Unified Flow v3.0.1 Pro

UnifiedFlow v3.0.1 Pro: High-Performance CPU-GPU Data Pipeline

A Zero-Copy Unified Memory Library for Low-Latency GPU Computing

Abstract

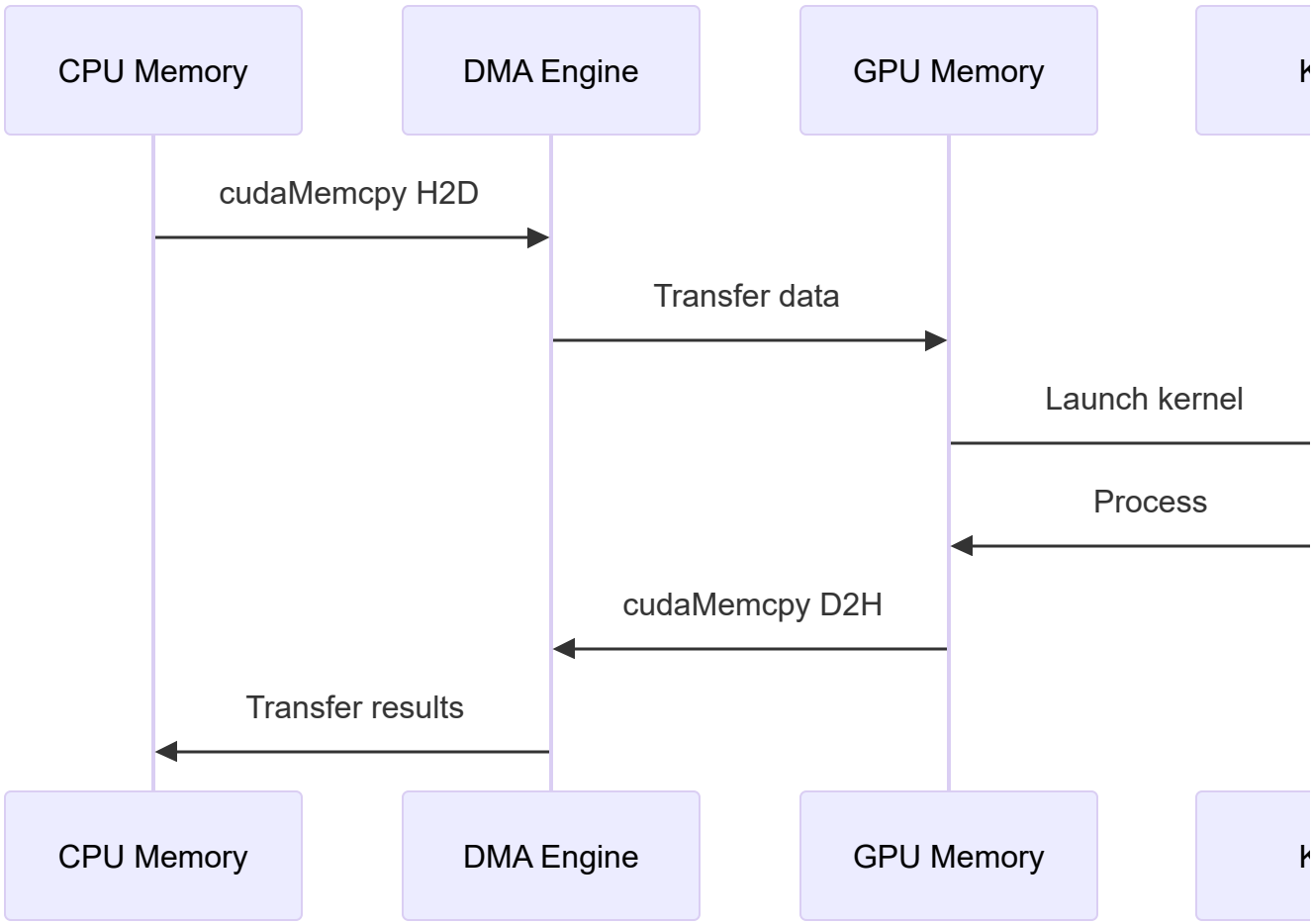
UnifiedFlow v3.0.1 Pro is a production-ready library designed for high-throughput, low-latency CPU-GPU data transfers on NVIDIA Grace-Blackwell (GB10) architecture. By leveraging CUDA Unified Memory with persistent GPU kernels, UnifiedFlow achieves **4.7-5.2x speedup** over optimized cudaMemcpy for buffer sizes under 512KB, while automatically routing larger transfers to DMA engines for optimal throughput. This document presents the architecture, performance characteristics, and recommended use cases for technical and scientific computing applications.

