The Lightweight Protocol CLIC: Performance of an MPI implementation on CLIC

Díaz, A. F.; Ortega,J.; Cañas, A.; Fernández, F.J.;Prieto.A

Departamento de Arquitectura y Tecnología de Computadores,
Universidad de Granada

Email: afdiaz@atc.ugr.es, julio@atc.ugr.es, acanas@atc.ugr.es, jfernand@atc.ugr.es,aprieto@ugr.es

Introduction

- Usually Cluster Communications in Linux are based on TCP/IP
- Portability is reduced if we try to use NIC & Architecture optimizations

  (NICs can be obsoleted in a short time)
- OS can’t offer full network performances:
  - Latency / Bandwith.
  - Problems with multiple NICs. (Channel Bonding)
  - Herogenous Networks.
  - Reliable Broadcast/ Multicast.

What is CLIC ?

- CLIC (Communication on LInux Cluster)
- Reliable Transport System optimized for Cluster Computing
- Developed on Linux (kernel module)
- Using OS resources:

  (*scheduler*, NIC drivers, kernel functions)

CLIC Features

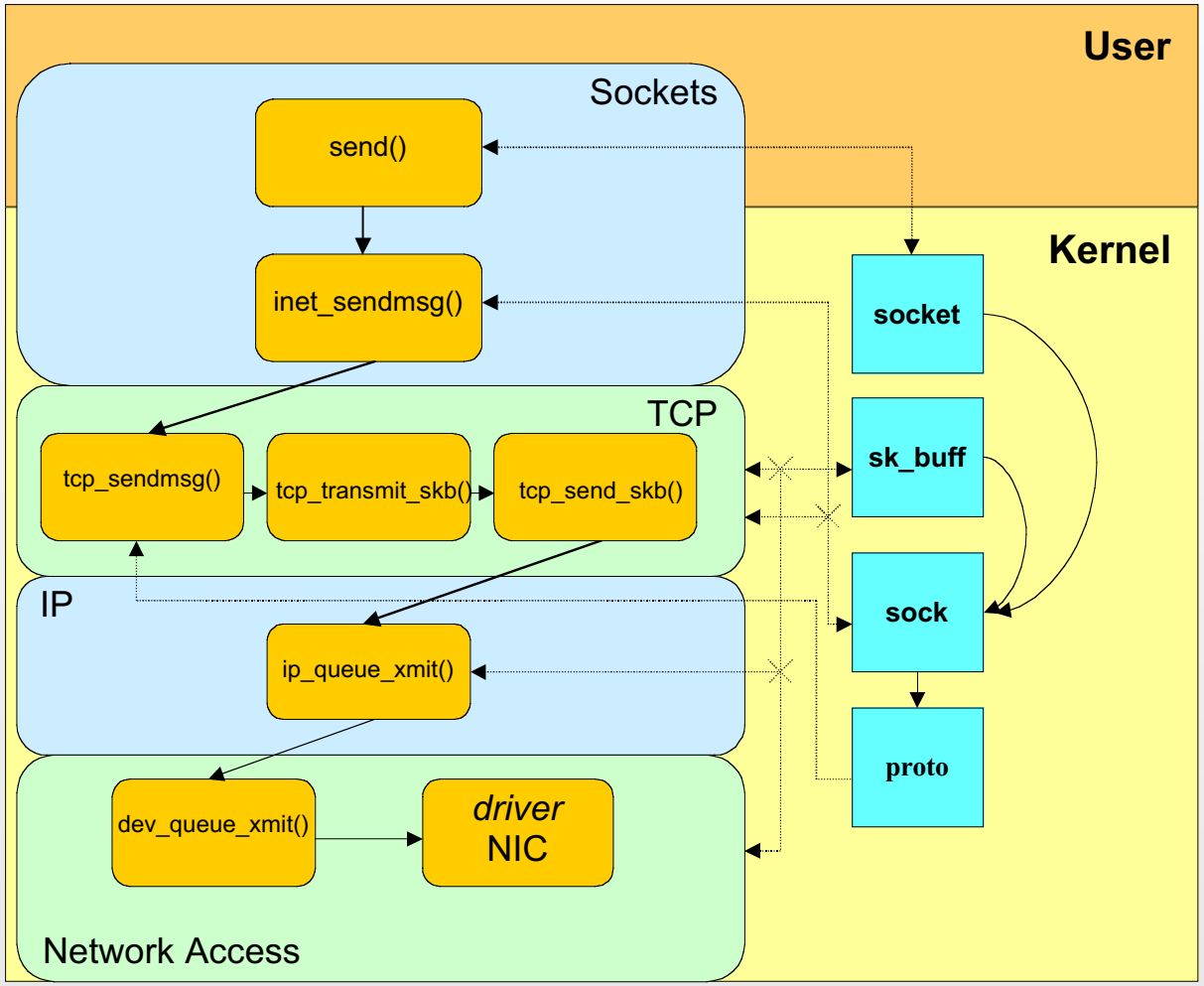
- Low *overhead SW*. Avoid TCP/IP stack.
- Own flow control & errors correcting mechanims: Unicast, Multicast & Broadcast
- More Send types: *Broadcast*, Asyncronous
- Architecture Independent:

