Performance and Analysis of the PCI-DDC Remote-Write Implementation

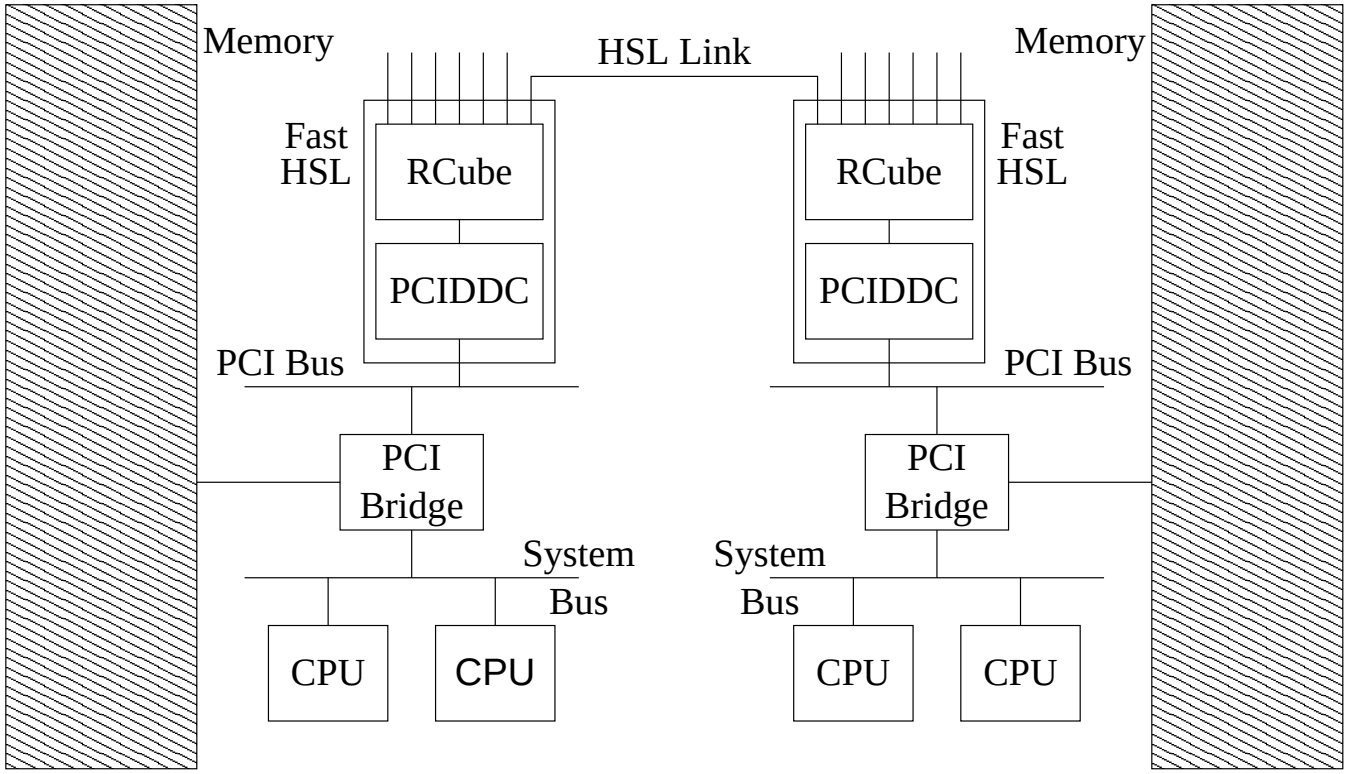
Éric Renault and Pierre David

PR*i*SM / UVSQ

Cluster Computing 2001 — October 10, 2001

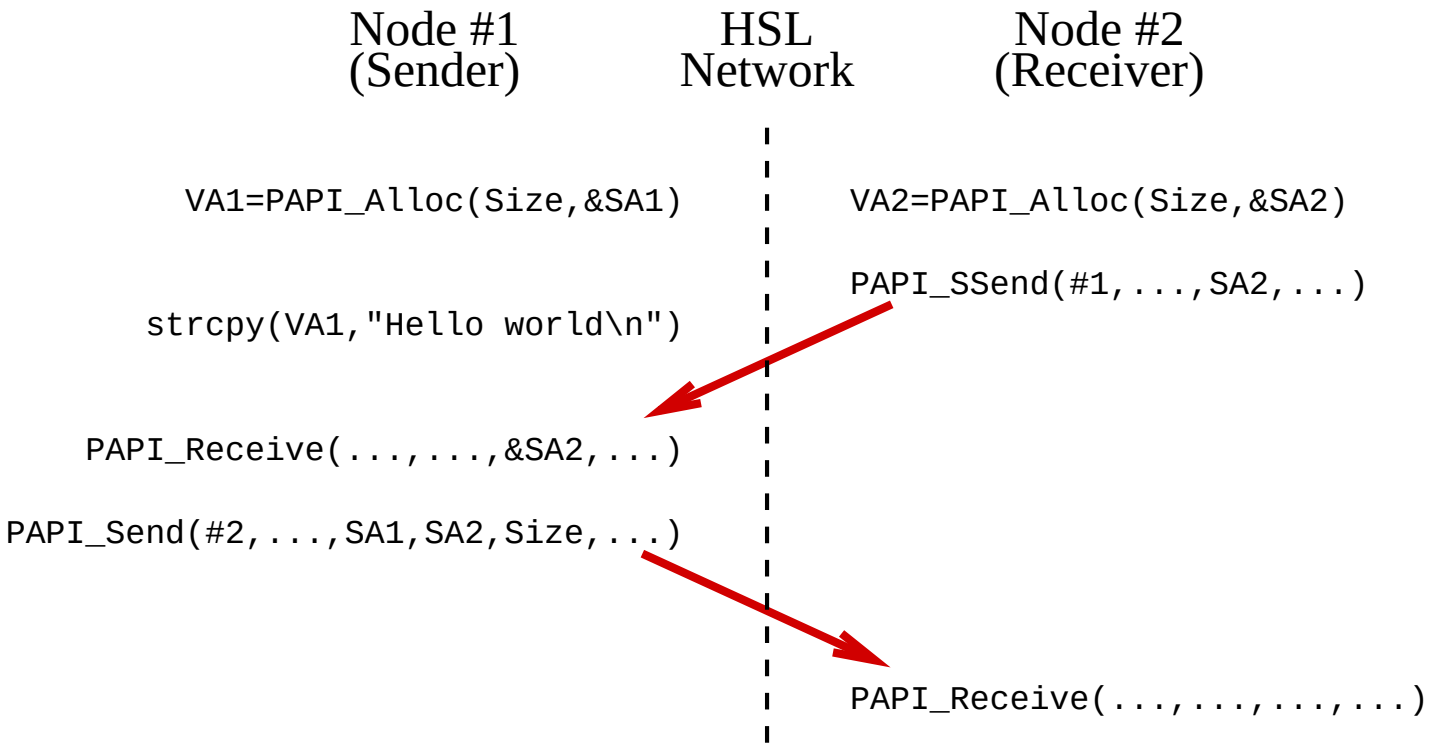
Contents

Multi-PC Machine



Two kinds of messages: “short” and “normal”.

Programming Model



Programming Interface

Network:

- `PAPI_Spread`
- `PAPI_Start`
- `PAPI_Stop`

Memory:

- `PAPI_Alloc`
- `PAPI_Free`

Topology:

- `PAPI_Name`
- `PAPI_Node`
- `PAPI_Width`

Barrier:

- `PAPI_GBarrier`
- `PAPI_SBarrier`

Short messages:

- `PAPI_[M]SSend`

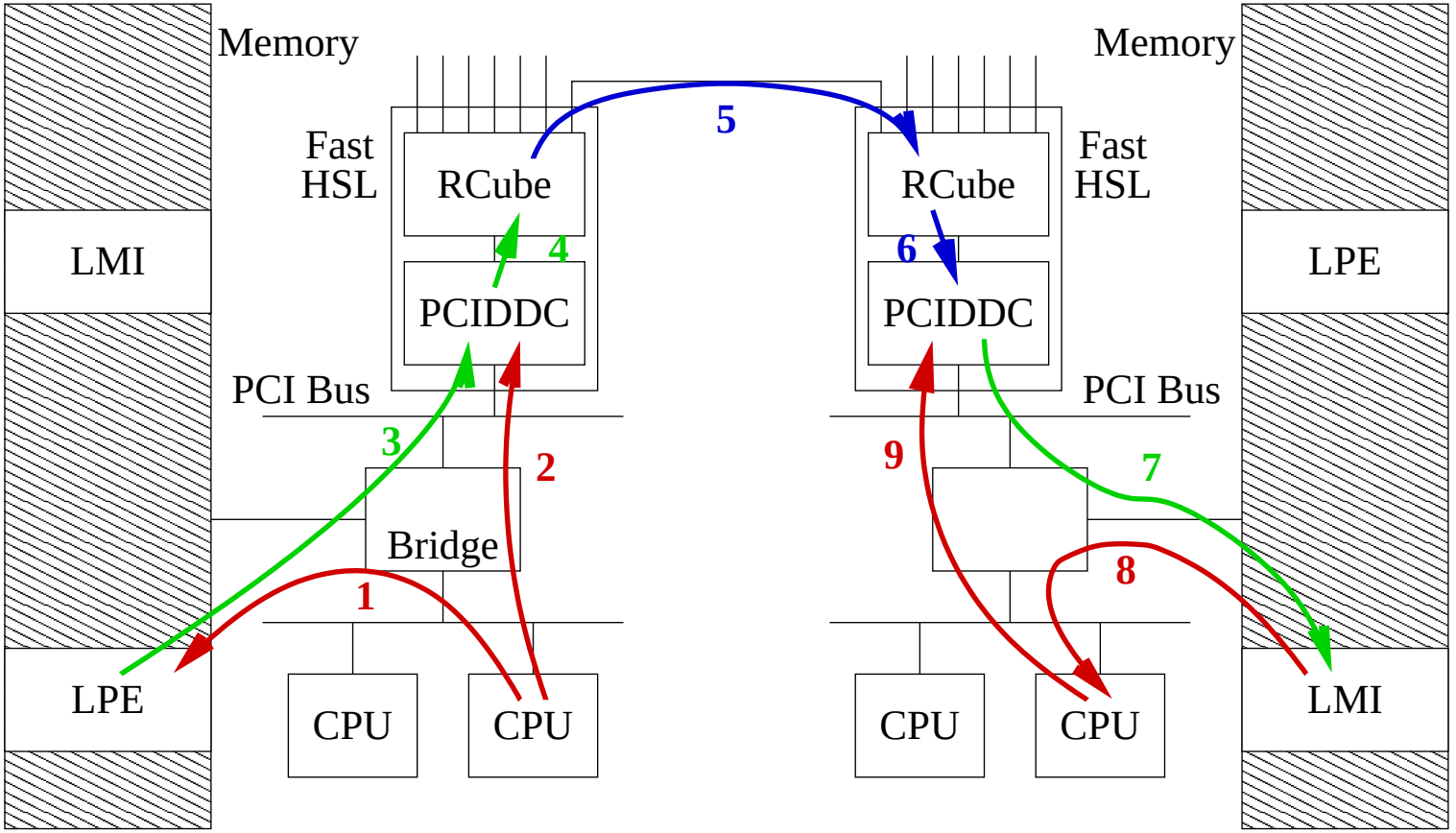
Normal messages:

