Low-Power Parallel Processing on GPUs
Looking Back (and Looking Forward)

Ben Juurlink

TU Berlin

LPGPU

LPGPU Factsheet

Contract number:	288653
Project coordinator:	Ben Juurlink, TU Berlin
EU project officer:	Markus Korn
Community contribution:	€ 2,820,000
Duration:	1 September 2011 – 31 October 2014
Project website:	http://lpgpu.org
Consortium partners (beneficiaries):	TU Berlin, codeplay, Geomerics, AiGameDev.com, Think Silicon, Uppsala Universitet

Agenda

- Intro
  - What?
  - Why?
  - How?
- LPGPU power simulation framework
- Video decoding on GPUs
- TSIMD GPU architecture
- Conclusions & outlook

What? – Why? – How?

- What is the problem?
  - GPUs not flexible enough to efficiently execute appealing applications
  - They consume too much power
  - There's little / no tool support for estimating and reducing power

Src: notebookcheck.com

Src: bottlerocketapps.com

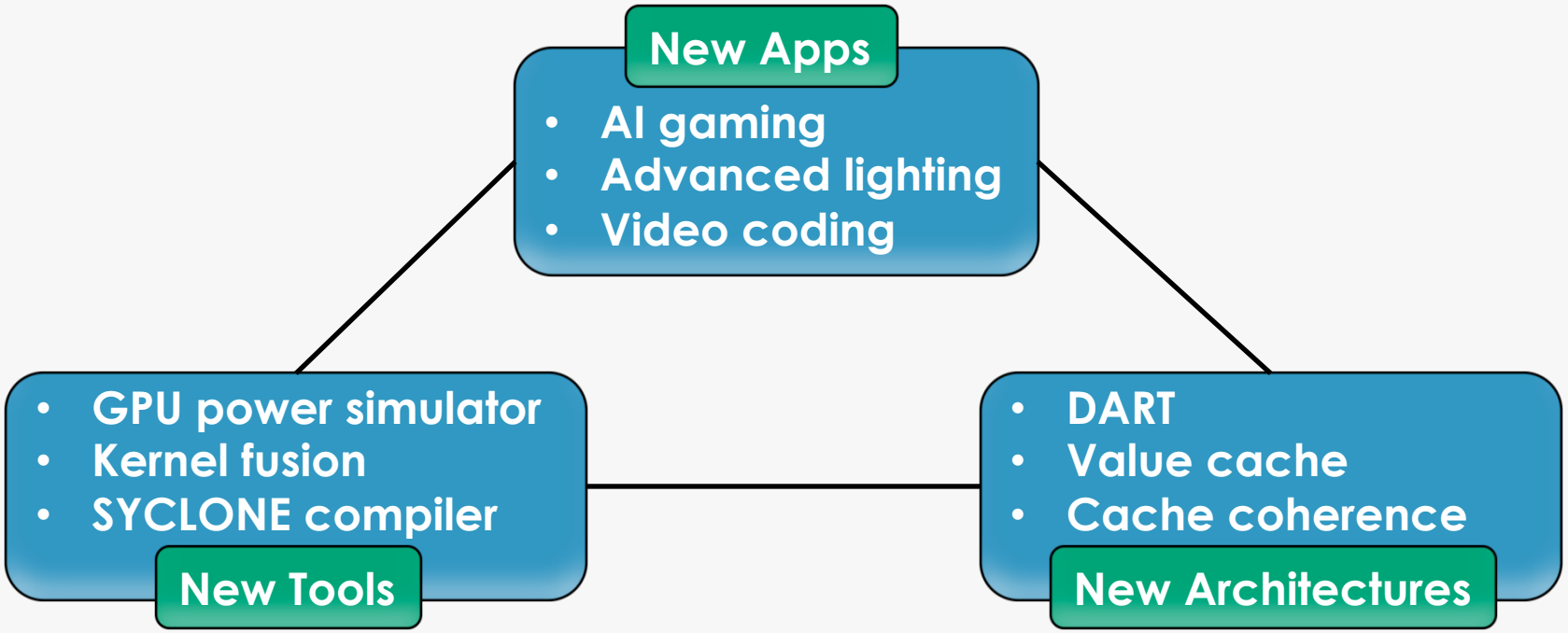
LPGPU

What? – Why? – How?

- Why is this a problem for European industry?
  - Graphics key software industry in Europe
  - Mobile devices now dominant form factor for computing worldwide
  - European companies lead design of mobile phone CPUs and GPUs, and are world leaders in video-games technology
  - Companies need to make large investments in R&D for graphics; vital that they have reliable information

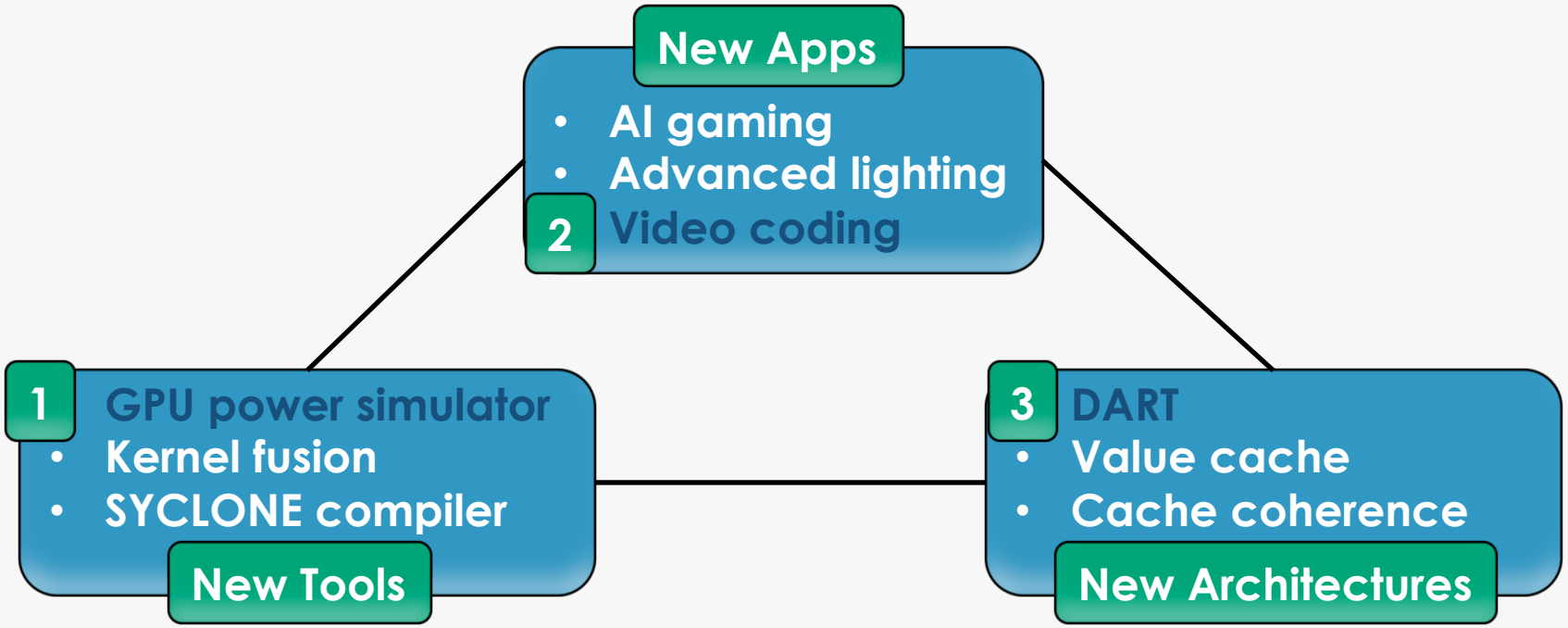
What? – Why? – How?
LPGPU Approach

- How does the LPGPU project try to solve these problems?
  - By porting apps to & developing apps for GPUs
  - By developing architectural techniques to reduce power consumption
  - By developing toolset to estimate and reduce power consumption



