Zero Copy MPI Derived Datatype Communication Over InfiniBand

Gopalakrishnan Santhanaraman
Jiesheng Wu
D.K.Panda

Network Based Computing Lab
The Ohio State University

Presentation Layout

- Introduction
- Background and Existing approaches
- Motivation for new Scatter/Gather (SGRS) approach
- Design and implementation issues
- Performance Evaluation
- Conclusions and Future work

Introduction

- Non-contiguous data communication is common in scientific applications.
    - Decomposition of multi dimensional volumes, FFT, finite element codes
    - NAS BENCHMARKS, LINPACK
- MPI provides derived datatype interface to facilitate this kind of data movement
- Current Implementations of derived datatypes not very efficient

Presentation Layout

- Introduction
- Background and Existing Approaches
- Motivation for new Scatter/Gather(SGRS) approach
- Design and Implementation Issues
- Performance Evaluation
- Conclusions

Related Work

- Improve datatype processing
- Optimized packing and Unpacking Procedures
- Taking advantage of network features to improve non contiguous datatype communication

InfiniBand Overview

- Emerging interconnect based on Open standards
- Provides low latency and high Bandwidth
- Several Novel features
  - RDMA
  - Scatter/Gather
  - Atomic operations
- VAPI – low level interface (API) over InfiniBand

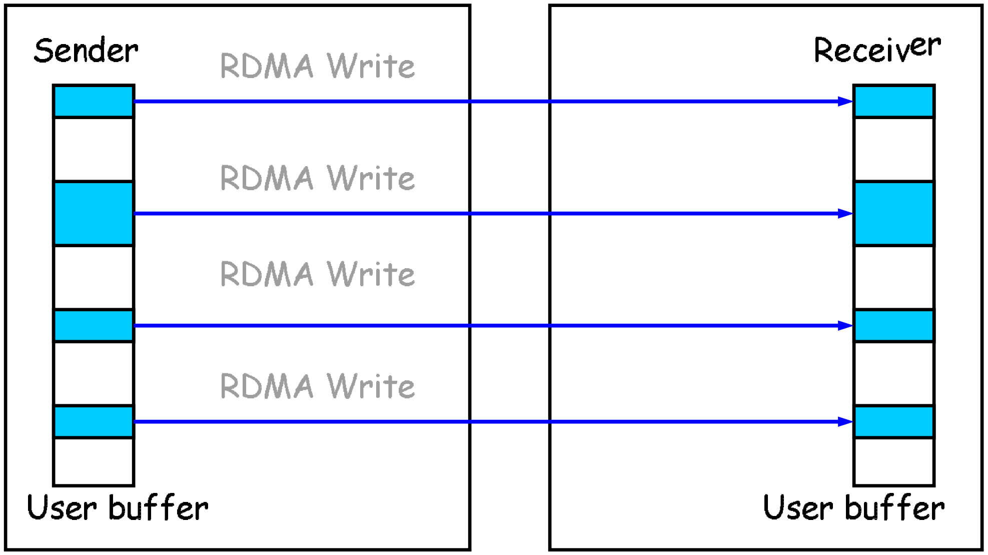
Our Previous Work

- Different Approaches
  - Pack/Unpack Based Approach
    - Copy on both sides
    - Pipeline packing, network communication and unpacking
  - Reduced Copy
    - RDMA write with Gather on sender side
    - RDMA read with Scatter on receiver side
  - Zero Copy
    - Multiple RDMA writes on sender side (Multi-W scheme)

Jiesheng Wu, Pete Wyckoff, and Dhabaleswar K. Panda. High Performance Implementation of MPI Datatype Communication over InfiniBand. In Int'l Parallel and Distributed Processing Symposium (IPDPS 04), April, 2004

Conclusions of Previous Work

- For small messages with eager protocol, segment pack/unpack is best.
- For messages in rendezvous protocol range, zero copy schemes are beneficial.
  - Multi-W zero copy scheme was proposed.



Limitations of Earlier Approaches

- RDMA write/gather, RDMA read/scatter
  - □ Needs copy in order to handle non-contiguity on both sides
- Multi-W
  - □ For large number of small segments, performance degrades.
    - Overhead of large number of RDMA operations
    - Poor network utilization
- Motivation to explore other zero copy schemes

- Problem statement

  How can we utilize the advanced features provided by modern interconnects like InfiniBand to handle non-contiguous data communication efficiently and overcome the above limitations?

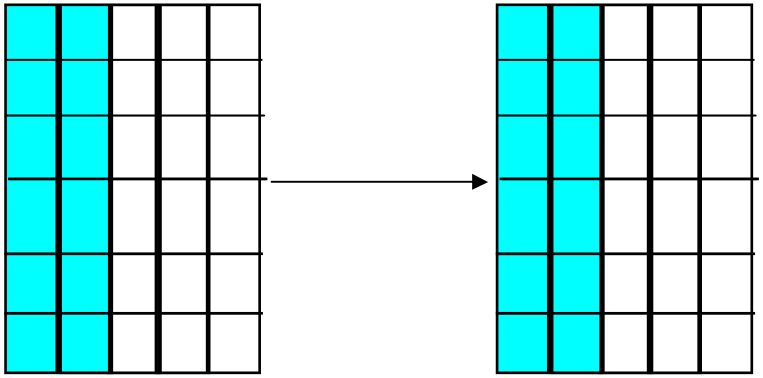
Presentation Layout

- Introduction
- Background and Existing approaches
- Motivation for New Scatter/Gather (SGRS) Approach
- Design and Implementation issues
- Performance Evaluation
- Conclusions and Future work

Semantics of send/gather, receive/scatter feature

- Based on send/receive channel semantics
- Handles non-contiguity on both send/receive sides which is the most generic case
- To implement datatype using this feature needs a synchronization phase. Hence applicable for messages which fall under the rendezvous protocol

VAPI level Comparison
Multi-W vs SGRS



Observations

- For a fixed number of segments SGRS approach outperforms the Multi-W approach for different message sizes
- For a fixed message size with increasing degree of non-contiguity,
  - SGRS scheme degradation is negligible
  - Multi-W degradation is significant

Presentation Layout

- Introduction
- Background and Existing approaches
- Motivation for new Scatter/Gather approach
- Design and Implementation issues
- Performance Evaluation
- Conclusions

