Understanding and Optimizing Deep Learning Cold-Start Latency on Edge Devices

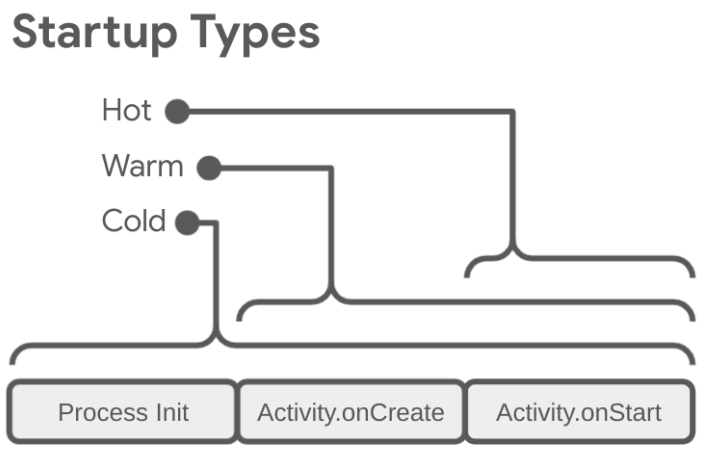
MobiSys'23

Rongjie Yi, Ting Cao, Ao Zhou, Xiao Ma, Shangguang Wang, Mengwei Xu

BUPT, Microsoft Research

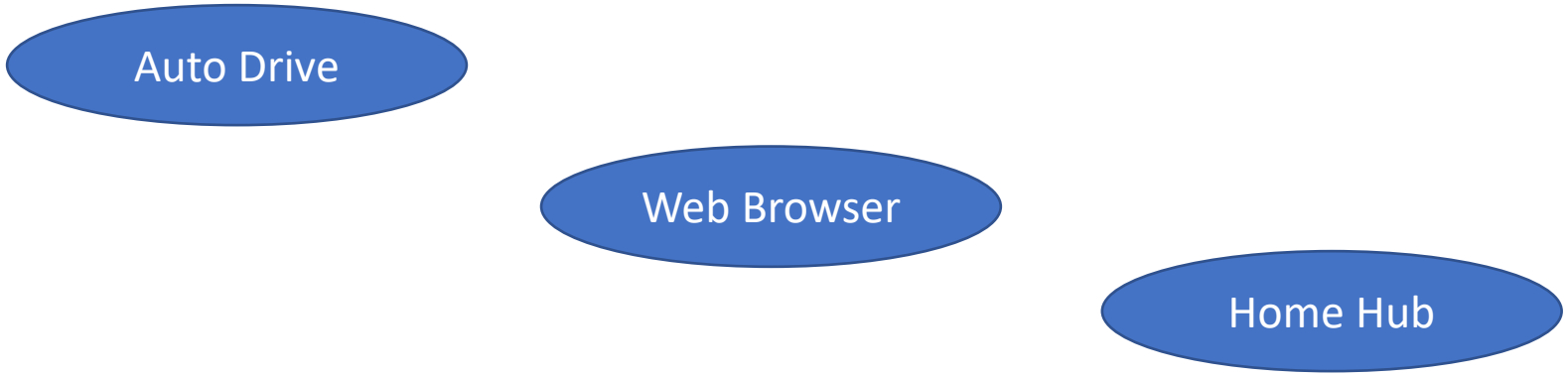
Introduction

- Cold Inference vs. Warm Inference



Introduction

- Why cold-start inference?
  - The number of DNNs per device is explosively increasing.
  - The complexity of DNNs are increasing as well.
  - User experience and application QoE.



Introduction

- Existing frameworks are not ready to boost cold inference as fast as warm inference.

```
char parampath[256];
sprintf(parampath, MODEL_DIR "%s.param", comment);
net.load_param(parampath);

DataReaderFromEmpty dr;
net.load_model(dr);

const std::vector<const char*>& input_names = net.input_names();
const std::vector<const char*>& output_names = net.output_names();

if (g_enable_cooling_down)
{
    // sleep 10 seconds for cooling down SOC  :(
    ncnn::sleep(10 * 1000);
}

ncnn::Mat out;
```

```
// warm up
for (int i = 0; i < g_warmup_loop_count; i++)
{
    ncnn::Extractor ex = net.create_extractor();
    ex.input(input_names[0], in);
    ex.extract(output_names[0], out);
}

double time_min = DBL_MAX;
double time_max = -DBL_MAX;
double time_avg = 0;

for (int i = 0; i < g_loop_count; i++)
{
    double start = ncnn::get_current_time();

    {
        ncnn::Extractor ex = net.create_extractor();
        ex.input(input_names[0], in);
        ex.extract(output_names[0], out);
    }

    double end = ncnn::get_current_time();

    double time = end - start;

    time_min = std::min(time_min, time);
    time_max = std::max(time_max, time);
    time_avg += time;
}
```

