Tuning Lustre in a LNet routed environment

Or putting those net params in sync

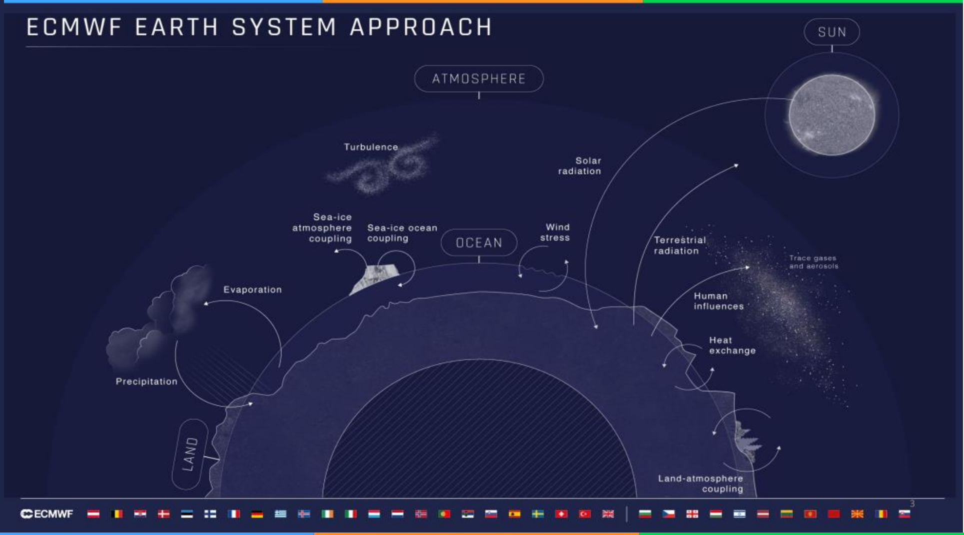
Sebastien Piechurski
Storage & I/O performance expert
27/09/2022

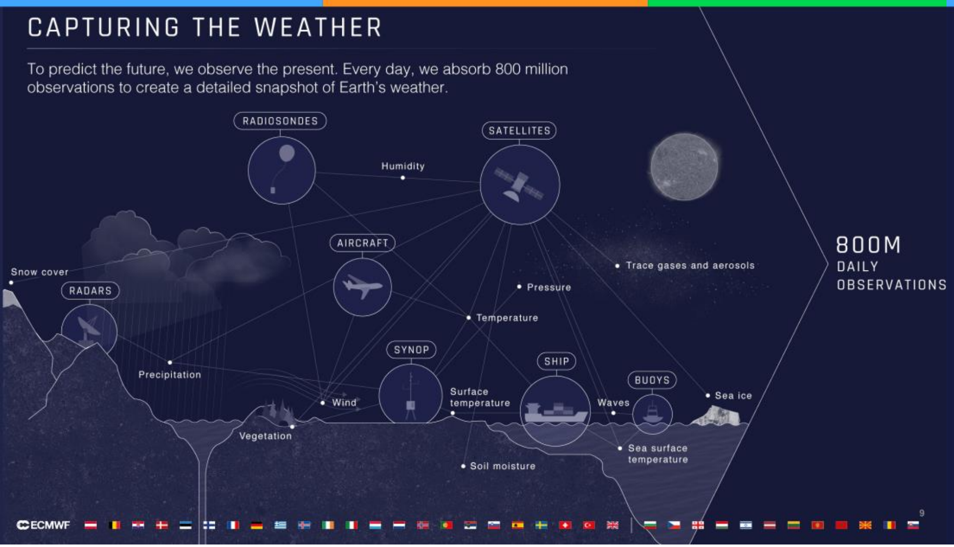
Atos

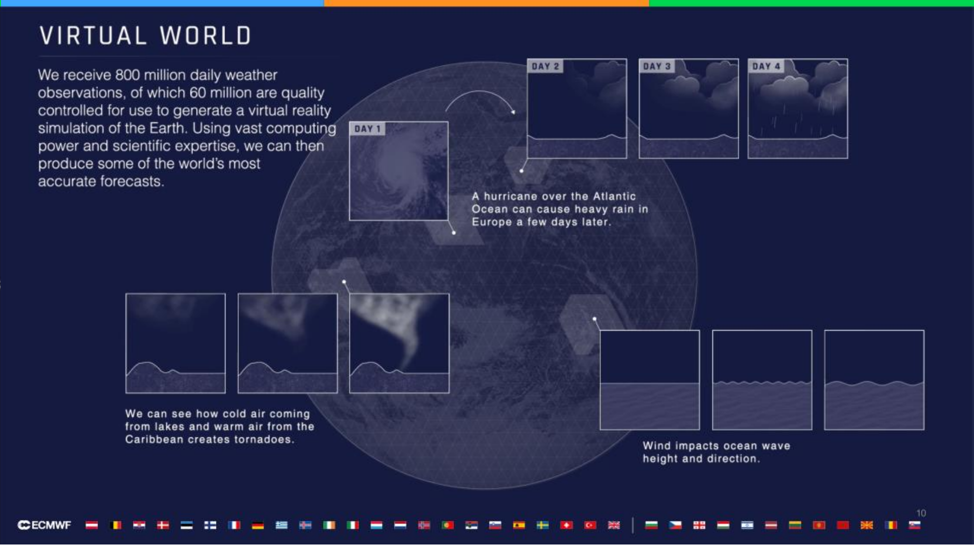
01. ECMWF Presentation

Atos


ECMWF
The strength
of a common goal
ECMWF | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

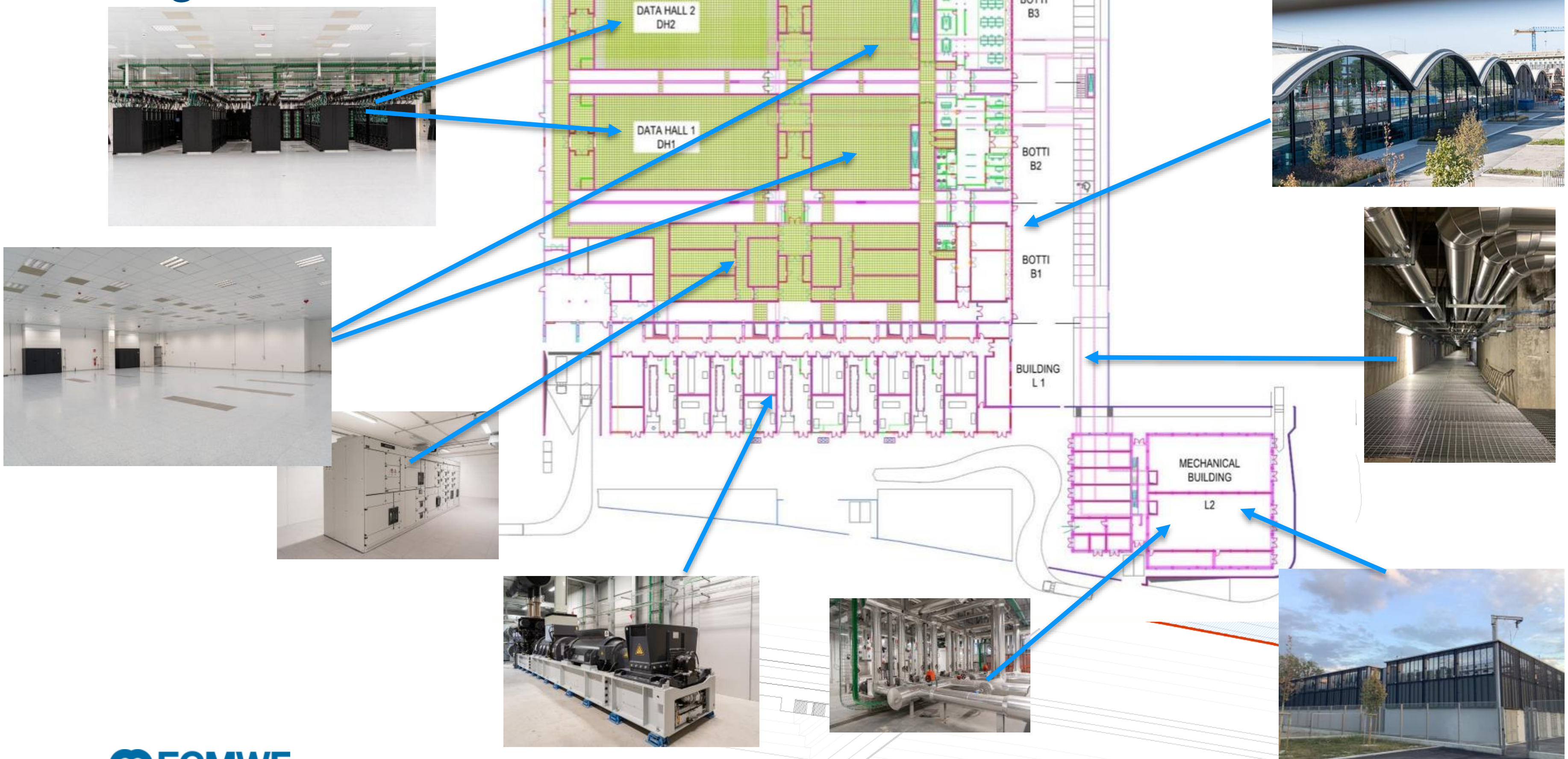

ECMWF EARTH SYSTEM APPROACH
SUN
ATMOSPHERE
Turbulence
Solar radiation
Sea-ice atmosphere coupling
Sea-ice ocean coupling
OCEAN
Wind stress
Terrestrial radiation
Trace gases and aerosols
Evaporation
Human influences
Heat exchange
Precipitation
LAND
Land-atmosphere coupling
ECMWF
3


CAPTURING THE WEATHER
To predict the future, we observe the present. Every day, we absorb 800 million observations to create a detailed snapshot of Earth's weather.
RADIOSONDES
SATELLITES
Humidity
AIRCRAFT
Snow cover
RADARS
Trace gases and aerosols
Pressure
Temperature
800M
DAILY
OBSERVATIONS
Precipitation
SYNOP
SHIP
BUOYS
Surface temperature
Wind
Waves
Sea ice
Vegetation
Sea surface temperature
Soil moisture
ECMWF
9


VIRTUAL WORLD
We receive 800 million daily weather observations, of which 60 million are quality controlled for use to generate a virtual reality simulation of the Earth. Using vast computing power and scientific expertise, we can then produce some of the world's most accurate forecasts.
DAY 1
DAY 2
DAY 3
DAY 4
A hurricane over the Atlantic Ocean can cause heavy rain in Europe a few days later.
We can see how cold air coming from lakes and warm air from the Caribbean creates tornadoes.
Wind impacts ocean wave height and direction.
ECMWF
10

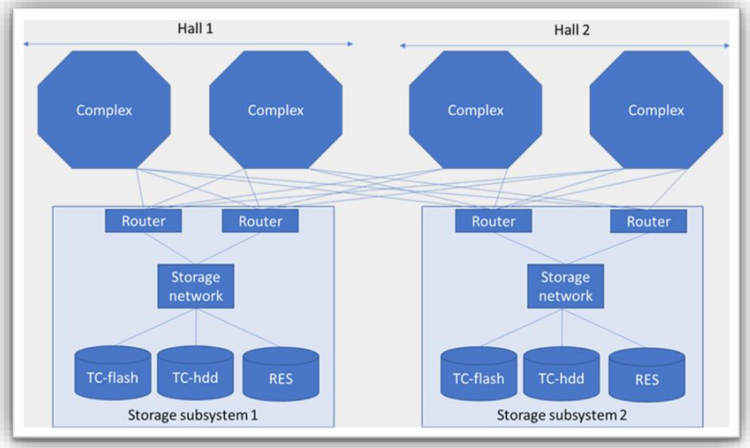
Bologna Data Centre's High-Performance-Computing Facility

Bologna Data Centre



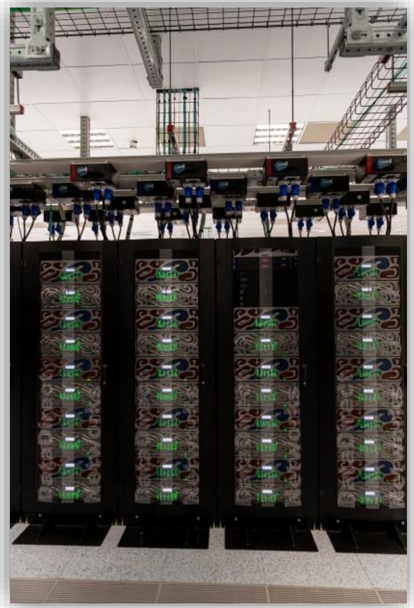
Atos BullSequana XH2000

- 4 Complexes
    - Two in each hall
    - Each Complex consists of two clusters:
    - Parallel:
        - ATOS XH2000 Water cooled racks
        - Arranged in 5 “cells”, 4 racks per cell
        - IB HDR Fat Tree in each cell. Each cell connected to every other cell
        - 1920 nodes for parallel compute
        - AMD Rome 64 core processors
    - General Purpose
        - 112 nodes for general purpose use
            - More memory per node, local SSD
- One Slurm scheduler in each complex



