CS4803DGC Design Game Consoles

Spring 2009

Prof. Hyesoon Kim

Georgia Tech College of Computing

Workload Characterizations

- Benchmarking is critical to make a design decision and measuring performance
  - Performance evaluations:
    - Design decisions
      - Earlier time : analytical based evaluations
      - From 90's: heavy rely on simulations.
    - Processor evaluations
  - Workload characterizations: better understand the workloads

Measuring Performance

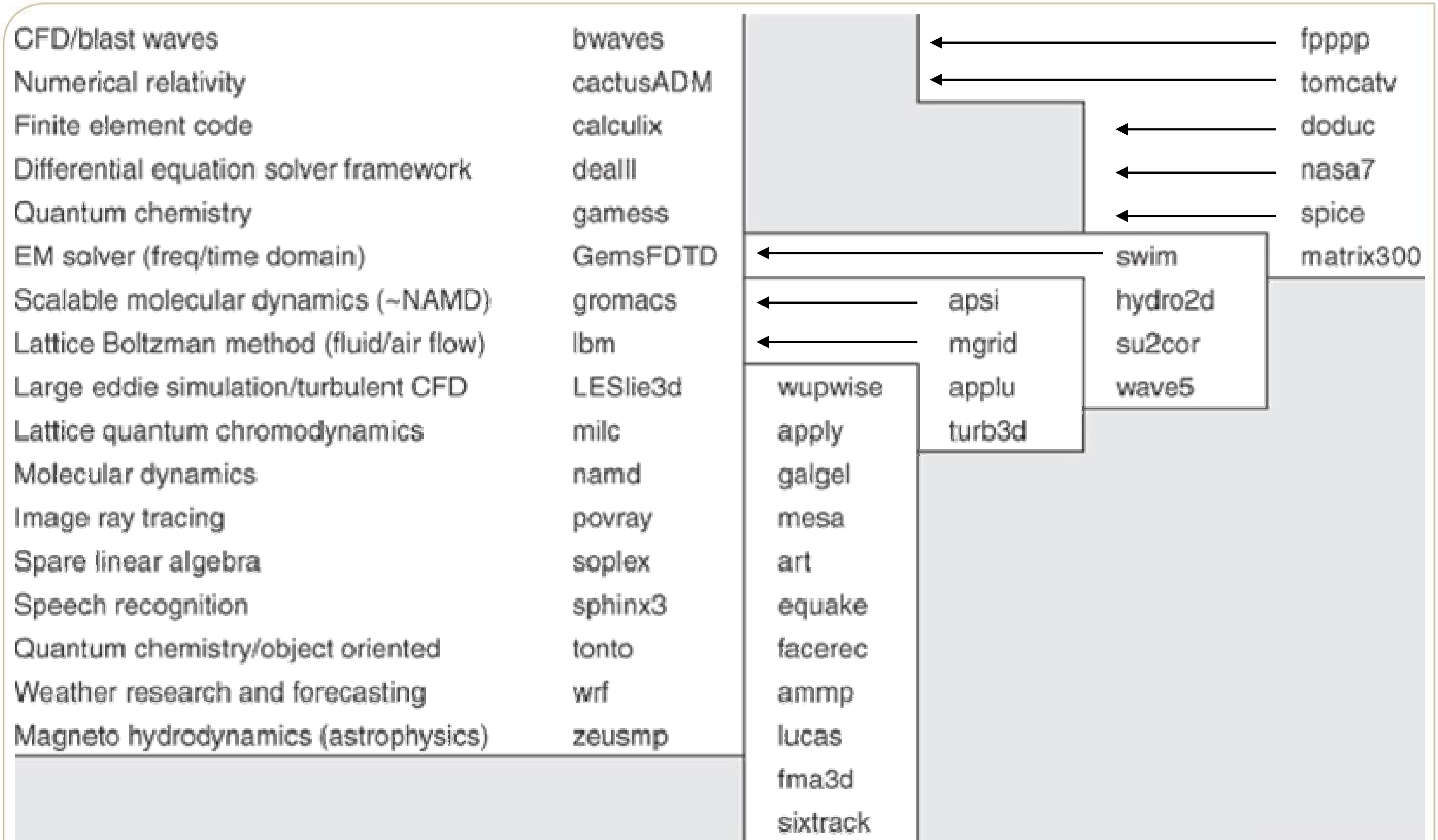
- Benchmarks
  - Real applications and application suites
    - E.g., SPEC CPU2000, SPEC2006, TPC-C, TPC-H, EEMBC, MediaBench, PARSEC, SYSmark
  - Kernels
    - "Representative" parts of real applications
    - Easier and quicker to set up and run
    - Often not really representative of the entire app
  - Toy programs, synthetic benchmarks, etc.
    - Not very useful for reporting
    - Sometimes used to test/stress specific functions/features