Introduction

- A breakdown of ResNet-50 cold inference latency

Device Platform Processor	Google Pixel 5 CPU	Jetson TX2 GPU
Weights reading	36.52 ms	43.03 ms
Memory allocation	1.34 ms	0.69 ms
GPU preparation	-	3004.01 ms
Weights transformation	1135.28 ms	1616.84 ms
Model execution	190.12 ms	802.77 ms
Total cold inference	1363.23 ms	5467.48 ms
Warm inference	185.82 ms	137.02 ms

Introduction

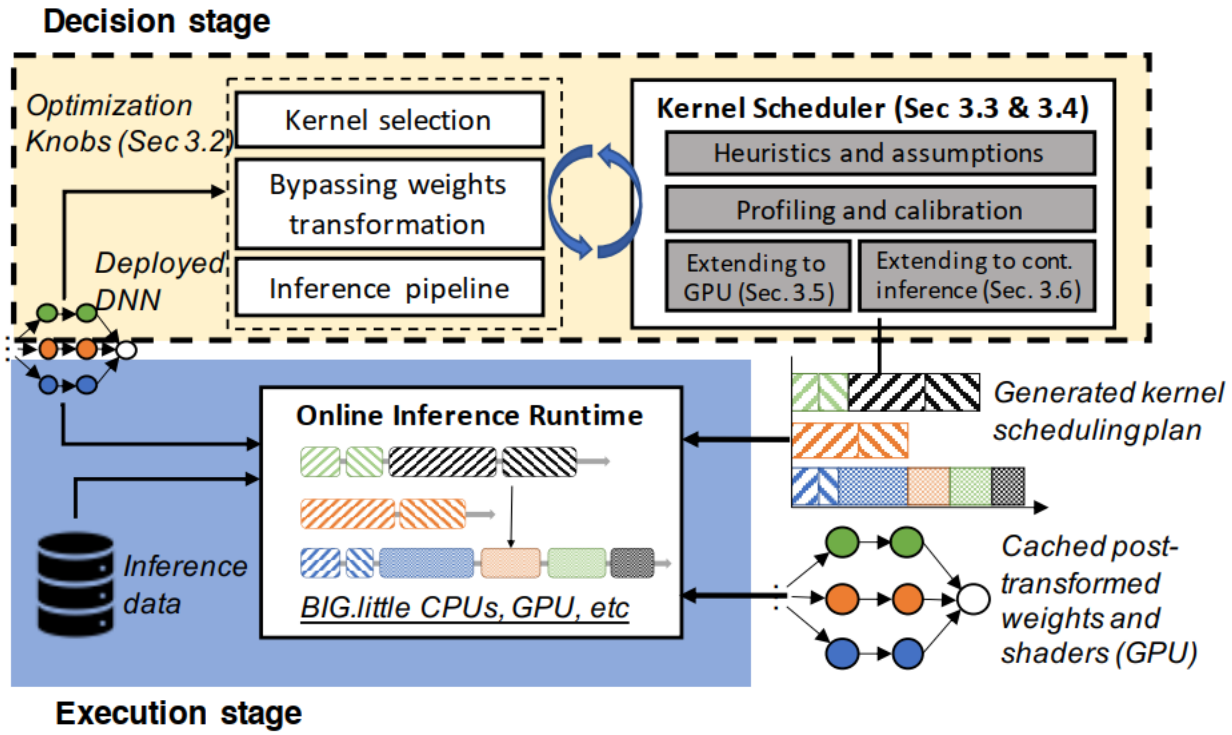
- A breakdown of ResNet-50 cold inference latency

Device Platform Processor	Google Pixel 5 CPU	Jetson TX2 GPU
Weights reading	36.52 ms	43.03 ms
Memory allocation	1.34 ms	0.69 ms
GPU preparation	-	3004.01 ms
Weights transformation	1135.28 ms	1616.84 ms
Model execution	190.12 ms	802.77 ms
Total cold inference	1363.23 ms	5467.48 ms
Warm inference	185.82 ms	137.02 ms

Weights reading: reading weights from disk.
Memory allocation: requesting memory from OS.
Weights transformation: converting raw weights into the proper format.
Model execution: forward process.

