Parallelism Analysis of Prominent Desktop Applications – An 18 Year Perspective

Siying Feng, Subhankar Pal, Yichen Yang, Ronald G. Dreslinski
University of Michigan
Ann Arbor, Michigan, USA

If you build it,
they will come

If you build it,
will they ~~will~~ come?

Prior Works Studying Parallelism in Desktops

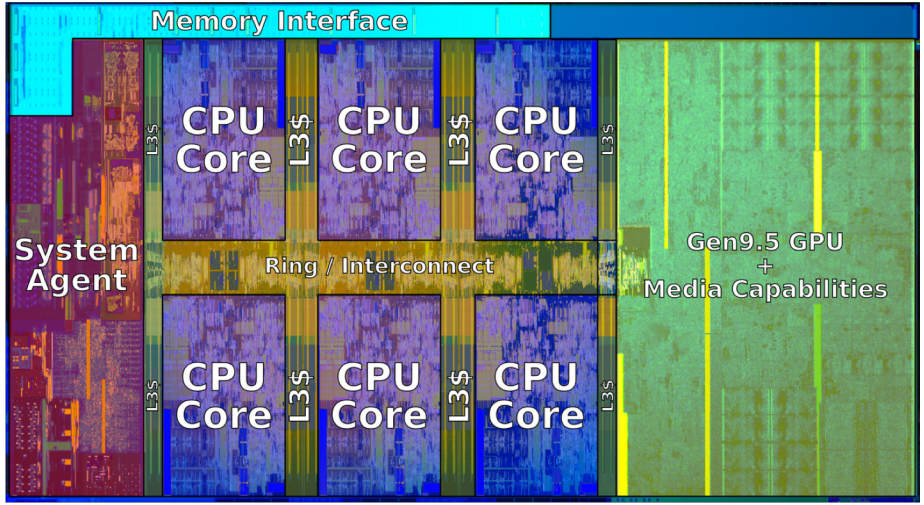
- Flautner *et al*. [ASPLOS 2000] evaluated a symmetric multiprocessor (SMP) with 4 cores
  - Most applications had less than 2 threads running concurrently on average
  - A second processor improved the responsiveness of interactive applications

- Blake *et al*. [ISCA 2010] studied an 8-core processor with Simultaneous Multi-Threading
  - 2-3 processors cores were still more than sufficient for most applications
  - GPU was mostly underutilized

Evolution in Hardware

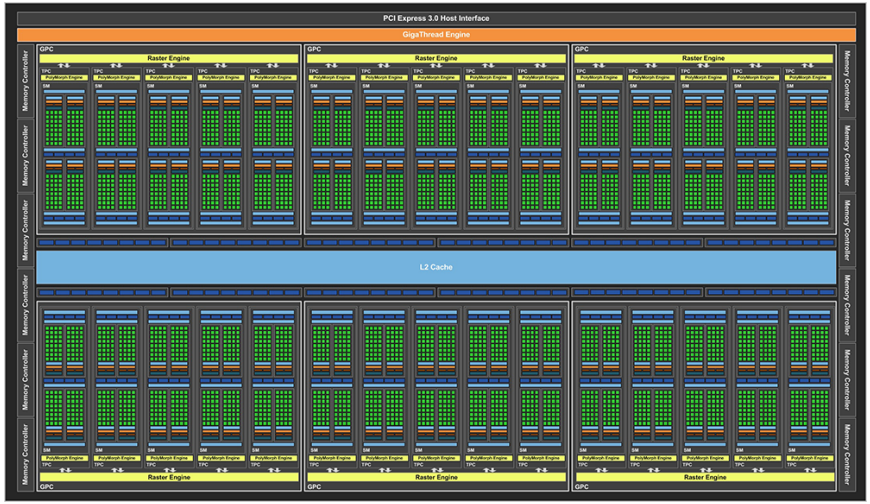
- Desktop machines have evolved into a combination of CPU, GPU and fixed-function hardware