  Autoconfiguration, *Channel Bonding*.
- Heterogeneous Topologies. Routing
- Multiple Transport Characteristics.
- Multithread & SMP Support. Spinlocks & mutex

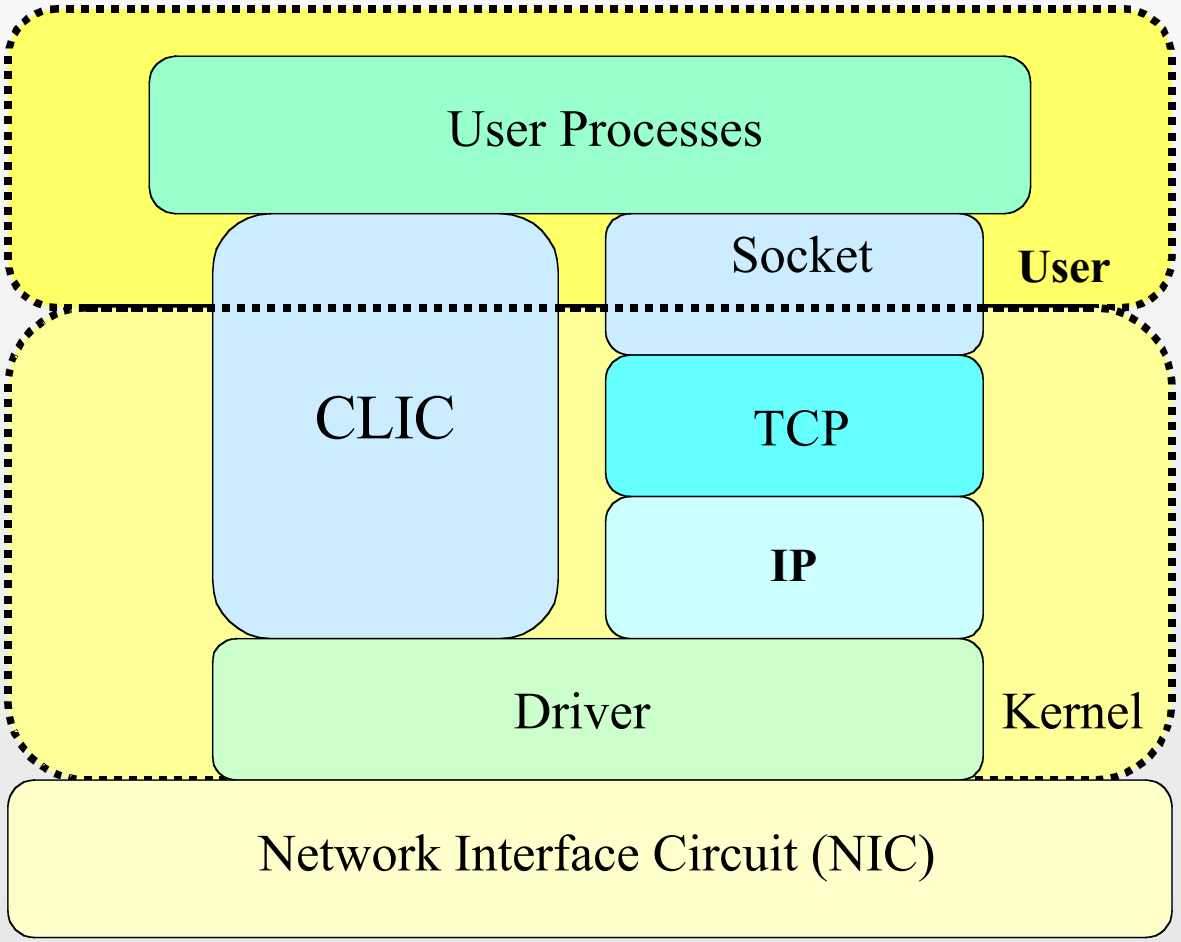
CLIC vs. other Systems

Features	CLIC	GAMMA	VIA	Beowulf	Mad. II	U-NET
NIC & Arch. Indep.	YES	NO	YES	YES	YES	No
Threads, SMP	YES	NO	YES	YES	YES	YES
Multiprotocol Supp.	YES	NO	YES	Op.Syst.	YES	NO
Secure System	YES	NO	NO	NO	NO	NO
Uni. Flow-Control	YES	YES	NO	Op.Syst.	YES	NO
Broadcast Flow-Ctl.	YES	NO	NO	NO	NO	NO
