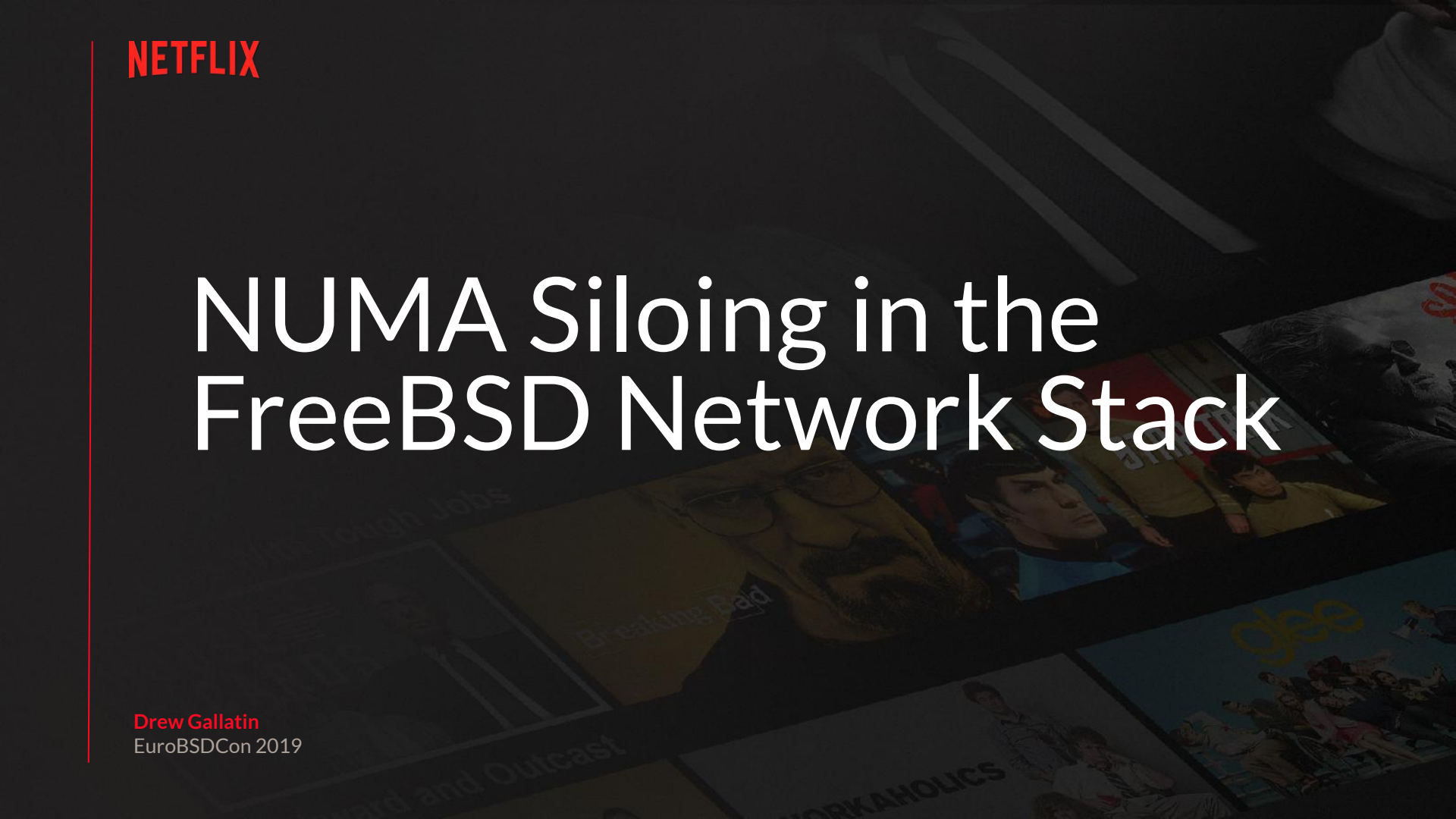
NETFLIX

NUMA Siloing in the FreeBSD Network Stack

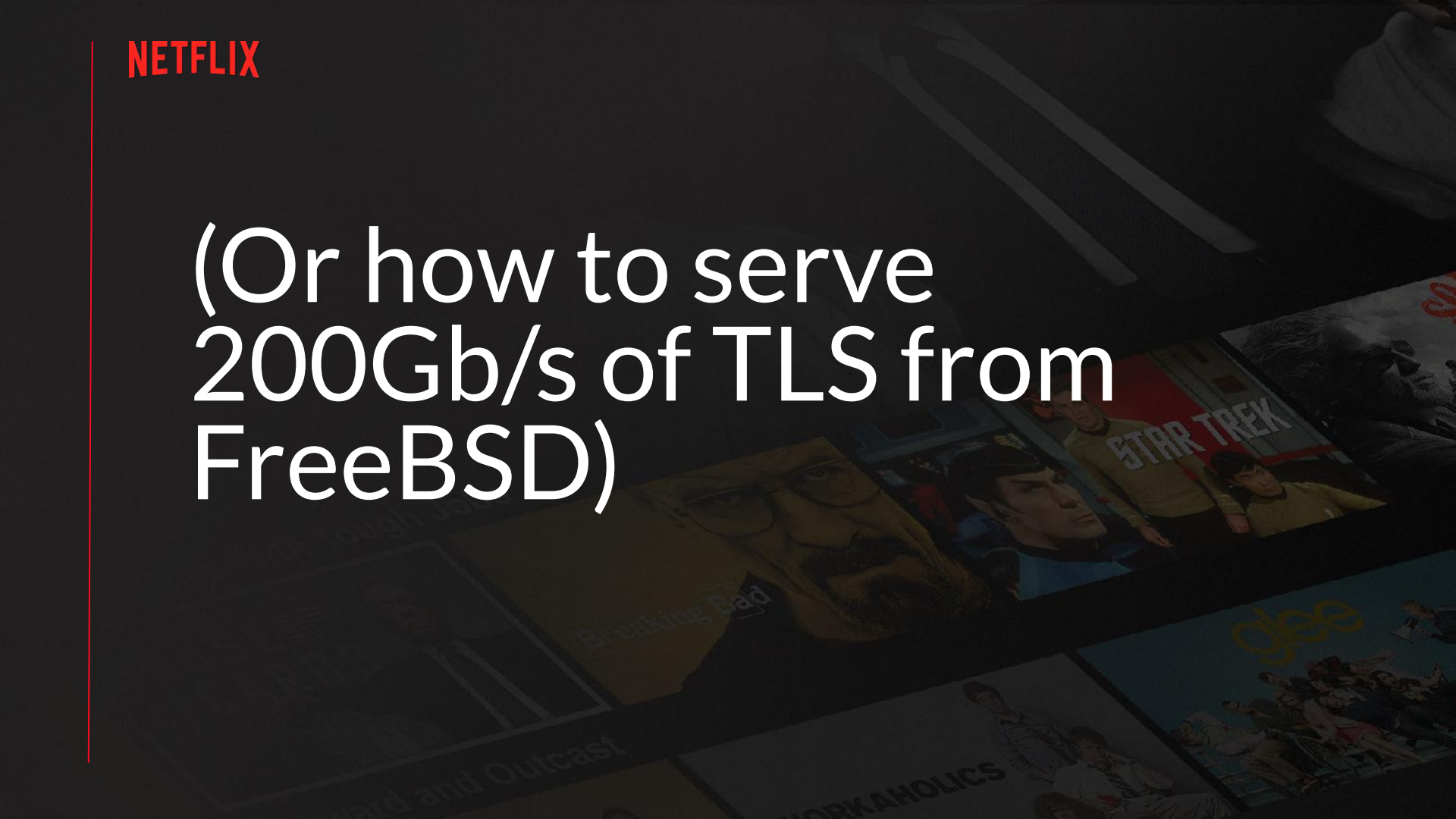


Drew Gallatin

EuroBSDCon 2019

NETFLIX

(Or how to serve 200Gb/s of TLS from FreeBSD)



NETFLIX

Motivation:

- Since 2016, Netflix has been able to serve 100Gb/s of TLS encrypted video traffic from a single server.
- How can we serve ~200Gb/s of video from a single server?

NETFLIX

Netflix Video Serving Workload

- FreeBSD-current
- NGINX web server
- Video served via sendfile(2) and encrypted using software kTLS
  - TCP_TXTLS_ENABLE from tcp(4)

NETFLIX

Why do we need NUMA for 200Gb/s ?

NETFLIX

Netflix Video Serving Hardware for 100Gb/s

- Intel “Broadwell” Xeon (original 100g)
  - 60GB/s mem bw
  - 40 lanes PCIe Gen3
    - ~32GB/s of IO bandwidth
- Intel “Skylake” & “Cascade Lake” Xeon (new 100g)
  - 90GB/s mem bw
  - 48 lanes PCIe Gen 3
    - ~38GB/s of IO bandwidth

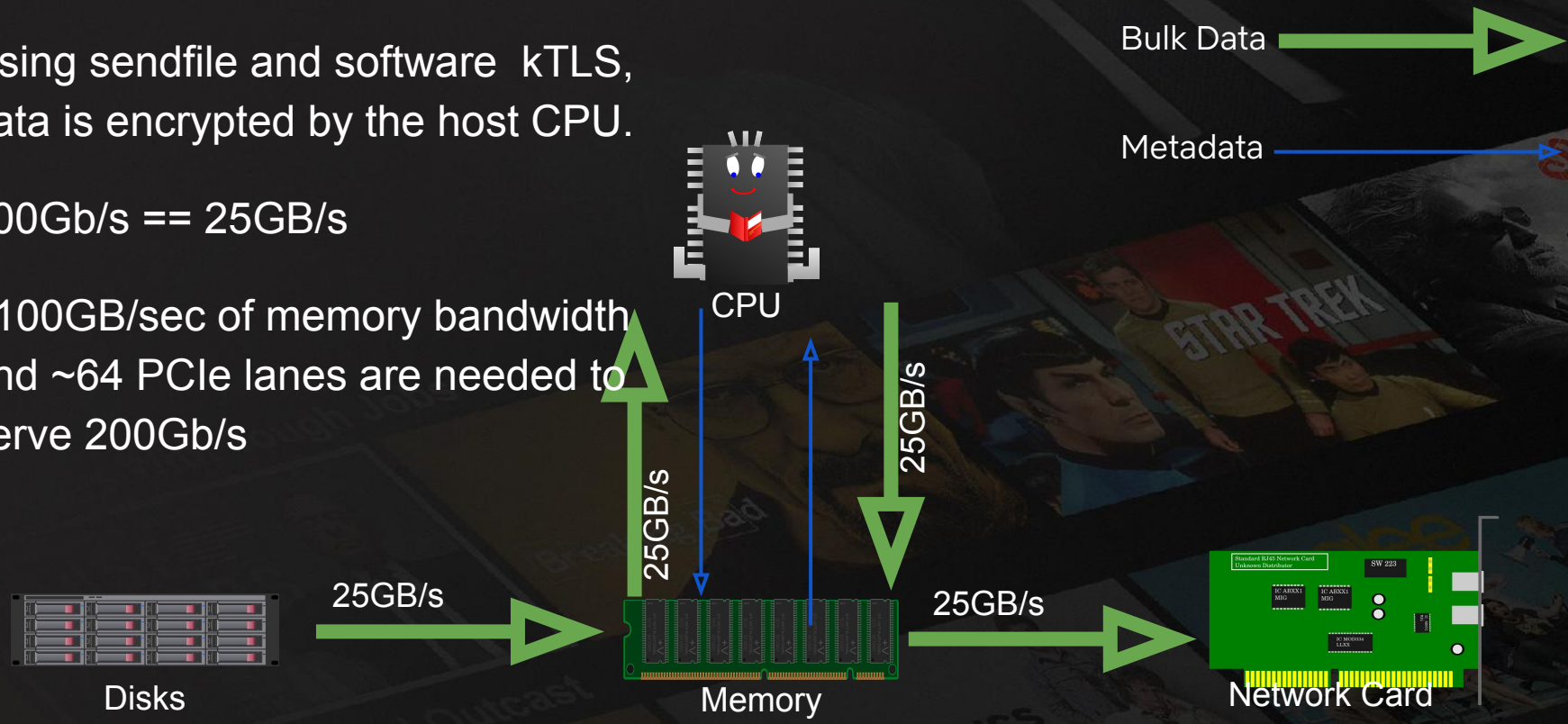
NETFLIX

Netflix 200Gb/s Video Serving Data Flow

Using sendfile and software kTLS, data is encrypted by the host CPU.

200Gb/s == 25GB/s

~100GB/sec of memory bandwidth and ~64 PCIe lanes are needed to serve 200Gb/s



NETFLIX

Netflix Video Serving Hardware for 200Gb/s (Intel) "Throw another CPU socket at it"

- 2x Intel "Skylake" / "Cascade Lake" Xeon
  - Dual Xeon(R) Silver 4116 / 4216
  - 2 UPI links connecting Xeons
  - 180GB/s (2 x 90GB/s) mem bw
  - 96 (2 x 48) lanes PCIe Gen 3
    - ~75GB/s IO bandwidth

NETFLIX

Netflix Video Serving Hardware for 200Gb/s (Intel)

- 8x PCIe Gen3 x4 NVME
  - 4 per NUMA node
- 2x PCIe Gen3 x16 100GbE NIC
  - 1 per NUMA node

NETFLIX

Netflix Video Serving Hardware for 200Gb/s (AMD) “4 chips in 1 socket”

- AMD EPYC “Naples” / “Rome”
  - 7551 & 7502P
  - Single socket, quad “Chiplet”
  - Infinity Fabric connecting chiplets
  - 120-150GB/s mem bw
  - 128 lanes PCIe Gen 3 (Gen 4 for 7502P)
    - 100GB/sec IO BW (200GB/s Gen 4)

NETFLIX

Netflix Video Serving Hardware for 200Gb/s (AMD) “4 chips in 1 socket”

- 8x PCIe Gen3 x4 NVME
  - 2 per NUMA node
- 4x PCIe Gen3 x16 100GbE NIC
  - 1 per NUMA node

NETFLIX

Initial 200G prototype performance:

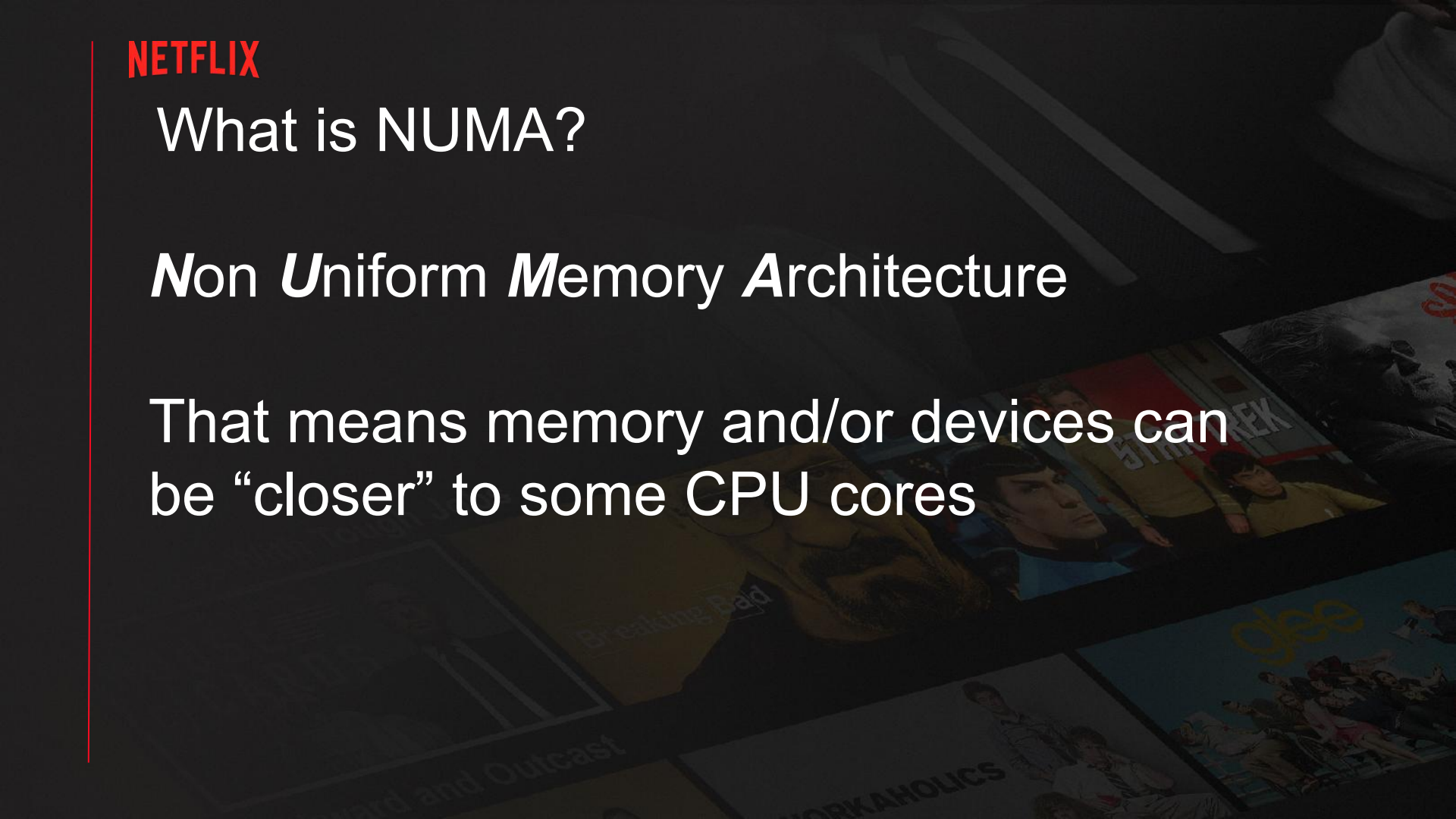
- 85Gb/s (AMD)
- 130Gb/s (Intel)
- 80% CPU
- ~40% QPI saturation
  - Measured by Intel's pcm.x tool from the intel-pcm port
- Unknown Infinity Fabric saturation
  - AMD's tools are lacking (even on Linux)

NETFLIX

What is NUMA?