What? – Why? – How?
LPGPU Approach

- How does the LPGPU project try to solve these problems?
  - By porting apps to & developing apps for GPUs
  - By developing architectural techniques to reduce power consumption
  - By developing toolset to estimate and reduce power consumption



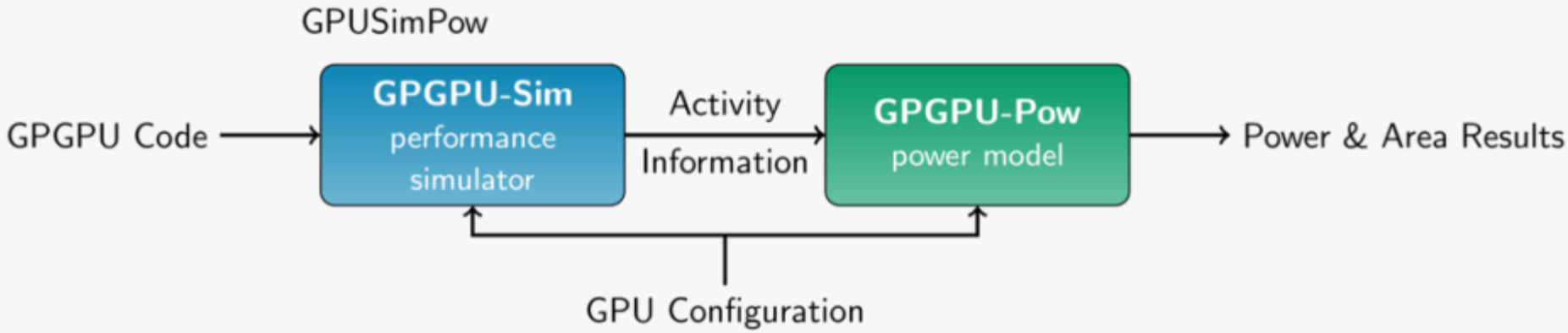
LGPU Power Simulation Framework

Objectives and Achievements

- Objectives:
  - To estimate power of state-of-the-art and emerging games and other applications on various GPU architectures
  - To evaluate architectural and programming optimizations proposed in the LPGPU project
  - Limitations of current approaches
    - Measurement based power models
    - Cannot explore design space
- Achievements:
  - Developed first GPU power simulator
  - Built custom measurement setup to validate simulator results
  - Estimated and measured power of state-of-the-art applications
  - Used for performance and power trade-offs evaluation

LPGPU Power Simulation Framework

High-Level Overview

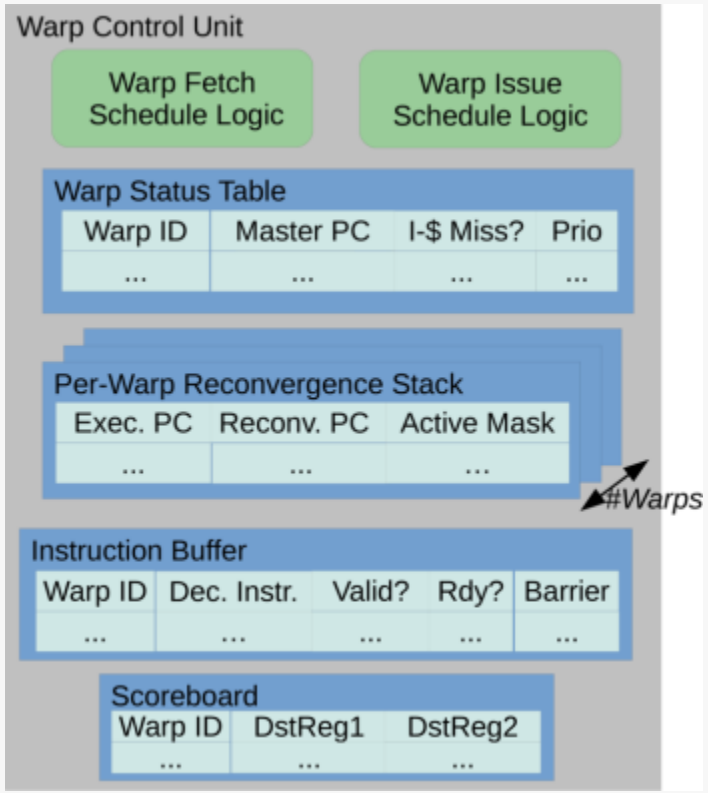


- Key components
  - GPGPU-Sim performance simulator: extracts activity factors
  - GPGPU-Pow power model: substantially modified McPAT with power models of many GPU components
  - Integrated modified GPGPU-Sim and GPGPU-Pow

Power Modeling of GPU Components

- Power models for GPU components added to McPAT:
  - Warp control unit (Warp status table, Instruction buffers, Reconvergence stacks, Scoreboarding logic, Instruction decoder logic, Schedulers)
  - GPU style register file
  - Execution units (INT, FP32, SFU)
  - Load-store unit (Coalescer, Bank conflict checker, AGU array, Per-core constant cache slice, Shared memory, L2 cache)
  - GDDR
- Analytical and measurement-based power modeling of GPU components:
  - CACTI for regular components such as caches
  - Measurement-based models for irregular components such as FUs

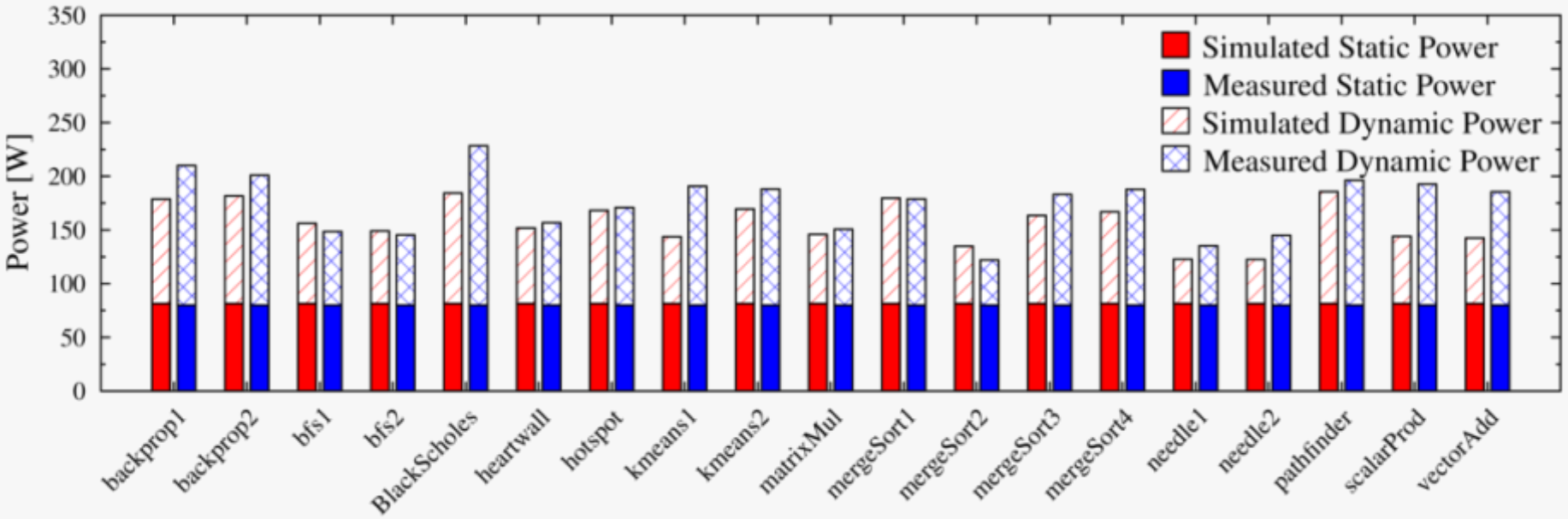
High level overview of warp control unit



