## 2. Measuring Power using perf/RAPL

Run the following 4 benchmarks, gathering the Energy and time values from perf for energycores, energy-gpu, energy-pkg and energy-dram. Use these values to calculate the average power for each.

Make a table with the 4 rows being cores, gpu, pkg, ram and the columns being the 4 benchmarks,list the average Power from each.

| perf | benchmarks | | | |
|---|---|---|---|---|
| | -sleep | -stream | -matrix-matrix multiply | -iozone |
| Cores | 0.002J/s | 12.9J/s | 15.57J/s | 0.58J/s |
| Gpu | 0J/s | 0J/s | 0J/s | 0J/s |
| Pkg | 3.88J/s | 21.8J/s | 23.53J/s | 4.97J/s |
| ram | 0.652J/s | 2.54J/s | 0.9J/s | 0.74J/s |

Answer the following questions:

### (a) Which benchmark causes the cores to use the highest average power?

Matrix matrix multiply

### (b) Which benchmark causes the RAM to use the highest average power?

Stream

### (c) Did any of the benchmarks use GPU power? Why not?

No, none of the benchmarks use GPU power.
Because sleep is shutting off the screen, so it would not use GPU power.

The STREAM benchmark is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels. So the GPU is not used in this benchmark.

The matrix and matrix multiply is mainly using the CPU and does not use the GPU.

IOzone is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. So it does not use the GPU.

## 3. Calculating Energy-Delay and Energy-Delay-Squared

Create a table that for each thread count (1,2,4,8) shows the elapsed time, the Energy in Joules, the Energy-Delay value, and the Energy-Delay-Squared value.

| | threads | | | |
|---|---|---|---|---|
| | 1 | 2 | 4 | 8 |
| elapsed time | 168.3 | 136.8 | 131.7 | 200.7 |
| energy in Joules | 2854.56 | 3414.78 | 4199.93 | 7210.66 |

| | | | | |
|---|---|---|---|---|
| energy-delay value=time*energy | 480422.4 | 467141.9 | 553130.8 | 1447179.5 |
| energy delay squared value=time^2*energy | 80855098 | 63905012 | 72847323 | 290448918 |

Answer the following questions:

(a) Which thread count has the fastest time?

Thread 4 counts the fastest time.

(b) Which thread count has the lowest energy?

Thread 1 has the lowest energy.

(c) Which thread count has the lowest energy-delay?

Thread 2 has the lowest energy-delay.

(d) Which thread count has the lowest energy-delay2?

Thread 2 has the lowest energy-delay^2.

(e) How well does this benchmark scale when adding additional threads? What could explain the scaling behavior you see?

When the threads increase to 2, the elapsed time drop down dramatically and when it reach to 4, the time does not improve a lot. But when the threads increase to 8, the time cost more. The time decrease because using more threads helps to compute the data so the time decrease. But it will increase because of overhead that manage the switching the cores and threads, it will not save time and cost more time doing these, so the time increases.