CS4450

Computer Networks:
Architecture and Protocols

Lecture 26
Where's the puck going?

Spring 2018
Rachit Agarwal

CORNELL UNIVERSITY FOUNDED A.D. 1865

Goals for Today's Lecture

I'd love to meet

the Batman

in the class

CS4450 Introduction to Computer Networks

Spring 2018
May 3, 2018

Name: Batman
NetID: 0000000000
Quiz 3
Time: 15 minutes

- **DO NOT open the quiz booklet until you are told to begin. You should write your name at the top.**
- Write your answers, in a reasonably neat and coherent way, in the space provided. If you wish for something to not be graded, please strike it out neatly.

Problem	Points	Score
1	20	
Total:	20	

“Final” Announcements :-)

- **Final: 05/22 @ 9AM, Gates G01**
- **Programming assignments (projects) are online**
  - As promised, we’ll provide full support
  - Please email Vishal and me if you need help
- **University has a policy on accommodation for TAs going on internships**
  - 2 TAs had to be recused — Justin and Reuben
  - Please note that they’ll not have office hours, but others will
- **Please fill out the course evaluation**
  - Easy way to get 5%
  - Please be constructive
    - I already know I am an asshole
    - Telling me that again may not help :-)

Goals for Today's Lecture

- **Quick review of what we have covered**
- **Quick view on what may be the future**

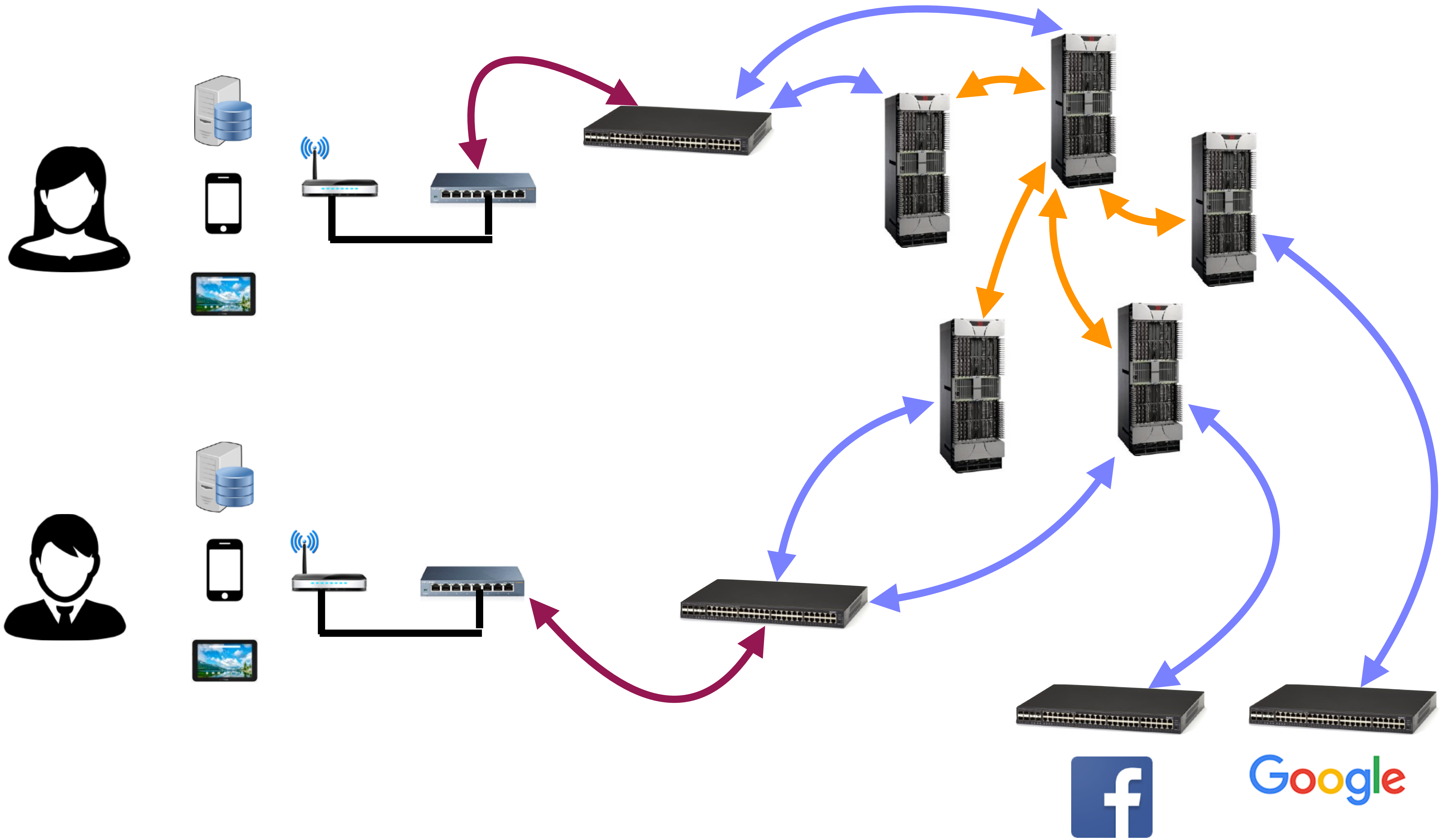
What have we covered?

- **How are computer networks designed and implemented?**

- **Why are computer networks designed the way they are designed?**

What is a computer network?

A set of network elements connected together, that implement a set of protocols for the purpose of sharing resources at the end hosts



Sharing networks

- **Two approaches**
  - Reservation (circuit switching)
  - Statistical multiplexing (packet switching)

- **Motivation for WHY modern networks use “packets”**

- **How to implement this?**

The end-to-end story

- Application opens a **socket** that allows it to connect to the **network stack**
- Maps **name** of the web site to its **address** using **DNS**
- The network stack at the source embeds the address and **port** for both the source and the destination in **packet header**
- Each **router** constructs a **routing table** using a distributed algorithm
- Each router uses destination address in the packet header to look up the **outgoing link** in the routing table
    - And when the link is free, forwards the packet
- When a packet arrives the destination:
    - The network stack at the destination uses the port to forward the packet to the right application

Realizing end-to-end design: Three Principles

- How to break system into modules
  - **Layering**
- Where are modules implemented
  - **End-to-End Principle**
- Where is state stored?
  - **Fate-Sharing**

Five Layers (Top - Down)

- **Application:** Providing network support for apps
- **Transport (L4):** (Reliable) end-to-end delivery
- **Network (L3):** Global best-effort delivery
- **Datalink (L2):** Local best-effort delivery
- **Physical:** Bits on wire

Link Layer (L2)

- **Broadcast medium:** Ethernet and CSMA/CD
- **We studied that Ethernet does not scale to large networks**
  - Motivation for switched Ethernet
- **Broadcast storm:** if using broadcast on switched Ethernet
  - Motivation for Spanning Tree Protocol
- **Limitations of Spanning Tree Protocol:**
  - Low bandwidth utilization, high latency, unnecessary processing
  - Does to scale to the entire Internet
  - Motivation for **routing protocols** in the Internet

Network Layer (L3)

- **Internet Protocol:**
  - Addressing, packet header as an interface, routing
- **Routing tables:**
  - Correctness and validity: Dead ends, loops
  - A collection of spanning trees, one per destination
- **Constructing valid routing tables (within an ISP)**
  - Link-state and distance-vector protocols
  - Focused a lot on learning via examples
  - Can still have loops: failures remain to be a pain
- **How to use routing tables**
  - **Packet header as an interface**
  - Learnt why packet headers look like the way they do

Network Layer (L3), Cont.

- **Internet Protocol:**
  - Addressing, packet header as an interface, routing
- **Addressing:**
  - Link layer uses "flat" addresses
  - **Does not scale to Internet**: motivation for IP addresses
  - **Scalability challenges**: Routing table sizes, #updates
  - Solution: **Hierarchical addressing**
- **Forwarding**
  - **Switch architecture**
  - Longest Prefix matching for forwarding at line rate
  - Scheduling using priorities

Network Layer (L3), Cont.