GPU Power Consumption Measurements

- Custom GPU measurement testbed
- Used for validation of power simulator and empirical power modeling
- Key features
  - Direct measurement of GPU power consumption
  - Uses special PCIe riser card and PCIe power cables to measure all card power supplies
  - High sampling speed (31.5 KHz) and accuracy (±3%)

Validation: Simulated vs Measured Power for GTX580



- Average relative error is 10.8%

Conclusions

- GPUSimPow first and most accurate power simulation framework for GPUs
  - First presented @ ISPASS (April 2013)
  - GPUWattch presented @ ISCA (June 2013)
    - developed independently
    - similar in spirit
    - own measurement show GPUSimPow more accurate
- Downloadable from www.aes.tu-berlin.de/gpusimpow
- Current and future activities:
  - Measuring activity
  - Continuous improvements to accuracy
    - Power model for MMU
    - More accurate modeling of process scaling
  - Support for 3D stacked DRAM

Motivation

- When LPGPU project started
  - H.264 most recent video codec
  - GPU deployed for several application domains
- Research question: *Can GPUs be deployed efficiently for highly irregular applications such as video codecs*?

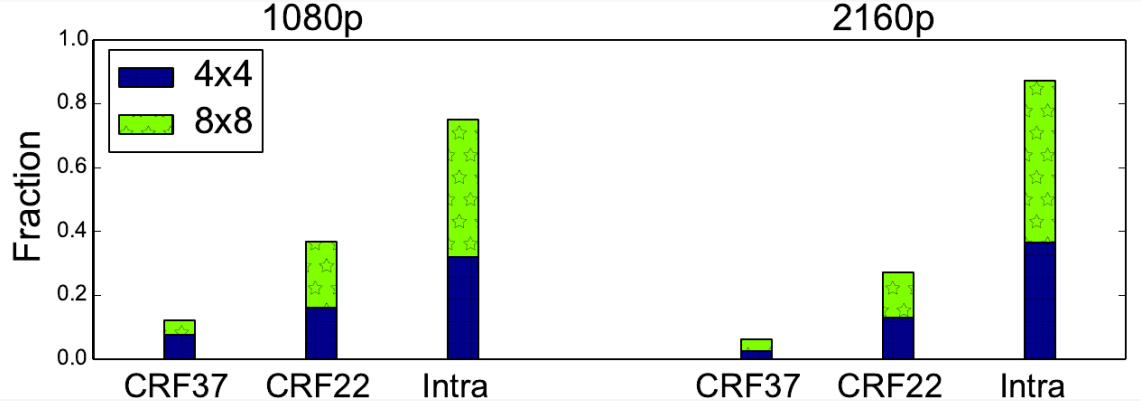
- Which kernels to offload to GPU?

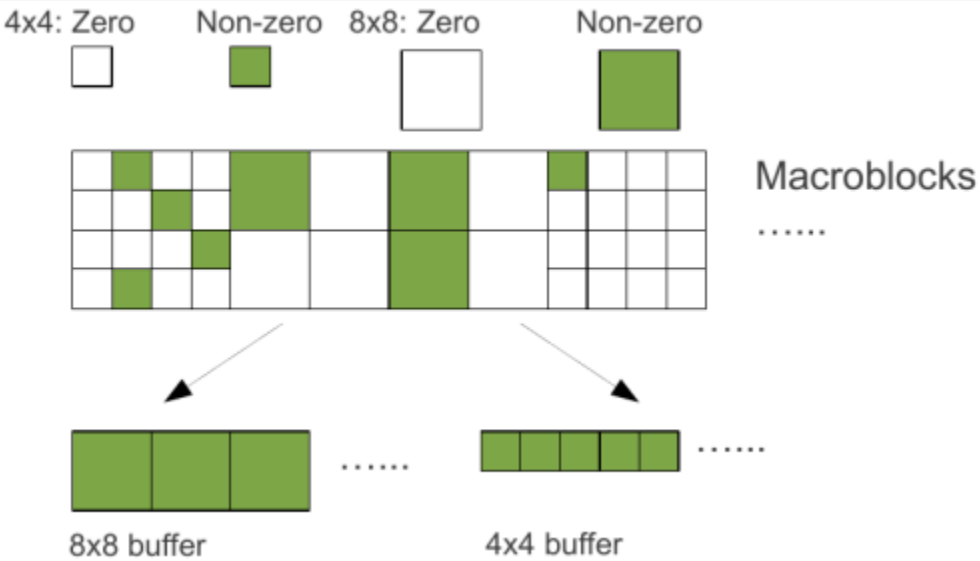
Kernel	entropy decoding	inverse transform	intra-prediction	motion compensation	deblocking filter
Parallelism	low	**high**	low	**high**	medium
Divergence	high	*low*	high	*medium*	high

➢ Offload inverse transform and motion compensation to GPUs

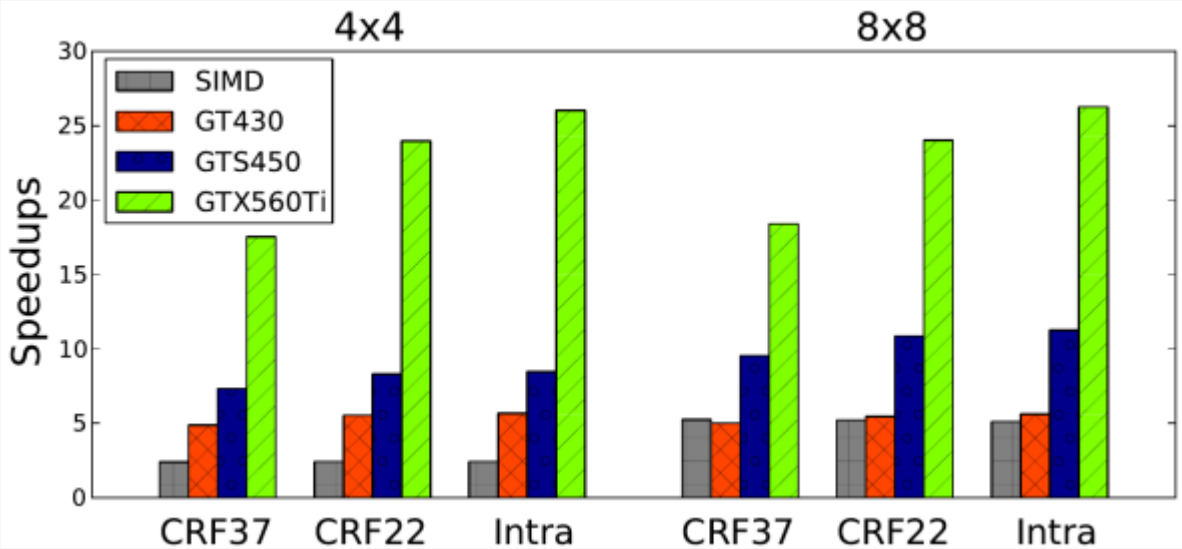
Offloading Inverse Transform

- Opportunity: Video sequences contain many blocks consisting of zero coefficients only
  - Corresponding computations can be skipped
- Challenge: branch divergence due to different block sizes
- Solution: compact and separate