MVAPICH Overview

- High Performance Implementation of MPI over InfiniBand
- Design based on MPICH and MVICH
  - Eager protocol for small messages
  - Rendezvous protocol for large messages
- Datatype Implementation currently uses the generic packing and unpacking scheme.
  - small datatype messages are packed/unpacked
  - large datatype messages both sides allocate pack/ unpack buffers dynamically

MVAPICH Software Distribution

- Open Source (current version is 0.9.4 released last week)
- Have been directly downloaded by more than 119 organizations and industry
- Available in the software stack distributions of IBA vendors

National Labs/Research Centers

Argonne National Laboratory
Cornell Theory Center
Center for Mathematics and Computer Science (The Netherlands)
Inst. for Experimental Physics (Germany)
Inst. for Program Structures and Data Organization (Germany)
Lawrence Berkeley National Laboratory
Los Alamos National Laboratory
Max Planck Institute for Astronomy (Germany)
NASA Ames Research Center
NCSA
National Center for Atmospheric Research
Ohio Supercomputer Center
Pacific Northwest National Laboratory
Pittsburgh Supercomputing Center
Research & Development Institute Kvant (Russia)
Science Applications International Corporation
Sandia National Laboratory

Universities

Georgia Tech
Indiana University
Korea Univ. (Korea)
Korea Inst. Of Science and Tech. (Korea)
Kyushu Univ. (Japan)
Mississippi State University
Moscow State University (Russia)
Northeastern University
Penn State University
Russian Academy of Sciences (Russia)
Stanford University
Technion (Israel)
Technical Univ. of Munchen (Germany)
Technical Univ. of Chemnitz (Germany)
Univ. of Geneva (Switzerland)
Univ. of Houston
Univ. of Karlsruhe (Germany)
Univ. of Massachusetts Lowell
Univ. of Paderborn (Germany)
Univ. of Potsdam (Germany)
Univ. of Rio Grande (Brazil)
Univ. of Sherbrooke (Canada)
Univ. of Stuttgart (Germany)
Univ. of Toronto (Canada)

MVAPICH Users (Cont'd)

Industry

Abba Technology
Advanced Clustering Tech.
AMD
Ammasso
Appro
Array Systems Comp. (Canada)
Atipa Technologies
Agilent Technologies
Clustars Supercomputing-Technology Inc. (China)
Clustervision (Netherlands)
Compusys (UK)
CSS Laboratories, Inc.
Dell
Delta Computer (Germany)
Emplics (Germany)
Fluent Inc.
ExaNet (Israel)
GraphStream, Inc.
HP
HP (France)

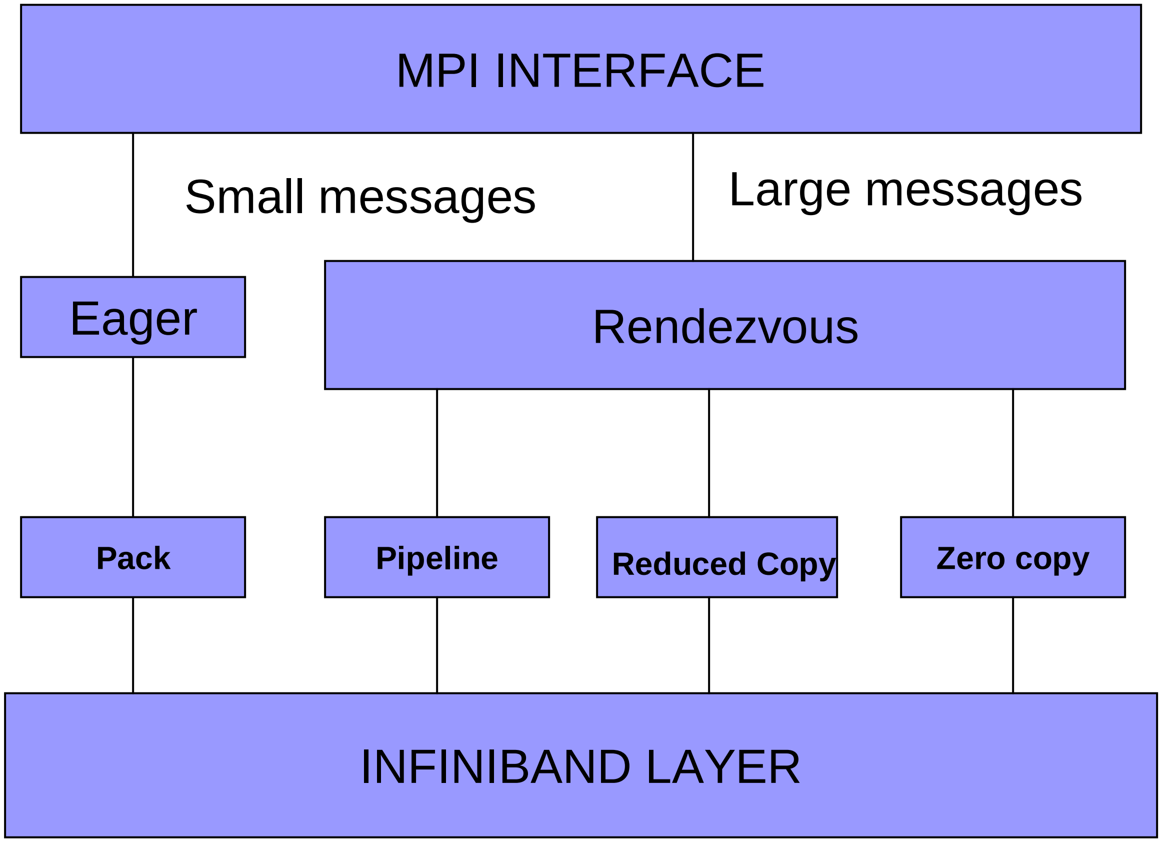
IBM
IBM (France)
IBM (Germany)
INTERSED (France)
InfiniCon
Intel
Intel (China)
Intel (Germany)
Intel Solution Services (Hong Kong)
Intel Solution Services (Japan)
JNI
Kraftway (Russia)
Langchao (China)
Linux Networx
Linvision (Netherlands)
Megaware (Germany)
Mercury Computer Systems
Mellanox Technologies
Meiosys (France)
Microway, Inc.
NEC (Japan)
NEC Solutions, Inc.
NEC (Singapore)
NICEVT (Russia)
OCF plc (United Kingdom)

OctigaBay (Canada)
PANTA Systems
ParTec (Germany)
PathScale, Inc.
Pultec (Japan)
Pyramid Computer (Germany)
Qlusters (Israel)
Raytheon Inc.
RLX Technologies
Rosta Ltd. (Russia)
SBC Technologies, Inc.
Scyld Software
SGI (Silicon Graphics, Inc.)
SKY Computers
Streamline Computing (UK)
Systran
Tomen
Telcordia Applied Research
Thales Underwater Systems (UK)
Transtec (Germany)
T-Platforms (Russia)
Topspin
Unisys
Voltaire
WorkstationsUK, Ltd. (UK)
Woven Systems, Inc.