- `PAPI_[M]Send`

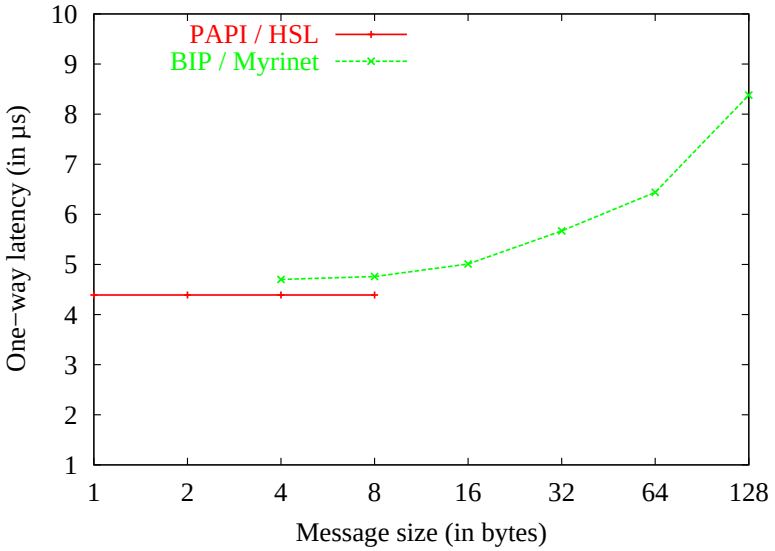
Acknowledgement:

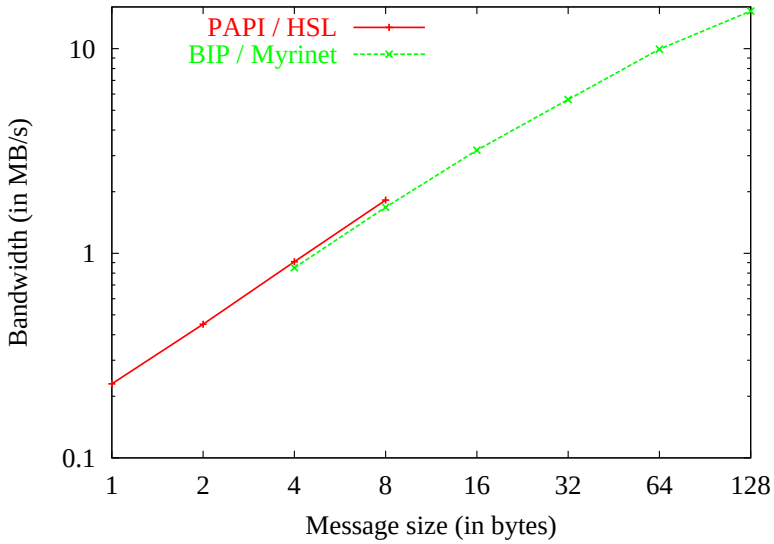
- `PAPI_[M]Receive`

Short Message Transfer



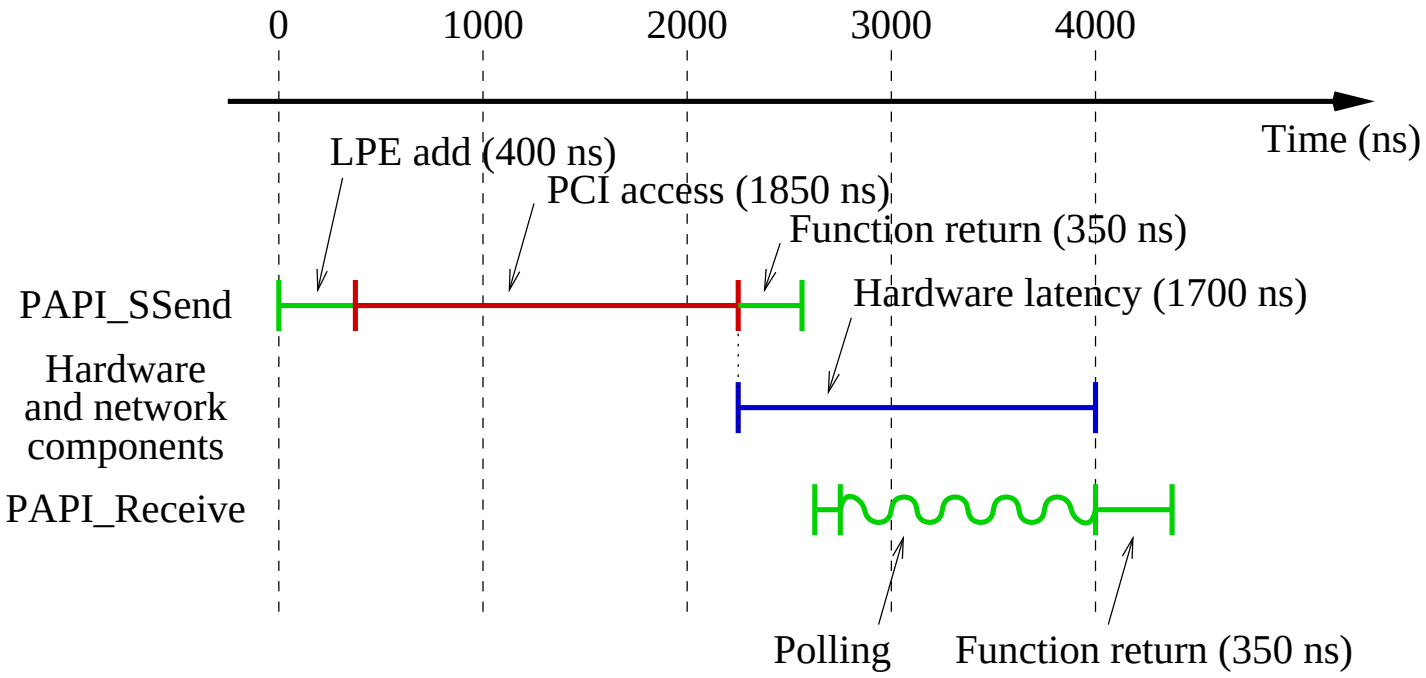
Performance in User-Level Messaging