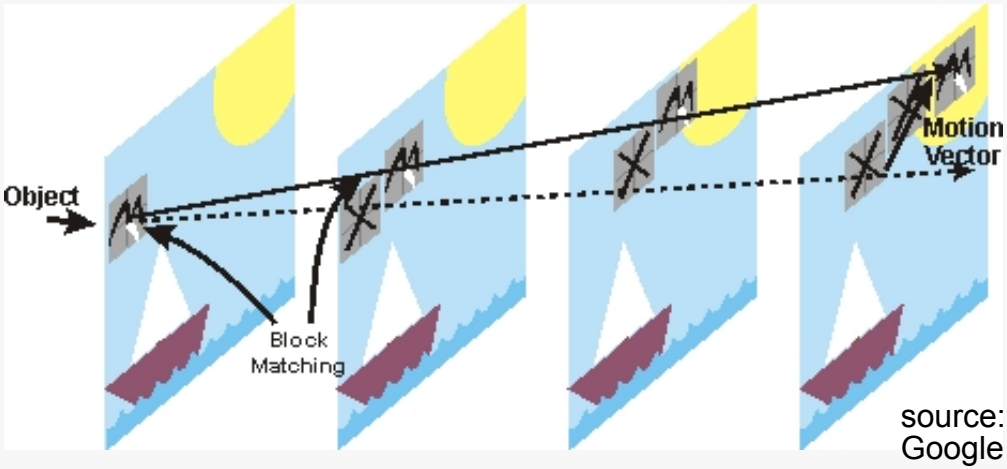
Inverse Transform Kernel-Level Speedup



- Over 25x speedup for both 4x4 and 8x8 inverse transform
- Performance scales well on NVIDIA Fermi GPUs:
  - GT430 → GTS450 → GTX560Ti: 2 → 4 → 8 SMs
- Details in paper:
  - B. Wang, M. Alvarez-Mesa, C. Chi, and B. Juurlink, "An Optimized Parallel IDCT on Graphics Processing Units", Proc. Euro-Par Workshops 2012.

Offloading Motion Compensation (MC)

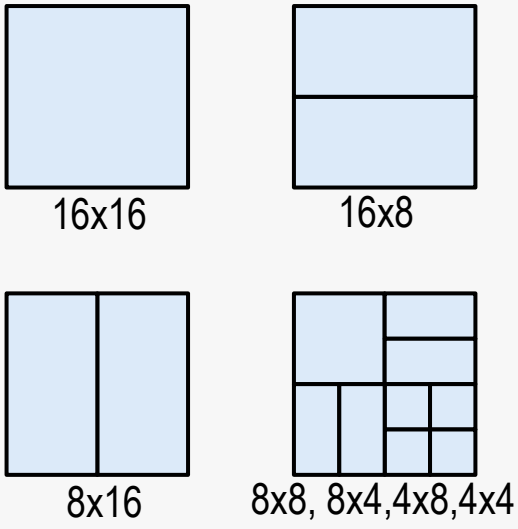
- Motion compensation main kernel in block-based video codecs



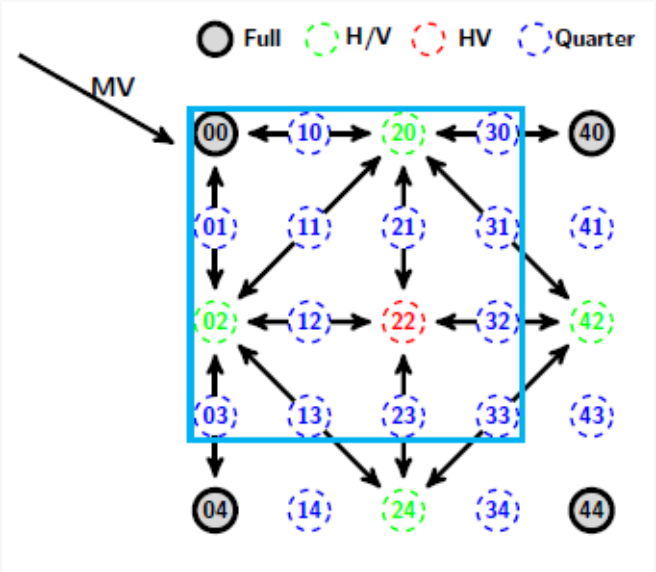
source: Google Image

- Implementation challenges:

many possible partition schemes:



many interpolation modes:

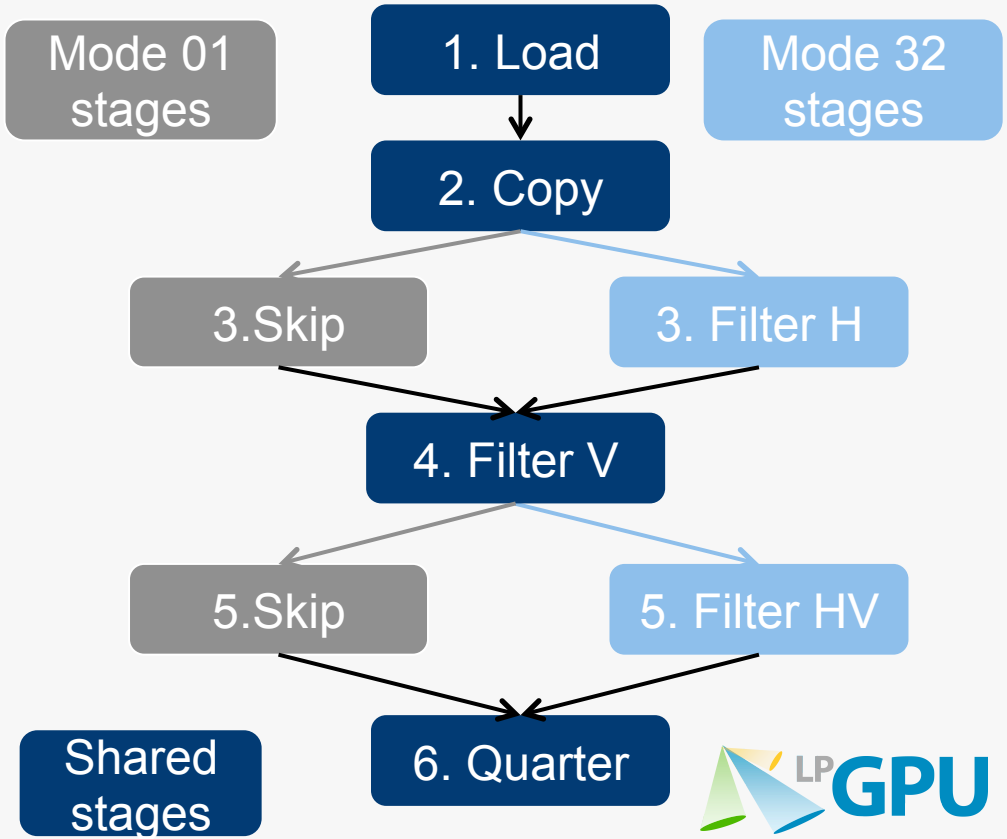


Multistage implementation: Reduced Divergence

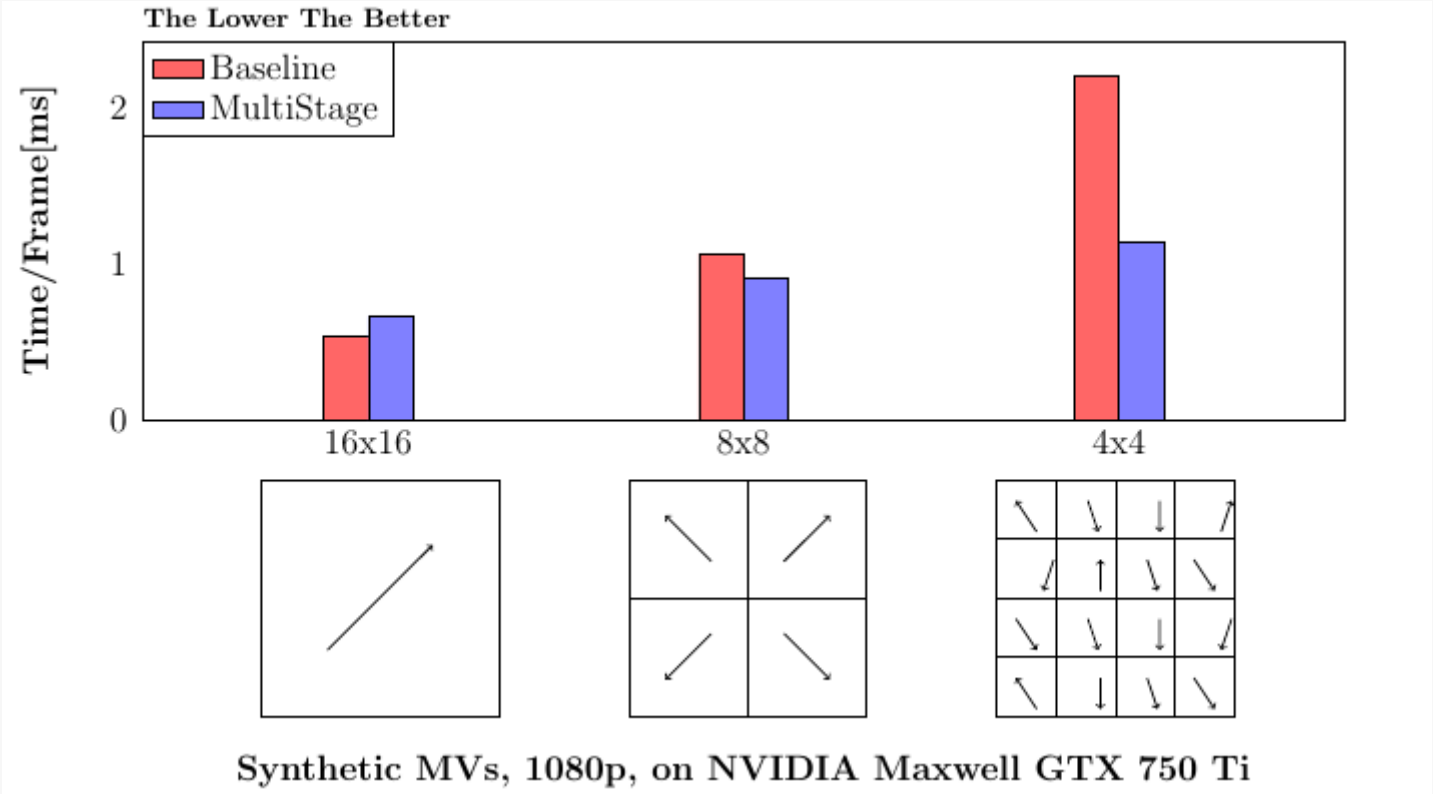
- Baseline:
  - 16 computation modes
  - Per mode implementation
  - Result in divergence!
  - Pseudo code:

```
switch(mode){
      case mode00:
            do_MC_mode00;
            break;
      case mode01:
            do_MC_mode01;
            break;
      .
      .
      case mode33:
            do_MC_mode33;
            break;
  }
```

- Multistage implementation:
  - Each mode consists of max. 6 stages
  - Some stages can be shared between modes
  - Divergence reduced



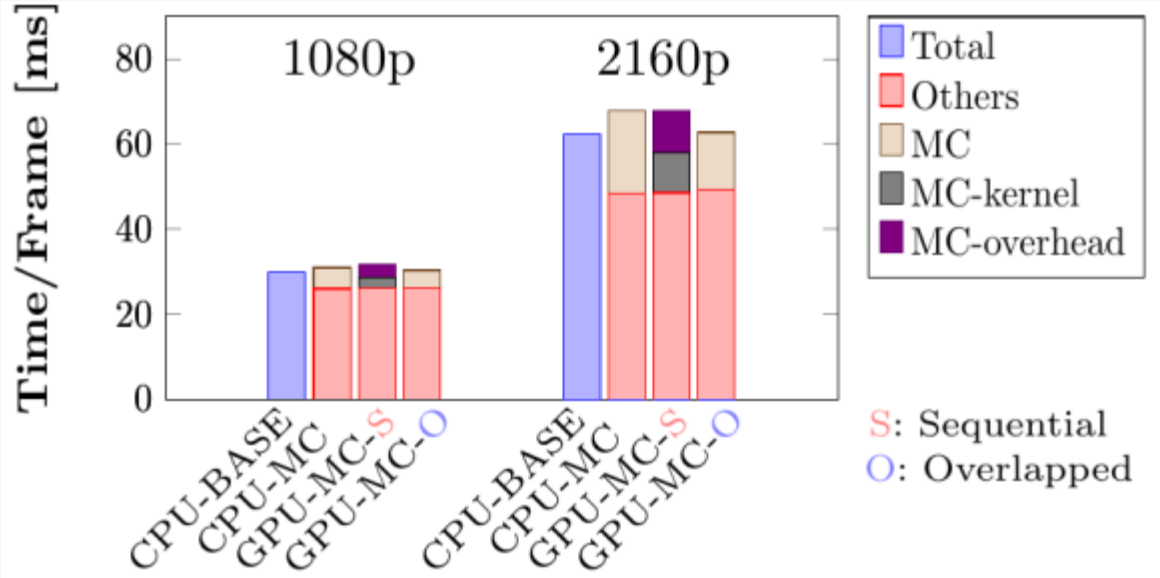
Motion Compensation Results



Synthetic MVs, 1080p, on NVIDIA Maxwell GTX 750 Ti

LPGPU

Application-Level Performance



Platform:

- Intel Sandybridge CPU
- NVIDIA Fermi GPU

- Speedup at kernel level (2x), but not at application-level
- Causes:
  - Motion compensation not only kernel
  - Overhead:
    - Memory copy
    - OpenCL runtime

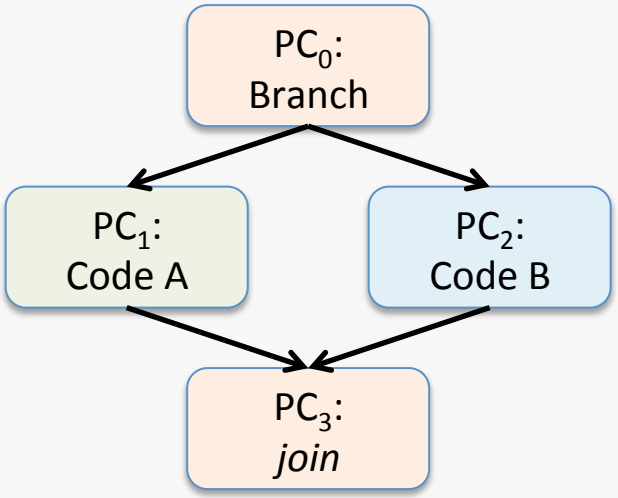
LP GPU

Conclusions

- Video Decoding on GPUs was and remains a challenge
  - Different architecture that keeps the benefits of GPUs while reducing their disadvantages better solution
- "*Parallel H.264/AVC Motion Compensation for GPUs using OpenCL*", accepted by IEEE TCSVT
- H.264 OpenCL decoder downloadable from:
  - http://www.aes.tu-berlin.de/menue/research/projects/high_performance_video_coding
- Current and future work:
  - Parallelizing HEVC in-loop filter using OpenCL
  - Offload HEVC motion compensation onto GPUs