Storage Subsystem

- Global Lustre parallel filesystems
  - Magnetic disk and Solid-State storage
  - in total, 10 independent DDN Exascaler filesystems
    - ES7990 & ES200NV appliances
  - Separate file systems for time critical operations and research
    - Time Critical Storage
      - 2 Lustre SSD 700TB file systems for production
      - 2 Lustre HDD 5PB file systems for short term storage
    - Research
      - 6 Lustre HDD 13PB file systems
  - Filesystems available to all clusters
- per-complex NFS storage for /usr/local
- Home and project from external NetApp and TrueNAS NFS filers
- Long term storage in the MARS and ECFS archives

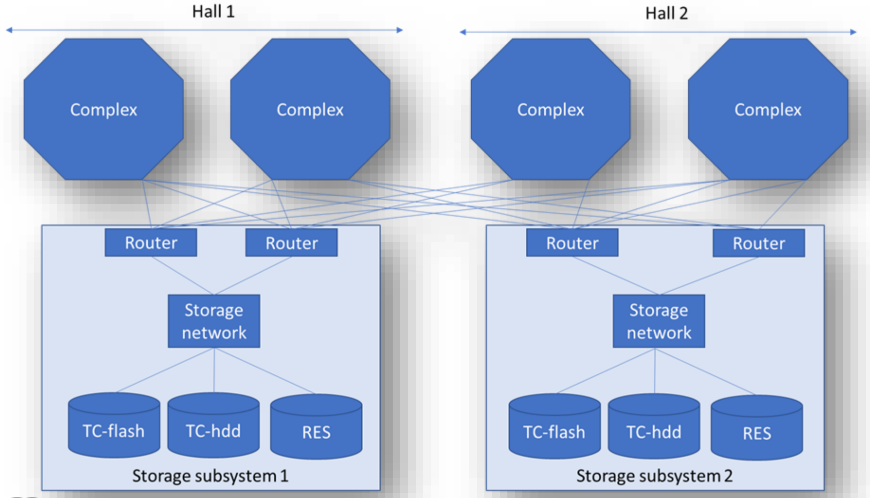


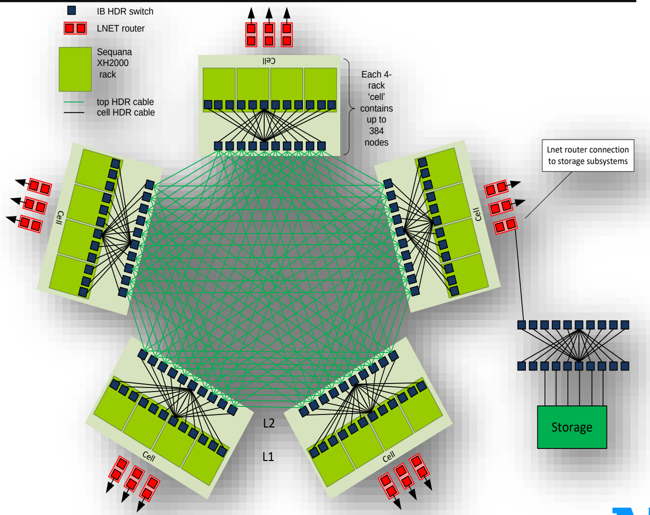
Atos HPC - Compute

Atos BullSequana XH2000 System	
Complexes	4
Each complex has	
Compute nodes	1,920
General purpose nodes	112
Racks	20 water-cooled, 2 air-cooled
Weight (kg)	42,000
Each node has	
Processor type	AMD Epyc Rome 7742 (7H12 in general purpose nodes)
Cores	64 cores/socket, 128 cores/node
Memory/node (GiB)	256 (compute nodes) / 512 (general purpose)
Total	
Memory	~2 PiB
Nodes	7,680 compute, 488 general purpose
Cores	~1 million

Atos HPC - Storage

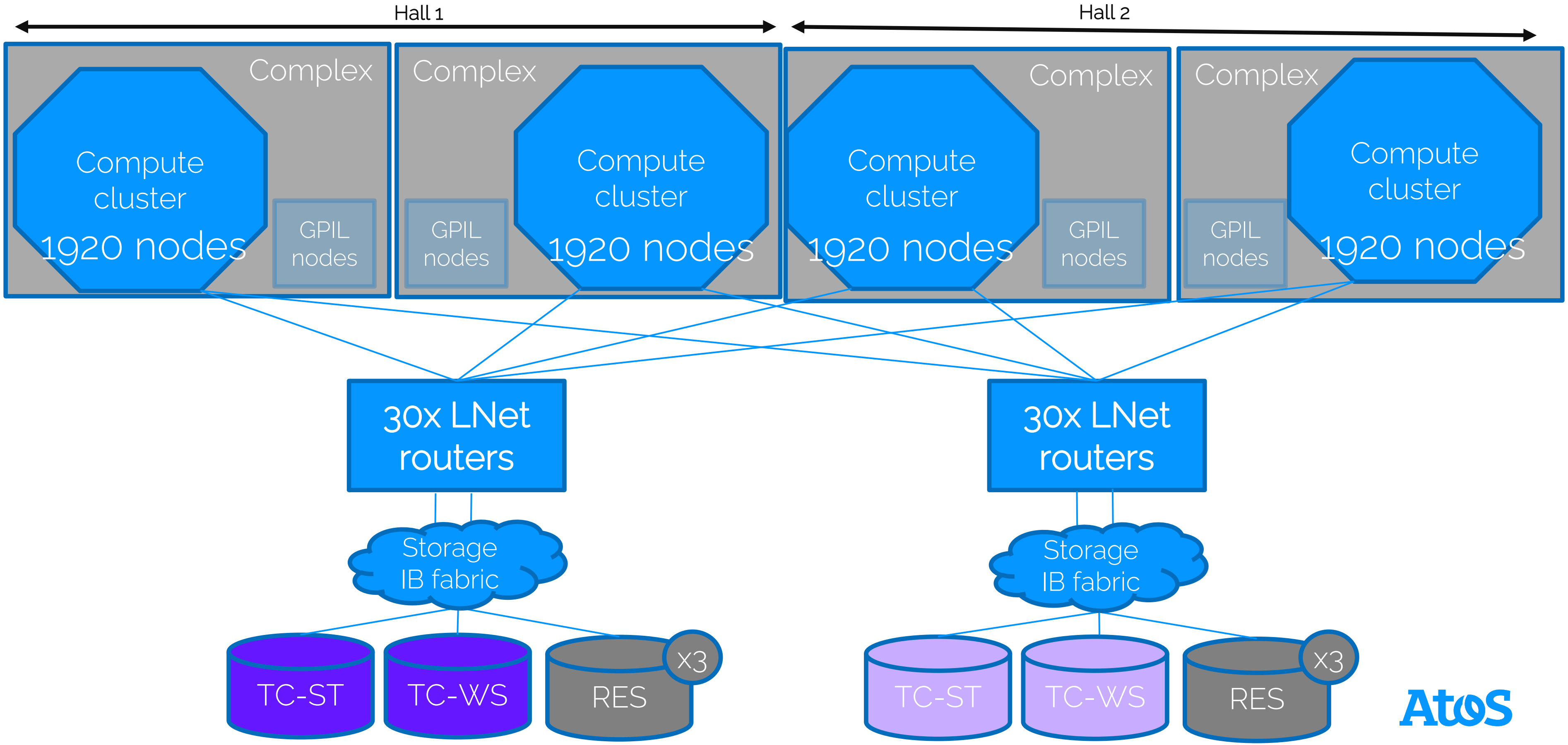
Type	Filesystems	Usable Capacity (PB)	IOR-Bandwidth (GB/s)
Storage for time critical operations			
Flash	2	0.7	307
Hard Disk	2	5.4	112
Storage for research			
Hard Disk	6	13	260





ECMWF Configuration

Global view



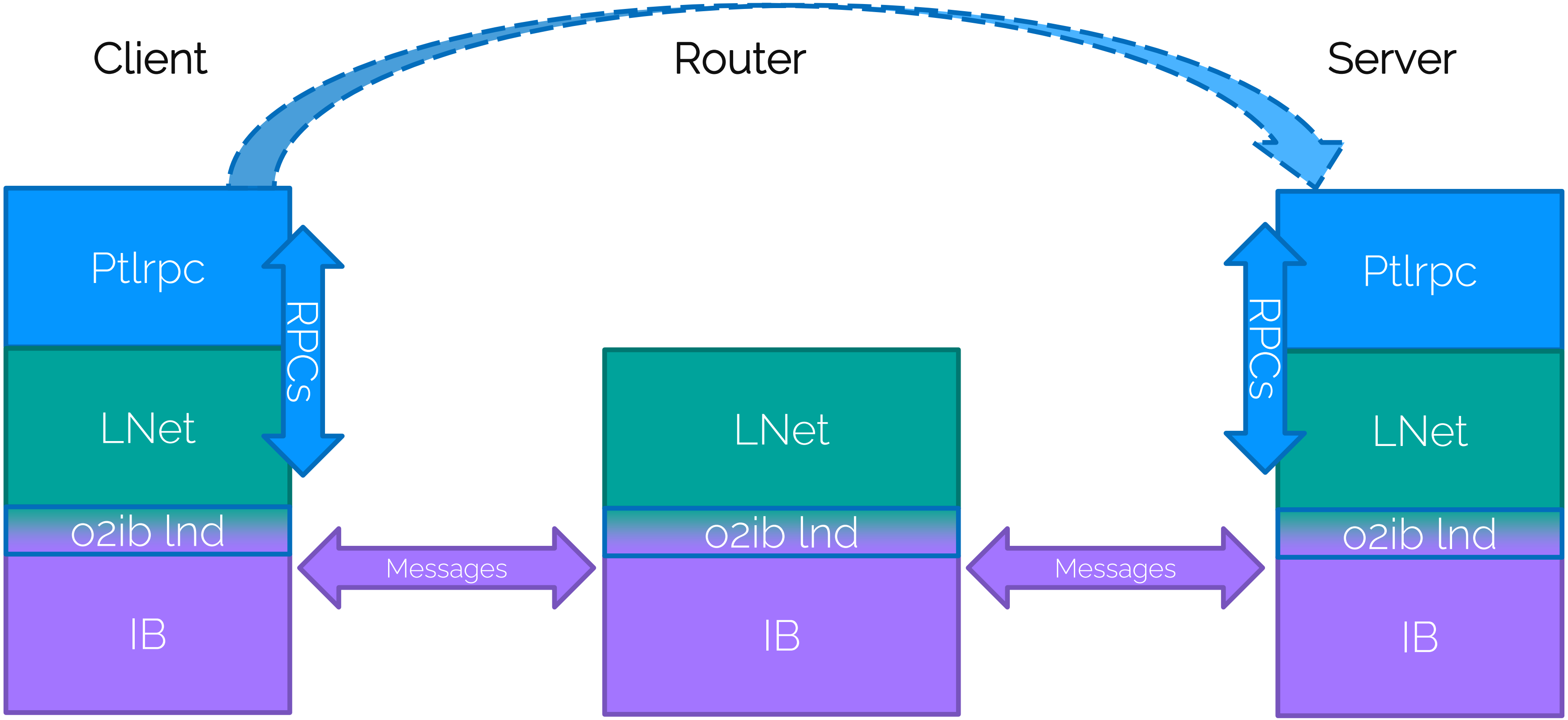
Atos

Lnet routing Theory

Disclaimer:
The following information is taken from our understanding while browsing through lustre 2.12 source and might be either outdated, incomplete or misinterpreted.

