Fraunhofer Institut
Rechnerarchitektur
und Softwaretechnik

Component-Based Communication Support for Parallel Applications running on Clusters of Workstations

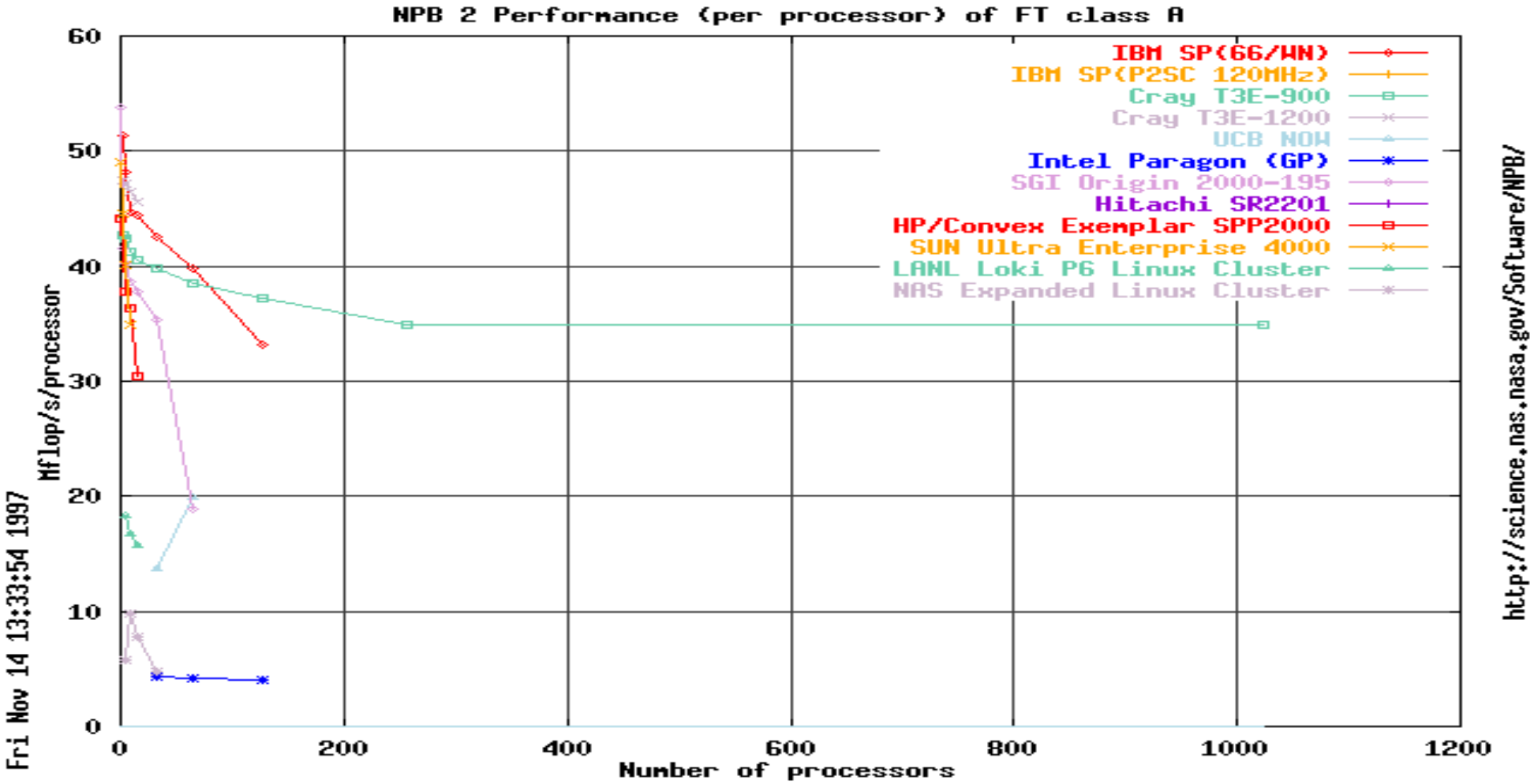
Antônio Augusto Fröhlich

Wolfgang Schöder-Preikschat

{guto|wosch}@first.gmd.de

September 2001

MPP x Cluster



What's wrong with our clusters?

