SCI-Europe 2000

SCI-MPICH - The Second Generation

Joachim Worringen

RWTH Scalable Computing Aachen

Lehrstuhl für Betriebssysteme

AGENDA

- Introduction
- Various Design Improvements
- Asynchronous Message Transfers
- MPI-IO via SCI
- Summary & Outlook

INTRODUCTION

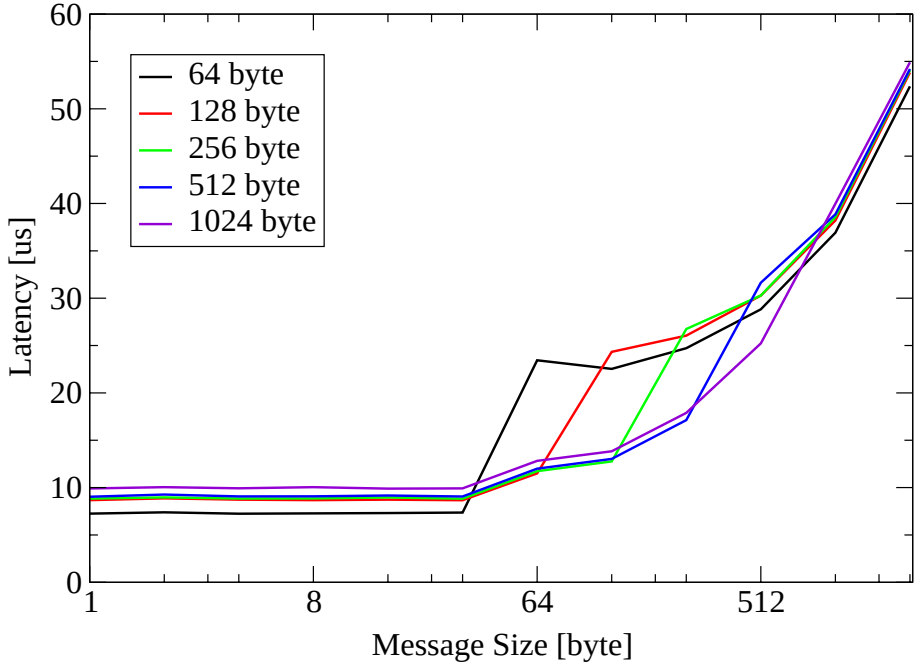
- SCI-MPICH
  - freely available MPI-1 implementation for SCI Interconnect
  - high, competitive performance
  - first presented on SCI Europe ‘99
- Improvement was required in terms of
  - reliability & stability
  - compatibility
  - usability
- Utilization of DMA transfer modes
  - support of asynchronous send operations
- Introduction of MPI-2 features
  - parallel I/O with MPI-IO API

STARTUP & SHUTDOWN

- Adaptive Memory Configuration
  - communication buffer setup customizable via configuration file
  - reduction of buffer sizes if not enough resources available
- Delayed Segment Connection
  - reduce startup time
  - spare system resources
- Resource Management
  - register allocation of all relevant system resources
- Watchdog
  - terminate local process if a remote process has terminated
  - free all allocated resources on shutdown

SHORT MESSAGES

- different sizes of stream buffers
  - efficiently use the available stream buffers
- longer „short“ messages
  - self-synchronizing messages of arbitrary length



ASYNCHRONOUS MESSAGE TRANSFER

- MPI offers asynchronous message transfer operations:

Sender

```
prepare send buffer
MPI_Isend( .... );
do other stuff
MPI_Wait( ... );
reuse send buffer
```

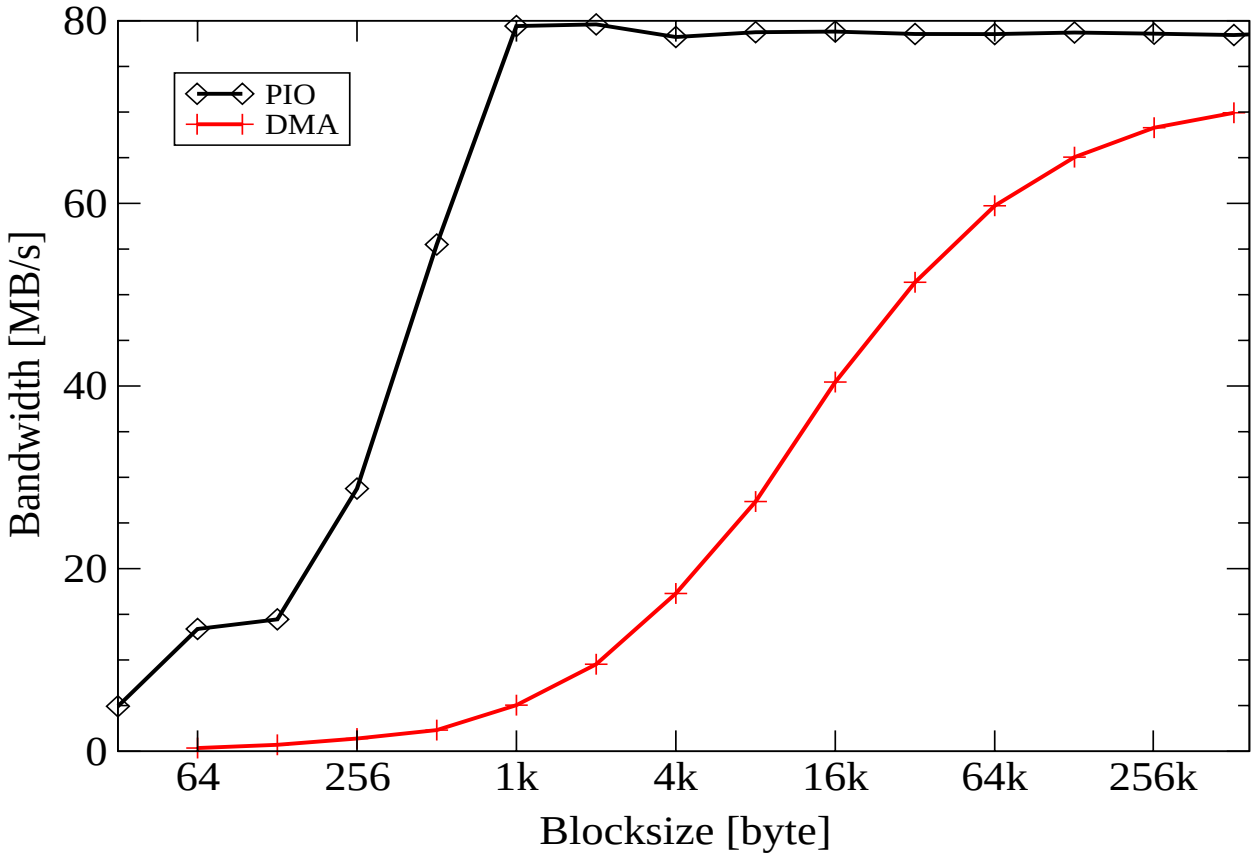
Receiver

```
prepare receive buffer
MPI_Irecv( .... );
do other stuff
MPI_Wait( ... );
use receive buffer
```