Lustre/LNet

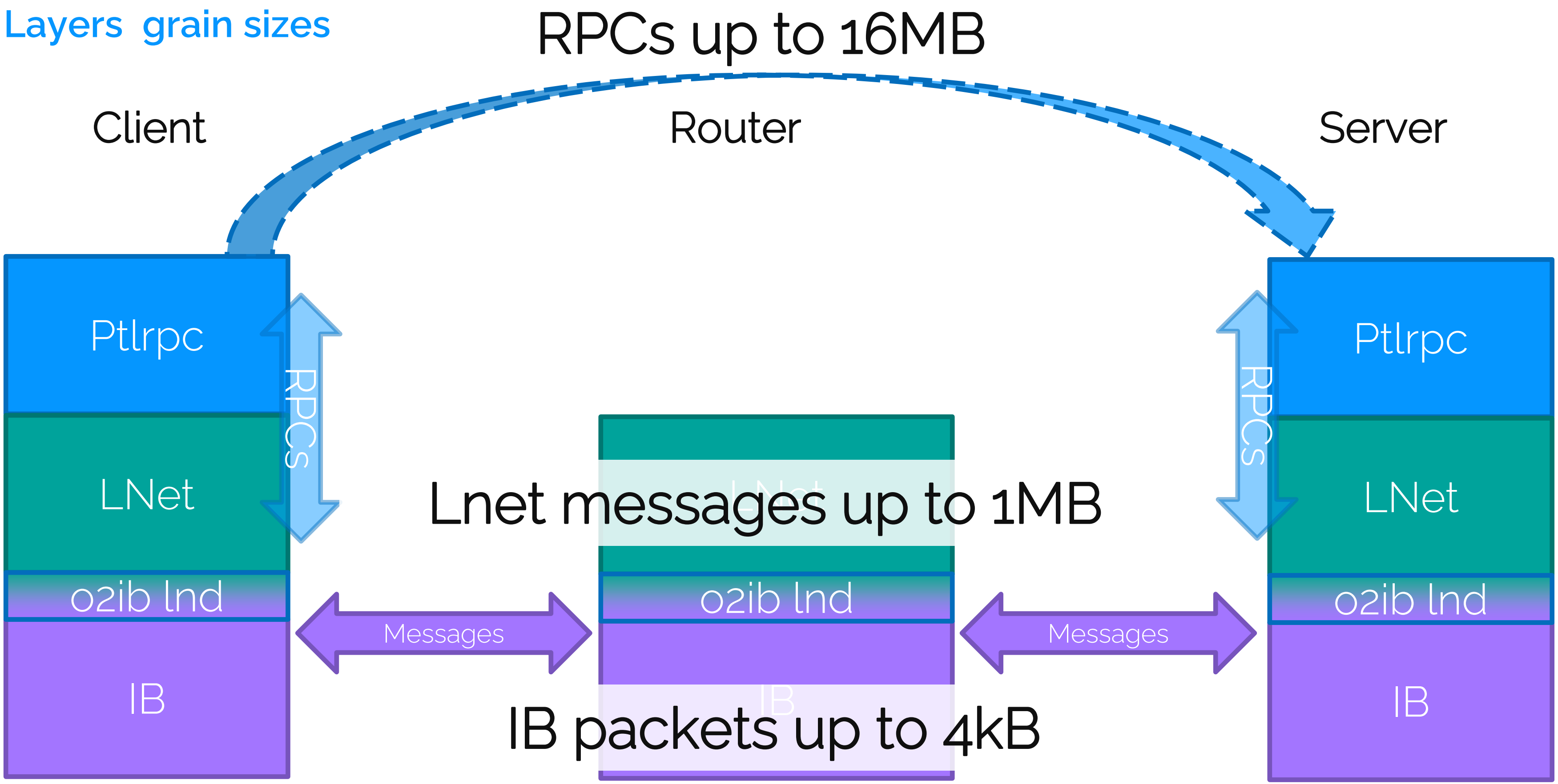
Layers



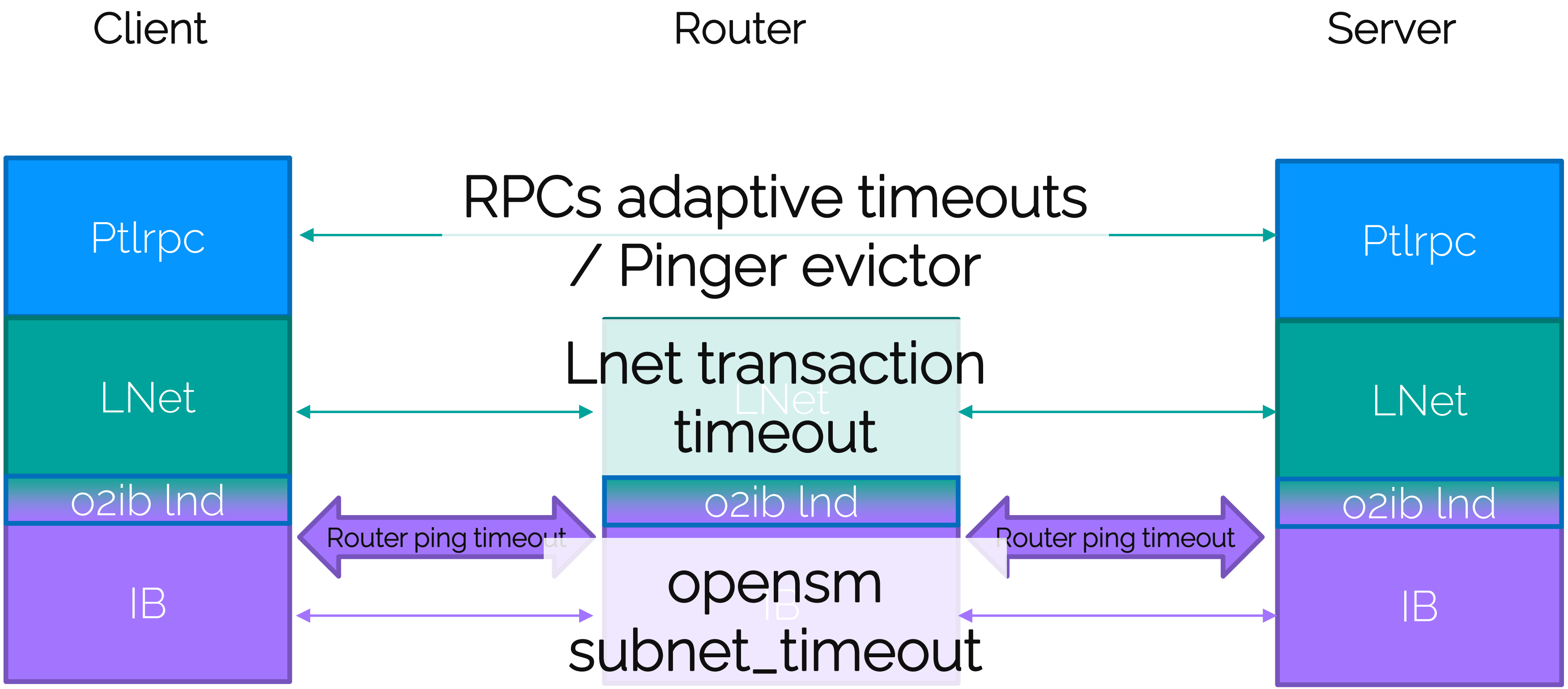
Atos

Lustre/LNet

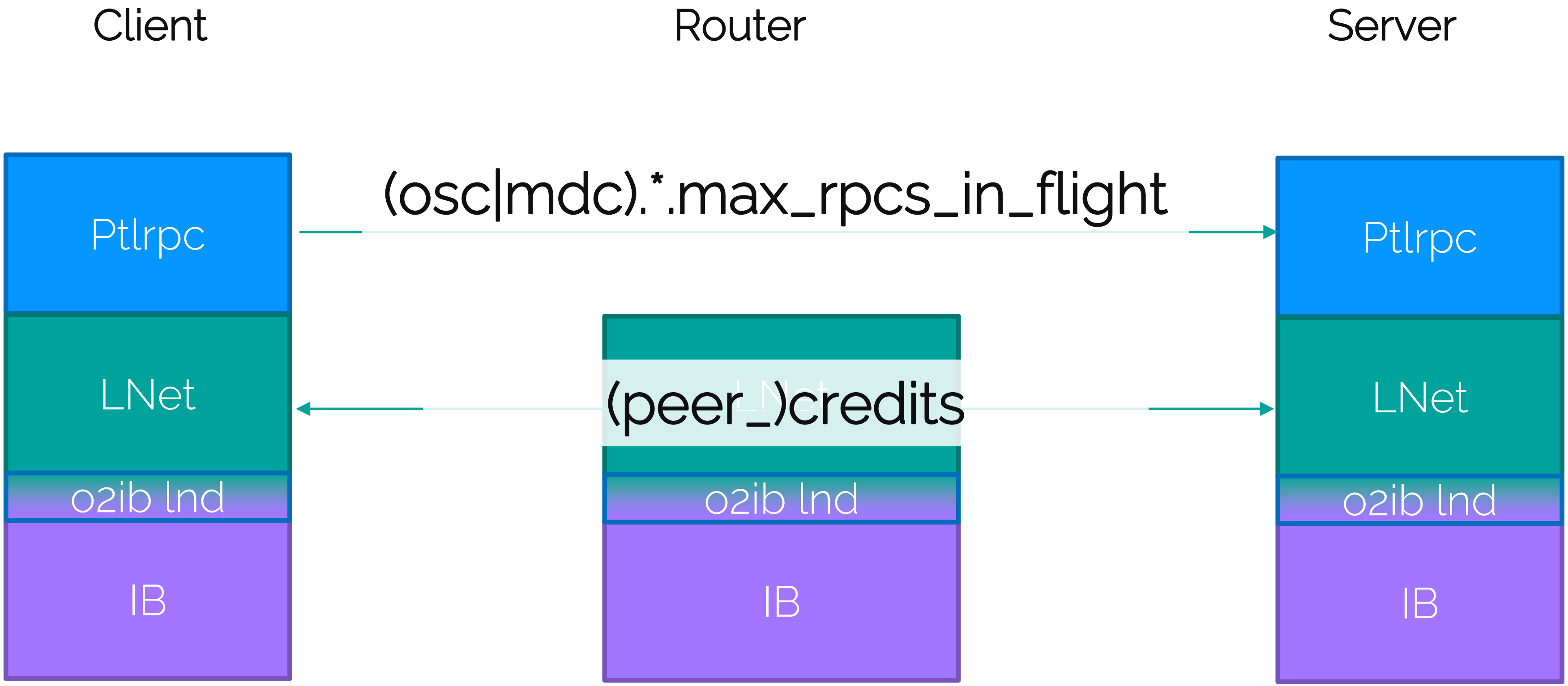
Layers grain sizes



Lustre/LNet
Layers timeouts

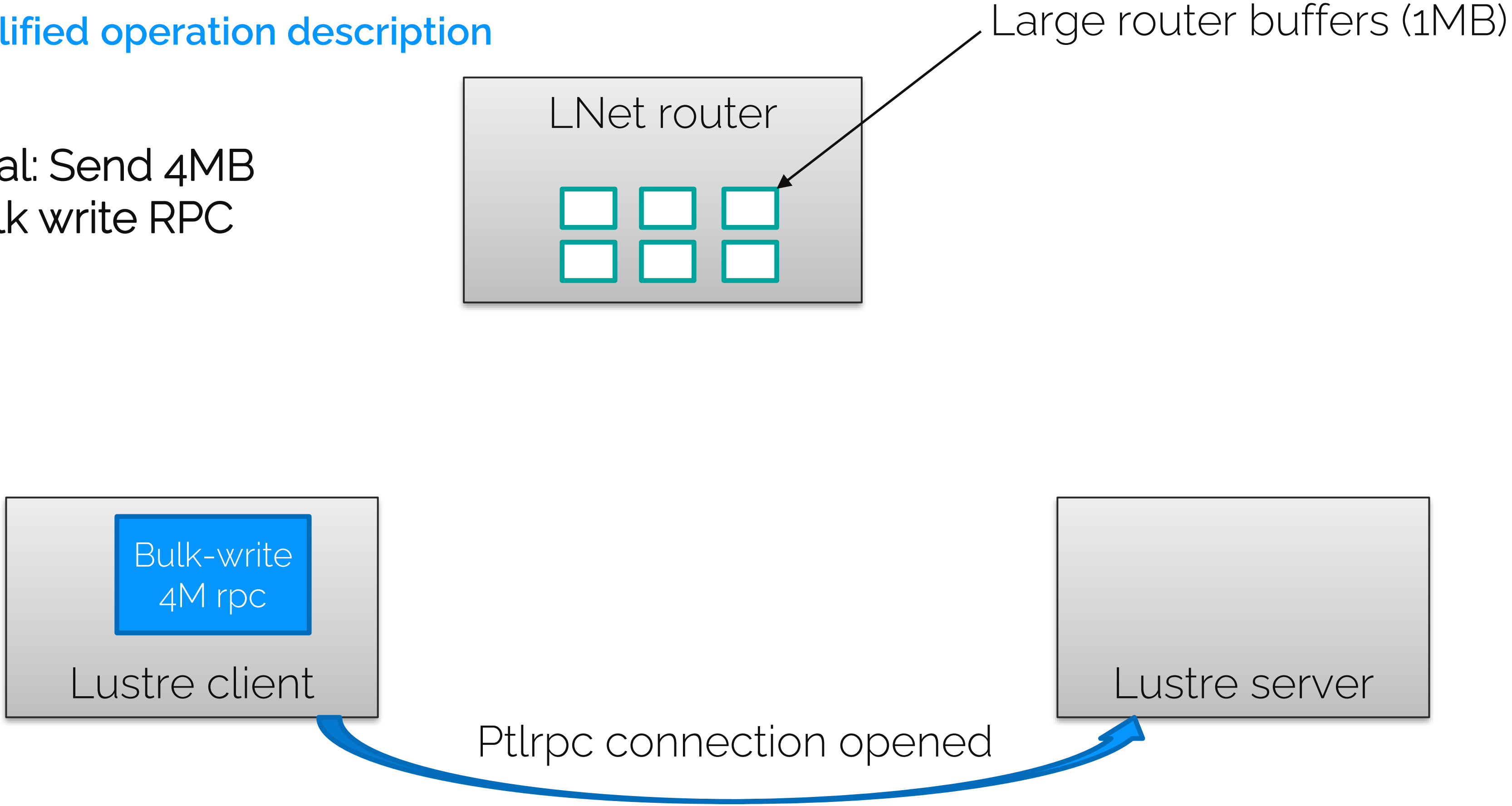


Lustre/LNet
Layers parallelism



LNet router
Simplified operation description

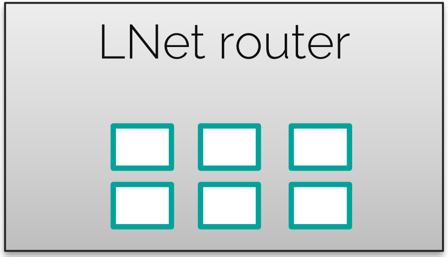
Goal: Send 4MB
Bulk write RPC



LNet router

Simplified operation description