Introduction

- System overview

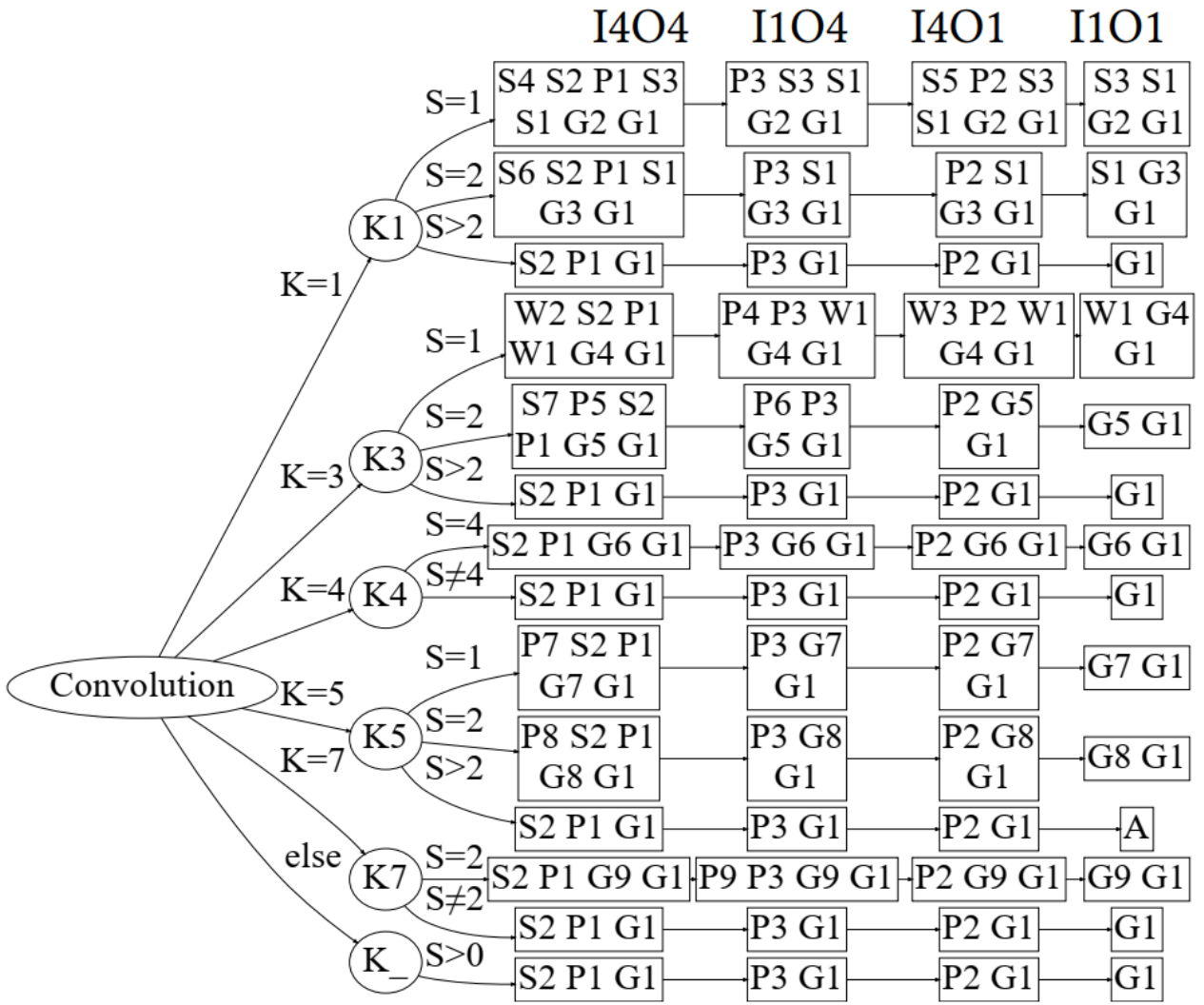


Kernel selection: select the optimal kernel for operator with the minimum latency in cold inference.
Weights transformation bypassing: Pre-saving transformed weights.
Inference pipeline: Pipeline processing stages by multiple cores on device.

