Throughput Oriented Runtimes for Manycore Clusters

Costin Iancu, Khaled Ibrahim

Tsukuba – March 19

We know how to build runtimes that scale with the node count (Latency)

BUT

We have little experience building runtimes that scale with cores per node (and nodes) **(Throughput)**


2012
2020

32 cores

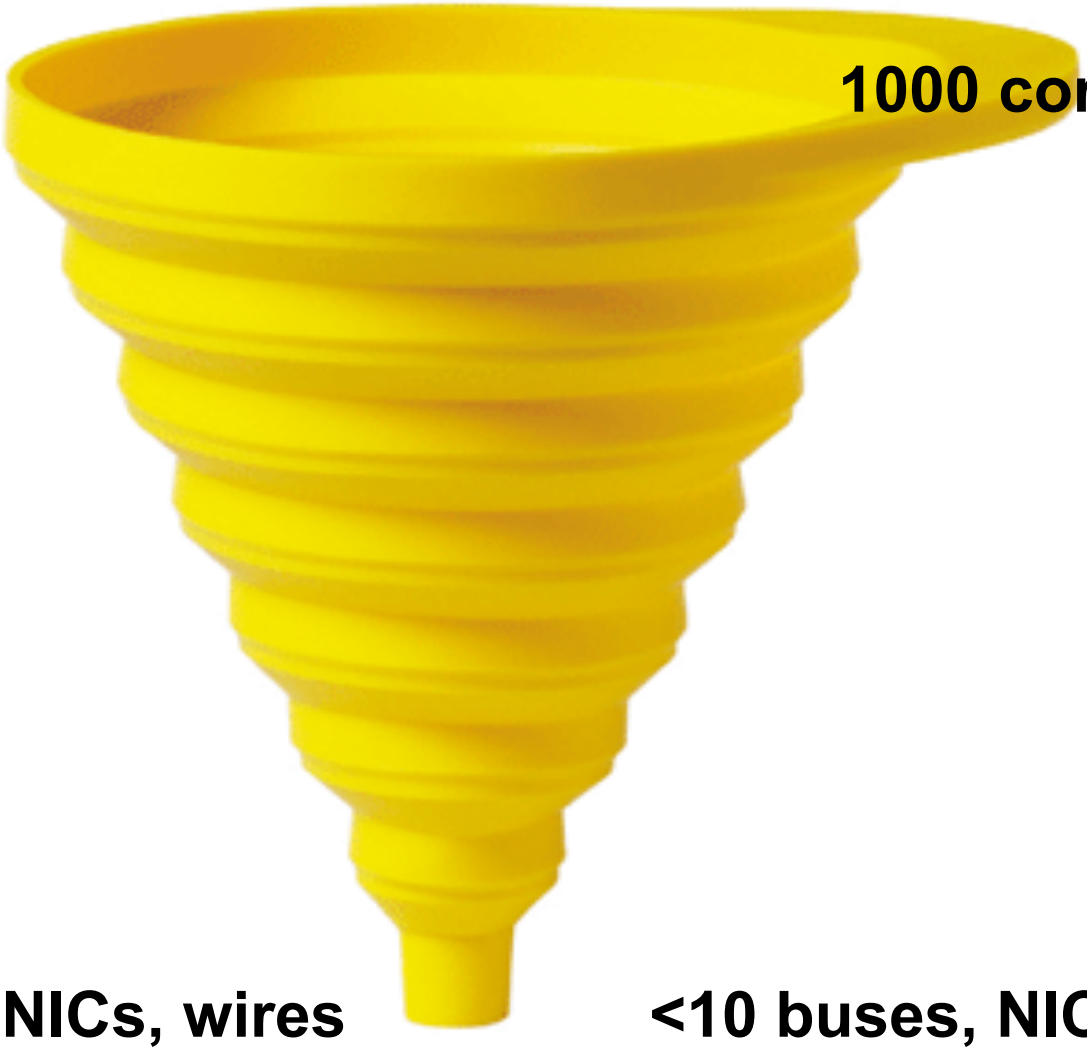
<10 buses, NICs, wires

2012

2020

32 cores

1000 cores



<10 buses, NICs, wires

<10 buses, NICs, wires

2012

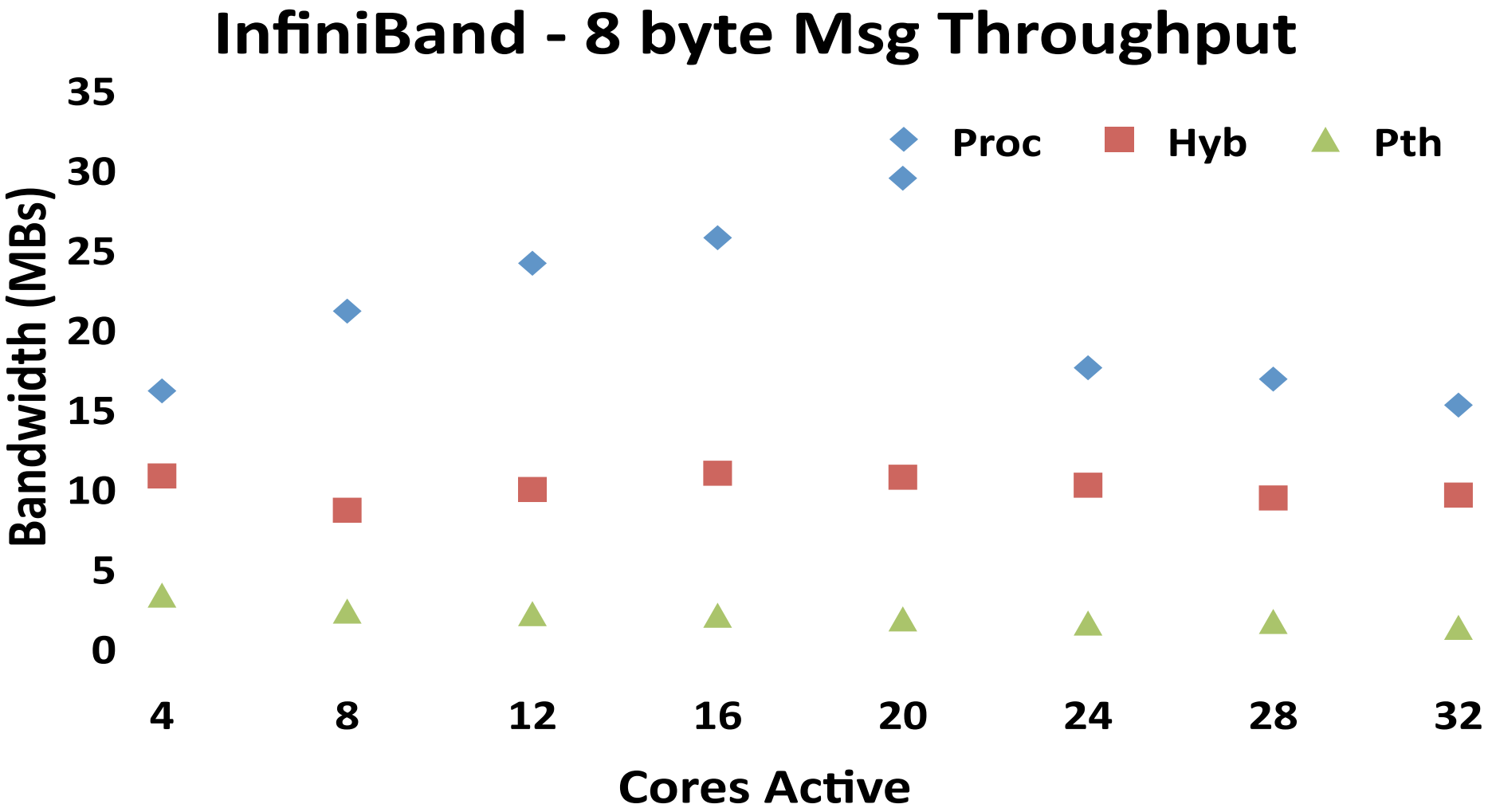
2020

Outline

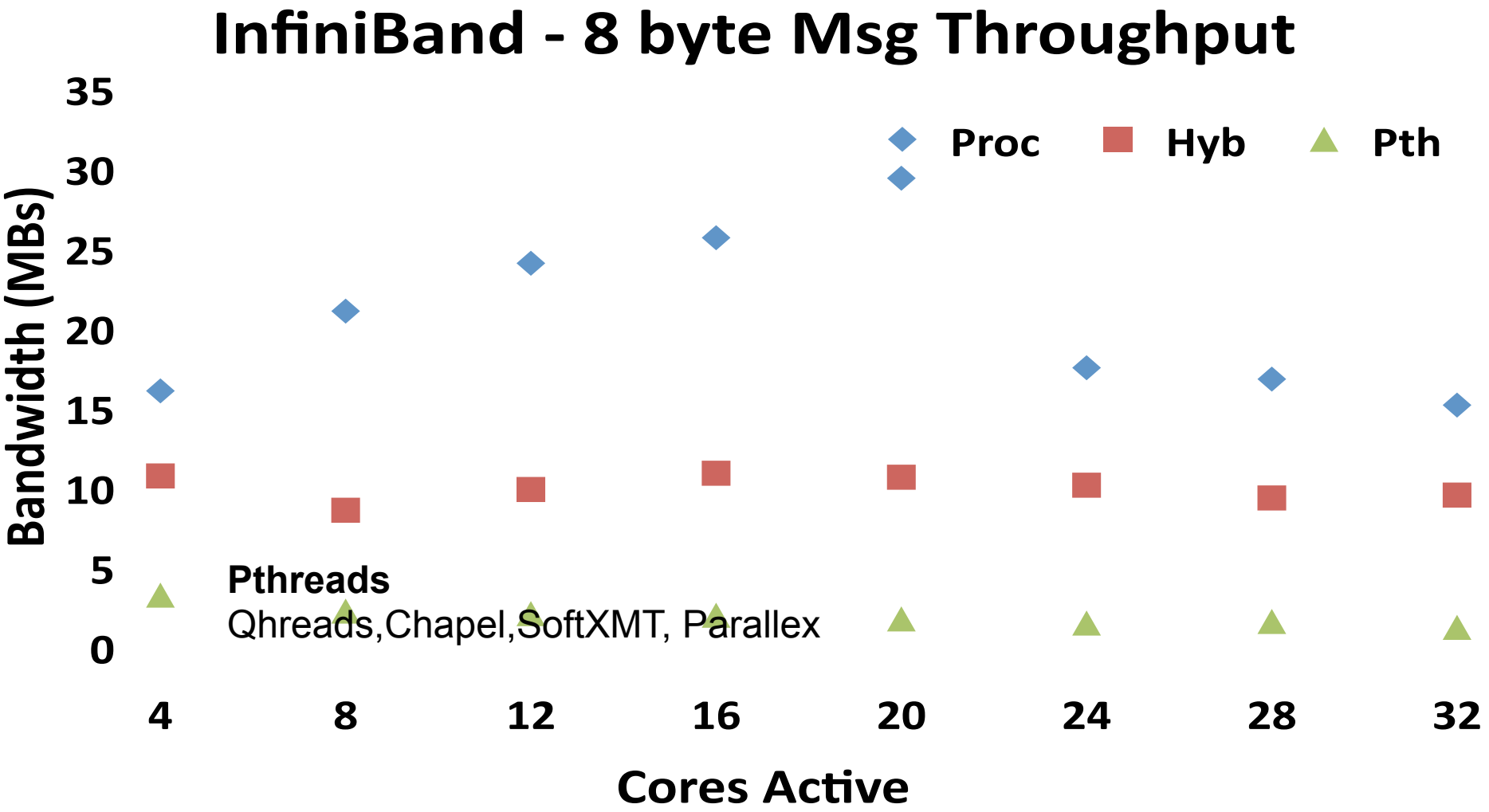
- Network performance in manycore clusters
- Designing a throughput oriented runtime
- Future work