Packet Recovery	YES	NO	NO	Op.Syst.	YES	NO
Channel Bonding	YES	NO	NO	YES	NO	NO
Sincr. & Asyncr.	YES	NO	NO	NO	YES	NO
Autoconfiguration	YES	NO	NO	NO	?	NO
Routing	YES	NO	NO	Op.Syst.	Het. No enr	NO
Send. same Host	YES	NO	YES	Op.Syst.	YES	NO

IP Stack


User
Kernel
Sockets
send()
inet_sendmsg()
TCP
tcp_sendmsg()
tcp_transmit_skb()
tcp_send_skb()
IP
ip_queue_xmit()
dev_queue_xmit()
driver
NIC
Network Access
socket
sk_buff
sock
proto

Avoiding IP Stack ...

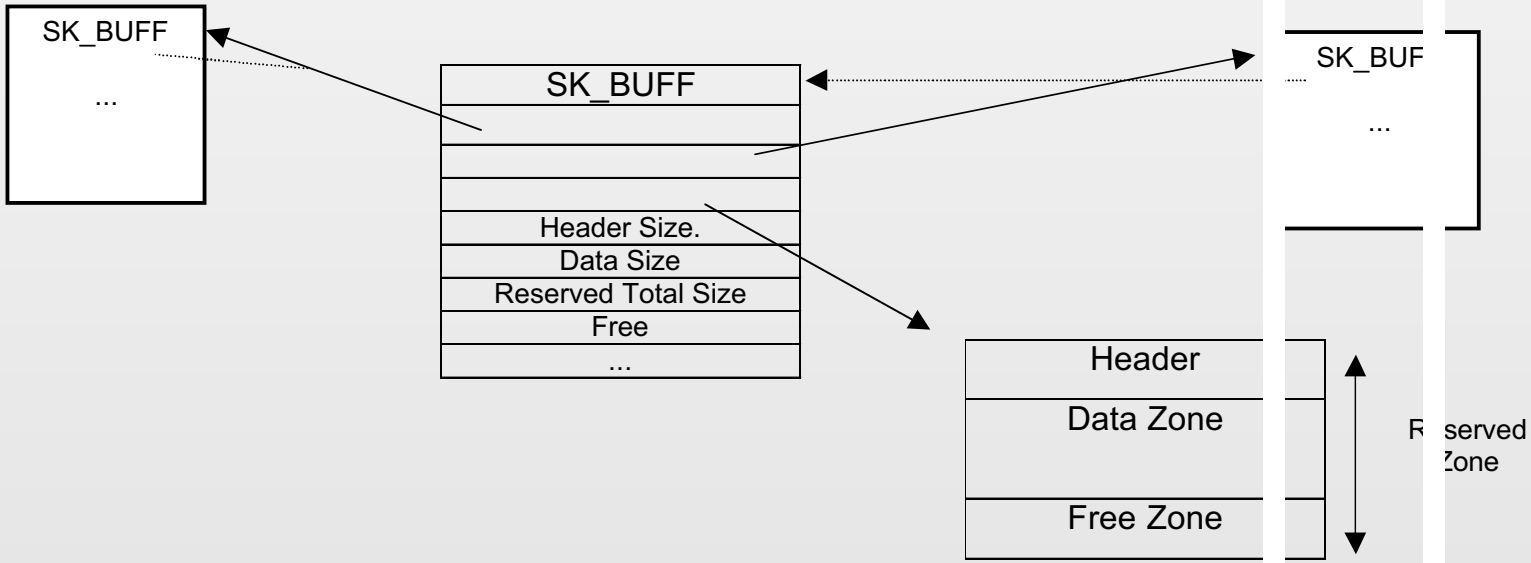


Sk_buff ←→ NIC

-Information exchange with NIC.