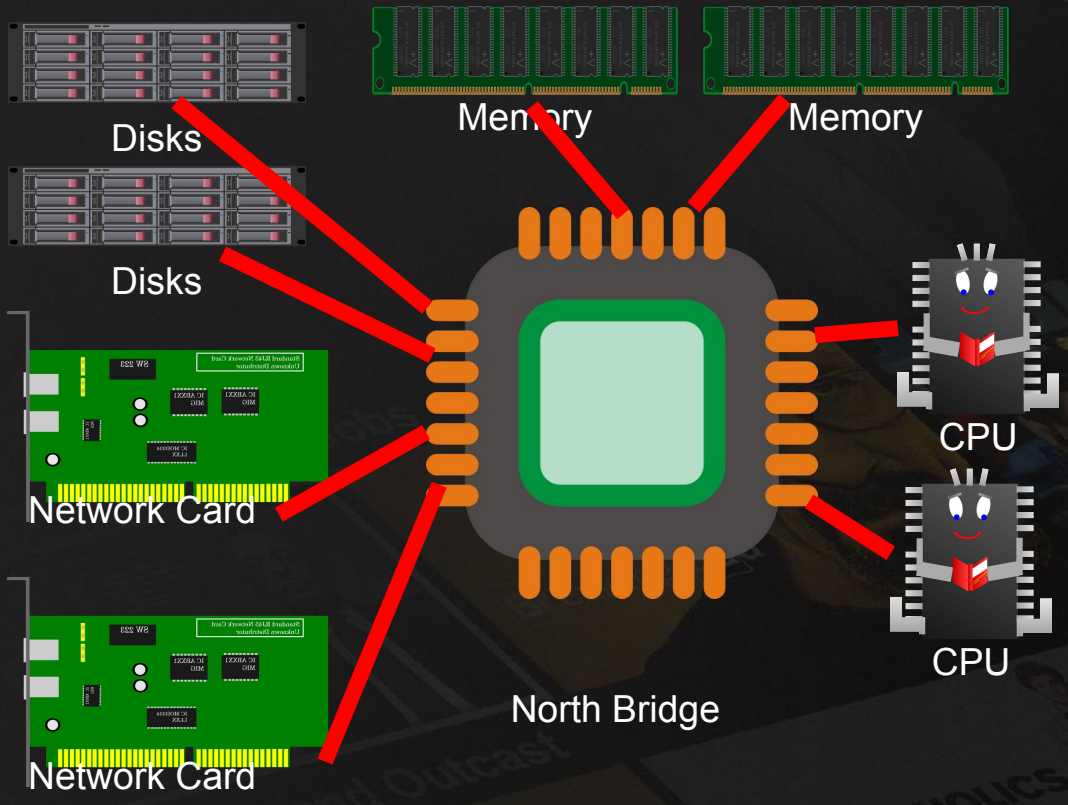
Non ***U***niform ***M***emory ***A***rchitecture

That means memory and/or devices can be “closer” to some CPU cores



NETFLIX

Multi Socket Before NUMA



Memory access was *UNIFORM:*

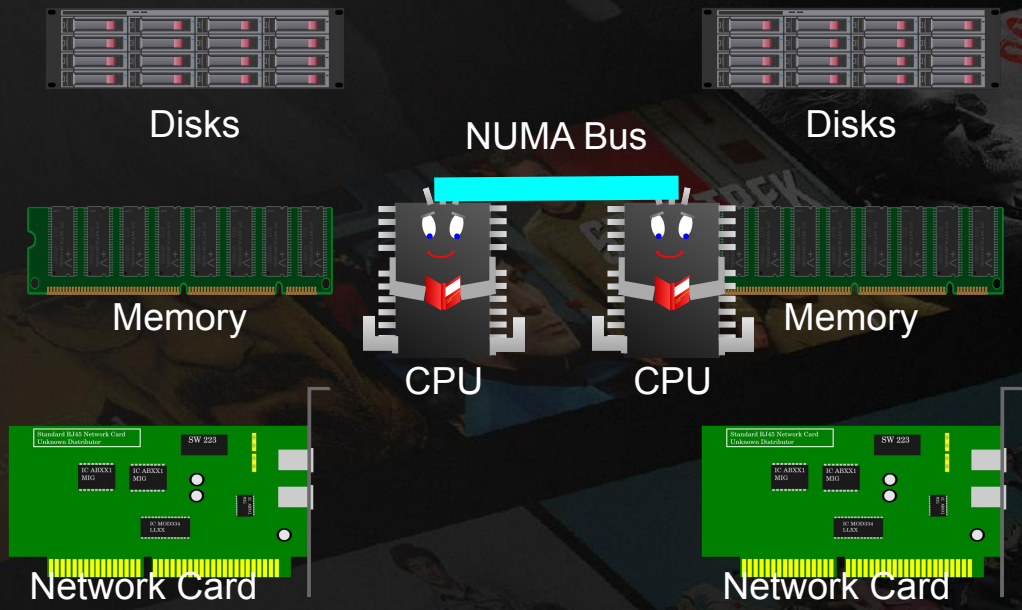
Each core had equal and direct access to all memory and IO devices.

NETFLIX

Multi Socket system with NUMA:

Memory access can be ***NON-UNIFORM***

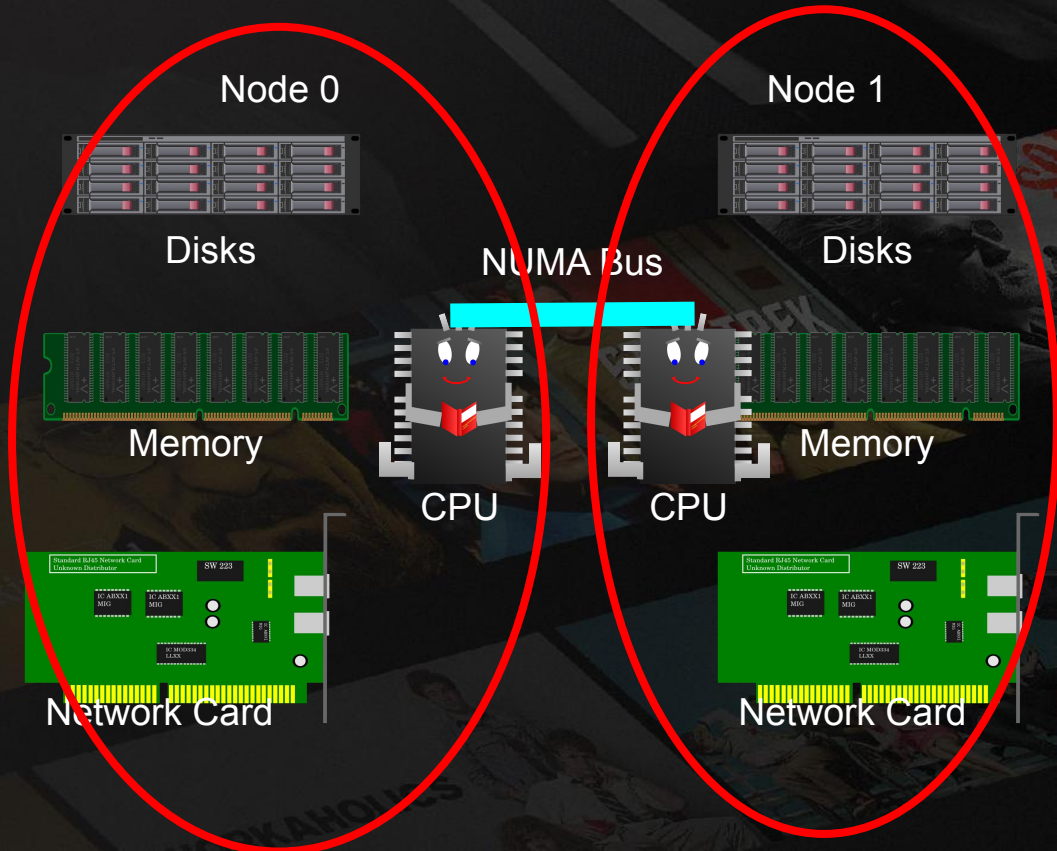
- Each core has unequal access to memory
- Each core has unequal access to I/O devices



NETFLIX

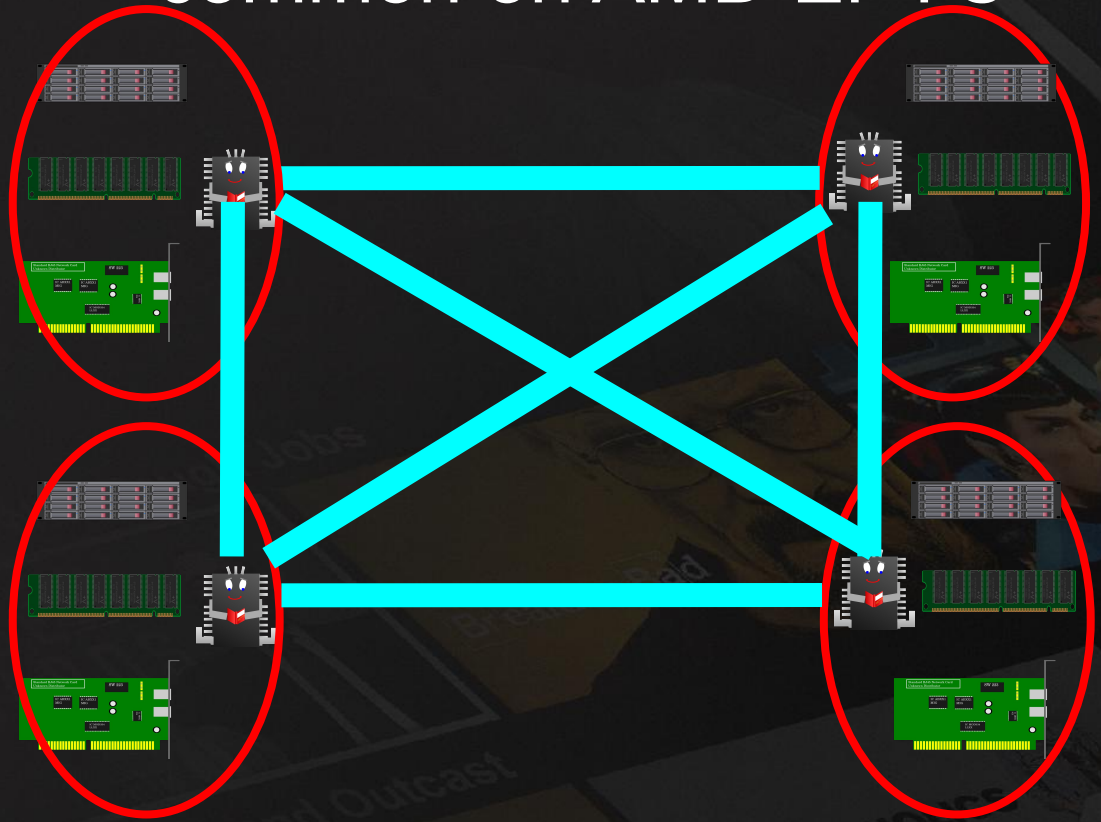
Present day NUMA:

Each locality zone called a “NUMA Domain” or “NUMA Node”



NETFLIX

4 Node configurations are common on AMD EPYC

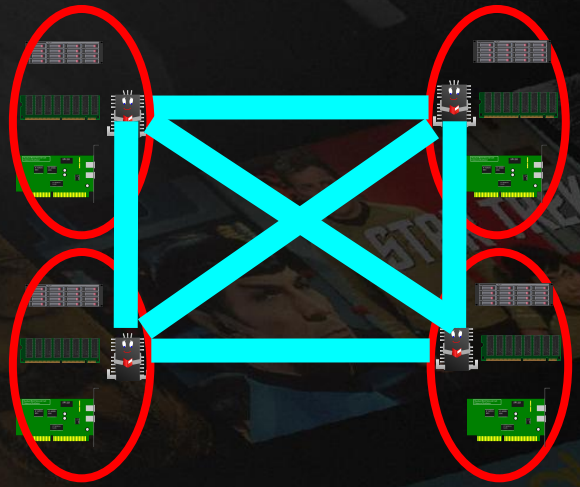


NETFLIX

Cross-Domain costs

Latency Penalty:

- ~50ns unloaded
- Much, much, much more than 50ns loaded

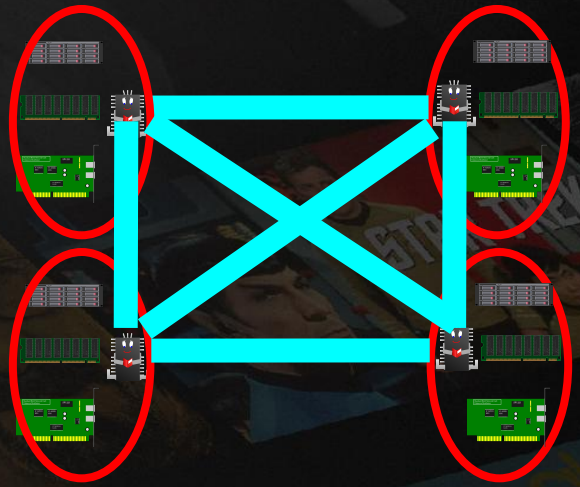


NETFLIX

Cross-Domain costs

Bandwidth Limit:

- Intel UPI
  - ~20GB/sec per link
  - Normally 2 or 3 links
- AMD Infinity Fabric
  - ~40GB/s



NETFLIX

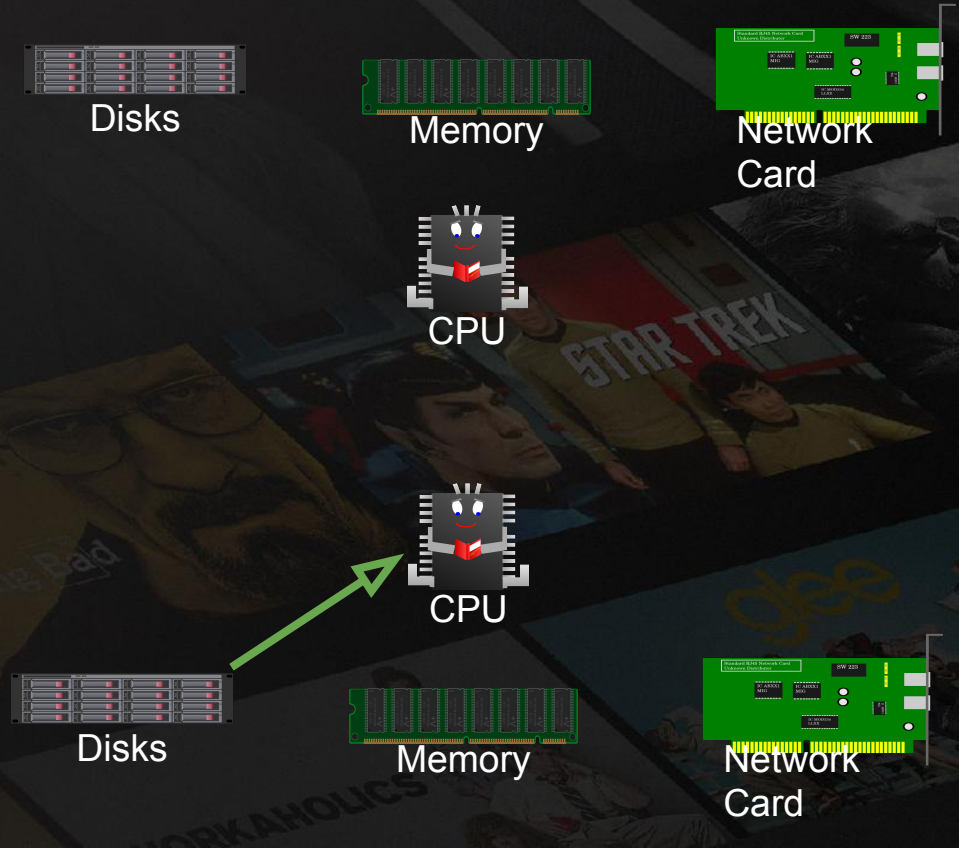
Strategy: Keep as much of our 100GB/sec of bulk data off the NUMA fabric is possible

- Bulk data congests NUMA fabric and leads to CPU stalls.

