OCP-TAP Presentation

Precise Time Applications

Dan Biederman
Technical Leader
Ethernet Products Group
dan.biederman@intel.com

Disclaimer – the views of this presentation are those of Dan Biederman and not necessarily Intel.

Notices & Disclaimers

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

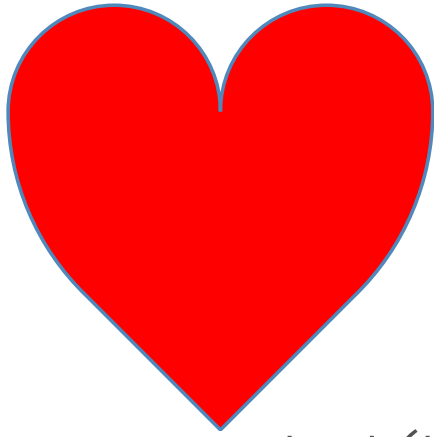
© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

Agenda

- Introduction (A love story!)
- Precise Time today
- Processor's Use of Precise Time
- Trends in Precise Time
- Ideas for the future
- Conclusion
- Q&A

Let's start off a little different. Let's have a Love Story.

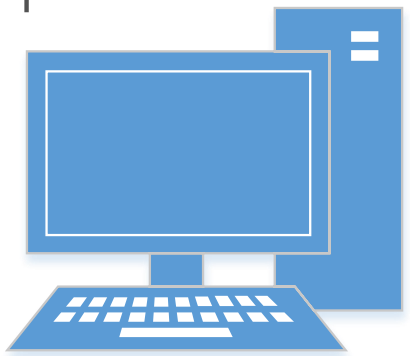
- Once there was a high school senior, who fell in love with a girl.



- She got grounded (bad grades)
- The graduating senior decided to save up all his summer job money to spend on the girl, once she got ungrounded

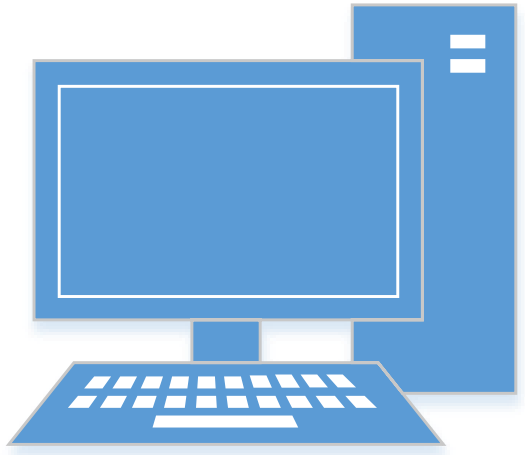
- After a few months, he saved about $2000.
- However, he found out that she liked someone else… and it was over.
- So, he spent the money on his next love, a computer:

Intel 386 SX
w/Turbo button
+ Printer



College Years

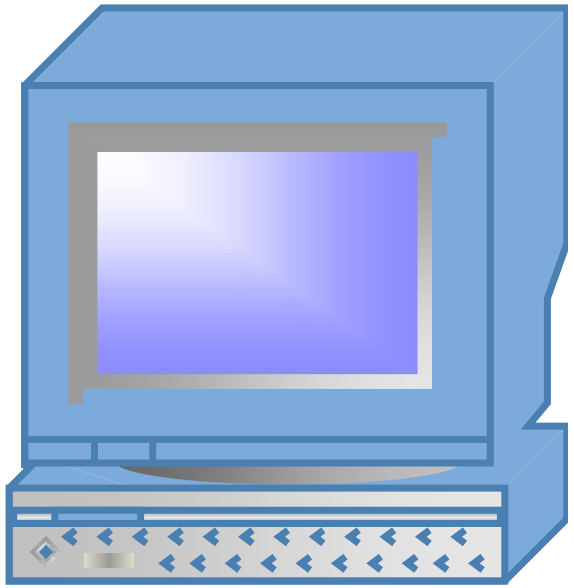
- The computer let him:
  - Write papers,
  - Run simulations,
  - Play games,
  - Meet and help people!



- After graduating, he went on to his masters.
  - Logic Design
  - Computer Architecture
  - Artificial Intelligence
- Unfortunately, doing his masters, the Intel 386 SX was too slow to run the latest AI training programs
- He would never finish…

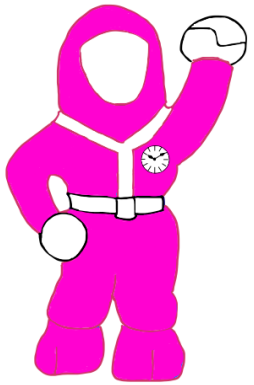
College Years: Masters

- Luckily, the university got a few Intel Pentium workstations
- Ran at 90 MHz!



Workstation

- Using these machines, he could train his neural networks, complete his thesis, and graduate with a Masters in EE.



- He also met his soon-to-be wife while doing his masters.
- They both wrote their Master's thesis on the Intel 386 SX…

Industry

- In industry, he worked on the first Industrial Ethernet HW product with IEEE1588 at Cisco
- He worked with other groups at Cisco, supporting their IEEE1588 efforts.
- Worked at two startups
- Hired later by Ericsson for his time synchronization experience

- Now part of Intel working on the next generation Ethernet Products
- Doing Wonderful Things with Precise Time Synchronization…
- But he needs help!

Precise Time Tools That Are Available "Today"

- IEEE1588
  - Available on the NIC
  - In some cases, directly to the NIOS II processor (i.e., FPGAs)
  - OCP-TAP Presentations on White Rabbit for Very High Precision and Accuracy
- PTM
  - Precise Time over PCIe
  - Converts Network Time to the processor time
  - FPGA's/NIC's Time to the CPU's TimeStamp Counter
- RDTSC
  - Read the Time Stamp Counter
  - The counter used to determine the current time as seen by the CPU
- Instructions that use precise time
  - TPAUSE

CPU Time → Synchronized Time

Time "now"

```
(1) clock_gettime(CLOCK_MONOTONIC_RAW, &now);
```

- Returns current TSC value scaled to nominal nanoseconds

```
(2) clock_gettime(CLOCK_MONOTONIC,     &now);
```

- Returns current TSC value scaled to track TAI, in nanoseconds

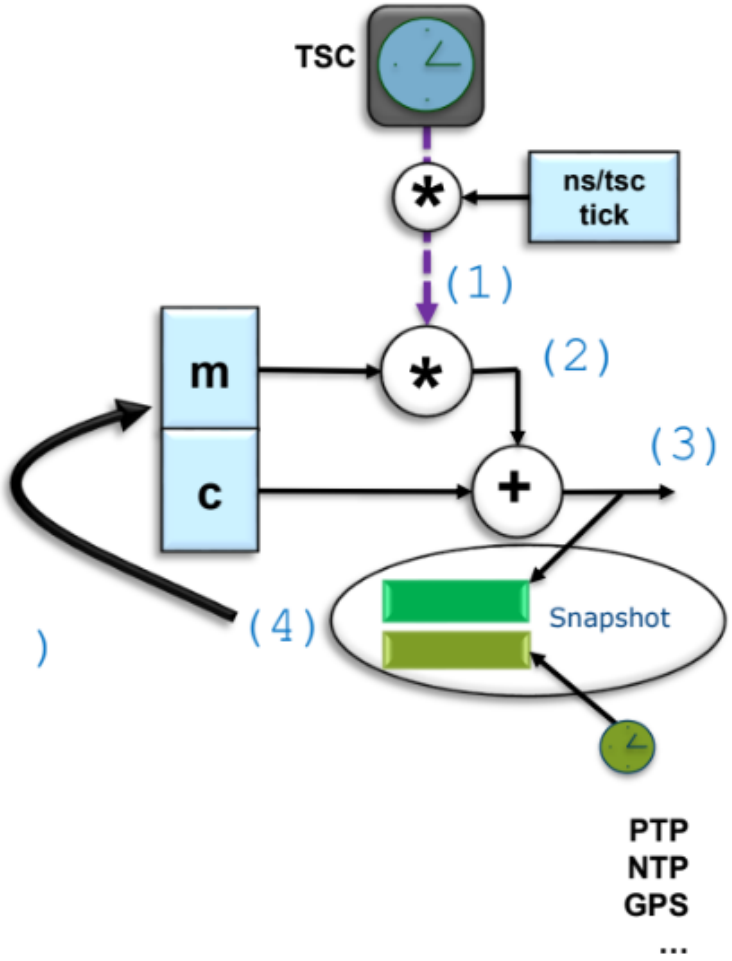
```
(3) clock_gettime(CLOCK_REALTIME,      &now);
```

