CMP334

Deep-Dive Into 100G networking & Elastic Fabric Adapter on Amazon EC2

Brendan Bouffler

Principal Product Manager, HPC & Batch

Amazon Web Services

Chris Liu

Snr Product Manager, EC2

Amazon Web Services

AWS re:Invent

Agenda

Amazon EC2 networking overview

100G use cases

Nitro architecture overview

Elastic Fabric Adapter on EC2

HPC use cases

Networking on AWS

Nitro system

- Lightweight hypervisor
- Nitro card
- Nitro security chip

Benefits of Nitro - Elastic Network Adapter

- Over 20x increase in PPS performance
- Reduces instance-to-instance latencies
- Enables 100 Gbps of bandwidth performance

Amazon EC2 Instance bandwidth capabilities

T3 instances – up to 5 Gbps network performance

Smaller instance sizes (non T3) - up to 10 Gbps or up to 25 Gbps network performance

Larger instance sizes - sustained 10, 25, 50 or 100 Gbps

Model	vCPU	Mem (GiB)	Network Performance (Gbps)
t3.nano	2	6	Up to 5
t3.micro	2	12	Up to 5
t3.small	2	24	Up to 5
t3.medium	2	24	Up to 5
t3.large	2	36	Up to 5
t3.xlarge	4	96	Up to 5
t3.2xlarge	8	192	Up to 5

Model	vCPU	Mem (GiB)	Network Performance (Gbps)
c5.large	2	4	Up to 10
c5.xlarge	4	8	Up to 10
c5.2xlarge	8	16	Up to 10
c5.4xlarge	16	32	Up to 10
c5.9xlarge	36	72	10
c5.18xlarge	72	144	25

Model	vCPU	Mem (GiB)	Network Performance (Gbps)
c5n.large	2	4	Up to 25
c5n.xlarge	4	8	Up to 25
c5n.2xlarge	8	16	Up to 25
c5n.4xlarge	16	32	Up to 25
c5n.9xlarge	36	72	50
c5n.18xlarge	72	144	100

High-bandwidth compute instances: C5n

Massively scalable performance

- C5n instances will offer up to 100 Gbps of network bandwidth
- Significant improvements in maximum bandwidth, packet per seconds, and packets processing
- Custom designed Nitro network cards
- Purpose-built to run network bound workloads including distributed cluster and database workloads, HPC, real-time communications and video streaming

Featuring

Intel Xeon Scalable (Skylake) processor

High-bandwidth compute instances: M5n/R5n

Massively scalable performance

- M5n/R5n Instances offer up to 100 Gbps of network bandwidth
- Significant improvements in maximum bandwidth, packet per seconds, and packets processing
- Custom designed Nitro network cards
- Also available with instance storage

Featuring Intel Xeon Scalable (Cascade Lake) processor intel

High-bandwidth GPU instances: P3dn

Optimized for distributed ML training

- One of the most powerful GPU instance available in the cloud
- Distributed machine learning training across multiple GPU instances
- 100 Gbps of networking throughput
- Based on NVIDIA's latest GPU Tesla V100 with 32GB of memory each
- The largest Amazon Elastic Compute Cloud (Amazon EC2) P3 instance size available

I/O Intensive compute instances: i3en

Dense SSD storage for data-intensive workloads

- One of the most I/O intensive instances available in the cloud
- 100 Gbps of networking throughput
- 96 vCPUs of Intel® Xeon® Scalable (Skylake) processors @ 3.1GHz
- 60 TB of total NVMe instance storage
- 768 GiB of memory

Featuring Intel Xeon Scalable (Skylake) processor

NVIDIA T4 Tensor Core GPUs: G4dn

Machine learning, video transcoding, game streaming, and remote graphics workstations applications

- up to four NVIDIA T4 Tensor Core GPUs
- 100 Gbps of networking throughput
- 64 vCPUs of Intel® Xeon® Scalable (Cascade Lake) processors
- Up to 1.8 TB of NVMe fast local storage
- 256 GiB of memory

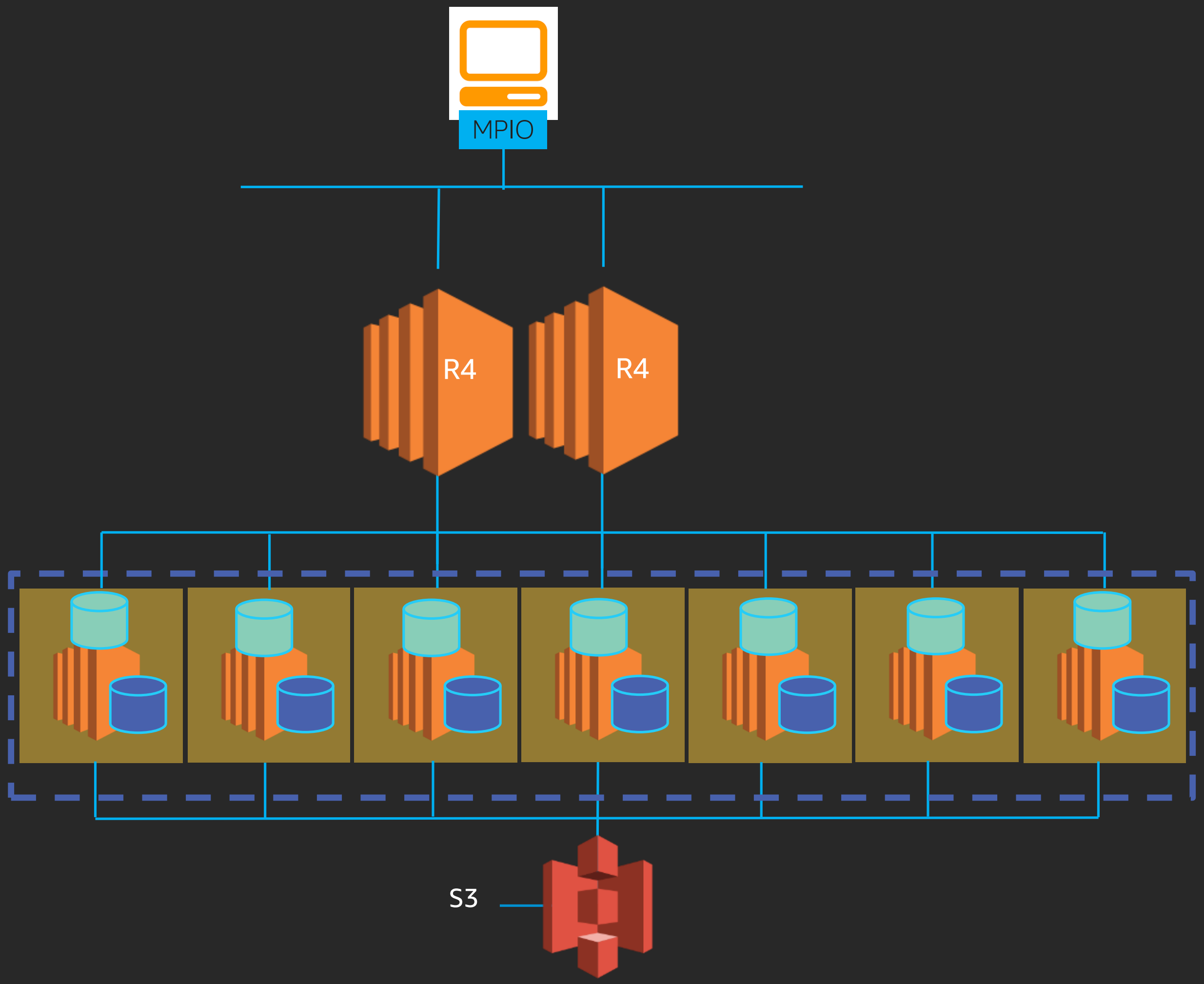
NEW!

Machine learning inference instances: INF1

Machine learning inference applications, interactive speech, computer vision, and predictive data analytics

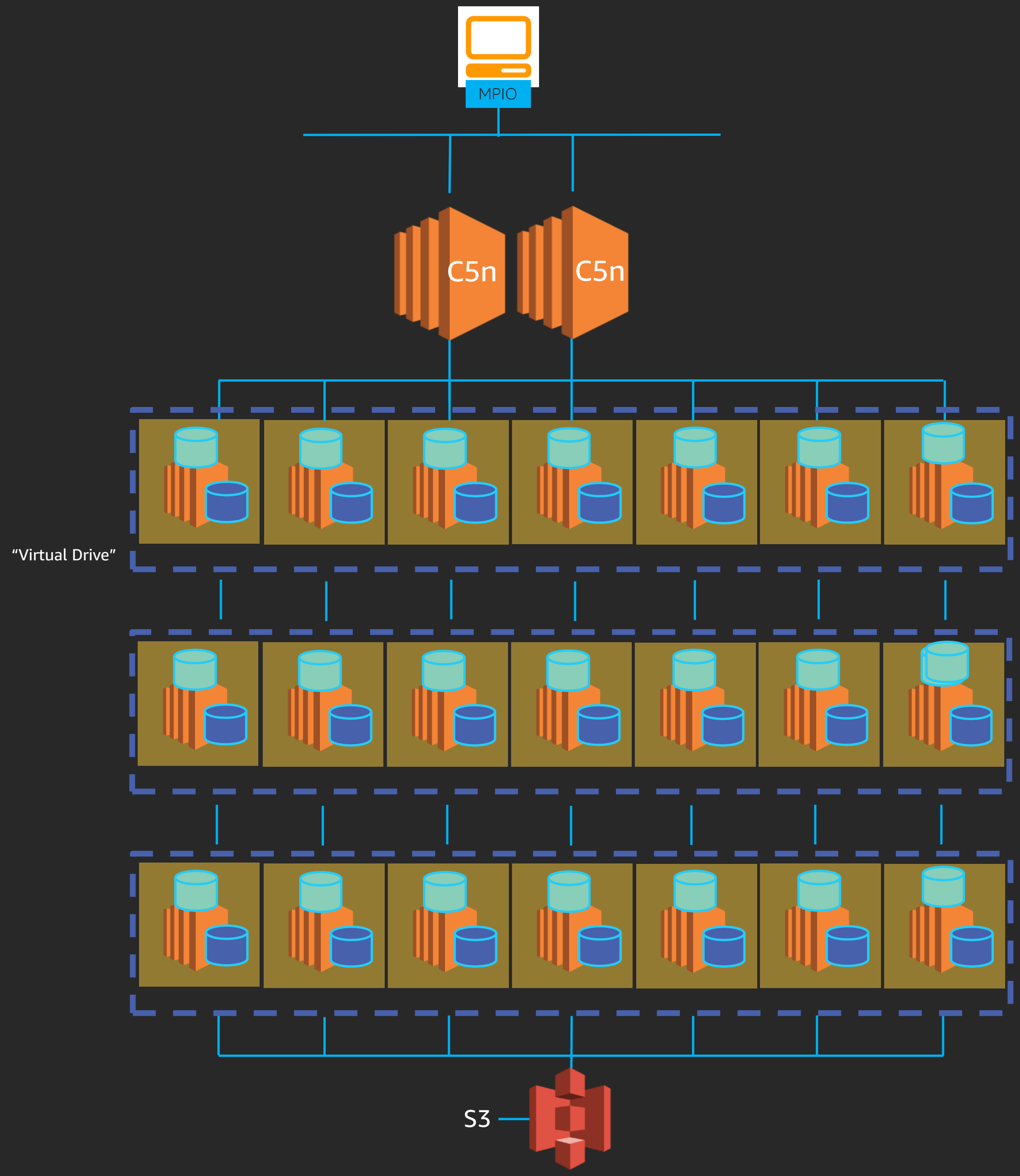
- up to four NVIDIA P4 Tensor Core GPUs
- 100 Gbps of networking throughput
- 64 vCPUs of Intel® Xeon® Scalable (Cascade Lake) processors
- Up to 1.8 TB of NVMe fast local storage
- 256 GiB of memory

Cloud Block Storage on C5n



Controllers needed a lot of bandwidth to send IO between the controllers and virtual drives to scale out workloads

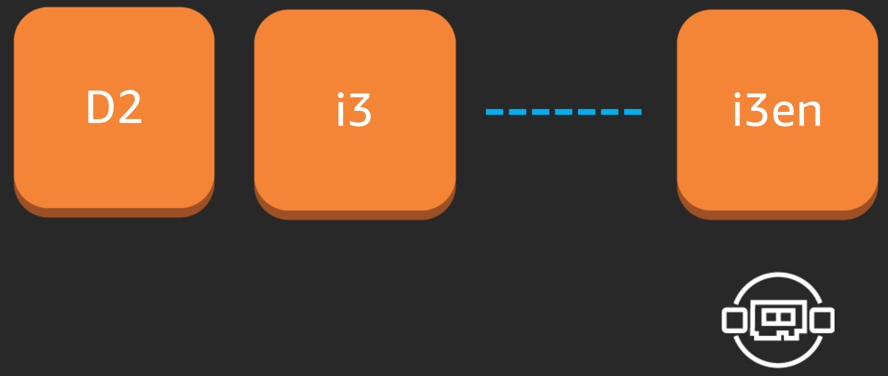
Cloud Block Storage on C5n



Pure Storage migrated from R4 instances to C5n to leverage increased bandwidth

The higher BW with C5n allows CBS to scale from 7 x i3 virtual drives to 21 VDs, and even further if needed

Datadog on I3en



At Datadog, we have a history of data intensive workloads. The D2 class served our initial requirements for high capacity, direct-attached storage. When the I3 instance emerged, the storage and networking revealed an incredible level of throughput that revolutionized the way we design our streaming workloads.

The I3en update takes this a step further with increased network and storage density that better fit our capacity model, reducing costs and minimizing MTTR.

We migrated a multi-petabyte workload to i3en without hesitation, immediately reaping huge gains in efficiency without requiring changes to software design or architecture.

