UC RIVERSIDE

Increasing Energy Efficiency of GPUs Through Hardware Resource Partitioning and Masking

MacREU 2021
Mika Shanela Carodan
Supervised by Professor Daniel Wong
Systems Optimization and Computer Architecture Lab (SoCal)

Research Initiatives

Systems Optimization and Computer Architecture Lab

Lab
Explore viable solutions for sharing resources in parallel computing systems

Project
Optimize performance efficiency and power consumption of parallel programs as we scale GPU hardware resources

Future
Promoting a runtime model that advances sustainability in GPU architecture

What is a GPU?

Graphics Processing Unit (GPU)

What is a GPU?

Graphics Processing Unit (GPU)

- **High computation efficiency**

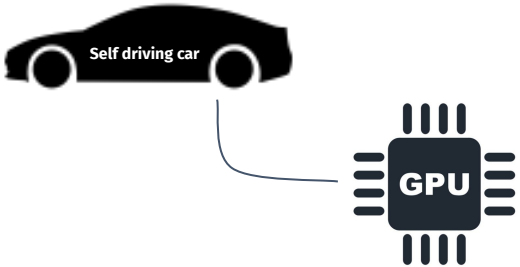
What is a GPU?

Graphics Processing Unit (GPU)

- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**

What is a GPU?

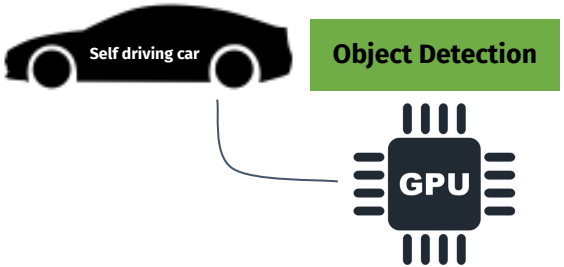
Graphics Processing Unit (GPU)



- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**
  - **Ex: Automated Cars are accelerated by automotive-grade GPUs**

What is a GPU?

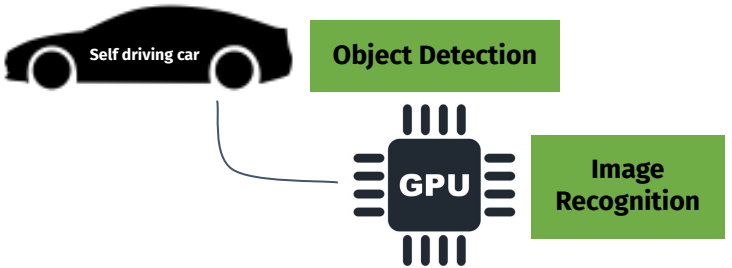
Graphics Processing Unit (GPU)



- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**
  - **Ex: Automated Cars are accelerated by automotive-grade GPUs**

What is a GPU?

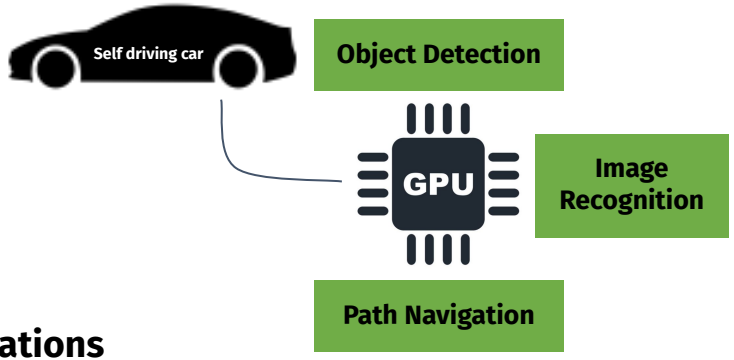
Graphics Processing Unit (GPU)



- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**
  - **Ex: Automated Cars are accelerated by automotive-grade GPUs**

What is a GPU?

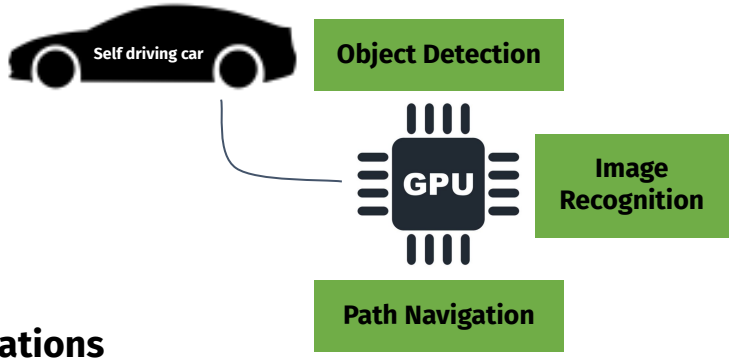
Graphics Processing Unit (GPU)



- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**
  - **Ex: Automated Cars are accelerated by automotive-grade GPUs**

What is a GPU?

Graphics Processing Unit (GPU)

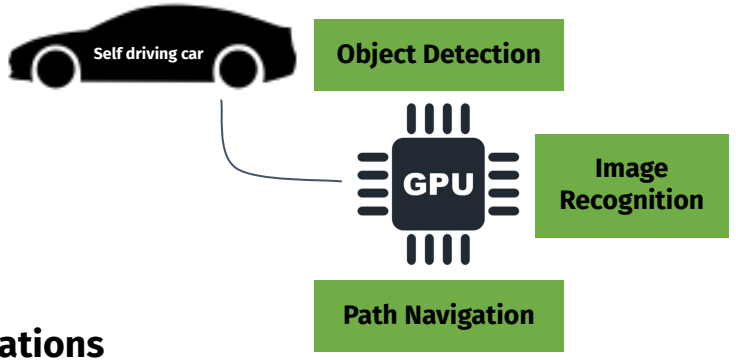


- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**
  - **Ex: Automated Cars are accelerated by automotive-grade GPUs**

Complication: limited power management in resource competing environment

What is a GPU?