Intel Coffee Lake



*https://en.wikichip.org/wiki/intel/core_i7/i7-8700k

NVIDIA GTX 1080 Ti



*https://international.download.nvidia.com/geforce-com/international/pdfs/GeForce_GTX_1080_Whitepaper_FINAL.pdf

Evolution in Software

- Virtual Reality

- Cryptocurrency Mining

- Personal Assistants

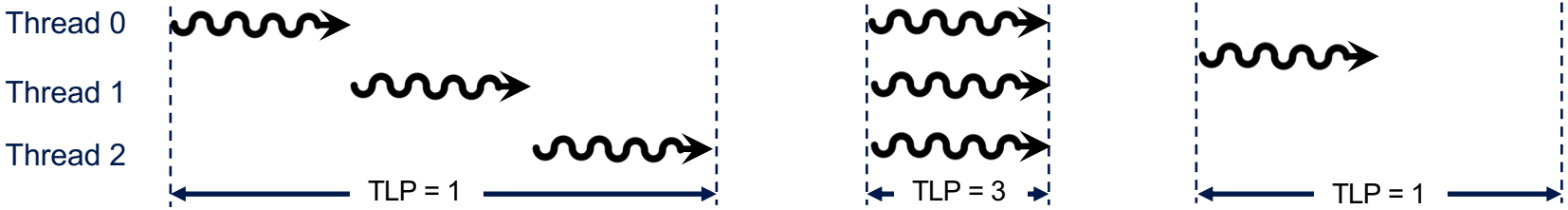
Source: google image

Outline

- Introduction
- Methodology
- Benchmarks
- Results
- Key Takeaways
- Conclusion

Metrics to Measure Parallelism

- **Thread-Level Parallelism (TLP)**: the amount of thread concurrency with idle time factored out



- **GPU utilization**: the percentage of time over the past sampling period during which one or more kernels were executing on the GPU
- **Key Takeaways** from the metrics
  - End users: guidelines on the amount of hardware to invest
  - Software developers: potential suggestions to further exploit parallelism

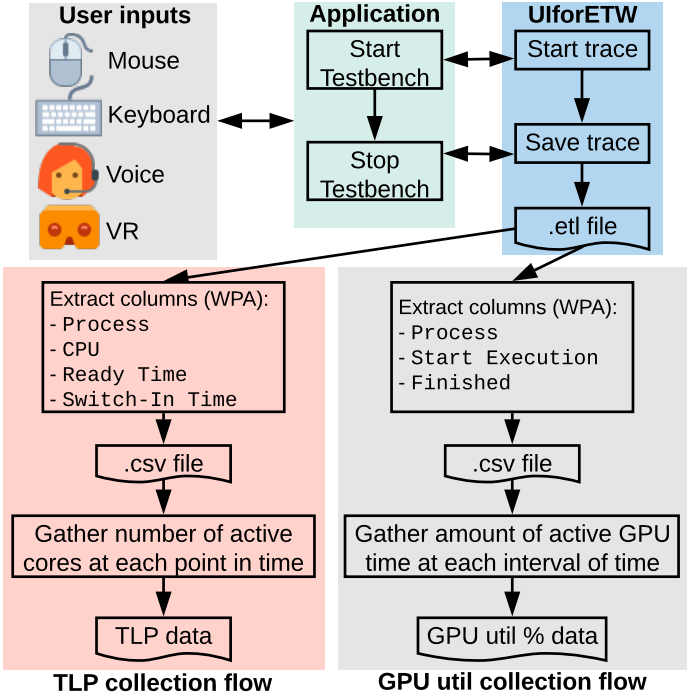
Methodology

- System setup

CPU	Intel Core i7-8700K, 3.70 – 4.70 GHz, 6 cores / 12 threads
Graphics	NVIDIA GTX 1080 Ti, 1481 MHz, 3584 CUDA cores
RAM	64 GB (16 GB x 4) DDR4 @ 3200 MHz
Storage	2 TB (1 TB x 2) PCIe NVMe SSD
OS	Windows 10

- Trace collection tool
  - Event Tracing for Windows (ETW)
  - Windows Performance Analyzer (WPA)