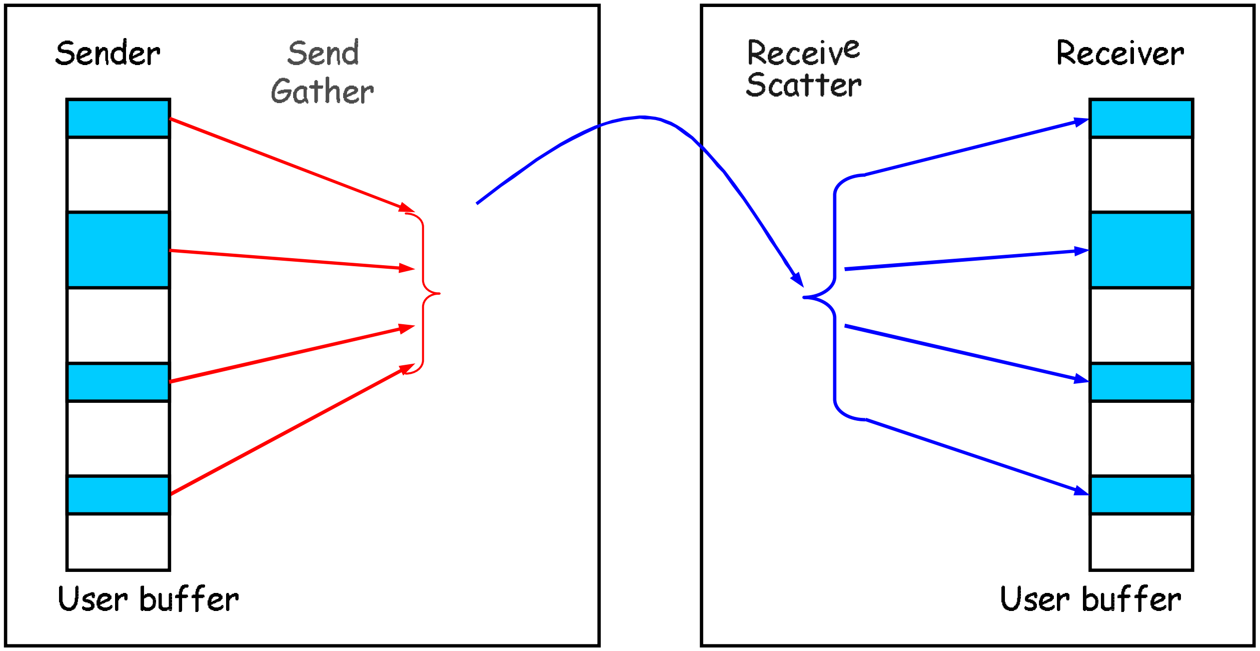
Larger IBA Clusters using MVAPICH and Top500 Rankings

- **1105-node cluster at Virginia Tech**
  - □ **3rd in Nov. ’03 ranking**
- **192-node cluster at Mississippi State University**
  - □ **150th in June ’04 ranking**
- **128-node cluster at Sandia/Livermore**
  - □ **111th in Nov ’03 ranking and 211th in June ’04 ranking**
- **256-node cluster at Los Alamos**
  - □ **116th in Nov ’03 ranking and 218th in June ’04 ranking**
- **128-node cluster at Ohio Supercomputer Center (OSC)**
  - □ **272th in June ’04 ranking**
- **More are getting installed ….**