- **Internet Protocol:**
  - Addressing, packet header as an interface, routing
- **Limitations of link-state and distance-vector routing:**
  - Require visibility of the entire Internet
  - **ISPs do not like that:** motivation for Inter-domain routing
  - **Border Gateway Protocol**
    - A simple modification of distance-vector protocol
- **Routing with policies**
  - **Customer-provider-peer relationships**
  - Gao-Rexford policies
- **Completes the network layer: provides connectivity**

Details for complete picture

- **DHCP: Dynamic Host Configuration Protocol**
    - For each host to figure out its IP address, local DNS, first-hop router
- **ARP: Address Resolution Protocol**
    - For finding other servers on the same local area network (L2)
    - Mapping from IP addresses to names (MAC addresses)
- **Domain Name System**
    - **Mapping Human readable destination names to IP addresses**
    - Hierarchical structure

A brief Jump to Application Layer

- **HTTP and the Web**
  - Client-server architecture
  - Performance issues, and improving performance via caching
  - Content Distribution Networks
- **Traffic Engineering, MPLS and Multicast**
  - For load balancing traffic in the network
  - And, for efficient content delivery
  - **Not included in the finals**

Transport Layer

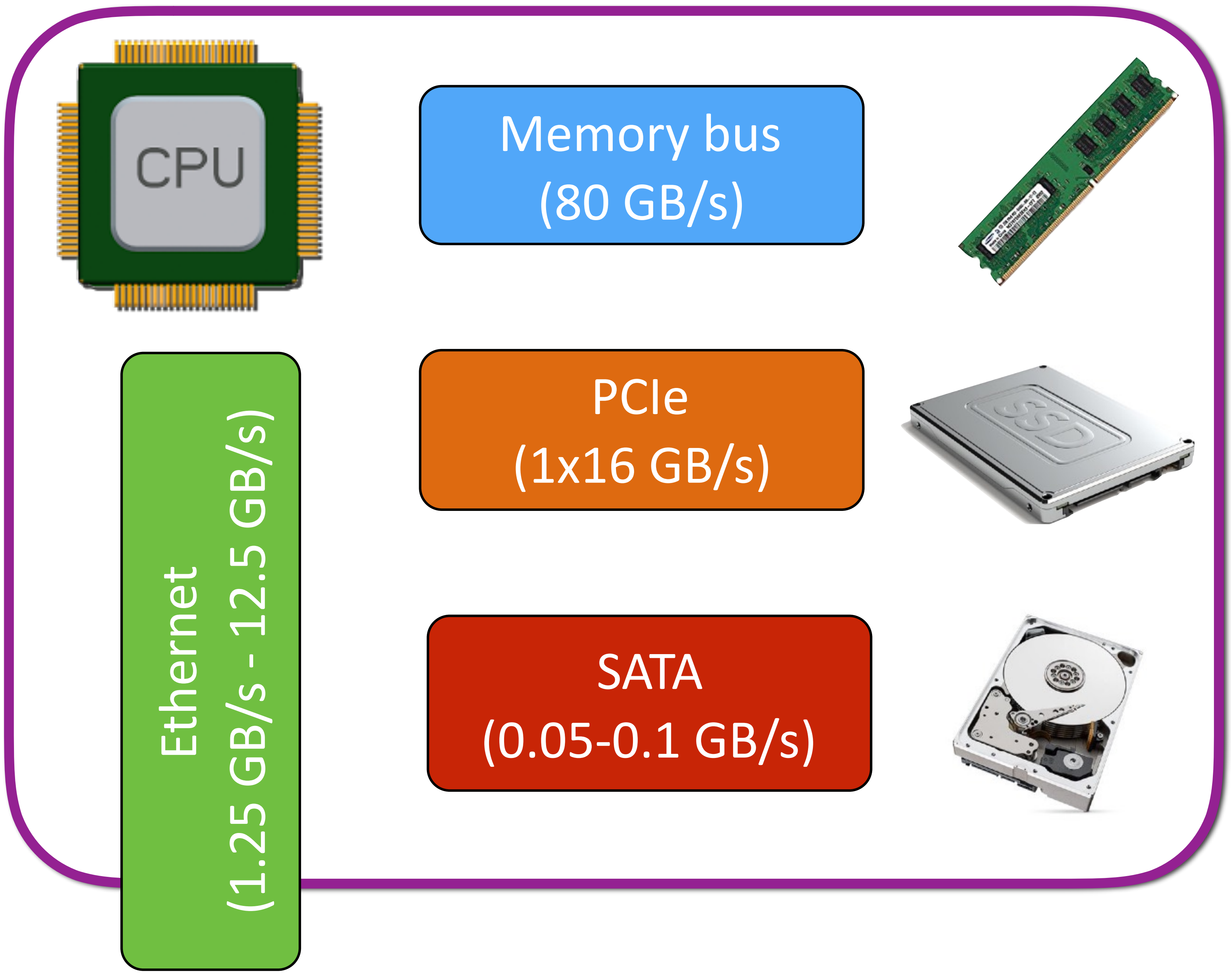
- **Goals of reliable transport**
  - **Correctness condition**
  - Why do we need ACKs, timers, window-based design
- **TCP**
  - Mostly implementation details following the above design
  - For fairness, flow performance and utilization
  - **Flow control**
    - Ensuring the sender does not overwhelm the receiver
    - Via receiver advertised window size
  - **Congestion control**
    - Dynamic adaptation to network path's capacity
    - Slow start, Additive-increase Multiplicative-decrease, timeouts

What may be the future?

Skate where the puck's going, not where it's been!

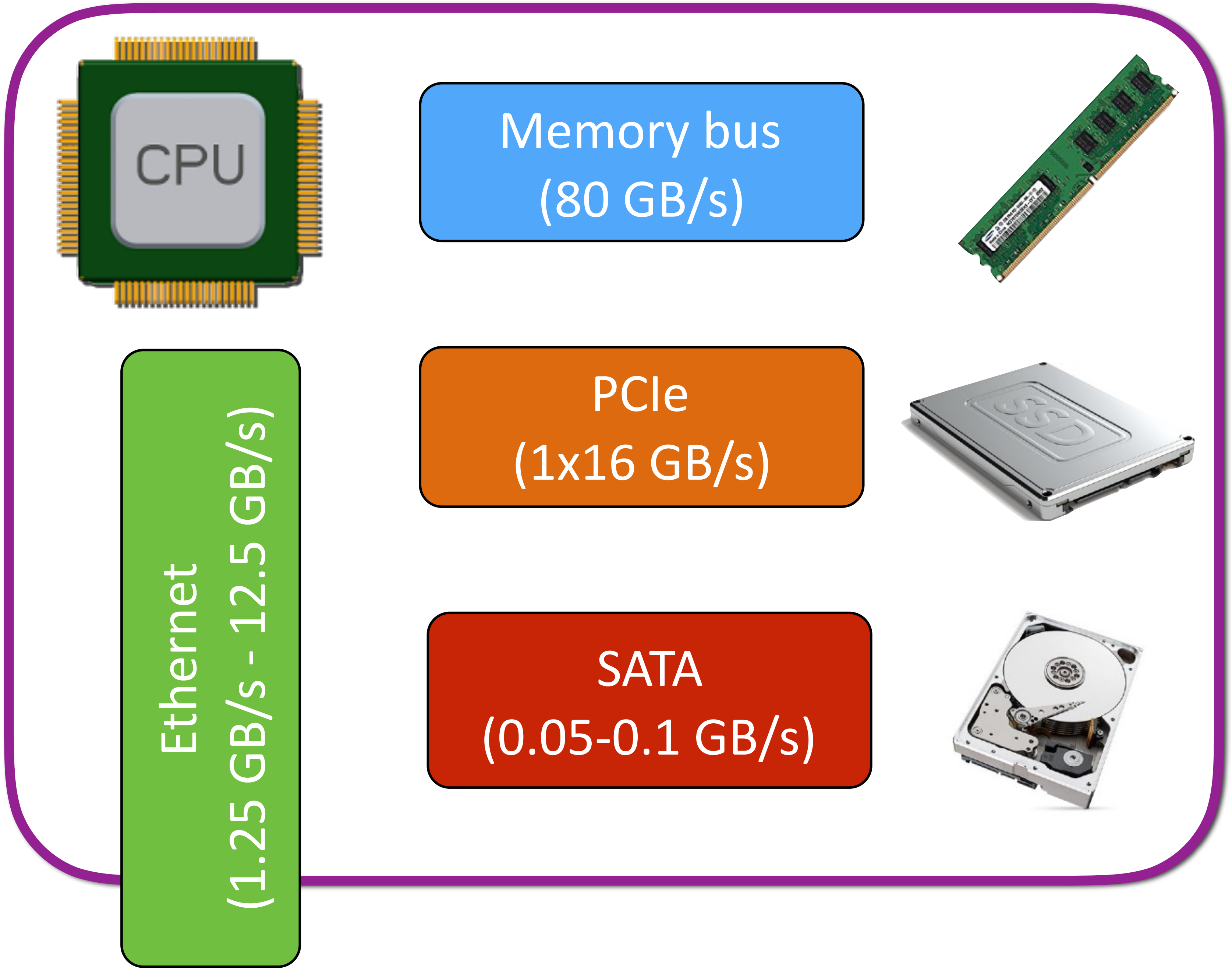
- Walter Gretzky

Where is the puck right now?

