Netgauge: A Network Performance Measurement Framework

T. Hoefler[1,2], T. Mehlan[2], A. Lumsdaine[1], W. Rehm[2]

[1]Open Systems Lab
Indiana University
Bloomington, USA

[2]Computer Architecture Group
Technical University of Chemnitz
Chemnitz, Germany

High Performance Computation Conference 2007
Houston, TX, USA
28th September 2007

Network Performance Measurement - Why?

HPC Wire

"The only genuinely objective benchmark is the one left on a person's trousers when they sit on a bench that has just been painted."

- vendors: present good numbers to customers
- customers: get the real numbers
- find bottlenecks in networks
- analyze communication protocols/overheads
- gain a better understanding of networks
- parametrize network models

There are dozens of benchmarks, why a new one?

Kevin McCurley

"There are lies, damn lies, and benchmarks."

- missing portability and comparability of many tools
- need a single tool with many "patterns" and "protocols"
- measurement methods often questionable (i.e., measuring 1000 messages and dividing by 1000 - outlier&pipelining issues)
- most tools measure only RTT
- parametrize network models at different layers
- ...

Contributions

Antoine de Saint-Exupery

“A designer knows he has arrived at perfection not when there is no longer anything to add, but when there is no longer anything to take away.”

- simple, extensible framework
- abstract interface definition to communication modules
- one- and two-sided protocol support
- combine efforts of algorithm designers (patterns, models) and hardware designers/vendors (protocol support)
- high-precision timing interface (macro)
- support for many networks and several example patterns