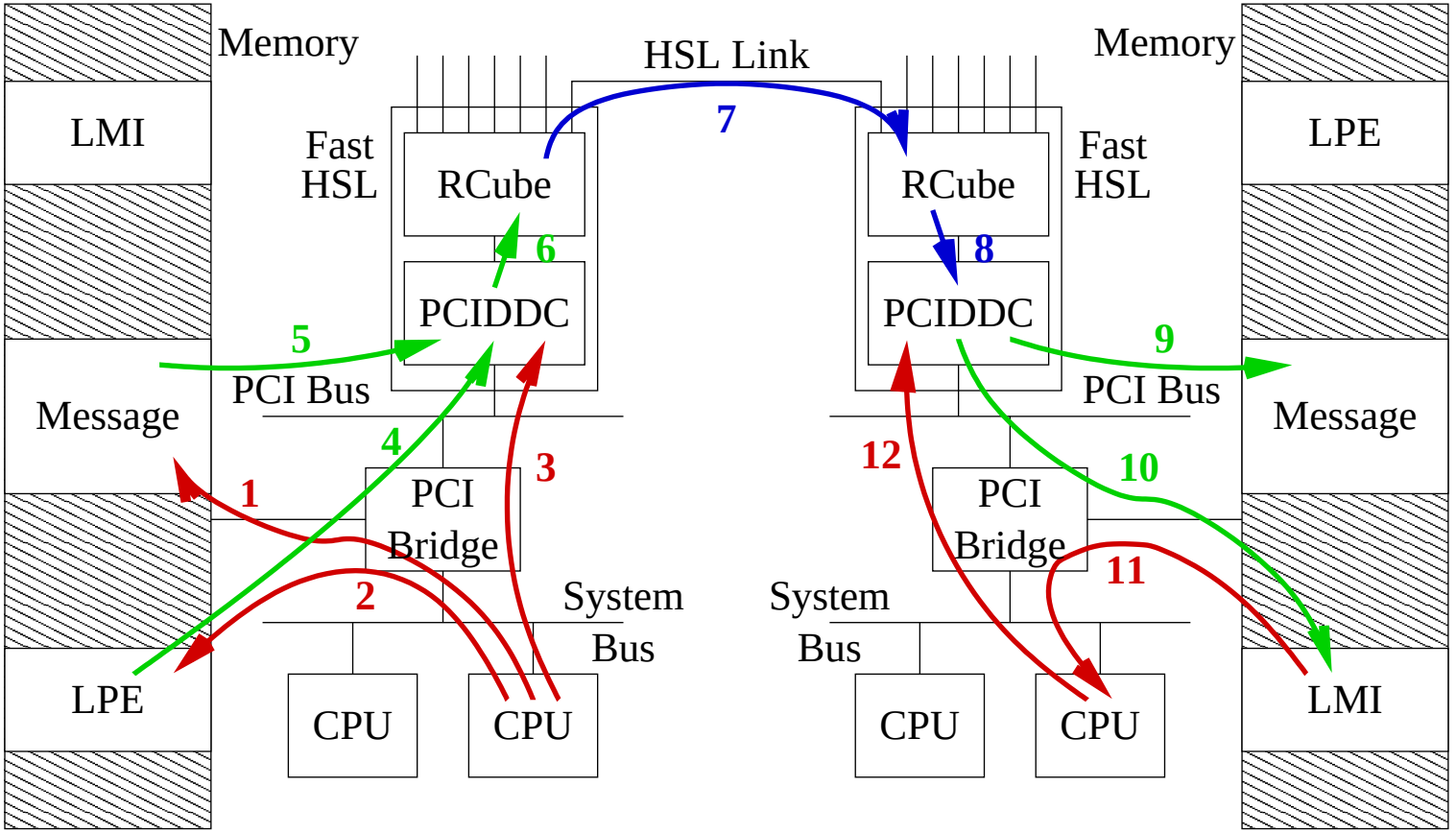
- PAPI : Hardware limit (Fast-HSL) to 8 bytes
- BIP : Software limit between 100 and 400 bytes (the maximum size for short messages is left to the user)