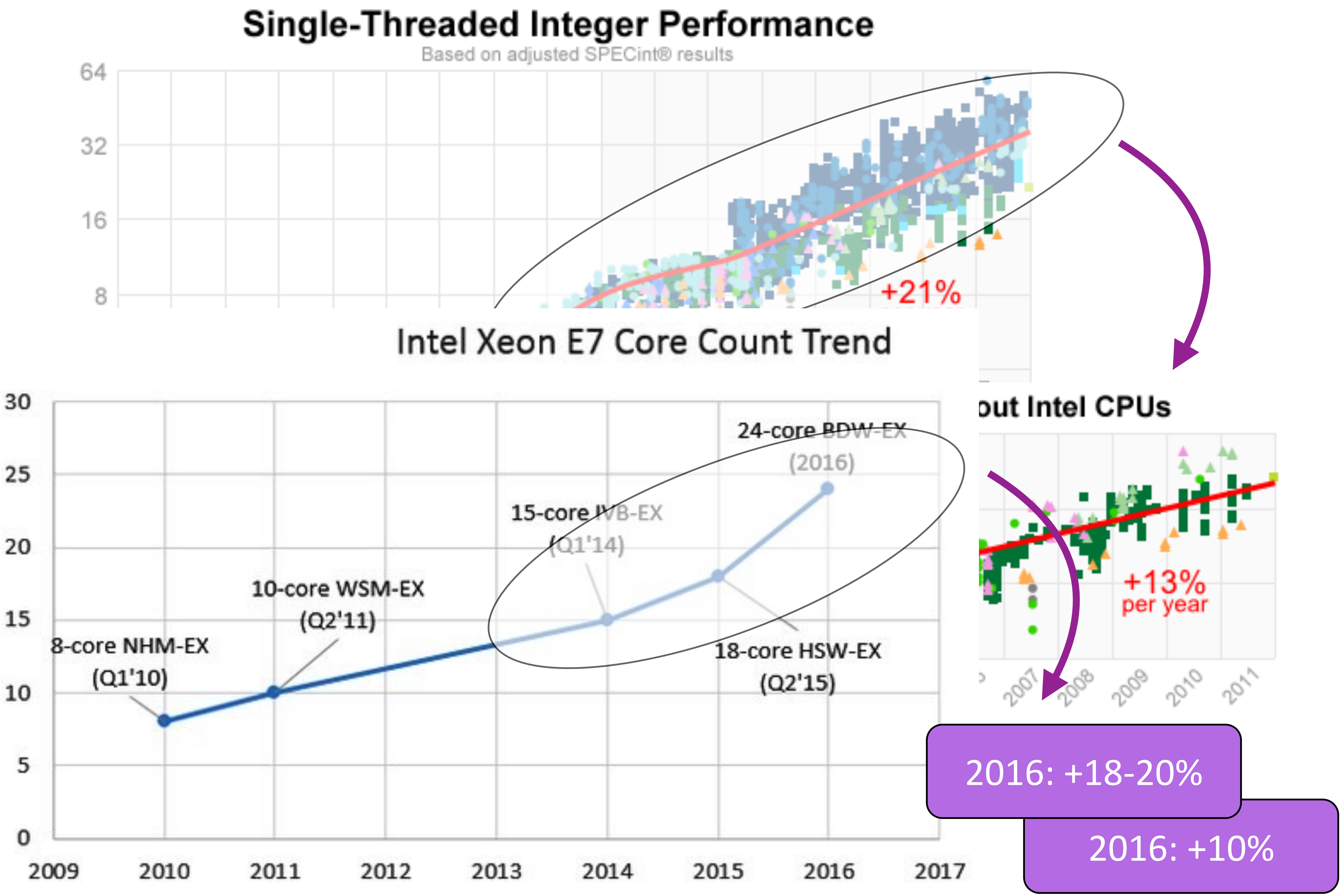


Size (TB)	Random Access (us)	Seq. Access (GB/s)
0.1	0.1	80
1	25	1x
10	4000	0.1x

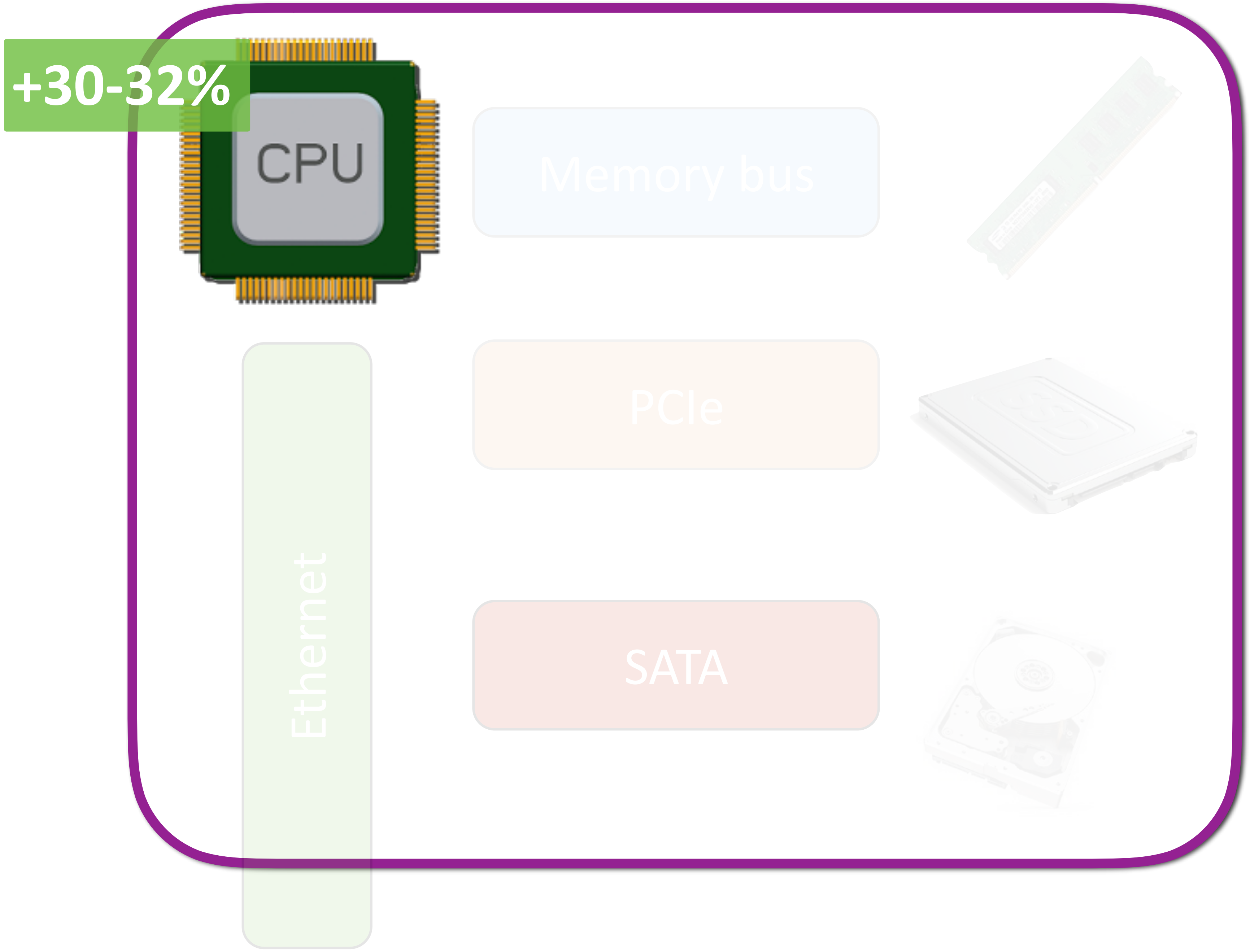
Where is the puck going?