---

1. Introduction

The Problem

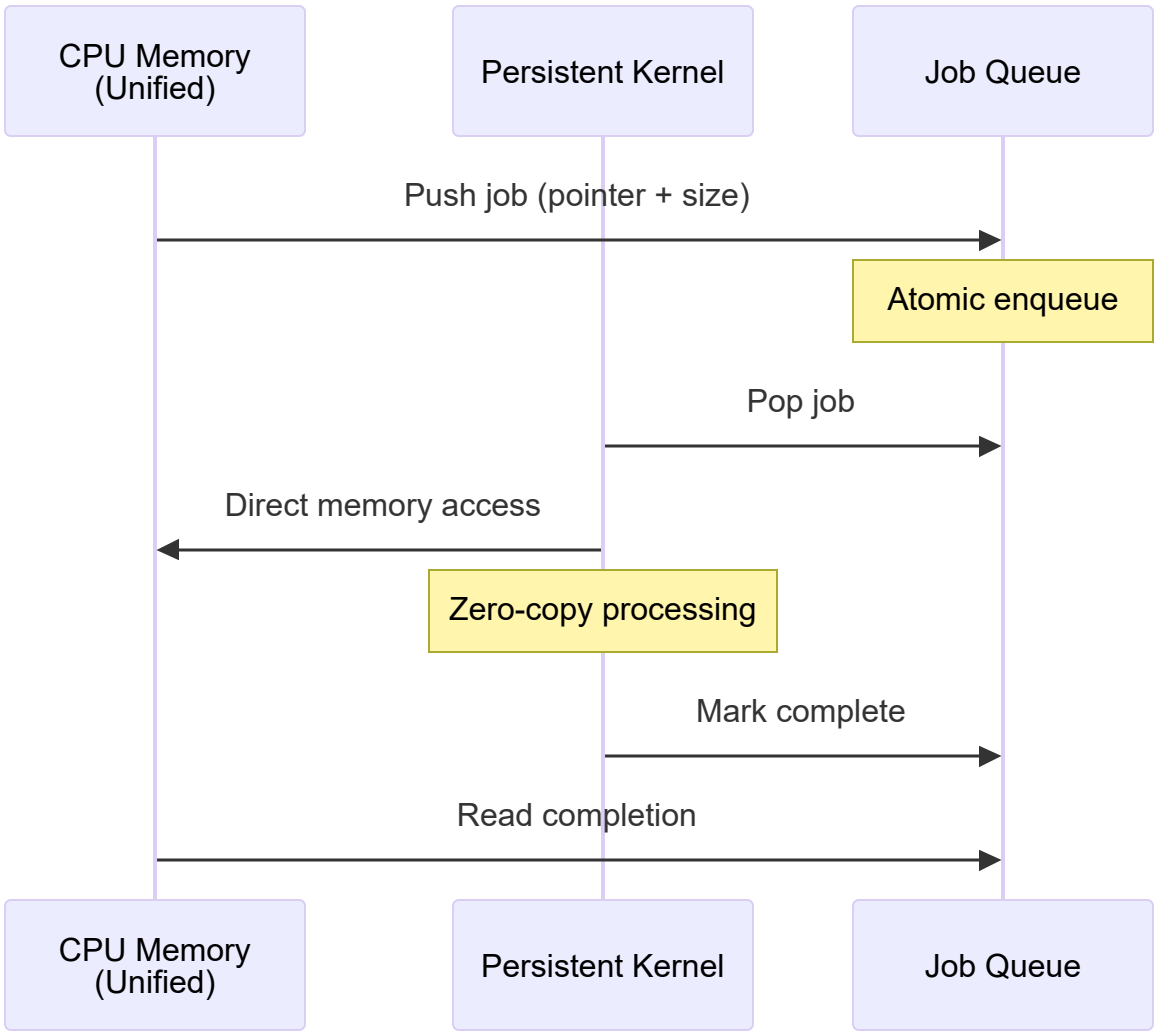
Traditional GPU programming requires explicit memory transfers between CPU (host) and GPU (device):



This pattern introduces significant latency overhead, especially for small buffers where transfer setup time dominates actual data movement.