NETFLIX Dual Xeon: Worst Case Data Flow

Steps to send data:

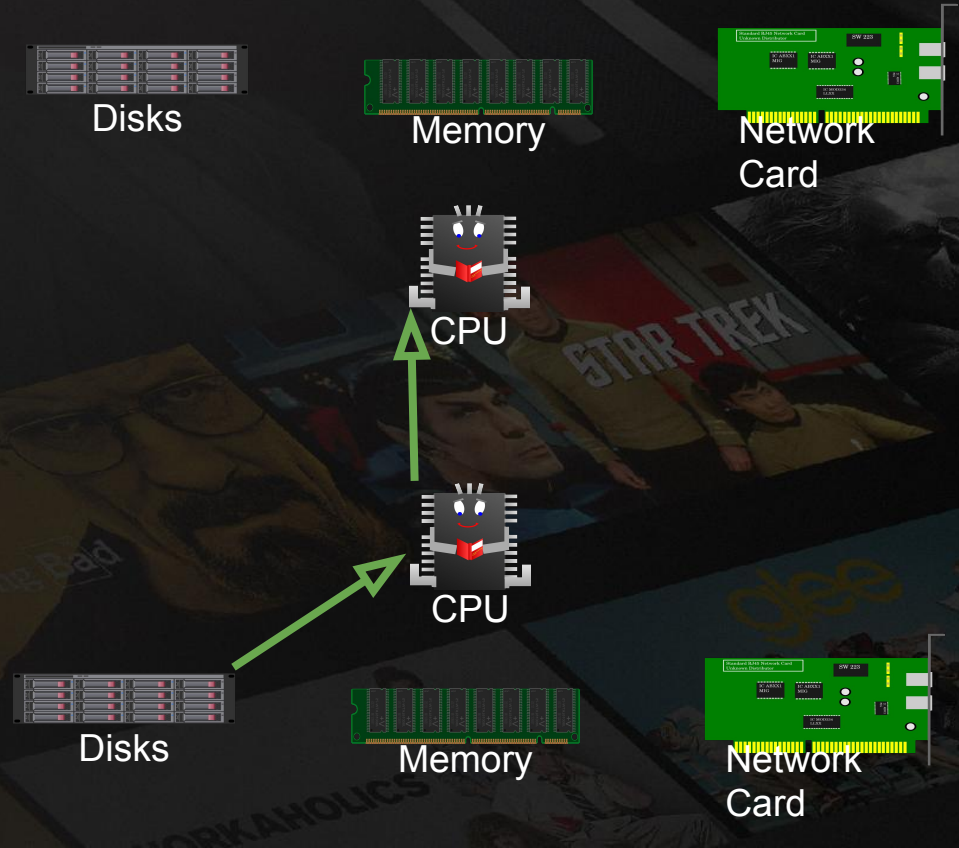
- DMA data from disk to memory



Dual Xeon: Worst Case Data Flow

Steps to send data:

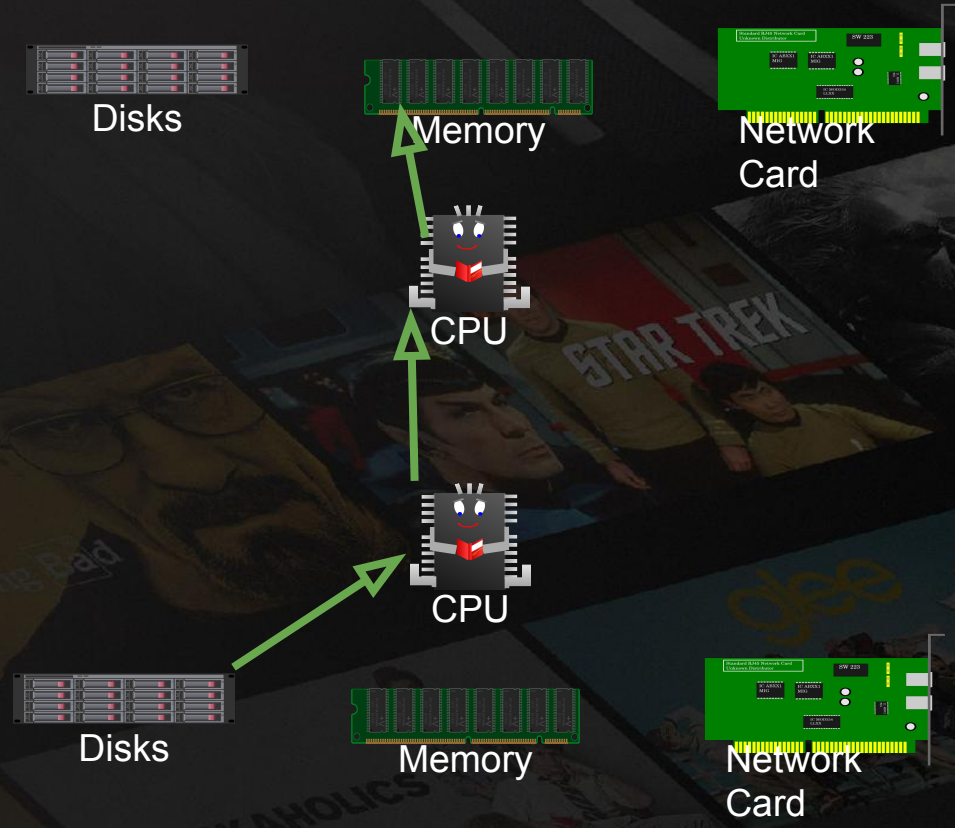
- DMA data from disk to memory
  - First NUMA bus crossing



Dual Xeon: Worst Case Data Flow

Steps to send data:

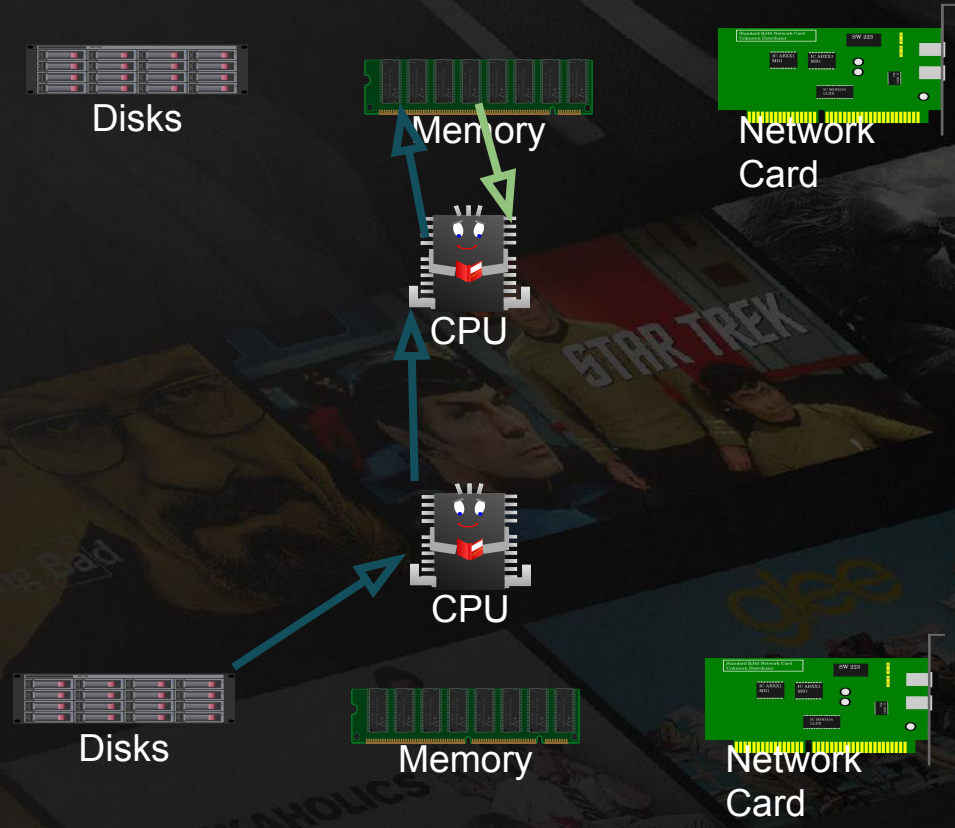
- DMA data from disk to memory
  - First NUMA bus crossing



Dual Xeon: Worst Case Data Flow

Steps to send data:

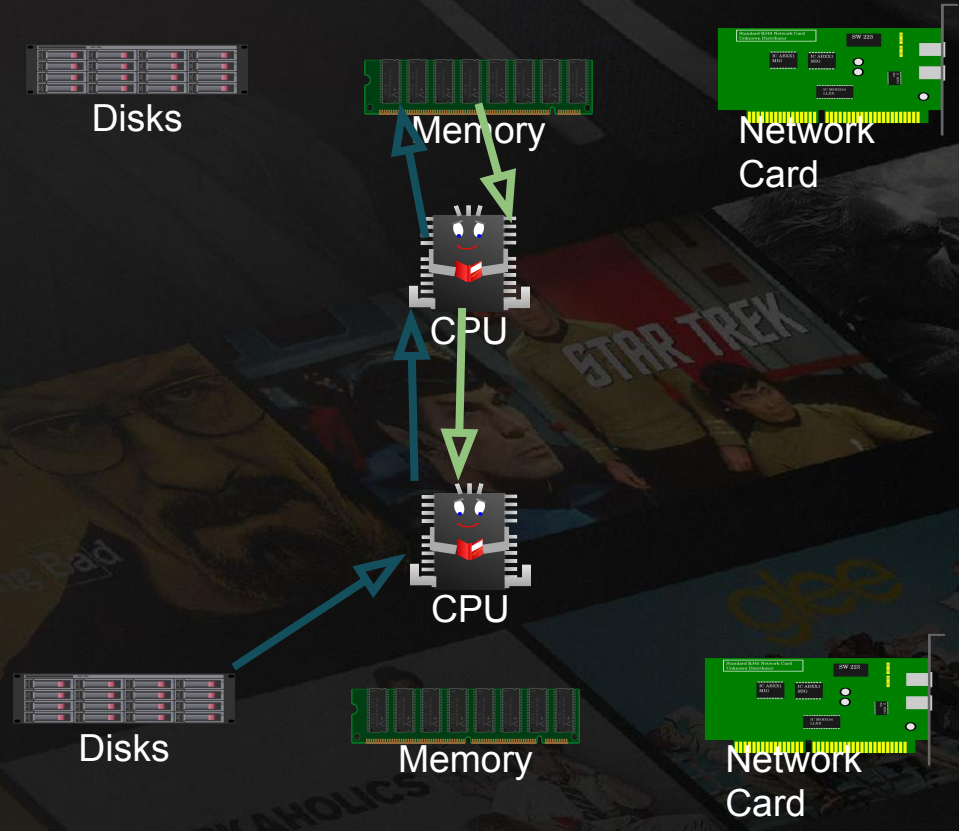
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption



Dual Xeon: Worst Case Data Flow

Steps to send data:

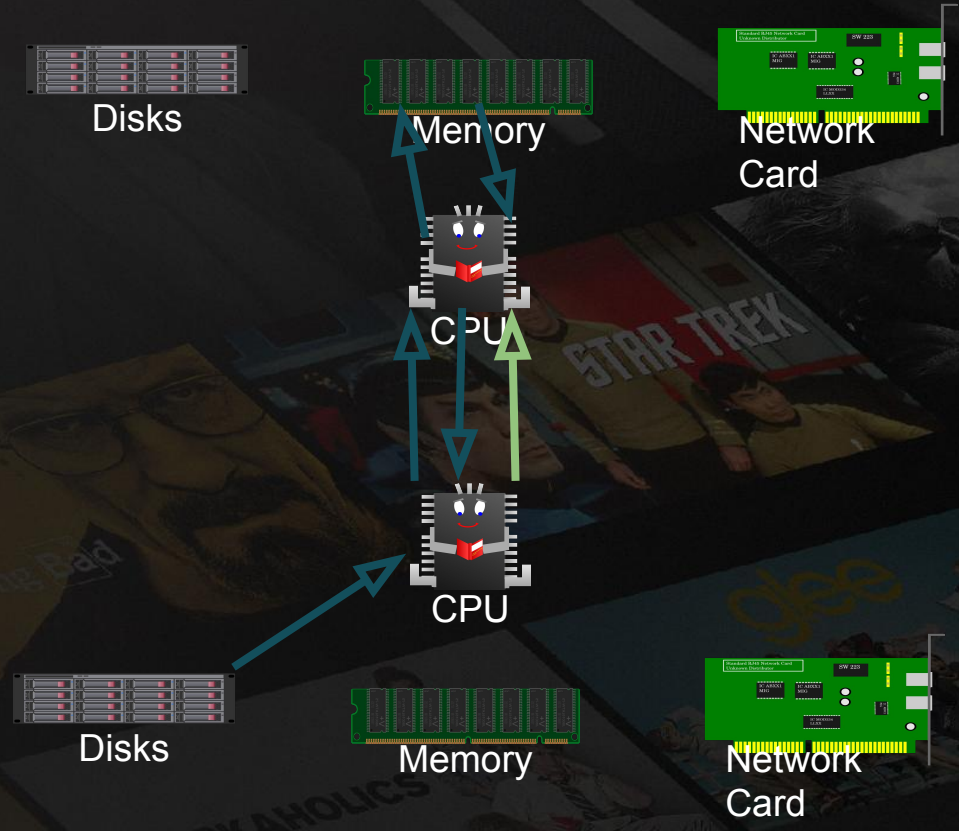
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption
  - Second NUMA crossing



Dual Xeon: Worst Case Data Flow

Steps to send data:

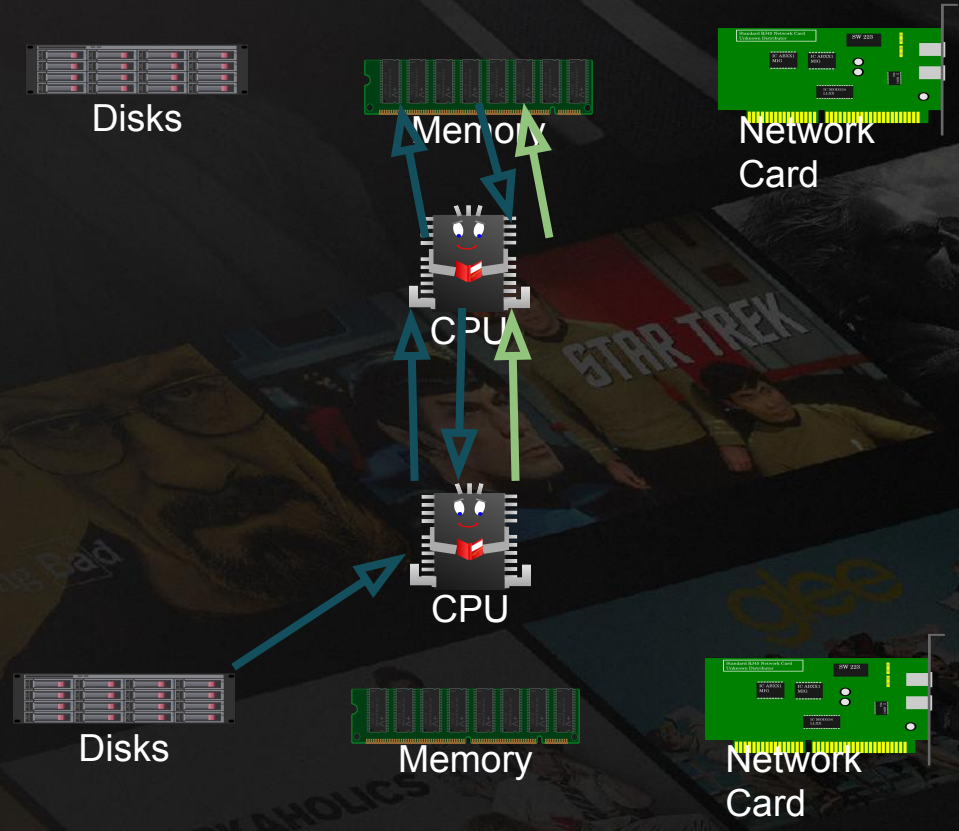
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption
  - Second NUMA crossing
- CPU writes encrypted data
  - Third NUMA crossing



