


A record and replay mechanism using programmable network interface cards

Laurent Lefèvre

INRIA / LIP (UMR CNRS, INRIA, ENS, UCB)

Laurent.lefevre@inria.fr

Dieter Kranzlmüller

GUP - Joh. Kepler Univ. Linz

Kranzlmueller@gup.jku.at

PDCN 2005 - Innsbruck - Feb. 2005

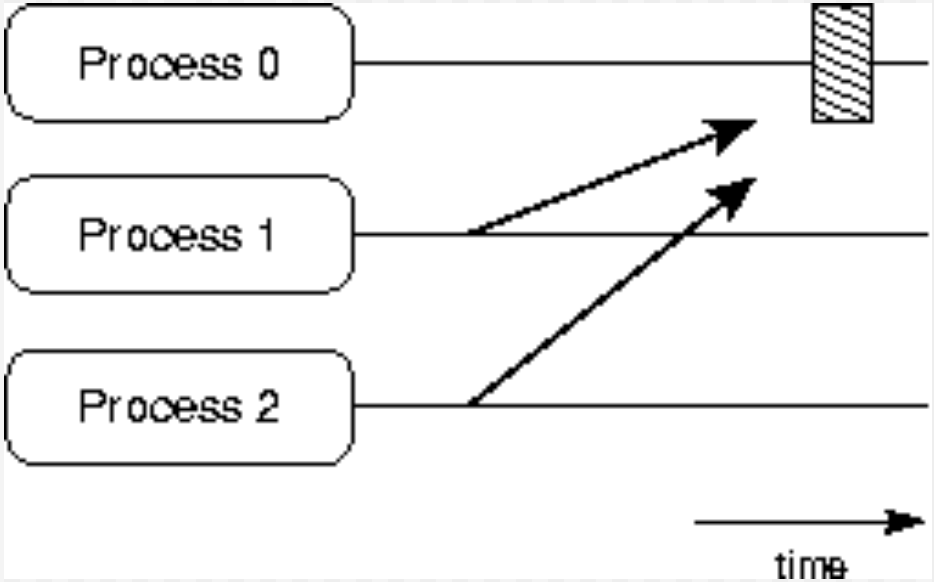
This research is partly supported by French "Programme d'Actions Intégrées Amadeus" funded by the French Ministery of Foreign Affairs and the Austrian Exchange Service (OAD), WTZ Program Amadeus under contract no. 13/2002

Nondeterministic parallel program behavior

- Parallel program
  - Same code
  - Same platform
  - Same input data
  - Different runs
  - ==> Different results !
- Reasons ?
  - Scheduling decisions of processor/ OS
  - Cache contents, cache conflicts
  - Memory access patterns
  - Network conflicts
  - Non determinism in the network

Example : MPI applications

- MPI_ANY_SOURCE
- Wilcard receive
- Race condition



Nondeterminism

- Irreproducibility problem
  - Cannot repeat a particular execution
  - No debugging actions possible
- Completeness problem
  - Cannot observe some errors
  - Impossible to test all possible executions
- Probe effect
  - Monitoring actions influence program

Monitoring …

… influences the observed program in

- Time
  - Events are delayed due to monitoring overhead
  - Ordering of events is perturbed
- Space
  - Storing monitoring data requires memory space

Our approach : Monitoring optimizations

- Minimization of monitor overhead through minimal invasive instrumentation
- Minimization of monitor overhead through exploitation of additional hardware
- Usage of clusters with programmable network hardware

