ch_hyperbone: A Device for MPI Programs Executions Using a Virtual Hypercube

Samuel L. V. Mello [1], Rafael Caiuta [1], Luis C. E. Bona [2],
Keiko V. O. Fonseca [2], Elias P. Duarte Jr [1]

Universidade Federal do Paraná
Departamento de Informática Caixa Postal 19018 Curitiba PR
{slucas,caiuta,elias}@inf.ufpr.br

Universidade Tecnológica Federal do Paraná
CPGEI Av. Sete de Setembro 3165 Curitiba PR
{bona,keiko}@cpgei.cefetpr.br

2006

Summary

- Distributed and Parallel Processing
- HyperBone
- MPI Standard
- ch_hyperbone
- Experimental Validation
- Conclusion

Introduction

- Motivation: Distributed and parallel processing on the Internet
- Environment subject to failures
- Failures in a single node can invalidate the whole computation

Introduction

- On the Internet, some nodes fail with more frequency than others
- A monitoring system helps to choose the most reliable nodes for running the program, increasing the probability of success

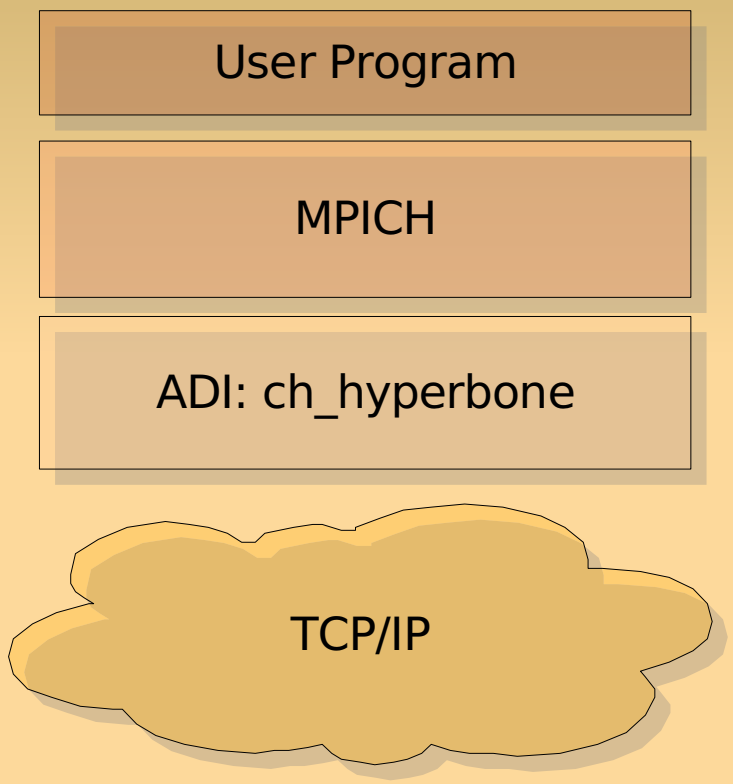
Introduction

- HyperBone is a scalable overlay network which provides monitoring and fault-tolerant routing using a virtual hypercube topology
- This presentation shows a software device that allows running MPI parallel programs over HyperBone

Introduction

- MPI (*Message Passing Interface*) is a standard for building parallel and distributed applications
- The MPICH library is an open source implementation of the MPI standard
- ADI (*Abstract Device Interface*) allows MPICH to use different transport systems to perform message delivery
- *ch_hyperbone* is an ADI device that uses HyperBone to perform message delivery