Network Performance

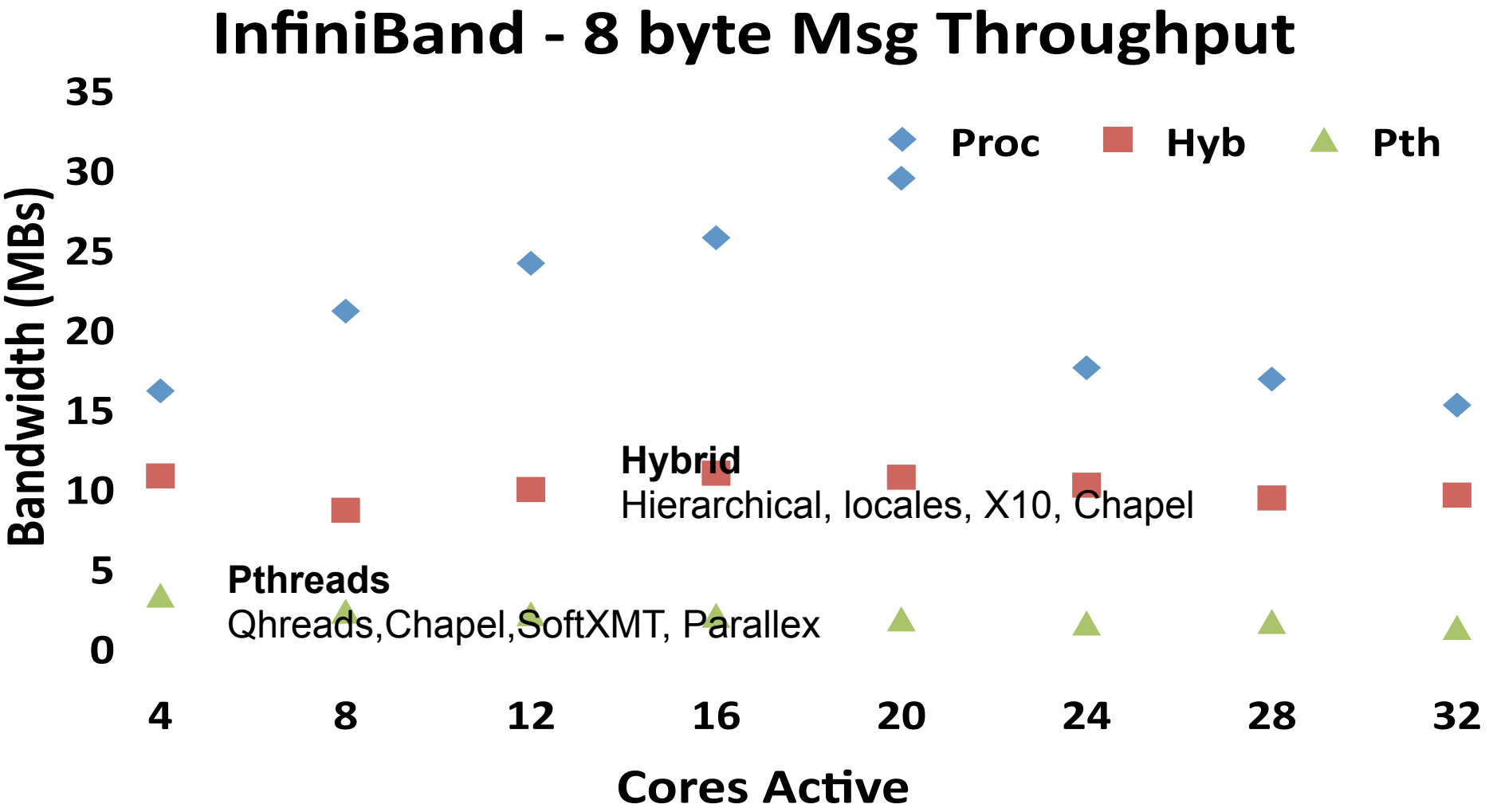
InfiniBand Performance



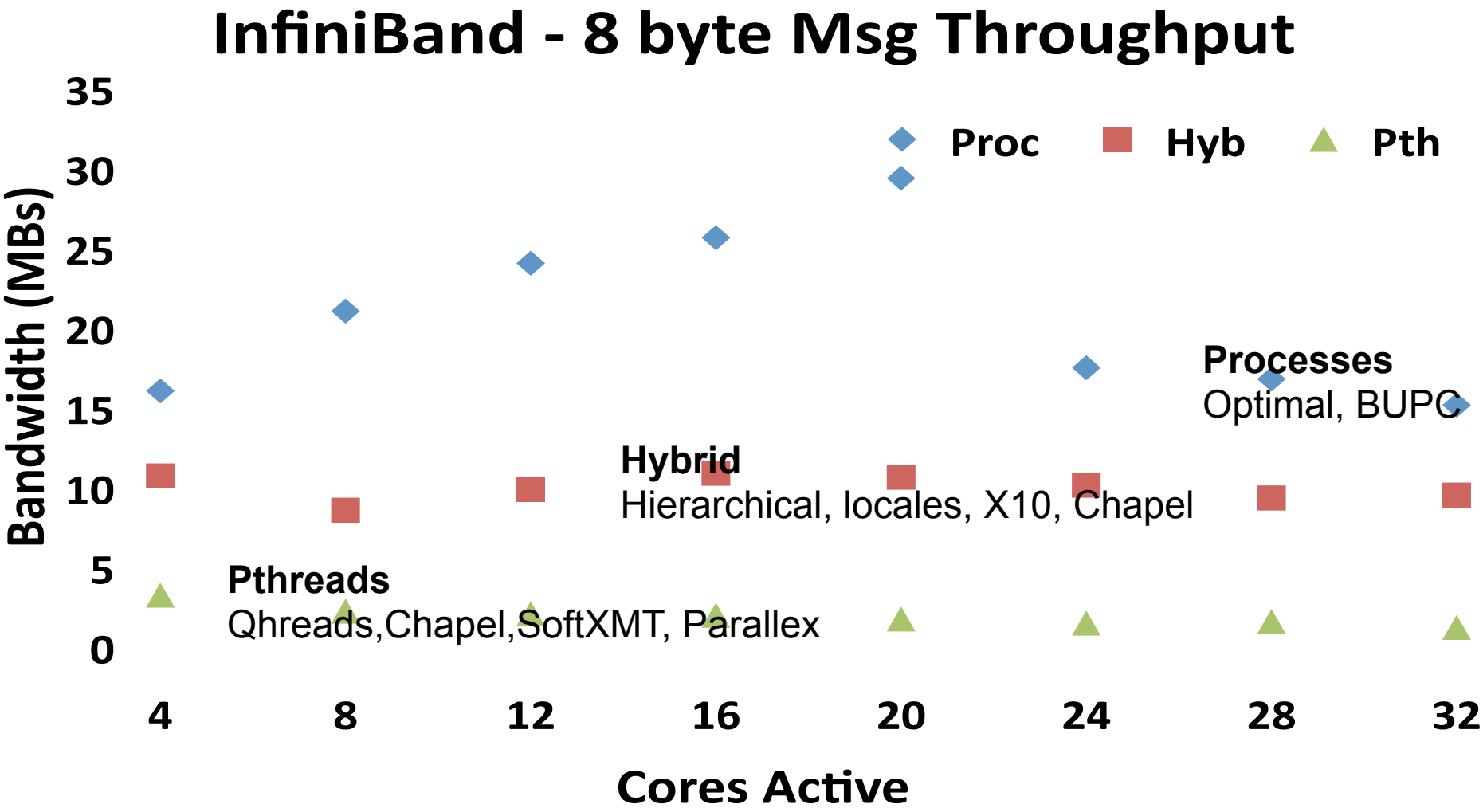
InfiniBand Performance



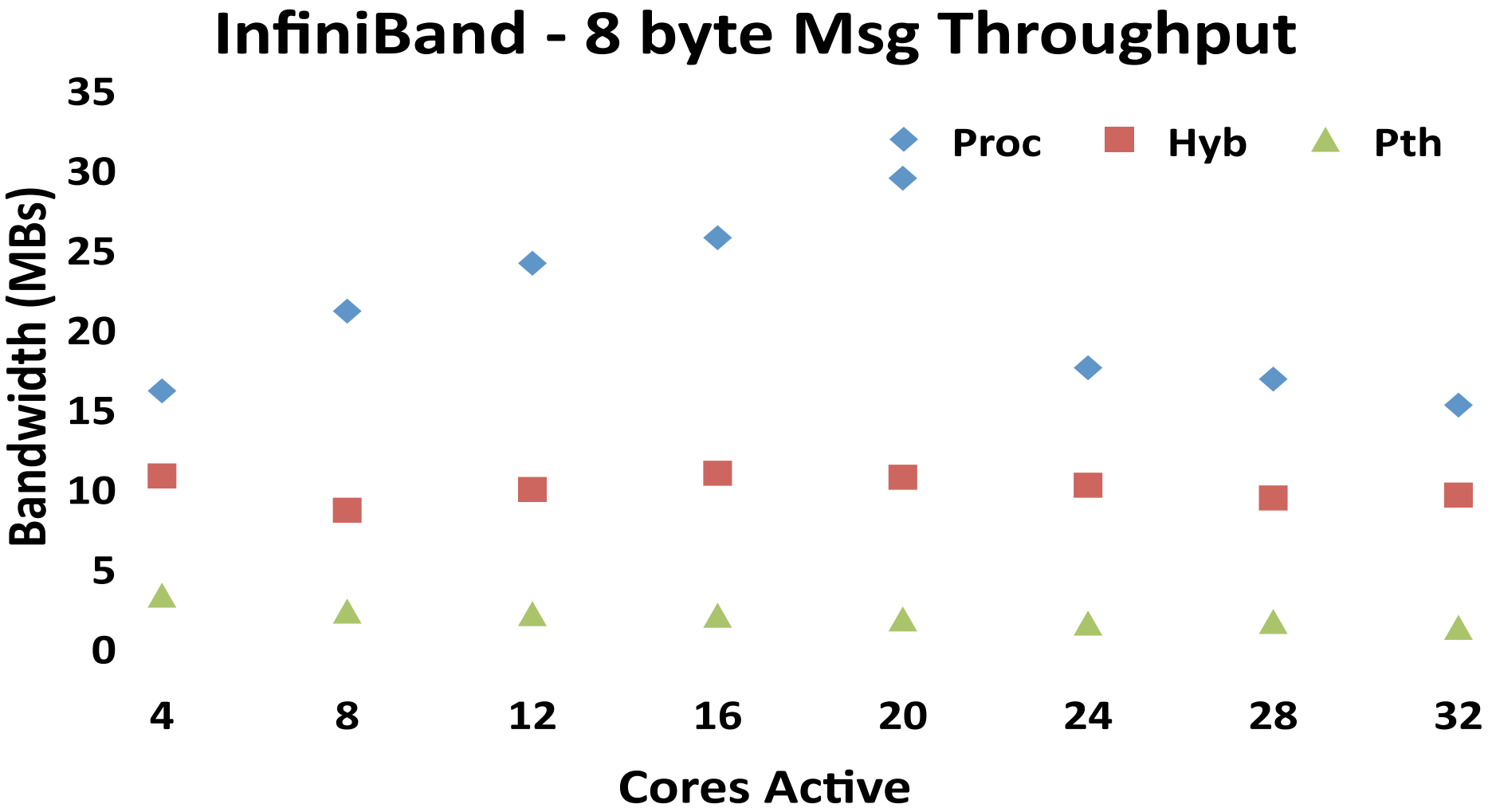
InfiniBand Performance



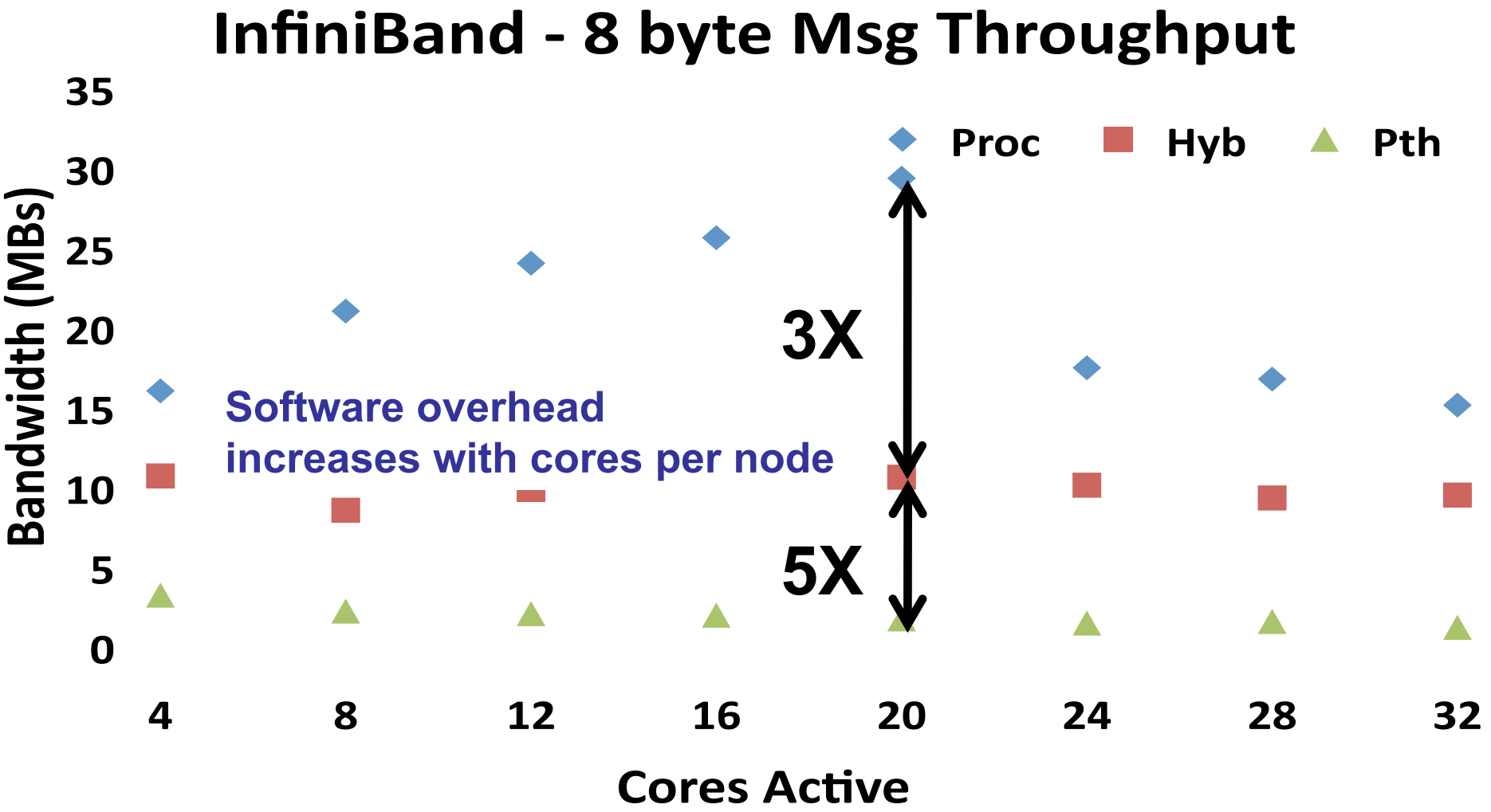
InfiniBand Performance



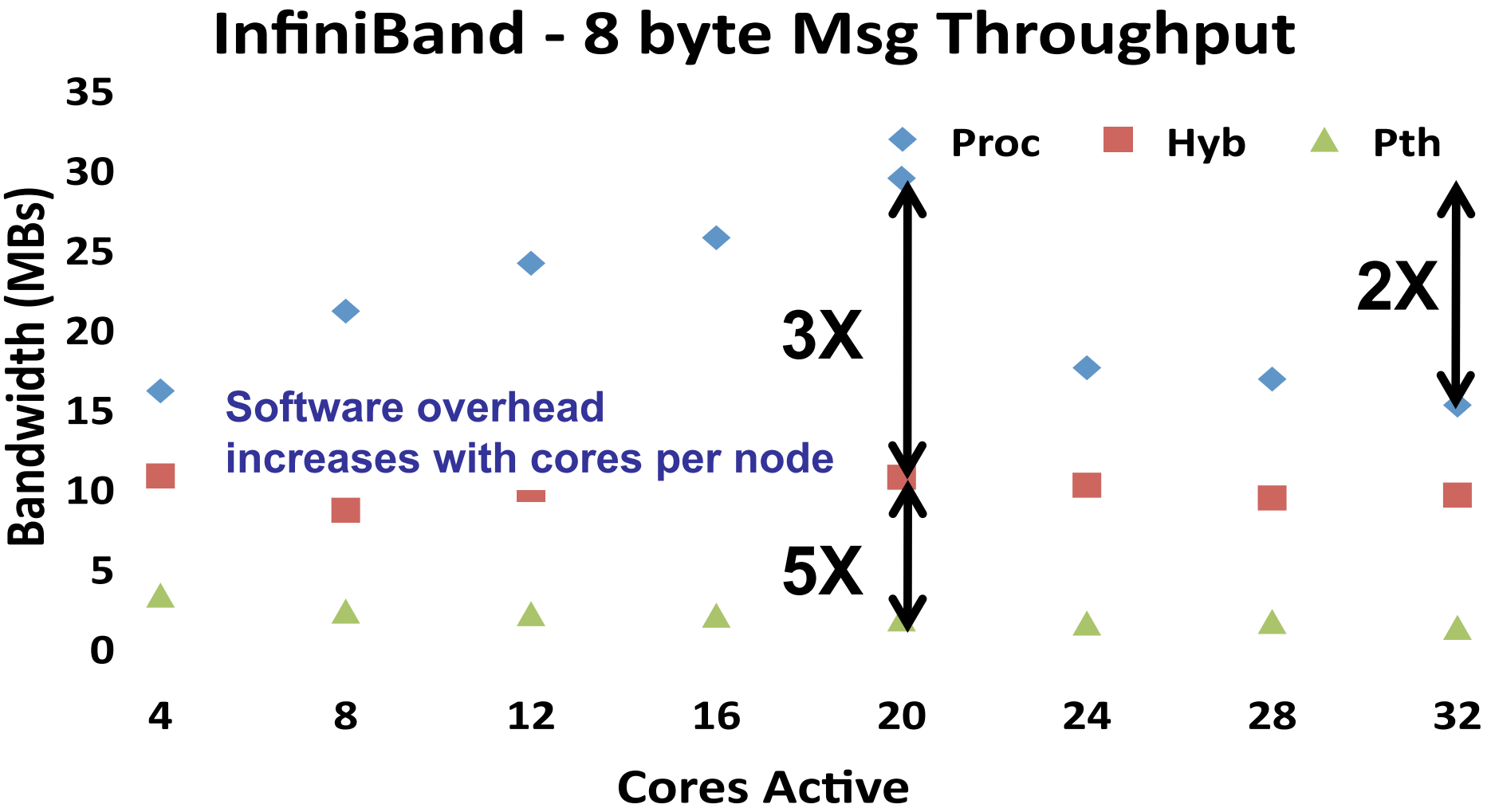
InfiniBand Performance



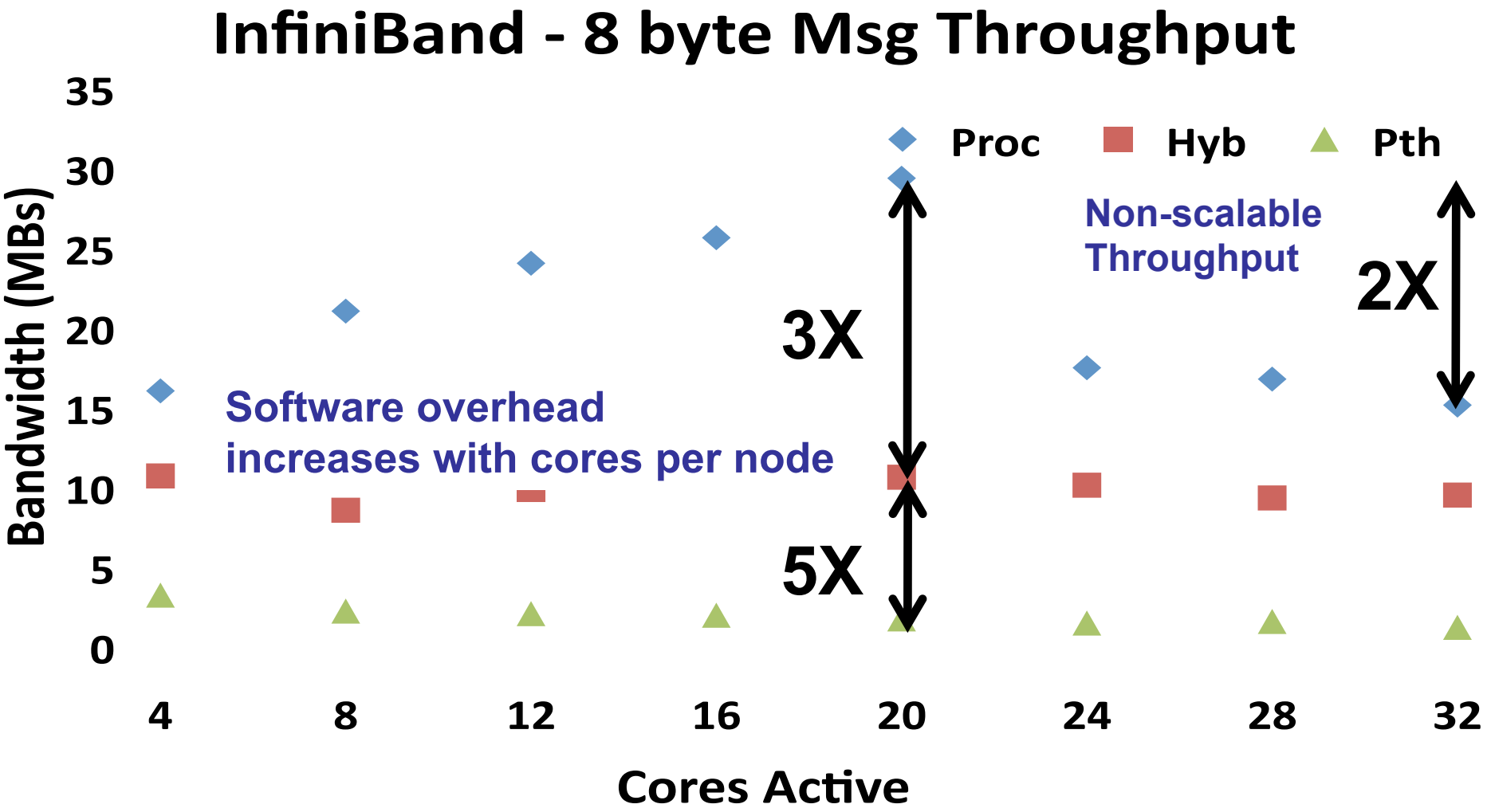
InfiniBand Performance