Framework For Handling Datatypes



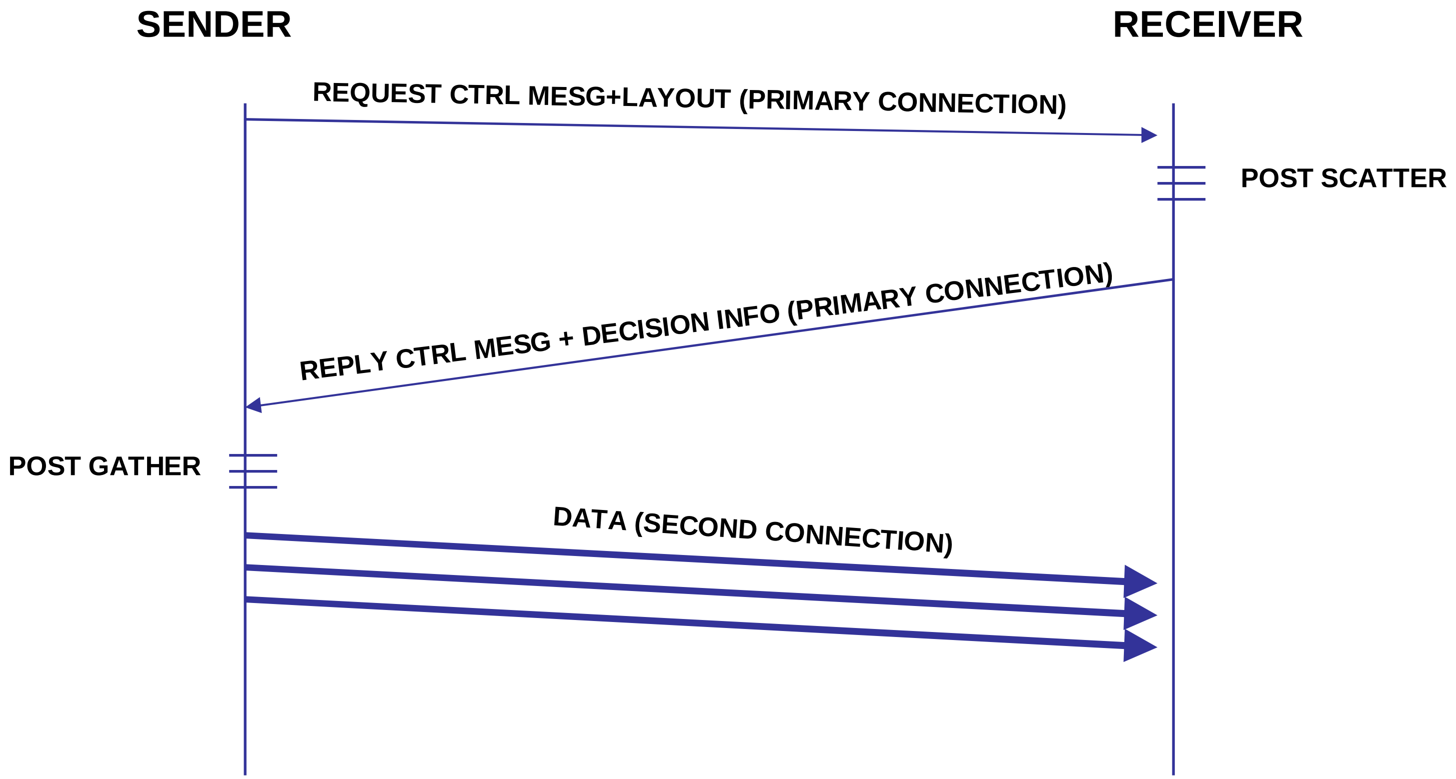
Basic Idea


Sender
Send
Gather
User buffer
Receive
Scatter
Receiver
User buffer

Design Issues

- Exchanging layout information
  - MPI datatype has only local semantics
  - Optimizing layout exchange
- Layout matching decision needs to be conveyed
- Registration and deregistration on user datatype message buffers
  - Unique issue due to non-contiguity in buffers
- Posting Descriptors
  - Upper limit on number of scatter gather descriptors.
  - Needs a secondary connection for transmitting non-contiguous data

SGRS COMMUNICATION PROTOCOL



SENDER

RECEIVER

REQUEST CTRL MESG+LAYOUT (PRIMARY CONNECTION)

POST SCATTER

REPLY CTRL MESG + DECISION INFO (PRIMARY CONNECTION)

POST GATHER

DATA (SECOND CONNECTION)

Layout Exchange and Matching Decision

- Take advantage of handshake messages in the rendezvous protocol to achieve this
  - Sender’s datatype layout is appended to Rendezvous start control message
  - The matching decision information is conveyed in the Rendezvous reply/clear to send message

- A layout cache mechanism is implemented to reduce overhead of layout transfer
  - Datatype information is exchanged only once
  - Only the index needs to be sent for future messages
    - *Datatype Cache mechanism proposed by Traff et al.*

Registration

- Registration and Deregistration on user datatype message buffers
  - Common issues in both the zero copy schemes
  - Unique issue due to non-contiguity in buffers
  - Use Optimistic Group Registration scheme

J. Wu, P. Wyckoff, and D. K. Panda. “Supporting Efficient Noncontiguous Access in PVFS over InfiniBand”. IEEE Cluster Computing 2003, Dec. 2003

Posting Descriptors

- Needs a separate Queue pair connection
  - Ordering
  - Scalability
- Upper limit on number of gather/scatter descriptor
  - Message might need to be chopped into multiple gather/scatter descriptors
  - Number of posted gather descriptors must be equal to the number of posted scatter
  - Needs a negotiation phase

Presentation Layout

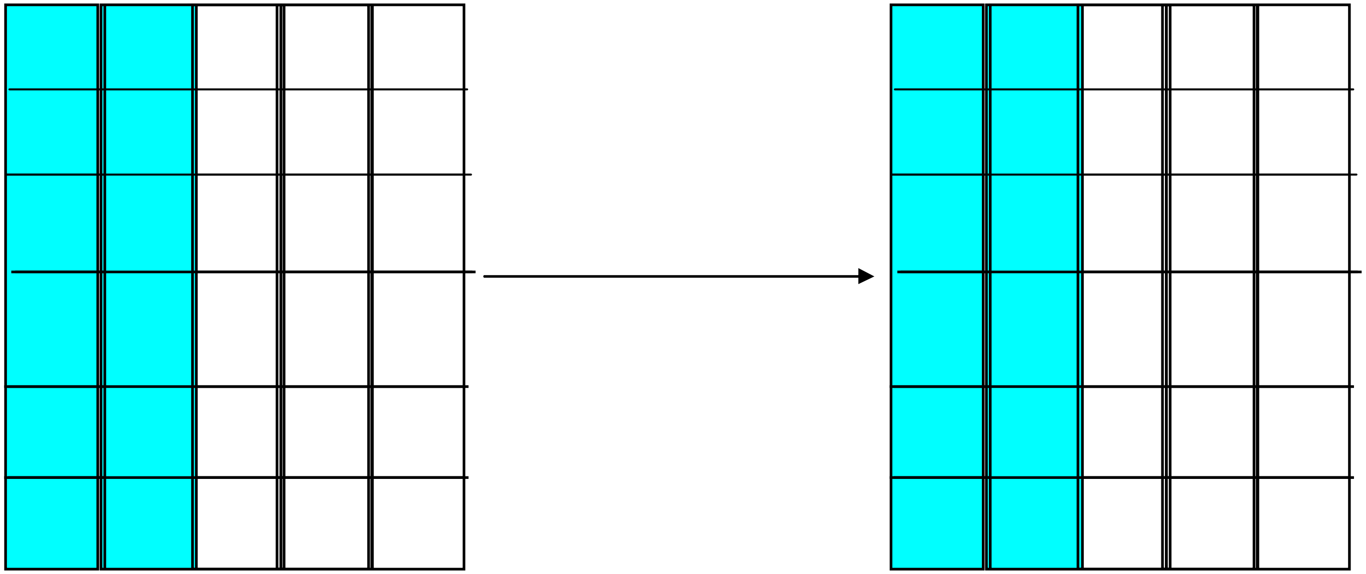
- Introduction
- Background and Existing approaches
- Motivation for new Scatter/Gather (SGRS) approach
- Design and Implementation issues
- Performance Evaluation
- Conclusions and Future work