-*sk_buff* needs memory copy sk_buff but:

architecture independence, pipeline in send process

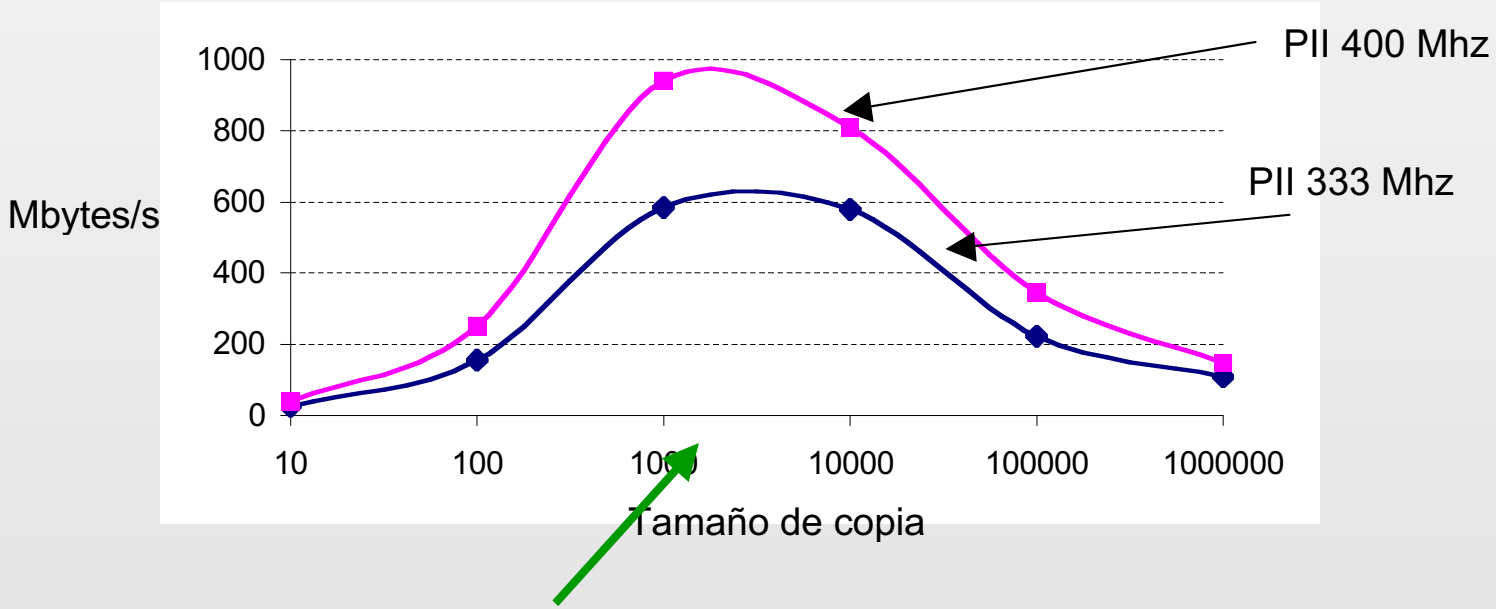


Bandwidth & Memory Cost

-Minimize number of memory copys, better.

-Using 0-copy is not allways better performances.

-Bottleneck is in the network, not in the memory.



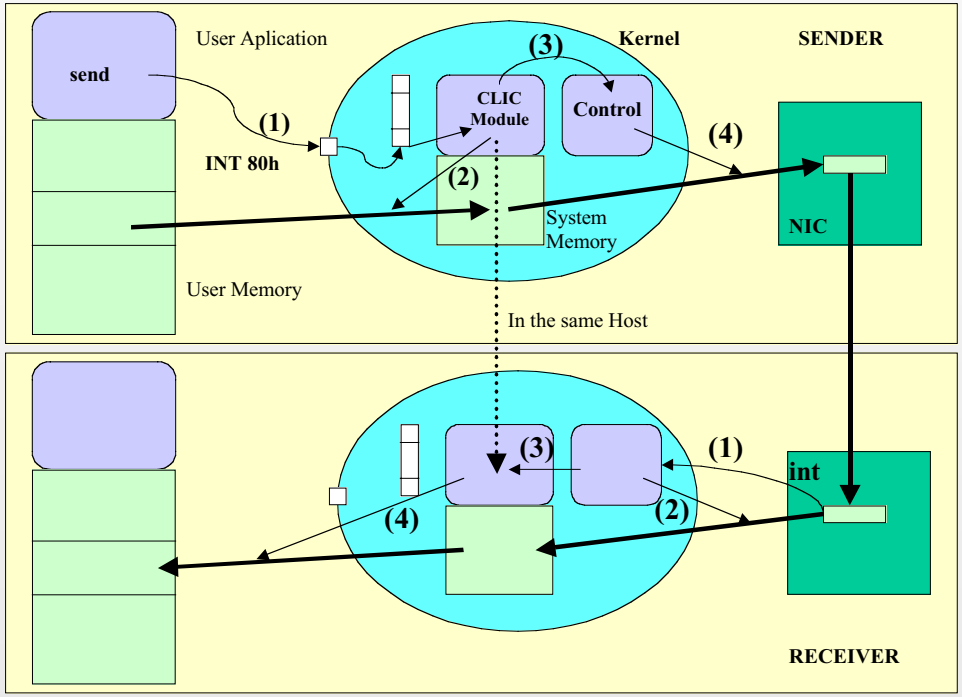
How CLIC works ...