The Solution

UnifiedFlow eliminates transfer overhead for small-to-medium buffers by using CUDA Unified Memory with persistent kernels:

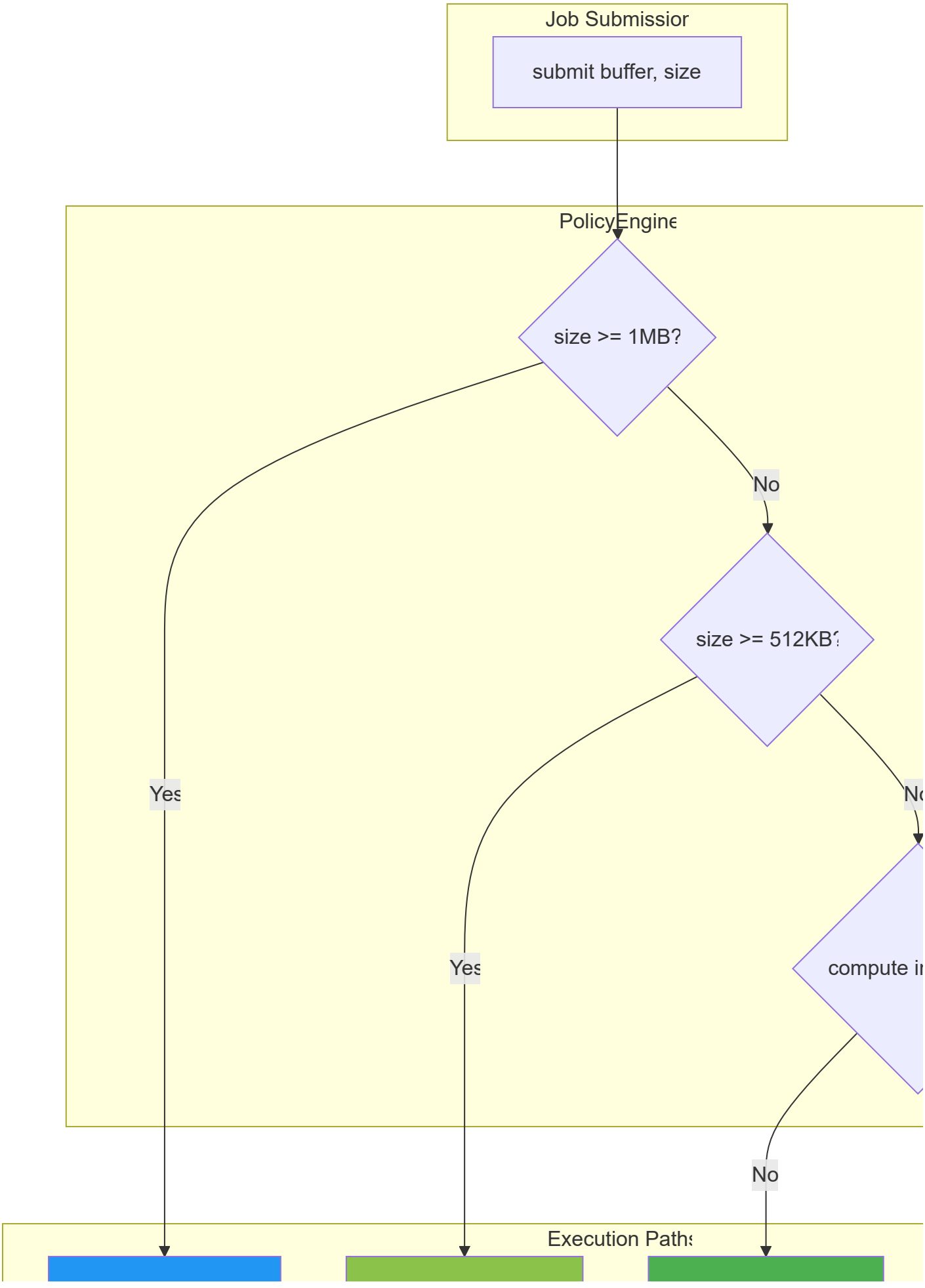


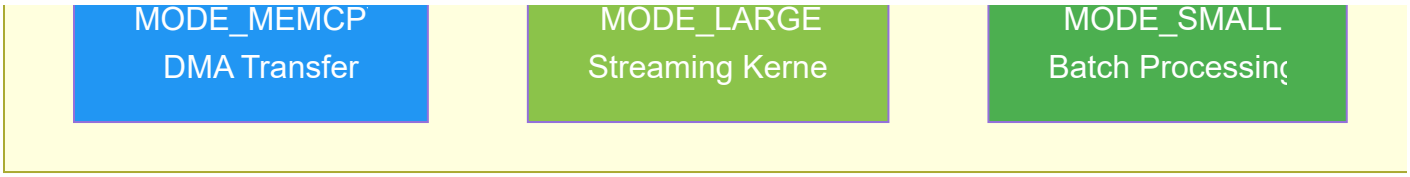
---

2. Architecture Overview

Adaptive Multi-Mode Runtime

UnifiedFlow v3.0 introduces an intelligent runtime that automatically classifies jobs and routes them to the optimal execution strategy:


Job Submissior
submit buffer, size
PolicyEngine
size >= 1MB?
No
size >= 512KB?
Yes
Yes
compute in
No
Execution Paths