SPEC CPU (integer)

	Benchmark name by SPEC generation				
SPEC2006 benchmark description	SPEC2006	SPEC2000	SPEC95	SPEC92	SPEC89
GNU C compiler					gcc
Interpreted string processing			perl		espresso
Combinatorial optimization		mcf			li
Block-sorting compression		bzip2		compress	eqntott
Go game (AI)	go	vortex	go	sc	
Video compression	h264avc	gzip	ijpeg		
Games/path finding	astar	eon	m88ksim		
Search gene sequence	hmmer	twolf			
Quantum computer simulation	libquantum	vortex			
Discrete event simulation library	omnetpp	vpr			
Chess game (AI)	sjeng	crafty			
XML parsing	xalancbmk	parser			

“Representative” applications keeps growing with time!

SPEC CPU (floating point)



Spec Input Sets

- Test, train and ref
- Test: simple checkup
- Train: profile input, feedback compilation
- Ref: real measurement. Design to run long enough to use for real system
  - -> Simulation?
- Reduced input set
- Statistical simulation
- Sampling

TPC Benchmarks

- Measure transaction-processing throughput
- Benchmarks for different scenarios
  - TPC-C: warehouses and sales transactions
  - TPC-H: ad-hoc decision support
  - TPC-W: web-based business transactions
- Difficult to set up and run on a simulator
  - Requires full OS support, a working DBMS
  - Long simulations to get stable results

Multiprocessor's benchmarks

- SPLASH: Scientific computing kernels
  - Who used parallel computers?
- PARSEC: More desktop oriented benchmarks
- NPB: NASA parallel computing benchmarks
- Not many

Performance Metrics

- GFLOPS, TFLOPS
- MIPS (Million instructions per second)

Normalizing & the Geometric Mean

- Speedup of arithmeitc means != arithmetic mean of speedup

- Use geometric mean: $\sqrt[n]{\prod_{i=1}^{n} \text{Normalized execution time on } i}$

- Neat property of the geometric mean: *Consistent whatever the reference machine*
- **Do not use the arithmetic mean for normalized execution times**

CPI/IPC

- Often when making comparisons in comp-arch studies:
  - Program (or set of) is the same for two CPUs
  - The clock speed is the same for two CPUs

- So we can just directly compare CPI's and often we use IPC's

Average CPI vs. "Average" IPC

- Average CPI = $(CPI_1 + CPI_2 + \ldots + CPI_n)/n$

- A.M. of IPC = $(IPC_1 + IPC_2 + \ldots + IPC_n)/n$

  Not Equal to A.M. of CPI!!!

- Must use *Harmonic Mean* to remain $\propto$ to runtime

Harmonic Mean

- $\text{H.M.}(x_1, x_2, x_3, \ldots, x_n) =$

$$\frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \ldots + \frac{1}{x_n}}$$

- What in the world is this?
  - Average of inverse relationships

A.M.(CPI) vs. H.M.(IPC)

- "Average" IPC = $\dfrac{1}{\text{A.M.(CPI)}}$

$$= \frac{1}{\frac{CPI_1}{n} + \frac{CPI_2}{n} + \frac{CPI_3}{n} + \ldots + \frac{CPI_n}{n}}$$

$$= \frac{n}{CPI_1 + CPI_2 + CPI_3 + \ldots + CPI_n}$$

$$= \frac{n}{\frac{1}{IPC_1} + \frac{1}{IPC_2} + \frac{1}{IPC_3} + \ldots + \frac{1}{IPC_n}} = \text{H.M.(IPC)}$$

GPU Benchmarks

- Stanford graphics benchmarks
  - Simple graphics workload. Academic
- Mostly game applications
  - 3DMark:
  - http://www.futuremark.com/benchmarks/3dmarkvantage
  - Tom's hardware