Dual Xeon: Worst Case Data Flow

Steps to send data:

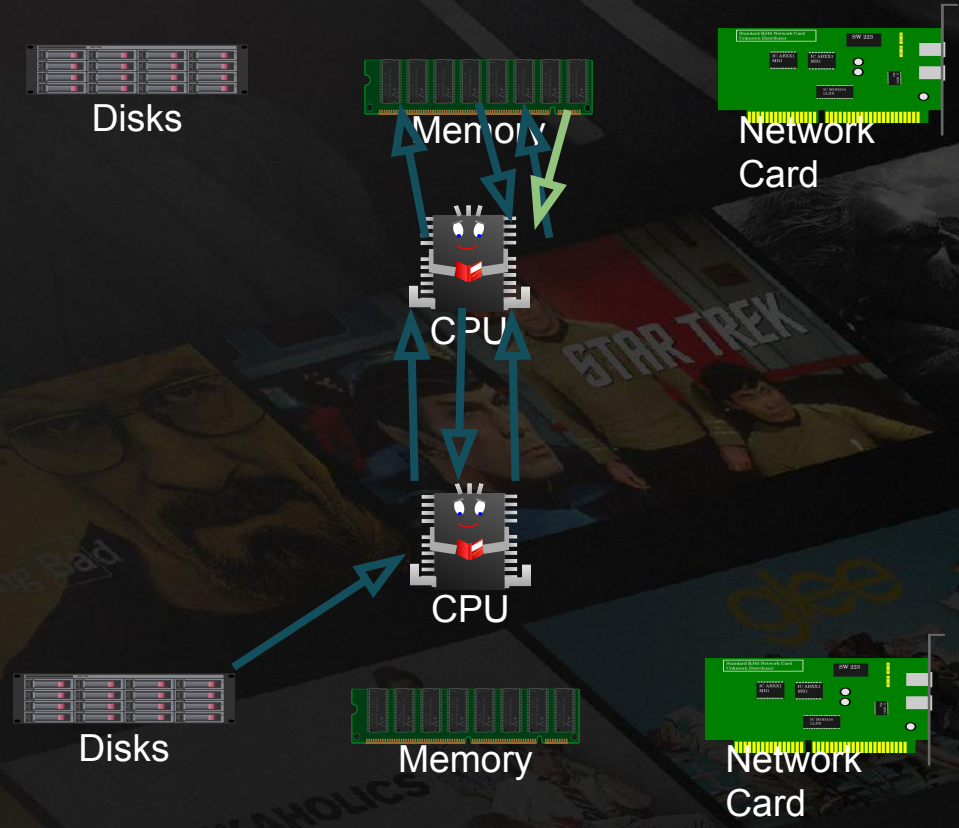
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption
  - Second NUMA crossing
- CPU writes encrypted data
  - Third NUMA crossing



Dual Xeon: Worst Case Data Flow

Steps to send data:

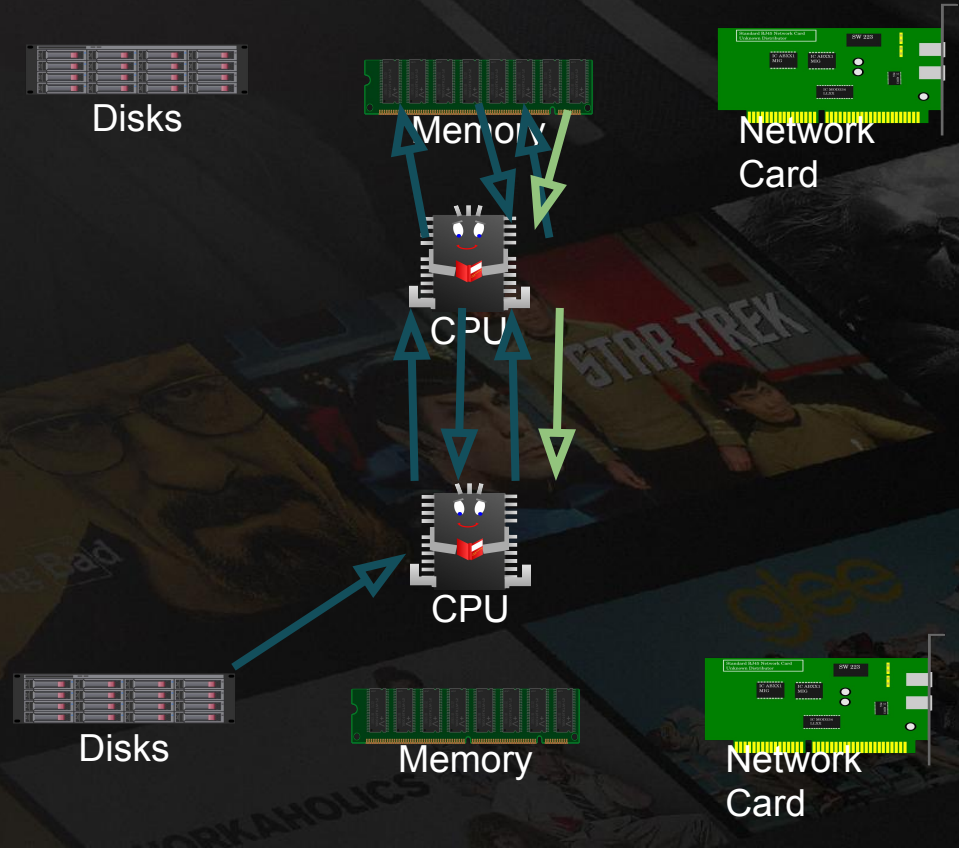
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption
  - Second NUMA crossing
- CPU writes encrypted data
  - Third NUMA crossing
- DMA from memory to Network



Dual Xeon: Worst Case Data Flow

Steps to send data:

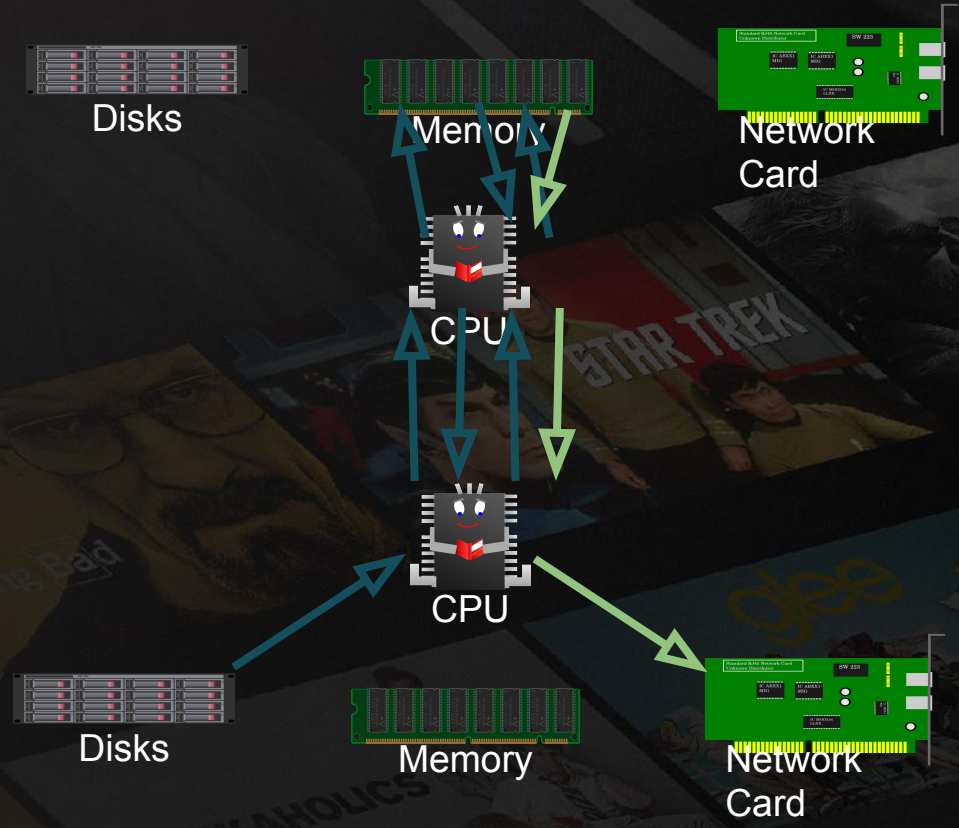
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption
  - Second NUMA crossing
- CPU writes encrypted data
  - Third NUMA crossing
- DMA from memory to Network
  - Fourth NUMA crossing



Dual Xeon: Worst Case Data Flow

Steps to send data:

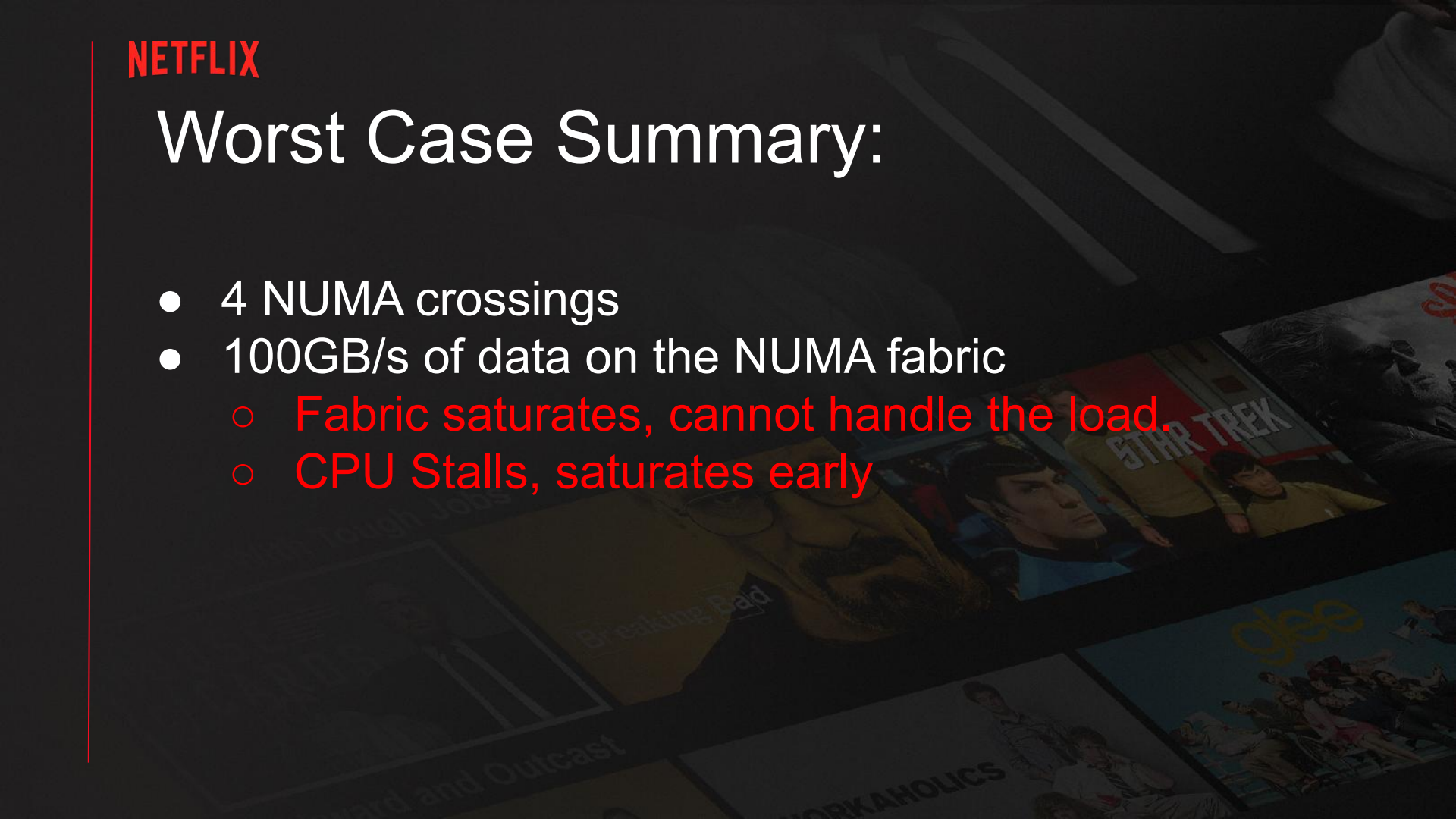
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU reads data for encryption
  - Second NUMA crossing
- CPU writes encrypted data
  - Third NUMA crossing
- DMA from memory to Network
  - Fourth NUMA crossing



NETFLIX

Worst Case Summary:

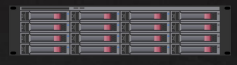
- 4 NUMA crossings
- 100GB/s of data on the NUMA fabric
  - Fabric saturates, cannot handle the load.
  - CPU Stalls, saturates early



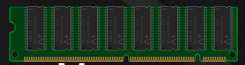
NETFLIX Dual Xeon: Best Case Data Flow

Steps to send data:

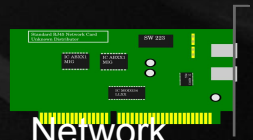
- DMA data from disk to memory



Disks



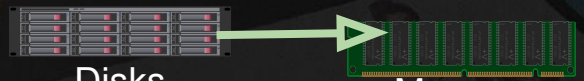
Memory



Network Card

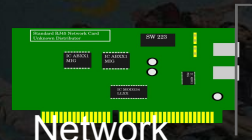
CPU

CPU



Disks

Memory

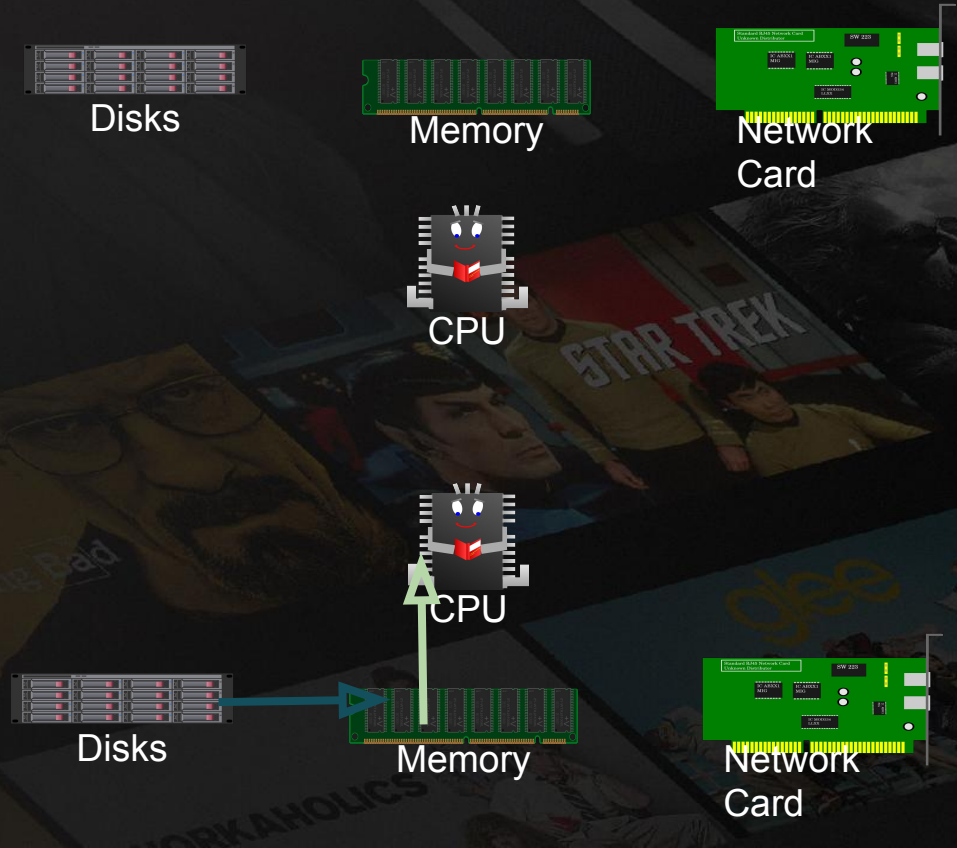


Network Card

NETFLIX Dual Xeon: Best Case Data Flow

Steps to send data:

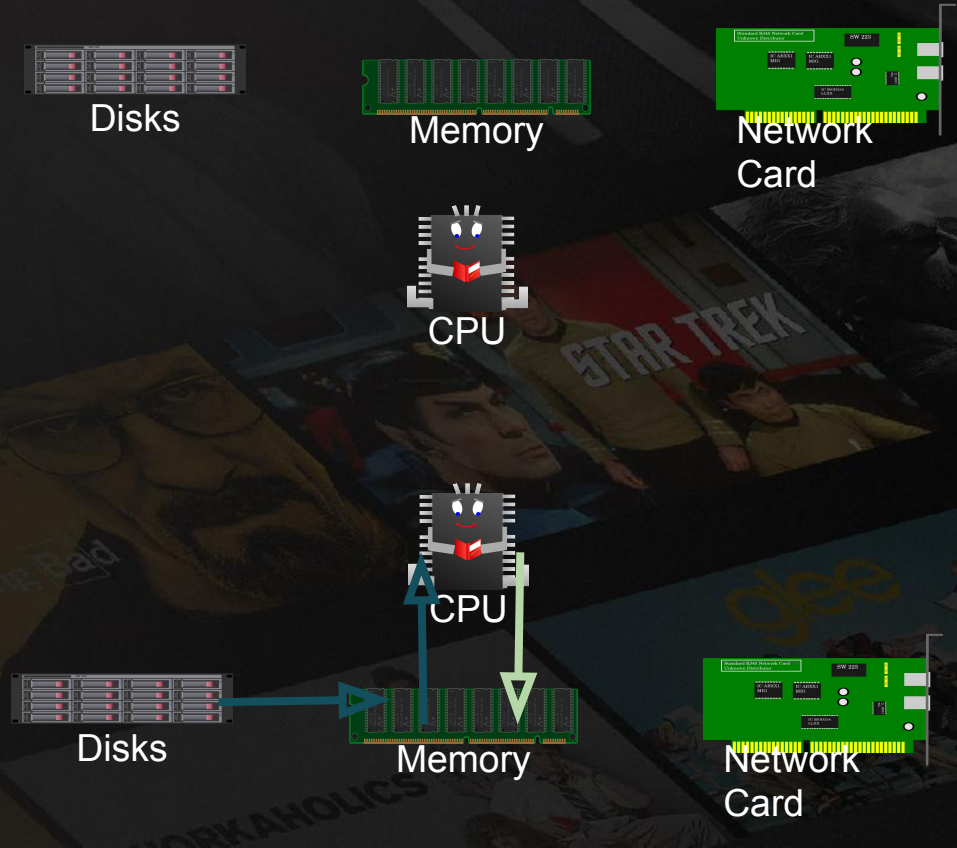
- DMA data from disk to memory
- CPU Reads data for encryption



Dual Xeon: Best Case Data Flow

Steps to send data:

- DMA data from disk to memory
- CPU Reads data for encryption
- CPU Writes encrypted data

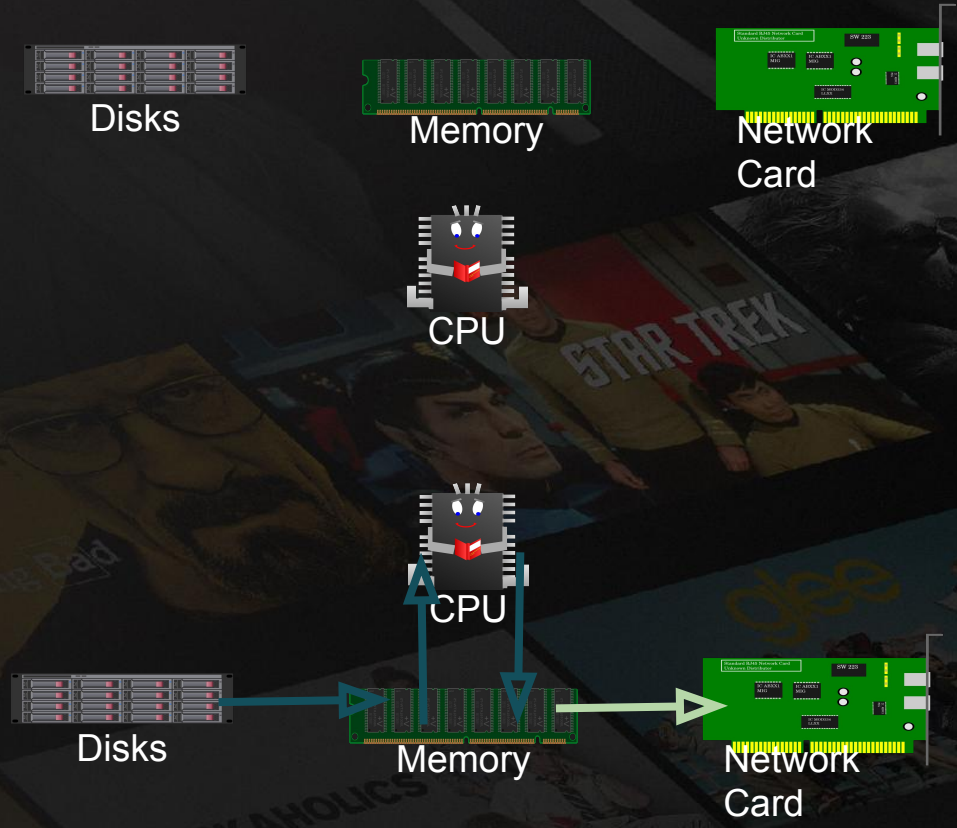


NETFLIX Dual Xeon: Best Case Data Flow

Steps to send data:

- DMA data from disk to memory
- CPU Reads data for encryption
- CPU Writes encrypted data
- DMA from memory to Network

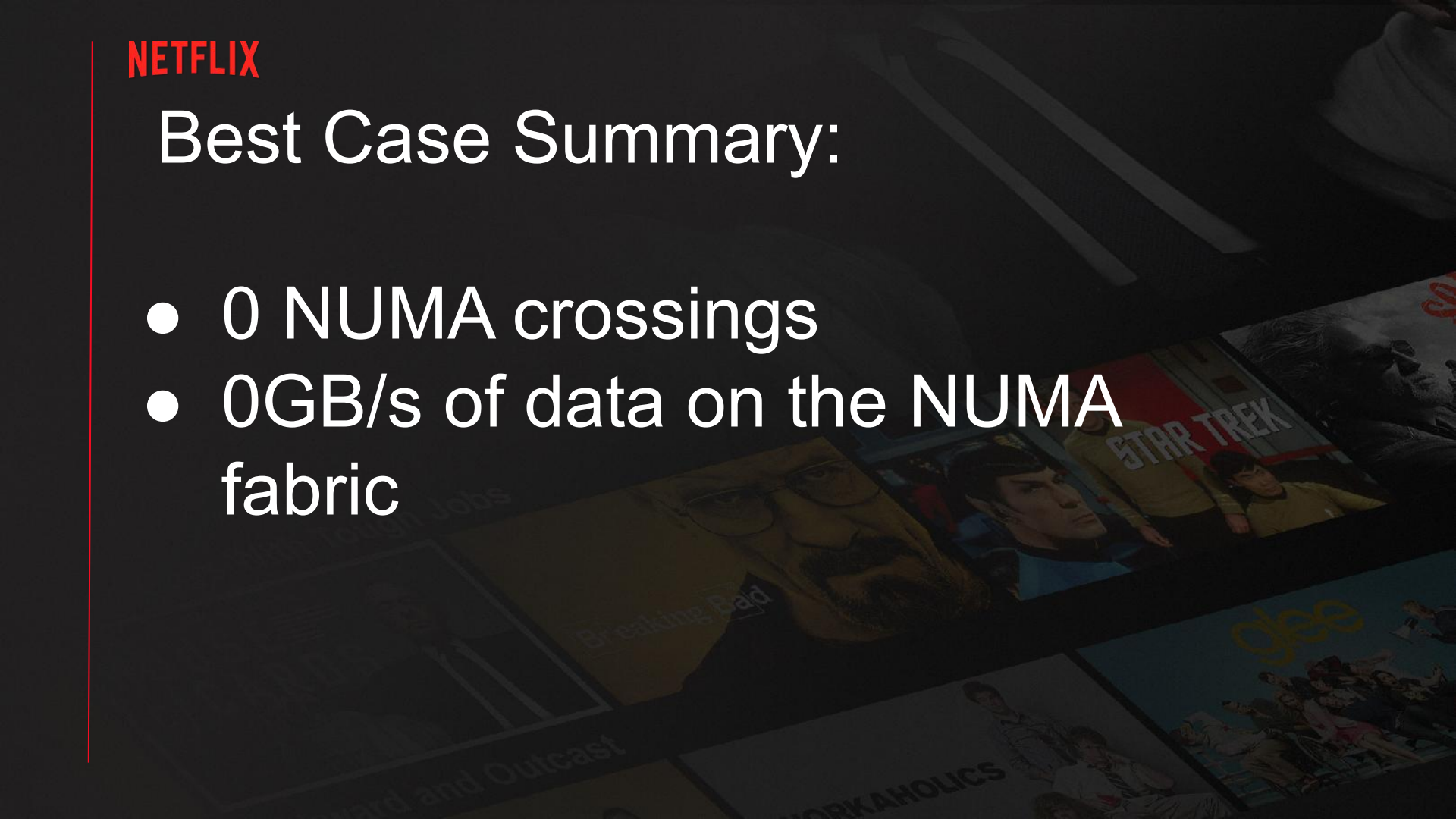
0 NUMA crossings!



NETFLIX

Best Case Summary:

- 0 NUMA crossings
- 0GB/s of data on the NUMA fabric



NETFLIX

How can we get as close as possible to the best case?

1 bhyve VM per NUMA Node, passing through NIC and disks?

- Doubles IPv4 address use
- More than 2x AWS cloud management overhead
  - Managing one physical & two virtual machines
- non-starter

NETFLIX

How can we get as close as possible to the best case?

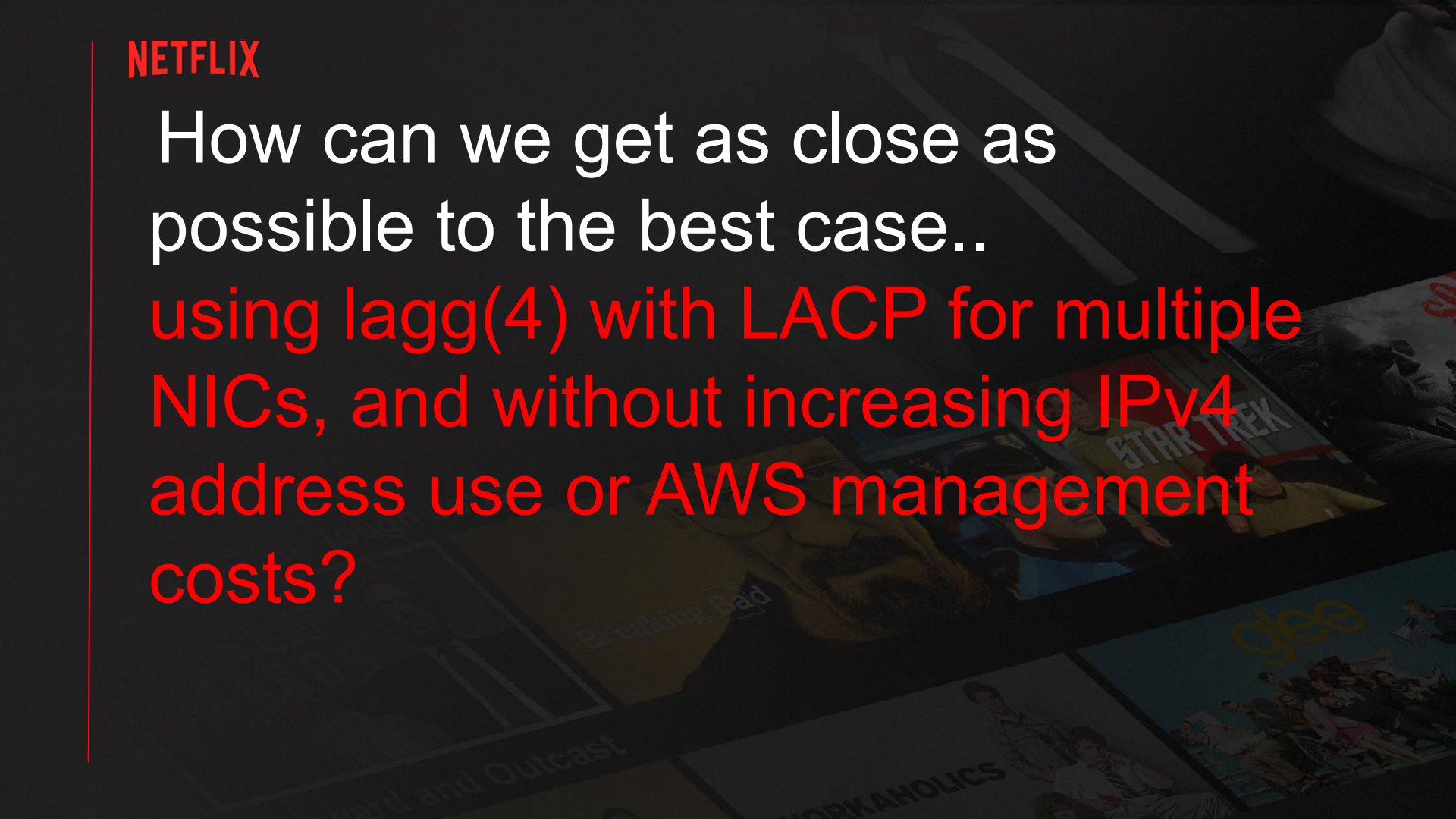
Content aware steering using multiple IP addresses?

- Doubles IPv4 address use
- Increases AWS cloud management overhead
- non-starter

NETFLIX

How can we get as close as possible to the best case..

using lagg(4) with LACP for multiple NICs, and without increasing IPv4 address use or AWS management costs?



NETFLIX

Impose order on the chaos.. *somehow*:

- Disk centric siloing
  - Try to do everything on the NUMA node where the content is stored
- Network centric siloing
  - Try to do as much as we can on the NUMA node that the LACP partner chose for us

NETFLIX

Disk centric siloing

- Associate disk controllers with NUMA nodes
- Associate NUMA affinity with files
- Associate network connections with NUMA nodes
- Move connections to be “close” to the disk where the contents file is stored.
- After the connection is moved, there will be 0 NUMA crossings!

NETFLIX

Disk centric siloing problems

- No way to tell link partner that we want LACP to direct traffic to a different switch/router port
  - So TCP acks and http requests will come in on the “wrong” port
- Moving connections can lead to TCP re-ordering due to using multiple egress NICs
- Some clients issue http GET requests for different content on the same TCP connection
  - Content may be on different NUMA domains!

NETFLIX

Network centric siloing

- Associate network connections with NUMA nodes
- Allocate local memory to back media files when they are DMA'ed from disk
- Allocate local memory for TLS crypto destination buffers & do SW crypto locally
- Run RACK / BBR TCP pacers with domain affinity
- Choose local lagg(4) egress port

NETFLIX

Associate network connections with NUMA nodes:

- Add a NUMA domain field to struct mbuf
  - r346281
- Embed device's NUMA domain struct ifnet
  - r346579
- Drivers tag received mbufs w/NUMA domain
  - r346677
- Add NUMA domain to struct inpcb
  - r346677

NETFLIX

Associate network connections with NUMA nodes (continued):

- Record NUMA domain into struct inpcb when TCP connection is “born”
  - r346677
- Ensure that a connection is given to an nginx worker bound to the correct domain
  - More on this later

NETFLIX

Allocate NUMA local memory to back video files

- Surprisingly easy.. Just run nginx with worker_cpu_affinity set to auto
- Default first-touch policy will cause the VM system to allocate pages on the same node as the nginx worker
- Thanks to kib@ and alc@ for pointing out a large patch that I had to sendfile() and vm_page_alloc() was not needed

NETFLIX

Allocate local memory for TLS crypto destination buffers & do SW crypto locally

- kTLS worker threads are run with domain affinity
- kTLS worker threads have a domain allocation policy to prefer the local NUMA domain
  - This ensures crypto destination buffers are allocated on the local NUMA domain
  - D21648

NETFLIX

How to choose local lagg(4) egress port?