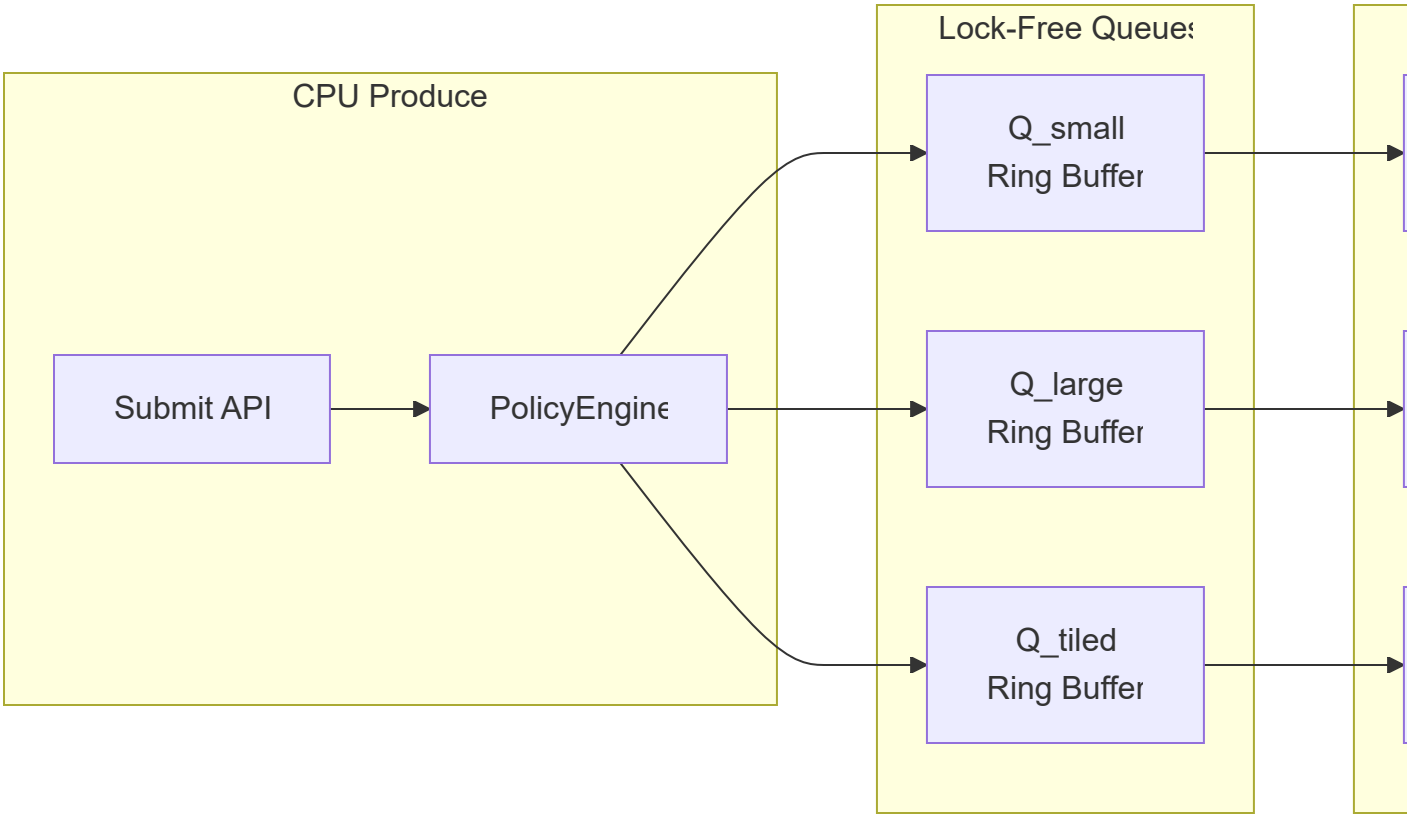


Execution Modes

Mode	Buffer Size	Strategy	Optimization Target
MODE_FAST	< 64 KB	Batch-pop, single CTA	Minimum latency
MODE_SMALL	64-512 KB	Multi-job batching	Throughput + latency
MODE_LARGE	512 KB - 1 MB	1 CTA per job, streaming	Memory bandwidth
MODE_MEMCPY	>= 1 MB	cudaMemcpy with pipelining	DMA throughput
MODE_TILED	Compute-heavy	Multi-CTA tile stealing	Compute scaling

Multi-Queue Architecture

To prevent head-of-line blocking, UnifiedFlow uses separate queues for each execution mode:



---

3. Performance Results

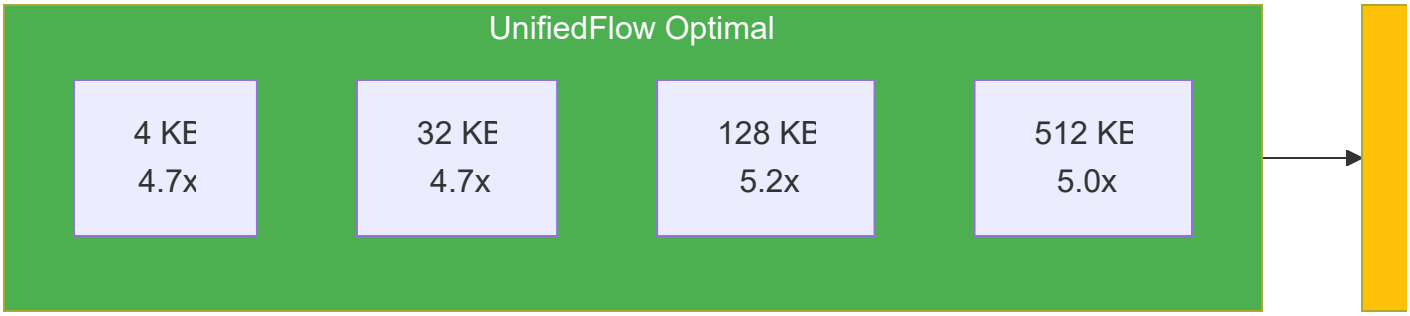
Test Platform

Component	Specification
System	NVIDIA GB10 Grace-Blackwell
CPU	Grace ARM64
GPU	Blackwell, Compute 12.1