Experimental Evaluation

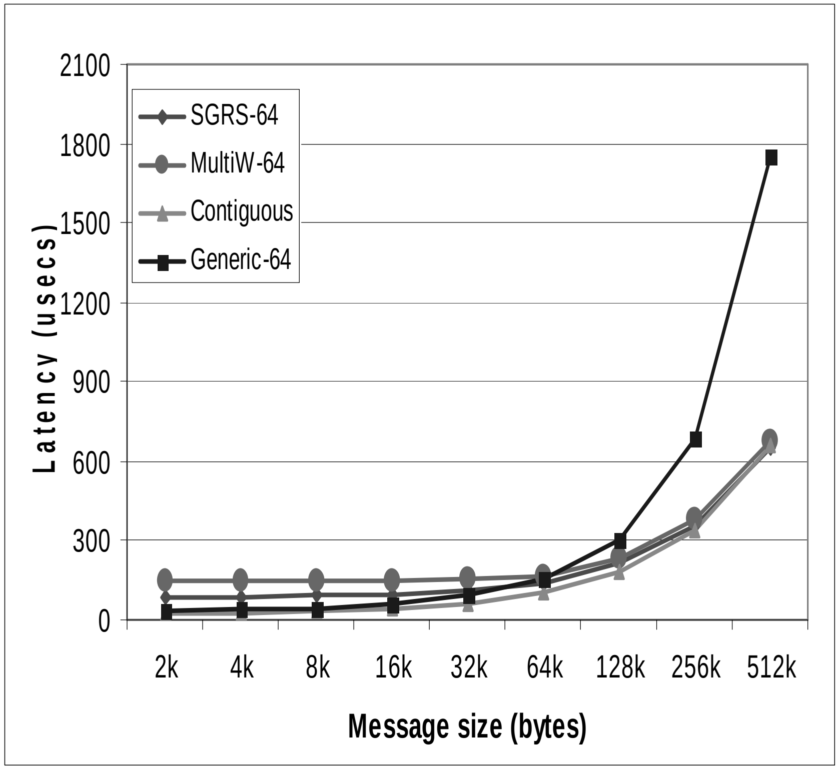
- Experimental Test bed
  - Cluster of 8 Supermicro nodes
    - Dual Xeon 3.0 GHz processors
    - 512 KB L2 Cache, PCI-X 64bit 133 MHz bus
    - InfiniHost SDK version 3.0.1
    - Physical memory 1GB DDR-SDRAM memory
- Experiments conducted
  - Latency, Bandwidth with vector datatype
  - Collective latency (MPI_Alltoall)
  - CPU overhead tests
  - Impact of layout cache