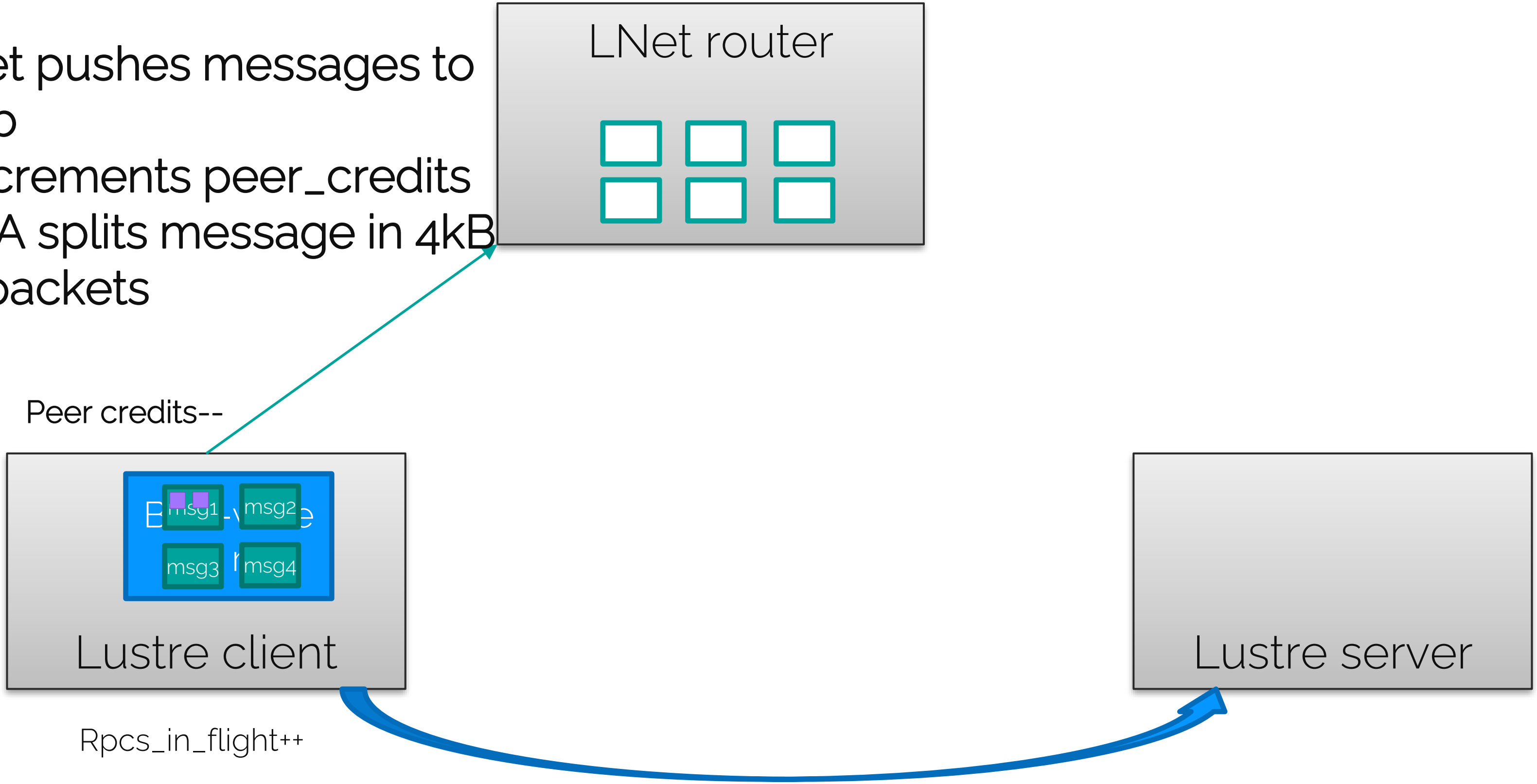
- Ptlrpc layer pushes rpc to Lnet layer.
- Splits into 1MB messages
- Increments rpcs_in_flight



LNet router

Simplified operation description

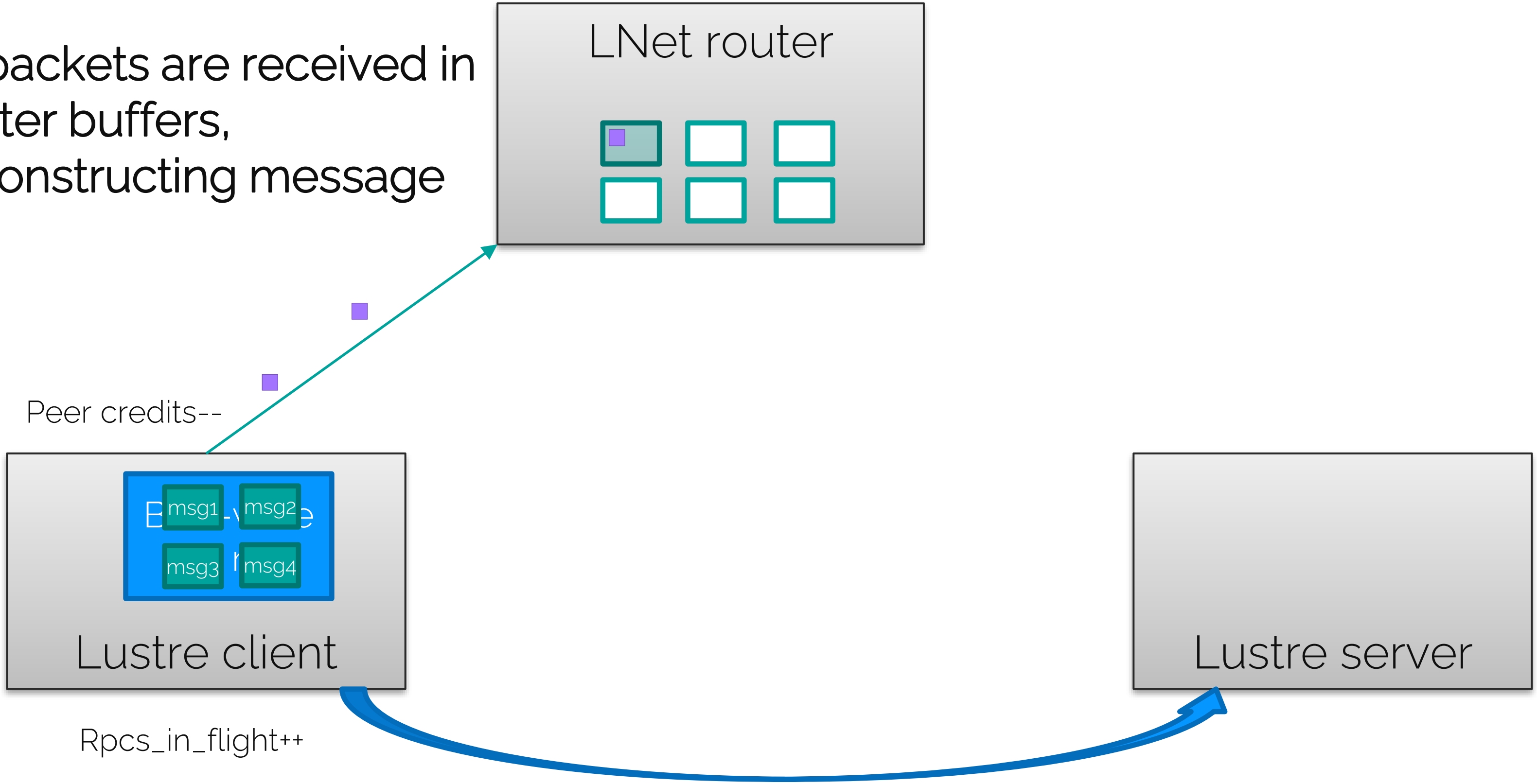
- Lnet pushes messages to o2ib
- Decrements peer_credits
- HCA splits message in 4kB IB packets



LNet router

Simplified operation description

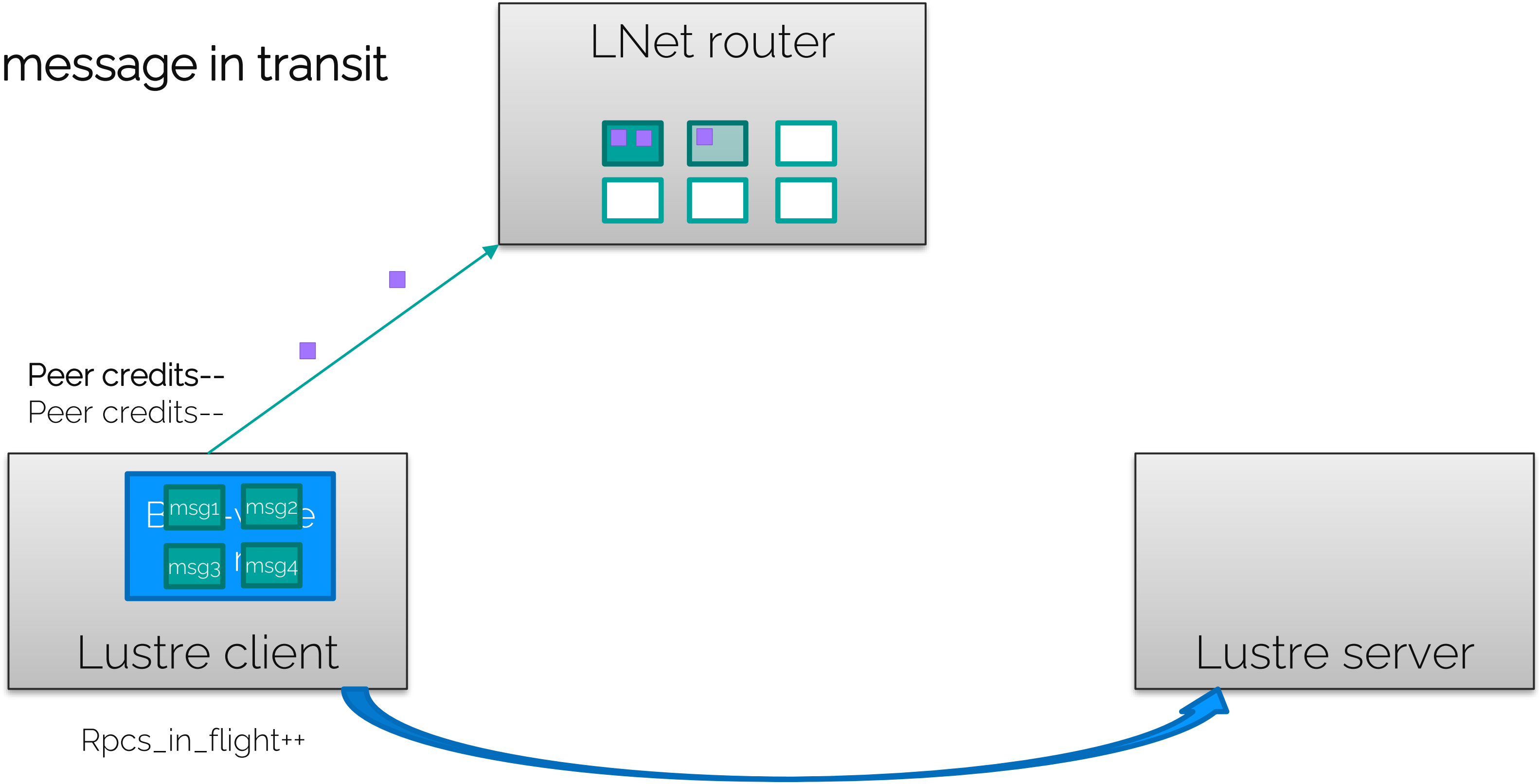
- IB packets are received in router buffers, reconstructing message



