

---

**Second International Workshop  
on Operating Systems, Programming Environments and Management Tools  
for High-Performance Computing on Clusters  
(COSET-2)**

**Remote-Write Communication Protocol  
For Clusters and Grids**

**Ouissem Ben Fredj & Éric Renault**

**GET / INT - France**

June 19, 2005

# Roadmap

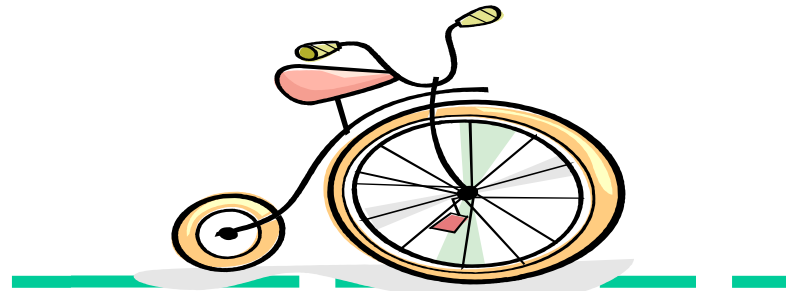
- High-speed network use
  - Programming models
  - Remote-write programming model
- Communication's critical path
  - Memory management
  - Host – NIC interface
  - Queues management
  - Data transfer
  - Communication control
- Conclusion & future works

# High-speed network use

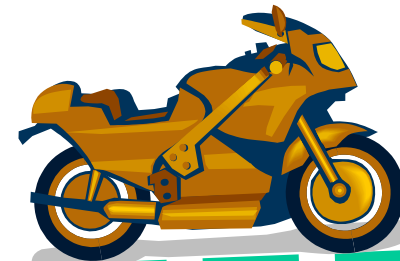
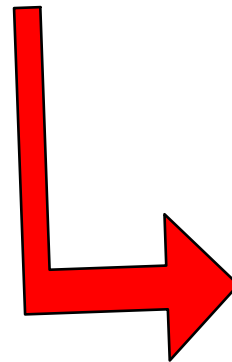
Traditional communication software fails to follow the hardware performance