- Trace collection workflow



Testing Strategy

- Goal: maintain consistency among different test iterations
- Testing automation:
  - Tool: AutoIt
    - An automation language that simulate keystrokes and mouse activities at a user-specified time
  - Automation scripts create no significant distortion to results
- Manual testing:
  - Applications accepting user inputs that cannot be reproduced by automation tools
  - Personal Assistant:
    - Fixed sequence of requests and questions
    - Strict timing constraints
    - Same voice
  - VR Gaming:
    - Campaign mode/ fixed checkpoints to create similar scenarios

Benchmarks

- A suite of common desktop applications based on their popularity among users
- Maximized overlap with the benchmarks used in prior works

*https://github.com/SiyingFeng1995/Desktop_Parallelism_Analysis_2018

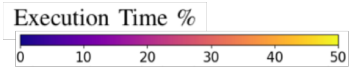
Outline

- Introduction
- Methodology
- Benchmarks
- Results
- Key Takeaways
- Conclusion

Overview

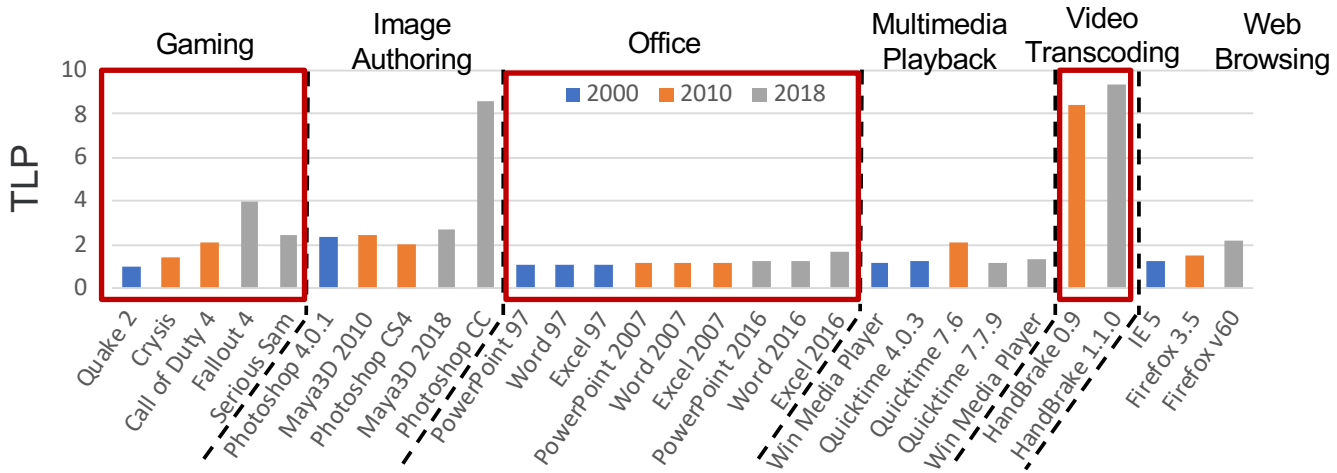
Category	Application	Execution Time (%)													TLP		GPU Util. (%)	
		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Avg.	σ	Avg.	σ
Image Authoring	Adobe Photoshop CC														8.6	0.10	1.6	0.2
	Autodesk Maya 3D 2019														2.7	0.08	9.9	0.2
	Autodesk AutoCAD LT														1.2	0.02	9.0	0.9
Office	Adobe Acrobat Pro DC														1.3	0.00	0.0	0.0
	Microsoft Excel 2016														2.1	0.03	2.1	0.0
	Microsoft PowerPoint 2016														1.2	0.01	4.0	0.1
	Microsoft Word 2016														1.3	0.01	1.7	0.0
	Microsoft Outlook 2016														1.3	0.05	2.5	0.2
Multimedia Playback	QuickTime Player 7.7.9														1.1	0.02	16.4	0.1
	Windows Media Player 12.0														1.3	0.19	16.1	0.0
	VLC Media Player 3.0.3														1.8	0.18	15.7	0.9
Video Authoring	CyberLink PowerDirector v16														4.3	0.03	6.3	0.1
	Adobe Premier Pro CC														1.8	0.02	0.6	0.0
Video Transcoding	HandBrake 1.1.0														9.4	0.04	0.4	0.0
	WinX HD Video Converter 5.12.1														9.2	0.02	13.6	0.1
Web Browsing	Firefox v60														2.2	0.13	8.6	0.5
	Chrome v60														2.2	0.13	5.1	0.6
	Edge 42.17134.1.0														2.0	0.02	4.0	0.2
VR Gaming	Arizona Sunshine 1.5.11046														3.4	0.23	68.2	0.8
	Fallout 4 VR 1.2														4.0	0.15	84.9	1.7
	RAW Data 1.1.0														2.6	0.13	90.9	1.4
	Serious Sam VR BFE 341433														2.4	0.10	72.2	1.7
	Space Pirate Trainer 1.01														2.7	0.11	61.6	0.5
	Project CARS 2 1.7.1.0														3.8	0.16	80.2	2.1
Cryptocurrency Mining	Bitcoin Miner 1.54.0														5.4	0.15	98.9	1.1
	EasyMiner v.0.87														11.9	0.02	96.1	0.4
	PhoenixMiner 3.0c														1.0	0.01	100.0	0.1
	Windows Ethereum Miner 1.5.27														1.0	0.01	99.7	0.1
Personal Assistant	Cortana														1.4	0.04	2.7	0.0
	Braina 1.43														1.1	0.02	0.0	0.0