Where is the puck going? (CPU performance)

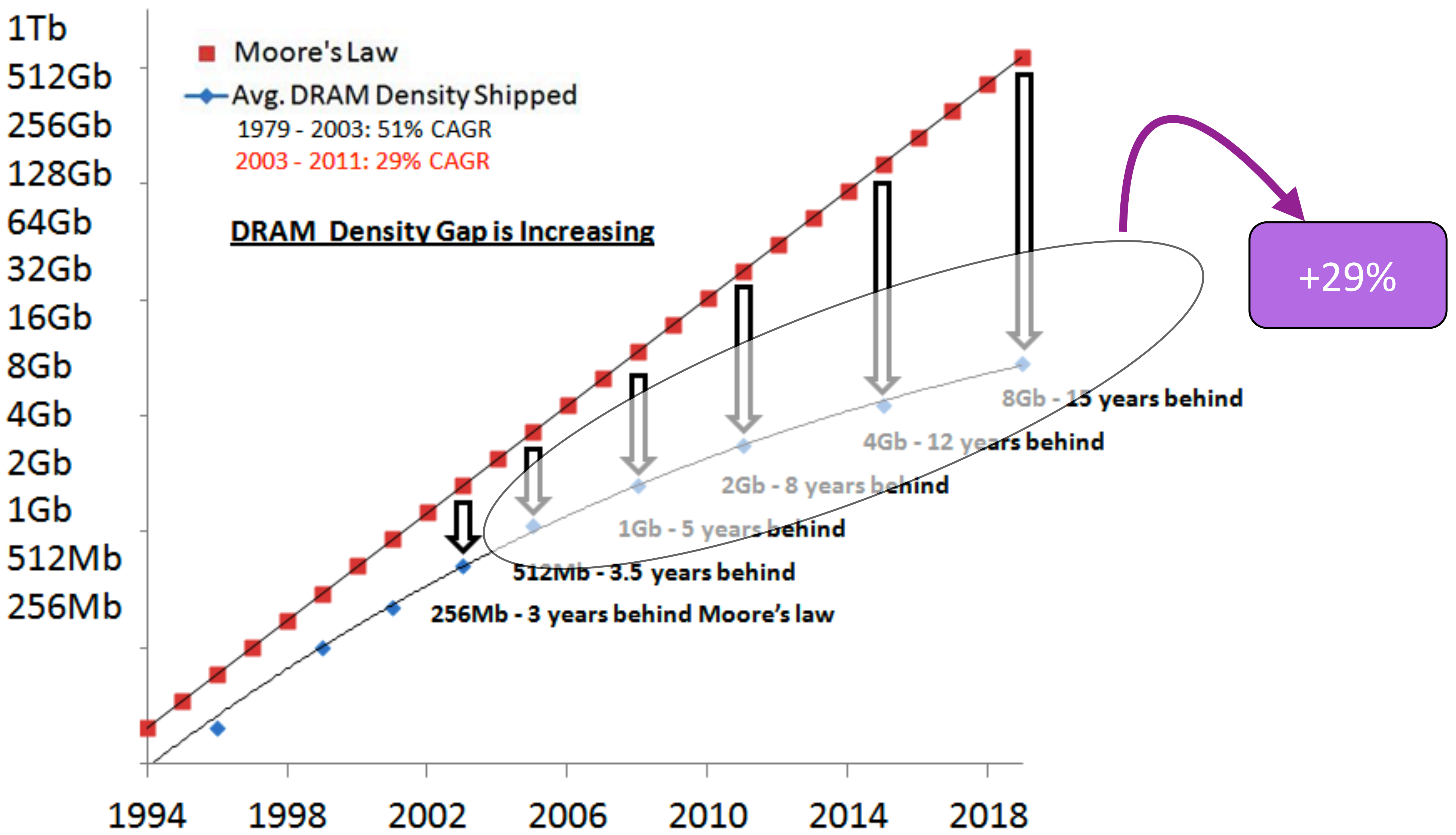


Where is the puck going?

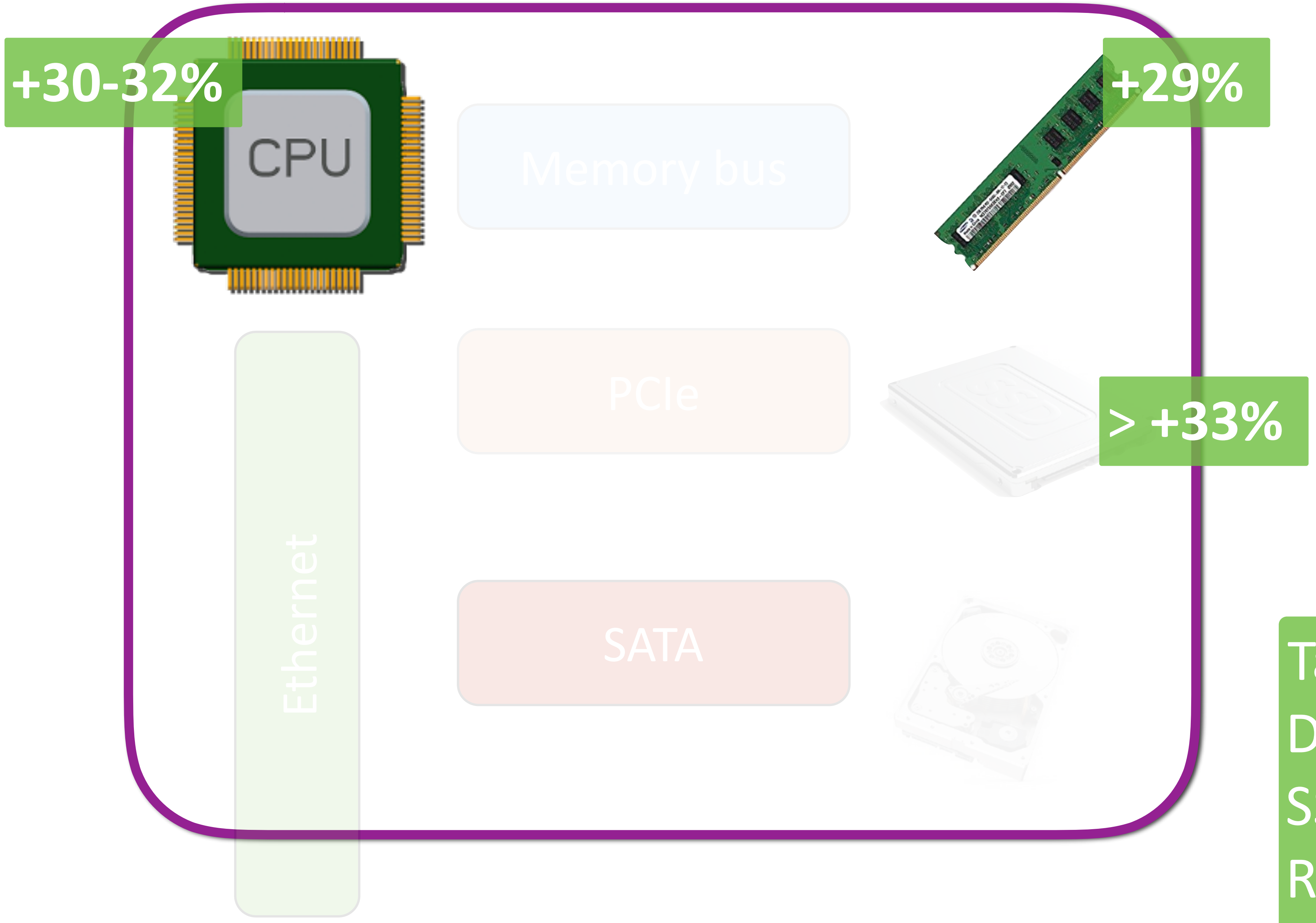


- **#Cores: +18-20%**
- **Per core: +10%**

Where is the puck going? (DRAM capacity)