- Returns CLOCK_MONOTONIC + (now-1/1/1970) [incl. leap seconds]

Cross-Timestamp

```
(4) ioctl(phc_fd,PTP_SYS_OFFSET[_PRECISE], &offset )
```

- returns the triple:
  - eth_ptp_time; realtime; monotonic_raw



POSIX: Piecewise-Linear Clock Model: y[n]=mx[n]+c

Precise Time at the Processor: RDTSC and TPAUSE

- RDTSC – Allows you to read the Time Stamp Counter (TSC)
- TPAUSE – is an instruction that waits for a value of the TSC
- Note: check your processor to confirm support.

RDTSC—Read Time-Stamp Counter

Opcode*	Instruction	Op/ En	64-Bit Mode	Compat/ Leg Mode	Description
0F 31	RDTSC	ZO	Valid	Valid	Read time-stamp counter into EDX:EAX.

Instruction Operand Encoding

Op/En	Operand 1	Operand 2	Operand 3	Operand 4
ZO	NA	NA	NA	NA

Description

Reads the current value of the processor's time-stamp counter (a 64-bit MSR) into the EDX:EAX registers. The EDX register is loaded with the high-order 32 bits of the MSR and the EAX register is loaded with the low-order 32 bits. (On processors that support the Intel 64 architecture, the high-order 32 bits of each of RAX and RDX are cleared.)

TPAUSE—Timed PAUSE

Opcode / Instruction	Op/ En	64/32 bit Mode Support	CPUID Feature Flag	Description
66 0F AE /6 TPAUSE r32, <edx>, <eax>	A	V/V	WAITPKG	Directs the processor to enter an implementation-dependent optimized state until the TSC reaches the value in EDX:EAX.

Instruction Operand Encoding[1]

Op/En	Tuple	Operand 1	Operand 2	Operand 3	Operand 4
A	NA	ModRM:r/m (r)	NA	NA	NA

Description

TPAUSE instructs the processor to enter an implementation-dependent optimized state. There are two such optimized states to choose from: light-weight power/performance optimized state, and improved power/performance optimized state. The selection between the two is governed by the explicit input register bit[0] source operand.

From: Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D and 4

https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html

Real Time Use Case

From James Coleman (Intel PE)

- You need to get to the Ferry by 10am
- If you are late, the trip is useless
- What do you do?
  - Your time needs to be in sync with the Ferry Time
    - Ferry Timetable
    - Captain must meet schedule
    - You are dependent on that schedule
  - It is a 20-minute trip
  - Leave 30 minutes early
    - 10-minute buffer
  - Check you watch, if late (i.e. traffic)
    - Drive faster
    - Use the HOV lane (even if just 1 person)
    - Run late yellow lights
    - California Stops (Glide through stop signs)
  - If you ever hit a point where you can't get there by 10am. You turn around.

Internet of Things Real Time Use Case*

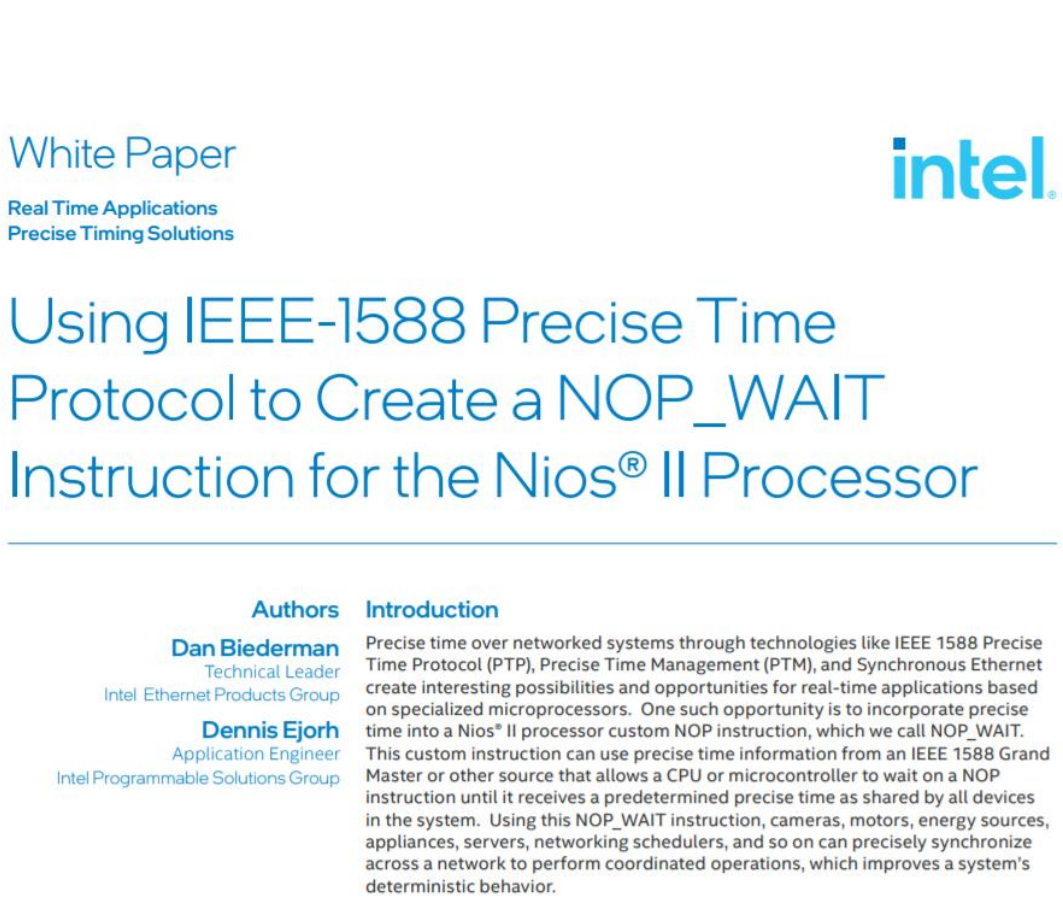
- In IOT, there are real-time compute situations
- You must complete a task by a defined time
- What do you do:
  - Give extra time, for the worst possible case (99.9999%)
  - Throw extra processing at the problem
- What could you do?
  - Monitor progress and time
  - What to do if you are running late?