- **The average TLP across all benchmarks is 3.1**
- **The GPU is mostly underutilized**



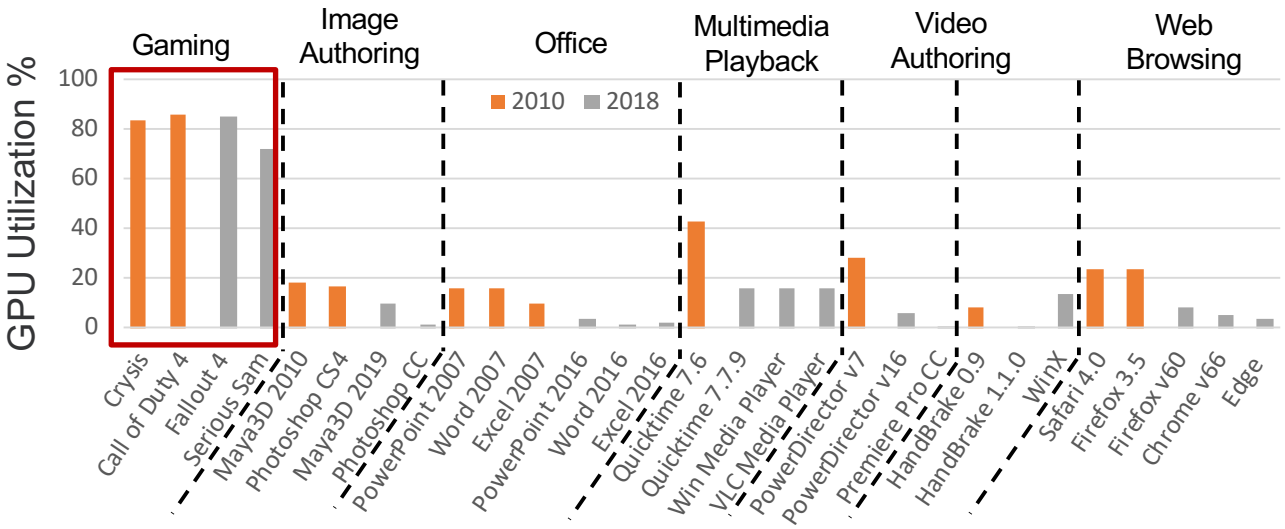
An 18-year Perspective

- **Most applications present higher TLP**
  - The escalation in computation complexity of VR games results in a noticeable rise in TLP
  - Applications showing high concurrency in prior works exhibit a further increase in TLP
  - Applications with little growth in average TLP display progress in instantaneous TLP