SENDER

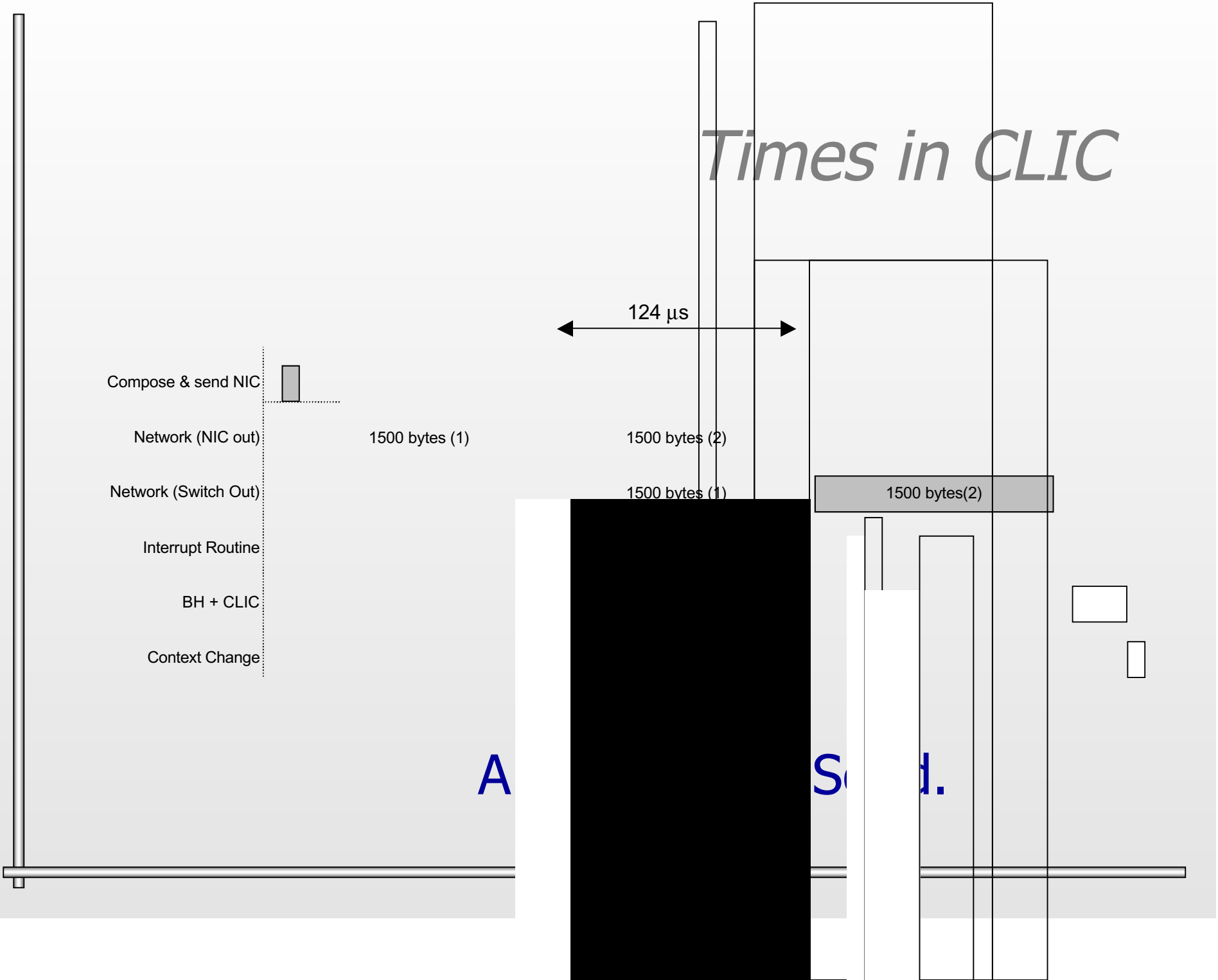
(1) System Call
(2) Copying to *sk_buf*
(3) Flow Control
(4) Sending to NIC (*driver*)

RECEIVER

(1) Interrupt Routine
(2) *Botton Halves*
(3) CLIC control & proc.
(4) Send user's memory