Instance bandwidth limits

Between instances:

Up to 100 Gbps within the VPC or peered VPC

To Amazon S3:

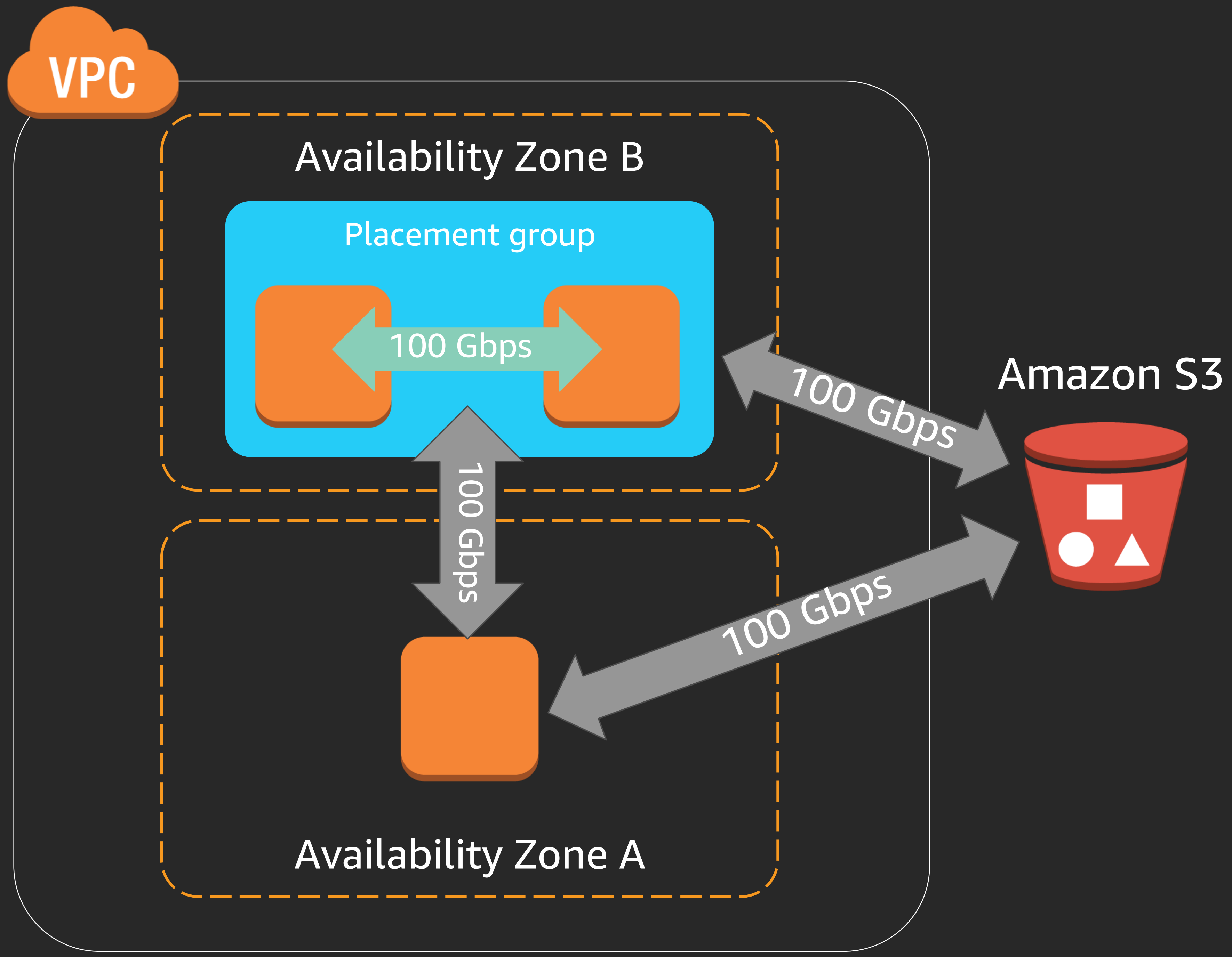
Up to 100 Gbps to VPC Endpoints and public IPs in the same Region

Placement group:

Placement Group – 10 Gbps flow limit

Everything else:

5 Gbps aggregate per instance with 5 Gbps flows



Nitro Overview:

AWS Nitro

Launched in November 2017

In development since 2013

All new launches use Nitro

Purpose-built hardware/software

Hypervisor built for AWS

Nitro in three parts

Nitro Cards

Amazon VPC Networking

Amazon Elastic Block Store (Amazon EBS)

Instance Storage

System Controller

Nitro Security Chip

Integrated into motherboard

Protects hardware resources

Hardware Root of Trust

Nitro Hypervisor

Lightweight hypervisor

Memory and CPU allocation

Bare-metal-like performance

Nitro cards

Amazon ENA PCIe Controller

Amazon VPC Data Plane

Amazon VPC

NVMe PCIe Controller

Amazon EBS Data Plane

Amazon Elastic Block Store

NVMe PCIe Controller

Transparent Encryption

Instance Storage

System Control

Root of Trust

Nitro Control

Nitro card for Amazon VPC

Amazon VPC

Amazon ENA Controller

- Drivers available for all major operating systems
- Independent of fabric

Amazon VPC Data Plane

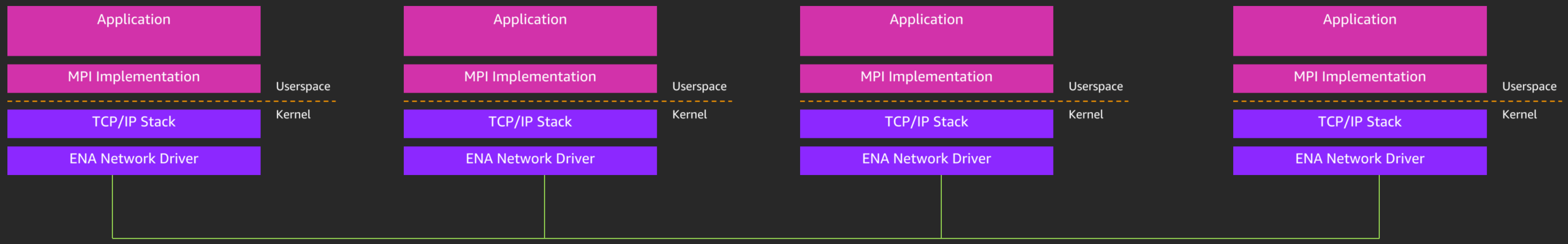
- Encapsulation
- Security Groups
- Limiters
- Routing

Turning this into an HPC tool

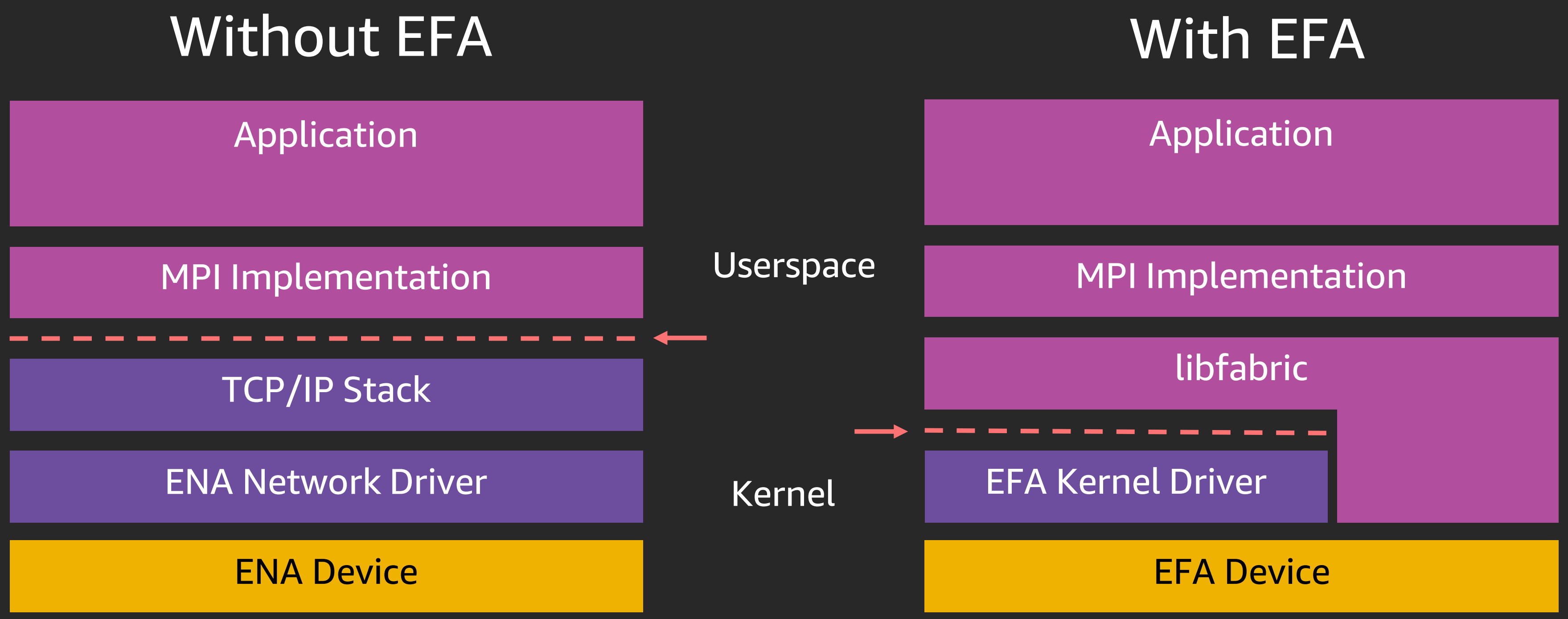
What do HPC users need?

- High performance
- Reliability
- Low latency/overhead
- Consistency and a jitter-free network
- Fairness

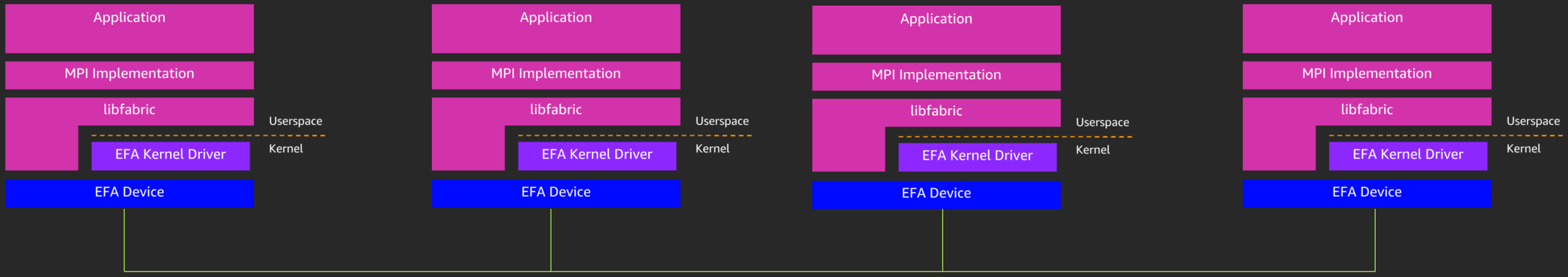
HPC software stack in Amazon EC2 (before Nitro & EFA)



HPC software stack with EFA



HPC software stack with EFA



Scalable Reliable Datagram (SRD)

A reliable high-performance lower-latency network transport

Inspired by Infiniband Reliable Datagram, without the drawbacks

- No limit on the number of outstanding messages per context

Out-of-order delivery – no head-of-line blocking

- Messages are independent: in many cases, application/middleware can restore ordering only if/when needed
- Same motivation as weak/relaxed memory ordering

Packet spraying over multiple ECMP paths

- Rapidly adapt to hot spots
- Fast and transparent recovery from network failures

Congestion control designed for large-scale cloud

- Maintains high throughput in the face of packet drops
- Minimize latency jitter

New network transport protocol

Designed for AWS's unique datacenter network

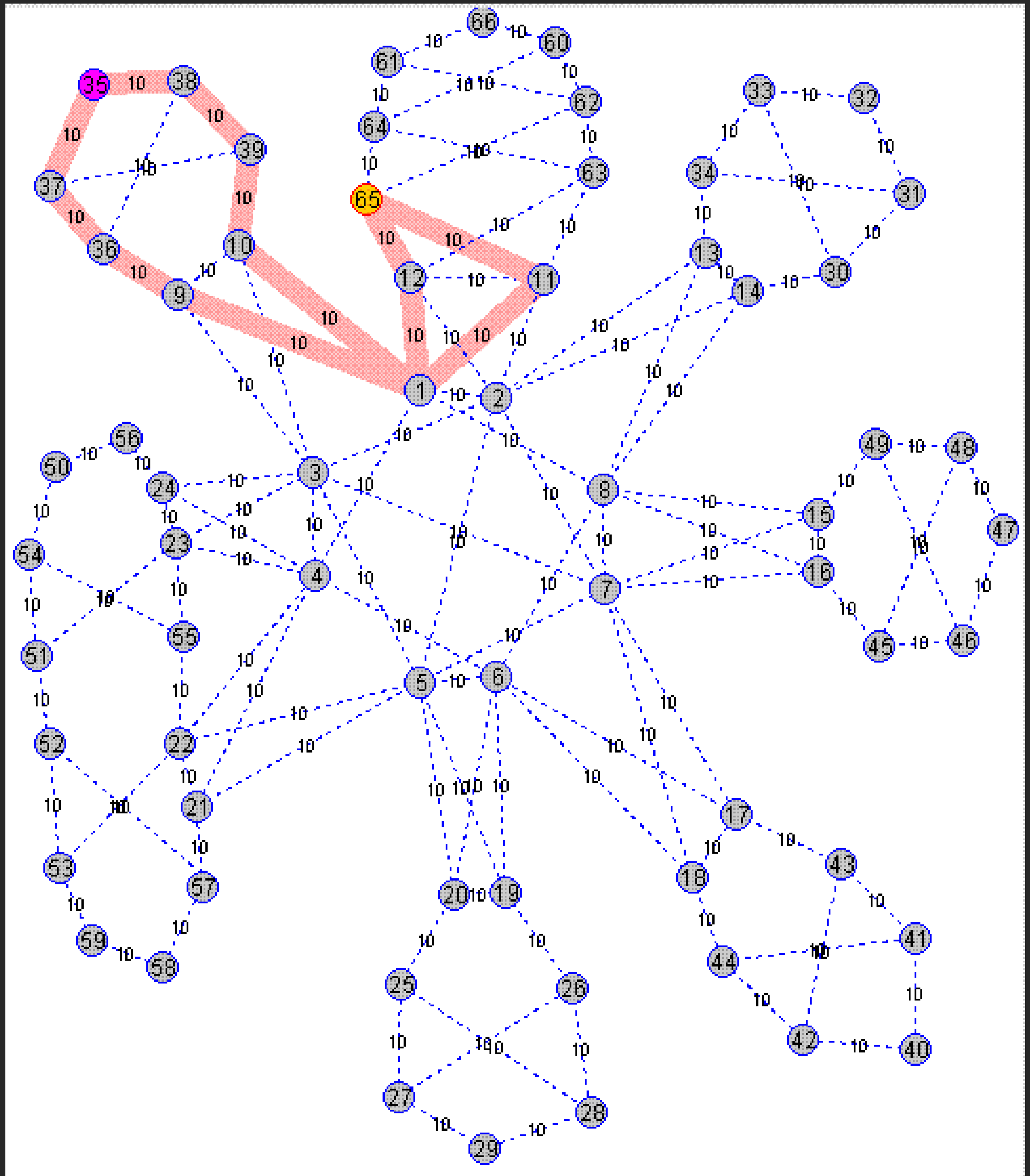
Part of AWS's 3rd generation Nitro Chip

Exposed as reliable datagram interface in EFA

TCP vs Infiniband vs SRD

TCP	Infiniband	SRD
Stream	Messages	Messages
In-order	In-order	Out-of-order
Single path	Single (ish) path	ECMP spraying with load balancing
High limit on retransmit timeout (>50ms)	Static user-configured timeout (log scale)	Dynamically estimated timeout (µs resolution)
Loss-based congestion control	Semi-static rate limiting (limited set of supported rates)	Dynamic rate limiting
Inefficient software stack	Transport offload with scaling limitations	Scalable transport offload (same number of QPs regardless cluster size)

Multipath routing



Equal-cost multi-path routing (**ECMP**) is a routing strategy where next-hop packet forwarding to a single destination can occur over **multiple "best paths"**. This can substantially increase bandwidth by load-balancing traffic over multiple paths.

Thanks to Wikipedia and it's contributors for the pithy explanation and Peter Ashwood-Smith for the snappy animated GIF explaining the concept.