Graphics Processing Unit (GPU)



- **High computation efficiency**
- **Accelerator for High Performance Computing (HPC) Applications**
  - **Ex: Automated Cars are accelerated by automotive-grade GPUs**

Complication: limited power management in resource competing environment

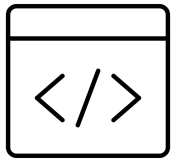
How can parallel applications practically share GPU resources without compromising performance and limiting power consumption?

Project Breakdown

Development Phases of an Energy Efficient Runtime Model

Performance Characterization
Bit-Hardware Mapping

Power Characterization
Benchmark System



Algorithm Development
Power Saving Policy Optimization

GPU Programming: Experimental Setup

Programming Tools and Frameworks

Getty Images

- **AMD Radeon MI50 Accelerator**
- **C++ and HIP Runtime API**

GPU Programming: Experimental Setup

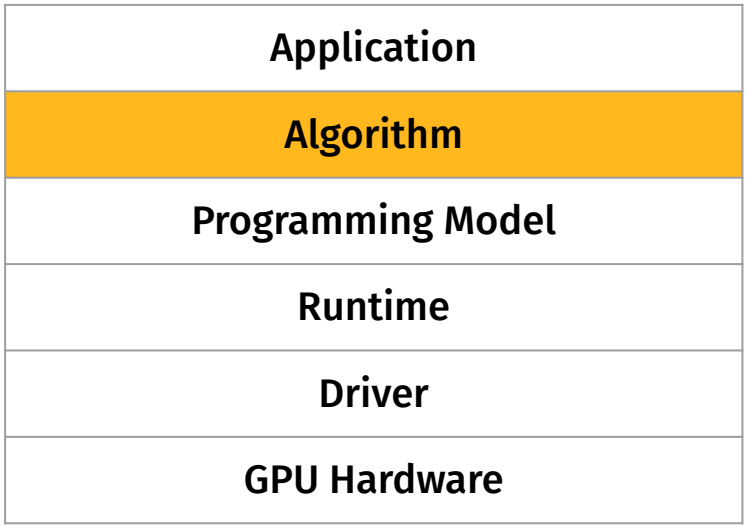
Programming Tools and Frameworks

Getty Images

- **AMD Radeon MI50 Accelerator**
- **C++ and HIP Runtime API**
- **hipExtStreamCreateWithCUMask() function**

GPU Programming: Experimental Setup

Programming Tools and Frameworks

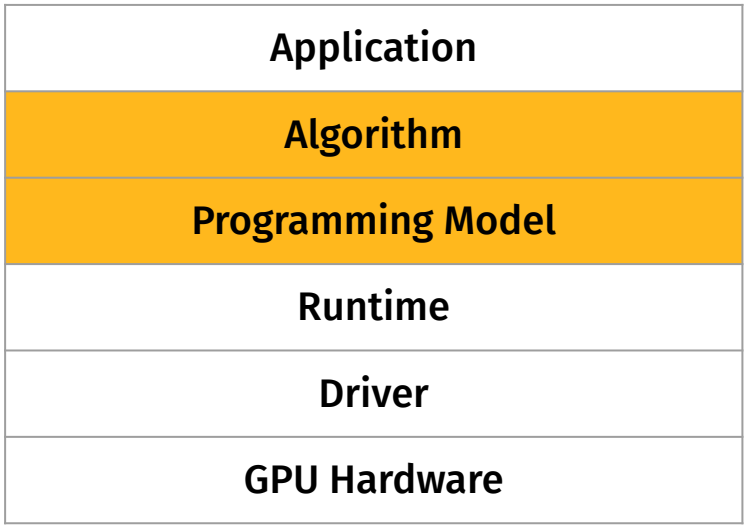


Software Stack

- **AMD Radeon MI50 Accelerator**
- **C++ and HIP Runtime API**
- **hipExtStreamCreateWithCUMask() function**

GPU Programming: Experimental Setup

Programming Tools and Frameworks

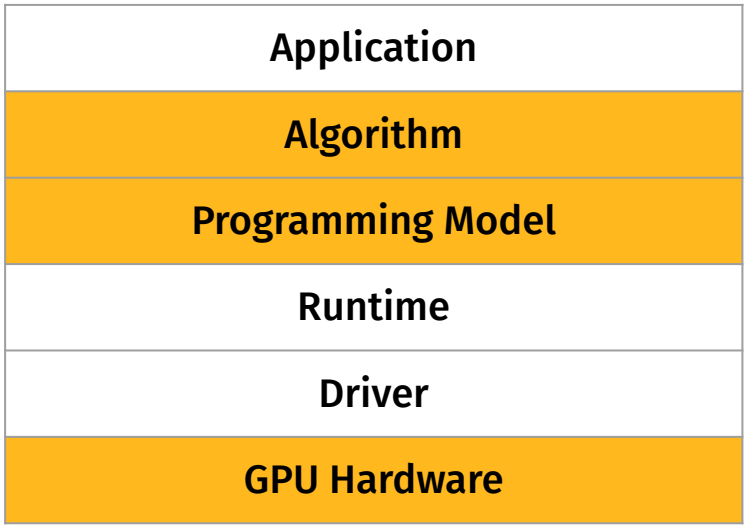


Software Stack

- **AMD Radeon MI50 Accelerator**
- **C++ and HIP Runtime API**
- **hipExtStreamCreateWithCUMask() function**