ch_hyperbone



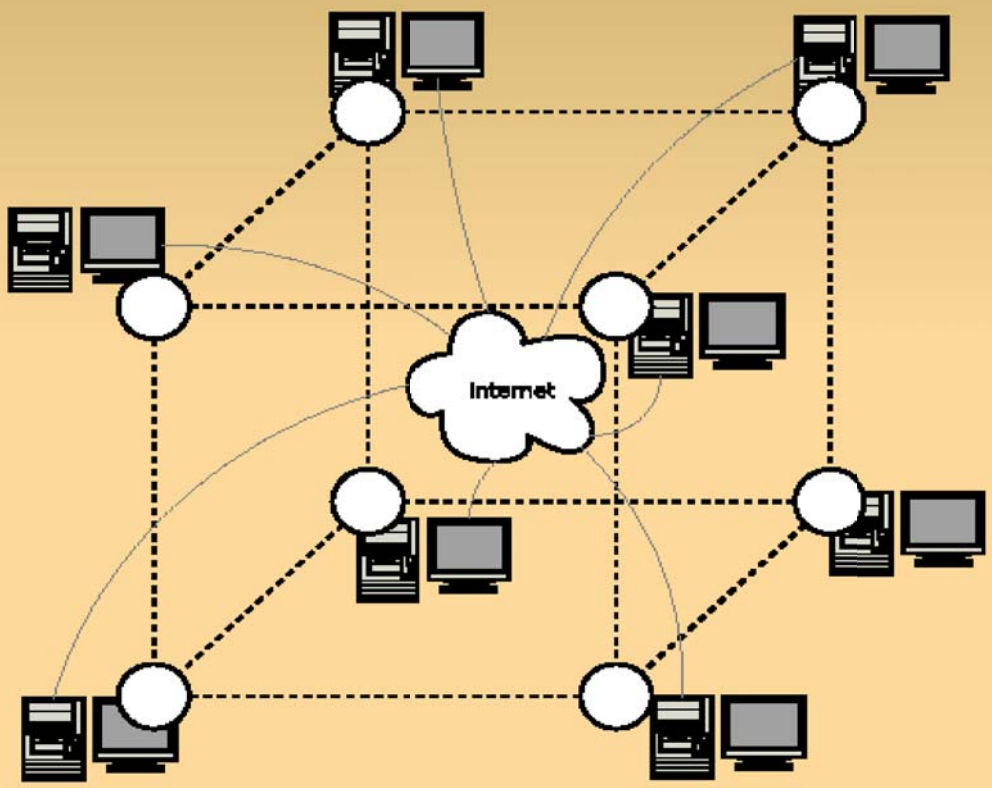
HyperBone

- A scalable overlay network
- Provides monitoring and message routing
- Nodes have identification numbers on the overlay network and connects to theirs neighbours on the virtual topology
- All communication (monitoring and message routing) is done through these connections