What is Elastic Fabric Adapter (EFA)

Scale tightly-coupled HPC applications on AWS

P3dn

G4dn

C5n

i3en

M5n/
M5dn

R5n/
R5dn

C5adn

NVIDIA

NVIDIA
V100 Tensor
Core GPUs

intel

Custom
Intel® Xeon®
Scalable
processor

AMD

Custom
AMD Rome
(coming
soon)

EFA

Elastic Fabric Adapter, best for large HPC workloads

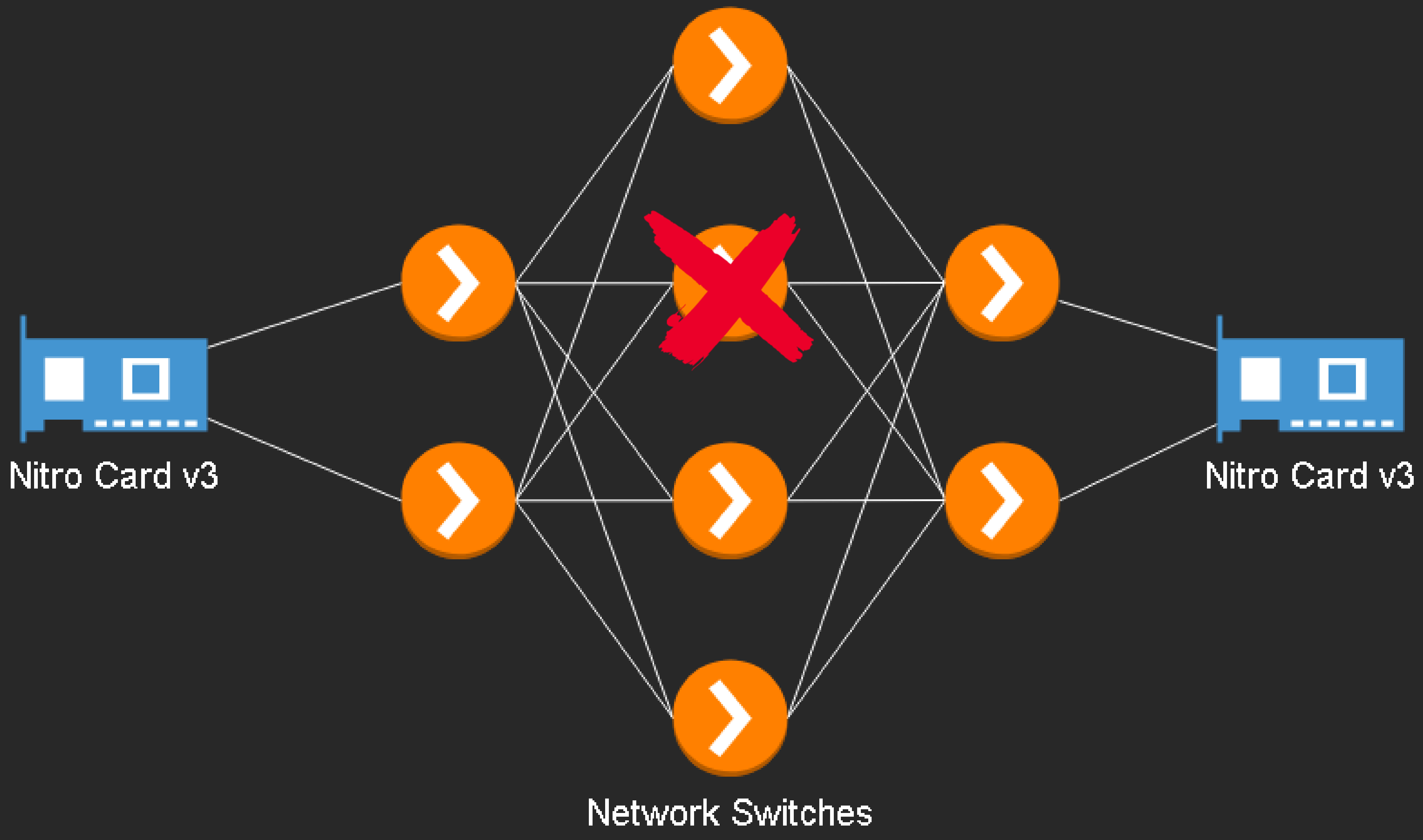
High data throughput

100 Gbps network bandwidth

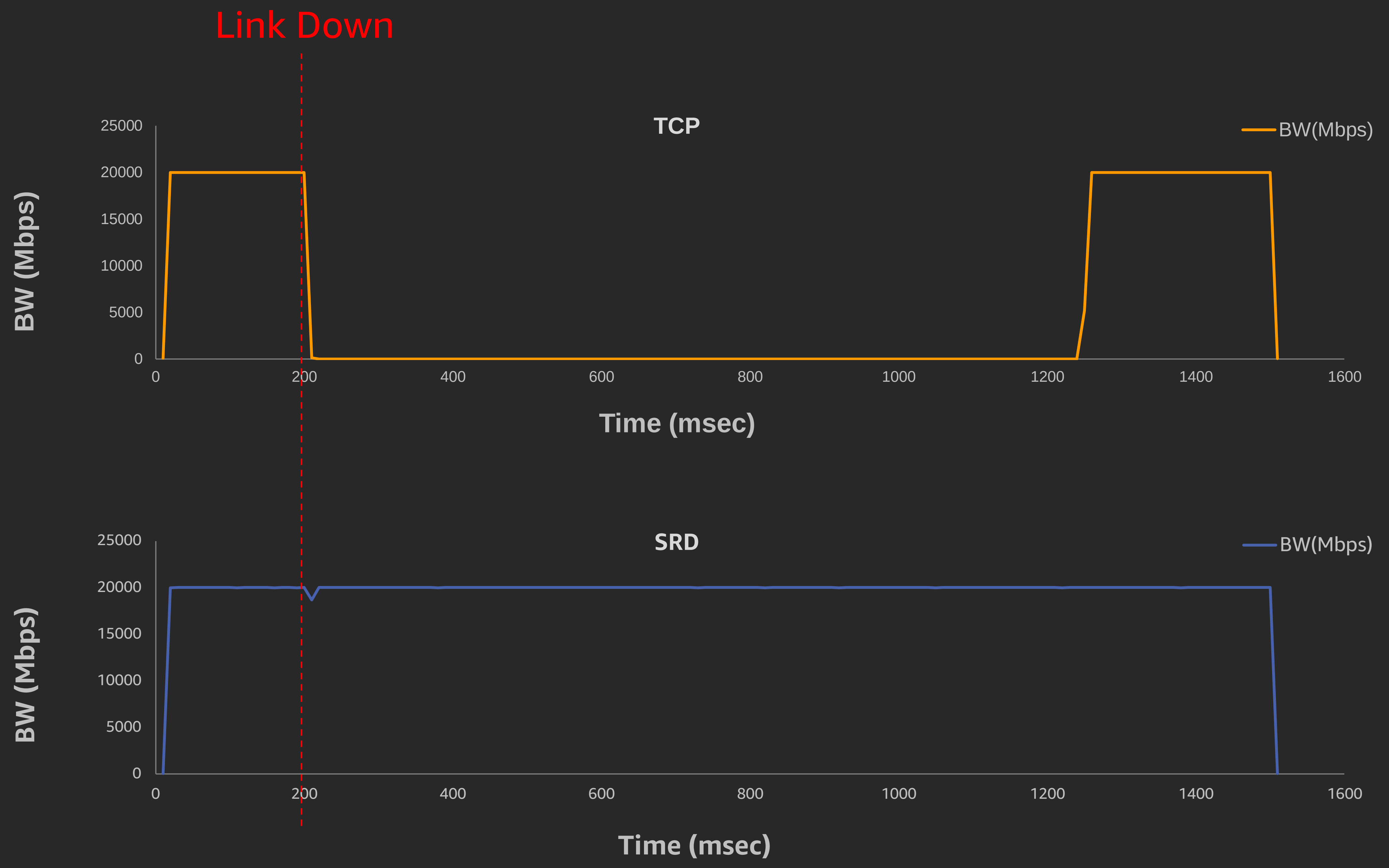
Congestion control for cloud scale and rapid packet loss recovery