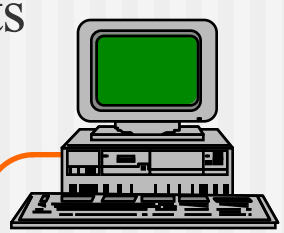
Myrinet clustering

Myrinet NICs

Link Cables
Fiber to 200m

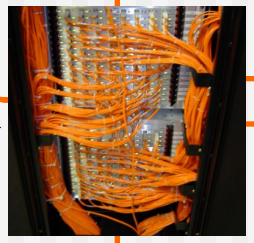
Myrinet Switches

Software
Host
NIC firmware

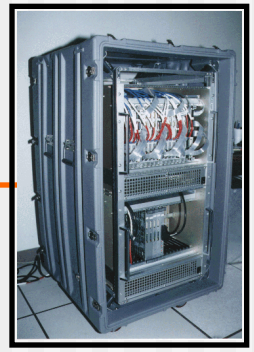


Desktop Hosts

In-Cabinet
Server Clusters



2+2
Gbits/s

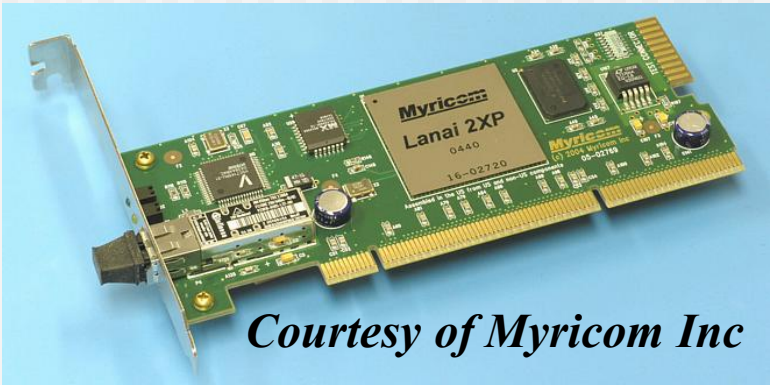


Embedded
Clusters

Courtesy of Myricom Inc

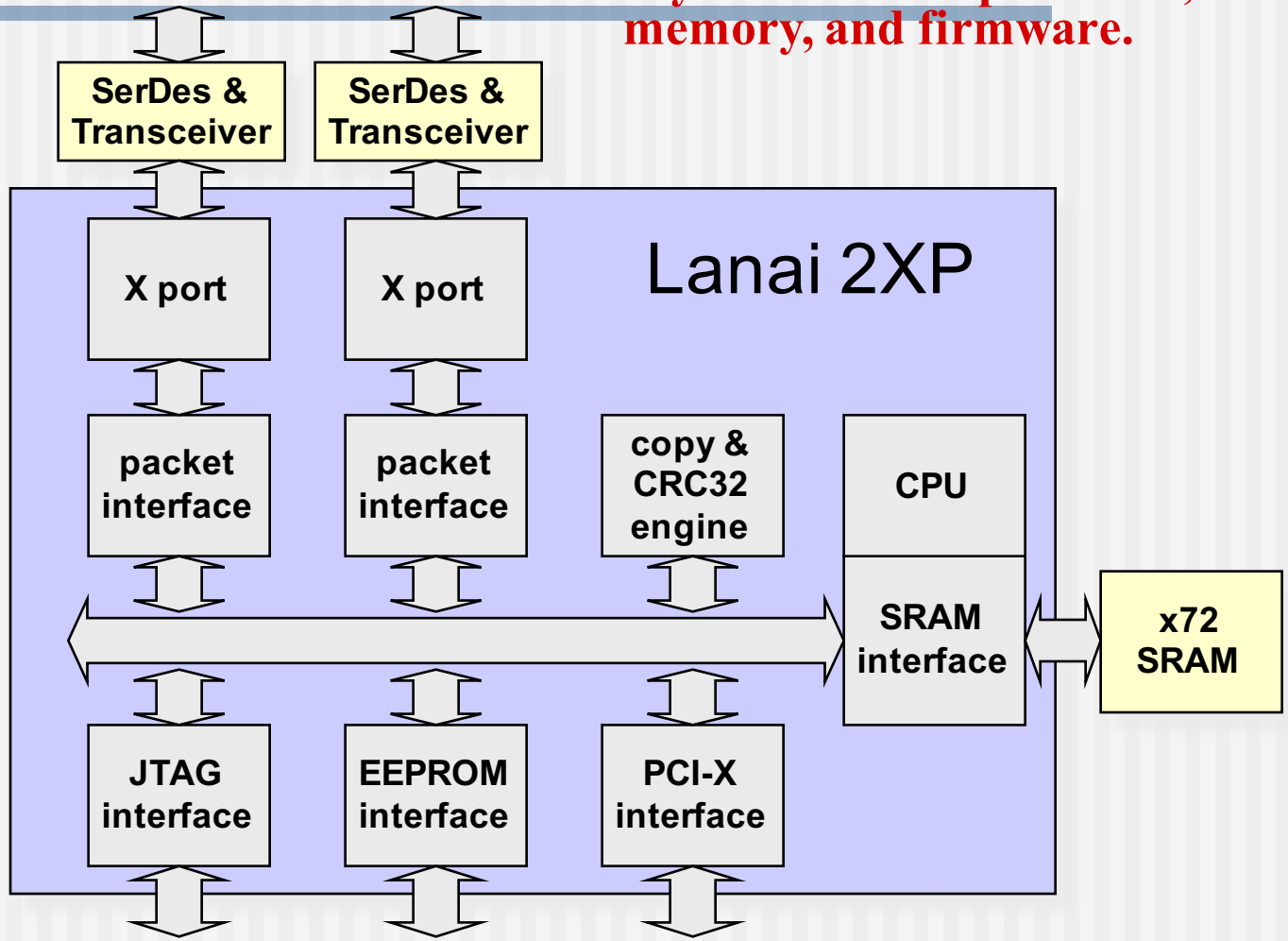
Programmable network cards

- Myrinet NIC
  - Processor on board (Lanai 9.2 RISC 200 Mhz)
  - Memory (2 MB)
  - Communications between host CPU and NIC:
    - Programmed Input/Output (PIO) :dedicated commands
      - Access memory locations
      - Extract NIC status
    - Direct memory access (DMA)
      - Transfert between host and NIC CPU
      - Idenpendant from host
  - GM software
    - Software library
    - Kernel module
    - Myricom Control Program (MCP)