Atos

LNet router
Simplified operation description

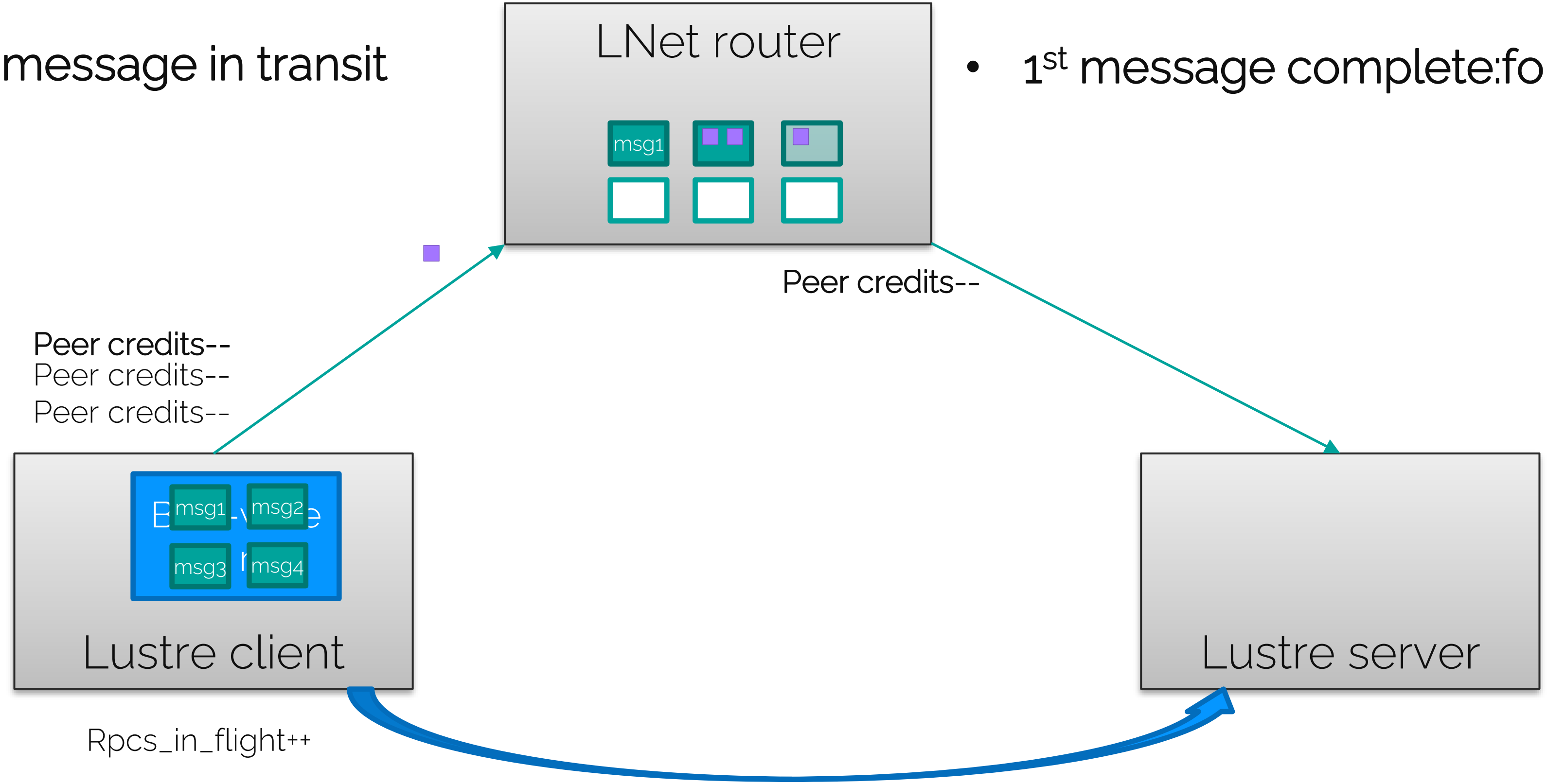
- 2nd message in transit



LNet router

Simplified operation description

- 3rd message in transit
- 1st message complete:forward !

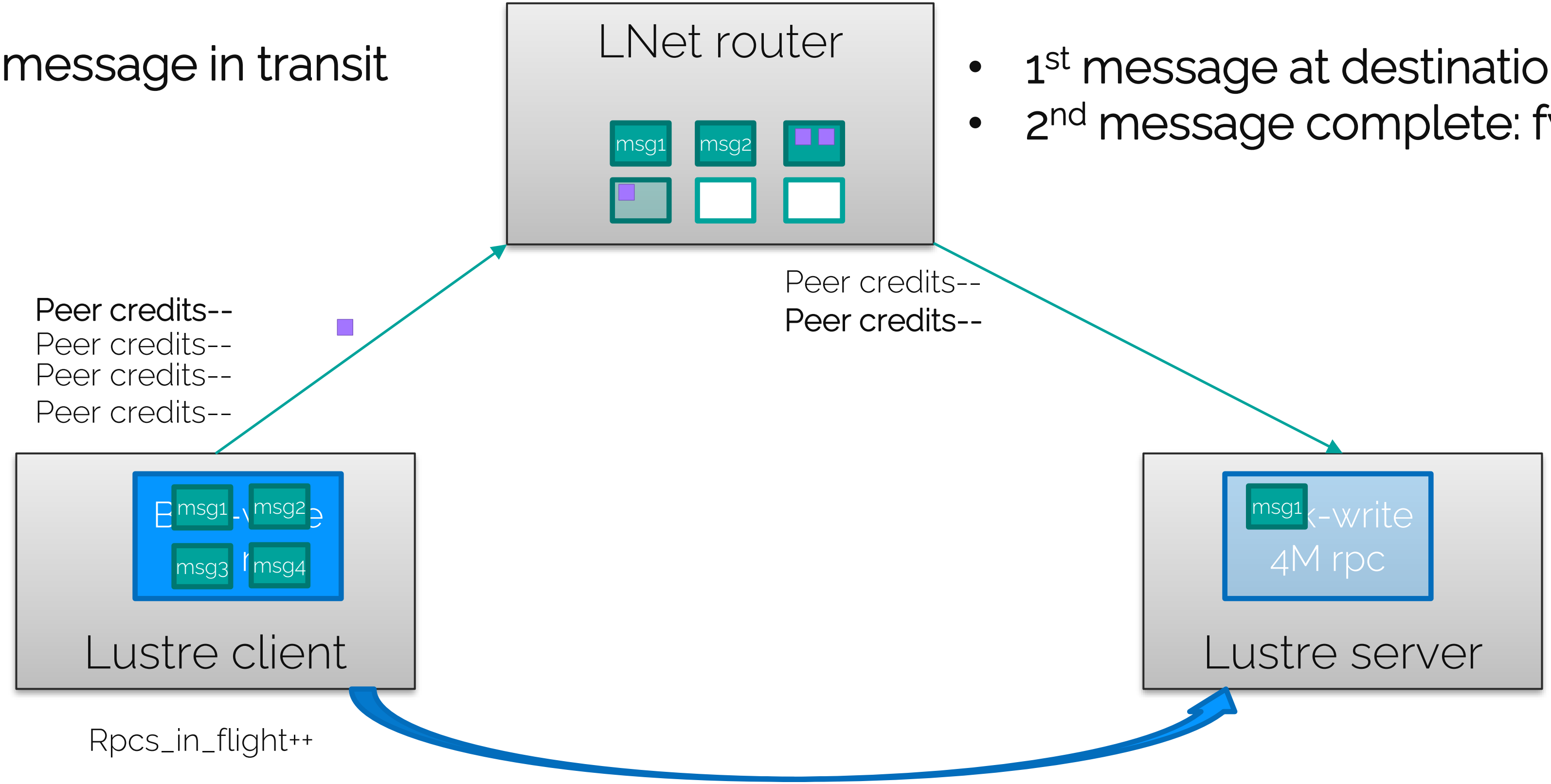


LNet router

Simplified operation description

- 4^{th} message in transit

- 1^{st} message at destination
- 2^{nd} message complete: fwd



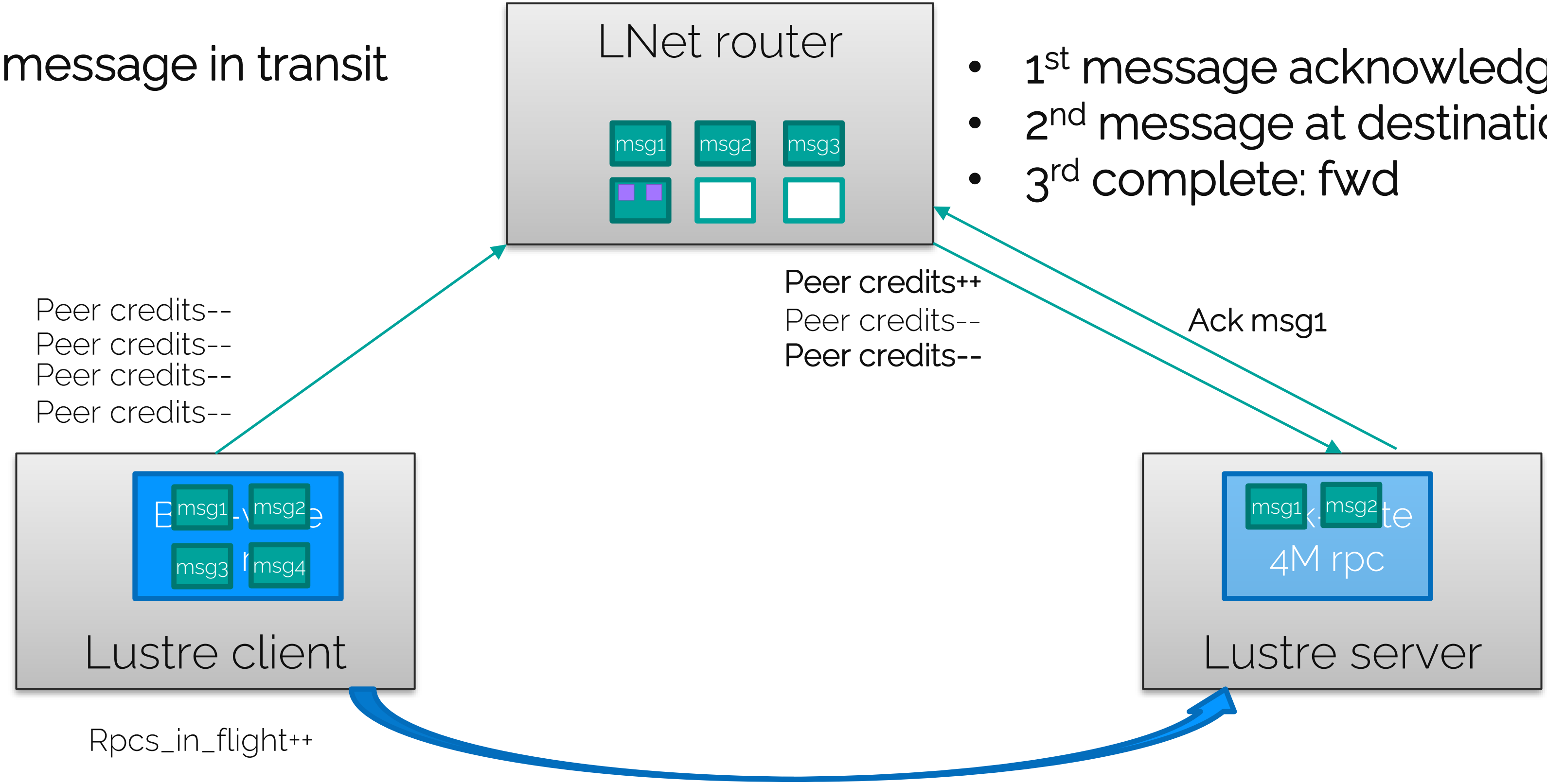
Atos

LNet router

Simplified operation description

- 4^{th} message in transit

- 1^{st} message acknowledge
- 2^{nd} message at destination
- 3^{rd} complete: fwd