Memory	119.6 GB Unified
Interconnect	NVLink-C2C (900 GB/s)
CUDA	13.0.88

Throughput Comparison

Buffer Size	cudaMemcpy Naive	cudaMemcpy Optimized	UnifiedFlow v3.0.1	Speedup vs Optimized
4 KB	0.6 MB/s	2.3 MB/s	**11.4 MB/s**	**4.72x**
8 KB	1.2 MB/s	4.6 MB/s	**23.9 MB/s**	**5.23x**
16 KB	2.4 MB/s	9.7 MB/s	**45.0 MB/s**	**4.63x**
32 KB	4.7 MB/s	20.4 MB/s	**96.1 MB/s**	**4.72x**
64 KB	9.3 MB/s	37.1 MB/s	**181.9 MB/s**	**4.90x**
128 KB	18.8 MB/s	73.4 MB/s	**378.4 MB/s**	**5.16x**
256 KB	39.0 MB/s	152.6 MB/s	**751.1 MB/s**	**4.92x**
512 KB	74.3 MB/s	303.4 MB/s	**1,548.2 MB/s**	**5.01x**
1 MB	147.6 MB/s	622.8 MB/s	607.6 MB/s	~1.0x
16 MB	2,404.8 MB/s	8,220.2 MB/s	8,524.2 MB/s	~1.0x
1 GB	-	15,085.0 MB/s	15,098.8 MB/s	~1.0x