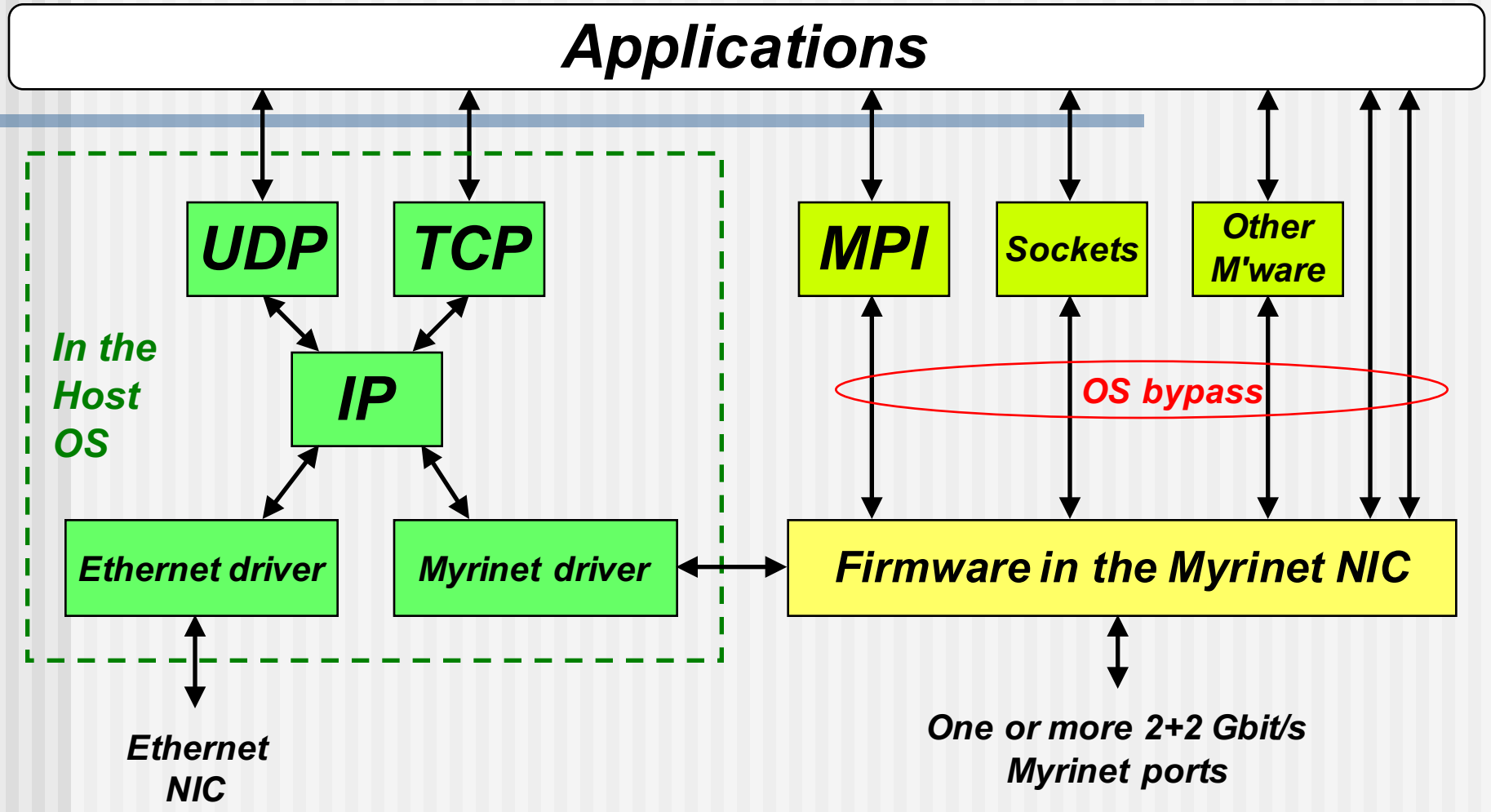
Myrinet NICs = Protocol Offload Engines

Myrinet NICs : processor, memory, and firmware.



Courtesy of Myricom Inc

Myrinet Software Interfaces



Courtesy of Myricom Inc

Monitoring on Programmable network cards

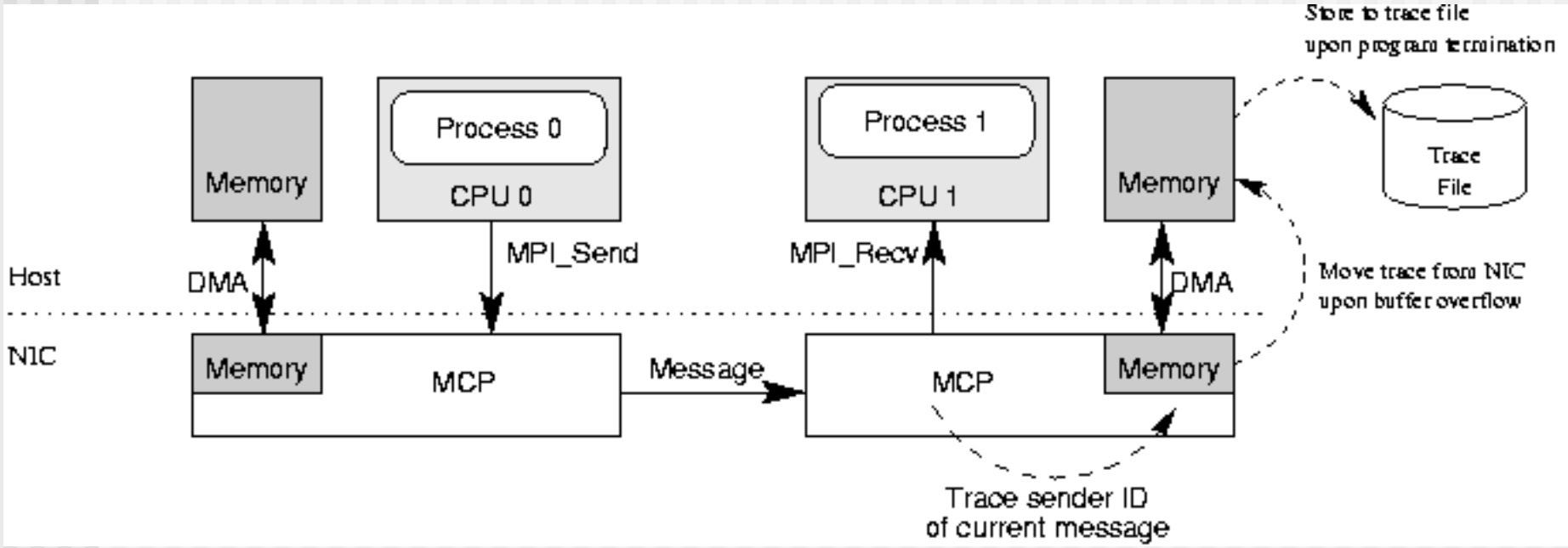
- We deploy Record actions from CPU host to NIC
- Architecture based on 3 steps :
  1. Preparation and instrumentation
  2. Recording execution
  3. Repeated replay phases

Preparation and instrumentation

- Loading modified MCP onto NIC
- Instrumentation of MPI program by including modified MPI header file
- Compiling application with modified MPICH library

Recording execution

- NIC buffer used to store order of incoming messages
- Critical step
- Optimizing based on semantics of MPI :
  - Delivery between 2 nodes arrive in the same order than generated by sender
  - We only trace messages on the receiver side