GPU Programming: Experimental Setup

Programming Tools and Frameworks

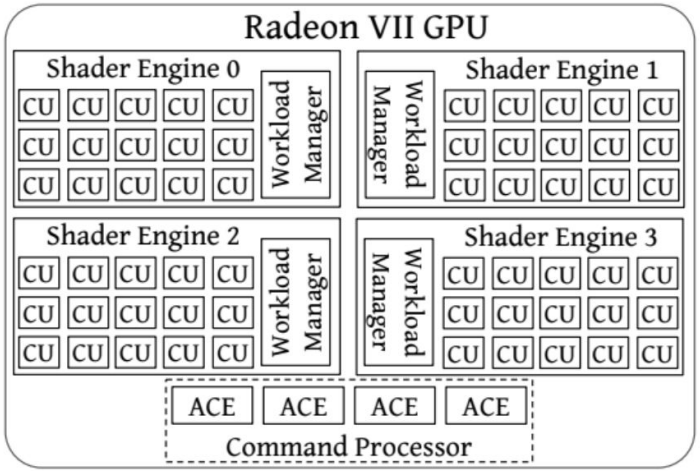


Software Stack

- AMD Radeon MI50 Accelerator
- C++ and HIP Runtime API
- hipExtStreamCreateWithCUMask() function