* from James Coleman

Comparison of Ferry to Real Time Computing

	Ferry	Real Time Computing
Schedule	Ferry Time-Table	Application Schedule / SLA
Time Base	Wall Clock / Watch / Cellphone	GPS / IEEE1588 / PTM / White Rabbit
Performance improvement #1	Drive Faster	Speed Up Processor Clock
Performance improvement #2	Use HOV Lane	Slow down other tasks.
Performance improvement #3	Run late yellow lights	Punt tasks to another processor
Performance improvement #4	California Stops (Glide through Stop Signs)	Allocate more resources to the task
If you are late	Miss the Ferry	Miss your SLA/Performance
Checking Time	Check Watch or Car's Clock	RDTSC or other clock mechanism (POSIX)

TPAUSE and Using Precise Time in FPGA Microcontrollers



White Paper

Real Time Applications
Precise Timing Solutions

intel

Using IEEE-1588 Precise Time Protocol to Create a NOP_WAIT Instruction for the Nios® II Processor

Authors

Dan Biederman
Technical Leader
Intel Ethernet Products Group

Dennis Ejorh
Application Engineer
Intel Programmable Solutions Group

Introduction

Precise time over networked systems through technologies like IEEE 1588 Precise Time Protocol (PTP), Precise Time Management (PTM), and Synchronous Ethernet create interesting possibilities and opportunities for real-time applications based on specialized microprocessors. One such opportunity is to incorporate precise time into a Nios® II processor custom NOP instruction, which we call NOP_WAIT. This custom instruction can use precise time information from an IEEE 1588 Grand Master or other source that allows a CPU or microcontroller to wait on a NOP instruction until it receives a predetermined precise time as shared by all devices in the system. Using this NOP_WAIT instruction, cameras, motors, energy sources, appliances, servers, networking schedulers, and so on can precisely synchronize across a network to perform coordinated operations, which improves a system's deterministic behavior.

- Wrote a white paper on how one can use IEEE1588 using an Intel NIOS II processor in a NOP instruction.
- Simple
  - Wait/Loop for a precise time
  - Proceeds to next instruction
  - Like TPAUSE, but no advance power savings modes
- Shows examples on how this may be used in the real world
- Dennis built the NIOS with this instruction and downloaded it to a MAX FPGA board.

https://www.intel.com/content/www/us/en/products/docs/programmable/nios-ii-white-paper.html

Time-Based Semaphore

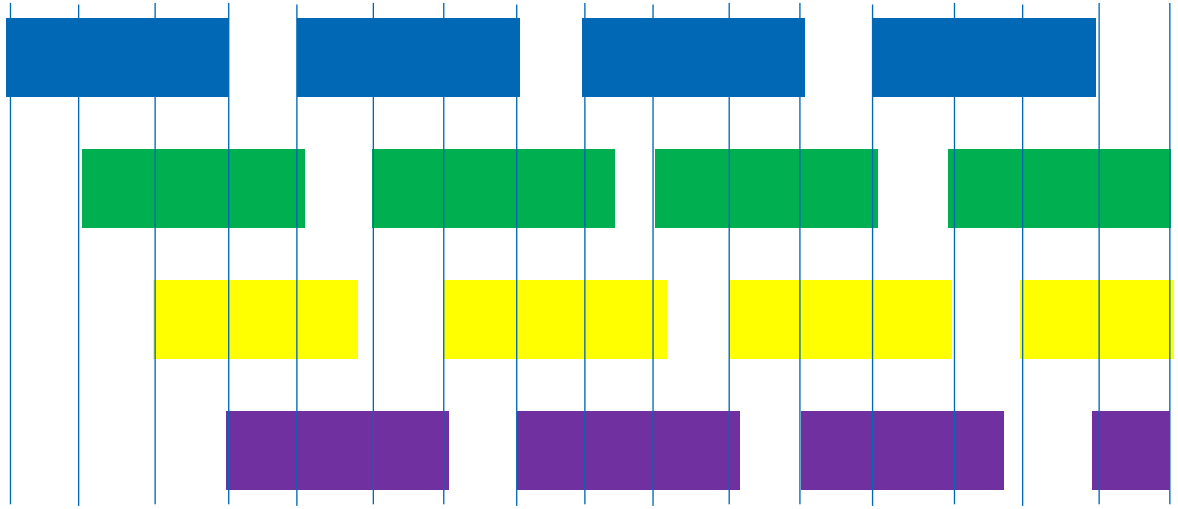
- Semaphore is a means of controlling accesses to a common resource
  - Locking mechanism
- With the No-Op Wait instruction, precise time can be part of the locking/control mechanism
  - Reduce traffic, reduce congestion
- Example: Freeway Metering Lights
  - The next car gets access to enter the freeway every 7-15 seconds.
  - Goal is to keep the cars on the freeway moving
  - Reduce congestion/accidents

TPAUSE for TDM-like Scarce Resource Usage

- Use TPAUSE to Delay Start of next Operation
- Use TPAUSE to stagger windows
- Some operations take longer than other
- All fit within the "Window"
- Very helpful if using the same operations / flow

  - Some scarce resources could be accessed at different times
  - Reducing risk of additional latency due to congestion/collisions at a shared resource
    - Shared resources include Network Port, Cache/Memory Interfaces, Encryption, Arithmetic Units, AI/ML and other Accelerators

Controlled Pacing of Instructions



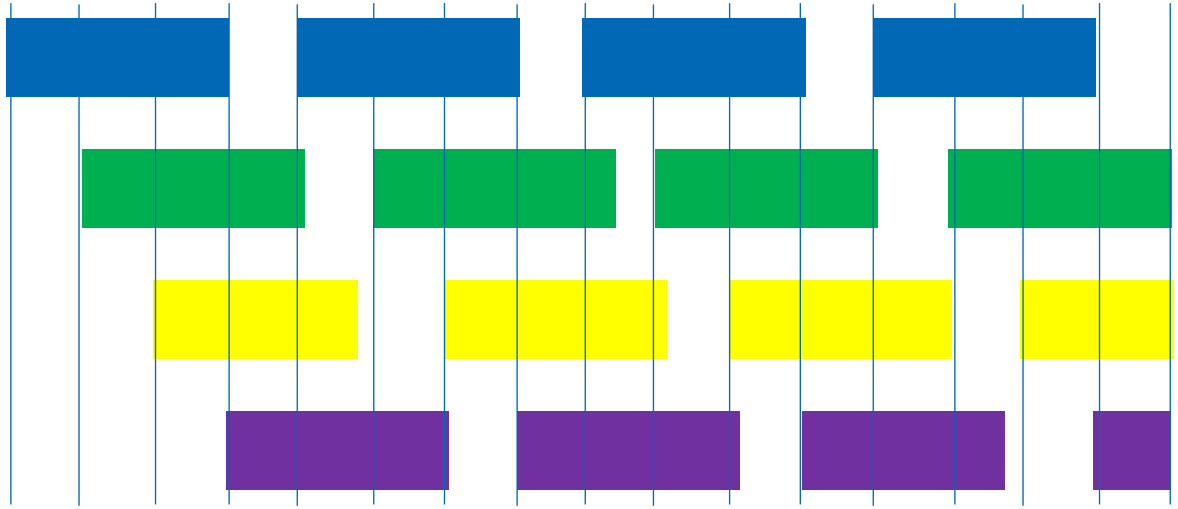
Controlled Pacing of Applications

Note: Out of order instructions could be executed if the TPAUSE instruction if not fenced properly.