Challenges

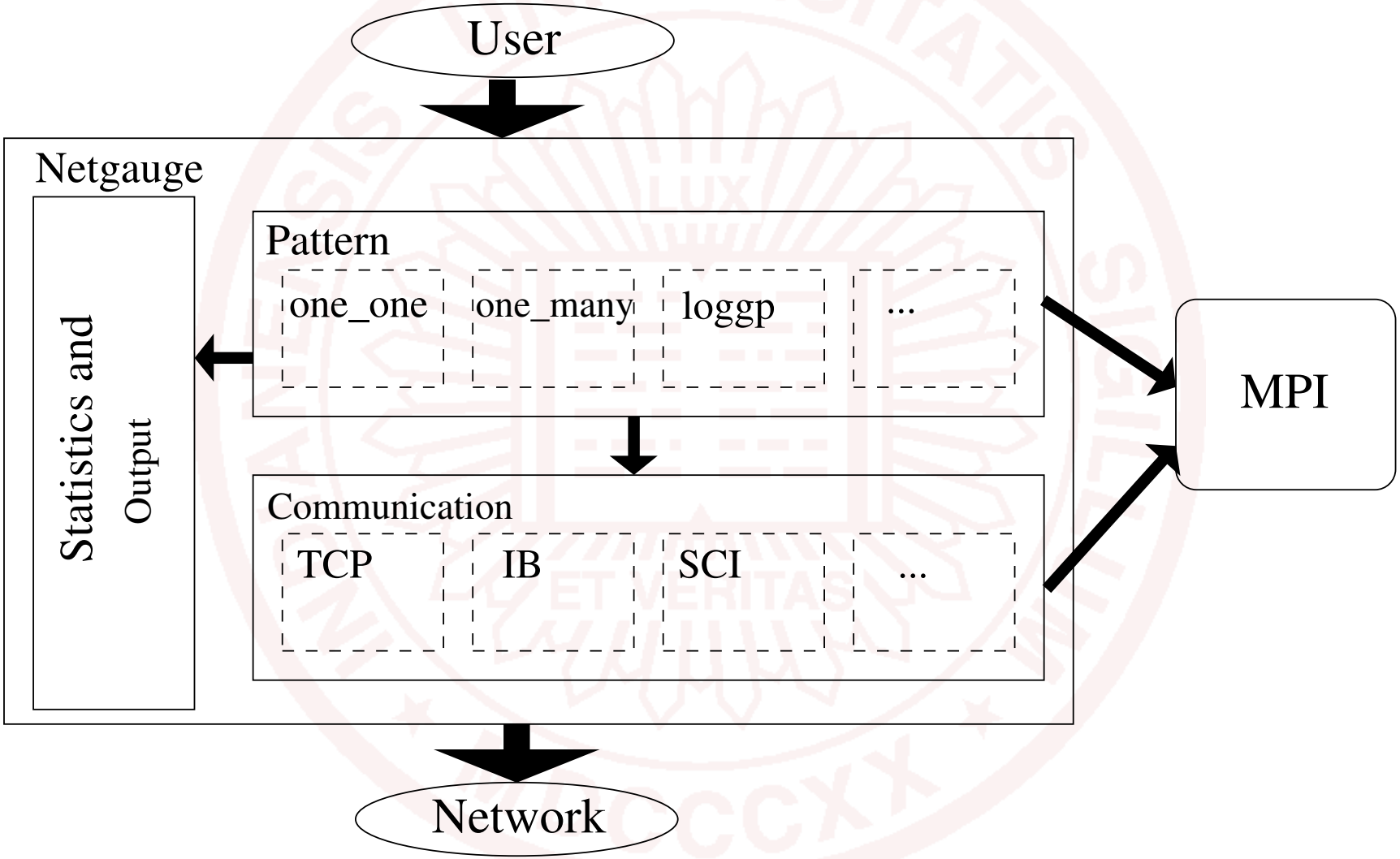
Paul Erdös

"I hope we'll be able to solve these problems before we leave. And when I say "before we leave", I mean "before we die.""

- design a single interface between communication and pattern layer (one- and two-sided)
- unify all network types (terms of reliability, memory pinning)
- keep protocol as simple as possible (e.g., no tags)
- pattern must be able to reflect applications
- portability
- accurate timing (single messages)

The Netgauge Framework

- uses a component architecture (cf. Open MPI, Lam/MPI)



The Pattern Framework

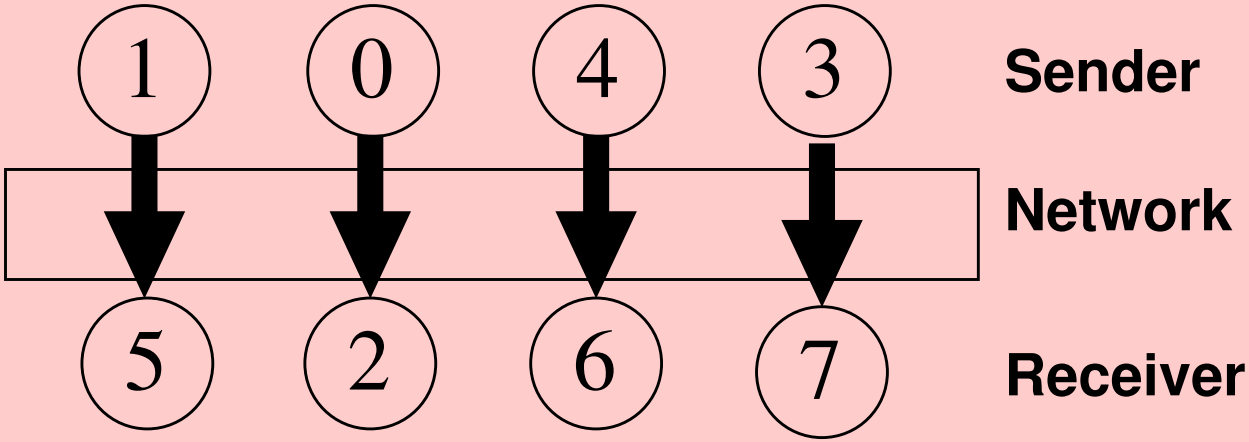
- core component of every benchmark
- implements the benchmark logic
- user parameters through command line
- may define needed capabilities of communication modules
- the Netgauge framework calls the pattern's benchmark function and passes a reference to an initialized communication module