Lower latency for message-passing and more effective application-layer comms

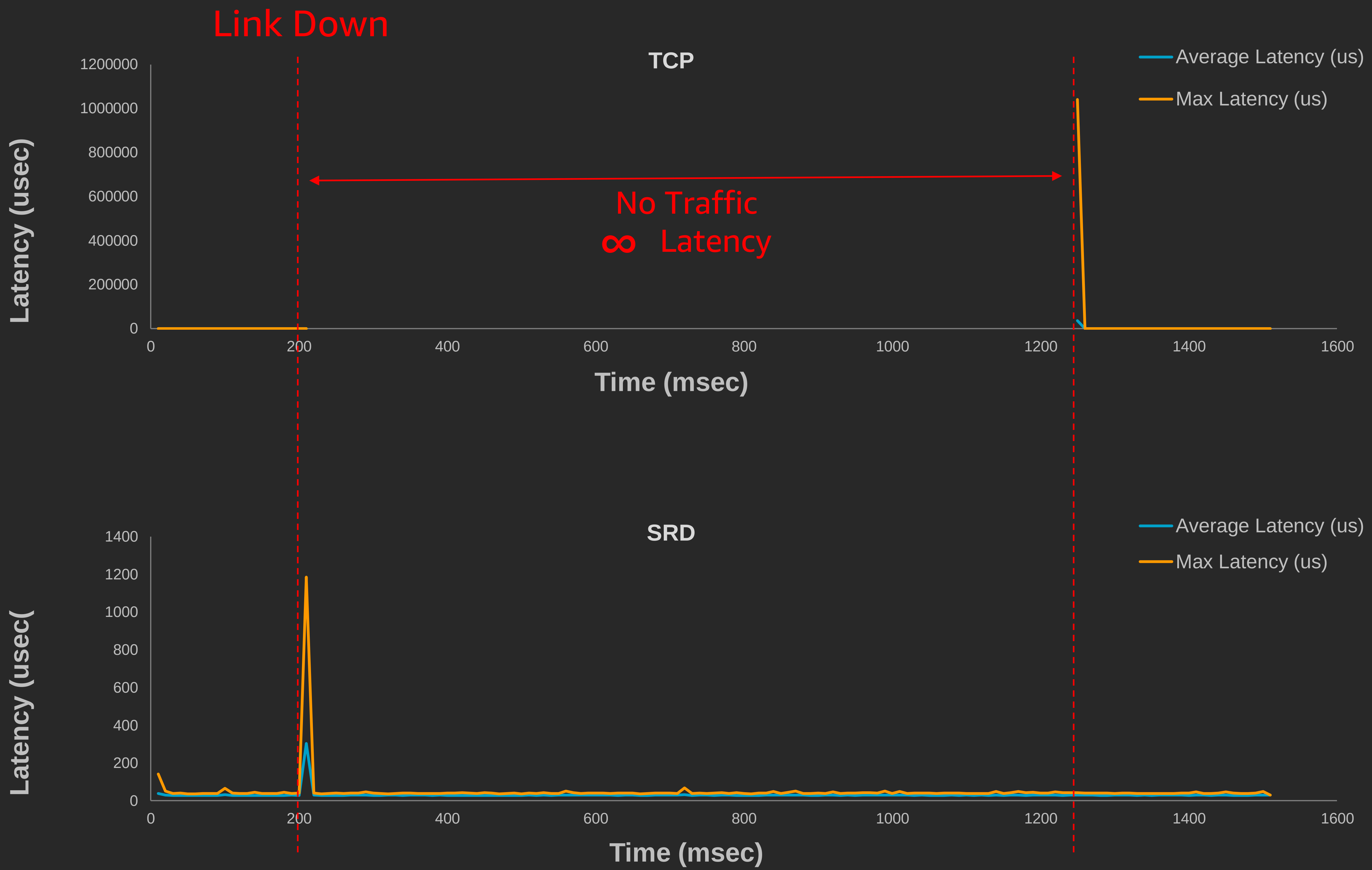
Link failure



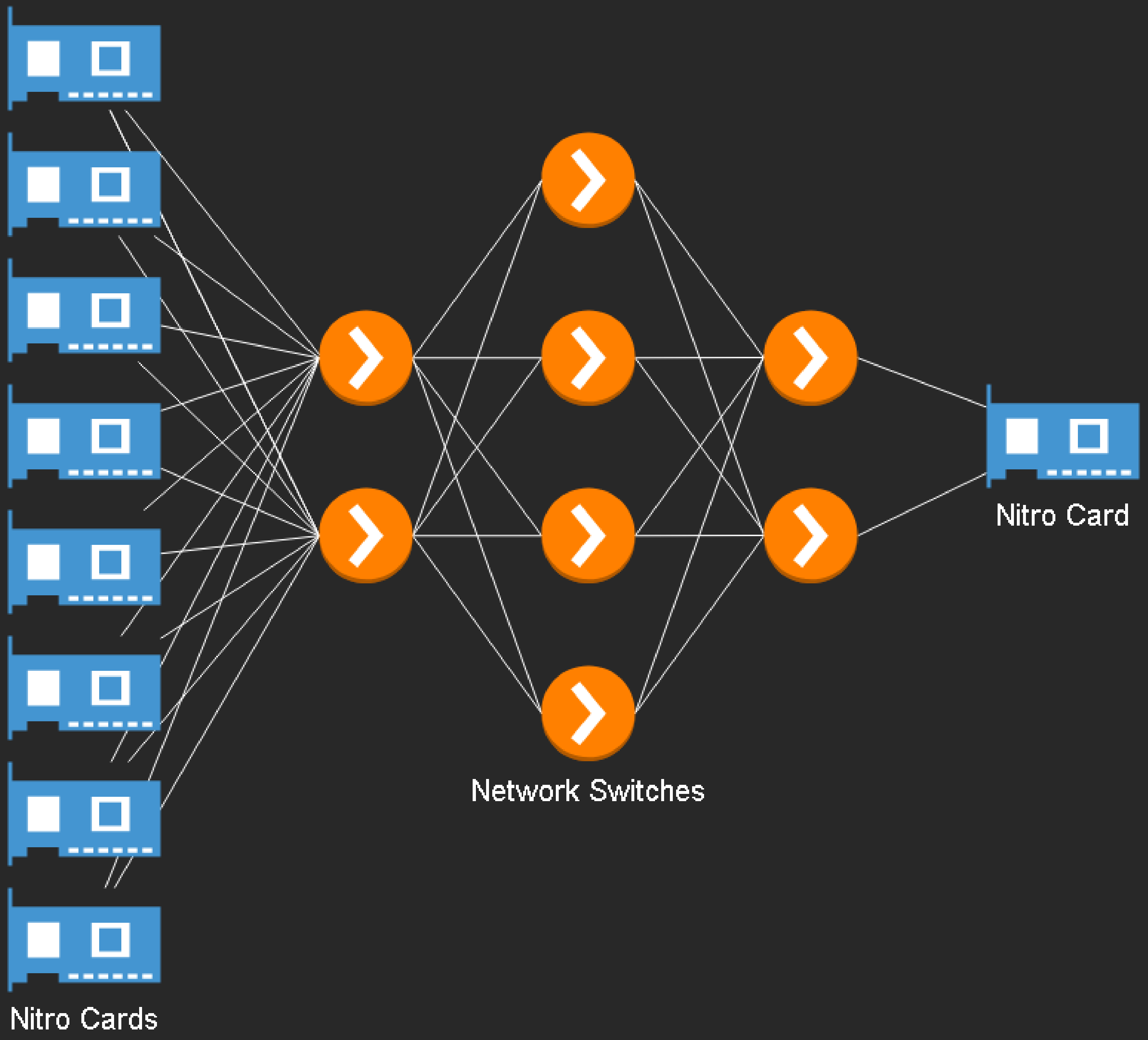
Link failure handling: Throughput



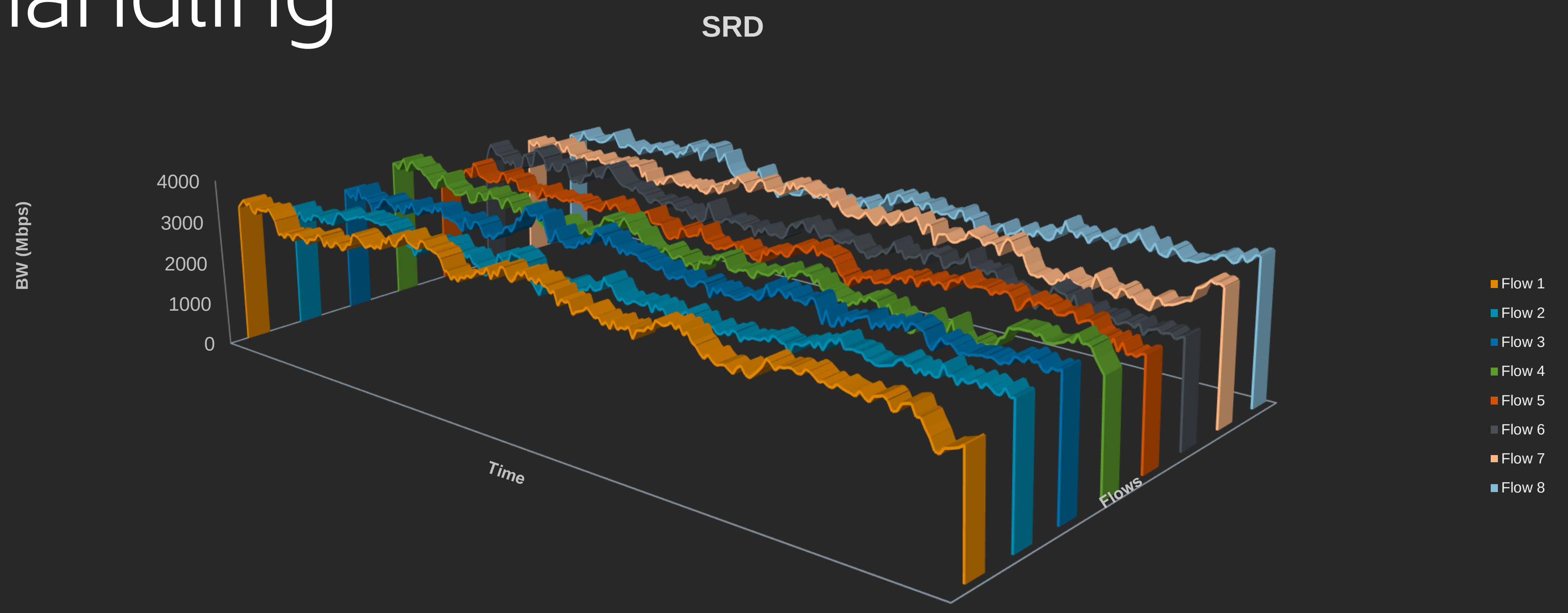
Link failure handling: Latency

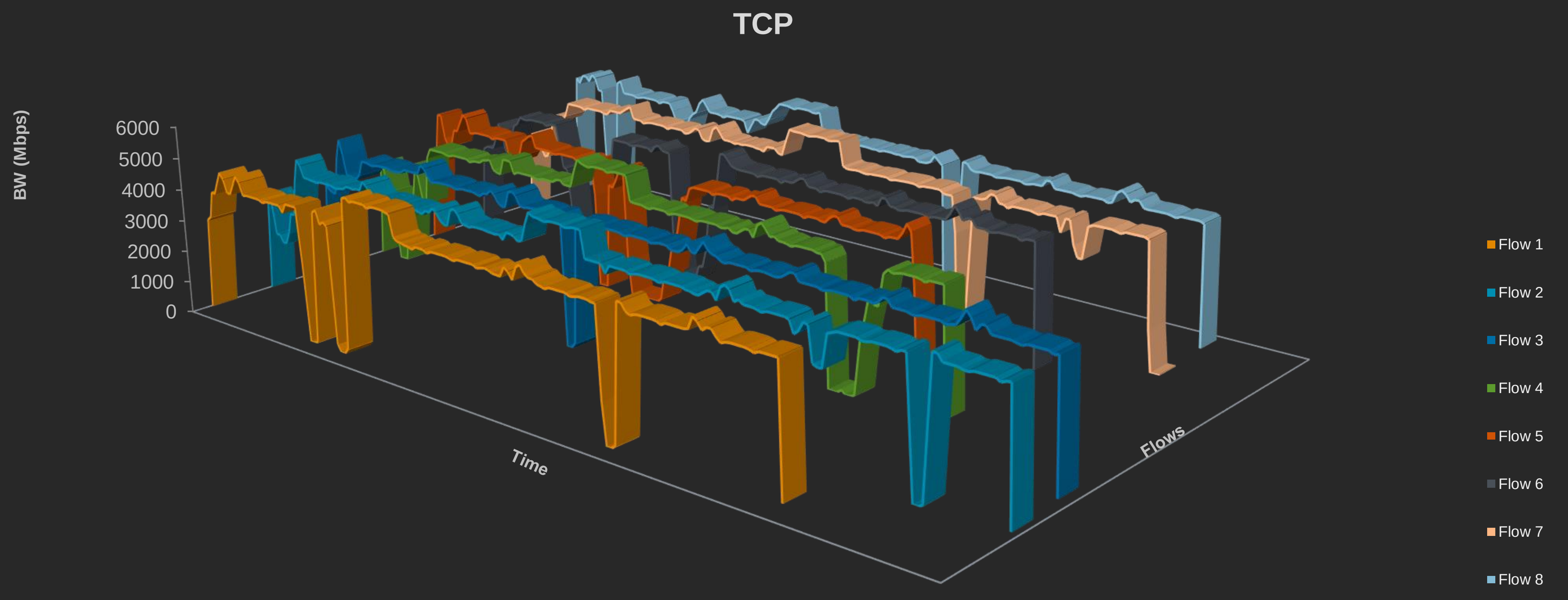


Incast

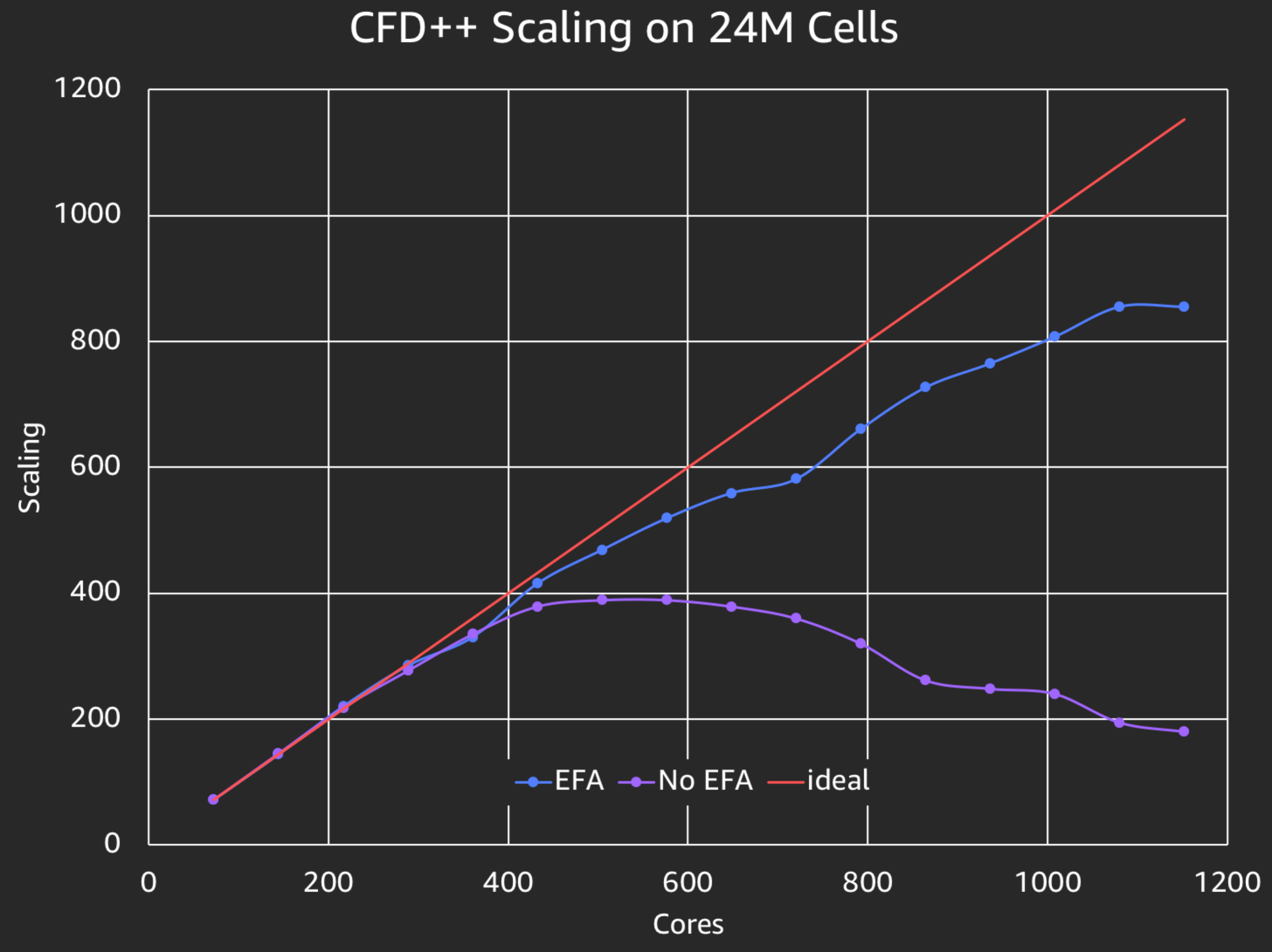


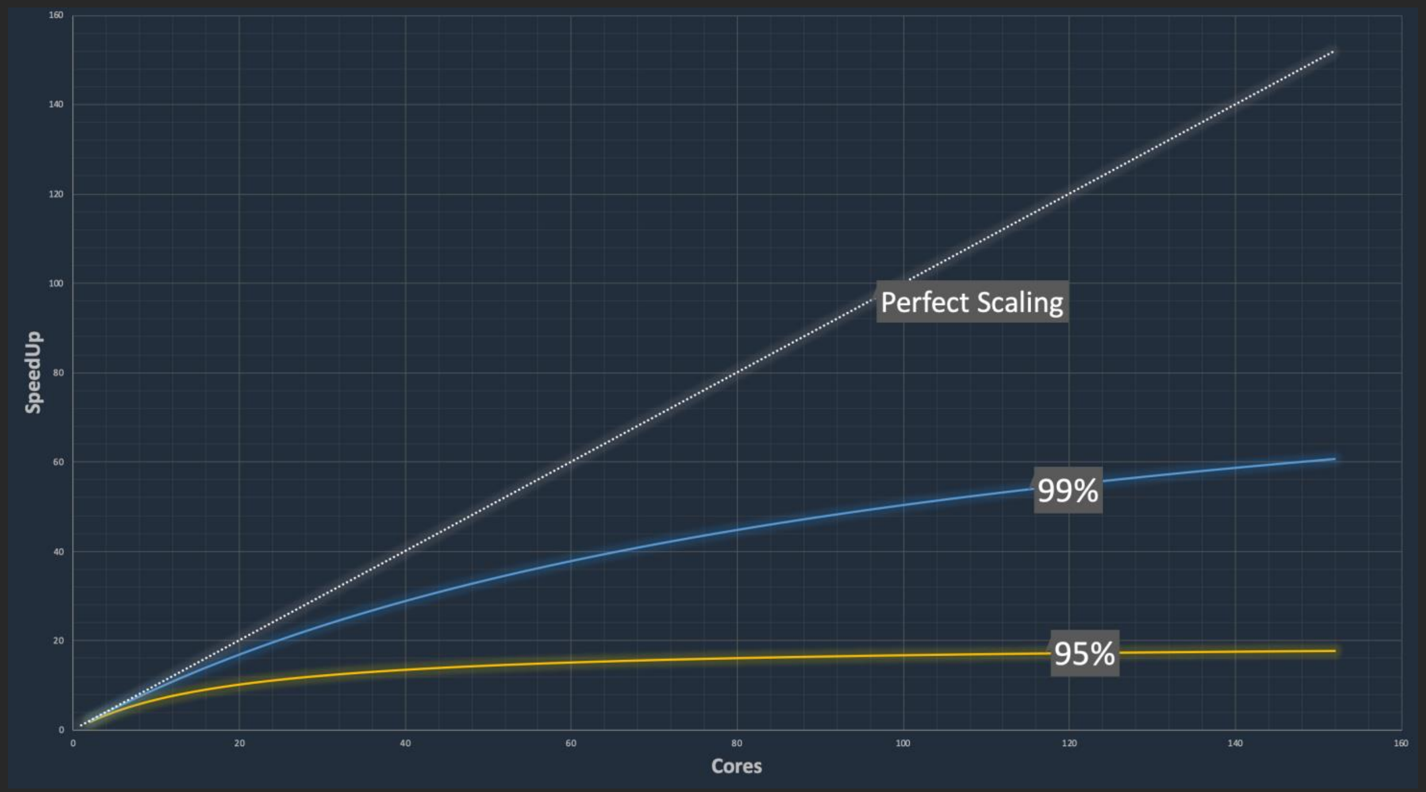
Incast handling





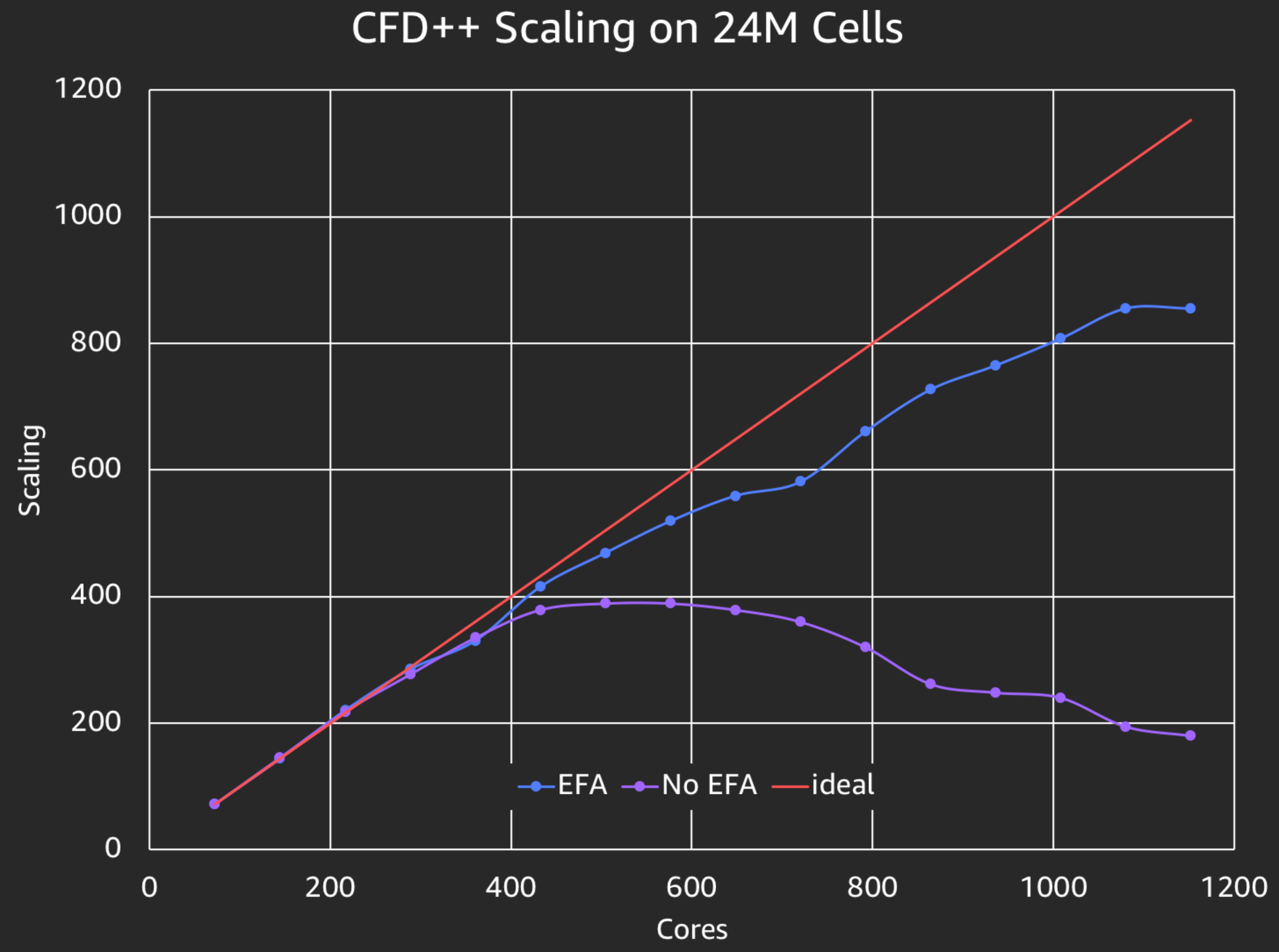
What can EFA do?