TPAUSE and Performance

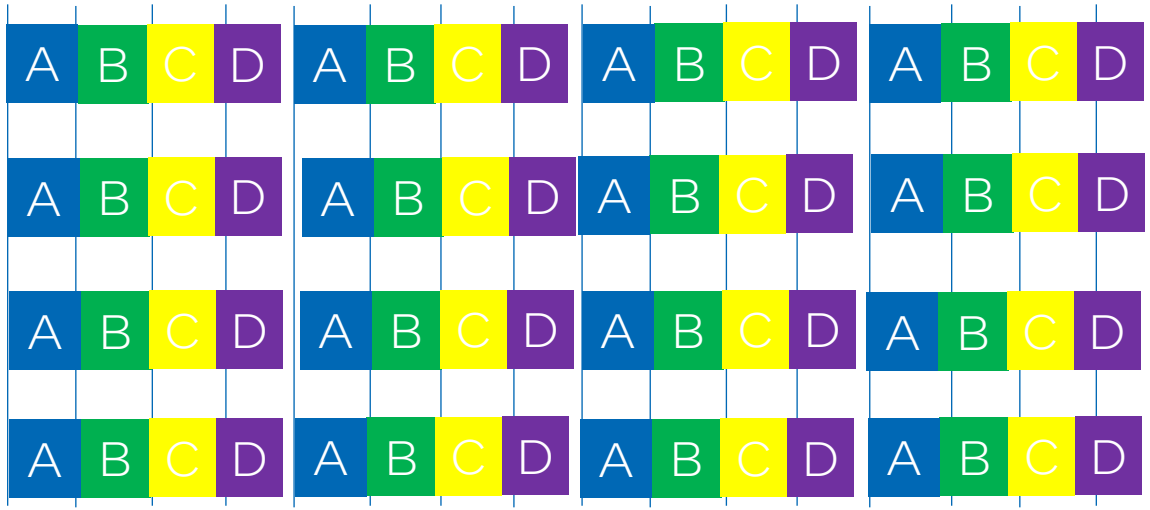
- Goal of 100 Million Operations per second using 4 processor
- Operation Example
  - Packet Processing
  - Database Operations
  - Internet of Things
  - AI/ML
- Each processor needs to handle:
  - 25 Million Operations per Second
  - One operation in 40ns
- Each processor could use TPAUSE to start the next operation

Controlled Pacing of Operations

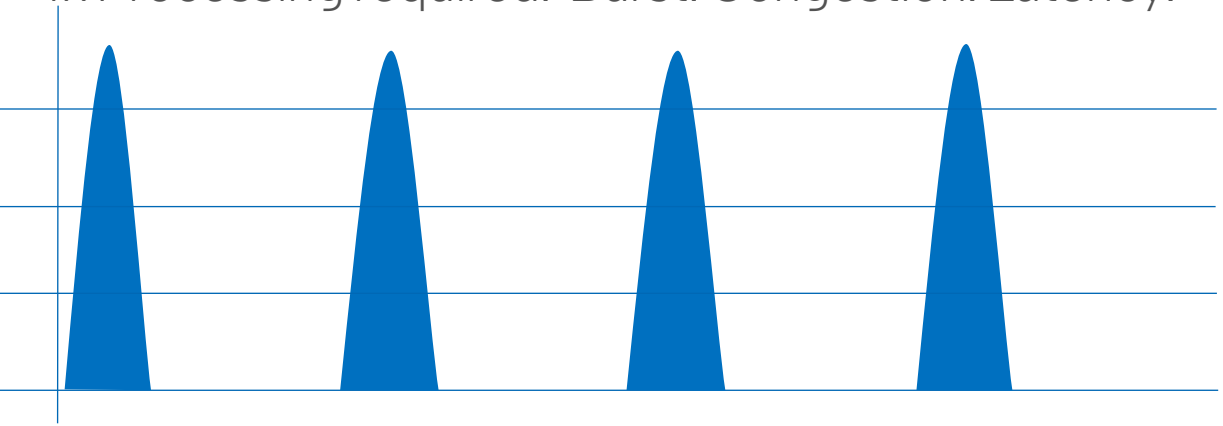


Processor 0 – 0, 40, 80, 120… ns
Processor 1 – 10, 50, 90, 130… ns
Processor 2 – 20, 60, 100, 140… ns
Processor 3 – 30, 70, 110, 150… ns

Worst Case (Unstaggered) Can Lead to Inefficiencies.



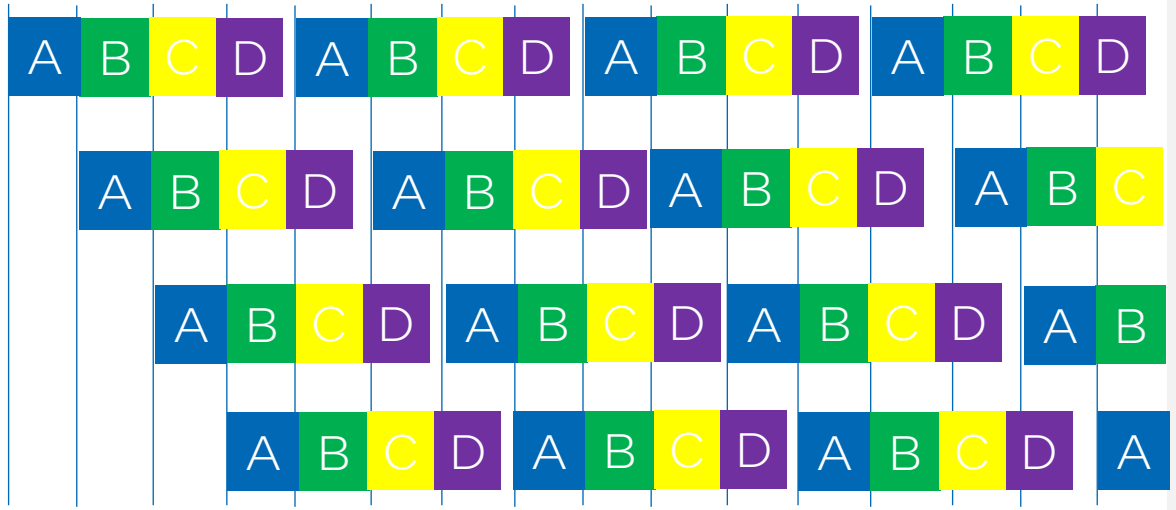
4x Processing required. Burst. Congestion. Latency.



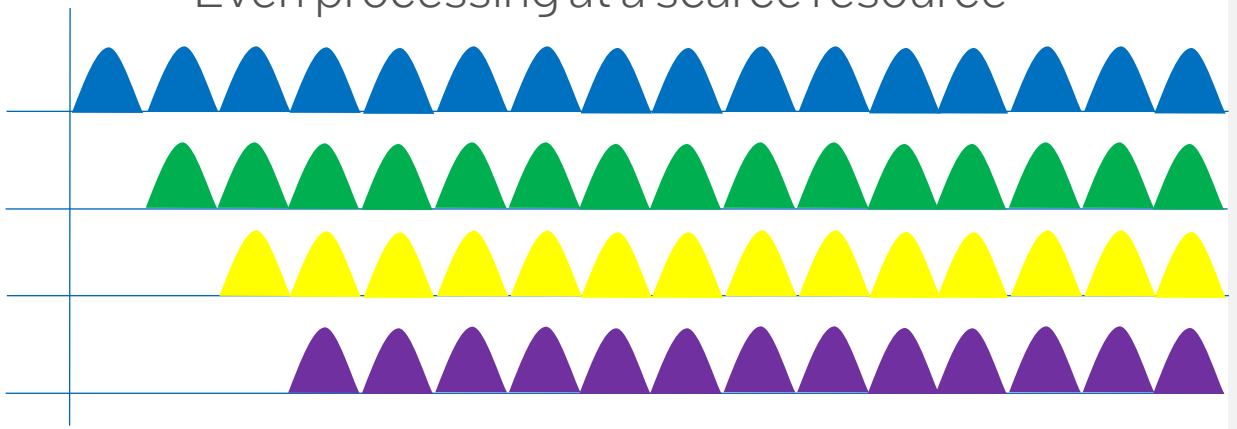
- Breaking every operation at a processor to use 4 resources:
  - Blue
  - Green
  - Yellow
  - Purple
- When Events are **NOT** Synchronized, the result can:
  - Be bursty
  - Cause contention
  - Add latency
- Figure Shows for Blue Resource
  - 4x processing required to handle burst
    - Imagine if there are 128 processors!!!
  - Processing is wider due to congestion
- Examples of shared resources includes Ingress Data, Egress Data, Semaphore, Accelerator, etc.