InfiniBand Performance

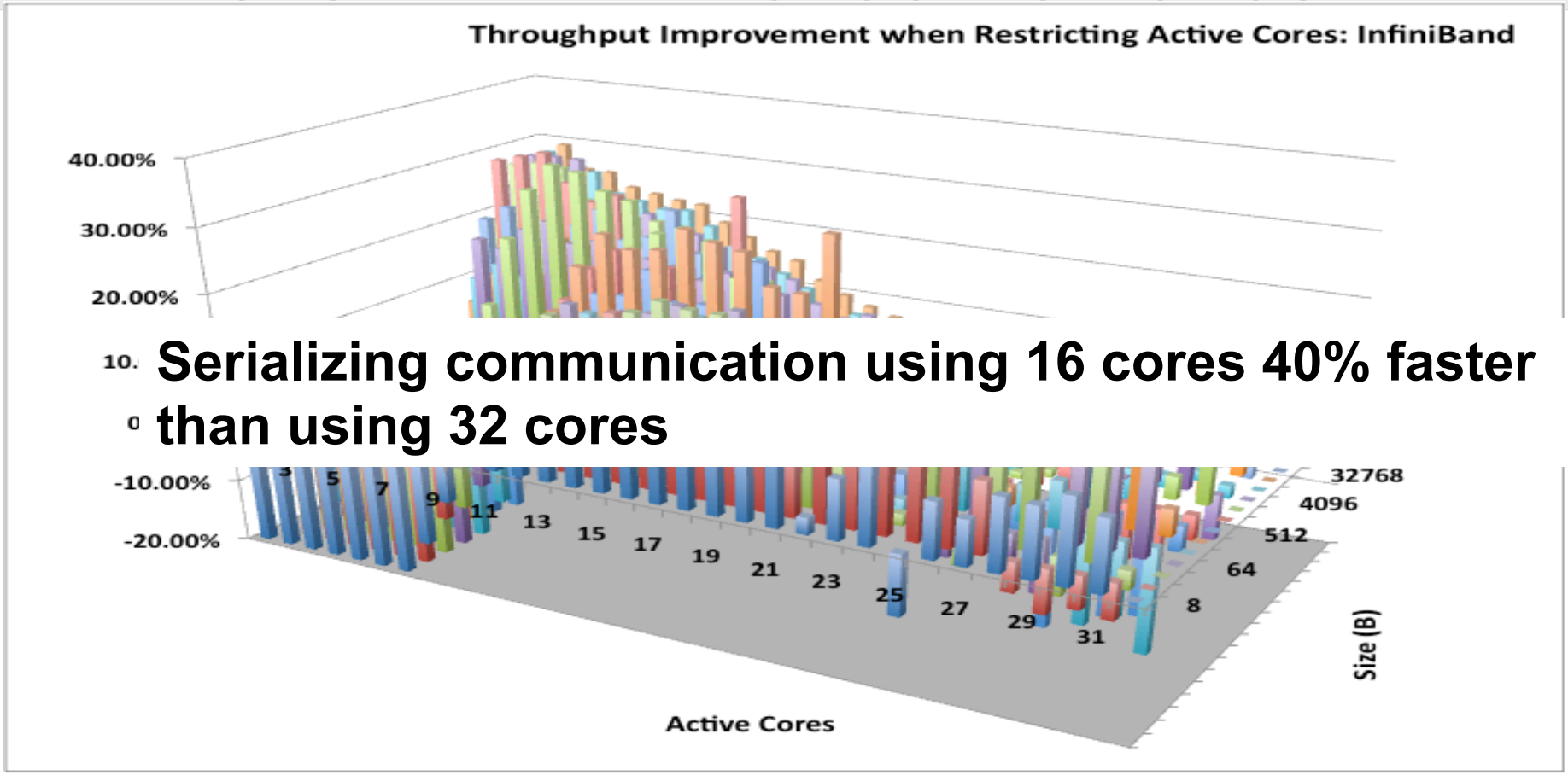


InfiniBand Performance



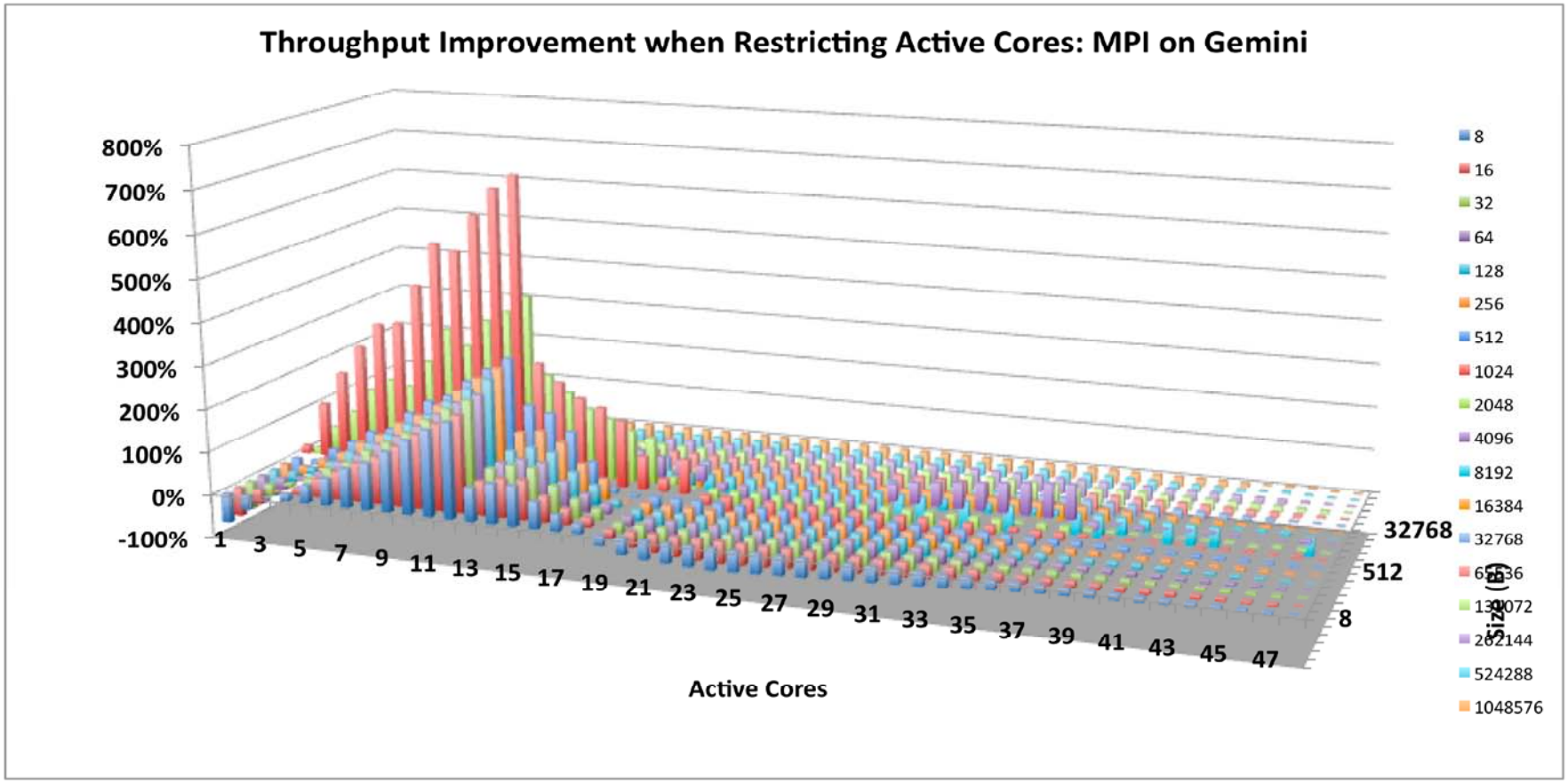
BUPC/GASNet on InfiniBand

BUPC/GASNet on InfiniBand

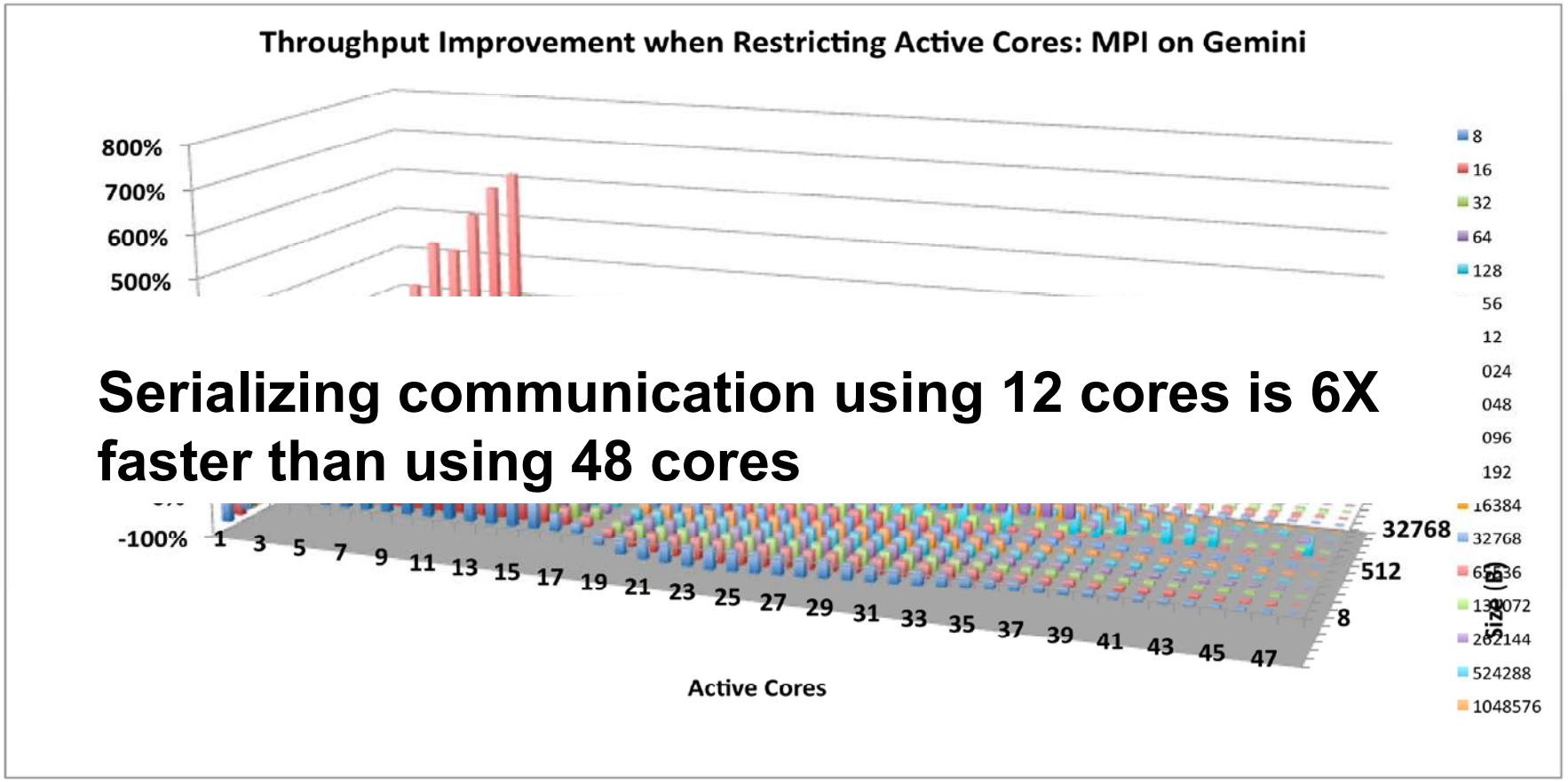


- Serializing communication using 16 cores 40% faster than using 32 cores

Cray MPI on Gemini



Cray MPI on Gemini



Serializing communication using 12 cores is 6X faster than using 48 cores