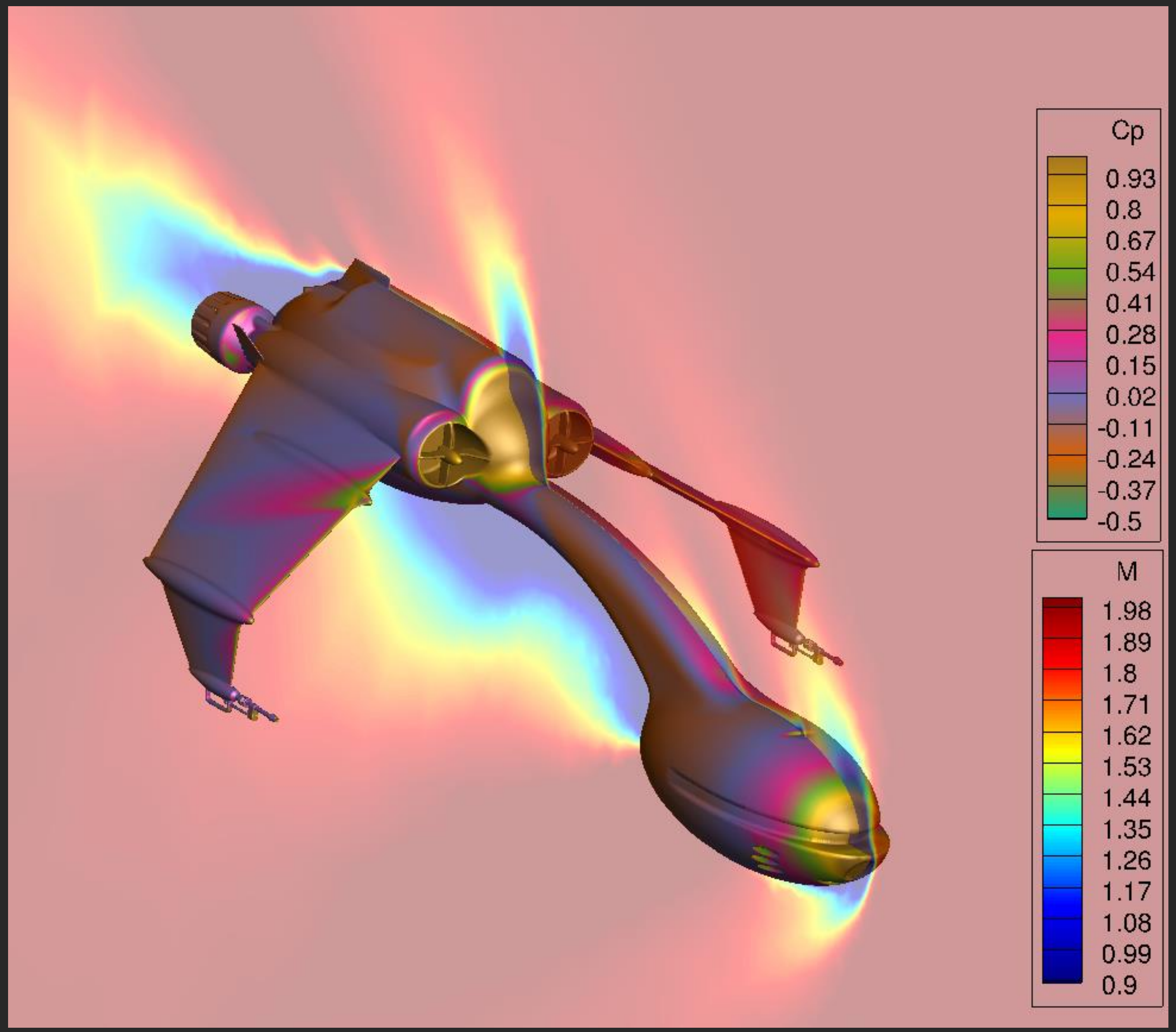




Amdahl's law (above) shows us how hard it is to scale an application even close to linearly.

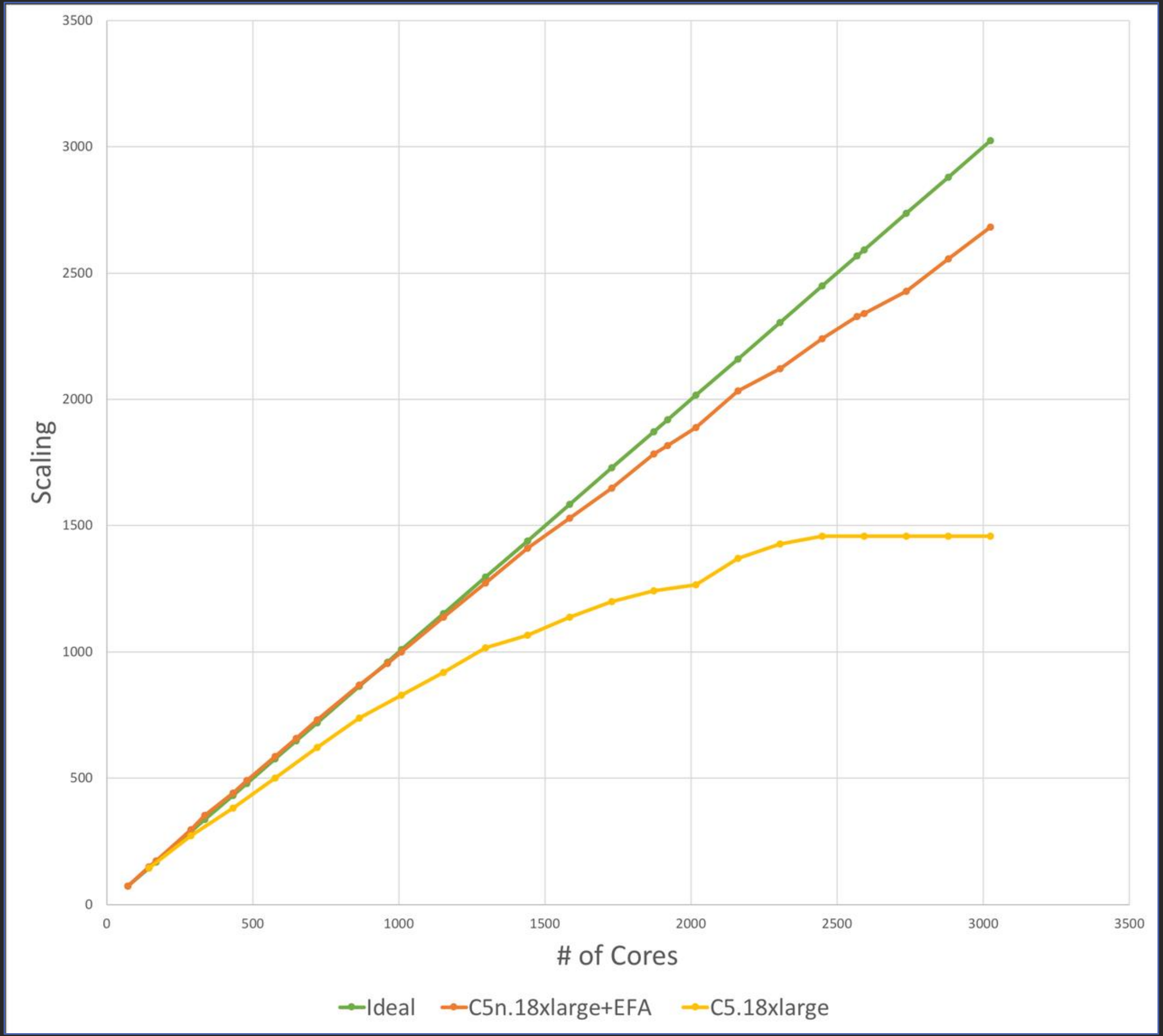
What can EFA do?

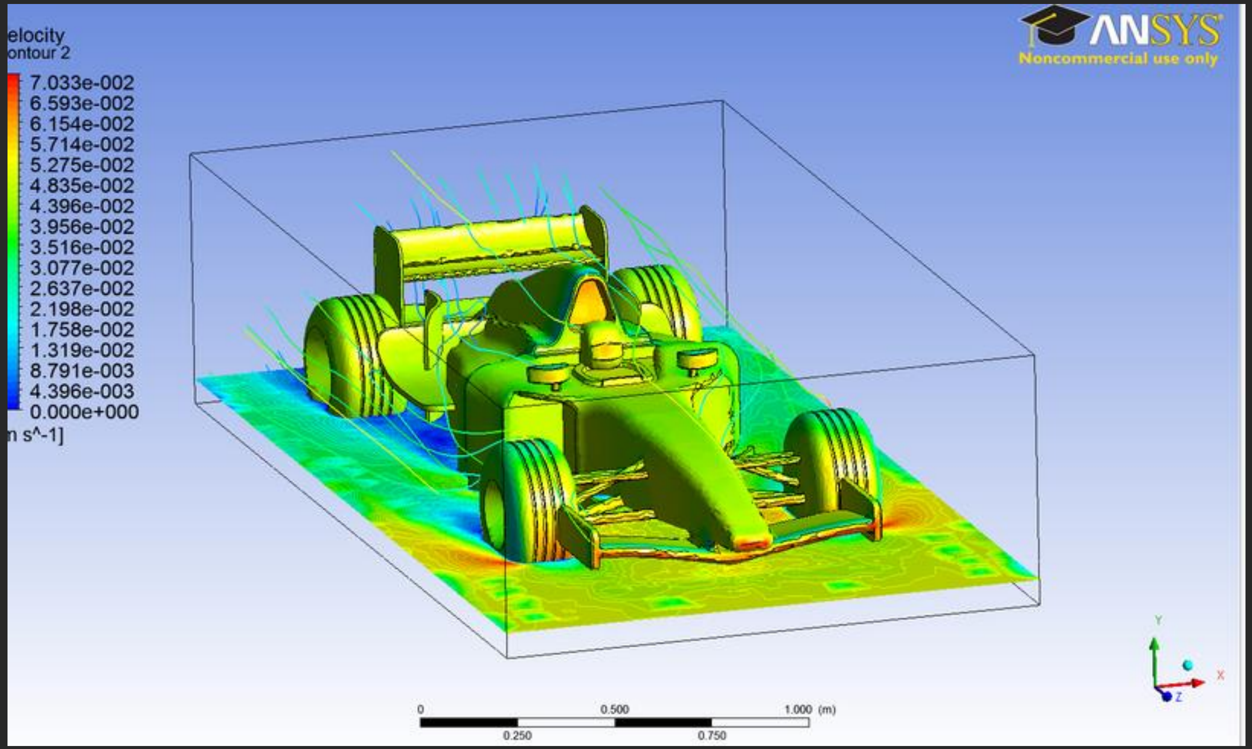




Thanks to Metacomp Technologies and the Klingon Empire. Garrrhhh.

ANSYS Fluent



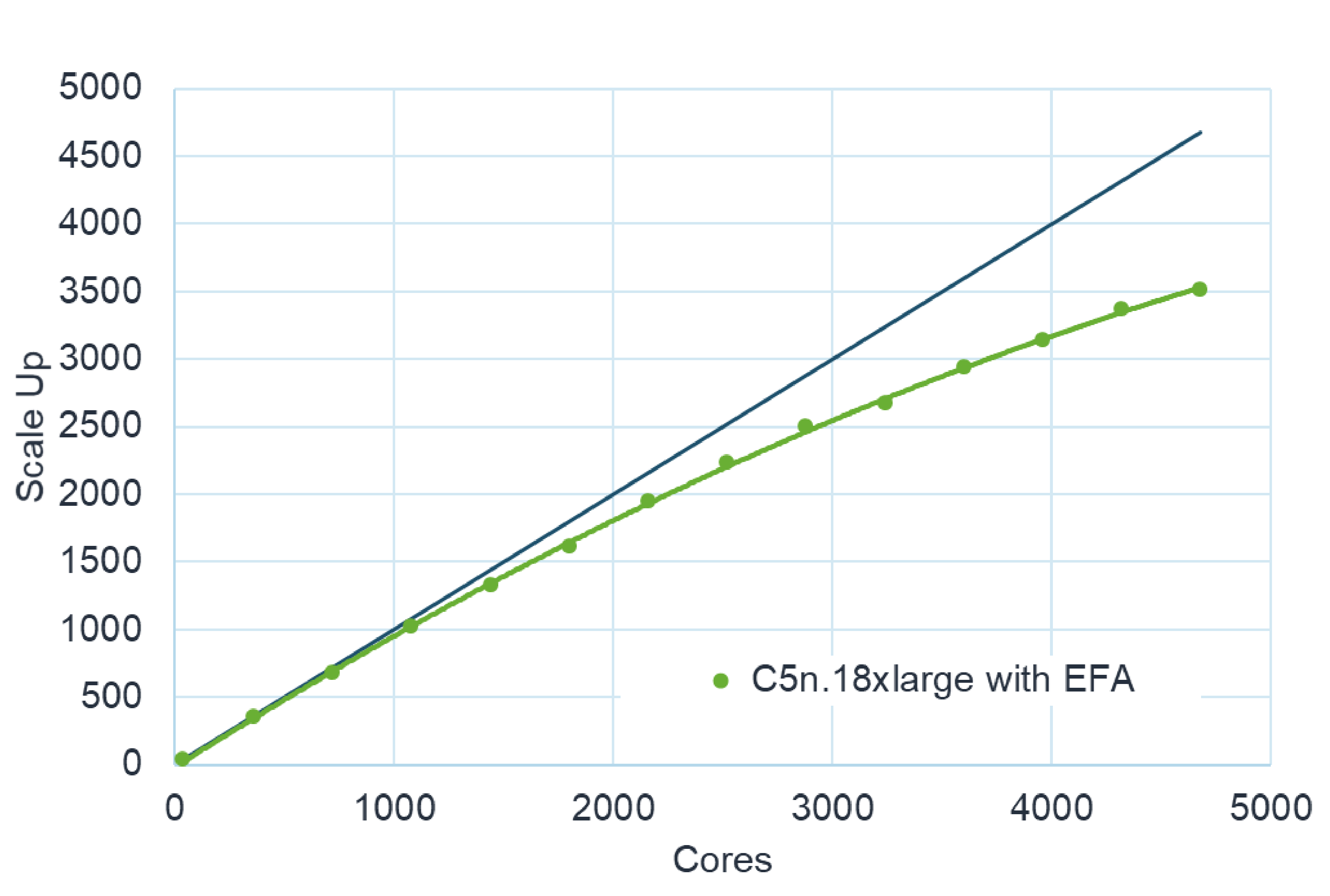


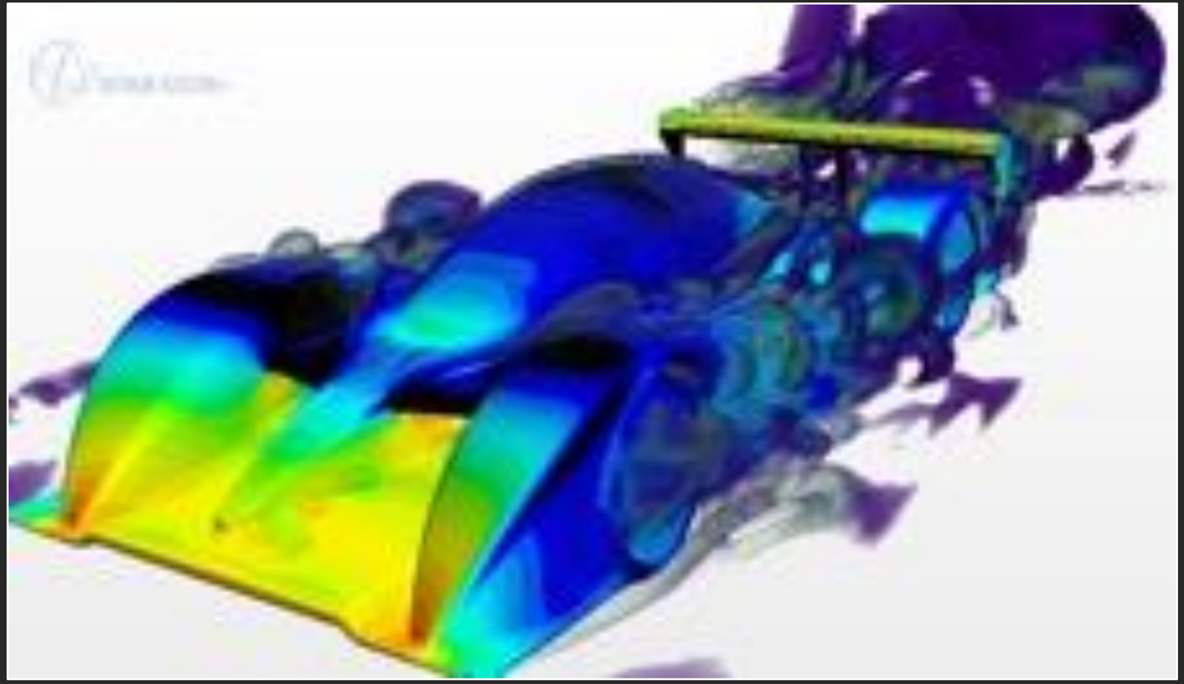
External flow over a Formula-1 Race Car (140M cell mesh)