What if Ideally Staggered

- Little to no contention
- Less burstiness
- Interconnect to resource is controlled by one processor
- Latency is idea, as only one processor is accessing the resource at a time

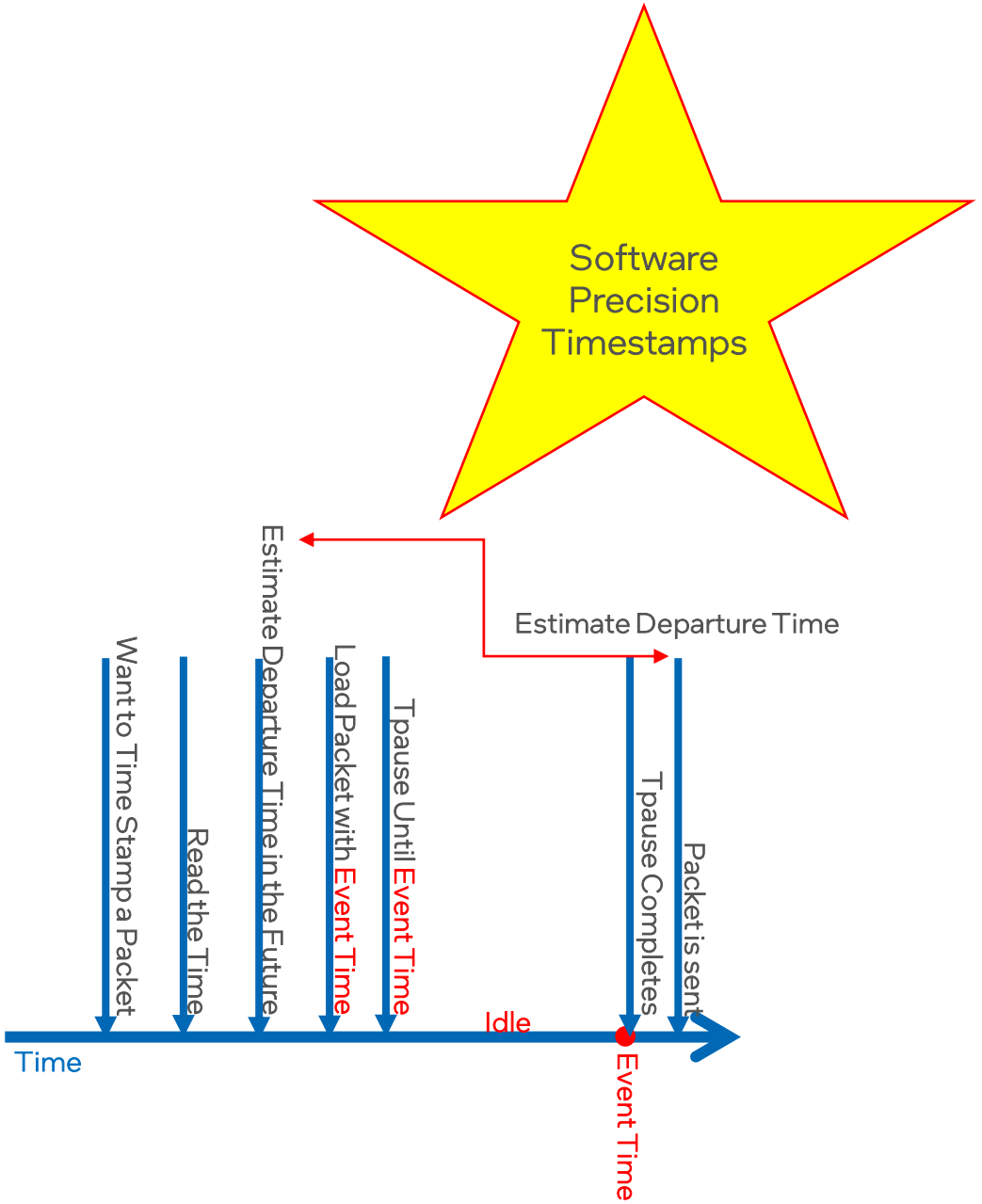


Even processing at a scarce resource



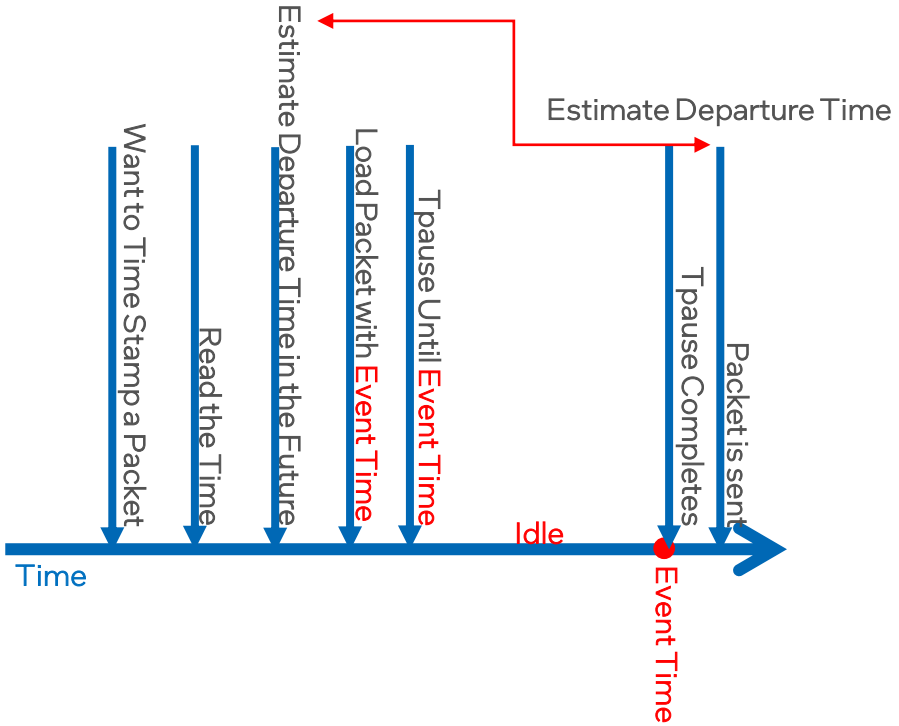
Precise Scheduling Resulting in Precise Software Timestamps

- Precisely time an event/resource
  - Schedule when a resource is accessed
  - Scheduling the start of a routine
  - For example:
    - Start of the routine to create a packet
    - Start of a descriptor transfer
- A pure SW solution adds jitter
  - Out of order instructions
  - Reading of the time
- Using hardware time in the NIC/IPU does not account for Time errors before the Hardware Timestamping
  - In-band Network Telemetry
  - Impossible to add counters in an ASIC that already exists…



Considerations When Using Tpause

- Out-of-Order Instructions
  - When will the Read of Time occur?
  - Fencing / MFENCE
- Run-to-Completion
  - No Preemption allowed
  - Don't want another thread to interrupt this process
- Idle Time
  - Long idle time can hurt processor performance
  - Affects on Cache
  - Look into Cache QoS/Locking
- Event time may not equal execution/departure time.