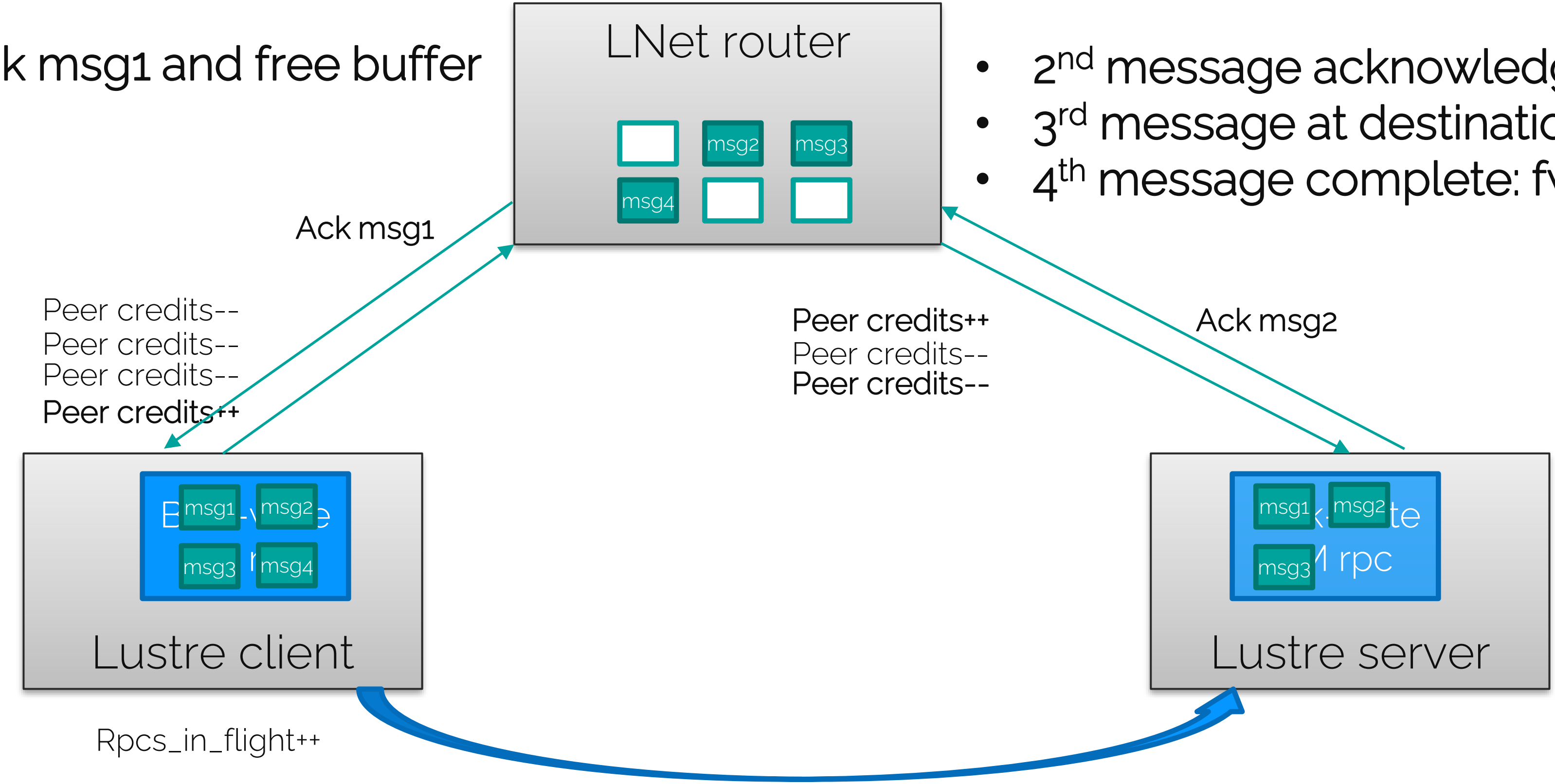


LNet router

Simplified operation description

- Ack msg1 and free buffer

- 2nd message acknowledge
- 3rd message at destination
- 4th message complete: fwd



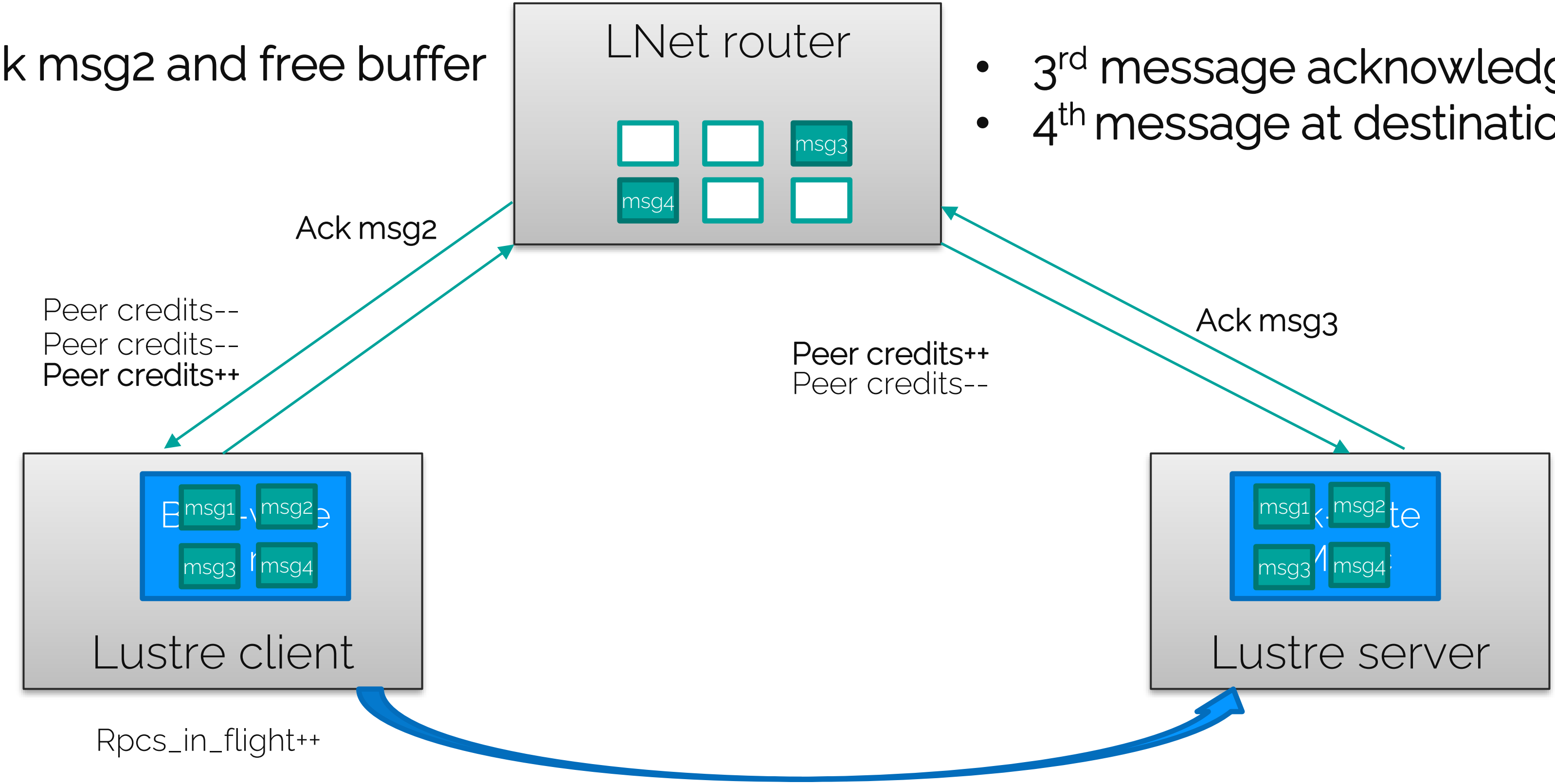
Atos

LNet router

Simplified operation description

- Ack msg2 and free buffer

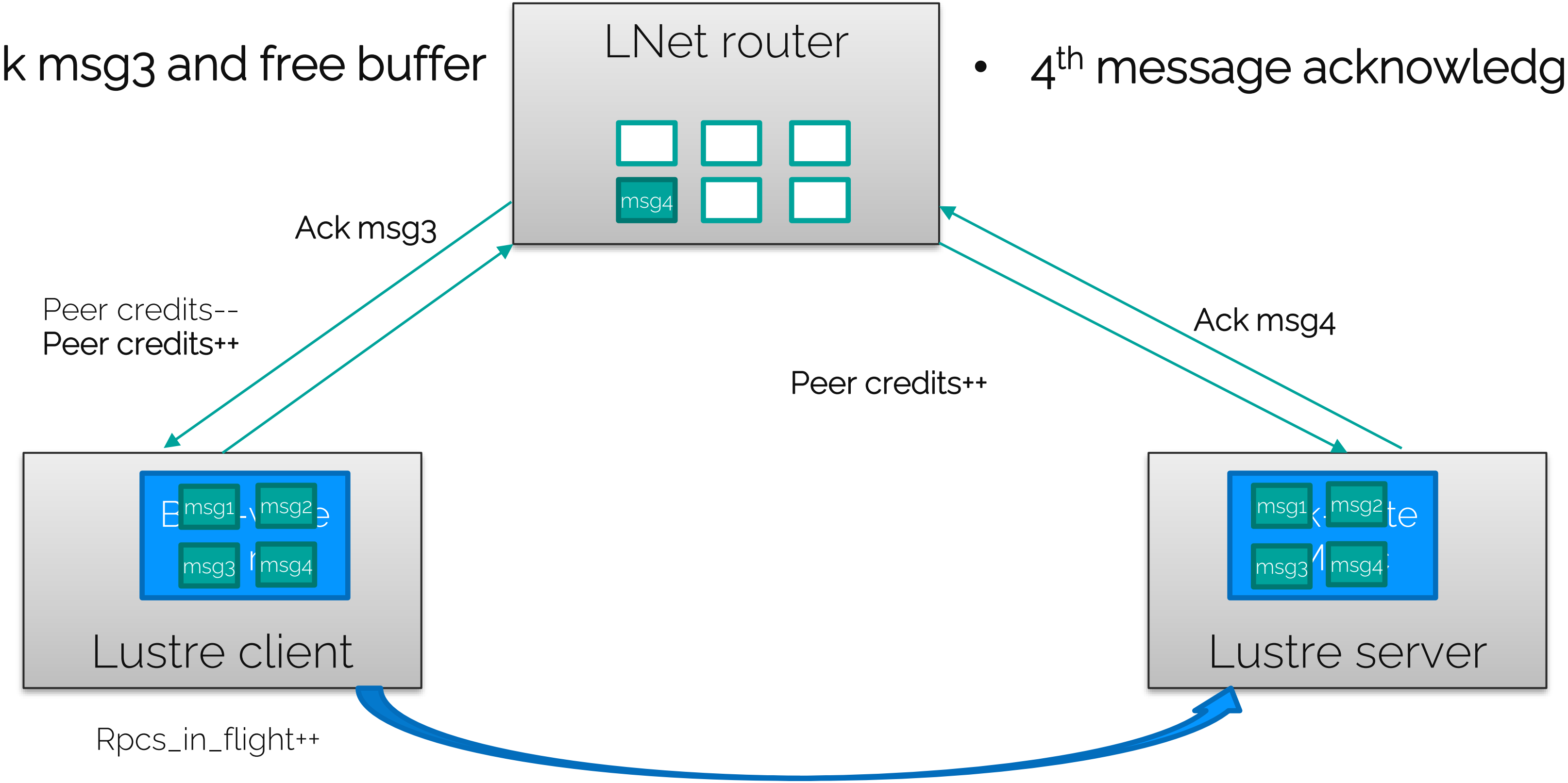
- 3rd message acknowledge
- 4th message at destination



LNet router

Simplified operation description

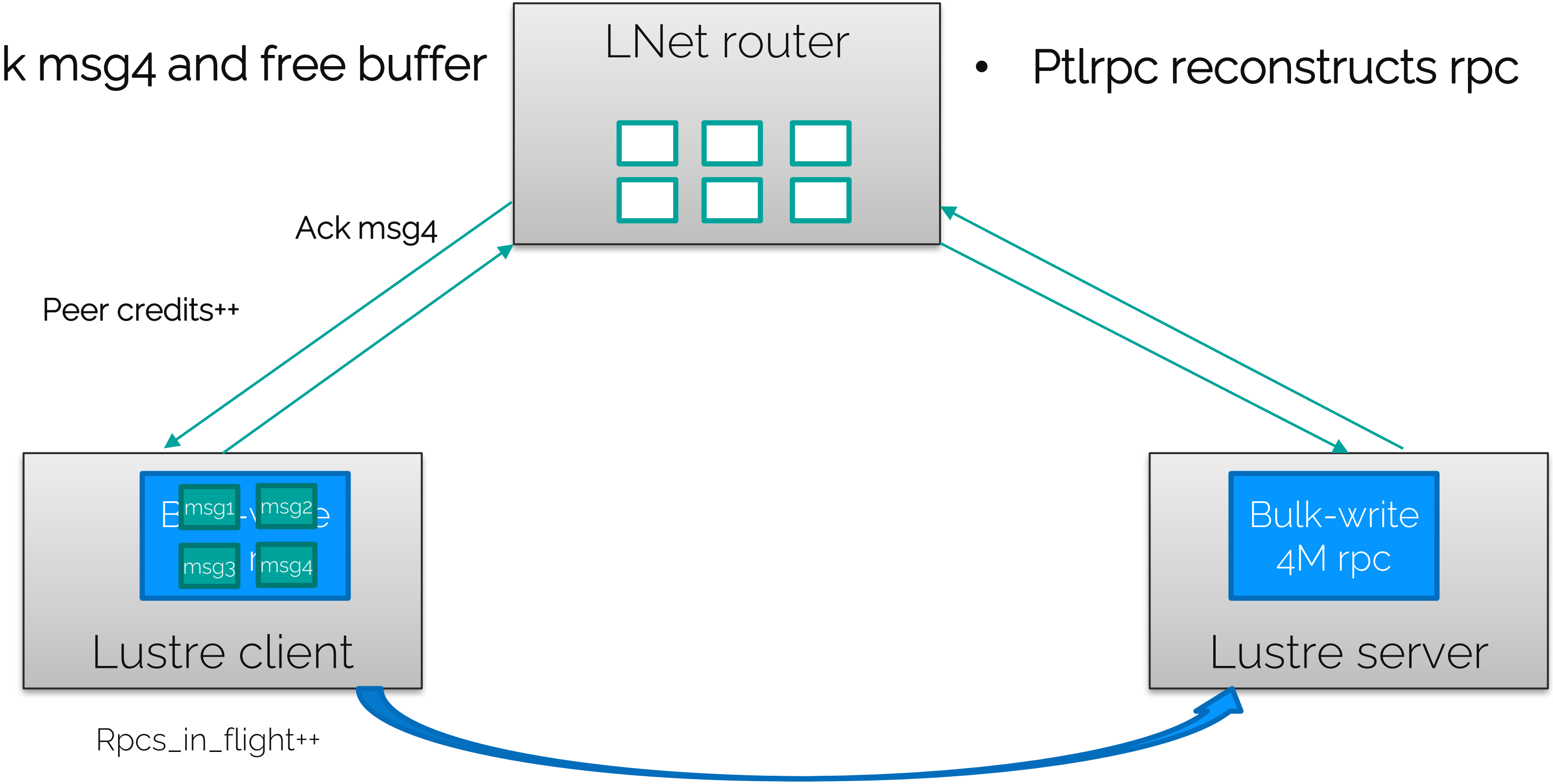
- Ack msg3 and free buffer
- 4th message acknowledge