The Communication Framework

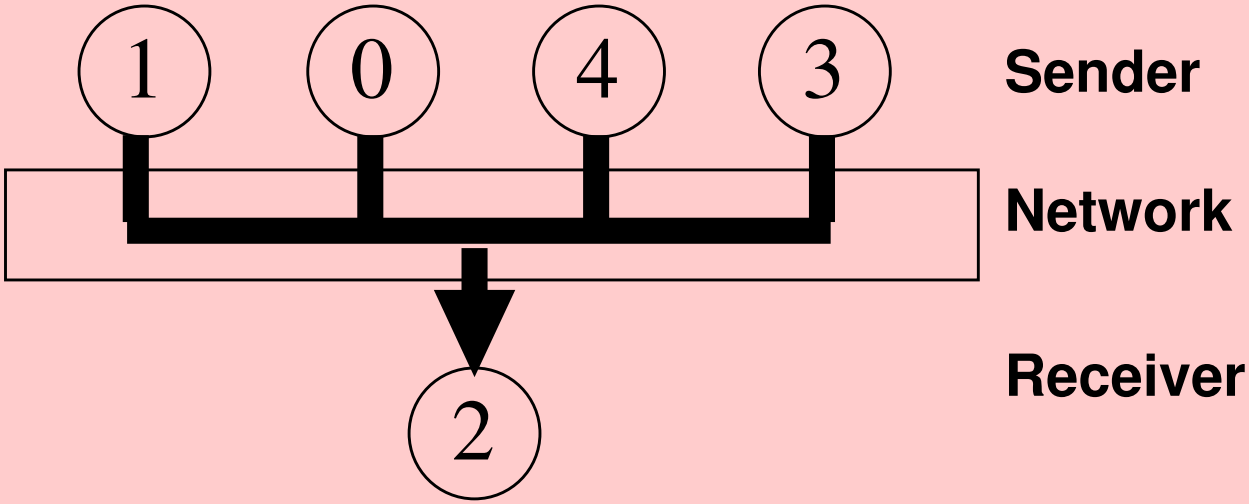
- defines interface to communication modules
- elements:
  - name (mnemonic string)
  - maximum message size (e.g., UDP)
  - additional header bytes (e.g., Raw ETH)
  - flags (reliable, channel semantics, memory registration)
  - init(), shutdown(), getopt() - optional
  - sendto(), recvfrom() - mandatory
  - isendto(), irecvfrom(), test() - optional, recommended

Simple Communication Patterns

1:1 communication



1:n, n:1 communication



Advanced Communication Patterns

pLogP

Kielmann et al. *"Fast Measurement of LogP Parameters for Message Passing Platforms"*

- uses scheme proposed in the paper to measure $o_s(s)$, $o_r(s)$ directly

LogGP

Alexandrov et. al. *"LogGP: Incorporating Long Messages into the LogP Model"*

- uses scheme described in Hoefler et al. *"Low-Overhead LogGP Parameter Assessment for Modern Interconnection Networks"*