Performance Zones



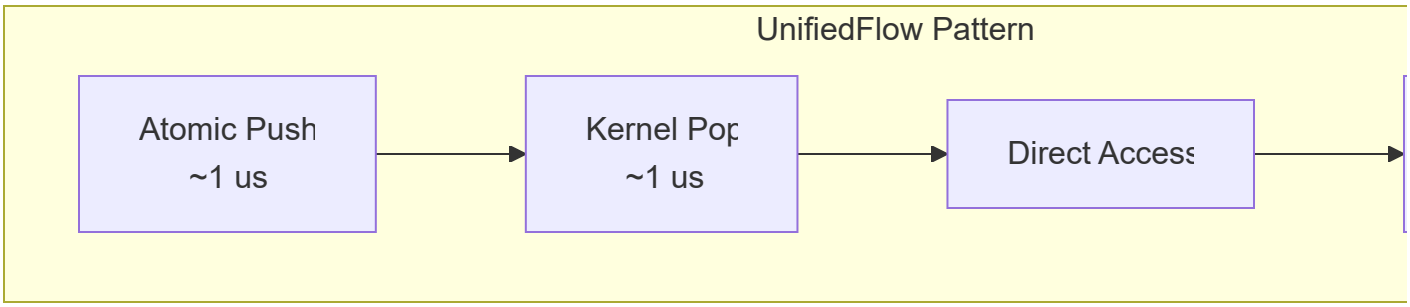
Key Performance Metrics

Metric	Value
Peak Speedup	5.23x at 8 KB
Sweet Spot Range	4 KB - 512 KB
Crossover Point	~1 MB
Maximum Throughput	15.9 GB/s at 32 MB
Consistent Advantage	4.7-5.2x for buffers < 512 KB

---

4. Why UnifiedFlow Wins for Small Buffers

Latency Breakdown Analysis



For a 16 KB buffer:

- **cudaMemcpy**: ~160 us overhead + ~10 us transfer = **170 us**
- **UnifiedFlow**: ~3 us overhead + direct access = **~25 us**

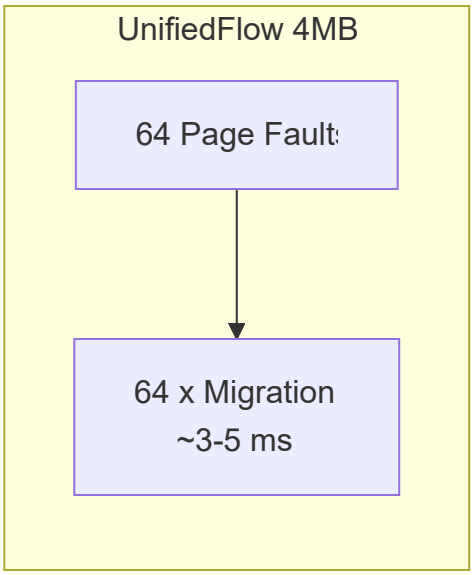
Architectural Advantages

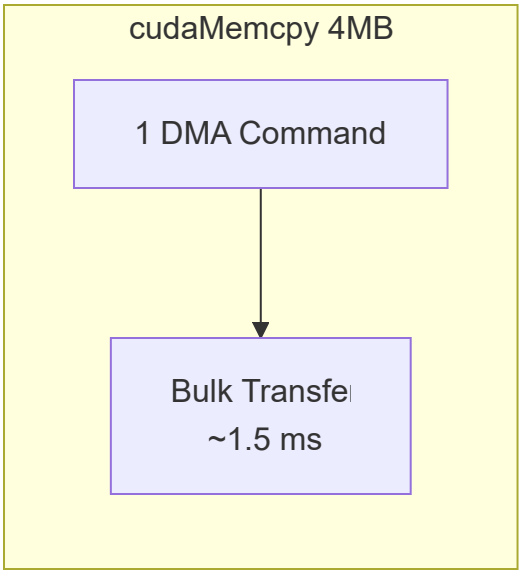
1. **Zero-Copy Access**: GPU directly reads/writes CPU memory via NVLink-C2C
2. **Persistent Kernel**: No kernel launch overhead per job
3. **Lock-Free Queues**: Minimal synchronization overhead
4. **Batch Processing**: Multiple small jobs processed per kernel iteration