High-Speed Networks

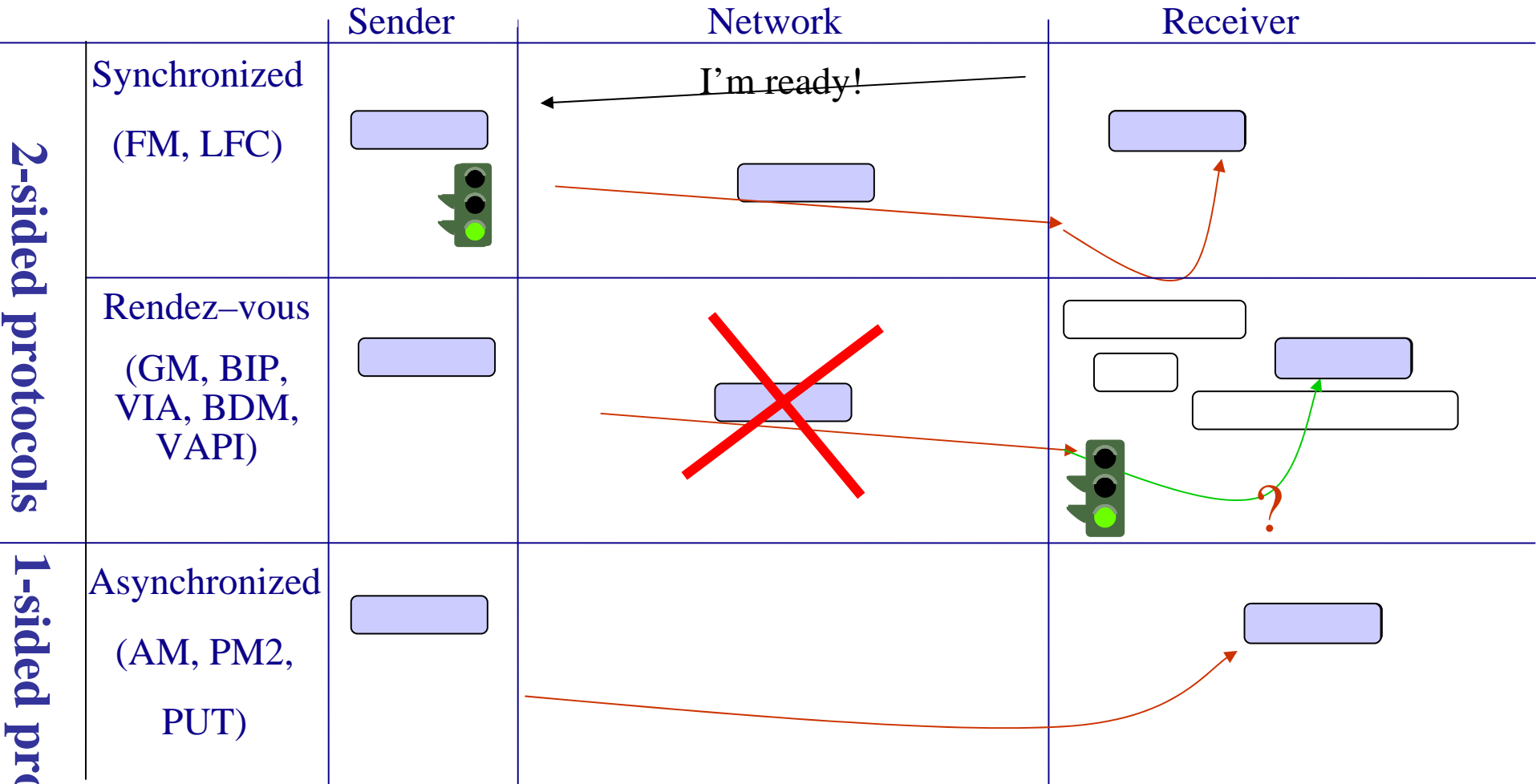


Traditional communication layers



Optimized communication layers

# Programming Models

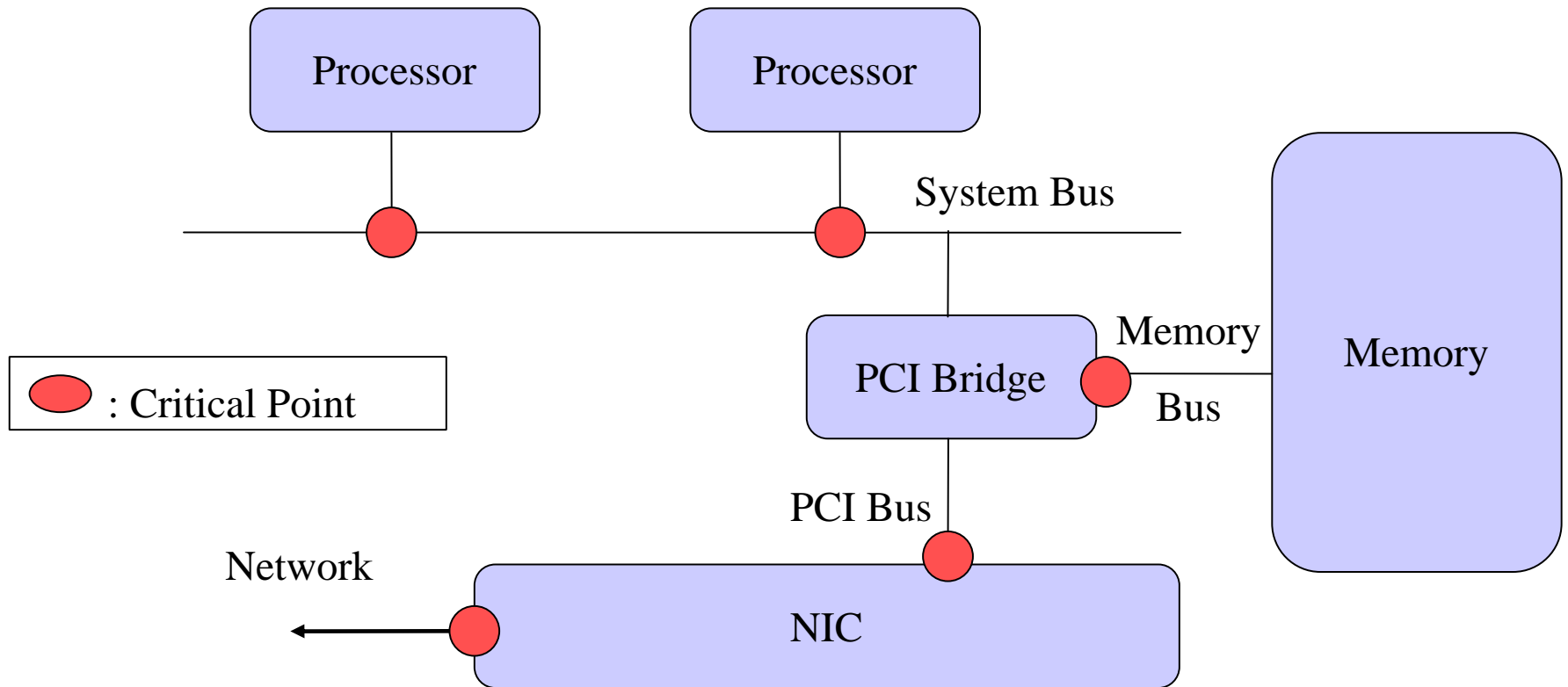


# Programming Models: RW case

RW uses two types of message:

- Normal message