Understanding GPU Architecture

Overview of the Hardware

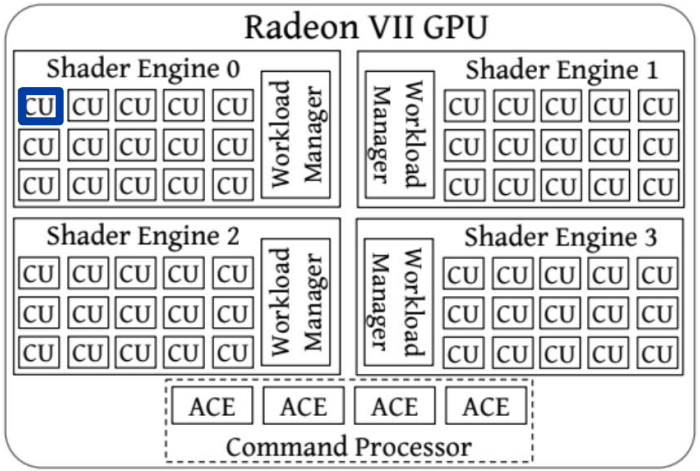


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

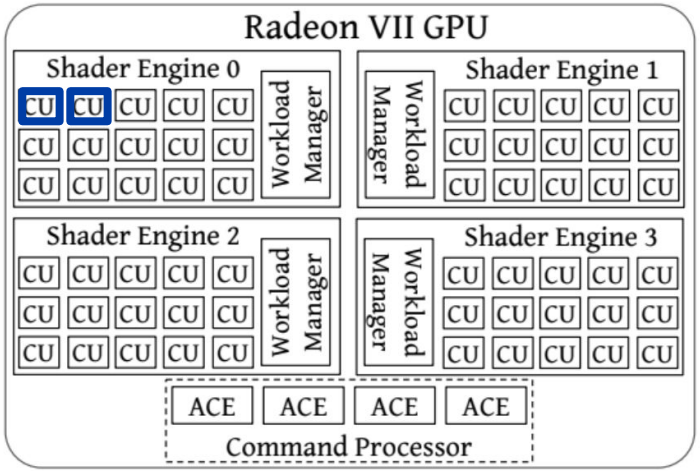


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

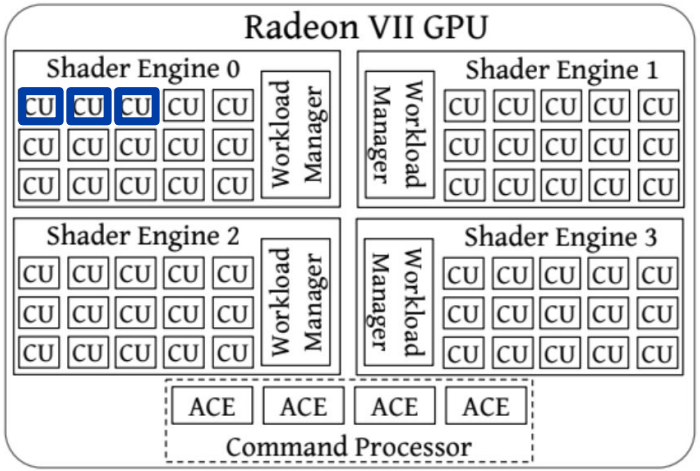


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

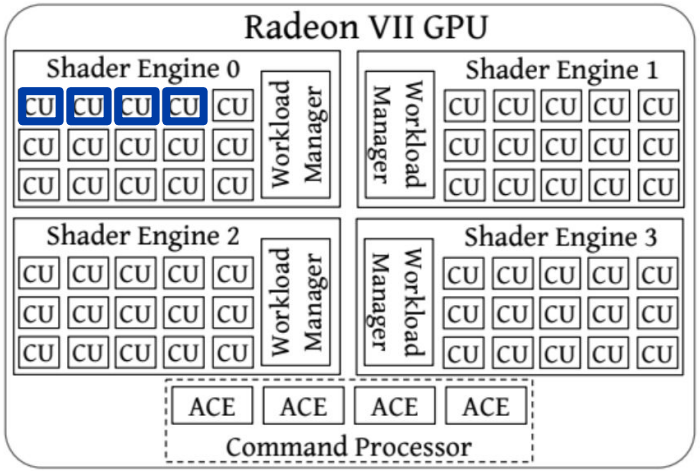


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

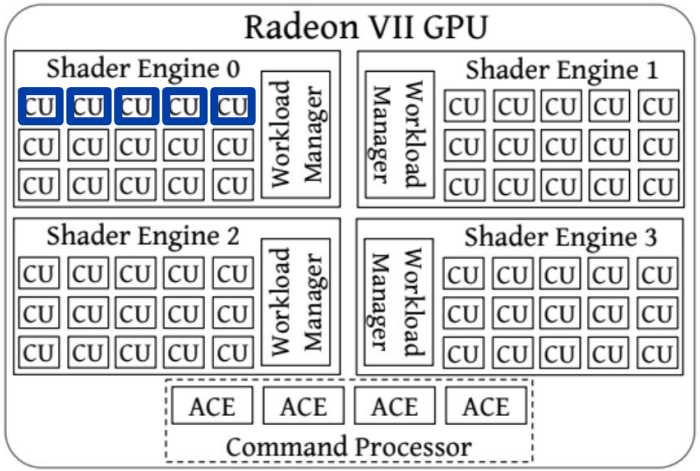


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

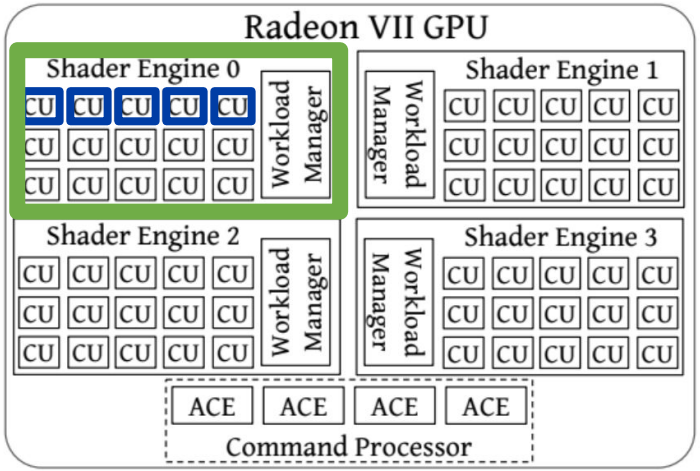


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

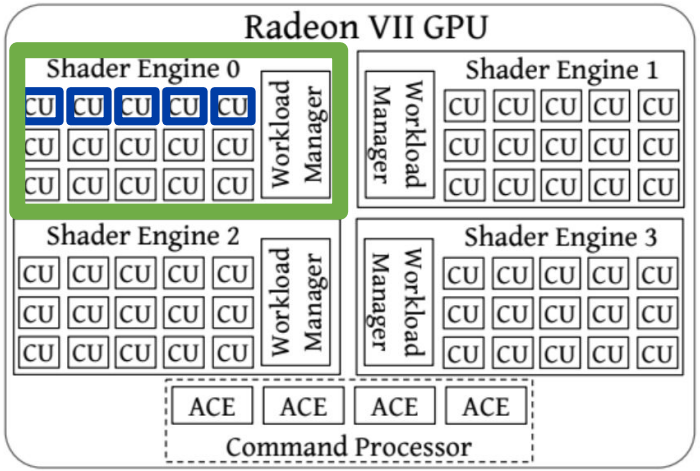


Otterness & Anderson

- **Role of Compute Units (CU)**

Understanding GPU Architecture

Overview of the Hardware

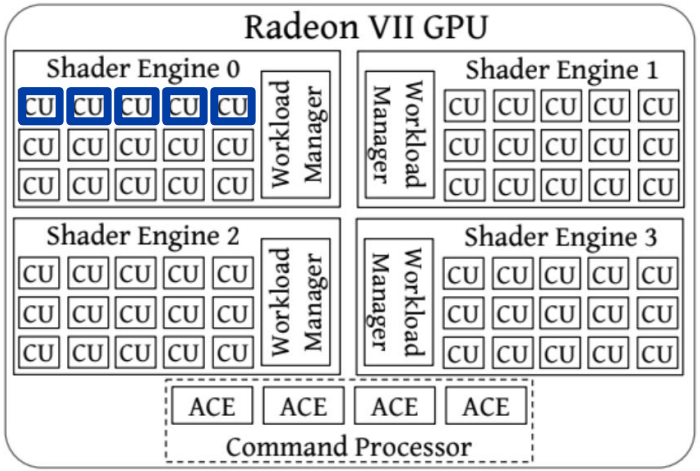


Otterness & Anderson

- **Role of Compute Units (CU)**
  - **Total: 60 CUs/60 Resources**

Understanding GPU Architecture

Overview of the Hardware

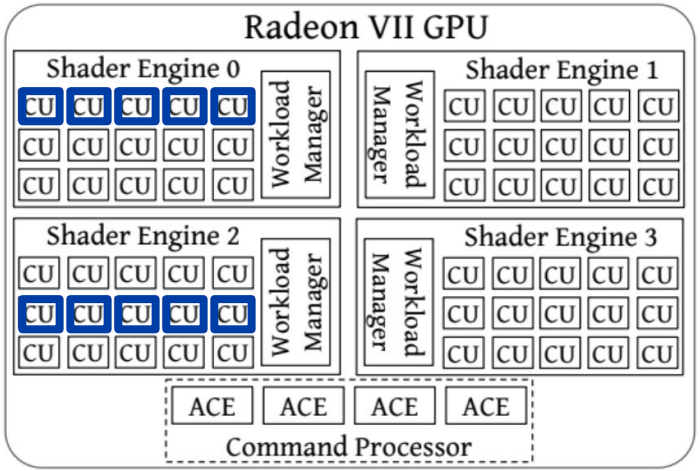


Otterness & Anderson

- **Role of Compute Units (CU)**
  - **Total: 60 CUs/60 Resources**
- **Manipulating workloads through CU Masking**
  - **CU Masking:**
    - **Turning blocks on/off**