Network Performance

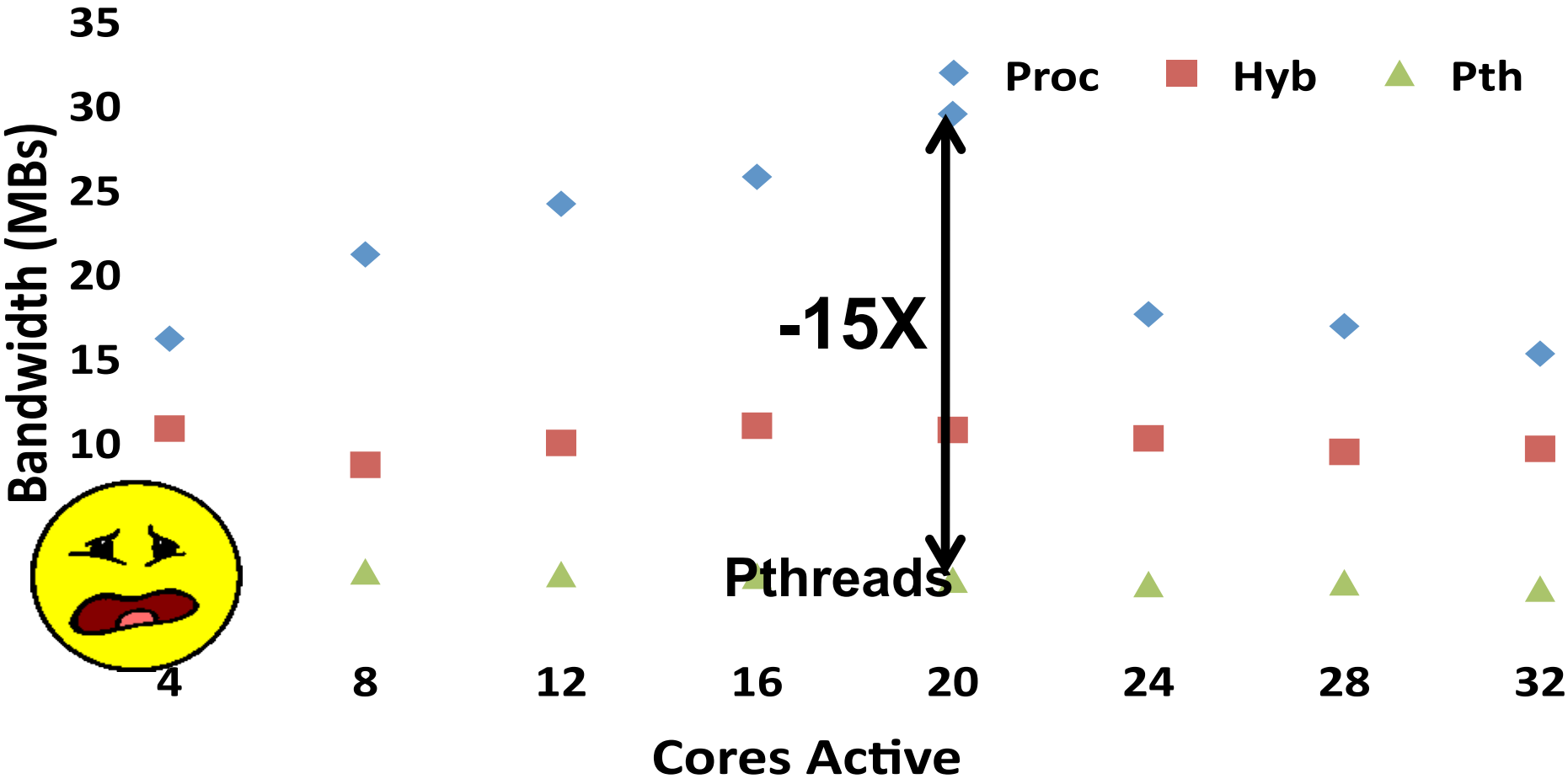
- **Modern NICs are saturated by a small number of messages (cores*messages = K)**
- **Low latency is not enough**
  - OpenIB Verbs and Cray DMAPP throughput worse than GASNet
- **Common behavior across software and hardware**
  - InfiniBand (Mellanox), Cray Gemini have similar behavior
  - MPI, OpenIB and DMAPP have similar behavior
- **Behavior is likely to get worse**

