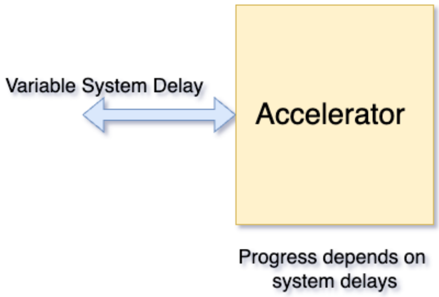
DFS Policy for ML Accelerators

Problem Objective: Design a dynamic DFS policy for ML Accelerators based on a given deadline (the computation should finish within this deadline).

Given a deadline for the complete execution of the accelerator, design a policy that puts the accelerator into low frequency modes when the accelerator progress is ahead of the deadline.

Accelerator Configuration: 64×64 PE array, 64KB SRAM each for IFMAP, OFMAP, FIL



Required Inputs for Calculating Progress:
System Cache Hit Delay, System Cache Miss Delay (Cache + DRAM delay), System Queueing delay, Scale-Sim DRAM Traces.

Assumptions:

System Clock (2 GHz) = 0.5 ns

Accelerator highest operating frequency = 500 MHz (2 ns)

Given deadline: 100 ms (10 FPS)

System Cache Delay ⇒ 15 system cycles = 7.5 ns

Off-chip DRAM Delay ⇒ 80 system cycles = 40 ns

System Queueing Delay ⇒ between 10 ns - 100 ns

DRAM Trace Data Example:

0: 10,11,12,13,14,15,16

1: 17,18,19,20,21,22,23

1. Study the scale-sim-accelergy infrastructure. The infrastructure integrates Scale-Sim Accelerator simulator with Energy Estimation model Accelergy.

   (scalesim-accelergy branch of scalesimv2 repo)

   It calculates the access counts of all the components in the accelerator at the end of the scale-sim simulation. Accelergy uses this access count to calculate the final energy. Run experiments to successfully generate traces and energy estimation for any network model on the given accelerator configuration. **(Deadline - 16th Feb)**

2. System Delay Estimation:

   For every line in the trace:

   Generate a random queuing delay - between 10 ns - 100 ns.

   Calculate the total System Delay (d1) which is Sum of random queueing delay and one delay out of System cache or DRAM delay (one is chosen randomly).

   Your line from the trace will now look like this:

   0: 10,11,12,13,14,15,16 (delay d1)

3. Progress Estimation:

   Keep track of the time elapsed in accelerator execution from the traces and the system delay.

   We will check the progress every few intervals (maybe after every 0.5 ms - this should be kept variable).

4. Compare the progress of the accelerator to the given deadline to decide how likely it is that the accelerator will meet its deadline.

   Parameters that can be used for this: Number of memory accesses completed, memory accesses left, time left from the deadline, etc. **(Deadline - 23 March)**

5. Implement a DFS policy in Scale-Sim that lowers the frequency of the accelerator if it is ahead of the required progress.

6. Once the execution is complete, calculate the total energy of the accelerator with DFS and compare it with the energy without DFS. **(Deadline - 26 April)**

Ideas for Implementation of DFS in Scale-Sim using Traces

We will implement DFS in scale-sim by post-processing the traces taking into account frequency changes and manipulating the final access count per interval.

Assume the following DFS levels: 500 MHz (2 ns) - 400 MHz (2.5 ns) - 300 MHz (3.3 ns) - 200 MHz (5 ns) - 100 MHz (10 ns) - 50 MHz (20 ns)

Lets Consider the traces for 10 consecutive cycles generated at assumed highest operating frequency of 500 MHz from Scale-Sim.

Current Compute Time = 10 cycles or 20 ns @ clock period = 2 ns

Let Interval time = 20 ns

Access Count in one interval without DFS = 60 addresses in 20 ns

Now Lets assume we change to frequency 400 MHz at 8 ns and to 200 MHz at 16ns

The updated post processing of trace will look like this:

  Cycle 0: 6 addresses (2 ns)

  Cycle 1: 6 addresses (4 ns)

  Cycle 2: 6 addresses (6 ns)

  Cycle 3: 6 addresses (8 ns)

  Cycle 4: 6 addresses (10.5 ns)

  Cycle 5: 6 addresses (13 ns)

  Cycle 6: 6 addresses (15.5 ns)

  Cycle 7: 6 addresses (20.5 ns)

  Cycle 8: 6 addresses (25.5 ns)

  Cycle 9: 6 addresses (30.5 ns)

  Updated Compute Time = 30.5 ns

Access Count in first Interval = 42 addresses

Access count in second interval = 18 addresses

⇒ The power trace will change accordingly.

Area, Power and Energy Number for Accelerator components at 500MHz:

ℹ️ **64 KB SRAM Bank**

Dynamic energy/access: 0.079079 nJ

Total leakage power of a bank: 75.3714 mW

Area: 0.44285 mm^2

ℹ️ **One PE unit**

Area: 5584.68 μm^2

Leakage Power/activation: 0.053 mW

Dynamic Energy/activation: 10.06 pJ