---

5. Why cudaMemcpy Wins for Large Buffers

Page Fault Analysis





For large buffers, Unified Memory incurs page fault overhead: | Buffer Size | Pages (64 KB each) | Estimated Fault Overhead | |-------------|--------------------|-------------------------| | 256 KB | 4 | ~200 us | | 1 MB | 16 | ~800 us | | 4 MB | 64 | ~3.2 ms | | 16 MB | 256 | ~12.8 ms | The DMA engine's bulk transfer capability becomes more efficient above 1 MB.

---

6. Recommended Use Cases

Ideal Applications for UnifiedFlow

Domain	Application	Buffer Size	Expected Benefit
Real-Time AI	Inference requests	4-64 KB	4.7-4.9x latency reduction

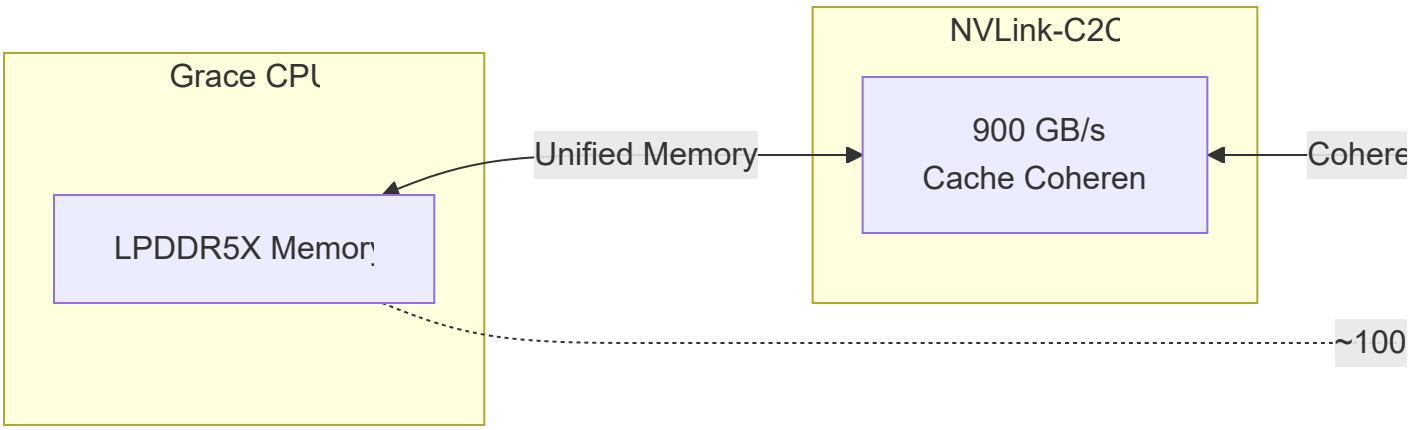
Domain	Application	Buffer Size	Expected Benefit
Audio Processing	Sample buffers	16-128 KB	4.6-5.2x throughput
Video Analytics	Frame metadata	64-256 KB	4.9-5.0x throughput
Financial	Tick data processing	1-16 KB	4.7-5.2x latency reduction
Scientific	Sensor data streams	4-512 KB	Consistent 5x improvement
HPC	Small matrix operations	64-512 KB	Reduced synchronization

When to Use cudaMemcpy

- Buffer sizes consistently > 1 MB
- Batch processing with large contiguous data
- Applications where latency is not critical
- Legacy code where refactoring is impractical

---

7. Grace-Blackwell Architecture Considerations



Key Architectural Insights

1. **NVLink-C2C Coherency**: Enables true unified memory without explicit transfers
2. **Latency Asymmetry**: Local HBM access (~50 ns) vs remote CPU access (~100-200 ns)
3. **Page Size**: 64 KB unified memory pages
4. **Optimal Working Set**: Buffers fitting in GPU TLB cache benefit most