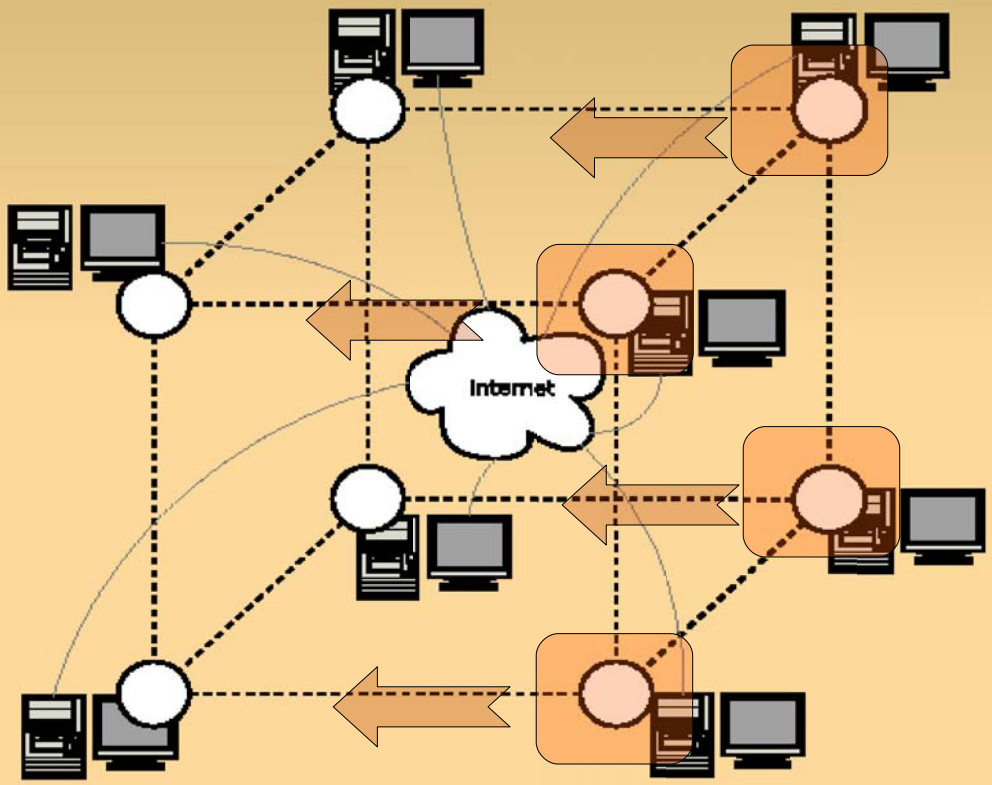
HyperBone

- When the number of nodes is power of 2 and all nodes are online (without failure) the topology is an hypercube
- If the number of nodes is not power of 2 or some node is failed, the system creates new connections to ensure the monitoring and message routing
- The monitoring algorithm used is called DiVHA (*Distributed Virtual Hypercube Algorithm*)