MPI Progress Rule does not guarantee overlapping of communication and computation
→ progress **between MPI function calls** required

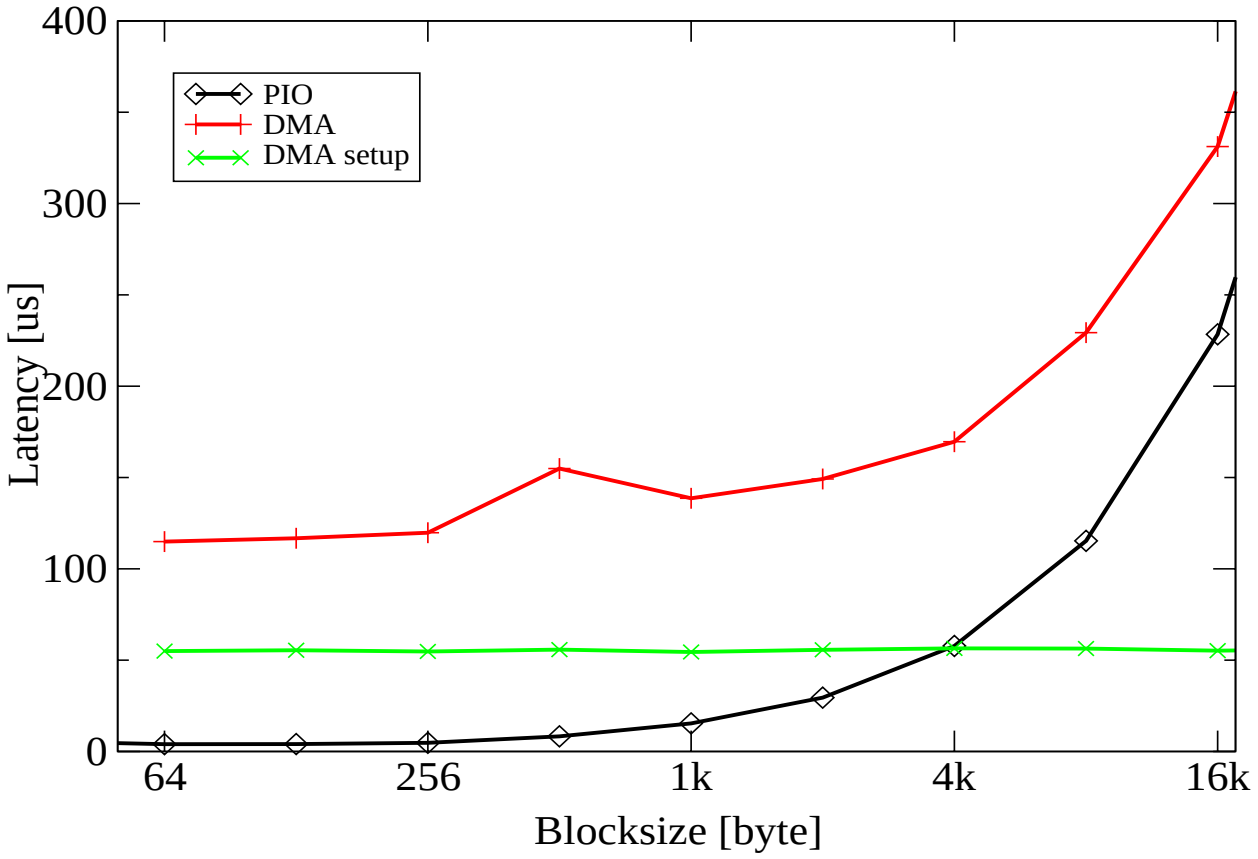
DATA TRANSFER VIA DMA

PIO vs. DMA: Bandwidth

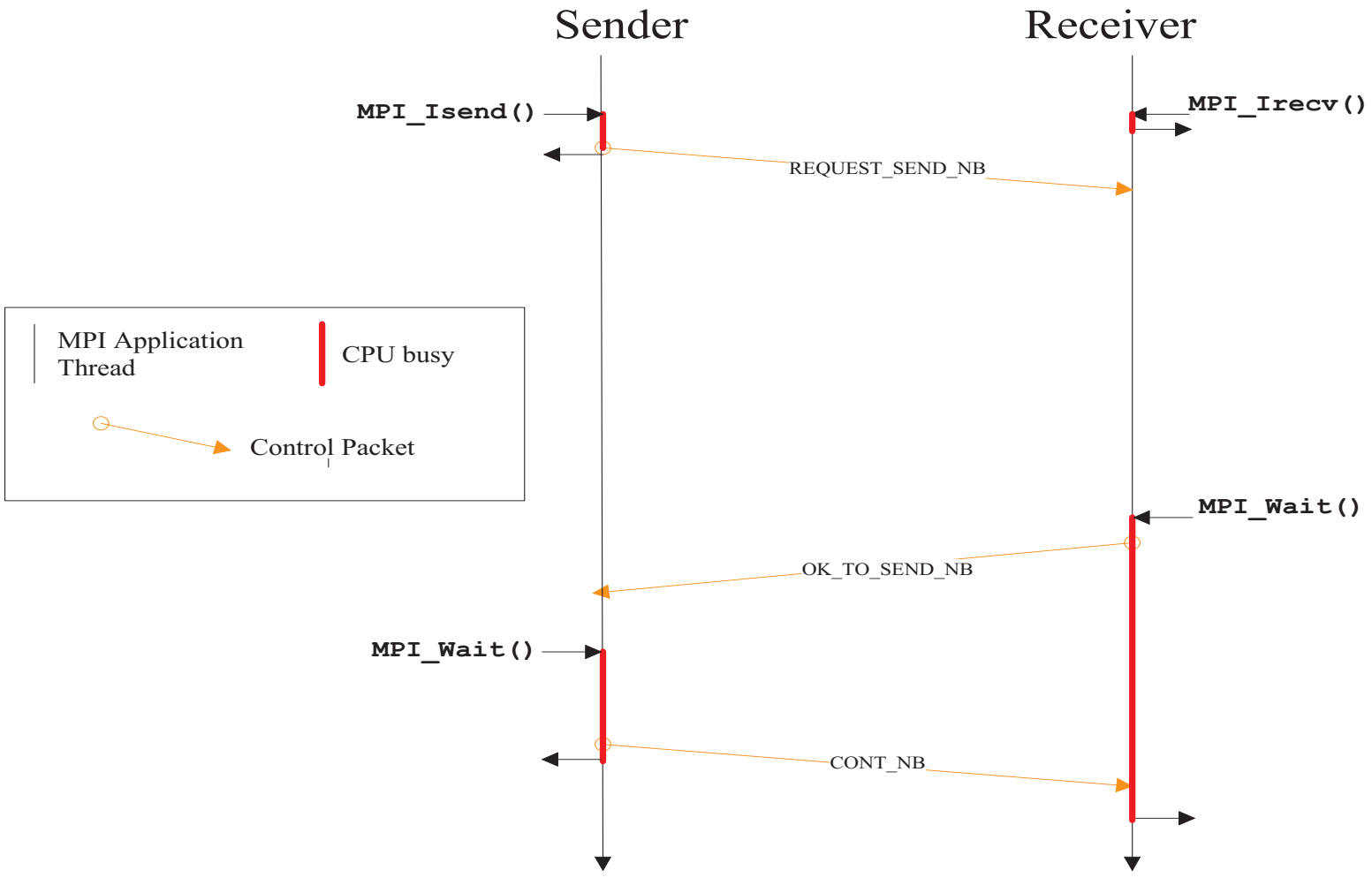


DATA TRANSFER VIA DMA