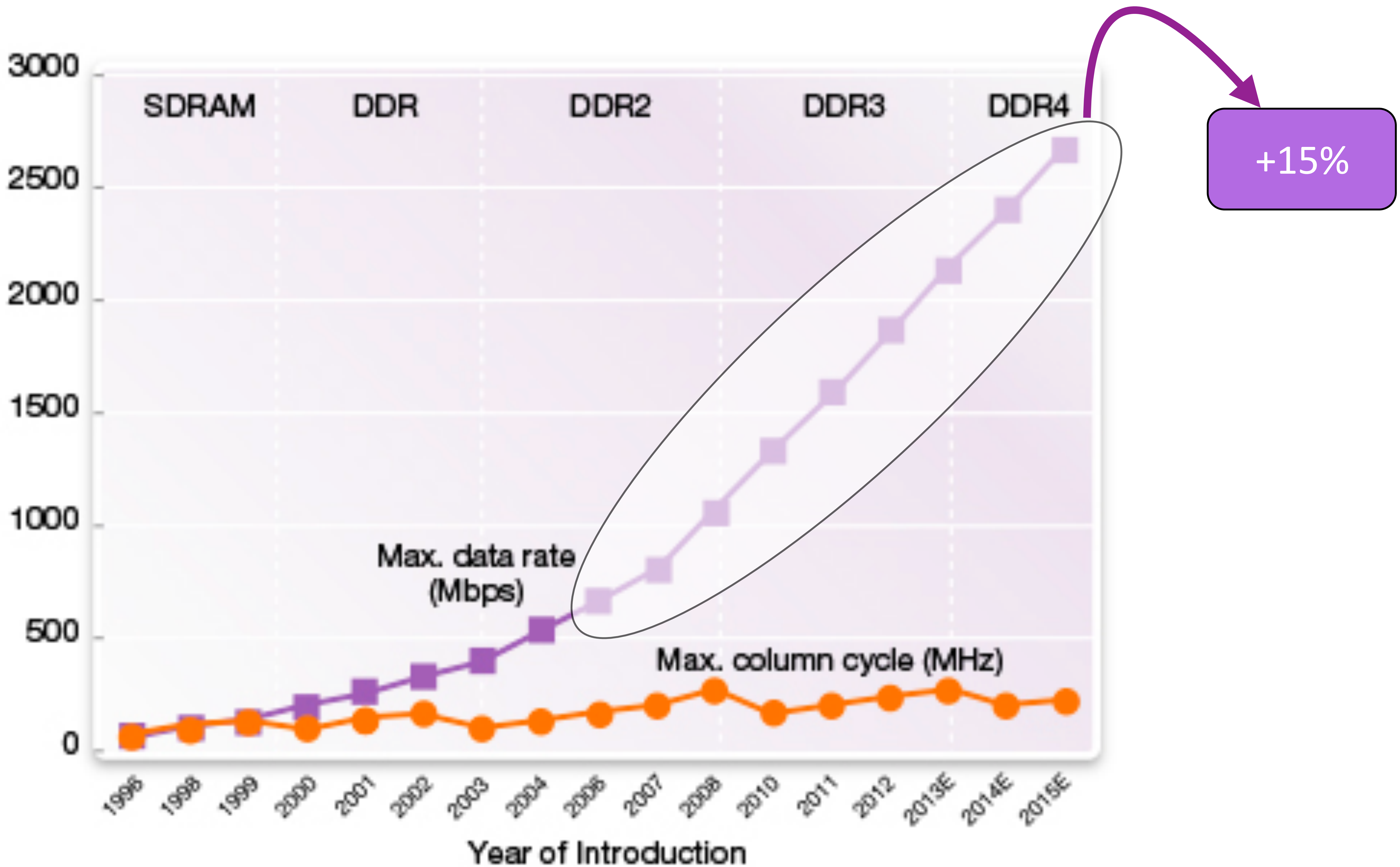
Where is the puck going?



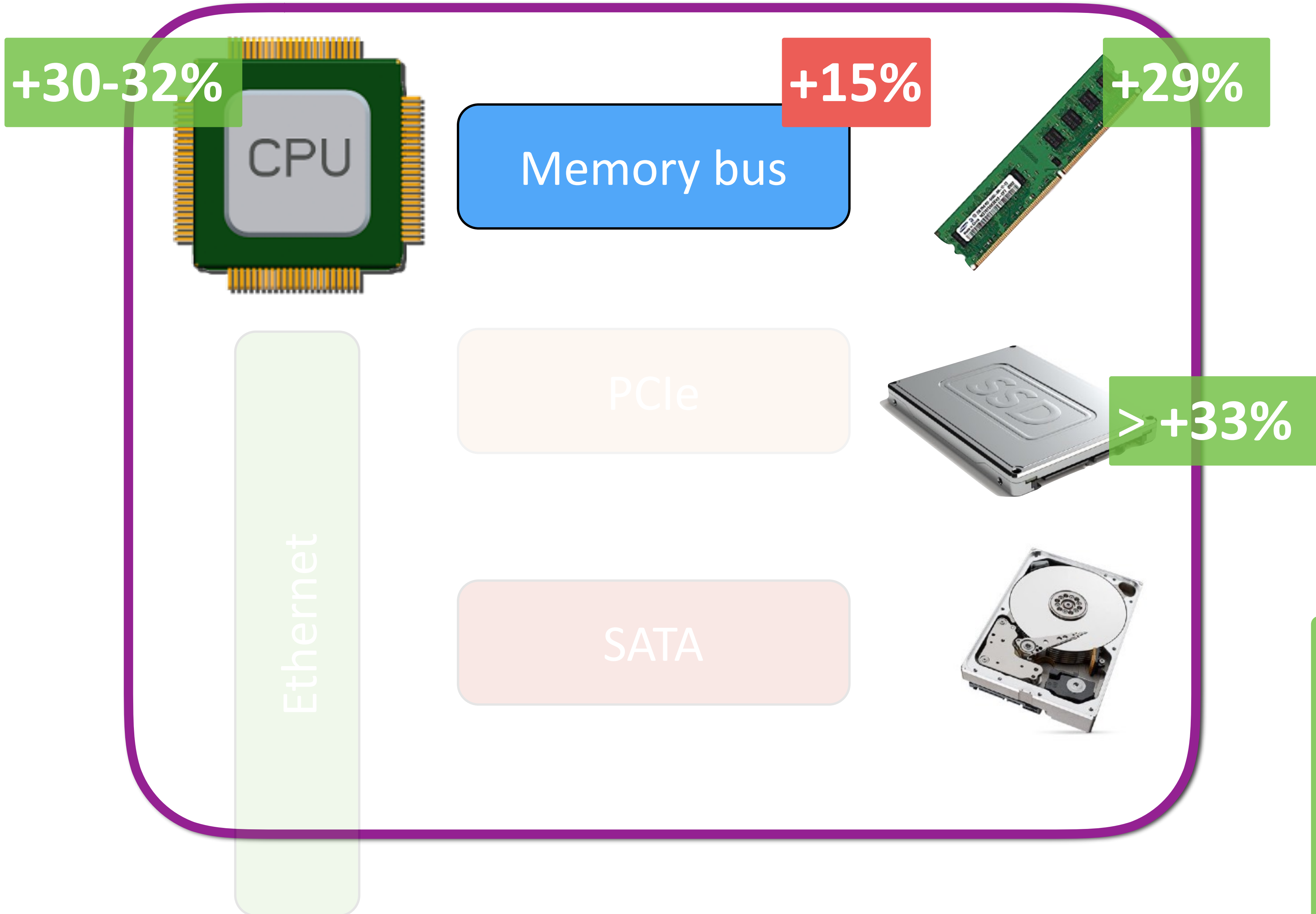
Tape is dead,
Disk is tape,
SSD is disk,
RAM is the king!