- Hardware
  - Microprocessors and high-speed networks seem to be ok
  - Memory is getting better
  - Inter-node communication is normal I/O
- Software
  - Run-time support system is inadequate
  - Compilers know little about distribution and parallelism
  - No single machine view

A possible solution

commodity hardware

and

custom software

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

Application-orientation

- Each application deserves its own run-time support system
- Application run-time demands at generation-time
  - Unknown => dynamic adaptation
  - Known => static configuration
    - Parallel and embedded applications
- Adaptability and configurability
  - Component-based software engineering

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

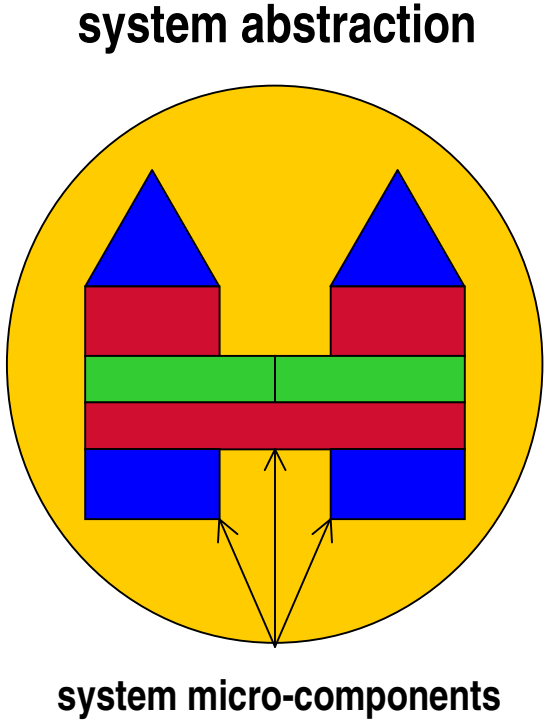
EPOS: Embedded Parallel Operating System

- EPOS operating system
  - Application-driven assemblage of system components
- EPOS components
  - Statically configurable, application-ready system abstractions
- EPOS and the real world
  - Intelligent visual tools for configuration
  - Invisibility
    - Standard libraries (Posix files, libc, libstdc++, libm)
    - Standard APIs (Posix threads, MPI)

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

Scenario-independent system abstractions

- Application-ready components
- Independent from execution scenario:
  - A mailbox, NOT a mailbox for an specific protocol or network
- Communication abstractions
  - Network adapter
    - datagram, stream, active message and asynchronous remote copy
  - Communicator
    - link, port, mailbox, DSM and ROI

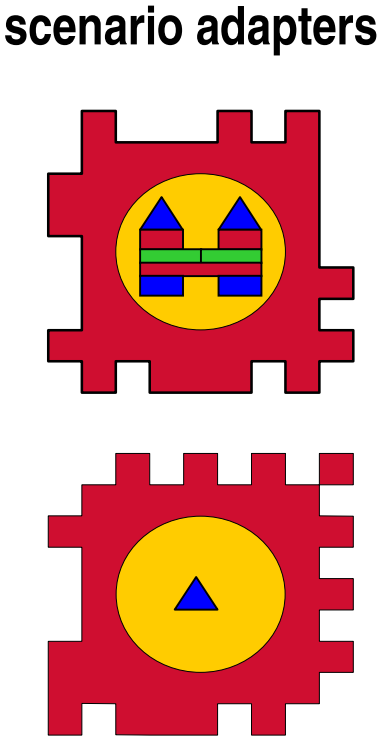


Fraunhofer FIRST Institut
Rechnerarchitektur
und Softwaretechnik

Scenario adapters

- Adapt existing abstractions to an execution scenario
- Decorate abstraction with scenario specific constructs