System Design

- Kernel selection
  - One operator has multiple kernels to implement

S1:sgemm S2:sgemm_pack4 S3:1x1s1_sgemm S4:1x1s1_sgemm_pack4
S5:1x1s1_sgemm_pack4to1 S6:1x1s2_sgemm_pack4
S7:3x3s2_sgemm_pack4 W1:3x3s1_winograd W2:3x3s1_winograd_pack4
W3:3x3s1_winograd_pack4to1 P1:pack4 P2:pack4to1 P3:pack1to4
P4:3x3s1_pack1to4 P5:3x3s2_pack4 P6:3x3s2_pack1to4 P7:5x5s1_pack4
P8:5x5s2_pack4 P9:7x7s2_pack1to4 G1:vanilla G2:1x1s1 G3:1x1s2
G4:3x3s1 G5:3x3s2 G6:4x4s4 G7:5x5s1 G8:5x5s2 G9:7x7s2



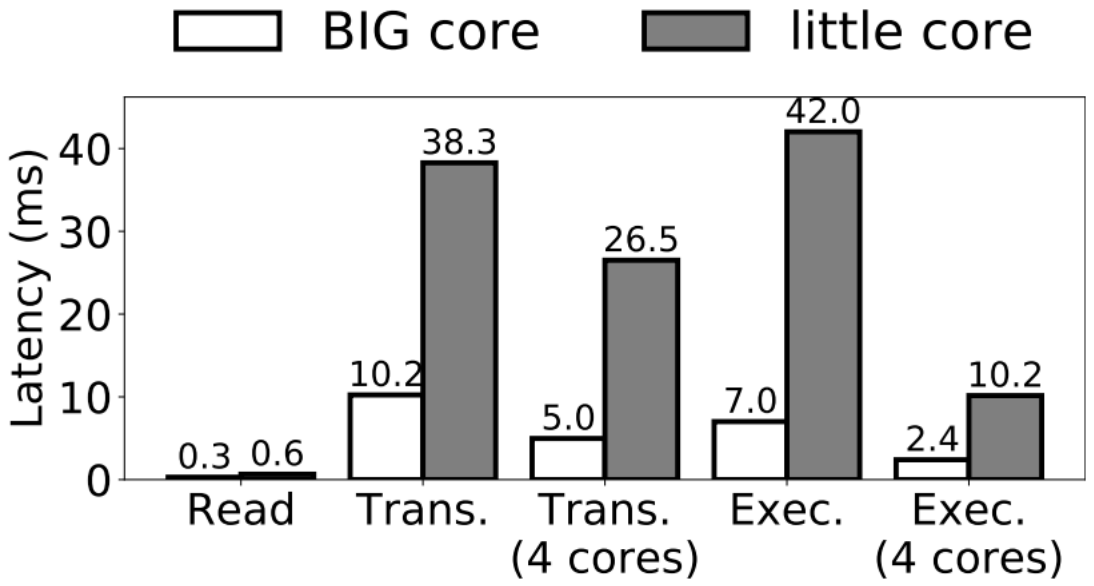
System Design

- Kernel selection
  - Different kernel has different processing latency in each stage.

Kernels	Cold Inference Time (ms)			
	Read Raw	Weights Trans.	Read Cache	Execution
3x3s1-winograd-pack4	0.70	38.23	5.23	2.98
sgemm-pack4	0.70	2.21	0.70	8.14
pack4	0.70	2.22	0.70	18.63
3x3s1-winograd	0.70	65.67	4.12	3.37
3x3s1	0.70	0.00	0.70	8.01
general	0.70	0.00	0.70	87.12