- Outgoing mbufs tagged with NUMA domain of TCP connection
  - r346677
- Lagg ports are organized into a hierarchical model, where we limit our choice of NICs to the set of NICs on the desired NUMA domain. (when ifconfig lagg0 use_numa is set)
  - r347055

NETFLIX

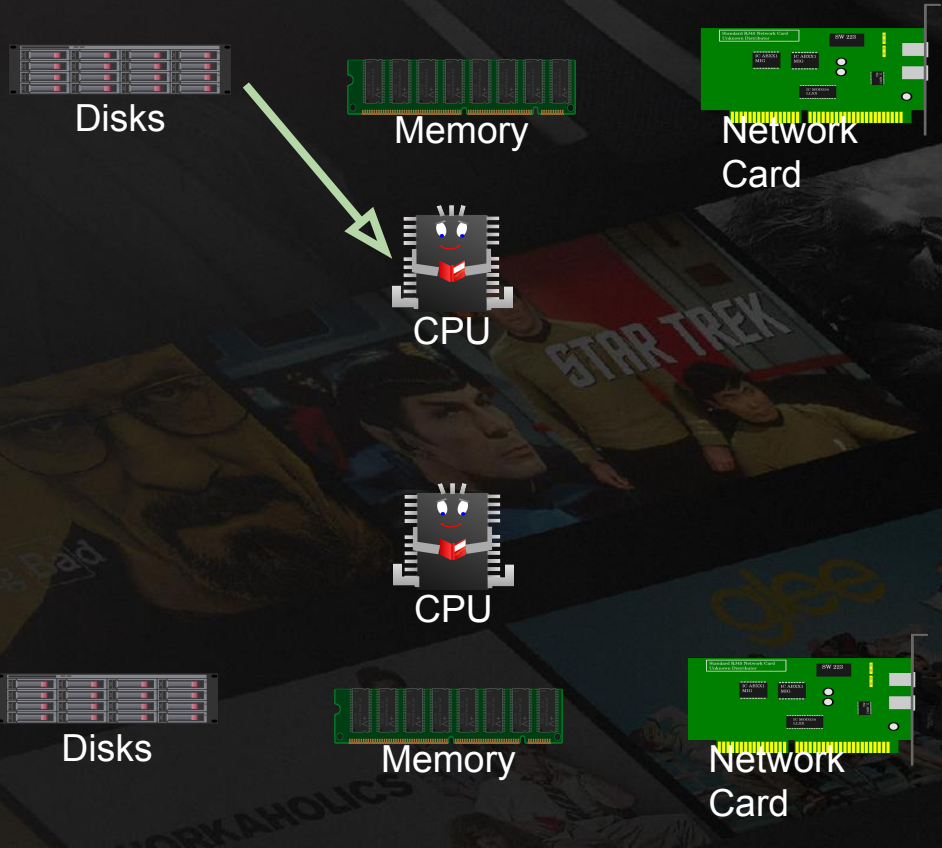
How to choose the correct nginx worker?

- Augment SO_REUSEPORT to make a new TCP_REUSEPORT_NUMA socket option
  - SO_REUSEPORT allows multiple threads / processes to share a listen socket
  - TCP_REUSEPORT_NUMA causes incoming connections to be filtered to only listen sockets on the same domain (with a fallback if there are no listeners on the same domain)
    - D21636

Dual Xeon: Worst Case Data Flow with NUMA siloing

Steps to send data:

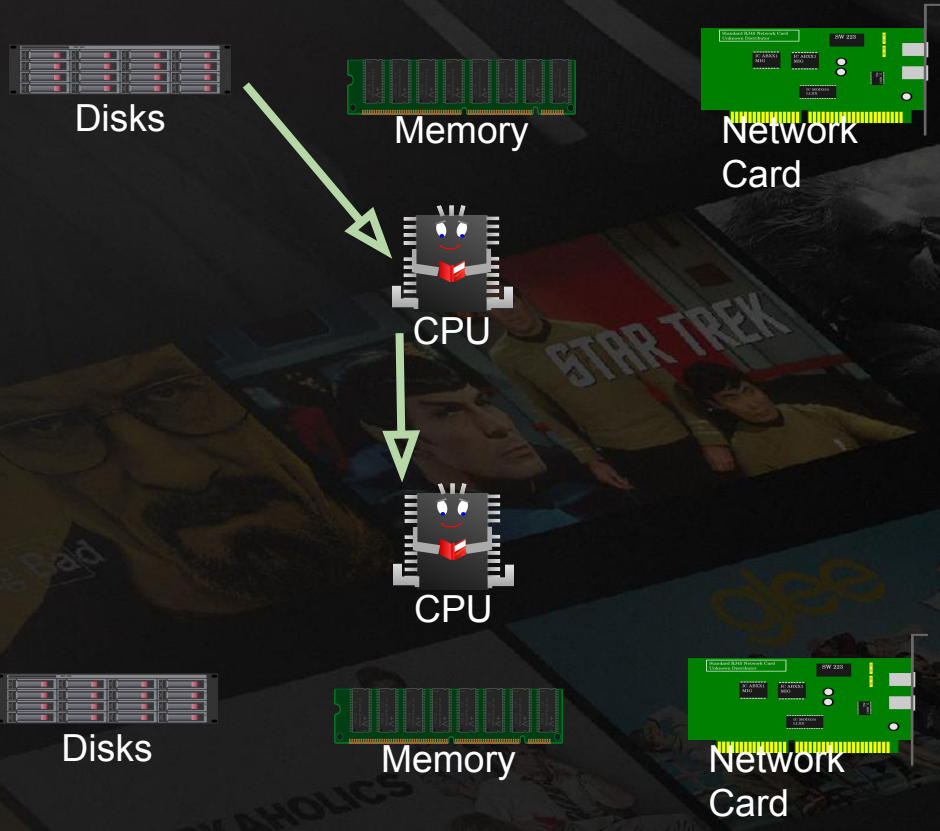
- DMA data from disk to memory



Dual Xeon: Worst Case Data Flow with NUMA siloing

Steps to send data:

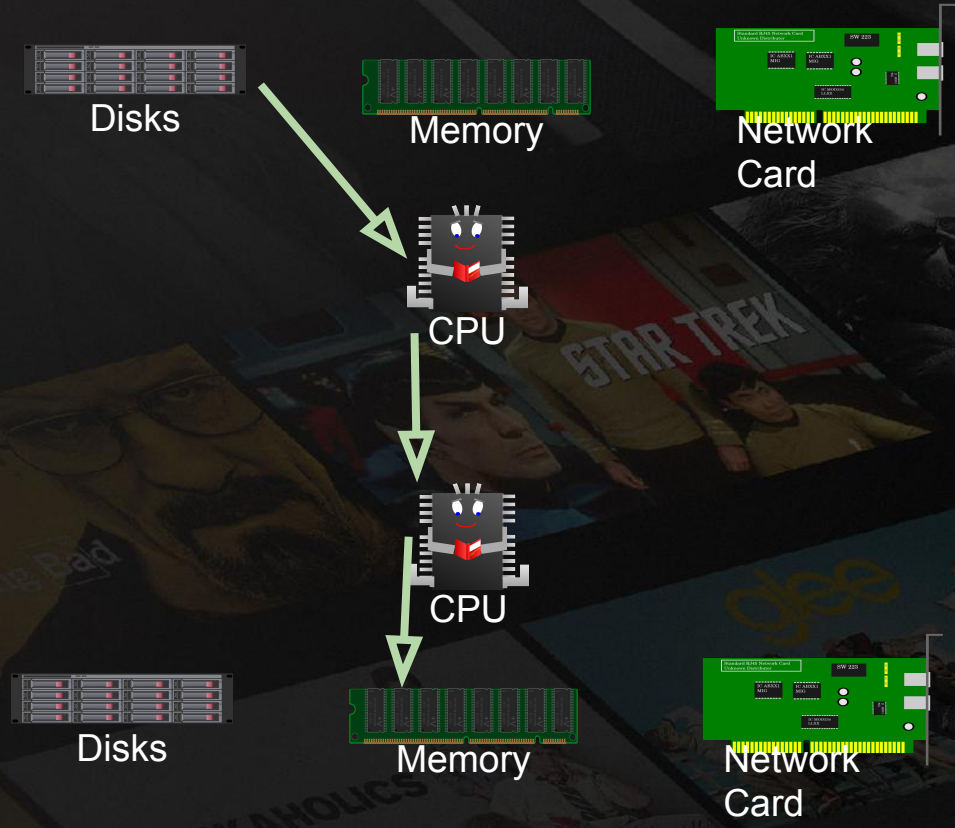
- DMA data from disk to memory
  - First NUMA bus Crossing



Dual Xeon: Worst Case Data Flow with NUMA siloing

Steps to send data:

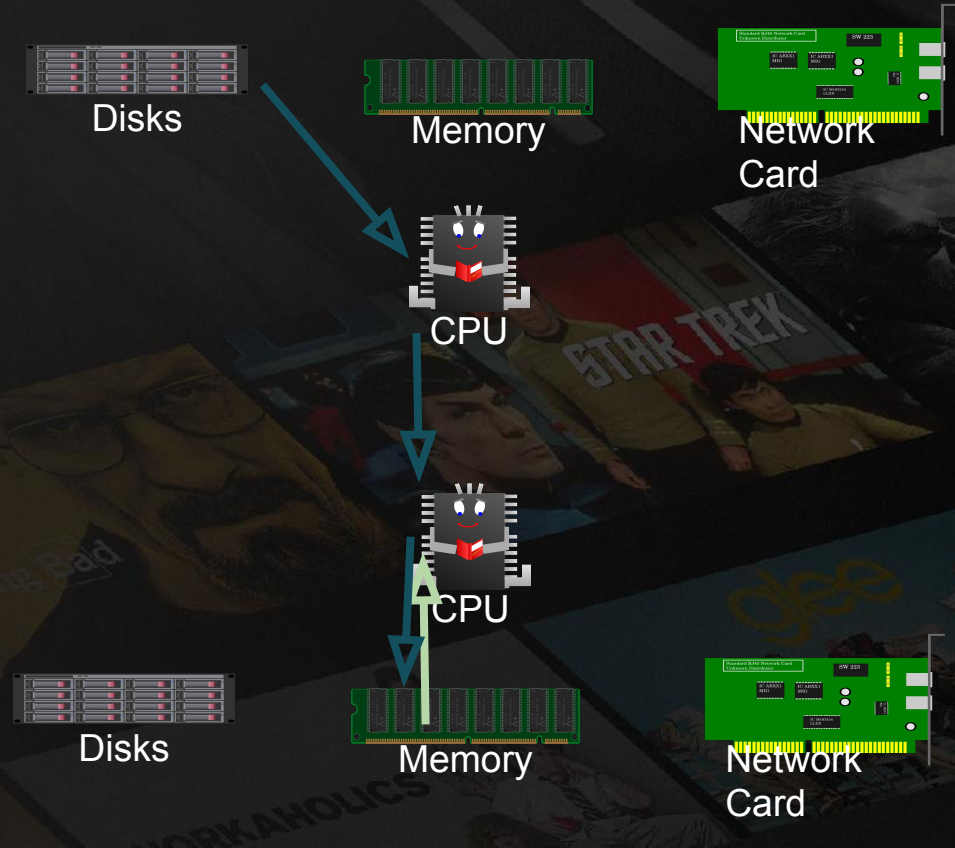
- DMA data from disk to memory
  - First NUMA bus Crossing



Dual Xeon: Worst Case Data Flow with NUMA siloing

Steps to send data:

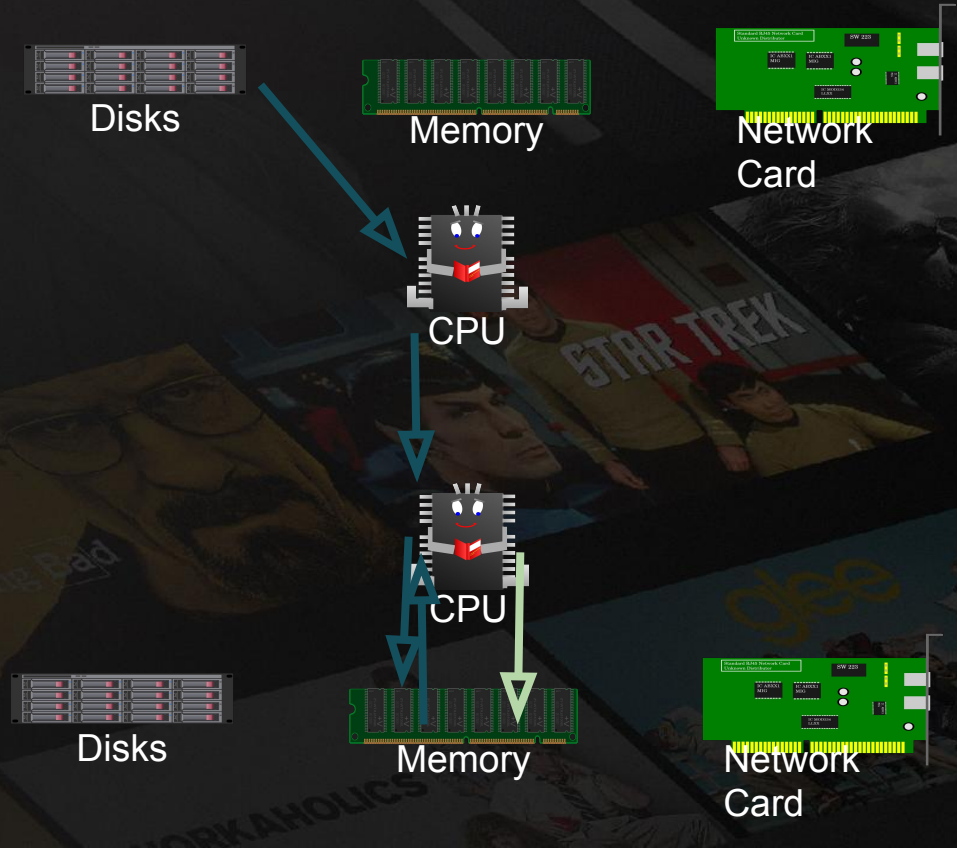
- DMA data from disk to memory
  - First NUMA bus crossing
- CPU Reads data for encryption



Dual Xeon: Worst Case Data Flow with NUMA siloing

Steps to send data:

- DMA data from disk to memory
  - First NUMA bus crossing
- CPU Reads data for encryption
- CPU Writes encrypted data

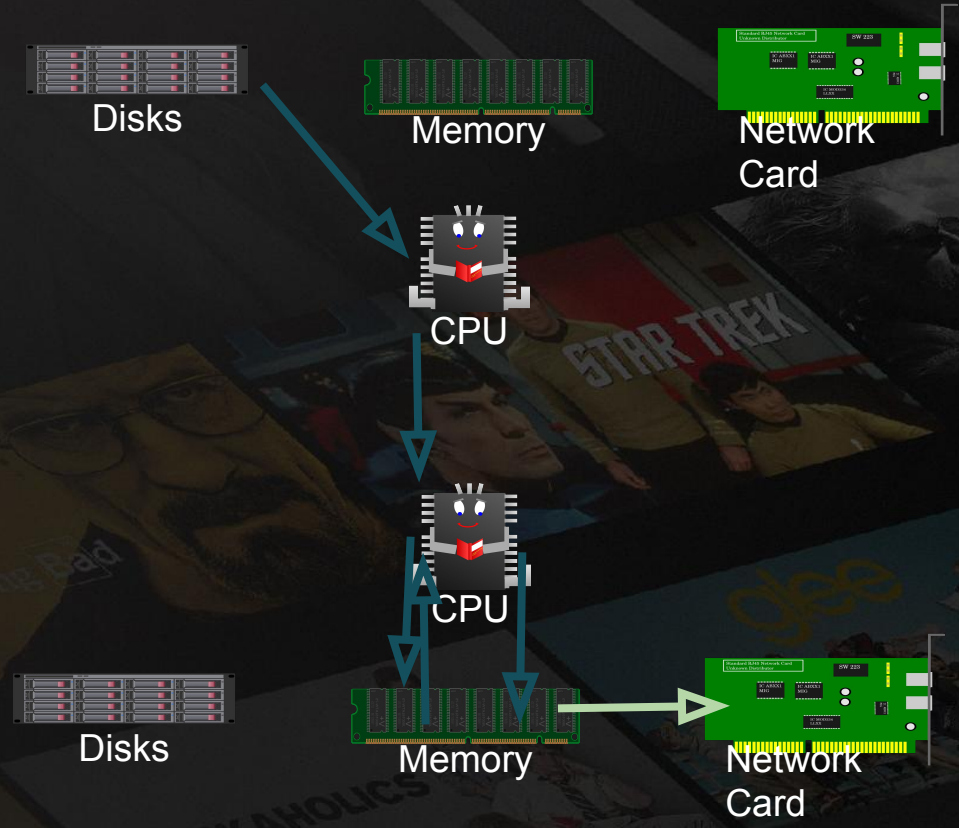


Dual Xeon: Worst Case Data Flow with NUMA siloing

Steps to send data:

- DMA data from disk to memory
  - First NUMA bus crossing
- CPU Reads data for encryption
- CPU Writes encrypted data
- DMA from memory to Network

1 NUMA bus crossing!



NETFLIX

Worst Case Summary:

- 1 NUMA crossing on average
  - 100% of disk reads across NUMA
- 25GB/s of data on the NUMA fabric
  - Still much less than 40GB/sec fabric bandwidth

NETFLIX

Average Case Summary Xeon (2 NUMA nodes):

- 0.5 NUMA crossings on average
  - 50% of disk reads across NUMA
- 12.5GB/s of data on the NUMA fabric
  - CPU does not saturate, we exceed 190Gb/s

NETFLIX

Average Case Summary EPYC (4 NUMA nodes):

- 0.75 NUMA crossings on average
  - 75% of disk reads across NUMA
- 18.75GB/s of data on the NUMA fabric
  - CPU does not saturate, we exceed 190Gb/s

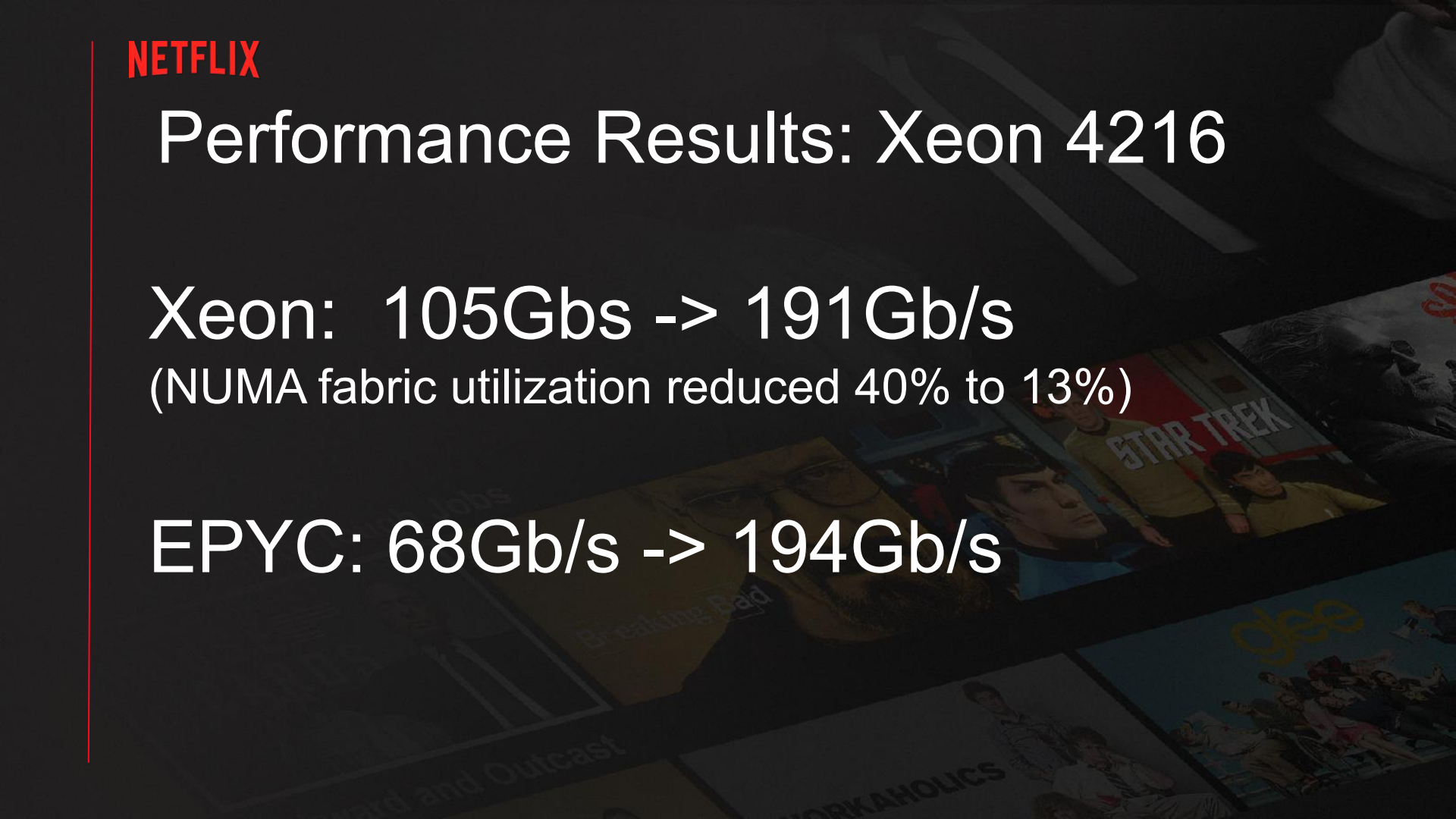
NETFLIX

Performance Results: Xeon 4216

Xeon: 105Gbs -> 191Gb/s

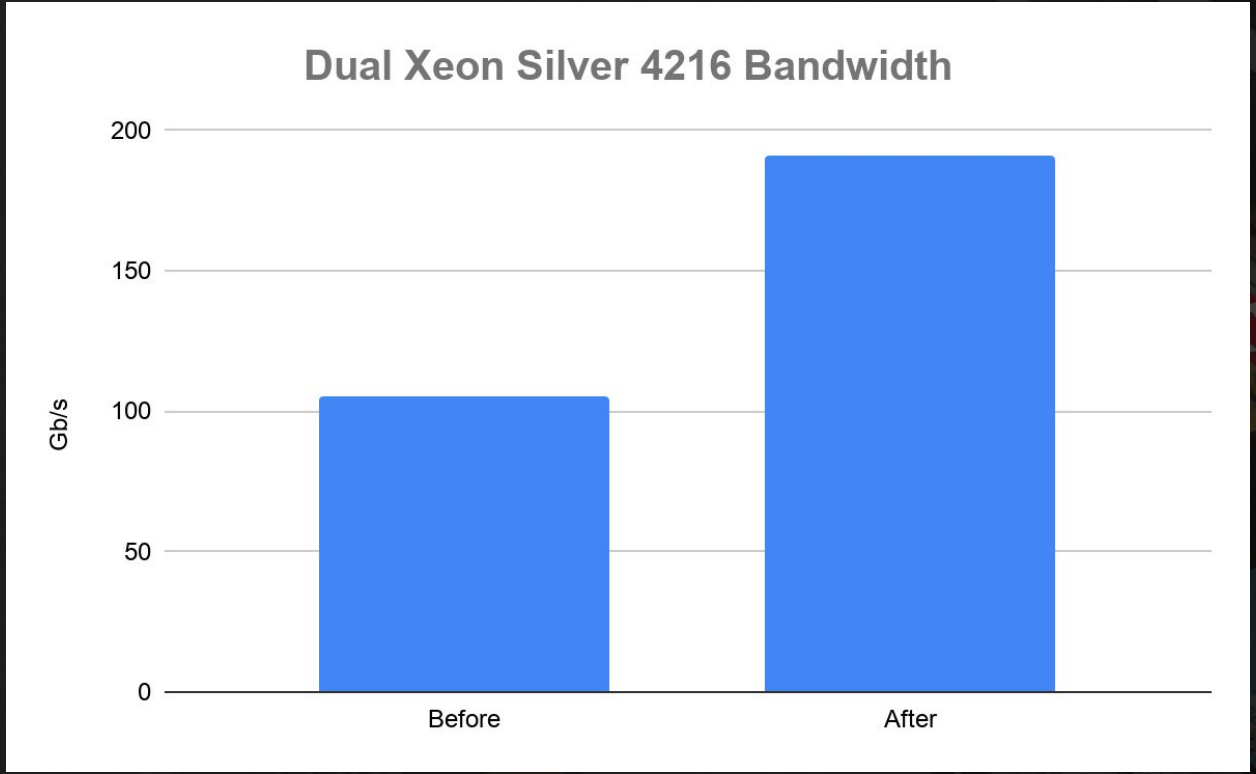
(NUMA fabric utilization reduced 40% to 13%)

EPYC: 68Gb/s -> 194Gb/s



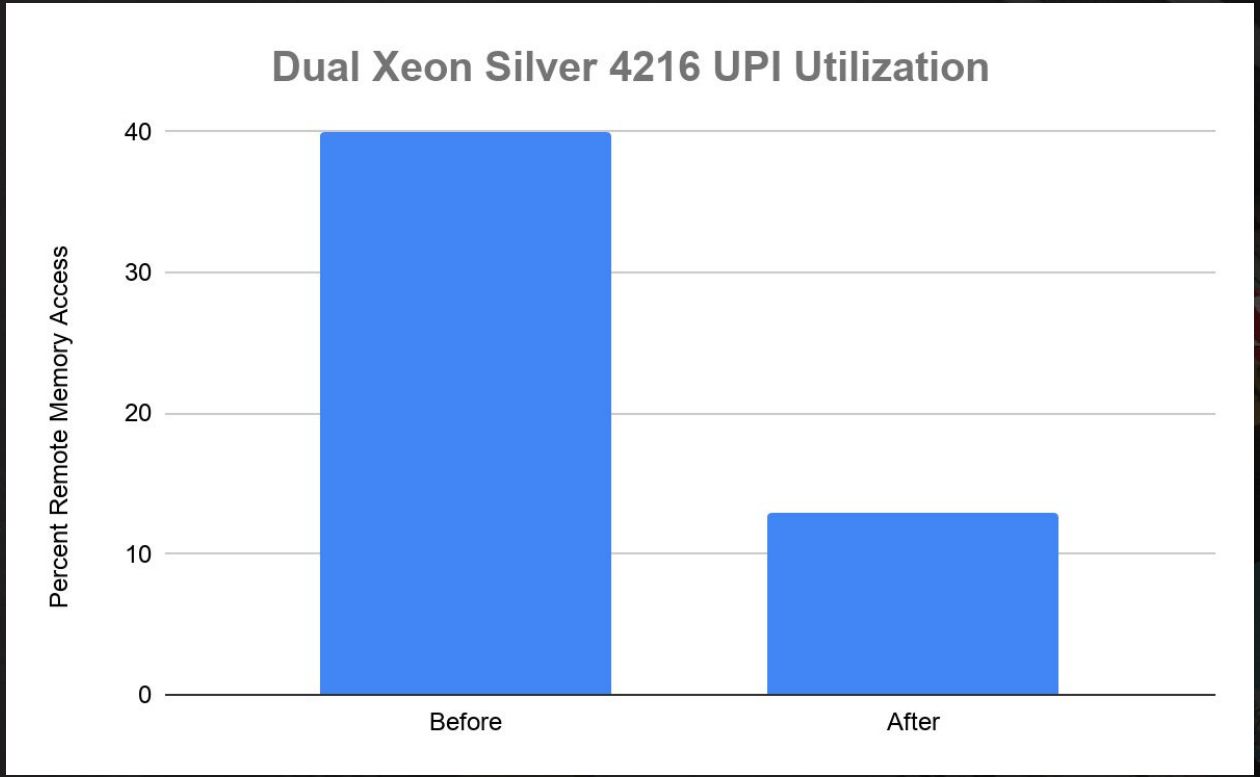
NETFLIX

Performance Results: Intel



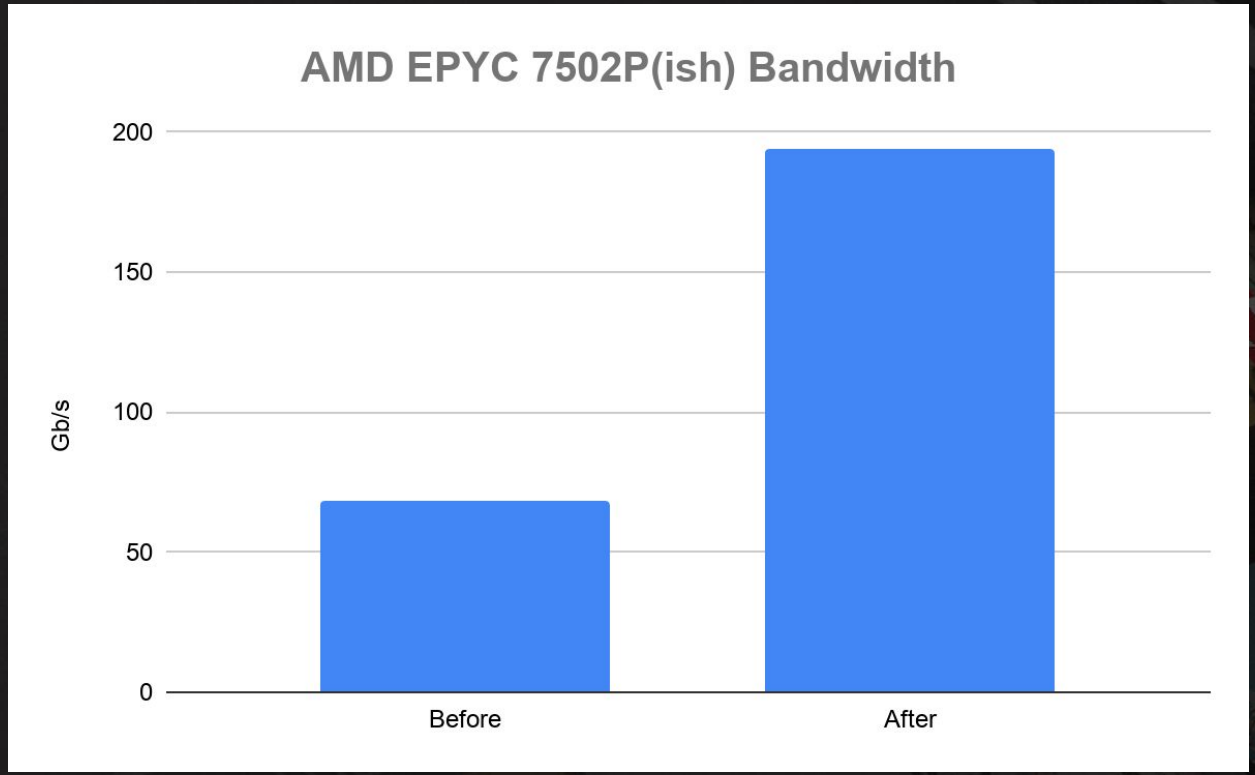
NETFLIX

Performance Results: Intel



NETFLIX

Performance Results: AMD



NETFLIX

Actual data from pcm.x @105Gb/s

```
Intel(r) UPI data traffic estimation in bytes (data traffic coming to CPU/socket through UPI links):

               UPI0    UPI1    |  UPI0   UPI1
-------------------------------------------------------------------------------------------------------------
---
 SKT    0       26 G    26 G   |   40% 40%
 SKT    1       28 G    28 G   |   42% 42%
-------------------------------------------------------------------------------------------------------------
---
Total UPI incoming data traffic:  109 G      UPI data traffic/[illegible]ntroller traffic: [illegible].40

Intel(r) UPI traffic estimation in bytes (data and non-data traffic outgoing from CPU/socket through UPI
links):

               UPI0    UPI1    |  UPI0   UPI1
-------------------------------------------------------------------------------------------------------------
---
 SKT    0       47 G    47 G   |   73% 73%
 SKT    1       46 G    46 G   |   69% 69%
-------------------------------------------------------------------------------------------------------------
---
Total UPI outgoing data and non-data traffic:  188 G
MEM (GB)->|  READ |  WRITE | PMM RD | PMM WR | CPU energy | DIMM energy |
-------------------------------------------------------------------------------------------------------------
---
 SKT    0       122.34  136.19         0.00    0.00    272.51         58.37
 SKT    1       8.24    6.99    0.00   0.00    231.16         28.63
-------------------------------------------------------------------------------------------------------------
---
       *       130.57  143.18         0.00    0.00    503.68         87.01
```