Understanding GPU Architecture

Overview of the Hardware

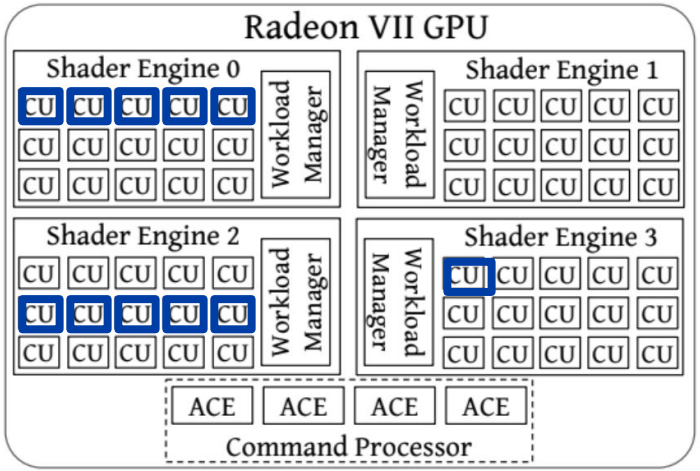


Otterness & Anderson

- **Role of Compute Units (CU)**
  - **Total: 60 CUs/60 Resources**
- **Manipulating workloads through CU Masking**
  - **CU Masking:**
    - **Turning blocks on/off**

Understanding GPU Architecture

Overview of the Hardware

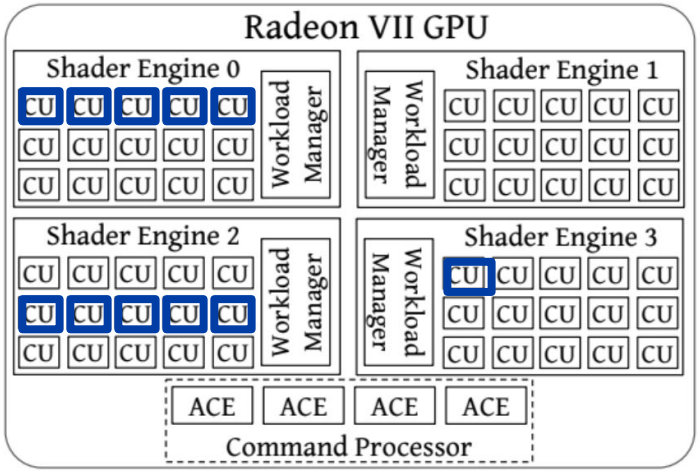


Otterness & Anderson

- **Role of Compute Units (CU)**
  - **Total: 60 CUs/60 Resources**
- **Manipulating workloads through CU Masking**
  - **CU Masking:**
    - **Turning blocks on/off**

Understanding GPU Architecture

Overview of the Hardware



Otterness & Anderson

- **Role of Compute Units (CU)**
  - **Total: 60 CUs/60 Resources**
- **Manipulating workloads through CU Masking**
  - **CU Masking:**
    - **Turning blocks on/off**

How viable are CU Masking techniques in managing power?

Current Phase: Performance Characterization

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id

+4

+4

CU Mask Bit	SM_ID
1	1
2	17
3	32
4	49
5	2
6	18
7	33
8	50
9	3
10	19

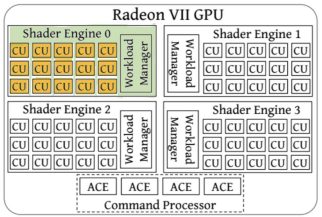
CU Mask Bit	SM_ID
11	34
12	51
13	4
14	20
15	35
16	52
17	5
18	21
19	36
20	53

CU Mask Bit	SM_ID
21	6
22	22
23	37
24	54
25	7
26	23
27	38
28	55
29	8
30	24

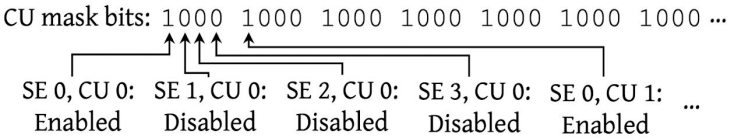
CU Mask Bit	SM_ID
31	39
32	56
33	9
34	25
35	40
36	57
37	10
38	26
39	41
40	58

CU Mask Bit	SM_ID
41	11
42	27
43	43
44	59
45	12
46	28
47	44
48	60
49	13
50	29

CU Mask Bit	SM_ID
51	45
52	61
53	14
54	30
55	46
56	62
57	15
58	31
59	47
60	63



GPU Representation



Mapping of CU Mask bits to SEs

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id

+4

+4

CU Mask Bit	SM_ID
1	1
2	17
3	32
4	49
5	2
6	18
7	33
8	50
9	3
10	19

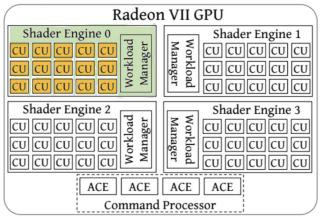
CU Mask Bit	SM_ID
11	34
12	51
13	4
14	20
15	35
16	52
17	5
18	21
19	36
20	53

CU Mask Bit	SM_ID
21	6
22	22
23	37
24	54
25	7
26	23
27	38
28	55
29	8
30	24

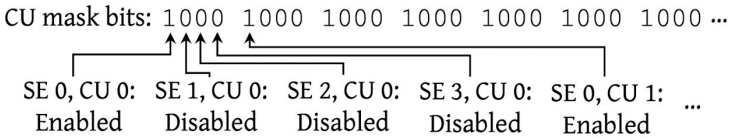
CU Mask Bit	SM_ID
31	39
32	56
33	9
34	25
35	40
36	57
37	10
38	26
39	41
40	58

CU Mask Bit	SM_ID
41	11
42	27
43	43
44	59
45	12
46	28
47	44
48	60
49	13
50	29

CU Mask Bit	SM_ID
51	45
52	61
53	14
54	30
55	46
56	62
57	15
58	31
59	47
60	63



GPU Representation



Mapping of CU Mask bits to SEs

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id

+4

+4

CU Mask Bit	SM_ID
1	1
2	17
3	32
4	49
5	2
6	18
7	33
8	50
9	3
10	19

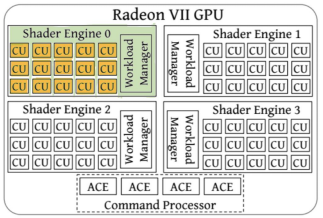
CU Mask Bit	SM_ID
11	34
12	51
13	4
14	20
15	35
16	52
17	5
18	21
19	36
20	53