At ~3,000 cores (~83 nodes), C5n+EFA shows ~89% scaling efficiency Vs ~48% using C5 w/o EFA

https://www.ansys.com/solutions/solutions-by-role/it-professionals/platform-support/benchmarks-overview/ansys-fluent-benchmarks/ansys-fluent-benchmarks-release-19/external-flow-over-a-formula-1-race-car

Siemens STAR-CCM+

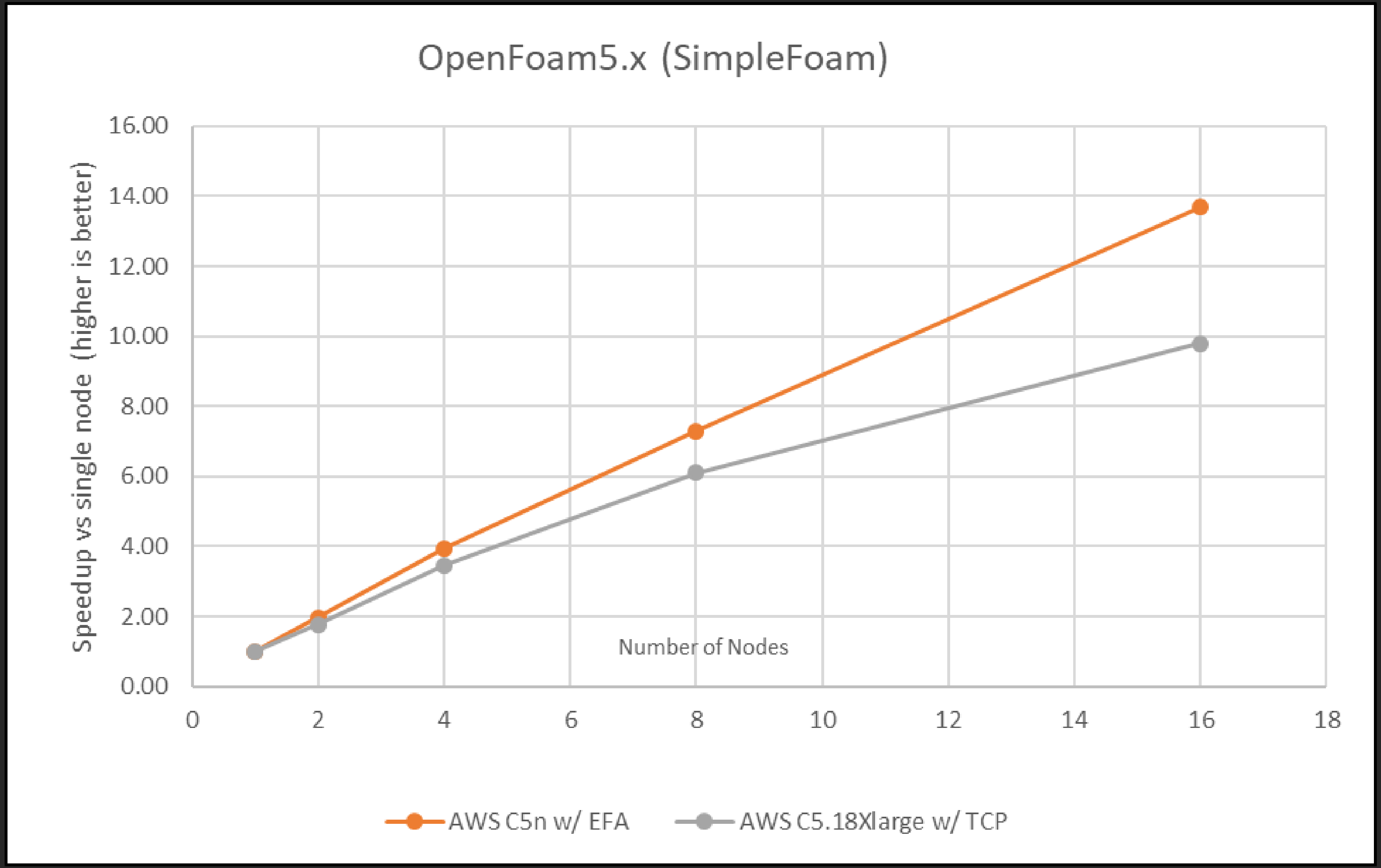


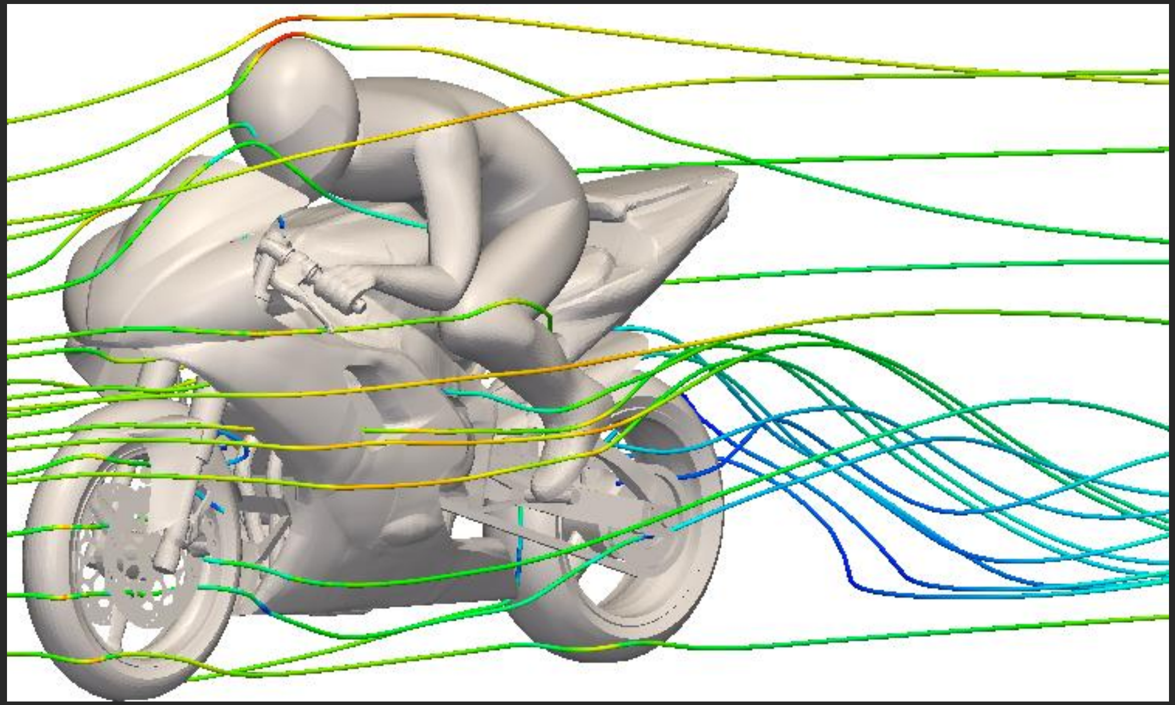


LeMans Racer (104M cell mesh) scaling with C5n+EFA with all cores fully utilized

At ~4,700 cores (~130 nodes), C5n+EFA shows ~76% scaling efficiency

OpenFOAM

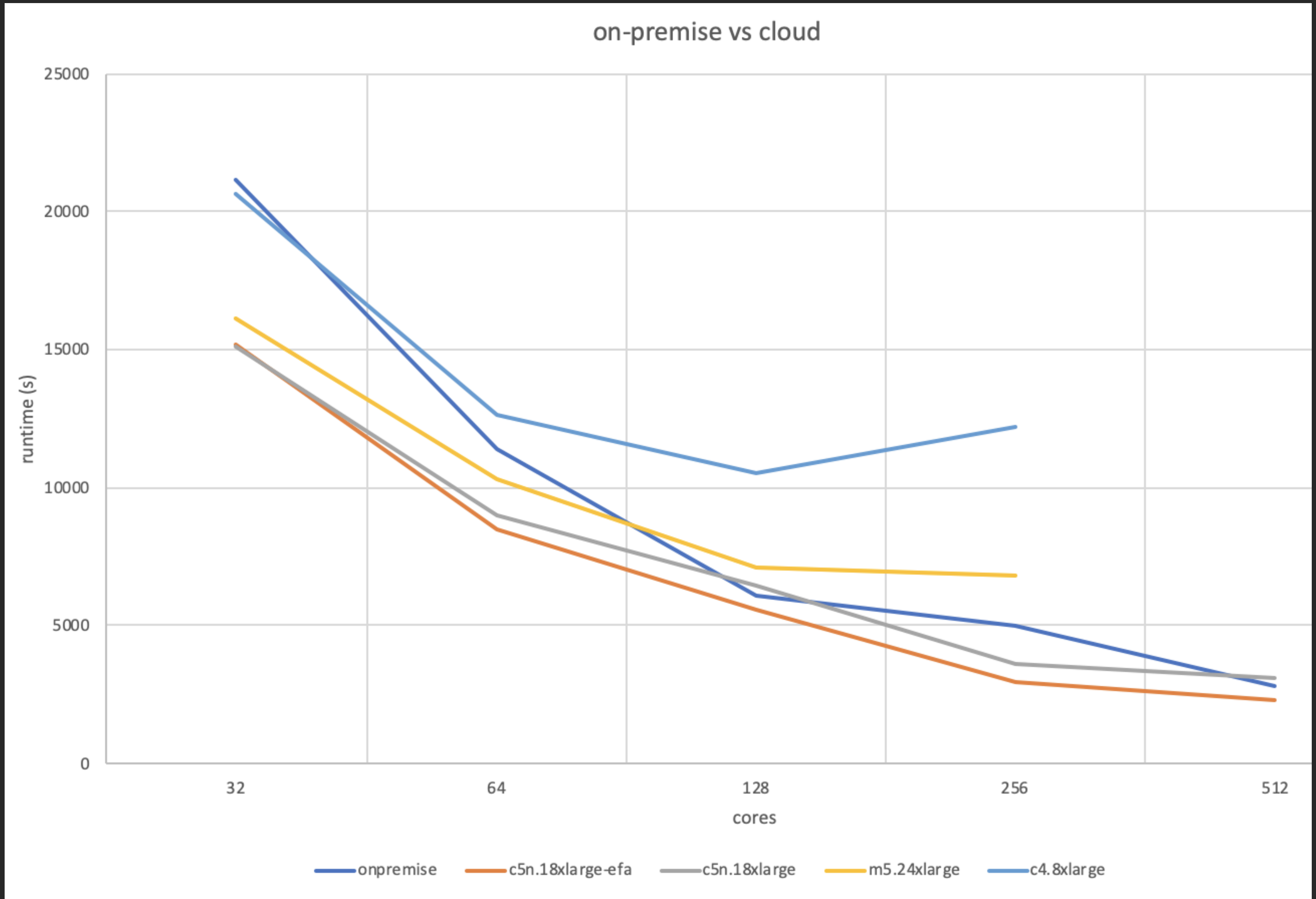




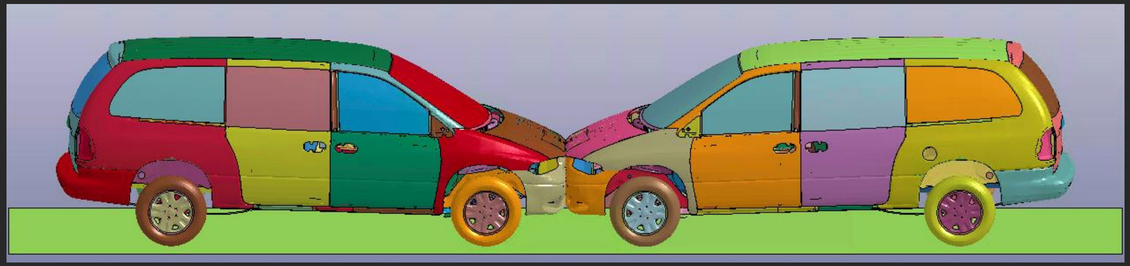
Motorbike (42M cell mesh) speedup with C5n+EFA Vs C5 without EFA

At 576 cores (16 nodes), C5n+EFA shows ~14X speedup Vs ~10X speedup with C5 without EFA

LSTC LS-DYNA



https://www.totalcae.com/learn/look-aws-efa-ls-dyna/



Car2Car time to completion with C5n + EFA Vs On-Premise, C5n, M5, and C4

At ~512 cores (~14 nodes), C5n+EFA shows ~25% faster time to completion over C5n w/o EFA

Weather

Background: Computing Resources

U.S. NAVAL RESEARCH LABORATORY

Navy Supercomputing

- Navy's arm of DoD HPC Modernization Program
- One of five DoD HPC Centers
- Headquartered with Naval Meteorology and Oceanography Command
- Supports various defense computational areas:
  - Climate/Weather/Ocean Modeling and Simulation
  - Computational Structural Mechanics
  - Computational Electromagnetics and Acoustics
  - Space and Astrophysical Science

Navy DSRC
Stennis Space Center, Mississippi

HPE SGI 8600
3.05 PFLOPS

Gaffney

Koehr

Cray XC40
2 PFLOPS

U.S. Naval Research Laboratory

	Navy DSRC - Conrad (Cray XC40)	AWS c4.8xlarge	Azure H16r	Penguin B30 queue	AWS c5n.18xlarge
CPU	- 2.3 GHz Intel Xeon E5-2698 v3 Broadwell - 32 core nodes	- 2.9 GHz Intel Xeon E5-2666 v3 Haswell - 18 core nodes	- 3.2 GHz Intel Xeon E5-2667 v3 - 14 core nodes	- 2.4 GHz Intel Xeon E5-2680 v4 Broadwell - 28 core nodes	- 3.0 GHz Intel Xeon Platinum w/ AVX-512 - 36 core nodes
Network	- Cray Aries / Dragonfly	- 25 Gbps ethernet with SRIOV	- FDR Infiniband	- Intel OmniPath	- AWS EFA

Software Configuration

- Intel Fortran 2018 update 1
- MPI:
  - Intel 2018 update 1
  - EFA: Open MPI 3.1.4
- HDF5 1.8.20

Platform Parameters

- Hyperthreading disabled
- AWS
  - us-west-2 (OR) region
  - Placement Groups
  - CloudFormation
- Azure
  - US Gov Arizona
  - VMSS
- Penguin on Demand
  - PBS resource manager

Talk at HPC User Forum at Argon National Labs in Sep 2019 by Daniel Arevalo from DeVine Consulting with work performed in conjunction with Tim Whitcomb @ NRL.