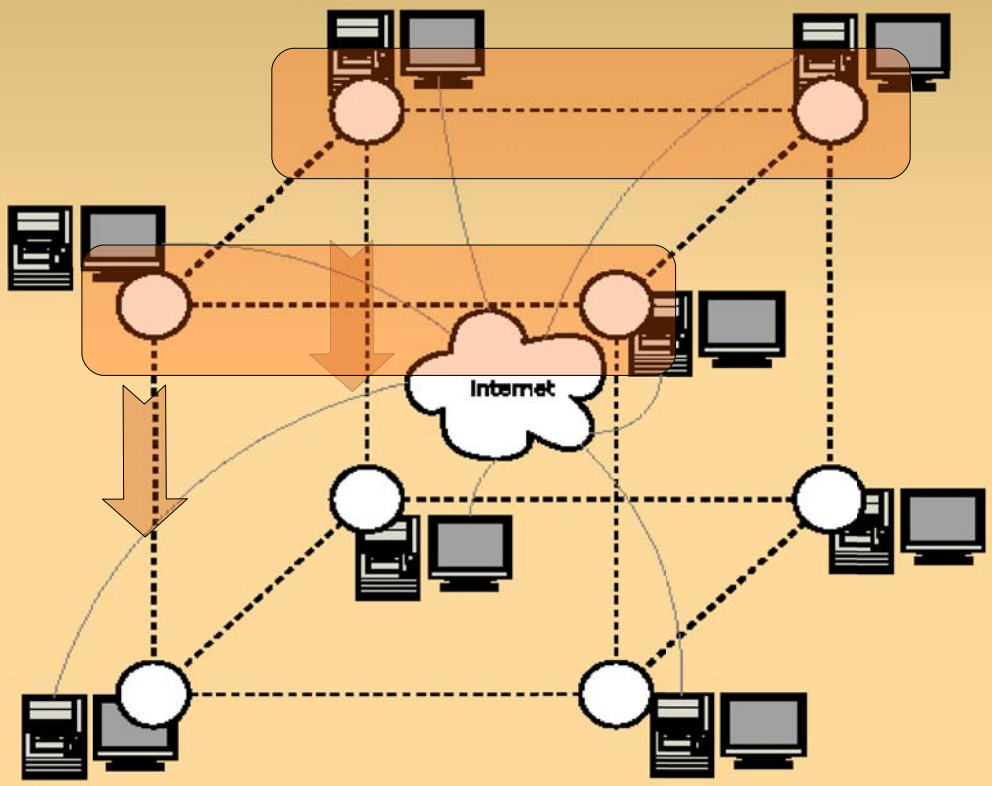
HyperBone



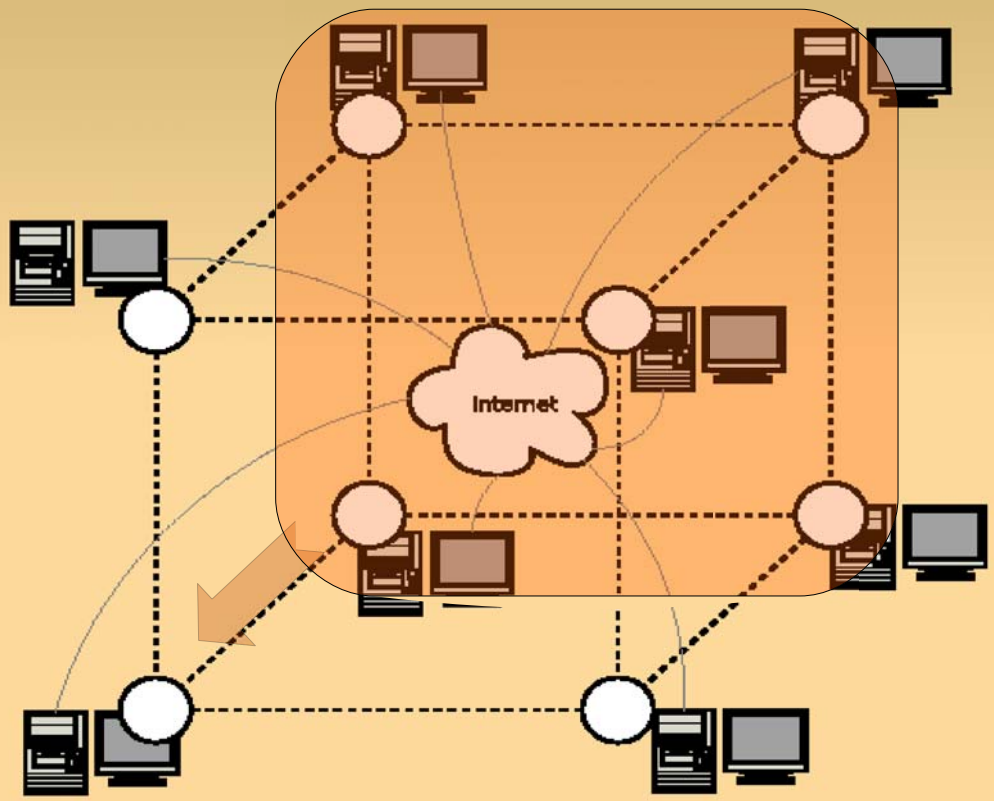
HyperBone



HyperBone



HyperBone



HyperBone

- Each node tests others according its position on the virtual topology (hypercube)
- Number of tests: up to *N log N* at each test round
- Messages are delivered up to *log N* routing steps
- The restrictions above stands even if the hypercube is not complete
- Every node knows the state of all others (within some delay), so it knows the more stable nodes

MPI Standard

- MPI (*Message Passing Interface*) defines an interface for designing distributed and parallel programs using message passing paradigm
- Largely used
- Abstracts low level implementation details of message passing

MPI Standard

- MPI defines operations for
  - Blocking / non-blocking communication
  - Peer-to-peer and collective communication
  - Synchronization (barriers)
  - Architecture independent data types
  - The user can create its own data types (structs)

MPI Standard

- MPICH library is an open source implementation of the MPI standard
- Structured in layers
- The lower layer is called ADI (*Abstract Device Interface*)
- Make easy porting for new architectures / devices

MPI Standard

- For each device that is able to perform message delivery, an ADI software device is implemented
- The library have several devices implemented, such as:
  - *ch_p4*: TCP connections
  - *ch_shmem*: shared memory
  - *ch_globus2*: globus computational grid

MPI Standard

- Each ADI software device is obligated to implement just a small set of basic functions
- The other features from the MPI Standard are achieved combining these basic functions
- Optionally, the device can implement other functions to provide better performance

ch_hyperbone

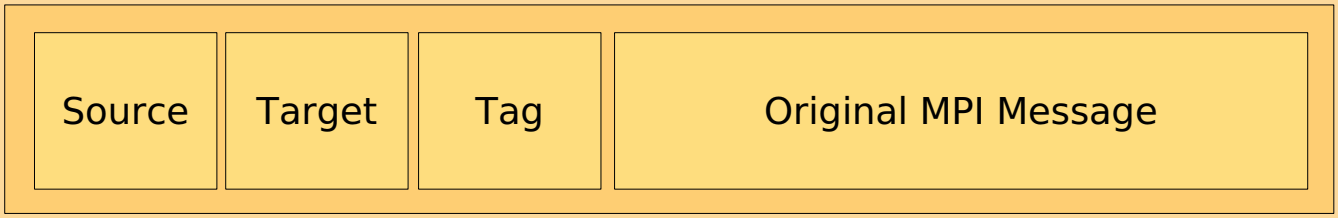
- ADI software device that uses Hyperbone to perform message delivery
- Implement just the mandatory basic functions from ADI

ch_hyperbone

- Program receives its identification number on the overlay network as parameter, added by launch script
- Each node have at least 3 active threads
  - Monitoring
  - Message switching
  - User application

ch_hyperbone

- Before sending each message, a header is added:
  - Source Identification (4 bytes)
  - Target Identification (4 bytes)
  - Tag (4 bytes)