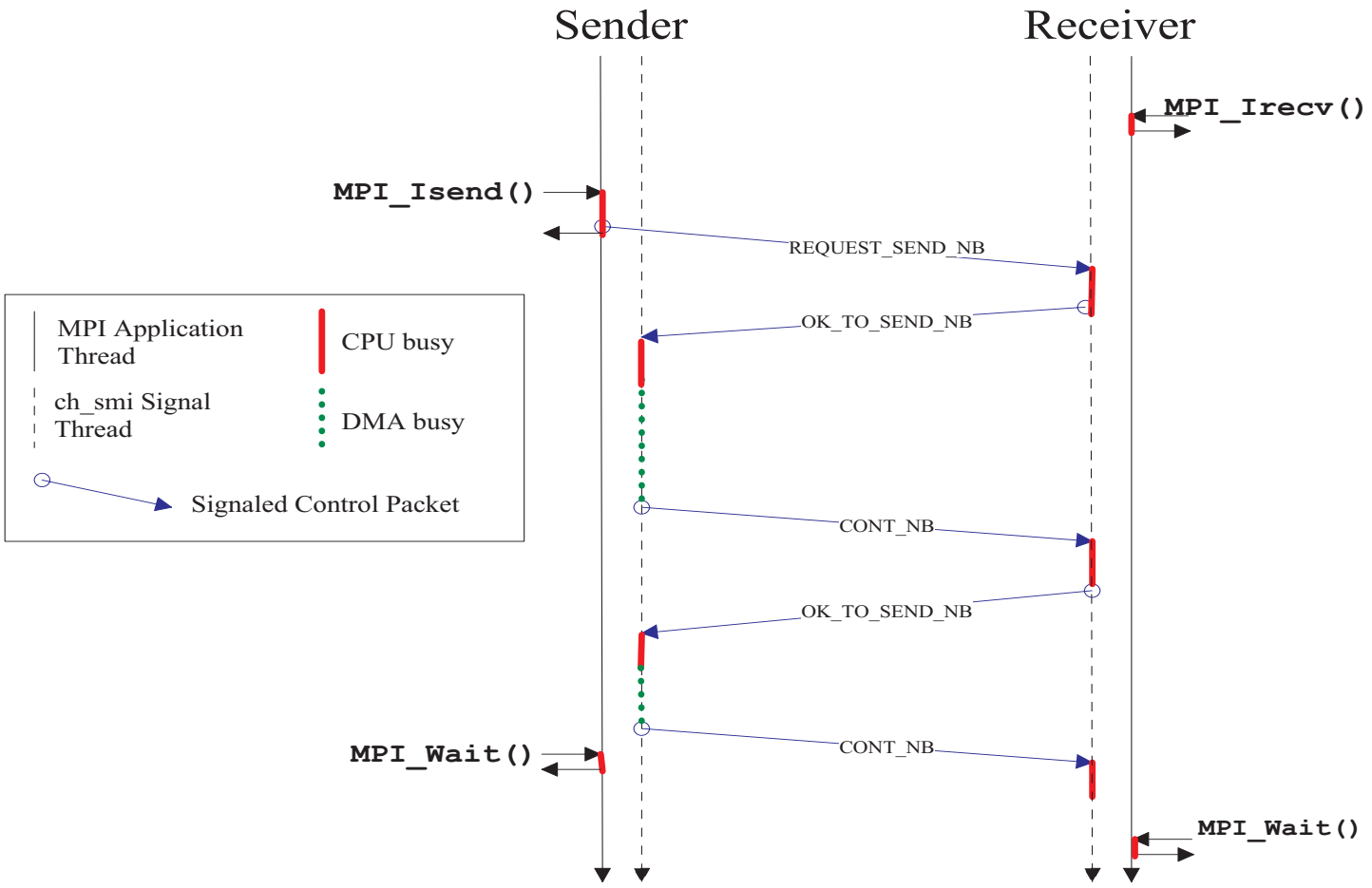
PIO vs. DMA: Latency



SYNCHRONOUS RENDEZ-VOUS

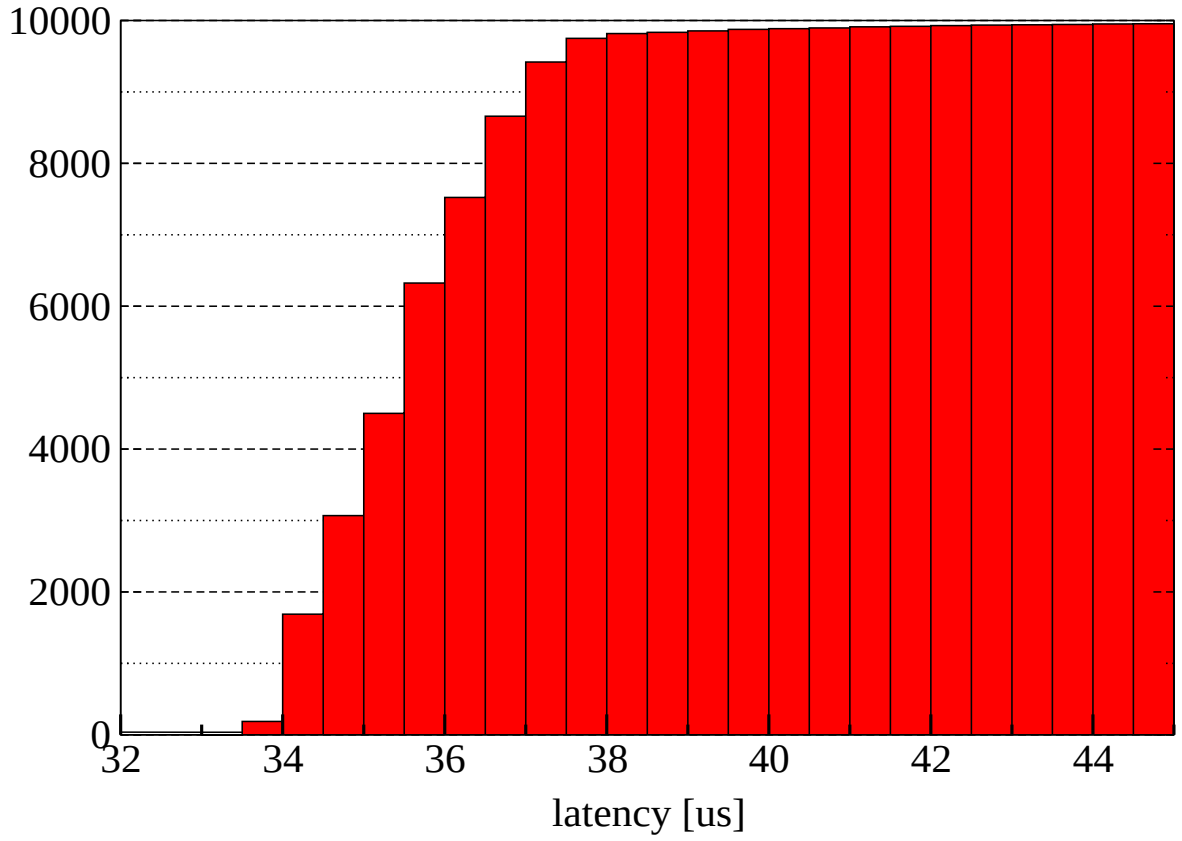


ASYNCHRONOUS RENDEZ-VOUS

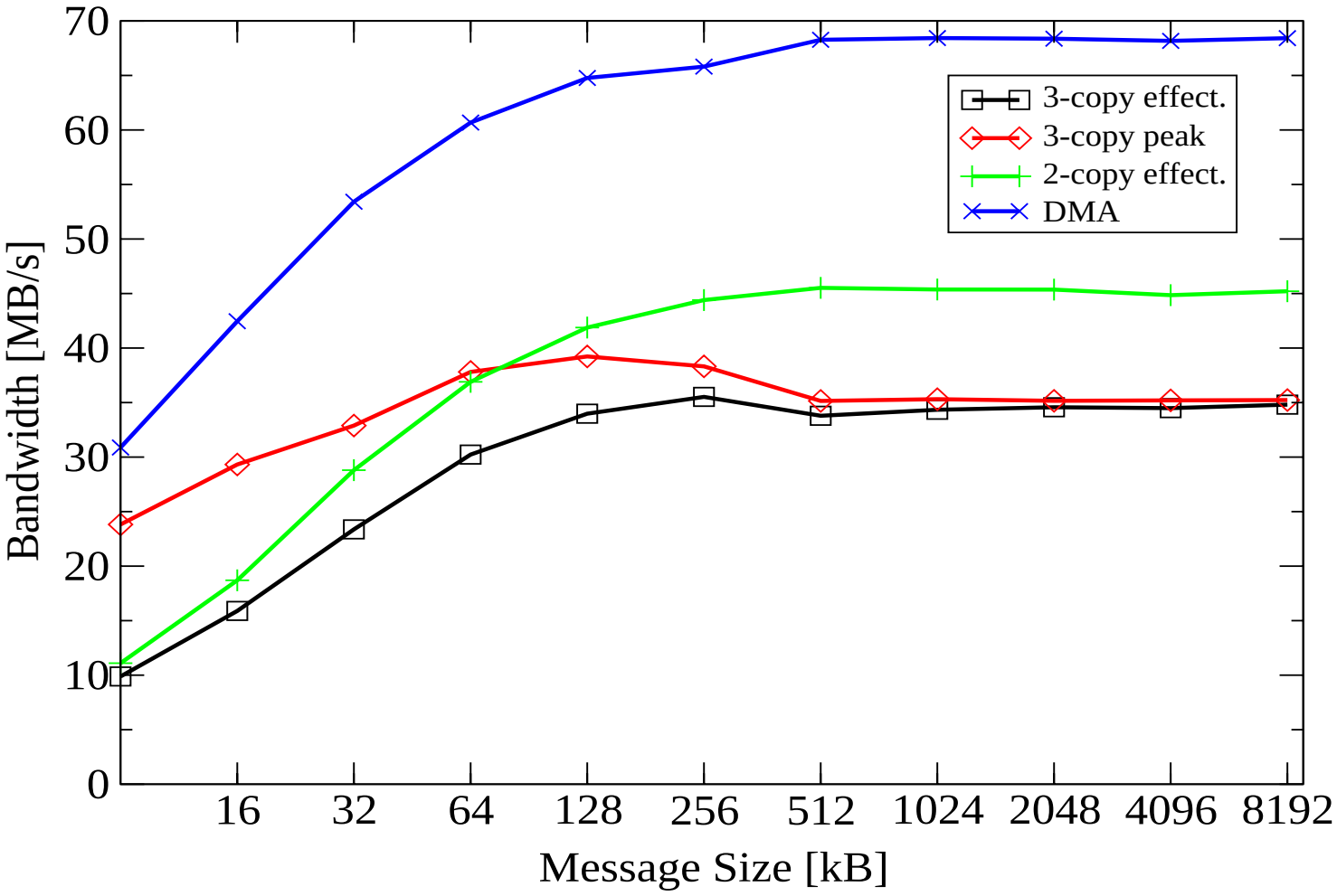


SCI REMOTE INTERRUPTS

Cumulative Histogramm of SCI Remote Interrupt Latency



PERFORMANCE



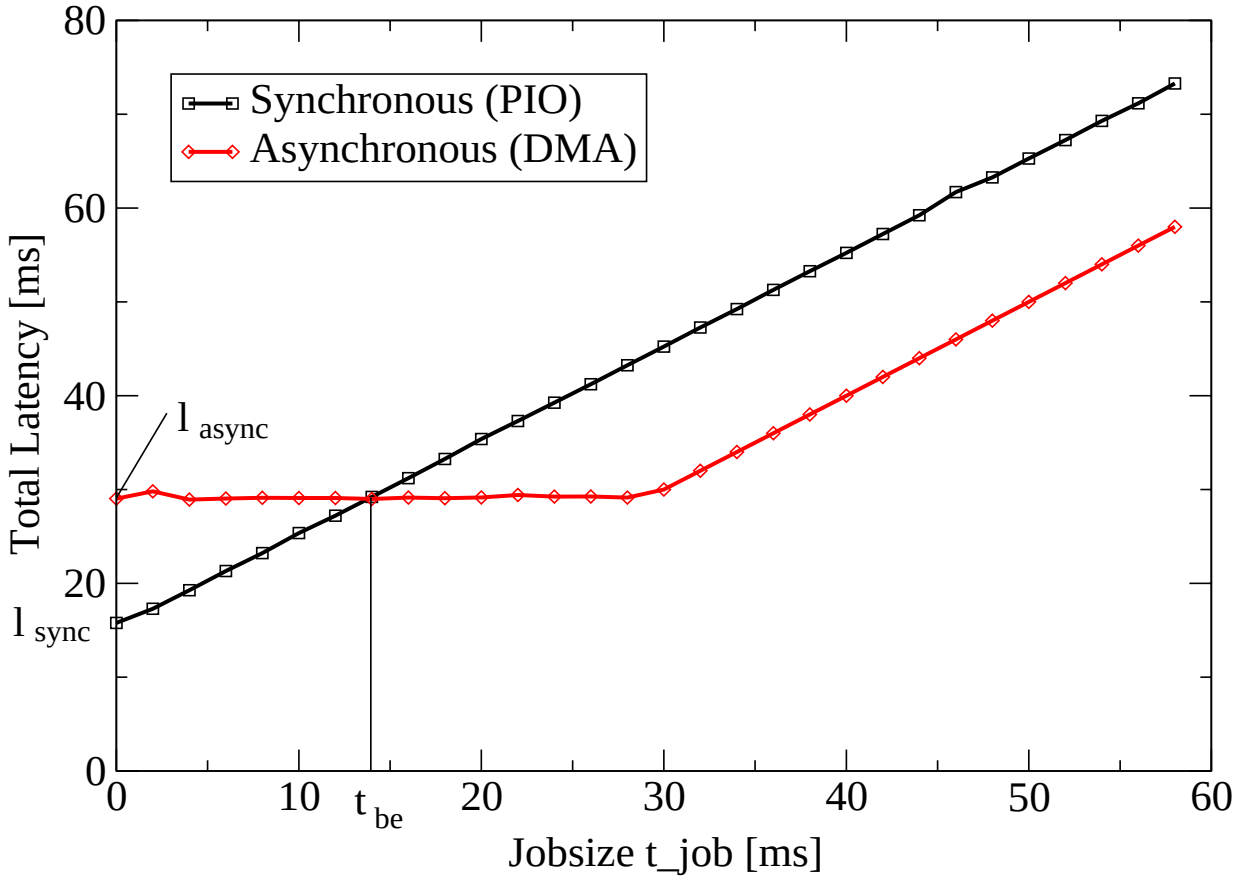