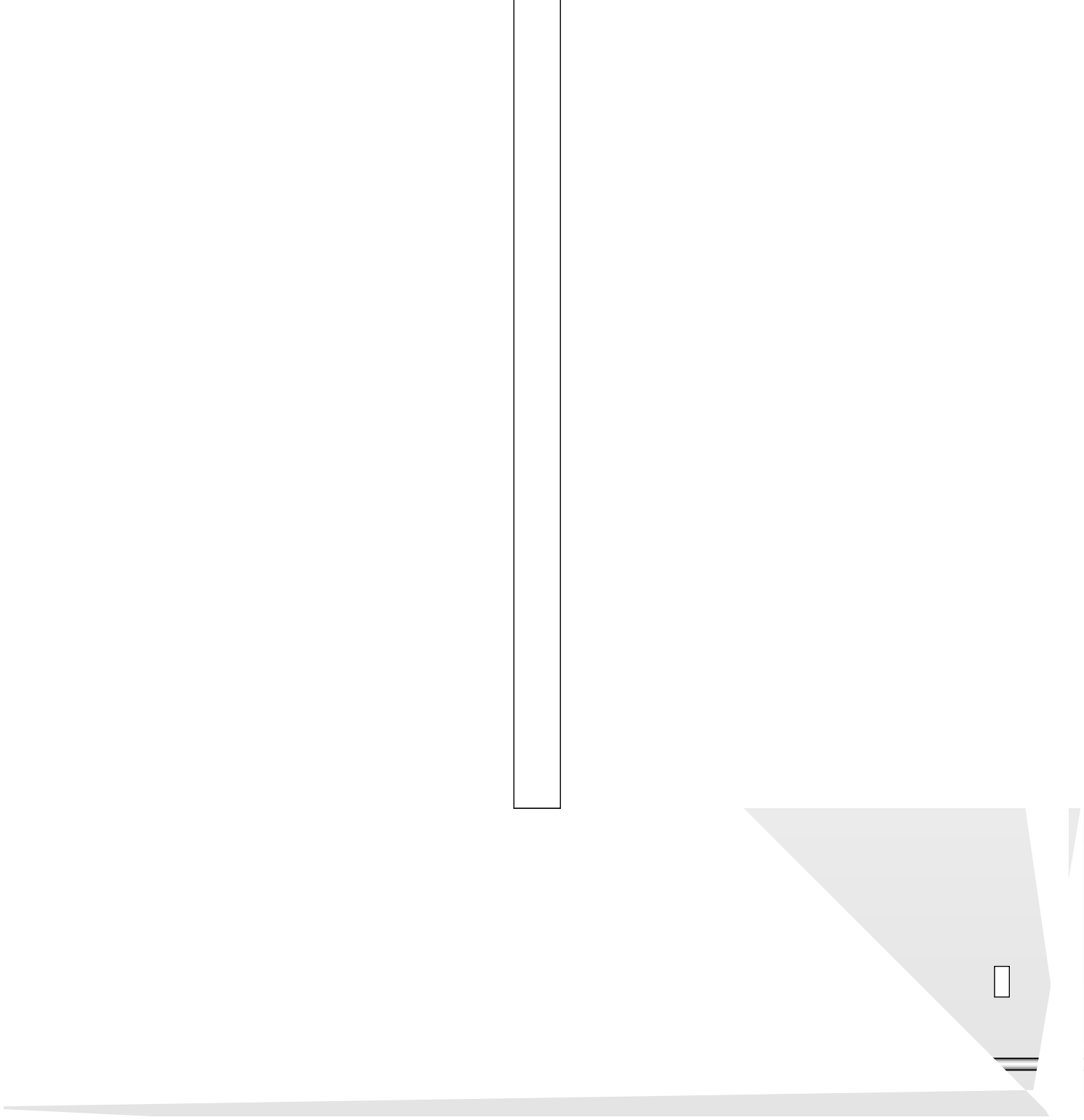


Times in CLIC



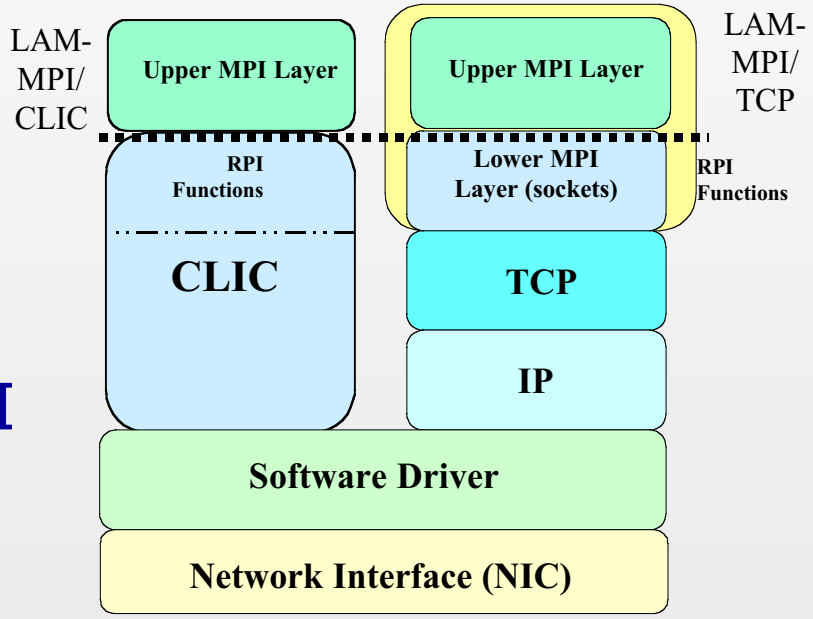
Other Optimizing OS CLIC…

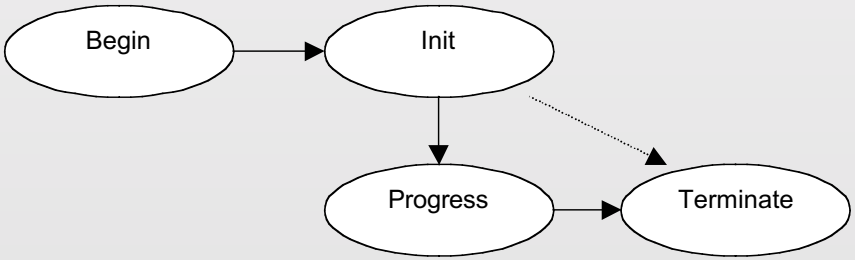
- SEND:
  - Memory managent (*sk_buffer* reuse).
  - Open communication ASAP.
  - Direct NIC Send.
- Receive:
  - Processing packets
- Both:
  - Flow Control optimized for full-speed.
  - Selective Lost packet re-transmitions.
  - Low overhead.



LAM-MPI over CLIC

- RPI Functions. C2C Model.