Motivation

- As shown, due to branch divergence video codecs difficult to implement efficiently on SIMD-GPUs



Space

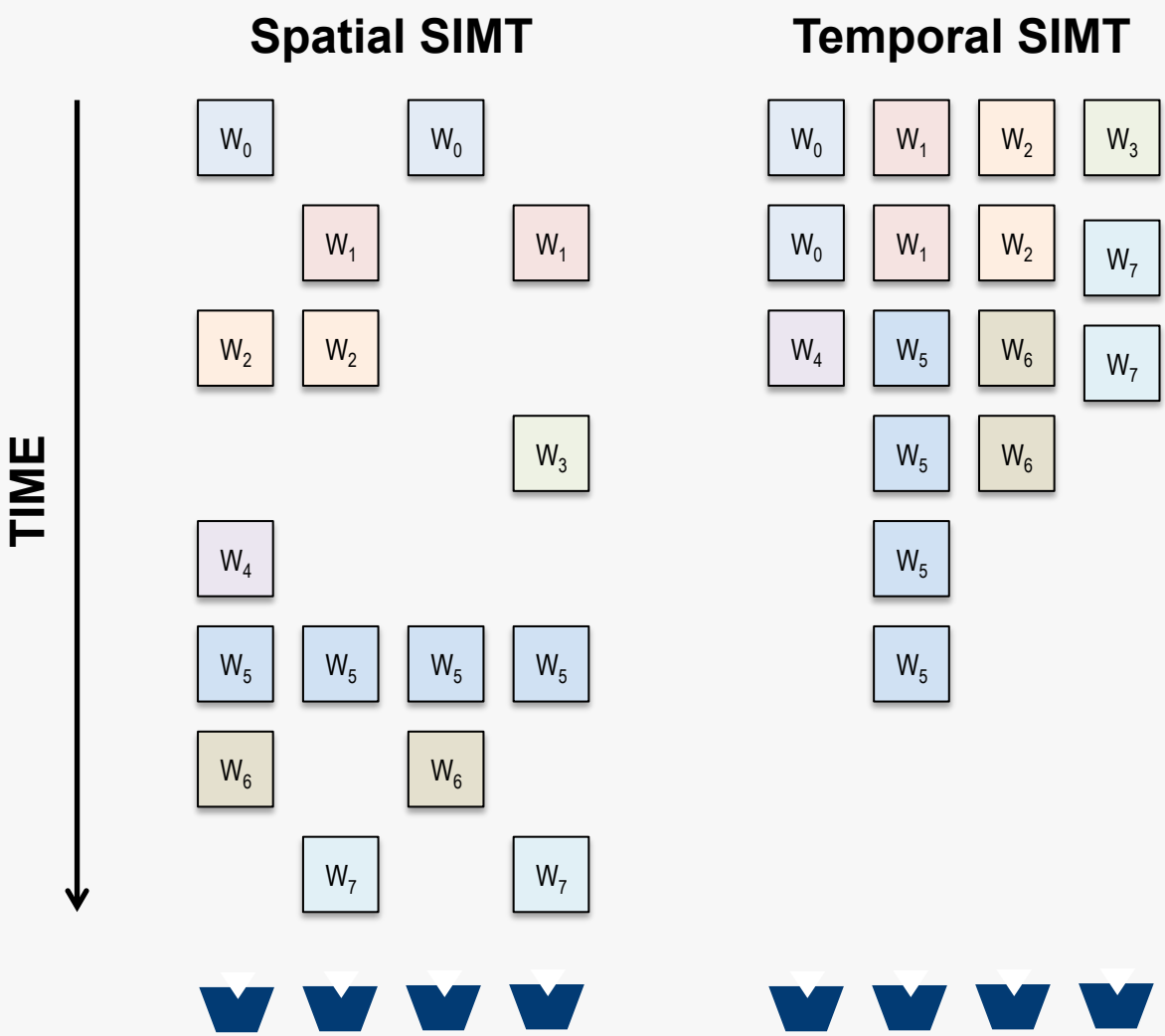
Time

T_0 PC_0	T_1 PC_0	T_2 PC_0	T_3 PC_0
T_0 PC_1	*waste*	T_2 PC_1	*waste*
waste	T_1 PC_2	*waste*	T_3 PC_2
T_0 PC_3	T_1 PC_3	T_2 PC_3	T_3 PC_3

- SIMD-GPUs power efficient because instruction frontend (fetching, decoding, …) shared between several warps
- Motivates different approach called Temporal SIMT
  - Our instantiation called Decoupled ARchitecture using Temporal simt (DART)

LPGPU

Temporal SIMT (DART)
Basic Idea

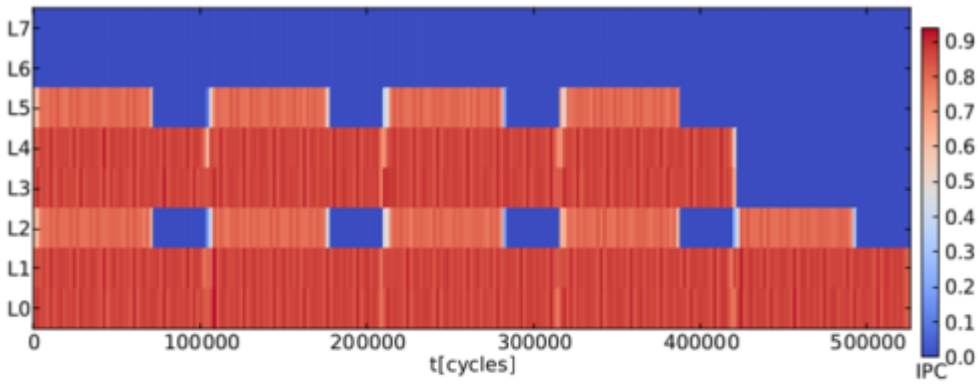


- Requires way to skip operations
- Next no-NOP in warp identified in instruction stream
- Fig illustrates filled pipeline
- S-SIMT and T-SIMT require same amount of resources

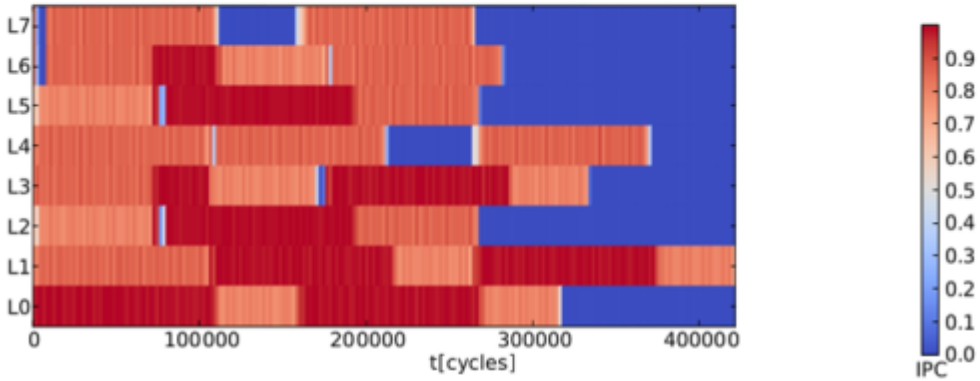
LPGPU

There's More to It Than Meets the Eye

- Load balancing issues in some benchmarks.
- To solve, needed to improve
  - register allocation
  - resource management
  - instruction scheduling
- Included and improved scalarization
  - Much easier in T-SIMT than in conventional S-SIMT
- For some benchmarks T-SIMT better; for others S-SIMT
  - Developed Spatial-Temporal (ST-)DART