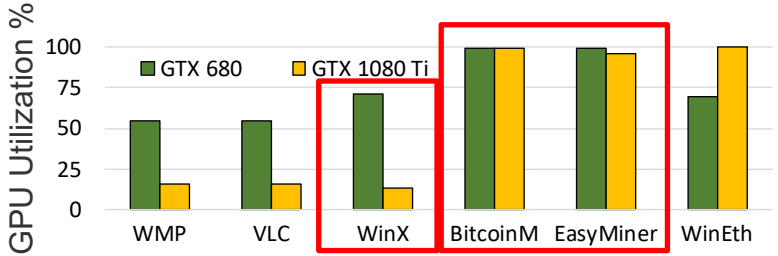
An 18-year Perspective

- **All benchmarks except for VR gaming show lower GPU utilization**
  - This can be attributed to advancements in GPU hardware



GPU Utilization

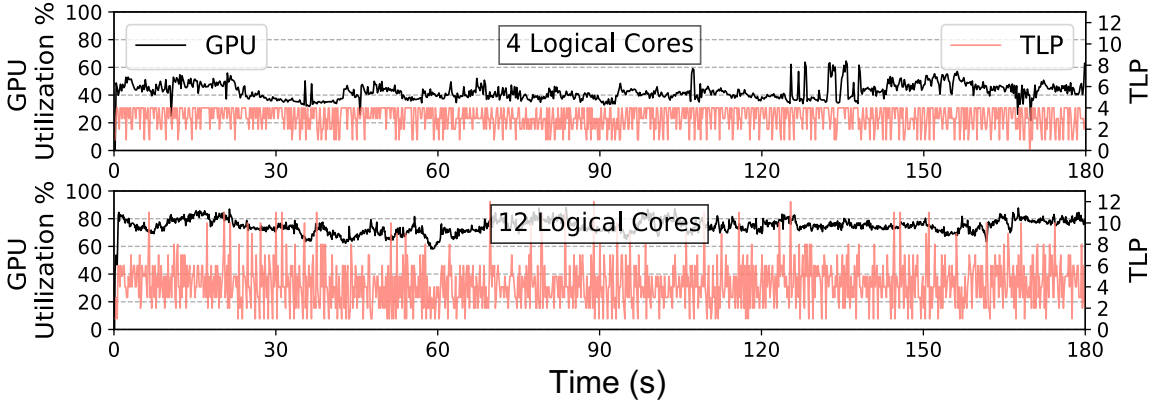
- GPU is underutilized for most applications
  - E.g. WinX achieves an average GPU utilization of 13.6%
- Certain applications take full advantage of GPU
  - E.g. Crypto Miners have GPU utilization over 95%
- **A lower-end GPU achieves *similar* performance but a higher utilization**