Large (Exa) Scale = Throughput

- **Future large scale systems are manycore**
  - Homogeneous or heterogeneous hardware
  - Asymmetric software architecture or performance
- **Multithreading proven to work for irregular applications**
- **Many research projects advocate "fine grained" multi-threading** (SoftXMT, Qthreads, ParalleX, SWARM)
  - Energy efficiency
  - Data movement
  - Latency hiding
  - Load balancing
  - Performance and productivity
- **Fine grained multithreading requires network throughput**

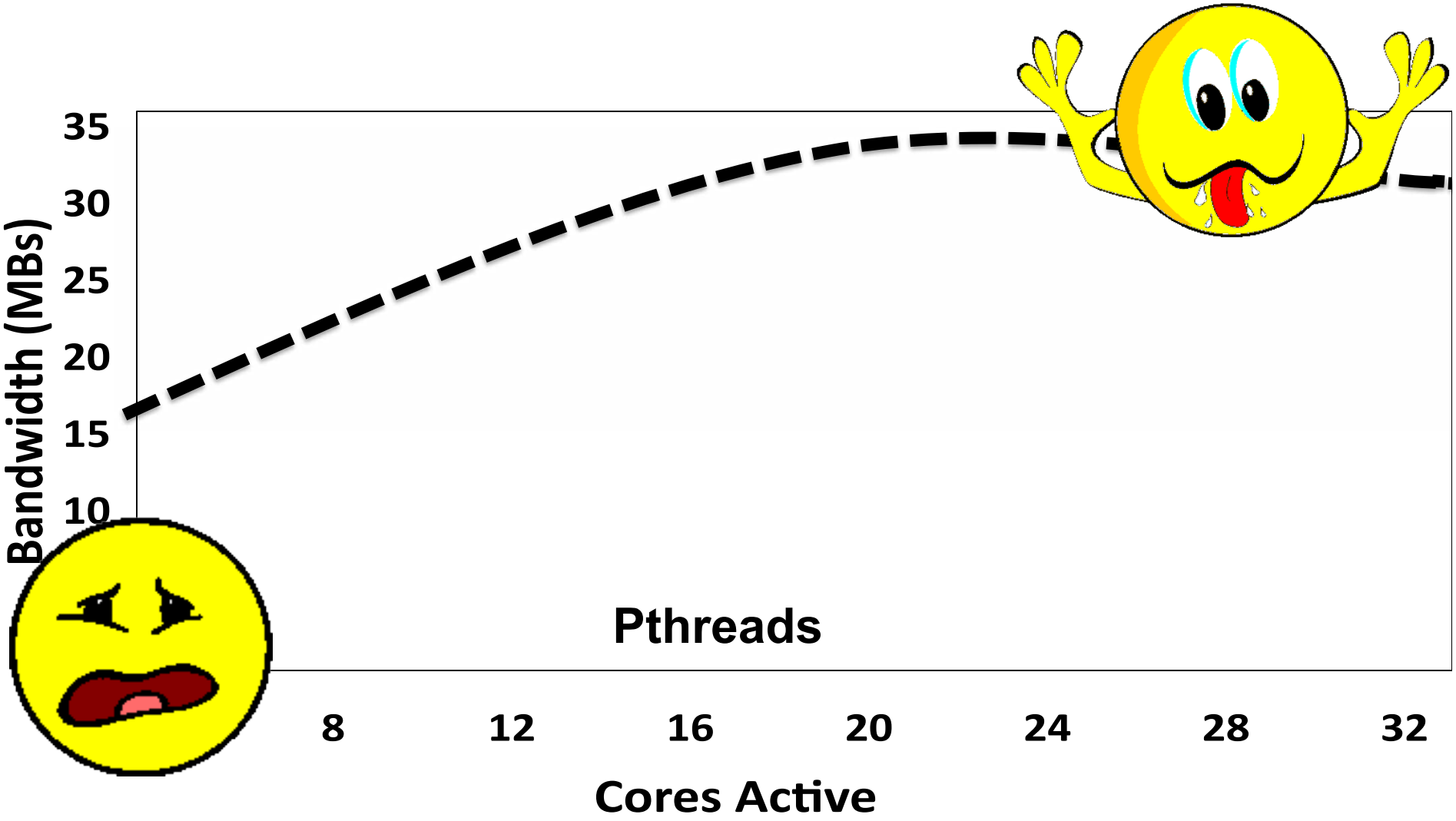
We do not know that we do not know how to build integrated "threading" and networking runtimes that provide good throughput

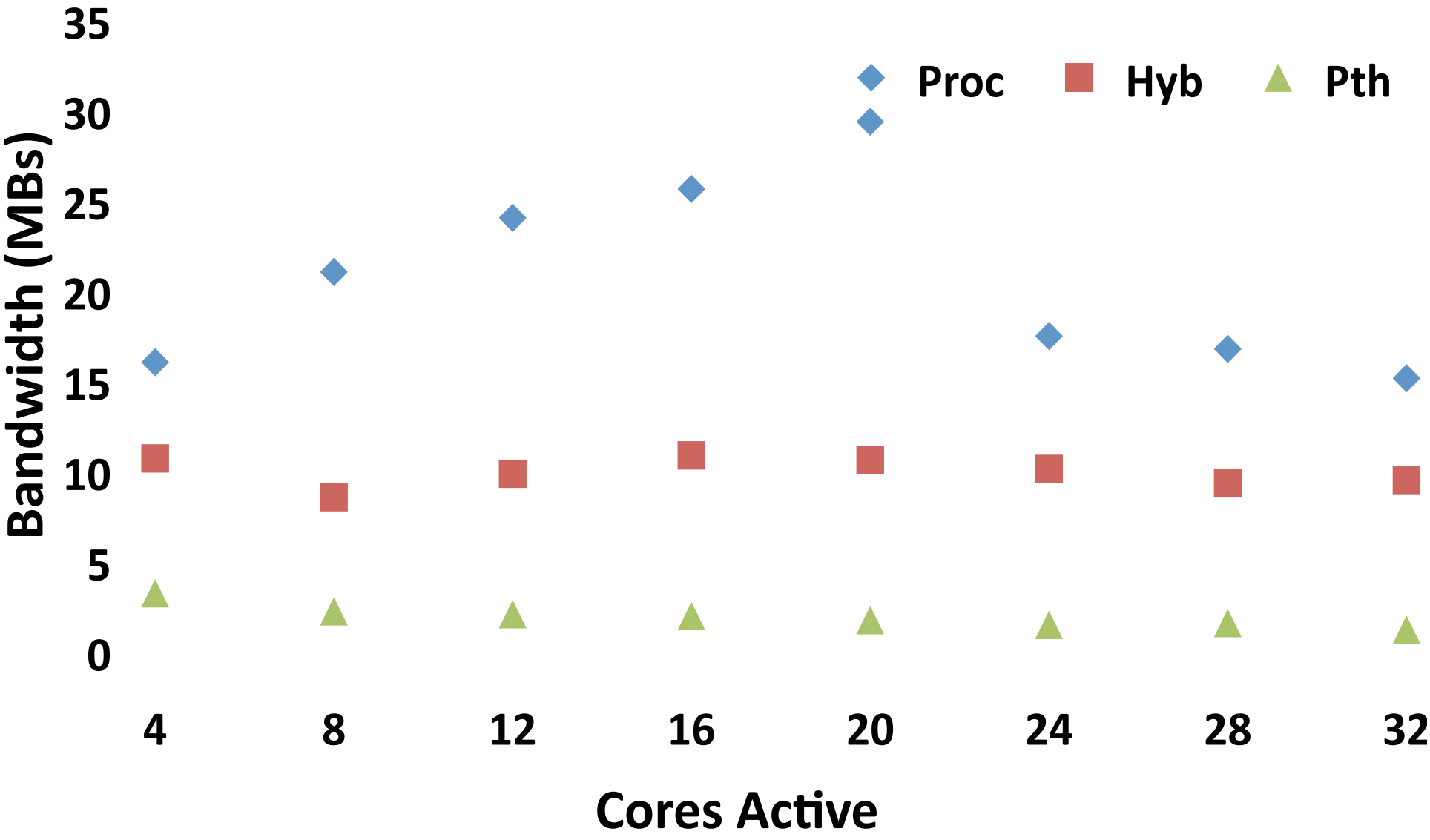
(building user level threading mapped on pthreads)

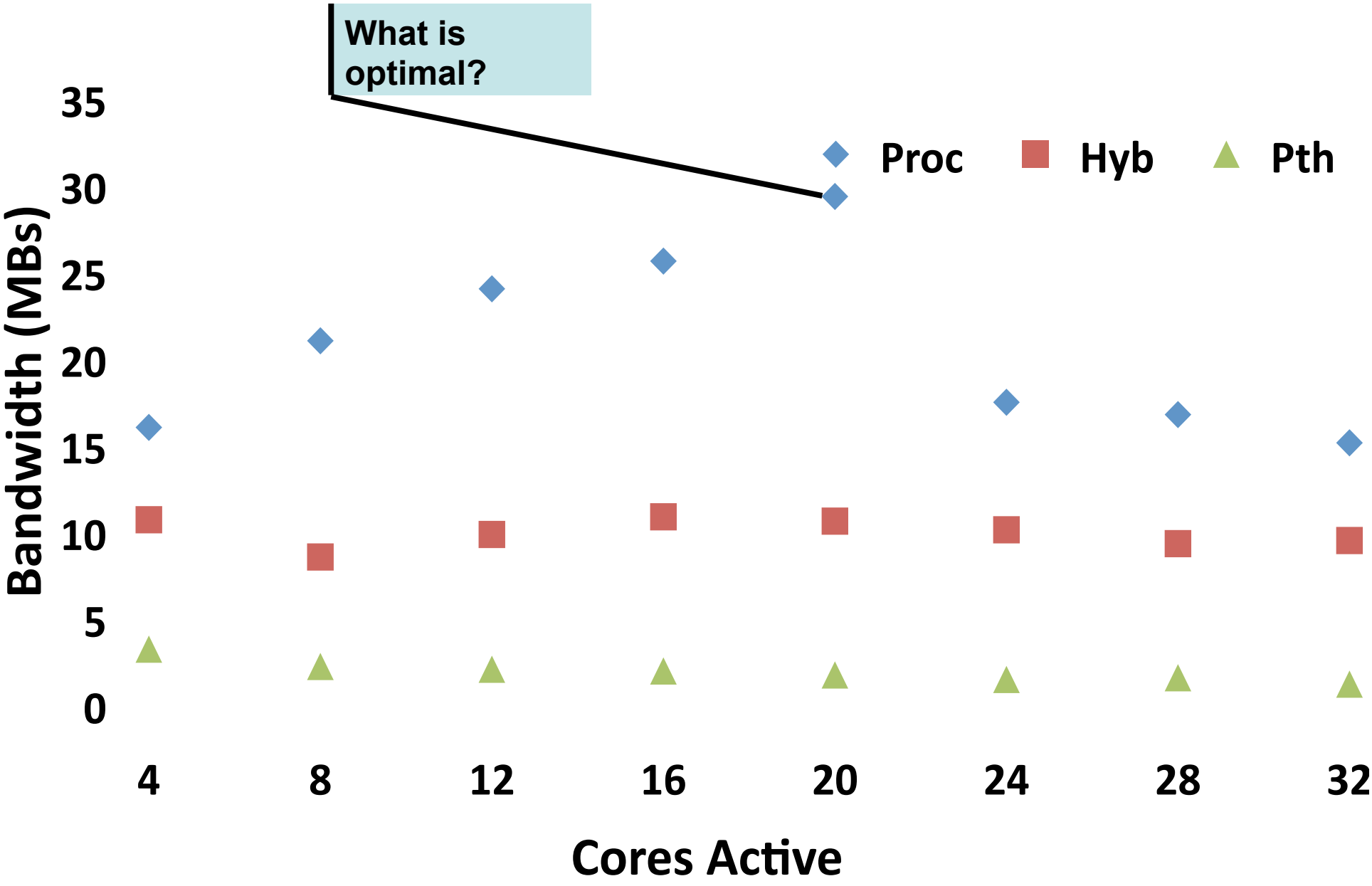

35
30
25
20
15
10
Bandwidth (MBs)
Proc
Hyb
Pth
-15X
Pthreads
4
8
12
16
20
24
28
32
Cores Active