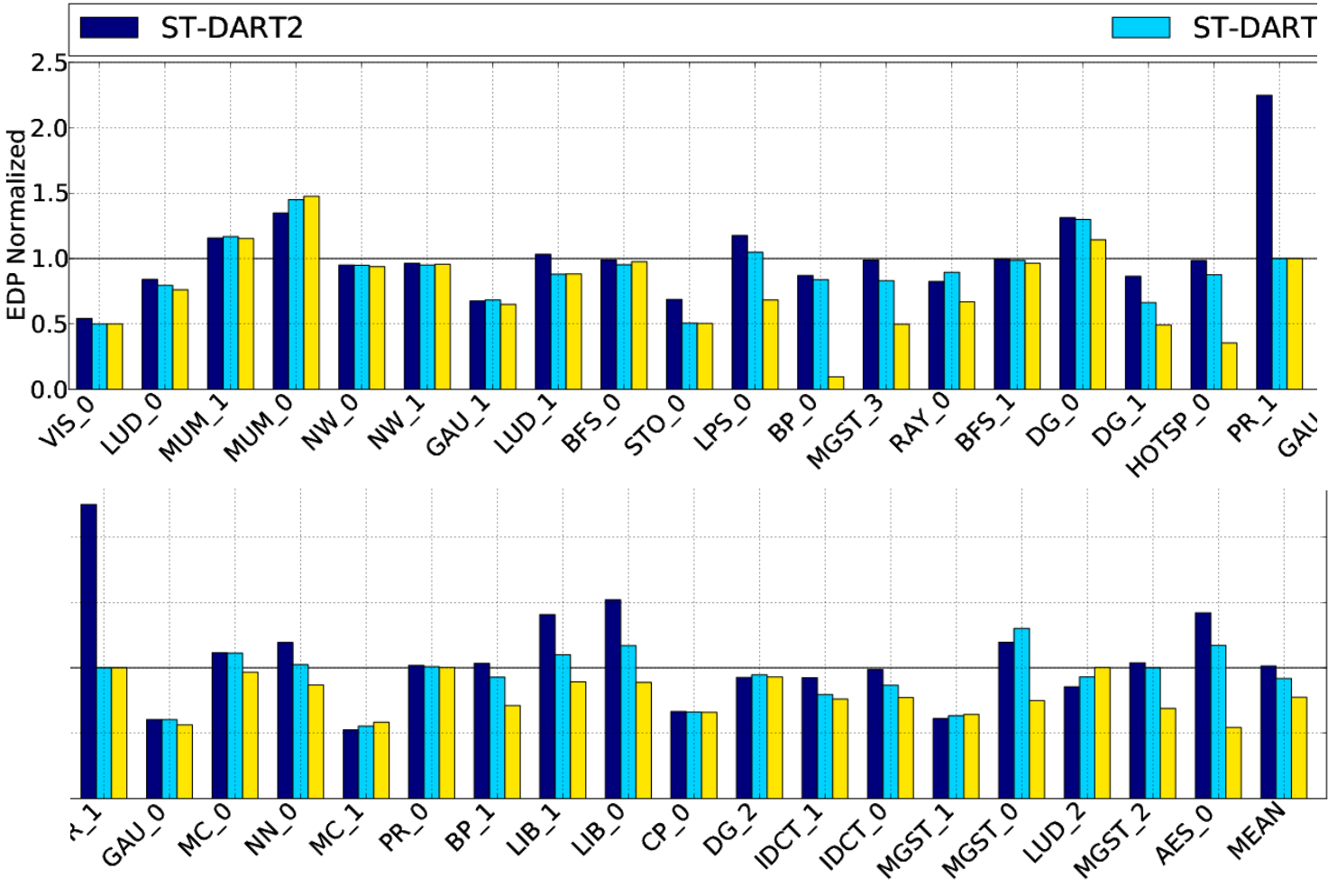
(a) DG_0 benchmark on unoptimized DART



(b) DG_0 benchmark on optimized DART

Temporal SIMT (DART)

Experimental Results



- LPGPU MC_1 and VIS_0 Kernels show EDP improvement of ~50%
- Average EDP Improvement close to 25%

Conclusions

- NVidia patent only provides rough description of Temporal SIMT
- We are first to provide actual implementation
- Discovery: there's much more to it than meets the eye
- Were able to obtain improvements compared to conventional SIMT but required substantial R&D

LPGPU Conclusions

- Many more achievements in LPGPU than can be presented in 20-25 minutes
- This talk focused on achievements led by TU Berlin
- For other achievements, come see our poster in the European projects poster session
- Current and future work
  - Power simulator
    - measure activity more accurately
  - Video decoding on GPUs
    - H.265 / HEVC / MPEG-H
  - DART architecture
    - finish huge design space exploration

Src: blog.atrinternational.com

Thank You!

- The audience for listening
- The European Union for buying lunch
- My girlfriend for her patience
- My team
- And all others that contributed some{time, where, how}

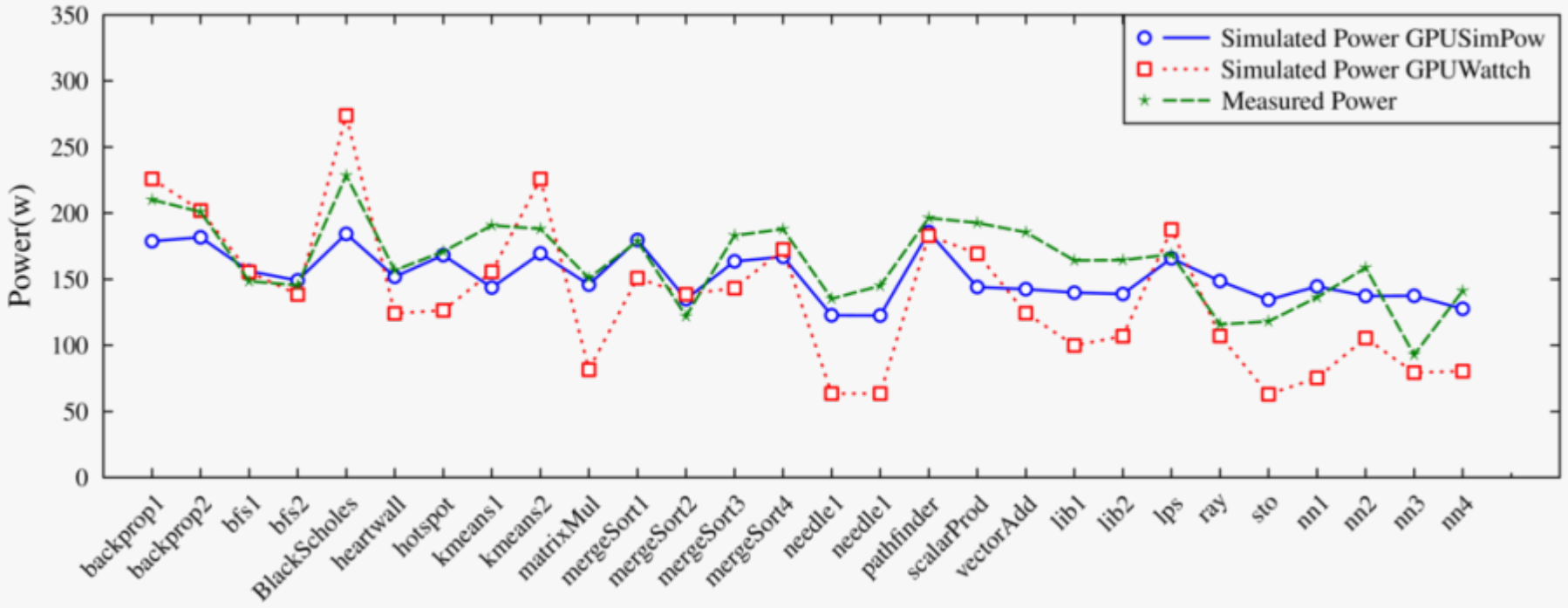
- Questions?