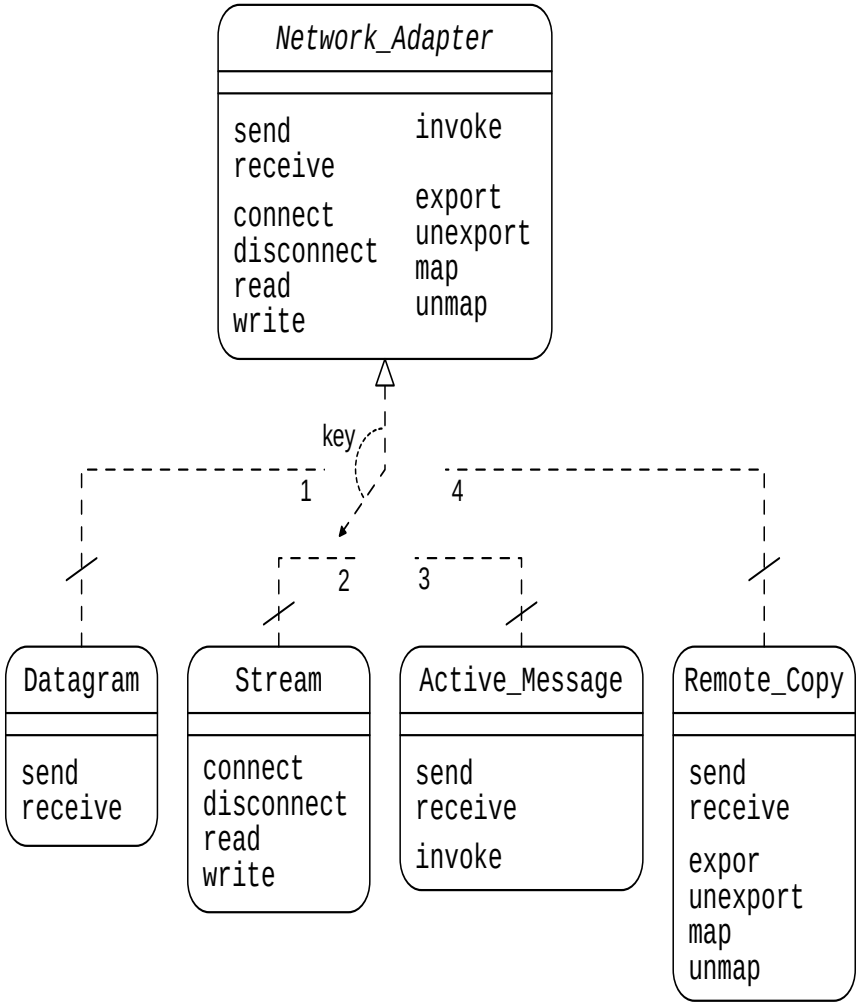
- Communication scenarios
  - kernel, library, protected, unprotected, multi/single task, multi/single thread



Inflated interfaces

- Export system abstractions to applications
- Well-known to application programmers
- Comprehensive
- Promote requirement analysis

- Communication interfaces
  - network adapter and communicator



Framework

- Staticly metaprogrammed
  - Run at compile-time
  - No run-time overhead
  - Implemented as C++ templates
- Abstraction types
  - Wrapped
    - No allocation or sharing control
    - No cross-domain invocation
    - Always embedded in the application
  - Controlled
    - Allocation and sharing control is possible
    - Cross-domain invocation is possible
    - Embedded in the application or packed in a kernel