Love Story Continued…

- Computers have had a significant impact on our hero's life and everyone's lives.
- He would give back something to the computing industry to show his appreciation.
  - something to make computer performance better.
- Can precise time help increase overall performance of:
  - A computing device?
  - A data center or application?

- As time becomes more precise at the processor, new features can be implemented .

A Futurist's View

- Harvard Business Review published a Strategy paper:
- **How to Do Strategic Planning Like a Futurist by** Amy Webb
- She gives a framework for strategic planning, that seems to work well to envision the evolution of Precise Time Sync applications and use cases

- Reference:
- https://hbr.org/2019/07/how-to-do-strategic-planning-like-a-futurist

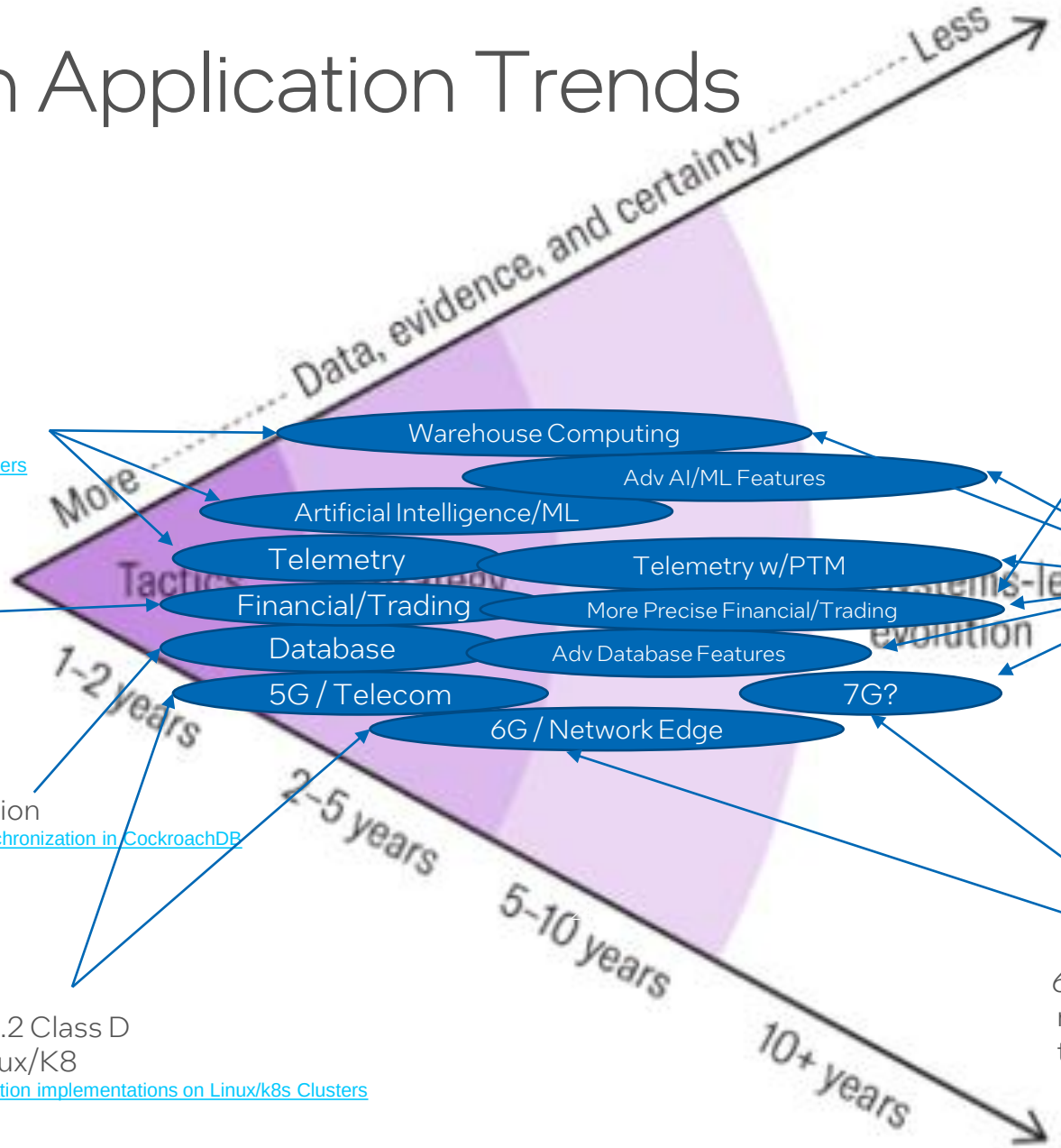
A Futurist's Framework for Strategic Planning

Instead of arbitrarily assigning goals on a quarterly or yearly time line, use a cone instead. First identify highly probable events for which there's already data or evidence, and then work outward. Each section of the cone is a strategic approach, and it encompasses the one before it until you reach major systems-level evolution at your company.

Source: Amy Webb, Future Today Institute

HBR

Precise Time in Application Trends



Applications See Benefits of Synchronization
Getting Precision Time Synchronization to End Users

MIFID-2
IEEE1588 in HFT

Databases Need Synchronization
Fearless Global Transactions using Clock Synchronization in CockroachDB

ITU-T 8273.2 Class D
Servers/Linux/K8
Open Synchronization implementations on Linux/k8s Clusters

Precision Financial Applications
CloudEx

Precision at the Application will improve
PCIe PTM: Timing in the Last Inch

6G, 7G and or Network Edge Applications may have tighter requirements, even at the processor

Disclaimer: Trends as seen by the author in April 2022. Different people/companies may have different views on the timelines and use cases. Reality may be different...

Many More Applications May See Significant Performance Improvements

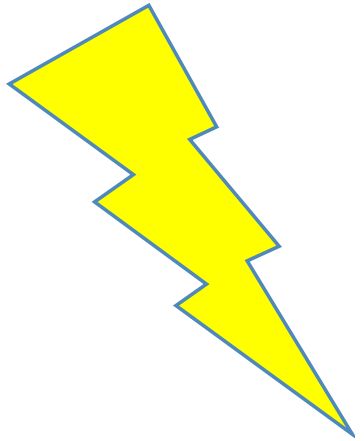
Databases
Telemetry
Internet of Things
Artificial Intelligence
Cloud Gaming
Warehouse Computing
CSP-ISP Co-Location
Network Function Virtualization

Scheduling
Service Mesh
Tail Latency
Software Defined Networking
Financial
Alternate Reality
Packet Pacing
ITU-T 8273
DC like a PC

Machine Learning
High Frequency Trading
Cloud Service
Virtual Reality
Microservices
High Performance Computing
Resource Sharing and Alignment
Network Edge

Mission to Make Computer Performance Better

- You can help our hero on his mission:
  - Present your precise time use cases and data in OCP-TAP so others can learn and reference them
  - Share your use case and data directly with Intel and other vendors, so that they can be analyzed, and if appropriate, implemented.
- Neither OCP-TAP nor Intel can fix what it can't see and understand…