Time Diagram

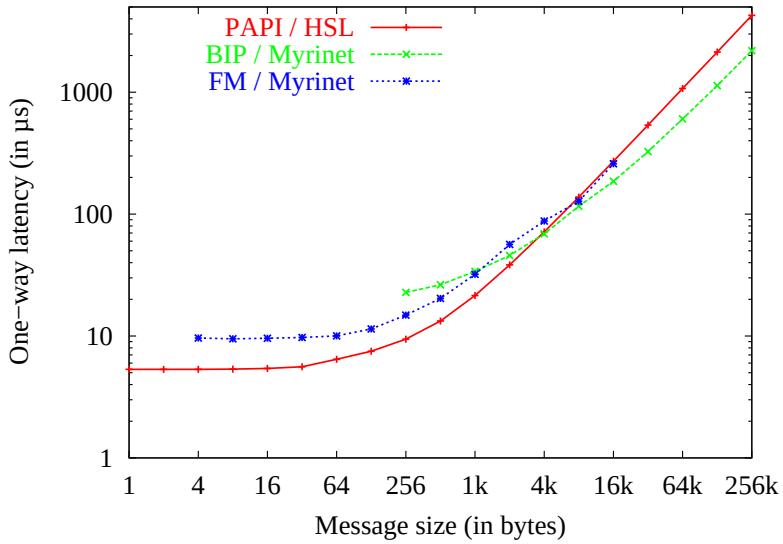


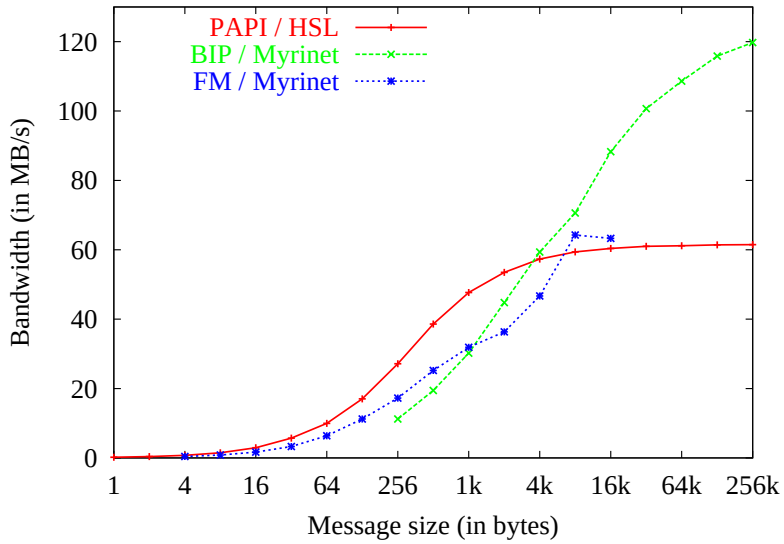
- PAPI / HSL overhead: 750 ns, i.e. 17.4%
- PCI access overhead: 1.85 μs, i.e. 43%