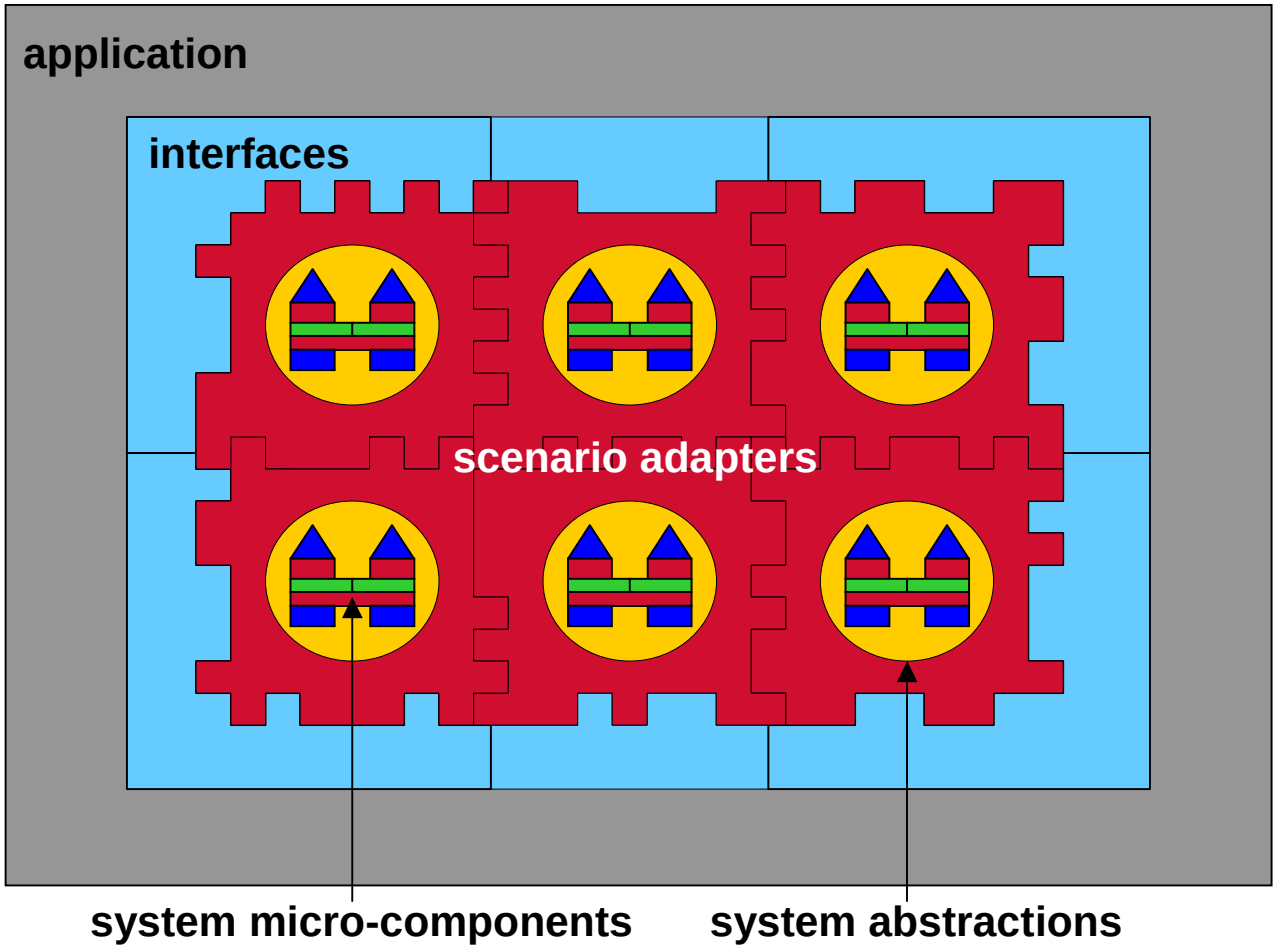
Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

Automatic generation

- Application refers to inflated interfaces
- Requirement analyser parses the application
  - Which interfaces are referred to?
  - How are they referred to?
- Expert system selects the realizations that better match the referred inflated interfaces
- An application-oriented operating system is compiled