LNet router

Simplified operation description

- Ack msg4 and free buffer
- Ptlrpc reconstructs rpc



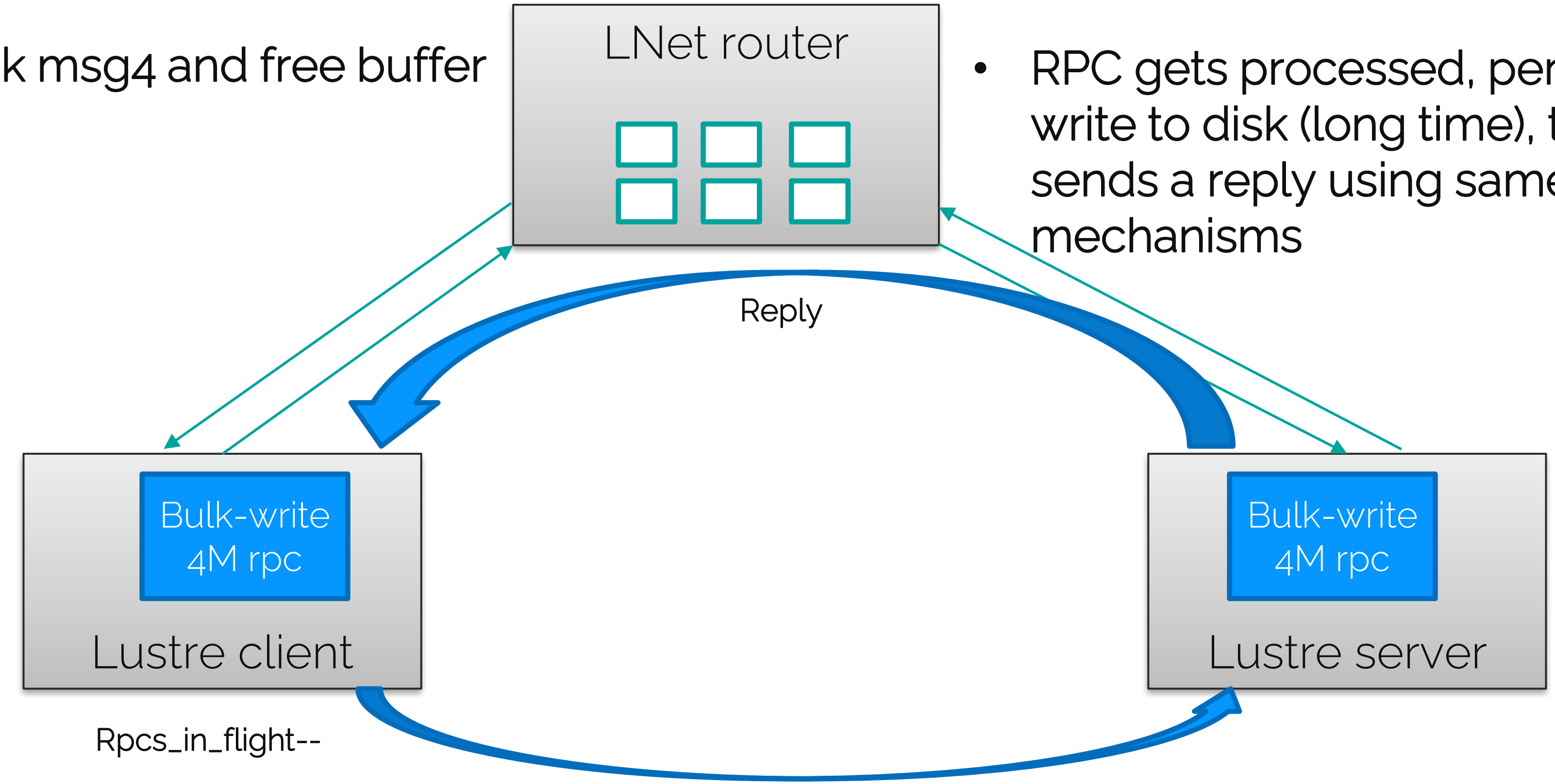
Atos

LNet router

Simplified operation description

- Ack msg4 and free buffer

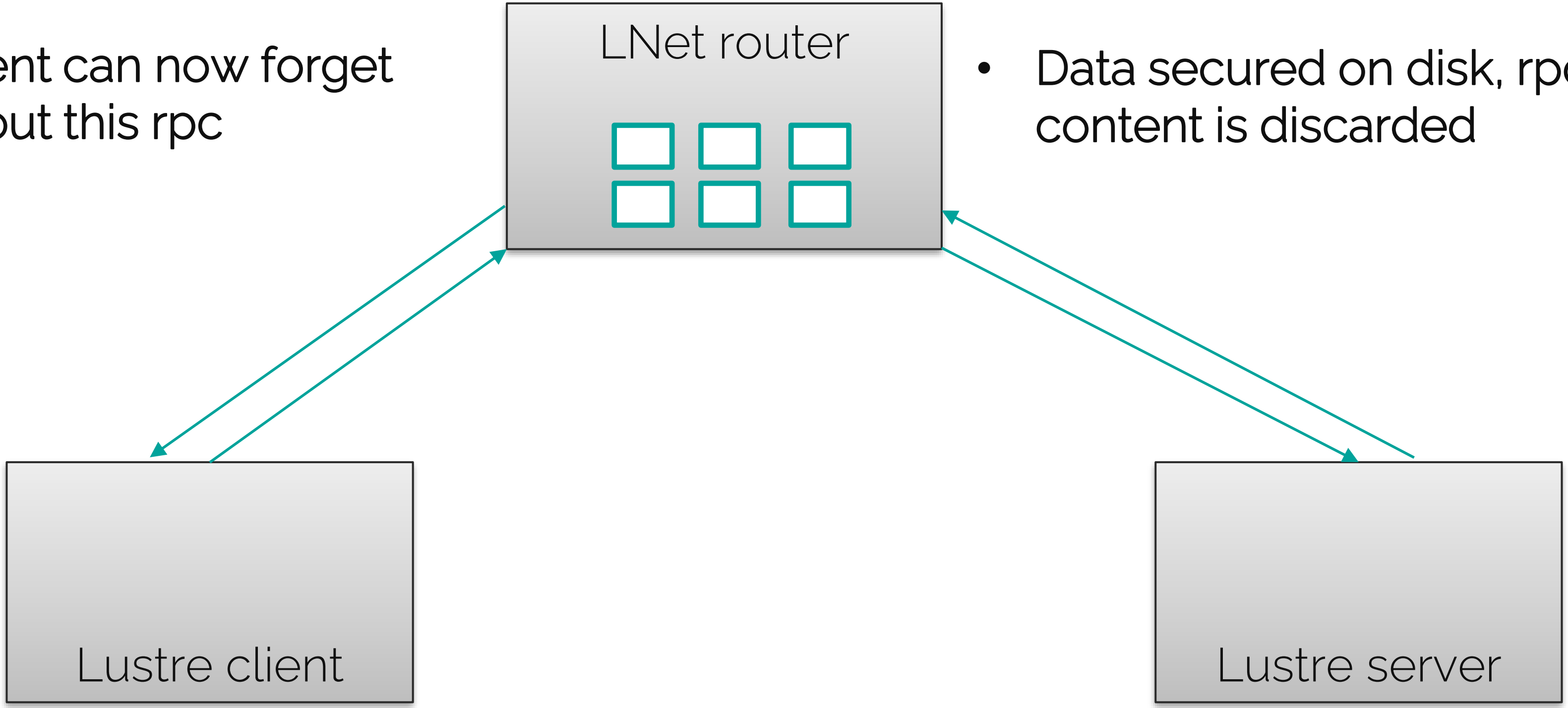
- RPC gets processed, performs write to disk (long time), then sends a reply using same Lnet mechanisms



Atos

LNet router
Simplified operation description

- Client can now forget about this rpc



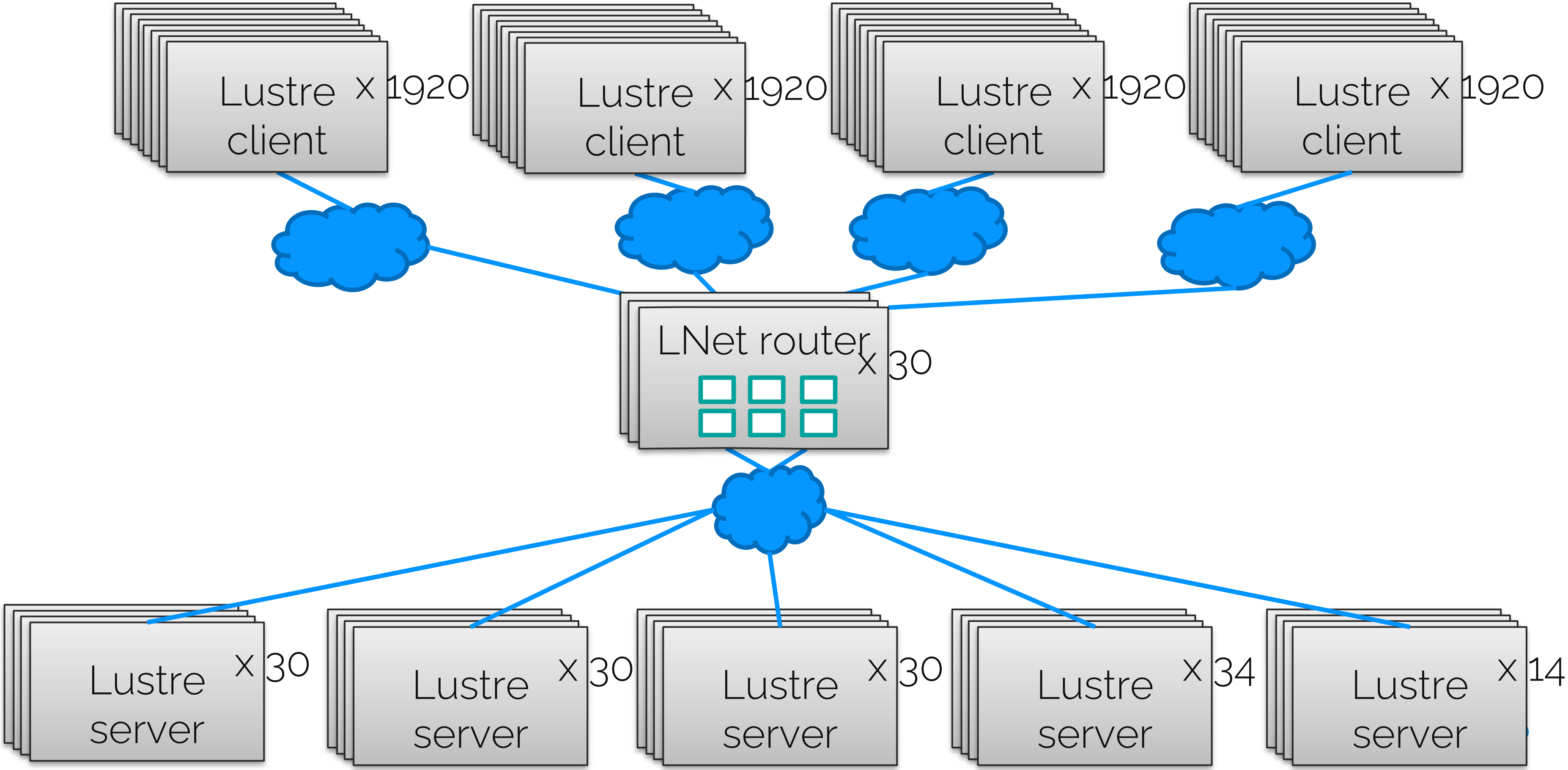
- Data secured on disk, rpc content is discarded