CU Mask Bit	SM_ID
21	6
22	22
23	37
24	54
25	7
26	23
27	38
28	55
29	8
30	24

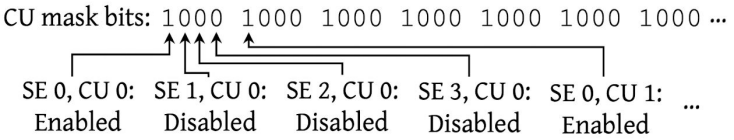
CU Mask Bit	SM_ID
31	39
32	56
33	9
34	25
35	40
36	57
37	10
38	26
39	41
40	58

CU Mask Bit	SM_ID
41	11
42	27
43	43
44	59
45	12
46	28
47	44
48	60
49	13
50	29

CU Mask Bit	SM_ID
51	45
52	61
53	14
54	30
55	46
56	62
57	15
58	31
59	47
60	63



GPU Representation



Mapping of CU Mask bits to SEs

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id

+4

+4

CU Mask Bit	SM_ID
1	1
2	17
3	32
4	49
5	2
6	18
7	33
8	50
9	3
10	19

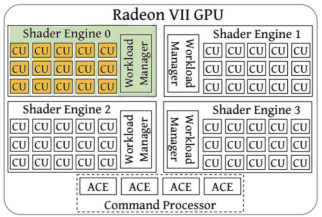
CU Mask Bit	SM_ID
11	34
12	51
13	4
14	20
15	35
16	52
17	5
18	21
19	36
20	53

CU Mask Bit	SM_ID
21	6
22	22
23	37
24	54
25	7
26	23
27	38
28	55
29	8
30	24

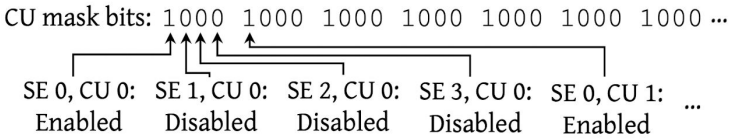
CU Mask Bit	SM_ID
31	39
32	56
33	9
34	25
35	40
36	57
37	10
38	26
39	41
40	58

CU Mask Bit	SM_ID
41	11
42	27
43	43
44	59
45	12
46	28
47	44
48	60
49	13
50	29

CU Mask Bit	SM_ID
51	45
52	61
53	14
54	30
55	46
56	62
57	15
58	31
59	47
60	63



GPU Representation



Mapping of CU Mask bits to SEs

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id

CU Mask Bit	SM_ID
1	1
2	17
3	32
4	49
5	2
6	18
7	33
8	50
9	3
10	19

+4

+4

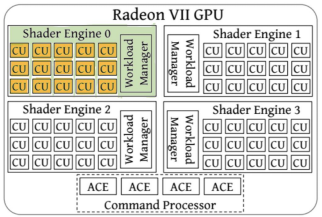
CU Mask Bit	SM_ID
11	34
12	51
13	4
14	20
15	35
16	52
17	5
18	21
19	36
20	53

CU Mask Bit	SM_ID
21	6
22	22
23	37
24	54
25	7
26	23
27	38
28	55
29	8
30	24

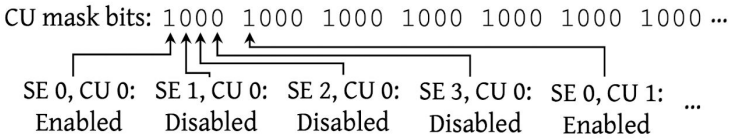
CU Mask Bit	SM_ID
31	39
32	56
33	9
34	25
35	40
36	57
37	10
38	26
39	41
40	58

CU Mask Bit	SM_ID
41	11
42	27
43	43
44	59
45	12
46	28
47	44
48	60
49	13
50	29

CU Mask Bit	SM_ID
51	45
52	61
53	14
54	30
55	46
56	62
57	15
58	31
59	47
60	63



GPU Representation



Mapping of CU Mask bits to SEs

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id

+4

+4

CU Mask Bit	SM_ID
1	1
2	17
3	32
4	49
5	2
6	18
7	33
8	50
9	3
10	19

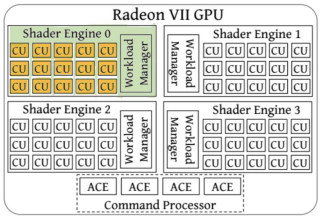
CU Mask Bit	SM_ID
11	34
12	51
13	4
14	20
15	35
16	52
17	5
18	21
19	36
20	53

CU Mask Bit	SM_ID
21	6
22	22
23	37
24	54
25	7
26	23
27	38
28	55
29	8
30	24

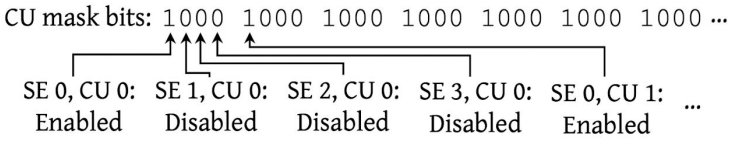
CU Mask Bit	SM_ID
31	39
32	56
33	9
34	25
35	40
36	57
37	10
38	26
39	41
40	58