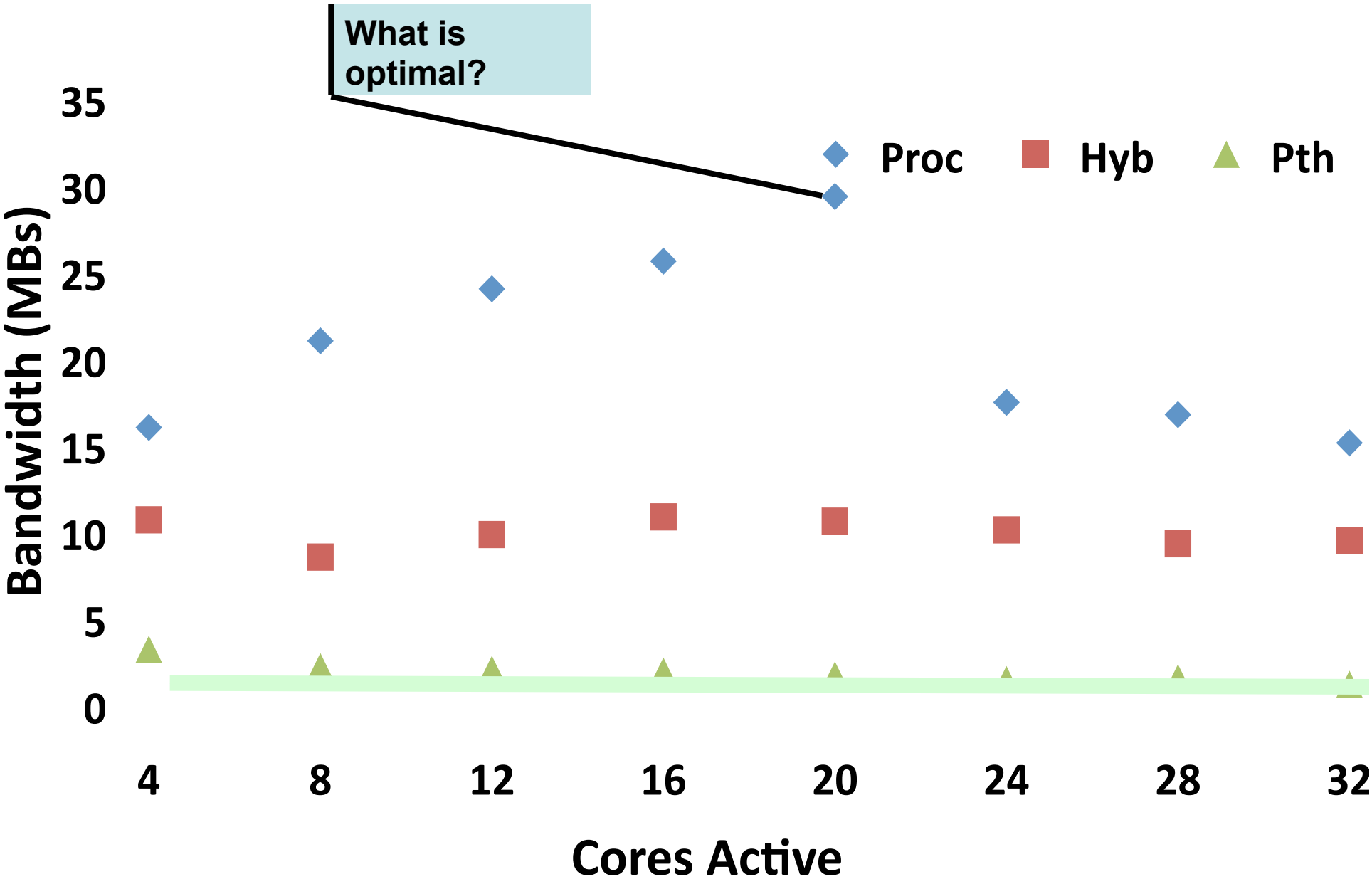
Throughput Oriented Runtime for Large Scale Manycore Systems (THOR)