- Hyperbone provides a function that returns the next hop to reach the target

ch_hyperbone

- When receiving a message, the node verify if it is the target or the message needs to be routed to the target
- If it's the target, the header is removed and the message stored in a buffer, from where it's readed by MPICH

Experimental Validation

- Tests on a single cluster (UFPR, 20 x86 PCs, Gigabit ethernet)
  - MergeSort for hypercubes
  - Bandwidth evaluation
  - 3D Image rendering

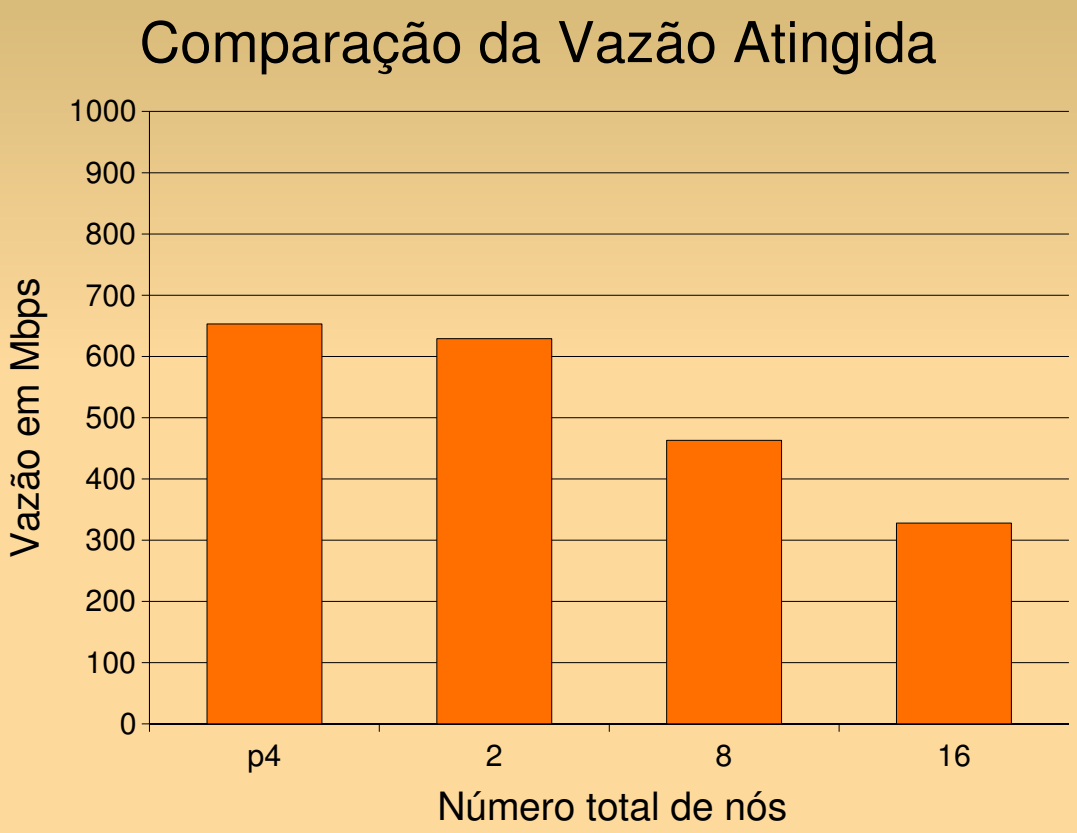
MergeSort for Hypercubes

- In the beginning, each node chooses a random number and put it in an array
- At each iteration, half of the active nodes send its arrays to the other half, that mergesort it with their own arrays
- The nodes that send theirs arrays are removed from the algorithm and the hypercube dimension shrank
- At the end, the node 0 have all choosed numbers in its own array

Bandwidth Evaluation

- Bandwidth is evaluated between the most distant nodes on the hypercube (worst case, biggest amount of routing steps)
- Evaluation of the time spent to transfer 100.000 messages of 8KB each
- Compared to *ch_p4* performance
- Nominal bandwidth: 1Gbps

