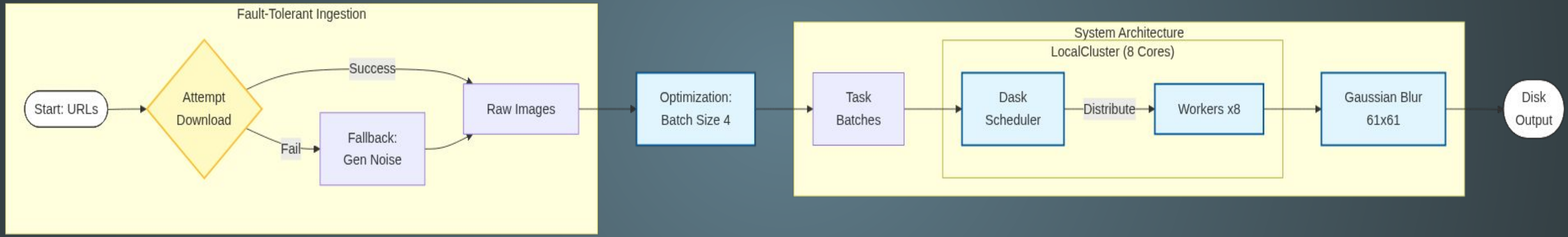
ParaBlur: Benchmarking High-Throughput Image Anonymization through Reproducible Study of Fault-Tolerant Dask Orchestration vs. Sequential Architectures

Ahmed Lamha

22301148

CSE449

Methodology



For Baseline Control, OpenCV internal multithreading **disabled** to ensure a true single-core comparison during sequential run