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

Application-oriented EPOS

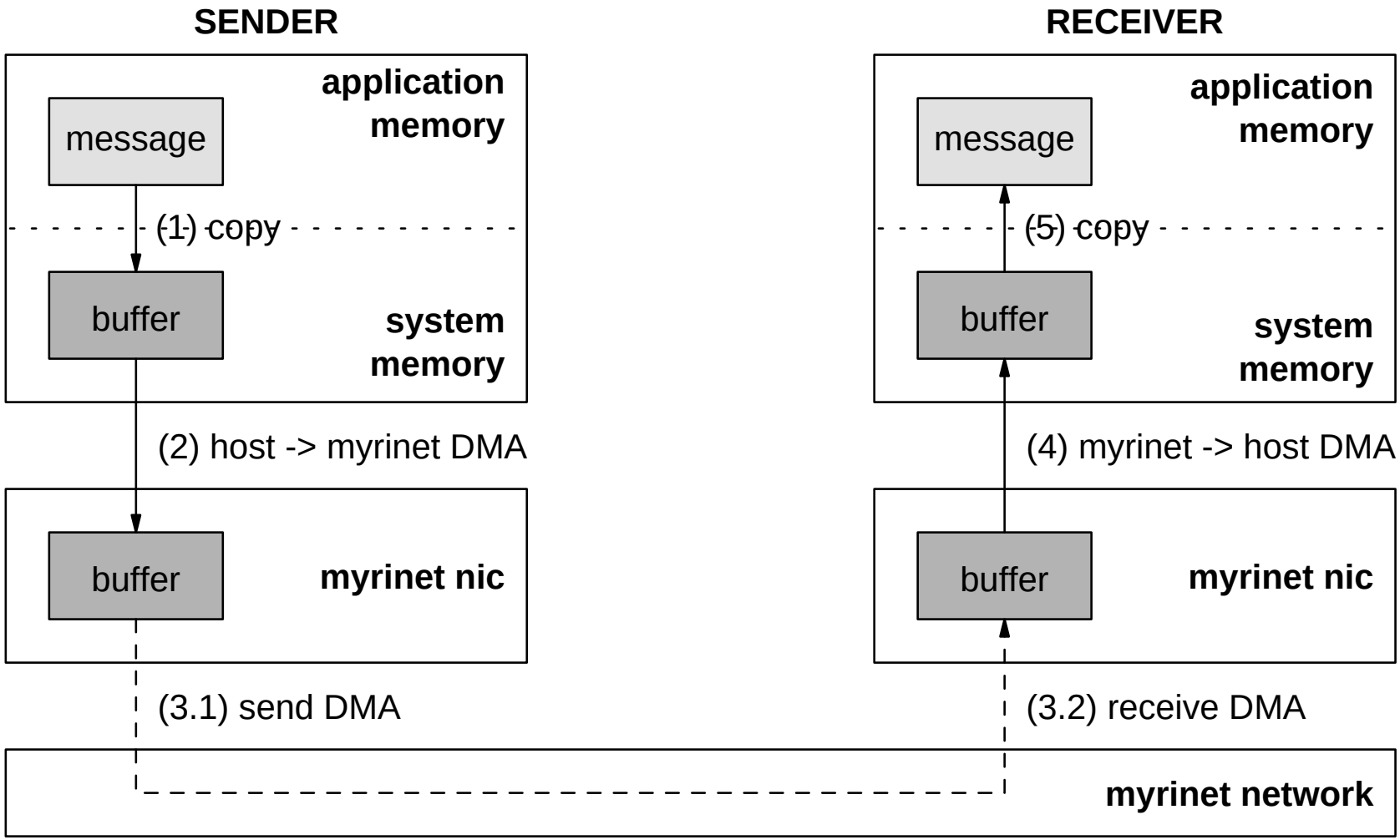


The SNOW cluster

- Working nodes (16)
  - AMD Atlon at 550 MHz, 128 MB at 100 MHz
  - 32 bits PCI bus at 33 MHz
- Networks
  - Switched, full-duplex, 100 Mbps Ethernet with giga ethernet up link
  - Full-duplex, 1.28 Gbps Myrinet on a cross-bar
- Performance
  - Memory copy bandwidth => 140 MB/s
  - Host / Myrinet DMA bandwidth => 130 MB/s
  - Myrinet bandwidth => 160 MB/s