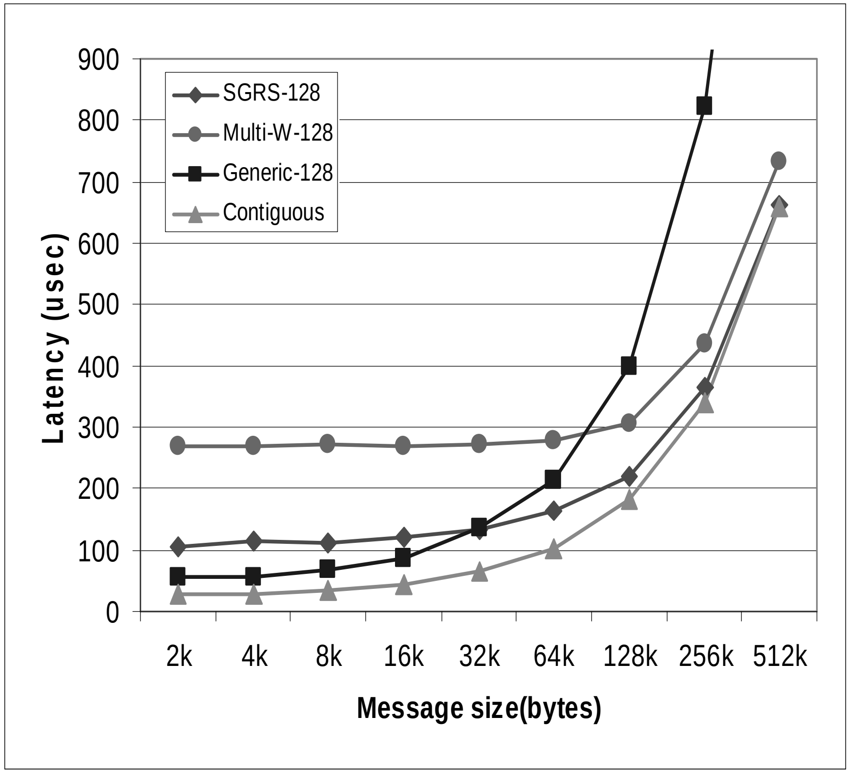
Vector Datatype Test

A vector (multiple columns in a 64x4096 integer array) test



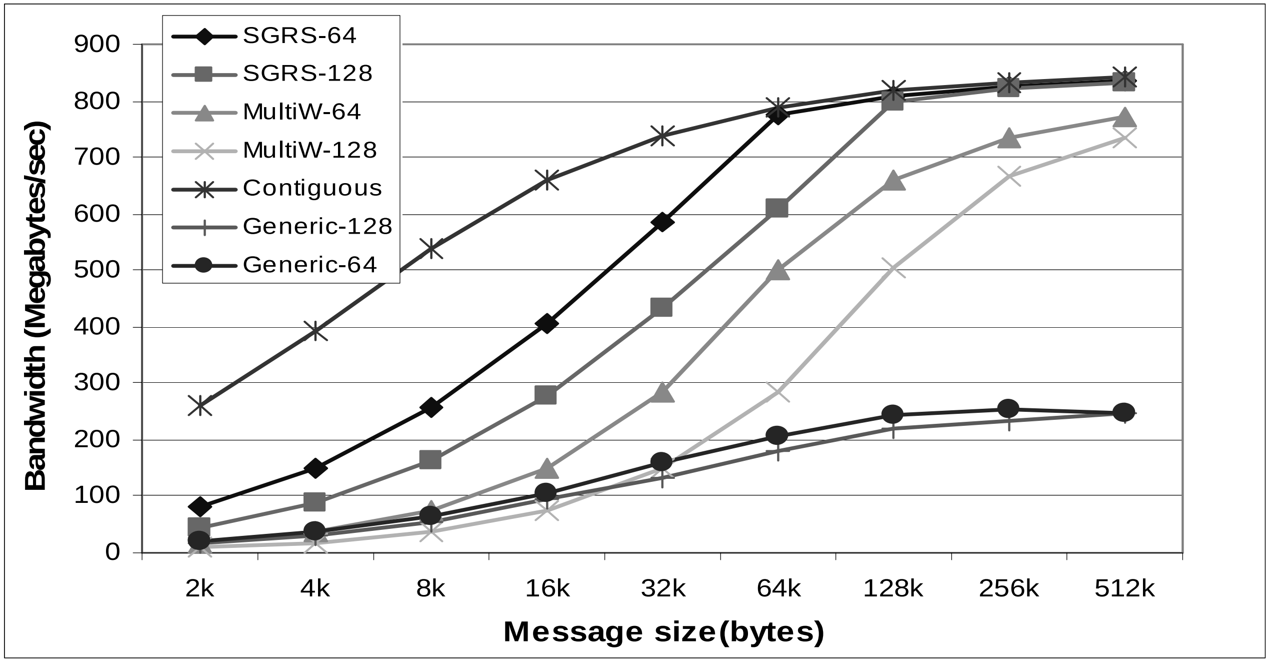
MPI Level Vector Latency





- SGRS scheme reduces latency by up to 62% as compared to Multi-W

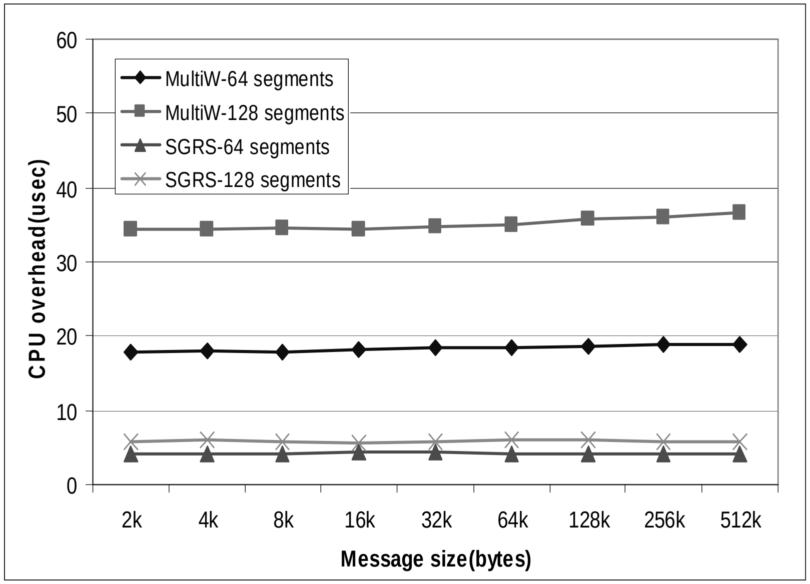
MPI Level Vector Bandwidth



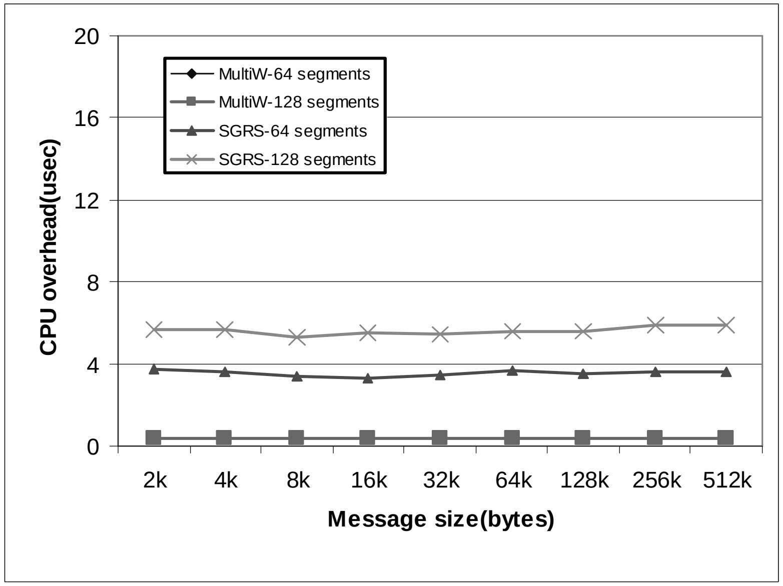
- SGRS scheme gives the best performance
- For large messages we get Bandwidth close to that of contiguous Bandwidth