OVERLAPPING

- **Overlap**-benchmark to validate overlapping of communication & computation:

```
latency = MPI_Wtime()
if (sender)
  MPI_Isend(msg, msgsize)
  while (elapsed_time < jobsize)
    spin
  MPI_Wait()
else
  MPI_Recv()
latency = MPI_Wtime() - latency
```

OVERLAPPING

- **overlap** for a 1MB message and varying jobsize



$t_{be} = 14ms$

roughly equivalent to

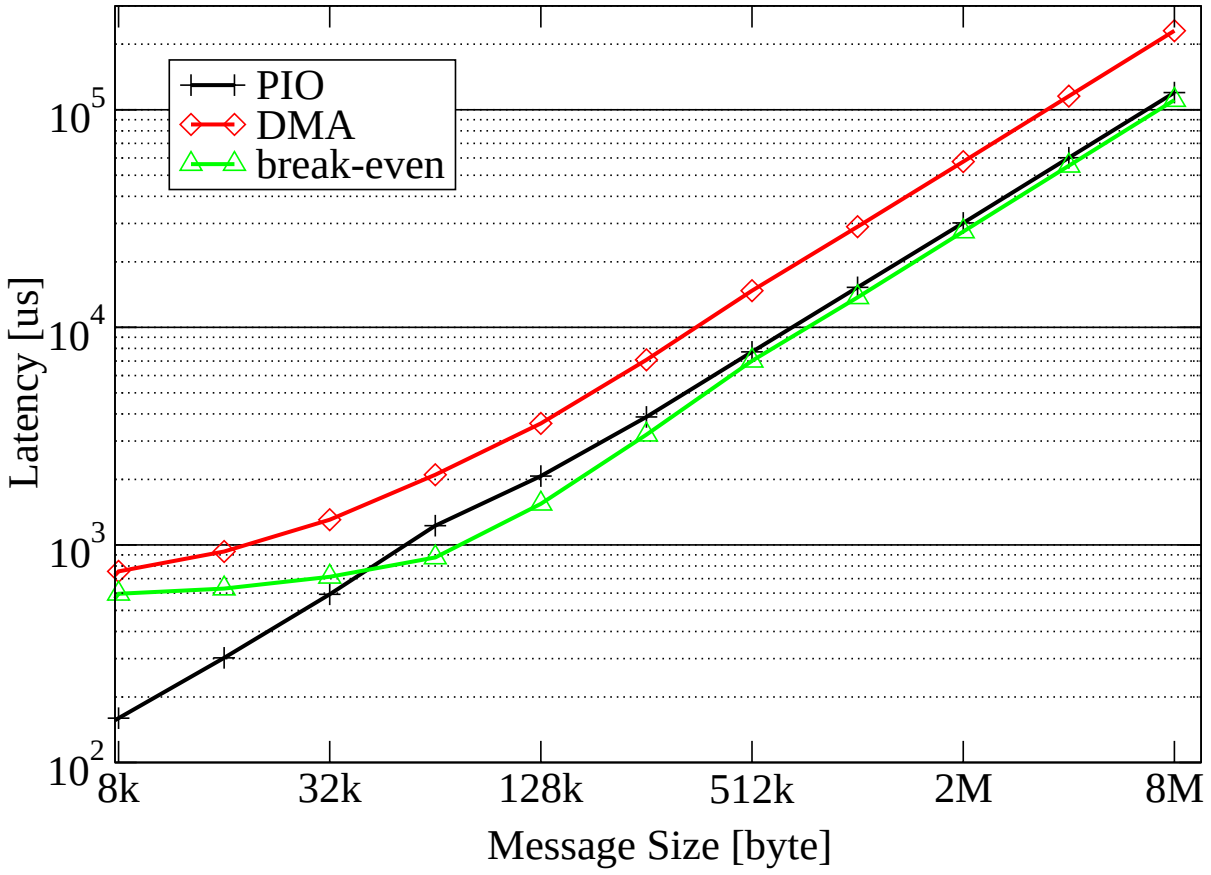
$10^6 \times A = A + C \times B$

$20 \times 10^3 \; A = A + \sin B$

operations on Pentium II @ 450MHz

BREAK-EVEN POINT

- Break-even-point for varying message size:

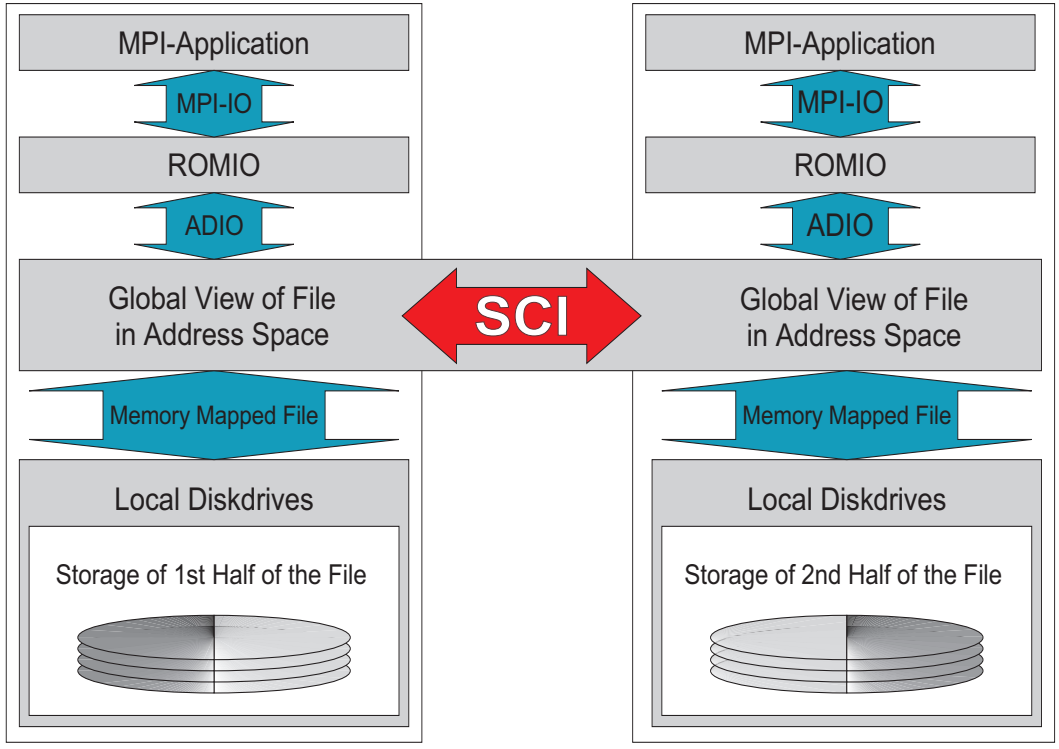


MPI-IO VIA SCI

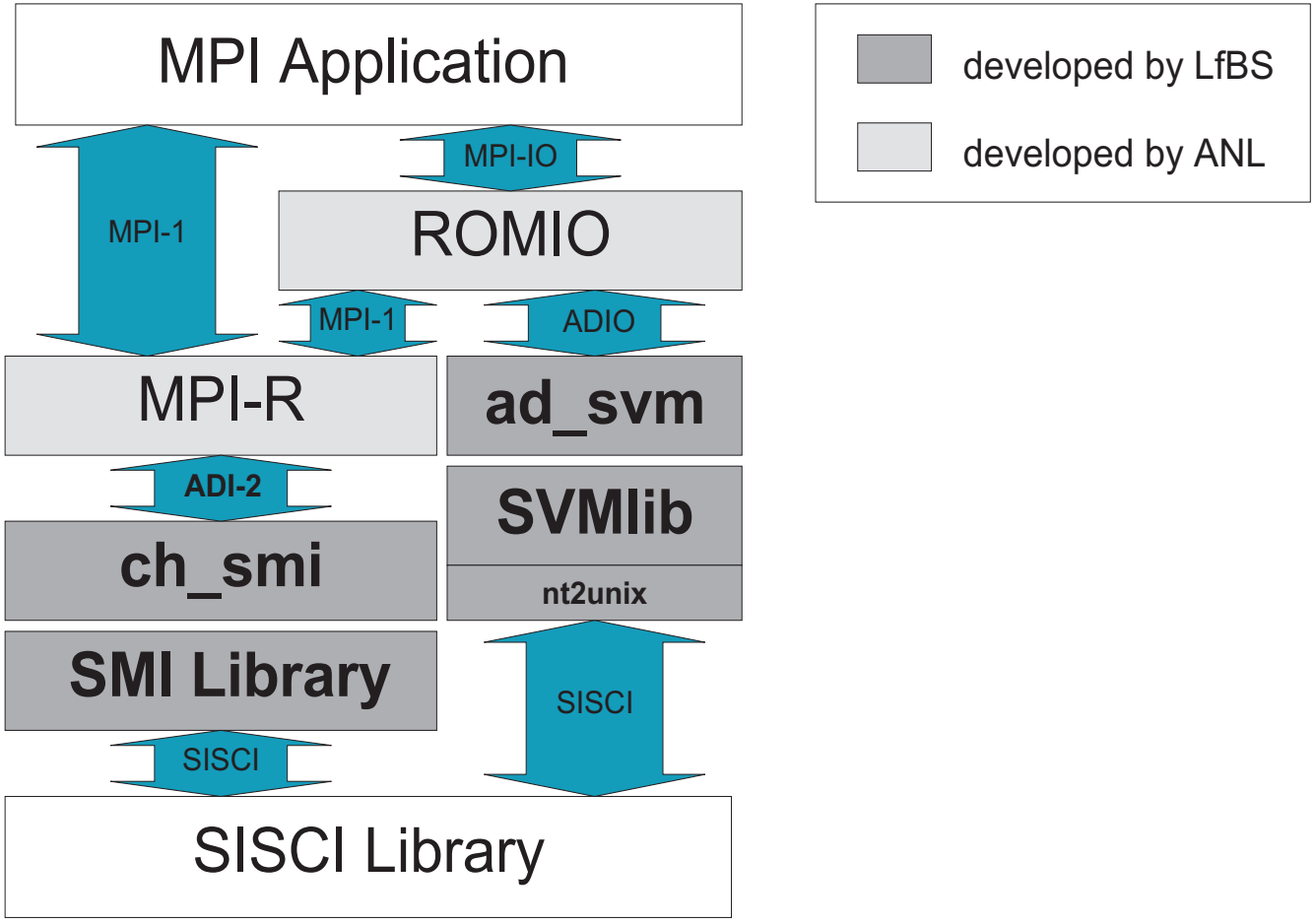
- Performance of I/O operations is critical for certain classes of MPI programms:
  - reading intial data
  - writing the results
  - checkpointing
  - out-of-core data sets
  - temporary files
  - visualization data
- I/O of MPI programs is inherently parallel, often collective → distributed storage layout for good performance
- Use of high-performance interconnect for peer-to-peer communication