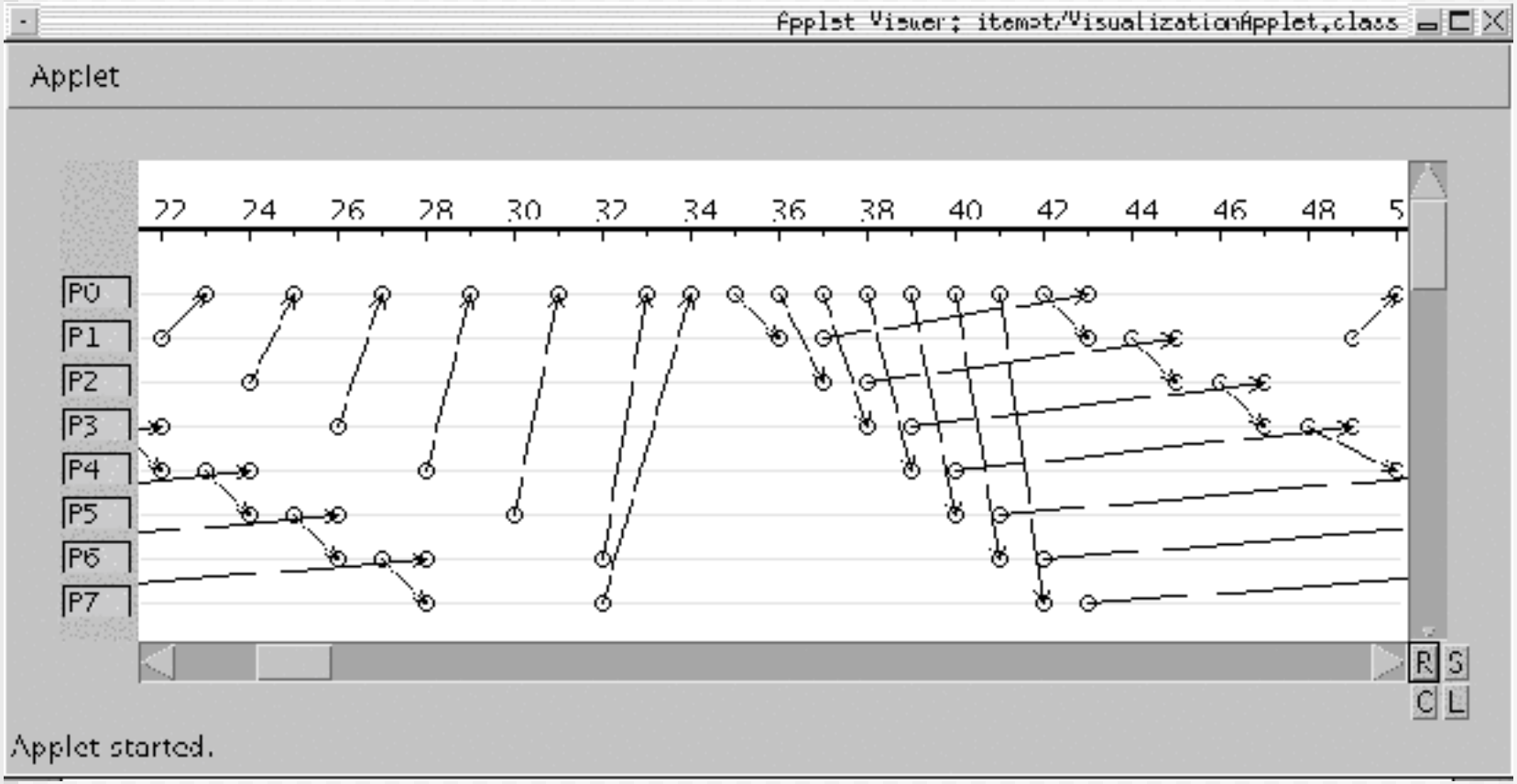
Recording execution

- Upon initialization of MPI program : memory reservation on NIC to store order of incoming messages
- If buffer full : transfer asynchronously to host memory during execution
- After execution : file generation of monitoring information extracted from NIC