CPU Overhead

Sender side Overhead

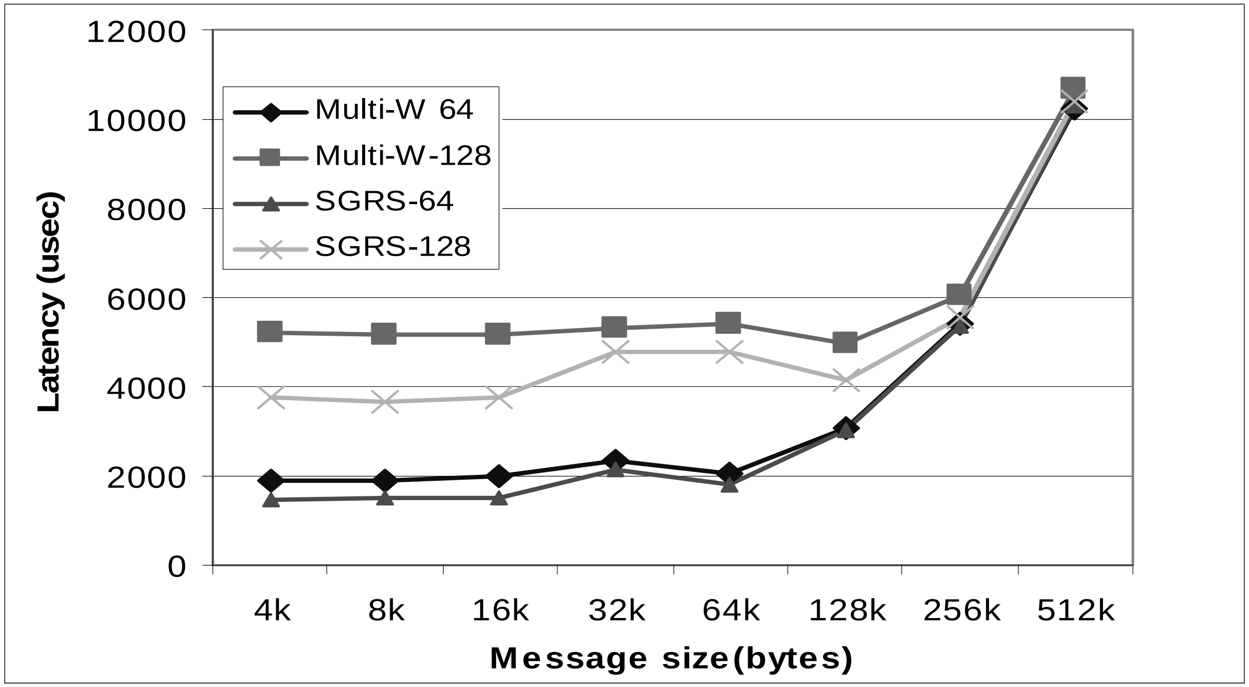


Receiver side Overhead



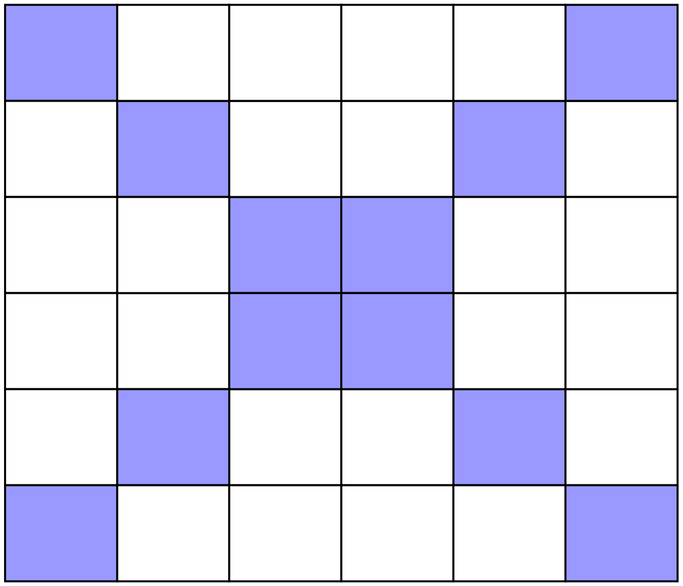
- The CPU overhead associated with SGRS protocol is relatively low

MPI_Alltoall Latency



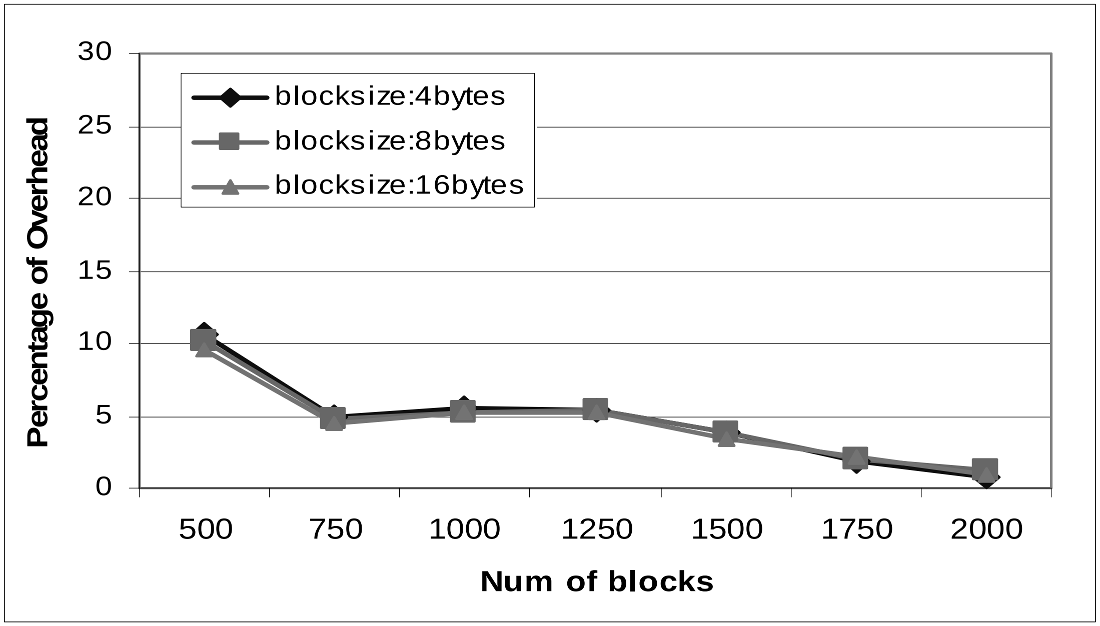
- The Alltoall latency test shows significant improvement for the SGRS approach

Synthetic Benchmark to Measure Impact of Layout Caching



- Need to transfer the two diagonals of a square matrix.
- Diagonal elements are actually blocks.
- Need significant layout size to describe it

Effect of Layout Cache



- Layout cache shows benefits for certain scenarios
- Layout itself is contiguous as compared to the data that it describes

Presentation Layout

- Introduction
- Background and Existing approaches
- Motivation for new Scatter/Gather (SGRS) approach
- Design and implementation issues
- Performance Evaluation
- Conclusions and Future work

Conclusions and Future Work

- Provided a new zero-copy scheme for datatype communication over InfiniBand
- The new scheme outperforms the existing schemes
  - Latency can be improved by up to 62%
  - Bandwidth can be increased by up to 400%
  - Collective communication like Alltoall can derive potential benefits
  - Layout cache is shown to be beneficial for some scenarios

- Future Work
  - Evaluate the effectiveness of this scheme at application level
  - Provide a comprehensive solution that internally uses multiple schemes to achieve best performance

Thank You!