In practice

Atos

In practice

At ECMWF: first attempt



In practice

At ECMWF: first attempt

Frequent occurrences of:

```
LNetError: (o2iblnd_cb.c:3506:kiblnd_check_conns()) Timed out RDMA with X.X.X.X@o2ib20
LNet:(o2iblnd_cb.c:413:kiblnd_handle_rx()) PUT_NACK from X.X.X.X@o2ib20
```

-> Client evictions

=> Dirty page discards

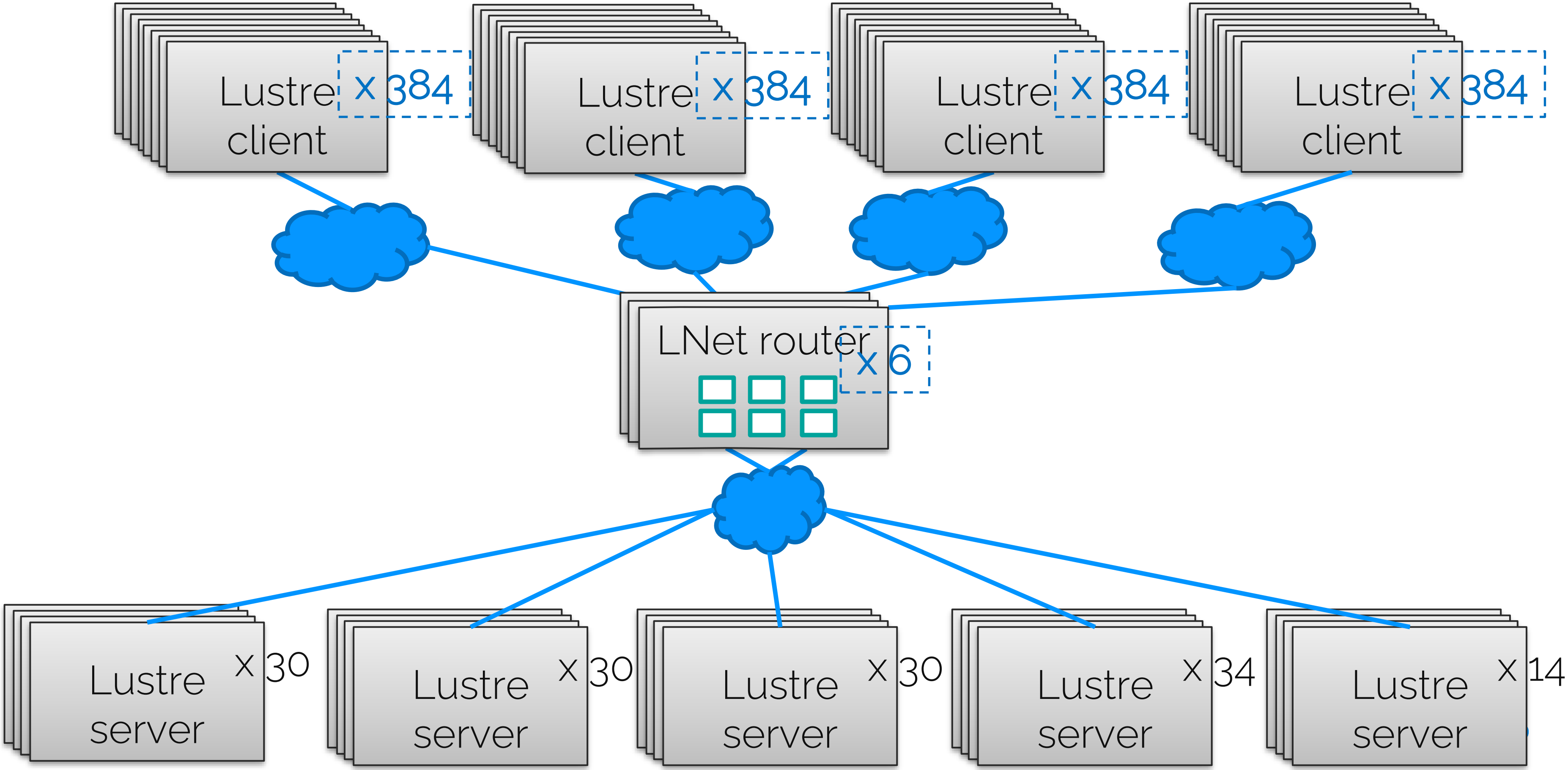
=> I/O errors on applications

- First analysis:
  - As all clients use all routers, clients are able to send a large number of messages (#clients x peer_credits x #routers) at destination of the servers, but servers can only service so many message at once => RDMA timeouts

- Mitigation: reduce number of clients per router
  - Assign 6 routers to each islets of 384 clients
    - Get the closest routers from a topology point of view to also limit IB routing congestion

In practice

At ECMWF: first attempt

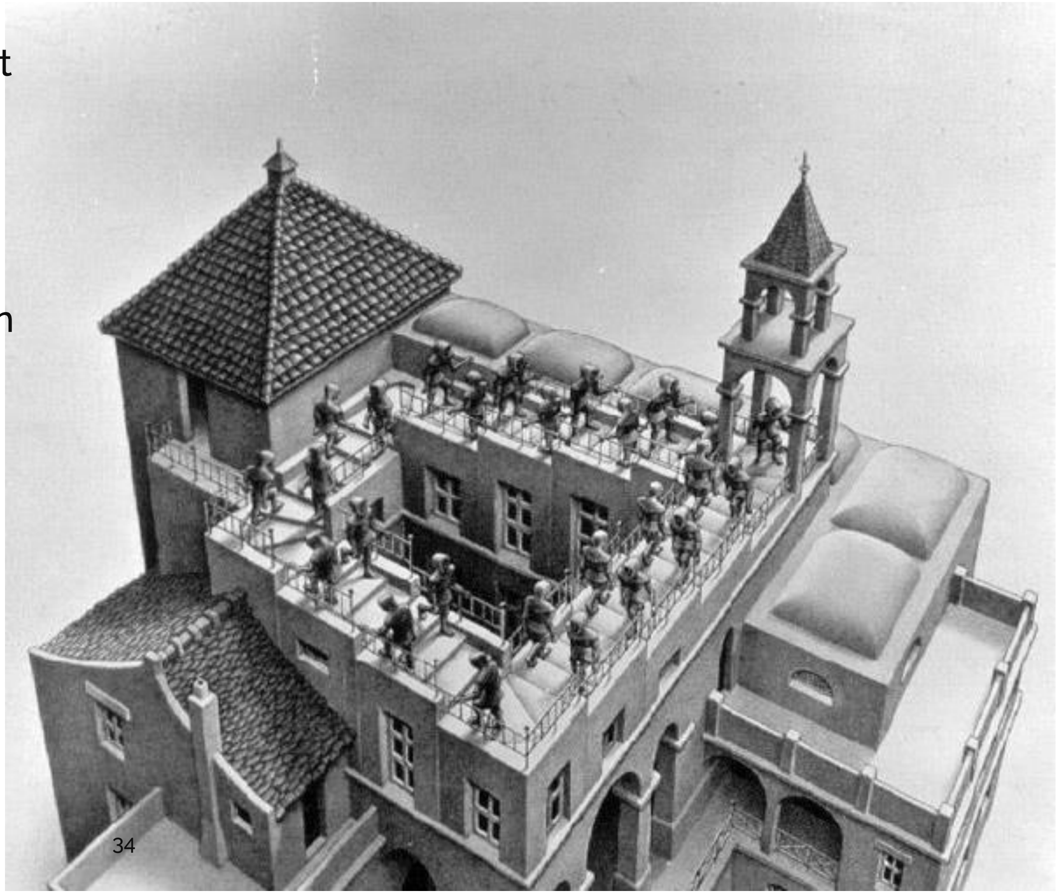


In practice

At ECMWF: second attempt

- Mitigation improved reliability, but still get occurrences of RDMA timeouts
- Need to tune parameters, but each new modification causes other troubles

⇒ Have to understand relationship between parameters



Browsing through the parameters

Credits and buffers

- ko2iblnd **peer_credits**: maximum number of unacked messages sent to a single peer (router for our case)
  - set to 32 for maximum single node performance (all nodes)
- ko2iblnd **credits**: maximum number of unacked messages sent globally
  - Set high enough to use peer_credits on all facing peers
  - On clients: (#routers * peer_credits)
    - ECMWF case: 6 routers * 2 storage fabrics * 32 peer_credits = 384
  - On routers: (#clients + #servers) * peer_credits
    - ECMWF case: (384*4 + 138) * 32 = 53568 => rounded to 65536
- lnet **[tiny|small|large]_router_buffers** (routers only):
  - Pre-allocated memory to store different types of messages to be forwarded
  - Ideally there should be enough buffers to accomodate for maximum number of simultaneous messages
    - **(#clients + #servers) x peer_credits**
  - /!\ Has to fit in the router memory: Large = 257 pages (~1MB); small = 1 page (4kB); tiny = only a few bytes
  - ECMWF case: (384 clients * 4 clusters + 138 servers) * 32 = 53568 => rounded to 65536 (~64GB of RAM)

Atos

Browsing through the parameters

Low layers timeouts

- Opensm **subnet_timeout** (default 18): An IB packet stalled on a port for more than 4.096 * 2^18 microseconds = ~1 second is dropped. Retransmission of the packet is retried 7 times
- lnet **lnet_transaction_timeout | lnet_retry_count**
  - Timeout and number of retransmission for a single message
    - A retransmission is attempted every (lnet_transaction_timeout – 1)/(lnet_retry_count + 1)
  - The retransmission should occur after the IB packet drop timeout above
  - As each message transmission is sent on a different router in a round-robin fashion, having enough retransmission to try every configured router increases probability to get a working path
  - ECMWF: timeout = 61; retry_count = 5
    - 6 attempts (=number of configured routers)
    - every 10 seconds (after the IB timeout of 7s)

Atos

Browsing through the parameters

Low layers timeouts

- lnet **live_router_check_interval**: when to ping a router to check if it is still alive (also gets its interface status to avoid routes with down paths)
- lnet **router_ping_timeout**: time after which a router is considered down if it did not reply to the ping

- **live_router_check_interval** + **lnet_router_ping_timeout** should be kept below (lnet_transaction_timeout-1)/(lnet_retry_count+1)*lnet_retry_count
  - If router fails at time of first message transmission, it is possible to detect and set router as down before the last transaction retry
- ECMWF case: check_interval = 30 ; ping_timeout = 15
  - 30 + 15 < (61 – 1)/(5 + 1) * 5 = 50

- **dead_router_check_interval**: when to ping a dead router to check if it is still dead
  - keep same as live_router_check_interval

Browsing through the parameters

Ptlrpc adaptive timeouts

Server receives request

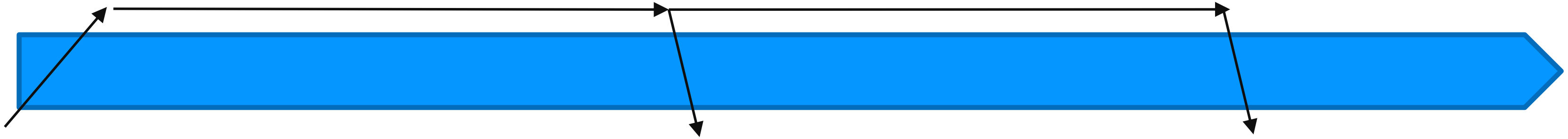
t=0+Δ

Server notifies it needs more time to service request (early_reply)
Add at_extra to deadline

t=deadline – at_early_margin

Server done processing request, sends reply

t < new_deadline



t=0

Client sends an rpc with deadline set to
at_current * 1,25 + 5
at_min <= at_service <= at_max

Client also updates
deadline += at_extra

Client updates
at_current

Atos

Browsing through the parameters

Ptlrpc layer adaptive timeouts

- ptlrpc **at_min**: minimum value for the adaptive timeout (at_current)
  - Should be higher than lnet_transaction_timeout to allow all retries to occur at lower layers
  - At ECMWF: 75 seconds
- ptlrpc **at_early_margin**: servers will send early_reply at deadline – at_early_margin
  - Should be high enough so that several attempts at lnet level can be performed during early_reply before reaching current deadline
  - At ECMWF: 25 seconds (allows for 2 retries at early_reply+10 and early_reply+20)
- ptlrpc **at_extra**: value by which the deadline is extended at each new early_reply
  - Should be higher than at_early_margin
  - At ECMWF: 50 seconds (2 * at_early_margin)
- ptlrpc **at_max**: maximum value for the adaptive timeout (at_current), there will be no early replies sent past this value
  - To be set accordingly with system's load expectations. Has an impact on recovery time if IR can't operate
  - At ECMWF: 600 seconds

Credits

- Thanks to Alexandre Louvet for performing most of the code hacking work which allowed this presentation and his always supportive presence
- Thanks to ECMWF team for giving me authorization and material to illustrate with a concrete example.

- Some wiki pages that also served during the preparation of this presentation:
- https://wiki.lustre.org/Lustre_Resiliency:_Understanding_Lustre_Message_Loss_and_Tuning_for_Resiliency
- https://wiki.lustre.org/LNet_Router_Config_Guide

Questions / Remarks

Thank you!