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

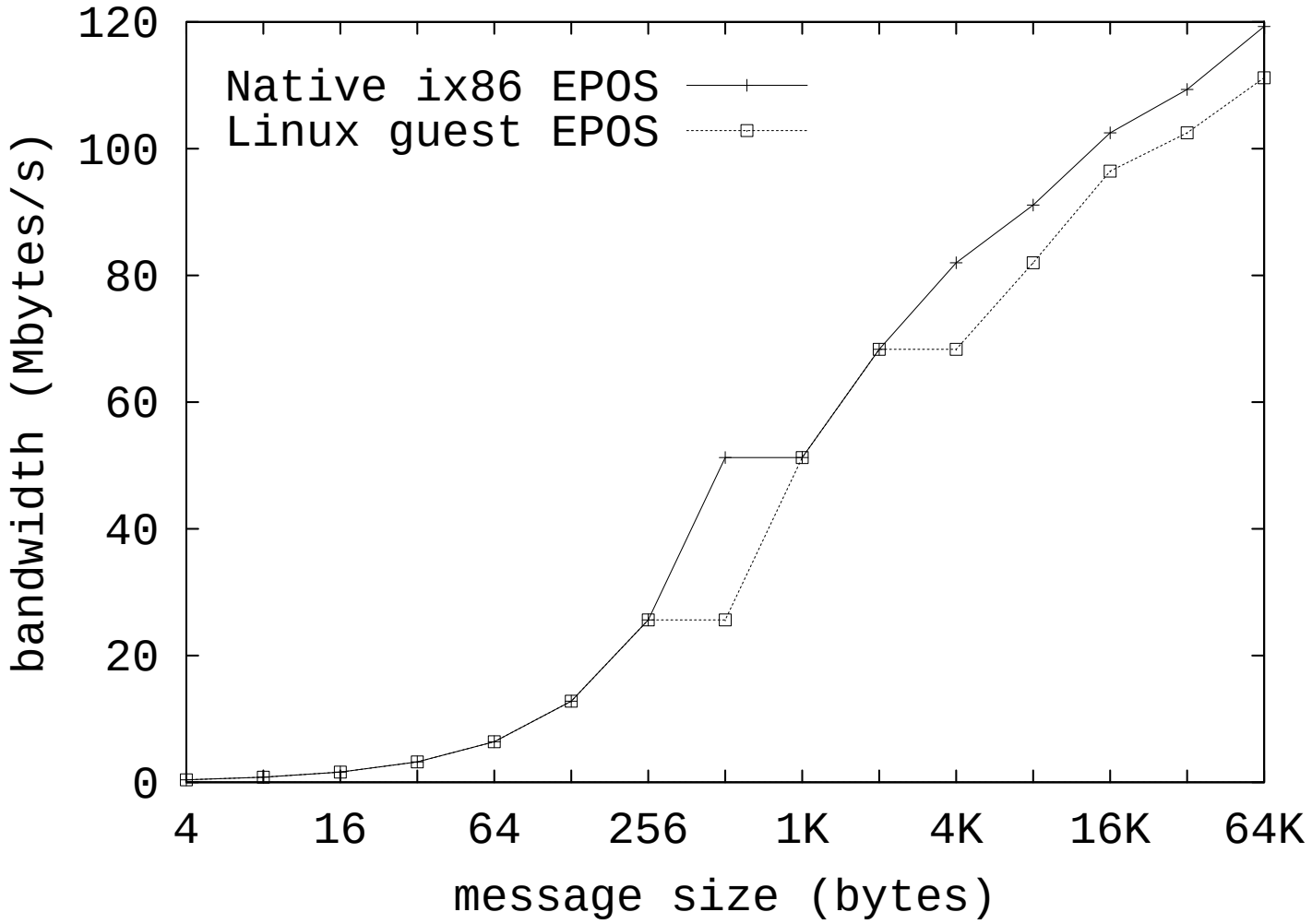
Communication pipeline



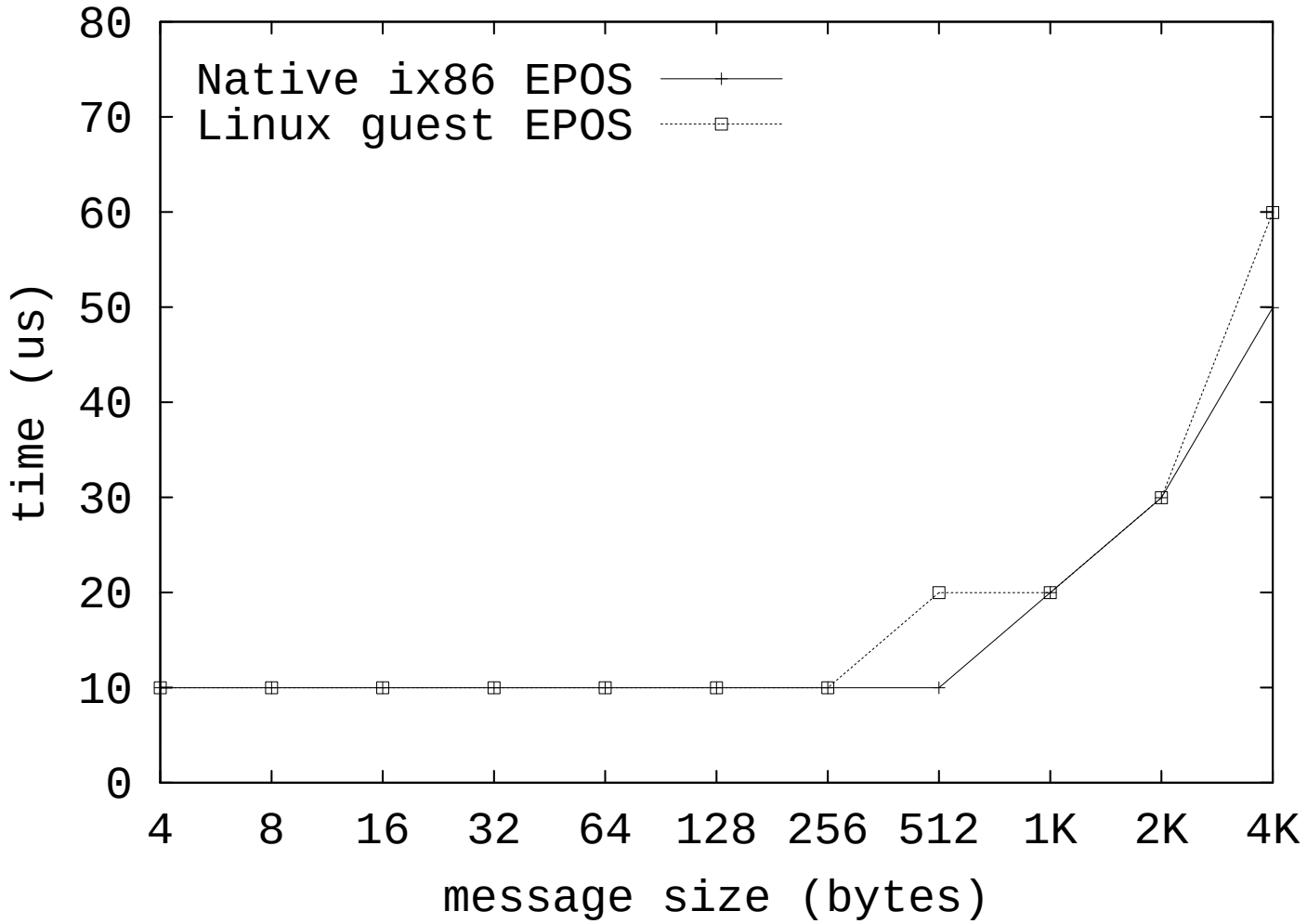
Short messages

- Pipeline synchronization is relativelly expensive
- Programmed I/O
  - Write conbine cache policy
  - Burst PCI transaction
- Short message < 256 bytes

Bandwidth


Native ix86 EPOS
Linux guest EPOS
bandwidth (Mbytes/s)
message size (bytes)
0
20
40
60
80
100
120
4
16
64
256
1K
4K
16K
64K

Latency


Native ix86 EPOS
Linux guest EPOS
time (us)
message size (bytes)
0
10
20
30
40
50
60
70
80
4
8
16
32
64
128
256
512
1K
2K
4K

Conclusion

- EPOS communication system, although highly adaptable, presents very low overhead
- Tools are being implemented and the system abstraction repository is growing
- First real applications are being negotiate
- MPI adaptation layer is missing