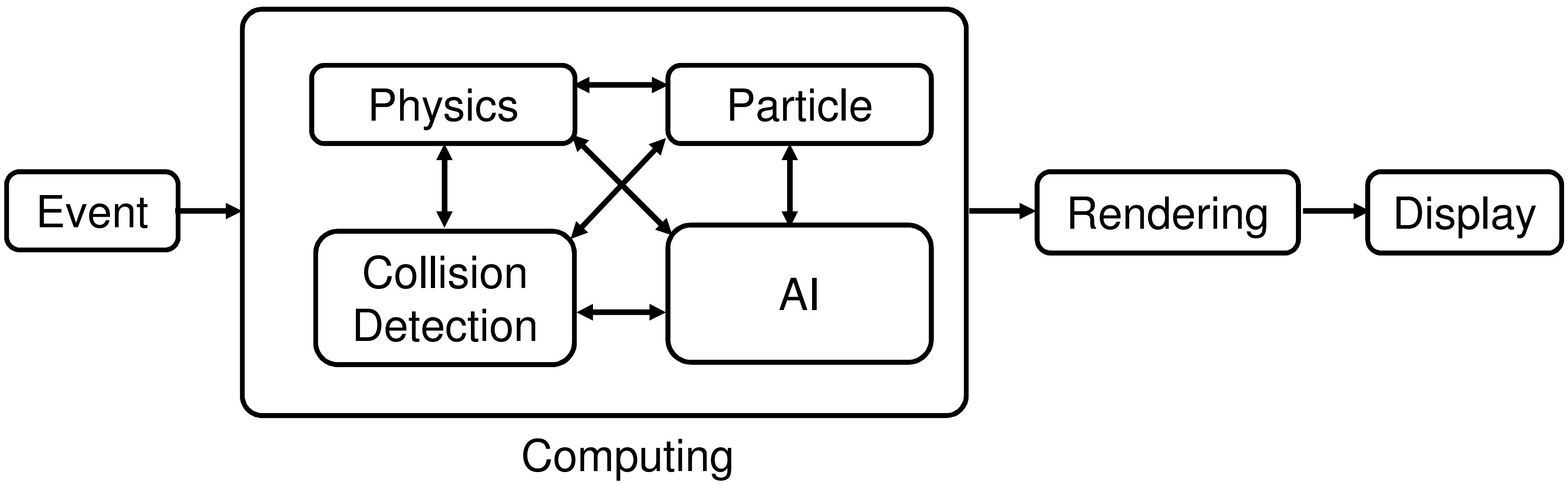
Game Workload Charcterizations

- Still graphics is the major performance bottlenecks
- Previous research: emphasis on graphics

Game workloads

- Several genres of video games
  - First Person Shooter
    - Fast-paced, graphically enhanced
    - Focus of this presentation
  - Role-Playing Games
    - Lower graphics and slower play
  - Board Games
    - Just plain boring