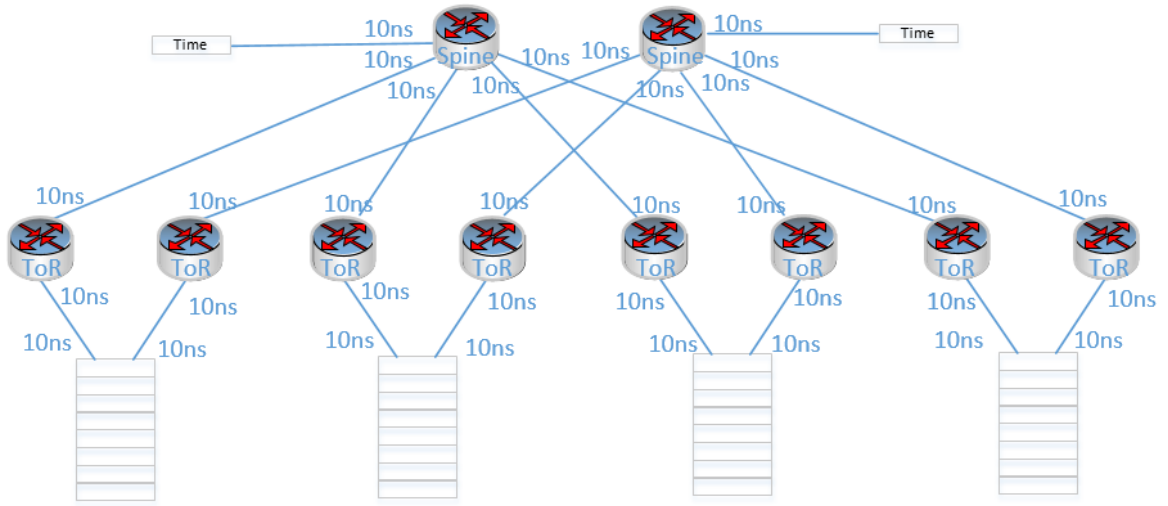


Clock image taken from: https://www.opencompute.org/wiki/Time_Appliances_Project

Mission to Make Computer Performance Better

- Questions to ask:
  - Can inefficiencies from being proactive, be reduced with more reactive techniques.
  - Can your bottlenecks be reduced or removed by precise time synchronization?
  - What Proof of Concepts can easily be done?
  - Let's look at some ideas.

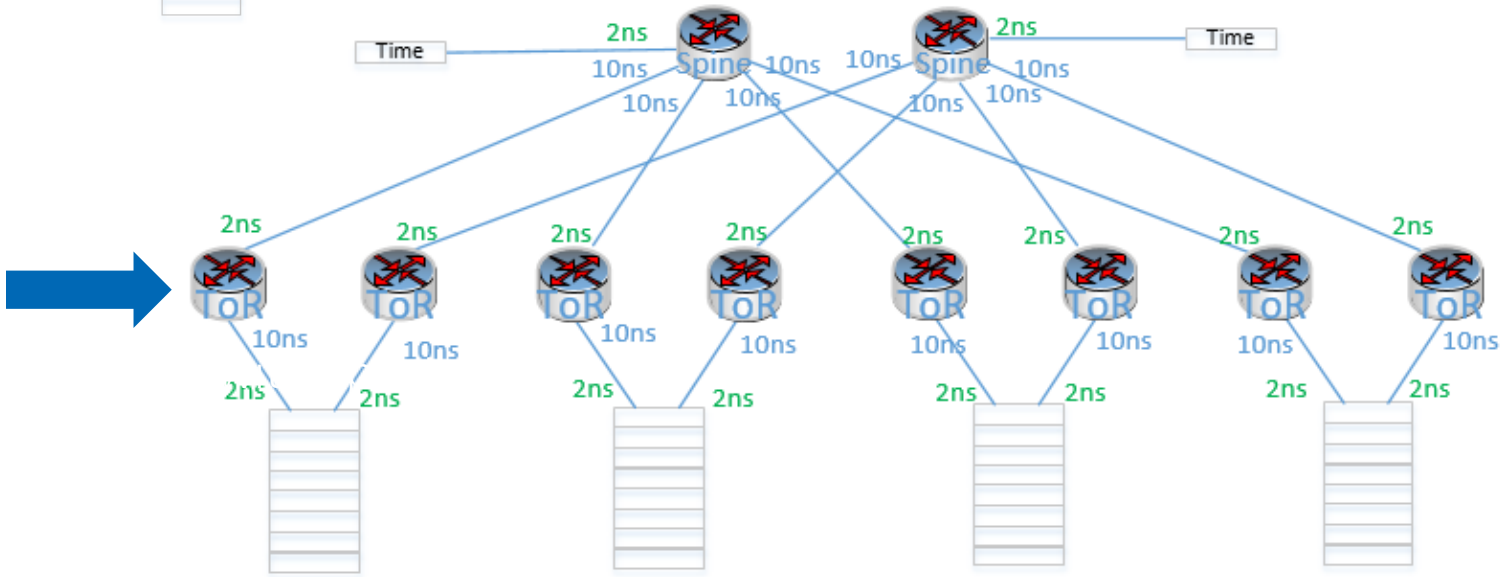
Ideas for the Future: Precise Brownfield Data Center



Could we make Optical Modules that implement White Rabbit technologies and place them in brownfield equipment

Shows about 90ns of jitter between servers

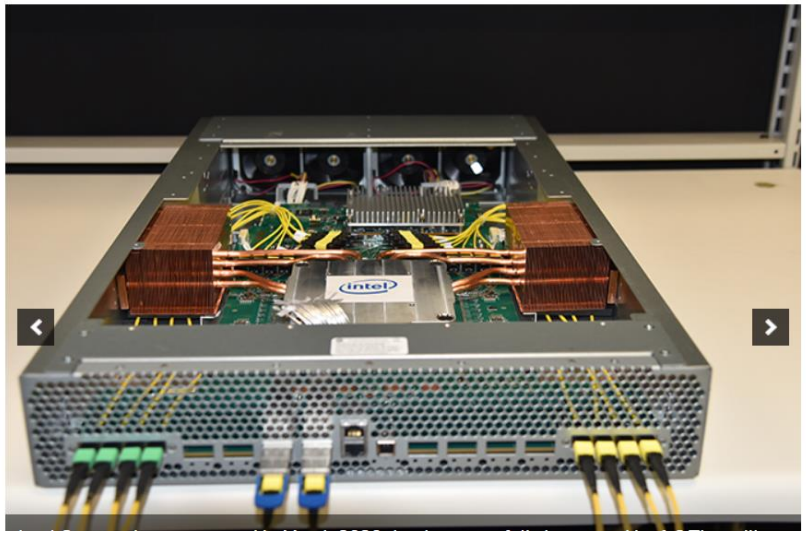
Simply replacing the uplinks could reduce the jitter by 40+%



Disclaimer: These are just ideas. They may or may not ever come to life in industry

Consider Latency

Intel Demonstrates Industry-First Co-Packaged Optics Ethernet Switch



BY LUIZ BARROSO, MIKE MARTY, DAVID PATTERSON, AND PARTHASARATHY RANGANATHAN

Attack of the Killer Microseconds

FEC Killed The Cut-Through Switch

Omer S. Sella
University of Cambridge
omer.sella@cl.cam.ac.uk

Andrew W. Moore
University of Cambridge
andrew.moore@cl.cam.ac.uk

Noa Zilberman
University of Cambridge
noa.zilberman@cl.cam.ac.uk

ABSTRACT

Latency penalty in Ethernet links beyond 10Gb/s is due to forward error correction (FEC) blocks. In the worst case a single-hop penalty approaches the latency of an entire cut-through switch. Latency jitter is also introduced, making latency prediction harder, with large peak to peak variance. These factors stretch the tail of latency distribution in Rack-scale systems and Data Centers, which in turn degrades performance of distributed applications. We analyse the underlying mechanisms, calculate lower bounds and propose a different approach that would reduce the penalty, allow control over latency and feedback for application level optimisation.