Replaying

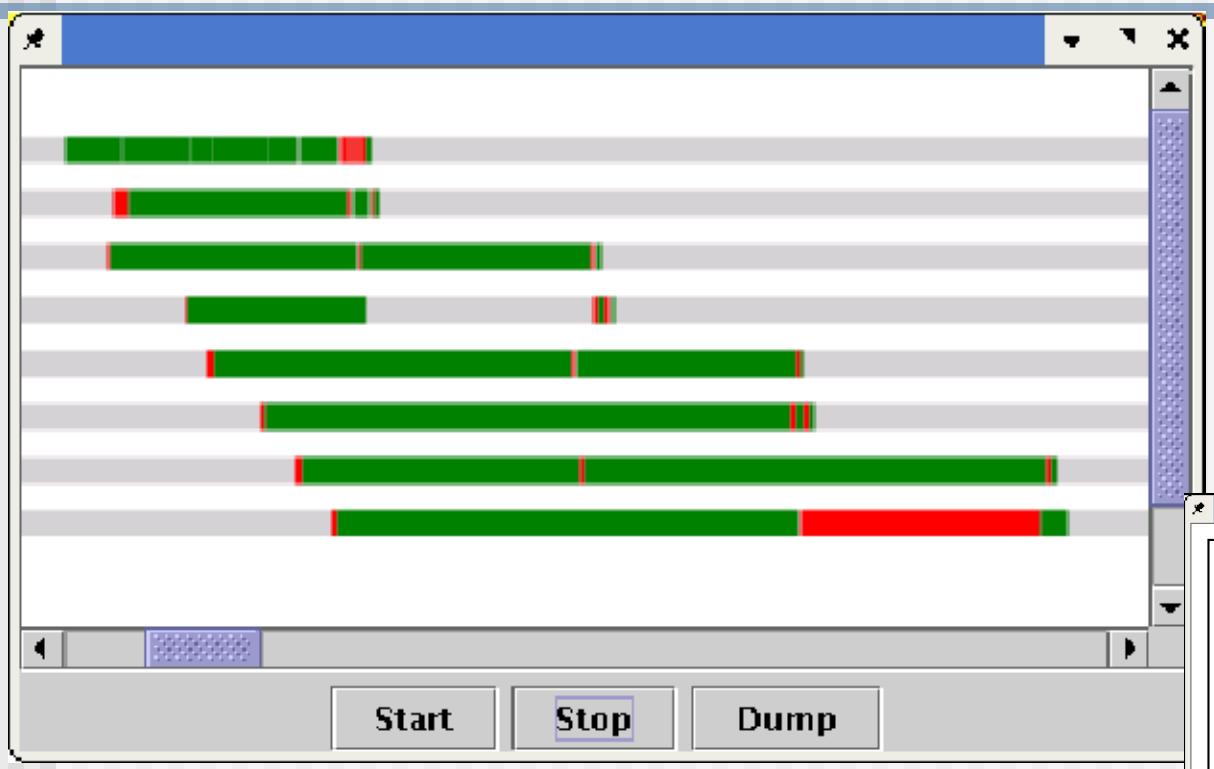
- To increase amount of observation data
- To perform program analysis
- Only hosts are involved
- Using dedicated graphical environments (DeWiz)

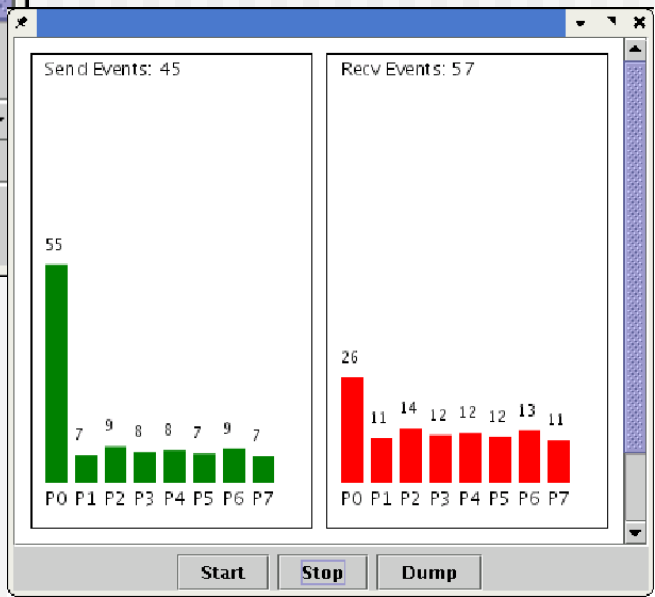
Replaying



Debugging tool DeWiz screenshot with events collected on programmable card

Time graph, counter analysis





Conclusion and current work

- Advantages:
  - Minimal intrusion of during initial record phase
  - Eliminating irreproducibility effect
  - Decreasing the probe effect
- Monitoring without user knowledge

- Tools to manipulae events graph
- Adding QoS functionality on the NIC to filter monitoring actions
- Deploying record and replay mechanisms inside programmable switch