Src: dogonews.com, “Stump Your Parents With These (Easy) Science Questions”

LP GPU

Backup slides

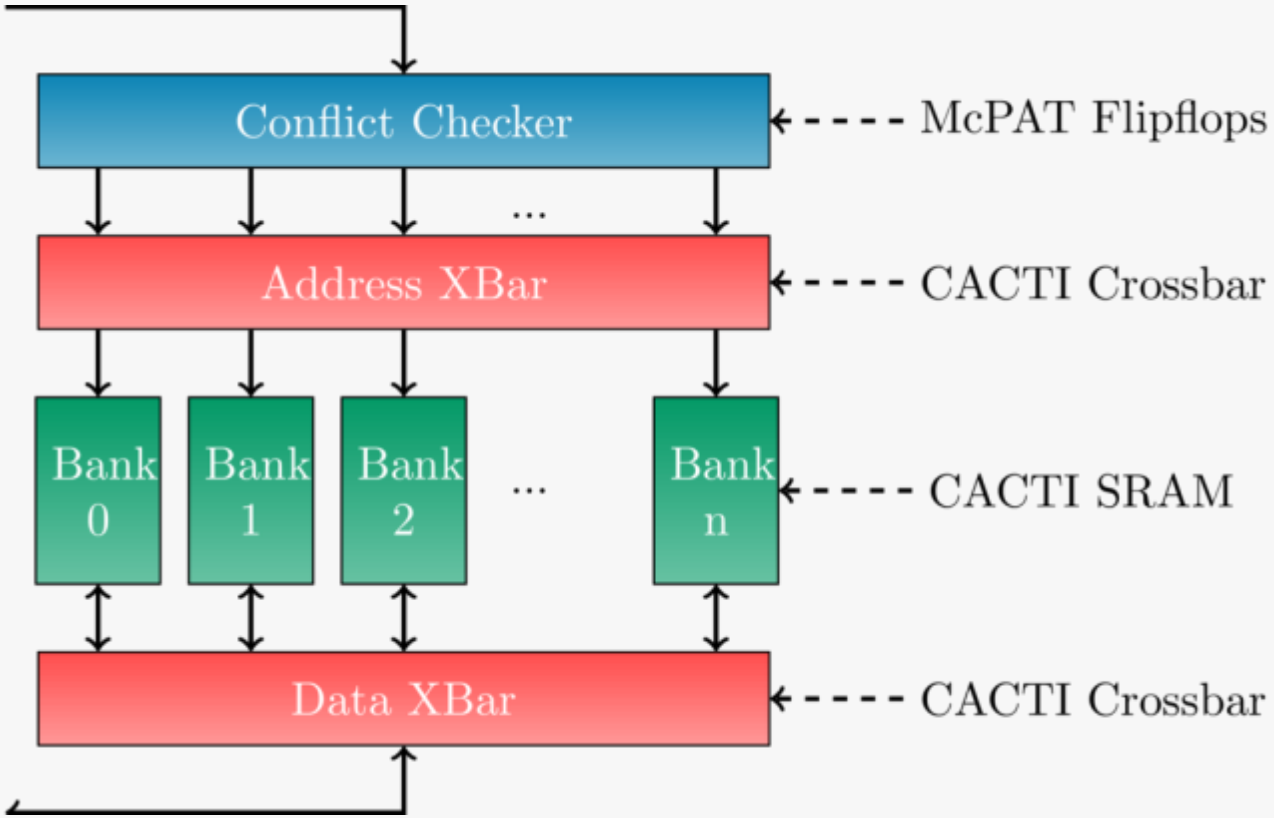
GPUSimPow-GPUWattch Comparison



- GPGPU-Sim average relative error 10.8%
- GPUWattch average relative error 20.5%
- GPUSimPow follows measured power more closely

GPU Components Power Modeling Approach

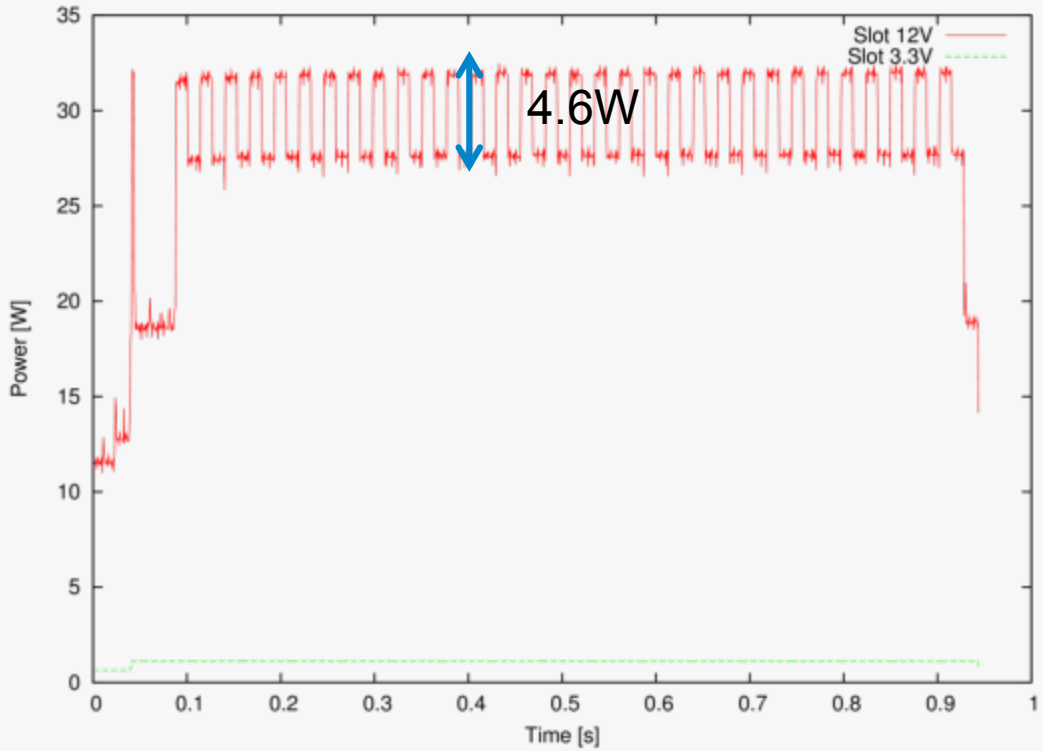
- Mixture of analytical and measurement based models
  - CACTI for regular components like caches



Example of shared memory power model

GPU Components Power Modeling Approach

- Measurement based models for irregular components like FU
  - Micro-benchmarks to stress the component
  - Measure power consumed



Example of FU power modeling

- 12 SMs @1.34GHz consume 4.6W on GT240
- Energy per operation is 37.9 pJ