CCS CONCEPTS

• **Networks** → **Physical links**; *Error detection and error correction*; *Network control algorithms*;

1 INTRODUCTION

Latency has been long known to have an adverse effect on systems, from the annoyance users feel when a website is slow to load, to application performance degradation [27]. Patterson *et al.* [20] observed over a decade ago that bandwidth improvements are made at the expense of latency, and in particular that the rate of network latency improvement stagnates next to the rate of bandwidth improvement.

Over the last decade, network bandwidth has improved from

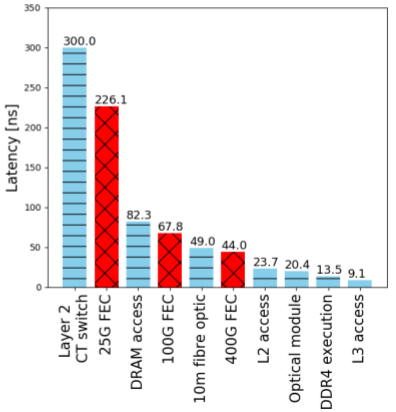
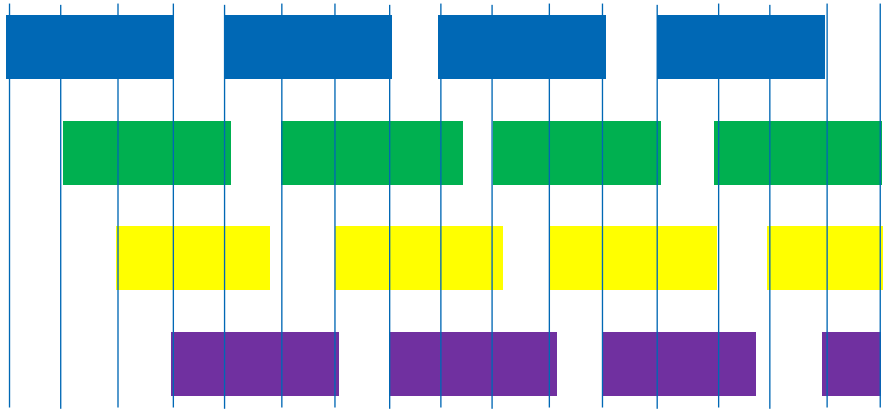


Figure 1: Comparing the scale of latency in components of networked-systems. FEC induced latency is marked in red.

Attack of the Killer Microseconds **By Luiz Barroso, Mike Marty, David Patterson, Parthasarathy Ranganathan; Communications of the ACM, April 2017, Vol. 60 No. 4, Pages 48-54**
NEAT '18: Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies August 2018 Pages 15–20https://doi.org/10.1145/3229574.3229577
https://newsroom.intel.com/news/intel-demonstrates-industry-first-co-packaged-optics-ethernet-switch/#gs.w0015b

Advantages of Precise Time-Based Scheduling, Semaphores, and Pacing



Intel Media Transport Library SMPTE ST 2110

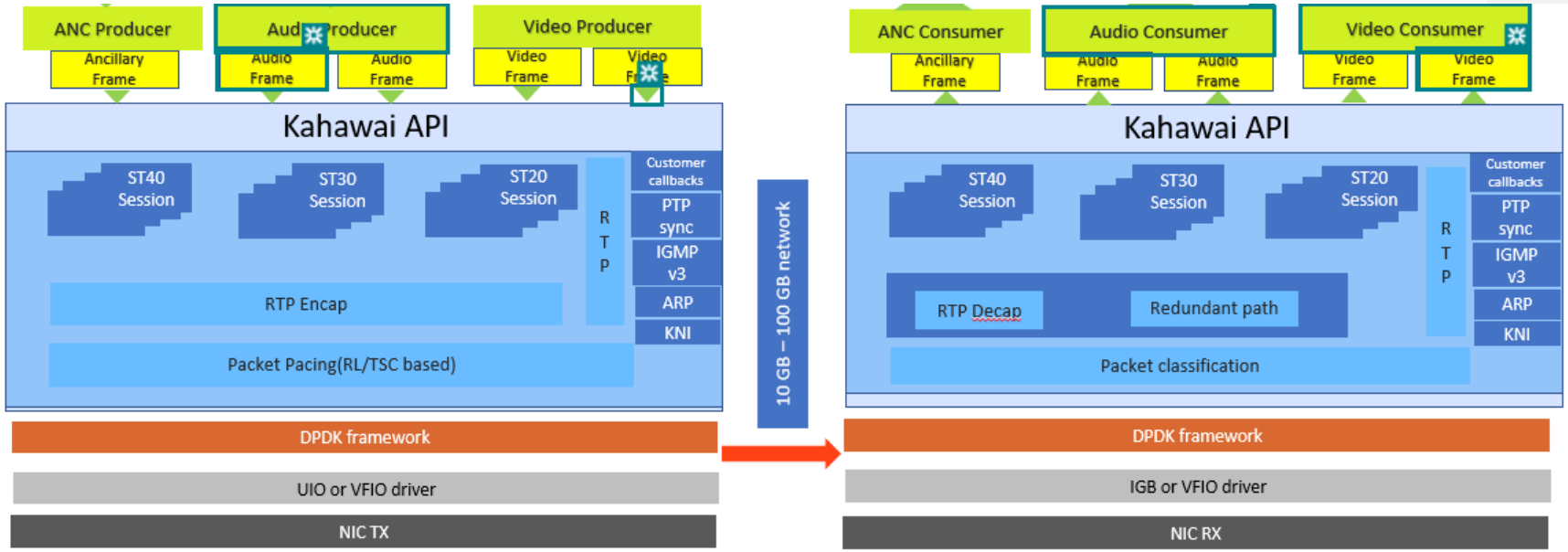


Image from Raul Diaz: Kahawai: Intel SMPTE ST 2110 Library

Resources

- Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D and 4

https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html

- Using IEEE-1588 Precise Time Protocol to Create a NOP_WAIT Instruction for the NIOS II Processor.

https://www.intel.com/content/www/us/en/products/docs/programmable/nios-ii-white-paper.html

- Intel Ethernet 700 Series Precision Time Protocol

https://www.intel.com/content/www/us/en/products/docs/network-io/ethernet/network-adapters/ethernet-adapter-xxv710-da2t-brief.html

- High-Precision Time Synchronization for 5G RAN Intel Video

https://www.intel.com/content/www/us/en/products/docs/network-io/ethernet/network-adapters/ethernet-network-adapter-e810-xxvda4t-video.html

Conclusion

- Hope you enjoyed the story and learned a few things about
  - Precise Time today
  - Processor's Use of Precise Time
  - Trends in Precise Time
  - Ideas for the future
- Hopefully, you can help our hero by sharing your use cases and ideas at OCP-TAP, so we call all learn and make compute and application performance better.

What wonderful thing can you do with precise time?

Thank you!

Q&A


intel