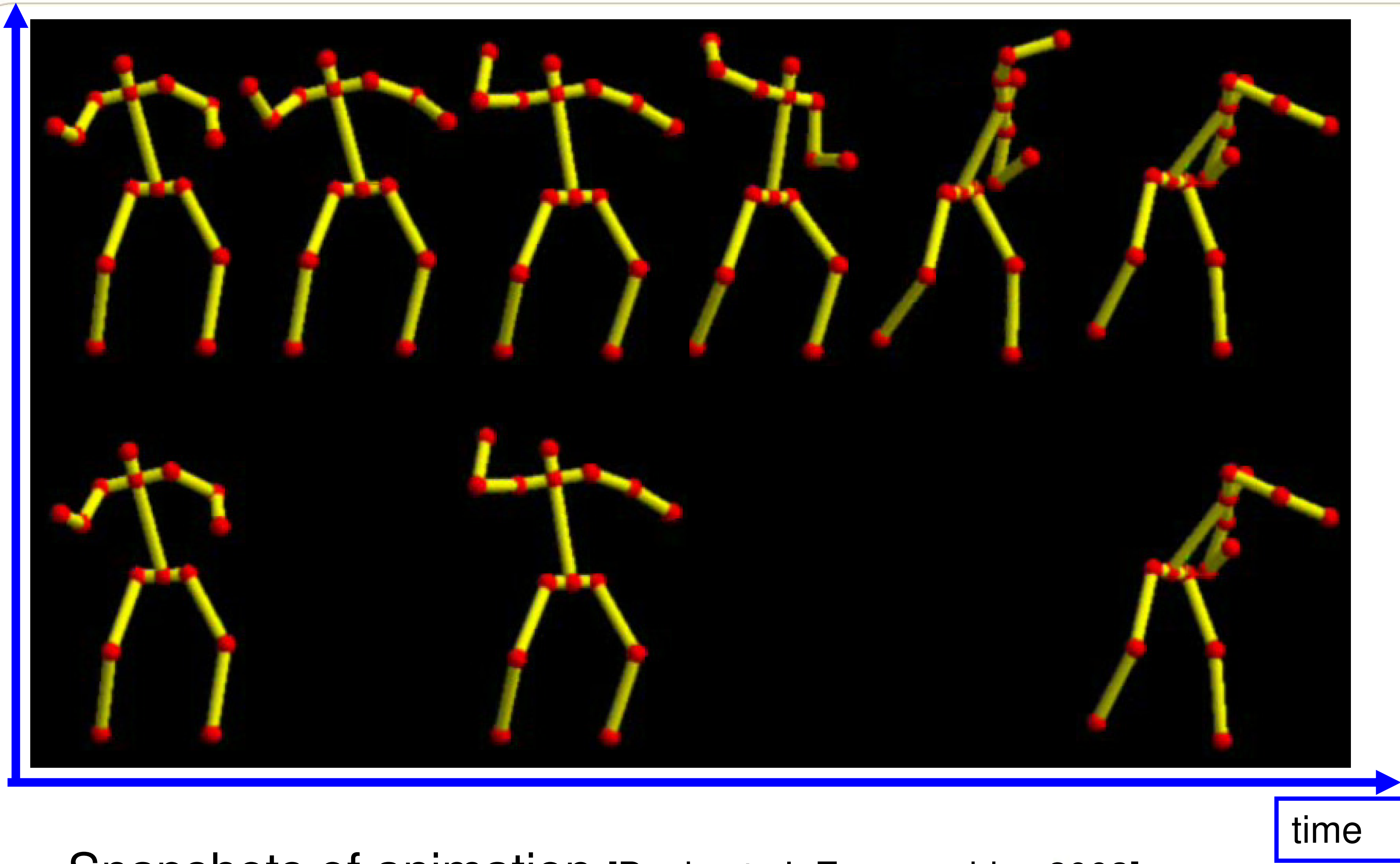
Overview of Game Engine



Frame Rates

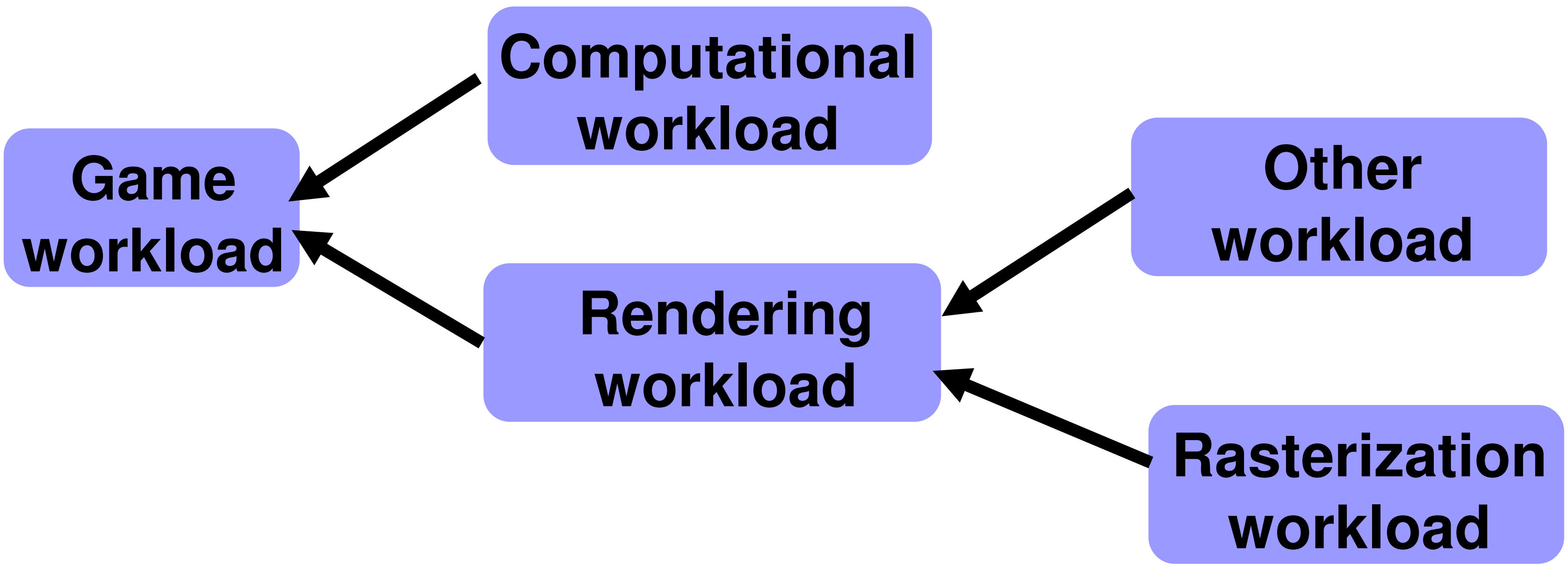
- Current game design principles:
  - higher frame rates imply the better game quality
- Recent study on frame rates [Claypool et al. MMCN 2006]
  - very high frame rates are not necessary, very low frame rates impact the game quality severely