Communication Modules

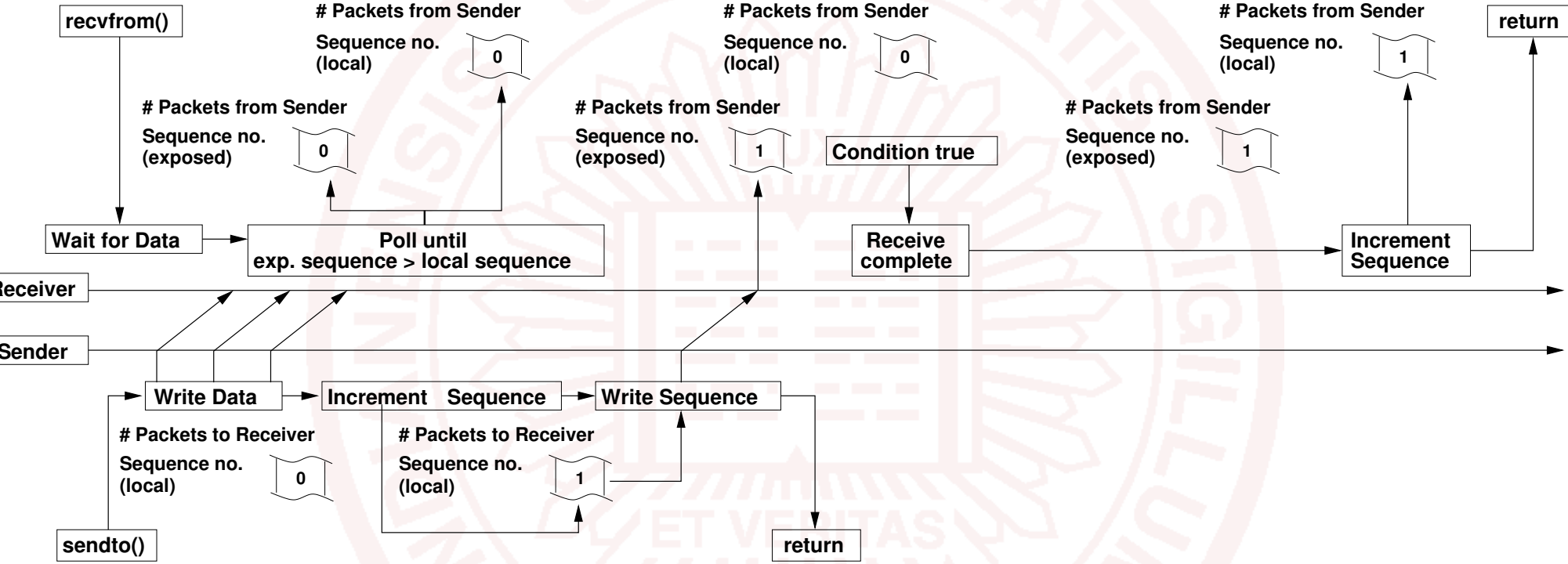
Two-sided Communication Modules

- MPI (blocking and non-blocking)
- Socket Based (UDP, TCP, ETH, EDP, ESP)
- Myrinet/GM (blocking)
- InfiniBand