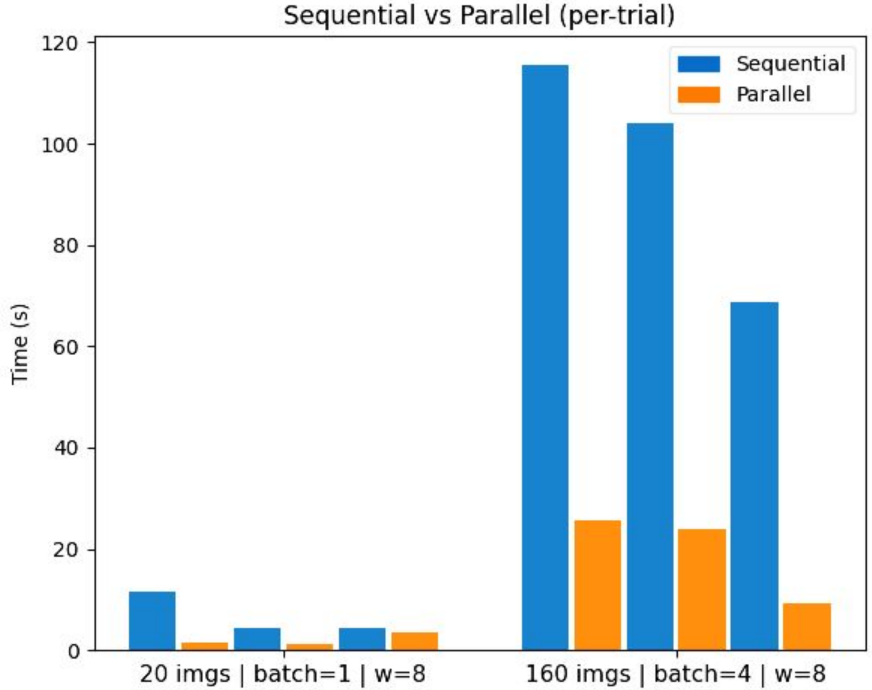
Phase 2 Results

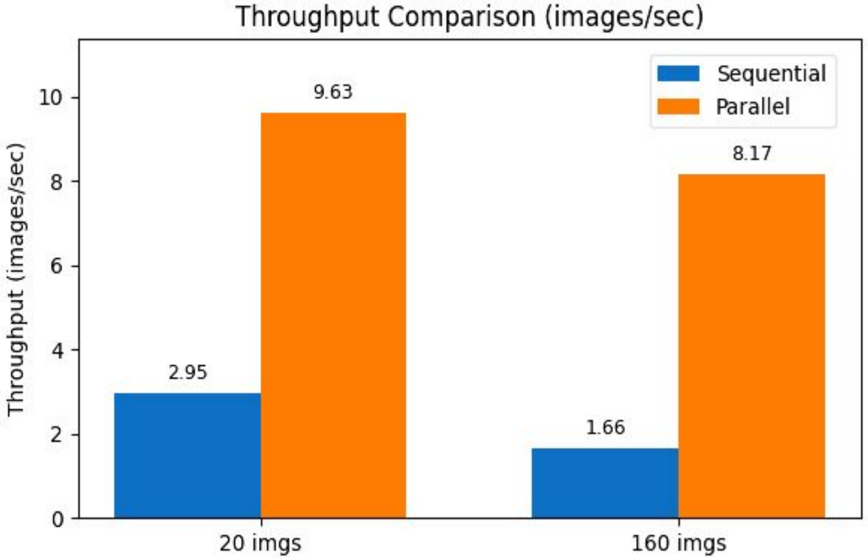
```
[TRIAL 1] seq=115.41s | par=25.53s | speedup=4.52x
[TRIAL 2] seq=104.15s | par=24.00s | speedup=4.34x
[TRIAL 3] seq=68.79s | par=9.25s | speedup=7.43x
==========================================
      BENCHMARK SUMMARY
==========================================
Sequential: median 104.15s (min 68.79s, max 115.41s, n=3)
Parallel: median 24.00s (min 9.25s, max 25.53s, n=3)
Speedup: median 4.52x (n=3)
Efficiency: median 56.5% (Speedup/Cores)
CSV report: benchmark_results.csv
==========================================
```

56.5% Parallel Efficiency aligns closely with the benchmark of **53.4%** (Fauzie et al., 2023), validating that our distributed overhead is standard.

Stability: Sequential throughput collapsed by **44%** under load, while Parallel retained **85%** of peak performance.

Phase 1 vs. Phase 2 Results





Parallel Speedup: **3.57x** **vs** **4.52x**

Fauzie et al.(2023) observed that parallel processing offers the most significant gains on datasets exceeding 100MB, suggesting that the **image batching strategy** was essential to overcome the overhead of smaller individual files.

Amdahl's Law for Validation

$$s = \frac{1}{(1-p) + \frac{p}{N}}$$

Phase 1 (20 img, batch=1, N=8 & s= 3.57x)

Solving for p:

$$\frac{1}{s} = 1 - p\left(1 - \frac{1}{N}\right) \implies p = \frac{1 - \frac{1}{s}}{1 - \frac{1}{N}}$$

$$p = \frac{1 - \frac{1}{3.57}}{1 - \frac{1}{8}} = \frac{0.7199}{0.875} \approx \mathbf{0.8227} \ (82.3\%)$$

$$(1-p): 1 - 0.8227 = \mathbf{0.1773} \ (17.7\%)$$

The **17.7%** serial overhead indicates significant latency from the scheduler and rapid task switching.

Phase 2 (160 img, batch=4, N=8 & s= 4.52x)

Solving for p:

$$p = \frac{1 - \frac{1}{4.52}}{1 - \frac{1}{8}} = \frac{0.7788}{0.875} \approx \mathbf{0.8900} \ (89.0\%)$$

$$(1-p): 1 - 0.8900 = \mathbf{0.1100} \ (11.0\%)$$

Optimization reduced the serial bottleneck (1-p) by ~38% (**from 17.7% to 11.0%**). This remaining 11% represents the hardware's hard I/O limit that can't be parallelised.