System Design

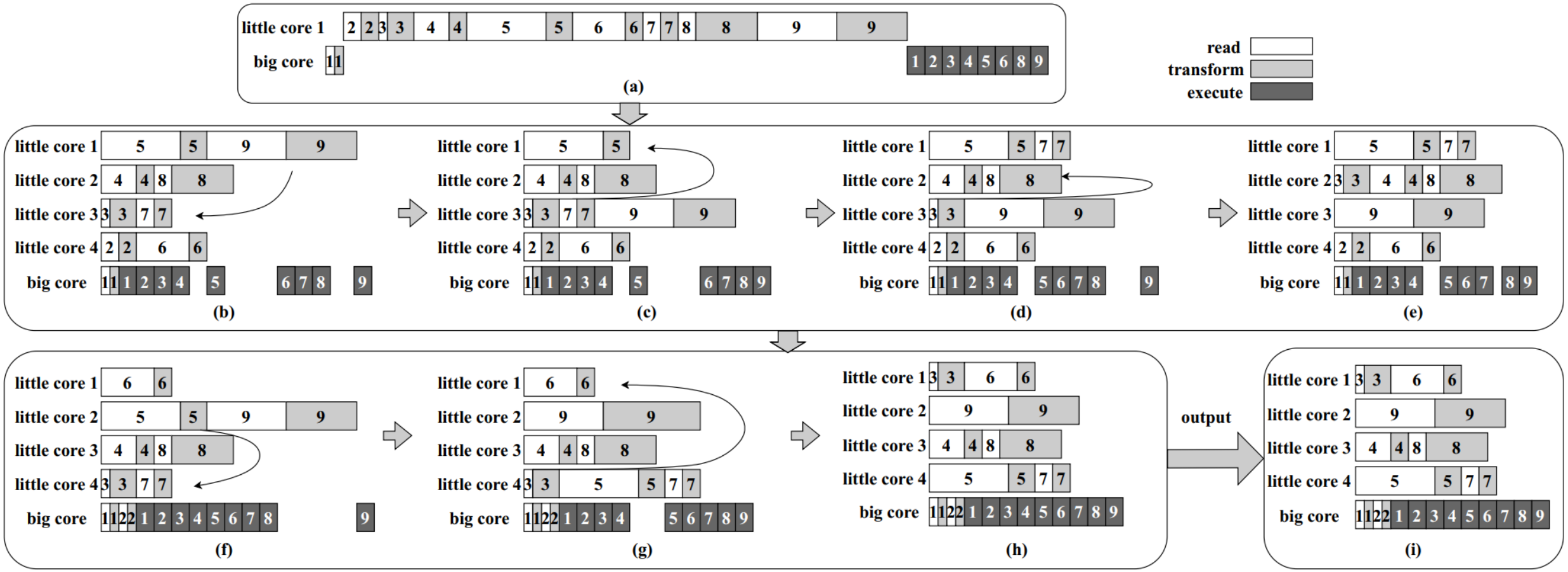
- Kernel selection
  - Scheduling the selection of kernels relied on different hardware



System Design

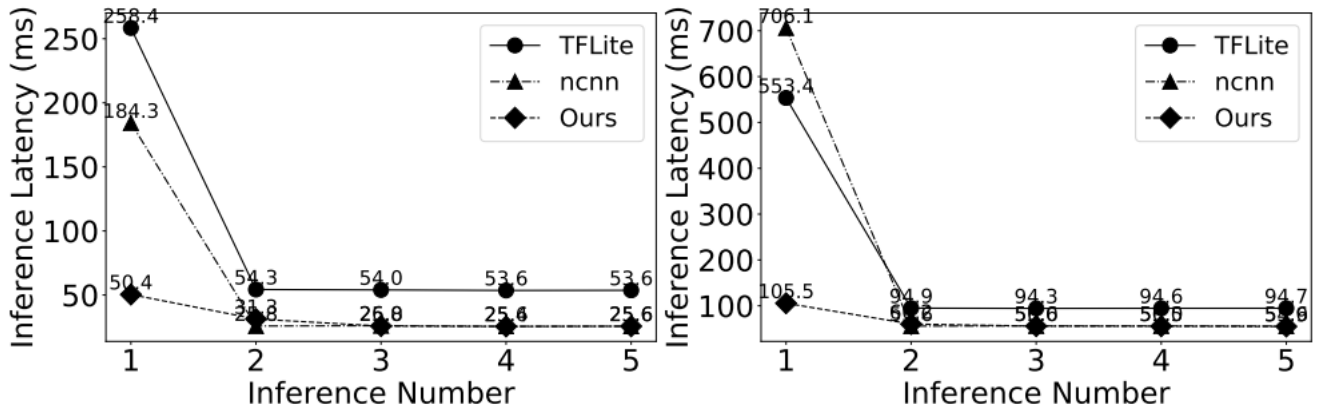
Execution only on big cores.
Others can be placed to all cores.
An iterative and heuristic scheduling.