Normal Message Transfer



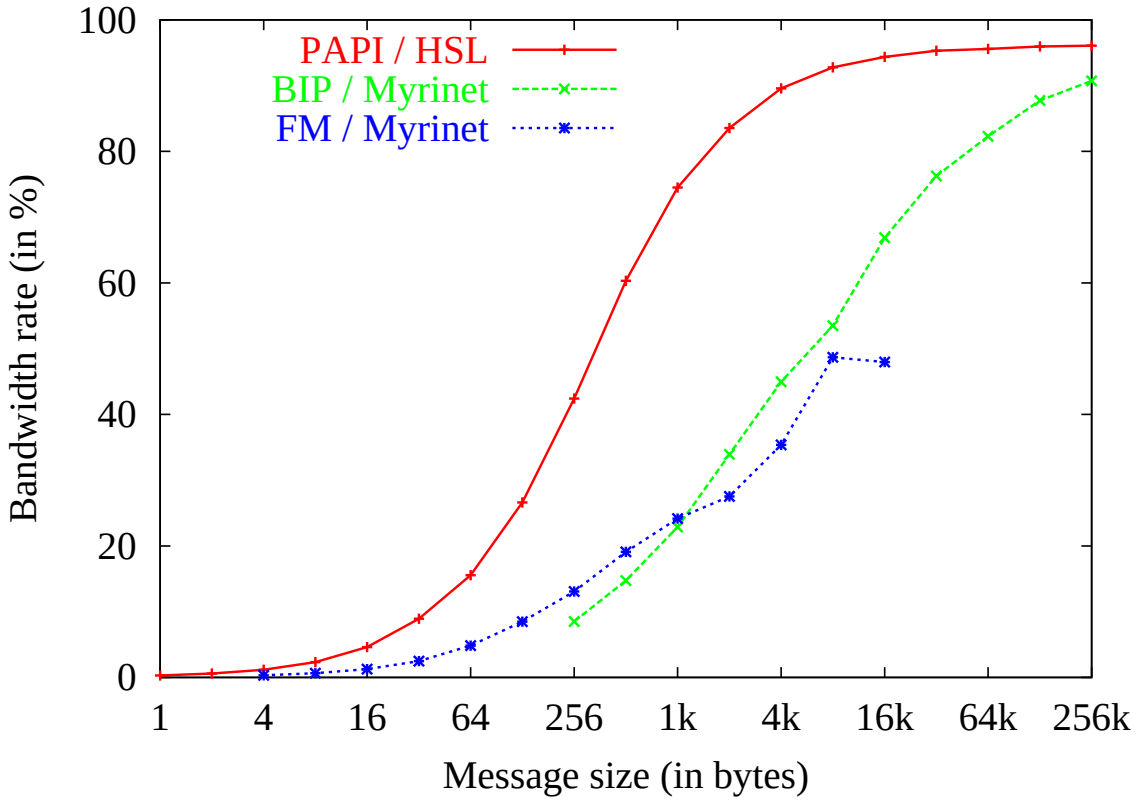
Performance in User-Level Messaging





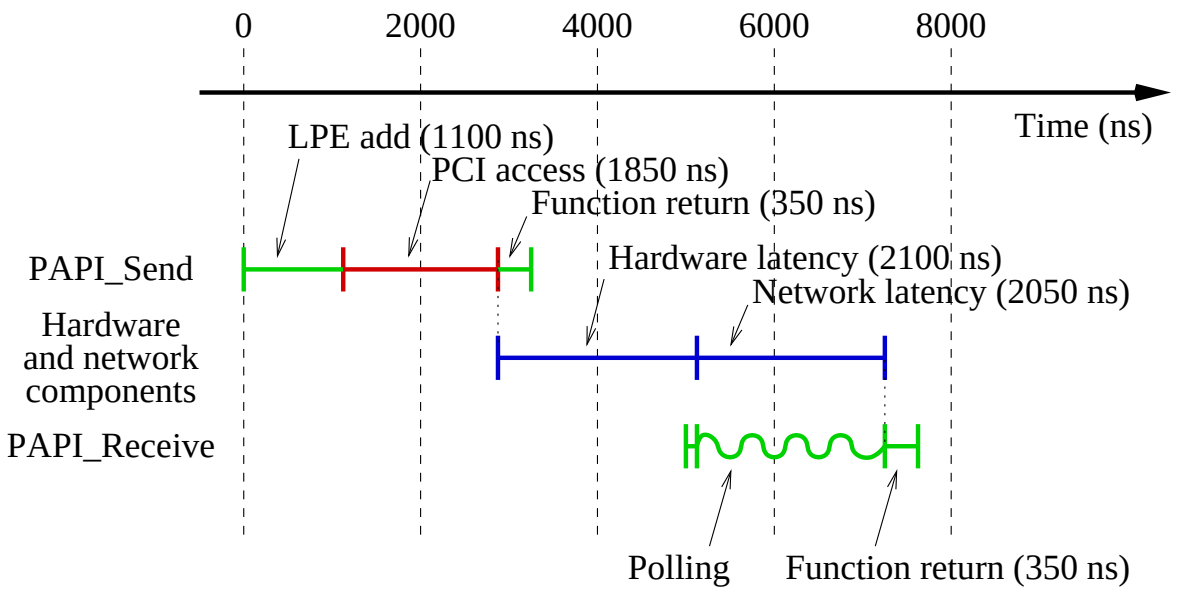
- PAPI : BW limited by the HSL link (64 MB/s)
- BIP and FM : BW limited by the PCI bus (132 MB/s)

Rate of User Bandwidth



- $n_{1/2} = 321$ bytes
- Bandwidth : 90% for 4 kB and 95% for 32 kB

Time Diagram (example with a 128-byte message)



- PAPI / HSL overhead is constant and equal to 1.45 μs
- I.e. 26.9% for 1 byte, 10% for 560 bytes, 1% for 8600 bytes and 0.13% for 64 kB

Conclusion

PAPI:

. provides a lightweight interface for the HSL network

. uses a large part of the bandwidth, even for small messages

. is limited by the HSL-network bandwidth

Two points are highlighted:

. too much time is wasted in PCI bus accesses

. another notification mechanism should be developed

Future Work

- Defining a global architecture for the Remote Write
  - Other networks: Myrinet, SCI, Memory Channel, ...
  - Other operating systems: Linux, Sun OS, ...
  - Other languages: C++, Java, ...

- Developing applications using this programming model
  - DaSSF (a generic simulator developed at Dartmouth College)
  - Any kind of applications