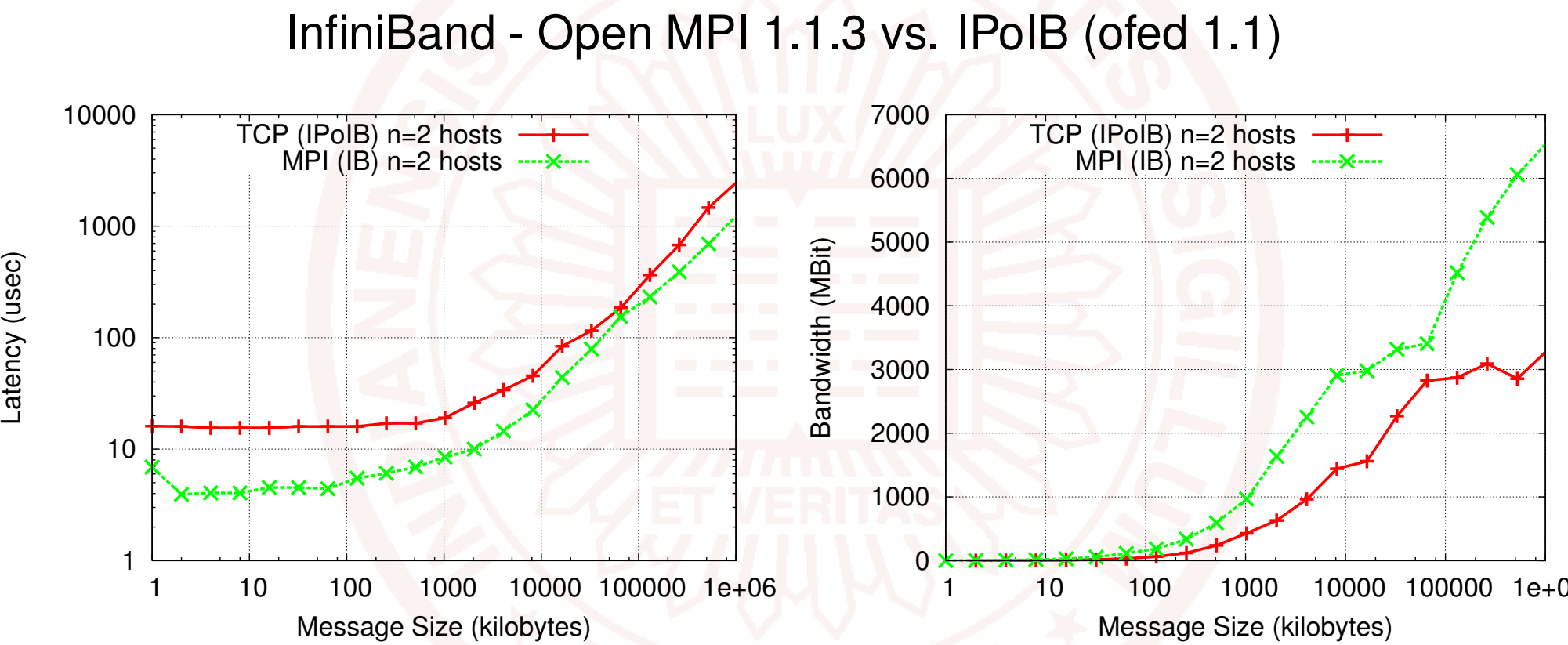
One-sided Communication Modules

- ARMCI (using ARMCI_Put())
- MPI-2 One Sided
- Scalable Coherent Interface (SCI)