  - Send buffer address
  - Receive buffer address
  - Length
  - Destination process
- Short message
  - Value
  - Destination process

# Communication's Critical Path



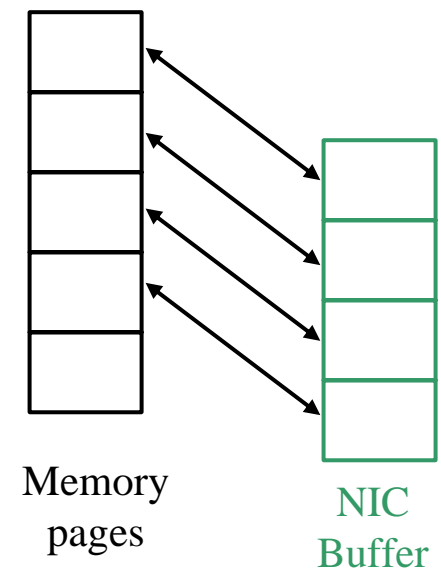
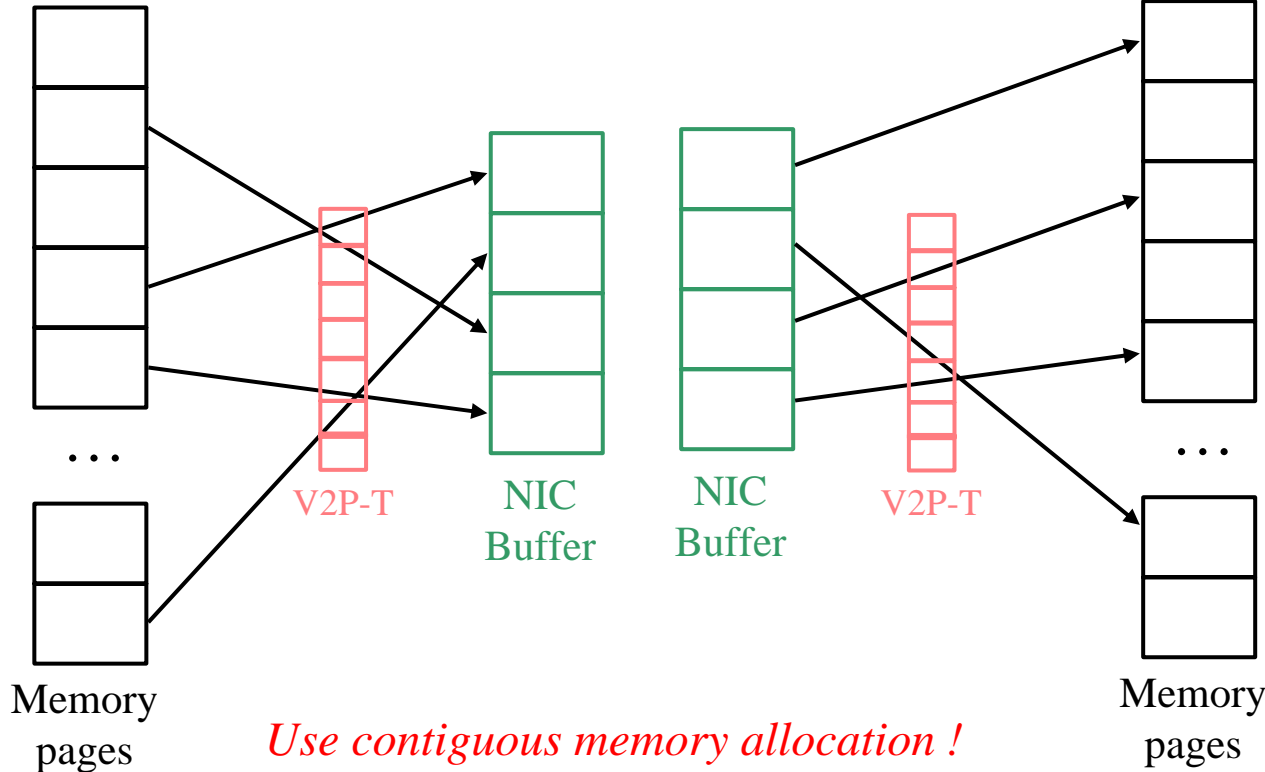
# Memory Management - DMA