CU Mask Bit	SM_ID
41	11
42	27
43	43
44	59
45	12
46	28
47	44
48	60
49	13
50	29

CU Mask Bit	SM_ID
51	45
52	61
53	14
54	30
55	46
56	62
57	15
58	31
59	47
60	63



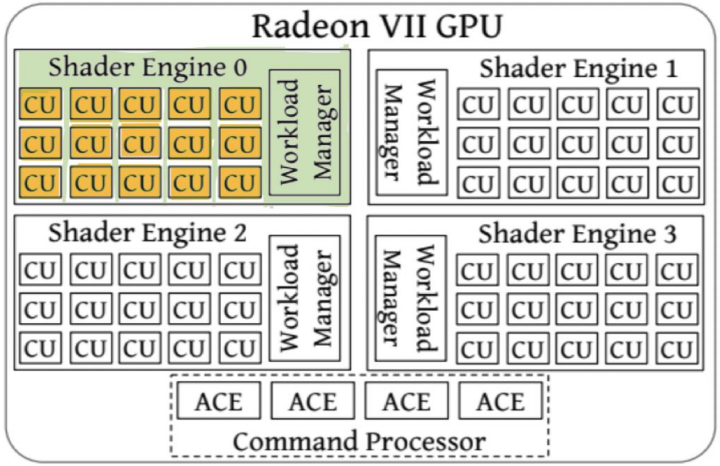
GPU Representation



Mapping of CU Mask bits to SEs

CU Mask-SM_id Bit Mappings

Profiling individual CUs to its corresponding active CU Id



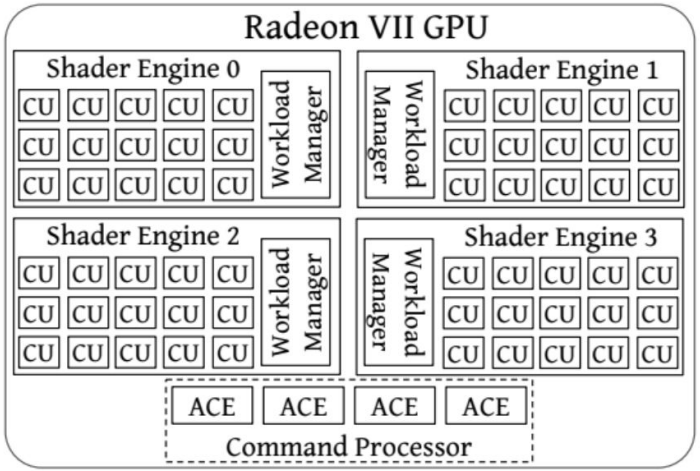
GPU Representation

We can now manually activate each of the CU masks since we know which bit position activates a specific CU ID