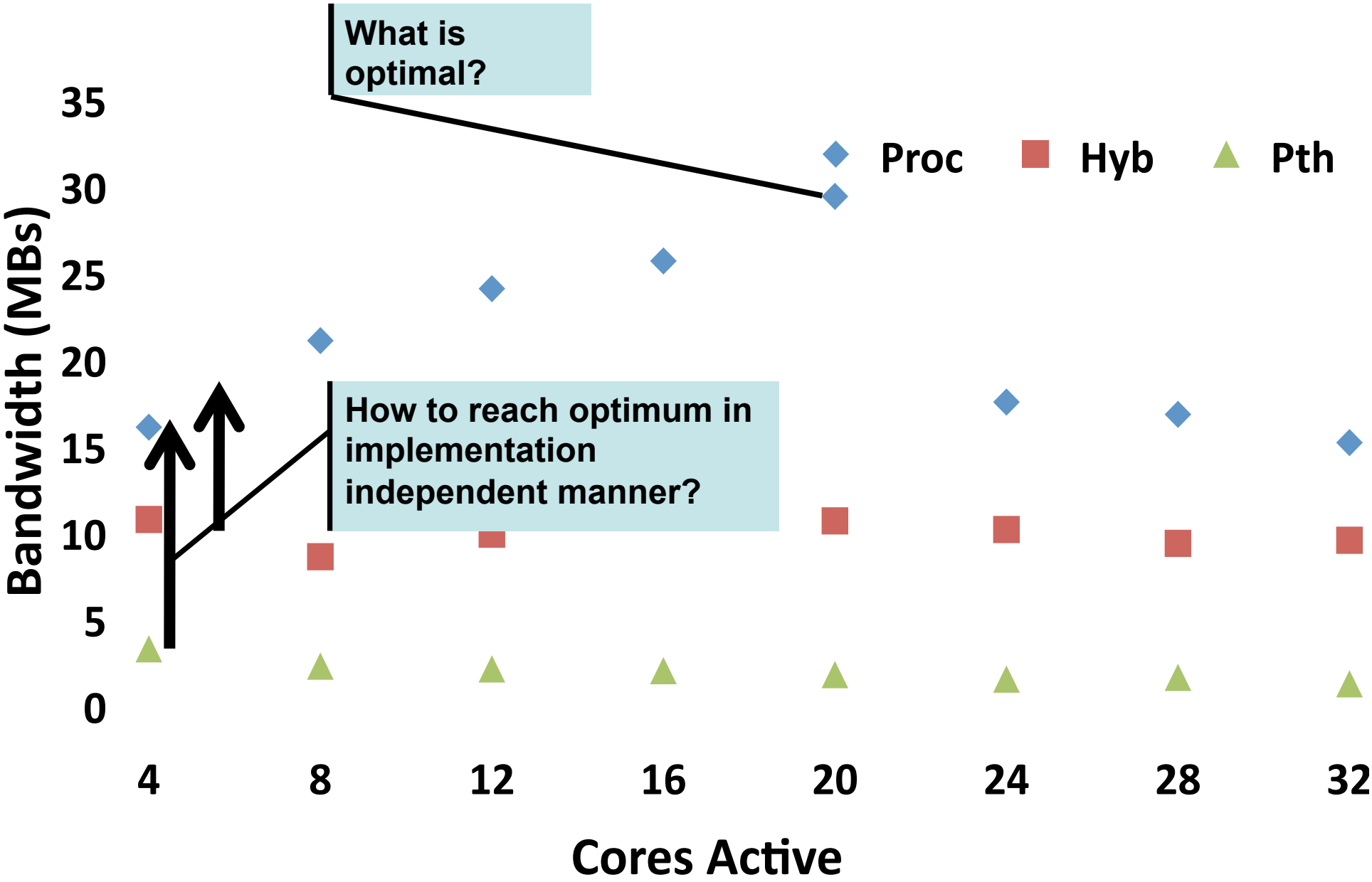
THOR

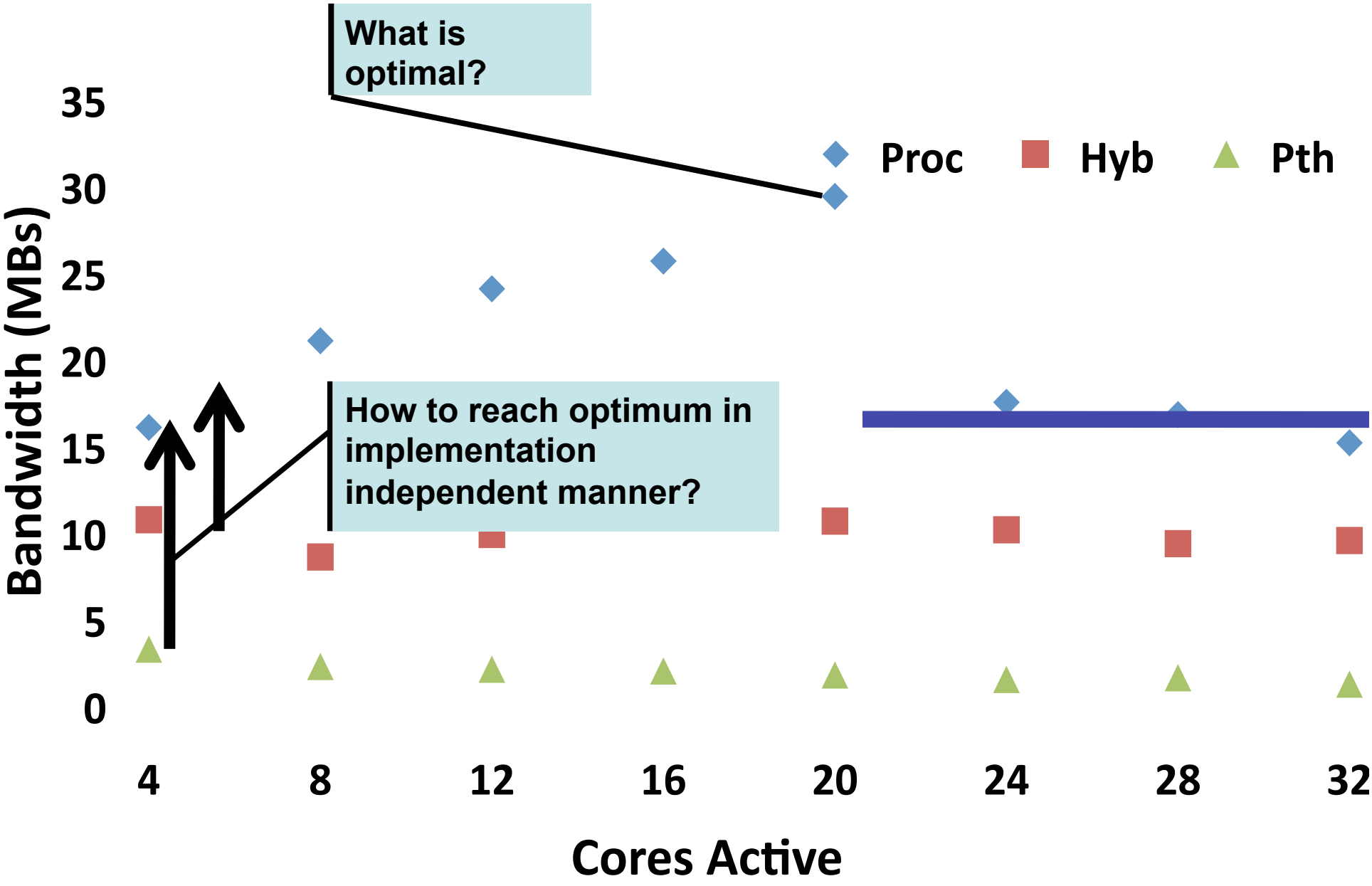

Bandwidth (MBs)
35
30
25
20
15
10
Pthreads
8
12
16
20
24
28
32
Cores Active

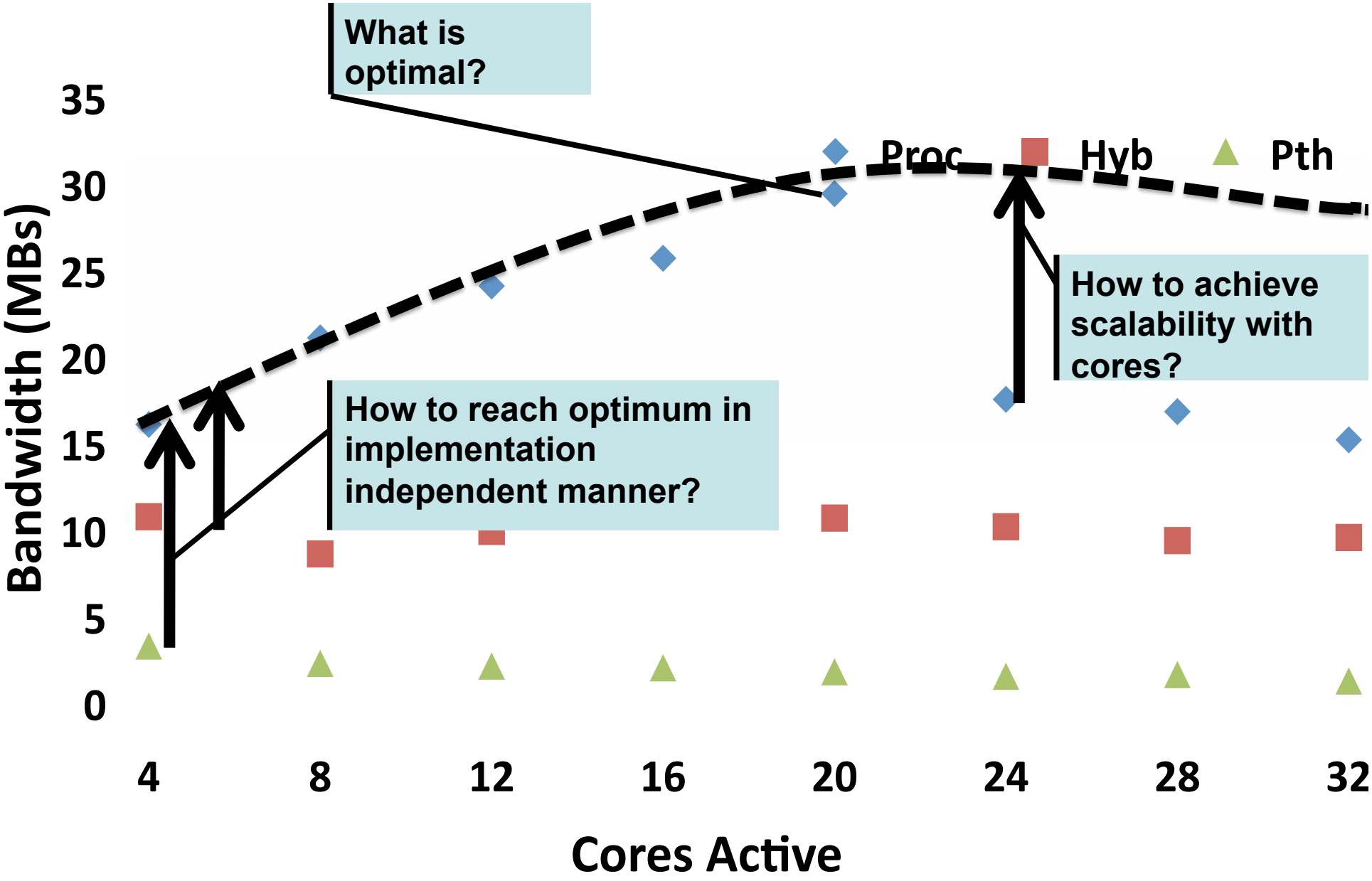

Bandwidth (MBs)
35
30
25
20
15
10
5
0
Proc
Hyb
Pth
4
8
12
16
20
24
28
32
Cores Active

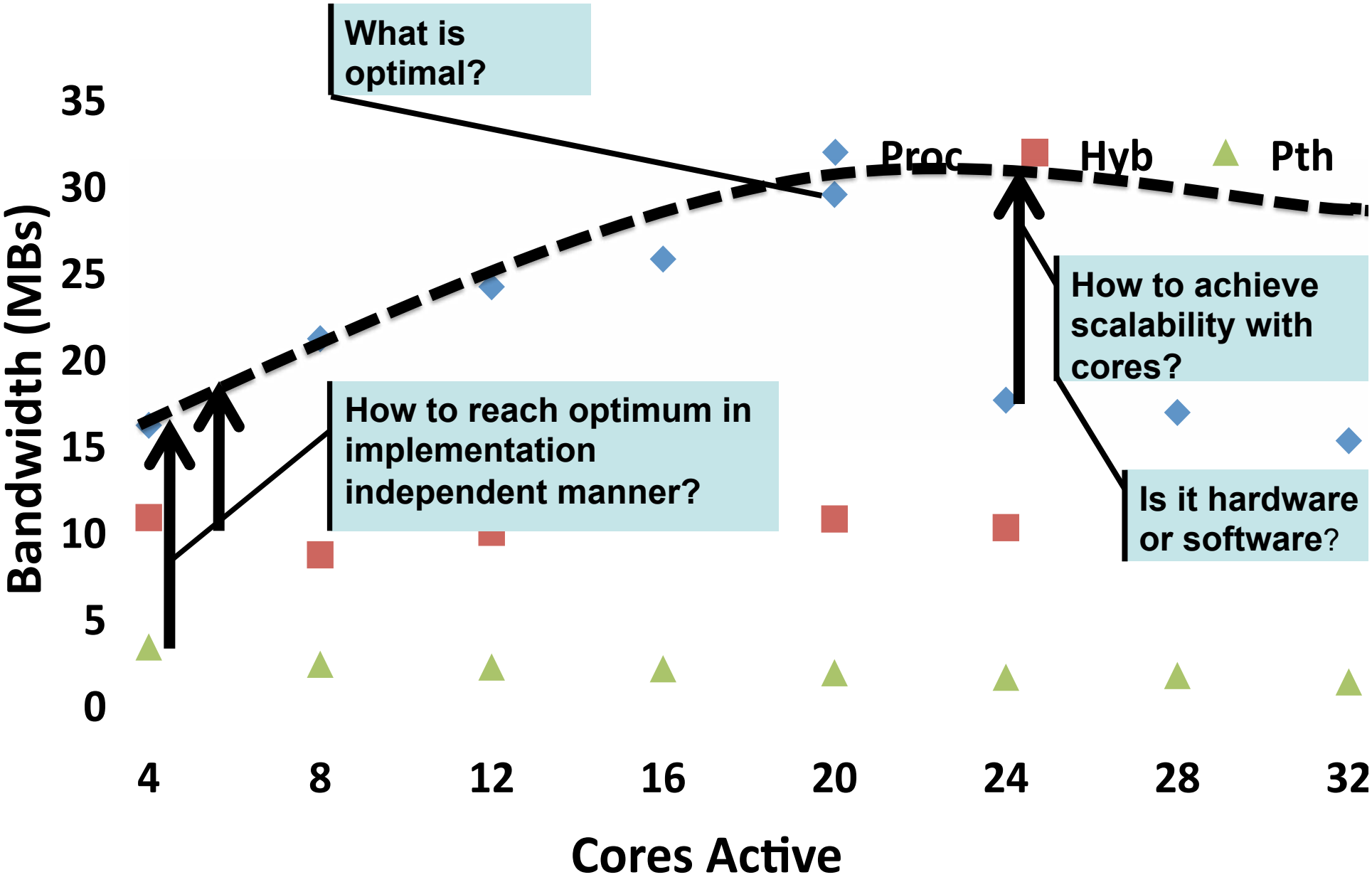

What is optimal?
Proc
Hyb
Pth
Bandwidth (MBs)
35
30
25
20
15
10
5
0
4
8
12
16
20
24
28
32
Cores Active


What is optimal?
Proc
Hyb
Pth
Bandwidth (MBs)
35
30
25
20
15
10
5
0
4
8
12
16
20
24
28
32
Cores Active


What is optimal?
How to reach optimum in implementation independent manner?
Proc
Hyb
Pth
Bandwidth (MBs)
35
30
25
20
15
10
5
0
4
8
12
16
20
24
28
32
Cores Active


What is optimal?
How to reach optimum in implementation independent manner?
Proc
Hyb
Pth
Bandwidth (MBs)
35
30
25
20
15
10
5
0
4
8
12
16
20
24
28
32
Cores Active

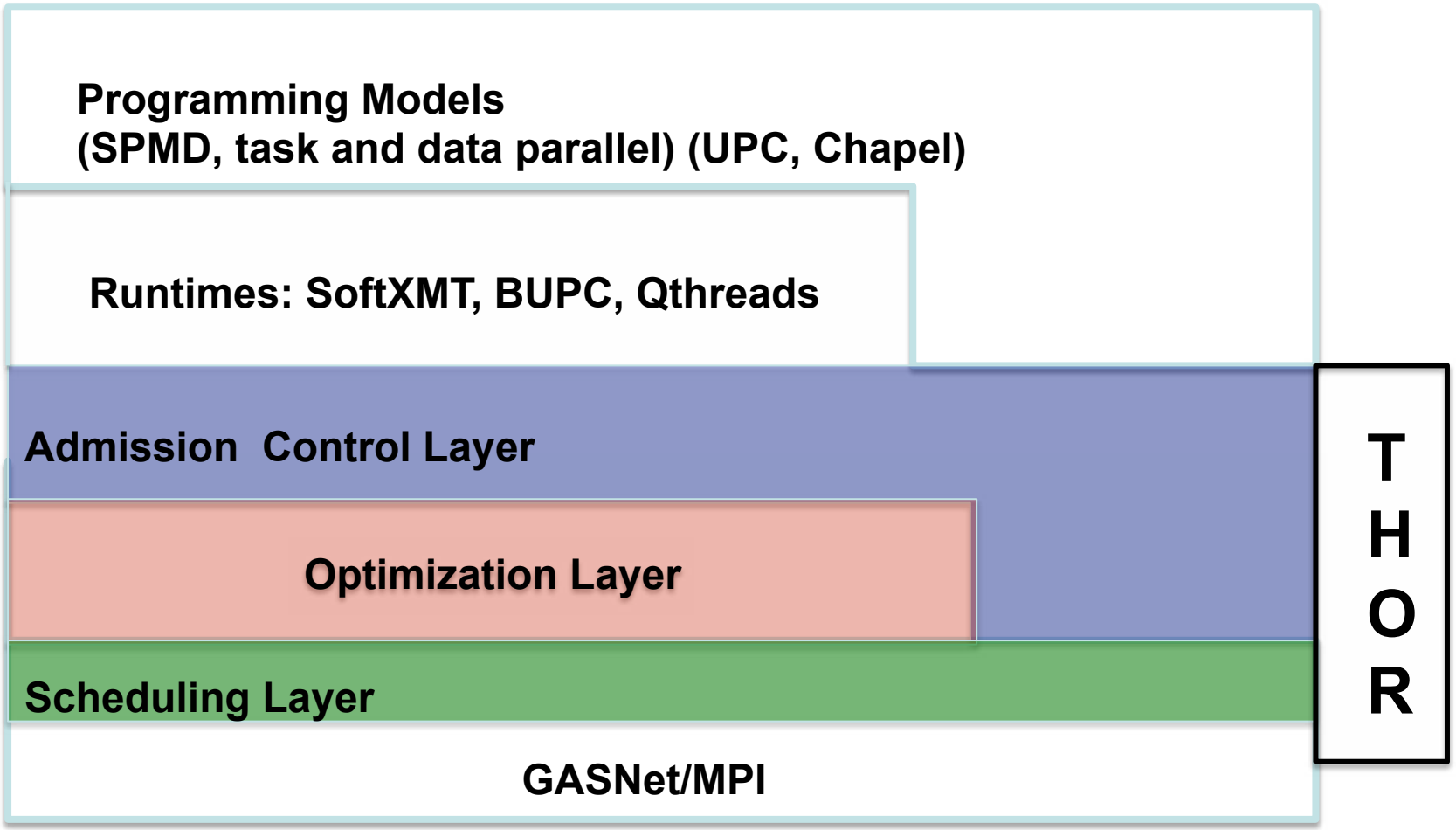

What is optimal?
How to reach optimum in implementation independent manner?
How to achieve scalability with cores?
Proc
Hyb
Pth
Bandwidth (MBs)
0
5
10
15
20
25
30
35
4
8
12
16
20
24
28
32
Cores Active


What is optimal?
How to reach optimum in implementation independent manner?
How to achieve scalability with cores?
Is it hardware or software?
Proc
Hyb
Pth
Bandwidth (MBs)
0
5
10
15
20
25
30
35
4
8
12
16
20
24
28
32
Cores Active

Goals

- **Efficiency and productivity layer for heavily threaded/asynchronous runtimes**
  - Decouple application/runtime level concurrency from runtime concurrency
  - Manage asynchrony for clients
  - Provide optimal throughput for
    - Any implementation (pthreads, procs …)
    - Any hardware architecture (asymmetric, heterogeneous)
    - Any message mix
    - Any source, target

THOR Architecture



Driven by runtime analysis and performance models

Thor Layers

- **Admission Control Layer**
  - Congestion Avoidance
  - Flow Control
  - Memory Consistency/Ordering
  - Dispatch to Optimization Services
- **Optimization Layer**
  - Coalescing
  - Aggregation
- **Scheduling Layer**
  - Reordering
  - Instantiate and Retire Communication to Network