- Multiple process, nodes and users of MPI >> CLIC_module
- MPI Requests are converted into CLIC messages with MPI characteristics (Buffered, Sincronous, Ready) & (comm,tid, source, dest)
- Request can progres automaticaly informing Upper MPI Layer.





CLIC Performances

Message Size	PVM	MPI-TCP	MPI-CLIC	CLIC
10	224	140	62	59
100	248	153	102	94
1000	454	379	204	196
10000	1285	1279	943	932
100000	9018	8990	8690	8615

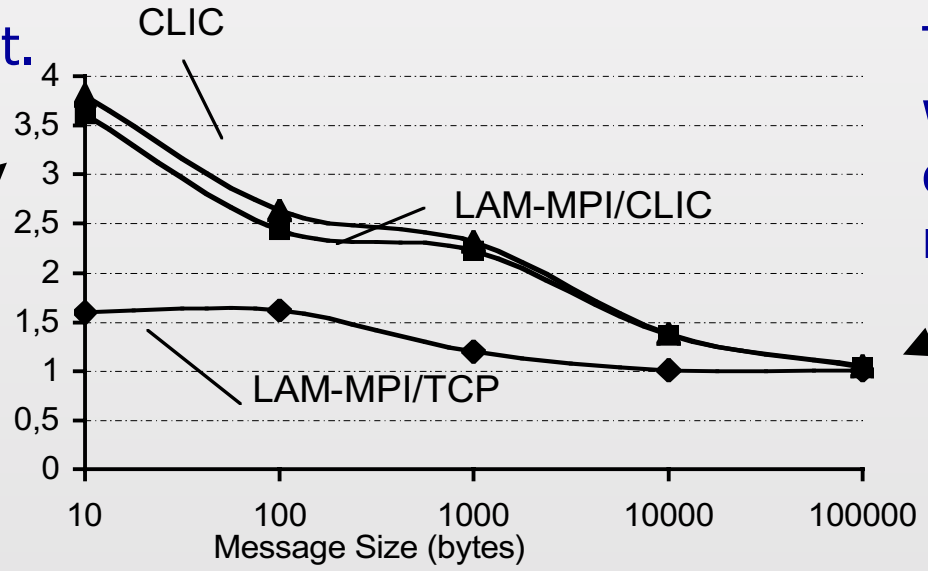
Data transfers

(time in μs)