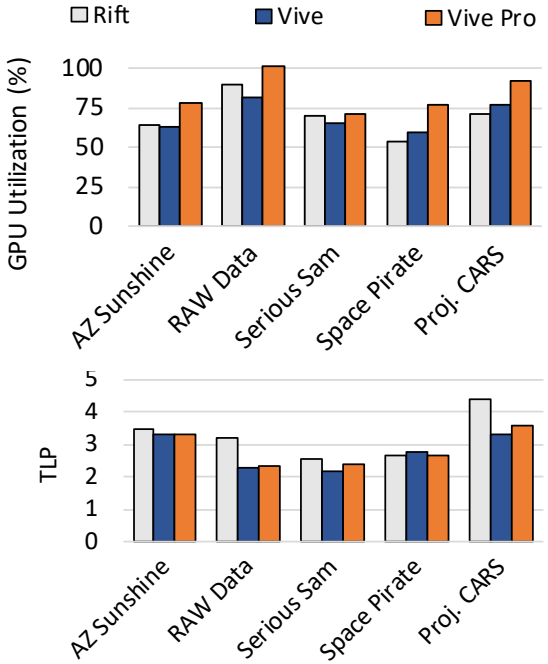


Category	Application	GPU Util. (%) Avg.	σ
Image Authoring	Adobe Photoshop CC	1.6	0.2
	Autodesk Maya 3D 2019	9.9	0.2
	Autodesk AutoCAD LT	9.0	0.9
Office	Adobe Acrobat Pro DC	0.0	0.0
	Microsoft Excel 2016	2.1	0.0
	Microsoft PowerPoint 2016	4.0	0.1
	Microsoft Word 2016	1.7	0.0
	Microsoft Outlook 2016	2.5	0.2
Multimedia Playback	QuickTime Player 7.7.9	16.4	0.1
	Windows Media Player 12.0	16.1	0.0
	VLC Media Player 3.0.3	15.7	0.9
Video Authoring	CyberLink PowerDirector v16	6.3	0.1
	Adobe Premier Pro CC	0.6	0.0
Video Transcoding	HandBrake 1.1.0	0.4	0.0
	WinX HD Video Converter 5.12.1	13.6	0.1
Web Browsing	Firefox v60	8.6	0.5
	Chrome v60	5.1	0.6
	Edge 42.17134.1.0	4.0	0.2
VR Gaming	Arizona Sunshine 1.5.11046	68.2	0.8
	Fallout 4 VR 1.2	84.9	1.7
	RAW Data 1.1.0	90.9	1.4
	Serious Sam VR BFE 341433	72.2	1.7
	Space Pirate Trainer 1.01	61.6	0.5
	Project CARS 2 1.7.1.0	80.2	2.1
Cryptocurrency Mining	Bitcoin Miner 1.54.0	98.9	1.1
	EasyMiner v.0.87	96.1	0.4
	PhoenixMiner 3.0c	100.0	0.1
	Windows Ethereum Miner 1.5.27	99.7	0.1
Personal Assistant	Cortana	2.7	0.0
	Braina 1.43	0.0	0.0

Virtual Reality (VR) Gaming

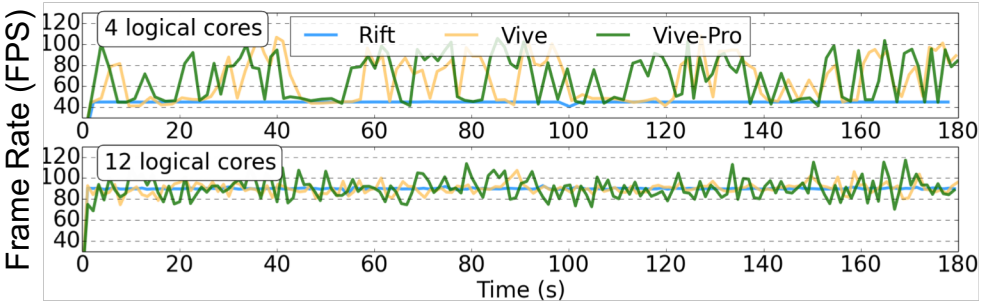
- **GPU utilization generally correlates with resolution**
  - Vive/Rift: 2160 x 1200
  - Vive Pro: 2880 x 1600
- **Instantaneous TLP reaches maximum in burst** but mostly remains between 2 - 6