Mapping One-sided to Two-sided

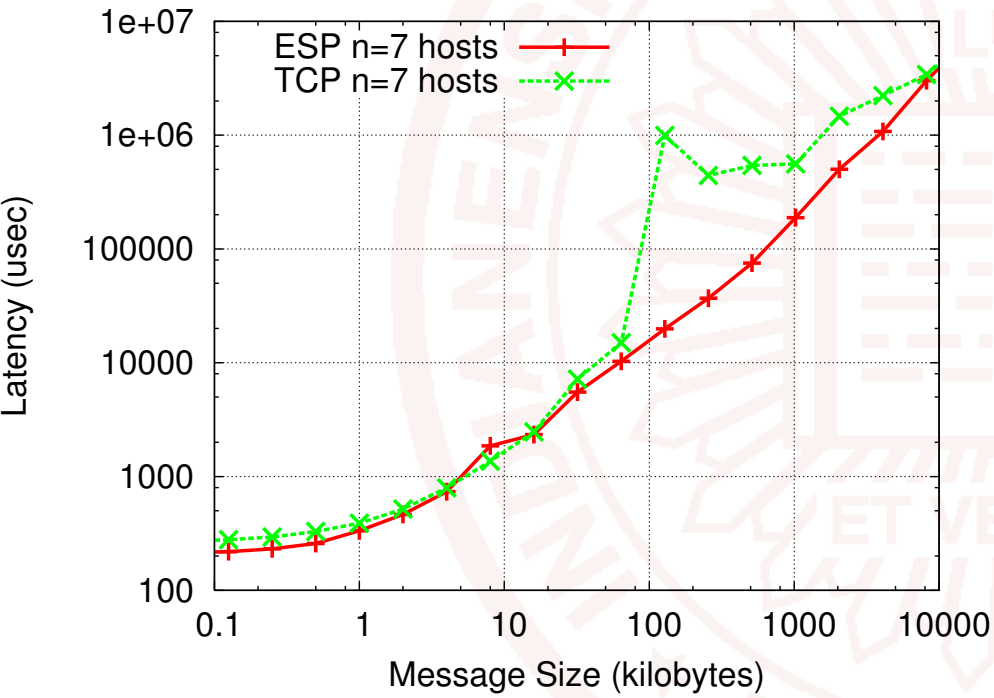


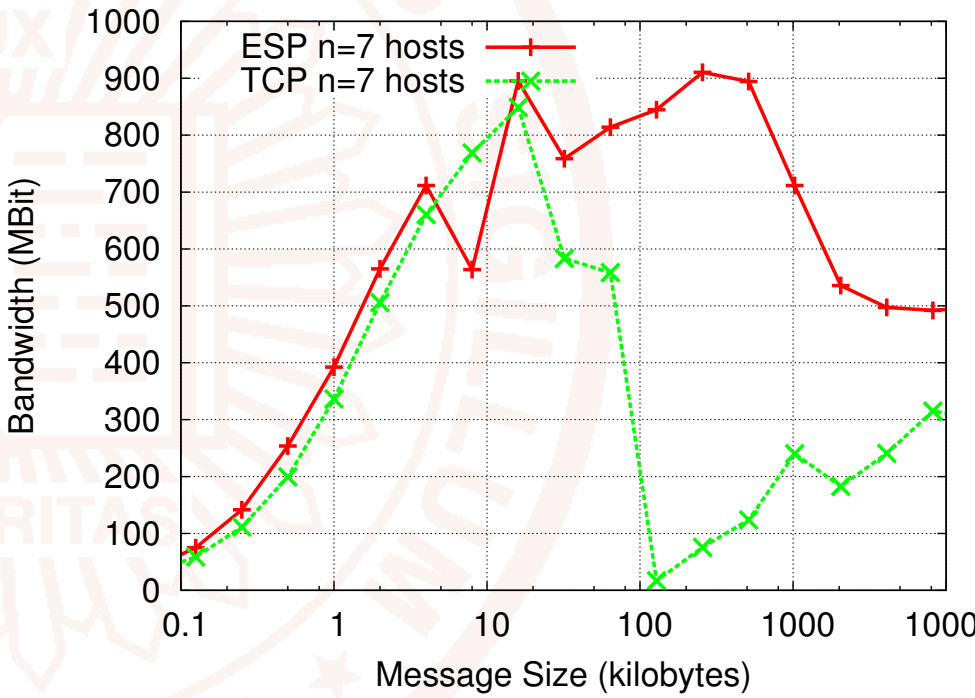
Benchmark Results - 1:1



Benchmark Results - 1:n

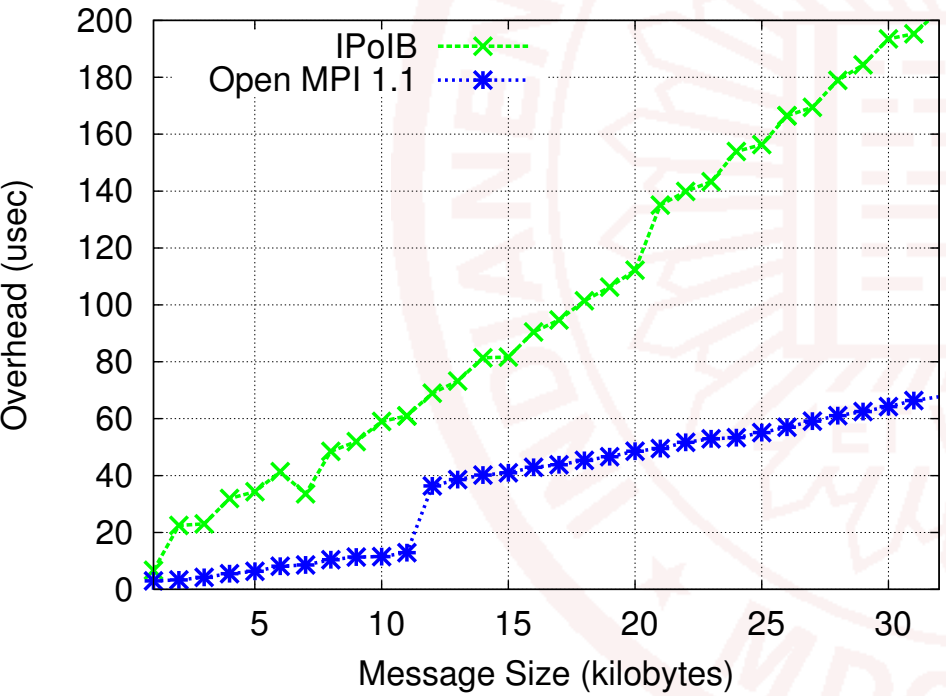
1:7 communication - TCP (Linux 2.6, Reno) vs. ESP