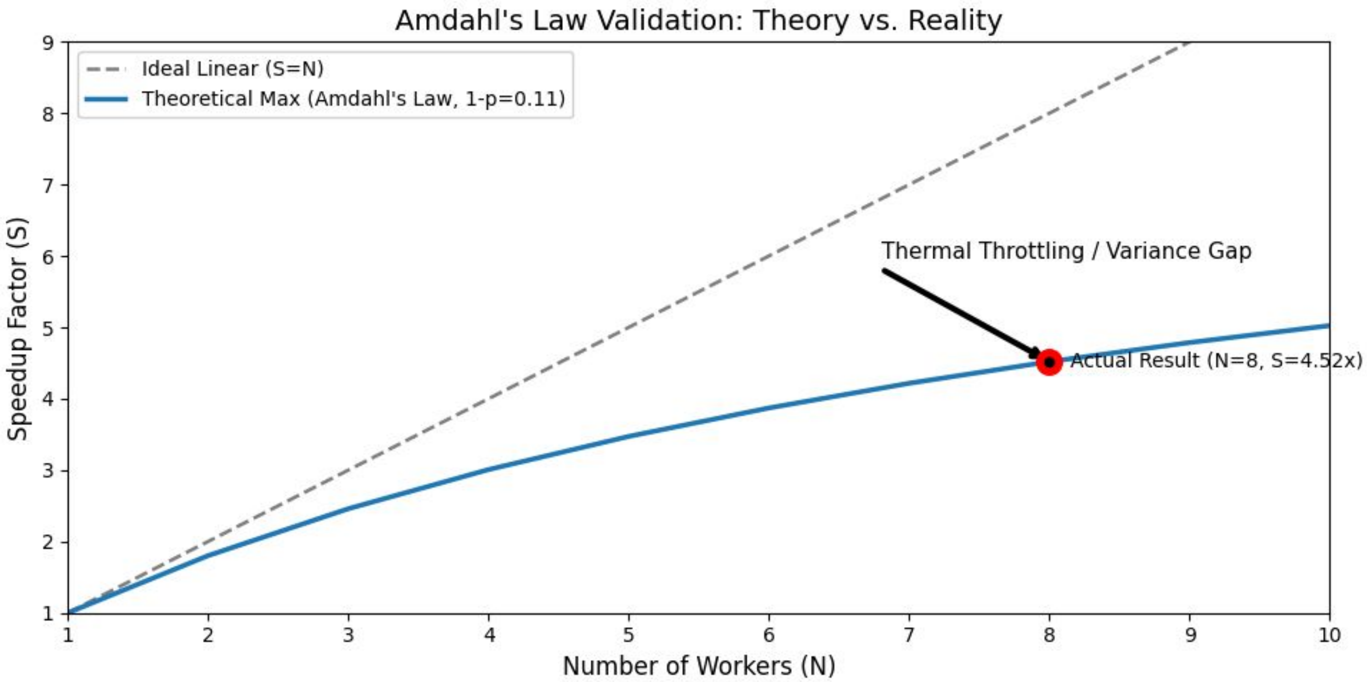
Amdahl’s Law for Validation

0.11 (1 - p): The 11% of time spent on *non-parallel* tasks (Disk I/O, Scheduler overhead)

0.89 (p): The 89% of time spent on *parallel* tasks (Gaussian Blur)

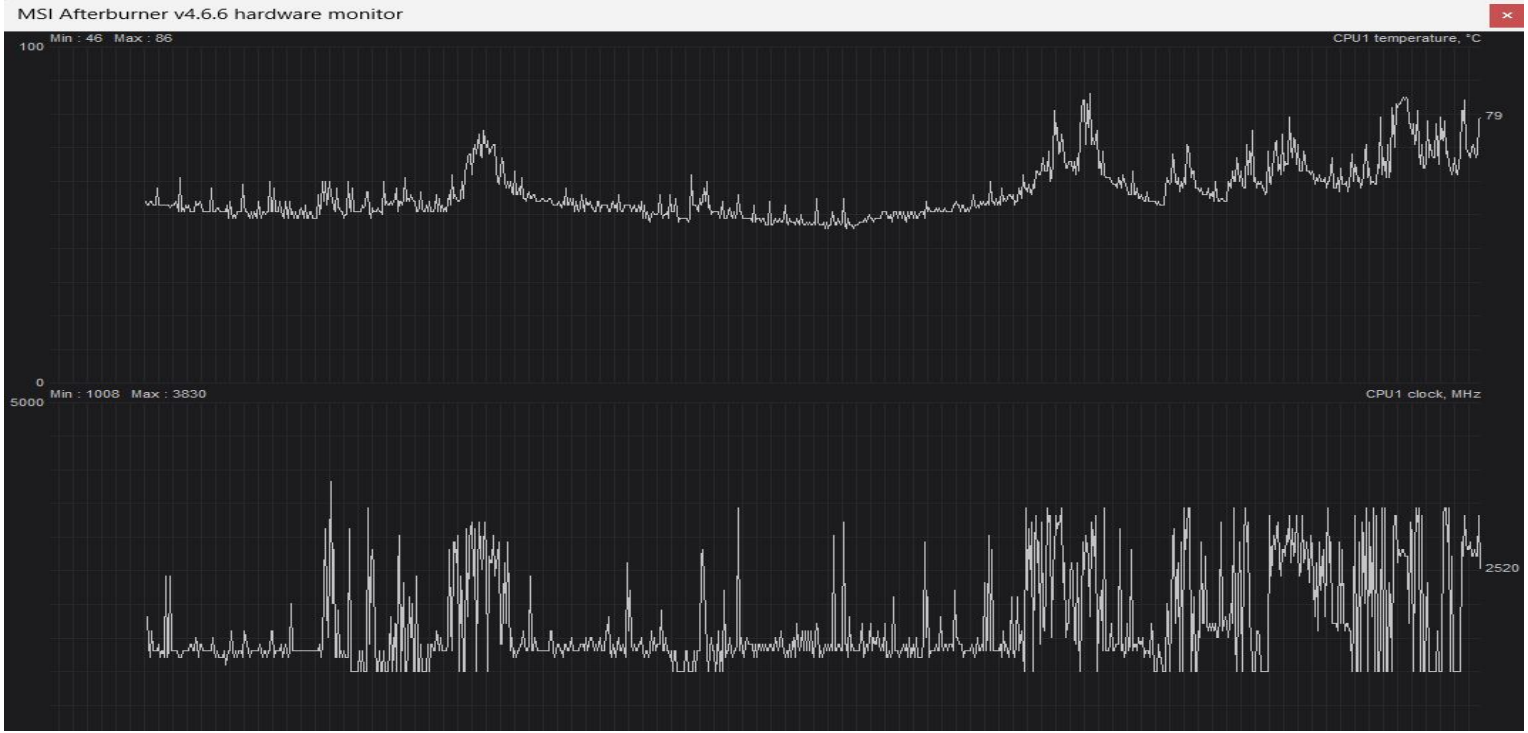
The absolute theoretical speedup limit according to Amdahl’s law:

$$\boxed{S_{max} = \frac{1}{0.11}} \quad \boxed{\approx 9.0909...}$$



Hence, the speedup obtained (**4.52x**) is **validated** because it adheres to the theoretical ceiling established by Amdahl's Law (**9.1x**)

Limitations and Constraints



Hardware monitoring confirms **thermal throttling** as CPU temperature peaks at **86°C**, the clock speed forcibly drops from **3.83 GHz** to **2.52 GHz** to prevent overheating which directly causes the mentioned performance variance.

Limitations and Constraints

1. **I/O Saturation (The 11% Limit):**

   Primary performance bottleneck is because of the hard drive (Disk I/O). **Ibrahim et al.(2021)** classify Gaussian Blur as a highly 'memory-intensive' algorithm, so the observed I/O saturation is **valid**.

2. **Thermal Throttling:**

   Performance drops after the first run because the laptop heats up.
   The CPU automatically slows down to protect itself, causing runtime to fluctuate between **9.25s** (cold) and **24s** (hot).

3. **Startup Overhead:**

   Creating a new worker cluster for every single batch adds a fixed delay.

Future Work

1. Fix I/O Bottleneck through Asynchronous Processing:

Workers effectively send processed data into a memory queue and a separate, dedicated thread handles the slow disk writing in the background,

2. Eliminate Overheating through Cloud Deployment:

Migrate from a *LocalCluster* (single laptop) to a **Cloud-Based Cluster** (e.g., Dask Kubernetes) to completely eliminate the single-machine heat limit.

3. Reduce Startup Delay:

Keep the worker cluster running in the background instead of restarting it for every new task so that the latency of bootstrapping workers for every new batch is removed.

References

Fauzie, A. N., Sakti, S. P., & Rahmadwati. (2023). Parallel implementation of Gaussian filter image processing on a cluster of single board computer. *Jurnal EECCIS*, *17*(3), 82–88.

DOI: https://doi.org/10.21776/jeeccis.v17i3.1672

Ibrahim, N. M., ElFarag, A. A., & Kadry, R. (2021). Gaussian blur through parallel computing. In *Proceedings of the International Conference on Image Processing and Vision Engineering* (pp. 175–179). SCITEPRESS – Science and Technology Publications.

DOI: https://dl.acm.org/doi/10.5220/0010513301750179

MSI Afterburner. Micro-Star International Co., Ltd. Available at: https://www.msi.com/Landing/afterburner/graphics-cards