For more information, please visit the

NBC Home Page

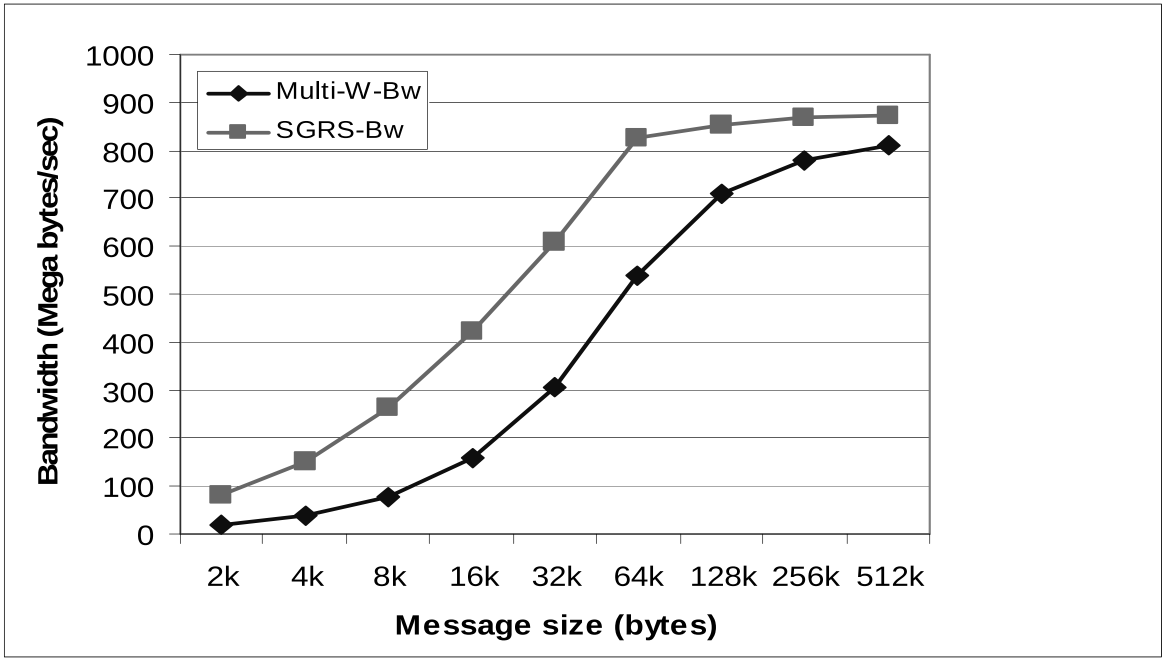
http://nowlab.cis.ohio-state.edu

Network Based Computing Laboratory

The Ohio State University

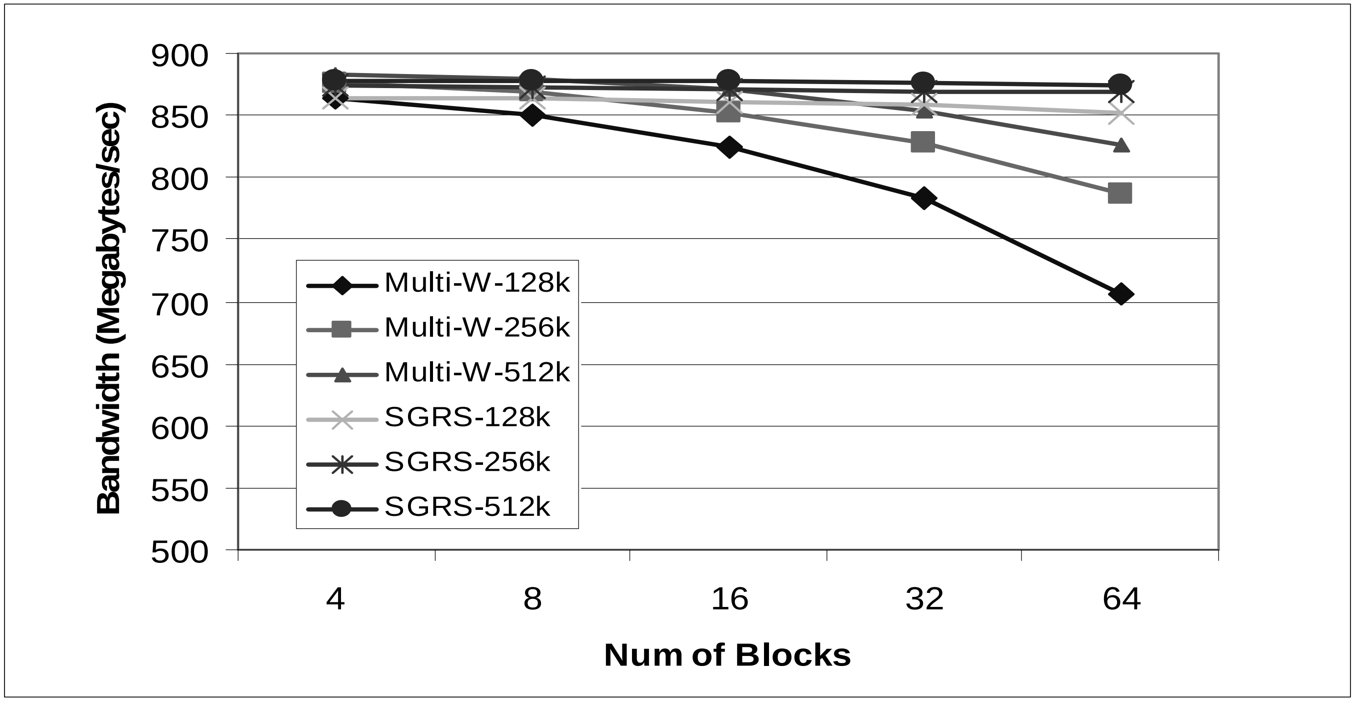
BACKUP SLIDES

Vapi Level Bandwidth Comparison SGRS vs. Multi-W



- SGRS scheme consistently outperforms the Multi-W

Effect of degree of non-contiguity



- SGRS scheme fares better with increased non-contiguity