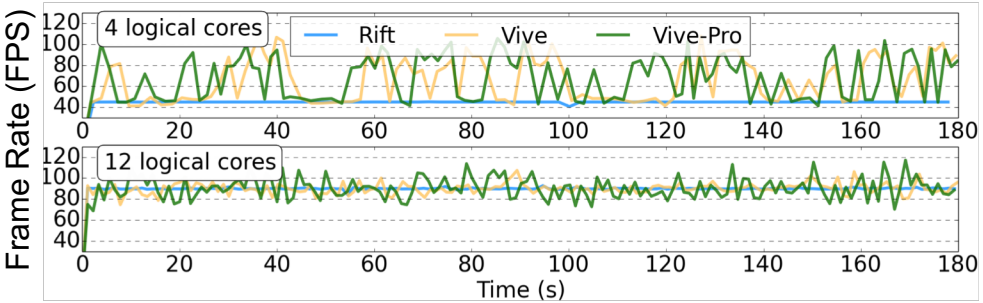
Virtual Reality (VR) Gaming

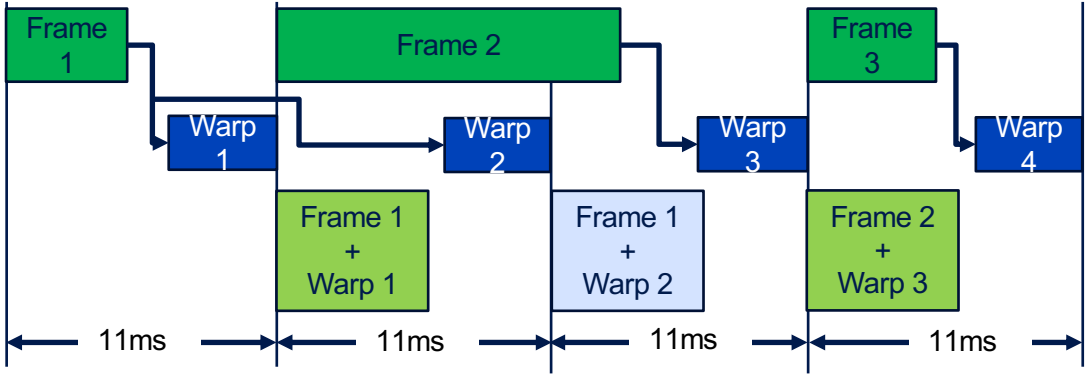
- Frame rate decreases if hardware resource is compromised
  - Vive (Pro): **asynchronous reprojection**
    - improve user experience
  - Rift: **asynchronous spacewarp**
    - lower hardware requirements



Virtual Reality (VR) Gaming

- Frame rate decreases if hardware resource is compromised
  - Vive (Pro): **asynchronous reprojection**
    - improve user experience
  - Rift: **asynchronous spacewarp**
    - lower hardware requirements

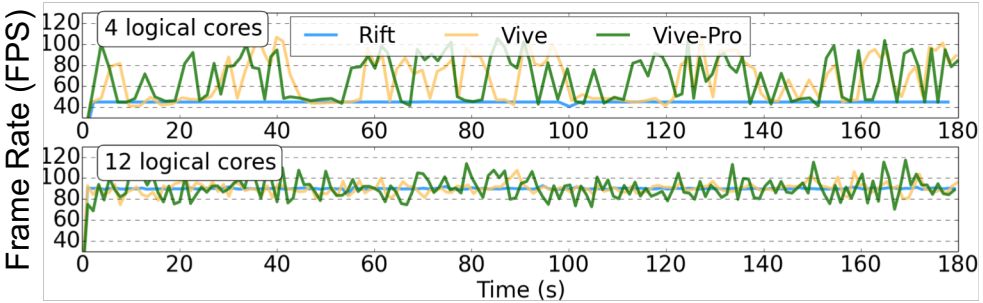


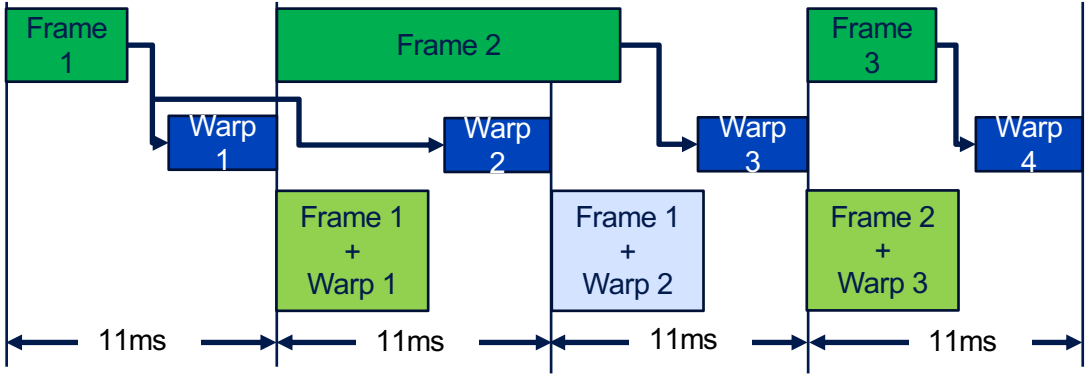


Asynchronous Reprojection (Vive / Vive Pro)