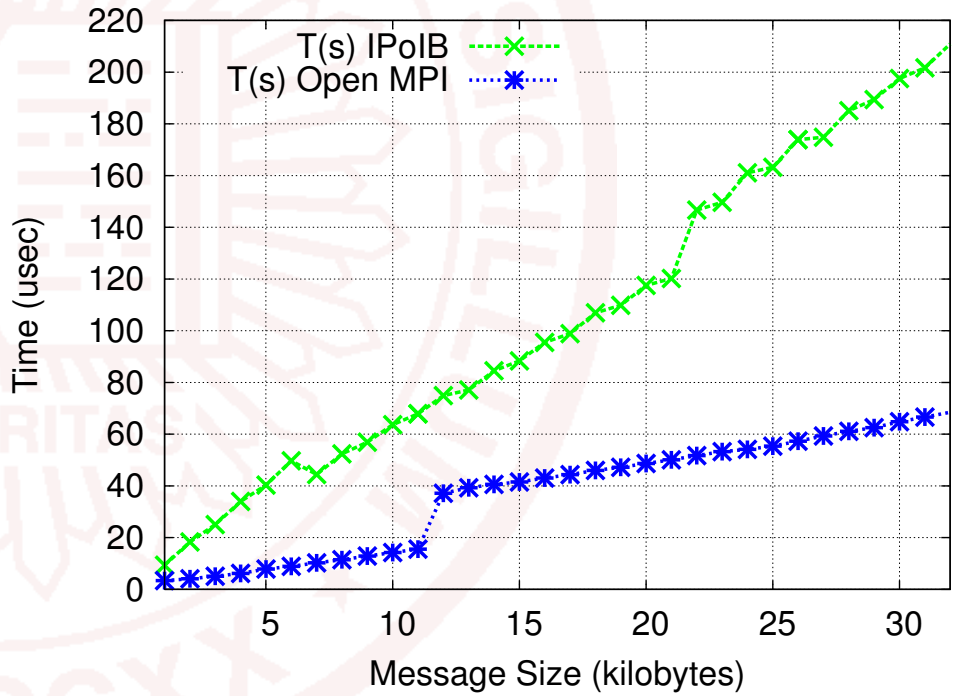




Benchmark Results - LogGP

LogGP overhead and g,G curves
Open MPI 1.1.3 ($g = 19.75, G = 0.0016$) vs.
IPoIB (ofed 1.1, $g = 7.79, G = 0.0061$)





Conclusions and Future Work

Conclusions

- easy to use and extend
- enables complex communication patterns
- large number of supported protocols

Future Work:

- addition of new communication modules
- application-specific communication pattern
- $\Rightarrow$ We would like to collaborate with scientists!

Download/Further Information

http://www.unixer.de/research/netgauge

Conclusions and Future Work

Conclusions

- easy to use and extend
- enables complex communication patterns
- large number of supported protocols

Future Work:

- addition of new communication modules
- application-specific communication pattern
- $\Rightarrow$ We would like to collaborate with scientists!

Download/Further Information

http://www.unixer.de/research/netgauge

Conclusions and Future Work

Conclusions

- easy to use and extend
- enables complex communication patterns
- large number of supported protocols

Future Work:

- addition of new communication modules
- application-specific communication pattern
- $\Rightarrow$ We would like to collaborate with scientists!

Download/Further Information

`http://www.unixer.de/research/netgauge`