A First Cut: Reduce Frame Rates



Snapshots of animation [Davis et al. Eurographics 2003]

Game workloads



Game workload characterization

- Case study
  - Workload characterization of 3D games, Roca, et al. IISWC 2006 [WOR]
  - Use ATTILA

TABLE III

AVERAGE INDICES PER BATCH AND FRAME AND TOTAL BW

Game/Timedemo	avg. indexes per batch	avg. indexes per frame	bytes per index	BW @100fps
UT2004/Primeval	1110	249285	2	50 MB/s
Doom3/trdemo1	275	196416	4	79 MB/s
Doom3/trdemo2	304	136548	4	55 MB/s
Quake4/demo4	405	172330	4	69 MB/s
Quake4/guru5	166	135051	4	54 MB/s
Riddick/MainFrame	356	214965	2	43 MB/s
Riddick/PrisonArea	658	239425	2	48 MB/s
FEAR/built-in demo	641	331374	2	66 MB/s
FEAR/interval2	1085	307202	2	61 MB/s
Half Life 2 LC/built-in	736	328919	2	66 MB/s
Oblivion/Anvil Castle	998	711196	2	142 MB/s
Splinter Cell 3/first level	308	177300	2	35 MB/s

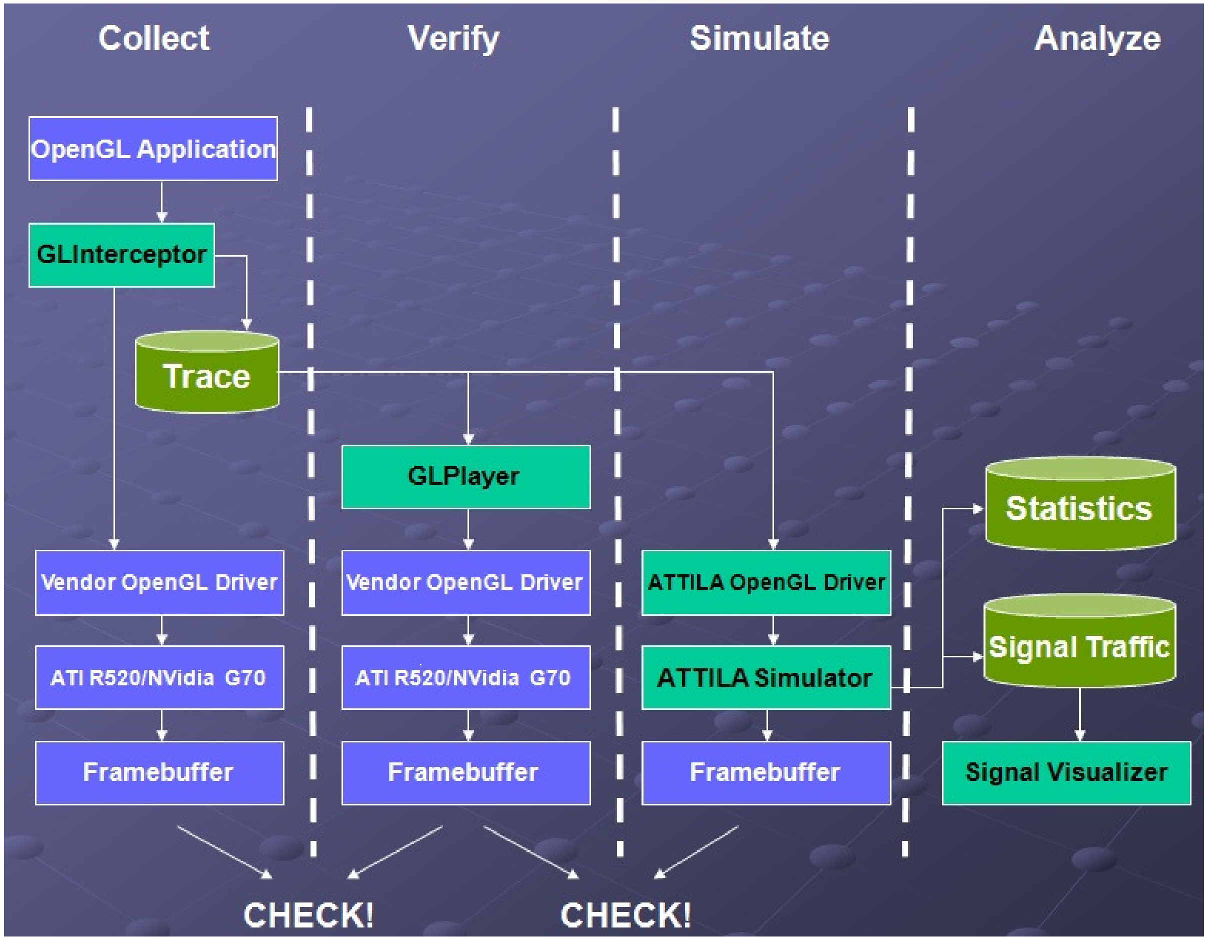
Characterization Items

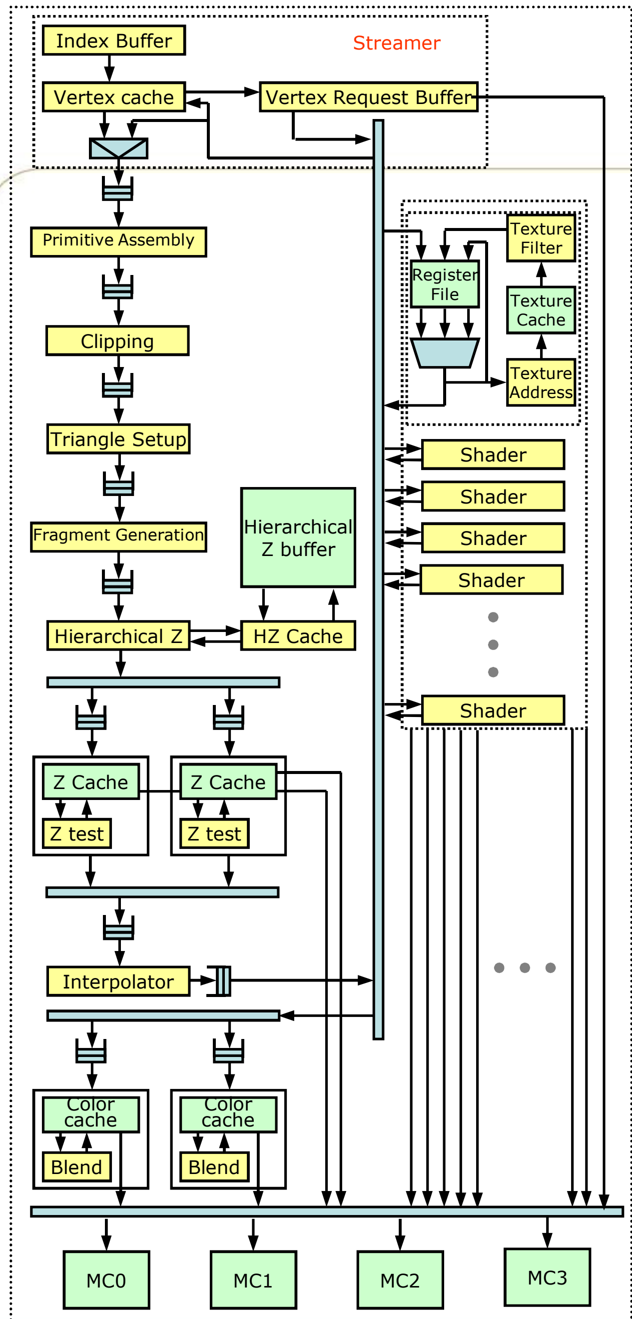
- Average primitives per frame
- Average vertex shader instructions
- Vertex cache hit ratio
- System bus bandwidths
- Percentage of clipped, culled, and traversed triangles
- Average trianglesizes