Next Phase: Power Characterization and Allocation Policy Optimization

Power Savings Policy

Power Characterization & Allocation Policy Optimization

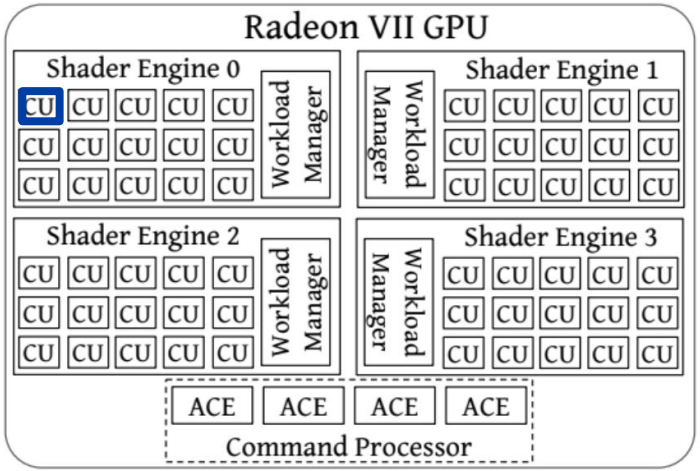


Otterness & Anderson

- **Power Monitoring System**

Power Savings Policy

Power Characterization & Allocation Policy Optimization

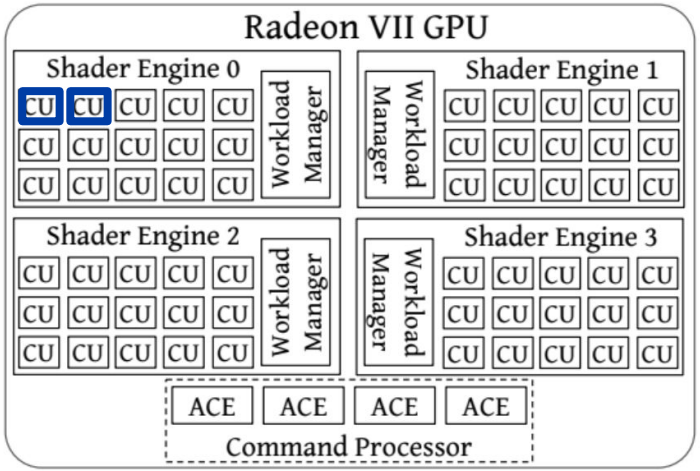


Otterness & Anderson

- **Power Monitoring System**

Power Savings Policy

Power Characterization & Allocation Policy Optimization

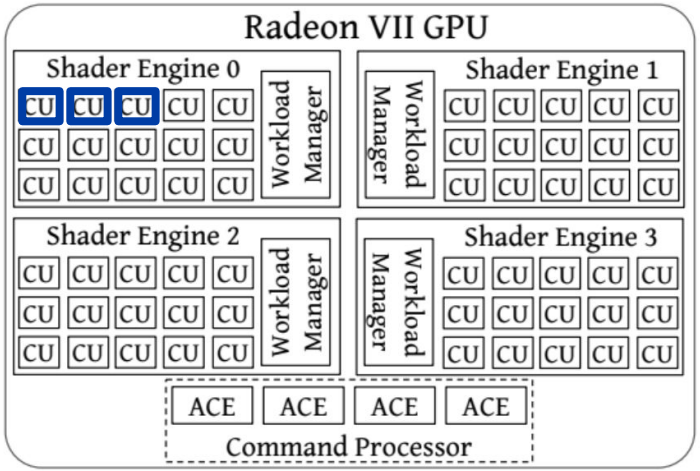


Otterness & Anderson

- **Power Monitoring System**

Power Savings Policy

Power Characterization & Allocation Policy Optimization

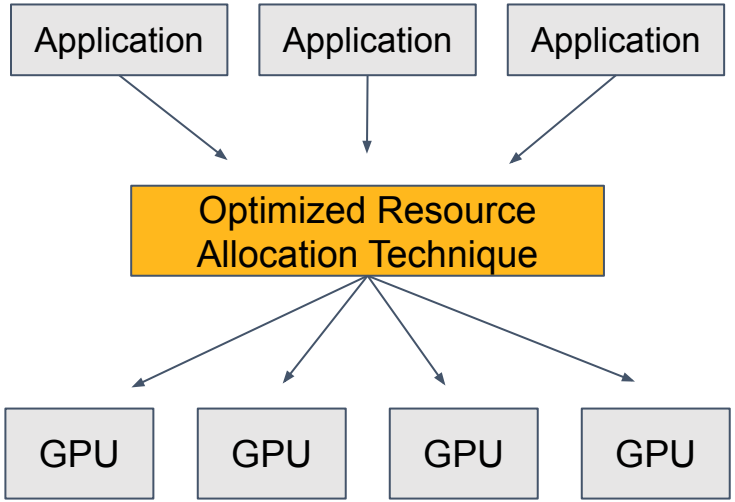


Otterness & Anderson

- **Power Monitoring System**

Power Saving Policy

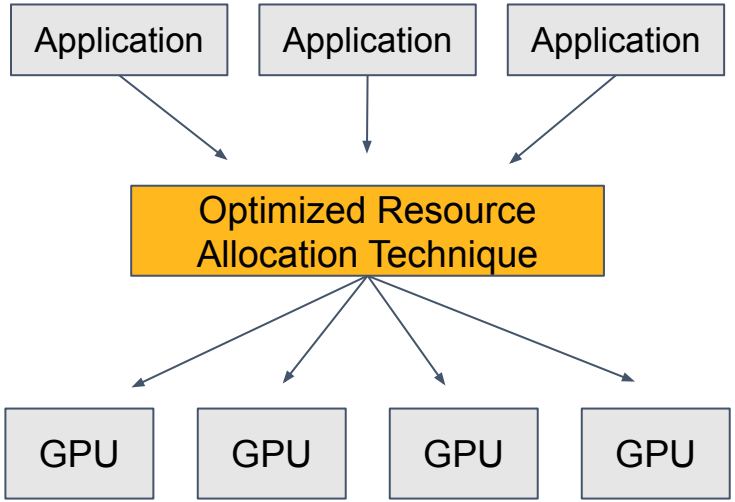
Power Characterization & Allocation Policy Optimization



- **Power Monitoring System**
- **Algorithm Development**

Power Savings Policy

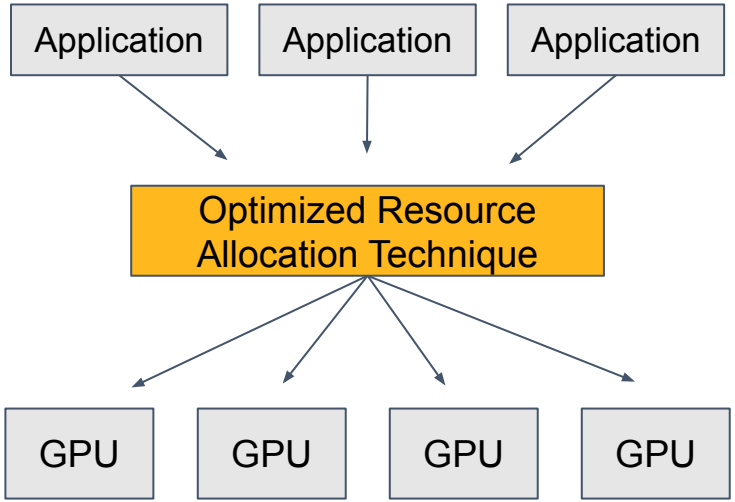
Power Characterization & Allocation Policy Optimization



- **Power Monitoring System**
- **Algorithm Development**
- **Runtime Evaluation**
  - **Gigaflops: floating point operations per second**

Power Savings Policy

Power Characterization & Allocation Policy Optimization



- **Power Monitoring System**
- **Algorithm Development**
- **Runtime Evaluation**

How does the GPU's power consumption change in relation to the number of active CUs?

Moving Forward

Predicting future trends and applications

- **Runtime model for other Parallel Computing Systems**

Moving Forward

Predicting future trends and applications

- **Runtime model for other Parallel Computing Systems**
- **Sustainability of GPUs as an High Performance Computing (HPC) accelerator**

Reference Papers

Increasing Energy Efficiency of GPUs Through Hardware Resource Partitioning and Masking

- **Chow, Marcus N. (2018). Characterizing Dynamic Frequency and Thread Blocking Scaling in GPUs: Challenges and Opportunities.**
- **Anderson, James H. Otterness, Nathan. Exploring AMD GPU Scheduling Details by Experimenting With "Worst Practices". *International Conference on Real-Time Networks and Systems (RTNS)***
- **Wikipedia.**

Acknowledgements

Increasing Energy Efficiency of GPUs Through Hardware Resource Partitioning and Masking

- Prof. Daniel Wong
- Marcus Chow, ENCS Ph.D.
- Prof. Ludwig Bartels
- Rebecca Ryan
- Marissa Moreno
- MacREU
- NSF
- University of California, Riverside