- Jim Gray

Where is the puck going? (Memory bus)



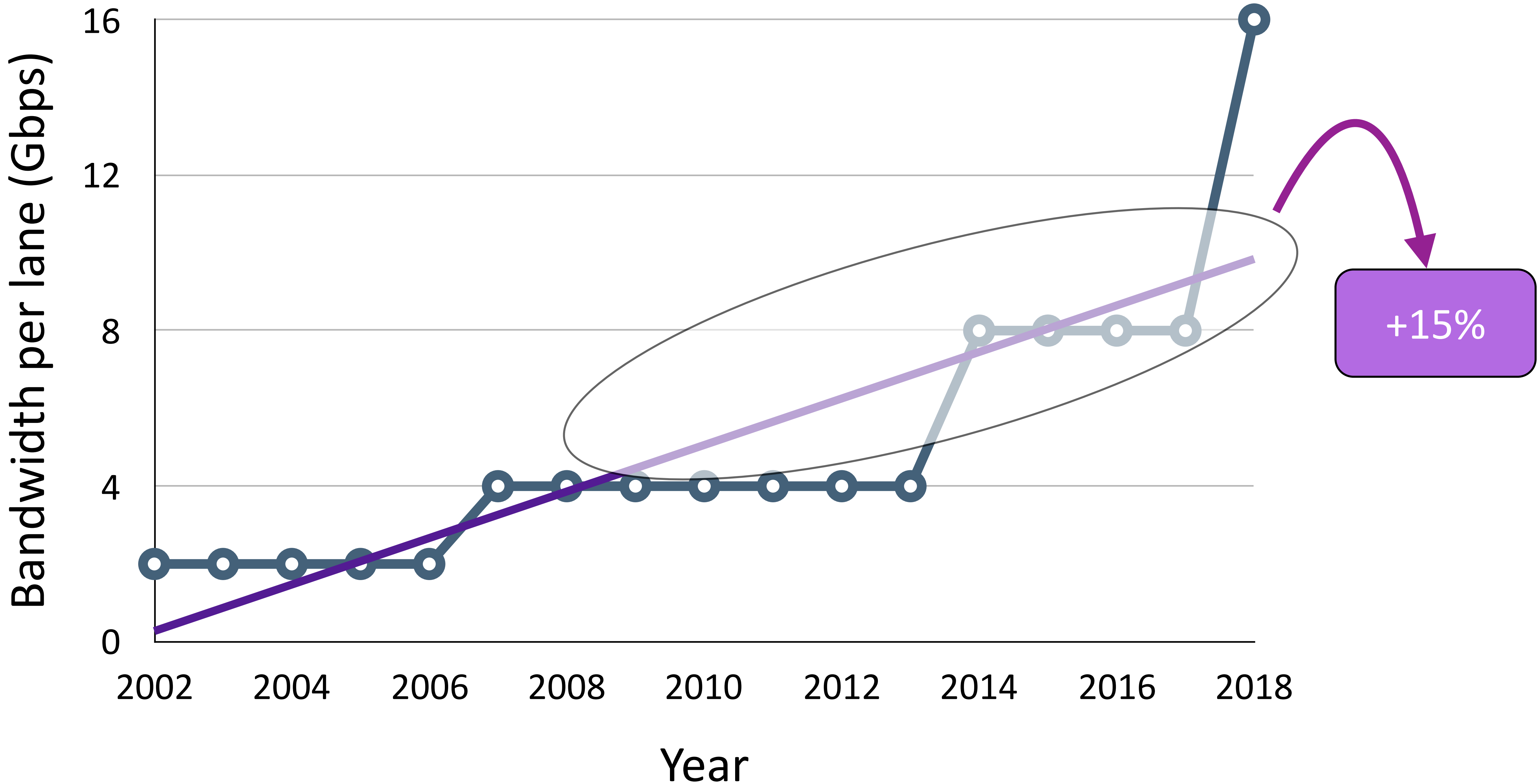
Where is the puck going?



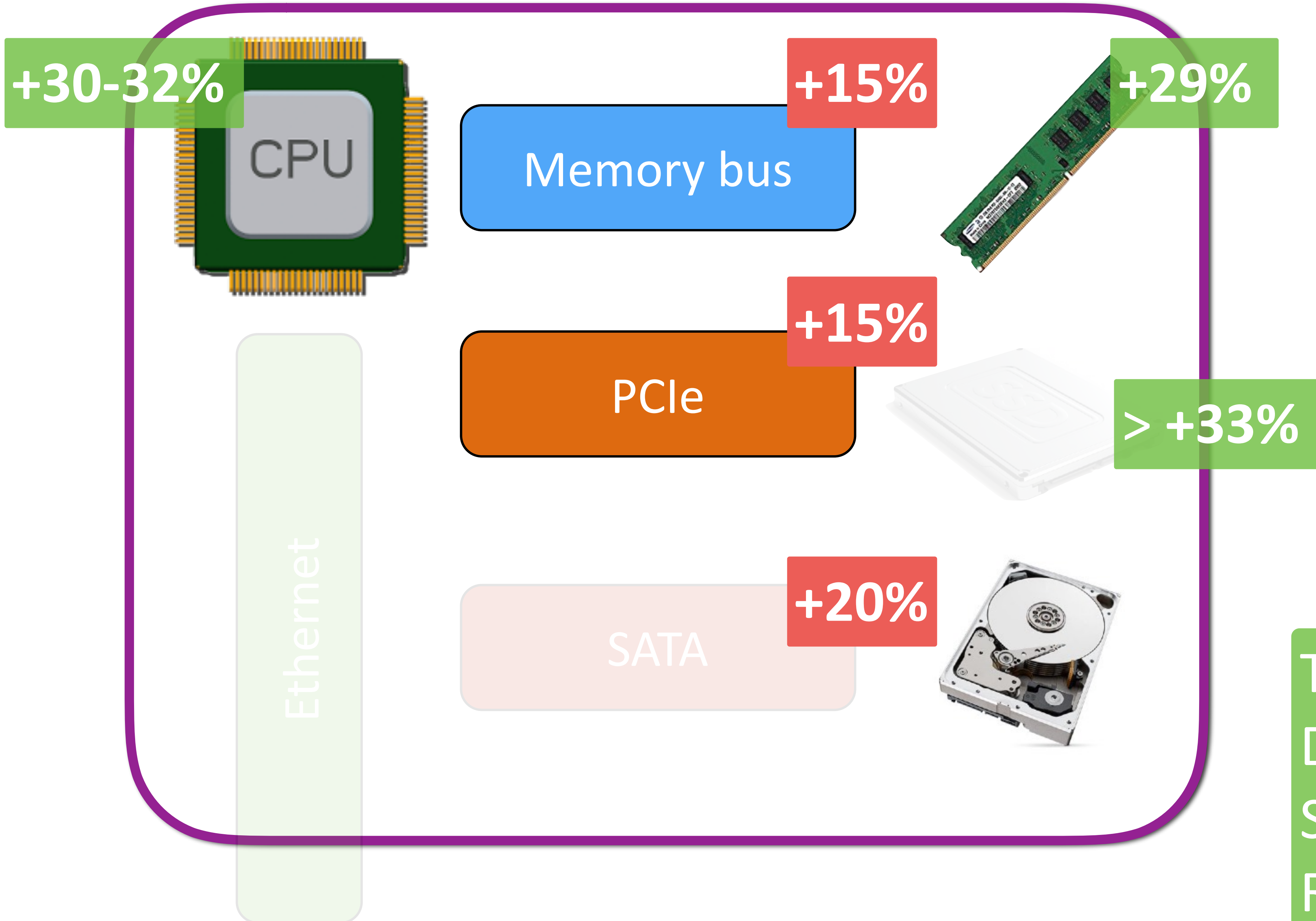
Tape is dead,
Disk is tape,
SSD is disk,
RAM is the king!

- Jim Gray

Where is the puck going? (PCIe)



Where is the puck going?