ATTILA

- GPU execution driven simulator
- https://attilaac.upc.edu/wiki/index.php/Architecture
- Can simulate OpenGL at this moments

Attila Frame





- Attila architecture

Unit	Size	Element width
Streamer	48	16x4x32 bits
Primitive Assembly	8	3x16x4x32 bits
Clipping	4	3x4x32 bits
Triangle Setup	12	3x4x32 bits
Fragment Generation	16	3x4x32 bits
Hierarchical Z	64	(2x16+4x32)x4 bits
Z Tests	64	(2x16+4x32)x4 bits
Interpolator	---	---
Color Write	64	(2x16+4x32)x4 bits
Unified Shader (vertex)	12+4	16x4x32 bits
Unified Shader (fragment)	240+16	10x4x32 bits

Table 2. Queue sizes and number of threads in the ATTILA reference architecture

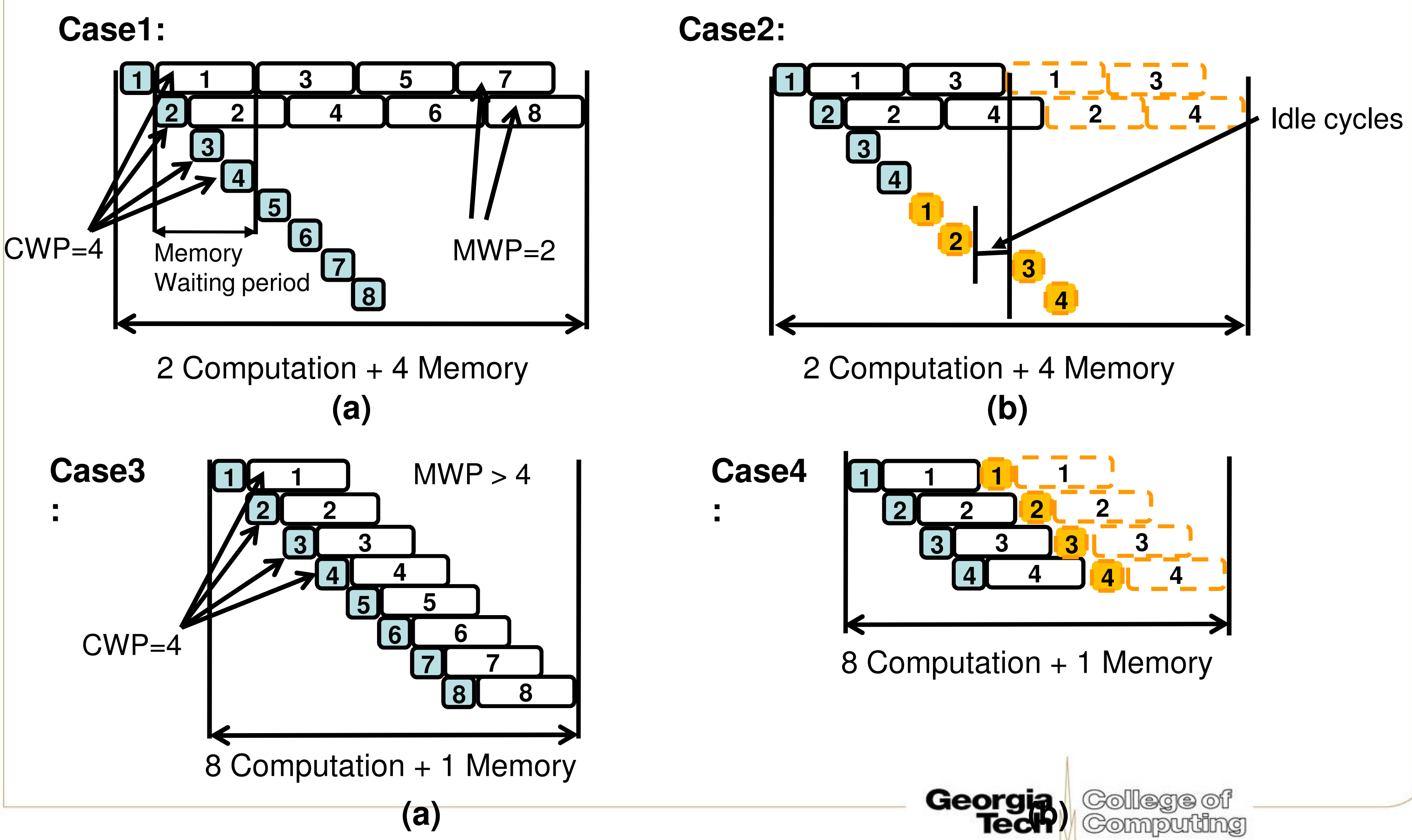
Simulation

- Execution driven:
  - Correctness, long development time,
  - Execute binary
- Trace driven
  - Easy to develop
  - Simulation time could be shorten
  - Large trace file size

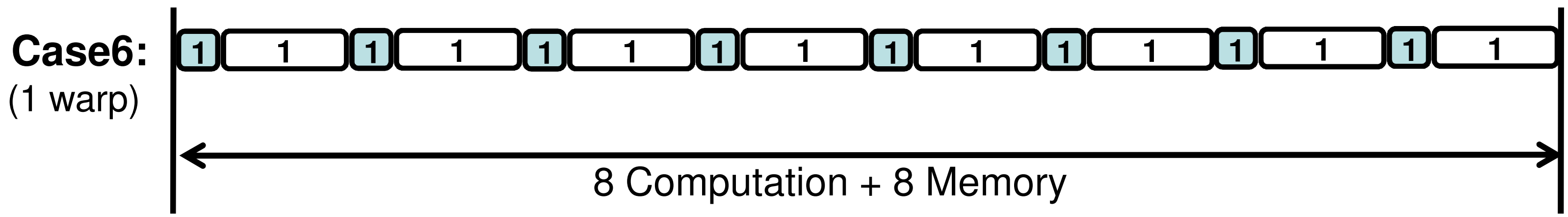
Analytical Model

- No simulation is required
- To provide insights
- Statistical Methods
- CPU
  - First-order
- GPU
  - Warp level parallelism

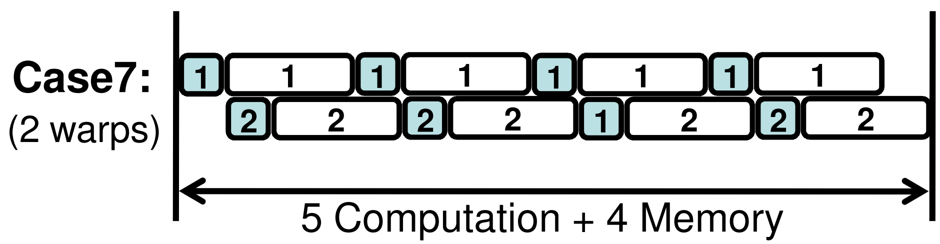
GPU Analytical Model



GPU Analytical Model



(a)



(b)

CPU workload characterizations

- Hardware performance counters
  - Built in counters (instruction count, cache misses, branch mispredicitons)
- Profiler
- Architecture simulator
- Characterized items
  - Cache miss, branch mispredicition, row-buffer hit ratio

Final Design Review

- Top design
  - (instruction, data flow from memory to CPU and GPU), Data/control signals
- CPU design
  - Pipeline stages, **SMT support**, Fetch address calculation, branch misprediction, cache miss handling path
  - Memory address calculation stage, vector processing units
  - At least 5 MUXes, register, ALU, latches,
  - Memory system: Load/store buffers, queues
- GPU Design
  - Show at least 10 ALUs

Additional Features

- One of the following items
  - Detailed CPU pipeline design (more muxes and more adders)
  - Detailed survey (more information from other sources)
  - Detailed GPU pipeline design (more muxes and more adders)
  - Detailed memory system (more queues)
  - Detailed memory controller

FAQ

- I/O → just a box
- Cache just one box or (tag + data)
- Report: explanations are required.
- ECC: just a box
- Design review: 30 min
  - Feedback for final report