- Kernel scheduling



System Design

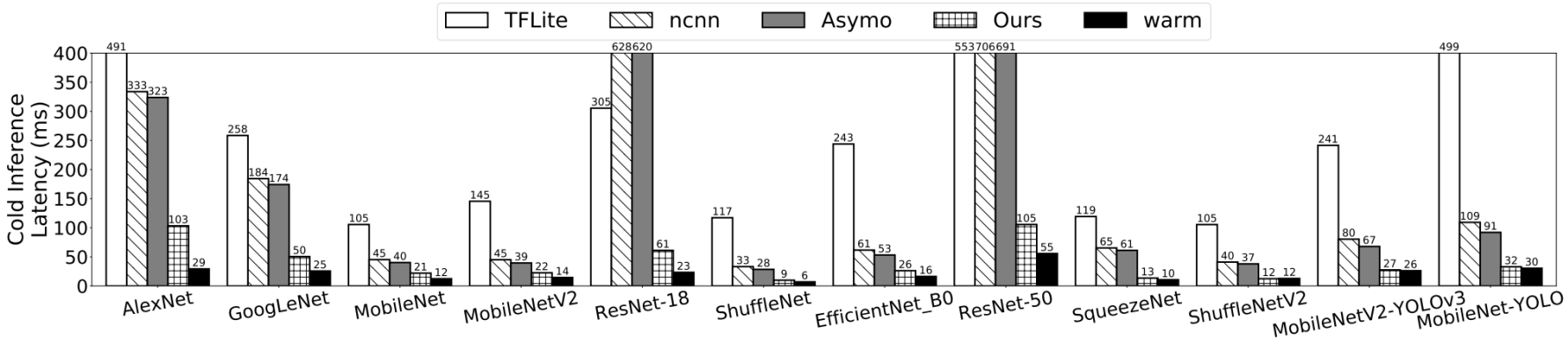
- Extending to GPU
  - GPU is viewed as big core and CPU as little core.
- Extending to continuous inference
  - Cold and warm inference mode.
  - Prepare warm inference kernel in the idle of cold inference kernel selection.



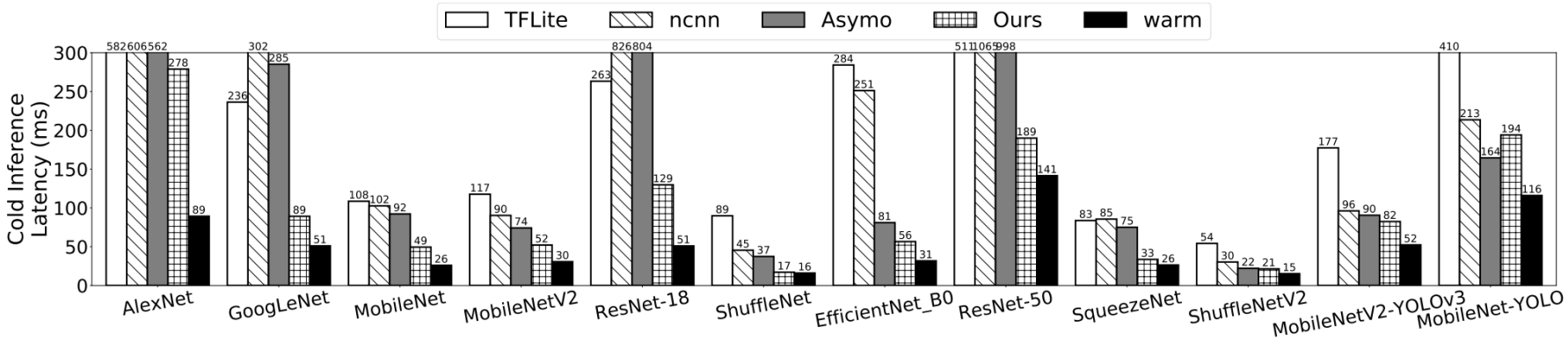
(a) GoogLeNet (b) ResNet50

Results

- Cold inference latency



(a) Meizu 16T CPU



(b) Google Pixel 5 CPU

Conclusion

- New environment ignored by inference framework: cold-inference.
- Constraints:
  - Scheduling is based on operator profiling.
  - Memory foot-print is large.

Thank You!

May 25, 2023

Presented by Mengyang Liu