Network: Fast Ethernet & Switch

Speedup w.r.t. PVM-TCP

CLIC has lower Software overhead

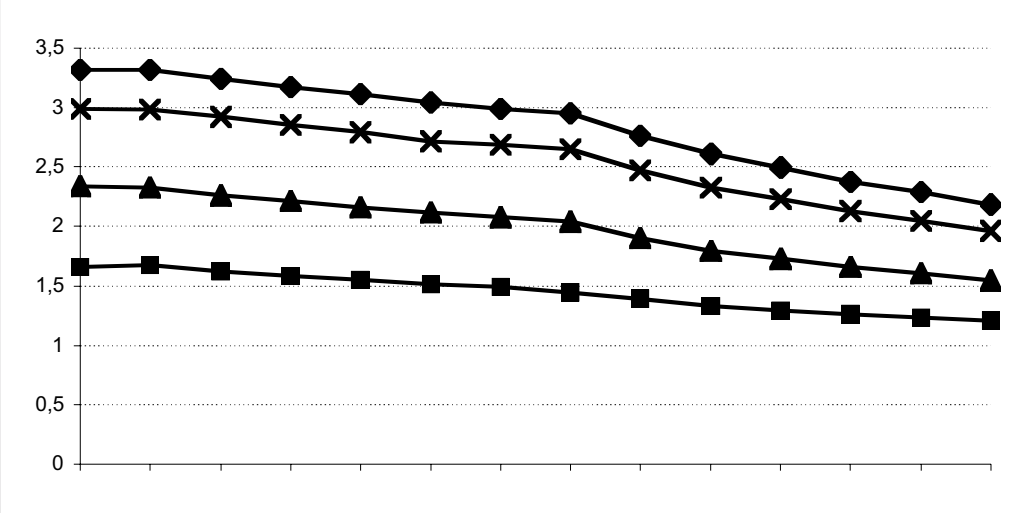


TCP/IP works while packets crossing the network

CLIC Performances (II)

Matrix Product:

Latency is also important...



64 Matriz Size 2048

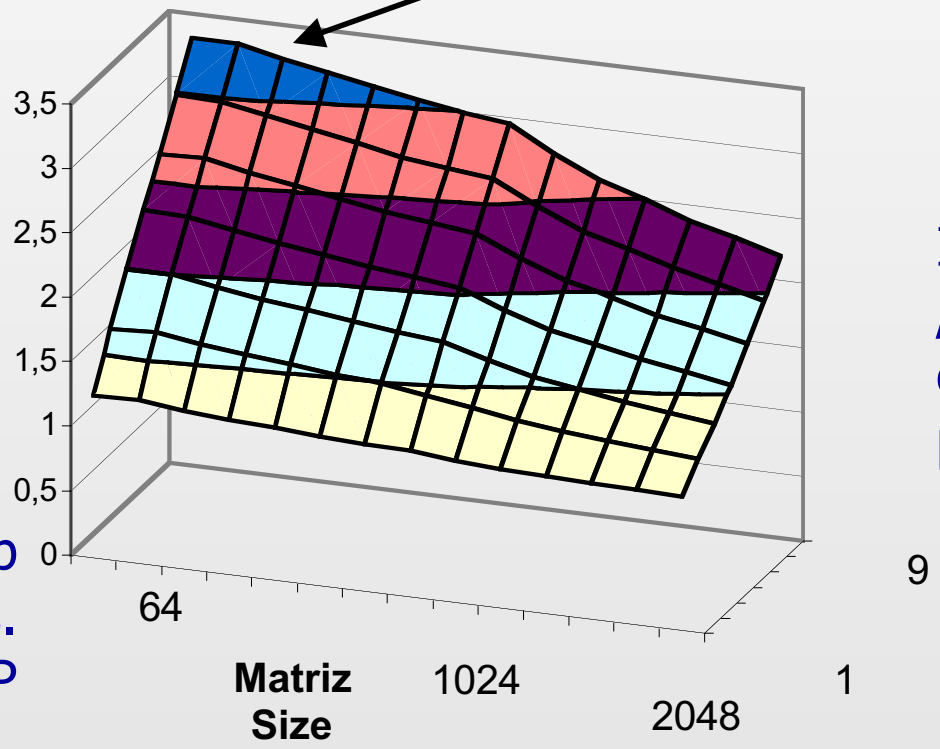
Speedup MPI-CLIC vs. MPI-TCP

More nodes, better performances if you use *broadcast send*

CLIC Performances (III)

Matrix Product:

Latency is also important...



Speedup MPI-CLIC vs. MPI-TCP

If you use Broadcast /Multicast CLIC can get even better performances

Conclusions

- A communication layer (CLIC) is proposed to use OS efficiently.
- Upper layer systems (PVM, MPI,…) can be efficiently used on top of CLIC.
- CLIC improves the performance of the comunications so user-level aplications can take advantage of network features. (Better latency & Bandwith, *Broadcast,Channel Bonding*).
- This System can be upgraded to new networking techonogies. (Future study)