Bandwidth



Worst case bandwidth

ch_p4:
653 Mbps

2 nodes:
629 Mbps

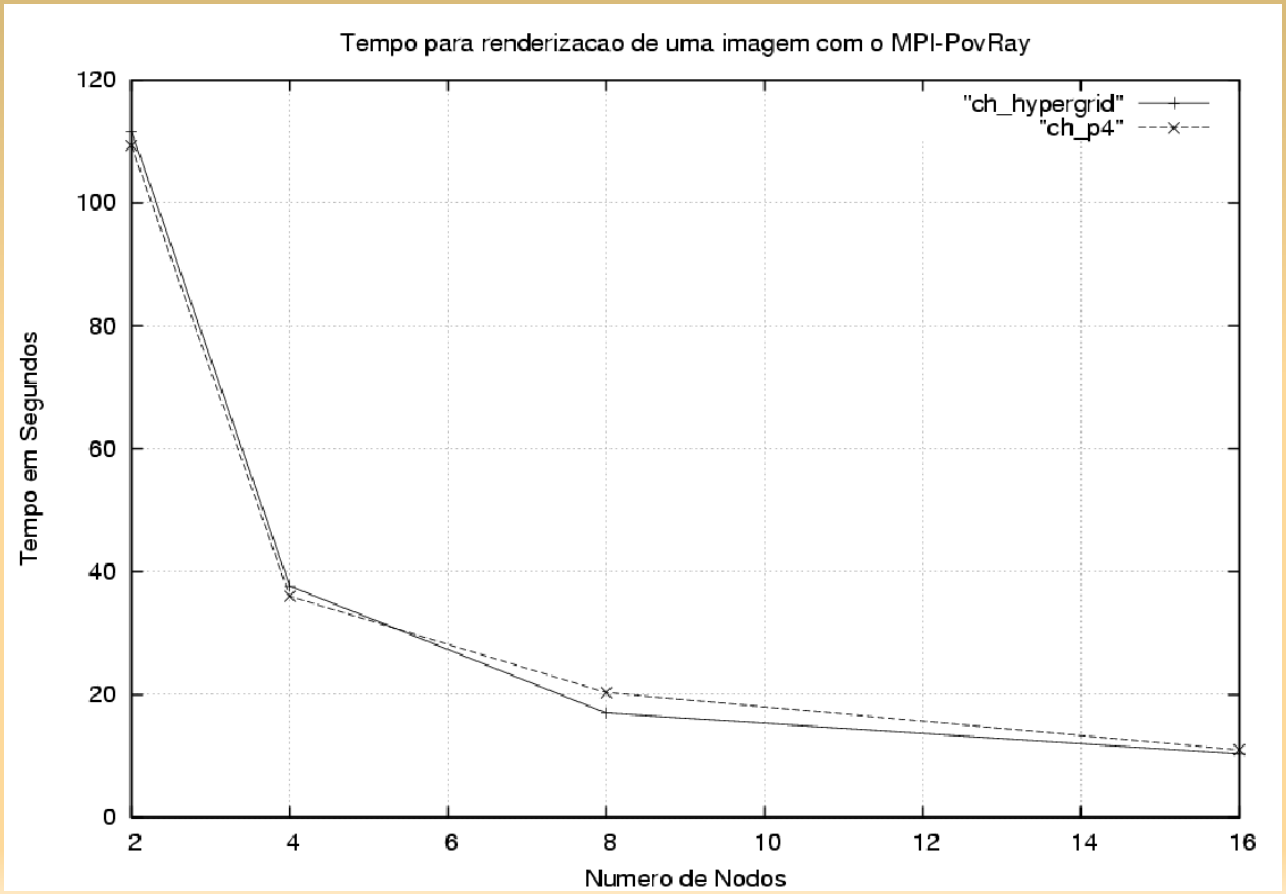
8 nodes:
463 Mbps

16 nodes:
328 Mbps

3D Image Rendering

- The program used was MPI-PovRay
- The same image was rendered using 2, 4, 8 e 16 nodes using *ch_hyperbone* and *ch_p4* devices
- Program takes significantly more time performing computation than communicating with others nodes
- Performance close to *ch_p4*'s

3D Image Rendering



Conclusion

- HyperBone is an overlay network that provides monitoring and message routing
- ch_hyperbone allows MPI parallel and distributed programs to use HyperBone as message delivery system
- Three tests executed shows that:
  - Routing in the virtual topology shrinks the bandwidth
  - For some applications the overall impact of bandwidth shrink can be small