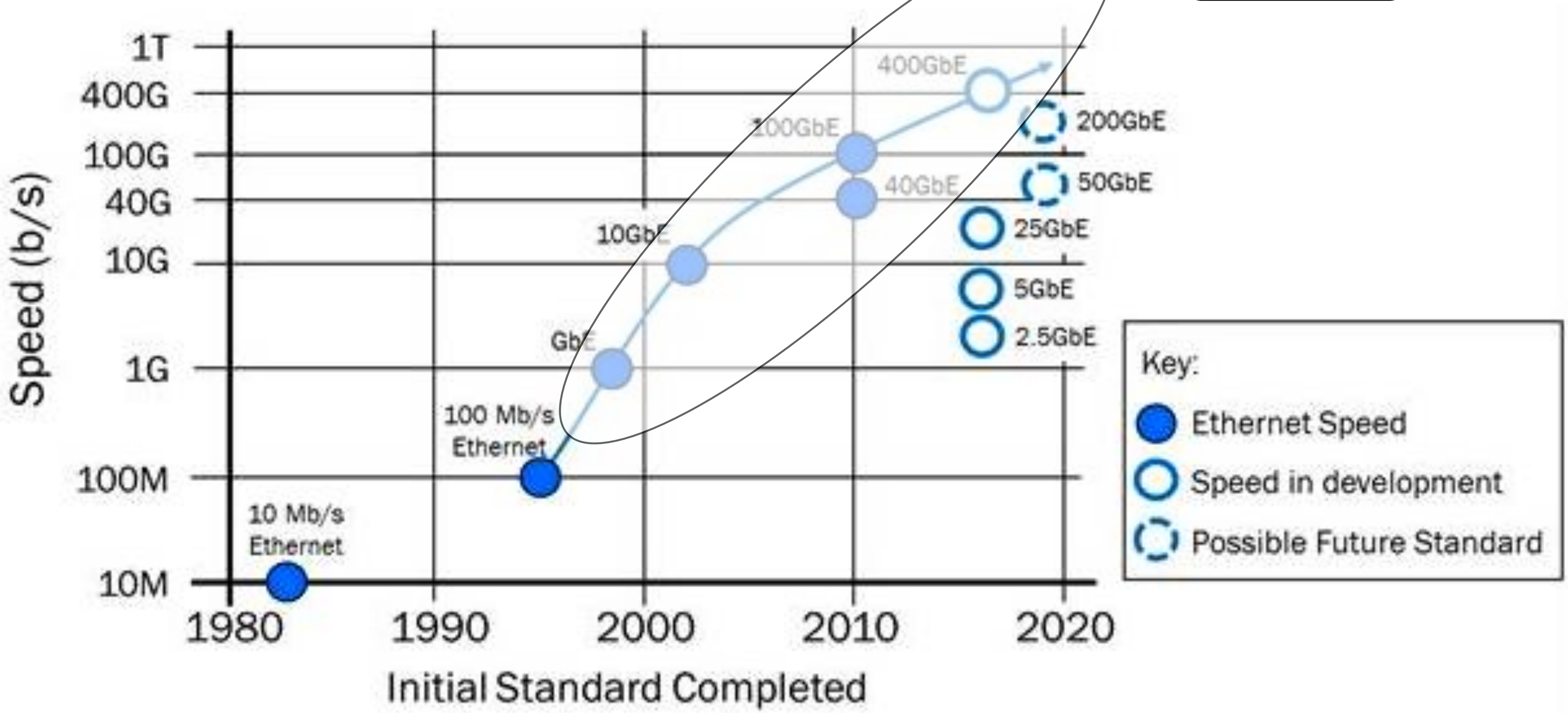


Tape is dead,
Disk is tape,
SSD is disk,
RAM is the king!

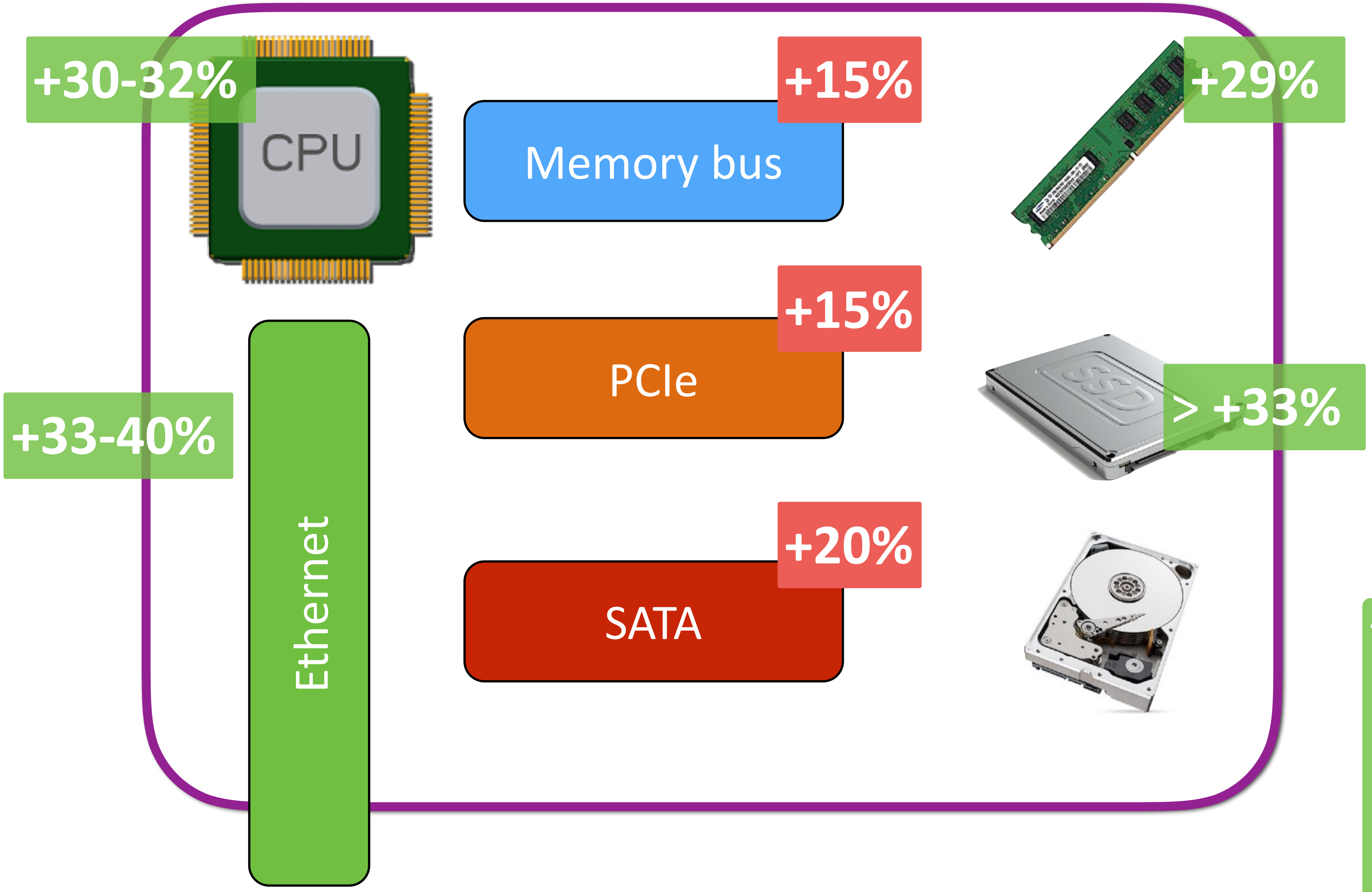
- Jim Gray

Where is the puck going? (Ethernet)