Scalable/Portable Design

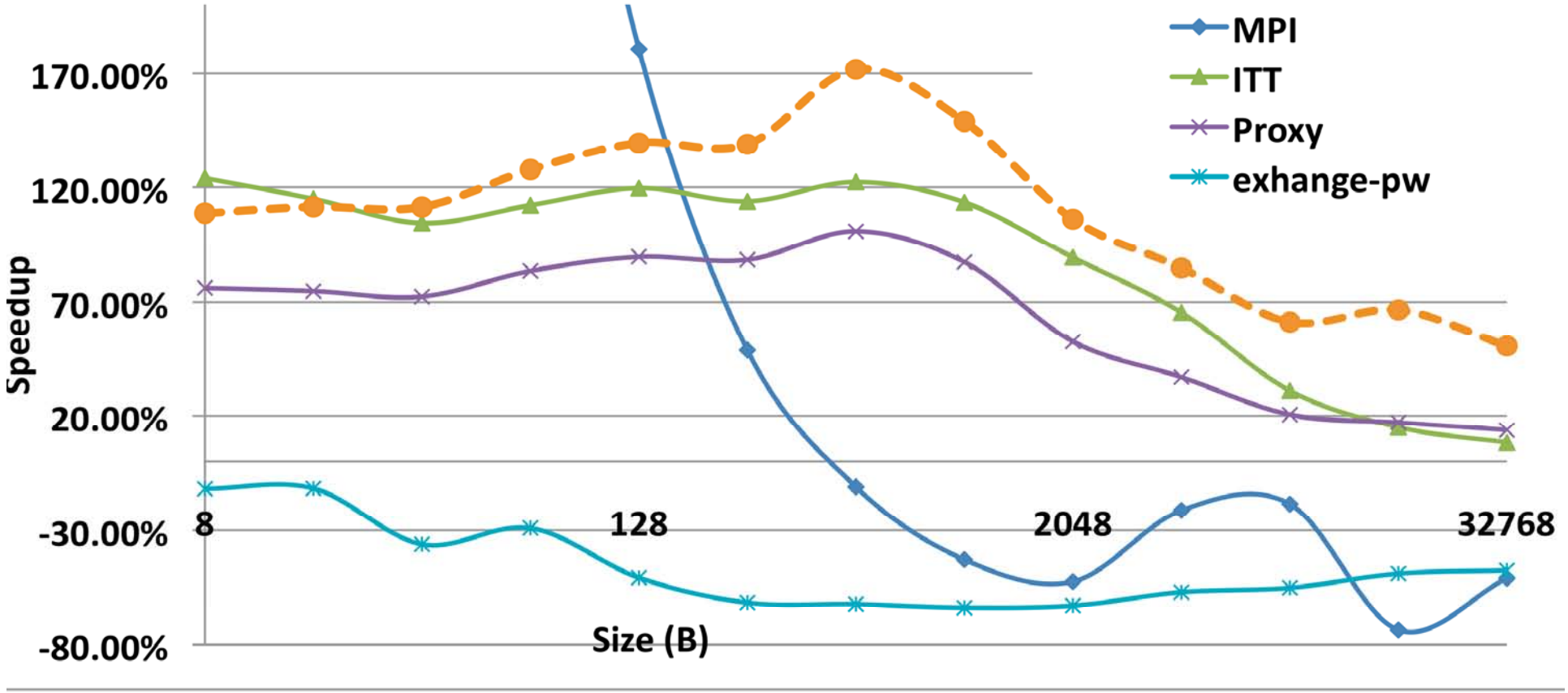
- **Multiple implementations for asymmetric/ heterogeneous hardware**
  - Inline: mechanisms implemented in a distributed manner, e.g. GASNet/MPI
  - Proxy: servers acting on behalf of clients
- **Open loop control**
  - With as little “global” state as possible
- **Declarative behavior**
  - Intuitive (human descriptions) – e.g. train of messages
  - Annotated by compilers/optimizers

Scalability with Cores

- **Congestion avoidance runtime prototyped**
  - BUPC/GASNet/InfiniBand
  - Cray UPC/DMAPP/Gemini
- **Admission Control + Scheduling Layer**
  - Not well tuned yet
- **Results:**
  - **4X** performance improvement for all-to-all
  - **70%** improvement on GUPS/HPCC RA
  - **17%** on NAS Parallel Benchmarks

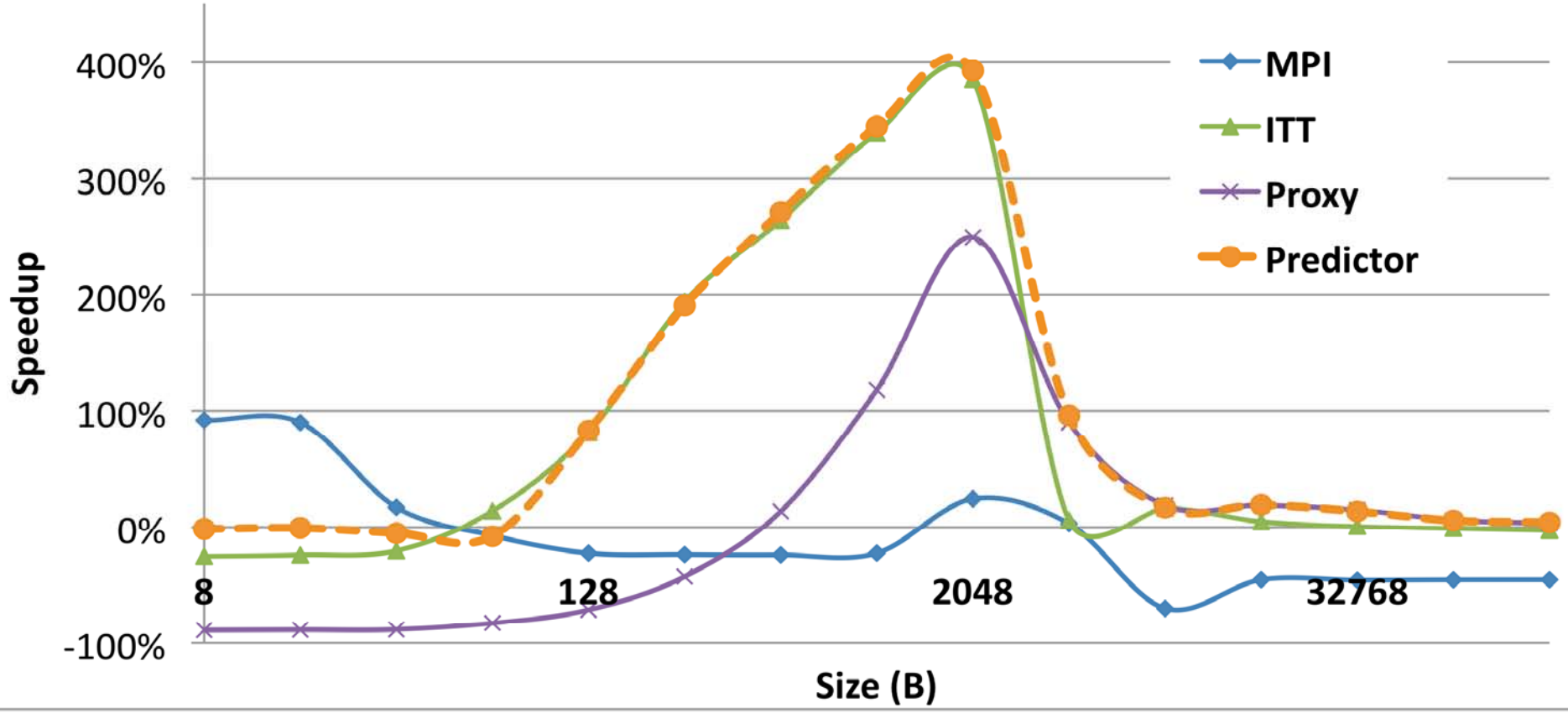
To appear as "Congestion Avoidance on Manycore Clusters" in ICS 2012

All-to-all InfiniBand 1024 Cores



Speedup over GASNet tuned all-to-all - 2x Performance Portable – single implementation

All-to-all Gemini 768 Cores



Speedup over Cray UPC all-to-all - 4x Performance Portable – single implementation

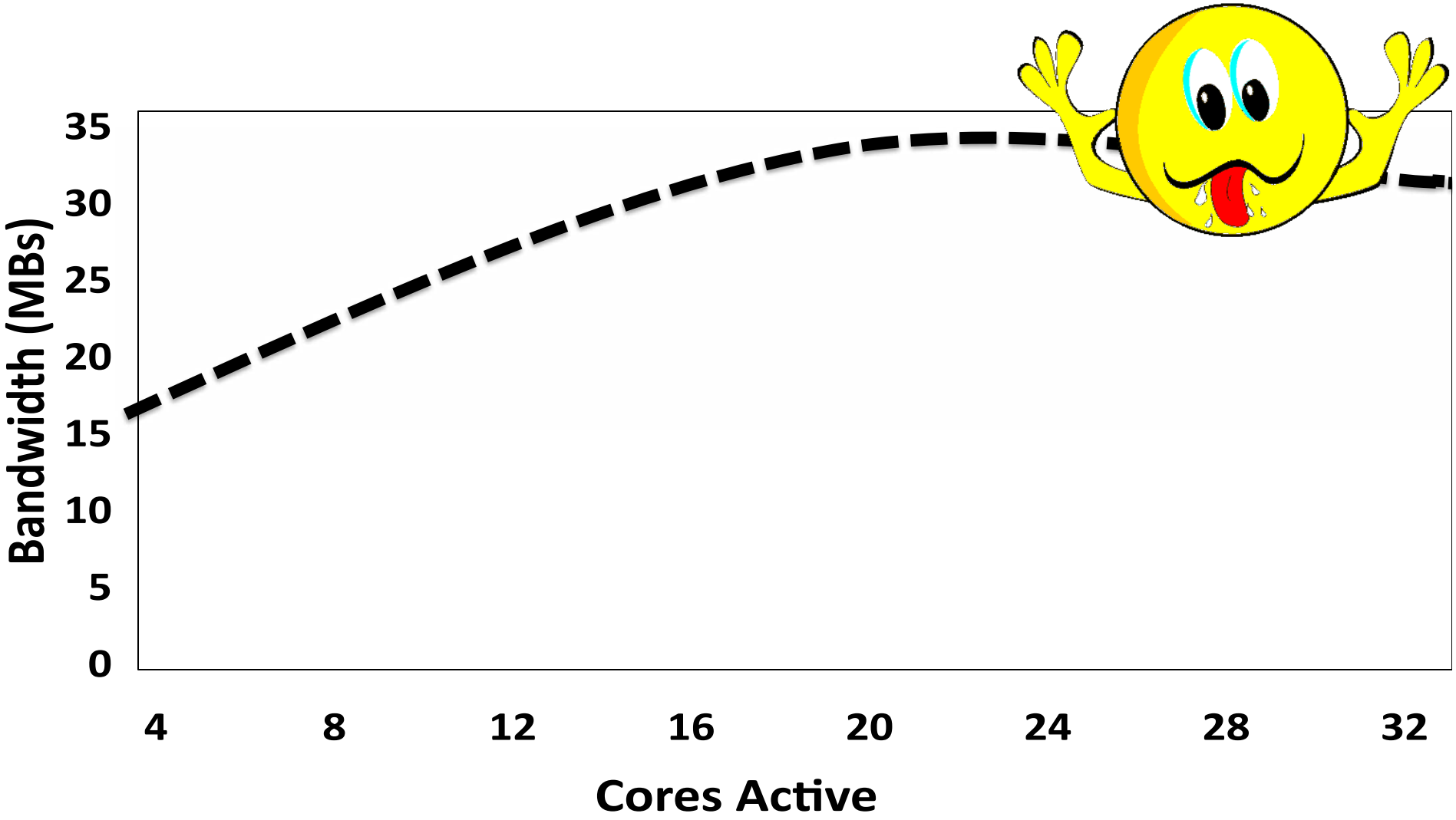
Future Work

Principles of HW/SW Design

- Why don't we get maximal throughput?
- How do we get maximal throughput?

Thank You!

THOR


Bandwidth (MBs)
35
30
25
20
15
10
5
0
4
8
12
16
20
24
28
32
Cores Active

THOR Timeline

- **Year 1:** Performance study and prototype implementation
- **Year 2:** Admission Control Layer + Scheduling Layer
- **Year 3-4:** Optimization Layer and tuning on target systems

- **Demonstration:**
  - Languages: UPC, Chapel
  - Runtimes: BUPC, SoftXMT (Qthreads?)
  - Networks: InfiniBand, Cray
  - CPUs: Intel/AMD, Knigths Corner/Landing (other suggestions)