Problem Statement: GPU-Accelerated String Matching

Traditional CPU-based pattern matching (e.g., searching for multiple keywords in large text datasets) often becomes a bottleneck in data-intensive applications like intrusion detection, spam filtering, and big data analytics. To boost performance, this project aims to implement string matching on a GPU, leveraging parallel kernels to scan large volumes of text at high speed.

Objective

• **Develop a GPU-based pattern-matching system** using an algorithm such as Aho-Corasick (or another advanced multi-pattern approach).

• **Demonstrate high-throughput scanning** on large text datasets, surpassing CPU-only solutions.

Key Requirements

1. **Core Algorithm**: Must implement a robust pattern-matching algorithm (Aho-Corasick, Knuth-Morris-Pratt, or others) adapted for parallel processing.

2. **GPU Utilization**: Must use GPU kernels effectively, handling data transfer overhead and concurrency.

3. **Scalability**: Handle large-scale text corpora (millions of characters) with minimal performance degradation.

4. **Metrics & Benchmarking**: Provide clear comparisons (CPU vs. GPU throughput, latency) to validate performance gains.

5. **Memory Management**: Optimize memory layouts (global, shared) for peak GPU performance.

Constraints

• **No Specialized Hardware** Beyond a standard GPU environment (e.g., CUDA/OpenCL).

• **Accuracy** Ensure correctness for all matched patterns; no false positives/negatives.

• **Time Limit** Must show a functioning prototype within the hackathon’s timeframe (24 hours or as specified).

Deliverables

• **Working Prototype**: Demonstrate the GPU-accelerated matching on sample or synthetic datasets.

• **Performance Report**: Document speedups, resource usage, and any limitations or optimizations.

• **Source Code & README**: Provide well-structured code and instructions to run, compile, and test.

Impact

A successful solution will empower large-scale text analytics, security scanning, and other data-intensive workflows by harnessing the parallel nature of GPUs, reducing processing time, and improving overall system efficiency.

Success Criteria:

You might assume a scale like:

• Text Size: 1–5 GB in total (millions of lines or words).

• Patterns: 100–1,000 search terms (or more).

• Throughput Goal: At least several MB/s to GB/s of scanning, aiming for noticeable speedups (2x–10x) compared to a CPU baseline.

• Time Target: Processing a 1GB text file in under a few seconds, depending on GPU power.

These assumptions make the challenge non-trivial yet feasible for a GPU-accelerated string-matching prototype in a hackathon.