DESIGN CONCEPT

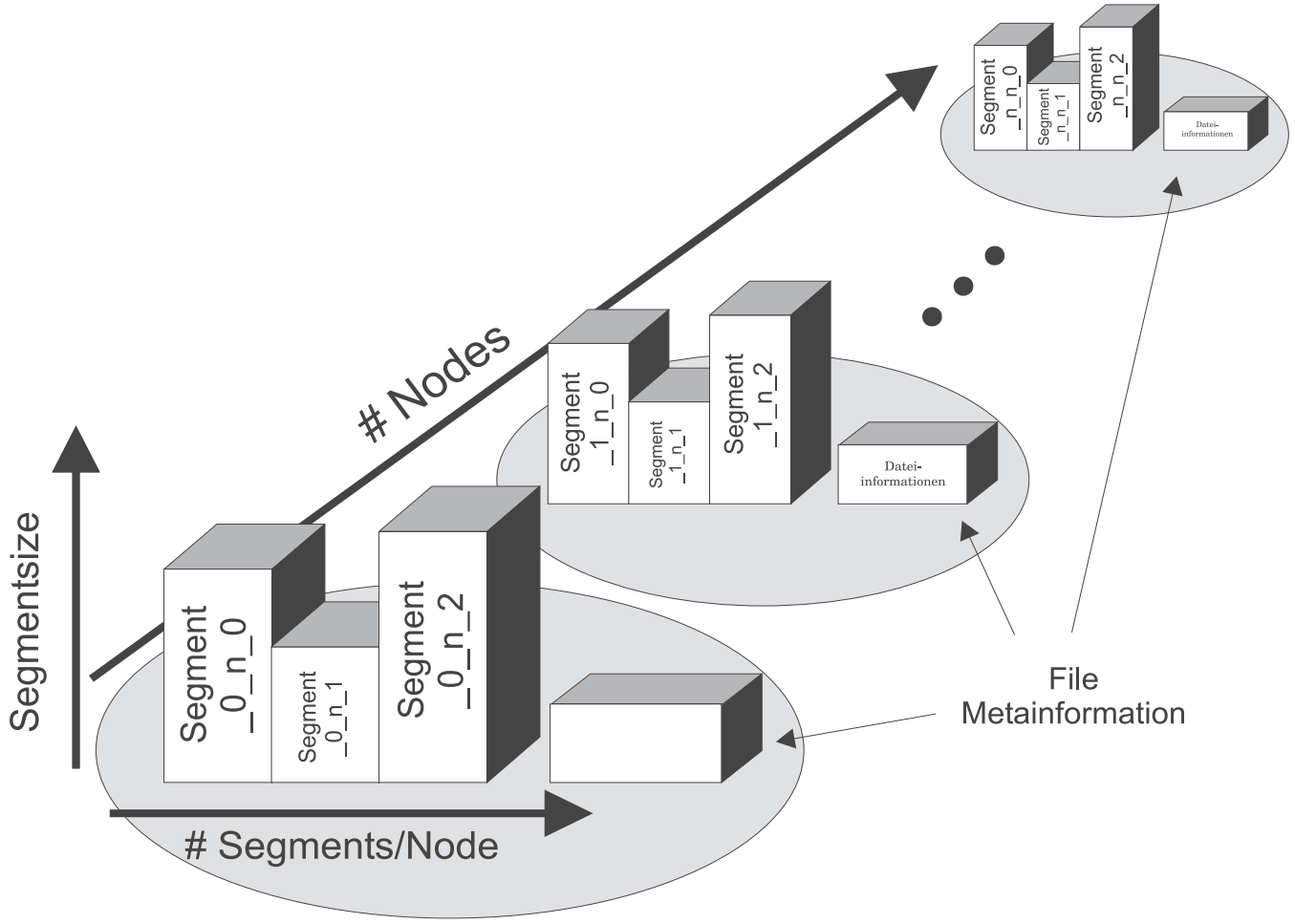
- Physically distributing the file according to I/O requests
- Providing a global view of the file
  - accessing remote data via SCI