NETFLIX

Actual data from nstat (Xeon)

```
c333.sjc002.dev# nstat 15
 InMpps OMpps  InGbs  OGbs err TCP Est %CPU syscalls csw    irq GBfree
  1.43   8.32   1.19  99.41  0  74690   62.96 176262 696238 157743 114.11
  1.50   8.69   1.22 103.86  0  76091   63.99 171871 698812 158257 113.44
  1.48   8.48   1.14 101.31  0  77022   64.09 178515 701432 157910 112.74
  1.49   8.56   1.17 102.22  0  78057   65.17 173108 696963 157307 111.95
  1.49   8.61   1.12 102.91  0  78654   64.65 169636 700402 157492 111.41
  1.51   8.81   1.09 105.27  0  79617   73.67 167101 683410 155796 111.05
  1.52   8.80   1.18 105.14  0  80721   69.15 173158 694059 156457 109.98
  1.48   8.38   1.13 100.03  0  81477   65.05 182784 698356 157515 109.25
  1.51   8.61   1.13 102.88  0  82449   65.90 181321 701593 157383 108.98
  1.49   8.64   1.13 103.23  0  82404   69.43 187668 697208 156812 108.39
  1.47   8.59   1.12 102.69  0  79792   63.68 149610 676066 155931 108.10
  1.56   9.01   1.30 107.76  0  78127   69.11 138318 685625 156429 107.79
^C
c333.sjc002.dev# sysctl hw.model
hw.model: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
c333.sjc002.dev# sysctl hw.ncpu
hw.ncpu: 64
```

NETFLIX

Actual data from pcm.x @191Gb/s

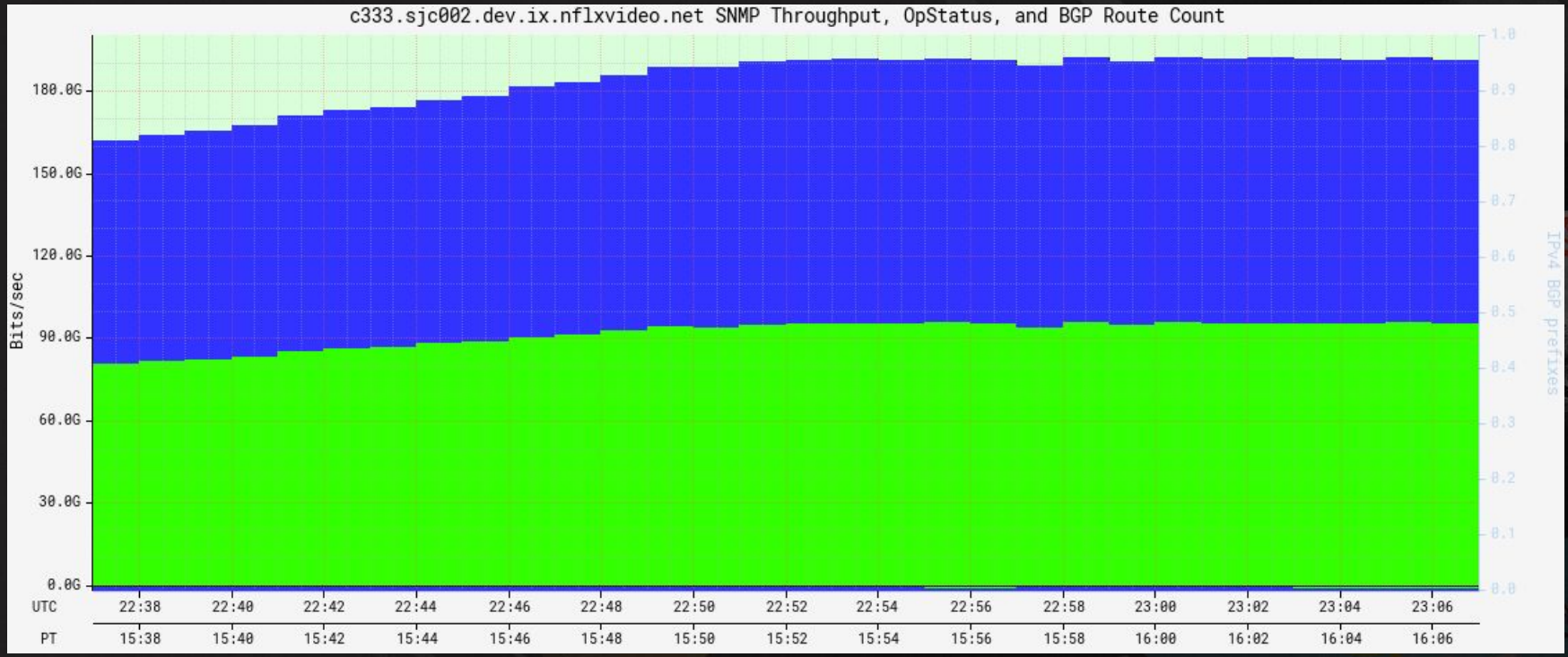
```
Intel(r) UPI data traffic estimation in bytes (data traffic coming to CPU/socket through UPI links):

               UPI0    UPI1    |  UPI0   UPI1
-------------------------------------------------------------------------------------------------------------
---
 SKT    0       3926 M   3930 M   |   18%     18%
 SKT    1       3958 M   3954 M   |   18%     18%
-------------------------------------------------------------------------------------------------------------
---
Total UPI incoming data traffic:   15 G      UPI data traffic/[illegible]ntroller traffic: [illegible].13

Intel(r) UPI traffic estimation in bytes (data and non-data traffic outgoing from CPU/socket through UPI
links):

               UPI0    UPI1    |  UPI0   UPI1
-------------------------------------------------------------------------------------------------------------
---
 SKT    0       11 G    11 G   |   52% 53%
 SKT    1       11 G    11 G   |   52% 52%
-------------------------------------------------------------------------------------------------------------
---
Total UPI outgoing data and non-data traffic:   45 G
MEM (GB)->|  READ |  WRITE | PMM RD | PMM WR | CPU energy | DIMM energy |
-------------------------------------------------------------------------------------------------------------
---
 SKT    0       32.28   28.00   0.00    0.00    99.01   16.97
 SKT    1       32.77   28.18   0.00    0.00    100.75          18.31
-------------------------------------------------------------------------------------------------------------
---
       *       65.05   56.19   0.00    0.00    199.76          35.28
```

NETFLIX

Actual data from nstat (Xeon)

```
c333.sjc002.dev# nstat 15
 InMpps OMpps  InGbs  OGbs err TCP Est %CPU syscalls csw    irq GBfree
  2.78  15.90   1.89 190.33  0 150083   72.75 255685 1005743 167659  4.47
  2.81  15.94   1.97 190.83  0 150371   73.61 259173 1004664 167348  4.48
  2.80  15.93   1.88 190.67  0 150363   74.01 254862 997954 167199  5.09
  2.81  16.00   1.99 191.61  0 151127   73.54 257203 1000740 167202  5.21
  2.80  16.00   1.93 191.61  0 151216   74.38 257396 1001018 167243  4.58
  2.79  16.00   1.86 191.63  0 151559   73.53 256385 1001693 167205  4.60
  2.79  15.96   1.85 191.11  0 151548   74.65 256022 995630 166977  4.51
  2.82  16.00   1.93 191.62  0 152176   74.43 259680 1001880 166890  4.72
  2.83  16.00   1.87 191.61  0 152258   74.47 259494 1003018 166839  4.01
  2.84  16.00   1.88 191.62  0 152805   74.41 258864 1003171 166727  4.56
  2.85  15.98   1.89 191.47  0 153473   76.12 260823 995460 166377  3.84
  2.85  15.99   1.83 191.61  0 153864   74.90 259149 1003219 166484  4.44
  2.87  15.99   1.88 191.60  0 154081   76.02 261867 1001356 166192  3.80
  2.88  15.99   1.97 191.62  0 154421   75.73 262377 1003028 166250  4.76
^C
c333.sjc002.dev# sysctl hw.model
hw.model: Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
c333.sjc002.dev# sysctl hw.ncpu
hw.ncpu: 64
```

NETFLIX


c333.sjc002.dev.ix.nflxvideo.net SNMP Throughput, OpStatus, and BGP Route Count
Bits/sec
180.0G
150.0G
120.0G
90.0G
60.0G
30.0G
0.0G
IPv4 BGP prefixes
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
UTC
22:38
22:40
22:42
22:44
22:46
22:48
22:50
22:52
22:54
22:56
22:58
23:00
23:02
23:04
23:06
PT
15:38
15:40
15:42
15:44
15:46
15:48
15:50
15:52
15:54
15:56
15:58
16:00
16:02
16:04
16:06

NETFLIX

Actual data from nstat (EPYC)

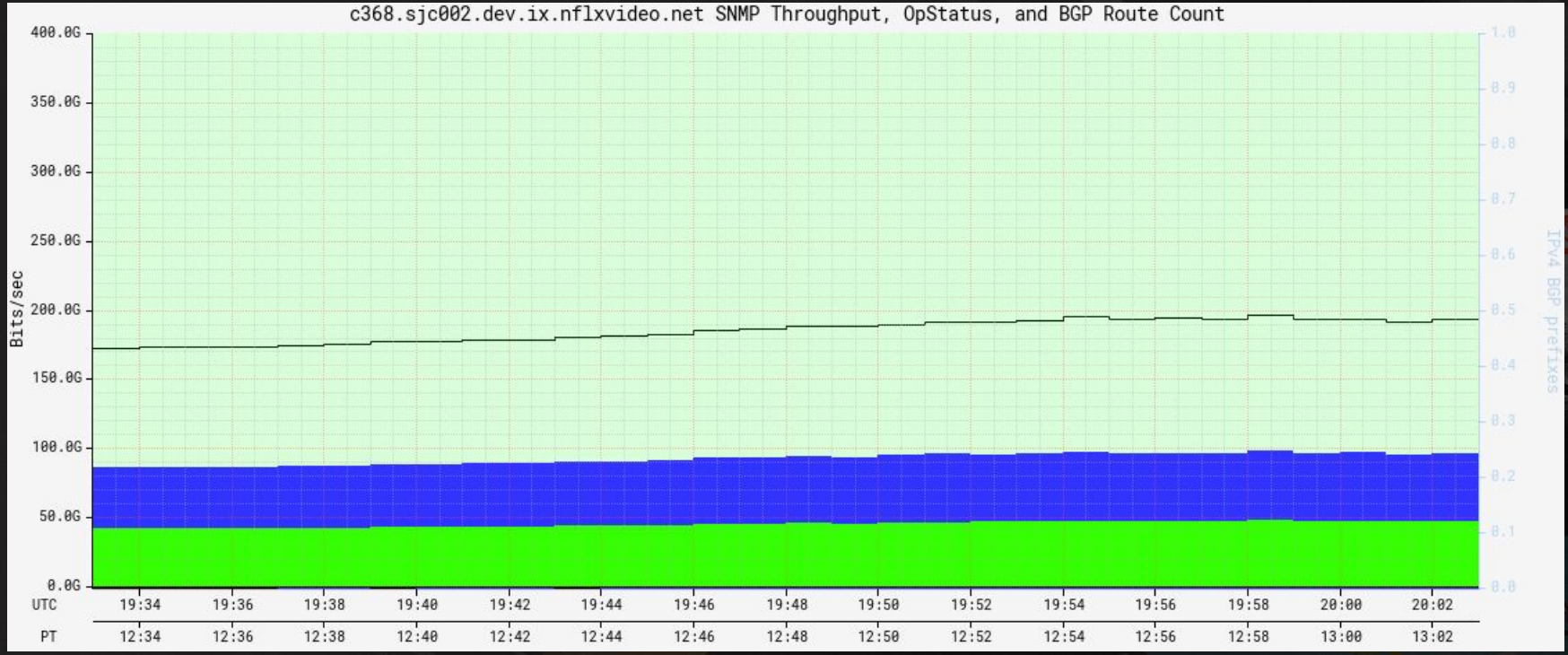
```
c368.sjc002.dev# nstat 15
  InMpps OMpps  InGbs  OGbs err TCP Est %CPU syscalls csw   irq GBfree5
  0.94   5.12   0.91  60.96  0  43687   30.53 151574 672314 100726 181.99
  0.97   5.26   0.80  62.68  0  45468   34.36 150603 666261  97171 178.94
  0.97   5.33   0.82  63.54  0  48308   47.74 147954 632247  90170 178.57
  0.99   5.49   0.75  65.49  0  50633   69.63 129971 565679  76805 178.57
  1.02   5.47   0.87  65.24  0  52431   73.24 127194 555176  74026 179.48
  1.02   5.45   0.78  64.97  0  53261   72.05 122047 553728  73717 178.82
  1.01   5.42   0.81  64.54  0  53534   72.13 118832 550171  74031 178.36
  1.01   5.47   0.77  65.25  0  53336   67.82 113505 561734  76411 178.58
  1.01   5.46   0.80  65.04  0  54342   66.57 113245 563114  77554 179.42
  1.01   5.43   0.73  64.66  0  55936   64.35 109110 565602  79069 178.87
  1.00   5.36   0.76  63.81  0  58181   62.75 106302 573547  82039 178.33
  0.99   5.39   0.63  64.32  0  55587   55.30  82142 565436  84060 180.00
  1.05   5.76   0.66  68.72  0  54378   53.83  80994 575478  85235 179.12
  0.92   5.03   0.57  60.06  0  53749   54.38  81413 563538  82983 178.60
  0.98   5.34   0.61  63.70  0  53911   53.30  81227 569083  84893 179.35
c368.sjc002.dev# sysctl hw.ncpu
hw.ncpu: 64
c368.sjc002.dev# sysctl hw.model
hw.model: AMD x
```

NETFLIX

Actual data from nstat (EPYC)

```
c368.sjc002.dev# nstat 15
  InMpps OMpps  InGbs  OGbs err TCP Est %CPU syscalls csw   irq GBfree5
  2.72  16.06   1.89 191.62  0 152688   63.18 239427 624677  64309  8.45
  2.72  16.11   1.77 192.28  0 153085   62.51 239378 629312  64761  8.43
  2.73  16.15   1.75 192.71  0 153462   63.56 241100 629712  64713  8.42
  2.74  16.15   1.76 192.80  0 153519   63.71 235867 628167  64608  8.42
  2.72  16.06   1.72 191.62  0 153445   62.12 237885 629998  64338  8.42
  2.72  16.15   1.73 192.72  0 153521   62.62 236958 627838  64571  8.41
  2.71  16.04   1.75 191.47  0 153659   62.43 239150 628649  64474  8.41
  2.72  16.12   1.74 192.36  0 153683   63.31 236789 627122  64427  8.37
  2.71  16.17   1.68 193.02  0 154106   63.69 237980 625786  64237  8.37
  2.70  16.08   1.73 191.98  0 154027   61.68 238149 627631  64273  8.36
  2.74  16.26   1.79 194.14  0 154469   64.59 240687 627712  64643  8.36
  2.74  16.28   1.80 194.29  0 154681   64.73 237997 627571  64813  8.33
^C
c368.sjc002.dev# sysctl hw.ncpu
hw.ncpu: 64
c368.sjc002.dev# sysctl hw.model
hw.model: AMD x[redacted]
```

NETFLIX


c368.sjc002.dev.ix.nflxvideo.net SNMP Throughput, OpStatus, and BGP Route Count
Bits/sec
400.0G
350.0G
300.0G
250.0G
200.0G
150.0G
100.0G
50.0G
0.0G
IPv4 BGP prefixes
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
UTC
19:34
19:36
19:38
19:40
19:42
19:44
19:46
19:48
19:50
19:52
19:54
19:56
19:58
20:00
20:02
PT
12:34
12:36
12:38
12:40
12:42
12:44
12:46
12:48
12:50
12:52
12:54
12:56
12:58
13:00
13:02

Thank you

Slides at:
https://people.freebsd.org/~gallatin/talks/euro2019.pdf

NETFLIX