Non Contiguous Memory Allocation

Contiguous Memory Allocation

Gather (Send)

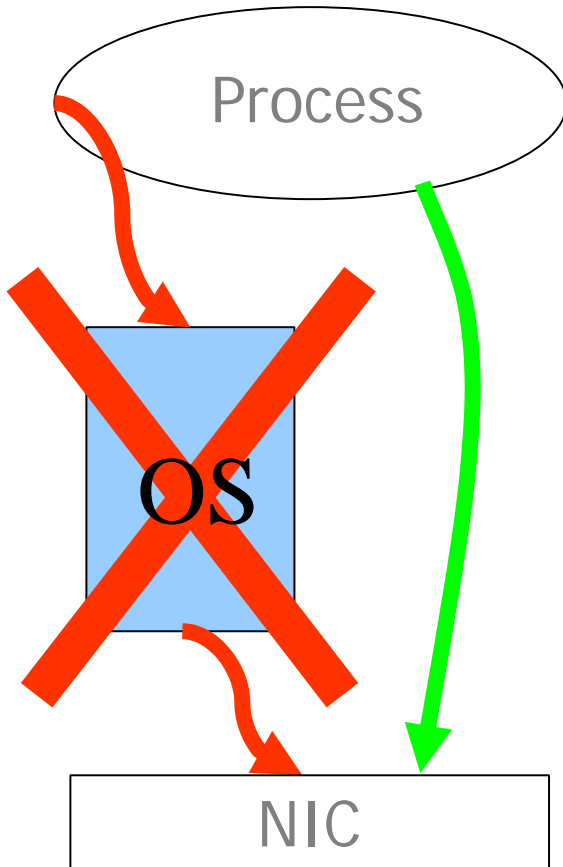
Scatter (Receive)



# Host – NIC Interface

- Host → NIC
  - **PIO**: for Bytes
  - **Write-combining (WC)**: for Structures
  - **DMA**: for large buffers
  