PROTOTYPE IMPLEMENTATION

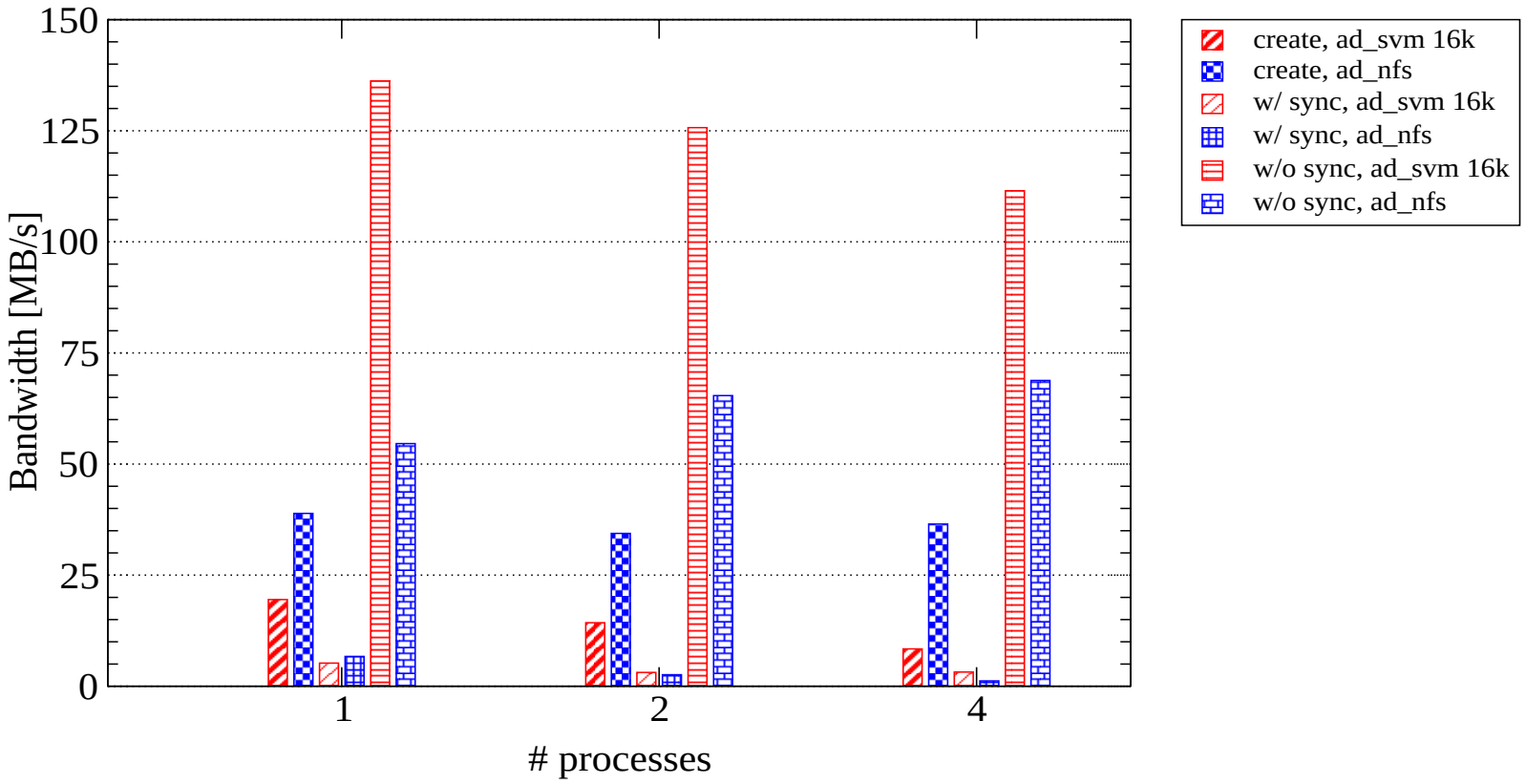


DATA DISTRIBUTION



PERFORMANCE

- **perf** Benchmark: non-collective write access
  - 64 MB per process, min. per-process bandwidth



SUMMARY

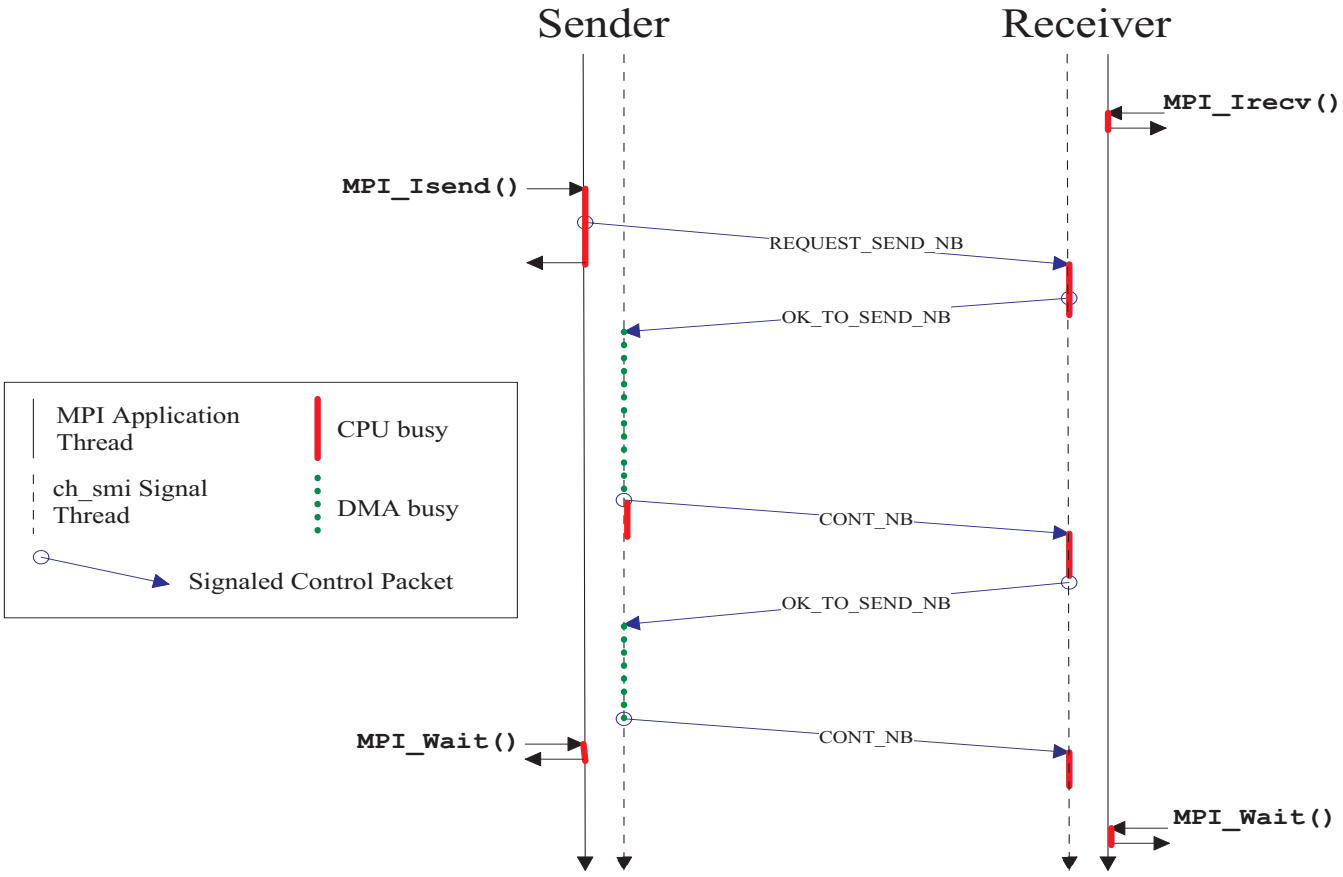
- SCI-MPICH has become
  - **more compatible**
    (Linux, Solaris, NT / Sparc, x86 / Dolphin, Scali)
  - **more stable**
    (passes all MPICH suites, PMB, and other tests)
  - **more flexible**
    (DMA transfers, run-time configurable)
  - **easier to use**
    (improved startup, watchdog)
- More efficient support for DMA via SCI is needed
- Few MPI-Programmers make use of overlapping computation & communication
- MPI-IO via SCI has high performance potential

ONGOING DEVELOPMENT

- MPI-IO
  - store distribution information in SCI shared memory
  - adapt MPI-IO layer to ADIO device characteristics
- Single-Sided Communication
  - important part of MPI-2 specification
- Multi-Adapter Support
  - use of multiple PCI-SCI adapters to increase performance
- Communication- and Platform-Heterogenity
  - use of multiple, independant communication devices
  - running applications across different operating systems
  - coupling of mulitple MPI clusters („The Grid“)

ASYNCHRONOUS RENDEZ-VOUS

- Protocol version with DMA-precopy:



FILESYSTEM INFRASTRUCTURE

- internal and external access to distributed files
- redundant storage of files for increased reliability

I/O System for Dedicated MPI-Cluster with SCI Interconnect