+33-40%



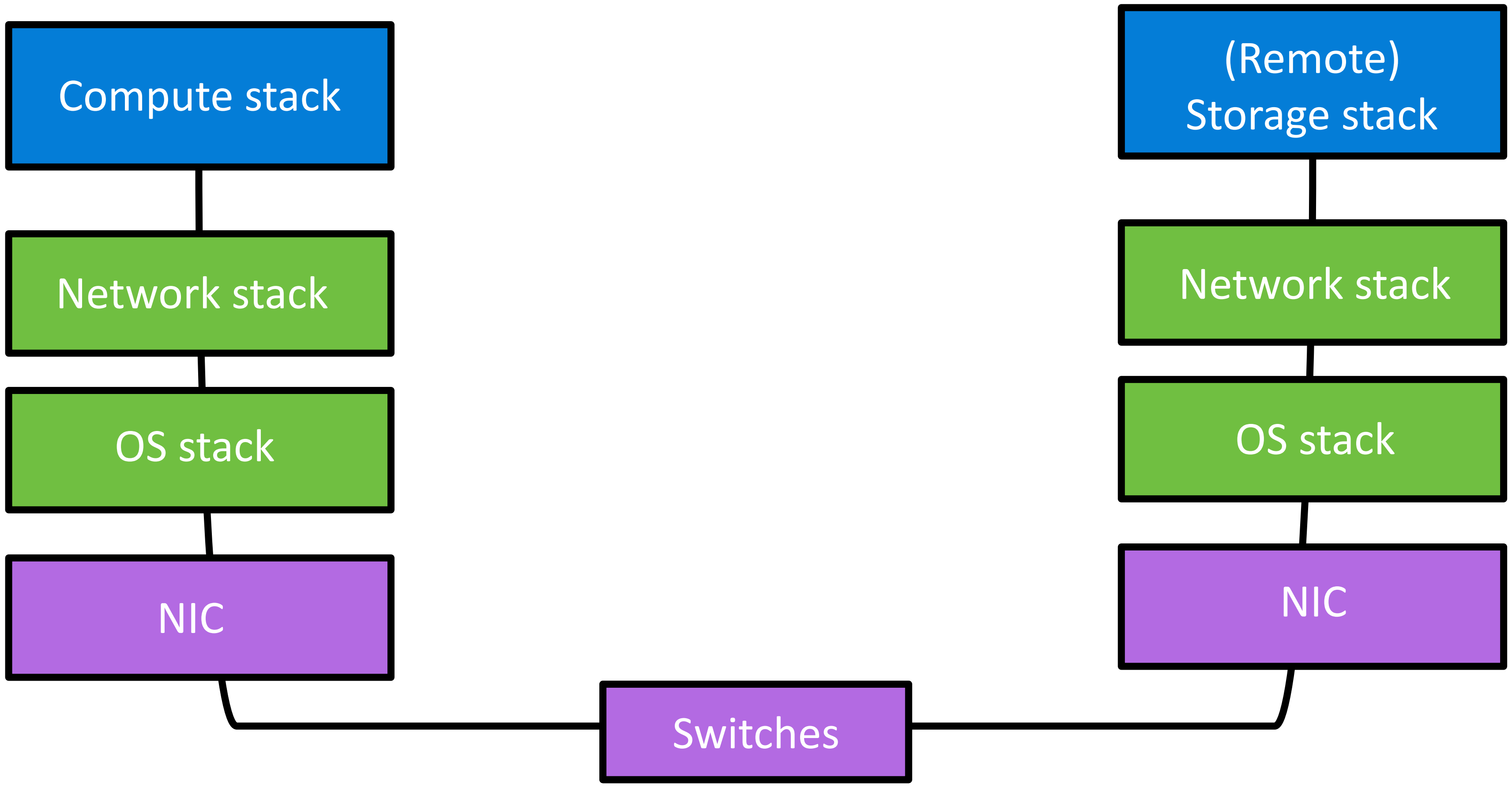
Where is the puck going?



Tape is dead,
Disk is tape,
SSD is disk,
RAM is the king!

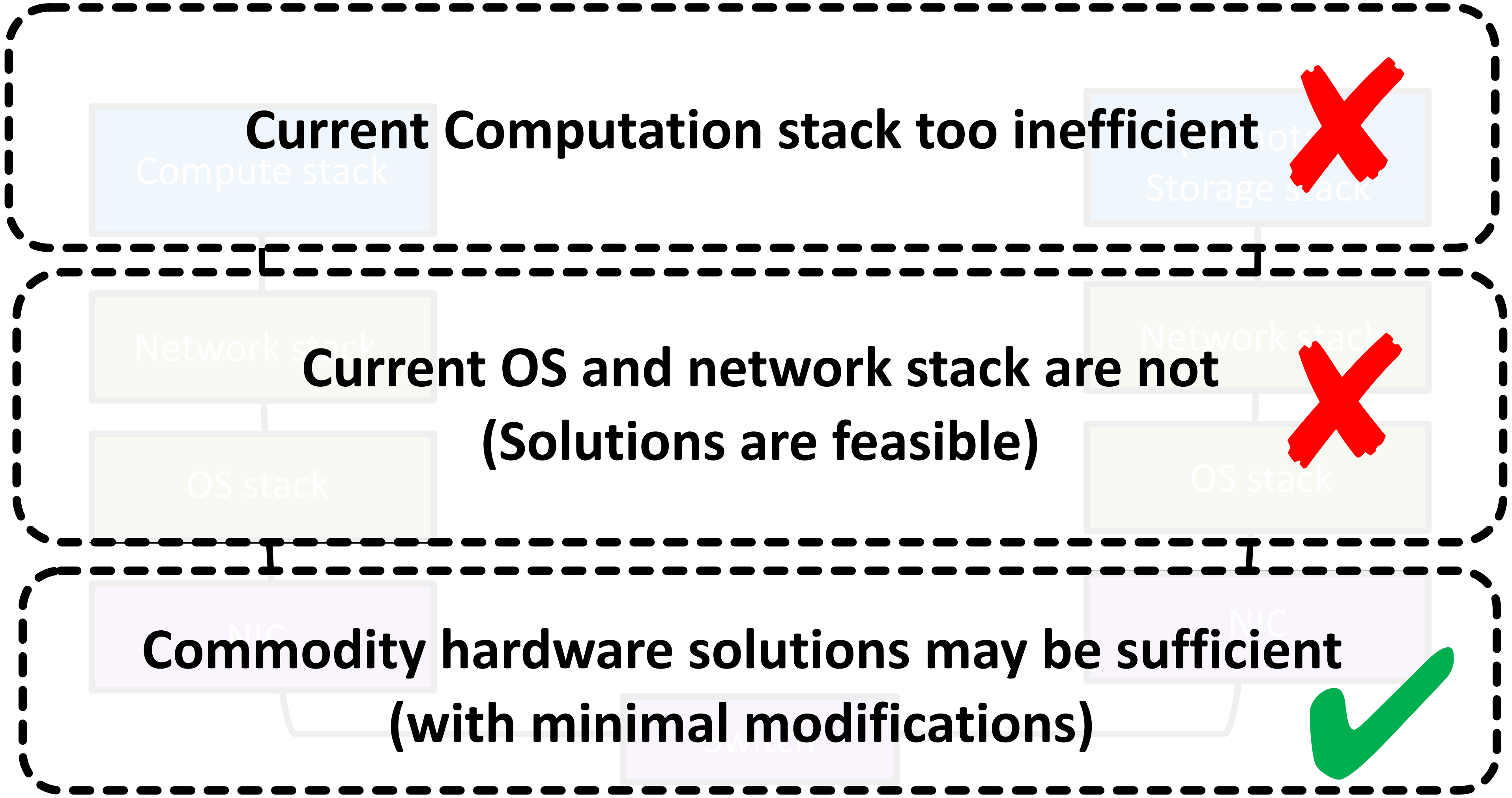
- Jim Gray

Where is the puck going? — End-to-end networks



	~2010 (1Gbps)		2017 (40Gbps)	
	Inter-rack	**%**	**Inter-rack**	**%**
OS	[2 x 0.95]	5	2 x 0.95	27
Data copy	[2 x 1.00]	5	2 x 1.00	29
Switching	6 x 0.45	7	6 x 0.24	20
Propagation	4 x (0.20 + 0.02)	2	4 x (0.20 + 0.02)	13
Transmission	30.52	81	0.81	11
TOTAL	**38**		**7.03**	

Disaggregated architectures — Network Fabric Research



- Network fabric must provide 40-100Gbps bandwidth
    - Feasible with existing hardware
- Network fabric must provide 2-5us latency
    - Requires redesign of OS and network stack

Current network stacks are the bottleneck

- **Lot of research in "hardware offload"**
  - Implementing TCP (and other mechanisms) on hardware
  - Lot of interesting challenges
- **Lot of research in low-latency transport design**
  - TCP was not designed for low latency
  - New transport protocols for ultra low-latency
- **Lot of research in kernel-bypass**
  - TCP requires processing each and every packet
  - 1Gbps links: 90,000 packets per second
  - 100Gbps links: 9 million packets per second
  - Extremely high CPU requirements
  - Bypass the kernel entirely
    - Implement congestion control in user space

Closing Thoughts

- **These are exciting times for computer networking**
  - The first ever since the invention of the Internet
  - You are witness to the transformation
- **And, I am glad I got the chance to introduce you to this world :-)**
  - You have made me a better teacher
  - Thank you!!!
- **Wherever you end up:**
  - Please remember me
  - Say hello if you see me
  - Remember, there is nothing more important than
    - **Knowing the fundamentals**
    - **Being happy!**