- NIC → Host
  - **DMA**
  
  - *Short message: sent by PIO or WC*
  - *Normal message: seamless transfer*

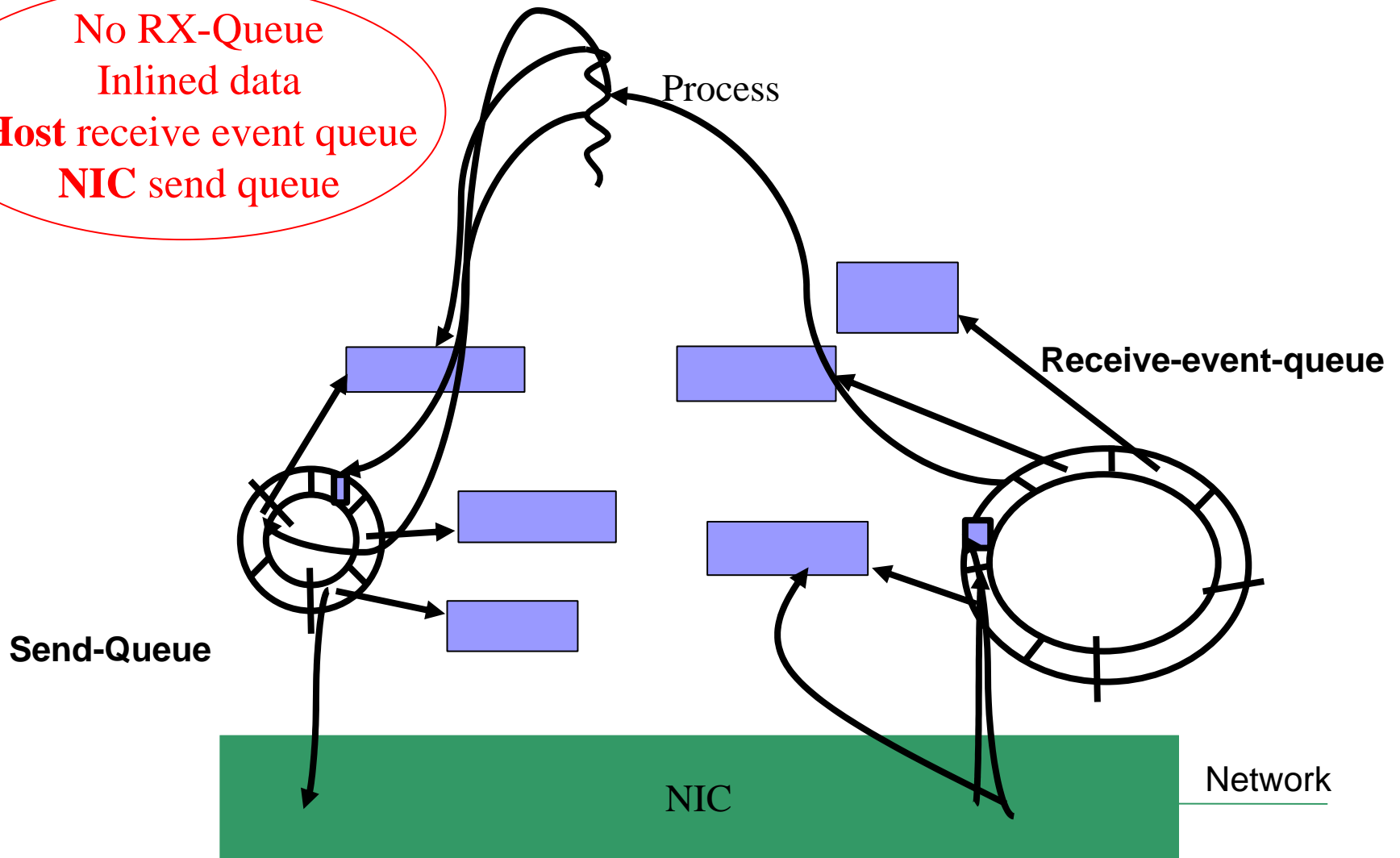
# Recommended Optimizations



- OS-Bypass
- Zero-Copy
- Seamless Transfer

# Queues Management & Data Transfer

No RX-Queue  
Inlined data  
Host receive event queue  
NIC send queue



# Message reception: Interrupt Vs. Polling

	Interrupt	Polling
One Execution / data available		X
Fine-Grain Applications		X
Coarse-Grain Applications	X	
Unprotected Critical Sections		X
Asynchronous Communications	X	

Polling and Interrupt Costs are App-dependent !

- *Let the user choose either polling or interrupt !*
- *Use interrupt to signal exceptions !*

# Conclusion

- One-sided protocol is easy and efficient
- Seamless transfer: PIO, WC, and DMA
- Physical contiguous memory recommended
- RW distinguishes between short and normal msg
- Transfer recommendations:
  - Zero-copy
  - OS-bypass
  - NIC Send-Queue
  - Host receive-Event-Queue
- Cost of Interrupt and polling is app-dependent

# Present and Future Work

- RW implementation over Myrinet
- Performance (5.5 $\mu$ s, 2 Gbps)
- Implement RW over InfiniBand and Ethernet
- Experiment on Grid