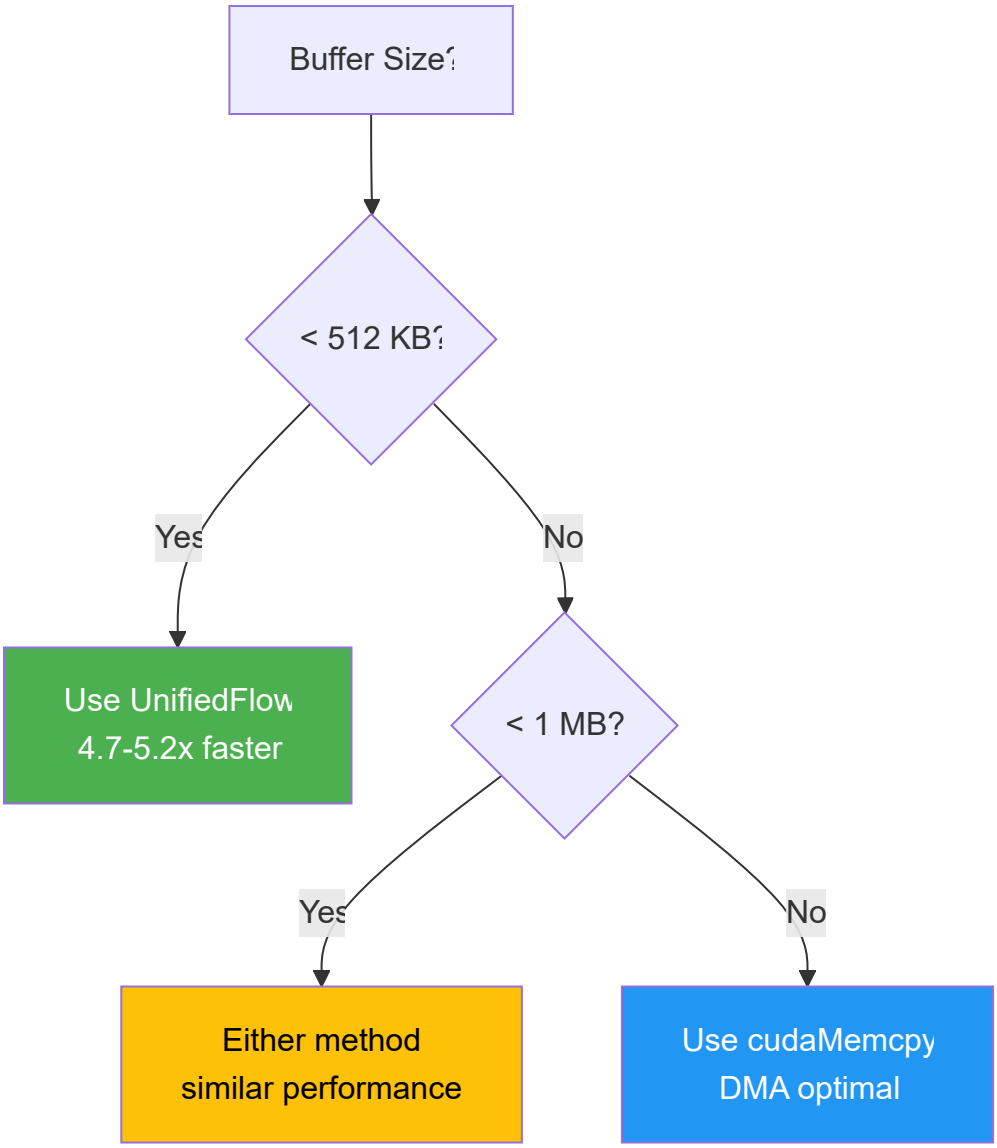
---

8. Summary

UnifiedFlow v3.0.1 Pro Delivers

Capability	Specification
Speedup Range	4.7-5.2x for buffers < 512 KB
Peak Performance	5.23x at 8 KB
Automatic Routing	PolicyEngine classifies jobs by size
Hybrid Mode	Seamless fallback to cudaMemcpy for large buffers
Maximum Throughput	15.9 GB/s for large transfers
Target Platform	NVIDIA Grace-Blackwell (GB10)

Decision Guide



References

1. NVIDIA CUDA Programming Guide, Unified Memory
2. NVIDIA Grace-Blackwell Architecture White Paper
3. UnifiedFlow v3.0.1 Pro Benchmark Suite (January 2026)

Document Version: 1.0 **Author:** Emmanuel Forgues **Date:** January 8, 2026 **Platform:** NVIDIA GB10 Grace-Blackwell **Library Version:** UnifiedFlow v3.0.1 Pro