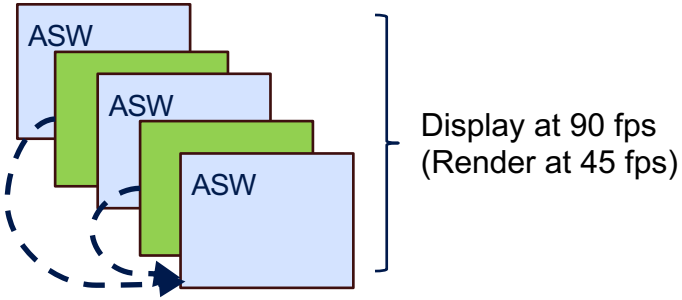
Virtual Reality (VR) Gaming

- Frame rate decreases if hardware resource is compromised
  - Vive (Pro): **asynchronous reprojection**
    - improve user experience
  - Rift: **asynchronous spacewarp**
    - lower hardware requirements





Asynchronous Reprojection (Vive / Vive Pro)



Asynchronous Spacewarp (Rift)

More Result Analyses in the Paper

- Core Scaling
- Simultaneous Multi-Threading (SMT)
- GPU Offloading
- Web Browsing

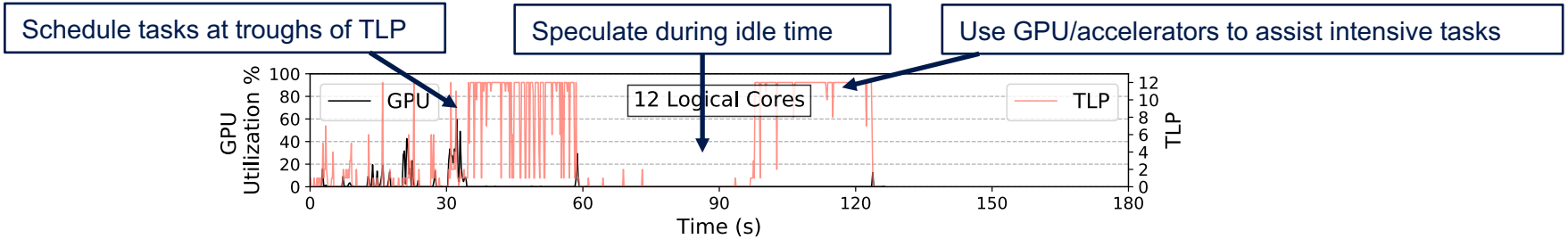
Outline

- Introduction
- Motivation
- Methodology
- Benchmarks
- Results
- Key Takeaways
- Conclusion

Key Takeaways

- End User
  - TLP and GPU utilization work as guidelines on the amount of hardware to invest
  - Employing many processor cores and a high-end GPU does not always bring benefits

- Software developers
  - TLP and GPU utilization show potential areas for software to further exploit parallelism
    - OS can schedule other tasks at troughs in TLP during execution to achieve optimized utilization
    - Idle time can be used for future activity predication and speculative execution
    - GPU and dedicated accelerators can be exploited to assist compute-intensive tasks when possible



Key Takeaways & Conclusion

- Appreciable progress has been made in exploiting parallelism
  - With sufficient scope to further improve hardware utilization
  - Emerging applications demonstrate good hardware utilization
- Software has improved to take advantage of the parallelism available in CPU
  - Noticeable increases were seen in many applications
  - Applications with non-significant average TLP change show improvements in instantaneous TLP
- GPU is underutilized under most circumstances
  - Improvements in hardware resources outpaced that in parallelism harnessed by software
  - Emerging applications exhibits high GPU utilization

Questions?