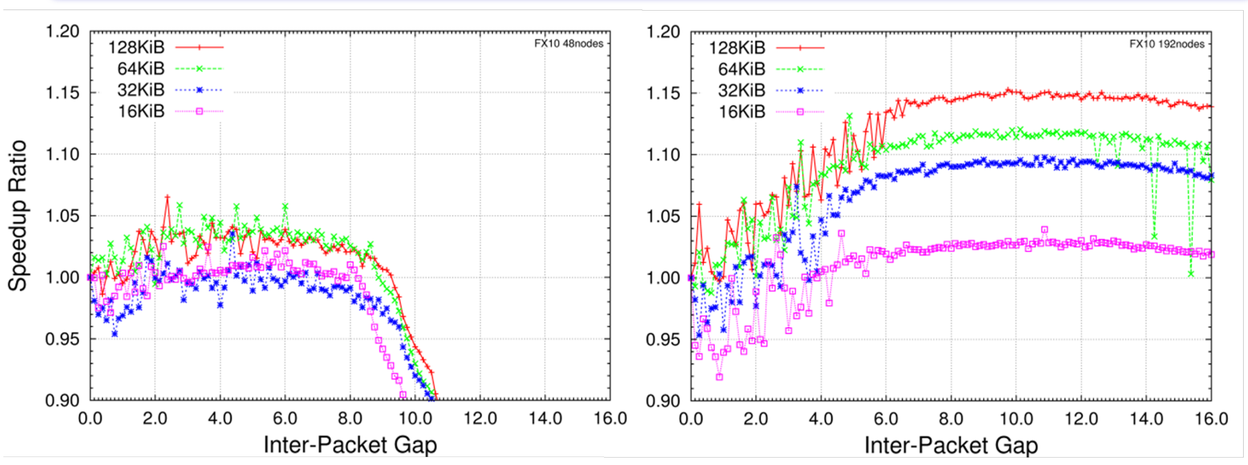


Execution time and speedup ratio on various node sizes
(3D Torus, PWX)

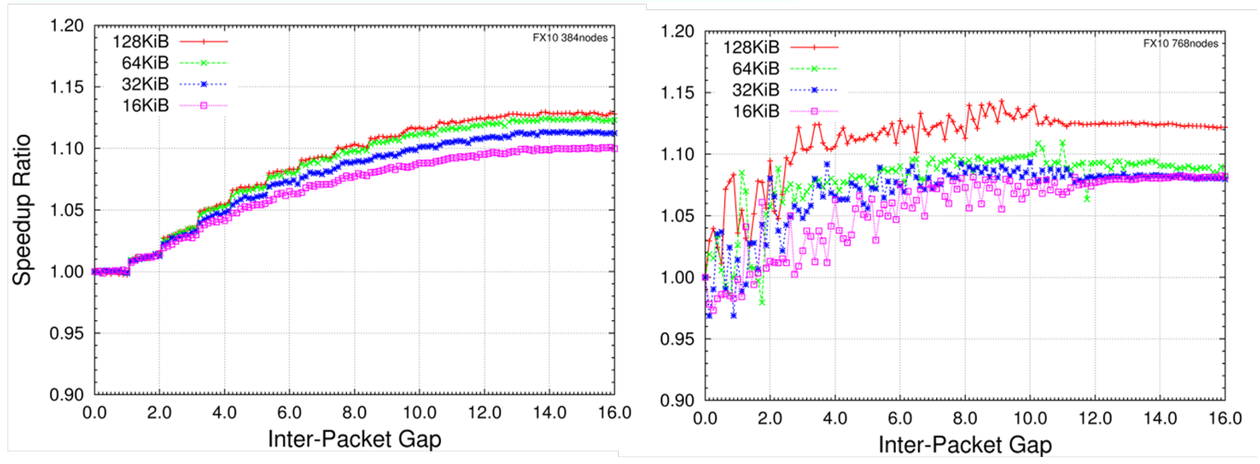
☑ Speedup ratio increases according to node size

Experimental Results on Real Machine (FX10)

- Random ring communications (from HPCC) with packet pacing.
- Measured on Fujitsu FX10 at Kyushu Univ., JAPAN.



48 nodes (768 procs.) 192 nodes (3,072 procs.)



384 nodes (6,144 procs.) 768 nodes (12,288 procs.)

☑ Optimum pacing point and effect of packet pacing grows as node size increases

Summary

Packet pacing : controls packet injection *aggressively* to accelerate HPC comm.

1. Each heavy traffic communication has its own **optimum pacing point**.
2. The **MOD pacing strategy** finds the best pacing point.
3. The **Effectiveness improves** by message size and/or node size.

Hidetomo Shibamura <shibamura@isit.or.jp>

Institute of Systems, Information Technologies and Nanotechnologies

Japan Science and Technology Agency, CREST