U.S. NAVAL RESEARCH LABORATORY

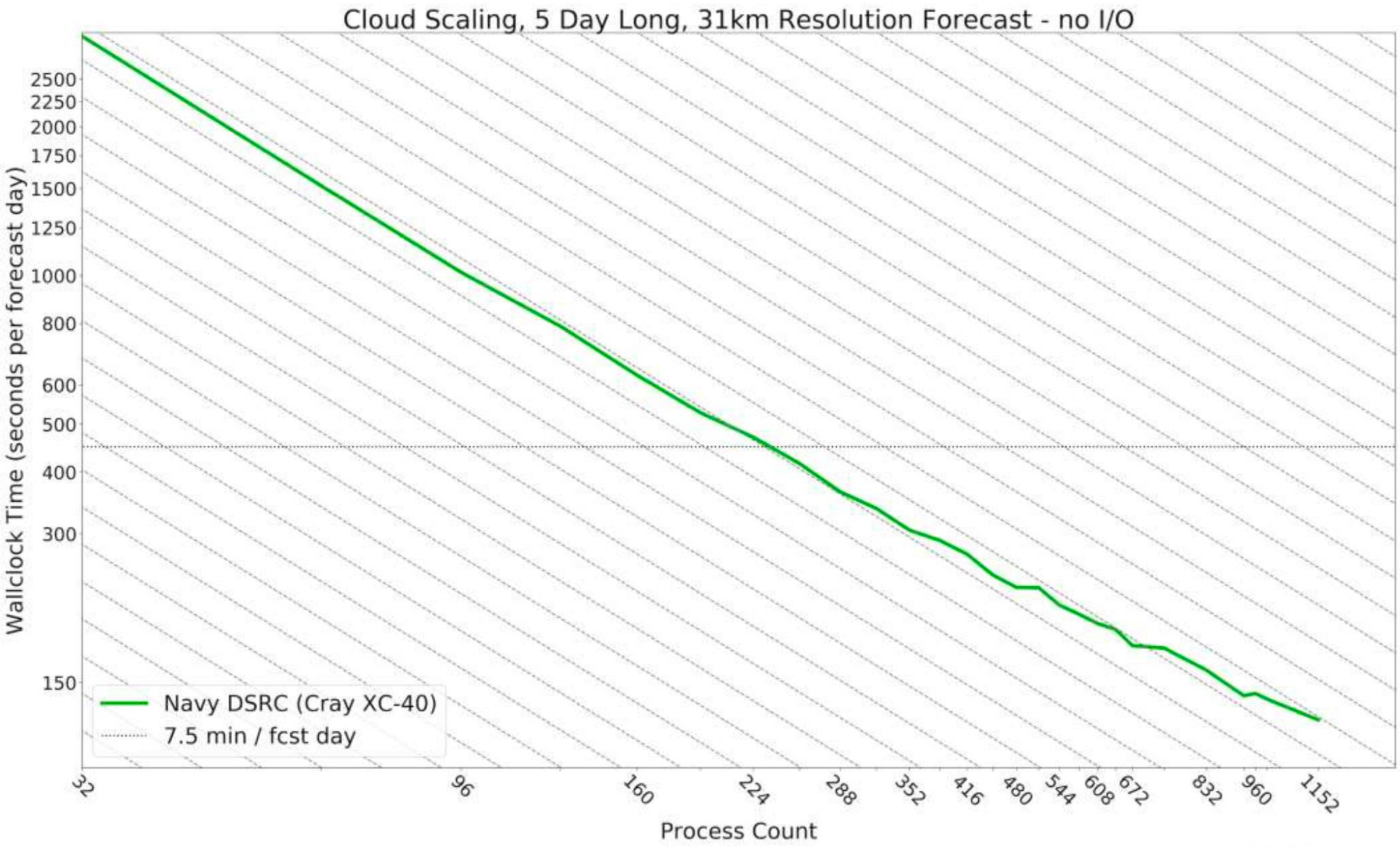
Low Resolution Forecast: Performance - Navy DSRC

Platform Specifications:

- 2.3 GHz Intel Xeon E5-2698 v3 Broadwell
- 32 core nodes
- Cray Aries / Dragonfly
- Cray Linux

Results

- Tested on Conrad
- Good scaling
  - minimal variability
  - minor increase in slope.

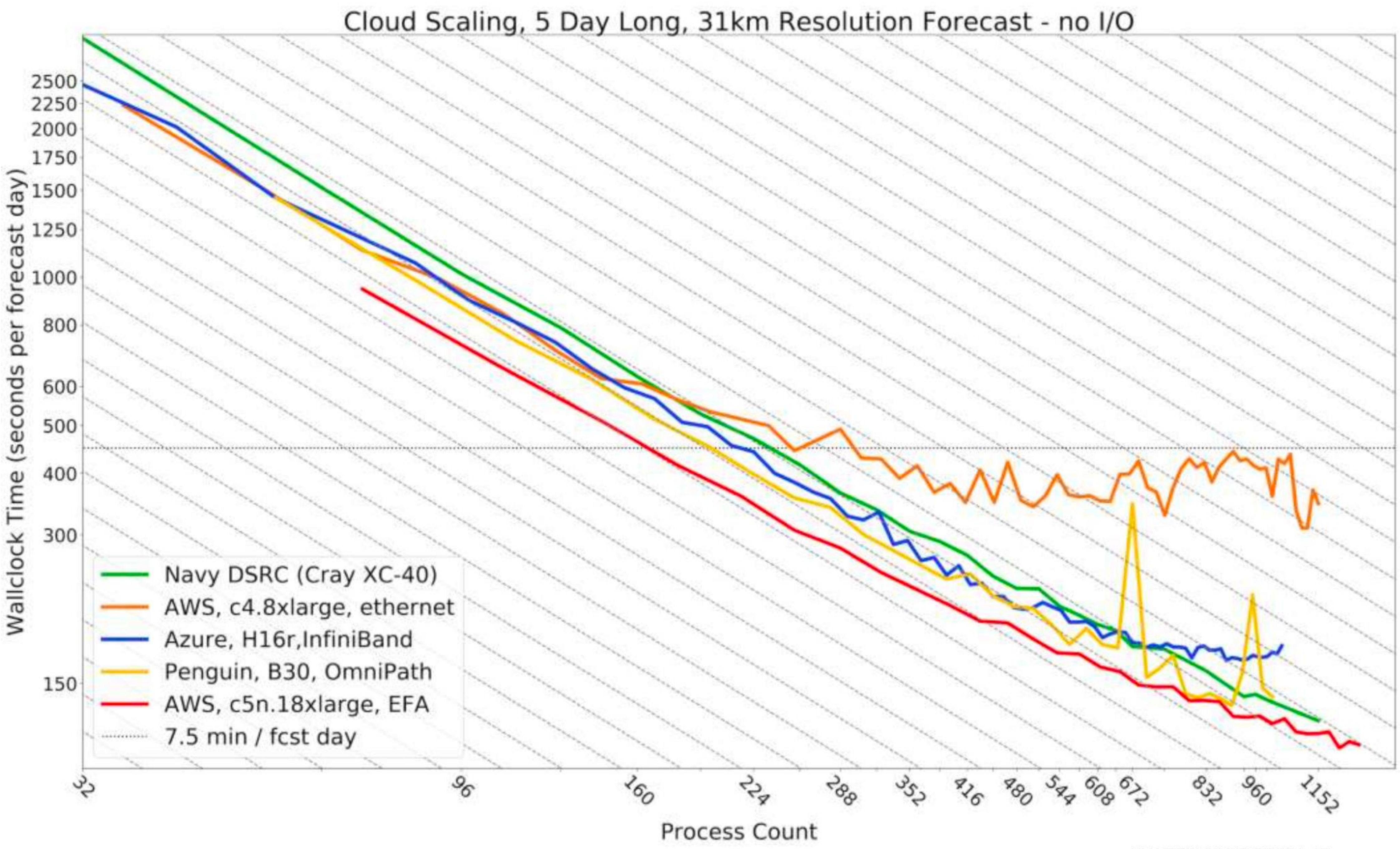


U.S. NAVAL RESEARCH LABORATORY

Low Resolution Forecast: Performance - Comparison

Performance Improvements:
c5n with EFA on AWS EC2

- At the highest core counts:
  - 13% faster than Penguin
  - 39% faster than Azure
  - 192% faster than previous AWS
  - 6% faster than Navy DSRC
- Min size estimated to meet 7.5 min:
  - 6% faster than Penguin
  - 16% faster than Azure
  - 74% faster than previous AWS
  - 29% faster than Navy DSRC
- Min size cost estimate:
  - Penguin: $12.95
  - Azure: $21.31
  - Previous AWS: $18.65
  - C5n with EFA: $13.76



High Resolution Forecast: Performance - Comparison

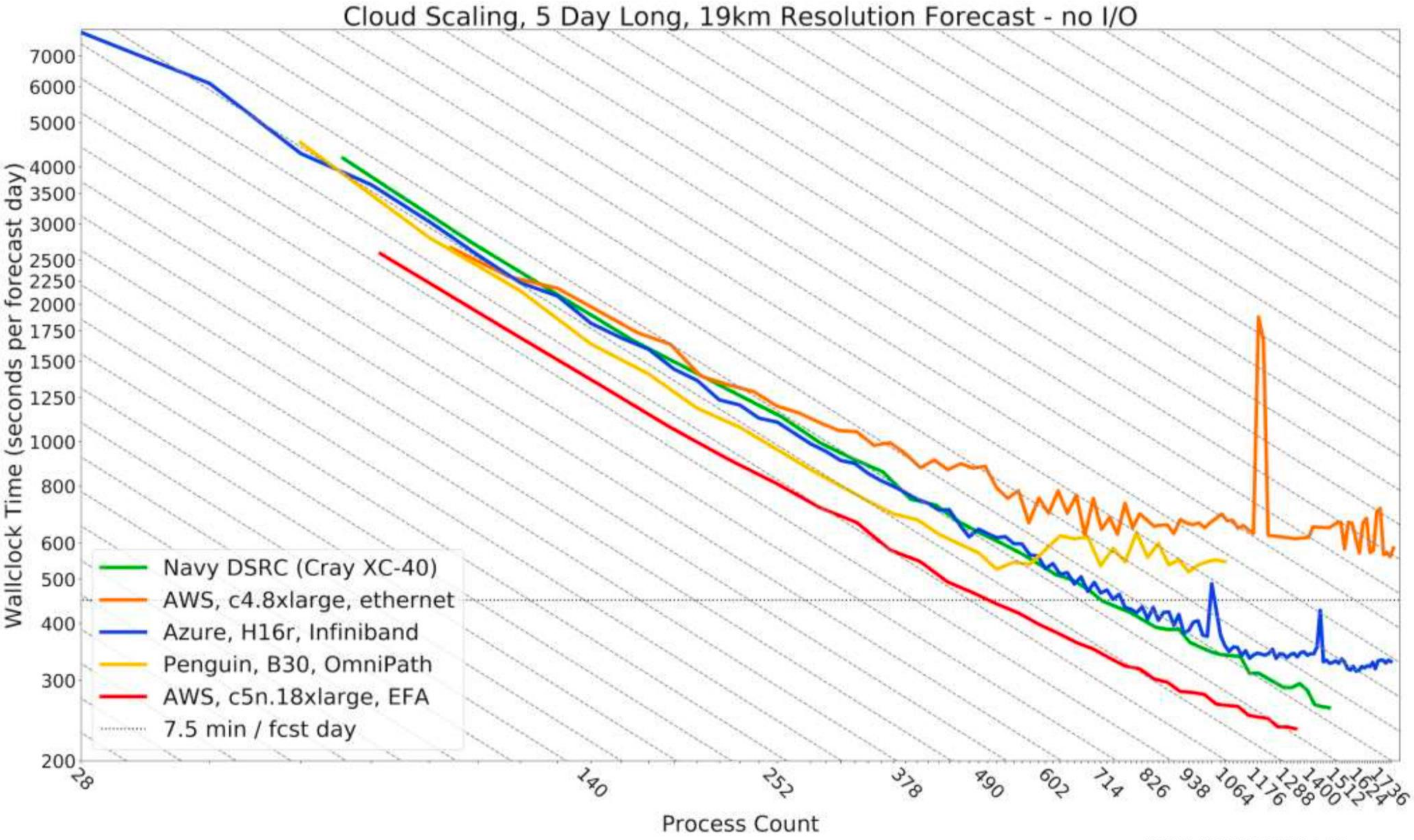
Performance Improvements:

C5n with EFA on AWS EC2

- At the highest core counts:
  - 107% faster than Penguin
  - 43% faster than Azure
  - 160% faster than previous AWS
  - 25% faster than Navy DSRC

- Min size estimated to meet 7.5 min:
  - 33% faster than Azure
  - 23% faster than Navy DSRC

- Min size forecast cost estimate:
  - Azure: $82.97
  - C5n with EFA: $44.02



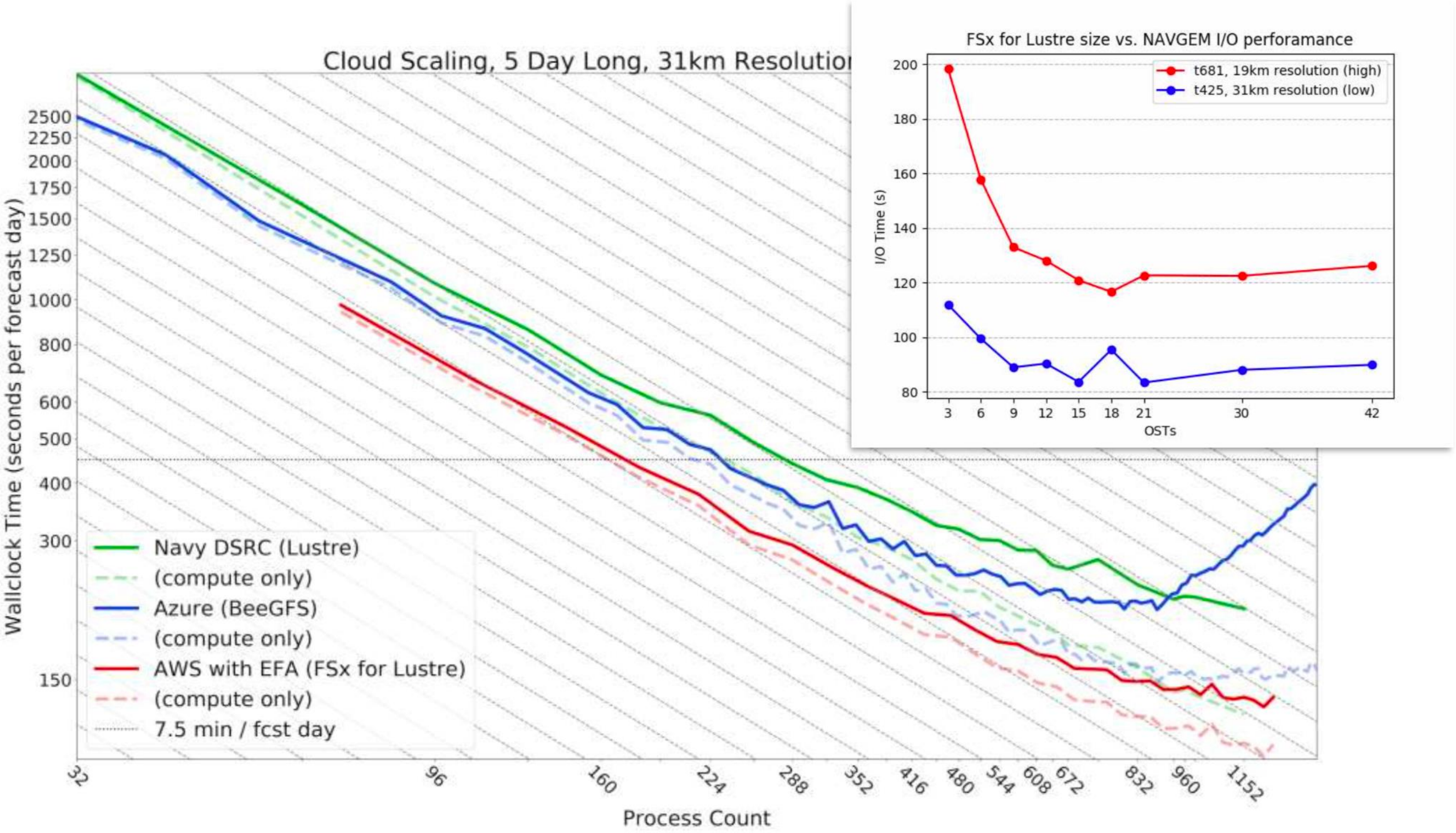
Next Steps

Incorporate I/O

- Initial testing looks promising.
- More “tuning” necessary to optimize results.

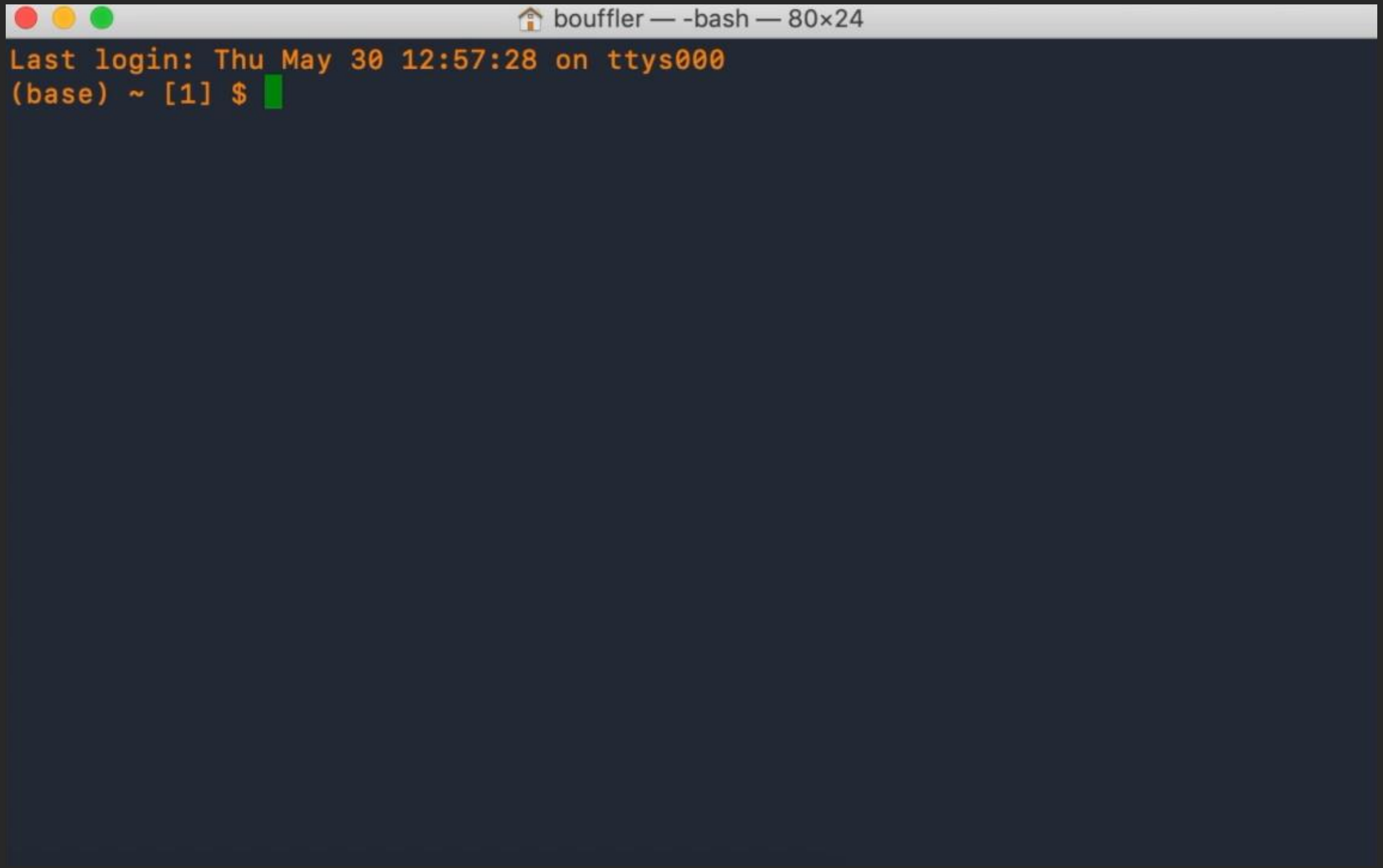
Future Areas of Research

- Test full NAVGEM ensemble
- Test next-generation forecast programs – NEPTUNE
- Incorporate on-going updates to cloud systems to further reduce costs and optimize performance.



Infrastructure is code

Not a 5-yearly refresh



- **Iteratively** decide on the best CPU, GPU, memory or I/O architecture for your workload
- Test multiple options in **parallel** rather than sequentially
- **Dispose** of what you don't need (mercilessly, and without harming any animals :-)
- Make **CI/CD** part of your HPC practice

Getting your hands on EFA – AWS ParallelCluster

```
[global]
cluster_template = efa
update_check = true
sanity_check = true

[aws]
aws_region_name = [your_aws_region]

[cluster efa]
key_name =                  [your_keypair]
vpc_settings =              public
base_os =                   alinux
master_instance_type =      c5.xlarge
compute_instance_type =     c5n.18xlarge
placement_group =           DYNAMIC
enable_efa =                compute

[vpc public]
vpc_id = [your_vpc]
```

```
$ pcluster create efa
Status: CREATE_COMPLETE
MasterServer: RUNNING
MasterPublicIP: 3.215.238.41
ClusterUser: ec2-user
MasterPrivateIP: 172.31.25.64
```

```
[cluster] $ module avail
openmpi/3.1.3 use.own
[cluster] $ module load openmpi/3.1.4
[cluster] $ which mpirun
/opt/amazon/efa/bin/mpirun
```

https://aws.amazon.com/blogs/opensource/scale-hpc-workloads-elastic-fabric-adapter-and-aws-parallelcluster/

AWS ParallelCluster + Amazon FSx for Lustre

Compute clusters & storage clusters

- Specify an FSx Lustre requirement for your compute cluster
- Define the S3 bucket you'd like to hydrate the filesystem from/to
- If you're replicating S3 (or someone else is – e.g., a public data source), you could spin up clusters in multiple regions, following the lowest spot price without losing capability

FSx

Configuration for an attached FSx Lustre file system. See FSx CreateFileSystem for more information.

FSx Lustre is supported when `base_os = centos7 | alinux`.

When using an Amazon Linux `custom_ami`, the kernel must be >= `4.14.104-78.84.amzn1.x86_64`. See Installing the Lustre Client for instructions.

Note FSx is not currently supported when using `awsbatch` as a scheduler.

If using an existing file system, it must be associated to a security group that allows inbound and outbound TCP traffic from `0.0.0.0/0` through port `988`. This is done by automatically when not using `vpc_security_group_id`.

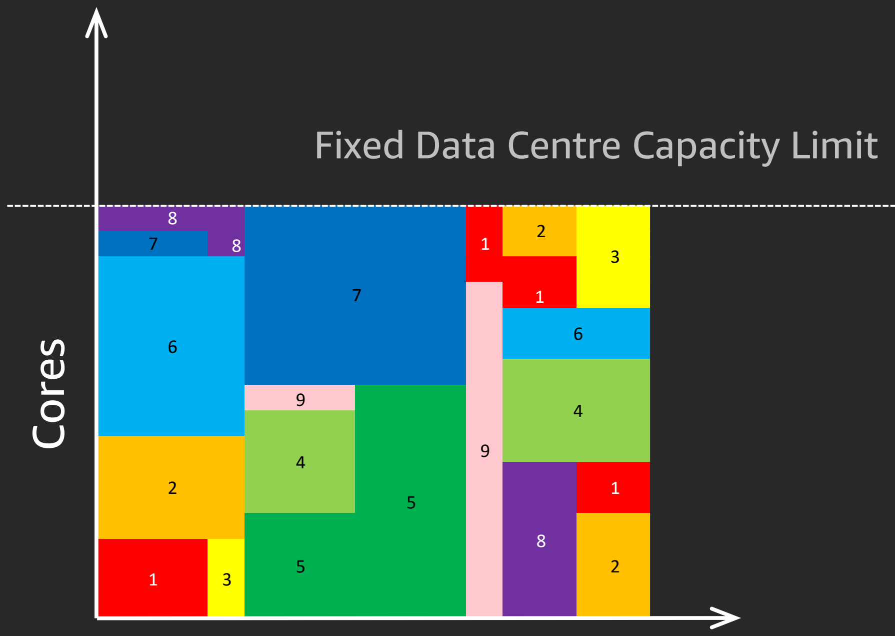
Use an existing FSx file system by specifying `fsx_fs_id`.

```
[fsx fs]
shared_dir = /fsx
fsx_fs_id = fs-073c3803dca3e28a6
```

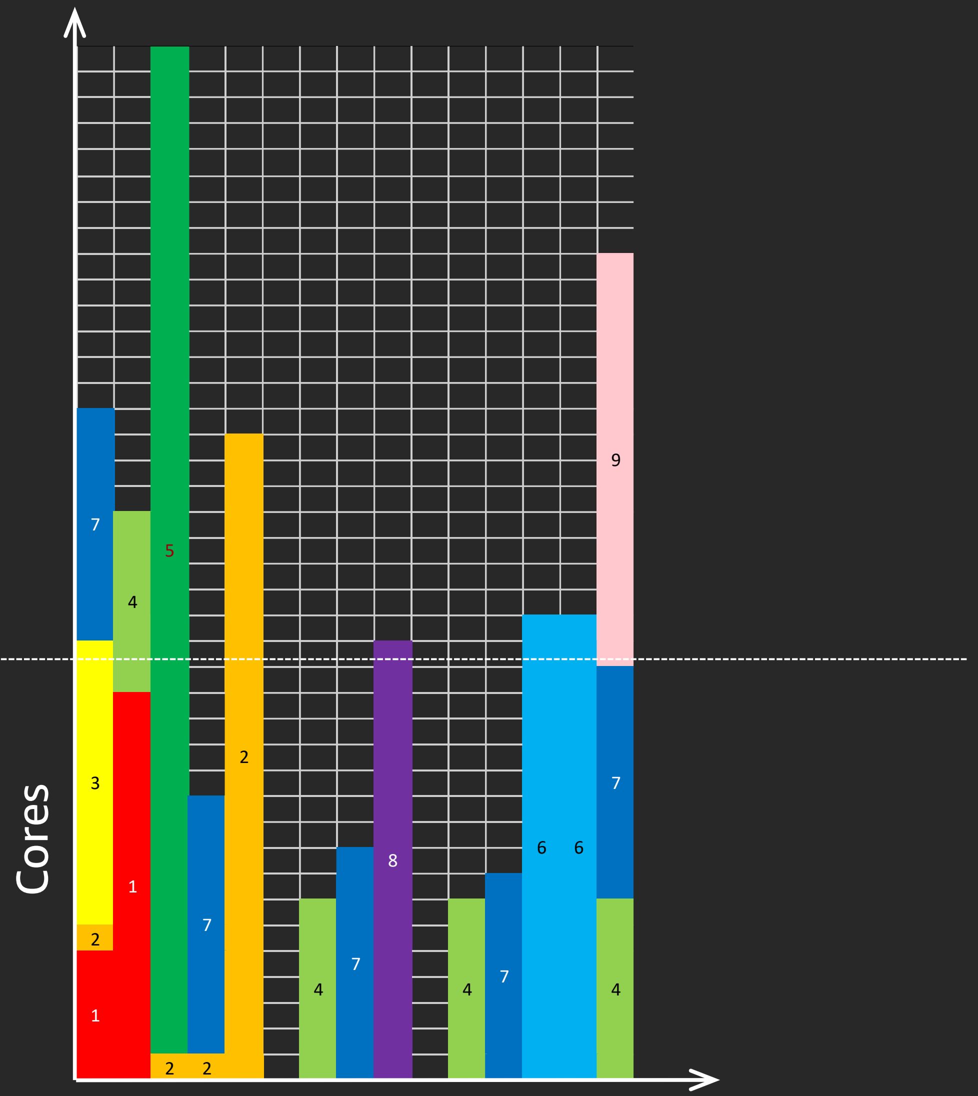
Or create and configure a new file system, with the following parameters

```
[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
import_path = s3://bucket
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
weekly_maintenance_start_time = 1:00:00
```

The metric for success should be time-to-results



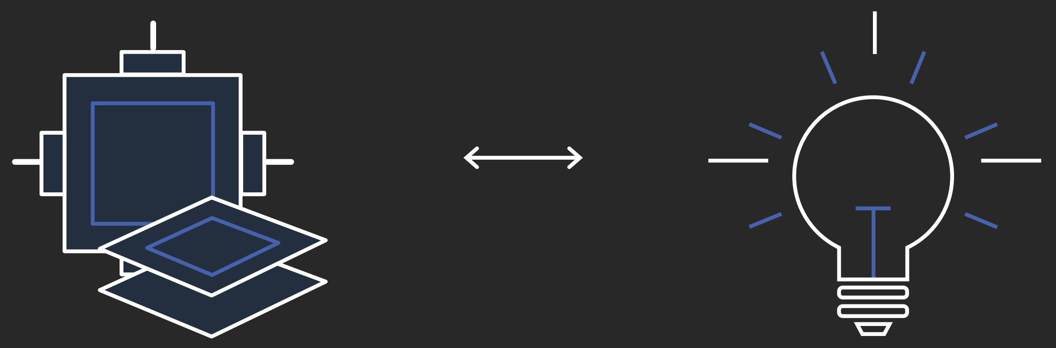
Finite capacity, usually with long queues to wait in.



Massive capacity when needed to speed up time to results, and agile environment when additional hardware and software experimentation is needed.

"For every $1 spent on HPC, businesses see $463 in incremental revenues and $44 in incremental profit."*

Source: Hyperion ROI Study (http://www.hyperionresearch.com/roi-with-hpc/)



HPC on AWS

Flexible configuration and virtually unlimited scalability to grow and shrink your infrastructure as your HPC workloads dictate, not the other way around

Thank you!

Brendan Bouffler
bouffler@amazon.com

Chris Liu
chrliu@amazon.com